编号：W473-CN5-01

SYSMAC CJ 系列

CJ2H-CPU□□-EIP

CJ2 CPU单元软件

操 作 手 册

编号：W473-CN5-01

SYSMAC CJ 系列 CJ2 CPU 单元软件

操作手册

OMRON

OMRON

特约经销商

编号：W473-CN5-01

注：规格等随时可能变化，恕不另行通知。

上海印刷
200903S01

© OMRON, 2008

版权所有，事先未经欧姆龙公司书面许可，本手册中的任何部分不可用任何形式，或用任何手段，机械的、电子的、照相、录制或其他方式进行复制、存入检索系统或传送。

使用本手册所包含的信息不负专利责任。然而，因为欧姆龙公司不断努力改进其高质量的产品，所以本手册中所含有的信息可能随时改变而不另行通知。在编写本手册时，注意了一切可能的注意事项，对于仍然可能出现的错误或遗漏欧姆龙公司将不承担责任，对于使用本手册中所包含的信息导致的损害也将不承担任何责任。

SYSMAC CJ 系列
CJ2H-CPU □□ -EIP
CJ2 CPU 单元软件

操作手册

2008 年 7 月修订

引言

感谢您购买CJ2H-CPU □□ -EIP CJ 系列CPU 单元。本手册包含使用CJ2H-CPU □□ -EIP 所必须的信息。在使用 CJ2H-CPU □□ -EIP 前请通读并理解本手册。

面向的读者

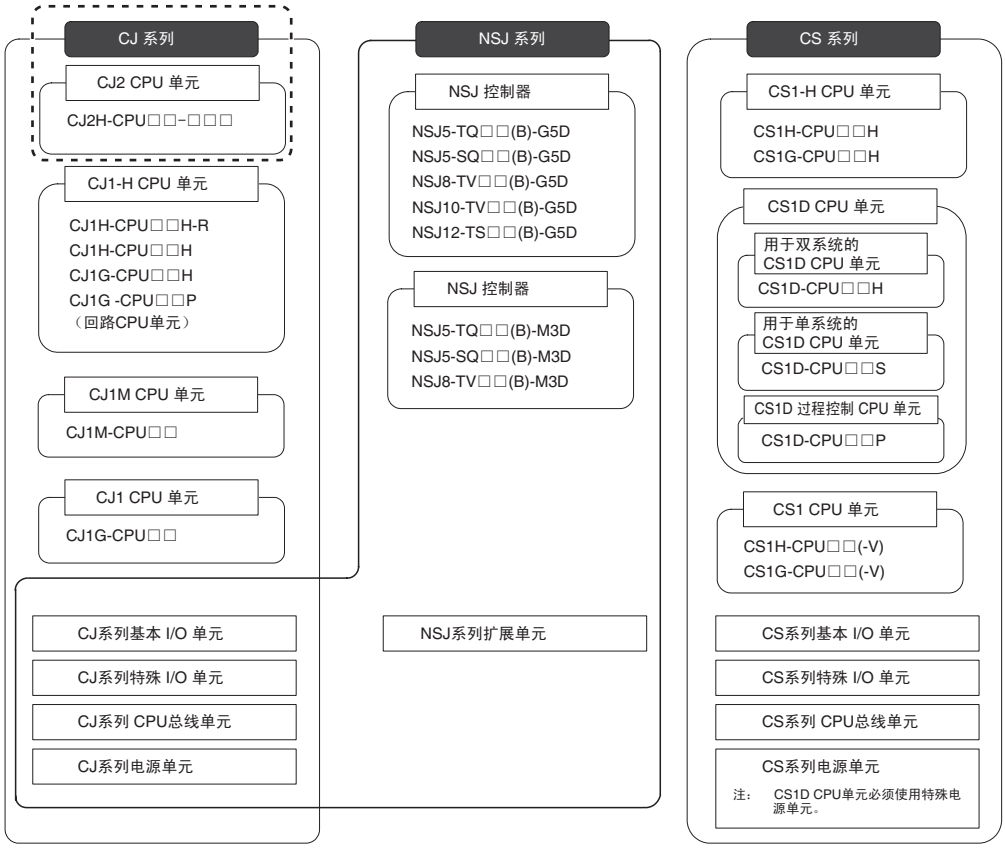
本手册是为下列人员编写的，他必须具有电气系统知识 （电气工程师或同等水平者）。

- 从事 FA 系统安装的人员。
- 从事 FA 系统设计的人员。
- 从事 FA 系统及设备管理的人员。

适用的产品

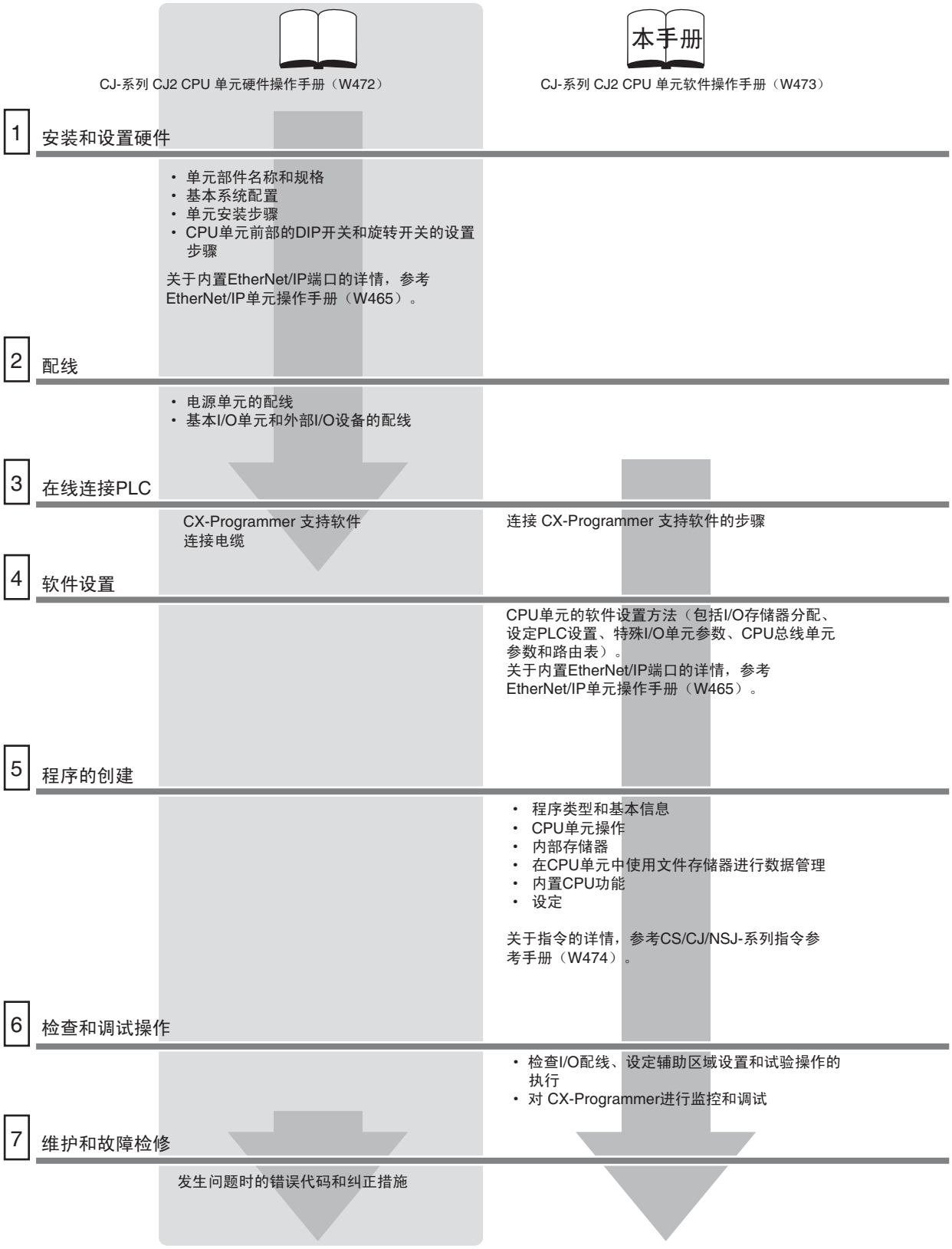
● CJ 系列的定义

在本手册中，CJ 系列是一个指代 CJ2 CPU 单元、CJ1-H CPU 单元、CJ1M CPU 单元和 CJ1 CPU 单元的通用术语。



CJ2 CPU 单元手册

以下手册中提供 CJ2 CPU 单元的信息。关于所需信息请参考相应的手册。



章节构成

CJ2 CPU 手册由下述章节构成。请根据需要参见相关章节。

硬件操作手册 （W472）

章节	内容
第一章 概述	本章节中阐述了 CJ2 CPU 单元的概述及其特性和规格。
第二章 基本系统配置和设备	本章节中阐述了 CJ2 CPU 单元的系统配置。
第三章 构成与功能	本章节中阐述了 CPU 单元及构成单元的各部分名称与功能。
第四章 支持软件	本章节中阐述了 CJ2 PLC 的编程用软件及其调试的相关内容，以及如何连接编程设备与 PLC。
第五章 安装	本章节中阐述了如何安装与接线 CJ2 PLC。
第六章 故障检修	本章节中阐述了如何处理 CJ2 PLC 运行时出现的故障检修问题。
第七章 检查和维护	本章节中阐述了 CJ2 PLC 所需的定期检查及维护信息 （如电池及电源使用寿命、以及如何更换电池等）。
第八章 备份操作	本章节中阐述了用于数据备份和恢复的操作。
附录	附录中提供了单元规格、单元尺寸、致命 / 非致命错误的详细信息、连接到 CPU 单元串行端口的信息、在计算机上安装 USB 驱动的步骤、以及负载短路保护和线路断开检测的信息。

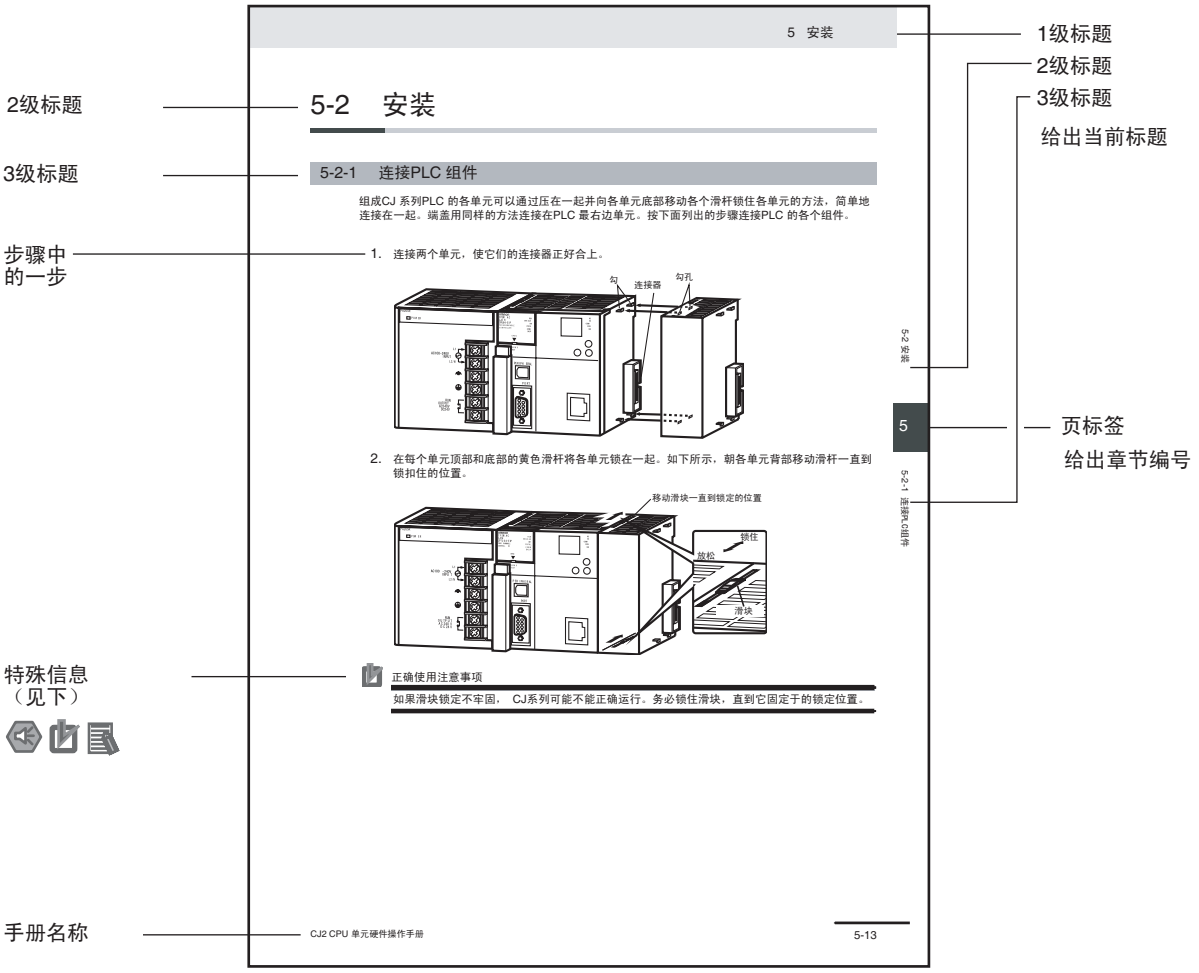
软件操作手册 （W473）（本手册）

章节	内容
第一章 概述	本章节中阐述了 CJ2 CPU 单元的概述及其特性和规格。
第二章 CPU 单元内部存储器	本章节中阐述了 CPU 单元中的存储器类型及数据存储。
第三章 CPU 单元运行	本章节中阐述了 CPU 单元的内部运行。
第四章 CPU 单元初始设定	本章节中阐述了 CPU 单元的初始设定。
第五章 编程	本章节中阐述了程序类型及编程的细节，例如，符号及编程指令。
第六章 I/O 存储器区	本章节中阐述了 CPU 单元中的 I/O 存储器区。
第七章 文件操作	本章节中阐述了可在 CPU 单元中保存文件详情及其存储目的地和文件操作。
第八章 I/O 分配与单元设定	本章节中阐述了用于 CPU 单元和其他单元之间数据交换的 I/O 分配详情。
第九章 PLC 设置	本章节中阐述了设定 PLC 设置 （用于执行 CPU 单元的基本设定）的详情。
第十章 CPU 单元功能	本章节中阐述了 CPU 中的内置功能。
第十一章 编程设备与通信	本章节中阐述了连接 CJ2 CPU 单元与 CX-Programmer 或其他支持软件或设备的步骤。
第十二章 CPU 单元循环时间	本章节中阐述了如何监控和计算循环时间。
附录	附录中提供了编程指令、指令执行时间、步数、辅助区字和位、 PLC 存储器地址的实际有效存储器分配、电源中断时 I/O 存储器运行以及 CJ 系列与 CS 系列 PLC 比较的详细信息。

手册结构

页面结构

本手册使用以下的页面结构。



本说明仅提供一个样本，文字内容可能不会出现在本手册中。

特殊信息

本手册中的特殊信息分类如下：

安全使用注意事项

关于该做什么与不该做什么的注意事项以保证产品的安全使用。

正确使用注意事项

关于该做什么与不该做什么的注意事项以保证正确的操作和运行。

附加信息

增加理解或进行简便操作的附加信息。

本手册中的章节

1 概述

2 CPU单元内部存储器

3 CPU单元运行

4 CPU单元初始设定

5 编程

6 I/O存储器区

7 文件操作

8 I/O分配与单元设定

9 PLC设置

10 CPU单元功能

11 编程设备与通信

12 CPU单元循环时间

A 附录

目录

引言 1

CJ2 CPU 单元手册 2

手册结构 4

本手册中的章节 5

安全注意事项 19

应用注意事项 25

运行环境注意事项 29

规定及标准 30

相关手册 32

第 1 章 概述

1-1 CJ2 CPU 单元概述 1-2
1-1-1 概述 1-2
1-1-2 CJ2 CPU 单元特长 1-3
1-2 基本操作步骤 1-7

第 2 章 CPU 单元内部存储器

2-1 概述 2-2
2-1-1 存储器配置 2-2
2-1-2 存储器区和保存的数据 2-3
2-1-3 从编程设备向 CPU 单元传送数据 2-4

第 3 章 CPU 单元运行

3-1 CPU 单元内部运行 3-2
3-1-1 概述 3-2
3-1-2 循环时间 3-3
3-1-3 电源中断时的处理 3-5
3-2 CPU 单元运行模式 3-7
3-2-1 运行模式 3-7
3-2-2 运行模式的检查 3-7
3-2-3 运行模式的变更 3-8
3-2-4 运行模式的详细信息 3-12

第 4 章 CPU 单元初始设定

4-1 CPU 单元初始化概述 4-2
4-1-1 CPU 单元初始设定 4-2
4-2 PLC 设置 4-7
4-2-1 PLC 的缺省设定 4-7

4-3 I/O 表的创建 4-8
4-3-1 I/O 表 4-8
4-3-2 自动分配 4-9
4-3-3 手动分配 4-9
4-4 路由表的设定 4-10
4-4-1 路由表 4-10
4-4-2 需要路由表的情况 4-11
4-4-3 路由表的设定与传送 4-12
4-5 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字 4-14
4-5-1 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字 4-14
4-5-2 设定步骤 4-14
4-6 CPU 总线单元设置区 4-15
4-6-1 CPU 总线单元设置区 4-15
4-6-2 设定步骤 4-15

第 5 章 编程

5-1 编程 5-3
5-1-1 编程概述 5-3
5-1-2 梯形图基本概念 5-5
5-1-3 ST 语言 5-6
5-1-4 SFC 概述 5-7
5-2 任务 5-9
5-2-1 任务概述 5-9
5-2-2 循环任务 5-12
5-2-3 中断任务 5-17
5-2-4 设计任务 5-26
5-3 段 5-36
5-3-1 段概述 5-36
5-4 功能块 5-38
5-4-1 功能块 5-38
5-4-2 功能块特性 5-38
5-4-3 功能块规格 5-40
5-5 符号（变量） 5-41
5-5-1 概述 5-41
5-5-2 符号（变量）类型 5-41
5-5-3 全局符号（变量） 5-43
5-5-4 本地符号（变量） 5-43
5-5-5 网络符号（变量） 5-44
5-5-6 功能块中的变量 5-47
5-5-7 符号（变量）数据类型 5-48
5-5-8 符号（变量）的自动地址分配 5-50
5-6 指令 5-51
5-6-1 指令的基本理解 5-51
5-6-2 定义操作数 5-57
5-6-3 数据格式 5-65
5-6-4 I/O 刷新时序 5-68
5-7 变址寄存器 5-74
5-7-1 什么是变址寄存器？ 5-74
5-7-2 使用变址寄存器 5-74
5-7-3 与变址寄存器相关的处理 5-78
5-7-4 监控变址寄存器 5-78
5-7-5 在任务之间共享变址寄存器和数据寄存器 5-80
5-8 指定地址偏移 5-82
5-8-1 概述 5-82
5-8-2 地址偏移应用的示例 5-84

5-9 程序的检查 5-86
5-9-1 CX-Programmer 输入时出错 5-86
5-9-2 用 CX-Programmer 检查程序 5-86
5-9-3 用仿真工具调试 5-86
5-9-4 程序执行检查 5-88
5-10 注意事项 5-92
5-10-1 条件标志 5-92
5-10-2 特殊程序段 5-96

第 6 章 I/O 存储器区

6-1 I/O 存储器区 6-2
6-1-1 I/O 存储器区概述 6-2
6-1-2 I/O 存储区结构 6-4
6-1-3 保持 I/O 存储器值 6-5
6-2 I/O 区 6-7
6-2-1 输入位 6-7
6-2-2 输出位 6-9
6-3 数据链接区 6-12
6-4 CPU 总线单元区 6-13
6-5 特殊 I/O 单元区 6-14
6-6 DeviceNet 区 6-15
6-7 工作区 6-16
6-8 保持区 6-17
6-9 辅助区 6-19
6-10 暂存继电器区 6-20
6-11 数据存储器区 6-21
6-12 扩展数据存储器区 6-23
6-13 定时器区 6-25
6-14 计数器区 6-27
6-15 任务标志 6-28
6-16 变址寄存器 6-29
6-17 数据寄存器 6-34
6-18 条件标志 6-36
6-19 时针脉冲 6-38

第 7 章 文件操作

7-1 文件操作 7-2
7-1-1 文件存储器的类型 7-2
7-1-2 初始化文件存储器 7-3
7-1-3 存储卡注意事项 7-4
7-2 保存在文件存储器的文件类型 7-6
7-2-1 文件类型 7-6
7-2-2 创建和保存用于文件存储器的文件 7-9
7-3 文件存储操作 7-10
7-3-1 文件存储操作的类型 7-10
7-3-2 文件存储操作步骤和文件存储器文件 7-11
7-3-3 文件使用的限制 7-18
7-3-4 文件大小 7-18
7-3-5 支持软件和文件存储器文件之间的关系 7-20

第 8 章　I/O 分配和单元设定

8-1 I/O 分配 8-2
8-1-1 I/O 分配 8-2
8-1-2 自动分配 8-5
8-1-3 手动分配 8-9
8-1-4 I/O 表错误和注意事项 8-17
8-2 设定 CPU 总线单元和特殊 I/O 单元 8-20
8-2-1 设定参数 8-20
8-2-2 数据交换 8-24

第 9 章　PLC 设置

9-1 PLC 设置概述 9-2
9-2 PLC 设置设定 9-4
9-2-1 启动操作设定 9-4
9-2-2 CPU 单元设定 9-7
9-2-3 时间和中断设定 9-13
9-2-4 特殊 I/O 单元循环刷新 9-18
9-2-5 基本 I/O 单元机架响应时间 9-19
9-2-6 串行端口设定 9-21
9-2-7 外设服务 9-26
9-2-8 FINS 保护 9-27

第 10 章　CPU 单元功能

10-1 时钟功能 10-3
10-1-1 时钟功能 10-3
10-1-2 存储器中保存的时间数据 10-4
10-1-3 自由运行定时器 10-7
10-2 循环时间 / 高速处理 10-8
10-2-1 最小循环时间 10-8
10-2-2 最大循环时间 10-9
10-2-3 监控循环时间 10-10
10-2-4 高速输入 10-10
10-2-5 后台执行 10-11
10-3 启动设定和维护 10-20
10-3-1 启动时和运行模式变更时的保持设定 10-20
10-3-2 电源 OFF 检测延迟设定 10-21
10-3-3 禁止电源 OFF 中断 10-22
10-3-4 RUN 输出 10-23
10-3-5 在启动时自动传输 10-24
10-4 单元管理功能 10-31
10-4-1 基本 I/O 单元管理 10-31
10-4-2 CPU 总线单元标志 / 位 10-32
10-4-3 特殊 I/O 单元标志 / 位 10-33
10-5 存储器管理功能 10-35
10-5-1 自动备份 10-35
10-5-2 EM 文件存储器功能 10-36
10-5-3 注释存储器 10-36
10-5-4 操作中替换全部程序 10-37
10-6 安全功能 10-44
10-6-1 使用 DIP 开关的写保护功能 10-44
10-6-2 使用密码的读保护 10-44
10-6-3 使用产品批号进行程序操作保护 10-49
10-6-4 FINS 命令写保护 10-50
10-6-5 PLC 名称 10-53

10-7 调试 10-56
10-7-1 强制置位 / 复位 10-56
10-7-2 测试输入 10-57
10-7-3 微分监控 10-57
10-7-4 在线编辑 10-58
10-7-5 关闭输出 10-59
10-7-6 数据跟踪 10-60
10-7-7 存储错误时的停止位 10-64
10-7-8 故障报警指令 10-65
10-7-9 模拟系统错误 10-65
10-7-10 故障点检测 10-67

第 11 章 编程设备与通信

11-1 从 CX-Programmer 访问 PLC 11-2
11-1-1 概述 11-2
11-1-2 可访问 PLC 的系统配置 11-4
11-1-3 从 CX-Programmer 访问 PLC 11-8
11-1-4 自动在线连接 11-11
11-2 串行通信 11-15
11-2-1 串行通信概述 11-15
11-3 通信网络 11-21

第 12 章 CPU 单元循环时间

12-1 监控循环时间 12-2
12-1-1 监控循环时间 12-2
12-2 计算循环时间 12-3
12-2-1 CPU 单元操作流程图 12-3
12-2-2 循环时间概述 12-4
12-2-3 各单元的 I/O 单元刷新时间 12-6
12-2-4 循环时间计算实例 12-8
12-2-5 在线编辑导致的循环时间延长 12-10
12-2-6 I/O 响应时间 12-10

附录

A-1 指令功能 A-3
A-1-1 顺序输入指令 A-3
A-1-2 顺序输出指令 A-5
A-1-3 顺序控制指令 A-6
A-1-4 定时器和计数器指令 A-10
A-1-5 比较指令 A-14
A-1-6 数据传送指令 A-17
A-1-7 数据移位指令 A-20
A-1-8 递增 / 递减指令 A-24
A-1-9 四则运算指令 A-24
A-1-10 转换指令 A-30
A-1-11 逻辑指令 A-36
A-1-12 特殊算术指令 A-37
A-1-13 浮点数运算指令 A-39
A-1-14 双精度浮点数指令 A-43
A-1-15 表格数据处理指令 A-46
A-1-16 跟踪指令 A-50
A-1-17 数据控制指令 A-51
A-1-18 子程序指令 A-54
A-1-19 中断控制指令 A-55
A-1-20 步指令 A-56

A-1-21 基本 I/O 单元指令 A-56
A-1-22 串行通信指令 A-59
A-1-23 网络指令 A-60
A-1-24 文件存储指令 A-63
A-1-25 显示指令 A-65
A-1-26 时钟指令 A-65
A-1-27 调试指令 A-66
A-1-28 故障诊断指令 A-66
A-1-29 其它指令 A-67
A-1-30 块程序指令 A-68
A-1-31 文本字符串处理指令 A-73
A-1-32 任务控制指令 A-76
A-1-33 型号转换指令 A-76
A-1-34 特殊功能块指令 A-77

A-2 指令执行时间和步数 A-78
A-2-1 顺序输入指令 A-79
A-2-2 顺序输出指令 A-79
A-2-3 顺序控制指令 A-80
A-2-4 定时器和计数器指令 A-81
A-2-5 比较指令 A-82
A-2-6 数据传送指令 A-83
A-2-7 数据移位指令 A-83
A-2-8 递增 / 递减指令 A-85
A-2-9 四则运算指令 A-85
A-2-10 转换指令 A-86
A-2-11 逻辑指令 A-89
A-2-12 特殊算术指令 A-89
A-2-13 浮点数运算指令 A-89
A-2-14 双精度浮点数指令 A-91
A-2-15 表格数据处理指令 A-92
A-2-16 跟踪指令 A-94
A-2-17 数据控制指令 A-94
A-2-18 子程序指令 A-95
A-2-19 中断控制指令 A-96
A-2-20 步指令 A-96
A-2-21 基本 I/O 单元指令 A-96
A-2-22 串行通信指令 A-97
A-2-23 网络指令 A-97
A-2-24 文件存储指令 A-98
A-2-25 显示指令 A-98
A-2-26 时钟指令 A-98
A-2-27 调试指令 A-98
A-2-28 故障诊断指令 A-99
A-2-29 其它指令 A-99
A-2-30 块程序指令 A-99
A-2-31 文本字符串处理指令 A-101
A-2-32 任务控制指令 A-102
A-2-33 型号转换指令 A-102
A-2-34 特殊功能块指令 A-102
A-2-35 SFC 指令 A-103
A-2-36 功能块实例执行时间 A-103

A-3 辅助区 A-105
A-3-1 只读区 （通过系统设置） A-105
A-3-2 读 / 写区 （由用户设置） A-124
A-3-3 辅助区操作细节 A-132

A-4 PLC 存储器地址的内存印象图 A-135
A-4-1 PLC 存储器地址 A-135
A-4-2 存储器分配 A-136

A-5 电源中断时操作 A-137
A-5-1 电源掉电操作 A-137
A-5-2 电源掉电的指令执行 A-139

A-6 通过 Windows XP （SP2 及以上）或 Windows Vista 连接 EtherNet/IP A-141
A-6-1 变更 Windows 防火墙设置 A-141
A-7 PLC 比较表：CJ 系列 和 CS 系列 PLC A-143

阅读并理解本手册

请在使用产品前阅读并理解本手册。如您有任何问题或意见，请与您的欧姆龙代表联系。

保证内容和责任限定

保证内容
欧姆龙的产品保证是指产品自售出起一年 （或其它指定期间）内在材料和工艺上没有缺陷。 欧姆龙不对产品的某一指定用途作出保证或表达 （不论是明示或暗示）关于侵权，适销性，或适用性。买方或用户承认是自主决定这些产品将适当地满足其预定用途。欧姆龙放弃所有其他明示或暗示的保证。

责任限定
因本公司产品引起的特别损失、间接损失、及其他相关损失等情况，本公司不承担任何责任。 在任何情况下，欧姆龙声明对超过产品价格的责任免责。 在任何情况下，欧姆龙对与产品相关的保证、维修或其他主张不负责；除非经欧姆龙的分析后，确认该产品被适当的使用、储藏、安装和维护，并且没有受到污染、滥用、误用、不当修改或修理。

应用注意事项

使用的适用性

欧姆龙对于客户在其应用中的产品组合或产品使用的标准、代码或规则不承担责任。

根据客户的要求，欧姆龙将提供相应的第三方认证来明确适用于产品的额定值和使用限制。该信息本身不足以完全决定产品与其他产品、设备、系统及其他应用或组合的适用性。

以下为一些必须引起特别注意的应用场合，且下述内容并非为包括所有可能的产品用途，也不表示所列用途对产品均适用：

- 户外使用、遭受潜在化学污染或电干扰处使用、或未在本手册中提及的条件或用途。
- 核能控制系统、燃烧系统、铁路系统、航空系统、医疗器材、娱乐机械、车辆、安全设备和服从分离工业或政府管控的领域。
- 可能对生命或财产造成风险的系统、机器和设备。

请了解并遵守所有产品可用性的禁止条款。

在没有明确系统整体的危险性的情况下，不得将本公司产品用于与人身财产安全密切相关的场合。欧姆龙的产品适用评估并安装于经过全面考虑的设备或系统。

可编程产品

使用可编程产品时，欧姆龙不对用户的程序或其引起的后果负担任何责任。

不承诺事项

规格的变更

基于产品的改进和其他理由，产品规格及附属品可能发生变更。

当公布的规格、性能改变，或重大的结构改变时，公司将改变型号。但对一些产品的规格发生改变时不作另行通知。基于你的要求，对你的应用可指定特别的型号或设立关键的规格，请及时与各销售网点的人员联系，确认所购产品的实际规格。

尺寸和重量

尺寸和重量仅为名义上的，即使已说明了公差，也不能用于制造用途。

性能数据

本手册所给出的性能数据仅是提供给用户作为确定适用性的参考，并不予以担保。其仅为表示在欧姆龙测试条件下的结果，用户必须将其与实际应用条件相联系。实际性能遵从欧姆龙的保证内容和责任限定。

错误和疏忽

本手册中的信息已仔细核对并认为是准确的；但对于文字、印刷或校对错误和疏忽不承担责任。

安全注意事项

安全注意信息的定义

以下标识用于本手册中，以提供 CJ 系列 PLC 的安全使用所需的注意事项。提供的安全注意事项对于安全使用非常重要。因此，请在仔细阅读并理解相关内容的基础上进行使用。

⚠ 警告	表示一个紧迫的危险情况，如不避免可能导致死亡或严重伤害。此外，还可能导致严重的财产损失。
⚠ 注意	表示一个潜在的危险情况，如不避免可能导致轻度或中度伤害，或财产损失。



安全使用注意事项

关于该做什么与不该做什么的注意事项，以保证产品的安全使用。



正确使用注意事项

关于该做什么与不该做什么的注意事项，以保证正确的操作和运行。

符号



三角符号表示注意事项 （包括警告）。具体内容显示在三角中并通过文本解释。该示例表示的是关于触电的注意事项。



圆圈和斜线符号表示为禁止的操作。具体内容显示在圆圈中并通过文本解释。



填充的圆圈符号表示强制进行的操作。具体内容显示在圆圈中并通过文本解释。该示例表示的是一般的强制事项。



三角符号表示注意事项 （包括警告）。具体内容显示在三角中并通过文本解释。该示例表示的是一般通用注意事项。



三角符号表示注意事项 （包括警告）。具体内容显示在三角中并通过文本解释。该示例表示的是高温注意事项。

⚠ 警告

在供电的同时不要试图拆卸任何单元，否则可能导致触电。



在供电的同时不要触及任一端子或端子台，否则可能导致触电。



为了在因 PLC 误动作或其他影响 PLC 操作的外部因素引起异常时保证系统安全，在外部电路中 （即不在可编程序控制器中）要设有安全措施，否则可能导致严重事故。



- 在外部控制电路中必须设有紧急停止电路、联锁电路、限位电路以及类似的安全措施。
- 在自诊断功能检测任何错误时或在执行严重故障报警 （FALS）指令时，PLC 会将所有输出置 OFF。为了保证系统的安全，必须设有外部安全措施作为这种错误的预防措施。
- 由于输出继电器的卡死、烧坏或输出晶体管的损坏，PLC 输出可能保持在 ON 或 OFF。为了保证系统的安全必须设有外部安全措施，作为这个问题的预防措施。
- 在 DC 24V 输出（PLC 的服务电源）过载或短路时，电压可能下降并使各输出变为 OFF。为了保证系统的安全必须设有外部安全措施，作为这个问题的预防措施。

在使用外部工具将存储在文件存储器 （存储卡或 EM 文件存储器）中的数据文件传送到 CPU 单元的 I/O 区 （CIO）之前要确认安全。否则，与输出单元连接的各设备可能误动作，而不顾 CPU 单元的运行模式。



顾客必须采取自动防故障措施以确保发生由损坏的信号线、瞬间电源中断或其它原因导致的不正确、缺失或异常信号时的安全性。若不采取适当措施则异常操作可能导致严重事故。



⚠ 警告

恢复数据时，在执行恢复操作前，仔细检查所选的数据为需要恢复的正确数据。根据所选数据的内容，控制系统可能在数据恢复后发生意外的操作。



某些特殊 I/O 单元和 CPU 总线单元随保存在 CPU 单元中的参数一起运行（例如， DM 区中的分配字、数据链接表或 Ethernet 设定）。如果对于所选的 CPU 总线单元或特殊 I/O 单元具有任何限制，该限制的信息将显示在 PLC 备份工具的信息区中。检查限制，然后在备份或恢复数据时确认选择了 CPU 单元以及 CPU 总线单元或特殊 I/O 单元。如果没有选择这两个单元而根据备份或恢复的数据启动设备，控制系统可能发生意外的操作。



如果要保存的数据包含具有备份限制的单元，该限制的信息将显示在 PLC 备份工具的信息区中。检查限制信息并采取必要的预防措施。数据恢复后对设备进行操作时，控制系统可能出现意外的操作。



在 PLC 操作期间对数据进行恢复前，需确认 PLC 停止工作不会出现问题。如果 PLC 在意外的时间停止，控制系统可能会出现意外的操作。



恢复数据后，确认关闭 PLC 电源然后再打开。如果电源没有复位，系统可能不会随恢复的数据进行更新，并且控制系统可能发生意外的操作。



强制状态的数据可以备份但不能被恢复。恢复包含强制状态的数据后，在开始操作前根据需要通过 CX-Programmer 执行强制置位 / 强制复位的步骤。根据强制状态的不同，控制系统可能会出现意外的操作。



⚠ 注意

请在确认延长循环时间不会引起不良影响后，再执行在线编辑。
否则，输入信号可能为不可读取。



在传送一程序给其他节点或改变 I/O 存储器区的内容时，请先确认目的节点的安全。若不进行安全确认，则可能导致伤害发生。



当用户程序和参数数据被写入 CPU 单元时， CJ2 CPU 单元自动将其备份至闪存。但是， I/O 存储器 （包括 DM、 EM 和保持区）不写入闪存。

DM、 EM 和保持区在电源中断期间由电池来保留。若发生电池错误，电源中断后这些区中的内容可能不正确。 DM、 EM 和保持区的内容被用于控制外部输出，无论电池错误标志 （A402.04）何时打开都要防止不当输出。



以操作手册中规定的力矩来拧紧 AC 电源单元端子台上的螺丝。螺丝松动可能引起燃烧或误动作。



不要在供电的同时或在电源断开后立即碰触电源。电源单元过热可能会导致烫伤。



当将计算机或其它外围设备连接至安装了非隔离单元 （CS1W-CLK1 □ /5 □ （-V1）或 CS1W-ETN01）的 PLC 时，如果外部电源的 24 V 侧接地并且外部设备的 0V 侧接地，将发生短路。连接外围设备到这种类型的 PLC 时，将外部电源的 0V 侧接地或根本不将外部电源接地。



⚠ 注意



如果将符号或存储器地址（仅 ST 编程允许的符号）指定给梯形图或 ST 编程中的数组变量，需确认被指定的部件编号没有超出最大存储器区范围。例如，写入程序，仅当通过使用符号比较指令或 IF 语句间接指定的符号或地址在存储器区内时，执行处理。

指定的部件编号超出对符号指定的存储器区的最大范围将导致在不同存储器区访问数据，并可能导致意外的操作。

如果将符号或存储器地址指定给梯形图中的偏移值，则当通过字地址或符号间接指定偏移值时，须编程以确保起始地址的存储器区没有超出限制范围。例如，写入程序，通过使用输入比较指令或其他指令，令仅当间接指定未使末尾地址超出存储器区时，执行处理。

如果间接指定使地址超出起始地址区，则将导致在不同存储器区访问数据，并可能导致意外的操作。

应用注意事项

使用 CJ 系列 PLC 系统时要遵循下列各注意事项。

● 电源

- 请使用操作手册中规定的电源电压，不正确的电压可能导致误动作或燃烧。
- 超出电源单元的容量可能导致 CPU 单元或其他单元不能启动。
- 请采取适当措施保证提供具有额定电压和频率的指定电源。请特别注意供电不稳定的地方，不正确的电源可能导致误动作。
- 在着手做下列中的任一项前，总是将 PLC 的电源置于 OFF。否则可能引起误动作或触电。
  - 安装或拆卸电源单元、 I/O 单元、 CPU 单元或其他任何单元；
  - 装配各单元；
  - 设定 DIP 开关或旋转开关；
  - 电缆连接或系统接线；
  - 连接或断开连接器。
- 当在端子间接交叉线时，两个端子的总电流将在线路中流动。确认接线的电流容量充足。
- 使用带有更换通知功能的电源单元时，遵守下列注意事项。
  - 当电源单元前端的显示器开始交替显示 “0.0”和 “A02”或报警输出自动关闭时，请在 6 个月内更换电源单元。
  - 从电线和高压线上分离报警输出线。
  - 不要对报警输出施加超过额定电压的电压和连接超过额定负载的负载。
  - 当保存产品超过3个月时维持-20～30℃的环境温度和25%～70%的湿度以保持更换通知功能。
  - 不正确的安装将降低散热性，延迟更换通知信号，并可能使内部元件老化或损坏。总是使用标准安装方法。
- 不要在关闭电源后立即触摸电源单元上的端子。由于残余电压可能引发触电。
- 连接计算机到 PLC 时请注意下述注意事项，以防止因电位差导致故障的发生。
  - 连接计算机与PLC时，请先将计算机电源插头从AC供电口拔下。由于AC适配器中的漏电流，也可能导致计算机与 PLC 之间的电位差。连接计算机与 PLC 后，再次从 AC 适配器进行供电。
  - 如果计算机带有 FG 端子时，请连接 PLC 的 FG （GR）端子以保持相同的电位。
- 如果将计算机接地到其他位置，则将根据接地条件而异可能发生电位差。

● 安装

- 请不要将 PLC 安装在高频率干扰源附近。
- 在接触单元前，为使所有聚积的静电放电，务必先接触接地金属物。否则可能导致误动作或伤害。

- 务必使连接器、端子台、扩展电缆和其他项目与锁扣装置正确地锁合。不正确的锁合可能导致误动作。
- 将电源单元、CPU 单元、I/O 单元、特殊 I/O 单元或 CPU 总线单元连接在一起后，滑动单元顶部和底部的滑卡 （直至其就位）。若滑卡没有锁定到位，单元可能无法正确动作。若滑卡没有锁定，可能无法实现正确的功能。
- 确保将 CPU 单元所附带的端板安装于 CPU 机架的右端。若未盖上端板， PLC 将误动作。

● 接线

- 请正确连接所有连接器。
- 在通电前，请对所有接线进行双重检查。不正确的接线可能导致燃烧。
- 确保以相关手册中指定的紧固力矩拧紧所有端子台螺丝和电缆连接器螺丝。不正确的拧紧力矩可能导致故障。
- 紧固端子时不要在端子台上施加大于 100 N 的力。
- 安装单元应在完全地检查端子台和连接器后进行。
- 接线时让标签附于单元上。若移除该标签可能因异物进入单元而导致故障。
- 为了保证散热，在接线完成后请将标签除去。保留所贴的标签可能导致误动作。
- 请使用压接端子进行接线。不要将裸绞合线直接与端子连接。否则可能导致燃烧。
- 切勿将高于额定输入电压的电压施加于输入单元，过电压可能导致燃烧。
- 在安装单元时总是连接到接地电阻不大于 100Ω 的接地上。没有连接到不大于 100Ω 的接地电阻的接地可能导致触电。
- 在短接电源单元的 GR 和 LG 端子时，必须有一个不大于 100Ω 接地电阻的接地。
- 为了在信号线断开、瞬时电源中断或其他原因引起的不正确、丢失或异常信号事件时保证安全，用户必须采取故障安全措施。
- 切勿将超出最大开关能力的电压或负载施加到输出单元。否则将可能因过电压或过载导致燃烧。
- 不要拽拉电缆或弯折电缆超过其应有限度。做其中的任一种动作都可能导致电缆断开。
- 不要在电缆或其他接线上放置物品。放置物品可能导致电缆断裂。
- 不要使用市售 RS-232C 个人计算机电缆。请务必使用本手册中所列指定电缆或根据手册规格制作电缆。使用市售电缆可能损坏外部设备或 CPU 单元。
- 不要将CPU单元上RS-232C 端口的针脚6（+5 V 电源线）连接至除了CJ1WCIF11适配器或NT-AL001适配器之外的任何设备上。否则可能损坏外部设备或 CPU 单元。

● 处理

- 若施加了全部绝缘强度测试电压或使用开关突然关闭电源，可能损坏电源单元，可使用可调节电阻器逐渐增加或减少电压。
- 执行耐电压测试或绝缘电阻测试前，请先从电源单元上将线路接地端子 （LG）与功能接地端子（GR）分离。否则将可能导致起火。
- 在开始操作前确保 DIP 开关和 DM 区设置正确。

- 确保恢复操作前将所需的 DM 区、保持区和其他存储器区的数据传送到 CPU 单元。
- 在着手进行下列的任一项之前，请确认在系统中不会发生不利影响。否则可能导致不可预料的动作。
  - 改变 PLC 的运行模式 （包括启动运行模式的设置）；
  - 强制置位 / 强制复位存储器中的任一位；
  - 改变存储器中任一字或任一设置值的当前值。
- 不要试图拆卸、修理或改装任何单元。任何这类做法或企图都可能导致误动作、火灾或触电。
- 勿使 PLC 坠落或使其遭受过度振动或冲击。
- 在调换零部件时，务必确认新零部件的额定值是否正确。否则可能导致误动作或燃烧。
- 在运输或存储单元或电路板时，静电会破坏LSI和IC，为防止其受静电影响请保持正常储存温度，并应用抗静电材料将 PCB 包上。
- 不要裸手接触各电路板和安装在其上的各部件。各电路板上有尖的引线和其它零部件，如果处理不当可能引起伤害。
- 如果 CPU 单元在没有安装电池的情况下放置一段时间且没有接通电源，然后安装电池但不接通电源，电池的寿命将会缩短。
- 更换电池前，在开始进行更换步骤前将电源打开至少5分钟，并在电源关闭后的5分钟内完成电池的更换。如果没有遵守该注意事项，存储器的内容可能被损坏。
- 如果在编程中应用电池错误标志，则在您更换电池前，应确保当电源接通时系统检测到电池错误情况下的系统安全。
- 不要短接电池端子或将电池充电、解体、加热或焚烧。不要使电池受到强烈冲击。诸如此类的行动都可能导致电池漏电、绝缘击穿、发热或爆火。请将掉落在地上或另外受到过度冲击的电池去除。使用受过冲击的电池可能导致漏电。
- UL 标准要求电池只由有经验的技术人员更换。确认有经验的技术人员进行电池更换。根据本手册中提供的以下步骤更换电池。



廢電池請回收

- 若设置了不当数据链接表或参数则可能导致意外操作。即使设置了适当的数据表和参数，也要确认受控系统在开始或停止数据链接时将不受负面影响。
- 当路由器表从编程设备传送至 CPU 单元时 CPU 总线单元将重启。读取和允许新的路由器表需要重启这些单元。确保系统在允许 CPU 总线单元重置前不受负面影响。
  - 在从编程设备（手持编程器或 CX-Programmer）生成一个 AUTOEXEC.IOM 文件以在启动期间自动传送数据时，将第一个写地址设置在 D20000，并确保所写数据的大小不超过 DM 区的大小。当在启动时从存储器卡读数据文件时，数据会被写入 CPU 单元，（从 D20000 开始，即使在生成 AUTOEXEC.IOM 文件时设置了另外地址）。此外，如果超过 DM 区 （使用 CX-Programmer 时是可能的），则剩余数据会被写到 EM 区。
- CJ 系列 CPU 单元出厂时已安装了电池，且内部时钟已设置了时间。在使用 CJ 系列 CPU 单元前不必再清除存储器或设置时钟。这点与 CS 系列 CS1 CPU 单元不同。

- CJ2 CPU 单元内的用户程序和参数区数据，备份在内置闪存储器内。执行备份操作中时，CPU 单元前面的 BKUP 指示灯会点亮。当 BKUP 指示灯点亮时，不要断开 CPU 单元的电源，否则数据将无法进行备份。
- 用户程序在单元上实际运行前，为了正确执行要对其和单元进行检查。没有检查程序和参数设置可能导致不可预料的操作。
- 在 I/O 表中设定特殊 I/O 单元或 CPU 总线单元时，在重启单元前，需仔细检查连接目标的设备安全。
- CS1 或 CJ1 CPU 单元的程序文件（.OBJ）不能通过存储卡被直接传送到 CJ2 CPU 单元。必须先通过 CX-Programmer 将其转换为 CJ2 CPU 单元可用的文件。

● 外部电路

- 用户必须在外部电路中（即，不在可编程序控制器内）设有联锁电路、限位电路和类似的安全措施。
- 请务必在接通 PLC 电源后再接通控制系统电源。若先接通控制系统电源后再接通 PLC 电源，则将可能因在接通 PLC 电源时 DC 输出单元上输出单元和其他单元会瞬时变为 ON，而导致控制系统信号中的临时错误。
- 用户必须采取必要的措施，以确保在发生因内部电路故障（如，发生在继电器、晶体管或其他元器件内）而导致输出单元的输出保持为 ON 时的系统安全。
- 如果 I/O 保持位变为 ON，则在 PLC 从 RUN 或 MONITOR 模式切换为 PROGRAM 模式时，PLC 的输出不会变为 OFF，而会保持其先前状态，确保在这个发生时外部负载不会引起危险情况发生。（当操作因致命错误停止时，包括是由 FALS （007）指令产生的，输出单元的所有输出都会变为 OFF，而只是内部输出状态会被保持）。
- 请安装外部断路器和采取其他安全措施，防止外部接线短路。防短路安全措施不充分可能导致燃烧。
- 请根据各地的规定，对本产品及电池进行废弃处理。
- 在读或写存储卡时不要断开 PLC 电源。此外，在 BUSY 指示灯点亮时，不要取出存储卡。否则可能造成存储卡不可用。
- 要取出存储卡，先按存储卡电源开关，等待 BUSY 指示灯熄灭，然后取出存储卡。

运行环境注意事项

● 按照本手册中的指示正确执行安装。

● 请勿在下列场所操作控制系统：

- 阳光直射出处；
- 温度或湿度超过规格中规定范围的场所；
- 由于温度急剧变化容易引起凝露的场所；
- 有腐蚀性气体或易燃性气体的场所；
- 有尘埃（特别是铁屑）或盐雾的场所；
- 暴露于水、油、或化学品的场所；
- 易受直接冲击或振动的场所。

● 在将系统安装在下列场所时需要采取适当而充分的预防措施：

- 有静电或其他形式噪声的场所；
- 有强电磁场的场所；
- 可能暴露于放射性的场所；
- 靠近动力电源的场所。

规定及标准

符合 EC 指令

适用规程

- EMC 规程
- 低电压规程

概念

● EMC 规程

欧姆龙公司的所有装置都符合 EC 规程，也符合有关 EMC 标准，所以它们可以很方便地装入其他装置或所有机械。为了符合 EMC 标准，对各实际产品都作了检验（参见下注）。然而各产品是否符合用户所用的系统，必须由用户来检验。

符合 EC 规程的欧姆龙装置的 EMC 相关性能，随装有欧姆龙装置的设备的配置、接线、和其他条件或控制柜的不同而不同。

因此，为了确认各装置和整个机械符合 EMC 标准，用户必须执行最后检查。

* 适用的 EMC （电磁兼容）标准如下：
EMS （电磁敏感度）：
CS 系列：EN 61131-2 和 EN 61000-6-2
CJ 系列：EN 61000-6-2

* EMI （电磁干扰）：
EN 61000-6-4 （辐射发射：10 m 调整率）

● 低电压规程

始终保证装置工作在 AC 50 ～ 1,000 V 的电压和 DC 75 ～ 1,500 V 的电压范围中，满足 PLC 所要求的安全标准 (EN61131-2)。

● 符合 EC 规程

CJ 系列PLC 符合EC 规程。为了保证使用CJ 系列PLC 的机械或装置符合EC规程，PLC 必须安装如下：

- CJ 系列 PLC 必须安装在控制柜内。
- DC 电源单元和 I/O 单元用的 DC 电源必须采用加强绝缘或双重绝缘。
- CJ 系列 PLC 符合 EC 规程，也符合发射标准（EN61000-6-4）。辐射发射特性（10m 调整率）可能随所用的控制柜的配置、与控制柜连接的其他设备、接线和其它条件的不同而不同。因此，用户必须确认整个机械或设备符合 EC 规程。

适合船舶标准

本产品适用各种造船标准。所适用各船舶标准根据特定使用条件而异，可能在部分场合无法使用本产品。因此，请在将本产品用于船舶前，先向您的欧姆龙代表询问。

各造船标准的使用条件 （NK、 LR）

● 船桥 （Bridge）及甲板 （Deck）以外的使用条件

- PLC 必须安装在控制柜内。
- 控制柜的开闭口等间隙须使用垫衬或其他材料进行完全填充或覆盖。

● 船桥 （Bridge）及甲板 （Deck）的使用条件 （仅由 NK 认证）

- PLC 必须安装在控制柜内。
- 控制柜的开闭口等间隙须使用垫衬或其他材料进行完全填充或覆盖。
- 必须将下述噪声滤波器连接到电源线。

抗噪声滤波器

制造商	Cosel Co., Ltd.
型号	TAH-06-683

商标

SYSMAC 为欧姆龙公司生产的可编程控制器的注册商标。

CX-One 为欧姆龙公司生产的编程软件的注册商标。

Windows 是美国微软公司的注册商标。

本文中记载的其他系统名和产品名都分别为各家公司的商标或注册商标。

相关手册

CJ 系列 CJ2 CPU 单元 PLC 本体的相关手册，为如下表所示，请进行参阅。

手册	样本编号	型号	应用	内容
CJ 系列 CJ2 CPU 单元软件操作手册（本手册）	W473	CJ2H-CPU □□ -EIP	CJ2 CPU 单元的软件规格	描述了 CJ2 CPU 单元的以下内容： • CPU 单元操作 • 内部存储器 • 编程 • 设定 • 内置到 CPU 单元的功能 参考了硬件操作手册 （W472）。
CJ 系列 CJ2 CPU 单元硬件操作手册	W472	CJ2H-CPU □□ -EIP	CJ2 CPU 单元的硬件规格	描述了 CJ2 CPU 单元的以下内容： • 概述和特性 • 基本系统配置 • 各部件名称和功能 • 安装和设置步骤 • 错误的纠正措施 参考了软件操作手册 （W473）
EtherNet/IP 单元操作手册	W465	CJ2H-CPU □□ -EIP CS1W-EIP21 CJ1W-EIP21	CJ2 CPU 单元的内置 EtherNet/IP 端口的使用	描述了内置 EtherNet/IP 端口和 EtherNet/IP 单元。 描述了基本设置、标签数据链接、 FINS 通信以及其他功能。
CS/CJ/NSJ 系列指令参考手册	W474	CJ2H-CPU □□ -EIP CS1G/H-CPU □□ H CS1G/H-CPU □□ -V1 CJ1G/H-CPU □□ H CJ1G-CPU □□ CJ1M-CPU □□ NSJ □ - □□□□ (B)-G5D NSJ □ - □□□□ (B)-M3D	指令信息	描述了每个编程指令的详细信息。 编程时也可参考软件操作手册 （W473）。
CS/CJ/CP/NSJ 系列通信命令参考手册	W342	CS1G/H-CPU □□ H CS1G/H-CPU □□ -V1 CS1D-CPU □□ H CS1D-CPU □□ S CS1W-SCU □□ -V1 CS1W-SCB □□ -V1 CJ1H-CPU □□ H-R CJ1G/H-CPU □□ H CJ1G-CPU □□ P CJ1M-CPU □□ CJ1G-CPU □□ CJ1W-SCU □□ -V1 CP1H-X □□□□ - □ CP1H-XA □□□□ - □ CP1H-Y □□□□ - □ NSJ □ - □□□□ (B)-G5D NSJ □ - □□□□ (B)-M3D	CS/CJ/CP 系列 CPU 单元和 NSJ 系列控制器的通信信息	描述了 C 模式命令和 FINS 命令。 关于使用 C 模式命令或 FINS 命令与 CPU 单元通信的命令的详细描述，请参考该手册。 注　本手册描述了用于寻址 CPU 单元的通信指令。与所使用通信路径无关 （可经由 CPU 单元上的串行端口，串行通信单元 / 板上的通信端口，以及通信单元等）。关于寻址特殊 I/O 单元或 CPU 总线单元的通信指令，请参见相关单元的操作手册。
CX-One 设置手册	W463	CXONE-AL □□ C-V3/AL □□ D-V3	CX-One 的软件安装	提供了 CX-One FA 综合工具包的概述及安装步骤描述。
CX-Programmer 操作手册	W446	WS02-CX □□ -V □	Windows 计算机的支持软件 CX-Programmer 操作步骤	描述了 CX-Programmer 的操作步骤。 编程时也可参考软件操作手册 （W473）和 CS/CJ/NSJ 系列指令参考手册 （W474）。
CX-Programmer 操作手册功能块	W447			
CX-Programmer 操作手册 SFC 编程	W469			

手册	样本编号	型号	应用	内容
CS/CJ/CP/NSJ 系列 CX-Simulator 操作手册	W366	WS02-SIMC1-E	Windows 计算机中使用。CX-Simulator 的操作步骤及 支持软件CX-Programmer 6.1 及以上版本中，使用仿真功能。	描述了 CX-Simulator 的操作步骤。 进行仿真时，也可参考 CX-Programmer 操作手册（W446）、软件操作手册 （W473）和 CS/CJ/NSJ 系列指令参考手册 （W474）。
CS/CJ/CP/NSJ 系列 CX-Integrator 网络配置软件操作手册	W464	CXONE-AL □□ C-V3/ CXONE-AL □□ D-V3	网络设置和监控	描述了 CX-Integrator 的操作步骤。

1

概述

本章节为 CJ2 CPU 单元的概述。

1-1 CJ2 CPU 单元概述 1-2
1-1-1 概述 1-2
1-1-2 CJ2 CPU 单元特长 1-3
1-2 基本操作步骤 1-7

1-1 CJ2 CPU 单元概述

1-1-1 概述

SYSMAC CJ2 CPU 单元为具有大存储容量的高速高功能型 CPU 单元。相关性能如下所述。

● 高速、大容量存储器

基本性能高速化，并且提升了存储器容量。以实现更从容的机械控制。

● 内置 EtherNet/IP 端口

CJ2 CPU 将 EtherNet/IP 开放式网络作为标准配置。可通过支持软件实现 PLC 间数据链接、 PLC 间报文通信、 FTP 传送等通用 Ethernet 通信。

● 用于支持软件接口的一般用途网络

可使用市售的电缆，将支持软件及设备经由 USB 端口 /EtherNet/IP 端口连接到一般用途网络。

● 标签访问

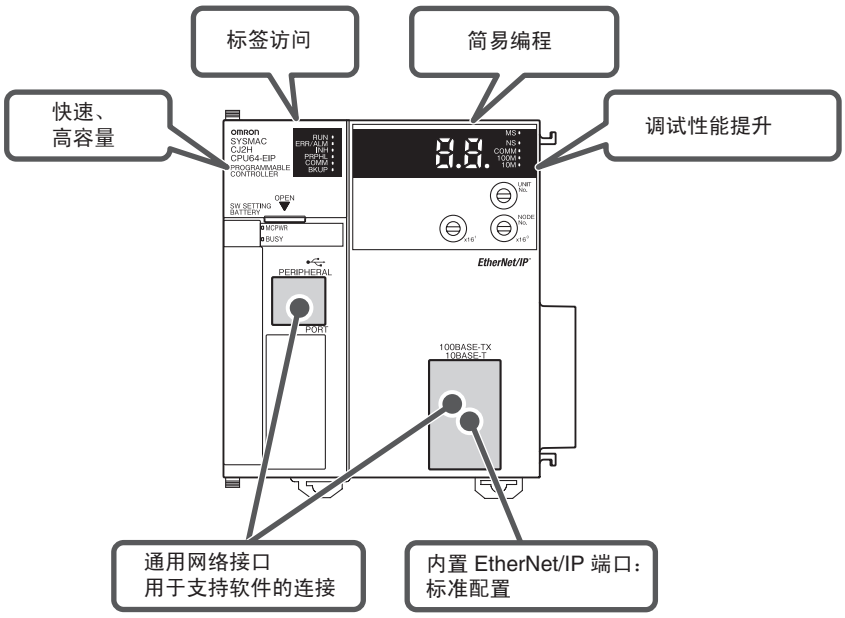
CJ2 CPU 单元具有标签访问功能，可对标签名称、 I/O 地址等进行管理。可通过使用标签名称实现从外部设备进行访问，不再需要使用 I/O 地址。

● 编程更简单

CJ2 CPU 可提供具有高可读性的编程环境，包括 DM/EM 区的位指定、地址偏移指定、数组变量等。

● 强化调试功能

提升了在线编辑和数据追踪功能。大幅度提升了调试效率。



1-1-2 CJ2 CPU 单元特长

高速、大容量存储器

● 高速处理

实现基本指令 0.016 μs、特殊指令 0.048 μs、即时刷新指令 0.99 μs 的高速处理。

● 大程序容量

CJ2 CPU 单元可提供最大 400K 步的大容量编程。

● 大容量数据存储器

CJ2 CPU 单元可提供最大 800 K 字 （25 组）的大容量 EM （扩展数据存储器）区。

● 最大 128 个循环任务数

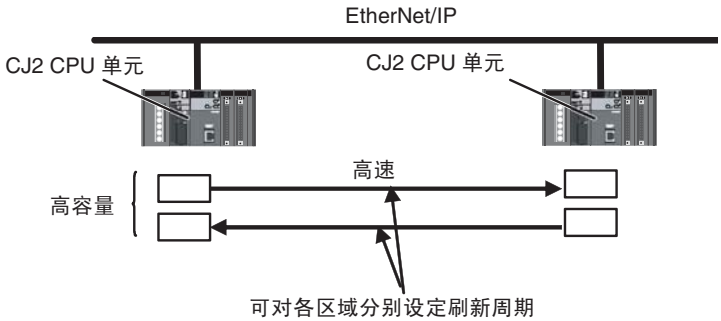
最大可将用户程序分割成 128 个任务。通过此操作，可实现在更小单位下的编程，令操作更容易，并可缩短循环时间。

内置 EtherNet/IP 端口

CJ2 CPU 标准配置内置 EtherNet/IP 端口，并支持与 Ethernet 单元（包括诸如 FTP 服务器、时间调整等特点）及开放式网络 EtherNet/IP 相同的 FINS Ethernet 功能。

● 高速、大容量，并可由用户自设定刷新周期的数据链接

CJ2 CPU 单元可在高速、大容量特性下，对各数据链接区域分别进行刷新周期设定。



此外，使用 EtherNet/IP 数据链接工具时，可在与 Controller Link 数据链接自动设定相类同的界面下对数据链接进行设定。

用于支持软件接口的一般用途网络

● 内置 USB 和 EtherNet/IP 端口

可通过市售的 USB 电缆直接连接计算机的 USB 端口。此外，还可经 USB 对 EtherNet/IP 网络上的 PLC 进行访问。

● 支持 PLC 命名功能避免了支持软件的错误连接

CJ2 CPU 单元可记忆存储用户自定义的 PLC 名称。当支持软件连接到正在工作的 PLC 时，可通过校验 PLC 名称避免支持软件的错误连接。

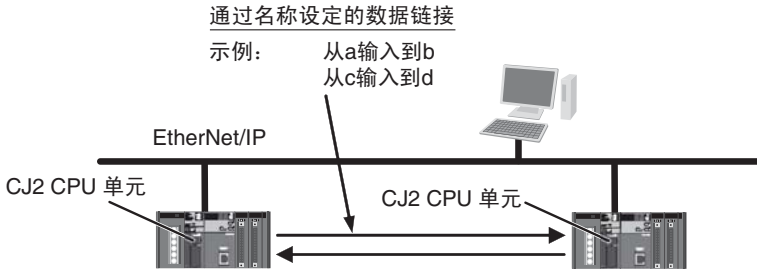
标签访问

● 通过网络变量 （标签），实现灵活的程序变更

CJ2 CPU 单元支持网络变量 （标签）。通过使用 CPU 单元内的标签名称服务功能，并将各标签及地址保存在 CPU 单元的变量表内，由此实现如下操作。

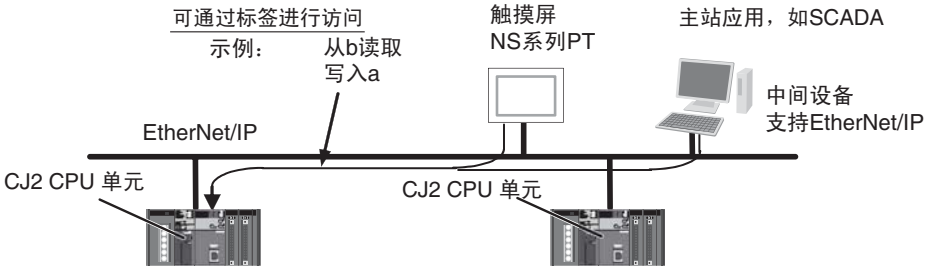
通过标签名称进行数据链接设定

EtherNet/IP 上的数据链接中，可不使用地址而仅通过变量名称对收发信区域进行指定。据此，可在保持以标签名称设定的数据链接区域设定不变的状态下，仅将用作参照表的变量表进行变更即可，从而可实现更灵活的设计变更。



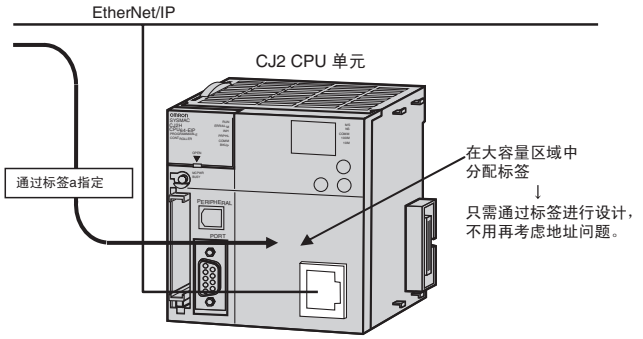
通过标签名称从上位设备进行访问

可通过标签名称从上位设备 （如触摸屏 NS 系列可编程终端等）进行访问。由此，使得 PT 设备等的主站显示及程序创建更为方便。



通过标签的自动分配功能，无需再考虑地址问题

通过 CX-Programmer 变量表的地址自动分配功能，可对大容量的 EM 区中的标签进行自动分配，因而可在不用再考虑地址问题的状态下，实现数据链接设计及上位设备的访问。



编程更简单

● 地址偏移指定

指令操作数中进行地址指定时，可将已指定的地址作为起始位置，在 [ ] 中指定偏移位置。例如，通过在 [ ] 中设定字地址以指定偏移位置，则可根据此字内容，对地址进行动态指定。

示例
W0.00[D0]：W0.00 为起始地址，D0 值为偏移内容。若 D0 为 &3，则为指定 W0.03。

● 数组变量的索引变量指定

通过对数组变量下标指定符号 （变量），可以根据符号 （变量）的值动态地指定 （数组的）项。

示例
a[b]：通过变量 b 的值来指定数组变量 a[ ] 的项。

● 在 DM/EM 区中使用位地址

可将以往仅可用于字地址访问的数据存储器区域（DM、 EM），扩展至可用位地址的访问。由此，扩展了位地址的工作区域。

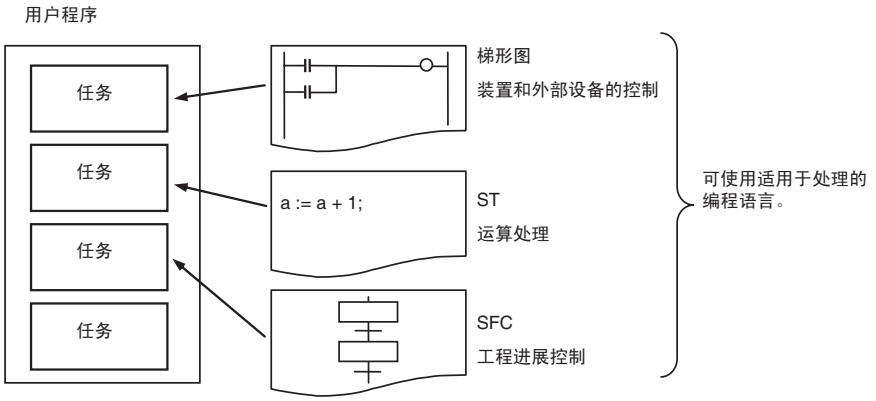
例，D10.00 为指定 D10 的位 00。

● 各指令的定时器 / 计数器当前值更新方式的选择

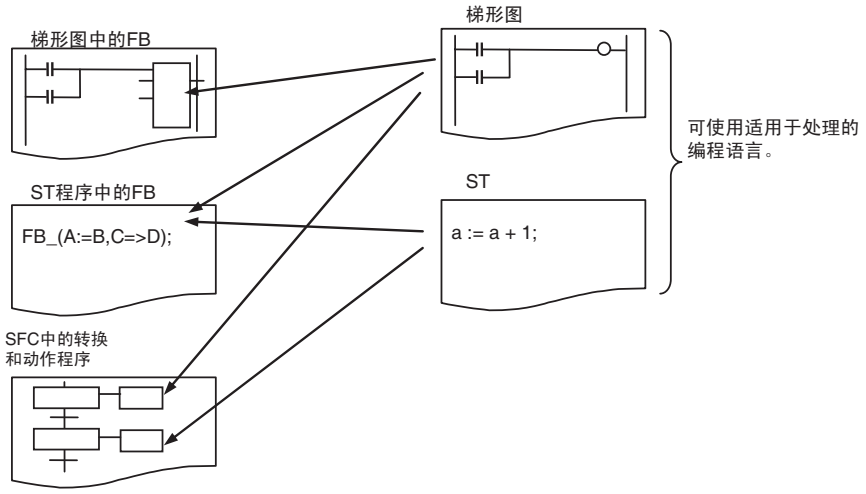
可根据各指令分别对定时器 / 计数器的当前值更新方式 （BCD 方式 /BIN 方式）进行选择。例如，可同时使用 TIM （BCD）和 TIMX （BIN）指令。

● 编程中的最适语言组合

可在用户程序中自由组合梯形图语言、 ST 语言和 SFC 语言。通过对应特定处理内容使用最适合的语言，实现减少编程的时间。



此外，还可在梯形图 /ST 编程的功能块 （FB）中、 SFC 的转换以及动作程序中自由使用梯形图语言和 ST 语言。由此，可实现最合适的编程结构。



提升调试功能

● 降低在线编辑的循环时间影响

将在线编辑执行时额外的循环时间缩短至约 1 ms，最大程度减少了调试期间对设备的影响。

● 快速、高容量的数据追踪

可追踪多达 32K 字的数据 （比原有多 8 倍），具有丰富的触发条件可对数据进行长时间的持续追踪。

1-2 基本操作步骤

通常进行以下的步骤。

1. 设定设备和硬件

安装电源单元、CPU单元、其他单元和端盖。
设定所需的DIP开关和旋转开关。

参考CJ2 CPU单元硬件操作手册（样本编号W472）的第3章各部分名称和功能以及第5章安装。

2. 配线

参考以下手册对电源、I/O和通信进行配线。

参考CJ2 CPU单元硬件操作手册（样本编号W472）的第5章安装。
参考CJ2 CPU单元软件操作手册（样本编号W473）的第11章编程设备和通信。

3. 在线连接到PLC

将计算机在线连接到PLC。

参考CJ2 CPU单元硬件操作手册（样本编号W472）的第4章支持软件。

4. I/O分配

使用CX-Programmer，将CPU单元中的I/O存储器分配至已安装的单元。

参考CJ2 CPU单元软件操作手册（样本编号W473）的第8章分配和单元设定。

5. 软件设置

对PLC进行软件设置。

- CPU单元初始化：参考CJ2 CPU单元软件操作手册（样本编号W473）的第4章CPU单元初始化。
- 对于特殊I/O单元和CPU总线单元的DM区域：参考CJ2 CPU单元软件操作手册（样本编号W473）的第8章I/O分配和单元设定。
- PLC设置：参考CJ2 CPU单元软件操作手册（样本编号W473）的第9章PLC设置。

6. 创建程序

使用CX-Programmer创建程序。

参考CJ2 CPU单元软件操作手册（样本编号W473）的第5章理解编程。

7. 测试操作

测试I/O写入、AR区域设定和执行试操作。CX-Programmer可用于监控和调试。

参考CJ2 CPU单元软件操作手册（样本编号W473）的10-7 调试。

8. 基本程序操作

设定运行模式为RUN模式开始操作。

2

CPU 单元内部存储器

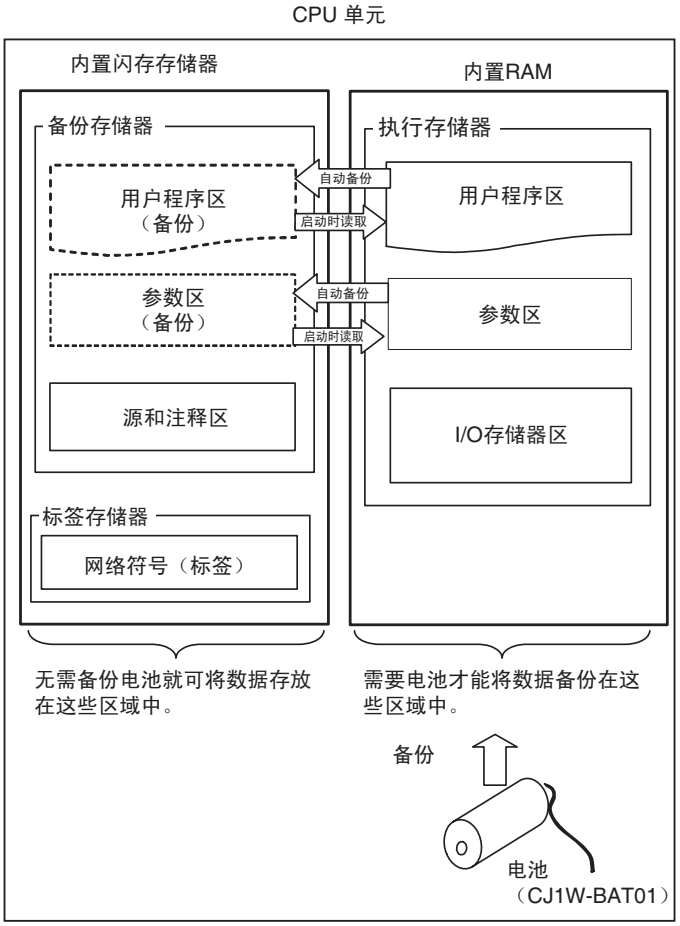
本章描述了 CPU 单元中的存储器区。

2-1 概述 2-2
2-1-1 存储器配置 2-2
2-1-2 存储器区和保存的数据 2-3
2-1-3 从编程设备向 CPU 单元传送数据 2-4

2-1 概述

2-1-1 存储器配置

如下图所示，CPU 单元的内部存储器由内置闪存存储器和内置 RAM 组成。内置 RAM 作为执行存储器使用，内置闪存存储器作为备份存储器使用。



内置闪存存储器

内置闪存存储器备份用户程序和参数，并保存源代码、注释数据和网络符号 （变量）。无需电池即可将数据保存在该区域中。 I/O 存储器不会被备份。

内置 RAM

内置 RAM 为 CPU 单元的执行存储器。内置 RAM 中保存用户程序、参数和 I/O 存储器，并且需要电池进行备份。

如果电池停止工作（例如，电池电压过低或者没有安装电池时），I/O 存储器的数据将丢失。用户程序和参数备份到内置闪存存储器，因此不会丢失。

正确使用注意事项

电池电压过低或没有安装电池时，将发生以下情况。

- 电源OFF时，由电池保持的I/O存储器区的数据包括DM、EM、HR区中的值将会丢失或变得不稳定。
- 时钟将停止，并且所有与时钟相关的数据将变得不稳定。
- 错误记录将不会被保存。
- 输出 OFF 位将变得不稳定。

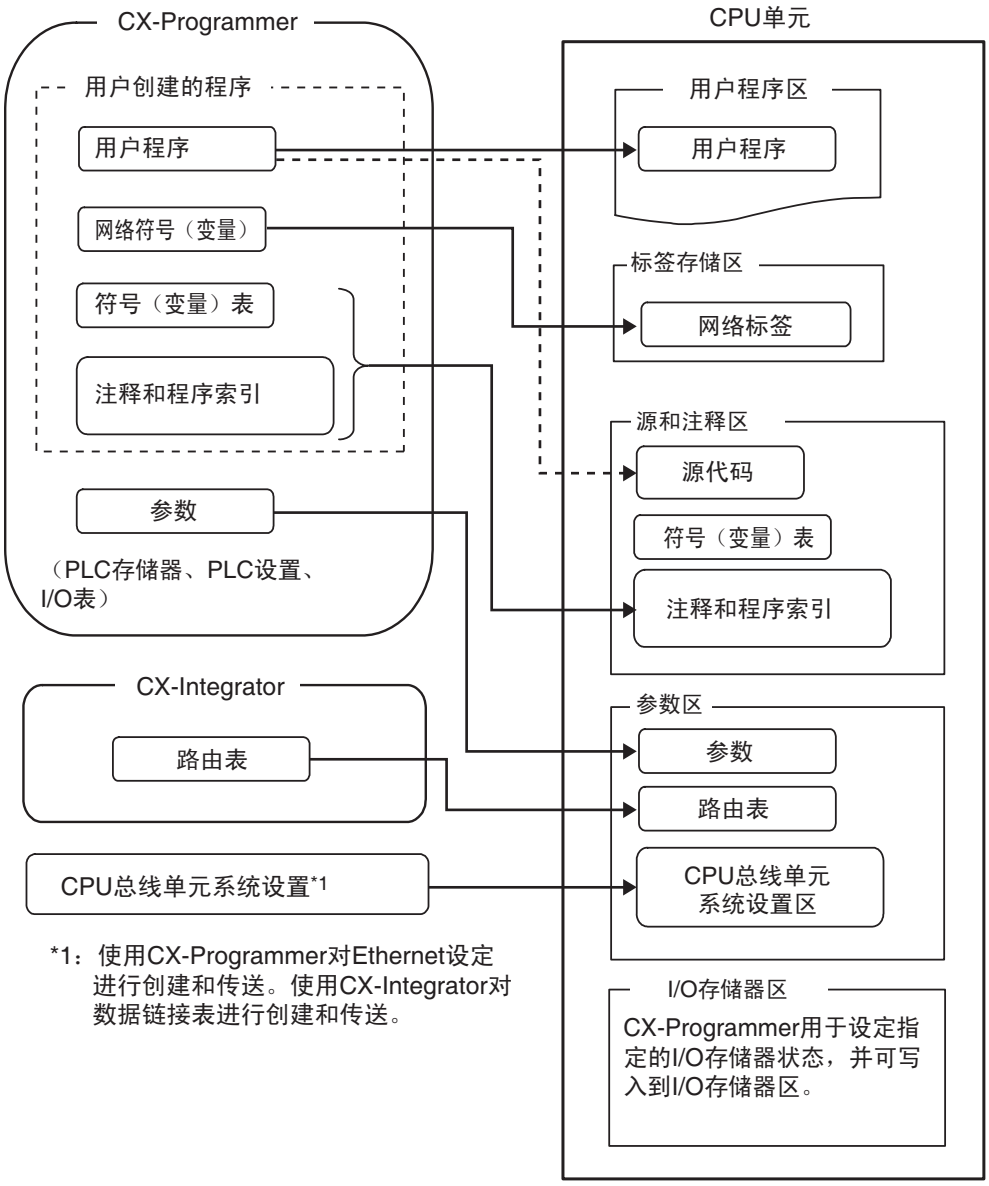
2-1-2 存储器区和保存的数据

下表列出了 CPU 单元存储器区和每个区中保存的数据。

存储器区和保存的数据	描述	内置闪存存储器	内置 RAM
用户程序区	用户程序区保存目标代码用于执行由 CX-Programmer 创建的用户程序。	保存	保存
参数区	参数区保存 PLC 的初始设定。	保存	保存
PLC 名称	CPU 单元的名称由 CX-Programmer 保存、读取和检验，以防止 CX-Programmer 连接至错误的 PLC。		
PLC 设置	在 PLC 设置中使用软件开关进行各种初始设定。参考第 9 章 PLC 设置。		
I/O 表	I/O 表提供用户指定单元的安装状态的信息。参考第 8 章 I/O 分配与单元设定。		
路由表	路由表由用于 FIN 通信的网络参数组成。通过 CX-Integrator 进行指定。参考 CX-Integrator 操作手册 （样本编号 W464）。		
CPU 总线单元设置	CPU 总线单元设置保存指定 CPU 总线单元的初始设定。包括的设定有 Ethernet 单元的 Ethernet 设定、 Controller Link 单元的数据链接参数等。		
I/O 存储器区	I/O 存储器区用于从用户程序中读取和写入数据。根据使用目的分配到下面的区域中。 • CPU 单元的电源复位时清除数据的区域以及保存数据的区域。 • 与其他单元交换数据时的区域以及在内部使用时的区域。	---	保存
源和注释区	源和注释区用于保存程序源代码和使用 CX-Programmer 创建的注释。	保存	---
源代码	程序的源代码 （在任务和功能块中，使用梯形图、 ST、 SFC 语言）。		
符号 （变量）表	符号表包含通过 CX-Programmer 创建的符号 （符号 （变量）名称、地址、 I/O 注释）。		
注释	通过 CX-Programmer 创建的注释，包含标注和行注释。		
程序索引	程序索引提供由 CX-Programmer 创建的程序段的信息以及程序注释。		
网络符号 （变量）（标签）	网络符号的数据在全局符号 （变量）表中。	保存	---

2-1-3 从编程设备向 CPU 单元传送数据

如下图所示，使用 CX-Programmer 或 CX-Integrator 创建的数据被传送到 CPU 单元的内部存储器中。



3

CPU 单元运行

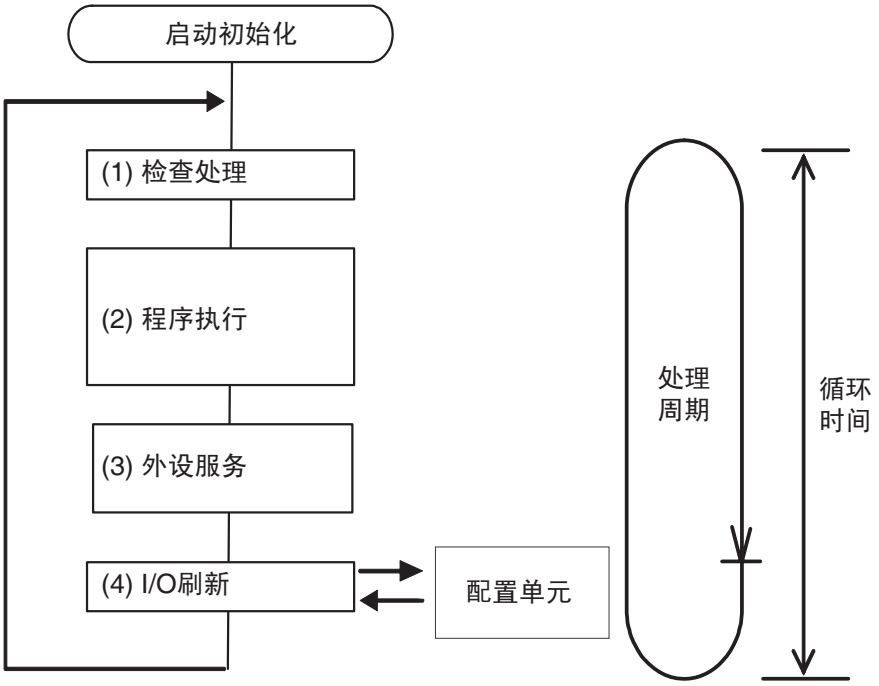
本章描述了 CPU 单元的内部运行以及可对 CPU 单元设定的运行模式。

3-1 CPU 单元内部运行 3-2
3-1-1 概述 3-2
3-1-2 循环时间 3-3
3-1-3 电源中断时的处理 3-5
3-2 CPU 单元运行模式 3-7
3-2-1 运行模式 3-7
3-2-2 运行模式的检查 3-7
3-2-3 运行模式的变更 3-8
3-2-4 运行模式的详细信息 3-12

3-1 CPU 单元内部运行

3-1-1 概述

CPU 单元循环执行用户程序时，将数据写入到内部 I/O 存储器区。刷新 I/O 和进行外设服务时，在外部交换数据。下图显示的是 CPU 单元的内部运行。



启动初始化

每次 PLC 电源变为 ON 时将执行以下的处理。

- 检测已连接的单元 （I/O 分配）
- 比较 I/O 注册表与已连接的单元
- 按 IOM 保持位的状态清除 I/O 存储器的非保持区
- 按强制状态保持位的状态清除强制状态
- 如果安装了存储卡，则引导使用存储卡中自动传送的文件
- 执行自诊断 （检查用户存储器）
- 恢复用户程序
- 更新 PLC 设置

处理循环

启动初始化后，CPU 单元将重复执行四个处理步骤（检查、程序执行、I/O 刷新和外设服务）。完成一个循环所需的时间称为循环时间。

- 检查
  该步骤是执行用户程序的准备。通过其对处理所需的硬件和软件进行检查。检查步骤所需的时间称为头时间。

- 程序执行
  该步骤是指在用户程序中执行指令。当运行模式被设为 RUN 或 MONITOR 模式时，用户程序中的指令被执行。在 CPU 单元 I/O 存储器区中指令的 I/O 处理进程是以位和字执行。
- 外设服务
  当编程设备或单元需求信息时，进行该步骤。数据从 CX-Programmer 写入，或执行网络通信单元的指令。
- I/O 刷新
  该步骤通过 I/O 存储器区与单元交换数据。该处理必须在程序执行后执行，因此当前循环内执行的 I/O 刷新不会被中断 （即不会产生时间碎片）。

3-1-2 循环时间

循环时间为检查、程序执行、外设服务和 I/O 刷新的总时间。每个循环的循环时间通常会根据所执行的处理发生改变。每个处理在循环时间内执行。关于如何计算循环时间请参考第 12 章 CPU 单元循环时间。



附加信息

可通过在 PLC 设置中设定最小循环时间来创建固定循环时间。最小循环时间必须大于正常循环时间。关于如何设定最小循环时间，请参考 10-2-1 最小循环时间。

检查

每个循环的检查都具有头时间。以下为所检查的项目。

- 电池错误
- 存储卡
- DIP 开关
- I/O 总线
- 用户程序存储器

程序执行

此处指的是在用户程序中指令的处理时间。

- 该处理时间包含在 READY 状态下的循环任务，以及在发生中断状态下的中断任务指令所需的处理时间。不包括在程序间跳转的指令执行。
- 处理时间取决于指令数数量和具体使用的指令。同时也与指定的操作数有关。（例如，已处理数据的数量）。
- 在RUN或MONITOR模式下，OFF执行条件下的指令执行需要内部处理时间。只有在以下的情况时，无内部处理，并无需处理时间。
  - WAIT 状态下的循环任务
  - 在 JMP （004）、 CJP （510）、 CJPN （511）指令之间跳转的情况



附加信息

如需缩短循环时间，可以考虑进行以下的编程。

- 将程序分隔为循环任务，当有任务不需要被执行时，可将该任务置为 WAIT 状态。
- 当指令不需要被执行时，可在 JMP （004）、 CJP （510）、 CJPN （511）指令之间进行跳转。

外设服务

外设服务是指对外部服务设备的不定期事件进行服务。这包括外部设备产生的事件，也包括对外部设备的服务请求。

对于 CJ 系列 PLC，大多数的外设服务涉及 FINS 指令。系统中为每种服务分配一定量的时间，并且每个循环执行。如果在所分配的时间段内不能完成全部服务，则将持续到下一循环中完成剩余的服务。如果推迟了过多的外设服务，需使用 CX-Programmer 在 PLC 设置中将固定服务时间增加 10% 以上。

● 外设服务时间

外设服务时间设定为缺省的情况下，将上一循环时间的 10% 指定为总外设时间。因此，外设服务时间取决于前个循环时间。

因需要跨多个循环完成而导致外设服务延误时，通过 CX-Programmer 在 PLC 设置中将固定服务时间设为大于循环时间的 10% 的值。需注意的是延长服务时间的同时也将增加循环时间。

● 外设服务类型

单元	服务
特殊 I/O 单元事件服务	• 对于特殊 I/O 单元及 CPU 总线单元中的 FINS 命令进行非预定服务（例如请求启动外部中断任务）。 • 对于来自 CPU 的用于上述所有单元的 FINS 命令进行非预定服务。
CPU 总线单元事件服务	
内置 USB 端口服务	• 通过内置 USB 端口或 RS-232C 端口收到的来自编程设备、PT 或上位计算机的 FINS 或上位机链接命令的非预定服务 （例如传送程序、监视、强制置位 / 复位操作或在线编辑的请求）。 • 由内置 USB 端口或 RS-232C 端口传送的来自 CPU 单元的非预定服务（非请求的通信）。
RS-232C 端口服务	
通信端口服务	• 对使用通信端口 （内部逻辑端口）的 SEND （090） /SEND2 （491）、RECV （098） /RECV2 （492）、 CMND （490） /CMND2 （493）或 PMCR （260） /PMCR2 （264）指令执行的网络通信、串行通信或文件存储器访问进行服务。 • 对使用通信端口执行的后台任务进行服务。
文件存取服务	对存储卡或 EM 文件存储器进行文件读 / 写操作。
在线编辑服务	运行在线编辑时，在内置闪存存储器中对在线编辑的备份内容进行服务。
备份服务	程序或参数发生变更时，在内置闪存存储器中对备份的变更内容进行服务。



附加信息

特殊 I/O 单元、CPU 总线单元、内置 USB 端口、RS-232C 端口、文件访问和通信端口的服务在缺省配置状态下只分配整个循环时间的 10% （缺省值是可改变的）。
如果将服务分解到多个循环中，将推迟服务的完成，设定相同的分配时间（所有服务的时间相同）而不是在 PLC 设置中设定执行时间的百分比。

I/O 刷新

I/O 单元的处理时间取决于 PLC 中使用配置单元的类型和数量。每个单元的处理时间是固定的。

通过停止特殊 I/O 单元的循环 I/O 刷新，可以缩短处理时间。要停止特殊 I/O 单元的 I/O 刷新，需在 CX-Programmer 中设定 SIOU 刷新标签页的参数。下表列出了 PLC 单元的刷新处理。

单元			最大数据交换	数据交换区域
基本 I/O 单元			由单元决定	I/O 位区域
特殊 I/O 单元	在 CIO 区域的分配字		10 字 / 单元 （单元决定）	特殊 I/O 单元区域
	单元指定数据	CompoBus/S 主站单元	由单元决定	远程 I/O 通信设置的字
CPU 总线单元	在 CIO 区域的分配字		25 字 / 单元	CPU 总线单元区域
	在 DM 区域的分配字		100 字 / 单元	CPU 总线单元字
	单元指定数据 （参考右侧）	内置 EtherNet/IP 端口	由使用的 EtherNet/IP 功能决定	标签数据链接区域
		Controller Link 单元与 SYSMAC LINK 单元	由单元决定	数据链接设置字 （固定或由用户设定来分配）
		DeviceNet 单元	由单元决定	远程 I/O 通信设置字 （固定或由用户设定来分配）
		串行通信单元	协议宏决定	协议宏设置的通信数据
		Ethernet 单元	由单元决定	由特殊控制位操作的初始化的 Socket 服务的通信数据

注　指令已被执行而不在正常 I/O 刷新时，也可对单元进行的数据交换，这称为即时刷新。通过在某些指令后附加感叹号 （!），或使用 IORF （097）、 FIORF （225）、 DLNK （226）指令来进行即时刷新。

3-1-3　电源中断时的处理

CPU 单元运行时发生电源中断并确认后，将完成正在执行的指令并进行下面的电源中断处理。

将根据电源 OFF 中断任务的设定来执行电源中断时的处理。 *1

- 如果启用了电源 OFF 中断任务，任务被执行后， CPU 单元将立即复位。
- 如果没有启用电源 OFF 中断任务， CPU 单元将立即复位。 *2

*1　由于发生电源中断，电源 OFF 中断任务在 CPU 单元复位前执行。

*2 运行期间发生电源中断时，可在执行指定范围的指令后进行电源中断处理。

若 AC 电源发生瞬时电源中断的时间小于 10 ms，或 DC 电源小于 2ms 的情况下，系统将持续运行。如果电源状态较差的情况下，从检测到电源中断直到确认电源中断的时间可能会延长。

关于电源中断时的处理，请参考 A-5 电源中断操作。

3-2 CPU 单元运行模式

3-2-1 运行模式

可以通过设定运行模式对 CPU 单元的运行状态进行控制，并可控制是否能在 CPU 单元中进行设定。CPU 单元有三种运行模式。

● RUN 模式：

RUN 模式用于系统的实际运行并提供最快速的运行。
执行程序。
不可进行强制置位 / 复位，不能改变 I/O 存储器中的值，并且不能进行在线编辑。

● MONITOR 模式：

MONITOR 模式用于试运行和调整。
执行程序。
可进行强制置位 / 复位，可改变 I/O 存储器中的值，并且可以进行在线编辑。

● PROGRAM 模式：

PROGRAM 模式用于传送程序、 PLC 设置和创建 I/O 表。

不执行程序。

可通过使用连接到 CPU 单元的 CX-Programmer 更改运行模式。

3-2-2 运行模式的检查

CPU 单元的前面板指示灯

CPU 单元前面板上的 RUN 指示灯表示的运行模式如下所示。

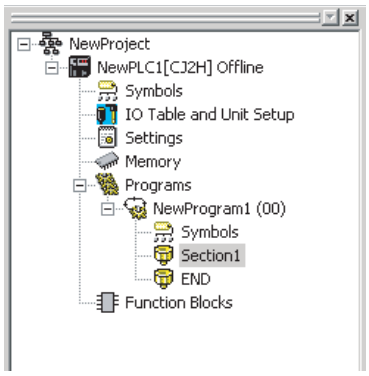
运行模式	CPU 单元的 RUN 指示灯	备注
PROGRAM 模式	OFF	---
RUN 或 MONITOR 模式	闪烁绿	用 CX-Programmer 查看为 RUN 模式或 MONITOR 模式。

CX-Programmer

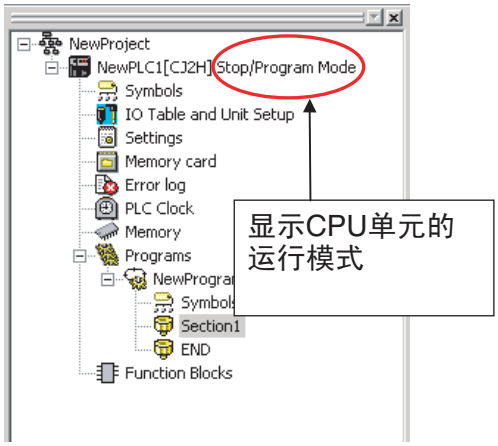
可通过 CX-Programmer 的工程树形图或状态栏检查运行模式。

● 工程树形图

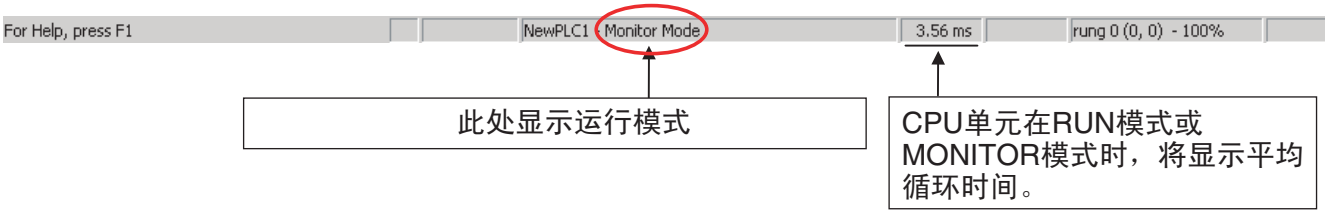
离线



在线



● 状态栏



附加信息

使用 CJ1W-PA205R 电源可实现外部 RUN 输出。RUN 模式或 MONITOR 模式时 CJ1W-PA205R 上的 RUN 输出 （触点输出）将变为 ON，出现致命错误的情况除外。

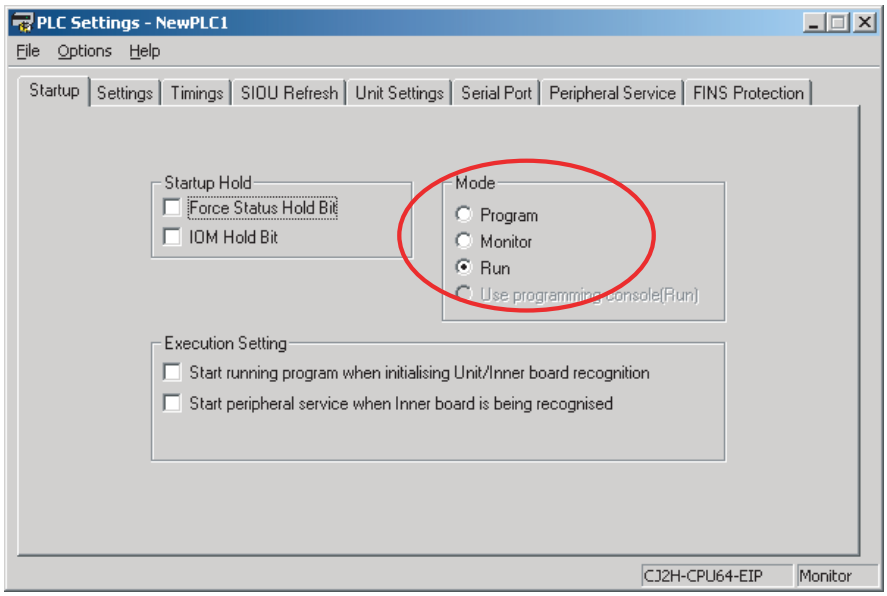
3-2-3 运行模式的变更

运行模式的变更

运行模式可通过 CX-Programmer 进行变更。

● 启动模式的变更

CPU 单元变为 ON 时的缺省运行模式为 RUN 模式。要将启动模式变更为 PROGRAM 模式或 MONITOR 模式，需通过 CX-Programmer 在 PLC 设置的启动设置中设定所需的模式。



● 启动后运行模式的变更

选用以下步骤之一。

- 在运行模式菜单中选择 PROGRAM、 MONITOR 或 RUN 模式。
- 在工程树形图中右键点击 PLC，然后在运行模式菜单中选择 PROGRAM、MONITOR 或 RUN 模式。

模式变更和电源中断后的存储器状态

当运行模式在 RUN 模式、MONITOR 模式和 PROGRAM 模式之间发生变更时，I/O 存储器的某些部分，如 CIO 区和工作区将被清除。这些区域被称为非保持区域。如需在运行模式变更时保持该区的内容，可在辅助区域中将 IOM 保持位 （A500.12）置为 ON。

模式变更	非保持区域	保持区域
	• I/O 区 • 数据链接区 • CPU 总线单元区 • 特殊 I/O 单元区 • 工作区 • 定时器 PV 和完成标志 • 变址寄存器 • 数据寄存器 • 任务标志 （是否保持辅助区域位 / 字取决于地址）。	• HR 区 • DM 区 • EM 区 • 计数器 PV 和完成标志 （是否保持辅助区域位 / 字取决于地址）。
从 RUN 或 MONITOR 到 PROGRAM 模式	清除 *1	保持
从 PROGRAM 到 RUN 或 MONITOR 模式		
从 RUN 到 MONITOR 或 MONITOR 到 RUN 模式	保持	保持
电源中断复位 *2	清除 *3	保持

*1 如果 IOM 保持位为 OFF，存储器被清除。如果为 ON，数据按如下情况被保持：

I/O 存储器保持位状态（A500.12）	I/O 存储器			分配给输出单元的输出位		
	PROGRAM 和 RUN/MONITOR 模式之间模式转换	致命错误		PROGRAM 和 RUN/MONITOR 模式之间模式转换	致命错误	
		FALS 以外的致命错误	FALS 的执行		FALS 以外的致命错误	FALS 的执行
OFF	清除	清除	保持	OFF	OFF	
ON	保持	保持		保持	OFF: 发生致命错误时， CPU 单元的 I/O 存储器中的值将被保持，但基本输出单元的输出将变为 OFF。	

*2 当发生电源中断复位时， IOM 保持位 （A500.12）将变为 OFF。如需保持状态，可使用 CX-Programmer 在 PLC 设置的启动保持设定标签页中选择 IOM 保持位选项框，然后传送 PLC 设置。

*3 任务标志将被清除，与 IOM 保持位 （A500.12）的状态无关。

运行变更后基本输出单元的输出状态

当运行模式在 RUN 或 MONITOR 和 PROGRAM 模式之间按缺省设置转换时，基本输出单元的输出将根据模式变更时的 I/O 存储器状态进行刷新。

以下基本输出单元的输出状态，取决于 IOM 保持位 （A500.12）为 ON 或 OFF。

- 如果 A500.12 为 OFF，当模式变为 PROGRAM 模式时，输出位将被清除，基本输出单元的输出也将变为 OFF。
- 如果 A500.12 为 ON，当模式变为 PROGRAM 模式时，输出位将被保持，基本输出单元的输出也将被保持。

正确使用注意事项

请注意将 IOM 保持位 （A500.12）设为 ON。

当模式从 RUN 或 MONITOR 变为 PROGRAM 时，输出的 I/O 存储器状态将不会被清除 （变为 OFF），并且变更为 PROGRAM 模式之前的状态将被保持。因此再次从 PROGRAM 模式变更为 RUN 或 MONITOR 模式时，输出将变为 PROGRAM 模式之前的 I/O 存储器的值。

如果由于致命错误 （包括 FALS （007）指令的执行）发生运行停止，即使 CPU 单元中的 I/O 存储器状态被保持，基本输出单元的所有输出也将变为 OFF。

PROGRAM 模式中基本输出单元的输出状态

在 PROGRAM 模式中，基本输出单元的输出也会随 I/O 存储器输出位的状态进行刷新。因此当使用 CX-Programmer 或其他支持软件变更 I/O 存储器中的输出位时，将从基本输出单元输出状态。

！ 警告

即使在程序被停止时（如，即使在 PROGRAM 模式下） CPU 单元也会刷新 I/O。因此，在变更分配给基本 I/O 单元、特殊 I/O 单元或 CPU 总线单元的存储器中任何部分的状态前，请先对安全进行充分确认。对分配给任一单元的数据进行变更，都可能导致连接到单元的负载发生不可预料的操作。下列操作将可能导致存储器的状态变更。

- 将 I/O 存储器数据从编程设备传送到 CPU 单元。
- 从编程设备变更存储器中的当前值。
- 从编程设备进行强制置位 / 复位。
- 从存储卡或 EM 文件存储器传送 I/O 存储器文件到 CPU 单元。
- 从上位计算机或从网络上的另一 PLC 传送到 I/O 存储器。

要确保基本输出单元的输出保持 OFF，需打开输出 OFF 位 （A500.15）（见注）。当所有输出为 OFF 时（即输出 OFF 位为 ON）， CPU 单元前部的 INH 指示灯将变亮。

注　运行模式发生变更且电源关闭再打开时，即输出保持为 OFF 时，输出 OFF 位 （A500.15）的状态被保持。

3-2-4 运行模式的详细信息

下表显示的是每个运行模式中的状态。

运行模式		PROGRAM	MONITOR	RUN
目的		停止运行和传送程序	测试运行和进行调整	主运行
程序状态		停止	执行	执行
RUN 指示灯		OFF	ON	ON
使用 CX-Programmer 的操作	创建 I/O 表	可以	不可以	
	传送 PLC 设置			
	下载程序			
	检查程序			
	在线编辑	可以		不可以
	通过强制置位 / 复位检查配线			
	更改 I/O 存储器的 PV			
	更改定时器 / 计数器 的 PV 与 SV			
	上传程序	可以		
	监控 I/O 存储器			
	监控程序			
I/O 刷新		执行		
循环任务		停止	运行开始时任务设为开始，并随 TASK ON 指令的执行开始任务。否则任务将被停止。	
中断任务		停止	满足中断条件时将被执行	
对基本输出单元的输出分配输出位 *1	IOM 保持位 OFF	变为 PROGRAM 模式后立即置 OFF*2	取决于程序	取决于程序
	IOM 保持位 ON	变位 PROGRAM 模式后状态被保持 *2		

*1 输出 OFF 位 （A500.15）为 ON 时，基本输出单元的输出将置为 OFF 而不管输出模式与 I/O 存储器状态。即使电源单元变为 ON，输出也将保持 OFF。

*2 即使在 PROGRAM 模式下，如使用支持软件或 PT 改变存储器状态，输出单元的输出将被刷新。

4

CPU 单元初始设定

本章描述了启动时对 CPU 单元执行的初始处理。

4-1 CPU 单元初始化概述 4-2
4-1-1 CPU 单元初始设定 4-2
4-2 PLC 设置 4-7
4-2-1 PLC 的缺省设定 4-7
4-3 I/O 表的创建 4-8
4-3-1 I/O 表 4-8
4-3-2 自动分配 4-9
4-3-3 手动分配 4-9
4-4 路由表的设定 4-10
4-4-1 路由表 4-10
4-4-2 需要路由表的情况 4-11
4-4-3 路由表的设定与传送 4-12
4-5 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字 4-14
4-5-1 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字 4-14
4-5-2 设定步骤 4-14
4-6 CPU 总线单元设置区 4-15
4-6-1 CPU 总线单元设置区 4-15
4-6-2 设定步骤 4-15

4-1 CPU 单元初始化概述

4-1-1 CPU 单元初始设定

CPU 单元前部的 DIP 开关与旋转开关的硬件设定，以及通过 CX-Programmer 或其他软件设置参数的软件设定都必须进行 CPU 单元所需的初始设定。

硬件设定

● DIP 开关

根据应用设置 DIP 开关。

位置	目的	设定方法
CPU 单元前部的 DIP 开关	主要目的： • 程序写保护：　位 1 ON • 启动时自动传送：　位 2 ON • 通过 RS-232C 的 Toolbus 连接：位 5 ON • 用户自定义位：　位 6 ON • 简捷备份：　位 7 ON	在电源关闭时进行设定，然后再打开电源。

● 旋转开关

在 CJ2 CPU 单元上对内置 EtherNet/IP 端口设置单元编号和节点地址的方法与 CPU 总线单元相同。根据单元编号的设置，在 CIO 区和 DM 区中将字分配到 CPU 总线单元区的内置 EtherNet/IP 端口。

关于硬件设定的详细信息，请参考 CJ2 CPU 单元硬件操作手册的 3-1 CPU 单元 （样本编号 W472）。

软件设定

下表给出了适用单元的软件设定的用途与设定方法。

单元	设定		用途	设定方法	通过编程设备创建的文件	备份目的地
CPU 单元	参数区	PLC 名称	防止 CPU 单元的连接不正确	CX-Programmer	CX-Programmer 项目文件 （.CXP）	内置闪存存储器
		PLC 设置	使用非缺省规格			
		注册 I/O 表	检查不正确的连接、需要自动分配或为未来的改变预留 I/O 字及其他特殊应用的设定			
		路由表	当连接了一个以上的 FINS 网络通信单元时必要 (*1)	CX-Integrator	路由表文件（.RTG）	
Ethernet 单元		CPU 总线单元设置区 （CPU 总线单元设置）（*2）	使用 Ethernet 单元	CX-Programmer	CX-Programmer 项目文件 （.CXP）	内置闪存存储器
Controller Link 单元（数据链接表）			对 Controller Link 使用用户自定义数据链接	CX-Integrator	数据链接表（.CL2）	
FL-net 单元			使用 FL-net 单元	CX-FLnet（FL-net 支持软件）	FL-net 支持软件设定文件 （.CSV）	
特殊 I/O 单元和 CPU 总线单元	I/O 存储器	DM 区字分配（*2）到特殊 I/O 单元或CPU 总线单元	使用特殊 I/O 单元或 CPU 总线单元	• CX-Programmer • 用户程序	CX-Programmer 项目文件 （.CXP）	内置 RAM （电池备份）

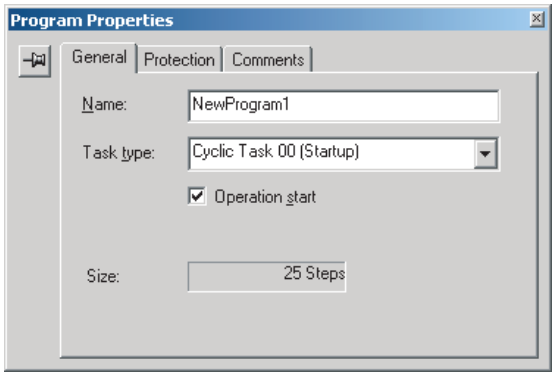
*1 以下单元被分类为 FINS 网络通信单元。

- 分类为 CPU 总线单元的网络通信单元：
  Controller Link 单元、 SYSMAC LINK 单元、 EtherNet 单元、 DeviceNet 单元和 FL-net 单元
- 对 FINS 网络通信使用 CJ2 CPU 单元的内置 EtherNet/IP 端口
- 使用带串行通信单元的串行网关并使用路由表
  路由表不适用于特殊 I/O 单元的网络通信 （例如， CompoNet 主站单元）。

*2 以上的特殊 I/O 单元或 CPU 总线单元的初始设定保存在 CPU 单元中。保存在特殊 I/O 单元和 CPU 总线单元中的数据和程序通过特定的单元支持软件单独创建，然后通过 CPU 单元传送到特殊 I/O 单元和 CPU 总线单元。

附加信息

程序任务的设定
使用 CX-Programmer 在以下的 Program Properties 设置分配程序到任务的初始设定。



● PLC 名称

PLC 名称为用户对 CPU 单元设定的名称。在 CX-Programmer 的 PLC 菜单中选择“PLC Info”-“PLC Info”进行该设定。

CX-Programmer 在线时，系统将检查注册到 PLC 的名称是否与工程中的 PLC 名称相符。

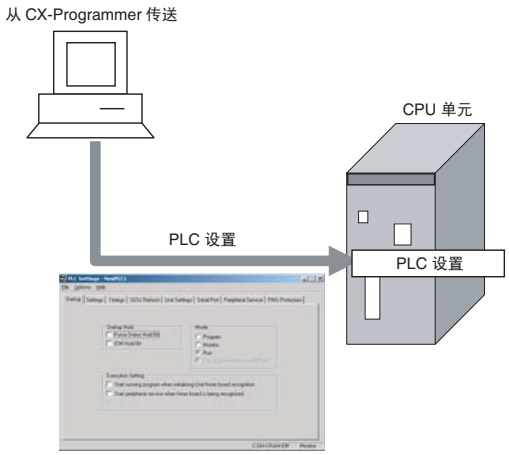
● PLC 设置

PLC 设置用于对使用非缺省设置的 CPU 单元进行改变。以下为 CPU 单元缺省设置的示例。

启动模式：RUN 模式
RS-232C 端口：上位链接 （Host Link）
固定服务时间：循环时间的 10%

要使用除这些缺省设置以外的设定，需使用 CX-Programmer 改变 PLC 设置，并将 PLC 设置传送到 CPU 单元。

⇒ 从 CX-Programmer 传送 PLC 设置

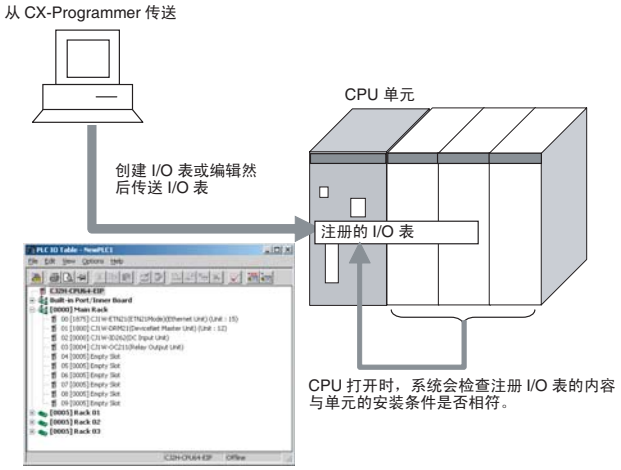


● I/O 表

在 CPU 单元中手动设置单元槽位或手动分配 I/O 时，创建 I/O 表来检测不正确的单元连接。缺省情况下，每次电源打开时以单元连接的顺序自动分配 CPU 单元的 I/O。

通过使用以下操作之一从 CX-Programmer 创建 I/O 表。

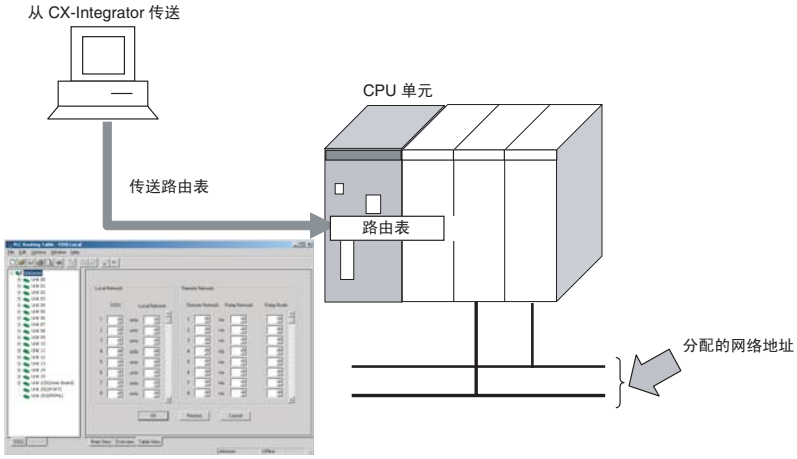
- 在线：对已连接单元创建 I/O 表的操作步骤。
- 离线：编辑 I/O 表 （带或不带安装的单元），然后将其传送到 CPU 单元。



● 路由表

如果 CPU 单元安装了一个以上的 FINS 网络通信单元 （CPU 总线单元），必须通过 CX-Integrator 创建路由表。在本地网络表中，网络地址被分配到每个 FINS 网络通信单元。

通过 CX-Integrator 设置本地网络表，并将其传送到 CPU 单元。



● CPU 总线单元设置区

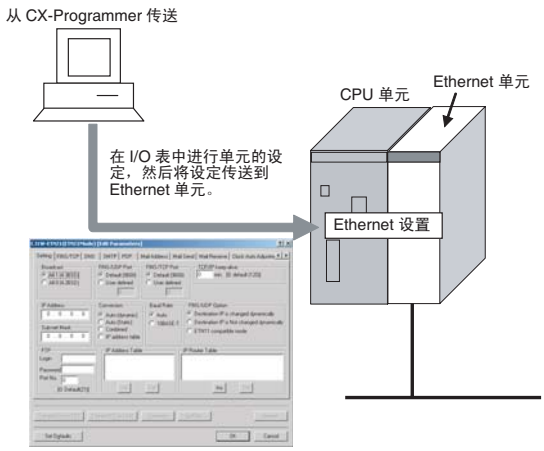
如果使用了特殊的 CPU 总线单元，例如 Ethernet 单元、Controller Link 单元或 FL-net 单元等，需对这些单元进行特定设定并将其传送到 CPU 单元。

示例：使用 Ethernet 单元

使用 CX-Programmer 进行 Ethernet 设定。

要进行 Ethernet 设定，在 CX-Programmer 的 I/O 表中右键单击 Ethernet 单元，然后选择“编辑 CPU 总线单元设定”。

在 CX-Programmer 的 I/O 表中对 Ethernet 单元进行设定，然后将 I/O 表传送到 CPU 单元。



附加信息

如果 Controller Link 单元使用用户自定义数据链接表，设置数据链接表，然后将其保存在 CPU 单元中。

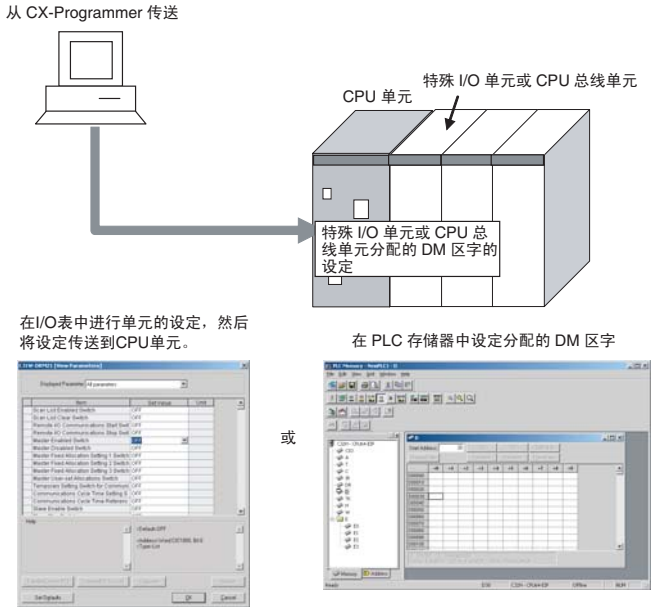
● 特殊 I/O 单元与 CPU 总线单元的 DM 区字分配

如果使用特殊 I/O 单元或 CPU 总线单元，需设定在 DM 区中分配的字，然后将设定传送到 CPU 单元。

通过 CX-Programmer 使用以下方法之一设定分配的 DM 区字。

- 在I/O表中右键单击各特殊I/O单元或CPU总线单元，然后选择“编辑CPU总线单元设定”（Edit CPU Bus Unit Settings）来设置分配的 DM 区字。
- 在 PLC 存储器中设置分配的 DM 区字。

通过CX-Programmer设置各特殊I/O单元或CPU总线单元分配的DM区字，并将设定传送到CPU单元。



4-2 PLC 设置

PLC 设置包含 CPU 单元的基本设定。如果 CJ2 CPU 单元使用非缺省设置，必须改变 PLC 设置中的参数。通过 CX-Programmer 可对 PLC 设置中的参数进行设定。

4-2-1 PLC 的缺省设定

下表提供了 PLC 设置中的缺省设定。要改变这些设定，需通过 CX-Programmer 编辑 PLC 设置，然后将 PLC 设置传送到 CPU 单元。

CX-Programmer PLC 设置标签页	设定项目		缺省设定
启动	启动保持	强制状态保持位	启动时不保持
		IOM 保持位	启动时不保持
	模式		RUN 模式
	执行设定	在初始化单元 / 内插板识别前开始运行程序	不开始运行
设定	执行过程	不检测低电池	检测
		检测重复刷新错误	检测
		停止 CPU 的指令错误	不检测
		不登记 FAL 的错误记录	登记错误记录
	后台执行		不在后台执行
	FB 中的通信指令设定	重试计数	0 次
		响应超时	2 s
时序	监视循环时间		1000 ms （1 s）
	固定循环时间		无设定
	定时中断间隔		10 ms
	电源掉电检测时间		0 ms
	电源掉电中断禁止		ON
SIOU 刷新	禁止 SIOU 周期刷新		不禁止
单元设定	基本 I/O 单元的输入响应时间		8 ms
串行端口 （RS-232C 端口内置到 CPU 单元）	模式 （设定模式必须将 CPU 单元 DIP 开关位 5 置 OFF）		上位链接 （Host Link）
外设服务	同步 / 异步通信		正常
	对所有事件设定时间		循环时间的 10%
FINS 保护	通过网络设定 FINS 写保护		禁止

关于 PLC 设置的详细信息，请参考第 9 章 PLC 设置。

4-3 I/O 表的创建

仅在以下情况创建 I/O 表。通过使用 CX-Programmer 来创建 I/O 表。

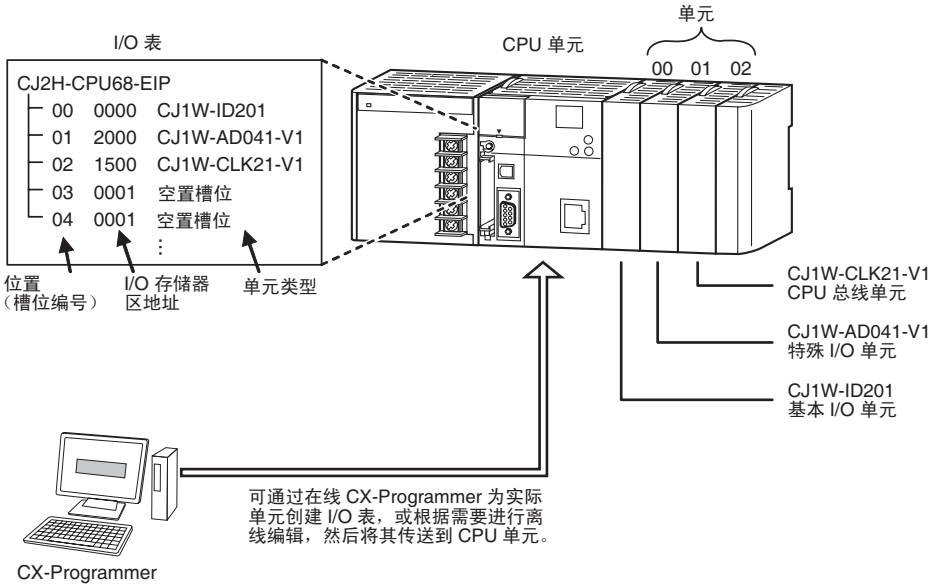
- 检查不正确的连接
- 保留字

在无需检查不正确连接并可使用自动分配的情况下，不需要创建 I/O 表。

4-3-1 I/O 表

连接到 CPU 单元的单元的类型和位置登记在 I/O 表中。如果 I/O 表被创建， CPU 单元打开时，系统将检查实际连接到 CPU 的单元的类型和位置是否与登记在 I/O 表中的数据相符。

CX-Programmer 在线时，可根据实际连接的单元创建 I/O 表，或当 CX-Programmer 离线时，手动设定并将其传送到 CPU 单元。



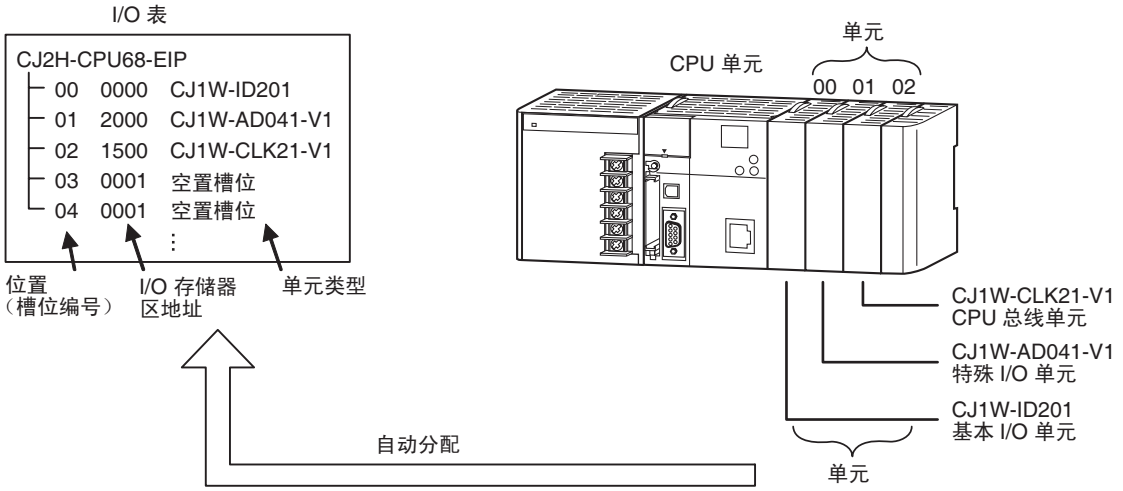
根据这个数据 CPU 单元会自动检验连接的单元 （实际 I/O 表）。

- 如果不相符，将发生 I/O 设置错误，并且运行不会开始。

4-3-2 自动分配

自动分配时，I/O 表不由用户创建。每次电源打开时，I/O 存储器会根据单元的位置分配到每个单元。分配的存储器用于单元的数据交换。

如果有未注册的 I/O 表，系统将不会检查单元配置。



4-3-3 手动分配

手动分配时，用户创建的 I/O 表称为注册 I/O 表。

可通过 CX-Programmer 使用以下方法之一创建 I/O 表。

- 根据实际单元配置创建 I/O 表
  该方法中，通过在线连接 CX-Programmer，根据实际安装单元的配置自动创建 I/O 表。
- 手动编辑 I/O 表并将其传送到 CPU 单元
  该方法中，在离线时通过使用 CX-Programmer 将各单元注册到所需的位置，据此手动创建 I/O 表，完成的 I/O 表将被传送到 CPU 单元。

不论使用哪种方法，PLC 打开时，如果连接单元的配置与 I/O 表中的单元配置不同，将出现 I/O 设置错误（A401.10 将转为 ON）并且不会开始运行。

关于 I/O 分配方法的详细信息，请参考 8-1 I/O 分配。关于创建 I/O 表的详细信息，请参考 8-2 设定 CPU 总线单元和特殊 I/O 单元。

4-4 路由表的设定

如果 PLC 安装了一个以上的 FINS 网络通信单元，必须通过 CX-Integrator 进行路由表的设定，并执行以下操作。

- 通过支持软件或梯形图程序中的指令切换被访问的网络。
- 跨网络层执行通信。

4-4-1 路由表

在欧姆龙网络通信单元（CPU 总线单元）之间使用的通信方法称为 FINS，但传输路径不在 FINS 协议中指定。执行 FINS 网络通信时，由路由表定义网络地址设定与所需的通信路径。

适用单元

以下单元的网络地址在路由表中设定。

- CPU 总线单元的网络通信单元：
  Controller Link 单元、SYSMAC LINK 单元、EtherNet 单元、用于 FINS 网络的 EtherNet/IP 单元、DeviceNet 单元和 FL-net 单元
- 用于 FINS 网络通信的 CJ2 CPU 单元内置 EtherNet/IP 端口
- 使用串行网关跨网络层执行通信时，串行通信单元上的串行端口

被分类为特殊 I/O 单元的网络通信单元（例如， CompoNet 主站单元）不需要在路由表中设定。

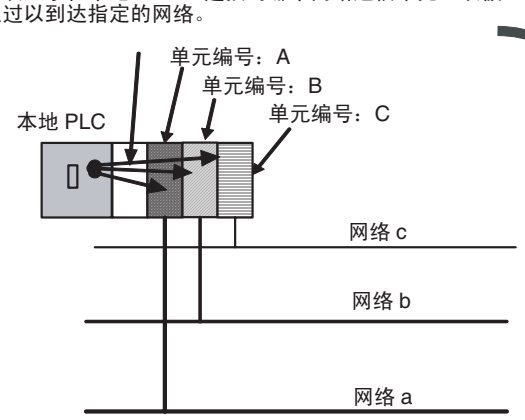
路由表的组成

路由表由以下两种表组成。

● 本地网络表

对连接在本地 PLC 的网络通信单元（CPU 总线单元）设置网络地址 1 ～ 127。如果 PLC 安装了一个以上的网络通信单元， CPU 单元使用本地网络表来识别安装到本地 PLC 的网络通信单元以及对应的通信网络。

本表显示在本地 PLC 上连接的哪个网络通信单元必须被通过以到达指定的网络。



本地网络表

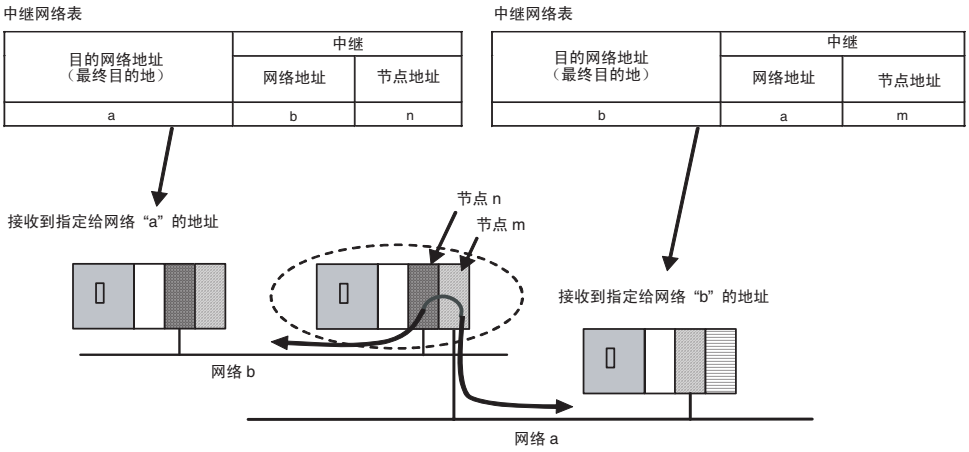
网络地址	单元编号
a	A
b	B
c	C

本地网络表提供了连接到本地 PLC 的通信单元的单元编号以及连接到该单元用于 FINS 通信的 FINS 网络的对应地址。

● 中继网络表

设定中继网络表用于跨网络层执行 FINS 通信。中继网络表指示如何中继通信直到报文到达最终目的地。该表提供了到达本地 PLC 并未连接的目的网络 （最终网络）路径上的第一个中继点 （即第一个到达的点）的网络地址与节点地址。

通过这些中继点到达目的网络。



4-4-2 需要路由表的情况

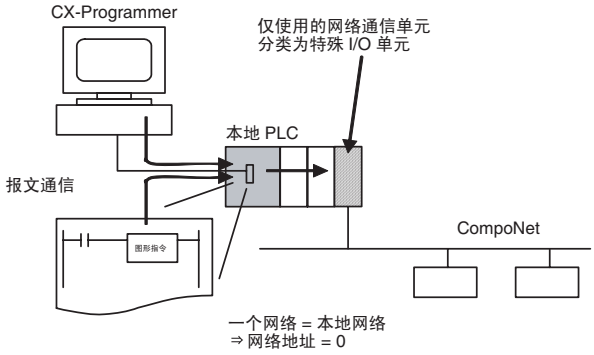
根据应用的需要，在 CPU 单元中设置路由表 （本地网络表和中继网络表）。

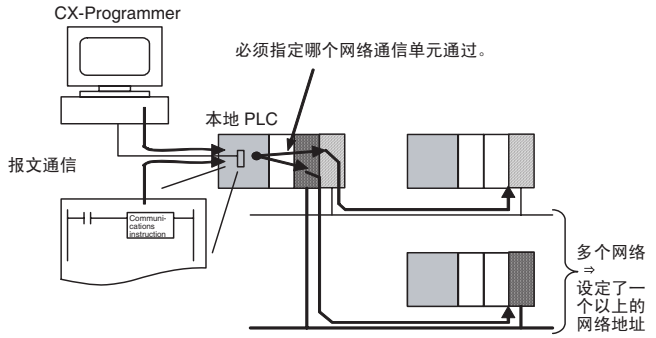
正确使用注意事项

需要本地网络表的条件

中继没有执行时，如果 PLC 连接了一个以上的网络通信单元，需要本地网络表。

下表显示了根据网络使用条件是否需要路由表。

网络使用条件	系统配置	路由表	
		本地网络表	中继网络表
PLC 没有连接网络通信单元，或仅连接了分类为特殊 I/O 单元（如 CompoNet）的网络通信单元。（*1）		不需要	不需要

网络使用条件	系统配置	路由表	
		本地网络表	中继网络表
PLC 连接了一个以上的分类为 CPU 总线单元的通信单元。		网络中的所有 PLC 需要 *2	跨网络层执行通信时需要 *3

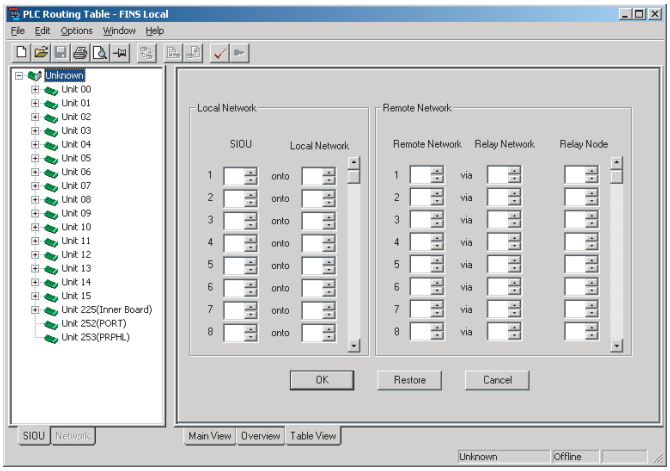
*1 如果通过编程设备来访问网络中的节点，将远程网络地址设为 0。

*2 如果没有本地网络表，并通过编程设备或梯形图程序中的指令来访问网络地址 0， FINS 通信将自动访问单元编号最小的网络通信单元的网络。通过这个方法，在没有设定本地网络表时，可将要访问的网络的网络通信单元的编号设为最小来访问该网络。

*3 如果没有跨网络层执行通信，则无需设定中继网络表。

4-4-3 路由表的设定与传送

1. 启动 CX-Integrator
2. 在工具菜单中选择 “开启路由表”（Start Routing Table）。
3. 选择 “FINS 本地”（FINS Local）。
4. 在 Table View 标签页中编辑路由表。



5. 在线连接 CX-Integrator，然后在选项菜单中选择 “传送到 PLC”（Transfer to PLC）将路由表传送到 CPU 单元。
6. 在文件菜单中选择 “保存本地路由表文件”（Save Local Routing Table File）。

正确使用注意事项

路由表数据文件
路由表保存在 CX-Integrator 创建的单独文件 （.rtg）中。不包含在 CX-Programmer 的项目文件（.cxp）中。

4-5 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字

如果使用特殊 I/O 单元或 CPU 总线单元，必须进行这些设定。使用 CX-Programmer 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字。

4-5-1 设定分配到特殊 I/O 单元和 CPU 总线单元的 DM 区字

特殊 I/O 单元和 CPU 总线单元的初始设定是 CPU 单元 I/O 存储器的 DM 区分配字的设定。电源打开时，分配的 DM 区字的设定从 CPU 单元传送到特殊 I/O 单元和 CPU 总线单元。

分配字的实际设定和传送时间取决于特殊 I/O 单元和 CPU 总线单元的类型和区。详细信息，请参考特殊 I/O 单元或 CPU 总线单元的操作手册。

从以下地址分配 DM 区字。

● 特殊 I/O 单元的初始设定

每个单元根据单元前部旋转开关上的单元编号设定分配 100 个字，从 D20000 ～ D29599 之间。

● CPU 总线单元的初始设定

每个单元根据单元前部旋转开关上的单元编号设定分配 100 个字，从 D30000 ～ D31599 之间。



附加信息

对特殊 I/O 单元和 CPU 总线单元的设定也包含在特定单元的 CPU 总线单元启动区中，例如 Ethernet 单元和 Controller Link 单元，数据和程序也保存在特殊 I/O 单元和 CPU 总线单元中（使用特殊单元的支持软件创建）。

4-5-2 设定步骤

使用以下方法之一进行设定。

- 选择CX-Programmer中PLC I/O表窗口中的单元，对特殊I/O单元和CPU总线单元进行设定，然后将设定传送到 CPU 单元。
- 在CX-Programmer的PLC存储器窗口中设定分配DM区字的单独地址的数据，然后将设定传送到CPU单元。
- 通过使用 MOV （021）或用户程序中的其他指令设定分配 DM 区字的单独地址的数据。

关于通过 CX-Programmer 分配 DM 区字到 CPU 总线单元和特殊 I/O 单元的步骤，请参考 8-2 设定 CPU 总线单元和特殊 I/O 单元。

4-6 CPU 总线单元设置区

4-6-1 CPU 总线单元设置区

CPU 单元安装了保存特定 CPU 总线单元设定的设置区。以下三种类型的单元与设定使用该区。

- Ethernet 单元： Ethernet 设定
- Controller Link 单元： 数据链接表 （使用用户自定义表时）
- FL-net 单元： FL-net 设定

4-6-2 设定步骤

使用以下支持软件进行设定。

数据	支持软件
Ethernet 单元设定	在 CX-Programmer 的 I/O 表中对特殊 I/O 单元或 CPU 总线单元进行设定，或使用 HTML 进行设定。
Controller Link 单元数据链接表	在 CX-Integrator 中使用数据链接表进行设定。
FL-net 单元设定	使用 CX-FLnet FL-net 支持软件进行设定。

用于 CPU 总线单元设置区的存储器

CPU 总线单元设置区可使用的存储器最大容量为 10,752 字节。根据 PLC 中 CPU 总线单元的组合设计系统，使得用于 CPU 总线单元设置区的存储器在限制范围内。如果超出限制，一些单元可能只能使用缺省设置运行，一些单元可能不能运行。

● 用于 CPU 总线单元设置区的存储器

单元	型号	占用长度（字节）
Controller Link 单元	CJ1W-CLK21-V1	512
Ethernet 单元	CJ1W-ETN11	412
	CJ1W-ETN21	994
FL-net 单元	CJ1W-FLN22	988
EtherNet/IP 单元	CJ1W-EIP21	0
串行通信单元	CJ1W-SCU21/31/41-V1	
DeviceNet 单元	CJ1W-DRM21	
位置控制单元	CJ1W-NCF71	
运动控制单元	CJ1W-MCH71	
保存和处理单元	CJ1W-SPU01-V2	

占用 0 字节的单元不使用 CPU 总线单元设置区。

5

编程

本章描述了 CJ2 CPU 单元编程的基本原理。

5-1 编程 5-3
5-1-1 编程概述 5-3
5-1-2 梯形图基本概念 5-5
5-1-3 ST 语言 5-6
5-1-4 SFC 概述 5-7
5-2 任务 5-9
5-2-1 任务概述 5-9
5-2-2 循环任务 5-12
5-2-3 中断任务 5-17
5-2-4 设计任务 5-26
5-3 段 5-36
5-3-1 段概述 5-36
5-4 功能块 5-38
5-4-1 功能块 5-38
5-4-2 功能块特性 5-38
5-4-3 功能块规格 5-40
5-5 符号（变量） 5-41
5-5-1 概述 5-41
5-5-2 符号（变量）类型 5-41
5-5-3 全局符号（变量） 5-43
5-5-4 本地符号（变量） 5-43
5-5-5 网络符号（变量） 5-44
5-5-6 功能块中的变量 5-47
5-5-7 符号（变量）数据类型 5-48
5-5-8 符号（变量）的自动地址分配 5-50
5-6 指令 5-51
5-6-1 指令的基本理解 5-51
5-6-2 定义操作数 5-57
5-6-3 数据格式 5-65

5-6-4 I/O 刷新时序 5-68
5-7 变址寄存器 5-74
5-7-1 什么是变址寄存器? 5-74
5-7-2 使用变址寄存器 5-74
5-7-3 与变址寄存器相关的处理 5-78
5-7-4 监控变址寄存器 5-78
5-7-5 在任务之间共享变址寄存器和数据寄存器 5-80
5-8 指定地址偏移 5-82
5-8-1 概述 5-82
5-8-2 地址偏移应用的示例 5-84
5-9 程序的检查 5-86
5-9-1 CX-Programmer 输入时出错 5-86
5-9-2 用 CX-Programmer 检查程序 5-86
5-9-3 用仿真工具调试 5-86
5-9-4 程序执行检查 5-88
5-10 注意事项 5-92
5-10-1 条件标志 5-92
5-10-2 特殊程序段 5-96

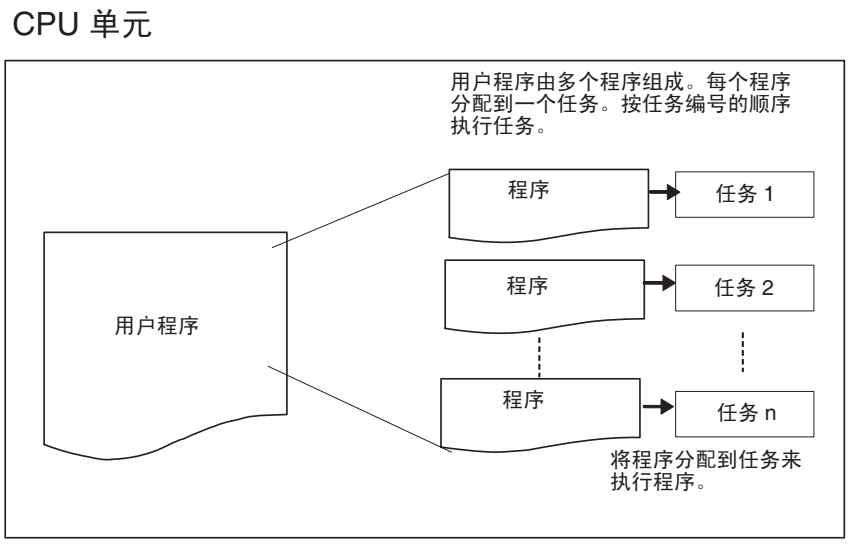
5-1 编程

5-1-1 编程概述

用户程序的结构

用户程序通过 CX-programmer 创建。程序由以下部分组成。

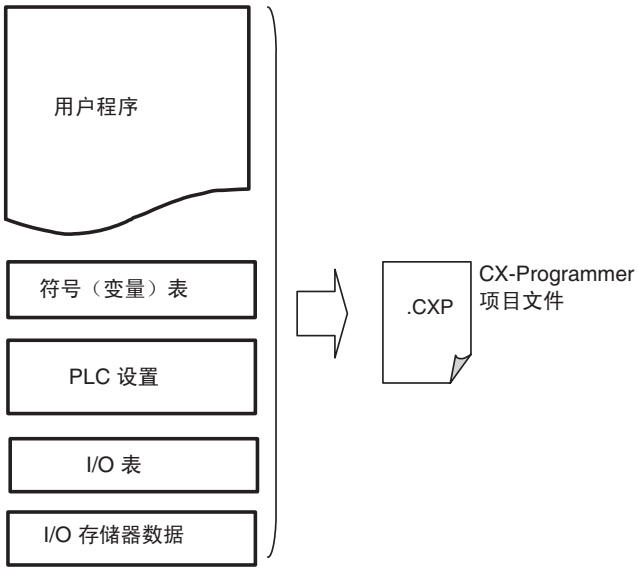
- 程序
  整个用户程序由多个程序组成。每个程序以 END （001）指令结束。
- 任务 （最小可执行单元）
  将程序分配到任务来执行程序 （在 CX-Programmer 中，在特性中指定任务编号作为程序属性）。任务包含循环任务 （通过正常循环处理执行）、中断任务 （满足中断条件时执行）、定时中断任务 （在规定的时间间隔执行）及掉电中断任务 （在电源中断时执行）。



- 段
  通过 CX-Programmer 创建和显示程序时，一个程序可分为任意个部分。每个部分称为一个段。段通常用于使程序便于理解。
- 子程序和功能块
  在一个程序中，可以创建子程序和功能块。

用户程序数据

整个用户程序和其他参数，如符号（变量）表、PLC 设置数据、I/O 表及 I/O 存储器数据等，保存在 CX-Programmer 项目文件 （.CXP）中。

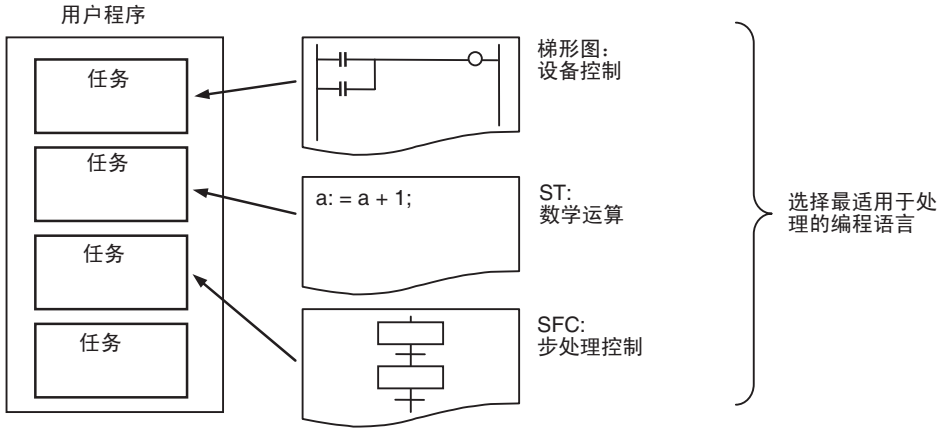


编程语言

可使用以下三种编程语言中的任一种编写程序。

- 梯形图
- 结构文本 （ST）
- 顺序功能图 （SFC）

以梯形图、 ST 或 SFC 编写的程序分配给任务来执行程序。



可根据不同目的使用不同语言，对每个处理使用最适用的语言，例如梯形图编程用于设备控制，或 ST 用于数学运算。

每种语言可在以下位置使用。

位置 语言	任务	SFC 中的动作程序和转换程序	功能块定义中的算法
梯形图	是	是	是
ST	是	是	是
SFC	是	---	---

可对每种语言使用以下寻址方法。

寻址方法 / 语言	物理地址	符号 （变量）
梯形图编程	是	是
ST	*1 视情况而定	是
SFC	*2 视情况而定	*2 视情况而定

*1 可通过将物理地址分配给符号 （变量）来使用物理地址。

*2 可在 SFC 图的布尔 （Boolean）动作和转换中使用物理地址和符号 （变量）。

程序容量

可用于所有用户程序的 CJ2 CPU 单元的最大程序容量 （即所有任务的最大容量）由下表给出。

型号	程序容量	I/O 容量
CJ2H-CPU68-EIP	400K 步	2,560 点
CJ2H-CPU67-EIP	250K 步	
CJ2H-CPU66-EIP	150K 步	
CJ2H-CPU65-EIP	100K 步	
CJ2H-CPU64-EIP	50K 步	

可以通过在 CX-programmer 中选择 “显示” -“显示存储器”检查程序容量。

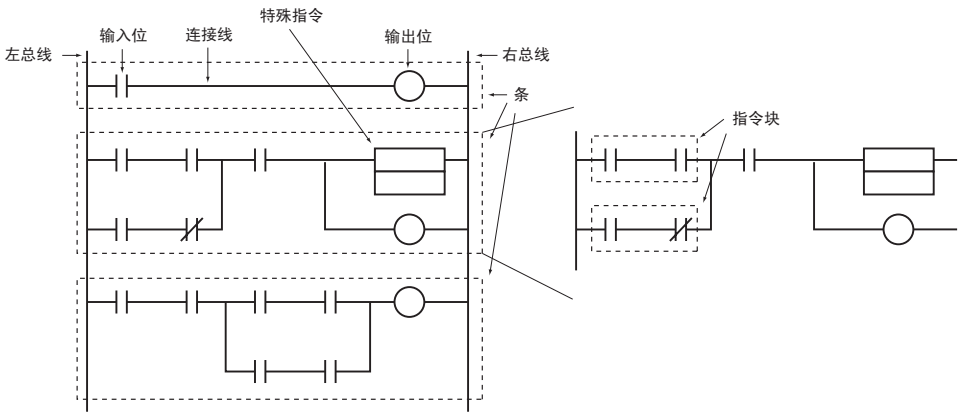
梯形图指令的长度取决于使用的特殊指令与操作数。详细信息，请参考 A-2 指令执行时间和步数。

5-1-2 梯形图基本概念

梯形图逻辑是一种根据与电路相似的形式编写的 PLC 基本语言。指令的执行顺序是按记录在存储器中的指令次序 （助记符的顺序）进行。请正确地理解编程的基本概念以及执行顺序。

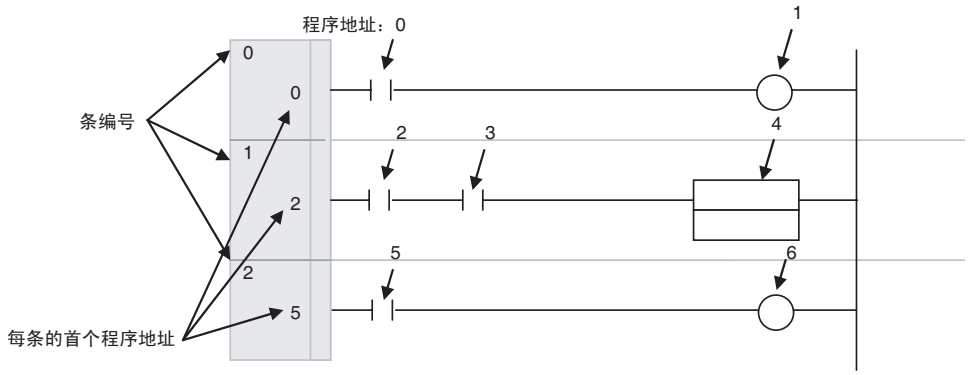
● 梯形图的基本结构

梯形图由左和右总线、连接线、输入位、输出位和特殊指令组成，一段程序由一个或多个程序条组成。当总线平行分开时，一个程序条是一个可分割的单元。对助记符而言，一个条是包含从 LD/LD NOT 指令开始到下一个 LD/LD NOT 指令前输出的所有指令，一个程序条由表示逻辑加载的 LD/LD NOT 指令开始的指令块组成。



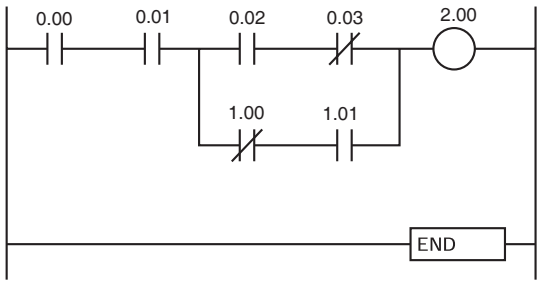
附加信息

对于 CX-Programmer，每个条的左侧显示条编号和每条的首个程序地址。



● 助记符

它有程序地址，一个程序地址对应一条指令。程序地址是由从 0 开始的六位组成。



程序地址	指令（助记符）	操作数
0	LD	0.00
1	AND	0.01
2	LD	0.02
3	ANDNOT	0.03
4	LDNOT	1.00
5	AND	1.01
6	OR LD	
7	AND LD	
8	OUT	2.00
9	END	

5-1-3 ST 语言

ST （结构文本）语言是一种由 IEC 61131-3 标准定义的用于工业控制 （PLC 为主）的高级语言代码。标准控制语句、操作和功能使 ST 语言适用于难以用梯形图程序编写的数学运算。

ST 语言不支持所有可用梯形图编写的处理。支持符合 IEC 61131-3 标准的 ST 语言。

附加信息

关于 ST 编程规格、表示方法和输入步骤的详细信息，请参考 CX-Programmer 操作手册：功能块和结构文本 （样本编号 W447）。

● ST 编程的特性

- 具有多种控制语句可用，例如循环语句和 IF-THEN-ELSE 语句，多种运算符例如数学运算符、比较运算符、与 / 或运算符，以及多种数学函数、字符串提取和合并功能、存储卡处理功能、字符串传送功能和三角函数。
- 可通过高级语言例如 C 语言来编写程序，并可包含注释以使程序简单易读。
- ST 程序像普通程序一样可被上传和下载，但不能在任务单元中上传和下载 ST 程序任务。
- 功能块 （梯形图或 ST 语言）可在 ST 程序中引用。
- 一维数组变量可支持应用中的简单数据处理。

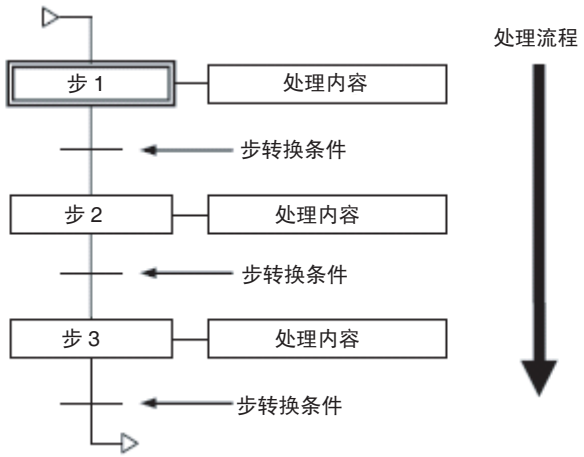
● 规格

项目	规格
ST 程序单元	• 任务 • 功能块定义的算法 • SFC 中的动作程序和转换程序
地址指定步骤	使用符号 （变量）指定地址。可分配任何地址给符号 （变量）。
强制置位 / 强制复位	支持
在线编辑	ST 程序编辑

5-1-4 SFC 概述

SFC （顺序功能图）语言是一种图形编程语言，用于说明以控制顺序处理为主的步处理程序。

SFC 为步流程的图形表示，带有步处理和每步动作的条件说明，允许用户对顺序处理的控制进行编程。



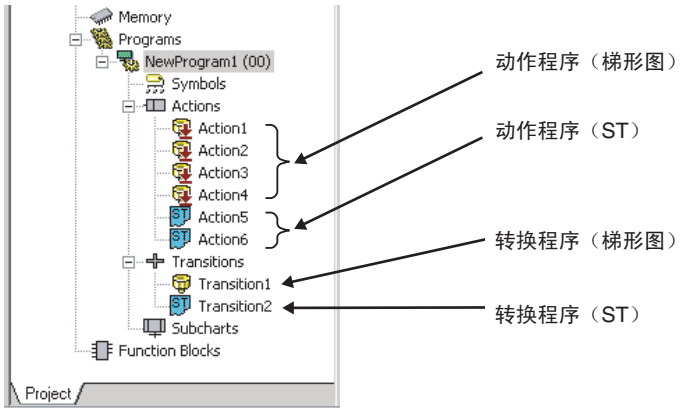
附加信息

关于 SFC 编程规格、表示方法和输入步骤的详细信息，请参考 CX-programmer 操作手册：SFC 编程 （样本编号 W469）。

● SFC 编程的特性

梯形图和 ST 作为 SFC 程序的一部分

SFC 程序内的步动作和转换条件也适用于梯形图或 ST。



● SFC 图的同时显示 / 编辑与动作 / 转换程序

在线编辑和调试

SFC 可进行在线编辑。此外，调试时可隐藏动作块，并可检查步处理状态。即使动作块被隐藏， SFC 图编辑器将在程序视图中显示动作程序。

● 规格

项目	规格
SFC 程序单元	任务 （1 任务 = 1 SFC 图）
支持 SFC 程序的任务分配	循环或附加循环任务
SFC 要素	步、转换、动作、跳转、子图 注：SFC 要素自动登记到本地变量中。
地址指定方法	可对 SFC 中的布尔 （Boolean）动作或转换指定符号 （变量）或物理地址。 如果使用 ST 编程，则不能指定物理地址。
强制置位 / 强制复位	可在线对逐步执行的转换进行强制置位 / 强制复位。
在线编辑	启用 SFC 编辑、动作添加 / 删除和转换添加 / 删除。

