

SYSMAC CJ 系列

CJ2H-CPU6□-EIP

CJ2H-CPU6□

CJ2M-CPU□□

CJ2 CPU 单元软件

操作手册

OMRON

© OMRON, 2008

版权所有。未经 OMRON 公司事先书面允许，不得将本出版物的任何部分以任何形式或任何方式（机械、电子、照相、录制或其它方式）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。由于 OMRON 公司始终致力于改进其高质量产品，所以本手册所包含的信息可随时改变而不另行通知。虽然在编制本手册时注意了一切可能的注意事项，但对于仍然可能出现的错误或遗漏，OMRON 公司不承担任何责任。同样，由于使用本手册所包含的信息而造成的损害也不承担任何责任。

SYSMAC CJ 系列
CJ2H-CPU6□-EIP
CJ2H-CPU6□
CJ2M-CPU□□
CJ2 CPU 单元软件

操作手册

2023 年 4 月修订

引言

感谢您购买 CJ 系列 CJ2H-CPU6□(-EIP)/CJ2M-CPU□□ 可编程控制器。本手册包含使用 CJ2H-CPU6□(-EIP) 或 CJ2M-CPU□□ 时所需的信息。在使用 CJ2H-CPU6□(-EIP) 或 CJ2M-CPU□□ 前，请通读并理解本手册。

目标读者

本手册主要供下列人员使用，这些人员必须具备电气系统相关知识（电气工程师或同等水平者）。

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

适用产品

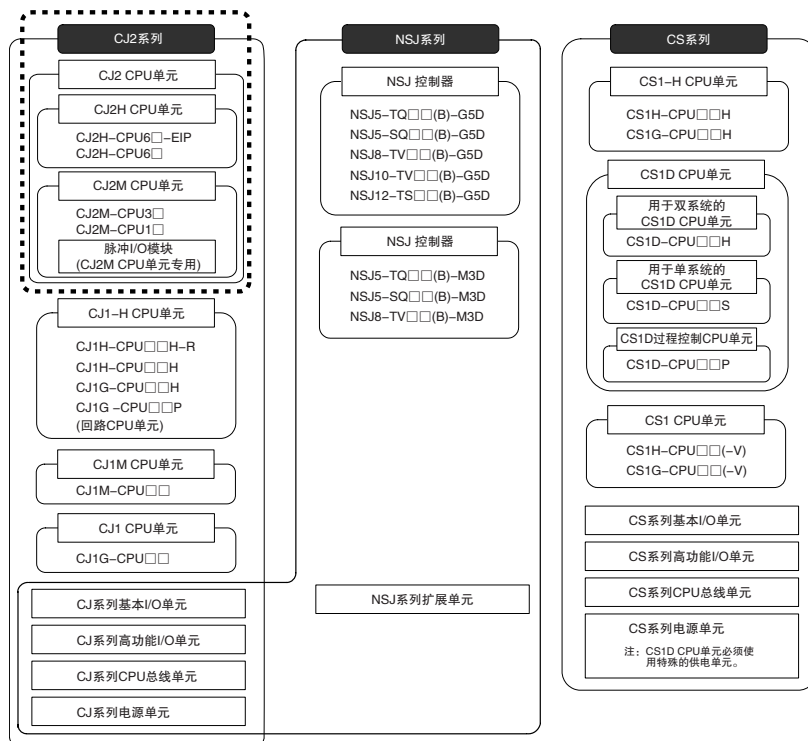
CJ 系列 CJ2 CPU 单元

- CJ2H-CPU6□-EIP
- CJ2H-CPU6□
- CJ2M-CPU3□
- CJ2M-CPU1□

本手册采用以下通用规则表示 CPU 单元。

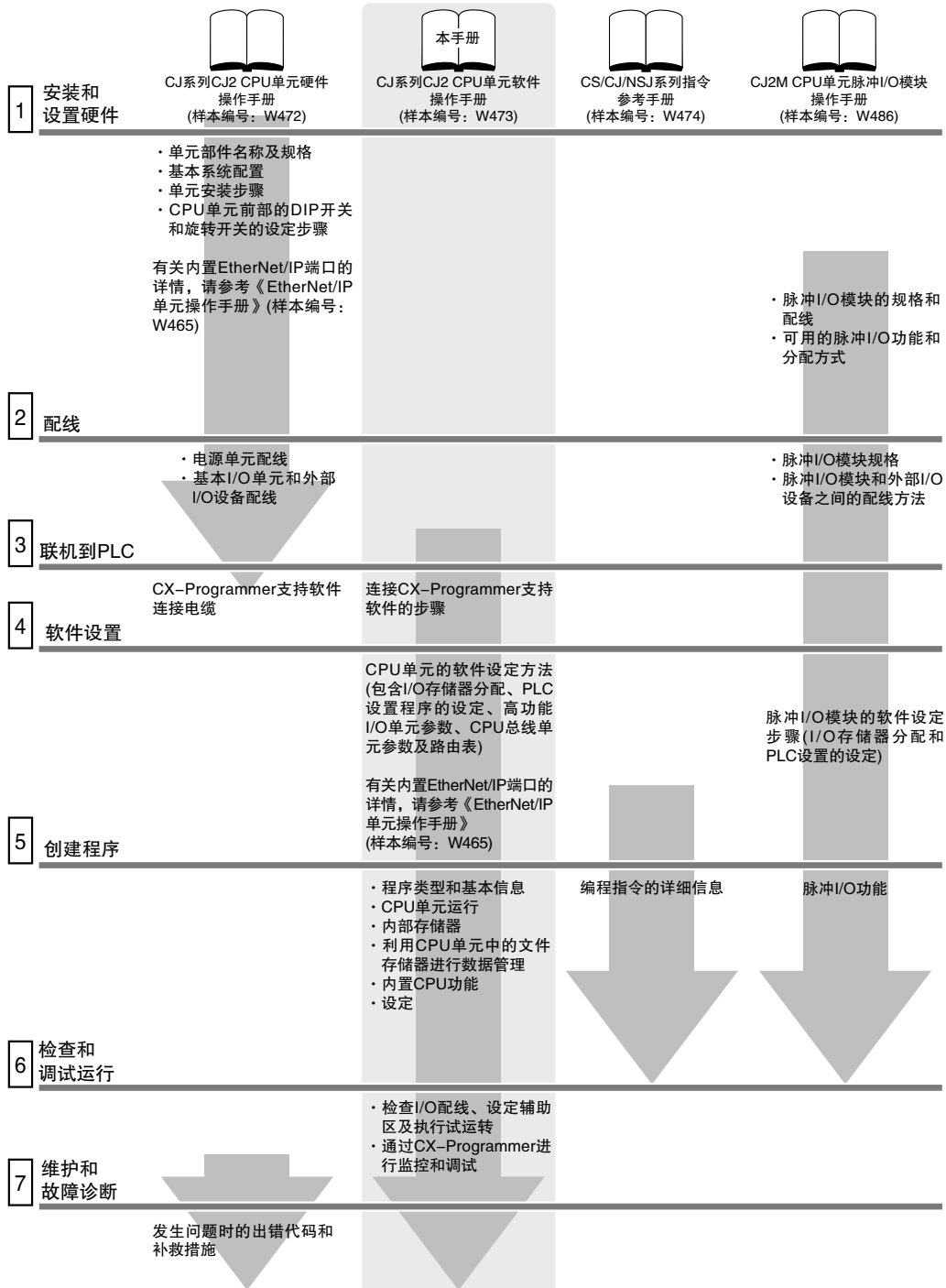
- CJ2H-CPU6□(-EIP)：表示 CJ2H-CPU6□-EIP 和 CJ2H-CPU6□CPU 单元。
- CJ2M-CPU□□：表示 CJ2M-CPU3□ 和 CJ2M-CPU1□CPU 单元。

可选脉冲 I/O 模块可与 CJ2M CPU 单元组合使用，实现脉冲输入输出功能。



CJ2 CPU 单元手册

以下手册中提供与 CJ2 CPU 单元有关的信息。请参考相应的手册以获取所需信息。



手册的构成部分

CJ2 CPU 操作手册由下表列出的章节构成。请根据需要参考相关章节。

硬件操作手册 (样本编号: W472)

章节	内容
第 1 章 概述	本章节概括介绍了 CJ2 CPU 单元, 并列出了一系列与之相关的特性和规格参数。
第 2 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 3 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和功能。
第 4 章 支持软件	本章节中阐述了用于实现编程及调试作业的各类支持软件, 以及如何连接 PLC 与支持软件。
第 5 章 安装	本章节中阐述了 CPU 单元的安装位置, 以及如何在 CPU 单元和构成单元之间进行配线。
第 6 章 故障诊断	本章节中阐述了如何在系统运行期间查看错误状态, 并针对这些错误提供了相应的纠正措施。
第 7 章 维护和检查	本章节中阐述了电池及供电单元的使用寿命及所需的定期检查, 以及如何更换电池。
第 8 章 备份操作	本章节中阐述了 PLC 数据的备份操作。
附录	附录中包含单元尺寸规格、致命 / 非致命错误详情、连接 CPU 单元串行端口时的注意事项、在计算机上安装 USB 驱动的步骤以及与负载短路保护和线路断开检测相关的信息。

软件操作手册 (样本编号: W473)(本手册)

章节	内容
第 1 章 概述	本章节概括介绍了 CJ2 CPU 单元, 并列出了一系列与之相关的特性和规格参数。
第 2 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存储的数据。
第 3 章 CPU 单元运行	本章节中阐述了 CPU 单元的内部运行原理。
第 4 章 CPU 单元的初始化	本章节中阐述了 CPU 单元的初始设定。
第 5 章 编程	本章节中阐述了程序类型以及与编程相关的细节内容 (如符号和编程指令)。
第 6 章 I/O 存储区	本章节阐述了 CPU 单元中的 I/O 存储区。
第 7 章 文件操作	本章节中阐述了 CPU 单元中存储的文件、文件的存储位置以及相关的文件操作。
第 8 章 I/O 分配与单元设定	本章节中阐述了用于在 CPU 单元和其它单元之间进行数据交换的 I/O 分配详情。
第 9 章 PLC 设置	本章节中阐述了 PLC 设置 (用于执行 CPU 单元的基本设定) 的设定详情。
第 10 章 CPU 单元的功能	本章节中阐述了 CPU 内置的各类功能。
第 11 章 编程设备与通信	本章节中阐述了 CJ2 CPU 单元与 CX-Programmer 或其它支持软件和设备进行连接的步骤。
第 12 章 CPU 单元的循环时间	本章节中阐述了如何监控和计算 CPU 单元的循环时间。
附录	附录中包含了编程指令、执行时间、步数、辅助区字和位、PLC 存储器地址 (连续的) 映射图、断电时的 I/O 存储区操作及 CJ 系列和 CS 系列 PLC 的比较等信息。

指令参考手册 (样本编号: W474)

章节	内容
第 1 章 指令的基本理解	本章节中阐述了与 CS/CJ/NSJ 系列 CPU 单元梯形图程序设计相关的基本信息。
第 2 章 指令概要	本章节概括介绍了 CS/CJ/NSJ 系列 CPU 单元使用的指令。
第 3 章 指令	本章节对 CS/CJ/NSJ 系列 CPU 单元所支持指令的功能和操作数进行了说明, 并列出了一系列的程序样例。
第 4 章 指令执行时间和步数	本章节对每条 CS/CJ/NSJ 系列 CPU 单元指令的执行时间进行了说明。
附录	附录中包含 CS/CJ/NSJ 系列 CPU 单元所有指令的名称、功能代码和助记符以及一张 ASCII 码表。

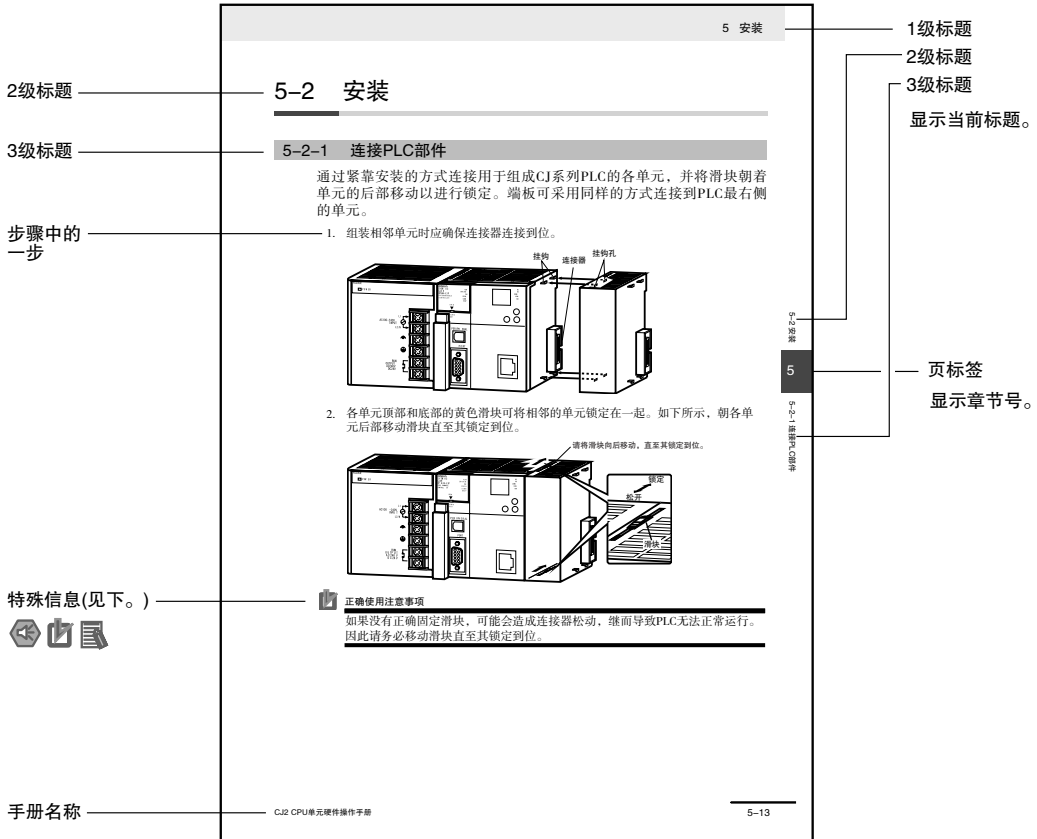
脉冲 I/O 模块操作手册 (样本编号: W486)

章节	内容
第 1 章 概述	本章节概括介绍了脉冲 I/O 模块以及 CJ2M 的脉冲 I/O 功能。
第 2 章 I/O 应用步骤和功能分配	本章节介绍了 CJ2M CPU 单元的脉冲功能及其全面应用流程和功能分配。
第 3 章 脉冲 I/O 模块的 I/O 规格和配线方法	本章节介绍了规格并描述了脉冲 I/O 模块的配线方法。
第 4 章 通用 I/O	本章节介绍了通用 I/O。
第 5 章 快速响应输入	本章节介绍了可用于输入比循环时间更短的信号的快速响应功能。
第 6 章 中断	本章节介绍了中断输入功能。
第 7 章 高速计数器	本章节介绍了高速计数器输入和高速计数器中断功能。
第 8 章 脉冲输出	本章节介绍了定位功能的相关内容, 如梯形控制、S 曲线控制、点动及原点搜索功能。
第 9 章 PWM 输出	本章节介绍了可变占空比脉冲 (PWM) 输出情况。
附录	附录中包含了脉冲输出标志变化列表、与其它型号的对照表及性能列表。

手册结构

页面结构

本手册的页面结构如下。



本图仅用作示例，本手册中可能无相关的文字描述。

特殊信息

本手册的特殊信息可分为以下几种：



安全使用注意事项

表示允许和禁止事项，以确保安全使用本产品。



正确使用注意事项

表示允许和禁止事项，以确保正常运行和性能。



附加信息

用于加深理解或简化操作。

本手册中的章节

1	概述	10	CPU单元功能	1	10
2	CPU单元内部存储器	11	编程设备与通信	2	11
3	CPU单元运行	12	CPU单元循环时间	3	12
4	CPU单元初始化	A	附录	4	A
5	编程			5	
6	I/O存储区			6	
7	文件操作			7	
8	I/O分配与单元设定			8	
9	PLC设置			9	

目录

引言	1
CJ2 CPU 单元手册	2
手册结构	5
本手册中的章节	7
安全注意事项	21
应用注意事项	25
运行环境注意事项	29
规范及标准	30
CJ2 CPU 单元的版本说明	32
相关手册	38

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 单元特性	1-4
1-2 基本操作步骤	1-12

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 存储区及存储的数据	2-3
2-1-3 将数据从编程设备传送到 CPU 单元	2-4

第 3 章 CPU 单元运行

3-1 CPU 单元的内部运行	3-2
3-1-1 概述	3-2
3-1-2 循环时间	3-4
3-1-3 电源中断时的处理	3-7
3-2 CPU 单元运行模式	3-8
3-2-1 运行模式	3-8
3-2-2 检查运行模式	3-9
3-2-3 运行模式的变更	3-10
3-2-4 运行模式详情	3-14

第 4 章 CPU 单元初始化

4-1 CPU 单元初始化概述	4-2
4-1-1 CPU 单元初始化设定	4-2
4-2 PLC 设置	4-7
4-3 创建 I/O 表	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 设定路由表	4-10
4-4-1 路由表	4-10
4-4-2 要求使用路由表的情况	4-12
4-4-3 设定和传送路由表	4-13
4-5 为高性能 I/O 单元和 CPU 总线单元设定 DM 区分配字	4-14
4-5-1 为高性能 I/O 单元和 CPU 总线单元设定 DM 区分配字	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 编程概要	5-3
5-1-2 基本梯形图概念	5-6
5-1-3 ST 语言	5-8
5-1-4 SFC 概要	5-9
5-2 任务	5-11
5-2-1 任务概要	5-11
5-2-2 循环任务	5-14
5-2-3 中断任务	5-20
5-2-4 指定任务	5-30
5-3 段	5-40
5-3-1 程序段概要	5-40
5-4 功能块	5-42
5-4-1 功能块	5-42
5-4-2 功能块的特点	5-43
5-4-3 功能块规格	5-44
5-5 符号	5-47
5-5-1 概述	5-47
5-5-2 符号的类型	5-48
5-5-3 全局符号	5-50
5-5-4 局部符号	5-50
5-5-5 网络符号 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)	5-51
5-5-6 功能块中的变量	5-55
5-5-7 符号数据类型	5-56
5-5-8 将地址自动分配到符号	5-61
5-6 指令	5-62
5-6-1 指令的基本理解	5-62
5-6-2 指定操作数	5-69
5-6-3 数据格式	5-77
5-6-4 I/O 刷新时序	5-81
5-7 变址寄存器	5-86
5-7-1 何谓变址寄存器?	5-86
5-7-2 使用变址寄存器	5-86
5-7-3 与变址寄存器相关的处理	5-91

5-7-4	监控变址寄存器.....	5-92
5-7-5	在多任务之间共享变址寄存器和数据寄存器.....	5-93
5-8	指定地址偏移.....	5-95
5-8-1	概述.....	5-95
5-8-2	地址偏移应用示例.....	5-97
5-9	检查程序.....	5-98
5-9-1	CX-Programmer 输入期间的错误.....	5-98
5-9-2	通过 CX-Programmer 检查程序.....	5-98
5-9-3	通过仿真器进行调试.....	5-99
5-9-4	程序执行检查.....	5-102
5-10	注意事项.....	5-105
5-10-1	条件标志.....	5-105
5-10-2	特殊程序段.....	5-110

第 6 章 I/O 存储区

6-1	I/O 存储区.....	6-2
6-1-1	I/O 存储区概述.....	6-2
6-1-2	I/O 存储区结构.....	6-3
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	同步数据刷新区.....	6-13
6-5	CPU 总线单元区.....	6-14
6-6	高性能 I/O 单元区.....	6-15
6-7	脉冲 I/O 区.....	6-16
6-8	串行 PLC 链接区.....	6-17
6-9	DeviceNet 区.....	6-18
6-10	工作区.....	6-19
6-11	保持区.....	6-20
6-12	辅助区.....	6-21
6-13	暂存继电器区.....	6-22
6-14	数据存储区.....	6-23
6-15	扩展数据存储区.....	6-26
6-16	定时器区.....	6-30
6-17	计数器区.....	6-32
6-18	任务标志.....	6-33
6-19	变址寄存器.....	6-34
6-20	数据寄存器.....	6-39
6-21	条件标志.....	6-41
6-22	时钟脉冲.....	6-43

第 7 章 文件操作

7-1 文件存储器	7-2
7-1-1 文件存储器类型	7-2
7-1-2 文件存储器初始化	7-3
7-1-3 存储卡注意事项	7-5
7-2 文件存储器中存储的文件类型	7-7
7-2-1 文件类型	7-7
7-2-2 文件存储器文件的创建和保存	7-10
7-3 文件存储器操作	7-11
7-3-1 文件存储器操作类型	7-11
7-3-2 文件存储器操作步骤和文件存储器文件	7-13
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	7-19

第 8 章 I/O 分配与单元设定

8-1 I/O 分配	8-2
8-1-1 I/O 分配	8-2
8-1-2 自动分配	8-4
8-1-3 手动分配	8-8
8-1-4 I/O 表错误和注意事项	8-16
8-2 设定 CPU 总线单元和高功能 I/O 单元	8-18
8-2-1 设定参数	8-18
8-2-2 数据交换	8-22

第 9 章 PLC 设置

9-1 PLC 设置概述	9-2
9-2 PLC 设置中的设定	9-4
9-3 PLC 设置中的设定	9-5
9-3-1 启动运行设定	9-5
9-3-2 CPU 单元的设定	9-8
9-3-3 时间 / 同步设定	9-14
9-3-4 高功能 I/O 单元的循环刷新	9-19
9-3-5 基本 I/O 单元的机架响应时间	9-21
9-3-6 串行端口设定	9-23
9-3-7 外设服务	9-30
9-3-8 FINS 保护	9-31
9-3-9 I/O 模块	9-32

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 存储器中存储的时间数据	10-4
10-1-3 自由运行定时器	10-6
10-2 循环时间 / 高速处理	10-7
10-2-1 最小循环时间	10-7
10-2-2 最大循环时间	10-8
10-2-3 监视循环时间	10-9
10-2-4 高速输入	10-9
10-2-5 后台执行	10-10
10-2-6 高速中断功能	10-18

10-3 启动设定和维护	10-20
10-3-1 运行模式变更和启动时的保持设定	10-20
10-3-2 掉电检测延迟设定	10-22
10-3-3 禁止电源掉电中断	10-23
10-3-4 RUN 输出	10-24
10-3-5 启动时自动传送	10-25
10-4 单元管理功能	10-32
10-4-1 基本 I/O 单元管理	10-32
10-4-2 CPU 总线单元标志 / 位	10-34
10-4-3 高功能 I/O 单元标志 / 位	10-35
10-5 存储器管理功能	10-36
10-5-1 自动备份	10-36
10-5-2 EM 文件存储器功能	10-37
10-5-3 注释存储器	10-38
10-5-4 在运行过程中替换整个程序	10-39
10-6 安全功能	10-46
10-6-1 通过 DIP 开关实现写保护	10-46
10-6-2 通过密码实现读保护	10-46
10-6-3 使用产品批号实现程序操作保护	10-51
10-6-4 针对 FINS 命令的写保护	10-52
10-6-5 PLC 名称	10-56
10-7 调试	10-59
10-7-1 强制置位 / 复位	10-59
10-7-2 测试输入	10-60
10-7-3 微分监控	10-60
10-7-4 联机编辑	10-61
10-7-5 输出置 OFF	10-62
10-7-6 数据跟踪	10-63
10-7-7 存储出错停止位	10-70
10-7-8 故障报警指令	10-71
10-7-9 模拟系统错误	10-72
10-7-10 故障点检测	10-73
10-8 单元同步运转	10-75
10-8-1 概述	10-75
10-8-2 有关单元同步运转的详细信息	10-78
10-8-3 单元同步运转规格	10-81
10-8-4 同步数据刷新	10-82
10-8-5 单元同步运转的使用限制	10-86
10-8-6 应用步骤	10-88
10-8-7 PLC 设置	10-89
10-8-8 编写同步中断任务	10-91
10-8-9 与单元同步运转相关的调整和故障诊断作业	10-92

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 自动联机	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	11-27

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	12-4
12-2-1 CPU 单元操作流程	12-4
12-2-2 循环时间概述	12-5
12-2-3 各单元的 I/O 单元刷新时间	12-7
12-2-4 循环时间计算示例	12-10
12-2-5 联机编辑引起的循环时间延长	12-12
12-2-6 I/O 响应时间	12-12
12-2-7 输入中断的响应时间	12-13
12-2-8 串行 PLC 链接的响应性能	12-14

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-7
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-18
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-29
A-1-11 逻辑指令	A-35
A-1-12 特殊算术指令	A-37
A-1-13 浮点算术运算指令	A-38
A-1-14 双精度浮点数指令	A-42
A-1-15 表格数据处理指令	A-44
A-1-16 跟踪指令	A-49
A-1-17 数据控制指令	A-50
A-1-18 子程序指令	A-53
A-1-19 中断控制指令	A-54
A-1-20 高速计数器 / 脉冲输出指令	A-56
A-1-21 步指令	A-58
A-1-22 基本 I/O 单元指令	A-58
A-1-23 串行通信指令	A-61
A-1-24 网络指令	A-62
A-1-25 文件存储指令	A-65
A-1-26 显示指令	A-66
A-1-27 时钟指令	A-67
A-1-28 调试指令	A-68
A-1-29 故障诊断指令	A-68
A-1-30 其它指令	A-69
A-1-31 块编程指令	A-70
A-1-32 文本字符串处理指令	A-74
A-1-33 任务控制指令	A-76
A-1-34 型号转换指令	A-77
A-1-35 特殊功能块指令	A-78
A-2 指令执行时间和步数	A-79
A-2-1 顺序输入指令	A-81
A-2-2 顺序输出指令	A-82
A-2-3 顺序控制指令	A-83
A-2-4 定时器和计数器指令	A-84
A-2-5 比较指令	A-85
A-2-6 数据传送指令	A-86
A-2-7 数据移位指令	A-87
A-2-8 递增 / 递减指令	A-88

A-2-9	四则运算指令	A-88
A-2-10	转换指令	A-89
A-2-11	逻辑指令	A-91
A-2-12	特殊算术指令	A-91
A-2-13	浮点算术运算指令	A-92
A-2-14	双精度浮点数指令	A-93
A-2-15	表格数据处理指令	A-94
A-2-16	跟踪指令	A-96
A-2-17	数据控制指令	A-96
A-2-18	子程序指令	A-97
A-2-19	中断控制指令	A-97
A-2-20	高速计数器 / 脉冲输出指令	A-97
A-2-21	步指令	A-99
A-2-22	基本 I/O 单元指令	A-99
A-2-23	串行通信指令	A-100
A-2-24	网络指令	A-101
A-2-25	文件存储指令	A-101
A-2-26	显示指令	A-101
A-2-27	时钟指令	A-101
A-2-28	调试指令	A-102
A-2-29	故障诊断指令	A-102
A-2-30	其它指令	A-102
A-2-31	块编程指令	A-103
A-2-32	文本字符串处理指令	A-104
A-2-33	任务控制指令	A-105
A-2-34	型号转换指令	A-105
A-2-35	特殊功能块指令	A-105
A-2-36	SFC 指令	A-106
A-2-37	功能块实例执行时间	A-106
A-3	辅助区	A-108
A-3-1	只读区 (由系统设定)	A-108
A-3-2	读 / 写区 (由用户设定)	A-138
A-3-3	辅助区操作的详细信息	A-146
A-4	PLC 存储器地址的存储器映射	A-149
A-4-1	PLC 存储器地址	A-149
A-4-2	存储器映射	A-150
A-5	断电时的操作	A-151
A-5-1	断电操作	A-151
A-5-2	断电的指令执行	A-152
A-6	在 Windows XP(SP2 或以上)、Windows Vista 或 Windows 7 中进行 EtherNet/IP 连接	A-155
A-6-1	更改 Windows 防火墙设定	A-155
A-7	PLC 比较表: CJ 系列和 CS 系列 PLC	A-157
A-8	各单元版本支持的功能	A-161
A-8-1	CJ2H CPU 单元	A-161
A-8-2	CJ2M CPU 单元	A-162
索引	索引 -1	
修订记录	修订 -1	

阅读和理解本手册

请在使用产品前仔细阅读本手册。如有任何问题或意见，请垂询 OMRON 代理商。

保证及有限责任声明

保证声明

OMRON 的排他性保证是指产品自售出起一年 (或其它指定期间) 内在材料和工艺上无缺陷。

OMRON 未以明示或暗示的方式表述或保证产品的非侵权性、适销性或特定用途的适用性。任何买方或用户均同意自主决定这些产品是否适当满足其预定用途。OMRON 对于所有其它明示或暗示的保证概不负责。

有限责任

OMRON 对于任何与产品相关的特殊、间接或直接损坏、利润损失或商业损失概不负责，不论此类索赔是基于合同、保证、疏忽还是严格责任。

不论何种情况，OMRON 对于任何诉讼行为的赔偿责任均不得超过相关产品的价格。

除非 OMRON 经分析后确认产品的使用、存放、安装和维护得当且未遭污染、滥用、误用或者不当改造或修理，否则在任何情况下，OMRON 对于与产品相关的保证、修理或其它主张不承担任何责任。

应用注意事项

适用性声明

OMRON 对于客户在其应用中的产品组合或产品使用的标准、规范或条例方面的合规性不承担任何责任。

根据客户的要求，OMRON 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。此信息本身不足以充分确定产品与终端产品、机器、系统及其它应用或用途组合的适用性。

以下为必须特别注意的应用示例，但下述内容并非包括产品所有可能的用途，也并不暗示所列用途均适用于该类产品：

- 户外用途、涉及潜在化学污染或电气干扰的用途或本手册中未说明的状况或用途。
- 核能控制系统、燃烧系统、铁道系统、航空系统、医疗设备、娱乐器械、车辆、安全设备及个别行业或政府规范监管的设备。
- 存在人身或财产安全隐患的系统、机器及设备。

请务必了解并遵守产品适用的所有禁用条款。

如果产品整体设计不足以应对此类风险，且未在整个设备或系统内针对特定用途妥善调校并安装 OMRON 产品，则不得将产品用在存在严重人身或财产隐患的场合。

可编程产品

使用可编程产品时，OMRON 不对用户的程序或其引起的后果承担任何责任。

免责声明

规格变更

基于产品改进和其它原因，产品规格及附件可能会随时变更。

公司通常在公布规格、性能或重大结构变更后更改型号。但对某些产品规格进行变更时并不另行通知。在不
确定规格时，我们会根据您的要求为您的应用场合指定特殊的型号或设立关键的规格。请随时垂询 OMRON
代理商以确认所购产品的实际规格。

尺寸与重量

尺寸和重量仅为公称值，即使已说明公差，也不得用于制造用途。

性能参数

本手册所提供的性能参数仅供用户确定适用性时使用，并不予以保证。这些数据仅表示在 OMRON 测试条件
下的结果，用户必须将其与实际应用条件相联系。实际性能遵守 OMRON 保证声明和有限责任条款的规定。

错误与疏漏

本手册所述信息经仔细审核，确信准确无误；但对于笔误、排版或校对错误或疏漏，我方概不负责。

安全注意事项

安全注意信息的定义

本手册使用下列标识，以提供 CJ 系列 PLC 安全使用所需的注意事项。所述安全注意事项至关重要，因此，请务必仔细阅读并遵守所有安全注意事项的相关信息。



警告

表示潜在的危險状况，如不加以避免，将会造成死亡或严重伤害。此外，还可能会造成严重的财产损失。



注意

表示潜在的危險状况，如不加以避免，可能会造成轻度或中度伤害或财产损失。



安全使用注意事项

表示允许和禁止事项，以确保安全使用本产品。



正确使用注意事项

表示允许和禁止事项，以确保正常运行和性能。

符号



三角形符号表示注意事项 (包括警告)，具体内容显示在三角形内并附带文字说明。此示例表示小心触电。



圆圈加斜杠符号表示禁止操作，具体内容显示在圆圈内并附带文字说明。



实心圆圈符号表示强制性操作，具体内容显示在圆圈内并附带文字说明。此示例表示一般的强制性事项。



三角形符号表示注意事项 (包括警告)，具体内容显示在三角形内并附带文字说明。此示例表示一般注意事项。



三角形符号表示注意事项 (包括警告)，具体内容显示在三角形内并附带文字说明。此示例表示小心表面高温。

警告

请勿在通电状态下试图拆卸任何单元或接触单元内部，否则可能会导致触电。



请勿在通电状态下接触任何端子或端子台，否则可能会导致触电。



为了在因可编程序控制器误动作或其它影响可编程序控制器操作的外部因素引起异常时确保系统安全，应在外部电路中（并非在可编程序控制器内部）设置以下安全措施，“可编程序控制器”表示 CPU 单元和其它所有单元，并在本手册中缩写为“PLC”。否则可能会导致严重事故。



- 在自诊断功能检测到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。但是，I/O 控制部分和 I/O 存储器中的错误及其它自诊断功能无法检测的错误仍然会引发意外动作。为应对上述错误，必须设置外部安全措施以确保系统安全。
- 若输出继电器卡死、烧毁或输出晶体管毁损，PLC 输出可能会保持在 ON 或 OFF 状态。为应对上述问题，必须设置外部安全措施以确保系统安全。
- 在计算机系统和编程设备中采取相应的安全措施，即使在数据链接通信或远程 I/O 通信过程中发生通信错误或误动作，也能确保整体系统的安全。

在确认安全后方可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)。否则连接到输出单元的设备可能会发生误动作，而无视 CPU 单元的运行模式。



用户必须采取相应的故障安全措施。这样，即使在因信号线路损坏、瞬时断电或其它原因导致信号错误、丢失或异常的情况下，也能确保安全。若不采取适当的措施，则可能会因操作不当而导致严重事故。



警告

防病毒保护

请在连接控制系统的电脑上安装最新版本的企业级杀毒软件并及时维护。



防止非法访问

请对本公司产品采取下列防范非法访问的措施。

- 导入物理控制，确保只有授权人员才能访问控制系统及设备
- 通过将控制系统及设备的网络连接限制在最低程度，防止未获信任的设备访问
- 通过部署防火墙，将控制系统及设备的网络与 IT 网络隔离（断开未使用的通信端口、限制通信主机）
- 如需远程访问控制系统及设备，应使用虚拟专用网络（VPN）
- 在控制系统及设备的远程访问中导入多重要素认证
- 采用复杂密码并频繁更换
- 如需在控制系统或设备上使用 USB 存储器等外部存储设备，应事先进行病毒扫描



数据输入输出保护

请确认备份、范围检查等妥当性，以防对控制系统和设备的输入输出数据受到意外修改。

- 检查数据范围
- 利用备份确认妥当性，完善还原准备，以防数据遭到篡改或发生异常
- 进行安全设计如紧急停机、应急运行等，以应对数据遭到篡改及异常情况



丢失数据的复原

请定期进行设定数据的备份和维护，以防数据丢失。



经由全局地址使用内部网络时，一旦连接至 SCADA、HMI 等未经授权的终端或未经授权的服务器，可能会面临恶意伪装、数据篡改等网络安全问题。

请客户自行采取充分有效的安全防护措施以防范网络攻击，例如限制终端访问，使用配备安全功能的终端，对面板设置区域实施上锁管理等。



构建内部网络时，可能会因电缆断线、未经授权的网络设备的影响，导致通信故障的发生。

请采取充分有效的安全防护措施，例如通过对面板设置区域实施上锁管理等方法，限制无权限人员对网络设备的物理访问。



使用配备 SD 存储卡功能的设备时，可能存在第三方通过拔出或非法卸载移动存储介质等方式非法获取、篡改、替换移动存储介质内的文件及数据的安全风险。

请客户自行采取充分有效的安全防护措施，包括但不限于对面板设置区域实施上锁管理、门禁管理等方式，以限制无权限人员对控制器的物理访问，对移动存储介质采取妥善的管理措施等等。



⚠ 注意

请务必在确认延长循环时间不会引起不良影响后，再执行联机编辑。否则，可能无法读取输入信号。



对目的结点进行安全确认后，方可向其它结点传送程序、PLC 设置、I/O 表、I/O 存储器数据或参数，或者修改这些项目中的内容。传送或修改数据可能会导致意外动作。



当将用户程序和参数数据写入 CPU 单元时，CJ2 CPU 单元会自动将其备份至闪存。但是 I/O 存储区（包括 DM、EM 和保持区）中的数据将不会写入闪存。

在断电状态下，可通过电池保持 DM、EM 和保持区中的数据。若存在电池故障，则无法保证这些区域中的数据在断电后的准确性。如果 DM、EM 和保持区中的内容用于控制外部输出，则无论电池出错标志位 (A402.04) 是否置 ON，都应防止输出不当。



请按照操作手册中规定的扭矩值拧紧 AC 电源单元上的端子螺钉。螺钉松动可能会导致电源单元烧毁或故障。



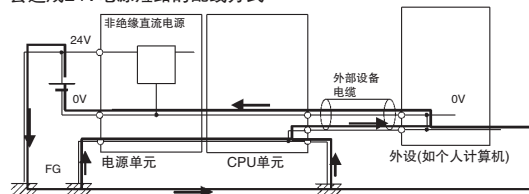
请勿在通电状态下或刚关闭电源时触摸电源单元。此时电源单元温度很高，会导致灼伤。



当个人计算机或其它外部设备连接至安装了非隔离型电源单元 (CJ1W-PD022) 的 PLC 时，请将外部电源的 0V 侧接地，也可不将外部电源接地。如果使用了错误的接地方法，外部电源将会发生短路。切勿将 24V 侧接地，具体原因如下图所示。



会造成 24V 电源短路的配线方式



应用注意事项

使用 CJ 系列 PLC 时，应遵循以下注意事项。

● 电源

- 请始终使用操作手册中规定的电源电压。电源电压错误会导致误动作或设备烧毁。
- 若超出电源单元的容量，可能会导致 CPU 单元或其它单元无法启动。
- 采取适当措施，确保提供额定电压和频率的电源。请特别注意供电不稳定的地方，供电错误会导致误动作。
- 着手进行以下任何事项前，请务必关闭 PLC 的电源。否则，可能会导致误动作或触电。
 - 安装或拆卸电源单元、I/O 单元、CPU 单元、选件板、脉冲 I/O 模块或其它任何单元；
 - 组装单元；
 - 设定 DIP 开关或旋转开关；
 - 连接电缆或对系统配线；
 - 连接或断开连接器；
- 若在端子间进行交叉配线，则电线中的电流将为所有端子的总电流。请确保电线具有足够的电流容量。
- 使用支持“更换通知”功能的电源单元时，请务必遵循以下注意事项。
 - 如果电源单元前端的显示器开始交替显示 0.0 和 A02 或者报警输出自动关闭时，请在六个月内更换电源单元。
 - 确保使报警输出电缆远离电源线路及高压线路。
 - 在报警输出处施加的电压或连接的负载不得超过额定规格。
 - 若电源单元闲置三个月以上的时间，则应将其保存在 $-20 \sim 30 \text{ }^{\circ}\text{C}$ /25% ~ 70%RH 的环境中，以保持其“更换通知”功能。
 - 如果电源单元安装不当，则会使热量累积，从而可能会导致更换通知信号在错误的时间出现或造成内部元件老化或损坏。请仅使用标准安装方法。
- 请勿在电源刚关闭时触碰电源单元上的端子。端子上的残留电压会导致触电。
- 连接计算机到 PLC 时请注意下列注意事项，以防止因电位差引发的故障。
 - 连接台式计算机与 PLC 之前，请从 AC 插座上断开计算机的电源插头。AC 适配器的中的残余电流可能会在计算机和 PLC 之间产生一个电位差。连接完成后，方可再次通过 AC 适配器供电。
 - 如果计算机带有 FG 端子，应将其连至 PLC 上的 FG(GR) 端子，以确保它们的电位相同。
- 如果将计算机分开接地，则会因接地条件的不同产生电位差。

● 安装

- 请勿将 PLC 安装在高频噪声干扰源附近。
- 触碰单元前，请务必先触摸接地金属体以消除静电累积。否则，可能会导致误动作或设备损坏。
- 确认端子台、连接器、存储卡、选件板、脉冲 I/O 单元、扩展电缆和其它带锁扣装置的部件均锁合到位。
- 连接邻近单元后，电源单元、CPU 单元、I/O 单元、高功能 I/O 单元、CPU 总线单元及脉冲 I/O 模块顶部或底部的滑块必须完全锁定（即卡入到位）。如果滑块未锁到位，单元可能会无法正常运转。如果滑轨没有锁定，可能无法实现正确的功能。

● 配线

- 请遵循本手册中的说明以正确执行配线作业。
- 在接通电源前，应仔细检查所有的配线及开关设定。配线错误可能会导致设备烧毁。
- 请务必按照相关手册规定的扭矩值紧固所有的端子螺钉和电缆连接器螺钉。若施加的紧固扭矩不当，可能会导致误动作。
- 对安装位置进行彻底检查后，方可安装端子台和连接器。
- 配线时应将标签保留在单元上。若撕去标签，可能会因异物落入单元导致误动作。
- 为保证散热正常，请在配线完成后撕去标签。保留标签可能会导致误动作。
- 请使用压接端子进行配线。请勿用裸绞合线直接连接端子。否则可能会导致设备烧毁。
- 施加在输入单元上的电压不得超过额定输入电压值，否则可能会导致设备烧毁。
- 安装单元时，请务必连接一个接地电阻小于 100Ω 的接地体。否则可能会导致触电。
短接电源单元上的 GR 和 LG 端子时，同样必须安装一个接地电阻小于 100 Ω 的接地体。
- 请勿将超出最大开关容量的电压或负载施加到输出单元。过电压或过载可能会导致设备烧毁。
- 请勿过度拽拉或弯曲电缆。上述动作均可能导致电缆断裂。
- 请勿在电缆或其它配线上放置物品，否则可能会导致电缆断裂。
- 请勿使用市售的 RS-232C 个人计算机电缆。应始终使用本手册列出的专用电缆或按照手册中的规格制作电缆。使用市售电缆可能会损坏外部设备或 CPU 单元。
- 切勿将 CPU 单元 RS-232C 端口上的引脚 6(5V 电源) 连接到除 NT-AL001 链接适配器、CJ1W-CIF11 转换器及可编程终端 (NV3W-M□20L) 以外的任何设备，否则会导致外部设备或 CPU 单元损坏。

● 处理

- 若在绝缘强度试验中施加了最大电压或使用开关突然关闭电源，可能会导致电源单元损坏。请使用可变电阻器逐渐调高或调低电压。
- 在执行耐压测试或绝缘电阻测试前，应将电源单元上的线路接地端子 (LG) 和功能接地端子 (GR) 分开，否则可能会导致设备烧毁。
- 开始操作前，应确保 DIP 开关和 DM 区的设定正确无误。
- 更换 CPU 单元、高功能 I/O 单元或 CPU 总线单元后，应确保 DM 区、保持区和其它存储区需要的数据在恢复操作前已传送到 CPU 单元。
- 在进行以下任何一项操作前，请确认不会对设备造成不良影响，否则可能会导致意外操作。
 - 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意位。
 - 修改存储器中任何字或设置值的当前值。
- 切勿试图拆解、修理或改造任何单元，否则可能导致误动作、起火或触电。
- 请勿使 PLC 坠落或使其遭受过度振动或冲击。
- 若 PLC 在没有安装电池且没有接通电源的情况下闲置了较长时间，然后仍在不通电的情况下安装电池，将缩短电池的使用寿命。
- 请在电池出现故障时或规定的电池后备时间过后立即更换电池。务必安装一枚标示生产日期不超过两年的备用电池。
- 更换电池前，应在更换步骤开始的至少 5 分钟之前接通电源，并在电源关闭后的 5 分钟内完成更换作业。若未遵守该注意事项，存储器中的内容可能会遭到破坏。
- 如果在对应用程序编程时使用电池出错标志，则在更换电池前，即使在电源接通的情况下系统检测到电池错误，也应确认系统安全。
- 请勿短接电池端子或对电池进行充电、拆解、加热或焚烧。请勿使电池受到猛烈的冲击。上述操作会导致电池漏液、破裂、发热或起火。若电池掉落到地板上或受到过度冲击，请丢弃该电池。受过冲击的电池在使用过程中可能会发生漏电。
- UL 标准规定电池的更换作业只能由资深的技术人员完成。因此请确保指派资深的工程师完成电池的更换工作。请根据本手册给出的步骤更换电池。

- 请按照当地的法规要求对产品和电池进行废置处理。



廢電池請回收

- 若 I/O 保持位置 ON，则当从 RUN 或 MONITOR 模式切换到 PROGRAM 模式时，PLC 的输出不会置 OFF，并将保持其原有状态。请确保外部负载不会在上述过程中构成危险因素。(当因致命错误(包括 FALS(007) 指令产生的错误)导致停止运转时，将对输出单元的所有输出置 OFF，而只保持内部输出状态。)
- 若数据链接表或参数设定不当，可能会导致意外动作。即使数据链接表或参数设定正确，也应在启动或停止数据链接之前确认受控系统不会受到负面影响。
- 编写程序时，应确保数据链接通信过程中收到的任何数据仅在 CPU 单元(数据源)不存在任何错误的情况下使用。根据 CPU 单元状态标志位中的出错信息来查找数据源 CPU 单元中的错误。若数据源 CPU 单元中存在错误，则可能会发送错误的数据库。
- 当路由表从编程设备传送到 CPU 单元时，所有的 CPU 总线单元将重新启动，以读取和启用新路由表。传送路由表前，应确认系统不会受到负面影响。
- 在 PLC 运转过程中传送标签数据链接参数时，相关结点间的标签数据链接将会断开。传送标签数据链接参数前，应确认系统不会受到负面影响。
- 若与网络通信存在冲突，输出状态将取决于使用的设备。使用带输出的设备时，应确认那些动作会与网络通信发生冲突，并在需要时执行相关的安全措施。
- 当通过编程设备(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，应将第一个写地址设为 D20000，并确保所写数据的大小不超过 DM 区的大小。若在启动时从存储卡读取数据文件，即使在创建 AUTOEXEC.IOM 文件的过程中设定了其它地址，数据也将从 D20000 开始写入 CPU 单元。同样，当数据大小超出 DM 区大小时(使用 CX-Programmer 时会出现这种情况)，超出的部分将被写入 EM 区。
- CJ2 CPU 单元中的用户程序和参数区数据备份在内置闪存中。备份操作执行过程中，CPU 单元正面的 BKUP 指示灯将会亮起。此时请勿切断 CPU 单元的电源，否则将无法完成数据备份。
- 实际运行前，应对用户程序和单元参数设定进行检查以确保其正确执行。若未进行检查，则可能会导致意外动作。
- 在 I/O 表中设定高功能 I/O 单元或 CPU 总线单元时，应在重新启动单元前仔细检查连接目标处的设备安全性。
- 当读取或写入存储卡时，请勿切断 PLC 的电源。此外，请勿在 BUSY 指示灯亮起的情况下取出存储卡，否则会造成存储卡无法使用。
若要取出存储卡，应按存储卡电源的开关，直至 BUSY 指示灯熄灭后方可取出存储卡。
- 恢复数据时，应在执行恢复操作前确认选定的数据正确无误。若选定数据的内容错误，控制系统可能会在数据恢复后出现意外动作。
- 部分高功能 I/O 单元和 CPU 总线单元是根据 CPU 单元中存储的参数(即分配至 DM 区的字、数据链接表或 Ethernet 设定)运行的。若选用的 CPU 总线单元或高功能 I/O 单元存在任何限制条件，PLC 备份工具的信息区中将会显示与这些限制条件相关的信息。查看限制条件后，请务必在备份或恢复数据时选择 CPU 单元和 CPU 总线单元或高功能 I/O 单元。若在未选中两台单元的情况下开始对设备进行数据备份或恢复，则控制系统可能会出现意外动作。
- 若用于存储数据的单元在备份操作方面存在限制条件，PLC 备份工具的信息区中将会显示与这些限制条件相关的信息。请查看这些信息并采取必要的应对措施，否则在数据恢复后操作设备会使控制系统出现意外动作。
- 若要在 PLC 操作过程中恢复数据，应先确认 PLC 停止操作后不存在任何问题。如果 PLC 意外停止，控制系统可能会出现意外动作。
- 请务必切断 PLC 的电源，并在恢复数据后重新接通电源。若没有重新启动电源，控制系统中的数据将不会更新为恢复后的数据，并有可能发生意外动作。
- 对于包含强制状态的数据，只可进行备份操作，而无法进行恢复。恢复包含强制状态的数据后，应按需要在操作前通过 CX-Programmer 执行强制置位或强制复位。根据强制状态的不同，控制系统可能会出现意外动作。
- 在梯形图或 ST 语言编程中，若将符号或存储器地址(ST 语言编程)指定为数组变量的后缀，请确保指定的元素数量没有超过最大存储配额。
若指定的元素数量超过了为符号指定的最大存储配额，系统会从另一个存储区中读取数据，从而导致意外动作。

- 若将符号或地址指定为梯形图中的偏移量，应在编程时确保使用字或符号间接指定的偏移量没有超出存储区的起始地址。
若间接指定值超出了存储区的起始地址，系统会从另一个存储区中读取数据，从而导致意外动作。

● 外部电路

- 请务必在接通 PLC 的电源后再接通控制系统的电源。若先接通控制系统电源后再接通 PLC 电源，则在接通 PLC 电源时，DC 输出单元和其它单元上的输出端子上的状态会瞬间变为 ON，从而导致控制系统信号临时出错。
- 为防止外部配线短路，请安装外部断路器和采取其它安全措施。防短路安全措施不充分可能会导致设备烧毁。

运行环境注意事项

- 请遵循本手册中的说明以正确执行安装作业。

- 请勿在下列场所运行控制系统：
 - 阳光直射处；
 - 温度或湿度超出规格中规定范围的场所；
 - 由于温度急剧变化易造成结露现象的场所；
 - 存在腐蚀性气体或易燃性气体的场所；
 - 存在粉尘 (尤其是铁屑) 或盐雾的场所；
 - 暴露于水、油类或化学品的场所；
 - 易受冲击或振动的场所。

- 将系统安装在下列场所时，应采取适当和有效的预防措施：
 - 存在静电或其它形式噪声的场所；
 - 存在强电磁场的场所；
 - 可能暴露于放射性污染的场所；
 - 靠近动力电源的场所。

规范及标准

符合 EC 指令

适用指令

- EMC 指令
- 低电压指令

概念

● EMC 指令

OMRON 公司的所有设备均符合 EC 指令和 EMC 标准, 因此它们更容易与其它设备或整机进行集成。为符合 EMC 标准, 对各实际产品都作了检验(参见下注)。但是, 对于这些产品是否符合客户所用系统的标准, 必须由客户进行确认。

符合 EC 指令的 OMRON 设备的相关 EMC 性能因配置、配线、设备的其它条件或 OMRON 设备安装的控制柜而异。

因此, 为确认设备和整机是否符合 EMC 标准, 客户必须进行最终检验。

* 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(辐射发射特性: 10m 调整率)

● 低电压指令

始终确保设备工作在 50 ~ 1,000VAC/75 ~ 1,500VDC 的电压范围内, 满足 PLC 所要求的安全标准(EN61131-2)。

● 符合 EC 指令

CJ 系列 PLC 符合 EC 指令。为确保使用 CJ 系列 PLC 的机器或设备符合 EC 指令, 必须遵照以下指示安装 PLC:

- CJ 系列 PLC 必须安装在控制柜内。
- 连接至 DC 电源单元和 I/O 单元的 DC 电源必须进行加强绝缘或双重绝缘。
- CJ 系列 PLC 符合 EC 指令和通用辐射标准(EN 61000-6-4)。辐射发射特性(10m 调整率)因控制柜的配置、连至控制柜的其它设备、配线和其它条件而异。因此, 必须确认整机或设备是否符合 EC 指令。

符合造船标准

本产品符合下列造船标准。对造船标准的适用性取决于特定的使用条件。本产品在部分场合可能无法使用。因此，请在船舶上使用 PLC 之前联系当地的 OMRON 代理商。

NK 和 LR 造船标准的使用条件

- 船桥及甲板以外的使用条件
 - PLC 必须安装在控制柜内。
 - 控制柜开闭口的间隙必须使用垫衬或其它材料完全填充或覆盖。
- 船桥及甲板的使用条件 (仅由 NK 认证)
 - PLC 必须安装在控制柜内。
 - 控制柜开闭口的间隙必须使用垫衬或其它材料完全填充或覆盖。
 - 必须将下列噪声滤波器连接至电源线路中。

噪声滤波器

制造商	Cosel Co., Ltd.
型号	TAH-06-683

商标

SYSMAC 为欧姆龙株式会社开发的可编程序控制器的注册商标。

CX-One 为欧姆龙株式会社开发的编程软件的注册商标。

Windows 是美国微软公司的注册商标。

本手册引用的其它系统名称和产品名称均是其各自持有者的商标或注册商标。

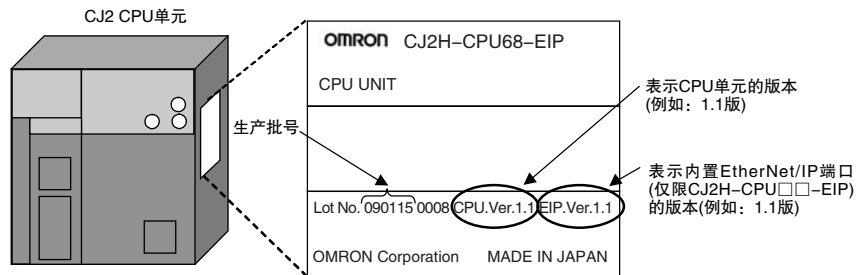
CJ2 CPU 单元的版本说明

单元版本

OMRON 制定了一套“单元版本”规则，可根据版本升级前后的功能差异对 CJ2 CPU 单元进行管理。

产品上的单元版本标记法

如下图所示，单元版本标记在产品铭牌上的批号右侧，以便管理这些产品的单元版本。



通过支持软件确认单元版本

使用 CX-Programmer 确认单元版本 (下述任意一种方法均可)。

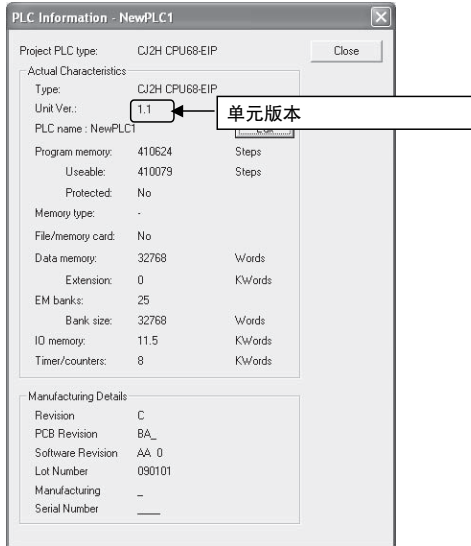
- 使用“PLC Information” (PLC 信息) 对话框
- 使用“Unit Manufacturing Information” (单元生产信息) 对话框 (该方法亦可用于高性能 I/O 单元和 CPU 总线单元。)

● “PLC Information” (PLC 信息) 对话框

1. 采用下列方法中的任意一种来显示“PLC Information” (PLC 信息) 窗口。

- 在已知设备类型和 CPU 类型的情况下，可在“Change PLC” (变更 PLC) 对话框中选择这些类型并连线，然后依次点击菜单中的“PLC” - “Edit” (编辑) - “Information” (信息)。
- 在设备类型和 CPU 类型未知且通过串行线路与 CPU 单元直连的情况下，可先点击菜单中的“PLC” - “Auto Online” (自动连线) 进行连线，然后选择菜单中的“PLC” - “Edit” (编辑) - “Information” (信息)。

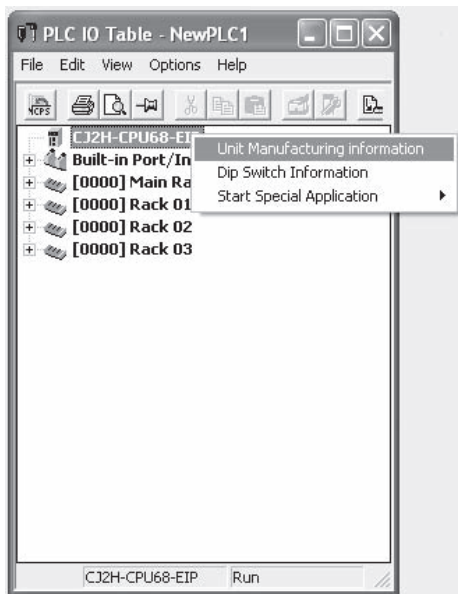
2. 无论采用哪种方法，下页中的“PLC Information” (PLC 信息) 窗口均会显示。



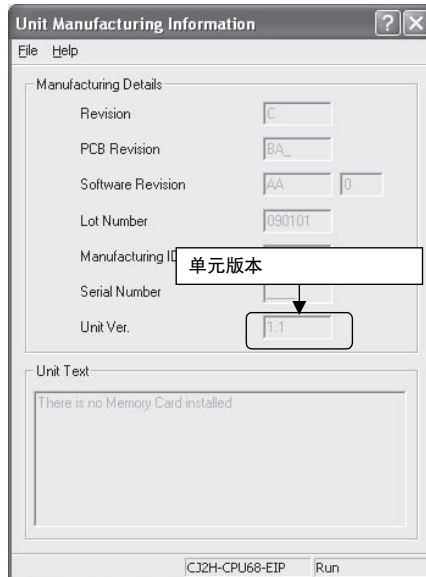
通过上述对话框确认 CPU 单元的版本。

- “Unit Manufacturing Information” (单元生产信息) 对话框

1. 在“IO Table” (IO 列表) 窗口中右击选择“Unit Manufacturing information – CPU Unit” (单元生产信息 – CPU 单元)。



2. 此时将显示以下 “Unit Manufacturing information” (单元生产信息) 窗口。



通过上述对话框确认已连线的 CPU 单元的版本。

3. 通过单元版本标签确认单元版本
CPU 单元附带下列单元版本标签。



可将这些标签贴在先前 CPU 单元的前面，用于区分不同版本的 CPU 单元。

单元版本

项目	型号	单元版本
CJ2H CPU 单元	CJ2H-CPU6□-EIP	1.0 版 (内置 EtherNet/IP 端口: 2.0 版)
		1.1 版 (内置 EtherNet/IP 端口: 2.0 版)
		1.2 版 (内置 EtherNet/IP 端口: 2.0 版)
		1.3 版 (内置 EtherNet/IP 端口: 2.0 版)
	CJ2H-CPU6□	1.1 版
		1.2 版
		1.3 版
CJ2M CPU 单元	CJ2M-CPU3□	1.0 版 (内置 EtherNet/IP 端口: 2.0 版)
		2.0 版 (内置 EtherNet/IP 端口: 2.0 版)
		2.0 版 (内置 EtherNet/IP 端口: 2.1 版)
	CJ2M-CPU1□	1.0 版 2.0 版

单元版本和编程设备

单元版本和 CX-Programmer 版本之间的关系如下表所示。有关各单元版本支持的功能，请参考“A-8 各单元版本支持的功能”。

● 单元版本和编程设备

CPU 单元		功能		所需的编程设备						编程器
				CX-Programmer						
				7.1 版或更早版本	8.0 版	8.2 版或更高版本	9.0 版	9.1 版	9.12 版或更高版本	
CJ2H	CJ2H-CPU6□-EIP 1.0 版	1.0 版的功能		---	允许	允许	允许	允许	允许	---*3
	CJ2H-CPU6□-EIP 1.1 版	1.1 版新增的功能	使用新功能	---	---	允许*2	允许	允许	允许	
			不使用新功能	---	允许*1	允许	允许	允许	允许	
	CJ2H-CPU6□ 1.1 版	1.1 版新增的功能	使用新功能	---	---	允许*2	允许	允许	允许	
			不使用新功能	---	---	允许	允许	允许	允许	
	CJ2H-CPU6□-EIP 1.2 版	1.2 版新增的功能	使用新功能	---	---	---	允许	允许	允许	
			不使用新功能	---	允许*1	允许*1	允许	允许	允许	
	CJ2H-CPU6□ 1.2 版	1.2 版新增的功能	使用新功能	---	---	---	允许	允许	允许	
			不使用新功能	---	允许*1	允许*1	允许	允许	允许	
	CJ2H-CPU6□-EIP 1.3 版	1.3 版新增的功能	使用新功能	---	---	---	允许	允许	允许	
			不使用新功能	---	允许*1	允许*1	允许	允许	允许	
	CJ2H-CPU6□ 1.3 版	1.3 版新增的功能	使用新功能	---	---	---	允许	允许	允许	
			不使用新功能	---	允许*1	允许*1	允许	允许	允许	
	CJ2M	CJ2M-CPU□□1.0 版	1.0 版的功能		---	---	---	---	允许	
CJ2M-CPU□□2.0 版		2.0 版新增的功能	使用新功能	---	---	---	---	---	允许	
	不使用新功能		---	---	---	---	允许*1	允许		

*1 若不使用针对升级版 CPU 单元强化的功能，则无需升级 CX-Programmer 的版本。

*2 若要使用 1.1 版 CPU 单元新增的功能，则需配备 CX-Programmer 8.2 版或更高版本。此外，CX-Programmer 8.02 版也支持高速中断功能，并可在 MONITOR 模式下修改最小循环时间设定。



*3 CJ2 CPU 单元无法使用手持式编程器。

● 关于 PLC 型号的下拉式列表

CX-Programmer 的“Change PLC”(变更 PLC)对话框中的下拉式列表并未对单元版本进行细化区分。请按下表进行选择，无需考虑系统版本。

系列	CPU 单元	型号	CX-Programmer 9.0 版或更高版本的“Change PLC”(变更 PLC)对话框中列出的 PLC 型号
CJ 系列	CJ2H CPU 单元	CJ2H-CPU6□-EIP CJ2H-CPU6□	CJ2H
	CJ2M CPU 单元	CJ2M-CPU3□ CJ2M-CPU1□	CJ2M

对 CX-Programmer 上的单元版本问题进行故障诊断

问题	原因	措施
 <p>显示上述信息后，输出窗口中的“Compile”（编译）选项页上会出现一行编译错误信息。</p>	<p>试图将包含仅受高版本 CPU 单元支持的指令的程序下载至低版本的 CPU 单元中。</p>	<p>检查程序内容或者改用高版本的 CPU 单元。</p>
	<p>试图将包含仅受高版本 CPU 单元支持的设定的 PLC 设置下载至低版本的 CPU 单元中。</p>	<p>检查 PLC 设置中的设定或改用高版本的 CPU 单元。</p>
<p>在从 PLC 传送至 CX-Programmer 的程序中显示“????”信息。</p>	<p>试图将包含仅受高版本 CX-Programmer 支持的指令的程序上传至低版本的 CX-Programmer 中。</p>	<p>无法将新指令上传至低版本的 CX-Programmer 中。请使用高版本的 CX-Programmer。</p>

相关手册

下表所示为集成 CJ 系列 CJ2 CPU 单元的 PLC 产品的相关手册，请使用这些手册作为参考。

手册	样本编号	型号	应用	描述
CJ 系列 CJ2 CPU 单元 软件操作手册（本手册）	W473	CJ2H-CPU6□-EIP CJ2H-CPU6□ CJ2M-CPU□□	CJ2 CPU 单元的软件规格	本手册对 CJ2 CPU 单元进行了以下几个方面的描述： · CPU 单元运行 · 内部存储器 · 编程 · 设定 · CPU 单元的内置功能 请同时参考《硬件操作手册》(样本编号: W472)。
CJ 系列 CJ2 CPU 单元 硬件操作手册	W472	CJ2H-CPU6□-EIP CJ2H-CPU6□ CJ2M-CPU□□	CJ2 CPU 单元的硬件规格	本手册对 CJ2 CPU 单元进行了以下几个方面的描述： · 概述和功能 · 基本系统配置 · 各部分名称及功能 · 安装及设定步骤 · 故障纠正措施 请同时参考《软件操作手册》(样本编号: W473)。
CS/CJ 系列 EtherNet/IP 单元操作手册	W465	CJ2H-CPU6□-EIP CJ2M-CPU3□ CS1W-EIP21 CJ1W-EIP21	使用 CJ2 CPU 单元的内置 EtherNet/IP 端口	对 EtherNet/IP 端口和 EtherNet/IP 单元进行了说明。 对基本设定、标签数据链接、FINS 通信及其它功能进行了说明。
CJ 系列 CJ2M CPU 单元 脉冲 I/O 模块操作 手册	W486	CJ2M-CPU□□ + CJ2M-MD21□	有关在 CJ2M CPU 单元上使用 脉冲 I/O 的信息	对 CJ2M CPU 单元进行了以下几个方面的描述： · 规格和配线方法 · I/O 功能 · 快速响应输入 · 中断功能 · 高速计数器 · 脉冲输出 · PWM 输出 编程时，请将该手册与《指令参考手册》(样本编号: W474) 结合使用。
CS/CJ/NSJ 系列指令参 考手册	W474	CJ2H-CPU6□-EIP CJ2H-CPU6□ CJ2M-CPU□□ CS1G/H-CPU□□H CS1G/H-CPU□□-V1 CJ1G/H-CPU□□H CJ1G-CPU□□ CJ1M-CPU□□ NSJ□-□□□□(B)-G5D NSJ□-□□□□(B)-M3D	有关指令的信息	本手册对各程序指令进行了详细说明。 编程时请参考《软件操作手册》(样本编号: W473)。

手册	样本编号	型号	应用	描述
CS/CJ/CP/NSJ 系列通信命令参考手册	W342	CJ2H-CPU6□-EIP CJ2H-CPU6□ CJ2M-CPU□□ 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□□□□-□ CP1L-M/L□□□□-□□ CP1E-E/N□□□□D□-□ NSJ□-□□□□(B)-G5D NSJ□-□□□□(B)-M3D	有关 CS/CJ/CP 系列 CPU 单元与 NSJ 系列控制器进行通信的信息	本手册对 C 模式和 FINS 命令进行了说明。 有关通过 C 模式或 FINS 命令与 CPU 单元进行通信的详细说明, 请参考本手册。 注 本手册中对寻址 CPU 单元的通信指令进行了说明。这与所使用的通信路径无关, 以下端口均可作为通信路径: CPU 单元的串行端口、串行通信单元 / 选件板及通信单元的通信端口。对于寻址高性能 I/O 单元或 CPU 总线单元的通信指令, 请参考相关单元的操作手册。
CX-One 设置手册	W463	CXONE-AL□□C-V□/ AL□□D-V□	CX-One 的软件安装	本手册概括介绍了 CX-One FA 整合工具包并列出了相关的安装步骤。
CX-Programmer 操作手册	W446	WS02-CX□□-V□	Windows 系统计算机的支持软件 CX-Programmer 的操作步骤	本手册对 CX-Programmer 的操作步骤进行了说明。 请在编程时参考《软件操作手册》(样本编号: W473) 和《CS/CJ/NSJ 系列指令操作手册》(样本编号: W474)。
CX-Programmer 操作手册之功能块 / 结构化文本篇	W447			
CX-Programmer 操作手册之 SFC 编程篇	W469			
CS/CJ/CP/NSJ 系列 CX-Simulator 操作手册	W366	WS02-SIMC1-E	CX-Simulator 仿真支持软件在 Windows 系统计算机中的操作步骤 CX-Programmer 6.1 版或更高版本中使用仿真功能	本手册对 CX-Simulator 的操作步骤进行了说明。 仿真时另请参考《CX-Programmer 操作手册》(样本编号: W446)、《软件操作手册》(样本编号: W473) 和《CS/CJ/NSJ 系列指令参考手册》(样本编号: W474)。
CS/CJ/CP/NSJ 系列 CX-Integrator 网络配置软件操作手册	W464	CXONE-AL□□C-V□/ CXONE-AL□□D-V□	网络设置和监控	本手册对 CX-Integrator 的操作步骤进行了说明。

1

概述

本章节对 CJ2 CPU 单元进行了概述。

1-1 CJ2 CPU 单元概述	1-2
1-1-1 概述	1-2
1-1-2 CJ2 CPU 单元特性	1-4
1-2 基本操作步骤	1-12

1-1 CJ2 CPU 单元概述

1-1-1 概述

SYSMAC CJ2 系列 CPU 单元是一种多功能 CPU 单元，其相关特性如下所示。

- **处理速度快，存储容量大**

基本性能高速化，并且提升了存储器容量，机械控制更加游刃有余。

- **内置 EtherNet/IP 端口 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)**

CJ2 CPU 单元标配 EtherNet/IP 开放式网络，可通过连接外部设备实现 PLC 间数据链接、PLC 间报文通信及 FTP 传送等常规的 Ethernet 通信功能。

- **用于支持软件接口的通用网络**

使用市售的电缆，即可将支持软件及设备经由 USB 端口 /EtherNet/IP 端口连接到通用网络。(仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 提供了 EtherNet/IP 端口。)

- **标签访问 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)**

CJ2 CPU 单元具备一个标签名称服务器，可对标签名称和 I/O 地址进行管理。即使在 I/O 地址未知的条件下，也可通过使用标签名称从外部设备进行访问。

- **单元同步运转 (1.1 版或更高版本的 CJ2H CPU 单元)**

可对 CPU 总线单元和高功能 I/O 单元的处理时序进行同步，最大程度地降低从输入、处理到输出过程中的时序波动，可更加轻松地确保其应用性能。

- **编程更简单**

CJ2 CPU 单元提供了一个可读性极强的编程环境，其中包括 DM/EM 区的位寻址、地址偏移设定、数组变量使用等特性。

- **强化调试功能**

对联机编辑和数据跟踪功能进行了强化，从而大幅提升调试效率。

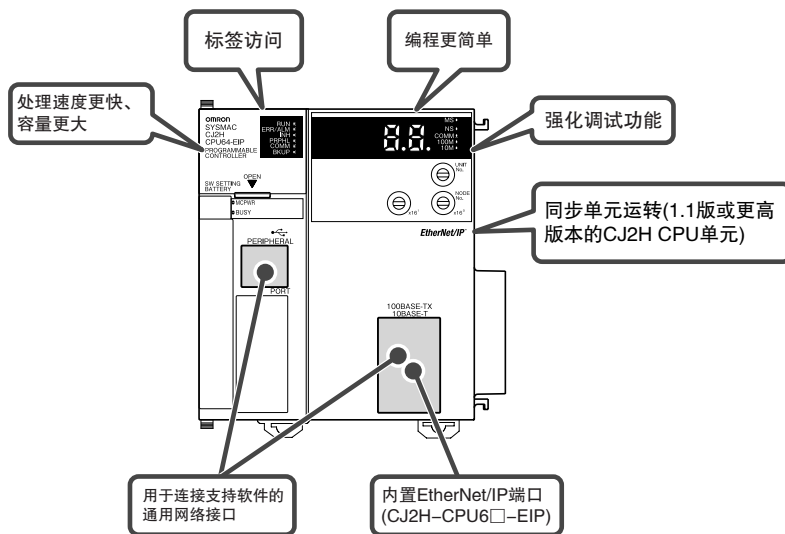
- **更大的脉冲 I/O 容量 (仅 2.0 版或更高版本的 CJ2M CPU 单元支持)**

将可选脉冲 I/O 模块安装至 CJ2M CPU 单元，即可实现多达 4 轴的脉冲 I/O 操作。

- **更丰富的串行通信端口 (仅限 CJ2M-CPU3□)**

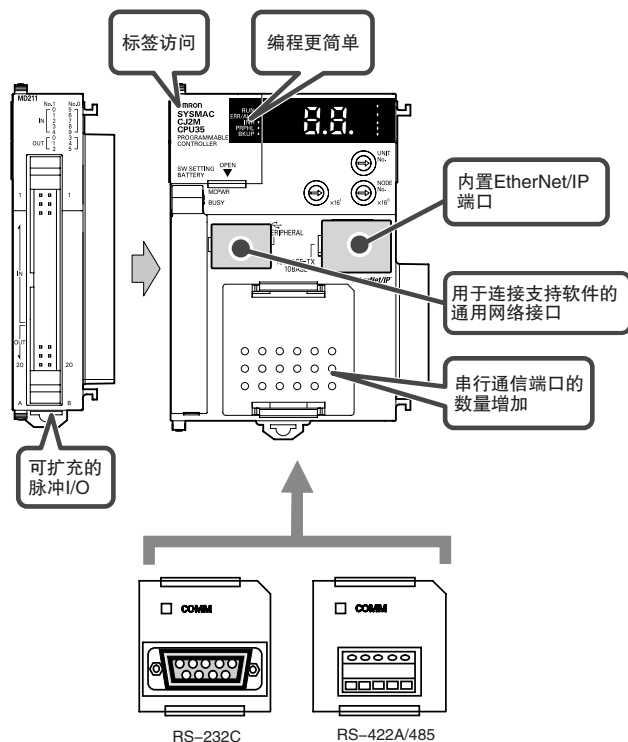
对于标准型 CJ2M CPU 单元 (CJ2M-CPU3□)，可在其标配的 EtherNet/IP 端口上加装 RS-232C 或 RS-422A/485 选件板，从而增加串行端口的数量。

CJ2H CPU 单元

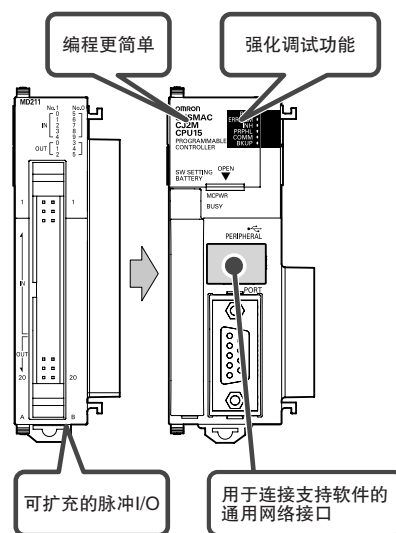


CJ2M CPU 单元

标准型 CPU 单元 (CJ2M-CPU3□)



简易型 CPU 单元 (CJ2M-CPU1□)



1-1-2 CJ2 CPU 单元特性

处理速度快，存储容量大**● 高速处理**

实现对基本指令 (CJ2H: 0.016 μ s 以上, CJ2M: 0.04 μ s 以上)、特殊指令 (CJ2H: 0.05 μ s 以上, CJ2M: 0.06 μ s 以上) 和即时刷新指令 (CJ2H: 0.99 μ s 以上, CJ2M: 1.26 μ s 以上) 的高速处理。

● 大程序容量

CJ2 CPU 单元提供高达 400K 步的大程序容量。

● 特殊功能块定义区域 (仅限 CJ2M)

CJ2M CPU 单元中提供一个名为 FB 程序区的专用区域, 用于存储功能块的定义。(FB 程序区的容量为 20K 步。) 您可利用先前程序中的功能块来创建新的程序部件和架构, 同时又可降低用户程序区的使用率。任何超出 FB 程序区容量范围的功能块定义将存储在用户程序区中。

● 大容量数据存储

CJ2 CPU 单元提供高达 800K 字 (25 组) 的大容量 EM 区。

● 最多支持 128 个循环任务

用户程序最多可划分为 128 个任务。通过此操作, 可实现更小单位下的编程, 从而简化编程作业并缩短循环时间。

● 更高的中断任务执行效能 (1.1 版或更高版本的 CJ2H CPU 单元)

即使在正常使用情况下, 1.1 版 CJ2H CPU 单元的中断任务的开销处理时间相比 1.0 版还是缩短了约 20%。此外, 利用高速中断功能还可提高对下述情况 (带有某些限制条件) 的执行效能。

- 大幅降低中断任务的开销处理时间 (中断任务启动时间 + 循环任务返回时间)

例如: I/O 中断任务在正常操作情况下的处理时间为 37 μ s, 而在使用高速中断功能的情况下可缩短至 25 μ s。

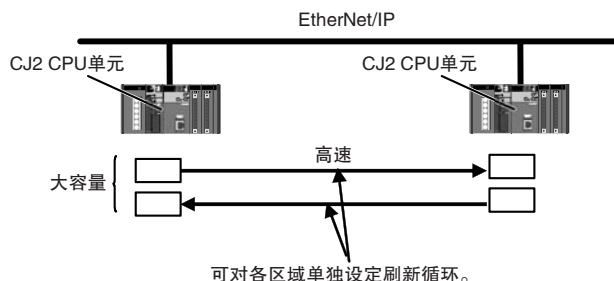
- 定时中断任务的最小间隔设定为 0.1ms (1.0 版的最小间隔设定为 0.2ms。)

内置 EtherNet/IP 端口 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)

CJ2 CPU 标配内置 EtherNet/IP 端口, 并支持与 EtherNet 单元 (包括 FTP 服务器、时间调整等功能) 和 EtherNet/IP 开放式网络相同的 FINS Ethernet 功能。

- 可自由设定刷新循环的高速大容量数据链接

CJ2 CPU 单元提供高速大容量数据链接功能, 并可单独设定各数据链接区的刷新循环。



此外, 使用 EtherNet/IP 数据链接工具时, 可在与 Controller Link 数据链接自动设置类似的界面下实现数据链接的设定。

用于支持软件接口的通用网络

- 外围 USB 和 EtherNet/IP 端口

可通过市售的 USB 电缆经由个人计算机的 USB 端口直接连接至 PLC。此外, 对于 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□, 还可通过 USB 端口访问 EtherNet/IP 网络中的 PLC。

- 可通过支持软件定义 PLC 名称, 有效防止连接至错误的 PLC

CJ2 CPU 单元可保存自定义的 PLC 名称。使用支持软件进行 PLC 连线时, 可通过支持软件确认 PLC 名称, 有效避免错误连接。

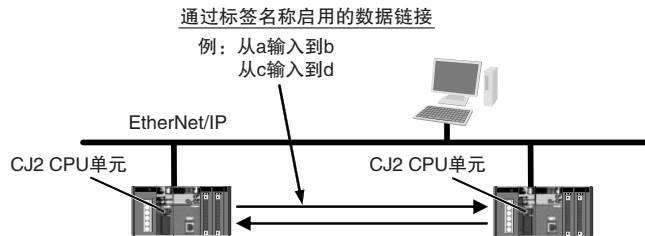
标签访问 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)

● 通过网络符号 (标签) 灵活实现程序变更

CJ2 CPU 单元支持网络符号 (标签)。通过 CPU 单元内置的标签名称服务器, 可将标签名称和地址存储在 CPU 单元的符号表内, 由此实现如下功能。

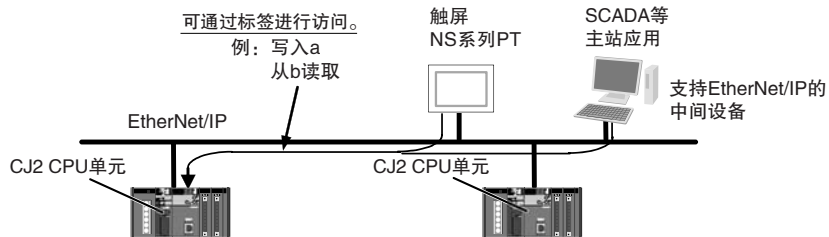
通过标签名称设定数据链接

在 EtherNet/IP 网络的数据链接中, 可不使用地址而仅通过网络符号指定收发区域。在通过标签指定的数据链接区中的内容保持不变的情况下, 仅需修改包含标签名称和地址的符号表即可灵活实现设计变更。



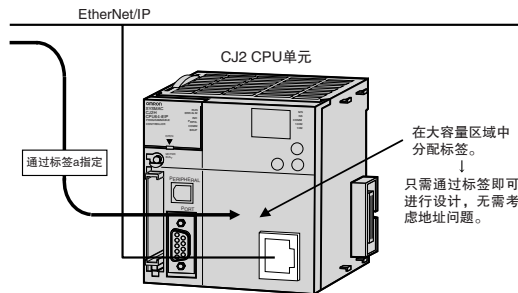
通过标签从上位设备进行访问

可通过标签从上位设备 (如 NS 系列触屏式可编程终端等) 进行访问, 从而使得在 PT 等设备中创建上位设备画面和程序更为方便。



支持标签自动分配功能, 无需考虑地址问题

通过 CX-Programmer 符号 (标签) 表的地址自动分配功能, 可对大容量 EM 区的标签进行自动分配, 因而无需考虑地址问题即可实现数据链接设计及从上位设备进行访问。

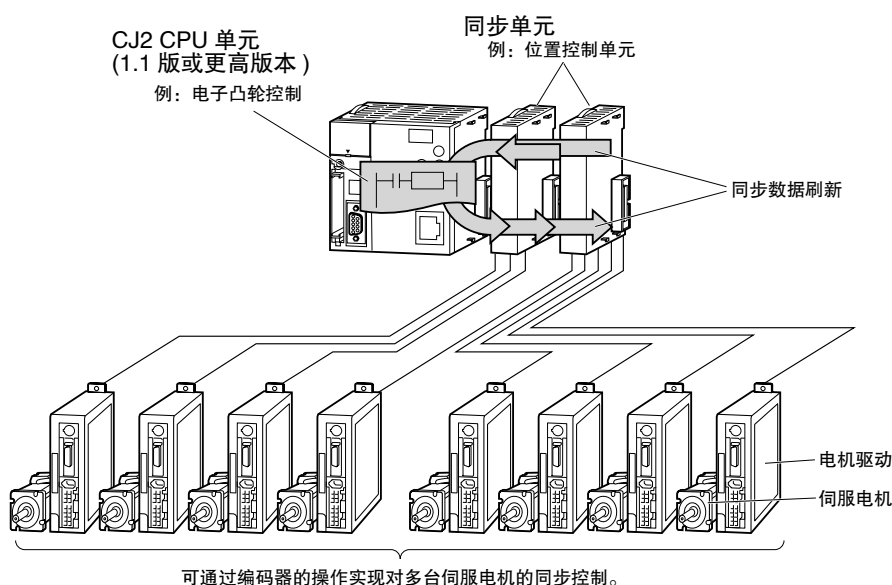


单元同步运转 (1.1 版或更高版本的 CJ2H CPU 单元)

- CJ2 CPU 单元可将 CPU 总线单元和高功能 I/O 单元执行的处理循环的起始部分与一个指定的循环进行同步。

可在 CPU 单元和同步单元 * 之间进行同步数据刷新, 刷新后的数据可在同步中断任务操作中使用。由此, 可最大程度地降低从输入、处理到输出过程中的时序波动, 从而实现多轴控制。对于需要进行精确时序定位的运动控制领域 (如电子凸轮控制等) 而言, 这一特性可以提供极大的帮助。

* 同步单元指支持单元同步运转功能的 CPU 总线单元和高功能 I/O 单元。截至 2010 年 7 月为止, 仅 CJ1W-NC214、CJ1W-NC234、CJ1W-NC414 和 CJ1W-NC434 位置控制单元支持同步单元运转。



编程更简单

- 指定地址偏移

为指令操作数指定地址时, 可在地址后面的括号中指定偏移量。例如, 通过在括号中设定字地址来指定偏移量, 即可根据该字的内容对地址进行动态指定。

示例

W0.00[D0]: W0.00 为起始地址, D0 的内容为偏移量。若 D0 为 &3, 则指定为 W0.03。

- 为数组变量索引指定符号

通过为数组变量下标指定符号, 可根据符号值实现元素的动态指定。

示例

a[b]: 将符号 b 的值指定为数组变量 a[] 的元素。

- 创建和使用数据结构

CX-Programmer 9.0 版或更高版本中可实现数据结构的创建和使用, 这样可以方便地将 CPU 单元中的 I/O 存储器用作数据库或使用库中的数据。

● 可在 DM/EM 区中使用位地址

以往仅可通过字地址对 DM 和 EM 区进行寻址，而无法指定位地址。而现在通过启用 DM 和 EM 区内的位地址，即可扩展位地址的工作区。

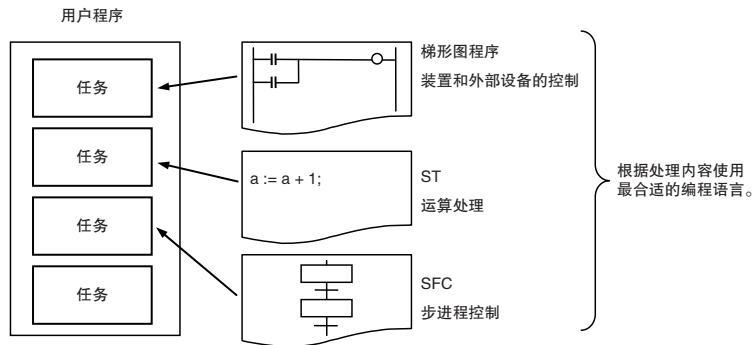
例如，D10.00 用于指定 D10 的 00 位。

● 根据各指令单独选择定时器 / 计数器当前值 (PV) 的刷新方式

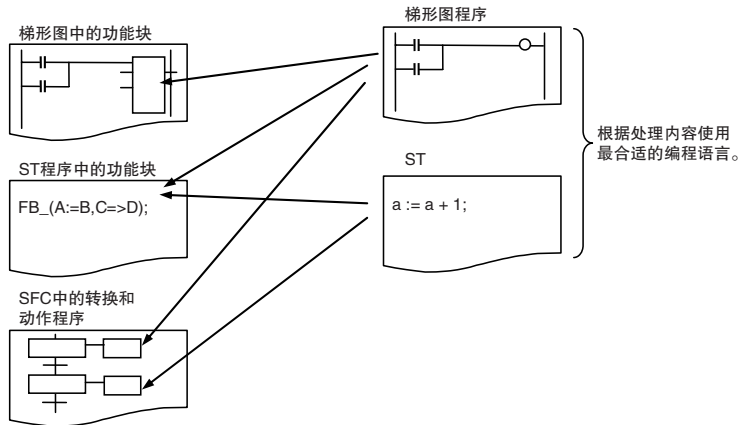
可根据各指令分别选择定时器/计数器的当前值刷新方式(BCD/二进制)。例如，TIM(BCD)和TIMX(BIN)指令可同时使用。

● 可在结构化编程中使用最佳的语言组合

可在用户程序中自由组合梯形图语言、ST 语言和 SFC 语言。根据特定的处理内容选择最适合的语言组合，可有效降低编程时间。



此外，还可在梯形图和 ST 程序的功能块 (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，从而实现结构化编程的优化。



强化调试功能

- 减少联机编辑操作造成的延时影响

将联机编辑操作造成的附加循环时间缩短至约 1ms，从而最大程度地降低调试期间对设备运转的影响。

- 快速、大容量数据跟踪

可跟踪高达 32K 字的数据 (原有水平的 8 倍以上)，具有丰富的触发条件，适合对数据进行长时间的持续跟踪。

- 用于 9.12 版或更高版本的 CX-Programmer 的 I/O 模块辅助区选择功能 (仅限 CJ2M CPU 单元)

对于 9.12 版或更高版本的 CX-Programmer，可方便地从列表中选择需要跟踪的数据。

- 可对指定 EM 区进行强制置位 / 复位

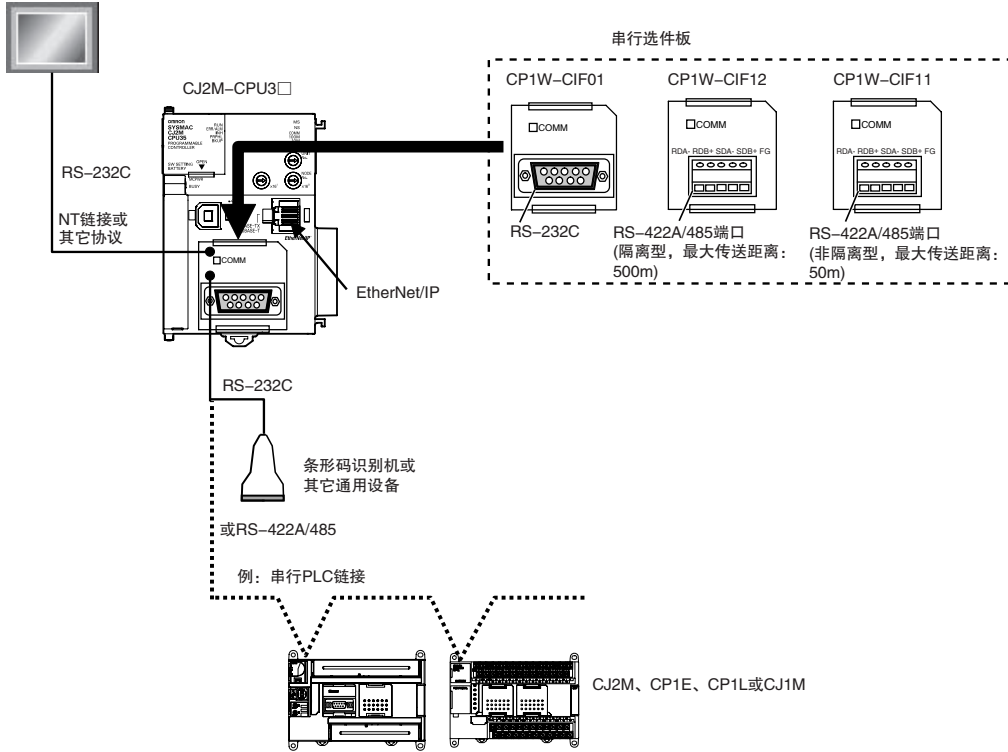
CX-Programmer 8.3 版或更高版本可进行相应的参数设定，从而实现指定 EM 区的强制置位 / 复位。(仅 CJ2H CPU 单元 (1.2 版或更高版本) 和 CJ2M CPU 单元支持此项功能。)

可加装串行选件卡 (仅限 CJ2M-CPU3□)

CJ2M-CPU3□ 可加装一块串行通信选件板 (附带单个 RS-232C 端口或 RS-422A/485 端口)。

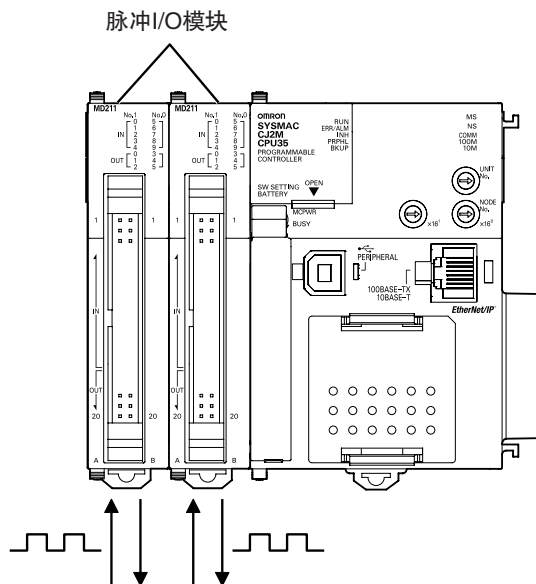
凭借串行端口, 可方便地连接条形码识别机等通用设备及 PT、CJ/CP 系列 PLC 及变频器等其它设备。

示例: NS系列PT



使用脉冲 I/O 和中断输入 (仅限 CJ2M CPU 单元)

CJ2M CPU 单元最多支持安装 2 个脉冲 I/O 模块。可实现多项输入功能 (如通用输入、中断输入、高速计数器输入和快速响应输入) 和输出功能 (如通用输出、脉冲输出和可变占空比脉冲输出)，为各种类型的应用场合提供支持。



- | | |
|---------------------------|-----------------------------|
| • 普通输入：20点(每个脉冲I/O模块10点) | • 普通输出：12点(每个脉冲I/O模块6点) |
| • 中断输入：8点(每个脉冲I/O模块4点) | • 脉冲输出：4点(每个脉冲I/O模块2点) |
| • 高速计数器输入：4点(每个脉冲I/O模块2点) | • 可变占空比脉冲输出：4点(每个脉冲I/O模块2点) |
| • 快速响应输入：8点(每个脉冲I/O模块4点) | |

1-2 基本操作步骤

通常情况下应遵循以下操作步骤。

1. 设置设备和硬件

安装电源单元、CPU单元、其它单元和端板。按照需要设定DIP开关和旋转开关。

请参考《CJ2 CPU单元硬件操作手册》(样本编号: W472)中的“第3章 各部分名称与功能”和“第5章 安装”。

2. 配线

进行电源、I/O和通信配线。请参考下列手册。

请参考《CJ2 CPU单元硬件操作手册》(样本编号: W472)的“第5章 安装”。请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“第11章 编程设备与通信”。请参考《CJ2M CPU单元脉冲I/O模块操作手册》(样本编号: W486)中的“第3章 I/O模块的规格和配线”。

3. 联机到PLC

将个人计算机和PLC联机。

请参考《CJ2 CPU单元硬件操作手册》(样本编号: W472)中的“第4章 支持软件”。

4. I/O分配

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

请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“第8章 I/O分配与单元设定”。请参考《CJ2M CPU单元脉冲I/O模块操作手册》(样本编号: W486)中的“第2章 I/O应用步骤和功能分配”。

5. 软件设置

进行PLC的软件设置。

- 请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“第4章 CPU单元的初始化”。
- 高功能I/O单元和CPU总线单元的DM区设定: 请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“第8章 I/O分配与单元设定”。
- PLC设置: 请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“第9章 PLC设置”。

6. 创建程序

使用CX-Programmer创建程序。

请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“第5章 编程”。

7. 检查操作

检查I/O配线和AR区设定并执行试运转。CX-Programmer可用于监控和调试。

请参考《CJ2 CPU单元软件操作手册》(样本编号: W473)中的“10-7 调试”。

8. 基本程序操作

将运行模式设为RUN模式以开始运行。

2

CPU 单元内部存储器

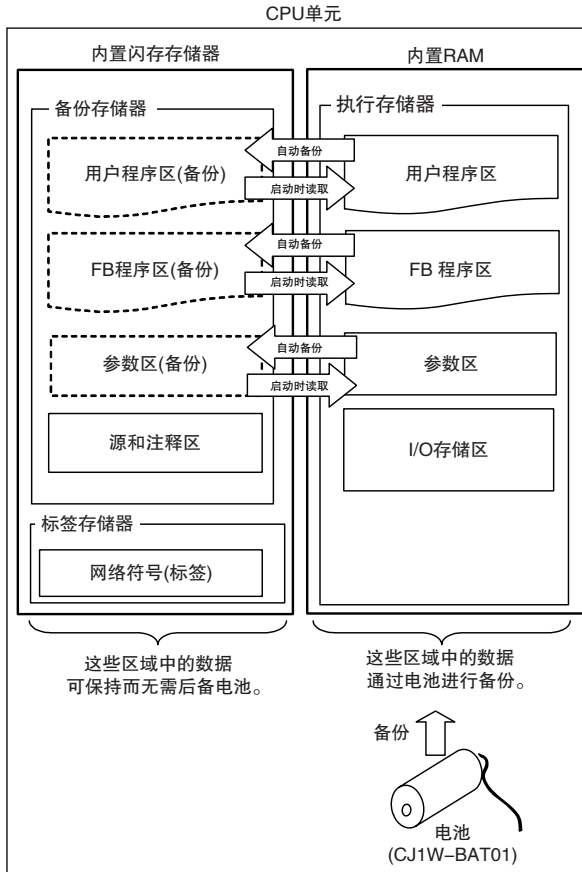
本章节阐述了 CPU 单元中包含的存储区。

2-1 概述	2-2
2-1-1 存储器配置	2-2
2-1-2 存储区及存储的数据	2-3
2-1-3 将数据从编程设备传送到 CPU 单元	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 用作执行存储器，内置闪存存储器用作备份存储器。



内置闪存存储器

内置闪存存储器备份用户程序和参数，并存储程序源代码、注释和网络符号。该区域中的数据可保持而无需电池。I/O 存储器中的数据不会被备份。

内置 RAM

内置 RAM 为 CPU 单元的执行存储器，用户程序、参数和 I/O 存储器数据存储在内置 RAM 中，内置 RAM 通过电池进行备份。

如果电池不工作（例如，电池电压过低或者未安装电池时），I/O 存储器的数据将丢失。由于用户程序和参数已备份到内置闪存存储器中，因此不会丢失。



正确使用注意事项

当电池电压过低或者未安装电池时，将发生下述情况。

- I/O 存储器区中的数据将丢失，或者数据的值将变得不稳定（包括 DM、EM 和 HR 区，这些区中的值在电源断开时通过电池来保持。）。
- 时钟将停止，且与时钟相关的数据将变得不稳定。
- 出错记录将不会保持。
- 输出 OFF 位将变得不稳定。

2-1-2 存储区及存储的数据

下表列出了各个 CPU 单元存储区及存储的数据。

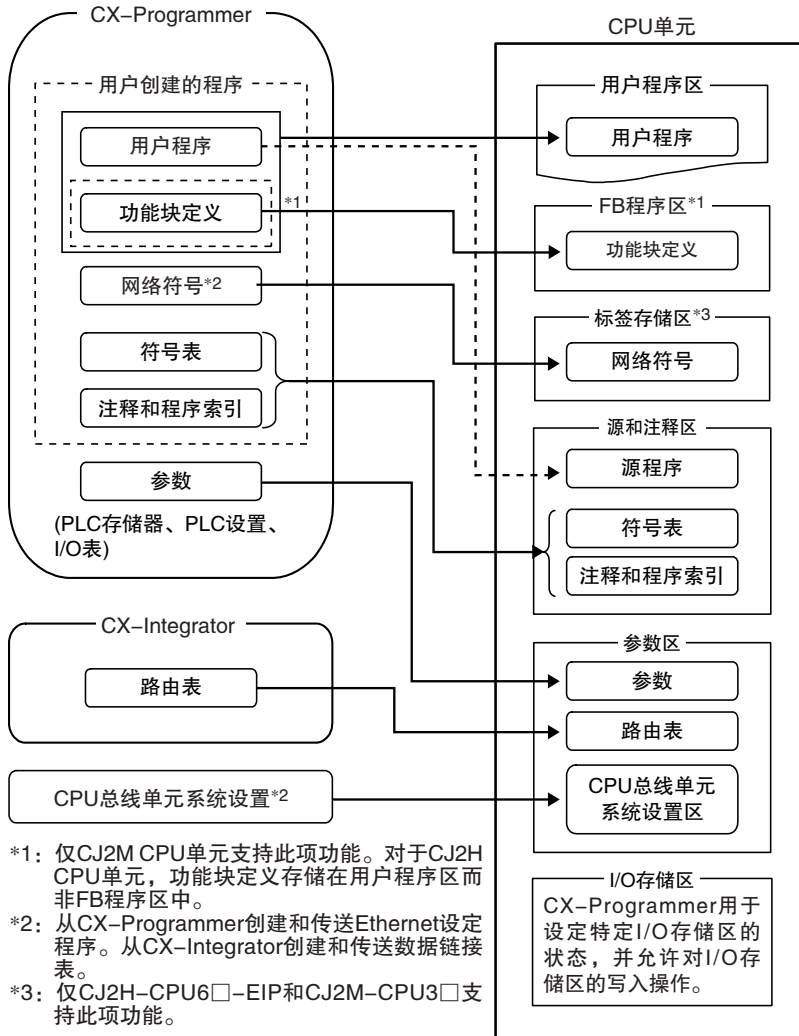
存储区及存储的数据	详细信息	内置闪存存储器	内置 RAM
用户程序区	存储目标代码，以用于执行 CX-Programmer 创建的用户程序。	存储	存储
FB 程序区*1	FB 程序区存储通过 CX-Programmer 创建的功能块定义。	存储	存储
参数区	参数区存储 PLC 的初始设定。	存储	存储
PLC 名称	CPU 单元的名称可通过 CX-Programmer 进行存储、读取和验证，以防止 CX-Programmer 连接到错误的 PLC。		
PLC 设置	使用软件开关在 PLC 设置中进行各种初始设定。请参考“第 9 章 PLC 设置”。		
I/O 表	I/O 表提供用户所指定单元的安装状态的信息。请参考“第 8 章 I/O 分配与单元设定”。		
路由表	路由表是用于 FINS 通信的网络参数。路由表通过 CX-Integrator 进行指定。请参考《CX-Integrator 操作手册》(样本编号: W464)。		
CPU 总线单元设置	CPU 总线单元设置存储特定 CPU 总线单元的初始设定。该设置包括用于 Ethernet 单元的 Ethernet 设定和用于 Controller Link 单元的数据链接参数等设定。		
I/O 存储区	I/O 存储区用于读 / 写用户程序。该存储区根据用途分成下述几个区域。 一个当 CPU 单元电源复位时清除数据的区域以及一个保存数据的区域。 一个与其它单元交换数据的区域以及一个内部使用区域。	---	存储
源和注释区	源和注释区用于存储通过 CX-Programmer 创建的程序源代码和注释。	存储	---
源代码	(通过梯形图、ST 和 SFC 语言编写的任务和功能块中的) 程序的源代码。		
符号表	包含通过 CX-Programmer 创建的符号 (符号名称、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	提供由 CX-Programmer 创建的程序段信息以及程序注释。		
网络符号 (标签)*2	全局符号表中用于网络符号的数据。	存储	---

*1 仅 CJ2M CPU 单元支持此项功能。对于 CJ2H CPU 单元而言，功能块定义存储在用户程序区而非 FB 程序区中。

*2 仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 支持此项功能。

2-1-3 将数据从编程设备传送到 CPU 单元

如下图所示，将通过 CX-Programmer 或 CX-Integrator 创建的数据传送到 CPU 单元的内部存储器。



3

CPU 单元运行

3

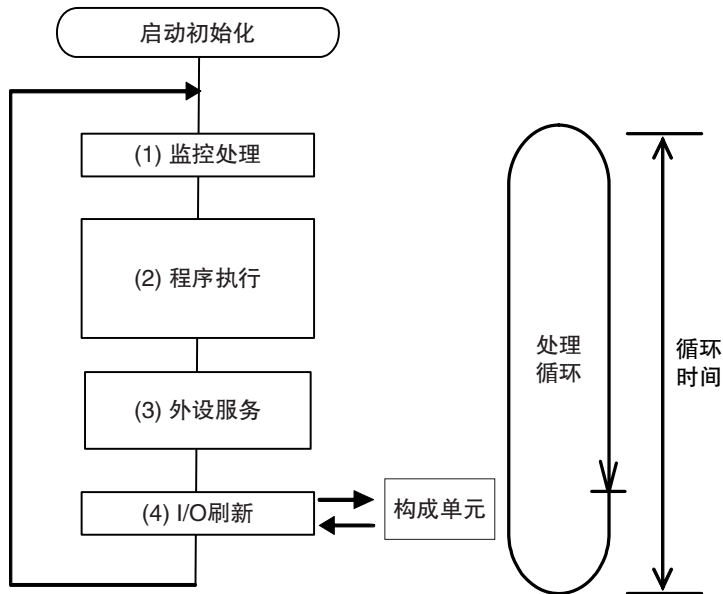
本章描述了 CPU 单元的内部运行以及可对 CPU 单元设定的运行模式。

3-1 CPU 单元的内部运行	3-2
3-1-1 概述	3-2
3-1-2 循环时间	3-4
3-1-3 电源中断时的处理	3-7
3-2 CPU 单元运行模式	3-8
3-2-1 运行模式	3-8
3-2-2 检查运行模式	3-9
3-2-3 运行模式的变更	3-10
3-2-4 运行模式详情	3-14

3-1 CPU 单元的内部运行

3-1-1 概述

CPU 单元循环执行用户程序时，将数据写入内部 I/O 存储区中。当刷新 I/O 和进行外设服务时，与外部进行数据交换。下图所示为 CPU 单元的内部运行。



启动初始化

每当接通 PLC 的电源时，均会执行一次下述过程。

- 检测已连接的单元 (I/O 分配)
- 比较已注册的 I/O 表和已连接的单元
- 根据 IOM 保持位的状态来清除 I/O 存储器中的非保持区
- 根据强制状态保持位的状态来清除强制状态
- 如果安装了存储卡，则使用该存储卡中的自动传送文件进行自动引导
- 执行自诊断 (用户存储区检查)
- 恢复用户程序
- 更新 PLC 设置

处理循环

启动初始化之后，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）。完成一个循环所需时间称为循环时间。

- 检查
该步骤为执行用户程序的做准备。该步骤检查处理所需的硬件和软件。检查步骤所需的时间称为开销时间。
- 程序执行
该步骤执行用户程序中的指令。当运行模式设定为 RUN 或 MONITOR 模式时，将执行用户程序中的指令。指令的 I/O 处理对 CPU 单元的 I/O 存储区中的位和字进行执行。
- 外设服务
当编程设备或单元需要信息时，将执行该处理。从 CX-Programmer 写入数据，或者执行网络通信单元的指令。
- I/O 刷新
该步骤通过 I/O 存储区与单元交换数据。该处理步骤始终在程序执行之后执行。I/O 刷新在当前循环内无中断执行（即：不产生时间碎片）。

3-1-2 循环时间

循环时间是检查、程序执行、外设服务和 I/O 刷新的总时间。循环时间通常会因所执行的各个处理而波动。每个处理均在循环时间内执行。有关计算循环时间的说明，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恒定循环时间。最小循环时间必须大于正常循环时间。有关设定最小循环时间的说明，请参考“10-2-1 最小循环时间”。

检查

检查每个循环时均会产生开销时间。将检查下列各项。

- 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

程序执行时间是用户程序中指令的处理时间。

- 该处理时间包括处于 READY 状态下的循环任务中的指令和满足中断条件的中断任务中的指令的处理时间。但不包括程序中跳转的指令的处理时间。
- 处理时间取决于指令的数量以及所使用的特定指令。另外，还取决于所指定的操作数 (例如, 要处理的数据量)。
- 对于 RUN 或 MONITOR 模式, OFF 执行条件下的指令需要内部处理时间。下述情况下, 不执行内部处理, 因而不需要处理时间。
 - WAIT 状态下的循环任务
 - 因 JMP(004)、CJP(510) 或 CJPN(511) 指令而跳转的指令



附加信息

必须缩短循环时间时, 可考虑下述编程方法。

- 将程序划分成循环任务, 并在任务无需执行时将任务置于 WAIT 状态。
- 当某些指令无需执行时, 使用 JMP(004)、CJP(510) 和 CJPN(511) 指令来跳转这些指令。

外设服务

外设服务是指对外部设备的不定期事件进行服务。该服务包括来自外部设备的事件，也包括对外部设备的服务请求。

对于 CJ 系列 PLC，大多数外设服务涉及 FINS 指令。在系统中指定的时间量将分配到各类型的服务中并在每个循环中执行。如果在已分配的时间段内无法完成所有服务，则剩余的服务将在下一循环中执行。如果外设服务延迟的时间过长，则可使用 CX-Programmer 将 PLC 设置中的固定服务时间增加 10% 以上。

● 外设服务时间

采用默认的外设服务时间设定时，将上一循环时间的 10% 指定为总外设服务时间。因此，外设服务时间取决于上一循环时间。

当由于完成外设服务所需的循环过多而导致外设服务延迟时，可通过 CX-Programmer 在 PLC 设置中将固定服务时间设定为大于循环时间的 10%。需注意的是，增加服务时间也将增加循环时间。

● 外设服务类型

单元	服务
高功能 I/O 单元的事件服务 CPU 总线单元的事件服务	<ul style="list-style-type: none"> 来自高功能 I/O 单元和 CPU 总线单元的 FINS 命令的非预定服务（例如，请求启动外部中断任务）。 从 CPU 单元到上述单元的 FINS 命令的非预定服务。
外设 USB 端口服务 串行端口服务	<ul style="list-style-type: none"> 对从编程设备、PT 或上位计算机经由外设 USB 或串行端口接收到的 FINS 或上位链接命令进行非预定服务（例如传送编程、监控、强制置位 / 复位操作或联机编辑的请求）。 从 CPU 单元发出的从外设 USB 或串行端口传送来的非预定服务（非请求的通信）。
通信端口服务	<ul style="list-style-type: none"> 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) 指令执行网络通信、串行通信或文件存储器访问的服务。 使用通信端口执行后台处理的服务。
文件访问服务	对存储卡或 EM 文件存储器进行的文件读 / 写操作。
联机编辑服务	正在处理联机编辑时，将通过联机编辑功能所编辑的内容备份到内置闪存存储器中的服务。
备份服务	当程序或参数改变时，将改变后的内容备份到内置闪存存储器中的服务。



附加信息

服务时间分配到高功能 I/O 单元、CPU 总线单元、外设 USB 端口、串行端口、文件访问和通信端口。如果服务分解到多个循环中，导致服务延迟完成，则可在 PLC 设置的执行时间设定项中设定相同的分配时间（对所有服务的时间相同），而非一个百分比。

I/O 刷新

I/O 单元的处理时间取决于 PLC 中所使用的配置单元的类型和数量。各单元的处理时间是固定的。

可通过停止高功能 I/O 单元的循环 I/O 刷新来缩短处理时间。若要停止高功能 I/O 单元的 I/O 刷新，可在 CX-Programmer 的 SIOU 刷新选项卡页中设定参数。下表列出了 PLC 单元的刷新处理信息。

单元		最大数据交换	数据交换区	
基本 I/O 单元		取决于单元	I/O 位区	
高功能 I/O 单元	CIO 区分配字	10 字 / 单元 (取决于单元)	高功能 I/O 单元区	
	单元特定数据 CompoBus/S 主站单元	取决于单元	为远程 I/O 通信设定的字	
CPU 总线单元	CIO 区分配字	25 字 / 单元	CPU 总线单元区	
	DM 区分配字	100 字 / 单元	CPU 总线单元字	
	单元特定数据 (参考右栏)	内置 EtherNet/IP 端口 * EtherNet/IP 单元	取决于所使用的 EtherNet/IP 功能	标签数据链接区
		Controller Link 单元和 SYSMAC LINK 单元	取决于单元	为数据链接所设定的字 (固定或由用户设定来分配)
		DeviceNet 单元	取决于单元	为远程 I/O 通信设定的字 (固定或由用户设定来分配)
		串行通信单元	取决于协议宏	为协议宏所设定的通信数据
Ethernet 单元		取决于单元	为通过特定控制位操作进行初始化的套接字服务所设定的通信数据	

* 仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 支持此项功能。CJ2H-CPU6□-EIP 的内置 EtherNet/IP 端口可以提供与 CJ1W-EIP21 EtherNet/IP 单元相同的性能和功能。而 CJ2M-CPU3□ 的内置 EtherNet/IP 端口在性能方面有较大的差异。详情请参考《EtherNet/IP 单元操作手册》(样本编号: W465)。



附加信息

当执行指令时而非正常的 I/O 刷新期间，可与单元交换数据。该过程称为即时刷新。在某些指令之前加上感叹号 (!)，或者使用 IORF(097)、FIORF(225) 和 DLNK(226) 指令即可进行即时刷新。

下列指令通过直接处理可用于提高与某些高功能 I/O 单元或 CPU 总线单元进行数据交换的速度。详情请参考《指令参考手册》(样本编号: W474)。

- 模拟量输入直接转换 (AIDC(216))(对于 CJ1W-AD042 高速模拟量输入单元)*¹
- 模拟量输出直接转换 (AODC(217))(对于 CJ1W-DA042V 高速模拟量输出单元)*¹
- 通过串行通信单元 / 串行端口直接发送 (DTXDU(262))(仅限 CJ1W-SCU□2)*¹
- 通过串行通信单元 / 串行端口直接接收 (DRXDU(261))(仅限 CJ1W-SCU□2)*¹
- PCU 高速定位 (NCDMV(218))(仅限 CJ1W-NC□□4 和 CJ1W-NC□)*²
- PCU 定位触发器 (NCDTR(219))(仅限 CJ1W-NC□81)*²

*¹ 仅 CJ2H-CPU6□(-EIP)(1.1 版或更高版本) 和 CJ2M CPU 单元支持这些指令。

*² 仅 1.3 版或更高版本的 CJ2H-CPU6□(-EIP) 和 CJ2M CPU 单元支持这些指令。

3-1-3 电源中断时的处理

如果在 CPU 单元运行过程中发生电源中断并且中断得到确认，则将当前正在执行的指令执行完，然后将执行下述电源中断处理。

电源中断时的处理将根据对电源 OFF 中断任务所做的设定来执行。^{*1}

- 如果已允许电源 OFF 中断任务，则将执行该任务，然后立即复位 CPU 单元。
- 如果未允许电源 OFF 中断任务，则将立即复位^{*2} CPU 单元。

^{*1} 因电源中断的缘故，将在对 CPU 单元复位之前立即执行电源 OFF 中断任务。

^{*2} 当在运行过程中发生电源中断时，允许在执行指定范围的指令之后执行电源中断处理。

若 AC 电源发生短于 10ms 的瞬时断电，或者 DC 电源发生短于 2ms 的瞬时断电，则系统将连续运行。在供电状况较差的情况下，可延长从检出电源中断到确认为电源中断之间的时间。

有关发生电源中断时的处理，请参考“A-5 断电时的操作”。

3-2 CPU 单元运行模式

3-2-1 运行模式

可设定运行模式以控制 CPU 单元的运行状况并控制是否可在 CPU 单元中进行设定。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的运行速度。

将执行程序。

无法对位进行强制置位 / 复位，无法改变 I/O 存储器中的值，且无法进行联机编辑。

- **MONITOR 模式：**

MONITOR 模式用于试运行和调整。

将执行程序。

可以对位进行强制置位 / 复位，可以改变 I/O 存储器中的值，且可以进行联机编辑。

- **PROGRAM 模式：**

PROGRAM 模式用于传送程序和 PLC 设置以及创建 I/O 表。

将不执行程序。

可使用与 CPU 单元连接的 CX-Programmer 来切换运行模式。

3-2-2 检查运行模式

CPU 单元上的前面板指示灯

CPU 单元前面板上 RUN 指示灯以如下方式指示运行模式。

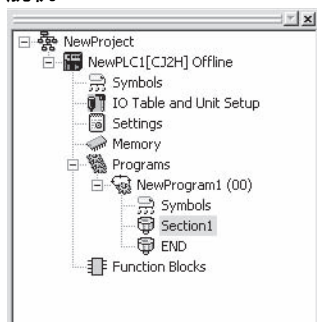
运行模式	CPU 单元上的 RUN 指示灯	说明
PROGRAM 模式	OFF	---
RUN 或 MONITOR 模式	亮绿灯	使用 CX-Programmer 来查看运行模式为 RUN 还是 MONITOR 模式。

CX-Programmer

可在 CX-Programmer 的项目树形图或状态栏中查看运行模式。

● 项目树形图

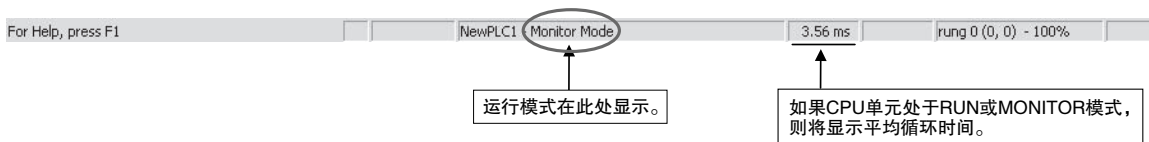
脱机



联机



● 状态栏



附加信息

使用 CJ1W-PA205R 电源可实现外部 RUN 输出。除非存在致命错误, 否则 CJ1W-PA205R 上的 RUN 输出 (触点输出) 将在 RUN 或 MONITOR 模式下变为 ON。

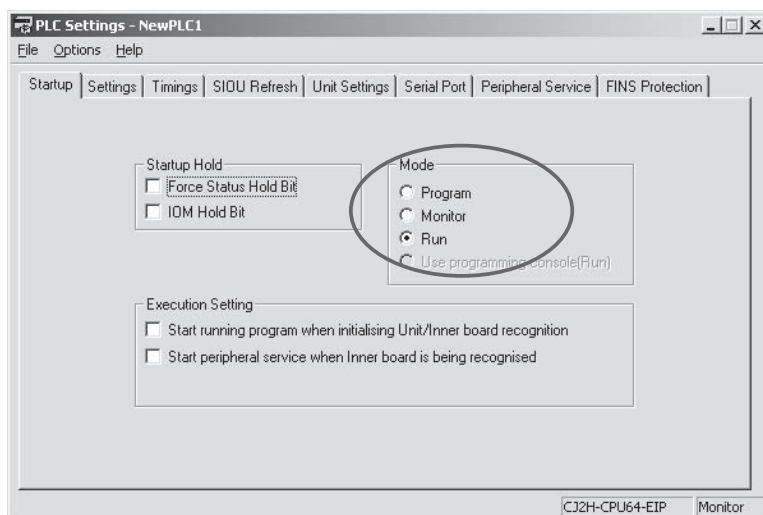
3-2-3 运行模式的变更

运行模式的变更

运行模式可通过 CX-Programmer 进行变更。

- 变更启动模式

CPU 单元置 ON 时，默认运行模式为 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模式。



- 启动后变更运行模式

请使用下列步骤之一进行变更。

- 在“运行模式”菜单中选择 PROGRAM、MONITOR 或 RUN。
- 在项目树形图中右键点击 PLC，然后再在“运行模式”菜单中选择 PROGRAM、MONITOR 或 RUN。

模式变更和电源中断后的存储器状态

在 RUN 或 MONITOR 模式和 PROGRAM 之间变更后，I/O 存储器的某些部分，例如 CIO 区和工作区将被清除。这些区域被称为非保持区。若要在运行模式变更后保持这些区中的内容，可将辅助区中的 IOM 保持位 (A500.12) 置为 ON。

模式变更	非保持区	保持区
	<ul style="list-style-type: none"> · I/O 区 · 数据链接区 · CPU 总线单元区 · 高功能 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 (根据地址决定辅助区位 / 字保持与否) 	<ul style="list-style-type: none"> · HR 区 · DM 区 · EM 区 · 计数器 PV 和完成标志 (根据地址决定辅助区位 / 字保持与否)
从 RUN 或 MONITOR 到 PROGRAM	清除 *1	保持
PROGRAM 至 RUN 或 MONITOR		
从 RUN 到 MONITOR 或从 MONITOR 到 RUN	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，则清除存储器。如果为 ON，则数据将保持，详细情况如下：

I/O 存储器保持位状态 (A500.12)	I/O 存储器			分配到输出单元的输出位		
	模式在 PROGRAM 和 RUN/MONITOR 之间变更	致命错误		模式在 PROGRAM 和 RUN/MONITOR 之间变更	致命错误	
		除 FALS 之外的致命错误	FALS 已执行		除 FALS 之外的致命错误	FALS 已执行
OFF	清除	清除	保持	OFF	OFF	OFF
ON	保持	保持	保持	保持	OFF： 发生致命错误时，CPU 单元的 I/O 存储器中的值将保持，但来自基本输出单元的输出将变 OFF。	

*2 当电源中断被复位时，IOM 保持位 (A500.12) 将变 OFF。为保持状态，请通过 CX-Programmer 在 PLC 设置的“Startup Hold Settings” (启动保持设定) 选项页中选择“IOM Hold Bit” (IOM 保持位) 复选框，然后传送 PLC 设置。

*3 无论 IOM 保持位 (A500.12) 的状态如何，任务标志均会被清除。

模式变更后来自基本输出单元的输出状态

当运行模式在 RUN 或 MONITOR 模式与 PROGRAM 模式之间以默认设定进行变更时，将根据模式变更时所存在的 I/O 存储器状态来刷新来自基本输出单元的输出。

根据 IOM 保持位 (A500.12) 是 ON 还是 OFF，基本输出单元的输出状态将如下。

- 如果 A500.12 是 OFF，则当模式变更为 PROGRAM 模式时，输出位将被清除，因此来自基本输出单元的输出将变 OFF。
- 如果 A500.12 是 ON，则当模式变为 PROGRAM 模式时，输出位的状态将保持，因此来自基本输出单元的输出状态也将保持。



正确使用注意事项

将 IOM 保持位 (A500.12) 设定为 ON 的注意事项

当模式从 RUN 或 MONITOR 模式变更为 PROGRAM 模式时，输出的 I/O 存储器状态将不会被清除 (即：变 OFF)，且变为 PROGRAM 模式之前的状态将保持。因此，当再次从 PROGRAM 模式变为 RUN 或 MONITOR 模式时，将输出变为 PROGRAM 模式之前的 I/O 存储器值。

但如果运行因致命错误而停止 (包括执行 FALS(007) 指令)，则来自基本输出单元的所有输出均将变 OFF，即使 CPU 单元中的 I/O 存储器状态被保持。

PROGRAM 模式下来自基本输出单元的输出状态

即使在 PROGRAM 模式下，来自基本输出单元的输出也将通过 I/O 存储器的输出位状态来刷新。因此，当通过 CX-Programmer 或其它支持软件改变了 I/O 存储器中的输出位时，其状态将从基本输出单元输出。



警告

即使在程序停止时（即：即使在 PROGRAM 模式下），CPU 单元也会刷新 I/O。在改变分配到基本 I/O 单元、高功能 I/O 单元或 CPU 总线单元的存储器中的任何部分的状态之前，请事先充分确认安全性。对分配到任何单元的数据进行任何变更均有可能导致对连接到单元上的负载进行意外操作。任何下列操作均有可能导致存储器状态改变。

- 将 I/O 存储器输出从编程设备传送到 CPU 单元
- 从编程设备改变存储器中的当前值
- 从编程设备对位进行强制置位 / 复位
- 从存储卡或 EM 文件存储器将 I/O 存储器文件传送到 CPU 单元
- 从上位计算机或网络中的另一台 PLC 传送 I/O 存储器

为确保来自基本输出单元的输出保持 OFF，可将输出 OFF 位 (A500.15) 置 ON。(见注) 当所有输出均为 OFF 时（即：当输出 OFF 位为 ON 时），CPU 单元前面板上的 INH 指示灯将点亮。

注 当运行模式变更以及电源变 OFF 和变 ON 时，输出 OFF 位 (A500.15) 的状态将保持，即输出将保持为 OFF。

3-2-4 运行模式详情

下表列出了各运行模式期间的状态。

运行模式		PROGRAM	MONITOR	RUN
目的		停止操作和传送程序	测试操作和进行调整	主运行
程序状态		停止	执行	执行
RUN 指示灯		OFF	ON	ON
通过 CX-Programmer 操作	创建 I/O 表	允许	不允许	
	传送 PLC 设置			
	下载程序			
	检查程序			
	联机编辑	允许	不允许	
	通过强制置位 / 复位位检查配线			
	变更 I/O 存储器的 PV			
	变更定时器 / 计数器的 PV 和 SV			
	下载程序	允许		
	监控 I/O 存储器			
监控程序				
I/O 刷新		执行		
循环任务		停止	设定为在运行开始时启动的任务和通过执行 TASK ON 指令启动的任务。其它任务停止执行。	
中断任务		停止	当满足中断条件时执行。	
来自基本输出单元分配位的输出 *1	IOM 保持位 OFF	在变为 PROGRAM 模式之后立即 OFF。*2	取决于程序	取决于程序
	IOM 保持位 ON	在变为 PROGRAM 模式之后保持状态。*2		

*1 当输出 OFF 位 (A500.15) 为 ON 时, 无论运行模式和 I/O 存储器状态如何, 来自基本输出单元的输出均将变 OFF。即使接通电源, 输出也将保持为 OFF。

*2 即使在 PROGRAM 模式下, 如果通过支持软件或 PT 改变了存储器状态, 则来自输出单元的输出也将被刷新。

4

CPU 单元初始化

本章描述了启动时对 CPU 单元执行的初始化处理。

4

4-1 CPU 单元初始化概述	4-2
4-1-1 CPU 单元初始化设定	4-2
4-2 PLC 设置	4-7
4-3 创建 I/O 表	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 设定路由表	4-10
4-4-1 路由表	4-10
4-4-2 要求使用路由表的情况	4-12
4-4-3 设定和传送路由表	4-13
4-5 为高功能 I/O 单元和 CPU 总线单元设定 DM 区分配字	4-14
4-5-1 为高功能 I/O 单元和 CPU 总线单元设定 DM 区分配字	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 单元初始化设定