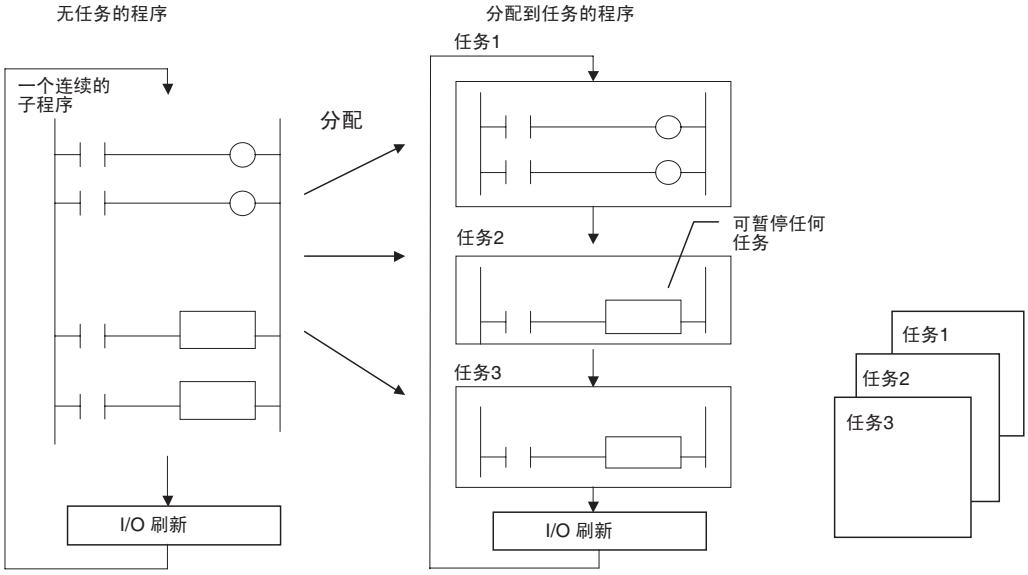
5-2 任务

5-2-1 任务概述

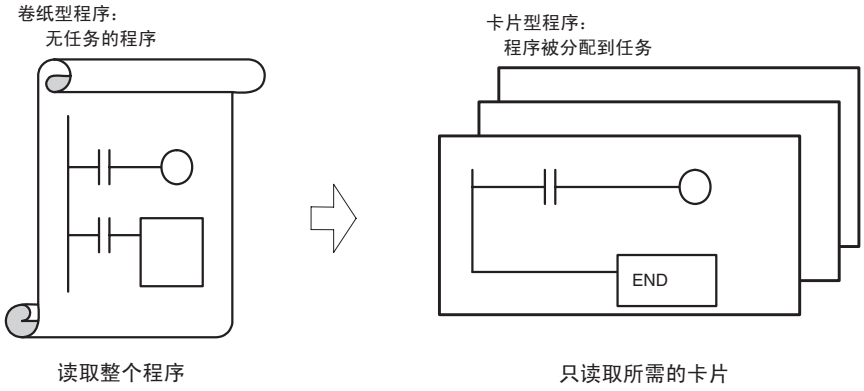
任务的概念

任务用于将程序分成大的单元并按执行每个单元或程序的顺序进行指定。任何一个任务中的指令可用于启用或禁止其他任务的执行。

这就意味几个程序块可组成不同的任务，根据当前生产模式或执行的步骤，仅执行特定的任务。因此操作 （循环时间）由于程序仅在需要时执行而大大改进。



不带任务的程序就像从开始处读一卷纸，带任务的程序则像单独读卡片。每张卡片可启用或禁用其他卡片。读到被禁用的卡片时将跳过。

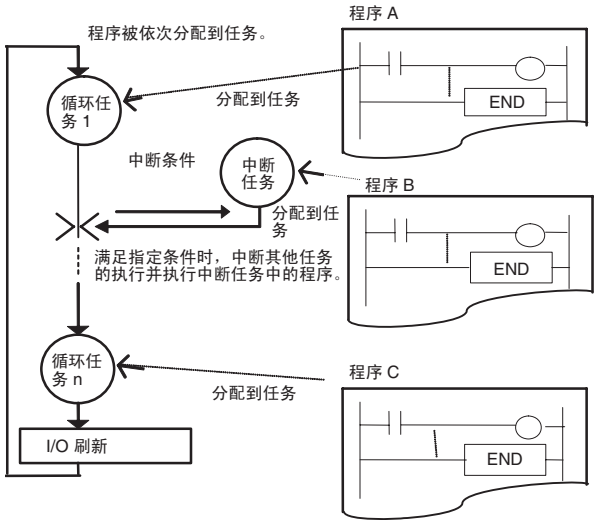


有两种基本的任务类型。

任务类型	描述	适用的程序语言	执行条件
循环任务	每循环执行一次。	梯形图 SFC ST	以下方法之一用于将任务置于 “就绪”状态时执行。 • 使用 CX-Programmer 设定“运行开始时激活”（程序特性设定） • 通过任务控制指令将任务转为 ON
中断任务 *1， *2	满足特定条件时执行。正在执行的处理被中断。	梯形图 ST	当满足所设定的中断条件时，中断任务置为 “就绪”状态。可对以下每个中断任务设定条件。 • 掉电中断任务 • 定时中断任务 • I/O 中断任务 • 外部中断任务

*1 通过使用任务控制指令来将中断任务转为 ON，在中断任务中可将循环执行（即每循环执行一次）作为循环任务来执行。（这些任务称为附加循环任务）。

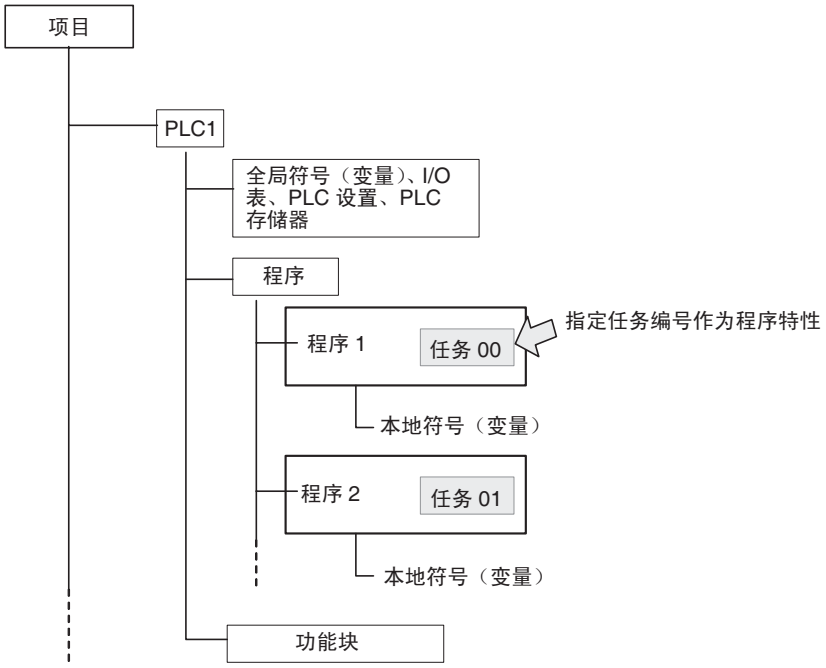
*2 请勿在中断任务中使用 SFC 程序。它们将无法正常执行。但中断任务作为附加循环任务执行时，SFC 程序可用。



任务的 CX-Programmer 操作

使用 CX-Programmer 设定程序的任务编号作为程序特性。

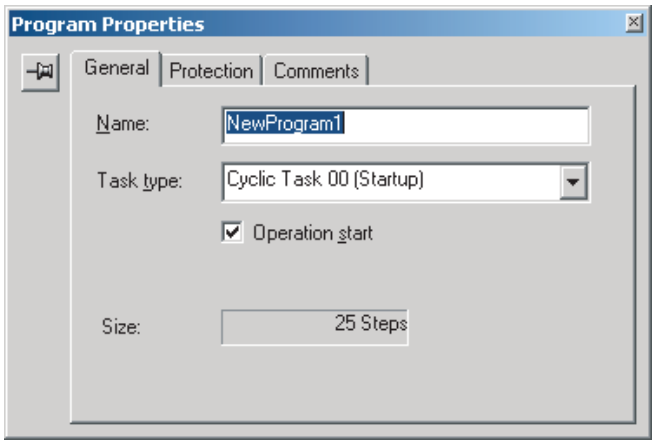
● 任务编号的设定



定义每个程序的任务类型和编号作为其属性。

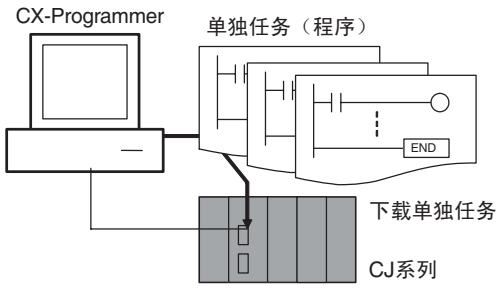
1. 选择 “View/Properties”，或者单击右键并在弹出菜单上选择 “Properties”，显示将要分配给任务的程序。

2. 选择 “General”标签，并选择其中的 “Task Type”和 “Task No”项。对于循环任务，如果想在操作开始时执行任务，单击 “Operation start”检查框。



● 下载或上传单独任务

单独的程序任务可通过 CX-Programmer 上传和下载。



● 功能块使用的限制

对包含功能块的程序不能下载单独任务 （可以上传）。

● 使用 SFC 程序和 ST 程序的限制

分配到 SFC 程序或 ST 程序的任务不能单独上传或下载。

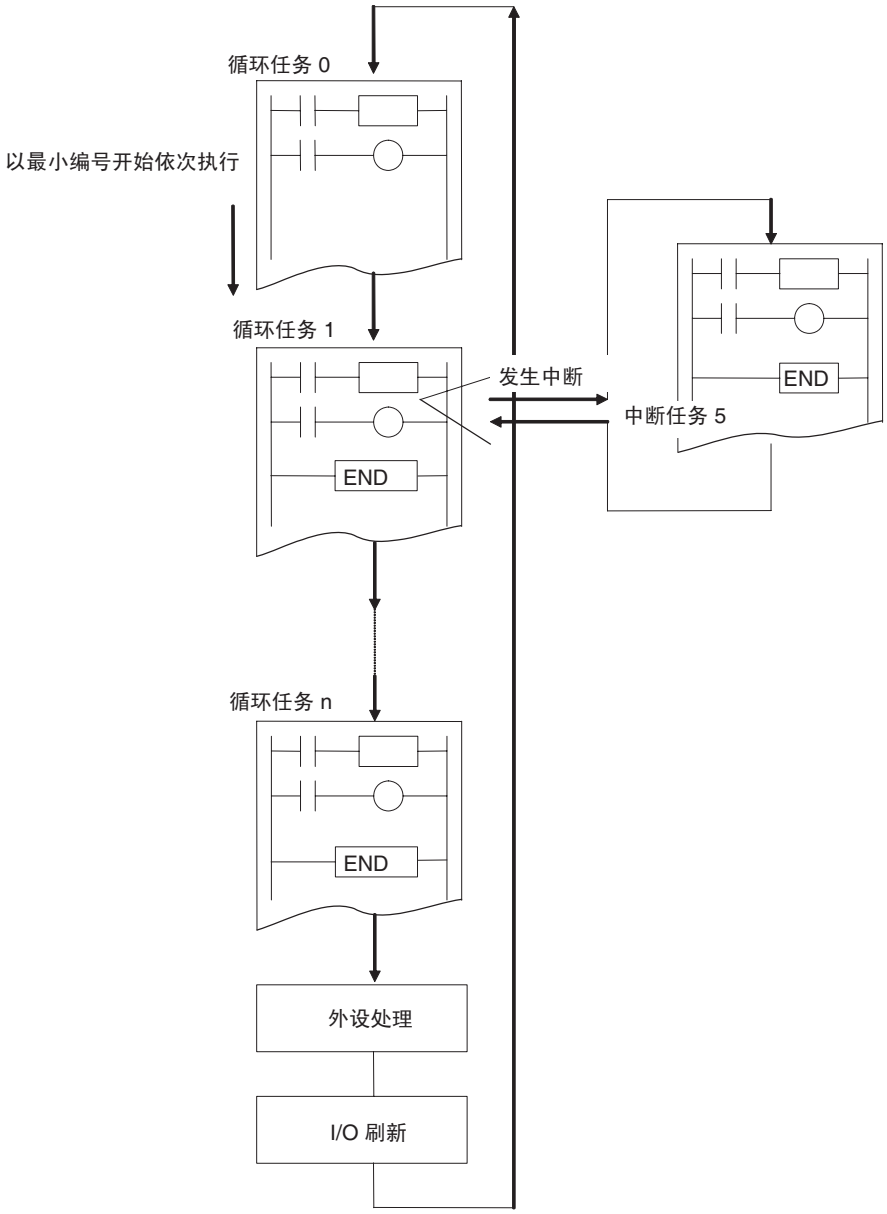
5-2-2 循环任务

循环任务

循环任务是以最小任务编号为开始的顺序在每循环执行一次。共有 128 个任务可被使用 （循环任务编号 0 ～ 127）。可通过使用 CX-Programmer 或任务控制指令，设定 “运行开始时激活”的特性来开始任务。关于任务控制指令的信息，请参考 A-2 指令执行时间和步数。如果使用附加循环循环，最多可使用 384 个循环任务。

正确使用注意事项

请勿使用任务控制指令 （TKON （820）和 TKOFF （821））来控制包含 SFC 程序的任务。否则将不能正常执行。



附加信息

- 所有条件标志 （ER、 CY、相等、 AER 等）和指令条件在任务开始时被清除。因此，在两个任务间的条件标志不可读。
- 联锁（如 IL 和 ILC 指令）、跳转（如 JMP、CJP 和 JME 指令）和子程序（如 SBS、RET 和 SBN 指令）必须在单独任务内完成。例如，跳转不能在一个任务和另一个间执行。如果一个以上的任务使用了子程序，需使用全局子程序 （GSBS （750）、 GRET （752）或 GSBN （751）指令）。

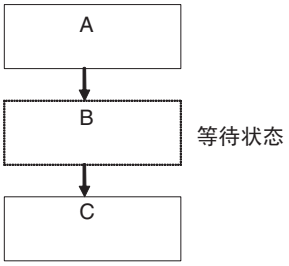
循环任务状态

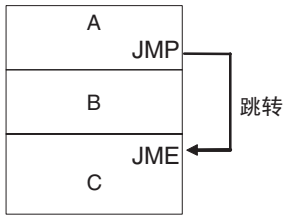
● 就绪状态

循环任务以最小任务编号开始依次执行。运行模式从 PROGRAM 转换为 RUN 或 MONITOR 模式时，“运行开始时激活”的属性将被设为可执行。 TASK ON 指令可用于将状态从 “等待”转换为 “就绪”。 TASK OFF 指令用于停止本地任务，在 TASK OFF 指令后的任务将不会被执行。

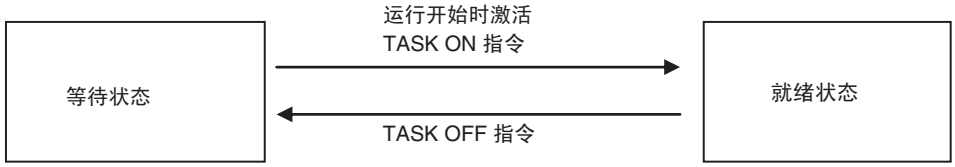
● 等待状态

在 “等待”状态中，任务的指令将不会执行。使用 TASK OFF 指令将使状态从 “就绪”转换为 “等待”。通过将全部的用户程序分配到任务，然后用任务控制指令开始和停止任务来缩短循环时间。需要执行全部程序的不同部分时，可将程序分配到任务。





● 状态转换



附加信息

“等待”状态的效果相当于一个跳转 （JMP-JME），处于 “等待”状态的任务的输出状态将保持不变。

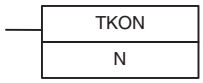
使用循环任务

● 任务控制指令

使用任务控制指令来开始或停止程序的一个循环任务。可使用指令根据下列表中的程序语言来开始和停止任务。

- 梯形图编程和 ST 编程： TKON(820) （TASK ON 指令）和 TKOF(821) （TASK OFF 指令）
- SFC 编程： SFCON(789) （SFC ON 指令）和 SFCOFF(790) （SFC OFF 指令）

- TASK ON 和 TASK OFF 指令



N = 任务编号

当执行条件为 ON 时，任务将变为“就绪”状态。对应的任务标志将转为 ON。



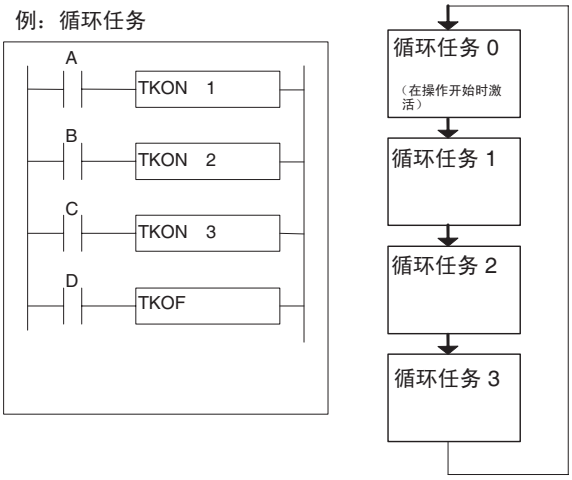
N = 任务编号

当执行条件为 ON 时，任务将变为“等待”状态。对应的任务标志将转为 ON。

在任何时候，都可用 TASK ON 和 TASK OFF 指令在“就绪”状态和“等待”状态间对任何循环任务进行状态切换。处于“就绪”状态的循环任务在随后的循环中将保持此状态，同样，在随后的循环中处于“等待”状态的循环任务也将保持状态不变。 TASK ON 和 TASK OFF 指令仅能用于循环任务，不可用于中断任务。

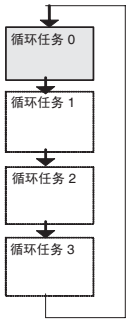
注　在每个循环内必须至少要有一个循环任务处于“就绪”状态，否则，任务错误标志（A295.12）将变为 ON，CPU 将停止运行。

例：循环任务

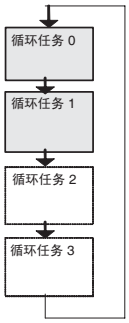


1）运行开始时任务 0 将转为“就绪”状态。

其他任务将保持“禁止”状态。

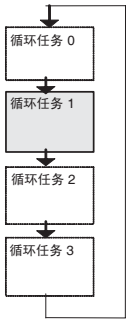


2）如果 A 为 ON，任务 1 将转为“就绪”状态，任务 2 和任务 3 将保持“禁止”状态。



3）如果 D 为 ON，任务 0 将转为“等待”状态。

其他任务将保持它们的当前状态。

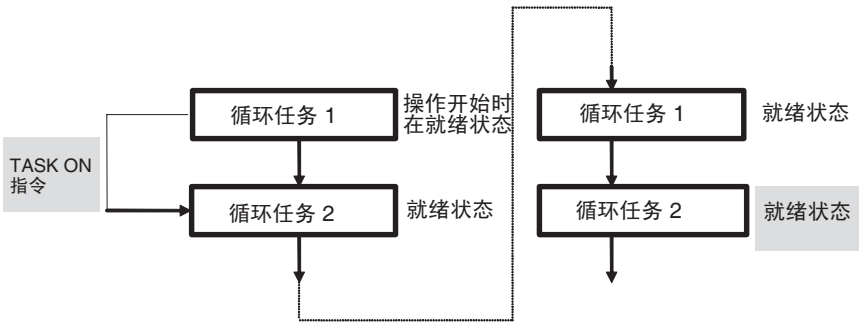


：就绪状态

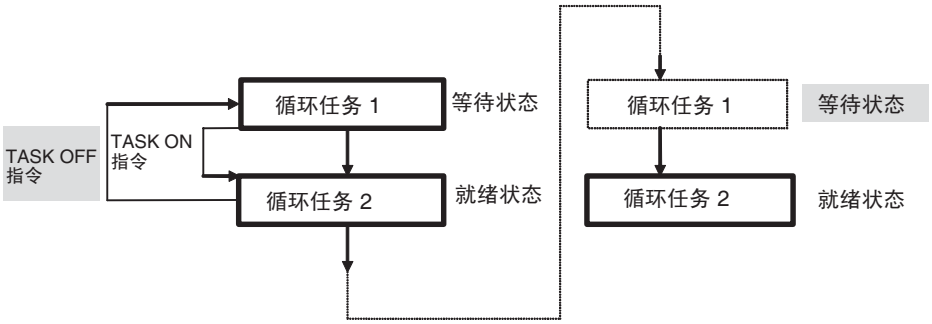
：等待 / 禁止状态

● 任务和执行循环

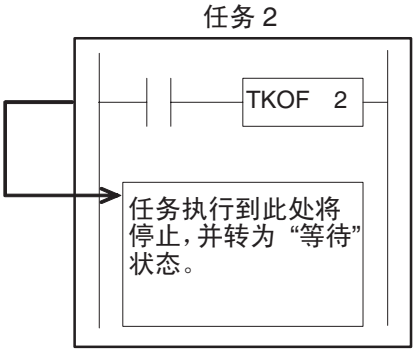
处于“就绪”状态的循环任务（包括附加循环任务）在随后的循环中将保持状态不变。



处于“等待”状态的循环任务在随后的循环中保持状态不变。为了把任务从“等待”状态切换为“就绪”状态，必须使用 TASK ON 指令。



如果在任务中执行 TASK OFF 指令，那么任务将在执行到 TASK OFF 指令时停止执行，并且任务变为“等待”状态。



循环任务编号与执行循环

- 如果任务 m 打开任务 n 且 m > n, 那么任务 n 将在下个循环中进入 “就绪”状态。
  例： 如果任务 5 打开任务 2，那么任务 2 将在下个循环中进入 “就绪”状态。
- 如果任务 m 打开任务 n 且 m < n, 那么任务 n 将在本循环中进入 “就绪”状态。
  例： 如果任务 2 打开任务 5，那么任务 5 将在本循环中进入 “就绪”状态。
- 如果任务 m 置任务 n 为 “等待”状态且 m > n，那么任务 n 将在下个循环中进入 “等待”状态。
  例： 如果任务 5 置任务 2 为 “等待”状态，那么任务 2 将在下个循环中进入 “等待”状态。
- 如果任务 m 置任务 n 为 “等待”状态且 m < n，那么任务 n 将在本循环中进入 “等待”状态。
  例： 如果任务 2 置任务 5 为 “等待”状态，那么任务 5 将在本循环中进入 “等待”状态。

I/O 存储器与任务的关系

- 除变址寄存器（IR）与数据寄存器（DR）以外的 I/O 存储器被任务共享。例如，在循环任务 1 中的位 CIO 10.00 与循环任务 2 中的位 CIO 10.00 表示存储器中的相同位。对除变址寄存器（IR）与数据寄存器（DR）之外的 I/O 存储器访问了一个以上任务时须十分小心，因为随一个任务改变的值可能被其它任务用到。
- 变址寄存器（IR）与数据寄存器（DR）在用法上有两种不同方式：1）独立于任务，或 2）被所有任务共享。
  - 对于独立寄存器而言，循环任务 1 中使用的 IR0 与循环任务 2 中使用的 IR0 是不同的。
  - 对于共享寄存器而言，循环任务 1 中使用的 IR0 与循环任务 2 中使用的 IR0 是相同的。

通过 CX-Programmer 可对寄存器是独立的还是共享的进行设定。

I/O 存储器	与任务的关系
CIO、辅助存储器、数据存储器和除 IR 与 DR 区 以外的所有存储器区 *1	被其他任务共享
变址寄存器（IR）和数据寄存器（DR） *2	被每个任务独立使用

*1 当前 EM 组也被任务共享。因此，如果当前 EM 组编号在循环任务 1 中改变，那么新的当前 EM 组编号对循环任务 2 同样有效。

*2 中断任务启动时不设定 IR 和 DR 值。如果在中断任务中使用 IR 和 DR，它们的值必须在中断任务中由 MOVR /MOVRW（传输至寄存器和定时器 / 计数器当前值至寄存器）指令设定。在中断任务执行完毕后，IR 和 DR 值自动返回为中断之前的值。

5-2-3 中断任务

在一个循环内的任何时候，只要下列条件中的任意一个有效，就执行中断任务。

如果发生中断，不管 CPU 单元是否正在执行循环任务中的指令、刷新 I/O 或执行外设服务，中断任务将在循环的任意点执行。

中断任务类型

可使用以下类型的中断任务。

- 掉电中断任务
  在掉电前立即执行掉电中断任务。
- 定时中断任务
  用户程序的某部分需要定时执行时在程序中使用定时中断任务，例如用于在固定间隔对运行状态的监控。
- I/O 中断任务
  在中断输入单元连接到 CPU 单元的输入为 ON 时执行 I/O 中断任务。
- 外部中断任务
  当一个特殊 I/O 单元或 CPU 总线单元要求中断时，执行外部中断任务。
- 附加循环任务
  附加循环任务是视为循环任务的中断任务。在这种情况下，目的不是与其他中断任务一样进行中断处理，而是增加循环任务的编号。共有 128 个循环任务可被使用。但是，如果使用了附加循环任务，则共有 384 个循环任务可被使用。

● 中断任务列表

任务	任务编号	执行条件	相关设定
掉电中断任务	1	CPU 单元为 OFF	• PLC 设置的 OFF 中断任务
定时中断任务 0 和 1	2 和 3	CPU 单元内部定时器设定的各特定时间	• 使用 MSKS （690）（设定中断）指令设定中断间隔 （0 ～ 9999）。 • 见 PLC 设置中的定时中断时间单位（10 ms/1.0 ms/0.1 ms）。
I/O 中断任务 0 ～ 31*1	100 ～ 131	CPU 机架上的中断输入单元的输入位为 ON	• 使用 MSKS （690）（设定中断）指令为 CPU 机架上的中断输入单元分配输入。
外部中断 0 ～ 255*2	0 ～ 255	当 CPU 机架上的特殊 I/O 单元或 CPU 总线单元请求中断时 *2	无 （一直有效）
附加循环任务 0 ～ 255	0 ～ 255	任务在 “就绪”状态时的各个循环 （必须使用任务控制指令）	无 （一直有效）

*1 中断输入单元必须连接在 CPU 机架上。I/O 中断单元若被连接在其它地方，将不能被用来请求 I/O 中断任务的执行。

*2 特殊的I/O单元或CPU总线单元必须被连接在CPU机架上。单元若被连接在其它地方，将不能用来产生外部中断。

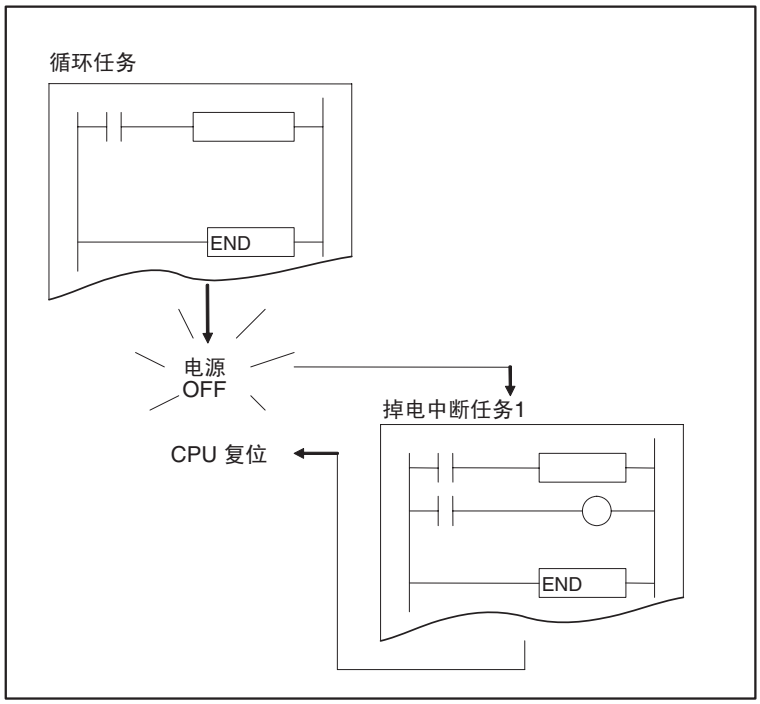
正确使用注意事项

请勿在中断任务中使用 SFC 程序。它们将无法正常执行。但中断任务作为附加循环任务执行时，SFC 程序可用。

ST 程序在中断任务中可用。

掉电中断任务

电源中断时执行该任务。电源中断时，电源单元将供电 5V 到 CPU 单元持续 10 ms，在这段时间内将执行掉电中断任务。（如果使用了 CJ1W-PD022 电源单元，电源将只供电 1 ms，并且不能使用掉电中断任务。）

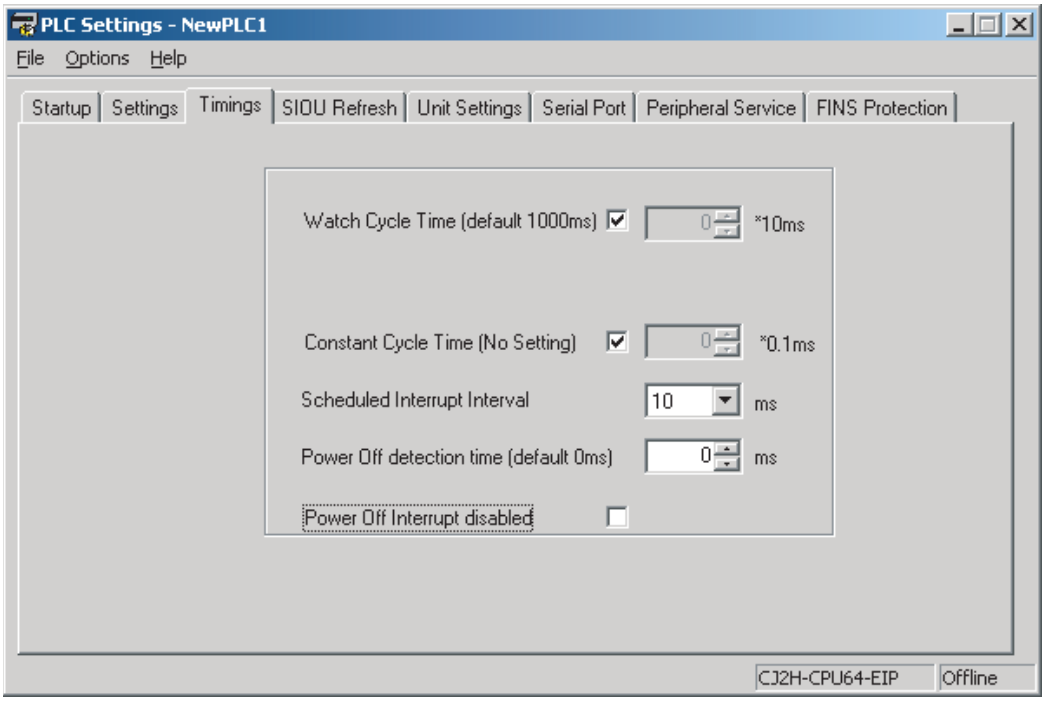


● 掉电中断任务的中断响应时间

掉电中断任务在电源确认中断的 0.1 ms 内执行。

● 掉电中断任务的设定

在 CX-Programmer 的 PLC 设置对话框的 “Timings”标签页中启用掉电中断任务。

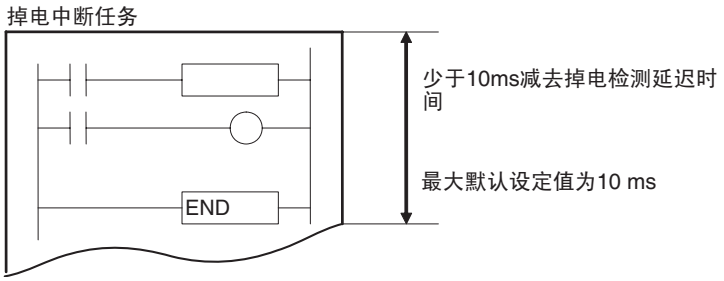


在循环任务开始执行时，掉电中断任务在 PLC 设置中的缺省状态为禁止状态。要启用掉电中断任务，需清除 PLC 设置中的掉电中断禁止选项。

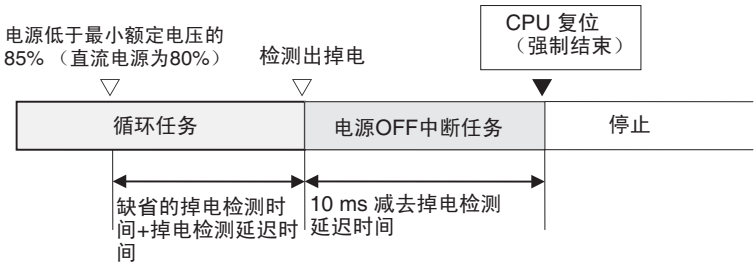
在PLC 的缺省设置中，掉电中断任务在10ms 后停止执行。即掉电中断任务的执行时间必须少于10ms。

如果在 PLC 设置中设定掉电检测延迟时间，那么掉电中断任务将在 10ms 减去在 PLC 设置中设定的掉电检测延迟时间后停止。在这种情况下，掉电中断任务的执行时间必须少于 10ms 减去在 PLC 设置中设定的掉电检测延迟时间。

例：如果在PLC 设置中设定的掉电检测延迟时间为4ms，那么执行时间必须少于10 减去4ms，即6ms。



注　当电源供应降低到最小额定电压的 85% （直流电源为 80%）以下时，认为掉电，掉电中断任务实际执行前所用的时间为缺省的掉电检测时间（交流电源为 10 ～ 25ms，直流电源为 2 ～ 5ms 时）加上在 PLC 设置中设定的掉电检测延迟时间（0 ～ 10ms）。在这段时间内将执行循环任务。



● 掉电中断任务的限制

在线编辑期间不可执行

如果电源在在线编辑期间中断，不能执行掉电中断任务。

不能在掉电中断任务中使用的指令

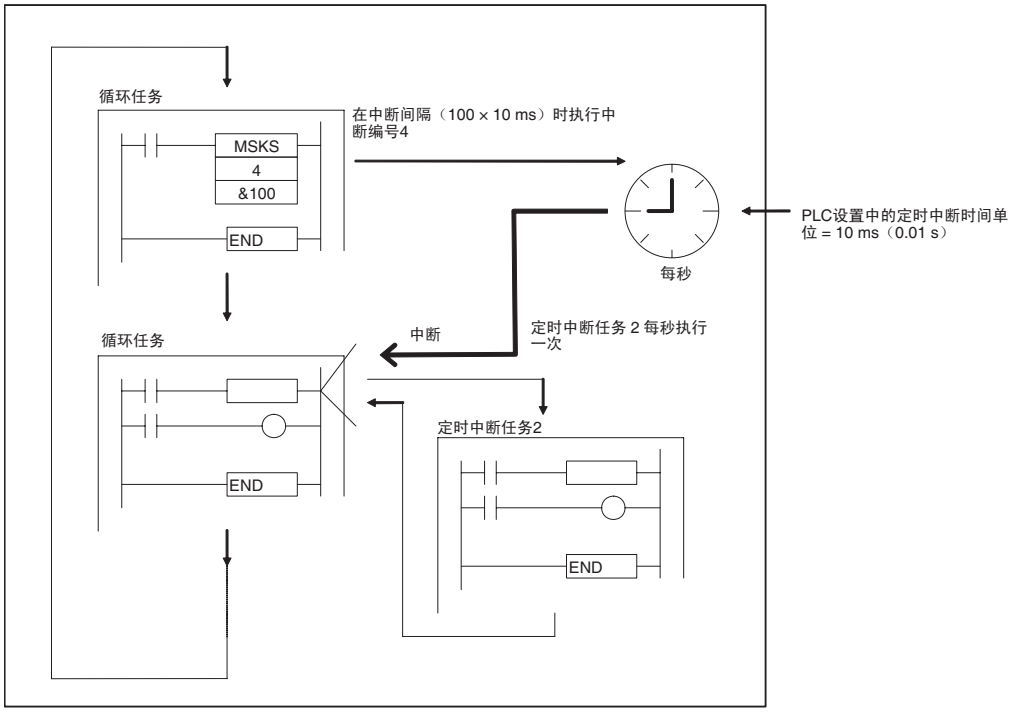
以下指令不能在掉电中断任务中使用：

- 读数据文件：FREAD （700）
- 写数据文件：FWRIT （701）
- 写文本文件：TWRIT （704）
- 网络发送：SEND （090）
- 网络接收：RECV （098）
- 发送命令：CMND （490）
- 网络发送 2：SEND2 （491）
- 网络接收 2：RECV2 （492）
- 发送命令 2：CMND2 （493）
- 发送：TXD （236）
- 接收：RXD （235）
- 协议宏指令：PMCR （260）
- 协议宏指令 2：PMCR2 （264）
- 显式报文发送：EXPLT （270）
- 直接获取属性：EGATR （271）
- 直接设置属性：ESATR （272）

- 直接字读取：ECHRD （273）
- 直接字写入：ECHWR （274）
- 通过串行通信单元传送：TXDU （256）
- 通过串行通信单元接收：RXDU （255）

定时中断任务

通过使用 CPU 单元中的时钟，可在指定的时间间隔执行任务。共有两种任务可被执行（仅中断任务 2 和 3）。



● 中断编号和定时中断任务编号

中断编号	定时中断任务
4	2
5	3

● 执行定时中断任务

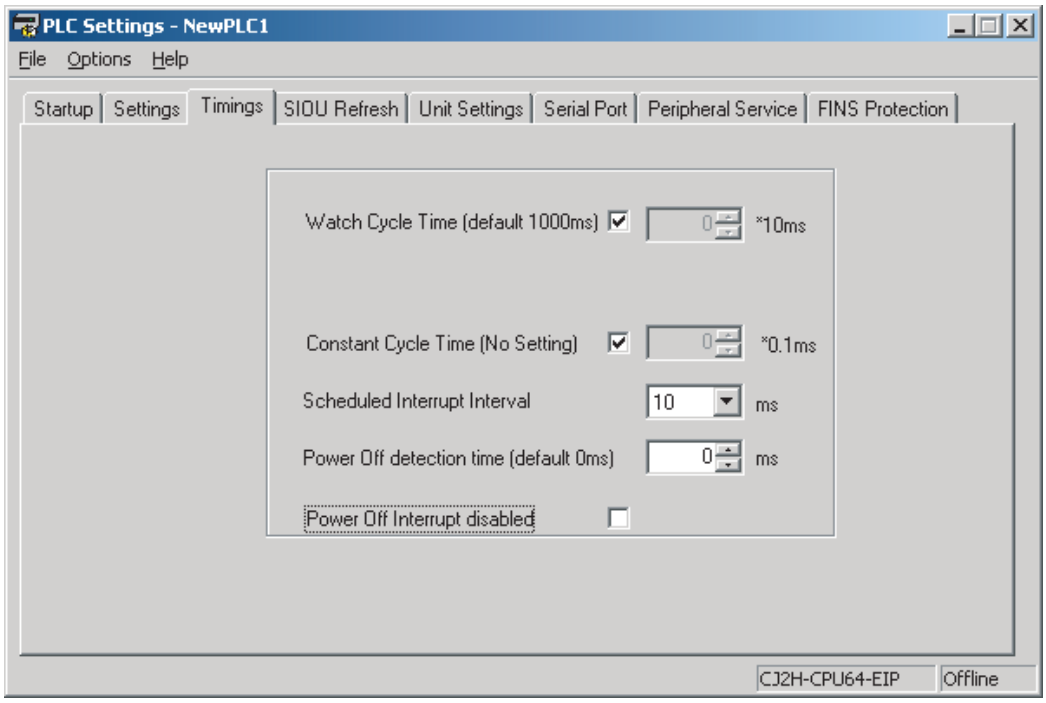
通过使用 MSKS （690）（设定中断）指令来设定定时中断任务的中断时间。在 PLC 设置中可将中断时间单位设为 10 ms、 1.0 ms 或 0.1 ms。

● MSKS （690）和 PLC 设置

在循环任务开始执行时，定时中断任务在 PLC 设置中的缺省状态为禁止状态。执行下面步骤可激活定时中断任务。

1. 在循环任务中执行 MSKS （690）（设定中断）指令并为指定的定时中断设定时间 （循环）。

2. 在 CX-Programmer 的 PLC 设置对话框的 “Timings”标签页中设定定时中断时间单位。



正确使用注意事项

如果缩短中断时间设定并增加定时中断任务的执行频率，需注意由于循环时间增加，将会影响循环任务和附加循环任务的执行时间。

● 定时中断任务的中断响应时间

定时中断任务的中断响应时间为 MSKS （690）指令指定的定时时间经过后直到中断任务实际执行完成所消耗的时间。

定时中断任务的最大中断响应时间为 0.1 ms。

此外，由于实际开始定时中断任务所需的时间间隔，与指定的中断任务时间具有 50 μs 的偏差。

注　如果中断任务已被执行时发生定时中断，首先将完成中断任务的执行，然后再执行定时任务。

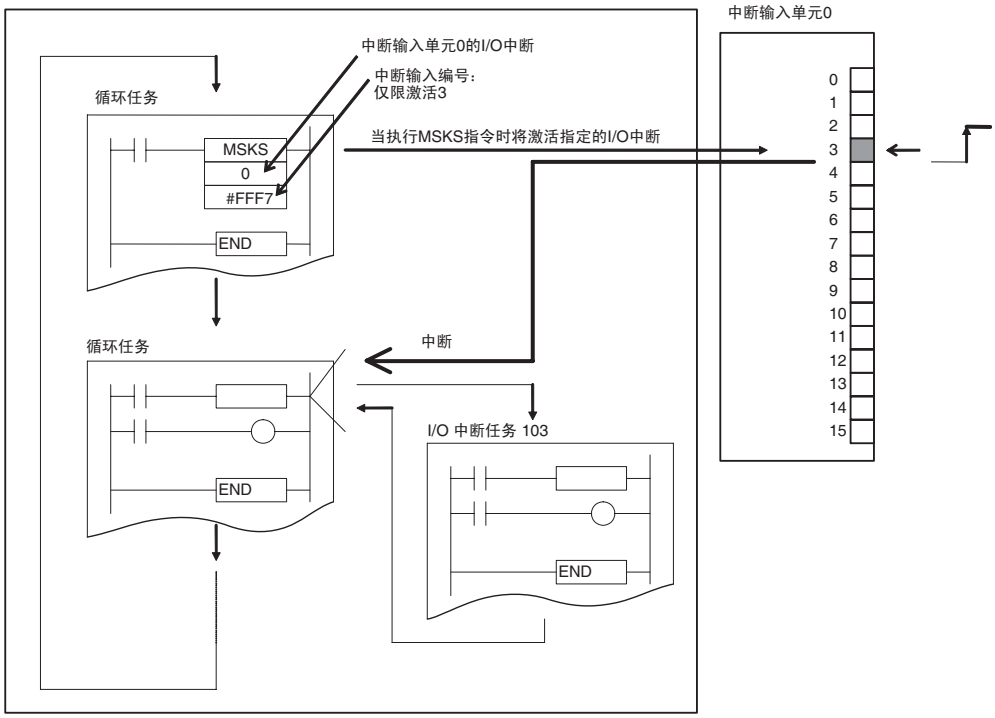
I/O 中断任务

如果中断输入单元的输入为 ON 时，执行 I/O 中断任务 （但是单元必须在 CPU 机架的槽位 0 ～ 3 中）。可创建的最大任务编号为 32 （中断任务编号 100 ～ 131）。

● 执行中断任务

为了激活 I/O 中断，要执行 MSKS （690）（设定中断）指令。

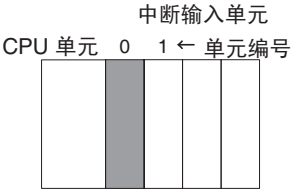
例：下面的例子演示了中断输入单元编号为 0（单元 0 和 1 中最左面的）的中断输入 3 为 ON 时执行 I/O 中断任务 103。



中断输入单元编号、输入编号和 I/O 中断任务编号

中断输入单元编号 *1	输入编号	I/O 中断任务
0	0 ～ 15	100 ～ 115
1	0 ～ 15	116 ～ 131

*1 中断输入单元编号从 CPU 单元开始，以 0 到 1 依次排列。

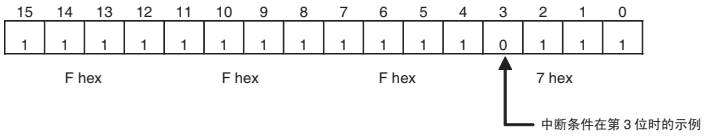


● MSKS （690）

要激活 I/O 中断任务，在循环任务的程序中执行 MSKS （690）（设定中断）指令，并激活指定中断单元的中断编号所指定的中断。

MSKS （690）指令的 S 操作数 （第二个操作数）：
FFF7 hex 中的各位对应中断输入单元的中断输入。中断输入编号 0 ～ 15 对应位 0 ～ 15。

例：



正确使用注意事项

请勿激活不需要的 I/O 中断任务。如果中断输入被干扰触发且无相应的中断任务可执行时，将导致一个致命错误 （任务错误）并使程序终止。

● I/O 中断任务的中断响应时间

I/O 中断任务的中断响应时间是从 CJ1W-INT01 中断输入单元的输入转为 ON 或 OFF 开始直到中断任务被执行的时间。

I/O 中断任务的中断响应时间的长度取决于硬件响应时间和软件响应时间的总时间。

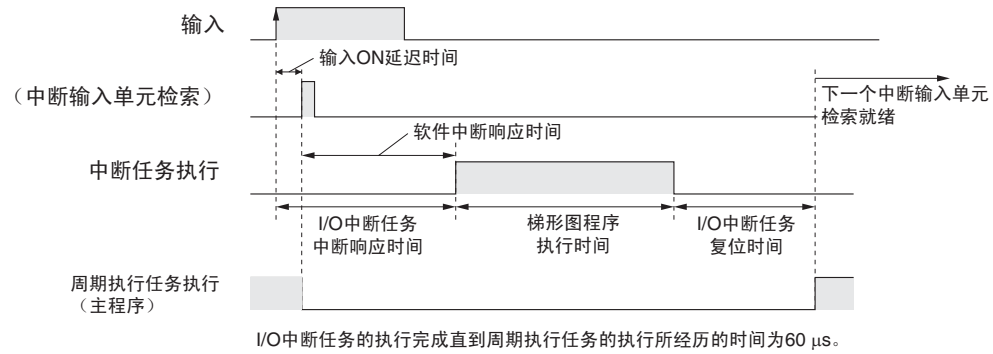
使用中断输入单元

项目	时间
硬件响应	上升沿微分：0.05 ms，下降沿微分：0.5 ms
软件中断响应	30 μs

注　I/O 中断任务可在用户程序、 I/O 刷新、外设服务或监视执行的期间被执行。如果中断发生时，指令正在循环任务中执行，指令的执行将被中断，并将执行 I/O 中断任务。在以上处理操作期间，中断响应时间不会受中断输入单元的输入转为 ON 的影响。

以下图片显示的是 I/O 中断任务的中断响应时间。

I/O 中断响应时间 = ON 响应时间 （输入）+ 软件中断响应时间



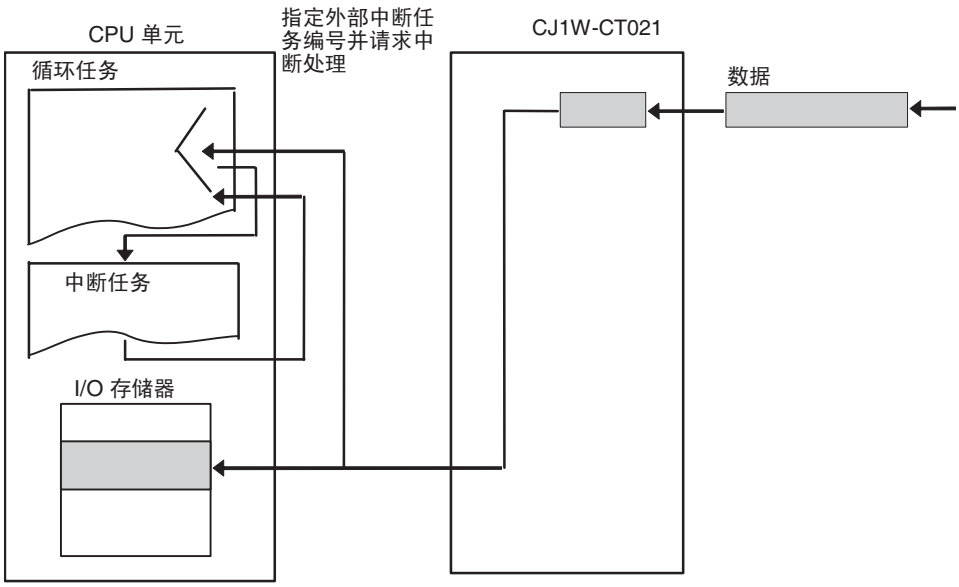
I/O中断任务的执行完成直到周期执行任务的执行所经历的时间为60 μs。

外部中断

当 PLC 的 CPU 单元包含特殊 I/O 单元或 CPU 总线单元时可执行中断处理。除非程序包含某个特定任务编号的外部中断任务，否则不必在 CPU 单元中作设置。特殊 I/O 单元或 CPU 总线单元必须连接到 CPU 机架上以启动外部中断。

必须启用外部中断任务的接收。任务的最大编号为 256 （中断任务编号 0 ～ 255）。

例：CJ1W-CT021 高速计数器单元的外部中断



除 I/O 刷新外，要与 CPU 单元就本单元的 2 点外部控制输入和 32 点输出的状态信息进行通信时，需设置外部控制输入或输出使其触发外部中断任务。

正确使用注意事项

请勿对一个以上的中断任务使用相同的编号。

如果外部中断任务 （0 ～ 255）与掉电任务 （任务 1）、定时中断任务 （任务 2 或 3）或 I/O 中断任务（100 ～ 131）有相同的编号时，中断任务将在两个条件（外部中断条件或其它中断条件）中的任何一个满足时执行。

● 外部中断任务

外部中断任务的中断响应时间取决于请求 CPU 单元外部中断任务的特定单元（特殊 I/O 单元或 CJ 系列 CPU 总线单元）与中断所需的服务类型。详细信息，请参考对使用单元适用的操作手册。

附加循环任务

中断任务与普通循环任务一样在每个循环被执行。附加循环任务 （中断任务编号 0 ～ 255）将在普通循环任务 （循环任务编号 0 ～ 127）执行后，从编号最低的任务开始依次执行。最大附加循环任务的数量为 256 （中断任务编号：0 ～ 255）。

正确使用注意事项

请勿将分配给附加循环任务的任务编号用于中断任务。如果附加循环任务与掉电中断任务、定时中断任务或 I/O 中断任务有相同的编号，将同时作为附加循环任务和中断任务来执行。

注　1　TKON （820）和 TKOF （821）可在附加循环任务中输入和执行，但任务作为中断任务时，不可执行。

2 普通循环任务和附加循环任务的区别如下表所示。

项目	附加循环任务	普通循环任务
启动时激活	不可设定	由 CX-Programmer 设定
任务标志	不支持	支持 （循环任务编号 0 ～ 127 对应任务标志 TK000 ～ TK127）
初始任务执行标志 （A200.15）和任务开始标志 （A200.14）	不支持	支持
变址 （IR）和数据 （DR）寄存器值	任务开始时未定义 （与普通中断任务相同）。每个循环开始的值是未定义的，必须在使用前设置数值，前一循环所设的数值无法读取。	在运行开始时未定义，其值在上一循环内设定并且可读。

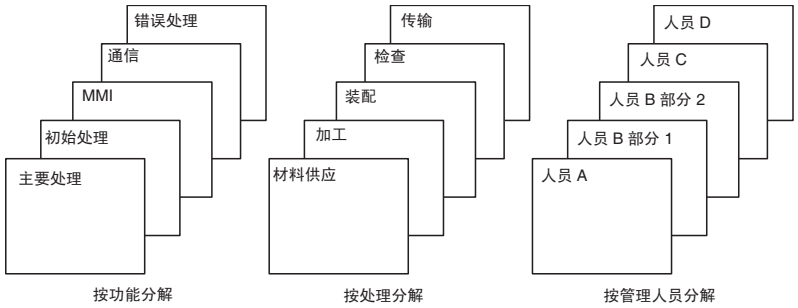
5-2-4 设计任务

步骤

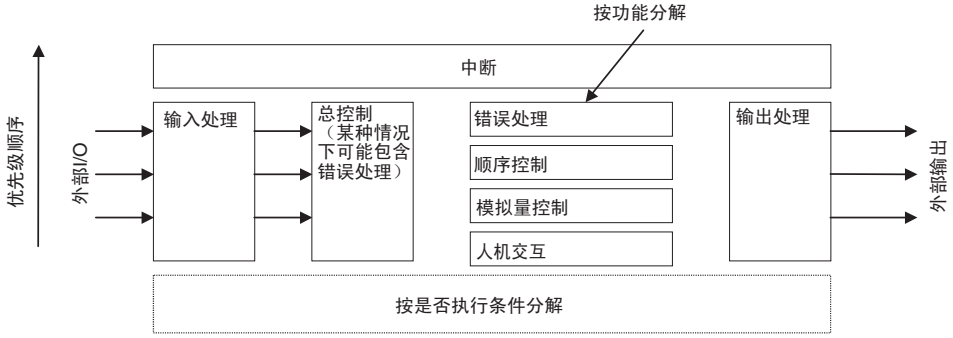
任务设计对于建立一个具有高可靠性和维护简单的系统十分重要。请注意以下几点。

- 可以基于以下几点将程序分解成任务。
  - 根据对状态转换的理解，考虑执行与不执行的特定条件；
  - 考虑有无外部 I/O 的情况；
  - 考虑功能与进行管理的人员；
  - 考虑执行的优先级顺序。把处理分成循环任务和中断任务。将最小编号分配给优先级最高的任务。

例 1：根据功能与进行管理的人员设计任务。

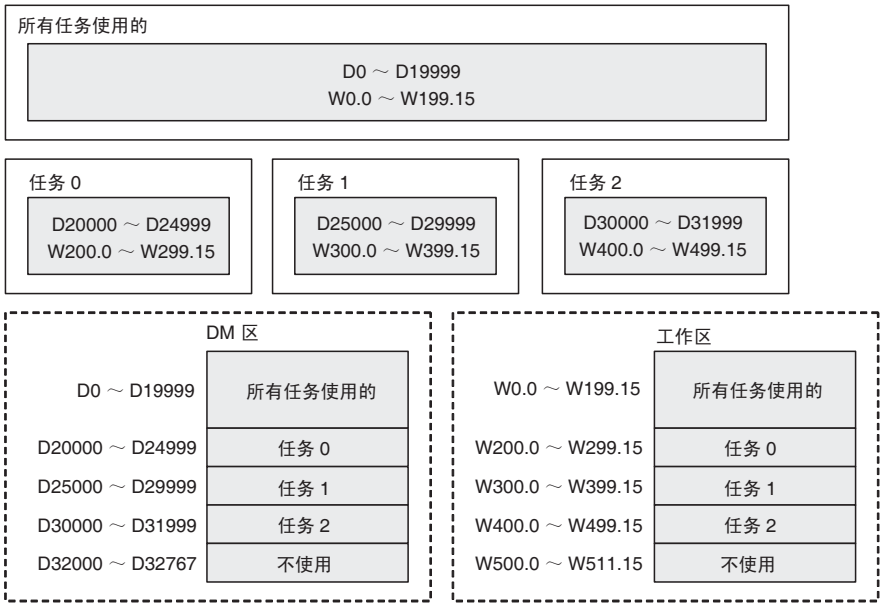


例 2：根据执行的优先级设计任务。
与处理任务相比，将较小的编号分配给控制任务。



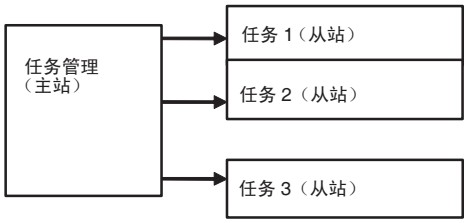
- 以一种能保证独立的方式分解和设计程序，并使任务间交换的数据量为绝对最小。

  组织所有任务使用的 I/O 存储器与仅在单独任务中使用的 I/O 存储器。组织并分配 I/O 存储器的一定范围仅用于每个任务。

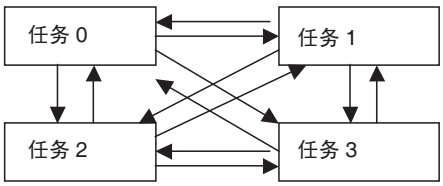


- 使用一个任务来开始和停止其他任务，并控制所有的用户程序。

  良好示例：在任务间定义主站 - 从站关系。



  不良示例：如果以无组织的方式执行开始和停止任务，将导致控制困难。



- 给高优先级的中断任务分配较低的编号。
- 一个处于“就绪”状态下的任务只要它本身或其它任务没有把它切换为“等待”状态，那么它将在随后的循环中执行。如果处理在任务之间被分支的话，必须为其它任务插入一个 TKOF （821）（任务 OFF）指令。
- 在满足执行指令的执行条件下使用初始任务执行标志 （A200.15）或任务开始标志 （A200.14）用于初始化任务。初始任务执行标志将在每个任务的第一次执行期间打开。任务开始标志在每次任务进入“就绪”状态时打开。

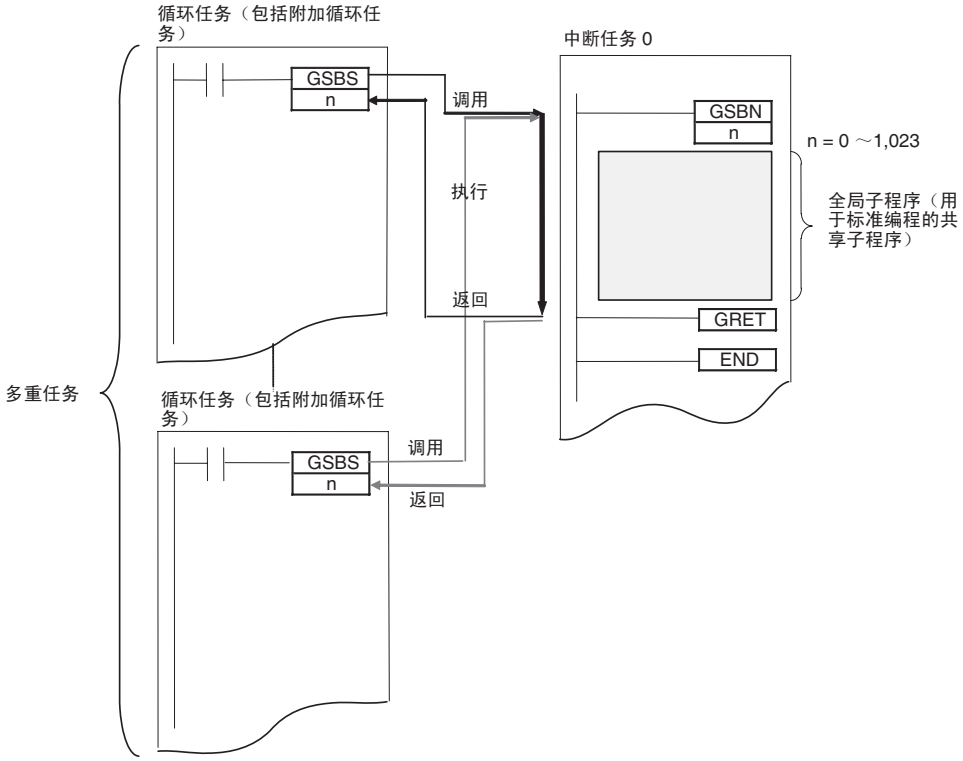
附加信息

全局子程序

根据常规子程序指令，某个任务内的子程序不能被其它任务调用。在编号为 0 的中断任务中可生成全局子程序，并且这些子程序可被循环任务 （包括附加循环任务）调用。

可用 GSBS 指令来调用一个全局子程序。子程序编号必须在 0 ～ 1,023 之间。在编号为 0 的中断任务末尾 （就在 END （001）指令之前）的 GSBN 指令和 GRET 指令之间定义全局子程序。

全局子程序可用来生成一个在任何需要的时候都可调用的标准程序库。



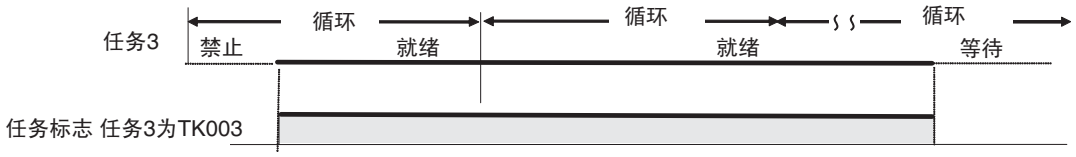
与任务相关的标志

● 循环任务相关标志

下列标志位仅在普通循环任务中工作，不能在附加循环任务中工作。

任务标志 （TK000 ～ TK127）

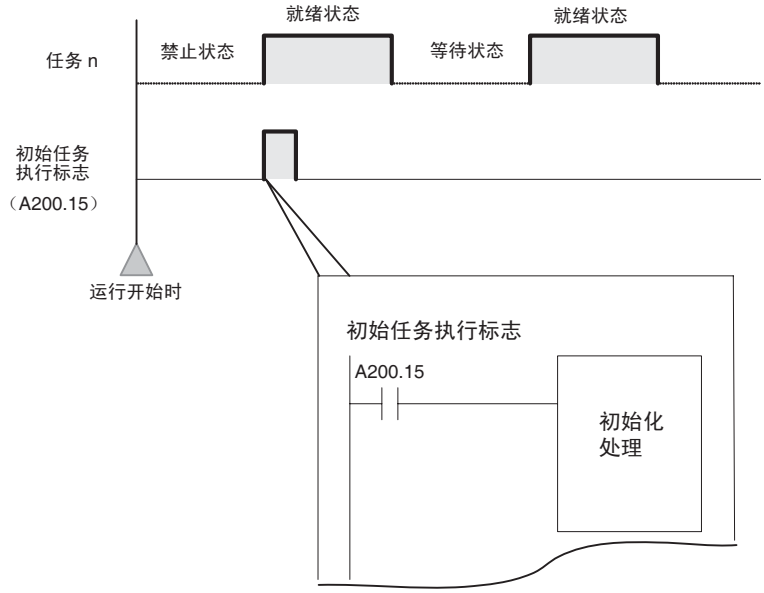
使用该标志检查任务是否正在被执行。任务标志在循环任务处于 “就绪”状态时置 ON，在任务处于“禁止”状态（INI）或“等待”状态（WAIT）时置 OFF。任务编号 00 ～ 127 对应 TK000 ～ TK127。



注　任务标志仅能在循环任务中使用，不能在中断任务中使用。就一个中断任务而言，如果在运行开始后执行一个中断任务， A441.15 将转为 ON，并且需要最大处理时间的那个中断任务编号将以两位十六进制数保存在 A441.00 ～ A441.07 中。

初始任务执行标志 （A200.15）

操作期间仅使用该标志执行一次初始处理。任务第一次执行时转为 ON，任务第一次执行完毕后转为 OFF。

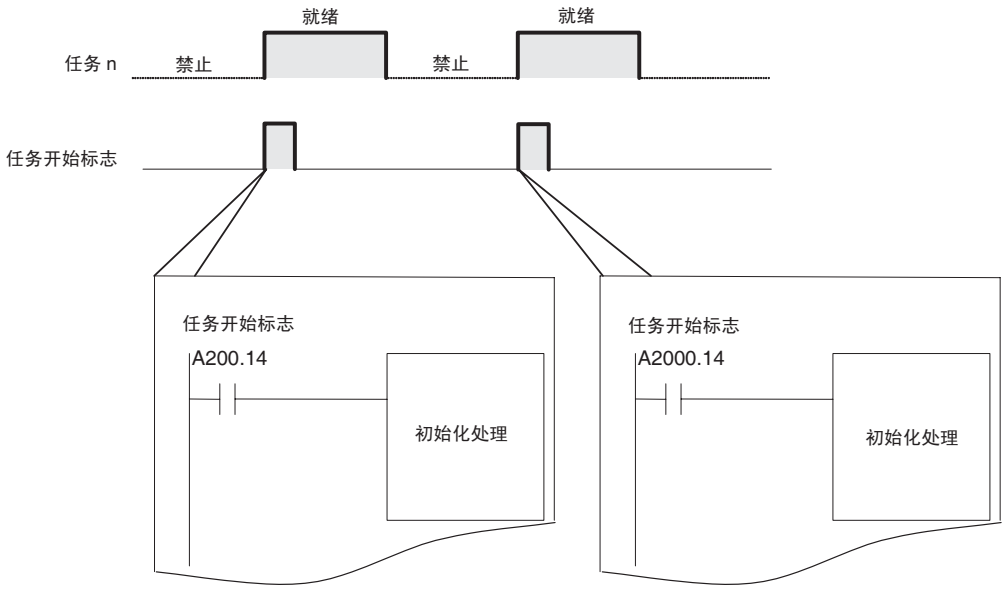


任务第一次执行时初始任务执行标志转为 ON。使用该标志以触发初始化处理。

注　尽管使用 TKON （820）指令可使处于 “等待”状态下的循环任务切换回 “就绪”状态，但这不能被视为初始执行，初始任务执行标志 （20015）不会转为 ON。需使用任务开始标志 （A200.14）。

任务开始标志 （A200.14）

无论何时，只要循环任务从 “等待”状态切换为 “就绪”状态，其任务开始标志位都将转为 ON（然而，初始任务执行标志 （A200.15）只有在每次任务状态从 “等待”状态切换为 “就绪”状态时才转为 ON 一次）。



无论何时，只要任务从“等待”状态变为“就绪”状态，任务开始标志都可用来执行初始化处理，也就是当任务处于 “等待”状态时使用 TRON （820）指令激活任务。

● 中断任务标志和字

最大的中断任务处理时间 （A440）

最大的中断任务处理时间是按二进制数据格式以 0.1ms 为单位存储的，并在运行开始时清除。

最大处理时间的中断任务 （A441）

最大处理时间的中断任务编号以二进制数据格式保存。这里，8000 ～ 80FF Hex 对应任务编号 00 ～ FF Hex。

在运行开始后的第一个中断发生时，A441.15 转为 ON。随后，中断任务最大处理时间将以两位十六进制数保存在最低位中，并且在开始运行时清除。

重复刷新错误标志 （非致命错误）（A402.13）

如果在PLC 设置中启用了重复刷新错误检测，当发生重复刷新错误时，重复刷新错误标志将转为ON。

重复刷新错误原因标志 （A426.15） / 产生重复刷新错误的任务编号 （A426.00 ～ 426.11）

A402.13	中断任务错误描述	A426.15	A426.00 ～ 426.11
中断任务错误（如果在 PLC 设置中启用了中断任务错误检测功能）	如果在 PLC 设置中启用了重复刷新错误检测，在同一特殊 I/O 单元中满足以下条件时，重复刷新错误标志将转为 ON。 • 在中断任务中执行的 IORF、 FIORF、 IORD 或 IOWR 指令与在循环任务中执行的 IORF、 FIORF、 IORD 或 IOWR 指令之间有冲突。 • 在中断任务中执行的 IORF、 FIORF、 IORD 或 IOWR 指令与 CPU 单元的 I/O 刷新 （结束刷新）之间有冲突。 如果在 PLC 设置中启用了特殊 I/O 单元的循环刷新，在同一特殊 I/O 单元中执行 IORF、 FIORF、 IORD 或 IOWR 指令时，将发生重复刷新并产生中断任务错误。	1 （ON）	正在刷新的特殊 I/O 单元的单元编号将以 12 位二进制数据保存 （单元编号 0 ～ 95：000 ～ 05F Hex）。

程序停止时的任务编号 （A294）

由于程序错误而使任务停止，此时当前任务编号和任务类型将保存如下：

类型	A294
中断任务	8000 ～ 80FF Hex （对应中断任务编号 0 ～ 255）
循环任务	0000 ～ 007F Hex （对应任务编号 0 ～ 127）

● 与所有任务相关的标志

任务错误标志 （A295.12）

如果发生以下任务错误之一时，任务错误标志转为 ON。

- 在一个循环内没有循环任务。
- 分配给一个循环任务（包括附加循环任务）的程序不存在（在使用 CX-Programmer 时这种情况不会发生）。
- 处于激活状态的中断任务无指派程序 （包括附加循环任务）。

程序停止时的任务编号 （A294）

由于程序错误而使任务停止，此时当前任务编号和任务类型将保存如下：

类型	A294
循环任务	0000 ～ 007F Hex （对应任务编号 0 ～ 127）
中断任务 （包括附加循环任务）	8000 ～ 80FF Hex （对应中断任务编号 0 ～ 255）

这个信息使得用户可以很容易地判断出在何处发生了致命错误，在致命错误消除后，信息将被清除。任务运行终止处的程序地址保存在 A298 （程序地址的最低位）和 A299 （程序地址的最高位）中。

注意事项

● 指令使用限制

必须在同一任务中使用的指令

以下指令必须在同一任务中使用。

助记符	指令
JMP/JME	跳转 / 跳转结束
CJP/JME	条件跳转 / 跳转结束
CJPN/JME	条件不满足跳转 / 条件跳转结束
JMP0/JME0	多重跳转 / 跳转结束
FOR/NEXT	FOR/NEXT
IL/ILC	联锁 / 联锁清除
SBS/SBN/RET	子程序调用 / 子程序进入 / 子程序返回 （如果使用全局子程序指令 （GSBS （750）、 GSBN （751）和 GRET （752）），也可从其他任务调用子程序。）
MCRO/SBN/RET	宏指令 / 子程序进入 / 子程序返回
BPRG/BEND	程序块开始 / 程序块结束
STEP/STEP	定义步进程序

中断任务中不允许使用的指令

下列指令不允许在中断任务中出现，任何企图在中断任务中使用下列指令都将导致错误标志（P_ER）变为 ON，且指令停止执行。但如果中断任务作为附加循环任务使用时，则可以使用下列指令。

助记符	指令
TKON	任务开启
TKOF	任务关闭
STEP	定义步进
SNXT	定义下一步
STUP	更改串行端口设置
DI	使中断禁止
EI	使中断有效
SFCON、SFCOFF、 SFCPR、SFCPRN	SFC 任务控制指令

- 在中断任务中，下列指令的操作结果是不确定的：100MS 定时器：TIM 和 TIMX（550）、10MS 定时器：TIMH（015）和 TIMHX（551）、1MS 定时器：TMHH（540）和 TMHHX（552）、0.1MS 定时器：TIMU（541）和 TIMUX（556）、0.01MS 定时器：TIMUH（544）和 TIMUHX（557）、累计定时器：TTIM（087）和 TTIMX（555）、多输出定时器：MTIM（543）和 MTIMX（544）、长时间定时器：TIML（542）和 TIMLX（553）、定时器等待：TIMW（813）和 TIMWX（816）、高速定时器等待：TMHW（815）和 TMHWX（817）、PID 控制：PID（190）、错误点检测：FPD（269）、更改串行端口设置：STUP （237）。

- 下列指令不能在掉电中断任务中使用（即使使用也将不执行，但错误标志（P_ER）不会置 ON）：读数据文件：FREAD（700）、写数据文件：FWRIT（701）、写文本文件：TWRIT（704））、网络发送：SEND（090）、网络接收：RECV（098）、发送命令：CMND（490）、发送：TXD（236）、接收：RXD（235）、协议宏指令：PMCR（260）、显式报文发送：EXPLT（270）、直接获取属性：EGATR（271）、直接设置属性：ESATR（272）、直接字读取：ECHRD（273）、直接字写入：ECHWR（274）、通过串行通信单元传送：TXDU（256）和通过串行通信单元接收：RXDU（255）。



附加信息

定时器操作与任务的关系

定时器指令操作如下：

- 定时器编号 0 ～ 2047
  即使包含定时器的任务在“等待”状态或返回“就绪”状态，定时器当前值将继续刷新。任务返回“就绪”状态时，如果在包含有效 TIM 指令的任务变为“等待”状态，并且当前值为 0 时执行 TIM 指令，那么完成标志位将转为 ON（定时器完成标志仅在指令执行时更新）。当前值将继续刷新。
- 定时器编号 2048 ～ 4095
  任务处于“等待”状态时，任务开始时的定时器当前值将不会被刷新。但是，在 PLC 特性中有一项设定可用于对定时器 0 ～ 2047 和定时器 2048 ～ 4095 启用相同操作。

条件标志与任务的关系

在每个任务执行前，所有的条件标志将复位。因此，在任务 2 中不能读取任务 1 结束时的条件标志状态。然而，CCS（282）和 CCL（283）指令可以从程序的其它部分读条件标志状态，即从其它任务中读取条件标志状态。

● 中断任务注意事项

中断任务优先级

在掉电中断任务允许执行时，其它中断任务都将停止执行。在掉电中断任务执行完毕后，CPU 复位，但是被中止的中断任务将不再执行。

以下部分描述了其他中断任务的优先级。

多个中断申请同时出现

除掉电中断任务外，无论什么时候，如果多个中断申请同时出现，那么中断任务将按下面的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

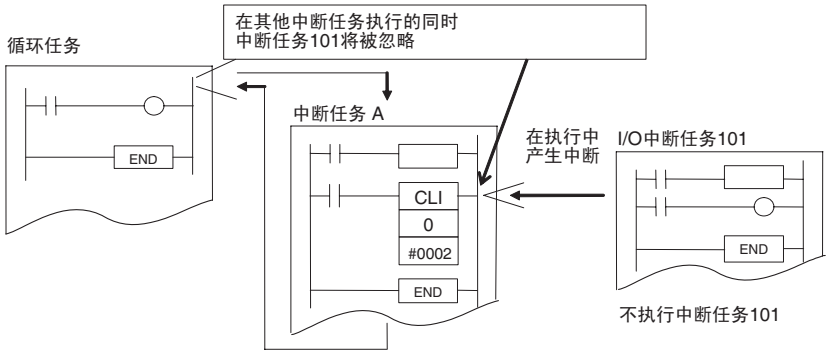
在各种类型的中断任务内，如果产生多个中断任务，那么将从编号最低的中断任务开始依次执行。

中断任务执行过程中的中断

如果当一个中断任务正在执行时发生另一个中断，那么这个中断的相应任务将在原先的中断任务执行完毕后才被执行。

附加信息

对于 CS 系列 CPU 而言，当一个中断任务正在执行时发生一个特殊 I/O 中断申请，如果你不想保存和执行这个特殊 I/O 中断任务，可在正在执行的中断任务中执行 CLI （清除中断）指令清除内部自动保存的中断编号。但不可取消定时中断和外部中断。



附加信息

对于所有中断任务而言，仅有一个中断被存储器记录，并且已经在执行的中断不被存储器记录。由于定时中断的优先级最低并且在每个时刻仅能记录一个中断，因此定时中断任务很有可能被跳过而不执行。

● 应用注意事项

对特殊 I/O 单元执行 IORF （097）、 FIORF （225）、 IORD （222）或 IOWR （223）指令。

如使用了特殊 I/O 单元并且中断任务执行了 IORF （097）、 FIORF （225）、 IORD （222）或 IOWR （223）指令，必须在 PLC 设置中禁止特殊 I/O 单元的循环刷新。

如通过循环刷新或 I/O 刷新指令对特殊 I/O 单元进行了刷新，试图在中断任务中用 IORF （097）或 FIORF （225）指令刷新同一特殊 I/O 单元时，或者用 IORD （222）或 IOWR （223）指令对同一特殊 I/O 单元读 / 写数据时，将发生重复刷新错误。在这种情况下，IORF（097）、FIORF（225）、IORD（222）或 IOWR （223）指令不被执行，但错误标志 （P_ER）将转为 ON。循环刷新将正常执行。

如果在 PLC 设置中，启用了重复刷新错误检测，当发生重复刷新错误错误时，A402.13 （重复刷新错误标志）将转为 ON，并且特殊 I/O 单元的单元编号将保存到 A426 （重复刷新错误任务编号）中去。

● 在循环任务的特殊处理中禁止中断

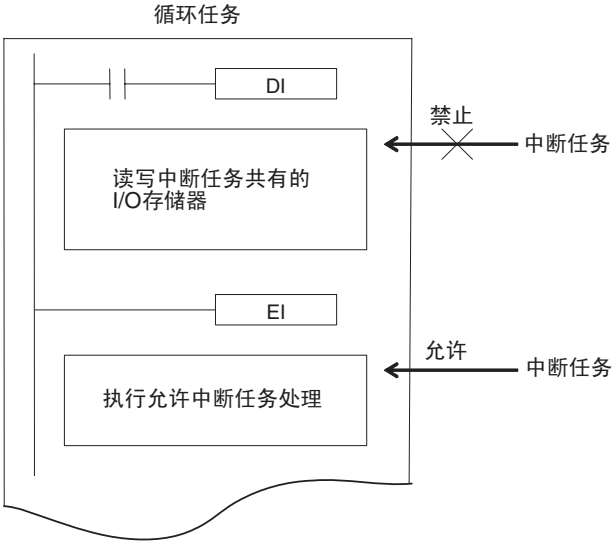
在下列情况下，处理中止并执行中断任务。

- 某个指令执行时
- 基本 I/O 单元或 CPU 总线单元刷新时
- 外设服务时

确保循环和中断任务间的数据一致性

如果循环任务和中断任务对同一 I/O 存储地址进行读写，数据可能不一致。在循环任务指令存取存储器过程中，执行下列步骤使中断禁止。

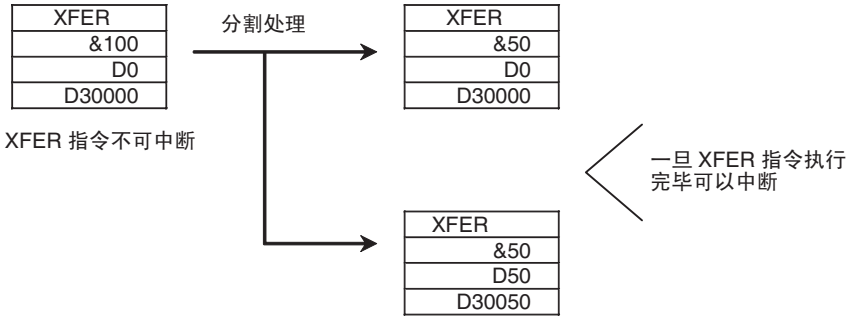
- 在用一个循环任务指令进行读写前，立刻使用 DI （693）（中断禁止）指令时中断任务不能执行。
- 在处理完成后立刻使用 EI （694）（中断允许）指令，使中断任务有效。



在要求响应接收与处理的指令（如网络指令或串行通信指令）执行过程中，即使用了 DI（693）和 EI（694）指令使中断任务禁止，仍然有可能发生数据一致性的问题。

附加信息

不能中止位计数 （BCNT）、块置位 （BSET）和块传送 （XFER）指令的执行去执行中断任务，即推迟中断响应，在中断任务执行前完成这些指令的执行。为防止这种情况，必须把这些指令的数据处理分成几个部分，下面演示了 XFER 的分配。

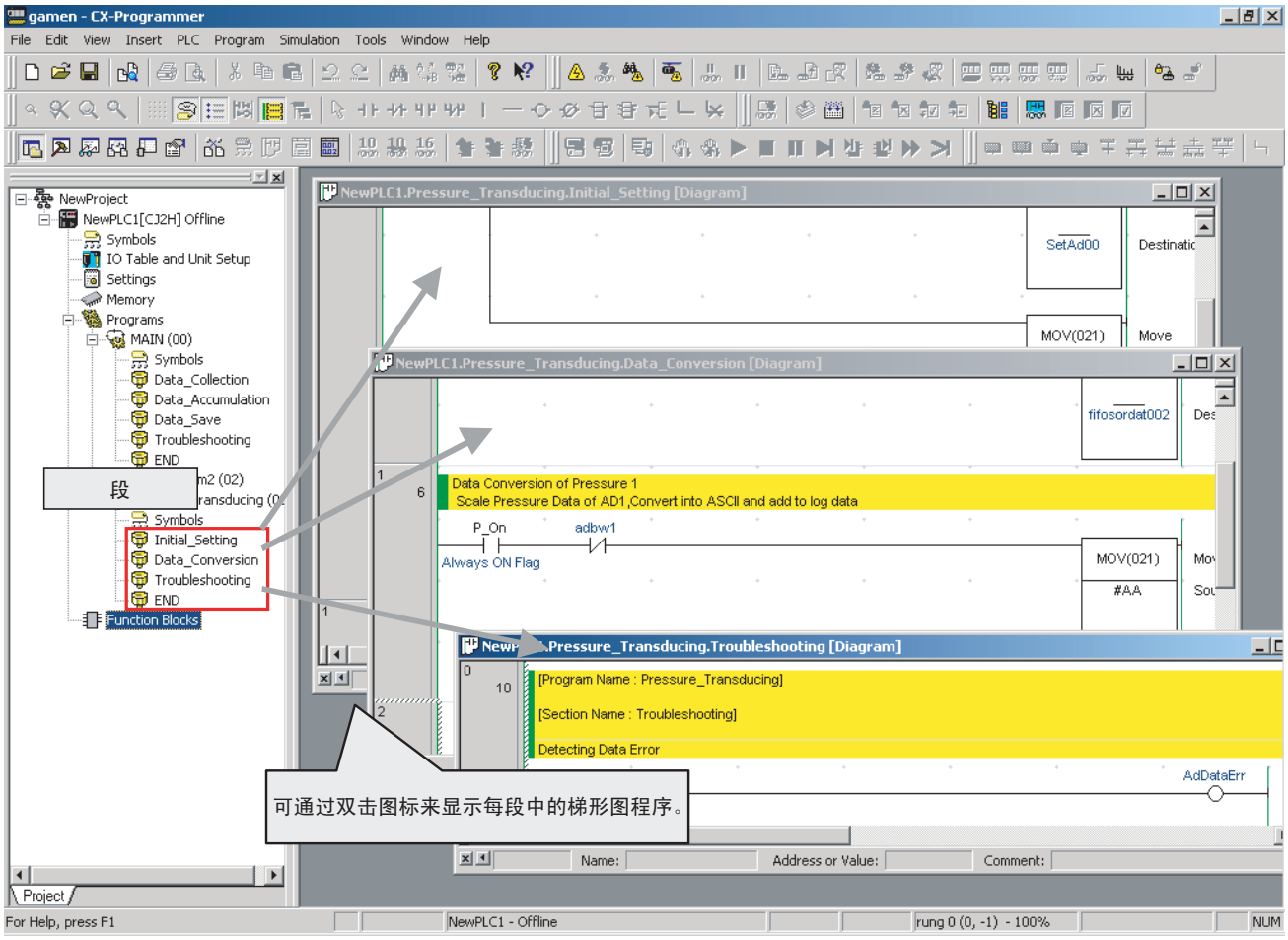


5-3 段

5-3-1 段概述

程序可创建并显示为段，并通过 CX-Programmer 将程序分配到功能单元。任务中的任何程序都可分为段。段的使用改进了程序的易读性并简化了编辑。

- 更明确的处理
  可对段起任何名称来表示在该段中执行的处理或控制。
- 简单的编辑
  通过使用鼠标拖动和点击段来改变段的顺序，并且还可对其进行复制和粘帖。
- 无限制的使用
  每个程序可创建的段的数量没有限制。



● 段的数量

可使用的段的数量没有限制。与以前一样，也可以创建仅有一个段的程序。

● 段执行的顺序

段以出现在工程树（从顶部到低部）的顺序依次执行。程序地址以任务执行的顺序自动连续下去。每段中的条编号从 0 开始。

● 段名称

段名称为“Section □”，带一串从 1 开始自动分配的连续数作为后缀。可通过用户自定义来更改名称。

● 程序分段的优点

将程序分为段有以下的优点。

- 一次只能从 CPU 单元上传程序的一个段。如果已经上传了程序的一个段，开始在线编辑所需的时间将会缩短。对于在线编辑，一次只能下载程序的一个段。此外，无法进行以一个段为单位的下载。
- 移动段可以用来改变总程序执行的顺序，可将段移动到其他程序中，并且在工程树上可以对其进行复制。

这实现了与总程序相比，更小的标准化编程模块。

● 与段相关的文件存储器文件

段包含在 CX-Programmer 项目文件 （.CXP）中。 CPU 单元的段名称、段注释和程序注释保存在程序索引文件 （PROGRAMS.IDX）中。



附加信息

任务用来控制任务程序是否被执行。另一方面，在此之下的用户自定义程序中，创建分立的段。段的创建主要用于使程序易于理解，见以下的应用示例。

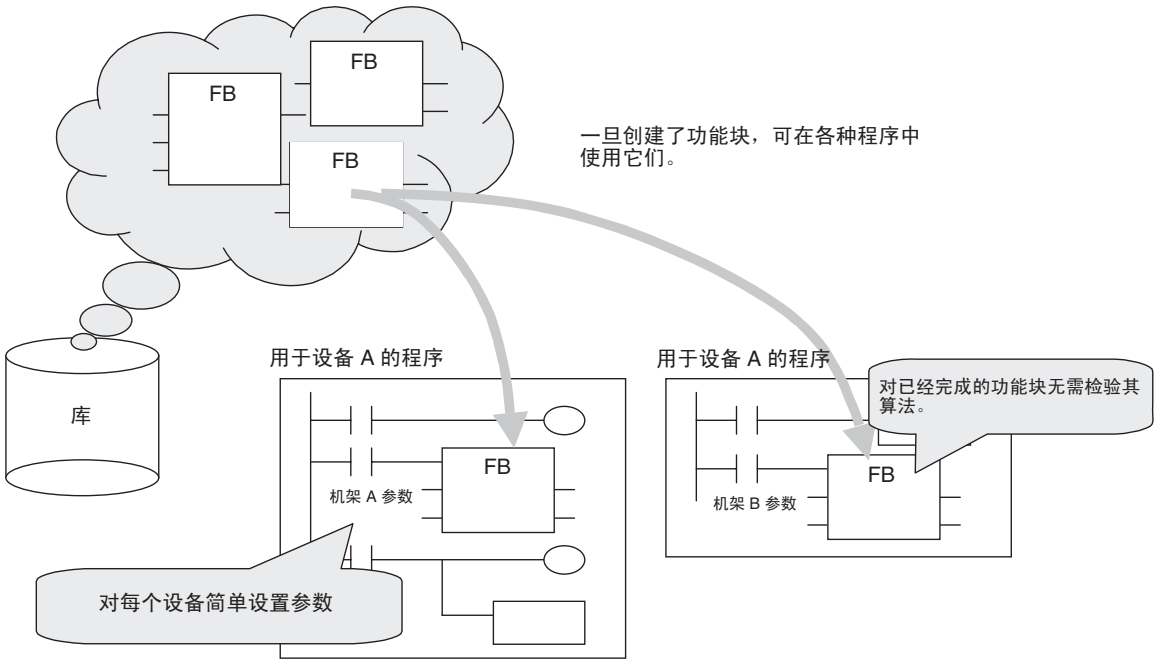
- 不管任务是否被执行，需要进行相同处理时，初始处理和主要处理可分配到不同的段中。
- 普通程序区和子程序可分配到段中。

5-4 功能块

5-4-1 功能块

功能块（FB）是一种基本的程序元素，包含用于预先定义的标准处理功能的程序。一旦你创建了功能块并将其保存在库中，可通过选择系统所需的功能块、将其放置到程序中以及设定 I/O 参数来简单地对其使用。优秀的功能块可极大地改进程序发展的效率。

OMRON 功能块符合 IEC 61131-3。



● 功能块特性

- 重复使用性
  功能块可保存在库中以使其能被简单地重复使用。一旦程序员理解了功能块，可通过设置参数简单地对其进行使用。这极大地改善了编程效率。
- 减少调试工作
  重复使用已经过调试的高质量功能块消除了再次调试的必要，因此能够减少调试工作。
- 数据保护
  功能块内部的变量不能从外部被直接访问，因此数据是被保护的 （数据不会被无意地改变）。

5-4-2 功能块特性

● 功能块的优点

功能块允许复杂的编程单元被简单地重复使用。一旦在功能块中创建了标准编程并保存在文件中，可通过在程序中放置功能块并设定功能块的 I/O 参数来简单地对其进行重复使用。在创建 / 调试程序时，重复使用已存在的功能块将有效节省时间、减少代码错误，并使程序易于理解。

● 功能块的输入地址

功能块的地址通常是通过变量而非实际的物理地址输入。程序员使用这些变量来设定参数（地址或值）。每次功能块放置到程序中时，每个变量使用的地址由 CX-Programmer 自动分配。

● 嵌套

功能块可在另一个功能块中嵌套。共有 8 个嵌套等级可用，包括首个功能块的调用。

● Smart FB 库

Smart FB 库是用于改进 OMRON PLC 单元与 FA 组件之间操作的一套功能块。无需创建梯形图程序来使用基本单元和 FA 组件功能。

● FB 定义的在线编辑

运行期间可更改 FB 定义，因此 FB 定义可在调试时被快速编辑。此外，即使在必须 24 小时 / 每天运行的环境下也可放心地使用 FB。

● 保护 FB 定义

通过对在项目文件中分配的功能块定义设定密码，根据目的来保护定义，可以防止未经许可的操作、编辑或程序的误用。

● FB 生成功能

可将现有的 PLC 编程转换为 FB 来进行重复使用。

5-4-3 功能块规格

项目		规格	
定义和创建功能块	功能块定义数	2,048 最大	
	功能块名称	64 字符最大	
	变量	变量名称	15,000 字符最大
		变量类型	输入变量 （输入）、输出变量 （输出）、输入 - 输出变量 （输入输出）、内部变量 （内部）和外部变量 （外部）
		在功能块中使用的变量数 （不包括内部变量、外部变量、EN 和 EN0）	每个功能块定义的最大变量数 • 输入 - 输出变量：16 最大 • 输入变量 + 输入 - 输出变量：64 最大 • 输出变量 + 输入 - 输出变量：64 最大
		变量使用的地址分配	自动分配 （分配范围由用户设定）
		实际地址规格	支持
		数组规格	支持 （仅在一个空间排列，并且只能用于内部变量和输入 - 输出变量）
	语言	功能块可用梯形图编程语言或结构文本来创建。	
创建示例	示例数	2,048 最大	
	示例名称	15,000 字符最大	
功能块作为文件保存	项目文件	项目文件 （.cxp/cxt）包含功能块定义和示例。	
	程序 / 网络符号 （变量）文件	文件存储器程序文件 （*.obj）包含功能块定义和示例。	
	功能块库文件	可将每个功能块定义以单个文件 （.cxf）保存，用于在其他项目中重复使用。	

附加信息

关于功能块的规格、表示方法和输入步骤，请参考 CX-Programmer 操作手册：功能块与结构文本（样本编号 W447）。

5-5 符号（变量）

5-5-1 概述

符号（变量）

● 符号（变量）表

可通过将字符串注册为符号（变量）来指定 I/O 存储器区的地址或常数。在 CX-Programmer 的符号表中注册符号。带符号的编程允许用名称来编程而不必知道实际地址。

符号（变量）表与其他参数一起保存在 CX-Programmer 的项目文件（.CXP）中，如用户程序和 I/O 表。

● 使用符号（变量）的条件

使用符号（变量）为必须或可选取决于编程语言以及符号是在功能块的内部还是外部使用，如下表中所给出的。

程序要素	编程语言	指定符号（变量）I
在功能块外部编程	梯形图	可选
	ST	必须
	SFC	可选
在功能块内部编程（在功能块内部，符号被称为“变量”）。	梯形图	必须
	ST	

5-5-2 符号（变量）类型

以下的符号类型可被使用。

程序符号（变量）

以下三种类型的符号（变量）可在程序中使用。

● 符号（变量）类型

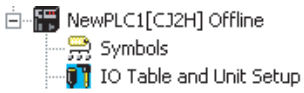
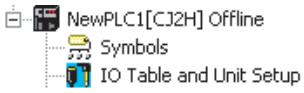
- 全局符号
  在 CPU 单元中可从任何任务访问全局符号。
- 本地符号
  只能从一个任务访问本地符号。
- 网络符号
  可使用标签。从外部设备来访问在 CPU 单元中已定义网络符号的本地 CPU 单元的 I/O 存储器。网络符号也可以在本地 CPU 单元中使用。

● 地址分配

可根据以下因素将地址分配到程序中使用的符号（变量）。

- 用户规格

- 使用 CX-Programmer 自动分配
  在 CX-Programmer 的 PLC 菜单中选择“存储器分配”-“自动地址分配”，设定自动分配使用的存储器区。

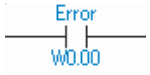
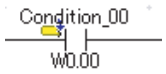
符号（变量）类型	名称	CX-Programmer 工程树	范围			地址和 I/O 注释（不带符号（变量）名称）
			使用网络符号（变量）访问	通过其他任务访问	通过本地任务访问	
程序符号	全局符号	PLC 树 	不支持	支持	支持	支持
	本地符号	程序树 		不支持	支持	不支持
	网络符号	在 PLC 树中的全局变量表 	支持	支持	支持	不支持

注　“全局”和“本地”表示符号（变量）应用的范围。存储器地址的范围不可设定。因此，在以下情况中将发生一个警告而不是错误，并可传送用户程序。

- 两个不同的本地符号使用了相同的地址。
- 全局符号与本地符号使用了相同的地址。

附加信息

在 CX-Programmer 程序中，可通过以下字符颜色和表示符号（变量）的图标来辨别全局符号、本地符号和网络符号。

符号（变量）类型	显示颜色	示例（默认颜色）
全局符号	黑（固定）	
本地符号	蓝（默认） 选择“工具”-“选项”，显示“外观”标签页，然后选择本地符号项目来改变颜色。	
网络符号（变量）	黑（固定） 表示网络符号的图标（ ）将在符号名称下面显示 。	

功能块中的变量

只能通过功能块的算法访问变量。

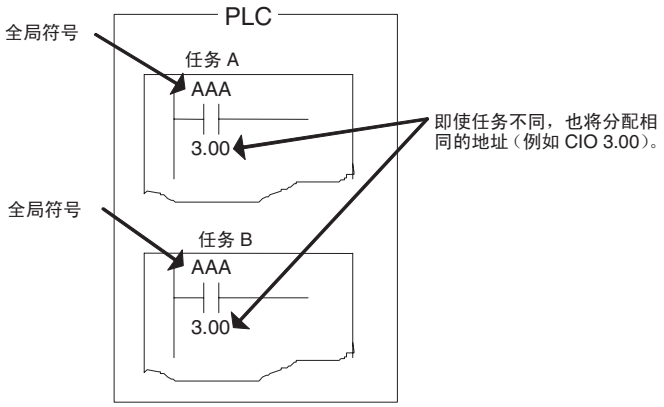
名称	CX-programmer 工程树	范围		符号和 I/O 注释 （不带符号名称）
		使用网络符号（变量）访问	从功能块外部访问	
变量 （功能块中使用的符号）	功能块中的定义 	不支持 （不能对外部变量注册网络符号）	不支持	不支持

一些变量的规格与在功能块外部使用的符号的规格不同。通过在 CX-Programmer 的菜单中选择“存储器分配”-“功能块 /SFC 存储器”，设定用于变量的区。

5-5-3 全局符号 （变量）

全局符号 （变量）支持目标 CPU 单元中所有的任务。例如，名称为“AAA”的符号在目标 CPU 单元中的所有任务里有相同地址。

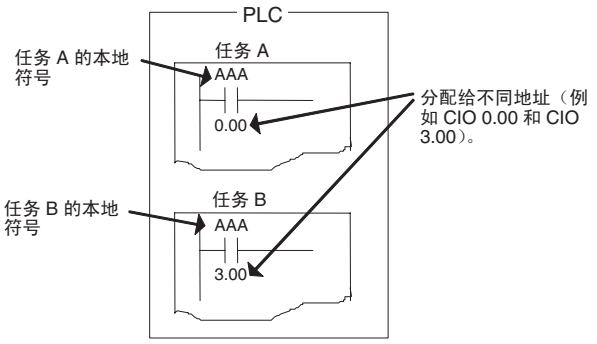
例如：如果名称为“AAA”的符号设为全局符号，即使任务不同，也将分配相同的地址 （例如， CIO 3.00）。



5-5-4 本地符号 （变量）

只能从本地符号 （变量）定义过的任务访问它们。每个本地符号仅在一个任务中被支持。每个任务的本地符号分别指定，因此不同任务的相同符号将作为不同符号进行处理。

例如，名称为“AAA”的符号被设为本地符号，可以对带有相同“AAA”符号的不同任务分配不同的地址 （例如 CIO 0.00 和 CIO 3.00）。



5-5-5 网络符号 （变量）

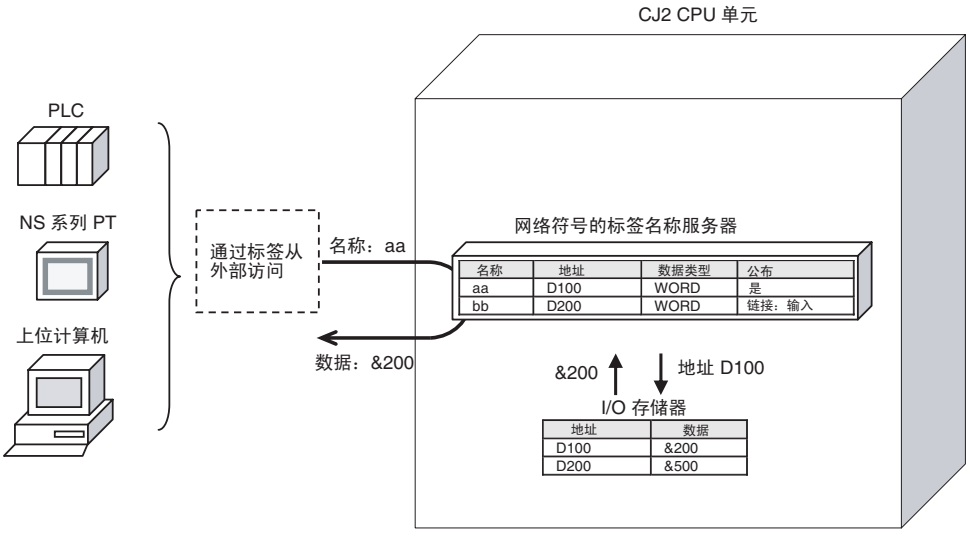
概述

通过在 CPU 单元中定义的网络符号 （变量），标签可用于从外部设备访问本地 CPU 单元的 I/O 存储器。CJ2 CPU 单元有一个内部标签名称服务器，可将网络符号转换为实际 I/O 地址，因此能够通过从外部使用标签访问网络符号来访问 CPU 单元的 I/O 存储器。

网络符号可用于以下应用。

- EtherNet/IP 上其他 PLC 的数据链接
- 通过上位计算机或 PT 从外部访问标签

在 CX-Programmer 的全局符号表中设定网络符号。



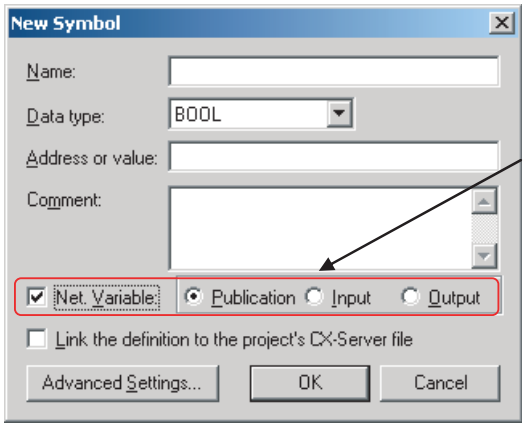
设定网络符号 （变量）

使用以下步骤来设定网络符号 （变量）。

1. 右键单击全局变量表并选择 “插入符号”。

注　　不能在本地符号表中指定网络符号。

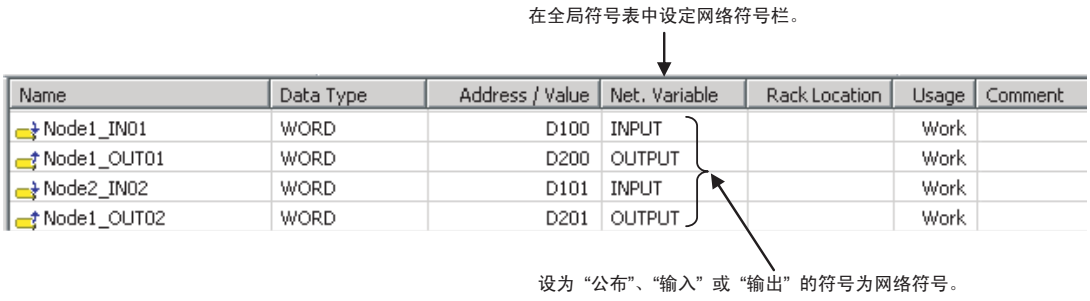
2. 将显示以下新符号对话框。
选择 “网络，变量” 检查框，然后选择 “公布”、“输入” 或 “输出” 选项，并点击 OK 键。



网络符号名称最多可使用 48 个字符。不能通过名称来反映事件。

符号类型	描述	全局符号表中的网络符号栏	描述
网络符号 （变量）	通过标签可从外部访问符号 （变量）	公布	• 通过 CIP 报文通信可从外部访问符号 （变量） • 不支持标签数据链接
		输入	• 通过 CIP 报文通信或标签数据链接可从外部访问符号 （变量） • 使用标签数据链接时，符号用于数据输入 （从其他 CPU 单元到本地 CPU 单元）
		输出	• 通过 CIP 报文通信或标签数据链接可从外部访问符号 （变量） • 使用标签数据链接时，符号用于数据输出 （从本地 CPU 单元到其他 CPU 单元）

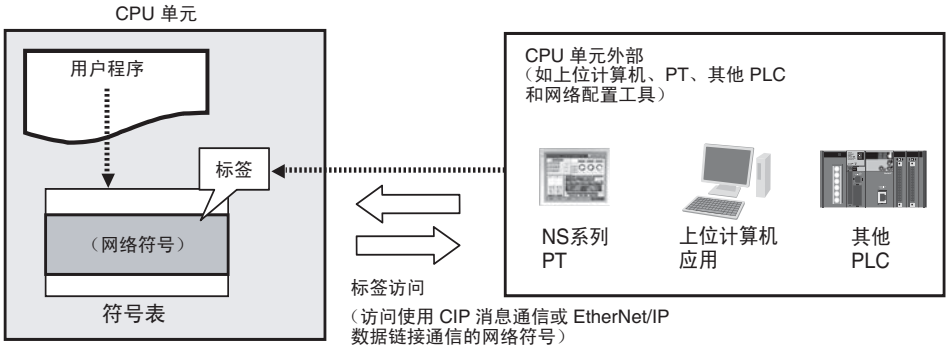
● 全局符号表中的网络符号 （变量）栏



网络符号 （变量）使用条件和处理

● 从外部处理网络符号 （变量）

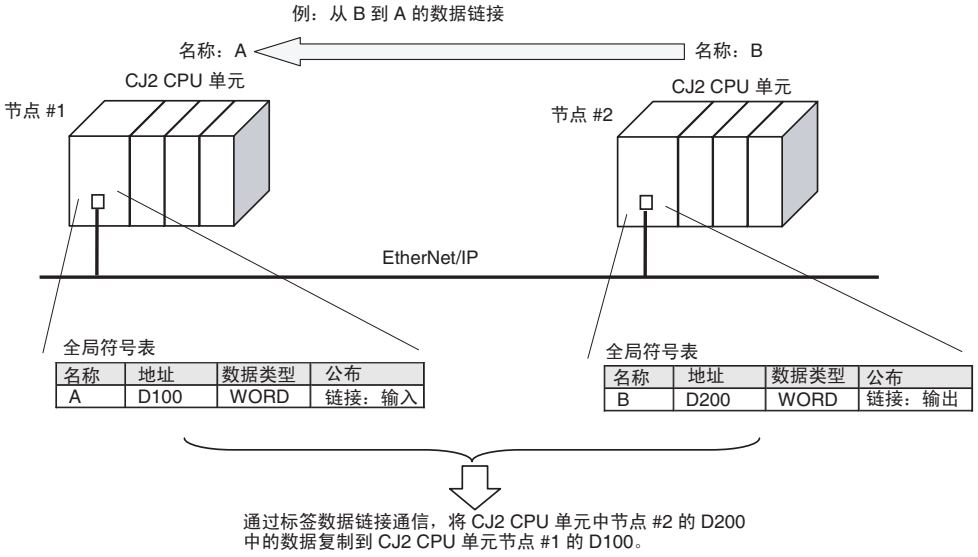
CPU 单元中设为网络符号的符号从 PLC 外部 （如上位计算机、 PT、网络配置工具或其他 PLC）作为标签被识别。



网络符号 （变量）应用

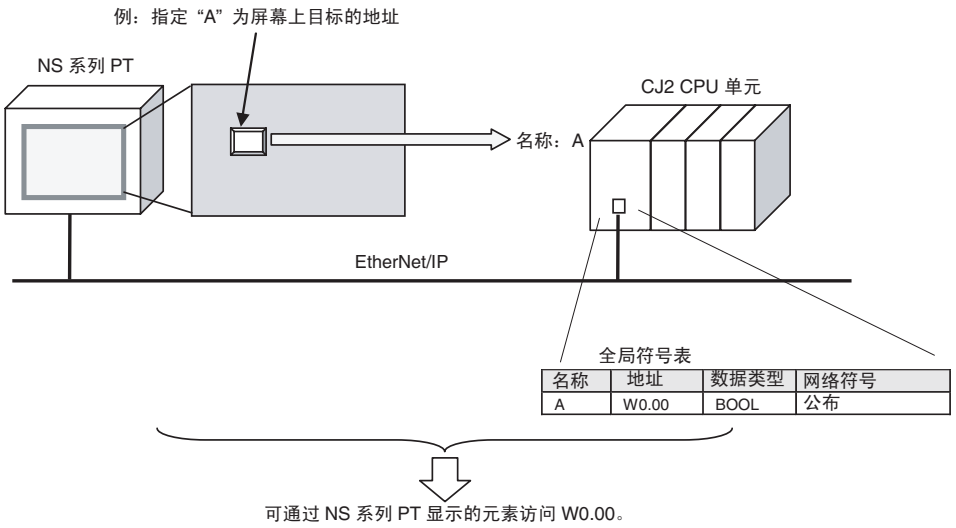
● 使用网络符号 （变量）作为 EtherNet/IP 数据链接标签

通过 EtherNet/IP 标签数据链接通信，可用标签指定远程 PLC 中的数据。在全局符号表的网络符号栏中将网络符号设为 “输入”或 “输出”。



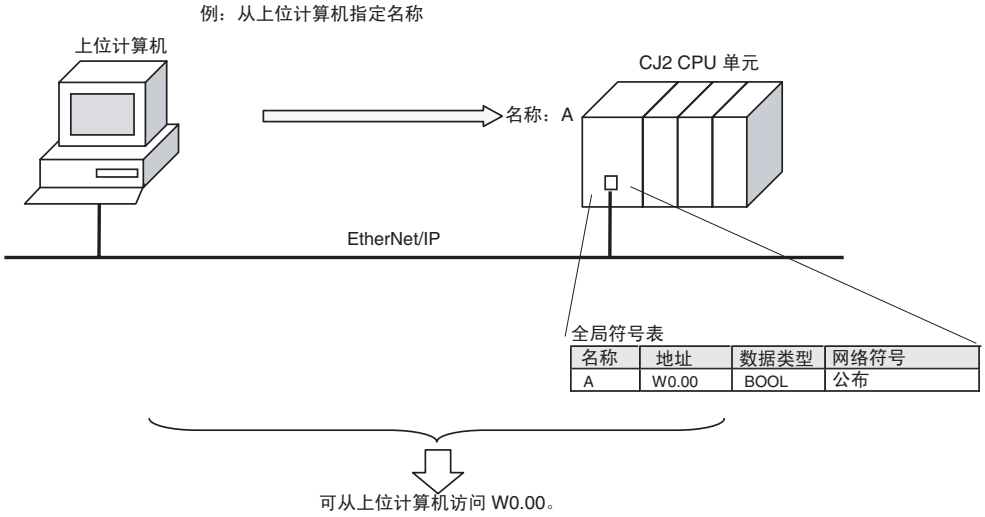
● 通过 NS 系列 PT 在通信中使用标签

通过 NS 系列 PT 显示的目标可指定远程 PLC 中的数据。在全局符号表的网络 （变量）符号栏中将网络 （变量）符号设为 “公布”。



● 通过上位计算机在 CIP 报文通信中使用标签

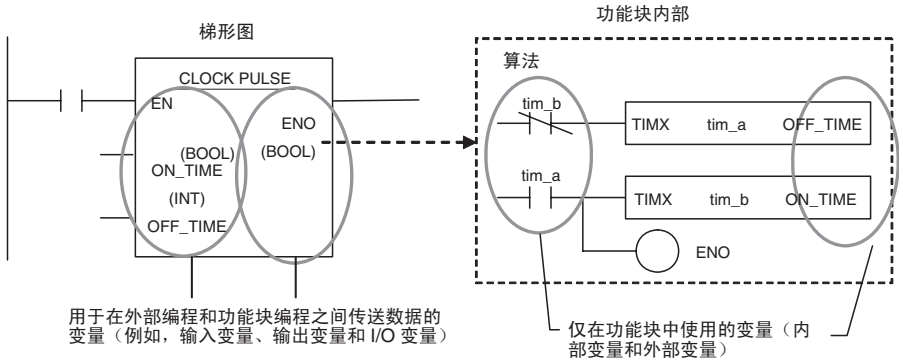
可从上位计算机用标签指定远程 PLC 中的数据。在全局符号表中将网络（变量）符号设为“公布”。



5-5-6 功能块中的变量

功能块中的程序全部以变量写入而不是实际地址。功能块中变量的变量类型和规格与功能块外部的符号（变量）不同。关于功能块中变量的详细信息，请参考 CX-Programmer 操作手册：功能块和结构文本（样本编号 W447）。

● 功能块变量区域



5-5-7 符号（变量）数据类型

可以对符号分配的地址指定数据类型。

可对符号（变量）设定的数据类型

可在功能块内部指定的数据类型与功能块外部指定的数据类型不同。每个指定的可用数据类型如下表所示。

数据类型	编程中的符号（变量）	功能块中的变量
BOOL（位）	可以	可以
UINT（1 字不带符号的二进制）	可以	可以
UDINT（2 字不带符号的二进制）	可以	可以
ULINT（4 字不带符号的二进制）	可以	可以
UINT BCD（1 字不带符号的二进制）*1	可以	不可以
INT（1 字带符号的二进制）	可以	可以
DINT（2 字带符号的二进制）	可以	可以
LINT（4 字带符号的二进制）	可以	可以
UDINT BCD（2 字带符号的二进制）*1	可以	不可以
ULINT BCD（4 字带符号的二进制）*1	可以	不可以
REAL（2 字浮点）	可以	可以
LREAL（4 字浮点）	可以	可以
CHANNEL（字）*1	可以	不可以
NUMBER（常数或编号）*2	可以	不可以
WORD（1 字十六进制）	可以	可以
WORD（2 字十六进制）	可以	可以
LWORD（4 字十六进制）	可以	可以
STRING（字符串：1 ～ 255 ASCII 字符）	可以	可以

*1 数据类型将被转换为以下的 ST 编程。

- UINT BCD: WORD
- UDINT BCD: DWORD
- ULINT BCD: LWORD
- CHANNEL: WORD

*2 这些数据类型不能在 ST 编程中使用。否则将发生程序错误。

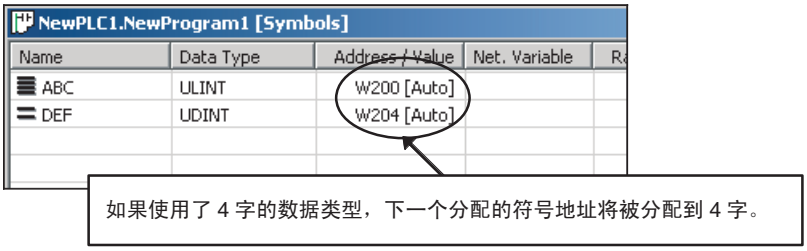
数据类型的应用

对符号 （变量）设定的数据类型可用于以下功能。

● 使用自动地址分配的应用

通过 CX-Programmer 使用自动地址分配时，根据符号的数据类型来分配地址。

示例：设定 ULINT （4 字不带符号二进制）数据类型



指定数组

数组可作为符号 （变量）属性 （仅 1 维数组）指定。除 STRING 和 NUMBER 以外可将所有数据类型指定为数组。

如果将数据属性一致的多个符号 （变量）作为一个组来进行管理，使用数组指定。进行以下的步骤来使用数组指定。

1. 在新符号 （变量）对话框或编辑符号对话框中输入符号名称，然后单击 “预设定” 按钮。将显示预设定对话框。
2. 在数组设定区域选择 “数组符号” 选项，然后输入最大项号数。

写入数组变量时，指定变量名称后面的括号中的后缀。

例：如果符号 （变量）名称为 PV，并且最大项数为 3，在指令操作数中将 PV[0]、 PV[1] 和 PV[2] 指定为数组变量。

数组的后缀以元素编号 0 开始依次指定。可以直接输入项编号，或者通过输入符号 （变量）或存储器地址来间接输入。

！ 注意

如果在梯形图或 ST 编程中将符号 （变量）或存储器地址 （只能对 ST 编程使用符号）指定为数组变量的后缀，确保指定的项编号没有超出最大存储器区范围。例如，写入程序，只有当间接指定在存储器范围内时，才能通过使用符号比较指令或 IF 语句来执行处理。对符号指定的项编号超出存储器区的最大范围将导致在不同的存储器区中访问数据，并可能出现意外的操作。



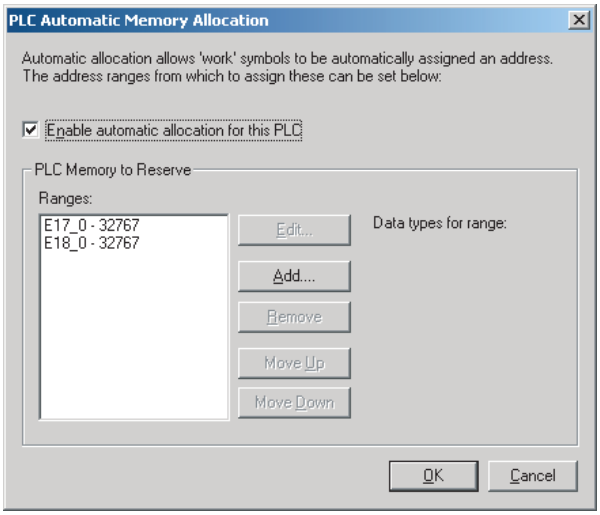
带字地址或符号 （变量）为后缀的数组变量不能作为操作数用于指令的立即刷新版本。

5-5-8 符号 （变量）的自动地址分配

编程时可将地址自动分配到符号 （变量）。在 CX-Programmer 的 PLC 菜单中选择 “存储器分配” - “自动地址分配”，然后设定自动分配的范围和数据类型。对于 CJ2 CPU 单元，CIO 区、辅助区、保持区、工作区、 DM 区或 EM 区中的地址可被分配。

在 EM 区中自动分配符号 （变量）地址时，推荐从最高 EM 区组的后面开始使用所需的数量。

下面的示例显示的是在 CJ2H-CPU68-EIP CPU 单元的组 17 和 18 hex （EM 区组 00 ～ 18 hex）中自动分配符号 （变量）地址的设定。



关于自动地址分配的区设定的详细信息，请参考 CX-Programmer 操作手册 （样本编号 W446）。

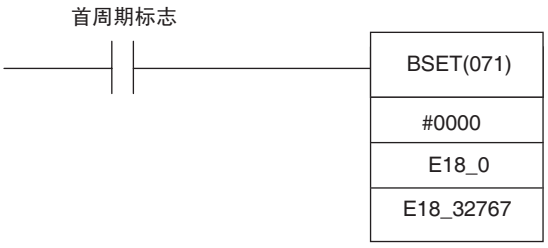
附加信息

在 EM 区组中设定的自动地址分配的位可被强制置位 / 强制复位。关于可被强制置位 / 强制复位的组的信息，请参考 6-12 扩展数据存储器区。

安全使用注意事项

电源关闭或操作模式改变时， EM 区中的数据被备份。在使用输出位指定为 BOOL 的数据时需小心。必要时，根据需要包含程序以清除存储器。

示例：电源为 ON， EM 区组 18 hex 设位自动地址分配时，以下指令可用于将组 18 hex 全部清零。

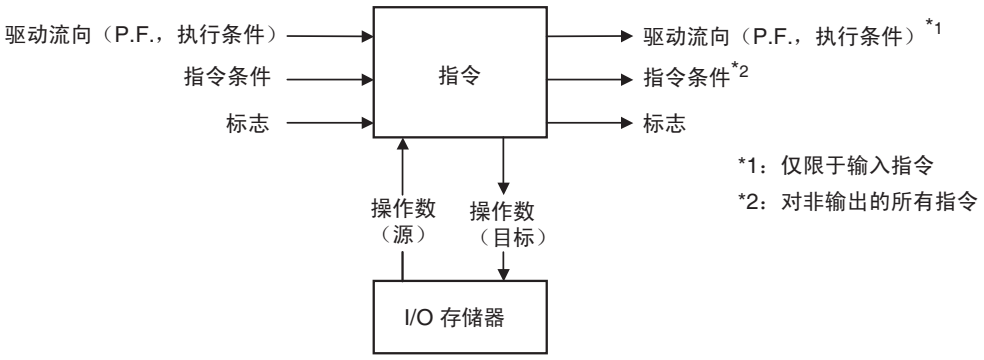


5-6 指令

5-6-1 指令的基本理解

指令结构

程序由指令组成。一条指令从输入到输出的基本结构如下图所示。

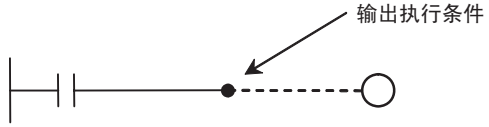


● 驱动流向

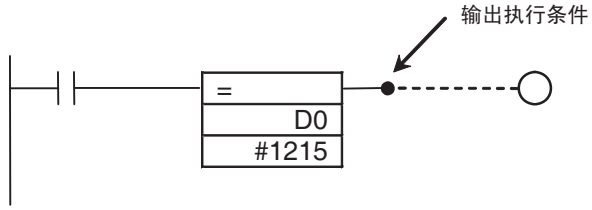
当程序正常执行时，驱动流向是用于控制执行和指令的执行条件。在梯形图程序中，驱动流向表示执行条件的状态。

输入指令

- Load 指令表示一个逻辑开始和输出执行条件。

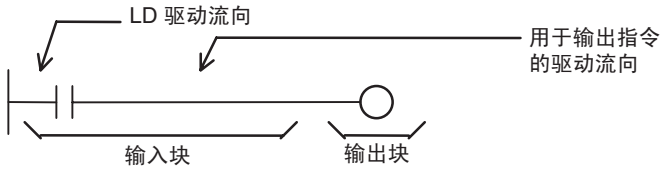


- 中间的指令输入驱动流向作为执行条件，并输出驱动流向到中间的指令或输出指令。



输出指令

用驱动流向作为执行条件，输出指令执行所有功能。



● 指令条件

指令条件是一些关系到下列指令的输出的所有指令的执行的特殊条件。当指令条件处于决定是否执行一条指令时，它比驱动流向 （P.F.）具有更高的优先级。根据指令执行条件，一条指令可能不执行或可能以不同方式执行。在每个任务开始时指令条件复位 （取消），即任务改变时它们被复位。

下列指令成对使用，用于设定和取消指令条件。这些成对指令必须在同一任务中使用。

指令条件	描述	设定指令	取消指令
联锁	联锁可锁定一部分程序。执行特殊条件，例如输出位变 OFF、定时器复位、计数器保持。	IL （002）	ILC （003）
BREAK （514）执行	执行过程中以 FOR （512）- NEXT （513）循环结束 （直到 NEXT （513）指令前的所有指令都不执行）。	BREAK （514）	NEXT （513）
	执行一条 JMP0 （515）～ JME0 （516）跳转指令。	JMP0 （515）	JME0 （516）
块程序执行	从 BPRG （096）～ BEND （801）执行程序块。	BPRG （096）	BEND （801）

● 标志

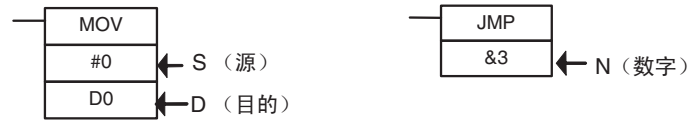
在这里，标志是在指令间作为接口的一个位。

输入标志		输出标志	
标志	描述	标志	描述
进位 （CY）标志	进位标志在数据移位指令和加法 / 减法指令中用作非特定的操作数。	条件标志	条件标志包括常 ON/OFF 标志，以及那些由指令执行结果更新的标志。在用户程序中，这些标志可由标签表示，例如 P_On、P_Off、P_ER、P_CY、P_EQ 等而不用地址表示。
用于特殊指令的标志	这些标志包括用于 FPD （269）指令的示教位标志和网络通信允许标志。	用于特殊指令的标志	这些标志包括存储卡指令标志和 MSG （046）执行完成标志。

● 操作数

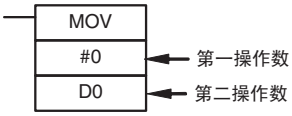
操作数指定预置的指令参数 （梯形图中的框），它用于指定 I/O 存储器区的内容或常数。输入地址或常数作为操作数后，指令就能被执行。操作数分成源操作数、目的操作数或数字操作数。

例



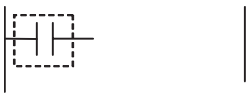
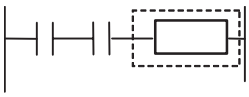
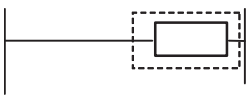
操作数类型		操作数符号（变量）	描述	
源	指定要读取数据或常数的地址。	S	源操作数	源操作数而非控制数据 （C）
		C	控制数据	控制数据是在源操作数中的复合数据，它取决于位状态而各具有不同含义。
目的 （结果）	指定要写入数据处的地址。	D	---	
数字	指定用于指令中的特殊数字，例如跳转编号或子程序编号。	N	---	

注　从指令的最上面开始，操作数也称作为第一操作数，第二操作数，依此类推。



指令位置和执行条件

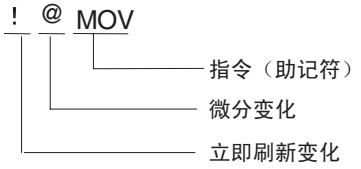
下表给出了指令允许的位置。指令分成需要执行条件的指令和不需要执行条件的指令。关于各指令的详细信息，请参考第 3 章 指令功能。

指令		位置	执行条件	图	示例
输入指令	逻辑开始（Load 指令）	直接与左总线或指令块的开始连接	不需要		LD, LD TST（350）, LD >（及其它符号 （变量）比较指令）
	中间指令	在逻辑开始和输出指令间	需要		AND, OR, AND TEST（350）, AND > （及其它 ADD 符号 （变量）比较指令）, UP （521）, DOWN （522）, NOT （520）等
输出指令		直接与右总线连接	需要		大多数指令包括 OUT 和 MOV （021）
			不需要		END （001）, JME（005）, FOR（512）, ILC（003）等

指令变化

下列变化允许指令以微分执行条件来执行并当指令执行时刷新指令 （立即刷新）。

变化		符号（变量）	描述
微分	ON	@	当执行条件变为 ON 时，指令执行微分操作。
	OFF	%	当执行条件变为 OFF 时，指令执行微分操作。
立即刷新		!	当指令执行时，由操作数或特殊 I/O 单元字指定 I/O 区的刷新数据。



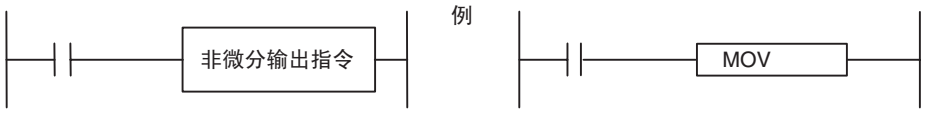
执行条件

可使用以下两种基本和特殊指令。

- 非微分指令：每个循环执行。
- 微分指令：仅执行一个循环。

● 非微分指令

输出指令 （需要输入条件的指令）：
当满足执行条件时 （ON 或 OFF），指令每个循环执行一次。



输入指令 （逻辑开始和中间指令）：
这些指令读取位状态、在每个循环作比较、测位或完成其它处理。如果其结果为 ON，驱动流向就输出 （即执行的条件变为 ON）。



● 输入微分指令

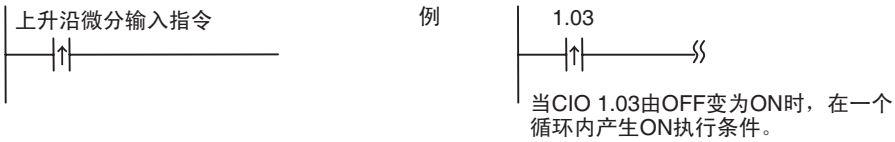
上升沿微分指令 （指令前加 @ 符号）

输出指令：
这类指令仅当循环中执行条件由 OFF 转为 ON 时执行，后面循环不再执行。

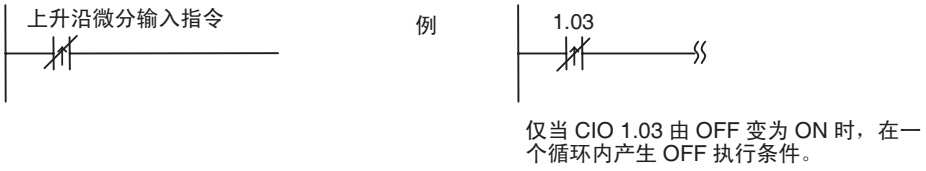


仅当CIO 1.02由OFF变为 ON时，执行MOV指令一次

输入指令 （逻辑开始和中间指令）：
当使开关从 OFF 到 ON 时，指令在每次循环读位状态、作比较、测试位或完成其它处理，并输出一个 ON 执行条件 （驱动流向）。到下一个循环执行条件变为 OFF。

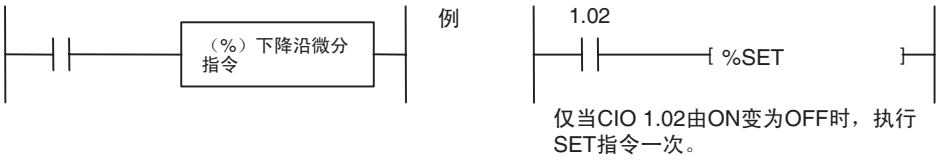


输入指令 （逻辑开始和中间指令）：
当使开关从 OFF 到 ON 时，指令在每次循环读位状态、作比较、测试位或完成其它处理，并输出一个 OFF 执行条件 （驱动流向停止）。到下一个循环执行条件变为 ON。



下降沿微分指令 （指令前加%符号）

输出指令：
这类指令仅当循环中执行条件由 ON 转为 OFF 时执行，后面循环不再执行。



输入指令 （逻辑开始和中间指令）：
当开关从 ON 到 OFF 时，指令在每次循环读位状态、作比较、测试位或完成其它处理，并输出一个执行条件 （驱动流向）。到下一个循环执行条件变为 OFF。

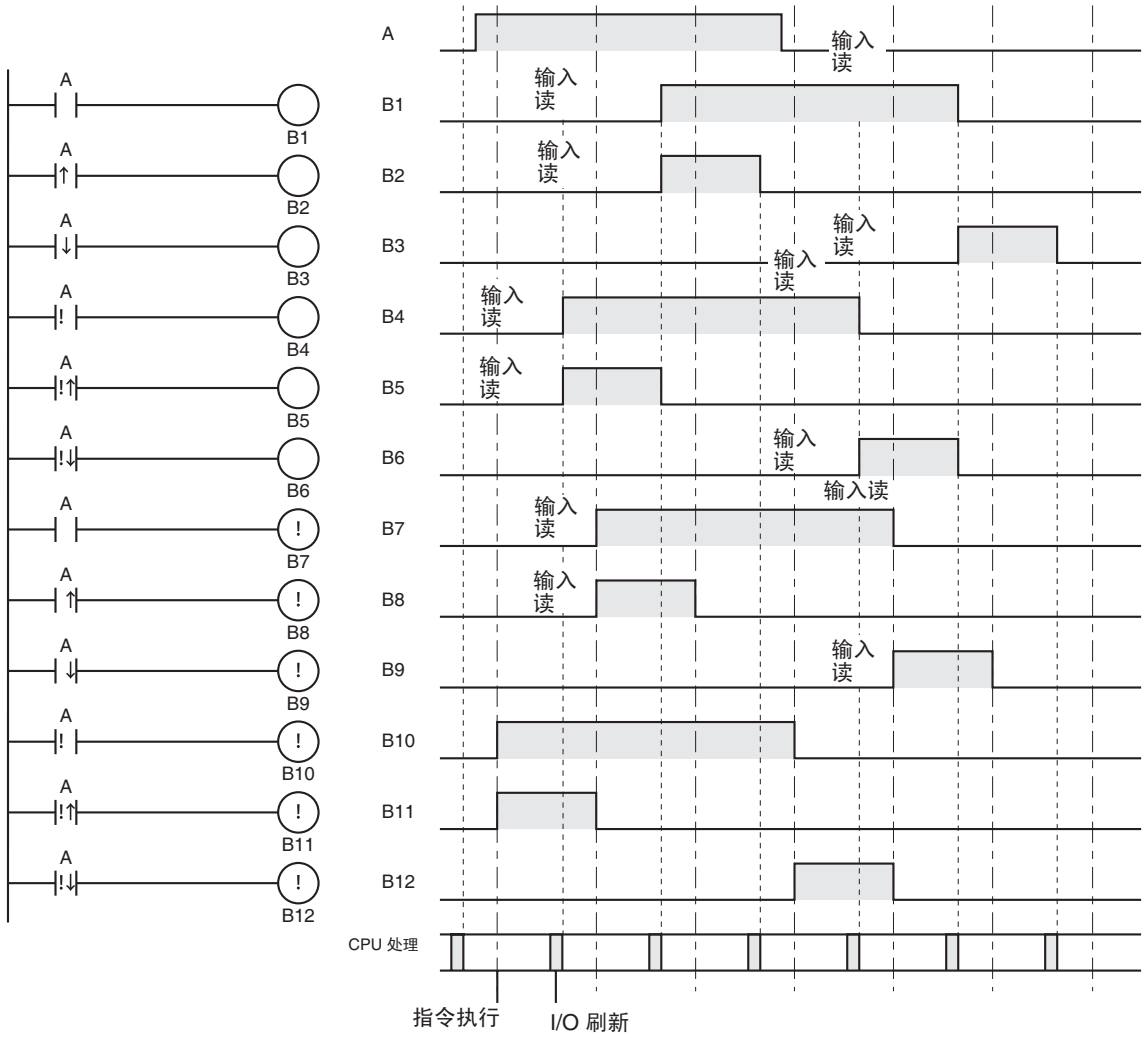


输入指令 （逻辑开始和中间指令）：
当开关从 ON 到 OFF 时，指令在每次循环读位状态、作比较、测试位或完成其它处理，并输出一个 OFF 执行条件 （驱动流向停止）。到下一个循环执行条件变为 ON。



I/O 指令时序

下列时序图给出了只用 LD 和 OUT 指令组成的程序中各个指令的不同运行时序。

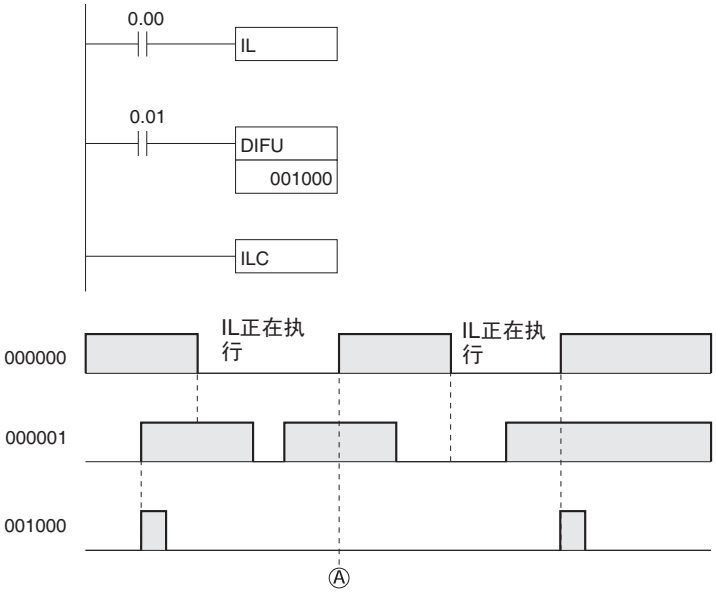


● 微分指令

- 微分指令具有区分以前的状态是 ON 还是 OFF 的内部标志。在运行开始时，上升沿微分指令（DIFU 和 @ 指令）的前状态标志设置为 ON，而下降沿微分指令（DIFU 和 % 指令）的前状态标志设置为 OFF。这防止微分输出在运行开始时出现意外的输出。
- 当执行条件为 ON，且前状态标志为 OFF 时，上升沿微分指令 （DIFU 或 @ 指令）将输出 ON。

在联锁中使用微分指令 （IL-ILC 指令）

在下面例子中，用于微分指令的前状态标志保持以前的联锁状态，并且由于在联锁有效时不刷新，在 A 点不会输出一个微分输出。



在跳转 （JMP （004）- JME （005）指令）中使用微分指令

如同联锁一样，当指令跳转，用于微分指令的前状态标志不会改变，即前状态标志保持不变。

- 对于下降沿微分指令（DIFD（014）或带 a% 的指令），仅当前状态标志为 ON 时，输入变为 OFF 时，输出将变为 ON。
- 对于上升沿和下降沿微分指令，在下一循环中输出都将变为 OFF。

正确使用注意事项

请不要使用常 P_On 标志或 A200.11（第一循环标志）作为上升沿微分指令的输入位，请不要使用常 P_Off 标志作为下降微分指令的输入位。如果使用了其中之一，这条指令将永不执行。

5-6-2 定义操作数

I/O 存储器区寻址

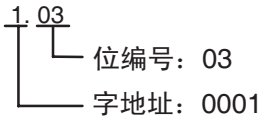
● 位地址

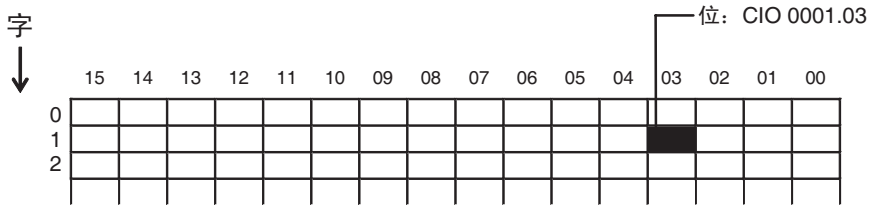
□□□□. □□

位编号（00～15）

字地址

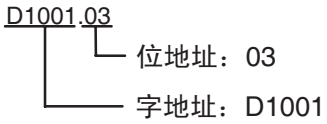
例：在 CIO 区字地址 1，位地址 03 的地址如下所示。

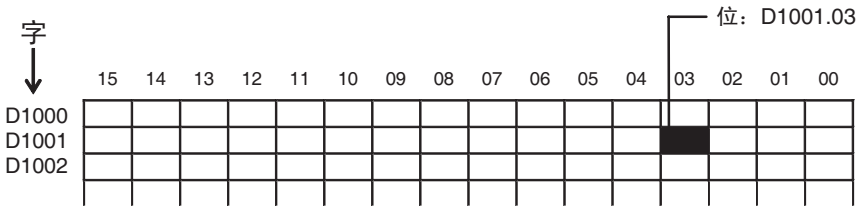




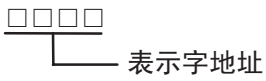
对于 CJ2 CPU 单元，可在 DM 区和 EM 区中指定位地址。

例：DM 区





● 字地址

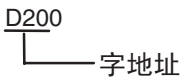


例：I/O 区

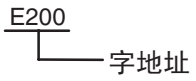


DM 和 EM 区地址在地址前加前缀 “D”或 “E”，如下地址 D200 所示。

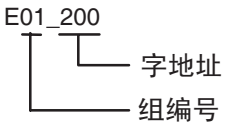
例：DM 区



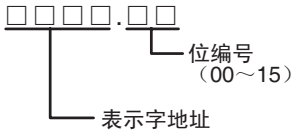
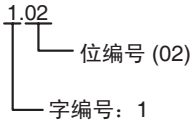
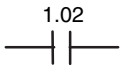
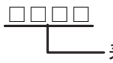
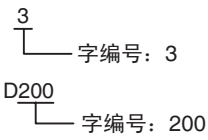
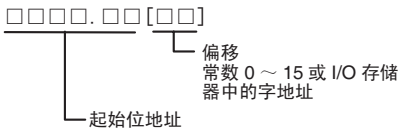
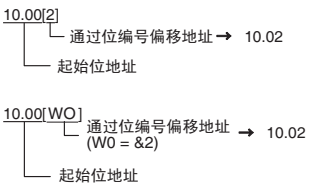
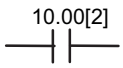
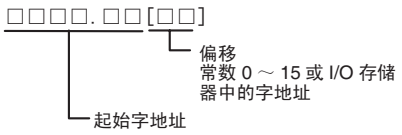
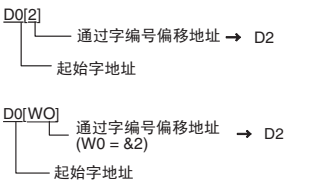
例：EM 区

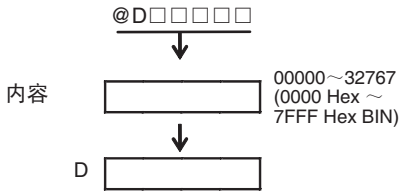
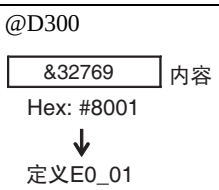
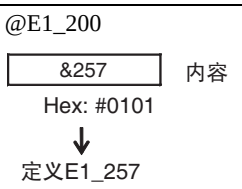
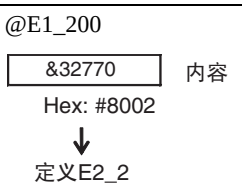


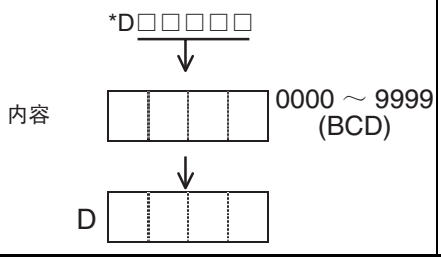
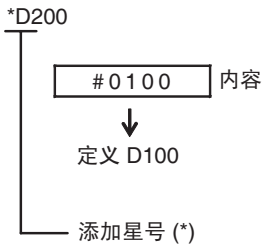
例：EM 区组 1



定义操作数

操作数	描述	注释	应用示例
定义位地址	用字和位编号直接定义一位地址 （输入位）。  访问定时器 / 计数器完成标志和当前值采用相同地址，任务标志也只有一个地址。	 	
定义字地址	字的编号直接用于定义 16 位的字地址  	 	MOV 3 D200
定义位地址的偏移	在空格内，定义位编号来偏移指定的起始位地址。  也可对起始位地址指定符号 （变量）。不管使用的是物理地址还是符号，只能使用保持、工作、 DM 和 EM 区地址。 常数或 I/O 存储器中的字地址可用于偏移。如果指定了字地址，字的内容直接用作偏移。	 	
定义字地址的偏移	在空格内，定义位编号来偏移指定的起始位地址。  也可对起始字地址指定符号 （变量）。不管使用的是物理地址还是符号，只能使用保持、工作、 DM 和 EM 区地址。 常数或 I/O 存储器中的字地址可用于偏移。如果指定了字地址，字的内容直接用作偏移。	 	MOV 3 D0[200]

操作数	描述	注释	应用示例
以二进制模式定义 DM/EM 间接地址 *1	从数据区开始定义偏移。地址的内容是以二进制数据 （00000 ～ 32767）在 DM 区或 EM 区中定义一个地址。通过在前面添加 @ 符号 （变量），以二进制模式定义一个间接地址。 		
	如果 @D □□□□□的内容在 0000 ～ 7FFF hex （十进制 0 ～ 32767）之间，地址定义在 D0 ～ D32767。		MOV #0001 @D300
	如果 @D □□□□□的内容在 8000 ～ FFFF hex （十进制 32768 ～ 65535）之间，地址定义在 EM 区组 0 的 E0 _0 ～ E0 _32767。		
	如果 @E □ _ □□□□□的内容在 0000 ～ 7FFF hex （十进制 0 ～ 32767）之间，在指定组中地址定义为 E □ _0 ～ E □ _32767。		MOV #0001@E1_200
	如果 @E □ _ □□□□□的内容在 8000 ～ FFFF hex （十进制 32768 ～ 65535）之间，在指定组 @ 下面的组中地址定义为 E （□ +1） _0 ～ E （□ +1） _32767。		
	*1 当用二进制模式定义一个间接地址时，DM 区和 EM 区（组 0 ～ 18 hex）作为连续存储地址处理。如果带 @ 符号（变量）地址的内容大于 32767，那么该地址将被作为 EM 区组 0 中从 0 开始持续的地址。 例： 如果 DM 区字的内容为 32768，那么 EM 区组 0 中的 E0_0 被指定。 如果 EM 区组编号定义为 “n”，字的内容大于 32767，那么地址就会定义在 EM 区组 N+1 中从 0 开始连续的地址。 例： 如果 EM 区中组 2 为 32768，那么 EM 区组 3 中的 E3_0 被指定。		

操作数	描述	注释	应用示例
以 BCD 模式定义 DM/EM 间接地址	从数据区开始定义偏移。地址的内容将作为 BCD 码数据 （0000 ～ 9999）在 DM 区或 EM 区中定义一个字地址。通过在前面添加星号 （*），以 BCD 模式定义一个间接地址。 		MOV #0001 *D200

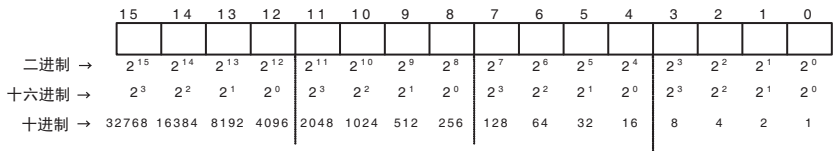
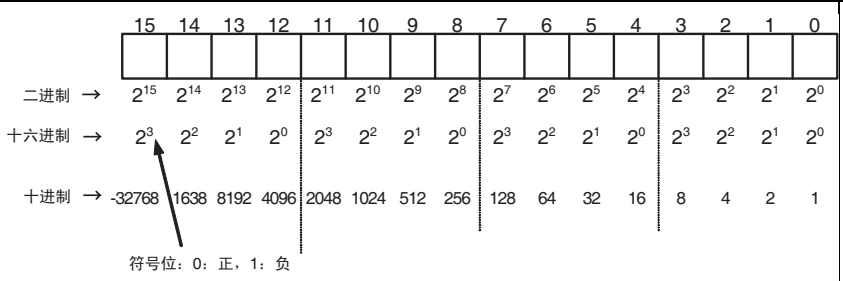
操作数	描述		注释	应用示例
直接定义一个寄存器	通过定义 IR □ （□：0 ～ 15）或 DR □ （□：0 ～ 15）直接定义一个变址寄存器 （IR）或数据寄存器（DR）。		IR0 IR1	MOVR 1.02 IR0 CI0 0010 的 PLC 存储器地址储存到 IR0 中。 MOVR 10 IR1 CIO 0010 的 PLC 存储器地址储存到 IR1 中。
采用寄存器定义一个间接地址	间接地址（无偏移）	PLC 存储器的位或字由 IR □定义。 指令操作数的位和字由 ,IR □定义。	,IR0 ,IR1	LD ,IR0 读取 IR0 作为 PLC 存储器地址的位。 MOV #0001 ,IR1 将 IR1 储存到 PLC 存储器地址的字 #0001。
	常数偏移	将 IR □中内容加或减一常数作为 PLC 存储器地址的位或字。 将 +/- 常数定义为 ,IR □。常数偏移范围从 -2048 ～ +2047 （十进制）。当指令执行时，偏移值转换为二进制数据。	+5 ,IR0 31 ,IR1	LD +5 ,IR0 读取 IR0 + 5 作为 PLC 存储器地址的位。 MOV #0001 +31 ,IR1 将 IR1 + 31 储存到 PLC 存储器地址的字 #0001。
	DR 偏移	将 DR □内容加上 IR □内容作为 PLC 存储器地址的位或字。 用 DR □ , IR □ . DR （数据寄存器）的内容作带符号二进制数据处理。如果带符号二进制值为负数，IR □的内容作负偏移。	DR0 ,IR0 DR0 ,IR1	LD DR0 ,IR0 读取 IR0+DR0 中的值作为 PLC 存储器地址的位。 MOV #0001 DR0 ,IR1 将 IR1+DR0 中的值储存到 PLC 存储器地址的字 #0001。
	自动递增	将 IR □中内容加 1 或加 2 作为 PLC 存储器地址。 +1：定义为 ,IR □ + +2：定义为 ,IR □ + +	,IR0++ ,IR1+	LD ,IR0 ++ 读取 IR0 中 PLC 存储器地址的位后，将 IR0 中内容加 2。 MOV #0001 ,IR1 + 将 IR1 中的值储存到 PLC 存储器地址的字 #0001 后，将 IR1 中内容加 1。
	自动递减	将 IR □中内容减 1 或减 2 作为 PLC 存储器地址。 -1：定义为 ,-IR □ -2：定义为 ,- -IR □	,--IR0 ,IR1	LD ,- -IR0 将 IR0 中的内容减 2 后，读取 IR0 中 PLC 存储器地址的位。 MOV #0001 ,-IR1 将 IR1 中的内容减 1 后， IR1 储存到 PLC 存储器地址的字 #0001。

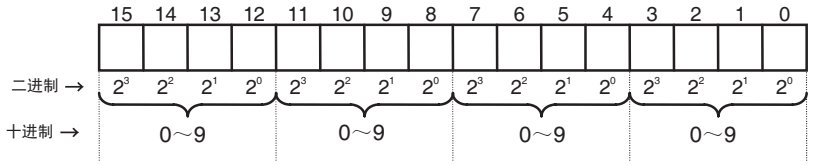
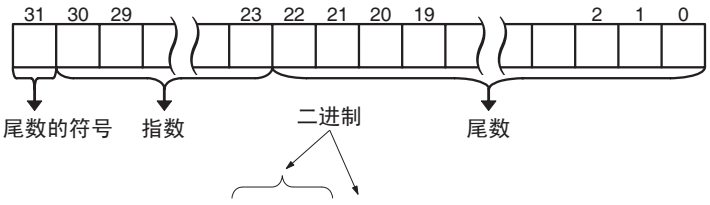
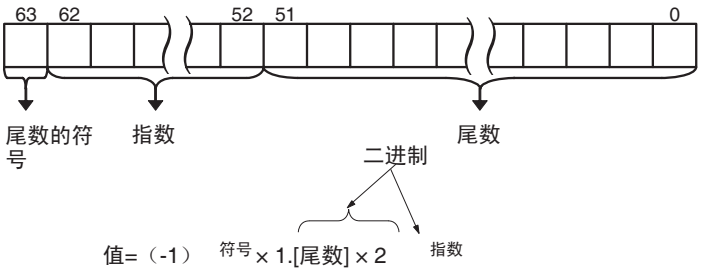
数据	操作数	数据格式	符号（变量）	范围	应用示例
16 位常数	所有二进制数据或一定范围内的二进制数据	不带符号二进制	#	#0000 ～ #FFFF	MOV #0100 D0 在 D0 中储存 #0100 hex （&256 十进制）。 +#0009 #0001 D1 在 D1 中储存 #000A hex （&10 十进制）。
		带符号十进制	±	-32768 ～ +32767	MOV -100 D0 在 D0 中储存 -100 十进制 （#FF9C hex）。 +-9 -1 D1 在 D1 中储存 -10 十进制 （#FFF6 hex）。
		不带符号十进制	&	&0 ～ &65535	MOV &256 D0 在 D0 中储存 -256 十进制 （#0100 hex）。 +&9 &1 D1 在 D1 中储存 -10 十进制 （#000A hex）。
	所有 BCD 码数据或一定范围内的 BCD 码数据	BCD	#	#0000 ～ #9999	MOV #0100 D0 在 D0 中储存 #0100 （BCD）。 +B #0009 #0001 D1 在 D1 中储存 #0010 （BCD）。
32 位常数	所有二进制数据或一定范围内的二进制数据	不带符号二进制	#	#00000000 ～ #FFFFFFFF	MOVL #12345678 D0 在 D0 和 D1 中储存 #12345678 hex。 D1 D0 1234 5678
		带符号二进制	+	-2147483648 ～ +2147483647	MOVL -12345678 D0 在 D0 和 D1 中储存 -12345678 十进制。
		不带符号十进制	&	&0 ～ &4294967295	MOVL &12345678 D0 在 D0 和 D1 中储存 &12345678 十进制。
	所有 BCD 码数据或一定范围内的 BCD 码数据	BCD	#	#00000000 ～ #99999999	MOVL #12345678 D0 在 D0 和 D1 中储存 #12345678 （BCD）。

<table>
<tr><th>数据</th><th>描述</th><th>符号
（变量）</th><th>示例</th><th>---</th></tr>
<tr>
<td>文本字符串</td>
<td>文本字符串数据以 ASCII 码方式保存 （除特殊字符外，一个字符占一个字节），顺序从最高位到最低位字节和从最低位 （最小）到最高位字。
如果字符数为奇数个，那么就将 00 Hex （NUL 代码）保存到最后一个字的最低位上。
如果字符数为偶数个，那么就将 0000 Hex （2NUL 代码）保存到末尾字 +1 后面空闲字节的最低和最高位上。</td>
<td>---</td>
<td>↓ ABCDE
<table>
<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>
=
<table>
<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>
ABCD
<table>
<tr><td>‘A’</td><td>‘B’</td></tr>
<tr><td>‘C’</td><td>‘D’</td></tr>
<tr><td>NUL</td><td>NUL</td></tr>
</table>
=
<table>
<tr><td>41</td><td>42</td></tr>
<tr><td>43</td><td>44</td></tr>
<tr><td>00</td><td>00</td></tr>
</table>
</td>
<td>MOV$ D100 D200
<table>
<tr><td>D100</td><td>41</td><td>42</td></tr>
<tr><td>D101</td><td>43</td><td>44</td></tr>
<tr><td>D102</td><td>45</td><td>00</td></tr>
</table>
↓
<table>
<tr><td>D200</td><td>41</td><td>42</td></tr>
<tr><td>D201</td><td>43</td><td>44</td></tr>
<tr><td>D202</td><td>45</td><td>00</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td colspan="4">可用于文本字符串的 ASCII 字符包括字母数字字符、日文片假名和符号 （除特殊字符外），这些字符如下表所示。
<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="16">高位 4 位</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td rowspan="16">低位 4 位</td><td>0</td><td></td><td></td><td>sp</td><td>0</td><td>@</td><td>P</td><td>`</td><td>p</td><td></td><td></td><td></td><td>ー</td><td>タ</td><td>ミ</td><td></td><td></td></tr>
<tr><td>1</td><td></td><td></td><td>!</td><td>1</td><td>A</td><td>Q</td><td>a</td><td>q</td><td></td><td></td><td>。</td><td>ア</td><td>チ</td><td>ム</td><td></td><td></td></tr>
<tr><td>2</td><td></td><td></td><td>″</td><td>2</td><td>B</td><td>R</td><td>b</td><td>r</td><td></td><td></td><td>「</td><td>イ</td><td>ツ</td><td>メ</td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td>#</td><td>3</td><td>C</td><td>S</td><td>c</td><td>s</td><td></td><td></td><td>」</td><td>ウ</td><td>テ</td><td>モ</td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td>$</td><td>4</td><td>D</td><td>T</td><td>d</td><td>t</td><td></td><td></td><td>、</td><td>エ</td><td>ト</td><td>ヤ</td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td>%</td><td>5</td><td>E</td><td>U</td><td>e</td><td>u</td><td></td><td></td><td>・</td><td>オ</td><td>ナ</td><td>ユ</td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td>&</td><td>6</td><td>F</td><td>V</td><td>f</td><td>v</td><td></td><td></td><td>ヲ</td><td>カ</td><td>ニ</td><td>ヨ</td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td>′</td><td>7</td><td>G</td><td>W</td><td>g</td><td>w</td><td></td><td></td><td>ァ</td><td>キ</td><td>ヌ</td><td>ラ</td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td>(</td><td>8</td><td>H</td><td>X</td><td>h</td><td>x</td><td></td><td></td><td>ィ</td><td>ク</td><td>ネ</td><td>リ</td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td>)</td><td>9</td><td>I</td><td>Y</td><td>i</td><td>y</td><td></td><td></td><td>ゥ</td><td>ケ</td><td>ノ</td><td>ル</td><td></td><td></td></tr>
<tr><td>A</td><td></td><td></td><td>*</td><td>:</td><td>J</td><td>Z</td><td>j</td><td>z</td><td></td><td></td><td>ェ</td><td>コ</td><td>ハ</td><td>レ</td><td></td><td></td></tr>
<tr><td>B</td><td></td><td></td><td>+</td><td>;</td><td>K</td><td>[</td><td>k</td><td>{</td><td></td><td></td><td>ォ</td><td>サ</td><td>ヒ</td><td>ロ</td><td></td><td></td></tr>
<tr><td>C</td><td></td><td></td><td>,</td><td><</td><td>L</td><td>¥</td><td>l</td><td>|</td><td></td><td></td><td>ャ</td><td>シ</td><td>フ</td><td>ワ</td><td></td><td></td></tr>
<tr><td>D</td><td></td><td></td><td>−</td><td>=</td><td>M</td><td>]</td><td>m</td><td>}</td><td></td><td></td><td>ュ</td><td>ス</td><td>ヘ</td><td>ン</td><td></td><td></td></tr>
<tr><td>E</td><td></td><td></td><td>.</td><td>></td><td>N</td><td>^</td><td>n</td><td>~</td><td></td><td></td><td>ョ</td><td>セ</td><td>ホ</td><td>゛</td><td></td><td></td></tr>
<tr><td>F</td><td></td><td></td><td>/</td><td>?</td><td>O</td><td>_</td><td>o</td><td></td><td></td><td></td><td>ッ</td><td>ソ</td><td>マ</td><td>゜</td><td></td><td></td></tr>
</table>
</td>
</tr>
</table>

5-6-3 数据格式

下表给出了 CJ 系列中可处理的数据格式。

数据类型	数据格式	十进制	4 位十六进制
不带符号的二进制数据		&0 ～ &65535	#0000 ～ #FFFF
带符号的二进制数据	 使用最高位作为符号位，将数据作为带符号的二进制 16 位数据来进行处理。值以 4 位十六进制来表示。 正数：如果最高位为 OFF，表示一个非负的值。对于 4 位十六进制来说，值为 0000 ～ 7FFF hex。 负数：如果最高位为 ON，表示一个负的值。对于 4 位十六进制来说，值为 8000 ～ FFFF hex，并且由 2 的补码表示负数的绝对值（十进制）。	0 ～ -32768 0 ～ +32767	负： #8000 ～ #FFFF 正： #0000 ～ #7FFF

数据类型	数据格式	十进制	4 位十六进制
BCD（二 - 十进制编码）		#0 ～ #9999	#0000 ～ #9999
单精度十进制浮点数	 数值 = （-1）符号 × 1.[尾数] × 2指数 符号（第31位） 1：负或0：正 尾数 从00位到22位的23位内容为尾数，即小数点后面的部分1.□□□□.....，以二进制形式表示。 指数 从23位到30位的8位内容为指数。二进制形式的指数用127+n表示2^n。 这是符合 IEEE754 标准的单精度浮点数格式，并且它仅用于浮点数转换或运算指令中。使用这种数据格式时，可用 CX-Programmer 的 I/O 存储器编辑和监控显示来进行设定或监控。为此，用户不必了解这种数据格式的细节，只要知道这种格式占用 2 个字。	---	---
双精度十进制浮点数	 值=（-1） 符号 × 1.[尾数] × 2 指数 符号（第63位） 1：负或0：正 尾数 从00位到51位的52位内容为尾数，即小数点后面的部分1.□□□□.....，以二进制形式表示。 指数 从52位到62位的11位内容为指数。二进制形式的指数用1023+n表示2^n。 这是符合 IEEE754 标准的双精度浮点数格式，并且它仅用于浮点数转换或运算指令中。使用这种数据格式时，可用 CX-Programmer 的 I/O 存储器编辑和监控显示来进行设定或监控。为此，用户不必了解这种数据格式的细节，只要知道这种格式占用 4 个字。	---	---



附加信息

- 补码
  一般以 x 为基数的补码用 x-1 减去一个给定数字的所有位，然后再在最低位加 1 求得 （例如：7556 的十进制补码是：9999-7556 +1= 2444）。一个补码通常用于减法和其它功能如加法等。

  例如：根据 8954-7556 = 1398，8954 + （7556 的十进制补码）=8954 + 2444=11398。如果不计最高位，我们得到减法运算的结果 1398。

- 二进制补码
  一个二进制补码是一个以 2 为基数的补码，这里，我们可用 1 （2-1=1）减所有位再加 1 求得。

  例如 ：二进制数 1101 的二进制补码是 1111（F Hex）-1101（D Hex）+1（1 Hex）=0011（3 Hex）。下面是以 4 位十六进制 （Hex）表示。
  一个 a Hex 的二进制补码 b Hex 是：FFFF Hex - a Hex + 0001 Hex = b Hex。所以 “a Hex”的二进制补码 b Hex 的方法为：b Hex = 10000 Hex - a Hex。

  例如：求 3039 Hex 的二进制补码，采用：10000 Hex - 3039 Hex =CFC7 Hex。
  同样地方用 a Hex= 10000 Hex - b Hex 方法从二进制补码 b Hex 求得 a Hex 值。

  例如：由二进制补码 CFC7 Hex 求它的真值，采用 10000 Hex - CFC7 Hex =3039Hex。
  CJ 系列有两条指令：NEG （160）（二进制求补）和 NEGL （161）（双字二进制求补）这二条指令可以用于由一个真值求二进制补码或者由二进制补码求其真值。

● 以 1 字数据表示的值

值（十进制）	二进制表示		十六进制表示	BCD 表示（十进制）
	十进制表示			
	不带符号	带符号		
1	&1	+1	#0001	#0001
2	&2	+2	#0002	#0002
3	&3	+3	#0003	#0003
4	&4	+4	#0004	#0004
5	&5	+5	#0005	#0005
6	&6	+6	#0006	#0006
7	&7	+7	#0007	#0007
8	&8	+8	#0008	#0008
9	&9	+9	#0009	#0009
10	&10	+10	#000A	#0010
11	&11	+11	#000B	#0011
12	&12	+12	#000C	#0012
13	&13	+13	#000D	#0013
14	&14	+14	#000E	#0014
15	&15	+15	#000F	#0015
16	&16	+16	#0010	#0016
:	:	:	:	:
9999	&9999	+9999	#270F	#9999
10000	&10000	+10000	#2710	不适用
	:	:	:	
32767	&32767	+32767	#7FFF	
32768	&32768	不适用	#8000	
:	:		:	
65535	&65535		#FFFF	

-1	不适用	-1	#FFFF	不适用
:		:	:	
-32768		-32768	#8000	
-32769		不适用	不适用	

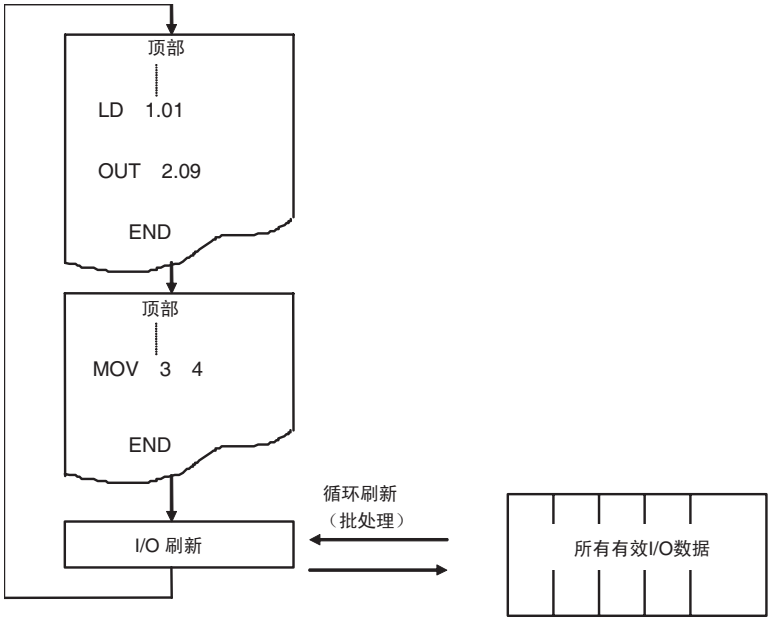
5-6-4 I/O 刷新时序

以下的方法用于刷新外部 I/O。

- 循环刷新
- 立即刷新（用 ! 定义的指令和 IORF（097）、FIORF（225）或 DLNK（226）指令）

循环刷新

对于循环刷新，在“就绪”（READY）状态时所有循环任务执行完毕后，执行 I/O 刷新一次。



如果在任务中需要刷新 I/O，在 END （001）指令前对所有需要的字执行一条 IORF （097）指令。

立即刷新

对于立即刷新，不管程序中的指令在何时执行，都执行刷新。

● 立即刷新的类型

有两种方法可以执行立即刷新：使用指令并定义立即刷新变量 （！）或使用立即刷新的特殊指令。

带刷新变量 （!）的指令

通过使用指令的立即刷新变量 （!），指令执行时可执行刷新 （在循环中）。

立即刷新的特殊指令

以下三种指令可用于决定哪个单元的存储器被刷新。

IORF （097）指令：基本 I/O 单元和特殊 I/O 单元 （仅在 CIO 区中分配字）

FIORF （225）指令：特殊 I/O 单元

DLNK （226）指令：CPU 总线单元

下表显示了这些单元可通过哪条指令被刷新。

单元	带立即刷新变量（!）的指令	IORF （097）指令	FIORF （225）指令	DLNK （226）指令
基本 I/O 单元	是	是	否	否
在 CIO 区中分配的字				

单元	带立即刷新变量（!）的指令	IORF （097）指令	FIORF （225）指令	DLNK （226）指令
特殊 I/O 单元	否	是 *2	是 *2	否
在 CIO 区中分配的字				
在 DM 区中分配的字		否		
CPU 总线单元	否	否	否	是
在 CIO 区中分配的字				
在 DM 区中分配的字				
单元指定存储器 *1				

*1 EtherNet/IP 单元、 Controller Link 单元、 SYSMAC LINK 单元的数据连接及 DeviceNet 远程 I/O 通信。

*2 通过使用 IORF （097）指令或 FIORF （225）指令，可刷新特殊 I/O 单元 CIO 区中分配的字。 FIORF （225）指令的指令执行时间小于 IORF （097）指令。由于在特殊 I/O 单元操作数中指定了所需的单元编号，因此更易于使用。

正确使用注意事项

立即刷新变量的执行时间要大于指令的常规变量，因此请注意将需要更长的循环时间。

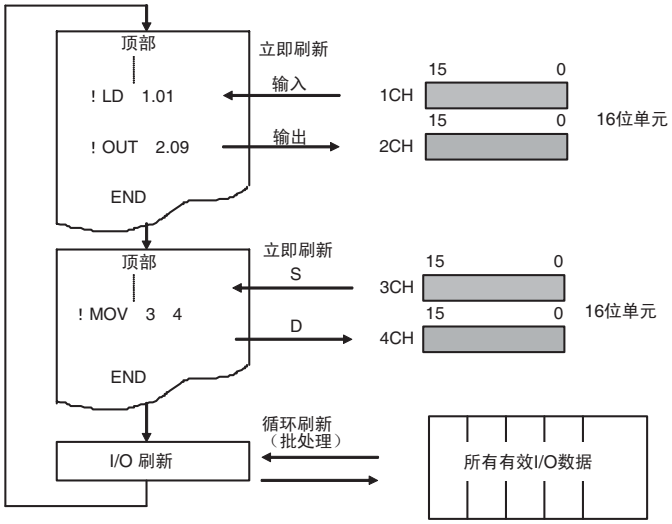
● 带刷新变量 （!）的指令

在指令前面添加一个感叹号来定义立即刷新。

- 如果一个有效 I/O 位被定义为一个操作数，如下所示指令执行时 I/O 将刷新。

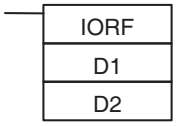
单元	刷新数据
基本 I/O 单元	对包含 16 位的位，I/O 将被刷新。

- 当一条指令字操作数被定义，作为所定义的这 16 位， I/O 将被刷新。
- 在一条指令被执行前，作为输入或源操作数的输入将被刷新。
- 在一条指令被执行后，作为输出或目的 （D）操作数的输出将被刷新。



● IORF （097）：I/O 刷新

IORF （097）用于对基本 I/O 单元实际 I/O 在一个循环中的所有数据以及分配到特殊 I/O 单元的 CIO 区字的数据进行刷新。

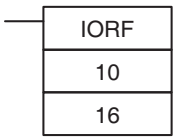


D1：起始字

D2：结束字

当执行IORF（097）时，从D1到D2包括的所有字中的数据被刷新。

例：



当执行IORF（097）时，刷新从CIO 10～CIO 16的7个字。

当基本 I/O 单元的输入或输出数据需要高速响应时，在计算指令的前后使用 IORF （097）。

正确使用注意事项

IORF （097）需要相对长的指令执行时间，并且执行时间的增长量与需刷新的字的数目成正比，因此它会显著的延长循环时间。请注意不要使循环时间变得过长。

● FIORF （225）：特殊 I/O 单元 I/O 刷新

FIORF （225）用于必要时根据指定单元编号，刷新以下特殊 I/O 单元中的数据。



例：



可在 PLC 设置中禁用特殊 I/O 单元的常规循环 I/O 刷新（通过将单元的特殊 I/O 单元循环刷新禁用位置 ON），仅在必要时通过执行 FIORF （225）指令执行单元的 I/O 刷新。当特殊 I/O 单元连接到 PLC 时，该功能可以防止 PLC 循环时间的延长。

● 使用 DLNK （226）：CPU 总线单元 I/O 刷新

DLNK （226）可用于增加刷新频率。以下数据可被刷新。

- 在 CIO 区中分配到单元的字
- 在 DM 区中分配到单元的字
- 对单元特殊的数据 *1

*1 对 CPU 总线单元特殊的数据包括：Controller Link 单元或 SYSMAC LINK 单元的数据链接，以及 DeviceNet 单元的远程 I/O 数据链接。



例：



*1 如果单元编号为 1 的 CPU 总线单元是 Controller Link 单元，数据链接将被刷新。

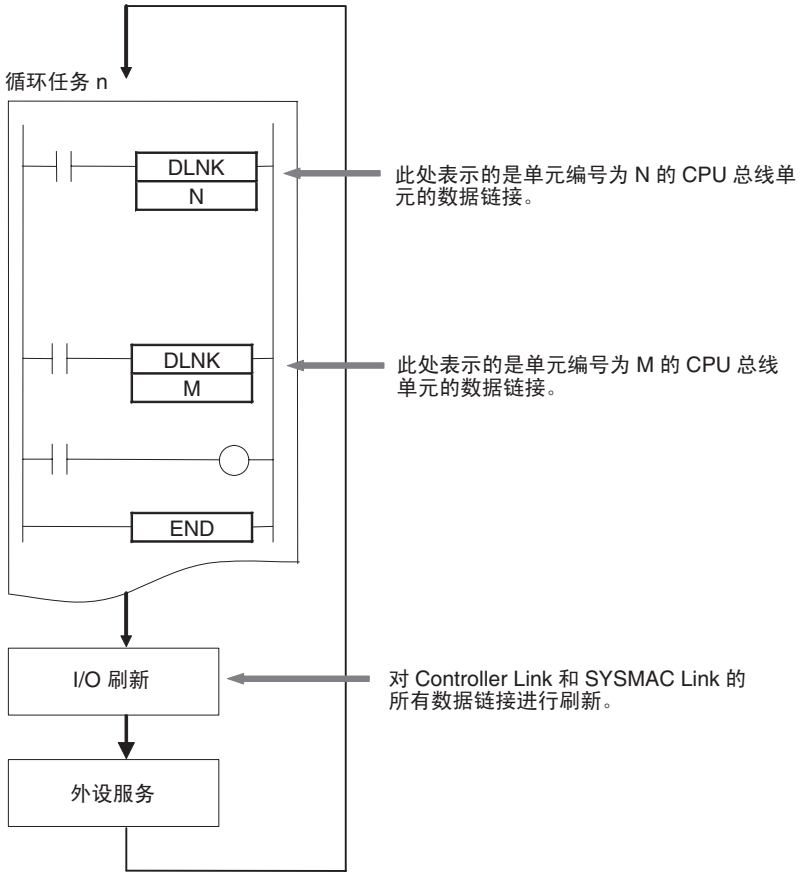
应用示例：

在一个长循环下，Controller Link 数据链接的刷新间隔将非常长。对于 Controller Link 单元通过执行 DLNK （226）可以缩短间隔，从而增加数据链接刷新的频率。

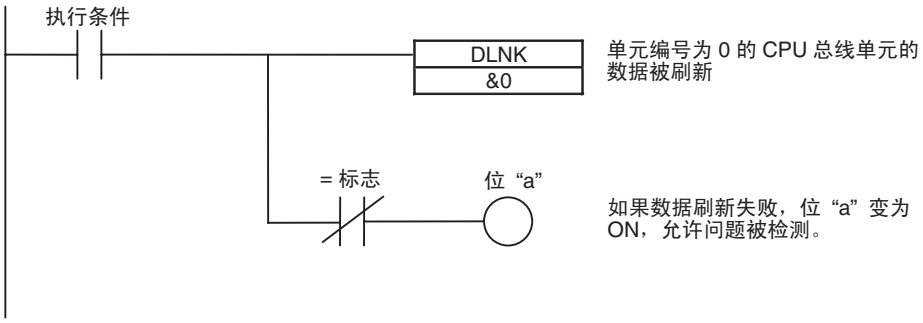
附加信息

下图显示的是一个 DLNK （226）指令的应用示例。

例：数据链接



如果对 CPU 总线单元执行 DLNK （226）出现占线刷新数据，数据将不能刷新并且相等标志将变为 OFF。通常，相等标识必须按如下编程以确保刷新正常完成。

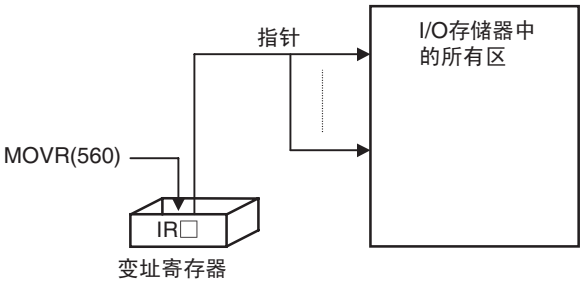


5-7 变址寄存器

5-7-1 什么是变址寄存器？

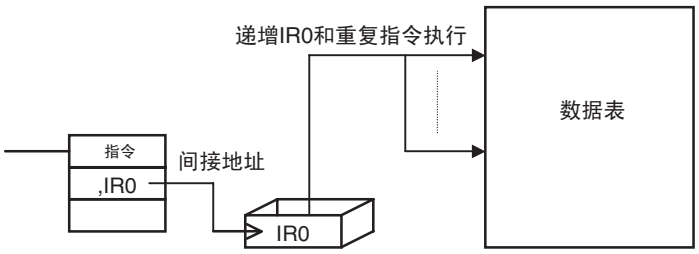
变址寄存器功能是作为指针用来指定 PLC 存储器地址，这个地址是 I/O 存储器中的绝对存储器地址。使用 MOVR （560）或 MOVRW （561）在变址寄存器中保存 PLC 存储器地址后，在其它指令中输入这一变址寄存器作为一个操作数，使保存的 PLC 存储器地址成为间接地址。

变址寄存器的优势在于它们能够指定在 I/O 存储器中的任何一个位或字，包括定时器和计数器 PV。



5-7-2 使用变址寄存器

当变址寄存器与循环 （如 FOR- NEXT 循环）指令共同使用时，是一种有用的工具。变址寄存器的内容可以非常容易地递增、递减和偏移，因此一个循环中少量的指令可以非常有效地处理连续数据的表格。



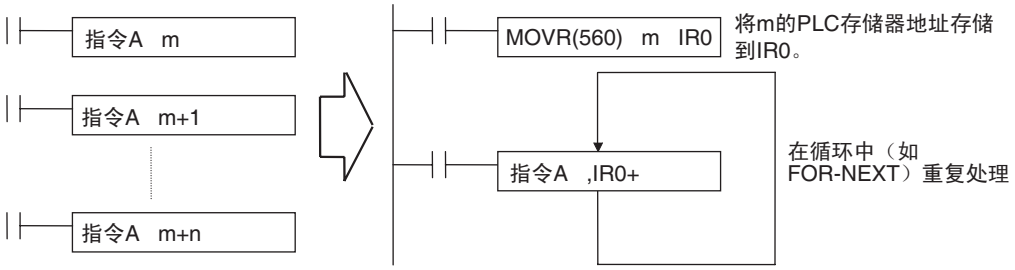
通常可以通过以下步骤使用变址寄存器：

1. 在一个变址寄存器中使用 MOVR （560）保存所需位或字的 PLC 存储器地址。
2. 在任何指令中把变址寄存器指定为一个操作数，作为所需的位或字的间接地址。
3. 偏移或递增原始 PLC 存储器地址 （见下）使指针指向另一个地址。
4. 继续步骤 2 和 3，执行地址的任一位数上的指令。

注　使用以下任一方法可偏移或递增变址寄存器。

- 变址寄存器的间接地址：
  带自动递增 （,IR □ + 或 ,IR □ ++）的间接地址、带自动递减 （,-IR □或 ,--IR □）的间接地址、带常数偏移的间接指定 （常数 ,IR □）、带 DR 偏移 （DR □ ,IR □）的间接地址。
- 用于变址寄存器直接寻址的指令
  BINARY ADD （+L） 、 BINARY SUBTRACT （-L） 、DOUBLE INCREMENT BINARY （++L）、DOUBLE DECREMENT BINARY （--L）。

例：



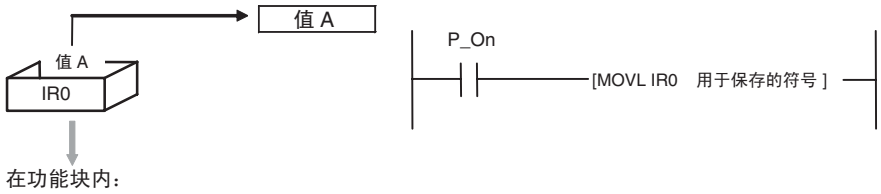
下例显示了在一个程序循环中用变址寄存器可替换一长串的指令。在本例中，指令 A 重复了 n+1 次来执行一些操作，如读入和比较表中的值。

正确使用注意事项

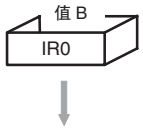
- 在功能块内部使用变址寄存器 IR0 ～ IR15 时，在其他功能块或功能块外部的程序中使用同一变址寄存器将在两者之间产生竞争，并使程序不能正确执行。因此，使用变址寄存器（IR0 ～ IR15）时，当功能块启动时（或变址寄存器被使用前），以及功能块完成时（或变址寄存器已被使用后），必须在指针中保存变址寄存器的值，在程序中合并处理将变址寄存器的值恢复到已保存的值。

  例：启动功能块（或使用变址寄存器前）

  1. 保存 IR 的值（例如 A）。

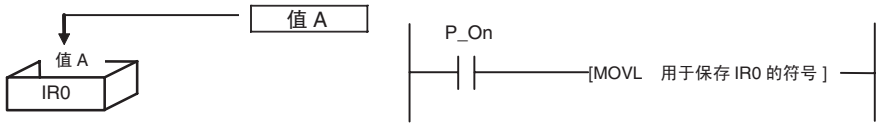


  2. 使用 IR。



  功能块启动时（或使用功能块前）：

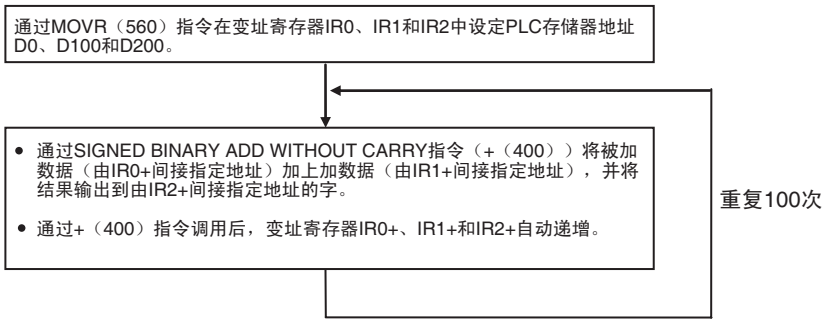
  3. 将 IR 恢复为保存的值（例如 A）。

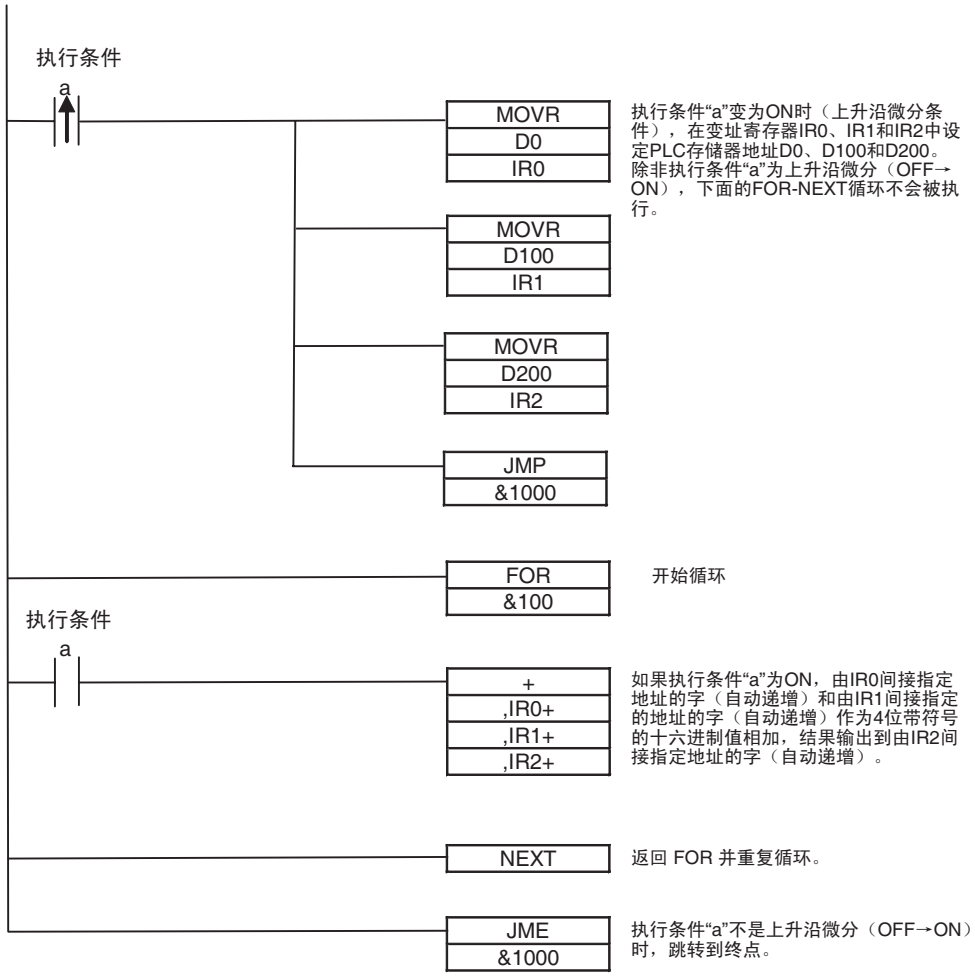


- 必须在使用变址寄存器前设定值。如果在没有设定值时使用变址寄存器，操作将会不稳定。

变址寄存器的应用示例

将 D0 ～ D99 的数据 （被加数据）加上 D100 ～ D199 的数据 （加数据），并且将相加结果输出到 D200 ～ D299 中。通过变址寄存器指定单加法指令的操作数，通过递增变址寄存器和重复地执行加法指令来进行加法操作。





附加信息

通过下表中的指令可对变址寄存器直接指定地址。

● 变址寄存器的直接寻址

对于变址寄存器 IR0 ～ IR15，变址寄存器的长度为每个寄存器 2 字，因此使用双字指令（在助记符中带 “L”）。

指令组	指令名称	助记符	基本功能
数据传送指令	传送到寄存器	MOVR（560）	在变址寄存器中存储 PLC 存储器地址的位或字。
	将定时器 / 计数器 PV 传送到寄存器	MOVRW（561）	
表格数据处理指令	设置记录位置	SETR （635）	
	取得记录编号	GETR （636）	输出存储在变址寄存器中的 PLC 存储器地址。
追踪指令	记录搜索指令	RSRCH □（功能代码：360 ～ 364、370 ～ 374、380 ～ 384）	输出首个与变址寄存器匹配的 PLC 存储器地址。
	记录分类指令	RSORT □（功能代码：203 205）	将一个比分类范围中最近的 PLC 存储器地址高的地址输出到变址寄存器。
数据传送指令	双字传送	MOVL （498）	变址寄存器之间传送。用于交换和比较。
	双字数据交换	XCGL （562）	
比较指令 *1	双精度相等	=L （301）	
	双精度不等	< >L （306）	
	双精度小于	< L （311）	
	双精度小于或等于	< =L （316）	
	双精度大于	>L （321）	
	双精度大于或等于	>=L （326）	
	双精度比较	CMPL （060）	
递增 / 递减指令 *1	双精度二进制递增	++L （591）	通过递增、递减或偏移内容使变址寄存器的 PLC 存储器地址改变。
	双精度二进制递减	--L （593）	
符号（变量）数字指令 *1	双精度有符号无进位加	+L （401）	
	双精度有符号无进位减	-L （411）	
特殊指令	从 CV 转换地址	FRMCV（284）	在 CMV1/CV 系列和 CJ 系列地址之间转换实际 PLC 存储器地址。
	将地址转换到 CV	TOCV （285）	

*1 每个寄存器（IR0 ～ IR15）包含两个字。对它们使用双字指令（在助记符中带 “L”）。

5-7-3 与变址寄存器相关的处理

CJ 系列 CPU 单元数据表格处理指令具有变址寄存器的求补功能。这些指令可大致分成栈处理指令和表格处理指令。

处理			目的	指令
栈处理			FIFO （先进先出）操作或 LIFO （后进先出）数据表格和在数据表格中进行读、写、插入、删除或数据项目计数。	SSET（630）、PUSH（632）、 FIFO（633）、 LIFO （634）、 SREAD（639）、 SWRITE （640）、 SINS（641）、 SDEL （642）、SNUM（638）
表格处理	一字记录表格（范围指令）	基本处理	找出整数或浮点数的十进制值，例如最大值、最小值或校验和。	MIN （183）、 MAX （182）、 MINL（175）、 MAXL （174）、 MINF（177）、 MAXDF （176）、 MIND（179）、 MAXD （178）、 SRCH（181）、 SUM （184）和 FCS（180）
		特殊处理	执行其它各种表格处理，例如比较或分类。	将下列指令与变址寄存器共用，如：SRCH （181）、 MAX （182）、 MIN（183）和比较指令
	多字记录表格		在由用户注册的指定区（字数和范围）搜索和分类记录。	RSRCH < （360）、 RSRCH <=（361）、 RSRCH =、（362）、RSRCH >= （363）、 RSRCH >（364）、RSRCH2 < （370）、RSRCH2 <= （371）、RSRCH2 =（372）、 RSRCH2 >= （373）、RSRCH2 > （374）、RSRCH4 <（380）、 RSRCH4 <= （381）、RSRCH4 = （382）、 RSRCH4 >=（383）、 RSRCH4 > （384）、RSORT（204）、RSORT2（204）和RSORT （205）
			在由用户注册的指定区（字数和范围）执行记录的操作。	将下列指令与变址寄存器共用。 • DIM （631）、 SETR （635）和 GETR （636）指令 • 其他指令 （例如比较指令）

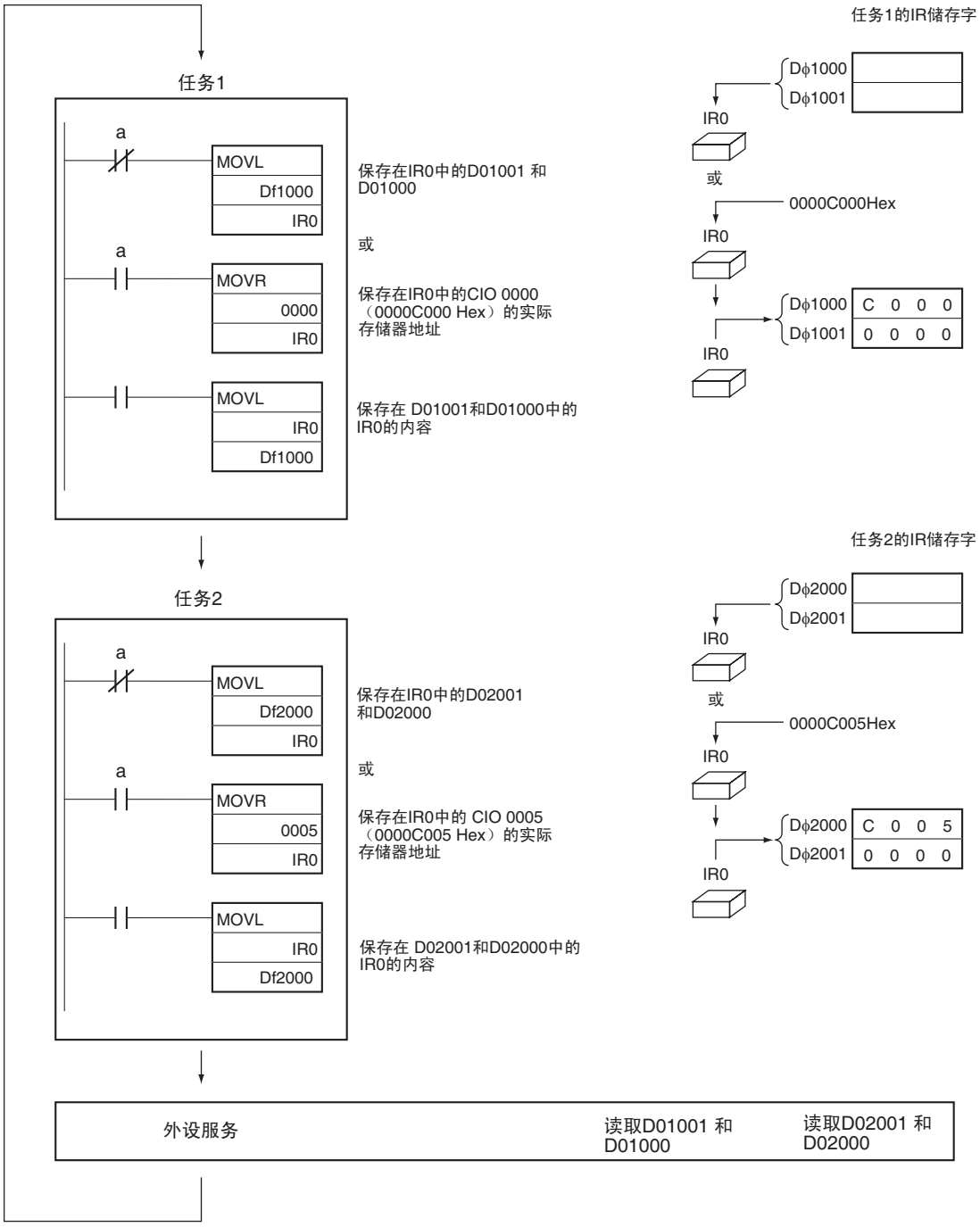
5-7-4 监控变址寄存器

可监控变址寄存器如下：

- 使用 CX-Programmer 监控用于每个任务的变址寄存器的最终值。
- 为了使用上位链接（Host Link）命令或 FINS 命令监控变址寄存器，需写入程序将变址寄存器值从一个任务保存到另一个区（如 DM 区）的任务结束处，并在每个任务开始处从储存字（如 DM 区）读取变址寄存器值。在其他区中 （如 DM 区）保存的每个任务的值可以通过使用 CX-Programmer、上位链接（Host Link）命令或 FINS 命令被编辑。

● 示例

注　　确保在变址寄存器中使用 PLC 存储器地址。


任务1
a
MOVL
Df1000
IR0
保存在IR0中的D01001 和D01000
或
a
MOVR
0000
IR0
保存在IR0中的CIO 0000（0000C000 Hex）的实际存储器地址
MOVL
IR0
Df1000
保存在 D01001和D01000中的IR0的内容
任务2
a
MOVL
Df2000
IR0
保存在IR0中的D02001和D02000
或
a
MOVR
0005
IR0
保存在IR0中的 CIO 0005（0000C005 Hex）的实际存储器地址
MOVL
IR0
Df2000
保存在 D02001和D02000中的IR0的内容
外设服务
读取D01001 和D01000
读取D02001 和D02000
任务1的IR储存字
Dϕ1000
Dϕ1001
IR0
或
0000C000Hex
C 0 0 0
0 0 0 0
任务2的IR储存字
Dϕ2000
Dϕ2001
0000C005Hex
C 0 0 5
0 0 0 0

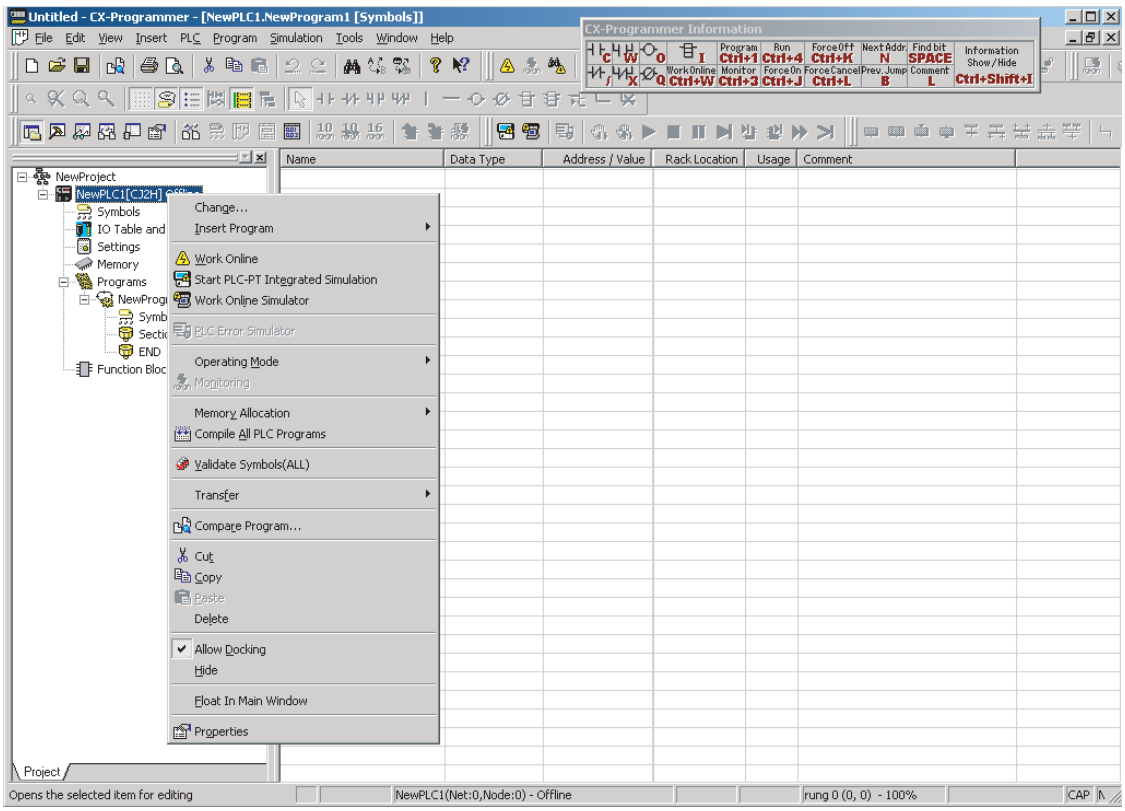
5-7-5 在任务之间共享变址寄存器和数据寄存器

在任务之间可以共享变址寄存器和数据寄存器（IR/DR）。常规设置是给各任务分配单独的寄存器。当前的设置可以在 A99.14 中确认。

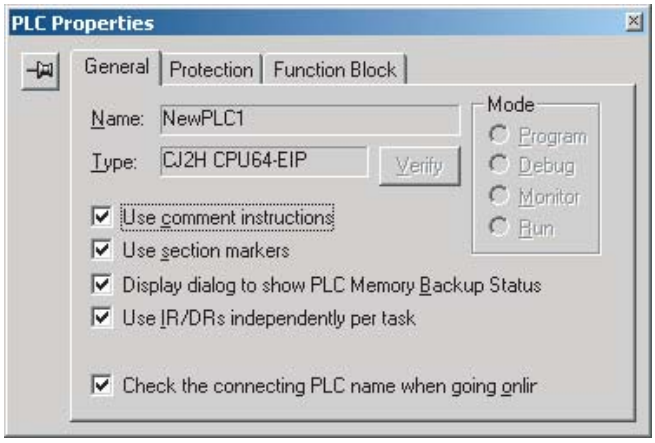
● 设置方法

使用 CX-Programmer 设置共享变址和数据寄存器。不能用手持编程器设置。

1. 在 CX-Programmer 工程树中选择 PLC，点击鼠标右键。



2. 选择 “Properties”（属性）。将显示以下的 “PLC Properties”（PLC 属性）对话框。



如果对每一项任务需要单独的变址和数据寄存器，在 "Use IR/DRs independently per task" 选框中打勾。所有任务使用共享变址和数据寄存器时，将打勾记号去掉。

● 辅助区标志和字

名称	地址	说明
任务之间的 IR/DR 操作	A99.14	说明任务之间变址和数据寄存器是否共享。 0：每一项任务使用单独的寄存器 （默认） 1：所有任务共享寄存器



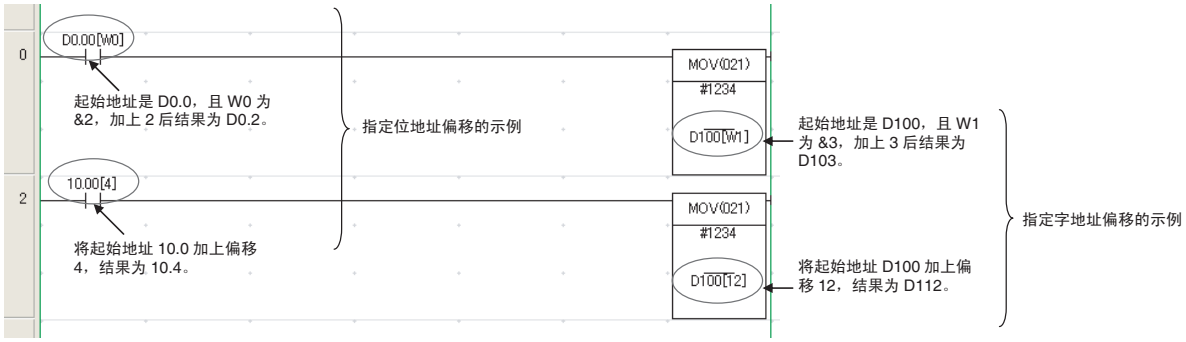
附加信息

- 在两个或多个任务之间如果需要相同内容，可使用共享变址寄存器和数据寄存器消除在任务之间存取寄存器内容的需求。关于存取寄存器内容的信息，请参考 CS 系列操作手册 （样本编号 W339）或 CJ 系列操作手册 （样本编号 W393）中关于变址寄存器的章节。
- 当变址和数据寄存器被共享时，任务之间的切换时间将有所加快。如果寄存器没被使用或如果每一项任务对单独的寄存器没有特别的要求，这时推荐设置共享寄存器。

5-8 指定地址偏移

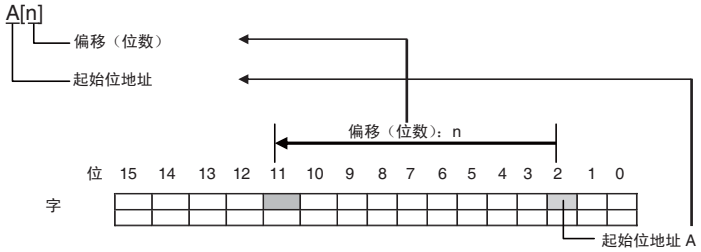
5-8-1 概述

在指令操作数中指定地址时，通过在指定地址后的括号中指定一个偏移可以改变指定的地址。



● 位地址

通过从 A （起始位地址）开始由 n （位数）指定的数表示位地址的偏移。



起始位地址

可以通过位地址或符号 （变量）（除 STRING 或 NUMBER 数据类型外）来指定起始位地址。只能对 H、 W、 DM 和 EM 区的地址进行偏移。 I/O 注释表示的是该起始位地址的 I/O 注释。

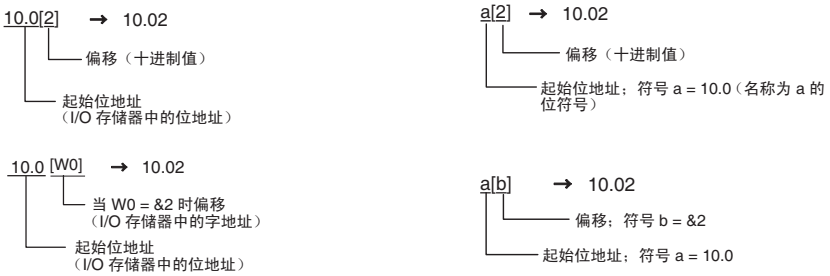
偏移

偏移可以被定义为十进制常数、字地址 （但不能指定 CIO 区地址）或 1 字符号 （变量）（带以下数据类型的符号：INT、 UINT、 WORD、 CHANNEL）。如果指定了字地址，指定字的内容用于偏移。

如果对指定字的偏移超出了位 15，偏移将从下一个字的位 00 继续。如果偏移为间接指定，通过输入比较或其他指令来确认末尾位地址没有超过上限。

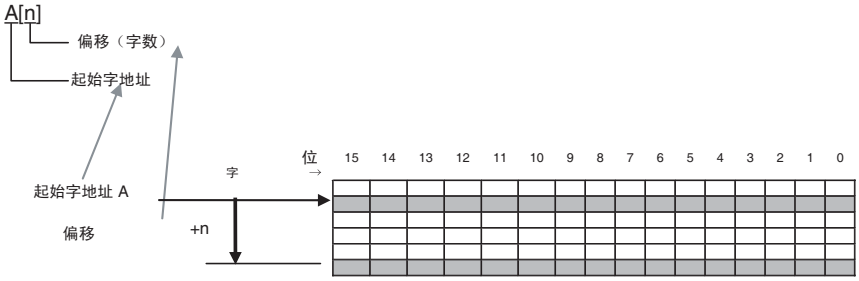
如果偏移位编号超出了起始位地址的存储器区，末尾位地址将在下一个存储器区中，顺序由实际 PLC 存储器地址决定 （关于内存图中区的数组信息，请参考 A-4 PLC 存储器地址的内存印象图）。

例：



● 字地址

通过从 A （起始位地址）开始由 n （偏移字数）指定的数表示字地址的偏移。



起始字地址

可以通过字地址或符号 （符号）（除 STRING 或 NUMBER 数据类型外）来指定起始字地址。只能对 H、 W、 DM 和 EM 区的地址进行偏移。 I/O 注释表示的是该起始字地址的 I/O 注释。

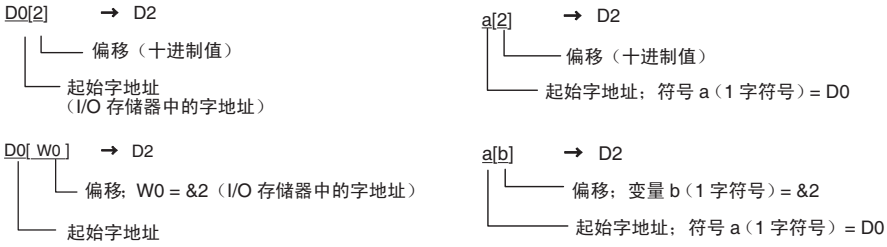
偏移

偏移可以被定义为十进制常数、字地址 （但不能指定 CIO 区地址）或 1 字符号 （符号）（带以下数据类型的符号：INT、 UINT、 WORD、 CHANNEL）。如果指定了字地址，指定字的内容用于偏移。

如果偏移为间接指定，通过输入比较或其他指令来确认末尾位地址没有超过上限。

如果偏移字编号超出了起始位地址的存储器区，末尾字地址将在下一个存储器区中，顺序由实际 PLC 存储器地址决定 （关于内存图中区的数组信息，请参考 A-4 PLC 存储器地址的内存印象图）。

例：



! 注意

如果将符号或存储器地址指定给梯形图中的偏移值，则当通过字地址或符号（变量）间接指定偏移值时，编程须确保起始地址的存储器区没有超出限制范围。

例如，写入程序，通过使用输入比较指令或其他指令，仅当间接指定未使末尾地址超出存储器区时，执行处理。

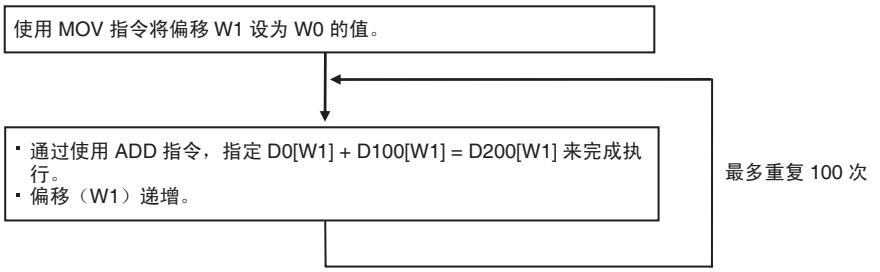
如果间接指定使地址超出起始地址区，系统将访问其他区中的数据，并可能导致意外的操作。

5-8-2 地址偏移应用的示例

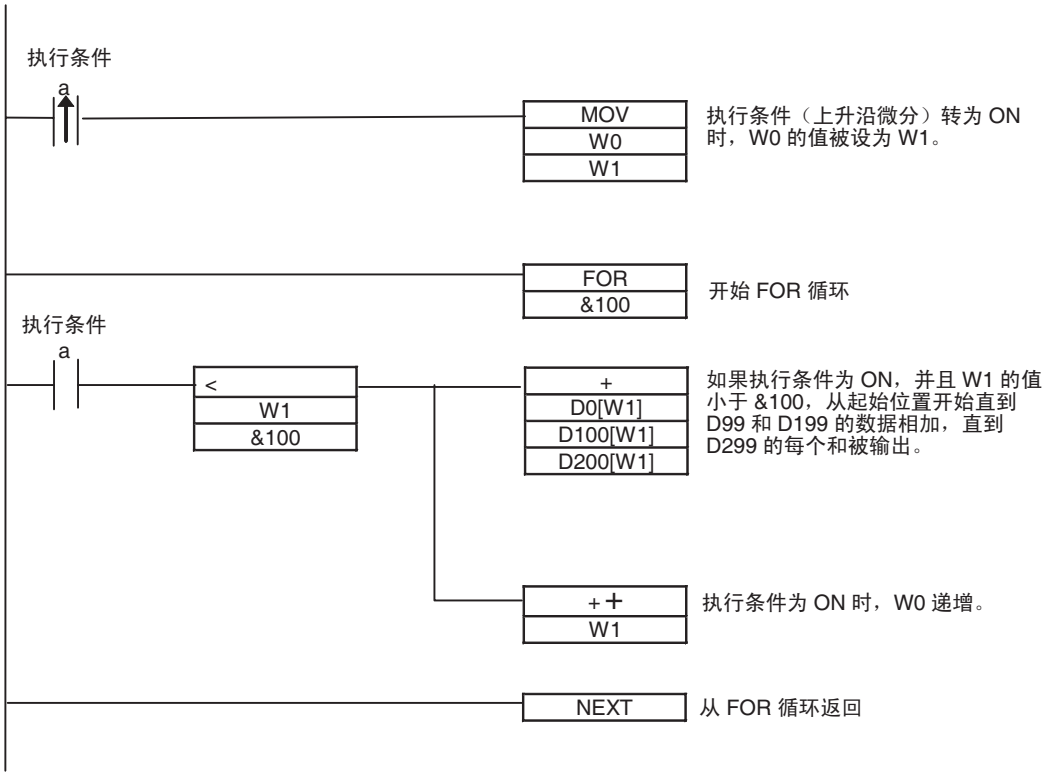
通过在 I/O 存储器中对括号中的偏移指定字地址，可动态地指定偏移。指定字地址的内容将作为偏移使用。例如，通过递增括号中的值，仅使用一条指令就能够使地址增加来完成执行。

● 梯形图程序的示例

本例中，使用了两个连续数据区：D0 ～ D99 和 D100 ～ D199。将对应字的内容从指定起始点 W0 开始到该区的末尾进行相加，它们的和输出到由指定偏移 D200 开始的 D200 ～ D299。例如，如果 W0 为 30，则从 D30 ～ D99 和 D130 ～ D199 的对应字相加，输出的和被输出到 D230 ～ D299。



执行的每个处理由输入比较指令（<）作为执行条件，使 W1 不会超出 &100，确保没有超出间接地址范围的上限。


执行条件
a
MOV
W0
W1
执行条件（上升沿微分）转为 ON 时，W0 的值被设为 W1。
FOR
&100
开始 FOR 循环
执行条件
a
<
W1
&100
+
D0[W1]
D100[W1]
D200[W1]
如果执行条件为 ON，并且 W1 的值小于 &100，从起始位置开始直到 D99 和 D199 的数据相加，直到 D299 的每个和被输出。
++
W1
执行条件为 ON 时，W0 递增。
NEXT
从 FOR 循环返回

5-9 程序的检查

CJ 系列程序可在下列时刻检查。

- 在 CX-Programmer 输入操作时作输入检查；
- 用 CX-Programmer 检查；
- 在程序执行时作指令检查；
- 在程序执行时作致命错误检查 （程序错误）。

5-9-1 CX-Programmer 输入时出错

在下列时刻，CX-Programmer 会自动检查程序。

时刻	检查内容
输入梯形图时	指令输入、操作数输入、编程格式
存储文件时	所有指令的操作数和编程格式
下载文件时	CJ 系列支持的型号和所有指令的操作数
在线编辑时	容量等

检查的结果会在输出窗口的文本标签中给出。同样以梯形图显示的程序会对非法程序部分的左总线用红色显示。

5-9-2 用 CX-Programmer 检查程序

可通过 CX-Programmer 检查用户程序。检查程序时，用户可以指定程序检查的四种等级：A、B、C（以错误的严重程度）或用户检查级。

CX-Programmer 不对指令中间接地址的操作数作范围错误检查。如果一个指令的操作数数据为无效，在程序执行检查时 ER 标志置 ON，相关内容在下章描述。详细信息，请参考 CS/CJ/NSJ 系列指令参考手册（样本编号 W474）。

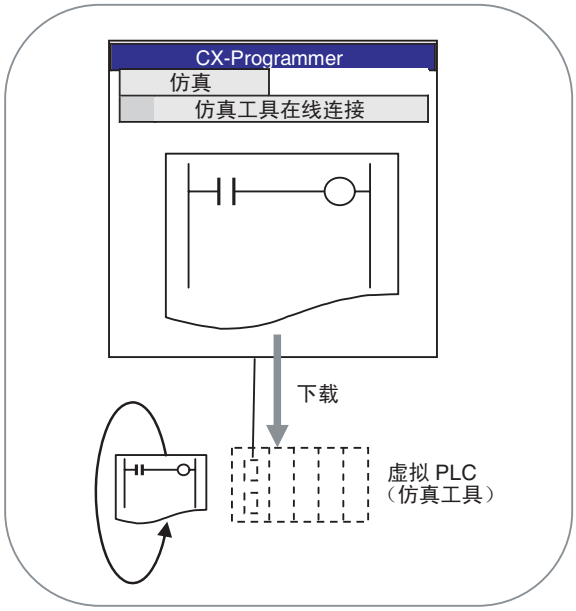
关于 CX-Programmer 检查的详细信息，请参考 CX-Programmer 操作手册 （样本编号 W446）。

5-9-3 用仿真工具调试

可在计算机上仿真 CPU 单元的操作而无须连接到实际 PLC 进行编程。

梯形图程序操作的检查

通过在CX-Programmer的CX-Simulator中启动仿真工具，可由虚拟PLC对已创建的程序进行检查和调试。

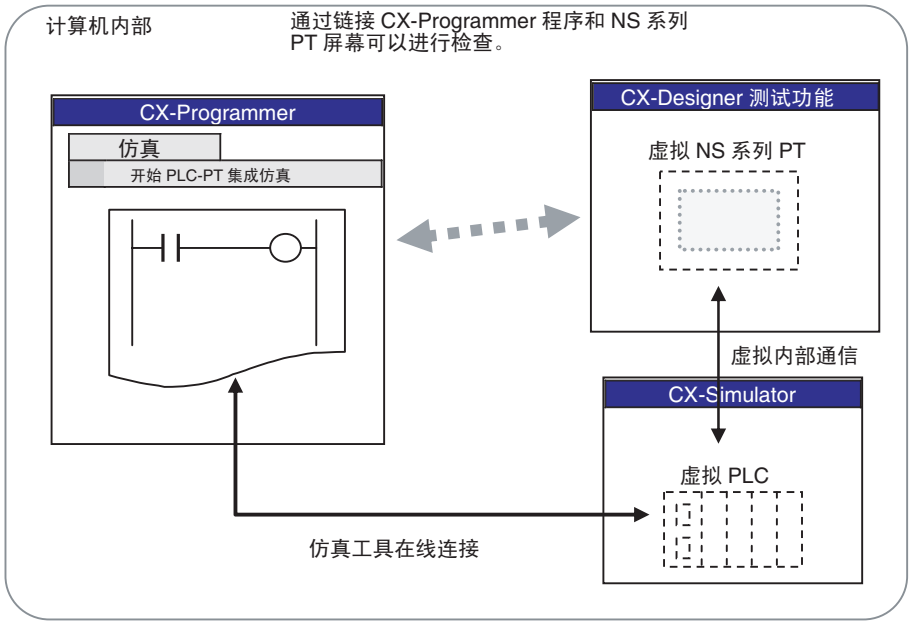


除传送程序和监控外，也可通过仿真工具使用以下功能。关于调试步骤的详细信息，请参考 CX-Programmer 操作手册 （样本编号 W446）。

- 执行步运行、连续步运行或扫描运行。
- 指定断点、开始点和 I/O 断开条件。
- 检查执行数和每个任务的执行时间。
- 模拟中断任务的执行。
- 强制置位和强制复位。

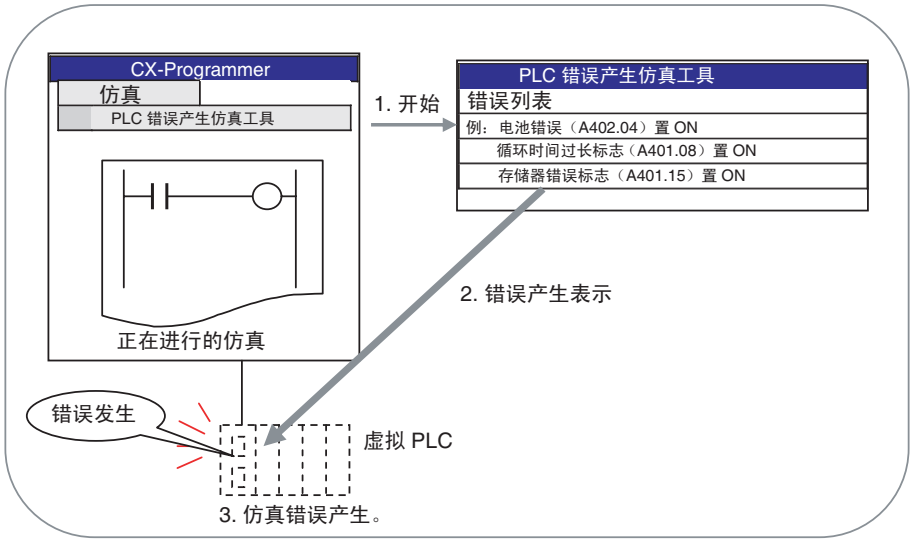
调试 PT 与梯形图程序间的操作：集成仿真

对于 PLC-PT 集成仿真，可以调试 NS 系列 PT 屏幕数据和 CJ 系列 PLC 程序之间的操作。通过链接 CX-programmer 仿真功能和 CX-Designer PT 屏幕设计软件的离线测试功能来实现该功能。仅使用一台计算机而不用通过电缆连接计算机和 PT，就能通过梯形图程序来调试屏幕和屏幕控制。关于调试步骤的信息，请参考 CX-Programmer 操作手册 （样本编号 W446）。



错误仿真功能

通过 CX-Programmer 对梯形图程序仿真期间，在虚拟 PLC 中可能会产生系统错误。当使用特殊操作窗口产生 PLC 系统错误为所需的致命或非致命系统错误时，可简单地检查 NS 系列 PT 的梯形图程序的操作。关于调试步骤的信息，请参考 CX-Programmer 操作手册 （样本编号 W446）。



注　与实际错误不同，即使使用 PLC 错误产生仿真功能生成一个致命错误，梯形图程序的执行将不会停止。

附加信息

也可以通过使用 FAL （006）或 FALS （007）指令产生系统错误。

5-9-4　程序执行检查

程序执行时，可使用 CX-Programmer 执行以下检查。

下列检查是在指令执行时完成。

错误类型	错误标志变为 ON	停止 / 继续运行
指令处理错误	ER 标志 当错误出现，如果指定停止运行，指令处理错误标志（A295.08）也将变为 ON。	在 PLC 设置中有一项设定可对指令处理错误发生时是否停止或继续运行作确定。默认设定作继续运行操作。 仅当确定为停止操作，程序错误标志会产生并停止运行。
存取错误	AER 标志 当错误出现，如果指定停止运行，存取错误标志 （A295.10）将会变为 ON。	在 PLC 设置中有一项设定可对指令处理错误发生时是否停止或继续运行作确定。默认设定作继续运行操作。 仅当确定为停止操作，程序错误标志会产生并停止运行。
非法指令错误	非法指令错误标志 （A295.14）	致命 （程序错误）
用户程序区溢出错误	用户程序区溢出错误标志（A295.15）	致命 （程序错误）

● 指令处理错误 （错误标志 ON 错误）

- 当执行一条指令，如果提供的数据不正确或试图执行一条任务之外的指令，指令处理错误就会出现。这里，在指令处理开始时所需的数据被检查并作为检查结果，该指令不执行。P_ER 标志（错误标志）将变为 ON， P_EQ 和 P_N 标志可能保持不变或变为 OFF，这取决于这条指令。
  如果这条指令 （不包括输入指令）正常结束，通常 P_ER 标志 （错误标志）将变为 OFF。使 P_ER 标志变为 ON 的条件随各指令而不同。关于各指令描述的详细信息，请参考 CS/CJ/NSJ 系列指令参考手册 （样本编号 W474）。
- 假如指令处理错误出现，P_ER 标志变为 ON。在 PLC 设置时设定指令错误停止运行，这时运行将停止（致命错误），并且指令处理出错标志 （A295.08）变为 ON。

● 非法存取错误 （P_AER 标志 ON 错误）

- 当定义指令操作数的地址被存取时，非法存取错误表示以下方式中的一种存取错误区域。
  - 对参数区进行读或写操作。
  - 对未安装的内存区域进行写操作。 *1
  - 对定义为 EM 文件存储器的 EM 区进行写操作。
  - 对只读区进行写操作。
  - 在间接 DM/EM 地址中定义为 BCD 码，但 DM/EM 中的内容不是 BCD 码 （如：*D1 为 #A000）。
- 如果存取错误出现，指令处理仍将继续，并且错误标志（ER标志）不变为ON，但存取错误标志（P_AER 标志）将会变为 ON。

*1 一个存取错误会在下列情况出现：

- 对当前组指定 EM 地址超过 32767 （例如 E32768）。
- 对间接 EM 地址用 BIN 模式，指定字内容为 8000 ～ FFFF Hex（例如：@EC-00001 内容为 #8000），指定末级组（例如：C）。
- 对间接 EM 地址用 BIN 模式，指定字内容为 8000 ～ FFFF Hex（例如：@EC-00001 内容为 #8000），指定当前组（例如：C）。
- 含有一个内部存储器位地址的 IR 寄存器用作字地址，或含有一个内部存储器字地址的 IR 寄存器用作位地址。

- 如果在 PLC 设置中将指令错误设定为停止运行，当一个非法存取错误出现则 AER 标志变为 ON，运行停止（致命错误），且 “非法存取错误标志”（A295.10）会变为 ON。



附加信息

存取错误标志 （P_AER 标志）在一个任务执行后不会被清除。如果指令错误设置为继续运行状态，这个标志可以在 END （001）指令执行前得以监视，以知道在任务程序中是否有一个非法存取错误出现 （如果这个 AER 标志用 CX-Programmer 监视，最终的 P_AER 标志的状态是所有用户程序执行后可检测到的状态）。

● 其它错误

非法指令错误

非法指令错误表示试图执行的指令数据不在系统定义范围内，只要程序是用 CX-Programmer 生成，一般不会出现该错误。

在非常少的情况中，即使出现该错误，也将其视为程序错误对待，运行将会停止 （致命错误），并且非法指令标志 （A295.14）将会变为 ON。

用户程序区溢出错误

用户程序区溢出错误表示试图在定义为程序存储区的用户程序区的最后地址后存放执行指令数据，只要程序是用 CX-Programmer 生成，一般不会出现该错误。

在非常少的情况中，即使出现该错误，也将其视为程序错误对待，运行将会停止（致命错误），且 UM 溢出标志 （A295.15）将会变为 ON。



附加信息

如果错误标志 （P_ER）或存取错误标志 （P_AER）变为 ON，它将被视为程序错误对待，并可用它使 CPU 从运行变为停止。可在 PLC 设置中对程序错误指定操作。

● 程序错误

程序错误	描述	对应标志
无 END 指令	在程序中未出现一条 END 指令	无 END 标志 （A295.11）变为 ON。
任务执行过程中错误	无任务在循环中就绪。 没有程序分配给一个任务。 尽管满足中断任务的指令执行条件，对应的中断任务编号未出现。	任务错误标志 （295.12）变为 ON。
指令处理错误 （ER 标志为 ON）和在 PLC 设置中将指令错误设为停止运行	当试图去执行一条指令时，在操作数中提供了错误的数据值。	如果在 PLC 设置中将指令错误设为停止运行， ER 标志和指令处理错误标志 （A295.08）都变为 ON。

程序错误	描述	对应标志
非法存取错误 （AER 标志为 ON）和在 PLC 设置中将指令错误设为停止运行	对参数区进行读或写操作。 对未安装的存储器区进行读或写操作。 对定义为 EM 文件存储器的 EM 区进行读或写操作。 对只读区进行写操作。 在间接 DM/EM 地址中定义为 BCD 码，但定义的值不是 BCD 码。	如果在 PLC 设置中将指令错误设为停止运行， AER 标志和非法存取错误标志 （A295.10）变为 ON。
间接 DM/EM BCD 码错误和在 PLC 设置中将指令错误设为停止运行	在间接 DM/EM 地址中定义为 BCD 码，但定义的值不是 BCD 码。	在间接 DM/EM 地址中定义为 BCD 码，但定义的值不是 BCD 码。
微分地址溢出错误	在在线编辑过程中，插入或删除了超过 131,071 条微分指令。	微分溢出错误标志 （A295.13）变为 ON。
非法指令错误	试图执行一条不能执行的指令。	UM （用户存储器）溢出标志 （A295.14）变为 ON。
UM （用户存储器）溢出错误	试图在定义为程序存储区的用户程序区的最后地址后存放执行指令数据。	UM （用户存储器）溢出标志 （A295.15）变为 ON。

5-10 注意事项

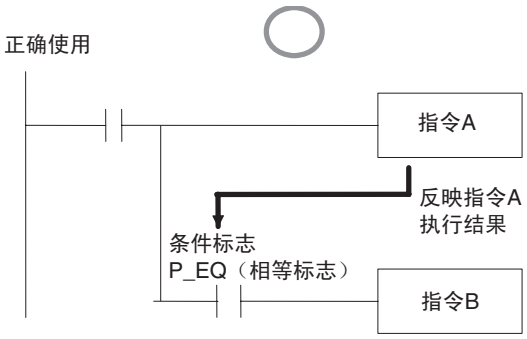
5-10-1 条件标志

使用条件标志

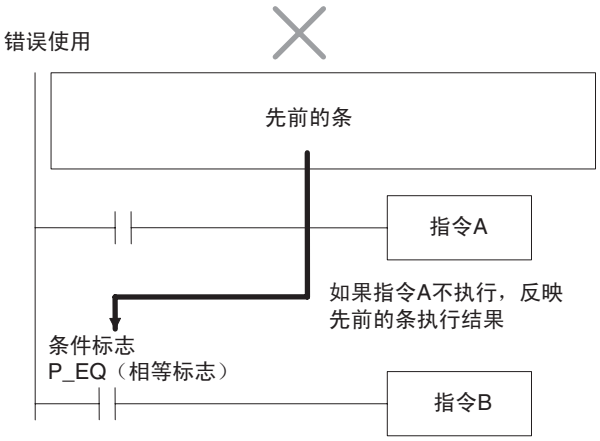
条件标志由所有指令共享。因为它们的状态取决于各指令的执行结果，而在一个循环中经常变化。所以在反映指令执行结果后立即在同一执行条件分支处使用条件标志，绝对不要将条件标志与总线直接相连，因为这样做会导致反映其它指令的执行结果。

例：使用指令 A 执行结果

助记符



相同的执行条件 （a）用于指令 A 和 B，指令 B 执行条件还取决于指令 A 的执行结果。本例中，仅当指令 A 执行后，指令 B 执行取决于指令 A 执行后的条件标志。

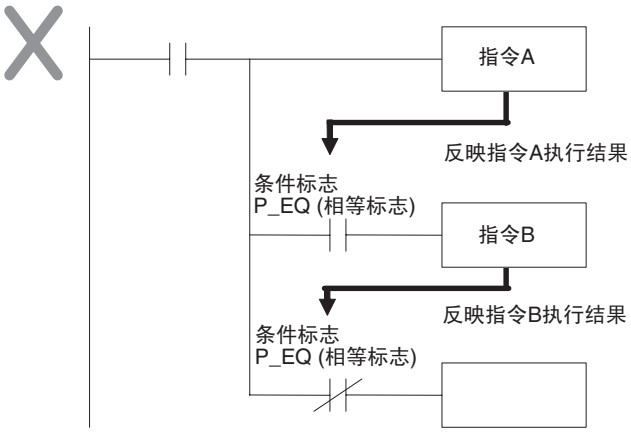


如果条件标志直接与左总线相连，当指令 A 不执行时，指令 B 的执行将取决于前面条的执行结果。

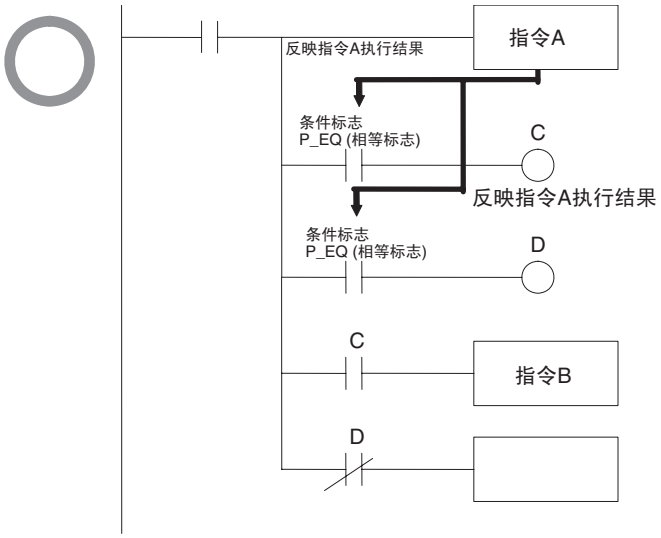
使用中断任务时，即使在循环任务执行期间，当满足开始条件时运行中断任务。在这种情况下，处理中断任务到循环任务的切换时，即使中断任务中的标志有 ON/OFF 切换，条件标志也会恢复到原有状态。

● 使用执行结果的常开 （N.O.）和常闭 （N.C.）输入

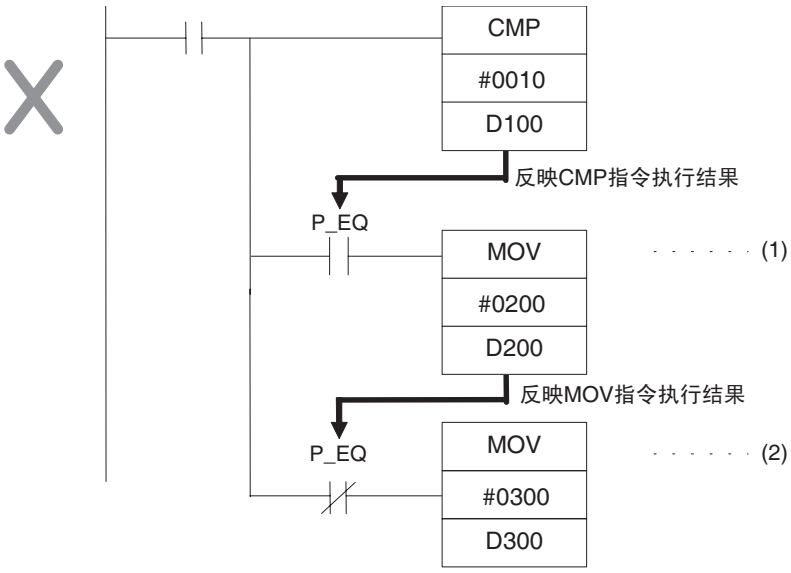
如下所示，尽管输入位的常开和常闭从相同输出分支引出执行，但条件标志将取决于指令 B 的执行结果。



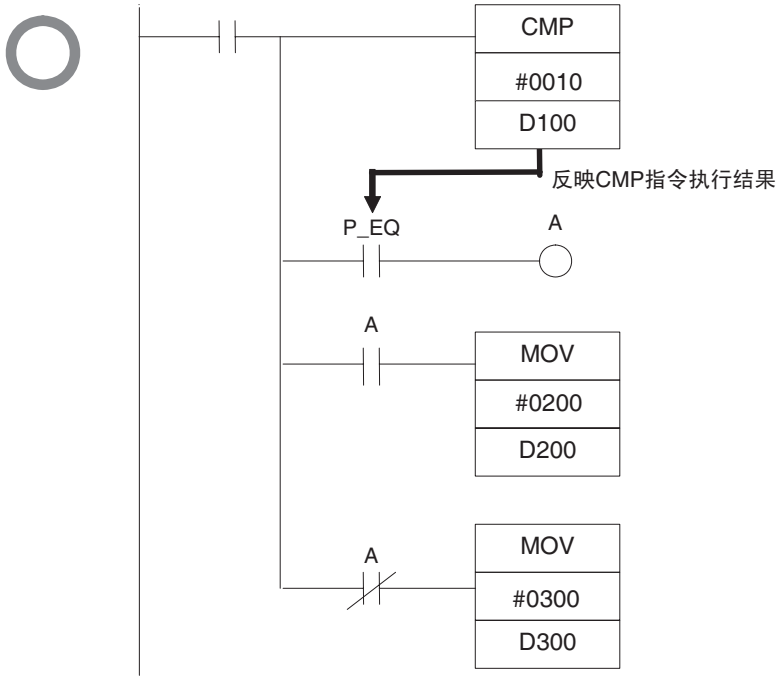
确认每一个结果从输出指令一次获取，保证指令 B 的指令结果不影响条件标志。



例：下列中如果 D100 的内容为 #0010，则将 #0200 传送到 D200 ；如果 D100 内容不为 #0010，则将 #0300 传送到 D300。



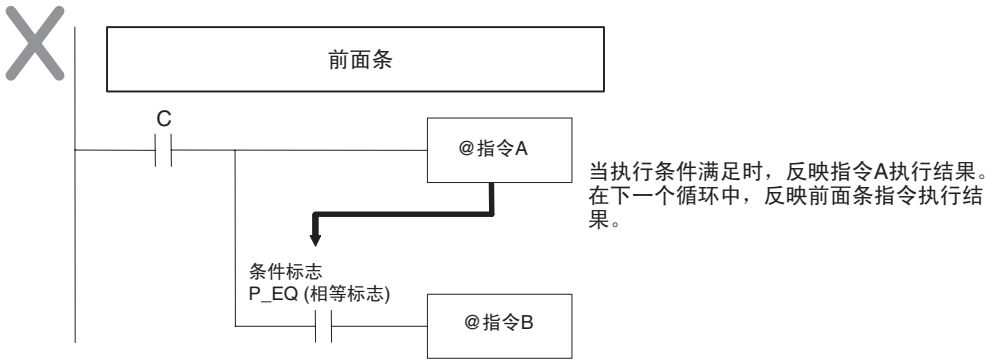
对指令（1），如果在上面条中 D100 的内容为 #0010，相等标志将变为 ON， #0200 将传送到 D200 中去。但由于源数据 #0200 不是 0000 Hex, 则使得等于标志变为 OFF。在（2）中的 MOV 指令将执行把 #0300 传送到 D300 中去的指令。所以，如下所示，应插入一条以防止第一条 MOV 指令的执行结果刷新条件标志。



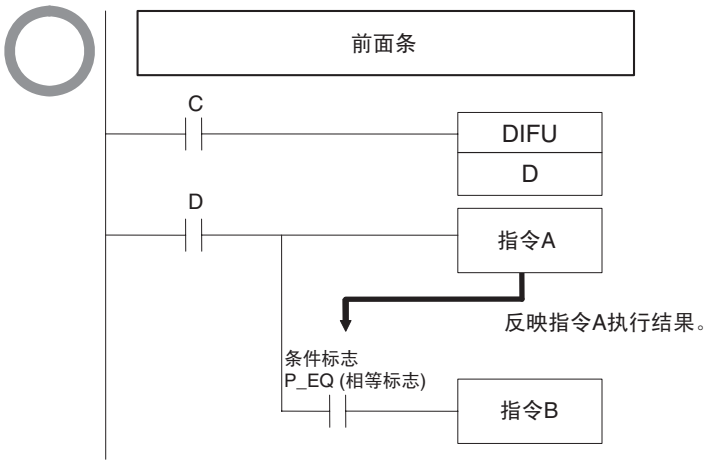
● 使用微分指令执行结果

对于微分指令，仅当执行条件满足，指令的执行结果才会反映在条件标志上，前面条指令执行结果（而不是这微分指令的执行结果）将在下个循环中反映在条件标志上。所以，必须知道如果使用了微分指令执行结果后，下一个循环条件标志会如何变化。

在下列中，仅当执行条件 C 满足，指令 A 和 B 才会执行，但下面的问题将会出现：指令 B 执行时，会取指令 A 的执行结果条件标志状态。如果在指令 A 执行后，执行条件 C 在下一个循环中仍保持 ON 状态，并且由于前面条反映的执行结果由 OFF 变为 ON 时，那么指令 B 就会发生意外的执行 （根据执行条件）。



然而在这种情况下，如下所示指令 A 和 B 不是微分指令，用 DIFU （DIFD）指令来取代。指令 A 和 B 两者都成为上升沿 （或下降沿）微分指令并只执行一个循环。



附加信息

CONDITION FLAG SAVE 和 CONDITION FLAG LOAD （CCS （282）和 CCL （283））指令可用于保存和获取条件标志状态，这两条指令可在一个任务或另一任务中的其它位置访问条件标志的状态。

主要条件变 ON 的条件标志

● 错误标志 （P_ER）

在一些特殊条件下，ER 标志将变为 ON，例如用于指令的操作数出错。当 ER 标志变为 ON，指令将不执行。

当 ER 标志为 ON，其它一些条件标志状态如 <、>、OF、UF 等不再发生变化，而 = 和 N 标志状态将随指令而变化。

参考 CS/CJ/NSJ 系列指令参考手册 （样本编号 W474）对各指令引起 ER 标志变为 ON 的条件说明。需注意有些指令会不管条件而将 ER 标志置为 OFF。

附加信息

PLC 设置可设定当指令出错 ER 为 ON 时，运行是否要停止。在默认设定情况下，当 ER 标志变为 ON，运行将继续。如果定义停止运行，当 ER 标志变为 ON，运行停止 （当程序错误对待），运行停止点的程序地址将存入 A298 到 A299 内。同时，A295.08 变为 ON。

● 相等标志 （P_EQ）

除了当比较结果相等 （=），相等标志对所有指令都是暂时的。它由系统自动设定并会改变。在前一条指令使相等标志变为 ON （OFF）后，相等标志可由下一条指令变为 OFF （ON）。例如，当 MOV 或其它传送指令将 0000 Hex 作为源数据传送，相等标志就会变 ON，而其它所有时候都为 OFF。即使一条指令将相等标志变为 ON，传送指令一执行，相等标志马上会根据传送的源数据是否为 0000 Hex 而使状态变为 ON 或 OFF。

● 进位标志 （P_CY）

进位标志用于移位指令、带进位的加法和减法指令、加法和减法指令的进位和借位以及特殊 I/O 单元指令、 PID 指令和 FPD 指令。请注意下面的注意事项。

- 进位标志可由某些指令的执行结果置 ON（OFF），也可由另一些指令的执行结果置 OFF（ON），请在使用进位标志时确认其反映的结果是需要的。

● 小于和大于标志 （P_LT, P_GT）

小于和 大于标志是用于比较指令以及 LMT、 BAND、 ZONE、 PID 和其它指令中。

即使小于或大于标志由某些指令执行的结果置 ON （OFF），但它也能由其它指令置 OFF （ON）。

● 负标志 （P_N）

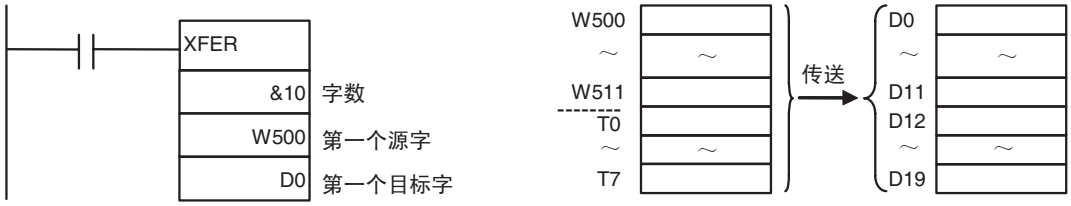
某些指令，当指令执行结果的最高位 “1” 时， N 标志变为 ON，而对其它指令， N 标志变为 OFF 是无条件的。

● 多字的特殊操作数

如果一个操作数定义为多字的，指令将按所定义的执行，即使操作数的所有字不在同一个数据区内。这种情况下，字将按 PLC 内存地址依次存放。错误标志不会变为 ON。

如例子所示，如果 20 个字传送到以 W500 开始的内存，工作区用 XFER （070）执行一个块传送功能。这里工作区的结束地址为 W511，传送将会超出范围，然而出错标志不会为 ON，指令照样执行。在 PLC 内存中，定时器当前值存放在工作区后的内存里，这样对下面的指令传送到 W500 ～ W511 的内容将会转到 D0 到 D11，T0 到 T7 的当前值传送到 D12 到 D19。

注　对 PLC 将特殊内存地址，参考附录 PLC 存储器地址的内存印象图。



5-10-2 特殊程序段

CJ 系列程序具有控制指令条件的特殊程序段。下面的特殊程序段是可用的。

程序段	指令	指令条件	状态
子程序	SBS、 SBN 和 RET 指令 GSBS、 GSBN 和 GRET 指令	子程序执行	执行在 SBN 和 RET 指令间的程序段
IL - ILC 段	IL 和 ILC 指令	程序段联锁	输出位变为 OFF，定时器复位。其它指令不执行，且以前状态保持。
步进梯形图段	STEP 指令		

程序段	指令	指令条件	状态
FOR-NEXT 循环	FOR 指令和 NEXT 指令	程序断开	循环
JMP0 - JME0 段	JMP0 指令和 JME0 指令		跳转
块程序段	BPRG 指令和 BEND 指令	块程序执行	执行在 BPRG 和 BEND 指令间助记符列出的块程序

指令组合

下表中给出的特殊指令可用在其它程序段中。

	子程序	IL - ILC 段	步进梯形图段	FOR - NEXT 循环	JMP0 - JME0 段	块程序段
子程序	不可以	不可以	不可以	不可以	不可以	不可以
IL - ILC	可以	不可以	不可以	可以	可以	不可以
步进梯形图段	不可以	可以	不可以	不可以	可以	不可以
FOR - NEXT 循环	可以	可以	不可以	可以	可以	不可以
JMP0 - JME0	可以	可以	不可以	不可以	不可以	不可以
块程序段	可以	可以	可以	不可以	可以	不可以

注　指定程序区域的指令不能用于其它任务的程序中。详细信息，请参考第 5-2-4 任务设计。

子程序

把所有的子程序都放在所有程序的 END （001）指令前，但子程序在其他程序之后 （而且，任一子程序不能放在步进梯形图、块程序、FOR-NEXT 或 JMP0-JME0 段中）。如果将不是子程序的程序放在子程序（SBN ～ RET）之后，这段程序将不被执行。



子程序中不允许使用的指令

以下指令不允许用于子程序中。

功能	助记符	指令
步进控制处理	STEP （008）	定义步进梯形图
	SNXT （009）	通过步进梯形图

注　块程序段
子程序可包含块程序段。当执行从子程序返回主程序时，如果块程序处于等待状态，在下次调用时，块程序段仍保持等待状态。

步进梯形图程序段中不允许使用的指令

以下指令不允许用于步进梯形图程序段中。

功能	助记符	指令
顺序控制	FOR （512）、 NEXT （513）和 BREAK （514）	FOR、 NEXT 和中断循环
	END （001）	END
	IL （002）和 ILC （003）	联锁和联锁清除
	JMP （004）和 JME （005）	跳转和跳转结束
	CJP （510）和 CJPN （511）	条件跳转和非条件跳转
	JMP0 （515）和 JME0 （516）	多路跳转和多路跳转结束
子程序	SBN （092）、RET （093）、GSBN （751）和 GRET （752）	子程序进入和子程序返回
块程序	IF （802）（NOT）、ELSE （803）、和 IEND （804）	块条件分支 （NOT）、非块条件分支，块条件分支结束
	BPRG （096）和 BEND （801）	块程序开始 / 结束
	EXIT （806） （NOT）	块条件退出 （非）
	LOOP （809）和 LEND （810） （NOT）	循环块和循环块结束 （非）
	WAIT （805） （NOT）	一个循环等待 （非）
	TIMW （813）	定时器等待
	TMHW （815）	高速定时器等待
	CNTW （814）	计数器等待
	BPPS （811）和 BPRS （812）	块程序暂停和重新启动

注　1　步进梯形图程序可用在联锁段 （在 IL 和 ILC 之间）内。当联锁为 ON 时，步进梯形图将被完全复位。
2　步进梯形图程序可用在多路跳转 （JMP0）和多路跳转结束 （JME0）间。

块程序段中不允许使用的指令

以下指令不允许用于块程序段中。

按功能分类	助记符	指令
顺序控制	FOR （512）、 NEXT （513）和 BREAK （514）	FOR、 NEXT 和中断循环
	IL （002）和 ILC （003）	联锁和联锁清除
	JMP0 （515）和 JME0 （516）	多路跳转和多路跳转结束
	END （001）	结束
顺序输入	UP （521）	条件 ON
	DOWN （522）	条件 OFF

按功能分类	助记符	指令
顺序输出	DIFU	上升沿微分
	DIFD	下降沿微分
	KEEP	保持
	OUT	输出
	OUT NOT	输出 （非）
定时器 / 计数器	TIM	100 ms 定时器
	TIMH	10 ms 定时器
	TMHH （540）	1 ms 定时器
	TIMU	0.1ms 定时器
	TMUH	0.01ms 定时器
	TTIM （087）	累加定时器
	TIML （542）	长定时器
	MTIM （543）	多路输出定时器
	CNT	计数器
	CNTR	可逆计数器
子程序	SBN （092）和 RET （093）	子程序进入和子程序返回
数据移位	SFT	移位
梯形图步进控制	STEP （008）和 SNXT （009）	步定义和步开始
数据控制	PID	PID 控制
块程序	BPRG （096）	块程序开始
故障诊断	FPD （269）	故障点检测

注　1 块程序可用于步进梯形图程序段中。

2 一个块程序可用于联锁段 （IL 和 ILC 间）中。当联锁为 ON 时，块程序将不执行。

3 一个块程序可用于多路跳转 （JMP0）和多路跳转结束 （JME0）指令间。

4 跳转指令 （JMP）和条件跳转指令 （CJP/CJPN）可用于块程序段中。但跳转 （JMP）和跳转结束 （JME）指令及条件跳转 （CJP/CJPN）和跳转结束 （JME）指令在块程序中必须成对使用，不允许单独使用。

6

I/O 存储器区

本章描述了 CPU 单元中的 I/O 存储器区。I/O 存储器是内置于 CPU 单元的一种存储器。

6-1 I/O 存储器区 6-2
6-1-1 I/O 存储器区概述 6-2
6-1-2 I/O 存储器区结构 6-4
6-1-3 保持 I/O 存储器值 6-5
6-2 I/O 区 6-7
6-2-1 输入位 6-7
6-2-2 输出位 6-9
6-3 数据链接区 6-12
6-4 CPU 总线单元区 6-13
6-5 特殊 I/O 单元区 6-14
6-6 DeviceNet 区 6-15
6-7 工作区 6-16
6-8 保持区 6-17
6-9 辅助区 6-19
6-10 暂存继电器区 6-20
6-11 数据存储器区 6-21
6-12 扩展数据存储器区 6-23
6-13 定时器区 6-25
6-14 计数器区 6-27
6-15 任务标志 6-28
6-16 变址寄存器 6-29
6-17 数据寄存器 6-34
6-18 条件标志 6-36
6-19 时针脉冲 6-38

6-1 I/O 存储器区

6-1-1 I/O 存储器区概述

可以使用指令操作数访问 I/O 存储器区。以下表格列出了 I/O 存储器中的区。

区名称		描述	参考
CIO 区 （核心 I/O 区）		CIO 区中的字用于数据交换，例如各种单元的 I/O 刷新。没有分配到单元的字可在程序中作为工作字和工作位使用。在 CIO 中指定地址时，无须输入 “CIO” 前缀。	---
	I/O 区	I/O 区中的字被分配给基本 I/O 单元的外部 I/O 端子。	6-2
	数据链接区	数据链接区中的字通过 Controller Link 单元用于网络中其他 PLC 单元的数据链接。	6-3
	CPU 总线单元区	CPU 总线单元区中的字被分配到 CPU 总线单元，用于传送状态信息。	6-4
	特殊 I/O 单元区	特殊 I/O 单元区中的字被分配到特殊 I/O 单元，用于传送状态信息。	6-5
	DeviceNet 区	DeviceNet 区中的字分配给从站，用于 DeviceNet 远程 I/O 通信。该区中的分配是固定的并且不能被改变。	6-6
	内部 I/O 区 *1	这些字只能在程序中使用。不能用于与外部 I/O 端子的 I/O 交换。	---
工作区 （W）*1		在工作区中的字只能在程序中使用。	6-7
保持区 （H）		保持区内的字只能在程序中使用，这些字的内容在 PLC 接通或者运行模式在 PROGRAM 和 RUN /MONITOR 模式之间切换时保持不变。	6-8
辅助区 （A）		辅助区包含监控和控制 PLC 操作的标志和控制位。	6-9
暂存继电器区 （TR）（TR 区）		TR 区域包含记录程序分支的 ON/OFF 状态的位。 TR 位只用作助记符。	6-10
数据存储器区 （D）（DM 区）		DM 区是多用途数据区。对于 CJ2 PLC， DM 区只能以字或位为单位被读写。 当 PLC 接通或运行模式在 PROGRAM 模式和 RUN /MONITOR 模式间切换时，DM 区字的内容保持不变。 DM 区中的一些字用于保存特殊 I/O 单元和 CPU 总线单元的初始设定。	6-11
扩展数据存储器区 （E）（EM 区）		EM 区域是多用途数据区。对于 CJ2 PLC， EM 区只能以字或位为单位被读写。 当 PLC 接通或运行模式在 PROGRAM 模式和 RUN /MONITOR 模式间切换时，EM 区字的内容保持不变。 EM 区被分为组。	6-12
定时器区	定时器完成标志区 （T）	当定时器设定的时间到达时，完成标志位转为 ON。	6-13
	定时器 PV 区（T）	定时器的 PV 增加或减少，用于定时器运行。	
计数器区	计数器完成标志区 （C）	当计数器达到设定值或计数为零时，完成标志位转为 ON。	6-14
	计数器 PV 区（C）	计数器的 PV 增加或减少，用于计数器运行。	

区名称	描述	参考
任务标志区 （TK）	对应循环任务在 “就绪”状态时，任务标志为 ON。	6-15
变址寄存器 （IR）	变址寄存器用于保存 PLC 存储器的地址 （在 RAM 区中的绝对存储器地址），用来间接寻址 I/O 存储器内的字。	6-16
数据寄存器 （DR）	间接寻址字时，数据寄存器用于偏移变址寄存器内的 PLC 存储器地址。	6-17
条件标志	条件标志是特殊标志，例如错误标志和进位标志，用于显示执行指令的结果。	6-18
时钟脉冲	时钟脉冲是按一定时间间隔转 ON 和 OFF 的特殊标志。	6-19

*1 有两个区提供工作位：CIO 区中的内部 I/O 区和工作区。首先在工作区中使用工作位。

6-1-2 I/O 存储器区结构

区域		大小	范围	外部 I/O 分配	位存取	字存取	存取		从编程设备修改	启动时或模式改变时的状态	强制位状态
							读	写			
CIO 区	I/O 区	2,560 位（160 字）	CIO 0 ～ CIO 159*1	基本 I/O 单元	OK	OK	OK	OK	OK	清除 *2	OK
	数据链接区	3,200 位（200 字）	CIO 1000 ～ CIO 1199	数据链接或 PLC 链接 （条件）	OK	OK	OK	OK	OK		OK
	CPU 总线单元区	6,400 位（400 字）	CIO 1500 ～ CIO 1899	CPU 总线单元（条件）	OK	OK	OK	OK	OK		OK
	特殊 I/O 单元区	15,360 位（960 字）	CIO 2000 ～ CIO 2959	特殊 I/O 单元（条件）	OK	OK	OK	OK	OK		OK
	DeviceNet 区	9,600 位（600 字）	CIO 3200 ～ CIO 3799	DeviceNet 主站（固定分配）（条件）	OK	OK	OK	OK	OK		OK
	内部 I/O 区	3,200 位（200 字） 37,504 位（2,344 字）	CIO 1300 ～ CIO 1499 CIO 3800 ～ CIO 6143	---	OK	OK	OK	OK	OK		OK
工作区		8,192 位（512 字）	W000 ～ W511	---	OK	OK	OK	OK	OK	清除 *2	OK
保持区 *3		8,192 位（512 字）	H000 ～ H511	---	OK	OK	OK	OK	OK	保持	OK
辅助区		48,128 位（3,008 字）	A000 ～ A447	---	OK	OK	OK	No	No	取决于地址	No
			A448 ～ A959	---	OK	OK	OK	OK	OK		
			A960 ～ A1471	---	OK	OK	OK	No	No		
			A10000 ～ A11535	---	OK	OK	OK	No	No		
TR 区		16 位	TR0 ～ TR15	---	OK	---	OK	OK	No	清除	No
DM 区		32,768 字	D00000 ～ D32767	---	OK	OK	OK	OK	OK	保持	No
EM 区		32,768 字 每组，25 组最大 （0 ～ 18 hex）	E00_0 ～ E18_32767	---	OK	OK	OK	OK	OK	保持	某些位可用 *7
定时器完成标志		4,096 位	T0 ～ T4095	---	OK	---	OK	OK	OK	清除	OK
计数器完成标志		4,096 位	C0 ～ C4095	---	OK	---	OK	OK	OK	保持	OK
定时器 PV		4,096 字	T0 ～ T4095	---	---	OK	OK	OK	OK	清除	No*4
计数器 PV		4,096 字	C0 ～ C4095	---	---	OK	OK	OK	OK	保持	No*5
任务标志区		128 位	TK000 ～ TK127	---	OK	---	OK	No	No	清除	No
变址寄存器 *6		16 寄存器	IR0 ～ IR15	---	OK	OK	仅间接寻址	仅用于特殊指令	No	清除	No
数据寄存器 *6		16 寄存器	DR0 ～ DR15	---	No	OK	OK	OK	No	清除	No

*1 通过修改分配给指定单元的首字可将 I/O 区扩展到 CIO 0160 ～ CIO 0999。可使用 CX-Programmer 进行首字的设定，以在 I/O 表中设定首字。首字的设定范围是 CIO 0 ～ CIO 900。

*2 如果 I/O 存储器保持标志 （A500.12）为 ON，当运行模式改变时，存储器的值将被保持。此外，如果 PLC 设置被设为在启动时保持 I/O 存储器保持标志的状态 （IOM 保持位参数），电源打开时，存储器的值将被保持。

*3 H512 ～ H1535 只能设为用于功能块存储器或 SFC 存储器。

*4 通过强制置位 / 复位定时器完成标志可以间接更新定时器的当前值。

*5 通过强制置位 / 复位计数器完成标志可以间接更新计数器的当前值。

*6 变址寄存器和数据寄存器可以由一个任务单独使用，也可以由所有任务共同使用。

*7 如果以下字用于自动地址分配，它们可在 EM 区中被强制置位 / 复位。

CJ2H-CPU64/65-EIP：E03_00000 ～ E03_32767
CJ2H-CPU65-EIP： E06_00000 ～ E09_32767
CJ2H-CPU67-EIP： E07_00000 ～ E0E_32767
CJ2H-CPU68-EIP： E11_00000 ～ E18_32767

6-1-3 保持 I/O 存储器值

CPU 单元操作状态改变时 I/O 存储器的清除

当 CPU 单元的操作状态改变时， I/O 存储器值 （除某些 I/O 存储器区外）被清除。

● 以下改变会使 I/O 存储器被清除

- 在 PROGRAM 模式和 RUN 或 MONITOR 模式之间改变运行模式
- 由于致命错误导致的运行停止 （由于 FALS （007）指令执行导致运行停止的情况除外）
- 复位电源 （关闭电源然后再打开）

● 保持 I/O 存储器的设定

即使 CPU 单元的操作状态发生改变，也能通过设定来保持 I/O 存储器。

- 保持 I/O 存储器用于运行模式改变和致命错误：
  将辅助区中的 IOM 保持位 （A500.12）置 ON。
- 电源复位时保持 I/O 存储器
  将辅助区中的 IOM 保持位（A500.12）置 ON，并在 PLC 设置启动标签页的启动保持区中选择 IOM 保持位。

安全使用注意事项

进行保持 I/O 区的设定

当 CPU 单元的运行模式从 RUN 或 MONITOR 模式变为 PROGRAM 模式时， I/O 存储器中的输出位将不会被清除（不会变为 OFF）。变为 PROGRAM 模式前的状态将被保持。当模式从 PROGRAM 模式变为 RUN 或 MONITOR 模式时，将输出原有的 I/O 存储器值。

由于致命错误（包括由于 FALS（007）指令的执行）导致运行停止时， CPU 单元中的 I/O 存储器值将被保持，但输出单元的输出将全部变为 OFF。

● 根据 I/O 存储器区的 I/O 存储器保持状态

以下表格给出了每个 I/O 存储器区的保持状态。

区		CPU 单元运行模式改变	致命错误		电源变为 ON
			FALS（007）执行	其他致命错误	
CIO 区（核心 I/O 区）	I/O 区	可能保持 *1（取决于设定）	保持	可能保持 *1（取决于设定）	可能保持 *2（取决于设定）
	数据链接区				
	CPU 总线单元区				
	特殊 I/O 单元区				
	DeviceNet 区				
	内部 I/O 区				
工作区 （W）		可能保持 *1（取决于设定）	保持	可能保持 *1（取决于设定）	可能保持 *2（取决于设定）
保持区 （H）		保持			
辅助区 （A）		保持或清除取决于地址			
数据存储器区 （D）		保持			
扩展数据存储器区 （E）		保持			
定时器完成标志 （T）		可能保持 *1（取决于设定）	保持	可能保持 *1（取决于设定）	可能保持 *2（取决于设定）
定时器 PV （T）		可能保持 *1（取决于设定）	保持	可能保持 *1（取决于设定）	可能保持 *2（取决于设定）
计数器完成标志 （C）		保持			
计数器 PV （C）		保持			
任务标志区 （TK）		清除	保持	清除	清除
变址寄存器 （IR）		清除	保持	清除	清除
数据寄存器 （DR）		清除	保持	清除	清除

*1 将辅助区中的 IOM 保持位 （A500.12）置 ON 来保持这些区。

*2 将辅助区中的 IOM 保持位 （A500.12）置 ON，并在 PLC 设置启动标签页的启动保持区中选择 IOM 保持位检查框。

6-2 I/O 区

I/O 区的地址范围为字 CIO 0 ～ CIO 159，位 CIO 0.00 ～ CIO 159.15。通过改变分配到特殊单元的首字，可以将此地址范围扩展为 CIO 0000 ～ CIO 0999。即使扩展了 I/O 区，但能够分配给外部 I/O 最大位数仍是 2,560 个 （160 个字）。

I/O 区里的字被分配给基本 I/O 单元的 I/O 端子。分给基本 I/O 单元的字是根据槽的位置 （从左到右）和每个单元需要的字数。这些字连续地分配，有空槽时跳过。在 I/O 区那些没有被分配给 I/O 基本单元的字只能在程序内使用。

在 I/O 区里的位可以被强制置位和强制复位。

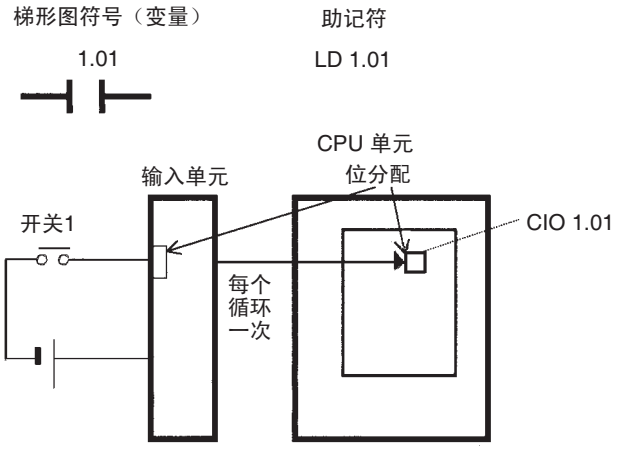
6-2-1 输入位

当将 I/O 区的位分配给输入单元时，I/O 区中的一个位称为一个输入位。输入位反映了设备的 ON/OFF 状态，例如按钮开关、限位开关、光电开关等。

PLC 中有三种刷新输入点状态的方法：正常 I/O 刷新、立即刷新和 IORF （097）刷新。

正常 I/O 刷新

当程序执行完成后，每个循环读入一次外部设备中的输入点的状态。在下面示例中，地址 CIO 1.01 分配给开关 1，一个连接到输入单元的输入端子的外部开关。在每个循环，开关 1 的 ON/OFF 状态都保存到 CIO 1.01。



立即刷新

通过在一条指令前输入一个感叹号，指定指令变量的立即刷新，这个指令的操作数应是一个输入位或输入字，刷新的执行如下所示。这个立即刷新是在每循环一次的正常 I/O 刷新以外执行的刷新。

● 位操作数

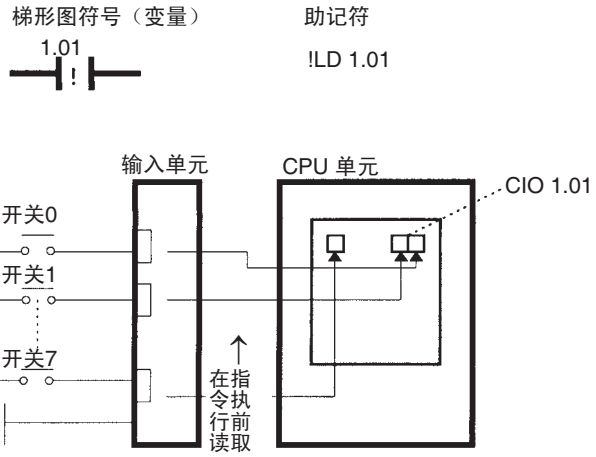
恰好在这条指令执行前，分配给包含指定位的那个字对应的 16 点 I/O 的 ON/OFF 状态被读入 PLC。

● 字操作数

恰好在这条指令执行前，分配给包含指定字的 16 个 I/O 点的 ON/OFF 状态被读入 PLC。

● 示例

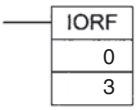
在下面例子中， CIO 1.01 分配给开关 1，该开关是一个连接到输入单元的输入端子的外部开关。恰好在指令 !LD 1.01 执行前读取开关 1 的 ON/OFF 状态并映射到 CIO 1.01 中。



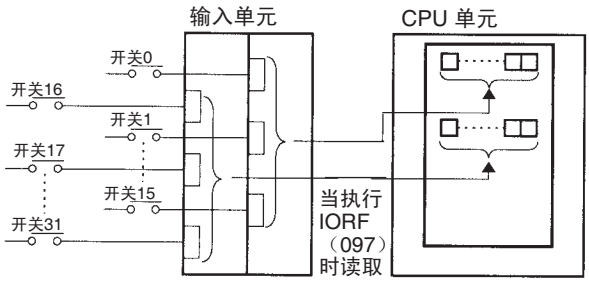
IORF （097）刷新

执行 IORF （097）（I/O 刷新）时，刷新指定范围内字的输入位，这种 I/O 刷新的方法是在每循环一次的正常 I/O 刷新以外执行的刷新。

下面的 IORF （097）指令，刷新在 I/O 区的从字 CIO 0 ～ CIO 3 所有 I/O 点的状态。从输入单元读取输入点的状态，并将输出位的状态写入到输出单元。



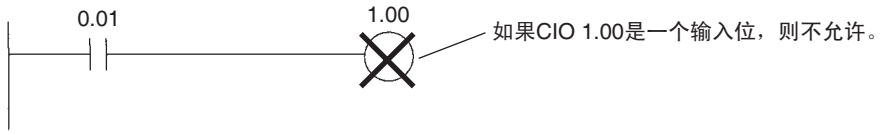
在下面例子中，从输入单元读取分配给 CIO 0 和 CIO 1 的输入点状态。（分配 CIO 2 和 CIO 3 到输出单元。）



输入位的限制

一个输入位在程序中可以用作常开条件和常闭条件，使用次数没有限制，可以以任意次序编程它的地址。

一个输入位不能用作输出指令的操作数。



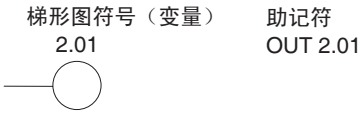
6-2-2 输出位

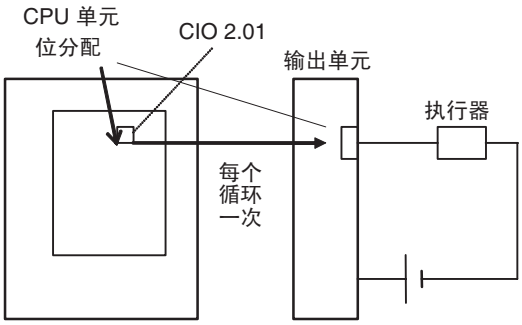
在将 I/O 区内的位分配给一个输出单元时，该位称之为输出位。一个输出位的 ON/OFF 状态会输出到外部设备，例如执行器。有三种方法刷新一个输出单元的输出位状态：正常 I/O 刷新、立即刷新和 IORF（097）刷新。

正常 I/O 刷新

每个循环在程序执行完成后将输出位的状态输出到外部设备。

在下面例子中，CIO 2.01 分配给一个执行器，一个连接到输出单元输出端子的外部设备，CIO 2.01 的 ON/OFF 状态每个循环一次地输出到这个执行器。





立即刷新

通过在一条指令前输入一个感叹号，指定指令变量的立即刷新，这个指令的操作数应是一个输入位或输入字，刷新的执行如下所示。这个立即刷新是在每循环一次的正常 I/O 刷新以外执行的刷新。

● 位操作数

紧接在该指令执行后，分配给包含指定位的那个字对应的 16 点 I/O 的 ON/OFF 状态将被输出到输出设备。

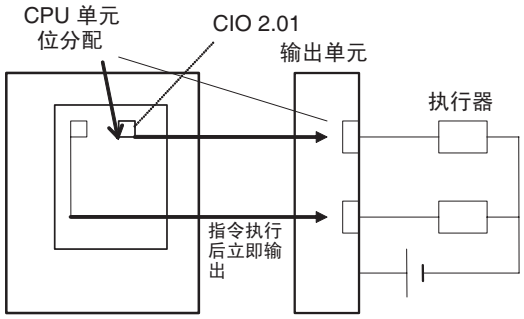
● 字操作数

紧接在该指令执行后，分配给指定字的 16 点 I/O 的 ON/OFF 状态会输出到输出设备。

● 示例

在下面例子中， CIO 2.01 分配给执行器，该执行器是连接到输出单元输出端子的外部设备，在指令 !OUT 2.01 执行后， CIO 2.01 的 ON/OFF 状态被立即输出到执行器。

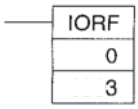




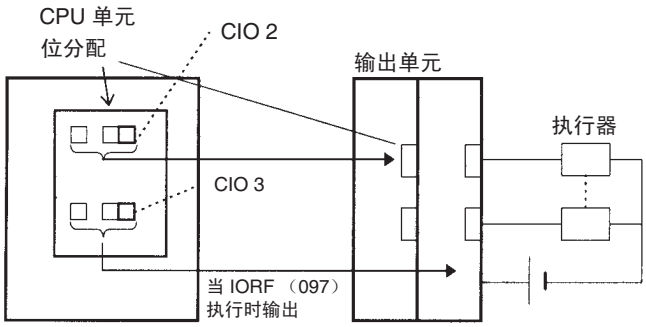
IORF （097）刷新

执行 IORF（097）（I/O 刷新）时，将字指定范围内的输出位 ON/OFF 状态输出到外部设备，该 I/O 刷新是在每循环一次的正常 I/O 刷新以外执行的刷新。

下面的 IORF （097）指令刷新所有在字 CIO 0 ～ CIO 3 的 I/O 区内 I/O 点的状态。从输入单元读取输入点的状态，输出位的状态被写入到输出单元。



在这个例子中，将分配给 CIO 2 和 CIO 3 的输入点状态输出到输出单元。（CIO 0 和 CIO 1 被分配给输入单元）。

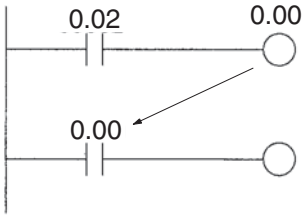


正确使用注意事项

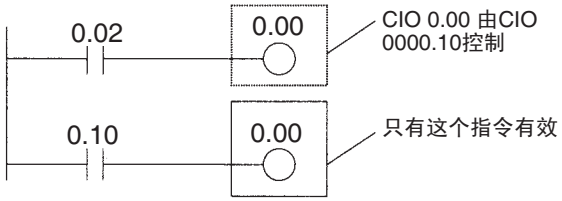
通过接通输出 OFF 位 （A500.15），可使所有的基本 I/O 单元和特殊 I/O 单元的输出关闭，这时即使实际的输出被关闭，输出位的状态不受影响。

输出位的限制

输出位能以任何次序编程，输出位可用作输入指令中的操作数，并且一个输出位可被用作常开和常闭条件，并且使用次数没有限制。



一个输出位在控制它状态的输出指令中只能使用一次。如果一个输出位在两条或多条输出指令中使用时，只有最后一个输出指令是有效的。



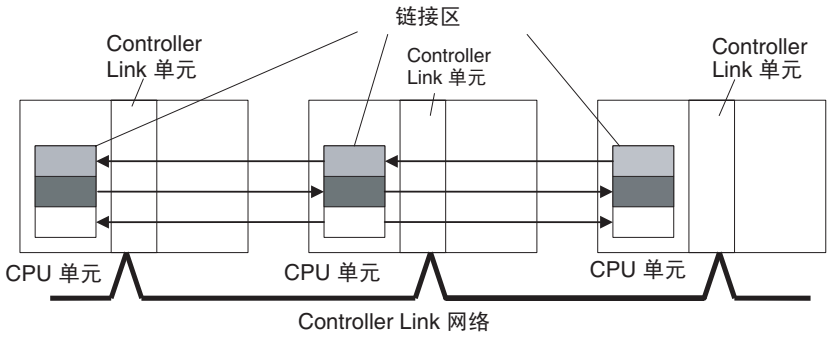
6-3 数据链接区

数据链接区地址范围为字 CIO 1000 ～ CIO 1199，位 CIO 1000.00 ～ 1199.15。当 LR 设为 Controller Link 网络的数据链接区时，链接区中的字用于数据链接。

一个数据链接单元（不依靠程序）通过安装在 PLC 的 Controller Link 单元，自动地在网络中与其它 CPU 单元共享数据链接区的数据。

数据链接可以自动产生（每个节点使用相同数目的字），或手动分配。当一个用户用手动定义数据链接时，他可以为每个节点分配任意数目的字，并且使该节点只接收或者只发送。详细信息，请参考 Controller Link 单元操作手册（样本编号 W309）。

当 LR 未被设为 Controller Link 网络的数据链接区时，数据链接区中的字可用在程序中。



在数据链接区的位可以被强制置位和强制复位。

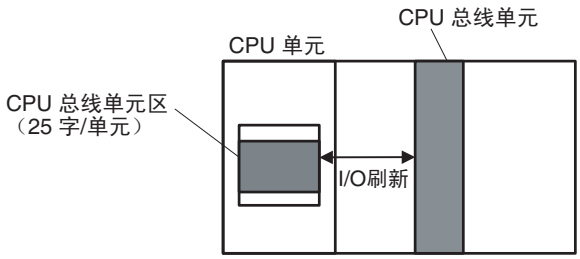
6-4 CPU 总线单元区

CPU 总线单元区的地址范围为从字 CIO 1500 ～ CIO 1899。

CPU 总线单元区内的字可以分配给 CPU 总线单元用于传输数据，例如单元的操作状态。每一个单元按照单元的单元编号设定分配 25 个字。

在以下情况时数据与 CPU 总线单元进行交换。

- 在 I/O 刷新期间
- DLNK （226）执行时



每个 CPU 总线单元按单元编号分配 25 个字，如下所示。

单元编号	分配字
0	CIO 1500 ～ CIO 1524
1	CIO 1525 ～ CIO 1549
2	CIO 1550 ～ CIO 1574
3	CIO 1575 ～ CIO 1599
4	CIO 1600 ～ CIO 1624
5	CIO 1625 ～ CIO 1649
6	CIO 1650 ～ CIO 1674
7	CIO 1675 ～ CIO 1699
8	CIO 1700 ～ CIO 1724
9	CIO 1725 ～ CIO 1749
A	CIO 1750 ～ CIO 1774
B	CIO 1775 ～ CIO 1799
C	CIO 1800 ～ CIO 1824
D	CIO 1825 ～ CIO 1849
E	CIO 1850 ～ CIO 1874
F	CIO 1875 ～ CIO 1899

25 个字的功能根据所使用的 CPU 总线单元而定。详细信息，请参考单元的操作手册。

CPU 总线单元区中没有分配给 CPU 总线单元的字只能用于程序中。

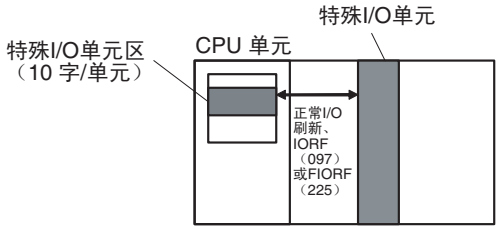
6-5 特殊 I/O 单元区

特殊 I/O 单元区地址范围为字 CIO 2000 ～ CIO 2959。

特殊 I/O 单元区的字分配给特殊 I/O 单元用于传输数据，例如单元的操作状态。每一个单元按照单元的单元编号设置分配 10 个字。共有 96 个单元可被使用，单元编号为 0 ～ 95。

在以下情况时特殊 I/O 单元区内的字被刷新：

- 正常 I/O 刷新时
- 执行 IORF （097）时
- 执行 FIORF （225）时



每个特殊 I/O 单元按单元编号分配 10 个字，如下所示。

单元编号	分配字
0	CIO 2000 ～ CIO 2009
1	CIO 2010 ～ CIO 2019
2	CIO 2020 ～ CIO 2029
3	CIO 2030 ～ CIO 2039
4	CIO 2040 ～ CIO 2049
5	CIO 2050 ～ CIO 2059
6	CIO 2060 ～ CIO 2069
7	CIO 2070 ～ CIO 2079
8	CIO 2080 ～ CIO 2089
9	CIO 2090 ～ CIO 2099
10 （A）	CIO 2100 ～ CIO 2109
11 （B）	CIO 2110 ～ CIO 2119
12 （C）	CIO 2120 ～ CIO 2129
13 （D）	CIO 2130 ～ CIO 2139
14 （E）	CIO 2140 ～ CIO 2149
15 （F）	CIO 2150 ～ CIO 2159
16	CIO 2160 ～ CIO 2169
17	CIO 2170 ～ CIO 2179
⋮	⋮
95	CIO 2950 ～ CIO 2959

分配给一个单元的 10 个字的功能根据所使用的特殊 I/O 单元而定。详细信息，请参考单元的操作手册。

特殊 I/O 单元区中没有分配给特殊 I/O 单元的字只能用于程序中。

6-6 DeviceNet 区

DeviceNet 区的地址范围为字 CIO 3200 ～ CIO 3799。

在 DeviceNet 区内的字分配给从站，用于 DeviceNet 远程 I/O 通信。通过 DeviceNet 单元与网络中的从站定期交换数据 （不依赖程序）。

使用固定分配时，按固定分配 1、 2 和 3 把这些字分配给从站。可选择这些固定区中的一个。

区	输出区 （主站到从站）	输入区 （从站到主站）
固定分配区 1	CIO 3200 ～ CIO 3263	CIO 3300 ～ CIO 3363
固定分配区 2	CIO 3400 ～ CIO 3463	CIO 3500 ～ CIO 3563
固定分配区 3	CIO 3600 ～ CIO 3663	CIO 3700 ～ CIO 3763

当使用固定分配的远程 I/O 从站功能时，下列字分配给 DeviceNet 单元。

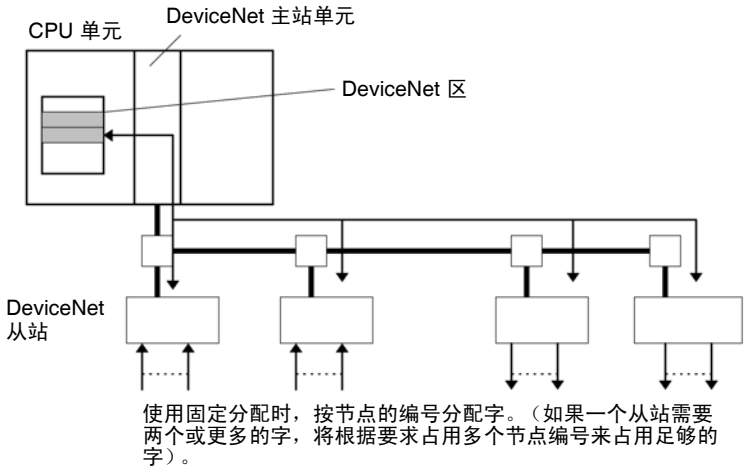
区	输出区 （主站到从站）	输入区 （从站到主站）
固定分配区 1	CIO 3370	CIO 3270
固定分配区 2	CIO 3570	CIO 3470
固定分配区 3	CIO 3770	CIO 3670

附加信息

在 DeviceNet 网络中有两种分配 I/O 的方法：按节点地址的固定分配和用户设置的分配。

- 使用固定的分配时，按节点地址在指定的固定分配区 （1 ～ 3）内的字被自动地分配给从站。
- 使用用户自定义分配时，用户可以将下列字分配给从站。
  CIO 0 ～ CIO 6143
  W0 ～ W511
  H0 ～ H511
  D0 ～ D32767
  E00_0 ～ E0C_32767，组：0 ～ 18 hex
  （不能指定高于 C 的 EM 组）。

关于字分配的详细信息，请参考 DeviceNet 操作手册 （样本编号 W267）。



6-7 工作区

工作区有 512 个字，地址范围为 W0 ～ W511。该区中的字不能用于外部 I/O 端子的 I/O。可在程序内部使用这些字。

在 CIO 区的内部 I/O 区中没有使用的字（CIO 1300 ～ CIO 1499 和 CIO 3800 ～ CIO 6143）也可以用于程序中。在 CIO 区没有使用的字可能会在新版 CPU 单元中分配到新的功能。因此在使用内部 I/O 区中的字前，首先应使用在工作区中可用的字。

6-8 保持区

保持区的地址范围为字 H000 ～ H511。这些字只能用于程序中。当电源关闭或运行模式从 PROGRAM 改变为 RUN /MONITOR 模式或反之时，保持区位的状态将会被保持。

保持区的位可以按任何顺序用在程序中，可用作常开或常闭条件，可任意多次调用。

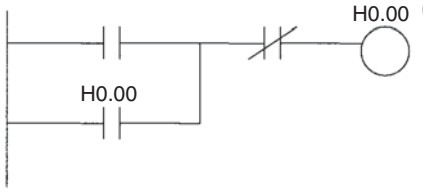
● 保持区初始化

当 PLC 的电源重启时，或者 PLC 的运行模式从 PROGRAM 改变为 RUN /MONITOR 模式或反之时，保持区的数据不会被清除。

如果保持区的位被编程到 IL （002）和 ILC （003）之间，并且 IL （002）的执行条件为 OFF 时，保持区的位将被清除。如果要在 IL（002）的执行条件为 OFF 时要仍保持一个位为 ON，则应在 IL（002）指令前用 SET 指令将该位置为 ON 。

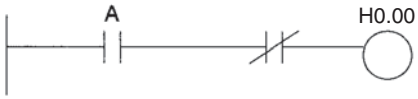
● 自保持位

当用保持区的位来编程一个自保持位时，即使在电源复位时，自保持位也不会被清除。



如果没有用保持区的位作为自保持位，在电源复位时，该位将变为 OFF 状态，且自保持位将会被清除。

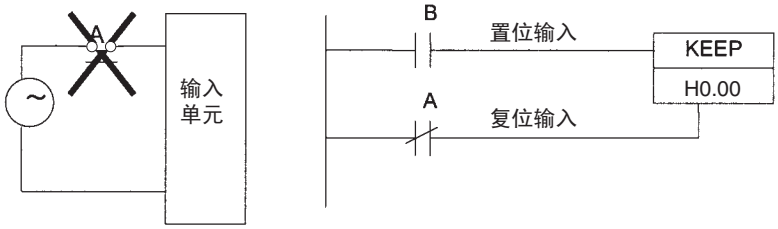
在下面图中，如果使用一个保持区的位但其未编程为自保持位的情况下，在电源复位时，由于执行条件 A 该位将会变为 OFF 状态。



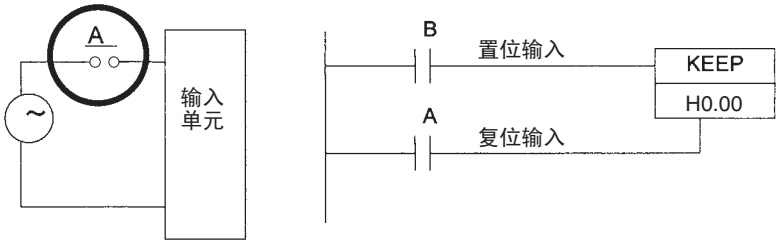
保持区的字从 H512 ～ H1535 可被设置用于功能块或 SFC 程序。这些字不可指定为程序中的指令操作数。

● 注意事项

当一个保持区的位用在一个 KEEP （011）指令中时，如果输入设备使用交流电源供电时，决不能用常闭条件作为复位输入。当电源关断或暂时中断时，这个输入位将先于 PLC 内部电源进入 OFF 状态，并使该保持区的位复位。



应使用如下图所示的配置。



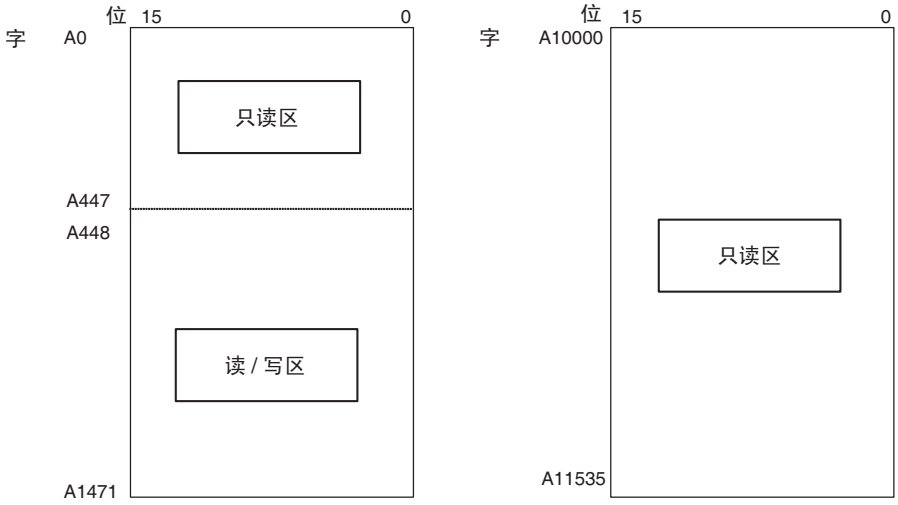
在程序中使用位地址的顺序或者使用常开或常闭条件编程的次数没有限制。

6-9 辅助区

辅助区的地址范围为字 A0 ～ A1471 和 A1000 ～ A11535，位 A0.00 ～ A1471.15 和 A1000.00 ～ A11535.15。这些字预先分配给标志和控制位用于监控和控制操作。

有一些字或位是由系统自动设定的，其他的由用户手动设定。辅助区包含由自诊断、初始设定、控制位和状态数据设定的错误标志。（关于辅助区位和字功能的信息，请参考 A-3 辅助区）。

A0～A447和A10000～A11535是只读的，但从A448 ～ A1471 可通过程序或者CX-Programmer进行读写。



用 CX-Programmer 可以实现下面的操作，向辅助区写入数据：在监控编程地址 （设置数值对话框）时改变当前值或编辑 PLC 数据表后向 PLC 传送数据。此外，在线置位 / 复位操作不能用于辅助区中的位。参考 CX-Programmer 操作手册 （样本编号 W414）。

正确使用注意事项

更新版的 CPU 单元可能会给未定义的辅助区的字或位分配功能。请勿将辅助区中未定义的字或位作为工作字或位在用户程序中使用。

6-10 暂存继电器区

TR 区包含 16 个位，地址范围为 TR0 ～ TR15。当有几个输出分支并且不能使用联锁时，TR 位是非常有用的。

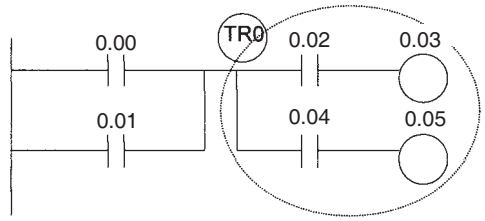
TR 位可以根据需要以任意顺序多次使用，且只要同一 TR 位不是用在同一指令块中使用两次。

TR 位只能用在 OUT 和 LD 指令中，OUT 指令（OUT TR0 ～ OUT TR15）存储分支点的 ON /OFF 状态，而 LD 指令则是调用保存的支路点的 ON /OFF 状态。

不能使用 CX-Programmer 改变 TR 位。

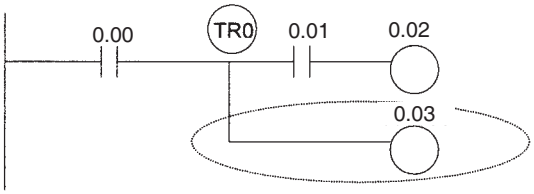
● 应用示例

在这个例子中，说明当一个分支点直接连接两个输出时要使用一个 TR 位。



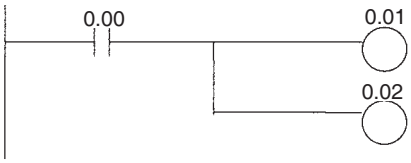
指令	操作数
LD	0.00
OR	0.01
OUT	TR 0
AND	0.02
OUT	0.03
LD	TR 0
AND	0.04
OUT	0.05

在这个例子中，一个 TR 位用于将输出连接到一个没有独立执行条件的支路点上的情况。

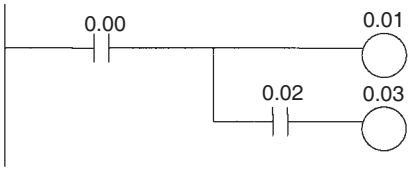


指令	操作数
LD	0.00
OUT	TR 0
AND	0.01
OUT	0.02
LD	TR 0
OUT	0.03

当分支点后没有可执行的条件，或者只在指令块的最后一行有一个执行条件时，没有必要使用 TR 位。



指令	操作数
LD	0.00
OUT	0.01
OUT	0.02



指令	操作数
LD	0.00
OUT	0.01
AND	0.02
OUT	0.03

6-11 数据存储器区

DM 的地址范围为字 D0 ～ D32767，这数据区能用作通用数据存储，并只能以字或位的形式进行存取和管理。

当 PLC 的电源重启或者 CPU 单元运行模式从 PROGRAM 模式转变为 RUN/MONITOR 模式或者是相反情况时， DM 区的数据将保持不变。

在 DM 区的位不能被强制置位或者强制复位。

间接寻址

可以用两种方法间接寻址 DM 区里的字：二进制模式和 BCD 模式。

● 二进制模式寻址 （@D）

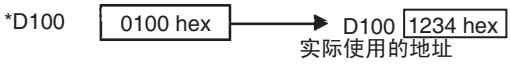
当一个“@”字符写在一个 DM 地址前，该 DM 字的内容作为二进制处理并且该指令将按这个二进制地址的 DM 字操作，整个 DM 区 （D0 ～ D32767）可用十六进制值 0 ～ 7FFF 来间接寻址。



● BCD 模式寻址 （*D）

当一个“*”字符写在一个 DM 地址前，该 DM 字的内容作为 BCD 处理并且该指令将按这个 BCD 地址的 DM 字操作，只有部分 DM 区 （D0 ～ D09999）可用 BCD 值 0 ～ 9999 来间接寻址。

例：[MOV #1234 *D100]



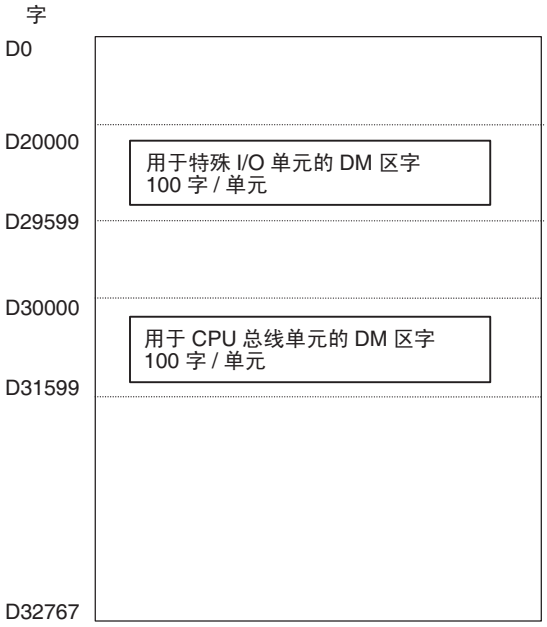
分配给特殊 I/O 单元和 CPU 总线单元的 DM 区

部分 DM 区分配给特殊 I/O 单元和 CPU 总线单元，用作如初始化单元设定等功能。如果在 PLC 中没有使用对应单元，这些字可用于通用数据存储。

这些单元的数据传输时间是不同的，但属于下列三种情况的一种。

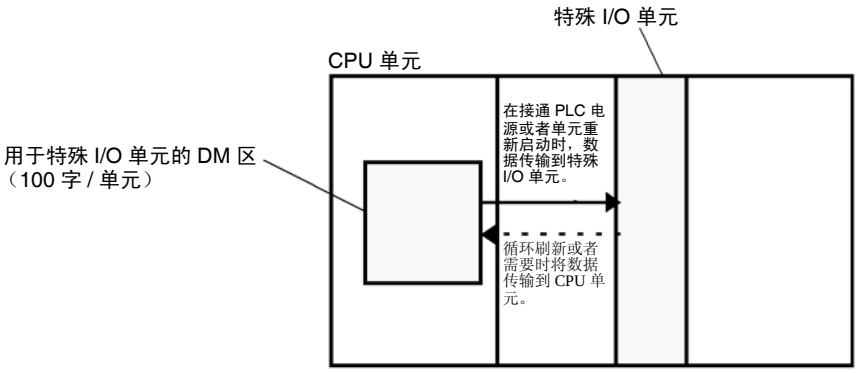
- 在 PLC 的电源接通或者该单元重新启动时传输数据。
- 每循环一次传输数据。
- 当有请求时传输数据。

关于数据传输时间的详细信息，请参考单元的操作手册。



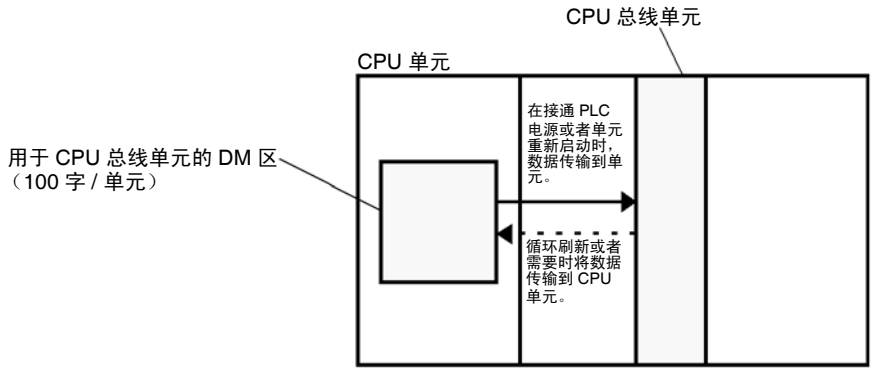
● 特殊 I/O 单元 （D20000 ～ D29599）

每个特殊 I/O 单元分配 100 个字 （根据单元号 0 ～ 95）。关于这些字的功能的详细信息，请参考单元的操作手册。



● CPU 总线单元 （D30000 ～ D31599）

每个 CPU 总线单元分配 100 个字（根据单元号 0 ～ F）。关于这些字的功能的详细信息，请参考单元的操作手册。对于一些 CPU 总线单元（如 Ethernet 单元）的初始化设定必须登记到 CPU 单元的参数区。通过编程设备登录这些数据。



6-12 扩展数据存储器区

EM 区的地址范围为字 E0_0 ～ E18_32767。 EM 区分为 24 个组 （0 ～ 18 hex）。地址中在 “E” 后面的十六进制数表示组。

EM 区用于通用数据存储和处理并且可按字或位访问。

当 PLC 的电源重启或者 CPU 单元运行模式从 PROGRAM 模式转变为 RUN /MONITOR 模式或者是相反情况时，在 EM 区的数据将保持不变。

自动地址分配

我们建议使用 EM 区来自动分配地址给符号 （变量）。如果使用了 EM 区，将可以对以下的字进行强制置位 / 复位。

型号	使用自动地址分配时，强制置位 / 复位的字
CJ2H-CPU64-EIP	E03_0 ～ E03_32767
CJ2H-CPU65-EIP	
CJ2H-CPU66-EIP	E06_0 ～ E09_32767
CJ2H-CPU67-EIP	E07_0 ～ E0E_32767
CJ2H-CPU68-EIP	E11_0 ～ E18_32767

关于使用自动地址分配的步骤，请参考 5-5-8 符号 （变量）的自动地址分配。

文件存储器

文件存储器可用于 CPU 单元的文件储存。详情请参考第 7 章 文件操作。

跟踪存储器

跟踪存储器用于数据跟踪的数据采样。详情请参考 10-7-6 数据跟踪。

直接指定 EM 地址

有两种方法来指定一个 EM 地址：同时指定组和地址或者指定在当前组中的一个地址。通常我们建议同时指定组和地址。

● 指定组和地址

用这种方法，在地址前指定组编号。例如 E2_10 表示为在组 2 中的地址 E10。

● 指定当前组的地址

用这种方法，只要指定地址。例如 E10 表示在当前组中指定地址为 E10。

- 当运行模式从 PROGRAM 模式转变为 RUN /MONITOR 模式，并且如果 IOM 保持位（A500.12）不是 ON 状态时，当前组将被复位为 0。
- 可通过 EMBC （281）改变当前组来访问其他组中的数据。
- A301 存有当前 EM 组编号。

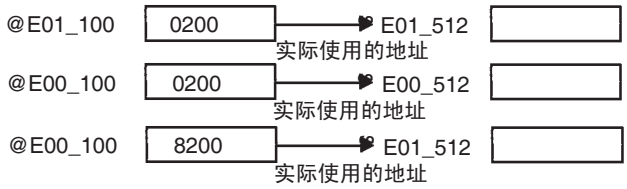
- 当通过循环任务进行程序处理时不会改变当前组。例如，如果当前 EM 组在任务 1 中变为了组 2，当前 EM 组在任务 2 中仍为组 2。如果在一个中断任务中组被改变，则当前组将会返回到其原始值（在原循环任务中的值）。

间接指定 EM 寻址

可以用两种方法间接寻址 EM 区里的字；二进制模式和 BCD 模式。

● 二进制模式寻址 （@E）

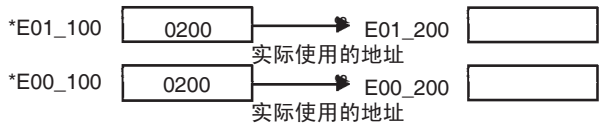
当 EM 地址前输入一个 “@” 字符时，该 EM 字的内容作为二进制处理，且指令将会在同一组或下一组的这个二进制地址内处理该 EM 字。在同一 EM 组内的所有字 （E0 ～ E32767），能用十六进制值 0 ～ 7FFF 进行间接寻址，并且能用十六进制值 8000 ～ FFFF 间接寻址下一个 EM 组（E0 ～ E32767）。



● BCD 模式寻址 （*E）

当 EM 地址前输入一个 “*” 字符时，该 EM 字的内容作为 BCD 值处理，以表示其他 EM 地址。

如果 BCD 值为 0 ～ 9999，末尾地址将在同一组中。



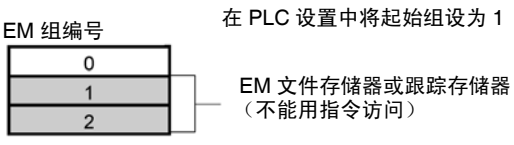
● EM 区与文件存储器或跟踪存储器的转化

可使用 CX-Programmer 指定 EM 区中的组，用于将 EM 区末尾指定的所有组转化为文件存储器或跟踪存储器。

文件存储器：
文件存储器可以用于储存文件，例如存储卡中储存的文件。最多可有 13 组转化为文件存储器。

跟踪存储器：
数据跟踪的结果可以储存在跟踪存储器中，增加了跟踪可储存结果的数量。

EM 组一旦转换为文件存储器或跟踪存储器，就不能使用指令通过用户程序进行访问。如果尝试访问，将会出现一个非法访问错误。



6-13 定时器区

共有 4,096 个定时器可以使用，编号为 T0 ～ T4095。有两个定时器数据区：定时器完成标志区和定时器当前值 （PV）区。

- 定时器完成标志 （T）
  定时器编号用于访问完成标志。当定时器的设定时间到达时，完成标志转为 ON。
- 定时器 PV （T）
  定时器编号也用于读写定时器的当前值 （16 位字）。当前值 （PV）的增加或减少作为定时器的运行。

当用定时器编号作为一个需要位数据的操作数时，定时器编号访问定时器的完成标志。当用定时器编号作为一个需要字数据的操作数时，定时器编号访问定时器的 PV 。

相同的定时器编号被以下所有指令使用：100MS 定时器 （TIM/TIMX （550））、 10MS 定时器 （TIMH （015） /TIMHX （551）） , 1MS 定时器 （TMHH （540） /TMHHX （552））、 0.1MS 定时器 （TIMU （541） /TIMUX （556））、 0.01MS 定时器 （TMUH （544） /TMUHX （557））、累计定时器 （TTIM （087） /TTIMX （555））、定时器等待 （TIMW （813） /TIMWX （816））和高速定时器等待 （TMHW （815） /TMHWX （817））。

正确使用注意事项

建议不要在两个定时器指令中使用同一定时器编号，如果它们同时计时将会产生不正确的结果。（如在两个或更多定时器指令用同一定时器编号，在程序检查时将会产生一个错误，但只要不在同一循环执行指令，定时器仍将会正常运行）。

下表表示何时定时器 PV 和完成标志被复位。

指令名称 *1	对 PV 和完成标志的影响			跳转和联锁操作	
	模式改变 （从 PROGRAM 到 RUN 或 MONITOR 或相反转换） *2	PLC 设置 *3	TRSET （549）	跳转 （JMP-JME） 或 “等待”状态中的任务 *4	联锁 （IL-ILC）
100MS 定时器： TIM/TIMX （550）	PV → 0 标志 → OFF	PV → 0 标志 → OFF	PV → 保持 标志 → OFF	在操作定时器内刷新的 PV	PV → SV （复位至 SV） 标志 → OFF
10MS 定时器： TIMH （015） /TIMHX （551）					
1MS 定时器：TMHH （540） /TMHHX （552）					
0.1MS 定时器：TIMU （541） /TIMUX （556）					
0.01MS 定时器：TMUH （544） /TMUHX （557）					
累计定时器： TTIM （087） /TTIMX （555）				PV 保持	PV 保持
定时器等待： TIMW （813） TIMWX （816）				在操作定时器内刷新的 PV	---
高速定时器等待： TMHW （815） /TMHWX （817）					---

*1 由于 TIML （542）、TIMLX （553）、MTIM （543）和 MTIMX （554）指令不使用定时器编号，因此它们在不同条件下重置。（TIML （542）和 TIMLX （553）重置为它们的设定值， MTIM （543）和 MTIMX （554）重置为 0。）详情请参考对这些指令的描述。

*2 如果 IOM 保持位 （A500.12）为 ON，在出现致命错误或者运行模式从 PROGRAM 转变为 RUN /MONITOR 模式或反之时， PV 和完成标志的内容将保持。当电源重启时，PV 和完成标志的内容将会被清除。详细信息，请参考 CS/CJ/NSJ 系列指令参考手册 （样本编号 W474）。

*3 如果 IOM 保持位 （A500.12）为 ON，且在 PLC 设置启动标签页中选择了启动保持区的 IOM 保持位检查框，则当 PLC 的电源重启时，PV 和完成标志的内容保持不变。

*4 即使在 JMP 和 JME 指令之间跳跃或在 “等待”状态的任务中，按定时器编号 T0 ～ T2047 编排的 100MS 定时器 （TIM/TIMX （550））、 10MS 定时器 （TIMH （015） /TIMHX （551））、 1MS 定时器 （TMHH （540） /TMHHX（552））、定时器等待（TIMW（813）/TIMWX（816）和高速定时器等待（TMHW（815）/TMHWX（817））的当前值也将被更新。按定时器编号 T2048 ～ T4095 编排的定时器器当前值在 JMP 和 JME 指令之间跳跃或在 “等待”状态的任务中将被保持。

- 定时器完成标志可以强制置位和强制复位。
- 使用定时器编号的顺序或对编程的常开或常闭条件的使用次数为没有限制。
- 除 0.1MS 定时器（TIMU（541）/TIMUX（556））和 0.01MS 定时器（TMUH（544）/TMUHX（557））以外的所有定时器的当前值可作为字数据读取。

6-14 计数器区

共有 4,096 个计数器可被使用，计数器编号为 C0 ～ C4095。有两个计数器区：计数器完成标志区和计数器当前值 （PV）区。

- 计数器完成标志 （C）
  计数器编号用于访问完成标志。到达计数器设定值时，完成标志转为 ON。
- 计数器 PV （C）
  计数器编号也用于读写计数器的当前值 （16 位）。当前值 （PV）的增加或减少作为计数器的运行。

当用计数器编号作为一个需要位数据的操作数时，计数器编号访问计数器的完成标志。当用计数器编号作为一个需要字数据的操作数时，计数器编号访问计数器的 PV。

相同的计数器编号被以下所有指令使用：CNT、CNTX（546）、CNTR（012）、CNTRX（548）、CNTW（814）和 CNTWX （818）。

正确使用注意事项

建议不要在两个计数器指令中使用同一计数器编号，如果它们同时计数将会产生不正确的结果。如果两个或更多计数器指令使用同一计数器编号，在程序检查时将会产生一个错误，但只要不在同一循环执行指令，计数器将会运行。

下表表示何时计数器 PV 和完成标志被复位。

指令名称	对 PV 和完成标志的影响					
	复位	模式改变	PLC 启动	复位输入	CNR （545） /CNRX （547）	联锁 （IL-ILC）
计数器：CNT/CNTX （546）	PV → 0000 标志 → OFF	保持	保持	复位	复位	保持
可逆计数器：CNTR （012） /CNTRX （548）						
计数器等待：CNTW （814） /CNTWX （818）						

- 计数器完成标志可以强制置位和强制复位。
- 即使 PV 能通过强制置位复位完成标志来间接刷新，但计数器 PV 不能强制置位或强制复位。
- 使用计数器编号的顺序或对编程的常开或常闭条件使用次数为没有限制。
- 计数器 PV 只能以字的形式读取且可以在程序中使用。

6-15 任务标志

任务标志范围为 TK0 ～ TK127，且对应于循环任务 0 ～ 127。

当一个循环任务处在就绪（READY）或运行（RUN）状态时，相应的任务标志为 ON，当一个循环任务在没有执行（INI）或等待（WAIT）状态时，则相应的任务标志为 OFF。

注　这些标志只表示循环任务的状态，并不反映附加循环任务或中断任务的状态将被清除。

任务标志在下列情况下被清除，而不管 IOM 保持位的状态（A500.12）。

- 运行模式从 PROGRAM 模式变为 RUN/MONITOR 模式或者相反情况。
- PLC 电源重启。

任务标志不能被强制置位和强制复位。

6-16 变址寄存器

16 个变址寄存器（IR0 ～ IR15）用于间接寻址。每个变址寄存器保存一个单独的 PLC 存储器地址，这是 I/O 存储器内字的绝对地址。 I/O 存储器区的地址与 CIO 区、 DM 区的不同。它们是连续的 RAM 地址。

变址寄存器可在每个任务中单独使用，也可被所有任务共享。

用户不能直接在变址寄存器中输入 PLC 存储器地址。需使用 MOVR （560）将一个常规数据区地址转换为与它等效的 PLC 存储器地址，且把转换值写入指定的变址寄存器。（用 MOVRW （561）在一个变址寄存器里设置一个定时器 / 计数器 PV 值的 PLC 存储器地址。）

附加信息

关于 PLC 存储器地址的详细信息，请参考 A-4 PLC 存储器地址的内存印象图。

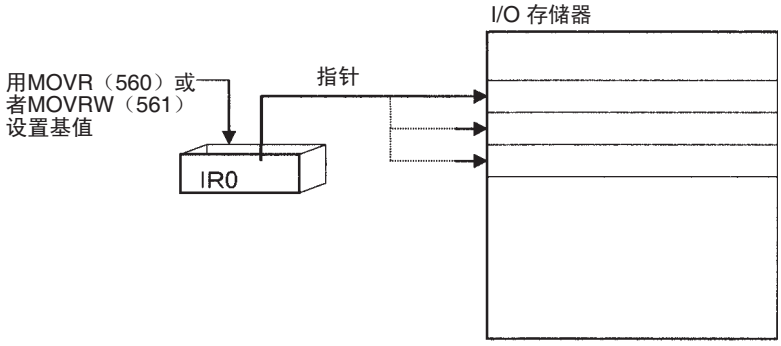
● 间接寻址

当使用一个变址寄存器作为带 “,” 前缀的操作数时，将由变址寄存器内的 PLC 存储器地址指示的字操作这条指令，而不是变址寄存器本身。原则上变址寄存器是 I/O 存储器的指针。

- 用 PLC 存储器地址可以无间断地指定 I/O 存储器里的所有地址（除变址寄存器、数据寄存器和条件标志外），而不需要指定数据区。
- 除基本间接寻址外，在一个变址寄存器里的 PLC 存储器地址可用一个常数或者数据寄存器偏移，自动增加或者自动减少。在循环语句中可用这个功能读或写数据，当指令执行时使其地址每次自动增加或者减少。

用偏移和增加 / 减少变量，可用 MOVR （560）或者 MOVRW （561）设置变址寄存器的基值，然后在每一个指令中作为指针修改。

对于指针操作，可使用直接指定变址寄存器 （MOVR （560）、 MOVRW （561）、递增指令、递减指令或算术指令）的特殊指令，或者也可使用间接偏移、自动递增或自动递减。



可设定 PLC 使变址寄存器在每个任务中单独使用或被所有任务共享。

变址寄存器在以下情况被清除：

- 在 PROGRAM 或 MONITOR 模式和 RUN 模式之间变更运行模式时
- PLC 的电源置 ON 时
- 发生致命错误时 （由于 FALS （007）指令导致致命错误的情况除外）

正确使用注意事项

当以变址寄存器对存储器进行间接寻址时，可以指定 I/O 存储器之外的区域并产生非法访问错误。关于 PLC 存储器地址的限制的详细信息，请参考 A-4 PLC 存储器地址的内存印象图。

下表显示的是用变址寄存器间接寻址 I/O 存储器时可能的变量。

变量	功能	语法	示例	
间接寻址	IR □的内容作为一个位或者字的 PLC 存储器地址。	,IR □	LD , IR0	IR0 的内容作为 PLC 存储器地址装载一个位。
带常数偏移的间接寻址	常数前缀加上 IR □的内容，所得结果作为一个位或者字的 PLC 存储器地址。 该常数可为 - 2,048 ～ 2,047 间的任一整数。	常数 ,IR □ （在常数中包括 + 或者 —）	LD +5, IR0	IR0 的内容加 5 作为 PLC 存储器的地址装载该位。
带 DR 偏移的间接寻址	数据寄存器内容加上 IR □的内容，所得结果作为一个位或字的 PLC 存储器地址。	DR □ , IR □	LD DR0 , IR0	DR0 的内容加 IR0 的内容作为 PLC 存储器的地址装载该位。
带自动递增的间接寻址	指令执行后 IR □的内容增加 1 或者 2 作为一个位或字的 PLC 存储器地址。	加 1： , IR □ + 加 2： , IR □ ++	LD , IR0++	IR0 的内容作为 PLC 存储器的地址装载该位然后 IR0 的内容加 2。
带自动递减的间接寻址	IR □的内容减 1 或者 2，其结果作为一个位或字的 PLC 存储器地址。	减 1： ,- IR □ 减 2： ,- - IR □	LD , - -IR0	IR0 的内容减 2 作为 PLC 存储器的地址装载该位。

注　IR □表示变址寄存器 IR0 ～ IR15。

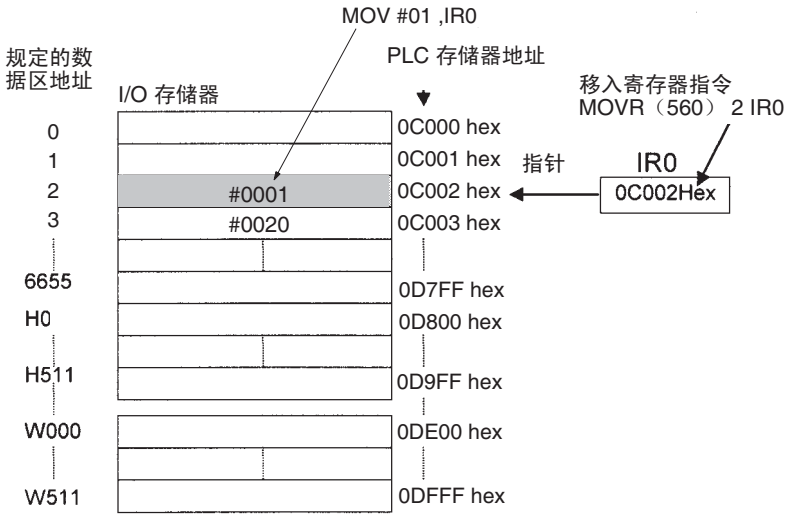
● 示例

这个例子说明怎样将一个字 （CIO 2） 的 PLC 存储器地址存到变址寄存器中 （IR0），并在一条指令中使用变址寄存器和自动增加变量的方法。

MOVR （560）2 IR0　　将 CIO 2 的 PLC 存储器地址保存到 IR0 中。

MOV （021）#1 ,IR0　　将 #1 写到包含在 IR0 内的 PLC 存储器地址中，如 CIO 2。

MOV （021）#20 +1, IR0　　读 IR0 的内容然后加 1 作为 PLC 存储器地址，如 CIO 3。



注　PLC 存储器地址列举在上图中，但是当使用变址寄存器时不必要知道 PLC 存储器地址。

由于操作数作为字数据处理而其它作为位数据处理，因此变址寄存器里的数据含义将根据所使用的操作数而不同。

- 字操作数：

MOVR （560） 0 IR2
MOV （021） D0 , IR2

当该操作数作为一个字处理时，则变址寄存器的内容作为一个字的 PLC 存储器地址。在这个例子中 MOVR （560）将 CIO 2 的 PLC 存储器地址置入 IR2 中，并且 MOV （021）指令将 D0 的内容复制到 CIO 2 中。

- 位操作数：

MOVR （560） 0.13 ,IR2
SET +5 , IR2

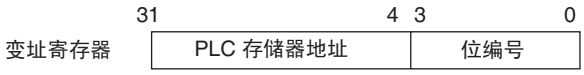
变址寄存器也可用于指定位，如以上 SET 指令中的一样。在这个例子中，MOVR （560）将 CIO 0.13 的 PLC 存储器地址置入 IR2 中， SET 指令将 5 加到这个 PLC 存储器地址上的位 13，于是位 CIO 1.02 转为 ON。

附加信息

当使用 MOVR （560）在变址寄存器中设定字地址时，地址保存如下：



当使用 MOVR （560）在变址寄存器中设定位地址时，地址保存如下：



● 直接寻址

当将一个没有 “,” 前缀的变址寄存器用作操作数时，该指令将操作变址寄存器自身表示的内容 （双字或者 “双精度” 值）。只有用下表所示的指令可直接寻址变址寄存器。用这些指令把变址寄存器作为指针。