通过 CPU 单元前面板上的 DIP 和旋转开关进行的硬件设定和 CX-Programmer 或其它软件进行的用于设定参数的软件设定这两种设定均必须进行设定，从而进行 CPU 所需的初始设定。

硬件设定

● DIP 开关

根据应用来设定 DIP 开关。

位置	目的	设定方法
CPU单元前部的DIP开关	主要用途： • 程序写保护：使开关位 1 变 ON。 • 启动时自动传送：使开关位 2 变 ON。 • 通过 RS-232C 进行 Toolbus 连接：使开关位 5 变 ON。 • 用户定义的开关位：使开关位 6 变 ON。 • 简易备份：使开关位 7 变 ON。	在电源断开的情况下进行设定，然后再接通电源。

● 旋转开关

为 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 上的内置 EtherNet/IP 端口分配了单元号和结点地址，分配方法与 CPU 总线单元相同。根据单元号设定，将字分配到 CIO 区和 DM 区的 CPU 总线单元区中的内置 EtherNet/IP 端口。

有关硬件设定的详情，请参考《CJ2 CPU 单元硬件操作手册》(样本编号: W472) 中的“3-1 CPU 单元”。

软件设定

下表列出了软件设定的应用和适用单元的设定方法。

单元	设定	应用	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	防止错误连接至 CPU 单元	CX-Programmer	CX-Programmer 项目文件 (.CXP)	内置闪存存储器
		PLC 设置	使用非默认规格			
		注册 I/O 表	检查是否存在错误连接以及是否需要自动分配或保留字以便将来分配或用作其它应用程序的特定设定。			
		路由表	当连接到一个以上的 FINS 网络通信单元时需要 (*1)	CX-Integrator	路由表 (.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 Links 使用用户设定的数据链接	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)		
高功能 I/O 单元和 CPU 总线单元	I/O 存储器	分配至高功能 I/O 单元或 CPU 总线的 DM 区字 (*2)	使用高功能 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 将下列单元归类为 FINS 网络通信单元。

- 归类为 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
- 将 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 的内置 EtherNet/IP 端口用于 FINS 网络通信
- 使用带串行通信单元的串行网关并使用路由表。
路由表不适用于高功能 I/O 单元的网络通信单元 (例如, CompoNet Master 单元)。

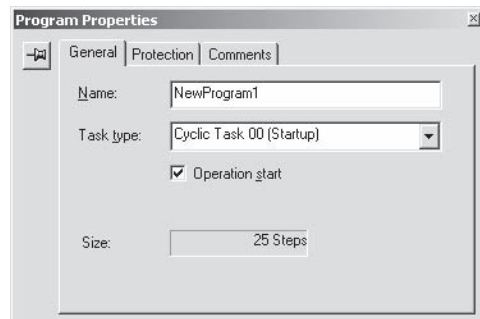
*2 高功能 I/O 单元或 CPU 总线单元的上述初始设定存储在 CPU 单元中。存储在高功能 I/O 单元和 CPU 总线单元中的数据和程序使用对单元而言特定的支持软件分别创建, 然后这些数据和程序经 CPU 单元被传送到高功能 I/O 单元和 CPU 总线单元。



附加信息

设定程序任务

用于将程序指定到任务的初始设定可使用 CX-Programmer 通过下述程序属性来设定。



● PLC 名称

是用户为 CPU 单元设定的名称。通过在 CX-Programmer 的 PLC 菜单中选择“PLC Info – PLC Info” (PLC 信息 – PLC 信息) 即可设定名称。

当 CX-Programmer 联机时，系统将检查 PLC 中注册的名称是否与项目中的 PLC 名称相符。

● PLC 设置

PLC 设置用于为采用非默认规定来使用 CPU 单元而进行变更。下列设定为 CPU 单元的默认设定例。

启动模式：RUN 模式

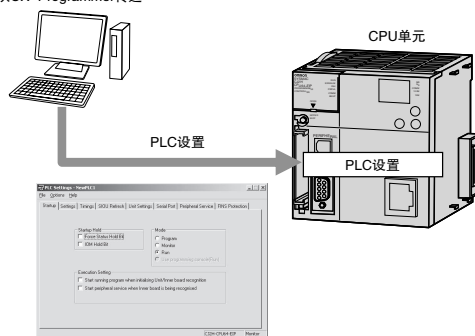
RS-232C 端口：上位链接

固定服务时间：循环时间的 10%

若要使用除这些默认设定之外的规格，可使用 CX-Programmer 来改变 PLC 设置，然后将 PLC 设置传送到 CPU 单元。

⇒ 从 CX-Programmer 传送 PLC 设置

从CX-Programmer传送



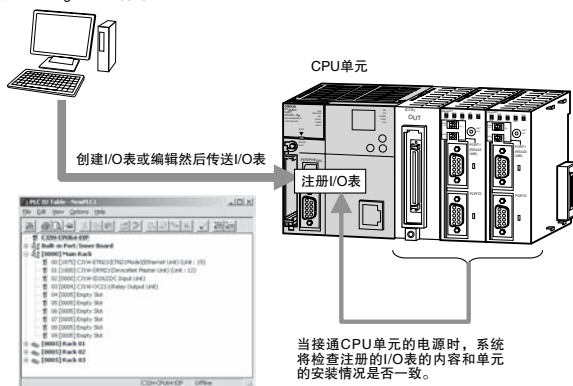
● I/O 表

当手动设定单元插槽或需要手动分配 CPU 单元中的 I/O 时，创建 I/O 表以检测错误的单元连接。默认情况下，每当电源接通时，CPU 单元 I/O 按照单元连接的顺序自动分配。

通过从 CX-Programmer 使用下述操作之一创建 I/O 表。

- 联接：执行创建相连单元的 I/O 表的步骤。
- 脱机：编辑 I/O 表 (安装有或未安装单元)，然后将表传送到 CPU 单元。

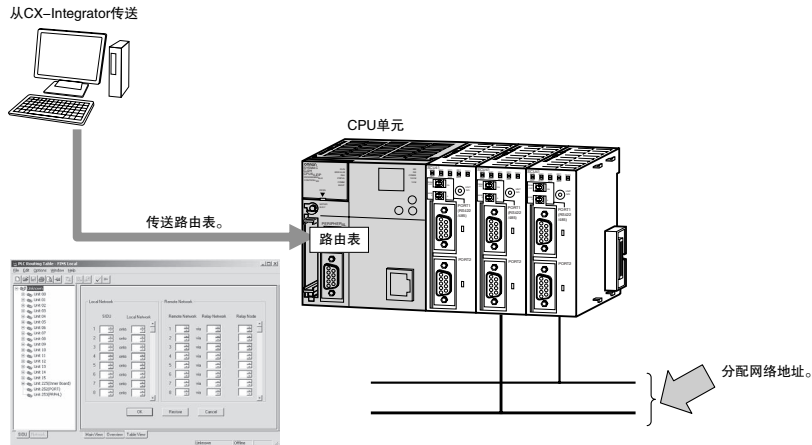
从CX-Programmer传送



● 路由表

如果有一个以上的 FINS 网络通信单元 (CPU 总线单元) 安装到 CPU 单元, 则必须通过 CX-Integrator 来创建路由表。在本地网络表中, 将为每个 FINS 网络通信单元分配一个网络地址。

通过 CX-Integrator 设定本地网络表, 然后将其传送到 CPU 单元。



● CPU 总线单元设置区

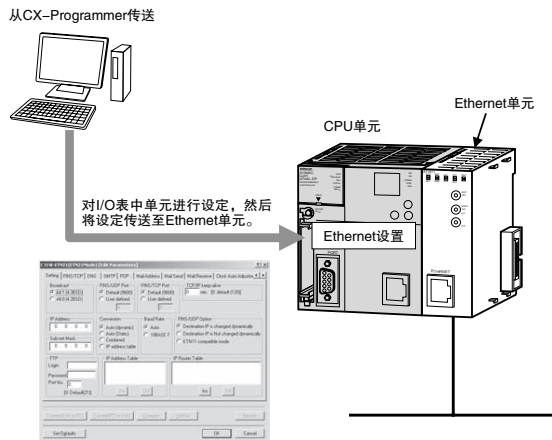
如果使用了特定的 CPU 总线单元, 例如 Ethernet 单元、Controller Link 单元或 FL-net 单元, 则必须对这些单元中的每个单元进行特定的设定, 然后将其传送到 CPU 单元。

例: 使用 Ethernet 单元

使用 CX-Programmer 进行 Ethernet 设定。

若要进行 Ethernet 设定, 用鼠标右击 CX-Programmer 的 I/O 表中的 Ethernet 单元, 然后选择“Edit CPU Bus Unit Settings” (编辑 CPU 总线单元设定)。

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



附加信息

如果不将用户设定的数据链接表用于 Controller Link 单元, 则设定数据链接表, 然后将其保存到 CPU 单元中。

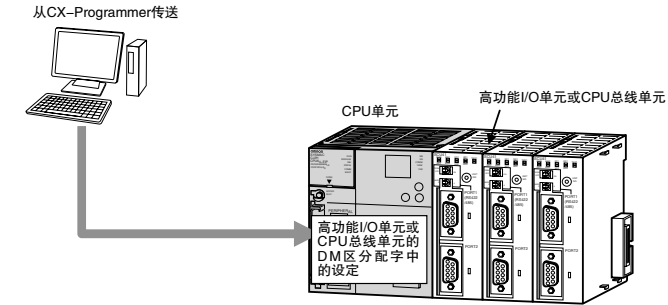
● 高性能 I/O 单元和 CPU 总线单元的 DM 区字分配

如果使用高性能 I/O 单元或 CPU 总线单元，则对在 DM 区分配字进行设定，然后将设定传送到 CPU 单元。

采用下述方法之一，通过 CX-Programmer 来设定 DM 分配区字。

- 用鼠标右击 I/O 表中的各高性能 I/O 单元或 CPU 总线单元，然后选择“Edit CPU Bus Unit Settings”（编辑 CPU 总线单元设定），以设定 DM 区分配字。
- 设定 PLC 存储器中的 DM 区分配字。

通过 CX-Programmer 为各高性能 I/O 单元或 CPU 总线单元设定 DM 区分配字，然后将设定传送到 CPU 单元。

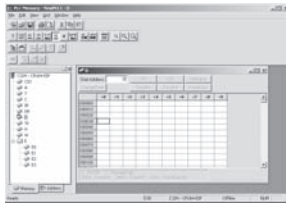


对I/O表中单元进行设定，然后将设定传送到CPU单元。



或

对PLC存储器中的DM区分配字进行设定。



4-2 PLC 设置

PLC 设置中包含 CPU 单元的基本设定。如果要将 CJ2 CPU 单元以非默认规格使用，则必须更改 PLC 设置中的参数。PLC 设置中的参数通过 CX-Programmer 进行设定。

有关 PLC 设置的详情，请参考“第 9 章 PLC 设置”。

4-3 创建 I/O 表

仅在下述情况下必须创建 I/O 表。I/O 表通过 CX-Programmer 创建。

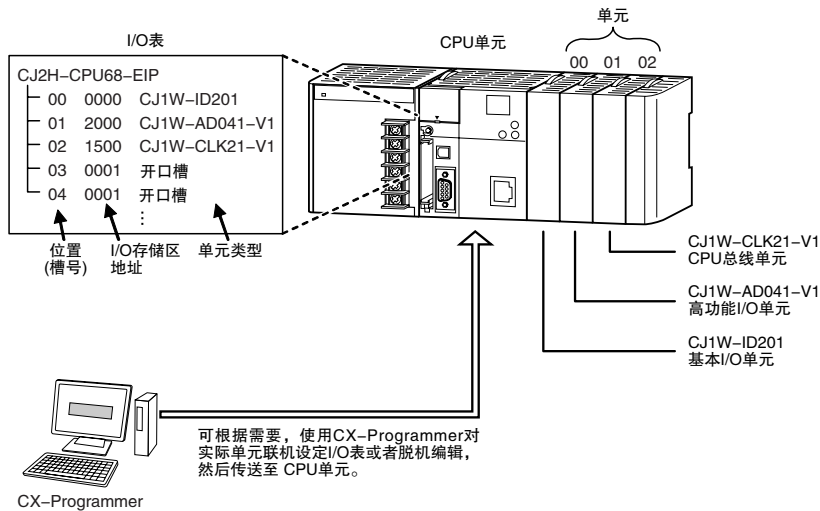
- 检查是否存在错误连接
- 保留字

如果无需检查是否存在错误连接并且可接受自动分配，则不必创建 I/O 表。

4-3-1 I/O 表

连接到 CPU 单元的单元的类型和位置在 I/O 表中注册。如果创建了 I/O 表，则在 CPU 单元的电源接通时，系统将检查实际连接到 CPU 上的单元的类型和位置是否与 I/O 表中注册的数据一致。

当 CX-Programmer 联机时，可基于实际连接的单元来设定 I/O 表；当 CX-Programmer 脱机时，可手动设定 I/O 表，然后将其传送到 CPU 单元。



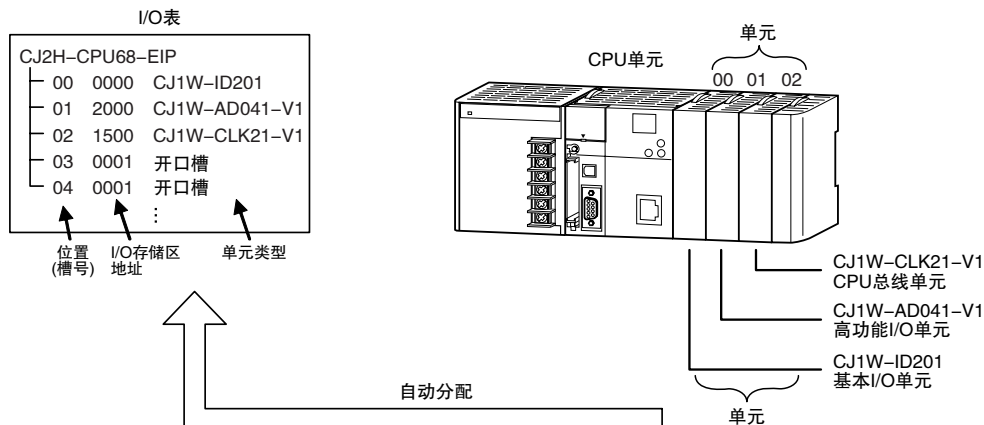
CPU 单元将根据该数据自动验证所连接的单元 (实际的 I/O 表)。

- 如果不一致，则将产生 I/O 设定错误，且不启动运行。

4-3-2 自动分配

采用自动分配时，I/O 表不由用户创建。每当电源接通时，均会基于单元的位置来将 I/O 存储器分配给各单元。所分配的存储器用于与单元进行数据交换。

不存在注册的 I/O 表，且系统将不检查单元配置。



4-3-3 手动分配

采用手动分配时，由用户创建称为“注册 I/O 表”的 I/O 表。

可通过从 CX-Programmer 使用下述方法之一来创建 I/O 表。

- 基于实际的单元配置来创建 I/O 表
采用该方法时，将在 CX-Programmer 联机的情况下，基于实际安装单元的配置来自动创建 I/O 表。
- 手动编辑 I/O 表，然后将其传送到 CPU 单元
采用该方法时，将使用 CX-Programmer 将各单元注册到所需位置的方法来手动脱机创建 I/O 表，然后将创建好的 I/O 表传送到 CPU 单元。

无论采用哪种方法，如果 PLC 接通电源时所连接的单元的配置与 I/O 表中的单元配置不同，则将产生 I/O 设定错误 (A401.10 将变 ON) 且运行将不启动。

有关 I/O 分配方法的详情，请参考“8-1 I/O 分配”。有关创建 I/O 表的详情，请参考“8-2 设定 CPU 总线单元和高性能 I/O 单元”。

4-4 设定路由表

如果 PLC 上安装了一个以上的 FINS 网络通信单元并且需要执行下述操作时,则必须通过 CX-Integrator 来设定路由表。

- 可通过支持软件或梯形图程序中的一条指令来切换正在访问的网络。
- 通信将跨网络层来执行。

4-4-1 路由表

在 OMRON 网络通信单元 (CPU 总线单元) 之间使用一种称为 FINS 的通信方法。但传送路径不在 FINS 协议中指定。路由表定义执行 FINS 网络通信时的网络地址设定和所需的通信路径。

适用单元

下述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 网络通信的 EtherNet/IP 单元、DeviceNet 单元和 FL-net 单元
- 当用于 FINS 网络通信时, CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 上的 EtherNet/IP 端口
- 当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

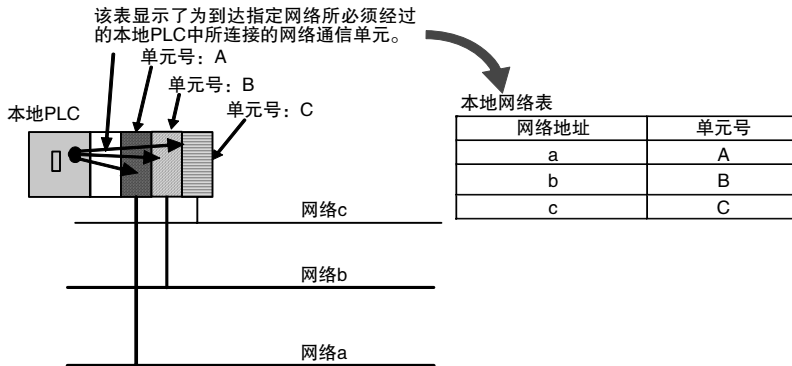
被归类为高功能 I/O 单元的网络通信单元 (例如 CompoNet 主站单元) 无需在路由表中设定。

组成路由表的表

路由表由以下两个表组成。

● 本地网络表

为本地 PLC 中连接的网络通信单元 (CPU 总线单元) 设定从 1 到 127 的网络地址。本地网络表由 CPU 单元使用, 用于在 PLC 上安装了一个以上的网络通信单元时识别安装到本地 PLC 上的网络通信单元以及对应的通信网络。

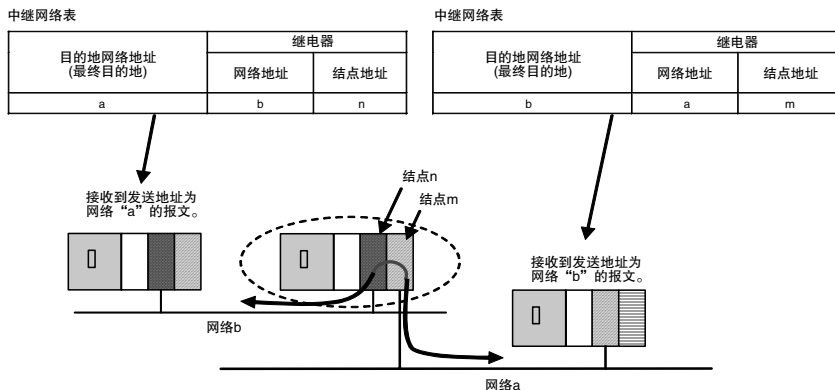


本地网络表给定连接在本地 PLC 中的网络通信单元的单元号和连接到该单元进行 FINS 通信的 FINS 网络的对应地址。

● 中继网络表

设定中继网络表用于跨网络层执行 FINS 通信。中继网络表规定使通信中继直至报文抵达最终目的地的方式。该表给定了从路线上的第一个中继点 (即: 要到达的第一个点) 到未连接到本地 PLC 的目的地网络 (最终网络) 的网络地址和结点地址。

通过在逐步途经各中继点即可到达目的地网络。



4-4-2 要求使用路由表的情况

根据应用程序的需要，在 CPU 单元中设定路由表（本地网络表和中继网络表）。



正确使用注意事项

要求使用本地网络表的情况

即使不执行中继，但如果 PLC 中连接了一个以上的网络通信单元，则要求使用本地网络表。

下表所示为根据网络使用情况而定，是否要求使用路由表。

网络使用条件	系统配置	路由表	
		本地网络表	中继网络表
PLC中未连接网络通信单元，或者仅连接了被归类为高功能 I/O 单元的网络通信单元（例如 CompoNet）。*1	<p>只有网络通信单元被归类为高功能 I/O 单元。</p> <p>本地 PLC</p> <p>CompoNet</p> <p>一个网络=本地网络 → 网络地址=0</p>	不需要	不需要
PLC中连接了一个以上的被归类为 CPU 总线单元的通信单元。	<p>必须指定将经过的网络通信单元。</p> <p>本地 PLC</p> <p>多个网络 → 设定一个以上的网络地址。</p>	网络中的所有 PLC 均要求 *2	如果将跨网络层执行通信，则要求。*3

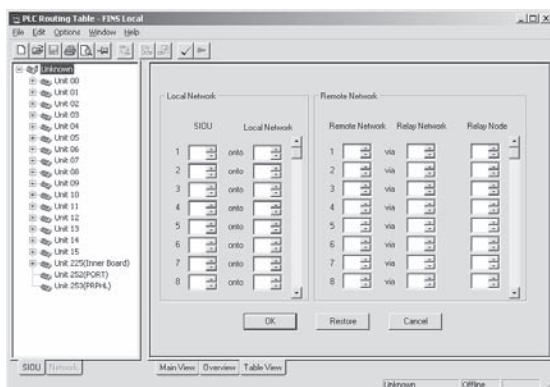
*1 如果将使用编程设备来访问网络中的结点，则将远程网络地址设定为 0。

*2 如果不存在本地网络表且使用编程设备或梯形图中的指令将网络地址设定为 0 来访问网络，则将在 FINS 通信中自动访问具有最小单元号的网络通信单元的网络。通过该方法，无需设定本地网络表，而只需将某网络的网络通信单元的单元号设定为网络中最小的单元号，即可访问该网络。

*3 如果不跨网络层执行通信，则无需设定中继网络表。

4-4-3 设定和传送路由表

1. 启动 CX-Integrator。
2. 从“Tools”（工具）菜单中选择“Start Routing Table”（启动路由表）。
3. 选择“FINS Local”（FINS 本地）。
4. 在“Table View”（表视图）选项页上编辑路由表。



5. 将 CX-Integrator 联机，然后从“Options”（选项）菜单中选择“Transfer to PLC”（传送到 PLC），从而将路由表传送到 CPU 单元。
6. 从“File”（文件）菜单中选择“Save Local Routing Table File”（保存本地路由表文件）。



正确使用注意事项

路由表数据文件

路由表存储在通过 CX-Integrator 创建的单个文件 (.rtg) 中，而非包含在 CX-Programmer 项目文件 (.cpx) 中。

4-5 为高性能 I/O 单元和 CPU 总线单元设定 DM 区分配字

如果使用高性能 I/O 单元或 CPU 总线单元，则必须进行这些设定。通过 CX-Programmer 来设定分配到高性能 I/O 单元和 CPU 总线单元的 DM 区字。

4-5-1 为高性能 I/O 单元和 CPU 总线单元设定 DM 区分配字

对高性能 I/O 单元和 CPU 总线单元所做的初始设定是将字分配到 CPU 单元中的 I/O 存储器的 DM 区。当电源接通时，将 DM 区分配字中的设定从 CPU 单元传送至高性能 I/O 单元和 CPU 总线单元。

DM 区分配字从下述地址进行分配。

● 高性能 I/O 单元的初始设定

根据在单元前面板上的旋转开关所设定的单元号，将为每个单元分配 D20000 ~ D29599 之间的一百个字。

● CPU 总线单元的初始设定

根据在单元前面板上的旋转开关所设定的单元号，将为每个单元分配 D30000 ~ D31599 之间的一百个字。

DM 区分配字的实际设定和传送时刻取决于高性能 I/O 单元和 CPU 总线单元和区的类型。详情请参考高性能 I/O 单元或 CPU 总线单元的操作手册。



附加信息

高性能 I/O 单元和 CPU 总线单元的设定也包含在特定单元 (例如 Ethernet 单元和 Controller Link 单元) 的 CPU 总线单元设置区中，并且数据和程序也存储在高性能 I/O 单元和 CPU 总线单元 (通过特定单元的支持软件创建) 中。

4-5-2 设定步骤

采用下述方法之一进行设定。

- 在 CX-Programmer 的 “PLC I/O Table” (PLC I/O 表) 窗口中选择单元，为高性能 I/O 单元和 CPU 总线单元进行设定，然后将设定传送至 CPU 单元。
- 在 CX-Programmer 的 “PLC Memory” (PLC 存储器) 窗口的 DM 区分配字中为各地址分别设定数据，然后将设定传送至 CPU 单元。
- 使用 MOV(021) 或用户程序中的其它指令来为 DM 区分配字中的各地址分别设定数据。

有关通过 CX-Programmer 将 DM 区字分配至 CPU 总线单元和高性能 I/O 单元的步骤，请参考 “8-2 设定 CPU 总线单元和高性能 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

设置区存储安装到 CPU 单元上的特定 CPU 总线单元的设置。下述三类单元和设定使用该区域。

- Ethernet 单元: Ethernet 设定
- Controller Link 单元: 数据链接表 (使用用户设定的表时)
- FL-net 单元: FL-net 设定

4-6-2 设定步骤

使用下述支持软件进行设定。

数据	支持软件
Ethernet 单元设定	对 CX-Programmer 的 I/O 表中的高功能 I/O 单元或 CPU 总线单元进行设定, 或者使用 HTML 进行设定。
Controller Link 单元数据链接表	使用 CX-Integrator 中的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

可用于 CPU 总线单元设置区的存储器的最大容量为 10,752 字节。设计系统时, 应使用于 CPU 总线单元设置区的存储器大小在由 PLC 中的 CPU 总线单元的组合所决定的限制范围内。如果超出限制范围, 则某些单元可能仅在默认设定下运行, 而某些单元可能不运行。

● 用于 CPU 总线单元设置区的存储器

单元	型号	使用的大小 (字节)
Controller Link 单元	CJ1W-CLK23	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	
	CJ1W-SCU22/32/42	
DeviceNet 单元	CJ1W-DRM21	
位置控制单元 (带 MECHATROLINK-II 通信模块)	CJ1W-NCF71(-MA)	
兼容 EtherCAT 协议的位置控制单元	CJ1W-NC281/481/881/F81	
	CJ1W-NC482/882	
运动控制单元	CJ1W-MCH71	
模拟量输入单元	CJ1W-ADG41	
存储和处理单元	CJ1W-SPU01-V2	

使用 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 单元编程的基本内容。

5-1 编程	5-3
5-1-1 编程概要	5-3
5-1-2 基本梯形图概念	5-6
5-1-3 ST 语言	5-8
5-1-4 SFC 概要	5-9
5-2 任务	5-11
5-2-1 任务概要	5-11
5-2-2 循环任务	5-14
5-2-3 中断任务	5-20
5-2-4 指定任务	5-30
5-3 段	5-40
5-3-1 程序段概要	5-40
5-4 功能块	5-42
5-4-1 功能块	5-42
5-4-2 功能块的特点	5-43
5-4-3 功能块规格	5-44
5-5 符号	5-47
5-5-1 概述	5-47
5-5-2 符号的类型	5-48
5-5-3 全局符号	5-50
5-5-4 局部符号	5-50
5-5-5 网络符号 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)	5-51
5-5-6 功能块中的变量	5-55
5-5-7 符号数据类型	5-56
5-5-8 将地址自动分配到符号	5-61
5-6 指令	5-62
5-6-1 指令的基本理解	5-62
5-6-2 指定操作数	5-69
5-6-3 数据格式	5-77
5-6-4 I/O 刷新时序	5-81
5-7 变址寄存器	5-86
5-7-1 何谓变址寄存器?	5-86
5-7-2 使用变址寄存器	5-86
5-7-3 与变址寄存器相关的处理	5-91
5-7-4 监控变址寄存器	5-92
5-7-5 在多任务之间共享变址寄存器和数据寄存器	5-93

5-8 指定地址偏移	5-95
5-8-1 概述	5-95
5-8-2 地址偏移应用示例	5-97
5-9 检查程序	5-98
5-9-1 CX-Programmer 输入期间的错误	5-98
5-9-2 通过 CX-Programmer 检查程序	5-98
5-9-3 通过仿真器进行调试	5-99
5-9-4 程序执行检查	5-102
5-10 注意事项	5-105
5-10-1 条件标志	5-105
5-10-2 特殊程序段	5-110

5-1 编程

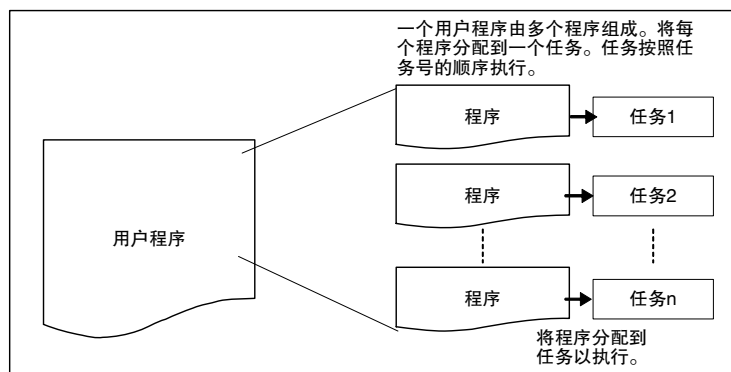
5-1-1 编程概要

用户程序结构

用户程序通过 CX-programmer 创建。程序由以下部分组成。

- 程序
一个完整的用户程序由多个程序组成。每个程序均以 END(001) 指令结束。
- 任务 (最小可执行单位)
将程序指定给任务以执行该程序。(在 CX-Programmer 中, 任务号在属性中作为程序属性进行指定。)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 (在指定的时间间隔内执行) 和电源 OFF 中断任务 (电源中断时执行)。

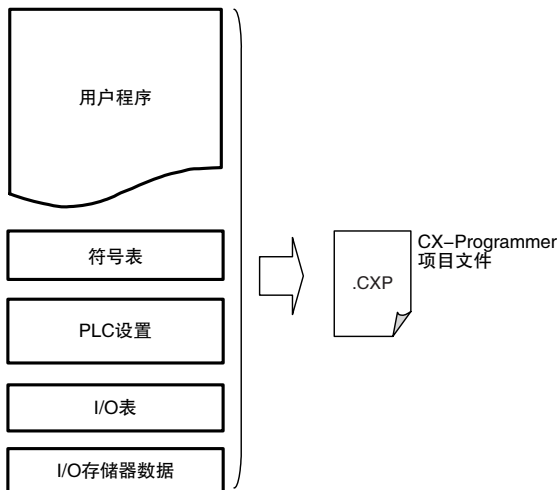
CPU单元



- 程序段
当通过 CX-Programmer 创建并显示程序时, 一个程序可分为任意个部分。每个部分称为一个程序段。程序段的目的是使程序便于理解。
- 子程序和功能块
在一个程序中, 可创建子程序和功能块。

用户程序数据

一个完成的程序在以 CX-Programmer 项目文件 (.CXP) 的形式与其它参数 (例如符号表、PLC 设置数据、I/O 表和 I/O 存储器数据) 一起保存。

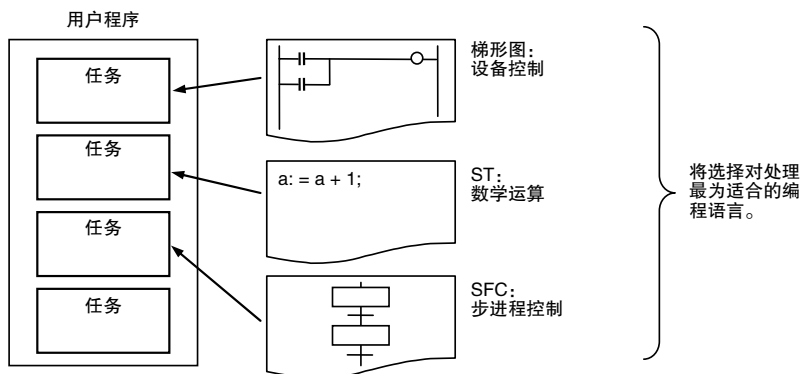


编程语言

可使用下述三种编程语言之一编写程序。

- 梯形图
- 结构化文本 (ST)
- 顺序功能图 (SFC)

将用梯形图、ST 或 SFC 编写的程序指定到任务，以便执行这些程序。



可为不同目的使用不同的语言，从而对各个过程使用最佳语言，例如为设备控制使用梯形图编程语言，为算术处理使用 ST 语言。

各语言可在下述位置使用。

语言	位置	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		允许	允许	允许
ST		允许	允许	允许
SFC		允许	---	---

可对各语言使用下述寻址方法。

语言	寻址方法	物理地址	符号
梯形图程序		允许	允许
ST		根据情况而定 *1	允许
SFC		根据情况而定 *2	根据情况而定 *2

*1 可通过将物理地址指定到符号的方法来使用物理地址。

*2 物理地址和符号用于 SFC 图中的布尔运算和转换。

程序容量

CJ2 CPU 单元为所有用户程序提供的最大程序容量 (即: 所有任务的总容量) 如下表所示。

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	
CJ2M-CPU□5	60K 步	
CJ2M-CPU□4	30K 步	
CJ2M-CPU□3	20K 步	
CJ2M-CPU□2	10K 步	
CJ2M-CPU□1	5K 步	

可通过在 CX-Programmer 中选择 “View - Memory View” (查看 - 查看存储器) 来查看程序容量。

梯形图指令的长度取决于所用的具体指令与操作数。详情请参考 “A-2 指令执行时间和步数”。

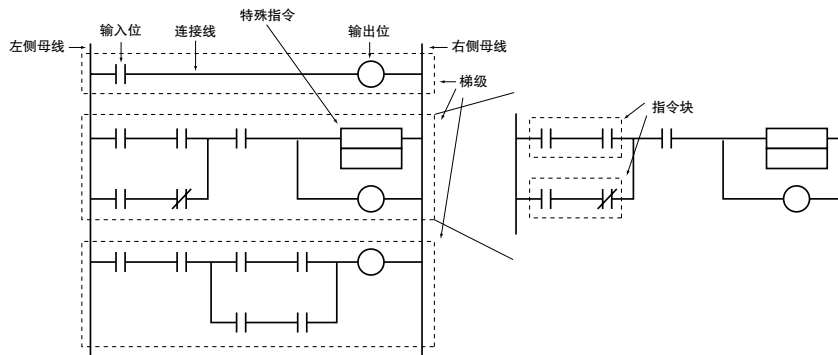
有关用于功能块的存储器容量, 请参考 “5-4-3 功能块规格”。

5-1-2 基本梯形图概念

梯形图逻辑是用于 PLC 的基本语言，其编写形式类似于电路图。指令按照其在存储器中的顺序（助记符顺序）执行。正确理解基本编程概念和执行顺序非常重要。

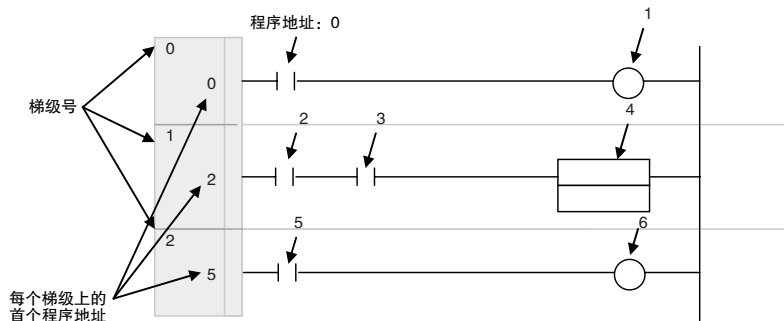
● 梯形图的通用结构

梯形图由左侧和右侧母线、连接线、输入位、输出位和特殊指令组成。程序由一个或多个程序梯级组成。程序梯级是在水平分离母线时可分离的单位。在助记符形式下，梯级是从 LD/LD NOT 指令到紧跟下一条 LD/LD NOT 指令之前的输出指令的所有指令。程序梯级由表示逻辑开始的 LD/LD NOT 指令开头的指令块组成。



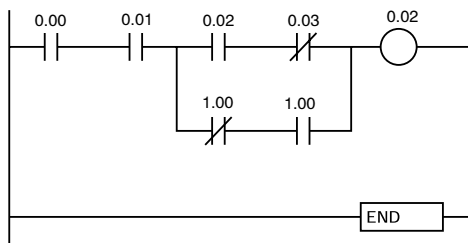
附加信息

使用 CX-Programmer 时，梯级号和各梯级上的第一个程序地址显示在各梯级的左边。



● 助记符

带程序地址，一个程序地址相当于一条指令。程序地址由从 0 开始的六个数位组成。



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(结构化文本)语言是根据 IEC 61131-3 标准设计的用于工业控制(主要是 PLC)的高级语言代码。标准控制语句、运算符和功能使得 ST 语言成为在梯形图编程中难以编写的算术处理的理想语言。

ST 语言不支持可在梯形图中编写的所有处理。支持符合 IEC 61131-3 标准的 ST 语言。



附加信息

有关 ST 编程规范、符号和输入步骤的详情，请参考《CX-Programmer 操作手册：功能块和结构化文本》(样本编号：W447)。

● ST 编程的特点

- 提供了大量控制语句，例如循环语句和 IF-THEN-ELSE 语句，还提供了大量算术运算符、比较运算符和 AND/OR 运算符以及大量数学函数、字符串抽取和合并函数、存储卡处理功能、字符串传送功能和三角函数。
- 可像高级语言(例如 C 语言)一样编写程序，并且还可包含注释，以增加程序的可读性。
- ST 程序可像普通程序一样上传和下载，但 ST 程序任务无法以任务单位上传和下载。
- ST 程序中可调用功能块(梯形图或 ST 语言)。
- 支持一维数组变量，以便在应用程序中提供更为方便的数据处理。

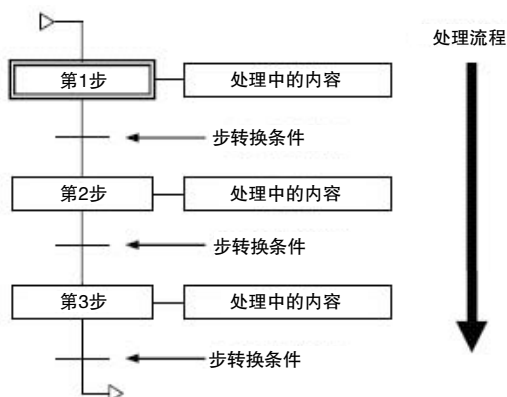
● 规格

项目	规格
ST 程序单位	<ul style="list-style-type: none"> · 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	地址通过符号来指定。可将任何地址分配到符号。
强制置位和强制复位	支持
联机编辑	ST 程序编辑

5-1-4 SFC 概要

SFC(顺序功能图)语言是一种图形编程语言,开发用于步进程序的说明,步进程序主要用于控制顺序处理。

SFC 通过其与步流程之间的图形关系以及对步进和各步中动作的条件描述,允许用户对顺序处理的控制进行编程。



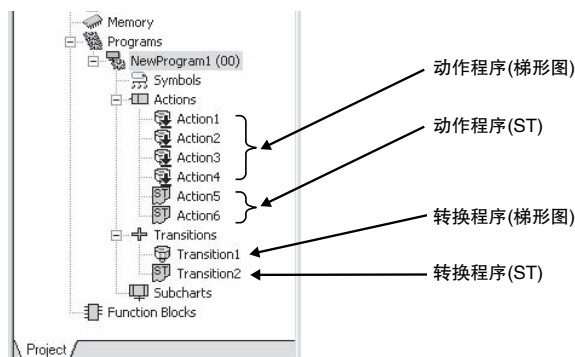
附加信息

有关 SFC 编程规范、符号和输入步骤的详情,请参考《CX-Programmer 操作手册: SFC 编程》(样本编号: W469)。

● SFC 编程的特点

梯形图和 ST 作为 SFC 程序的一部分

SFC 程序内部的步动作和转换条件可编码到梯形图或 ST 中。



- SFC 图和动作 / 转换程序的同步显示 / 编辑

- 联机编辑和调试

SFC 图可联机编辑。此外，可在调试时隐藏动作块，并可检查步进程状态。即使在动作块隐藏的同时，SFC 图编辑器也将在程序视图中显示动作程序。

- 规格

项目	规格
SFC 程序单位	任务 (1 个任务 = 1 个 SFC 图)
支持 SFC 程序分配的任务	循环或附加循环任务
SFC 元素	步、转换、动作、跳转、子图 注：SFC 元素将自动注册到局部变量中。
地址指定方法	可在 SFC 中为布尔运算或转换指定一个符号或物理地址。但如果使用 ST 编程，则无法指定物理地址。
强制置位和强制复位	可对转换进行联机强制置位 / 复位，从而逐步执行程序。
联机编辑	允许 SFC 编辑、动作添加 / 删除和转换添加 / 删除。

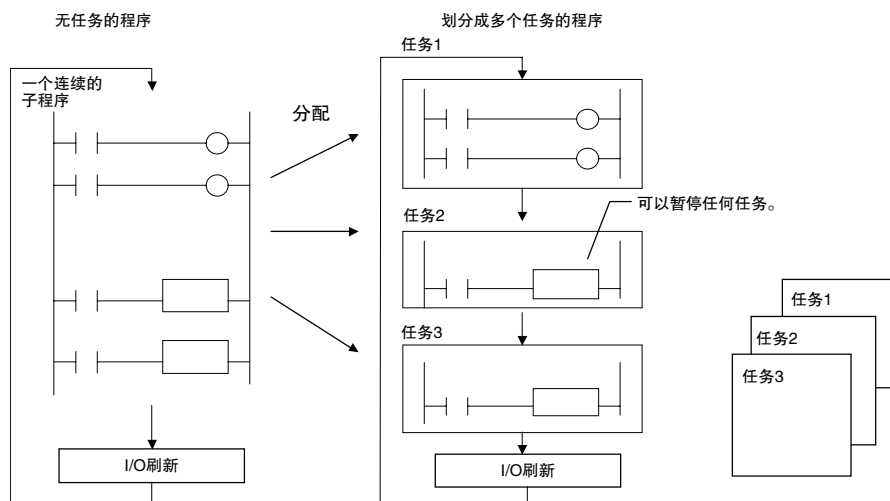
5-2 任务

5-2-1 任务概要

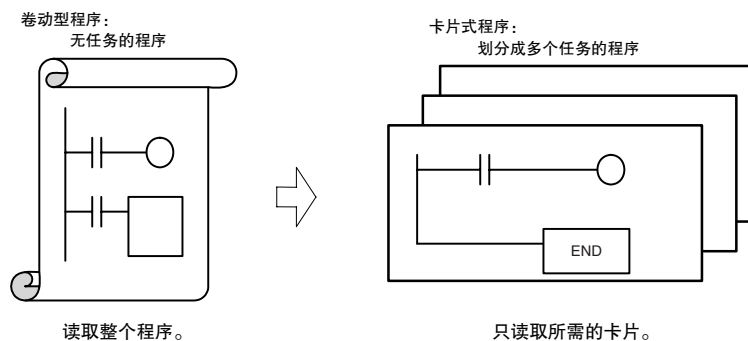
任务的概念

任务用于将一个程序划分成大量单位并指定执行各单位或编程的顺序。任一任务中的指令可用于允许或禁止其它任务的执行。

这意味着可将数个程序元件集结成不同任务，并且可根据当前产品型号或正在执行的处理，仅执行特定任务。该功能允许对不同处理在不同任务间进行切换。因此，性能（循环时间）大幅提高，因为将根据需要仅执行所需程序。



无任务的程序像是从头开始读取整个卷，而有任务的程序则像是单张读取卡。每张卡可允许或禁止其它卡。读取禁止的卡的操作将被跳过。

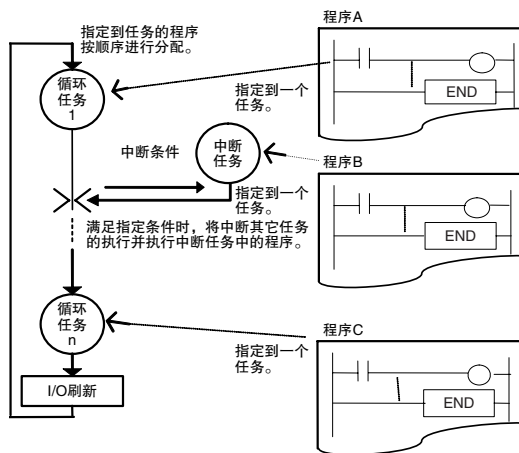


任务基本上分为两种类型。

任务类型	描述	适用编程语言	执行条件
循环任务	每个循环执行一次	梯形图 SFC ST	当采用下述方法之一将任务置于 READY 状态时执行。 <ul style="list-style-type: none"> 通过 CX-Programmer 来设定 “Activated at the start of operation” (在运行启动时激活) 属性 (程序属性设定) 通过任务控制指令使任务变 ON
中断任务 *1, *2	出现特定条件时执行。中断正在执行的进程。	梯形图 ST	当出现为某任务设定的中断条件时, 中断任务进入 READY 状态。可为下列每个中断任务设定条件。 <ul style="list-style-type: none"> 电源 OFF 中断任务 定时中断任务 I/O 中断任务 外部中断任务 输入中断任务 (仅限 CJ2M CPU 单元)

*1 通过使用任务控制指令将中断任务变 ON, 即可像循环任务一样对中断任务执行循环操作 (即: 每个循环执行一次)。(这些任务称为附加循环任务。)

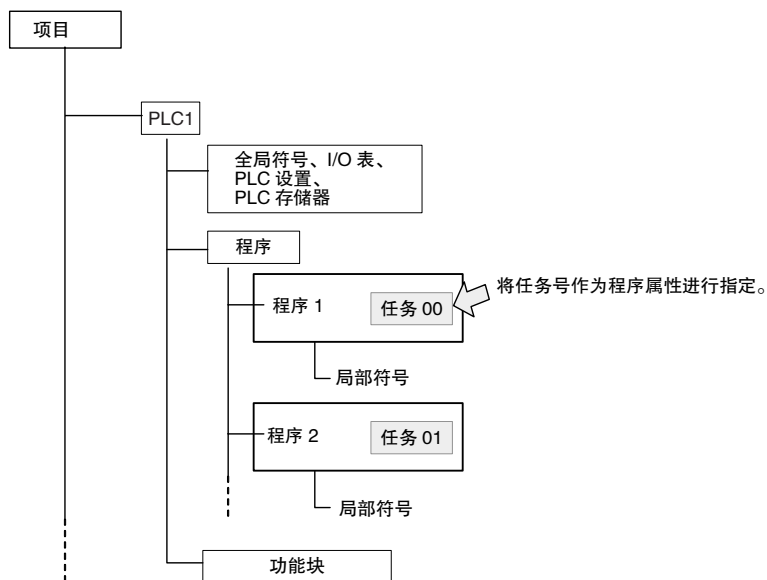
*2 请勿在中断任务中使用 SFC 程序。这些程序将无法正常运行。如果将中断任务作为附加循环任务执行, 则可使用 SFC 程序。



CX-Programmer 对任务的运行

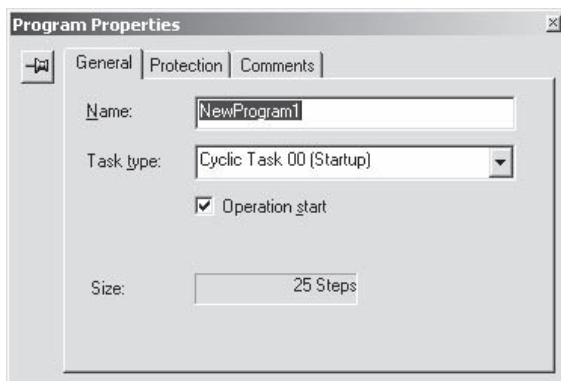
通过 CX-Programmer 将程序的任务号作为程序属性进行设定。

● 任务号的设定



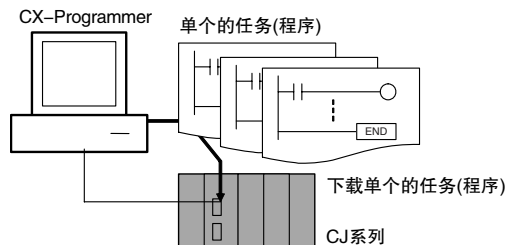
将任务类型和任务号作为各程序的属性进行指定。

1. 选择“View/Properties”（查看 / 属性），或者单击右键并从弹出菜单中选择“Properties”（属性），以显示将被分配到任务中的程序。
2. 选择“General”（通用）选项卡，然后选择“Task Type”（任务类型）和“Task No.”（任务号）。对于循环任务，如果想要在运行启动时开始执行任务，则选择“Operation start”（运行启动）复选框。



- 下载和上传单个任务

可从 CX-Programmer 分别上传和下载各个程序任务。



- 功能块使用的限制

无法为包含功能块的程序分别下载单个任务 (但可以上传)。

- 使用 SFC 程序和 ST 程序的限制

已指定到 SFC 程序或 ST 程序的任务无法分别上传或下载。

5-2-2 循环任务

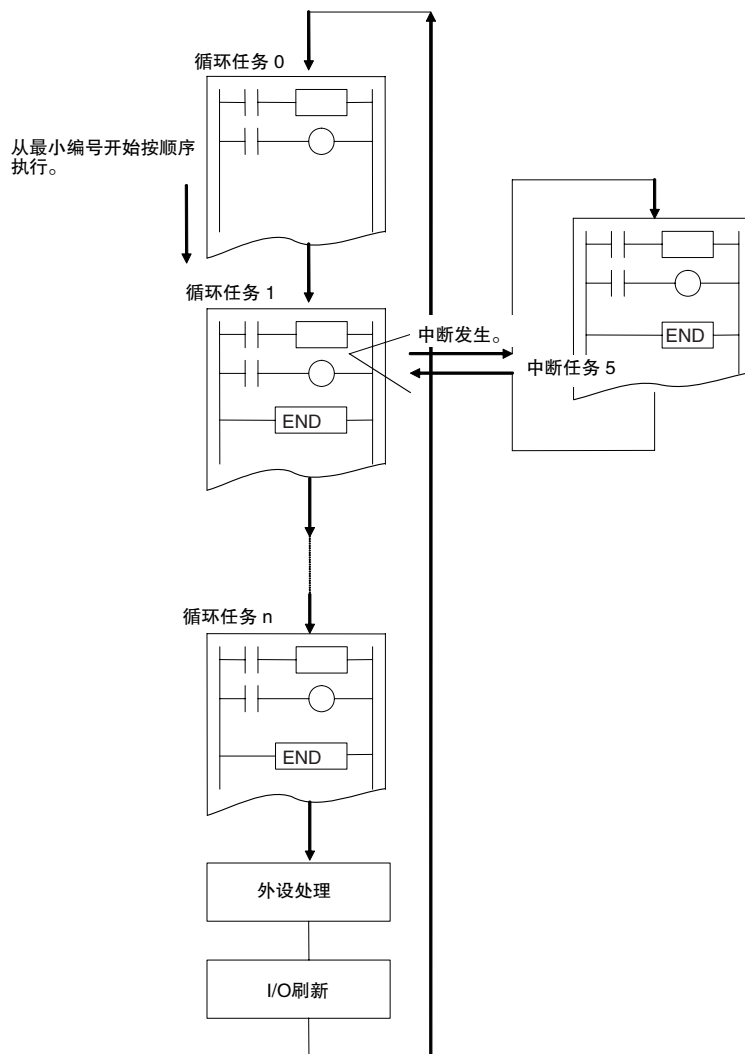
循环任务

循环任务从最小的任务号开始按顺序每个循环执行一次。最多可执行 128 个任务 (循环任务号 0 ~ 127)。通过使用 CX-Programmer 或任务控制指令来设定 “Activated at the start of operation” (在运行启动时激活) 属性, 即可启动任务。有关任务控制指令的说明, 请参考 “A-2 指令执行时间和步数”。如果使用附加循环任务, 则最多可使用 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。否则这些指令将无法正常运行。



附加信息

- 在任务开始时，所有条件标志 (ER、CY、等于标志、AER 等) 和指令条件将被清除。因此，无法在两个任务之间读取条件标志。
- 互锁 (例如 IL 和 ILC 指令)、跳转 (例如 JMP、CJP 和 JME 指令) 和子程序 (例如 SBS、RET 和 SBN 指令) 必须在各单独的任务中保持完整。例如，无法执行从一个任务到另一个任务的跳转。如果子程序将由一个以上的任务使用，请使用全局子程序 ((GSBS(750)、GRET(752) 或 GSBN(751) 指令)。

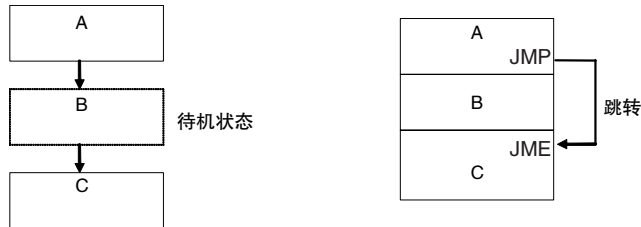
循环任务状态

● READY 状态

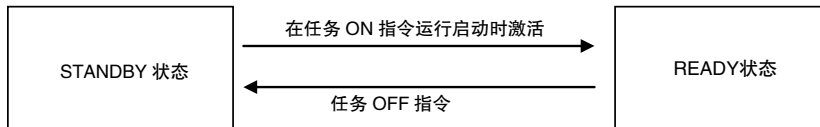
循环任务从最小的任务号开始按顺序执行。当运行模式从 PROGRAM 模式变为 RUN 或 MONITOR 模式时，设定了“Activated at the start of operation”（在运行启动时激活）属性的任务将可执行。可使用任务 ON 指令将状态从 STANDBY 变为 READY。如果使用了任务 OFF 指令来停止局部任务，则任务将无法突破任务 OFF 指令执行。

● STANDBY 状态

无法为处于 STANDBY 状态的任务执行指令。使用任务 OFF 指令可将状态从 READY 变为 STANDBY。通过将整个用户程序划分成多个任务，然后用任务控制指令来启动和停止任务，即可缩短循环时间。可根据整个程序的不同部分需要执行的时刻来将程序划分成任务。



● 状态转换



附加信息

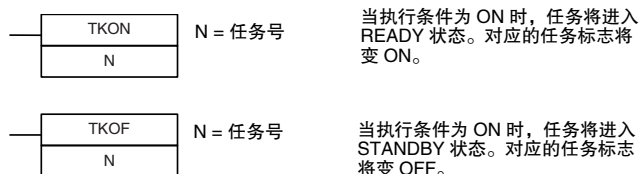
STANDBY 状态的作用与跳转 (JMP-JME) 指令完全相同。将保持来自 STANDBY 任务的输出状态。

使用循环任务

● 任务控制指令

使用任务控制指令从程序启动或停止循环任务。用于启动和停止任务的指令取决于编程语言，如下图所示。

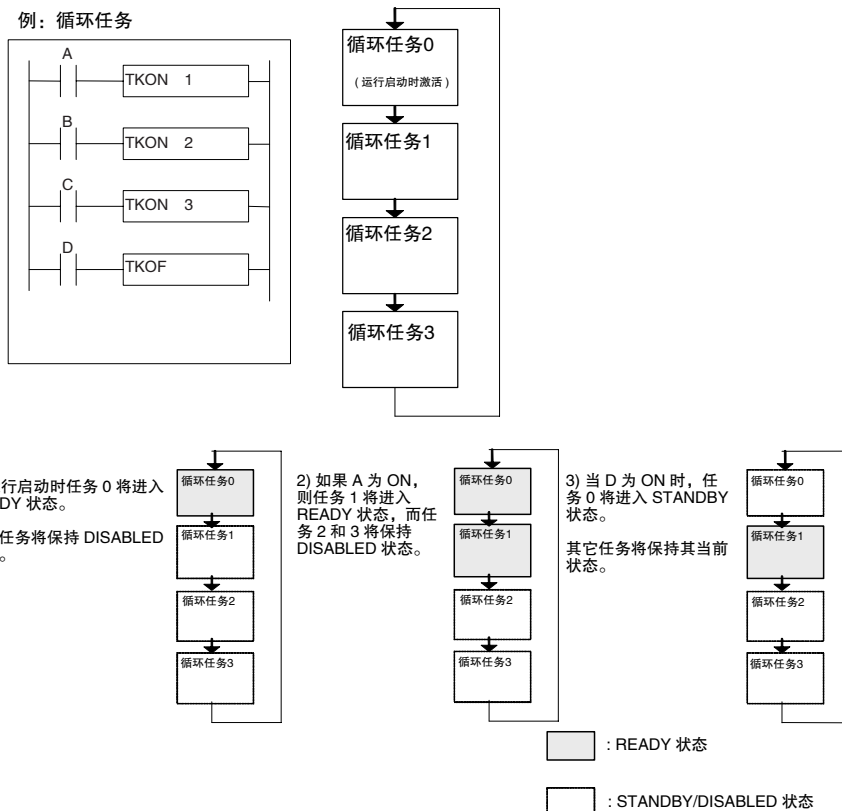
- 梯形图编程和 ST 编程：TKON(820)(任务 ON 指令) 和 TKOF(821)(任务 OFF 指令)
 - SFC 编程：SFCON(789)(SFC ON 指令) 和 SFCOFF(790)(SFC OFF 指令)
- 任务 ON 和任务 OFF 指令



任务 ON 和任务 OFF 指令可用于在任意时刻使任意循环任务在 READY 和 STANDBY 状态之间转变。处于 READY 状态的循环任务在后续循环中将保持该状态，而处于 STANDBY 状态的循环任务也将将在后续循环中保持该状态。任务 ON 和任务 OFF 指令仅可在循环任务中使用。这些指令无法在中断任务中使用。

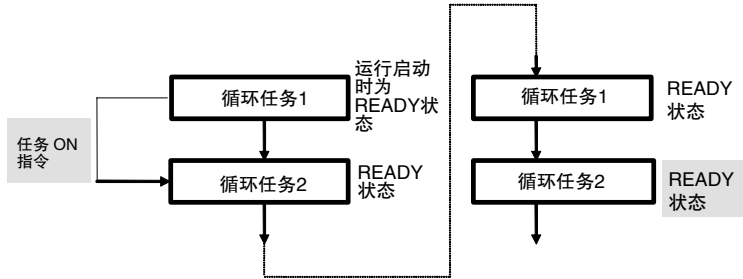
注 每个循环中必须至少有一个循环任务处于 READY 状态。如果无循环任务处于 READY 状态，则任务出错标志 (A295.12) 将变 ON，且 CPU 单元将停止运行。

例：循环任务

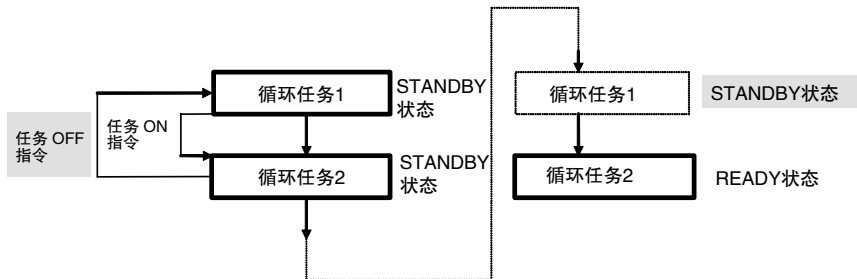


● 任务和执行循环

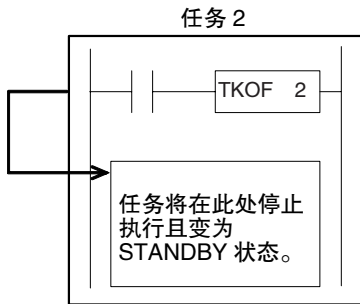
处于 READY 状态的循环任务 (包括附加循环任务) 将在后续循环中保持该状态。



处于 STANDBY 状态的循环任务也将在后续循环中保持该状态。若要将某个任务从 STANDBY 状态切换到 READY 状态, 则必须用任务 ON 指令使该任务变 ON。



如果为某个任务本身执行了任务 OFF 指令, 则该任务将在执行任务 OFF 指令的位置停止执行, 并且该任务将变为 STANDBY 状态。



循环任务号和执行循环

- 如果任务 m 使任务 n 变 ON 且 $m > n$, 则任务 n 将在下一循环进入 READY 状态。
例: 如果任务 5 使任务 2 变 ON, 则任务 2 将在下一循环进入 READY 状态。
- 如果任务 m 使任务 n 变 ON 且 $m < n$, 则任务 n 将在同一循环进入 READY 状态。
例: 如果任务 2 使任务 5 变 ON, 则任务 5 将在同一循环进入 READY 状态。
- 如果任务 m 将任务 n 置于 STANDBY 状态且 $m > n$, 则将在下一循环进入 STANDBY 状态。
例: 如果任务 5 使任务 2 进入 STANDBY 状态, 则任务 2 将在下一循环进入 STANDBY 状态。
- 如果任务 m 将任务 n 置于 STANDBY 状态且 $m < n$, 则任务 n 将在同一循环进入 STANDBY 状态。
例: 如果任务 2 使任务 5 进入 STANDBY 状态, 则任务 5 将在同一循环进入 STANDBY 状态。

任务与 I/O 存储器的关系

- I/O 存储器而非变址寄存器 (IR) 和数据寄存器 (DR) 由任务共享。例如, 循环任务 1 中使用的位 CIO 10.00 和循环任务 2 中使用的位 CIO 10.00 指的是存储器中的同一个位。如果 I/O 存储器而非变址寄存器 (IR) 和数据寄存器 (DR) 由一个以上的任务访问, 则某个在一个任务中会改变的值若用于另一个任务时, 请格外小心。
- 使用变址寄存器 (IR) 和数据寄存器 (DR) 有两种不同的方法: 1) 各任务间独立或 2) 由所有任务共享。
 - 在寄存器独立的情况下, 举例而言, 循环任务 1 所使用的 IRO 与循环任务 2 所使用的 IRO 不同。
 - 在寄存器共享的情况下, 举例而言, 循环任务 1 所使用的 IRO 与循环任务 2 所使用的 IRO 相同。

使寄存器是独立还是共享的设定通过 CX-Programmer 进行设定。

I/O 存储器	与任务的关系
CIO、辅助区、数据存储器 and 除 IR 和 DR 区之外的所有其它存储区。*1	与其它任务共享。
变址寄存器 (IR) 和数据寄存器 (DR)*2	对各任务独立使用。

*1 当前的 EM 存储区块也由各任务共享。因此, 举例而言, 如果当前的 EM 存储区块号随循环任务 1 而改变, 则新当前 EM 存储区块号也将对循环任务 2 有效。

*2 当中断任务启动时, 不设定 IR 和 DR 值。如果在某个中断任务中使用 IR 和 DR, 则必须在该中断任务内通过 MOVR/MOVRW(传送至寄存器和传送定时器 / 计数器的当前值 (PV) 至寄存器) 来设定这两个值。已执行中断任务后, IR 和 DR 将自动返回到其在中断任务执行之前的值。

5-2-3 中断任务

如果满足下述任何条件，则可在循环中的任意时刻执行中断任务。

如果产生中断，则无论 CPU 单元当前是否正在执行循环任务中的某条指令，均会在循环中的任意点执行中断任务。

中断任务的类型

可使用下列类型的中断任务。

- 电源 OFF 中断任务
电源 OFF 中断任务将在电源断开后立即执行。
- 定时中断任务
定时中断任务用于要求定时执行用户程序的特定部分 (例如以固定间隔监控运行状态) 的程序中。
- I/O 中断任务
当与 CPU 单元相连的中断输入单元的某个输入变 ON 时，将执行 I/O 中断任务。
- 外部中断任务
当高功能 I/O 单元或 CPU 总线单元申请中断时，将执行外部中断任务。
- 输入中断任务 (仅限 CJ2M CPU 单元)
当与 CJ2M CPU 单元相连的脉冲 I/O 模块的某个输入变 ON 时，将执行输入中断任务。
- 附加循环任务
附加循环任务是作为循环任务处理的中断任务。在这种情况下，目的并非进行中断处理，因为此时并非要执行其它中断任务，而是要增加循环任务的数量。最多可使用 128 个循环任务。但若使用附加循环任务，则最多可使用 384 个循环任务。

● 中断任务列表

任务	任务号	执行条件	相关设定
电源 OFF 中断任务	1	使 CPU 单元变 OFF	· PLC 设置的 OFF 中断任务
定时中断任务 0 和 1*1	2 和 3	CPU 单元内部定时器上设定的各个特定时间长度	· 使用 MSKS(690)(设定中断屏蔽)指令来设定中断间隔。(1 ~ 9999) · 请参考“PLC 设置”中的“定时中断时间单位”。(10ms/1.0ms/0.1ms)
I/O 中断任务 0 ~ 31*2	100 ~ 131	来自 CPU 机架上的中断输入单元的输入位变为 ON。	· 使用 MSKS(690)(设定中断屏蔽)指令来分配来自 CPU 机架上的中断输入单元的输入。
外部中断 0 ~ 255*3	0 ~ 255	当通过 CPU 机架 *3 上的高功能 I/O 单元或 CPU 总线单元发送请求时	无 (始终有效)
输入中断任务	140 ~ 147	当与 CJ2M CPU 单元相连的脉冲 I/O 模块的输入变 ON 或 OFF 时, 或者当增量计数器或减量计数器计数时。	· 使用 MSKS(690)(设定中断屏蔽)指令来解除对指定输入的中断屏蔽。
附加循环任务 0 ~ 255	0 ~ 255	当任务处于 READY 状态时的每个循环。(必须使用任务控制指令。)	无 (始终有效)

*1 如果正在使用同步单元操作, 则无法使用定时中断任务。

*2 必须将中断输入单元连接到 CPU 机架上。如果将 I/O 中断单元连接到其它任何位置, 则无法使用该单元申请执行 I/O 中断任务。

*3 必须将高功能 I/O 单元或 CPU 总线单元连接到 CPU 机架上。如果将上述单元连接到其它任何位置, 则无法使用这些单元来产生外部中断。



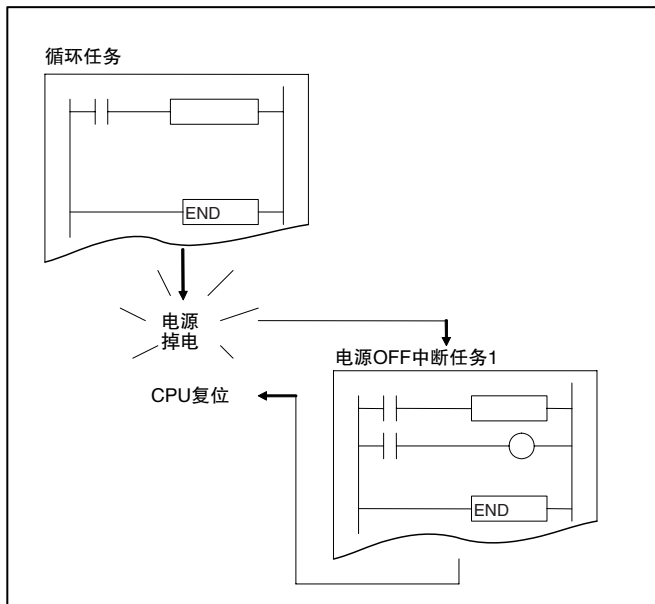
正确使用注意事项

请勿在中断任务中使用 SFC 程序。这些程序将无法正常执行。如果将中断任务作为附加循环任务执行, 则可使用 SFC 程序。

可在中断任务中使用 ST 程序。

电源 OFF 中断任务

电源中断时将执行该任务。电源中断时，电源单元将继续为 CPU 单元提供 10ms 的 5V 的电源，并在该期间执行电源 OFF 中断任务。(如果使用 CJ1W-PD022 电源单元，则电源将继续供应仅 1ms，因而无法使用电源 OFF 中断任务。)



● 电源 OFF 中断任务的中断开销时间

电源 OFF 中断任务在电源被确认为中断后的 0.1ms 内执行。

● 执行电源 OFF 中断任务的设定

在 CX-Programmer 的“PLC Settings” (PLC 设定) 对话框的“Timings/Synchronous” (定时 / 同步) 选项页中允许电源 OFF 中断任务。

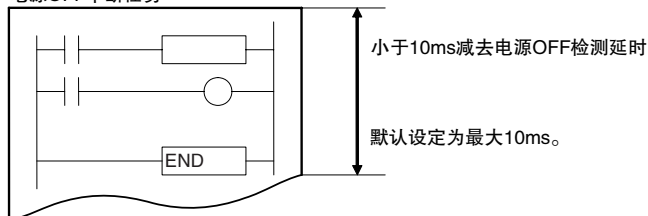
在循环任务的执行启动时，默认的 PLC 设置中禁止电源 OFF 中断任务。若要允许电源 OFF 中断任务，请在 PLC 设置中清除对电源 OFF 中断选项的选择。

在默认的 PLC 设置中，电源 OFF 中断任务将在 10ms 之后停止。电源 OFF 中断任务的执行时间必须小于 10ms。

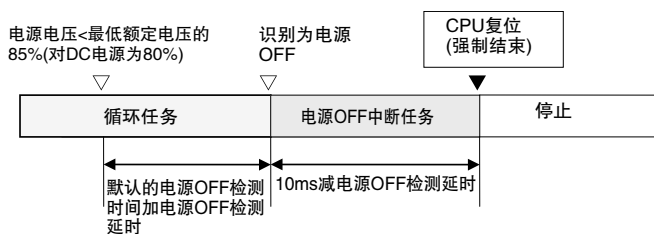
如果在 PLC 设置中设定了电源 OFF 检测延时，则电源 OFF 中断任务将在 10ms 减去 PLC 设置中的电源 OFF 检测延时设定之后停止。在这种情况下，电源 OFF 中断任务的执行时间必须小于 10ms 减去 PLC 设置中的电源 OFF 检测延时设定。

例：如果在 PLC 设置中将电源 OFF 检测延时设定为 4ms，则执行时间必须小于 10ms 减 4ms，即 6ms。

电源OFF中断任务



注 当电源降至最低额定电压的 85%(对于 DC 电源为 80%) 以下时, 将识别为电源 OFF 的情况, 并且在实际执行电源 OFF 中断任务之前的时间为默认的电源 OFF 检测时间 (对 AC 电源为 10 ~ 25ms, 对 DC 电源为 2 ~ 5ms) 加上 PLC 设置中的电源 OFF 检测延时 (0 ~ 10ms)。循环任务将执行该时间长度。



● 使用电源 OFF 中断任务的限制

联机编辑时不允许执行

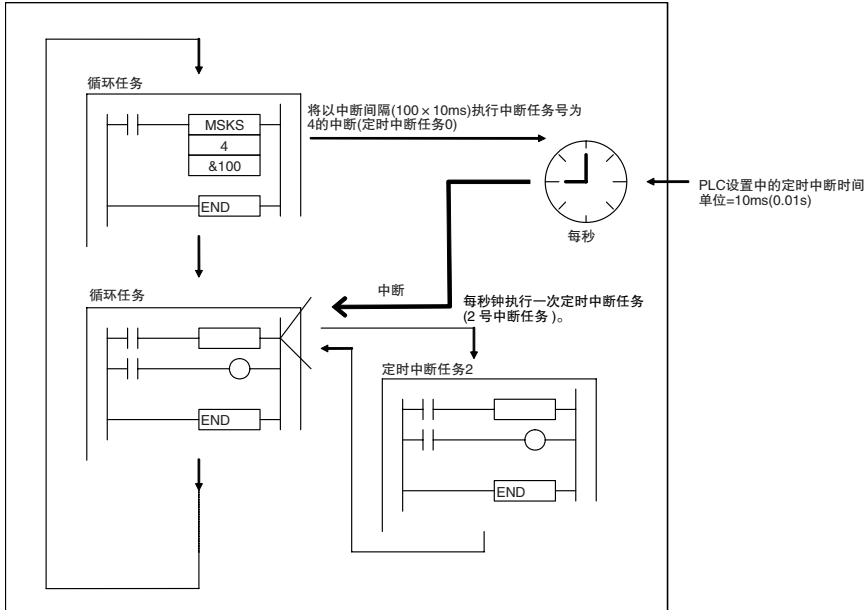
如果在联机编辑期间发生电源中断, 则电源 OFF 中断任务将无法执行。

电源 OFF 中断任务中无法使用的指令

除了那些在任何中断任务中均无法使用的指令之外, 还有一些指令无法在电源 OFF 中断任务中使用。详情请参考“5-2-4 指定任务”。

定时中断任务

通过使用 CPU 单元中的定时器，可使任务以指定的时间间隔执行。可为定时中断 0 和 1 执行最多两个中断任务（中断任务 2 和 3）。



● 中断任务号和中断号

	中断任务号	中断号 (MSKS(690))
定时中断 0	2	4
定时中断 1	3	5

● 执行定时中断任务

当 CPU 单元的运行启动时，定时中断任务将被禁止。可通过 MSKS(690)（设定中断屏蔽）指令来允许这些中断任务。定时中断任务的时间间隔由 PLC 设置中的定时中断时间单位的设定以及执行 MSKS(690) 指令时通过操作数指定的定时中断时间来决定。

· 设定定时中断时间单位

使用 CX-Programmer 的“PLC Settings”（PLC 设定）对话框中的“Timings/Synchronous”（定时/同步）选项页上的“Scheduled Interrupt Interval”（定时中断间隔）设定来设定由 MSKS(690) 指令所设定的中断时间。

可将时间单位设定为 10ms、1.0ms 或 0.1ms。默认设定为 10ms。

· 使用 MSKS(690) 指令来设定定时中断时间

设定定时中断任务的执行时间间隔。

可将时间单位设定为十进制的 &1 ~ &9999(#0001 ~ #270F Hex) 之间的值。

例：如果将定时中断时间单位设定为 0.1ms 并将定时中断时间设定为十进制的 &5，则时间间隔将为 $0.1 \times 5 = 0.5\text{ms}$ 。

可为定时中断任务设定的最小时间间隔如下：

- CJ2H CPU 单元：0.2ms
- CJ2M CPU 单元：0.4ms

然而，可对 1.1 版或更高版本的 CJ2H CPU 单元使用高速中断功能，从而将定时中断 0（中断任务 2）的中断间隔设定为 0.1ms。该设定不能用于其它中断。有关高速中断的详情，请参考“10-2-6 高速中断功能”。



正确使用注意事项

- 缩短定时中断的时间间隔并提高定时中断任务的执行频率时请务必小心，因为直到循环任务执行和附加循环任务完成的时间以及整体循环时间将延长。
- 如果正在使用同步单元操作，则无法使用定时中断任务。

● 定时中断任务的中断开销时间

从由 MSKS(690) 指令所设定的指定时间已过直到中断任务实际执行为止的时间称为定时中断启动时间。在已执行中断任务程序之后返回被中断的处理所需的时间称为循环任务返回时间。中断任务启动时间和循环任务返回时间的组合称为中断开销时间。

	项目	时间	
		CJ2H CPU 单元	CJ2M CPU 单元
定时中断任务的中断开销时间	中断任务启动时间	22 μ s 或 13 μ s*1 (对于 1.0 版单元为 27 μ s)	30 μ s
	循环任务返回时间	11 μ s 或 8 μ s*1 (对于 1.0 版单元为 15 μ s)	11 μ s

*1 使用高速中断功能

有关高速中断功能的说明，请参考“10-2-6 高速中断功能”。

● 通过 MSKS(690) 复位和重启

对于 CJ2M CPU 单元，可指定通过 MSKS(690) 指令在启动定时中断时使内部定时器复位（复位启动）。该功能允许生成一个直到启动第一个中断为止的一致时间而无需使用 CLI(691) 指令。启动定时中断时，通过 MSKS(690) 指令来设定定时中断时间（即：某个中断与其下一个中断之间的间隔）。然而，在执行 MSKS(690) 指令之后直到第一次启动中断任务为止的时间取决于内部定时器的当前值。因此，如果不执行 CLI(691) 指令，则到第一次启动中断任务为止的时间间隔将不一致。但对于 CJ2M CPU 单元，即使不执行 CLI(691) 指令，也可在启动时使内部定时器复位，从而使得到第一次启动中断任务为止的时间保持一致。

定时中断的 MSKS(690) 操作数

操作数	设定值
N(中断标识符)	4: 定时中断 0, 正常运行 (内部定时器不复位, 中断任务 2) 5: 定时中断 1, 正常运行 (内部定时器不复位, 中断任务 3) 14: 定时中断 0, 复位启动 (仅限 CJ2M CPU 单元, 中断任务 2) 15: 定时中断 1, 复位启动 (仅限 CJ2M CPU 单元, 中断任务 3)

● 通过 MSKR(692) 读取内部定时器的当前值 (PV)

对于 CJ2M CPU 单元，可读取用于测量定时中断时间的内部定时器的当前值。可读取直到定时中断启动为止或者自上一次定时中断以来已经过的时间。MSKR(692) 指令用于读取内部定时器的当前值。时间单位与用于定时中断时间（即：在 PLC 设置中设定的定时中断间隔）的单位相同。

定时中断的 MSKR(692) 操作数

操作数	设定值
N(中断标识符)	4: 读取中断任务 0(中断任务 2) 的定时中断时间 (设定值) 5: 读取中断任务 1(中断任务 3) 的定时中断时间 (设定值) 14: 读取中断任务 0 的内部定时器的当前值 (仅限 CJ2M CPU 单元, 中断任务 2) 15: 读取中断任务 1 的内部定时器的当前值 (仅限 CJ2M CPU 单元, 中断任务 3)

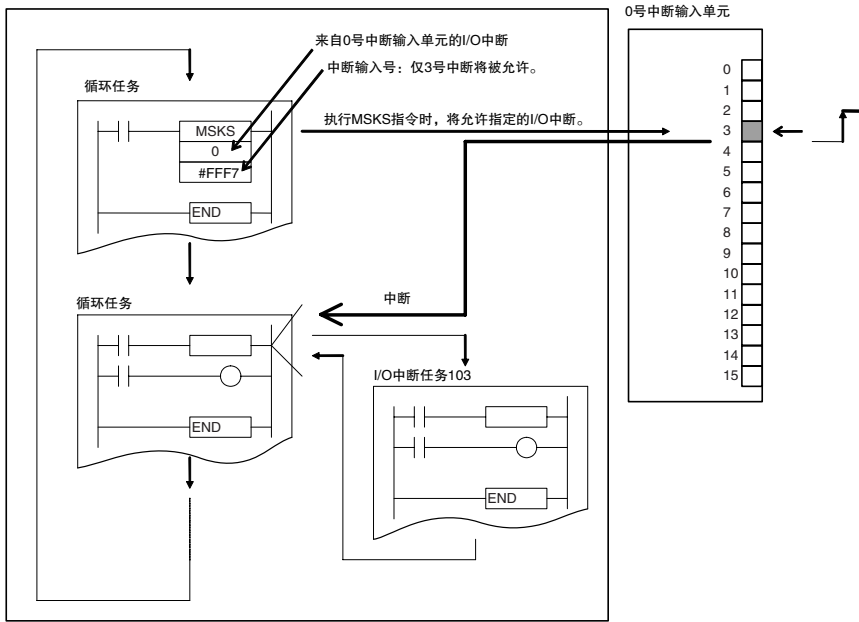
I/O 中断任务

当中断输入单元的某个输入变ON时, 将执行I/O中断任务。最多可创建32个任务(中断任务号100 ~ 131)。

● 执行中断任务

若要允许 I/O 中断任务, 可执行 MSKS(690)(设定中断屏蔽) 指令。

例: 下例所示为当0号中断输入单元的3号中断输入(单元0和1的最左位)为ON时, 执行I/O中断任务。



中断输入单元号、输入号和 I/O 中断任务号

中断输入单元号 *1	输入号	I/O 中断任务
0	0 ~ 15	100 ~ 115
1	0 ~ 15	116 ~ 131

*1 中断输入单元号从 CPU 单元开始按照从 0 到 1 的顺序排列。



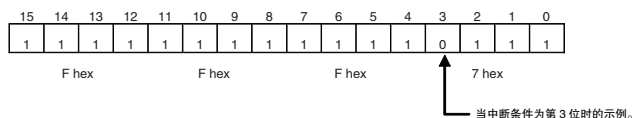
● MSKS(690)

若要允许 I/O 中断任务，可在循环任务的程序中执行 MSKS(690)(设定中断屏蔽) 指令，从而允许指定中断单元的指定中断号的中断。

MSKS(690) 指令的操作数 S(第二操作数):

FFF7 Hex 的各个位分别对应中断输入单元的中断输入。中断输入号 0 ~ 15 对应位 0 ~ 15。

例:



正确使用注意事项

不允许不需要的 I/O 中断任务。如果通过干扰触发了中断输入且无对应的中断任务号，则将产生致命错误 (任务错误) 并导致程序停止运行。

● I/O 中断任务的中断开销时间

从中断输入单元 (CJ1W-INT01) 的输入位变 ON(或 OFF) 直到 CPU 单元接收到中断信号为止的时间取决于硬件的响应时间。从接收到中断信号直到实际执行 I/O 中断任务为止的时间称为 I/O 中断任务的中断任务启动时间。在已执行中断任务程序之后返回中断之前的处理所需的时间称为循环任务返回时间。中断任务启动时间和循环任务返回时间的组合称为中断开销时间。

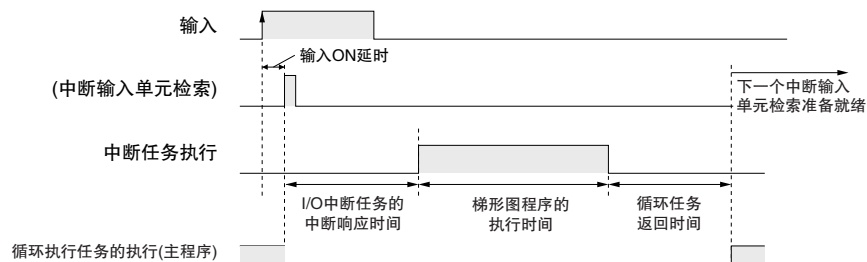
使用中断输入单元

项目	时间	
	CJ2H CPU 单元	CJ2M CPU 单元
硬件响应 (CJ1W-INT01)	上升沿微分: 0.05ms, 下降沿微分: 0.5ms	
中断开销处理时间	中断任务启动时间	26 μ s 或 17 μ s ^{*1} (对于 1.0 版单元为 30 μ s)
	循环任务返回时间	11 μ s 或 8 μ s ^{*1} (对于 1.0 版单元为 15 μ s)
		31 μ s
		10 μ s

*1 使用高速中断功能

有关高速中断功能的说明，请参考“10-2-6 高速中断功能”。

注 可在执行用户程序、I/O 刷新、外设服务或开销处理期间执行 I/O 中断任务。(即使正在执行某条指令，也可执行 I/O 中断任务。该指令将被中断。) 如果在执行某个中断任务期间发生输入中断，则不会立即执行该输入中断。而是首先执行完当前中断任务，然后在经过循环任务返回时间和中断任务启动时间之后再执行新的中断。

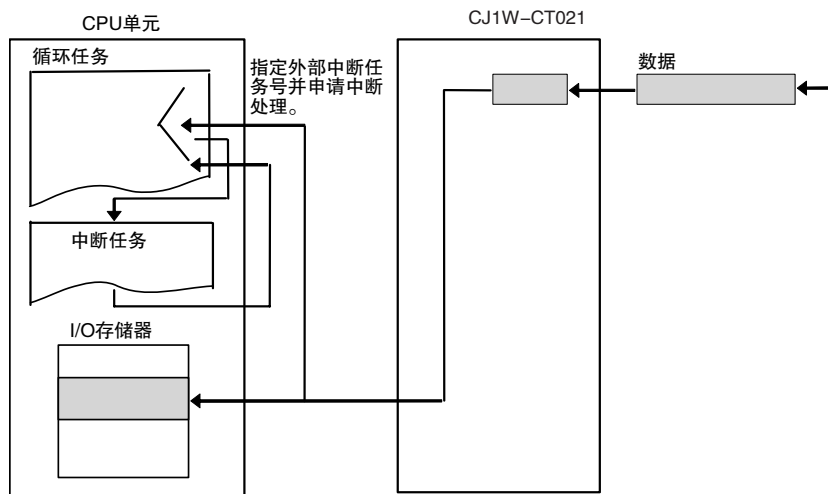


外部中断

将在包含高性能 I/O 单元或 CPU 总线单元的 PLC 中的 CPU 单元中执行中断处理。无需在 CPU 单元中进行设定，除非程序中包含外部特定任务号的中断任务。必须将高性能 I/O 单元或 CPU 总线单元连接到 CPU 机架上，以允许外部中断。

始终允许接收外部中断任务的操作。任务的最大数量为 256(中断任务号 0 ~ 255)。

例：来自 CJ1W-CT021 高速计数器单元的外部中断



若要就本单元除 I/O 刷新之外的 2 个外部控制输入和 32 个输出的状态信息与 CPU 单元进行通信，请将外部控制输入或输出设定为可触发外部中断任务。



正确使用注意事项

请勿对一个以上的中断任务使用同一个任务号。

如果外部中断任务 (0 ~ 255) 与电源 OFF 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) 具有相同的任务号，则该外部中断任务将在满足任一中断条件 (外部中断或其他中断条件) 时执行。

● 脉冲 I/O 模块的外部中断

当在 CJ2M CPU 单元上使用脉冲 I/O 模块时，可根据高速计数器输入的比较结果来执行外部中断任务。采用目标比较，可将比较条件 (目标值和计数方向) 随同对应的中断任务号一起注册到 CTBL(882) 比较表中。当高速计数器的当前值 (PV) 与注册的目标值一致时，将执行指定的中断任务。

采用范围比较，可将比较条件 (上、下限和进入或离开范围) 随同对应的中断任务号一起注册到 CTBL(882) 比较表中。满足比较条件时，将执行指定的中断任务。

● 外部中断任务的开销时间

对于外部中断任务，直到 CPU 单元接收到中断信号为止的时间取决于申请从 CPU 单元执行外部中断任务的单元或板的类型（即：是高功能 I/O 单元还是 CJ 系列 CPU 总线单元）以及中断申请的服务类型。详情请参考相关单元的手册。

在 CPU 单元接收到中断信号之后的中断开销时间与 I/O 中断任务相同。

输入中断任务（仅限 CJ2M CPU 单元）

当将与 CJ2M CPU 单元相连的脉冲 I/O 模块的输入用作中断输入时，则在输入变 ON 或 OFF 时（称为直接模式），或者当增量计数器或减量计数器计数时（称为计数器模式），将执行一个中断任务。对每个输入执行的中断任务将固定且无法改变（中断任务 140 ~ 147）。设定中断屏蔽指令（MSKS(690)）用于允许中断和将运行模式设定为之间模式或计数器模式。

附加循环任务

附加循环任务像普通的循环任务一样，可在每个循环中执行。在普通的循环任务执行完成后，从最小的任务号开始执行附加循环任务（中断任务号 0 ~ 255）。附加循环任务的最大数量为 256（中断任务号：0 ~ 255）。



正确使用注意事项

请勿对中断任务使用分配给附加循环任务的任务号。与电源 OFF 中断任务、定时中断任务或 I/O 中断任务的任务号相同的附加循环任务将兼作附加循环任务和中断任务执行。

- 注 1 可在附加循环任务中输入和执行 TKON(820) 与 TKOF(821) 指令，但在附加循环任务作为中断任务执行时，将不执行这两条指令。
- 2 普通循环任务和附加循环任务之间的区别如下表所示。

项目	附加循环任务	普通循环任务
启动时激活	不允许进行设定。	从 CX-Programmer 进行设定
任务标志	不支持	支持（循环任务号 0 ~ 127 与任务标志 TK000 ~ TK127 对应。）
初始任务执行标志 (A200.15) 和任务启动标志 (A200.14)	不支持	支持
变址 (IR) 和数据 (DR) 寄存器的值	启动任务时不定义（与普通中断任务相同）。在每个循环开始时不定义值。使用值之前始终对值进行设定。无法读取在上一个循环中设定的值。	在运行开始时不定义。可以读取在上一个循环中设定的值。

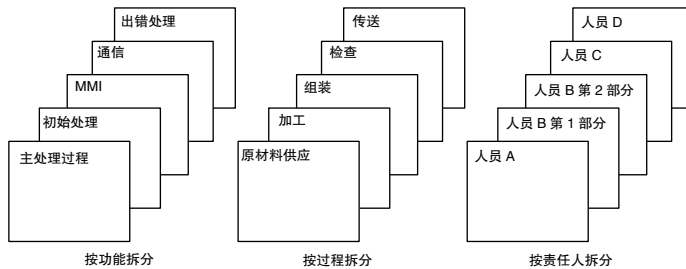
5-2-4 指定任务

准则

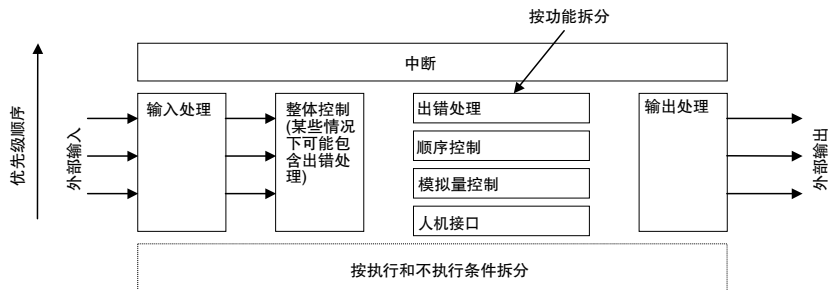
任务设计对于构建可靠性高和易于维护的系统至关重要。请注意下述要点。

- 请基于下述要点将一个程序划分为多个任务。
 - 充分了解状态转换，考虑执行和不执行的特定条件。
 - 考虑是否存在外部 I/O。
 - 考虑功能和责任人。
 - 考虑按优先级的顺序执行。将处理划分成循环任务和中断任务。为优先级最高的任务指定最小的任务号。

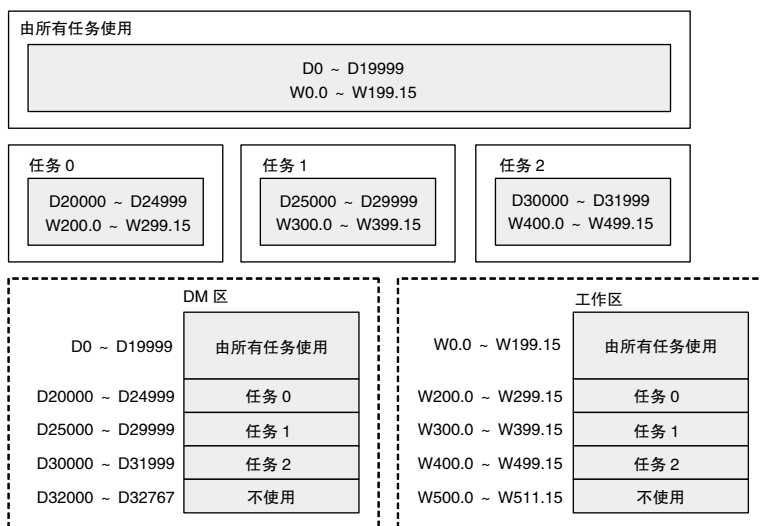
例 1：根据功能和责任人来设计任务。



例 2：根据执行的优先级来设计任务
将最小的任务号分配给控制任务而非处理任务。

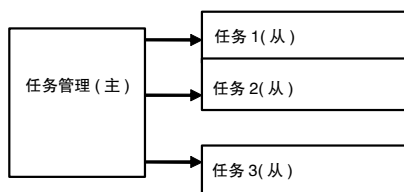


- 拆分和设计程序以确保自主性。将各任务之间的数据交换量保持在绝对最小值。
组织由所有任务使用的 I/O 存储器和仅由单个任务使用的 I/O 存储器。组织和分配仅由每个任务使用的 I/O 存储器范围。

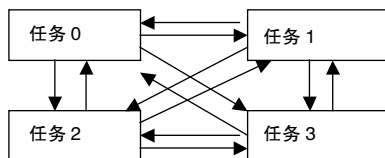


- 使用一个任务来启动和停止其它任务，从而控制整个用户程序。

正确例：定义任务之间的主 - 从关系。



错误例：如果以无组织的方式执行任务的启动和停止，则难以控制。



- 将较小的任务号分配给高优先级的中断任务。
- 处于 READY 状态的任务将在后续循环中执行，只要该任务不被自身或另一任务变为 STANDBY 状态。如果要在任务之间进行分支处理，请务必插入其它任务的 TKOF(821) (任务 OFF) 指令。
- 在执行条件中使用初始任务执行标志 (A200.15) 或任务启动标志 (A200.14) 以执行指令，从而对任务进行初始化。在每个任务的第一次执行期间，初始任务执行标志将为 ON。每当任务进入 READY 状态时，任务启动标志将变 ON。



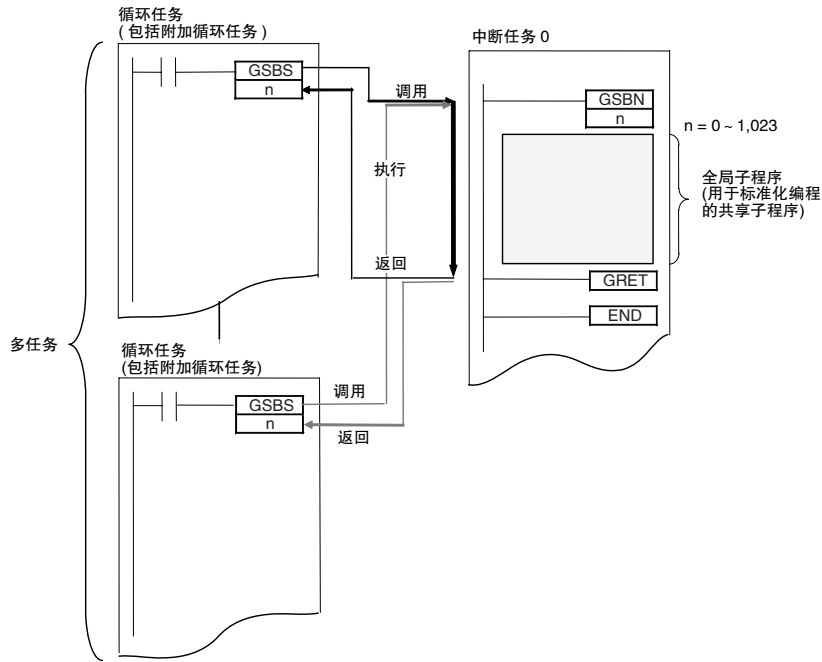
附加信息

全局子程序

对于常规的子程序指令，无法从一个任务调用另一个任务中的子程序。可以中断任务号 0 来创建全局子程序，并且可从任意循环任务（包括附加循环任务）调用这些子程序。

GSBS 指令用于调用全局子程序。子程序号必须介于 0 ~ 1,023 之间。全局子程序在中断任务号 0 的末尾处（紧跟 END(001) 指令之前）的 GSBN 和 GRET 指令之间进行定义。

全局子程序可用于创建标准程序段的库，这些程序段可在需要的时候随时调用。



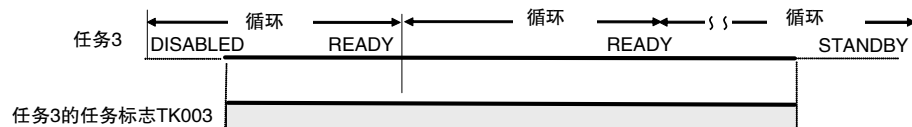
与任务相关的标志

● 与循环任务相关的标志

下述标志仅用于普通循环任务，不能用于附加循环任务。

任务标志 (TK000 ~ TK127)

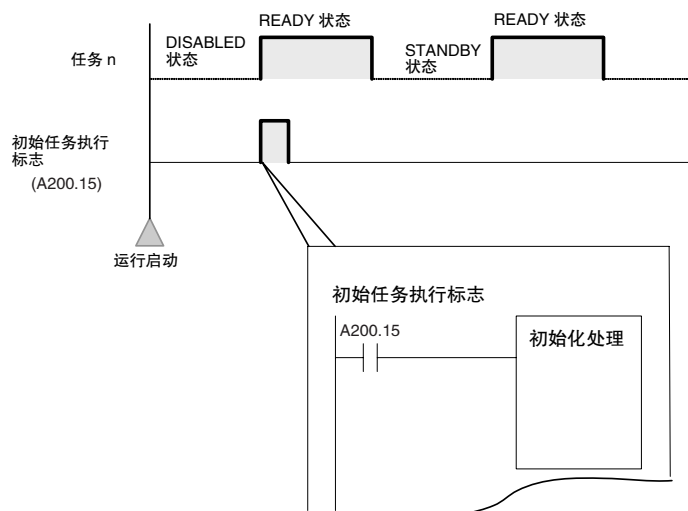
使用该标志来查看当前是否正在执行任务。当循环任务处于 READY 状态时，任务标志变 ON；而当任务处于 DISABLED (INI) 或 STANDBY (WAIT) 状态时，任务标志变 OFF。任务号 00 ~ 127 与 TK000 ~ TK127 对应。



注 任务标志仅用于循环任务，不能用于中断任务。对于中断任务，请使用与中断任务相关的标志，例如字 A440(中断任务最长处理时间)或字 A441(处理时间最长的中断任务)。有关与中断任务相关标志的说明见下文。

初始任务执行标志 (A200.15)

使用该标志在运行期间仅执行一次初始处理。该标志在初次执行某个任务时变 ON，在该任务执行完成后变 OFF。

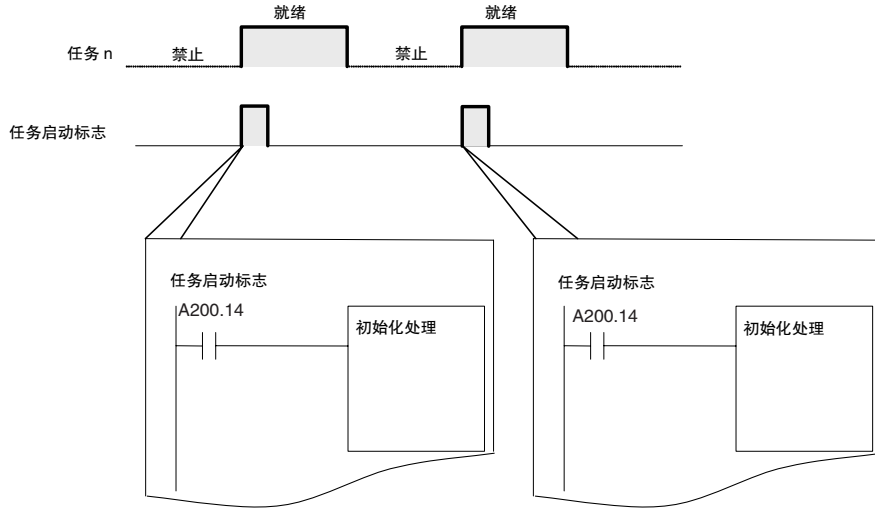


在初次执行某个任务时，初始任务执行标志将变 ON。使用该标志来触发初始化处理。

注 即使使用 TKON(820) 指令将循环任务从 STANDBY 状态变回 READY 状态，也不认为是初始执行，并且初始执行标志 (20015) 将不会再次变 ON。使用任务启动标志 (A200.14)。

任务启动标志 (A200.14)

只要循环任务从 STANDBY 状态变为 READY 状态，任务启动标志均会变 ON(尽管初始任务执行标志 (A200.15) 仅变 ON 一次，任务启动标志在每次状态从 STANDBY 变为 READY 时均会变 ON)。



当通过TKON(820)指令允许了处于STANDBY状态的的任务时，只要该任务从STANDBY状态变为READY状态，任务启动标志均可用于执行初始化处理。

● 中断任务标志和字

中断任务的最长处理时间 (A440)*

中断任务的最长处理时间作为二进制数据以 0.1ms 为单位进行存储，并在操作开始时被清除。

处理时间最长的中断任务 (A441)*

将处理时间最长的中断任务号作为二进制数据存储。此处，8000 ~ 80FF Hex 与任务号 00 ~ FF Hex 对应。

运行开始时该数据清零。

重复刷新出错标志 (非致命错误)(A402.13)

当在 PLC 设置中设定为允许重复刷新错误检测时，如果发生重复刷新错误，则重复刷新出错标志将变 ON。

* 如果在 PLC 设置中允许高速中断功能，则字 A440 和 A441 中的值将无效。

重复刷新错误原因标志 (A426.15)/ 产生重复刷新错误的任务号 (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 (在 PLC 设置中设定为允许中断任务错误检测时)	<p>当在 PLC 设置中设定为允许重复刷新错误检测时, 如果在同一个高功能 I/O 单元中发生以下情况时, 重复刷新出错标志将变 ON。</p> <ul style="list-style-type: none"> · 中断任务中执行的 IORF、FIORF、IORD 或 IOWR 指令与循环任务中执行的 IORF、FIORF、IORD 或 IOWR 指令存在冲突。 · 中断任务中执行的 IORF、FIORF、IORD 或 IOWR 指令与 CPU 单元的 I/O 刷新 (END 刷新) 存在冲突。 <p>在 PLC 设置中设定为允许高功能 I/O 单元的循环刷新, 且对同一个高功能 I/O 单元执行了 IORF、FIORF、IORD 或 IOWR 指令时, 将发生重复刷新的现象并且将产生中断任务错误。</p>	1 (ON)	正在刷新的高功能 I/O 单元的单元号将以 12 位二进制数据存储 (单元号 0 ~ 95: 000 ~ 05F Hex)。

程序停止时的任务号 (A294)

当某个任务因程序出错而停止执行时的任务类型和当前任务号将按如下方式存储:

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

● 与所有任务相关的标志

任务出错标志 (A295.12)

如果发生下述任一错误, 则任务出错标志将变 ON。

- 在循环期间无循环任务处于 READY 状态。
- 分配给某个循环任务 (包括附加循环任务) 的程序不存在。(使用 CX-Programmer 时将不会发生该情况。)
- 无程序分配给激活的中断任务 (包括附加循环任务)。

程序停止时的任务号 (A294)

当某个任务因程序出错而停止执行时的任务类型和当前任务号将按如下方式存储:

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

该信息便于确定发生致命错误的位置, 并且当致命错误被清除时, 该信息不会被清除。任务停止运行处的程序地址存储在 A298(程序地址的最右位) 和 A299(程序地址的最左位) 中。

注意事项

● 指令使用限制

必须处于同一个任务中的指令

下述指令必须位于同一个任务中。

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	条件跳转非 / 条件跳转结束
JMP0/JME0	多路跳转 / 跳转结束
FOR/NEXT	FOR/NEXT
IL/ILC	互锁 / 互锁清除
SBS/SBN/RET	子程序调用 / 子程序入口 / 子程序返回 (如果使用全局子程序指令 (GSBS(750)、GSBN(751) 和 GRET(752)), 则可从另一个任务调用子程序。)
MCRO/SBN/RET	宏 / 子程序入口 / 子程序返回
BPRG/BEND	块程序开始 / 块程序结束
STEP/STEP	步定义

不允许在中断任务中使用的指令

不能将下列指令放入中断任务中。试图在中断任务中执行这些指令之一将导致出错标志 (P_ER) 变 ON 且该指令将不执行。如果中断任务用作附加循环任务, 则可使用下述指令。

助记符	指令
TKON	任务 ON
TKOF	任务 OFF
STEP	步定义
SNXT	步启动
STUP	修改串行端口设定
DI	禁止中断
EI	允许中断
SFCON、SFCOFF、SFCPR、SFCPRN	SFC 任务控制指令

- 下述指令在中断任务中的运行将无法预测：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)，累加定时器：TTIM(087) 和 TTIMX(555)，多路输出定时器：MTIM(543) 和 MTIMX(554)，长定时器：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) 以及故障点检测：FPD(269)。
- 除上述指令之外，下述指令也无法在电源 OFF 中断任务中使用。
如果在电源 OFF 中断任务中使用这些指令，则出错标志 (P_ER) 将不变 ON，并且指令将不执行。
读数据文件：FREAD(700)，写数据文件：FWRITE(701)，写文本文件：TWRITE(704)，网络发送：SEND(090)，网络接收：RECV(098)，发布命令：CMND(490)，发送：TXD(236)，接收：RXD(235) 以及协议宏：PMCR(260)，显式报文发送：EXPLT(270)，显式获取属性：EGATR(271)，显式设定属性：ESATR(272)，显式字读取：ECHR(273)，显式字写入：ECHWR(274)，通过串行通信单元传送：TXDU(256) 以及通过串行通信单元接收：RXDU(255)。

- 在 CJ2H CPU 单元的 PLC 设置中设定了允许高速中断的情况下，无法在中断任务中使用下述指令。否则将产生错误。

选择 EM 存储区块 (EMBC(281)), 网络发送 (SEND(090)), 网络接收 (RECV(098)), 发布命令 (CMND(490)), 协议宏 (PMCR(260)), 通过串行通信单元传送 (TXDU(256)), 通过串行通信单元接收 (RXDU(255)), 显式报文发送 (EXPLT(720)), 显式获取属性 (EGATR(721)), 显式设定属性 (ESATR(722)), 显式字读取 (ECHR(723)) 和显式字写入 (ECHWR(724))。

如果在 PLC 设置中指定了后台处理，则表数据处理指令、字符串处理指令或数据移位指令执行时将产生指令处理错误。



附加信息

任务与定时器运行的关系

定时器的运行方式如下：

- 定时器号 0 ~ 2047
即使包含定时器的任务从 STANDBY 状态返回至 READY 状态，定时器当前值也将继续更新。当任务返回 READY 状态时，如果包含激活的 TIM 指令的任务进入 STANDBY 状态且当前值为 0，则完成标志将在 TIM 指令一执行时即变 ON。(定时器的完成标志仅在指令执行时更新。)当前值将继续更新。
- 定时器号 2048 ~ 4095
当任务处于 STANDBY 状态时，任务中已启动的定时器的当前值将不会更新。但 PLC 属性中有一个设定，可用于对定时器 2048 ~ 4095 允许与定时器 0 ~ 2047 相同的操作。

任务与条件标志的关系

在执行各任务之前将清除所有条件标志。因此无法在任务 2 中读取任务 1 结束时的条件标志状态。CCS(282) 和 CCL(283) 可用于从程序的另一部分 (即另一任务) 读取条件标志状态。

● 有关中断任务的注意事项

中断任务的优先级

发生电源 OFF 中断时，将停止另一中断任务的执行，以允许执行电源 OFF 中断任务。CPU 单元将被复位，但在电源 OFF 中断任务执行之后，将不执行被停止的中断任务。

其它中断任务的优先级在下述章节中描述。

同时发生多个中断

当同时发生多个中断时，除电源 OFF 中断任务之外的其它中断任务将按如下优先级的顺序执行。

I/O 中断任务 > 外部中断任务 > 输入中断任务 * > 高速计数器目标一致 * > 定时中断任务

如果同时发生了一个以上的中断，则多种中断任务中的每个任务将按顺序从最小的任务号开始执行。

* 仅当在 CJ2M CPU 单元上使用脉冲 I/O 模块时支持。

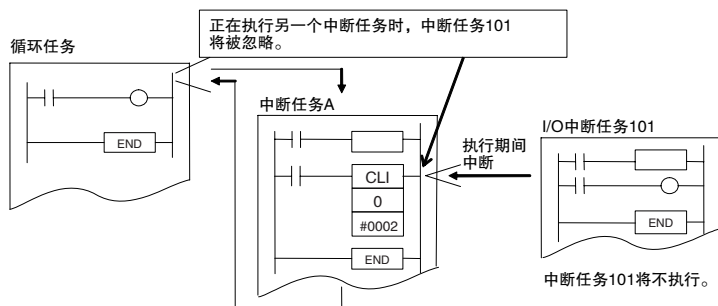
中断任务执行期间发生中断

如果正在执行某个中断任务的同时发生了另一个中断，则该中断的任务将在原中断执行完成后才会执行。



附加信息

如果在一个中断任务执行期间发生另一个特定编号的 I/O 中断任务，并且不希望 CPU 单元保存和执行该 I/O 中断任务，则可从其它任务执行 CLI(清除中断)指令，以清除内部保存的中断号。无法取消定时中断和外部中断。



附加信息

存储器中将为每个中断任务仅记录一个中断，并且无法为已执行的中断任务记录中断。由于定时中断的优先级较低并且一次只记录一个中断，因此可以跳过定时中断。

● 应用注意事项

对高性能 I/O 单元执行 IORF(097)、FIORF(225)、IORD(222) 或 IOWR(223) 指令

如果使用高性能 I/O 单元并且将从某个中断任务执行 IORF(097)、FIORF(225)、IORD(222) 或 IOWR(223) 指令，则请务必在 PLC 设置中禁止对该高性能 I/O 单元的循环刷新。

正在通过循环刷新或者某条 I/O 刷新指令对高性能 I/O 单元进行刷新时，如果试图通过中断任务中的 IORF(097) 或 FIORF(225) 指令来刷新同一个高性能 I/O 单元，或者试图通过 IORD(222) 或 IOWR(223) 指令对同一个高性能 I/O 单元读 / 写数据，则将发生重复刷新错误。在这种情况下，IORF(097)、FIORF(225)、IORD(222) 或 IOWR(223) 指令将不执行，但出错标志 (P_ER) 将不变 ON。循环刷新将正常执行。

如果在 PLC 设置中设定为允许重复刷新错误检测，则当发生重复刷新错误时，A402.13(重复刷新出错标志) 将变 ON，且高性能 I/O 单元的单元号将存储在 A426(重复刷新错误任务号) 中。

● 特定处理期间通过循环任务禁止中断

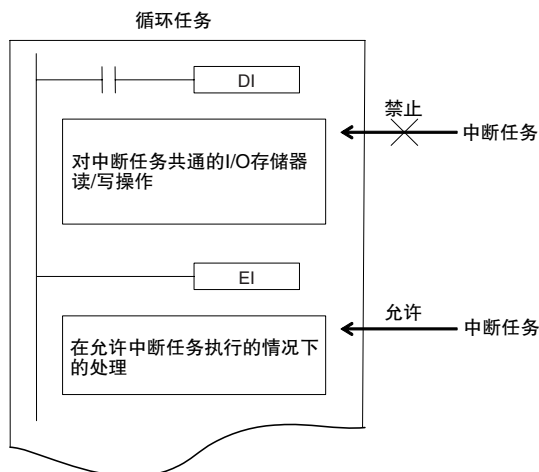
如果在下述情况下发生中断，则处理将被中断并将执行中断任务。

- 正在执行某条指令时。
- 基本 I/O 单元或高性能 I/O 单元刷新期间
- 外设服务期间

确保循环和中断任务之间的数据并发

如果循环任务和中断任务读写同一个 I/O 存储器地址，则可能不保证数据的并发性。使用下述步骤来禁止通过循环任务指令访问存储器期间的中断。

- 在即将通过循环任务指令读 / 写之前，使用 DI(693)(禁止中断) 指令来禁止中断任务的执行。
- 在处理完之后立即使用 EI(694)(允许中断) 指令以允许中断任务的执行。

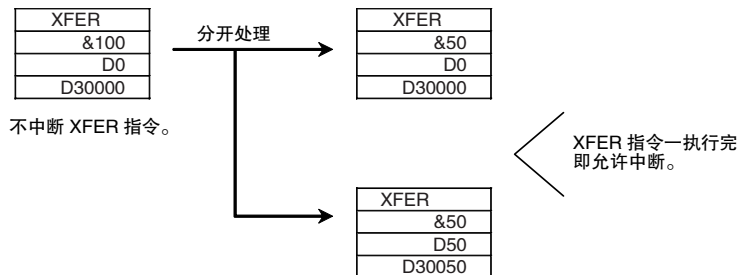


在执行要求响应接收和处理的指令 (例如网络指令或串行通信指令) 期间，即使使用 DI(693) 和 EI(694) 指令来禁止中断任务，也有可能发生数据并发方面的问题。



附加信息

将不会为执行中断任务而中断 BIT COUNTER (BCNT)、BLOCK SET (BSET) 和 BLOCK TRANSFER (XFER) 指令的执行，即在执行中断任务之前将延迟对中断的响应，以便先执行完这些指令。为防止出现该情况，可将对这些指令的数据处理分成一条以上的指令，如下图中对 XFER 所做的处理。

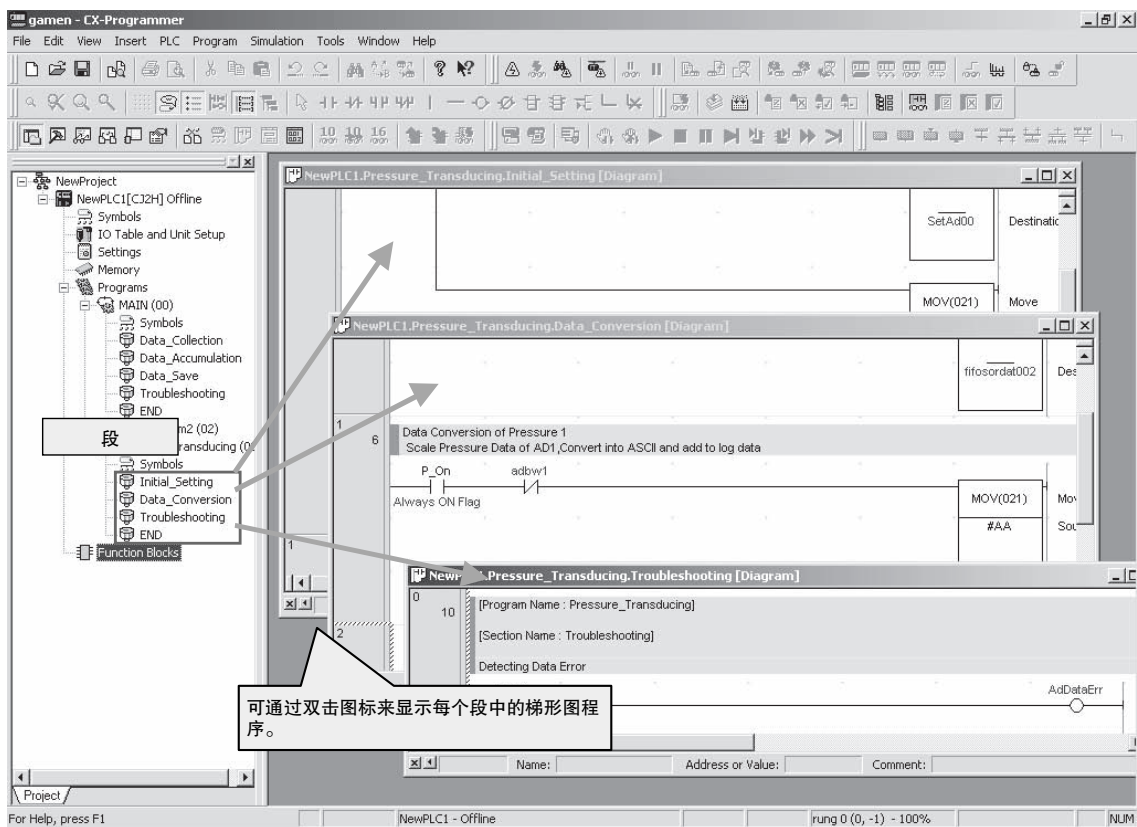


5-3 段

5-3-1 程序段概要

使用 CX-Programmer 时，可按程序段来创建和显示程序，从而将程序划分成多个功能单元。一个任务中的任何程序均可分成多段。使用段有助于提高程序的可读性及简化编辑。

- 处理步骤更清晰
可为段赋任何名称，以表明在该段中执行的处理或控制。
- 编辑更容易
可通过鼠标拖放段来改变段的顺序，并可复制和粘贴段。
- 使用不受限制
每个程序中能够创建的段的数量不受限制。



● 段的数量

可以使用的段的数量不受限制。此外，如前所述，还可创建仅由一个段组成的程序。

● 段的执行顺序

段按照其在项目树中出现的顺序（即从上到下）执行。程序按照任务执行的顺序自动连续向前寻址。每个段中的梯级号从 0 开始。

● 段名称

段名称将为带序列号作为后缀的“段□”，序列号从 1 开始自动分配。可通过用户定义来改变名称。

● 将程序划分成段的优点

将程序划分成段有下述优点。

- 程序可以从 CPU 单元每次上传一个段。如果事先上传程序的某个段，则启动联机编辑所需的时间将缩短。每次只能下载程序的一个段进行联机编辑。也不能以每次一个段的方式下载段。
- 可移动段来改变整个程序的执行顺序。可将段移到其它程序，还可复制到项目树上。

该功能实现了比完整的程序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。针对某个 CPU 单元的段名称、段注释和程序注释存储在程序索引文件 (PROGRAMS.IDX) 中。



附加信息

任务用于控制是否执行任务程序。另一方面，段是在用户定义的程序中以更低层次创建的部分。创建段的目的是使程序便于理解，如下述应用程序例所示。

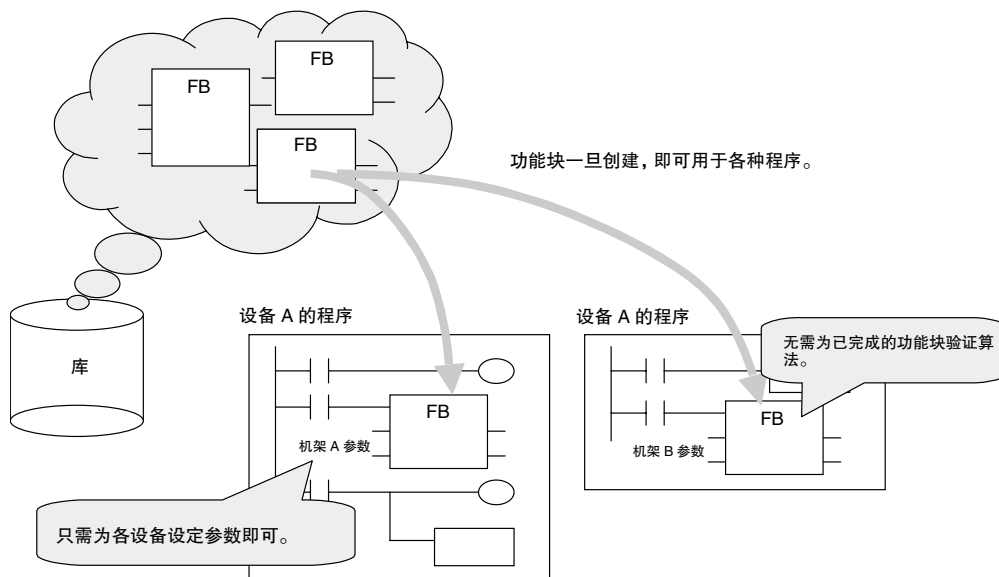
- 如果只要执行任务时则要求执行相同的处理，则可将初始处理和主处理划分成一个任务中的不同段。
- 普通程序区和子程序可划分成多个段。

5-4 功能块

5-4-1 功能块

功能块 (FB) 是基本的程序元素, 包含事先已定义的、用于标准处理功能的程序部分。创建好功能块并将其保存到库中之后, 只需选择系统所需的功能块、将功能块放入程序中并设定 I/O 参数, 即可在编程时使用这些功能块。优秀的功能块可显著提高程序开发效率。

OMRON 功能块符合 IEC 61131-3 标准。



● 功能块的特点

- 复用性
功能块保存在库中, 因而方便复用。程序员理解功能块之后, 即可通过设定参数来轻松使用功能块, 从而显著提高了编程效率。
- 调试工作量减轻
复用已经过调试的高质量的功能块可消除再次调试的必要, 因而减轻了调试的工作量。
- 数据保护
无法从外部直接访问功能块内部的变量, 因而为数据提供了保护。(不会意外修改数据。)

5-4-2 功能块的特点

● 功能块地址的输入

通常使用变量而非实际的物理地址来输入功能块的地址。程序员使用这些变量来设定参数（即：地址或值）。用于各变量的地址在每次将功能块放入程序中时通过 CX-Programmer 自动分配。

● 嵌套

一个功能块可在另一功能块内部嵌套。包括调用第一层功能块在内，最多可嵌套 8 层。

● Smart FB 库

Smart FB 库是功能块的集合，可改善 OMRON PLC 单元和 FA 组件之间的操作。无需为使用基本单元和 FA 组件功能而创建梯形图。

● FB 定义的联机编辑

可在运行过程中改变 FB 定义，因而可在调试过程中迅速编辑 FB 定义。此外，即使是在必须 24 小时 / 天运行的设备中也可放心使用 FB。

● 保护 FB 定义

可通过为项目文件中分配的功能块定义设定密码以及根据功能块定义的用途来保护其定义的方式，防止对程序进行未经授权的操作、编辑或滥用。

● FB 生成功能

只需将现有 PLC 程序转换成 FB，即可复用这些程序。

5-4-3 功能块规格

项目	规格		
定义和创建功能块	功能块定义的数量	CJ2H-CPU6□(-EIP) 或 CJ2M-CPU□5/□4: 最多 2,048 个 CJ2M-CPU□3/□2/□1: 最多 256 个	
	功能块名称	最多 64 个字符	
	变量	变量名称	最多 15,000 个字符
		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)
		功能块中使用的变量个数(不包括内部变量、外部变量、EN 和 ENO)	每个功能块定义的变量的最大数量 · 输入 - 输出变量: 最多 16 个 · 输入变量 + 输入 - 输出变量: 最多 64 个 · 输出变量 + 输入 - 输出变量: 最多 64 个
		变量所使用地址的分配	自动分配 (分配范围可由用户设定。)
	实际地址指定	支持	
	数组指定	支持 (仅限一维数组, 且仅限内部变量和输入 - 输出变量)	
语言	可用梯形图编程语言或结构化文本创建功能块。		
创建实例	实例数	CJ2H-CPU6□(-EIP) 或 CJ2M-CPU□5/□4: 最多 2,048 个 CJ2M-CPU□3/□2/□1: 最多 256 个	
	实例名称	最多 15,000 个字符	
FB 程序区	20K 步 仅 CJ2M CPU 单元支持。该区域用于存储功能块定义。		
将功能块 作为文件存储	项目文件	项目文件 (.exp/ext) 包括功能块定义和实例。	
	程序 / 网络符号文件	文件存储器程序文件 (*.obj) 包括功能块定义和实例。	
	功能块库文件	可将每个功能块定义作为单个文件 (.exf) 存储, 以便在其它项目中复用。	

功能块所使用的步数

使用功能块时, 下述两项需要使用存储器:

1. 功能块定义
2. 当将功能块作为功能块实例粘贴到用户程序中时的实例生成处理

因此, 存储器中使用的步数将随程序中创建的功能块实例数 (第 2 项) 增加而增加。

用于功能块的存储区

用于功能块的存储器取决于所使用的 CJ2 CPU 单元的型号，如下表所示。

CJ2M CPU 单元中有一个称为 FB 程序区的特殊区域，用于存储功能块的定义。而 CJ2H CPU 单元中则无该区域。

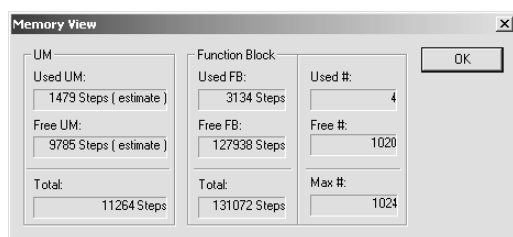
CPU 单元	型号	用于功能块的存储区
CJ2H	CJ2H-CPU6□-EIP CJ2H-CPU6□	用户程序区
CJ2M	CJ2M-CPU3□ CJ2M-CPU1□	创建功能块实例 (即: 调用功能块定义和传送参数) 将使用用户程序存储器。 功能块定义使用 FB 程序区 (用于功能块的特殊区域) 中的存储器。如果超过 FB 程序区的容量, 则将使用用户程序区。