虽然变址寄存器通常可用作间接寻址，但它们在其它指令中不能用于直接寻址。

指令组	指令名称	助记符
数据移动指令	传送到寄存器	MOVR （560）
	将定时器 / 计数器 PV 传送到寄存器	MOVRW （561）
	双精度传送	MOVL （498）
	双精度数据交换	XCGL （562）
表数据处理指令	设置记录位置	SETR （635）
	取得记录编号	GETR （636）
跟踪指令	无符号 1 字记录搜索指令	RSRCH （360 ～ 364）
	无符号 1 字记录分类	RSORT （203）
递增 / 递减指令	双精度二进制递增	++L （591）
	双精度二进制递减	- -L （593）
比较指令	双精度相等	=L （301）
	双精度不等	< >L （306）
	双精度小于	< L （311）
	双精度小于或等于	< =L （316）
	双精度大于	> L （321）
	双精度大于或等于	> =L （326）
	双精度比较	CMPL （060）
符号 （变量）算术指令	双精度有符号无进位加	+L （401）
	双精度有符号无进位减	-L （411）

SRCH （181）、 MAX （182）和 MIN （183）指令能将带期望值 （搜索值、最大值或最小值）字的 PLC 存储器地址输出到 IR0 中。并能在后面的指令中使用 IR0 访问该字的内容。

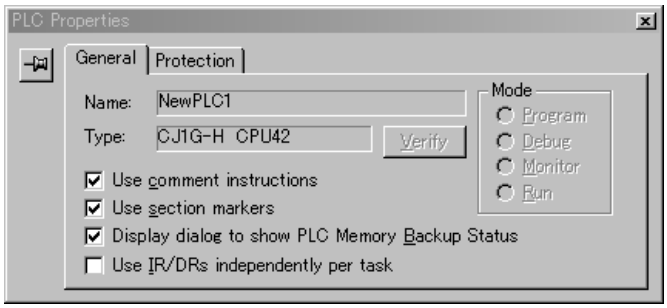
● 注意事项

- 在未将 PLC 存储器地址设置寄存器以前，不要使用变址寄存器。如果使用没有设置值的寄存器，指针操作将会不可靠。
- 在启动一个中断任务时，在变址寄存器里的值是不确定的。当在一个中断任务中使用变址寄存器时，通常使用之前用 MOVR（560）或 MOVRW（561）指令在变址寄存器中设置 PLC 存储器地址。
- 每一个变址寄存器的任务是独立执行的，它们不会相互影响，例如在任务 1 中使用的 IR0 与在任务 2 中使用的 IR0 是不同的。因此，每一个变址寄存器任务可用 16 个变址寄存器。
- 从 CX-Programmer 可以读取执行循环的最后一个任务中的变址寄存器。如果使用相同编号的变址寄存器来执行多重任务，用 CX-Programmer 只能读出最后一个执行的任务的变址寄存器的值。用 CX-Programmer 不能写入变址寄存器的值。

● 共享变址寄存器

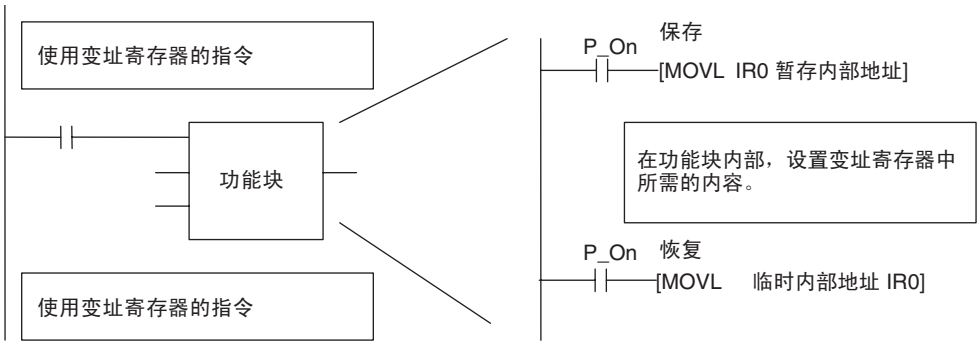
下面在 CX-Programmer PLC 属性对话框中可作下列设置，以控制任务间共享变址和数据寄存器。

要在所有任务间共享变址寄存器，在 PLC 属性对话框中取消 “每个任务单独使用的 IR/DR”（Use IR/DRs independently per task）选项框选择。



附加信息

当功能块被调用时，在功能块内部使用的变址寄存器的内容可能会毁坏。必须在调用功能块前保存变址寄存器的内容，并在离开功能块后恢复其内容。在功能块内部，设置变址寄存器中所需的内容。



6-17 数据寄存器

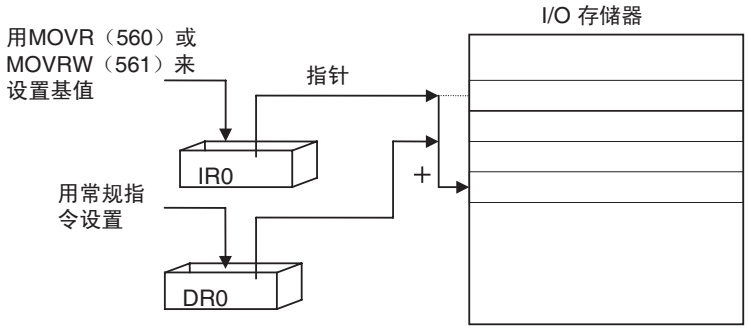
当间接寻址字地址时，这十六个数据寄存器（DR0 ～ DR15）用作偏移变址寄存器中的 PLC 存储器地址。当间接寻址字地址时，数据寄存器可用于指定加到变址寄存器的偏移。

数据寄存器可在每个任务中单独使用，也可被所有任务共享。

数据寄存器中的值可加到变址寄存器中的 PLC 存储器地址上，以指定 I/O 存储器中一个位或者一个字的绝对存储器地址。由于数据寄存器含有带符号的二进制数据，因此变址寄存器的内容能向前或者向后地址偏移。

可用常规指令将数据保存到数据寄存器中。

数据寄存器不能被强制置位或强制复位。



● 示例

下面的例子说明怎样使用数据寄存器偏移变址寄存器中的 PLC 存储器地址。

LD DR0 ,IR0
把 DR0 的内容加到 IR0 中，并装载在 PLC 存储器地址的位。

MOV （021） #1 DR0 ,IR1
把 DR0 的内容加到 IR1 中，并将 #1 写入 PLC 存储器地址。

● 值的范围

数据寄存器的内容是带符号的二进制数，因此其值范围是 - 32,768 ～ 32,767。（负值作为 2 的补数给出）。

十六进制内容	十进制表示
8000 ～ FFFF	-32,768 ～ -1
0000 ～ 7FFF	0 ～ 32,767

● 数据寄存器初始化

在下列情况下数据寄存器被清除：

- 运行模式从 PROGRAM 模式变为 RUN/MONITOR 模式或反之，且 IOM 保持位是 OFF。
- PLC 的电源重启并且 IOM 保持位是 OFF，或者在 PLC 设置中设为不保持。
- 发生致命错误 （由于 FALS （007）指令执行导致的情况除外）。

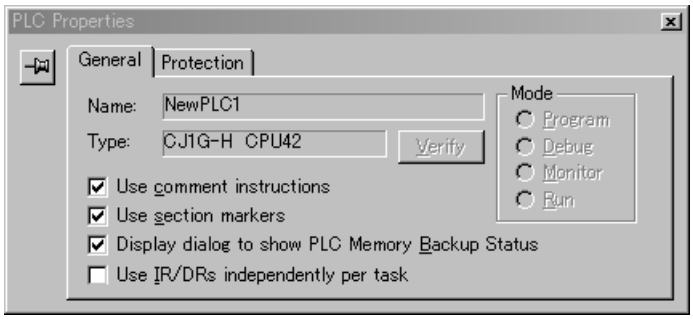
● IOM 保持位操作

如果 IOM 保持位 （A500.12）为 ON，当出现一个 FALS 错误或运行模式从 PROGRAM 模式变为 RUN/MONITOR 模式或者相反时，数据寄存器将不会被清除。

如果 IOM 保持位 （A500.12）为 ON，且在 PLC 设置的启动标签页中的启动保持区下，选择 IOM 保持位选择框，那么当 PLC 电源重启时，数据寄存器将不会被清除。

● 共享数据寄存器

下面在 CX-Programmer PLC 属性对话框中可作下列设置控制任务间共享变址和数据寄存器。



附加信息

不管是否在程序中使用变址和数据寄存器，我们推荐将 PLC 属性设为在任务间共享变址和数据寄存器。

6-18 条件标志

条件标志包含错误标志、进位标志和其他表示指令执行结果的标志，也包含了常 ON 和常 OFF 标志。在早期的 PLC 中，这些标志放在辅助区。

条件标志不同于全局符号 （变量）或地址，如 P_CY 和 P_ER。这些标志不能用指令或 CX-Programmer 直接写入。

当程序切换任务时，所有条件标志被清除，因此错误标志、存取错误标志和其他标志的状态只保持在发生错误的任务中。

条件标志不能被强制置位和强制复位。

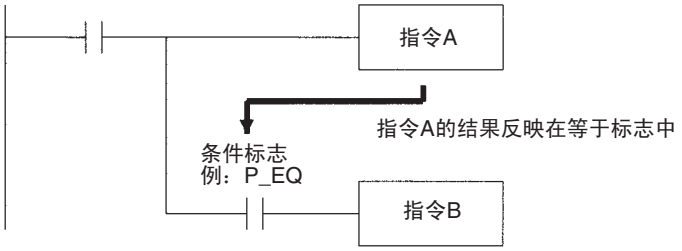
● 条件标志概述

下表总结了条件标志的功能。

名称	符号（变量）	标识	功能
错误标志	P_ER	ER	当在一个指令里的操作数数据不正确 （一个指令处理错误）时转为 ON 表示因一个错误使一个指令结束操作。 当 PLC 设置中设置为一个指令出错时 （指令操作错误）停止操作，当错误标志为 ON 时，程序将停止执行，并且指令处理错误标志 （A295.08）将转为 ON。
存取错误标志	P_AER	AER	当发生一个非法存取错误时，转为 ON。非法存取错误表示一个指令试图访问一个不能被访问的存储器区。 当 PLC 设置中设置为出现一个指令错误 （指令错误操作）时停止操作，存取错误标志转为 ON 时，将停止程序执行，且指令处理错误标志 （A4295.10）将转为 ON。
进位标志	P_CY	CY	当一个算术运算结果产生一个进位或者由一个数据移动指令把 “1” 移进进位标志时，进位标志转为 ON。 进位标志是某些数据移动和符号 （变量）算术指令结果的一部分。 .
大于标志	P_GT	>	当比较指令的第一个操作数大于第二个操作数或者其值超出规定的范围，该标志为 ON。
等于标志	P_EQ	=	当比较指令的两个操作数相等，也就是计算结果为 0 时，该标志为 ON。
小于标志	P_LT	<	当比较指令的第一个操作数小于第二个操作数或者其值在规定的范围内，该标志为 ON。
取反标志	P_N	N	当结果的最高有效位 （符号位）是 ON 时，该标志为 ON。
溢出标志	P_OF	OF	当运算结果超出结果字的范围时，该标志为 ON。
下溢出标志	P_UF	UF	当运算结果下溢出结果字范围时，该标志为 ON。
大于或等于标志	P_GE	>=	当比较指令的第一个操作数大于或等于第二个操作数时，该标志为 ON。
不等于标志	P_NE	< >	当比较指令的两个操作数不相等时该标志为 ON。
小于或等于标志	P_LE	< =	当比较指令的第一个操作数小于或等于第二个操作数时，该标志为 ON。
常 ON 标志	P_On	ON	始终 ON。
常 OFF 标志	P_Off	OFF	始终 OFF。

● 使用条件标志

条件标志由所有指令共享，因此它们的状态在一个循环内也许会经常改变。所以，必须在指令执行后立即读取条件标志。最好在一个支路中具有同一执行条件。



指令	操作数
LD	
指令A	
AND	P_EQ
指令B	

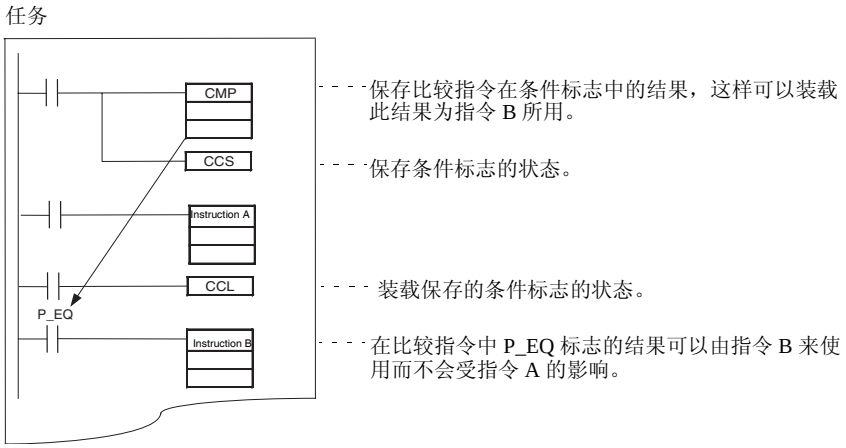
正确使用注意事项

- 由于条件标志由所有指令共享，因此通过一个单独中断任务可以从其设定的过程中改变程序操作。当编写程序时必须考虑到中断的影响。
- 当程序切换任务时，条件标志被清除。因此条件标志的状态不能传送给另一任务，例如不能在任务 2 中读取在任务 1 中的标志状态。

● 保存和装载条件标志状态

条件标志状态指令 （CCS （282）和 CCL （283）可用于保存和装载在一个任务中不同位置或者在不同的任务或循环中的条件标志状态。

下面的例子表示在一个任务的不同位置，怎样使用条件标志。

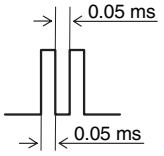
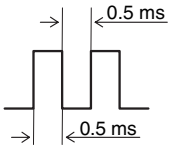
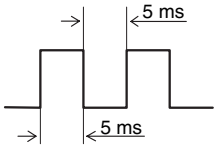
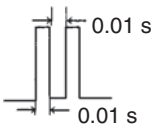
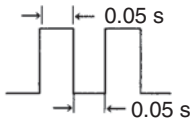
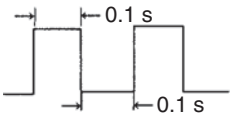
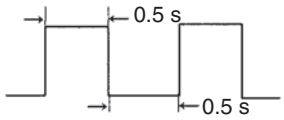
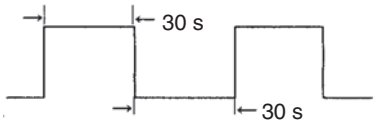


6-19 时针脉冲

时钟脉冲是由系统产生的，按一定时间间隔转 ON 和 OFF 的标志。

时钟脉冲是只读的：它们不能由指令或者 CX-Programmer 改写。在运行开始时，时钟脉冲被清除。

时钟脉冲是用全局符号 （变量）来指定的，例如 P_0_1ms 和 P_1ms。

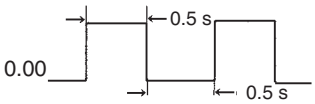
名称	符号（变量）	手持编程器名称	操作	
0.1 ms 时钟脉冲	P_0_1ms	0.1 ms		ON 0.05 ms OFF 0.05 ms
1 ms 时钟脉冲	P_1ms	1 ms		ON 0.5 ms OFF 0.5 ms
0.01 s 时钟脉冲	P_0_01s	0.01 s		ON 5 ms OFF 5 ms
0.02 s 时钟脉冲	P_0_02_s	0.02 s		ON 0.01 s OFF 0.01 s
0.1 s 时钟脉冲	P_0_1s	0.1 s		ON 0.05 s OFF 0.05 s
0.2 s 时钟脉冲	P_0_2s	0.2 s		ON 0.1 s OFF 0.1 s
1 s 时钟脉冲	P_1s	1 s		ON 0.5 s OFF 0.5 s
1 min 时钟脉冲	P_1min	1 min		ON 30 s OFF 30 s

● 使用时钟脉冲

下面例子以 0.5 s 的间隔接通和关闭 CIO 0.00。



指令	操作数
LD	P_1s
OUT	0.00



● 时钟脉冲刷新

即使在程序执行期间，时钟脉冲也会被刷新。程序开始和结束时的 ON/OFF 状态可能会不一致。

● 时钟脉冲错误

时钟脉冲中的最大错误为 0.01% （在 25 ℃时）。对于长期的基于时间的控制，我们建议使用内部时钟代替时钟脉冲。确保允许内部时钟的错误。

7

文件操作

本章描述了可对 CJ2 CPU 单元执行的文件操作。

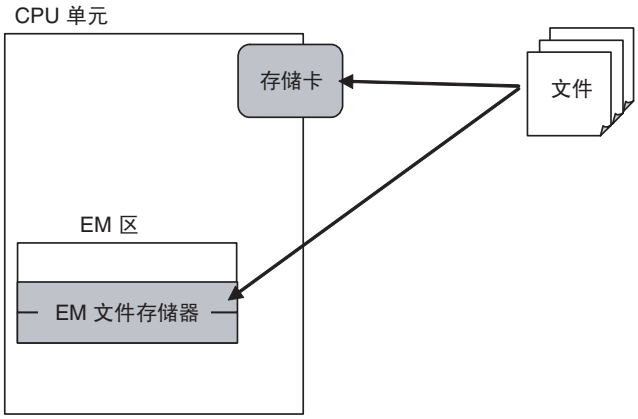
7-1 文件操作 7-2
7-1-1 文件存储器的类型 7-2
7-1-2 初始化文件存储器 7-3
7-1-3 存储卡注意事项 7-4
7-2 保存在文件存储器的文件类型 7-6
7-2-1 文件类型 7-6
7-2-2 创建和保存用于文件存储器的文件 7-9
7-3 文件存储操作 7-10
7-3-1 文件存储操作的类型 7-10
7-3-2 文件存储操作步骤和文件存储器文件 7-11
7-3-3 文件使用的限制 7-18
7-3-4 文件大小 7-18
7-3-5 支持软件和文件存储器文件之间的关系 7-20

7-1 文件操作

7-1-1 文件存储器的类型

文件存储器可用于保存 CJ 系列 PLC 中的文件。可使用两种类型的文件存储器。

- 存储卡
- EM 区中指定的范围，称为 EM 文件存储器



对于 CJ 系列的 PLC，存储卡和 EM 区的指定范围可作为存储器使用，以用于保存文件，或者用于将整个用户程序、 I/O 存储器和参数区作为文件保存。

种类	存储器类型	存储器容量	可储存的文件类型
存储卡	闪存存储器	HMC-EF183：128 M 字节 HMC-EF283：256 M 字节 HMC-EF583：512 M 字节	• 用户程序 • 注释文件 • 程序索引文件 • 符号（变量）表文件 • 参数文件 • 单元和板备份文件（仅存储卡）
内置 RAM（EM 文件存储器）	RAM	EM 区共有 13 组，包括从 EM 区中指定的组到 EM 区中最后的组。	

7-1-2 初始化文件存储器

存储卡

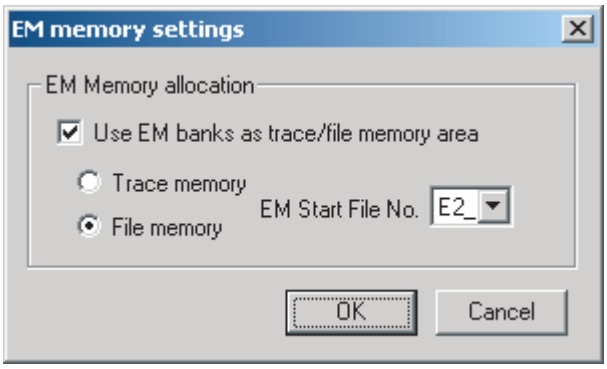
使用 CX-Programmer 中的存储卡对话框用于初始化存储卡。但是， HMC-EF □□□存储卡已经默认初始化，因此在第一次使用时，不需要进行初始化。

EM 文件存储器

根据以下步骤指定 EM 文件存储器的首个组作为文件存储器使用，并且初始化 EM 文件存储器。

1. 在 CX-Programmer 中，选择“PLC”-“存储器分配”（Memory Allocate）-“EM 存储器设置”（EM Memory Settings）。

2. 在 EM 存储器设置对话框中选择 “文件存储器”（File Memory），然后设置 EM 起始文件编号。

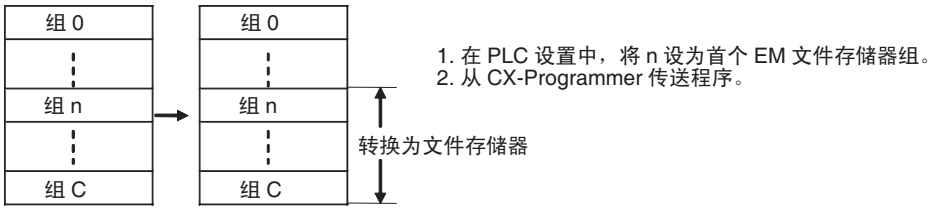


注　不可将 EM 区同时用于文件存储器和跟踪存储器。只能使用其中的一种功能。

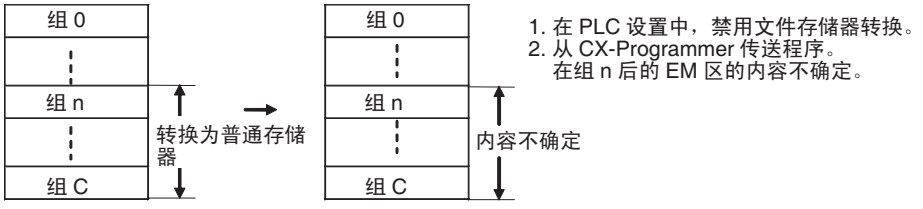
3. 在线连接到 CPU 单元，并传送用户程序，然后就可以使用 EM 文件存储器。

● 改变 EM 文件存储器设定

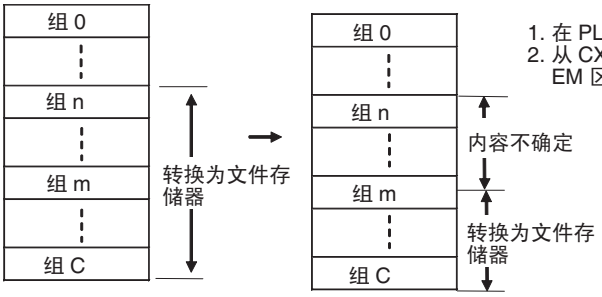
下图显示了将 EM 区从指定组转换为文件存储器的最后组。



下图显示了将用于文件存储器的 EM 区的组可恢复为普通 EM 区存储器。



文件存储器的起始组编号可更改。



1. 在 PLC 设置中，将 n 改为 m 作为首个 EM 文件存储器组。
2. 从 CX-Programmer 传送程序。
   EM 区中组 n ～ m 的内容是不确定的。

7-1-3 存储卡注意事项

在使用存储卡前须确定下列的项目。

格式化

出售前已格式化的存储卡，在购买后不必再对它们作格式化处理。对曾使用过的存储卡要作格式化处理，通常使用 CX-Programmer 在 CPU 单元中处理。如果存储卡已被笔记本电脑或其它的计算机直接格式化，则 CPU 单元可能不能识别此存储卡。如此类状况发生，即使它在 CPU 单元中被再格式化，也不能够再使用此存储卡。

根目录下的文件数

存放在存储卡中根目录下的文件数有一个限定（正如硬盘有一个限定一样）。虽然限定取决于存储卡的类型和格式，但是它将被限定在 128 和 512 个文件之间。当应用程序在特定的间隔以 LOG 文件或其它的文件编写时，文件应写在子目录下而不要放在根目录下。在计算机上或使用 CMND（490）指令可创建一个子目录。

写入次数

一般而言，对闪存存储器而言写操作的次数没有限制。然而对于存储卡，使用说明书中建议，写操作不超过 100,000 次。例如，如果存储卡每 10 分钟写入一次，则 100,000 次写操作相当于可运行 2 年时间。

文件最小长度

如果储存在存储卡上的 DM 区中仅包含许多诸如几个字数据的小文件，那么就不可能充分地利用存储卡容量。例如，如果使用 4,096 个字节长度单元的存储卡，不管文件有多小每个文件至少用去存储器的 4,096 字节。如果你把 10 个字的 DM 区数据存放在存储卡上，即使实际的文件大小只有 68 个字节，它同样占用存储器的 4,096 字节。使用类似小文件将会减少存储卡的利用率。但是，如果为了提高利用率而减少存储单元长度，则数据存取速率将会下降。

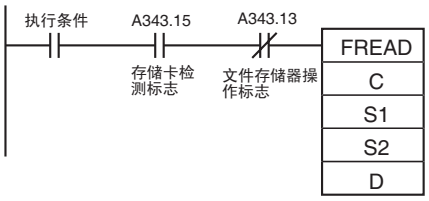
存储卡的单元分配大小可使用 DOS 中的 CHKDSK 命令来检查。特殊处理在此处不再详述，关于更多分配单元长度的信息，请参考普通计算机的相关信息。

存储卡识别时间

插入存储卡后，CPU 单元一般需要几秒时间来识别存储卡。所需的时间取决于 PLC 的循环时间、存储卡容量、保存在存储卡中的文件数量以及其他因素。例如，如果循环时间为 0.4 ms，并且所有 PLC 设置参数设为默认值，用于识别 HMC-EF183 存储卡的时间约为 8 s。

存储卡的存取

- 当“BUSY”指示灯点亮时（即指示灯显示存储卡正被存取），不可将 CPU 单元的电源关断 , 否则，存储卡就会损坏而无法使用。
- 当“BUSY”指示灯点亮时 , 不可拆卸 CPU 单元上的存储卡。先关断存储卡电源开关，并在“BUSY”指示灯熄灭后才能拆卸存储卡。否则，存储卡就会损坏而无法使用。
- 在存储卡被插入之后，需等待数秒时间以便 CPU 单元识别。当接通电源或插入存储卡之后立刻访问存储卡，必须把存储卡的确认标志 （A343.15）常开触点作为输入条件编入程序中，如下图所示。



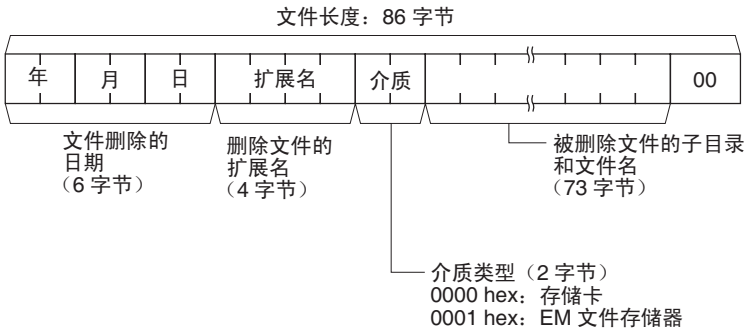
访问文件存储器期间电源中断

在 CPU 访问文件存储器 （存储卡或 EM 文件存储器）期间，如果电源中断，存储卡的内容可能不正确，正被更新的文件可能不能正确地重写。在某些情况下，存储卡本身可能被损坏。

此时，受其影响的文件在下一次电源接通时被系统自动删除。相应的文件删除通告标志 （存储卡为 A395.06， EM 文件存储器为 A395.07）将变为 ON。在下一次电源断开时，这个标志将变为 OFF。

当一个文件被删除时，在存储卡或 EM 文件存储器的根目录中将建立一个删除记录文件 （DELETE.TMP） 。用二进制编辑器可以检查以下信息：被删除文件的数据、存在的文件存储器 （介质）的类型、子目录、文件名和扩展名。当需要时，重新建立或重新复制被删除的文件。

下图表示了删除记录文件的结构



7-2 保存在文件存储器的文件类型

文件存储器可在下列应用中使用：

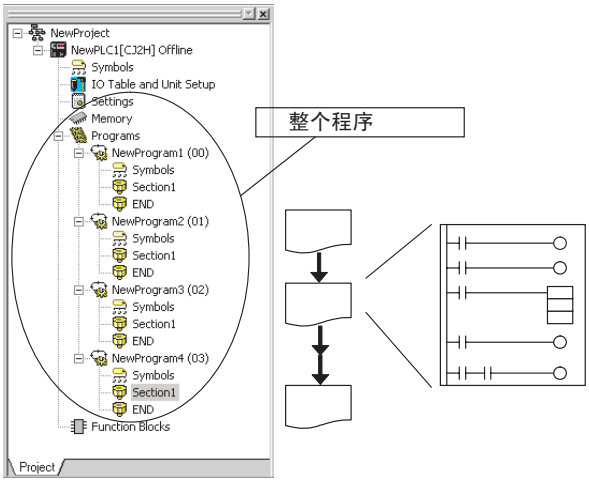
- 程序 / 网络符号 （变量）文件
- 参数文件
- 数据文件
- 符号 （变量）表文件 （除网络符号 （变量）外）
- 注释文件
- 程序索引文件
- 单元备份文件

7-2-1 文件类型

程序 / 网络符号 （变量）文件

程序 / 网络符号（变量）文件包含 CPU 单元的用户程序（在循环任务和中断任务中的程序）和网络符号（变量）（在全局符号 （变量）表中的网络符号 （变量））。

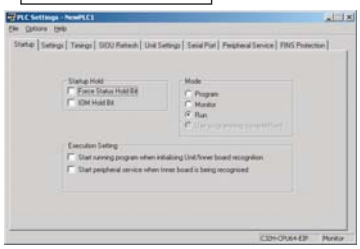
文件也包含了每个程序的属性。



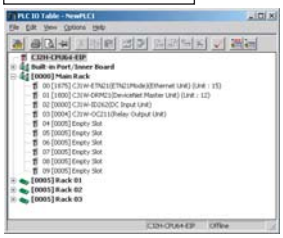
参数文件

参数文件包含 CPU 单元的内部参数区数据。参数区数据包含了 PLC 名称、PLC 设置、I/O 表、CPU 总线单元设定 （包括数据链接参数）和路由表。

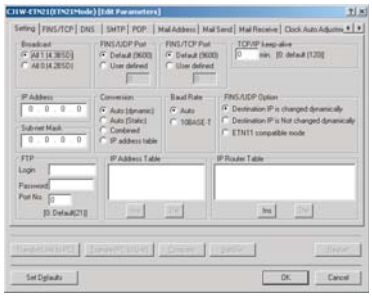
PLC 设置



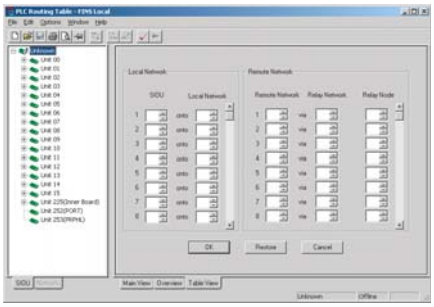
I/O 表



CPU 总线单元设定

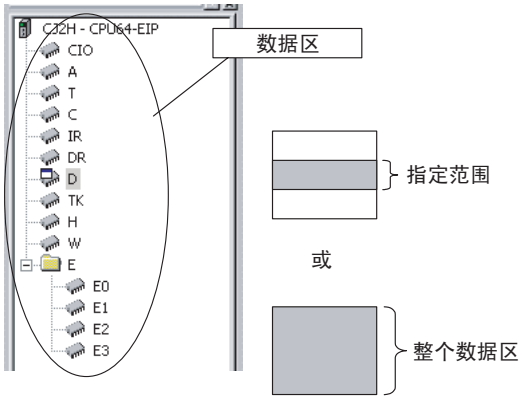


路由表



数据文件

数据文件包含一个 I/O 存储器数据区的数据，以字 （16 位）为单位。可以保存数据区的所有数据或只保存指定范围内的地址。以下 6 种数据区的任一种都可被保存：CIO、保持、工作、辅助、 DM 或 EM 区。



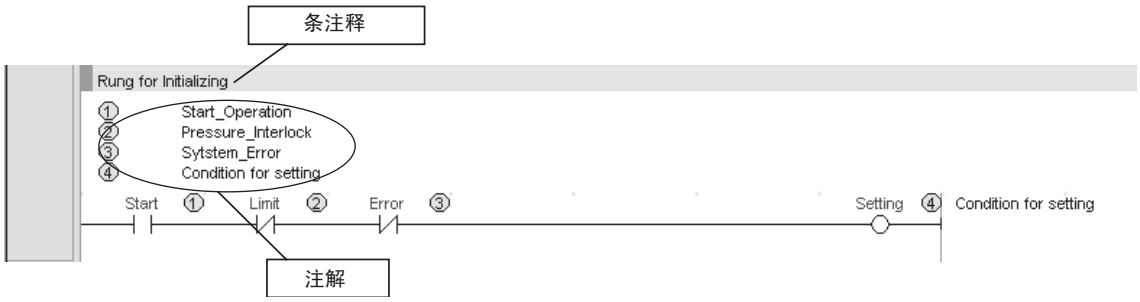
符号 （变量）表文件 （除网络符号 （变量）外）

符号 （变量）表文件 （除网络符号 （变量））包含在 CX-Programmer 中使用的符号 （变量）表信息以及自动分配的 PLC 地址。



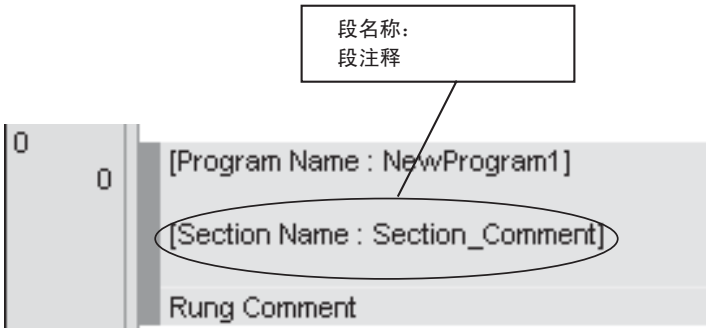
注释文件

注释文件包含在 CX-Programmer 中使用的注释信息。



程序索引文件

程序索引文件包含在 CX-Programmer 中使用的段信息。



单元备份文件

单元备份文件包含 PLC 单元的内部数据，该数据由简捷备份功能使用。当执行简捷备份操作时，这些文件被创建。各单元的内部数据被保存。

示例：

DeviceNet 单元：设备参数

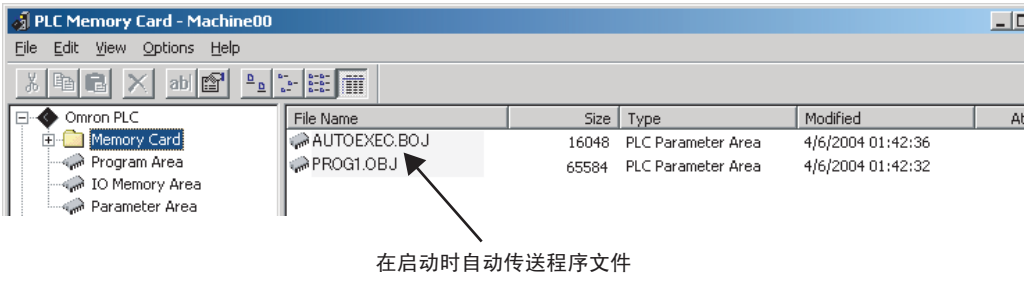
串行通信单元：协议宏数据

7-2-2 创建和保存用于文件存储器的文件

用户程序、 I/O 存储器和参数区数据

在 CX-Programmer 中使用 PLC 存储卡对话框用于传送数据。

1. 安装已经被初始化的 CPU 单元的存储卡或 EM 文件存储器。关于初始化 EM 文件存储器的信息，请参考 7-1-2 初始化文件存储器。
2. 在线连接 CX-Programmer 与 CPU 单元。
3. 在 PLC 存储卡对话框中选择传送目的地（存储卡或 EM 文件存储器），然后在文件菜单中选择“传送”。
   CPU 单元中的数据 （用户程序、 I/O 存储器和参数区）将在存储卡或 EM 文件存储器中作为文件保存。



符号 （变量）表、注释和程序索引

使用 CX-Programmer 传送数据。

1. 将已格式化的存储卡插入 CPU 单元或格式化 EM 文件存储器。关于初始化 EM 文件存储器的信息，请参考 7-1-2 初始化文件存储器。
2. 在线连接 CX-Programmer 与 CPU 单元。
3. 在 PLC 菜单中选择“传送”，然后选择“To PLC”（传送到 PLC）或“From PLC”（从 PLC 传送）。
4. 选择 “符号”（Symbols）或 “注释”（Comments）之一作为数据用于传送。

注　如果在 CPU 单元中安装了存储卡，数据只能传送到存储卡。（不能传送到 EM 文件存储器）。

7-3 文件存储操作

7-3-1 文件存储操作的类型

以下操作可用于文件存储器：

- 从 CX-Programmer 的操作
- 在启动时自动传送
- 简捷备份功能
- FREAD （700）和 FWRIT （701）指令
- 使用辅助区控制位替换全部程序
- FINS 命令

从 CX-Programmer 的操作

关于以下步骤的详细信息，请参考 CX-Programmer 操作手册。

● 在 CX-Programmer 的存储卡对话框中传送数据

以下文件可被创建、读取和写入。

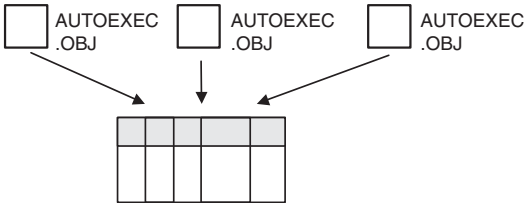
- 在启动时自动传送 （程序 / 网络符号 （变量）文件、数据文件、参数文件）
- 通用文件 （程序 / 网络符号 （变量）文件、数据文件、参数文件）
- CX-Programmer 文件 （符号 （变量）表、注释文件、程序索引文件）

● 在线 / 离线传送数据

通用文件 （程序 / 网络符号 （变量）文件、数据文件和参数文件）可在线传送至 PLC 或离线传送至计算机存储设备。

启动时自动传送文件

转换的程序、参数和 I/O 存储器可保存在存储卡中，并且在启动时能被存储卡读取。（详细信息，请参考第 10 章 CPU 单元功能）。



简捷备份操作

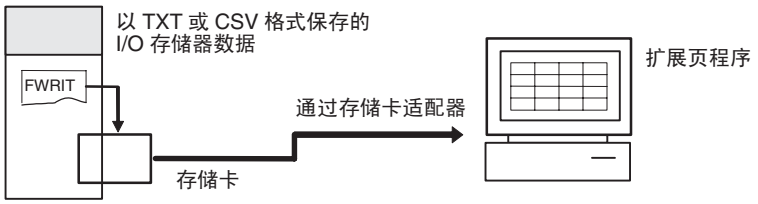
简捷备份操作允许备份和恢复所有 PLC 数据到存储卡或从存储卡恢复数据，包括 CPU 单元、特殊 I/O 单元和 CPU 总线单元中的数据，而无须使用编程设备。（详细信息，请参考 CJ2 CPU 单元硬件操作手册）。

备份恢复数据

所有 PLC 数据，包括 CPU 单元、特殊 I/O 单元和 CPU 总线单元中的数据，可通过 PLC 备份工具在计算机中备份和恢复。（详细信息，请参考 CJ2 CPU 单元硬件操作手册）。

FWRIT （701） /FREAD （700）指令

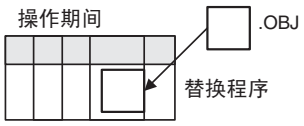
通过在用户程序中使用 FWRIT （701）指令，能够以文本格式或 CSV 格式将 I/O 存储器数据保存在存储卡或 EM 文件存储器中。可通过存储卡适配器将其传送到计算机，并能通过电子数据表程序进行编辑。



相反，类似特殊 I/O 单元设定的数据可通过电子数据表程序以文本或 CSV 格式创建，保存在存储卡，并通过 FREAD （700）指令读取到 CPU 单元。详细信息，请参考 CS/CJ/NSJ 系列指令参考手册 （样本编号 W474）。

使用辅助区控制位在操作期间替换用户程序

使用辅助区控制位可在操作期间替换整个用户程序的程序本身（无需编程设备）。（详细信息，请参考第 10 章 CPU 单元功能。）



FINS 命令

通过使用 FINS 命令可以控制文件存储器。可从使用上位链接（Host Link）协议连接的计算机处发送 FINS 命令。在用户程序中，网络中的另一台 PLC 可使用 CMND （490）发送 FINS 命令，本地 PLC 可使用 CMND （490）发送 FINS 命令。（详细信息，请参考通信命令参考手册）。

7-3-2 文件存储操作步骤和文件存储器文件

下表总结了每种类型的文件存储操作可控制的文件存储器文件。

读：从文件存储器传送文件到 CPU 单元。

写：从 CPU 单元传送文件到文件存储器。

OK：可能； ---：不可能

操作步骤	介质	描述	文件类型				
			程序 / 网络符号（变量）文件	数据文件	参数区数据	符号（变量）表、注释文件、程序索引文件	单元备份文件
从 CX-Programmer 的操作	存储卡、EM 文件存储器或计算机存储设备	创建	OK （通用文件） OK （启动时自动传送的文件）			OK （CX-Programmer 文件）	---
		读					
		写					
		其他操作					
启动时自动传送	存储卡	读	OK （启动时自动传送的文件）			---	
		写	---				
简捷备份功能	存储卡	读	OK （简捷备份文件）				
		写					
使用 PLC 备份工具备份或恢复	仅计算机存储设备	读	OK （PLC 备份文件）				
		写					
FREAD（700）和 FWRIT（701）指令	存储卡或 EM 文件存储器	从一个文件读取数据	---	OK （通用文件）	---	---	
		将数据写入到一个文件					
辅助区控制位在操作期间替代所有程序	存储卡	读	OK （通用文件）	---			
FINS 命令	存储卡或 EM 文件存储器	读	OK （通用文件）			---	
		写					
		其他操作					

启动时自动传送文件

这些文件用于批读取启动时的用户程序 / 网络符号（变量）、参数数据和 I/O 存储器。文件名固定为 AUTOEXEC 或 REPLACE。详细信息，请参考 10-3-5 在启动时自动传输。

备份文件

这些文件可通过使用 CPU 单元前部的 DIP 开关或存储卡的电源开关保存在存储卡中及备份所有 PLC 数据。文件名固定为 BACKUP □□。详细信息，请参考 CJ2 CPU 单元硬件操作手册（样本编号 W472）的 8-3 简捷备份中的信息。

PLC 备份工具文件

这些文件备份所有 PLC 数据，并通过 PLC 备份工具的备份和恢复操作进行创建。详细信息，请参考 CJ2 CPU 单元硬件操作手册 （样本编号 W472）的 8-2 使用计算机备份数据。

CX-Programmer 文件

这些文件由 CX-Programmer 创建。在文件存储器中创建的文件名是固定的。当项目被传送时，可选择以下存储器操作之一作为这些 CX-Programmer 文件的传送目的地。

- 注释存储器 （在 CPU 单元闪存存储器中）
- 存储卡
- EM 文件存储器

类型	名称	描述	注解
符号 （变量）表文件 （除网络符号（变量）外）	SYMBOLS.SYM	全局符号 （变量）表 （除网络符号（变量）外）和本地符号 （变量）表	• 包括 CX-Programmer 的全局 / 本地符号（变量）表中的变量、地址、数据类型和 I/O 注释信息。不包括网络符号 （变量）。 • 也包括在 CX-Programmer 的自动 PLC 地址分配中设定的数据。
注释文件	COMMENTS.CMT	条注释和注释 （注解）	包含 CX-Programmer 的条注释和注释信息。
程序索引文件	PROGRAM.IDX	节名称、节注释	包含 CX-Programmer 的段分隔符信息，但是分隔符位置取决于程序中的段分隔指令。

注　对于 CX-Programmer，上表中的符号（变量）表文件和注释文件（除网络符号（变量）外）可在 CX-Programmer 和计算机 RAM 之间以及在计算机 RAM 与存储器储存设备之间在线传送。

当项目从 CX-Programmer 传送至 CPU 单元时，符号 （变量）表、注释文件和程序索引文件可以保存在 CPU 单元内置的闪存存储器中。当执行简捷备份操作时，保存在注释存储器中的符号 （变量）表、注释文件和程序索引文件将自动创建为备份文件，并保存在存储卡中。

通用文件

可使用 CX-Programmer、 FINS 命令、指令和辅助区位对这些文件执行读、写和其他操作。

文件由用户自定义。

文件类型	文件名 *1	扩展名	内容	描述	
程序 / 网络符号（变量）文件	********	.OBJ	• 用户程序 • 网络符号（变量）	CPU 单元的循环任务和中断任务中的程序和网络符号 （变量）。	
参数文件	********	.STD	• PLC 设置 • 注册 I/O 表 • 路由表 • CPU 总线单元设置区	• CPU 单元中的参数区数据 • 用户不必区分在文件中的数据类型。	
数据文件	********	.IOM	I/O 存储器指定区域	• 从一个区的起始字到末尾字的字数据（16 位）递增。 • 可以是 CIO、 HR （保持区）、 WR （工作区）、 AR（辅助区）、 DM （数据存储器）或 EM （扩展存储器）区。	二进制格式
		.TXT			文本格式 （非分隔符或制表符） *2
		.CSV			CSV 格式 （逗号分隔）

*1 上述 “********”表示的文件名由 8 位或以下的 ASCII 字符构成。

*2 文本和 CSV 数据文件只能通过 FWRIT （701）和 FREAD （700）指令进行读写。不能通过 CX-Programmer 读写。

● 通用数据文件

数据文件类型

通用数据文件的文件扩展名有三种：IOM 、 TXT 和 CSV。

扩展名	内容	用途
.IOM	二进制格式 CS/CJ 系列数据格式	I/O 存储器备份
.TXT*1	在这些数据格式中， I/O 存储器的 1 字段或 2 字段转换为 ASCII 码。记录可使用回车分隔。	通过电子数据表软件交换数据
.CSV		

*1 读 / 写 TXT 和 CSV 数据文件：
TXT 和 CSV 数据文件只能通过 FREAD （700）和 FWRIT （701）进行读写。

以下 6 种数据格式在文本和 CSV 文件中使用。

扩展名	数据格式	内容	
		每个字段的 I/O 存储器长度	分隔符
.TXT*1	无分隔字	1 字	无
	无分隔双字	2 字	无
	制表符分隔字	1 字	制表符代码
	制表符分隔双字	2 字	制表符代码
.CSV*1	逗号分隔字	1 字	逗号
	逗号分隔双字	2 字	逗号

*1 有关字符的注意事项：
如果 TXT 或 CSV 文件包含的不是十六进制字符（0 ～ 9、A ～ F 或 a ～ f），则不能把数据正确地写到 I/O 存储器中。

- 字段长度注意事项：
  在使用单字时，如果 TXT 或 CSV 文件字段不是 4 位十六进制数，则不能把数据正确地写到 I/O 存储器中。同样在使用双字时，如果 TXT 或 CSV 文件字段不是 8 位十六进制数，则也不能把数据正确地写到 I/O 存储器中。
- 存储顺序：
  在使用单字时，I/O 存储器数据被转换成 ASCII 码并以一个字段从最低到最高的 I/O 存储器地址按顺序储存。
  在使用双字时，I/O 存储器数据被转换成 ASCII 码并以两个字段从最低到最高的 I/O 存储器地址按顺序储存。（2 字段以内数据，首先储存高地址字，然后储存低地址字）。
- 分隔符：
  没有分隔符时，字段被连续排列然后储存。当被逗号分隔时，在储存之前，逗号被插入字段之间。当被制表符分隔时，在储存之前，制表符代码被插入字段之间。
  当分隔符（逗号或制表符）被 FREAD（700）指定，数据则以一字段的（逗点或制表符）分隔形式被读取。
- 回车：
  当未使用回车时数据是连续捆包的。
  当使用了回车时，指定数量的字段后将插入一个回车代码。若文件中使用了回车，则从文件开头（开始读取字或开始写入字）的偏移不可在 FREAD （700） /FWRIT （701）指令中指定。
- 字段数：
  文件中数据的总量取决于在 FWRIT （701）指令中指定的字段数 （写入项目数）以及字 / 字段的数量。
  长度由字段数指定。
  对于字数据， I/O 存储器中的 1 字 =1 字段；
  对于长字数据， I/O 存储器中的 2 字 =1 字段。

数据文件名称

数据文件不指明保存的是什么数据，如保存的是哪个存储区的数据。因此必须给文件一个可表示其内容的文件名以便于文件管理，如下例所示。
例：D00100.IOM、 CIO0020.IOM
即使写入到数据文件的原始数据（IOM、TXT 或 CSV）是来自不相同的区，文件开始的数据总是写入到指定的 I/O 存储器中的起始地址。例如：如果通过一个编程设备把某个文件中的 CIO 数据写入 DM 区，CPU 单元的 DM 区读取这些数据时并不会出现任何关于区域不同的提示。

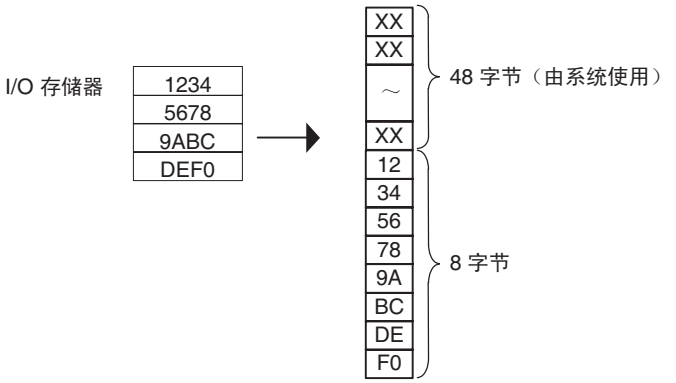
附加信息

包含十六进制数 （0 ～ 9， A ～ F）的 TXT 和 CSV 格式的数据文件允许 I/O 存储器中的数字数据与电子数据表程序进行数据交换。

I/O 存储器中的 ASCII 数据可以通过 TWRIT（704）（写文本文件）梯形图编程指令或 ST 语言中的存储卡处理功能 WRITE_TEXT （ ），作为文本文件保存。

附加信息

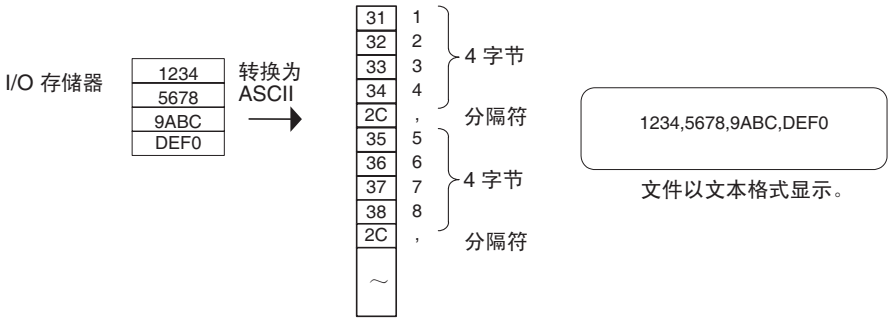
下面的例子给出了 I/O 存储器中包含四个字的数据文件 （ABC.IOM）的二进制数据结构：1234 Hex、 5678 Hex、 9ABC Hex 和 DEF0 Hex。然而用户在通常操作时并不需要考虑数据格式。



ABC.IOM 的内容

- 单字段的 CSV 和 TXT 数据文件结构

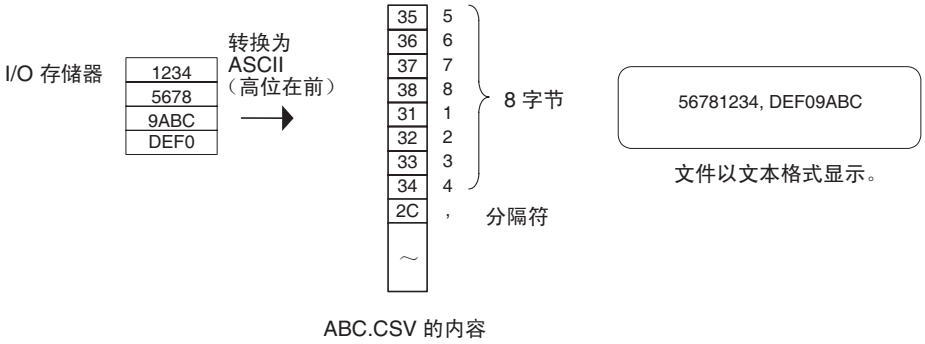
下面的例子给出了 I/O 存储器中包含四个字的单字段 CSV 数据文件 （ABC.CSV）的数据结构：1234 Hex、 5678 Hex、 9ABC Hex 和 DEF0 Hex。



ABC.CSV 的内容

- 双字段的 CSV/TXT 数据文件结构

下面的例子给出了 I/O 存储器中包含四个字的双字段 CSV 数据文件 （ABC.CSV）的结构：1234 Hex、 5678 Hex、 9ABC Hex 和 DEF0 Hex。



附加信息

用电子数据表软件创建数据文件

使用电子数据表 （如：Microsoft Excel）软件创建 TXT 和 CSV 数据文件须按下列步骤进行。

- 设定单元内容为字符。
- 若为单字段，则每个单元输入 4 个字符；若为双字段，每个单元输入 8 个字符。例如，单字段时应输入 000A，不能仅输入 A。
- 确定每单元输入的只能是十六进制字符（0 ～ 9、A ～ F 或 a ～ f）。其它字符和代码一概不能使用。

若要把十六进制数保存在 I/O 存储器中，把电子数据表的十进制输入转换成十六进制是很有帮助的。按照下列步骤进行十六进制转换。

(1) 从工具菜单中选择 “Add-Ins...”。

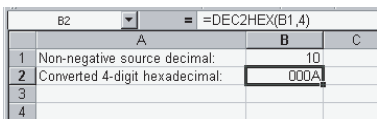
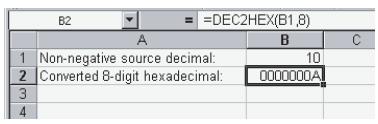
(2) 从 Add-Ins 菜单中选择 “Analysis ToolPak”。

(3) 从使用功能单元的插入菜单中选择 “Function”。

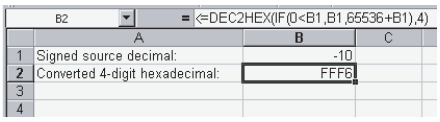
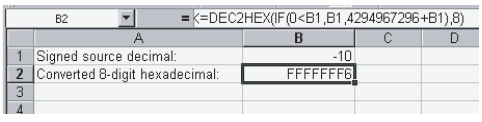
(4) 从工程分类字段选择 “DEC2HEX （数值，数字）”。

(5) 若需转换 4 位十六进制数时，输入下列数值变量：IF （0<=cell location,cell location,65535+cell location）
若需转换 8 位十六进制数时，输入下列数值变量：IF（0<=cell location,cell location,4294967296 +cell location）

例 1：非负数十进制数值的输入

项目	将无符号十进制转换成 4 位十六进制数	将无符号十进制转换成 8 位十六进制数
功能使用	DEC2HEX （cell_location,4）	DEC2HEX （cell_location,8）
示例	输入十进制数 10，并转换成 4 位十六进制数 000A。 	输入十进制数 10，并转换成 8 位十六进制数 0000000A。 

例 2：带符号十进制数值的输入

项目	将无符号十进制转换成 4 位十六进制数	将无符号十进制转换成 8 位十六进制数
功能使用	DEC2HEX （IF （0<=cell_location, cell_location, 65536+cell_location） ,4）	DEC2HEX （IF （0<=cell_location,cell_location, 4294967296+cell_location） ,8）
示例	输入十进制数 -10，并转换成 4 位十六进制数 FFF6。 	输入十进制数 -10，并转换成 8 位十六进制数 FFFFFFF6。 

7-3-3 文件使用的限制

在 DOS 中可对文件格式化，因此它可在带有 Windows 操作系统的计算机上作为常规文件使用。

● 文件名

文件可通过文件名和扩展名加以识别，如下所示。文件名可使用下列的字符写成：字母 a ～ z、 A ～ Z、数字 0 ～ 9、！ 、 &、 $、 #、 '、 {、 }、 -、 ^、 (、) 和 _。

下列的字符不能用于文件名：,, ., /, \, ?, *, “, :, ;,<、 >、 =、 +、空格和 2 个字节的字符。

文件扩展名取决于储存文件的类型。

● 目录

CS/CJ 系列 CPU 单元可以访问子目录中的文件。存储器中的文件可放置在指定在目录中，一个目录中最多可达 5 个子目录 （包括根目录），使用手持编程器的情况除外。目录路径的最大长度为 65 个字符。当使用类似 Windows 的操作系统创建存储卡子目录，请勿超出最大子目录个数 （5 个子目录）。

7-3-4 文件大小

以字节表示的文件大小可通过下表中所列的公式计算得出。

文件类型	文件大小
数据文件 （.IOM）	（字数× 2）+ 48 字节 例如：整个 DM 区 （D0 ～ D32767） （32,768 字× 2）+ 48 = 65,584 字节

文件类型	文件大小
数据文件 （.TXT 或 .CSV）	文件大小取决于分隔符和回车符的使用数。分隔符代码是 1 个字节，而回车符代码是 2 个字节。 例 1：无分隔字，没有回车 123456789ABCDEF012345678 占用 24 个字节。 例 2：每 2 个字段分隔字回车。 1234,5678↵ 9ABC,DEF0↵ 1234,5678↵ 占用 33 个字节。 例 3：每 2 个字段分隔双字回车。 56781234,DEF01234↵ 56781234↵ 占用 29 个字节。
程序文件 （.OBJ）	（使用的步数× 4）+ 48 字节 *1
参数文件 （.STD）	16,048 字节

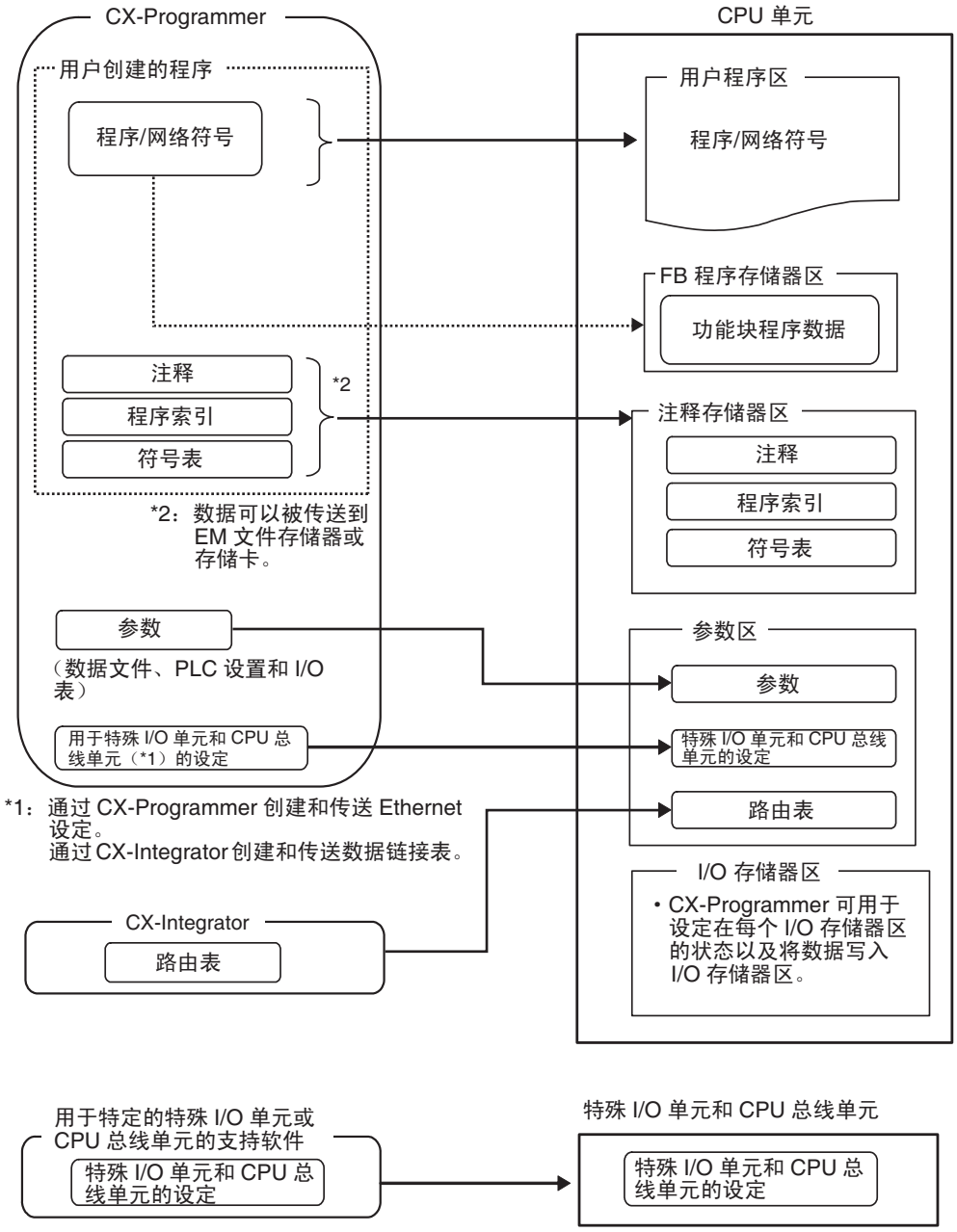
*1 在程序文件中，通过把全部 UM 步数减去可利用的 UM 步数来计算步数。这些值在 CX-Programmer 的对比报告（Cross-Reference）中给出。详细信息，请参考 CX-Programmer 操作手册。

7-3-5 支持软件和文件存储器文件之间的关系

下表给出了每个支持软件包文件和文件存储器文件之间的关系。

支持软件	支持软件文件	内容	文件存储器文件	保存区
CX-Programmer	CX-Programmer 项目文件 （扩展名：.cxp）	用户程序 / 网络符号 （变量）	程序 / 网络符号 （变量）文件 （扩展名：.OBJ）	CPU 单元
		功能块		
		符号 （变量）表 （包括符号名称和对应地址、 I/O 注释） 注：不包括网络符号 （变量）	符号 （变量）表 （除网络符号 （变量）外）（SYMBOLS.SYM）	
		注释 （条注释和注解， I/O 注释除外）注：程序索引被省略	注释文件 （COMMENTS.CMT）	
		PLC 名称	参数文件 （扩展名：.STD）*1	
		PLC 设置		
		注册的 I/O 表		
		Ethernet 单元设置		
	数据链接表文件 （.cl2）	Controller Link 数据链接表		
	CX-Programmer 项目文件 （扩展名：.cxp） 单元设定 （XML 文件）：CPS 内容	PLC 存储器 （I/O 存储器） （分配的 I/O 存储器和以下分配的 DM 区字）	数据文件 （扩展名：.IOM）	
		分配的 I/O 存储器和分配的 DM 区字		
CX-Integrator	路由表文件 （扩展名：.rtg）	路由表	参数文件 （扩展名：.STD）*1	
FL-net 支持软件	特殊文件 （.csv）	FL-net 设定		
用于特定的特殊 I/O 单元或 CPU 总线单元的支持软件	文件由特定的支持软件创建用于特殊 I/O 单元和 CPU 总线单元	数据不包含在以上文件中	单元备份文件 （BACKUP □□ .PRM）	特殊 I/O 单元和 CPU 总线单元

*1 在文件存储器中，使用支持软件创建的文件都将以单一文件的形式来创建。


CX-Programmer
用户创建的程序
程序/网络符号
注释
程序索引
符号表
*2
*2：数据可以被传送到EM 文件存储器或存储卡。
参数
（数据文件、PLC 设置和 I/O 表）
用于特殊 I/O 单元和 CPU 总线单元（*1）的设定
*1：通过 CX-Programmer 创建和传送 Ethernet 设定。
通过 CX-Integrator 创建和传送数据链接表。
CX-Integrator
路由表
CPU 单元
用户程序区
程序/网络符号
FB 程序存储器区
功能块程序数据
注释存储器区
注释
程序索引
符号表
参数区
参数
特殊 I/O 单元和 CPU 总线单元的设定
路由表
I/O 存储器区
• CX-Programmer 可用于设定在每个 I/O 存储器区的状态以及将数据写入 I/O 存储器区。
用于特定的特殊 I/O 单元或 CPU 总线单元的支持软件
特殊 I/O 单元和 CPU 总线单元的设定
特殊 I/O 单元和 CPU 总线单元
特殊 I/O 单元和 CPU 总线单元的设定

8

I/O 分配和单元设定

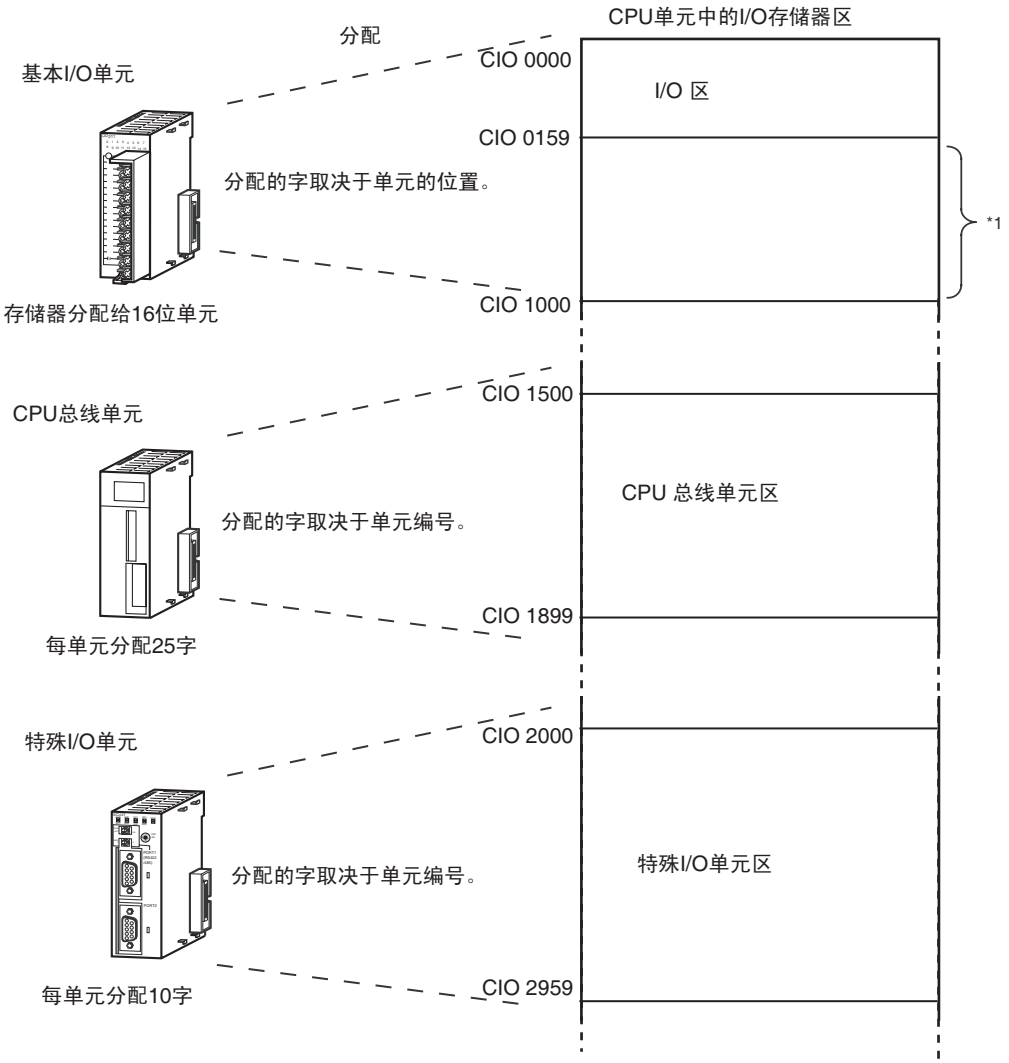
本章描述了存储器如何被分配到基本 I/O 单元、特殊 I/O 单元和 CPU 总线单元，以及如何设定特殊 I/O 单元和 CPU 总线单元。

8-1 I/O 分配 8-2
8-1-1 I/O 分配 8-2
8-1-2 自动分配 8-5
8-1-3 手动分配 8-9
8-1-4 I/O 表错误和注意事项 8-17
8-2 设定 CPU 总线单元和特殊 I/O 单元 8-20
8-2-1 设定参数 8-20
8-2-2 数据交换 8-24

8-1 I/O 分配

8-1-1 I/O 分配

用于在 CPU 单元和其他单元之间交换数据的 CPU 单元 I/O 存储器中的分配字被称为 I/O 分配。存储器被不同地分配给基本 I/O 单元、特殊 I/O 单元和 CPU 总线单元。



*1 每个机架的首字可通过CX-Programmer设为一个在CIO 0000和CIO 0999之间的地址（默认设定：从CIO 0000开始的连续分配）。

I/O 分配方法

使用以下两种方法之一分配 I/O。

- 自动分配 （默认）
  当电源为 ON 时， I/O 将根据连接单元的状态进行分配 （例如槽位置和单元编号）。
- 手动分配 （I/O 表的创建）
  通过 CX-Programmer 创建 I/O 表用于指定所需的分配。如果 I/O 表被创建，当 CPU 单元电源为 ON 时，I/O 表与连接单元的状态不符，将发生错误。

下表显示了两种方法之间的不同。根据目的和系统需要选择方法。

分配方法 / 项目	自动分配	手动分配	
		由连接的单元创建 I/O 表	由用户自定义分配创建 I/O 表
目的	• 不需要检查不正确的单元时使用该方法。 • 不需要用户自定义分配时使用该方法 （包含预留不使用的字）。	使用该方法用于检查不正确的单元。	• 使用该方法创建用户自定义分配 （包含预留不使用的字）。 • 使用该方法分配不带实际 I/O 单元的 I/O （离线）。
创建 I/O 表	不需要	需要	
根据 I/O 表检验连接的单元	不检验单元	检验单元 （如果连接的单元与 I/O 表不一致，则不会开始操作）。	
用于分配 I/O 的操作	自动分配	手动分配	
CX-Programmer	不需要	需要	
连接创建 I/O 表所需的单元	需要	需要	不需要
分配方法	自动分配 （以 CPU 单元中的顺序）		手动分配 （可对每组指定首地址）。

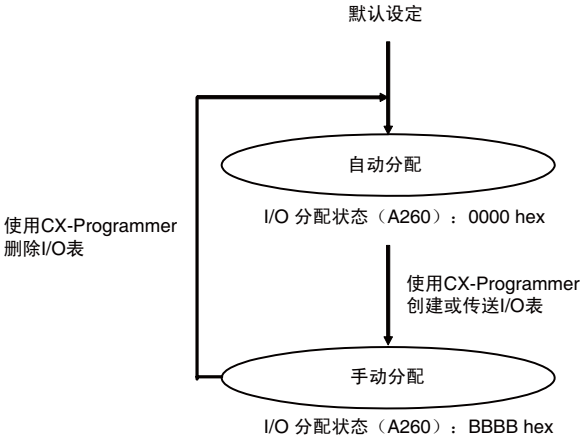
● 检查 I/O 分配方法

可使用 I/O 分配状态字 （A260）来检查 I/O 分配被设为手动还是自动。

地址	名称	内容	
A260	I/O 分配状态	0000 hex	自动 I/O 分配
		BBBB hex	手动分配

● I/O 分配状态转换

在下面的方式中将改变 I/O 分配方法。



使用 CX-Programmer 删除 I/O 表时，在 I/O 分配状态返回自动分配的同时，每个机架的首字将被清除。此外， CPU 单元的系统设置区将被初始化。

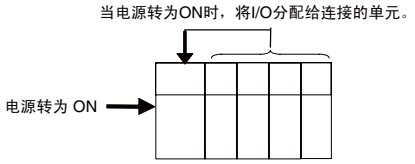
8-1-2 自动分配

自动分配 （默认）

使用该方法时，根据连接单元的状态分配 I/O。 I/O 被自动分配，因此不需要从编程设备进行操作。

使用自动分配时，每次供应给 PLC 的电源转为 ON 时， I/O 分配将被更新。

如果要将手动分配改为自动分配，需使用编程设备删除 I/O 表。



附加信息

与手动分配 （I/O 表不是由自动分配创建的）不同，因此不检验连接的单元。

分配方法

对于自动分配，如下表所示， I/O 将被分配给连接的单元。不需要进行用户操作。

单元分配	分配方法
基本 I/O 单元	根据位置进行分配
特殊 I/O 单元	根据前面板旋转开关设定进行分配 （即单元编号）
CPU 总线单元	

每个单元分类的 I/O 分配方法

本节描述每个单元分类（基本 I/O 单元、特殊 I/O 单元和 CPU 总线单元）的 I/O 分配方法。

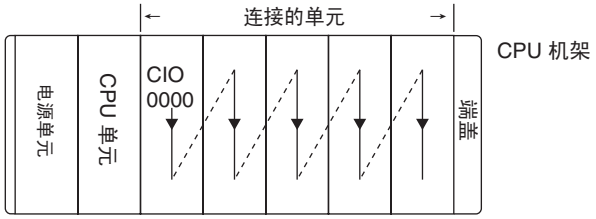
● 基本 I/O 单元的 I/O 分配

给 CJ 系列基本 I/O 单元分配的是 I/O 区中的字（CIO 0000 ～ CIO 0159）。基本 I/O 单元可安装到 CPU 机架或扩展机架上。可以使用以下方法。

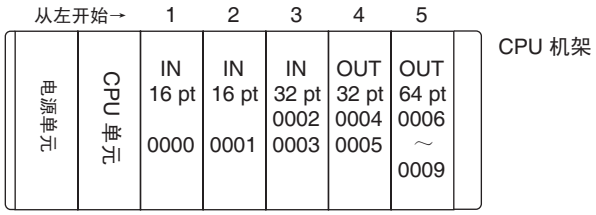
CPU 机架上的基本 I/O 单元

从距离 CPU 单元最近的单元开始从左到右，依次分配字到 CPU 机架上的基本 I/O 单元。每个单元都分配到其所需数量的字。

注　具有 1 ～ 16 点 I/O 的单元被分配到 16 位，而具有 17 ～ 32 点 I/O 的单元被分配到 32 位。



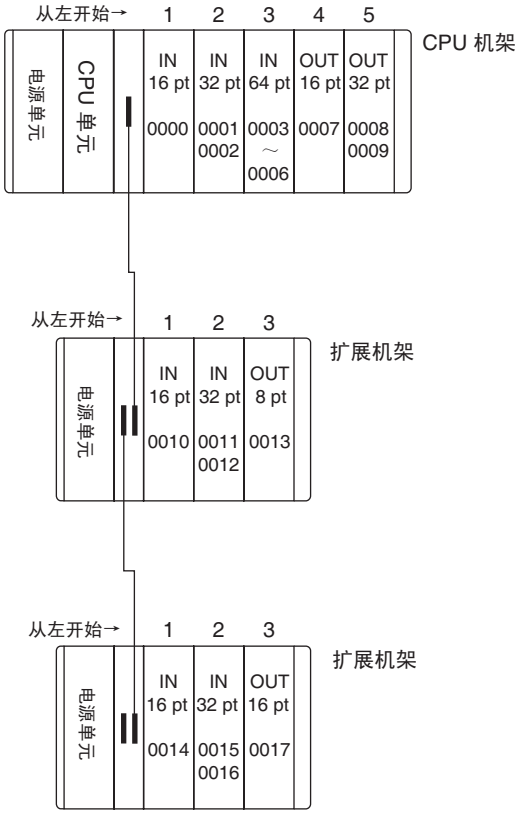
例



扩展机架上的基本 I/O 单元

基本 I/O 单元的 I/O 分配从 CPU 机架延续到直接连接到 CPU 机架上的扩展机架。与 CPU 机架上的单元一样，字从左到右分配且每个单元都分配到其所需数量的字。

例



● 特殊 I/O 单元的 I/O 分配

每个 CJ 系列特殊 I/O 单元根据在单元中所设的单元编号被分配到特殊 I/O 单元区 （CIO 2000 ～ CIO 2959）的 10 个字。特殊 I/O 单元可以安装到 CPU 机架或扩展机架上。下表显示了特殊 I/O 单元中的哪些字被分配到每个单元。

单元编号	字数	分配的字
0	10	CIO 2000 ～ CIO 2009
1	10	CIO 2010 ～ CIO 2019
2	10	CIO 2020 ～ CIO 2029
:	:	:
15	10	CIO 2150 ～ CIO 2159
: :	: :	: :
95	10	CIO 2950 ～ CIO 2959

在对基本 I/O 单元进行 I/O 分配时，与特殊 I/O 单元无关，含有特殊 I/O 单元的位置不分配 I/O 区中的任何字。

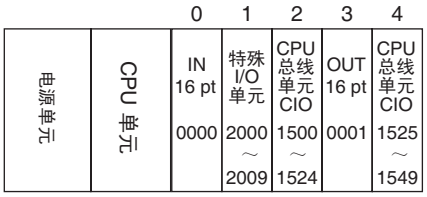
● CPU 总线单元的 I/O 分配

每个 CJ 系列 CPU 总线单元根据在单元中所设的单元编号被分配到 CPU 总线单元区（CIO 1500 ～ CIO 1899）的 25 个字。 CPU 总线单元可以安装到 CPU 机架或扩展机架上。下表显示了 CPU 总线单元中的哪些字被分配到每个单元。

单元编号	字数	分配的字
0	25	CIO 1500 ～ CIO 1524
1	25	CIO 1525 ～ CIO 1549
2	25	CIO 1550 ～ CIO 1574
:	:	:
15	25	CIO 1875 ～ CIO 1899

在对基本 I/O 单元进行 I/O 分配时，与 CPU 总线单元无关。含有 CPU 总线单元的位置不分配 I/O 区中的任何字。CPU 总线单元的单元编号与特殊 I/O 单元的单元编号不同。对 CPU 总线单元和特殊 I/O 单元单元使用相同的单元编号将不会导致单元编号的重复使用。

例



槽位	单元	型号	需要的字	分配的字	单元编号	组
0	16 点 DC 输入单元	CJ1W-ID211	1	CIO 0000	---	基本 I/O 单元
1	模拟量输入单元	CJ1W-AD081	10	CIO 2000 ～ CIO 2009	0	特殊 I/O 单元
2	串行通信单元	CJ1W-SCU41	25	CIO 1525 ～ CIO 1549	1	CPU 总线单元
3	16 点晶体管输出单元	CJ1W-OD211	1	CIO 0001	---	基本 I/O 单元
4	Controller Link 单元	CJ1W-CLK21	25	CIO 1550 ～ CIO 1574	2	CPU 总线单元
---	CPU 单元上的内置 EtherNet/IP 端口 *1	CJ2H-CPU68-E1P	25	CIO 1500 ～ CIO 1524	0	CPU 单元 （端口视为 CPU 总线单元）

*1 对于 CJ2 CPU 单元，字被分配到作为 CPU 总线单元的内置 EtherNet/IP 端口。这些字用于保存端口的网络通信状态。同样对于其他 CPU 总线单元，为根据对 CJ2 CPU 单元前部的旋转开关所设的单元编号来分配字。

8-1-3 手动分配

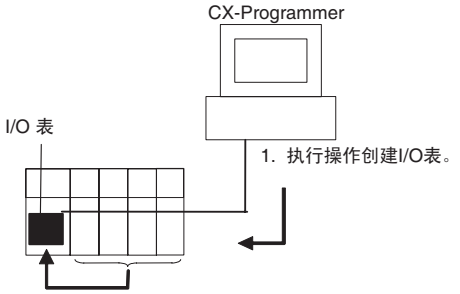
手动分配

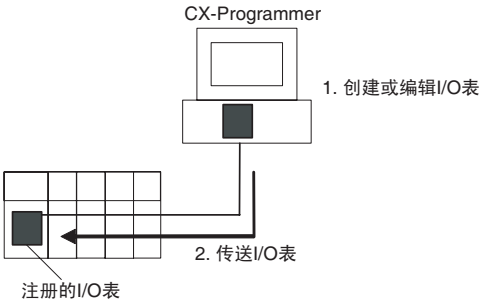
使用手动根据用户创建的 I/O 表来分配 I/O。如果 I/O 表被创建，当电源打开时，将根据 I/O 表检验连接的单元。

● 创建 I/O 表

使用以下方法之一创建 I/O 表。

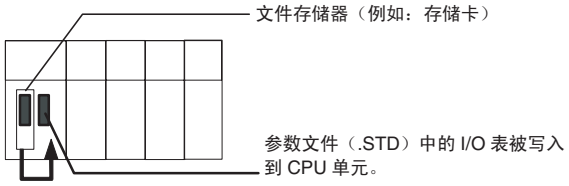
根据连接的单元创建 I/O 表	根据用户自定义分配创建 I/O 表
在线连接 CX-Programmer 时，根据连接的单元创建 I/O 表。	CX-Programmer 离线时，使用 CX-Programmer 创建所需的 I/O 表，并将其传送到 CPU 单元。





附加信息

可将 PLC 设为电源打开时，自动将保存在存储卡中的 I/O 表传送到 CPU 单元。



● 单元检验

如果创建了 I/O 表，启动时会对注册的 I/O 表与实际 I/O 进行比较。如果它们不匹配，A401.10 将转为 ON 来表示 I/O 设定错误，并且不能进行运行。

根据连接的单元创建 I/O 表

● 分配方法

分配 I/O 的方法与自动分配相同。关于分配方法的信息，请参考 8-1-2 自动分配。

● I/O 分配步骤

CX-Programmer 在线时，进行以下步骤用于对连接的单元创建 I/O 表。

1. 在主窗口的工程树中双击 “IO 表”。将显示 I/O 表窗口。
2. 选择 “选项”然后选择 “创建”。安装到机架的单元型号和位置将被写入到 CPU 单元作为注册的 I/O 表。

● 比较 I/O 表的步骤

CX-Programmer 在线时，进行以下步骤用于比较 I/O 表与实际连接的单元。

1. 在主窗口的工程树中双击 “IO 表”。将显示 I/O 表窗口。
2. 选择 “选项”然后选择 “检验”。安装到机架的单元型号和位置将与注册的 I/O 表进行比较。
3. 检查显示的结果。

根据用户自定义分配创建 I/O 表

● 分配方法

与根据连接单元的自动分配 I/O 不同，使用以下方法可手动分配 I/O 来创建所需的 I/O 表。

- 将字分配到单元机架的任意位置
- 预留字
- 指定机架上的首字
- 不根据实际单元分配 I/O

字被分配如下：

单元分类	分配方法
基本 I/O 单元	对设定有一些限制，但可进行以下设定： 可以指定分配到任意机架上的任何单元的字。连续的字将根据它们连接的次序被分配到以下单元。这样可以创建 63 个用户自定义分配组。
特殊 I/O 单元	根据前面板旋转开关设定 （即单元编号）来分配字。
CPU 总线单元	

● I/O 分配步骤

使用 CX-Programmer 创建 I/O 表，然后将其传送到 PLC。离线创建 I/O 表，然后在线将其传送到 PLC。

1. 在主窗口的工程树中双击 “IO 表”。将显示 I/O 表窗口。
2. 双击要进行编辑的机架。将显示所选机架的槽位。
3. 右键要分配的单元的槽位，并从下拉菜单中选择单元。
4. 编辑 I/O 后，通过选择“选项”-“传送到 PLC”（Options - Transfer to PLC）将其传送到 CPU 单元。

● 给机架位置分配首字以及预留未使用的字

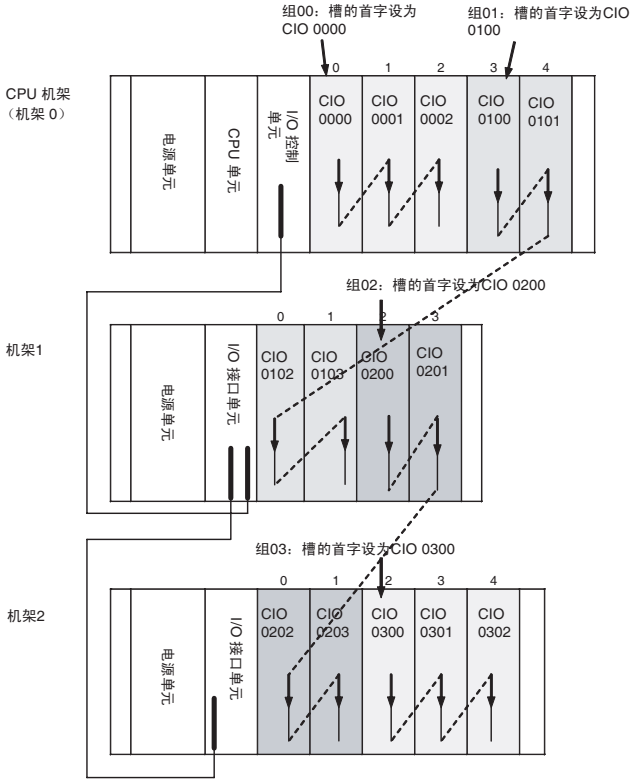
可使用 CX-Programmer 对任意机架中任意位置的单元设置首字。在那之后，指定的首字后面的连续的字将以单元连接的次序分配到下一个单元。可通过在每组中设定首个单元的首字将其分为 63 个组。

分配方法

对机架位置（称为“槽位”）设定首字时，必须对 CPU 机架上的槽位 00 设定首字。然后可对任意机架上的任意槽位设定首字，共可设定 63 个其他的槽位。

- 对每个槽位设定的首字，以创建由其槽位开始的组。
- 从首个单元的组中指定的字开始，对字进行分配，并继续从左到右将连续的字分配到每个单元，直到下一个组（即直到设定了槽位首字的下一个单元）。下一组可从同一机架或随后的机架开始。
- 对于组00，可对CPU机架的槽位00设定首字。对于组01～63，可对任意机架上的任意槽位设定首字。

例如，在每个机架的中间设定槽位的首字。仅 16 点单元可被使用。



槽位首字的设定

组	机架	槽位	字
00 *	CPU 机架	00	CIO 0000
01	CPU 机架	03	CIO 0100
02	机架 1	02	CIO 0200
03	机架 2	02	CIO 0300

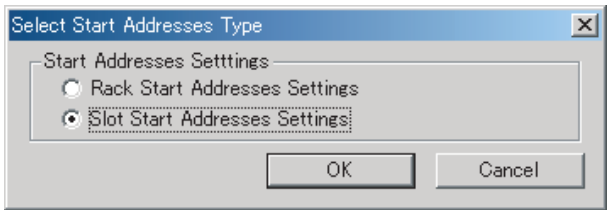
* 组 00 必须从 CPU 机架上的槽位 00 开始。可以设定任何字。可对组 01 ～ 63 设定任何机架上的任何槽位。

从 CX-Programmer 设置槽位的首字

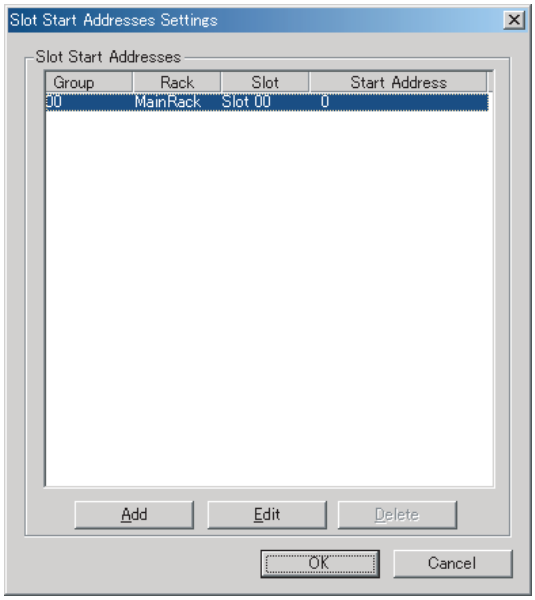
进行以下步骤来设定机架首字。

1. 从 I/O 表窗口上的选项（Option）菜单中选择“机架 / 槽位起始地址”（Rack/Slot Start Addresses）将显示下面的对话框。

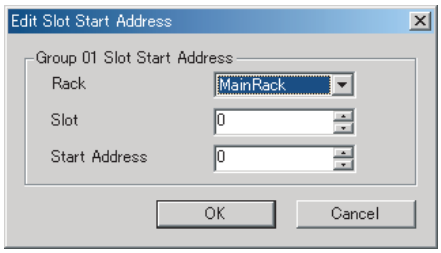
2. 选择 “槽位起始地址设定”（Slot Start Address Settings）选项并点击 “OK”按钮。



3. 在出现的对话框中，设定 CPU 机架槽位 00 的首字。



4. 要从 CIO 0000 改变设定，点击“编辑”（Edit）按钮，将出现以下对话框。



5. 设定所需的字并点击“OK”按钮。

6. 要设置其它组的槽位首字，点击“添加”（Add）按钮并对机架、槽位和字作适当的设定。

设定	设定范围	默认值	备注
组	00 ～ 63	00	组编号按组显示和设置的顺序自动分配。
机架	CPU 机架 （“主机架”） 机架 1 ～ 7	CPU 机架	组 00 总是从 CPU 机架上的槽位 00 开始。
槽位	00 ～ 03	0	
首字	0 ～ 999	0	---

给机架分配首字

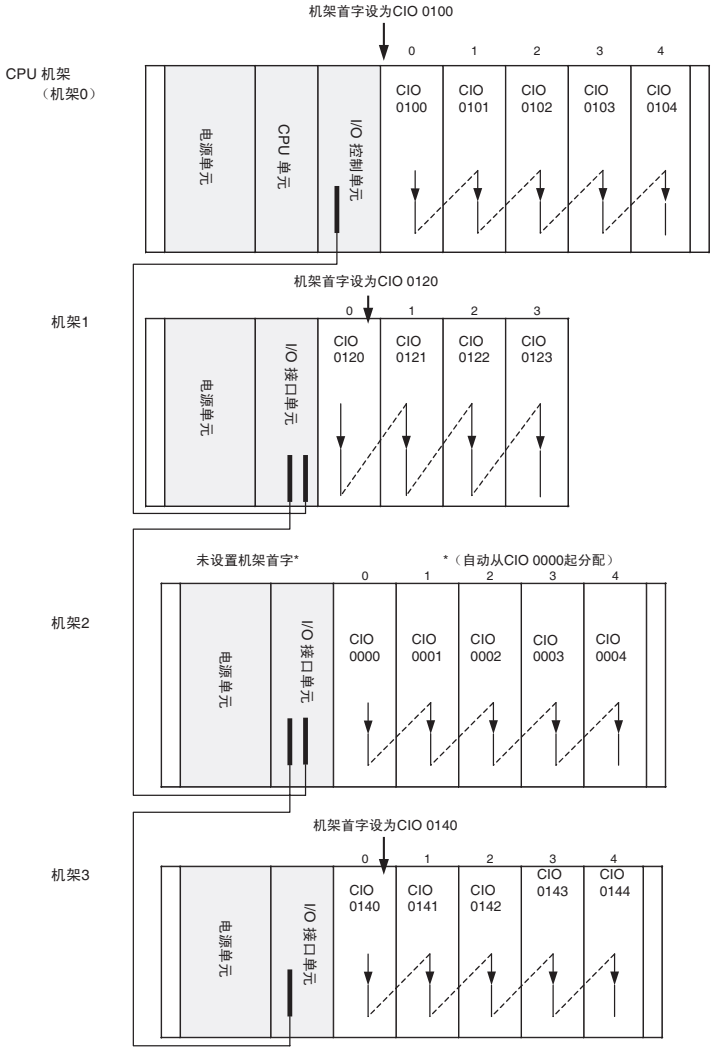
在 CJ 系列 PLC 中，可以用 CX-Programmer 的 I/O 表编辑操作给每个机架分配首字。例如 , 可以设置 CPU 机架，给它分配从 CIO 0000 开始的字；下一个机架，从 CIO 0100 开始的字；再下一个机架，从 CIO 0200 开始的字等。这种方法的特点是不用计算 CPU 机架的路线，就能方便地检查单元的字分配。

● 字分配

对于已经设置首字地址的机架，字从指定的首字开始，按照单元安装的顺序进行分配 （从左到右），空槽位不分配字。

对于没有设置首字地址的机架，从分配给以前的机架的最后一个字开始，字按照机架号的顺序（从最低到最高）连续分配，对于没有设置首字的机架，从第一个机架的 CIO 0000 开始分配。

示例：给机架设置首字



机架首字设定

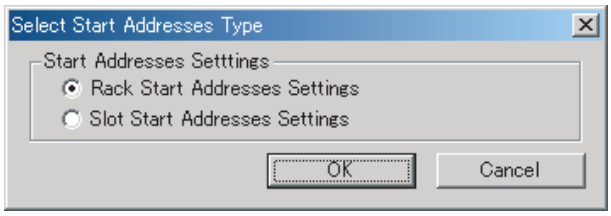
机架	首字
CPU 机架	CIO 0100
机架 1	CIO 0120
机架 2	0000
机架 3	0140

注　机架编号（0 ～ 3）是根据机架以电缆物理连接的顺序而固定的。CPU 机架始终为机架 0，其它机架为机架 1 ～ 3。这些编号不可更改。

从 CX-Programmer 设置机架首字

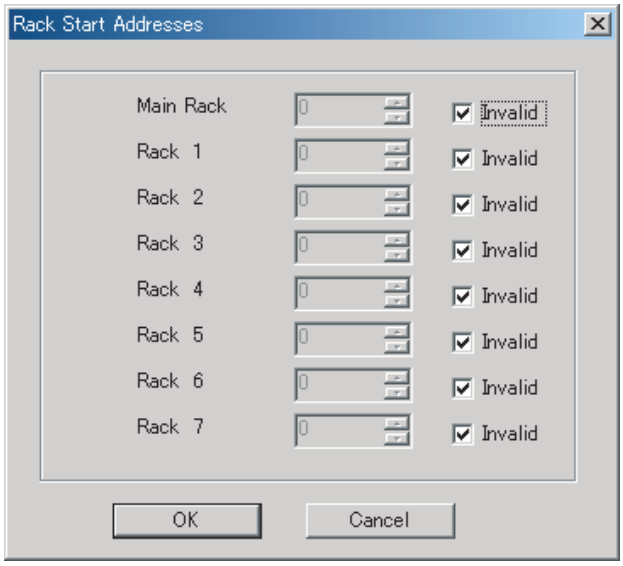
使用以下步骤设置机架首字。

1. 从 I/O 表窗口上的 Option 菜单中选择“机架 / 槽位起始地址”（Rack/Slot Start Addresses）将显示下面的对话框。



2. 选择 “机架起始地址设定”（Rack Start Addresses Settings）选项并点击 “OK”按钮。

3. 在出现的对话框中，去掉不能用机架首字设定的检查标记，然后设定 CPU 机架和扩展机架 （1 ～ 7）的首字地址。



设定	设定范围	缺省	备注
机架起始地址	0 ～ 900	0	所有机架都相同
无效	选择或清除	选择 （无效）	

注　1　任何 CPU 单元型号能设置最多 3 个机架。

2　尽管 CX-Programmer 窗口将显示 7 个机架 , 对于 CJ2 CPU 单元只能设置 3 个机架。

正确使用注意事项

- 保证进行首字设置时分配的字不重叠。机架的首字设置可以是从 CIO 0000 到 CIO 0900 的任何地址。如果两个机架被分配了相同的字，或者两个机架的首字为 CIO 0901 或更高，就不能创建 I/O 表，并且扩展机架编号重复标志 （A409.00 ～ A409.03：机架编号 0 ～ 3）将转为 ON。重复错误标志 （致命错误）（A401.13）也将转为 ON。
- 设置机架的首字分配后必须注册I/O表。要注册I/O表，需在CX-Programmer的I/O表窗口选择“选项”（Options）然后选择 “创建”（Create）。 I/O 表注册操作注册分配给机架的 I/O 字。
- 如果注册 I/O 表后，实际的系统配置被改变，使得字数或单元类型与 I/O 表不匹配，将发生 I/O 检验错误（A402.09）或 I/O 设定错误（A401.10）。也可能发生 CPU 总线单元设定错误（A402.03）或特殊 I/O 单元设定错误 （A402.02）。拆卸单元时 , 可以用 I/O 表修改操作为丢失的单元预留字。如果改变或增加了单元, 程序中这个单元的分配字后的所有字将被改变, 并必须再次执行I/O表注册操作。
- 如果使用 CX-Programmer 删除了 I/O 表，机架的首字设定也将被删除。

8-1-4 I/O 表错误和注意事项

I/O 表错误

● I/O 设定错误和编号重复错误

当执行创建、编辑或传送 I/O 表（使用用户自定义 I/O 分配）的步骤时，不能在以下情况中创建 I/O 表。

问题	错误	描述	改正
连接的单元和 I/O 表之间的检验错误	I/O 设定错误（致命错误）	出现检验错误， I/O 设定错误标志（致命错误）（A401.10）将转为 ON。也就是说，注册的 I/O 表与在基本系统 （CPU 机架和扩展机架）中实际连接的单元的型号和位置不匹配。 不能进行操作。 CPU 单元前部的 RUN 指示灯将变为 OFF， ERR/ALARM 指示灯将闪烁红。	• 如果单元编号不正确，关闭电源并且正确连接适当的单元。 • 如果单元编号正确，通过 CX-Programmer 比较 I/O 表来确认单元的差异，关闭电源，然后改正单元连接。 • 如果在 I/O 表中有一个错误，使用 CX-Programmer 重新建立或编辑 I/O 表来改正错误。
对特殊 I/O 单元或 CPU 总线单元设定了两次相同的单元编号。	重复错误 （致命错误）	在以下任何情况中，重复错误标志（A401.13）将转为 ON。 • 对一个以上的 CPU 总线单元设置了相同的单元编号。 • 对一个以上的特殊 I/O 单元设置了相同的单元编号。 • 对一个以上的基本 I/O 单元分配了相同的字。 • 对一个以上的扩展机架设置了相同的机架编号。 不能进行操作。 CPU 单元前部的 RUN 指示灯将变为 OFF， ERR/ALARM 指示灯将闪烁红。	检查 CPU 总线单元或特殊 I/O 单元的单元编号，消除重复的编号，然后关闭机架电源，并再次接通。 检查在 A409.00 ～ A409.03 中位为 ON 的机架编号上单元的分配。改正分配，使之不重复，包括其他机架。关闭机架电源，并再次接通。 检查在 A409.00 ～ A409.03 中表示的机架中的首字的设置，并用 CX-Programmer 将设置修改为 CIO 0900 以下的有效字地址。

此外，当 I/O 点数过多错误标志（致命错误）（A401.11）或 I/O 总线错误标志（致命错误）（A401.14）为 ON 时，不能使用 I/O 表。

● I/O 表创建错误的详细信息

从 CX-Programmer 创建 I/O 表时，每当错误发生时 A261（I/O 表创建错误详细信息）的内容会给出引起错误的单元的信息。这个信息会使查找引起问题的单元和 I/O 表故障检修变得更容易。

名称		地址		内容
		字	位	
I/O 表创建错误的详细信息	CPU 总线单元设置区初始化错误标志	A261	00	ON：CPU 总线单元设置中的错误。 正常生成 I/O 表时变为 OFF。
	I/O 溢出标志		02	ON：超过最大 I/O 点数。 正常生成 I/O 表时变为 OFF。
	重复错误标志		03	ON：同一单元编号被多次使用。 正常生成 I/O 表时变为 OFF。
	I/O 总线错误标志		04	ON：I/O 总线错误。 正常生成 I/O 表时变为 OFF。
	特殊 I/O 单元错误标志		07	ON：特殊 I/O 单元中的错误。 正常生成 I/O 表时变为 OFF。
	I/O 未确认错误标志		09	ON：未完成 I/O 检测。 正常生成 I/O 表时变为 OFF。

设定 I/O 表的注意事项

设定 I/O 表后，如果单元发生改变，检查字分配中是否有重复。可以想象，I/O 表注册后可能发生字分配的重复。例如，将 1 字单元更换为 2 字单元所导致的结果。这种情况下新单元所需的额外字仍将分配到下一个单元。

当 PLC 转为 ON 时，CPU 单元根据实际连接到 PLC 的单元来检查已注册的 I/O 表。若存在重复，将发生错误且不可再编辑 I/O 表。若发生这种情况，必须编辑 I/O 表并重新传送。

当 I/O 表被编辑时，CX-Programmer 检查在首字设定中是否存在重复和设定范围错误，以用于帮助消除问题。

使用存储卡时的注意事项

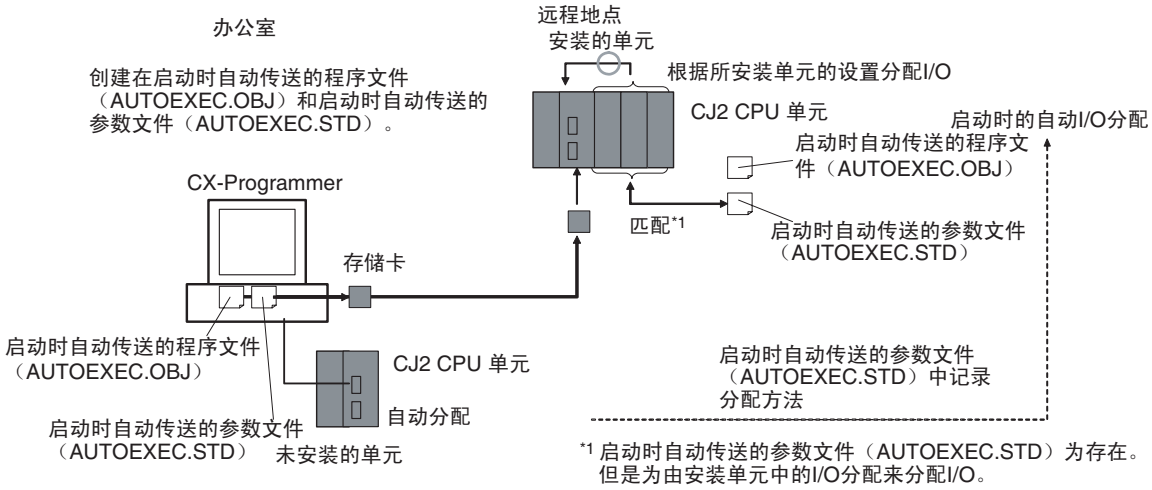
用于创建 CPU 单元 I/O 表的 I/O 分配方法 （启动时自动 I/O 分配或用户自定义 I/O 分配）被记录在启动时自动传送的参数文件中 （AUTOEXEC.STD）。当存储卡在启动时执行自动传送时，自动检测记录的方法并用于在启动时自动分配 I/O 或检验 I/O 表。

以下描述解释了通过在电源启动时创建用于自动传送的参数文件 （AUTOEXEC.STD），在 CPU 单元中创建 I/O 表的两种不同方法。

- 自动分配 *1
  存储卡的参数文件中用于在启动时自动传送的 I/O 表被禁用，并且根据在 PLC 中实际连接的单元，在启动时使用自动 I/O 分配来分配 I/O。

*1 对于启动时的自动 I/O 分配，不提前创建 I/O 表，并且每次电源被转为 ON 的时候，实际安装的基本 I/O 单元的 I/O 分配被自动进行。

- 手动分配
  存储卡的参数文件中用于在启动时自动传送的 I/O 表被被启用，CPU 单元检验没有在 PLC 中实际连接单元的 I/O 表。



8-2 设定 CPU 总线单元和特殊 I/O 单元

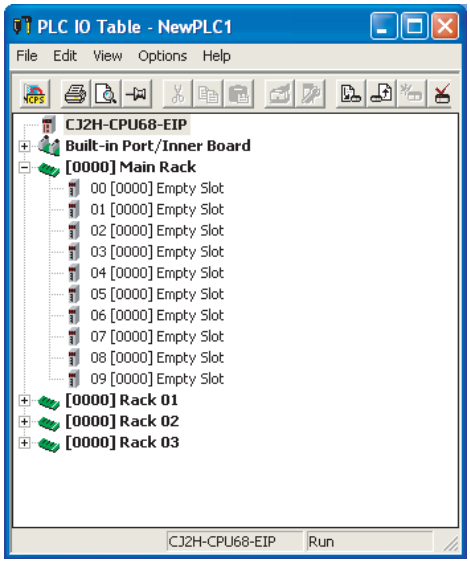
8-2-1 设定参数

CX-Programmer 可以用于在分配到 CJ 系列 CPU 总线单元和特殊 I/O 单元的 DM 区和 CIO 区字中设定参数，并设定 EtherNet 单元的 CPU 总线单元设置。 I/O 表窗口在 CX-Programmer 中使用。

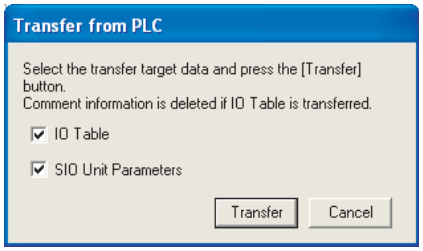
在线或离线都可进行设定。如果进行离线设定，必须在线传送它们。

本节描述了在创建 I/O 表并将其从 CPU 单元传送到 I/O 表窗口之后，编辑 I/O 表以及设定 CPU 总线单元和特殊 I/O 单元的示例。步骤与离线注册 CPU 总线单元和特殊 I/O 单元到 I/O 表，然后编辑 CPU 总线单元和特殊 I/O 单元的设定相同。

1. 在线连接 CX-Programmer 并打开 I/O 表窗口。

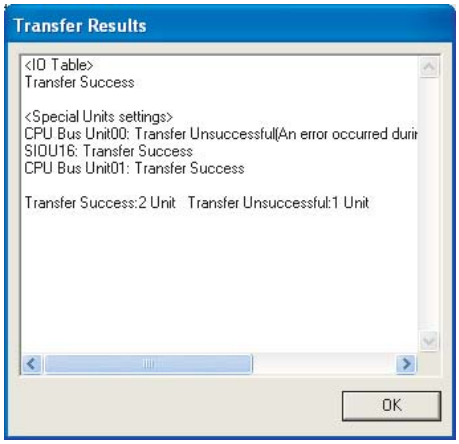


2. 在 I/O 表窗口菜单中选择“选项”（Options）-“从 PLC 传送”（Transfer from the PLC）。将显示下图所示的从 PLC 传送对话框。

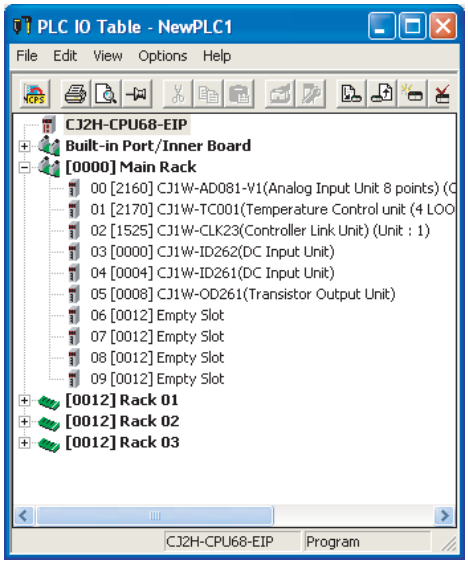


此处，可选择 “I/O 表”（I/O Table）和 “SIO 单元参数”（SIO Unit Parameters）选项。

3. 选择你想要传送的数据的选项（检查框），并点击传送按钮。在这种情况下，两个选项都被选择。I/O 表和单元参数数据将从 PLC 传送，并显示传送结果。

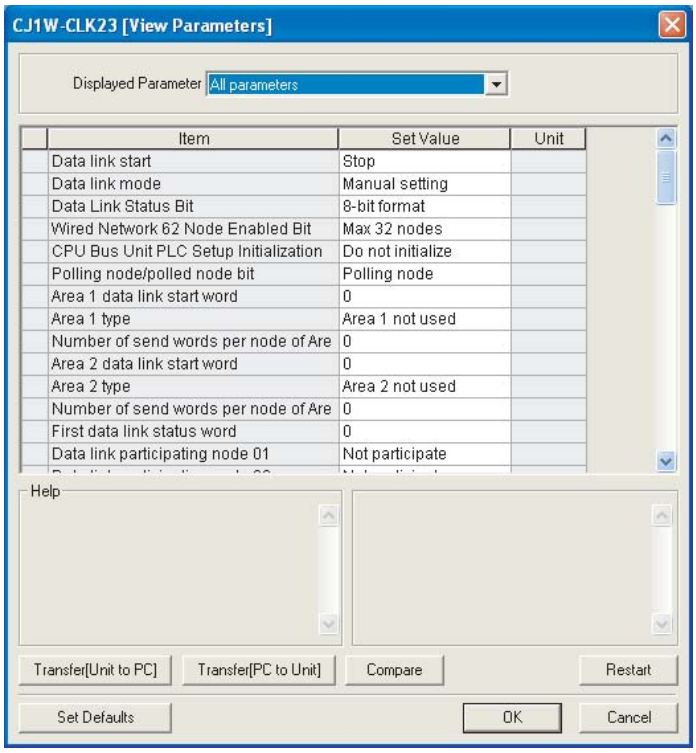


4. 检查传送结果并点击 “OK” 键，将显示从 PLC 传送的 I/O 表。

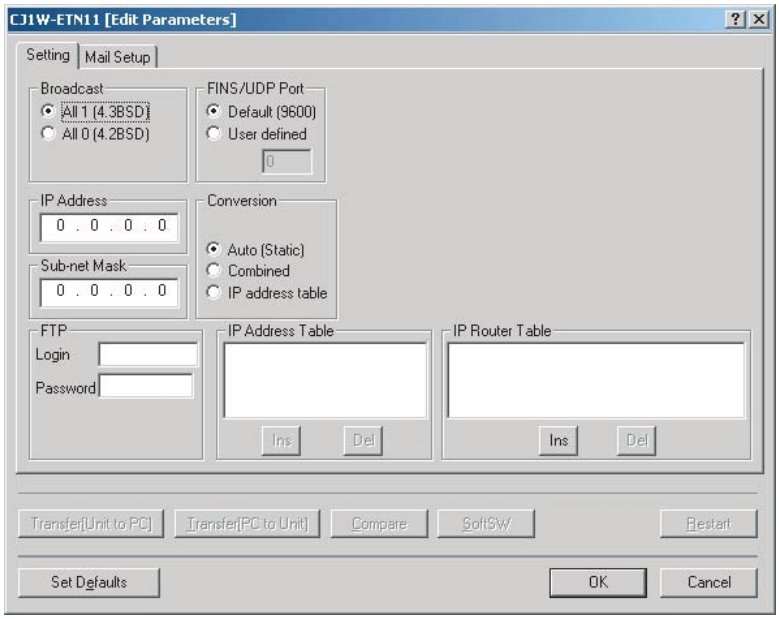


5. 要编辑特殊 I/O 单元或 CPU 总线单元的参数，选择单元并双击单元或从菜单中选择“编辑”（Edit）-“SIO 单元参数”（SIO Unit Parameters）。将显示所选单元的编辑参数对话框。

- 在此案例中，将显示 Controller Link 单元的编辑参数对话框，用于编辑单元的 CPU 总线单元分配 DM 设定。

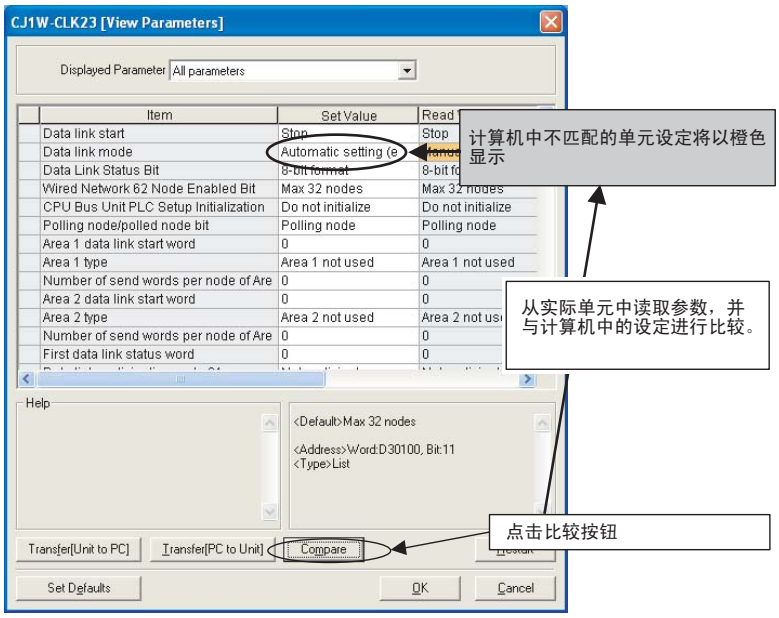


- 在此案例中，将显示EtherNet单元的编辑参数对话框，用于编辑在DM区中分配的字的CPU总线单元设定。



6. 编辑参数，如果需要的话点击 “比较”（Compare）按钮。如果在 PLC 在线时点击了 “比较”按钮，软件会立即检查 CPU 单元中的 I/O 表设定是否匹配单元设定（例如，在 DM 区和 CIO 区中分配的字）。如果内容不匹配，从实际单元中读取的设定将显示为橙色。

以下示例显示了 Controller Link 单元的比较设定的结果。



7. 如果想要下载每个单元的参数设定到 CPU 单元，点击 “下载到单元”（Download to Unit）按钮。

正确使用注意事项

在 I/O 表窗口中编辑特殊 I/O 单元或 CPU 总线单元设定，在编辑参数对话框中点击了 “传送 PC 到单元”（Transfer PC to Unit）按钮时，在单元的 DM 区或 CPU 总线单元设置区分配的参数只能传送到实际的PLC。它们不会被自动启动，除非你启用了它们否则不能使用。点击“复位”（Reset）按钮来启用已经传送的参数设定。

附加信息

保存参数设定数据文件

- 在编辑参数对话框中设定的数据可以保存为以 .xml 为文件扩展名的参数设定文件。要保存该参数设定，在 I/O 表中右键所需的特殊 I/O 单元或 CPU 总线单元，并在弹出菜单中选择 “保存参数”（Save Parameters）。
- 也可以从参数设定文件（以 .xml 为扩展名）读取保存的参数设定，但必须是用于相同型号的特殊 I/O 单元或 CPU 总线单元的设定。

8-2-2 数据交换

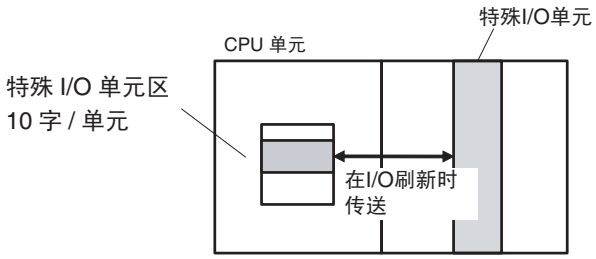
本节描述了数据是如何在特殊 I/O 单元或 CPU 总线单元和 CPU 单元之间进行交换的。

特殊 I/O 单元

特殊 I/O 单元区 （I/O 刷新）

在特殊 I/O 单元区的 I/O 刷新时数据每个循环交换。基本上，根据特殊 I/O 单元的单元编号设定给每个特殊 I/O 单元分配 10 个字。详细信息请参考各特殊 I/O 单元的操作手册。

特殊 I/O 单元区范围从 CIO 2000 ～ CIO 2959 （10 字× 96 单元）。



分配到 DM 区的字传送

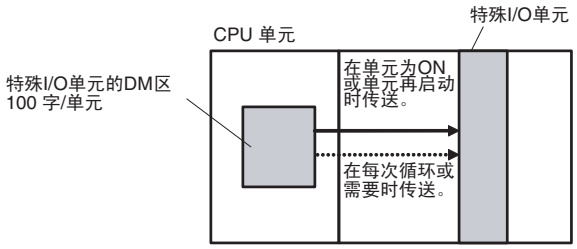
分配给每个单元的字有三次可传送数据，数据传送的时序取决于使用的型号。

1. 在 PLC 变为 ON 时传送数据。
2. 在再启动单元时传送数据。
3. 在需要时传送数据。

有些型号可双向传送数据，从 DM 区向单元和从单元向 DM 区。关于数据传送的详细信息，请参考单元的操作手册。

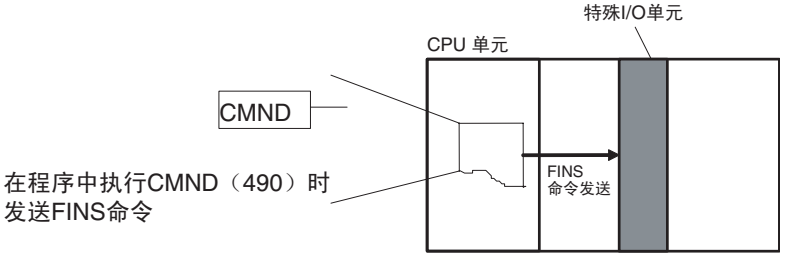
DM 区中的特殊 I/O 单元字：D20000 ～ D29599 （100 字× 96 单元）

数据传送到这些字用于特殊 I/O 单元的初始设定。从程序变更分配字的内容以改变系统时，特殊 I/O 单元再启动位 （A502.00 ～ A507.15）必须转为 ON 来再启动单元。

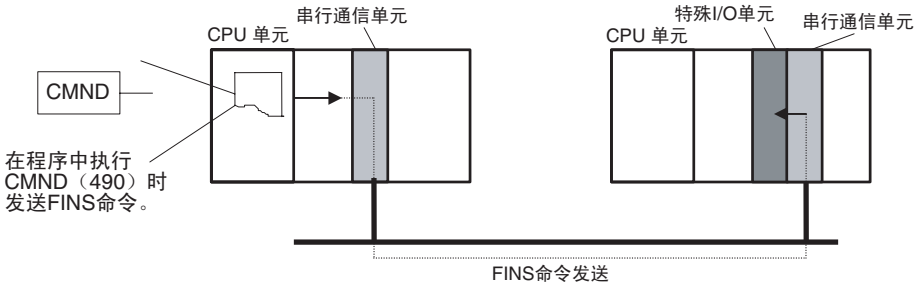


● FINS 命令

CMND （490）指令可以添加到梯形图程序，用于发送 FINS 命令给特殊 I/O 单元。



注　可将 FINS 命令发送给网络中其它 PLC 的特殊 I/O 单元，而不仅是本地 PLC。



● 特殊 I/O 单元初始化

特殊 I/O 单元在 PLC 电源接通或单元的再启动位 （A502.00 ～ A507.15）变为 ON 时被初始化。在单元初始化的同时，单元的特殊 I/O 单元初始化标志 （A330.00 ～ A335.15）将变为 ON。

在特殊 I/O 单元的初始化标志为 ON 的同时，不会对其进行 I/O 刷新（循环 I/O 刷新或由 IORF（097）/FIORF （225）刷新）。

● 禁止特殊 I/O 单元循环刷新

根据设置在每个单元前面的单元编号给每个特殊I/O 单元分配特殊I/O 单元区（CIO 2000～CIO 2959）中的十个字，在 CPU 单元内每个循环作 I/O 刷新时 （即在 END （001）指令执行后），特殊 I/O 单元区中的数据被刷新。

如果安装过多的特殊 I/O 单元，则 I/O 刷新可能需要很长时间。如果 I/O 刷新使用过多时间，可以将特定的特殊 I/O 单元的 PLC 设置设为禁止循环刷新 （特殊 I/O 单元循环刷新禁止位在 PLC 设置中的地址是 226 ～ 231）。

如果 I/O 刷新时间过短，则单元的内部处理将不能跟上，特殊 I/O 单元错误标志 （A402.06）将变为 ON，且特殊 I/O 单元可能会操作不正确。在这种情况下，在 PLC 设置中设定一最小循环时间就可延长循环时间，或可禁止特殊 I/O 单元的循环 I/O 刷新。

在循环刷新已被禁止时，使用 IORF（097）或 FIORF（225）在程序执行时可刷新特殊 I/O 单元的数据。

正确使用注意事项

在特殊 I/O 单元的中断任务中可执行 IORF （097）、 FIORF （225）、 IORD （222）和 IOWR （223）。这样做时，必须在 PLC 设置中禁止特殊 I/O 单元的循环刷新。如果没有禁止循环刷新，在中断任务中执行以下任何一个处理，都将发生致命错误并且重复刷新错误标志（A402.13）转为 ON。

- 使用 IORF （097）或 FIORF （225）对相同的特殊 I/O 单元执行了 I/O 刷新。
- 使用 IORD （222）或 IOWR （223）对相同的特殊 I/O 单元从存储器区读取或写入数据。

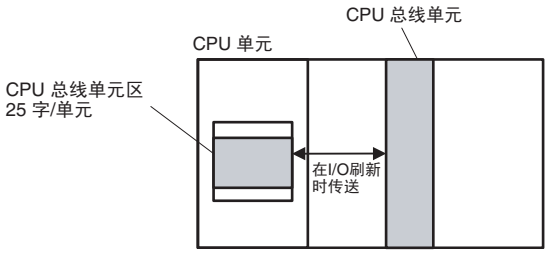
不论何时禁止特殊 I/O 单元的循环刷新，务必在操作时使程序中至少每 11 秒用 IORF （097）或 FIORF（225）刷新该单元的 I/O。如果没有每 11 秒进行刷新，则会在特殊 I/O 单元内发生 CPU 单元服务监控错误。

CPU 总线单元

● CPU 总线单元区 （I/O 刷新）

在 CPU 总线单元区的 I/O 刷新时每循环交换数据。基本上，根据 CPU 总线单元的单元编号设定给每个 CPU 总线单元分配 25 个字。 CPU 总线单元实际用的字数各异。

特殊 I/O 单元区的范围从 CIO 1500 ～ CIO 1899 （25 字× 16 单元）。



注　为刷新分配给指定单元编号的 CPU 总线单元的 CIO 区字，在梯形图程序中可执行 CPU 总线 I/O 刷新指令（DLNK （226））。

● 分配到 DM 区的字传送

根据单元编号给每个单元分配 100 字。

注　这些字不能用于 CPU 总线单元的所有型号。

CPU 总线单元的 DM 区字：D30000 ～ D31599 （100 字× 16 单元）

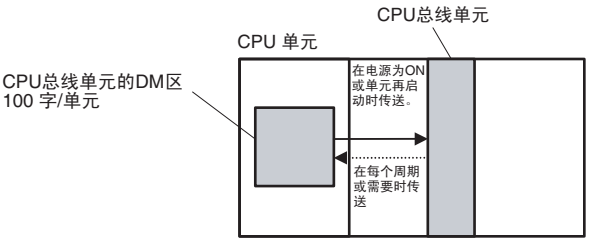
在以下时间传送这些字的内容。实际传送的数据取决于单元的型号。

- 在 PLC 变为 ON 时传送数据。
- 每个循环传送数据。
- 在需要时传送数据。

注　为刷新分配给指定单元编号的 CPU 总线单元的 DM 区字，在梯形图程序中可执行 CPU 总线 I/O 刷新指令（DLNK （226））。

有些型号双向传送数据，从 DM 区向单元和从单元向 DM 区。关于数据传送的详细信息，请参考单元的操作手册。

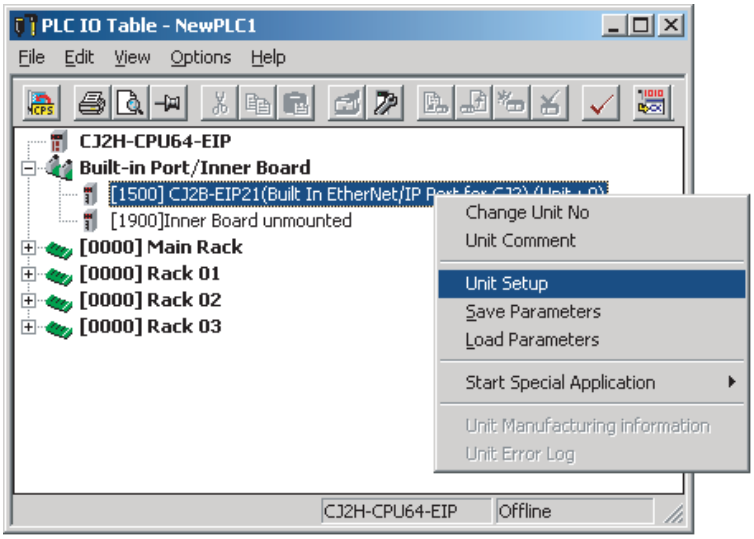
这 100 个字通常是用来保持 CPU 总线单元的初始设定。在从程序改变这个区中的内容以反映系统中的改变时，必须将受影响的单元的再启动位 （A501.00 ～ A501.15）转为 ON，从而再启动单元。



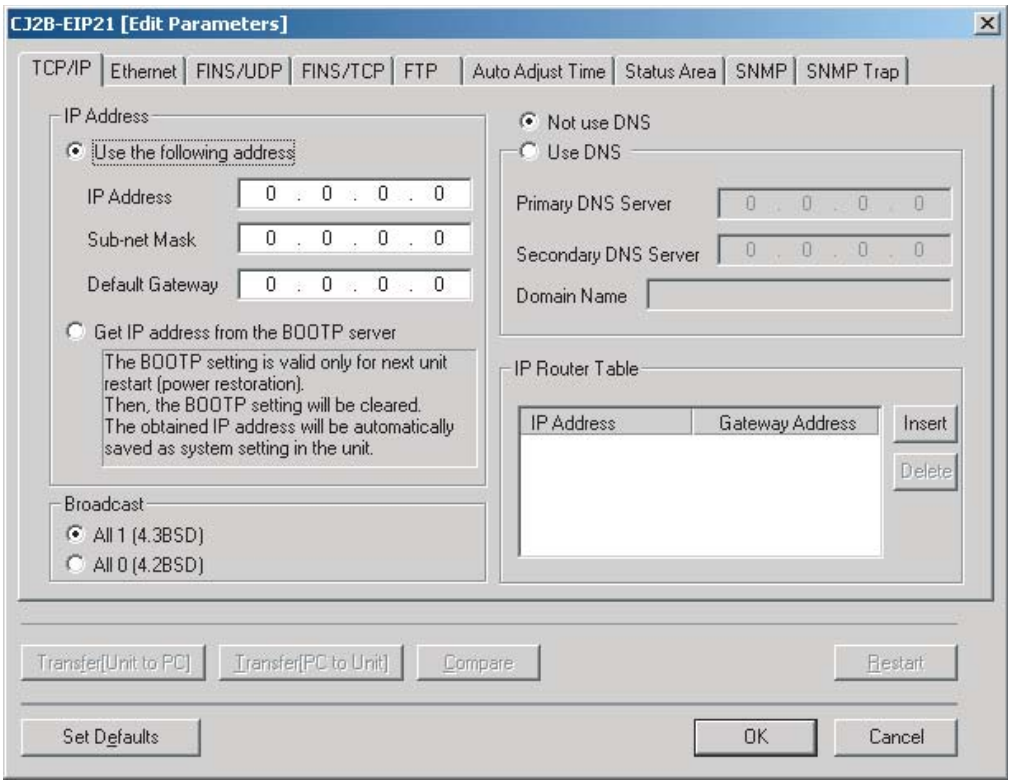
附加信息

根据旋转开关设定，内置 EtherNet/IP 端口可视作 CPU 总线单元。进行以下步骤来设定通信参数。

(1) 在 PLC 的 I/O 表的内置端口 / 内插板下选择 CJ2B-EIP21 （CJ2 的内置 EtherNet/IP 端口）。右键并选择 “单元设置”（Unit Setup）。



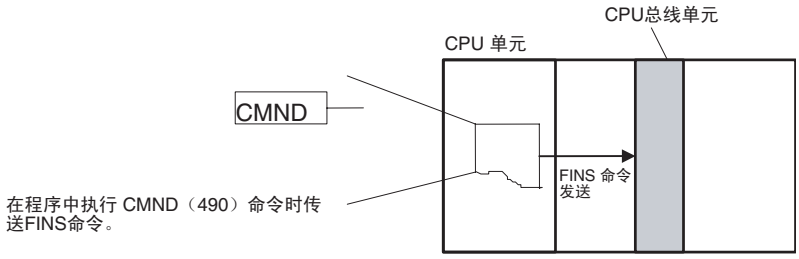
(2) 设定参数。



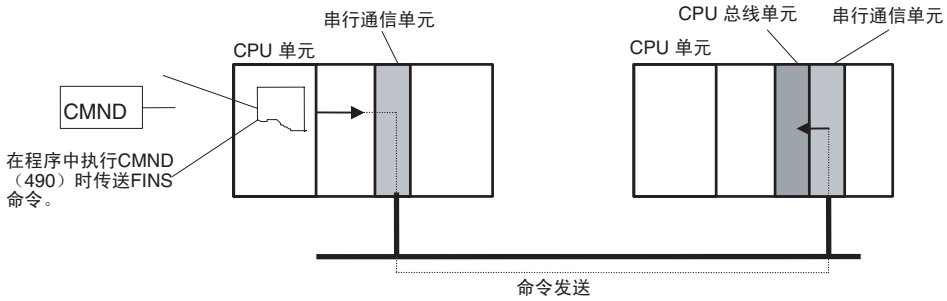
关于内置EtherNet/IP端口的参数的信息，请参考EtherNet/IP单元操作手册（样本编号W465）。

● FINS 命令

CMND （490）指令可以添加到梯形图程序中，以发送 FINS 命令给 CPU 总线单元。



注　可将 FINS 命令发送给网络中其它 PLC 的 CPU 总线单元，而不仅是本地 PLC 。



● CPU 总线单元初始化

CPU 总线单元在 PLC 电源接通或单元的再启动位（A501.00 ～ A501.15）变为 ON 时被初始化，在单元初始化的同时，单元的 CPU 总线单元初始化标志 （A302.00 ～ A302.15）会变为 ON。
在 CPU 总线单元的初始化标志为 ON 时，不会对其进行循环 I/O 刷新。

9

PLC 设置

本章节中描述了用于设定 PLC 操作的 PLC 设置中相关参数的详细内容。

9-1 PLC 设置概述 9-2

9-2 PLC 设置设定 9-4

9-2-1 启动操作设定 9-4

9-2-2 CPU 单元设定 9-7

9-2-3 时间和中断设定 9-13

9-2-4 特殊 I/O 单元循环刷新 9-18

9-2-5 基本 I/O 单元机架响应时间 9-19

9-2-6 串行端口设定 9-21

9-2-7 外设服务 9-26

9-2-8 FINS 保护 9-27

9-1 PLC 设置概述

PLC 设置含有基本 CPU 单元软件参数设定，用户可以通过变更设定，以自定义 PLC 操作，这些设定可以从手持编程器或其它编程设备来改变。可如下表所示，进行 PLC 设置的变更。

应用	参数
在下列情况下必须改变 CJ 系列基本 I/O 单元的输入响应时间设定： • 在基本 I/O 单元内发生振颤或干扰。 • 接收间隔大于循环时间的短脉冲。	基本 I/O 单元机架响应时间
在 PLC 的电源接通时，必须保留 I/O 存储器所有区域的数据（包括 CIO 区，工作区，定时器标志和 PV，任务标志，变址寄存器和数据寄存器）。	启动时 IOM 保持位状态
在 PLC 的电源接通时，必须保留从编程设备 （包括手持编程器）强制置位或强制复位的位的状态。	启动时强制状态保持位状态
• 调试 PLC， • 变更启动模式为 PROGRAM 或 MONITOR 模式。	启动模式
使用无电池操作时不需要电池电压过低错误检测。	检测电池电压过低
使重复刷新的检测无效。	检测重复刷新错误
RS-232C 端口不用于手持编程器或 CX-Programmer （外设总线）通信速度自动检测，且不使用默认的上位机链接通信设定，如 9,600 bps。 *1	RS-232C 端口设定
要通过 NT 链接加速与 PT 的通信。	将外设端口或 RS-232C 端口通信端口波特率设置在 “高速 NT 链接”。
要将定时中断的间隔设置在单位为 1 ms （或 0.1ms）而不是 10 ms。	定时中断时间单位
调试时查找指令错误。	指令错误下的 CPU 停止
通过最小循环时间设定，以创建一固定的 I/O 刷新循环。	最小循环时间
要设置一个除 1 秒以外的最大循环时间 （10 ms ～ 40,000 ms）。	监视循环时间
电源中断时进行特殊处理。	电源 OFF 中断任务 *2
要延迟电源中断的检测。	电源 OFF 检测延迟时间 *1
• 要执行中断任务中的 IORF 或 FIORF 。 • 使用许多特殊 I/O 单元时，要缩短平均循环时间。 • 要延长特殊 I/O 单元的 I/O 刷新间隔。	特殊 I/O 单元循环刷新

应用	参数
不在错误记录中记录 FAL （006）和 FPD （269）的用户定义的错误	FAL 错误记录登记
要降低由文本串处理引起的循环时间的波动。	表数据，文本串和数据移位指令的后台执行
• 电源 ON 时，部分单元需要等候片刻后启动。 • 启动 CPU 单元操作，而不要等待各单元完成启动处理	执行设定

*1 要变更设定 PLC 设置时， CPU 单元前面的 DIP 开关的位 5 必须为 OFF。

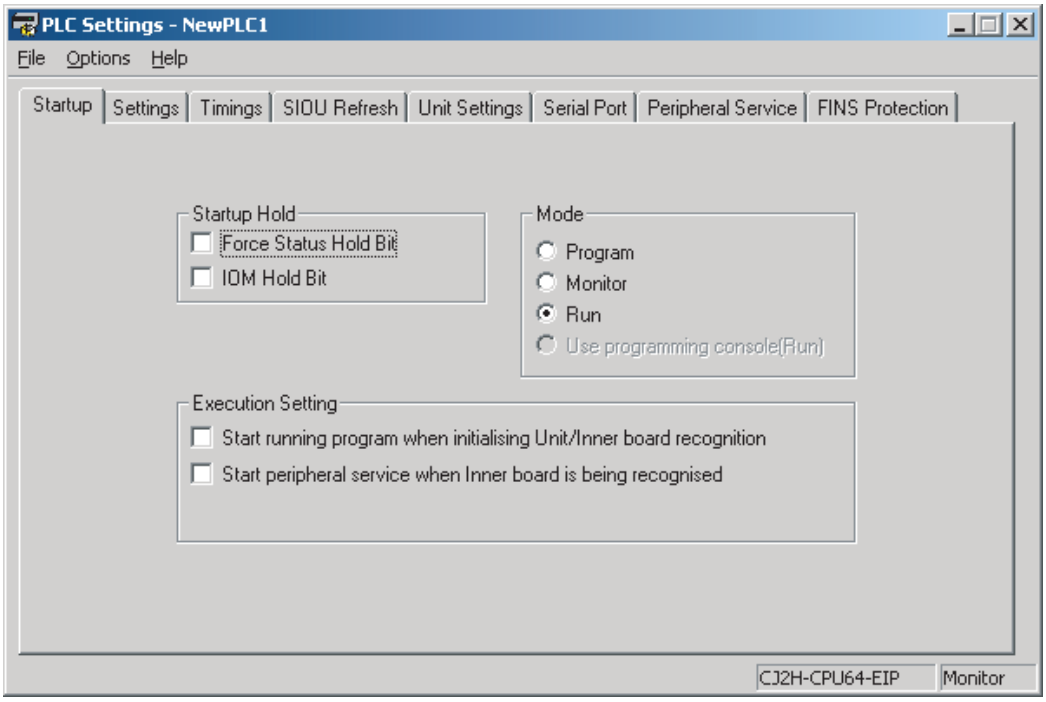
*2 安装了 CJ1W-PD022 电源单元时不支持此些设定。

● 辅助区中的相关字 / 位

名称	地址	描述	访问
PLC 设置错误标志（非致命错误）	A40210	当 PLC 设置中发生设定错误时 ON。	只读

9-2 PLC 设置设定

9-2-1 启动操作设定



启动保持设定

● **强制状态保持位启动保持设定**

通过此参数，设定在启动时是否保留辅助区中强制状态保持位 （A500.13）的状态。

参数	设定	默认	功能	相关标志和字
强制状态保持位启动保持设定	OFF：清除 ON：保留	OFF	设定在启动时是否保留辅助区中强制状态保持位 （A500.13）的状态。	A500.13 （强制状态保持位）

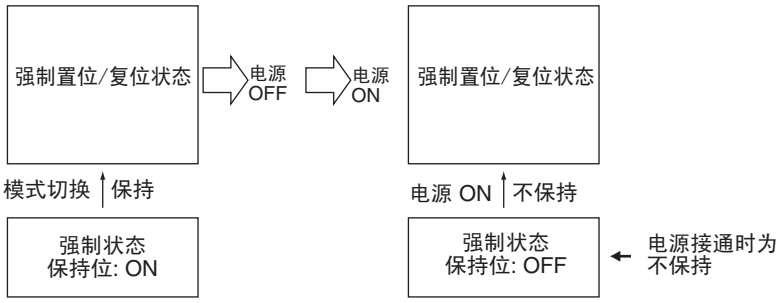
- 启动时强制状态保持位与本设定的关系

可将强制状态保持位（A500.13）设为 ON，则当 CPU 单元运行模式在 PROGRAM 模式 MONITOR 模式间切换时，将保持所有被强制置位 / 复位的位的强制状态。当 PLC 打开时，除非是使用了本 PLC 设置参数设定进行保护，否则强制状态保持位将为 OFF。

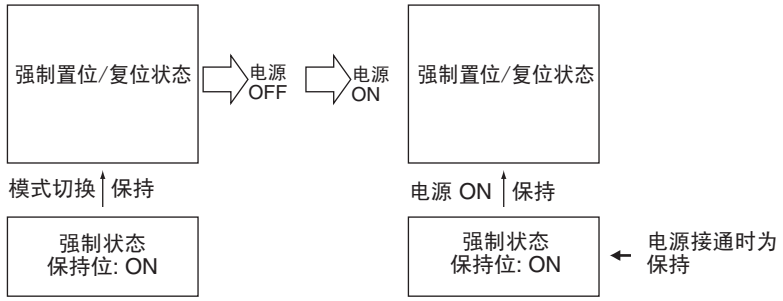
如果强制状态保持位启动保持参数设定为 ON，则当 PLC 打开时强制状态保持位的状态将受保护。如果此参数设定为 ON 且强制状态保持位也为 ON，则当 PLC 打开时所有的强制置位 / 复位都将保持其强制状态。

如果因电池电压过低导致存储器无法保持，则无论此参数设定为 ON 或 OFF，强制保持位都将清除。

OFF: 启动时清除强制状态保持位



ON: 启动时保护强制状态保持位



● IOM 保持位的启动保持设定

通过此参数设置是否在启动时保持辅助区中的 IOM 保持位状态。

参数	设定	默认	功能	相关标志和字
IOM 保持位启动保持设定	OFF：清除 ON：保持	OFF	此设置决定是否在启动时保持 IOM 保持位（A500.12）的状态。 需要在电源接通时保持 I/O 存储器的所有数据时，要设定 IOM 保持位为 ON 并将此设定置 ON。	A500.12（IOM 保持位）

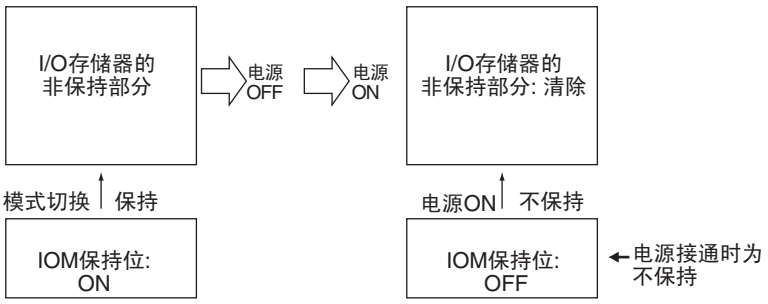
- IOM 保持位与此参数的关系

  当 CPU 单元运行模式从 PROGRAM 模式转为 RUN/MONITOR 模式或反之时，可将 IOM 保持位（A500.12）设置为 ON 以保持 I/O 存储器中的所有数据。当 PLC 转为 ON 时，若 PLC 设置中的设定在非保护的状态下时 IOM 保持位将自动转为 OFF。

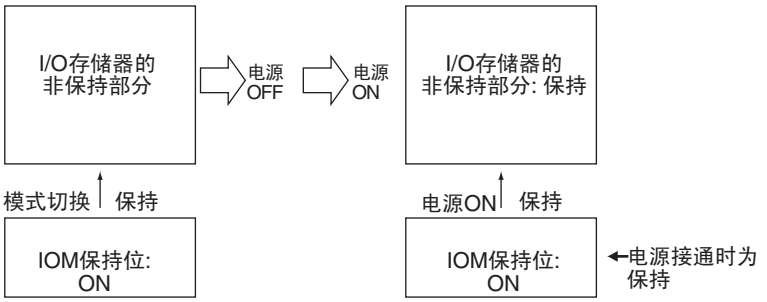
  如果启动时的 IOM 保持位状态设置为 ON，则在 PLC 转为 ON 时 IOM 保持位的状态将受到保护。若此参数设置为 ON 且 IOM 保持为也为 ON，则在 PLC 转为 ON 时 I/O 存储器中的所有数据都将为保持。

  如果因电池电压过低导致存储器无法保持，则无论此参数为 ON/OFF， IOM 保持位都将被清除。

OFF:启动时清除IOM保持位



ON: 启动时保护IOM保持位



模式

● PROGRAM、MONITOR、RUN 模式

设定启动时使用的模式。

参数	设定	默认	功能	相关标志和字
模式	• 程序：PROGRAM 模式 • 监控：MONITOR 模式 • 运行：RUN 模式	RUN 模式	设定启动时的 CPU 单元运行模式。	---

执行设定

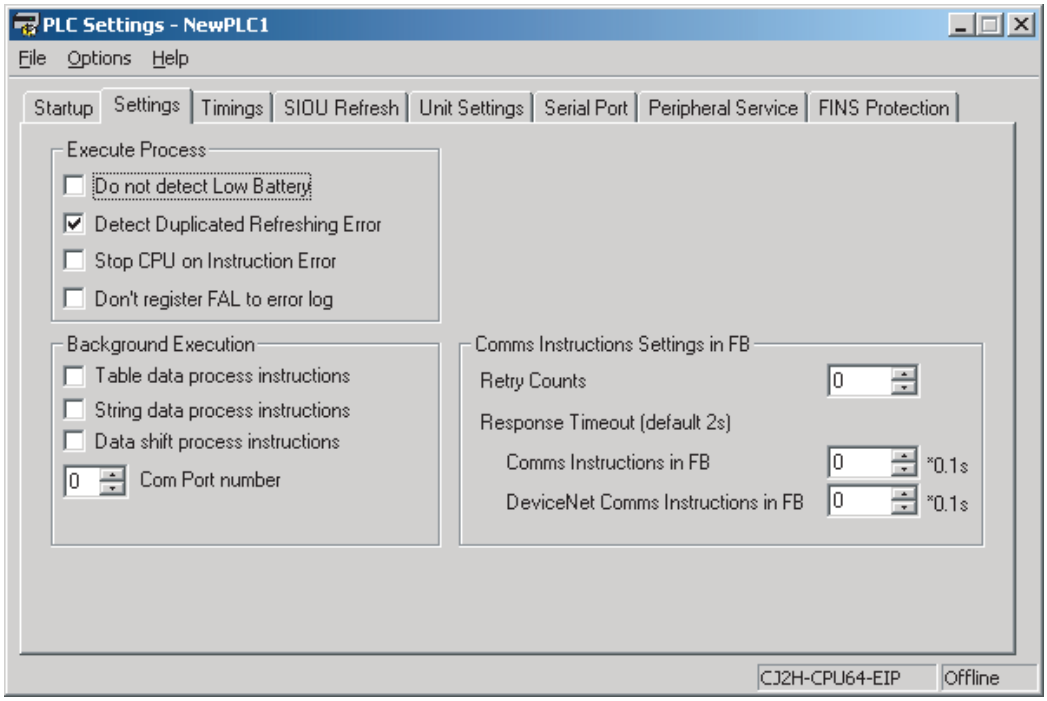
● 初始化单元 / 内插板识别时的启动运行模式

设定在启动时是否等待指定单元的启动运行处理。

参数	设定	默认	功能	相关标志和字
初始化单元 / 内插板识别时的启动运行程序	OFF：等待 ON：不等待	OFF	在 RUN/MONITOR 模式下开始运行 CPU 单元时，若出现一个或多个单元 *1 为未完成启动运行处理时，可将此设定设置为 ON （不等待）。 若要等待所有单元完成启动运行，则将此设定设置为 OFF （等待）。	---

*1 此设定仅应用于指定单元。若设定为 “不等待”，则 CPU 单元将仅不等待指定的单元，例如，启动时仍将等待其他指定单元。

9-2-2 CPU 单元设定



执行处理

● 不检测电池电压过低 （不使用电池）

设定是否检查电池错误 （默认：检测）。不使用电池时请设定为不检测。详细相关内容，请参见 CJ2 CPU 单元硬件操作手册 （Cat. No. W472）。

参数	设定	默认	功能	相关标志和字
不检测电池低电压	OFF：检测 ON：不检测	OFF	设定是否检查 CPU 单元电池错误。若设定为 OFF （检测）并检测到电池错误，则 CPU 单元上的 ERR/ALM 指示灯为闪烁且电池错误标志 （A402.04）将置 ON，但 CPU 单元运行将继续。	A402.04 （电池错误标志）

附加信息

无电池操作

- **不使用电池**

不使用电池进行 PLC 操作时，下述数据将被保持。

- 用户程序和网络符号 （变量）。
- 参数 （如， PLC 设置，注册 I/O 表，路由表， CPU 总线单元设定， PLC 名称）
- 保存在 CPU 总线单元和特殊 I/O 单元的非易失性存储器中的数据。（如，串行通信单元中的协议宏数据）

下述数据将不保持，且值会不稳定。

- I/O 存储器 （包括保持区、 DM、 EM 和辅助区）
- CPU 单元内置的时钟。

- **不使用电池操作设定**
  - PLC 设置

    PLC 设置如下所述。

    （1）不使用电池时 I/O 存储器将不稳定，因此请将 IOM 保持位设定为不保持，以使在启动时清除 IOM 保持位。

    （2）不使用电池时强制置位 / 复位状态将不稳定，因此请将强制状态保持位设定为不保持，以使在启动时清除强制状态保持位。

    （3）选择不检测电池低电压。
  - 编程

    不使用电池时输出 OFF 位的操作将不稳定，因此请插入下述指令，以使输出 OFF 为不置 ON。

    LD P_Off

    OUT A500.15

- **不使用电池时的操作注意事项**

不使用电池时，请注意下述各点。

- 初始化数据

  DM 区、 EM 区域、和其他 I/O 存储器区中的值将不稳定，因此请确保从程序设定初始值。

  示例：

  LD A200.11

  MOV #0918 D0 // 运行启动时的 D0 初始值。

- **时钟不稳定**

CPU 单元中内置的时钟将不运行且值不稳定。因此，将无法正确显示错误记录中记录的日期与时间数据。此外，若文件保存在存储卡上，则创建文件的日期与时间数据也将不稳定。

正确使用注意事项

CPU 单元中的 DM、EM、HR 和 AR 区内容将不备份到内置闪存中。电源关闭或中断时 DM、EM、HR 和 AR 区内容为通过电池进行保持。若发生电池错误则这些数据都将遗失。可通过在程序中采取应对措施，通过电池错误标志 （A402.04）重新初始化数据。

● 检测重复刷新错误 （设定是否检测重复刷新错误）

设定是否检测重复刷新错误 （默认：检测）。如果此参数设置为默认 （ON：检测），则如下例检测重复刷新错误。

参数	设定	默认	功能	相关标志和字
检测重复刷新错误	OFF：不检测 ON：检测	ON	此参数设定是否检测重复刷新错误。若此设定为 ON （检测）且检测到错误，则 CPU 单元上的 ERR/ALM 指示灯将闪烁且重复刷新错误标志 （A402.13）将置 ON。 CPU 单元继续运行。	A402.13 （重复刷新错误标志）

● 指令错误时停止 CPU

此设定决定是否在发生指令执行错误时停止运行 （默认：不停止）。程序调试时可使用指令错误时停止 CPU 的设定。若下述任一标志置 ON，则将生成程序错误并视作指令错误。如果此时设置了指令错误时停止 CPU，则将停止 CPU 运行。



参数	设定	默认	功能	相关标志和字
指令错误时停止 CPU	OFF：继续 ON：停止	OFF	此设定决定是将指令错误（指令处理错误（ER）和非法访问错误（AER））视作非致命错误或者是致命错误。	A295.08（指令处理错误标志），A295.09（间接 DM/EM BCD 码错误标志，A295.10（非法访问错误标志）（若此设定设置为 OFF，则即使发生指令错误，这些标志也将不置 ON）

● 不记录 FAL 到错误记录

此参数决定是否在发生用户定义的 FAL 错误时记录到错误记录中。

参数	设定	默认	功能	相关标志和字
不记录 FAL 到错误记录	OFF：在错误记录中记录用户定义的 FAL 错误 ON：不在错误记录中记录用户定义的 FAL 错误	OFF	此设定决定是否在发生以 FAL（006）创建的用户定义 FAL 错误时和 FPD（269）的时间监控时，将其记录到错误记录中（A100 ～ A199）。 设置为 ON 时为不记录这些错误。	---

后台执行设定

下述指令对循环时间有很大影响，具体要根据以此指令处理的数据量决定。通过后台执行设定可将指令的处理分割到多个循环，以减少对循环时间的影响。详细内容请参见 10-2-5 后台执行。

● 表数据处理指令

此参数设定是否通过后台处理表数据指令。

参数	设定	默认	功能	相关标志和字
表数据处理指令	OFF：不在后台执行 ON：在后台执行	OFF	设定是否将表数据指令进行跨多个循环时间执行。（如，进行后台处理）	---

● 字符串数据处理指令

此参数设定是否通过后台处理字符串数据指令。

参数	设定	默认	功能	相关标志和字
字符串数据处理指令	OFF：不在后台执行 ON：在后台执行	OFF	设定是否将字符串数据指令进行跨多个循环时间执行。（如，进行后台处理）	---

● 数据移位处理指令

此参数设定是否通过后台处理数据移位指令。

参数	设定	默认	功能	相关标志和字
数据移位处理指令	OFF：不在后台执行 ON：在后台执行	OFF	设定是否将数据移位指令进行跨多个循环时间执行。（如，进行后台处理）	---

● 通信端口编号

此参数设定后台处理中所使用的通信端口编号。

参数	设定	默认	功能	相关标志和字
通信端口编号	0 ～ 7：通信端口 0 ～ 7 （内部逻辑端口）	0 （No. 0）	用于后台执行的通信端口编号 （内部逻辑端口）。	---

FB 中设定的通信指令 （用于 FB 库）

下述参数仅可使用于欧姆龙功能块。

重新发送次数和响应监控时间必须由用户在 PLC 设置中的 FB 通信指令设置中进行设定，特别当使用来自欧姆龙 FB 的功能块来执行 FINS 报文或 DeviceNet 显式报文通信时。在此 PLC 设置中所设的欧姆龙 FB 库的值将被自动保存在相关辅助区字 A580 ～ A582 中并由来自欧姆龙 FB 库的功能块所使用。

● 重试次数

此参数决定在功能块中执行通信指令的重试次数。

参数	设定	默认	功能	相关标志和字
重试次数	0 ～ 15	0	设定在功能块中执行 DeviceNet 显式报文或 FINS 报文发送时的指令发送重试次数。	A580.00 ～ A580.03

● FB 中的通信指令响应超时

此参数决定在功能块中执行通信指令的响应监控时间。

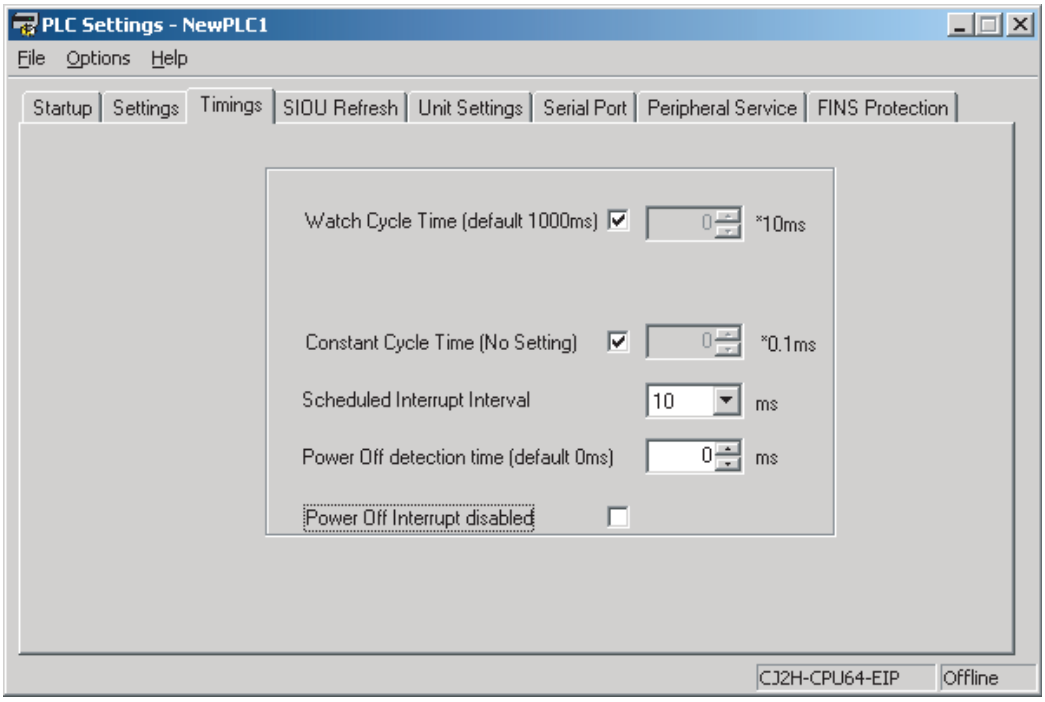
参数	设定	默认	功能	相关标志和字
FB 中的通信指令设定	0001 ～ 65535（单位：0.1 s，0.1 ～ 6553.5） 0000：2 s	0000：2 s	功能块中执行 FINS 指令时，在设定的时间内若无响应返回则发生响应超时。	A581

● FB 中的 DeviceNet 通信指令响应超时

此参数决定在功能块中执行 DeviceNet 通信指令的响应监控时间。

参数	设定	默认	功能	相关标志和字
FB 中的 DeviceNet 通信指令	0001 ～ FFFF（单位：0.1 s，0.1 ～ 6553.5） 0000：2 s	0000：2 s	功能块中执行显式报文指令时，在设定的时间内若无响应返回则发生响应超时。	A582

9-2-3 时间和中断设定



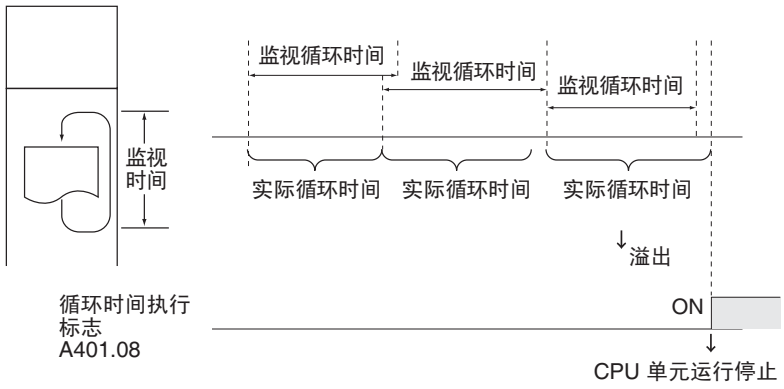
● 监视循环时间

此参数用于设定监视循环时间，以设置除默认值 （1000 ms）以外的设定值。

参数	设定	默认	功能	相关标志和字
监视循环时间	ON：默认（1,000 ms：1s） OFF：设定时间	ON（1,000 ms：1s）	设定为 OFF：设定监视循环时间。	A401.08 （循环时间太长标志）
	10 ～ 40,000 ms（单位 10ms）	0	这个设定只在上述参数被设定为 OFF 时才有效。如果循环时间超过这个设定，则循环时间太长标志 （A401.08）将会置 ON。	A264 和 A265（当前循环时间）

- 变更监视循环时间

  如果循环时间超出了监视 （最大）循环时间设定，则循环时间太长标志 （A401.08）将置 ON，并且 PLC 运行将停止。如果正常循环时间超出了默认监视循环时间设定 （1s），则必须要变更此参数。



注　监视循环时间的默认值为 1s（1,000 ms）。

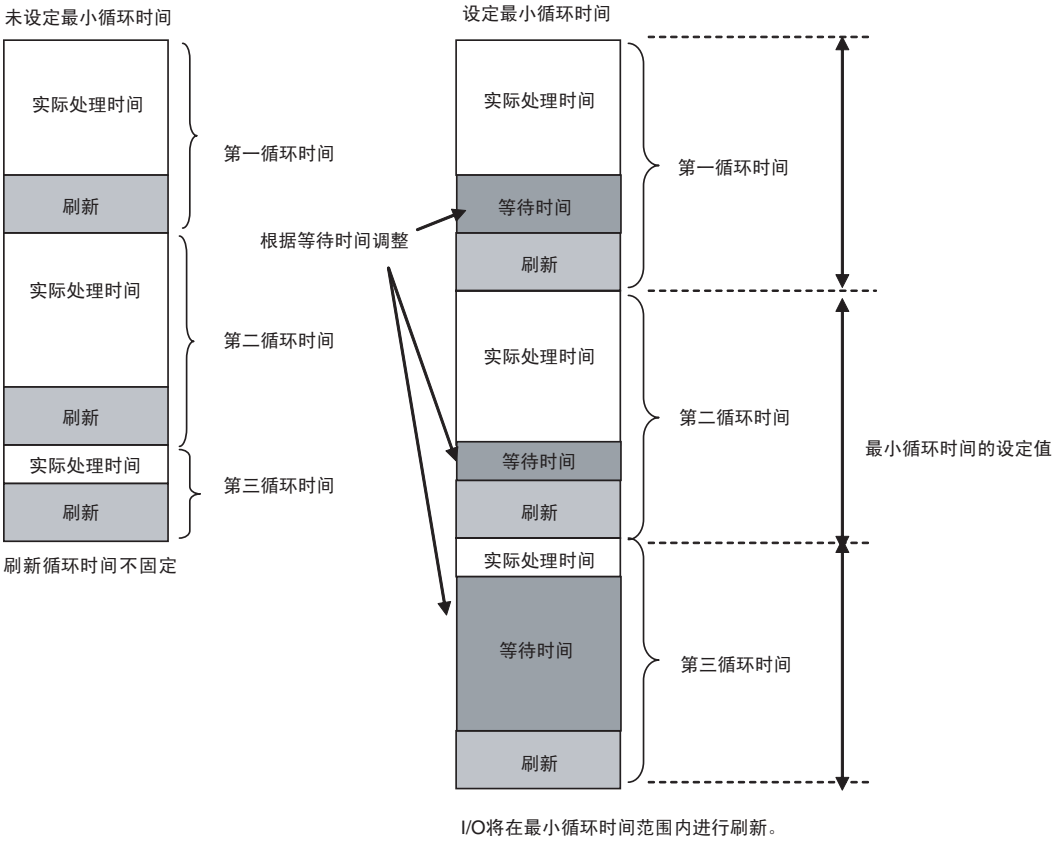
● 循环时间

使用最小循环时间功能时，以此参数设定最小循环时间 （默认：可变的循环时间）。

参数	设定	默认	功能	相关标志和字
固定循环时间	OFF：最小循环时间 ON：可变	ON （可变）	设定为 OFF 时为使用最小循环时间。若使用最小循环时间，则必须设定循环时间。	---
	0.2 ～ 32,000 ms（单位 0.1ms）	---	设定为 0.2 ～ 32000.0 以指定最小循环时间。若最小循环时间小于设定，则将进行扩展直到经过设定时间。可变的循环时间时将此设定置 0。	---

- 最小循环时间的有效条件

  将最小循环时间设定为非零值，以消除 I/O 响应中的不一致。仅当实际循环时间短于最小循环时间设定时此参数才为有效。若实际循环时间长于最小循环时间设定，则实际循环时间将保持不变更。



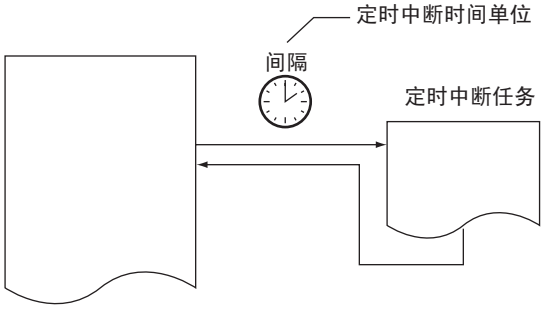
● 定时中断间隔

通过此参数将定时中断间隔中使用的时间单元设定为 1ms、0.1ms 或默认的 10 ms。

参数	设定	默认	功能	相关标志和字
定时中断间隔	• 10 ms • 1.0 ms • 0.1 ms	10 ms	以此设定确定用于定时中断间隔设定的时间单位。	---

- 定时中断时间单位

通过此参数设定用于定时中断间隔设定的时间单位。从程序以 MSKS（690）指令设定定时中断间隔。



● 电源 OFF 检测延迟时间

使用此参数，将电源 OFF 检测延迟时间设定为 0ms 以外的值。

参数	设定	默认	功能	相关标志和字
电源 OFF 检测时间	0 ～ 10 ms（单位 1ms）	0 ms	由此设定确定从检测电源中断到确认电源中断有多少延迟 （在电源电压下降至低于额定值的 85% 后，AC 电源约为 10 ～ 25 ms 而 DC 电源为 2 ～ 5 ms），默认设定是 0 ms。 当电源 OFF 中断任务允许时，则会在确认电源中断后进行执行。如果电源 OFF 中断任务禁止，则 CPU 会复位而且运行会停止。	---

注　当安装了 CJ1W-PD022 电源单元时不支持此设定。

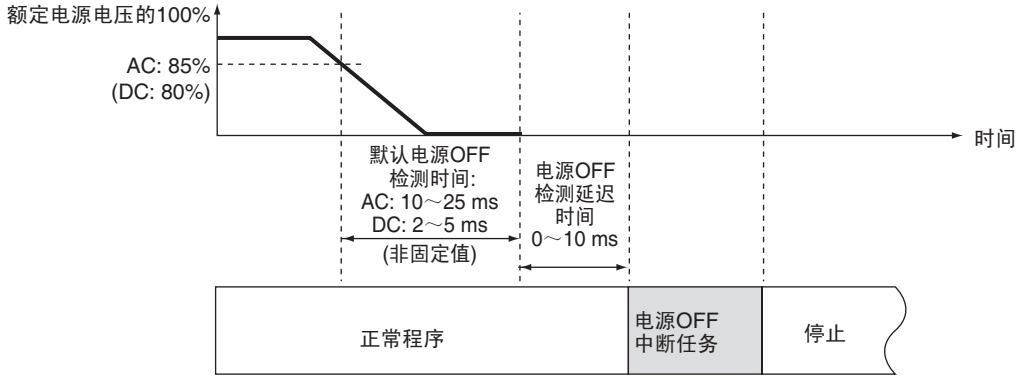
电源 OFF 检测延迟时间

可使用此参数确定从检测电源中断到建立电源中断且正常程序停止为止，需要延迟多少，其设定可在 0 ～ 10 ms 之间。

在因电源不良引起的暂时中断导致 PLC 运行停止时，将时间延长直至检测电源中断。

在启始电源中断检测时间后，内部 5V DC 电源下降至 0V DC 最长需用 10 ms，此 10 ms 为电源保持时间。其相当于电源 OFF 检测时间与确认电源中断所需处理时间相加。AC 电源时电源 OFF 延迟时间为 10 ～ 25ms（非固定值）， CJ1W-PD025 DC 电源单元时为 2 ～ 5 ms ， CJ1W-PD022 DC 电源单元时为 2 ～ 10 ms。

注　当安装了 CJ1W-PD022 电源单元时不支持此设定。（详情请参见 A-5 电源中断时操作）



注　电源 OFF 中断任务的执行时间必须小于 10 ms 减去电源 OFF 检测延迟时间。关于电源 OFF 中断任务的详情，请参见 5-2-3 中断任务。关于电源 OFF 时 CPU 单元操作的详情，请参见 A-5 电源中断时操作。

● 电源 OFF 中断禁止

此参数决定是否使电源 OFF 中断有效。

参数	设定	默认	功能	相关标志和字
电源 OFF 中断禁止	ON：禁止 OFF：允许	ON	当此设定置 OFF （允许）时，在电源被中断时会执行电源 OFF 中断任务。	---

注　当安装了 CJ1W-PD022 电源单元时不支持此设定。

● 电源 OFF 中断任务

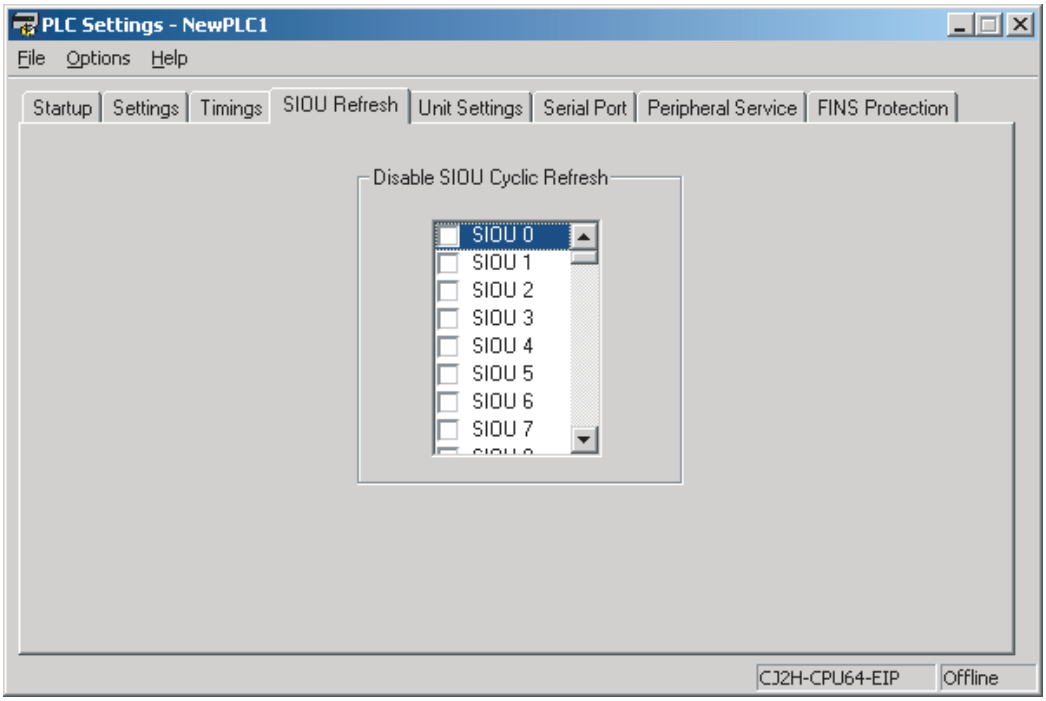
此参数决定了是否在检测到电源中断时执行电源 OFF 中断任务。（当此参数设定为 ON 时，在检测到电源中断时正常程序将停止）。

当经过了电源保持时间 （电源中断后的处理时间 + 电源 OFF 检测延迟时间）后，电源 OFF 中断任务将停止。最大电源保持时间为 10ms。

当已设定了电源 OFF 检测延迟时间时，请确保电源 OFF 中断任务可在有效时间内执行（10ms 减去电源 OFF 检测延迟时间）。

注　当安装了 CJ1W-PD022 电源单元时不支持此设定。（请参见 A-5 电源中断时操作）

9-2-4 特殊 I/O 单元循环刷新



禁止 SIOU 循环刷新

此参数可指定是否使禁止特殊 I/O 单元 （SIOU）的循环刷新。

参数	设定	默认	功能	相关标志和字
单元 0 ～ 95 的循环刷新 用于各 15 个单元一组的参数	OFF：允许 ON：禁止	OFF （禁止）	此设定决定是否在特殊 I/O 单元的循环刷新期间，在指定单元和特殊 I/O 单元的分配字 （10 字 / 单元）之间交换数据。	---

● 特殊 I/O 单元循环刷新禁止时

在一中断任务中，以 IORF （097）或 FIORF （225）指令对特殊 I/O 单元进行刷新时，或者以 IORD （222）或 IOWR （223）指令读取 / 写入数据到特殊 I/O 单元时，始终通过此参数禁止对应单元的循环刷新。

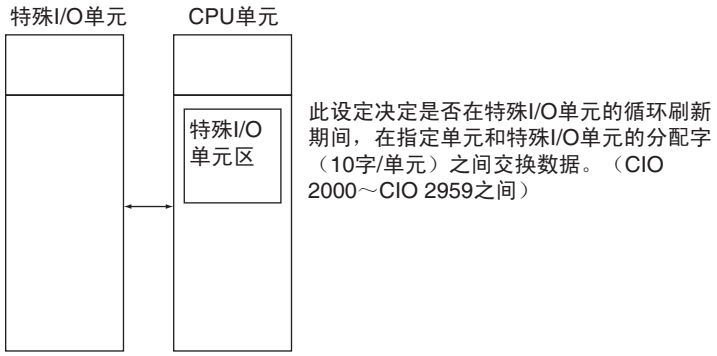
若循环刷新设定为 OFF （有效）且在中断任务中执行下述任一处理，则重复刷新将导致一非致命错误，并且重复刷新错误标志 （A402.13）将为 ON。

- 使用 IORF （097）或 FIORF （225）指令对同一特殊 I/O 单元进行 I/O 刷新。
- 对同一特殊 I/O 单元使用 IORD （222）或 IOWR （223）指令，从存储器区进行数据读取或写入。

● 禁止特殊 I/O 单元的循环刷新：

若在一中断任务中使用 IORF （097）， FIORF （225），IORD （222）或 IOWR （223）指令刷新特殊 I/O 单元，则特殊 I/O 单元的循环刷新始终为禁止。若在特殊 I/O 单元的循环刷新为有效时，在一中断任务中执行下述任一项，则将发生重复刷新错误（非致命）且重复刷新错误标志（A402.13）将为 ON。

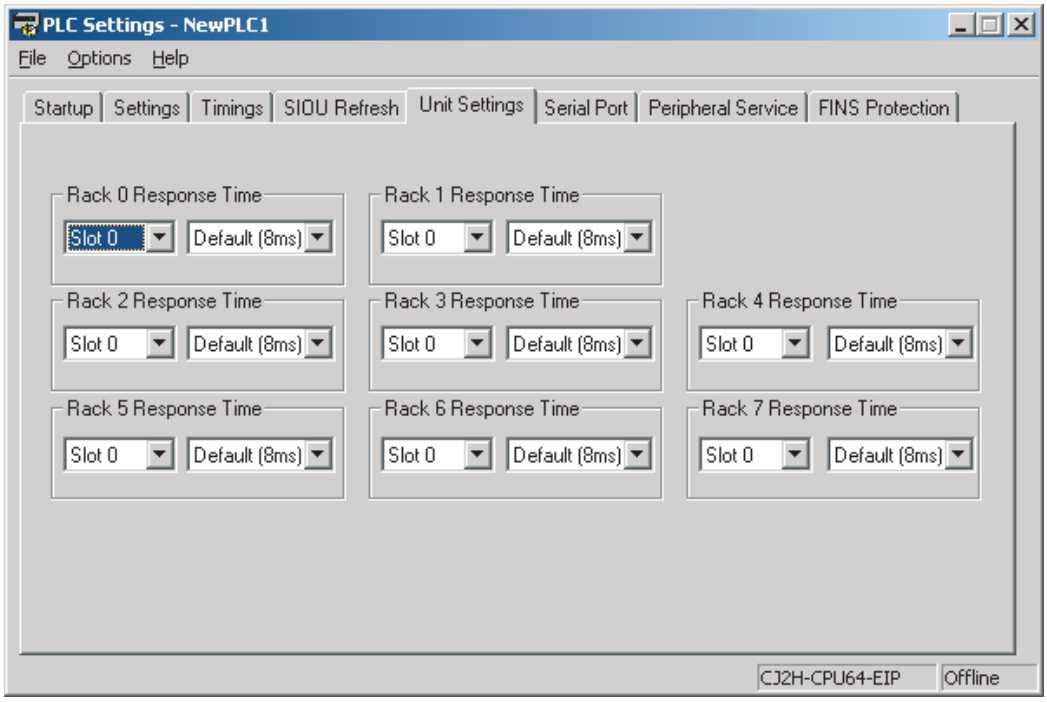
- 使用 IORF （097） /FIORF （225）指令对同一特殊 I/O 单元进行 I/O 刷新。
- 对同一特殊 I/O 单元使用 IORD （222）或 IOWR （223）指令，从存储器区进行数据读取或写入。



正确使用注意事项

禁止特殊 I/O 单元的循环刷新时，请确保在运行期间，程序中以 IORF（097）或 FIORF（225）指令刷新的单元 I/O 为至少每 11 秒。若非为每 11 秒刷新一次，否则特殊 I/O 单元中将发生 CPU 单元服务监控错误。（ 特殊 I/O 单元上的 ERH 指示灯和 RUN 指示灯将为亮灯）

9-2-5 基本 I/O 单元机架响应时间



机架 0 ～ 7 的响应时间

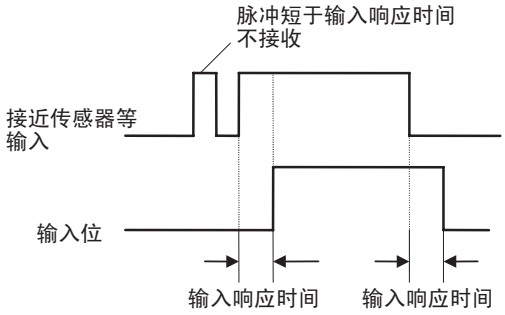
此参数用于设定基本 I/O 单元的输入响应时间为除默认 8ms 以外的值。

PLC 的电源供应必须为关闭，并在传送 PLC 设置到 CPU 单元后再接通电源。

参数	设定	默认	功能	相关标志和字
机架 0, 槽位 0 ～ 9 机架 1, 槽位 0 ～ 9 机架 2, 槽位 0 ～ 9 机架 3, 槽位 0 ～ 9 机架 4, 槽位 0 ～ 9 机架 5, 槽位 0 ～ 9 机架 6, 槽位 0 ～ 9 机架 7, 槽位 0 ～ 9	默认：8 ms 无滤波器 0.5 ms 1 ms 2 ms 4 ms 8 ms 16 ms 32 ms	8 ms	设定 CJ 系列基本 I/O 单元的输入响应时间（ON 响应时间 =OFF 响应时间） 若设定为 0ms，则会因单元间隔要素的延迟，在电源接通时产生最大 20 μs 的延迟，电源关闭时为最大 300μs。	A220 ～ A259: 基本 I/O 单元的实际输入响应时间

● 基本 I/O 单元机架响应时间变更

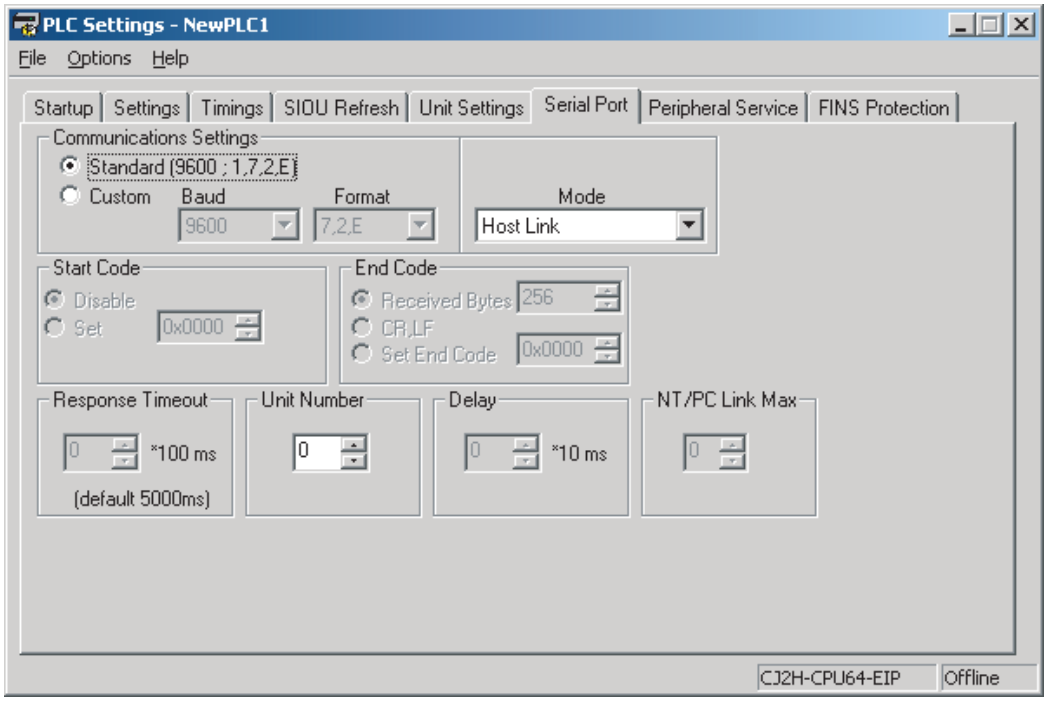
按机架或槽位，可对基本 I/O 单元设置机架响应时间。增大设定可减小振颤或噪音的影响，减小设定可允许较短输入脉冲的接收。不可将 ON 响应时间或 OFF 响应时间设置为小于循环时间。



当 PLC 打开时将传送机架响应时间设定到基本 I/O 单元。

当单元设定变更时，将其保存在 A220 ～ A259 中 （用于基本 I/O 单元的实际输入响应时间）。当在 PROGRAM 模式下的 PLC 变更 PLC 设置中的设定时， PLC 设置的设定将与单元中的实际设定不同。此时，可检查 A220 ～ A259 中的值，确定单元中实际设定的输入响应时间。

9-2-6 串行端口设定



当 CPU 单元上的 DIP 开关位 5 为 OFF （默认），则下述参数为有效。

通信设定

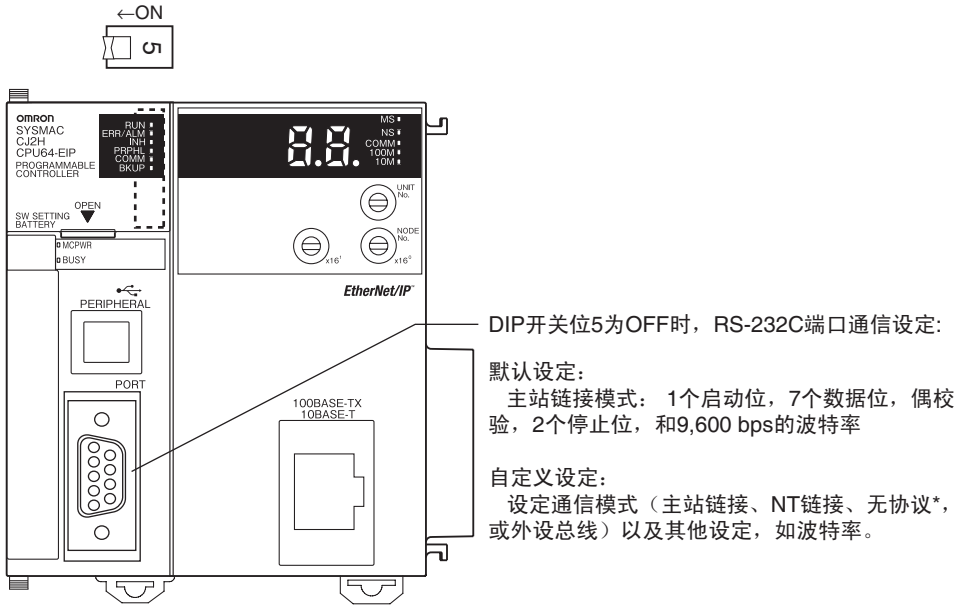
参数	设定	默认	功能	相关标志和字
通信设定	• 标准 （9600：1, 7, 2, E）：标准设定 • 自定义：任何设定	标准	标准设定 （9600：1, 7, 2, E）为用于上位链接 （Host Link）模式， 1 个启动位， 7 个数据位，偶校验， 2 个停止位和 9,600 bps 的波特率。 设置自定义设定以使用其他通信设定。	A619.02 （RS-232C 端口设定变更标志）

- 串行端口设定

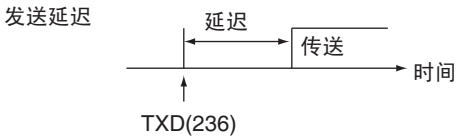
  需要对此些设定的默认设置进行变更时，请对 PLC 设置中的串行端口设定进行设置。选择无协议模式时请指定帧格式。

  通过 STUP （237）指令变更端口设定。当执行 STUP （237）时变更中标志 （A619.02）置 ON 且在端口设定已变更时置 OFF。

  当 CPU 单元前面的 DIP 开关位 5 为 ON 时，CPU 单元自动检测连接到 RS-232C 端口的编程设备（包括手持编程器）的通信参数。这些自动检测的参数并不保存在 PLC 设置中。



可在无协议模式下设定延迟。此延迟的操作如下图所示。



下表所述为用于无协议模式下传送和发送的报文格式。

		结束码设定		
		无	有	CR+LF
启始码设定	无	DATA	DATA ED	DATA CR+LF
	有	ST DATA	ST DATA ED	ST DATA CR+LF
接收字节数		数据：1 ～ 256 字节		

选择标准 （9600：1,7,2,E）

若选择标准 （9600：1,7,2,E），则为无条件的使用下述通信设定。

- 波特率：9,600 bps
- 启始位：1 位
- 数据长度：7 位
- 停止位：2 位
- 校验：偶
- 模式 （串行通信模式）：上位链接 （Host Link）

注　其他设定和模式为无效。

使用下述步骤设置自定义设定的通信。

1. 选择通信设定的自定义选项。
2. “模式”下选择通信模式。
3. 进行其他设定。

选择自定义

若选择自定义设定，则可选择下述波特率，参数和模式。

波特率 （选择自定义设定时）

设定	表示	设定	表示	相关标志和字
115200	11,5200 bps	4800	4,800 bps	A619.02 （RS-232C 端口设定变更中标志）
57600	57,600 bps	2400	2,400 bps	
38400	38,400 bps	1200	1,200 bps	
19200	19,200 bps	600	600 bps	
9600 （默认）	9,600 bps	300	300 bps	

● 格式 （选择自定义设定时）

选择数据	数据长度	停止位	校验	相关标志和字
7,1,E	7	1 位	偶	A619.02 （RS-232C 端口设定变更中标志）
7,1,O			奇	
7,1,N			无	
7,2,E （默认）		2 位	偶	
7,2,O			奇	
7,2,N			无	
8,1,E	8	1 位	偶	
8,1,O			奇	
8,1,N			无	
8,2,E		2 位	偶	
8,2,O			奇	
8,2,N			无	

模式 （选择自定义设定时）

选择用于内置 RS-232C 端口的串行通信模式。

参数	设定	默认	功能	相关标志和字
模式	• 上位链接 （Host Link） • NT 链接 （1:N） • RS-232C （无协议） • 外设总线 （工具总线） • 串行网关	上位链接 （Host Link） *1	设定是否在上位链接 （Host Link）模式下或其他串行通信模式下操作 RS-232C 端口。 *2	A619.02 （RS-232C 端口设定变更中标志）

*1 上位链接 （Host Link）（SYSMAC WAY）为用于与一般上位计算机连接的通信模式。

*2 1:1 NT 链接下无法进行 PT 通信。

● 各选择模式的相关设定

模式	相关设定
上位链接 （Host Link） （默认）	波特率，参数，单元编号
NT 链接 （1:N）	波特率，最大 NT/PC 链接
RS-232C （无协议）	波特率，参数，启始码，结束码，延迟
外设总线 （工具总线）	波特率
串行网关	波特率，参数，响应监控时间

● 上位链接 （Host Link）设定

通过计算机或其他上位设备，使用上位链接 （Host Link）进行通信时以及当波特率不为 9,600 bps 时使用此设定。

- 单元编号 （上位链接 （Host Link）模式下 CPU 单元用）

设定 PLC 从站的单元编号。

参数	设定	默认	功能	CPU 单元中更新时间	相关标志和字
单元编号	0 ～ 31	0	上位链接 （Host Link）下进行 1:N （N=2 ～ 32）连接时， CPU 单元的单元编号设定。	每个循环	A619.02 （RS-232C 端口设定变更中标志）

● NT 链接设定

与欧姆龙可编程终端 （NS 系列 /NT 系列）进行通信时，使用 NT 链接模式。

波特率 （bps）

参数	设定	默认	功能	相关标志和字
波特率	115,200, 38,400 bps	115,200 bps	从 CX-Programmer 设定此值时，推荐设定为 115,200 bps。	A619.02 （RS-232C 端口设定变更中标志）

最大 NT/PC 链接

参数	设定	默认	功能	相关标志和字
最大 NT/PC 链接	0 ～ 7	1	设定可连接到 PLC 的最大 PT 单元数。	A619.02 （RS-232C 端口设定变更中标志）

● RS-232C （无协议）设定

通过条形码读码器或其他设备执行无协议通信时，使用无协议模式。

启始码

参数	设定	默认	功能	相关标志和字
启始码	无效或设置	无效	设定以无协议通信进行发送或接收的数据帧格式是否使用启始码。	---
	0x0000 ～ 00FF（0000 ～ 00FF hex：“0x”与 hex 表示相同）	0x0000（00 hex）	以十六进制数值设定启始码	

结束码

参数	设定	默认	功能	相关标志和字
结束码	• 接收字节 （无结束码） • CR LF • 设定结束码	接收字节	设定以无协议通信进行发送或接收的数据帧格式所使用的结束码类型。	---
接收字节	256 或 1 ～ 255	256	如果选择了用于结束码的接收字节数，则字节单元中设定的数据长度将用于无协议通信的发送与接收。 结束码和启始码的长度不包括在数据长度中。 以一 TXD （236） /RXD （235）指令可接收 / 发送的默认最大数据长度为 256 字节。	
设定结束码	0x0000 ～ 00FF（0000 ～ 00FF hex：“0x”与 hex 表示相同）	0x00（00 hex）	指定 “结束代码”时，以十六进制值进行设定。	

延迟

参数	设定	默认	功能	相关标志和字
延迟	0 ～ 99990 ms （单位 10ms）	0 ms	执行 TXD （236）指令时，经过所设置的发送延迟后，将从 RS-232C 发送数据。	---

● 外设总线 （工具总线）设定

通过 CX-Programmer 或其他编程设备执行通信时，选择使用外设总线模式。

波特率

参数	设定	默认	功能	相关标志和字
波特率	9600, 19200, 38400, 57600, 115200 bps	115200 bps	指定外设总线 （工具总线）	A619.02 （RS-232C 端口设定变更中标志）

通过将 CPU 单元上的 DIP 开关位 5 设定为 ON， CX-Programmer 可自动检测波特率并连接外设总线。

● 串行网关设定

下述任一情况下，选择串行网关模式。

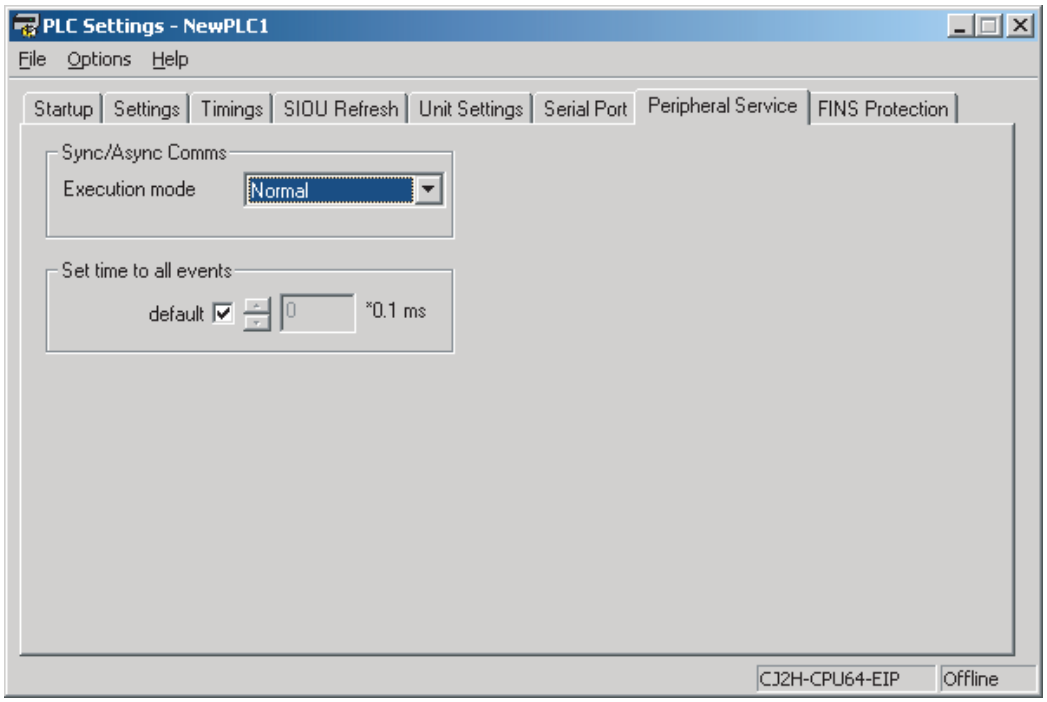
- 通过 CompoWay/F，与欧姆龙元器件进行通信时。
- 通过 Modbus，与变频器或其他设备进行通信时。
- 通过上位链接 （Host Link） FINS 指令，与下位 PLC 进行通信时。

响应监控时间

参数	设定	默认	功能 *1	相关标志和字
响应监控时间	5000 ms, 100 ～ 25500 ms （单位 100ms）	0 (50000 ms)	对从通过串行网关功能转换到指定协议的的 FINS 指令发送开始，到接收到响应为止的时间进行监控。 默认：5 s； PLC 设置：0.1 ～ 25.5 s	A619.02 （RS-232C 端口设定变更中标志）

*1 如果发生超时，则将返回 FINS 结束码 0205 hex （响应超时）到 FINS 源。

9-2-7 外设服务



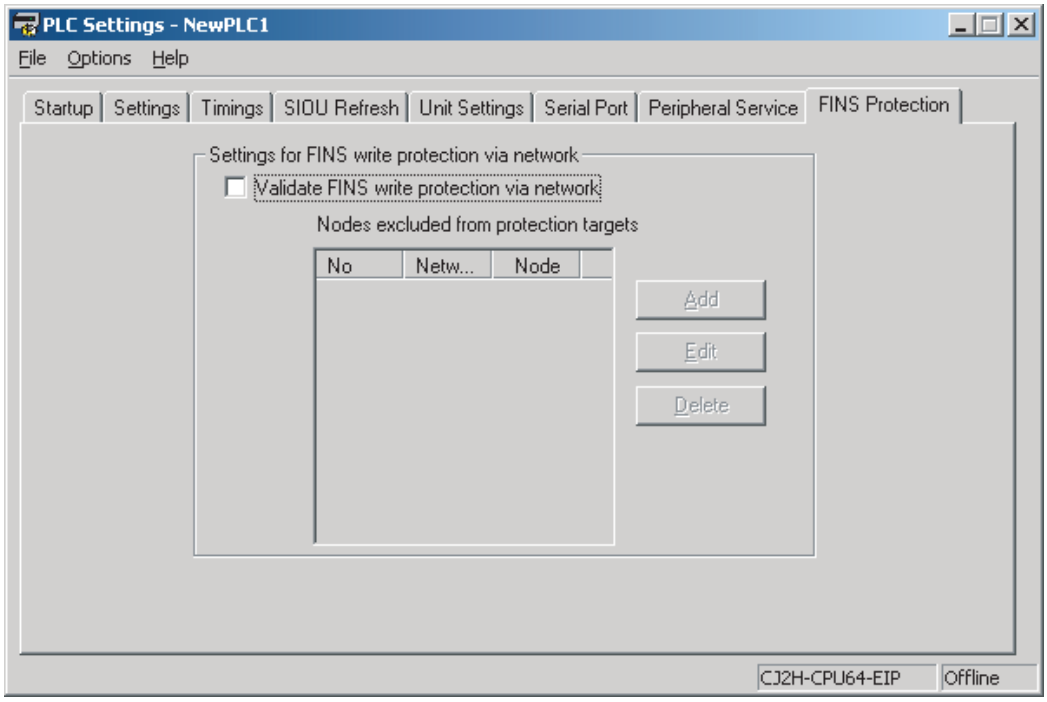
● CPU 处理模式

参数	设定	默认	功能	相关标志和字
执行模式	正常模式	正常模式	CJ2 CPU 单元仅支持正常模式。	---

● 设定所有事件时间

参数	设定	默认	功能	相关标志和字
设定所有事件时间	默认：循环时间的 10% 0.1 ～ 3,276.7 ms	循环时间的 10%（如果循环时间的 10% 小于 0.1 ms，则为 0.1 ms）	设定用于所有外设服务的最大时间。	---

9-2-8 FINS 保护



经由网络的 FINS 写保护设定

● 经由网络的 FINS 写保护有效

通过此参数设定经由网络的 FINS 指令写保护是否有效。

当使用串行连接方式以外的连接方式进行连接时，使用此参数以防止经由网络对存储器进行意外写入。若此参数为有效，则仅可从 “保护对象外节点”参数下设定的节点才可进行写入。

参数	设定	默认	功能	相关标志和字
经由网络 FINS 写保护有效	ON：禁止 FINS 写保护 OFF：允许 FINS 写保护	OFF	对使用 FINS 指令的 CPU 单元设定允许 / 禁止经由网络的写保护 （除串行连接以外）。	---

● 保护对象外节点

设定即使在保护为有效时，也可进行写入的节点。

参数	设定	默认	功能	相关标志和字
保护对象外节点	设定即使保护为有效时，也可进行 FINS 写入操作的节点和网络。将自动设定写保护对象外的节点数。 最大可设定 32 个节点。若不进行下述这些设定 （如，节点数为 0），则写保护将禁止除本地节点以外的所有节点写入。 *1			
	0 ～ 127	---	FINS 指令源网络地址	---
	1 ～ 255*2	---	FINS 指令源节点地址	---

*1 仅当 FINS 写保护为有效时，此设定才有效。

*2 可设定 255 （FF hex），以包括指定网络中的所有节点。

10

CPU 单元功能

本章节描述了 CPU 单元内置的功能。

10-1 时钟功能 10-3
10-1-1 时钟功能 10-3
10-1-2 存储器中保存的时间数据 10-4
10-1-3 自由运行定时器 10-7
10-2 循环时间 / 高速处理 10-8
10-2-1 最小循环时间 10-8
10-2-2 最大循环时间 10-9
10-2-3 监视循环时间 10-10
10-2-4 高速输入 10-10
10-2-5 后台执行 10-11
10-3 启动设定和维护 10-20
10-3-1 启动时和运行模式变更时的保持设定 10-20
10-3-2 电源 OFF 检测延迟设定 10-21
10-3-3 禁止电源 OFF 中断 10-22
10-3-4 RUN 输出 10-23
10-3-5 在启动时自动传输 10-24
10-4 单元管理功能 10-31
10-4-1 基本 I/O 单元管理 10-31
10-4-2 CPU 总线单元标志 / 位 10-32
10-4-3 特殊 I/O 单元标志 / 位 10-33
10-5 存储器管理功能 10-35
10-5-1 自动备份 10-35
10-5-2 EM 文件存储器功能 10-36
10-5-3 注释存储器 10-36
10-5-4 操作中替换全部程序 10-37
10-6 安全功能 10-44
10-6-1 使用 DIP 开关的写保护功能 10-44
10-6-2 使用密码的读保护 10-44
10-6-3 使用产品批号进行程序操作保护 10-49
10-6-4 FINS 命令写保护 10-50
10-6-5 PLC 名称 10-53

10-7 调试 10-56
10-7-1 强制置位 / 复位 10-56
10-7-2 测试输入 10-57
10-7-3 微分监控 10-57
10-7-4 在线编辑 10-58
10-7-5 关闭输出 10-59
10-7-6 数据跟踪 10-60
10-7-7 存储错误时的停止位 10-64
10-7-8 故障报警指令 10-65
10-7-9 模拟系统错误 10-65
10-7-10 故障点检测 10-67

10-1 时钟功能

10-1-1 时钟功能

时钟数据

内置 CJ2 CPU 单元中的时钟。保存时钟数据到下述的辅助区字中。

名称	地址	描述
时钟数据	在此保存 CPU 单元中内置时钟的时钟数据 （BCD）。	
	A351.00 ～ A351.07	秒：00 ～ 59 （BCD）
	A351.08 ～ A351.15	分：00 ～ 59 （BCD）
	A352.00 ～ A352.07	时：00 ～ 23 （BCD）
	A352.08 ～ A352.15	日：01 ～ 31 （BCD）
	A353.00 ～ A353.07	月：01 ～ 12 （BCD）
	A353.08 ～ A353.15	年：00 ～ 99 （BCD）
	A354.00 ～ A354.07	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

若未连接电池或电池电压低，则 A351 ～ A354 中的时钟数据不稳定。因此，请不要在未连接可靠电池的情况下使用时钟数据。

设定时钟数据

通过在线连接 CX-Programmer 和双击项目树中 PLC 时钟图标或执行 DATE （735）指令，可对此数据进行设定。此外，时钟数据可在秒与小时之间进行转换以及执行增量 / 减量。

● 时钟指令

指令	助记符	功能
时钟调整	DATE(735)	在指定源字的设定中变更内部时钟设定。
小时→秒	SEC(065)	将小时 / 分 / 秒表示的时间转换成仅以秒表示的等值时间。
秒→小时	HMS(066)	把以秒表示的时间转换成以小时 / 分 / 秒表示的等值时间。
日历加法	CADD(730)	把指定字中的日历数据和时间相加。
日历减法	CSUB(731)	把指定字中的日历数据减时间数据。

10-1-2 存储器中保存的时间数据

在辅助区字内自动保存电源置 ON 时的时间数据、电源置 OFF 时的时间数据、电源中断次数、总电源 ON 时间、用户存储器 （程序和参数）写入时间，以及运行开始和结束时间。请不要在未连接电池或电池电压过低时使用存储器中保存的时间数据。

电源 ON 时钟数据

在下述辅助区字内保存 PLC 置 ON 的年、月、日和时间。

名称	地址	描述
电源 ON 时钟数据 1	A720 ～ A722	数据为 BCD 格式。 A720.00 ～ A720.07 ： 秒 （00 ～ 59） A720.08 ～ A720.15 ： 分 （00 ～ 59） A721.00 ～ A721.07 ： 时 （00 ～ 23） A721.08 ～ A721.15 ： 日 （01 ～ 31） A722.00 ～ A722.07 ： 月 （01 ～ 12） A722.08 ～ A722.15 ： 年 （00 ～ 99）
电源 ON 时钟数据 2	A723 ～ A725	这写字中保存了倒数第二次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 3	A726 ～ A728	这写字中保存了倒数第三次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 4	A729 ～ A731	这写字中保存了倒数第四次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 5	A732 ～ A734	这写字中保存了倒数第五次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 6	A735 ～ A737	这写字中保存了倒数第六次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。

名称	地址	描述
电源 ON 时钟数据 7	A738 ～ A740	这写字中保存了倒数第七次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 8	A741 ～ A743	这写字中保存了倒数第八次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 9	A744 ～ A746	这写字中保存了倒数第九次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。
电源 ON 时钟数据 10	A747 ～ A749	这写字中保存了倒数第十次的电源 ON 时间。数据以 BCD 格式保存，存储格式与字 A720 ～ A722 相同。

电源中断时间

在下述辅助区字中保存 PLC 转为 OFF 的日期与时间。电源中断次数也保存在下述辅助区字中。

名称	地址	描述
电源中断时间	A512 和 A513	保存电源中断时间 （BCD）。每次发生电源中断时更新内容。 A512.00 ～ A512.07：秒 （00 ～ 59） A512.08 ～ A512.15：分 （00 ～ 59） A513.00 ～ A513.07：时 （00 ～ 23） A513.08 ～ A513.15：日 （01 ～ 31）
电源中断次数	A514	保存从电源首次置 ON 开始的电源中断次数 （二进制）。复位此值时，请将当前值覆写为 0000。

总电源 ON 时间

在下述辅助区字中保存 PLC 置 ON 的总时间。

名称	地址	描述
总电源 ON 时间	A523	保存 PLC 置 ON 的总时间 （16 位二进制），以 10 小时为增量单元。复位此值时，请将当前值覆写为 0000 hex.。

用户程序和参数版本时间

这些辅助区字中保存写入数据到用户程序或参数（如，PLC 设置，I/O 表，路由表或 CPU 总线单元设置）的时间和日期。

名称	地址	描述
用户程序时间	A090 ～ A093	这些字中以 BCD 格式保存用户程序最近覆写的日期与时间。 A090.00 ～ A090.07：秒 （00 ～ 59） A090.08 ～ A090.15：分 （00 ～ 59） A091.00 ～ A091.07：时 （00 ～ 23） A091.08 ～ A091.15：日 （01 ～ 31） A092.00 ～ A092.07：月 （01 ～ 12） A092.08 ～ A092.15：年 （00 ～ 99） A093.00 ～ A093.07：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）
参数时间	A094 ～ A097	这些字中以 BCD 格式保存参数最近覆写的日期与时间。 A094.00 ～ A094.07：秒 （00 ～ 59） A094.08 ～ A094.15：分 （00 ～ 59） A095.00 ～ A095.07：时 （00 ～ 23） A095.08 ～ A095.15：日 （01 ～ 31） A096.00 ～ A096.07：月 （01 ～ 12） A096.08 ～ A096.15：年 （00 ～ 99） A097.00 ～ A097.07：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）

操作启动 / 结束时间

这些辅助区字中自动保存 PLC 运行启动和停止的日期与时间。

名称	地址	描述
操作启动时间	A515 ～ A517	将因运行模式切换为 RUN 或 MONITOR 模式而导致操作起始的时间，以 BCD 形式保存于此。 A515.00 ～ A515.07：秒 （00 ～ 59） A515.08 ～ A515.15：分 （00 ～ 59） A516.00 ～ A516.07：时 （00 ～ 23） A516.08 ～ A516.15：日 （01 ～ 31） A517.00 ～ A517.07：月 （01 ～ 12） A517.08 ～ A517.15：年 （00 ～ 99） 直到运行开始时，之前的起始时间在电源打开后被保存。
操作结束时间	A518 ～ A520	将因运行模式变更为 PROGRAM 模式而导致操作停止的时间，以 BCD 形式保存在此。 A518.00 ～ A518.07：秒 （00 ～ 59） A518.08 ～ A518.15：分 （00 ～ 59） A519.00 ～ A519.07：时 （00 ～ 23） A519.08 ～ A519.15：日 （01 ～ 31） A520.00 ～ A520.07：月 （01 ～ 12） A520.08 ～ A520.15：年 （00 ～ 99） 若在操作中发生错误，错误时间将被保存。若此后运行模式变更为 PROGRAM 模式，进入 PROGRAM 模式的时间将被保存。

10-1-3 自由运行定时器

电源置 ON 后所使用的系统定时器为包含在辅助区字 A0、A1 和 A2 中。这些定时器可用于在不使用定时器指令的情况下计算时间间隔。

例如，可对处理 A 和处理 B 之间的间隔进行计算而无须计时器指令。这是由计算处理 A 的 A0 值和处理 B 的 A0 值之间的差异来实现的。该间隔以 10 ms 为单位计算。

名称	地址	描述
以 10ms 为增量的自由运行定时器	A0	当电源打开时设置 0000 hex，且该值每 10 ms 自动增加 1。该值达到 FFFF hex （655,350 ms）后返回 0000 hex，然后继续每 10 ms 自动增加 1。
以 100ms 为增量的自由运行定时器	A1	当电源打开时设置 0000 hex，且该值每 100 ms 自动增加 1。该值达到 FFFF hex （6,553,500 ms）后返回 0000 hex，然后继续每 100 ms 自动增加 1。
以 1s 为增量的自由运行定时器	A2	当电源打开时设置 0000 hex，且该值每 1s 自动增加 1。该值达到 FFFF hex （65,535 s）后返回 0000 hex，然后继续每 1s 自动增加 1。

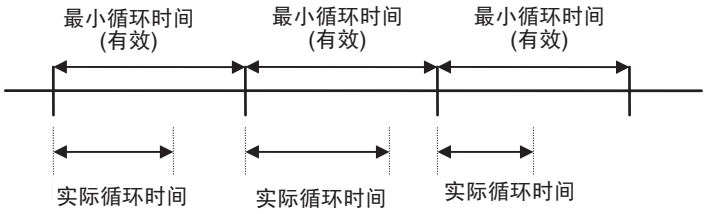
注　当运行模式变更为 RUN 模式时，自动以 1 为增量进行继续。

10-2 循环时间 / 高速处理

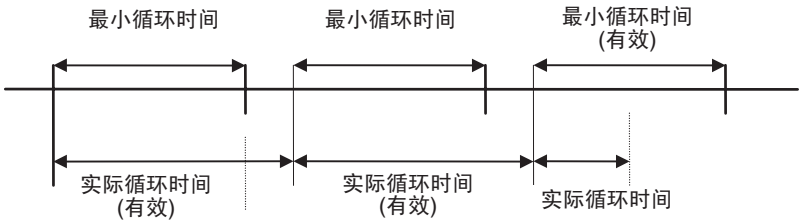
10-2-1 最小循环时间

可在 PLC 中设置最小 （或固定）循环时间。可通过以固定周期重复执行程序，消除 I/O 响应的变化。

在 PLC 设置中指定最小循环时间 （0.1 ～ 32,000 ms），以 0.1ms 为单位。

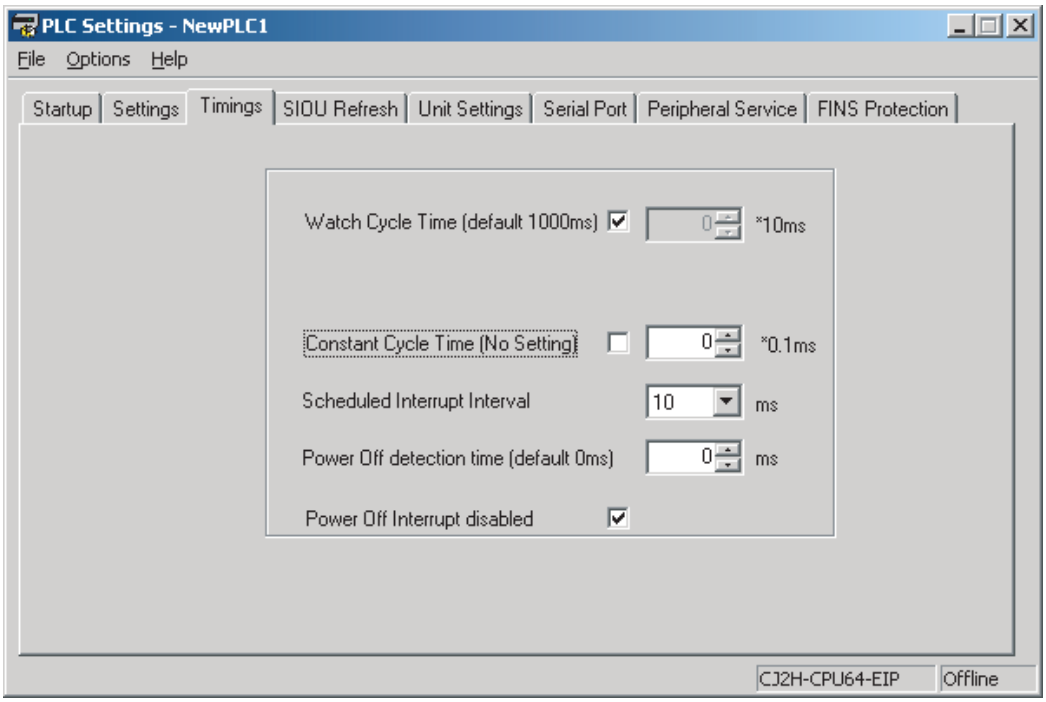


如果实际循环时间长于最小循环时间设定，则最小循环时间功能将为无效并以实际循环时间进行运行。



- PLC 设置

当使用 CX-Programmer 时，请在 “Timing”（定时）标签页上进行设定。

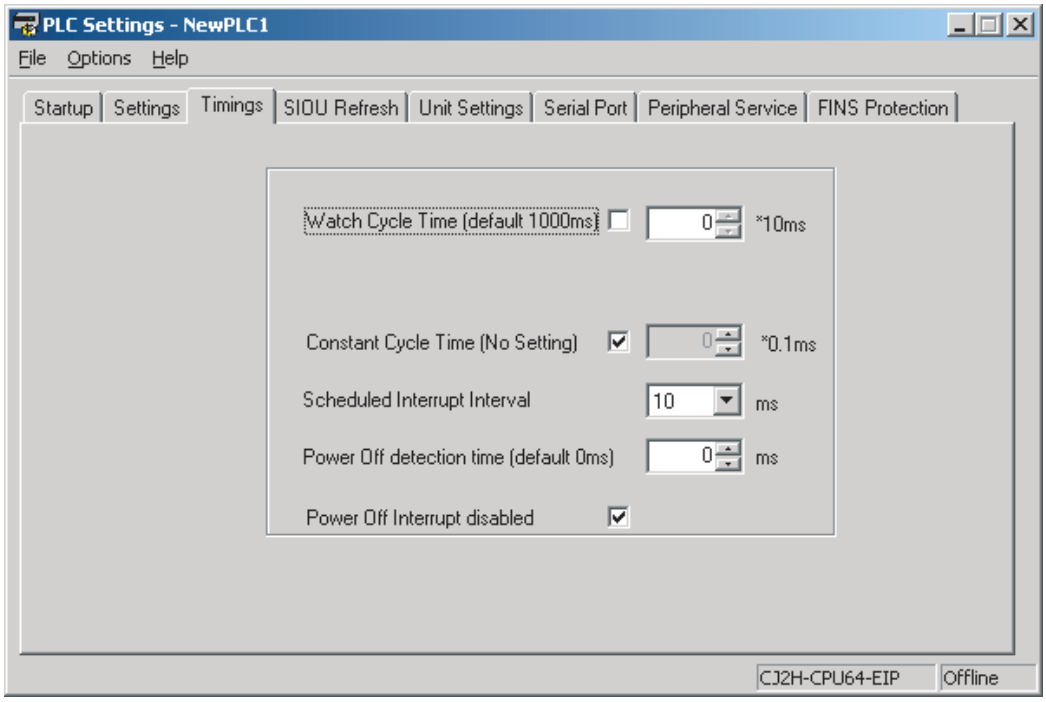


10-2-2 最大循环时间

如果循环时间超出了最大循环时间设定，则 CPU 单元将停止运行。循环时间太长标志 （A401.08）将置 ON。默认最大循环时间为 1s。

● PLC 设置

使用 CX-Programmer 时，在 “Timings”（定时）标签页上的监视循环时间中设定最大循环时间。



● 辅助区标志和字

名称	地址	描述
循环时间太长标志	A401.08	如果循环时间超出了监视循环时间设定，则 A401.08 将置 ON。

10-2-3 监视循环时间

每个循环中，辅助区的 A262 ～ A263 中保存最大循环时间、 A264 ～ A265 和 A266 ～ A267 中保存当前循环时间。

● 辅助区标志和字

名称	地址	描述
最大循环时间 （以 0.1ms 为增量）	A262 和 A263	这些字中保存以 0.1 ms 为增量单位的最大循环时间，以每循环进行刷新，并以 8 位十六进制数保存在下述范围中： 0 ～ 429,496,729.5 ms （0 ～ FFFF FFFF） 最低位保存在 A262 中，最高位保存在 A263 中。
当前循环时间 （以 0.1ms 为增量）	A264 和 A265	这些字中保存以 0.1 ms 为增量单位的当前循环时间，以每循环进行刷新，并以 8 位十六进制数保存在下述范围中： 0 ～ 429,496,729.5 ms （0 ～ FFFF FFFF） 最低位保存在 A264 中，最高位保存在 A265 中。
当前循环时间 （以 0.01ms 为增量）	A266 和 A267	这些字中保存以 0.01 ms 为增量单位的当前循环时间，以每循环进行刷新，并以 8 位十六进制数保存在下述范围中： 0 ～ 42,949,672.95 ms （0 ～ FFFF FFFF） 最低位保存在 A266 中，最高位保存在 A267 中。

可通过 CX-Programmer 读取最后 8 个循环的平均循环时间。

附加信息

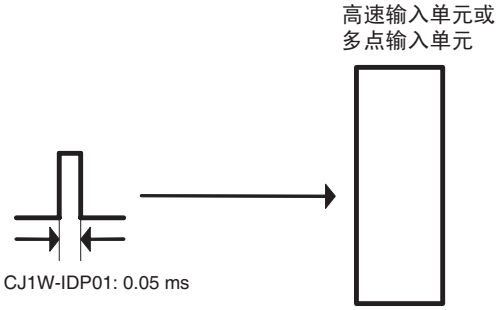
可使用下述方式，减少 CJ 系列 PLC 的循环时间。

- 将不执行的任务置于 WAIT （等待）状态。
- 将 JMP （004），CJP （510）或 CJPN （511）指令与 JME （005）指令结合使用，以跳过不需要执行的程序段。

10-2-4 高速输入

当要接收短于循环时间的脉冲时，请使用 CJ1W-IDP01 高速输入单元。

CJ1W-IDP01 高速输入单元可接收脉冲宽度 （ON 时间） 0.05 ms 的脉冲。



进行输入刷新时，输入内部存储器的出入将被清除。

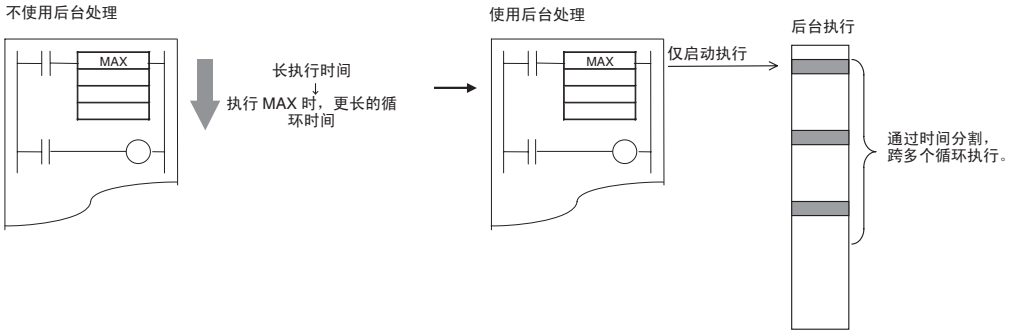
10-2-5 后台执行

进行表数据处理（如数据搜索）和文本串处理（如文本串搜索）等需要较多处理时间的操作，可能因其需要较长的指令执行时间而造成循环时间的暂时性延长，并导致循环时间的较大波动。

可使用后台执行（时间分割）将下述指令分割到多个循环进行执行，以控制循环时间的波动。可通过 PLC 设置将各指令类型设定为后台执行。

- 表数据处理指令
- 文本串处理指令
- 数据移位指令（仅异步移位寄存器）

通过将上述指令设置为后台执行，可有效控制循环时间的临时增长。



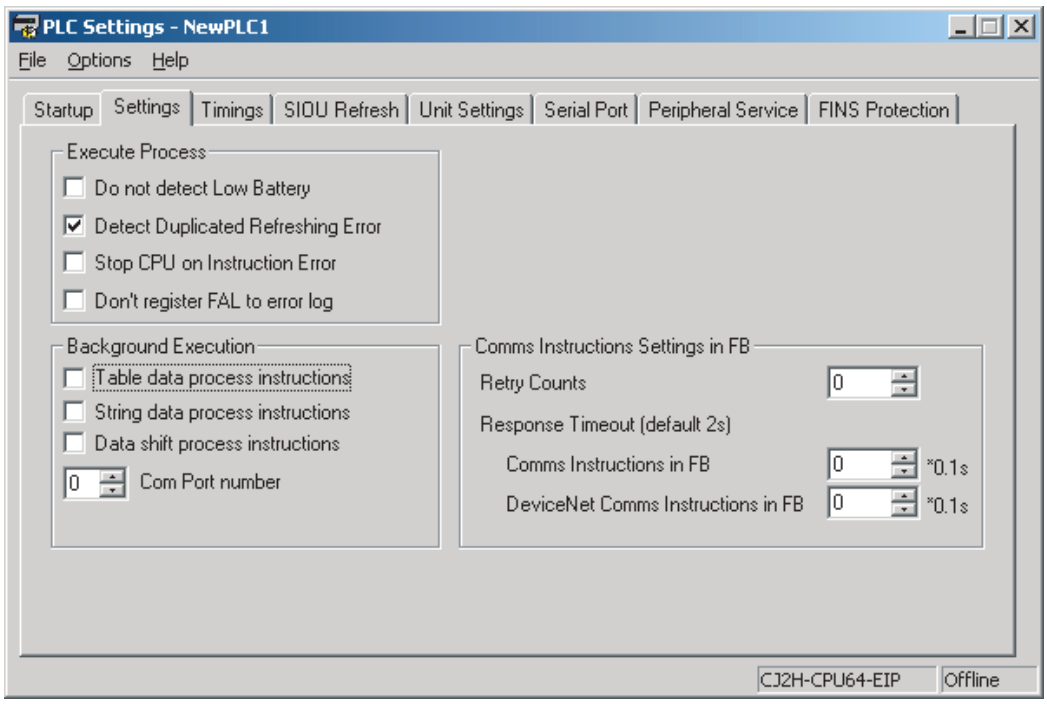
● 适用指令

下述指令在用于功能块时将无法执行后台处理。这些指令将以正常处理进行执行。

分组	指令	助记符
表数据处理指令	搜索数据	SRCH
	交换字节	SWAP
	找出最大值	MAX
		MAXL
		MAXF
		MAXD
	找出最小值	MIN
		MINL
		MINF
		MIND
	求和	SUM
	帧检查和	FCS
数据移位指令	异步移位寄存器	ASFT
文本串处理指令	字符串传送	MOV$
	字符串链接	+$
	从左边取出字符	LEFT$
	从右边取出字符	RGHT$
	从中间取出字符	MID$
	找出字符串	FIND$
	字符串长度	LEN$
	替代字符串	RPLC$
	删除字符串	DEL$
	交换字符串	XCHG$
	清除字符串	CLR$
	插入字符串	INS$

● 步骤

1. 使用 CX-Programmer 在 PLC 设置对话框中的设定标签页上对后台执行进行设定，以使所需指令的后台执行为有效。



2. 在PLC设置中设定用于后台执行的逻辑端口编号。此端口编号将用于所有在后台进行处理的指令。

所有的后台执行都将使用同一个端口。因此在执行一个指令的后台处理时其他指令将不会开始执行。使用通信端口允许标志对指定后台执行的指令进行控制，以确保在同一时间仅有一个指令在进行后台处理。

3. 如果执行某一已指定后台执行的指令，则仅当在循环中符合执行条件并且在同一循环内无法完成时才开始执行。

4. 开始后台执行时，指定端口的通信端口允许标志将置 OFF。

5. 后台执行将跨多个循环继续执行。

6. 当处理执行完成时，指定端口的通信端口允许标志将置 ON，以允许其他指令在后台进行处理。

● 正常执行指令与后台执行指令的区别

正常执行指令与后台执行指令的区别如下所示。

输出到变址寄存器 （IR）

如果执行 MAX （182），MAXL （174），MAXF （176），MAXD （178），MIN （183），MINL （175），MINF （177）或 MIND （179），把包含有最大或最小值字的 I/O 存储器变换地址输出到一个变址寄存器，则地址将不会输出到变址寄存器而是输出到 A595 和 A596。

为了将地址保存在一个变址寄存器中（与正常执行方式相同），须用一个数据移动指令（如：MOVL（498））将在 A595 和 A596 中的地址复制到一个变址寄存器上。

条件标志

条件标志不会随后台进行处理指令的执行而更新。为了存取条件标志状态，执行一个以同样方式影响条件标志的指令（如下例所示），随后存取条件标志。

输出到变址寄存器 IR00

如果执行 SRCH （181），把包含有匹配值（如果有几个，选第一个字）字的 I/O 存储器变换地址输出到一个变址寄存器，地址将不会输出到变址寄存器而是输出到 A595 和 A596。

输出到数据寄存器（SRCH （181））

如果执行 SRCH（181），把匹配数据输出到一个数据寄存器，这个数据不会输出到数据寄存器而是输出到 A597。

匹配文本字符串

如果 SRCH （181）找到匹配数据，它不会将等于标志 （=）置 ON，而是将 A598.01 置 ON。

指令错误

如果在后台处理的一个指令发生一个指令执行错误或非法访问错误，ER 或 AER 标志不会置 ON，而是 A395.10 置 ON。 A395.10 将一直为 ON，直到在后台处理下一条指令。

输出到数据寄存器（MAX （182）或 MIN （183））

如果 MAX（182），MAXL（174），MAXF（176），MAXD（178），MIN（183），MINL（175），MINF （177）或 MIND （179）和指定为最小或最大值输出字的一个数据寄存器 （DR0 ～ DR15）一起执行，将发生一个指令执行错误，且 ER 标志将变为 ON。

● 辅助区标志和字

名称	地址	描述
通信端口允许标志	A202.00 ～ A202.07	当用相应端口号执行网络指令或以相应端口号运行后台执行时置 ON。位 00 ～ 07 对应通信端口 0 ～ 7。 如果对存储卡进行简捷备份操作，进行写或比较操作时，将自动分配通信端口，并且在操作期间相应标志变为 ON，当操作完成后，将变为 OFF。
通信端口错误标志	A219.00 ～ A219.07	网络指令执行期间发生错误时变为 ON，正常完成时 OFF。位 00 ～ 07 对应通信端口号 0 ～ 7。 如果对存储卡进行简捷备份操作，进行写或比较操作时，将自动分配通信端口。如果一个错误发生相应标志变为 ON，并当简捷备份操作正常结束时，变为 OFF。
通信端口完成码	A203 ～ A210	执行网络指令后，将在这些字中保存相应端口号的完成码。当后台执行完成时，内容将被清除为 0000 hex。字 A203 ～ A210 对应通信端口 0 ～ 7。 如果对存储卡进行简捷备份操作，进行写或比较操作时，将自动分配通信端口，并将一个完成码存储在相应的字中。
后台执行 ER/AER 标志	A395.10	当后台执行的一条指令发生执行错误或非法访问错误时，变为 ON。当电源接通或运行开始时，变为 OFF。
后台执行 IR00 输出	A595 和 A596	当一后台执行指令的输出为指定用于一个变址寄存器时，这些字接收输出。 IR00 没有输出。 范围：0000 0000 ～ FFFF FFFF hex 低 4 位数字：A595，高 4 位数字：A596
后台执行 DR00 输出	A597	当一后台执行指令的输出为指定用于一个数据寄存器时，这些字接收输出。 DR00 没有输出。 范围：0000 ～ FFFF hex
后台执行 等于标志 （=）输出	A598.01	为在后台执行的 SRCH （181）找到匹配数据后，这个标志变为 ON。

附加信息

CPU 单元中的内部逻辑端口既可用于后台执行也可用于下述指令：

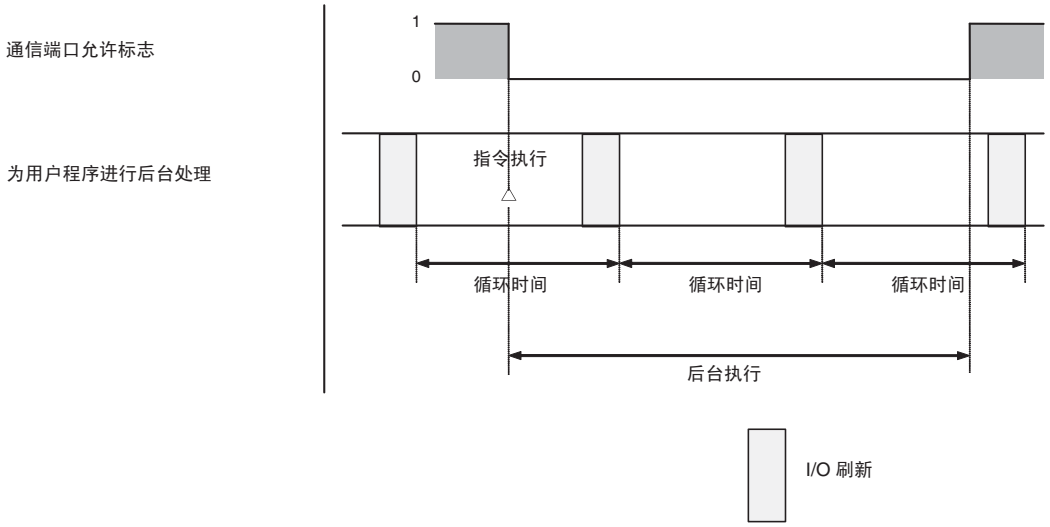
- SEND （090）， RECV （098），和 CMND （490）（网络通信指令）。
- PMCR （260）（协议宏）
- TXDU （256）和 RXDU （255）（使用串行通信指令单元的无协议通信指令）

后台指令和以上指令不可在同一端口上同时执行。使用通信端口允许标志，以确保在任一时间在每个端口仅仅执行一种指令。

注　当一个端口的通信端口允许标志变为 OFF 时，如果有一指令指定在这端口进行后台执行，则 ER 标志将变为 ON，并且将不执行后台指令。

● 通信端口允许标志

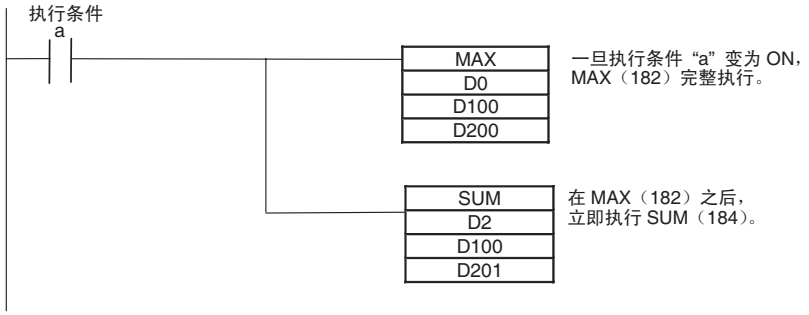
当处理未在端口上进行时，通信端口允许标志为 ON，当处理正在端口上进行时，通信端口允许标志为 OFF。



程序示例 1

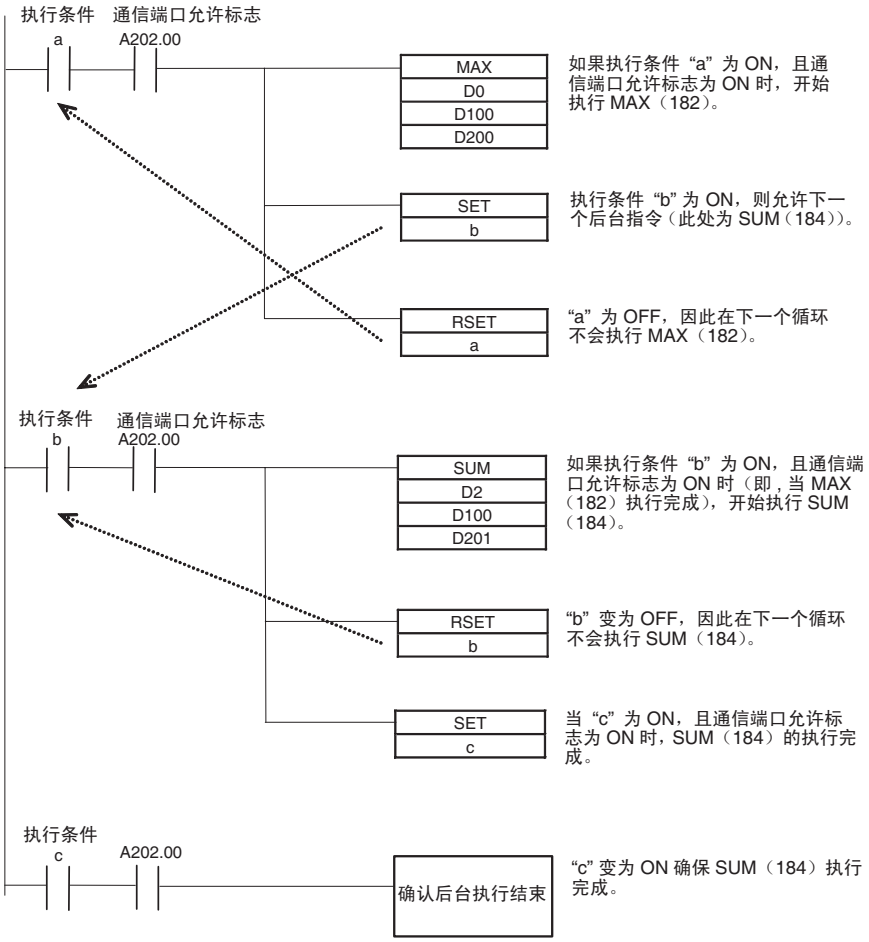
● 无后台执行的常规程序

如下所示，当指令执行后，完成处理。



● 带后台执行程序

使用后台执行功能，变更程序，以使仅当特定的通信端口允许标志变为 ON （即：仅当该端口未被用于后台执行或网络通信）时，MAX（182）才能执行。同样，输入状态受到 SET 和 RESET 指令控制，确保以正确的次序执行处理。（在下列实例中通信端口 0 用于后台执行）。

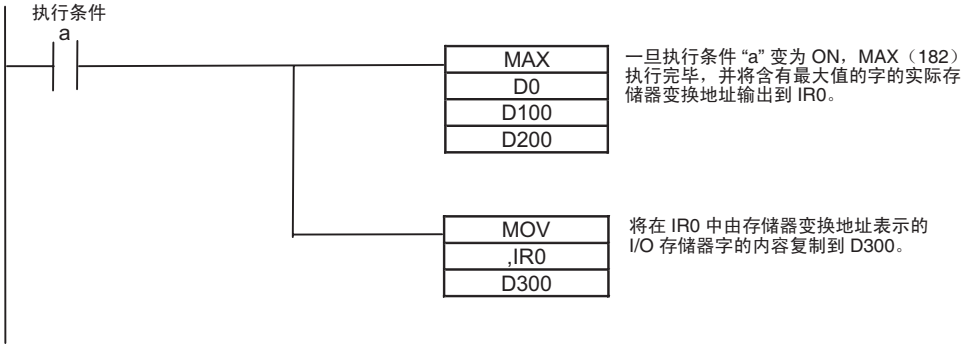


程序示例 2

本例说明了当变址寄存器的输出被指定 （如：MAX （182）， MIN （183），和 SRCH （181））时的后台执行。

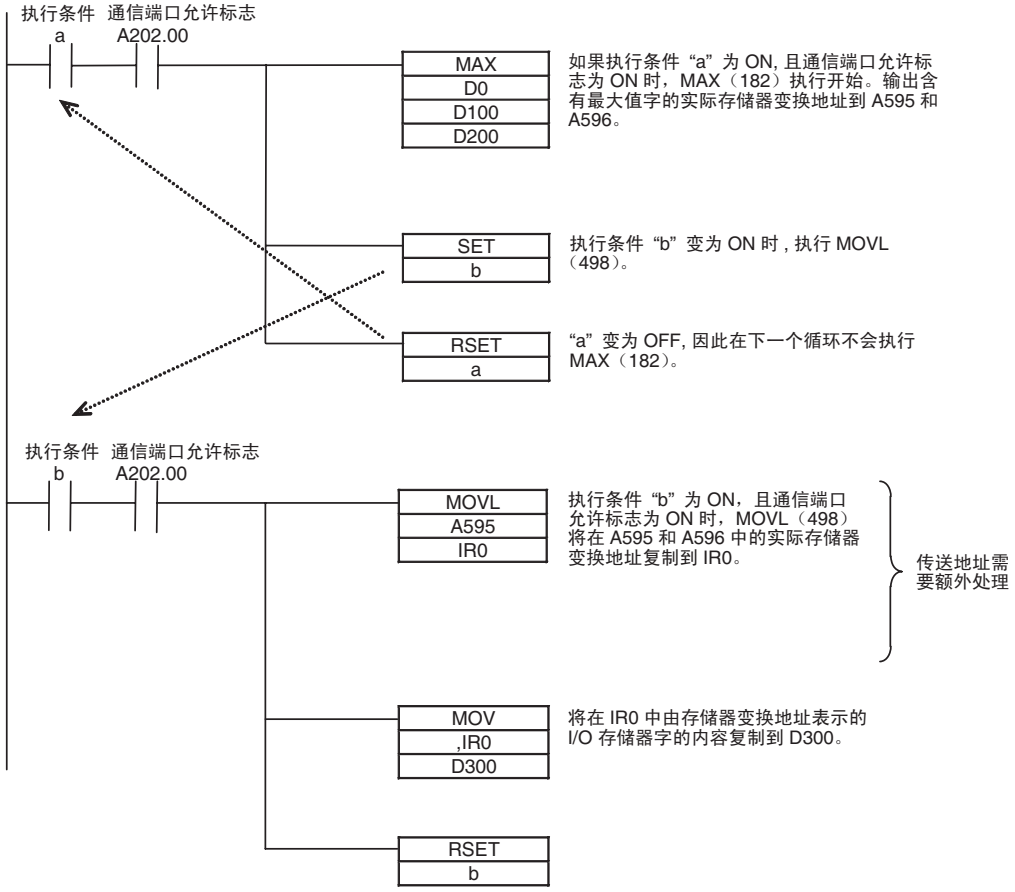
● 无后台执行的常规程序

如下所示，输出含有最大或最小值字的 PLC 存储器变换地址到一个变址寄存器。



● 带后台执行程序

通过后台执行功能，输出含有最大或最小值字的 PLC 存储器变换地址到 A595 和 A596。随后使用 MOVL （498）将 PLC 存储器变换地址传送到变址寄存器。

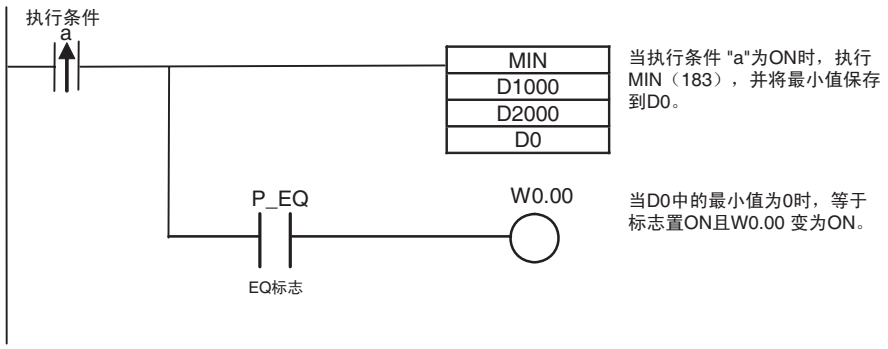


程序示例 3

此示例所示为参考条件标志时的后台执行。

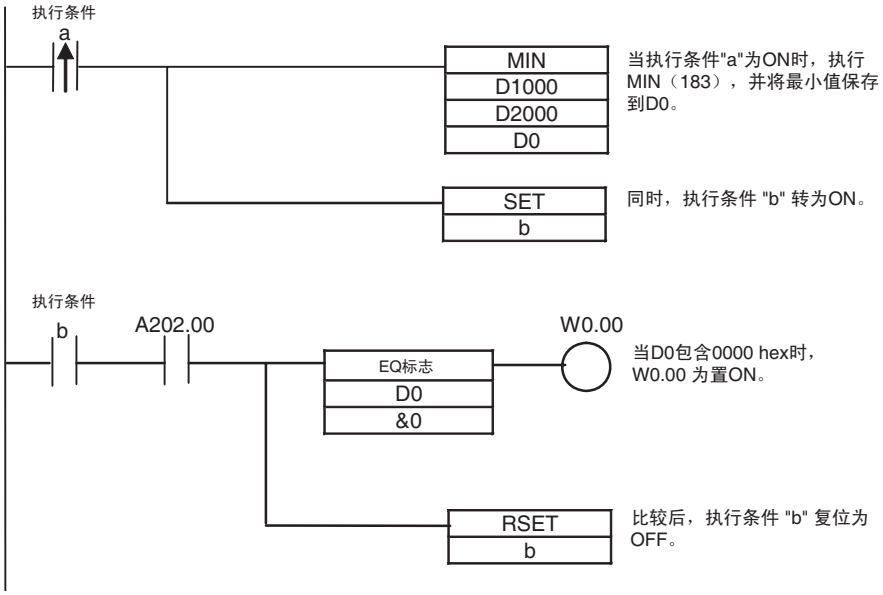
● 无后台执行的常规程序

执行 MIN （183）后立即确认等于标志 （=）的状态，以检查是否通过执行 MIN （183）找出最小值。



● 带后台执行程序

如下图所示，使用 AND = （300）指令以检查由 MIN （183）找出的最小值是否为 0。



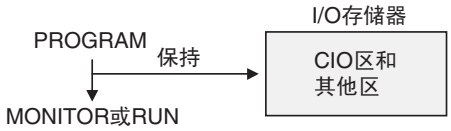
10-3 启动设定和维护

10-3-1 启动时和运行模式变更时的保持设定

运行模式变更

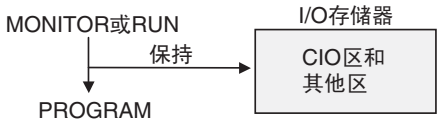
● 启动程序执行

当 CPU 单元从 PROGRAM 模式转换为 RUN/MONITOR 模式以启动程序执行时，将 IOM 保持位（A500.12）置 ON 以保持 I/O 存储器中的所有数据。



● 停止程序执行

IOM 保持位（A500.12）为 ON 时，则当 CPU 单元从 RUN/MONITOR 模式转换为 PROGRAM 模式以停止程序执行时， I/O 存储器中的所有数据将被保持。



下述区为保持：CIO 区（I/O 区域，数据链接区，CPU 总线单元区，指定 I/O 单元区，DeviceNet 区和内置 I/O 区），工作区，定时器完成标志和定时器当前值。

● 辅助区标志和字

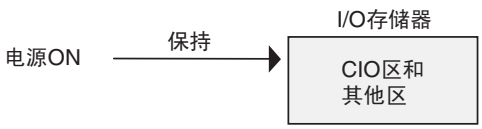
名称	地址	描述
IOM 保持位	A500.12	当此位为 ON 时，则当运行模式在 PROGRAM 和 RUN/MONITOR 之间切换时或电源打开时，都将保持所有的 I/O 存储器数据。 ON：运行模式变更时 I/O 存储器为保持。 OFF：运行模式变更时 I/O 存储器为清除。

IOM 保持位为 ON 时，程序执行停止时的所有输出单元的输出都将为保持。当程序再次启动时，输出将以程序停止前的状态继续执行。

（当 IOM 保持位为 OFF，则将在清除输出后执行指令）。

PLC 电源 ON

为了在电源接通时保持 I/O 存储器中的所有数据，必须将 IOM 保持位 （A500.12）设置为 ON 并在 PLC 设置中对其设定保护。

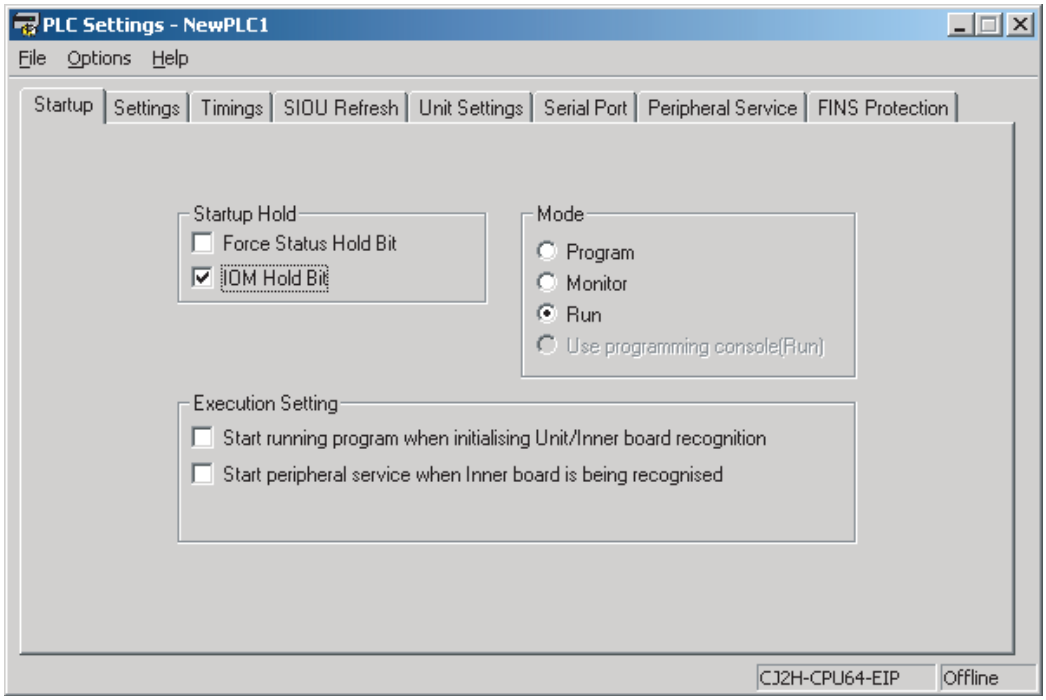


● 辅助区标志和字

名称	地址	描述
IOM 保持位	A500.12	当此位为 ON 时，则当运行模式在 PROGRAM 和 RUN/MONITOR 之间切换时或电源打开时，都将保持所有的 I/O 存储器数据。 ON：运行模式变更时 I/O 存储器为保持。 OFF：运行模式变更时 I/O 存储器为清除。

● PLC 设置

使用 CX-Programmer 时，在启动标签页上的启动保持区中选择 IOM 保持位选择框。



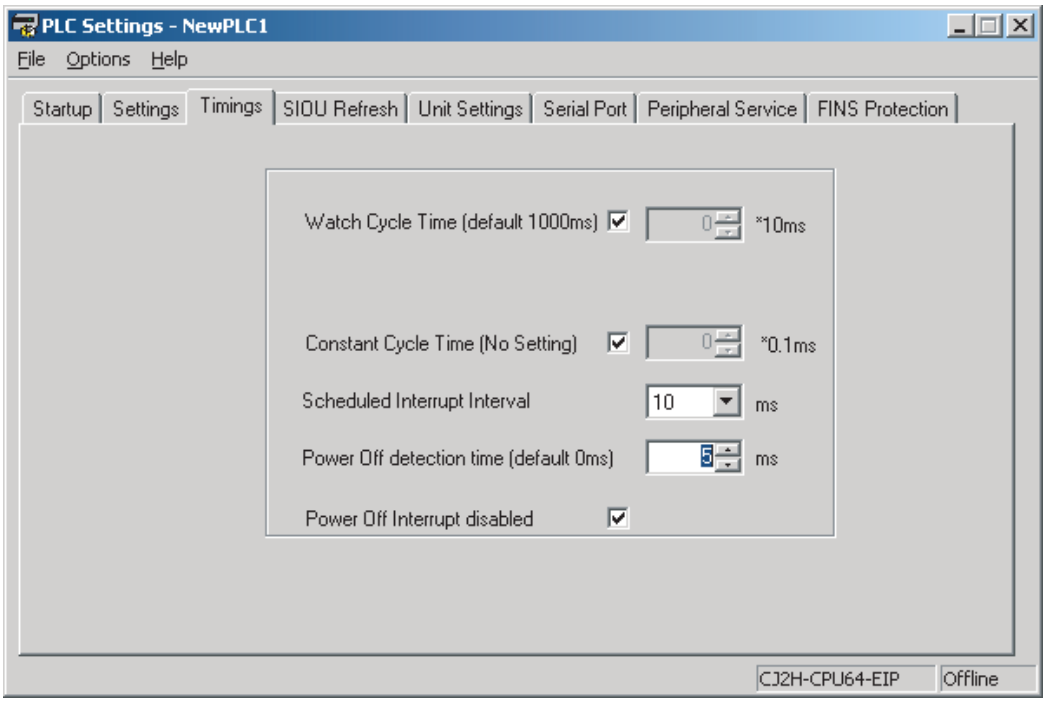
10-3-2 电源 OFF 检测延迟设定

默认情况下，将检测到 10ms 及以上 AC 电源中断 （DC 电源中断为 2ms），电源电压下降到最小额定值的 85% 以下 （DC 电源供应的 80%）后约 10 ～ 25ms （DC 电源为 2 ～ 5ms）将检测为电源中断。

可在 PLC 设置中进行设定以扩展此时间。在电源 OFF 中断任务允许执行时，将在确认电源中断时执行其设定，否则 CPU 将复位并且停止运行。

PLC 设置

使用 CX-Programmer 时，在定时 （Timings）标签页上的电源 OFF 检测时间 （Power Off detection time）区内进行设定。



附加信息

若不使用电源 OFF 中断任务，则将电源 OFF 检测时间设定为 10ms 或以下作为电源中断测量。若使用 CJ1W-PD022 电源供应单元，则无法设定延迟，因此请使用默认设定 0 ms。

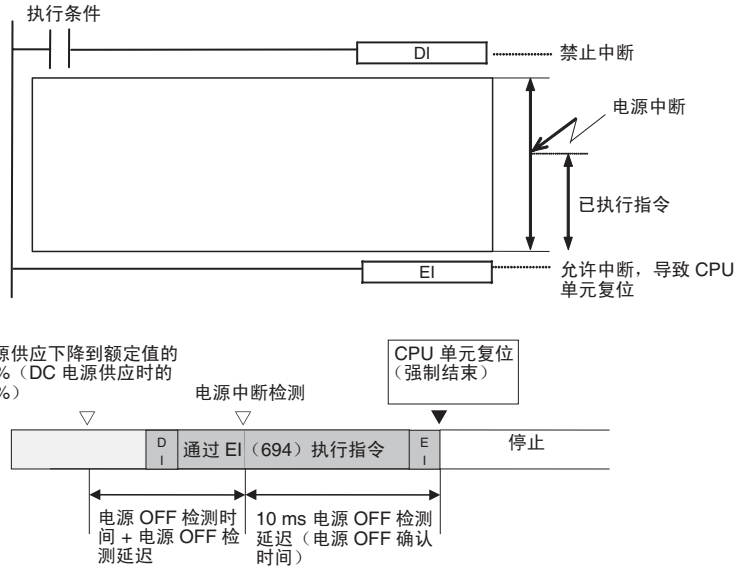
10-3-3 禁止电源 OFF 中断

程序区域可受保护以避免受电源 OFF 中断影响，因此即使电源中断，在 CPU 单元复位前它们仍可以运行。此功能可通过 DISABLE INTERRUPTS （DI （693））和 ENABLE INTERRUPTS （EI （694））指令实现。

此功能通过指令组实现，即：这套指令必须成组执行，因此在下一次电源接通时，不会用中间存储的数据开始执行操作。

步骤

1. 在 A530 ～ A5A5 hex 中设置电源 OFF 中断禁止设定，使禁止电源 OFF 中断功能有效。
2. 在 PLC 设置中使禁止电源 OFF 中断功能有效 （这是缺省设定）。
3. 在受保护的程序段前使用 DI （693）指令禁止中断，并在其后使用 EI （694）指令使允许中断。在在 DI （693）和 EI （694）指令之间执行期间，即使发生电源中断，也将在执行电源 OFF 中断之前完成 DI （693）和 EI （694）之间的所有指令。



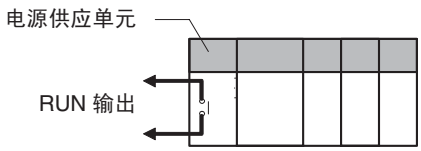
● 辅助区标志和字

名称	地址	表示
电源 OFF 中断禁止设定	A530	在执行 EI （694）之前，允许使用 DI （693）禁止电源 OFF 中断处理 （除执行电源 OFF 中断任务之外）。 A5A5 hex：允许使用 DI （693）禁止电源 OFF 中断处理。 任何其它值：不允许使用 DI （693）禁止电源 OFF 中断处理。

10-3-4 RUN 输出

RUN 输出

CJ1W-PA205R 电源单元配备了 RUN 输出。当 CPU 单元在 RUN 或 MONITOR 模式下操作时，该输出点为 ON （关闭）。



名称	操作
RUN 输出	MONITOR 或 RUN 模式：ON （关闭） PROGRAM 模式：OFF （打开）

使用 RUN 输出可建立一个外部安全电路，如：一个急停电路，以防止外部电源提供电源给输出单元，除非 PLC 在运行。

注　使用的电源供应单元无 RUN 输出时，可通过对常 ON 标志 （A1）（Always ON Flag （A1））进行编程，以作为一个输出单元的输出点执行条件，从而建立 （与 RUN 输出）等效的输出。

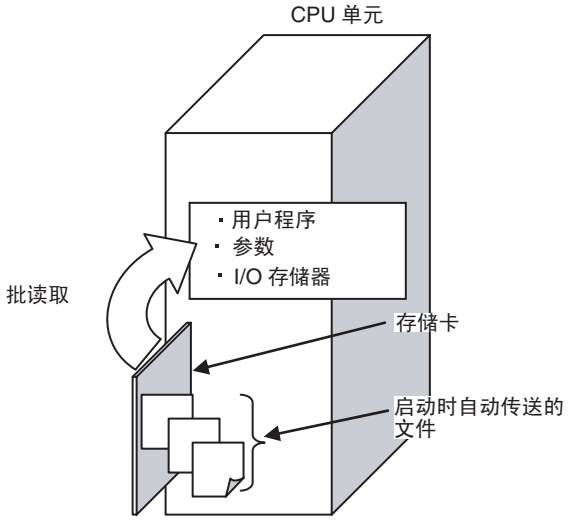
安全使用注意事项

如果在 PLC 的电源供应之前，接通输出单元的外部电源，则在 PLC 刚一接通的瞬时，输出单元将会出错。为了防止任何错误，加上一个外部电路，以在电源供应到 PLC 之前，防止接通输出单元的外部电源。创建一个上述的失效保险电路，以确保仅当 PLC 在 RUN 模式或 MONITOR 模式下操作时，电源才通过外部电源供电。

10-3-5 在启动时自动传输

概述

当电源接通时，使用初启时自动传输功能将存储卡中的用户程序，参数和 I/O 存储器数据读到 CPU 单元。可在 CX-Programmer 的存储卡窗口中创建启动时自动传输的文件，并保存在存储卡中。



此功能不可用于读取 EM 文件存储器。

附加信息

如果在不使用电池的状态下执行操作，则 I/O 存储器将不稳定。因此，可使用此功能以确保始终使用正确值。

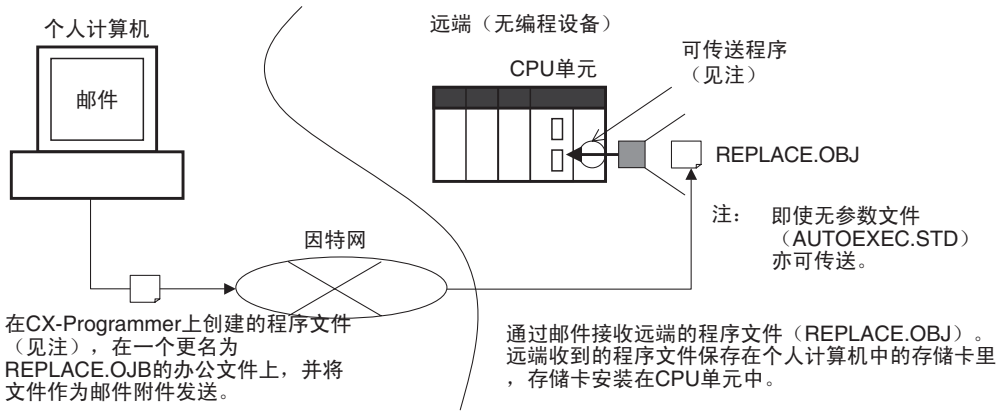
启动时自动传送的类型

可通过两种方式执行在启动时自动传送：带参数文件或不带参数文件传送 （AUTOEXEC.STD）。可根据所使用的两种传送类型创建文件名称。

启动时自动传送的类型	文件名称	应用
带参数文件的传送	AUTOEXEC 或 ATEXEC □□	覆写程序和网络设定
不带参数文件的传送	REPLACE 或 REPLC □□	覆写程序

● 不带参数文件的自动传送文件的应用示例

程序 / 网络符号 （变量）文件 （.OBJ）可在办公室中离线创建 （即，不使用实际设备），并传送至远程位置而无需参数文件 （.STD）。程序 / 网络符号 （变量）文件可保存在远端的存储卡中而无需使用编程设备，并且存储卡可用于在启动时自动将程序传送至 CPU 单元。

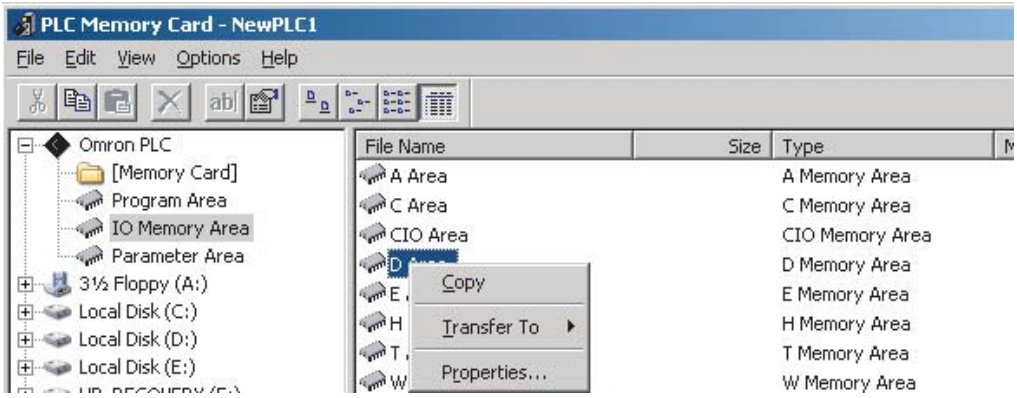


步骤

● 创建并传送启动时自动传送的文件

通过 CX-Programmer 创建用于启动时自动传送的文件，并将其传送到 CPU 单元中安装的存储卡。

- 程序区和参数区：
  右击并选择 “Transfer to-Memory Card”（传送到 - 存储卡）。输入启动时自动传送的文件名称，并进行创建和传送。
- I/O 存储器区：
  右击右侧面板中的 D 区和 E 区。输入启动时自动传送的文件名称，并进行创建和传送。

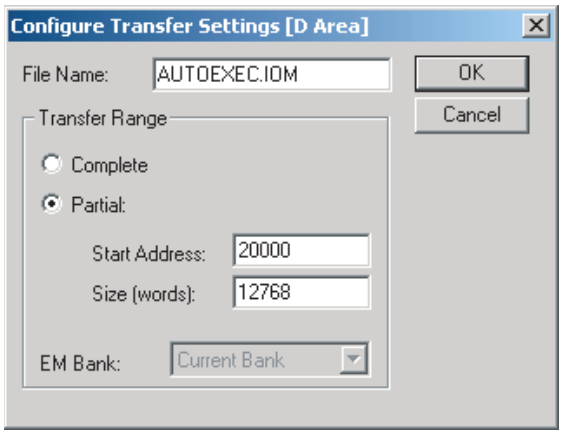


下述文件名为用于启动时自动传送。

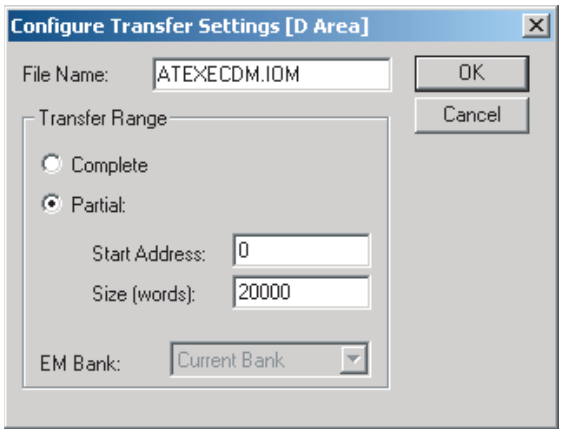
文件类型	内容	启动时自动传送 不带参数文件	是否必须	启动时自动传送 不带参数文件	是否必须
程序 / 网络符号（变量）文件	用户程序	AUTOEXEC.OBJ	是	REPLACE.OBJ	是
参数文件	• PLC 名称 • PLC 设置 • 注册的 I/O 表 • 路由表 • CPU 总线单元设置区	AUTOEXEC.STD	是	无	---
数据文件	分配 DM 区字给特殊 I/O 单元和 CPU 总线单元（D20000 ～ D32767）	AUTOEXEC.IOM	---	REPLACE.IOM	---
	DM 区 （D0 ～ D19999）	ATEXECDM.IOM	---	REPLACDM.IOM	---
	EM 区	AUTOEM □□ .IOM（□□：00 ～ 18）	---	RPLCEM □□ .IOM （□□ :00 ～ 18）	---

例如，对 DM 区指定下述起始地址和大小。

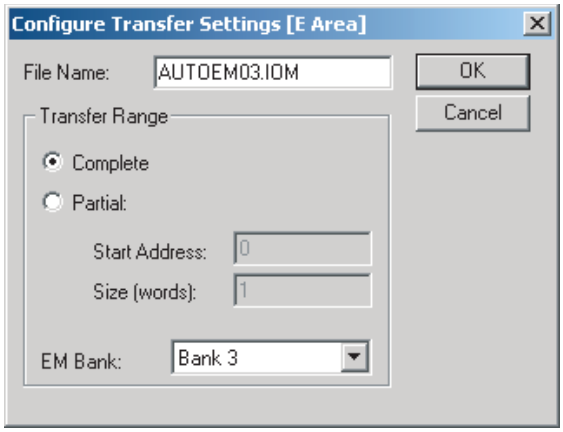
- 分配 DM 区字给特殊 I/O 单元和 CPU 总线单元。



- DM 区：



- EM 区：



● 执行启动时自动传送

1. 将 PLC 关闭。
2. 将 CPU 单元前的 DIP 开关位 2 置 ON。

   如果位 7 为 ON，则将运行简捷备份。请确保其为 OFF。
3. 使用已保存的所需文件，在 CPU 单元中安装存储卡。
4. 将 PLC 打开。

   将开始启动时自动传送，并且 BUSY 指示灯将闪烁黄色。但完成传送时 BUSY 指示灯将关闭。如果启动时自动传送出现故障，则将发生卡传送错误，且 CPU 单元将停止运行。

● CPU 单元前的 DIP 开关

开关位编号	名称	设定
2	启动时自动传送	ON：启动时执行自动传送 OFF：启动时不执行自动传送

● 辅助区标志和字

名称	地址	描述
卡传送错误标志 （致命错误）	A401.03	当从存储卡进行启动时自动传送失败时转为 ON。 CPU 单元将停止运行，且 CPU 单元前的 ERR/ALM 指示灯将亮灯。

启动时自动传送的数据文件

下述六种类型的数据文件为用于启动时自动传送。

启动时自动传送的数据文件		内容	范围	起始地址	尺寸
带参数文件进行启动时自动传送	不带参数文件进行启动时自动传送				
AUTOEXEC.IOM	REPLACE.IOM	分配 DM 区字给特殊 I/O 单元和 CPU 总线单元 （D20000 ～ D32767）	部分区	20000	12,768 字
ATEXECDM.IOM	REPLACDM.IOM	DM 区 （D00000 ～ D19999）	全部区	0	20,000 字
AUTOEM □□ .IOM （□□：00 ～ 18）	AUTOEM □□ .IOM （□□：00 ～ 18）	EM	全部区	---	---

当创建用于启动时自动传送的的数据文件时，请确保设定上述各区类型和起始地址。关于存储尺寸大小，请设定直到地址末尾的大小尺寸。

启动时，将从 D20000，D0 和 E □ _0 开始传送存储卡中数据文件上的所有数据。如果在整个区中没有足够的字，则将不会变更区中剩余的字。

带参数文件进行启动时自动传送

使用下述文件名称。下表右端栏内标志为 “是”的文件，为在执行启动时自动传送功能时必须存在于存储卡内的文件。

文件类型	文件名称	扩展名	内容	说明	是否必须
程序 / 网络符号（变量）文件	AUTOEXEC	.OBJ	用户程序及网络符号 （变量）	• 循环任务和中断任务中程序。 • 此文件必须在存储卡中，以执行启动时自动传送。 • 参数文件 （AUTOEXEC.STD）为必须在存储卡中，以执行启动时自动传送。	是
参数文件	AUTOEXEC	.STD	• PLC 名称 • PLC 设置 • I/O 表 • 路由表 • CPU 总线单元设置区	• CPU 单元的参数区数据。 • 电源接通时，参数将自动保存到 CPU 单元中的指定位置。 • 用户不需要在文件中单独指定数据。 • 当程序 / 网络符号（变量）文件为 AUTOEXEC.OBJ 时，此参数文件必须在存储卡中，以执行启动时自动传送。	是

文件类型	文件名称	扩展名	内容	说明	是否必须
数据文件	AUTOEXEC	.IOM	DM 区数据 （包含 D20000 开始的指定数据字数）	• 从 D20000 开始将 DM 区数据保存在 AUTOEXEC.IOM 文件中。 • 启动时，由 D20000 开始传送从指定文件开始的所有数据。如果在整个区中没有足够的字，则将不会变更区中剩余的字。 • 此文件并非在执行启动时自动传送中所必须的。	---
	ATEXECDM	.IOM	DM 区数据 （包含 D0 开始的指定数据字数）	• 从 D0 开始将 DM 区数据保存在 ATEXECDM.IOM 文件中。 • 启动时，由 D0 开始传送从指定文件开始的所有数据。如果在整个区中没有足够的字，则将不会变更区中剩余的字。 • 此文件并非在执行启动时自动传送中所必须的。 如果其内容与 AUTOEXEC.IOM 文件的内容重叠，则 ATEXECDM.IOM 的文件将具有优先权。	---
数据文件，连续	AUTOEM □□	.IOM	BANK No. □□ 的 DM 区数据（包含 E □□ _0 开始的指定数据字数）	• 从 E □ _0 开始将 BANK No. □的 DM 区数据保存在 AUTOEM □□ .IOM 文件中。 （方框表示 BANK No. 0 ～ C。BANK 的最大组编号为根据 CPU 单元的型号决定）。 • 启动时，由E□_0开始传送从指定文件开始的所有数据。如果在整个区中没有足够的字，则将不会变更区中剩余的字。 • 此文件并非在执行启动时自动传送中所必须的。	---

如果 AUTOEXEC.IOM 和 ATEXECDM.IOM 文件内容重叠，则 ATEXECDM.IOM 文件具有优先写入权利。（因为文件的载入顺序为：AUTOEXEC.IOM 在前， ATEXECDM.IOM 在后）。

附加信息

可结合使用启动时自动传送和通过特殊辅助区控制位替换整个程序。通过启动时自动传送可实现以一个程序替换另一个程序，并在操作期间使用辅助区控制位替换程序。

不带参数文件进行启动时自动传送

使用下述文件名称。下表右端栏内标志为“是”的文件，在执行启动时自动传送功能时必须存在于存储卡内的文件。

文件类型	文件名称	扩展名	内容	说明	是否必须
程序 / 网络符号（变量）文件	REPLACE	.OBJ	用户程序	• 内容与 AUTOEXEC.OBJ 相同。 • 即使参数文件（AUTOEXEC.STD）不存在于存储卡，启动时也将传送程序 / 网络符号 （变量）文件 。	是
参数文件	不需要	---	---	• 无论在存储卡中使用何种文件名称，都不传送此参数文件。	---
数据文件	REPLACE	.IOM	DM 区数据 （包含 D20000 开始的指定数据字数）	• 内容与 AUTOEXEC.IOM 相同。 • 如果程序 / 网络符号（变量）文件（REPLACE.OBJ）存在于存储卡，则将在启动时传送此文件中的数据。	---
	REPLCDM	.IOM	DM 区数据 （包含 D0 开始的指定数据字数）	• 内容与 ATEXECDM.IOM 相同 • 如果程序 / 网络符号（变量）文件（REPLACE.OBJ）存在于存储卡，则将在启动时传送此文件中的数据。	---
	RPLCEM □□	.IOM	BANK No. □□ 的 DM 区数据（包含 E □□ _0 开始的指定数据字数）	• 内容与 AUTOEM □□ .IOM 相同 • 如果程序 / 网络符号（变量）文件（REPLACE.OBJ）存在于存储卡，则将在启动时传送此文件中的数据。	---

附加信息

通过从分配给特殊 I/O 单元的 DM 区字 （D20000 ～ D29599）和分配给 CPU 总线单元的 DM 区字 （D30000 ～ D31599）保存数据到存储卡的 AUTOEXEC.IOM 文件，启动时可从存储卡读取特殊 I/O 单元和 CPU 总线单元的设置数据。可通过此方式使用存储卡来管理各设备的特殊 I/O 单元和 CPU 总线单元的系统数据库。

启动时自动传送所需大致时间

用户程序和网络符号 （变量）的大小	自动传送所需的时间（从接通电源到开始运行）
用户程序：120K 步	9 s
用户程序：120K 步，网络符号 （变量）：20,000	59 s

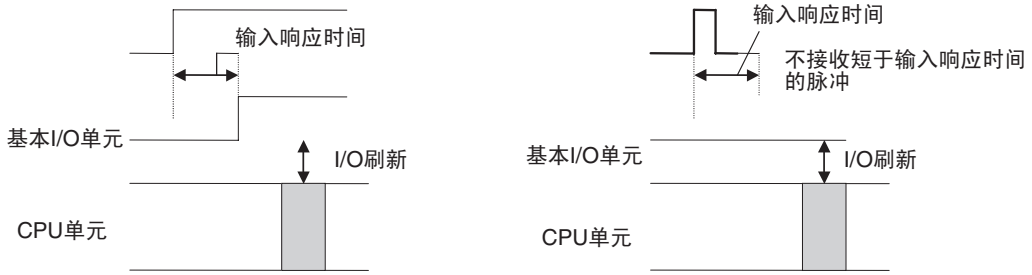
10-4 单元管理功能

10-4-1 基本 I/O 单元管理

● 基本 I/O 单元的输入响应时间设定

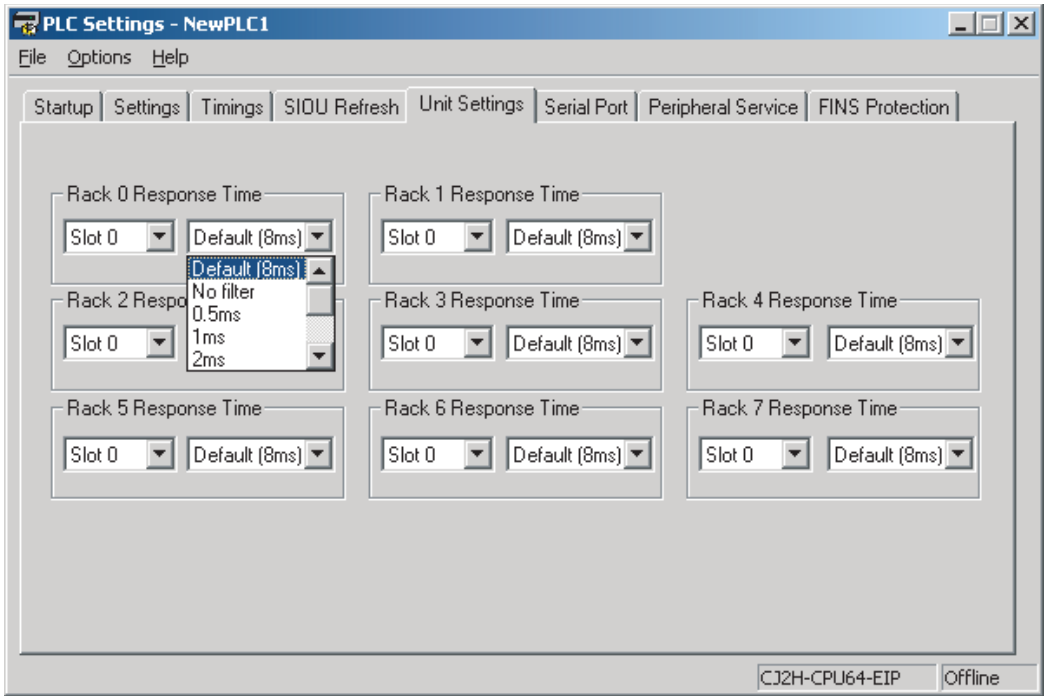
可通过机架和槽位编号设定基本 I/O 单元的输入响应时间。可增加输入响应时间以减少振颤和噪声。减少输入响应时间（但需保持脉冲宽度长于循环时间）以允许接收较短脉冲输入。

注　可通过高速输入单元输入短于循环时间的脉冲。详情请参见 *10-2-4 高速输入* 。



PLC 设置

使用 CX-Programmer 时，在单元设定标签上对机架 0 ～ 7 的响应时间进行设定。



辅助区标志和字

名称	地址	描述
基本 I/O 单元中 I/O 响应时间	A220.00 ～ A259.15	包含 CJ 系列基本 I/O 单元的当前 I/O 响应时间。

● 基本 I/O 单元错误管理

下表为与基本 I/O 单元相关的错误信息。

辅助区标志和字

名称	地址	描述
基本 I/O 单元错误标志（非致命错误）	A402.12	在基本 I/O 单元中发生错误时 ON。
基本 I/O 单元错误，机架编号	A408.08 ～ A408.15	当在基本 I/O 单元中发生错误时，保存错误发生处的二进制机架编号 （A402.12 为 ON 时）。
基本 I/O 单元错误，槽位编号	A408.00 ～ A408.07	当在基本 I/O 单元中发生错误时，保存错误发生处的二进制槽位编号 （A402.12 为 ON 时）。
基本 I/O 单元状态区 （机架 0 槽位 0 ～机架 3 槽位 9）	A050.00 ～ A069.15	表示基本 I/O 单元 （从机架 0 上的槽位 0 ～机架 3 上的槽位 9）的报警状态 （负载短路保护）
中断输入单元位置错误标志	A405.08	未将中断输入单元连接在紧靠机架上 CPU 单元侧的四个位置中 （槽位 0 ～ 3）时 ON。

10-4-2 CPU 总线单元标志 / 位

● 重启和初始化 CPU 总线单元

使用 CPU 总线单元，在不关闭 PLC 的情况下，通过将下述重启位变为 ON，可变更分配到 CPU 单元的 DM 区字的初始设定。通常，可从用户程序执行此操作。

辅助区标志和字

名称	地址	描述
CPU 总线单元重启位	A501.00 ～ A501.15	这些位对应于 CPU 总线单元 00 ～ 15 （单元编号 0 ～ F）。将一个位从 OFF 转变为 ON ，可以重新启动相应的单元。
CPU 总线单元初始化标志	A302.00 ～ A302.15	这些位对应于 CPU 总线单元 00 ～ 15 （单元编号 0 ～ F）。当电源接通或单元的重启位 （A501.00 ～ A501.15 ）转为 ON，相应的单元初始化时，它的标志位将为 ON。

● CPU 总线单元错误

下表为与 CPU 总线单元相关的错误信息。

辅助区标志和字

名称	地址	描述
CPU 总线单元编号重复标志	A410.00 ～ A410.15	当 CPU 单元的单元编号重复时，重复错误标志（A401.13）和在 A410 里对应的标志位转为 ON。 位 00 ～ 15 对应于单元号码 0 ～ F。
CPU 总线单元设置错误标志（非致命错误）	A402.03	当安装的 CPU 总线单元与在 I/O 表内登录的 CPU 总线单元不相符合时该标志为 ON。
CPU 总线单元设置错误，单元编号标志	A427.00 ～ A427.15	当一个 CPU 总线单元设置出现错误时，A402.03 和 A427 中相对应的标志转为 ON。
CPU 总线单元错误标志（非致命错误）	A402.07	当 CPU 单元和一个 CPU 总线单元间交换数据时发生错误（包括 CPU 总线单元自身错误）的情况下，该标志为 ON 。
CPU 总线单元错误，单元编号标志	A417.00 ～ A417.15	当 CPU 单元和一个 CPU 总线单元间交换数据时发生错误的情况， CPU 总线单元错误标志 （A402.07）和在 A417 中对应的标志转为 ON。 位 00 ～ 15 对应于单元号码 0 ～ F。

10-4-3 特殊 I/O 单元标志 / 位

● 重启和初始化特殊 I/O 单元

对特殊 I/O 单元，在不关闭 PLC 的情况下，通过将下述重启位变为 ON，可变更分配到 CPU 单元的 DM 区字的初始设定。通常，可从用户程序执行此操作。

辅助区标志和字

名称	地址	描述
特殊 I/O 单元重启位	A502.00 ～ A507.15	位 A502.00 ～ A507.15 对应于特殊 I/O 单元 0 ～ 95 。将这些位置 ON 可使相对应的特殊 I/O 单元重新启动。
特殊 I/O 单元初始化标志	A330.00 ～ A335.15	这些标志对应于特殊 I/O 单元 0 ～ 95，当相应单元的电源接通或单元的重新启动位 （A502.00 ～ A507.15）转 ON 后，进行初始化时，其标志将为 ON。位 A330.00 ～ A335.15 对应于单元编号 0 ～ 95。

● 特殊 I/O 单元错误

下表为与特殊 I/O 单元相关的错误信息。

辅助区标志和字

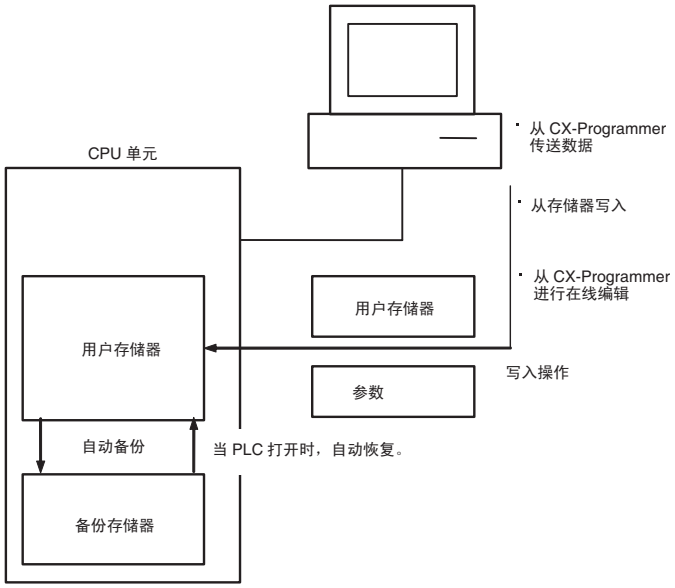
名称	地址	描述
特殊 I/O 单元编号重复标志	A411.00 ～ A416.15	当一个特殊 I/O 单元的单元号重复时，重复错误标志（A401.13）和 A411 ～ A416 的相应标志变为 ON 。 位 A411.00 ～ A416.15 对应于单元编号 0 ～ 95。
特殊 I/O 单元设置错误标志（非致命错误）	A402.02	当安装的特殊 I/O 单元与在 I/O 表中登录的特殊 I/O 单元不相符合时，该标志为 ON。
特殊 I/O 单元设置错误，单元编号标志	A428.00 ～ A433.15	当出现一个特殊 I/O 单元设置错误时，A402.02 和这些字中相应的标志变为 ON。位 A428.00 ～ A433.15 对应于单元编号 0 ～ 95。
特殊 I/O 单元错误标志（非致命错误）	A402.06	在 CPU 单元和一个特殊 I/O 单元交换数据中发生一个错误时 （包括特殊 I/O 单元自身发生错误时），该标志为 ON 。
特殊 I/O 单元错误，单元编号标志	A418.00 ～ A423.15	在 CPU 单元和一个特殊 I/O 单元交换数据中发生一个错误时。特殊 I/O 单元错误标志 （A402.06）和这些字中相应的标志变为 ON 。

10-5 存储器管理功能

10-5-1 自动备份

CPU 单元中写入或变更用户程序或参数时，都将自动备份到备份存储器。

- 下述数据为自动备份：用户程序，参数（包括 PLC 名称，PLC 设置，I/O 表，路由表以及 CPU 总线单元数据，如数据链接表）。
- CPU 单元中写入 / 变更用户程序或参数时，都将自动对数据进行备份，包括来自 CX-Programmer 数据传送操作、在线编辑、以及来自存储卡或 EM 文件存储器的数据传送等
- 启动时，自动写入到备份存储器的用户程序和参数数据将被传送到 CPU 单元中的用户存储器。



- 当数据正写入到备份存储器时，CPU 单元前的 BKUP 指示灯将亮灯。从 CX-Programmer 或文件存储器传送数据，或者执行在线编辑后，请不要在备份操作完成前切断 CPU 单元的电源（直到 BKUP 指示灯熄灭）。否则数据将丢失。
- 仅对在线编辑且 CPU 单元中使用电池的情况下，才可实现即使在备份操作完成前切断电源时，CPU 单元仍将以之前的状态进行重启 （即 BKUP 指示灯为亮灯）。请注意下述各项：

  1. 启动 CPU 单元需要较长时间。

  2. 在长时间未接通 CPU 单元电源的情况下，即使在 CPU 单元中使用电池，也请在切断电源前确认备份操作已完成。

- 备份数据所需要的时间 （BKUP 指示灯亮灯的时间）为根据用户程序的大小决定，如下表所示。

用户程序大小	备份处理时间	
	MONITOR 模式	PROGRAM 模式
100 K 步	20 s （循环时间 3 ms）	3 s
250 K 步	47 s （循环时间 7 ms）	6 s
400 K 步	75 s （循环时间 11 ms）	10 s

注　接通 CPU 单元电源时 BKUP 指示灯将亮灯。

正确使用注意事项

- 当写入用户程序和参数数据到CPU单元时，将他们自动备份到闪存。I/O存储器（包括DM、EM、保持和辅助区）不写入闪存。发生电源中断时可使用电池保持 DM、EM 和保持区。如果电池发生故障，则发生电源中断后这些区中的数据将不再准确。如果 DM、 EM、保持和辅助区的内容用于控制外部输出，则请注意当电池错误标志 （A402.04）为 ON 时，避免进行不适当的输出。
- 如果在 CPU 单元备份存储器中检测到错误，则将发生备份存储器错误 （非致命）。如果发生此错误，则数据将不会在再次接通电源时恢复，并且将发生存储器错误。因此，推荐在关闭电源前对数据进行保存。

附加信息

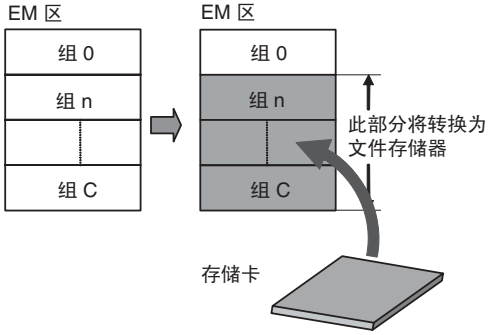
当通过 CX-Programmer 传送备份数据，而非使用普通的数据传送（PLC - Transfer（PLC- 传送）），则将通过 CX-Programmer 在存储器备份状态窗口中显示备份状态。为使此显示有效，请选取 PLC 属性中的显示备份状态对话框，并在显示菜下选择该窗口。普通的传送操作下，将在程序和其他数据传送状态后，将备份状态显示在传送窗口中。

● 辅助区标志

名称	地址	表示
备份存储器错误标志	A315.15	备份存储器故障时置 ON

10-5-2 EM 文件存储器功能

在 CPU 单元中可使用 EM 区替代存储卡保存文件。将指定起始组后的组用作文件存储器。关于转换 EM 区为文件存储器的相关内容 ，请参见 7-1-2 初始化文件存储器。



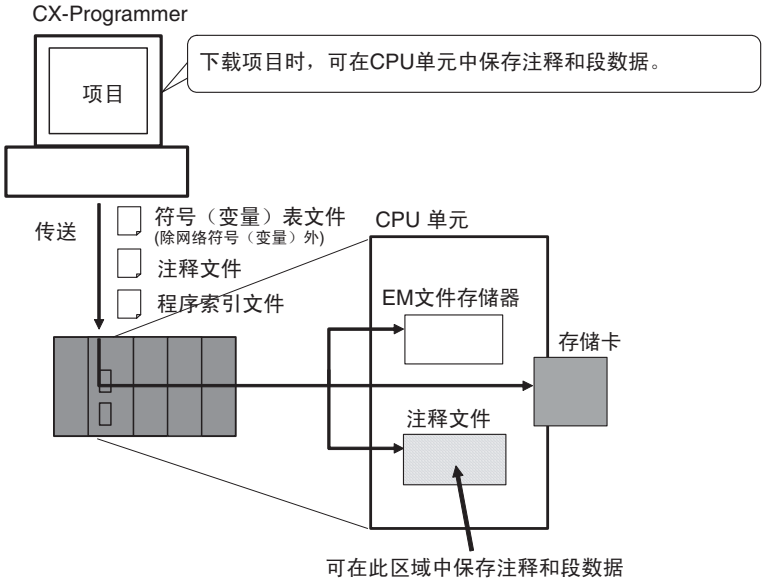
1. 通过CX-Programmer在EM区域设定中指定起始组。
2. 当通过 CX-Programmer 传送程序时，可使用 EM 文件存储器。

可选择将部分 EM 区转换为文件存储器或跟踪存储器。如果转换为文件存储器，则可将 EM 区的转换部分用作存储卡。

10-5-3 注释存储器

在 CPU 单元的内置备份存储器中具有注释存储器。即使存储卡或 EM 文件存储器为无效，也可从注释存储器对下述注释 / 段数据进行保存或读取。

- 符号（变量）表文件（包括 CX-Programmer 符号（变量）以及 I/O 注释，但不包括网络符号（变量））。
- 注释文件 （CX-Programmer 条注释及其他注释）
- 程序索引文件 （CX-Programmer 段名称，段注释以及程序注释）。

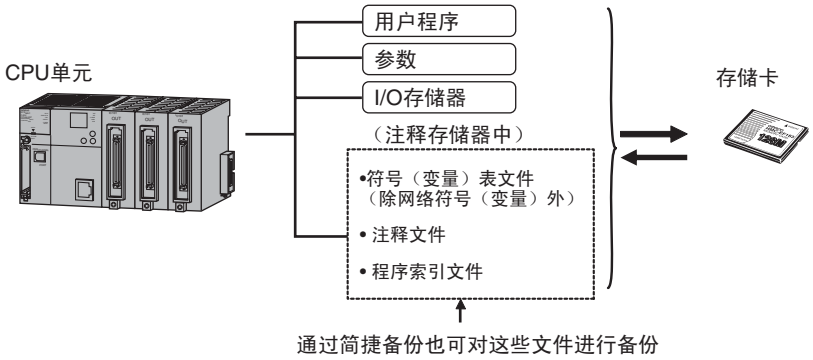


下载项目时，可选择下述存储位置作为注释数据和段数据的传送目的地。

- 注释存储器 （CPU 单元的备份闪存中）
- 存储卡
- EM 文件存储器

当执行简捷备份操作时，可将下述保存在注释存储器中的文件备份到存储卡，或从存储卡将文件恢复到注释存储器。

- 符号（变量）表文件（包括 CX-Programmer 符号（变量）以及 I/O 注释，但不包括网络符号（变量））。
- 注释文件 （CX-Programmer 条注释及其他注释）
- 程序索引文件 （CX-Programmer 段名称，段注释以及程序注释）。

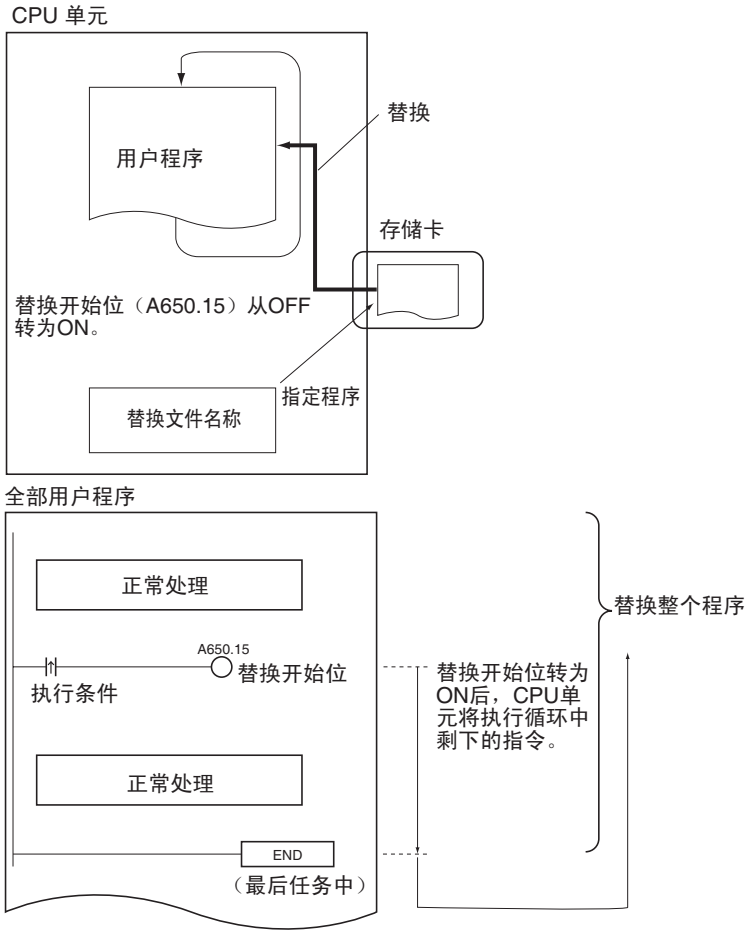


如果发生错误或添加一个相同规格的 CPU 单元时，可在不使用 CX-Programmer 的情况下，对 CPU 单元中的所有数据 （包括 I/O 注释）进行备份 / 恢复。

10-5-4 操作中替换全部程序

概述

在 RUN 或 MONITOR 模式下，通过将替换开始位 （A650.15）置 ON，可在运行中替换整个任务程序。使用事先在辅助区中指定的文件名称，可从存储卡中读取程序 / 网络符号 （变量）文件 （.OBJ），并且其将在当前循环的末尾对程序进行替换。



通过从 CX-Programmer 将替换开始位转为 ON，也可实现在程序执行停止时 （PROGRAM 模式）对其进行替换。

- 不可从 EM 文件区读取替换程序文件。
- 可在程序中的任何位置 （程序地址），将替换开始位 （A650.15）置 ON。
- 当替换开始位从 OFF 转为 ON 后， CPU 单元将执行循环中剩下的指令。
- 程序替换中将不会执行程序。当程序替换完成后，程序将再次开始（与将 CPU 单元从 PROGRAM 模式切换为 RUN/MONITOR 时相同）。
- 当替换开始位从OFF转为ON的情况下，将在循环的末尾替换程序。（即，在程序的最后任务中执行END（001）后）

程序替换时的 CPU 操作

在程序替换时， CPU 单元的操作将如下所述：

- 程序执行：停止
- 循环时间监控：未监控

● 程序替换中及程序替换后操作继续

- 当 IOM 保持位 （A500.12）为 ON 时，将保持下述存储器区的数据：CIO 区，工作区域（W），定时器完成标志 （T），索引寄存器 （IR），数据寄存器 （DR）以及当前 EM 组编号。

注　　程序替换时，将清除定时器当前值。

- 当传送程序时，如果IOM保持位为ON，则程序替换前的正在输出的负载将在替换后继续输出。请确保在程序替换后，外部负载的操作正确。
- 如果强制状态保持位 （A500.13）为 ON，则强制置位 / 复位的状态在程序替换前后将始终为保持。
- 屏蔽中断。
- 如果正执行数据跟踪，则其将停止。
- 无论 IOM 保持位为 ON 或 OFF，微分标志都将被初始化。

● 程序替换后操作

- 循环任务的状态为根据运行 - 开始属性而定。（其状态为与将 PLC 从 PROGRAM 模式切换到 RUN/MONITOR 模式时的状态相同）。
- 继续程序执行后的第一个循环时第一循环标志(A200.11)将为ON。（其状态为与将PLC从PROGRAM模式切换到 RUN/MONITOR 模式时的状态相同）。

步骤

1. 在 CPU 单元中安装包含程序文件的存储卡。

2. 在辅助区中设置程序文件名 （A654 ～ A657）和程序密码 （A651），并将替换开始位（A650.15）置 ON。

附加信息

如果需要在程序替换前后保持 I/O 存储器数据的状态，请将 IOM 保持位 （A500.12）置 ON。

如果需要在程序替换前后保持强制置位 / 复位的状态，请将强制状态保持位 （A500.13）置 ON。

安全使用注意事项

如果在替换程序前， IOM 保持位 （A500.12）为 ON，则程序替换后， I/O 存储器中的位状态将为保持。请确保在使用相同 I/O 存储器数据时外部负载的操作正确。

如果在替换程序前，强制状态保持位 （A500.13）为 ON，则程序替换后， I/O 存储器中的强制置位 / 复位状态将为保持。请确保在使用相同强制置位 / 复位状态时外部负载的操作正确。

替换文件

从存储卡读取指定的程序 / 网络符号 （变量）文件 。

文件	File 名称和扩展名	在辅助区中将替换开始为（A650.15）置 ON	指定替换文件名 （********）
程序 / 网络符号 （变量）文件	********.OBJ	替换 CPU 单元的用户程序	程序替换前，将替换程序文件名写入 A654 ～ A657。

程序替换条件

下述为用于运行期间程序替换的条件。

- 在程序文件名称字（A654～A657）中指定的程序/网络符号（变量）文件存在于存储卡的根目录下。
- 通过 CPU 单元检测到了存储卡 （A343.15 必须为 ON）。
- 发生非致命错误。
- 没有文件存储器操作正在执行 （A343.13 必须为 OFF）。
- 未将数据写入程序区。
- 访问权有效。（例如，未从 CX-Programmer 传送数据到 PLC）。

注　可在任何运行模式下传送程序。

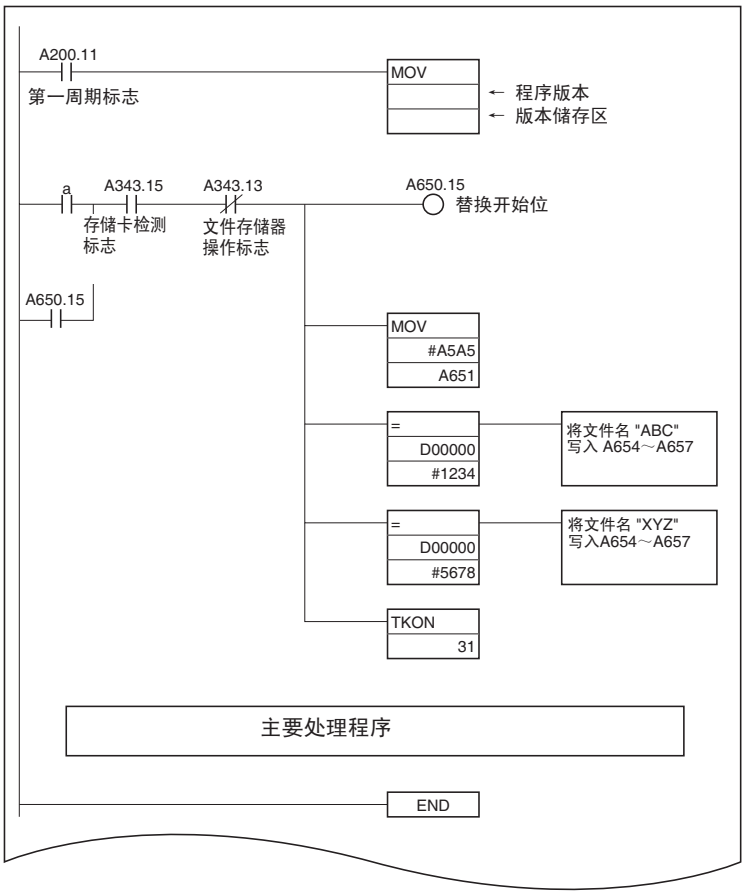
程序示例

● 程序示例 1

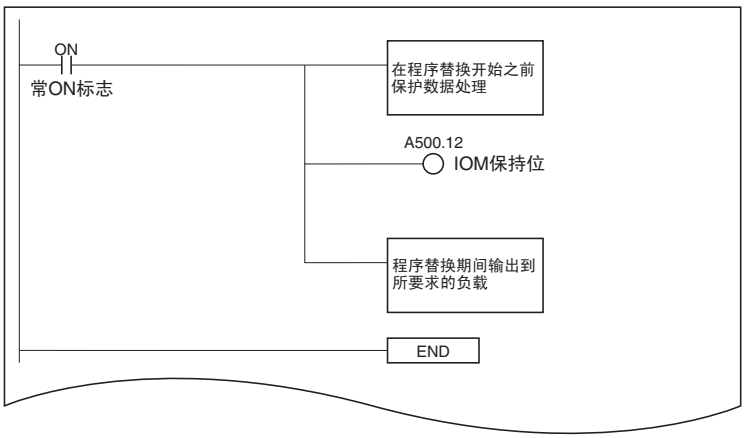
将程序 / 网络符号 （变量）文件 ABC.OBJ 和 XYZ.OBJ 存储在存储卡中，根据 D0 的值选择一个程序或选择其它。当要选择 ABC.OBJ 时，将 D0 设定为 #1234 ；或当要选定 XYZ.OBJ 时，将 D0 设定为 #5678。

在程序替换或 IOM 保持位处理之前，开始和执行另一个任务，以完成所要求的处理。

主要任务 （循环任务0）

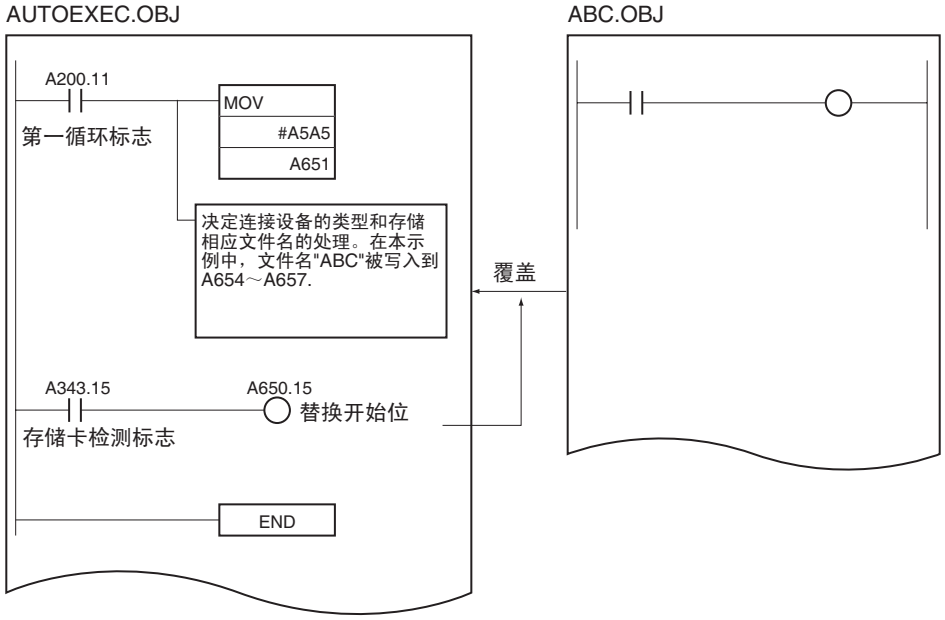


程序替换期间保护数据的任务
（循环任务31，启动时处于待机状态）



● 程序示例 2

将用于多个设备的程序 / 网络符号 （变量）文件以及用于启动时自动传送的程序 / 网络符号 （变量）文件 （AUTOEXEC.OBJ 或 REPLACE.OBJ）保存到存储卡中。当 PLC 打开时，读取启动时自动传送的文件，随后，不同设备的程序被一个程序文件替换。



辅助区标志和字

名称	地址	描述
文件存储器操作标志	A343.13	在以下的任何执行情况下为 ON，当执行完成时变为 OFF。 • 存储卡检测 • 通过 CMND（490）对本地 CPU 单元发送一 FINS 命令。 • 执行 FREAD （700）或 FWRIT （701）。 • 通过辅助区控制位 （A650.15），对程序进行覆写 • 简捷备份操作
存储卡检测标志	A343.15	当一个存储卡被检测到时该位为 ON。 当没有检测到一个存储卡时该标志为 OFF。
IOM 保持位	A500.12	当接通电源或将 CPU 单元的运行模式在 PROGRAM 模式与 RUN/MONITOR 模式间变更时，将此位置 ON 以保持 I/O 存储器的状态。 ON：变更运行模式时，保持 I/O 存储器的状态。 OFF：变更运行模式时，清除 I/O 存储器的状态。
强制状态保持位	A500.13	当接通电源或将 CPU 单元的运行模式在 PROGRAM 模式与 RUN/MONITOR 模式间变更时，将此位置 ON 以保持已强制置位 / 复位的位状态。

名称	地址	描述
程序替代结束代码	A650.00 ～ A650.07	• 正常结束 （A650.14 为 OFF）： 01 Hex：程序 / 网络符号 （变量）文件 （.OBJ）替换程序 • 错误结束 （A650.14 为 ON）： 00 hex：致命错误 01 hex：存储器错误 11 hex：写保护 12 hex：A651 中程序密码错误 21 hex：未安装存储卡 22 hex：指定文件不存在 23 hex：指定文件过大 （存储器错误） 31 hex：正执行了下述操作之一： - 文件存储器操作 - 正写入用户程序 - 变更运行模式
替换错误标志	A650.14	当 A650.15 已从 OFF 转为 ON 要替代程序时，如有错误发生则该标志为 ON。当 A650.15 再次从 OFF 转为 ON 时，此标志将变为 OFF。
替换开始位	A650.15	通过设定程序密码 （A651）为 A5A5 hex，使此位有效时，则当此位从 OFF 转为 ON 时，开始替换程序。在程序替换期间，不要再次将此位从 OFF 转为 ON。 接通电源或当程序替换完成时 （正常或无错误），此位将自动转为 OFF。 可从 CX-Programmer、PT 或上位计算机来读取此位的状态，以确认程序替换是否完成。
程序密码	A651	在此字中输入密码，以使程序替换生效。 A5A5 hex：替换开始位 （A650.15）有效 其他值：替换开始位 （A650.15）无效 接通电源或当程序替换完成时 （正常或无错误），此位将自动转为 OFF。
程序文件名称	A654 ～ A657	在启动程序替换前，将以 ASCII 格式在此字中保存程序文件名称，最多可用 8 个字符指定文件名，自动添加扩展名 .OBJ 。 文件名以下列顺序保存：A654 ～ A657 （即从低字到高字）且从高字节到低字节。如果一个文件名小于 8 个字符，在最低剩余字节和最高剩余字节用空格 （20 hex）填充。在文件名中不能使用 NULL 字符和空格字符。 例如：文件名称为 ABC.OBJ： 15 0 A654 41 42 A655 43 20 A656 20 20 A657 20 20

10-6 安全功能

下述内容为应用于 CJ 系列用户程序的保护功能。

10-6-1 使用 DIP 开关的写保护功能

可对用户程序以及参数区 （PLC 设置或 I/O 表）中的数据设置写保护功能，以防止来自 CX-Programmer 的覆写。此功能可用于保护数据不受意外写入操作的破坏。

设置写保护时，将 CPU 单元的 DIP 开关位 1 置 ON。使用此功能时，仍可通过 CX-Programmer 进行读取或显示。

● CPU 单元前的 DIP 开关

开关位编号	名称	设定
1	用户程序存储器 写保护	ON：保护 ON OFF：保护 OFF

10-6-2 使用密码的读保护

可通过设定密码进行读保护。使用此功能时，将无法显示或编辑受保护的用户程序、指定任务以及功能块，除非在 CX-Programmer 中输入正确的密码。

保护类型

可使用密码进行下述类型的读保护。

● UM 读保护

禁止从 CPU 单元对用户程序进行读取。

● 任务读保护

禁止在 CX-Programmer 中显示任务。虽然可从 CPU 单元读取用户程序，但受保护任务将无法显示在 CX-Programmer 上。

附加信息

- 即使程序中的单个任务设定了读保护，依然可将整个程序传送到其他 CPU 单元。对于受密码保护的任务，其读保护功能仍保持有效。
- 当使用 CX-Programmer 对计算机存储器与 CPU 单元中的用户程序进行比较时，也将比较密码保护任务。

● 功能块保护

可将下述保护功能应用到指定的块。

- 写 / 显示保护
  将禁止在 CX-Programmer 中显示功能块，因此将无法对其进行变更。

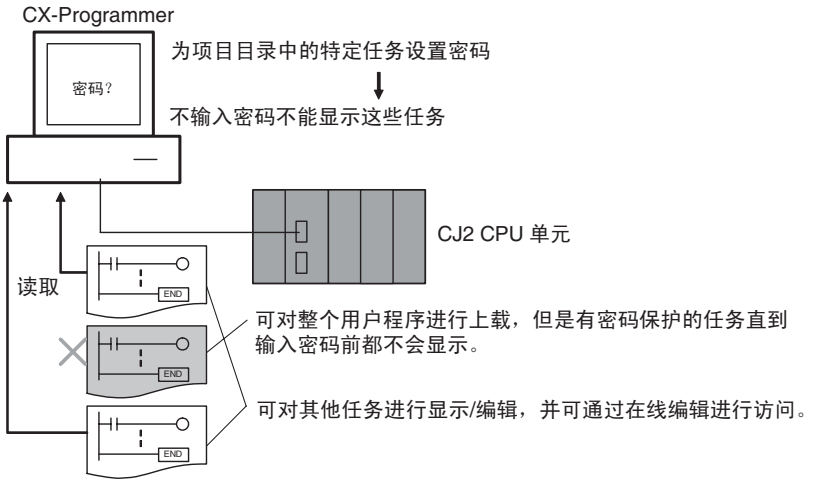
- 写保护
  将允许在 CX-Programmer 中显示功能块，但禁止对其进行变更。

附加信息

即使应用了任务读保护，在包含功能块的用户程序中，仍可读取出功能块的定义部分。如果要对功能块进行读保护，则请使用功能块保护功能。

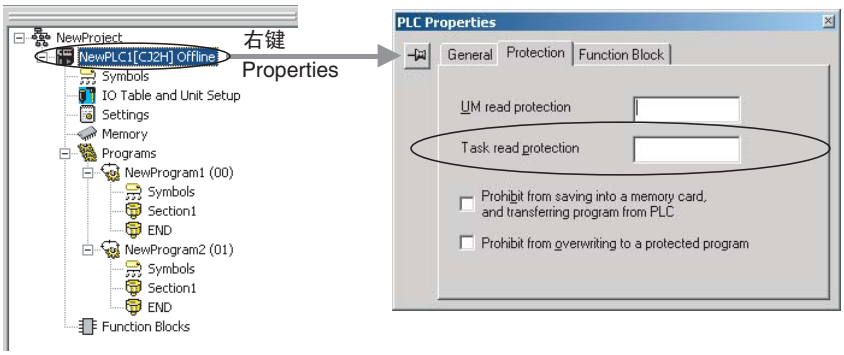
附加信息

如果已经设置了 UM 读保护，则将无法设置任务读保护。但是，可以在已经设置了任务读保护后再设置 UM 读保护。



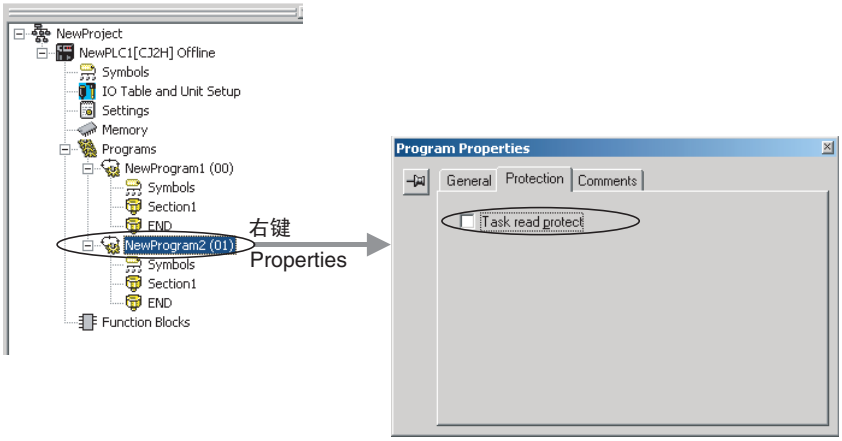
操作步骤

1. 显示 PLC Properties （PLC 属性）窗口的 Protection （保护）选项并在 Task read protection （任务读保护）框中注册一个密码。

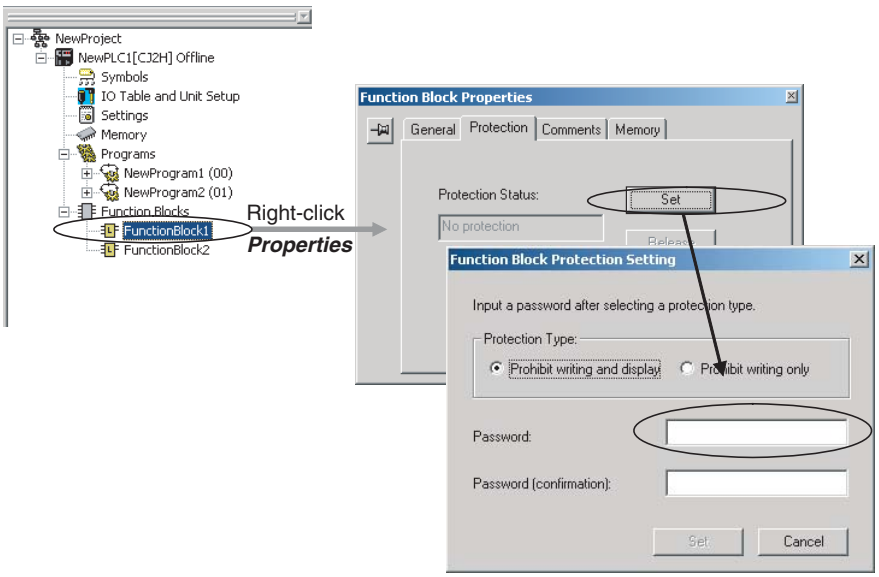


- UM read protection （UM 读保护）区
  输入 UM 读保护的密码。
- Task read protection （任务读保护）区
  输入任务读保护的密码。

2. 选择将被密码保护的任务，并在 “Program Properties ”（程序属性）选项页中选择 “Task read protection ”（任务读保护）选项。



3. 要将读保护应用到功能块（FB）定义，请选择要进行保护的功能块，在“Function block properties”（功能块属性）中点击 “Set”（设置）按钮，并在 “Function Block Protections Setting”区 （功能块保护设定）区中输入密码。



4. 在线连接并执行下面的步骤 a 或 b。

(a) 传送程序并设置密码保护：
选择 “PLC-Transfer-To PLC” （PLC- 传送至 -PLC）来传送程序。步骤 2 中注册的任务将受到密码保护。