检查功能块存储器的使用情况

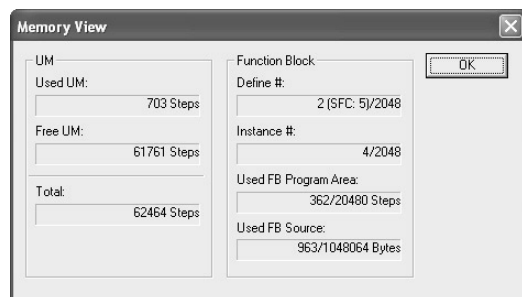
从 9.0 版或更高版本的 CX-Programmer 中选择 “View – Memory View” (查看 – 查看存储器)。

此时将显示下述 “Memory View” (查看存储器) 对话框。

- 当 PLC 型号 (“设备类型”) 为 CS/CJ 系列 PLC(3.0 版或更高版本) 而非 CJ2M 时



- 当 PLC 型号 (“设备类型”) 为 CJ2M 时



附加信息

有关 “存储器查看对话框” 的详情, 请参考《CX-Programmer 操作手册》(样本编号: W446)。

计算功能块所使用的步数

可使用下列公式计算功能块所使用的步数。

所使用的步数
= 实例数 × (调用部分的大小 m + I/O 参数传送部分的大小 n × 参数个数) + 指令在功能块定义中的步数 p(见注)

注 当在多个位置创建了同一个功能块定义(即:用于多个实例)时,则存储器将无法用于第二个及第二个以后的实例中的功能块定义中的指令步数(p)。因此,在上述公式中,不将实例数乘以功能块定义中的指令步数(p)。

项目		步数	
m	调用部分	57 步	
n	I/O 参数传送部分 数据类型如括号中所示。	1 位 (BOOL) 输入符号或输出符号	6 步
		1 个字 (INT、UINT、WORD) 输入符号或输出符号	6 步
		2 个字 (DINT、UDINT、DWORD、REAL) 输入符号或输出符号	6 步
		4 个字 (LINT、ULINT、LWORD、LREAL) 输入符号或输出符号	12 步
		I/O 符号	18 步
p	指令在功能块定义中的步数	指令的总步数(与标准用户程序相同) + 27 步	

例

5 个单字 (INT) 数据类型的输入符号、5 个单字 (INT) 数据类型的输出符号和 100 步的功能块定义:

1 个实例中的步数 = 57 + (5 + 5) × 6 步 + 100 步 + 27 步 = 244 步

如果用标准文本语言编写功能块,则无法计算实际的步数。可在功能块定义的属性中查看功能块定义中的指令步数。



附加信息

有关功能块规定、标注方法和输入步骤的说明,请参考《CX-Programmer 操作手册:功能块和结构化文本》(样本编号:W447)。

5-5 符号

5-5-1 概述

符号

- 符号表

可通过将字符串注册为符号的方法，使用字符串来指定 I/O 存储区地址或常数。将符号注册到 CX-Programmer 的符号表中。采用符号编程只需使用名称而无需知道实际地址。

符号表随同其它参数（例如用户程序和 I/O 表）一起保存在 CX-Programmer 项目文件 (.CXP) 中。

- 使用符号的条件

符号的使用是“可选”还是“需要”取决于编程语言以及是在功能块内部还是外部使用符号，如下表所示。

程序元素	编程语言	指定符号
在功能块外部编程	梯形图程序	可选
	ST	需要
	SFC	可选
在功能块内部编程（在功能块内部，符号称为变量。）	梯形图程序	需要
	ST	

5-5-2 符号的类型

可使用下列类型的符号。

程序符号

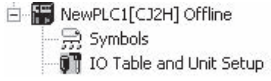

● 符号的类型

- 全局符号
是指可从 CPU 单元中的任意任务访问的符号。
- 局部符号
是指只能从一个任务访问的符号。
- 网络符号 (仅限 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□)
可从外部设备使用标签, 经 CPU 单元中定义的网络符号来访问本地 CPU 单元的 I/O 存储器。网络符号还可用于本地 CPU 单元中。

● 地址分配

基于下述因素将地址分配至程序中所使用的符号。

- 用户规定
- 使用 CX-Programmer 进行自动分配
通过从 CX-Programmer 的 PLC 菜单中选择 “Memory Allocation – Automatic Address Allocation” (存储器分配 – 自动地址分配), 即可设定为用于自动分配的存储区。

符号的类型	名称	CX-Programmer 项目树形图	范围			地址和 I/O 注释 (无符号名称)
			使用网络符号访问	从其它任务访问	从局部任务访问	
程序符号	全局符号	PLC 树 	不支持	支持	支持	支持
	局部符号	程序树 		不支持	支持	不支持
	网络符号	PLC 树形图中的全局变量表 	支持	支持	支持	不支持

注 “全局”和“局部”只表示符号的应用范围, 而与存储器地址的应用范围无关。因此, 在下述情况下仅发出报警而非报错, 而且仍可传送用户程序。

- 两个不同的局部符号使用同一个地址
- 一个全局符号与一个局部符号使用同一个地址



附加信息

在 CX-Programmer 程序中，全局符号、局部符号和网络符号可通过下列字符颜色和表示符号的图标来识别。

符号的类型	显示颜色	例 (默认颜色)
全局符号	黑 (固定)	
局部符号	蓝 (默认) 选择 “Tools – Options” (工具 – 选项)， 显示 “Appearance” (外观) 选项页，然 后选择要改变颜色的局部符号项。	
网络符号	黑 (固定) 符号名称下方将显示一个用于表示网络 符号的图标 ()。	

功能块中的变量

只能通过功能块中的算法来访问变量。

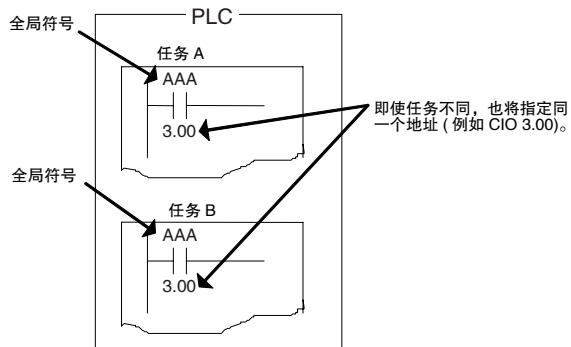
名称	CX-Programmer 项目 树形图	范围		地址和 I/O 注释 (无符 号名称)
		使用网络符号访问	从功能块外部访问	
变量 (即: 在功能 块中使用的符号)	功能块中的定义 	不支持 (无法为外 部变量注册网络符 号)	不支持	不支持

变量的某些规定与在功能块外部使用的符号的规定不同。通过在 CX-Programmer 中的 PLC 菜单下选择 “Memory Allocation – Function Block/SFC Memory” (存储器分配 – 功能块 /SFC 存储器) 来设定变量使用的区域。

5-5-3 全局符号

全局符号是目标 CPU 单元中的所有任务均支持的符号。因此，举例而言，名称为“AAA”的符号在目标 CPU 单元的所有任务中将使用同一个地址。

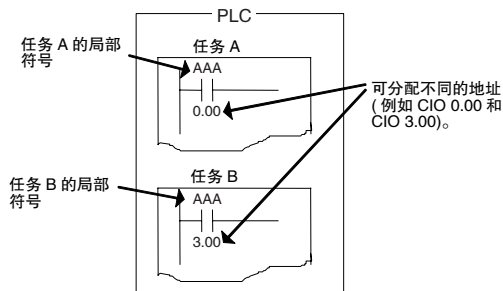
例：如果将名称为“AAA”的符号设定为全局符号，则即使任务不同，也将指定同一个地址(例如 CIO 3.00)。



5-5-4 局部符号

局部符号只能从定义这些符号的任务进行访问。每个局部符号只在一个任务中支持。局部符号分别为每个任务指定，因此对于不同任务，完全相同的符号也将作为不同的符号处理。

例如，如果将名称为“AAA”的符号设定为局部符号，则可通过同一个“AAA”符号来为不同的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 (仅限 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□)

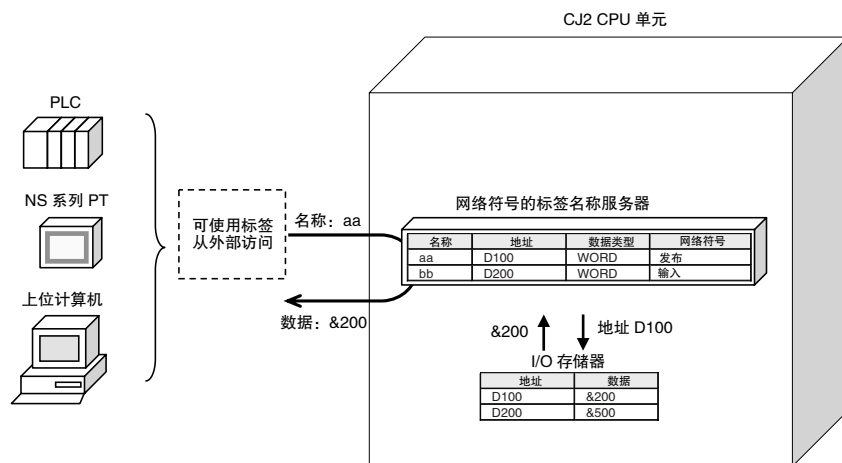
概述

可从外部设备使用标签，经 CPU 单元中定义的网络符号来访问本地 CPU 单元的 I/O 存储器。CJ2 CPU 单元带有一个内部标签名称服务器，可将网络符号转换成实际的 I/O 地址，因此可通过使用标签来访问网络符号的方法，从外部访问 CPU 单元的 I/O 存储器。

网络符号可用于以下场合。

- 与 EtherNet/IP 上的其它 PLC 进行数据链接
- 通过上位机或 PT 从外部访问标签。

在 CX-Programmer 的全局符号表中设定网络符号。



附加信息

有关创建标签数据链接的步骤，请参考《EtherNet/IP 单元操作手册》(样本编号: W465)。

设定网络符号

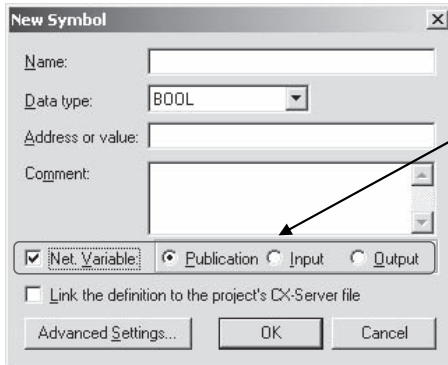
使用下述步骤来设定网络符号。

1. 在全局变量表上点击右键并选择“Insert Symbol”（插入符号）。

注 无法在局部符号表中指定网络符号。

2. 此时将显示下述“New Symbol”（新符号）对话框。

选择“Net, Variable”（网络 - 变量）复选框，然后选择“Publication”（发布）、“Input”（输入）或“Output”（输出）选项，然后单击“OK”（确定）按钮。



选择“Net, Variable”（网络 - 变量）复选框，然后选择“Publication”（发布）、“Input”（输入）或“Output”（输出）选项。

网络符号名称最多可使用 48 个字符。名称不区分字母大小写。

符号的类型	描述	全局符号表中的网络符号栏	描述
网络符号	可使用标签从外部访问的符号。	发布	<ul style="list-style-type: none"> 可使用 CIP 报文通信从外部访问的符号。 不支持标签数据链接。
		输入	<ul style="list-style-type: none"> 可使用 CIP 报文通信或标签数据链接从外部访问的符号。 使用标签数据链接时用于数据输入(从其它 CPU 单元到本地 CPU 单元)的符号。
		输出	<ul style="list-style-type: none"> 可使用 CIP 报文通信或标签数据链接从外部访问的符号。 使用标签数据链接时用于数据输出(从本地 CPU 单元到其它 CPU 单元)的符号。

● 全局符号表中的网络符号栏

在全局符号表的网络符号栏中进行设定。

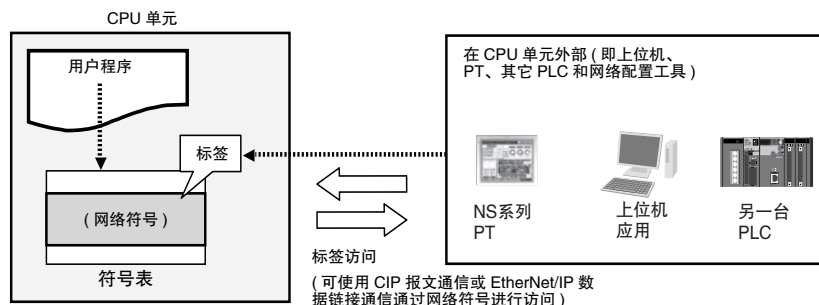
Name	Data Type	Address / Value	Net. Variable	Rack Location	Usage	Comment
Node1_IN01	WORD	D100	INPUT		Work	
Node1_OUT01	WORD	D200	OUTPUT		Work	
Node2_IN02	WORD	D101	INPUT		Work	
Node1_OUT02	WORD	D201	OUTPUT		Work	

这些可设定为“发布”、“输入”或“输出”的符号将作为网络符号。

网络符号的使用条件和操作

● 从外部操作网络符号

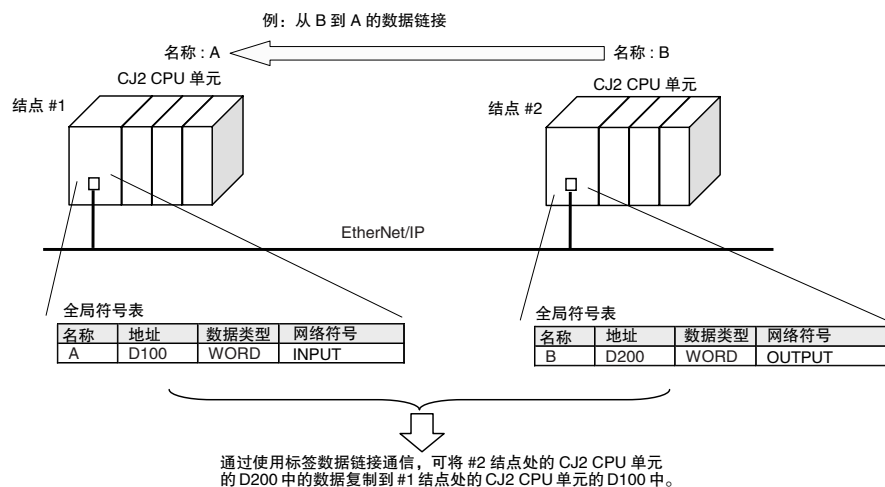
在 CPU 单元中设定为网络符号的符号将从 PLC 外部 (例如通过上位机、PT、网络配置工具或其它 PLC) 作为标签识别。



网络符号应用

● 将网络符号用作 EtherNet/IP 数据链接标签

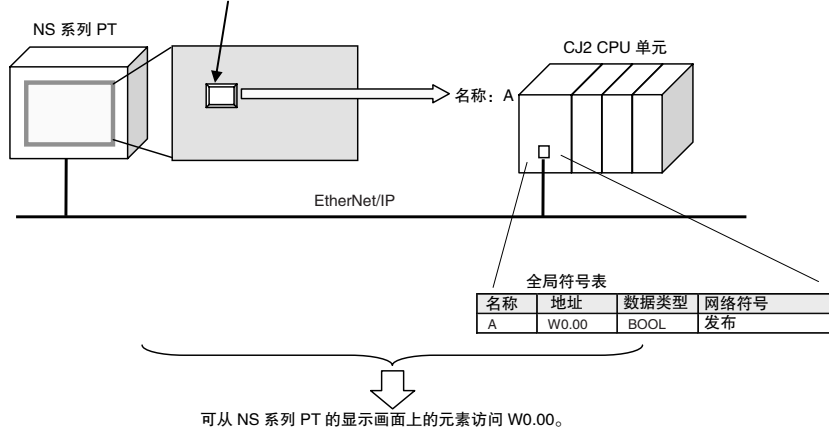
可使用 EtherNet/IP 标签数据链接通信, 通过标签来指定远程 PLC 中的数据。在全局符号表中的网络符号栏中将网络符号设定为 “Input” (输入) 或 “Output” (输出)。



● 在与 NS 系列 PT 的通信中使用标签

可使用 NS 系列 PT 的显示屏上的目标，通过标签来指定远程 PLC 中的数据。在全局符号表中的“Network Symbol”（网络符号）栏中将网络符号设定为“Publication”（发布）。

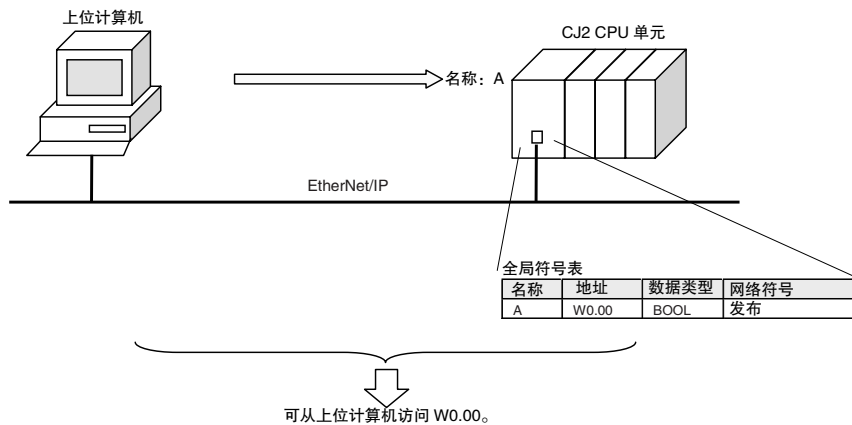
例：将“A”指定为画面上目标的地址。



● 从上位计算机将标签用于 CIP 报文通信

从上位计算机通过标签指定远程 PLC 中的数据。在全局符号表中将网络符号设定为“Publish”（发布）。

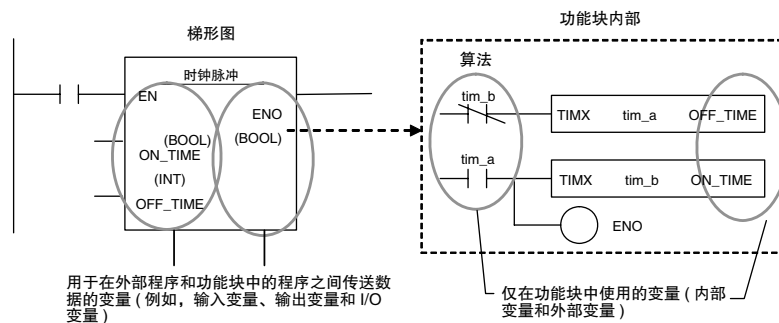
例：从上位计算机通过名称指定。



5-5-6 功能块中的变量

功能块中的程序全部通过变量而非实际地址来编写。功能块中变量的变量类型和规定与功能块外部的符号不同。有关功能块中变量的详情，请参考《CX-Programmer 操作手册：功能块和结构化文本》(样本编号：W447)。

● 功能块变量域



5-5-7 符号数据类型

可为分配给符号的地址指定数据类型。

可为符号设定的数据类型

在功能块内部可指定的数据类型可能会与在功能块外部指定的类型不同。这两种情况下可使用的数据类型在下表中指定。

数据类型	程序中的符号	功能块中的变量
BOOL(位)	允许	允许
UINT(二进制, 单字, 无符号)	允许	允许
UDINT(二进制, 双字, 无符号)	允许	允许
ULINT(二进制, 四字, 无符号)	允许	允许
INT(二进制, 单字, 带符号)	允许	允许
DINT(二进制, 双字, 带符号)	允许	允许
LINT(二进制, 四字, 带符号)	允许	允许
UINT BCD(二进制, 单字, 无符号)*1	允许	不允许
UDINT BCD(二进制, 双字, 带符号)*1	允许	不允许
ULINT BCD(二进制, 四字, 带符号)*1	允许	不允许
REAL(双字, 浮点数)	允许	允许
LREAL(四字, 浮点数)	允许	允许
CHANNEL(字)*1	允许	不允许
NUMBER(常数或数值)*2	允许	不允许
WORD(十六进制, 单字)	允许	允许
DWORD(十六进制, 双字)	允许	允许
LWORD(十六进制, 四字)	允许	允许
STRING(字符串: ASCII 字符, 1 ~ 255)	允许	允许
TIMER	不允许	允许
COUNTER	不允许	允许
用户自定义的数据类型(数据结构)	允许	允许*3

*1 数据类型在 ST 编程中将如下转换:

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

*2 这些数据类型无法在 ST 编程中使用。否则将产生程序错误。

*3 仅限功能块中通过梯形图写入的内部符号。

数据类型的应用

为符号设定的数据类型可用于下述指令。

● 使用自动地址分配的应用

当通过 CX-Programmer 来使用自动地址分配功能时，可根据为符号设定的数据类型来分配地址。

例：设定 ULINT(二进制，四字，无符号)数据类型

Name	Data Type	Address/Value	Net. Variable	R
ABC	ULINT	W200 [Auto]		
DEF	UDINT	W204 [Auto]		

如果使用四个字的数据类型，则下一个分配符号的地址将相隔四个字。

指定数组

可将数组作为符号属性指定(仅限一维数组)。可为除 STRING 和 NUMBER 以外的所有数据类型指定数组。

如果将多个具有相同数据属性的符号作为一个组进行管理，则可使用数组规定。使用数组规定时，可采用下述步骤。

1. 在“New Symbol”(新符号)或“Edit Symbol”(编辑符号)对话框中输入符号名称，然后单击“Advanced Settings”(高级设定)按钮。此时将显示“Advanced Settings”(高级设定)对话框。
2. 在“Array Settings”(数组设定)字段中选择“Array Symbol”(数组符号)选项，然后输入数组元素的最大个数。

写入数组变量时，请在变量名称后指定一个带括号的后缀。

例：如果符号名称为PV且元素的最大个数为3，则在指令操作数中将数组变量指定为PV[0]、PV[1]和PV[2]。

用从 0 开始的数组元素编号来指定数组的后缀。可直接输入元素号，也可通过输入一个符号或存储器地址来间接指定元素号。

⚠ 注意

如果在梯形图或 ST 语言编程时为数组变量的下标指定符号或存储地址(ST 语言编程仅允许指定符号)从而间接指定数组元素，请确保元素的数目不超过最大存储区范围。

若指定的元素数目超出为该符号所指定的存储区的最大范围，将导致访问另一存储器内的数据，从而可能会导致意外动作。



以字地址或符号作为后缀的数组变量无法用作即时刷新指令的操作数。

用户自定义的数据类型 (数据结构)

当在 CJ2 CPU 单元上使用 9.0 版或更高版本的 CX-Programmer 时，可作为用户定义的数据类型来创建数据结构。

● 数据结构

数据结构是一种用户自定义的数据类型，集合了一种以上的数据类型。可以为数据类型指定名称。使用用户自定义数据类型的变量名称随同数据结构中的一个变量的名称一起指定。整个数据结构称为结构变量，且数据结构内的变量称为成员。



附加信息

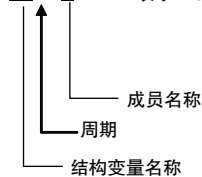
- 数组也可用于处理多条数据。但数组数据类型与数据结构的区别在于，数组数据类型包含相同数据类型的数据，这些数据可通过指定与数组开头的偏移来访问。而数据结构则可包含相同或不同数据类型的数据，这些数据类型通过成员名称来访问。此外，使用 CX-Programmer 时，可为数据结构中的数据类型指定名称。
- 可将结构变量放入数组中。
- 此外，可将数组变量用作数据结构的成员。

● 结构变量的梯形图符号和输入

在梯形图中，结构变量名称和成员名称用句点分隔。

例：

S1 . A = 表示“S1的成员A”。

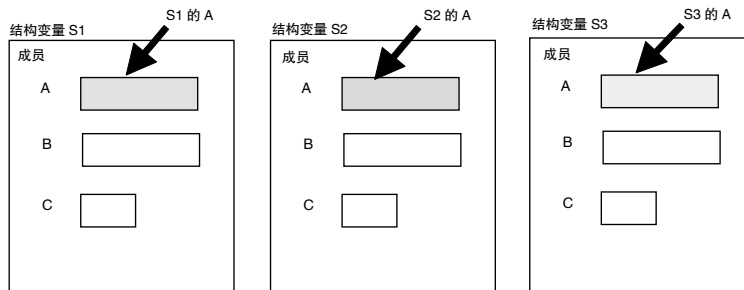


● 数据结构的应用

当指定成组的数据时，可指定一个特定组的成员。换言之，可按“主 - 从”的架构来指定数据。

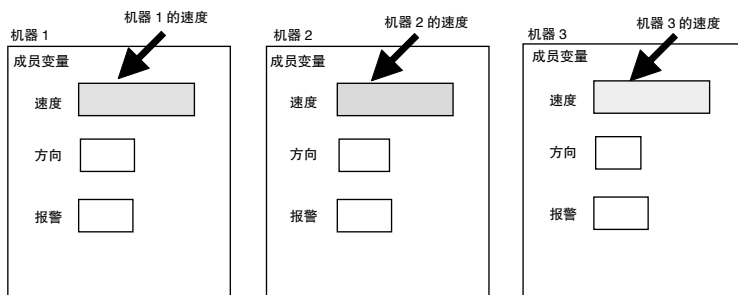
不借助数组来使用数据结构

均带有成员 A、B 和 C 的结构变量 S1、S2 和 S3 的示例



例如，可为每台机器指定同一个成员（例如速度）。

称为“机器 1”、“机器 2”和“机器 3”的结构变量

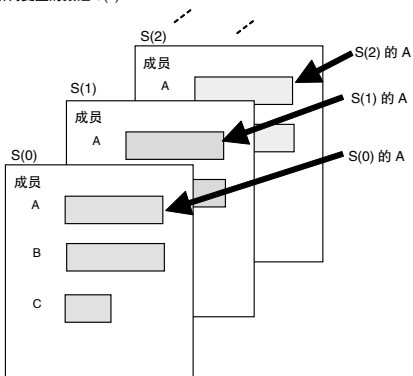


将数据结构和数组组合使用

将结构变量放入数组中

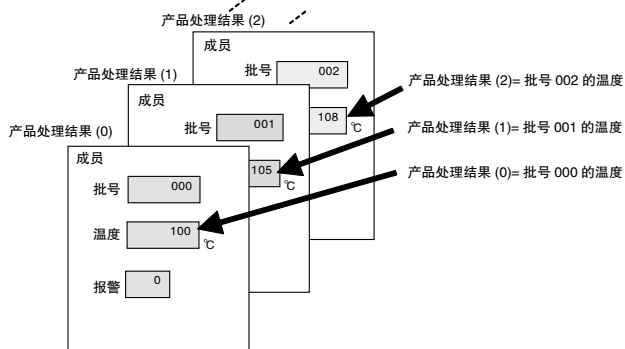
当存在大量相同形式的数据库时，可将结构变量放入数组中，正如为不同产品设定的配方数据一样。该功能用于创建数据库。在这种情况下，结构变量成为一条记录，而每个成员成为数据库中的字段。

结构变量的数组 S(x)



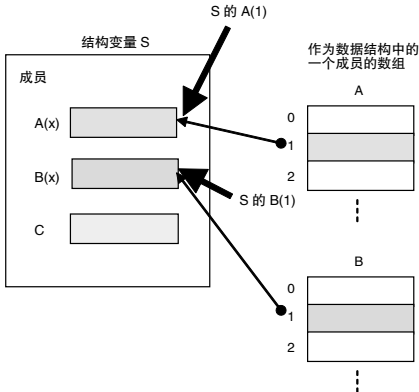
例如，可将其用于指定各生产批号的处理结果。

数组中的结构变量示例：产品处理结果 (x)

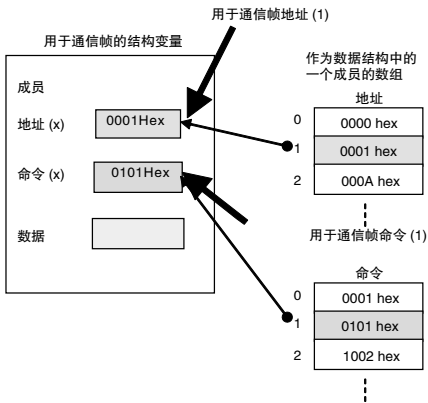


将数组变量用作数据结构的成员

当数据具有特定的整体结构，且数据中的成员各自包含库形式的多个元素时，可使用数组作为数据结构的成员。成员可从数组自由指定。当从数据库构建数据时，可采用上述方法。



例：可通过选择数据库中的元素来创建通信帧。



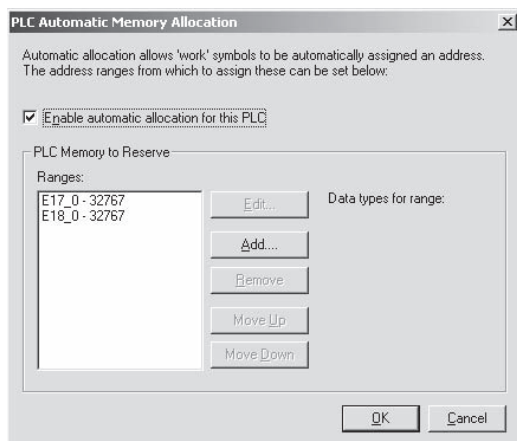
有关数据结构规范和创建步骤的详情，请参考《CX-Programmer 操作手册：功能块和结构化文本》(样本编号：W447)。

5-5-8 将地址自动分配到符号

编程时, 可将地址自动分配到符号。从 CX-Programmer 的 PLC 菜单中选择 “Memory Allocation – Automatic Address Allocation” (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, 可分配 CIO 区、辅助区、保持区、工作区、DM 区或 EM 区中的地址。

当自动分配 EM 区中的符号地址时, 建议从编号最大的 EM 区存储区块开始从后向前分配所需数量的存储区块。

下例所示为自动分配 CJ2H-CPU68-EIP CPU 单元 (带有 EM 区存储区块 00 ~ 18 Hex) 的存储区块 17 和 18 Hex 中的符号地址的设定。



有关自动地址分配的区域设定详情, 请参考《CX-Programmer 操作手册》(样本编号: W446)。



附加信息

- 对 CJ2H CPU 单元而言, 可对被设定为自动地址分配的 EM 存储区块中的位进行强制置位 / 复位。有关其中的位可强制置位 / 复位的存储区块的说明, 请参考 “6-15 扩展数据存储区”。
- 无法对结构变量进行自动地址分配。



安全使用注意事项

当电源断开或运行模式改变时, 将备份 EM 区中的数据。使用被指定为 BOOL 型数据的输出位时请务必小心。必要时, 请加入清除存储器的程序段。

例: 若 EM 区存储区块 18 Hex 被设定为进行自动地址分配, 则可使用下列指令在电源接通时将存储区块 18 Hex 清为全零。

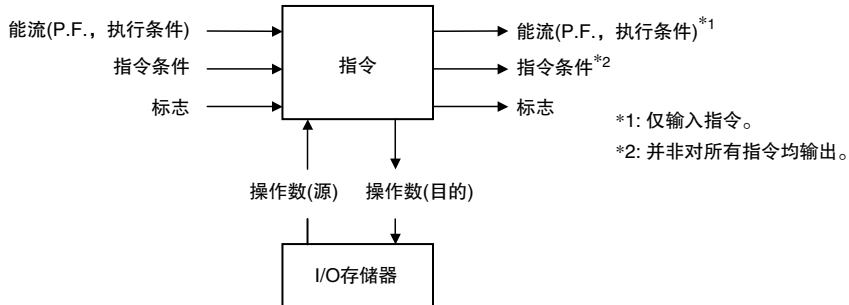


5-6 指令

5-6-1 指令的基本理解

指令结构

程序由多条指令组成。指令输入输出的基本结构如下图所示。

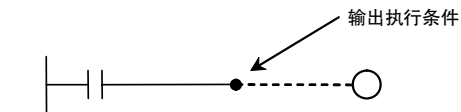


● 能流

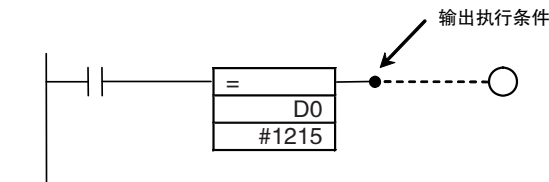
能流是在程序正常执行时用于控制执行和指令的执行条件。在梯形图程序中，能流表示执行条件的状态。

输入指令

- 载入指令表示一个逻辑开始并输出执行条件。

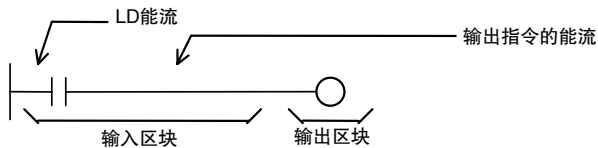


- 中间指令输入能流作为执行条件，并向中间指令或输出指令输出能流。



输出指令

输出指令使用能流作为执行条件，执行所有功能。



● 指令条件

指令条件是整体指令执行相关的特殊条件，通过下述指令输出。在决定是否执行某条指令时，指令条件的优先级高于能流 (P.F.)。根据指令条件而定，某条指令可能不会执行，或者可能以另一种方式执行。指令条件在每个任务开始时复位 (取消)，即当任务改变时，指令条件复位。

下列指令成对使用，用于设定和取消特定的指令条件。这些成对的指令必须用于同一个任务中。

指令条件	描述	设定指令	取消指令
互锁	互锁可使程序的某些部分不起作用。对输出位置 OFF、使定时器复位和保持计数器等特殊条件将生效。	IL(002)	ILC(003)
BREAK(514)执行	结束执行过程中的 FOR(512) - NEXT(513) 循环。(不执行从当前指令到 NEXT(513) 指令之间的所有指令。)	BREAK(514)	NEXT(513)
	执行一条从 JMP0(515) 到 JME0(516) 的跳转指令。	JMP0(515)	JME0(516)
块程序执行	执行从 BPRG(096) ~ BEND(801) 的程序块。	BPRG(096)	BEND(801)

● 标志

此处的标志是用作指令间接口的一个位。

输入标志		输出标志	
标志	描述	标志	描述
进位 (CY) 标志	进位标志用作数据移位指令和加法 / 减法指令中的未指定操作数。	条件标志	条件标志包括常 ON/OFF 标志以及由指令执行结果更新的标志。在用户程序中，这些标志可通过 P_On、P_Off、P_ER、P_CY、P_EQ 等标签而非地址来指定。
用于特殊指令的标志	这些标志包括 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

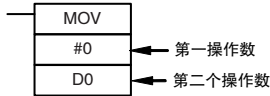
操作数指定预置的指令参数 (梯形图中的框)，这些参数用于指定 I/O 存储区的内容或常数。输入地址或常数作为操作数后，指令即会被执行。操作数分为源操作数、目的操作数或编号操作数。

例



操作数类型		操作数符号	描述	
源	指定要读取数据或常数的地址。	S	源操作数	控制数据 (C) 以外的源操作数
		C	控制数据	源操作数中的复合数据，其含义取决于位状态。
目的 (结果)	指定要写入数据的地址。	D	---	---
编号	指定指令中使用的特定编号，例如跳转编号或子程序编号。	N	---	---

注 操作数还可从指令顶部起称为第一操作数、第二操作数……等依次类推。



指令位置和执行条件

下表给出了指令允许的位置。指令分为需要执行条件的指令和不需要执行条件的指令。有关指令的详情请参考“**A-1 指令功能**”。

指令	位置	执行条件	图	例
输入指令	逻辑开始 (载入指令)	不需要		LD, LD TST(350), LD> (及其它符号比较指令)
	中间指令	需要		AND, OR, AND TEST (350), AND>(及其它 ADD 符号比较指令), UP(521), DOWN(522), NOT(520) 等
输出指令	直接与右侧母线连接	需要		包括 OUT 和 MOV(021) 在内的大多数指令
		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数据 (即时刷新)。

变化	符号	描述	
微分	ON	@	当执行条件变为 ON 时, 执行微分操作的指令。
	OFF	%	当执行条件变为 OFF 时, 执行微分操作的指令。
即时刷新	!	执行指令时, 刷新由操作数或高性能 I/O 单元中的字指定的 I/O 区中的数据。	



执行条件

可使用以下两种基本和特殊指令。

- 非微分指令: 每个循环均执行
- 微分指令: 仅执行一次

● 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 (ON 或 OFF) 时, 这些指令每个循环均执行一次。



输入指令 (逻辑开始和中间指令):

此类指令用于在每次循环中读取位状态、作比较、测试位或执行其它类型的处理。如果结果为 ON, 则输出能流 (即执行条件变为 ON)。

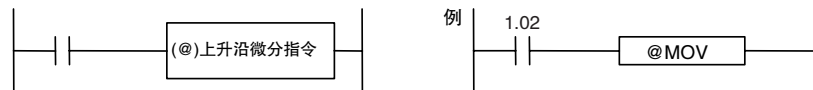


● 输入微分指令

上升沿微分指令 (指令前加 @ 符号)

输出指令:

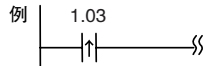
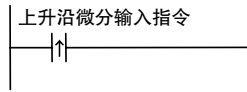
仅当循环过程中执行条件从 OFF 变为 ON 时执行指令, 且后续循环不再执行。



当 CIO 1.02 从 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):

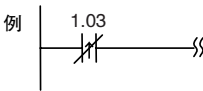
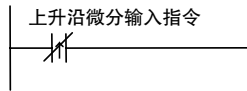
指令在每个循环中读取位状态、作比较、测试位或执行其它类型的处理, 并且当结果从 OFF 变为 ON 时, 将输出一个 ON 执行条件 (能流)。执行条件将在下一个循环变为 OFF。



例 仅当 CIO 1.03 从 OFF 变为 ON 时, 为一个循环生成 ON 执行条件。

输入指令 (逻辑开始和中间指令):

指令在每个循环中读取位状态、作比较、测试位或执行其它类型的处理, 并且当结果从 OFF 变为 ON 时, 将输出一个 OFF 执行条件 (能流停止)。执行条件将在下一个循环变为 ON。

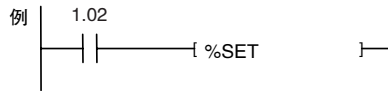
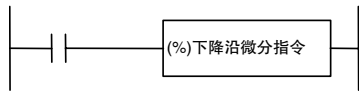


例 仅当 CIO 1.03 从 OFF 变为 ON 时, 为一个循环生成 OFF 执行条件。

下降沿微分指令 (指令前加 % 符号)

输出指令:

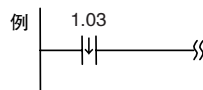
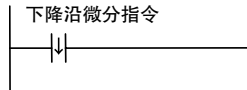
仅当循环过程中执行条件从 ON 变为 OFF 时执行指令, 且后续循环不再执行。



例 当 CIO 1.02 从 ON 变为 OFF 时, 执行 SET 指令一次。

输入指令 (逻辑开始和中间指令):

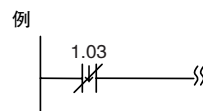
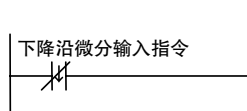
指令在每个循环中读取位状态、作比较、测试位或执行其它类型的处理, 并且当结果从 ON 变为 OFF 时, 将输出执行条件 (能流)。执行条件将在下一个循环变为 OFF。



例 当 CIO 1.03 从 ON 变为 OFF 时将变为 ON, 且在一个循环后变为 OFF。

输入指令 (逻辑开始和中间指令):

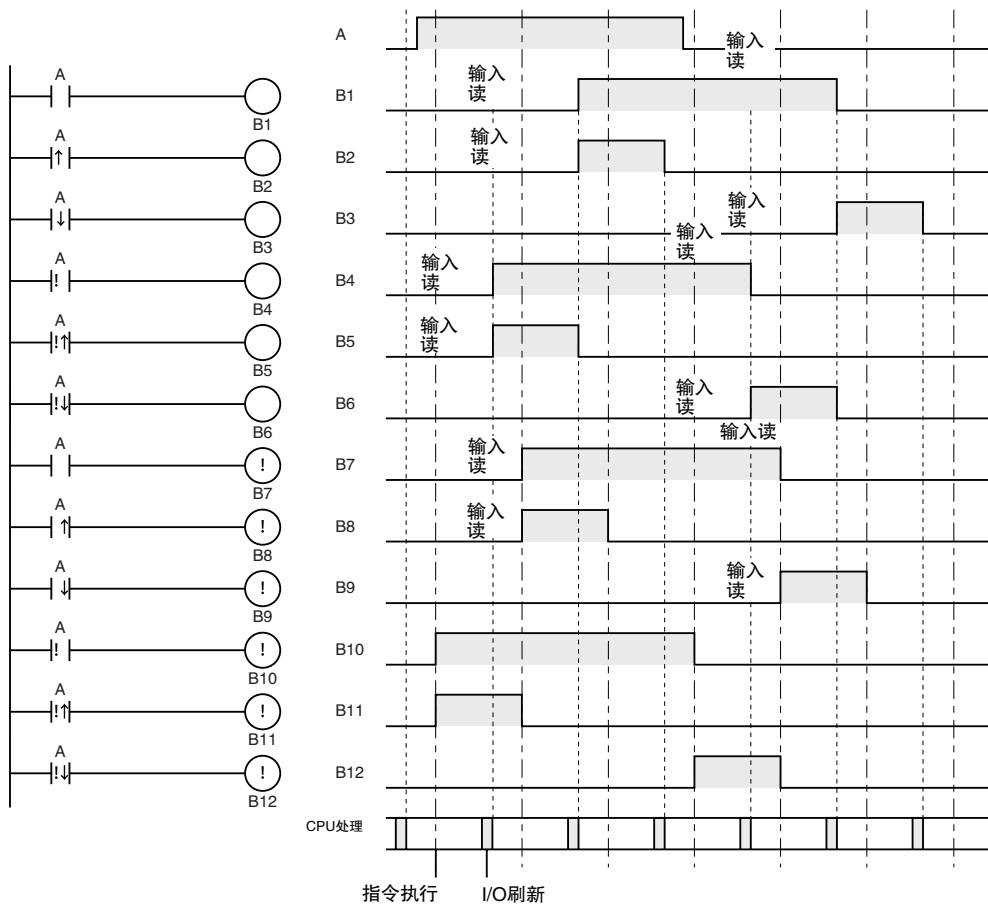
指令在每个循环中读取位状态、作比较、测试位或执行其它类型的处理, 并且当结果从 ON 变为 OFF 时, 将输出一个 OFF 执行条件 (能流停止)。执行条件将在下一个循环变为 ON。



例 仅当 CIO 1.03 从 ON 变为 OFF 时, 为一个循环生成 OFF 执行条件。

I/O 指令时序

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

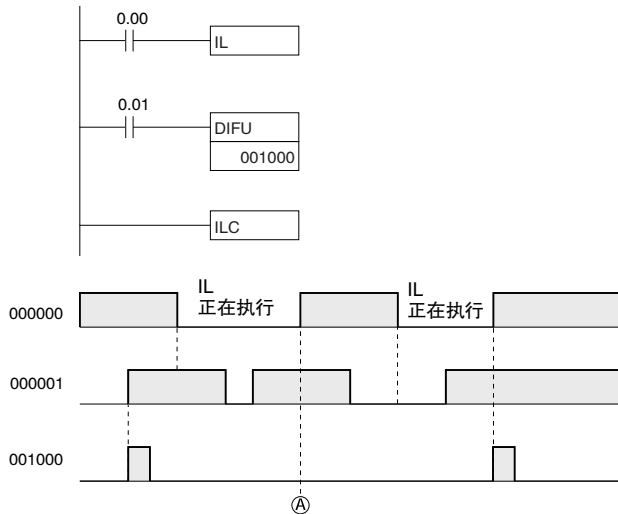


● 微分指令

- 微分指令带有用于辨别前值是 ON 还是 OFF 的内部标志。运行开始时，对上升沿微分指令 (DIFU 和 @ 指令) 的前值标志置 ON，并对下降沿微分指令 (DIFD 和 % 指令) 的前值标志置 OFF。这样可防止在运行开始时产生意外微分输出。
- 仅当执行条件为 ON 且前值标志为 OFF 时，上升沿微分指令 (DIFU 或 @ 指令) 将输出 ON。

在互锁中使用微分指令 (IL–ILC 指令)

在下例中，微分指令的前值标志保持先前的互锁状态值，且在 A 点不会输出一个微分输出，因为在互锁有效时该标志不会被刷新。



在跳转 (JMP(004)–JME(005) 指令) 中使用微分指令

与互锁相同，当指令跳转时，微分指令的前值标志不会改变，即保持前值。

- 对于下降沿微分指令 (DIFD(014) 或带 % 的指令)，仅当前值标志为 ON 时，输出会在输入变为 OFF 时置 ON。
- 对于上升沿和下降沿微分指令，输出均会在下一循环中置 OFF。



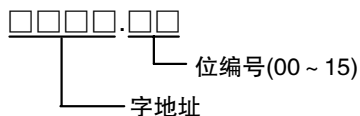
正确使用注意事项

请勿使用常 P_On 标志或 A200.11(首循环标志) 作为上升沿微分指令的输入位。请勿使用常 P_Off 标志作为下降沿微分指令的输入位。否则，该指令将永不执行。

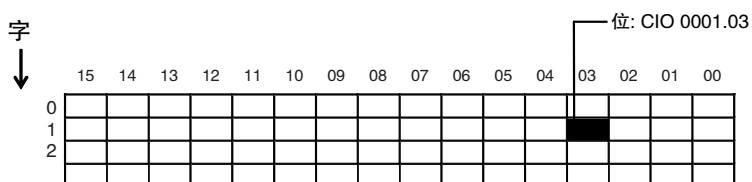
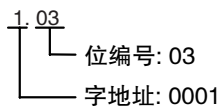
5-6-2 指定操作数

I/O 存储区寻址

● 位地址

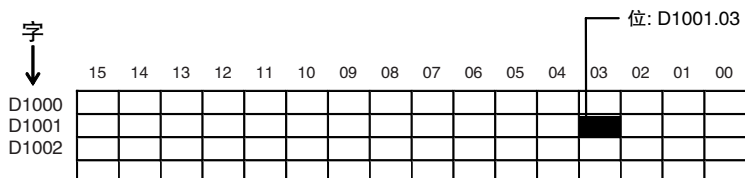
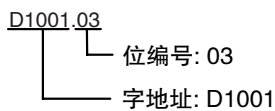


例：CIO 区中第 1 个字的第 03 位的地址如下所示。

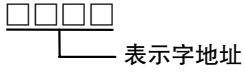


对于 CJ2 CPU 单元，位地址可在 DM 和 EM 区中指定。

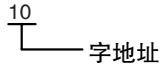
例：DM 区



● 字地址

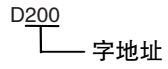


例：I/O 区

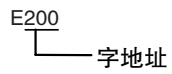


DM 和 EM 区地址在地址前加前缀 “D” 或 “E”，如下例中的地址 D200。

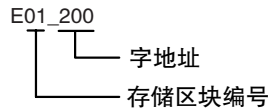
例：DM 区



例：EM 区



例：EM 区存储区块 1



指定操作数

操作数	描述	注释	应用示例
指定位地址	<p>通过直接指定字和位编号来定义位地址 (输入位)。</p> <p>*1</p>		
指定字地址	<p>通过直接指定字编号来定义 16 位字的地址。</p>		MOV 3 D200
指定位地址的偏移	<p>在括号内指定位编号来偏移指定的起始地址。</p> <p>也可用符号指定起始位地址。无论使用的是物理地址还是符号，只能使用保持区、工作区、DM 区和 EM 区地址。 可使用常数或 I/O 存储器中的字地址来指定偏移。若指定了字地址，则直接将字的内容用作偏移。</p>		
指定字地址的偏移	<p>在括号内指定位编号来偏移指定的起始地址。</p> <p>也可使用符号指定起始字地址。无论使用的是物理地址还是符号，只能使用保持区、工作区、DM 区和 EM 区地址。 可使用常数或 I/O 存储器中的字地址来指定偏移。若指定了字地址，则直接将字的内容用作偏移。</p>		MOV 3 D0[200]

*1 访问定时器 / 计数器完成标志和当前值时采用相同地址。任务标志也只有一个地址。

操作数	描述	注释	应用示例
以二进制模式指定间接 DM/EM 寻址 *2	<p>指定从数据区起始位置的偏移。将地址的内容作为二进制数 (00000 ~ 32767) 处理, 以指定 DM 区或 EM 区中的字地址。前面添加 @ 符号, 表示采用二进制模式指定间接寻址。</p> <p style="text-align: center;">@D□□□□□</p> <p style="text-align: center;">↓</p> <p>内容 00000 ~ 32767 (0000 Hex ~ 7FFF Hex, BIN)</p> <p style="text-align: center;">↓</p> <p>D </p>		
	如果 @D□□□□□ 的内容在 0000 ~ 7FFF Hex(十进制的 0 ~ 32767) 之间, 则地址将被定义为 D0 ~ D32767。	<p>@D300</p> <p style="text-align: center;"> </p> <p style="text-align: center;"> 内容</p> <p style="text-align: center;"> Hex: #0100</p> <p style="text-align: center;"> ↓</p> <p style="text-align: center;"> 定义为D256。</p> <p style="text-align: center;"> ——— 添加@符号。</p>	MOV #0001 @D300
	如果 @D□□□□□ 的内容在 8000 ~ FFFF Hex(十进制的 32768 ~ 65535) 之间, 则地址将被定义为 EM 区的存储区块 0 中的 E0_0 ~ E0_32767。	<p>@D300</p> <p style="text-align: center;"> </p> <p style="text-align: center;"> 内容</p> <p style="text-align: center;"> Hex: #8001</p> <p style="text-align: center;"> ↓</p> <p style="text-align: center;"> 定义为E0_01。</p>	
	如果 @E□□□□□□ 的内容在 0000 Hex ~ 7FFF Hex(0 ~ 32767) 之间, 则地址将被定义为指定存储区块中的 E□_0 ~ E□_32767。	<p>@E1_200</p> <p style="text-align: center;"> </p> <p style="text-align: center;"> 内容</p> <p style="text-align: center;"> Hex: #0101</p> <p style="text-align: center;"> ↓</p> <p style="text-align: center;"> 定义为E1_257。</p>	MOV #0001@E1_200
	如果 @E□□□□□□ 的内容在 8000 ~ FFFF Hex(十进制的32768 ~ 65535)之间, 则地址将被定义为紧随所指定的存储区块 @ 之后的存储区块中的 E(□+1)_0 ~ E(□+1)_32767。	<p>@E1_200</p> <p style="text-align: center;"> </p> <p style="text-align: center;"> 内容</p> <p style="text-align: center;"> Hex: #8002</p> <p style="text-align: center;"> ↓</p> <p style="text-align: center;"> 定义为E2_2。</p>	

*2 以二进制模式定义间接地址时, 请将 DM 区和 EM 区 (存储区块 0 ~ 18 Hex) 作为一系列地址对待。如果带 @ 符号的地址的内容超过 32767, 则该地址将被认为是 EM 区的存储区块 0 中从地址 0 开始的地址。

例: 如果 DM 区字的内容为 32768, 则地址将被定义为 EM 区的存储区块 0 中的 E0_0。

如果 EM 区的存储区块编号被指定为 “n” 且字的内容超过 32767, 则地址将被定义为扩展 EM 区的存储区块 n+1 中从 0 开始的地址。

例: 如果 EM 区的存储区块 2 的内容为 32768, 则地址定义为 EM 区的存储区块 3 中的 E3_0。

操作数	描述	注释	应用示例
以 BCD 模式指定间接 DM/EM 寻址	<p>指定从数据区起始位置的偏移。将地址的内容作为 BCD 数据 (0000 ~ 9999) 处理，以指定 DM 区或 EM 区中的字地址。在地址前添加星号 (*), 从而以 BCD 模式指定间接地址。</p> <p>*D□□□□</p> <p>↓</p> <p>内容 □□□□ 0000 ~ 9999 (BCD)</p> <p>↓</p> <p>D □□□□</p>	<p>*D200</p> <p>└─┬─┘</p> <p>└─┬─┘ #0100 内容</p> <p>↓</p> <p>定义为D100</p> <p>└─┬─┘</p> <p>└─┬─┘ 添加星号(*)</p>	MOV #0001 *D200

操作数	描述		注释	应用示例
直接定义寄存器	通过指定 IR□(□: 0 ~ 15) 或 DR□(□: 0 ~ 15) 来直接定义变址寄存器 (IR) 或数据寄存器 (DR)。		IR0	MOVR 1.02 IR0 将 CIO 0010 的 PLC 存储器地址存储到 IR0。
			IR1	MOVR 10 IR1 将 CIO 0010 的 PLC 存储器地址存储到 IR1。
使用寄存器定义间接地址	间接地址 (无偏移)	将定义为 IR@ 包含的 PLC 存储器地址的位或字。 指令操作数的位和字由 ,IR□ 定义。	,IR0	LD ,IR0 读取 IR0 作为 PLC 存储器地址的位。 MOV #0001 ,IR1 将 IR1 存储到 PLC 存储器的字 #0001。
			,IR1	
	常数偏移	将 IR□ 中的内容加或减一个常数作为 PLC 存储器地址的位或字。 将 +/- 常数指定到 ,IR□。常数偏移范围从 -2048 ~ +2047(十进制)。执行指令时, 偏移值转换为二进制数。	+5 ,IR0	LD +5, IR0 读取 IR0+5 作为 PLC 存储器地址的位。 MOV #0001 +31 ,IR1 将 IR1+31 存储到 PLC 存储器地址的字 #0001。
			31 ,IR1	
	DR 偏移	将 DR□ 内容加上 IR□ 内容作为 PLC 存储器地址的位或字。 指定 DR□, IR□。DR(数据寄存器) 内容将作为带符号二进制数处理。如果带符号二进制数为负值, 则对 IR□ 的内容进行负偏移。	DR0 ,IR0	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 中的值存储到 PLC 存储器地址的字 #0001。
自动递增	将 IR□ 中的内容递增 1 或 2 后作为 PLC 存储器地址。 +1: 定义为 ,IR□+ +2: 定义为 ,IR□++	,IR0++	LD, IR0 ++ 读取 IR0 中 PLC 存储器地址的位后, 将 IR0 的内容加 2。 MOV #0001 ,IR1 + 将 IR1 中的值存储到 PLC 存储器地址的字 #0001 后, 将 IR1 的内容加 1。	
		,IR1+		
自动递减	将 IR□ 中的内容递减 1 或 2 后作为 PLC 存储器地址。 -1: 定义为 ,-IR□ -2: 定义为 ,-IR□	,--IR0	LD, --IR0 将 IR0 的内容递减 2 后, 读取 IR0 中 PLC 存储器地址的位。 MOV #0001 ,-IR1 将 IR1 的内容减 1 后, 将 IR1 中的值存储到 PLC 存储器地址的字 #0001。	
		,IR1		

数据	操作数	数据格式	符号	范围	应用示例				
16 位常数	所有二进制数 或限定范围内的 二进制数	无符号二 进制	#	#0000 ~ #FFFF	MOV #0100 D0 将十六进制数 #0100(十进制数 &256) 存储到 D0 中。 +#0009 #0001 D1 将十六进制数 #000A(十进制数 &10) 存储到 D1 中。				
		带符号十 进制	±	-32768 ~ +32767	MOV -100 D0 将十进制数 -100(十六进制数 #FF9C) 存储到 D0 中。 +-9 -1 D1 将十进制数 -10(十六进制数 #FFF6) 存储到 D1 中。				
		无符号十 进制	&	&0 ~ &65535	MOV &256 D0 将十进制数 -256(十六进制数 #0100) 存储到 D0 中。 +&9 &1 D1 将十进制数 -10(十六进制数 #000A) 存储到 D1 中。				
	所有 BCD 数据 或限定范围内的 BCD 数据	BCD	#	#0000 ~ #9999	MOV #0100 D0 将 #0100(BCD) 存储到 D0 中。 +B #0009 #0001 D1 将 #0010(BCD) 存储到 D1 中。				
32 位常数	所有二进制数 或限定范围内的 二进制数	无符号二 进制	#	#00000000 ~ #FFFFFFFF	MOVL #12345678 D0 将十六进制数 #12345678 存储到 D0 和 D1 中。 <table border="1" style="margin-left: 20px;"> <tr> <td>D1</td> <td>D0</td> </tr> <tr> <td>1234</td> <td>5678</td> </tr> </table>	D1	D0	1234	5678
		D1	D0						
		1234	5678						
	带符号二 进制数	+	-2147483648 ~ +2147483647	MOVL -12345678 D0 将十进制数 -12345678 存储到 D0 和 D1 中。					
无符号十 进制	&	&0 ~ &4294967295	MOVL &12345678 D0 将十进制数 &12345678 存储到 D0 和 D1 中。						
所有 BCD 数据 或限定范围内的 BCD 数据	BCD	#	#00000000 ~ #99999999	MOVL #12345678 D0 将 #12345678(BCD) 存储到 D0 和 D1 中。					

数据	描述	符号	例	---																																										
文本串	<p>文本字符串数据以 ASCII 码的形式存储(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。</p> <p>如果字符的个数为奇数, 则将 00 Hex(NUL代码)保存到最后一个字的最低位上。</p> <p>如果字符的个数为偶数, 则将 0000 Hex(2个 NUL代码)保存到最后一个字 +1 的字的低位和高位空闲字节上。</p>	---	<p>↓ ABCDE</p> <table border="1"> <tr><td>'A'</td><td>'B'</td></tr> <tr><td>'C'</td><td>'D'</td></tr> <tr><td>'E'</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>45</td><td>00</td></tr> </table> <p>ABCD</p> <table border="1"> <tr><td>'A'</td><td>'B'</td></tr> <tr><td>'C'</td><td>'D'</td></tr> <tr><td>NUL</td><td>NUL</td></tr> </table> <p> </p> <table border="1"> <tr><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td></tr> <tr><td>00</td><td>00</td></tr> </table>	'A'	'B'	'C'	'D'	'E'	NUL	41	42	43	44	45	00	'A'	'B'	'C'	'D'	NUL	NUL	41	42	43	44	00	00	<p>MOV\$ D100 D200</p> <table border="1"> <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> <p>↓</p> <table border="1"> <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>	D100	41	42	D101	43	44	D102	45	00	D200	41	42	D201	43	44	D202	45	00
'A'	'B'																																													
'C'	'D'																																													
'E'	NUL																																													
41	42																																													
43	44																																													
45	00																																													
'A'	'B'																																													
'C'	'D'																																													
NUL	NUL																																													
41	42																																													
43	44																																													
00	00																																													
D100	41	42																																												
D101	43	44																																												
D102	45	00																																												
D200	41	42																																												
D201	43	44																																												
D202	45	00																																												

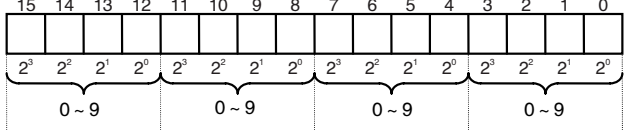
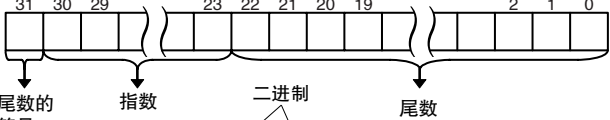

文本字符串中可使用的 ASCII 字符包括字母数字字符、日文片假名和符号(特殊字符除外)。这些字符如下表所示。

		高 4 位数位																		
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
低 4 位 数 位	0			Sp	0	@	P	\	p						一	タ	ミ			
	1			!	1	A	Q	a	q						。	ア	チ	ム		
	2			"	2	B	R	b	r						「	イ	ツ	メ		
	3			#	3	C	S	c	s						」	ウ	テ	モ		
	4			\$	4	D	T	d	t						、	エ	ト	ヤ		
	5			%	5	E	U	e	u						・	オ	ナ	ユ		
	6			&	6	F	V	f	v							ヲ	カ	ニ	ヨ	
	7			'	7	G	W	g	w							ア	キ	ヌ	ラ	
	8			(8	H	X	h	x							イ	ク	ネ	リ	
	9)	9	I	Y	i	y							ウ	ケ	ノ	ル	
	A			*	:	J	Z	j	z							エ	コ	ハ	レ	
	B			+	;	K	[k	{							オ	サ	ヒ	ロ	
	C			,	<	L	¥									ヤ	シ	フ	ワ	
	D			-	=	M]	m	}							ユ	ス	ヘ	ン	
	E			.	>	N	^	n	~							ヨ	セ	ホ	°	
	F			/	?	O	_	o								ツ	ソ	マ	°	

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
无符号二进制	<p>二进制 →</p> <p>Hex →</p> <p>十进制 →</p>	&0 ~ &65535	#0000 ~ #FFFF
带符号二进制数	<p>二进制 →</p> <p>Hex →</p> <p>十进制 →</p> <p>符号位: 0: 正, 1: 负</p> <p>使用最高位作为符号位, 将数据作为带符号的 16 位二进制数来处理。值采用 4 位十六进制表示。 正数: 如果最高位为 OFF, 则表示非负值。以 4 位十六进制表示的值的范围为 0000 ~ 7FFF。 负数: 如果最高位为 ON, 表示负值。对于 4 位十六进制数据, 值的范围为 8000 ~ FFFF Hex, 并用 2 的补码来表示该负值的绝对值(十进制)。</p>	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(二进制编码的十进制数)	 <p>二进制 →</p> <p>十进制 →</p>	#0 ~ #9999	#0000 ~ #9999
单精度十进制浮点数	 <p>尾数的符号</p> <p>指数</p> <p>二进制</p> <p>尾数</p> <p>值 = (-1)^{符号} × 1.[尾数] × 2^{指数} 符号(位31) 1: 负或0: 正</p> <p>尾数 从第00位到第22位的23位表示尾数, 即十进制数 1.□□□□.....的小数点后面的部分, 用二进制形式表示。</p> <p>指数 从第23位到第30位的8位表示指数。二进制形式的指数用 127+n表示2ⁿ。</p> <p>该格式符合单精度浮点数的IEEE754标准, 仅用于转换或计算浮点数的指令中。使用该格式时, 可通过 CX-Programmer 的 I/O 存储器编辑和监控画面来进行设定或监控。因而用户无需了解该格式的细节, 只需知道该格式占用 2 个字。</p>	---	---
双精度十进制浮点数	 <p>尾数的符号</p> <p>指数</p> <p>二进制</p> <p>尾数</p> <p>值 = (-1)^{符号} × 1.[尾数] × 2^{指数} 符号(位63) 1: 负或0: 正</p> <p>尾数 从第00位到第51位的52位表示尾数, 即十进制数 1.□□□□.....的小数点后面的部分, 用二进制形式表示。</p> <p>指数 从第52位到第62位的11位表示指数。二进制形式的指数用 1023+n表示2ⁿ。</p> <p>该格式符合双精度浮点数的 IEEE 754 标准, 仅用于转换或计算浮点数的指令中。使用该格式时, 可通过 CX-Programmer 的 I/O 存储器编辑和监控画面来进行设定或监控。因而用户无需了解该格式的细节, 只需知道该格式占用 4 个字。</p>	---	---



附加信息

· 补码

通常而言，基数 x 的补码是指用 $x-1$ 减去一个给定数的所有位，然后再在最低位加 1 所得到的数。(例：7556 以 10 为基数的补码为 $9999-7556+1=2444$ 。) 补码用于表达减法和其它功能 (例如加法)。

例：因为 $8954-7556=1398$ ， $8954+(7556 \text{ 以 } 10 \text{ 为基数的补数})=8954+2444=11398$ 。如果不计最高位，即可得到减法运算的结果为 1398。

· 二进制补码

二进制补码是以 2 为基数的补码。计算时，可用 $1(2-1=1)$ 减去所有位再加 1 计算得出。

例：二进制数 1101 的二进制补码为 $1111(\text{F Hex})-1101(\text{D Hex})+1(1 \text{ Hex})=0011(3 \text{ Hex})$ 。下表给出了该值用 4 位数十六进制表示的形式。

十六进制数 $a \text{ Hex}$ 的二进制补数 $b \text{ Hex}$ 的计算方法为： $\text{FFFF Hex}-a \text{ Hex}+0001 \text{ Hex}=b \text{ Hex}$ 。因此，十六进制数 $a \text{ Hex}$ 的二进制补数 $b \text{ Hex}$ 还可计算如下： $b \text{ Hex}=10000 \text{ Hex}-a \text{ Hex}$ 。

例：若要确定 3039 Hex 的二进制补数，可计算如下： $10000 \text{ Hex}-3039 \text{ Hex}=\text{CFC7 Hex}$ 。

另外，也可用 $a \text{ Hex}=10000 \text{ Hex}-b \text{ Hex}$ 的方法来计算 $b \text{ Hex}$ 的二进制补数 $a \text{ Hex}$ 。

若要从二进制补数 CFC7 Hex 求真值，可计算如下： $10000 \text{ Hex}-\text{CFC7 Hex}=3039 \text{ Hex}$ 。

CJ 系列有 2 条指令，即 $\text{NEG}(160)$ (二进制求补) 和 $\text{NEGL}(161)$ (双字二进制求补) 可用于从真值求二进制补数，或者从二进制补数求真值。

● 用单字数据表示的值

值 (十进制)	二进制表示			BCD 表示 (十进制)
	十进制表示		十六进制表示	
	无符号	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	

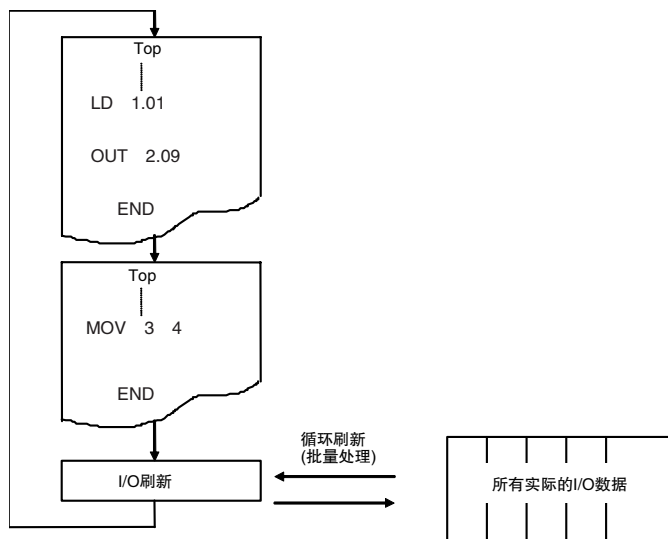
5-6-4 I/O 刷新时序

以下方法用于刷新外部 I/O。

- 循环刷新
- 即时刷新 (带 ! 规定的指令, IORF(097)、FIORF(225) 或 DLNK(226))

循环刷新

使用循环刷新时, 在执行完处于 READY 状态的所有循环任务时, 将立即执行 I/O 刷新。



如果某个任务中要求进行 I/O 刷新, 则应在执行 END(001) 指令之前, 先对所有要求的字执行 IORF(097) 指令。

即时刷新

使用即时刷新时，只要在程序中执行一条指令即进行刷新。

● 即时刷新的类型

执行即时刷新有两种方法：使用指令并指定即时刷新变化 (!)，或者使用即时刷新专用的特殊指令。

带刷新变化 (!) 的指令

可在执行指令时 (即循环期间) 使用指令的即时刷新变化 (!) 来执行刷新。

即时刷新专用的特殊指令

根据要刷新的存储器所在的单元而定，可使用下列三种指令。

IORF(097) 指令：基本 I/O 单元和高功能 I/O 单元 (仅限 CIO 区中的分配字)

FIORF(225) 指令：高功能 I/O 单元

DLNK(226)：CPU 总线单元

下表中列出了各单元与其对应的刷新指令。

单元	带即时刷新变化 (!) 的指令	IORF(097) 指令	FIORF(225) 指令	DLNK(226) 指令
基本 I/O 单元	允许	允许	不允许	不允许
CIO 区分配字				
高功能 I/O 单元	不允许	允许 *2	允许 *2	不允许
CIO 区分配字		不允许		
DM 区分配字				
CPU 总线单元	不允许	不允许	不允许	允许
CIO 区分配字				
DM 区分配字				
单元特定存储器 *1				
用于 CJ2M CPU 单元的脉冲 I/O 模块	允许 *3	不允许	不允许	不允许
CIO 区分配字				

*1 EtherNet/IP 单元、控制器链接单元、SYSMAC LINK 单元和 DeviceNet 远程 I/O 通信的数据链接。

*2 可通过 IORF(097) 或 FIORF(225) 指令进行刷新的高功能 I/O 单元中的 CIO 区分配字。FIORF(225) 指令的指令执行时间比 IORF(097) 指令短。另外使用也更为方便，因为只需在操作数中指定高功能 I/O 单元的单元号即可。

*3 只能够对对被设定为普通输入或快速响应输入的输入以及被设定为普通输出的输出进行即时刷新。



正确使用注意事项

即时刷新变化的执行时间比指令的常规变化要长，因此请务必小心，因为循环时间将变长。

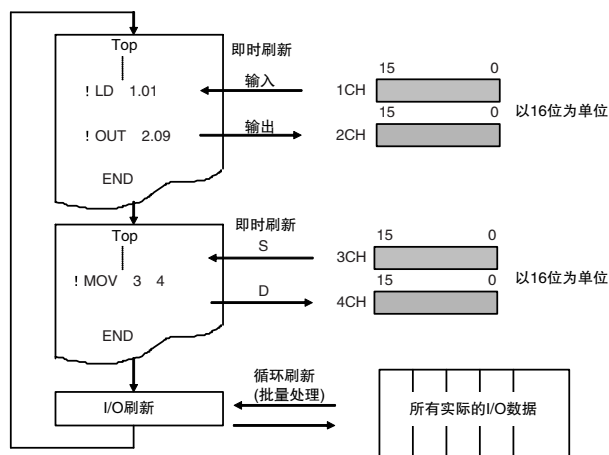
● 带刷新变化 (!) 的指令

请在指令前添加感叹号 (!) 以指定即时刷新。

- 如果将实际的 I/O 位指定为操作数，则指令执行时，I/O 将进行如下刷新。

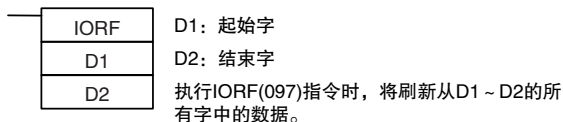
单元	被刷新的数据
基本 I/O 单元	将对包含该位的 16 个位进行 I/O 刷新。

- 当为某条指令指定了一个字操作数时，将对指定的 16 个位进行 I/O 刷新。
- 将在即将执行指令对输入或源操作数进行输入刷新。
- 将在刚执行指令之后对输出或目的地 (D) 操作数进行输出刷新。

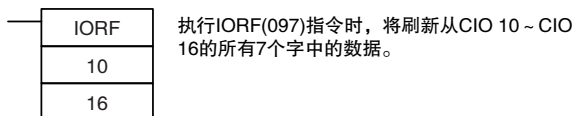


● IORF(097): I/O 刷新

IORF(097) 指令可用于刷新循环期间的实际 I/O 单元中的所有数据以及分配到高性能 I/O 单元的 CIO 区分配字。



例:



但在进行那些使用从基本 I/O 单元输入的数据或输出至基本 I/O 单元的数据进行的计算操作需要高速响应时，可在紧接着计算指令之前和之后使用 IORF(097)。

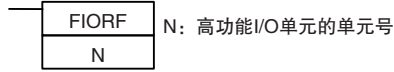


正确使用注意事项

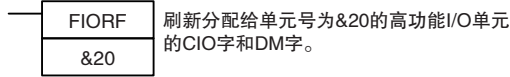
IORF(097) 具有相对较长的指令执行时间，并且该执行时间随刷新的字数成比例增加，因此将显著增加循环时间。请务必小心，避免循环时间变得过长。

● FIORF(225): 高性能 I/O 单元 I/O 刷新

FIORF(225) 指令可用于仅在必要时刷新指定单元号的高性能 I/O 单元中的下述数据。



例:

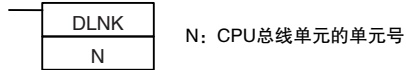


可在 PLC 设置中禁止高性能 I/O 单元的常规循环 I/O 刷新 (通过对该单元的高性能 I/O 单元循环刷新禁止位置 ON 来设定), 从而只在需要时通过执行 FIORF(225) 指令来对该单元进行 I/O 刷新。当高性能 I/O 单元连接到 PLC 中时, 该功能可防止 PLC 的循环时间增加。

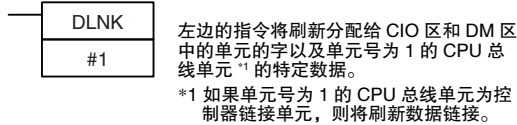
● 使用 DLNK(226): CPU 总线单元 I/O 刷新

DLNK(226) 指令可用于提高 CPU 总线单元的 CIO 区和 DM 区分配字以及指定到 CPU 总线单元的数据的刷新频率。*1

*1 CPU 总线单元的特定数据包括控制器链接单元或 SYSMAC LINK 单元以及 DeviceNet 单元的远程 I/O 的数据连接。



例:



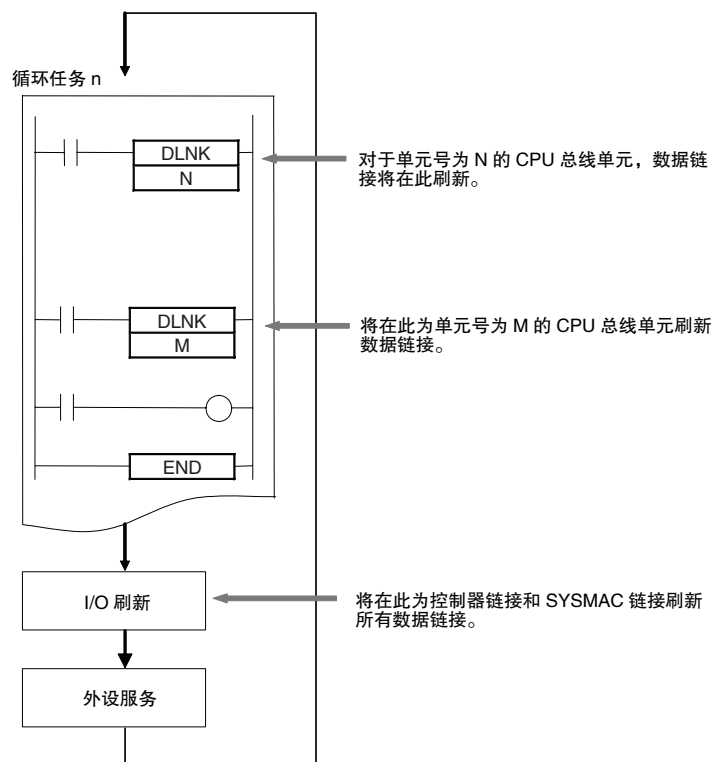
应用实例: 循环时间较长时, 控制器链接的数据链接的刷新间隔可能会非常长。可通过对控制器链接单元执行 DLNK(226) 指令以提高数据链接刷新频率的方式来缩短该间隔。



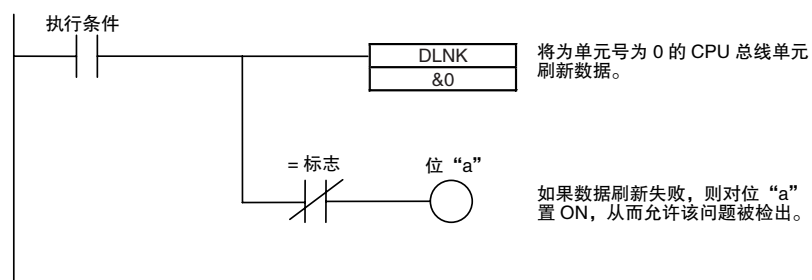
附加信息

下图所示为 DLNK(226) 指令的应用实例。

例：数据链接



如果对正在刷新数据的 CPU 总线单元执行 DLNK(226) 指令，则将不刷新数据，且等于标志将置 OFF。通常，等于标志应按下图方式进行编程，以确保刷新操作正常完成。

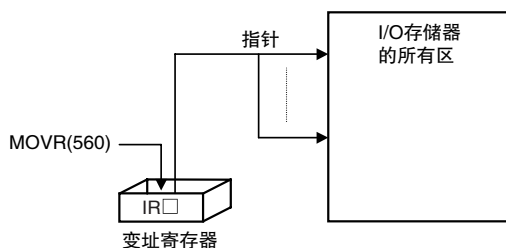


5-7 变址寄存器

5-7-1 何谓变址寄存器?

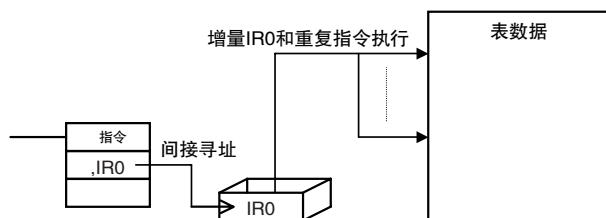
变址寄存器用作指定 PLC 存储器地址的指针,该地址为 I/O 存储器中的绝对存储地址。在通过 MOVR(560) 或 MOVRW(561) 指令将 PLC 存储器地址存储到变址寄存器中时,请将变址寄存器作为其它指令的操作数进行输入,从而对所存储的 PLC 存储器地址进行间接寻址。

变址寄存器的优势在于可指定 I/O 存储器中的任意位或字,包括定时器和计数器当前值(PV)。



5-7-2 使用变址寄存器

变址寄存器与 FOR-NEXT 循环组合使用时,功能非常强大。变址寄存器的内容可轻松递增、递减和偏移,因此使用循环中的少数几条指令即可高效处理连续数据表。



变址寄存器的基本使用步骤如下:

1. 使用 MOVR(560) 指令将指定位或字的 PLC 存储器地址存储到变址寄存器中。
2. 可在几乎所有指令中将变址寄存器指定为操作数,从而对所需的位或字进行间接寻址。
3. 对原 PLC 存储器地址进行偏移或递增(如下),从而使指针重新指向另一个地址。
4. 继续步骤 2 和 3,从而对任意数量的地址执行该指令。

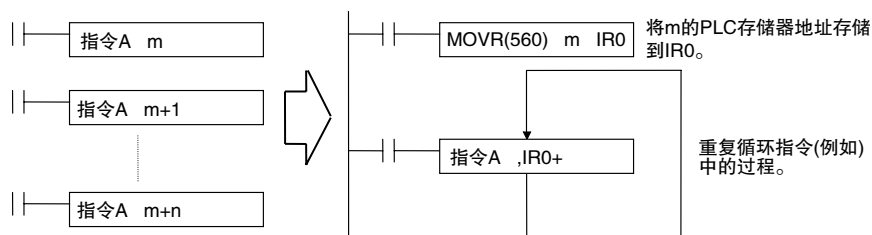
注 使用下述任一方法来使变址寄存器偏移或递增。

· 变址寄存器的间接寻址:

通过自动递增 (IR□+ 或 ,IR□++) 进行间接寻址、通过自动递减 (,-IR□ 或 ,,-IR□) 进行间接寻址,通过常数偏移 (常数 ,IR□) 进行间接参照,通过 DR 偏移 (DR□,IR□) 进行间接寻址。

- 变址寄存器的间接寻址指令：
二进制加 (+L)，二进制减 (-L)，双字二进制递增 (++L)，双字二进制递减 (--L)

例：



下例说明了用程序循环中的变址寄存器来取代一长串指令的方法。在这种情况下，指令 A 重复 n+1 次，以执行某些操作（例如读和比较值的表）。



正确使用注意事项

即使输入条件为 OFF，也将执行下述指令。因此，当在任何这些指令的某一操作数中使用自动递增或自动递减指令 (IR+ 或 IR-) 指定了间接存储器地址时，无论输入条件如何（在每个循环中递增还是递减），变址寄存器 (IR) 中的值均在每个循环中刷新。编写程序时必须考虑这一点。

类型	指令
顺序输入指令	LD、LD NOT、AND、AND NOT、OR、OR NOT、LD TST(350)、LD TSTN(351)、AND TST(350)、AND TSTN(351)、OR TST(350)、OR TSTN(351)
顺序输出指令	OUT、OUT NOT、DIFU(013)、DIFD(014)
顺序控制指令	JMP(004)、FOR(512)
定时器和计数器指令	TIM/TIMX(550)、TIMH(015)/TIMHX(551)、TMHH(540)/TMHHX(552)、TIMU(541)/TIMUX(556)、TMUH(544)/TMUHX(557)、TTIM(087)/TTIMX(555)、TIML(542)/TIMLX(553)、MTIM(533)/MTIMX(554)、CNT/CNTX(546)、CINTR(012)/CINTRX(548)
比较指令	符号比较指令 (LD、AND、OR = 等 (功能代码：300、305、310、320 和 325))
单精度浮点算术运算指令	单精度浮点数据比较(LD、AND、OR = F等(功能代码：329 ~ 334))
双精度浮点算术运算指令	双精度浮点数据比较(LD、AND、OR = D等(功能代码：335 ~ 340))
块编程指令	BPPS(811)、BPRS(812)、EXIT(806)、EXIT(806) NOT、IF(802)、IF(802) NOT、WAIT(805)、WAIT(805) NOT、TIMW(813)/TIMWX(816)、CNTW(814)/CNTWX(818)、TMHW(815)/TMHWX(817)、LEND(810)、LEND(810) NOT
文本串处理指令	串比较 (LD、AND、OR = \$ 等 (功能代码：670 ~ 675))

下述梯形图编程示例说明了如何处理变址寄存器。

例 1

梯形图程序：

```
LD P_Off
OUT, IR0+
```

运行：当 PLC 存储器地址 CIO 0.13 存储在 IR0 中时。

输入指令为 OFF(P_Off 为常 OFF 标志)，因此 OUT 指令将 CIO 0.13(该地址通过 IR0 间接寻址) 置为 OFF。将执行 OUT 指令，因此 IR0 将递增。从而将存储 PLC 存储器地址 CIO 0.14(通过使 IR0 中的地址递增 1 得到)。由此，在下一个循环中 OUT 指令将使 CIO 0.14 中的值变 OFF。

例 2

梯形图程序:

```
LD P_Off
SET, IR0+
```

运行: 当 PLC 存储器地址 CIO 0.13 存储在 IR0 中时。

输入条件为 OFF(P_Off 为常 OFF 标志), 因此不执行 SET 指令。由此, IR0 不递增, 且 IR0 中存储的值将保持为 PLC 存储器地址 CIO 0.13。

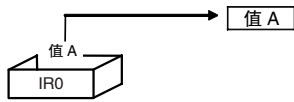


正确使用注意事项

- 若在功能块内部使用变址寄存器 IR0 ~ IR15, 则在其它功能块内部或功能块外部的程序中使用同一个变址寄存器将在两个实例间形成竞争, 并且程序将无法正常运行。因此, 在使用变址寄存器 (IR0 ~ IR15) 时, 请务必在功能块启动 (或者在使用变址寄存器之前) 的时间点保存变址寄存器的值, 将处理并入程序中, 从而使变址寄存器返回至保存的值。

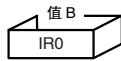
例: 启动功能块 (或使用变址寄存器之前):

1. 保存 IR 的值 (例如 A)。



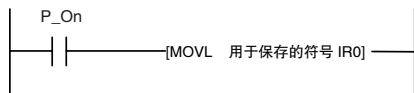
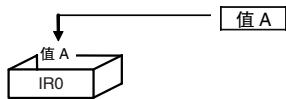
在功能块内部:

2. 使用 IR。



在功能块起始处 (或使用变址寄存器之前):

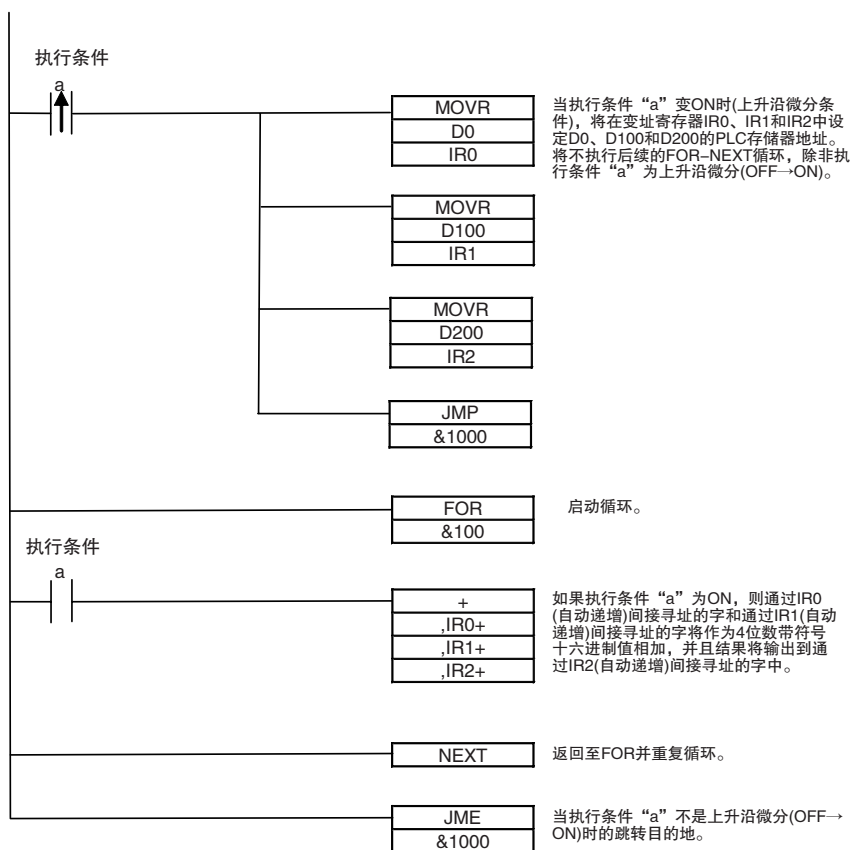
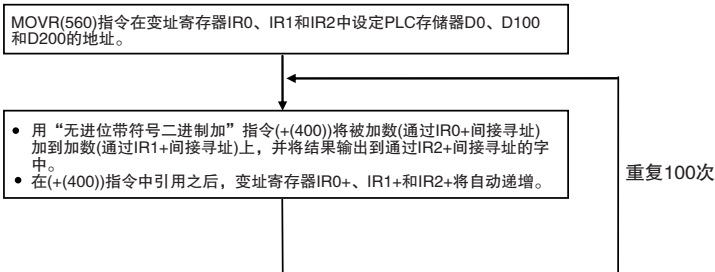
3. 将 IR 返回给保存的值 (例如 A)。



- 使用变址寄存器之前请务必设定值。若变址寄存器在未设定值的情况下使用, 则运行将不稳定。

变址寄存器的应用实例

将 D0 ~ D99(被加数) 加到 D100 ~ D199(加数) 上, 并将加法的结果输出到 D200 ~ D299。通过变址寄存器指定单条加法指令的操作数, 并通过使寄存器递增的方法来执行加法操作, 从而重复执行加法指令。



附加信息

变址寄存器仅可在下表所示的指令中进行直接寻址。

● 变址寄存器的直接寻址

对于寄存器 IR0 ~ IR15，变址寄存器的大小为每个寄存器两个字，因此请使用双字指令（助记符中带一个“L”符号）。

指令分组	指令名称	助记符	首要功能
数据传送指令	传送至寄存器	MOV(560)	将一个位或一个字的 PLC 存储器地址存储到变址寄存器中。
	传送定时器/计数器的当前值(PV)至寄存器	MOVW(561)	
表数据处理指令	设置记录位置	SETR(635)	输出存储在变址寄存器中的 PLC 存储器地址。
	获取记录号	GETR(636)	
跟踪指令	记录搜索指令	RSRCH□ (功能代码: 360 ~ 364, 370 ~ 374 和 380 ~ 384)	将符合条件的记录的 PLC 存储器地址输出到变址寄存器中。
	记录排序指令	RSORT□ (功能代码: 203, 205)	将比排好序的范围中的最后一个地址的 PLC 存储器地址大 1 的地址输出到变址寄存器。
数据传送指令	双字传送	MOVL(498)	在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	双字等于	=L(301)	
	双字不等于	<>L(306)	
	双字小于	<L(311)	
	双字与小于等于	<=L(316)	
	双字大于	>L(321)	
	双字与大于等于	>=L(326)	
	双字比较	CMPL(060)	
递增 / 递减指令	双字二进制递增	++L(591)	通过使变址寄存器中的内容递增、递减或偏移来改变该变址寄存器中的 PLC 存储器地址。
	双字二进制递减	--L(593)	
四则运算指令	无进位带符号双字二进制加	+L(401)	
	无借位带符号双字二进制减	-L(411)	
特殊指令	将 CV 值转换成地址值	FRMCV(284)	将 PLC 存储器地址在 CVM1/CV 系列和 CJ 系列地址之间进行转换。
	将地址值转换成 CV 值	TOCV(285)	

5-7-3 与变址寄存器相关的处理

CJ 系列 CPU 单元的表数据处理指令来补充变址寄存器的功能。表数据处理指令大体上可分为堆栈处理指令和表处理指令。

处理过程		目的	指令
堆栈处理		操作 FIFO(先进先出)或 LIFO(后进先出)数据表,然后读、写、插入、删除或对数据表中的数据条目进行计数。	SSET(630)、PUSH(632)、FIFO(633)、LIFO(634)、SREAD(639)、SWRIT(640)、SINS(641)、SDEL(642)、SNUM(638)
表处理	带单字记录的表 (范围指令)	基本处理	寻找整型或浮点型十进制值,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	执行多种其它表处理,例如比较或排序。 将变址寄存器与 SRCH(181)、MAX(182)、MIN(183)等指令以及比较指令组合使用。
	带多字记录的表	搜索和排序记录用户注册的指定区域(字和范围的编号)内的记录。 对记录用户注册的指定区域(字和范围的编号)内的记录执行操作。	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 RSORT4(205) 将下列指令与变址寄存器组合使用。 · DIM(631)、SETR(635)和 GETR(636)指令 · 其它指令(例如比较指令)

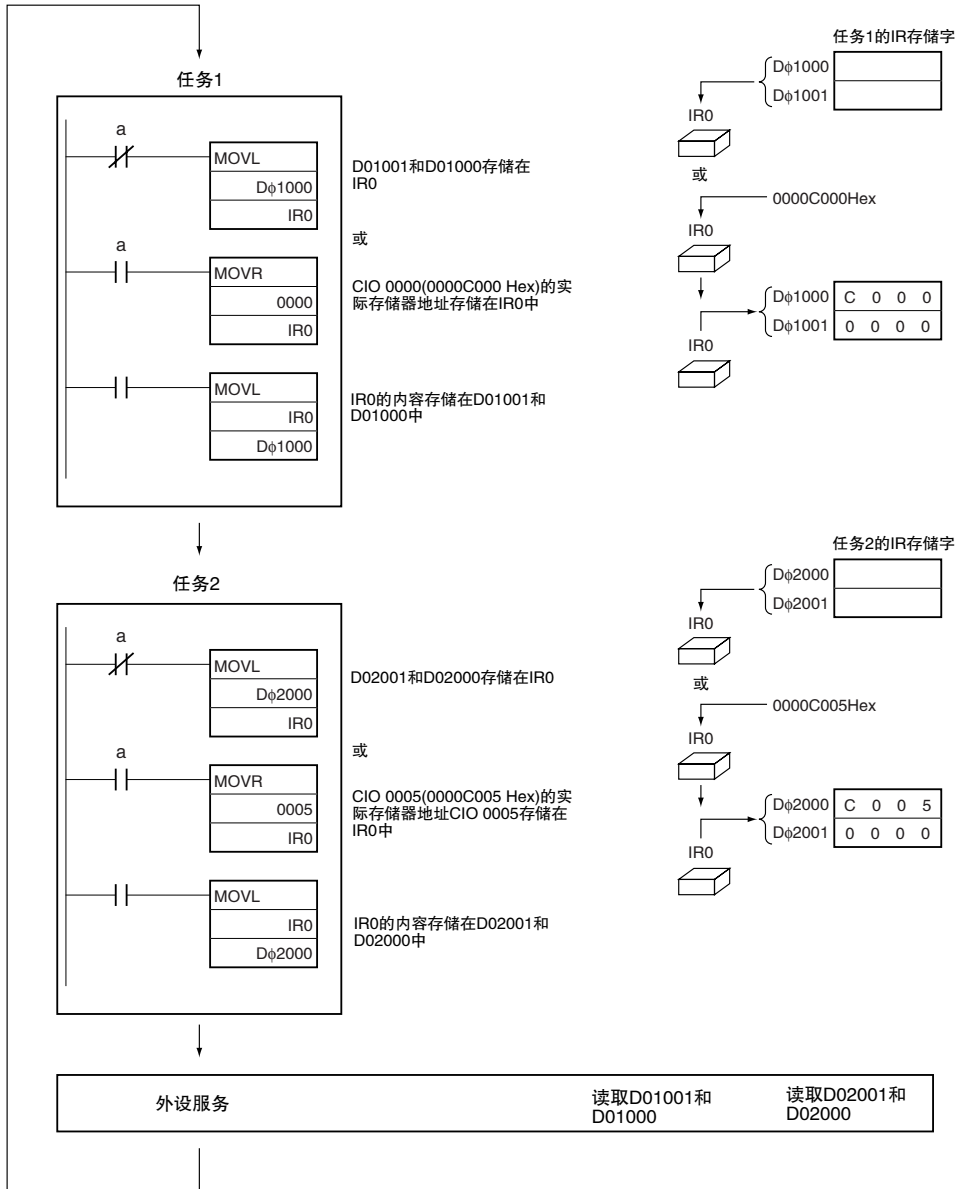
5-7-4 监控变址寄存器

可通过下述方法来监控变址寄存器：

- 使用 CX-Programmer 来为每个任务监控最终的变址寄存器值。
- 使用上位链接指令或 FINS 指令来监控变址寄存器的值，将程序编写为在各任务结束时将来自各任务的变址寄存器值存储到另一区域（例如 DM 区），以及在各任务开始时从存储字（例如 DM 区）读取变址寄存器的值。随后即可使用 CX-Programmer、上位链接命令或 FINS 命令编辑在其它区域（例如 DM 区）中为各任务存储的值。

● 例

注 请务必使用变址寄存器中的 PLC 存储器地址。



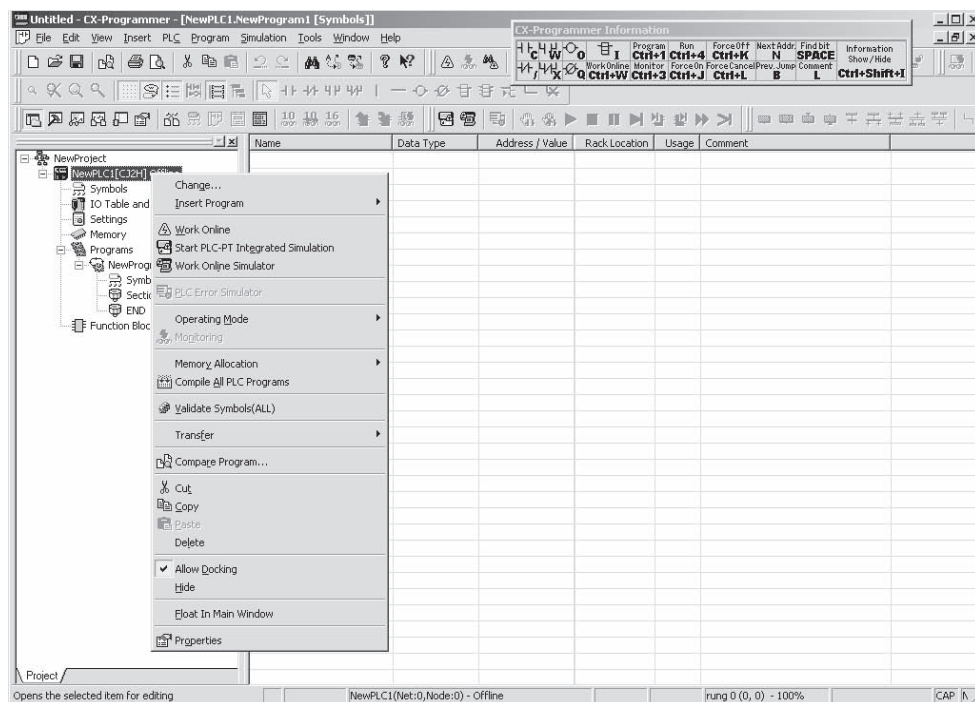
5-7-5 在多任务之间共享变址寄存器和数据寄存器

变址寄存器和数据寄存器 (IR/DR) 可由多任务共享。通常的设定是各任务分开使用各自的寄存器。当前设定可在 A99.14 中确认。

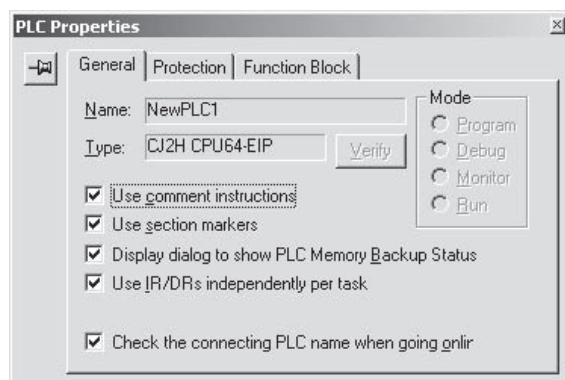
● 设定方法

使用 CX-Programmer 来设定共享变址寄存器和数据寄存器。无法从手持式编程器进行该项设定。

1. 在 CX-Programmer 项目树形图中选择一个 PLC，然后单击鼠标右键。



2. 选择“Properties” (属性)。此时将显示下述“PLC Properties” (PLC 属性) 对话框。



如果各任务要求分开使用各自的变址寄存器和数据寄存器，请保留对每个任务独立使用 IR/DR 的勾选标记。若要对所有任务使用共享的变址寄存器和数据寄存器，则去除勾选标记。

● 辅助区标志和字

名称	地址	描述
多任务之间的 IR/DR 操作	A99.14	表示是否在多任务之间共享变址寄存器和数据寄存器。 0: 对各任务分开使用寄存器 (默认) 1: 对所有任务共享寄存器



附加信息

- 当两个或两个以上的任务需要使用相同的内容时，可使用共享变址寄存器和数据寄存器，从而免去在多任务之间存储和载入寄存器内容的操作。有关存储和载入变址寄存器内容的说明，请参考“6-19 变址寄存器”。
- 共享变址寄存器和数据寄存器时，任务之间的切换时间将变短。如果不使用寄存器，或者没有在各任务中独立使用寄存器的特别需要，则建议设定共享寄存器。

5-8 指定地址偏移

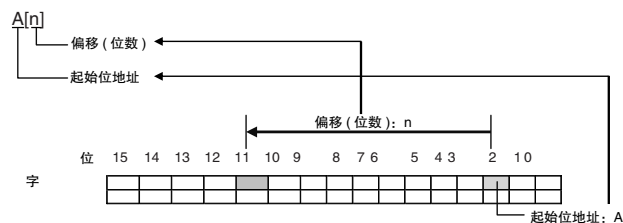
5-8-1 概述

在指令操作数中指定了地址时，可通过在指定地址后面的括号中指定偏移量的方式来改变指定地址。



● 位地址

从 A (起始位地址) 开始利用 n 指定的数 (位数) 表示位地址偏移。



起始位地址

可使用位地址或符号 (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 区中的地址进行偏移操作。I/O 注释表示对该起始位地址的 I/O 注释。

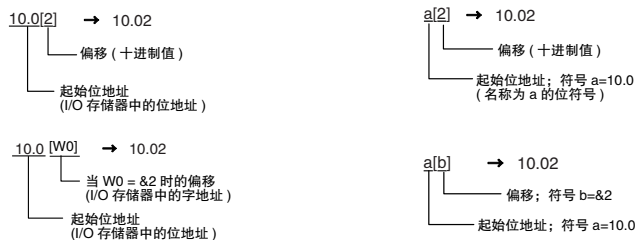
偏移量

偏移可定义为十进制常数、字地址 (但不能指定 CIO 区地址) 或单字符 (即: INT、UINT、WORD 和 CHANNEL 数据类型的符号)。如果指定字地址, 则指定字的内容用作偏移。

如果指定字中的偏移超出了位 15, 则偏移将从下一个字中的位 00 继续。如果间接指定了偏移, 请务必通过输入比较指令或其它指令来确保末尾位地址未超出上限。

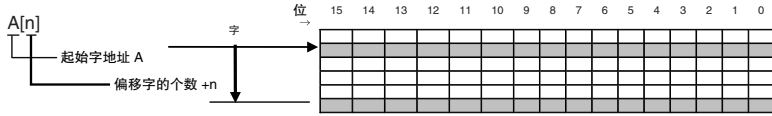
如果偏移位的位数超出了起始位地址的存储区, 则末尾位地址将位于下一个存储区中, 顺序由实际的 PLC 存储器地址决定。(有关存储器映射中的存储区安排的说明, 请参考“**A-4 PLC 存储器地址的存储器映射**”。)

例:



● 字地址

从 A(起始字地址) 开始利用 n 指定的数 (偏移字数) 表示字地址偏移。



起始字地址

可使用字地址或符号 (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 区中的地址进行偏移操作。I/O 注释表示对该起始字地址的 I/O 注释。

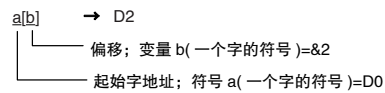
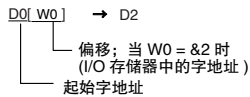
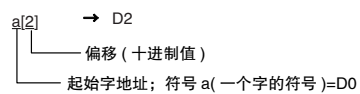
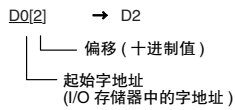
偏移量

偏移可定义为十进制常数、字地址 (但不能指定 CIO 区地址) 或单字符 (即: INT、UINT、WORD 和 CHANNEL 数据类型的符号)。如果指定字地址, 则指定字的内容用作偏移。

如果间接指定了偏移, 请务必通过输入比较指令或其它指令来确保末尾位地址未超出上限。

如果偏移字的字数超出了起始字地址的存储区, 则末尾字地址将位于下一个存储区中, 顺序由实际的 PLC 存储器地址决定。(有关存储器映射中的存储区安排的说明, 请参考“A-4 PLC 存储器地址的存储器映射”。)

例:



⚠ 注意

编程时, 若在梯形图程序中使用符号或地址来直接指定偏移量, 则不得超出起始地址的存储区范围。

如果间接指定导致地址超出起始地址存储区, 则系统将访问下一个区中的数据, 并可能出现意外操作。

⚠

5-8-2 地址偏移应用示例

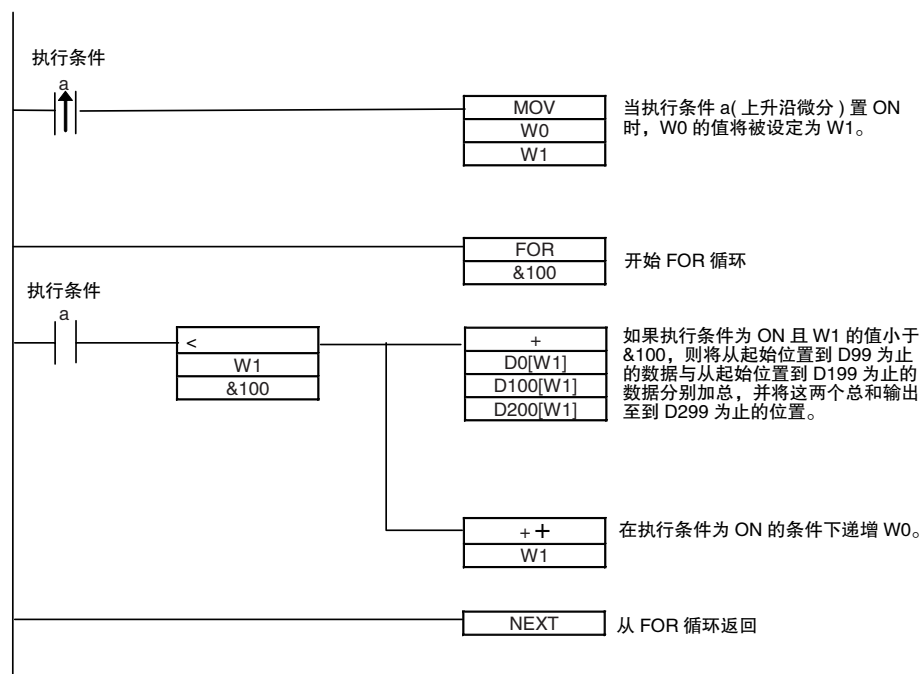
只需在 I/O 存储器中对括号中的偏移指定字地址，即可动态指定偏移。而指定字地址中的内容将用作偏移。例如，通过递增括号中的值，仅使用一条指令就能够使地址增加，从而执行相应的操作。

● 梯形图示例

该例中使用了两个连续数据区，即 D0 ~ D99 和 D100 ~ D199。将从指定的起始点 W0 开始到区域末尾为止的对应字的内容相加，并将和输出到从 D200 偏移指定地址开始的 D200 ~ D299 中的相应位置。例如，如果 W0 为 30，则将 D30 ~ D99 和 D130 ~ D199 的对应字相加，然后将和输出到 D230 ~ D299 中。



执行每个处理时都采用输入比较指令 (<) 作为执行条件，使 W1 不会超出 &100 的范围，从而确保不会超出间接寻址范围的上限。



5-9 检查程序

CJ 系列程序的检查有如下四个阶段。

- CX-Programmer 输入操作期间的输入检查
- 由 CX-Programmer 进行的程序检查
- 执行期间的指令检查
- 执行期间的致命错误检查 (程序出错)

5-9-1 CX-Programmer 输入期间的错误

程序将由 CX-Programmer 在下述时间自动进行检查。

检查时间	检查
输入梯形图时	指令输入、操作数输入、编程模板
加载文件时	所有指令和所有编程模板的所有操作数
下载文件时	CJ 系列所支持的型号和所有指令的所有操作数
联机编辑时	容量等

检查结果将输出至 “Output” (输出) 窗口的文本选项卡。此外,非法程序段的左侧母线将在梯形图视图中以红色显示。

5-9-2 通过 CX-Programmer 检查程序

在 CX-Programmer 中可对用户程序进行检查。对程序进行检查时,用户可为程序检查指定 A、B、C 或自定义检查等级这四个等级 (按错误的严重性顺序) 中的任一个。

CX-Programmer 不检查指令中间接寻址的操作数的范围错误。如果指令的操作数无效,则在指令执行检查期间 ER 标志将变 ON,这部分内容将在下一节中讲述。详情请参考《CS/CJ/NSJ 系列指令参考手册》(样本编号: W474)。

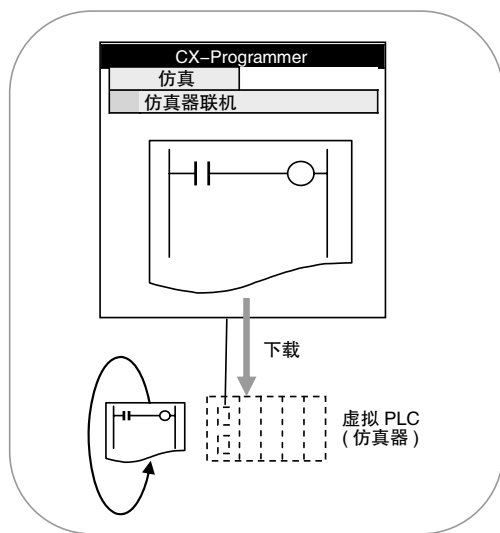
有关 CX-Programmer 检查的详情,请参考《CX-Programmer 操作手册》(样本编号: W446)。

5-9-3 通过仿真器进行调试

无需连接至实际的 PLC，而只需在计算机上仿真 CPU 单元的运行，即可对程序进行调试。

检查梯形图程序的运行

通过从 CX-Programmer 启动 CX-Simulator 中的仿真器，即可用虚拟 PLC 对已创建的程序进行检查和调试。

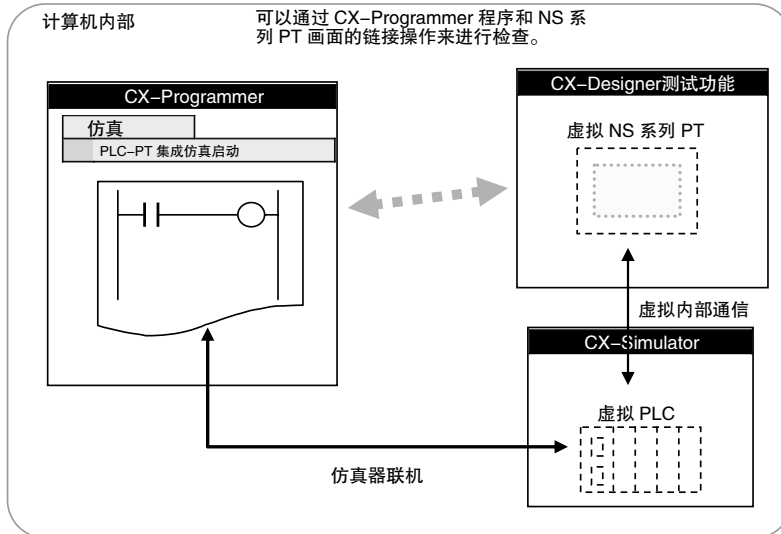


除了传送程序和监控之外，使用仿真器还可实现下述功能。有关调试步骤的详情，请参考《CX-Programmer 操作手册》(样本编号：W446)。

- 执行步运行、连续步运行或扫描运行。
- 指定中断点、起始点和 I/O 中断条件。
- 检查各任务的执行步数和执行时间。
- 仿真执行中断任务。
- 对位进行强制置位和强制复位。

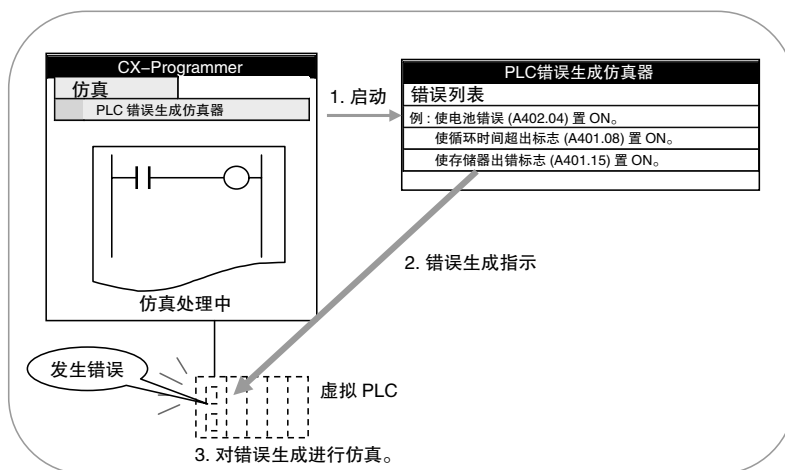
通过 PT 和梯形图程序之间的运行进行调试：集成仿真

使用 PLC-PT 集成仿真，即可实现在 NS 系列 PT 的画面数据和 CJ 系列 PLC 程序之间的调试运行。该功能通过链接 CX-programmer 的仿真器功能和 CX-Designer PT 画面设计软件的脱机测试功能来实现。从而允许仅使用计算机而不必通过线缆连接计算机和 PT，即可从梯形图程序对画面进行调试和画面控制。PLC-PT 集成仿真功能还可从 CX-Programmer 启动。有关调试步骤的说明，请参考《CX-Programmer 操作手册》(样本编号：W446)。



错误仿真功能

使用 CX-Programmer，可在梯形图程序仿真过程中通过虚拟 PLC 生成系统错误。当发生 PLC 系统错误时，通过使用特殊操作窗口生成所需的致命或非致命系统错误，即可轻松检查 NS 系列 PT 上的梯形图程序的运行情况。有关调试步骤的说明，请参考《CX-Programmer 操作手册》(样本编号：W446)。



注 与发生实际错误时不同，即使使用 PLC 错误发生仿真功能生成了致命错误时，梯形图也不会停止执行。



附加信息

另外也可在 PLC 中通过使用 FAL(006) 或 FALS(007) 指令来生成系统错误。

5-9-4 程序执行检查

正在执行程序时，可使用 CX-Programmer 来执行下列检查。

指令执行期间可执行下列四项检查。

错误类型	发生错误时变 ON 的标志	停止 / 继续运行
指令处理错误	ER 标志 如果指定了“Stop Operation”（停止运行），则在发生错误时，指令处理出错标志 (A295.08) 也将变 ON。	PLC 设置中有一项设定可用于指定在发生指令处理错误时是停止还是继续运行。默认设定是继续运行。 仅当指定了“Stop Operation”时，才会生成程序错误并停止运行。
访问错误	AER 标志 如果指定了“Stop Operation”（停止运行），则在发生错误时，访问出错标志 (A295.10) 也将变 ON。	PLC 设置中有一项设定可用于指定在发生指令处理错误时是停止还是继续运行。默认设定是继续运行。 仅当指定了“Stop Operation”时，才会生成程序错误并停止运行。
非法指令错误	非法指令出错标志 (A295.14)	致命错误 (程序错误)
用户程序区溢出错误	用户程序区溢出出错标志 (A295.15)	致命错误 (程序错误)

指令处理错误 (P_ER 标志 ON 错误)

- 在执行某条指令时若提供了不正确的数据，或者试图执行任务外部的指令，则将产生指令处理错误。在此，将会在指令处理开始时检查所需数据，结果是不执行指令，P_ER 标志 (出错标志) 将变 ON，并且 P_EQ 和 P_N 标志根据指令而定可能会保持不变或者变 OFF。如果指令 (包括输入指令) 正常结束，则 P_ER 标志 (出错标志) 将变 OFF。使 P_ER 标志变 ON 的条件将因指令不同而不同。详情请参考《CS/CJ/NSJ 系列指令参考手册》(样本编号: W474) 中对各条指令的说明。
- 如果在 PLC 设置中将“Instruction Errors” (指令错误) 设定为“Stop Operation” (停止运行)，则在发生指令处理错误和 P_ER 标志变 ON 时，运行将停止 (致命错误) 且指令处理出错标志 (A295.08) 将变 ON。

非法访问错误 (P_AER 标志 ON 错误)

- 非法访问错误表示在访问用于指定指令的操作数的地址时，通过下述方式之一访问了错误区域。
 - 对参数区执行了读或写操作。
 - 对未安装的存储区执行了写操作 *1
 - 对指定为 EM 文件存储器的 EM 区存储区块执行了写操作。
 - 对只读区执行了写操作。
 - 在间接 DM/EM 地址中以 BCD 模式指定的值不是 BCD (例如 *D1 中的内容为 #A000)。
 - 若发生访问错误，则指令处理将继续且出错标志 (ER 标志) 将不会变 ON，但访问出错标志 (P_AER 标志) 将变 ON。
- *1 发生下述情况时将产生访问错误：
- 当指定的 EM 地址超过 32767 (例如 E32768) 时 (对于当前存储区块)。
 - 对间接 EM 地址以 BIN 模式指定了最后一个存储区块，且指定字中的内容为 8000 ~ FFFF Hex (例如 @EC_00001 中的内容为 #8000)。
 - 对间接 EM 地址以 BIN 模式指定了当前存储区块，且指定字中的内容为 8000 ~ FFFF Hex (例如 @EC_00001 中的内容为 #8000)。
 - 将内容为位的内部存储器地址的 IR 寄存器用作字地址时，或者将内容为字的内部寄存器地址的 IR 寄存器用作位地址时。

- 如果在 PLC 设置中将 “Instruction Errors” (指令错误) 设定为 “Stop Operation” (停止运行), 则在发生非法访问错误和 AER 标志变 ON 时, 运行将停止(致命错误)且非法访问出错标志(A295.10)将变 ON。



附加信息

任务执行完之后, 不清除访问出错标志 (P_AER 标志)。如果将 “Instruction Errors” (指令错误) 设定为 “Continue Operation” (继续运行), 则可监控该标志, 直到即将执行 END(001) 指令前为止, 从而检查任务程序中是否发生了非法访问错误。(如果在 CX-Programmer 上对 AER 标志进行监控, 则可监控在执行完整用户程序之后 P_AER 标志的最终状态。)

其它错误

● 非法指令错误

非法指令出错标志试图执行未在系统中定义的指令数据。只要程序是通过 CX-Programmer 创建的, 则通常不会发生该错误。

但在极少的情况下, 该错误的确会发生。该错误将被作为程序错误处理, 运行将停止 (致命错误), 且非法指令标志 (A295.14) 将变 ON。

● 用户程序区溢出错误

用户程序区溢出错误表示试图执行存储地址超出定义为程序存储区的用户程序区中的最后一个地址的指令数据。只要程序是通过 CX-Programmer 创建的, 则通常不会发生该错误。

但在极少的情况下, 该错误的确会发生。该错误将被作为程序错误处理, 运行将停止 (致命错误), 且 UM 溢出标志 (A295.15) 将变 ON。



附加信息

如果出错标志 (P_ER) 或非法访问出错标志 (P_AER) 变 ON, 则该错误将作为程序错误处理, 并且可用于停止 CPU 单元运行。在 PLC 设置中指定对于程序错误的操作。

● 程序错误

程序错误	描述	相关标志
无 END 指令	程序中未出现 END 指令	无 END 标志 (A295.11) 变 ON。
任务执行期间出错	循环中没有准备就绪的任务。 没有程序分配到某个任务。 尽管满足中断任务的执行条件, 但对应的中断任务号不存在。	任务出错标志 (295.12) 变 ON。
指令处理错误 (P_ER 标志 ON) 且在 PLC 设置中为 “Instruction Errors” (指令错误) 设定了 “Stop Operation” (停止运行)	在试图执行一条指令时, 在操作数中提供了错误的数值。	如果在 PLC 设置中为 “Instruction Errors” (指令错误) 设定了 “Stop Operation” (停止运行), 则 ER 标志将变 ON 且指令处理出错标志 (A295.08) 将变 ON。
非法访问错误 (P_AER 标志 ON) 且在 PLC 设置中为 “Instruction Errors” (指令错误) 设定了 “Stop Operation” (停止运行)	对参数区执行了读或写操作。 对未安装的存储区执行了读或写操作。 对指定为 EM 文件存储器的 EM 区存储区块执行了读或写操作。 对只读区执行了写操作。 在间接 DM/EM 地址中以 BCD 模式指定的值不是 BCD。	如果在 PLC 设置中为 “Instruction Errors” (指令错误) 设定了 “Stop Operation” (停止运行), 则 AER 标志将变 ON 且非法访问出错标志 (A295.10) 将变 ON。
间接 DM/EM BCD 错误并在 PLC 设置中为 “Instruction Errors” (指令错误) 设定了 “Stop Operation” (停止运行)	在间接 DM/EM 地址中以 BCD 模式指定的值不是 BCD。	访问出错标志将变 ON。如果在 PLC 设置中选择了 “Instruction Error” (指令错误) 复选框中的 “Stop CPU” (停止 CPU), 则间接 DM/EM BCD 出错标志 (A295.09) 也将变 ON。
溢出错误的微分地址	在联机编辑过程中, 插入或删除了 131,071 条以上的微分指令。	微分溢出出错标志 (A295.13) 变 ON。
非法指令错误	试图执行一条无法执行的指令。	UM(用户存储器)溢出标志 (A295.14) 变 ON。
UM(用户存储器)溢出错误	试图执行存储地址超出定义为程序存储区的用户存储器 (UM) 的最后一个地址的指令数据。	UM(用户存储器)溢出标志 (A295.15) 变 ON。

5-10 注意事项

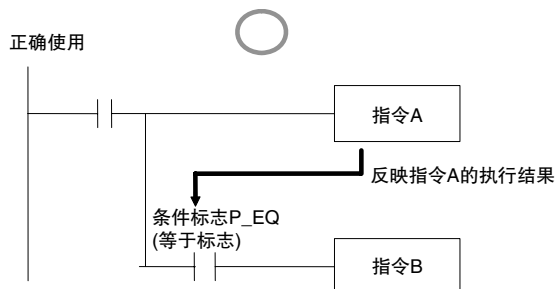
5-10-1 条件标志

使用条件标志

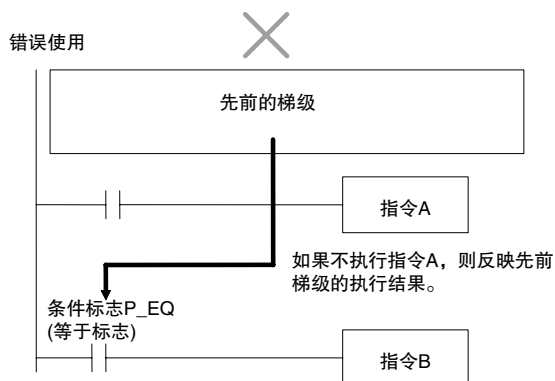
条件标志由所有指令共享，并且在循环期间将根据各指令的执行结果而改变。因此，请务必在具有相同执行条件的分支输出上，在刚执行完指令的位置使用条件标志，以反映指令执行结果。切勿将条件标志直接连接至母线，否则将导致条件标志反映其它指令的执行结果。

例：使用指令 A 的执行结果

助记符



对指令 A 和 B 使用同一个执行条件 (a)，从而根据指令 A 的执行结果来执行指令 B。在这种情况下，仅当执行了指令 A 时，才会根据条件标志来执行指令 B。

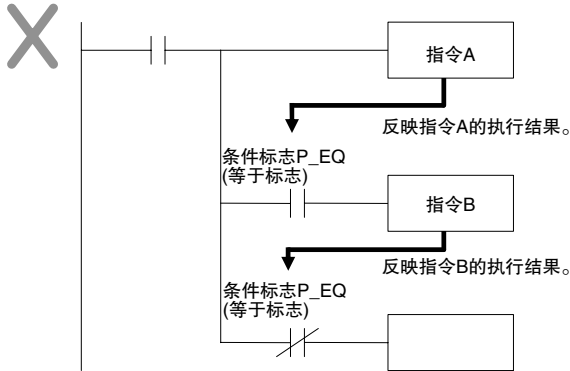


当将条件标志直接连接到左侧母线时，如果未执行指令 A，则将根据前一个梯级的执行结果来执行指令 B。

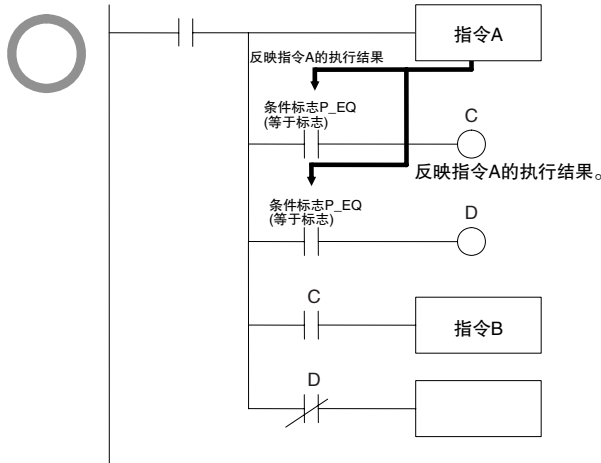
正在使用中断任务时，如果满足中断任务的启动条件，则即使正处于循环任务执行期间，中断任务也将运行。在这种情况下，即使条件标志在中断任务中变为 ON/OFF，当处理从中断任务切换回循环任务时，条件标志也将返回其原状态。

● 使用 N.C. 和 N.O. 输入中的执行结果

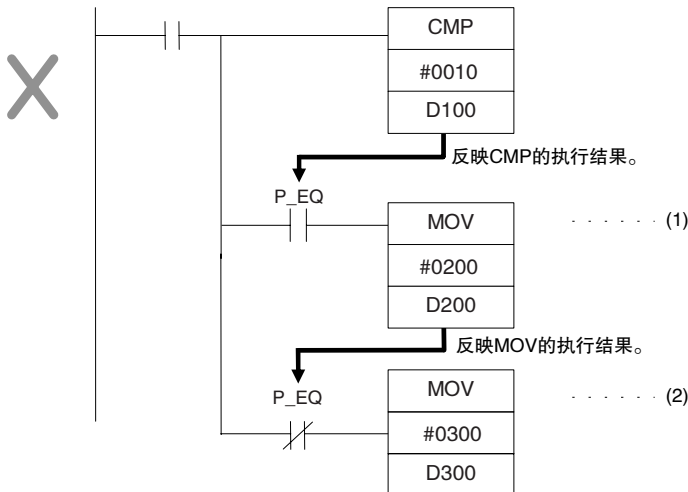
如下例所示，即使从同一个输出分支执行 N.C. 和 N.O. 输入位，条件标志也将取用指令 B 的执行结果。



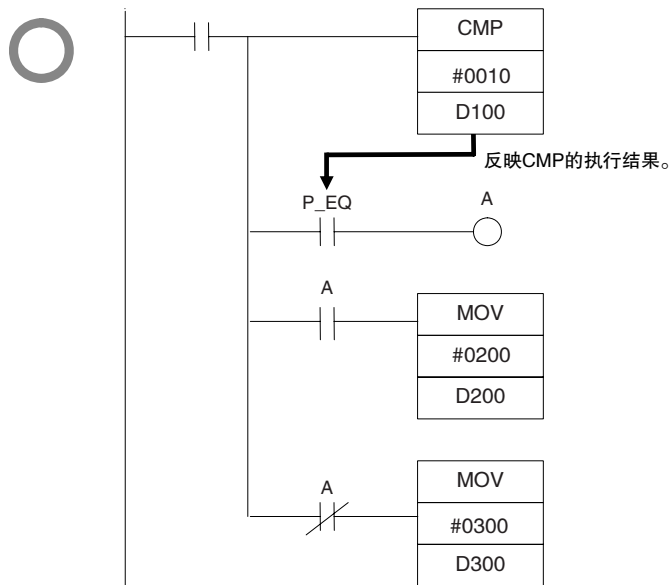
请务必使每个结果均由一条输出指令取用，以确保不取用指令 B 的执行结果。



例：下例中，如果 D100 的内容为 #0010，则将 #0200 移至 D200；如果 D100 的内容不为 #0010，则将 #0300 移至 D300。



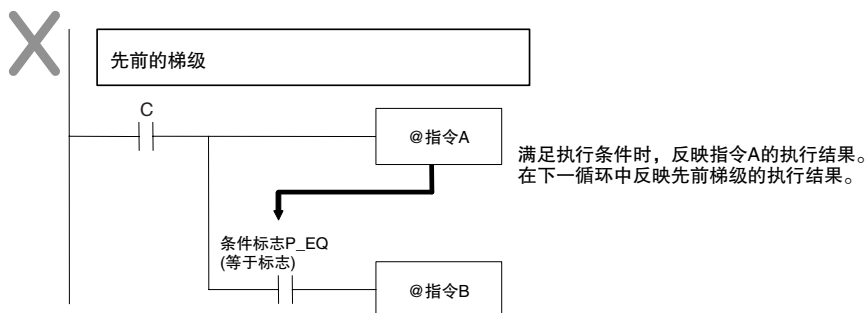
如果上面梯级中的 D100 的内容为 #0010，则等于标志将变 ON。指令 (1) 将 #0200 移至 D200，但随后等于标志变为 OFF，因为 #0200 源数据不为 0000 Hex。随后将执行 (2) 处的 MOV 指令，并将 #0300 移至 D300。因此必须在下方插入一个梯级，以防止取用第一条 MOV 指令的执行结果，如下所示。



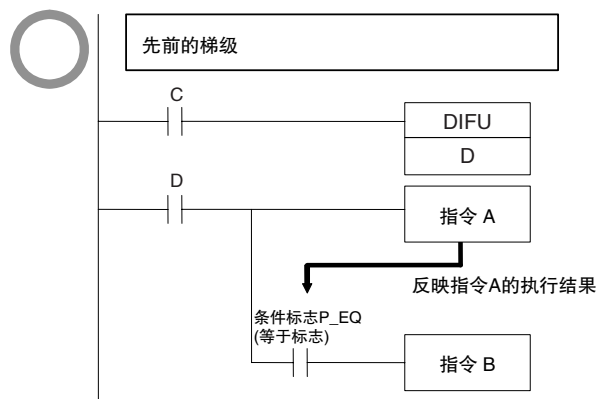
● 使用来自微分指令的执行结果

若使用微分指令，则仅当满足执行条件时，微分指令的执行结果将反映在条件标志中，且前一个梯级的结果（而非微分指令的执行结果）将反映在下一个循环的条件标志中。因此如果要使用微分指令的执行结果，则必须了解下一个循环中条件标志的变化情况。

下例中，仅当满足执行条件 C 时，将执行指令 A 和 B，但若指令 B 取用来自指令 A 的执行结果，则将发生下述问题。如果在执行了指令 A 之后，执行条件 C 在下一个循环中保持 ON，则当条件标志由于反映前一个梯级的执行结果而从 OFF 变为 ON 时，指令 B 的执行情况将出乎意料（因执行条件所致）。



因而在该情况下，指令 A 和 B 不是微分指令，将使用 DIFU(或 DIFD) 指令，如下所示，从而使指令 A 和 B 均以上升沿 (或下降沿) 微分条件仅执行一个循环。



附加信息

可使用条件标志保存和条件标志载入 (CCS(282) 和 CCL(283)) 指令来保存和载入条件标志的状态。这些指令可用于在同一个任务的其它位置或另一个任务中访问条件标志的状态。

主条件置 ON 条件标志

● 出错标志 (P_ER)

在特殊条件下 (例如某条指令的操作数不正确时)，出错标志将变 ON。当出错标志变 ON 时，将不执行指令。

当出错标志变 ON 时，其它条件标志 (例如 <、>、OF 和 UF 标志) 的状态将不改变，且 = 和 N 标志的状态将因指令不同而不同。

有关会导致出错标志变为 ON 的条件，请参考《CS/CJ/NSJ 系列指令参考手册》(样本编号：W474) 中对于各条指令的描述。请务必小心，因为某些指令无论条件如何，均会使出错标志变 OFF。



附加信息

PLC 设置中对于发生指令错误时的设定可决定在出错标志变 ON 时运行是否停止。默认设定下，当出错标志变 ON 时，运行将继续。如果指定了当出错标志变 ON 时停止运行且运行停止 (作为程序出错处理)，则运行停止点的程序地址将存储在 A298 ~ A299 中。与此同时，A295.08 将变 ON。

● 等于标志 (P_EQ)

等于标志是对于所有指令的临时性标志，除了当比较结果为等于 (=) 时。该标志由系统自动设定，并且会变化。等于标志在由前一条指令置 ON(OFF) 之后，可以再由后续指令置 OFF(ON)。例如，当 MOV 指令或其它传送指令将 0000 Hex 作为源数据进行传送时，等于标志将变 ON，而在所有其它情况下将变 OFF。即使某条指令将等于标志置 ON，传送指令也将立即执行，并根据传送指令的源数据是否为 0000 Hex 来决定将等于标志置 ON 还是置 OFF。

● 进位标志 (P_CY)

CY 标志用于移位指令、带进位输入的加法和减法指令、产生借位和进位的加法和减法指令，并和高功能 I/O 单元指令、PID 指令以及 FPD 指令一起使用。请留心以下注意事项。

- CY 标志用于移位指令、带进位输入的加法和减法指令、产生借位和进位的加法和减法指令，并和高功能 I/O 单元指令、PID 指令以及 FPD 指令一起使用。请留心以下注意事项。
- CY 标志可由某条特定指令的执行结果置 ON(OFF)，并由另一条指令置 OFF(ON)。使用进位标志时，请确保该标志中反映所需的结果。

● 小于和大于标志 (P_LT, P_GT)

> 和 < 标志用于比较指令以及 LMT、BAND、ZONE、PID 和其它指令中。

> 或 < 标志即使由某条特定指令的执行结果置 ON(OFF)，也可由另一条指令置 OFF(ON)。

● 负标志 (P_N)

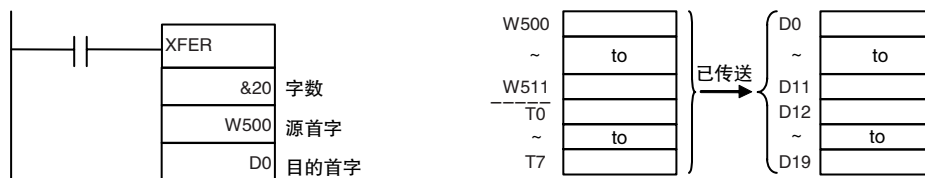
对于某些特定指令，当指令执行结果字的最左位为“1”时，N 标志将置 OFF；而对于其它指令，N 标志将无条件置 OFF。

● 为多个字指定操作数

即使指定需要多个字的操作数时使得该操作数的所有字并非在同一个区域内，指令也将按所编写的方式执行。在这种情况下，字将按在 PLC 存储器地址中的顺序取用。出错标志将不变 ON。

举例而言，如果指定传送从 W500 开始的 20 个字，考虑通过 XFER(070) 指令执行块传送的结果。在此，将超出以 W511 结尾的工作区，但该指令仍将执行，而不会使出错标志置 ON。在 PLC 存储器地址中，定时器的当前值将在工作区之后的存储器中保持，并且对于下述指令，W500 ~ W511 将被传送到 D0 ~ D11，T0 ~ T7 的当前值将被传送到 D12 ~ D19。

注 有关特定 PLC 存储器地址的说明，请参考附录“PLC 存储器地址的存储器映射”。



5-10-2 特殊程序段

CJ 系列程序具有可控制指令条件的特殊程序段。有下列特殊程序段可供使用。

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 指令间的子程序段。
IL-ILC 段	IL 和 ILC 指令	段被互锁	输出位变为 OFF，且定时器复位。不执行其它指令并保持先前状态。
步梯形图段	STEP 指令		
FOR-NEXT 循环	FOR 指令和 NEXT 指令	处理中断。	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	块程序正在执行。	执行位于 BPRG 和 BEND 之间以助记符列出的块程序。

指令组合

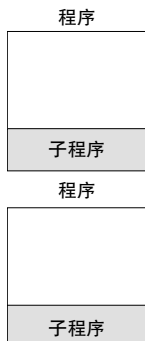
下表列出了可用于其它程序段内部的特殊指令。

	子程序	IL-ILC 段	步梯形图段	FOR-NEXT 循环	JMP0-JME0 段	块程序段
子程序	不支持	不支持	不支持	不支持	不支持	不支持
IL-ILC	支持	不支持	不支持	支持	支持	不支持
步梯形图段	不支持	支持	不支持	不支持	支持	不支持
FOR-NEXT 循环	支持	支持	不支持	支持	支持	不支持
JMP0-JME0	支持	支持	不支持	不支持	不支持	不支持
块程序段	支持	支持	支持	不支持	支持	不支持

注 无法在其它任务的程序中使用指定程序区的指令。详情请参考“5-2-4 指定任务”。

子程序

在所有程序中，将所有子程序一起放在紧接着 END(001) 指令之前并且在除子程序以外的程序段之后的位置。(因此，不能将子程序放在步梯形图、块程序、FOR-NEXT 或者 JMP0 - JME0 段中。)如果在子程序 (SBN ~ RET) 之后放了除子程序之外的程序段，则该程序段将不执行。



子程序中不可使用的指令

不能将下述指令放在子程序中。

功能	助记符	指令
过程步控制	STEP(008)	定义步梯形图段
	SNXT(009)	经步梯形图的步

注 子程序中可包括块程序段。但如果在执行从子程序返回至主程序时块程序处于 WAIT 状态，则块程序段在下次被调用时将保持 WAIT 状态。

步梯形图程序段中不可使用的指令

步梯形图程序段中不支持下述指令。

功能	助记符	指令
顺序控制	FOR(512)、NEXT(513) 和 BREAK(514)	FOR-NEXT 循环和循环中断
	END(001)	结束
	IL(002) 和 ILC(003)	互锁和互锁清除
	JMP(004) 和 JME(005)	跳转和跳转结束
	CJP(510) 和 CJPN(511)	条件跳转和条件跳转非
	JMPO(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)	子程序入口和子程序返回
块程序	IF(802)(NOT)、ELSE(803) 和 IEND(804)	条件块分支 (非)、条件块分支 ELSE 和条件块分支结束
	BPRG(096) 和 BEND(801)	块程序开始 / 结束
	EXIT(806)(NOT)	条件块退出 (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 步梯形图程序可在互锁段 (在 IL 和 ILC 之间) 内使用。当互锁为 ON 时, 步梯形图将完全复位。
 2 步梯形图程序段可用于多路跳转 (JMPO) 和多路跳转结束 (JME0) 之间。

块程序段中不支持的指令

不能将下列指令放入块程序段中。

按功能分类	助记符	指令
顺序控制	FOR(512)、NEXT(513) 和 BREAK(514)	FOR-NEXT 循环和循环中断
	IL(002) 和 ILC(003)	互锁和互锁清除
	JMPO(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
顺序输出	DIFU	上升沿微分
	DIFD	下降沿微分
	KEEP	保持
	OUT	输出
	OUT NOT	反相输出
定时器 / 计数器	TIM	100ms 定时器
	TIMH	10ms 定时器
	TMHH(540)	1ms 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)	子程序入口和子程序返回
数据移位	SFT	移位
梯形图步控制	STEP(008) 和 SNXT(009)	步定义和步启动
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 块程序可在步梯形图程序段中使用。
 - 2 块程序可在互锁段 (在 IL 和 ILC 之间) 中使用。当互锁为 ON 时, 块程序段将不执行。
 - 3 块程序段可用于多路跳转 (JMPO) 和多路跳转结束 (JME0) 之间。
 - 4 跳转指令 (JMP) 和条件跳转指令 (CJP/CJPN) 可在块程序段中使用。跳转 (JMP) 和跳转结束 (JME) 指令以及条件跳转 (CJP/CJPN) 和跳转结束 (JME) 指令在块程序段中必须成对使用, 否则无法使用。除非这些指令成对使用, 否则程序将无法正常运行。

6

I/O 存储区

本章节阐述了 CPU 单元中的 I/O 存储区。I/O 存储器是内置于 CPU 单元的一种存储器。

6-1	I/O 存储区	6-2
6-1-1	I/O 存储区概述	6-2
6-1-2	I/O 存储区结构	6-3
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	同步数据刷新区	6-13
6-5	CPU 总线单元区	6-14
6-6	高性能 I/O 单元区	6-15
6-7	脉冲 I/O 区	6-16
6-8	串行 PLC 链接区	6-17
6-9	DeviceNet 区	6-18
6-10	工作区	6-19
6-11	保持区	6-20
6-12	辅助区	6-21
6-13	暂存继电器区	6-22
6-14	数据存储区	6-23
6-15	扩展数据存储区	6-26
6-16	定时器区	6-30
6-17	计数器区	6-32
6-18	任务标志	6-33
6-19	变址寄存器	6-34
6-20	数据寄存器	6-39
6-21	条件标志	6-41
6-22	时钟脉冲	6-43

6-1 I/O 存储区

6-1-1 I/O 存储区概述

I/O 存储区可通过指令操作数进行访问。下表列出了 I/O 存储器中的各个区。

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,如各种单元的 I/O 刷新。未分配给单元的字可在程序中作为工作字和工作位使用。指定 CIO 区中的地址时,无需输入“CIO”前缀。	---	
I/O 区	I/O 区中的字分配给基本 I/O 单元的外部 I/O 端子。	6-7	
数据链接区	数据链接区中的字用于通过 Controller Link 单元与网络中的其它 PLC 单元进行数据链接。	6-12	
同步数据刷新区	使用同步单元运转功能时,同步数据刷新区中的字作为同步刷新数据进行分配。	6-13	
CPU 总线单元区	CPU 总线单元区中的字分配给 CPU 总线单元,用于传送状态信息。	6-14	
高性能 I/O 单元区	高性能 I/O 单元区中的字分配给高性能 I/O 单元,用于传送状态信息。	6-15	
脉冲 I/O 区	在连接脉冲 I/O 模块的情况下,脉冲 I/O 区中的字分配给各项脉冲 I/O 功能。仅 CJ2M CPU 单元支持该区。	6-16	
串行 PLC 链接区	该区为串行 PLC 链接专用。通过串行端口即可实现 CPU 单元之间的数据交换,而无需通信编程。仅 CJ2M CPU 单元支持该区。	6-17	
DeviceNet 区	DeviceNet 区中的字分配给从站,用于实现 DeviceNet 远程 I/O 通信。该区中的分配为固定形式,不可变更。	6-18	
内部 I/O 区 *1	这些字只能在程序中使用,而不得用于与外部 I/O 端子进行 I/O 交换。	---	
工作区 (W) *1	工作区中的字只能在程序中使用。	6-19	
保持区 (H)	保持区中的字只能在程序中使用。当 PLC 的电源接通或运行模式在 PROGRAM 和 RUN 或 MONITOR 间切换时,该区中的字将保持其内容。	6-20	
辅助区 (A)	辅助区中包含用于监视和控制 PLC 运行的各个标志和控制位。	6-21	
暂存继电器区 (TR)(TR 区)	TR 区包含记录程序分支 ON/OFF 状态的位。TR 位仅可在使用助记符编程时使用。	6-22	
数据存储区 (D) (DM 区)	DM 区为多功能数据区。对于 CJ2 PLC 而言,DM 区可采用字或位为单位进行读写。当 PLC 的电源接通或运行模式在 PROGRAM 和 RUN 或 MONITOR 间切换时,该区中的字将保持其内容。 DM 区中的某些字用于存储高性能 I/O 单元和 CPU 总线单元的初始设定。	6-23	
扩展数据存储区 (E) (EM 区)	EM 区为多功能数据区。对于 CJ2 PLC 而言,EM 区可采用字或位为单位进行读写。当 PLC 的电源接通或运行模式在 PROGRAM 和 RUN 或 MONITOR 间切换时,该区中的字将保持其内容。EM 区划分为多个存储区块。	6-26	
定时器区	定时器完成标志区 (T)	当定时器设定时间已过时,完成标志将置 ON。	6-30
	定时器当前值(PV)区(T)	定时器的当前值 (PV) 随定时器的运行而增大或减小。	
计数器区	计数器完成标志区 (C)	当计数器达到设定值或减量计数至零时,完成标志置 ON。	6-32
	计数器当前值(PV)区(C)	计数器的当前值 (PV) 随计数器的运行而增大或减小。	
任务标志区 (TK)	当相应的循环任务处于“READY”状态时,任务标志置 ON。	6-33	
变址寄存器 (IR)	变址寄存器用于存储 PLC 存储器地址 (RAM 中的绝对地址),以实现 I/O 存储器字的间接寻址。	6-34	
数据寄存器 (DR)	对字进行间接寻址时,数据寄存器用于偏移变址寄存器中的 PLC 存储器地址。	6-39	
条件标志	条件标志是一种特殊的标志(如出错标志和进位标志),用于表明指令的执行结果。	6-41	
时钟脉冲	时钟脉冲是一种按照一定时间间隔置 ON 和 OFF 的特殊标志。	6-43	

*1 CIO 区中的内部 I/O 区和工作区两个区均可提供工作位,但请首先使用工作区中的工作位。

6-1-2 I/O 存储区结构

区	大小	范围	外部 I/O 分配	按位访问	按字访问	访问		通过编程设备进行修改	启动时或模式变更时的状态	强制位状态	参考	
						读	写					
C I O 区	I/O 区	2,560 位 (160 字)	CIO 0 ~ CIO 159*1	基本 I/O 单元	允许	允许	允许	允许	允许	清除*2	允许	6-7
	数据链接区	3,200 位 (200 字)	CIO 1000 ~ CIO 1199	数据链接或 PLC 链接 (附有条件规定)	允许	允许	允许	允许	允许		允许	6-12
	同步数据刷新区	1,536 位 (96 字)	CIO 1200 ~ CIO 1295	同步单元*3	允许	允许	允许	允许	允许		允许	6-13
	CPU 总线单元区	6,400 位 (400 字)	CIO 1500 ~ CIO 1899	CPU 总线单元 (附有条件规定)	允许	允许	允许	允许	允许		允许	6-14
	高性能 I/O 单元区	15,360 位 (960 字)	CIO 2000 ~ CIO 2959	高性能 I/O 单元 (附有条件规定)	允许	允许	允许	允许	允许		允许	6-15
	脉冲 I/O 区	32 位 (4 字)	CIO 2960 ~ CIO 2963	脉冲 I/O 模块*4	允许	允许	允许	允许	允许		允许	6-16
	串行 PLC 链接区	1,440 位 (90 字)	CIO 3100 ~ CIO 3189	已链接的 PLC*4	允许	允许	允许	允许	允许		允许	6-17
	DeviceNet 区	9,600 位 (600 字)	CIO 3200 ~ CIO 3799	DeviceNet 主站 (固定分配) (附有条件规定)	允许	允许	允许	允许	允许		允许	6-18
	内部 I/O 区	3,200 位 (200 字) 37,504 位 (2,344 字)	CIO 1300 ~ CIO 1499 CIO 3800 ~ CIO 6143	---	允许	允许	允许	允许	允许		允许	---
工作区	8,192 位 (512 字)	W000 ~ W511	---	允许	允许	允许	允许	允许	清除*2	允许	6-19	
保持区*5	8,192 位 (512 字)	H000 ~ H511	---	允许	允许	允许	允许	允许	保持	允许	6-20	
辅助区	48,128 位 (3,008 字)	A000 ~ A447	---	允许	允许	允许	不允许	不允许	取决于地址	不允许	6-21	
		A448 ~ A959	---	允许	允许	允许	允许	允许				
		A960 ~ A1471*6	---	允许	允许	允许	允许	允许				
		A10000 ~ A11535*6	---	允许	允许	允许	不允许	不允许				

*1 只需通过修改分配给指定单元的首字即可将 I/O 区扩展到包含 CIO 0160 ~ CIO 0999。首字可利用 CX-Programmer 在 I/O 表中进行设定，其设定范围为 CIO 0 ~ CIO 900。

*2 若 I/O 存储器保持标志 (A500.12) 为 ON，则在运行模式发生变更时，将保持存储器中的值。此外，若在 PLC 设置中进行设定以在启动时保持 I/O 存储器保持标志 (即 IOM 保持位参数) 的状态，则在接通电源时，将保持存储器中的值。

*3 仅 CJ2H CPU 单元支持该区。“同步单元”是指支持单元同步运转功能的 CPU 总线单元和高功能 I/O 单元。

*4 仅 CJ2M CPU 单元支持该区。

*5 H512 ~ H515 可设定为仅用作功能块存储器或 SFC 存储器。

*6 为实现辅助区的扩展，CJ2 CPU 单元新增了 A960 ~ A1471 和 A10000 ~ A11535。这些字无法被不支持 CJ2 CPU 单元的 CPU 总线单元、高性能 I/O 单元、PT 以及支持软件访问。
仅下述 CPU 总线单元和高功能 I/O 单元支持 CJ2 CPU 单元。

- EtherNet/IP 单元: CJ1W-EIP21
- 位置控制单元: CJ1W-NC214、CJ1W-NC234、CJ1W-NC281、CJ1W-NC414、CJ1W-NC434、CJ1W-NC481 和 CJ1W-NC881
- 模拟量输入单元: CJ1W-AD042
- 模拟量输出单元: CJ1W-DA042V
- 串行通信单元: CJ1W-SCU22、CJ1W-SCU32 和 CJ1W-SCU42

区	大小	范围	外部 I/O 分配	按位访问	按字访问	访问		通过编程设备进行修改	启动时或模式变更时的状态	强制位状态	参考
						读	写				
TR 区	16 位	TR0 ~ TR15	---	允许	---	允许	允许	不允许	清除	不允许	6-22
DM 区	32,768 字	D00000 ~ D32767	---	允许*8	允许	允许	允许	允许	保持	不允许	6-23
EM 区	32,768字/存储区块, 最多 25 个存储区块 (0 ~ 18 Hex)	E00_0 ~ E18_32767*8	---	允许*8	允许	允许	允许	允许	保持	可通过设定进行启用。*9	6-26
定时器完成标志	4,096 位	T0 ~ T4095	---	允许	---	允许	允许	允许	清除	允许	6-30
计数器完成标志	4,096 位	C0 ~ C4095	---	允许	---	允许	允许	允许	保持	允许	6-32
定时器当前值 (PV)	4,096 字	T0 ~ T4095	---	---	允许	允许	允许	允许	清除	不允许*10	6-30
计数器当前值 (PV)	4,096 字	C0 ~ C4095	---	---	允许	允许	允许	允许	保持	不允许*11	6-32
任务标志区	128 位	TK000 ~ TK127	---	允许	---	允许	不允许	不允许	清除	不允许	6-33
变址寄存器 *7	16 个寄存器	IR0 ~ IR15	---	允许	允许	仅限间接寻址	仅限特定指令	不允许	清除	不允许	6-34
数据寄存器 *7	16 个寄存器	DR0 ~ DR15	---	不允许	允许	允许	允许	不允许	清除	不允许	6-39
条件标志	例如: 常 ON 标志	CX-Programmer 全局符号表中的系统符号 (例如: P_On)	---	允许	---	允许	不允许	不允许	清除	不允许	6-41
脉冲位	例如: 1s 时钟脉冲	CX-Programmer 全局符号表中的系统符号 (例如: P_1s)	---	允许	---	允许	不允许	不允许	清除	不允许	6-43

*7 变址寄存器和数据寄存器既可供各个任务单独使用, 又可供所有任务共享。

*8 为实现 EM 区的扩展, CJ2 CPU 单元的 EM 区新增了存储区块 D ~ 18 Hex。此外, CJ2 CPU 单元还增加了一项新特性—对 DM 和 EM 区中的位进行寻址的能力。
EM 区的存储区块 D ~ 18 Hex 和 DM/EM 区中的位地址无法被不支持 CJ2 CPU 单元的 CPU 总线单元、高性能 I/O 单元、PT 以及支持软件访问和使用。
仅下述 CPU 总线单元和高性能 I/O 单元支持 CJ2 CPU 单元。

- EtherNet/IP 单元: CJ1W-EIP21
- 位置控制单元: CJ1W-NC214、CJ1W-NC234、CJ1W-NC281、CJ1W-NC414、CJ1W-NC434、CJ1W-NC481 和 CJ1W-NC881
- 模拟量输入单元: CJ1W-AD042
- 模拟量输出单元: CJ1W-DA042V
- 串行通信单元: CJ1W-SCU22、CJ1W-SCU32 和 CJ1W-SCU42

*9 可对特定存储区块中的位以及其后的所有存储区块进行强制置位 / 复位。(此即为 EM 区的“强制置位 / 复位功能”。) 对 CJ2H CPU 单元而言, 可对 EM 区中设定为“自动地址分配”的后续存储区块中的位进行强制置位 / 复位。

- CJ2H-CPU64/65(-EIP): E03_0 ~ E03_32767
- CJ2H-CPU65(-EIP): E06_0 ~ E09_32767
- CJ2H-CPU67(-EIP): E07_0 ~ E0E_32767
- CJ2H-CPU68(-EIP): E11_0 ~ E18_32767

*10 可通过对定时器完成标志进行强制置位 / 复位实现对定时器当前值 (PV) 的间接刷新。

*11 可通过对计数器完成标志进行强制置位 / 复位实现对计数器当前值 (PV) 的间接刷新。

6-1-3 保持 I/O 存储器值

CPU 单元运行状态变化时清除 I/O 存储器

当 CPU 单元运行状态发生变化时，将清除 I/O 存储器值（某些 I/O 存储器区除外）。

● 导致 I/O 存储器被清除的各种变化

- PROGRAM 和 RUN 或 MONITOR 运行模式之间的切换
- 因致命错误而导致的运行停止（因执行 FALS(007) 指令导致的运行停止的情况除外）
- 电源复位（即断电后再上电）

● 保持 I/O 存储器的各种设定

可设定为即使在 CPU 单元的运行状态发生变化时亦可保持 I/O 存储器。

- 在运行模式变化和产生致命错误时保持 I/O 存储器：
将辅助区中的 IOM 保持位 (A500.12) 置 ON。
- 在电源复位时保持 I/O 存储器：
将辅助区中的 IOM 保持位 (A500.12) 置 ON，并在 PLC 设置的“Startup”（启动）选项页中的“Startup Hold Area”（启动保持区）中勾选“IOM Hold Bit”（IOM 保持位）复选框。



安全使用注意事项

进行设定以保持 I/O 区时

当 CPU 单元的运行模式从 RUN 或 MONITOR 切换到 PROGRAM 时，将不清除 I/O 存储器中的输出位（即不会被置 OFF），而是保持切换至 PROGRAM 模式前的状态。然后，当运行模式从 PROGRAM 切换到 RUN 或 MONITOR 时，将输出先前 I/O 存储器的值。

当因致命错误（含 FALS(007) 指令的执行）而导致运行停止时，CPU 单元的 I/O 存储器中的值将被保持，而输出单元中的所有输出将被置 OFF。

● I/O 存储区中对应的 I/O 存储器保持状态

下表所示为各个 I/O 存储区中的保持状态。

区		CPU 单元运行模式切换时	发生致命错误时		电源接通时
			执行 FALS(007)指令时	发生其它致命错误时	
CIO 区 (核心 I/O 区)	I/O 区	可能会保持 *2 (取决于设定)	保持	可能会保持 *2 (取决于设定)	可能会保持 *3 (取决于设定)
	数据链接区				
	同步数据刷新区 *1				
	CPU 总线单元区				
	高功能 I/O 单元区				
	脉冲 I/O 区 *4				
	串行 PLC 链接区 *4				
	DeviceNet 区				
内部 I/O 区					
工作区 (W)		可能会保持 *2 (取决于设定)	保持	可能会保持 *2 (取决于设定)	可能会保持 *3 (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)			
数据存储区 (D)		保持			
扩展数据存储区 (E)		保持			
定时器完成标志 (T)		可能会保持 *2 (取决于设定)	保持	可能会保持 *2 (取决于设定)	可能会保持 *3 (取决于设定)
定时器当前值 (PV)(T)		可能会保持 *2 (取决于设定)	保持	可能会保持 *2 (取决于设定)	可能会保持 *3 (取决于设定)
计数器完成标志 (C)		保持			
计数器当前值 (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 仅 CJ2H CPU 单元支持该区。

*2 对辅助区中的 IOM 保持位 (A500.12) 置 ON 以保持这些区。

*3 对辅助区中的 IOM 保持位 (A500.12) 置 ON, 并在 PLC 设置的 “Startup” (启动) 选项页中的 “Startup Hold Area” (启动保持区) 中勾选 “IOM Hold Bit” (IOM 保持位) 复选框。

*4 仅 CJ2M CPU 单元支持该区。

6-2 I/O 区

I/O 区的地址范围为 CIO 0 ~ CIO 159(字地址) 和 CIO 0.00 ~ CIO 159.15(位地址)。

I/O 区中的字分配给基本 I/O 单元的 I/O 端子。分配给基本 I/O 单元的字是根据插槽位置(从左至右)和所需的字数进行连续分配的,在遇到空槽时会自行跳过。I/O 区中未分配给基本 I/O 单元的字仅可在程序中使用。

可对 I/O 区中的位进行强制置位 / 复位。

注 只需通过修改分配给特定单元的首字即可将该区扩展到 CIO 0000 ~ CIO 0999。即使在 I/O 扩展后,可分配给外部 I/O 的最大位数仍然为 2,560(160 字)。

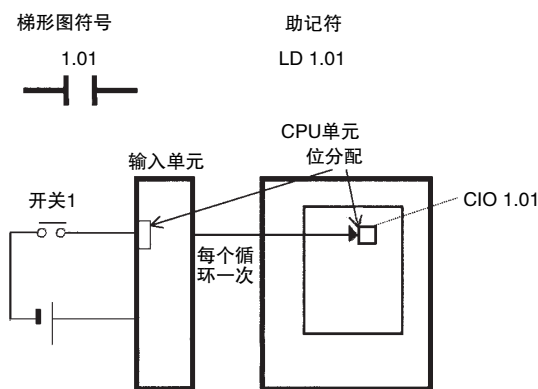
6-2-1 输入位

当将 I/O 区中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。输入位反映了各种设备(如按钮开关、限位开关和光电开关等)的 ON/OFF 状态。

PLC 中有三种输入点状态刷新方法: 正常 I/O 刷新、即时刷新和 IORF(097)刷新。

正常 I/O 刷新

在程序执行完成后,将在每个循环中读入一次外部设备的输入点状态。下例中, CIO 1.01 被分配给开关 1(即连接至输入单元输入端子上的一个外部开关);且在每个循环中,开关 1 的 ON/OFF 状态均被存储到 CIO 1.01 中。



即时刷新

当通过在指令前输入一个感叹号而指定指令的即时刷新变化，且指令的操作数为一个输入位或字时，刷新的执行如下所示。该即时刷新是在每循环一次的正常 I/O 刷新之外执行的刷新。

- 位操作数

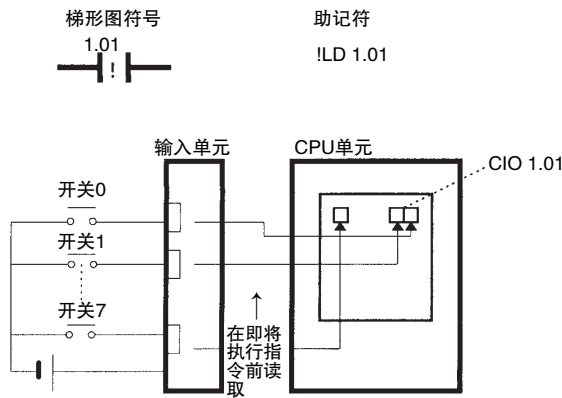
在即将执行指令前，分配给包含指定位的字的 16 点 I/O 的 ON/OFF 状态将被读入 PLC 中。

- 字操作数

在即将执行指令前，分配给指定字的 16 点 I/O 的 ON/OFF 状态将被读入 PLC 中。

- 示例

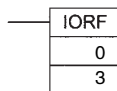
下例中，CIO 1.01 被分配给开关 1 (即连接至输入单元输入端子上面的一个外部开关)；且在即将执行 !LD 1.01 指令前，开关 1 的 ON/OFF 状态将被读入并反映在 CIO 1.01 中。



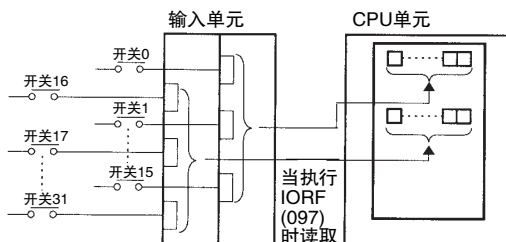
IORF(097) 刷新

执行 IORF(097)(I/O 刷新) 指令时，将对指定范围内的输入位进行刷新。该 I/O 刷新是在每循环一次的正常 I/O 刷新之外执行的刷新。

如下所示，IORF(097) 指令将对 I/O 区字 CIO 0 ~ CIO 3 中的所有 I/O 点的状态执行刷新；并从输入单元中读取输入点的状态以及将输出位的状态写入到输出单元中。

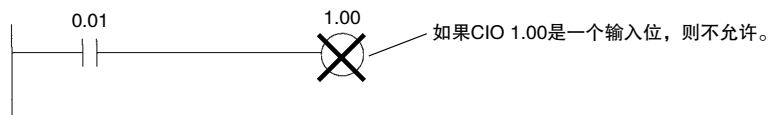


下例中，将从输入单元中读取分配给 CIO 0 和 CIO 1 的输入点的状态。(CIO 2 和 CIO 3 被分配给输出单元。)



输入位的限制

输入位在程序中用作常开和常闭条件的次数没有限制。地址可按任意顺序编程。
输入位不可用作输出指令的操作数。



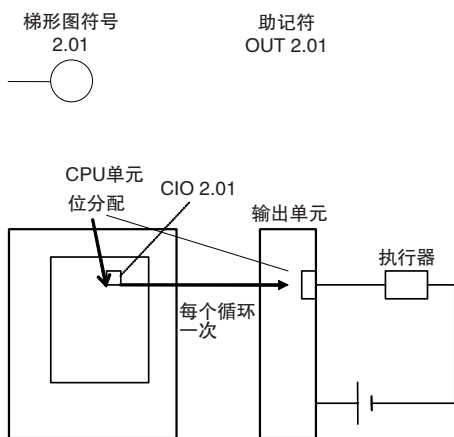
6-2-2 输出位

当将 I/O 区中的位分配给输出单元时, I/O 区中的一个位称为一个输出位。输出位的 ON/OFF 状态会被输出到执行器等设备中。输出单元有三种输出位状态刷新方法: 正常 I/O 刷新、即时刷新和 IORF(097) 刷新。

正常 I/O 刷新

在程序执行完后, 输出位的状态在每个循环中均将被输出至外部设备中。

下例中, CIO 2.01 被分配给执行器 (即连接到输出单元输出端子上的外部设备), 且 CIO 2.01 的 ON/OFF 状态在每个循环中均被输出至该执行器中。



即时刷新

当通过在指令前输入一个感叹号而指定指令的即时刷新变化, 且指令的操作数为一个输出位或字时, 刷新的执行如下所示。该即时刷新是在每循环一次的正常 I/O 刷新之外执行的刷新。

● 位操作数

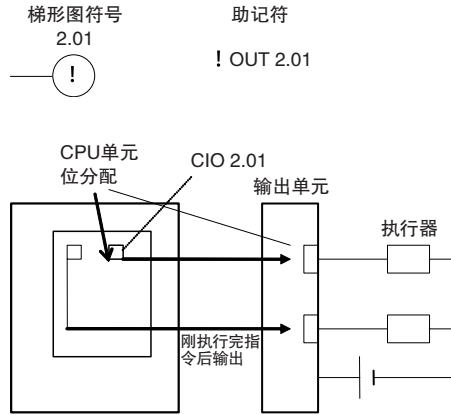
在刚执行完指令后, 分配给包含指定位的字的 16 点 I/O 的 ON/OFF 状态将被输出至输出设备中。

● 字操作数

在刚执行完指令后, 分配给指定字的 16 点 I/O 的 ON/OFF 状态将被输出至输出设备中。

● 示例

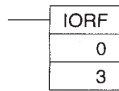
下例中，CIO 2.01 被分配给执行器（即连接到输出单元输出端子上的外部设备），且 CIO 2.01 的 ON/OFF 状态会在刚执行完 !OUT 2.01 指令后被输出至执行器中。



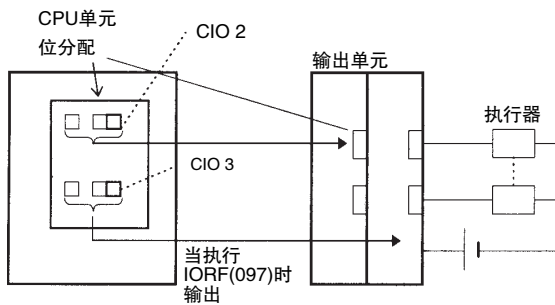
IORF(097) 刷新

执行 IORF(097)(I/O 刷新) 指令时，指定范围内字的输出位的 ON/OFF 状态将被输出至其外部设备中。该 I/O 刷新是在每循环一次的正常 I/O 刷新之外执行的刷新。

如下所示，IORF(097) 指令将对 I/O 区字 CIO 0 ~ CIO 3 中的所有 I/O 点的状态执行刷新；并从输入单元中读取输入点的状态以及将输出位的状态写入到输出单元中。



下例中，分配给 CIO 2 和 CIO 3 的输入点的状态被输出至输出单元中。(CIO 0 和 CIO 1 被分配给输入单元。)

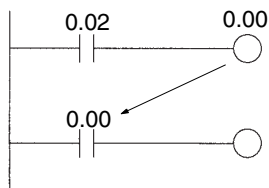


正确使用注意事项

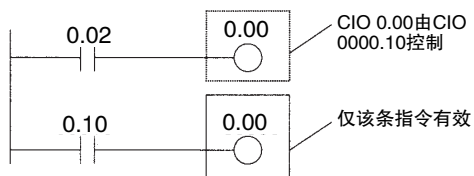
只需对输出 OFF 位 (A500.15) 置 ON，即可实现对所有基本 I/O 单元和高功能 I/O 单元的输出置 OFF。即使实际输出均被置 OFF，输出位的状态也不会受到影响。

输出位的限制

输出位可按任意顺序编程，且可用作输入指令的操作数。输出位在程序中用作常开和常闭条件的次数没有限制。



输出位仅可在控制其状态的单条指令中使用。若输出位的状态受两条或两条以上的指令控制，则只有最后一条指令才有效。



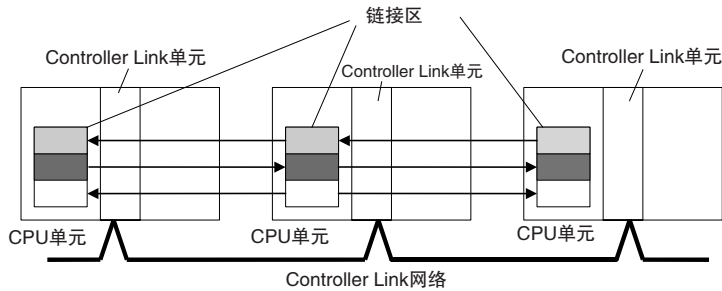
6-3 数据链接区

数据链接区的地址范围为 CIO 1000 ~ CIO 1199(字地址) 和 CIO 1000.00 ~ CIO 1199.15(位地址)。当 LR 被设定为 Controller Link 网络的数据链接区时，数据链接区中的字用于实现数据链接。

数据链接通过连接至 PLC 的 Controller Link 单元，自动地（不依靠程序）与网络中其它 CPU 单元数据链接区中的数据实现共享。

数据链接可以自动（各个结点均采用相同数目的字）或手动生成。当用户手动定义数据链接时，可为每个结点分配任意数目的字且可将各结点属性定义为“只接收”或“只发送”。详情请参考《Controller Link 单元操作手册》（样本编号：W309）。

当 LR 未被设定为数据链接区（在未使用 Controller Link 网络的情况下）时，可在程序中使用数据链接区中的字。



可对数据链接区中的位进行强制置位 / 复位。

6-4 同步数据刷新区

同步数据刷新区的地址范围为 CIO 1200 ~ CIO 1295(字地址) 和 CIO 1200.00 ~ CIO 1295.15(位地址)。

在 CJ2H CPU 单元采用同步单元运转功能时，同步数据刷新区用于实现 CPU 单元和“同步单元”之间的数据交换。仅 CJ2H CPU 单元支持该区。详情请参考“10-8-4 同步数据刷新”。

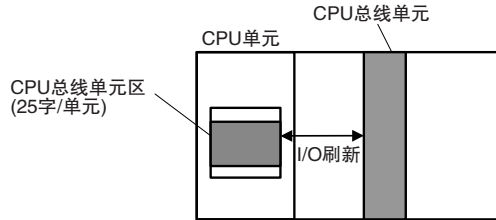
6-5 CPU 总线单元区

CPU 总线电源的地址范围为 CIO 1500 ~ CIO 1899(字地址) 和 CIO 1500.00 ~ CIO 1899.15(位地址)。

CPU 总线单元中的字分配给 CPU 总线单元, 用于传送数据 (如单元的运行状态等)。每个单元均会按照其单元号设定分配 25 个字。

与 CPU 总线单元进行的数据交换会在下述情况下进行:

- I/O 刷新时段
- 执行 DLNK(226) 指令时



每个 CPU 总线单元均会按照其单元号分配 25 个字, 具体如下表所示:

单元号	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

上述 25 个字的功能取决于正在使用的 CPU 总线单元。详情请参考相关单元操作手册。

CPU 总线单元区中未分配给 CPU 总线单元的字仅在程序中使用。

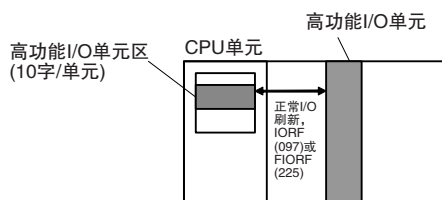
6-6 高性能 I/O 单元区

高性能 I/O 单元区的地址范围为 CIO 2000 ~ CIO 2959(字地址) 和 CIO 2000.00 ~ CIO 2959.15(位地址)。

高性能 I/O 单元区中的字分配给高性能 I/O 单元, 用于传送数据 (如各单元的运行状态等)。每个单元均会按照其单元号设定分配 10 个字。最多可供使用的单元为 96 个, 其编号为 0 ~ 95。

高性能 I/O 单元区中的字可在下述情况下进行刷新:

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



每个高性能 I/O 单元均会按照其单元号分配 10 个字, 具体如下表所示:

单元号	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

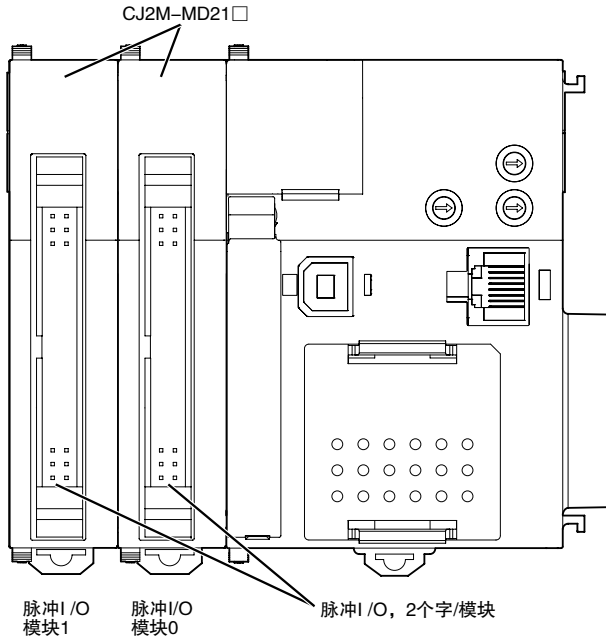
分配给各单元的 10 个字的功能取决于所使用的高性能 I/O 单元。详情请参考相关单元操作手册。

高性能 I/O 单元区中未分配给高性能 I/O 单元的字仅在程序中使用。

6-7 脉冲 I/O 区

脉冲 I/O 区的地址范围为 CIO 2960 ~ CIO 2963(字地址) 和 CIO 2960.00 ~ CIO 2963.15(位地址)。

在脉冲 I/O 模块连接至 CJ2M CPU 单元的情况下，脉冲 I/O 区中的字分配给各项脉冲 I/O 功能。每个脉冲 I/O 模块均会按照其模块号分配 2 个字，总共可分配 4 个字 (脉冲 I/O 模块 0 和脉冲 I/O 模块 1)。



具体分配如下所示：

字地址	模块号
CIO 2960 和 CIO 2961	0
CIO 2962 和 CIO 2963	1

有关脉冲 I/O 模块分配字详情，请参考《CJ2M CPU 单元脉冲 I/O 模块操作手册》(样本编号：W486)。

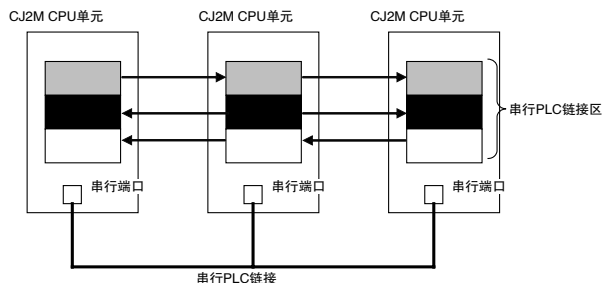
6-8 串行 PLC 链接区

串行 PLC 链接区的地址范围为 CIO 3100 ~ CIO 3189(字地址) 和 CIO 3100.00 ~ CIO 3189.15(位地址)。

串行 PLC 链接区供串行 PLC 链接使用, 可用于同其它 PLC 的数据链接。对于串行 PLC 链接而言, 只需使用串行端口而无需进行通信编程即可实现数据交换。

串行 PLC 链接区中的字会根据主站 PLC 的 PLC 设置中的设定进行自动分配。

- 串行 PLC 链接模式
- 串行 PLC 链接发送字数
- 串行 PLC 链接最大单元数



串行 PLC 链接区中未分配给串行 PLC 链接的字仅可在程序中作为工作位使用。可对串行 PLC 链接区中的位进行强制置位 / 复位。

在下述情况下, 该工作区中的内容将被清除:

- 运行模式从 PROGRAM 切换至 RUN/MONITOR 或反之之时
- PLC 电源复位时
- 通过 CX-Programmer 清除该区时
- 当因致命错误而导致运行停止时 (在因执行 FALS(007) 指令而导致运行停止时不清除该区的情况除外)

● 将 IOM 保持位 (A500.12) 设定为 ON

如果 IOM 保持位 (A500.12) 为 ON, 则在发生致命错误或运行模式从 PROGRAM 切换至 RUN/MONITOR 时或反之之时, 将保持该区中的内容。但是, 在电源重启时, 该区中的内容将被清除。

● 将 IOM 保持位 (A500.12) 设定为 ON 并对 PLC 设置中的 IOM 保持位自身进行保护

即使在电源重启时, 仍将保持该区中的内容。

6-9 DeviceNet 区

DeviceNet 区的地址范围为 CIO 3200 ~ CIO 3799(字地址) 和 CIO 3200.00 ~ CIO 3799.15(位地址)。

DeviceNet 区中的字分配给从站，用于实现 DeviceNet 远程 I/O 通信。数据通过 DeviceNet 单元实现与网络从站之间的定期交换 (不依靠程序)。

采用固定分配方式时，将按照固定分配设定 1、2、3 把字分配给各个从站。可选择固定区中的任何一个区。

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 I/O 从站功能时，将把下述字分配给 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 I/O 分配方法：固定分配 (根据结点地址) 和用户自定义分配。

- 采用固定分配方式时，将按照结点地址把字自动分配给指定的固定分配区 (1 ~ 3) 中的从站。
- 采用用户自定义分配方式时，用户可将下述字分配给各个从站。

CIO 0 ~ CIO 6143

W0 ~ W511

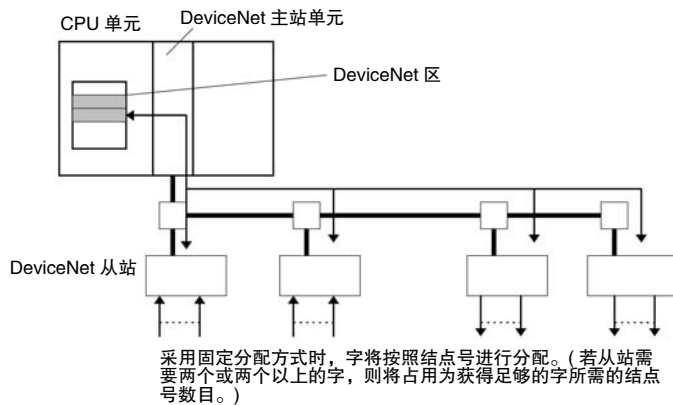
H0 ~ H511

D0 ~ D32767

E00_0 ~ E0C_32767, 存储区块: 0 ~ 18 Hex

(无法指定高于 C 的 EM 存储区块)

有关字分配详情，请参考《DeviceNet 操作手册》(样本编号: W267)。



6-10 工作区

工作区中包含 512 个字，其地址范围为 W0 ~ W511(字地址) 和 W0.00 ~ W511.15(位地址)。该区中的字不可用于带外部 I/O 端子的 I/O 单元，而仅可在程序内部使用。

在 CIO 区(CIO 1300 ~ CIO 1499 和 CIO 3800 ~ CIO 6143) 的内部 I/O 区中存在的未使用字亦可在程序中使用。但是，在之后版本的 CPU 中可能会为 CIO 区中未使用的字分配新的功能。因此，在使用内部 I/O 区中的字前，请务必首先使用工作区中的可用字。

6-11 保持区

保持区的地址范围为 H000 ~ H511(字地址) 和 H000.00 ~ H511.15(位地址)。这些字只能在程序中使用。当电源关闭或当运行模式从 PROGRAM 切换为 RUN/MONITOR 时或反之, 将保持该区中的位状态。

保持区位可在程序中按任意顺序使用, 且可根据需要随时用作常开或常闭条件。

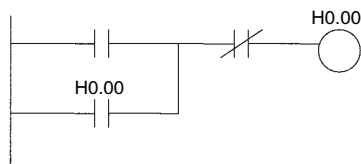
● 保持区初始化

当 PLC 电源重启时或当 CPU 单元运行模式从 PROGRAM 切换为 RUN/MONITOR 时或反之, 将不清除保持区中的数据。

当保持位编入 IL(002) 和 ILC(003) 指令之间且 IL(002) 的执行条件为 OFF 时, 保持位将被清除。若要使某个位即使在 IL(002) 指令的执行条件为 OFF 时也保持 ON, 则需在紧接着 IL(002) 指令前使用置位指令对该位置 ON。

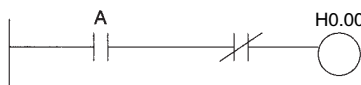
● 自保持位

当用保持区位对一个自保持位进行编程时, 即使电源已复位, 自保持位也不会被清除。



如果未使用保持区位作为自保持位, 则在电源复位时, 该位将变为 OFF 状态且自保持位将被清除。

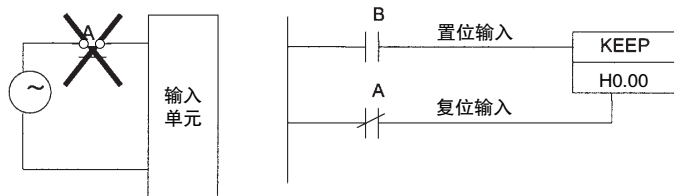
如下图所示, 如果使用一个保持区位, 但并未将其作为自保持位进行编程, 则在电源复位时, 该位将因执行条件 A 而变为 OFF 状态。



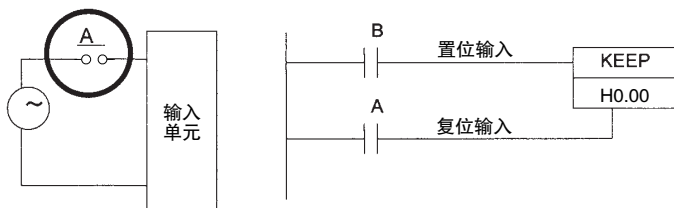
保持区的字 H512 ~ H5135 可被设定用于功能块或 SFC 程序中, 但不可指定为用户程序中的指令操作数。

● 注意事项

当在 KEEP(011) 指令中使用一个保持位时, 如果输入设备采用交流电源, 则切勿对复位输入使用常闭条件。当电源断开或暂时中断, 输入将先于 PLC 内部电源进入 OFF 状态, 而保持区位则被复位。



为此, 应采用如下所示的配置。



位地址的使用顺序或常开或常闭条件的编程次数没有限制。

6-12 辅助区

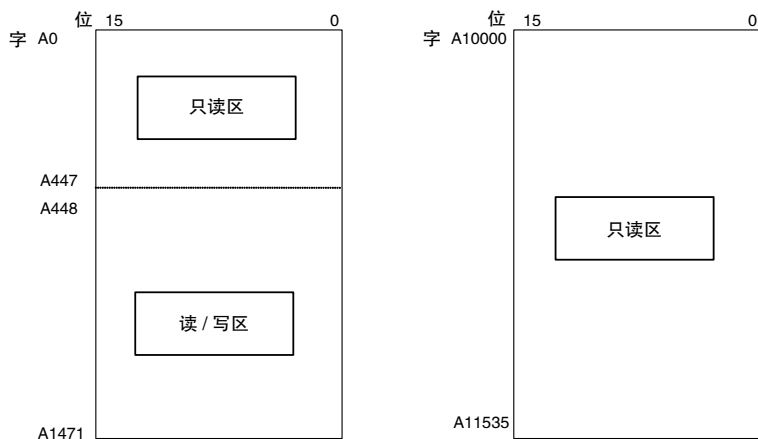
辅助区的地址范围为 A0 ~ A1471 和 A1000 ~ A11535(字地址) 以及 A0.00 ~ A1471.15 和 A1000.00 ~ A11535.15(位地址)。这些字被预先分配给标志和控制位，用于监视和控制运行状况。

部分字和位由系统自动设定，其余则由用户自行设定和操作。辅助区包含由自诊断、初始设定、控制位和状态数据设定的出错标志。(有关辅助区位和字的功能详情，请参考“**A-3 辅助区**”。)

A0 ~ A447 和 A10000 ~ A11535 为只读状态，而 A448 ~ A1471 则可通过程序或 CX-Programmer 进行读写。为实现辅助区的扩展，CJ2 CPU 单元新增了 A960 ~ A1471 和 A10000 ~ A11535。这些字无法被不支持 CJ2 CPU 单元的 CPU 总线单元、高功能 I/O 单元、PT 以及支持软件访问。

仅下述 CPU 总线单元和高功能 I/O 单元支持 CJ2 CPU 单元。

- EtherNet/IP 单元：CJ1W-EIP21
- 位置控制单元：CJ1W-NC214、CJ1W-NC234、CJ1W-NC281、CJ1W-NC414、CJ1W-NC434、CJ1W-NC481 和 CJ1W-NC881
- 模拟量输入单元：CJ1W-AD042
- 模拟量输出单元：CJ1W-DA042V
- 串行通信单元：CJ1W-SCU22、CJ1W-SCU32 和 CJ1W-SCU42



通过 CX-Programmer 执行下述操作便可向辅助区中写入数据：在监控编程地址（即“设定值”对话框）的同时改变当前值或在编辑 PLC 数据表后向 PLC 传送数据。此外，联机置位 / 复位操作无法用于辅助区中的位。请参考《CX-Programmer 操作手册》（样本编号：W414）。



正确使用注意事项

在今后对 CPU 单元升级时，可能会将功能分配给未定义的辅助区中的字或位。请勿在用户程序中将未定义的辅助区中的字或位作为工作字或工作位使用。

6-13 暂存继电器区

TR 区包含 16 个位，其地址范围为 TR0 ~ TR15。TR 位可用于存在多个输出分支且无法使用互锁的情况。

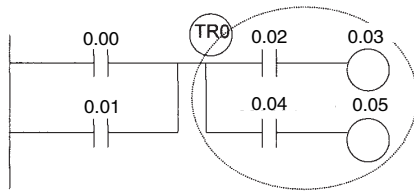
只要未在同一指令块中两次使用同一个 TR 位，则可按所需次数、以任意顺序使用 TR 位。

TR 位仅可与 OUT 和 LD 指令一起使用。其中，OUT 指令 (OUT TR0 ~ OUT TR15) 用于存储分支点的 ON/OFF 状态，LD 指令用于调出已存储的分支点的 ON/OFF 状态。

TR 位的状态无法通过 CX-Programmer 进行修改。

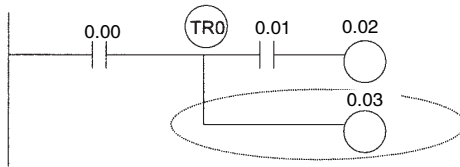
● 应用示例

如下例所示，当两个输出直接连接至一个分支点时，需要使用一个 TR 位。



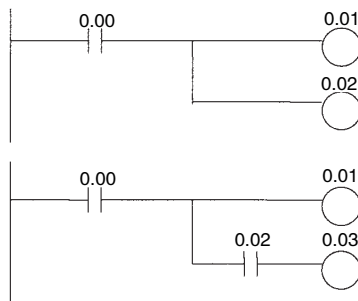
指令	操作数
LD	0.00
OR	0.01
OUT	TR 0
AND	0.02
OUT	0.03
LD	TR 0
AND	0.04
OUT	0.05

如下例所示，当一个无独立执行条件的输出连接至一个分支点时，需要使用一个 TR 位。



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

当在分支点后没有执行条件或仅在指令块的最后一行存在执行条件，无需使用 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

6-14 数据存储区

DM 区的地址范围为 D0 ~ D32767(字地址)。该数据区用于常规数据存储和处理,可按字或位进行访问。当 PLC 电源重启时或当 CPU 单元运行模式从 PROGRAM 切换为 RUN/MONITOR 时或反之,DM 区中的数据将保持不变。

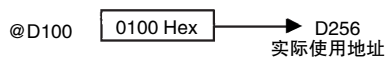
DM 区中的位无法被强制置位 / 复位。

间接寻址

DM 区中字的间接寻址模式有两种:二进制模式和 BCD 模式。

● 二进制模式寻址 (@D)

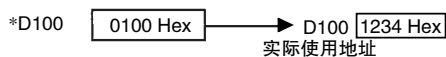
当在一个 DM 地址前添加一个 “@” 符号时,该 DM 字的内容将被作为二进制处理,且指令将对该二进制地址处的 DM 字执行相关操作。整个 DM 区 (D0 ~ D32767) 均可利用十六进制值 0 ~ 7FFF 进行间接寻址。



● BCD 模式寻址 (*D)

当在一个 DM 地址前添加一个 “*” 符号时,该 DM 字的内容将被作为 BCD 处理,且指令将对该 BCD 地址处的 DM 字执行相关操作。只有部分 DM 区 (D0 ~ D09999) 可利用 BCD 值 0 ~ 9999 进行间接寻址。

例如: [MOV #1234 *D100]



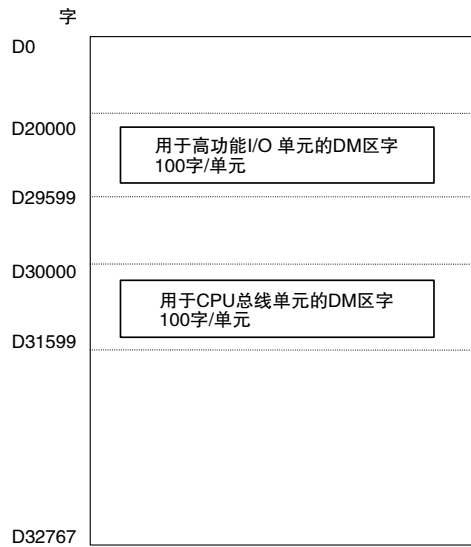
分配给高性能 I/O 单元和 CPU 总线单元的 DM 区

部分 DM 区会分配给高性能 I/O 单元和 CPU 总线单元，用于实现单元初始化设定等功能。如果在 PLC 中未使用相应的单元，这些字可用于常规数据存储。

这些单元何时进行数据传送不尽相同，下述三种情况之一时会进行传送：

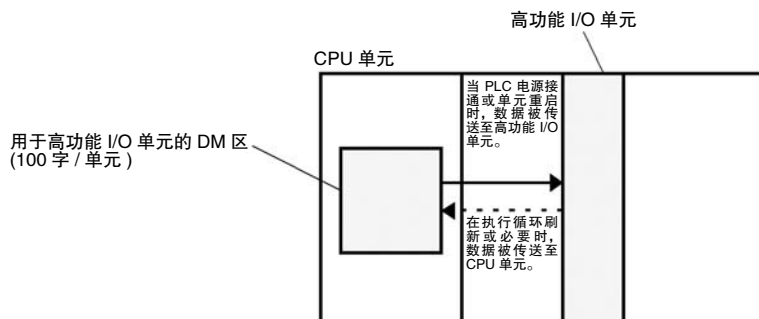
- 在 PLC 电源接通或单元重启时
- 每个循环传送一次
- 请求时

有关何时进行数据传送的详情，请参考相关单元操作手册。



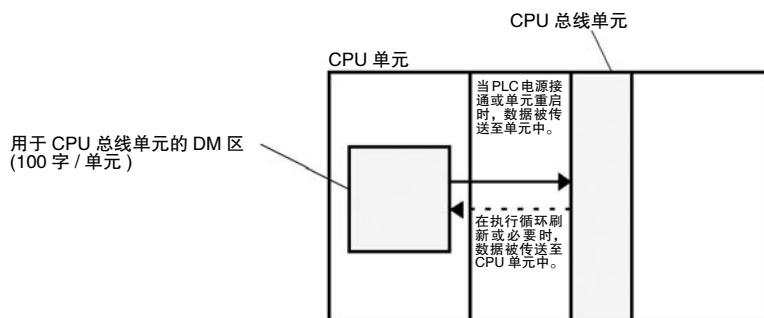
● 高性能 I/O 单元 (D20000 ~ D29599)

每个高性能 I/O 单元分配 100 字 (依据单元号 0 ~ 95)。有关这些字的功能详情, 请参考相关单元操作手册。



● CPU 总线单元 (D30000 ~ D31599)

每个 CPU 总线单元分配 100 字 (依据单元号 0 ~ F)。有关这些字的功能详情, 请参考相关单元操作手册。对于某些 CPU 总线单元 (如 Ethernet 单元) 而言, 还须通过编程设备将初始化设定 (数据) 注册到 CPU 单元的参数区中。



6-15 扩展数据存储区

EM 区的地址范围为 E0_0 ~ E18_32767(字地址)。EM 区划分为 24 个存储区块 (0 ~ 18 Hex)。其中，地址中“E”之后的十六进制数表示“存储区块”。

EM 区用于常规数据存储和操作，可按字或位进行访问。

当 PLC 电源重启时或当 CPU 单元运行模式从 PROGRAM 切换为 RUN/MONITOR 时或反之，EM 区中的数据将保持不变。

除了作为常规数据区外，EM 区还可通过设定用于自动地址分配、跟踪存储器和文件存储器。

为实现 EM 区的扩展，CJ2 CPU 单元的 EM 区新增了存储区块 D ~ 18 Hex(E0D_0 ~ E18_32767)。这些存储区块无法被不支持 CJ2 CPU 单元的 CPU 总线单元、高性能 I/O 单元、PT 以及支持软件访问。仅下述 CPU 总线单元和高性能 I/O 单元支持 CJ2 CPU 单元。

- EtherNet/IP 单元：CJ1W-EIP21
- 位置控制单元：CJ1W-NC214、CJ1W-NC234、CJ1W-NC281、CJ1W-NC414、CJ1W-NC434、CJ1W-NC481 和 CJ1W-NC881
- 模拟量输入单元：CJ1W-AD042
- 模拟量输出单元：CJ1W-DA042V
- 串行通信单元：CJ1W-SCU22、CJ1W-SCU32 和 CJ1W-SCU42

自动地址分配

我们建议使用 EM 区为符号自动分配地址。对 CJ2H CPU 单元而言，在采用自动地址分配功能时，可对下述 EM 存储区块中的位进行强制置位 / 复位。

型号	使用自动地址分配时，可对其中的位进行强制置位 / 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

有关自动地址分配的执行步骤，请参考“5-5-8 将地址自动分配到符号”。



附加信息

1.2 版或更高版本的 CPU 单元均可对所有 EM 区存储区块中的位执行强制置位 / 复位。

对于 CJ2M CPU 单元而言，仅可在不使用自动地址分配的情况下，对使用 EM 区强制置位 / 复位功能指定的存储区块实现 EM 区相关位的强制置位 / 复位。

型号	允许强制置位 / 复位的存储区块
CJ2M-CPU□1	E0
CJ2M-CPU□2	
CJ2M-CPU□3	
CJ2M-CPU□4	E0 ~ E3
CJ2M-CPU□5	

EM 区强制置位 / 复位功能

可使用 CX-Programmer 进行参数设定，从而启用对指定 EM 区存储区块以及所有后续 EM 区存储区块中的位的强制置位 / 复位。（默认设定下禁止该功能。）

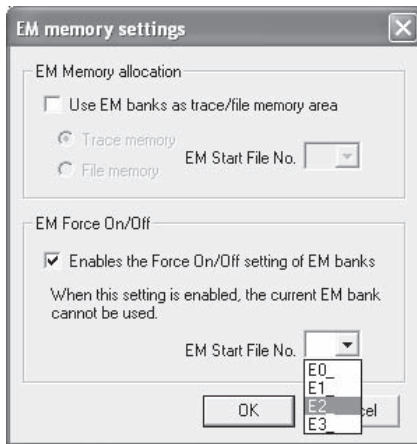
即使在未使用自动地址分配区时，该功能仍可启用对 EM 区中位的强制置位 / 复位。该功能可对所有的 EM 区存储区块进行设定。

● 设定步骤

1. 通过 CX-Programmer 选择 “PLC – Memory Allocate – EM Memory Settings” (PLC- 存储器分配 – EM 存储器设定)。^{*}

此时将显示 EM 存储器设定对话框。

2. 勾选相关复选框，启用 EM 区强制置位 / 复位功能，并设定首个 EM 区存储区块。



3. 将 CPU 单元进行联机，然后传送用户程序。

^{*} 使用 EM 区强制置位 / 复位功能时，需使用 8.3 版或更高版本的 CX-Programmer。



正确使用注意事项

- 在启用 EM 区强制置位 / 复位功能时，指定为 EM 区强制置位 / 复位功能的存储区块不可作为当前 EM 区存储区块使用。
- EM 区强制置位 / 复位功能不可用于当前正用作跟踪存储器或文件存储器的 EM 区存储区块。
- 强制置位 / 复位可在 MONITOR 或 PROGRAM 模式下执行，而不可在 RUN 模式下执行。有关对位进行强制置位 / 复位的操作步骤详情，请参考 “10-7-1 强制置位 / 复位”。

文件存储器

文件存储器用于存储 CPU 单元使用的各种文件。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采样数据。详情请参考 “10-7-6 数据跟踪”。

直接指定 EM 地址

指定 EM 地址有两种方法：同时指定存储区块和地址或指定当前存储区块地址。一般情况下，我们建议同时指定存储区块和地址。

● 指定存储区块和地址

采用该方法时，需紧接着地址前指定存储区块号。例如，E2_10 指定存储区块 2 中的地址 E10。

● 指定当前存储区块地址

采用该方法时，仅指定地址。例如，E10 指定当前存储区块中的地址 E10。

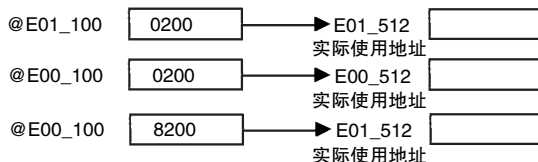
- 除非 IOM 保持位 (A500.12) 为 ON，否则在运行模式从 PROGRAM 切换至 RUN/MONITOR 时，当前存储区块将被复位为零。
- 可通过 EMBC(281) 指令修改当前存储区块，以访问其它存储区块中的数据。
- A301 中含有当前 EM 存储区块号。
- 随着程序不断执行循环任务，当前存储区块保持不变。例如，若当前 EM 存储区块在任务 1 中变为存储区块 2，那么在任务 2 中其仍将为存储区块 2。若当前存储区块已在中断任务中发生变化，则其将返回其原始值 (即在源循环任务中的值)。

间接指定 EM 地址

EM 区中字的间接寻址模式有两种：二进制模式和 BCD 模式。

● 二进制模式寻址 (@E)

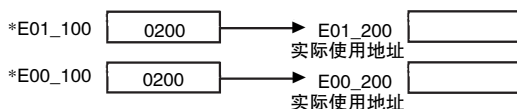
当在一个 EM 地址前添加一个“@”符号时，该 EM 字的内容将被作为二进制值处理，以表示另一个 EM 区地址。而指令将对同一个存储区块或下一个存储区块中该二进制地址处的 EM 字执行操作。同一个 EM 存储区块 (E0 ~ E32767) 中的所有字均可使用十六进制值 0 ~ 7FFF 进行间接寻址，而下一个 EM 存储区块 (E0 ~ E32767) 中的字则可使用十六进制值 8000 ~ FFFF 进行间接寻址。



● BCD 模式寻址 (*E)

当在一个 EM 地址前添加一个“*”符号时，该 EM 字的内容将被作为 BCD 值处理，以表示另一个 EM 区地址。

若 BCD 值介于 0 和 9999 之间，则最后一个地址也将位于同一个存储区块中。



● EM 区与文件存储器或跟踪存储器的转换

只需通过 CX-Programmer 指定 EM 区中的一个存储区块，即可将自指定存储区块开始至 EM 区末尾所有的存储区块转换为文件存储器或跟踪存储器。

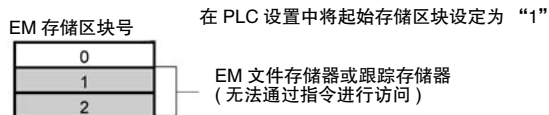
文件存储器：

文件存储器可用于存储文件，其形式如同在存储卡中存储文件一样。最多可将 13 个存储区块转换为文件存储器。

跟踪存储器：

跟踪存储器可用于保存数据跟踪结果，以增加一轮跟踪可存储的结果数量。

一旦 EM 存储区块被转换为文件存储器或跟踪存储器，就无法在用户程序中利用指令对其进行访问。如果尝试访问，则会产生非法访问错误。



6-16 定时器区

最多可使用 4,096 个定时器，其编号范围为 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)
定时器号用于访问完成标志。当定时器设定时间已过时，完成标志将置 ON。
- 定时器当前值 (PV)(T)
定时器号也可用于读写定时器的当前值 (16 位字)。当前值 (PV) 随定时器的运行而增大或减小。

当在需要位数据的操作数中使用定时器号时，定时器号可访问完成标志。当在需要字数据的操作数中使用定时器号时，定时器号可访问当前值 (PV)。

在下述所有指令中可使用相同的定时器号：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))、累加定时器 (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

不建议在两条定时器指令中使用同一个定时器号，否则它们同时计时时将会无法正常运行。(如果在两条或两条以上的定时器指令中使用同一个定时器号，则在程序检查时将会出错；但只要不在同一个循环中执行这些指令，定时器仍将正常运行。)

下表列出了定时器当前值 (PV) 和完成标志何时复位的说明。

指令名称 *1	对当前值 (PV) 和完成标志的影响			跳转和互锁时的动作	
	模式切换 (由 PROGRAM 到 RUN/MONITOR 或反之) 时 *2	PLC 启动时 *3	执行 TRSET(549) 时	跳转时 (JMP-JME) 或任务处于“WAIT”状态时 *4	互锁时 (IL-ILC)
100ms 定时器: TIM/TIMX(550)	当前值 (PV) → 0 标志 → OFF	当前值 (PV) → 0 标志 → OFF	当前值 (PV) → 保持 标志 → OFF	在操作定时器期间刷新当前值 (PV)	当前值 (PV) → 设定值 (SV) (复位为设定值 (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)*1					
0.01ms 定时器: TMUH(544)/TMUHX(557)*1					
累加定时器: TTIM(087)/TTIMX(555)					
定时器等待: TIMW(813)/TIMWX(816)				在操作定时器期间刷新当前值 (PV)	---
高速定时器等待: TMHW(815)/TMHWX(817)					---

*1 由于 TIML(542)、TIMLX(553)、MTIM(543) 和 MTIMX(554) 指令不使用定时器号, 因此将在不同条件下进行复位。(TIML(542) 和 TIMLX(553) 复位为其设定值, 而 MTIM(543) 和 MTIMX(554) 则复位为零。)有关详情, 请参考相关指令介绍。

*2 如果 IOM 保持位 (A500.12) 为 ON, 则在发生致命错误或运行模式从 PROGRAM 切换至 RUN/MONITOR 时或反之, 将保持当前值 (PV) 和完成标志。当电源重启时, 将清除当前值 (PV) 和完成标志。有关详情, 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 且已在 PLC 设置的“Startup” (启动) 选项项的“Startup Hold Area” (启动保持区) 中勾选“IOM Hold Bit” (IOM 保持位) 复选框, 则在电源重启时将保持当前值 (PV) 和完成标志。

*4 即使在 JMP 和 JME 指令之间发生跳转或当任务处于“WAIT”状态时, 仍将对使用定时器号 T0 ~ T2047 编程的 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)) 的当前值进行更新; 而使用定时器号 T2048 ~ T4095 编程的定时器的当前值, 则在发生跳转或当任务处于“WAIT”状态时保持不变。

- 可对定时器完成标志进行强制置位 / 复位。
- 定时器编号的使用顺序或常开或常闭条件的编程次数没有限制。
- 除 0.1ms 定时器 (TIMU(541)/TIMUX(556)) 和 0.01ms 定时器 (TMUH(544)/TMUHX(557)) 之外, 所有定时器的当前数据均可作为字数据进行读取。

6-17 计数器区

最多可使用 4,096 个计数器，其编号范围为 C0 ~ C4095。有两个计数器数据区：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)
计数器号用于访问完成标志。当达到计数器的设定值时，完成标志将置 ON。
- 计数器当前值 (PV)(C)
计数器号还可用于读写计数器的当前值 (16 位)。当前值 (PV) 随计数器的运行而增大或减小。

当在需要位数据的操作数中使用计数器号时，计数器号可访问计数器的完成标志。当在需要字数据的操作数中使用计数器号时，计数器号可访问计数器的当前值 (PV)。

在下述指令中可使用相同的计数器号：CNT、CNTX(546)、CNTR(012)、CNTRX(548)、CNTW(814) 和 CNTWX(818)。



正确使用注意事项

不建议在两条计数器指令中使用同一个计数器号，否则它们同时计数时将会无法正常运行。如果在两条或两条以上的计数器指令中使用同一个计数器号，则在程序检查时将会出错；但只要不在同一个循环中执行这些指令，计数器仍将正常运行。

下表列出了计数器当前值 (PV) 和完成标志何时复位的说明。

指令名称	对当前值 (PV) 和完成标志的影响					
	复位时	模式切换时	PLC 启动时	复位输入时	执行 CNR(545)/CN RX(547) 时	互锁时 (IL-ILC)
计数器：CNT/CNTX(546)	当前值 (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)						

- 可对计数器完成标志进行强制置位 / 复位。
- 计数器编号的使用顺序或常开或常闭条件的编程次数没有限制。
- 计数器当前值 (PV) 可作为字数据读取并用于编程。

6-18 任务标志

任务标志范围为 TK0 ~ TK127，分别对应循环任务 0 ~ 127。

任务标志将在相应循环任务处于“READY”或“RUN”状态时以及处于“INI”或“WAIT”状态时分别变为 ON 和 OFF。

注 这些标志只表示循环任务的状态，而不反映附加循环任务或中断任务的状态。

无论 IOM 保持位 (A500.12) 的状态如何，任务标志在下述情况下均会被清除。

- 运行模式从 PROGRAM 切换至 RUN/MONITOR 或与之相反时。
- PLC 电源重启时。

不可对任务标志进行强制置位 / 复位。

6-19 变址寄存器

16 个变址寄存器 (IR0 ~ IR15) 用于执行间接寻址。每个变址寄存器均可保留一个单独的 PLC 存储器地址, 即 I/O 存储器中字的绝对存储地址。这些地址与 CIO 区和 DM 区等中的 I/O 存储区地址不同是连续的 RAM 地址。

变址寄存器既可被各个任务所独用, 也可被全部任务所共享。

用户无法在变址寄存器中直接输入 PLC 存储器地址, 而需使用 MOVR(560) 指令将常规数据区地址转换为其对等的 PLC 存储器地址, 然后再将该值写入指定的变址寄存器中。(用 MOVW(561) 指令在变址寄存器中设定定时器 / 计数器当前值 (PV) 的 PLC 存储器地址。)



附加信息

有关 PLC 存储器地址的更多详情, 请参考 “A-4 PLC 存储器地址的存储器映射”。

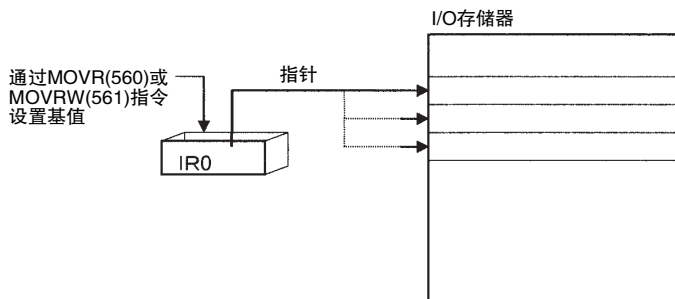
● 间接寻址

当变址寄存器作为一个带 “,” 前缀的操作数使用时, 指令将对变址寄存器中 PLC 存储器地址指示的字进行操作, 而不会对变址寄存器本身进行操作。基本而言, 变址寄存器为 I/O 存储器的指针。

- 利用 PLC 存储器地址可以无间断地指定 I/O 存储器中的所有地址 (变址寄存器、数据寄存器和完成标志除外), 而无需指定数据区。
- 除基本的间接寻址外, 还可利用常数或数据寄存器使变址寄存器中的 PLC 存储器地址发生偏移、自动递增或自动递减。该类功能可用于在循环中读写数据, 从而使地址在每次执行指令时均会递增或递减。

若要使用偏移和递增 / 递减变化, 可利用 MOVR(560) 或 MOVW(561) 指令将变址寄存器设定为基值, 然后在执行每条指令时, 作为指针进行修改。

对于指针操作, 既可采用可直接指定变址寄存器的特殊指令 (MOVR(560)/MOVW(561)、递增指令、递减指令或算术指令), 亦可采用间接偏移、自动递增或自动递减。



只需在 PLC 中进行相关设定, 即可使变址寄存器为各个任务所独用或为全部任务所共享。

在下述情况下, 变址寄存器将被清除:

- 运行模式在 PROGRAM 或 MONITOR 模式和 RUN 模式间切换时
- 接通 PLC 的电源时
- 发生致命错误时 (因 FALS(007) 而产生致命错误时除外)



正确使用注意事项

当利用变址寄存器对存储器进行间接寻址时, 可能会指定 I/O 存储器之外的区域并产生非法访问错误。有关 PLC 存储器地址的上下限详情, 请参考 “A-4 PLC 存储器地址的存储器映射”。

下表所示为利用变址寄存器对 I/O 存储器进行间接寻址时的可用变化形式。

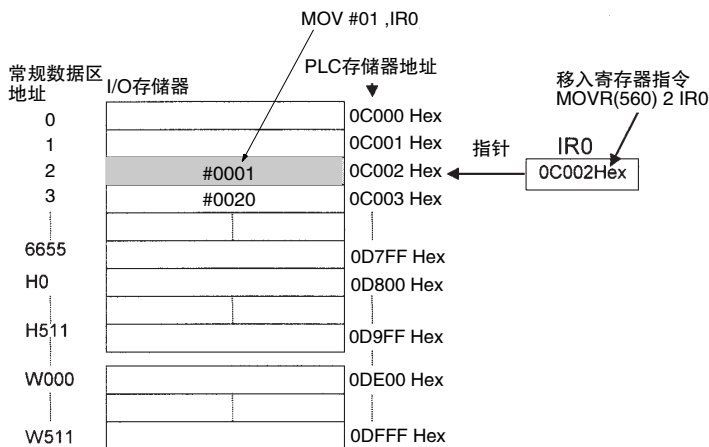
变化	功能	语法	示例
间接寻址	将 IR□ 中的内容作为 PLC 存储器地址 (位地址或字地址) 进行处理。	,IR□	LD, IR0 载入包含在 IR0 中的 PLC 存储器地址的位。
利用常数偏移进行间接寻址	将常数作为前缀添加到 IR□ 的内容中, 并将结果作为 PLC 存储器地址 (位地址或字地址) 进行处理。 该常数可为 -2,048 ~ 2,047 中的任意整数。	常数, IR□ (常数中包含 “+” 或 “-” 号)	LD +5, IR0 在将 5 加到 IR0 的内容中后, 载入该 PLC 存储器地址处的位。
利用 DR 偏移进行间接寻址	将数据寄存器的内容添加到 IR□ 的内容中, 并将结果作为 PLC 存储器地址 (位地址或字地址) 进行处理。	DR□, IR□	LD DR0, IR0 在将 DR0 中的内容加到 IR0 的内容中后, 载入该 PLC 存储器地址处的位。
利用自动递增进行间接寻址	在其与 PLC 存储器地址 (位地址或字地址) 进行参照后, IR□ 中的内容将递增 1 或 2。	递增 1: ,IR□+ 递增 2: ,IR□++	LD, IR0 ++ 载入包含在 IR0 中的 PLC 存储器地址后, IR0 中的内容递增 2。
利用自动递减进行间接寻址	在 IR□ 中的内容递减 1 或 2 后, 将结果作为 PLC 存储器地址 (位地址或字地址) 进行处理。	递减 1: ,-IR□ 递减 2: ,-IR□	LD, --IR0 在 IR0 中的内容递减 2 后, 载入该 PLC 存储器地址处的位。

注 IR□ 表示变址寄存器 IR0 ~ IR15。

● 示例

下述所示为在变址寄存器 (IR0) 中存储一个字形式的 PLC 存储器地址 (CIO 2)、在指令中使用变址寄存器以及使用自动递增变化的方法示例。

- MOV R(560) 2 IR0 在 IR0 中存储 PLC 存储器地址 CIO 2。
 MOV(021) #1, IR0 将 #1 写入 IR0 中包含的 PLC 存储器地址 (即 CIO 2) 中。
 MOV(021) #20 +1, IR0 在读取 IR0 并加 1 后, 将 #20 写入 PLC 存储器地址 (即 CIO 3) 中。



注 虽然上图列出了 PLC 存储器地址, 但在使用变址寄存器时无需知道该地址。

由于某些操作数作为字数据处理, 而另一些操作数则作为位数据处理, 因此其在变址寄存器中的含义将因各操作数的具体使用情况而异。

- 字操作数:
MOV R(560) 0 IR2
MOV(021) D0, IR2

当操作数作为字数据处理时, 变址寄存器中的内容将作为一个字形式的 PLC 存储器地址处理。本例中, MOV R(560) 指令将 CIO 2 的 PLC 存储器地址置入 IR2 中, 而 MOV(021) 指令则将 D0 中的内容复制到 CIO 2 中。

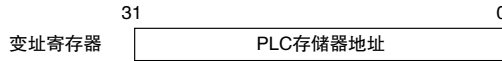
- 位操作数:
MOV R(560) 0.13, IR2
SET +5, IR2

变址寄存器亦可用于指定位，即如同在上述的 SET 指令一样。本例中，MOVR(560) 指令将 CIO 0.13 的 PLC 存储器位地址置入 IR2 中；然后，SET 指令对位 13 至该 PLC 存储器地址加 “+5”，从而对 CIO 1.02 位置 ON。

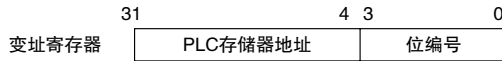


附加信息

当利用 MOVR(560) 指令在变址寄存器中设定字地址时，其地址存储如下所示：



当利用 MOVR(560) 指令在变址寄存器中设定位地址时，其地址存储如下所示：



正确使用注意事项

- 请务必在使用变址寄存器前设定变址寄存器 (IR) 值。若在未预设变址寄存器值的情况下而使用之，将无法实现可靠运行。
- 在启动一个中断任务后，将无法保证变址寄存器中值的可靠性。因此，当在一个中断任务中使用变址寄存器时，请务必在使用前利用 MOVR(560) 或 MOVRW(561) 指令对其值进行设定。(MOVRW(561)指令用于设定定时器/计数器的当前值(PV)，而 MOVR(560)指令用于设定其它值。)



附加信息

- 当变址寄存器作为不带 “,” 前缀的操作数使用时，指令将对变址寄存器自身的内容 (一个双字或 “双精度” 值) 进行操作。变址寄存器仅可在下表所示的指令中进行直接寻址。请使用下述指令对变址寄存器按指针形式进行操作。
- 虽然通常情况下，变址寄存器可用于间接寻址，但在任何其它指令中均无法实现直接寻址。

● 直接寻址

指令分组	指令名称	助记符
数据传送指令	传送至寄存器	MOVR(560)
	传送定时器 / 计数器的当前值 (PV) 至寄存器	MOVRW(561)
	双字传送	MOVL(498)
	双字数据交换	XCGL(562)
表数据处理指令	设置记录位置	SETR(635)
	获取记录号	GETR(636)
跟踪指令	无符号单字记录搜索指令	RSRCH (360 ~ 364)
	无符号单字记录排序	RSORT(203)
递增 / 递减指令	双字二进制递增	++L(591)
	双字二进制递减	--L(593)
比较指令	双字等于	=L(301)
	双字不等于	<>L(306)
	双字小于	<L(311)
	双字与小于等于	<=L(316)
	双字大于	>L(321)
	双字与大于等于	>=L(326)
	双字比较	CMPL(060)
四则运算指令	无进位带符号双字二进制加	+L(401)
	无借位带符号双字二进制减	-L(411)

SRCH(181)、MAX(182) 和 MIN(183) 指令可将带有所需值 (搜索值、最大值或最小值) 的 PLC 存储器地址 (字地址) 输出到 IRO 中。此时, 可在后续指令中使用 IRO, 以访问该字的内容。

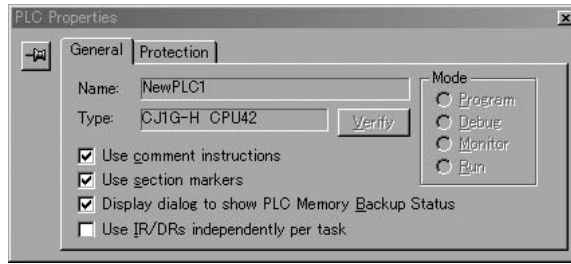
● 注意事项

- 在未设定 PLC 存储器地址前, 请勿使用变址寄存器。如果在未预设变址寄存器值的情况下使用, 将无法保证指针操作的可靠性。
- 由于在启动中断任务时, 将无法预测变址寄存器中的值, 因此, 若要在中断任务中使用变址寄存器, 请务必于在该任务中使用变址寄存器前、利用 MOVR(560) 或 MOVRW(561) 指令对 PLC 存储器地址进行设定。
- 各个任务中的变址寄存器均为独立执行状态, 因此相互之间不会产生影响。例如任务 1 中 IRO 和任务 2 中的 IRO 是不同的。因此, 每个变址寄存器任务均具有 16 个变址寄存器。
- 利用 CX-Programmer 仅可读取循环范围内最后执行任务所使用的变址寄存器。如果利用带有相同编号的变址寄存器执行多项任务, 则仅可利用 CX-Programmer 读取在多项任务循环范围内最后执行任务所使用的变址寄存器值, 而不可利用 CX-Programmer 写入变址寄存器值。

● 变址寄存器共享

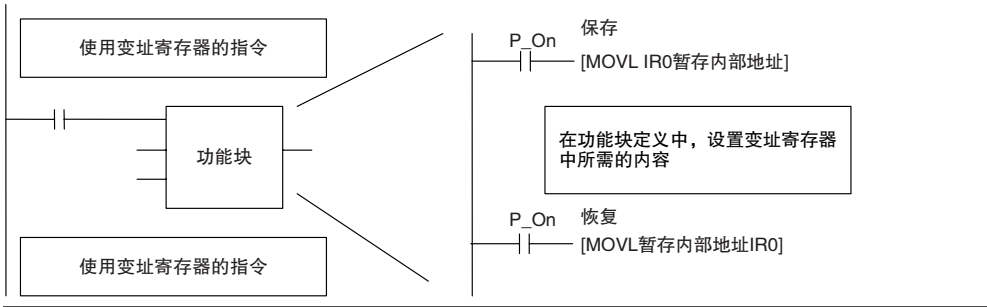
可在下示 CX-Programmer 的 “PLC Properties” (PLC 属性) 对话框中进行设定, 以控制任务间共享的变址和数据寄存器。

若要在任务间共享变址寄存器, 则请勿勾选 “PLC Properties” (PLC 属性) 对话框中的 “Use IR/DRs independently per task” (各任务独自使用 IR/DR) 复选框。



 附加信息

在调用功能块时，其中的变址寄存器内容可能会遭到破坏。因此，请务必在调用程序块前保存变址寄存器中的内容，并在调用结束后恢复其中的内容。请在功能块中设置变址寄存器中所需的内容。



6-20 数据寄存器

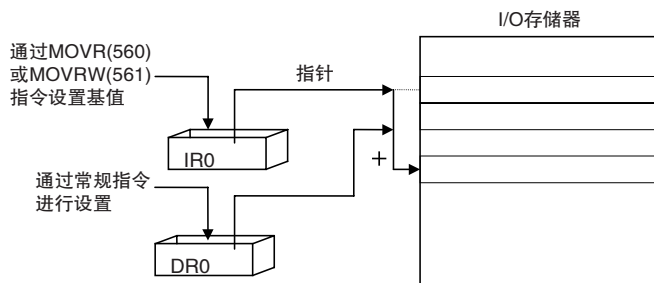
在对字地址执行间接寻址时，16 个数据寄存器 (DR0 ~ DR15) 用于对变址寄存器中的 PLC 存储器地址进行偏移。在对字地址进行间接寻址时，数据寄存器可用于指定加到变址寄存器上的偏移量。

数据寄存器既可供各个任务单独使用，亦可供所有任务共享。

数据寄存器中的值可加到变址寄存器的 PLC 存储器地址中，以指定 I/O 存储器中的位地址或字地址的绝对地址。由于数据寄存器中包含带符号二进制数据，因此变址寄存器中的内容可向前地址或后地址进行偏移。

常规指令可用于存储数据寄存器中的数据。

数据寄存器无法进行强制置位 / 复位。



● 示例

下示为利用数据寄存器在变址寄存器中执行 PLC 存储器地址偏移的方法示例。

```
LD DR0, IR0
```

在将 DR0 中的内容加到 IR0 的内容中后，载入该 PLC 存储器地址处的位。

```
MOV(021) #1 DR0, IR1
```

在将 DR0 中的内容加到 IR1 中后，将 #1 写入该 PLC 存储器地址中。

● 值的范围

由于数据寄存器中内容被作为带符号二进制数据处理，因此其范围为 $-32,768 \sim 32,767$ 。(负值以二进制补码的形式表示。)

十六进制内容	等值的十进制
8000 ~ FFFF	-32,768 ~ -1
0000 ~ 7FFF	0 ~ 32,767

● 数据寄存器初始化

在下述情况下，数据寄存器将被清除：

- 运行模式从 PROGRAM 切换至 RUN/MONITOR 或反之且 IOM 保持位为 OFF 时。
- PLC 电源重启且 IOM 保持位为 OFF 或在 PLC 设置中设定为“不保持”时。
- 发生致命错误时 (因 FALS(007) 而产生致命错误时除外)。

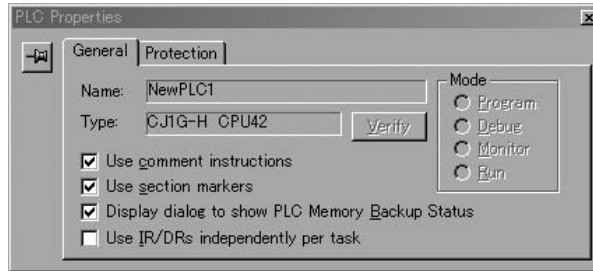
● IOM 保持位操作

如果 IOM 保持位 (A500.12) 为 ON，则在发生 FALS 错误或运行模式从 PROGRAM 切换至 RUN/MONITOR 时或反之，将不会清除数据寄存器。

如果 IOM 保持位 (A500.12) 为 ON 且已在 PLC 设置的“Startup” (启动) 选项页的“Startup Hold Area” (启动保持区) 中勾选“IOM Hold Bit” (IOM 保持位) 复选框，则在 PLC 电源重启时将不会清除数据寄存器。

● 数据寄存器共享

可在下示 CX-Programmer 的“PLC Properties”(PLC 属性)对话框中进行设定,以控制任务间共享的变址和数据寄存器。



附加信息

无论是否在程序中使用变址和数据寄存器,我们均建议将“PLC properties”(PLC 属性)设定为“在任务间共享变址和数据寄存器”。

6-21 条件标志

条件标志包括出错标志、进位标志、其它表示指令执行结果的标志以及常 ON 和常 OFF 标志。在早期的 PLC 中，这些标志均位于辅助区中。

条件标志由 P_CY 和 P_ER 等全局符号指定，而非通过地址指定；其无法通过指令或 CX-Programmer 直接写入。

由于程序进行任务切换时，所有的条件标志均被清除，因此，仅可保持出错任务中的出错标志、访问出错标志以及其它标志的状态。

不可对条件标志进行强制置位 / 复位。

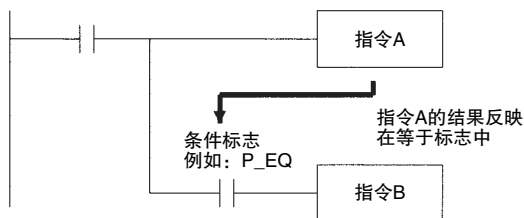
● 条件标志概述

下表所示为各条件标志的功能概述。

名称	符号	标签	功能
出错标志	P_ER	ER	当某条指令中的操作数不正确（指令处理出错）时变为 ON，表示指令由于出现错误而结束。 当在 PLC 设置设定为出现错误（指令操作错误）即停止运行时，则出错标志为 ON 时，程序将停止执行且指令处理出错标志（A295.08）也将变为 ON。
访问出错标志	P_AER	AER	当出现非法存取错误时为 ON。非法存取错误表示某指令试图访问禁止访问的存储器区。 当在 PLC 设置设定为出现错误（指令操作错误）即停止运行时，则当访问出错标志为 ON 时，程序将停止执行且指令处理出错标志（A295.10）也将变为 ON。
进位标志	P_CY	CY	当由于某一算术运算产生一个进位或者由某条数据移位指令将“1”下移入进位标志时，进位标志变为 ON。 进位标志为某些数据移位指令和符号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其值超出规定范围时，该标志为 ON。
等于标志	P_EQ	=	当比较指令的两个操作数相等或计算结果为 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其值小于规定范围时，该标志为 ON。
负标志	P_N	N	当结果的最高有效位（符号位）为 ON 时，该标志为 ON。
上溢标志	P_OF	OF	当运算结果超出结果字（s）上限时，该标志为 ON。
下溢标志	P_UF	UF	当运算结果低于结果字（s）下限时，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	始终为 ON。
常 OFF 标志	P_Off	OFF	始终为 OFF。

● 条件标志的使用

由于条件标志由所有指令共享，因此，在一个单独循环内，其状态可能会经常发生变化。为此，请务必在执行完指令后立即读取条件标志的状态，最好能在采用相同执行条件的分支处进行读取。



指令	操作数
LD	
指令A	
AND	P_EQ
指令B	



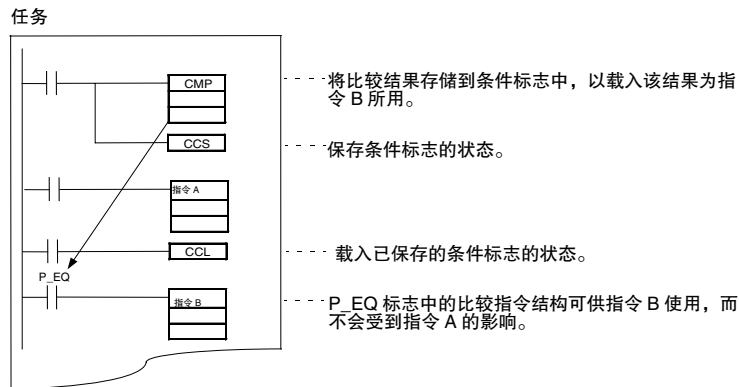
正确使用注意事项

- 由于条件标志由所有指令共享，因此，程序可能会因某个单独任务发生中断而改变其预期运行进程。编程时，请务必考虑到中断造成的影响。
- 由于程序进行任务切换时，条件标志均被清除，因此条件标志的状态无法传递至另一个任务中。例如：在任务 2 中无法读取任务 1 中的条件标志状态。

● 条件标志状态的保存和载入

条件标志状态指令 (CCS(282) 和 CCL(283)) 用于保存和载入在一个任务 (程序) 中的不同位置或在不同任务或循环中的条件标志的状态。

下示为在同一个任务的不同位置处使用等于标志的方法示例。

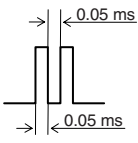
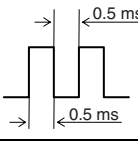
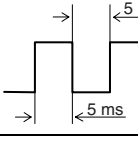
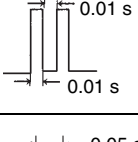
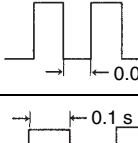
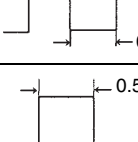
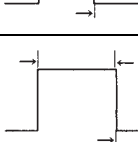



6-22 时钟脉冲

时钟脉冲是系统按照一定时间间隔置 ON 和 OFF 的标志。

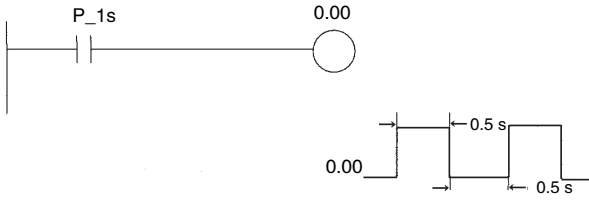
时钟脉冲为只读形式，不可通过指令或 CX-Programmer 进行覆写；其会在运行开始时被清除。

时钟脉冲采用 P_0_1ms 和 P_1ms 等全局符号进行指定。

名称	符号	编程器上显示的名称	动作
0.1ms 时钟脉冲	P_0_1ms	0.1ms	 ON: 0.05ms OFF: 0.05ms
1ms 时钟脉冲	P_1ms	1ms	 ON: 0.5ms OFF: 0.5ms
0.01s 时钟脉冲	P_0_01s	0.01s	 ON: 5ms OFF: 5ms
0.02s 时钟脉冲	P_0_02_s	0.02s	 ON: 0.01s OFF: 0.01s
0.1s 时钟脉冲	P_0_1s	0.1s	 ON: 0.05s OFF: 0.05s
0.2s 时钟脉冲	P_0_2s	0.2s	 ON: 0.1s OFF: 0.1s
1s 时钟脉冲	P_1s	1s	 ON: 0.5s OFF: 0.5s
1min 时钟脉冲	P_1min	1min	 ON: 30s OFF: 30s

- 时钟脉冲的使用

下例中，CIO 0.00 将按照 0.5s 的间隔置 ON 和 OFF。



指令	操作数
LD	P_1s
OUT	0.00

- 时钟脉冲刷新

即使在执行程序过程中，时钟脉冲也会被刷新。因此在程序开始和结束时，ON/OFF 状态可能会不同。

- 时钟脉冲错误

时钟脉冲的最大错误为 0.01%(25 °C 时)。对于基于时间的长期控制而言，我们建议采用内部时钟代替时钟脉冲，以确保容许内部时钟出错。

7

文件操作

本章节阐述了对 CJ2 CPU 单元执行的文件操作。

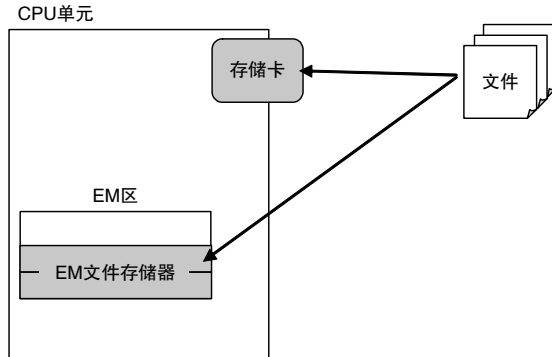
7-1 文件存储器	7-2
7-1-1 文件存储器类型	7-2
7-1-2 文件存储器初始化	7-3
7-1-3 存储卡注意事项	7-5
7-2 文件存储器中存储的文件类型	7-7
7-2-1 文件类型	7-7
7-2-2 文件存储器文件的创建和保存	7-10
7-3 文件存储器操作	7-11
7-3-1 文件存储器操作类型	7-11
7-3-2 文件存储器操作步骤和文件存储器文件	7-13
7-3-3 文件使用的限制	7-18
7-3-4 文件大小	7-18
7-3-5 支持软件和文件存储器文件之间的关系	7-19

7-1 文件存储器

7-1-1 文件存储器类型

文件存储器可用于存储 CJ 系列 PLC 中的文件。其两种文件存储器类型如下所示：

- 存储卡
- EM 区中的指定范围，即所谓的“EM 文件存储器”。



对于 CJ 系列 PLC 而言，存储卡和指定范围的 EM 区可作为存储器使用，用于存储文件，或将整个用户程序、I/O 存储器和参数区以文件形式进行保存。

类别	存储器类型	存储器容量	可存储文件类型
存储卡	闪存	HMC-EF183: 128M 字节 HMC-EF283: 256M 字节 HMC-EF583: 512M 字节	<ul style="list-style-type: none"> · 程序文件 · 注释文件 · 程序索引文件
内置 RAM (EM 文件存储器)	RAM	EM 区共划分为 13 个存储区块，即从在 EM 区中指定的存储区块到 EM 区中的最后一个存储区块。	<ul style="list-style-type: none"> · 符号表文件 · 参数文件 · 单元和板备份文件 (仅限存储卡)

7-1-2 文件存储器初始化

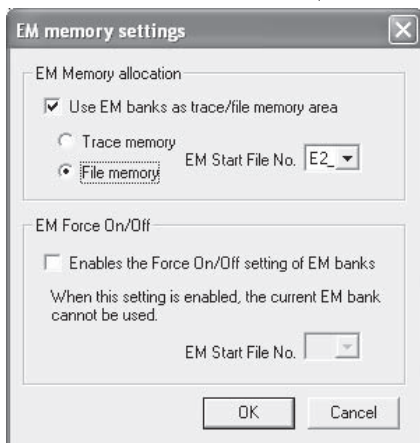
存储卡

利用 CX-Programmer 中的存储卡对话框即可实现存储卡的初始化。由于 HMC-EF□□□ 存储卡已经过默认初始化处理，因此首次使用时，无需再进行初始化。

EM 文件存储器

请按照下述步骤指定作为文件存储器使用的 EM 文件存储器中的首个存储区块，并对 EM 文件存储器进行初始化处理。

1. 在 CX-Programmer 中选择 “PLC – Memory Allocate – EM Memory Settings” (PLC – 存储器分配 – EM 存储器设定)。
2. 选择 “EM memory settings” (EM 存储器设定) 对话框中的 “File Memory” (文件存储器) 选项，然后设定 “EM Start File No.” (EM 起始文件号)。



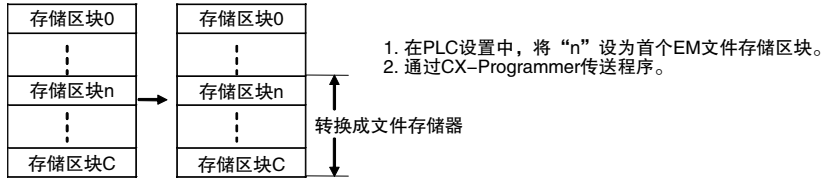
附加信息

- EM 区无法同时用作文件存储器和跟踪存储器，每次只能使用一种功能。
- 用作跟踪存储器和文件存储器的 EM 存储区块无法执行强制置位 / 复位。

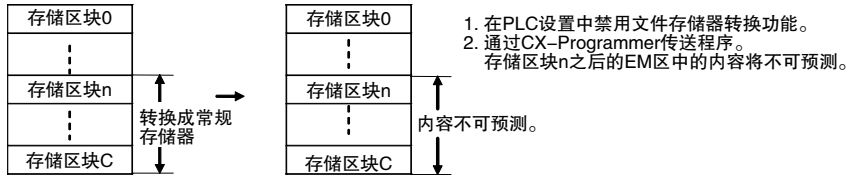
3. 将 CPU 单元进行联机，并传送用户程序，然后即可使用 EM 文件存储器。

● EM 文件存储器设定的变更

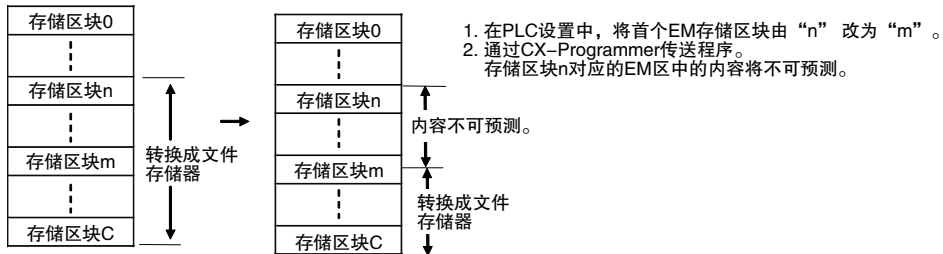
下图所示为将 EM 区 (自指定存储区块到最后一个存储区块) 转换成文件存储器的情形。



下图所示为将作为文件存储器的 EM 存储区块恢复为常规 EM 存储区的情形。



文件存储器的起始存储区块号可以变更。



7-1-3 存储卡注意事项

使用存储卡前应确认以下事项。

格式

由于存储卡已在出货前进行过格式化处理，因此购买后无需再执行格式化处理。若要在使用后进行格式化，应将存储卡插入 CPU 单元，并利用 CX-Programmer 进行操作。若直接在笔记本电脑或其它计算机上格式化存储卡，CPU 单元可能会无法识别存储卡。此时，即使在 CPU 单元上再次执行格式化处理，该存储卡也将无法使用。

根目录下的文件数

存储卡根目录下的文件存放数是有限制的（类似于硬盘）。该限制取决于存储卡的类型和格式，一般介于 128 ~ 512 个文件之间。在使用按照指定间隔写入日志文件或其它文件的应用场合时，应将该类文件写入子目录中，而非根目录中。可在计算机上使用 CMND(490) 指令创建子目录。

写入次数

一般来说，闪存的写入次数是有限制的。对于具有质保期限的存储卡而言，其反复写入次数为 100,000 次。例如，如果每隔 10 分钟对存储卡执行一次写入操作，则其写入次数在两年内便会超过 100,000 次。

最小文件大小

如果存储卡内保存了大量零星的小文件（如仅包含几个字的 DM 区数据文件），将无法充分利用存储卡的空间。例如，若使用单元配额大小为 4,096 字节的存储卡，则无论文件大小如何，每个文件均会至少占用 4,096 字节的存储空间。若要将 10 字的 DM 区数据保存至存储卡，即使文件的实际大小仅为 68 字节，也会占用 4,096 字节的存储空间。若大量保存此类小文件，则会大幅降低存储卡空间的利用率。若为提高利用率而降低单元配额，则会降低访问速度。

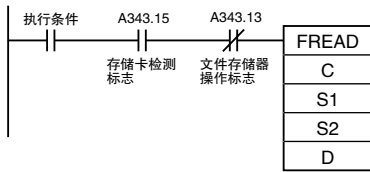
存储卡的单元配额可通过在 DOS 提示符下运行 CHKDSK 来进行检查。此处不再赘述。更多有关单元配额的信息，请参考通用计算机文献。

存储卡识别时间

插入存储卡后，CPU 单元通常需要花费几秒钟进行识别。该所需时间取决于 PLC 的循环时间、存储卡容量、存储卡上存储的文件数以及其它因素。例如，在循环时间为 0.4ms 且所有 PLC 设置中的参数均为默认值的情况下，HMC-EF183 存储卡的识别时间约为 8s。

存储卡的访问

- 切勿在“BUSY”指示灯点亮的情况下（即指示灯表示正在访问存储卡时）关闭 CPU 单元的电源，否则可能会造成存储卡无法使用。
- 切勿在“BUSY”指示灯点亮的情况下从 CPU 单元中取出存储卡。此时应按下存储卡电源“OFF”按钮，等待“BUSY”指示灯熄灭后，方可取出存储卡，否则可能会造成存储卡无法使用。
- 插入存储卡后，CPU 单元将需要花费几秒钟进行识别。若要在接通电源或插入存储卡后立即访问存储卡，应按下图所示为存储卡识别标志 (A343.15) 设置一个常闭输入条件。



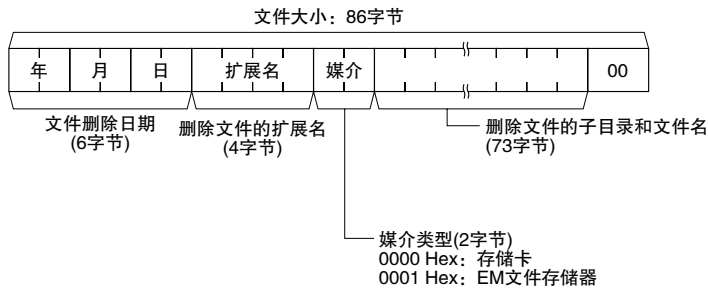
访问文件存储器时的电源中断

如果在 CPU 访问文件存储器 (存储卡或 EM 文件存储器) 时发生电源中断, 则存储卡中的内容可能会不准确: 正在更新的文件可能未被正确覆写, 且在某些情况下, 存储卡本身可能也会遭到破坏。

在下次接通电源时, 系统会自动将已受影响的文件删除。而相应的文件删除通知标志 (存储卡: A395.06 ; EM 文件存储器: A395.07) 将在此时变为 ON, 并在下次关闭电源时变为 OFF。

当一个文件被删除时, 将在存储卡或 EM 文件存储器的根目录下创建一个删除日志文件 (DELETE.TMP)。可利用二进制格式编辑器对该删除日志文件进行读取, 以核查下述信息: 删除文件的日期、现有文件存储器 (媒介) 的类型、子目录、文件名及扩展名。必要时, 请重新创建或重新复制被删除的文件。

下图所示为删除日志文件的结构。



7-2 文件存储器中存储的文件类型

文件存储器可用于下述情况：

- 程序 / 网络符号文件
- 参数文件
- 数据文件
- 符号表文件 (网络符号除外)
- 注释文件
- 程序索引文件
- 单元备份文件

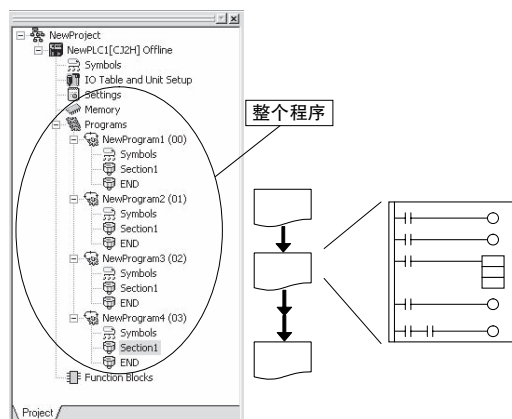
7-2-1 文件类型

程序 / 网络符号文件

程序 / 网络符号文件包含 CPU 单元的用户程序 (即循环任务和中断任务中的程序) 和网络符号 (即全局符号表中的网络符号)*。

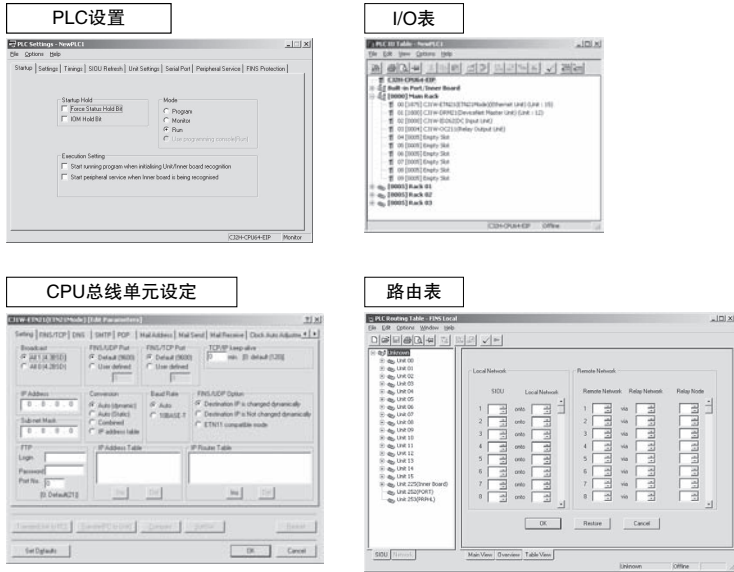
此外, 该文件还包含每个程序的属性。

* 仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 支持此项功能。



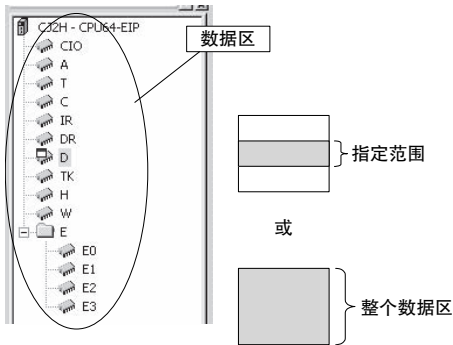
参数文件

参数文件包含 CPU 单元的内部参数区数据，即 PLC 的名称、PLC 设置，I/O 表、CPU 总线单元设定 (含数据链接参数) 和路由表。



数据文件

数据文件包含一个 I/O 存储器数据区中的数据，以字 (16 位) 为单位。其可用于存储整个数据区或指定地址范围内的所有数据。数据文件可存储下述 6 个数据区中的任意一个区：CIO 区、保持区、工作区、辅助区、DM 区和 EM 区。



符号表文件 (网络符号除外)

符号表文件 (网络符号除外) 包含 CX-Programmer 使用的符号表信息以及自动分配的 PLC 地址。

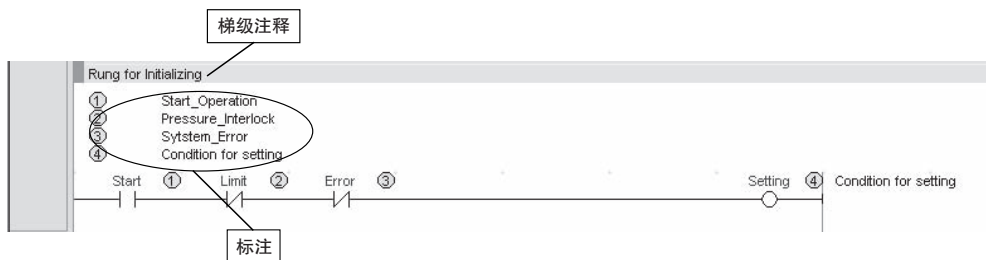


The screenshot shows the 'NewProject' window with a tree view containing 'NewPLC1 ([24H] Offline)' and 'NewProgram1 (00)'. Both have 'Symbols' folders. Callouts point to '全局符号表' (Global Symbol Table) and '局部符号表' (Local Symbol Table).