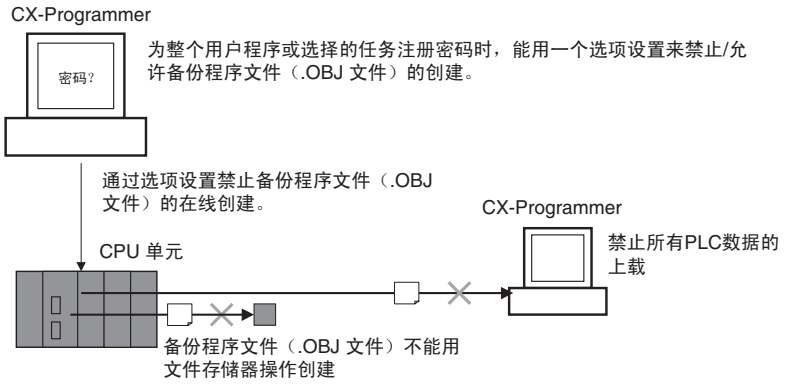
(b) 不传送程序，设置密码保护：
选择 “PLC-Protection-Set Password ”（PLC- 保护 - 设置密码）并点击 OK 按钮。步骤 2 中注册的任务将受到密码保护。

允许 / 禁止保存到存储卡

● 概述

当从 CX-Programmer 对整个程序或单个任务进行读保护时，可以设置一个选项来允许或禁止 .OBJ 程序 / 网络符号 （变量）文件的创建 / 备份。如果用这个设置禁止了程序 / 网络符号 （变量）文件的创建 / 备份，就不能用文件存储器操作来创建程序 / 网络符号 （变量）文件 （.OBJ 文件）。（这个设置既禁止在线传送到存储器卡 /EM 文件存储器，也禁止上载到 CX-Programmer 的 PLC 数据的离线存储）。

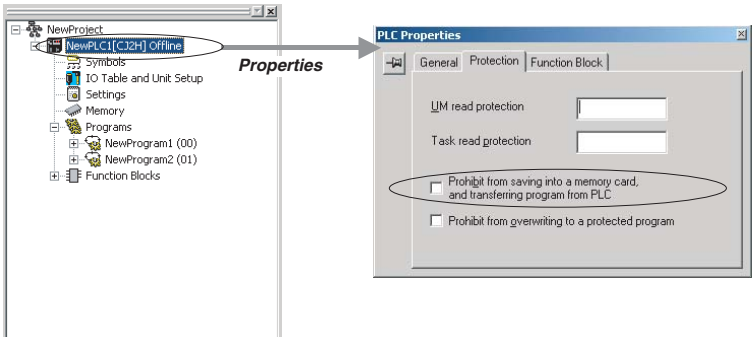
禁止创建文件存储器程序文件有助于防止对用户程序的非法拷贝。



- 当程序 / 网络符号（变量）文的创建被禁止时，仍能进行简捷备份操作，但是不能创建备份程序 / 网络符号 （变量）文件 （BACKUP.OBJ）。
- 当程序读保护无效时，程序能被拷贝。
- 除非程序被传送到了 CPU 单元，否则允许 / 禁止创建程序 / 网络符号（变量）文件的设置将不发生作用。变更此设置后一定要传送程序。

● 操作步骤

1. 在 “UM read protection”（UM 读保护）框或 “Task read protection”（任务读保护）框下注册密码，并在 CX-Programmer 的 “PLC Properties”（PLC 属性）对话框的 “Protection”（保护）标签页中选择 “Prohibit from saving into a memory card, and transferring program from PLC”。



2. 选择“PLC-Transfer-To PLC”（PLC- 传送至 -PLC）传送文件或选择“PLC-Protection-Set Password”（PLC- 保护 - 设置密码）并点击 OK 按钮。

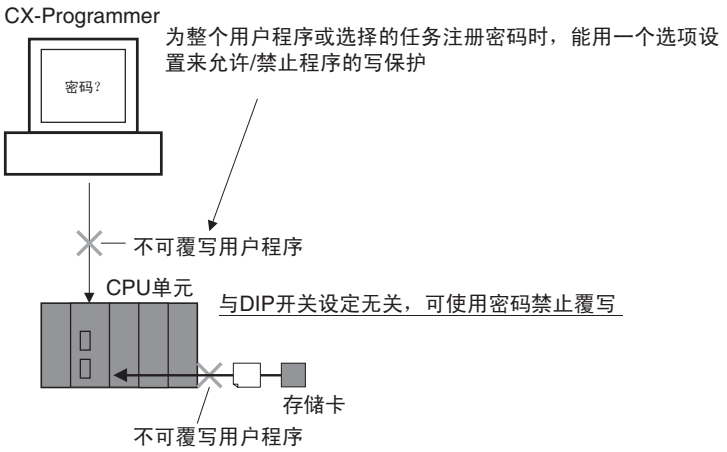
● 用途

- 可使用此选项，以防止通过密码将程序传送出 PLC。

允许 / 禁止程序覆写

● 概述

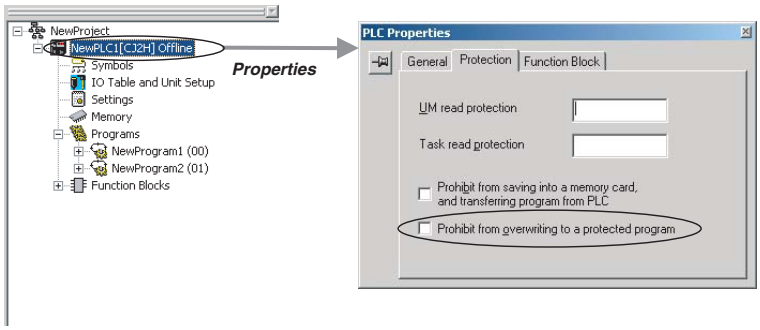
除了使用如上所述的 DIP 开关外，在为整个程序或选择的任务注册密码时，如果从 CX-Programmer 选择了写保护选项，则程序（或选择的任务）也可实现写保护。写保护设置能防止对程序的非法或偶然覆写。



- 如果在注册密码时，通过选择此选项来对选择的任务或程序进行写保护，则仅具有密码保护的任务（程序）受到保护，并将无法被覆写。通过在线编辑和任务下载等操作，仍可对其它任务或程序进行覆写。
- 当程序读保护无效时，所有任务 （程序）都可被覆写。
- 除非程序被传送到了CPU 单元，否则允许/ 禁止程序文件的设置将不发生作用。变更此设置后一定要传送程序。

● 操作步骤

1. 在 “UM read protection”（UM 读保护）框或 “Task read protection”（任务读保护）框下注册密码，并在 CX-Programmer 的 “PLC Properties”（PLC 属性）对话框的 “Protection”（保护）标签页中选择 “Prohibit from overwriting to a protected program”。



2. 选择 “PLC-Transfer-To PLC ”（PLC- 传送至 -PLC）来传送程序或选择 “PLC-Protection- Set Password”（PLC- 保护 - 设置密码）并点击 OK 按钮。

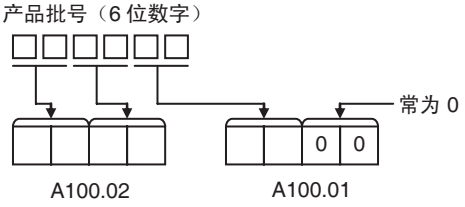
与密码保护相关的辅助区标志和位

名称	位地址	描述
UM 读保护标志	A99.00	指示 PLC （整个用户程序）是否有读保护 OFF：未设置 UM 读保护 ON：设置 UM 读保护
任务读保护标志	A99.01	指示所选择的程序任务是否有读保护 OFF：未设置任务读保护 ON：设置任务读保护
读保护的程序写保护	A99.02	指示是否选择了写保护选项，以防止对程序的意外覆写。 OFF：允许覆写 ON：禁止覆写 （写保护）
程序备份的允许 / 禁止位	A99.03	指示在设置了 UM 读保护或任务读保护时，是否能创建备份程序 / 网络符号 （变量）文件 （.OBJ 文件）。 OFF：允许备份程序文件的创建 ON：禁止备份程序文件的创建

10-6-3 使用产品批号进行程序操作保护

通过存储在辅助区域的字 A100.01 和 A100.02 中的产品批号，可对程序进行操作保护。用户不可对产品批号进行更改。

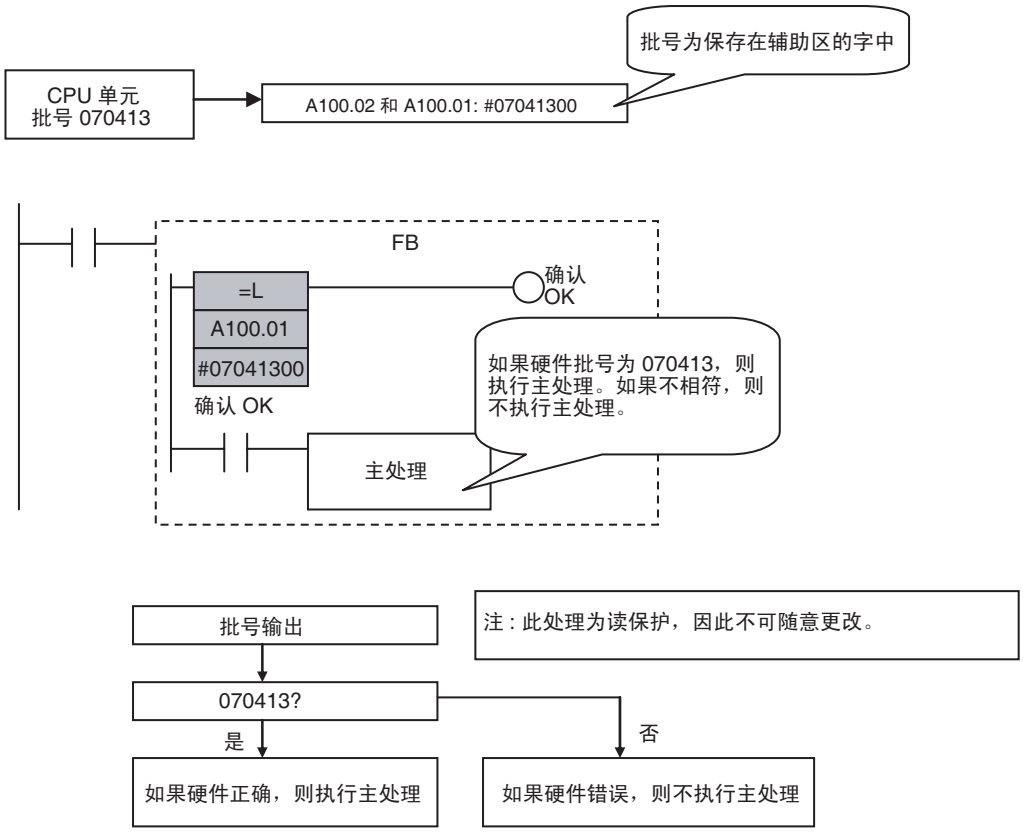
- 产品批号为 6 位数字，最左 4 位为保存在 A100.02 中，且最右 2 位为保存在 A100.01 中。



产品批号的存储示例

生产日期	产品批号	A100.02	A100.01
June 20, 2008	080620	0806	2000

应用示例：运行程序仅可用于带有指定产品批号的 CPU 单元。



10-6-4 FINS 命令写保护

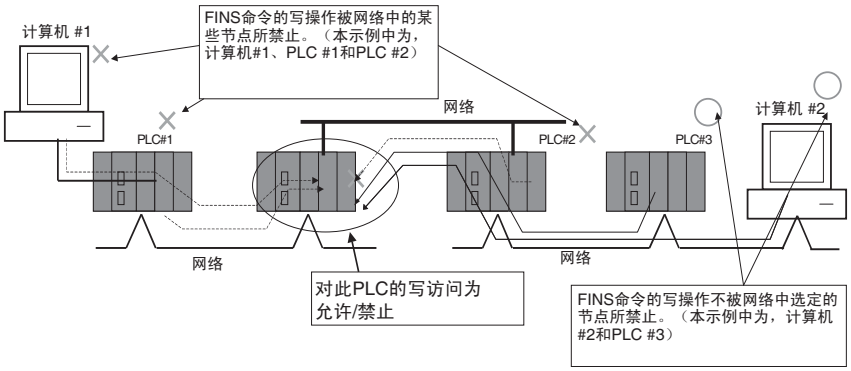
可以禁止以 FINS 命令的形式通过网络发送写操作和其它编辑操作到 PLC 的 CPU 单元（包括从 CX-Programmer， CX-Protocol， CX-Process 和其它使用 FinsGateway 的应用发送的写操作，但直接串行连接除外）。不禁止读处理。

FINS 写保护能禁止写处理，如用户程序、PLC 设置或 I/O 存储器等的下载、变更运行模式以及进行在线编辑。

可以在写保护中将所选择的节点排除在外，以使数据可从这些节点进行写入。

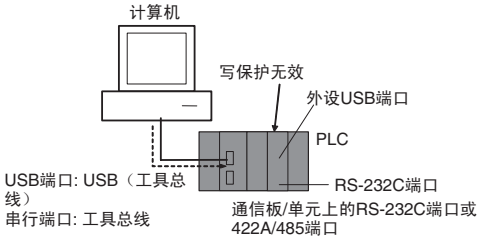
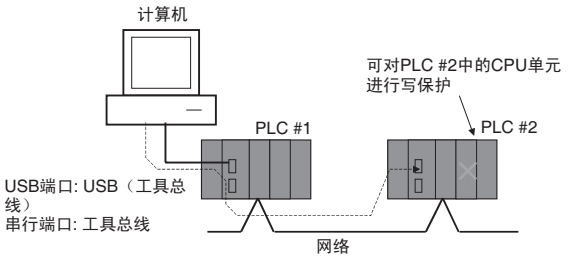
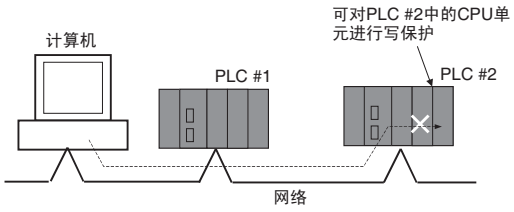
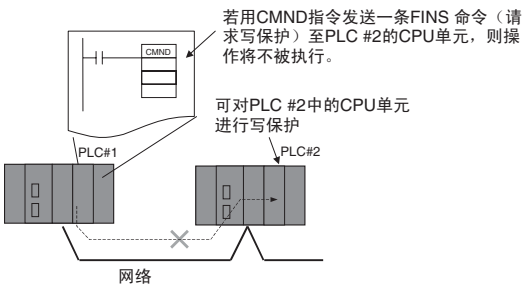
CPU 单元中的一个事件记录中自动记录了所有通过网络发送的写处理，并且可通过 FINS 命令读取此记录。

示例：



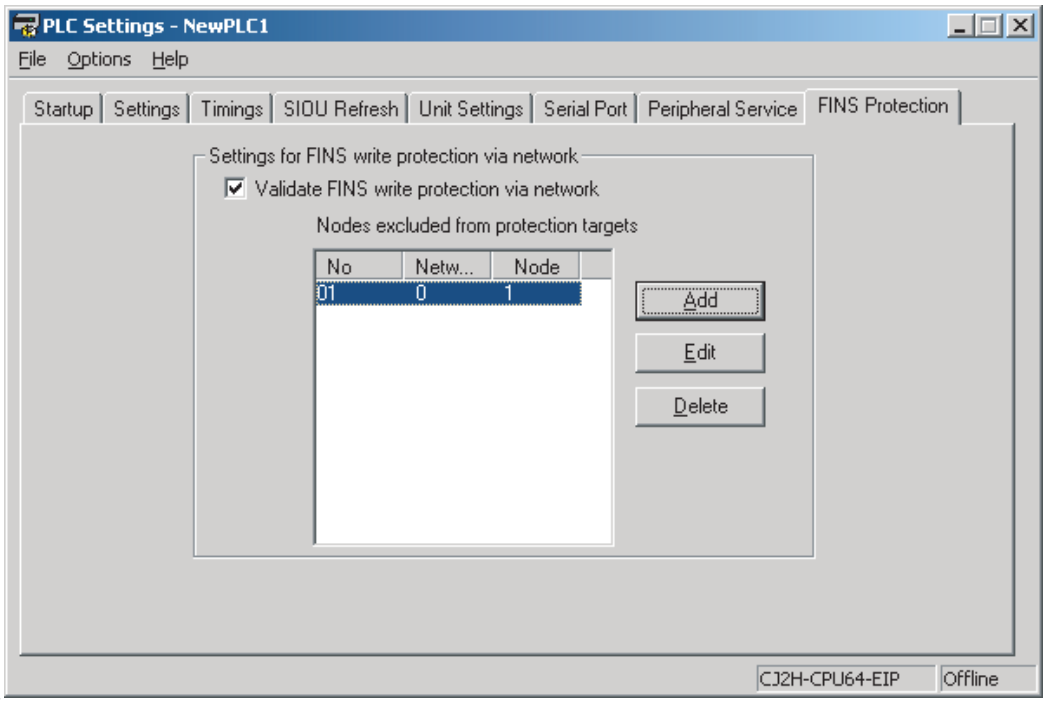
注　该功能仅禁止由 FINS 命令写入，因此对除了 FINS 命令以外的其它功能（诸如数据链接）作出的写保护无效。

● 写保护模式示例

连接模式		图 （示例）	写保护
从串行直接连接的计算机	直接连接至 PLC		不可应用
	网关连接 （串行到网络）至 PLC		可应用
从网络直接连接的计算机			可应用
从网络中的另一台 PLC			可应用

● 操作

使用 CX-Programmer 时，打开 PLC 设置中的 FINS 保护标签并选择“Validate FINS write protection via network”（使用 FINS 写保护）选项。当选择了该选项时，该 CPU 单元将不可通过网络（串行直接连接除外）使用 FINS 命令来执行写操作。此外，要允许来自特定节点的写操作时，可在 PLC 设置的“保护对象外节点指定”中设定其节点的网络地址及节点地址，（最多可将 32 个节点不设定为 FINS 写保护）。

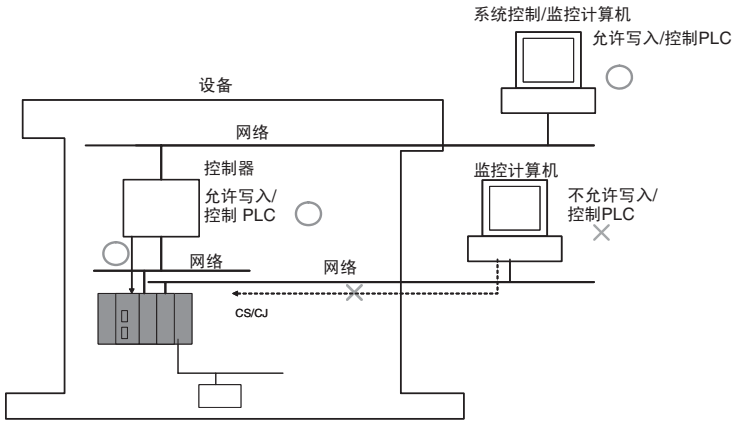


● PLC 设置

项目	描述	设定	默认设定
使用 FINS 写保护	设置 CPU 是否对通过网络发送的 FINS 命令予以写保护（并不禁止通过串行直接连接发送的 FINS 命令）。	OFF：禁止写保护 ON：允许写保护	OFF：禁止写保护
保护对象外节点指定	该区列出了网络中不受 FINS 写保护限制的节点。可指定 32 个节点。 仅当允许 FINS 写保护时这些设定有效。		
	网络地址： FINS 命令源的网络地址	00 ～ 7F hex	
	节点地址： FINS 命令源的节点地址	01 ～ FE hex, 或 FF hex （FF hex：未指定节点地址）	
不包括在 FINS 写保护内的节点数 （请勿设置该数值。应由 CX-Programmer 自动计算）。	包含不受 FINS 写保护的节点数。 若指定了 0 （不存在未被写保护的节点），则除了本地节点之外的所有节点均禁止 FINS 写入命令 仅当允许 FINS 写保护时该设定有效。	0 ～ 32 （00 ～ 20 hex） （数值 0 表示所有节点被写保护）	0 （所有节点被写保护）

● 用法

- 可对系统进行配置，以使 PLC 只能由网络中的授权节点所写入。（例如，当系统的控制 / 监控计算机是唯一被允许在一个设备中对控制器进行写入的节点时使用该功能）。
  通过限制可写入 PLC 的节点数，可以防止由数据监控期间的意外覆盖所造成的系统问题。



- 受网络 FINS 写保护限制的运行

FINS 写命令

下列 FINS 命令当通过网络发送至 CPU 单元时受 FINS 写保护的限制。

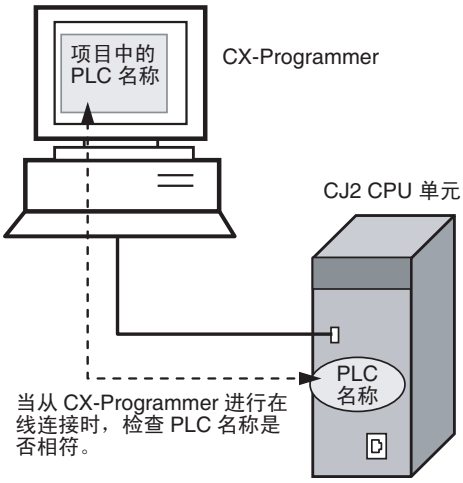
代码	命令名称
0102 hex	存储器区写入
0103 hex	存储器区填充
0105 hex	存储器区传送
0202 hex	参数区写入
0203 hex	参数区填充 （清除）
0307 hex	程序区写入
0308 hex	程序区清除
0401 hex	运行
0402 hex	停止
0702 hex	时钟写入
0C01 hex	获得访问权

代码	命令名称
2101 hex	错误清除
2103 hex	错误日志指针清除
2203 hex	单一文件写入
2204 hex	文件存储器格式化
2205 hex	文件删除
2207 hex	文件复制
2208 hex	更改文件名
220A hex	存储器区 - 文件传送
220B hex	参数区 - 文件传送
220C hex	程序区 - 文件传送
2215 hex	创建 / 删除目录
2301 hex	强制置位 / 复位
2302 hex	取消强制置位 / 复位

10-6-5 PLC 名称

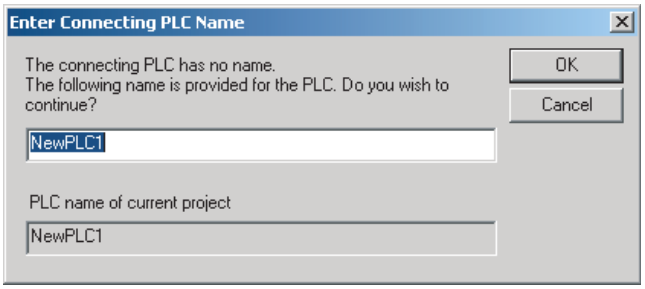
PLC 名称

当从 CX-Programmer 进行在线连接时，可检查项目中的 PLC 名称与连接对象处 CPU 单元的 PLC 名称是否相符。这样可避免从 CX-Programmer 进行不正确连接。



注册 PLC 名称

当对使用默认设定或已清除所有存储器的 CJ2 CPU 单元进行在线编辑时，将显示“Enter Connecting PLC Name”（输入连接 PLC 名称）对话框。

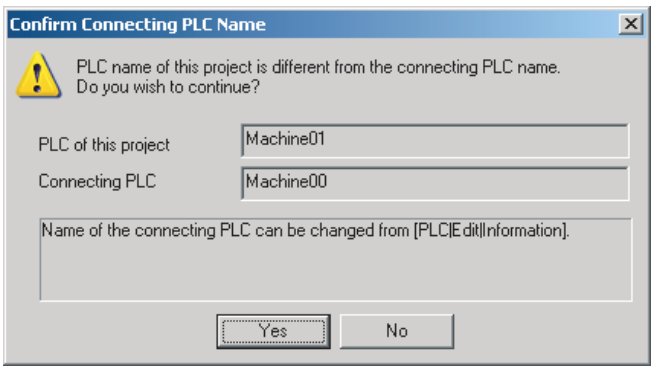


输入 PLC 名称，注册到在线连接的 CJ2 CPU 单元。然后，点击 “OK”（确认）按钮。如果未输入 PLC 名称，则点击 “Cancel”（取消）按钮。

PLC 名称不匹配

如果 PLC 名称保存在 CJ2 CPU 单元中，则在进行在线连接时，系统将自动检查连接对象处 CPU 单元的 PLC 名称是否与项目中的 PLC 名称相符。

如果连接对象处 CJ2 CPU 单元中保存的 PLC 名称与项目中的 PLC 名称不相符，则将显示下述警告。选择是否继续在线连接。



- “Yes ”（是）按钮
  点击 “Yes ”（是）按钮，选择以不同名称继续进行在线连接。
- “No” （否）按钮
  点击 “No” （否）按钮，选择离线。

附加信息

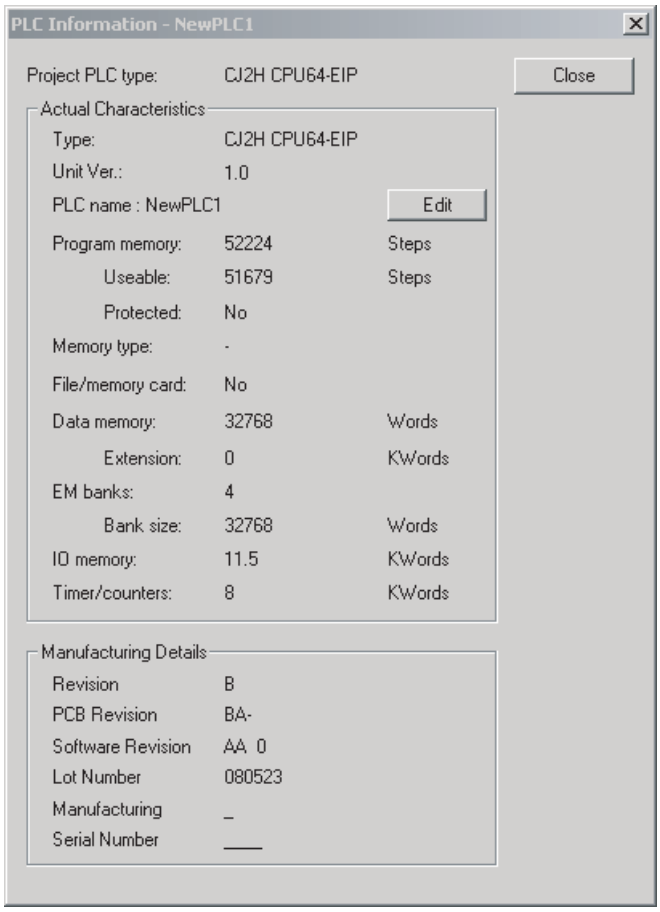
不需要进行自动的 PLC 名称匹配时，取消 PLC 属性对话框中的 “Check the connecting PLC name when going online”选择框。

变更 PLC 名称

使用下述步骤变更保存在 CJ2 CPU 单元中的 PLC 名称。

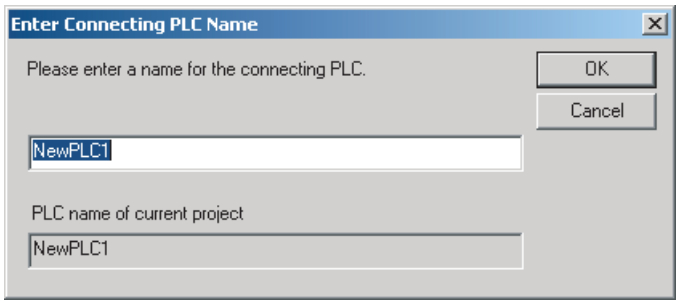
1. 以 CX-Programmer 进行在线连接，并从 PLC 菜单下选择 “Edit - Information”（编辑 - 信息）。

   将显示下述对话框。



2. 点击 PLC 名称区右侧的 “Edit”（编辑）按钮。

   将显示下述的 “Enter Connecting PLC Name”（输入连接 PLC 名称）对话框。



3. 输入 PLC 名称以注册到连接对象 PLC，并点击 “OK”（确认）按钮。

10-7 调试

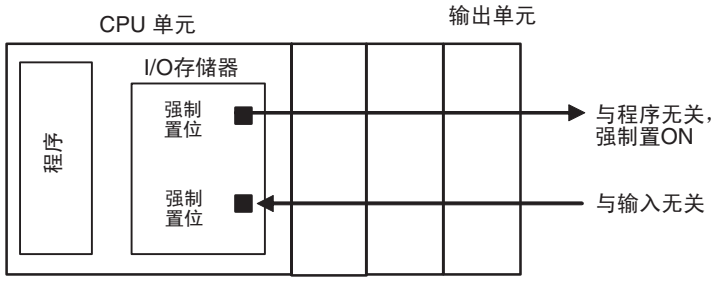
10-7-1 强制置位 / 复位

CX-Programmer 可以对指定位 （CIO 区，辅助区，HR 区，和定时器 / 计数器完成标志）进行强制置位（ON）或复位 （OFF）。强制状态将比程序或 I/O 刷新时的状态输出具有更高的优先权。此状态不能通过指令重写，并且无论程序或外部输入的状态如何，都将对其进行保存，直到通过 CX-Programmer 将其清除。

强制置位 / 复位操作可用于在试运行操作期间进行强制输入 / 输出或在调试期间强制一定的条件。

强制置位 / 复位操作为可在 MONITOR 或 PROGRAM 方式下执行，但不可在 RUN 模式下执行。

注　同时打开强制状态保持位 （A500.13）和 IOM 保持位 （A500.12），以在切换运行模式时保持已强制置位或复位的位状态。
打开强制状态保持位 （A500.13）和 IOM 保持位 （A500.12），并在 PLC 设置的启动设置中，设定强制状态保持位为在启动时保持强制状态保持位的状态。通过此操作，实现在电源接通时保持已强制置位或复位的位状态。



可对下述区中的位进行强制置位和复位。

CIO 区（I/O 位，数据链接位，CPU 总线单元位，特殊 I/O 单元位，内部 I/O 位），工作区，定时器完成标志， HR 区，计数器完成标志以及用于自动地址分配的 EM 区部分 *1。

*1 CJ2H-CPU64/65-EIP：EM 组 3 hex
CJ2H-CPU66-EIP：EM 组 6 ～ 9 hex
CJ2H-CPU67-EIP：EM 组 7 ～ E hex
CJ2H-CPU68-EIP：EM 组 11 ～ 18 hex

● CX-Programmer 操作

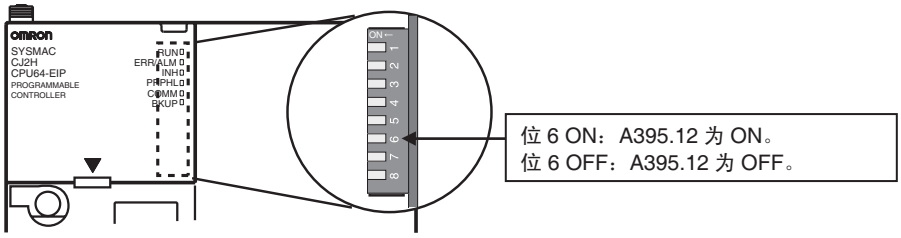
- 为强制置位 / 复位选择位。
- 选择强制置位或复位。
- 清除强制状态 （包括同时清除所有强制状态）

● 强制状态保持位

名称	地址	功能	访问
强制状态保持位	A500.13	将此位置 ON，以在接通电源时或在 PROGRAM 和 RUN/MONITOR 间变更模式时，保持已强制置位 / 复位的位状态。	读 / 写

10-7-2 测试输入

将 CPU 单元前的 DIP 开关位 6 的 ON/OFF 状态保存到辅助区中 DIP 开关位状态标志 （A395.12）中。为了进行调试或其他目的，可对输入进行手动操作，以在不使用输入单元的情况下进行测试。

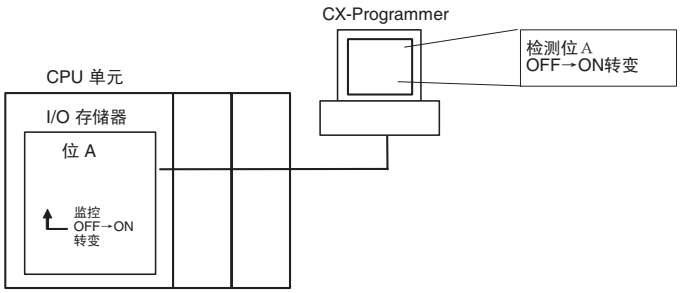


● 辅助区标志和字

名称	地址	功能	访问
DIP 开关位状态标志	A395.12	保存 CPU 单元 DIP 开关的位 6 设定状态。（每循环刷新）	只读

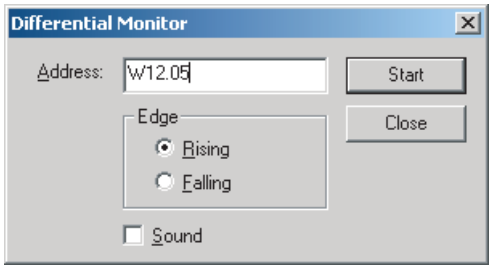
10-7-3 微分监控

当 CPU 单元检测到由 CX-Programmer 编程设备设定的位已从 OFF 转为 ON 或从 ON 转为 OFF 时，将在微分监控完成标志 （A508.09）中将指明结果。当对微分监控设定的条件满足时，标志将变为 ON。 CX-Programmer 将在屏幕上监控和显示这些结果。



● CX-Programmer 编程设备的操作

1. 右键点击微分监控选定位。
2. 从 PLC 菜单点击微分监控。将显示 “Differential Monitor”（微分监控）对话框。

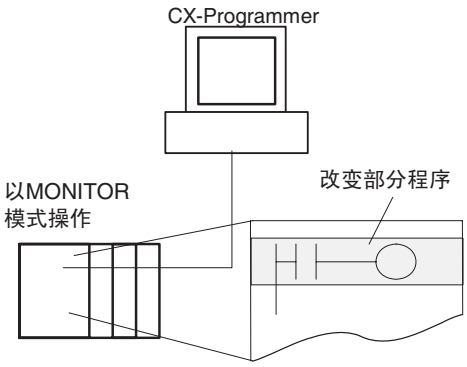


3. 点击 “Rising”（上升）或 “Falling”（下降）。
4. 点击 “Start”（开始）按钮。当检测到指定的变化时，蜂鸣器将响起，并且计数值将递增。
5. 点击 “Stop”（停止）按钮。停止微分监控。

10-7-4 在线编辑

CPU 单元在 MONITOR 或 PROGRAM 方式下时，直接使用 CX-Programmer 在线编辑功能，在 CPU 单元中增加或改变部分程序。从 CX-Programmer 一次可以增加或改变一个或多个程序部分。因此，此功能适用于在不停止 CPU 单元运行的情况下，进行程序的小修改。

对于不同的编辑任务，可以同时通过两台以上的计算机运行 CX-Programmer 进行在线编辑。



当在 MONITOR 方式下对 CPU 单元中程序在线编辑，循环将增加一至数个循环的时间。

在线编辑后，在闪存存储器器中备份数据，循环时间也将增加。在这个期间 BKUP 指示器将为亮灯。备份的进度显示在 CX-Programmer 上。每循环的循环时间延迟将为约 1 ms。

在不等待备份完成的情况下，可以连续编辑的数目是有限的。实际数目取决于执行的编辑类型，但最多可达到 400 项编辑。

如果超过限制数目，在 CX-Programmer 上将显示一条信息，在 CPU 单元完成备份数据之前不能继续编辑。当备份完成后，则将可再次继续在线编辑。

正确使用注意事项

在 MONITOR 方式下，当使用在线编辑对程序进行重写时，循环将比正常情况来得长，因此必须确认延长的时间量没有超过在 PLC 设置中设定的循环监控时间。如果它确实超过了监控时间，那么将发生循环时间超时， CPU 单元将停止。在变更为 RUN /MONITOR 模式之前，请首先选定 PROGRAM 模式来重新启动 CPU 单元。

注 1 将要进行在线编辑的程序中所保存的 DIFFERENTIATE UP 指令 （DIFU 或带@的指令）以及 DIFFERENTIATE DOWN 指令 （DIFD 或带 % 的指令）的内部状态 （微分标志）进行初始化。

2 如果进行在线编辑的任务中包含一个块程序，则由 AIT（805），TIMW（813），TIMWX（816），CNTW （814），CNTWX （817），TMHW （815）或 TMHWX （818）创建的 WAIT 状态将被 BPPS （811）清除，同时下一个操作将从头开始。

● 从 CX-Programmer 执行在线编辑

1. 显示将被编辑的程序部分。
2. 选择将被编辑的指令。
3. 选择程序，在线编辑和开始。
4. 编辑指令。

5. 选择程序，在线编辑并传送变更。指令将被检查，如果没有错误，他们将被传输到 CPU 单元。CPU 单元中的指令将被重写，并且此时的循环时间将增长。

! 警告

仅在证实延长的循环不影响操作后，才可继续在线编辑。如果循环太长时，输入信号将可能无法输入。





附加信息

临时禁止在线编辑

可以在一个指定时间禁止在线编辑，确保在那个循环中机器控制的响应特性。禁止在线编辑可避免因某人执行在线编辑而导致循环时间的意外增长。

- 禁止在线编辑设定
  下述两种设定都可用于禁止在线编辑。
  将在线编辑禁止位确认器 （A527.00 ～ A527.07）设定为 5A hex。
  将在线编辑禁止位 （A527.09）置 ON。
- 禁止在线编辑时操作
  如果有来自 CX-Programmer 的在线编辑请求，则系统将进入待机状态并且不执行在线编辑。在线编辑等待标志 （A201.10）将置 ON。随后，当用户将在线编辑禁止位 （A527.09）转为 OFF 时，执行在线编辑。如果在线编辑已为待机，则任何在线编辑操作都将被忽略。

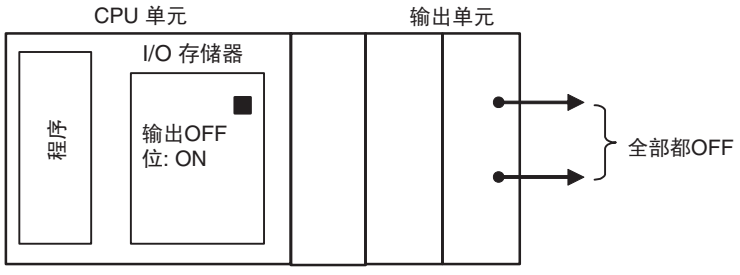
● 相关辅助位 / 字

名称	地址	描述
在线编辑禁止位确认器	A527.00 ～ A527.07	对在线编辑禁止位 （A527.09）进行确认。 5A hex：在线编辑禁止位确认 非 5A hex：在线编辑禁止位未确认
在线编辑禁止位	A527.09	要在线编辑禁止时，将此位置 ON 并将在线编辑禁止位确认器 （A527.00 ～ A527.07）设定为 5A。
在线编辑等待标志	A201.10	当因在线编辑被禁止，而使在线编辑处于待机状态时，置 ON。

10-7-5 关闭输出

将输出 OFF 位 （A500.15）转为 OFF 时，所有输出单元的输出都将为 OFF，且这与 I/O 存储器中的输出状态无关。当单元在 RUN 或 MONITOR 模式下操作时，可将此用于紧急的错误处理。CPU 单元前的 INH 指示灯将为亮黄灯。

若使用电池时，则与电源 ON/OFF 无关，输出 OFF 位的状态将保持。



附加信息

如果运行模式从 RUN/MONITOR 模式切换为 PROGRAM 模式，则在默认状态下（即 IOM 保持位（A500.12）为 OFF），将清除所有的输出 （即为 OFF）。如果通过用户程序将 I/O 存储器中的输出位转为 ON，则 I/O 刷新时输出将置 ON。无论运行模式为何，都要将所有输出单元的输出转为 OFF 时，请使用输出 OFF 位。

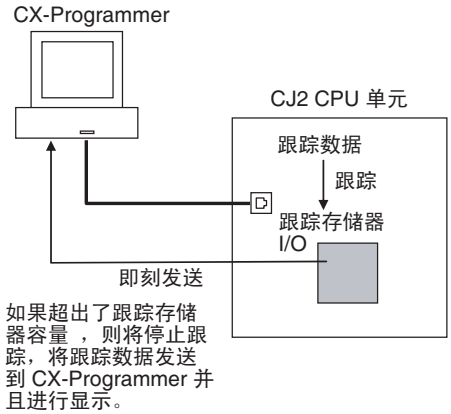
10-7-6 数据跟踪

数据跟踪功能可使用下述任一定时方法，对指定 I/O 存储器数据进行采样。将采样数据保存在跟踪存储器中，随后可从 CX-Programmer 对这些数据进行读取和检查，并作为文件保存。

可使用下述两种跟踪方式。

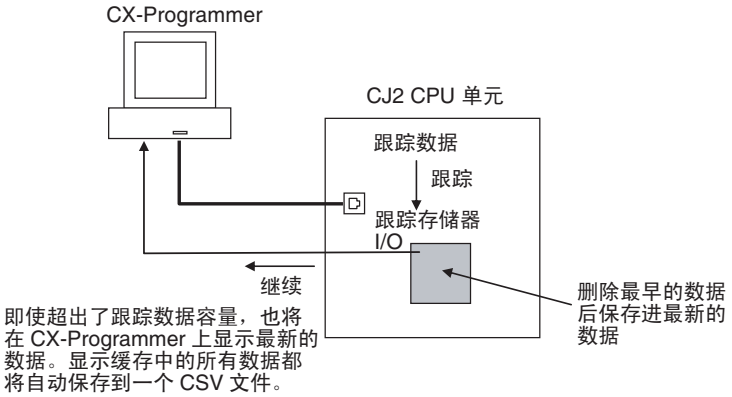
● 正常跟踪

当在短时间内执行高速数据收集时，使用此方法，如用于调试。如果超出了跟踪存储器容量，则将停止跟踪，并显示跟踪存储器中最近的跟踪结果，且同时将数据保存在一个 CSV 文件中。



● 持续跟踪

当在较长期间内执行高速数据收集时，使用此方法。即使超出了跟踪存储器容量，也将继续跟踪，并将持续读取跟踪数据到 CX-Programmer。任何时候当显示缓存出现饱和时，就自动将数据保存到 CSV 文件。以此通过多个 CSV 文件，实现了长时间数据跟踪开始到结束的结果数据保存。



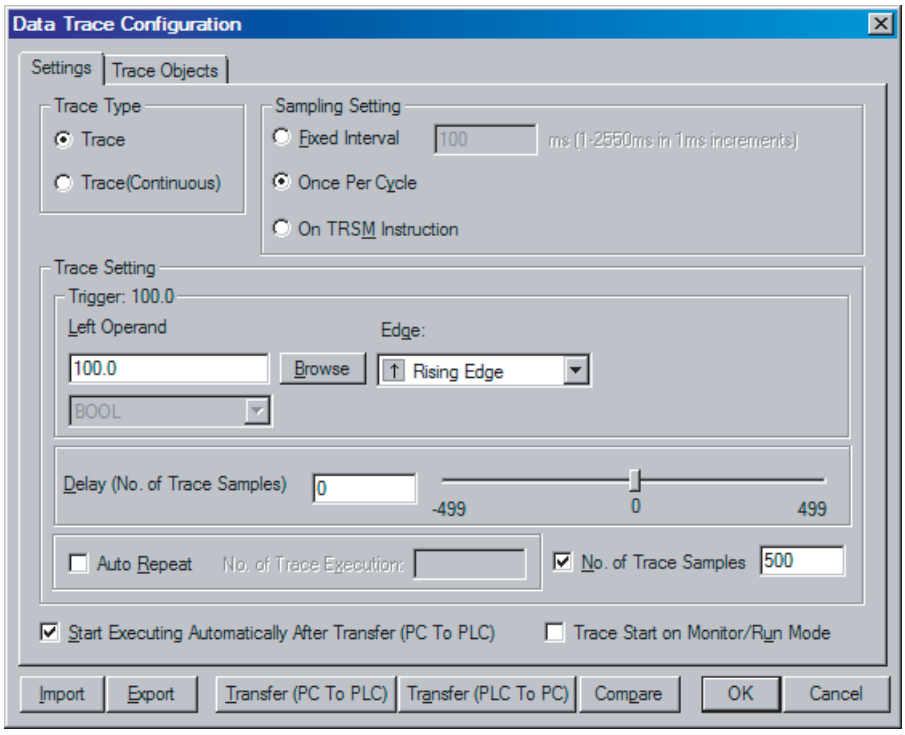
● 数据采样和定时

- 可在 I/O 存储器中指定最多 31 位和 16 个字对其采样。
- 可指定下述数据类型进行采样。
  - BOOL （位）
  - UINT （单字，无符号，二进制）
  - UDINT （双字，无符号，二进制）
  - ULINT （四字，无符号，二进制）
  - UINT BCD （单字，无符号，二进制）
  - INT （单字，带符号，二进制）
  - DINT （双字，带符号，二进制）
  - LINT （四字，带符号，二进制）
  - UDINT BCD （双字，带符号，二进制）
  - REAL （双字，浮点）
  - LREAL （四字，浮点）
  - CHANNEL （字）
  - WORD （单字，十六进制）
  - DWORD （双字，十六进制）
  - LWORD （四字，十六进制）
- I/O 存储器可以下述时间进行采样。
  - 指定采样时间 （1 ～ 2,550 ms，以 1ms 为单位）
  - 每循环一次
  - 当执行跟踪存储器指令 （TRSM （045）指令）时

● 数据跟踪步骤概述

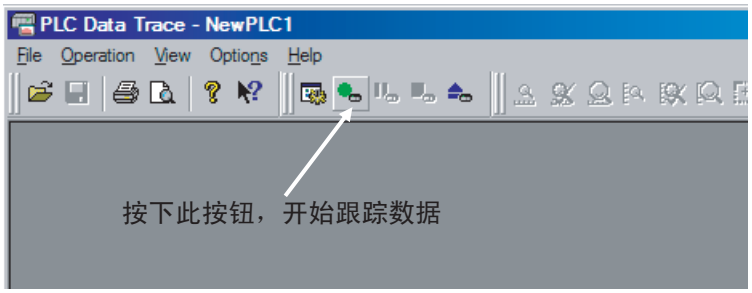
通过下述步骤进行跟踪。

1. 使用 CX-Programmer 设定跟踪参数 （选择 “PLC - Data Trace （PLC- 数据跟踪）” 或 “PLC - Time Chart Monitor（PLC-时间图表监视器）”，以打开数据跟踪配置窗口，随后选择“Execute - Set（执行-设定）”）。

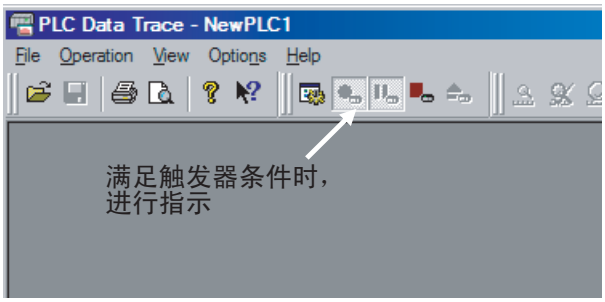


- 执行正常跟踪
  在 “Trace Type”（跟踪类型）区中选择 “Trace”（跟踪）。在 “Trace Objects”（跟踪对象）标签页上设定采样数据的地址。可指定多个地址。在 “Settings”（设定）标签页上的 “Trace Setting”（跟踪设定）区中设定触发条件和延迟时间。
- 执行持续跟踪
  在 “Trace Type”（跟踪类型）区中选择 “Trace （continuous） ”（跟踪 （持续） ） 。在 “Trace Objects”（跟踪对象）标签页上设定采样数据的地址。可指定多个地址。在 “Settings”（设定）标签页上的 “Trace Setting”（跟踪设定）区中指定屏幕显示缓存大小、跟踪时间以及保存所收集数据的 CSV 文件的文件夹。

2. 将 “Trace Start Bit”（跟踪开始位） （A508.15）置 ON 或按下如下按钮，开始跟踪。

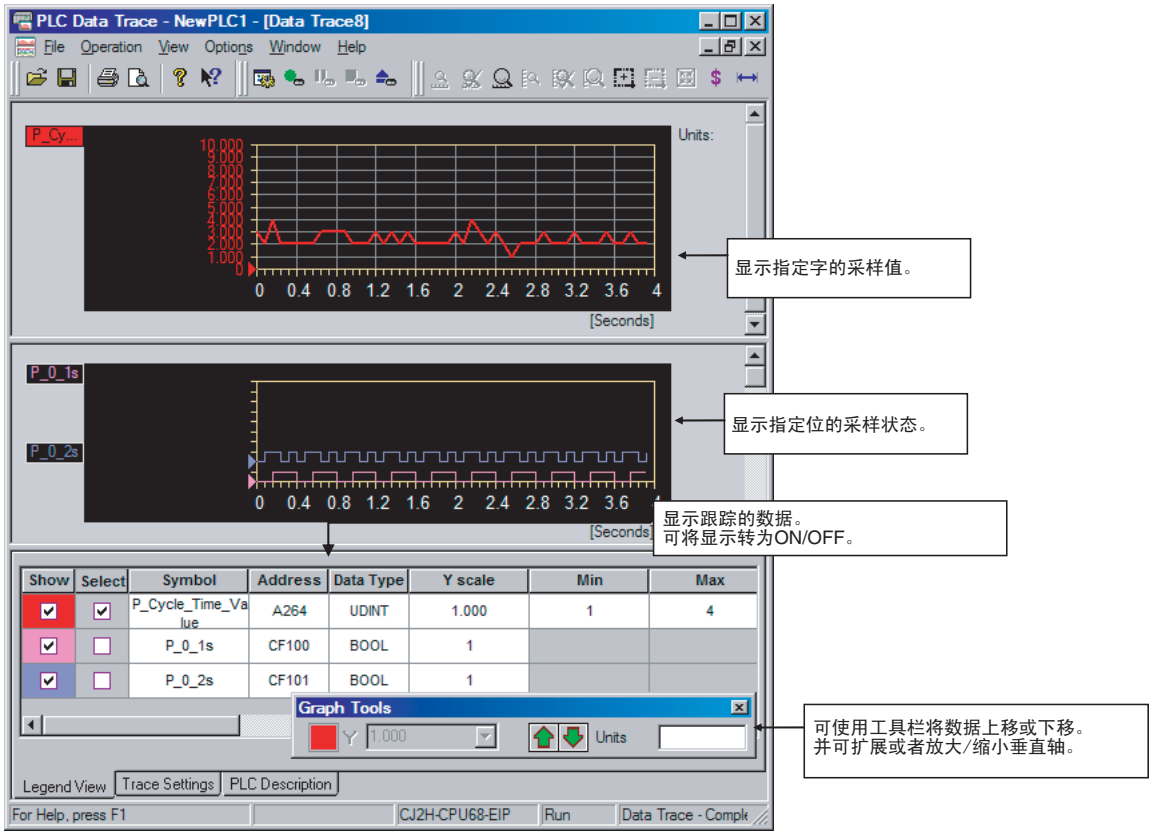


当满足触发器条件时，将如下显示。



● 数据跟踪窗口

下述配置为数据跟踪窗口的基本概述。



关于步骤和设定的详细内容，请参见 CX-Programmer 操作手册 （Cat. No. W446）。

● CJ2 CPU 单元数据跟踪规格

下表为 CJ2 CPU 单元和 CJ1 CPU 单元的数据跟踪规格比较。

项目	CJ2 CPU 单元	CJ1 CPU 单元
存储器容量	CPU64-EIP/65-EIP：8,000 字 CPU66-EIP：16,000 字 CPU67-EIP/68-EIP：32,000 字 可使用 EM 区部分替代数据跟踪区。 （最大 EM 区跟踪数据容量： 32 K 字× 25 组）。	4,000 字
采样大小	采样位：31 采样字：16 单字数据 = 最大 16 双字数据 = 最大 8 四字数据 = 最大 4	采样位：31 采样字：最大 6
循环 （常数时）	1 ～ 2,550 ms （以 1ms 为增量单位）	10 ～ 2,550 ms （以 10ms 为增量单位）
启动时自动开始	可	不可
持续跟踪	通过在超出跟踪存储器容量前将跟踪数据读取到 CX-Programmer，实现持续跟踪。	无
用于设定触发器长度的数据长度	数据长度：1 字， 2 字或 4 字	1 字
触发器的条件格式	条件中可使用 AND 和 OR，并可使用＝， >，≥， <，≤，≠。	条件中可使用 AND 和 OR，但仅可使用＝。
延迟值	-32,768 ～ 32,767	-1,999 ～ 2,000

10-7-7 存储错误时的停止位

如果梯形图程序中的指令不在输入参数的范围内时，错误标志 （P_ER）将为 ON。当执行指令时，可传送此标志以检查指令错误。

如果在 PLC 设置中选择 “Stop CPU on Instruction Error”（指令错误操作时停止 CPU），则当发生下述情形时产生程序错误，并将停止运行。可使用此功能检查是否发生指令错误。

● 导致程序错误 （A295）的指令错误

名称	地址	描述
指令处理错误标志	A295.08	当发生指令处理错误且 PLC 设置设为遇指令错误停止运行时，此标记和错误标记 （P_ER）将置 ON。
间接 DM/EM BCD 错误标志	A295.09	当出现一个间接 DM/EM BCD 错误，并且 PLC 设置中设定为间接 DM/EM BCD 错误时停止运行，此标志将转为 ON。
非法访问错误标志	A295.10	当出现一个非法访问错误 （如，系统区，对转换到 EM 区的区域进行 R/W 访问，对写保护区进行写访问，或间接 DM/EM BCD 错误），此标志将转为 ON。

当因程序错误导致运行停止时，停止位置将如下所示进行保存。

● 程序错误标志 （A294）

名称	地址	描述
程序错误任务	A294	提供了因程序错误导致程序执行停止时正在执行的任务类型和数据。 0000 ～ 007F hex （对应于任务 0 ～ 127） 8000 ～ 80FF hex （对应于中断任务 0 ～ 255，包括额外循环任务）

可用于检查发生致命错误的任务。当清除致命错误时，程序错误任务也将被清除。

● 程序错误位置 （A299：上位字节， A298：下位字节）

如果因程序错误导致执行停止，则用于下述停止位置的程序地址将被保存。

注　　程序地址为仅对梯形图程序有效。其不可用于 ST 或 SFC 语言程序。

- 梯形图程序中错误：
  从梯形图程序的开始处计算地址。
- 梯形图动作程序中错误：
  从梯形图动作程序的开始处计算地址。
- 转换程序中错误：
  从转换程序的开始处计算地址。
- 功能块中错误：
  从功能块程序的开始处计算地址。

10-7-8 故障报警指令

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，而 FALS （007）产生一个停止程序执行的致命错误。

当满足用户定义错误条件 （FAL （006）或 FALS （007）执行条件）时，故障报警指令将执行，同时进行以下处理。

1. FAL 错误标志 （A402.15）或 FALS 错误标志 （A401.06）转为 ON。
2. 相应的错误码写入辅助区中 A400。
3. 错误码和发生的时间存储在错误记录中。
4. 位于 CPU 单元前面的错误指示灯将闪烁或点亮。
5. 如果 FAL （006）执行完毕， CPU 单元将继续运行。
   如果 FALS （007）执行完毕，CPU 单元将停止运行。（程序执行停止）。

FAL （006）的操作



当执行条件 A 变为 ON，生成 FAL 编号 002 的错误，A402.15（FAL 错误标志）转为 ON，A360.02（FAL 编号 002 标志）转为 ON。程序执行继续。

通过执行 FAL 编号 00 的 FAL （006）或从 CX-Programmer 执行错误读取 / 清除操作，可以清除由 FAL （006）产生的错误。

FALS （007）的操作



当执行条件 B 变为 ON，生成 FALS 编号 003 的错误，A401.06（FALS 错误标志）变为 ON。程序执行停止。

通过消除错误的原因和从CX-Programmer执行错误读取/ 清除操作，可以清除由FALS（006）产生的错误。

10-7-9 模拟系统错误

可使用 FAL （006）和 FALS （007）有意地产生致命或非致命系统错误。系统调试时，可使用这个方法在可编程终端 （PT）或其它操作接口上测试显示信息。

使用以下步骤：

1. 在 A529 将 FAL 或 FALS 编号设定为模拟使用。（当 FAL（006）和 FALS（007）模拟错误时，使用 A529）。
2. 将 FAL 或 FALS 编号设定为模拟使用，作为 FAL （006）或 FALS （007）的第一个操作数。
3. 将错误码和错误设定为模拟，作为 FAL （006）或 FALS （007）的第二个操作数（S 和 S+1）。指定 FAL （006）为非致命错误，指定 FALS （007）为致命错误。

为了模拟多个系统错误，在 A529 指定相同值为第一个操作数，如上所述多次使用 FAL （006）或 FAL（007）指令。

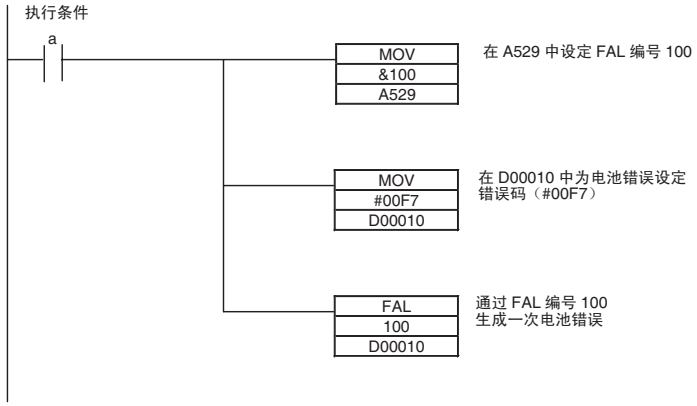
正确使用注意事项

可使用此功能，通过在 CPU 单元中运行模拟错误检测，以检查应用中的检测运行是否正常。当系统运行时，可通过删除 FAL （006）或 FALS （007）指令，或者通过始终输入 OFF （P_Off）信号作为输入条件来禁止此功能。

● 辅助区标志和字

名称	地址	操作
用于系统错误模拟的 FAL/FALS	A529	设定一个虚拟 FAL/FALS 编号，用于模拟系统错误。 0001 ～ 01FF Hex：FAL/FALS 编号 1 ～ 511 或 0000 或 0200 ～ FFFF Hex：无 FAL/FALS 编号用于系统模拟错误。

● 电池错误示例



正确使用注意事项

通过执行 FAL （006）和 FALS （007）指令，清除模拟的致命和非致命系统错误。使用与实际系统错误时相同的方法清除模拟系统错误。相关详细内容，请参见 CJ2 CPU 单元硬件操作手册（Cat. No. W472）的第 6 章故障检修。使用 FAL （006）和 FALS （007）模拟的所有系统错误都可通过再次启动电源进行清除。

10-7-10 故障点检测

执行 FPD （269）指令进行时间监控和逻辑诊断。在一个特定的监控时间内，如果诊断输出不变为 ON，时间监控功能产生一个非致命错误。逻辑诊断功能指明哪个输入可防止诊断输出变为 ON。

时间监控功能

在一个特定的监控时间内，如果诊断输出不变为 ON，执行 FPD（269）并使进位标志为 ON，启动计时。进位标志可通过编程作为一个错误处理块的执行条件。 FPD （269）也可通过编程生成一个所需 FAL 编号的非致命 FAL 错误。

产生一个 FAL 错误时，当前信息将被记录并在 CX-Programmer 上显示。可设定 FPD （269）在信息之前输出逻辑诊断的结果 （防止诊断输出变为 ON 的位地址）。

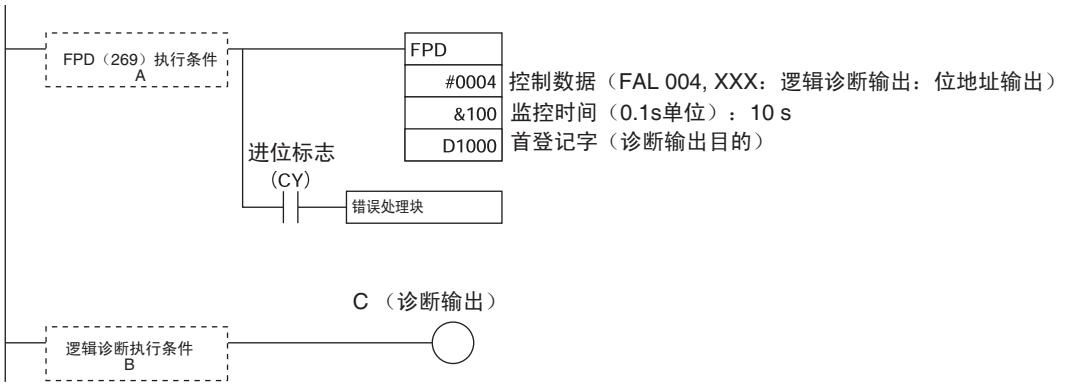
可使用教学功能自动决定实际所需的时间，以用于诊断输出置 ON 和设定监控时间。

逻辑诊断功能

FPD （269）决定是那个输入位引起诊断输出保持 OFF，并且输出那个位的地址。输出可以设定为位地址输出 （PLC 存储器地址）或信息输出 （ASCII）。

如果选定位地址输出，可将位的 PLC 存储器地址传送到一个变址寄存器中，在以后的处理中可以间接寻址变址寄存器。

如果选定信息输出，位地址将以一个可在 CX-Programmer 中显示的 ASCII 信息登记记录。



- **时间监控：**

  输入 A 后监控在 10 秒内输出 C 是否变为 ON。如果 C 在 10 秒内没有变为 ON，将检测出故障且进位标志置 ON。进位标志执行错误处理块程序。同时生成编号 004 的 FAL 错误 （非致命错误）。

- **逻辑诊断：**

  FPD （269）决定在块 B 中的输入位，防止输出 C 继续 ON。输出那个位地址到 D1000。

● 辅助区标志和字

名称	地址	操作
错误码	A400	出现一个错误时，它的错误码存储在 A400。
FAL 错误标志	A402.15	执行 FAL （006）时，为 ON。
FALS 错误标志	A401.06	执行 FALS （007）时，为 ON。
执行 FAL 编号标志	A360 ～ A391	当一个 FAL （006）或 FALS （007）错误发生时，相应标志变为 ON。
错误记录区	A100 ～ A199	错误记录区域包含有最近发生的 20 个错误信息。
错误记录指针	A300	作为从错误记录区 （A100）开始点的一个偏置，当一个错误发生时，错误记录指针递增 1，以指出下一个错误记录将记录的位置。
错误记录指针重置位	A500.14	使这位为 ON，重置位错误记录指示器 （A300）位 00。
FPD 教学位	A598.00	执行 FPD （269）时，若要自动设定监控时间时，可将此位置 ON。

11

编程设备与通信

本章描述了如何从 CX-Programmer 访问 PLC 以及串行通信和跨网络通信。

11-1 从 CX-Programmer 访问 PLC 11-2
11-1-1 概述 11-2
11-1-2 可访问 PLC 的系统配置 11-4
11-1-3 从 CX-Programmer 访问 PLC 11-8
11-1-4 自动在线连接 11-11
11-2 串行通信 11-15
11-2-1 串行通信概述 11-15
11-3 通信网络 11-21

11-1 从 CX-Programmer 访问 PLC

11-1-1 概述

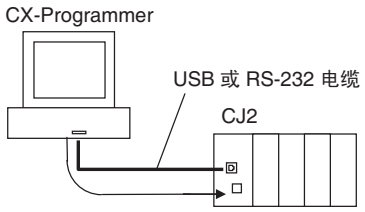
本节提供了关于如何从 CX-Programmer 访问 PLC 的概述。

连接到 PLC

可使用下述任一方法，实现从 CX-Programmer 访问 PLC。

● 直接连接到 PLC

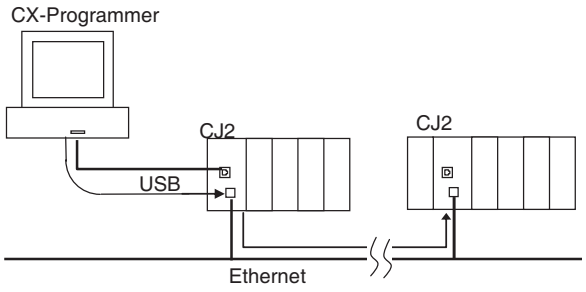
通过串行端口，可对直接连接到 CX-Programmer 的 PLC 进行在线连接。



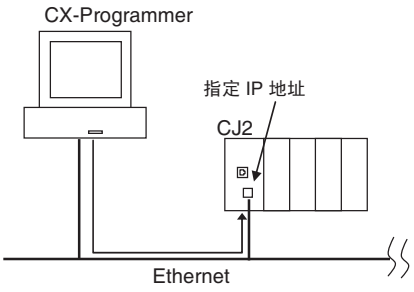
● 在网络上访问 PLC

可使用下述任一方法，在网络上访问 PLC。

通过直接连接到 CX-Programmer 的 PLC 的串行端口，在网络上在线连接 PLC。



通过直接连接到网络的计算机，在网络上在线连接 PLC。



在线连接到 PLC 的步骤

以下两种步骤之一可以用于从 CX-Programmer 在线连接到 PLC。

● 改变 PLC 对话框

设定该方法用于访问 CX-Programmer 项目的目标 PLC。当创建项目时进行该设定，但在项目创建后可被改变。

● 自动在线连接

自动在线连接用于访问已经建立连接的 PLC，并且无需创建 CX-Programmer 项目就可使用。

直接连接

直接连接为用于通过串行端口对直接连接到 CX-Programmer 的 PLC 进行在线连接。

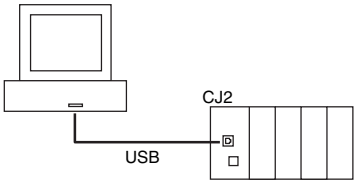
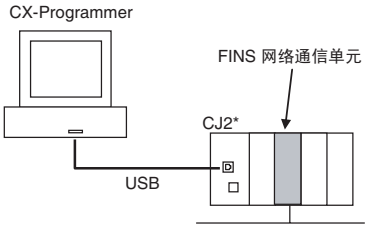
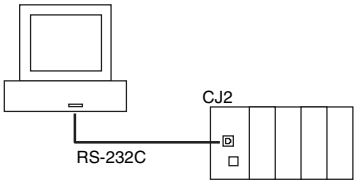
EtherNet/IP 节点连接

EtherNet/IP 节点连接用于访问已经通过 EtherNet/IP 建立连接的 PLC。即使不知道目标 PLC 的 IP 地址，也能通过搜索来访问 PLC。

11-1-2 可访问 PLC 的系统配置

与 PLC 的直接串行连接

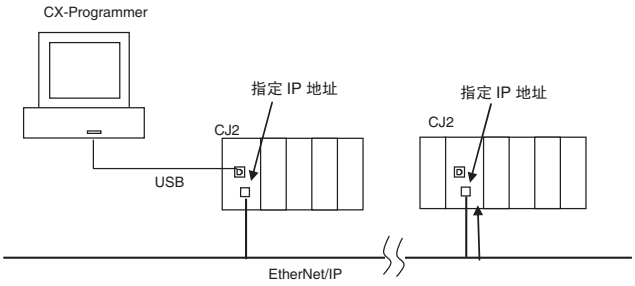
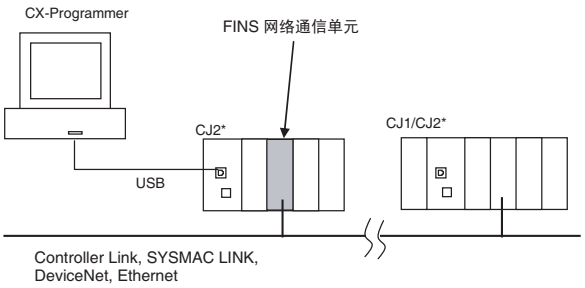
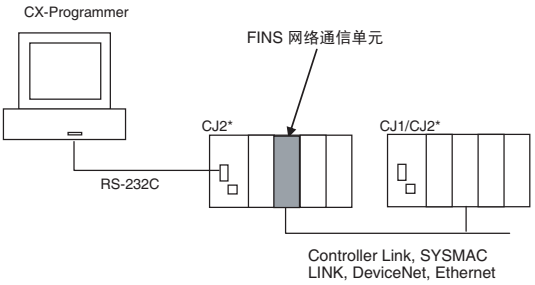
选择下表中给出的网络类型。

连接电缆	系统配置	改变 PLC 对话框（网络类型）	自动在线连接	
			直接连接	EtherNet/IP 连接
USB 电缆	CX-Programmer  	USB	可访问	不可访问
	访问 FINS 网络 CX-Programmer  * 必须设定本地网络表。	工具总线 （USB 端口）	可访问	不可访问
RS-232C 电缆	CX-Programmer  	工具总线	可访问	不可访问

在网络上访问 PLC

● 访问通过串行端口连接的 PLC

选择下表中给出的网络类型。

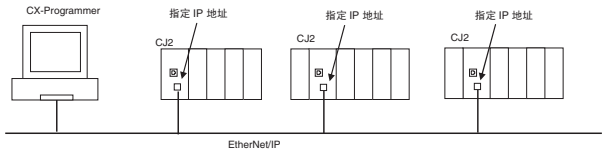
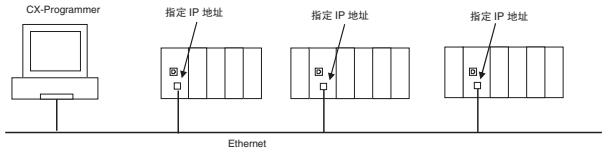
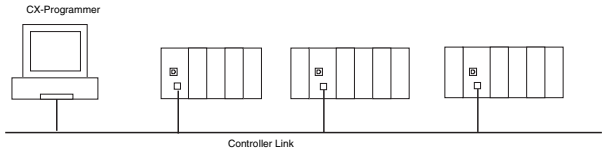
连接电缆	系统配置	改变 PLC 对话框（网络类型）	自动在线连接	
			直接连接	EtherNet/IP 连接
USB 电缆	通过 USB 访问 EtherNet/IP 网络 （*1） 	USB	不可访问	可访问
	通过 USB 访问 FINS 网络 （*2）  * 必须设定本地网络表。	工具总线（USB 端口）	不可访问	不可访问
RS-232C 电缆	通过 RS-232C 访问 FINS 网络 （*2）  * 必须设定本地网络表。	工具总线	不可访问	不可访问

*1 如果在对话框中设定了网络类型用于改变 PLC 型号设定，则只能在本地网络中连接 EtherNet/IP。不能跨网络层进行连接。如果 EtherNet/IP 连接设为自动在线连接，则可以进行跨网络层的连接。

*2 在 FINS 网络间可以进行跨越网络层的连接。

● 通过网络的连接

选择下表中给出的网络类型。

连接电缆	系统配置	改变 PLC 对话框（网络类型）	自动在线连接	
			直接连接	EtherNet/IP 连接
EtherNet/IP	访问 EtherNet/IP 网络 （*1）  必须设定连接目标的 IP 地址。	EtherNet/IP	不可访问	可访问
Ethernet	将 Ethernet 作为 FINS 网络访问 （*2）  必须设定连接目的地的 FINS 网络地址、节点地址和 IP 地址。	Ethernet，Ethernet（FINS/TOP）	不可访问	不可访问
Controller Link		Controller Link	不可访问	不可访问

*1 如果在对话框中设定了网络类型用于改变 PLC 型号设定，则只能在本地网络中连接 EtherNet/IP。不能跨网络层进行连接。如果 EtherNet/IP 连接设为自动在线连接，则可以进行跨网络层的连接。

*2 在 FINS 网络间可以进行跨越网络层的连接。

正确使用注意事项

通过内置 EtherNet/IP 端口与支持软件进行连接时，发生以下任一错误，连接不能进行。

I/O 总线错误

I/O 点数过多错误

单元编号重复错误

机架编号重复错误

I/O 设置错误

如果发生以上任一错误而不能与支持软件连接时，可通过 CPU 单元上的 USB 端口与支持软件进行连接。

正确使用注意事项

使用 EtherNet/IP 连接前的注意事项

Windows XP （SP2 及以上）和 Windows Vista 系统的更好的防火墙安全增加了 Ethernet 端口上数据通信的限制。从计算机上的 EtherNet 端口对以下的 PLC 之一使用 EtherNet/IP 连接 *1 时，必须改变 Windows 防火墙的设置才能启用 CX-Programmer 通信。

*1：EtherNet/IP 连接包括以下情况：

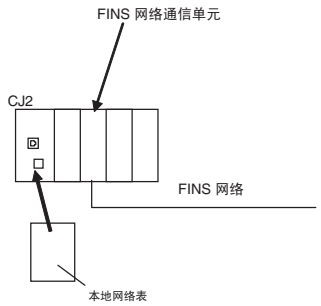
- 网络类型设为 EtherNet/IP 的在线连接
- 在 PLC 菜单中选择了“自动在线”-“EtherNet/IP 节点在线”时，自动在线连接到 EtherNet/IP 网络上的 PLC。

关于操作步骤，参考 A-6 通过 Windows XP （SP2 及以上）或 Windows Vista 连接 EtherNet/IP。

附加信息

必须在以下情况创建和传送路由表。

如果一台或以上的 FINS 网络通信单元（*）连接到 CJ2 CPU 单元，必须通过 CX-Integrator 为路由表创建和传送本地网络表，以确保通过与支持软件的连接能够访问内置 EtherNet/IP 网络和 FINS 网络。



* “FINS 网络通信单元”表示的是一个 Controller Link 单元、 SYSMAC LINK 单元、 Ethernet 单元、 DeviceNet 单元或 FL-net 单元，其中不包含 CompoNet 单元。此外，它还表示对路由表的串行通信使用了串行网关。

附加信息

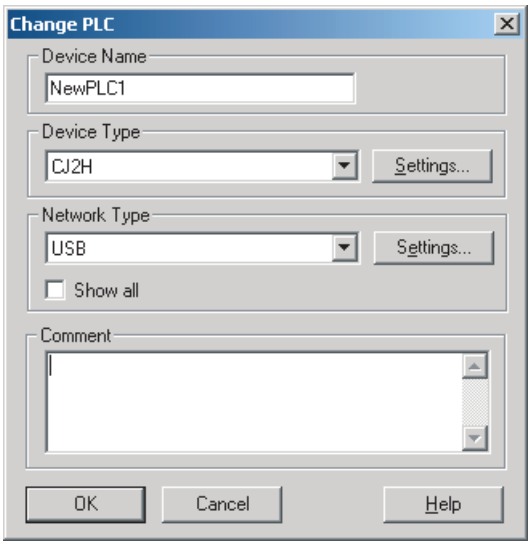
如果一个或以上的 FINS 网络通信单元连接到 CJ2 CPU 单元，创建并传送本地网络表。如果试图从网络地址设为 0 （默认值）的支持软件进行访问，而不创建和传送本地网络表，你将在以下情况中访问内置 EtherNet/IP 端口的节点地址。

如果内置 EtherNet/IP 端口的单元编号 （默认值：0）小于 FINS 通信单元的单元编号的情况。

11-1-3 从 CX-Programmer 访问 PLC

改变 PLC 对话框的步骤

创建新项目时，在 “Change PLC”（改变 PLC 对话框）中进行以下步骤来选择连接到 PLC 的方法。



1. 在 “设备类型 （Device Type）” 区域中选择 “CJ2H”。

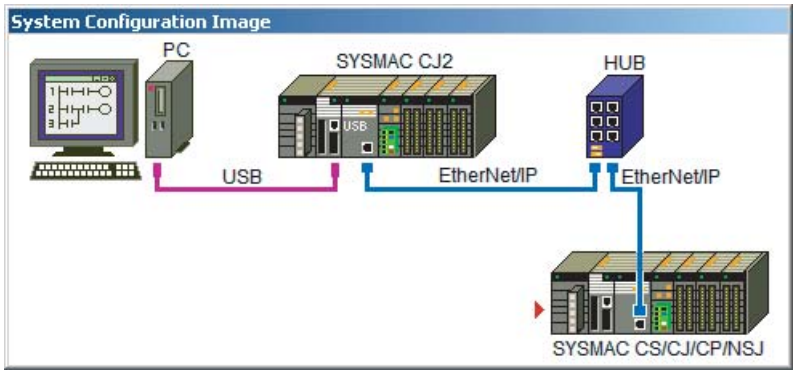
2. 在“网络类型（Network Type）”区域中选择连接到 PLC 的方法。可选择以下方法之一作为默认值。
   - USB
   - EtherNet/IP

   必要时，通过点击 “显示所有 （Show All）” 按钮，可显示其他网络类型。根据下表中的两个条件选择网络类型。

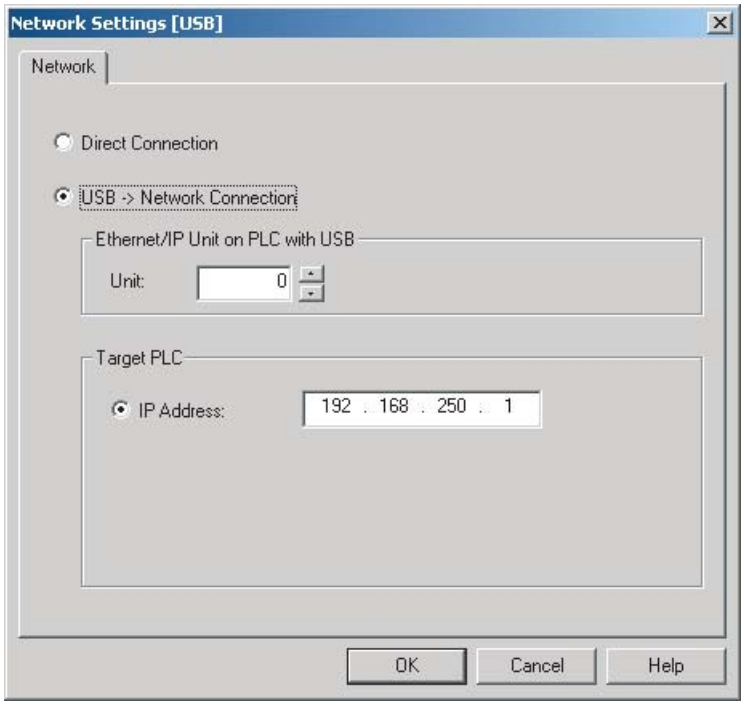
条件 1：连接到 PLC的电缆类型	条件 2	网络类型	
USB	---	USB	默认显示
Ethernet	访问 EtherNet/IP 网络	EtherNet/IP	
USB	通过 USB 访问 FINS 网络	工具总线（USB 端口）	点击 “显示所有” 按钮显示
Ethernet	将 Ethernet 网络作为 FINS 网络来访问	Ethernet	
		Ethernet （FINS/TCP）	
RS-232C	---	工具总线	
		SYSMAC WAY	
Controller Link （由 Controller Link 板连接）	---	Controller Link	
Fins 网关	---	Fins 网关	

3. 点击 “设置” 按钮，将显示所选网络类型的网络设定对话框。
   在本示例中，网络类型为默认显示，即说明了关于 USB 和 EtherNet/IP 的设定。关于其他网络类型设定的详细信息，参考 CX-Programmer 操作手册 （样本编号 W446）。

● 网络类型：USB



● 在网络标签页中选择 “USB-> 网络连接 （USB -> Network Connection）”



直接连接：

选择该选项来访问通过 USB 直接连接的 PLC。

USB -> 网络连接：

选择该选项通过 USB 端口访问 EtherNet/IP 网络 （单独网络）中的 PLC。进行以下设定：

通过 USB 连接 PLC 上的 EtherNet/IP 单元：

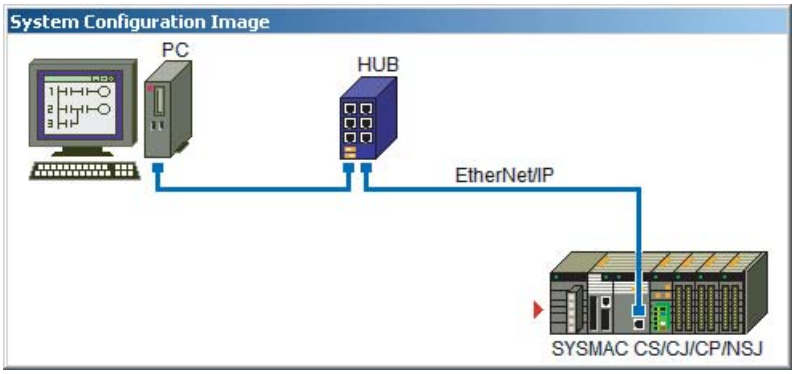
使用内置 EtherNet/IP 端口时，在单元区域设定 “F”。

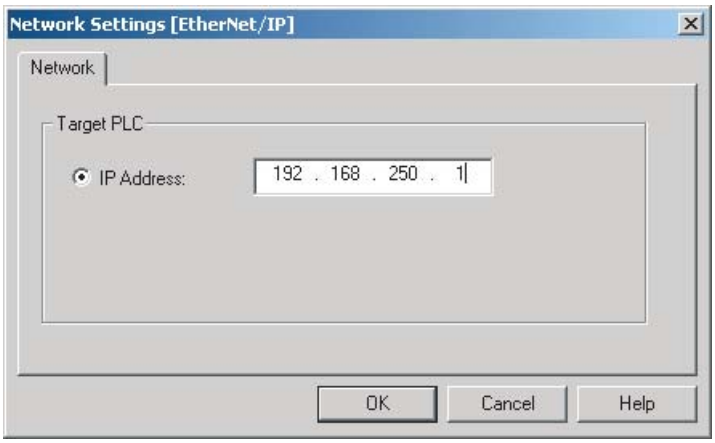
目标 PLC 的 IP 地址：

设定内置 EtherNet/IP 端口的 IP 地址。

默认设定是 “192.168.250. 节点 _ 地址”。

● 网络类型：EtherNet/IP





目标 PLC 的 IP 地址：

设定内置 EtherNet/IP 端口的 IP 地址。
默认设定是 “192.168.250. 节点 _ 地址”。

● 设定计算机的 IP 地址

通过 EtherNet/IP 连接到 PLC，计算机的 IP 地址必须设为符合内置 EtherNet/IP 端口的 IP 地址。例如，如果内置 EtherNet/IP 端口的 IP 地址设为默认值，则对计算机设定以下的 IP 地址。

例：192.168.250.2

最低位（上位 ID）的值是在 CPU 单元前部的内置 EtherNet/IP 端口（1 ～ 254）设定的除 FINS 节点地址以外的数值。

11-1-4 自动在线连接

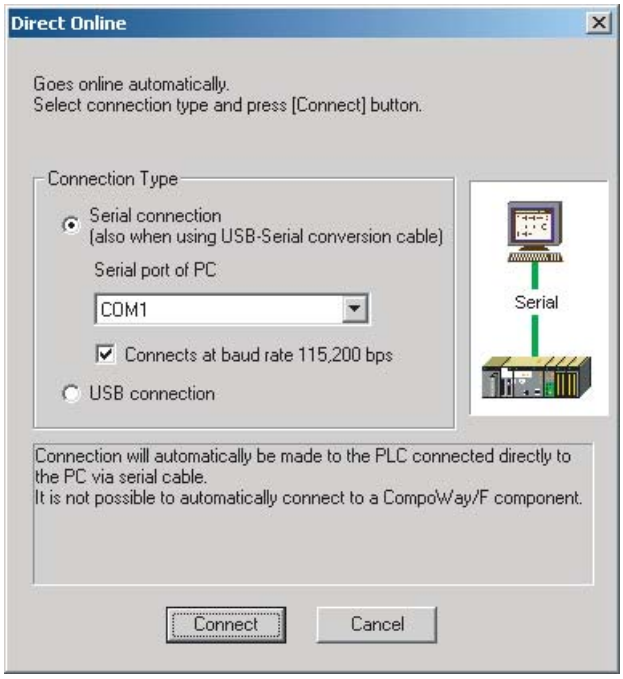
如果目标 PLC 的项目在计算机上不可用，可通过在线连接从 PLC 上传程序。

直接串行连接

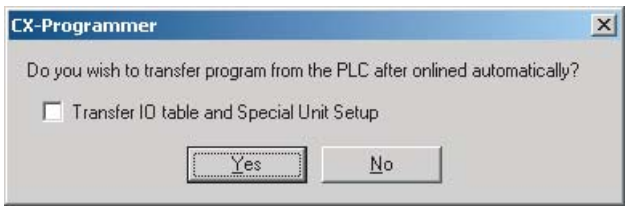
执行自动在线连接时，将对可使用的计算机串行端口进行自动搜索。当适用的串行端口被发现时，自动连接 CX-Programmer 到连接在串行端口上的 PLC。

1. 在 PLC 菜单中选择 “自动在线” - “直接在线”。直接在线对话框将显示如下。

2. 选择以下方法之一用于将计算机连接到 PLC。
   - 串行连接 （包含将计算机上的 USB 连接转换为 PLC 上的 RS-232C 连接）
   - USB 连接



从计算机的 USB 端口连接到 CJ2 CPU 单元的 USB 端口时，选择 “USB”连接，然后点击 “连接”按钮，将显示以下的对话框。



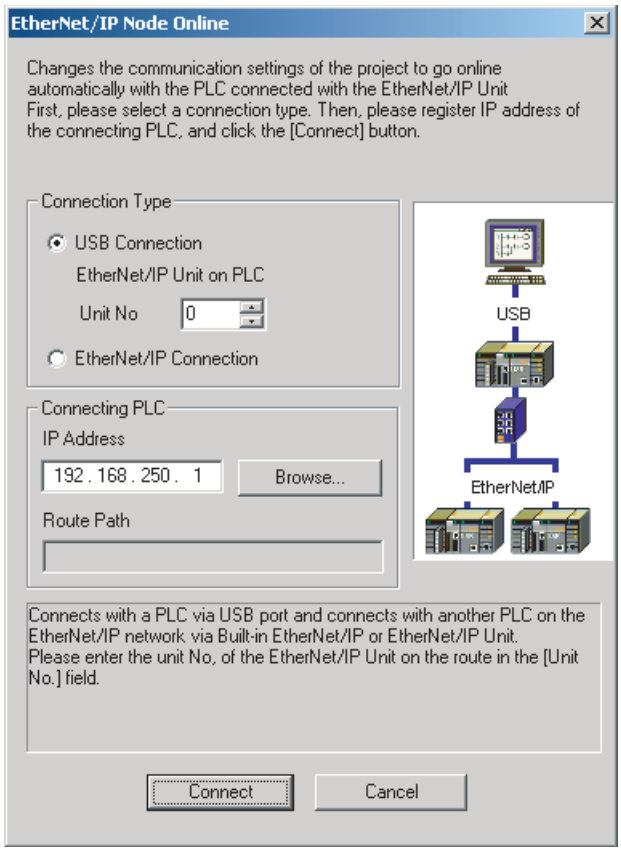
在线自动连接后，如果要将程序、 PLC 设置和 I/O 表上传到计算机，需点击 “Yes”按钮。 CX-Programmer 将自动连接到 PLC，并且这些项目将从 PLC 上传。

EtherNet/IP 节点连接

通过 EtherNet/IP 网络，连接到 EtherNet/IP 网络的 PLC 可被连接。如果不知道目标 PLC 的 IP 地址，可对其进行搜索来访问 PLC。

1. 在 PLC 菜单中选择 “自动在线” - “EtherNet/IP 节点在线”。

2. 将显示 EtherNet/IP 节点在线对话框。



3. 根据连接类型选择连接到目标 PLC 的方法。

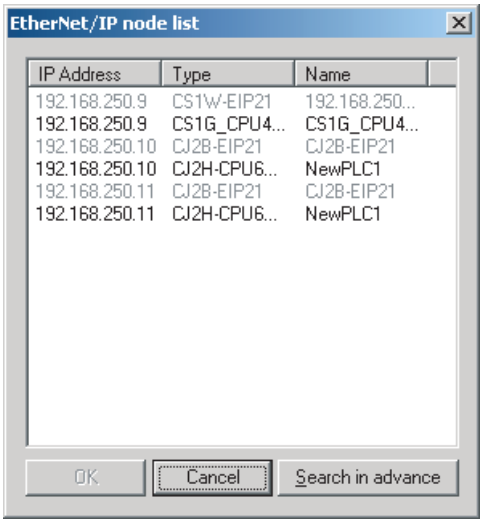
   - USB 连接
     选择 USB 连接来通过 USB 端口直接连接到 PLC，然后通过内置 EtherNet/IP 端口或 EtherNet/IP 单元，在线连接 EtherNet/IP 网络 （单独网络）上的 PLC。
   - EtherNet/IP 连接
     选择 EtherNet/IP 连接来将计算机连接到 EtherNet/IP 网络，并通过 EtherNet/IP 在网络上 （单独网络）在线连接 PLC。

4. 根据连接的 PLC，选择对目标 PLC 设定 IP 地址的方法。

   - IP 地址
     直接输入目标 PLC 的 IP 地址。
   - 浏览
     搜索在线连接到网络的 PLC，并选择目标 PLC。当不知道目标 PLC 的 IP 地址时，选择该方法。

连接的节点可作为列表或通过网络层显示。

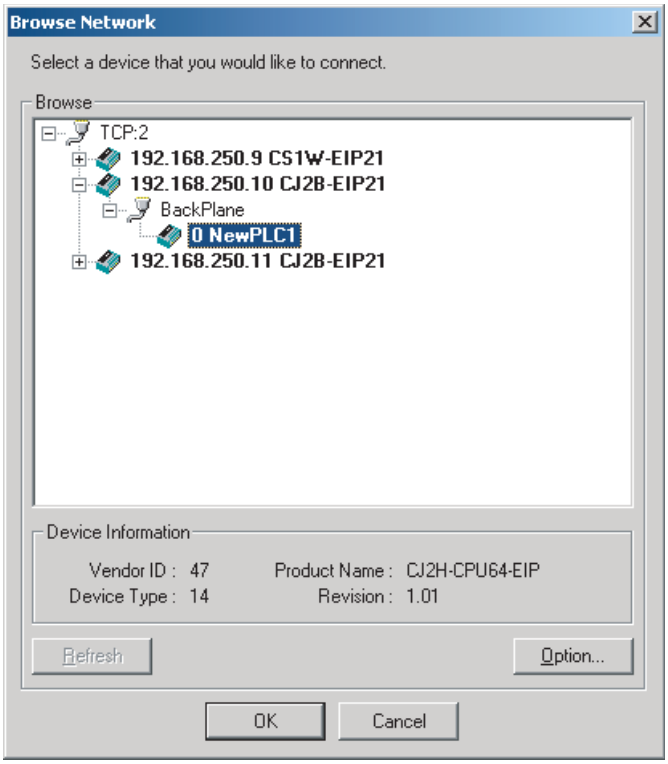
节点列表 （显示用于浏览）



- 将显示 IP 地址、设备类型和名称。选择 IP 地址并点击 “OK”按钮。

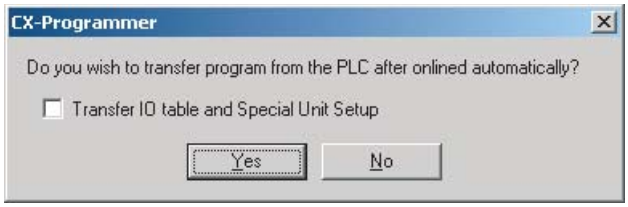
预先搜索

在 EtherNet/IP 节点列表对话框中点击 “预先搜索”按钮，显示以下对话框。



- 点击加号 （+）栏显示该项目下面的所有等级。选择要连接的 PLC，并点击 “OK”按钮。

5. 点击 “连接”按钮来进行在线连接。将显示以下对话框。



在线自动连接后，如果要将程序、 PLC 设置和 I/O 表上传到计算机，需点击 “Yes”按钮。计算机将自动连接到 PLC，并且这些项目将从 PLC 上传。

11-2 串行通信

11-2-1 串行通信概述

串行通信口模式（协议）可在 CPU 单元的 PLC 设置中切换，根据所选的协议配置下列系统。

协议

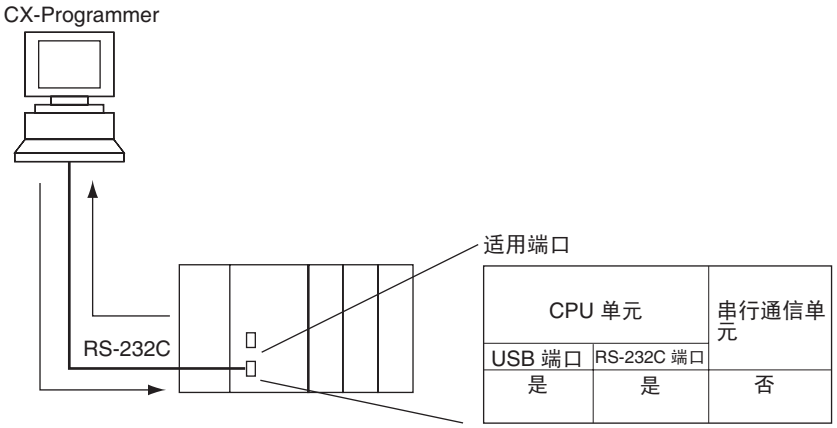
下列协议支持串行通信。

协议 （串行通信模式）	主连接	使用	适用的命令、通信指令
外设总线（工具总线） （仅 CPU 单元上的 RS-232C 端口。不支持串行通信单元。）	CX-Programmer	计算机的编程设备和 PLC 之间的通信。	无
上位链接（Host Link） （SYSMAC WAY）从站	计算机 欧姆龙可编程终端	上位计算机和 PLC 之间的通信。命令可从 PLC 发送到计算机。	上位链接（Host Link）命令 / FINS 命令，命令可从 PLC 发送到计算机。
无协议 *1	通用外部设备	通用设备的无协议通信。	TXD（236）指令、 RXD（235）指令、 TXDU（256）指令、RXDU（255）指令
NT 链接（1：N）	欧姆龙可编程终端	通过直接访问与可编程终端进行的高速通信。	无
串行网关	欧姆龙部件 PLC	PLC 将收到的 FINS 命令转换为 CompoWay/F、 Modbus 或上位链接（Host Link）协议，然后将转换后的命令传输至串行线 。	---
CompoWay/F 主站 *1	CompoWay/F 从站	将在串行端口收到的 FINS 命令（压缩的 CompoWay/F 命令）转换为 CompoWay/F 命令。	FINS 命令 收到 2803 hex（包括使用 CMND（490）发送 FINS 命令）
Modbus 主站 *1 （仅串行通信单元。不支持 CPU 单元上的 RS-232C 端口）。	Modbus 从站	将在串行端口收到的 FINS 命令（压缩的 Modbus 命令）转换为 Modbus 命令。	FINS 命令 收到 2804 hex 或 2805 hex（包括使用 CMND（490）发送 FINS 命令）
上位链接（Host Link）FINS（SYSWAY）主站 *1 （仅串行通信单元。不支持 CPU 单元上的 RS-232C 端口）。	上位链接（Host Link）FINS（SYSWAY）从站（PLC）	将 FINS 命令转换为上位链接（Host Link）中的压缩 FINS 命令。	除发送到串行端口以外的任何收到的 FINS 命令 （包括使用 CMND（490）发送 FINS 命令）
协议宏 （仅串行通信单元。不支持 CPU 单元上的 RS-232C 端口）。	通用外部设备	根据外部设备的通信规格发送和接收报文（通信帧）。 （SYSMAC-PST 用于通过设置各种参数创建协议）	PMCR（260）指令

*1 仅单元版本 1.2 及以上的串行通信单元。

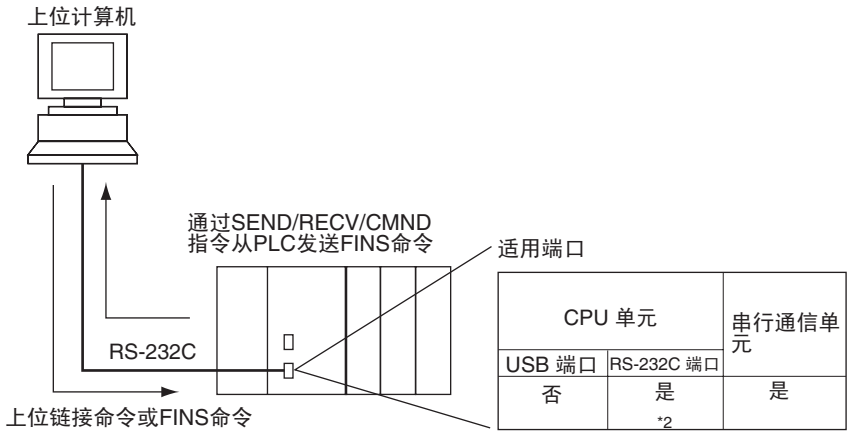
外设总线 （工具总线 ）

CX-One 支持软件，例如 CX-Programmer。



上位链接 （Host Link）系统从站 （SYSMAC WAY 模式 1：N）

上位链接（Host Link）系统允许读 / 写 PLC 的 I/O 存储器，通过执行上位链接（Host Link）命令或 FINS 命令从上位计算机 （计算机 *1 或可编程终端）修改操作模式，命令由帧头引导，并跟一个结尾。相反，FINS 命令（由帧头引导并跟一个结尾）可通过执行网络通信指令（SEND（090）/RECV（098）/CMND（490））从 PLC 发送到通过上位链接 （Host Link）系统连接的计算机。

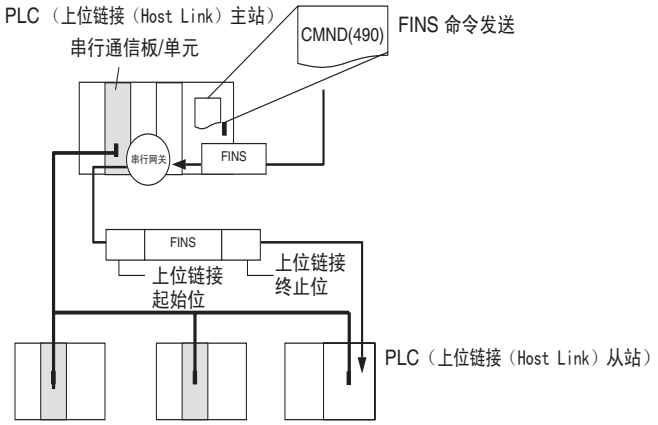


*1 CX-One 支持软件除外。

*2 把 CPU 单元前面板上的 DIP 开关位 5 设为 OFF，并把 PLC 设置中的串行通信模式设为上位链接（Host Link）。

附加信息

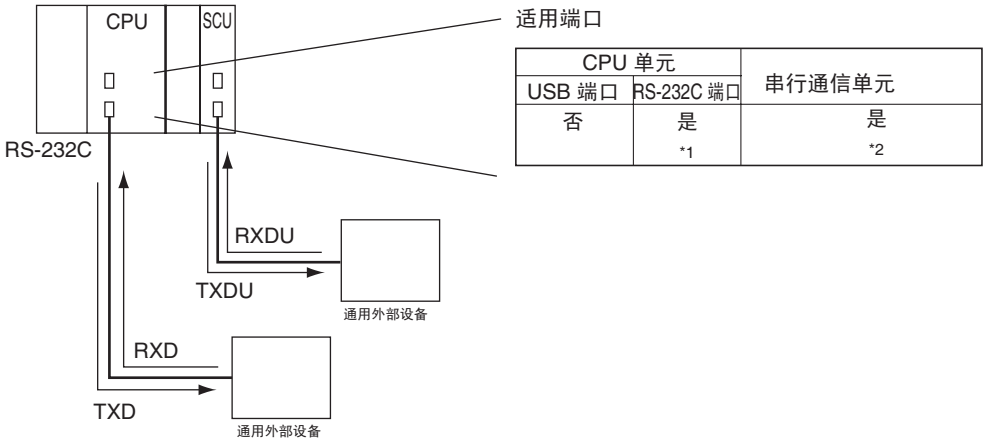
上位链接 （Host Link）主站功能可在使用单元版本 1.2 及以上的串行通信单元时，通过串行网关发送 CMND （490）指令来执行。



无协议通信系统

无协议通信允许简单的数据传输，例如用通信端口 I/O 指令输入条形码数据、输出打印机数据。可设置起始和完成代码，无协议通信也可用 RS 和 CS 信号控制。

下图根据使用的通信端口和数据传送 （发送和接收）的方向，显示了每个通信端口 I/O 指令的用途。



适用端口

CPU 单元		串行通信单元
USB 端口	RS-232C 端口	
否	是 *1	是 *2

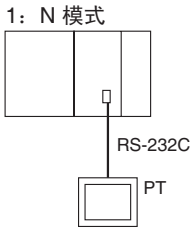
*1 把 CPU 单元前面板上 DIP 开关位 5 设为 OFF，并把 PLC 设置中的串行通信模式设为无协议通信。

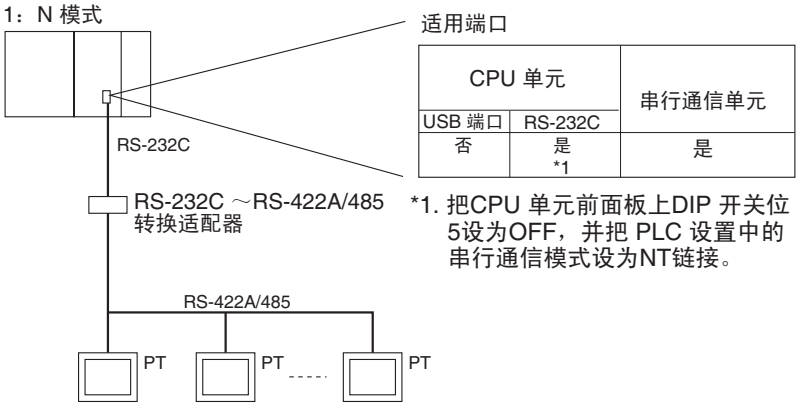
*2 仅单元版本 1.2 或更新的串行通信单元支持无协议通信。

NT 链接系统 （1：N 模式）

如果用 RS-232C 端口把 PLC 和可编程终端（PT）连在一起，可在 PLC 的 I/O 存储器中进行 PT 的状态控制区、状态通知区以及例如触摸开关、指示灯和内存图对象的分配。 NT 链接系统允许由 PLC 控制 PT，PT 可定期从 PLC 的状态控制区读数据，并在此区有改变时执行必要的操作。通过把数据从 PT 写到 PLC 的状态通知区， PT 可与 PLC 进行通信。NT 链接系统可控制和监视 PT 的状态，而无需使用 PLC 梯形图程序。 PLC 与 PT 之比是 1:N， N 大于等于 1。

PT 通信设置设为 1:N NT 链接，每个 PLC 可连接一个或多个 PT。





正确使用注意事项

- PLC 可连接到支持 1:N NT 链接的任意 PT 端口。PLC 不能连接到 NT30 或 NT30C 上的 RS-232C 端口，因为这些端口仅支持 1:1 NT 链接。
- 如果CPU 单元的循环时间为800ms或更长，不能使用NT20S、NT600S、NT30、NT30C、NT620S、NT620C 和 NT625C （即使仅连接其中的一个 PT）。
- 仅在PT 连接到CPU 单元上的RS-232C时可使用PT 的手持编程器功能（扩展功能）。连接到串行通信单元的 RS-232C 或 RS-422A/485 端口时不能使用此功能。
- 不能同时使用 PT 完成手持编程器功能和 PT 完成标准 PT 功能。
- 一个以上 PT 连接到同一个 PLC 时，确认每一个 PT 分配了唯一的单元编号。如果一个以上 PT 设置了相同的单元编号，将发生故障。

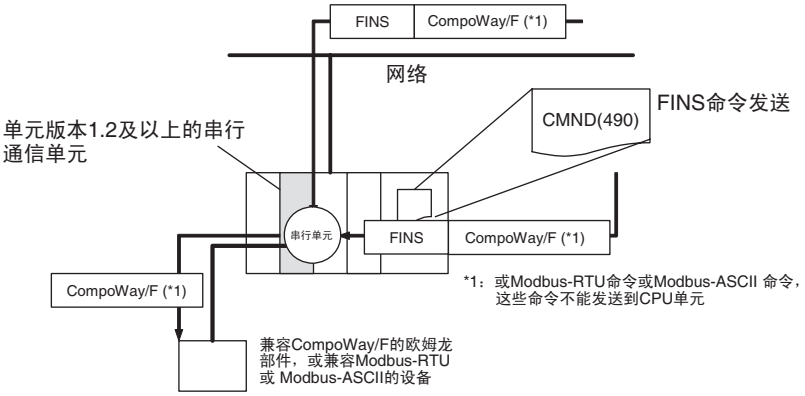
附加信息

1:1 和 1:N NT 链接协议不互相兼容，即它们是独立的串行通信协议。

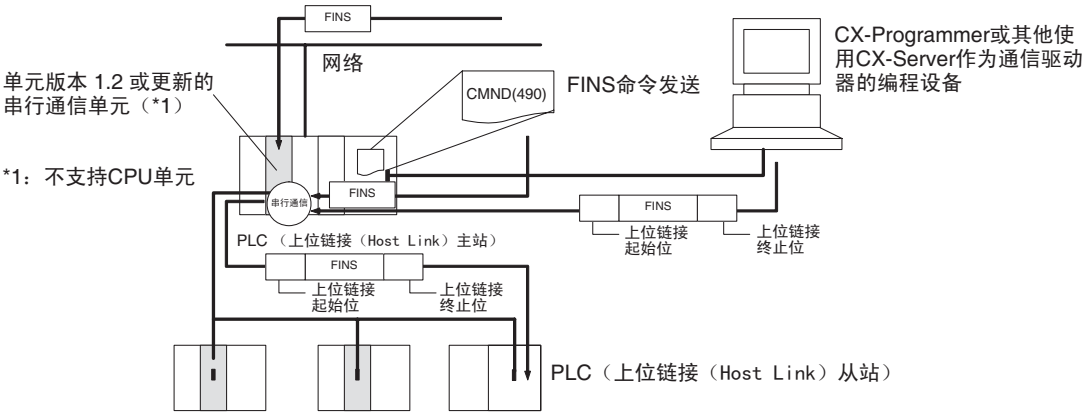
串行网关模式

收到的 FINS 报文会根据报文自动转换为 CompoWay/F。当使用了单元版本 1.2 及以上的串行通信单元时，收到的 FINS 报文会根据报文自动转换为 CompoWay/F、Modbus-RTU、Modbus-ASCII 或上位链接（Host Link） FINS 之一。

● CompoWay/F， Modbus-RTU， Modbus-ASCII



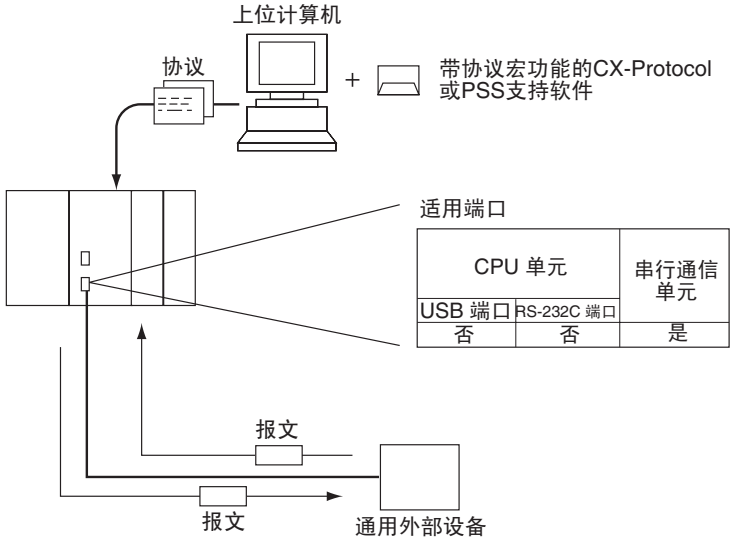
● 上位链接 （Host Link） FINS



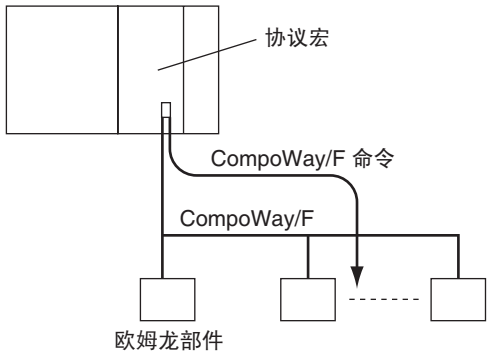
协议宏 （仅串行通信单元）

CX-Protocol 用于创建根据通用外部设备的通信定义（半双工或全双工，异步）进行的通用外部设备的数据传输步骤 （协议），所创建的协议记录在串行通信单元中，在 CPU 单元简单地执行 PMCR （260）指令就能对外部设备发送和接收数据。欧姆龙设备，比如温控器、智能信号处理器、条形码阅读器和调制解调器的数据通信协议，作为标准协议被支持 （*1）。用户也可根据应用需要改变这些协议。

*1 CX-Protocol 和串行通信单元提供标准协议。



注　CompoWay/F （上位功能）
CJ 系列 CPU 单元可作为主机向连接在系统中的欧姆龙部件传送 CompoWay/F 命令，使用协议宏功能标准协议中 CompoWay/F 发送 / 接收的顺序来执行 CompoWay/F 命令。



单元 / 协议兼容性

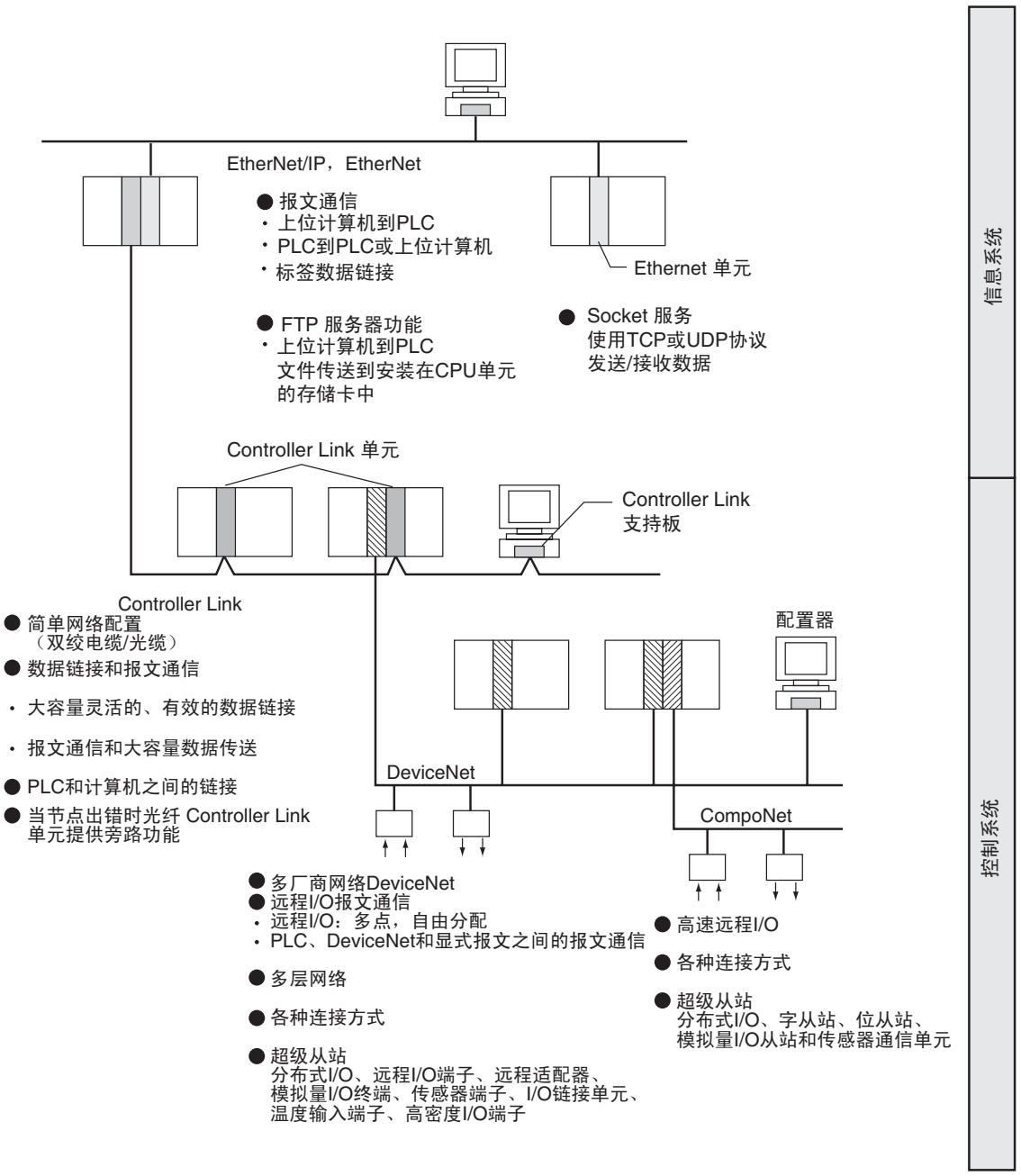
单元	型号	端口	协议 （串行通信模式）					
			外设总线（工具总线）	上位链接（Host Link）（除 CX-One 以外）	无协议通信	协议宏	NT 链接（1:N 模式）	串行网关（*1）
CPU 单元	CJ2H-CPU68-EIP CJ2H-CPU67-EIP CJ2H-CPU66-EIP CJ2H-CPU65-EIP CJ2H-CPU64-EIP	RS-232C	是	是	是	---	是	是
串行通信单元	CJ1W-SCU41-V1 CJ1W-SCU31-V1 CJ1W-SCU21-V1	RS422/485	---	是	是 （*1）	是	是	是
		RS-232C	---	是	是 （*1）	是	是	是

*1　仅单元版本 1.2 或更新的串行通信单元才支持。但是对于 CPU 单元，只有自动转换为 CompoWay/F 可对串行网关协议使用。

11-3 通信网络

通信网络配置

使用 CJ 系列 PLC 时可配置下列网络。

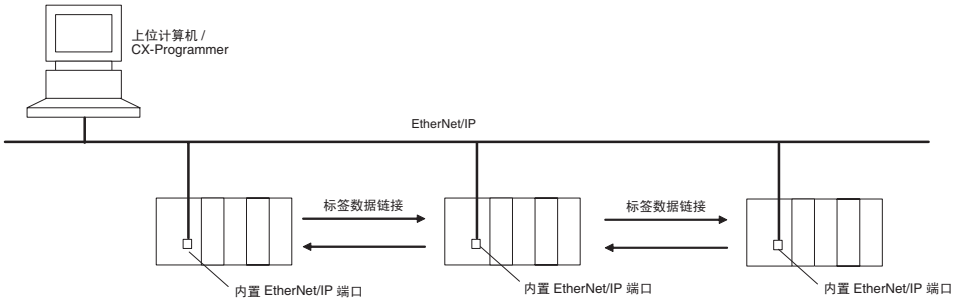


● EtherNet/IP 和 Ethernet

可通过 Ethernet 电缆进行标签数据链接和 FINS 网络通信。此外，从连接到 Ethernet 的上位计算机执行 PLC 的 FTP 命令，可读或写安装在 CPU 单元存储卡上的文件的内容。可用 UDP 和 TCP 协议发送和接收数据。这些功能使上位信息网络具有更强的兼容性。

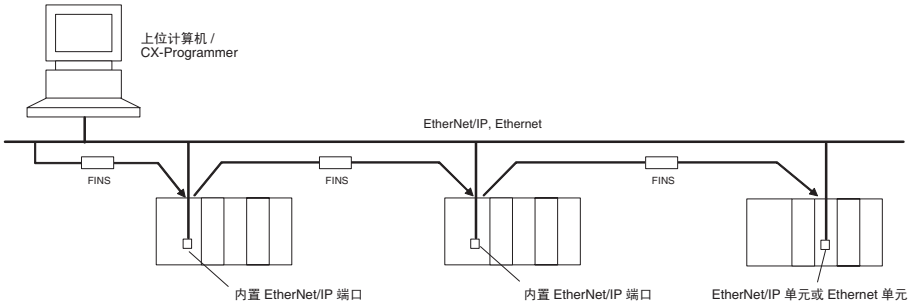
标签数据链接

数据链接可在 PLC 与 PLC 之间以用户自定义每个区的刷新周期进行高速、大容量的数据传送。



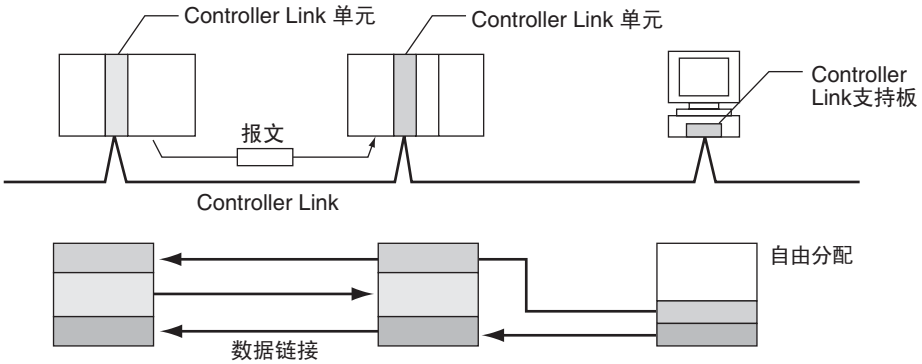
FINS 网络通信

连接到内置 EtherNet/IP 端口，或者是连接了 EtherNet/IP 单元或 Ethernet 单元的 CPU 单元，为允许与其他 PLC 或 PLC 与 Ethernet 网络中上位计算机之间的 FINS 报文通信。



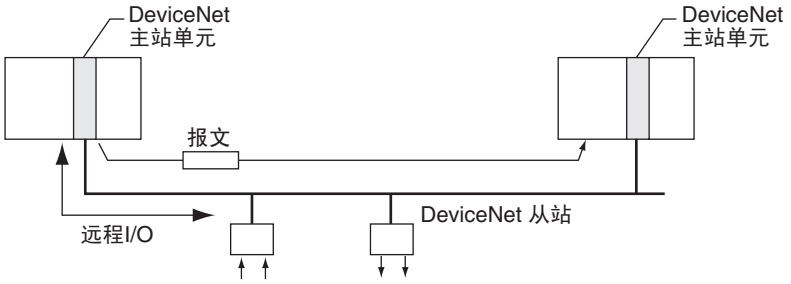
● Controller Link

Controller Link 网络是 OMRON PLC FA 的特殊网络。Controller Link 单元连接到允许建立 PLC 之间的数据链接的网络，以便不经过编程就可共享数据，并在需要时可在允许独立控制和数据传送的 PLC 之间进行 FINS 报文通信。Controller Link 网络连接使用双绞电缆或光缆。PLC 和计算机之间也可进行数据链接和报文通信，数据链接允许大容量和自由分配。 FINS 报文通信也允许大容量数据传送。



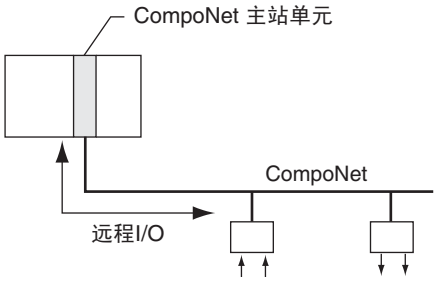
● DeviceNet

DeviceNet 是由多位控制和信息系统组成的多厂商网络，符合开放现场 DeviceNet 规格。连接 DeviceNet 主站单元到网络上实现网络上的 PLC 和从站之间能够进行远程 I/O 通信。远程 I/O 通信允许大容量 I/O 和用户自定义分配。模拟量 I/O 终端也可用于从站。PLC 之间和 PLC 与其它公司生产的 DeviceNet 设备之间可以进行报文通信。



● CompoNet

CompoNet 是主要用于 I/O 控制的高速、多点、多节点、多厂商网络。在 PLC 中连接 CompoNet 主站单元允许 PLC 和 CompoNet 从站之间的远程 I/O 通信。在最多 1 ms 循环时间内进行 1,024 点高速通信。共有 2,560 点和 384 节点可被连接。



通信网络概述

系统	网络	功能	通信	通信设备
信息网络	EtherNet/IP	上位计算机和 PLC 之间	FINS 报文通信	内置 EtherNet/IP 端口
		PLC 之间		
		上位计算机和安装在 CPU 单元内的存储卡之间	FTP 服务器	
	Controller Link	PLC 和直接连接到网络上的计算机之间	FINS 报文通信	Controller Link 支持板或 Controller Link 单元
			数据链接（偏移，简单设置）	
	RS-232C → Controller Link	上位链接（Host Link）计算机和网络上的 PLC 之间	上位链接（Host Link）命令和网关	RS-232C 电缆和 Controller Link 单元
控制网络	EtherNet/IP	PLC 之间	标签数据链接	内置 EtherNet/IP 端口
	Controller Link	PLC 之间	FINS 报文通信	Controller Link 单元
			数据链接（偏移，简单设置）	
	DeviceNet		开放网络中的 FINS 报文通信	DeviceNet 主站单元和配置器
	DeviceNet	PLC 和网络设备（从站）	开放网络中的大容量远程 I/O（固定或自由分配）	DeviceNet 主站单元和配置器
	CompoNet		开放网络中的高速、多点、多节点远程 I/O	CompoNet 主站单元

12

CPU 单元循环时间

本章描述了用于在 CPU 单元中进行处理的循环时间。

12-1 监控循环时间 12-2
12-1-1 监控循环时间 12-2
12-2 计算循环时间 12-3
12-2-1 CPU 单元操作流程图 12-3
12-2-2 循环时间概述 12-4
12-2-3 各单元的 I/O 单元刷新时间 12-6
12-2-4 循环时间计算实例 12-8
12-2-5 在线编辑导致的循环时间延长 12-10
12-2-6 I/O 响应时间 12-10

12-1 监控循环时间

12-1-1 监控循环时间

当 CX-Programmer 在线连接到 CPU 单元时，平均、最大和最小循环时间可被监控。

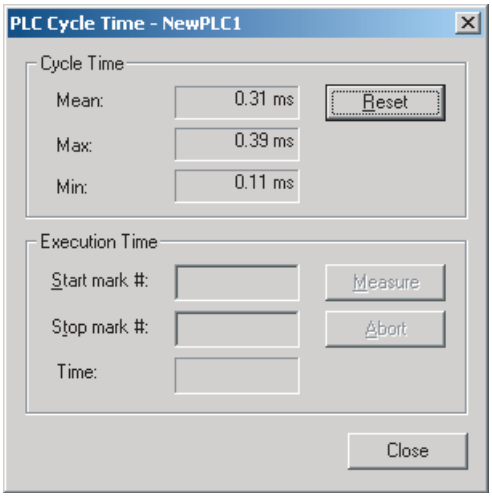
监控平均值

在线连接到 PLC 时，当 CPU 单元在除 PROGRAM 模式以外的任何模式时，将在状态栏中显示平均循环时间。



监控最大和最小值

在 PLC 菜单中选择 “编辑” - “循环时间”。将显示以下的 PLC 循环时间对话框。



平均 （平均值）、最大和最小循环时间从顶部开始依次显示。点击 “复位” 按钮来重新计算和显示循环时间的值。

附加信息

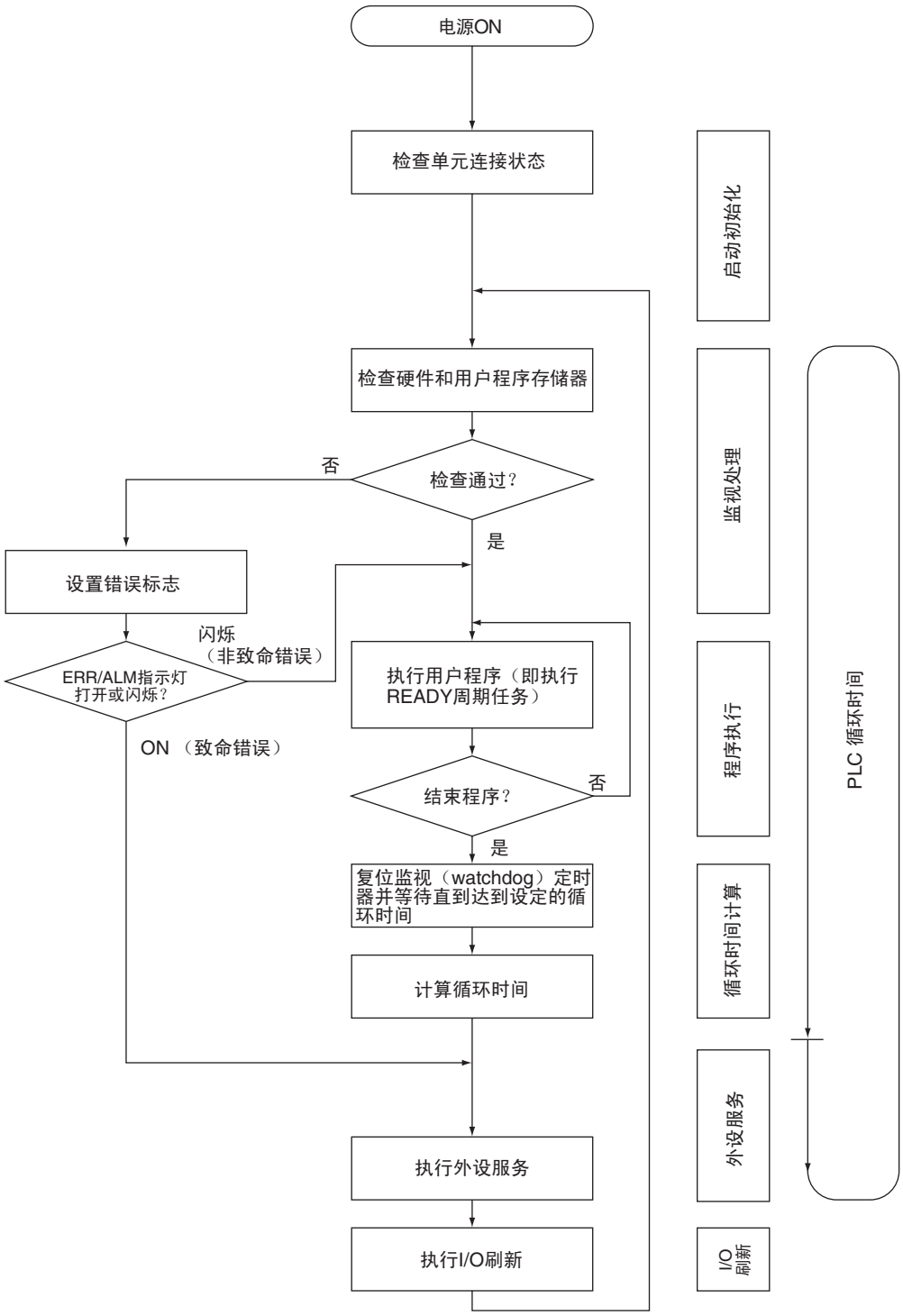
循环时间平均值 （= 当前值）和最大值保存在以下辅助区的字中。

- 当前循环时间 （= 当前值）（单位：0.1 ms）
  A264 （低字节）和 A265 （高字节）
- 当前循环时间 （= 当前值）（单位：0.01 ms）
  A266 （低字节）和 A267 （高字节）
- 最大循环时间 （单位：0.01 ms）
  A262 （低字节）和 A263 （高字节）

12-2 计算循环时间

12-2-1 CPU 单元操作流程图

在下列流程图中可以看到 CJ 系列 CPU 单元从监视处理到 I/O 刷新的重复循环中的处理数据。



12-2-2 循环时间概述

循环时间取决于下列条件。

- 用户程序中的指令类型和数目（在一个循环内执行所有的循环任务，以及满足执行条件的中断任务）。
- 基本 I/O 单元的类型和数目。
- 特殊 I/O 单元和 CPU 总线单元的类型和数目。
- 对以下单元 / 板的特定服务
  - EtherNet/IP 标签数据链接刷新和交换的标签数据链接字的数目。
  - 数据链接的刷新和 Controller Link 的数据链接字的数目。
  - DeviceNet 单元的远程 I/O 和远程 I/O 字的数量。
  - 协议宏的使用和最大通信报文。
  - Ethernet 单元的特殊控制位的 Socket 服务和发送 / 接收字的数目。
- 在 PLC 设置中固定的循环时间设定。
- 对文件存储器中的文件存取，以及进出文件存储器传输的数据量。
- 对于特殊 I/O 单元、 CPU 总线单元，和通信端口的事件服务。
- 内置 USB 端口和 RS-232C 端口的使用。
- 在 PLC 设置中设定的固定的外设服务执行时间。

注 1 循环时间不受用户程序内所使用的任务数量多少的影响。影响循环时间的是那些在循环中处于 READY 状态的循环任务。

2 当运行模式从 MONITOR 切换到 RUN 时，循环时间将延长 10 ms （然而，这将不会导致循环时间超出限定值）。

循环时间是 PLC 在执行下列 5 种操作所需的总时间。

循环时间 = （1） + （2） + （3） + （4） + （5）

1. 监视

详细说明	处理时间和起伏原因
检查 I/O 总线和用户程序存储器，检查电池错误并刷新时钟。	CJ2 CPU 单元：0.20 ms

2. 程序执行

详细说明	处理时间和起伏原因
执行用户程序，并计算执行程序指令所耗的总时间。	总的指令执行时间

3. 循环时间计算

详细说明	处理时间和起伏原因
在 PLC 设置中设定了最小 （固定的）循环时间时，等待到达指定的循环时间。 计算循环时间。	若未固定循环时间，则此步骤 3 所需的时间约为 0。 循环时间固定时，步骤 3 所需时间为预设的固定循环时间减去实际循环时间 （（1） + （2） + （4） + （5））。

4. I/O 刷新

详细说明			处理时间和起伏原因
基本 I/O 单元	刷新基本 I/O 单元，首先对每个单元的从 CPU 单元到输出单元的输出进行刷新，然后刷新从输入单元到 CPU 单元的输入。		每个单元的 I/O 刷新时间乘以所使用的单元数量。
特殊 I/O 单元	在 CIO 区中分配的字		每个单元的 I/O 刷新时间乘以所使用的单元数量。
	单元特定的数据	例：CompoBus/S 远程 I/O	
CPU 总线单元	在 CIO 和 DM 区中分配的字		每个单元的 I/O 刷新时间乘以所使用的单元数量。
	单元特定的数据	例： • EtherNet/IP 标签数据链接刷新 • Controller Link 数据链接刷新 • DeviceNet 远程 I/O 通信 • 协议宏的发送 / 接收数据 • Ethernet 单元的特殊控制位的 Socket 服务	

5. 外设服务

支持以下外围设备服务。

- 通信端口的服务 （内部逻辑端口）
  当使用梯形图程序中的通信指令或者类似后台操作或简捷备份的功能时，用于通信端口的服务被执行。
- 特殊 I/O 单元或 CPU 总线单元的事件服务
  无论何时当单元连接到 PLC 时，事件服务被执行。在通过通信单元使用支持软件的场合也会被执行。
- USB 端口服务
  使用 USB 端口时， USB 端口服务被执行。
- RS-232C 端口服务
  使用 RS-232C 端口时， RS-232C 端口服务被执行。
- 文件存取服务
  对存储卡或 EM 文件存储器进行存取时，文件存取服务被执行。
- 在线编辑服务
  执行在线编辑时，在线编辑服务被执行。
- 备份服务
  程序或参数改变时，备份服务被执行。

当外围设备服务没有执行时，不需要时间用于处理。外围设备循环的最小和最大处理时间取决于 PLC 设置中所有事件参数的设定时间，如下表所示。

		外设服务执行时间	
		最小	最大
所有事件参数的设定时间	禁止	0 ms （服务没有执行时）	循环时间的 10% （或当循环时间的 10% 小于 0.1 ms 时，为 0.1 ms）
	允许		设定时间 （0.1 ～ 3,276.7 ms）

12-2-3 各单元的 I/O 单元刷新时间

● 典型的基本 I/O 单元刷新时间

名称	型号	每单元 I/O 刷新时间
8/16 点 DC 输入单元	CJ1W-ID201/211	0.0014 ms
32 点 DC 输入单元	CJ1W-ID231/232	0.0023 ms
64 点 DC 输入单元	CJ1W-ID261/262	0.0041 ms
8/16 点 AC 输入单元	CJ1W-IA201/111	0.0014 ms
16 点中断输入单元	CJ1W-INT01	0.0014 ms
16 点 快速响应输入单元	CJ1W-IDP01	0.0014 ms
8/16 点继电器输出单元	CJ1W-OC201/211	0.0014 ms
8 点 晶闸管输出单元	CJ1W-OA201	0.0014 ms
8/16 点晶体管输出单元	CJ1W-OD201/202/203/204/211/212	0.0014 ms
32 点晶体管输出单元	CJ1W-OD231/232/233	0.0023 ms
64 点晶体管输出单元	CJ1W-OD261/262/263	0.0041 ms
DC 24V 输入 / 晶体管输出单元（16 输入 /16 输出）	CJ1W-MD231/232/233	0.0023 ms
DC 24V 输入 / 晶体管输出单元（32 输入 /32 输出）	CJ1W-MD261/263	0.0041 ms
TTL 输入 /TTL 输出单元（16 输入 /16 输出）	CJ1W-MD563	0.0041 ms
B7A 接口单元 （64 输入）	CJ1W-B7A14	0.0041 ms
B7A 接口单元 （64 输出）	CJ1W-B7A04	0.0041 ms
B7A 接口单元（32 输入 /32 输出）	CJ1W-B7A22	0.0041 ms

● 典型的特殊 I/O 单元刷新时间

名称	型号	每单元 I/O 刷新时间
模拟量输入单元	CJ1W-AD041/081 （V1）	0.05 ms
模拟量输出单元	CJ1W-DA021/041/08V	0.05 ms
模拟量 I/O 单元	CJ1W-MAD42	0.05 ms
温度控制单元	CJ1W-TC □□□	0.05 ms
位置控制单元	CJ1W-NC113/133	0.12 ms
		用于传送数据的每条指令 （IOWR/IORD） + 0.8 ms
	CJ1W-NC213/233	0.14 ms
		用于传送数据的每条指令 （IOWR/IORD） + 0.8 ms
	CJ1W-NC413/433	0.17 ms
		用于传送数据的每条指令 （IOWR/IORD） + 0.7 ms
ID 传感器单元	CJ1W-V600C11	0.12 ms
	CJ1W-V600C12	0.14 ms
高速计数器单元	CJ1W-CT021	0.10 ms

名称	型号		每单元 I/O 刷新时间
CompoNet 主站单元	CJ1W-CRM21	通信模式编号 0	0.13 ms
		通信模式编号 1	0.14 ms
		通信模式编号 2	0.17 ms
		通信模式编号 3	0.21 ms
		通信模式编号 8（*1）	0.081 + （0.0012 ×分配字的编号） ms
CompoBus/S 主站单元	CJ1W-SRM21	指定 1 个单元编号	0.08 ms
		指定 2 个单元编号	0.08 ms

*1 分配字的编号为实际分配给连接从站的 I/O 存储器中字的实际编号。

● 由 CPU 总线单元引起循环时间的增加

名称	型号	增量	备注
Controller Link 单元	CJ1W-CLK21（V1）	0.19 ms	• 将增加 .0.8 μs ×数据链接字的数目 • 在使用报文服务时，将增加事件执行时间。
串行通信单元	CJ1W-SCU41-V1 CJ1W-SCU21-V1 CJ1W-SCU31-V1	0.20 ms	当执行协议宏时将增加下列时间： 0.4 μs ×最大的发送和接收字的数目 （0 ～ 500 字） 当使用上位链接 （Host Link）或 1:N NT 链接时，将增加事件执行时间。
Ethernet 单元	CJ1W-ENT11/21	0.1 ms	若以软件开关执行 Socket 服务，将增加 1.4 μs ×发送 / 接收的字节数。 当执行 FINS 通信服务、 CMND 指令的 Socket 服务或 FTP 服务时，将增加事件执行时间。
EtherNet/IP 单元	CJ1W-EIP21	0.1 ms	使用标签数据链接时增加以下的值。 传送字的数目× 0.33 μs 若使用报文服务，事件执行时间也必须增加。
FL-net 单元	CJ1W-FLN22	0.25 ms + 数据链接字的数目 × 1.6 μs	数据链接字数是由节点发送和接收的字数。若使用报文服务，事件执行时间必须被分别增加。
DeviceNet 单元	CJ1W-DRM21	0.4 ms + 每个分配的字 0.7 μs	包括所有分配给从站的字，包括未使用的从站的字。 对于报文通信要把通信字的数目加到左边的算式中。
带 MECHATROLINK-II 通信的位置控制单元	CJ1W-NCF71	根据连接到单元的服务驱动器轴的数目 1 轴：0.2 ms， 3 轴：0.3 ms， 6 轴：0.4 ms，16 轴：0.8 ms	---
带 MECHATROLINK-II 通信的运动控制单元	CJ1W-MCH71	0.2 + （运动任务的数目 × 4 + 轴数 × 5 + 对通用 I/O 分配字的数目）× 0.001（ms）	---
SYSMAC SPU 单元	CJ1W-SPU01-V2	0.2 ms + 采样字的数目 × 0.8 μs	---

12-2-4 循环时间计算实例

● 例 1：基于基本指令和基本 I/O 单元的应用

下例显示了当 PLC 中只连接了基本 I/O 单元时用于计算循环时间的方法，程序由基本和数据传送指令的 20K 步组成。此处使用了 CJ2H-CPU6 □ -EIP CPU 单元。

条件

项目	详细说明	
CPU 单元	CJ2H-CPU6 □ -EIP	
CJ 系列 CPU 机架	CJ1W-ID261 64 点输入单元	2 单元
	CJ1W-OD261 64 点输出单元	2 单元
用户程序	20 K 步	LD 指令：10 K 步 MOV 指令：10 K 步 （LD：每个指令 1 步 MOV：每个指令 3 步）
内置 USB 端口连接	是或否	
固定循环时间处理	否	
RS-232C 端口连接	否	
其他设备 （特殊 I/O 单元、 CPU 总线单元和文件存取）的外设服务	否	

计算实例

处理名称	计算	处理时间	
		编程设备连接到内置 USB 端口	编程设备不连接到内置 USB 端口
（1）监视	---	0.20 ms	0.20 ms
（2）执行程序	0.016 μs × 10,000 + 0.14 μs/ 3 步 × 10,000	0.63 ms	0.63 ms
（3）最小循环时间的循环时间计算	（未设置固定循环时间）	0 ms	0 ms
（4） I/O 刷新	0.0039 ms × 2 + 0.0039 ms × 2	0.0164 ms	0.0164 ms
（5）外设服务	（仅内置 USB 端口连接）	0.1 ms	0 ms
循环时间	(1)+ (2)+ (3)+ (4)+ (5)	0.95 ms	0.85 ms

● 例 2：包含计算和特殊 I/O 单元的应用

下例显示了当 PLC 中只连接了基本 I/O 单元和特殊 I/O 单元时用于计算循环时间的方法，程序由基本指令、数据传送指令和浮点计算指令的 20K 步组成。此处使用了 CJ2H-CPU6 □ -EIP CPU 单元。

条件

项目	详细说明	
CPU 单元	CJ2H-CPU6 □ -EIP	
CJ 系列 CPU 机架	CJ1W-ID261 64 点输入单元	2 单元
	CJ1W-OD261 64 点输出单元	2 单元
	CJ1W-AD081 模拟量输入单元	2 单元
	CJ1W-NC413 位置控制单元	2 单元

项目	详细说明	
用户程序	20 K 步	LD 指令：12 K 步 MOV 指令：6 K 步 +F 指令：2K 步 （LD：每个指令 1 步 MOV：每个指令 3 步 +F：每个指令 4 步）
内置 USB 端口连接	是或否	
固定循环时间处理	否	
RS-232C 端口连接	否	
其他设备 （特殊 I/O 单元、 CPU 总线单元和文件存取）的外设服务	否	

计算实例

处理名称	计算	处理时间	
		编程设备连接到内置 USB 端口	编程设备不连接到内置 USB 端口
（1）监视	---	0.20 ms	0.20 ms
（2）执行程序	0.016 ms × 12,000 + 0.14 μs/ 3 步 × 6,000 + 0.24 μs/ 4 步 × 2,000	0.59 ms	0.59 ms
（3）最小循环时间的循环时间计算	（未设置固定循环时间）	0 ms	0 ms
（4） I/O 刷新	0.0041 ms × 2 + 0.0041 ms × 2 + 0.05 ms × 2 + 0.17 ms × 2	0.4564 ms	0.4564 ms
（5）外设服务	（仅内置 USB 端口连接）	0.1 ms	0 ms
循环时间	(1)+ (2)+ (3)+ (4)+ (5)	1.35 ms	1.25 ms

12-2-5 在线编辑导致的循环时间延长

当 CPU 在 MONITOR 模式下运行时，从 CX-Programmer 执行在线编辑。由于需对 CPU 单元程序进行写处理，循环时间将延长约 1ms。如果不想要延长一定的循环时间，使用在线编辑禁止 （待机）功能。详情请参考 10-7-4 在线编辑。

12-2-6 I/O 响应时间

I/O 响应时间是指从输入单元的输入接通， CJ 系列 CPU 单元识别数据，至用户程序执行，一直到结果输出到输出单元的输出端子的这段时间。

I/O 响应时间的长度取决于以下几个条件。

- 输入位转 ON 的时间
- 循环时间
- 安装输入和输出单元的机架类型 （CPU 机架或扩展机架）

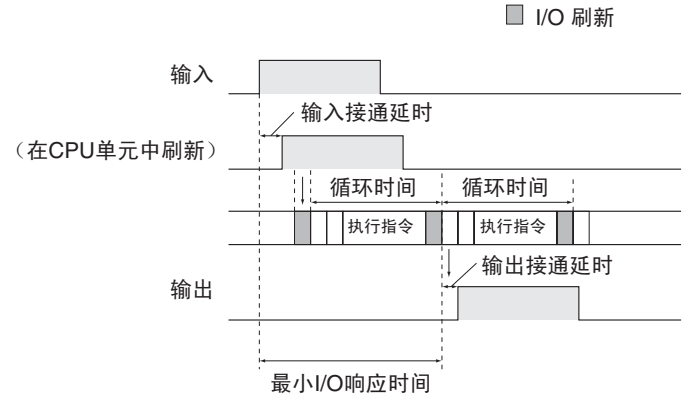
CJ 系列基本 I/O 单元响应时间

● 最小 I/O 响应时间

恰好在 CPU 单元 I/O 刷新之前取出数据，此时的 I/O 响应时间最小。

最小 I/O 响应时间是输入接通延时、循环时间和输出接通延时的总和。

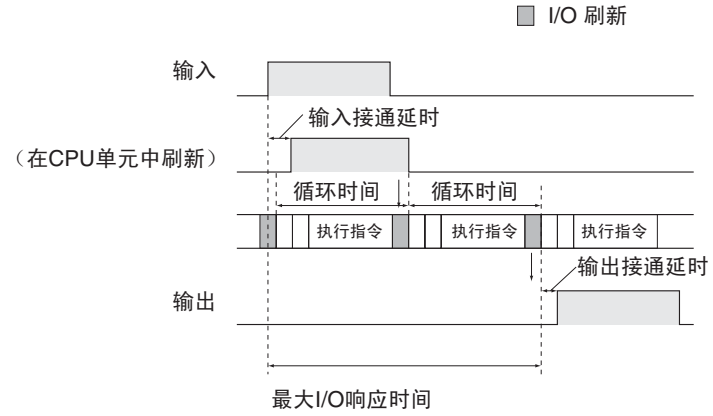
注　输入和输出接通时间的延迟根据所使用单元的不同而不同。



● 最大 I/O 响应时间

恰好在输入单元 I/O 刷新之后取出数据，此时的 I/O 响应时间最大。

最大 I/O 响应时间是指输入接通延时、（循环时间× 2）和输出接通延时的总和。



● 计算实例

条件：

输入接通延时：　1.5 ms
输出接通延时：　0.2 ms
循环时间：　20.0 ms

最小 I/O 响应时间 = 1.5 ms + 20 ms + 0.2 ms = 21.7 ms

最大 I/O 响应时间 = 1.5 ms + （20 ms × 2） + 0.2 ms = 41.7 ms

App

附录

附录中提供了关于电源中断时 CPU 单元运行、CPU 单元支持的指令、辅助区字和位、指令执行时间、PLC 存储器地址的实际有效存储器分配、以及 CJ 系列与 CS 系列 PLC 比较等的详细信息。

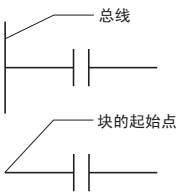
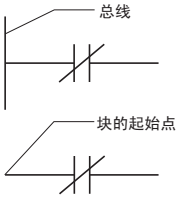
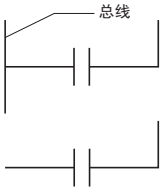
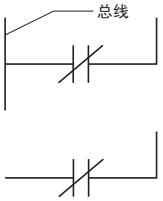
A-1 指令功能 A-3
A-1-1 顺序输入指令 A-3
A-1-2 顺序输出指令 A-5
A-1-3 顺序控制指令 A-6
A-1-4 定时器和计数器指令 A-10
A-1-5 比较指令 A-14
A-1-6 数据传送指令 A-17
A-1-7 数据移位指令 A-20
A-1-8 递增 / 递减指令 A-24
A-1-9 四则运算指令 A-24
A-1-10 转换指令 A-30
A-1-11 逻辑指令 A-36
A-1-12 特殊算术指令 A-37
A-1-13 浮点数运算指令 A-39
A-1-14 双精度浮点数指令 A-43
A-1-15 表格数据处理指令 A-46
A-1-16 跟踪指令 A-50
A-1-17 数据控制指令 A-51
A-1-18 子程序指令 A-54
A-1-19 中断控制指令 A-55
A-1-20 步指令 A-56
A-1-21 基本 I/O 单元指令 A-56
A-1-22 串行通信指令 A-59
A-1-23 网络指令 A-60
A-1-24 文件存储指令 A-63
A-1-25 显示指令 A-65
A-1-26 时钟指令 A-65
A-1-27 调试指令 A-66
A-1-28 故障诊断指令 A-66
A-1-29 其它指令 A-67
A-1-30 块程序指令 A-68
A-1-31 文本字符串处理指令 A-73

A-1-32 任务控制指令 A-76
A-1-33 型号转换指令 A-76
A-1-34 特殊功能块指令 A-77
A-2 指令执行时间和步数 A-78
A-2-1 顺序输入指令 A-79
A-2-2 顺序输出指令 A-79
A-2-3 顺序控制指令 A-80
A-2-4 定时器和计数器指令 A-81
A-2-5 比较指令 A-82
A-2-6 数据传送指令 A-83
A-2-7 数据移位指令 A-83
A-2-8 递增 / 递减指令 A-85
A-2-9 四则运算指令 A-85
A-2-10 转换指令 A-86
A-2-11 逻辑指令 A-89
A-2-12 特殊算术指令 A-89
A-2-13 浮点数运算指令 A-89
A-2-14 双精度浮点数指令 A-91
A-2-15 表格数据处理指令 A-92
A-2-16 跟踪指令 A-94
A-2-17 数据控制指令 A-94
A-2-18 子程序指令 A-95
A-2-19 中断控制指令 A-96
A-2-20 步指令 A-96
A-2-21 基本 I/O 单元指令 A-96
A-2-22 串行通信指令 A-97
A-2-23 网络指令 A-97
A-2-24 文件存储指令 A-98
A-2-25 显示指令 A-98
A-2-26 时钟指令 A-98
A-2-27 调试指令 A-98
A-2-28 故障诊断指令 A-99
A-2-29 其它指令 A-99
A-2-30 块程序指令 A-99
A-2-31 文本字符串处理指令 A-101
A-2-32 任务控制指令 A-102
A-2-33 型号转换指令 A-102
A-2-34 特殊功能块指令 A-102
A-2-35 SFC 指令 A-103
A-2-36 功能块实例执行时间 A-103
A-3 辅助区 A-105
A-3-1 只读区 （通过系统设置） A-105
A-3-2 读 / 写区 （ 由用户设置） A-124
A-3-3 辅助区操作细节 A-132
A-4 PLC 存储器地址的内存印象图 A-135
A-4-1 PLC 存储器地址 A-135
A-4-2 存储器分配 A-136
A-5 电源中断时操作 A-137
A-5-1 电源掉电操作 A-137
A-5-2 电源掉电的指令执行 A-139
A-6 通过 Windows XP （SP2 及以上）或 Windows Vista 连接 EtherNet/IP . . . A-141
A-6-1 变更 Windows 防火墙设置 A-141
A-7 PLC 比较表：CJ 系列 和 CS 系列 PLC A-143

A-1 指令功能

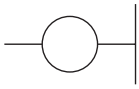
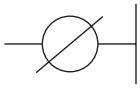
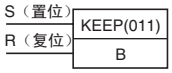
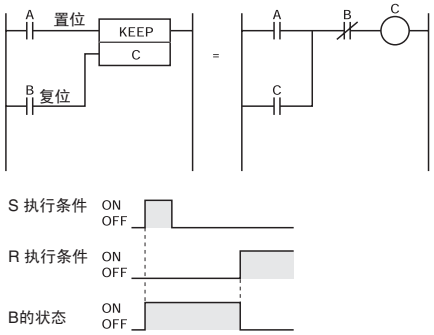
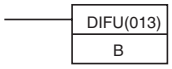
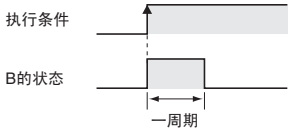
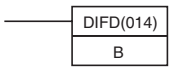
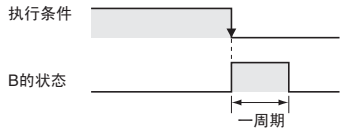
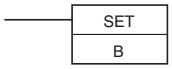
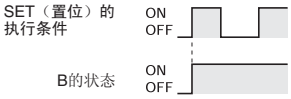
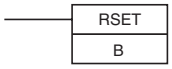
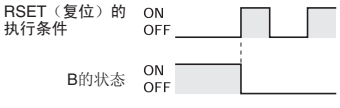
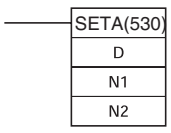
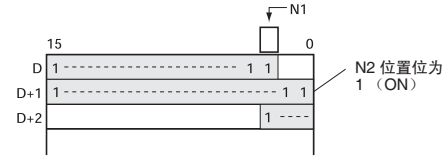
CJ2 CPU 单元支持下述指令。
详细内容，请参见 CS/CJ/NSJ 系列指令参考手册 （Cat. No. W474）。

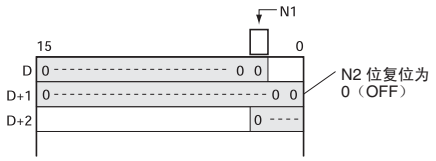
A-1-1 顺序输入指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
装载	LD @ LD %LD !LD ! @ LD !%LD	——		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	逻辑开始	不要
装载非	LD NOT @ LD NOT*2 %LD NOT !LD NOT ! @ LD NOT !%LD NOT	——		指定一个逻辑开始，并且用指定操作位的 ON/OFF 状态取非建立一个 ON/OFF 执行条件。	逻辑开始	不要
与	AND @ AND %AND !AND ! @ AND !%AND	——	符号 	把指定的操作位状态和当前执行条件进行逻辑与操作。	行线上继续	要
与非	AND NOT @ AND NOT %AND NOT !AND NOT ! @ AND NOT !%AND NOT	——	符号 	把指定操作位的状态取非并和当前执行条件进行逻辑与。	行线上继续	要
或	OR @ OR %OR !OR ! @ OR !%OR	——		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	行线上继续	要
或非	OR NOT @ OR NOT %OR NOT !OR NOT ! @ OR NOT !%OR NOT	——		把指定位状态取非和当前执行条件进行逻辑或操作。	行线上继续	要
与装载	AND LD	——		取逻辑块之间的逻辑与 LD ～ 逻辑块 A LD ～ 逻辑块 B AND LD ····· 逻辑块A和逻辑块B间串连	行线上继续	要

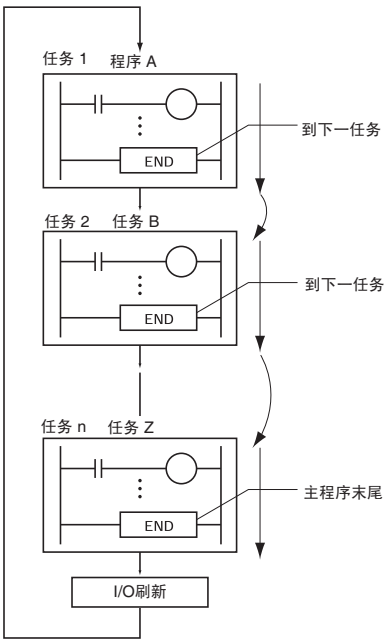
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
或装载	OR LD	——	逻辑块 逻辑块	取逻辑块之间的逻辑或 LD ～ 逻辑块 A LD ～ 逻辑块 B OR LD ····· 逻辑块A和逻辑块B间并连	行线上继续	要
非	NOT	520		执行条件取非	行线上继续	要
ON 条件	UP	521	UP(521)	当执行条件从 OFF → ON 时，UP （521）把执行条件在一个循环内变 ON。	行线上继续	要
OFF 条件	DOWN	522	DOWN(522)	当执行条件从 ON → OFF 时，DOWN （522）把执行条件在一个循环内变 ON。	行线上继续	要
位测试	LD TST	350	TST(350) S N 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	逻辑开始	不要
位测试非	LD TSTN	351	TSTN(351) S N 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。	逻辑开始	不要
与位测试	AND TST	350	AND TST(350) S N 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。	行线上继续	要
与位测试非	AND TSTN	351	AND TSTN(351) S N 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。	行线上继续	要
或位测试	OR TST	350	TST(350) S N 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。	行线上继续	要
或位测试非	OR TSTN	351	TSTN(351) S N 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。	行线上继续	要