符号	地址	数据类型	I/O注释
P_CY		BOOL	CP004
P_Cycle_Time_Error		BOOL	A401.00
P_Cycle_Time_Value		LDINT	A254
P_EH		WORD	A450
P_EH0		WORD	A461
P_EH1		WORD	A462
P_EH2		WORD	A463
P_EH3		WORD	A464
P_EH4		WORD	A465
P_EH5		WORD	A466
P_EH6		WORD	A467
P_EH7		WORD	A468
P_EH8		WORD	A469
P_EH9		WORD	A470
P_EH4		WORD	A471
P_EH8		WORD	A472
P_EH9		WORD	A473
P_EQ		BOOL	CP006
P_EQ		BOOL	CP008

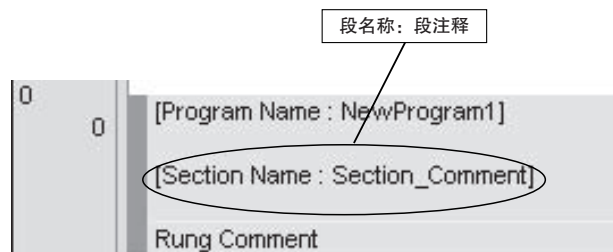
注释文件

注释文件包含 CX-Programmer 使用的注释信息。



程序索引文件

程序索引文件包含 CX-Programmer 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 (即简易备份功能所使用的数据)。其在执行简易备份功能时进行创建, 用于存储各个单元的内部数据。

例如:

DeviceNet 单元: Device 参数

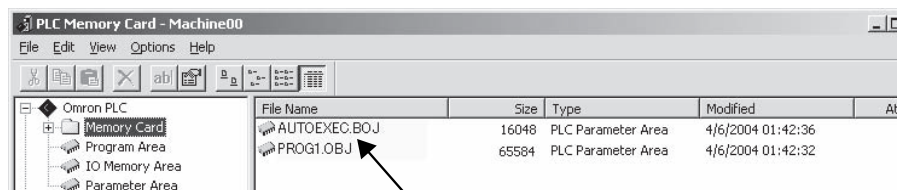
串行通信单元: 协议宏数据

7-2-2 文件存储器文件的创建和保存

用户程序、I/O 存储器 and 参数区数据

利用 CX-Programmer 中的 “PLC Memory Card” (PLC 存储卡) 对话框实现数据传送。

1. 将经过初始化处理的存储卡插到 CPU 单元上或对 EM 文件存储器执行初始化处理。有关 EM 文件存储器初始化的信息，请参考 “7-1-2 文件存储器初始化”。
2. 将 CX-Programmer 与 CPU 单元进行联机。
3. 在 “PLC Memory Card” (PLC 存储卡) 对话框中选择传送目的地 (即存储卡或 EM 文件存储器)，然后选择文件菜单中的 “Transfer” (传送) 选项。
CPU 单元中的数据 (即用户程序、I/O 存储器和参数区) 将以文件形式被保存到存储卡或 EM 文件存储器中。



启动时自动传送的程序文件

符号表、注释和程序索引

利用 CX-Programmer 实现数据传送。

1. 将经过格式化处理的存储卡插入 CPU 单元中或对 EM 文件存储器进行格式化处理。有关 EM 文件存储器初始化的信息，请参考 “7-1-2 文件存储器初始化”。
2. 将 CX-Programmer 与 CPU 单元进行联机。
3. 在 PLC 菜单中选择 “Transfer” (传送)，然后再选择 “To PLC” (传送至 PLC) 或 “From PLC” (从 PLC 传送) 选项。
4. 选择 “Symbols” (符号) 或 “Comments” (注释) 作为数据进行传送。

注 如果已将存储卡插入 CPU 单元中，则仅可通过存储卡进行数据传送。(无法通过 EM 文件存储器进行传送。)

7-3 文件存储器操作

7-3-1 文件存储器操作类型

利用文件存储器执行的操作如下所示：

- 利用 CX-Programmer 执行的操作
- 启动时的自动传送
- 简易备份功能
- FREAD(700) 和 FWRT(701) 指令
- 利用辅助区控制位替换整个程序
- FINS 命令

利用 CX-Programmer 执行的操作

有关下述步骤详情，请参考《CX-Programmer 操作手册》。

● 在 CX-Programmer 中的“Memory Card”（存储卡）对话框中传送数据

可创建、读取和写入文件如下所示：

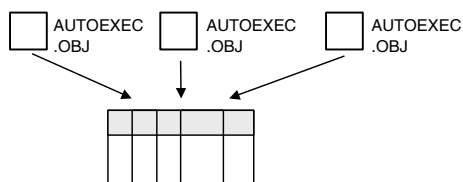
- 启动时自动传送的文件（程序 / 网络符号文件、数据文件和参数文件）
- 通用文件（程序 / 网络符号文件、数据文件和参数文件）
- CX-Programmer 文件（符号文件、注释文件和程序索引文件）

● 联机 / 脱机传送数据

通用文件（即程序 / 网络符号文件、数据文件和参数文件）可在联机状态下传送至 PLC 或在脱机状态下传送至计算机存储设备中。

启动时的文件自动传送

转换程序、参数和 I/O 存储器可存储在存储卡中，并在启动时从存储卡中进行读取。（详情请参考“第 10 章 CPU 单元功能”。）



简易备份操作

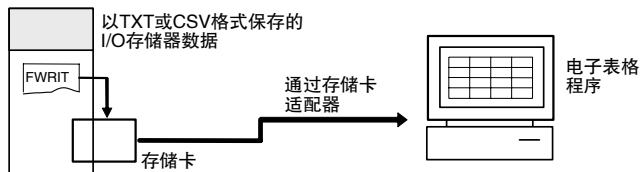
简易备份功能可在无需编程设备的情况下，将所有的 PLC 数据备份至存储卡中或从存储卡中恢复所有的 PLC 数据；其中，PLC 数据包括 CPU 单元、高功能 I/O 单元和 CPU 总线单元中的数据。（详情请参考《CJ2 CPU 单元硬件操作手册》（样本编号：W472）。）

备份恢复操作

所有的 PLC 数据，包括 CPU 单元、高功能 I/O 单元和 CPU 总线单元中的数据，均可通过 PLC 备份工具在计算机上进行备份和恢复。（详情请参考《CJ2 CPU 单元硬件操作手册》（样本编号：W472）。）

FWRIT(701)/FREAD(700) 指令

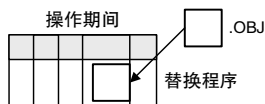
通过在用户程序中使用 FWRIT(701) 指令，即可将文本格式或 CSV 格式的 I/O 存储器数据保存到存储卡或 EM 文件存储器中。然后，通过存储卡适配器即可实现数据传送且可利用电子表格程序进行编辑。



反之，如高性能 I/O 单元设定等数据可通过电子表格程序以文本格式或 CSV 格式进行创建、存储在存储卡中，并利用 FREAD(700) 指令读取到 CPU 单元中。有关详情，请参考《CS/CJ/NSJ 系列指令参考手册》(样本编号：W474)。

操作期间利用辅助区控制位操作进行的用户程序替换

操作期间，利用辅助区控制位即可实现对整个用户程序的替换。(该替换是通过“程序本身”而非“编程设备”实现的。)(详情请参考“第 10 章 CPU 单元功能”。)



FINS 命令

使用 FINS 命令即可实现对文件存储器的控制。FINS 命令可由采用上位链接协议的相连计算机进行发送。在用户程序中，网络中的另一台 PLC 可利用 CMND(490) 指令发送 FINS 命令，本地 PLC 可利用 CMND(490) 指令发送 FINS 命令。(详情请参考《通信指令参考手册》(样本编号：W474)。)

7-3-2 文件存储器操作步骤和文件存储器文件

下表总结了由各种文件存储器操作所控制的文件存储器文件。

读：从文件存储器将文件传送至 CPU 单元。

写：从 CPU 单元将文件传送至文件存储器。

OK：允许；---：不允许

操作步骤	媒介	操作	文件类型				
			程序 / 网络符号文件	数据文件	参数区数据	符号表、注释文件和程序索引文件	单元备份文件
利用 CX-Programmer 执行操作	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 (CX-Programmer 文件)	---
		读	OK(启动时自动传送的文件)				
		写					
		其它操作					
启动时的自动传送	存储卡	读	OK(启动时自动传送的文件)			---	
		写	---				
简易备份功能	存储卡	读	OK(简易备份文件)				
		写					
利用 PLC 备份工具进行备份或恢复	仅限计算机存储设备	读	OK(PLC 备份文件)				
		写					
FREAD(700) 和 FWRT(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)	---	---	
		向一个文件写入数据					
在操作期间利用辅助区控制位操作替换整个程序	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					
		其它操作					

启动时自动传送的文件

该类文件用于在启动时批量读取用户程序/网络符号、参数数据和 I/O 存储器，其文件名固定为“**AUTOEXEC**”或“**REPLACE**”。详情请参考“10-3-5 启动时自动传送”。

备份文件

利用 CPU 单元前部的 DIP 开关或存储卡的电源开关即可将该类文件保存到存储卡中，以及备份所有的 PLC 数据。文件名固定为“**BACKUP□□**”。详情请参考《CJ2 CPU 单元硬件操作手册》(样本编号：W472)中的“8-3 简易备份”。

PLC 备份工具文件

该类文件可备份所有的 PLC 数据，并可通过 PLC 备份工具执行的备份和恢复操作进行创建。详情请参考《CJ2 CPU 单元硬件操作手册》(样本编号：W472)中的“8-2 使用计算机备份数据”。

CX-Programmer 文件

该类文件由 CX-Programmer 创建。在文件存储器中创建的文件名为固定形式。当项目被传送时，可选择下述任何一种存储媒介作为该类 CX-Programmer 文件的传送目的地。

- 注释存储器 (位于 CPU 单元的闪存内)
- 存储卡
- EM 文件存储器

类型	名称	内容	备注
符号表文件 (网络符号除外)	SYMBOLS.SYM	全局符号表(网络符号除外)和局部符号表	<ul style="list-style-type: none"> · 包含 CX-Programmer 全局 / 局部符号表中的变量、地址、数据类型和 I/O 注释信息。(网络符号除外) · 还包含在 CX-Programmer 自动 PLC 地址分配中设定的数据。
注释文件	COMMENTS.CMT	梯级注释和注释(标注)	包含 CX-Programmer 的梯级注释和注释信息。
程序索引文件	PROGRAM.IDX	段名称和段注释	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符指令。

注 利用 CX-Programmer，可在 CX-Programmer 和个人计算机 RAM 之间以及个人计算机 RAM 和存储器的存储设备之间实现对上表所含符号表文件和注释文件(网络符号除外)的联机传送。

当从 CX-Programmer 将一个项目传送至 CPU 单元时，符号表、注释文件和程序索引文件可被存储到 CPU 单元的内置闪存中；而当执行简易备份操作时，存储在注释存储器中的符号表、注释文件和程序索引文件则被自动创建为备份文件并存储到存储卡中。

通用文件

利用 CX-Programmer、FINS 命令、指令和辅助区位即可实现对该类文件的读写及其它操作。

文件名由用户进行定义。

文件类型	文件名 *1	扩展名	内容	备注	
程序 / 网络符号文件	*****	.OBJ	<ul style="list-style-type: none"> · 用户程序 · 网络符号 *2 	CPU 单元循环任务和中断任务中的程序和网络符号	
参数文件	*****	.STD	<ul style="list-style-type: none"> · PLC 设置 · 已注册的 I/O 表 · 路由表 · CPU 总线单元设置区 	<ul style="list-style-type: none"> · CPU 单元的参数区数据 · 用户无需区分文件中的数据。 	
数据文件	*****	.IOM	指定范围的 I/O 存储器	<ul style="list-style-type: none"> · 从一个区的起始字到结束字之间的字数据(16 位)递增。 · 可为下述各区: CIO、HR(保持区)、WR(工作区)、AR(辅助区)、DM(数据存储区)或 EM(扩展存储区)。 	二进制数据格式
		.TXT			文本格式(无分隔符或制表符)*3
		.CSV			CSV 格式(采用逗号分隔符)

*1 上述 ***** 表示文件名由 8 个或 8 个以下的 ASCII 字符构成。

*2 仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 支持此项功能。

*3 文本格式和 CSV 格式的数据文件仅可通过 FWRIT(701) 和 FREAD(700) 指令而非 CX-Programmer 进行读写。

● 通用数据文件

数据文件类型

通用数据文件扩展名有三种：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 .CSV	在此类数据格式中，I/O 存储器中的单字或双字字段会被转换成 ASCII 数据。记录可利用回车进行分隔。	利用电子表格软件进行数据交换

*1 读写 TXT 和 CSV 数据文件：

TXT 和 CSV 数据文件仅可利用 FREAD(700) 和 FWRTIT(701) 指令进行读写。

下述 6 种数据格式可用于文本文件和 CSV 文件中。

扩展名	数据格式	内容	
		每个字段的 I/O 存储器大小	分隔符
.TXT*1	无分隔符的字	单字	无
	无分隔符的双字	双字	无
	带制表位的字	单字	制表位代码
	带制表位的双字	双字	制表位代码
.CSV*1	带逗号分隔符的字	单字	逗号
	带逗号分隔符的双字	双字	逗号

*1 字符注意事项：

若 TXT 或 CSV 文件中含有十六进制字符(0 ~ 9、A ~ F 或 a ~ f)以外的字符，则不能将数据正确写入到 I/O 存储器中。

· 字段大小注意事项：

使用单字时，若 TXT 或 CSV 文件中含有非 4 位十六进制的字段，则不能将数据正确写入到 I/O 存储器中。同样在使用双字时，若 TXT 或 CSV 文件中含有非 8 位十六进制的字段，则不能将数据正确写入到 I/O 存储器中。

· 存储顺序：

使用单字时，I/O 存储器中的数据会被转换成 ASCII 码，并按照从最低到最高 I/O 存储器地址的顺序依次存储到一个单字字段中。

使用双字时，I/O 存储器中的数据会被转换成 ASCII 码，并按照从最低到最高 I/O 存储器地址的顺序依次存储到一个双字字段中。(在双字字段中，先存储高位地址字，后存储低位地址字。)

· 分隔符：

在无分隔符的情况下，会在连续排列字段后再进行存储。在带逗号分隔符时，进行存储前会先在字段间插入逗号。带制表位分隔符时，进行存储前会先在字段间插入制表位代码。

当在 FREAD(700) 指令中指定分隔符(逗号或制表位)时，数据则被读取为附带单字分隔符(逗号或制表位)的数据。

· 回车：

在不使用回车时，会将数据进行连续排列。

在使用回车时，会在指定编号的字段之后插入回车代码。若在文件中使用回车，则无法利用 FREAD(700)/FWRTIT(701) 指令指定离文件开头(起始读取字或起始写入字)的偏移量。

· 字段数量：

文件中的数据总量取决于在 FWRTIT(701) 指令中指定的字段数(写入项目数)以及字数/字段数。长度由字段数进行指定。

对于字数据而言，I/O 存储器中的 1 个字 = 1 个字段

对于长字数据而言，I/O 存储器中的 2 个字 = 1 个字段。

数据文件名

由于数据文件不包含指明保存的数据等信息，即保存的是哪个存储区中的数据，因此，请务必采用如下例所示能够指明保存内容的文件名，以便于进行文件管理。

例如：D00100.IOM，CIO0020.IOM

即使写入到数据文件 (IOM、TXT 或 CSV) 中的原始数据来自不同的区，文件起始处的数据仍将被写入到 I/O 存储器中指定的起始地址中。例如，在利用编程设备将文件中的 CIO 数据写入到 DM 区中的情况下，该数据将被读取到 CPU 单元的 DM 区中，而不会出现任何表示区域不同的提示。



附加信息

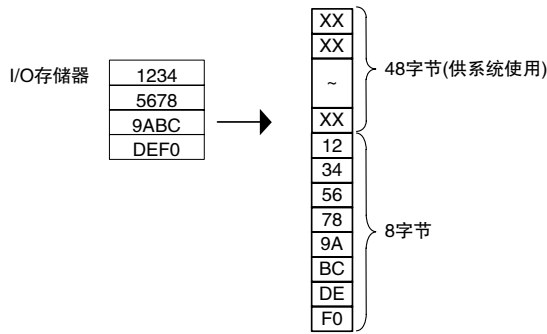
由于 TXT 和 CSV 格式的数据文件含有十六进制数据 (0 ~ 9、A ~ F)，因此可以实现 I/O 存储器中的数值与电子表格程序之间的数据交换。

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函数 WRITE_TEXT() 保存为文本文件。



附加信息

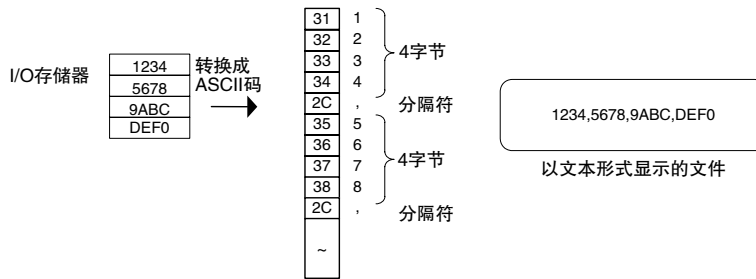
下图所示为 I/O 存储器中包含 4 字的数据文件 (ABC.IOM) 的二进制数据结构：1234 Hex、5678 Hex、9ABC Hex 和 DEF0 Hex。然而用户在正常的操作时并不需要考虑数据格式。



ABC.IOM 的内容

· 单字字段形式的 CSV 和 TXT 数据文件结构

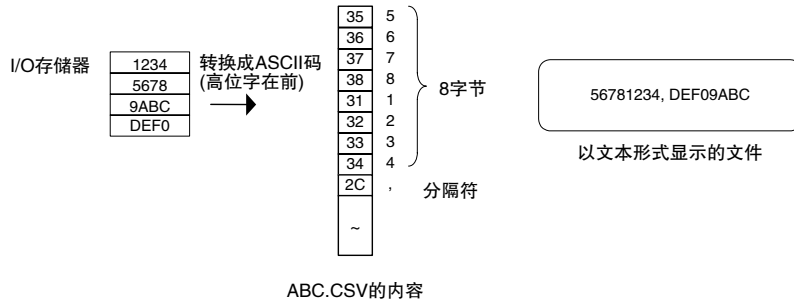
下图所示为 I/O 存储器中包含 4 字 (单字字段形式) 的 CSV 数据文件 (ABC.CSV) 的数据结构：1234 Hex、5678 Hex、9ABC Hex 和 DEF0 Hex。



ABC.CSV 的内容

· 双字段形式的 CSV 和 TXT 数据文件结构

下图所示为 I/O 存储器中包含 4 字 (双字段形式) 的 CSV 数据文件 (ABC.CSV) 的数据结构: 1234 Hex、5678 Hex、9ABC Hex 和 DEF0 Hex。



附加信息

利用电子表格软件创建数据文件

利用电子表格软件 (如 Microsoft Excel 等) 创建 TXT 和 CSV 数据文件的操作步骤如下所示:

- 将单元格内容设定为字符形式。
- 若采用单字段, 则在每个单元格中输入 4 个字符; 若采用双字段, 则在每个单元格中输入 8 个字符。例如, 在采用单字段时不应只输入 A, 而应输入 000A。
- 单元格中只能输入十六进制字符 (0 ~ 9、A ~ F 或 a ~ f), 而无法使用其它字符和代码。

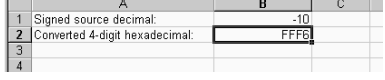

若要在 I/O 存储器中存储十六进制数值, 则将电子表格中的十进制输入转换成十六进制将大有益处。请按照下述步骤进行十进制到十六进制的转换:

- (1) 从工具菜单中选择 “Add-Ins...” (加载宏) 选项。
- (2) 在 “Add-Ins” (加载宏) 菜单中选择 “Analysis ToolPak” (分析工具库) 选项。
- (3) 从插入函数菜单中选择单元格要使用的 “Function” (函数)。
- (4) 从 “Category” (选择类别) 字段的 “Engineering” (工程函数) 中选择 “DEC2HEX(number, digits)” (DEC2HEX(数值, 数位))。
- (5) 若要转换成 4 位十六进制值, 请在数字变量处输入下述信息:
IF(0<= 单元格位置, 单元格位置, 65535+ 单元格位置)
若要转换成 8 位十六进制值, 请在数字变量处输入下述信息:
IF(0<= 单元格位置, 单元格位置, 4294967296+ 单元格位置)。

例 1: 非负十进制值的输入

项目	将无符号十进制数转换成 4 位十六进制数	将无符号十进制数转换成 8 位十六进制数
应用函数	DEC2HEX(单元格_位置, 4)	DEC2HEX(单元格_位置, 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：带符号十进制值的输入

项目	将无符号十进制数转换成 4 位十六进制数	将无符号十进制数转换成 8 位十六进制数
应用函数	DEC2HEX(IF(0<= 单元格_位置, 单元格_位置, 65536+ 单元格_位置), 4)	DEC2HEX(IF(0<= 单元格_位置, 单元格_位置, 4294967296+ 单元格_位置), 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

由于文件采用 DOS 模式下的格式化处理, 因此可在配备 Windows 系统的计算机中作为常规文件使用。

● 文件名

文件可通过文件名和扩展名加以识别, 具体如下所示。文件名中可使用的字符如下所示: 字母 a ~ z、A ~ Z; 数字 0 ~ 9; 特殊字符 !、&、\$、#、'、{、}、-、^、(、) 和 _。

文件名中不可使用下述字符: 、.、/、\、?、*、"、:、;、<、>、=、+、空格和双字节字符。

文件扩展名取决于存储文件的类型。

● 目录

CS/CJ 系列 CPU 单元可访问子目录中的文件。请在要存储文件的文件存储器中指定目录位置。除非使用手持式编程器, 否则一个目录最多可指定 5 个子目录 (包括根目录)。目录路径的最大长度为 65 个字符。当使用 Windows 等操作系统创建存储卡子目录时, 请勿超出最大子目录数 (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: 整个 DM 区 (D0 ~ D32767) (32,768 字 × 2)+48=65,584 字节
数据文件 (.TXT 或 .CSV)	文件大小取决于分隔符和回车使用数。分隔符代码和回车代码分别占用一个字节和两个字节。 例 1: 既无分隔符又无回车的字 123456789ABCDEF012345678 占用 24 个字节。 例 2: 带分隔符单字, 且每两个字段带一个回车 1234,5678↵ 9ABC, DEF0↵ 1234,5678 ↵ 占用 33 个字节。 例 3: 带分隔符双字, 且每两个字段带一个回车 56781234, DEF01234↵ 56781234 ↵ 占用 29 个字节。
程序文件 (.OBJ)	(使用步数 × 4)+48 字节 *1
参数文件 (.STD)	16,048 字节

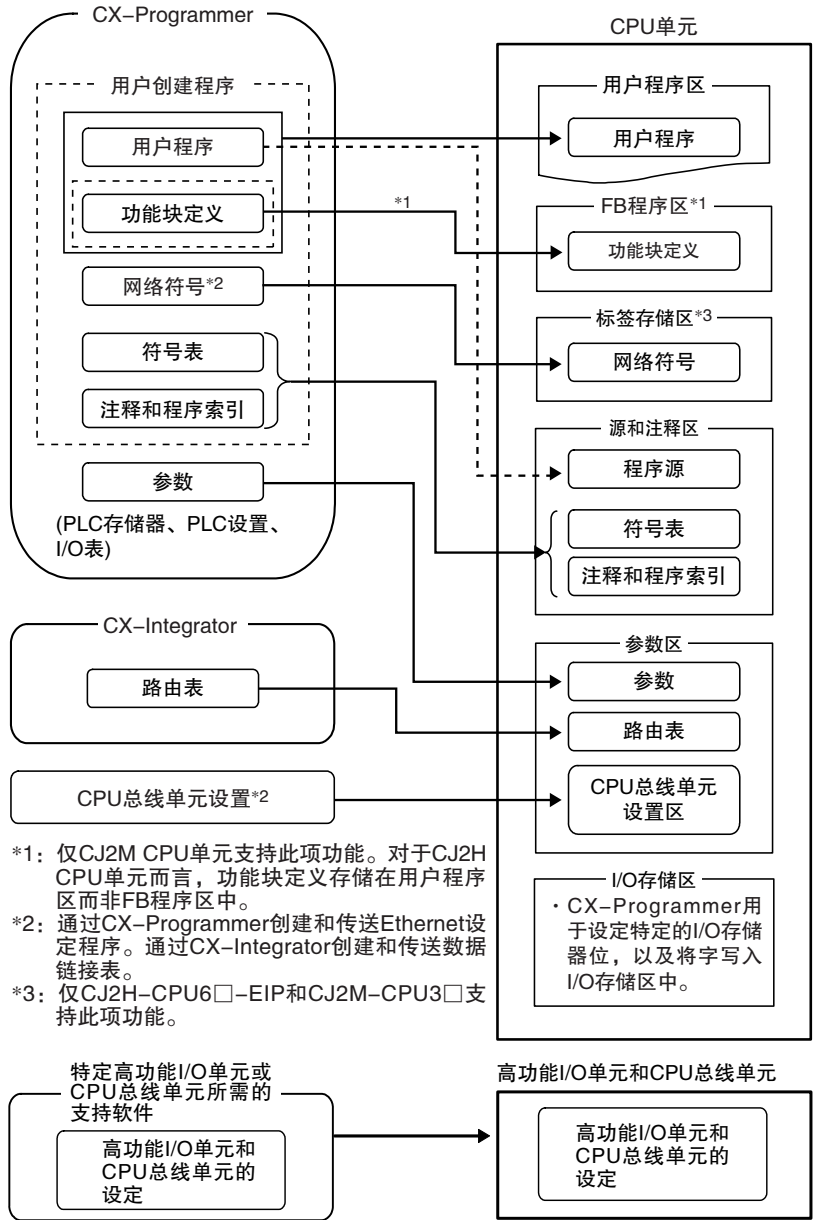
*1 将总 UM 步数减去可用的 UM 步数即可计算出程序文件中的步数, 在 CX-Programmer 的交互参考报告中这类值可显示。详情请参考《CX-Programmer 操作手册》。

7-3-5 支持软件和文件存储器文件之间的关系

下表所示为各个支持软件包文件与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	存储区
CX-Programmer	CX-Programmer项目文件(扩展名: .CXP)	用户程序 / 网络符号 功能块	程序 / 网络符号文件 (扩展名: OBJ)	CPU 单元
		符号表(包括符号名称、对应地址和 I/O 注释) 注: 网络符号除外。	符号表(网络符号除外) (SYMBOLS.SYM)	
		注释(I/O注释的行注释和标注除外)	注释文件(COMMENTS.CMT)	
		PLC 名称	参数文件(扩展名: STD)*1	
		PLC 设置		
		已注册的 I/O 表		
	Ethernet 单元设定			
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存储器和分配的DM区字) 分配的I/O存储器和分配的DM区字	数据文件(扩展名: IOM)		
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	指定文件(.csv)	FL-net 设定		
特定高功能I/O单元或CPU总线单元所需的支持软件	文件由高功能I/O单元和CPU总线单元的特定支持软件进行创建	非上述文件所包含的数据	单元备份文件(BACKUP□□.PRM)	高功能I/O单元和CPU总线单元

*1 在文件存储器中, 利用支持软件应用程序创建的文件将采用单一文件形式进行创建。



8

I/O 分配与单元设定

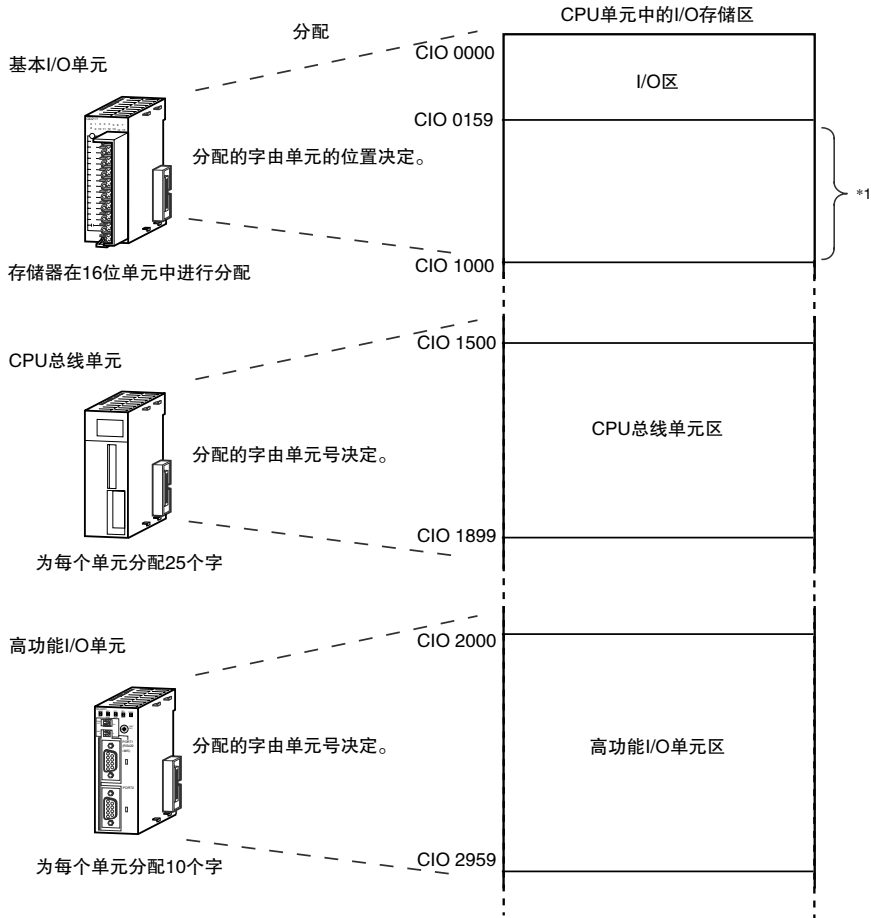
本节描述了如何将存储器分配至基本 I/O 单元、高性能 I/O 单元和 CPU 总线单元，以及如何设定高性能 I/O 单元和 CPU 总线单元。

8-1 I/O 分配	8-2
8-1-1 I/O 分配	8-2
8-1-2 自动分配	8-4
8-1-3 手动分配	8-8
8-1-4 I/O 表错误和注意事项	8-16
8-2 设定 CPU 总线单元和高性能 I/O 单元	8-18
8-2-1 设定参数	8-18
8-2-2 数据交换	8-22

8-1 I/O 分配

8-1-1 I/O 分配

将字分配到 CPU 单元的 I/O 存储器中，用于在 CPU 单元和其它单元之间交换数据，该过程称为 I/O 分配。为基本 I/O 单元、高性能 I/O 单元和 CPU 总线单元分配不同的存储器。



*1 可从CX-Programmer将每个机架上的首字设定到CIO 0000 ~ CIO 0999之间的地址 (默认设定: 从CIO 0000开始连续分配)。

I/O 分配方法

使用下述两种方法之一来分配 I/O。

- 自动分配 (默认)
当接通单元的电源时, 将根据相连单元的状态 (例如插槽位置和单元号) 来分配 I/O。
- 手动分配 (创建 I/O 表)
可通过使用 CX-Programmer 创建 I/O 表的方法来指定所需分配。创建了 I/O 表时, 如果在接通 CPU 单元的电源时 I/O 表与相连单元的状态不相符, 则将出错。

下表列出了这两种方法的差异。根据目的和系统的需要来选择方法。

项目	分配方法	手动分配	
		通过相连的单元来创建 I/O 表	通过用户设定的分配来创建 I/O 表
目的	<ul style="list-style-type: none"> • 当不需要检查是否存在不正确的单元时可使用该方法。 • 当不要求用户设定的分配时 (包括保留未使用的字), 可使用该方法。 	使用该方法来检查是否存在不正确的单元。	<ul style="list-style-type: none"> • 使用该方法来创建用户设定的分配 (包括保留未使用的字)。 • 使用该方法在无实际的 I/O 单元时 (即脱机) 分配 I/O。
创建 I/O 表	不需要	需要	
对照 I/O 表来验证所连接的单元	不验证单元	验证单元 (如果所连接的单元与 I/O 表不一致, 则运行无法启动。)	
用于分配 I/O 的操作	自动分配	手动分配	
CX-Programmer	不需要	需要	
所连接的单元是否需要创建 I/O 表	需要	需要	不需要
分配方法	自动分配 (从 CPU 单元开始按顺序分配)		手动分配 (可为各组指定首地址。)

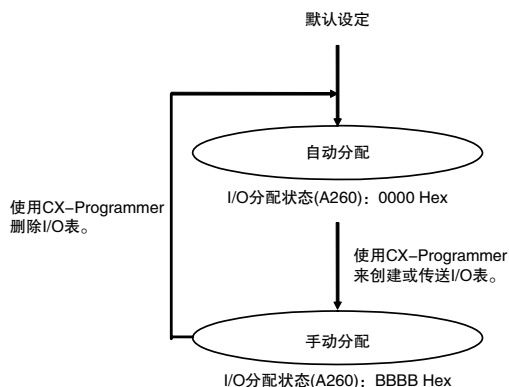
● 检查 I/O 分配方法

可使用 I/O 分配状态字 (A260) 来检查是手动还是自动设定了 I/O 分配。

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

I/O 分配方法可按如下方式改变。

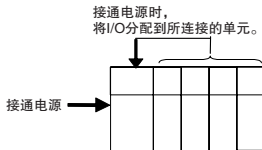


当使用 CX-Programmer 删除 I/O 表时, 每个机架的首字将在 I/O 分配状态返回至自动分配状态的同时被清除。此外, CPU 单元系统设置区将被初始化。

8-1-2 自动分配

自动分配 (默认)

使用该方法，根据所连接单元的状态来分配 I/O。对 I/O 进行自动分配，因此不需要从编程设备进行操作。若使用自动分配，则在每次接通 PLC 的电源时均会更新 I/O 分配。若要从手动分配变为自动分配，可使用编程设备删除 I/O 表。

**附加信息**

与手动分配不同，不通过自动分配来创建 I/O 表，因此不对所连接的单元进行验证。

分配方法

采用自动分配时，将对所连接的单元进行 I/O 分配，如下表中所述。无需用户操作。

单元类别	分配方法
基本 I/O 单元	根据位置按顺序进行分配。
高功能 I/O 单元	根据前面板旋转开关的设定 (即单元号) 进行分配。
CPU 总线单元	

针对各单元分类的 I/O 分配方法

本节描述了针对各单元分类 (基本 I/O 单元、高性能 I/O 单元和 CPU 总线单元) 的 I/O 分配方法。

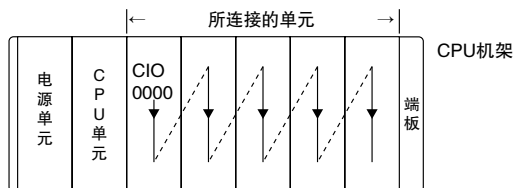
● 对基本 I/O 单元的 I/O 分配

将 I/O 区中的字分配给 CJ 系列基本 I/O 单元 (CIO 0000 ~ CIO 0159)。这些单元可连接至 CPU 机架或扩展机架。使用下述方法。

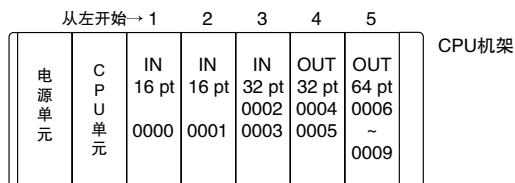
CPU 机架中的基本 I/O 单元

将从离 CPU 单元最近的单元开始的字从左到右分配给 CPU 机架上的基本 I/O 单元。为各单元分配各自所需的字数。

注 为带有 1 ~ 16 点 I/O 的单元分配 16 个位, 为带有 17 ~ 32 点 I/O 的单元分配 32 个位。



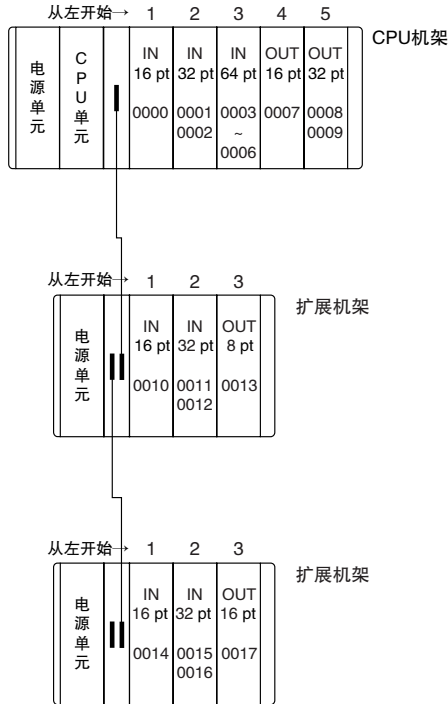
例



扩展机架中的基本 I/O 单元

对从 CPU 机架到直接连接到 CPU 机架的扩展机架对基本 I/O 单元连续分配 I/O。字从左到右进行分配，并且对各单元分配各自所需的字数，正如 CPU 机架中的单元一样。

例



● **对高功能 I/O 单元的 I/O 分配**

根据单元中设定的单元号，对各 CJ 系列高功能 I/O 单元分配高功能 I/O 单元区 (CIO 2000 ~ CIO 2959) 中的 10 个字。可将高功能 I/O 单元安装到 CPU 机架或扩展机架。下表列出了分配给各单元的高功能 I/O 单元区中的字。

单元号	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

对基本 I/O 单元进行 I/O 分配期间，将忽略高功能 I/O 单元。包含高功能 I/O 单元的位置不分配 I/O 区中的任何字。

● 对 CPU 总线单元的 I/O 分配

根据单元中设定的单元号, 对各 CJ 系列 CPU 总线单元分配 CPU 总线单元区 (CIO 1500 ~ CIO 1899) 中的 25 个字。可将 CPU 总线单元安装到 CPU 机架或扩展机架。下表列出了分配给各单元的 CPU 总线单元区中的字。

单元号	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

对基本 I/O 单元进行 I/O 分配期间, 将忽略 CPU 总线单元。包含 CPU 总线单元的位置不分配 I/O 区中的任何字。CPU 总线单元的单元号与高功能 I/O 单元的单元号不同。对于 CPU 总线单元和高功能 I/O 单元使用同一个单元号不会导致重复使用单元号。

例

	0	1	2	3	4
电源单元	IN 16 pt 0000	高功能 I/O 单元 2000 ~ 2009	CPU 总线 单元 1500 ~ 1524	OUT 16 pt 0001	CPU 总线 单元 1525 ~ 1549

插槽	单元	型号	需要的字数	分配的字	单元号	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	高功能 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 点转换输出单元	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- EIP	25	CIO 1500 ~ CIO 1524	0	CPU 单元 (将该端口 作为 CPU 总线单元。)

*1 对于 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 单元, 将作为 CPU 总线单元为内置的 EtherNet/IP 端口分配字。这些字用于存储该端口的网络通信状态。与对 CPU 总线单元的处理相同, 将根据通过 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 的前面板上的旋转开关设定的单元号来分配字。

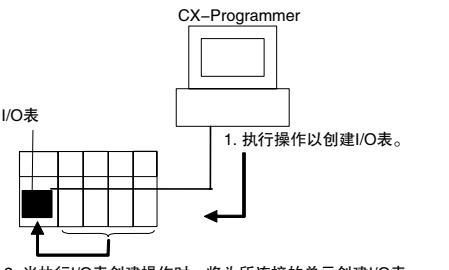
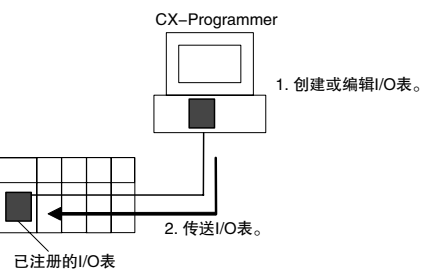
8-1-3 手动分配

手动分配

使用手动分配，根据用户创建的 I/O 表来分配 I/O。如果创建了 I/O 表，则在接通电源时，将对照 I/O 表对所连接的单元进行验证。

● 创建 I/O 表

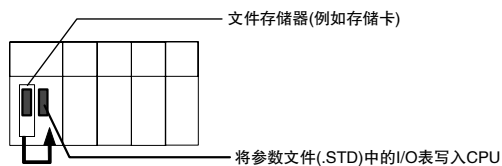
使用下述方法之一来创建 I/O 表。

通过相连的单元来创建 I/O 表	通过用户设定的分配来创建 I/O 表
<p>通过所连接的单元以及联机的 CX-Programmer 来创建 I/O 表。</p> 	<p>在 CX-Programmer 脱机时，使用 CX-Programmer 根据需要创建 I/O 表，然后将 I/O 表传送至 CPU 单元。</p> 



附加信息

可将 PLC 设定为当接通电源时将保存在存储卡中的 I/O 表自动传送至 CPU 单元。



● 单元验证

如果创建了 I/O 表，则在启动时将注册的 I/O 表与实际 I/O 进行对比。如果二者不匹配，则 A401.10 将变 ON，以表示存在 I/O 设定错误，并且无法运行。

通过相连的单元来创建 I/O 表

● 分配方法

采用与自动分配相同的方法来分配 I/O。有关分配方法的详情，请参考“8-1-2 自动分配”。

● I/O 分配步骤

在 CX-Programmer 联机的情况下，采用下述步骤来为相连的单元创建 I/O 表。

1. 在主窗口项目树形图上双击“IO Table” (I/O 表) 标签，将显示“IO Table” (I/O 表) 窗口。
2. 选择“Options” (选项)，然后选择“Create” (创建)。安装至机架的单元的型号和位置将作为注册的 I/O 表写入 CPU 单元。

● 比较 I/O 表的步骤

在 CX-Programmer 联机的情况下，采用下述步骤将 I/O 表与实际相连的单元进行比较。

1. 在主窗口项目树形图上双击“IO Table” (I/O 表) 标签，将显示“IO Table” (I/O 表) 窗口。
2. 选择“Options” (选项)，然后选择“Verify” (验证)。将安装至机架的单元的型号和位置与注册的 I/O 表进行比较。显示比较结果时，查看这些结果。

通过用户指定的分配来创建 I/O 表

● 分配方法

除了为所连接的单元自动分配 I/O 之外，还可采用下述方法以创建所需 I/O 表的形式来手动分配 I/O。

- 将字分配给机架上任意位置处的单元
- 保留字
- 指定机架上的首字
- 无实际单元的情况下分配 I/O

字按如下方式进行分配：

单元类别	分配方法
基本 I/O 单元	存在一些设定方面的限制，但不包括下述情况： 可指定分配至任意机架上的任意单元的字。将连续字按下述各单元连接的顺序分配至这些单元。可执行该操作以创建 63 个用户设定的分配组。
高性能 I/O 单元	根据前面板旋转开关的设定 (即单元号) 来分配字。
CPU 总线单元	
脉冲 I/O 模块	离 CPU 单元最近的脉冲 I/O 模块称为“脉冲 I/O 模块 0”，另一个则称为“脉冲 I/O 模块 1”。

● I/O 分配步骤

使用 CX-Programmer 创建 I/O 表，然后将 I/O 表传送至 PLC。脱机创建 I/O 表，然后将 I/O 表联机传送至 PLC。

1. 在主窗口项目树形图上双击 “IO Table” (IO 表) 标签，将显示 “I/O Table” (I/O 表) 窗口。
2. 双击需要编辑的机架。将显示所选机架的插槽。
3. 右键点击要指定到某个单元的插槽，然后从下拉菜单中选择单元。
4. 编辑 I/O 表之后，通过选择 “Options – Transfer to PLC” (选项 – 传送至 PLC) 将这些表传送到 CPU 单元。

● 为机架分配首字并保留未使用的字

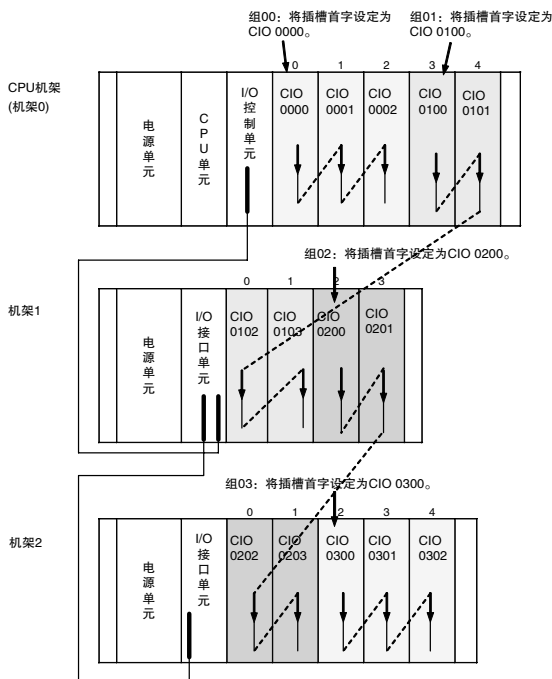
可使用 CX-Programmer 来为任意机架中任意位置处的单元设定首字。此后，将指定的首字之后的连续字按单元连接的顺序分配至下一单元。通过为各组中的首个单元设定首字，可以设定 63 个组。

分配方法

为机架位置 (称为插槽) 设定首字时，必须为 CPU 机架上的插槽 00 设定首字。随后即可为任意机架上的任意插槽 (最多 63 个其它插槽) 设定首字。

- 为某个插槽设定的首字将创建一个以该插槽开始的组。
- 将从指定字开始的字分配至组中的第一个单元，然后按从左到右的顺序将连续的字继续分配至各单元，直至下一个组为止 (即：直至设定了插槽首字的下一个单元为止)。下一个组可从同一个机架或下一个机架开始。
- 对于组 00，为 CPU 机架的插槽 00 设定首字。对于组 01 ~ 63，可为任意机架上的任意插槽设定首字。

例如，在各机架的中间位置设定插槽首字。仅使用 16 点单元。



插槽首字设定

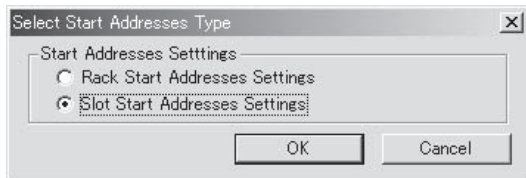
组	机架	插槽	字
00 *	CPU 机架	00	CIO 0000
01	CPU 机架	03	CIO 0100
02	机架 1	02	CIO 0200
03	机架 2	02	CIO 0300

* 组 00 必须在 CPU 机架上的插槽 00 处开始。可设定任意字。可为组 01 ~ 63 设定任意机架上的任意插槽。

从 CX-Programmer 设定插槽首字

使用下述步骤来设定机架首字。

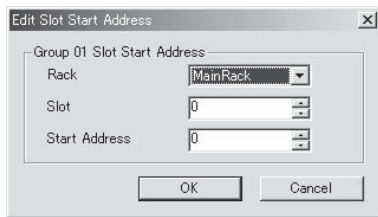
1. 从 “I/O Table” (I/O 表) 窗口的 “Option” (选项) 菜单中选择 “Rack/Slot Start Addresses” (机架 / 插槽起始地址)。此时将显示以下对话框。
2. 选择 “Slot Start Addresses Settings” (插槽起始地址设定) 选项并单击 “OK” (确定) 按钮。



3. 在显示的对话框中，为 CPU 机架上的插槽 00 设定首字。



4. 若要修改原本的 CIO 0000 设定，请点击“OK”（确定）按钮。将出现如下对话框。



5. 选择所需的字并单击“OK”（确定）按钮。
6. 若要为其它组设定插槽首字，请单击“Add”（添加）按钮并对机架、插槽和字进行合适的设定。

设定	设定范围	默认值	说明
组	00 ~ 63	00	组成员按照组显示和设定的顺序自动进行分配。
机架	CPU 机架（“主机架”）的机架 1 ~ 7	CPU 机架	组00必须始终从CPU机架上的插槽00处开始。
插槽	00 ~ 03	0	
首字	0 ~ 999	0	---

将首字分配至机架

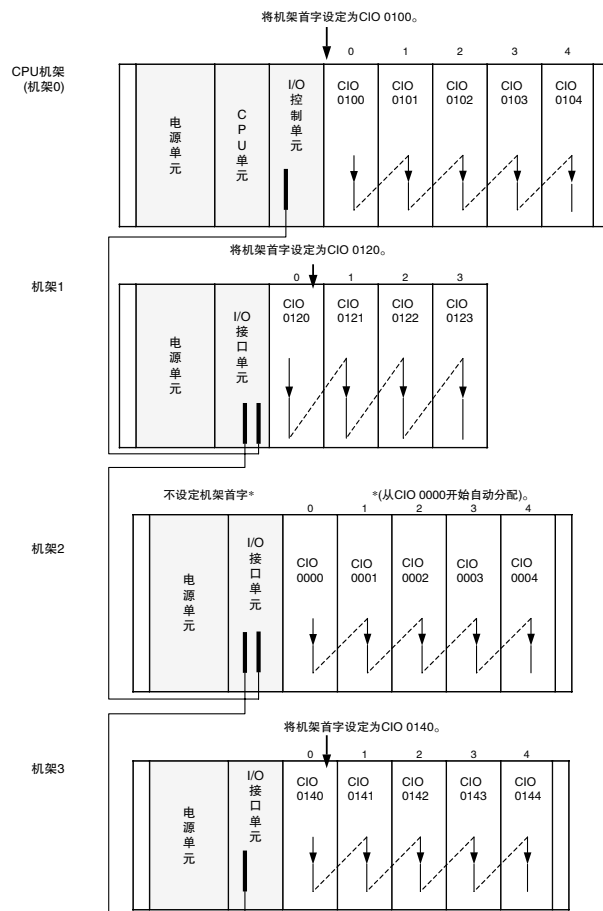
在 CJ 系列 PLC 中，可通过 CX-Programmers 的 I/O 表编辑操作来设定分配至各机架的首字。例如，可将 CPU 机架设定为分配从 CIO 0000 开始的字；为下一个机架分配从 CIO 0100 开始的字；为再下一个机架分配从 CIO 0200 开始的字；其余以此类推。该操作便于检查分配至各单元的字，而无需顺着 CPU 机架进行计算。

● 字分配

对于已为其设定首字地址的机架，按照各单元安装的顺序（从左到右）从指定的首字开始将字分配至单元。不会将字分配至空插槽。

对于未设定首字地址的机架，字按照机架号的顺序（从低到高）从分配至上一个机架的最后一个字开始进行分配，并对未设定首字的首个机架从 CIO 0000 开始分配。

例：为机架设定首字



机架首字设定

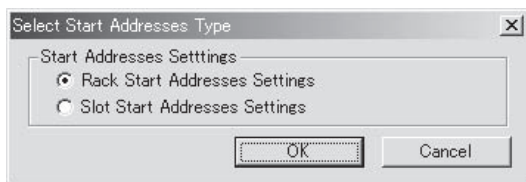
机架	首字
CPU 机架	CIO 0100
机架 1	CIO 0120
机架 2	0000
机架 3	0140

注 机架号 (0 ~ 3) 根据用缆线实际连接的机架顺序而固定。CPU 机架始终为机架 0, 而其它机架按顺序分别为机架 1 ~ 3。这些机架号无法改变。

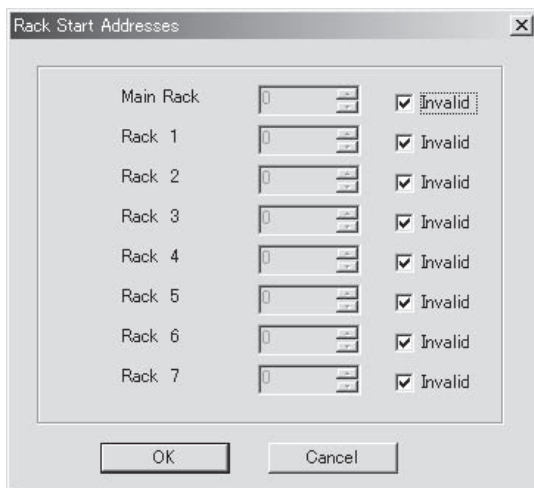
从 CX-Programmer 设定机架首字

使用下述步骤来设定机架首字。

1. 从 “I/O Table” (I/O 表) 窗口的 “Option” (选项) 菜单中选择 “Rack/Slot Start Addresses” (机架 / 插槽起始地址)。此时将显示以下对话框。



2. 选择 “Rack Start Addresses Settings” (机架起始地址设定) 选项并单击 “OK” (确定) 按钮。
3. 在显示的对话框中, 去掉对禁止机架首字设定的勾选, 并为 CPU 机架和扩展机架 (1 ~ 7) 设定首字地址。



设定	设定范围	默认值	说明
机架起始地址	0 ~ 900	0	对所有机架相同
无效	选择或清除选择	选择 (无效)	

注 1 可为任意型号的 CPU 单元最多设定 3 个机架。

2 尽管 CX-Programmer 窗口中将显示 7 个机架, 但只能为 CJ2 CPU 单元设定 3 个机架。



正确使用注意事项

- 设定首字时，请务必注意勿使分配的字重叠。机架的首字可设定为 CIO 0000 ~ CIO 0900 的任意地址。如果为两个机架分配了同一个字，或者如果将两个机架的首字设定为 CIO 0901 或更大的地址，则无法创建 I/O 表且扩展机架号重复标志 (A409.00 ~ A409.03: 机架号 0 ~ 3) 将变为 ON。重复错误标志 (致命错误) (A401.13) 也将变为 ON。
- 请务必在为机架设定首字分配之后再注册 I/O 表。若要注册 I/O 表，可在 CX-Programmer 的 “I/O Table” (I/O 表) 窗口中选择 “Options” (选项) 然后选择 “Create” (创建)。I/O 表注册操作用于注册分配至机架的 I/O 字。
- 如果在注册 I/O 表之后改变了实际系统配置，从而导致字数或单元型号与 I/O 表中的设定不匹配时，将产生 I/O 验证错误 (A402.09) 或 I/O 设定错误 (A401.10)。可能会同时产生 CPU 总线单元设定错误 (A402.03) 或高功能 I/O 单元设定错误 (A402.02)。如果拆下了某个单元，则可使用 I/O 表变更操作为缺失的单元保留字。如果变更或添加了某个单元，则程序中该单元分配字之后的所有字均将改变，并且必须再次执行 I/O 表注册操作。
- 若使用 CX-Programmer 删除了 I/O 表，则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

当已执行创建、编辑或传送 I/O 表 (即: 使用用户设定的 I/O 分配) 的步骤时, 在以下情况下无法创建 I/O 表。

问题	错误	描述	改正
所连接的单元与 I/O 表之间的验证错误	I/O 设定错误 (致命错误)	如果存在验证错误, 即如果注册的 I/O 表与实际连接到基本系统 (即 CPU 机架和扩展机架) 中的单元的类型与位置不相符, 则 I/O 设定出错标志 (致命错误)(A401.10) 将变 ON。 此时将无法运行。 CPU 单元前面板上的 RUN 指示灯将熄灭, 且 ERR/ALARM 指示灯将以红色闪烁。	<ul style="list-style-type: none"> 如果单元号不正确, 请关闭电源并正确连接相应的单元。 如果单元号正确, 则可通过在 CX-Programmer 中比较 I/O 表来确认单元设定的矛盾之处, 然后关闭电源并纠正单元连接。 如果在 I/O 表中存在错误, 则应重新创建 I/O 表或使用 CX-Programmer 来编辑 I/O 表, 以纠正错误。
为高性能 I/O 单元或 CPU 总线单元两次设定了同一个单元号。	重复错误 (致命错误)	在下述任一情况下, 重复错误标志 (A401.13) 将变为 ON。 <ul style="list-style-type: none"> 为一个以上的 CPU 总线单元设定了同一个单元号。 为一个以上的高性能 I/O 单元设定了同一个单元号。 为一个以上的基本 I/O 单元分配了同一个字。 为一个以上的扩展机架设定了同一个机架号。 此时将无法运行。 CPU 单元前面板上的 RUN 指示灯将熄灭, 且 ERR/ALARM 指示灯将以红色闪烁。	<p>检查 CPU 总线单元或高性能 I/O 单元的单元号, 消除重复的编号, 然后关闭机架的电源并再次接通。</p> <p>检查 A409.00 ~ A409.03 中的位为 ON 的机架号上的单元的分配情况。纠正分配, 使得对相同的字不进行重复分配 (包括分配至其它机架上的单元), 然后关闭机架的电源并再次接通。</p> <p>检查 A409.00 ~ A409.03 所表示的机架中的首字设定, 并用 CX-Programmer 将设定改为 CIO 0900 以下的有效字地址。</p>

此外, 当 I/O 点数过多标志 (致命错误)(A401.11) 为 ON 或 I/O 总线出错标志 (致命错误)(A401.14) 为 ON 时, 无法使用 I/O 表。

● 有关 I/O 表创建错误的详细信息

当从 CX-Programmer 创建 I/O 表时只要发生错误, 则 A261 中的内容 (I/O 表创建错误详情) 将提供有关出错单元的信息。该信息将便于通过故障诊断 I/O 表来查找出错的单元。

名称	地址		内容	
	字	位		
有关 I/O 表创建错误的详细信息	CPU 总线单元设置区初始化出错标志	A261	00	ON: CPU 总线单元设置中的错误 当正常生成 I/O 表时变为 OFF。
	I/O 溢出标志		02	ON: 最大 I/O 点数发生溢出。 当正常生成 I/O 表时变为 OFF。
	重复错误标志		03	ON: 同一个单元号使用了一次以上。 当正常生成 I/O 表时变为 OFF。
	I/O 总线出错标志		04	ON: I/O 总线错误 当正常生成 I/O 表时变为 OFF。
	高性能 I/O 单元出错标志		07	ON: 高性能 I/O 单元中出错 当正常生成 I/O 表时变为 OFF。
	I/O 未确认出错标志		09	ON: I/O 检测尚未完成。 当正常生成 I/O 表时变为 OFF。

有关设定 I/O 表的注意事项

设定 I/O 表之后，如需改变单元，请检查是否存在任何重复分配字的现象。但可以想象，在注册了 I/O 表之后可能会发生重复分配字的现象，例如用双字单元替换单字单元之后可能会发生该现象。在这种情况下，新单元额外所需的字仍将分配至下一个单元。

当改变单元后接通 PLC 的电源时，CPU 单元将对照 PLC 中所连接的实际单元来检查注册的 I/O 表。若存在任何重复现象，则将产生错误且无法继续编辑 I/O 表。发生该现象时，必须编辑 I/O 表并再次传送这些表。

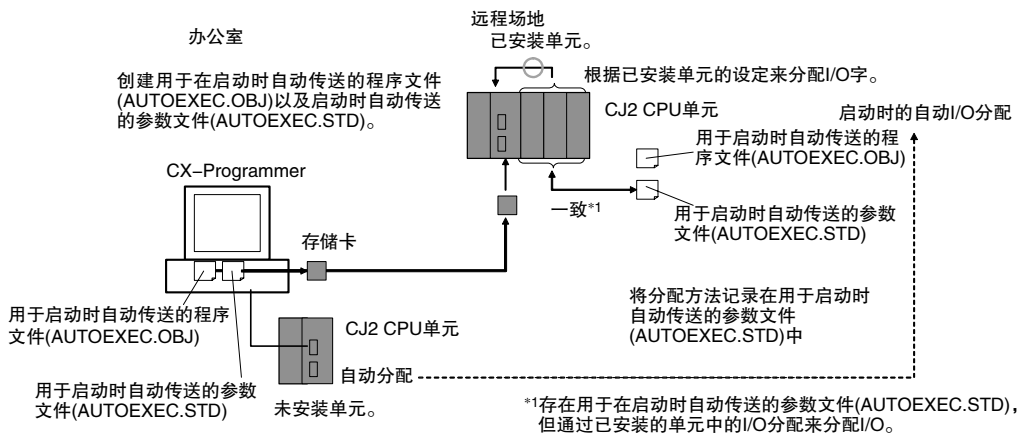
编辑了 I/O 表之后，CX-Programmer 将检查首字设定中是否存在重复现象，以及是否存在设定范围错误，从而帮助解决问题。

使用存储卡时的注意事项

用于创建 CPU 单元的 I/O 表的 I/O 分配方法（启动时自动分配 I/O 或者使用用户设定的 I/O 分配）记录在参数文件中，以便在启动时自动传送（AUTOEXEC.STD）。当从存储卡执行启动时自动传送的操作时，将自动检测所记录的方法并使用该方法在启动时自动分配 I/O 或验证 I/O 表。

下文说明了用于在 CPU 单元中创建 I/O 表的两种不同方法，这两种方法均在通电启动时通过创建参数文件来进行自动传送（AUTOEXEC.STD）。

- 自动分配*1
禁止使用存储卡中的用于在启动时自动传送的参数文件中的 I/O 表，并在启动时根据 PLC 中所连接的实际单元来自动分配 I/O。
- *1 采用启动时自动分配 I/O 的方法时，不事先创建 I/O 表，而是在每次接通电源时，对实际连接的基本 I/O 单元自动分配 I/O。
- 手动分配
允许使用存储卡中的用于在启动时自动传送的参数文件中的 I/O 表，且 CPU 单元对照 PLC 中实际连接的单元来验证 I/O 表。



8-2 设定 CPU 总线单元和高功能 I/O 单元

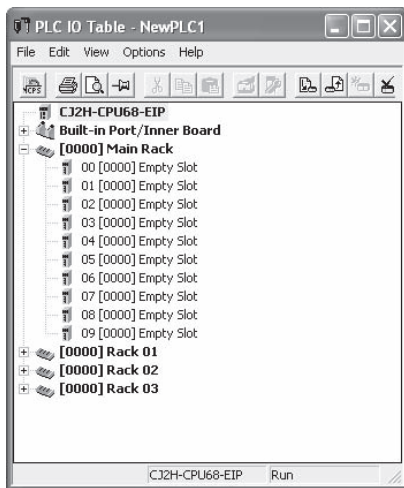
8-2-1 设定参数

CX-Programmer 可用于在分配至 CJ 系列 CPU 单元和高功能 I/O 单元的 DM 区和 CIO 区字中设定参数，并为 EtherNet 单元设定 CPU 总线单元设置。在 CX-Programmer 中使用 “I/O Table” (I/O 表) 窗口。

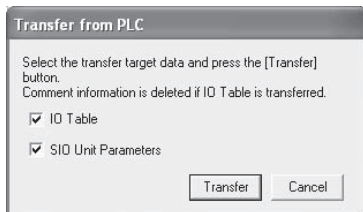
可在联机抑或脱机情况下进行设定。若在脱机情况下设定，则必须在联机时传送这些设定。

本节描述了在创建 I/O 表并将表从 CPU 单元传送到 “I/O Table” (I/O 表) 窗口之后，对 I/O 表进行编辑以及为 CPU 总线单元和高功能 I/O 单元进行设定的实例。操作步骤与脱机将 CPU 总线单元和高功能 I/O 单元注册到 I/O 表中并随后为 CPU 总线单元和高功能 I/O 单元编辑设定的步骤相同。

1. 将 CX-Programmer 联机并打开 “I/O Table” (I/O 表) 窗口。



2. 从 “I/O Table” (I/O 表) 窗口菜单中选择 “Options – Transfer from the PLC” (选项 – 从 PLC 传送)。将显示 “Transfer from the PLC” (从 PLC 传送) 对话框，如下图所示。

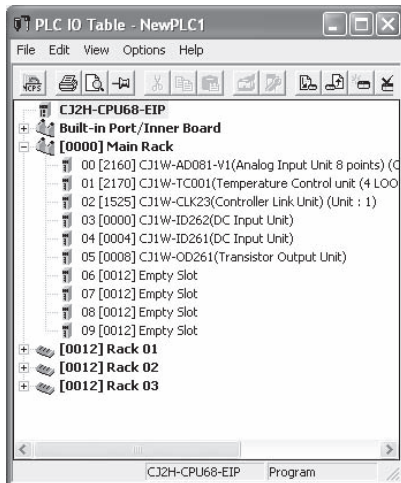


此时可选择 “I/O Table” (I/O 表) 和 “SIO Unit Parameters” (SIO 单元参数) 选项。

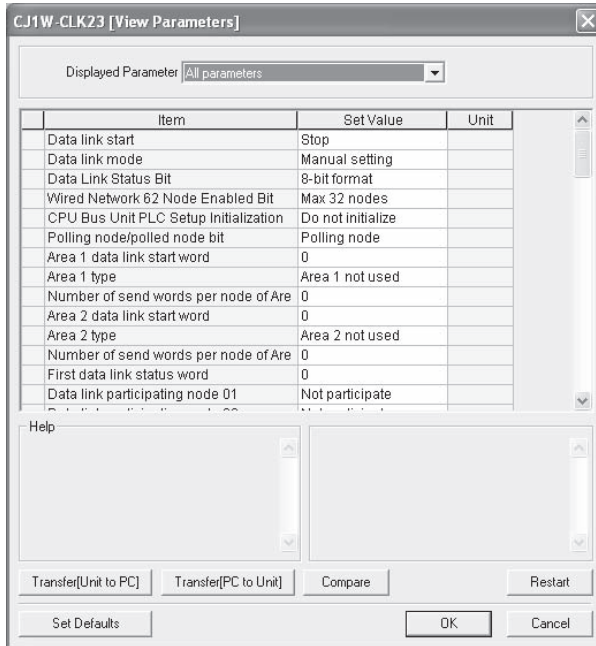
3. 为想要传送的数据选择选项 (勾选复选框), 然后单击 “Transfer” (传送) 按钮。该例中同时选择了两个选择。将从 PLC 传送 I/O 表和单元参数数据, 并将显示传送结果。



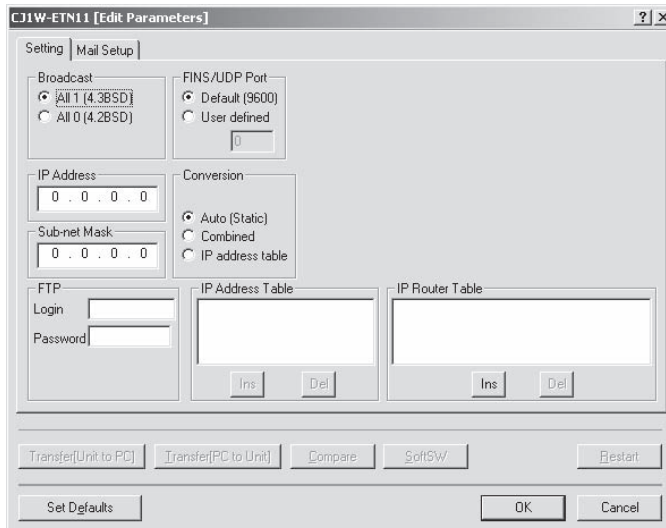
4. 检查传送结果址并单击 “OK” (确定) 按钮。将显示从 PLC 传送的 I/O 表。



5. 若要编辑高功能 I/O 单元或 CPU 总线单元的参数，可选择单元，然后双击该单元或者从菜单中选择“Edit – SIO Unit Parameters”（编辑 – SIO 单元参数）。将显示所选单元的“Edit Parameters”（编辑参数）对话框。
- 在这种情况下，可在已显示的 Controller Link 单元的“Edit Parameters”（编辑参数）对话框中编辑该单元的“CPU Bus Unit Allocation DM Settings”（CPU 总线单元分配 DM 设定）。

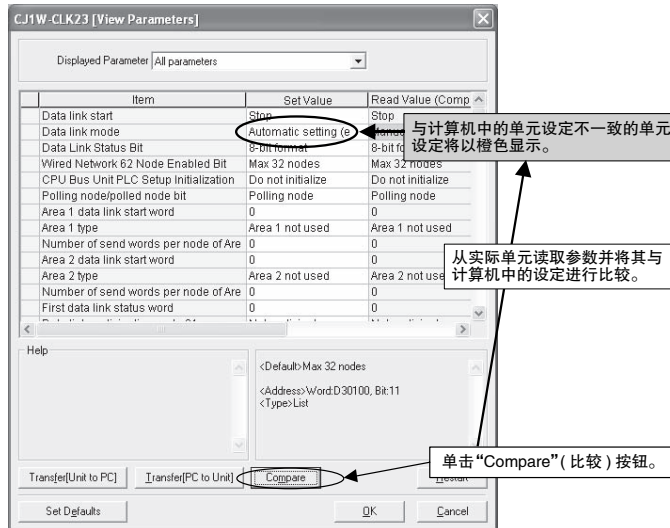


- 在这种情况下，可在已显示的 EtherNet 单元的“Edit Parameters”（编辑参数）对话框中编辑 DM 区分配字中的“CPU Bus Unit Settings”（CPU 总线单元设定）。



6. 编辑参数并根据需要单击“Compare”(比较)按钮。如果在 PLC 联机的情况下单击了“Compare”(比较)按钮,则软件将立即检查 I/O 表设定是否与 CPU 单元中的单元设定(例如 DM 区和 CIO 区分配字中的设定)一致。如果内容不一致,则从实际单元读取的设定将以橙色显示。

下列显示了对于 Controller Link 单元的设置进行比较的结果。



7. 如果想要将为各单元设定的参数下载至 CPU 单元,可单击“Download to Unit”(下载至单元)按钮。



正确使用注意事项

当在“I/O Table”(I/O表)窗口中编辑了高功能I/O单元或CPU总线单元设定时,若在“Edit Parameters”(编辑参数)对话框中单击了“Transfer PC to Unit”(从 PLC 传送至单元)按钮,则 DM 区或 CPU 总线单元设置区中为单元分配的参数仅传送至实际的 PLC。除非手动将这些参数设定为允许,否则不会自动设定为允许。单击“Reset”(复位)按钮,以将已传送的参数设定为允许。



附加信息

存储参数设定数据文件

- 在“Edit Parameters”(编辑参数)对话框中设定的数据可保存到扩展名为 xml 的参数设定文件中。若要保存参数设定,可在 I/O 表中右键点击所需的高功能 I/O 单元或 CPU 总线单元,然后从弹出菜单中选择“Save Parameters”(保存参数)。
- 也可从参数设定文件(扩展名为 xml)读取已保存的参数设定,但这些设定必须是同一种型号的高功能 I/O 单元或 CPU 总线单元的设定。

8-2-2 数据交换

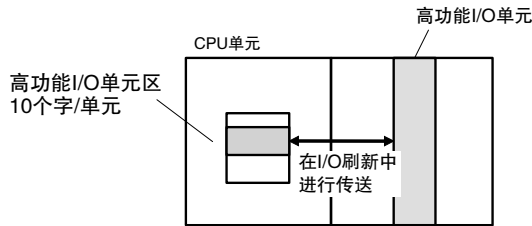
本节描述了如何在高性能 I/O 单元或 CPU 总线单元与 CPU 单元之间交换数据。

高性能 I/O 单元

高性能 I/O 单元区 (I/O 刷新)

每个循环中，在高性能 I/O 单元区的 I/O 刷新期间交换数据。通常情况下，根据各高性能 I/O 单元的单元号设定对单元分配 10 个字。详情请参考各高性能 I/O 单元的操作手册。

高性能 I/O 单元区的范围为 CIO 2000 ~ CIO 2959(10 个字 × 96 个单元)。



传送 DM 区分配字

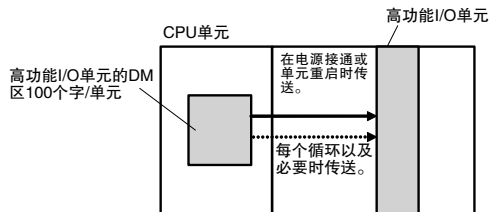
有三种情况可能通过分配至各单元的字来传送数据。数据传送的情况取决于所使用的型号。

- 接通 PLC 的电源时传送数据。
- 重启单元时传送数据。
- 必要时传送数据。

某些型号可双向传送数据，即从 DM 区传送至单元或者从单元传送至 DM 区。有关数据传送的详情，请参考相关单元的操作手册。

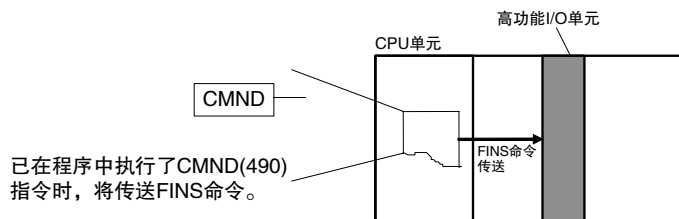
DM 区中的高性能 I/O 单元字：D20000 ~ D29599(100 个字 × 96 个单元)

将数据传送至 DM 区中的这些字，用作高性能 I/O 单元的初始设定。当从程序中改变这些分配字的内容从而改变系统时，必须对高性能 I/O 单元的重启位 (A502.00 ~ A507.15) 置 ON，以重启单元。

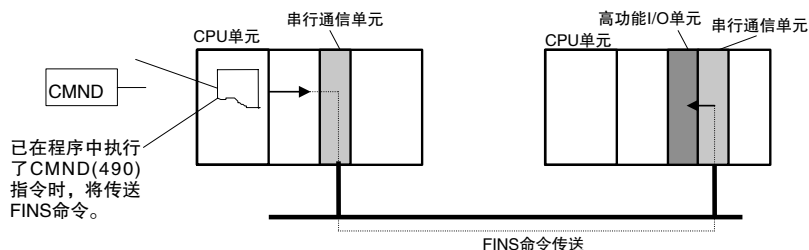


● FINS 命令

可将 CMND(490) 指令添加到梯形图程序中，从而向高性能 I/O 单元发送一条 FINS 命令。



注 可将 FINS 命令传送至网络中的其它 PLC、而不仅仅是本地 PLC 中的高性能 I/O 单元。



● 高性能 I/O 单元初始化

当接通 PLC 的电源或者对单元的重启位 (A502.00 ~ A507.15) 置 ON 时，将对高性能 I/O 单元进行初始化。正在对单元进行初始化时，单元的高性能 I/O 单元初始化标志 (A330.00 ~ A335.15) 将变 ON。

在其初始化标志为 ON 时，将不执行 I/O 刷新 (循环 I/O 刷新或者通过 IORF(097) 或 FIORF(225) 指令进行刷新)。

● 禁止高性能 I/O 单元的循环刷新

根据在各单元的前面板上所设定的单元号，对高性能 I/O 单元区 (CIO 2000 ~ CIO 2959) 中的各高性能 I/O 单元分配 10 个字。在每个循环的 I/O 刷新期间 (刚执行 END(001) 指令之后)，在 CPU 单元中对高性能 I/O 单元区中的数据刷新。

如果安装了大量高性能 I/O 单元，则 I/O 刷新可能会花费较长时间。如果 I/O 刷新花费较长时间，则可将 PLC 设置设定为禁止对特定的高性能 I/O 单元进行循环刷新。(高性能 I/O 单元循环刷新禁止位的地址在 PLC 设置中设定为 226 ~ 231。)

如果 I/O 刷新时间过短，则单元的内部处理将无法跟上节奏，因此高性能 I/O 单元出错标志 (A402.06) 将变为 ON，且高性能 I/O 单元可能无法正常运行。在这种情况下，可通过在 PLC 设置中设定一个最小循环时间来延迟循环时间，否则会禁止高性能 I/O 单元的循环 I/O 刷新。

禁止了循环刷新时，可在程序执行期间通过 IORF(097) 或 FIORF(225) 指令来刷新高性能 I/O 单元的数据。



正确使用注意事项

可从中断任务对高性能 I/O 单元执行 IORF(097)、FIORF(225)、IORD(222) 和 IOWR(223) 指令。执行这些指令时，请务必在 PLC 设置中设定为禁止高性能 I/O 单元的循环刷新。如果不禁止循环刷新，并且在中断任务中执行了下述任一步骤，则将产生非致命错误且重复刷新出错标志 (A402.13) 将置 ON。

- 使用 IORF(097) 或 FIORF(225) 指令可对同一个高性能 I/O 单元执行 I/O 刷新。
- 将使用 IORD(222) 或 IOWR(223) 指令对同一个高性能 I/O 单元执行针对存储区的数据读/写操作。

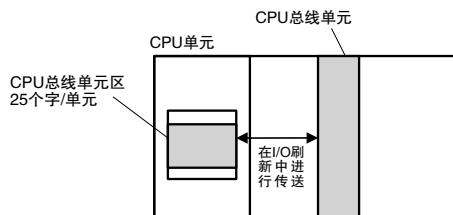
只要禁止了高性能 I/O 单元的循环刷新时，均务必确保在程序中使用 IORF(097) 或 FIORF(225) 指令在运行期间至少每 11 秒钟对该单元进行一次 I/O 刷新。如果不每 11 秒钟刷新一次，则高性能 I/O 单元中将产生 CPU 单元服务监控错误。

CPU 总线单元

● CPU 总线单元区 (I/O 刷新)

每个循环中, 在 CPU 总线单元区的 I/O 刷新期间交换数据。通常情况下, 根据各 CPU 总线单元的单元号设定对单元分配 25 个字。CPU 总线单元实际使用的字数会有所不同。

高性能 I/O 单元区的范围为 CIO 1500 ~ CIO 1899(25 个字 × 16 个单元)。



注 可在梯形图程序中执行CPU总线I/O刷新指令(DLNK(226)), 以刷新分配至指定单元号的CPU总线单元的CIO区字。

● 传送 DM 区分配字

根据单元号为各单元分配 100 个字。

注 这些字并非用于所有型号的 CPU 总线单元。

CPU 总线单元的 DM 区字: D30000 ~ D31599(100 个字 × 16 个单元)

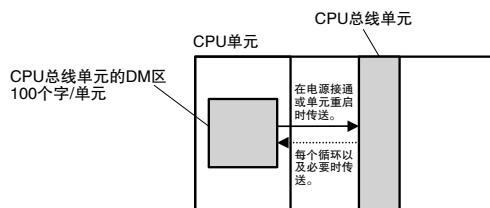
下述情况下将传送这些字的内容。实际传送的内容取决于单元的类型。

- 接通 PLC 的电源时传送数据。
- 各循环中传送数据。
- 必要时传送数据。

注 可在梯形图程序中执行CPU总线I/O刷新指令(DLNK(226)), 以刷新分配至指定单元号的CPU总线单元的DM区字。

某些型号可双向传送数据, 即从 DM 区传送至单元或者从单元传送至 DM 区。有关数据传送的详情, 请参考相关单元的操作手册。

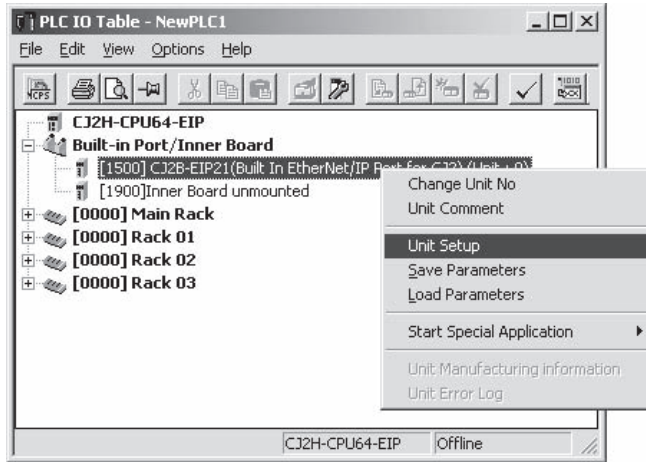
这 100 个字通常用于为 CPU 总线单元保持初始设定。当从程序中改变该区的内容从而在系统中反映变化时, 必须对受影响的单元的重启位 (A501.00 ~ A501.15) 置 ON, 以重启单元。



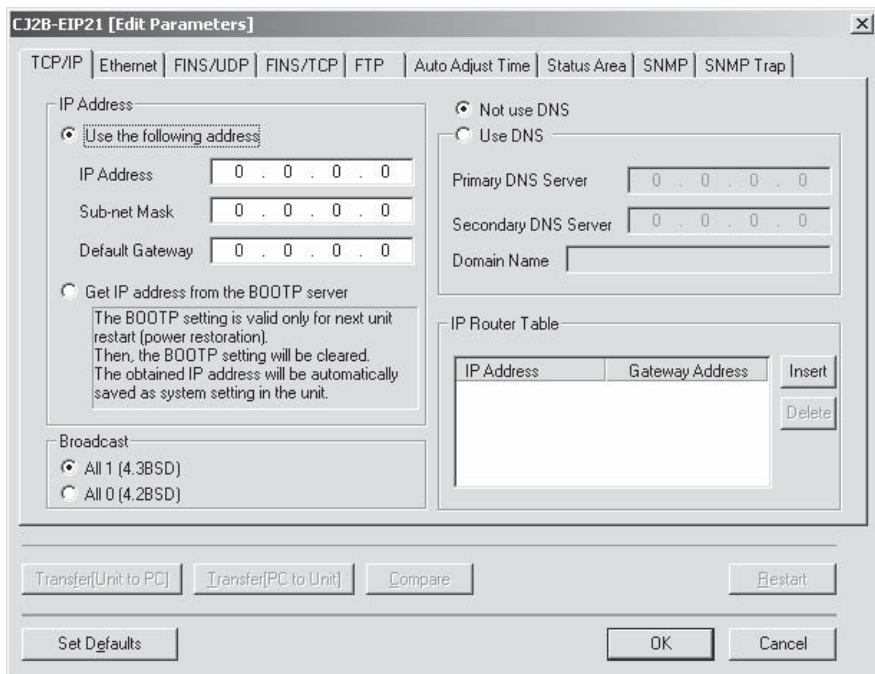
附加信息

根据旋转开关的设定情况, 将 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 单元的内置 EtherNet/IP 端口作为 CPU 总线单元处理。使用下述步骤来设定通信参数。

- (1) 在 PLC 的 I/O 表的 “Built-in Port/Inner Board” (内置端口 / 内部板) 中选择 CJ2B-EIP21 (对于 CJ2H 为内置的 EtherNet/IP 端口) 或 CJ2M-EIP21 (对于 CJ2M 为内置的 EtherNet/IP 端口)。点击右键并选择 “Unit Setup” (单元设置)。



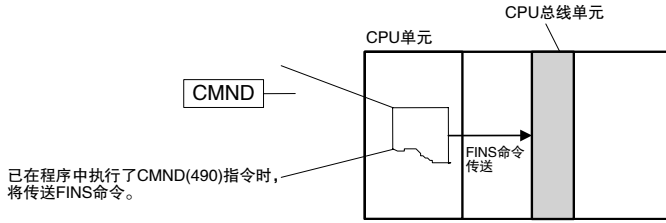
- (2) 对参数进行设定。



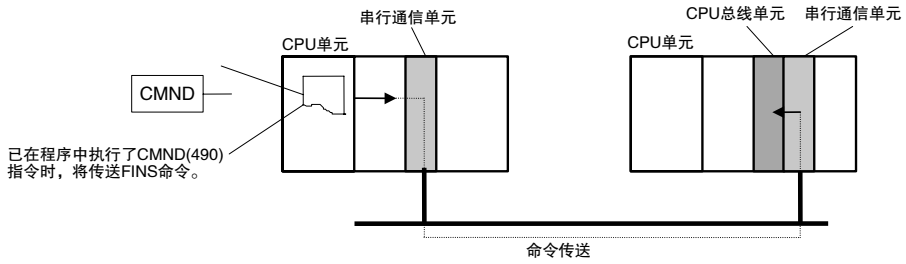
有关内置 EtherNet/IP 端口的参数说明，请参考《EtherNet/IP 单元操作手册》(样本编号: W465)。

● FINS 命令

可将 CMND(490) 指令添加到梯形图程序中，从而向 CPU 总线单元发布一条 FINS 命令。



注 可将 FINS 命令传送至网络中的其它 PLC、而不仅仅是本地 PLC 中的 CPU 总线单元。



● CPU 总线单元初始化

当接通 PLC 的电源或者对单元的重启位 (A501.00 ~ A501.15) 置 ON 时，将对 CPU 总线单元进行初始化。正在对单元进行初始化时，单元的 CPU 总线单元初始化标志 (A302.00 ~ A302.15) 将变 ON。当 CPU 单元的初始化标志为 ON 时，将不对该单元执行循环 I/O 刷新。

9

PLC 设置

本章描述了 PLC 设置中的参数，这些参数用于设定 PLC 运行的选项。

9-1 PLC 设置概述	9-2
9-2 PLC 设置中的设定	9-4
9-3 PLC 设置中的设定	9-5
9-3-1 启动运行设定	9-5
9-3-2 CPU 单元的设定	9-8
9-3-3 时间 / 同步设定	9-14
9-3-4 高性能 I/O 单元的循环刷新	9-19
9-3-5 基本 I/O 单元的机架响应时间	9-21
9-3-6 串行端口设定	9-23
9-3-7 外设服务	9-30
9-3-8 FINS 保护	9-31
9-3-9 I/O 模块	9-32

9-1 PLC 设置概述

PLC 设置包含 CPU 单元软件的基本参数设定，用户可修改这些设定以便自定义 PLC 运行。可从手持式编程器或其它编程设备来更改这些设定。请在下列情况下更改 PLC 设置。

应用	参数
下述情况下，必须更改 CJ 系列基本 I/O 单元的输入响应时间设定。 · 基本 I/O 单元中产生颤振或噪声。 · 正在为长于循环时间的间隔接收短脉冲输入。	基本 I/O 单元的机架响应时间
在接通 PLC 的电源时，必须保持 I/O 存储器的所有区域 (包括 CIO 区、工作区、定时器标志和当前值 (PV)、任务标志、变址寄存器以及数据寄存器) 中的数据。	启动时 IOM 保持位的状态
在接通 PLC 的电源时，必须保持从编程设备强制置位或强制复位的位的状态。 · 调试 PLC ; · 将启动模式切换至 PROGRAM 或 MONITOR 模式。	启动时强制状态保持位的状态 启动模式
无电池操作时，不需要电池电压过低错误检测。	检测电池低电压
禁止检测重复刷新操作。	检测重复刷新错误
RS-232C 端口无法以手持式编程器或 CX-Programmer(外设总线) 通信速度自动检测功能来使用，并且将不使用默认上位链接通信设定，例如 9,600bps。*1	RS-232C 端口设定
想要通过 PT 经由 NT 链接来提高通信速度。	将外设端口或 RS-232C 端口的通信端口波特率设定为高速 NT 链接。
想要以 1ms(或 0.1ms) 而非 10ms 为单位来设定定时中断功能的间隔。	定时中断时间单位
想要对 I/O 中断任务或定时中断任务使用高速中断功能。	允许高速中断功能
想要使用同步单元运行功能。	使用同步运转功能
调试时查找指令错误。	当指令出错时停止 CPU 运行
想要设定最短循环时间以创建恒定的 I/O 刷新循环。	最小循环时间
想要将最大循环时间设定为除 1 秒之外的值 (10ms ~ 40,000ms)。	监视循环时间
电源中断时执行特殊处理。	电源 OFF 中断任务 *2
想要延迟对电源中断的检测。 · 想要在中断任务中执行 IORF 或 FIORF 指令。 · 在使用大量高性能 I/O 单元时想要缩短平均循环时间。 · 想要延长高性能 I/O 单元的 I/O 刷新间隔。	电源 OFF 检测延时 *1 高性能 I/O 单元的循环刷新
不想在出错日志中记录 FAL(006) 和 FPD(269) 的用户自定义错误。	FAL 出错日志登记
想要减小因文本串处理导致的循环时间波动。 · 有些单元在接通电源时需要花时间启动。 · 但不希望这些单元完成启动过程以启动 CPU 单元的运行。	后台执行表数据、文本串和数据移位指令 执行设定

*1 若要改变 PLC 设置中的设定，则必须使 CPU 单元前面板上的 DIP 开关的第 5 位处于 OFF 状态。

*2 如果安装了 CJ1W-PD022，则无法使用这些设定。

● 辅助区中的相关位和字

名称	地址	描述	访问权限
PLC 设置出错标志 (非致命错误)	A402.10	当在 PLC 设置中发生设定错误时置 ON。	只读

9-2 PLC 设置中的设定

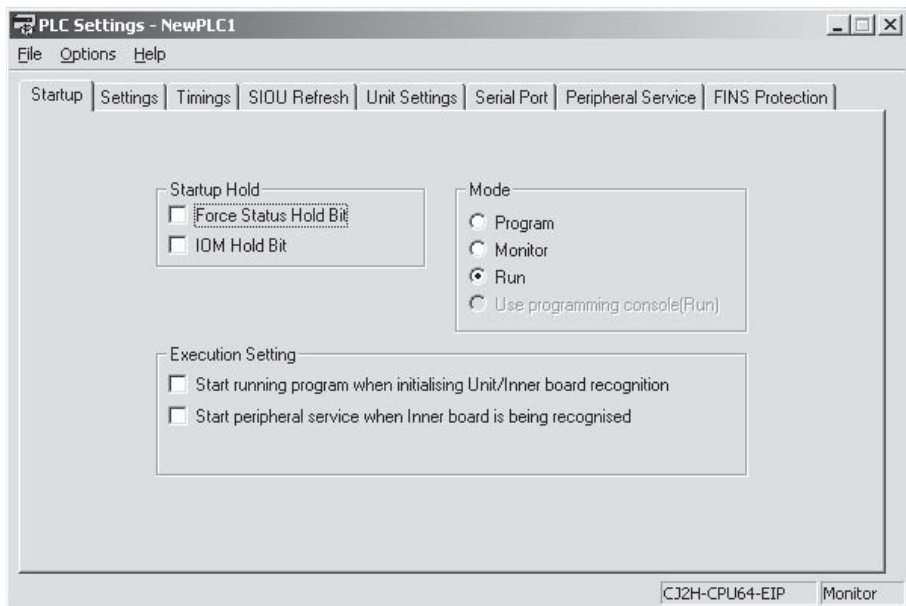
下表中列出了 PLC 设置中的默认设定。若要改变设定，可使用 CX-Programmer 来编辑 PLC 设置，然后将 PLC 设置传送到 CPU 单元。

CX-Programmer 的 PLC 设置 选项页	设定名称		适用 CPU 单元		默认值	页码
			CJ2H CPU 单元	CJ2M CPU 单元		
启动	启动保持设定	强制状态保持位	支持	支持	接通电源时不保持。	9-5
		IOM 保持位	支持	支持	接通电源时不保持。	9-6
	操作模式		支持	支持	RUN 模式	9-7
	执行设定	在初始化单元 / 内部板的识别状态之前启动程序运行	支持	支持	不启动。	9-7
CPU 单元的设定	执行进程设定	不检测电池低电压	支持	支持	检测	9-8
		检测重复刷新错误	支持	支持	检测	9-10
		当指令出错时停止 CPU 运行	支持	支持	不停止	9-10
		FAL 出错日志登记	支持	支持	注册到出错日志中。	9-11
	后台执行设定		支持	支持	不以后台执行。	9-12
	FB 中的通信指令设定	重试计数	支持	支持	0 次	9-13
响应超时		支持	支持	2s	9-13	
时间 / 同步设定	监视循环时间		支持	支持	1,000ms(1s)	9-14
	恒定循环时间 (最小循环时间)		支持	支持	不恒定	9-15
	定时中断间隔		支持	支持	10ms	9-16
	电源 OFF 检测时间		支持	支持	0ms	9-17
	电源 OFF 中断		支持	支持	不使用	9-18
	允许高速中断功能		支持	不支持	不允许。	9-18
	使用同步运转功能		支持	不支持	不使用	9-18
高功能 I/O 单元的循环刷新	禁止 SIOU 循环刷新		支持	支持	不禁止。	9-19
单元设定	基本 I/O 单元的输入响应时间		支持	支持	8ms	9-19
串行端口	模式 (若要设定模式，则必须使 CPU 单元上的 DIP 开关的第 5 位处于 OFF (默认) 状态。)		支持	支持 *	上位链接	9-23
外设服务	执行模式		支持	支持	正常	9-30
	将 “Time” (时间) 设定为 “All Events” (所有事件)		支持	支持	循环时间的 10%	9-30
FINS 保护	经由网络的 FINS 写保护设定		支持	支持	禁止 FINS 写保护。	9-31
I/O 模块	脉冲 I/O 模块的功能分配和详细设定		不支持	支持	---	9-32

* 不能选择用于串行 PLC 链接。

9-3 PLC 设置中的设定

9-3-1 启动运行设定



启动保持设定

● 强制状态保持位的启动保持设定

该参数用于设定启动时是否保持辅助区中的强制状态保持位 (A500.13)。

参数	设定	默认值	功能	相关标志和字
强制状态保持位的启动保持设定	OFF: 清除 ON: 保持	OFF	该设定决定启动时是否保持强制状态保持位 (A500.13) 的状态。	A500.13(强制状态保持位)

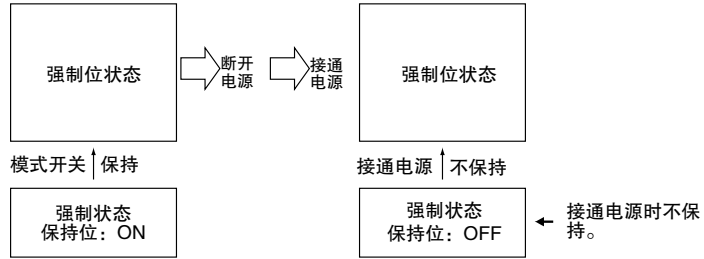
· 启动时强制状态保持位和该参数之间的关系

可将强制状态保持位 (A500.13) 置 ON, 从而保持当 CPU 单元的运行模式在 PROGRAM 和 MONITOR 之间切换时被强制置位或强制复位的所有位的状态。接通 PLC 的电源时, 强制状态保持位本身将置 OFF, 除非通过 PLC 设置中的参数设定对其进行保护。

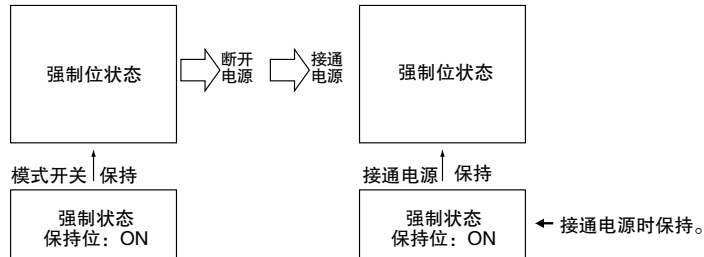
如果强制状态保持位的启动保持参数设定为 ON, 则当接通 PLC 的电源时, 强制状态保持位的状态将受到保护。如果将该参数设定为 ON 且强制状态保持位本身为 ON, 则所有在接通 PLC 的电源时, 所有强制置位和强制复位的位均将保持其强制状态。

如果由于电池电压低导致无法保持存储器, 则无论将该参数设定为 ON 还是 OFF, 强制状态保持位均会被清除。

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

该参数用于设定启动时是否保持辅助区中的 IOM 保持位的状态。

参数	设定	默认值	功能	相关标志和字
IOM 保持位的启动保持设定	OFF：清除 ON：保持	OFF	该设定决定启动时是否保持 IOM 保持位 (A500.12) 的状态。 若想在电源接通时保持 I/O 存储器中的所有数据，可将 IOM 保持位置 ON 并将该设定设为 ON。	A500.12(IOM保持位)

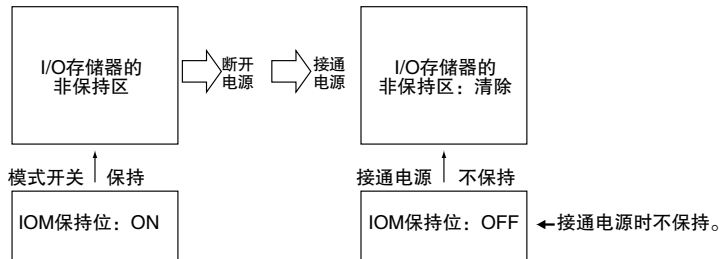
· IOM 保持位和该参数之间的关系

可将 IOM 保持位 (A500.12) 置 ON，从而在 CPU 单元的运行模式从 PROGRAM 切换为 RUN 或 MONITOR 或者反之之时保持 I/O 存储器中的所有数据。接通 PLC 的电源时，IOM 保持位本身将置 OFF，除非通过 PLC 设置中的设定对其进行保护。

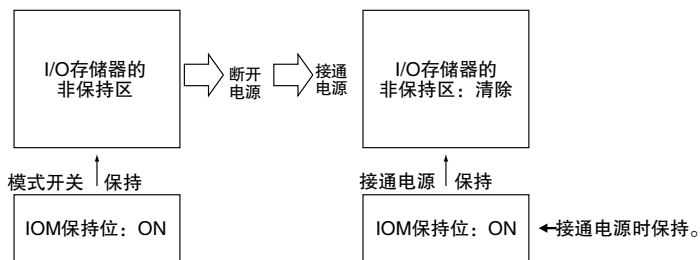
如果 IOM 保持位状态的启动设定为 ON，则当接通 PLC 的电源时，IOM 保持位的状态将受到保护。如果将该参数设定为 ON 且 IOM 保持位本身为 ON，则所有在接通 PLC 的电源时，则将保持 I/O 存储器中的所有数据。

如果由于电池电压低导致无法保持存储器，则无论将该参数设定为 ON 还是 OFF，IOM 保持位均会被清除。

OFF: 启动时清除IOM保持位



ON: 启动时保护IOM保持位



模式

● PROGRAM、MONITOR 或 RUN

设定启动时使用的运行模式。

参数	设定	默认值	功能	相关标志和字
模式	<ul style="list-style-type: none"> 编程: PROGRAM 模式 监视: 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

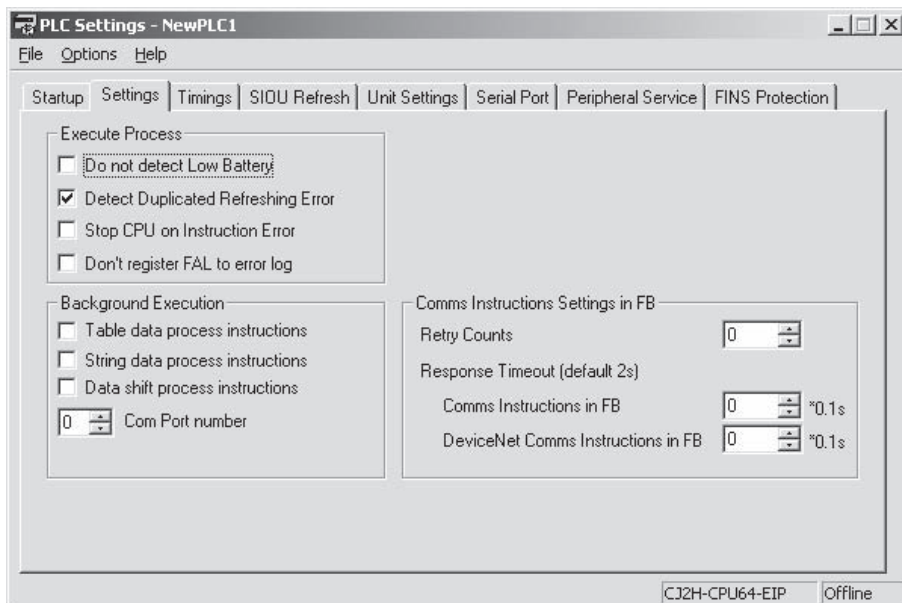
● 在初始化单元 / 内部板的识别状态之前启动程序运行

设定启动时是否等待特定单元的启动处理。

参数	设定	默认值	功能	相关标志和字
在初始化单元 / 内部板的识别状态时启动程序运行	OFF: 等待单元。 ON: 不等待单元。	OFF	如果要在一个或一个以上的单元*1尚未完成启动处理时以 RUN 或 MONITOR 模式启动 CPU 单元, 可将该设定设为 ON(不等待单元)。 若要等待所有单元完成启动处理, 可将该设定设为 OFF(等待单元)。	---

*1 该设定仅应用于特定单元。如果设定为“不等待”, 则 CPU 单元将只是不等待那些特定的单元, 即 CPU 单元仍将等待所有其它单元启动。

9-3-2 CPU 单元的设定



执行过程

● 不检测低电池电压 (无电池运行)

设定是否检测电池错误 (默认: 检测)。使用 “Do not detect” (不检测) 设定来进行无电池运行。详情请参考《CJ2 CPU 单元硬件操作手册》(样本编号: W472) 中的说明。

参数	设定	默认值	功能	相关标志和字
不检测电池低电压	OFF: 检测 ON: 不检测	OFF	该设定决定是否检测 CPU 单元的电池错误。如果将该设定设为 OFF(检测) 并且检测到电池错误, 则 CPU 单元上的 ERR/ALM 指示灯将闪烁, 且电池出错标志 (A402.04) 将置 ON, 但 CPU 单元将继续运行。	A402.04 (电池出错标志)



附加信息

无电池操作

· 无电池运行

当 PLC 无电池运行时将保持下述数据。

- 用户程序和网络符号
- 参数 (例如: PLC 设置、注册的 I/O 表、路由表、CPU 总线单元设定、PLC 名称)
- 保存在 CPU 总线单元和高功能 I/O 单元的非易失性存储器中的数据 (例如串行通信单元中的协议宏数据)

不保持下述数据。这些数据的值将不稳定。

- I/O 存储器 (包括保持区、DM、EM 和辅助区)
- 内置到 CPU 单元中的时钟

· 设定无电池运行

· PLC 设置

按如下步骤设定 PLC 设置。

- (1) 无电池时 I/O 存储器将不稳定, 因此请清除 “IOM Hold Bit” (IOM 保持位) 的勾选标记, 从而在启动时清除 IOM 保持位。
- (2) 无电池时强制状态将不稳定, 因此请清除 “Forced Status Hold Bit” (强制状态保持位) 的勾选标记, 从而在启动时清除强制状态保持位。
- (3) 勾选 “Do not detect Low Battery” (不检测低电池电压) 复选框。

· 编程

无电池时输出 OFF 位的操作将不稳定, 因此请插入下述指令, 从而使输出 OFF 位不置 ON。

```
LD P_Off
OUT A500.15
```

· 无电池运行的注意事项

无电池运行时, 请注意以下要点。

· 初始化数据

DM区、EM区和其它I/O存储区中的数据值将不稳定。请务必从程序中设定这些数据的初始值。

例:

```
LD A200.11
MOV #0918 D0 // 在运行开始时初始化 D0 值
```

· 时钟不稳定

内置到 CPU 单元中的时钟将不运行, 且值将不稳定。因此, 出错日志中记录的日期和时间数据将无法正确显示。此外, 如果将文件保存在存储卡上, 则文件创建的日期和时间将不稳定。



正确使用注意事项

CPU 单元中的 DM、EM、HR 和 AR 区不备份至内部的闪存存储器中。当电源关闭或中断时，DM、EM、HR 和 AR 区中的内容通过电池保持。如果发生电池错误，则该数据可能会丢失。使用电池出错标志 (A402.04)，在程序中采取措施对数据进行初始化。

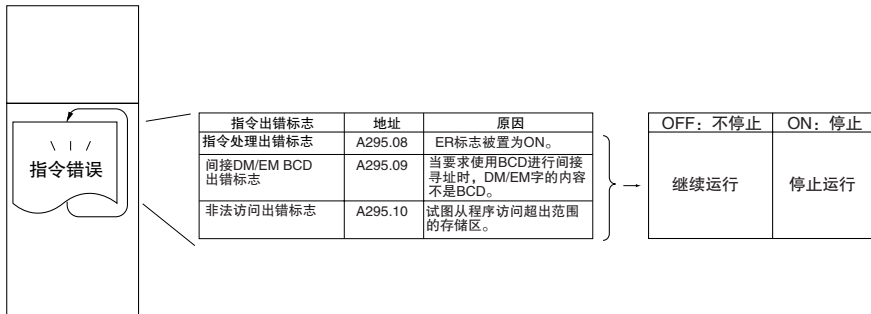
● **检测重复刷新错误 (设定是否检测重复刷新错误)**

该设定决定是否检测重复刷新错误 (默认：检测)。如果将该参数设定为默认值 (ON：检测)，则在上述情况下将检测重复刷新错误。

参数	设定	默认值	功能	相关标志和字
检测重复刷新错误	OFF：不检测 ON：检测	ON	该设定决定是否检测重复刷新错误。如果将该设定设定为 ON(检测) 并且检测到错误，则 CPU 单元上的 ERR/ALM 指示灯将闪烁，且重复刷新出错标志 (A402.13) 将置 ON。CPU 单元将继续运行。	A402.13(重复刷新出错标志)

● **发生指令错误时停止 CPU 运行**

该设定决定在发生指令执行错误时是否停止运行 (默认：不停止)。调试程序时，采用“发生指令错误时停止 CPU 单元的运行”的设定。如果设定为“发生指令错误时停止 CPU 单元的运行”，则在上述任意标志置 ON 时将产生一条程序错误 (作为指令错误)，以停止 CPU 单元的运行。



参数	设定	默认值	功能	相关标志和字
当指令出错时停止 CPU 运行	OFF：继续 ON：停止	OFF	该设定决定将指令错误(指令处理错误 (ER) 和非法访问错误 (AER) 作为非致命错误还是致命错误来对待。	A295.08(指令处理出错标志)、A295.09(间接 DM/EM BCD 出错标志)、A295.10(非法访问出错标志) (如果将该设定设定为 OFF，则即使发生指令错误，这些标志也将不置 ON。)

● 不将 FAL 注册到出错日志中

该参数决定在发生用户自定义的 FAL 错误时是否将错误注册到出错日志中。

参数	设定	默认值	功能	相关标志和字
不将 FAL 注册到出错日志中	OFF: 将用户自定义的 FAL 错误注册到出错日志中。 ON: 不将用户自定义的 FAL 错误注册到出错日志中。	OFF	该设定决定是否将通过 FAL(006) 指令创建的用户自定义 FAL 错误和 FPD(269) 指令的时间监控数据记录到出错日志中 (A100 ~ A199)。将其设定为 ON 时, 不记录这些错误。	---

后台执行设定

下述指令将对循环时间产生重大影响，具体取决于这些指令所处理的数据量。后台执行设定可用于将对指令分割到一个以上的循环中进行处理，从而减小对循环时间的影响。详情请参考“10-2-5 后台执行”。

● 表数据处理指令

该参数决定是否以后台处理表数据指令。

参数	设定	默认值	功能	相关标志和字
表数据处理指令	OFF: 不以后台执行。 ON: 以后台执行。	OFF	该设定决定是否将表数据指令分割到多个循环时间内进行处理(即后台执行)。	---

● 串数据处理指令

该参数决定是否以后台处理串数据指令。

参数	设定	默认值	功能	相关标志和字
串数据处理指令	OFF: 不以后台执行。 ON: 以后台执行。	OFF	该设定决定是否将串数据指令分割到多个循环时间内进行处理(即后台执行)。	---

● 数据移位处理指令

该参数决定是否以后台处理数据移位指令。

参数	设定	默认值	功能	相关标志和字
数据移位处理指令	OFF: 不以后台执行。 ON: 以后台执行。	OFF	该设定决定是否将数据移位指令分割到多个循环时间内进行处理(即后台执行)。	---

● 通信端口号

该参数设定在后台处理中使用的通信端口号。

参数	设定	默认值	功能	相关标志和字
通信端口号	0 ~ 7: 通信端口 0 ~ 7(内部逻辑端口)	0(0号)	将用于后台执行的通信端口号(内部逻辑端口)。	---



正确使用注意事项

如果在 PLC 设置中设定为允许高速中断功能，则无法在中断任务中执行后台处理。此时将会发生指令处理错误。

FB 中的通信指令设定 (对于 FB 库)

下述参数仅用于 OMRON 功能块，不用于任何其它应用场合。

尤其在通过 OMRON FB 库中的功能块执行 FINS 报文或 DeviceNet 显式报文通信时，用户必须在 PLC 设置的 FB 通信指令设定中设定重新发送次数和响应监控时间。在 PLC 设置中为 OMRON FB 库设定的值将会被自动存储到相关的辅助区字 A580 ~ A582 中，并可由 OMRON FB 库中的功能块进行调用。

● 重试计数

该参数决定在功能块中重试执行通信指令的次数。

参数	设定	默认值	功能	相关标志和字
重试计数	0 ~ 15	0	设定在功能块内执行 DeviceNet 显式报文或 FINS 报文时发送命令的重试次数。	A580.00 ~ A580.03

● FB 中的通信指令响应超时

该参数决定对在功能块中执行通信指令的响应监控时间。

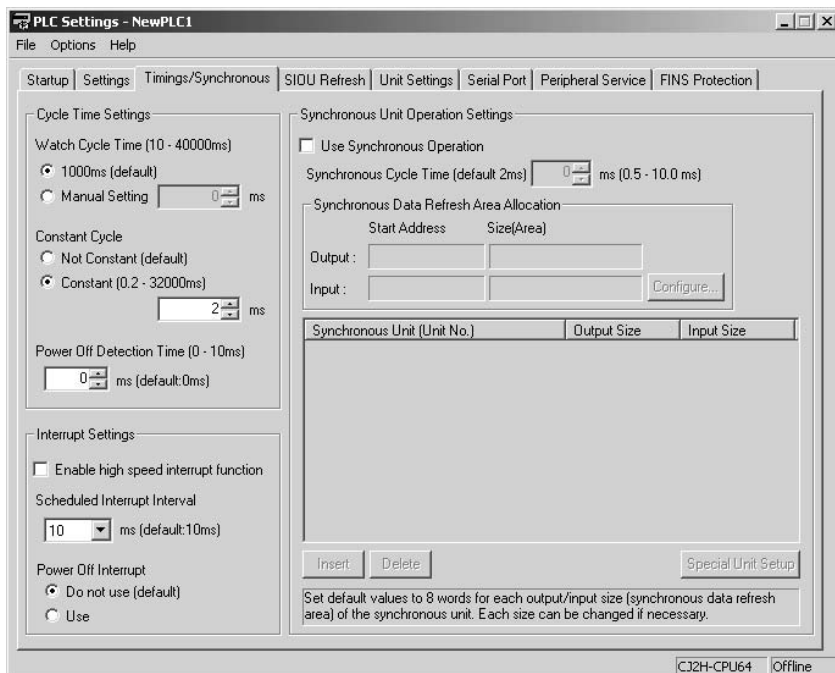
参数	设定	默认值	功能	相关标志和字
FB 中的通信指令设定	0001 ~ 65535 (单位: 0.1s, 0.1 ~ 6553.5) 0000: 2s	0000: 2s	当在此处设定的时间内未对功能块内部执行的 FINS 命令作出响应时, 即发生响应超时。	A581

● FB 中的 DeviceNet 通信指令响应超时

该参数决定对在功能块中执行 DeviceNet 通信指令的响应监控时间。

参数	设定	默认值	功能	相关标志和字
FB 中的 DeviceNet 通信指令设定	0001 ~ FFFF (单位: 0.1s, 0.1 ~ 6553.5) 0000: 2s	0000: 2s	当在此处设定的时间内未对功能块内部执行的显式报文命令作出响应时, 即发生响应超时。	A582

9-3-3 时间 / 同步设定



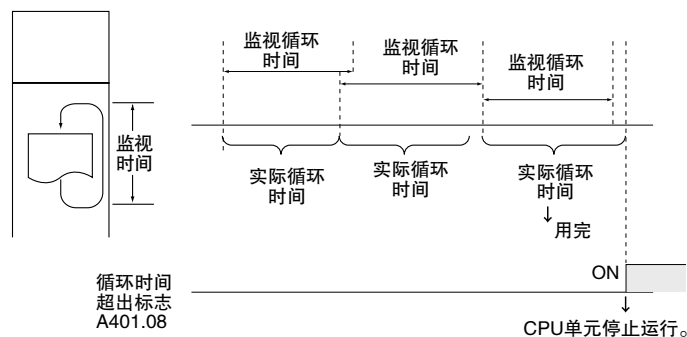
● 监视循环时间

该参数用于将“Watch Cycle Time”（监视循环时间）设定为除默认设定 (1000ms) 之外的值。

参数	设定	默认值	功能	相关标志和字
监视循环时间	ON: 默认 (1,000ms: 1s) OFF: 手动设定	ON (1,000ms: 1s)	设定为 OFF, 从而将所有监视循环时间设定为除默认值 1s 之外的其它值。	A401.08(循环时间超出标志)
	10 ~ 40,000ms (以 10ms 为增量)	0	仅当将上述参数设定为 OFF(手动设定) 时该设定有效。如果循环时间超过为监视循环时间设定的值, 则 A401.08(循环时间超出标志) 将变为 ON。	A264 和 A265 (当前循环时间)

· 需要改变监视循环时间时

如果循环时间超过监视 (最大) 循环时间的设定值, 则循环时间超出标志 (A401.08) 将变为 ON, 且 PLC 将停止运行。如果正常循环时间超出默认的监视循环时间设定, 即 1s, 则必须改变该参数。



注 监视循环时间的默认值为 1s(1,000ms)。

● 循环时间

该参数设定在使用最小循环时间功能时的最小循环时间 (默认值: 可变循环时间)。

参数	设定	默认值	功能	相关标志和字
恒定循环时间	OFF: 最小循环时间 ON: 可变循环时间	ON(可变)	将该设定设为 OFF 以使用最小循环时间。如果要使用最小循环时间, 则必须设定循环时间。	---
	0.2 ~ 32,000ms(以 0.1ms 为增量)	---	设定为 0.2 ~ 32000.0 以指定最小循环时间。如果循环时间小于该设定, 则循环时间将延长直到该时间过去位置。若要使用可变循环时间, 则使该设定为 0。	---



附加信息

可从 CX-Programmer 的显示画面来改变最小循环时间, 从而在 CPU 单元处于 MONITOR 模式时监控循环时间 (1.1 版或更高版本的单元)。

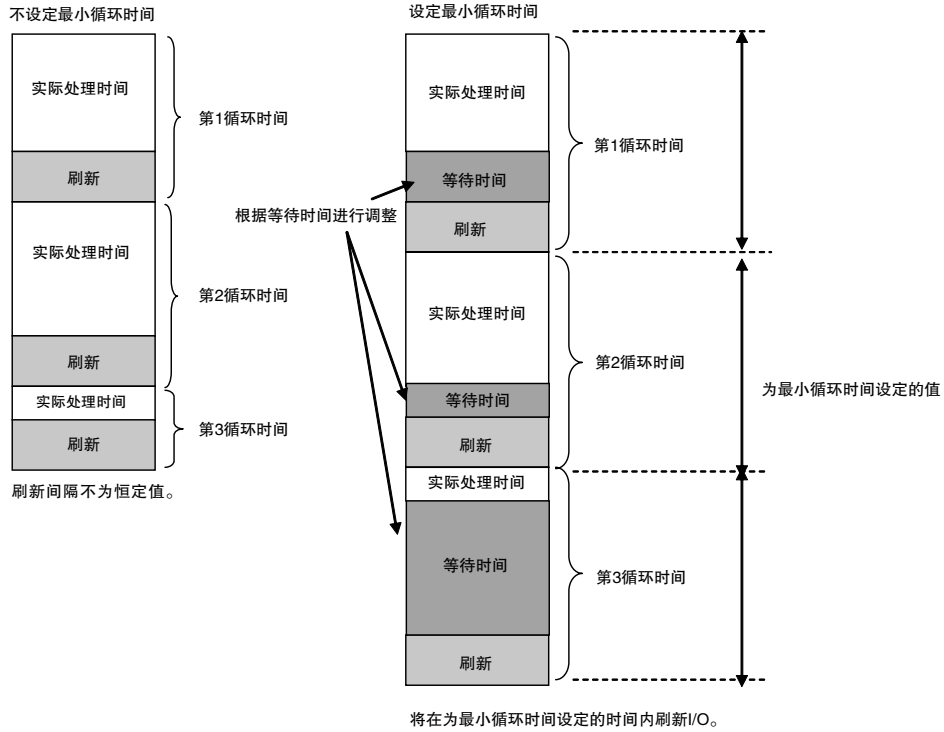


正确使用注意事项

如果设定了较长的循环时间, 则对支持软件的服务间隔也将延长, 从而可能会导致联机操作响应迟缓以及联机困难。

· 允许最小循环时间的条件

将最小循环时间设定为非零值，可消除 I/O 响应中的矛盾之处。仅当实际的循环时间小于最小循环时间设定时，该参数有效。如果实际的循环时间大于最小循环时间设定，则实际的循环时间将保持不变。



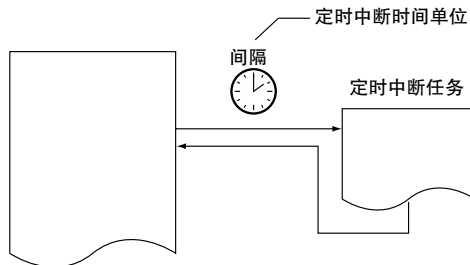
● 定时中断间隔

该参数将用于定时中断间隔的时间单位设定为 1ms、0.1ms 或者默认值 10ms。

参数	设定	默认值	功能	相关标志和字
定时中断间隔	<ul style="list-style-type: none"> · 10ms · 1.0ms · 0.1ms 	10ms	该设定决定用于定时中断间隔设定中的时间单位。	---

· 定时中断时间单位

该参数设定用于定时中断间隔设定的时间单位。通过 MSKS(690) 指令从程序中设定定时中断间隔。



● 电源 OFF 检测延时

该参数用于将电源 OFF 检测延时设定为除 0ms 以外的值。

参数	设定	默认值	功能	相关标志和字
电源 OFF 检测时间	0 ~ 10ms (以 1ms 为增量)	0ms	该设定决定从检测到电源中断(在电源电压降至额定值的 85% 以下约 10 ~ 25ms(对于 AC 电源)或 2 ~ 5ms(对于 DC 电源)之后)到确认为电源中断之间的延迟量。默认设定为 0ms。 当设定了允许电源 OFF 中断任务时,则在确认了电源中断时将执行该中断任务。如果设定为禁止电源 OFF 中断任务,则 CPU 单元将复位且运行将停止。	---

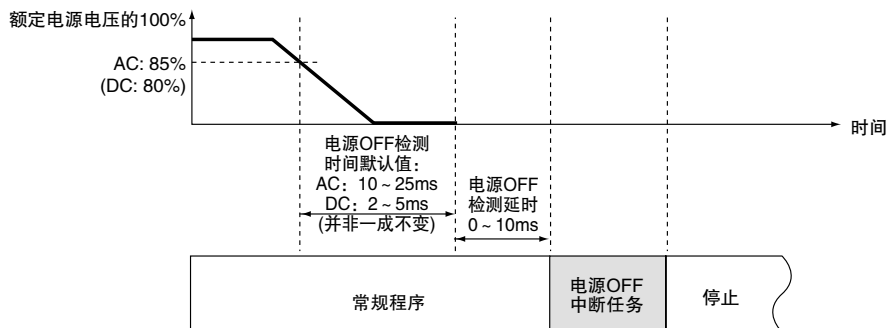
注 安装了 CJ1W-PD022 电源单元时不支持该参数。

电源 OFF 检测延时

该参数决定从检测到电源中断直到确认电源中断成立和常规程序停止为止的延迟量。可将该参数设定为 0 ~ 10ms 之间。

当因存在故障的电源中发生暂时性电源中断导致 PLC 运行停止时,将延时延长到检测出电源中断为止。在经过初始电源中断检测时间之后,内部 5VDC 电源降至 0VDC 最长需要 10ms 的时间。这 10ms 钟即为电源保持时间。该时间等于电源 OFF 检测时间加上确认电源中断所需的处理时间。AC 电源的电源 OFF 延时为 10 ~ 25ms(并非始终一致),CJ1W-PD025 DC 电源单元的电源 OFF 延时为 2 ~ 5ms, CJ1W-PD022 DC 电源单元的电源 OFF 延时为 2 ~ 10ms。

注 安装了 CJ1W-PD022 电源单元时不支持该参数。(请参考“A-5 断电时的操作”。)



注 电源 OFF 中断任务中程序的执行时间必须小于 10ms 减去电源 OFF 检测延时。有关电源 OFF 中断任务的说明,请参考“5-2-3 中断任务”。有关发生电源中断时的操作,请参考“A-5 断电时的操作”。

● 禁止电源 OFF 中断

该参数决定是否允许电源 OFF 中断任务。

参数	设定	默认值	功能	相关标志和字
禁止电源 OFF 中断	ON: 禁止 OFF: 允许	ON	当将该参数设定为 OFF(允许)时,则在发生电源中断时将执行电源 OFF 中断任务。	---

注 安装了 CJ1W-PD022 电源单元时不支持该参数。

● 电源 OFF 中断任务

该参数决定在检测到电源中断时是否执行电源 OFF 中断任务。(当将该参数设定为 ON 时,常规程序将在检测到电源中断时停止。)

当电源保持时间(电源中断后的处理时间 + 电源 OFF 检测延时)已过时,电源 OFF 中断任务将停止。最大电源保持时间为 10ms。

若需设定电源 OFF 检测延时,则请确保电源 OFF 中断任务能够在可用时间内(10ms 减去电源 OFF 检测延时)执行。

注 安装了 CJ1W-PD022 电源单元时不支持该参数。(请参考“A-5 断电时的操作”。)

● 允许高速中断功能 (仅限 CJ2H CPU 单元)

该参数用于设定允许或是禁止高速中断功能。

参数	设定	默认值	功能	相关标志和字
允许高速中断功能	OFF: 禁止高速中断功能 ON: 允许高速中断功能	OFF(禁止高速中断)	选择了“Enable high-speed interrupt function”(允许高速中断功能)选项时,即允许使用高速中断功能。	---

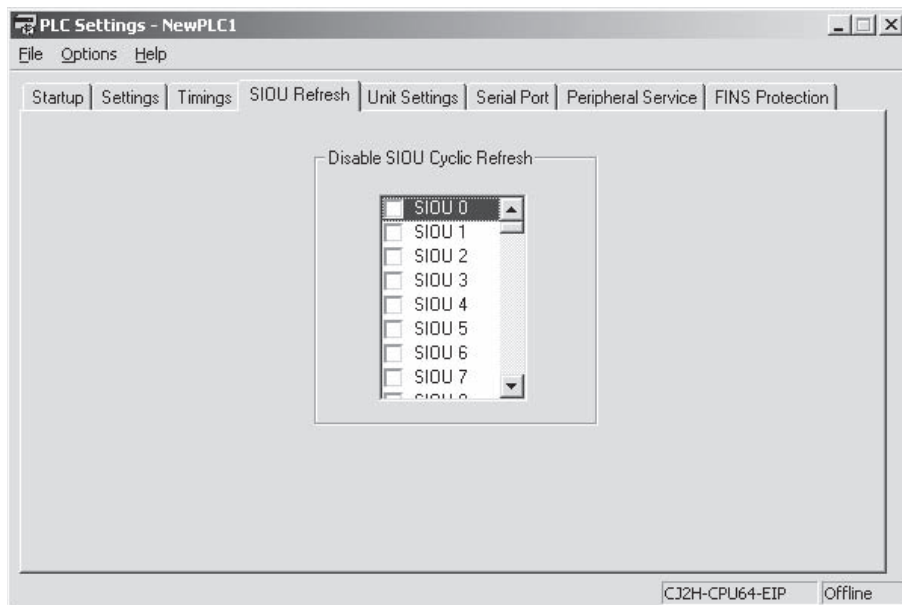
高速中断功能可提升中断任务在某些限制条件下的执行效能。详情请参考“10-2-6 高速中断功能”。

如果 PLC 设置中勾选了“Use Synchronous Operation”(使用同步运行)复选框,则将自动选择“Enable high-speed interrupt function”(允许高速中断功能)复选框。

● 同步单元运行设定 (仅限 CJ2H CPU 单元)

若要使用同步单元运行功能,则需要这些设定。详情请参考“10-8-7 PLC 设置”。

9-3-4 高性能 I/O 单元的循环刷新



禁止 SIOU 循环刷新

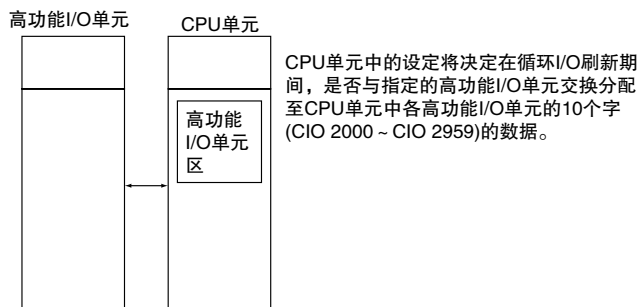
这些参数指定是否禁止对高性能 I/O 单元 (SIOU) 的循环刷新。

参数	设定	默认值	功能	相关标志和字
对单元 0 ~ 95 进行循环刷新 15 个单元中的每个组均有一个参数	OFF: 允许 ON: 禁止	OFF(禁止)	这些设定决定在对高性能 I/O 单元进行循环刷新期间, 是否在指定单元和高性能 I/O 单元的分配字 (10 个字 / 单元) 之间交换数据。	---

● 禁止高功能 I/O 单元的循环刷新：

如果将要使用 IORF(097)、FIORF(225)、IORD(222) 或 IOWR(223) 指令在中断任务中刷新高功能 I/O 单元，则请始终禁止对高功能 I/O 单元的循环刷新。如果在对高功能 I/O 单元允许了循环刷新的情况下，在中断任务中执行了下述任一步骤，则将产生重复刷新错误（非致命错误），且重复刷新出错标志(A402.13)将置 ON。

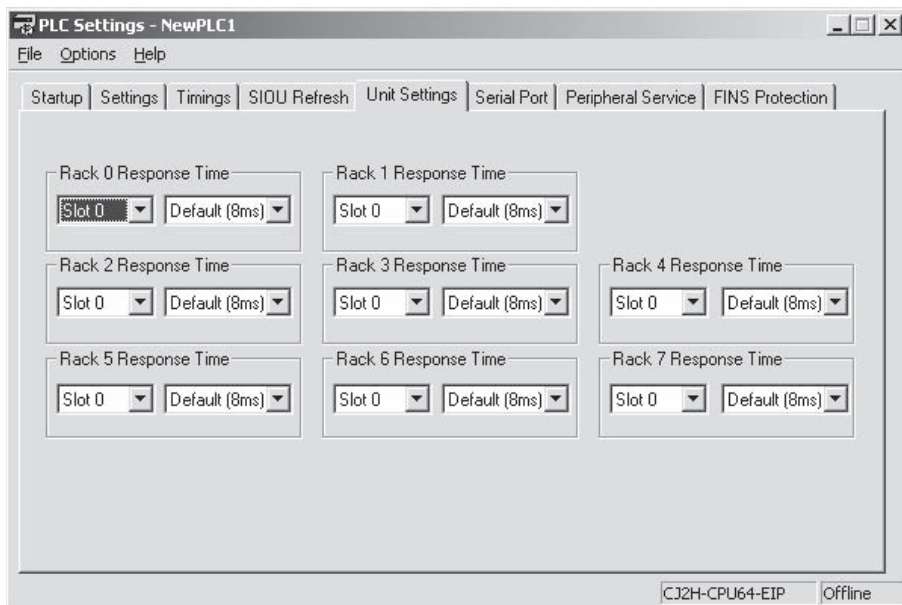
- 通过 IORF(097)/FIORF(225) 指令对同一个高功能 I/O 单元执行 I/O 刷新。
- 通过 IORD(222)/IOWR(223) 指令对同一个高功能 I/O 单元中的存储区执行数据读 / 写操作。



正确使用注意事项

只要禁止了高功能 I/O 单元的循环刷新时，均务必确保在程序中使用 IORF(097) 或 FIORF(225) 指令在运行期间至少每 11 秒钟对该单元进行一次 I/O 刷新。如果不每 11 秒钟刷新一次，则高功能 I/O 单元中将产生 CPU 单元服务监控错误。（高功能 I/O 单元上的 ERH 和 RUN 指示灯将点亮。）

9-3-5 基本 I/O 单元的机架响应时间



机架 0 ~ 7 机架响应时间

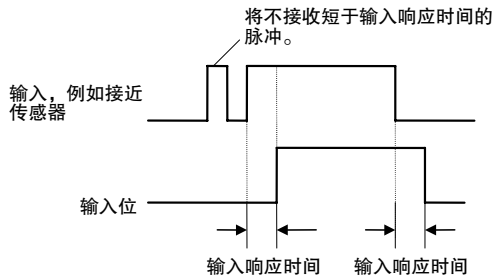
这些参数用于将基本 I/O 单元的输入响应时间设定为除默认设定 (8ms) 之外的值。

将 PLC 设置传送至 CPU 单元之后，必须断开供至 PLC 的电源然后再重新接通。

参数	设定	默认值	功能	相关标志和字
机架 0, 插槽 0 ~ 9	默认: 8ms	8ms	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)。 如果未设定滤波器 (0ms), 则在 ON 或 OFF 响应时间中将有一个基于单元的内部元件的延迟。有关各单元的 ON/OFF 响应时间延迟的详情, 请参考《CJ2 CPU 单元硬件操作手册》(样本编号: W472)。	A220 ~ A259: 基本 I/O 单元的实际输入响应时间
机架 1, 插槽 0 ~ 9	无滤波器			
机架 2, 插槽 0 ~ 9	0.5ms			
机架 3, 插槽 0 ~ 9	1ms			
机架 4, 插槽 0 ~ 9	2ms			
机架 5, 插槽 0 ~ 9	4ms			
机架 6, 插槽 0 ~ 9	8ms			
机架 7, 插槽 0 ~ 9	16ms			
机架 7, 插槽 0 ~ 9	32ms			

● 更改基本 I/O 单元的机架响应时间

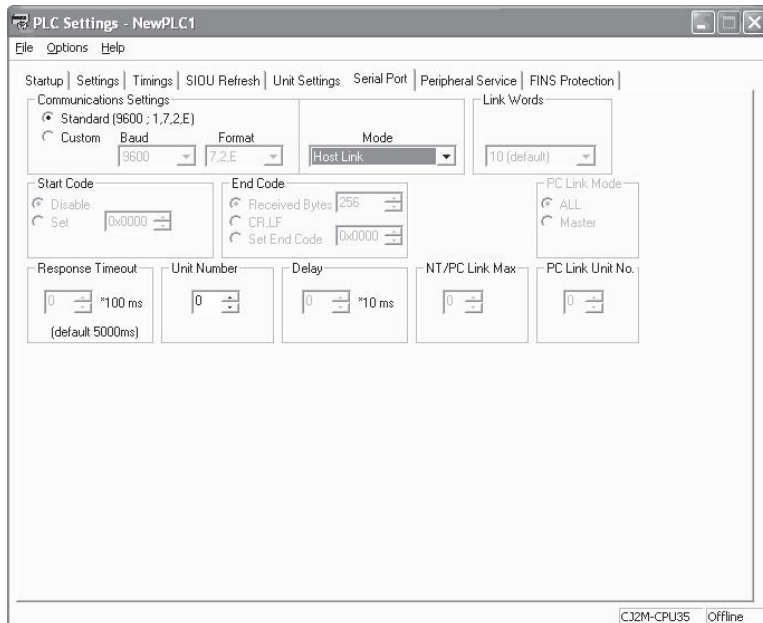
可按机架和插槽来为基本 I/O 单元设定机架响应时间。增大设定将减轻颤振和干扰效应，而减小设定将缩短输入脉冲的接收时间。请勿将 ON 响应时间或 OFF 响应时间设定为小于循环时间。



接通 PLC 的电源时，机架响应时间设定将传送至基本 I/O 单元。

改变了单元的设定时，这些设定将存储到 A220 ~ A259(基本 I/O 单元的实际输入响应时间)中。当在 PLC 处于 PROGRAM 模式下改变了 PLC 设置中的设定时，PLC 设置中的设定将与单元中的实际设定不同。在这种情况下，可通过查看 A220 ~ A259 中的值来了解单元中实际设定的输入响应时间。

9-3-6 串行端口设定



当 CPU 单元上的 DIP 开关的第 5 位为 OFF(默认) 时, 下列参数有效。

通信设定

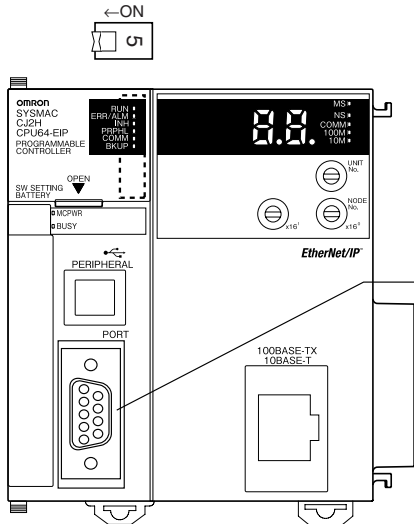
参数	设定	默认值	功能	相关标志和字
通信设定	<ul style="list-style-type: none"> 标准 (9600: 1, 7, 2, E): 标准设定 自定义: 任意设定 	标准	标准设定 (9600: 1, 7, 2, E) 用于上位链接模式, 1 个起始位, 7 个数据位, 偶校验, 2 个停止位, 且波特率为 9,600bps。设定为自定义设定可使用任何其它通信设定。	A619.02 (串行端口设定修改标志)

- 串行端口设定

需要将 PLC 设置中的串行端口设定从默认值修改为其它设定时, 请对其进行设定。选择了无协议模式时指定帧格式。

还可通过 STUP(237) 指令来修改端口设定。执行了 STUP(237) 指令时, 串行端口设定修改标志 (A619.02) 将置 ON; 修改了端口设定时, 该标志将置 OFF。

当 CPU 单元前面板上的 DIP 开关的第 5 位为 ON 时, CPU 将自动检测连接到 RS-232C 端口的编程设备 (包括手持式编程器) 的通信参数。这些自动检测的参数不存储在 PLC 设置中。



当DIP开关的插针5处于OFF状态时的串行端口通信设定:

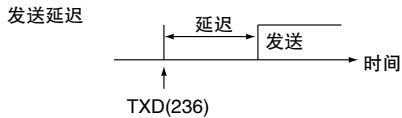
默认设定:

上位链接模式, 1个起始位, 7个数据位, 偶校验, 2个停止位, 且波特率为9,600bps

自定义设定:

设定通信模式(上位链接, NT链接, 无协议*或外设总线)和其它设定, 例如波特率。

可在无协议模式下设定发送延迟。该延迟的操作如下图所示。



下表中列出了可设定为在无协议模式下发送和接收的报文格式。

		结束码设定		
		无	支持	CR+LF
起始码设定	无	DATA	DATA ED	DATA CR+LF
	支持	ST DATA	ST DATA ED	ST DATA CR+LF
接收的字节数		数据: 1 ~ 256 字节		

选择 “Standard (9600: 1,7,2,E)” (标准 (9600: 1,7,2,E))

如果选择了 “Standard (9600: 1,7,2,E)” (标准 (9600: 1,7,2,E)), 则将无条件使用下列通信设定。

- 波特率: 9,600bps
- 起始位: 1 位
- 数据长度: 7 位
- 停止位: 2 位
- 奇偶校验: 偶校验
- 模式 (串行通信模式): 上位链接

注 模式和其它设定将被禁止。

使用下述步骤将通信设为自定义设定。

1. 选择 “Custom” (自定义) 作为通信设定。
2. 在 “Mode” (模式) 字段中选择串行通信模式。
3. 进行其它设定。

选择 “Custom” (自定义)

如果选择了自定义设定, 则可选择下述波特率、参数和模式。

波特率 (选择了自定义设定时)

设定	含义	设定	含义	相关标志和字
115200	11,5200bps	4800	4,800bps	A619.02(串行端口设定修改标志)
57600	57,600bps	2400	2,400bps	
38400	38,400bps	1200	1,200bps	
19200	19,200bps	600	600bps	
9600(默认)	9,600bps	300	300bps	

● 格式 (选择了自定义设定时)

所选数据	数据长度	停止位	奇偶校验	相关标志和字
7,1,E	7	1 位	偶	A619.02(串行端口设定修改标志)
7,1,O			奇校验	
7,1,N			无	
7,2,E(默认)		2 位	偶	
7,2,O			奇校验	
7,2,N			无	
8,1,E	8	1 位	偶	
8,1,O			奇校验	
8,1,N			无	
8,2,E		2 位	偶	
8,2,O			奇校验	
8,2,N			无	

模式 (选择了自定义设定时)

为内置 RS-232C 端口选择串行通信模式。

参数	设定	默认值	功能	相关标志和字
模式	<ul style="list-style-type: none"> · 上位链接 · NT 链接 (1:N) · RS-232C(无协议) · 外设总线 (Toolbus) · 串行网关 · 串行 PLC 链接 (主站) · 串行 PLC 链接 (从站) 	上位链接 *1	该设定决定串行端口是以上位链接模式还是其它串行通信模式运行。 *2	A619.02 (串行端口设定修改标志)

*1 上位链接 (SYSMAC WAY) 是用于与通用的上位计算机进行连接的通信模式。

*2 将无法与被设定为 1:1 NT 链接的 PT 进行通信。

● 各选择模式的相关设定

模式	相关设定
上位链接 (默认)	波特率、参数、单元号
NT 链接 (1:N)	波特率、NT/PC 链接的最大单元号
RS-232C(无协议)	波特率、参数、起始码、结束码、延迟
外设总线 (Tool Bus)	波特率
串行网关	波特率、参数、响应监控时间
串行 PLC 链接 (主站)	通信设定、模式、链接字数、链接方法、NT 链接或串行 PLC 链接中的最大单元号 (No. NT/PC Link Max.)
串行 PLC 链接 (从站)	通信设定、模式、串行 PLC 链接从站单元号

● 上位链接设定

通过与计算机或其它主机设备的主机链接来执行通信时，或者将要使用除 9,600bps 以外的波特率时，可使用上位链接模式。

· 单元号 (对于上位链接模式下的 CPU 单元)

为 PLC 从站设定单元号。

参数	设定	默认值	功能	相关标志和字	更新 CPU 单元中的时间
单元号	0 ~ 31	0	该设定决定当 CPU 单元以 1 到 N(N=2 ~ 32) 上位链接模式进行连接时的单元号。	A619.02 (串行端口设定修改标志)	每个循环

● NT 链接设定

NT 链接模式用于执行与 OMRON 可编程终端 (NS 系列或 NT 系列) 的通信。

波特率 (bps)

参数	设定	默认值	功能	相关标志和字
波特率	115,200, 38,400bps	115,200bps	当从 CX-Programmer 设定该值时，建议设定为 115,200bps。	A619.02 (串行端口设定修改标志)

NT/PC 链接的最大单元号

参数	设定	默认值	功能	相关标志和字
NT/PC 链接的最大单元号	0 ~ 7	1	该设定决定可连接到 PLC 的 PT 的最大单元号。	A619.02 (串行端口设定修改标志)

● RS-232C(无协议) 设定

无协议模式用于执行与条形码读码器或其它设备的无协议通信。

起始码

参数	设定	默认值	功能	相关标志和字
起始码	禁止或设定	禁止	设定是否对要通过无协议通信发送和接收的数据的帧格式使用起始码。	---
	0x0000 ~ 00FF (0000 ~ 00FF hex: "0x" 表示与 Hex 相同。)	0x0000 (00 Hex)	将起始码作为十六进制值设定。	

结束码

参数	设定	默认值	功能	相关标志和字
结束码	<ul style="list-style-type: none"> · 接收字节 (无结束码) · CR LF · 设定结束码 	接收字节	设定要通过无协议通信发送和接收的数据的帧格式将要使用的结束码类型。	---
接收字节	256 或 1 ~ 255	256	如果为结束码选择了接收的字节数, 则将为无协议通信发送和接收设定以字节为单位的数据长度。 结束码和起始码的长度不含在数据长度中。 一条TXD(236)/RXD(235)指令可发送或接收的最大数据长度默认为 256 字节。	
设定结束码	0x0000 ~ 00FF (0000 ~ 00FF hex: "0x" 表示与 Hex 相同。)	0x00 (00 Hex)	如果为结束码指定了结束码, 则在此将其作为十六进制值进行设定。	

延迟

参数	设定	默认值	功能	相关标志和字
延迟	0 ~ 99990ms (以 10ms 为增量)	0ms	执行了 TXD(236) 指令时, 将在此处所设定的延迟时间过后, 从串行端口发送数据。	---

● 外设总线 (Toolbus) 设定

选择外设总线模式来执行与 CX-Programmer 或其它编程设备的通信。

波特率

参数	设定	默认值	功能	相关标志和字
波特率	9600, 19200, 38400, 57600, 115200bps	115200 bps	指定了外设总线 (Toolbus)。	A619.02 (串行端口设定修改标志)

通过将 CPU 单元上的 DIP 开关的第 5 位设定为 ON, 即可自动检测 CX-Programmer 处的波特率并连接外设总线。

● 串行网关设定

选择串行网关协议来执行通过 CompoWay/F 与 OMRON 部件的通信。

响应监控时间

参数	设定	默认值	功能 *1	相关标志和字
响应监控时间	5000ms, 100 ~ 25500 ms (以 100ms 为增量)	0(50000 ms)	监控从发送使用串行网关转换成指定协议的 FINS 命令开始直到接收到响应为止的时间。 默认: 5s; PLC 设置: 0.1 ~ 25.5s	A619.02 (串行端口设定修改标志)

*1 如果发生超时, 则将 FINS 接收码 0205 Hex(响应超时) 返回至 FINS 源。

● 串行 PLC 链接主站单元设定 (仅限 CJ2M CPU 单元)

选择串行 PLC 链接主站单元, 从而允许在 CJ2M CPU 单元之间或 CJ2M CPU 单元与 CJ1M/CP1H/CP1L/CP1E CPU 单元之间交换数据, 而无需特殊编程。

通信设定

参数	设定	默认值	功能	相关标志和字
波特率	38400, 115200bps	115200bps	选择在指定串行 PLC 链接主站单元时的波特率。	A619.02(串行端口设定修改标志)

链接字数

参数	设定	默认值	功能	相关标志和字
链接字	1 ~ 10 个字	10 个字	该参数仅可在主站单元中进行设定, 设定在串行 PLC 链接区中每个结点所使用的字数。	A619.02(串行端口设定修改标志)

PLC 链接方法

参数	设定	默认值	功能	相关标志和字
链接方法	全站链接方式或主站链接方式	全站链接	该设定指定对于串行PLC链接的链接方法。 该参数仅可在主站单元中进行设定，	A619.02(串行端口设定修改标志)

NT/ 串行 PLC 链接的最大单元号

参数	设定	默认值	功能	相关标志和字
NT/PC 链接的最大单元号	0 ~ 7	0	该设定决定当为串行 PLC 链接使用 1:N 连接时，连接到主站单元上的从站单元的最大单元号。 该参数仅可在主站单元中进行设定，	A619.02(串行端口设定修改标志)

- 串行 PLC 链接从站单元设定 (仅限 CJ2M CPU 单元)

选择串行 PLC 链接从站单元，从而允许在 CJ2M CPU 单元之间或 CJ2M CPU 单元与 CJ1M/CP1H/CP1L/CP1E CPU 单元之间交换数据，而无需特殊编程。

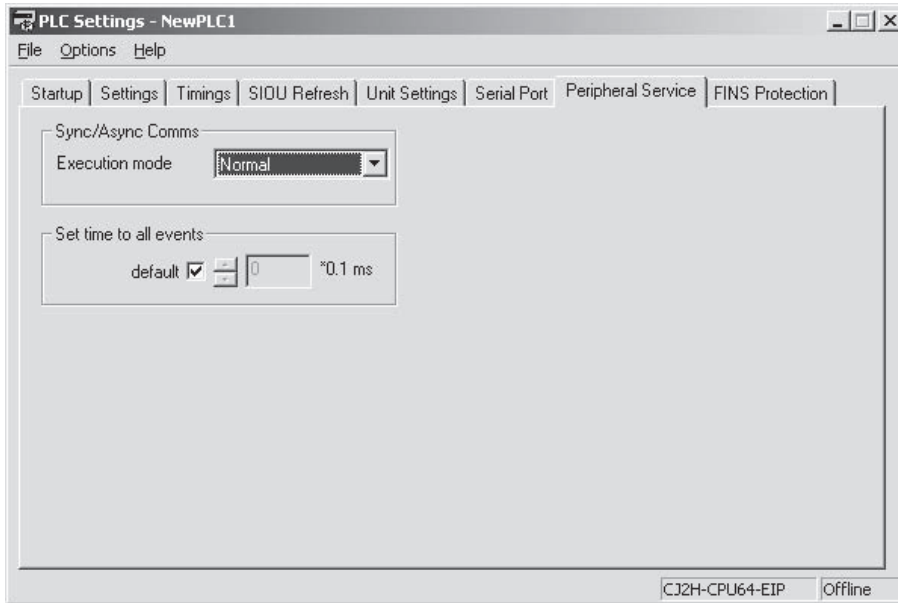
通信设定

参数	设定	默认值	功能	相关标志和字
波特率	38400, 115200bps	115200bps	选择在指定串行PLC链接从站单元时的波特率。	A619.02(串行端口设定修改标志)

串行 PLC 链接从站单元的单元号

参数	设定	默认值	功能	相关标志和字
串行 PLC 链接的单元号	0 ~ 7	0	设定当为串行 PLC 链接使用 1:N 连接时，连接到主站单元上的从站单元的单元号。	A619.02(串行端口设定修改标志)

9-3-7 外设服务



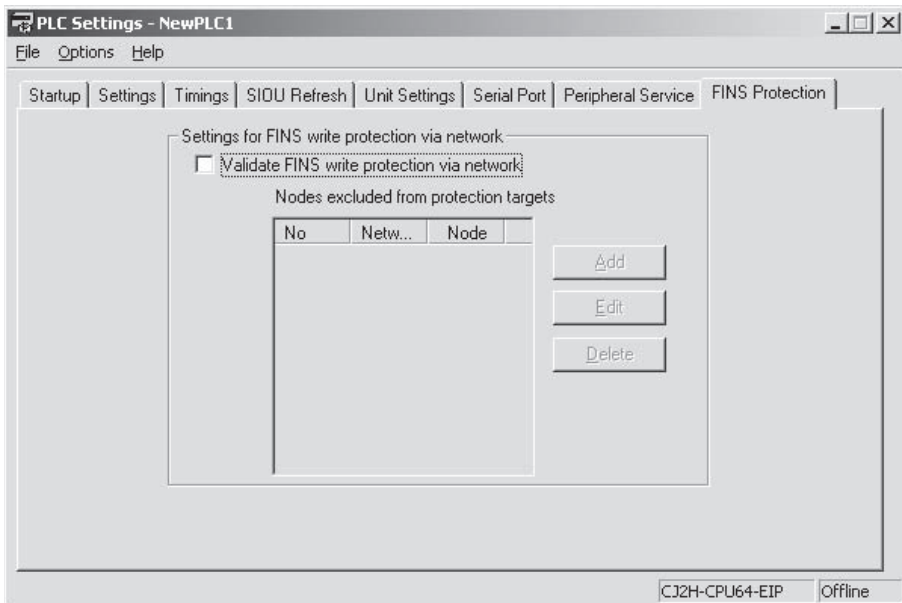
- Sync/Async 通信 (CPU 处理模式)

参数	设定	默认值	功能	相关标志和字
执行模式	普通模式	普通模式	CJ2 CPU 单元仅支持普通模式。	---

- 将 “Time” (时间) 设定为 “All Events” (所有事件)

参数	设定	默认值	功能	相关标志和字
将时间设定为所有事件	默认: 循环时间的 10% 0.1 ~ 3,276.7ms	循环时间的 10%(或当循环时间的 10% 小于 0.1ms 时为 0.1ms)	设定将用于所有外设服务的最大时间。最大时间可设定为 0.1ms 或更大值。	---

9-3-8 FINS 保护



经由网络的 FINS 写保护设定

● 验证经由网络的 FINS 写保护

该参数决定是否对 FINS 命令使用经由网络的写保护。

该参数用于防止经由通过除串行连接之外其它连接方式所连接的网络意外擦写存储器。如果设定为允许该参数，则只能从“Nodes excluded from protection targets”（不作为保护对象的结点）参数所设定的结点进行写操作。

参数	设定	默认值	功能	相关标志和字
在网络中使用 FINS 写保护	ON: 禁止 FINS 写保护 OFF: 允许 FINS 写保护	OFF	允许或禁止 CPU 单元从经由网络发送的 FINS 命令执行写保护（即除串行连接之外的所有连接）。	---

● 不作为保护对象的结点

设定即使在允许写保护的情况下也可进行写操作的结点。

设定即使在允许写保护的情况下也可进行从其进行 FINS 写操作的结点和网络。将自动设定不作为写保护对象的结点总数。

最多可设定 32 个结点。如果未进行这些设定（即，如果结点总数为 0），则将对除本地结点之外的所有结点禁止写操作。

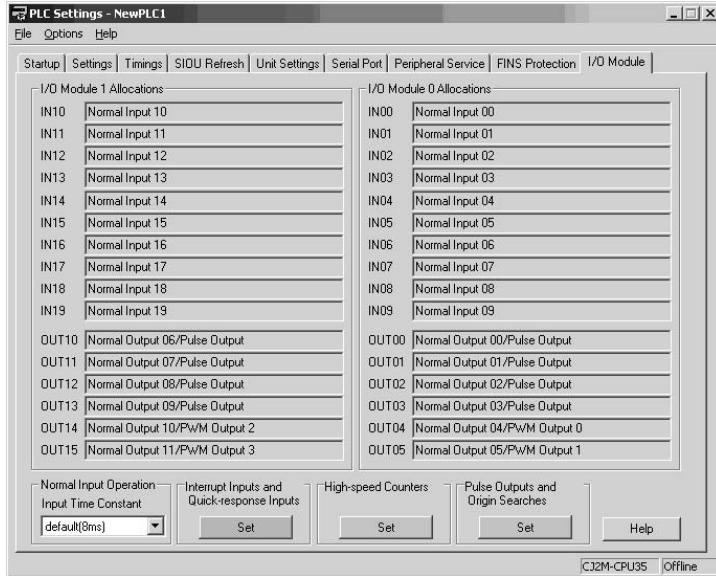
参数	设定	默认值	功能	相关标志和字
不作为保护对象的结点 *1	0 ~ 127	---	FINS 命令源网络地址	---
	1 ~ 255*2	---	FINS 命令源结点地址	---

*1 该设定仅在允许 FINS 写保护时有效。

*2 设定为 255(FF Hex) 时可包括指定网络中的所有结点。

9-3-9 I/O 模块

可将脉冲 I/O 模块用于 CJ2M CPU 单元。详情请参考《CJ2M CPU 单元脉冲 I/O 模块操作手册》(样本编号: W486)。



CPU 单元功能

本章节中阐述了 CPU 内置的各类功能。

有关 CJ2M-MD21□ 脉冲 I/O 模块与 CJ2M CPU 单元 (如高速计数器和脉冲输出) 连接时可使用的功能详情, 请参考《CJ2M CPU 单元脉冲 I/O 模块操作手册》(样本编号: W486)。

10-1 时钟功能	10-3
10-1-1 时钟功能	10-3
10-1-2 存储器中存储的时间数据	10-4
10-1-3 自由运行定时器	10-6
10-2 循环时间 / 高速处理	10-7
10-2-1 最小循环时间	10-7
10-2-2 最大循环时间	10-8
10-2-3 监视循环时间	10-9
10-2-4 高速输入	10-9
10-2-5 后台执行	10-10
10-2-6 高速中断功能	10-18
10-3 启动设定和维护	10-20
10-3-1 运行模式变更和启动时的保持设定	10-20
10-3-2 掉电检测延迟设定	10-22
10-3-3 禁止电源掉电中断	10-23
10-3-4 RUN 输出	10-24
10-3-5 启动时自动传送	10-25
10-4 单元管理功能	10-32
10-4-1 基本 I/O 单元管理	10-32
10-4-2 CPU 总线单元标志 / 位	10-34
10-4-3 高功能 I/O 单元标志 / 位	10-35
10-5 存储器管理功能	10-36
10-5-1 自动备份	10-36
10-5-2 EM 文件存储器功能	10-37
10-5-3 注释存储器	10-38
10-5-4 在运行过程中替换整个程序	10-39
10-6 安全功能	10-46
10-6-1 通过 DIP 开关实现写保护	10-46
10-6-2 通过密码实现读保护	10-46
10-6-3 使用产品批号实现程序操作保护	10-51
10-6-4 针对 FINS 命令的写保护	10-52
10-6-5 PLC 名称	10-56

10-7 调试	10-59
10-7-1 强制置位 / 复位	10-59
10-7-2 测试输入	10-60
10-7-3 微分监控	10-60
10-7-4 联机编辑	10-61
10-7-5 输出置 OFF	10-62
10-7-6 数据跟踪	10-63
10-7-7 存储出错停止位	10-70
10-7-8 故障报警指令	10-71
10-7-9 模拟系统错误	10-72
10-7-10 故障点检测	10-73
10-8 单元同步运转	10-75
10-8-1 概述	10-75
10-8-2 有关单元同步运转的详细信息	10-78
10-8-3 单元同步运转规格	10-81
10-8-4 同步数据刷新	10-82
10-8-5 单元同步运转的使用限制	10-86
10-8-6 应用步骤	10-88
10-8-7 PLC 设置	10-89
10-8-8 编写同步中断任务	10-91
10-8-9 与单元同步运转相关的调整和故障诊断作业	10-92

10-1 时钟功能

10-1-1 时钟功能

时钟数据

CJ2 CPU 单元内置时钟。

来自 CPU 单元时钟的时钟数据以 BCD 格式存储在下列辅助区相关位中。

名称	地址	描述
时钟数据	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

如果未连接电池或电池电压过低, 字 A351 ~ A354 中的时钟数据将会不稳定。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

通过与 CX-Programmer 进行联机和双击项目树中的 PLC 时钟图标或执行 DATE(735) 指令, 可对此数据进行设定。此外, 时钟数据还可执行递增 / 递减以及在秒和时之间进行转换。

● 时钟指令

指令	助记符	功能
时钟调整	DATE(735)	将内部时钟设定改为指定源字中的设定。
时→秒	SEC(065)	将以时 / 分 / 秒格式表示的时间数据转换成仅以秒表示的等值时间。
秒→时	HMS(066)	将以秒表示的时间数据转换成以时 / 分 / 秒格式表示的等值时间。
日历加	CADD(730)	在指定字的日历数据中增加时间。
日历减	CSUB(731)	在指定字的日历数据中减去时间。

10-1-2 存储器中存储的时间数据

辅助区字内可自动存储通电及断电时的时钟数据、电源中断次数、总通电时间、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。在未连接电池或电池电压过低的情况下,切勿使用存储在存储器中的时间数据。

通电时钟数据

下述辅助区字内存储了 PLC 的通电年份、月份、日期和时间。

名称	地址	描述
通电时钟数据 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)
通电时钟数据 2	A723 ~ A725	这些字中包含倒数第二次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 3	A726 ~ A728	这些字中包含倒数第三次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 4	A729 ~ A731	这些字中包含倒数第四次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 5	A732 ~ A734	这些字中包含倒数第五次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 6	A735 ~ A737	这些字中包含倒数第六次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 7	A738 ~ A740	这些字中包含倒数第七次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 8	A741 ~ A743	这些字中包含倒数第八次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 9	A744 ~ A746	这些字中包含倒数第九次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。
通电时钟数据 10	A747 ~ A749	这些字中包含倒数第十次通电时的启动时间 / 日期。 数据为 BCD 格式, 存储格式与字 A720 ~ A722 相同。

电源中断时间

下述辅助区字内存储了 PLC 的断电日期和时间以及电源中断次数。

名称	地址	描述
电源中断时间	A512 和 A513	这些字中包含电源中断时间 (BCD 格式)。每次断电时均更新该内容。 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	该字包含从电源第一次接通时起的断电次数 (二进制格式)。若要将该值复位, 请用 0000 覆盖当前值。

总通电时间

下述辅助区字内存储了 PLC 的总通电时间。

名称	地址	描述
总通电时间	A523	该字中包含 PLC 的总通电时间 (16 位二进制格式, 以 10 小时为增量单位)。若要将该值复位, 请用 0000 Hex 覆盖当前值。

用户程序和参数修改次数

这些辅助区字中存储了将数据写入用户程序或参数 (即 PLC 设置、I/O 表、路由表或 CPU 总线单元设置) 中的日期和时间。

名称	地址	描述
用户程序时间	A090 ~ A093	这些字中包含用户程序的最近覆写日期和时间 (BCD 格式)。 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 ~ 06) 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	这些字中包含参数的最近覆写日期和时间 (BCD 格式)。 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 ~ 06) 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

运行开始 / 结束时间

这些辅助区字将自动存储 PLC 运行开始和结束的日期和时间。

名称	地址	描述
运行开始时间	A515 ~ A517	此处以 BCD 格式存储将运行模式切换至 RUN 或 MONITOR 模式以开始运行的时间。 ^{*1} 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)
运行结束时间	A518 ~ A520	此处以 BCD 格式存储将运行模式切换至 PROGRAM 模式以停止运行的时间。 ^{*2} 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)

*1 上次运行开始的时间会在电源接通至开始运行的时间段内存储。

*2 如果在运行过程中发生错误, 则将存储出错时间。如果之后运行模式切换至 PROGRAM 模式, 则将存储进入 PROGRAM 模式的时间。

10-1-3 自由运行定时器

系统定时器在通电后的使用次数存储在辅助区中。这些定时器可用于计算时间间隔，而无需使用定时器指令。

名称	地址	描述
10ms 增量自由运行定时器	A0	当电源接通时将设为 0000 Hex，随后每 10ms 均会自动递增 1。当该值达到 FFFF Hex(655,350ms) 后将恢复为 0000 Hex，然后继续每 10ms 自动递增 1。
100ms 增量自由运行定时器	A1	当电源接通时将设为 0000 Hex，随后每 100ms 均会自动递增 1。当该值达到 FFFF Hex(6,553,500ms) 后将恢复为 0000 Hex，然后继续每 100ms 自动递增 1。
1s 增量自由运行定时器	A2	当电源接通时将设为 0000 Hex，随后每 1s 均会自动递增 1。当该值达到 FFFF Hex(65,535s) 后将恢复为 0000 Hex，然后继续每 1s 自动递增 1。

注 当运行模式切换为 RUN 模式时，将继续执行自动递增 1。

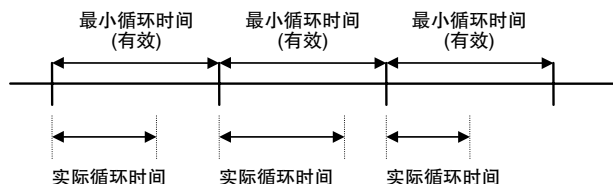
例如：通过计算 A0 中用于处理时间 A 和处理时间 B 之间的差值，即可得出处理 A 和处理 B 之间的时间间隔（以 10ms 为增量单位），而无需使用定时器指令。

10-2 循环时间 / 高速处理

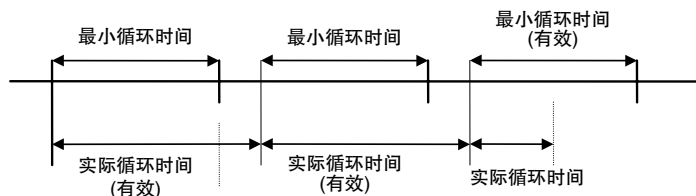
10-2-1 最小循环时间

PLC 中可以设定最小 (或固定) 循环时间。此外, 在固定的循环时间内重复运行程序可有效消除 I/O 响应时间的变化。

最小循环时间 (0.1 ~ 32,000ms) 在 PLC 设置中指定, 以 0.1ms 为单位。

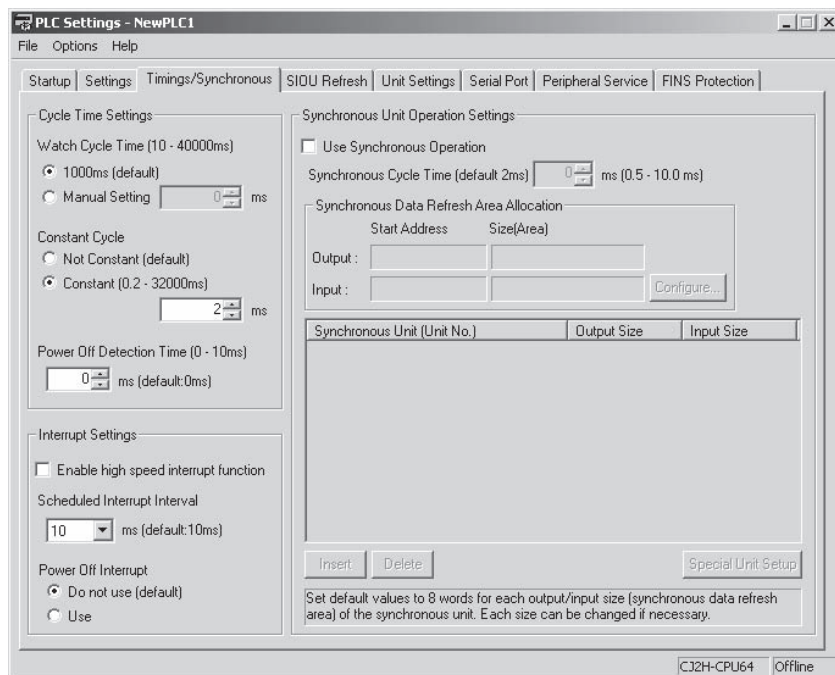


如果实际循环时间大于最小循环时间, 则最小循环时间功能将会失效, 且每个循环的循环时间都会发生变化。



● PLC 设置

使用 CX-Programmer 时, 可在 “Timings/Synchronous” (定时 / 同步) 选项页中进行相关设定。





附加信息

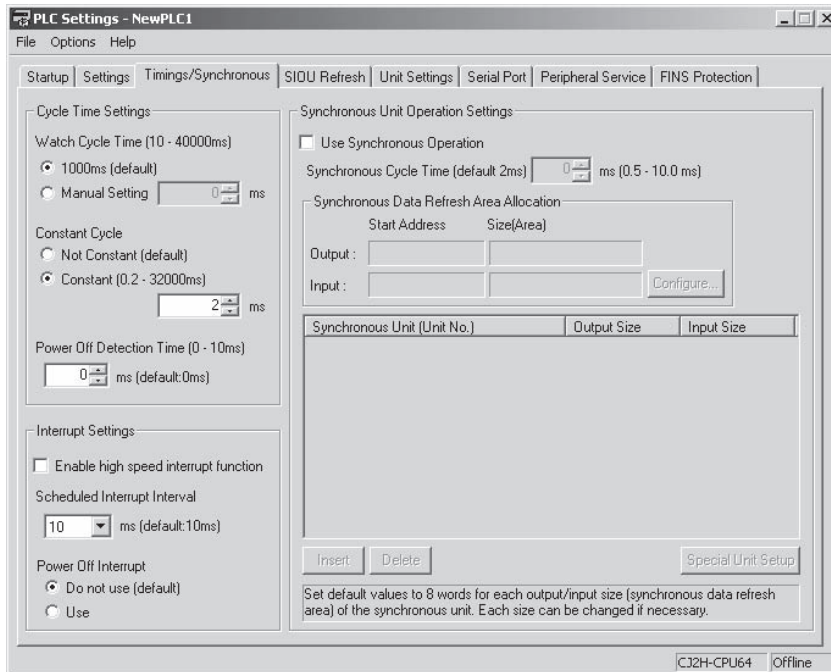
当 CPU 单元运行在 MONITOR 模式下时，可在 CX-Programmer 的 PLC 循环时间对话框中修改最小循环时间（恒定循环时间）。（仅 CJ2H CPU 单元（1.1 版或更高版本）和 CJ2M CPU 单元支持此项功能。）详情请参考“12-1 监控循环时间”。

10-2-2 最大循环时间

如果循环时间超过最大循环时间设定，CPU 单元将停止运行。循环时间超出标志（A401.08）将置 ON。默认最大循环时间为 1s。

● PLC 设置

使用 CX-Programmer 时，可在“Timings/Synchronous”（定时 / 同步）选项页的“Watch Cycle Time”（监视循环时间）选项中设定最大循环时间。



● 辅助区标志和字

名称	地址	描述
循环时间超出标志	A401.08	如果循环时间超过监视循环时间设定，A401.08将会置ON。

10-2-3 监视循环时间

对于每个循环，辅助区中的 A262 ~ A263 中用于存储最大循环时间，而 A264 ~ A265 和 A266 ~ A267 用于存储当前循环时间。

● 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	这些字用于在每个循环中存储以 0.1ms 为增量的 8 位十六进制最小循环时间，其范围如下： 0 ~ 429,496,729.5ms(0 ~ FFFF FFFF) 低位数存储在 A262 中，高位数存储在 A263 中。
当前循环时间 (以 0.1ms 为增量)	A264 和 A265	这些字用于在每个循环中存储以 0.1ms 为增量的 8 位十六进制当前循环时间，其范围如下： 0 ~ 429,496,729.5ms(0 ~ FFFF FFFF) 低位数存储在 A264 中，高位数存储在 A265 中。
当前循环时间 (以 0.01ms 为增量)	A266 和 A267	这些字用于在每个循环中存储以 0.01ms 为增量的 8 位十六进制当前循环时间，其范围如下： 0 ~ 42,949,672.95ms(0 ~ FFFF FFFF) 低位数存储在 A266 中，高位数存储在 A267 中。

CX-Programmer 可读取最近 8 个循环的平均循环时间。



附加信息

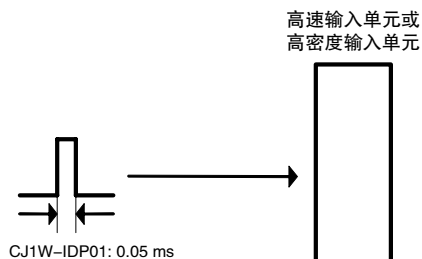
下述方法可有效缩短 CJ 系列 PLC 的循环时间：

- 将不执行的任务置于 WAIT 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

当要接收宽度小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) 为 0.05ms 的脉冲。



进行输入刷新时，输入内部存储器的输入状态将被清除。

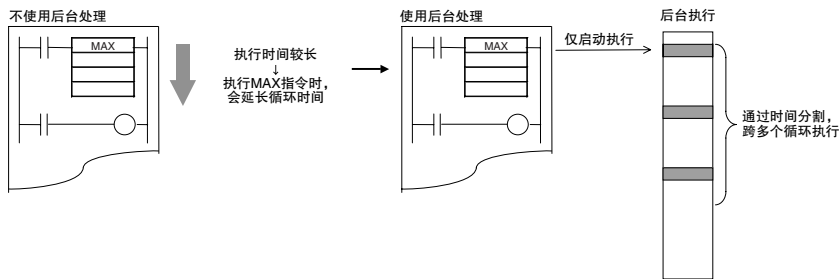
10-2-5 后台执行

由于执行表数据处理（如数据搜索）和文本串处理（如文本串搜索）需要耗用较长的时间，因此循环时间会因该时间的延长而出现较大波动。

后台执行（时间分割）功能可将下述指令分配到多个循环中执行，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）

将上述指令设定为后台执行，即可有效控制循环时间的暂时性延长。



正确使用注意事项

在 PLC 设置中启用高速中断的情况下，CJ2H CPU 单元的中断任务中将不会执行后台处理。此时将会发生指令处理错误。

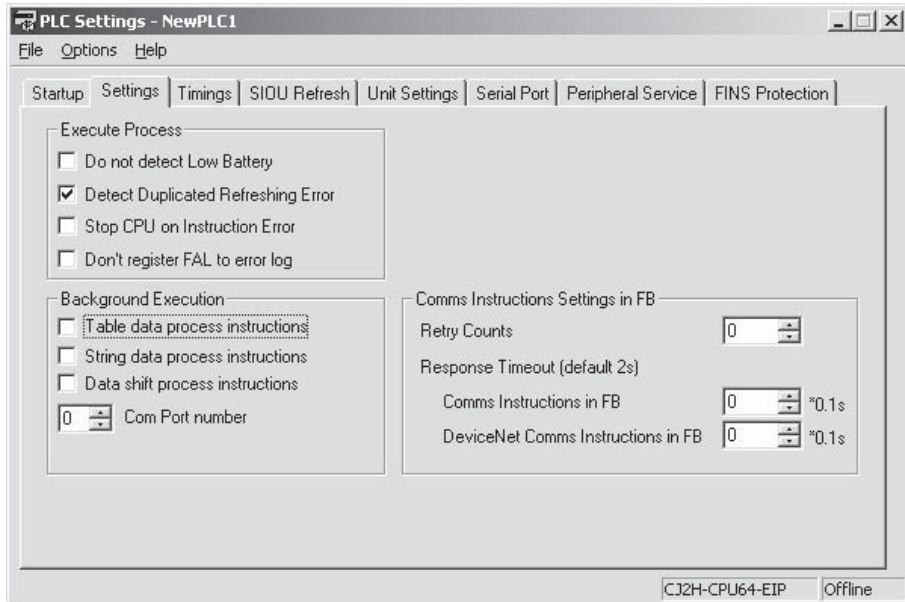
● 适用指令

当下列指令在功能块中使用，将不会执行后台处理，而将以正常的处理方式执行。

分组	指令	助记符
表数据处理指令	数据搜索	SRCH
	交换字节	SWAP
	寻找最大值	MAX
		MAXL
		MAXF
		MAXD
	寻找最小值	MIN
		MINL
		MINF
		MIND
求和	SUM	
帧校验和	FCS	
数据移位指令	异步移位寄存器	ASFT
文本串处理指令	传送串	MOV\$
	链接串	+\$
	取左串	LEFT\$
	取右串	RGHT\$
	取中间串	MID\$
	查找串	FIND\$
	计算串长度	LEN\$
	取代串	RPLC\$
	删除串	DEL\$
	交换串	XCHG\$
	清除串	CLR\$
插入串	INS\$	

● 步骤

1. 若要为所需指令启用后台执行功能，请在 CX-Programmer 的“PLC Settings” (PLC 设定) 对话框的“Settings” (设定) 选项页中进行后台执行设定。



2. 在 PLC 设置中设定用于后台执行的逻辑端口号。该端口号将用于所有在后台处理的指令。所有的后台执行均使用同一个端口。因此在执行一条指令的后台处理时，其它指令的后台处理将不会启动。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执行一条指令。
3. 如果执行某一条已指定后台执行的指令，则仅在满足执行条件且同一循环中无法完成指令执行的情况下，该指令才会在该循环中执行。
4. 开始后台执行时，端口对应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，端口对应的通信端口允许标志将置 ON，以允许在后台执行其它指令。

● 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。

输出至变址寄存器 (IR)

如果执行 MAX(182)、MAXL(174)、MAXF(176)、MAXD(178)、MIN(183)、MINL(175)、MINF(177) 或 MIND(179)，从而将包含最大或最小值字的 I/O 存储器映射地址输出到一个变址寄存器，则该地址将不会输出到变址寄存器而是输出到 A595 和 A596。

若要将地址存储在一个变址寄存器中 (与正常执行方式相同)，需使用数据传送指令 (如 MOVL(498)) 将 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处理指令的执行而更新。若要访问条件标志状态，应执行一条以同样方式影响条件标志的指令 (如下例所示)，随后再访问条件标志。

输出至变址寄存器 IR00

如果执行 SRCH(181)，从而将包含匹配值 (若存在多个，则选取第一个字) 的 I/O 存储器映射地址输出到一个变址寄存器，则该地址将不会输出到变址寄存器而是输出到 A595 和 A596。

输出至数据寄存器 (SRCH(181))

如果执行 SRCH(181)，从而将匹配数据输出到一个数据寄存器，则该数据将不会输出到数据寄存器而是输出到 A597。

匹配文本串

如果 SRCH(181) 找到匹配数据，则不会将等于标志置 ON，而是将 A598.01 置 ON。

指令错误

如果在后台进行处理的指令出现执行错误或非法访问错误，则不会将 ER 或 AER 标志置 ON，而是将 A395.10 置 ON。A395.10 将一直保持 ON 状态，直至在后台处理下一条指令。

输出至数据寄存器 (MAX(182) 或 MIN(183))

如果 MAX(182)、MAXL(174)、MAXF(176)、MAXD(178)、MIN(183)、MINL(175)、MINF(177) 或 MIND(179) 与指定为最小值或最大值输出字的数据寄存器 (DR0 ~ DR15) 一起执行，将会发生指令执行错误，且 ER 标志置 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	当以相应端口号执行网络指令或后台执行功能时置 ON。位 00 ~ 07 分别对应通信端口 0 ~ 7。 如果通过简易备份操作功能对存储卡进行写或比较操作时，将自动分配通信端口，且相应标志在操作期间置 ON，在操作完成后置 OFF。
通信端口出错标志	A219.00 ~ A219.07	当网络指令执行期间发生错误时置 ON，在正常完成时置 OFF。位 00 ~ 07 分别对应通信端口 0 ~ 7。 如果通过简易备份操作功能对存储卡进行写或比较操作时，将自动分配通信端口。如果发生错误，相应标志置 ON；如果简易备份操作正常结束，相应标志置 OFF。
通信端口完成码	A203 ~ A210	这些字中包含网络指令执行完毕后对应端口的完成代码。后台执行完成后，相关内容将被清为 0000 Hex。字 A203 ~ A210 分别对应通信端口 0 ~ 7。 如果通过简易备份操作功能对存储卡进行写或比较操作时，将自动分配通信端口，且相关字中将存储一个完成码。
后台执行的 ER/AER 标志	A395.10	指令在后台执行的过程中发生执行错误或非法访问错误时置 ON。当通电或开始运行时置 OFF。
后台执行的 IR00 输出	A595 和 A596	将后台执行指令的输出指定用于变址寄存器时，这些字用于接收输出，而 IR00 将不会接收到输出。 范围：0000 0000 ~ FFFF FFFF Hex 低四位：A595；高四位：A596
后台执行的 DR00 输出	A597	将后台执行指令的输出指定用于数据寄存器时，该字用于接收输出，而 DR00 将不会接收到输出。 范围：0000 ~ FFFF Hex
后台执行的等于标志输出	A598.01	当后台执行的 SRCH(181) 指令找到了匹配的数据时，该标志置 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：

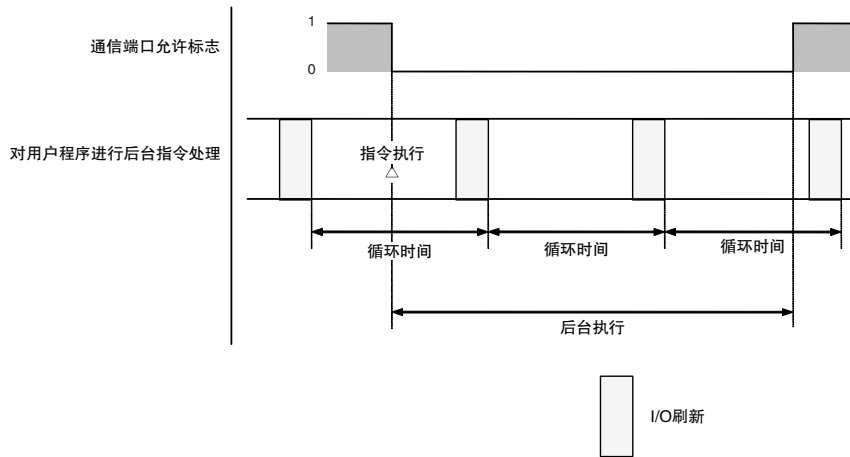
- SEND(090)、RECV(098) 和 CMND(490)(网络指令)
- PMCR(260)(协议宏)
- TXDU(256) 和 RXDU(255)(串行通信单元所使用的无协议通信)

后台指令和上述指令不可在同一端口上同时执行。请使用通信端口允许标志，以确保在任一时间每个端口仅执行一种指令。

注 当某个端口的通信端口允许标志置 OFF 时，如果指定某条指令在该端口进行后台执行，则 ER 标志将置 ON，且不执行后台指令。

● 通信端口允许标志

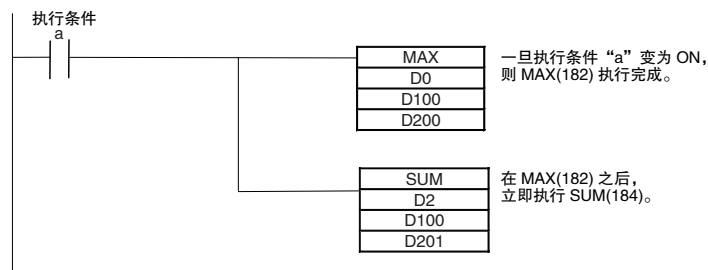
当未在端口上执行处理时，通信端口允许标志为 ON；当正在端口上执行处理时，通信端口允许标志为 OFF。



编程示例 1

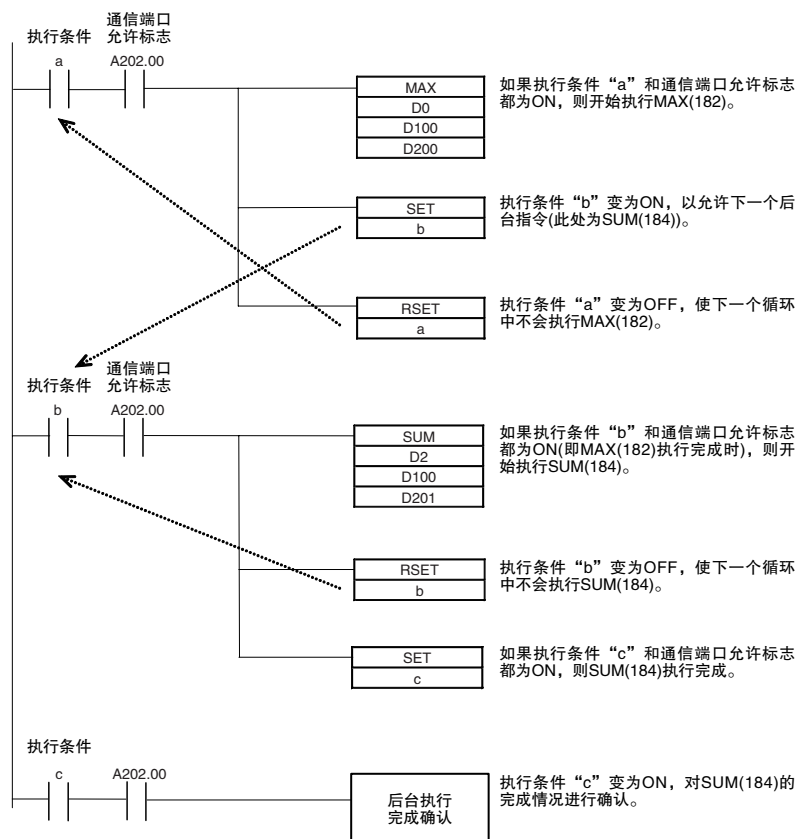
● 无后台执行的编程

如下所示，在指令执行后完成处理。



● 带后台执行的编程

若使用后台执行功能，则应对程序进行修改，从而使 MAX(182) 仅在特定的通信端口允许标志置 ON(即仅在该端口未被用于后台执行或网络通信) 时执行。此外，还可通过 SET 和 RSET 指令控制输入状态，确保以正确的次序执行处理。(在下例中，通信端口 0 用于后台执行。)

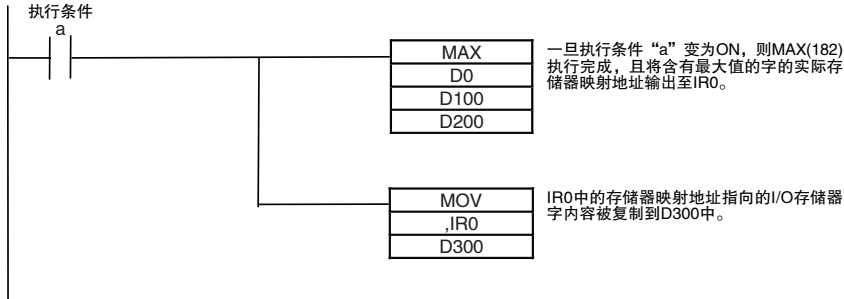


编程示例 2

本例所示为指定变址寄存器输出时的后台执行情形 (适用于 MAX(182)、MIN(183) 和 SRCH (181))。

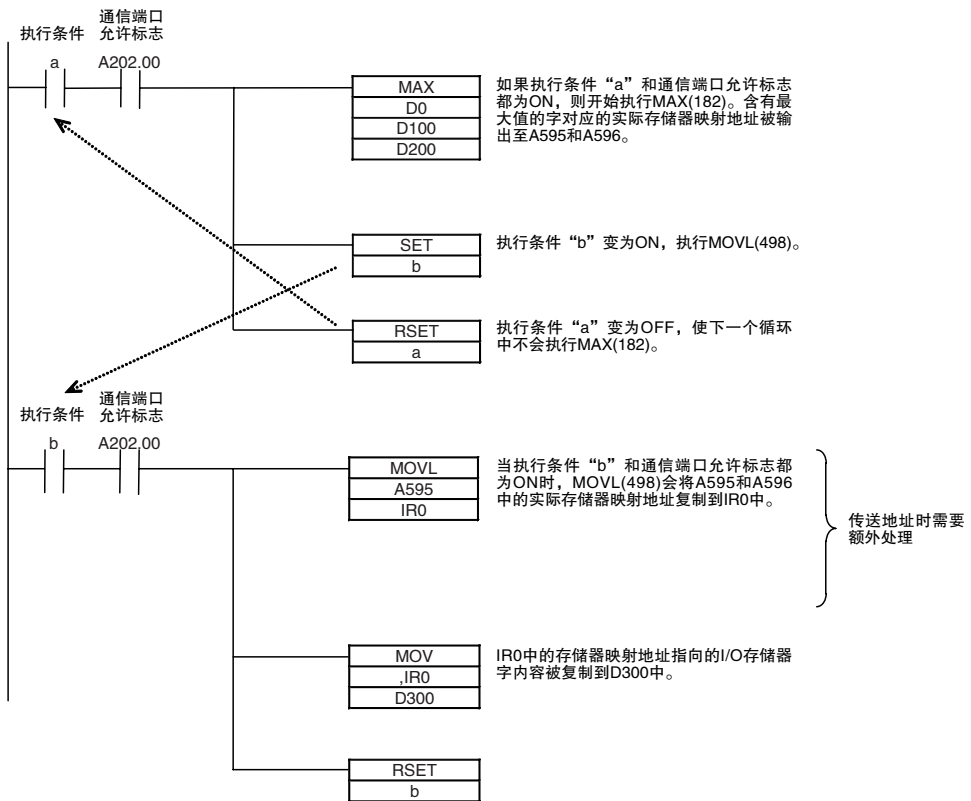
● 无后台执行的编程

如下所示, 包含最大值或最小值的字的 PLC 存储器映射地址被输出至变址寄存器。



● 带后台执行的编程

使用后台执行功能时, 包含最大值或最小值的字的 PLC 存储器映射地址首先被输出至 A595 和 A596, 然后再通过 MOVL(498) 指令将 PLC 存储器映射地址传送至变址寄存器。

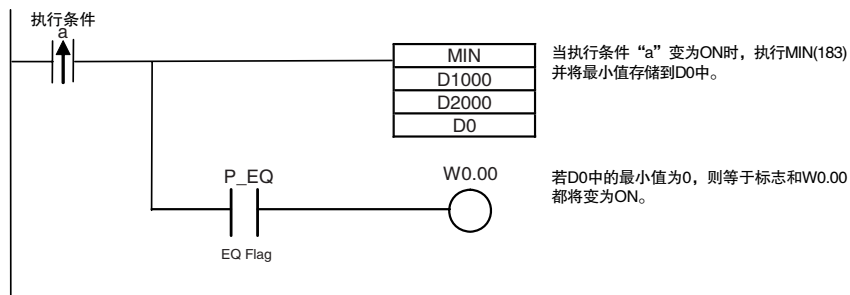


编程示例 3

本例所示为参考条件标志时的后台执行情形。

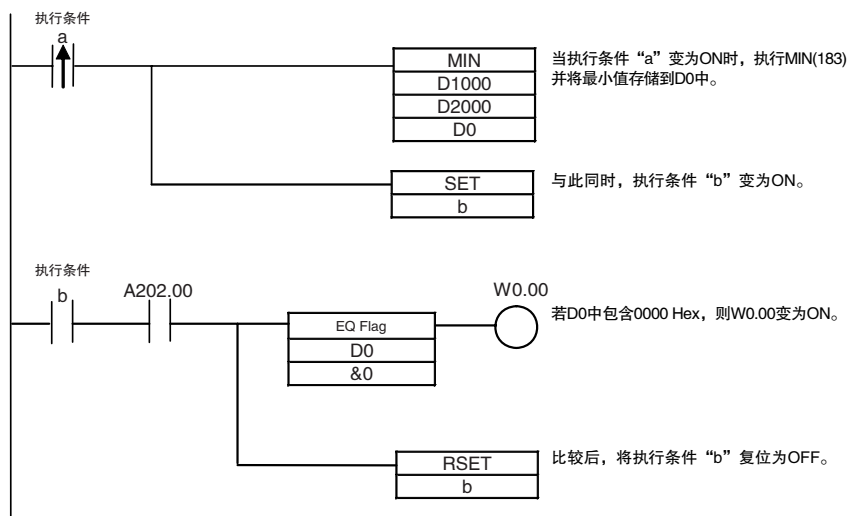
● 无后台执行的编程

为确认 MIN(183) 找到的最小值是否为 0，将在 MIN(183) 执行完成后检查等于标志的状态。



● 带后台执行的编程

如下图所示，使用 AND=(300) 指令确认 MIN(183) 找到的最小值是否为 0。



10-2-6 高速中断功能

高速中断功能可提升中断任务在某些限制条件下的执行效能 (1.1 版或更高版本)。

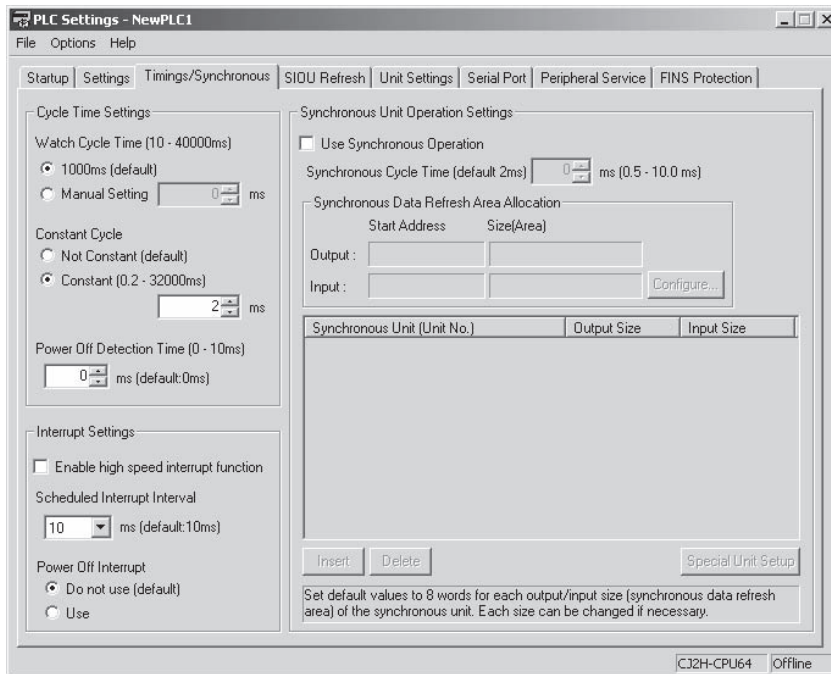
缩短中断开销处理时间

执行 I/O 中断任务、外部中断任务或定时中断任务时，请按照下列步骤缩短开销处理时间（即中断任务启动时间 + 循环任务返回时间）。

中断任务类型	中断开销处理时间	
	高速中断功能启用	高速中断功能禁止 (默认)
I/O 中断任务和外部中断任务	25 μ s (中断任务启动时间 17 μ s+ 循环任务返回时间 8 μ s)	37 μ s (中断任务启动时间 26 μ s+ 循环任务返回时间 11 μ s)
定时中断任务	21 μ s (中断任务启动时间 13 μ s+ 循环任务返回时间 8 μ s)	33 μ s (中断任务启动时间 22 μ s+ 循环任务返回时间 11 μ s)

● PLC 设置

使用 CX-Programmer 时，可在“Timings/Synchronous”（定时 / 同步）选项页中勾选“Enable high-speed interrupt function”（启用高速中断功能）复选框。



● 启用高速中断功能时的限制条件

启用高速中断功能时，存在下列限制条件。

- 下列指令 (如网络通信指令) 无法在中断任务中使用。

助记符	指令名称
EMBC(281)	选择 EM 存储区块
SEND(090)*	网络发送
RECV(098)*	网络接收
CMND(490)*	发布命令
PMCR(260)*	协议宏
TXDU(256)	通过串行通信单元发送
RXDU(255)	通过串行通信单元接收数据
EXPLT(720)、EGATR(721)、ESATR(722)、ECHR(723) 和 ECHWR(724)	显式信息发送 (EXPLT(720))、显式获取属性 (EGATR(721))、显式设定属性 (ESATR(722))、显式字读取 (ECHR(723)) 和显式字写入 (ECHWR(724))

* 启用高速中断功能后，即可在中断任务中使用 SEND2(491)、RECV2(492)、CMND2(493) 和 PMCR2(264)。

- 后台执行指令 (即表数据处理指令、文本串处理指令和数据移位指令) 无法在中断任务中使用。
- 下列辅助区字中的数据将变为无效：A440(中断任务最长处理时间) 和 A441(处理时间最长的中断任务)。
- 如下表所示，在启用高速中断功能时，内置 EtherNet/IP 端口和 EtherNet/IP 单元的刷新性能将会降低。

	高速中断功能启用	正常运行
开销处理时间(使用数据链接时需加入的时间)	100 μ s+ 传送字数 \times 0.87 μ s	100 μ s+ 传送字数 \times 0.33 μ s
EtherNet/IP 的 DLNK(226) 指令执行时间 (CPU 总线单元的即时 I/O 刷新)	910 μ s/1,000 字	430 μ s/1,000 字

定时中断任务的最小时间间隔

通过 MSKS(690) 可为定时中断 0(中断任务 2) 设定 0.1ms 的最小时间间隔。定时中断 1 无此项特性。

● 为定时中断 0 设定 0.1ms 最小时间间隔所需的条件

- 必须启用高速中断功能。(在 PLC 设置中勾选 “Enable high-speed interrupt function” (启用高速中断功能) 复选框。)

如果高速中断功能处于禁用状态，则使用 MSKS(690) 指定 0.1ms 最小时间间隔的操作会导致指令处理错误。

- 不得将设备直接连接至 CPU 单元的外设端口 (USB) 或串行端口。如果将支持软件 (如 CX-Programmer) 或 NS 系列可编程终端直接连接至 CPU 单元的外设端口 (USB) 或串行端口，则 0.1ms 时间间隔可能会变得不稳定。如果需要连接，请使用内置 EtherNet/IP 端口 (仅限 CJ2H-CPU6□-EIP) 或者通过 EtherNet/IP 单元或串行通信单元的通信端口来实现。



正确使用注意事项

高速中断功能可提升中断任务在某些限制条件下的执行效能。尝试运行前请务必确认所需的使用条件。应当注意的是，仅在支持软件 (如 CX-Programmer) 或 NS 系列可编程终端未直接连接到 CPU 单元的外设端口 (USB) 或串行端口的条件下，才可在定时中断的最小时间间隔被设定为 0.1ms 的情况下执行运转。如果采用直接连接方式，则可能无法以 0.1ms 的时间间隔执行运转。如果由于维护等其它原因必须将支持软件直接连接至 CPU 单元，则应确认未以 0.1ms 时间间隔执行定时中断的情况下，相关设备不会受到影响。

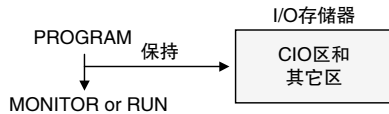
10-3 启动设定和维护

10-3-1 运行模式变更和启动时的保持设定

运行模式变更

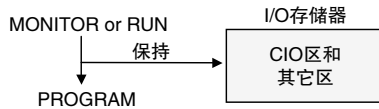
● 启动程序执行

当CPU单元从PROGRAM模式切换为RUN/MONITOR模式来启动程序执行，只需将IOM保持位(A500.12)置 ON 即可保持 I/O 存储器中的所有数据。



● 停止程序执行

在 IOM 保持位 (A500.12) 为 ON 的情况下，若将 CPU 单元从 RUN/MONITOR 模式切换为 PROGRAM 模式来停止程序执行，同样能保持 I/O 存储器中的所有数据。



IOM 保持位可保持下列区域中的数据：CIO 区 (I/O 区、数据链接区、CPU 总线单元区、高功能 I/O 单元区、DeviceNet 区和内部 I/O 区)、工作区、定时器完成标志和定时器当前值 (PV)。

● 辅助区标志和字

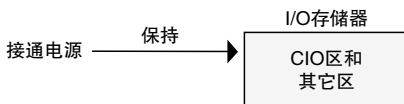
名称	地址	描述
IOM 保持位	A500.12	该位为 ON 时，可在运行模式在 PROGRAM 和 RUN/MONITOR 之间切换或通电时保持 I/O 存储器中的所有数据。 ON：变更运行模式时，I/O 存储器中的数据将被保持。 OFF：变更运行模式时，I/O 存储器中的数据将被清除。

在 IOM 保持位为 ON 的情况下，输出单元的所有输出状态均将在停止程序执行时保持。当程序再次开始运行时，输出状态将与程序停止运行前保持一致。

(在 IOM 保持位为 OFF 的情况下，指令将在所有输出被清零后执行。)

PLC 通电

为确保 PLC 通电时能保持 I/O 存储器中的所有数据，IOM 存储位 (A500.12) 必须为 ON 状态，且应在 PLC 设置中进行保护性设定。

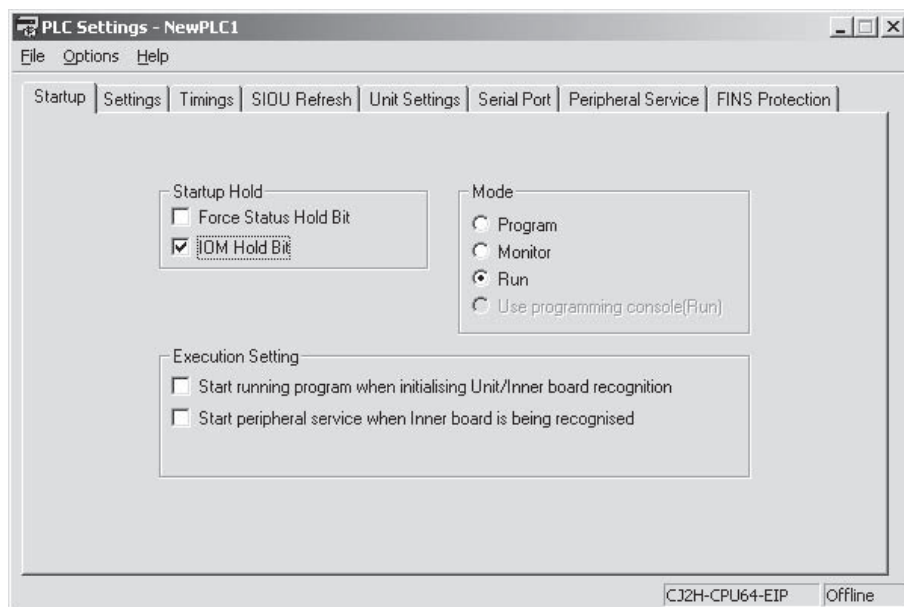


● 辅助区标志和字

名称	地址	描述
IOM 保持位	A500.12	该位为 ON 时，可在运行模式在 PROGRAM 和 RUN/MONITOR 之间切换或通电时保持 I/O 存储器中的所有数据。 ON：变更运行模式时，I/O 存储器中的数据将被保持。 OFF：变更运行模式时，I/O 存储器中的数据将被清除。

● PLC 设置

使用 CX-Programmer 时，请在“Startup”（启动）选项页下方的“Startup Hold”（启动保持）区中勾选“IOM Hold Bit”（IOM 保持位）复选框来完成设定。

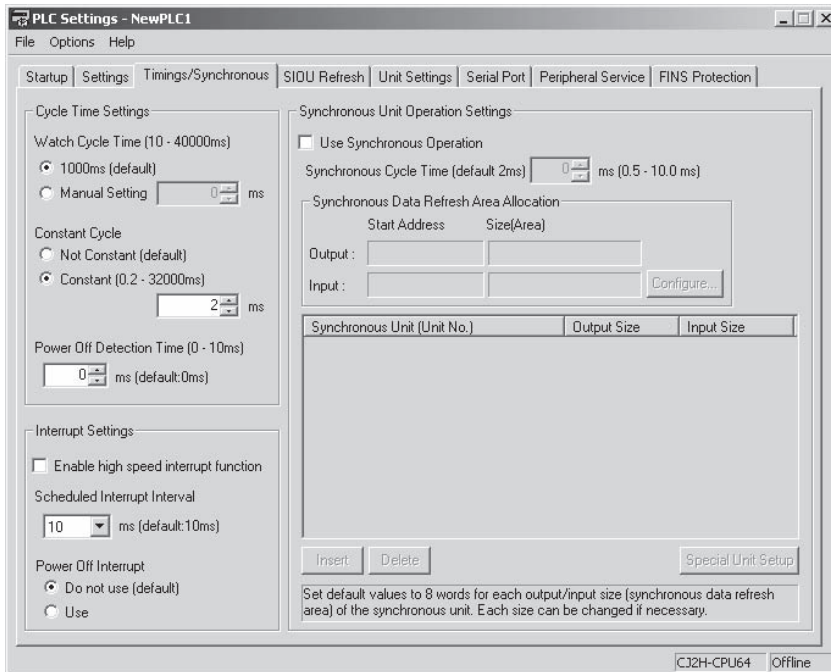


10-3-2 掉电检测延迟设定

默认情况下，设备可在电源电压降至最小额定值的 85%(DC 电源为 80%) 以下后约 10 ~ 25ms(DC 电源为 2 ~ 5ms) 内检测到 10ms 或更长时间的 AC 电源中断。PLC 设置中存在一项延长该时间的设定。在启用电源 OFF 中断任务的情况下，将在确认电源中断时执行该设定，否则 CPU 单元将复位并停止运行。

PLC 设置

使用 CX-Programmer 时，可在“Timings/Synchronous”（定时 / 同步）选项页的“Power Off detection time”（掉电检测时间）选项中设定。



附加信息

如果不想使用电源 OFF 中断任务，可将掉电检测时间设定为 10ms 或以下来应对电源中断事件。但在使用 CJ1W-PD022 电源单元的情况下，则无法设定延迟，因此请采用默认设定 0ms。

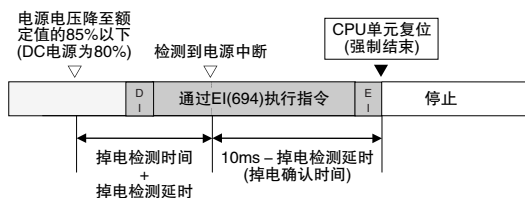
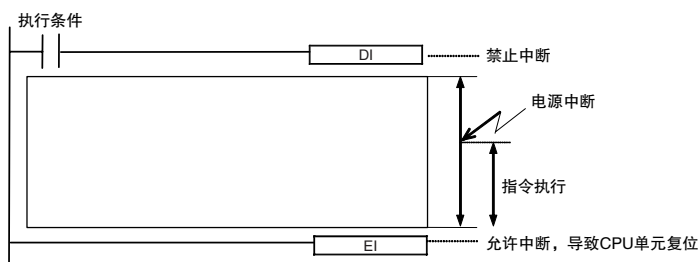
10-3-3 禁止电源掉电中断

可对程序区域采取电源掉电中断保护措施，即使电源中断，程序在 CPU 单元复位前仍可运行。该功能可通过禁止中断 (DI(693)) 和允许中断 (EI(694)) 指令实现。

该功能适用于必须成组执行的指令，以便在下一通电时不会使用中间存储数据开始执行操作。

步骤

1. 在 A530 ~ A5A5 Hex 中设置电源 OFF 中断禁止设定，以启用电源掉电中断禁止功能。
2. 在 PLC 设置中启用电源掉电中断禁止功能 (这是默认设定)。
3. 在需要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) 指令来启用中断。在 DI(693) 和 EI(694) 之间执行指令时，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。



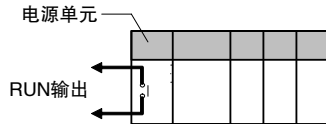
● 辅助区标志和字

名称	地址	含义
禁止电源掉电中断设定	A530	执行 EI(694) 前，允许使用 DI(693) 禁止电源掉电中断处理 (除执行电源掉电中断任务之外)。 A5A5 Hex: 允许使用 DI(693) 禁止电源掉电中断处理。 任何其它值: 不允许使用 DI(693) 禁止电源掉电中断处理。

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 模式下运行时，该输出点为 ON(闭合)。



名称	动作
RUN 输出	MONITOR 或 RUN 模式: ON(闭合)
	PROGRAM 模式: OFF(断开)

使用 RUN 输出可建立一个外部安全电路(如紧急停电路),以防止输出单元的外部电源在 PLC 不运行的情况下供电。

注 若使用不带 RUN 输出的电源单元,可对常 ON 标志(A1)进行编程,以作为输出单元的输出点执行条件,从而建立一个(与 RUN 输出)等效的输出。



安全使用注意事项

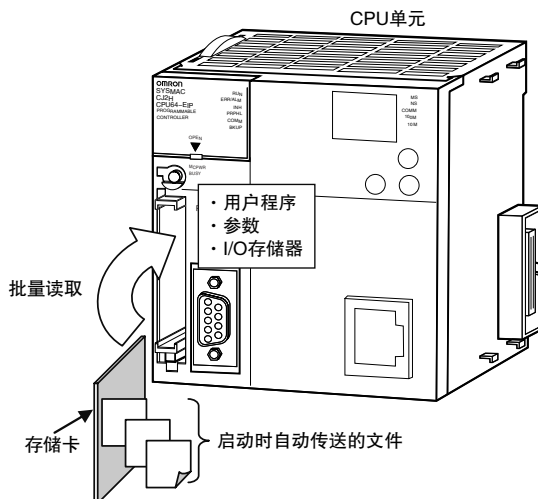
如果输出单元的外部电源在 PLC 电源之前接通,则在 PLC 电源接通的瞬间,输出单元可能会出现误动作。为了防止出现任何误动作,应增设一个外部电路以避免输出单元的外部电源先于 PLC 电源本身接通。因此请务必设计一个符合上述要求的故障安全电路,以确保仅当 PLC 在 RUN 或 MONITOR 模式下运行时才通过外部电源供电。

10-3-5 启动时自动传送

概述

当电源接通时,可使用启动时的自动传送功能将存储卡中的用户程序,参数和 I/O 存储器数据读取至 CPU 单元。

在 CX-Programmer 的“Memory Card”(存储卡)窗口中可创建启动时需要自动传送的文件,并存储在存储卡中。



该功能无法用于读取 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。因此该功能可确保始终输入正确的值。

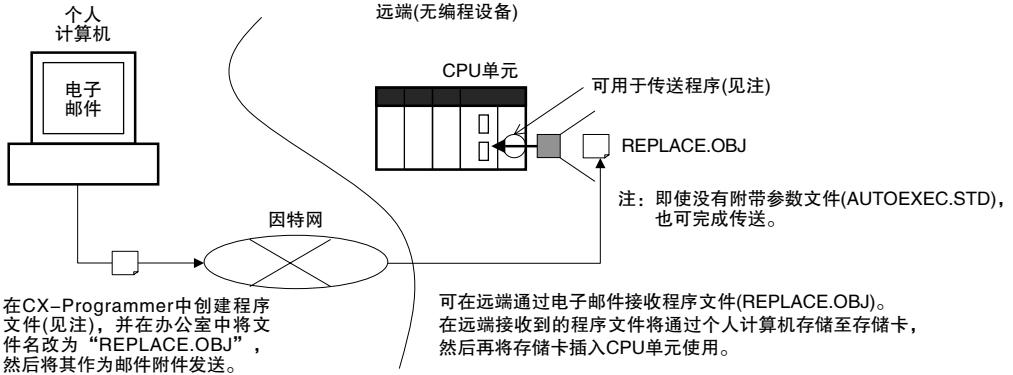
启动时自动传送的类型

启动时自动传送功能具备两种执行方式:带参数文件或不带参数文件(AUTOEXEC.STD)传送。创建的文件名取决于所使用的传送方式。

启动时自动传送的类型	文件名称	应用
带参数文件传送	AUTOEXEC 或 ATEXC□□	覆写程序和网络设定
不带参数文件传送	REPLACE 或 REPLC□□	覆写程序

● 自动传送文件 (不带参数文件) 的应用示例

在不使用参数文件 (.STD) 的情况下, 用户可在办公室中离线创建 (即无需连接实际设备) 程序 / 网络符号文件 (.OBJ) 并将其传送至一个远端位置。而远端的存储卡用于存储程序 / 网络符号 (变量) 文件, 并在启动时自动将程序传送至 CPU 单元, 期间无需使用编程设备。

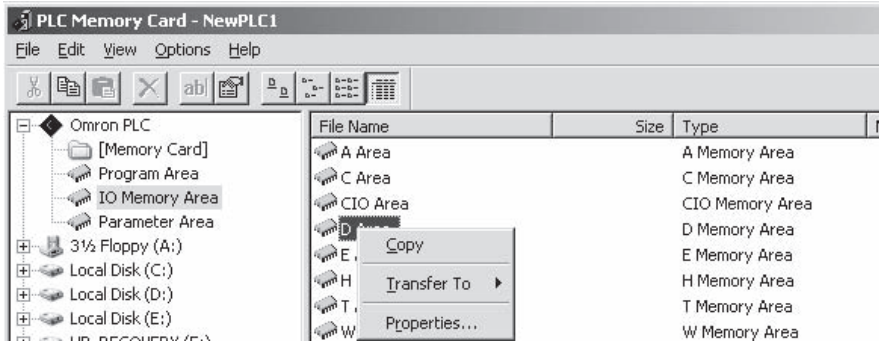


步骤

● 创建并传送启动时自动传送的文件

通过 CX-Programmer 创建启动时自动传送的文件, 并将其传送到 CPU 单元内置的存储卡中。

- 程序区和参数区:
右击并选择 “Transfer to” (传送至)- “Memory Card” (存储卡)。输入启动时自动传送的文件名, 并执行创建和传送操作。
- I/O 存储区:
右击右侧窗口中的 D 区或 E 区。输入启动时自动传送的文件名, 并执行创建和传送操作。



下表所示的文件名用于启动时的自动传送操作。

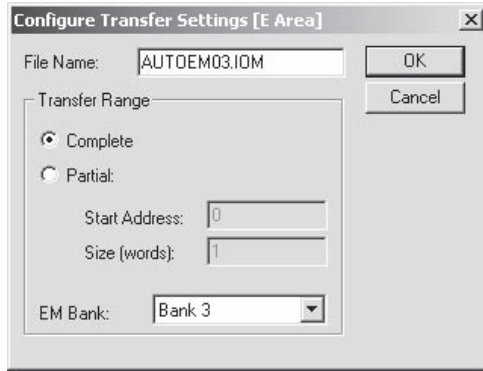
文件类型	内容	启动时自动传送 (带参数文件)	是否需要?	启动时自动传送 (不带参数文件)	是否需要?
程序/网络符号文件	用户程序	AUTOEXEC.OBJ	是	REPLACE.OBJ	是
参数文件	<ul style="list-style-type: none"> · PLC 名称 · PLC 设置 · 注册 I/O 表 · 路由表 · CPU 总线单元设置区 	AUTOEXEC.STD	是	无	---
数据文件	为高性能 I/O 单元和 CPU 总线单元 (D20000 ~ D32767) 分配的 DM 区字	AUTOEXEC.IOM	---	REPLACE.IOM	---
	DM 区 (D0 ~ D19999)	ATEXEC.DM.IOM	---	REPLAC.DM.IOM	---
	EM 区	AUTOEM□□.IOM (□□: 00 ~ 18 Hex)	---	RPLCEM□□.IOM (□□: 00 ~ 18 Hex)	---

例如，为 DM 区指定下列起始地址和区域大小。

- 为高性能 I/O 单元和 CPU 总线单元分配的 DM 区字：

- DM 区：

· EM 区：



● 执行启动时自动传送功能

1. 关闭 PLC 电源。

2. 将 CPU 单元前部的 DIP 开关的第 2 位设定为 ON。

当开关位 7 设定为 ON 时，将会启用简易备份功能，因此请务必将其设定为 OFF。

3. 将存有所需文件的存储卡插入 CPU 单元。

4. 接通 PLC 电源。

此时将开始启动时自动传送操作，且 BUSY 指示灯闪烁黄光。传送完成后，BUSY 指示灯熄灭。如果启动时自动传送操作失败，则说明发生了存储卡传送错误，CPU 单元将会停止运行。

● CPU 单元前部的 DIP 开关

开关位编号	名称	设定
2	启动时自动传送	ON: 执行启动时自动传送操作 OFF: 不执行启动时自动传送操作

● 辅助区标志和字

名称	地址	描述
存储卡传输出错标志 (致命错误)	A401.03	通过存储卡执行启动时自动传送的操作失败时置 ON。CPU 单元将停止运行且 CPU 单元前部上的 ERR/ALM 指示灯将点亮。

启动时自动传送的数据文件

下表所示的六种数据文件用于启动时的自动传送操作。

启动时自动传送的数据文件		内容	范围	起始地址	大小
启动时自动传送 (带参数文件)	启动时自动传送 (不带参数文件)				
AUTOEXEC.IOM	REPLACE.IOM	为高性能 I/O 单元和 CPU 总线单元 (D20000 ~ D32767) 分配的 DM 区字	部分区域	20000	12,768 字
ATEXEC.DM.IOM	REPLAC.DM.IOM	DM 区 (D00000 ~ D19999)	整个区域	0	20,000 字
AUTOEM□□.IOM (□□: 00 ~ 18)	AUTOEM□□.IOM (□□: 00 ~ 18)	EM	整个区域	---	---

创建用于启动时自动传送操作的数据文件时，请务必先设定上表中的各类数据区和起始地址。而存储大小的设定应与相应数据区的结束地址吻合。

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。如果在整个区域中没有足够的字，则将不会变更区域中剩余的字。

启动时自动传送 (带参数文件)

请使用下列文件名。在执行启动时自动传送操作时,存储卡中必须存有以下表最右栏中标有“是”的文件。

文件类型	文件名称	扩展名	内容	描述	是否需要?
程序 / 网络符号文件	AUTOEXEC	.OBJ	用户程序和 网络符号*1	<ul style="list-style-type: none"> 循环任务和中断任务中的程序 在执行启动时自动传送操作时,存储卡中必须存有该文件。 在执行启动时自动传送操作时,存储卡中也必须存有参数文件(AUTOEXEC.STD)。 	是
参数文件	AUTOEXEC	.STD	<ul style="list-style-type: none"> PLC 名称 PLC 设置 I/O 表 路由表 CPU总线单元设置区 	<ul style="list-style-type: none"> CPU 单元的参数区数据 电源接通时,参数将自动存储至CPU单元中的指定位置。 用户无需在文件中单独指定数据。 当程序/网络符号文件为 AUTOEXEC.OBJ 时,存储卡中必须存有该参数文件,以执行启动时自动传送操作。 	是
数据文件	AUTOEXEC	.IOM	DM 区数据 (包含从 D20000 开始的指定字数的数据)	<ul style="list-style-type: none"> 从 D20000 开始将 DM 区数据存储在 AUTOEXEC.IOM 文件中。 启动时,从 D20000 开始传送指定文件中的所有数据。如果在整个区域中没有足够的字,则将不会变更区域中剩余的字。 在执行启动时自动传送操作时,无需使用该文件。 	---
	ATEXECDM	.IOM	DM 区数据 (包含从 D0 开始的指定字数的数据)	<ul style="list-style-type: none"> 从 D0 开始将 DM 区数据存储在 ATEXECDM.IOM 文件中。 启动时,从 D0 开始传送指定文件中的所有数据。如果在整个区域中没有足够的字,则将不会变更区域中剩余的字。 在执行启动时自动传送操作时,无需使用该文件。*2 	---

*1 仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 支持网络符号。如果将符号文件 (.OBJ) 传送至 CJ2H-CPU6□ 或 CJ2M-CPU3□, A401.03 (存储卡传输出错标志) 将会置 ON。

*2 如果 ATEXECDM.IOM 和 AUTOEXEC.IOM 文件内容重叠,则 ATEXECDM.IOM 具备更高的写入优先级。

文件类型	文件名称	扩展名	内容	描述	是否需要?
数据文件 (续上)	AUTOEM□□	.IOM	存储区 E□□ 的 EM 区数据 (包含从 E□□_0 开始的指定字数的数据)	<ul style="list-style-type: none"> 将存储区 E□□ 的 EM 区数据 (从 E□□_0 开始) 存储至 AUTOEM□□.IOM 文件。(方框表示存储区 E□□_0 ~ C。存储区的最大数量取决于 CPU 单元的型号。) 启动时,将从文件的起始位置开始传送所有的数据 (从 E□□_0 开始)。如果在整个区域中没有足够的字,则将不会变更区域中剩余的字。 在执行启动时自动传送操作时,无需使用该文件。*2 	---

如果 ATEXECDM.IOM 和 AUTOEXEC.IOM 文件内容重叠,则前者具备更高的写入优先级。(因为文件的载入顺序为: AUTOEXEC.IOM 在前, ATEXECDM.IOM 在后。)



附加信息

启动时自动传送功能可与利用辅助区控制位替换整个程序的功能配合使用。启动时自动传送功能可实现以一个程序替换另一个程序的操作,并可在运行期间使用辅助区控制位替换程序。

启动时自动传送 (不带参数文件)

请使用下列文件名。在执行启动时自动传送操作时,存储卡中必须存有下列表最右栏中标有“是”的文件。

文件类型	文件名称	扩展名	内容	描述	是否需要?
程序 / 网络符号文件	REPLACE	.OBJ	用户程序	<ul style="list-style-type: none"> 文件内容与 AUTOEXEC.OBJ 相同。 即使存储卡中不带参数文件 (AUTOEXEC.STD), 仍可在启动时传送程序 / 网络符号文件。 	是
参数文件	不需要	---	---	<ul style="list-style-type: none"> 无论存储卡中使用何种文件名, 都不会传送该参数文件。 	---
数据文件	REPLACE	.IOM	DM 区数据 (包含从 D20000 开始的指定字数的数据)	<ul style="list-style-type: none"> 文件内容与 AUTOEXEC.IOM 相同。 如果存储卡中存有程序 / 网络符号文件 (REPLACE.OBJ), 将在启动时传送该文件中的数据。 	---
	REPLCDM	.IOM	DM 区数据 (包含从 D0 开始的指定字数的数据)	<ul style="list-style-type: none"> 文件内容与 ATEXECDM.IOM 相同。 如果存储卡中存有程序 / 网络符号文件 (REPLACE.OBJ), 将在启动时传送该文件中的数据。 	---
	RPLCEM□□	.IOM	组 □□ 的 EM 区数据 (包含从 E□□_0 开始的指定字数的数据)	<ul style="list-style-type: none"> 文件内容与 AUTOEM□□.IOM 相同。 如果存储卡中存有程序 / 网络符号文件 (REPLACE.OBJ), 将在启动时传送该文件中的数据。 	---



附加信息

将分配给高性能 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) 中的数据存储在存储卡中 (文件名为 AUTOEXEC.IOM), 则在启动时可从存储卡读取高性能 I/O 单元和 CPU 总线单元中的设置数据。通过上述方式使用存储卡可有效管理各设备的高性能 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的大致时间

用户程序和网络符号的大小	自动传送所需的时间 (从电源接通到运行开始)
用户程序: 120K 步	9s
用户程序: 120K 步, 网络符号: 20,000 个	59s

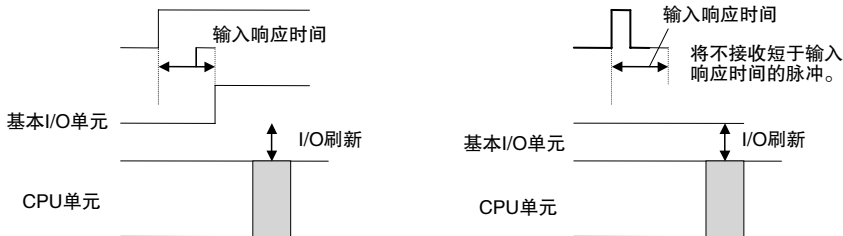
10-4 单元管理功能

10-4-1 基本 I/O 单元管理

● 设定基本 I/O 单元的输入响应时间

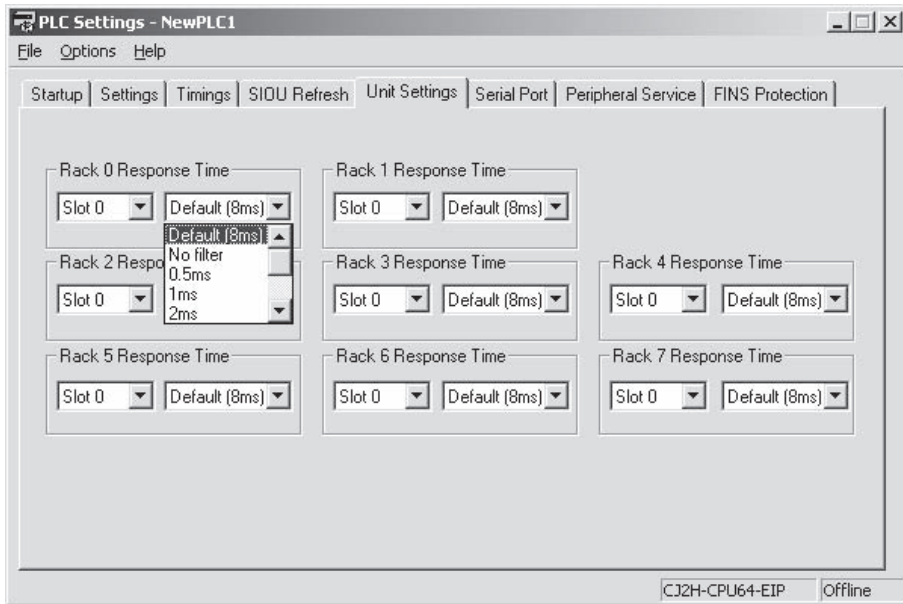
可通过机架和插槽编号设定基本 I/O 单元的输入响应时间。增加输入响应时间可有效降低振颤和干扰效应，而缩短输入响应时间（但应确保脉冲宽度始终长于循环时间）则有助于接收宽度较小的输入脉冲。

注 宽度小于循环时间的脉冲可通过高速输入单元输入。详情请参考“10-2-4 高速输入”。



PLC 设置

使用 CX-Programmer 时，请在“Unit Settings”（单元设定）选项页中对机架 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 单元的 I/O 响应时间	A220.00 ~ A259.15	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

下表列出了与基本 I/O 单元相关的出错信息。

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	基本 I/O 单元发生错误时为 ON。
基本 I/O 单元错误, 机架编号	A408.08 ~ A408.15	在基本 I/O 单元发生错误时, 存储出错位置的二进制机架编号 (A402.12 为 ON 时)。
基本 I/O 单元错误, 插槽编号	A408.00 ~ A408.07	在基本 I/O 单元发生错误时, 存储出错位置的二进制插槽编号 (A402.12 为 ON 时)。
基本 I/O 单元状态区 (机架 0 上的 插槽 0 ~ 机架 3 上的插槽 9)	A050.00 ~ A069.15	表示基本 I/O 单元的报警状态 (负载短路保护)。
中断输入单元位置出错标志	A405.08	未将中断输入单元连接至紧靠机架上 CPU 单元侧的四个位置 (插槽 0 ~ 3) 时为 ON。

10-4-2 CPU 总线单元标志 / 位

● 重启和初始化 CPU 总线单元

对于 CPU 总线单元，可在不关闭 PLC 的情况下，通过将下述重启位置 ON 的方式来变更分配到 CPU 单元的 DM 区字的初始设定。通常可通过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 CPU 总线单元 00 ~ 15(单元 0 ~ F)。将某个重启位置 ON 后，即可重启相应的单元。
CPU 总线单元初始化标志	A302.00 ~ A302.15	这些标志对应 CPU 总线单元 00 ~ 15(单元 0 ~ F)。电源接通或单元重启位 (A501.00 ~ A501.15) 置 ON 后，当对相应的单元执行初始化时，该标志将置 ON。

● CPU 总线单元错误

下表列出了与 CPU 总线单元相关的出错信息。

辅助区标志和字

名称	地址	描述
CPU 总线单元号重复标志	A410.00 ~ A410.15	当 CPU 总线单元的单元号发生重复时，重复出错标志 (A401.13) 和 A410 中的相应标志将会置 ON。 位 00 ~ 15 对应单元 0 ~ F。
CPU 总线单元设定出错标志 (非致命错误)	A402.03	安装的 CPU 总线单元与 I/O 表中注册的 CPU 总线单元不一致时为 ON。
CPU 总线单元设定错误，单元号标志	A427.00 ~ A427.15	当发生 CPU 总线单元设定错误时，A402.03 和 A427 中相应标志将会置 ON。 位 00 ~ 15 对应单元 0 ~ F。
CPU 总线单元出错标志 (非致命错误)	A402.07	当 CPU 单元和 CPU 总线单元之间的数据交换出现错误 (包括 CPU 总线单元本身的错误) 时为 ON。
CPU 总线单元错误，单元号标志	A417.00 ~ A417.15	当 CPU 单元和 CPU 总线单元之间的数据交换出现错误时，CPU 总线单元出错标志 (A402.07) 和 A417 中的相应标志将会置 ON。 位 00 ~ 15 对应单元 0 ~ F。

10-4-3 高功能 I/O 单元标志 / 位

● 重启和初始化高功能 I/O 单元

对于高功能 I/O 单元,可在不关闭 PLC 的情况下,通过将下述重启位置 ON 的方式来变更分配到 CPU 单元的 DM 区字的初始设定。通常可通过用户程序执行此操作。

辅助区标志和字

名称	地址	描述
高功能 I/O 单元重启位	A502.00 ~ A507.15	位 A502.00 ~ A507.15 对应高功能 I/O 单元 0 ~ 95。将某个重启位置 ON 后,即可重启相应的单元。
高功能 I/O 单元初始化标志	A330.00 ~ A335.15	这些标志对应高功能 I/O 单元 0 ~ 95。电源接通或单元重启位 (A502.00 ~ A507.15) 置 ON 后,当对相应的单元执行初始化时,该标志将置 ON。位 A330.00 ~ A335.15 对应单元 0 ~ 95。

● 高功能 I/O 单元错误

下表列出了与高功能 I/O 单元相关的出错信息。

辅助区标志和字

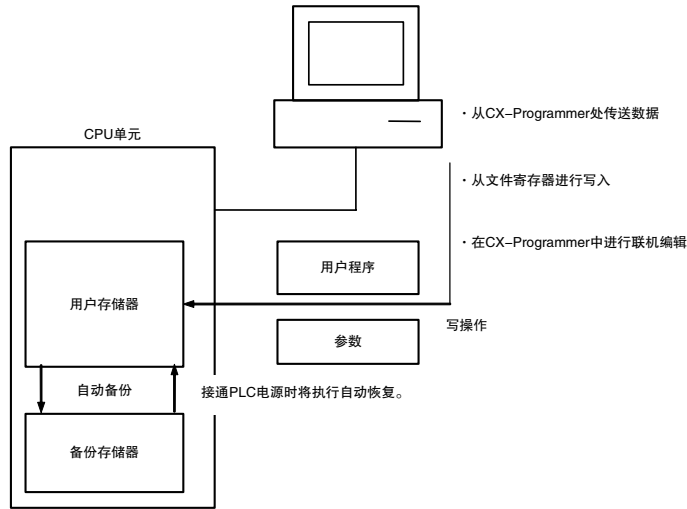
名称	地址	描述
高功能 I/O 单元号重复标志	A411.00 ~ A416.15	当高功能 I/O 单元的单元号发生重复时,重复出错标志 (A401.13) 和 A411 ~ A416 中的相应标志将会置 ON。位 A411.00 ~ A416.15 对应单元 0 ~ 95。
高功能 I/O 单元设定出错标志 (非致命错误)	A402.02	安装的高功能 I/O 单元与 I/O 表中注册的高功能 I/O 单元不一致时为 ON。
高功能 I/O 单元设定错误,单元号标志	A428.00 ~ A433.15	当发生高功能 I/O 单元设定错误时,A402.02 和这些字中的相应标志将会置 ON。位 A428.00 ~ A433.15 对应单元 0 ~ 95。
高功能 I/O 单元出错标志 (非致命错误)	A402.06	当 CPU 单元和高功能 I/O 单元之间的数据交换出现错误 (包括高功能 I/O 单元本身的错误) 时为 ON。
高功能 I/O 单元错误,单元号标志	A418.00 ~ A423.15	当 CPU 单元和高功能 I/O 单元之间的数据交换出现错误时,高功能 I/O 单元出错标志 (A402.06) 和这些字中的相应标志将会置 ON。

10-5 存储器管理功能

10-5-1 自动备份

在 CPU 单元中执行写入或变更操作时，用户程序或参数都将被自动备份到备份存储器中。

- 将对下列数据进行自动备份：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））。
- 在 CPU 单元中写入用户程序或参数（包括通过 CX-Programmer 传送数据、执行联机编辑、以及从存储卡或 EM 文件存储器传送数据等操作）时，将自动执行数据备份。
- 启动时，写入备份存储器的用户程序和参数数据将被自动传送到 CPU 单元的用户存储器中。



- 将数据写入备份存储器时，CPU 单元前部的 BKUP 指示灯将会点亮。通过 CX-Programmer 传送用户程序或者将文件存储器中的数据传送到参数区时，请勿在备份操作完成（直至 BKUP 指示灯熄灭时）前切断 CPU 单元电源。下表所示为 CPU 单元在自动备份过程中断电可能导致的情形。

● 自动备份过程中断电可能导致的情形

自动备份时间	适用数据	断电后的情形
传送程序时	用户程序	<ul style="list-style-type: none"> · 数据将会丢失。 · 下次接通电源时会发生存储器错误。
联机编辑时	用户程序： <ul style="list-style-type: none"> · 梯形图任务 	<ul style="list-style-type: none"> · 联机编辑期间所做的变更将会在下次接通电源时生效。（但在没有安装电池的情况下，所做的变更将不会生效，且会发生存储器错误。）
	用户程序： <ul style="list-style-type: none"> · ST 任务 · SFC 任务 · 功能块定义 · SFC 操作或转换程序 	<ul style="list-style-type: none"> · 联机编辑的内容将会丢失。
传送 PLC 设置数据、I/O 表、路由表或高性能 I/O 单元设定以及创建或删除 I/O 表时	参数	<ul style="list-style-type: none"> · 数据将会丢失。 · 下次接通电源时会发生存储器错误。

- 如果在 CPU 单元装有电池、且仅对梯形图任务进行联机编辑的情况下发生断电，则断电前的状态将在下次通电时恢复。但是请务必注意下列事项。
 - 启动 CPU 单元时需要更长的时间。
 - 在长时间未接通 CPU 单元电源的情况下，即使 CPU 单元中装有电池，也应确保在断电前执行了备份操作。
- 如下表所示，备份数据所需的时间（即 BKUP 指示灯点亮的时间）取决于用户程序的大小。

用户程序大小	备份处理时间	
	MONITOR 模式	PROGRAM 模式
100K 步	20s(循环时间 3ms)	3s
250K 步	47s(循环时间 7ms)	6s
400K 步	75s(循环时间 11ms)	10s

注 在 CPU 单元通电的情况下，BKUP 指示灯将会点亮。



正确使用注意事项

- 当将用户程序和参数数据写入 CPU 单元时，这些内容将自动备份至闪存。但是 I/O 存储区（包括 DM、EM、保持区和辅助区）中的数据将不会写入闪存。在断电状态下，可通过电池存储 DM、EM 和保持区中的数据。若存在电池故障，则无法保证这些区域中的数据在断电后的准确性。如果 DM、EM、保持区和辅助区中的内容用于控制外部输出，则无论电池出错标志 (A402.04) 是否置 ON，均应防止输出不当。
- 如果在 CPU 单元的备份存储器中检测到错误，则会发生备份存储器错误（非致命错误）。此时数据将不会在再次通电时恢复，并且将发生存储器错误。因此建议在关闭电源前对数据进行存储。



附加信息

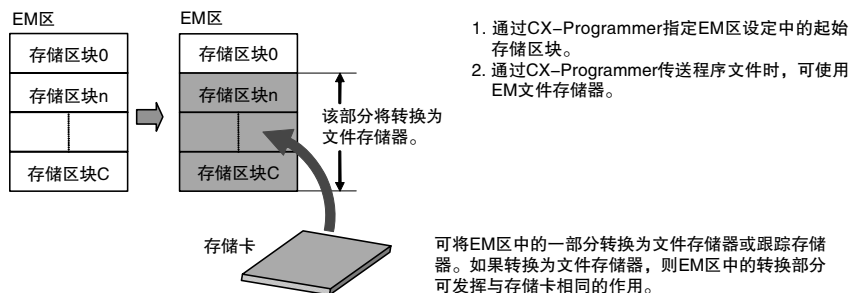
当通过 CX-Programmer 传送操作备份数据，而非使用普通的数据传送方式（“PLC - Transfer” (PLC- 传送)），则将通过 CX-Programmer 在存储器备份状态窗口中显示备份状态。若要显示该窗口，请选取 PLC 属性中的显示备份状态对话框，并在“View”（视图）菜单下选择该窗口。对于普通的传送操作，将在显示程序和其它数据的传送状态之后在传送窗口中显示备份状态。

● 辅助区标志

名称	地址	含义
备份存储器出错标志	A315.15	当备份存储器发生故障时置 ON。

10-5-2 EM 文件存储器功能

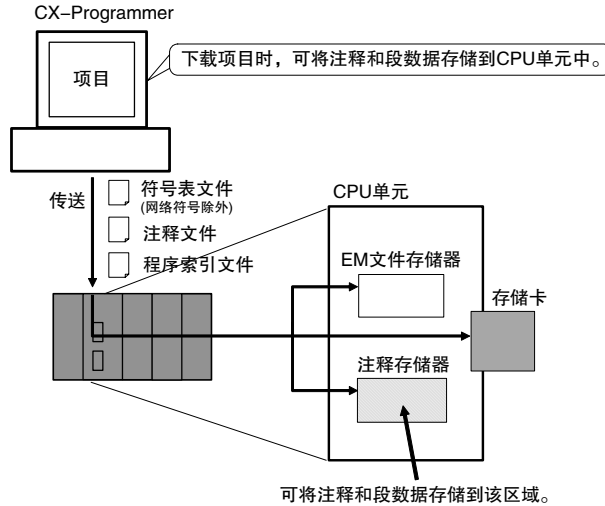
在 CPU 单元中可使用 EM 区替代存储卡存储文件。指定起始区块之后的区块将作为文件存储器。有关将 EM 区块转换为文件存储器的信息，请参考“7-1-2 文件存储器初始化”。



10-5-3 注释存储器

CPU 单元的内置备份存储器中具有注释存储器。即使没有存储卡或未指定 EM 文件存储器，也可通过注释存储器存储或读取下列注释 / 段数据。

- 符号表文件 (含 CX-Programmer 符号和 I/O 注释, 但不包括网络符号)
- 注释文件 (CX-Programmer 的梯级注释和其它注释)
- 程序索引文件 (CX-Programmer 的段名称、段注释和程序注释)

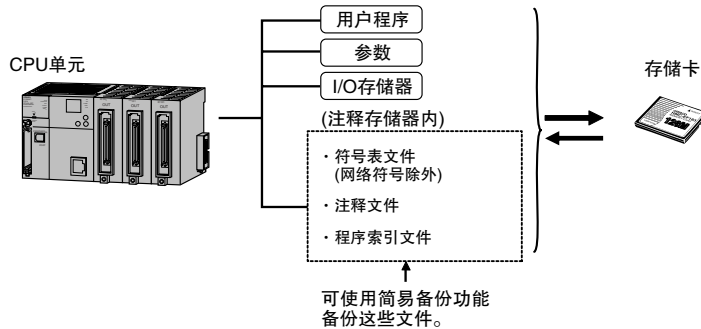


下载项目时, 可选择下列存储位置作为注释数据和段数据的传送目的地。

- 注释存储器 (在 CPU 单元的备份闪存内)
- 存储卡
- EM 文件存储器

当执行简易备份操作时, 可将存储在注释存储器中的下列文件备份至存储卡, 也可将这些文件从存储卡恢复至注释存储器。

- 符号表文件 (含 CX-Programmer 符号和 I/O 注释, 但不包括网络符号)
- 注释文件 (CX-Programmer 的梯级注释和其它注释)
- 程序索引文件 (CX-Programmer 的段名称、段注释和程序注释)

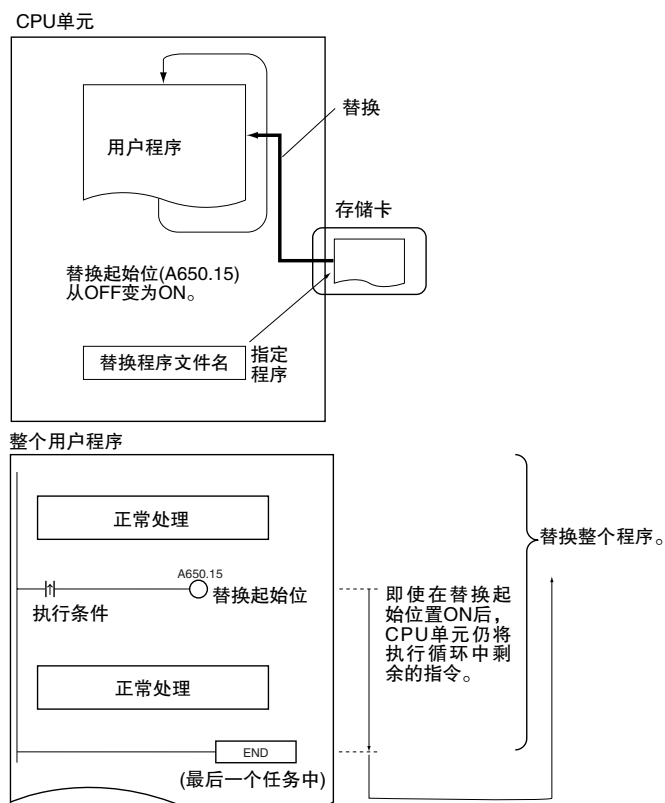


这样既可在发生错误或添加一个相同规格的 CPU 单元的情况下, 对 CPU 单元中的所有数据 (包括 I/O 注释) 执行备份 / 恢复, 而无需使用 CX-Programmer。

10-5-4 在运行过程中替换整个程序

概述

将替换起始位 (A650.15) 置 ON 后, 即可在 RUN 或 MONITOR 运行模式下替换整个任务程序。通过事先在辅助区中指定程序 / 网络符号文件 (.OBJ) 的文件名, 从存储卡中读取这些文件, 然后用其在当前循环的末尾替换程序。



通过 CX-Programmer 将替换起始位置 ON, 也可在程序执行停止时 (PROGRAM 模式) 对其进行替换。

- 无法从 EM 文件存储器读取替换程序文件。
- 可在程序中的任何位置 (程序地址) 将替换起始位 (A650.15) 置 ON。
- 替换起始位从 OFF 变为 ON 后, CPU 单元将执行循环中剩余的指令。
- 程序替换过程中将不会执行程序。程序替换完成后, 由于 CPU 单元从 PROGRAM 模式切换至 RUN 或 MONITOR 模式, 将再次开始执行程序。
- 当替换起始位从 OFF 变为 ON 时, 将在循环的末尾 (即在程序的最后一个任务中执行 END(001) 后) 替换程序。

替换程序时的 CPU 运行情况

在替换程序时，CPU 单元的运行情况如下所示：

- 程序执行：停止
- 循环时间监控：不监控

● 在替换程序时及替换程序后保持的操作

- 当 IOM 保持位 (A500.12) 为 ON 时，将保持下列存储区中的数据：CIO 区、工作区 (W)、定时器完成标志 (T)、变址寄存器 (IR)、数据寄存器 (DR) 和当前的 EM 存储区块号。

注 替换程序时，将清除定时器当前值 (PV)。

- 传送程序时，如果 IOM 保持位为 ON，则替换程序前正在输出的负载将在替换后继续输出。请确保在替换程序后，外部负载的运行情况正常。
- 如果强制状态保持位 (A500.13) 为 ON，则将在程序替换过程中始终保持强制置位 / 复位的状态。
- 屏蔽中断。
- 如果正在执行数据跟踪，则将停止操作。
- 无论 IOM 保持位为 ON 或 OFF，微分标志都将被初始化。

● 替换程序后的 CPU 运行情况

- 循环任务的状态取决于其运行 - 启动的属性。(其状态与将 PLC 从 PROGRAM 模式切换到 RUN/MONITOR 模式时的状态相同。)
- 在继续执行程序后的一个循环内，第一循环标志 (A200.11) 将会置 ON。(其状态与将 PLC 从 PROGRAM 模式切换到 RUN/MONITOR 模式时的状态相同。)

步骤

1. 将存有程序文件的存储卡插入 CPU 单元。
2. 在辅助区中设定程序文件名 (A654 ~ A657) 和程序密码 (A651)，然后将替换起始位 (A650.15) 置 ON。



附加信息

如果想要在程序替换过程中保持 I/O 存储器的状态，则应将 IOM 保持位 (A500.12) 置 ON。

如果想要在程序替换过程中保持强制置位和强制复位的状态，则应将强制状态保持位 (A500.13) 置 ON。



安全使用注意事项

如果 IOM 保持位 (A500.12) 在替换程序前为 ON，则将在替换程序后保持 I/O 存储器中的位状态。请确保在使用相同 I/O 存储器数据时外部负载的运行情况正常。

如果强制状态保持位 (A500.13) 在替换程序前为 ON，则将在替换程序后保持 I/O 存储器中的强制状态。请确保在使用相同强制状态时外部负载的运行情况正常。

替换文件

将从存储卡读取指定的程序 / 网络符号文件。

文件	文件名和扩展名	将辅助区中的替换起始位 (A650.15) 置 ON	指定替换文件名 (*****)
程序 / 网络符号文件	*****.OBJ	替换 CPU 单元的用户程序	替换程序前, 将替换程序文件名写入 A654 ~ A657。

替换程序所需的条件

若要在运行过程中替换程序, 应满足下列条件。

- 存储卡的根目录中存在有在程序文件名字 (A654 ~ A657) 中指定的程序 / 网络符号文件。
- CPU 单元已检测到存储卡 (A343.15 必须为 ON)。
- 未发生致命错误。
- 未在执行文件存储器操作 (A343.13 必须为 OFF)。
- 未在执行程序区数据写入操作。
- 具备访问权限。(例如, 未将数据从 CX-Programmer 传送至 PLC 时。)

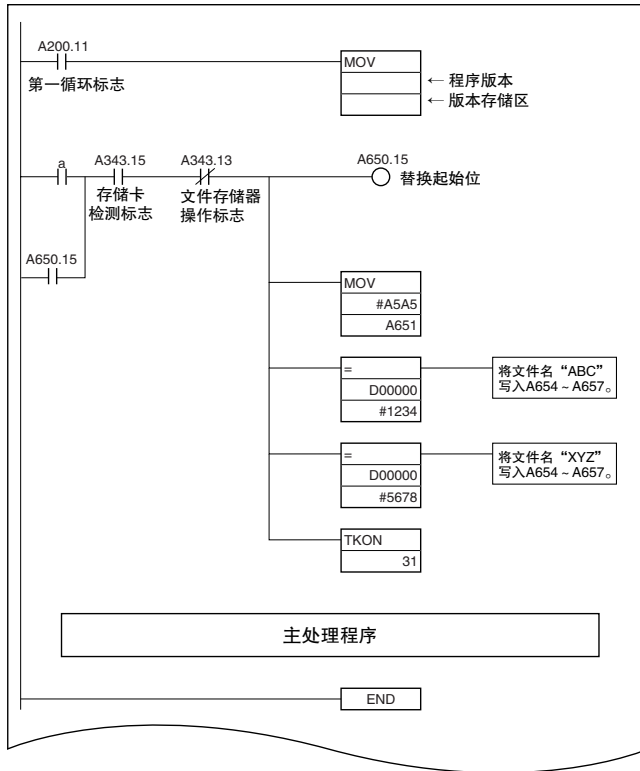
注 程序可在任何运行模式下传送。

程序示例

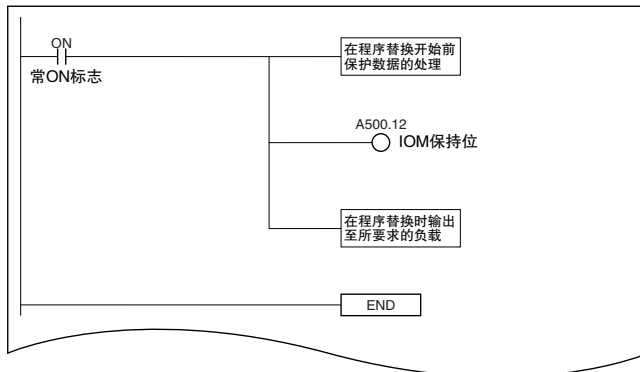
● 程序示例 1

下例中，程序 / 网络符号文件 ABC.OBJ 和 XYZ.OBJ 被存储在存储卡中，并根据 D0 的值选择一个程序或选择其它数据。若要选择 ABC.OBJ，将 D0 设定为 #1234；若要选择 XYZ.OBJ，将 D0 设定为 #5678。然后再启动另一个任务来执行替换程序前所需的处理（包括设定 IOM 保持位）。

主任务(循环任务0)

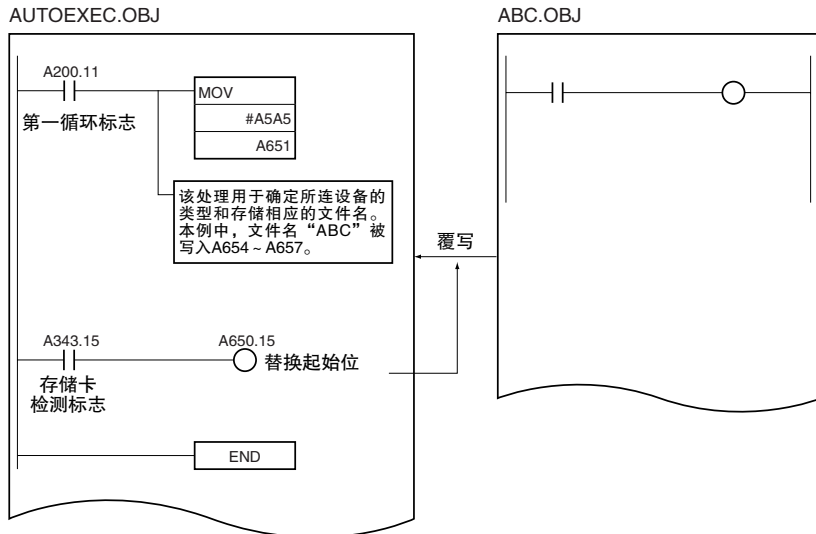


在程序替换时保护数据的任务
(循环任务31，启动时的待机状态)



● 程序示例 2

下例中，相关设备的程序 / 网络符号文件及启动时自动传送的程序 / 网络符号文件 (AUTOEXEC.OBJ 或 REPLACE.OBJ) 被存储在存储卡中，然后当打开 PLC 读取启动时自动传送的文件，并用相关设备的程序文件替换该程序文件。



辅助区标志和字

名称	地址	描述
文件存储器操作标志	A343.13	在执行下列任意操作时为 ON，不执行时为 OFF。 <ul style="list-style-type: none"> · 存储卡检测 · CPU 单元使用 CMND(490) 向自身发送一条 FINS 指令。 · 正在执行 FREAD(700) 或 FWRT(701)。 · 正在使用辅助区控制位 (A650.15) 覆盖程序。 · 正在执行简易备份操作。
存储卡检测标志	A343.15	检测到存储卡时为 ON，未检测到存储卡时为 OFF。
IOM 保持位	A500.12	当 CPU 单元在运行模式 PROGRAM 和 RUN 或 MONITOR 之间切换或接通电源时，将该位置 ON 可保持 I/O 存储器的状态。 ON: 变更运行模式时，I/O 存储器中的状态将被保持。 OFF: 变更运行模式时，I/O 存储器的状态将被清除。
强制状态保持位	A500.13	当 CPU 单元在运行模式 PROGRAM 和 RUN 或 MONITOR 之间切换或接通电源时，将该位置 ON 可保持强制置位或强制复位的位状态。
替换完成码	A650.00 ~ A650.07	<ul style="list-style-type: none"> · 程序替换正常代码 (A650.14 为 OFF): 01 Hex: 通过程序 / 网络符号文件 (.OBJ) 替换程序。 · “程序替换失败”代码 (A650.14 为 ON): 00 Hex: 发生致命错误。 01 Hex: 发生存储器错误。 11 Hex: 程序写保护。 12 Hex: A651 中的程序密码错误。 21 Hex: 未安装存储卡。 22 Hex: 指定文件不存在。 23 Hex: 指定文件太大 (存储器错误)。 31 Hex: 说明正在执行下列操作之一: - 正在执行文件存储器操作。 - 正在写入程序。 - 正在切换运行模式。
替换出错标志	A650.14	在 A650.15 已从 OFF 变为 ON 的情况下，若替换程序时出现错误，该标志置 ON。 当 A650.15 再次从 OFF 变为 ON 时，该标志置 OFF。

名称	地址	描述															
替换起始位	A650.15	<p>将程序密码 (A651) 设定为 A5A5 Hex, 使该位有效。当该位从 OFF 变为 ON 时, 则开始替换程序。在替换程序的过程中, 请勿再次将该位从 OFF 切换为 ON。</p> <p>程序替换操作完成 (正常或出错) 或电源接通时, 该位将自动置 OFF。</p> <p>位状态可通过 CX-Programmer、PT 或上位计算机读取, 以便确定程序替换是否完成。</p>															
程序密码	A651	<p>在该字中写入密码, 以允许程序替换操作。</p> <p>A5A5 Hex: 启用替换起始位 (A650.15)。</p> <p>其它值: 禁用替换起始位 (A650.15)。</p> <p>程序替换操作完成 (正常或出错) 或电源接通时, 该位将自动置 ON。</p>															
程序文件名	A654 ~ A657	<p>开始替换程序前, 将替换程序文件名以 ASCII 格式写入这些字。最长可指定 8 个字符的文件名, OBJ 扩展名则会自动添加。字符写入的起始位置为 A654 (即从高位字节开始)。如果文件名长度小于 8 个字符, 则应将空格代码 (20 Hex) 填入剩余的字节中。请勿将文件名中加入任何无效字符或空格。</p> <p>下例所示为程序文件 ABC.OBJ 对应的数据:</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th></th> <th>15</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>A654</td> <td>41</td> <td>42</td> </tr> <tr> <td>A655</td> <td>43</td> <td>20</td> </tr> <tr> <td>A656</td> <td>20</td> <td>20</td> </tr> <tr> <td>A657</td> <td>20</td> <td>20</td> </tr> </tbody> </table>		15	0	A654	41	42	A655	43	20	A656	20	20	A657	20	20
	15	0															
A654	41	42															
A655	43	20															
A656	20	20															
A657	20	20															

10-6 安全功能

下列保护措施适用于 CJ 系列机型的用户程序。

10-6-1 通过 DIP 开关实现写保护

设定写保护功能，使 CX-Programmer 无法对用户程序和参数区（如 PLC 设置和 I/O 表）中的数据进行覆写操作。这样可以有效防止程序被意外修改。

若要设定写保护功能，请将 CPU 单元 DIP 开关的第 1 位设定为 ON。但即使启用了这一功能，仍可通过 CX-Programmer 读取和显示程序。

● CPU 单元前部的 DIP 开关

开关位	名称	设定
1	用户程序存储器写保护	ON: 保护启用 OFF: 保护关闭

10-6-2 通过密码实现读保护

读保护功能可通过设定密码来实现。除非用户在 CX-Programmer 中输入密码，否则受保护程序中的用户程序、特定任务和功能块将始终无法显示或编辑。

保护类型

可使用密码实现下述类型的读保护。

● UM 读保护

禁止从 CPU 单元读取用户程序。

● 任务读保护

禁止在 CX-Programmer 中显示一个或多个任务。可从 CPU 单元中读取用户程序，但无法在 CX-Programmer 中显示受保护的任務。



附加信息

- 即使对程序中的个别任务进行了读保护设置，仍可将整个程序传送至另一个 CPU 单元。对于执行过加密保护的任務，任务读保护功能仍然有效。
- 通过 CX-Programmer 比较计算机存储器和 CPU 单元中的用户程序时，也将比较执行过加密保护的任務。

● 功能块保护

下列保护措施适用于所需的功塊。

- 写 / 显示保护
禁止在 CX-Programmer 中显示功塊，且无法对其进行修改。
- 写保护
允许在 CX-Programmer 中显示功塊，但无法对其进行修改。



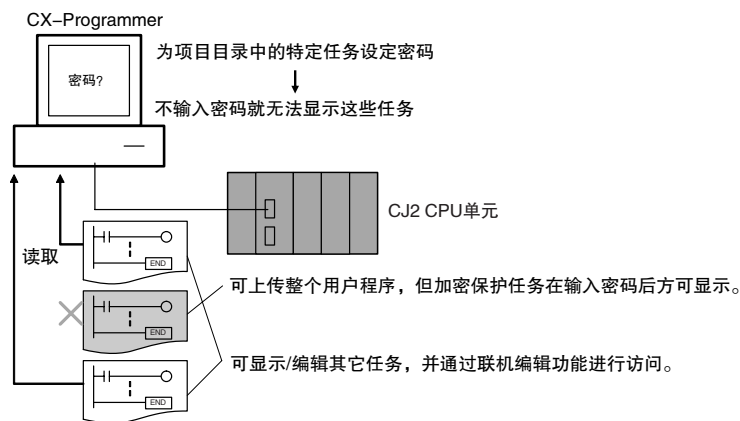
附加信息

在使用包含功能块的用户程序的情况下，即使启用了任务读保护功能，仍可读取功能块定义。若要对功能块进行读保护，请启用功能块保护功能。



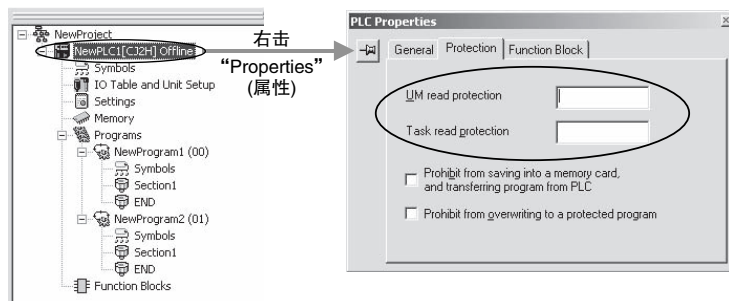
附加信息

若事先设定了 UM 读保护，将无法再设定任务读保护。但是在事先设定了任务读保护的情况下，还可设定 UM 读保护。



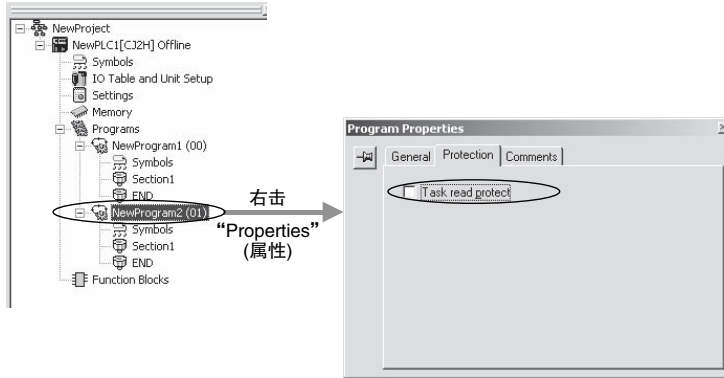
操作步骤

1. 打开“PLC Properties” (PLC 属性) 窗口中的“Protection” (保护) 选项页并输入保护密码。

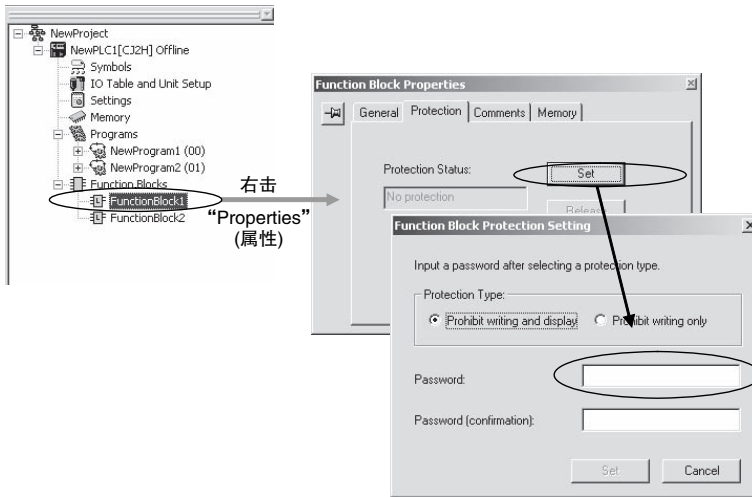


- “UM read protection” (UM 读保护) 文本框用于输入 UM 读保护密码。
- “Task read protection” (任务读保护) 文本框用于输入任务读保护密码。