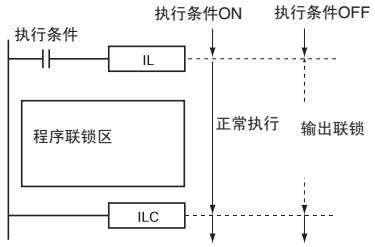
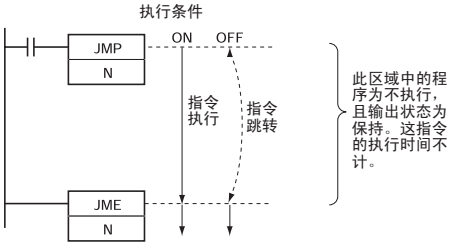
A-1-2 顺序输出指令

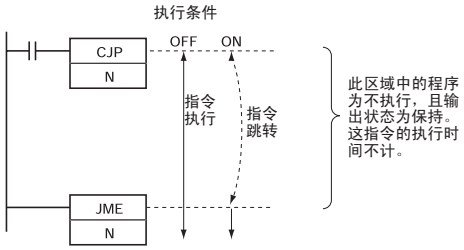
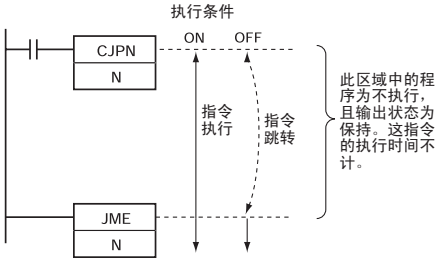
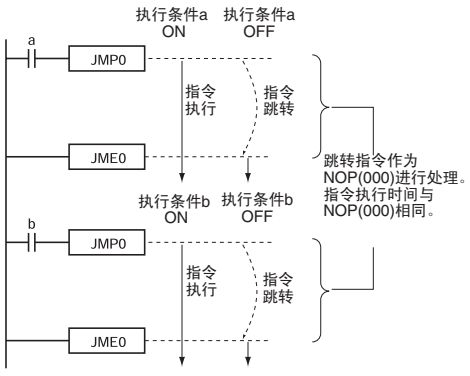
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
输出	OUT !OUT	——		把逻辑运算的结果 （执行条件）输出到指定位	输出	要
输出非	OUT NOT !OUT NOT	——		把逻辑运算的结果 （执行条件）取非，再输出到指定位。	输出	要
保持	KEEP !KEEP	011	 B: 位	进行自锁继电器操作  	输出	要
上升沿微分	DIFU !DIFU	013	 B: 位	当执行条件从OFF→ON变化（上升沿）时， DIFU(013)将所指定的位在一个循环内变为ON。 	输出	要
下降沿微分	DIFD !DIFD	014	 B: 位	当执行条件从ON→OFF变化（下降沿）时，DIFD(014)将所指定的位在一个循环内变为ON。 	输出	要
置位	SET @ SET %SET !SET ! @ SET !%SET	——	 B: 位	当执行条件为ON时，SET（复位）将操作位转为ON 	输出	要
复位	RSET @ RSET %RSET !RSET ! @ RSET !%RSET	——	 B: 位	当执行条件为ON时，RSET（复位）将操作位转为OFF 	输出	要
多位置位	SETA @ SETA	530	 D: 起始字 N1: 起始位 N2: 位数	SETA（530） 将指定位数的连续位转为ON。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多位复位	RSTA @ RSTA	531	RSTA(531) D N1 N2 D: 起始字 N1: 起始位 N2: 位数	RSTA（531）将指定位数的连续位转为OFF。 	输出	要
单一位复位	RSTB @ RSTB !RSTB ! @ RSTB	533	RSTB(533) D N D: 字地址 N: 位号	执行条件为 ON 时， RSTB （533）将指定字中的指定位转位 OFF。 与 RSET 指令不同， RSTB （533）可用于在 DM 或 EM 字中对某一位进行位的复位。	输出	要
单一位输出	OUTB @ OUTB !OUTB	534	OUTB(534) D N D: 字地址 N: 位号	OUTB （534）将逻辑处理的结果 （执行条件）输出到指定位。 与 OUT 指令不同， OUTB （534）可用于在 DM 或 EM 字中对位进行控制。	输出	要

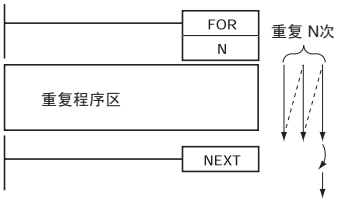
A-1-3 顺序控制指令

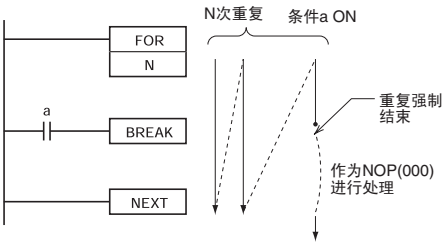
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
结束	END	001	END(001)	表示程序的末尾。 END（001）作为一个循环内的程序执行的完成。 END（001）指令后面的任何指令都不执行。 执行处理将进入下一任务。 当程序执行到最高任务号时，END（001）表示主程序的完结。 	输出	不要
空操作	NOP	000	——	此指令无功能。（NOP （000）不执行任何处理）	输出	不要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
联锁	IL	002	IL(002)	当IL（002）的执行条件为OFF时，IL（002）和ILC（003）间的所有输出联锁。IL（002）和ILC（003）通常为共同使用。 	输出	要
联锁解除	ILC	003	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）通常为共同使用。	输出	不要
多联锁微分保持	MILH	517	MILH (517) N D N: 联锁编号 D: 联锁状态位	当 MILH （517）的执行条件为 OFF 时，则 MILH （517）指令和下一 MILC （519）指令间的所有指令输出联锁。MILH （517）和 MILC （519）通常为共同使用。 可将 MILH （517） /MILC （519）联锁嵌入 （如， MILH （517） -MILH （517） -MILC （519） -MILC （519））。 如果在 MILH （517）及相应的 MILC （519）间具有微分指令 （DIFU， DIFD 以及带有@或 % 前缀的指令），则在建立了指令微分条件的情况下，将在清除联锁后执行指令。	输出	要
多联锁微分释放	MILR	518	MILR (518) N D N: 联锁编号 D: 联锁状态位	当 MILR （518）的执行条件为 OFF 时，则 MILR （518）指令和下一 MILC （519）指令间的所有指令输出联锁。MILR （518）和 MILC （519）通常为共同使用。 可将 MILR （518） /MILC （519）联锁嵌入 （如， MILR （518） -MILR （518） -MILC （519） -MILC （519））。 如果在 MILR （518）及相应 MILC （519）间具有微分指令 （DIFU， DIFD 以及带有@或 % 前缀的指令），则即使在建立了指令微分条件的情况下，在联锁清除后也不执行指令。	输出	要
多联锁解除	MILC	519	MILC (519) N N: 联锁编号	清除使用相同联锁编号的由 MILH （517）或 MILR （518）开始的联锁。 当 MILH （517） /MILR （518）的执行条件为 OFF 时，使用相同联锁编号的 MILH （517） /MILR （518）和对应 MILC （519）间的所有输出联锁。	输出	不要
跳转	JMP	004	JMP(004) N N: 跳转编号	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)指令。 JMP(004) 和JME(005)通常为共同使用。 	输出	要

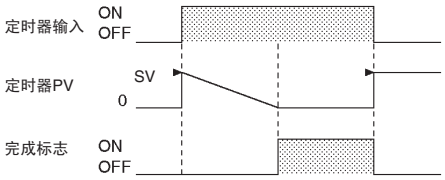
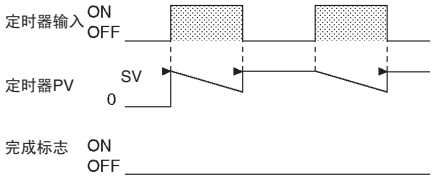
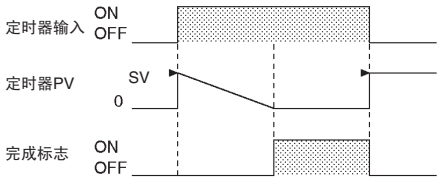
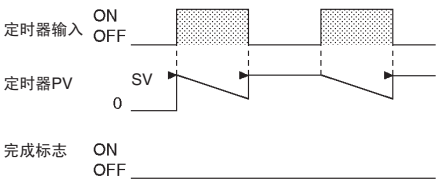
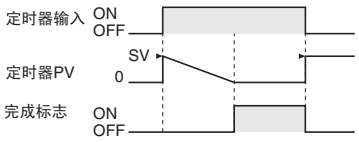
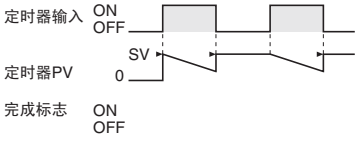
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
条件跳转	CJP	510	CJP(510) N N: 跳转编号	CJP(510)的运行基本与JMP(004)相反。 当CJP(510)的执行条件为ON时，程序执行直接跳转至与CJP(510)指令相同编号的第一个JME(005)指令。 CJP(510) 和JME(005)通常为共同使用。 	输出	要
条件跳转非	CJPN	511	CJPN(511) N N: 跳转编号	CJPN(511)的运行基本与JMP(004)相同。 当JMP(004)的执行条件为OFF时，程序执行直接跳转至与CJPN(511)指令相同编号的第一个JME(005)指令。 CJPN(511)和JME(005)通常为共同使用。 	输出	不要
跳转中止	JME	005	JME(005) N N: 跳转编号	表示跳转指令的目的地。	输出	不要
多次跳转	JMP0	515	JMP0(515)	当JMP0(515) 的执行条件为 OFF时，程序中从JMP0(515)输出到下一条JME0(516) 的所有指令都将作为NOP(000)进行处理。JMP0(515)和JME0(516)通常为共同使用。程序中可共同使用的次数并无任何限制。 	输出	要
多次跳转中止	JME0	516	JME0(516)	当 JMP0 （515）的执行条件为 OFF 时，程序中从 JMP0 （515）输出到下一条 JME0 （516）的所有指令都将作为 NOP （000）进行处理。 JMP0 （515）和 JME0 （516）通常为共同使用。程序中可共同使用的次数并无任何限制。	输出	不要

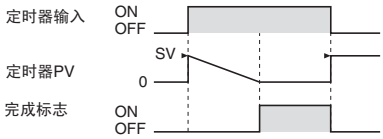
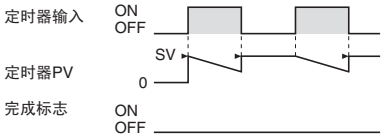
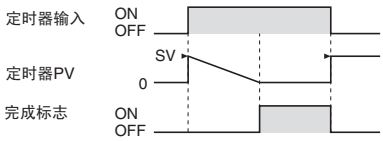
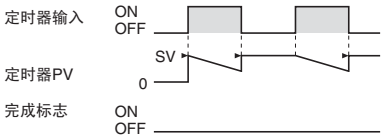
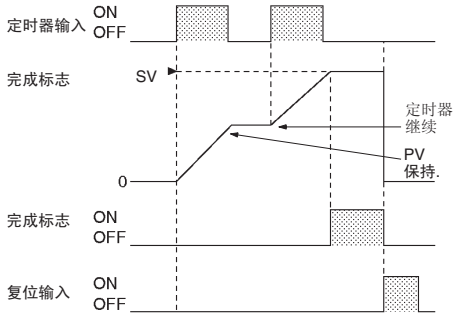
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件

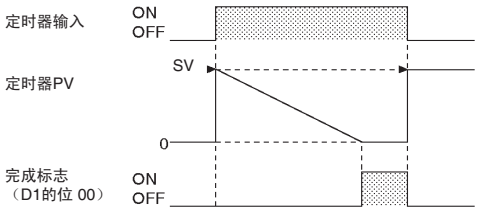
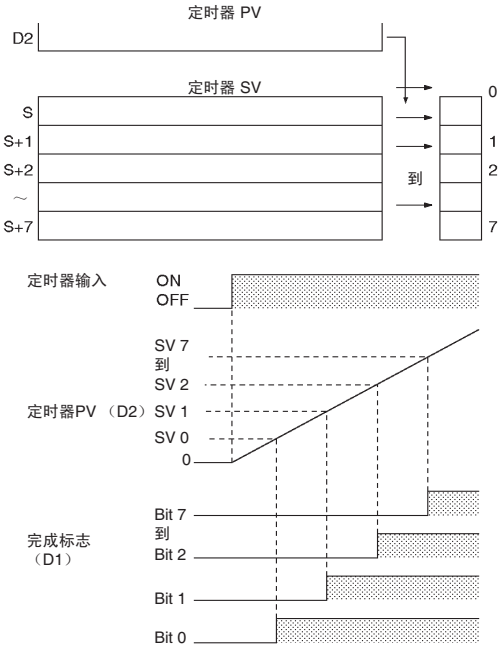
FOR 循环	FOR	512	FOR(512)
			N
			N: 循环数
				FOR(512)和NEXT(513)间指令以指定次数进行重复。
				FOR(512)和NEXT(513)通常为共同使用。
				 
					输出	不要

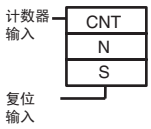
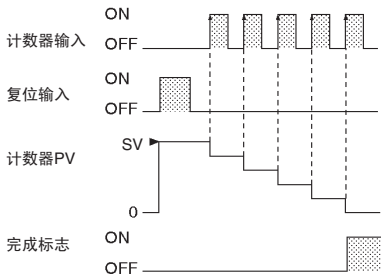
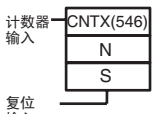
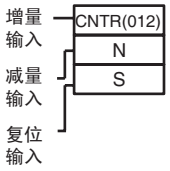
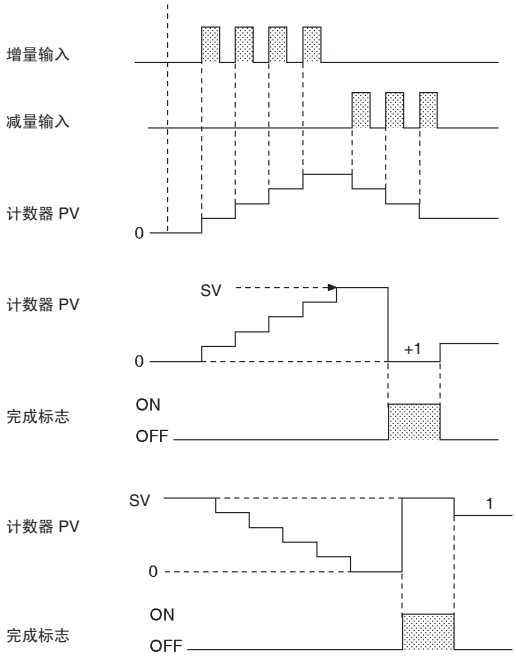
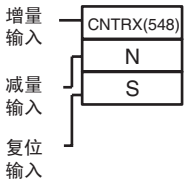
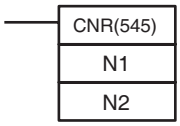
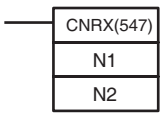
循环中止	BREAK	514	BREAK(514)	编写FOR-NEXT循环，以实现在所给定的条件下取消循环的执行。循环中剩余的指令将作为 NOP(000)指令进行处理。 	输出	要
下一个循环	NEXT	513	NEXT(513)	FOR （512）和 NEXT （513）间指令以指定次数进行重复。 FOR （512）和 NEXT （513）通常为共同使用。	输出	不要

A-1-4 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
100MS 定时器	TIM （BCD）	——	TIM N S N: 定时器编号 S: 设定值	TIM/TIMX（550） 以0.1s为单位作一减量定时器。 设定值（SV）的设定范围为0～999.9s（BCD）；0～6,553.5s（二进制）（十进制或十六进制）  完成标志转为ON前，定时器输入置OFF。 	输出	要
	TIMX （二进制）	550	TIMX(550) N S N: 定时器编号 S: 设定值			
10MS 定时器	TIMH （BCD）	015	TIMH(015) N S N: 定时器编号 S: 设定值	TIMH（015）/TIMHX（551） 以10ms为单位作一减量定时器。 设定值（SV）的设定范围为0～99.99s（BCD）；0～655.35s（二进制）（十进制或十六进制）  完成标志转为ON前，定时器输入置OFF。 	输出	要
	TIMHX （二进制）	551	TIMHX(551) N S N: 定时器编号 S: 设定值			
1MS 定时器	TMHH （BCD）	540	TMHH(540) N S N: 定时器编号 S: 设定值	TMHH （540） /TMHHX （552）以 1ms 为单位作一减量定时器。设定值 （SV）的设定范围为 0 ～ 9.999s （BCD）； 0 ～ 65.535s （二进制）（十进制或十六进制）。 TMHH （540）和 TMHHX （552）的时间表与上述 TIMH （015）的相同。  完成标志转为ON前，定时器输入置OFF。 	输出	要
	TMHHX （BCD）	552	TMHHX(552) N S N: 定时器编号 S: 设定值			

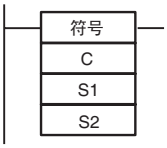
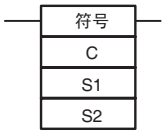
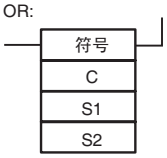
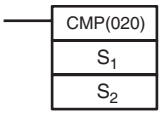
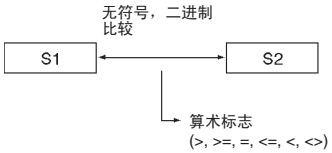
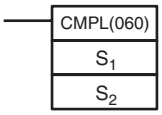
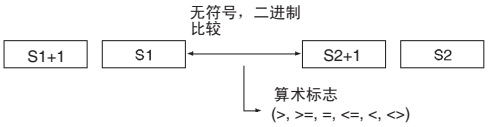
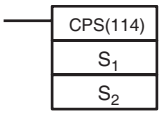
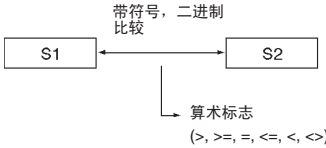
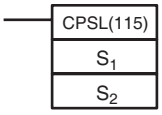
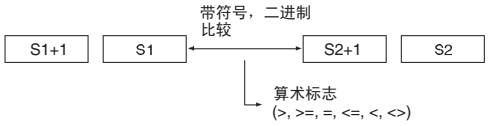
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
0.1MS 定时器	TIMU（BCD）	541	TIMU (541) N S N: 定时器编号 S: 设定值	TIMU （541） /TIMUX （556）以 0.1ms 为单位作一减量定时器。设定值 （SV）的设定范围为 0 ～ 0.9999s （BCD）； 0 ～ 6.5535s （二进制）（十进制或十六进制）。  完成标志转为ON前，定时器输入置OFF。 	输出	要
	TIMUX（BIN）	556	TIMUX(556) N S N: 定时器编号 S: 设定值			
0.01MS 定时器	TMUH（BCD）	544	TMUH(544) N S N: 定时器编号 S: 设定值	TMUH （544） /TMUHX （557）以 0.01ms 为单位作一减量定时器。设定值 （SV）的设定范围为 0 ～ 0.09999s （BCD）； 0 ～ 0.65535 s （二进制）（十进制或十六进制）。  完成标志转为ON前，定时器输入置OFF。 	输出	要
	TIMUX（BIN）	556	TMUHX(557) N S N: 定时器编号 S: 设定值			
累计定时器	TTIM（BCD）	087	定时器输入 TTIM(087) N S 复位输入 N: 定时器编号 S: 设定值	TTIM（087）/TTIMX（555） 以0.1s为单位作一减量定时器。 设定值（SV）的设定范围为0～999.9s（BCD）；0 ～6,553.5s（二进制）（十进制或十六进制） 	输出	要
	TTIMX（二进制）	555	定时器输入 TTIMX(555) N S 复位输入 N: 定时器编号 S: 设定值			

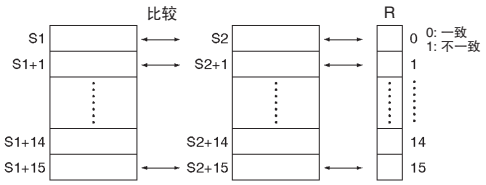
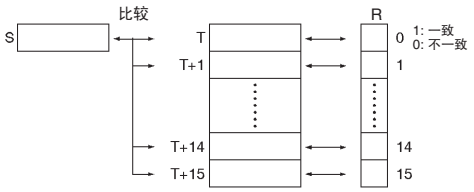
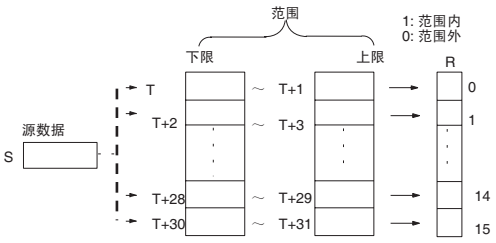
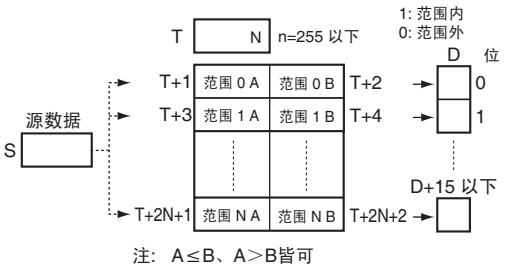
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
长时间定时器	TIML （BCD）	542	TIML(542) D1 D2 S D1: 完成标志 D2: PV 字 S: SV 字	TIML（542）/TIMLX（553） 以0.1s为单位作一减量定时器。BCD时为最大约115天；二进制时为最大约49,710天（十进制或十六进制）。 	输出	要
	TIMLX （二进制）	553	TIMLX(553) D1 D2 S D1: 完成标志 D2: PV 字 S: SV 字			
多输出定时器	MTIM （BCD）	543	MTIM(543) D1 D2 S D1: 完成标志 D2: PV 字 S: 第一个SV 字	MTIM（543）/MTIMX（554） 以0.1s为单位作一减量定时器。此定时器有8个独立的（SV）和完成标志。 设定值（SV）的设定范围为0～999.9s（BCD）；0～6,553.5s（二进制）（十进制或十六进制） 	输出	要
	MTIMX （二进制）	554	MTIMX(554) D1 D2 S D1: 完成标志 D2: PV 字 S: 第一个SV 字			
定时器复位	TRSET	549	TRSET(549) N N: 定时器编号	复位指定定时器。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
计数器	CNT （BCD）	——	 N: 计数器编号 S: 设定值	CNT/CNTX（546）作一减量计数器。 设定值（SV）的设定范围为0～9,999（BCD）； 0～65,535（二进制）（十进制或十六进制） 	输出	要
	CNTX （二进制）	546	 N: 计数器编号 S: 设定值			
可逆计数器	CNTR （BCD）	012	 N: 计数器编号 S: 设定值	CNTR（012）/CNTRX（548）作一可逆计数器 	输出	要
	CNTRX （二进制）	548	 N: 计数器编号 S: 设定值			
定时器 / 计数器复位	CNR @ CNR （BCD）	545	 N1: 范围中的首编号 N2: 范围中的末尾编号	CNR （545） /CNRX （547）在定时器 / 计数器数值的指定范围内进行定时器 / 计数器的复位。将设定值 （SV）设置为最大值：#9999 （BCD）； #FFFF （二进制）。	输出	要
	CNRX @ CNRX （二进制）	547	 N1: 范围中的首编号 N2: 范围中的末尾编号			

A-1-5 比较指令

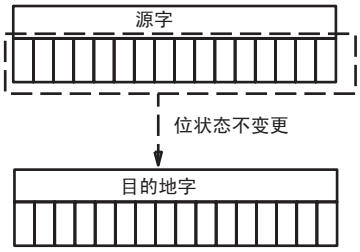
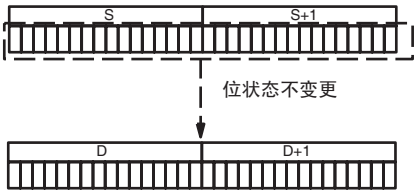
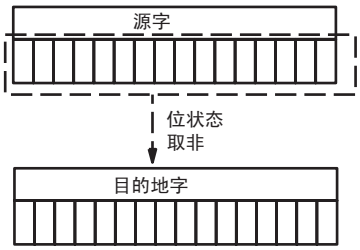
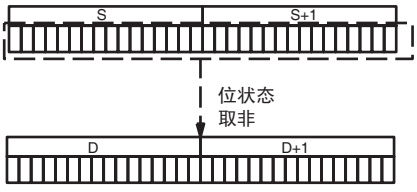
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
符号比较 （无符号）	LD, AND, OR+ =, <>, <, <=, >, >=	300 （=） 305 （<>） 310 （<） 315 （<=） 320 （>） 325 （>=）	LD 符号和选项 S_1 S_2 AND 符号和选项 S_1 S_2 OR 符号和选项 S_1 S_2 S_1: 比较数据 1 S_2: 比较数据 2	符号比较指令（无符号）以16位二进制数据形式对两个数值（常数与/或指定字的内容）进行比较。 并在比较结果为真时，形成一个ON执行条件。 共有3类符号比较指令：LD (LOAD)，AND，OR。 LD 比较结果为真时，执行条件置ON。 < S1 S2 AND 比较结果为真时，执行条件置ON。 < S1 S2 OR < S1 S2 比较结果为真时，执行条件置ON。	LD：逻辑开始 AND, OR：行线上继续	LD：不要 AND, OR：要
符号比较 （双字长，无符号）	LD, AND, OR+ =, <>, <, <=, >, >= + L	301 （=） 306 （<>） 311 （<） 316 （<=） 321 （>） 326 （>=）	S_1：比较数据 1 S_2：比较数据 2	符号比较指令 （双字长，无符号）以无符号 32 位二进制数据对两个数值 （常数与 / 或指定双字数据的内容）进行比较。并且在比较结果是真时，产生一个 ON 执行条件。共有 3 类符号比较指令：LD （LOAD）， AND， OR。	LD：逻辑开始 AND, OR：行线上继续	LD：不要 AND, OR：要
符号比较 （带符号）	LD, AND, OR+ =, <>, <, <=, >, >= +S	302 （=） 307 （<>） 312 （<） 317 （<=） 322 （>） 327 （>=）	S_1：比较数据 1 S_2：比较数据 2	符号比较指令 （带符号）以带符号 16 位二进制 （4 位十六进制）对两个数值 （常数与 / 或指定字的内容）进行比较。并且在比较结果是真时，产生一个 ON 执行条件。共有 3 类符号比较指令：LD （LOAD）， AND， OR。	LD：逻辑开始 AND, OR：行线上继续	LD：不要 AND, OR：要
符号比较 （双字长，带符号）	LD, AND, OR+ =, <>, <, <=, >, >= +SL	303 （=） 308 （<>） 313 （<） 318 （<=） 323 （>） 328 （>=）	S_1：比较数据 1 S_2：比较数据 2	符号比较指令 （双字长，带符号）以带符号 32 位二进制 （8 位十六进制）对两个数值 （常数与 / 或指定字的内容）进行比较。并且在比较结果是真时，产生一个 ON 执行条件。共有 3 类符号比较指令：LD （LOAD）， AND， OR。	LD：逻辑开始 AND, OR：行线上继续	LD：不要 AND, OR：要

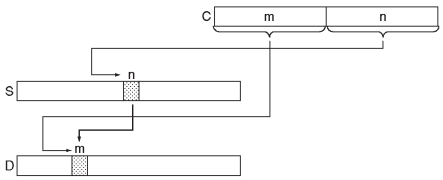
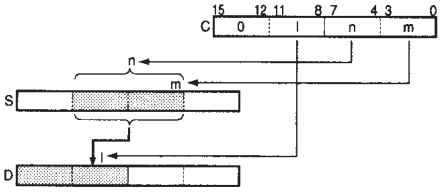
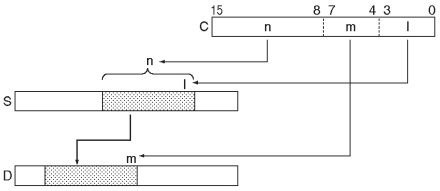
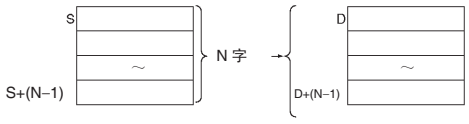
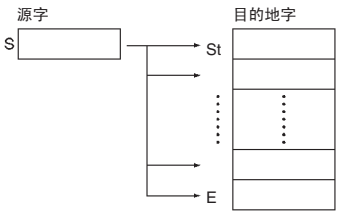
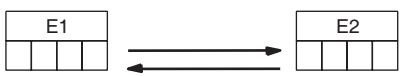
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
时间比较指令	LD, AND, OR+ = DT, <> DT, < DT, <= DT, > DT, >= DT	341 (= DT) 342 (<>DT) 343(< DT) 344(<= DT) 345(> DT) 346(>= DT)	LD (LOAD):  AND:  OR:  C: 控制字 S1: 当前时间的首字 S2: 比较时间的首字	时间比较指令比较两 BCD 时间值，并当确认比较条件后，将执行条件置 ON。 共有 3 类符号比较指令：LD （LOAD）， AND， OR。在比较中可将时间值 （年，月，日，小时，分，秒）屏蔽 / 不屏蔽，所以可以容易地创建日历定时器功能。	LD：逻辑开始 AND, OR：行线上继续	LD：不要 AND, OR：要
无符号比较	CMP !CMP	020	 S1: 比较数据 1 S2: 比较数据 2	比较两个无符号二进制值（常数与/或指定字的内容），并将结果输出到辅助区中的算术标志。 	输出	要
双字长无符号比较	CMPL	060	 S1: 比较数据 1 S2: 比较数据 2	比较两个双字长无符号二进制值（常数与/或指定字的内容），并将结果输出到辅助区中的算术标志。 	输出	要
带符号二进制比较	CPS !CPS	114	 S1: 比较数据 1 S2: 比较数据 2	比较两个带符号二进制值（常数与/或指定字的内容），并将结果输出到辅助区中的算术标志。 	输出	要
双字长带符号二进制比较	CPSL	115	 S1: 比较数据 1 S2: 比较数据 2	比较两个双字长带符号二进制值（常数与/或指定字的内容），并将结果输出到辅助区中的算术标志。 	输出	要

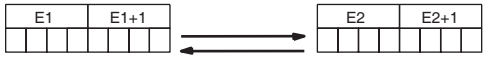
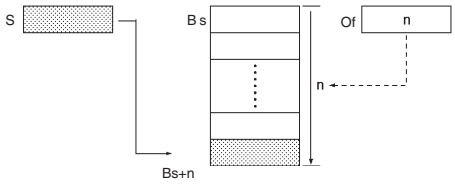
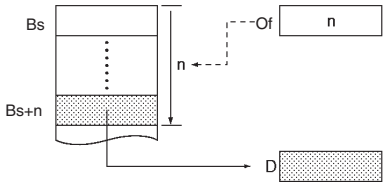
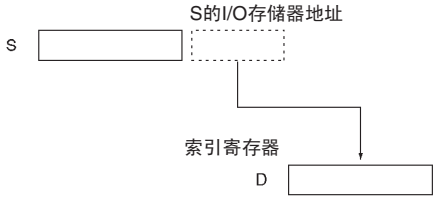
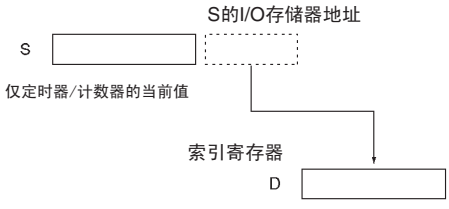
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多重比较	MCMP @ MCMP	019	MCMP(019) S_1 S_2 R S_1: 第1组的首字 S_2: 第2组的首字 R: 结果字	将16个连续字与另16个字进行比较，并当结果不一致时，将相应位转为ON。 	输出	要
表比较	TCMP @ TCMP	085	TCMP(085) S T R S: 源数据 T: 表的首字 R: 结果字	将源数据与16个字进行比较，并当结果一致时，将相应位转为ON。 	输出	要
无符号块比较	BCMP @ BCMP	068	BCMP(068) S T R S: 源数据 T: 表的首字 R: 结果字	比较源数据与16组比较数据（16个上下限范围），并当在范围内时，将相应的位转为ON。 	输出	要
扩展块比较	BCMP2 @ BCMP2	502	BCMP2(502) S T R S: 源数据 T: 块的首字 R: 结果字	比较源数据与最多 256 组数据 （由上下限范围），并当在范围内时，将相应的位转为 ON。  注: A≤B、A＞B皆可	输出	要
区域范围比较	ZCP	088	ZCP(088) CD LL UL CD: 比较数据 （单字） LL: 下限范围 UL: 上限范围	比较 CD 中 16 位无符号二进制值 （字内容或常数）与上下限范围 （LL 与 UL），并将结果输出到辅助区中的算术标志。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双区域范围比较	ZCPL	116	ZCPL(116) CD LL UL CD: 比较数据（双字） LL: 下限范围 UL: 上限范围	比较 CD 和 CD+1 中 32 位无符号二进制值 （字内容或常数）与上下限范围 （LL 与 UL），并将结果输出到辅助区中的算术标志。	输出	要

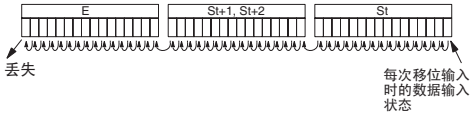
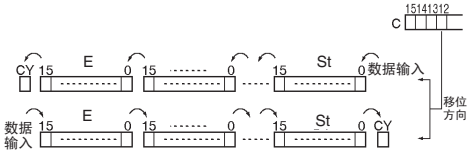
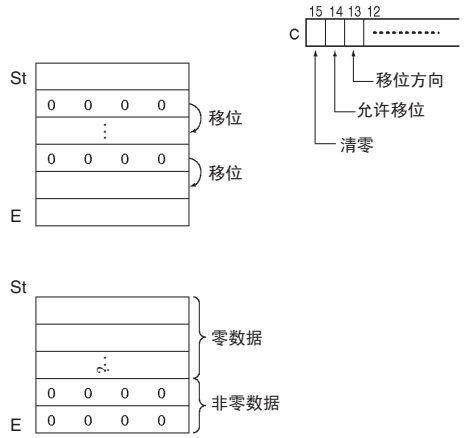
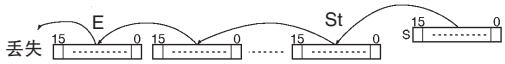
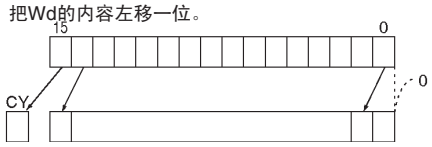
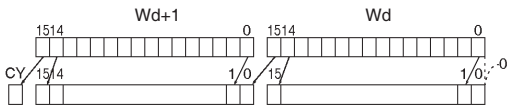
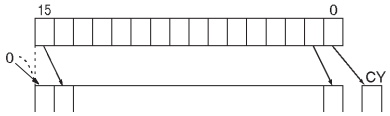
A-1-6 数据传送指令

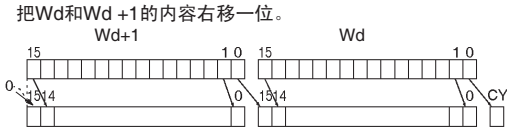
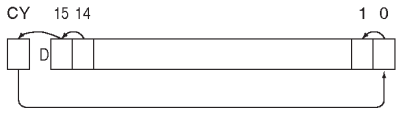
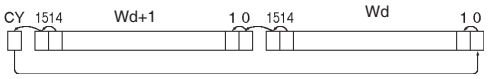
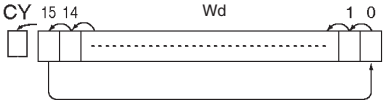
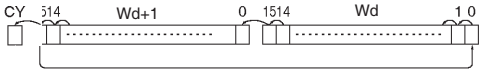
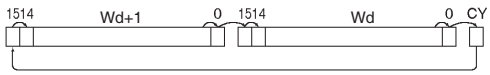
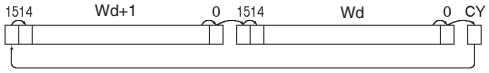
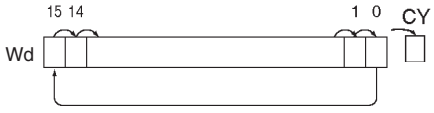
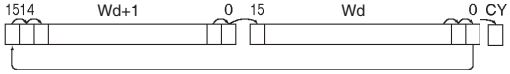
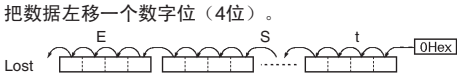
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
传送	MOV @ MOV !MOV ! @ MOV	021	MOV(021) S D S: 源 D: 目的地	传送一个数据字到指定字。 	输出	要
双字长传送	MOVL @ MOVL	498	MOVL(498) S D S: 源首字 D: 目的地首字	传送二个数据字到指定字。 	输出	要
取非传送	MVN @ MVN	022	MVN(022) S D S: 源 D: 目的地	传送一个数据字补码到指定字。 	输出	要
双字长取非传送	MVNL @ MVNL	499	MVNL(499) S D S: 源首字 D: 目的地首字	传送二个数据字补码到指定字。 	输出	要

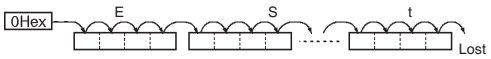
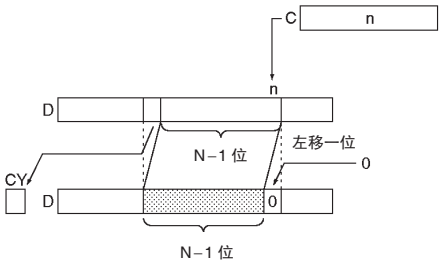
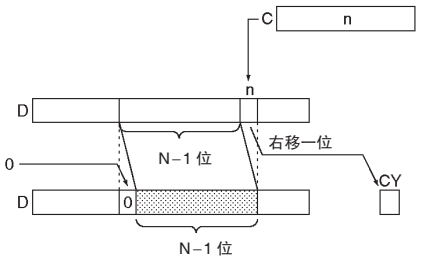
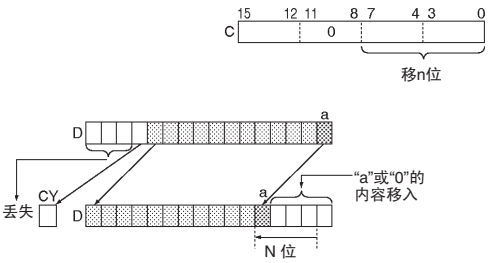
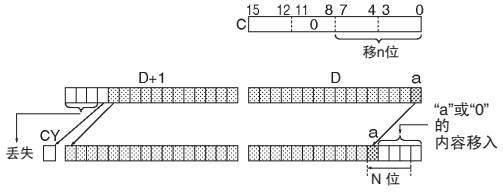
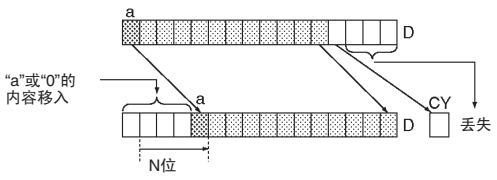
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
位传送	MOVB @ MOVB	082	MOVB(082) S C D S: 源字或数据 C: 控制字 D: 目的地字	传送指定位。 	输出	要
数字传送	MOVD @ MOVD	083	MOVD(083) S C D S: 源字或数据 C: 控制字 D: 目的地字	传送指定的一个或多个数字。 （各数字最多可为4位）。 	输出	要
多位传送	XFRB @ XFRB	062	XFRB(062) C S D C: 控制字 S: 源首字 D: 目的地首字	传送指定数目的连续位 	输出	要
块传送	XFER @ XFER	070	XFER(070) N S D N: 字数 S: 源首字 D: 目的地首字	传送指定数量的连续字 	输出	要
块置位	BSET @ BSET	071	BSET(071) S St E S: 源字 St: 起始字 E: 结束字	将相同字的内容复制到几个连续字中 	输出	要
数据交换	XCHG @ XCHG	073	XCHG(073) E1 E2 E1: 第一交换字 E2: 第二交换字	交换两个指定字的内容。 	输出	要

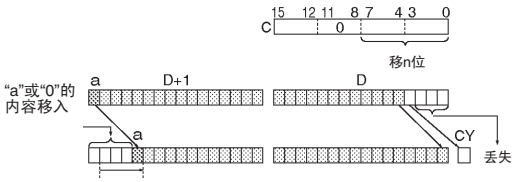
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字长数据交换	XCGL @ XCGL	562	XCGL(562) E1 E2 E1: 第一交换字 E2: 第二交换字	将一对连续字与另一对连续字进行交换。 	输出	要
单字分配	DIST @ DIST	080	DIST(080) S Bs Of S: 源字 Bs: 目的地基地址 Of: 偏移	通过添加偏移值到基地址计算出目的地字，并向其传送源字。 	输出	要
数据收集	COLL @ COLL	081	COLL(081) Bs Of D Bs: 源基地址 Of: 偏移 D: 目的地地址	通过添加偏移值到基地址计算出目的地字，并向其传送源字。 	输出	要
传送到寄存器	MOVR @ MOVR	560	MOVR(560) S D S: 源 (指定字或位) D: 目的地（索引寄存器）	在指定的索引寄存器中设置指定的字、位或定时器/计数器完成标志的内置I/O存储器地址。（使用MOVRW(561)设置索引寄存器中定时器/计数器PV的内置I/O存储器地址）。 	输出	要
将定时器 / 计数器当前值（PV）传送到寄存器	MOVRW @ MOVRW	561	MOVRW(561) S D S: 源（指定TC编号） D: 目的地（索引寄存器）	在指定的索引寄存器中设置指定的定时器/计数器当前值（PV）的内置I/O存储器地址。（使用MOVR(560)设置索引寄存器中字、位或定时器/计数器完成标志的内置I/O存储器地址）。 	输出	要

A-1-7 数据移位指令

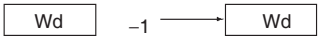
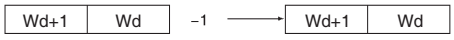
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
移位寄存器	SFT	010	数据输入 移位输入 复位输入 SFT(010) St E St: 起始字 E: 结束字	执行一移位寄存器 	输出	要
可逆移位寄存器	SFTR @ SFTR	084	SFTR(084) C St E C: 控制字 St: 起始字 E: 结束字	生成一个可左、可右的移位寄存器 	输出	要
异步移位寄存器	ASFT @ ASFT	017	ASFT(017) C St E C: 控制字 St: 起始字 E: 结束字	在指定字范围内朝St或E方向移动非零字，以取代0000Hex字数据。 	输出	要
字移位	WSFT @ WSFT	016	WSFT(016) S St E S: 源字 St: 起始字 E: 结束字	将St和E之间的数据按字移位。 	输出	要
算术左移	ASL @ ASL	025	ASL(025) Wd Wd: 字	把Wd的内容左移一位。 	输出	要
双字左移	ASLL @ ASLL	570	ASLL(570) Wd Wd: 字	把Wd和Wd +1的内容左移一位。 	输出	要
算术右移	ASR @ ASR	026	ASR(026) Wd Wd: 字	把Wd的内容右移一位。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字右移	ASRL @ ASRL	571	ASRL(571) Wd Wd: 字	把Wd和Wd +1的内容右移一位。 	输出	要
循环左移	ROL @ ROL	027	ROL(027) Wd Wd: 字	将Wd中的各位包括进位标志(CY)左移一位。 	输出	要
双字循环左移	ROLL @ ROLL	572	ROLL(572) Wd Wd: 字	将Wd和Wd +1中的各位包括进位标志(CY)左移一位。 	输出	要
无进位循环左移	RLNC @ RLNC	574	RLNC(574) Wd Wd: 字	将Wd中的各位不包括进位标志(CY)左移一位。 	输出	要
无进位双字循环左移	RLNL @ RLNL	576	RLNL(576) Wd 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	要
循环右移	ROR @ ROR	028	ROR(028) Wd Wd: 字	将Wd中的各位包括进位标志(CY)右移一位。 	输出	要
双字循环右移	RORL @ RORL	573	RORL(573) Wd Wd: 字	将Wd和Wd +1中的各位包括进位标志(CY)右移一位。 	输出	要
无进位循环右移	RRNC @ RRNC	575	RRNC(575) Wd 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	要
无进位双字循环右移	RRNL @ RRNL	577	RRNL(577) Wd Wd: 字	把Wd和Wd +1中的各位不包括进位标志(CY)右移一位。 Wd +1最右位的内容移至Wd的最左位和进位标志(CY)。 	输出	要
一个数字左移	SLD @ SLD	074	SLD(074) St E St: 起始字 E: 结束字	把数据左移一个数字位（4位）。 	输出	要

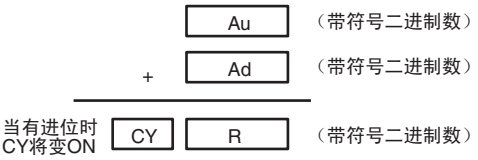
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
一个数字右移	SRD @ SRD	075	SRD(075) St E St: 起始字 E: 结束字	把数据右移一个数字位（4位）。 	输出	要
左移 N 位数据	NSFL @ NSFL	578	NSFL(578) D C N D: 移位的起始字 C: 起始位 N: 移位数据长度	向左移动指定的位数。 	输出	要
右移 N 位数据	NSFR @ NSFR	579	NSFR(579) D C N D: 移位的起始字 C: 起始位 N: 移位数据长度	向右移动指定的位数。 	输出	要
左移 N 位	NASL @ NASL	580	NASL(580) D C D: 移位字 C: 控制字	将指定的16位数据左移指定的位数。 	输出	要
双字左移 N 位	NSLL @ NSLL	582	NSLL(582) D C D: 移位字 C: 控制字	将指定的32位数据左移指定的位数。 	输出	要
右移 N 位	NASR @ NASR	581	NASR(581) D C D: 移位字 C: 控制字	将指定的16位数据右移指定的位数。 	输出	要

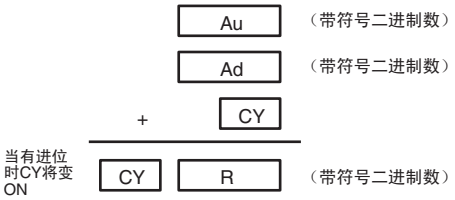
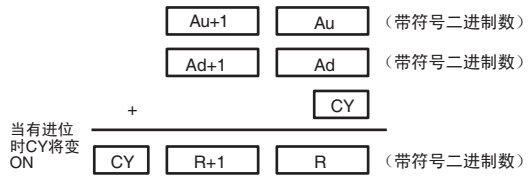
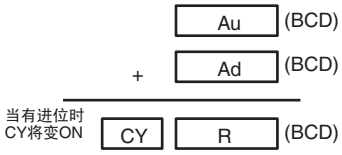
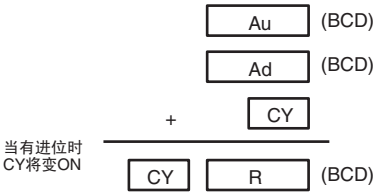
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字右移 N 位	NSRL @ NSRL	583	NSRL(583) D C D: 移位字 C: 控制字	将指定的32位数据右移指定的位数。 	输出	要

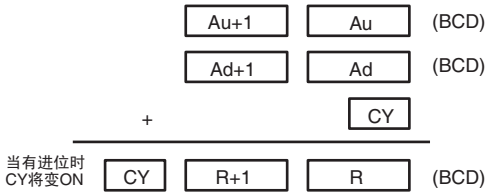
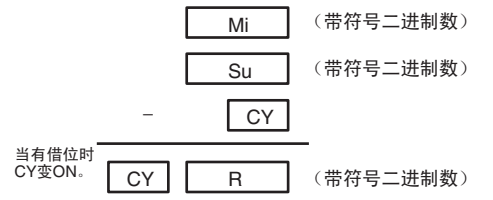
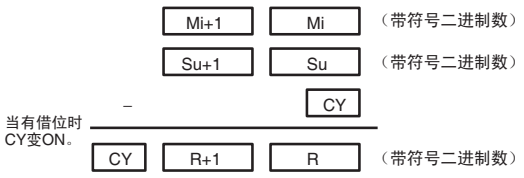
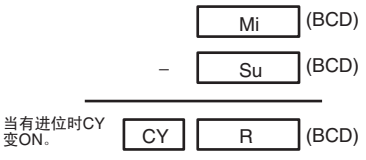
A-1-8 递增 / 递减指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
二进制递增	++ @ ++	590	++(590) Wd Wd: 字	将指定字的4位十六进制内容加1。 	输出	要
双字二进制递增	++L @ ++L	591	++L(591) Wd Wd: 字	将指定字的8位十六进制内容加1。 	输出	要
二进制递减	- - @ - -	592	– – (592) Wd Wd: 字	将指定字的4位十六进制内容减1。 	输出	要
双字二进制递减	- -L @ - -L	593	– – L(593) Wd Wd: 首字	将指定字的8位十六进制内容减1。 	输出	要
BCD 递增	++B @ ++B	594	++B(594) Wd Wd: 字	将指定字的4位BCD内容加1。 	输出	要
双字 BCD 递增	++BL @ ++BL	595	++BL(595) Wd Wd: 首字	将指定字的8位BCD内容加1。 	输出	要
BCD 递减	- -B @ - -B	596	– – B(596) Wd Wd: 字	将指定字的4位BCD内容减1。 	输出	要
双字 BCD 递减	- -BL @ - -BL	597	– – BL(597) Wd Wd: 首字	将指定字的8位BCD内容减1。 	输出	要

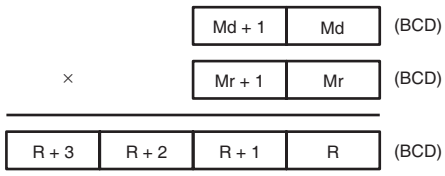
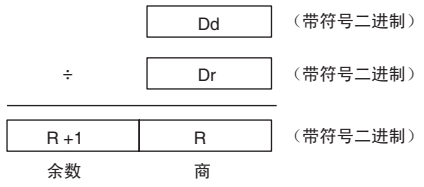
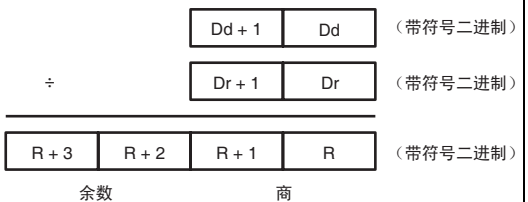
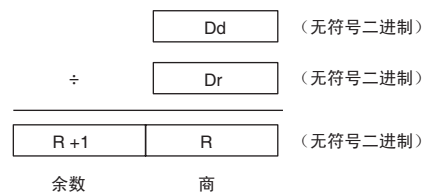
A-1-9 四则运算指令

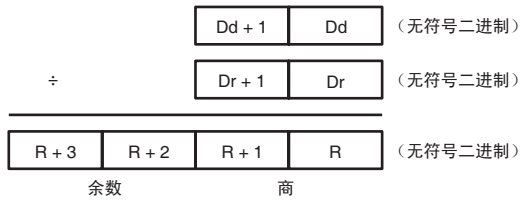
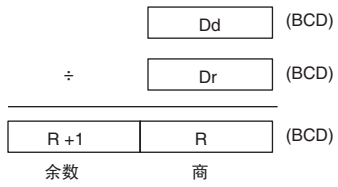
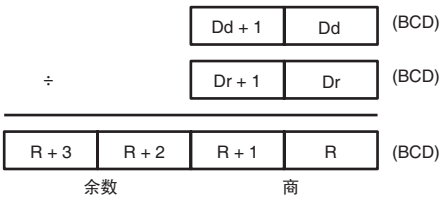
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无进位带符号二进制加法	+ @ +	400	+(400) Au Ad R Au: 被加字 Ad: 加字 R: 结果字	4位（单字）十六进制数据与/或常数相加。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无进位带符号双字二进制加法	+L @ +L	401	+L(401) Au Ad R Au: 被加首字 Ad: 加首字 R: 结果首字	8位（双字）十六进制数据与/或常数相加。 	输出	要
有进位带符号二进制加法	+C @ +C	402	+C(402) Au Ad R Au: 被加字 Ad: 加字 R: 结果字	4位（单字）十六进制数据与/或常数带进位标志（CY）相加。 	输出	要
有进位带符号双字二进制加法	+CL @ +CL	403	+CL(403) Au Ad R Au: 被加首字 Ad: 加首字 R: 结果首字	9位（双字）十六进制数据与/或常数带进位标志（CY）相加。 	输出	要
无进位 BCD 加法	+B @ +B	404	+B(404) Au Ad R Au: 被加字 Ad: 加字 R: 结果字	4位（单字）BCD数据与/或常数相加。 	输出	要
无进位双字 BCD 加法	+BL @ +BL	405	+BL(405) Au Ad R Au: 被加首字 Ad: 加首字 R: 结果首字	8位（双字）BCD数据与/或常数相加。 	输出	要
有进位 BCD 加法	+BC @ +BC	406	+BC(406) Au Ad R Au: 被加字 Ad: 加字 R: 结果字	4位（单字）BCD数据与/或常数带进位标志（CY）相加。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
有进位双字 BCD 加法	+BCL @ +BCL	407	+BCL(407) Au Ad R Au: 被加首字 Ad: 加首字 R: 结果首字	8位（双字）BCD数据与/或常数带进位标志（CY）相加。 	输出	要
无进位带符号二进制减法	- @ -	410	– (410) Mi Su R Mi: 被减字 Su: 减字 R: 结果字	4位（单字）十六进制数据与/或常数相减。 	输出	要
无进位带符号双字二进制减法	- L @ - L	411	–L(411) Mi Su R Mi: 被减字 Su: 减字 R: 结果字	8位（双字）十六进制数据与/或常数相减。 	输出	要
有进位带符号二进制减法	- C @ - C	412	–C(412) Mi Su R Mi: 被减字 Su: 减字 R: 结果字	4位（单字）十六进制数据与/或常数带进位标志（CY）相减。 	输出	要
有进位带符号双字二进制减法	- CL @ -- CL	413	–CL(413) Mi Su R Mi: 被减字 Su: 减字 R: 结果字	8位（双字）十六进制数据与/或常数带进位标志（CY）相减。 	输出	要
无进位 BCD 减法	- B @ - B	414	–B(414) Mi Su R Mi: 被减字 Su: 减字 R: 结果字	4位（单字）BCD数据与/或常数相减。 	输出	要

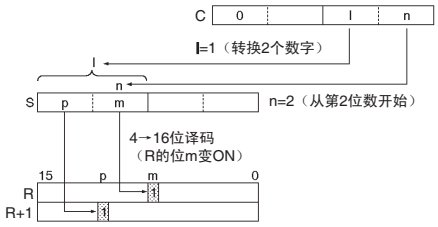
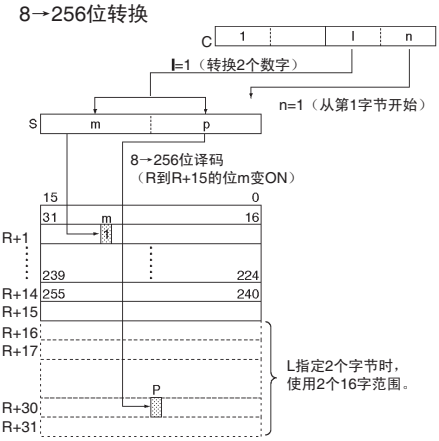
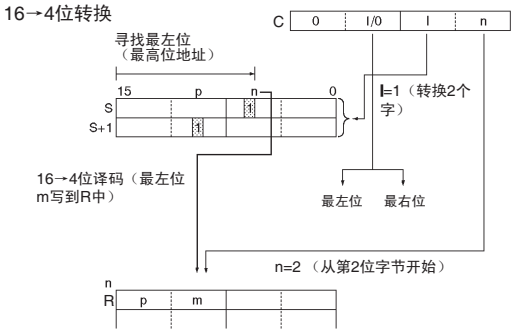
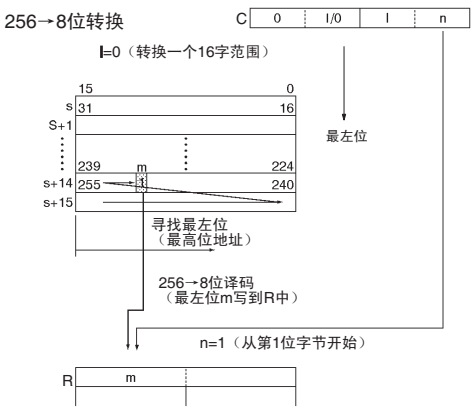
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无进位双字 BCD 减法	- BL @ - BL	415	–BL(415) Mi Su R Mi: 被减首字 Su: 减首字 R: 结果首字	8位（双字）BCD数据与/或常数相减。 Mi +1 Mi (BCD) – Su+1 Su (BCD) 当有借位时CY变ON。 CY R+1 R (BCD)	输出	要
有进位 BCD 减法	- BC @ - BC	416	–BC(416) Mi Su R Mi: 被减字 Su: 减字 R: 结果字	4位（单字）BCD数据与/或常数带进位标志（CY）相减。 Mi (BCD) Su (BCD) – CY 当有借位时CY变ON。 CY R (BCD)	输出	要
有进位双字 BCD 减法	- BCL @ - BCL	417	–BCL(417) Mi Su R Mi: 被减首字 Su: 减首字 R: 结果首字	8位（双字）BCD数据与/或常数带进位标志（CY）相减。 Mi +1 Mi (BCD) Su+1 Su (BCD) – CY 当有借位时CY变ON。 CY R+1 R (BCD)	输出	要
带符号二进制乘法	* @ *	420	*(420) Md Mr R Md: 被乘字 Mr: 乘字 R: 结果字	4位带符号十六进制数据与/或常数相乘。 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	要
带符号双字二进制乘法	*L @ *L	421	*L(421) Md Mr R Md: 被乘数首字 Mr: 乘数首字 R: 结果首字	8位带符号十六进制数据与/或常数相乘。 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	要
不带符号二进制乘法	*U @ *U	422	*U(422) Md Mr R Md: 被乘字 Mr: 乘字 R: 结果字	4位无符号十六进制数据与/或常数相乘。 Md （无符号二进制） × Mr （无符号二进制） R +1 | R （无符号二进制）	输出	要

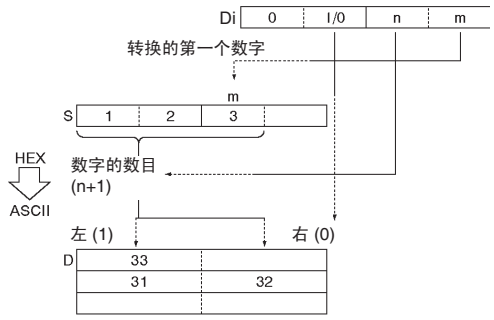
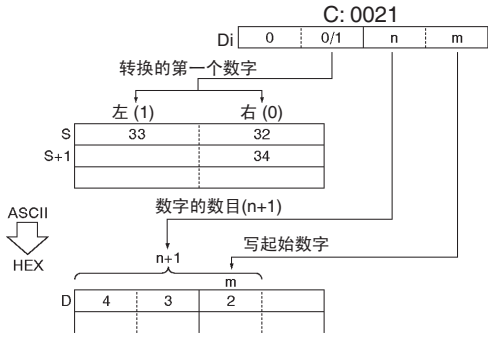
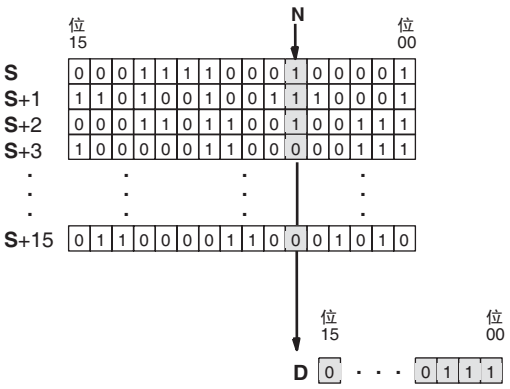
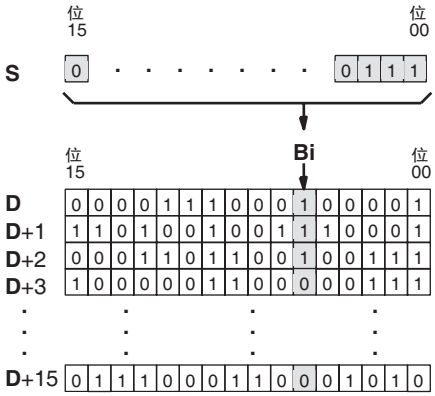
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
不带符号双字二进制乘法	*UL @ *UL	423	*UL(423) Md Mr R Md: 被乘数首字 Mr: 乘数首字 R: 结果首字	8位无符号十六进制数据与/或常数相乘。 	输出	要
BCD 码乘法	*B @ *B	424	*B(424) Md Mr R Md: 被乘字 Mr: 乘字 R: 结果字	4位（单字）BCD数据与/或常数相乘。 	输出	要
双字 BCD 码乘法	*BL @ *BL	425	*BL(425) Md Mr R Md: 被乘数首字 Mr: 乘数首字 R: 结果首字	8位（双字）BCD数据与/或常数相乘。 	输出	要
带符号二进制除法	/ @ /	430	/(430) Dd Dr R Dd: 被除字 Dr: 除字 R: 结果字	4位（单字）带符号十六进制数据与/或常数相除。 	输出	要
带符号双字二进制除法	/L @ /L	431	/L(431) Dd Dr R Dd: 被除数首字 Dr: 除数首字 R: 结果首字	8位（双字）带符号十六进制数据与/或常数相除。 	输出	要
不带符号二进制除法	/U @ /U	432	/U(432) Dd Dr R Dd: 被除字 Dr: 除字 R: 结果字	4位（单字）无符号十六进制数据与/或常数相除。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
不带符号双字二进制除法	/UL @ /UL	433	/UL(433) Dd Dr R Dd: 被除数首字 Dr: 除数首字 R: 结果首字	8位（双字）无符号十六进制数据与/或常数相除。 	输出	要
BCD 码除法	/B @ /B	434	/B(434) Dd Dr R Dd: 被除字 Dr: 除字 R: 结果字	4位（单字）BCD数据与/或常数相除。 	输出	要
双字 BCD 码除法	/BL @ /BL	435	/BL(435) Dd Dr R Dd: 被除数首字 Dr: 除数首字 R: 结果首字	8位（双字）BCD数据与/或常数相除。 	输出	要

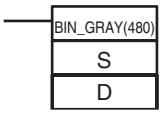
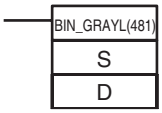
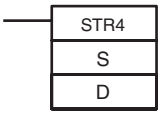
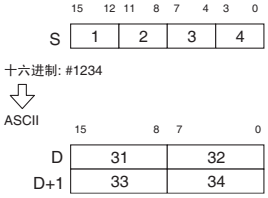
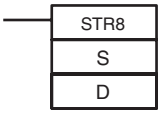
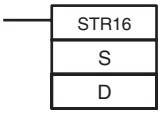
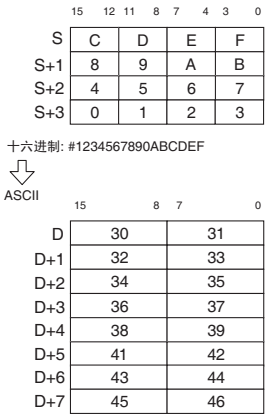
A-1-10 转换指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
BCD → 二进制	BIN @ BIN	023	BIN(023) S R S: 源字 R: 结果字	把BCD数据转换成二进制数据。 S (BCD) → R (BIN)	输出	要
双字 BCD → 二进制	BINL @ BINL	058	BINL(058) S R S: 源首字 R: 结果首字	8位BCD数据转换成8位十六进制数据（32位二进制数据）。 S (BCD) → R (BIN) S+1 (BCD) → R+1 (BIN)	输出	要
二进制→ BCD	BCD @ BCD	024	BCD(024) S R S: 源字 R: 结果字	把一个字的二进制数据转换成一个字的BCD数据。 S (BIN) → R (BCD)	输出	要
双字二进制→ BCD	BCDL @ BCDL	059	BCDL(059) S R S: 源首字 R: 结果首字	把8位十六进制（32位二进制）数据转换成8位BCD数据。 S (BIN) → R (BCD) S+1 (BIN) → R+1 (BCD)	输出	要
二进制求补	NEG @ NEG	160	NEG(160) S R S: 源字 R: 结果字	计算一个字的十六进制数据2的补码。 $\overline{(S)}$ → (R)　2的补码 (补码 + 1)	输出	要
双字二进制求补	NEGL @ NEGL	161	NEGL(161) S R S: 源首字 R: 结果首字	计算两个字的十六进制数据2的补码。 $\overline{(S+1, S)}$ → (R+1, R)　2的补码 (补码 + 1)	输出	要
16 位→ 32 位带符号二进制	SIGN @ SIGN	600	SIGN(600) S R S: 源字 R: 结果首字	把一个带符号的16位二进制值扩展为等同的32位。 MSB S MSB = 1: FFFF Hex MSB = 0: 0000 Hex D+1　D D = S的内容	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
数据译码	MLPX @ MLPX	076	MLPX(076) S C R S: 源字 C: 控制字 R: 结果首字	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  8→256位转换 	输出	要
数据编码	DMPX @ DMPX	077	DMPX(077) S R C S: 源字首字 R: 结果字 C: 控制字	在源字（16字范围）中寻找第一个或者最后一个ON位的位置，并将该值写到结果字中指定的数字（字节）。 16→4位转换  256→8位转换 	输出	要

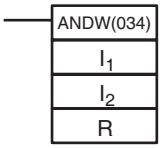
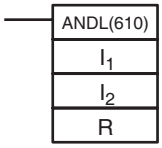
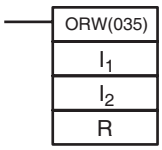
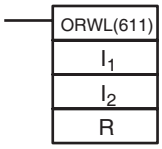
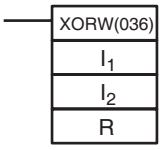
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
ASCII 转换码	ASC @ ASC	086	ASC(086) S Di D S: 源字 Di: 数字指定器 D: 目的地首字	把源字中的4位十六进制数转换成相应的8位ASCII码。 	输出	要
ASCII 码→十六进制	HEX @ HEX	162	HEX(162) S Di D S: 源首字 Di: 数字指定器 D: 目的地字	把源字中的最多4字节的ASCII数据转换成相应的十六进制数，并将这些数字写入指定的目的字中。 	输出	要
列→行	LINE @ LINE	063	LINE(063) S N D S: 源首字 N: 位号 D: 目的地字	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。 	输出	要
行→列	COLM @ COLM	064	COLM(064) S D N S: 源字 D: 目的地首字 N: 位号	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个连续字中的相同位号）。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
带符号的 BCD →二进制	BINS @ BINS	470	BINS(470) C S D C: 控制字 S: 源字 D: 目的地字	把一个字的带符号BCD数据转换成一个字的带符号二进制数据。 C 在C中指定带符号的BCD格式 S 带符号的BCD → D 带符号的二进制	输出	要
带符号的双字 BCD →二进制	BISL @ BISL	472	BISL(472) C S D C: 控制字 S: 源首字 D: 目的地首字	把双字带符号BCD数据转换成双字带符号的二进制数据。 C 在C中指定带符号的BCD格式 S 带符号的BCD → D 带符号的二进制 S+1 带符号的BCD D+1 带符号的二进制	输出	要
带符号的二进制 → BCD	BCDS @ BCDS	471	BCDS(471) C S D C: 控制字 S: 源字 D: 目的地字	把一个字的带符号二进制数据转换成一个字的带符号的BCD数据。 C 在C中指定带符号的BCD格式 S 带符号的二进制 → D 带符号的BCD	输出	要
带符号的双字二进制→ BCD	BDSL @ BDSL	473	BDSL(473) C S D C: 控制字 S: 源首字 D: 目的地首字	把双字带符号二进制数据转换成双字带符号的BCD数据。 C 在C中指定带符号的BCD格式 S 带符号的二进制 → D 带符号的BCD S+1 带符号的二进制 D+1 带符号的BCD	输出	要
格雷 （GRAY）码转换	GRY	474	GRY (474) C S D C: 控制字 S: 源字 D: 目的地首字	按照指定的分辨率，将指定字中的格雷 （GRAY）码数据转换为二进制、 BCD 或角度 （° ）数据。	输出	要
格雷 （GRAY）码→二进制 转换	GRAY _BIN @ GRAY_ BIN	478	GRAY_BIN(478) S D S: 源 (格雷（GRAY）码) D: 目的 (二进制代码)	将由 S 指定的格雷 （GRAY）码的一个字转换为二进制数据的一个字，并输出到 D。	输出	要
格雷 （GRAY）码→二进制双字转换	GRAY _BINL @ GRAY_ BINL	479	GRAY_BINL(479) S D S: 源首字 (格雷（GRAY）码) D: 目的地首字 (二进制)	将由 S 指定的格雷 （GRAY）码的二个字转换为二进制数据的二个字，并输出到 D。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
二进制→格雷（GRAY）码转换	BIN _GRAY @ BIN_ GRAY	480	 S: 源字 (二进制数据) D: 目的地字 (格雷（GRAY）码)	将由 S 指定的二进制数据的一个字转换为格雷 （GRAY）码的一个字，并输出到 D。	输出	要
二进制→格雷（GRAY）码双字转换	BIN _GRAYL @ BIN_ GRAYL	481	 S: 源首字 (二进制数据) D: 目的地首字 (格雷（GRAY）码)	将由 S 指定的二进制数据的二个字转换为格雷 （GRAY）码的二个字，并输出到 D。	输出	要
数值 4 位→ASCII	STR4 @ STR4	601	 S: 数字 D: ASCII文本	将 4 位十六进制数字 （#0000 ～ #FFFF）转换为 ASCII 数据（4 个字符）。 	输出	要
数值 8 位→ASCII	STR8 @ STR8	602	 S: 数字 D: ASCII文本	将 8 位十六进制数字 （#0000 0000 ～ #FFFF FFFF）转换为 ASCII 数据 （8 个字符）。 	输出	要
数值 16 位→ASCII	STR16 @ STR16	603	 S: 数字 D: ASCII文本	将 16 位十六进制数字 （#0000 0000 0000 0000 ～ #FFFF FFFF FFFF FFFF）转换为 ASCII 数据 （16 个字符）。 	输出	要

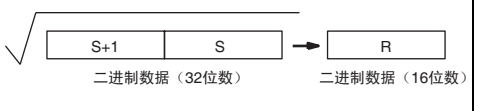
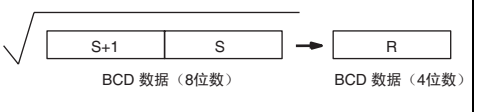
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
ASCII →数值 4 位	NUM4 @ NUM4	604	NUM4 S D S: ASCII文本 D: 数字	将 ASCII 数据的 4 字符转换为一个 4 位十六进制数值。 S: 31 32 / S+1: 33 34 ASCII ⇩ 十六进制 D: 1 2 3 4	输出	要
ASCII →数值 8 位	NUM8 @ NUM8	605	NUM8 S D S: ASCII文本 D: 数字	将 ASCII 数据的 8 字符转换为一个 8 位十六进制数值。 S: 31 32 / S+1: 33 34 / S+2: 35 36 / S+3: 37 38 ASCII ⇩ 十六进制 D: 5 6 7 8 / D+1: 1 2 3 4	输出	要
ASCII →数值 16 位	NUM16 @ NUM16	606	STR16(606) S D S: ASCII文本 D: 数字	将 ASCII 数据的 16 字符转换为一个 16 位十六进制数值。 S: 30 31 / S+1: 32 33 / S+2: 34 35 / S+3: 36 37 / S+4: 38 39 / S+5: 41 42 / S+6: 43 44 / S+7: 45 46 ASCII ⇩ 十六进制 D: C D E F / D+1: 8 9 A B / D+2: 4 5 6 7 / D+3: 0 1 2 3	输出	要

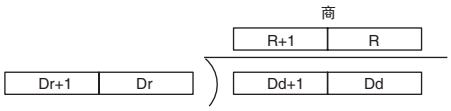
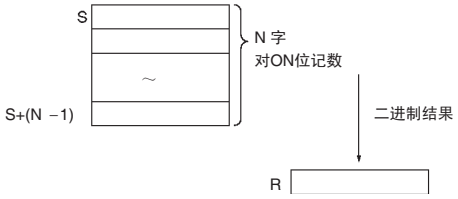
A-1-11 逻辑指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
逻辑与	ANDW @ ANDW	034	 I1: 输入 1 I2: 输入 2 R: 结果字	把一个单字数据和单字或常数的相应位作逻辑与运算。 **$I_1 . I_2 \rightarrow R$** **I1 | I2 | R** 1 | 1 | 1 1 | 0 | 0 0 | 1 | 0 0 | 0 | 0	输出	要
双字逻辑与	ANDL @ ANDL	610	 I1: 输入 1 I2: 输入 2 R: 结果字	把一个双字数据和双字或常数的相应位作逻辑与运算。 **$(I_1, I_1+1) . (I_2, I_2+1) \rightarrow (R, R+1)$** **I1, I1+1 | I2, I2+1 | R, R+1** 1 | 1 | 1 1 | 0 | 0 0 | 1 | 0 0 | 0 | 0	输出	要
逻辑或	ORW @ ORW	035	 I1: 输入 1 I2: 输入 2 R: 结果字	把一个单字数据和单字或常数的相应位作逻辑或运算。 **$I_1 + I_2 \rightarrow R$** **I1 | I2 | R** 1 | 1 | 1 1 | 0 | 1 0 | 1 | 1 0 | 0 | 0	输出	要
双字逻辑或	ORWL @ ORWL	611	 I1: 输入 1 I2: 输入 2 R: 结果字	把一双字数据和双字或常数的相应位作逻辑或运算。 **$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$** **I1, I1+1 | I2, I2+1 | R, R+1** 1 | 1 | 1 1 | 0 | 1 0 | 1 | 1 0 | 0 | 0	输出	要
异或	XORW @ XORW	036	 I1: 输入 1 I2: 输入 2 R: 结果字	把一单字数据和单字或常数的相应位作逻辑异或运算。 **$I_1 . I_2 + I_1 . I_2 \rightarrow R$** **I1 | I2 | R** 1 | 1 | 0 1 | 0 | 1 0 | 1 | 1 0 | 0 | 0	输出	要

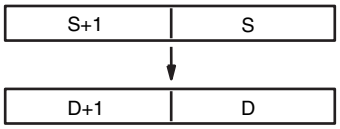
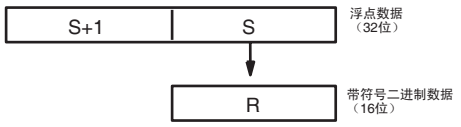
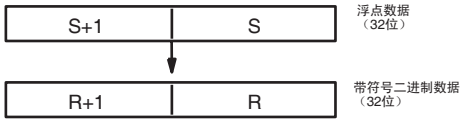
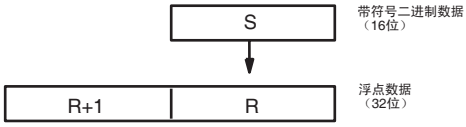
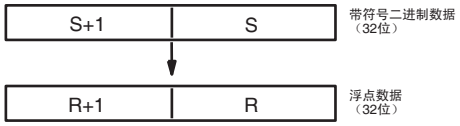
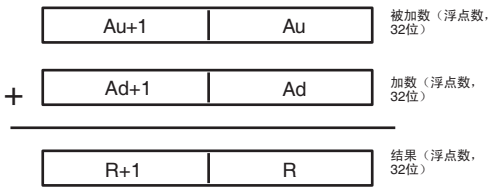
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字异或	XORL @ XORL	612	XORL(612) I_1 I_2 R I1: 输入 1 I2: 输入 2 R: 结果字	把一双字数据和双字或常数的相应位作逻辑异或运算。 $(I_1,I_1+1).(\overline{I_2,I_2+1})+(\overline{I_1,I_1+1}).(I_2,I_2+1) \rightarrow (R,R+1)$ I_1,I_1+1 | I_2,I_2+1 | R, R+1 1 | 1 | 0 1 | 0 | 1 0 | 1 | 1 0 | 0 | 0	输出	要
异或非	XNRW @ XNRW	037	XNRW(037) I_1 I_2 R I1: 输入 1 I2: 输入 2 R: 结果字	把一单字数据和单字或常数的相应位作逻辑异或非运算。 $I_1.I_2+I_1.I_2 \rightarrow R$ I_1 | I_2 | R 1 | 1 | 1 1 | 0 | 0 0 | 1 | 0 0 | 0 | 1	输出	要
双字异或非	XNRL @ XNRL	613	XNRL(613) I_1 I_2 R I1: 输入 1 I2: 输入 2 R: 结果字	把一双字数据和双字或常数的相应位作逻辑异或非运算。 $(I_1,I_1+1).(I_2,I_2+1)+(\overline{I_1,I_1+1}).(\overline{I_2,I_2+1}) \rightarrow (R,R+1)$ 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	要
求补	COM @ COM	029	COM(029) Wd Wd: 字	把Wd中所有ON位变OFF，并把所有OFF位变ON。 $\overline{Wd} \rightarrow Wd$: 1 → 0 和 0 → 1	输出	要
双字求补	COML @ COML	614	COML(614) Wd Wd: 字	把Wd 和 Wd+1中所有ON位变OFF，并把所有OFF位变ON。 $\overline{(Wd+1, Wd)} \rightarrow (Wd+1, Wd)$	输出	要

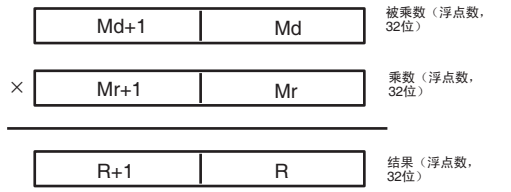
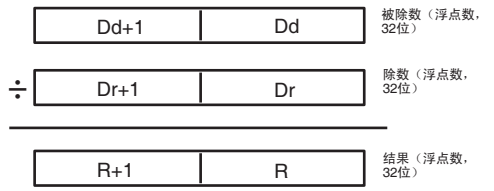
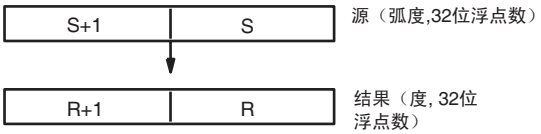
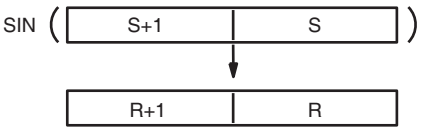
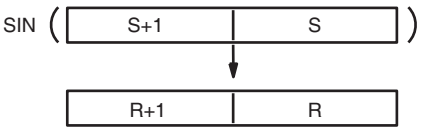
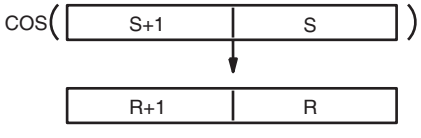
A-1-12 特殊算术指令

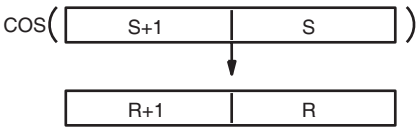
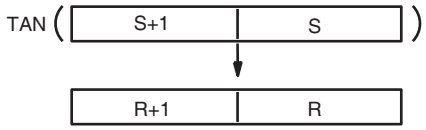
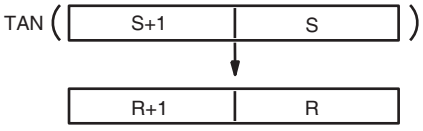
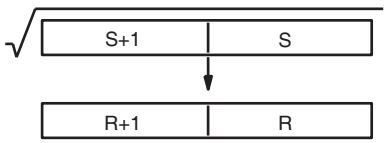
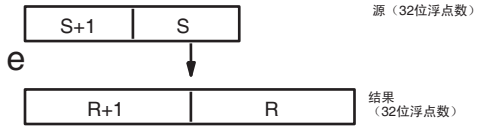
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
二进制平方根	ROTB @ ROTB	620	ROTB(620) S R S: 源首字 R: 结果字	算出指定字的32位二进制数的平方根，并且把结果的整数部分输出到指定结果字中。 	输出	要
BCD 平方根	ROOT @ ROOT	072	ROOT(072) S R S: 源首字 R: 结果字	求出8位数BCD码的平方根，并且把结果的整数部分输出到指定结果字中。 	输出	要

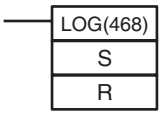
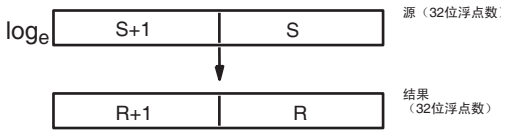
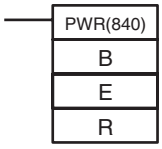
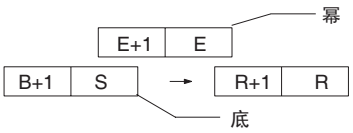
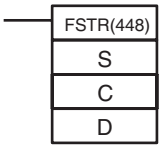
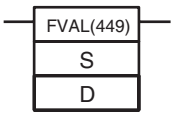
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
算术处理	APR @ APR	069	APR(069) C S R C: 控制字 S: 源数据 R: 结果字	计算源数据的正弦、余弦或线性逼近。 线性逼近功能允许 X， Y 之间的关系用线段来模拟。	输出	要
浮点数除法	FDIV @ FDIV	079	FDIV(079) Dd Dr R Dd: 被除数首字 Dr: 除数首字 R: 结果首字	一个7位浮点数和另一个浮点数相除。 浮点数用科学法表示（7位尾数和1位指数）。 	输出	要
位计数器	BCNT @ BCNT	067	BCNT(067) N S R N: 字数 S: 源首字 R: 结果字	在指定的字中对所有ON位计数。 	输出	要

A-1-13 浮点数运算指令

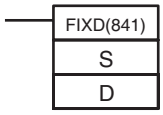
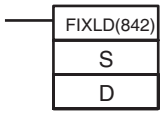
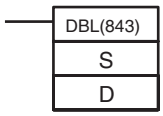
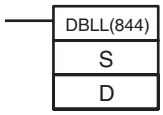
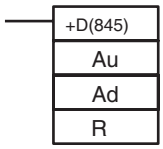
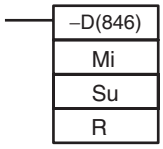
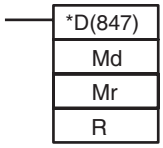
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
浮点 传送（单个）	MOVF	469	MOVF(469) S D S: 源首字 D: 目的地首字	将指定的 32 位浮点数传送到指定目的字。 	输出	要
浮点数 → 16 位	FIX @ FIX	450	FIX(450) S R S: 源首字 R: 结果字	把一个32位浮点数据转换成带符号的16位二进制数据，并且把结果存放进指定的结果字中。 	输出	要
浮点数 → 32 位	FIXL @ FIXL	451	FIXL(451) S R S: 源首字 R: 结果首字	把一个32位浮点数据转换成带符号的32位二进制数据，并且把结果存放进指定的结果字中。 	输出	要
16 位 →浮点数	FLT @ FLT	452	FLT(452) S R S: 源字 R: 结果首字	把带符号的16位二进制数据转换成一个32位浮点数据，并且把结果存放进指定的结果字中。 	输出	要
32 位→浮点数	FLTL @ FLTL	453	FLTL(453) S R S: 源首字 R: 结果首字	把带符号的32位二进制数据转换成一个32位浮点数据，并且把结果存放进指定的结果字中。 	输出	要
浮点数加法	+F @ +F	454	+F(454) Au Ad R Au: 被加数首字 AD: 加数首字 R: 结果首字	把两个32位浮点数相加，并且把结果存放进指定的结果字中。 	输出	要
浮点数减法	-F @ -F	455	–F(455) Mi Su R Mi: 被减数首字 Su: 减数首字 R: 结果首字	把一个32位浮点数从另一个32位浮点数中减去，并且把结果存放进指定的结果字中。 	输出	要

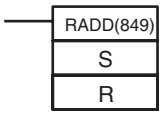
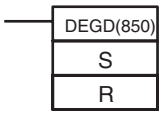
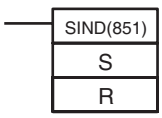
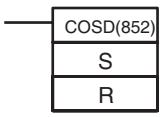
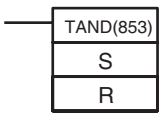
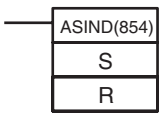
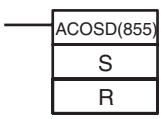
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
浮点数乘法	*F @ *F	456	*F(456) Md Mr R Md: 被乘数首字 Mr: 乘数首字 R: 结果首字	把二个32位浮点数据相乘，并且把结果存放进指定的结果字中。 	输出	要
浮点数除法	/F @ /F	457	/F(457) Dd Dr R Dd: 被除数首字 Dr: 除数首字 R: 结果首字	把一个32位浮点数据除以另一个32位浮点数据，并且把结果存放进指定的结果字中。 	输出	要
度 →弧度	RAD @ RAD	458	RAD(458) S R S: 源首字 R: 结果首字	把一个32位浮点数从度变为弧度， 并且把结果存放进指定的结果字中。 	输出	要
弧度 →度	DEG @ DEG	459	DEG(459) S R S: 源首字 R: 结果首字	把一个32位浮点数从弧度变为度， 并且把结果存放进指定的结果字中。 	输出	要
正弦	SIN @ SIN	460	SIN(460) S R S: 源首字 R: 结果首字	求出一个32位浮点数（弧度）的正弦,并且把结果存放进指定的结果字中。 	输出	要
高速正弦	SINQ @ SINQ	475	SINQ (475) S R S: 源首字 R: 结果首字	求出一个32位浮点数（弧度）的正弦,并且把结果存放进指定的结果字中。 	输出	要
余弦	COS @ COS	461	COS(461) S R S: 源首字 R: 结果首字	求出一个32位浮点数（弧度）的余弦,并且把结果存放进指定的结果字中。 	输出	要

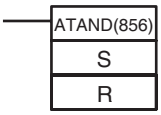
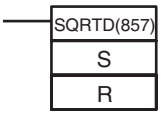
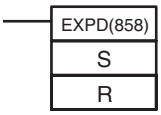
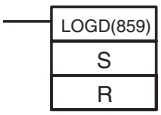
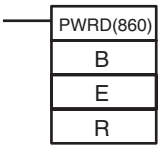
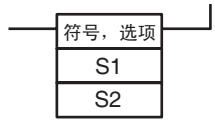
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
高速余弦	COSQ @ COSQ	476	COSQ(476) S R S: 源首字 R: 结果首字	求出一个32位浮点数（弧度）的余弦,并且把结果存放进指定的结果字中。 	输出	要
正切	TAN @ TAN	462	TAN(462) S R S: 源首字 R: 结果首字	求出一个32位浮点数（弧度）的正切，并且把结果存放进指定的结果字中。 	输出	要
高速正切	TANQ @ TANQ	477	TANQ(477) S R S: 源首字 R: 结果首字	求出一个32位浮点数（弧度）的正切，并且把结果存放进指定的结果字中。 	输出	要
反正弦	ASIN @ ASIN	463	ASIN(463) S R S: 源首字 R: 结果首字	求出一个32位浮点数的反正弦值，并且把结果存放进指定的结果字中。（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）	输出	要
反余弦	ACOS @ ACOS	464	ACOS(464) S R S: 源首字 R: 结果首字	求出一个32位浮点数的反余弦值，并且把结果存放进指定的结果字中。（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）	输出	要
反正切	ATAN @ ATAN	465	ATAN(465) S R S: 源首字 R: 结果首字	求出一个32位浮点数的反正切值，并且把结果存放进指定的结果字中。（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）	输出	要
平方根	SQRT @ SQRT	466	SQRT(466) S R S: 源首字 R: 结果首字	求出一个32位浮点数的平方根，并且把结果存放进指定的结果字中。 	输出	要
指数	EXP @ EXP	467	EXP(467) S R S: 源首字 R: 结果首字	求一个32位浮点数的自然指数（底为e），并且把结果存放进指定的结果字中。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
对数	LOG @ LOG	468	 S: 源首字 R: 结果首字	求一个32位浮点数的自然对数（底为e）， 并且把结果存放进指定的结果字中。 	输出	要
指数幂	PWR @ PWR	840	 B: 基础首字 E: 指数首字 R: 结果首字	自乘一个32位浮点数到另一个32位浮点数的幂。 	输出	要
浮点符号比较	LD, AND. 或 OR + =F, <>F, <F, <=F, >F, 或 >=F	329 (=F), 330 (<>F), 331 (<F), 332 (<=F), 333 (>F), 334 (>=F)	使用LD:  使用AND:  使用OR:  S1: 比较数据 1 S2: 比较数据 2	比较指定的单精度数据 （32 位）或常数，并且在比较结果是真时，产生一个 ON 执行条件。 浮点符号比较指令可用：LD （载入）， AND 和 OR。	LD： 逻辑开始 AND 或 OR：行线上继续	LD： 不要 AND 或 OR： 要
浮点数→ ASCII	FSTR @ FSTR	448	 S: 源首字 C: 控制字 D: 目的地字	把指定格式 （32 位小数点或指数格式）的单精度浮点数据转换成字符串数据 （ASCII），并且把结果存入目的字中。	输出	要
ASCII →浮点数	FVAL @ FVAL	449	 S: 源字 D: 目的地首字	把指定格式 （小数点或指数格式）表达的单精度浮点数数据字符串数据 （ASCII）转换成 32 位单精度浮点数数据，并且把结果存入目的字中。	输出	要

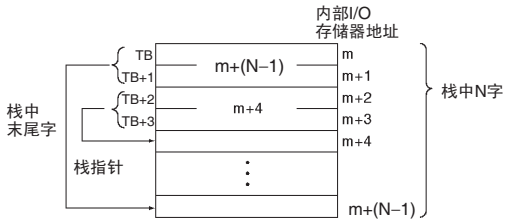
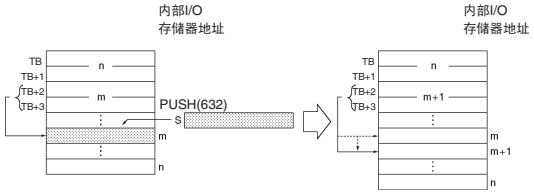
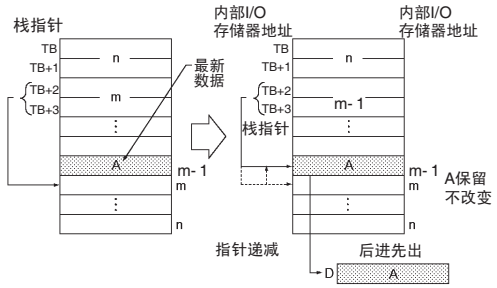
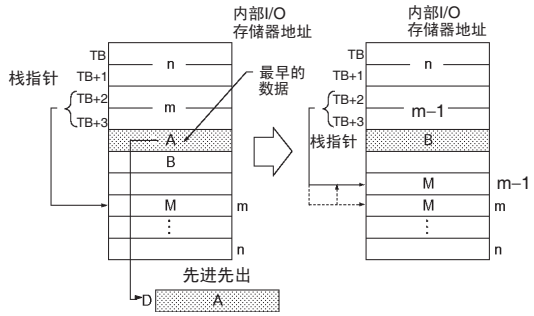
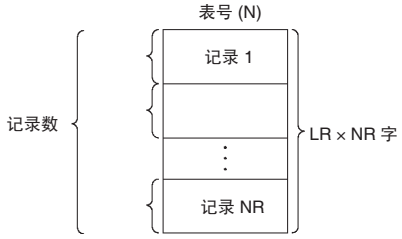
A-1-14 双精度浮点数指令

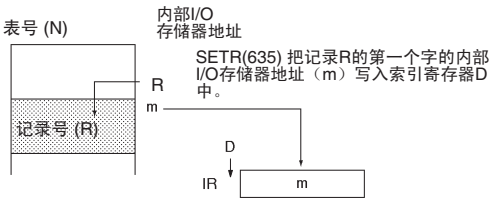
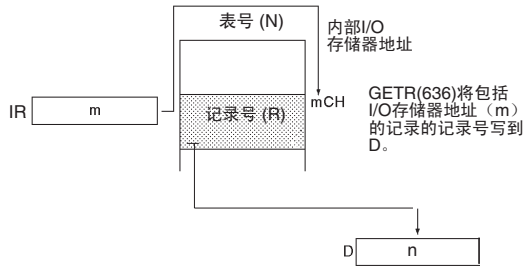
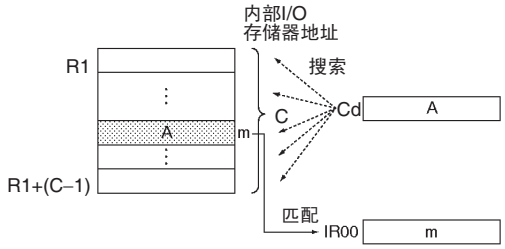
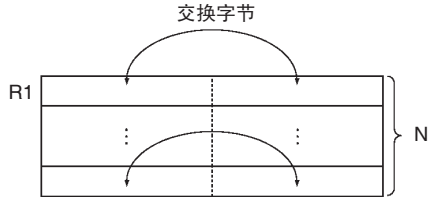
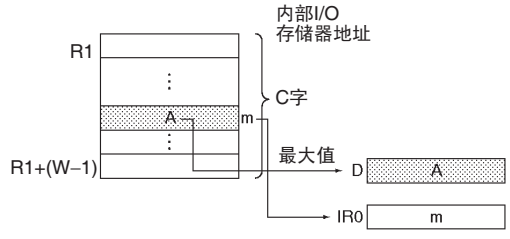
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字长浮点数→16 位二进制	FIXD @ FIXD	841	 S: 源首字 D: 目的字	把一个指定的双精度浮点数据 （64 位）转换成 16 位二进制数据，并且把结果存入到目的字中。	输出	要
双字长浮点数→32 位二进制	FIXLD @ FIXLD	842	 S: 源首字 D: 目的首字	把一个指定的双精度浮点数据 （64 位）转换成 32 位带符号二进制数据，并且把结果存入到目的字中。	输出	要
16 位二进制 → 双字长浮点数	DBL @ DBL	843	 S: 源字 D: 目的首字	把一个指定的 16 位带符号二进制数据转换成双精度浮点数据 （64 位），并且把结果存入到目的字中。	输出	要
32 位二进制→双字长浮点数	DBLL @ DBLL	844	 S: 源首字 D: 目的首字	把一个指定的 32 位带符号二进制数据转换成双精度浮点数据 （64 位），并且把结果存入到目的字中。	输出	要
双字长浮点数加	+D @ +D	845	 Au: 被加数首字 Ad: 加数首字 R: 结果首字	把两个指定的双精度浮点数值 （每个数均为 64 位）相加，并且把结果存入到结果字中。	输出	要
双字长浮点数减	-D @ -D	846	 Mi: 被减数首字 Su: 减数首字 R: 结果首字	把两个指定的双精度浮点数值 （每个数均为 64 位）相减，并且把结果存入到结果字中。	输出	要
双字长浮点数乘	*D @ *D	847	 Md: 被乘数首字 Mr: 乘数首字 R: 结果首字	把两个指定的双精度浮点数值 （每个数均为 64 位）相乘，并且把结果存入到结果字中。	输出	要

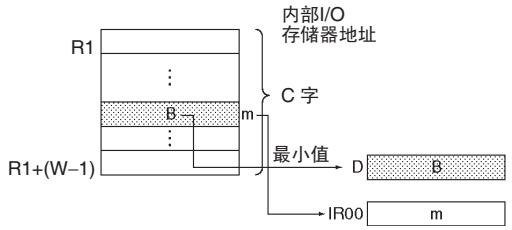
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字长浮点数除	/D @ /D	848	 Dd: 被除数首字 Dr: 除数首字 R: 结果首字	把两个指定的双精度浮点数值 （每个数均为 64 位）相除，并且把结果存入到结果字中。	输出	要
双字长度→弧度	RADD @ RADD	849	 S: 源首字 R: 结果首字	把一个指定的双精度浮点数 （64 位）从度变为弧度，并且把结果存入到结果字中。	输出	要
双字长弧度→度	DEGD @ DEGD	850	 S: 源首字 R: 结果首字	把一个指定的双精度浮点数据 （64 位）从弧度变为度，并且把结果存入到结果字中。	输出	要
双字长正弦	SIND @ SIND	851	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据 （弧度）的正弦，并且把结果存入到结果字中。	输出	要
双字长余弦	COSD @ COSD	852	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据 （弧度）的余弦，并且把结果存入到结果字中。	输出	要
双字长正切	TAND @ TAND	853	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据 （弧度）的正切，并且把结果存入到结果字中。	输出	要
双字长反正弦	ASIND @ ASIND	854	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据的反正弦值（弧度），并且把结果存入到结果字中。（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）。	输出	要
双字长反余弦	ACOSD @ ACOSD	855	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据的反余弦值（弧度），并且把结果存放到指定的结果字中。（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）。	输出	要

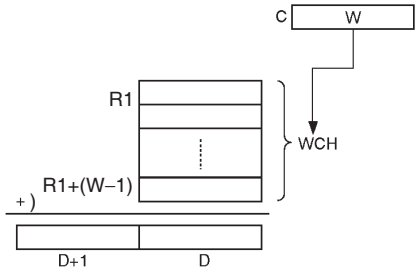
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字长反正切	ATAND @ ATAND	856	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据的反正切值（弧度），并且把结果存入到结果字中。（反正切函数与正切函数正好相反，它返回一个角度值，该角度产生正切值）。	输出	要
双字长平方根	SQRTD @ SQRTD	857	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据的平方根值，并且把结果存入到结果字中。	输出	要
双字长指数	EXPD @ EXPD	858	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据的自然指数（底为 e），并且把结果存入到结果字中。	输出	要
双字长对数	LOGD @ LOGD	859	 S: 源首字 R: 结果首字	求出一个指定的双精度 （64 位）浮点数据的自然对数（底为 e），并且把结果存入到结果字中。	输出	要
双字长指数幂	PWRD @ PWRD	860	 B: 基础首字 E: 指数首字 R: 结果首字	自乘一个指定的双精度 （64 位）浮点数到另一个指定的双精度 （64 位）浮点数，并且把结果存入到结果字中。	输出	要
双字长符号比较	LD, AND. 或 OR + =D, <>D, <D, <=D, >D, 或 >=D	LD, AND. 或 OR + 335 (=D), 336 (<>D), 337 (<D), 338 (<=D), 339 (>D), 340 (>=D)	使用 LD:  使用 AND:  使用 OR:  S1: 比较数据 1 S2: 比较数据 2	比较指定的双精度数据 （64 位）并且在比较结果是真时，产生一个 ON 执行条件。 双精度浮点符号比较指令可用三种符号：LD （载入），AND 和 OR。	LD： 逻辑开始 AND 或 OR：行线上继续	LD： 不要 AND 或 OR： 要

A-1-15 表格数据处理指令

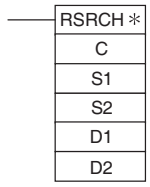
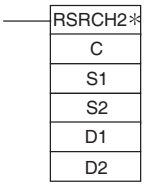
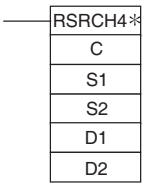
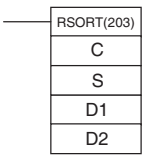
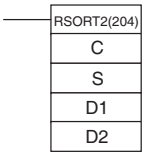
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
堆栈设置	SSET @ SSET	630	SSET(630) TB N TB: 栈首址 N: 编号字	从指定字开始定义一个栈的指定长度，并初始化此数据区的字全为零。 	输出	要
压入栈	PUSH @ PUSH	632	PUSH(632) TB S TB: 栈首址 S: 源字	把一个字的数据写入指定栈。 	输出	要
后进后出	LIFO @ LIFO	634	LIFO(634) TB D TB: 栈首址 D: 目的字	读出被写入指定栈中的最后一个数据（栈中最新数据）。 	输出	要
先进先出	FIFO @ FIFO	633	FIFO(633) TB D TB: 栈首址 D: 目的字	读出被写入指定栈中的第一个数据（栈中最早的数据）。 	输出	要
定义记录表	DIM @ DIM	631	DIM(631) N LR NR TB N: 表号 LR: 每个记录长度 NR: 记录数 TB: 栈首址	通过规定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。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
设置记录位置	SETR @ SETR	635	SETR(635) N R D N: 表号 R: 记录号 D: 目的索引寄存器	把指定记录的位置（第一个记录的内部I/O存储器地址）写进指定的索引寄存器中。 	输出	要
获得记录号	GETR @ GETR	636	GETR(636) N IR D N: 表号 IR: 索引寄存器 D: 目的字	返回包含在指定索引寄存器内的内部I/O存储器地址中记录的记录号。 	输出	要
数据搜索	SRCH @ SRCH	181	SRCH(181) C R1 Cd C: 控制首字 R1: 范围首字 Cd: 比较数据	在指定字的范围内搜索一个字的数据。 	输出	要
交换字节	SWAP BYTES	637	SWAP(637) N R1 N: 字数 R1: 范围首字	将范围内的所有字的左字节和右字节交换。 	输出	要
寻找最大值	MAX @ MAX	182	MAX(182) C R1 D C: 控制首字 R1: 范围首字 D: 目的字	在范围内寻找最大值。 	输出	要
寻找双字长数据最大值	MAXL @ MAXL	174	MAXL(174) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	以 R1 指定的范围中首字开始，将由 C 指定的数据项目数作为双字表数据。并将表中的最大值输出到 D+1 和 D。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
寻找浮点数据最大值	MAXF @ MAXF	176	MAXF(176) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	以 R1 指定的范围中首字开始，将由 C 指定的数据项目数作为单精度浮点数据表（双字数据）。并将表中的最大值输出到 D+1 和 D。	输出	要
寻找双字长浮点数据最大值	MAXD @ MAXD	178	MAXD(178) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	以 R1 指定的范围中首字开始，将由 C 指定的数据项目数作为双精度浮点数据表（双字数据）。并将表中的最大值输出到 D+1 和 D。	输出	要
寻找最小值	MIN @ MIN	183	MIN(183) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	在范围内寻找最小值。 	输出	要
寻找双字长数据最小值	MINL @ MINL	175	MINL(175) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	以 R1 指定的范围中首字开始，将由 C 指定的数据项目数作为双字表数据。并将表中的最小值输出到 D+1 和 D。	输出	要
寻找浮点数据最小值	MINF @ MINF	177	MINF(177) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	以 R1 指定的范围中首字开始，将由 C 指定的数据项目数作为单精度浮点数据表（双字数据）。并将表中的最小值输出到 D+1 和 D。	输出	要
寻找双字长浮点数据最小值	MIND @ MIND	179	MIND(179) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	以 R1 指定的范围中首字开始，将由 C 指定的数据项目数作为双精度浮点数据表（双字数据）。并将表中的最小值输出到 D+1 和 D。	输出	要

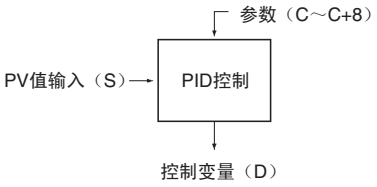
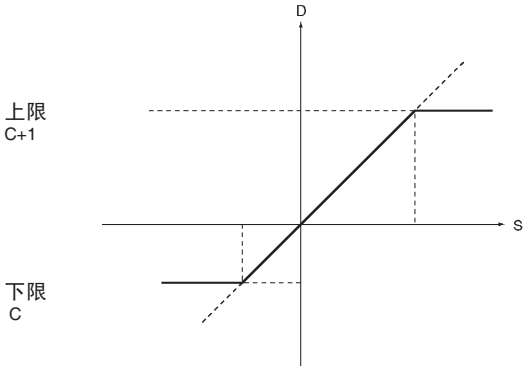
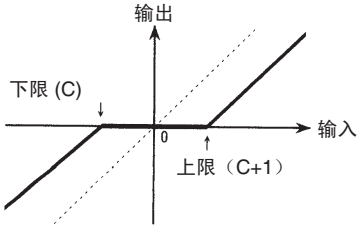
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
求和	SUM @ SUM	184	SUM(184) C R1 D C: 控制首字 R1: 范围首字 D: 目的首字	把范围内的字节或字相加并把结果输出到2个字中。 	输出	要
读栈大小	SNUM @ SNUM	638	SNUM(638) TB D TB: 栈首址 D: 目的字	在指定的栈中计栈数据量 （字数）。	输出	要
读栈数据	SREAD @ SREAD	639	SREAD(639) TB C D TB: 栈首址 C: 偏移值 D: 目的字	从栈中指定的数据元素读数据。偏移值表明了所需数据元素的位置（在当前指针位置前多少个数据元素）。	输出	要
覆盖栈数据	SWRIT @ SWRIT	640	SWRIT(640) TB C S TB: 栈首址 C: 偏移值 S: 源数据	把源数据写入栈中指定的数据元素中 （覆盖原来的数据）。偏移值表明所需数据元素的位置 （在当前指针位置前多少个数据元素）。	输出	要
插入栈数据	SINS @ SINS	641	SINS(641) TB C S TB: 栈首址 C: 偏移值 S: 源数据	把源数据插入栈中指定的数据元素中，并将栈中其余的数据向下移。偏移值指明了插入点的位置 （在当前指针位置前多少个数据元素）。	输出	要
删除栈数据	SDEL @ SDEL	642	SDEL(642) TB C D TB: 栈首址 C: 偏移值 D: 目的字	把栈中指定位置处的数据元素删除，并将栈中其他的数据向上移。偏移值指明了删除点的位置 （在当前指针位置前多少个数据元素）。	输出	要

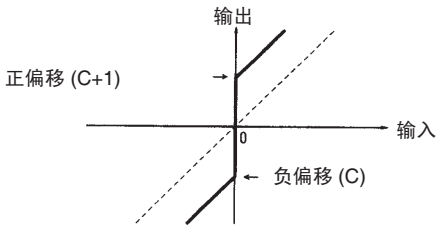
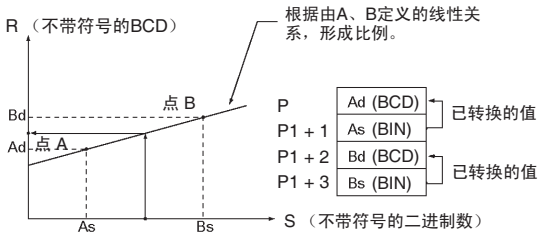
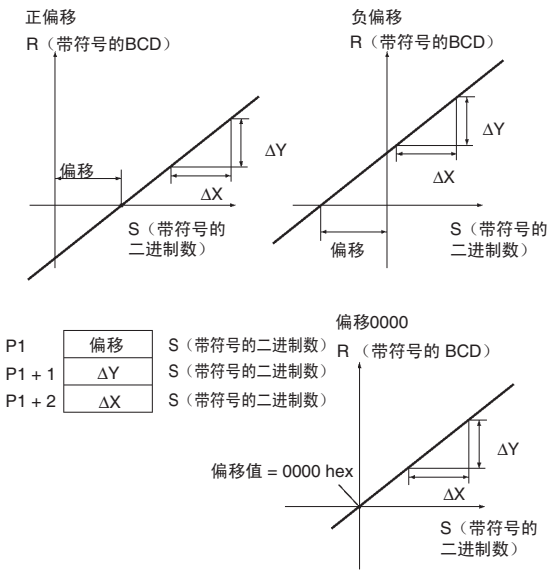
A-1-16 跟踪指令

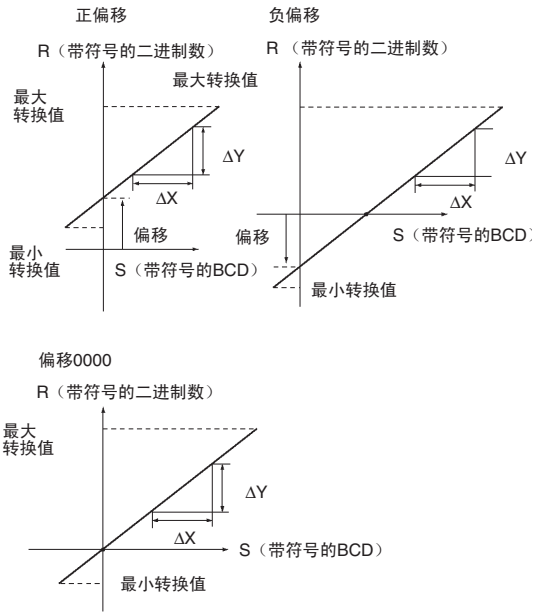
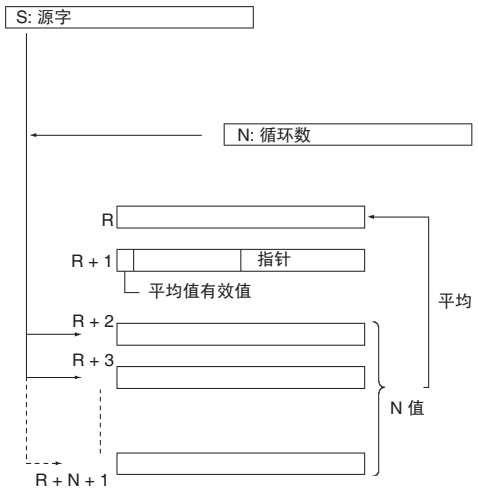
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
不带符号的单字记录搜索指令	RSRCH + <, <=, =, >, >= @ RSRCH + <, <=, =, >, >=	360（<）， 361（<=）， 362（=）， 363（>）， 364（>=）	 C: 控制首字 S1: 寻找的首记录的首字 S2: 寻找数据 D1: 目的首字 D2: 目的索引寄存器	从由 S1 指定的表开始，不带符号的单字记录搜索指令寻找由 S2 指定的数据 （1 字）。当记录与指定搜索条件匹配时，顺序地将记录号与数据输出到 D1。当索引寄存器指定用于 D2，则将相匹配记录的地址输出到索引寄存器。不使用索引寄存器输出时，将 D2 设定为 #00000000。	输出	要
不带符号的双字记录搜索指令	RSRCH2 + <, <=, =, >, >= @ RSRCH2 + <, <=, =, >, >=	370（<）， 371（<=）， 372（=）， 373（>）， 374（>=）	 C: 控制首字 S1: 寻找的首记录的首字 S2: 寻找数据首字 D1: 目的首字 D2: 目的索引寄存器	从由 S1 指定的表开始，不带符号的双字记录搜索指令寻找由 S2 指定的数据 （2 字）。当记录与指定搜索条件匹配时，顺序地将记录号与数据输出到 D1。当索引寄存器指定用于 D2，则将相匹配记录的地址输出到索引寄存器。不使用索引寄存器输出时，将 D2 设定为 #00000000。	输出	要
不带符号的四字记录搜索指令	RSRCH4 + <, <=, =, >, >= @ RSRCH4 + <, <=, =, >, >=	380（<）， 381（<=）， 382（=）， 383（>）， 384（>=）	 C: 控制首字 S1: 寻找的首记录的首字 S2: 寻找数据 D1: 目的首字 D2: 目的索引寄存器	从由 S1 指定的表开始，不带符号的四字记录搜索指令寻找由 S2 指定的数据 （4 字）。当记录与指定搜索条件匹配时，顺序地将记录号与数据输出到 D1。当索引寄存器指定用于 D2，则将相匹配记录的地址输出到索引寄存器。不使用索引寄存器输出时，将 D2 设定为 #00000000。	输出	要
不带符号的单字记录分类	RSORT @ RSORT	203	 C: 控制首字 S: 分类的首记录的首字 D1: 分类结果首字 D2: 目的索引寄存器	根据控制字，对由 S 指定的表中记录 （1 字）进行分类。	输出	要
不带符号的双字记录分类	RSORT2 @ RSORT2	204	 C: 控制首字 S: 分类的首记录的首字 D1: 分类结果首字 D2: 目的索引寄存器	根据控制字，对由 S 指定的表中记录 （2 字）进行分类。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
不带符号的四字记录分类	RSORT4 @ RSORT4	205	RSORT4(205) C S D1 D2 C: 控制首字 S: 分类的首记录的首字 D1: 分类结果首字 D2: 目的索引寄存器	根据控制字，对由 S 指定的表中记录 （4 字）进行分类。	输出	要

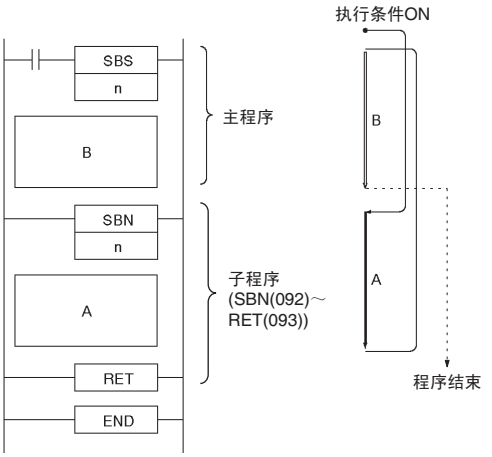
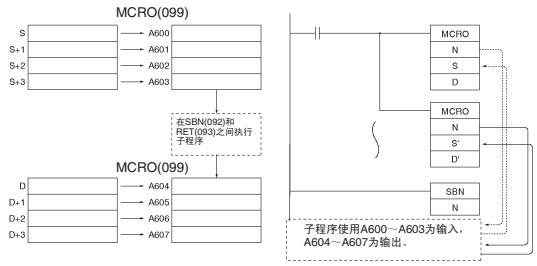
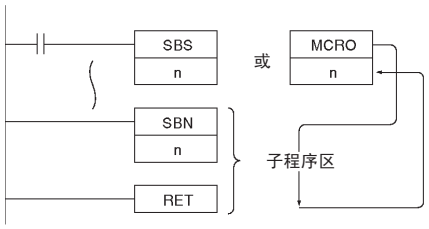
A-1-17　数据控制指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
PID 控制	PID	190	PID(190) S C D S: 输入字 C: 参数首字 D: 输出字	根据指定参数进行PID控制。 	输出	要
带自整定的 PID 控制	PIDAT	191	PIDAT(191) S C D S: 输入字 C: 参数首字 D: 输出字	根据指定的参数执行 PID 控制。 PID 常数可用 PIDAT （191）指令自整定。	输出	要
限位控制	LMT @ LMT	680	LMT(680) S C D S: 输入字 C: 限位首字 D: 输出字	根据输入数据是否在上、下限之内，控制输出数据。 	输出	要
静带控制	BAND @ BAND	681	BAND(681) S C D S: 输入字 C: 限位首字 D: 输出字	根据输入数据是否在静带范围内控制输出数据。 	输出	要

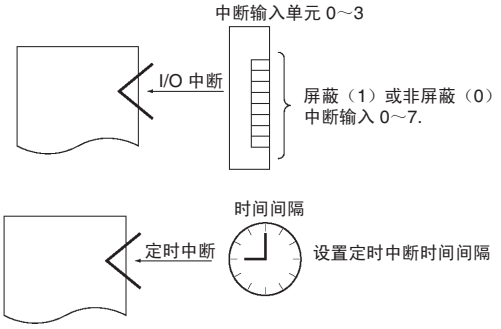
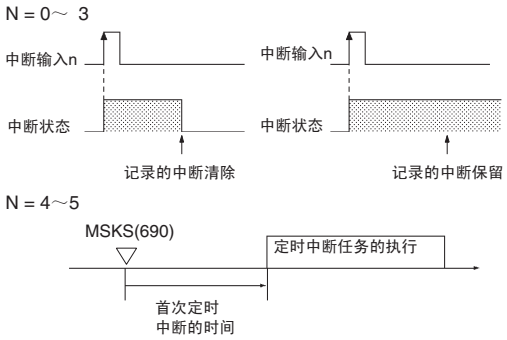
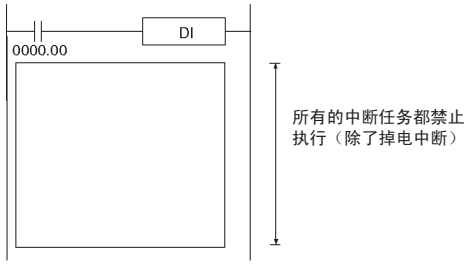
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
静域控制	ZONE @ ZONE	682	ZONE(682) S C D S: 输入字 C: 限位首字 D: 输出字	把指定的偏移值和输入数据相加并输出结果。 	输出	要
时间比例输出	TPO	685	TPO (685) S C R S: 输入字 C: 参数首字 R: 脉冲输出位	从指定字输入占空比或操作变量，根据指定的参数将占空比转换为时间比例输出，并从指定输出对结果进行输出。	输出	要
标度	SCL @ SCL	194	SCL(194) S P1 R S: 源字 P1: 参数首字 R: 结果字	根据指定的线性关系，把不带符号的二进制数据转换成不带符号的BCD数据。 	输出	要
标度 2	SCL2 @ SCL2	486	SCL2(486) S P1 R S: 源字 P1: 参数首字 R: 结果字	根据指定的线性关系，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。 	输出	要

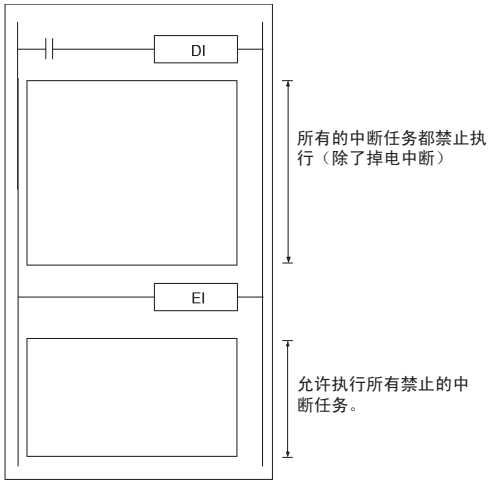
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
标度 3	SCL3 @ SCL3	487	SCL3(487) S P1 R S: 源字 P1: 参数首字 R: 结果字	根据指定的线性函数，把带符号的BCD数据转换成带符号的二进制函数。偏移值可在定义线性函数时输入。  	输出	要
平均值	AVG	195	AVG(195) S N R S: 源字 N: 循环数 R: 结果字	求出指定循环数目的输入字的平均值。 	输出	要

A-1-18 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
子程序调用	SBS @ SBS	091	SBS(091) N N: 子程序编号	调用指定子程序号的子程序并执行它。 	输出	要
宏	MCRO @ MCRO	099	MCRO(099) N S D N: 子程序编号 S: 输入参数首字 D: 输出参数首字	调用指定子程序号的子程序，并使用S～S+3中的输入参数执行子程序，输出参数至D～D+3。 	输出	要
子程序进入	SBN	092	SBN(092) N N: 子程序编号	指示指定子程序号的子程序的开始。 	输出	要
子程序返回	RET	093	RET(093)	指示子程序结束。	输出	要
全局子程序调用	GSBS	750	GSBS(750) N N: 子程序编号	调用指定子程序号的子程序并执行它。	输出	要
全局子程序进入	GSBN	751	GSBN(751) N N: 子程序编号	指示指定子程序号的子程序的开始。	输出	要
全局子程序返回	GRET	752	GRET(752)	指示子程序结束。	输出	要

A-1-19 中断控制指令

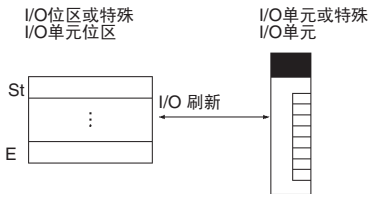
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
设置中断屏蔽	MSKS @ MSKS	690	MSKS(690) N C N: 中断编号 C: 控制数据	对I/O中断或定时中断设置中断处理。在PC刚上电时，I/O中断和定时中断都被屏蔽（禁止）。MSKS(690)可用于非屏蔽或屏蔽I/O中断，以及设置定时中断的时间间隔。 	输出	要
读中断屏蔽	MSKR @ MSKR	692	MSKR(692) N D N: 中断编号 D: 目的字	读取由 MSKS （690）设定的当前中断处理设定。	输出	要
清除中断	CLI @ CLI	691	CLI(691) N C N: 中断编号 C: 控制数据	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。 	输出	要
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	要

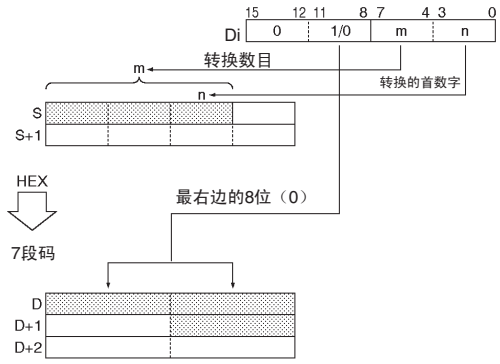
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
允许中断	EI	694	EI(694)	允许执行所有由DI（693）所禁止的中断任务。 	输出	要

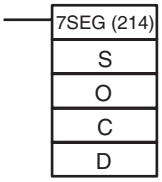
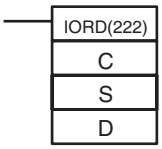
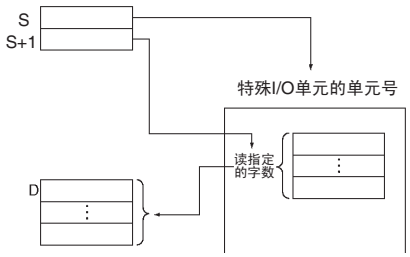
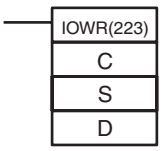
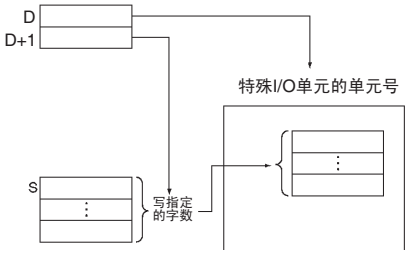
A-1-20 步指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
步定义	STEP	008	STEP(008) B B: 位	STEP （008）以下列 2 种方式作用 , 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）结束该步程序区 （例 : 步执行）。	输出	要
步启动	SNXT	009	SNXT(009) B B: 位	SNXT （0009）用于下列三种情况 : （1）开始步程序执行。 （2）继续到下一个步的控制位。 （3）结束步程序执行。	输出	要

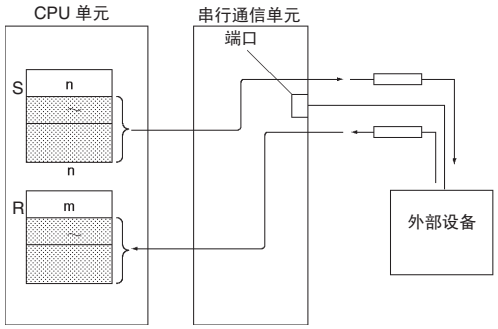
A-1-21 基本 I/O 单元指令

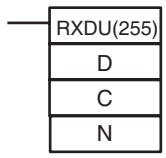
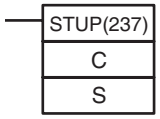
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
I/O 刷新	IORF @ IORF	097	IORF(097) St E St: 起始字 E: 结束字	刷新指定的I/O字。 	输出	要
特殊 I/O 单元 I/O 刷新	FIORF @ FIORF	225	FIORF(225) N N: 单元号	根据指定的单元号，立即刷新分配到特殊 I/O 单元的 I/O 字。	输出	要
CPU 总线单元 I/O 刷新	DLNK @ DLNK	226	DLNK(226) N N: 单元号	根据指定的单元号，立即刷新分配到 CPU 总线单元的 I/O 字。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
7 段译码	SDEC @ SDEC	078	SDEC(078) S Di D S: 源字 Di: 数字指定器 D: 目标首字	把指定数字中的十六进制数字转换成相应的8位7段显示码，并把它存入指定目标字中的高或低8位。 	输出	要
数字开关输入	DSW	210	DSW (210) I O D C1 C2 I: 数据输入字（D0～D3） O: 输出字 D: 结果首字 C1: 数字量 C2: 系统字	读取连接到输入单元或输出单元的外部数字开关（或指轮开关）上的值，并在指定字中保存 4 位或 8 位的 BCD 数据。	输出	要
十键输入	TKY	211	TKY (211) I D1 D2 I: 数据输入字 D1: 寄存器首字 D2: 键输入字	从连接到一个输入单元的十键键盘读取数值，并在指定字中保存最多 8 位的 BCD 数据。	输出	要
十六进制键输入	HKY	212	HKY (212) I O D C I: 数据输入字 O: 输出字 D: 寄存器首字 C: 系统字	从连接到一个输入单元和一个输出单元的十六进制键盘读取数值，并在指定字中保存最多 8 位的十六进制数据。	输出	要
矩阵输入	MTR	213	MTR (213) I O D C I: 数据输入字 O: 输出字 D: 目的首字 C: 系统字	从连接到一个输入单元和输出单元（使用 8 输入点和 8 输出点）的一个 8 × 8 矩阵输入最多 64 个信号，并在 4 个目的字中保存 64 位数据。	输出	要

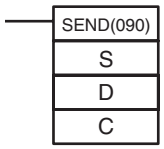
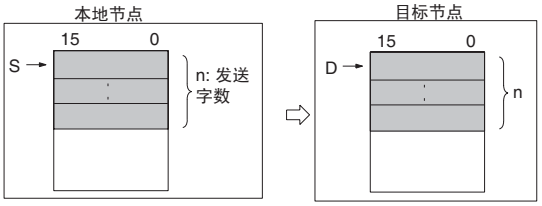
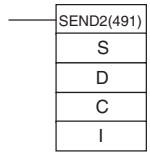
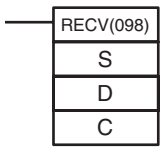
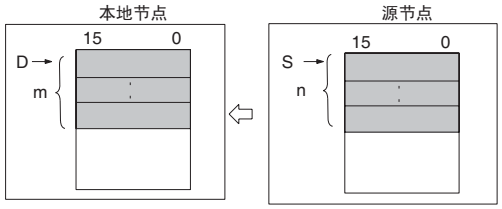
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
7 段显示输出	7SEG	214	 S: 源首字 O: 输出字 C: 控制字 D: 系统字	将源数据 （4 位或 8 位的 BCD）转换为 7 段码显示数据，并将数据输出到指定的输出字。	输出	要
智能 I/O 读	IORD @ IORD	222	 C: 控制数据 S: 传送源和字数 D: 传送目标和字数	读取特殊I/O单元或CPU总线单元存储器区的内容（见注）。 	输出	要
智能 I/O 写	IOWR @ IOWR	223	 C: 控制数据 S: 传送源和字数 D: 传送目标和字数	输出CPU单元I/O存储器区的内容到特殊I/O单元或CPU总线单元（见注）。 	输出	要

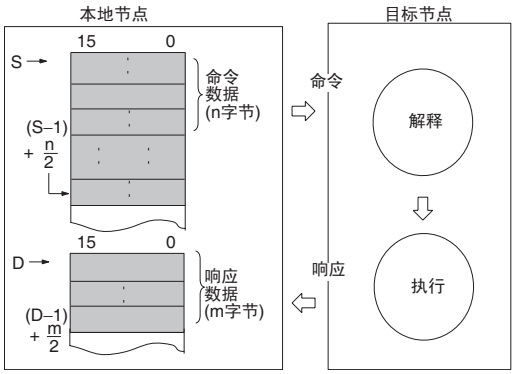
A-1-22 串行通信指令

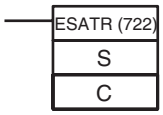
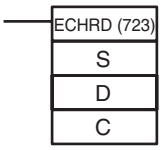
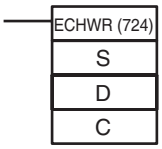
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
协议宏	PMCR @ PMCR	260	PMCR(260) C1 C2 S R C1: 控制字 1 C2: 控制字 2 S: 发送首字 R: 接收首字	调用和执行在串行通信板（仅CS系列）或串行通信单元中登录过的通信序列。 	输出	要
协议宏 2	PMCR2 @ PMCR2	264	PMCR2(264) C1 C2 S R I C1: 控制字 1 C2: 控制字 2 S: 发送首字 R: 接收首字 I: 通信信息首字	最多可同时使用 64 条 PMCR2 指令，使用同 PMCR。	输出	要
发送	TXD @ TXD	236	TXD(236) S C N S: 源首字 C: 控制字 N: 字节数 0000～0100 hex （0 ～256十进制数）	根据 PLC 设置中无协议模式的指定起始码 / 结束码，从 CPU 单元内置的 RS-232C 端口 （无协议模式）、串行通信板的串行端口或单元版本 1.2 及以上的单元 （无协议模式），输出指定字节数的数据 （无转换）。	输出	要
接收	RXD @ RXD	235	RXD(235) D C N D: 目的首字 C: 控制字 N: 保存字节数 0000～ 0100 hex (0～256十进制数)	根据 PLC 设置中无协议模式的指定起始码 / 结束码，使用从 CPU 单元内置的 RS-232C 端口 （无协议模式）、串行通信板的串行端口或单元版本 1.2 及以上的单元 （无协议模式）的指定首字开始，读取指定字节数的数据。	输出	要
经由串行通信单元传送	TXDU @ TXDU	256	TXDU(256) S C N S: 源首字 C: 控制首字 N: 字节数 (0000～0256 BCD)	根据所分配的 DM 区中指定起始码 / 结束码 （无协议模式），从一个单元版本 1.2 及以上的串行通信单元的串行端口 （无协议模式），输出指定字节数的数据 （无转换）。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
经由串行通信单元接收	RXDU @ RXDU	255	 D: 目的首字 C: 控制首字 N: 保存字节数 (0000～0100 hex)	根据所分配的 DM 区中指定起始码 / 结束码 （无协议模式），从单元版本 1.2 及以上的串行通信单元的串行端口，以指定的首字开始读取指定字节数的数据 （无转换）。	输出	要
修改串行口设置	STUP @ STUP	237	 C: 控制字 (端口) S: 源首字	修改 CPU 单元，串行通信单元 （CPU 总线单元），或串行通信板中串行输出口的通信参数。在 PLC 操作中， STUP （237）能使协议发生改变。	输出	要

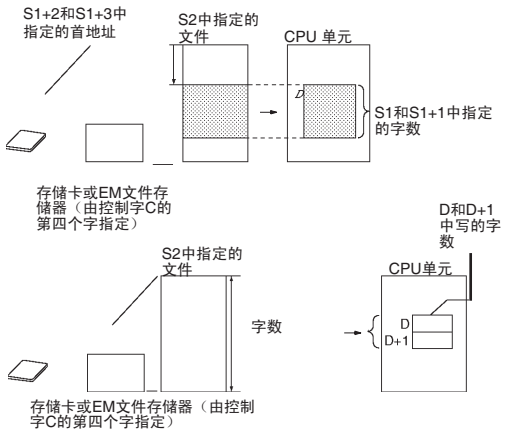
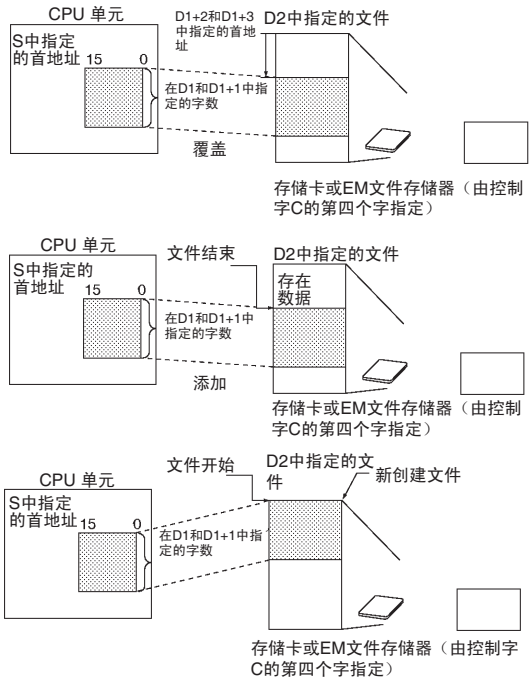
A-1-23 网络指令

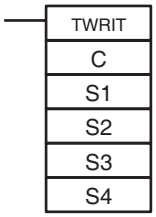
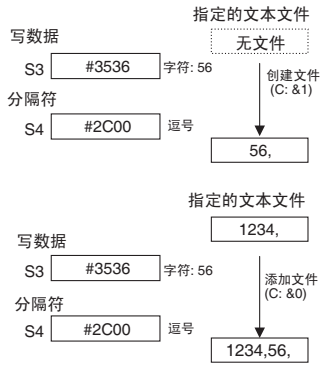
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
网络发送	SEND @ SEND	090	 S: 源首字 D: 目的首字 C: 控制首字	把数据传送到网中的一个节点。 	输出	要
网络发送 2	SEND2 @ DEND2	491	 S: 源首字 D: 目的首字 C: 控制首字 I: 通信信息首字	最多可同时使用 64 条 SEND2 指令，使用同 SEND。	输出	要
网络接收	RECV @ RECV	098	 S: 源首字 D: 目的首字 C: 控制首字	要求网络中的一个节点传送数据并接收此数据。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
网络接收 2	RECV2 @ RECV2	492	RECV2(492) S D C I S: 源首字 D: 目的首字 C: 控制首字 I: 通信信息首字	最多可同时使用 64 条 RECV2 指令，使用同 RECV。	输出	要
发布命令	CMND @ CMND	490	CMND(490) S D C S: 命令首字 D: 响应首字 C: 控制首字	发送FINS命令并接收响应。  	输出	要
发布命令 2	CMND2 @ CMND2	493	CMND2(493) S D C I S: 命令首字 D: 响应首字 C: 控制首字 I: 通信信息首字	最多可同时使用 64 条 CMND2 指令，使用同 CMND。	输出	要
显式报文发送	EXPLT	720	EXPLT (720) S D C S: 发送报文首字 D: 接收报文首字 C: 控制首字	以任何服务代码发送显式报文。	输出	要
显式获取属性	EGATR	721	EGATR (721) S D C S: 发送报文首字 D: 接收报文首字 C: 控制报文首字	读取带显式报文的状态信息 （获取单一属性，服务代码：0E hex）。	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
显式设置属性	ESATR	722	 S: 发送报文首字 C: 控制首字	写入带显式报文的状态信息 （设置单一属性，服务代码：0E hex）。	输出	要
显式字读取	ECHRD	723	 S:远程CPU单元中的源首字 D:本地CPU单元中的目的首字 C:控制首字	从网络中的远程 CPU 单元将数据读取到本地 CPU 单元 （远程 CPU 单元必须支持显式报文）。	输出	要
显式字写入	ECHWR	724	 S:本地CPU单元中的源首字 D:远程CPU单元中的目的首字 C:控制首字	从本地 CPU 单元将数据写入到网络中的远程 CPU 单元 （远程 CPU 单元必须支持显式报文）。	输出	要

A-1-24　文件存储指令

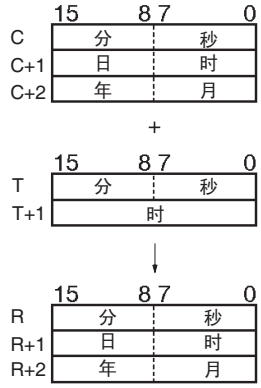
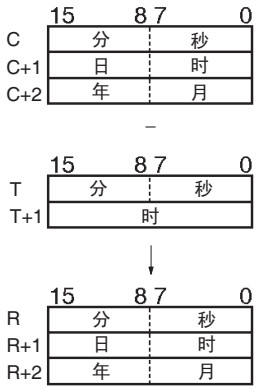
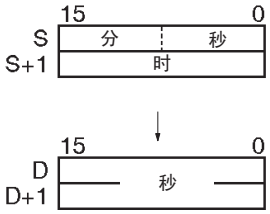
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
读数据文件	FREAD @ FREAD	700	FREAD(700) C S1 S2 D C: 控制字 S1: 源首字 S2: 文件名 D: 目的首字	把指定的数据或数据量从文件存储器的指定数据文件读取到CPU单元的指定数据区中。  	输出	要
写数据文件	FWRIT @ FWRIT	701	FWRIT(701) C D1 D2 S C: 控制字 D1: 目的首字 D2: 文件名 S: 源首字	从CPU单元中的数据区，以指定的数据内容覆盖或添加数据到文件存储器中指定的数据文件。若指定的文件不存在，则以相应文件名创建新文件。  	输出	要

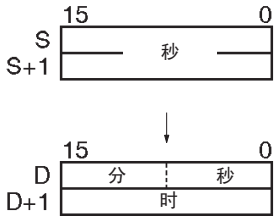
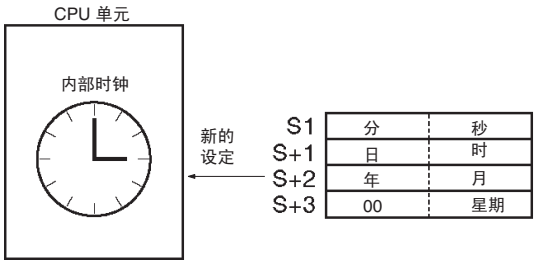
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
写文本文件	TWRIT @ TWRIT	704	 C: 控制字 S1: 写入字节数 S2: 文件名和目录 S3: 写数据 S4: 分隔符	从 I/O 存储器读取 ASCII 数据并以文本文件将数据保存在存储卡中（写入新文件或添加文件）。数据保存为 TXT 格式。  	输出	要

A-1-25 显示指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
显示报文	MSG @ MSG	046	MSG(046) N M N: 报文号 M: 报文首字	读指定的 16 个扩展 ASCII 字 , 并且在外围设备 （如编程器）上显示该输出报文。	输出	要

A-1-26 时钟指令

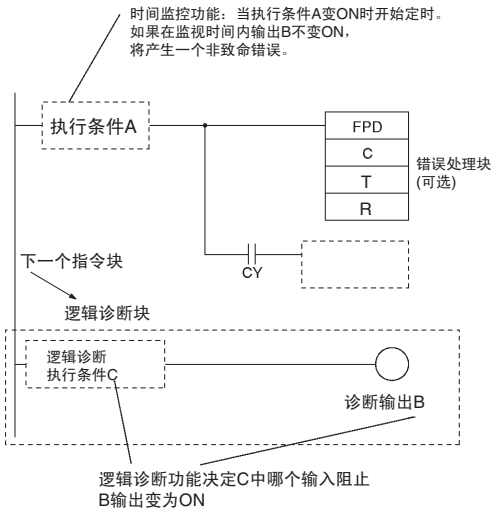
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
日历加法	CADD @ CADD	730	CADD(730) C T R C: 日历首字 T: 时间首字 R: 结果首字	把指定字中的日历数据和时间相加。 	输出	要
日历减法	CSUB @ CSUB	731	CSUB(731) C T R C: 日历首字 T: 时间首字 R: 结果首字	把指定字中的日历数据和时间相减。 	输出	要
小时→秒	SEC @ SEC	065	SEC(065) S D S: 源首字 D: 目的首字	将以小时/分/秒表示的时间转换成仅以秒表示的等值时间。 	输出	要

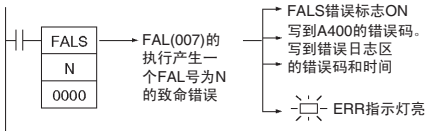
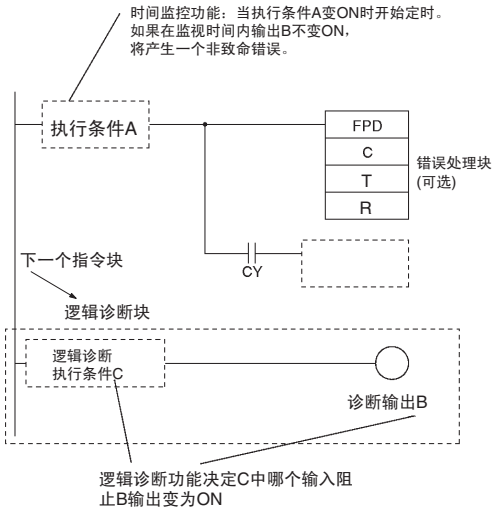
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
秒→小时	HMS @ HMS	066	HMS(066) S D S: 源首字 D: 目的首字	将以秒表示的时间转换成以小时/分/秒表示的等值时间。 	输出	要
时钟调整	DATE @ DATE	735	DATE(735) S S: 源首字	改变内部时钟设定为指定源字中的设定。 	输出	要

A-1-27 调试指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
跟踪存储器采样	TRSM	045	TRSM(045)	当执行 TRSM （045）时，预定位或字的状态被采样并存于跟踪存储器中。 TRSM （045）可用于程序中的任何地方，且使用任意次。	输出	不要

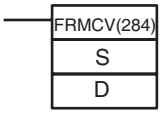
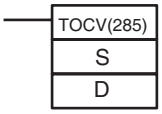
A-1-28 故障诊断指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
故障报警	FAL @ FAL	006	FAL(006) N S N：FAL号 S：产生消息首字 或错误代码	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。也产生系统非致命错误 	输出	要

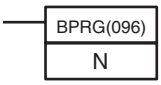
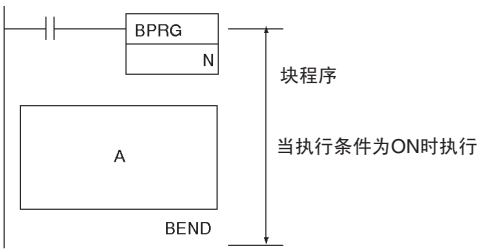
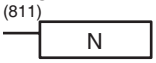
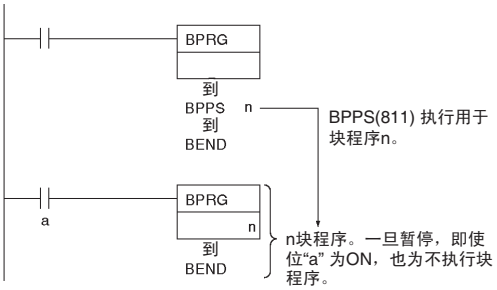
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
严重故障报警	FALS	007	FALS(007) N S N：FAL号 S：产生报文首字或错误代码	产生用户定义的致命错误， 此错误使PC机停止运行，并在系统中产生致命错误。 	输出	要
故障点检测	FPD	269	FPD(269) C T R C: 控制字 T: 监视时间 R: 寄存器首字	通过监视在FPD(269)执行和诊断输出执行之间的时间,诊断一个指令块中的错误,并且找出哪个输入阻止输出变ON。 	输出	要

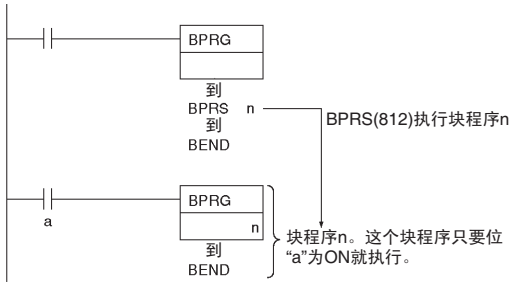
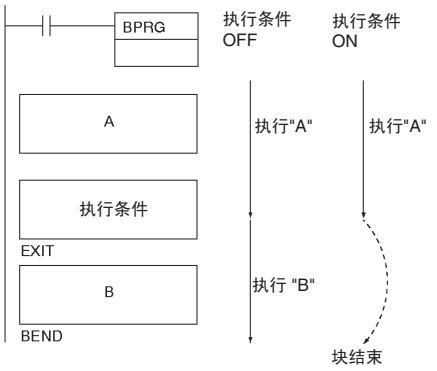
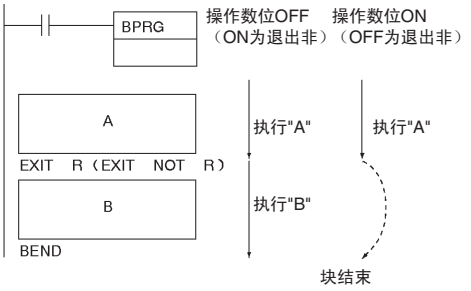
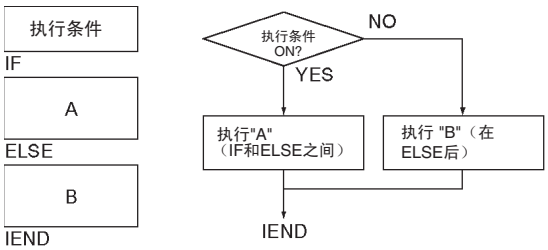
A-1-29　其它指令

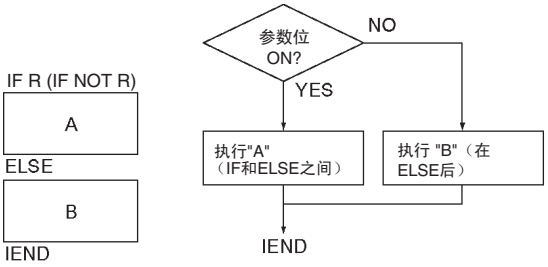
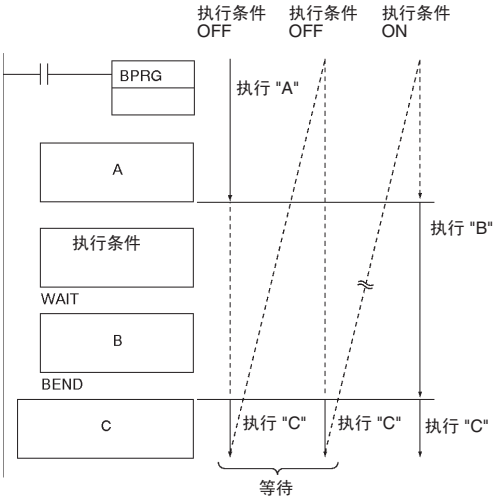
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
设置进位	STC @ STC	040	STC(040)	设置进位标志 （CY）。	输出	要
清除进位	CLC @ CLC	041	CLC(041)	进位标志 （CY）置 OFF。	输出	要
选择 EM 区	EMBC @ EMBC	281	EMBC(281) N N: EM区号	改变当前 EM 区。	输出	要
延长最大循环时间	WDT @ WDT	094	WDT(094) T T: 定时器设定	延长最大循环时间 , 但仅在此指令执行的循环时间。	输出	要
存储条件标志	CCS @ CCS	282	CCS(282)	存储条件标志状态。	输出	要
载入条件标志	CCL @ CCL	283	CCL(283)	读取已保存的条件标志状态。	输出	要

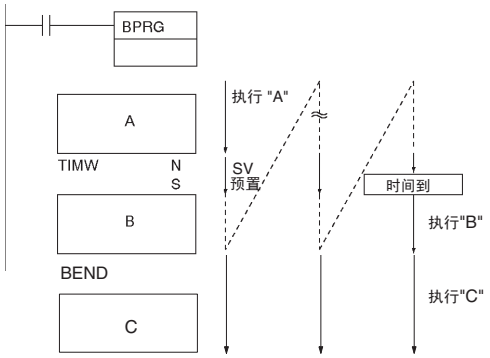
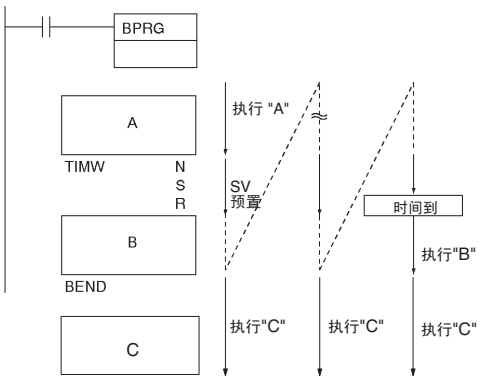
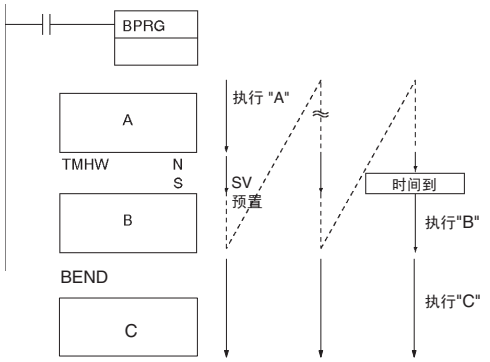
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
从 CV 变换地址	FRMCV @ FRMCV	284	 S:含有CV系列存储器地址的字 D: 目的索引寄存器	把 CV 系列 PLC 存储器地址变换到等同的 CS/CJ 系列 PLC 存储器地址。	输出	要
把地址变换到 CV	TOCV @ TOCV	285	 S: 含有CV系列存储器地址的字 D: 目的字	把 CS/CJ 系列 PLC 存储器地址变换到等同的 CV 系列 PLC 存储器地址。	输出	要

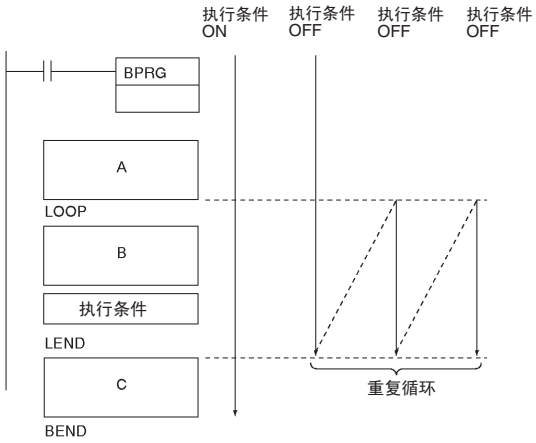
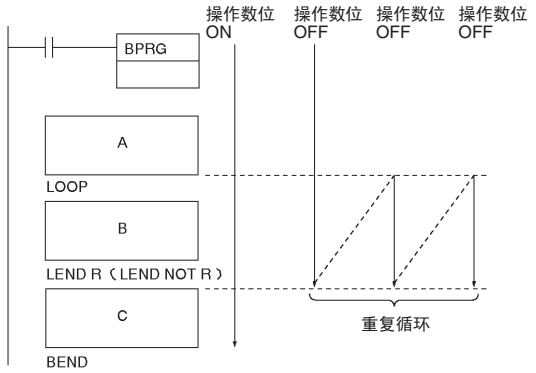
A-1-30 块程序指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
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)。 	输出	要
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）。	块程序	要
块程序暂停	BPPS	811	BPPS (811)  N: 块程序号	从另一个块程序暂停指定的块程序。 	块程序	要

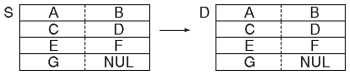
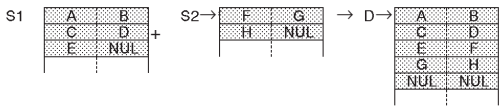
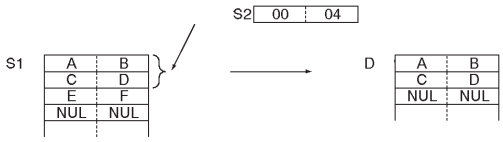
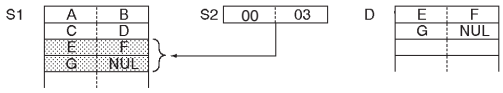
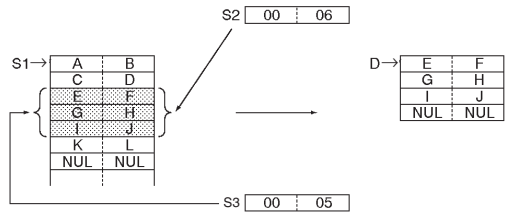
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
块程序重启	BPRS	812	BPRS (812) N N: 块程序号	从另一个块程序重新启动指定的块程序。 	块程序	要
条件块退出	EXIT	806	EXIT （806） B：位操作数	如果执行条件为ON，没有操作数的EXIT(806)将退出程序。 	块程序	要
条件块退出	EXIT	806	EXIT （806）B B：位操作数	如果执行条件为ON，没有操作数的EXIT(806)将退出程序。 	块程序	要
条件块退出非	EXIT NOT	806	EXIT NOT （806） B B：位操作数	如果执行条件为 OFF，没有操作数的 EXIT （806）将退出程序。	块程序	要
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，那么将执行 ELSE(803)和IEND(804)之间的指令。 	块程序	要

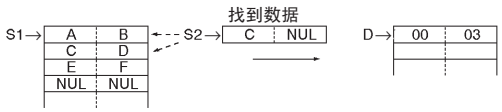
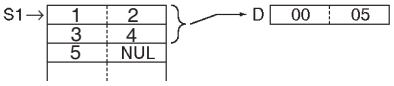
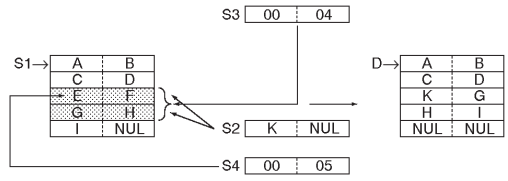
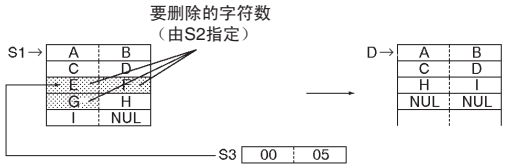
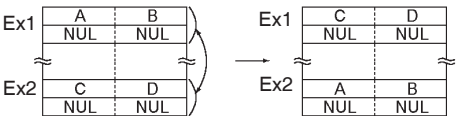
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
条件块分支	IF	802	IF （802） B B：位操作数	如果操作数位为ON，那么就执行IF(802)和ELSE(803) 之间的指令，如果操作数位为OFF，那么将执行 ELSE(803)和IEND(804)之间的指令。 	块程序	要
条件块分支（非）	IF NOT	802	IF （802）NOT B B：位操作数	如果操作数位为 ON, 那么就执行 IF （802）和 ELSE （803）之间的指令，如果操作数位 OFF, 那么将执行 ELSE （803）和 IEND （804）之间的指令。	块程序	要
条件块分支（ELSE）	ELSE	803	——	如果忽略 ELSE （803）指令，且操作数位为 ON, 那么就执行 IF （802 和 IEND （804）之间的指令。	块程序	要
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	要
一个循环与等待	WAIT	805	WAIT （805）	如果WAIT(805)的执行条件为OFF，那么将跳过块程序中的其余指令。 	块程序	要
一个循环与等待	WAIT	805	WAIT （805） B B：位操作数	如果操作数位为 OFF （WAIT NOT （805）为 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令。	块程序	要
一个循环与等待 （非）	WAIT NOT	805	WAIT （805）NOT B B：位操作数	如果操作数位为 OFF （WAIT NOT （805）为 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令。	块程序	要