2. 点击该任务并在“Program Properties”（程序属性）对话框中勾选“Task read protect”（任务读保护）复选框，以便为某个任务设定读保护。



3. 选择所需保护的功能块并点击功能块属性窗口中的“Set”（设定）按钮，然后在“Function Block Protection Setting”（功能块保护设定）对话框的文本框中输入密码，以便为功能块 (FB) 定义设定读保护。



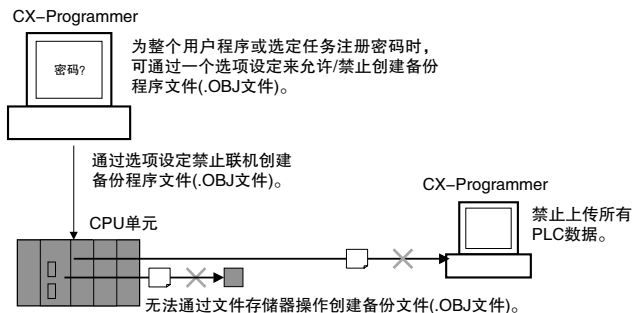
4. 联机并执行下述的步骤 a 或 b。
 - (a) 传送程序和设定密码保护：
选择“PLC – Transfer – To PLC”（PLC – 传送至 – PLC）来传送程序。步骤 2 中注册的任务将受到密码保护。
 - (b) 不传送程序，设定密码保护：
选择“PLC – Protection – Set Password”（PLC – 保护 – 设定密码）并点击“OK”（确定）按钮。步骤 2 中注册的任务将受到密码保护。

允许 / 禁止存储到存储卡

● 概述

当在 CX-Programmer 中对整个程序或单个任务进行读保护时，可通过一个选项来允许或禁止 .OBJ 程序 / 网络符号文件的创建 / 备份。如果通过此项设定禁止了程序 / 网络符号文件的创建 / 备份，就无法再通过文件存储器操作来创建程序 / 网络符号文件 (.OBJ 文件)。(该设定既可禁止将数据联机传送到存储器卡 / EM 文件存储器，也可禁止将脱机存储的 PLC 数据上传至 CX-Programmer。)

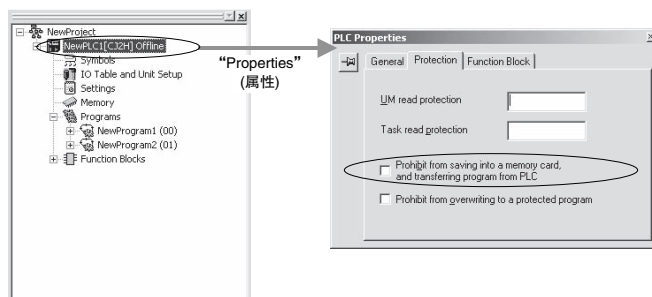
禁止创建存储器程序文件有助于防止对用户程序的非法拷贝。



- 在禁止创建程序 / 网络符号文件的情况下，仍可执行简易备份操作，但不会创建备份程序 / 网络符号文件 (BACKUP.OBJ)。
- 在未启用程序读保护功能的情况下，可对程序进行拷贝操作。
- 将程序传送至 CPU 单元后，用于允许 / 禁止创建程序 / 网络符号文件的设定才会生效。因此请务必在修改该设定后传送程序。

● 操作步骤

1. 在 CX-Programmer 的 PLC 属性对话框的“Protection”（保护）选项页中勾选“Prohibit from saving into a memory card, and transferring program from PLC”（禁止存储到存储卡，从 PLC 传送程序）复选框，以便在“UM read protection”（UM 读保护）或“Task read protection”（任务读保护）文本框中注册密码。



2. 选择“PLC - Transfer - To PLC”（PLC - 传送至 - PLC）来传送程序，或者选择“PLC - Protection - Set Password”（PLC - 保护 - 设定密码）并点击“OK”（确定）按钮。

● 用途

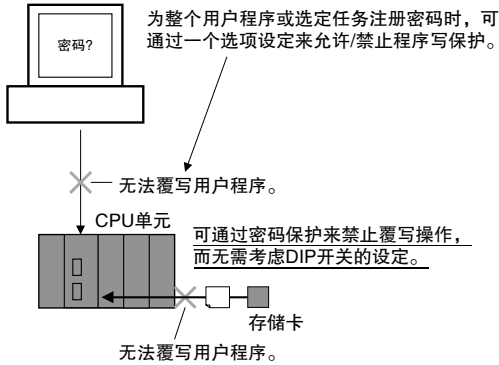
- 该选项可通过加密的方式防止程序被传送出 PLC。

允许和禁止程序覆写

● 概述

除了使用上文中的 DIP 开关外，在为整个程序或选定任务注册密码时，如果在 CX-Programmer 中选择了写保护选项，则程序（或选定任务）也可实现写保护。写保护设定可防止程序受到非法或意外覆写。

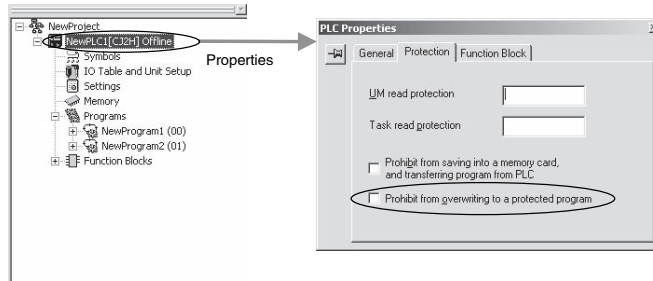
CX-Programmer



- 如果在注册密码时通过该选项来实现对选定任务或程序的写保护，则只有进行过加密处理的任务(程序)会受到覆写保护。通过联机编辑和任务下载等操作，仍可对其它任务或程序进行覆写。
- 当未启用程序读保护时，所有任务(程序)都可被覆写。
- 将程序传送至 CPU 单元后，用于允许 / 禁止创建程序文件的设定才会生效。因此请务必在修改该设定后传送程序。

● 操作步骤

1. 在 CX-Programmer 的 PLC 属性对话框的“Protection”（保护）选项页中勾选“Prohibit from overwriting to a protected program”（禁止覆写受保护的程序）复选框，以便在“UM read protection”（UM 读保护）或“Task read protection”（任务读保护）文本框中注册密码。



2. 选择“PLC - Transfer - To PLC”（PLC - 传送至 - PLC）来传送程序，或者选择“PLC - Protection - Set Password”（PLC - 保护 - 设定密码）并点击“OK”（确定）按钮。

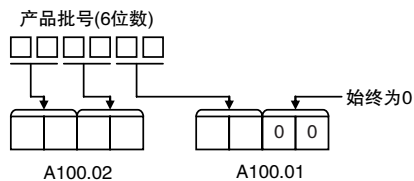
与密码保护相关的辅助区标志和位

名称	位地址	描述
UM 读保护标志	A99.00	表示是否对 PLC(整个用户程序) 设定读保护。 OFF: 未设定 UM 读保护。 ON: 已设定 UM 读保护。
任务读保护标志	A99.01	表示是否对选定程序任务设定读保护。 OFF: 未设定任务读保护。 ON: 已设定任务读保护。
针对读保护的程序写保护	A99.02	表示是否选择写保护选项来防止程序受到覆写。 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	表示是否在设定 UM 读保护或任务读保护时创建备份程序 / 网络符号文件 (.OBJ 文件)。 OFF: 允许创建备份程序文件 ON: 禁止创建备份程序文件

10-6-3 使用产品批号实现程序操作保护

通过存储在辅助区字 A100.01 和 A100.02 中的产品批号, 可对程序进行操作保护。用户无法对产品批号进行更改。

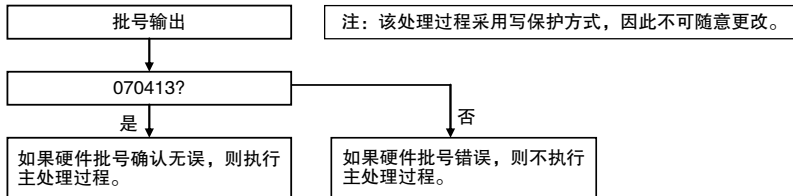
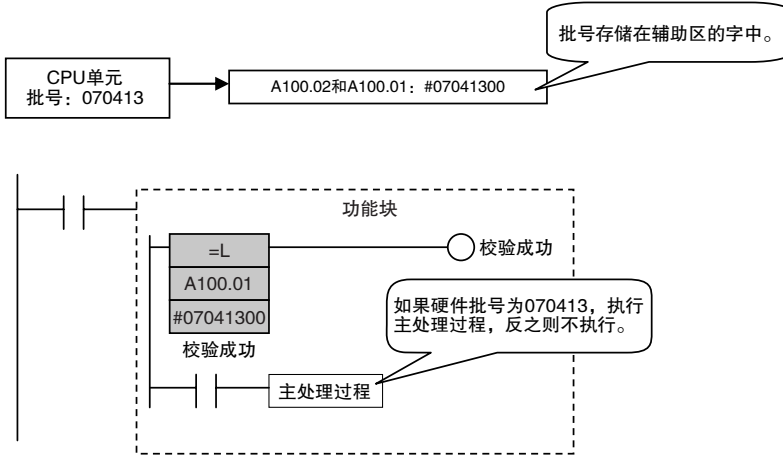
- 产品批号为 6 位数字。其左 4 位存储在 A100.02 中, 而右 2 位存储在 A100.01 中。



产品批号的存储示例

生产日期	产品批号	A100.02	A100.01
2008 年 6 月 20 日	080620	0806	2000

应用示例：仅针对带有特定产品批号的 CPU 单元运行程序



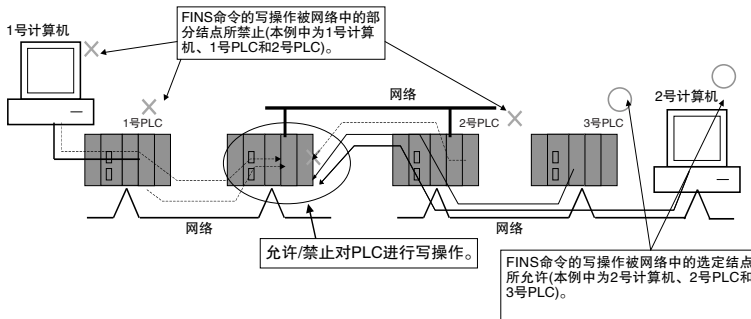
10-6-4 针对 FINS 命令的写保护

该功能可禁止以 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 的应用程序发送的写操作，但直接串行连接除外)，但此时不会禁止读处理。

FINS 写保护可禁止写处理，如用户程序、PLC 设置或 I/O 存储器数据下载、运行模式变更以及联机编辑。可将选定的结点排除在写保护范围以外，以便将数据从这些结点写入。

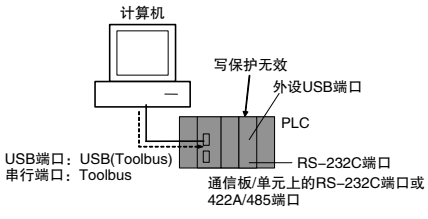
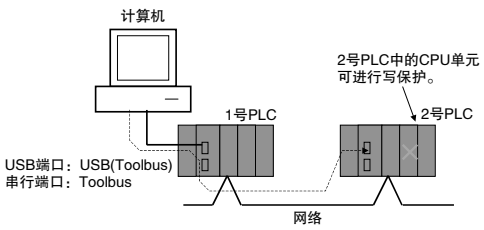
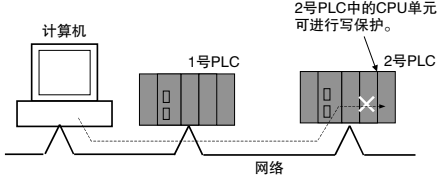
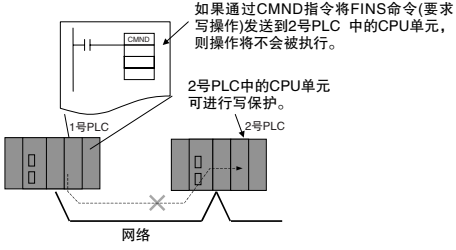
CPU 单元中的一个事件日志中自动记录了所有通过网络发送的写处理，并可通过 FINS 命令读取此记录。

例：



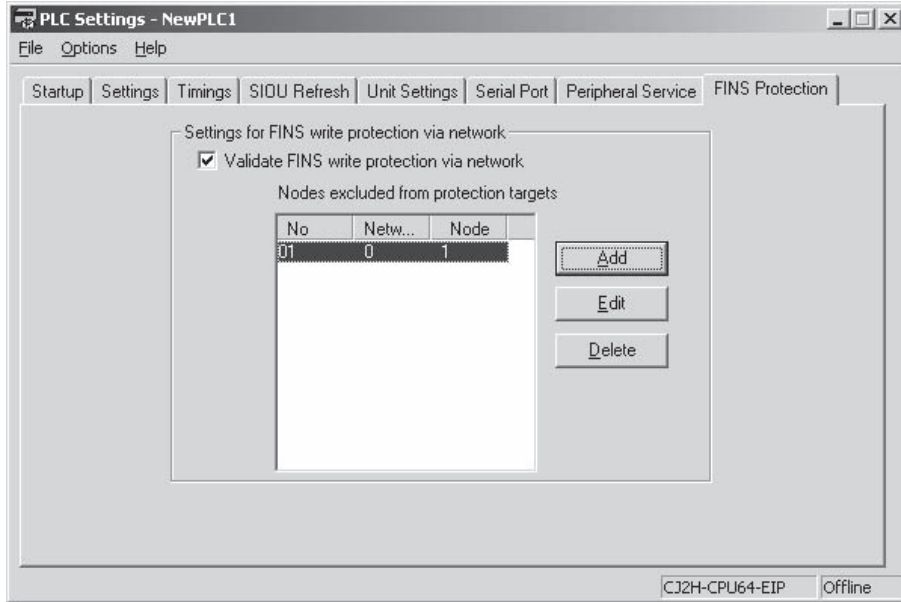
注 该功能仅针对 FINS 命令进行写保护，对于 FINS 命令以外的其它功能 (诸如数据链接) 做出的写操作无效。

● 写保护模式示例

连接模式	示意图 (示例)	写保护	
来自采用直接串行连接方式的计算机	与 PLC 直接连接	 <p>计算机</p> <p>写保护无效</p> <p>外设USB端口</p> <p>PLC</p> <p>USB端口: USB(Toolbus)</p> <p>串行端口: Toolbus</p> <p>通信板/单元上的RS-232C端口或422A/485端口</p>	无法使用
	与PLC进行网关连接(串行→网络)	 <p>计算机</p> <p>1号PLC</p> <p>2号PLC中的CPU单元可进行写保护。</p> <p>USB端口: USB(Toolbus)</p> <p>串行端口: Toolbus</p> <p>网络</p>	可以使用
来自采用直接网络连接方式的计算机		 <p>计算机</p> <p>1号PLC</p> <p>2号PLC中的CPU单元可进行写保护。</p> <p>网络</p>	可以使用
来自网络中的其它 PLC		 <p>1号PLC</p> <p>2号PLC中的CPU单元可进行写保护。</p> <p>网络</p> <p>如果通过CMND指令将FINS命令(要求写操作)发送到2号PLC中的CPU单元,则操作将不会被执行。</p>	可以使用

● 操作

使用 CX-Programmer 时，打开 PLC 设置中的“FINS Protection” (FINS 保护) 选项页并勾选“Validate FINS write protection via network” (在网络中使用 FINS 写保护) 选项。选择了该选项后，CPU 单元将无法通过网络 (直接串行连接除外) 使用 FINS 命令来执行写操作。若要允许从特定结点执行写操作，应在“Nodes excluded from protection targets” (不作为保护对象的结点) 中输入网络地址和结点地址。(最多可将 32 个节点设定为不进行 FINS 写保护。)

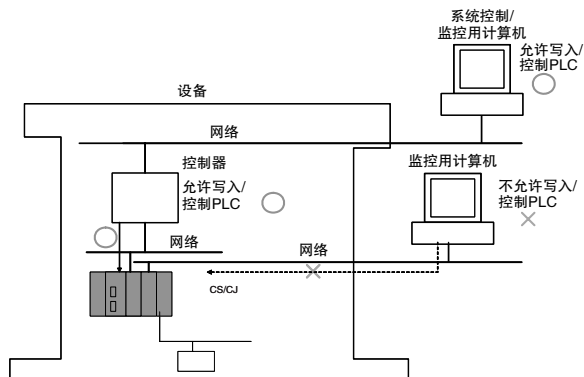


● PLC 设置

项目	描述	设定	默认设定
在网络中使用 FINS 写保护	确定是否针对通过网络发送的 FINS 命令设定 CPU 单元写保护 (不禁止通过直接串行连接发送的 FINS 命令)。	OFF: 禁止写保护 ON: 允许写保护	OFF: 禁止写保护
不进行写保护的结点	该区域列出了网络中不受 FINS 写保护限制的结点。最多可指定 32 个结点。 这些设定仅在启用 FINS 写保护时有效。		
	网络地址: FINS 命令源的网络地址	00 ~ 7F Hex	
	结点地址: FINS 命令源的结点地址	01 ~ FE Hex 或 FF Hex (FF Hex: 未指定结点地址)	
不进行 FINS 写保护的结点数 (请勿设定该数值, 应由 CX-Programmer 自动计算。)	包含不受 FINS 写保护限制的结点数。 若指定了 0 (对所有结点进行写保护), 则除了本地结点之外的所有节点均会禁止 FINS 写命令操作。 该设定仅在启用 FINS 写保护时有效。	0 ~ 32 (00 ~ 20 Hex) (数值 0 表示对所有结点均进行写保护。)	0 (所有结点均进行写保护。)

● 用途

- 可对系统进行配置，使 PLC 只能从网络中的指定结点处写入。（例如，当系统的控制 / 监控计算机作为整套设备中唯一有权限对控制器执行写入操作的结点时使用该功能。）
通过限制可写入 PLC 的结点数，可防止数据监控期间因发生意外覆写所造成的系统问题。



● 受网络 FINS 写保护限制的操作

FINS 写命令

当通过网络发送至 CPU 单元时，下列 FINS 命令会受到 FINS 写保护的限制。

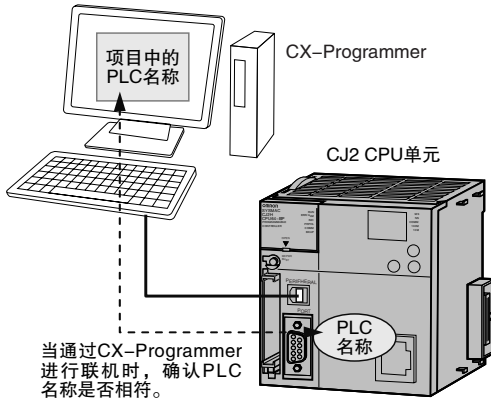
代码	命令名称
0102 Hex	存储区写入
0103 Hex	存储区填充
0105 Hex	存储区传送
0202 Hex	参数区写入
0203 Hex	参数区填充 (清除)
0307 Hex	程序区写入
0308 Hex	程序区清除
0401 Hex	运行
0402 Hex	停止
0702 Hex	时钟写入
0C01 Hex	获取访问权限

代码	命令名称
2101 Hex	错误清除
2103 Hex	出错日志指针清除
2203 Hex	单文件写入
2204 Hex	文件存储器格式化
2205 Hex	文件删除
2207 Hex	文件复制
2208 Hex	文件名修改
220A Hex	存储区 - 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

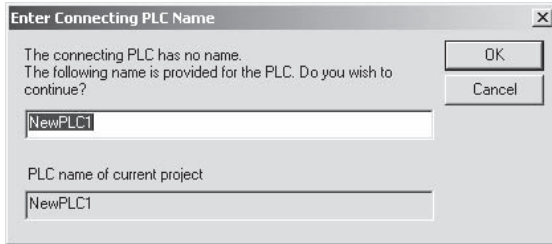
PLC 名称

通过 CX-Programmer 进行联机时，可确认项目中的 PLC 名称与连接对象处 CPU 单元的 PLC 名称是否相符。这样可以避免 CX-Programmer 处出现错误连接。



注册 PLC 名称

当对使用默认设定或存储器已清零的 CJ2 CPU 单元进行联机编辑时，将显示“Enter Connecting PLC Name”（输入连接 PLC 名称）对话框。

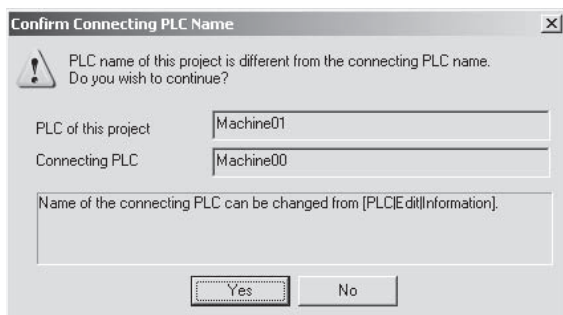


输入 PLC 名称并注册到联机的 CJ2 CPU 单元，然后点击“OK”（确认）按钮。如果不想输入 PLC 名称，则点击“Cancel”（取消）按钮。

PLC 名称不匹配

如果 PLC 名称存储在 CJ2 CPU 单元中，则在进行联机时，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 CJ2 CPU 单元中存储的 PLC 名称 (连接对象处) 与项目中的 PLC 名称不相符，则将显示下述警告框，提醒用户选择是否继续进行联机。



- “Yes” (是) 按钮
点击 “Yes” (是) 按钮，从而与不同名称的 PLC 联机。
- “No” (否) 按钮
点击 “No” (否) 选择脱机。



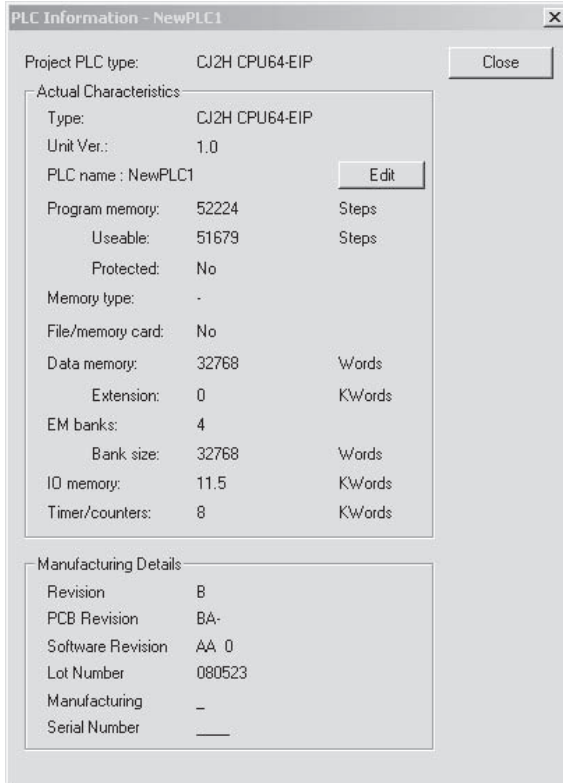
附加信息

若不需要自动检测 PLC 名称匹配，则应在 “PLC Properties” (PLC 属性) 对话框中取消对 “Check the connecting PLC name when going online” (在联机时检查所连接的 PLC 名称) 的选择。

修改 PLC 名称

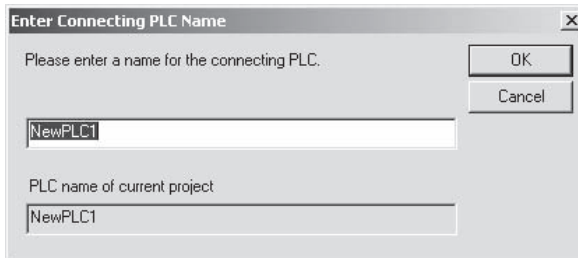
请按照以下步骤修改存储在 CJ2 CPU 单元中的 PLC 名称。

1. 与 CX-Programmer 联机，并在 PLC 菜单中选择 “Edit” (编辑) - “Information” (信息)。此时将显示以下对话框。



2. 点击 PLC 名称区域右侧的 “Edit” (编辑) 按钮。

此时将显示以下 “Enter Connecting PLC Name” (输入连接的 PLC 名称) 对话框。



3. 输入 PLC 名称并注册到连接目标 PLC，然后点击 “OK” (确认) 按钮。

10-7 调试

10-7-1 强制置位 / 复位

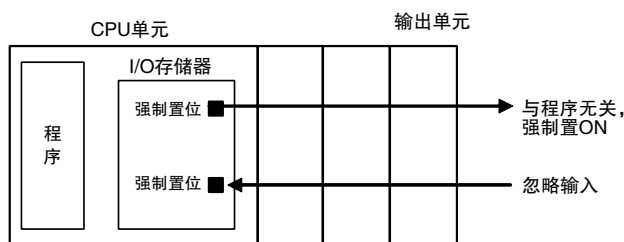
CX-Programmer 可对指定位 (CIO 区、辅助区、HR 区和定时器 / 计数器完成标志) 进行强制置位 (ON) 或复位 (OFF)。强制状态具备比程序或 I/O 刷新的状态输出更高的优先权。此状态不能通过指令覆写, 并且无论程序或外部输入的状态如何, 都将对其进行存储, 直到通过 CX-Programmer 将其清除。

强制置位 / 复位操作用于在试运行操作期间进行强制输入 / 输出或在调试期间施加某些强制条件。

强制置位 / 复位操作可在 MONITOR 或 PROGRAM 模式下执行, 但不可在 RUN 模式下执行。

注 强制状态保持位 (A500.13) 和 IOM 保持位 (A500.12) 可同时置 ON, 以便在切换运行模式时保持已强制置位或复位的位状态。

将强制状态保持位 (A500.13) 和 IOM 保持位 (A500.12) 置 ON, 并将 PLC 设置中的 “Forced Status Hold Bit at Startup” (启动时的强制状态保持位) 项设定为 “在启动时保持强制状态保持位的状态”。这样即可在通电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 区、数据链接区、CPU 总线单元区、高功能 I/O 单元区及内部 I/O 区中的位)、工作区、定时器完成标志、HR 区、计数器完成标志以及 EM 区中指定了强制置位 / 复位功能的区块 *1、*2

*1 仅有 CJ2H CPU 单元 (1.2 版或更高版本) 或 CJ2M CPU 单元支持 EM 区强制置位 / 复位功能, 并需要使用 CX-Programmer 8.3 版。

*2 对于 CJ2H CPU 单元, 可对下述区中指定了自动地址分配的 EM 存储区块进行强制置位 / 复位。

- CJ2H-CPU64/65(-EIP): EM 区块 (3 Hex)
- CJ2H-CPU66(-EIP): EM 区块 (6 ~ 9 Hex)
- CJ2H-CPU67(-EIP): EM 区块 (7 ~ E Hex)
- CJ2H-CPU68(-EIP): EM 区块 (11 ~ 18 Hex)

● CX-Programmer 操作

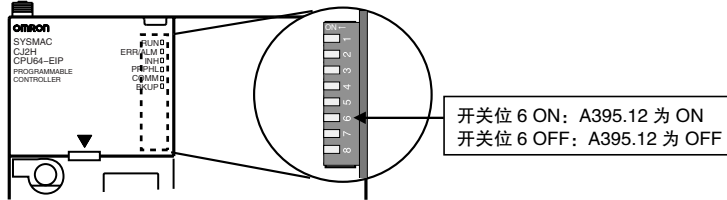
- 选择需要强制置位 / 复位的状态位。
- 选择强制置位或强制复位。
- 清除强制状态 (包括同时清除所有强制状态)。

● 强制状态保持位

名称	地址	功能	访问权限
强制状态保持位	A500.13	在 PROGRAM 和 RUN 或 MONITOR 模式之间进行切换或接通电源时, 将该位置 ON 可保持强制置位或强制复位的位状态。	读 / 写

10-7-2 测试输入

将 CPU 单元前部的 DIP 开关位 6 的 ON/OFF 状态存储到辅助区的 DIP 开关位状态标志 (A395.12) 中。若要 进行测试或其它操作，可对输入进行手动控制，以便在不使用输入单元的情况下进行测试。

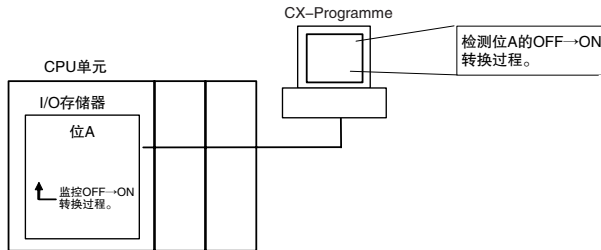


● 辅助区标志和字

名称	地址	功能	访问权限
DIP 开关位状态标志	A395.12	包含 CPU 单元 DIP 开关第 6 位的设定状态 (每循环刷新)。	只读

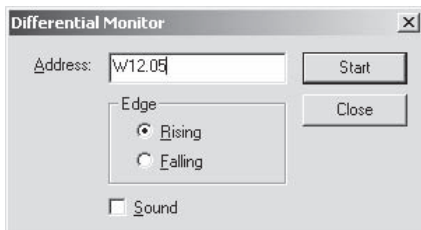
10-7-3 微分监控

当 CPU 单元检测到由 CX-Programmer 编程设备设定的位已从 OFF 变为 ON 或从 ON 变为 OFF 时，将在微分监控完成标志 (A508.09) 中指明监控结果。当满足针对微分监控设定的条件时，标志将变为 ON。CX-Programmer 将在屏幕上监控和显示这些结果。



● 通过 CX-Programmer 操作

1. 右击需要进行微分监控的位。
2. 点击 PLC 菜单中的 “Differential Monitor” (微分监控)。此时将显示微分监控对话框。

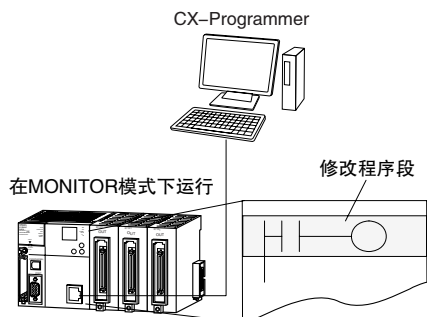


3. 点击 “Rising” (上升沿) 或 “Falling” (下降沿)。
4. 点击 “Start” (开始) 按钮。当检测到指定的变化时，蜂鸣器将响起，且计数值将递增。
5. 点击 “Stop” (停止) 按钮，使微分监控停止。

10-7-4 联机编辑

当 CPU 单元在 MONITOR 或 PROGRAM 模式下运行时，可直接使用 CX-Programmer 的联机编辑功能在 CPU 单元中添加或修改部分程序。CX-Programmer 一次可添加或修改一个或多个程序段。因此，该功能适于在 CPU 单元运行的情况下对程序进行小幅修改。

对于不同的编辑任务，可在多台计算机上运行 CX-Programmer 并执行联机编辑。



在 MONITOR 模式下对 CPU 单元中的程序进行联机编辑时，整个循环时间将增加至原有时间的一到数倍。若在联机编辑后在闪存中备份数据，循环时间也将增加。此时 BKUP 指示灯将会点亮。备份进度显示在 CX-Programmer 上。每个循环的延迟时间约为 1ms。

在不等待备份完成的情况下，可连续编辑的次数是有限的。实际次数取决于执行的编辑类型，但最多可进行 400 次编辑。

如果超过限制次数，则将在 CX-Programmer 上显示一条信息，且在 CPU 单元完成备份数据之前不能继续编辑。在备份完成后，方可继续进行联机编辑。



正确使用注意事项

在 MONITOR 模式下，若通过联机编辑操作覆写程序，循环时间将比正常情况来得长，因此必须确保延长的时间没有超过在 PLC 设置中设定的循环监控时间。如果延长的时间量超过了监控时间，则会发生循环时间超时并使 CPU 单元停止运行。请在切换为 RUN/MONITOR 模式前，通过选择 PROGRAM 模式来重启 CPU 单元。

注 1 需要进行联机编辑的程序中存储的上升沿微分指令 (DIFU 或带 @ 的指令) 以及下降沿微分指令 (DIFD 或带 % 的指令) 的内部状态 (微分标志) 将被初始化。

注 2 如果进行联机编辑的任务中包含一个块程序，则由 WAIT(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，同时下一轮执行操作将从头开始。

● 通过 CX-Programmer 进行联机编辑

1. 显示需要编辑的程序段。
2. 选择需要编辑的指令。
3. 分别选择 “Program” (程序)、 “Online Edit” (联机编辑) 和 “Begin” (开始)。
4. 编辑指令。

5. 分别选择 “Program” (程序)、 “Online Edit” (联机编辑) 和 “Send Changes” (发送变更)。此时将对指令进行检查。如果没有错误, 指令将被传送到 CPU 单元。CPU 单元中的指令将被覆写, 且此时的循环时间将会延长。

⚠ 注意

仅在确认延长后的循环时间不影响运行后, 方可继续进行联机编辑。如果循环时间太长, 将可能无法输入信号。



附加信息

临时禁止联机编辑

为确保某个循环中机器控制的响应特性, 可在指定的时间内禁止联机编辑。禁止联机编辑可避免因联机编辑而造成的循环时间意外延长。

- 用于禁止联机编辑的设定

下述两种设定都可用于禁止联机编辑。

将联机编辑禁止位确认标志 (A527.00 ~ A527.07) 设定为 5A Hex。

将联机编辑禁止位 (A527.09) 置 ON。

- 禁止联机编辑时的操作

如果 CX-Programmer 发出联机编辑请求, 则系统将进入待机状态而不执行联机编辑。联机编辑等待标志 (A201.10) 将会置 ON。随后, 当用户将联机编辑禁止位 (A527.09) 置 OFF 时, 将执行联机编辑。如果联机编辑处于待命状态, 则任何联机编辑操作都将被忽略。

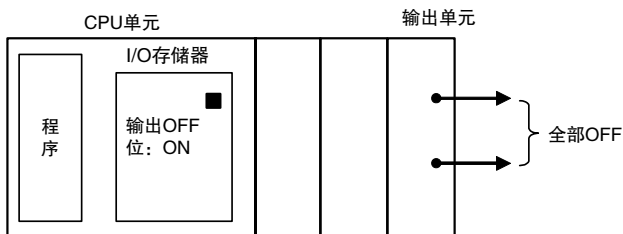
● 相关的辅助位 / 字

名称	地址	描述
联机编辑禁止位 确认标志	A527.00 ~ A527.07	确认联机编辑禁止位 (A527.09) 的有效性。 5A Hex: 联机编辑禁止位有效 非 5A Hex: 联机编辑禁止位无效
联机编辑禁止位	A527.09	若要禁止联机编辑, 请将该位置 ON, 并将联机编辑禁止位确认标志 (A527.00 ~ A527.07) 设定为 5A。
联机编辑等待标志	A201.10	由于禁止联机编辑而导致联机编辑过程进入待命状态时为 ON。

10-7-5 输出置 OFF

将输出 OFF 位 (A500.15) 置 ON 时, 所有输出单元的输出都将变为 OFF, 而与 I/O 存储器中的输出状态无关。当 CPU 单元在 RUN 或 MONITOR 模式下运行时, 该操作可作为出错后的紧急处理措施。此时, CPU 单元前部的 INH 指示灯将亮起黄光。

在使用电池的情况下, 输出 OFF 位的状态将保持, 而与电源的通断无关。





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（即置 OFF）。如果通过用户程序将 I/O 存储器中的输出位置 ON，则 I/O 刷新时输出将变为 ON。无论采用何种运行模式，都应通过输出 OFF 位将输出单元的所有输出置 OFF。

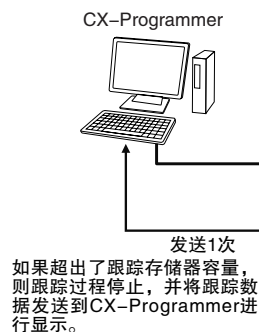
10-7-6 数据跟踪

数据跟踪功能可通过下述任意一种定时方法在特定情况下对指定 I/O 存储器的数据进行采样，并将采样数据存储在跟踪存储器中，随后可用 CX-Programmer 读取和查看这些数据或直接作为文件存储。

可使用下述两种跟踪方法。

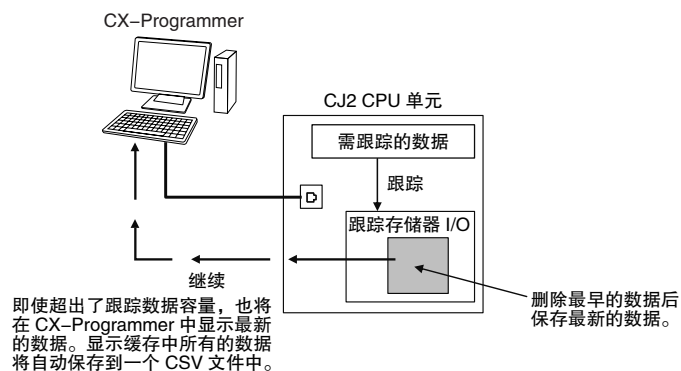
● 常规跟踪

该方法适用于短时间的高速数据采集（如调试）。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，同时将数据存储在一个 CSV 文件中。



● 持续跟踪

持续跟踪适用于长时间的高速数据采集。即使超出了跟踪存储器容量，跟踪过程仍会继续，直至将跟踪数据读取到 CX-Programmer。当显示缓存饱和时，跟踪数据将自动存储到 CSV 文件。通过使用多个 CSV 文件，即可自始至终地存储长时间数据跟踪过程的结果。



● 数据采样和定时

数据大小

I/O 存储器中最多可指定 31 位和 16 个字的数据进行采样。

数据类型

可指定下列数据类型进行采样。

- BOOL(位)
- UINT(二进制, 单字, 无符号)
- UDINT(二进制, 双字, 无符号)
- ULINT(二进制, 四字, 无符号)
- INT(二进制, 单字, 带符号)
- DINT(二进制, 双字, 带符号)
- LINT(二进制, 四字, 带符号)
- UINT BCD(二进制, 单字, 无符号)
- UDINT BCD(二进制, 双字, 无符号)
- ULINT BCD(二进制, 四字, 无符号)
- REAL(双字, 浮点数)
- LREAL(四字, 浮点数)
- CHANNEL(字)
- WORD(十六进制, 单字)
- DWORD(十六进制, 双字)
- LWORD(十六进制, 四字)

定时

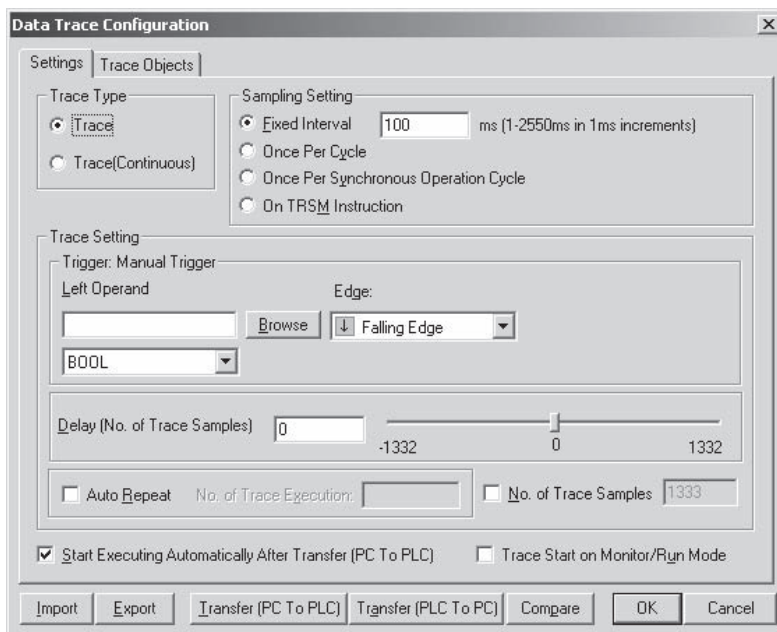
可通过下列定时方式对 I/O 存储器中的数据进行采样。

- 指定采样时间(1 ~ 2,550ms, 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
 - 每个同步运转周期一次(0.5 ~ 10.0ms, 以 0.1ms 为单位)*
- *如果同步运转周期小于 3.0ms, 则无法使用持续跟踪功能。

● 数据跟踪步骤概述

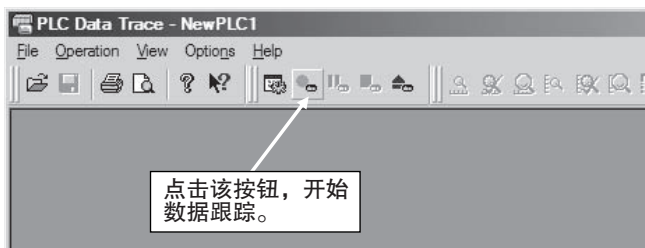
请按照下列步骤执行跟踪。

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 – 时序图监控)以打开数据跟踪配置对话框,然后选择“Execute – Set”(执行 – 设定))。



- 执行常规跟踪
在“Trace Type”(跟踪类型)区中选择“Trace”(跟踪),并在“Data Trace Configuration”(数据跟踪配置)选项页中设定采样数据的地址。此时可指定多个地址。然后在“Settings”(设定)选项页的“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)”(持续跟踪),并在“Data Trace Configuration”(数据跟踪配置)选项页中设定采样数据的地址。此时可指定多个地址。然后在“Settings”(设定)选项页的“Trace Setting”(跟踪设定)区中指定画面显示缓存大小、跟踪时间以及包含采样数据的 CSV 文件的存储路径。

2. 将采样起始位 (A508.15) 置 ON 或点击下列按钮,以启动跟踪过程。

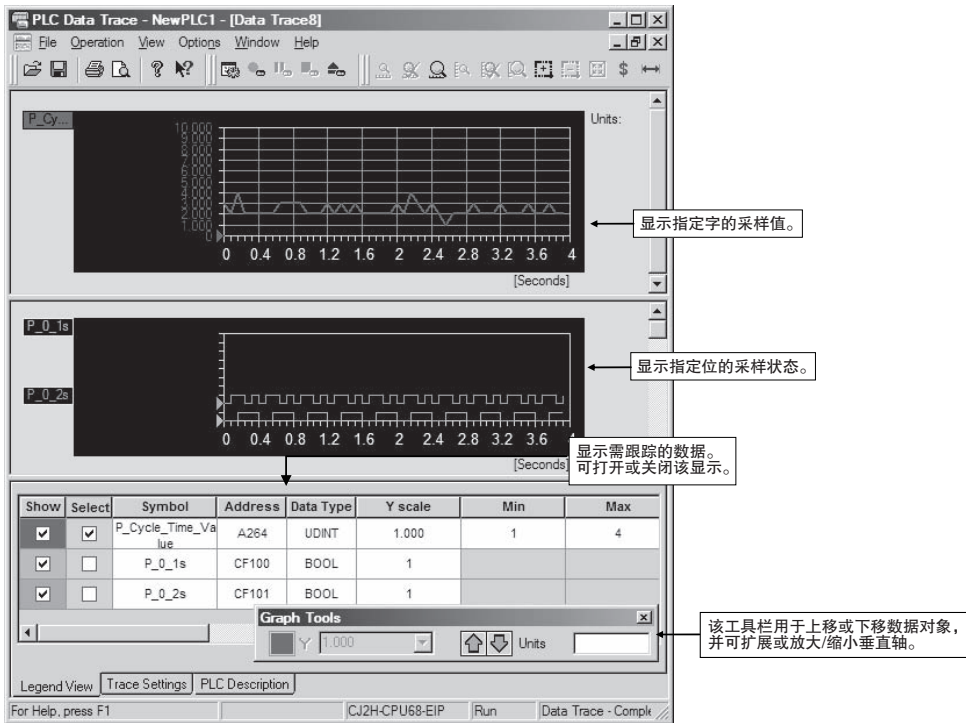


当满足触发条件时，将出现如下画面。



● 数据跟踪窗口

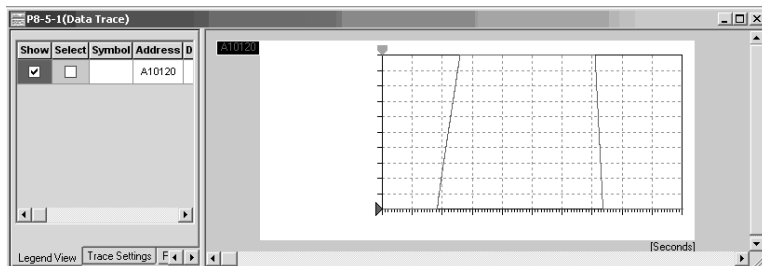
下图为数据跟踪窗口的概览。



有关相关步骤和设定详情，请参考《CX-Programmer 操作手册》(样本编号：W446)。

I/O 模块辅助区选择功能

当脉冲 I/O 模块与 CJ2M CPU 单元组合使用时，该功能可方便地选择与脉冲 I/O 模块相关的当前值 (PV) 和状态。在 9.12 版或更高版本的 CX-Programmer 中，可点击数据跟踪配置选项页中“**I/O Module AR Select**” (I/O 模块辅助区选择) 按钮来显示一个列表，并从中选择需要跟踪的当前值 (PV) 和状态标志。



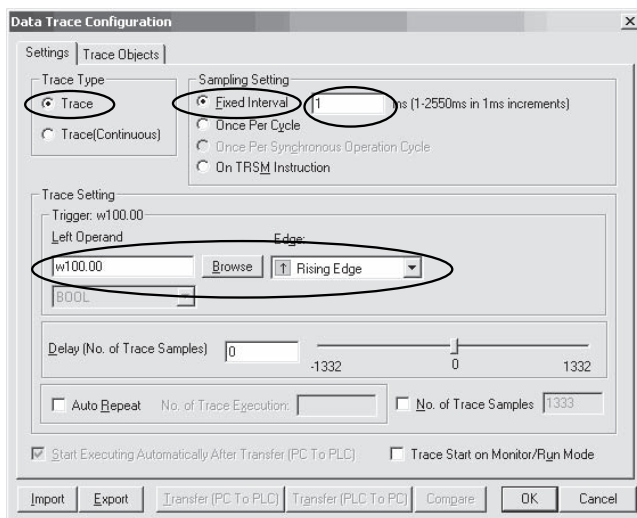
● I/O 模块辅助区选择功能的使用步骤

请按照下列步骤来使用 I/O 模块辅助区选择功能。

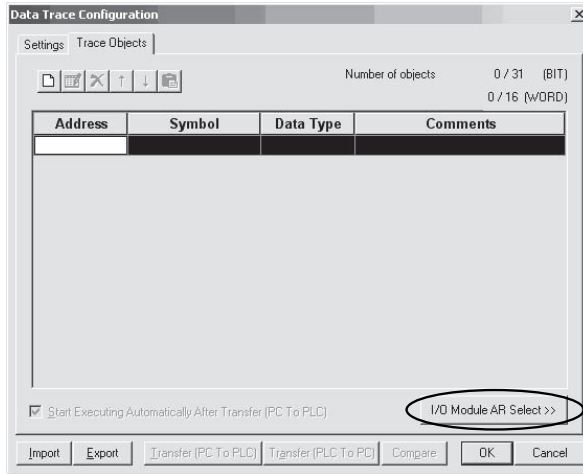
1. 使用 CX-Programmer 设定跟踪参数。

(选择“**PLC - Data Trace**” (PLC - 数据跟踪) 或“**PLC - Time Chart Monitor**” (PLC - 时序图监控) 以打开数据跟踪配置对话框，然后选择“**Execute - Set**” (执行 - 设定))。本例将以 1ms 为周期对脉冲输出频率进行跟踪。“**Settings**” (设定) 选项页中需进行下列设定。

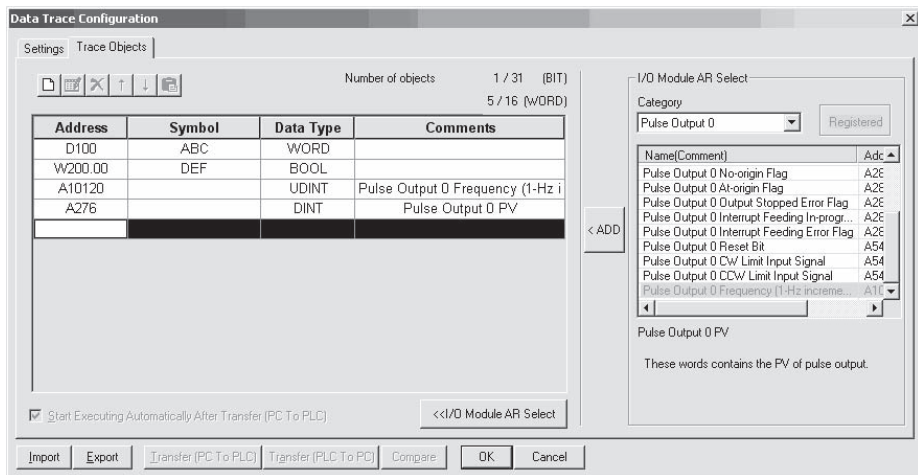
参数	CJ2 CPU 单元
跟踪类型	常规跟踪
采样设定	固定间隔：1ms
跟踪触发条件	输入启动跟踪的条件。



2. 点击“Data Trace Configuration”（数据跟踪配置）选项页中的“I/O Module AR Select”（I/O 模块辅助区选择）按钮。

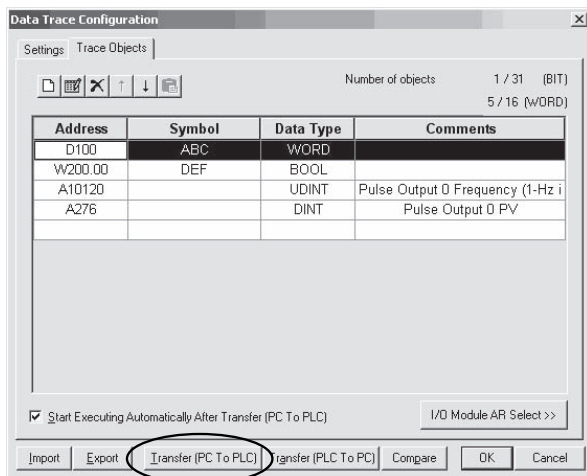


3. 在“Data Trace Configuration”（数据跟踪配置）选项页中选择需要跟踪的脉冲输出并点击“Add”（添加）按钮。若要取消跟踪，则选择需要取消跟踪的脉冲输出并点击“Delete”（删除）按钮。

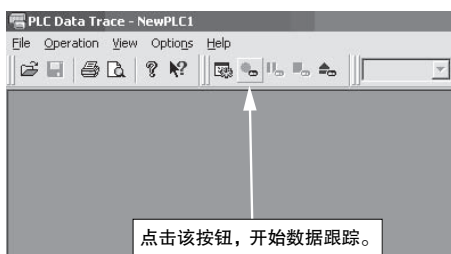


4. 完成设定后点击“OK”（确定）按钮。

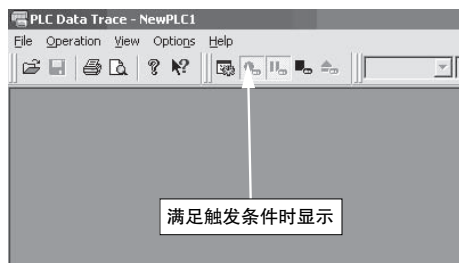
5. 点击 “Transfer (PC to PLC)” (传送 (PC → PLC)) 按钮，将设定值传送至 CPU 单元。



6. 将采样起始位 (A508.15) 置 ON 或点击下列按钮。



当满足触发条件时，将出现如下画面。



● CJ2 CPU 单元的数据跟踪规格

下表对 CJ2 CPU 单元和 CJ1 CPU 单元的数据跟踪规格进行了比较。

项目	CJ2 CPU 单元	CJ2M 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 部分 EM 区可作为数据跟踪 区使用(可容纳跟踪数据的 EM 区最大容量: 32K 字 × 25 个存储区块)。	8,000 字 部分 EM 区可作为数据跟踪 区使用(每 4 个存储区块最 多容纳 32K 字)。	4,000 字
采样大小	采样位数: 31 位 采样字数: 16 字 单字数据: 最多 16 个 双字数据: 最多 8 个 四字数据: 最多 4 个	采样位数: 31 位 采样字数: 最多 16 字 单字数据: 最多 16 个 双字数据: 最多 8 个 四字数据: 最多 4 个	采样位数: 31 位 采样字数: 最多 6 字
循环时间(恒定状态)	1 ~ 2,550ms (以 1ms 为增量)	1 ~ 2,550ms (以 1ms 为增量)	10 ~ 2,550ms (以 10ms 为增量)
开机时自动开始	支持	支持	不支持
连续跟踪	支持 在超出跟踪存储器容量 前,可将跟踪数据读取至 CX-Programmer 来实现连续 跟踪。	支持 在超出跟踪存储器容量 前,可将跟踪数据读取至 CX-Programmer 来实现连续 跟踪。	不支持
用于设定触发条件的数据长度	数据长度: 1 字、2 字或 4 字	数据长度: 1 字、2 字或 4 字	1 字
触发条件	可使用 =、>、≥、<、≤ 和 ≠。	可使用 =、>、≥、<、≤ 和 ≠。	仅可使用 =。
延迟值	-32,768 ~ 32,767	-32,768 ~ 32,767	-1,999 ~ 2,000

10-7-7 存储出错停止位

如果梯形图程序中的指令不在输入参数的范围内, 出错标志 (P_ER) 将会置 ON。执行指令时, 可通过该标志检查指令错误。

如果在 PLC 设置中勾选了“Stop CPU on Instruction Error”(发生指令错误时停止 CPU) 复选框, 则在发生下列任意一种指令错误的情况下, 将导致程序出错及停止运行。该功能可用于确认是否发生了指令错误。

● 导致程序出错的指令错误 (A295)

名称	地址	描述
指令处理出错标志	A295.08	当发生指令处理错误且 PLC 设置已设为在发生指令错误时停止运行时, 该标志和出错标志 (P_ER) 将置 ON。
间接 DM/EM BCD 出错标志	A295.09	当发生间接 DM/EM BCD 错误且 PLC 设置已设为在发生间接 DM/EM BCD 错误的情况下停止运行时, 该标志将置 ON。
非法访问出错标志	A295.10	当发生非法访问错误时(如对系统区进行访问、对转换为 EM 文件的区域进行读写操作、对写保护区进行写操作或发生间接 DM/EM BCD 错误时), 该标志将置 ON。

当因程序出错导致停止运行时, 将按下文所述存储停止位置。

● 程序出错任务 (A294)

名称	地址	描述
程序出错任务	A294	表示因程序出错导致程序执行停止时正在执行的任务的类型和编号。 0000 ~ 007F Hex(对应任务 0 ~ 127) 8000 ~ 80FF Hex(对应中断任务 0 ~ 255, 包括附加的循环任务)

可用于检查发生致命错误的任务。当清除致命错误时，程序出错任务也将被清除。

● 程序出错位置 (A299: 高字节, A298: 低字节)

如果因程序出错导致执行停止，则将存储下列停止位置对应的程序地址。

注 程序地址仅对梯形图程序有效，不可用于 ST 或 SFC 语言程序。

- 梯形图程序中的错误：
从梯形图程序的开头计算地址。
- 梯形图动作程序中的错误：
从梯形图动作程序的开头计算地址。
- 转换程序中的错误：
从转换程序的开头计算地址。
- 功能块程序中的错误：
从功能块程序的开头计算地址。

10-7-8 故障报警指令

FAL(006) 和 FALS(007) 指令用于产生用户自定义错误。FAL(006) 可产生一个非致命错误，而 FALS(007) 可产生一个停止程序执行的致命错误。

当满足用户自定义错误条件 (FAL(006) 或 FALS(007) 执行条件) 时，将执行故障报警指令及下列处理。

- FAL 出错标志 (A402.15) 或 FALS 出错标志 (A401.06) 置 ON。
- 将相应的出错代码写入辅助区的 A400 中。
- 将出错代码和发生时间存储在出错日志中。
- CPU 单元前部的错误指示灯闪烁或点亮。

执行 FAL(006) 指令后，CPU 单元将继续运行。

执行 FALS(007) 指令后，CPU 单元将停止运行。(程序执行将会停止。)

FAL(006) 的操作



当执行条件 A 变为 ON 时，将产生一个 FAL 编号为 002 的错误，且 A402.15(FAL 出错标志) 和 A360.02(FAL 编号为 002 的标志) 置 ON。此时继续执行程序。

通过执行 FAL 编号为 00 的 FAL(006) 指令或使用 CX-Programmer 执行错误读取 / 清除操作，可清除由 FAL(006) 产生的错误。

FALS(007) 的操作



当执行条件 B 变为 ON 时，将产生一个 FALS 编号为 003 的错误，且 A401.06(FALS 出错标志)置 ON。此时停止执行程序。

排除错误成因，并使用 CX-Programmer 执行错误读取 / 清除操作，方可清除由 FALS(007) 产生的错误。

10-7-9 模拟系统错误

可使用 FAL(006) 和 FALS(007) 指令模拟致命或非致命系统错误。进行系统调试时，该功能可在可编程终端 (PT) 或其它操作接口上测试显示信息。

请按照下列步骤进行操作。

1. 在 A529 中将 FAL 或 FALS 编号设定为“模拟使用”。(即在模拟 FAL(006) 和 FALS(007) 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) 的第一操作数。
3. 将出错代码和错误设定为“模拟使用”，作为 FAL(006) 或 FALS(007) 的第二操作数 (S 和 S+1)。这样即可为 FAL(006) 和 FALS(007) 分别指定一个非致命错误和致命错误。

若要模拟多个系统错误，则应在 A529 中指定相同的值作为第一操作数，并多次使用 FAL(006) 或 FALS(007) 指令 (带有不同的第二操作数)。



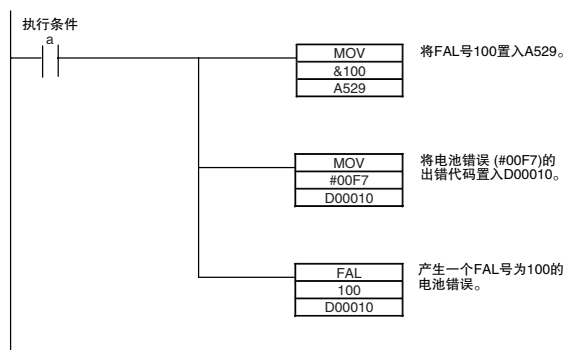
正确使用注意事项

该功能通过在 CPU 单元中模拟一个错误并进行检测，以确认各类应用场合中的检测功能是否正常。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	描述
用于模拟系统错误的 FAL/FALS 编号	A529	设置一个虚拟的 FAL/FALS 编号来模拟系统错误。 0001 ~ 01FF Hex: FAL/FALS 编号 1 ~ 511 0000 或 0200 ~ FFFF Hex: 无用于模拟系统错误的 FAL/FALS 编号。

● 电池错误示例



正确使用注意事项

若要清除模拟的致命和非致命系统错误，可执行 FAL(006) 和 FALS(007) 指令，具体方法与处理实际系统错误时相同。有关清除错误的方法，请参考《CJ2 CPU 单元硬件操作手册》(样本编号: W472) 中的“第 6 章 故障诊断”。所有由 FAL(006) 和 FALS(007) 模拟出的系统错误均可通过重启电源的方式来清除。

10-7-10 故障点检测

执行 FPD(269) 指令可实现时间监控和逻辑诊断。在特定的监控时间内，如果诊断输出没有变为 ON，则时间监控功能会产生一个非致命错误。而逻辑诊断功能用于指示阻止诊断输出变为 ON 的输入点。

时间监控功能

在特定的监控时间内，如果诊断输出没有变为 ON，则执行 FPD(269) 并使进位标志置 ON，从而启动计时。进位标志可通过编程操作作为一个错误处理块的执行条件。FPD(269) 也可通过编程操作生成一个带有所需 FAL 编号的非致命 FAL 错误。

产生 FAL 错误时，当前信息将被记录并在 CX-Programmer 上显示。对 FPD(269) 进行设定，从而在发出信息前输出逻辑诊断的结果 (防止诊断输出变为 ON 的位地址)。

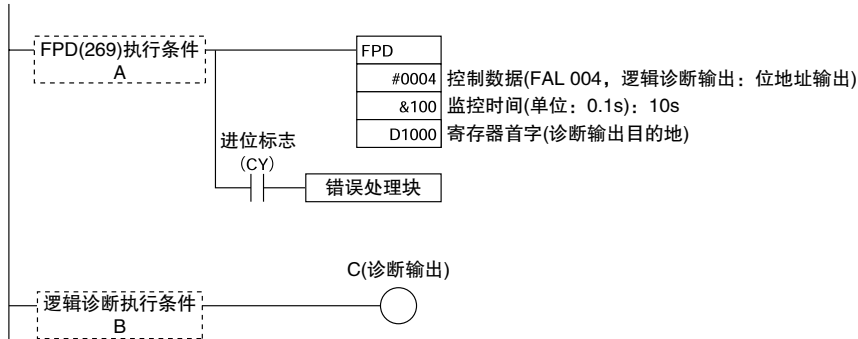
可通过教学功能自动确定诊断输出变为 ON 所需的实际时间及设定监控时间。

逻辑诊断功能

可通过 FPD(269) 确定导致诊断输出保持 OFF 状态的输入位，并输出该位的地址。输出可以设定为位地址输出 (PLC 存储器地址) 或信息输出 (ASCII)。

如果选择了位地址输出，可将位的 PLC 存储器地址传送到一个变址寄存器中，并在以后的处理中对变址寄存器进行间接寻址。

如果选择了信息输出，将以 ASCII 信息的形式记录位地址并显示在 CX-Programmer 中。



● 时间监控:

输入 A 后, 在 10 秒内监控输出 C 是否变为 ON。如果 C 在 10 秒内没有变为 ON, 则检测出故障并将进位标志置 ON。然后由进位标志执行错误处理快, 同时产生一个 FAL 编号为 004 的 FAL 错误 (非致命错误)。

● 逻辑诊断:

通过 FPD(269) 确定逻辑块 B 中阻止输出 C 变为 ON 的输入位, 并将该位的地址输出至 D1000。

● 辅助区标志和字

名称	地址	描述
出错代码	A400	发生错误时, 其对应的出错代码被存储在 A400 中。
FAL 出错标志	A402.15	执行 FAL(006) 指令时为 ON。
FALS 出错标志	A401.06	执行 FALS(007) 指令时为 ON。
已执行的 FAL 编号标志	A360 ~ A391	当发生 FAL(006) 或 FALS(007) 错误时, 对应的标志置 ON。
出错日志区	A100 ~ A199	出错日志区中包含最近发生的 20 个错误的信息。
出错日志指针	A300	当发生错误时, 出错日志指针递增 1 以指向下一条出错记录的记录位置, 该位置以距离出错日志区 (A100) 的起始位置的偏移量来表示。
出错日志指针重置位	A500.14	此位置 ON 以将出错日志指针 (A300) 重置为 00。
FPD 学习位	A598.00	执行 FPD(269) 时, 若要自动设定监控时间, 则应将该位置 ON。

10-8 单元同步运转

10-8-1 概述

单元同步运转功能

单元同步运转功能将 CPU 单元产生的同步信号作为一个指定周期，从而同步各 CPU 单元和同步单元的处理启动及这些单元之间的数据交换操作。同步单元是指支持“单元同步运转功能”的 CPU 总线单元和高功能 I/O 单元。

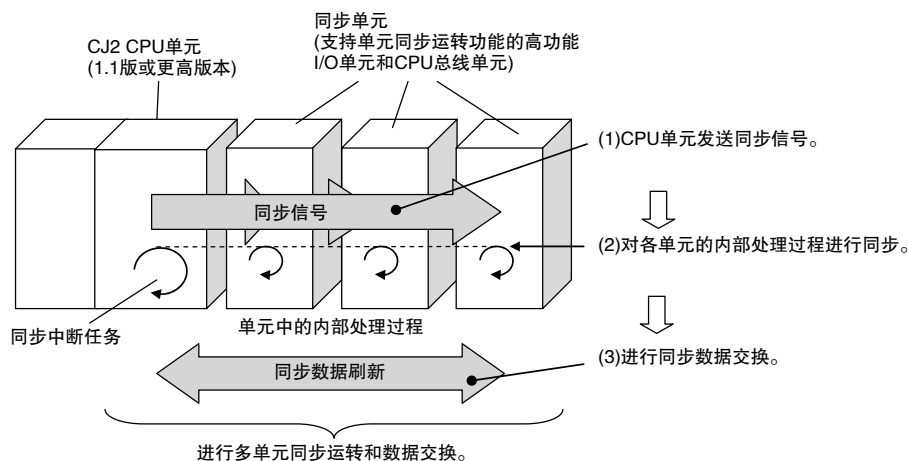
仅 1.1 版或更高版本的 CJ2H CPU 单元支持此项功能。

内部处理同步

为了与同步单元之间的数据交换实现同步，CPU 单元会执行一个中断任务。同步单元利用 CPU 单元发出的同步信号来同步启动所有单元的内部处理过程。

数据交换同步

CPU 单元与多台同步单元之间或同步单元之间在同步信号发出后进行数据交换。



正确使用注意事项

若要使用单元同步运转功能，必须同时满足下列所有要求。

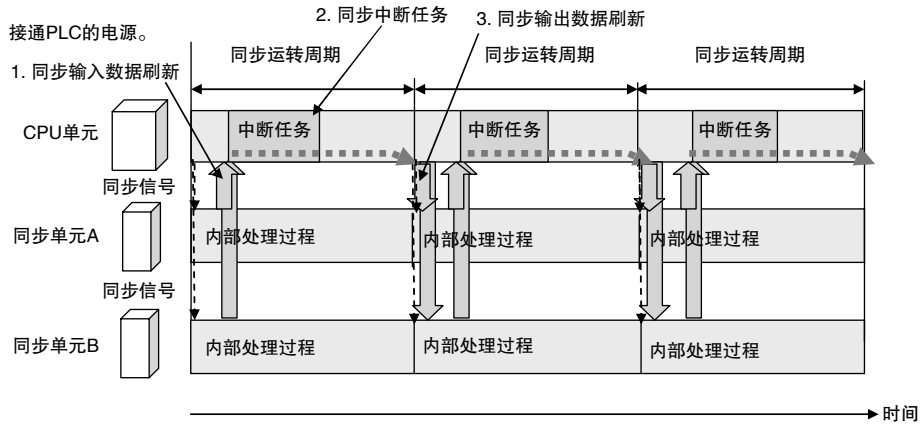
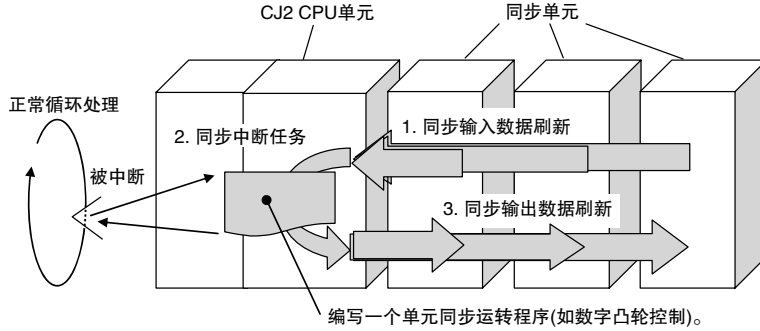
- 必须使用 1.1 版或更高版本的 CJ2H-CPU6□(-EIP)CPU 单元。
- 高功能 I/O 单元和 CPU 总线单元必须支持单元同步运转功能。(这些单位称为同步单元。)

单元同步运转的大致流程

单元同步运转是按照以下流程执行的。

1. 同步单元之间执行内部处理同步，并每隔一个同步运转周期将同步输入数据发送至 CPU 单元。
2. CPU 单元将同步数据存储至存储器，并执行针对单元同步运转的中断任务。该中断任务称为“同步中断任务”。

3. CPU 单元将同步中断任务的结果作为同步输出数据发送至所有的同步单元。



项目	描述
同步数据刷新	在同步运转周期内，CPU 单元和同步单元年之间进行同步数据刷新，且该过程不会受到 CPU 单元的正常循环时间的影响。同步数据刷新操作会占用一个被称为“同步数据刷新区”的特殊区域。该区域中的数据将作为同步数据在单元之间进行交换。
同步输入数据刷新	同步单元将同步数据刷新操作的输入数据发送至 CPU 单元。
同步输出数据刷新	CPU 单元将同步数据刷新操作的输出数据发送至同步单元。
同步运转周期	同步运转周期用于产生同步信号，从而对同步单元的内部处理时间点进行同步。该周期还可用于刷新同步数据。
同步中断任务	CPU 单元接收到同步数据刷新操作的输入数据后，将执行同步中断任务。中断任务 2(定时中断 0) 用于同步中断任务。 在不使用同步中断任务的情况下，也可执行单元同步运转。

单元同步运转设定概述

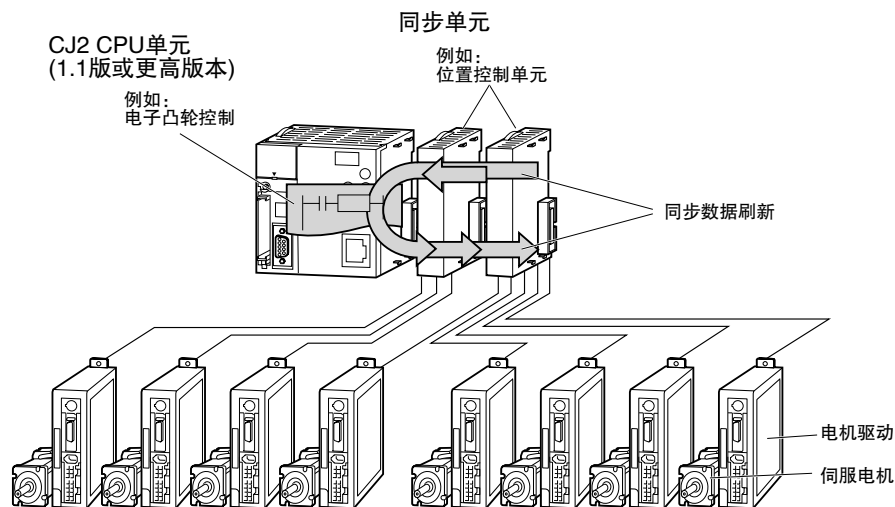
下表所示为同步单元运转所需的设定。

项目	描述	设定方法
单元同步运转	可允许和禁止同步单元运转功能。	PLC 设置
同步数据刷新	<ul style="list-style-type: none"> 必须注册单元同步运转中使用的同步单元。 必须分配同步数据刷新区。 必须为各同步单元设定同步数据。	PLC 设置 可通过 PLC 设置访问各同步单元的同步数据设定。
同步运转周期	设定同步周期时间。	PLC 设置
同步中断任务	执行该任务中的程序。	选择程序属性中的中断任务 2(定时中断 0)。 在不使用同步中断任务的情况下,也可执行单元同步运转。

应用示例

● 伺服电机同步运转

本例中,部分伺服电机可通过操作编码器实现同步运转。此时,由高速计数器单元输入至 CPU 单元的同步数据将在同步中断任务的同步运转程序(如数字凸轮程序)中进行处理,处理结果由 CPU 单元作为同步数据输出至部分位置控制单元,以便在进行位置控制时使用。





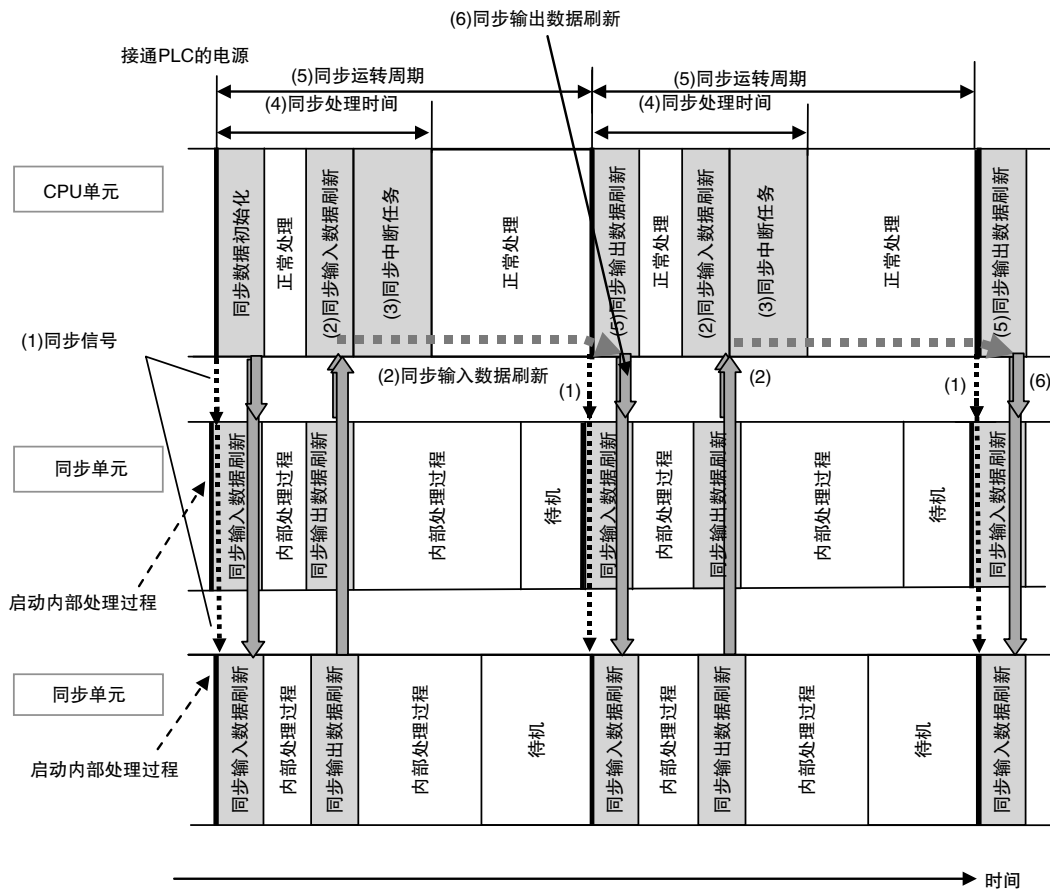
附加信息

CPU 单元和同步单元之间也可进行正常 I/O 刷新操作。下表所示为正常 I/O 刷新与同步数据刷新之间的区别。

项目	同步数据刷新	正常 I/O 刷新
数据交换时间	<ul style="list-style-type: none"> · 每个同步运转周期 (中断 CPU 单元的正常循环来执行同步数据刷新。) · 数据交换过程与同步单元的内部处理过程保持同步。 	<ul style="list-style-type: none"> · CPU 单元的每个正常循环 (执行用户程序后)。 · 正常 I/O 刷新的时间与同步单元中内部处理过程的时间无关。
数据交换周期	恒定 (始终等于指定的同步运转周期)	不恒定 (在设定了最小周期时间且周期时间不超过预设时间的情况下, 等于最小周期时间)。
同步单元的内部处理过程启动	同步	不同步

10-8-2 有关单元同步运转的详细信息

下图所示为同步单元运转的时序详解。



1. 同步信号
当 PLC 通电时，CPU 单元会产生一个同步信号，然后开始同步数据的初始化操作。与此同时，CPU 单元和同步单元开始内部处理过程。
2. 同步输入数据刷新 (同步单元→CPU 单元)
同步单元将同步数据发送至 CPU 单元。
3. 同步中断任务
CPU 单元执行同步中断任务。(在不使用同步中断任务的情况下，也可执行单元同步运转。)
4. 同步处理时间
同步处理时间为在一个同步运转周期中完成一系列同步运转处理所需的时间。该时间从产生同步信号时开始，直至同步中断任务执行完成后结束。同步处理时间过后，将执行正常处理过程直至同步控制周期结束。
同步处理时间必须小于同步运转周期。(在 CX-Programmer 的 “Synchronous Operation Status” (同步运转状态) 对话框中，可对同步处理时间的最大值和当前值进行监控。)
5. 同步运转周期
当同步运转周期结束时，将会再次产生一个同步信号。
6. 同步输出数据刷新 (CPU 单元→同步单元)
CPU 单元将同步数据发送至同步单元。
7. 从步骤 2 开始重复整个过程。



正确使用注意事项

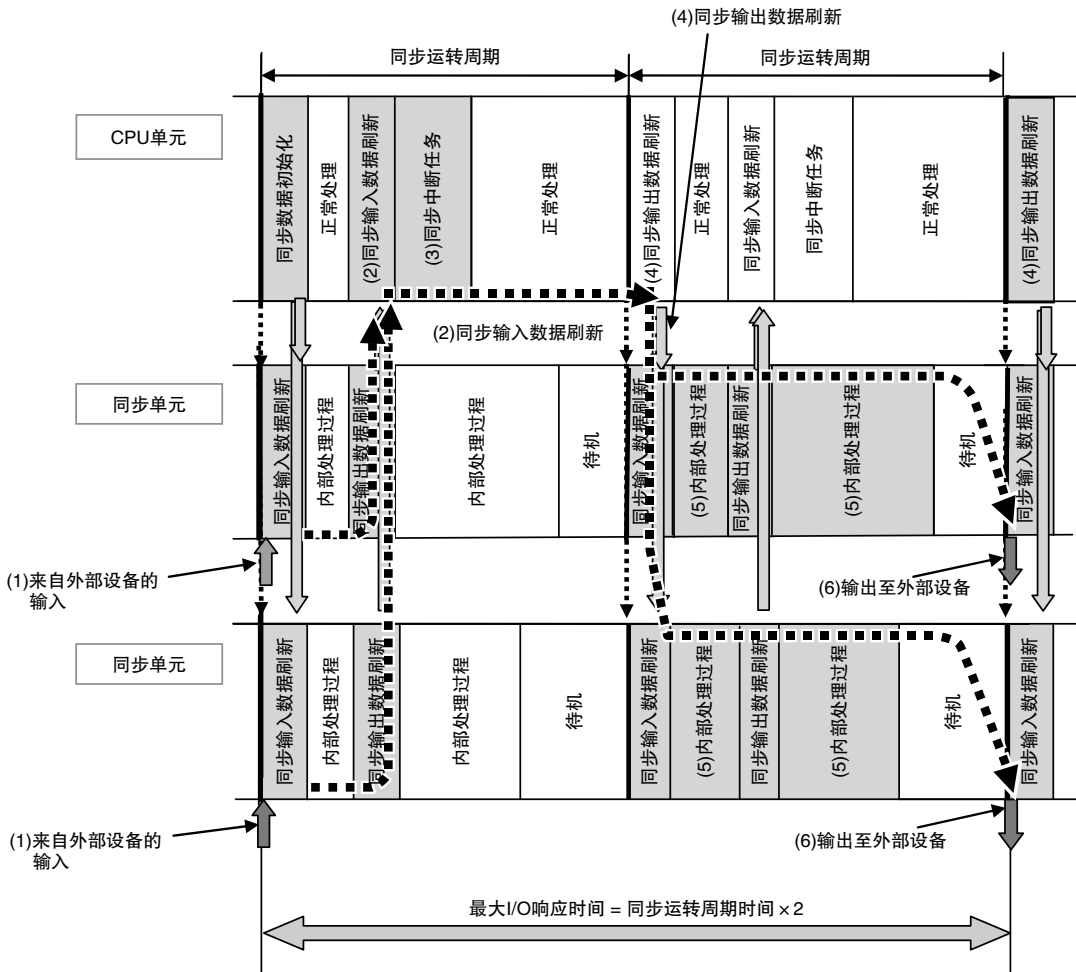
CPU 单元会在同步输入数据刷新、同步中断任务执行及同步输出数据刷新期间中断正常处理过程。可将 CPU 单元的正常周期时间延长至处理这些过程所需的时间。但在使用单元同步运转功能前，应确保延长的周期时间不会对控制系统产生不利影响。

● 同步单元的 I/O 响应时间

同步单元的 I/O 响应时间等于处理下述过程所需时间的总和。

- (1) 从外部设备输入至同步单元
- (2) 同步输入数据刷新 (同步单元 → CPU 单元)
- (3) 同步中断任务执行
- (4) 同步输出数据刷新 (CPU 单元 → 同步单元)
- (5) 同步单元中的内部处理过程
- (6) 从同步单元输出至外部设备

最大 I/O 响应时间相当于同步运转周期时间的两倍。



CJ1W-NC□□4 位置控制单元示例

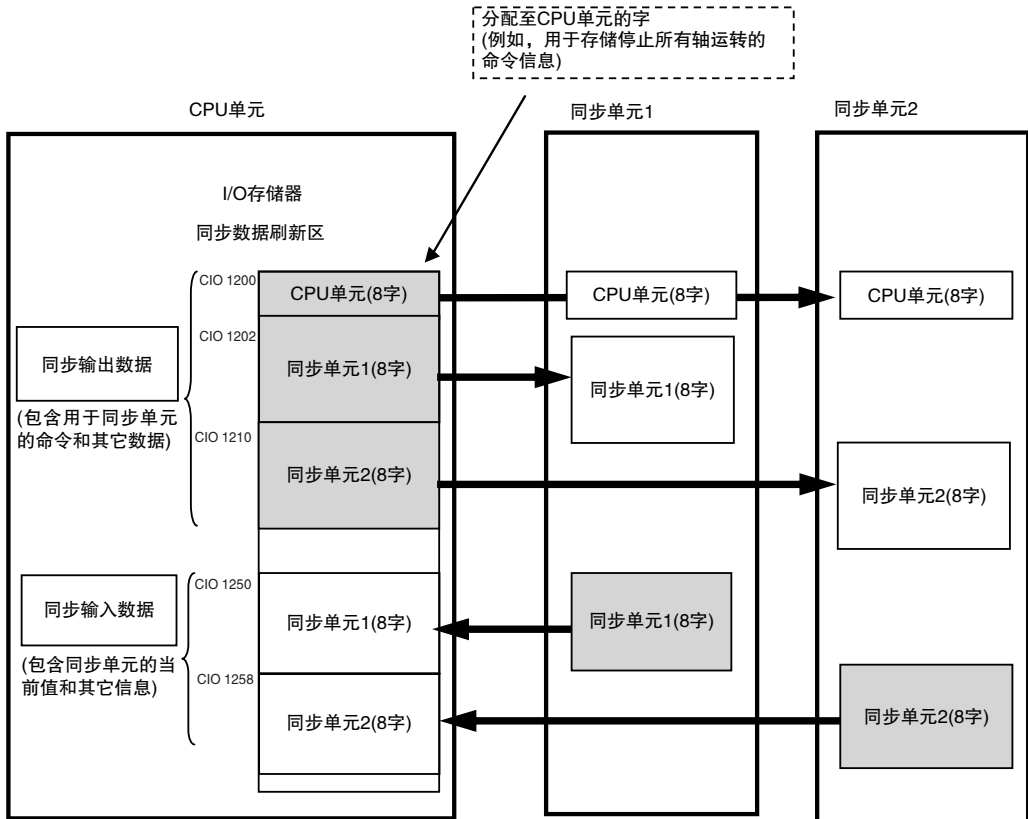
本位置控制单元的内部处理时间为 1ms，因此同步运转周期时间不得小于 1ms，从而推得本位置控制单元的 I/O 响应时间始终为 2ms。

10-8-3 单元同步运转规格

项目	规格
适用 CPU 单元	1.1 版或更高版本的 CJ2H-CPU6□(-EIP)CPU 单元
可用作同步单元的高功能 I/O 单元和 CPU 总线单元	CJ1W-NC□□4 位置控制单元
可安装的同步单元数	最多 10 个 最多 5 个 CJ1W-NC□□4 位置控制单元
同步单元的安装位置	同步单元必须安装在 CPU 机架上。(如果安装在扩展机架上,同步单元将无法发挥作用。)
同步运转周期时间	0.5 ~ 10ms(以 0.1ms 为增量单位) 可在 PLC 设置中设定。 必须满足下列条件: · 同步处理时间必须小于同步运转周期时间。 · 同步单元的内部处理时间必须小于同步运转周期时间。
同步运转周期时间的最大波动量	10 μ s
用于进行同步数据交换的 CPU 单元 I/O 存储区	同步数据刷新区被划分为两部分,分别用于存放输入数据(同步单元→CPU 单元)和输出数据(CPU 单元→同步单元)。同步数据刷新区的地址范围为 CIO 1200 ~ CIO 1295,其中的内容取决于所用的同步单元。
同步中断任务的数量	必须使用中断任务 2。 在不使用同步中断任务的情况下,也可执行单元同步运转。

10-8-4 同步数据刷新

CPU 单元的 CIO 区中的同步数据刷新区的起始地址始终为 CIO 1200。首先由 CPU 单元将输出数据发送至同步单元，随后再由同步单元将输入数据发送至 CPU 单元。



请在 PLC 设置的“Timings/Synchronous” (定时/同步) 选项页的“Synchronous Unit Operation Settings” (单元同步运转设定) 区中执行下述设定。

- 输出数据和输入数据的总计大小
- 各单元的输入和输出大小
- 输入区的起始地址

同步数据刷新区的地址范围为 CIO 1200 ~ CIO 1295。该区域中的首字始终为 CIO 1200，因此仅可设定其它地址。

I/O	项目	描述
输出数据	方向	CPU 单元→同步单元
	起始字	CIO 1200(固定)
	分配	PLC 设置中指定的数据大小被依次分配到 CPU 单元和同步单元 (按照同步单元的注册顺序)。CPU 单元可分配到两个或两个以上的字，而每个同步单元可分配到零个或多个字。
输入数据	方向	同步单元→CPU 单元
	起始字	起始字的地址可在 PLC 设置中设定，其范围在 CIO 1202 ~ CIO 1294 之间。
	分配	PLC 设置中指定的数据大小被依次分配到同步单元 (按照同步单元的注册顺序)。每个同步单元可分配到 0 ~ 16 个字。

● 分配示例

同步数据中的输入数据和输出数据的字分配示例

为同步数据刷新操作设定起始地址和数据大小。(本例使用单元 0 和单元 1。)

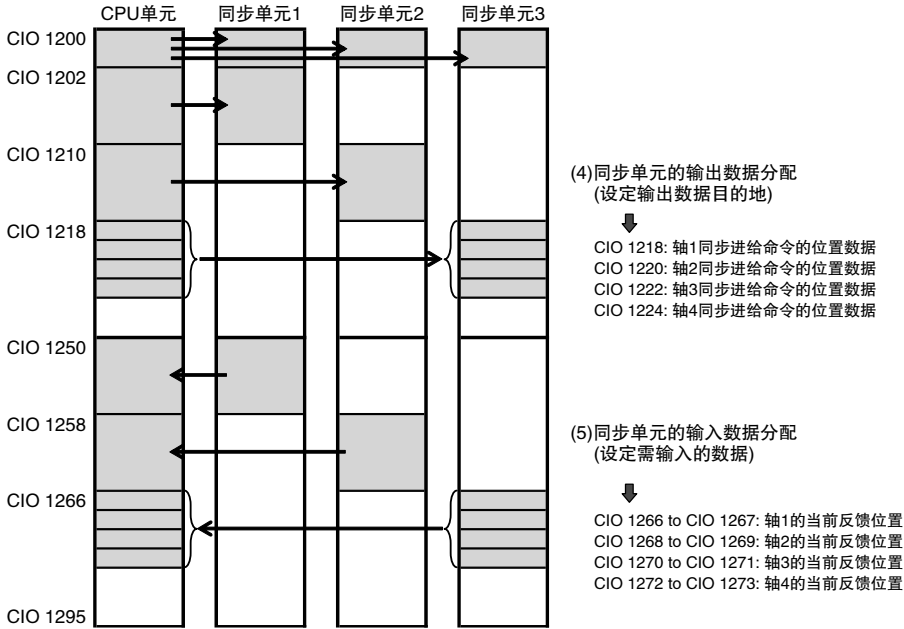
地址	项目	I/O
CIO 1200	分配至 CPU 单元的字 (数据大小设定：2 个字)	从 CPU 单元输出到同步单元
CIO 1201		
CIO 1202		
CIO 1203		
CIO 1204		
CIO 1205		
CIO 1206		
CIO 1207	分配至同步单元 0 的字 (数据大小设定：2 个字)	从同步单元输入到 CPU 单元
:		
CIO 1250		
CIO 1251		
CIO 1252		
CIO 1253		
CIO 1254		
CIO 1255	分配至同步单元 1 的字 (数据大小设定：2 个字)	
:		
CIO 1295		
	...	

使用位置控制单元进行同步数据刷新时所用 I/O 数据的设定示例

必须按照应用类型设定所分配的 I/O 数据。

- 输出数据
对于位置控制单元，应为各轴设定同步进给命令对应的位置数据所使用的字。
- 输入数据
对于位置控制单元，应为各轴设定当前命令值或当前反馈位置。

设定示例



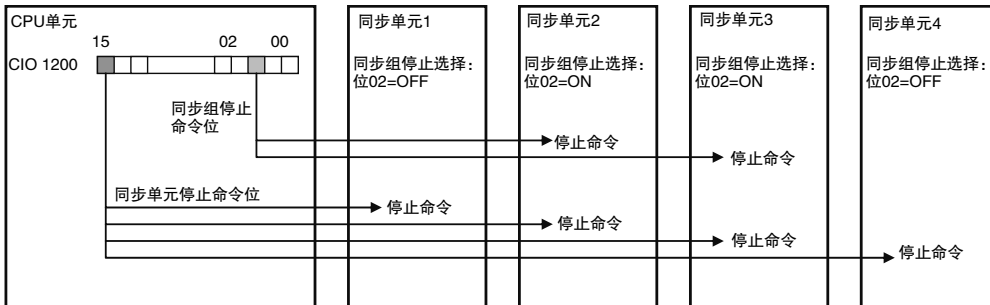
● 应用示例

对于位置控制单元、高速计数器单元和同步单元，请按下表所示使用 I/O 数据。

数据	应用示例
分配至 CPU 单元的字	同步组停止命令位、同步单元停止命令位 * 等
同步数据刷新操作的输出数据	目标位置数据 (X、Y、Z 和 U 轴)
同步数据刷新操作的输入数据	当前反馈数据 (X、Y、Z 和 U 轴)

* 同步组停止命令位和同步单元停止命令位

可使用 CPU 单元中的命令位来停止某些功能 (即针对单元同步运转的同步单元当前所执行的功能)。有两种命令位可供选择。同步单元停止命令位适用于所有同步单元，而同步组停止命令位适用于预设组的同步单元。



同步单元停止命令位

同步单元停止命令位对应同步数据刷新区中 CIO 1200 的位 15。如果该位置 ON，将会发送一条命令，以终止所有同步单元当前所执行的操作。

同步组停止命令位

同步组停止命令位对应同步数据刷新区中 CIO 1200 的位 00 ~ 位 14。如果其中的某一位置 ON，且同步组停止命令位设定中该位的状态也为 ON，将会发送一条命令，以终止所有同步单元当前所执行的操作。

下表所示为位置控制单元的轴参数中的同步组停止选择参数设定。

名称	位	功能	设定	默认值
同步组停止选择	00	这些位表示同步数据刷新区的 CIO 1200 中各状态位对应的同步单元运转方式。 OFF: 忽略位 00 ON: 位 00 置 ON 时停止	OFF, ON	OFF
	01	OFF: 忽略位 01 ON: 位 01 置 ON 时停止	OFF, ON	OFF
	02	OFF: 忽略位 02 ON: 位 02 置 ON 时停止	OFF, ON	OFF
	:	:		
	14	OFF: 忽略位 14 ON: 位 14 置 ON 时停止	OFF, ON	OFF
	15	---	---	---

10-8-5 单元同步运转的使用限制

● 对同步运转周期时间的限制

设定同步运转周期时间，以同时满足下列的两个条件。

1. 同步处理时间必须小于同步运转周期时间。
* 一般来说，应将同步运转周期时间设定为同步处理时间的 1.5 倍或更长。
2. 同步单元的内部处理时间必须小于同步运转周期时间。

1. 同步处理时间

同步处理时间等于下列两种时间的总和。

- (1) 从产生同步信号时开始，直至同步中断任务执行完成后的时间（即下列时间的总和：同步输出数据刷新时间、正常处理时间、同步输入数据刷新时间和同步输入任务执行时间）
- (2) 在第(1)条所述的处理过程中，执行任何 I/O 中断任务或外部中断任务所需的时间（在 CX-Programmer 的“Synchronous Operation Status”（同步运转状态）对话框中，可对同步处理时间的最大值和当前值进行监控。）

2. 同步单元的内部处理时间

CJ1W-NC□□4 位置控制单元的内部处理时间为 1ms，因此同步运转周期时间不得小于 1ms。

3. 无法满足上述条件

同步运转周期时间小于同步处理时间

在执行同步中断任务、I/O 中断任务或外部中断任务时，将进入下一个同步运转周期。这样会导致 CPU 单元出现同步处理超时错误。一旦进入该同步运转周期，将不会执行同步中断任务和同步数据刷新。

同步运转周期时间小于同步单元的内部处理时间

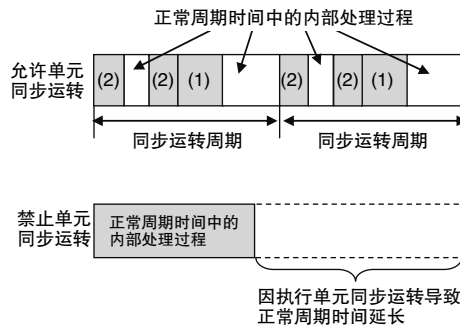
在同步单元完成内部处理前，将进入下一个同步运转周期。这样会导致同步单元出现同步处理超时错误，而同步单元将继续进行相关处理（例如，位置控制单元将继续进行轴操作）。

● 正常周期时间的延长

进行单元同步运转时，下列时间将会延长正常周期时间。

- (1) 同步中断时间对应的中断时间（与正常定时中断的情况相同）
- (2) 同步输入和输出刷新时间

注 上述时间等于同步处理时间减去当时所需的处理时间所得的值。



● 禁止定时中断任务 0 和 1

进行单元同步运转时，将无法使用定时中断 0 和 1 执行中断任务。因此，定时中断任务 0(中断任务 2) 和定时中断任务 1(中断任务 3) 是无法使用的，但中断任务 2 可用作同步中断任务。

● 对中断控制指令的限制

同步中断任务中无法使用 MSKS、MSKR 和 CLI 指令，仅可使用 DI 和 EI 指令。

● 对定时器指令的限制

使用 100ms 定时器 (TIM/TIMX(550))、10ms 定时器 (TIMH(015)/TIMHX(551)) 和 1ms 定时器 (TMHH(540)/TMHHX(552)) 指令时会受到以下限制。

- 定时器当前值 (PV) 的精度将产生最多达一个循环时间的误差。
- 如果循环时间超过 100ms，则定时器将无法正确运行。
- 如果上述任意一条指令在某个停止的任务中，或者被 JMP(004)、CJMP(510) 或 CJPN(511) 指令跳转而未执行，则定时器将无法正确运行。

● 在启用高速中断功能时对中断任务中所用指令的限制

如果启用了单元同步运转功能，则将在 PLC 设置中无条件启用高速中断功能。这样，部分指令（如网络通信指令）是无法在任何中断任务（包括同步中断任务）中使用的。详情请参考“10-2-6 高速中断功能”。

● 启用高速中断功能时的其它限制

- 中断任务中无法使用后台处理方式。
- 下列辅助区字中的数据将变为无效：A440(中断任务最长处理时间)和 A441(处理时间最长的中断任务)。
- 由于使用 EtherNet/IP 标签数据链接而延长的周期时间将会变得 longer。

● 同步单元的安装位置

同步单元必须安装在 CPU 机架上，而不得安装在扩展机架上。

10-8-6 应用步骤

使用单元同步运转功能的步骤如下所述。

1. 初始设定 (请参考“10-8-7 PLC 设置”。)

请在 CX-Programmer 的 PLC 设置的“Timings/Synchronous”(定时/同步)标签页中进行同步单元运转设定。

- (1) 选择“Use Synchronous Operation”(使用同步运转功能)复选框,以启用单元同步运转功能。
- (2) 设定同步运转周期时间。
- (3) 设定用于和同步数据刷新过程交换数据的字。
 - (a) 在“Synchronous Data Refresh Area Allocation”(同步数据刷新区分配)区中设定输入数据的起始地址。
 - (b) 注册同步单元。
 - (c) 设定各同步单元的输入和输出数据的起始地址和大小。

2. 同步单元的 I/O 数据分配

为各同步单元分配所用的 I/O 数据。打开 CX-Programmer, 点击 PLC 设置的“Timings/Synchronous”(定时/同步)标签页的“Synchronous Unit Operation Settings”(单元同步运转设定)区中的“Special Unit Setup”(高功能单元设置)按钮,以对各同步单元进行设定。有关具体设定的信息,请参考各同步单元的操作手册。

3. 在需要时编写同步中断任务。(请参考“10-8-8 编写同步中断任务”。)

在 CX-Programmer 中编写同步中断任务程序(针对中断任务 2)。

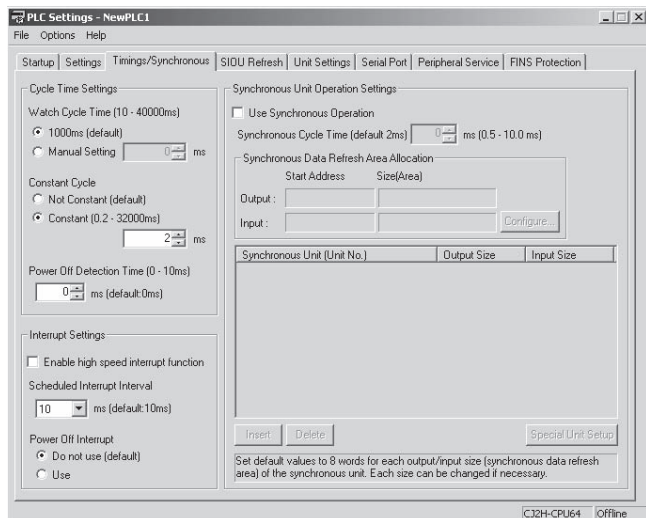
4. 单元同步运转的调试(请参考“10-8-9 与单元同步运转相关的调整和故障诊断作业”。)

在调整同步运转周期时间的同时,可通过 CX-Programmer 的“Synchronous Operation Status”(同步运转状态)对话框监控下列数值。

- 同步处理时间的当前值和最大值
- 同步运转周期时间的设定值
- 正常周期时间的当前值和最大值
- 与同步单元交换数据时发生的错误

10-8-7 PLC 设置

请在 PLC 设置的“Timings/Synchronous” (定时/同步) 选项页的“Synchronous Unit Operation Settings” (单元同步运转设定) 区中设定单元同步运转功能。



● 使用同步运转功能

该复选框用于启用和禁止单元同步运转功能。

参数	设定	默认值	描述	相关辅助区位和字
使用同步运转功能	未勾选时：不使用 勾选时：使用	未勾选时 (不使用)	勾选该复选框以使用单元同步运转功能。 如果启用了单元同步运转功能，则将无条件启用高速中断功能。	---

● 同步周期时间

设定同步运转周期时间。

参数	设定	默认值	描述	相关辅助区位和字
同步周期时间	0.5 ~ 10.0ms (以 0.1ms 为增量)	2.0ms	设定同步运转周期时间。	A10102(同步运转周期时间)



正确使用注意事项

同步运转周期时间必须满足下列条件。

- 同步处理时间必须小于同步运转周期时间。
- 同步单元的内部处理时间必须小于同步运转周期时间。

一般来说，应将同步运转周期时间设定为同步处理时间的 1.5 倍或更长。

● 同步数据刷新区分配

1. 同步单元的输出和输入数据设定

设定同步数据刷新操作的数据大小。

参数	设定	设定	默认值	描述	相关辅助区位和字
输出	起始地址	CIO 1200(固定)	CIO 1200	这是执行同步输出数据刷新时的首字地址。	---
	数据大小(分配区)	2 ~ 96 字 *	50 字	这是执行同步输出数据刷新时的数据大小。	---
输入	起始地址	CIO 1202 ~ CIO 1294 或不使用	CIO 1250	这是执行同步输入数据刷新时的首字地址。	---
	数据大小(分配区)	2 ~ 94字或不设定*	46 字	这是执行同步输入数据刷新时的数据大小。	---

* 设定起始地址时将自动计算数据大小。

2. 数据大小 (分配区)

输出数据

设定执行同步输出数据刷新时的数据大小，并自动计算起始地址。
分别设定各单元的输出大小。

参数	设定	默认值	描述	相关辅助区位和字
输出大小	CPU 单元: 2 字或更大 (直至达到最大值) 同步单元: 0 字或更大 (直至达到最大值)	CPU 单元: 2 字 同步单元: 8 字	设定执行同步输出数据刷新时的数据大小。	---

输入数据

设定执行同步输入数据刷新时的数据大小，并自动计算起始地址。
分别设定各单元的输入大小。

参数	设定	默认值	描述	相关辅助区位和字
输入大小	同步单元: 0 ~ 16 字	同步单元: 8 字	设定执行同步输入数据刷新时的数据大小。	---

● 同步单元 (单元号)

在 PLC 中注册需要执行单元同步运转的同步单元。一般情况下最多可注册 10 个单元，但实际可注册数取决于单元类型 (有时注册数小于 10)。详情请参考各单元的操作手册。

参数	设定	默认值	描述	相关辅助区位和字
同步单元	支持单元同步运转功能的 CPU 单元和其它单元的名称	仅 CPU 单元	点击空白行来显示同步单元列表，并从列表中选择所需的单元。	---

10-8-8 编写同步中断任务

编写在同步中断任务中执行的程序，并在程序属性中将任务类型设定为中断任务 2(定时中断 0)。该任务包含用于执行同步控制(如数字凸轮控制)的程序。例如，对于数字凸轮控制，该任务中将包含算术处理(APR)指令或用于控制凸轮曲线(基于实轴或虚轴)的标准功能块。

● 相关辅助区位和字

名称	字 / 位	描述	读 / 写
单元同步运转服务标志	A10100.00	使用单元同步运转功能时，当 CPU 单元对同步单元执行服务时，该标志将在第二个同步运转周期到来时置 ON。而在其它时间(包括启动或重启 PLC 后的第一个同步运转周期)，该标志一直保持 OFF 状态。 此外，在用户程序使用该标志可控制与单元同步运转相关的程序执行。	读
同步输入数据刷新出错代码	A10101	当 CPU 单元无法接收来自同步单元的同步输入数据时，将 0001 Hex 存储至该字。成功接收到同步输入数据后，则将 0000 Hex 存储至该字。该字中的内容在每个同步运转周期都会刷新。 此外，可在用户程序中将该字作为使用同步输入数据的条件。	读
同步运转周期时间	A10102	该字中包含当预设周期时间变为有效时，在 PLC 设置中设定的同步运转周期时间(以 0.1ms 为增量单位)。 此外，可在用户程序(如功能块程序)中使用该字读取预设的同步运转周期时间。	读

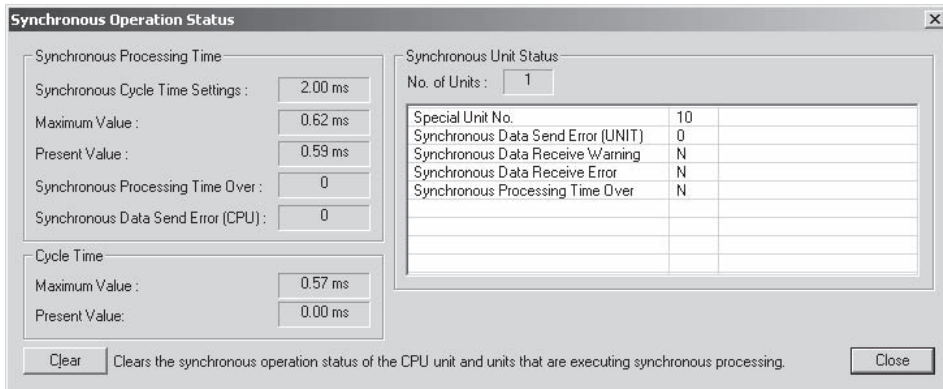
10-8-9 与单元同步运转相关的调整和故障诊断作业

检查单元同步运转状态

可通过 CX-Programmer 中的“Synchronous Operation Status”（同步运转状态）对话框或同步单元上的 SYNC 指示灯检查单元同步运转状态。

● CX-Programmer

使 CX-Programmer 与一台使用单元同步运转功能的 PLC 联机，并在 PLC 信息菜单中选择“PLC - Edit - Synchronous Operation Status”（PLC - 编辑 - 同步运转状态）。此时将显示下述“Synchronous Operation Status”（同步运转状态）对话框。

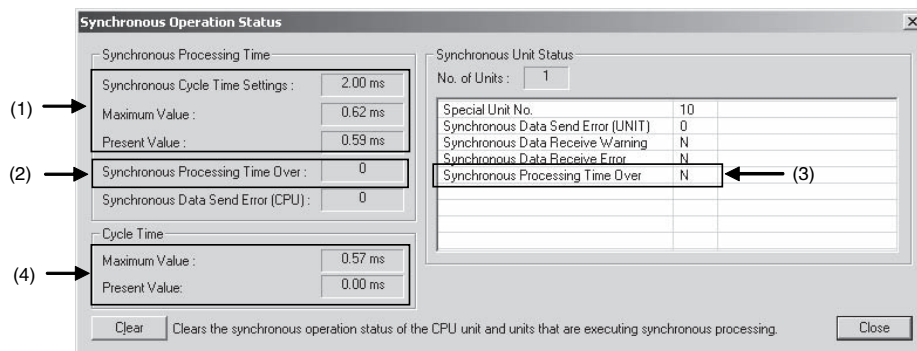


项目	描述	
同步处理时间	同步周期时间设定	即同步运转周期的设定值。
	最大值和当前值	即最大同步运转周期时间和当前同步运转周期时间。
	同步处理超时	即 CPU 单元的同步处理时间超出预设同步运转周期时间的次数。PLC 启动时会清除该数值。
	同步数据发送错误 (CPU)	即 CPU 单元将同步数据发送至同步单元时发生失败的次数。PLC 启动时会清除该数值。
周期时间	最大值和当前值	即正常周期时间的最大值和当前值。
同步单元状态	单元数	即为单元同步运转设定的单元数。
	高功能单元号	即作为同步单元使用的高功能 I/O 单元或 CPU 总线单元的单元号。
	同步数据发送错误 (UNIT)	即 CPU 单元接收同步单元发出的数据时发生失败的次数。PLC 启动时会清除该数值。
	同步数据接收报警	如果同步单元无法接收到 CPU 单元发出的同步数据 (即使只发生一次, 且包括 CPU 单元无法发送同步输出数据的情况), 该状态将变为有效。 如果 CPU 总线单元区或高功能 I/O 单元区中的单元报警复位状态位置 ON, 则会清除该状态。
	同步数据接收错误	如果同步单元连续两次或两次以上无法接收到 CPU 单元发出的同步数据, 该状态将变为有效。 如果 CPU 总线单元区或高功能 I/O 单元区中的单元错误复位状态位置 ON, 则会清除该状态。
清除按钮	同步处理超时	如果同步单元中的内部处理时间超过了同步运转周期时间 (一次或连续多次), 该状态将变为有效。 如果 CPU 总线单元区或高功能 I/O 单元区中的单元报警复位状态位置 ON, 则会清除该状态。
	该按钮用于清除单元同步运转状态及 CPU 单元和同步单元中的出错 / 报警值。	

调整同步运转周期时间

您可按照下列步骤确认是否设定了适当的同步运转周期时间，并在必要时进行调整。

- (1) 确认同步运转周期的最大值和当前值是否超出了预设值。
- (2) 确认 CPU 单元是否会频繁发生同步处理超时错误。
- (3) 确认同步单元是否会频繁发生同步处理超时报警。
- (4) 检查正常周期时间的最大值和当前值，确认正常周期时间是否会因执行了单元同步运转延长而超出允许范围。



● 检查同步单元前部的 SYNC 指示灯

当单元处于同步运转模式时，同步单元前部的 SYNC 指示灯将会亮起绿光。而在发生同步输出数据接收错误和其它错误时，该指示灯会熄灭。



附加信息

下表中的位将被分配至高功能 I/O 单元区或 CPU 总线单元区中的同步单元。在 CPU 单元的用户程序中，这些位有助于检查和控制同步单元的状态。

I/O 分类	大小	名称	功能
分配在高功能 I/O 单元区或 CPU 总线单元区中的输入位 (输入至 CPU 单元)	1 位	单元同步运转模式标志	当同步单元在单元同步运转模式下运行时，该标志将变为 ON。(该标志的状态与同步单元上的 SYNC 指示灯状态相一致。)
	1 位	同步数据接收报警标志	如果同步单元无法接收到 CPU 单元发出的同步数据 (即使只发生一次)，该标志将变为 ON。
	1 位	同步数据接收出错标志	如果同步单元连续两次或两次以上无法接收到 CPU 单元发出的同步数据，该标志将变为 ON。
	1 位	同步处理超时标志	如果同步单元中的内部处理时间超过了同步运转周期时间 (一次或连续多次)，该标志将变为 ON。
分配在高功能 I/O 单元区或 CPU 总线单元区中的输出位 (从 CPU 单元输出)	1 位	单元错误重置位	将该位置 ON 以重置同步单元中的错误 (如同步数据接收错误)。
	1 位	单元报警重置位	将该位置 ON 以重置同步单元中的报警 (如同步数据接收报警和同步处理超时报警)。

单元同步运转停止

同步信号监控错误

在正常运转时的 100ms 内或启动时 11s 内，如果 CPU 单元没有产生发送至同步单元的同步信号，所有同步单元的处理及对应的单元同步运转功能都会停止。

I/O 总线或同步单元错误

如果某个同步单元发生了 I/O 总线错误或下列任何一种单元错误，所有同步单元及对应的单元同步运转功能都会停止。

- I/O 设定错误
- 单元号重复错误
- 高功能 I/O 单元设定错误
- CPU 总线单元设定错误
- 高功能 I/O 单元错误
- CPU 总线单元错误

CX-Programmer 上将出现一个显示有错误的对话框。



附加信息

在单元同步运转过程中，即使只有一个同步单元发生重启，也将使同步运转周期停止并导致所有同步单元重启。仅在所有同步单元正常重启后，单元同步运转功能才会被恢复。

调整和故障诊断

在修改同步运转周期时间时，若存在下表所述的问题，则应进行相应的调整。

CX-Programmer 上的同步运转状态	问题	原因	纠正措施
同步处理时间区	频繁发生同步处理超时错误。	在同步运转周期内无法完成同步运转处理过程(同步数据刷新 + 同步中断任务执行)。	设定同步运转周期时间，使其大于同步运转处理时间(同步数据刷新时间+同步中断任务执行时间)。
	发生同步数据发送错误(CPU)。	CPU 单元内部发生了硬件错误。	更换 CPU 单元。
同步单元状态区	频繁发生同步数据发送错误(UNIT)。	在同步输入数据刷新期到来前，没有产生从同步单元发送到 CPU 单元的同步输入数据。	设定同步运转周期时间，使其大于同步单元的内部处理时间。
	发生同步处理超时报警。	在同步运转周期结束前，无法完成同步单元的内部处理过程。	设定同步运转周期时间，使其大于同步单元的内部处理时间。

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 自动联机	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	11-27

11-1 从 CX-Programmer 访问 PLC

11-1-1 概述

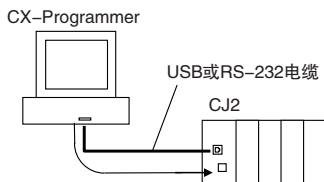
本节对如何从 CX-Programmer 访问 PLC 进行了概述。

连接到 PLC

可通过下述任一方法实现从 CX-Programmer 访问 PLC。

● 直接连接到 PLC

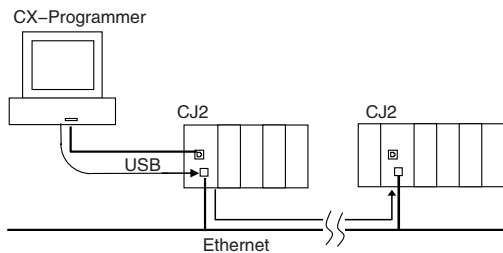
通过串行端口，可与直接连接到 CX-Programmer 的 PLC 联机。



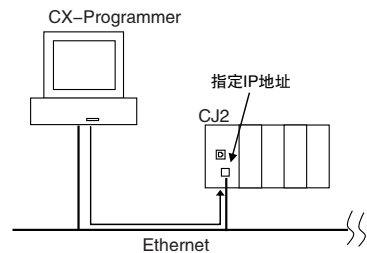
● 在网络上访问 PLC

可通过下述任一方法实现在网络上访问 PLC。

通过直接连接到 CX-Programmer 的 PLC 的串行端口与网络中的其它 PLC 联机。



通过直接连接到网络的个人计算机与网络中的 PLC 联机。



与 PLC 联机的步骤

通过以下两种步骤之一可实现 PLC 与 CX-Programmer 的联机。

- “Change PLC” (变更 PLC) 对话框

为 CX-Programmer 项目设定访问目标 PLC 的方式。在创建项目时可进行此项设定，而在创建项目后也可对设定进行修改。

- 自动联机

自动联机功能用于访问已经建立连接的 PLC。该功能无需创建 CX-Programmer 项目即可使用。

直接连接

直接连接方式通过串行端口实现与直接连接至 CX-Programmer 的 PLC 的联机。

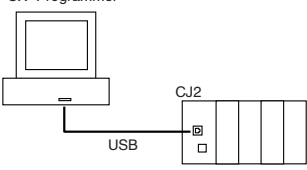
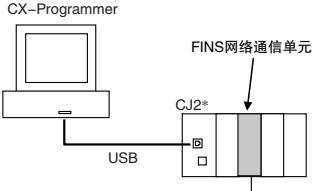
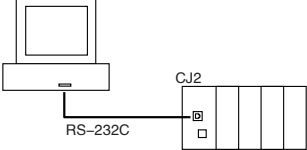
EtherNet/IP 结点连接

EtherNet/IP 结点连接方式通过 EtherNet/IP 访问已经建立连接的 PLC。即使不知道目标 PLC 的 IP 地址，也可通过搜索的方式访问 PLC。

11-1-2 可访问 PLC 的系统配置

与 PLC 的直接串行连接

请选择下表所示的网络类型。

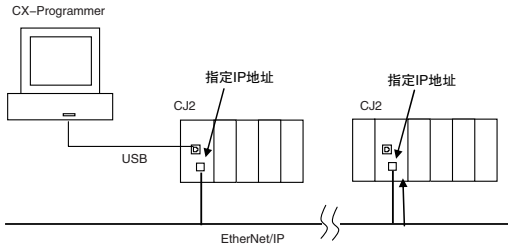
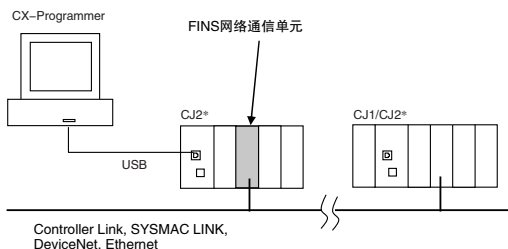
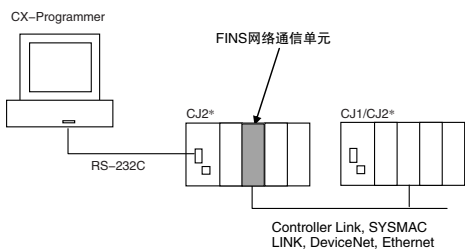
连接电缆	系统配置	“Change PLC” (变更 PLC) 对话框 (网络类型)	自动联机	
			直接连接	EtherNet/IP 连接
USB 电缆	 <p>CX-Programmer</p> <p>USB</p> <p>CJ2</p>	USB	可访问	不可访问
访问 FINS 网络	 <p>CX-Programmer</p> <p>USB</p> <p>CJ2*</p> <p>FINS网络通信单元</p>	Toolbus(USB 端口)	可访问	不可访问
RS-232C 电缆	 <p>CX-Programmer</p> <p>RS-232C</p> <p>CJ2</p>	Toolbus	可访问	不可访问

* 必须设定本地网络表。

在网络上访问 PLC

● 访问通过串行端口连接的 PLC

请选择下表所示的网络类型。

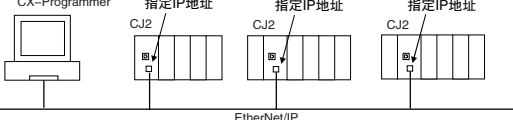
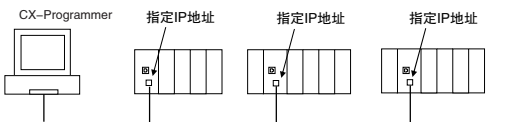
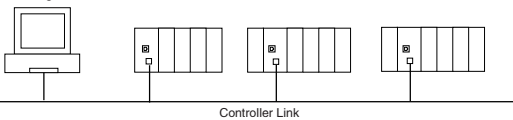
连接电缆	系统配置	“Change PLC” (变更 PLC) 对话框 (网络类型)	自动联机	
			直接连接	EtherNet/IP 连接
USB 电缆	通过 USB 访问 EtherNet/IP 网络 (*1) 	USB	不可访问	可访问
	通过 USB 访问 FINS 网络 (*2)  <p>Controller Link, SYSMAC LINK, DeviceNet, Ethernet</p> <p>* 必须设定本地网络表。</p>	Toolbus(USB 端口)	不可访问	不可访问
RS-232C 电缆	通过 RS-232C 访问 FINS 网络 (*2)  <p>Controller Link, SYSMAC LINK, DeviceNet, Ethernet</p> <p>* 必须设定本地网络表。</p>	Toolbus	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。此时跨网络层连接是无法实现的。如果通过自动联机功能进行 EtherNet/IP 连接，则可进行跨网络层连接。

*2 FINS 网络间可实现跨网络层连接。

● 通过网络连接

请选择下表所示的网络类型。

连接电缆	系统配置	“Change PLC” (变更 PLC) 对话框 (网络类型)	自动联机	
			直接连接	EtherNet/IP 连接
EtherNet/IP	访问 EtherNet/IP 网络 (*1)  必须设定连接目标处的 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、结点地址和 IP 地址。	Ethernet、Ethernet (FINS/TCP)	不可访问	不可访问
Controller Link	 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。此时跨网络层连接是无法实现的。如果通过自动联机功能进行 EtherNet/IP 连接，则可进行跨网络层连接。

*2 FINS 网络间可实现跨网络层连接。



正确使用注意事项

通过 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 的内置 EtherNet/IP 端口与支持软件进行连接时，若发生以下任一错误，则将无法进行连接。

I/O 总线错误 (CJ2M-CPU3□ 可进行连接)

I/O 点数过多错误

单元号重复错误

机架号重复错误

I/O 设定错误

如果发生上述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 系统采用了更严密的防火墙安全机制，其中增加了对 Ethernet 端口数据通信的限制。通过计算机上的 EtherNet 端口对下列任意一种 PLC 进行 EtherNet/IP 连接 *1 时，必须修改 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：

- 当网络类型设定为“EtherNet/IP”时进行联机
- 在 PLC 菜单中选择“Auto Online – EtherNet/IP Node Online”（自动联机 – EtherNet/IP 结点联机），与 EtherNet/IP 网络上的 PLC 自动联机。

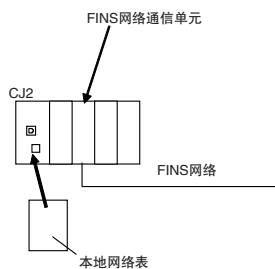
有关操作步骤，请参考“A-6 在 Windows XP(SP2 或以上)、Windows Vista 或 Windows 7 中进行 EtherNet/IP 连接”。



附加信息

路由表的创建和传送必须在下列情况下进行。

如果将一个或多个 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，以确保能通过支持软件访问内置 EtherNet/IP 网络（在 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□上）和 FINS 网络。



* “FINS 网络通信单元”表示 Controller Link 单元、SYSMAC LINK 单元、Ethernet 单元、DeviceNet 单元或 FL-net 单元，但不包含 CompoNet 单元。此外，它还表示在串行通信中使用了路由表和串行网关。



附加信息

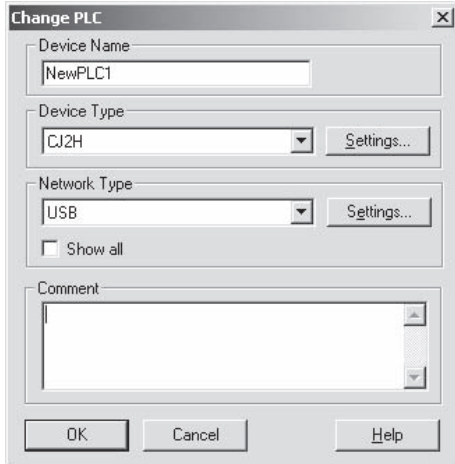
如果一个或多个 FINS 网络通信单元连接到 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□，则应创建和传送本地网络表。如果试图在网络地址设为 0（默认值）的情况下通过支持软件进行访问，而不创建和传送本地网络表，您可在下列情况下访问内置 EtherNet/IP 端口的结点地址：

内置 EtherNet/IP 端口的单元号（默认值：0）小于 FINS 通信单元的单元号。

11-1-3 从 CX-Programmer 访问 PLC

“Change PLC” (变更 PLC) 对话框的设定步骤

创建新项目时，请按照下列步骤在“Change PLC” (变更 PLC) 对话框中选择连接 PLC 的方式。以下为 CJ2H CPU 单元示例。



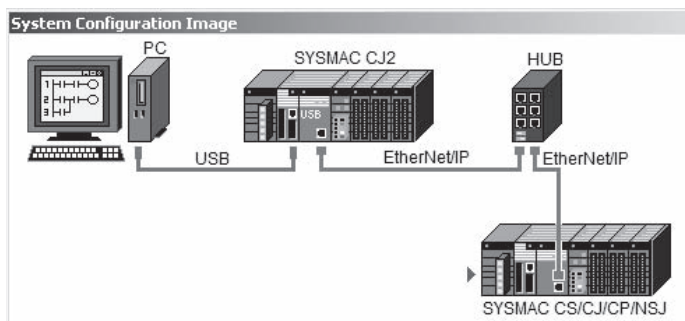
1. 在“Device Type” (设备类型) 列表中选择“CJ2H”。
2. 在“Network Type” (网络类型) 列表中选择连接到 PLC 的方式。可选择以下方式之一作为默认设定。
 - USB
 - EtherNet/IP

必要时，可点击“Show All” (显示所有) 按钮显示其它网络类型。请根据下表所示的两个条件选择网络类型。

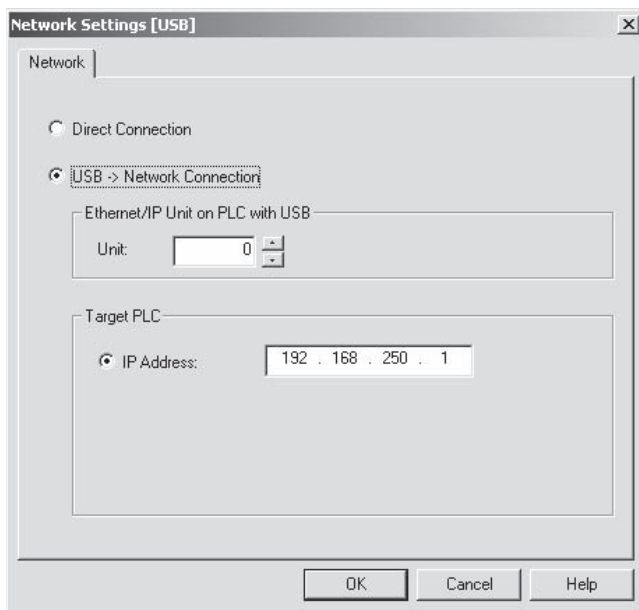
条件 1: 连接到 PLC 的电缆类型	条件 2	网络类型	
USB	---	USB	默认显示 通过点击“Show All” (显示所有) 按钮显示。
Ethernet	访问 EtherNet/IP 网络	EtherNet/IP	
USB	通过 USB 访问 FINS 网络	Toolbus(USB 端口)	
Ethernet	将 Ethernet 网络作为 FINS 网络访问	Ethernet	
		Ethernet(FINS/TCP)	
RS-232C	---	Toolbus	
		SYSMAC WAY	
Controller Link (连有 Controller Link 选件板)	---	Controller Link	
FINS 网关	---	FINS 网关	