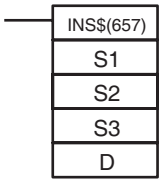
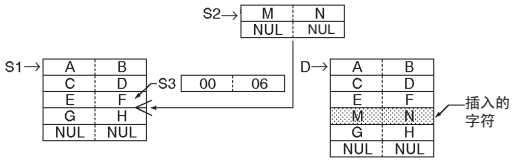
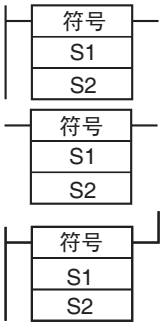
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
定时器等待	TIMW （BCD）	813	TIMW （813） N SV N：定时器号 SV：设定值	延迟执行块程序，直至指定的定时时间到。当经过定时器定时时间后，块程序继续执行TIMW(813)/TIMWX(816) 之后的下一个指令。 SV: 用BCD表示0～999.9，用二进制表示0～6,553.5。 	块程序	要
	TIMWX （二进制）	816	TIMWX （816） N SV N：定时器号 SV：设定值			
计数器等待	CNTW （BCD）	814	CNTW （814） N SV N：计数器号 SV：设定值 I：　计数输入	延迟执行块程序的剩余指令，直至到达指定的计数。当达到指定计数后，块程序继续执行CNTW(814)/CNTWX(817) 之后的下一个指令。 SV: 用BCD表示0～9,999，用二进制表示0～65,535。 	块程序	要
	CNTWX （二进制）	818	CNTWX （818） N SV N：计数器号 SV：设定值 I：　计数输入			
高速定时器等待	TMHW （BCD）	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，直至指定的定时时间到。当经过定时器定时时间后，块程序继续执行TMHW(815)之后的下一个指令。 SV: 用BCD表示0～99.99，用二进制表示0～655.35。 	块程序	要
	TMHWX （二进制）	817	TMHW （817） N SV N：　定时器号 SV：设定值			

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
循环	LOOP	809	——	LOOP(809)指定循环程序的开始 	块程序	要
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），直到 LEND （810）或 LEND （810） NOT 的操作数位分别变为 ON 或 OFF，或直到 LEND （810）的执行条件变为 ON。	块程序	要
循环结束	LEND	810	LEND （810） B B：位操作数	如果LEND(810)的操作数位为OFF (或LEND(810) NOT的操作数位为ON)，则循环执行从 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LEND(810) NOT的操作数位为OFF)，则循环将结束，并且从LEND(810) 或LEND(810) NOT后的下一条指令开始执行。  注　用于LEND(810) NOT的操作数位状态为相反。	块程序	要
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），直到 LEND （810）或 LEND （810） NOT 的操作数位分别变为 ON 或 OFF，或直到 LEND （810）的执行条件变为 ON。	块程序	要

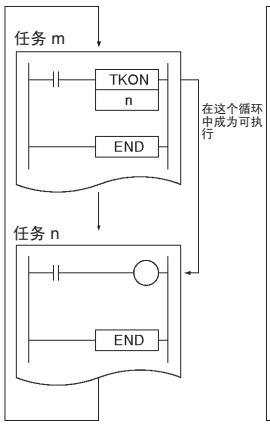
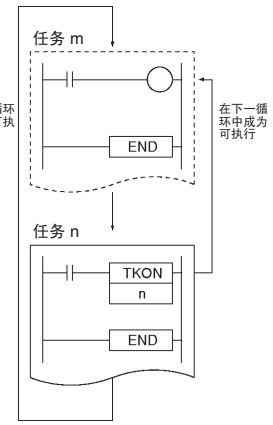
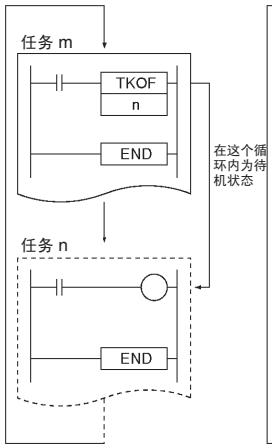
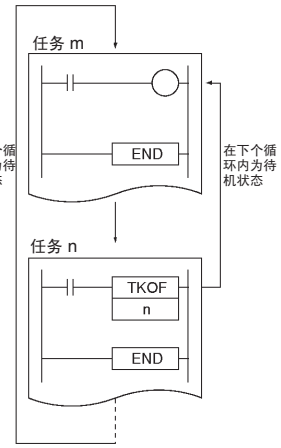
A-1-31　文本字符串处理指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
串发送	MOV$ @ MOV$	664	MOV$(664) S D S: 源首字 D: 目的首字	传送文本串。 	输出	要
链接串	+$ @ +$	656	+$(656) S1 S2 D S1: 文本串 1 S2: 文本串 2 D:　目的首字	把一个文本串与另一个文本串链接。 	输出	要
取左串	LEFT$ @ LEFT$	652	LEFT$(652) S1 S2 D S1: 文本串首字 S2: 字符数 D:　目的首字	从一文本串的左边（开始）取一指定的字符数。 	输出	要
取右串	RGHT$ @ RGHT$	653	RGHT$(653) S1 S2 D S1: 文本串首字 S2: 字符数 D: 目的首字	从一文本串的右边（结束）取一指定的字符数。 	输出	要
取中间串	MID$ @ MID$	654	MID$(654) S1 S2 S3 D S1: 文本串首字 S2: 字符数 S3: 开始位置 D:　目的首字	从一文本串中间任何位置读取一指定的字符数。 	输出	要

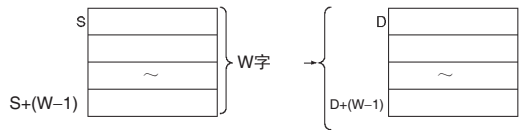
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
寻找串	FIND @ FIND$	660	FIND$(660) S1 S2 D S1: 文本串源首字 S2: 寻找文本串首字 D: 目的首字	在一文本串中找寻一指定的文本串。 	输出	要
计算串长	LEN$ @ LEN$	650	LEN$(650) S D S: 文本串首字 D: 目的首字	计算文本串的长度。 	输出	要
取代串	RPLC$ @ RPLC$	661	RPLC$(654) S1 S2 S3 S4 D S1: 文本串首字 S2: 取代的文本串首字 S3: 字符数 S4: 首位置 D: 目的首字	用一指定的文本串从指定位置取代一个文本串。 	输出	要
删除串	DEL$ @ DEL$	658	DEL$(658) S1 S2 S3 D S1: 文本串首字 S2: 字符数 S3: 开始位置 D: 目的首字	从文本串的中间删除指定的文本串。 	输出	要
交换串	XCHG$ @ XCHG$	665	XCHG$(665) Ex1 Ex2 Ex1: 交换首字 1 Ex2: 交换首字 2	用另一个指定的文本串取代一个指定的文本串。 	输出	要
清除串	CLR$ @ CLR$	666	CLR$(666) S S: 文本串首字	用NUL （00 hex）清除整个文本串。 	输出	要

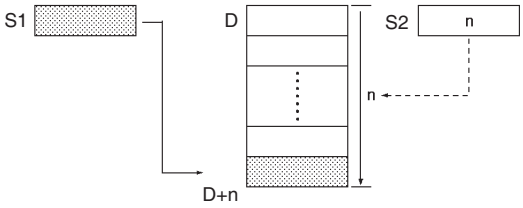
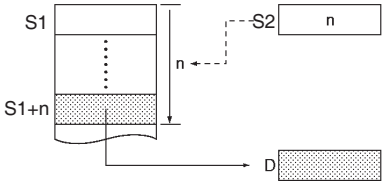
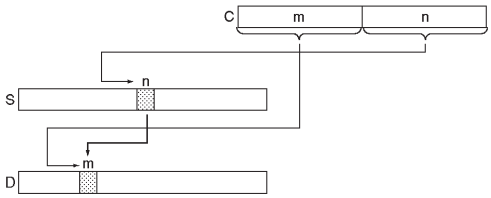
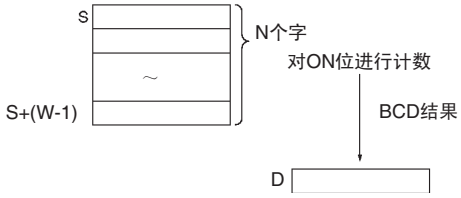
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
插入串	INS$ @ INS$	667	 S1: 基于文本串首字 S2: 插入文本串首字 S3: 开始位置 D: 目标首字	从文本串的中间插入一指定的文本串。 	输出	要
串比较	LD, AND, OR+ =$, <>$, <$, <=$, >$, >=$	670 (=$) 671 (<>$) 672 (<$) 673 (<=$) 674 (>$) 675 (>=$)	 S1: 文本串 1 S2: 文本串 2	根据 ASCII 代码值 , 串比较指令 （=$, <>$, <$, <=$, >$, >=$）从开始比较两个文本串。如果比较结果为真 , 那么就为 LOAD， AND，或 OR 产生一个 ON 执行条件。	LD：逻辑启动 AND, OR: 行线上继续	要

A-1-32 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
任务 ON	TKON @ TKON	820	TKON(820) N N: 任务号	使指定的任务可执行。 指定任务的任务号比本地任务的任务号高 (m<n)。  指定任务的任务号比本地任务的任务号低 (m>n)。 	输出	要
任务 OFF	TKOF @ TKOF	821	TKOF(821) N N: 任务号	使指定的任务为待机状态。 指定任务的任务号比本地任务的任务号高 (m<n)。  指定任务的任务号比本地任务的任务号低 (m>n)。 	输出	要

A-1-33 型号转换指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
块传送	XFERC @ XFERC	565	XFERC(565) N S D N: 字数 S: 源首字 D: 目的首字	传送指定数量的连续字。 	输出	要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
单字属性	DISTC @ DISTC	566	DISTC(566) S Bs Of S: 源首字 Bs: 目的基础地址 Of: 偏移量	将源字传送至目的字（通过增加偏移量到基础地址计算出）。  亦可写入堆栈中（压栈操作）。	输出	要
数据收集	COLLC @ COLLC	567	COLLC(567) Bs Of D Bs: 源基础地址 Of: 偏移量 (BCD) D: 目的字	将源字传送至目的字（通过增加偏移量到基础地址计算出）。  亦可从堆栈读取数据（堆栈读取操作）。	输出	要
移动位	MOVBC @ MOVBC	568	MOVBC(568) S C D S: 源字或数据 C: 控制字(BCD) D: 目的字	传送指定位 	输出	要
位计数器	BCNTC @ BCNTC	621	BCNTC(621) N S R N: 字数 (BCD) S: 源首字 R: 结果字	在指定字中计算打开ON位的总数。 	输出	要

A-1-34　特殊功能块指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
获取可变 ID	GETID @ GETID	286	GETID(286) S D1 D2 S: 变量或地址 D1: ID代码 D2: 目的字	对指定变量或地址输出 FINS 命令可变类型 （数据区）代码和字地址。该指令通常用于获取功能块中某个变量分配到的地址。	输出	要

A-2 指令执行时间和步数

下表列举了 CPU 单元所支持的所有指令的执行时间。

整个用户程序中指令总的执行时间，是指计算循环时间时的程序执行处理时间 （*1）。

*1 用户程序是分配给循环任务中的可执行任务和满足中断条件的中断任务。

根据使用的 CPU 单元不同以及指令执行的条件不同，大多数指令执行时间是不同的。

当执行条件是 OFF 时，执行时间也会变化。

下表中在长度 （步长）列中显示出各指令的长度。用户程序中每一条指令所需的步数不同，根据指令和它使用的操作数来决定。

程序中的步数不等同于指令数。

注 1 大多数指令支持微分形式 （用↑ ↓@和 % 表示）。指令指定微分形式将增加下列指令执行时间。

（单位：μs）

符号	CJ2 CPU 单元
	CPU6 □ H-EIP
↑或↓	+0.24
@或 %	+0.24

2 当指令不执行时，取下列时间作为指标。

（单位：μs）

CJ2 CPU 单元
CPU6 □ H-EIP
0.016

执行时间根据所执行指令的顺序以及其执行与否而异。使用下述计算公式计算最差情况下的最大变量。

指令步数× 0.016 （μs）

A-2-1 顺序输入指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
装载	LD	——	1 ～ 2	0.016	——
	!LD	——	2 ～ 14	0.99	——
装载非	LD NOT	——	1 ～ 2	0.016	——
	!LD NOT	——	2 ～ 14	0.99	——
与	AND	——	1 ～ 2	0.016	——
	!AND	——	2 ～ 14	0.99	——
与非	AND NOT	——	1 ～ 2	0.016	——
	!AND NOT	——	2 ～ 14	0.99	——
或	OR	——	1 ～ 2	0.016	——
	!OR	——	2 ～ 14	0.99	——
或非	OR NOT	——	1 ～ 2	0.016	——
	!OR NOT	——	2 ～ 14	0.99	——
与装载	AND LD	——	1	0.016	——
或装载	OR LD	——	1	0.016	——
非	NOT	520	1	0.016	——
ON 条件	UP	521	3	0.24	——
OFF 条件	DOWN	522	4	0.24	——
位测试	LD TST	350	4	0.11	——
位测试非	LD TSTN	351	4	0.11	——
与位测试非	AND TSTN	351	4	0.11	——
或位测试	OR TST	350	4	0.11	——
或位测试非	OR TSTN	351	4	0.11	——

A-2-2 顺序输出指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
输出	OUT	——	1 ～ 2	0.016	——
	!OUT	——	2 ～ 14	0.99	——
输出非	OUT NOT	——	1 ～ 2	0.016	——
	!OUT NOT	——	2 ～ 14	0.99	——
保持	KEEP	011	1 ～ 2	0.048	——
	!KEEP	011	16	0.99	——
上升沿微分	DIFU	013	2 ～ 2	0.21	——
下降沿微分	DIFD	014	2 ～ 2	0.21	——
置位	SET	——	1 ～ 2	0.016	——
	!SET	——	2 ～ 14	0.99	——
复位	RSET	——	1 ～ 2	0.016	——
	!RSET	——	2 ～ 14	0.99	——
多位置位	SETA	530	4	3.68	置 1 位
				15.5	置 1,000 位

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
多位复位	RSTA	531	4	3.7	复 1 位
				15.5	复 1,000 位
单一位置位	SETB	532	2	0.19	——
	!SETB	——	16	0.99	——
单一位复位	RSTB	533	2	0.19	——
	!RSTB	——	16	0.99	——
单一位输出	OUTB	534	2	0.19	——
	!OUTB	——	16	0.99	——

A-2-3 顺序控制指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
结束	END	001	1	5.0	——
空操作	NOP	000	1	0.016	——
联锁	IL	002	1	0.048	——
联锁解除	ILC	003	1	0.048	——
多联锁微分保持	MILH	517	3	5.0	联锁条件不满足 （输入条件 ON）
				6.2	联锁条件满足 （输入条件 OFF）
				4.0	联锁中再次满足联锁条件 （输入条件 OFF）
多联锁微分释放	MILR	518	3	5.0	联锁条件不满足 （输入条件 ON）
				6.2	联锁条件满足 （输入条件 OFF）
				4.0	联锁中再次满足联锁条件 （输入条件 OFF）
多联锁解除	MILC	519	2	3.4	非联锁中
				4.0	联锁中
跳转	JMP	004	2	0.31	——
跳转结束	JME	005	2	——	——
条件跳转	CJP	510	2	0.31	跳转条件满足 （输入条件 ON）
条件跳转非	CJPN	511	2	0.31	跳转条件满足 （输入条件 OFF）
多次跳转	JMP0	515	1	0.048	——
多次跳转中止	JME0	516	1	0.048	——
FOR 循环	FOR	512	2	0.18	指定一个常数
循环中止	BREAK	514	1	0.048	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
下一个循环	NEXT	513	1	0.14	当循环继续
				0.18	当循环结束

A-2-4 定时器和计数器指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
100MS 定时器	TIM	——	3	0.45	——
	TIMX	550		0.45	
10MS 定时器	TIMH	015	3	0.70	——
	TIMHX	551		0.46	
1MS 定时器	TMHH	540	3	0.69	——
	TMHHX	552		0.46	
0.1MS 定时器	TIMU	541	3	0.45	——
	TIMUX	556		0.45	
0.01MS 定时器	TMUH	544	3	0.45	——
	TMUHX	557		0.45	
累计定时器	TTIM	087	3	9.2	——
				6.9	当复位时
				5.0	当联锁时
	TTIMX	555	3	8.8	——
				6.8	当复位时
				5.0	当联锁时
长时间定时器	TIML	542	4 ～ 5	5.8	——
				3.9	当联锁时
	TIMLX	553	4 ～ 5	5.7	——
				3.6	当联锁时
多输出定时器	MTIM	543	4	6.4	——
				3.7	当复位时
	MTIMX	554	4	5.5	——
				3.4	当复位时
定时器复位	TRSET	549	2	0.32	——
计数器	CNT	——	3	0.51	——
	CNTX	546		0.51	
可逆计数器	CNTR	012	3	9.1	——
	CNTRX	548		8.0	
定时器 / 计数器复位	CNR	545	3	6.2	当复位 1 个字时
				2824	当复位 1,000 个字时
	CNRX	547	3	6.2	当复位 1 个字时
				2824	当复位 1,000 个字时

A-2-5 比较指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
符号比较 （无符号）	=	300	4	0.08	——
	<>	305			
	<	310			
	<=	315			
	>	320			
	>=	325			
符号比较 （双字长，无符号）	=L	301	4 ～ 7	0.08	——
	<>L	306			——
	<L	311			——
	<=L	316			——
	>L	321			——
	>=L	326			——
符号比较 （带符号）	=S	302	4	0.08	——
	<>S	307			
	<S	312			
	<=S	317			
	>S	322			
	>=S	327			
符号比较 （双字长，带符号）	=SL	303	4 ～ 7	0.08	——
	<>SL	308			
	<SL	313			
	<=SL	318			
	>SL	323			
	>=SL	328			
时间比较指令	=DT	341	4	16.3	——
	<>DT	342			
	<DT	343			
	<=DT	344			
	>DT	345			
	>=DT	346			
比较	CMP	020	3	0.032	——
	!CMP	020	30	2.06	——
双字比较	CMPL	060	3 ～ 5	0.064	——
带符号二进制比较	CPS	114	3 ～ 5	0.064	——
	!CPS	114	30	2.06	——
双字长带符号二进制比较	CPSL	115	3 ～ 5	0.064	——
表比较	TCMP	085	4	10.3	——
多重比较	MCMP	019	4	15.2	——
无符号块比较	BCMP	068	4	16.3	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
扩展块比较	BCMP2	502	4	5.0	数据字数：1
				217.2	数据字数：255
区域范围比较	ZCP	088	3	4.3	——
双区域范围比较	ZCPL	116	3 ～ 5	4.3	——

A-2-6 数据传送指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
传送	MOV	021	3	0.14	——
	!MOV	021	30	1.98	——
双字长传送	MOVL	498	3 ～ 4	0.26	——
取非传送	MVN	022	3	0.14	——
双字长取非传送	MVNL	499	3 ～ 4	0.26	——
位传送	MOVB	082	4	0.19	——
数字传送	MOVD	083	4	0.19	——
多位传送	XFRB	062	4	6.62	传送 1 位
				85.8	传送 255 位
块传送	XFER	070	4	0.29	传送 1 个字
				240.1	传送 1,000 个字
块置位	BSET	071	4	0.21	设置 1 个字
				142.2	设置 1,000 个字
数据交换	XCHG	073	3	0.32	——
双字长数据交换	XCGL	562	3 ～ 4	0.61	——
单字分配	DIST	080	4	4.5	——
数据收集	COLL	081	4	4.6	——
传送到寄存器	MOVR	560	3	0.064	——
将定时器 / 计数器当前值传送到寄存器	MOVRW	561	3	0.064	——

A-2-7 数据移位指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
移位寄存器	SFT	010	3	3.86	移位 1 个字
				3.18	移位 1,000 个字
可逆移位寄存器	SFTR	084	4	6.22	移位 1 个字
				457.17	移位 1,000 个字

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
异步移位寄存器	ASFT	017	4	5.3	移位 1 个字
				1357	移位 1,000 个字 *1
字移位	WSFT	016	4	3.8	移位 1 个字
				397.9	移位 1,000 个字
算术左移	ASL	025	2	0.18	——
双字左移	ASLL	570	2	0.32	——
算术右移	ASR	026	2	0.18	——
双字右移	ASRL	571	2	0.32	——
循环左移	ROL	027	2	0.18	——
双字循环左移	ROLL	572	2	0.32	——
无进位循环左移	RLNC	574	2	0.18	——
无进位双字循环左移	RLNL	576	2	0.32	——
循环右移	ROR	028	2	0.18	——
双字循环右移	RORL	573	2	0.32	——
无进位循环右移	RRNC	575	2	0.18	——
无进位双字循环右移	RRNL	577	2	0.32	——
一个数字左移	SLD	074	3	3.7	移位 1 个字
				317.8	移位 1,000 个字
一个数字右移	SRD	075	3	4.5	移位 1 个字
				479.5	移位 1,000 个字
左移 N 位数据	NSFL	578	4	4.6	移位 1 位
				31.5	移位 1,000 位
右移 N 位数据	NSFR	579	4	4.5	移位 1 位
				39.0	移位 1,000 位
左移 N 位	NASL	580	3	0.18	——
双字左移 N 位	NSLL	582	3	0.32	——
右移 N 位	NASR	581	3	0.18	——
双字右移 N 位	NSRL	583	3	0.32	——

*1 指令执行时间受数据量的很大影响，且这将影响循环时间。可通过指定后台执行以减少对循环时间的影响。详细内容，请参见 10-2-5 后台执行 。

A-2-8 递增 / 递减指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
二进制递增	++	590	2	0.18	——
双字二进制递增	++L	591	2	0.18	——
二进制递减	– –	592	2	0.18	——
双字二进制递减增	– – L	593	2	0.18	——
BCD 递增	++B	594	2	3.0	——
双字 BCD 递增	++BL	595	2	3.2	——
BCD 递减	– – B	596	2	3.0	——
双字 BCD 递减	– – BL	597	2	3.2	——

A-2-9 四则运算指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
无进位带符号二进制加法	+	400	4	0.18	——
无进位带符号双字二进制加法	+L	401	4 ～ 6	0.18	——
有进位带符号二进制加法	+C	402	4	0.18	——
有进位带符号双字二进制加法	+CL	403	4 ～ 6	0.18	——
无进位 BCD 加法	+B	404	4	4.0	——
无进位双字 BCD 加法	+BL	405	4 ～ 6	4.9	——
有进位 BCD 加法	+BC	406	4	4.4	——
有进位双字 BCD 加法	+BCL	407	4 ～ 6	5.2	——
无进位带符号二进制减法	–	410	4	0.18	——
无进位双字带符号二进制减法	– L	411	4 ～ 6	0.18	——
有进位带符号二进制减法	– C	412	4	0.18	——
有进位带符号双字二进制减法	– CL	413	4 ～ 6	0.18	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
无进位 BCD 减法	– B	414	4	4.1	——
无进位双字 BCD 减法	– BL	415	4 ～ 6	4.9	——
有进位 BCD 减法	– BC	416	4	4.5	——
有进位双字 BCD 减法	– BCL	417	4 ～ 6	5.3	——
带符号二进制乘法	*	420	4	0.26	——
带符号双字二进制乘法	*L	421	4 ～ 6	3.6	——
不带符号二进制乘法	*U	422	4	0.26	——
不带符号双字二进制乘法	*UL	423	4 ～ 6	3.6	——
BCD 码乘法	*B	424	4	3.6	——
双字 BCD 码乘法	*BL	425	4 ～ 6	4.9	——
带符号二进制除法	/	430	4	0.29	——
带符号双字二进制除法	/L	431	4 ～ 6	4.2	——
不带符号二进制除法	/U	432	4	0.29	——
不带符号双字二进制除法	/UL	433	4 ～ 6	3.8	——
BCD 码除法	/B	434	4	5.0	——
双字 BCD 码除法	/BL	435	4 ～ 6	4.8	——

A-2-10　转换指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
BCD →二进制	BIN	023	3	0.18	——
双字 BCD →双字二进制	BINL	058	3 ～ 4	3.3	——
二进制→ BCD	BCD	024	3	0.19	——
双字二进制→双字 BCD	BCDL	059	3 ～ 4	3.3	——
二进制求补	NEG	160	3	0.14	——
双字二进制求补	NEGL	161	3 ～ 4	0.26	——
16 位→ 32 位带符号二进制	SIGN	600	3	0.26	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs） CJ2H CPU6 □ -EIP	条件
数据译码	MLPX	076	4	0.29	1 位数字译码（4 ～ 6）
				0.79	4 位数字译码（4 ～ 6）
				2.9	1 位数字译码（8 ～ 256）
				4.8	4 位数字译码（8 ～ 256）
数据编码	DMPX	077	4	5.1	1 位数字编码（16 ～ 4）
				5.5	4 位数字编码（16 ～ 4）
				7.8	1 位数字编码（256 ～ 8）
				12.2	2 位数字编码（256 ～ 8）
ASCII 转换码	ASC	086	4	4.0	1 位数字转换到 ASCII
				4.6	4 位数字转换到 ASCII
ASCII 码→十六进制	HEX	162	4	3.3	转换到 1 位数字
列→行	LINE	063	4	10.5	——
行→列	COLM	064	4	13.8	——
带符号 BCD →二进制	BINS	470	4	3.6	数据格式设置 0
				3.6	数据格式设置 1
				3.6	数据格式设置 2
				3.6	数据格式设置 3
带符号双字 BCD →二进制	BISL	472	4 ～ 5	3.7	数据格式设置 0
				3.6	数据格式设置 1
				3.7	数据格式设置 2
				3.7	数据格式设置 3
带符号二进制→ BCD	BCDS	471	4	3.7	数据格式设置 0
				3.7	数据格式设置 1
				3.7	数据格式设置 2
				3.7	数据格式设置 3

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
带符号双字二进制→ BCD	BDSL	473	4 ～ 5	4.0	数据格式设置 0
				4.0	数据格式设置 1
				4.0	数据格式设置 2
				4.1	数据格式设置 3
格雷 （GRAY）码转换	GRY	474	4	26.5	8- 位二进制
				27.6	8- 位 BCD
				30.9	8- 位角度
				35.3	15- 位 二进制
				36.3	15- 位 BCD
				39.6	15- 位角度
				31.2	360° 二进制
				32.4	360° BCD
				35.7	360° 角度
格雷 （GRAY）码→二进制转换	GRAY_BIN	478	3	0.1	——
格雷 （GRAY）码→二进制双字转换	GRAY_BINL	479	3 ～ 4	0.1	——
二进制→格雷 （GRAY）码转换	BIN_GRAY	480	3	0.1	——
二进制→格雷 （GRAY）码双字转换	BIN_GRAYL	481	3 ～ 4	0.1	——
数值 4 位→ ASCII	STR4	601	3	8.4	——
数值 8 位→ ASCII	STR8	602	3 ～ 4	10.2	——
数值 16 位→ ASCII	STR16	603	3	15.8	——
ASCII →数值 4 位	NUM4	604	3 ～ 4	10.5	——
ASCII →数值 8 位	NUM8	605	3	14.8	——
ASCII →数值 16 位	NUM16	606	3	27.4	——

A-2-11 逻辑指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
逻辑与	ANDW	034	4	0.14	——
双字逻辑与	ANDL	610	4 ～ 6	0.26	——
逻辑或	ORW	035	4	0.18	——
双字逻辑或	ORWL	611	4 ～ 6	0.26	——
异或	XORW	036	4	0.18	——
双字异或	XORL	612	4 ～ 6	0.26	——
异或非	XNRW	037	4	0.18	——
双字异或非	XNRL	613	4 ～ 6	0.26	——
求补	COM	029	2	0.18	——
双字求补	COML	614	2	0.32	——

A-2-12 特殊算术指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
二进制平方根	ROTB	620	3	15.4	——
BCD 平方根	ROOT	072	3	7.0	——
算术处理	APR	069	4	4.6	指定 SIN 和 COS
				7.0	指定线段逼近
浮点数除法	FDIV	079	4	76.4	——
位计数器	BCNT	067	4	0.24	计数 1 个字

A-2-13 浮点数运算指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
浮点数→ 16 位	FIX	450	3 ～ 4	0.13	——
浮点数→ 32 位	FIXL	451	3 ～ 4	0.13	——
16 位→浮点数	FLT	452	3 ～ 4	0.13	——
32 位→浮点数	FLTL	453	3 ～ 4	0.13	——
浮点数加法	+F	454	4 ～ 6	0.24	——
浮点数减法	-F	455	4 ～ 6	0.24	——
浮点数除法	/F	457	4 ～ 6	0.4	——
浮点数乘法	*F	456	4 ～ 6	0.24	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
度→弧度	RAD	458	3 ～ 4	3.7	——
弧度→度	DEG	459	3 ～ 4	4.0	——
正弦	SIN	460	3 ～ 4	3.8	0° 指定
				4.5	45° 指定
				5.0	90° 指定
高速正弦	SINQ	475	8 ～ 9	0.59	0°， 45°，或 90° 指定
余弦	COS	461	3 ～ 4	3.7	0° 指定
				4.4	45° 指定
				5.3	90° 指定
高速余弦	COSQ	476	8 ～ 9	0.59	0°， 45°，或 90° 指定
正切	TAN	462	3 ～ 4	3.9	0° 指定
				6.1	45° 指定
高速正切	TANQ	477	15 ～ 16	1.2	0°， 45°，或 90° 指定
反正弦	ASIN	463	3 ～ 4	5.8	0° 指定
				24.8	45° 指定
				5.6	90° 指定
反余弦	ACOS	464	3 ～ 4	6.3	0° 指定
				25.7	45° 指定
				21.0	90° 指定
反正切	ATAN	465	3 ～ 4	4.0	0° 指定
				5.6	45° 指定
平方根	SQRT	466	3 ～ 4	0.42	——
指数	EXP	467	3 ～ 4	3.8	——
对数	LOG	468	3 ～ 4	5.8	——
指数幂	PWR	840	4 ～ 6	35.7	——
浮点符号比较	=F	329	3 ～ 5	0.13	——
	<>F	330			
	<F	331			
	<=F	332			
	>F	333			
	>=F	334			
浮点数→ ASCII	FSTR	448	4 ～ 5	53.8	——
ASCII →浮点数	FVAL	449	3	8.23	——
浮点传送 （单个）	MOVF	469	3 ～ 4	0.18	——

A-2-14 双精度浮点数指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
双精度浮点比较指令	=D	335	3	5.1	——
	<>D	336			
	<D	337			
	<=D	338			
	>D	339			
	>=D	340			
双字长浮点数→ 16 位二进制	FIXD	841	3	5.1	——
双字长浮点数→ 32 位二进制	FIXLD	842	3	5.1	——
16 位二进制 →双字长浮点数	DBL	843	3	5.1	——
32 位二进制→双字长浮点数	DBLL	844	3	5.1	——
双字长浮点数加	+D	845	4	6.0	——
双字长浮点数减	-D	846	4	6.1	——
双字长浮点数乘	*D	847	4	6.1	——
双字长浮点数除	/D	848	4	9.0	——
双字长度→弧度	RADD	849	3	6.1	——
双字长弧度→度	DEGD	850	3	6.0	——
双字长正弦	SIND	851	3	14.7	0° 指定
				20.4	45° 指定
				18.5	90° 指定
双字长余弦	COSD	852	3	14.1	0° 指定
				19.6	45° 指定
				19.1	90° 指定
双字长正切	TAND	853	3	7.3	0° 指定
				27.4	45° 指定
双字长反正弦	ASIND	854	3	7.5	0° 指定
				55.0	45° 指定
				7.3	90° 指定
双字长反余弦	ACOSD	855	3	8.3	0° 指定
				55.9	45° 指定
				43.7	90° 指定
双字长反正切	ATAND	856	3	6.1	0° 指定
				29.7	45° 指定
双字长平方根	SQRTD	857	3	16.6	——
双字长指数	EXPD	858	3	39.7	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
双字长对数	LOGD	859	3	35.5	——
双字长指数幂	PWRD	860	4	69.2	——

A-2-15 表格数据处理指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
堆栈设置	SSET	630	3	7.6	堆栈区指定 5 个字
				107.2	堆栈区指定 1,000 个字
压入栈	PUSH	632	3	4.9	——
后进后出	FIFO	633	3	6.9	堆栈区指定 5 个字
				360.0	堆栈区指定 1,000 个字
先进先出	LIFO	634	3	7.1	——
定义记录表	DIM	631	5	11.1	——
设置记录位置	SETR	635	4	3.8	——
获得记录号	GETR	636	4	4.6	——
数据搜索	SRCH	181	4	13.9	搜索 1 个字
				1940	搜索 1,000 个字 *1
交换字节	SWAP	637	3	10.1	交换 1 个字
				1421	交换 1,000 个字 *1

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
寻找最大值	MAX	182	4 ～ 5	4.8	搜索 1 个字：1
				1895	搜索 1,000 个字 *1
寻找双字长数据最大值	MAXL		4 ～ 5	4.8	搜索 1 个字：1
				1896	搜索 1,000 个字 *1
寻找浮点数据最大值	MAXF		4 ～ 5	5.24	搜索 1 个字
				2786	搜索 1,000 个字 *1
寻找双字长浮点数据最大值	MAXD		4 ～ 5	5.4	搜索 1 个字
				6324	搜索 1,000 个字 *1
寻找最小值	MIN	183	4 ～ 5	4.8	搜索 1 个字
				1896	搜索 1,000 个字 *1
寻找双字长数据最小值	MINL		4 ～ 5	4.8	搜索 1 个字
				1896	搜索 1,000 个字 *1
寻找浮点数据最小值	MINF		4 ～ 5	5.2	搜索 1 个字
				2786	搜索 1,000 个字 *1
寻找双字长浮点数据最小值	MIND		4 ～ 5	5.2	搜索 1 个字
				5789	搜索 1,000 个字 *1
求和	SUM	184	4	17.5	1 字相加
				900	1,000 字相加 *1
帧校验和	FCS	180	4	14.1	表长 1 字
				1235	表长 1,000 字 *1
读栈大小	SNUM	638	3	4.5	——
读栈数据	SREAD	639	4	6.0	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
覆盖栈数据	SWRIT	640	4	5.5	——
插入栈数据	SINS	641	4	8.2	——
				345.8	1,000 字表
删除栈数据	SDEL	642	4	8.4	——
				345.8	1,000 字表

*1 指令执行时间受数据量的很大影响，且这将影响循环时间。可通过指定后台执行以减少对循环时间的影响。详细内容，请参见 10-2-5 后台执行 。

A-2-16 跟踪指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
不带符号的单字记录搜索指令	RSRCH	360 ～ 364	6	14.0	记录数：1
				503.8	记录数：1,000
不带符号的双字记录搜索指令	RSRCH2	370 ～ 374	6	14.7	记录数：1
				841.0	记录数：1,000
不带符号的四字记录搜索指令	RSRCH4	380 ～ 384	6	17.1	记录数：1
				1544.5	记录数：1,000
不带符号的单字记录分类	RSORT	203	5	650.0	100 条记录，分散分类为无效，分类“99, 98, 97...0”到“0, 1, 2...99”（最差情况下）
不带符号的双字记录分类	RSORT2	204	5	915.0	
不带符号的四字记录分类	RSORT4	205	5	1480.0	

A-2-17 数据控制指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
PID 控制	PID	190	4	297.0	初始化执行
				234.0	输入 ON 并采样
				234.0	输入 ON 但不采样
				7.2	输入 OFF
带自整定的 PID 控制	PIDAT	191	4	302.0	初始化执行
				237.0	输入 ON 并采样
				73.2	输入 ON 但不采样
				7.3	输入 OFF
				120.0	自动的初始化执行

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
限位控制	LMT	680	4 ～ 5	10.8	——
静带控制	BAND	681	4 ～ 5	11.2	——
静域控制	ZONE	682	4 ～ 5	10.9	——
时间比例输出	TPO	685	4	6.9	输入 OFF
				37.2	输入 ON 以及占空比指定或输出限位无效
				37.4	输入 ON 以及操作变量指定或输出限位无效
标度	SCL	194	4	7.6	——
标度 2	SCL2	486	4	6.8	——
标度 3	SCL3	487	4	7.8	——
平均值	AVG	195	4	21.9	1 个操作的平均值
				212.0	64 个操作的平均值

A-2-18　子程序指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
子程序调用	SBS	091	2	0.90	——
子程序进入	SBN	092	2	——	——
子程序返回	RET	093	1	0.43	——
宏指令	MCRO	099	4	16.8	——
全局子程序返回	GSBS	750	2	0.90	——
全局子程序调用	GSBN	751	2	——	——
全局子程序进入	GRET	752	1	0.43	——

A-2-19 中断控制指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
设置中断屏蔽	MSKS	690	3	10.6	——
读中断屏蔽	MSKR	692	3	9.6	——
清除中断	CLI	691	3	10.1	——
禁止中断	DI	693	1	10.3	——
允许中断	EI	694	1	9.3	——

A-2-20 步指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
步定义	STEP	008	2	12.5	步控制位 ON
				11.8	步控制位 OFF
步启动	SNXT	009	2	4..0	——

A-2-21 基本 I/O 单元指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
I/O 刷新	IORF	097	3	10.1	基本 I/O 单元刷新 1 字（输入）
				10.5	基本 I/O 单元刷新 1 字（输出）
特殊 I/O 单元 I/O 刷新	FIORF	225	2	*1	——
CPU 总线单元 I/O 刷新	DLNK	226	4	234.0	分配 1 个字
7 段译码	SDEC	078	4	3.6	——
数字开关输入	DSW	210	6	24.8	4 位数字，数据输入值：0
				24.8	8 位数字，数据输入值：00
十键输入	TKY	211	4	7.2	数据输入值：00
				6.5	数据输入值：FF
十六进制键输入	HKY	212	5	25.9	数据输入值：00
				25.9	数据输入值：FF
矩阵输入	MTR	213	5	25.0	数据输入值：00
				25.0	数据输入值：FF
7 段显示输出	7SEG	214	5	31.4	4 位数字
				34.6	8 位数字

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
智能 I/O 读	IORD	222	4	*1	——
智能 I/O 写	IOWR	223	4	*1	——

*1 IORD， IORW 和 FIORF 指令的执行要根据其所应用的特殊 I/O 单元而定。

A-2-22 串行通信指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
协议宏	PMCR	260	5	57.8	直接指定
				77.0	操作数指定，发送 1 字，接收 1 字
协议宏 2	PMCR2	264	6	49.5	直接指定
				68.5	操作数指定，发送 1 字，接收 1 字
发送	TXD	236	4	57.5	发送 1 字节
				517.0	发送 256 字节
接收	RXD	235	4	78.5	存储 1 字节
				570.0	存储 256 字节
经由串行通信单元传送	TXDU	256	4	75.0	发送 1 字节
经由串行通信单元接收	RXDU	255	4	73.6	发送 1 字节
修改串行口设置	STUP	237	3	233.0	配址到 CPU 单元上的 COM 端口

A-2-23 网络指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
网络发送	SEND	090	4	49.5	——
网络发送 2	SEND2		5	44.0	——
网络接收	RECV	098	4	49.5	——
网络接收 2	RECV2		5	44.5	——
发布命令	CMND	490	4	58.5	——
发布命令 2	CMND2		5	53.0	——
显式报文发送	EXPLT	720	4	78.0	——
显式获取属性	EGATR	721	4	74.0	——
显式设置属性	ESATR	722	3	68.6	——

指令	助记符	代码	长度（步）	ON 执行时间（μs） CJ2H CPU6 □ -EIP	条件
显式字读取	ECHRD	723	4	64.6	——
显式字写入	ECHWR	724	4	64.2	——

A-2-24 文件存储指令

指令	助记符	代码	长度（步）	ON 执行时间（μs） CJ2H CPU6 □ -EIP	条件
读数据文件	FREAD	700	5	217.0	二进制
写数据文件	FWRIT	701	5	215.5	二进制
写文本文件	TWRIT	704	5	204.5	——

A-2-25 显示指令

指令	助记符	代码	长度（步）	ON 执行时间（μs） CJ2H CPU6 □ -EIP	条件
显示报文	MSG	046	3	6.9	显示报文
				6.6	删除显示的报文

A-2-26 时钟指令

指令	助记符	代码	长度（步）	ON 执行时间（μs） CJ2H CPU6 □ -EIP	条件
日历加法	CADD	730	4	15.6	——
日历减法	CSUB	731	4	16.4	——
小时→秒	SEC	065	3	3.6	——
秒→小时	HMS	066	3	3.6	——
时钟调整	DATE	735	2	29.3	——

A-2-27 调试指令

指令	助记符	代码	长度（步）	ON 执行时间（μs） CJ2H CPU6 □ -EIP	条件
跟踪存储器采样	TRSM	045	1	13.0	采样 1 位和 0 字
				31.6	采样 31 位和 6 字
				38.8	采样 31 位和 16 字

A-2-28 故障诊断指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
故障报警	FAL	006	3	32.0	记录错误
				78.9	按优先次序删除错误
				256.2	删除全部错误
				116.9	删除单个错误
严重故障报警	FALS	007	3	——	——
故障点检测	FPD	269	4	110.5	位地址输出，时间监控
				107.0	位地址输出，时间监控
				129.0	报文字符输出，时间监控
				158.5	报文字符输出，首条错误检测

A-2-29 其它指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
设置进位	STC	040	1	0.048	——
清除进位	CLC	041	1	0.048	——
设置 EM 块	EMBC	281	2	7.6	——
延长最大循环时间	WDT	094	2	7.6	——
存储条件标志	CCS	282	1	5.8	——
载入条件标志	CCL	283	1	6.4	——
从 CV 变换地址	FRMCV	284	3	9.4	——
转变地址到 CV	TOCV	285	3 ～ 4	10.3	——
外设服务禁止	IOSP	287	1	——	——
外设服务允许	IORS	288	1	——	——

A-2-30 块程序指令

指令	助记符	代码	长度（步）	ON 执行时间（μs）	条件
				CJ2H CPU6 □ -EIP	
块程序开始	BPRG	096	2	9.0	——

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
块程序结束	BEND	801	1	8.8	——
块程序暂停	BPPS	811	2	5.4	——
块程序重启	BPRS	812	2	3.6	——
条件块退出	（执行条件） EXIT	806	1	8.6	退出 （输入条件 ON）
				3.4	不退出 （输入条件 OFF）
条件块退出	EXIT （位地址）	806	2	9.8	退出 （位 ON）
				4.6	不退出 （位 OFF）
条件块退出 （非）	EXIT NOT （位地址）	806	2	9.9	退出 （位 OFF）
				4.6	不退出 （位 ON）
分支	IF （执行条件）	802	1	2.9	如果真 （输入条件 ON）
				4.8	如果假 （输入条件 OFF）
	IF （位地址）	802	2	1.2	如果真 （位 ON）
				6.1	如果假 （位 OFF）
分支 （非）	IF NOT （位地址）	802	2	6.1	如果真 （位 OFF）
				4.2	如果假 （位 ON）
分支	ELSE	803	1	4.5	如果真
				5.3	如果假
分支	IEND	804	1	5.3	如果真
				3.0	如果假
一个循环与等待	WAIT （执行条件）	805	1	10.0	不等待 （输入条件 ON）
				2.8	等待 （输入条件 OFF）
	WAIT （位地址）	805	2	9.2	不等待 （位 ON）
				4.1	等待 （位 OFF）
一个循环与等待 （非）	WAIT NOT （位地址）	805	2	9.2	不等待 （位 OFF）
				4.1	等待 （位 ON）
100MS 定时器等待	TIMW	813	3	15.6	默认设置
				16.0	正常执行
	TIMWX	816	3	15.1	默认设置
				16.0	正常执行
10MS 定时器等待	TMHW	815	3	15.7	默认设置
				17.5	正常执行
	TMHWX	817	3	15.2	默认设置
				16.4	正常执行

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
计数器等待	CNTW	814	4	13.7	默认设置
				13.4	正常执行
	CNTWX	818	4	13.1	默认设置
				13.5	正常执行
循环控制	LOOP	809	1	5.7	——
循环控制	LEND （执行条件）	810	1	5.3	不循环 （输入条件 ON）
				5.3	循环 （输入条件 OFF）
	LEND （位地址）	810	2	6.7	不循环 （位 ON）
				6.6	循环 （位 OFF）
循环控制 （非）	LEND NOT （位地址）	810	2	6.7	不循环 （位 OFF）
				6.6	循环 （位 ON）

A-2-31 文本字符串处理指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
串发送	MOV$	664	3	31.5	传送 1 个字符 *1
链接串	+$	656	4	55.6	1 个字符 +1 个字符 *1
取左串	LEFT$	652	4	33.5	从 2 个字符中获取 1 个字符 *1
取右串	RGHT$	653	4	33.4	从 2 个字符中获取 1 个字符 *1
取中间串	MID$	654	5	32.3	从 3 个字符中获取 1 个字符 *1
寻找串	FIND$	660	4	30.3	从 2 个字符中搜索 1 个字符 *1
计算串长	LEN$	650	3	14.0	检测 1 个字符 *1
取代串	RPLC$	661	6	110.2	用一个字符替换 2 个字符中的第一个字符 *1
删除串	DEL$	658	5	45.6	删除 2 个字符中前面的字符 *1
交换串	XCHG$	665	3	40.3	1 对 1 交换字符 *1
清除串	CLR$	666	2	15.9	清除 1 个字符 *1

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
插入串	INS$	657	5	84.9	在 2 个字符首字符之后插入 1 个字符 *1
串比较	=$	670	4	27.0	1 个字符与另一个字符比较
	<>$	671			
	<$	672			
	<=$	673			
	>$	674			
	=$	675			

*1 指令执行时间受数据量的很大影响，且这将影响循环时间。可通过指定后台执行以减少对循环时间的影响。详细内容，请参见 10-2-5 后台执行 。

A-2-32 任务控制指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
任务 ON	TKON	820	2	12.5	循环任务指定
				13.6	额外任务指定
任务 OFF	TKOF	821	2	240.0	循环任务指定
				15.5	额外任务指定

A-2-33 型号转换指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
块传送	XFERC	565	4	6.7	传送 1 字
				361.6	传送 1,000 字
单字属性	DISTC	566	4	4.6	数据属性
				6.0	堆栈操作
数据收集	COLLC	567	4	5.3	数据属性
				4.5	堆栈操作
				5.8	堆栈操作 1 字 FIFO 读取
				41.9	堆栈操作 1,000 字 FIFO 读取
移动位	MOVBC	568	4	4.9	——
位计数器	BCNTC	621	4	5.5	计数 1 字
				873.5	计数 1,000 字

A-2-34 特殊功能块指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
获取可变 ID	GETID	286	4	7.6	——

A-2-35 SFC 指令

指令	助记符	代码	长度 （步）	ON 执行时间 （μs）	条件
				CJ2H CPU6 □ -EIP	
步激活	SA	784	2	9.1	——
使步无效	SE	785	2	9.1	——
读取设置定时器	TSR	780	3	3.9	——
设置步定时器	TSW	781	3	5.7	——
SFC ON	SFCON	789	2	14.0	——
SFC OFF	SFCOFF	790	2	249.0	——
暂停 SFC （带复位）	SFCPR	793	2	249.0	——
暂停 SFC （无复位）	SFCPRN	791	2	249.0	——

A-2-36 功能块实例执行时间

当创建了功能块定义并将实例复制到用户程序时，使用以下等式来计算实例执行时对循环时间的影响。

实例执行对循环时间的影响
= 启动时间 （A）
+ I/O 参数传送处理时间 （B）
+ 功能块定义中的指令执行时间 （C）

下表显示了 A、 B 和 C 的时间长度。

操作			CPU 单元型号
			CJ2H-CPU6 □ H-EIP
A	启动时间	不包括 I/O 参数传送的启动时间	3.3 μs
B	I/O 参数传送处理时间 圆括号内为数据类型	1 位 I/O 变量 （BOOL）	0.24 μs
		1 字 I/O 变量 （INT， UINT， WORD）	0.19 μs
		2 字 I/O 变量 （DINT，UDINT，DWORD，REAL）	0.19 μs
		4 字 I/O 变量 （LINT， ULINT，LWORD， LREAL）	0.38 μs
C	功能块定义指令执行时间	总指令执行时间 （与标准用户程序相同）	

示例：CJ2H-CPU67H-EIP
带有一个 1 字数据类型 （INT）的输入变量 ： 3
带有一个 1 字数据类型 （INT）的输出变量 ： 2
功能块定义段中的总指令执行时间：10 μs
1 个实例的执行时间 = 3.3 μs + （3 + 2）× 0.19 μs + 10 μs = 14.25 μs

注　当同一功能块定义复制到多个位置时，执行时间根据多倍实例数来增长。



附加信息

功能块程序步数

当已经创建了功能块定义并将实例复制到用户程序时，使用以下等式来计算程序步数。

步数
= 实例数 × （调用部分的大小 m + I/O 参数传送部分的大小 n × 参数数） + 功能块定义 p 中的指令步数（见注）

注　当同样的功能块定义被复制到多个位置（即多个实例）时，后继实例的功能块定义（p）中的指令步数将不会减少。因此在以上等式中，实例数没有乘以功能块定义（p）中的指令步数。

内容			CJ2 CPU 单元
m	调用部分		57 步
n	I/O 参数传送处理时间 括号内为数据类型	1 位 I/O 变量（BOOL）	6 步
		1 字 I/O 变量（INT， UINT， WORD）	6 步
		2 字 I/O 变量（DINT，UDINT，DWORD，REAL）	6 步
		4 字 I/O 变量（LINT， ULINT， LWORD，LREAL）	12 步
p	功能块定义中的指令步数	指令总步数（ 与标准用户程序相同） + 27 步	

示例：
带有一个 1 字数据类型 （INT）的输入变量：5
带有一个 1 字数据类型 （INT）的输出变量：5
功能块定义段：100 步
1 个实例的步数 = 57 + （5 + 5）× 6 步 + 100 步 + 27 步 = 244 步

A-3 辅助区

A000 ～ A447：只读区； A448 ～ A1000：读 / 写区

A-3-1 只读区 （通过系统设置）

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A0	——	以 10 ms 为增量的自由运行定时器	该字包含了电源打开后使用的系统定时器。 当电源打开时 A0 设置为 0000 hex，且该值每 10 ms 自动增加 1。该值达到 FFFF hex （655,350 ms）后返回 0000 hex，然后继续每 10 ms 自动增加 1。 当运行模式切换为 RUN 模式时，定时器将继续增加。 例：可对处理 A 和处理 B 之间的间隔进行计算而无须计时器指令。这是由计算处理 A 的 A000 值和处理 B 的 A000 值之间的差异来实现的。该间隔以 10 ms 为单位计算。	保留	清除	电源打开后每 10 ms
A1	——	以 100ms 为增量的自由运行定时器	该字包含了电源打开后使用的系统定时器。 当电源打开时 A1 设置为 0000 hex，且该值每 100 ms 自动增加 1。该值达到 FFFF hex （6,553,500ms）后返回 0000 hex，然后继续每 100 ms 自动增加 1。 当运行模式切换为 RUN 模式时，定时器将继续增加。	保留	清除	电源打开后每 100 ms
A2	——	以 1s 为增量的自由运行定时器	该字包含了电源打开后使用的系统定时器。 当电源打开时 A2 设置为 0000 hex，且该值每 1s 自动增加 1。该值达到 FFFF hex （65,535 s）后返回 0000 hex，然后继续每 1s 自动增加 1。 当运行模式切换为 RUN 模式时，定时器将继续增加。	保留	清除	电源打开后每 1s
A50	A50.00 ～ A50.07	基本 I/O 单元信息， 0 号机架槽位 0	一个位变为 ON 表示给出负载短路保护功能的报警输出。 CJ1W-OD202 （每位 2 点）为最多使用 4 个 LSB， CJ1W-OD212，OD204，MD232 仅使用 1 个 LSB， CJ1W-OD232 为最多使用 2 个 LSB。 ON：短路 OFF：正常	——	——	每次循环刷新
	A50.08 ～ A50.15	基本 I/O 单元信息， 0 号机架槽位 1		——	——	
A51 ～ A69	A51.00 ～ A69.15	基本 I/O 单元的信息， 0 号机架槽位 2 到 3 号机架槽位 9		——	——	
A90 ～ A93	——	用户程序日期	保存用户程序的最后修改日期和时间 （以 BCD 码表示）。 A90.00 ～ A90.07：秒 （00 ～ 59） A90.08 ～ A90.15：分 （00 ～ 59） A91.00 ～ A91.07：时 （00 ～ 23） A91.08 ～ A91.15：日 （01 ～ 31） A92.00 ～ A92.07：月 （01 ～ 12） A92.08 ～ A92.15：年 （00 ～ 99） A93.08 ～ A93.07：星期 （00：星期天，01：星期一，02：星期二，03：星期三，04：星期四，05：星期五，06：星期六）	保留	保留	——
A94 ～ A97	——	参数日期	保存参数的最后修改日期和时间 （以 BCD 码表示）。 格式同上。	保留	保留	——

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A99	00	UM 读保护状态	表示 PLC 中的整个用户程序是否被读保护。 OFF：UM 无读保护 ON：UM 读保护	保留	保留	当设置或清除了保护时（单元版本 2 及以上）
	01	任务读保护状态	表示是否对单独任务设置了读保护。 OFF：任务无读保护 ON：任务读保护			——
	02	当设置了读保护时程序写保护的状态	表示程序是否被写保护。 OFF：允许写 ON：写保护			
	03	允许 / 禁止将程序备份至存储卡的状态	表示了是否创建备份程序文件（.OBJ）。 OFF：允许 ON：禁止			
	14	任务间 IR/DR 操作	打开此位以在所有任务间共享变址和数据寄存器。 关闭此位以在每个任务中使用单独的变址和数据寄存器。 OFF：单独 ON：共享（默认）			——
A100 ～ A199	——	错误记录区	发生错误时，错误代码、错误内容和错误的日期 / 时间存在错误记录区中。可存储 20 个最近的错误。 每个错误记录占 5 个字：这些字的功能如下： 第一字：错误代码（位 0 ～ 15） 第二字：错误内容（位 0 ～ 15） 错误内容： 辅助区字地址（带详细或 0000） 第三字：分（位 8 ～ 15），秒（位 0 ～ 7） 秒：00 ～ 59， BCD 分：00 ～ 59， BCD 第四字：日（位 8 ～ 15），时（位 0 ～ 7） 时：00 ～ 23， BCD 日：01 ～ 31， BCD 第五字：年（位 8 ～ 15）， 月（位 0 ～ 7） 年：00 ～ 99，BCD 月：00 ～ 12，BCD 由 FAL（006）和 FALS（007）产生的错误也存储在这个错误记录区里。 可通过 CX-Programmer 清除错误记录区。 如果错误记录区记满（20 条记录）并且此时又出现一个错误，则清除在 A100 ～ A104 中最早的记录。其他 19 条向下移动，新记录记在 A195 ～ A199。	保留	保留	发生错误时刷新 A50014 A300 A400

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A200	A200.11	第一周期标志	PLC 运行开始后 （从编程模式切换到运行或监视模式后）的一个周期，此标志为 ON 。 第一次周期为 ON 。	——	——	——
	A200.12	步标志	当步执行从 STEP （008）开始时一个周期为 ON，这个标志用于第一步开始时的初始化处理。 执行 STEP （008）后的第一个周期为 ON 。	清除	——	——
	A200.14	任务启动标志	当一个任务从等待或禁止状态切换到运行状态（STANDBY/DISABLED → READY），在这个任务的第一个周期内此标志置 ON 。 1：第一个周期为 ON （包括从等待或禁止状态切换过来） 0：其他 这个标志和 A200.15 唯一的不同是当任务从等待状态切换到运行状态 （STANDBY → READY）时这个标志也变为 ON。	清除	清除	——
	A200.15	第一次启动任务标志	当第一次执行任务时置 ON。用这个标志来检查当前任务是否是第一次执行，是否有必要进行初始化处理。 ON：第一次执行 OFF：非第一次执行或未执行	清除	——	——
A201	A201.10	在线编辑等待标志	正在等待在线编辑处理时 ON。 ON：等待在线编辑 OFF：不等待在线编辑 （如果等待中收到另一个在线编辑命令，则不会记录其他命令，而且会发生错误）	清除	清除	A527
	A201.11	在线编辑等待标志	执行在线编辑过程时 ON。 ON：在线编辑在进程中 OFF：在线编辑不在进程中	清除	清除	A527
A202	A202.00 ～ A202.07	通信端口允许标志	可以相对应的端口编号执行网络指令 （SEND、 RECV、 CMND、 PMCR、TXDU、RXDU）或后台操作时，标志为 ON。位 00 ～ 07 对应通信端口 0 ～ 7。 ON：没有执行网络指令 OFF：网络指令执行中 （端口忙） 在程序中两个或两个以上网络指令编程使用相同端口号的端口时，使用相应的标志作为一种运行条件，以防止指令被同时执行。 （当以一个给出编号端口执行网络指令时，该端口的标志将关闭）。 当执行一指令时清除。	清除	——	——
	A202.08	CJ2 指令允许标志	可使用 CJ2 指令时 ON。 默认状态下此标志为 ON。 此标志仅可用于下述指令：SEND2、CMND2、PMCR2、RECV2 。	——	根据内部状态更新（清除）	——
	A202.15	网络通信端口分配允许标志	执行通信指令 （SEND、 RECV、CMND、 PMCR、 TXDU、 RXDU）时，当有一通信端口可用于自动分配的情况下 ON。 ON：通信端口有效 OFF：通信端口无效 注：当同时使用 9 条及以上的通信指令时，使用此标志，在执行通信指令前确认是否有一通信端口可用于自动分配。	清除	——	——

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A203 ～ A210	——	通信端口完成代码	当执行网络指令（SEND、RECV、CMND 或 PMCR）或后台执行时，这些字中保存相应端口编号的完成代码。 （当后台执行完成时，相应的字将被清除为 0000 hex）。 字 A203 ～ A210 对应通信端口 0 ～ 7。 非零：错误代码 0000：正常条件 执行显式报文指令 （EXPLT、 EGATR、ESATR、ECHRD 或 ECHWR）时下列代码将被保存。 若显式通信错误标志为 OFF，则将保存 0000hex 。 若显式通信错误标志为 ON 且网络通信错误标志为 ON，则将保存 FINS 终止代码。 若显式通信错误标志为 ON 且网络通信错误标志为 OFF，则将保存显式报文终止代码。 通信期间，0000 hex 将被保存且当执行完成时，匹配代码也将被保存。运行开始时代码将被清除。 （当以所给出的端口执行了网络指令时，该端口的完成代码将被清为 0000）。 当执行一指令时清除。	清除	——	——
A211	——	CJ2 网络通信指令的有效端口数	当可使用的 CJ2 网络通信指令数到达为 0 时， A202.08 将为 OFF。此字仅可用于下述指令：SEND2、 CMND2、PMCR2、 RECV2。 此字中的内容可用于检验通信流量。	——	根据内部状态更新（清除）	——
A213	A213.00 ～ A213.07	显式通信错误标志	显式报文指令（EXPLT、EGATR、ESATR、ECHRD 或 ECHWR）执行中发生错误时 ON 。 位 00 ～ 07 对应通信端口 0 ～ 7。 ON：错误中止 OFF：正常中止 当显式报文无法发送或对显式报文返回了错误的响应时，相应位将为 ON。 状态将被保持到下次执行显式报文通信时。当执行下一条显式报文指令时该位将始终为 OFF。	清除	——	A219.00 ～ A219.07 A203 ～ A210
A214	A214.00 ～ A214.07	网络通信完成后的第一个周期标志	通信完成后每个标志都将只置 ON 一个周期。位 00 ～ 07 对应端口 0 ～ 7。使用保存在 A218 中的已使用的通信端口编号来决定访问哪个标志。 ON：仅通信完成后的第一个周期 OFF：其它状态 这些标志直到执行通信指令后的下一周期才起效。至少延迟一个周期后再对其进行访问。 使用 A218 中指定的端口编号 （已使用的通信端口编号）对所使用端口的正确位进行访问。	保留	清除	——
A215	A215.00 ～ A215.07	网络通信错误后的第一个周期标志	通信错误发生后每个标志都将只置 ON 一个周期。位 00 ～ 07 对应端口 0 ～ 7。使用保存在 A218 中的已使用的通信端口编号来决定访问哪个标志。 根据保存在 A203 ～ A210 中的通信端口完成代码来决定错误的原因。 ON：仅通信错误后的第一个周期 OFF：其他状态 这些标志直到执行通信指令后的下一周期才起效。至少延迟一个周期后再对其进行访问。 使用 A218 中指定的端口编号 （已使用的通信端口编号）对所使用端口的正确位进行访问。	保留	清除	——

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A216 和 A217	——	网络通信的完成代码保存地址	通信指令的完成代码自动保存在这些字中指定的 I/O 存储器地址。将此地址置入变址寄存器中并通过变址寄存器使用间接寻址来读取通信完成代码。	保留	清除	——
A218	——	已使用的通信端口编号	当使用自动通信端口分配来执行通信指令时，保存通信端口编号。 0000 ～ 0007hex：通信端口 0 ～ 7	保留	清除	——
A219	A219.00 ～ A219.07	通信端口错误标志	网络指令（SEND、 RECV、CMND 或 PMCR）执行期间发生错误时 ON。 位 00 ～ 07 对应通信端口 0 ～ 7。 ON：错误 OFF：正常	保留	——	——
A220 ～ A259	A220.00 ～ 259.15	基本 I/O 单元输入响应时间	这些字中包含实际输入响应时间。 0 ～ 17hex 在 PROGRAM 模式下的 PLC 设置中更改了基本 I/O 单元输入响应时间设定时，PLC 设置中的设定将与基本 I/O 单元中的实际值不匹配，除非电源关闭并重新打开。发生此类情况时，可通过这些字对实际值进行监控。	保留	见功能栏	PLC 设置（基本 I/O 单元输入响应时间设定）
A260	——	I/O 分配状态	指示 I/O 目前配置状况。例如，在启动时自动 I/O 分配或用户设定 I/O 分配。 0000 hex：自动 I/O 分配 BBBB hex：用户设置的 I/O 分配	保留	保留	——
A261	A261.00	I/O 表创建错误详细	CPU 总线单元中配置错误时 ON 正常生成 I/O 表时转为 OFF ON：CPU 总线单元中配置错误 OFF：正常生成 I/O 表	保留	清除	生成 I/O 表时
	A261.02		超出最大的 I/O 点数时 ON 正常生成 I/O 表时转为 OFF ON：超出最大的 I/O 点数 OFF：正常生成 I/O 表			A401.11（I/O 点数过多）
	A261.03		重复使用同一个单元编号时 ON 正常生成 I/O 表时转为 OFF ON：重复使用同一个单元编号 OFF：正常生成 I/O 表			A401.13（编号重复）
	A261.04		I/O 总线错误时 ON 正常生成 I/O 表时转为 OFF ON：I/O 总线错误 OFF：正常生成 I/O 表			A401.14（I/O 总线错误）
	A261.06		因无法检测到 SYSMAC BUS 从站而导致 I/O 表错误时 ON。 正常生成 I/O 表时转为 OFF。 ON：SYSMAC BUS 从站缺失 OFF：正常生成 I/O 表			——
	A261.07		特殊 I/O 单元错误时 ON 正常生成 I/O 表时转为 OFF ON：特殊 I/O 单元错误 OFF：正常生成 I/O 表			——
	A261.09		没有完成 I/O 检测时 ON， 正常生成 I/O 表时转为 OFF ON：没有完成 I/O 检测 OFF：正常生成 I/O 表			——
A262 和 A263	——	最大循环时间（以 0.1ms 增量为单位）	保存从 PC 启动操作开始的最大循环时间。循环时间为以 8 位数字 16 进制记录，最左边 4 个数字记录在 A263 中，最右边 4 个数字记录在 A262 中。 0 ～ FFFFFFFF：0 ～ 429,496,729.5 ms （以 0.1ms 增量为单位）	清除	清除	——

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A264 和 A265	——	当前循环时间（以 0.1ms 增量为单位）	保存当前循环时间（以 0.1ms 增量为单位）。各循环中以 8 位数字 16 进制记录，最左边 4 个数字记录在 A264 中，最右边 4 个数字记录在 A265 中。 00000000 ～ FFFFFFFF （0.0 ～ 429,496,729.5 ms）	清除	清除	——
A266 和 A267	——	当前循环时间（以 0.01ms 增量为单位）	保存当前循环时间（以 0.01ms 增量为单位）。各循环中以 8 位数字 16 进制记录，最左边 4 个数字记录在 A266 中，最右边 4 个数字记录在 A267 中。 00000000 ～ FFFFFFFF （0.0 ～ 42,949,672.95 ms）	清除	清除	——
A294	——	程序停止时的任务编号	该字包含了程序执行因程序错误而停止时，正在执行的任务的任务编号。 普通任务：0000 ～ 007F hex （任务 0 ～ 127） 中断任务：8000 ～ 80FF hex （任务 0 ～ 255） （A298 和 A299 包含程序执行停止时的程序地址）。	清除	清除	A298/A299

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A295	A295.08	指令处理错误标志	当发生指令处理错误且 PLC 设置已设为在指令错误发生时停止运行，则该标志和错误标志 （ER）将打开。此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 ON：错误标志打开 OFF：错误标志关闭	清除	清除	PLC 设置 （发生指令错误时的操作） 错误发生处的任务编号将被保存在 A294 中，程序地址将被保存在 A298 和 A299 中。
	A295.09	间接 DM/EM BCD 错误标志	当发生间接 DM/EM BCD 错误且 PLC 设置已设为在间接 DM/EM 错误发生时停止运行，则该标志和访问错误标志 （AER）将打开。（当选择了 BCD 模式但间接寻址的 DM 或 EM 字的内容不为 BCD 时发生此错误）。此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 ON：非 BCD OFF：正常			
	A295.10	非法访问错误标志	当非法访问的错误发生，并且 PC 设置已设为有非法访问时停止运行，则此时本标志和访问错误标志 （AER）都会转成 ON。（当存储器区发生非法访问时会出现这个错误）。此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 下列操作是非法访问： 读 / 写系统区 读 / 写 EM 文件存储器 写入写保护区域内 间接 DM/EM BCD 错误 （在 BCD 方式下） ON：非法访问 OFF：正常			
	A295.11	无 END 错误标志	在一个任务中的每个程序中缺 END （001）指令时为 ON。 此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 ON：无 END OFF：正常			错误发生处的任务编号将被保存在 A294 中，程序地址将被保存在 A298 和 A299 中。
	A295.12	任务错误标志	一个任务发生错误时 ON。以下情况会产生一个任务错误： 没有一个可执行的常规任务 （启动）。 无程序分配到任务。 ON：错误 OFF：正常			
	A295.13	微分溢出错误标志	相应微分指令超出微分标志的允许取值范围时 ON。 此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 ON：错误 OFF：正常			
	A295.14	非法指令错误标志	存储了无法执行的程序时 ON。 此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 ON：错误 OFF：正常			
	A295.15	UM 溢出错误误标志	超出了 UM （用户存储区）的最后一个地址时 ON。 此标志为 ON 时， CPU 单元运行将停止且 ERR/ALM 指示灯将亮。 ON：错误 OFF：正常			
A298 和 A299	——	程序停止处的地址	包含由于程序错误而导致程序停止执行时，指令的 8 位数字二进制程序地址。	清除	清除	（A294 中保存了程序停止执行时的任务号）

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A300	——	错误日志指示器	当错误发生时，错误日志指示器会以 1 为增量指向下一个错误日志要记录的位置，这是从错误日志区 （A100 ～ A199）开始的一个偏移量。 00 ～ 14 十六进制 通过将 A500.14 （错误复位位）从 OFF 转为 ON，可将错误日志指示器清除为 0。 当指示器指针是 14 hex （20 decimal）时，下一个错误日志记录位置是 A195 ～ A199。	保留	保留	错误发生时刷新 A500.14
A301	——	当前 EM 组——	4 位数字 16 进制表示的 EM 区段号，用指令 EMBC （281）可以更换当前 EM 区段号。 0000 ～ 0016 十六进制	清除	清除	——
A302	A302.00 ～ A302.15	CPU 总线单元初始化标志	CPU 总线单元重启位 （A501.00 ～ A501.15）从 OFF 到 ON 或接通电源后，且相应的 CPU 总线单元是在初始化时，这些标志为 ON。 位 00 ～位 15 对应单元号 0 ～ 15。 在程序中使用这些标志，以防止在初始化阶段使用 CPU 总线单元正在刷新的数据。在 CPU 总线单元的初始化时，不能执行 IORF （097）和 FIORF （225）指令 （仅 CJ2 和 CJ1H-R CPU 单元）。 OFF：不在初始阶段 ON：在初始阶段 （初始化后自动恢复为 0） 完成初始化完成后，这些位自动转回 OFF。	保留	清除	在初始化时写入 A501.00
A315	A315.12	标签存储器错误标志 （非致命错误）	保存网络符号 （变量）的标签存储器中发生错误时 ON。	保留	清除	——
A315	A315.15	备份存储器错误标志	写入内部闪存的备份数据区、源存储器区或注释存储器区错误时 ON。 顺利完成写入时，此位转为 OFF。	保留	清除	——
A330 ～ A335	A330.00 ～ A335.15	特殊 I/O 单元初始化标志	特殊 I/O 单元重启位 （A502.00 ～ A507.15）从 OFF 到 ON 或接通电源后，且相应的特殊 I/O 单元处在初始化阶段时，这些标志为 ON。 这些字中位所对应的的单元号 0 ～ 95 如下所示： A330.00 ～ A330.15：单元 0 ～ 15 A331.00 ～ A331.15：单元 16 ～ 31 —— A335.00 ～ A335.15：单元 80 ～ 95 在程序中使用这些标志，以防止在初始化阶段使用特殊 I/O 单元正在刷新的数据。在 CPU 总线单元的初始化时，不能执行 IORF （097）和 FIORF （225）指令 。 OFF：不在初始阶段 ON：在初始阶段 （初始化后自动恢复为 0） 完成初始化完成后，这些位自动转回 OFF。	保留	清除	A502.00 ～ A507.15
A336	A336.00 ～ A336.15	启动时检测的单元 （ 机架 0 ～ 3）	每个机架检测的单元数保存在 1 位 16 进制数字中 （0 ～ A hex）。 机架 0：A336.00 ～ A336.03 机架 1：A336.04 ～ A336.07 机架 2：A336.08 ～ A336.11 机架 3：A336.12 ～ A336.15 比如：机架 0 有 1 个单元，机架 1 有 4 个单元，机架 2 有 8 个单元，机架 3 有 10 个单元，如下所示： A336 = A 8 4 1	保留	清除	——
A339 和 A340	——	最大微分标志数	这些字中包含由微分指令使用的微分标志的最大数目。	见功能栏	清除	开始操作时写入 A295.13

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A343	A343.00 ～ A343.02	存储卡类型	如果安装了存储卡，表示安装的存储卡类别。 0：无 4：闪存 ROM 当接通 PLC 电源或接通存储卡电源开关时，存入类别信息。	保留	见功能栏	见功能栏
	A343.06	EM 文件存储器格式错误标志	在分配给文件内存第一个 EM 组中出现格式错误时 ON。 ON：格式错误 OFF：没有格式错误 （正常完成格式时自动转为 OFF）	保留	清除	——
	A343.07	存储卡格式错误标志	存储卡没有格式化或格式化错误时转 ON。（正常完成格式时，此标志转为 OFF）。 ON：格式错误 OFF：没有格式错误 接通 PLC 电源或接通存储卡电源开关时写入此标志。	保留	见功能栏	见功能栏
	A343.08	文件传送错误标志	在向文件存储区写入数据中发生错误时 ON。 ON：错误 OFF：没有错误	保留	清除	文件数据写入时刷新。
	A343.09	文件写错误标志	当因为写保护或数据超过文件存储器容量而使数据无法写入文件存储器时 ON。 ON：不可写入 OFF：正常	保留	清除	文件数据写入时刷新。
	A343.10	文件读错误标志	当因为故障 （文件损坏或数据被破坏）而使文件无法读取时 ON。 ON：不可读取 OFF：正常	保留	清除	文件数据读取时刷新。
	A343.11	文件缺失标志	当尝试读取不存在的文件或尝试将文件写入不存在的目录时 ON。 ON：指定文件或目录缺失 OFF：正常	保留	清除	文件数据读取时刷新。
	A343.13	文件存储器操作标志	在执行下列操作时 ON，不执行操作时为 OFF。 CMND 指令向本地 CPU 单元发送 FINS 命令。 一文件存储器指令的执行。 用辅助区的控制位更换程序。 简捷备份操作 ON：正在执行指令 OFF：没有执行指令	保留	清除	执行文件存储器指令时刷新
	A343.14	文件数据访问标志	正在访问文件数据时转 ON。用这个标志防止同时执行两个文件存储器指令。 ON：正在访问文件 OFF：未在访问文件	保留	清除	——
	A343.15	存储卡检测标志	检测到存储卡时转 ON。 没有检测到存储卡时转 OFF。 ON：检测到存储卡 OFF：未检测到存储卡	保留	清除	插入存储卡时或电源开启时刷新

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A344	A344.00 ～ A344.07	EM 文件存储器的起始组	保存了 EM 跟踪存储器或 EM 文件存储器中起始段的组编号。 在 EM 组中从起始组至最后一个组都被格式化用作文件存储器或跟踪存储器。若 EM 区组未被格式化为文件存储器或跟踪存储器，则 A344 将为 FFFF hex。 要将部分 EM 区转换为文件存储器或跟踪存储器，从 CX-Programmer 选择 PLC- 存储器分配 - EM 存储器设定，并选择文件存储器或跟踪存储器	保留	保留	PLC 设置（EM 文件设定有效参数和 EM 起始文件编号参数）
	A344.14	EM 跟踪存储器标志	A344 不为 FFFF hex 时，此标志 ON， EM 区的组由 A344.00 ～ A344.07 中指定的组开始到 EM 区的最后一个组都被格式化为跟踪存储器。	保留	保留	
	A344.15	EM 文件存储器标志	A344 不为 FFFF hex 时，此标志 ON， EM 区的组由 A344.00 ～ A344.07 中指定的组开始到 EM 区的最后一个组都被格式化为文件存储器。	保留	保留	
A346 和 A347	——	待传输的字数	这些字中保存 8 位 16 进制的数字，为准备通过 FREAD（700）或 FWRIT（701）指令传输的字的数目。当这些指令执行时，待传送的字数被写到 A346 和 A347 中。 当数据被传送时，这些字的值会递减。 A326 为保存最右边 4 个数字，A347 保存最左边 4 个数字。 核对这些字的内容，检查计划传送的字数是否已经成功传送。	保留	清除	在执行 FREAD 或 FWRIT 时写入。 数据成功传出时递减。
A351 ～ A354	——	日期 / 时钟区	这些字以 BCD 方式保存了 CPU 单元的内部时钟数据。这个时钟可通过由 CX-Programmer 使用 DATE（735）指令或 FINS 命令（CLOCK WRITE, 0702）进行设定。 A351.00 ～ A351.07：秒 （00 ～ 59）（BCD） A351.08 ～ A351.15：分 （00 ～ 59）（BCD） A352.00 ～ A352.07：时 （00 ～ 23）（BCD） A352.08 ～ A352.15：日 （01 ～ 31）（BCD） A353.00 ～ A353.07：月 （01 ～ 12）（BCD） A353.08 ～ A353.15：年 （00 ～ 99）（BCD） A354.00 ～ A354.07：星期 （00 ～ 06）（BCD） 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	保留	保留	每个循环写入
A360 ～ A391	A360.01 ～ A391.15	已执行的 FAL 数目标志	在执行 FAL（006）时，和 FAL 号对应的标志将变为 ON。位 A360.01 ～ A391.15 对应 FAL 号 001 ～ 511。 错误清除时标志变为 OFF。 ON：执行过 FAL OFF：没有执行 FAL	保留	清除	错误时刷新 A402.15

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A392	A392.04	串行端口错误标志	RS-232C 端口发生错误时为 ON。（外设总线模式或 1：N NT 链接模式无效）。 ON：错误 OFF：正常	保留	清除	错误发生时刷新
	A392.05	串行端口发送准备标志（无协议模式）	在无协议方式下 RS-232C 端口能够发送数据时为 ON。 ON：能够发送 OFF：不能发送	保留	清除	传送后写入
	A392.06	串行端口接收完成标志（无协议模式）	在无协议方式下 RS-232C 端口完成接收时为 ON。 • 指定了字节数目：在收到指定的字节数时为 ON。 • 指定了结束码：当收到结束码或收了 256 个字节时为 ON。	保留	清除	接收后写入
	A392.07	串行端口接收溢出标志 （无协议模式）	在无协议方式下通过 RS-232C 端口接收数据期间，发生数据溢出时 ON。 • 指定字节数时：在完成接收后尚未执行 RXD（235）指令前，接收了更多的数据时为 ON。 • 指定结束码时：收到结束码后尚未执行 RXD（235）指令前，接收了更多的数据时为 ON。 在结束码前收到 257 个字节时为 ON。 ON：溢出 OFF：无溢出	保留	清除	
A393	A393.00 ～ A393.07	串行端口 PT 通信标志	与 NT 链接模式下 RS-232C 端口与 PT 通信时，相应位将打开。 位 0 ～ 7 对应单元 0 ～ 7。 ON：正在通信 OFF：没有通信	保留	清除	在对话有正常响应时刷新
	A393.08 ～ A393.15	串行端口 PT 优先权登记标志	与 NT 链接模式下进行 RS-232C 端口通信时，具有优先权的 PT 用相应位将打开。 位 0 ～ 7 对应单元 0 ～ 7。 当收到优先权登记命令时写入这些标志。 ON：已登记优先权 OFF：未登记优先权	保留	清除	见功能栏
	A393.00 ～ A393.15	串行端口接收计数器（无协议模式）	表示 RS-232C 端口在无协议方式下接收数据的字节数（二进制）。	保留	清除	数据接收时刷新

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A395	A395.06	文件删除标志	系统删除断电时正在更新的存储卡内文件的遗留。 ON：删除文件 OFF：没有文件删除	清除	清除	当系统删除文件时刷新
	A395.07		系统删除断电时正在更新的 EM 文件存储器内文件的遗留。 ON：删除文件 OFF：没有文件删除	清除	清除	当系统删除文件时刷新
	A395.10	后台执行的 ER/AER 标志	后台处理期间发生指令处理错误或非法区域访问错误时 ON 。 ON：错误。电源打开时关闭 （0）。运行开始时关闭 （0） OFF：无错误。后台处理开始时关闭 （0）	清除	清除	——
	A395.11	存储器损坏检测标志	打开电源时检查到有损坏的存储区时为 ON 。 ON：存储器损坏 OFF：正常运行	保留	见功能栏	电源接通时刷新
	A395.12	DIP 开关位 6 状态标志	在 CPU 单元正面的 DIP 开关上位 6 的状态。每个循环写一次此标志。 ON：位 6 ON OFF：位 6 OFF	保留	见功能栏	每循环写入
A397	——	简捷备份写入容量	如果简捷备份写操作失败，A397 将包含请求完成写操作的存储卡容量。这个值是以千字节为单位 （且表示为写操作启动时存储卡没有指定容量）。 当完成简捷备份写操作时 A397 清除为 0000 hex。	保留	保留	执行写操作时刷新
A400	——	错误代码	当发生非致命错误 （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将被写入该字中。（参见 132 页的 *A-3-3 辅助区操作细节*） 同时发生两个或更多错误时，将记录最高错误代码。关于错误代码的详情请参阅第 631 页。	清除	清除	错误时刷新

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A401	A401.03	卡传送错误标志	电源打开时存储卡内容未正确读取到 CPU 单元时 ON。	错误清除操作后清除	清除	——
	A401.05	版本错误标志	CPU 单元版本不支持传送的用户程序时 ON。	错误清除操作后清除	清除	——
	A401.06	FALS 错误标志（致命错误）	由于 FALS （006）产生非致命错误时转 ON。 CPU 单元继续运作，且 ERR/ALM 指示灯会闪烁。 相应的错误代码写入到 A400 中。错误代码 C101 ～ C2FF 对应 FALS 001 ～ 511。 ON：执行 FALS （006） OFF：不执行 FALS （006） 当错误 FALS 错误 清除时此标志转 OFF。	清除	清除	错误时刷新 A400
	A401.08	循环时间太长标志 （致命错误）	循环时间超过 PLC 设置中设定的最大循环时间时为 ON （监视循环时间）。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 OFF：低于最大循环 ON：超出最大循环 错误清除时此标志转为 OFF。	清除	清除	超出最大循环时刷新。 PLC 设置 （监视循环时间）
	A401.09	程序错误标志（致命错误）	程序内容不正确时为 ON。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。发生错误的任务保存在 A294，程序地址保存在 A298 和 A299 。 所出现的程序错误的类别保存在 A295 的位 8 ～位 15。关于程序错误的细节内容，请参见 A295 的说明。 ON：错误 OFF：正常 错误清除时此标志转为 OFF。	清除	清除	A294， A295， A298 和 A299
	A401.10	I/O 设定错误标志（致命错误）	当注册 I/O 表与实际 I/O 表不相符 （如， PLC 中实际安装的单元）时，或对于 CJ2H-CPU □□ -EIP 单元，中断输入单元连接在错误位置 （非槽位 0 ～ 3 时，如 CPU 单元右侧的槽位 4）时为 ON。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 ON：错误 OFF：正常 错误清除时此标志转为 OFF。	清除	清除	A405.08
	A401.11	I/O 点数过多标志（致命错误）	基本 I/O 单元所使用的 I/O 点数超过此 PLC 所允许的最大范围时，或一个机架上连接的单元数超出 11 个时为 ON。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 ON：错误 OFF：正常 错误清除时此标志转为 OFF。	清除	清除	A407

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A401	A401.13	重复错误标志 （致命错误）	以下情况为 ON： • 两个 CPU 总线单元分配到相同的单元号。 • 两个特殊 I/O 单元分配到相同的单元号。 • 两个基本 I/O 单元分配到相同的数据区字。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 在 A409 ～ A416 内指示重复的单元号。 ON：重复错误 OFF：无重复 （错误清除时此标志转为 OFF）。	清除	清除	A410 ～ A416
	A401.14	I/O 总线错误标志 （致命错误）	CPU 单元与安装在槽上的一个单元间的数据传送错误时，或未连接端盖与 CPU 机架 / 扩展机架时为 ON。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 发生 I/O 总线错误的槽位号 （00 ～ 09）以二进制写入 A404.00 ～ A404.07，机架号 （00 ～ 07）以二进制写入到 A404.08 ～ A404.15。如果端盖没有和 CPU 的机架或扩展机架连接好，0E（16 进制）写到上述二个位置中。若在内置网络上发生 I/O 总线错误，则这些位中将保存 0B hex。 ON：错误 OFF：正常 （错误清除时此标志转为 OFF）。	清除	清除	A404
	A401.15	存储器错误标志 （致命错误）	在存储器中发生错误或在电源接通时发生从存储卡自动传送错误时为 ON 。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 ON：错误 OFF：正常 错误位置在 A403.00 ～ A403.08 中指示，如果是启动时自动传送错误 A40309 转 ON。 错误清除时此标志转为 OFF。 （如果 PLC 不断电，启动时自动传送错误不会清除）。	清除	清除	A403
A402	A402.00	其他非致命错误标志 （系统工作存储器错误）	• 发生除分配到 A402.01 ～ A402.15 以外的非致命错误时为 ON。（如，在线编辑的存储器中发生错误） • 其他非致命错误的详细内容保存在 A315 中。	错误清除操作后清除	清除	——
	A402.02	特殊 I/O 单元设定错误标志 （非致命错误）	安装的特殊 I/O 单元与在 I/O 表上登记的特殊 I/O 单元不一致时 ON。CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。 ON：错误 OFF：正常 在 A428 ～ A433 指示出现配置错误的单元号。 （错误清除时此标志转为 OFF）。	清除	清除	A428 ～ A433
	A402.03	CPU 总线单元设定错误标志 （非致命错误）	安装的 CPU 总线单元与在 I/O 表上登记的 CPU 总线单元不一致时 ON。CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。 ON：错误 OFF：正常 出现配置错误的单元号写入 A427 中。 （错误清除时此标志转为 OFF）。	清除	清除	A427

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A402	A402.04	电池错误标志 （非致命错误）	如果 CPU 单元的电池未连接或电池电压过低，而且在 PLC 设置中设定了检测电池错误时 ON。 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。 ON：错误 OFF：正常 此标志可以控制外部的报警灯或其他指示灯，指示电池需要更换。 （错误清除时此标志转为 OFF）。	清除	清除	PLC 配置 （检查电池错误）
	A402.06	特殊 I/O 单元错误标志 （非致命错误）	特殊 I/O 单元和 CPU 单元间交换数据错误时为 ON。（包括特殊 I/O 单元自身的错误）。 ON：错误 OFF：正常 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。错误的特殊 I/O 单元会停止运作。在 A418 ～ A423 中指示发生数据交换错误的单元号。 （错误清除时此标志转为 OFF）。	清除	清除	A418 ～ A423
	A402.07	CPU 总线单元错误标志 （非致命错误）	CPU 总线单元和 CPU 单元间交换数据错误时为 ON。（包括 CPU 总线单元自身的错误）。 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。错误的 CPU 总线单元会停止运作。在 A417 中指示发生数据交换错误的 CPU 总线单元的单元号。 ON：错误 OFF：正常 （错误清除时此标志转为 OFF）。	清除	清除	A417
	A402.10	PLC 设置错误标志 （非致命错误）	在 PLC 设置中有设定错误时 ON。 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。错误的位置写入 A406 中。 ON：错误 OFF：正常 （错误清除时此标志转为 OFF）。	清除	清除	A406
	A402.12	基本 I/O 单元错误标志 （非致命错误）	基本 I/O 单元中发生错误时 ON。 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。错误位置写入 A408 中。 ON：错误 OFF：正常 （错误清除时此标志转为 OFF）。	清除	清除	A408
	A402.13	重复刷新错误标志 （非致命错误）	PLC 设置中的检测重复刷新错误参数设置为检测重复刷新错误，并且在同一特殊 I/O 单元中发生如下所述的某一情况时，此标志为 ON。 • 一循环任务中的 FIORF （225）， IORF （097）， IORD （222）或 IOWR （223）与一中断任务中的 FIORF （225），IORF （097），IORD （222）或 IOWR （223）相抵触。 • I/O 刷新中时在一中断任务中执行 FIORF （225），IORF （097），IORD （222）或 IOWR （223）。 如果一特殊 I/O 单元的 PLC 设置中循环刷新为有效，并且在一中断任务中对同一特殊 I/O 单元执行 FIORF（225），IORF（097），IORD （222）或 IOWR （223），则将发生重复刷新错误。	清除	清除	A426， PLC 设置 （PLC 设置中的检测重复刷新错误参数设置为 “检测”）

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A402	A402.15	FAL 错误标志 （ 非致命错误）	通过执行 FAL （006）产生非致命错误时 ON。 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。 A360 ～ A391 中与 FALS （006）中指定 FAL 编号对应的位将会转为 ON，而且相应的错误代码将写入 A400 中。错误代码 4101 ～ 42FF 对应 FAL 编号 001 ～ 2FF （0 ～ 511）。 ON：发生 FALS （006）错误 OFF：没有执行 FALS （006） （错误清除时此标志转为 OFF）。	清除	清除	A360 ～ A391, A400
A403	A403.00 ～ A403.08	存储区错误位置	当发生存储器错误时，存储器错误标志 （A401.15）为 ON，而且下列标志将转为 ON ，指示出发生错误的存储区： A403.00：用户程序 A403.04：PLC 设置 A403.05：I/O 注册表 A403.07：路由表 A403.08：CPU 总线单元设置 错误时 CPU 单元继续运作，CPU 单元前面板的 ERR/ALM 指示灯会闪烁。 ON：错误 OFF：正常 （清除错误后，相应标志转为 OFF）。	清除	清除	A401.15
A404	A404.00 ～ A404.07	I/O 总线错误槽位编号	保存出现 I/O 总线错误处的 8 位二进制槽编号 （00 ～ 09）。如果端盖未接到 CPU 机架或扩展机架，存储内容为 F hex。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。I/O 总线错误标志 （A401.14）转 ON。 00 ～ 09 （槽位编号 00 ～ 09） （错误清除时此标志转为 OFF）。	清除	清除	A401.14
	A404.08 ～ A40.415	I/O 总线错误机架编号	保存出现 I/O 总线错误处的 8 位二进制机架号 （00 ～ 03）。如果端盖未接到 CPU 机架或扩展机架，存储内容为 F hex。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。I/O 总线错误标志 （A401.14）转 ON。 00 ～ 03 （机架编号 00 ～ 03） （错误清除时此标志转为 OFF）。	清除	清除	A401.14
A405	A405.08	中断输入单元位置错误标志	中断输入单元未连接到 CPU 机架上 CPU 单元侧旁的槽位 0 ～ 3 时 ON。 即使一个单元物理上是接在槽位 0 ～ 4 这五个位置中，但也可能在 I/O 表中登录名义单元，使它的登录位置与物理位置不一致。 ON：位置错误 OFF：位置正确 （错误清除时此标志转为 OFF）。	清除	清除	A401.10
	A405.15	外设服务时间过长标志	在并行处理模式下，如果外设服务时间超过 2 秒时转 ON ，这也将造成循环时间错误并使操作停止。 ON：过长 OFF：未过长	清除	清除	A268

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A407	A407.00 ～ A407.12	I/O 点数过多，详细	当发生下述任一值溢出或重复错误时，将以二进值数据保存相应值。 CPU 单元将停止操作并且 CPU 单元前面板的 ERR/ALM 指示灯会点亮。关于错误发生的详细信息，请参见 I/O 点数过多，原因（A407.13 ～ A407.15）。 1. 当 I/O 表中设置的 I/O 点数 （包括从站机架）超过此 CPU 单元允许的最大数时，在这里写入 I/O 点数。 2. 当中断输入点数超过 32 点的中断输入点数。 3. 当扩展机架编号超过允许最大值时，此处写入机架数。 （在错误时，相关值写入 A407.00 ～ A407.12，在错误消除后，这些位将被清除）。	清除	清除	A401.11, A407.13 ～ A407.15
	A407.13 ～ A407.15	I/O 点数过多，原因	这些位的 3 位二进制数值表示 I/O 点数过多错误的原因，并将值的内容写入 A407.00 ～ A407.12。 值 000 ～ 101 （0 ～ 5）对应 “I/O 点数过多，原因”中描述的原因 1 ～ 6。 000： I/O 总数过多 001：中断输入点过的多 101：机架过多 111：一个机架上的单元过多 （错误消除后，这些位将被清除）。	清除	清除	——
A408	A408.00 ～ A408.07	基本 I/O 单元错误，槽位编号	在一个基本 I/O 单元错误时，A402.12 将转为 ON，且发生错误处的槽编号将以二进制写入这里。 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。 00 ～ 09 十六进制 （槽位 0 ～ 9） （错误消除后，这些位将被清除）。	清除	清除	A402.12
	A408.08 ～ A408.15	基本 I/O 单元错误，机架编号	在一个基本 I/O 单元错误时，A402.12 将转为 ON，且发生错误处的机架编号将以二进制写入这里。 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。 00 ～ 03 十六进制 （机架 0 ～ 3） （错误消除后，这些位将被清除）。	清除	清除	A402.12
A409	A409.00 ～ A409.03	扩展机架编号重复标志	当从 CX-Programmer 设置扩展机架的起始字，并且二个机架的分配字有重叠或机架的起始字超过 CIO 0901 时，对应的标志位转 ON。位 00 ～ 03 对应机架 0 ～ 3。 ON：分配相同字到两个不同的机架或机架起始地址超出 CIO 0901 OFF：正常 （错误消除后，相应标志将被清除）。	清除	清除	——
A410	A410.00 ～ A410.15	CPU 总线单元编号重复标志	在 CPU 总线单元的编号重复时，重复错误标志 （A401.13）和在 A410 中对应的标志转 ON。位 00 ～ 15 对应单元号 0 ～ F。 CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 ON：有重复 OFF：无重复	清除	清除	A401.13

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A411 ～ A416	A411.00 ～ A416.15	特殊 I/O 单元编号重复标志	在特殊 I/O 单元的编号重复时，重复错误标志（A401.13）和 A411 ～ A416 中对应的标志转 ON。 位 00 ～ 15 对应于单元编号 0 ～ F。 （位 A411.00 ～ A416.15 对应于单元编号 000 ～ 05F （0 ～ 95）） CPU 单元将停止运行，且 CPU 前面板的 ERR/ALM 指示灯点亮。 由于扩展机架的起始字设置不当，将特殊 I/O 单元的字分配给扩展机架上的一个基本 I/O 单元时，对应的位也会转 ON。 ON：有重复 OFF：无重复	清除	清除	A401.13
A417	A417.00 ～ A417.15	CPU 总线单元错误，单元编号标志	在 CPU 单元和一个 CPU 总线单元之间的数据交换错误时， CPU 总线单元错误标志 （A402.07）和 A417 中对应错误发生处的单元编号的位将转为 ON 。位 00 ～ 15 对应于单元编号 0 ～ F。 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。 ON：错误 OFF：正常	清除	清除	A402.07
A418 ～ A423	A418.00 ～ A423.15	特殊 I/O 单元错误，单元编号标志	当 CPU 单元和一个特殊 I/O 单元之间的数据交换出现错误时，特殊 I/O 单元错误标志 （A402.06）转 ON。 A418 中位 00 ～ A423 中位 15，每一位对应一个单元编号 （0 ～ 95）。 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。 ON：错误 OFF：正常 如果单元的编号不确定，则没有对应标志转为 ON。 （在错误消除后，这些标志转为 OFF）。	清除	清除	A402.06
A426	A426.00 ～ A426.11	重复刷新错误，单元编号	A426.13 为 ON 时，这些位中保存了执行重复刷新的特殊 I/O 单元的单元编号。 错误消除后，这些位将被清除。 单元编号：000 ～ 05F （0 ～ 95） （在错误消除后，此标志转为 OFF）。	清除	清除	A402.13 A426.15
	A426.15	重复刷新错误，原因	A402.13 （重复刷新错误标志）为 ON 时，这个标志表示错误原因。 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。 ON：重复刷新	清除	清除	A402.13, A426.00 ～ A426.11
A427	A427.00 ～ A427.15	CPU 总线单元设定错误，单元编号标志	在出现 CPU 总线单元设置错误时， A402.03 和这个字中与此单元的单元号对应的位转为 ON。位 00 ～ 15 对应于单元号 0 ～ F。 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。 ON：设定错误 OFF：无设定错误	清除	清除	接通电源或识别出 I/O 时刷新 A402.03
A428 ～ A433	A428.00 ～ A433.15	特殊 I/O 单元设定错误，单元编号标志	在出现 CPU 总线单元设置错误时， A402.02 和这些字中与此单元的单元号对应的位转为 ON。位 00 ～ 15 对应于单元号 0 ～ F。 ON：设定错误 OFF：无设定错误 CPU 单元继续操作，前面板的 ERR/ALM 灯会闪烁。	清除	清除	接通电源或识别出 I/O 时刷新 A402.02

地址		名称	功能	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位					
A440	——	最大中断任务处理时间	包含以 0.1 ms 为单位的最大中断任务处理时间。 （在以最长时间处理此中断任务后，写入此值，并且在 PLC 开始运行时清零）。	清除	清除	见功能栏
A441	——	处理时间最长的中断任务	包含处理时间最长的中断任务的编号，十六进制值 8000 ～ 80FF 对应于任务号 00 ～ FF。在出现中断任务时位 15 转 ON。 （在以最长时间处理此中断任务后，写入此值，并且在 PLC 开始运行时清零）。	清除	清除	见功能栏
A446	A446.00 ～ A446.07	解除保护的次数	• 对每次解除保护进行计数。（如，因密码输入不正确而使保护失效） • 显示 UM 保护和任务保护解除的总次数。 • 当到达 255 （十进制）时，计数器停止计数。 • 当解除所有保护时，计数器将设定为 00 hex。	——	——	——

A-3-2 读 / 写区 （ 由用户设置）

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A500	A500.12	IOM 保持位	将此位置为 ON，以在从编程模式转换到运行 / 监视模式以及相反转换时保护 I/O 存储器的状态。 （如果 IOM 保持位本身的状态在 PLC 配置中 （IOM 保持位状态）是保护的，那么在 PLC 接通或电源中断时，I/O 存储区都是保持的）。	ON：保留 OFF：不保留	保留	清除 （可设定 PLC 设置为保持状态）	见功能栏 PLC 设置 （IOM 保持位状态设定）
	A500.13	强制状态保持位	将此位置为 ON，以在从编程模式转换到运行 / 监视以及相反方向转换模式时，保护已被强制置位或强制复位位的状态。在转换到 RUN 模式时被强制置位或复位的位全部恢复为它的缺省状态。 请注意始终将此位与 IOM 保持位 （A500.12）共同使用， IOM 保持位 （A500.12）要与此位同时打开。电源中断时，强制置位和强制复位位的状态也是保持的。 将 PLC 设置中的强制状态保持位状态置为保护，使在接通 PLC 时，保持强制置位和强制复位位的状态。	ON：保留 OFF：不保留	保留	清除 （可设定 PLC 设置为保持状态）	见功能栏 PLC 设置 （强制状态保持位状态设定）
	A500.14	错误记录复位位	将这个位置 ON，将错误记录指针 （A300）复位为 00。 错误记录区本身的内容 （A100 ～ A199）不会清除。（在错误记录指示器被复位后，这个位自动复位至零）。	OFF → ON：清除	保留	清除	A100 ～ A199, A300
	A500.15	输出 OFF 位	将这个位置为 ON 时，无论 I/O 存储器中输出位状态如何，基本 I/O 单元的所有输出全部转为 OFF。（与发生致命错误相同）。 根据单独的特殊 I/O 单元功能，分配输出位状态到 CIO 区中的特殊 I/O 单元 （如，外部模拟量输出）。	——	保留	保留	——
A501	A501.00 ～ A501.15	CPU 总线单元重启位	将这些位置 ON，使具有对应单元号的 CPU 总线单元重启 （初始化）。位 00 ～ 15 对应于单元编号 0 ～ F。 在重启位转 ON 时，对应的 CPU 总线单元初始化标志 （A302.00 ～ A302.15）也转 ON。在初始化完成时，重启位和初始化标志位同时转为 OFF。	OFF → ON：重启 ON → OFF：完成重启 单元完成重启时，由系统将它转为 OFF。	保留	清除	A302.00 ～ A302.15
A502 ～ A507	A502.00 ～ A507.15	特殊 I/O 单元位重启	将这些位置 ON，使具有对应单元号的特殊 I/O 单元重启 （初始化）。位 A502.00 ～ A507.15 对应于单元编号 0 ～ 95 。 在重启位转 ON 时，对应的特殊 I/O 单元的初始化标志 （A330.00 ～ A335.15） 也转 ON。在初始化完成时，重启位和初始化标志位同时转为 OFF。	OFF → ON：重启 ON ～ OFF：完成重启 单元完成重启时，由系统将它转为 OFF。	保留	清除	A330.00 ～ A335.15
A508	A508.09	微分监视完成标志	执行微分监视期间，在建立了微分监视条件时置 ON。 （在微分监视启动时，这个标志清零）。	ON：已建立监视条件 OFF：未建立	保留	清除	——
	A508.11	跟踪触发器监视标志	通过跟踪启动位 （A508.14）建立起触发条件时置 ON。通过采样启动位 （A508.15）启动下一次数据跟踪时置 OFF。	ON：触发条件建立 OFF：未建立或未跟踪	保留	清除	——
	A508.12	跟踪完成标志	在执行一次跟踪期间，当完成了跟踪存储器一个区域的采样时置 ON。 在下一个采样启动位 （A508.15）由 OFF 转为 ON 时置 OFF。	ON：完成跟踪 OFF：未跟踪或正在跟踪	保留	清除	——

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A508	A508.13	跟踪忙标志	当采样启动位 （A508.15）从 OFF 转为 ON 时置 ON。跟踪完成时置 OFF。	ON：正在跟踪 OFF：不在跟踪（不采样）	保留	清除	——
	A508.14	跟踪启动位	将此位从 OFF 置为 ON 以建立起触发条件。由延迟值（正 / 负）表示的偏移量决定哪一段采样的数据有效。	ON：已建立跟踪触发条件 OFF：未建立	保留	清除	——
	A508.15	采样启动位	当从 CX-Programmer 将此位从 OFF 置为 ON 时，启动一次数据跟踪，PC 用下列三种方法中一种开始向跟踪存储区存储数据： 1. 按规定时间间隔 （10 ～ 2,550 ms）采样数据。 2. 在程序中执行 TRSM （045）时采样数据。 3. 在每个循环结束时采样数据。 只能从 CX-Programmer 控制 A50815 的操作。	OFF → ON：启动数据跟踪 （采样） 通过编程设备置 ON。	保留	清除	——
A510 和 A511	——	启动时间	这字内包含电源接通的时间，每次接通电源时刷新内容，以 BCD 码储存数据。 A510.00 ～ A510.07：秒 （00 ～ 59） A510.08 ～ A510.15：分 （00 ～ 59） A511.00 ～ A511.07：时 （00 ～ 23） A511.08 ～ A511.15：日 （01 ～ 31）	见功能栏	保留	见功能栏	接通电源时刷新
A512 和 A513	——	断电时间	这字内包含电源断开的时间，每次电源中断时刷新内容，以 BCD 码储存数据。 A512.00 ～ A512.07：秒 （00 ～ 59） A512.08 ～ A512.15：分 （00 ～ 59） A513.00 ～ A513.07：时 （00 ～ 23） A513.08 ～ A513.15：日 （01 ～ 31） （启动时不清除）	见功能栏	保留	保留	断开电源时写入
A514	——	电源中断次数	包含从电源第一次接通起，电源的中断次数，以二进制存储数据，复位时用 0000 hex 复盖当前值。 （这个字在启动时不清零，但在存储器损坏检测标志（A395.11）转 ON 时被清除）。	0000 ～ FFFF 十六进制	保留	保留	电源接通时刷新 A395.11
A515 ～ A517	——	运行开始时间	作为将运行模式改为 RUN 或 MONITOR 模式的结果，运行开始的时间以 BCD 形式保存在此。 A515.00 ～ A515.07：秒 （00 ～ 59） A515.08 ～ A515.15：分 （00 ～ 59） A516.00 ～ A516.07：时 （00 ～ 23） A516.08 ～ A516.15：日 （01 ～ 31） A517.00 ～ A517.07：月 （01 ～ 12） A517.08 ～ A517.15：年 （00 ～ 99） 注：上次开始的时间在打开电源后被保存，直到运行开始。	见功能栏	保留	保留	见功能栏

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A518 ～ A520	——	运行结束时间	作为将运行模式改为 PROGRAM 模式的结果，运行停止的时间以 BCD 的形式保存在此。 A518.00 ～ A518.07：秒 （00 ～ 59） A518.08 ～ A518.15：分 （01 ～ 59） A519.00 ～ A519.07：时 （00 ～ 23） A519.08 ～ A519.15：日 （01 ～ 31） A520.00 ～ A520.07：月 （01 ～ 12） A520.08 ～ A520.15：年 （00 ～ 99） 注：若在运行中发生错误，错误的时间将被保存。若之后运行模式改为 PROGRAM 模式，进入 PROGRAM 模式的时间将被保存。	见功能栏	保留	保留	见功能栏
A523	——	电源接通总时间	包含以 10 小时为单位的 PLC 通电的累计时间，数据以二进制保存，并且每 10 小时刷新一下。在复位这个值时，用 0000 hex 复写。（在启动时不清除这个字，但在存储器损坏检测标志 （A395.11）转 ON 时，清除为 0000）。	0000 ～ FFFF 十六进制	保留	保留	——
A526	A526.00	串行端口重启位	此位置 ON 可重启 RS-232C 端口。（在端口以外设总线方式工作时，不能使用这个位）。重启处理完成时，自动将此位转为 OFF。	OFF → ON：重启	保留	清除	——
A527	A527.00 ～ A527.07	禁止在线编辑位有效	仅当这个字节的值为 5A 时，禁止在线编辑位 （A527.09）为有效。 从 CX-Programmer 将此字节设置为 5A，并将 A527.09 置为 ON。 （当 PLC 在 MONITOR 模式时，可通过在线编辑修改和增加程序）。	5 A： A527.09 允许 其他值： A527.09 禁止	保留	清除	A527.09
	A527.09	禁止在线编辑位	此位转为 ON，以禁止在线编辑，只有在 A527.00 ～ A527.07 置为 5A 时，对这个位的设置才有效。	ON：禁止 OFF：允许	保留	清除	A527.00 ～ A527.07
A528	A528.00 ～ A528.07	串行端口错误标志	这些标志指示 RS-232C 端口出现的错误类型，在 RS-232C 端口重启时，它们自动转为 OFF。 （这些标志在串行网关模式下有效。在外设总线模式下这些标志无效，在 NT 链接模式只有位 5 有效）。 轮流检测单元： 位 5：发生超时错误时打开 已检测单元： 位 3：发生帧错误时打开 位 4：发生超限错误时打开 位 5：发生超时错误时打开 这些位可由 CX-Programmer 来清除。	位 0 和 1：未使用 位 2：发生校验错误时打开 位 3：发生成帧错误时打开 位 4：发生超限错误时打开 位 5：发生超时错误时打开 位 6 和 7：未使用	保留	清除	——
A529	——	用于系统错误仿真的 FAL/FALS 编号	设置一个虚拟 FAL/FALS 编号，为使用 FAL （006）或 FALS （007）指令模拟系统错误。 在执行 FAL （006）或 FALS （007），并且 A529 中的编号与在指令操作数中指定的相同时，将生成指令操作数中所给出的系统错误以替代用户定义的错误。	0001 ～ 01FF hex：FAL/FALS 编号 1 ～ 511 0000 或 0200 ～ FFFF hex：没有系统错误仿真的 FAL/FALS 编号。（不会产生错误号）	保留	清除	——
A530	——	电源中断禁止区设定	将此置为 A5A5 Hex 以禁止 DI （693）和 EI （694）指令之间的电源中断 （除电源 OFF 中断任务以外）。	A5A5 Hex：允许掩膜电源中断处理。 其他：禁止掩膜电源中断处理。	清除	清除	——

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A580（见注）	A580.00 ～ A580.03	FB 通信指令重试	自动保存在 PLC 设置中指定的 FB 通信指令设定中的重试次数。	0 ～ F hex	与 PLC 设置中相同	清除	操作开始时写入
A581（见注）	——	FB 通信指令响应监控时间	自动保存 PLC 设置中所设的 FB 通信指令响应监控时间。	0001 ～ FFFF hex（单元：0.1 s; 范围：0.1 ～ 6553.5） 0000 hex：2 s	与 PLC 设置中相同	清除	操作开始时写入
A582（见注）	——	FB DeviceNet 通信指令响应监控时间	自动保存 PLC 设置中所设的 FB DeviceNet 通信指令响应监控时间。	0001 ～ FFFF hex（单元：0.1 s; 范围：0.1 ～ 6553.5） 0000 hex：2 s	与 PLC 设置中相同	清除	操作开始时写入
A595 和 A596	——	后台执行的 IR00 输出	在指定一个变址寄存器作为后台处理指令的输出时，A595 和 A596 将接收这个输出而不用 IR00。	0000 0000 ～ FFFF FFFF hex （A596 中保存最左位数字）	清除	清除	——
A597	——	后台执行的 DR00 输出	在指定一个数据寄存器作为后台处理的指令的输出时，A597 接收这个输出而不用 DR00。	0000 ～ FFFF hex	清除	清除	——
A598	A598.00	FPD 自学习位	将此位变为 ON 时，可以通过自学习功能自动设定监视时间。 当 A598.00 是 ON 时，执行条件变成 ON 后 FPD（269）测量诊断输出需要多长时间变成 ON，如果测得的时间超过监测时间，则将测得值乘以 1.5 作新的检测时间存储。 （只有字地址被指定为监测时间的操作数时，可以使用自学习功能）。	ON：教学监视时间 OFF：关闭教学功能	清除	清除	——
	A598.01	后台作业的相等标志	执行在后台作业中的指令 SRCH （181）找到了匹配的数据时置为 ON。	ON：在表中找到搜索数据 OFF：没有找到搜索数据	清除	清除	——
A600 ～ A603	——	宏区输入字	执行 MCRO （099）时，复制 4 个指定源字 （输入参数首字开始）的内容到此。并将复制的 4 字内容传递给子程序。	输入数据： 4 字	清除	清除	——
A604 ～ A607	——	宏区输出字	执行了在 MCRO （099）中指定的子程序后，将子程序的结果从 A604 至 A607 传送到指定目标字 （输出参数字）。	输出数据： 4 字	清除	清除	——
A619	A619.02	串行端口设定更改标志	RS-232C 端口的通信设置更改时为 ON。当执行 STUP （237）时此标志变为 ON，在设定值更改后此标志转为 OFF。	ON：变更 OFF：不变更	保留	清除	——

注　这些辅助区位 / 字不由用户所写。重新发送次数和响应监控时间必须由用户在 PLC 设置中的 FB 通信指令设置中进行设定，特别当从欧姆龙 FB 库使用功能块来执行 FINS 报文或 DeviceNet 显式报文通信时。 PLC 设置中欧姆龙 FB 库设定中所设的值将自动保存在相关的辅助区字 A580 ～ A582 中并由来自欧姆龙 FB 库的功能块所使用。

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A620	A620.01	通信单元 0，端口 1 设定更改标志	当某个端口的设定值正在更改时与之相对应的标志为 ON。 当执行了 STUP （237）时此标志变为 ON，在设定值发生变更后，通过来自串行通信单元事件，将标志转为 OFF。 用户也可以通过将这个标志转 ON，指示串行端口的设置已变更。	ON：变更 OFF：不变更	保留	清除	——
	A620.02	通信单元 0，端口 2 设定更改标志					——
	A620.03	通信单元 0，端口 3 设定更改标志					——
	A620.04	通信单元 0，端口 4 设定更改标志					——
A621 ～ A635	A621.00 ～ A635.04	通信单元 1 ～ 15，端口 1 ～ 4 设定变更标志	与上面相同。	ON：变更 OFF：不变更	保留	清除	——
A650	A650.14	更换错误标志	当更换起始位 （A650.15）打开以更换程序时置 ON，但会有错误。若更换起始位再次打开，更换错误标志将为 OFF。	ON：更换错误 OFF：无更换错误，或更换起始位（A65015）打开。	保留	清除	——
	A650.15	更换启动位	如果程序密码 （A651）是有效的 （A5A5 hex），则在更换启动位变为 ON 时程序开始替换。执行程序更换期间不可将更换启动位转为 OFF。 当电源接通或完成了程序替换，更换启动位会变为 OFF，与更换是否正常完成或是否有错误发生无关。 通过 CX-Programmer，PT 或主机读取更换启动位，可以确定是否正在执行程序替换。	ON：更换程序 OFF：完成了程序更换，或接通电源后。	保留	清除	——
A651	——	程序密码	替换程序的加密类型。 A5A5 hex：允许更换启动位 （A650.15） 其它任何值：禁止更换启动位 （A650.15） 当电源接通或完成了程序替换时，更换启动位会将转为 OFF，与更换是否正常完成或是否有错误发生无关。	——	保留	清除	——

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A654 ～ 657	——	程序文件名	当程序替换开始时，程序的文件名以 ASCII 形式存储。文件名长度不包括扩展名，小于 8 个字符。 文件名按以下列顺序存放：A654 ～ A657 （例如，从最低字到最高字）及从最高字节到最低的，如果文件名小于 8 个字符，最低剩下的字节和最高剩下的字节将由空格 （20 hex）填充，且 Null 和空格字符不能用在文件名中。 例：文件名是 ABC.OBJ 15 0 A654 41 42 A655 43 20 A656 20 20 A657 20 20	——	保留	清除	——
A720 ～ A722	——	电源打开 时钟数据 1	这些字中包含了与字 A510 ～ A511 中所存储的启动时间相同的时间数据。如，月和年等信息。 A720.00 ～ A720.07：秒 （00 ～ 59） A720.08 ～ A720.15：分 （00 ～ 59） A721.00 ～ A721.07：时 （00 ～ 23） A721.08 ～ A721.15：日 （01 ～ 31） A722.00 ～ A722.07：月 （01 ～ 12） A722.08 ～ A722.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A723 ～ A725	——	电源打开 时钟数据 2	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前一次启动的时间。 A723.00 ～ A723.07：秒 （00 ～ 59） A723.08 ～ A723.15：分 （00 ～ 59） A724.00 ～ A724.07：时 （00 ～ 23） A724.08 ～ A724.15：日 （01 ～ 31） A725.00 ～ A725.07：月 （01 ～ 12） A725.08 ～ A725.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A726 ～ A728	——	电源打开时钟数据 3	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前两次启动的时间。 A726.00 ～ A726.07：秒 （00 ～ 59） A726.08 ～ A726.15：分 （00 ～ 59） A727.00 ～ A727.07：时 （00 ～ 23） A727.08 ～ A727.15：日 （01 ～ 31） A728.00 ～ A728.07：月 （01 ～ 12） A728.08 ～ A728.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A729 ～ A731	——	电源打开时钟数据 4	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前三次启动的时间。 A729.00 ～ A729.07：秒 （00 ～ 59） A729.08 ～ A729.15：分 （00 ～ 59） A730.00 ～ A730.07：时 （00 ～ 23） A730.08 ～ A730.15：日 （01 ～ 31） A731.00 ～ A731.07：月 （01 ～ 12） A731.08 ～ A731.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。

地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A732 ～ A734	——	电源打开时钟数据 5	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前四次启动的时间。 A732.00 ～ A732.07：秒 （00 ～ 59） A732.08 ～ A732.15：分 （00 ～ 59） A733.00 ～ A733.07：时 （00 ～ 23） A733.08 ～ A733.15：日 （01 ～ 31） A734.00 ～ A734.07：月 （01 ～ 12） A734.08 ～ A734.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A735 ～ A737	——	电源打开时钟数据 6	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前五次启动的时间。 A735.00 ～ A735.07：秒 （00 ～ 59） A735.08 ～ A735.15：分 （00 ～ 59） A736.00 ～ A736.07：时 （00 ～ 23） A736.08 ～ A736.15：日 （01 ～ 31） A737.00 ～ A737.07：月 （01 ～ 12） A737.08 ～ A737.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A738 ～ A740	——	电源打开时钟数据 7	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前六次启动的时间。 A738.00 ～ A738.07：秒 （00 ～ 59） A738.08 ～ A738.15：分 （00 ～ 59） A739.00 ～ A739.07：时 （00 ～ 23） A739.08 ～ A739.15：日 （01 ～ 31） A740.00 ～ A740.07：月 （01 ～ 12） A740.08 ～ A740.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A741 ～ A743	——	电源打开时钟数据 8	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前七次启动的时间。 A741.00 ～ A741.07：秒 （00 ～ 59） A741.08 ～ A741.15：分 （00 ～ 59） A742.00 ～ A742.07：时 （00 ～ 23） A742.08 ～ A742.15：日 （01 ～ 31） A743.00 ～ A743.07：月 （01 ～ 12） A743.08 ～ A743.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A744 ～ A746	——	电源打开时钟数据 9	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前八次启动的时间。 A744.00 ～ A744.07：秒 （00 ～ 59） A744.08 ～ A744.15：分 （00 ～ 59） A745.00 ～ A745.07：时 （00 ～ 23） A745.08 ～ A745.15：日 （01 ～ 31） A746.00 ～ A746.07：月 （01 ～ 12） A746.08 ～ A746.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。
A747 ～ A749	——	电源打开时钟数据 10	这些字包含电源在保存于字 A510 ～ A511 中的启动时间之前九次启动的时间。 A747.00 ～ A747.07：秒 （00 ～ 59） A747.08 ～ A747.15：分 （00 ～ 59） A748.00 ～ A748.07：时 （00 ～ 23） A748.08 ～ A748.15：日 （01 ～ 31） A749.00 ～ A749.07：月 （01 ～ 12） A749.08 ～ A749.15：年 （00 ～ 99）	见功能栏	保留	保留	电源打开时写入。

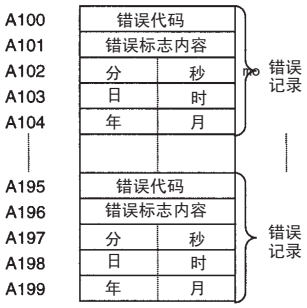
地址		名称	功能	设定	模式变更后状态	启动时状态	写时间 / 相关标志及设定
字	位						
A10001 ～ A10003	——	生产批号	CPU 单元的生产批号为以 BCD 数据保存。可通过下述区识别硬件。 hex A10000.00 ～ A10000.07：00 hex A10000.08 ～ A10000.15：00 hex A10001.00 ～ A10000.07：00 hex A10001.08 ～ A10001.15：日 （01 ～ 31） A10002.00 ～ A10002.07：月 （01 ～ 12） A10002.08 ～ A10002.15：年 （00 ～ 09） A10003.00 ～ A10003.07：00 hex A10003.08 ～ A10003.15：00 hex	——	——	——	——

注　在 CJ 系列 PLC 中在一个特定的只读存储区提供下列标志，并且可用下表给出的标记指定。这些标志不包含在辅助区中，详情请参见 6-18 条件标志和 6-19 时钟脉冲。

错误标志	条件标志区
存取错误标志	
进位标志	
大于标志	
等于标志	
小于标志	
取反标志	
溢出标志	
下溢标志	
大于或等于标	
不等于标志	
小于或等于标志	
常 ON 标志	
常 OFF 标志	
0.02 秒时钟脉冲	时钟脉冲区
0.1 秒 时钟脉冲	
0.2 秒时钟脉冲	
1 秒时钟脉冲	
1 分时钟脉冲	

A-3-3　辅助区操作细节

A100 ～ A199：错误记录区



如果在 1998 年 4 月 1 日 17 时 10 分 30 秒发生一个存储区错误（错误代码 80F1），并且在 PLC 设置中的错误（04 Hex）。此时在出错记录中将产生下列数据。

80	F1
00	04
10	30
01	17
98	04

如果在 1997 年 5 月 2 日 8 点 30 分 15 秒出现 FALS 编号为 001 的 FALS 错误，此时在出错记录中会产生下列数据。

C1	01
00	00
30	15
02	08
97	05

错误代码和错误标志

分类	错误代码	表示	错误标志
系统定义的致命错误	80F1	存储器错误	A403
	80C0 ～ 80C7 80CE, 80CF	I/O 总线错误	A404
	80E9	重复单元 / 机架编号错误	A410, A411 ～ 416 （见注 3）
	80E1	I/O 点数过多错误	A407
	80E0	I/O 设定错误	——
	80F0	参数错误	A295 ～ 299 （见注 4）
	809F	循环时间过长错误	——
	80EA	扩展机架编号重复错误	A409.00 ～ 409.07
用户定义的致命错误	C101 ～ C2FF	执行 FALS 指令 （见注 1）	——
用户定义的非致命错误	4101 ～ 42FF	执行 FAL 指令 （见注 2）	——
系统定义的非致命错误	008B	重复刷新错误	A426
	009A	基本 I/O 错误	A408
	009B	PLC 设置错误	A406
	0200 ～ 020F	CPU 总线单元错误	A417
	0300 ～ 035F	特殊 I/O 单元错误	A418 ～ 423 （见注 5）
	00F7	电池故障	——
	0400 ～ 040F	CPU 总线单元设定错误	A427
	0500 ～ 055F	特殊 I/O 单元设定错误	A428 ～ 433 （见注 5）
	00F1	备份存储器错误	A315.15
	00F2	标签存储器错误	A315.12
	00F3	其他非致命错误（如，系统工作存储器错误）	A402.00

注　1　C101 ～ C2FF 分别存储 FALS 编号 001 ～ 511 的故障。

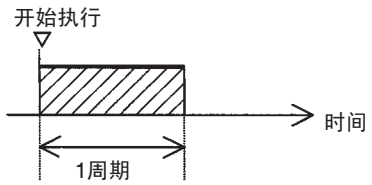
2　4101 ～ 42FF 分别存储 FAL 编号 001 ～ 511 的故障。

3　重复编号错误出错标志的内容如下：位 0 ～ 7：单元编号（二进制），00 ～ 5F Hex 为特殊 I/O 单元，00 ～ 0F Hex 为 CPU 总线单元。
位 8 ～ 14：全部为零。
位 15：单元类型，0 为 CPU 总线单元，1 为特殊 I/O 单元。

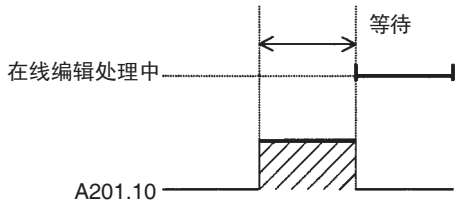
4　仅存储 A295 的内容，作为程序错误的出错标志内容。

5　存储 0000 Hex 作为出错标志的内容。

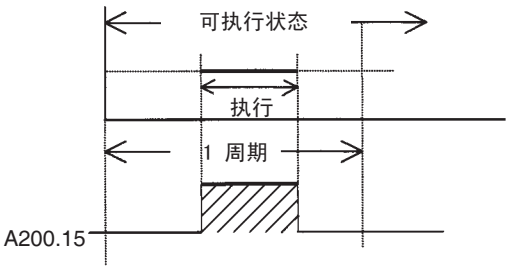
A200.11：第一循环标志



A201.10：在线编辑等待标志

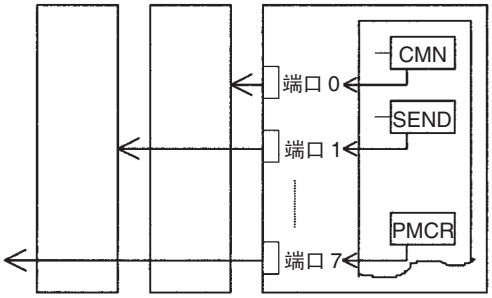


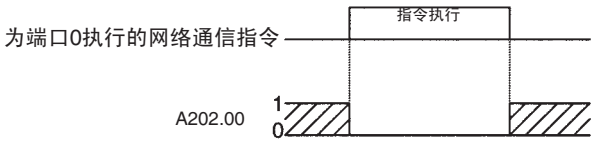
A200.15：初始化任务标志

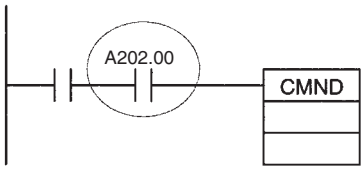


在一个任务进入可执行状态后的第一次执行中 A200.15 会置为 ON。且其只有在任务第一次任务的执行中会置为 ON，在以后的循环，不会置为 ON。

A202.00 ～ A202.07：通信端口允许标志

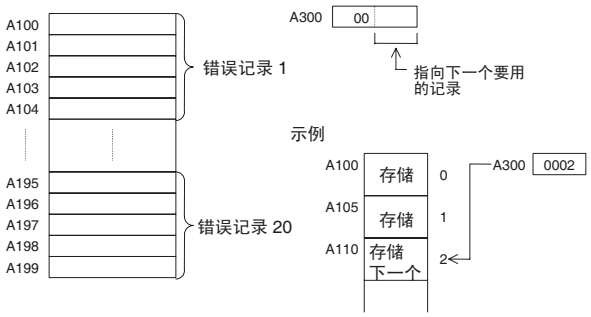




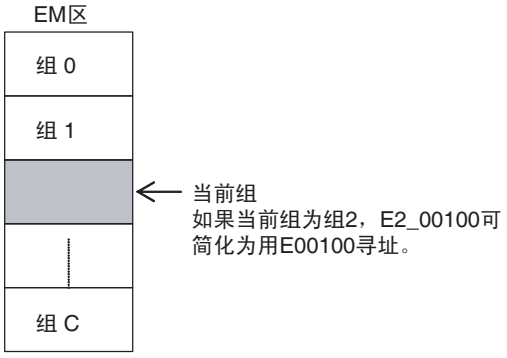


此程序设计为只有在 A202.00为ON时才执行 CMND（490）。

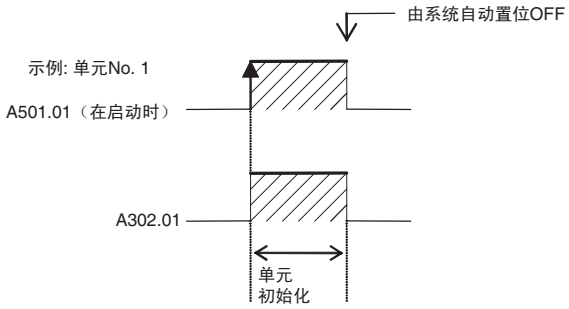
A300：错误记录指针



A301：当前 EM Bank



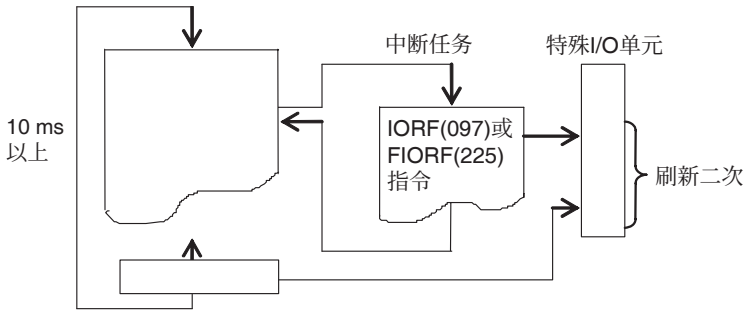
A501.01 ～ A501.15：CPU 总线单元重启位



A401.09：程序错误

	错误	地址
A401.09 为 ON 时的程序错误	UM 溢出错误标志	A295.15
	非法指令标志	A295.14
	分配溢出错误标志	A295.13
	任务错误标志	A259.12
	无 END （001）错误标志	A295.11
	非法区访问错误标志	A295.10
	DM/EM 间接寻址错误标志	A295.09

A426.15：重复刷新错误原因



A-4 PLC 存储器地址的内存印象图

A-4-1 PLC 存储器地址

PLC 存储器地址在变址寄存器 （IR00 ～ IR15）中设置，以对 I/O 存储器间接寻址。通常，使用 MOVE TO REGISTER （MOVR （560））和 MOVE TIMER/COUNTER PV TO REGISTER （MOVRW （561））指令将 PLC 存储器地址设入变址寄存器中。

一些指令，诸如 DATA SEARCH （SRCH （181）） 、 FIND MAXIMUM （MAX （182） ）和 FIND MINIMUM （MIN （183）），将处理结果输出至变址寄存器中以表示 PLC 存储器地址。

此外还有些指令，可直接指定变址寄存器使用由其它指令保存在其中的 PLC 存储器地址。这些指令包括 DOUBLE MOVE (MOVL(498))、一些符号比较指令 (=L、 <>L、 <L、 >L、 <=L 和 >=L)、 DOUBLE COMPARE (CMPL(060))、DOUBLE DATA EXCHANGE (XCGL(562))、DOUBLE INCREMENT BINARY (++L(591))、 DOUBLE DECREMENT BINARY (--L(593))、 DOUBLE SIGNED BINARY ADD WITHOUT CARRY (+L(401))、DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY (-L(411))、SET RECORD LOCATION (SETR(635)) 和 GET RECORD LOCATION (GETR(636))。

所有 PLC 存储器地址是连续的且用户必须知道存储器区的顺序和边界。作为参考，本附录末尾的表中提供了 PLC 存储器地址。

正确使用注意事项

应在任何可能之处避免在程序中直接设置 PLC 存储器地址。若在程序中设置了 PLC 存储器地址，程序与新 CPU 单元型号或存储器规划已更改的 CPU 单元的兼容性将降低。

正确使用注意事项

地址偏移和数组变量注意事项

使用下述功能时，可能要根据偏移值对下一数据区中的 PLC 存储器地址进行指定。

- 使用偏移值，间接指定地址。
- 在数组变量中间接指定要素编号。

编程时请注意防止数据区超出范围或考量本存储器规划中的数据区分配顺序。

A-4-2 存储器分配

下表所述为 I/O 存储器区的 PLC 存储器地址。I/O 存储器区为由电池进行备份。如果没有电池，则 I/O 存储器区的内容将损坏。

注　切勿访问为系统保留的区域。

分类	PLC 存储器地址（hex）	用户地址	区
I/O 存储器区	00000 ～ 0B7FF		系统备用
	0B800 ～ 0B801	TK0000 ～ TK0031	任务标志区
	0B802 ～ 0B83F		系统备用
	0B840 ～ 0B9FF	A000 ～ A447	只读辅助存储区
	0BA00 ～ 0BDFF	A448 ～ A1471	读 / 写辅助存储区
		A10000 ～ A11535	只读辅助存储区
	0BE00 ～ 0BEFF	T0000 ～ T4095	定时器完成标志
	0BF00 ～ 0BFFF	C0000 ～ T4095	计数器完成标志
	0C000 ～ 0D7FF	CIO 0000 ～ CIO 6143	CIO 区
	0D800 ～ 0D9FF	H000 ～ H511	保持区
	0DA00 ～ 0DDFF	H512 ～ H1535	保持区 这些字仅用于功能块。
	0DE00 ～ 0DFFF	W000 ～ W511	工作区
	0E000 ～ 0EFFF	T0000 ～ T4095	定时器 PV
	0F000 ～ 0FFFF	C0000 ～ C4095	计数器 PV
	10000 ～ 17FFF	D00000 ～ D32767	DM 区
	18000 ～ 1FFFF	E0_00000 ～ E0_32767	EM 区 组 0
	20000 ～ 27FFF	E1_00000 ～ E1_32767	EM 区 组 1 hex
	:	:	:
	D8000 ～ DFFFF	E18_00000 ～ E18_32767	EM 区 组 18 hex
	:	:	
	F8000 ～ FFFFF	E_00000 ～ E_32767	EM 区，当前组 （见注）

注　当前程序中指定的 EM 区组的内容被保存在这些地址中。例如，若指定了组 1，20000 ～ 27FFF 中同样的内容将被保存到 F8000 ～ FFFFF。

A-5 电源中断时操作

A-5-1 电源掉电操作

若 CPU 单元电源掉电时执行以下处理。若在 RUN 或 MONITOR 模式下电源电压降至最小额定电压的 85%（对于 CJ1W-PD025 DC 电源单元为 80%，对于 CJ1W-PD022 DC 电源单元为 90%）以下时将执行电源掉电处理。

1. CPU 单元停止工作。
2. 输出单元的输出全部转为 OFF。

注 1 与在 PLC 设置中设定的启动时 I/O 存储器保持位状态与设定无关，所有的输出位都将转为 OFF。

2 AC 电源单元额定电压的 85% :
100 V AC：85 V AC
200 V AC：170 V AC
100 ～ 240V 输入 （宽范围）：85 V AC

3 DC 电源单元额定电压的 80%/90% :
CJ1W-PD025：19.2 V DC
CJ1W-PD022：21.6 V DC

瞬时电源中断

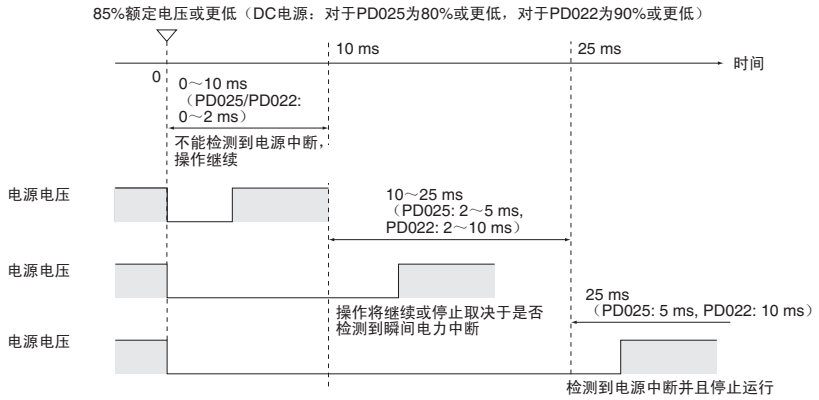
若电源仅是瞬时跌落，将进行以下处理 （电源瞬时中断）。

- 若瞬时电源中断时间小于 10 ms，系统将无条件持续运行。也就是说，电源电压为最小额定电压的 85% 或更低，然后恢复到 85% 或更高的时间不超过 10 ms。

注 使用了 DC 电源时， CJ1W-PD025 的电压为额定电压的 80%或更低时恢复至 80%或更高，或 CJ1W-PD022 的电压为额定电压的 90%或更低时恢复至 90%或更高只需少于 2ms。

- 持续大于 10ms 但小于 25ms（使用 DC 电源时，对于 CJ1W-PD025 为大于 2ms 但少于 5ms，对于 CJ1W-PD022 为大于 2ms 但少于 10ms）的瞬间电源掉电难以测定，可能检测到电源掉电也可能检测不到。
- 若瞬间电源掉电持续大于 25ms（使用了 DC 电源时，对于 CJ1W-PD025 为大于 5ms，对于 CJ1W-PD022 为大于 10ms）系统将无条件停止。

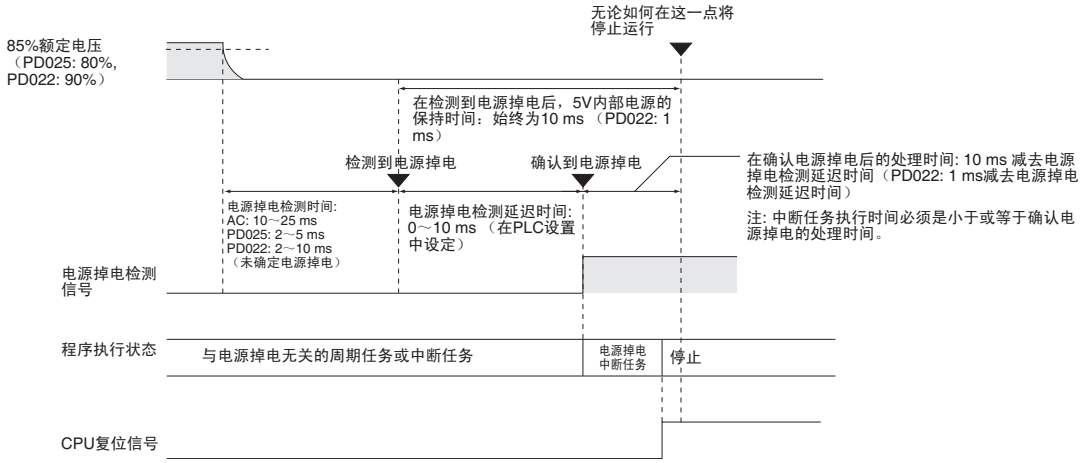
因此，若在上述项目 2 和 3 中给出的条件下操作停止则检测电源掉电所需的时间为 10 ～ 25ms（使用 DC 电源时，对于 CJ1W-PD025 为 2 ～ 5ms，对于 CJ1WPD022 为 2 ～ 10ms），停止操作所用的定时 （或开始执行电源掉电中断任务所用的定时）可通过在 PLC 设置中设定电源掉电检测延迟时间 (0 ～ 10 ms) 来延时。但是，在检测到瞬间电源掉电后 10ms 内操作总是将停止而不论 PLC 设置中如何设定。发生电源掉电时 CJ1W-PD022 电源单元的 5-VDC 输出的保持时间为 1ms，但是因此电源掉电中断任务禁止位和电源掉电检测时间参数为无效。



注　以上的时序图表示了当电源掉电检测时间被设定为 0 ms 时的一个实例。

电源掉电时序图

以下的时序图更详细的展示了 CPU 单元电源掉电时的操作。



电源掉电检测时间：　电压降到最小额定电压的 85％ （对于 CJ1W-PD025 DC 电源单元为 80％，对于 CJ1W-PD022 DC 电源单元为 90％）时检测到电源掉电所用的时间。

电源掉电检测延迟时间：　指从检测到电源掉电到确认电源掉电之间的延迟时间。在 PLC 设置中此时间可设置为 0 ～ 10ms。

如由于电源不稳定引起的电源中断，则在 PLC 设置中设定较长的电源掉电检测检测延迟时间 （最大 10ms）。CJ1W-PD022 电源单元只支持 1ms 的保持时间，因此该设定不可用。

电源保持时间：　指电源掉电后 5V 内部电源能保持的时间 （固定为 10ms）。电源掉电中断任务的执行时间不能超过 10ms 减去电源掉电检测延时时间所得的值 （电源掉电确认后的处理时间）。超过这段时间后，即使电源掉电中断任务没有完成也会被结束。 CJ1W-PD022 电源单元只支持 1ms 的保持时间，因此该设定不可用。

操作说明

- 在电源掉电检测时间（对于 AC 电源单元为 10 ～ 25ms 中某一时间，对于 CJ1W-PD025 DC 电源单元为 2 ～ 5ms 中某一时间，对于 CJ1W-PD022 DC 电源单元为 2 ～ 10ms 中某一时间）中，若 100 ～ 120 V DC 电源的电压降至最小额定电压的 85％（对于 CJ1W-PD025 DC 电源单元为 80％，对于 CJ1W-PD022 DC 电源单元为 90％）时将检测到电源掉电。

- 如果在 PLC 设置中设定了电源掉电检测延迟时间 （0 ～ 10 ms）则当超过此时间后执行以下操作：
  - 若禁止电源掉电中断任务 （默认的 PLC 设置设定），
    则 CPU 复位信号转为 ON 且 CPU 立即复位。
  - 若允许了电源掉电中断任务（在 PLC 设置中），在电源掉电中断任务执行后 CPU 复位信号将打开且 CPU 将复位。确保电源掉电中断任务将在 10ms 减去电源掉电检测延迟时间＝电源掉电后的处理时间内执行完毕。检测到电源掉电后 5V 内部电源将只被保持 10ms。发生电力掉电时 CJ1W-PD022 电源单元内部 5V 电源的保持时间为 1ms，因此电源掉电检测延迟时间和禁止电源掉电中断任务功能为不可用。

A-5-2 电源掉电的指令执行

若电源被中断且当 CPU 在 RUN 或 MONITOR 模式下操作时证实了中断，则当前正在执行的指令将被完成 *1 并将执行下列电源掉电处理。

- 若不允许电源掉电中断任务，CPU 单元将立即复位。
- 若允许了电源掉电中断任务，任务将被执行且 CPU 单元将被立即复位。

在 PLC 设置中允许或禁止电源掉电中断任务。

*1 仅当完成执行所需的时间小于等于电源掉电检测后的处理时间（10 ms －电源掉电检测延迟时间）时当前指令可执行完毕。若指令在这段时间内没有执行完毕，它将被中断，将执行上述处理。

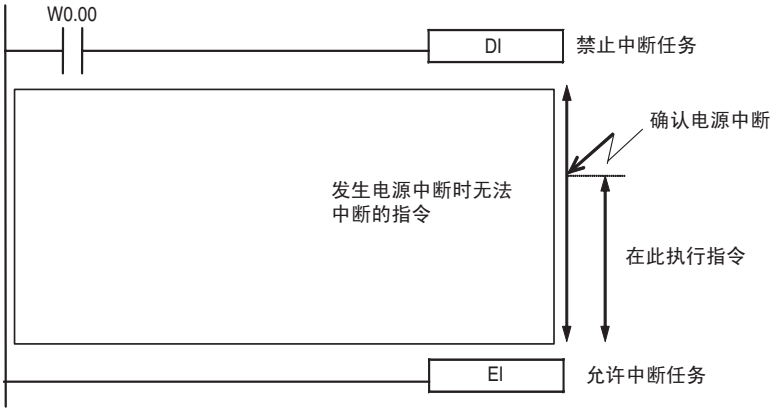
注 安装了 CJ1W-PD022 时检测到电源掉电后的处理时间为 1 ms。

在程序中禁止电源掉电处理

如果禁止了电源掉电中断任务，程序区也可以被保护不受电源掉电影响。由此，即使发生电源掉电，指令仍将在 CPU 单元执行电源掉电处理之前被执行。这可通过使用 DISABLE INTERRUPTS （DI（693））和 ENABLE INTERRUPTS （EI （694））指令来实现的。

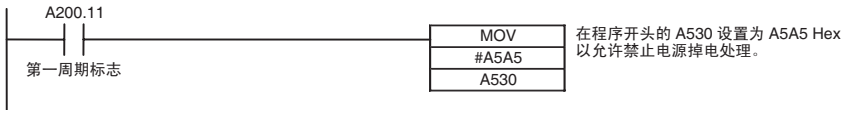
使用以下步骤。

1. 在程序段前插入 DI （693）以受禁止中断的保护然后在程序段后加上 EI （694）以允许中断。



2. 设置 A530 中禁止电源掉电中断为 A5A5 Hex，以禁止电源掉电处理。

注　电源掉电后 A530 通常被清除。为了防止这样，IOM 保持位（A500.12）必须为 ON 且 PLC 设置必须设为在启动时保持 IOM 保持位设定，或必须在程序开头包括以下类型的指令，以设置 A530 为 A5A5 Hex。



3. 在 PLC 设置中禁止电源掉电中断任务。

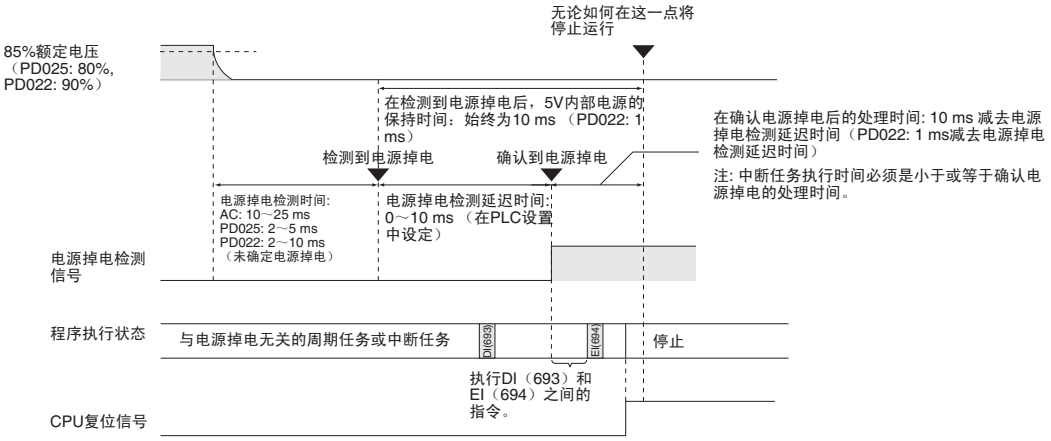
对于上述过程，所有 DI(693) 和 EI(694)(或 END) 之间的指令将在执行电源掉电中断之前完成 *1，即使在执行 DI(693) 和 EI(694) 之间的指令时发生电源掉电。

*1 仅当完成执行所需的时间小于等于电源掉电检测后的处理时间（10 ms －电源掉电检测延迟时间）时才可完成受保护的指令。若在这段时间内指令没有完成，将中断它们并执行上述处理。

注　1 若未在 PLC 设置中禁止电源掉电中断任务，则将执行电源掉电中断任务，并且一旦检测到电源掉电 CPU 就将被复位而不执行受保护指令。

2 若未在 PLC 设置中禁止电源掉电中断任务，将执行电源掉电中断任务，并且一旦检测到电源掉电 CPU 就将被复位而不执行受保护指令。

3 当安装了 CJ1W-PD022 时检测到电源掉电后的处理时间为 1 ms。



若将 A530 （电源中断禁止区设定）设定为除 A5A5 hex 以外的其他值以使屏蔽电源中断处理有效，则在当前指令执行完成后将启动电源中断处理。

附加信息

中断处理根据如下所示 A530 中的内容和 PLC 设置来执行。

电源中断处理		A530 （电源中断禁止区设定）	
		A5A5 hex （使屏蔽电源中断处理有效）	其他 （使屏蔽电源中断处理无效）
电源掉电中断任务（PLC 设置）	禁止	执行了所有 DI （693）和 EI （694）之间的指令，且 CPU 单元被复位。	当前指令的执行完成，且 CPU 单元被复位。
	允许	当前指令的执行已完成，执行了电源掉电中断任务，且 CPU 单元被复位。	

A-6 通过 Windows XP （SP2 及以上）或 Windows Vista 连接 EtherNet/IP

用于 Windows XP （SP2 及以上）和 Windows Vista 的更高版本防火墙保护中新增了关于 Ethernet 端口的数据通信限制。当从计算机上的 Ethernet 端口通过 EtherNet/IP 连接到 CPU 单元时，必须更改 Windows 防火墙的设置以使 CX-Programmer 通信为有效。

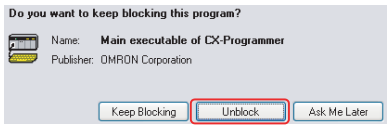
正确使用注意事项

Windows 防火墙主要设计用于保护避免受到来自外部设备 （如，通过因特网）的不适当访问。本章节中描述的关于变更 Windows 防火墙的设定，可使以 CX-Programmer 使用 EtherNet/IP 连接为有效。若在一公司网络中或其他网络中使用同一计算机，则请事先确认所做的变更不会影响网络安全。本章节中描述的变更内容为仅用于从 Ethernet 端口使用 EtherNet/IP 进行连接时。若通过其他端口进行连接 （如 USB 端口），则请不要进行变更。

A-6-1 变更 Windows 防火墙设置

Windows XP

1. 尝试通过一Ethernet端口在EtherNet/IP网络上连接CX-Programmer到PLC时，将显示Windows安全报警对话框。
2. 按下 “Unblock”键。

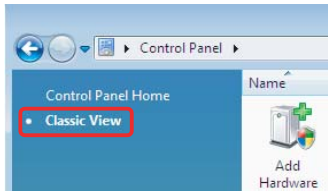


将接受来自 CX-Programmer 的 EtherNet/IP 连接，且 EtherNet/IP 连接将视为有效。

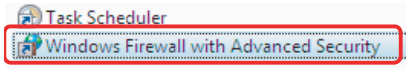
Windows Vista

尝试从 CX-Programmer 进行连接前，请先使用下述步骤对设定进行变更。
过程中可能显示用户帐号控制对话框。如果发生此类情况，请点击继续按钮并进行下一步骤。

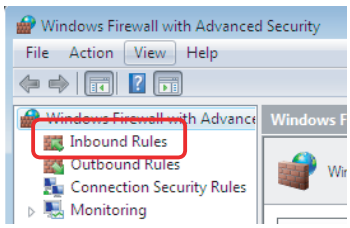
1. 从 Windows 开始菜单下选择 Control Panel（控制面板 ），并变更显示为 Classic View（经典视图）。



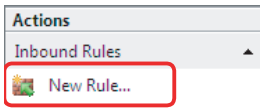
2. 打开管理工具并选择 Windows 防火墙的高级保护项 （Windows Firewall with Advanced Security），如下图所示。



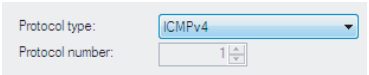
3. 在本地计算机的 Windows 防火墙的高级保护项（Windows Firewall with Advanced Security）下左侧的设定框中选择 “Inbound Rules”（进入规则）。



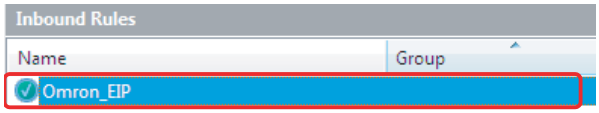
4. 在对话框右侧的动作区中，选择 “Inbound Rules”（进入规则）下的 “New Rule”（新建规则）。



5. 在新建进入规则的指导对话框中，根据下述对各步骤进行设定。点击下一步向前。

规则类型	选择 “ Custom”（自定义）
程序	选择 “All Programs”（所有程序）
协议和端口	选择 ICMPv4 作为协议类型。 
范围	全部选择 “Any IP address”（任一 IP 地址）
动作	选择 “ Allow the connection”（允许连接）
资料	选择 “ Domain, Private, and Public”（域、专用、公开）
名称	输入任意名称：如， Omron_EIP.

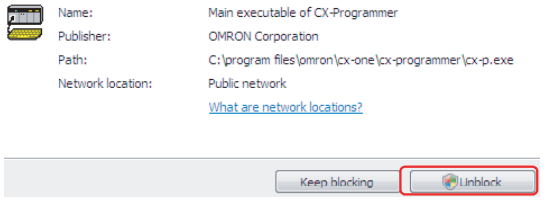
6. 点击完成按钮。您所定义的规则将注册在 “Inbound Rules”（进入规则）中 （如，Omron_EIP）。



7. 通过高级安全对话框关闭 Windows 防火墙。

8. 尝试通过一Ethernet端口在EtherNet/IP网络上连接CX-Programmer到PLC时，将显示Windows安全报警对话框。

9. 按下 “Unblock”键



将接受来自 CX-Programmer 的 EtherNet/IP 连接，且 EtherNet/IP 连接将视为有效

A-7 PLC 比较表：CJ 系列 和 CS 系列 PLC

下表所示仅为 CJ2 CPU 单元和其他 CS 系列及 CJ 系列 CPU 单元之间的功能比较。

功能比较

项目		CJ 系列				CS 系列
		CJ2 CPU 单元	CJ1-H-R CPU 单元	CJ1-H CPU 单元	CJ1-M CPU 单元	CS1-H CPU 单元
一般规格	尺寸 （高×深×宽）	90 × 65 × 80 mm	90 × 65 × 62 mm		90 × 65 × 31 mm 90 × 65 × 49 mm	130 × 101 × 71 mm
	重量	280 g 以下	200 g 以下		120 g 以下 170 g 以下	350 g 以下
	5 VDC 时电流消耗	0.82 A	0.99 A		0.58 A 0.64 A	0.82 A
	抗振动	符合 JIS 60068-2-6	符合 JIS C0040			
	耐冲击	符合 JIS 60068-2-27	符合 JIS C0041			
性能规格	最大程序容量	400K 步	250K 步			
	最大 I/O 容量	2,560 点				5,120 点
	系统超时	200 μs （包括 EtherNet/IP 端口超时）	130 μs	300 μs	500 μs	300 μs
	基本指令执行时间 （LD）	0.016 μs		0.02 μs	0.1 μs	0.02 μs
	特殊指令执行时间 （MOV）	0.144 μs		0.18 μs	0.3 μs	0.18 μs
	立即刷新指令执行时间 （!LD）	0.99 μs	21 μs		24 μs, 28 μs	21 μs
	中断任务启动时间	30 μs	40 μs	124 μs	169 μs	124 μs
	基本 I/O 刷新 （16 点单元）	1.4 μs		3 μs		
	特殊 I/O 刷新 （8 个模拟量输入）	50 μs		120 μs	160 μs	120 μs
	CPU 总线单元刷新 （带 EtherNet/IP 单元的标签数据链接）	200 μs + （传送字数× 0.33 μs）	200 μs + （传送字数× 0.7 μs）		270 μs+ （传送字数× 0.7 μs）	200 μs + （传送字数× 0.7 μs）
	最大 EM 区容量	32K 字× 25 组	32K 字× 13 组			
	源 / 注释存储器	3.5 MB	2 MB			
	功能块定义，动作和转换	2,048	1,024			
	适用存储卡	30, 64, 128, 256, 或 512 MB	30, 64, 或 128 MB			
	循环任务数	128 任务	32 任务			
	通信端口 （内部逻辑端口）	8 + 64 端口	8 端口			
	在线编辑的循环时间增加	约 1 ms	约 8 ms		约 14 ms	约 8 ms
	扩展机架的最大数	3 机架				7 机架
	中断输入单元的最大数	2 单元				4 单元
	CPU 总线单元的最大数	15 单元	16 单元			
	内置板	不适用				适用

项目		CJ 系列				CS 系列
		CJ2 CPU 单元	CJ1-H-R CPU 单元	CJ1-H CPU 单元	CJ1-M CPU 单元	CS1-H CPU 单元
功能规格	CPU 处理模式	正常处理模式	正常或并列处理模式			
	最小循环时间	0.2 ～ 32,000 ms （以 0.1ms 为增量单位）		1 ～ 32,000 ms （以 1ms 为增量单位）		
	监控循环时间单位	0.01 ms	0.1 ms			
	设定定时中断的时间单位	0.1, 1, 或 10 ms	1 或 10 ms		0.1, 1, 或 10 ms	1 或 10 ms
	通信端口	USB, RS-232C, 和 EtherNet/IP	外设端口和 RS-232C			
	串行 PLC 链接	不支持			支持	不支持
	标签名称服务器	提供	不提供			
	外部中断槽位	槽 0 ～ 3	槽 0 ～ 4			槽 0 ～ 9
	数据跟踪	32K 字以下 , 采样时间：1 ～ 2,550 ms, 支持持续跟踪	4K 字以下 , 采样时间：10 ～ 2,550 ms			
	使用批号进行写保护	支持	不支持			
	PLC 名称	支持	不支持			
	DM 和 EM 区中的寻址位	支持	不支持			
	EM 区的强制置位 / 复位	支持自动地址分配	不支持			
	将部分 EM 区转换为跟踪存储器	支持	不支持			
	指定寻址偏移	支持	不支持			
	同一程序中以 BCD 和二进制进行定时器刷新	支持	不支持			
	一继电器网络表中的进入数（路由表之一）	128 以下		32 以下		
	时钟脉冲	0.1 ms, 1 ms, 0.01 s, 0.02 s, 0.1 s, 0.2 s, 1 s, 和 1 以上		0.02 s, 0.1 s, 0.2 s, 1 s, 和 1 以上		