3. 点击“Settings”(设置)按钮。将显示所选网络类型的“Network Settings”(网络设定)对话框。本例对默认显示的网络类型(即 USB 和 EtherNet/IP)进行了说明。有关其它网络类型的设定详情,请参考《CX-Programmer 操作手册》(样本编号:W446)。

● 网络类型: USB



在“Network”(网络)选项页中选择“USB -> Network Connection”(USB -> 网络连接)。



“Direct Connection”(直接连接):

选择该选项来访问通过 USB 直接连接的 PLC。

“USB->Network Connection”(USB -> 网络连接):

选择该选项通过 USB 端口访问 EtherNet/IP 网络(单个网络)中的 PLC。请进行以下设定。

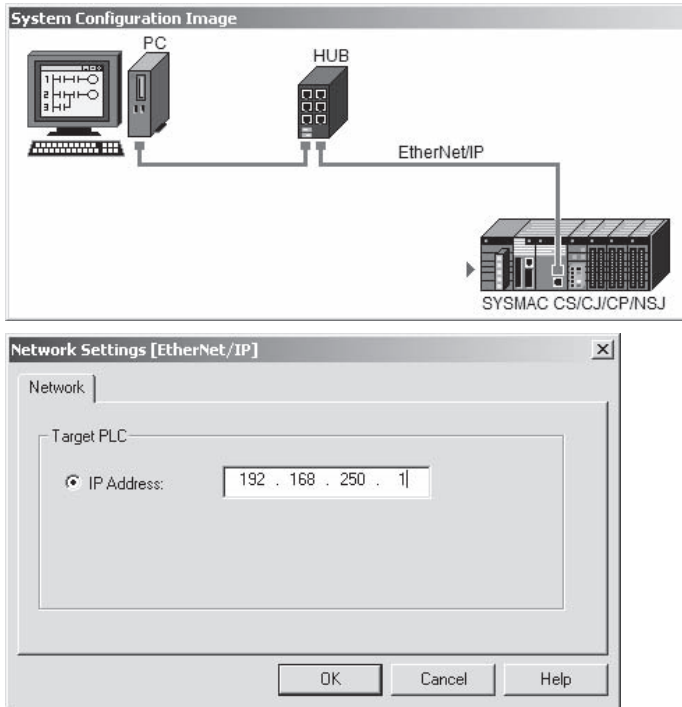
通过 USB 连接 PLC 上的 EtherNet/IP 单元:

连接 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 上的内置 EtherNet/IP 端口时, 将单元区域设定为“0”。

目标 PLC 的 IP 地址:

设定 CJ2H-CPU6□-EIP、CJ2M-CPU3□ 或 EtherNet/IP 单元上的内置 EtherNet/IP 端口的 IP 地址。CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 的内置 EtherNet/IP 端口的默认 IP 地址为: 192.168.250. 结点地址。

● 网络类型：EtherNet/IP



目标 PLC 的 IP 地址：

设定 CJ2H-CPU6□-EIP、CJ2M-CPU3□ 或 EtherNet/IP 单元上的内置 EtherNet/IP 端口的 IP 地址。
 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 的内置 EtherNet/IP 端口的默认 IP 地址为：192.168.250. 结点地址。

● 设定个人计算机的 IP 地址

通过 EtherNet/IP 连接到 PLC 时，所设定的个人计算机 IP 地址必须与 CJ2H-CPU6□-EIP、CJ2M-CPU3□ 或 EtherNet/IP 单元上的内置 EtherNet/IP 端口的 IP 地址保持一致。例如，如果 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ 上的内置 EtherNet/IP 端口的 IP 地址为默认设定，则应对个人计算机设定以下的 IP 地址。

• IP 地址

例：192.168.250.2
 自定义设定值(1 ~ 254)
 与内置EtherNet/IP端口IP地址同段的值

• 子网掩码

例：255.255.255.0.200
 C类掩码
 自定义设定值(0 ~ 252)

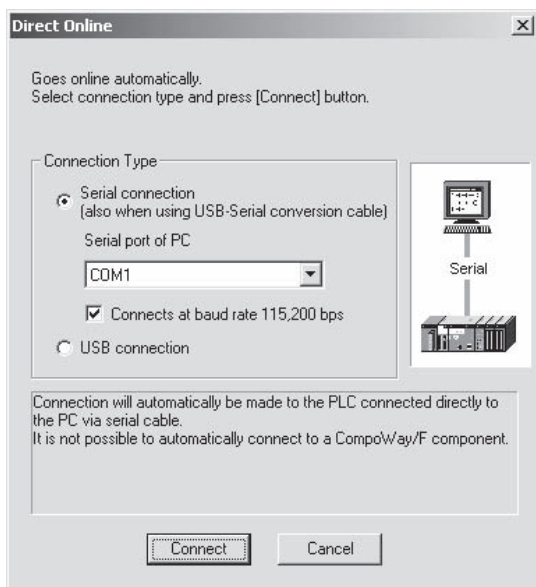
11-1-4 自动联机

如果个人计算机上没有目标 PLC 的项目，可进行联机操作并从 PLC 上传程序。

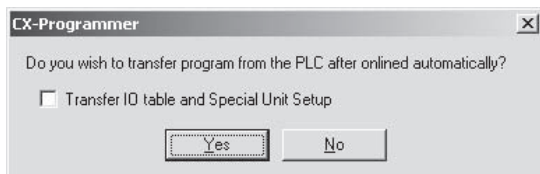
直接串行连接

执行自动联机时，将自动搜索可用的个人计算机串行端口。发现适用的串行端口时，CX-Programmer 将自动与连接到串行端口上的 PLC 联机。

1. 在 PLC 菜单中选择 **“Auto Online – Direct Online”** (自动联机 – 直接联机)，将显示如下所示的 **“Direct Online”** (直接联机) 对话框。
2. 选择以下任一方法将个人计算机连接到 PLC。
 - 串行连接 (包括将个人计算机处的 USB 连接转换为 PLC 的 RS-232C 连接)
 - USB 连接



从个人计算机的 USB 端口连接到 CJ2 CPU 单元的外设 USB 端口时，请选择 **“USB connection”** (USB 连接)，然后点击 **“Connect”** (连接) 按钮。将显示以下对话框。

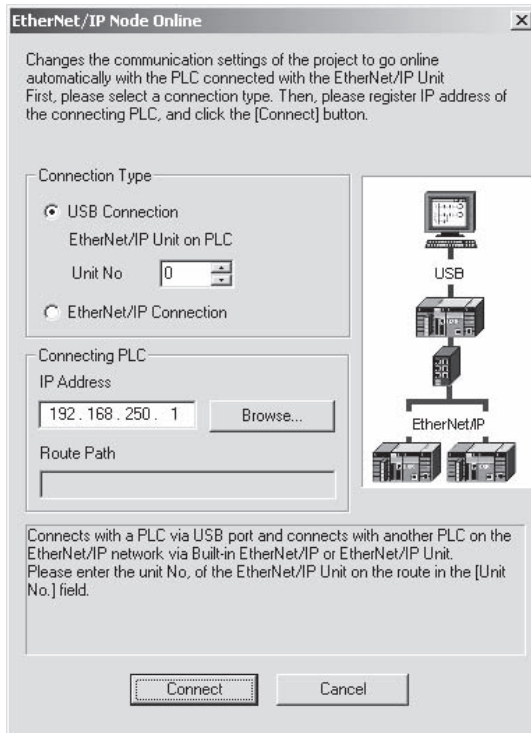


进行自动联机后，若要将程序、PLC 设置和 I/O 表上传至个人计算机，请点击 **“Yes”** (是) 按钮。CX-Programmer 将自动连接到 PLC，并从 PLC 上传这些项目。

EtherNet/IP 结点连接

通过 EtherNet/IP 网络，即可实现与连接到 EtherNet/IP 网络的 PLC 的连接。即使不知道目标 PLC 的 IP 地址，也可通过搜索的方式访问 PLC。

1. 在 PLC 菜单中选择“Auto online”(自动联机)–“EtherNet/IP Node Online”(EtherNet/IP 结点联机)。
2. 将显示“EtherNet/IP Node Online”(EtherNet/IP 结点联机)对话框。



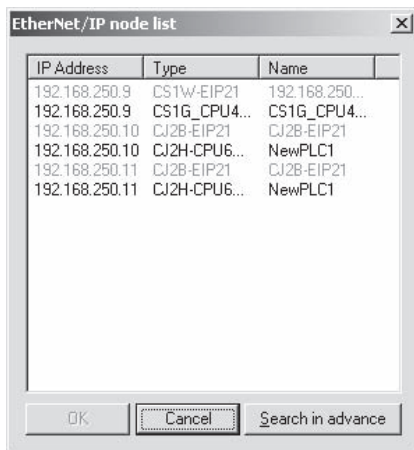
3. 根据连接类型选择连接到目标 PLC 的方式。
 - USB 连接
若选择 USB 连接方式，则将通过 USB 端口直接连接到 PLC，然后通过内置 EtherNet/IP 端口或 EtherNet/IP 单元与 EtherNet/IP 网络 (单个网络) 上的 PLC 联机。
 - EtherNet/IP 连接
若选择 EtherNet/IP 连接方式，则将计算机连接到 EtherNet/IP 网络，并通过 EtherNet/IP 与网络上 (单个网络) 的 PLC 联机。

4. 根据所连接的 PLC 选择目标 PLC 的 IP 地址的设定方式。

- IP 地址
直接输入目标 PLC 的 IP 地址。
- 浏览
搜索连接到的网络的联机 PLC，并选择目标 PLC。在不知道目标 PLC 的 IP 地址的情况下，该方法尤其适用。

连接的结点可作为列表或通过网络层显示。

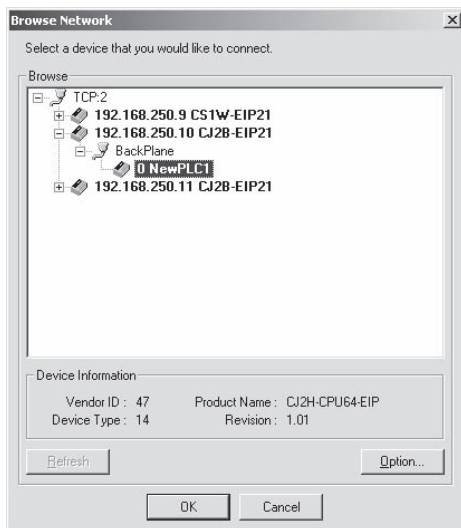
结点列表 (浏览显示)



- 列表中将显示 IP 地址、设备类型和名称。选择 IP 地址并点击“OK”（确定）按钮。

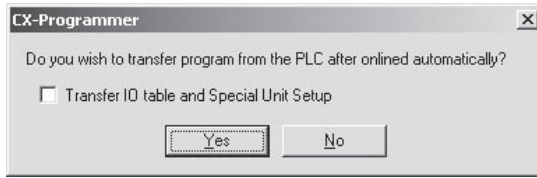
预搜索

在“EtherNet/IP node list” (EtherNet/IP 结点列表) 对话框中点击“Search in advance” (预搜索) 按钮，以显示下列窗口。



- 点击加号 (+) 图标，展开项目下的所有子项。然后选择需要连接的 PLC 并点击“OK”（确定）按钮。

5. 点击“Connect”（连接）按钮进行联机。此时将显示以下对话框。



进行自动联机后，若要将程序、PLC 设置和 I/O 表上传至个人计算机，请点击“**Yes**”（是）按钮。个人计算机将自动连接到 PLC，并从 PLC 上传这些项目。

11-2 串行通信

11-2-1 串行通信概述

串行通信端口模式 (协议) 可在 CPU 单元的 PLC 设置中进行切换, 并根据所选的协议配置下列系统。

协议

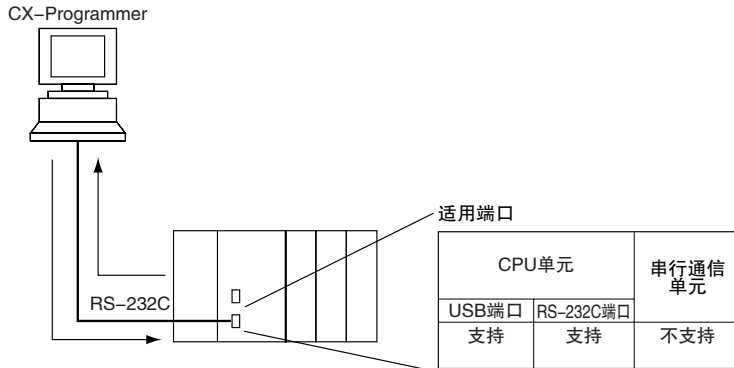
下列协议支持串行通信。

协议 (串行通信模式)	主连接	用途	适用的命令和通信指令
外设总线 (Toolbus) (仅限 CPU 单元上的 RS-232C 端口。 不支持串行通信单元。)	CX-Programmer	从计算机侧实现编程设备和 PLC 之间的通信。	无
上位链接 (SYSMAC WAY) 从站	个人计算机 OMRON 可编程终端	实现上位计算机和 PLC 之间的通信, 并可由 PLC 将命令发送到个人计算机。	上位链接命令 /FINS 命令 (可由 PLC 将命令发送到个人计算机)
无协议 *1	通用外部设备	实现通用设备之间的无协议通信。	TXD(236) 指令、RXD(235) 指令 TXDU(256) 指令、RXDU(255) 指令 DTXDU(262) 指令 DRXDU(261) 指令
NT 链接 (1:N)	OMRON 可编程终端	通过直接访问方式与可编程终端进行高速通信。	无
串行网关	OMRON 部件 PLC	PLC 将接收到的 FINS 命令转换为 CompoWay/F、Modbus 或上位链接协议, 然后将转换后的命令传送到串行线路。	---
CompoWay/F 主站 *1	CompoWay/F 从站	将串行端口接收到的 FINS 命令 (压缩的 CompoWay/F 命令) 转换为 CompoWay/F 命令。	FINS 命令 接收到 2803 Hex (包括使用 CMND(490) 发送 FINS 命令)
Modbus 主站 *1 (仅限串行通信单元。不支持 CPU 单元上的 RS-232C 端口。)	Modbus 从站	将串行端口接收到的 FINS 命令 (压缩的 Modbus 命令) 转换为 Modbus 命令。	FINS 命令 接收到 2804 Hex 或 2805 Hex (包括使用 CMND(490) 发送 FINS 命令)
上位链接 FINS(SYSWAY) 主站 *1 (仅限串行通信单元。不支持 CPU 单元上的 RS-232C 端口。)	上位链接 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接收到的 FINS 命令 (包括使用 CMND(490) 发送 FINS 命令)
协议宏 (仅限串行通信单元。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 用于通过设定各种参数创建协议。)	PMCR(260) 指令
串行 PLC 链接 (仅限 CJ2M CPU 单元)	OMRON PLC(CJ2M、CJ1M、CP1H、CP1L 和 CP1E CPU 单元)	可在 9 个 CPU 单元 (包括 1 个主站单元和 8 个从站单元) 之间实现多达 10 字 / 单元的共享容量。	无

*1 仅限 1.2 版或更高版本的串行通信单元。

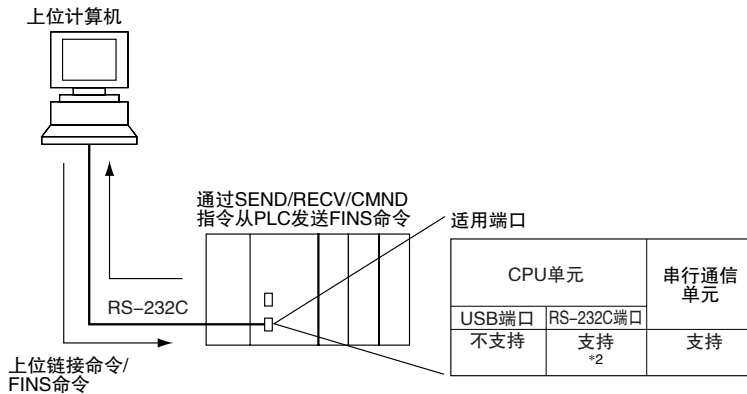
外设总线 (Toolbus)

CX-One 支持软件 (如 CX-Programmer)



上位链接系统从站 (SYSMAC WAY 模式 1:N)

上位链接系统允许读 / 写 PLC 的 I/O 存储器, 通过执行上位链接命令或 FINS 命令从上位计算机 (个人计算机 *1 或可编程终端) 修改操作模式, FINS 命令的开始和结束位置分别带有一个帧头和一个帧尾。或者, FINS 命令 (开始和结束位置分别带有一个帧头和一个帧尾) 可通过执行网络通信指令 (SEND(090)/RCV(098)/CMND(490)) 从 PLC 发送到通过上位链接 (Host Link) 系统连接的个人计算机。



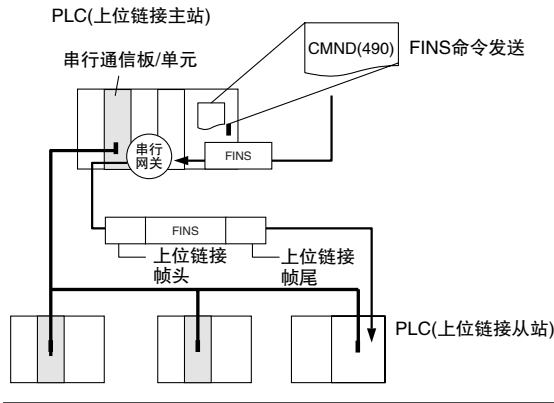
*1 CX-One 支持软件除外。

*2 将 CPU 单元前面板上的 DIP 开关位 5 设定为 OFF, 并将 PLC 设置中的串行通信模式设定为上位链接。



附加信息

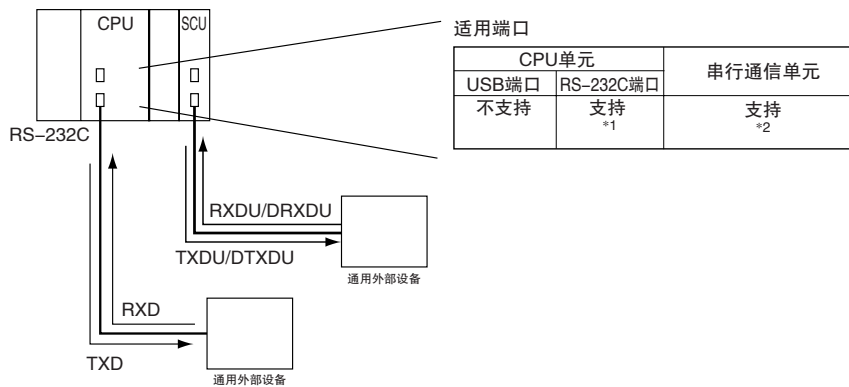
上位链接主站功能可在使用 1.2 版或更高版本的串行通信单元时，通过串行网关发送 CMND(490) 指令来执行。



无协议通信系统

无协议通信可实现简单的数据传送（如用通信端口 I/O 指令输入条形码数据和输出打印机数据），设定起始码和完成码及执行 RS 和 CS 信号控制。

下图根据使用的通信端口和数据传送（发送或接收）的方向，对每个通信端口 I/O 指令的用途进行了说明。



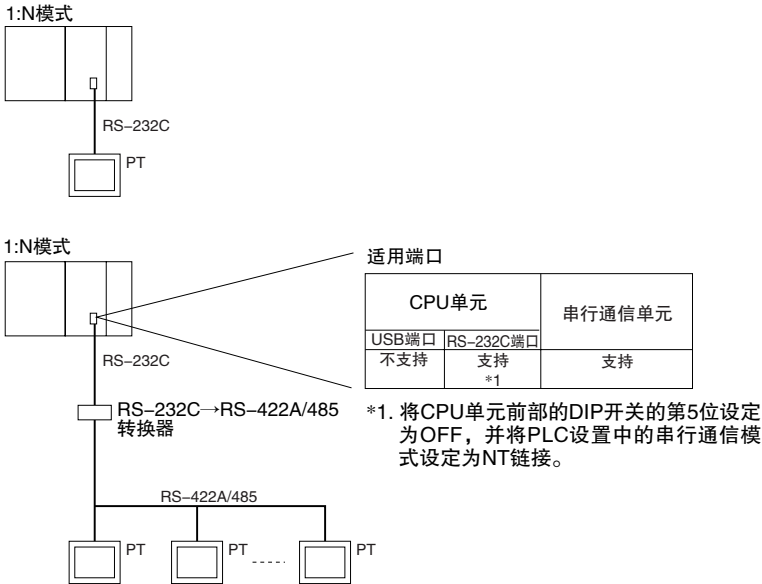
*1 将 CPU 单元前面板上的 DIP 开关位 5 设定为 OFF，并将 PLC 设置中的串行通信模式设定为无协议通信。

*2 仅 1.2 版或更高版本的串行通信单元支持无协议通信。

NT 链接系统 (1:N 模式)

如果通过 RS-232C 端口连接 PLC 和可编程终端 (PT) 等对象, 可在 PLC 的 I/O 存储器中分配 PT 的状态控制区、状态通知区以及例如触摸开关、指示灯和存储器映射。NT 链接系统允许由 PLC 控制 PT, PT 可定期从 PLC 的状态控制区读取数据, 并在该区域发生任何变化时执行必要的操作。PT 将数据写入 PLC 的状态通知区, 从而实现与 PLC 的通信。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 的比率是 1:N, 其中 N 大于等于 1。

对 1:N NT 链接中的 PT 通信方式进行设定。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。但不可将 PLC 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, 则无法使用 NT20S、NT600S、NT30、NT30C、NT620S、NT620C 和 NT625C(即使仅连接其中的一个 PT)。
- 将多个 PT 连接到同一个 PLC 时, 应确保每一个 PT 均被分配了唯一的单元号。如果多个 PT 使用相同的单元号, 将会出现误动作。



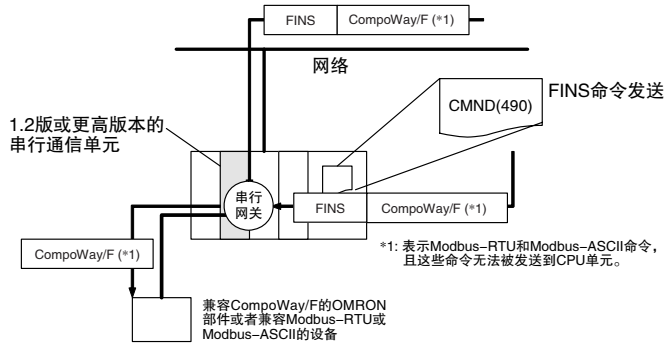
附加信息

1:1 和 1:N NT 链接协议互不兼容 (即属于独立的串行通信协议)。

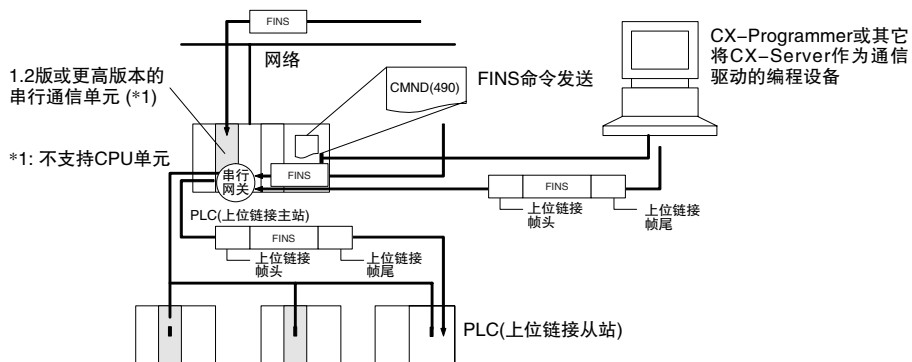
串行网关模式

接收到的 FINS 报文会根据其内容自动转换为 CompoWay/F。当使用 1.2 版或更高版本的串行通信单元时，接收到的 FINS 报文会根据其内容自动转换为 CompoWay/F、Modbus-RTU、Modbus-ASCII 或上位链接 FINS 中的一种协议格式。

● CompoWay/F、Modbus-RTU、Modbus-ASCII



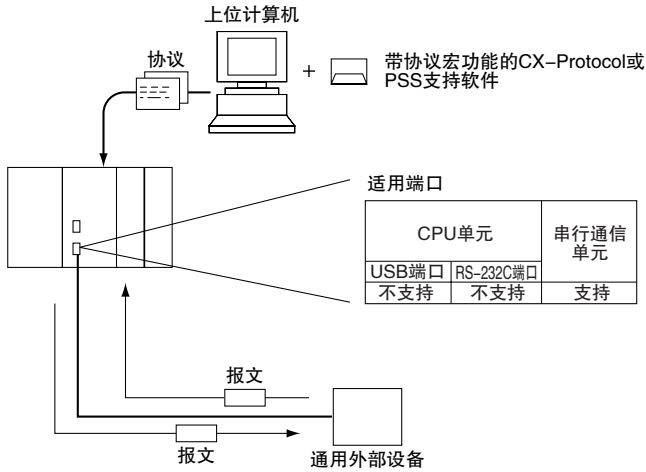
● 上位链接 FINS



协议宏 (仅限串行通信单元)

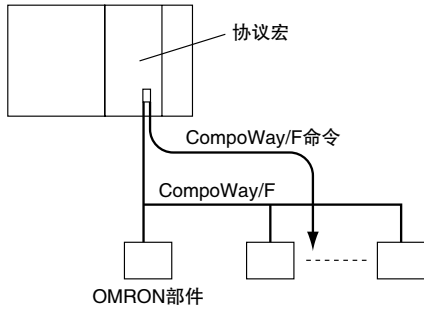
CX-Protocol 用于创建根据通用外部设备的通信方式 (半双工或全双工、异步) 进行的数据传送步骤 (协议)，所创建的协议记录在串行通信单元中，只需在 CPU 单元中执行 PMCR(260) 指令就能从外部设备发送和接收数据。OMRON 设备 (如温控器、智能信号处理器、条形码阅读器和调制解调器) 的数据通信协议可作为标准协议来支持。(*1) 用户也可根据应用需要来改变这些协议。

*1 CX-Protocol 和串行通信单元提供标准协议支持。



注 CompoWay/F(主站功能)

CJ 系列 CPU 单元可作为主机向连接在系统中的 OMRON 部件发送 CompoWay/F 命令，并按照协议宏功能标准协议中 CompoWay/F 的发送 / 接收顺序来执行 CompoWay/F 命令。



串行 PLC 链接 (仅限 CJ2M CPU 单元)

仅 CJ2M CPU 单元支持串行 PLC 链接。串行 PLC 链接允许在 CJ2M CPU 单元之间或 CJ2M CPU 单元和 CJ1M/CP1H/CP1L/CP1E CPU 单元之间交换数据, 而无需特殊编程。其间需使用串行 PLC 链接区 (CIO 3100 ~ CIO 3199), 并通过 RS-232C 或 RS-422A/485 连接 CPU 单元。CJ1W-CIF11 RS-422A 适配器用于将 CJ2M-CPU1□上的 RS-232C 串行端口转换为 RS-422A/485 端口。

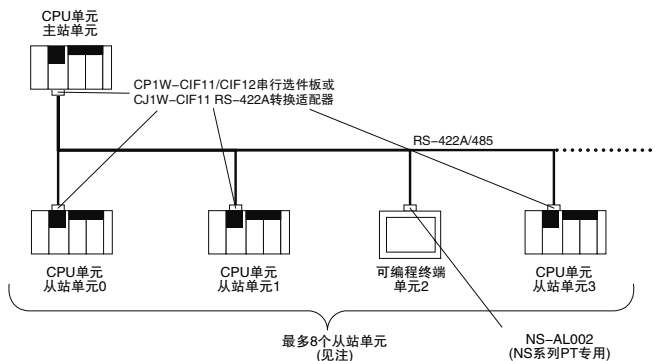
而设定了 1:N NT 链接协议的 PT 也可在网络中使用。设定为从站的 PT 经由网络与通过 1:N NT 链接协议设定为主站的 CPU 单元进行通信。但是当连接至 PT 时, 串行 PLC 链接区中的字内容 (针对 PT) 将变得不稳定。

● 运行规格

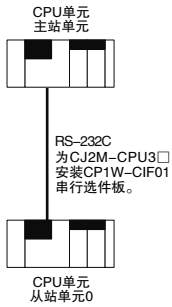
参数	设定
适用 PLC	CJ2M、CJ1M、CP1H、CP1L 和 CP1E
波特率	115,200 或 38,400bps
适用串行端口	CPU 单元上的串行端口 (CJ2M-CPU3□ 需使用串行选件板)
连接方式	RS-232C 或 RS-422A/485
分配字	串行 PLC 链接区字: CIO 3100 ~ CIO 3199(每个 CPU 单元最多分配 10 个字) 注: CP1E CPU 单元为串行 PLC 链接区分配了 CIO 200 ~ CIO 289。
最大单元数	最多 9 个单元, 其中包括 1 个主站单元和 8 个从站单元。 如果将同一条线路上的 PT 设定为 1:N NT 链接协议, 则构成单元 (最多 9 个) 中必须包含从站 PT 和 CPU 单元。
链接方式 (数据刷新方式)	全站链接方式或主站链接方式

● 系统配置

1:N 连接 CJ2M、CP1L、CP1H、CP1E 或 CJ1M CPU 单元 (最多 8 个结点)



1:1 连接 CJ2M、CP1L、CP1H、CP1E 或 CJ1M CPU 单元



注: 如果将同一条线路上的PT设定为串行PLC链接协议, 则构成单元(最多8个)中必须包含从站PT和CPU单元。

**正确使用注意事项**

CJ1M-CIF11 和 CP1W-CIF11 未采用隔离设计, 因此其最大传送距离仅为 50m。如果传送距离超过 50m, 则应使用 NT-AL001(隔离型), 且不得在传送路径上的任何位置使用 CJ1W-CIF11 或 CP1W-CIF11。如果仅使用 NT-AL001, 最大总传送距离可达到 500m。

● **步骤**

串行 PLC 链接的工作方式取决于 PLC 设置中的主站单元和从站单元设定 (如下所示)。

主站单元设定

1. 将 RS-232C 端口的串行通信模式设定为串行 PLC 链接主站单元。
2. 将链接方式设定为 “All” (所有) 或 “Polling Unit” (主站单元)。
3. 设定需链接的字数 (1 ~ 10)。
4. 设定串行 PLC 链接可使用的最大单元数 (0 ~ 7)。

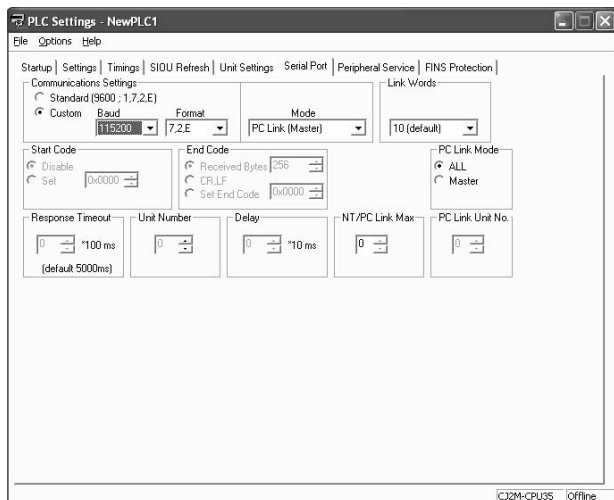
从站单元设定

1. 将 RS-232C 端口的串行通信模式设定为串行 PLC 链接从站单元。
2. 设定串行 PLC 链接可使用的最大从站单元数。

● PLC 设置

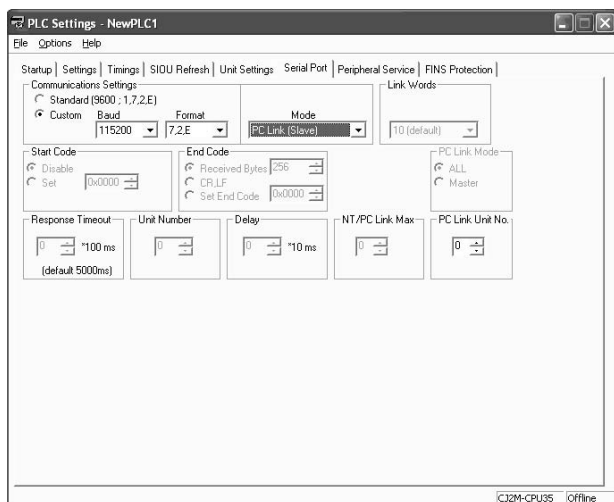
主站单元设定

“Serial Port” (串行端口) 选项页



参数	设定
通信设定	确保通信设定与所连接 PLC 的设定相符。 若要将所连接 PLC 设定为 115,200bps, 则点击 “Custom” (自定义) 选项并将波特率设定为 115,200, 而与参数设定中的选择无关。
模式	选择 PC 链接主站单元模式。
链接字数	该参数仅可在主站单元中进行设定, 默认值为 10 个字。
PC Link 方式	选择 “All” (所有) 或 “Polling Unit” (主站单元)。
NT/ 串行 PLC 链接的最大单元号 (NT/PC 最大链接数)	设定所连接从站的最大单元号。

从站单元中的设定



参数	设定
通信设定	确保通信设定与所连接 PLC 的设定相符。 若要将所连接 PLC 设定为 115,200bps, 则点击 “Custom” (自定义) 选项并将波特率设定为 115,200, 而与参数设定中的选择无关。
模式	选择 PC 链接从站单元。
串行 PLC 链接单元号	设定单元号 (0 ~ 7)。

● 链接方式 (数据刷新方式)

可选择以下两种数据刷新方式。

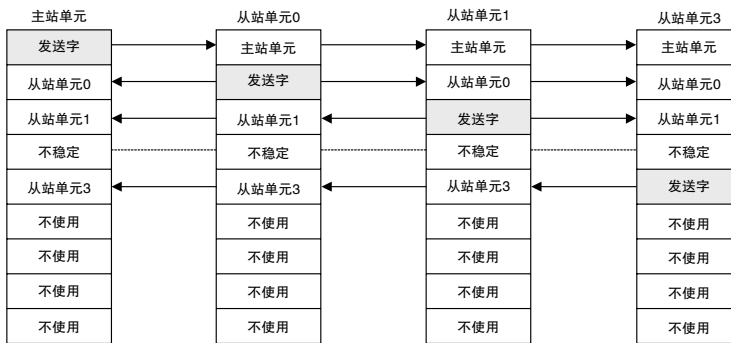
- 全站链接方式
- 主站单元链接方式

全站链接方式

主站和从站单元均可更新串行 PLC 链接中所有结点的数据。唯一例外的是未连接在网络中的从站数据和任何连接的 PT 的数据。这些数据区在所有结点中均不稳定。

示例：全站链接方式，最大单元号：3

下图中，由于网络或 PT 中不存在从站单元 2，因此分配给从站单元 2 的字在所有结点中均不稳定。

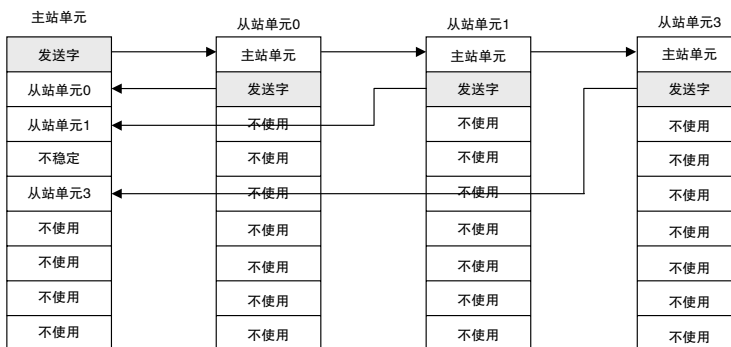


主站单元链接方式

串行 PLC 链接中仅主站单元可更新所有从站单元的数据，而从站单元仅更新主站单元的数据。主站单元链接方式的优势在于各从站单元中分配用于自身数据的地址均相同，因而可通过共用的梯形图程序进行数据访问。在所有结点中，未连接在网络中的从站单元及任何连接的 PT 的数据是不稳定的。

示例：主站单元链接方式，最大单元号：3

下图中，由于网络或 PT 中不存在从站单元 2，因此该数据在主站单元中是不稳定的。



● 分配字

全站链接方式

地址	链接字数	1 个字	2 个字	3 个字	~	10 个字
CIO 3100	主站单元	CIO 3100	CIO 3100 ~ CIO 3101	CIO 3100 ~ CIO 3102		CIO 3100 ~ CIO 3109
	从站单元 0	CIO 3101	CIO 3102 ~ CIO 3103	CIO 3103 ~ CIO 3105		CIO 3110 ~ CIO 3119
	从站单元 1	CIO 3102	CIO 3104 ~ CIO 3105	CIO 3106 ~ CIO 3108		CIO 3120 ~ CIO 3129
	从站单元 2	CIO 3103	CIO 3106 ~ CIO 3107	CIO 3109 ~ CIO 3111		CIO 3130 ~ CIO 3139
	从站单元 3	CIO 3104	CIO 3108 ~ CIO 3109	CIO 3112 ~ CIO 3114		CIO 3140 ~ CIO 3149
	从站单元 4	CIO 3105	CIO 3110 ~ CIO 3111	CIO 3115 ~ CIO 3117		CIO 3150 ~ CIO 3159
	从站单元 5	CIO 3106	CIO 3112 ~ CIO 3113	CIO 3118 ~ CIO 3120		CIO 3160 ~ CIO 3169
	从站单元 6	CIO 3107	CIO 3114 ~ CIO 3115	CIO 3121 ~ CIO 3123		CIO 3170 ~ CIO 3179
	从站单元 7	CIO 3108	CIO 3116 ~ CIO 3117	CIO 3124 ~ CIO 3126		CIO 3180 ~ CIO 3189
CIO 3199	不使用	CIO 3109 ~ CIO 3199	CIO 3118 ~ CIO 3199	CIO 3127 ~ CIO 3129		CIO 3190 ~ CIO 3199

主站单元链接方式

地址	链接字数	1 个字	2 个字	3 个字	~	10 个字
CIO 3100	主站单元	CIO 3100	CIO 3100 ~ CIO 3101	CIO 3100 ~ CIO 3102		CIO 3100 ~ CIO 3109
	从站单元 0	CIO 3101	CIO 3102 ~ CIO 3103	CIO 3103 ~ CIO 3105		CIO 3110 ~ CIO 3119
	从站单元 1	CIO 3102	CIO 3104 ~ CIO 3105	CIO 3106 ~ CIO 3108		CIO 3120 ~ CIO 3129
	从站单元 2	CIO 3103	CIO 3106 ~ CIO 3107	CIO 3109 ~ CIO 3111		CIO 3130 ~ CIO 3139
	从站单元 3	CIO 3104	CIO 3108 ~ CIO 3109	CIO 3112 ~ CIO 3114		CIO 3140 ~ CIO 3149
	从站单元 4	CIO 3105	CIO 3110 ~ CIO 3111	CIO 3115 ~ CIO 3117		CIO 3150 ~ CIO 3159
	从站单元 5	CIO 3106	CIO 3112 ~ CIO 3113	CIO 3118 ~ CIO 3120		CIO 3160 ~ CIO 3169
	从站单元 6	CIO 3107	CIO 3114 ~ CIO 3115	CIO 3121 ~ CIO 3123		CIO 3170 ~ CIO 3179
	从站单元 7	CIO 3108	CIO 3116 ~ CIO 3117	CIO 3124 ~ CIO 3126		CIO 3180 ~ CIO 3189
CIO 3199	不使用	CIO 3109 ~ CIO 3199	CIO 3118 ~ CIO 3199	CIO 3127 ~ CIO 3129		CIO 3190 ~ CIO 3199

● 辅助区中的相关位和字

名称	地址	功能	读 / 写	刷新时间
串行端口 PT 通信标志 *	A393.00 ~ A393.07	当串行端口在 NT 链接模式或串行 PLC 链接模式下通信时, 相应位将置 ON。 位 0 ~ 7 对应单元 0 ~ 7。 ON: 通信执行中 OFF: 通信未执行中	读	<ul style="list-style-type: none"> 接通电源时清除。 当串行端口在 NT 链接或串行 PLC 链接模式下时更新。 位 0 ~ 7 对应单元 0 ~ 7。
串行端口重启位	A526.00	置 ON 以重启串行端口。	读 / 写	<ul style="list-style-type: none"> 接通电源时清除。 将该位置 ON, 以在任何模式下 (Toolbus 模式除外) 重启串行端口。 注: 重启处理完成后, 此位自动置 OFF。
串行端口出错标志	A528.00 ~ A528.07	这些标志表示串行端口发生的错误类型。 位 0: 不使用 位 1: 不使用 位 2: 校验错误 位 3: 帧错误 位 4: 超限错误 位 5: 超时错误 位 6: 不使用 位 7: 不使用	读 / 写	<ul style="list-style-type: none"> 接通电源时清除。 这些标志表示串行端口发生的错误类型。 串行端口重启时系统自动将这些标志置 OFF。 仅下列位在串行 PLC 链接模式下有效。 主站单元: 位 5: 发生超时错误时置 ON 从站单元: 位 3: 发生帧错误时置 ON 位 4: 发生超限错误时置 ON 位 5: 发生超时错误时置 ON
串行端口设定修改标志	A619.02	对串行端口的通信参数进行修改时置 ON。 ON: 修改执行中 OFF: 修改未执行中	读 / 写	<ul style="list-style-type: none"> 接通电源时清除。 对串行端口的通信参数进行修改时置 ON。 执行修改串行端口设定 (STUP(237)) 指令时置 ON。 参数修改完成后置 OFF。

* 与已有的 1:N NT 链接相同, 在主站 CPU 单元中, 通过读取串行端口 PT 通信标志 (单元 0 ~ 7 对应 A393.00 ~ A393.07), 可检查串行 PLC 链接中的从站单元状态 (通信执行中 / 未执行中)。

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 (Toolbus)	上位链接 (CX-One 除外)	无协议通信	协议宏	NT 链接 (1:N 模式)	串行网关 *2	串行 PLC 链接 *1
CPU 单元	CJ2H-CPU68(-EIP) CJ2H-CPU67(-EIP) CJ2H-CPU66(-EIP) CJ2H-CPU65(-EIP) CJ2H-CPU64(-EIP)	RS-232C	支持	支持	支持	---	支持	支持	不支持
	CJ2M-CPU35 CJ2M-CPU34 CJ2M-CPU33 CJ2M-CPU32 CJ2M-CPU31	RS-232C 或 RS-422A/485*1	支持	支持	支持	---	支持	支持	支持
	CJ2M-CPU15 CJ2M-CPU14 CJ2M-CPU13 CJ2M-CPU12 CJ2M-CPU11	RS-232C	支持	支持	支持	---	支持	支持	支持
串行通信单元	CJ1W-SCU41-V1 CJ1W-SCU31-V1	RS422/485	---	支持	支持 *2	支持	支持	支持	不支持
	CJ1W-SCU21-V1 CJ1W-SCU22 CJ1W-SCU32 CJ1W-SCU42	RS-232C	---	支持	支持 *2	支持	支持	支持	不支持

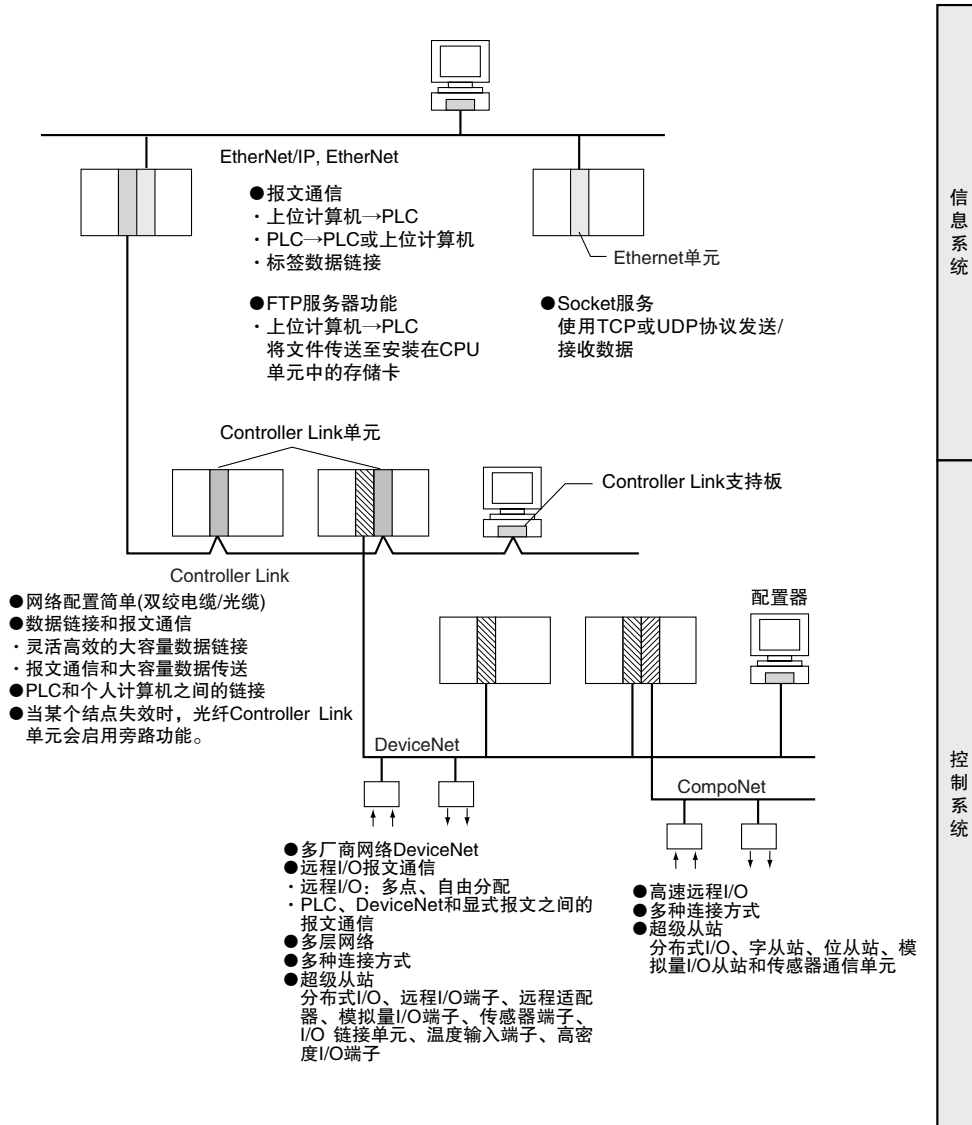
*1 可安装串行选件板来实现 RS-232C 或 RS-422A/485 通信。

*2 仅限 1.2 版或更高版本的串行通信单元。而且对于 CPU 单元, 只有串行网关协议才会支持自动转换至 CompoWay/F。

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。

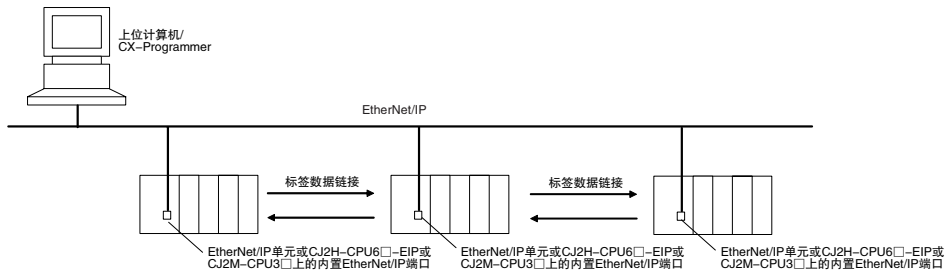


● EtherNet/IP 和 Ethernet

可通过 Ethernet 电缆进行标签数据链接和 FINS 网络通信。此外，通过连接到 Ethernet 的上位计算机执行 PLC 的 FTP 命令，即可对安装在 CPU 单元存储卡中的文件内容进行读写操作。数据收发时采用 UDP 和 TCP 协议，这些功能可提升与上位信息网络的兼容性。

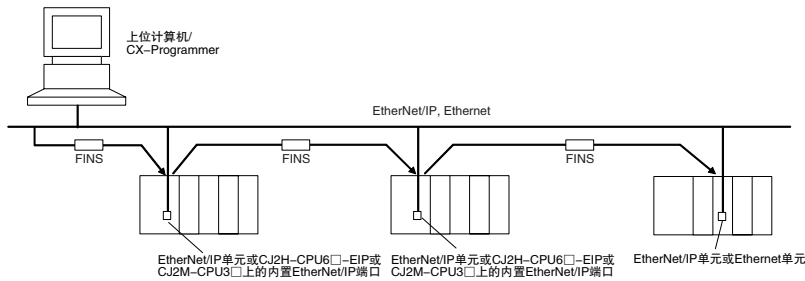
标签数据链接

通过在每个区自定义刷新周期，可在两台 PLC 之间执行数据链接功能，以实现高速、大容量的数据传送。



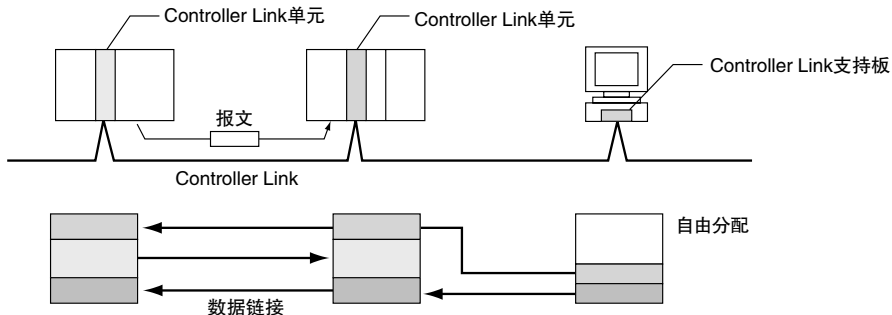
FINS 网络通信

连接到内置 EtherNet/IP 端口或与 EtherNet/IP 单元或 Ethernet 单元相连的 CPU 单元后，即可与 Ethernet 网络中的其它 PLC 以及在 PLC 和上位计算机之间实现 FINS 报文通信。



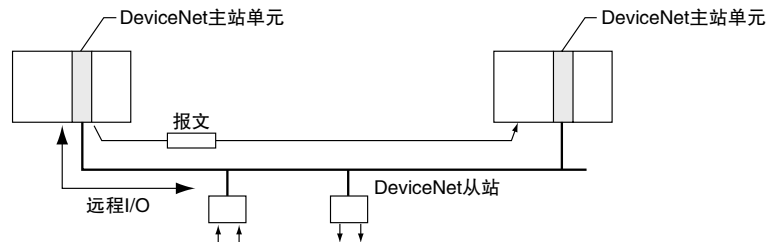
● Controller Link

Controller Link 网络是 OMRON PLC FA 领域的专用网络。将 Controller Link 单元连接到允许在 PLC 之间建立数据链接的网络，即可在编程的情况下共享数据，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 和个人计算机之间也可进行数据链接和报文通信。数据链接支持大容量数据传送和自由分配，而 FINS 报文通信也支持大容量数据传送。



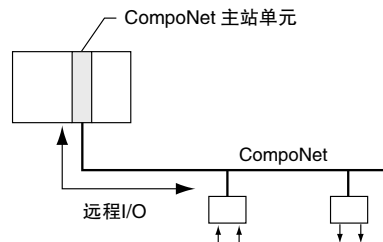
● DeviceNet

DeviceNet 是由多位控制和信息系统组成的多厂商网络技术，符合开放现场 DeviceNet 规格。若要在网络中的 PLC 和从站之间建立远程 I/O 通信，只需将 DeviceNet 主站单元连接至网络即可。远程 I/O 通信支持大容量 I/O 和用户自定义分配，并提供用于从站的模拟量 I/O 端子。PLC 之间以及 PLC 和其它公司生产的 DeviceNet 设备之间可进行报文通信。



● CompoNet

CompoNet 是一种用于实现 I/O 控制的高速、多点、多结点、多厂商网络技术。若要在网络中的 PLC 和 CompoNet 从站之间建立远程 I/O 通信，只需将 CompoNet 主站单元连接至网络即可。CompoNet 可在 1ms 循环时间内实现 1,024 点高速通信，并可连接 2,560 个点和 384 个结点。



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	CJ2H-CPU6□-EIP 或 CJ2M-CPU3□的内置 EtherNet/IP 端口
		PLC 之间		
	上位计算机和安装在 CPU 单元中的存储卡之间	FTP 服务器	EtherNet/IP 单元	
信息网络	Controller Link	PLC 和直接连接到网络的个人计算机之间	FINS 报文通信 数据链接 (偏移量, 简单设定)	Controller Link 支持板或 Controller Link 单元
	RS-232C → Controller Link	上位计算机和网络中的 PLC 之间	上位链接指令和网关	RS-232C 电缆和 Controller Link 单元
控制网络	EtherNet/IP	PLC 之间	标签数据链接	CJ2H-CPU6□-EIP 的内置 EtherNet/IP 端口 EtherNet/IP 单元
	Controller Link	PLC 之间	FINS 报文通信	Controller Link 单元
			数据链接 (偏移量, 简单设定)	
	DeviceNet	PLC 和网络设备 (从站)	开放式网络中的 FINS 报文通信	DeviceNet 主站单元和配置器
	DeviceNet		开放式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	开放式网络中的高速、多点、多结点远程 I/O		CompoNet 主站单元	

12

CPU 单元循环时间

12

本章描述了在 CPU 单元中进行处理所需的循环时间。

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	12-4
12-2-1 CPU 单元操作流程图	12-4
12-2-2 循环时间概述	12-5
12-2-3 各单元的 I/O 单元刷新时间	12-7
12-2-4 循环时间计算示例	12-10
12-2-5 联机编辑引起的循环时间延长	12-12
12-2-6 I/O 响应时间	12-12
12-2-7 输入中断的响应时间	12-13
12-2-8 串行 PLC 链接的响应性能	12-14

12-1 监控循环时间

12-1-1 监控循环时间

当 CX-Programmer 与 CPU 单元联机时，可监控循环时间的平均值、最大值和最小值。

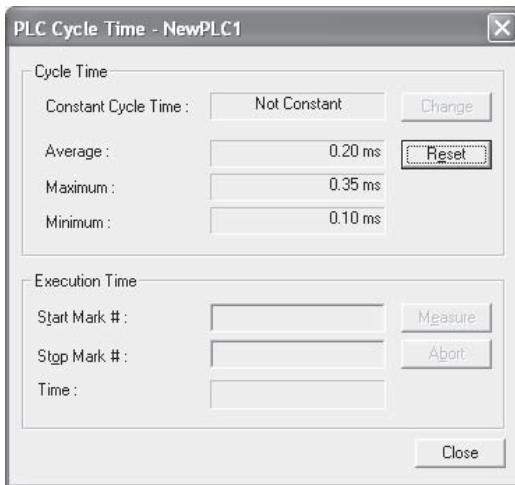
监控平均值

与 PLC 联机时，若 CPU 单元在 PROGRAM 模式以外的任何模式下运行，则将在状态栏中显示平均循环时间。



监控最大值和最小值

在 PLC 菜单中选择“Edit”（编辑）-“Cycle Time”（循环时间），将显示下述“PLC Cycle Time”（PLC 循环时间）对话框。



对话框中自上而下依次显示平均（平均值）、最大和最小循环时间。点击“Reset”（复位）按钮来重新计算和显示循环时间值。



附加信息

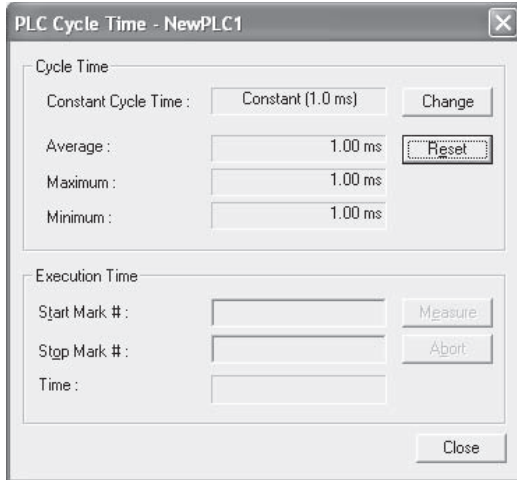
平均循环时间 (= 当前值) 和最大循环时间被存储在下列辅助区字中。

- 当前循环时间 (= 当前值) (单位: 0.1ms)
A264(低字节) 和 A265(高字节)
- 当前循环时间 (= 当前值) (单位: 0.01ms)
A266(低字节) 和 A267(高字节)
- 最大循环时间 (单位: 0.01ms)
A262(低字节) 和 A263(高字节)

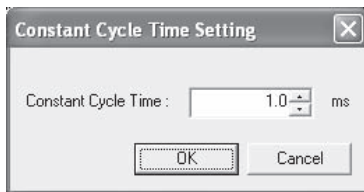
修改最小循环时间

设定最小循环时间时，可在 CPU 单元处于 MONITOR 模式的情况下修改最小循环时间。

1. 在 PLC 菜单中选择“PLC Info”(PLC 信息) - “Cycle Time”(循环时间)，将显示下述“PLC Cycle Time”(PLC 循环时间)对话框。



2. 点击“Change”(修改)按钮，将显示下述“Constant Cycle Time Setting”(恒定循环时间设定)对话框。

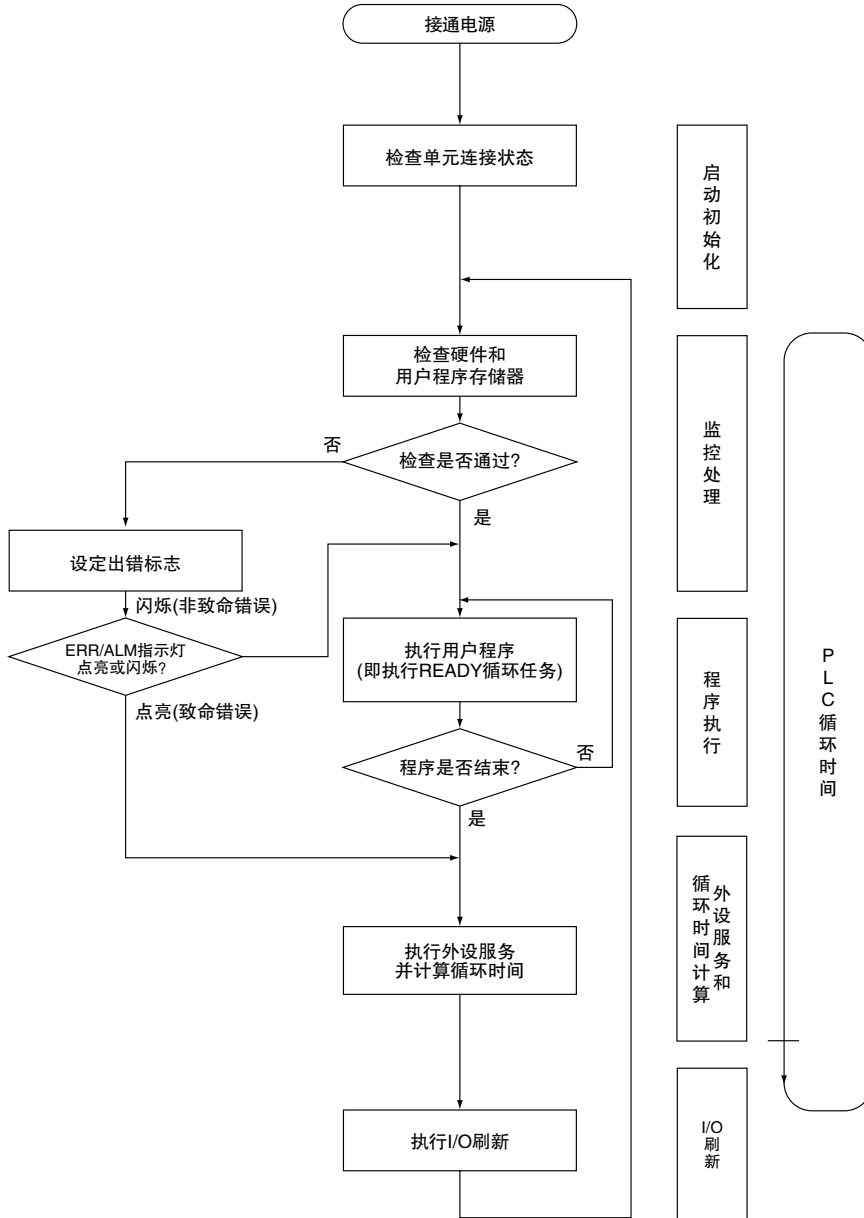


3. 修改“Constant Cycle Time”(恒定循环时间)调节框中的时间值并点击“OK”(确定)按钮，完成最小循环时间的修改。

12-2 计算循环时间

12-2-1 CPU 单元操作流程

下列流程图对 CJ 系列 CPU 单元从监控处理到 I/O 刷新的重复循环中的数据处理过程进行了说明。



12-2-2 循环时间概述

循环时间取决于下列条件：

- 用户程序 (在一个循环内执行的所有循环任务，以及满足执行条件的中断任务中) 中的指令类型和数量
- 基本 I/O 单元的类型和数量
- 高性能 I/O 单元和 CPU 总线单元的类型和数量
- 下列单元 / 选件板的特定服务
 - EtherNet/IP 标签数据链接刷新和标签数据链接字的交换数量
 - 数据链接刷新和 Controller Link 数据链接字数量
 - DeviceNet 单元的远程 I/O 和远程 I/O 字数
 - 协议宏的使用和最大通信报文
 - Ethernet 单元特定控制位的 Socket 服务和发送 / 接收字的数量
- PLC 设置中的恒定循环时间设定
- 对文件存储器的访问操作，以及文件存储器的进出数据量
- 高性能 I/O 单元、CPU 总线单元和通信端口的事件服务
- 外设 USB 端口和串行端口的使用
- PLC 设置中的固定外设服务执行时间

注 1 循环时间不受用户程序内所使用的任务数量的影响，而受循环中处于 READY 状态的循环任务的影响。

2 当运行模式从 MONITOR 切换到 RUN 时，循环时间将延长 10ms(这并不会导致循环时间超出限定值)。

3 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 的内置 EtherNet/IP 端口的性能因 EtherNet/IP 标签数据链接的数据处理方式而异。详情请参考《EtherNet/IP 单元操作手册》(样本编号：W465)。

循环时间是 PLC 在执行下列 5 种操作时所需的总时间。

循环时间 = (1)+(2)+(3)+(4)+(5)

1. 监控

详细信息	处理时间和起伏原因
检查 I/O 总线和用户程序存储器，确认是否存在电池错误并刷新时钟。	CJ2H-CPU6□-EIP: 0.2ms CJ2H-CPU6□: 0.1ms CJ2M-CPU1□: 0.16ms CJ2M-CPU3□: 0.27ms

2. 程序执行

详细信息	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需的总时间。	总指令执行时间

3. 循环时间计算

详细信息	处理时间和起伏原因
在 PLC 设置中设定最小 (固定的) 循环时间的情况下，在指定循环时间过后计算循环时间。	若未固定循环时间，则此步骤 3 所需的时间约为 0。 若固定了循环时间，步骤 3 所需的时间为预设恒定循环时间减去实际循环时间 ((1)+(2)+(4)+(5))。

4. I/O 刷新

详细信息			处理时间和起伏原因
基本 I/O 单元	刷新基本 I/O 单元：首先刷新从 CPU 单元到输出单元的输出，然后刷新从输入单元到 CPU 单元的输入。		各单元的 I/O 刷新时间乘以所使用的单元数。
高功能 I/O 单元	CIO 区分配字		各单元的 I/O 刷新时间乘以所使用的单元数。
	单元特定数据	例如：CompoBus/S 远程 I/O	
CPU 总线单元	在 CIO 和 DM 区中分配的字		各单元的 I/O 刷新时间乘以所使用的单元数。
	单元特定数据	示例： <ul style="list-style-type: none"> · 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 · 协议宏的数据发送 / 接收 · Ethernet 单元特定控制位的 Socket 服务 	

5. 外设服务

支持以下外围设备服务。

- 通信端口 (内部逻辑端口) 服务
当使用梯形图程序中的通信指令或者后台执行或简易备份等功能时，将执行通信端口服务。
- 高功能 I/O 单元或 CPU 总线单元的事件服务
无论何时将单元连接到 PLC 中，都将执行事件服务。在通过通信单元使用支持软件的情况下，也将执行事件服务。
- USB 端口服务
使用 USB 端口时，将执行 USB 端口服务。
- RS-232C 端口服务
使用 RS-232C 端口时，将执行 RS-232C 端口服务。
- 文件访问服务
当访问存储卡或 EM 文件存储器时，将执行文件访问服务。
- 联机编辑服务
当执行联机编辑时，将执行联机编辑服务。
- 备份服务
当修改程序或参数时，将执行备份服务。

当未执行外围设备服务时，无需预留处理时间。外围设备服务的最短和最长处理时间取决于 PLC 设置中的“Set Time to All Events” (所有事件的设定时间) 参数，如下表所示。

		外设服务执行时间	
		最小值	最大值
“Set Time to All Events” (所有事件的设定时间) 参数	禁止	0ms(未执行服务时)	循环时间的 10%(或当循环时间的 10% 小于 0.1ms 时为 0.1ms)
	允许		设定时间 (0.1 ~ 3,276.7ms)

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	各单元的 I/O 刷新时间	
		CJ2H	CJ2M
8/16 点 DC 输入单元	CJ1W-ID201/211/212	0.0014ms	0.0039ms
32 点 DC 输入单元	CJ1W-ID231/232/233	0.0023ms	0.0054ms
64 点 DC 输入单元	CJ1W-ID261/262	0.0041ms	0.0089ms
8/16 点 AC 输入单元	CJ1W-IA201/111	0.0014ms	0.0039ms
16 点中断输入单元	CJ1W-INT01	0.0014ms	0.0039ms
16 点快速响应输入单元	CJ1W-IDP01	0.0014ms	0.0039ms
8/16 点继电器输出单元	CJ1W-OC201/211	0.0014ms	0.0039ms
8 点晶闸管输出单元	CJ1W-OA201	0.0014ms	0.0039ms
8/16 点晶体管输出单元	CJ1W-OD201/202/203/204/211/212/213	0.0014ms	0.0039ms
32 点晶体管输出单元	CJ1W-OD231/232/233/234	0.0023ms	0.0054ms
64 点晶体管输出单元	CJ1W-OD261/262/263	0.0041ms	0.0089ms
24VDC 输入 / 晶体管输出单元 (16 点输入 / 16 点输出)	CJ1W-MD231/232/233	0.0023ms	0.0054ms
24VDC 输入 / 晶体管输出单元 (32 点输入 / 32 点输出)	CJ1W-MD261/263	0.0041ms	0.0089ms
TTL 输入 / TTL 输出单元 (16 点输入 / 16 点输出)	CJ1W-MD563	0.0041ms	0.0089ms
B7A 接口单元 (64 点输入)	CJ1W-B7A14	0.0041ms	0.0089ms
B7A 接口单元 (64 点输出)	CJ1W-B7A04	0.0041ms	0.0089ms
B7A 接口单元 (32 点输入 / 32 点输出)	CJ1W-B7A22	0.0041ms	0.0089ms

● 典型的高功能 I/O 单元刷新时间

名称	型号		各单元的 I/O 刷新时间	
			CJ2H	CJ2M
模拟量输入单元	CJ1W-AD041/081(V1)		0.05ms	0.07ms
	CJ1W-AD042		0.05ms	0.07ms
模拟量输出单元	CJ1W-DA021/041/08V		0.05ms	0.07ms
	CJ1W-DA042V		0.05ms	0.07ms
模拟量 I/O 单元	CJ1W-MAD42		0.05ms	0.07ms
温度控制单元	CJ1W-TC□□□□		0.05ms	0.25ms
位置控制单元	CJ1W-NC113/133		0.12ms	0.13ms
			用于传送数据的每条指令 (IOWR/IORD)+0.8ms	
	CJ1W-NC213/233		0.14ms	0.15ms
			用于传送数据的每条指令 (IOWR/IORD)+0.8ms	
	CJ1W-NC413/433		0.17ms	0.18ms
			用于传送数据的每条指令 (IOWR/IORD)+0.7ms	
ID 传感器单元	CJ1W-V600C11		0.12ms	0.13ms
	CJ1W-V600C12		0.14ms	0.15ms
	CJ1W-CT021		0.10ms	0.11ms
高速计数器单元	CJ1W-CT021		0.10ms	0.11ms
CompoNet 主站单元	CJ1W-CRM21	通信模式 0	0.13ms	0.14ms
		通信模式 1	0.14ms	0.15ms
		通信模式 2	0.17ms	0.17ms
		通信模式 3	0.21ms	0.22ms
		通信模式 8(*)	0.081 + (0.0012 × 分配的字数)ms	0.083 + (0.009 × 分配的字数)ms
CompoBus/S 主站单元	CJ1W-SRM21	分配 1 个单元号	0.08ms	0.08ms
		分配 2 个单元号	0.08ms	0.10ms

* 分配的字数为 I/O 存储器中实际分配给所连接从站的字数。

● 因 CPU 总线单元引起的循环时间延长

名称	型号	延长的时间		说明
		CJ2H	CJ2M	
Controller Link 单元	CJ1W-CLK23	0.1ms	0.2ms	使用数据链接时需增加下列时间。 · CJ2H CPU 单元: $0.1\text{ms} + 0.7\mu\text{s} \times \text{数据链接字数}$ · CJ2M CPU 单元: $0.1\text{ms} + 1.2\mu\text{s} \times \text{数据链接字数}$ 当使用报文服务时, 将会使事件执行时间延长。
串行通信单元	CJ1W-SCU41-V1 CJ1W-SCU21-V1 CJ1W-SCU31-V1 CJ1W-SCU42 CJ1W-SCU32 CJ1W-SCU22	0.20ms		执行协议宏时需增加下列时间。 · CJ2H CPU 单元: $0.4\mu\text{s} \times \text{发送或接受的最大数据字数} (0 \sim 500 \text{ 字})$ · CJ2M CPU 单元: $0.6\mu\text{s} \times \text{发送或接受的最大数据字数} (0 \text{ 个字或多个字})$ 当使用上位链接或 1:N NT 链接时, 将会使事件执行时间延长。
Ethernet 单元	CJ1W-ENT11/21	0.1ms	0.9ms	通过软开关执行 Socket 服务时需增加下列时间。 · CJ2H CPU 单元: $1.4\mu\text{s} \times \text{发送/接收的字节数}$ · CJ2M CPU 单元: $0.17\mu\text{s} \times \text{发送/接收的字节数}$ 当执行 FINS 通信服务、CMND 指令的 Socket 服务或 FTP 服务时, 将会使事件执行时间延长。
EtherNet/IP 单元	CJ1W-EIP21	0.1ms		使用标签数据链接时需增加下列时间。 CJ2H CPU 单元 $0.1\text{ms} + \text{传送字数} \times 0.33\mu\text{s}^*$ CJ2M CPU 单元 $0.1\text{ms} + \text{传送字数} \times 0.70\mu\text{s}^*$ 当使用报文服务时, 则必须同时延长事件执行时间。
FL-net 单元	CJ1W-FLN22	$0.25\text{ms} + \text{数据链接字数} \times 1.6\mu\text{s}$	$0.25\text{ms} + \text{数据链接字数} \times 1.7\mu\text{s}$	数据链接字数为某个结点发送和接收的数据字数。当使用报文服务时, 则必须单独延长事件执行时间。
DeviceNet 单元	CJ1W-DRM21	各分配字占用时间为 $0.4\text{ms} + 0.7\mu\text{s}$	各分配字占用时间为 $0.4\text{ms} + 0.4\mu\text{s}$	包含所有分配到从站的字及未使用的字。 对于报文通信, 需将通信字数加入左栏的计算公式中。
模拟量输入单元	CJ1W-ADG41	0.12ms	0.15ms	---
位置控制单元 (带 EtherCAT 通信模块)	CJ1W-NC881 (8 轴, 4 任务)	0.3ms	0.5ms	---
	CJ1W-NC481 (4 轴, 4 任务)	0.3ms	0.4ms	---
	CJ1W-NC281 (2 轴, 2 任务)	0.2ms	0.3ms	---
	CJ1W-NCF81 (16 轴, 4 任务)	0.5ms	0.7ms	---
	CJ1W-NC482 (4 轴, 4 任务)	0.3ms	0.4ms	远程输入: 20 字, 远程输出: 20 字
		0.7ms	1.0ms	远程输入: 640 字, 远程输出: 640 字
	CJ1W-NC882 (8 轴, 4 任务)	0.3ms	0.5ms	远程输入: 20 字, 远程输出: 20 字
		0.8ms	1.1ms	远程输入: 640 字, 远程输出: 640 字
位置控制单元 (带 MECHATROLINK-II 通信模块)	CJ1W-NCF71 (-MA)	根据连接到单元的伺服驱动轴数。 1 轴: 0.2ms, 3 轴: 0.3ms, 6 轴: 0.4ms, 16 轴: 0.8ms		---
运动控制单元 (带 MECHATROLINK-II 通信模块)	CJ1W-MCH71	$0.2 + (\text{运动任务数} \times 4 + \text{轴数} \times 5 + \text{分配到通用 I/O 的字数}) \times 0.001(\text{ms})$	$0.2 + (\text{运动任务数} \times 6 + \text{轴数} \times 1 + \text{分配到通用 I/O 的字数}) \times 0.001(\text{ms})$	---
SYSMAC SPU 单元	CJ1W-SPU01-V2	$0.2\text{ms} + \text{采样字数} \times 0.8\mu\text{s}$	$0.2\text{ms} + \text{采样字数} \times 1.1\mu\text{s}$	---

* 当高速中断功能结合 1.1 版或更高版本的 CJ2H CPU 单元使用时, 必须增加下列时间。

$$0.1\text{ms} + \text{传送字数} \times 0.87\mu\text{s}$$

12-2-4 循环时间计算示例

● 示例 1：基于基本指令和基本 I/O 单元的应用

下例所示为当 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 20K 步的基本和数据传送指令组成。此处使用了 CJ2H-CPU6□-EIP CPU 单元。

条件

项目	详细信息	
CPU 单元	CJ2H-CPU6□-EIP	
CJ 系列 CPU 单元	CJ1W-ID261 64 点输入单元	2 个
	CJ1W-OD261 64 点输出单元	2 个
用户程序	20K 步	LD 指令：10K 步 MOV 指令：10K 步 (LD：每条指令占用 1 步 MOV：每条指令占用 3 步)
外设 USB 端口连接	有或无	
恒定循环时间处理	无	
串行端口连接	无	
其它设备 (高性能 I/O 单元、CPU 总线单元和文件访问) 的外设服务	无	

计算示例

处理名称	计算	处理时间	
		连接到外设 USB 端口的编程设备	未连接到外设 USB 端口的编程设备
(1) 监控	---	0.20ms	0.20ms
(2) 程序执行	$0.016 \mu\text{s} \times 10,000 + 0.14 \mu\text{s} / 3 \text{ 步} \times 10,000$	0.63ms	0.63ms
(3) 最小循环时间的计算	(未设定恒定循环时间)	0ms	0ms
(4) I/O 刷新	$0.0039\text{ms} \times 2 + 0.0039\text{ms} \times 2$	0.0164ms	0.0164ms
(5) 外设服务	(仅限外设 USB 端口连接)	0.1ms	0ms
循环时间	(1) + (2) + (3) + (4) + (5)	0.95ms	0.85ms

● 示例 2: 包含计算和高功能 I/O 单元的应用

下例显示了当 PLC 中只连接了基本 I/O 单元和高功能 I/O 单元时用于计算循环时间的方法，程序由 20K 步的基本指令、数据传送指令和浮点计算指令组成。此处以 CJ2H-CPU6□-EIP CPU 单元为例。

条件

项目	详细信息	
CPU 单元	CJ2H-CPU6□-EIP	
CJ 系列 CPU 单元	CJ1W-ID261 64 点输入单元	2 个
	CJ1W-OD261 64 点输出单元	2 个
	CJ1W-AD081 模拟量输入单元	2 个
	CJ1W-NC413 位置控制单元	2 个
用户程序	20K 步	LD 指令: 12K 步 MOV 指令: 6K 步 +F 指令: 2K 步 (LD: 每条指令占用 1 步 MOV: 每条指令占用 3 步 +F: 每条指令占用 4 步)
外设 USB 端口连接	有或无	
恒定循环时间处理	无	
串行端口连接	无	
其它设备 (高功能 I/O 单元、CPU 总线单元和文件访问) 的外设服务	无	

计算示例

处理名称	计算	处理时间	
		连接到内置 USB 端口的编程设备	未连接到内置 USB 端口的编程设备
(1) 监控	---	0.20ms	0.20ms
(2) 程序执行	$0.016\text{ms} \times 12,000 + 0.14 \mu\text{s}/$ $3 \text{步} \times 6,000 + 0.24 \mu\text{s}/$ $4 \text{步} \times 2,000$	0.59ms	0.59ms
(3) 最小循环时间的计算	(未设定恒定循环时间)	0ms	0ms
(4) I/O 刷新	$0.0041\text{ms} \times 2 + 0.0041\text{ms} \times 2 +$ $0.05\text{ms} \times 2 + 0.17\text{ms} \times 2$	0.4564ms	0.4564ms
(5) 外设服务	(仅限外设 USB 端口连接)	0.1ms	0ms
循环时间	(1) + (2) + (3) + (4) + (5)	1.35ms	1.25ms

12-2-5 联机编辑引起的循环时间延长

当 CPU 单元在 MONITOR 模式下运行时，可通过 CX-Programmer 执行联机编辑。由于需对 CPU 单元程序进行写处理，循环时间将延长约 1ms。如果不想延长循环时间，可使用联机编辑禁止(待机)功能。详情请参考“10-7-4 联机编辑”。

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置 ON、CJ 系列 CPU 单元识别出数据、用户程序执行到将结果输出到输出单元的输出端子所需的时间。

I/O 响应时间长短取决于以下条件。

- 输入位置 ON 的时间
- 循环时间
- 装有输入和输出单元的机架类型 (CPU 机架或扩展机架)

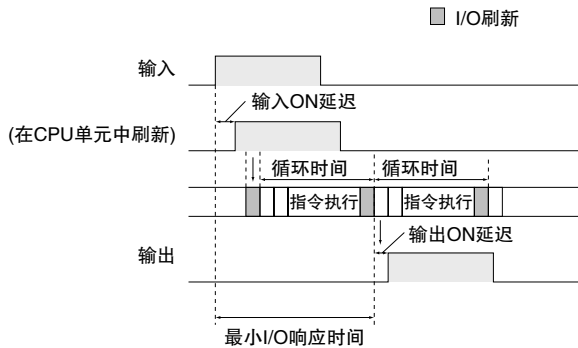
CJ 系列基本 I/O 单元的响应时间

● 最小 I/O 响应时间

紧接在 CPU 单元 I/O 刷新前检索数据时 I/O 响应时间最短。

最小 I/O 响应时间等于输入 ON 延迟、循环时间和输出 ON 延迟的总和。

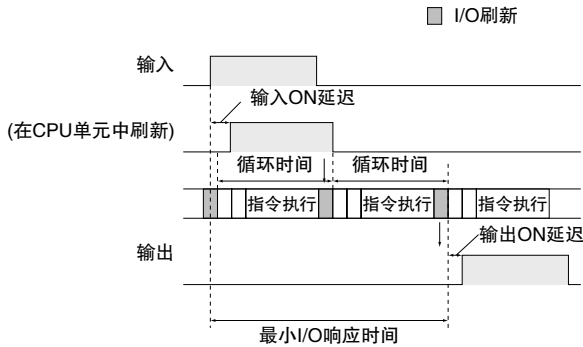
注 输入和输出 ON 延迟因使用的单元类型而异。



● 最大 I/O 响应时间

紧接在输入单元 I/O 刷新后检索数据时，I/O 响应时间最长。

最大 I/O 响应时间等于输入 ON 延迟、(循环时间 × 2) 和输出 ON 延迟的总和。



● 计算示例

条件：

输入 ON 延迟：1.5ms

输出 ON 延迟：0.2ms

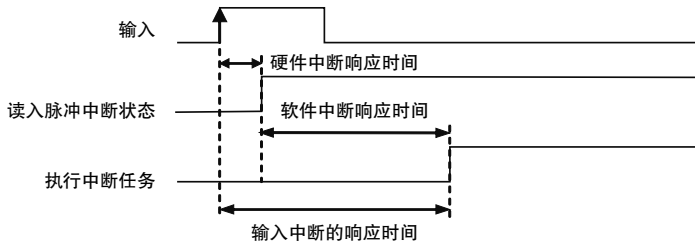
循环时间：20.0ms

最小 I/O 响应时间 = 1.5ms + 20ms + 0.2ms = 21.7ms

最大 I/O 响应时间 = 1.5ms + (20ms × 2) + 0.2ms = 41.7ms

12-2-7 输入中断的响应时间

输入中断的响应时间是指从 CJ2M CPU 单元所连接的脉冲 I/O 模块的脉冲输入置 ON 或置 OFF 到 I/O 中断任务执行所需的时间。输入中断任务的中断响应时间长短取决于硬件中断响应时间和软件中断响应时间的总和。



输入中断的响应时间 = 硬件中断响应时间 + 软件中断响应时间

硬件中断响应时间

上升沿 / 下降沿微分	中断响应时间
上升沿	30 μ s
下降沿	150 μ s

软件中断响应时间

中断类型	中断响应时间
直接模式下的中断输入	最小值: 33 μ s
计数器模式下的中断输入	最小值: 34 μ s

12-2-8 串行 PLC 链接的响应性能

通过串行 PLC 链接连接的 CPU 单元的响应时间 (主站到从站或从站到主站) 计算如下。

这些计算是在外设服务时间的长度足以处理所需通信时间的前提下进行的。

- 最大 I/O 响应时间 (不包括硬件延迟) [ms] =
主站单元循环时间 \times 2 + 通信循环时间 + 从站单元循环时间 \times 2 + 从站单元通信时间 + 4ms
- 最小 I/O 响应时间 (不包括硬件延迟) [ms] =
从站单元通信时间 + 0.54ms

参与从站结点数	已建立链接的从站单元数 (在主站设定的最大单元号范围内)。
非参与从站结点数	未参与链接的从站单元数 (在主站设定的最大单元号范围内)。
通信循环时间 [ms]	<ul style="list-style-type: none"> · 带两个或两个以上的从站单元 从站单元通信时间 \times 参与从站结点数 + 10 \times 非参与从站结点数 · 带一个从站单元 从站单元通信时间 + 10ms
从站单元通信时间 [ms]	<ul style="list-style-type: none"> · 波特率设定为标准模式: 主站单元循环时间 + 0.286 \times [(从站单元数 + 1) \times 链接字数 \times 2 + 12] · 波特率设定为快速模式: 主站单元循环时间 + 0.0955 \times [(从站单元数 + 1) \times 链接字数 \times 2 + 12]



附录

附录中包含关于电源中断时 CPU 单元的运行情况、CPU 单元支持的指令、辅助区字和位、指令执行时间、PLC 存储器地址的存储器映射以及 CJ 系列和 CS 系列 PLC 的比较等详细信息。

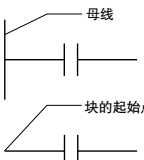
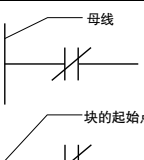
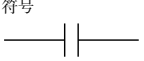

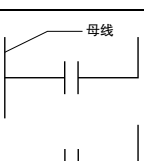
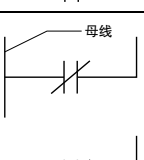
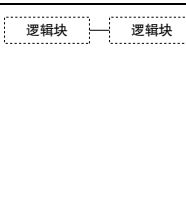
A-1 指令功能	A-3
A-1-1 顺序输入指令	A-3
A-1-2 顺序输出指令	A-5
A-1-3 顺序控制指令	A-7
A-1-4 定时器和计数器指令	A-10
A-1-5 比较指令	A-14
A-1-6 数据传送指令	A-18
A-1-7 数据移位指令	A-20
A-1-8 递增 / 递减指令	A-24
A-1-9 四则运算指令	A-24
A-1-10 转换指令	A-29
A-1-11 逻辑指令	A-35
A-1-12 特殊算术指令	A-37
A-1-13 浮点算术运算指令	A-38
A-1-14 双精度浮点数指令	A-42
A-1-15 表格数据处理指令	A-44
A-1-16 跟踪指令	A-49
A-1-17 数据控制指令	A-50
A-1-18 子程序指令	A-53
A-1-19 中断控制指令	A-54
A-1-20 高速计数器 / 脉冲输出指令	A-56
A-1-21 步指令	A-58
A-1-22 基本 I/O 单元指令	A-58
A-1-23 串行通信指令	A-61
A-1-24 网络指令	A-62
A-1-25 文件存储指令	A-65
A-1-26 显示指令	A-66
A-1-27 时钟指令	A-67
A-1-28 调试指令	A-68
A-1-29 故障诊断指令	A-68
A-1-30 其它指令	A-69
A-1-31 块编程指令	A-70
A-1-32 文本字符串处理指令	A-74
A-1-33 任务控制指令	A-76

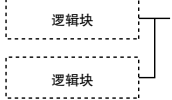
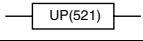
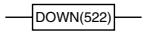
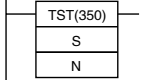
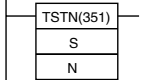
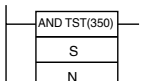
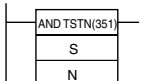
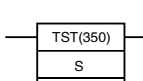
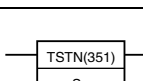
A-1-34 型号转换指令	A-77
A-1-35 特殊功能块指令	A-78
A-2 指令执行时间和步数	A-79
A-2-1 顺序输入指令	A-81
A-2-2 顺序输出指令	A-82
A-2-3 顺序控制指令	A-83
A-2-4 定时器和计数器指令	A-84
A-2-5 比较指令	A-85
A-2-6 数据传送指令	A-86
A-2-7 数据移位指令	A-87
A-2-8 递增 / 递减指令	A-88
A-2-9 四则运算指令	A-88
A-2-10 转换指令	A-89
A-2-11 逻辑指令	A-91
A-2-12 特殊算术指令	A-91
A-2-13 浮点算术运算指令	A-92
A-2-14 双精度浮点数指令	A-93
A-2-15 表格数据处理指令	A-94
A-2-16 跟踪指令	A-96
A-2-17 数据控制指令	A-96
A-2-18 子程序指令	A-97
A-2-19 中断控制指令	A-97
A-2-20 高速计数器 / 脉冲输出指令	A-97
A-2-21 步指令	A-99
A-2-22 基本 I/O 单元指令	A-99
A-2-23 串行通信指令	A-100
A-2-24 网络指令	A-101
A-2-25 文件存储指令	A-101
A-2-26 显示指令	A-101
A-2-27 时钟指令	A-101
A-2-28 调试指令	A-102
A-2-29 故障诊断指令	A-102
A-2-30 其它指令	A-102
A-2-31 块编程指令	A-103
A-2-32 文本字符串处理指令	A-104
A-2-33 任务控制指令	A-105
A-2-34 型号转换指令	A-105
A-2-35 特殊功能块指令	A-105
A-2-36 SFC 指令	A-106
A-2-37 功能块实例执行时间	A-106
A-3 辅助区	A-108
A-3-1 只读区 (由系统设定)	A-108
A-3-2 读 / 写区 (由用户设定)	A-138
A-3-3 辅助区操作的详细信息	A-146
A-4 PLC 存储器地址的存储器映射	A-149
A-4-1 PLC 存储器地址	A-149
A-4-2 存储器映射	A-150
A-5 断电时的操作	A-151
A-5-1 断电操作	A-151
A-5-2 断电的指令执行	A-152
A-6 在 Windows XP(SP2 或以上)、Windows Vista 或 Windows 7 中 进行 EtherNet/IP 连接	A-155
A-6-1 更改 Windows 防火墙设定	A-155
A-7 PLC 比较表: CJ 系列和 CS 系列 PLC	A-157
A-8 各单元版本支持的功能	A-161

A-1 指令功能

CJ2 CPU 单元支持下述指令。详情请参考《CS/CJ/NSJ 系列指令参考手册》(样本编号: W474)。

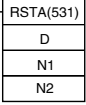
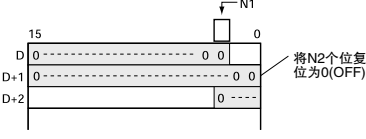



A-1-1 顺序输入指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
载入	LD @LD %LD !LD !@LD !%LD	---		指定一个逻辑开始并根据指定操作位的 ON/OFF 状态建立一个 ON/OFF 执行条件。	逻辑开始	不需要
载入非	LD NOT @LD NOT*2 %LD NOT !LD NOT !@LD NOT !%LD NOT	---		指定一个逻辑开始并根据指定操作位的 ON/OFF 状态取反建立一个 ON/OFF 执行条件。	逻辑开始	不需要
与	AND @AND %AND !AND !@AND !%AND	---	符号 	把指定操作位状态和当前执行条件进行逻辑与操作。	在梯级上连续	需要
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反并和当前执行条件进行逻辑与。	在梯级上连续	需要
或	OR @OR %OR !OR !@OR !%OR	---		将指定操作位的 ON/OFF 状态和当前执行条件进行逻辑或操作。	在梯级上连续	需要
或非	OR NOT @OR NOT %OR NOT !OR NOT !@OR NOT !%OR NOT	---		将指定操作位的状态取反后和当前执行条件进行逻辑或操作。	在梯级上连续	需要
逻辑块与	AND LD	---		在逻辑块之间执行逻辑与操作。 LD } 逻辑块 A ~ } LD } 逻辑块 B ~ } AND LD 把逻辑块A和逻辑块B串联起来	在梯级上连续	需要


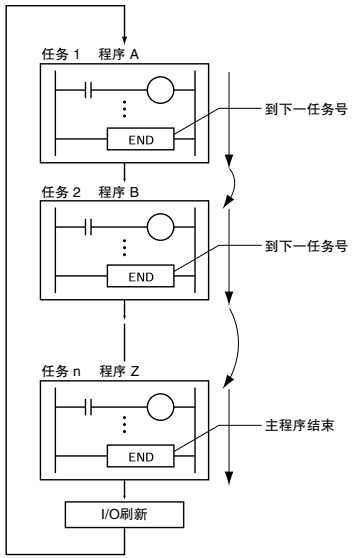

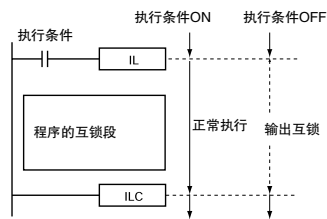
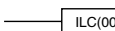
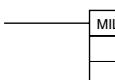
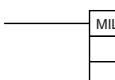
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
逻辑块或	OR LD	---		<p>在逻辑块之间执行逻辑或操作。</p> <p>LD } ~ } 逻辑块 A</p> <p>LD } ~ } 逻辑块 B</p> <p>OR LD 把逻辑块A和逻辑块B并联起来</p>	在梯级上连续	需要
非	NOT	520		执行条件取反	在梯级上连续	需要
条件 ON	UP	521		当执行条件从 OFF → ON 时, UP(521) 将执行条件在一个循环内置 ON。	在梯级上连续	需要
条件 OFF	DOWN	522		当执行条件从 ON → OFF 时, DOWN(522) 将执行条件在一个循环内置 ON。	在梯级上连续	需要
位测试	LD TST	350	 <p>S: 源字 N: 位号</p>	LD TST(350)、AND TST(350) 和 OR TST(350) 在程序中的作用类似于 LD、AND 和 OR；当指定字中的指定位为 ON 时, 执行条件为 ON, 指定位为 OFF 时, 执行条件为 OFF。	逻辑开始	不需要
位测试	LD TSTN	351	 <p>S: 源字 N: 位号</p>	LD TSTN(351)、AND TSTN(351) 和 OR TSTN(351) 在程序中的作用类似于 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, 指定位为 OFF 时, 执行条件为 ON。	逻辑开始	不需要
位测试	AND TST	350	 <p>S: 源字 N: 位号</p>	LD TST(350)、AND TST(350) 和 OR TST(350) 在程序中的作用类似于 LD、AND 和 OR；当指定字中的指定位为 ON 时, 执行条件为 ON, 指定位为 OFF 时, 执行条件为 OFF。	在梯级上连续	需要
位测试	AND TSTN	351	 <p>S: 源字 N: 位号</p>	LD TSTN(351)、AND TSTN(351) 和 OR TSTN(351) 在程序中的作用类似于 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, 指定位为 OFF 时, 执行条件为 ON。	在梯级上连续	需要
位测试	OR TST	350	 <p>S: 源字 N: 位号</p>	LD TST(350)、AND TST(350) 和 OR TST(350) 在程序中的作用类似于 LD、AND 和 OR；当指定字中的指定位为 ON 时, 执行条件为 ON, 指定位为 OFF 时, 执行条件为 OFF。	在梯级上连续	需要
位测试	OR TSTN	351	 <p>S: 源字 N: 位号</p>	LD TSTN(351)、AND TSTN(351) 和 OR TSTN(351) 在程序中的作用类似于 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, 指定位为 OFF 时, 执行条件为 ON。	在梯级上连续	需要



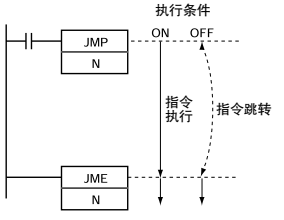

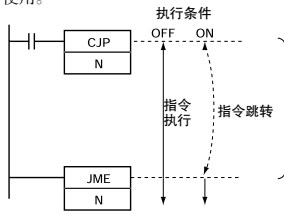

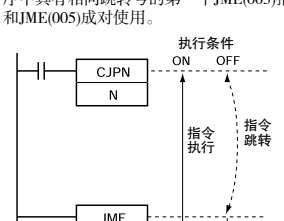

A-1-2 顺序输出指令


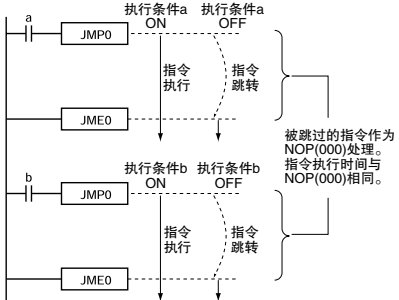



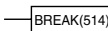
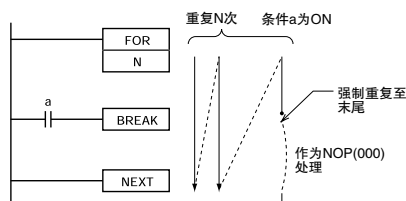

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
输出	OUT !OUT	---		把逻辑运算结果(执行条件)输出到指定位。	输出	需要
反相输出	OUT NOT !OUT NOT	---		把逻辑运算结果(执行条件)取反后再输出到指定位。	输出	需要
保持	KEEP !KEEP	011	 S(置位) R(复位) B: 位	进行锁存继电器操作。 S执行条件 R执行条件 B的状态	输出	需要
上升沿微分	DIFU !DIFU	013	 B: 位	当执行条件从OFF→ON变化(上升沿)时, DIFU(013)将指定位在一个循环内置ON。 	输出	需要
下降沿微分	DIFD !DIFD	014	 B: 位	当执行条件从ON→OFF变化(下降沿)时, DIFD(014)将指定位在一个循环内置ON。 	输出	需要
置位	SET @SET %SET !SET !@SET !%SET	---	 B: 位	当执行条件为ON时, SET指令将操作位置ON。 SET指令的执行条件 B的状态	输出	需要
复位	RSET @RSET %RSET !RSET !@RSET !%RSET	---	 B: 位	当执行条件为ON时, RSET指令将操作位置OFF。 RSET的执行条件 B的状态	输出	需要
多个位置位	SETA @SETA	530	 D: 起始字 N1: 起始位 N2: 位数	SETA(530)将指定的连续位置ON。 	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多个位复位	RSTA @RSTA	531	 <p>D: 起始字 N1: 起始位 N2: 位数</p>	<p>RSTA(531)将指定数量的连续位置OFF。</p> 	输出	需要
单个位置位	SETB @SETB !SETB !@SETB	532	 <p>D: 字地址 N: 位号</p>	<p>执行条件为ON时, SETB(532)将指定字中的指定位置ON。 与 SET 指令不同, SETB(532)可用于对 DM 或 EM 字中的位进行置位。</p>	单个位置位	SETB
单个位复位	RSTB @RSTB !RSTB !@RSTB	533	 <p>D: 字地址 N: 位号</p>	<p>执行条件为 ON 时, RSTB(533) 将指定字中的指定位置 OFF。 与 RSET 指令不同, RSTB(533) 可用于对 DM 或 EM 字中的位进行复位。</p>	输出	需要
单个位输出	OUTB @OUTB !OUTB	534	 <p>D: 字地址 N: 位号</p>	<p>OUTB(534)将逻辑运算的结果(执行条件)输出到指定位。 与 OUT 指令不同, OUTB(534)可用于控制 DM 或 EM 字中的位。</p>	输出	需要

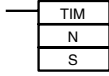
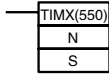
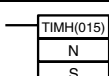
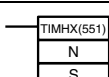
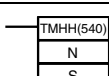
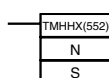
A-1-3 顺序控制指令


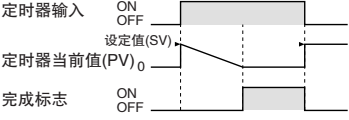

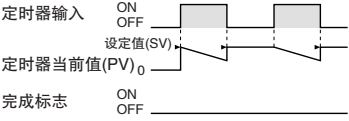

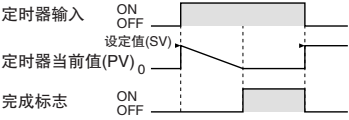

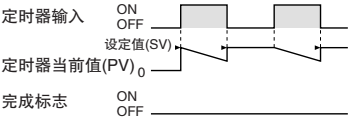
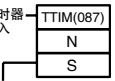
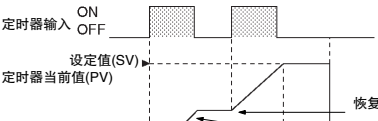
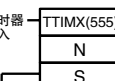
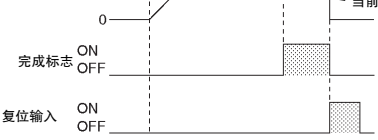
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
结束	END	001		<p>表示程序结束。END(001)完成一个循环内的程序执行。END(001)后面写入的任何指令均不执行。执行转至下一个任务号的程序。当程序执行到该程序中的最大任务号时，END(001)标志着整个主程序的结束。</p> 	输出	不需要
空操作	NOP	000	---	此指令无功能。(NOP(000)不执行任何处理。)	输出	不需要
互锁	IL	002		<p>当IL(002)的执行条件为OFF时，IL(002)和ILC(003)间的所有输出互锁。IL(002)和ILC(003)通常为成对使用。</p> 	输出	需要
互锁清除	ILC	003		当IL(002)的执行条件为OFF时，IL(002)和ILC(003)之间的所有输出均互锁。IL(002)和ILC(003)通常为成对使用。	输出	不需要
多路互锁微分保持	MILH	517	 N: 互锁号 D: 互锁状态位	<p>当MILH(517)的执行条件为OFF时，则MILH(517)指令和下一条MILC(519)指令间的所有指令输出互锁。MILH(517)和MILC(519)通常为成对使用。</p> <p>可将MILH(517)/MILC(519)互锁嵌套使用(如，MILH(517)-MILH(517)-MILC(519)-MILC(519))。</p> <p>如果在MILH(517)与相应的MILC(519)间有微分指令(DIFU、DIFD或者带有@或%前缀的指令)，则若指令的微分条件成立，将在互锁清除后执行指令。</p>	输出	需要
多路互锁微分释放	MILR	518	 N: 互锁号 D: 互锁状态位	<p>当MILR(518)的执行条件为OFF时，则MILR(518)指令和下一条MILC(519)指令间的所有指令输出互锁。MILR(518)和MILC(519)通常为成对使用。</p> <p>MILR(518)/MILC(519)互锁可以嵌套(例如：MILR(518)-MILR(518)-MILC(519)-MILC(519))。</p> <p>如果在MILR(518)与相应的MILC(519)间有微分指令(DIFU、DIFD或者带有@或%前缀的指令)，则即使指令的微分条件成立，在互锁清除后也不执行指令。</p>	输出	需要

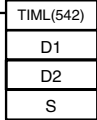
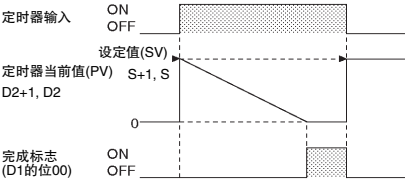
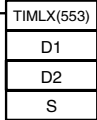
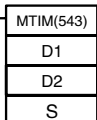
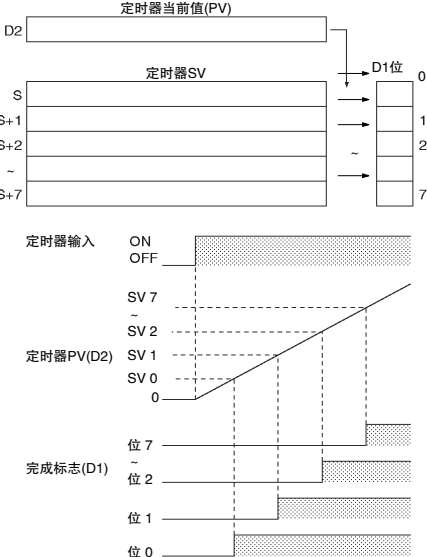
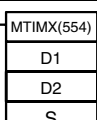
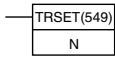
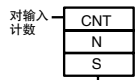
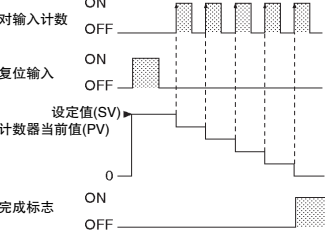
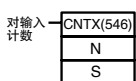
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多路互锁清除	MILC	519	 N: 互锁号	<p>清除以同一个互锁号的 MILH(517) 或 MILR(518) 开始的互锁。</p> <p>当 MILH(517)/MILR(518) 的执行条件为 OFF 时, 使用同一个互锁号的 MILH(517)/MILR(518) 和对应 MILC(519) 间的所有输出互锁。</p>	输出	不需要
跳转	JMP	004	 N: 跳转号	<p>当 JMP(004) 的执行条件为 OFF 时, 程序执行直接跳转至程序中具有同一个跳转号的第一条 JME(005) 指令。JMP(004) 和 JME(005) 成对使用。</p> 	输出	需要
条件跳转	CJP	510	 N: 跳转号	<p>CJP(510) 的操作与 JMP(004) 基本相反。当 CJP(510) 的执行条件为 ON 时, 程序执行直接跳转至与程序中具有同一个跳转号的第一条 JME(005) 指令。CJP(510) 和 JME(005) 成对使用。</p> 	输出	需要
条件跳转非	CJPN	511	 N: 跳转号	<p>CJPN(511) 的作用与 JMP(004) 几乎相同。</p> <p>当 CJPN(004) 的执行条件为 OFF 时, 程序执行直接跳转至程序中具有相同跳转号的第一个 JME(005) 指令。CJPN(511) 和 JME(005) 成对使用。</p> 	输出	不需要
跳转结束	JME	005	 N: 跳转号	表示跳转指令的目的地。	输出	不需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多路跳转	JMPO	515		<p>当JMPO(515)的执行条件为OFF时, 则程序中从JMPO(515)到下一条JME0(516)之间的所有指令都将作为NOP(000)处理。应成对使用JMPO(515)和JME0(516)。程序中能够成对使用这些指令的数量不受限制。</p> 	输出	需要
多路跳转结束	JME0	516		<p>当JMPO(515)的执行条件为OFF时, 则程序中从JMPO(515)到下一条JME0(516)之间的所有指令都将作为NOP(000)处理。应成对使用JMPO(515)和JME0(516)。程序中能够成对使用这些指令的数量不受限制。</p>	输出	不需要
FOR-NEXT 循环	FOR	512	 N: 循环次数	<p>以指定次数重复FOR(512)和NEXT(513)间的指令。FOR(512)和NEXT(513)成对使用。</p> 	输出	不需要
循环中断	BREAK	514		<p>编写在FOR-NEXT循环中, 以在给定执行条件下取消循环的执行。循环中的剩余指令将作为NOP(000)指令处理。</p> 	输出	需要
FOR-NEXT 循环	NEXT	513		<p>以指定次数重复FOR(512)和NEXT(513)间的指令。FOR(512)和NEXT(513)成对使用。</p>	输出	不需要

A-1-4 定时器和计数器指令



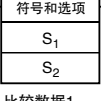
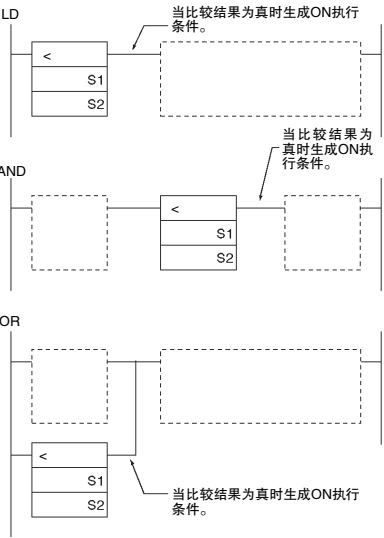
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
100ms 定时器	TIM (BCD)	---	 N: 定时器号 S: 设定值	<p>TIM/TIMX(550)定时器以0.1s为单位作减量计时。对于BCD, 设定值(SV)的设定范围为0~999.9s, 对于二进制(十进制或十六进制), 设定范围为0~6.553.5s。</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p> <p>在完成标志置ON之前, 定时器输入置OFF</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p>	输出	需要
	TIMX (二进制)	550	 N: 定时器号 S: 设定值	<p>TIMH(015)/TIMHX(551)定时器以10ms为单位作减量计时。对于BCD, 设定值(SV)的设定范围为0~99.99s, 对于二进制(十进制或十六进制), 设定范围为0~65.535s。</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p> <p>在完成标志置ON之前, 定时器输入置OFF</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p>		
10ms 定时器	TIMH (BCD)	015	 N: 定时器号 S: 设定值	<p>TIMHX(551)/TIMHX(551)定时器以10ms为单位作减量计时。对于BCD, 设定值(SV)的设定范围为0~99.99s, 对于二进制(十进制或十六进制), 设定范围为0~65.535s。</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p> <p>在完成标志置ON之前, 定时器输入置OFF</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p>	输出	需要
	TIMHX (二进制)	551	 N: 定时器号 S: 设定值	<p>TMHH(540)/TMHHX(552) 定时器以 1ms 为单位作减量计时。对于 BCD, 设定值 (SV) 的设定范围为 0 ~ 9.999s, 对于二进制 (十进制或十六进制), 设定范围为 0 ~ 65.535s。</p> <p>TMHH(540) 和 TMHHX(552) 的时序图与上述 TIMH(015) 的相同。</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p> <p>在完成标志置ON之前, 定时器输入置OFF</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p>		
1ms 定时器	TMHH (BCD)	540	 N: 定时器号 S: 设定值	<p>TMHHX(552)/TMHHX(552) 定时器以 1ms 为单位作减量计时。对于 BCD, 设定值 (SV) 的设定范围为 0 ~ 9.999s, 对于二进制 (十进制或十六进制), 设定范围为 0 ~ 65.535s。</p> <p>TMHH(540) 和 TMHHX(552) 的时序图与上述 TIMH(015) 的相同。</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p> <p>在完成标志置ON之前, 定时器输入置OFF</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p>	输出	需要
	TMHHX (BCD)	552	 N: 定时器号 S: 设定值	<p>TMHH(540) 和 TMHHX(552) 的时序图与上述 TIMH(015) 的相同。</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p> <p>在完成标志置ON之前, 定时器输入置OFF</p> <p>定时器输入 ON OFF</p> <p>设定值(SV)</p> <p>定时器当前值(PV) 0</p> <p>完成标志 ON OFF</p>		

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	 <p>N: 定时器号 S: 设定值</p>	TIMU(541)/TIMUX(556) 定时器以 0.1ms 为单位作减量计时。对于 BCD, 设定值(SV)的设定范围为 0 ~ 0.9999s, 对于二进制(十进制或十六进制), 设定范围为 0 ~ 6.5535s。 定时器输入  定时器当前值(PV) ₀ 完成标志	输出	需要
	TIMUX (BIN)	556	 <p>N: 定时器号 S: 设定值</p>	在完成标志置ON之前, 定时器输入置OFF 定时器输入  定时器当前值(PV) ₀ 完成标志		
0.01ms 定时器	TMUH (BCD)	544	 <p>N: 定时器号 S: 设定值</p>	TMUH(544)/TMUHX(557) 定时器以 0.01ms 为单位作减量计时。对于 BCD, 设定值(SV)的设定范围为 0 ~ 0.09999s, 对于二进制(十进制或十六进制), 设定范围为 0 ~ 0.65535s。 定时器输入  定时器当前值(PV) ₀ 完成标志	输出	需要
	TIMUX (BIN)	556	 <p>N: 定时器号 S: 设定值</p>	在完成标志置ON之前, 定时器输入置OFF 定时器输入  定时器当前值(PV) ₀ 完成标志		
累加定时器	TTIM (BCD)	087	 <p>N: 定时器号 S: 设定值</p>	TTIM(087)/TTIMX(555)定时器以0.1s为单位作增量计时。对于 BCD, 设定值(SV)的设定范围为0 ~ 999.9s, 对于二进制(十进制或十六进制), 设定范围为0 ~ 6.553.5s。 定时器输入  定时器当前值(PV) 完成标志	输出	需要
	TTIMX (二进制)	555	 <p>N: 定时器号 S: 设定值</p>	恢复计时 当前值(PV)保持 0 完成标志  完成标志 复位输入		

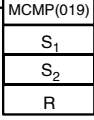
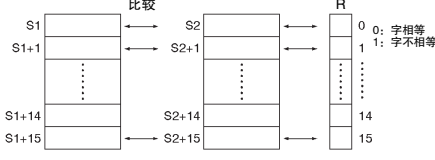
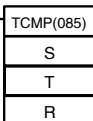
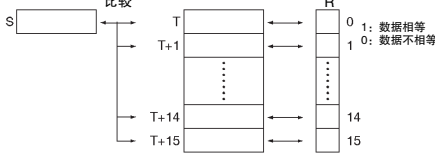
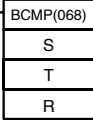
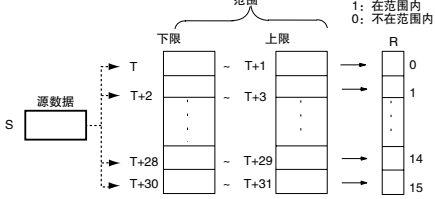
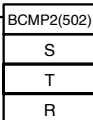
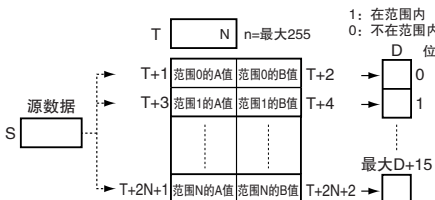
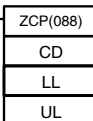
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
长定时器	TIML (BCD)	542	 <p>D1: 完成标志 D2: PV字 S: SV字</p>	<p>TIML(542)/TIMLX(553)定时器以0.1s为单位作减量计时。对于BCD, 定时器最高可设定到115天左右, 对二进制(十进制或十六进制), 最高可设定到49,710天左右。</p>  <p>定时器输入 ON OFF 设定值(SV) 定时器当前值(PV) S+1, S D2+1, D2 0 完成标志 (D1的位00) ON OFF</p>	输出	需要
	TIMLX (二进制)	553	 <p>D1: 完成标志 D2: PV字 S: SV字</p>			
多路输出定时器	MTIM (BCD)	543	 <p>D1: 完成标志 D2: PV字 S: SV首字</p>	<p>MTIM(543)/MTIMX(554)定时器以0.1s为单位作增量计时, 此定时器有8个独立的SV和完成标志。对于BCD, 设定值(SV)的设定范围为0~999.9s, 对于二进制(十进制或十六进制), 设定范围为0~6,553.5s。</p>  <p>定时器当前值(PV) D2 定时器SV S S+1 S+2 ~ S+7 D1位 0 1 2 ~ 7 定时器输入 ON OFF SV 7 ~ SV 2 定时器PV(D2) SV 1 SV 0 0 完成标志(D1) 位 7 ~ 位 2 位 1 位 0</p>	输出	需要
	MTIMX (二进制)	554	 <p>D1: 完成标志 D2: PV字 S: SV首字</p>			
定时器复位	TRSET	549	 <p>N: 定时器号</p>	使指定的定时器复位。	输出	需要
计数器	CNT (BCD)	---	 <p>对输入计数 复位输入 N: 计数器号 S: 设定值</p>	<p>CNT/CNTX(546)操作一个减量计数器。对于BCD, 设定值(SV)的设定范围为0~9,999, 对于二进制(十进制或十六进制), 设定范围为0~65,535。</p>  <p>对输入计数 ON OFF 复位输入 ON OFF 设定值(SV) 计数器当前值(PV) 0 完成标志 ON OFF</p>	输出	需要
	CNTX (二进制)	546	 <p>对输入计数 复位输入 N: 计数器号 S: 设定值</p>			

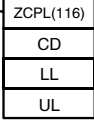


指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
可逆计数器	CNTR (BCD)	012	<p>N: 计数器号 S: 设定值</p>	<p>CNTR(012)/CNTRX(548)操作一个可逆计数器。</p>	输出	需要
	CNTRX (二进制)	548	<p>N: 计数器号 S: 设定值</p>	<p>CNTRX(548)操作一个可逆计数器。</p>	输出	需要
复位定时器 / 计数器	CNR @CNR (BCD)	545	<p>N1: 范围中的第一个编号 N2: 范围中的最后一个编号</p>	<p>CNR(545)/CNRX(547) 在定时器 / 计数器数值的指定范围内复位定时器 / 计数器。BCD 的设定值 (SV) 最大为 #9999 ; 二进制为 #FFFF。</p>	输出	需要
	CNRX @CNRX (二进制)	547	<p>N1: 范围中的第一个编号 N2: 范围中的最后一个编号</p>	<p>CNRX(547) 在定时器 / 计数器数值的指定范围内复位定时器 / 计数器。BCD 的设定值 (SV) 最大为 #9999 ; 二进制为 #FFFF。</p>	输出	需要

A-1-5 比较指令

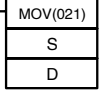
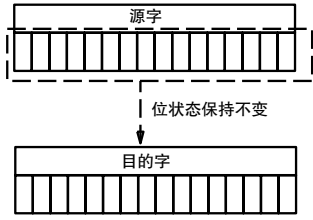
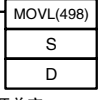
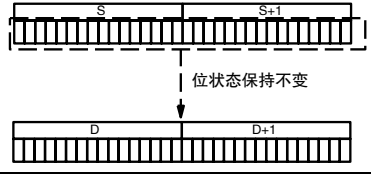
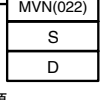
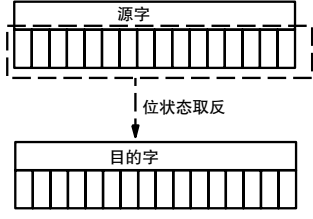

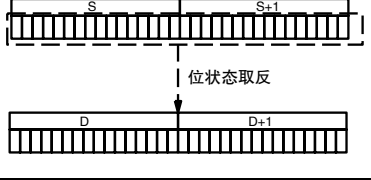
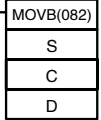
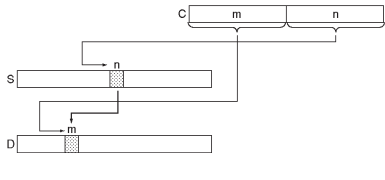
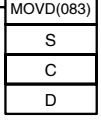
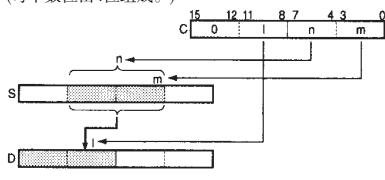
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
符号比较 (无符号)	LD、AND、 OR +=、<>、 <、<=、>、>=	300 (=) 305 (< >) 310 (<) 315 (<=) 320 (>) 325 (>=)	LD  AND  OR  S1: 比较数据1 S2: 比较数据2	符号比较指令(无符号)比较两个16位二进制数据的值(常数和/或指定字的内容),并在比较条件为真时生成一个ON执行条件。有三类符号比较指令: LD(载入)、AND和OR。 	LD: 逻辑开始 AND、 OR: 在梯级上连续	LD: 不需要 AND、 OR: 需要
符号比较 (双字, 无符号)	LD、AND、 OR +=、<>、 <、<=、>、>= +L	301 (=) 306 (< >) 311 (<) 316 (<=) 321 (>) 326 (>=)	S1: 比较数据 1 S2: 比较数据 2	符号比较指令(双字无符号)比较两个无符号32位二进制数的值(常数和/或指定的双字数据的内容),并在比较条件为真时生成一个ON执行条件。有三类符号比较指令: LD(载入)、AND和OR。	LD: 逻辑开始 AND、 OR: 在梯级上连续	LD: 不需要 AND、 OR: 需要
符号比较 (带符号)	LD、AND、 OR +=、<>、 <、<=、>、>= +S	302 (=) 307 (< >) 312 (<) 317 (<=) 322 (>) 327 (>=)	S1: 比较数据 1 S2: 比较数据 2	符号比较指令(带符号)比较两个带符号16位二进制(4位十六进制)数据的值(常数和/或指定字的内容),并在比较条件为真时生成一个ON执行条件。有三类符号比较指令: LD(载入)、AND和OR。	LD: 逻辑开始 AND、 OR: 在梯级上连续	LD: 不需要 AND、 OR: 需要
符号比较 (双字, 带符号)	LD、AND、 OR +=、<>、 <、<=、>、>= +SL	303 (=) 308 (< >) 313 (<) 318 (<=) 323 (>) 328 (>=)	S1: 比较数据 1 S2: 比较数据 2	符号比较指令(双字带符号)比较两个带符号32位二进制(8位十六进制)数的值(常数和/或指定的双字数据的内容),并在比较条件为真时生成一个ON执行条件。有三类符号比较指令: LD(载入)、AND和OR。	LD: 逻辑开始 AND、 OR: 在梯级上连续	LD: 不需要 AND、 OR: 需要

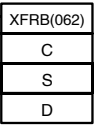
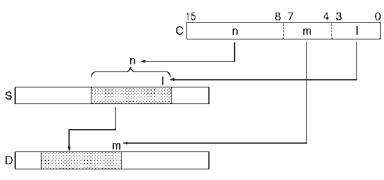
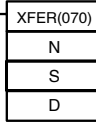
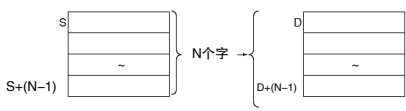
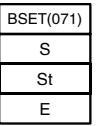
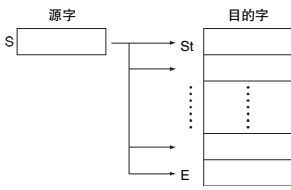

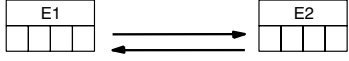

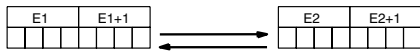
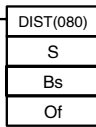
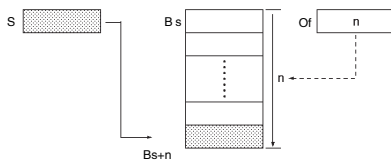
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
时间比较	LD, AND, OR + = DT, <> DT, < DT, <= DT, > DT, >= DT	341 (= DT) 342 (<> DT) 343 (< DT) 344 (<= DT) 345 (> DT) 346 (>= DT)	LD(载入): AND: OR: C: 控制字 S1: 当前时间的首字 S2: 比较时间的首字	时间比较指令比较两个 BCD 时间值, 并在比较条件为真时生成一个 ON 执行条件。 共有三类时间比较指令: LD(载入)、AND 及 OR。在比较中可将时间值(年、月、日、时、分及秒)屏蔽/不屏蔽, 因而可轻松创建日历定时器功能。	LD: 逻辑开始 AND, OR: 在梯级上连续	LD: 不需要 AND, OR: 需要
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(常数和/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	需要
双字无符号比较	CMPL	060	 S1: 比较数据1 S2: 比较数据2	比较两个双字无符号二进制值(常数和/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	需要
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(常数和/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	需要
带符号双字二进制比较	CPSL	115	 S1: 比较数据1 S2: 比较数据2	比较两个双字带符号二进制值(常数和/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	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多路比较	MCMP @MCMP	019	 <p>S1: 数据集1的首字 S2: 数据集2的首字 R: 结果字</p>	<p>将16个连续的字与另16个连续的字进行比较,并在这些字的内容不相等时,将结果字中的相应位置ON。</p>  <p>比较</p> <p>0: 字相等 1: 字不相等</p>	输出	需要
表比较	TCMP @TCMP	085	 <p>S: 源数据 T: 表首字 R: 结果字</p>	<p>将源数据与16个字的内容进行比较,并在内容相同时将结果字中的相应位置ON。</p>  <p>比较</p> <p>0: 数据相等 1: 数据不相等</p>	输出	需要
无符号块比较	BCMP @BCMP	068	 <p>S: 源数据 T: 表首字 R: 结果字</p>	<p>比较源数据与16个范围(由16个下限和16个上限定义),并当源数据在范围内时将结果字中的相应位置ON。</p>  <p>范围</p> <p>1: 在范围内 0: 不在范围内</p>	输出	需要
扩展块比较	BCMP2 @BCMP2	502	 <p>S: 源数据 T: 块首字 R: 结果字</p>	<p>可将源数据与最多256个范围(由上、下限定义)作比较,并当源数据处于其中某个范围内时,使结果字中的相应位置ON。</p>  <p>注: A可以小于B、等于B或大于B。</p>	输出	需要
区域范围比较	ZCP	088	 <p>CD: 比较数据(1个字) LL: 范围下限 UL: 范围上限</p>	<p>将CD中的16位无符号二进制值(字内容或常数)与由LL和UL定义的范围进行比较,并将结果输出到辅助区中的算术标志。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双字区域范围比较	ZCPL	116	 <p>CD: 比较数据(2个字) LL: 下限首字 UL: 上限首字</p>	将 CD 和 CD+1 中的 32 位无符号二进制值 (字内容或常数) 与由 LL 和 UL 定义的范围进行比较, 并将结果输出到辅助区中的算术标志。	输出	需要
带符号区域范围比较	ZCPS	117	 <p>CD: 比较数据(1个字) LL: 范围下限 UL: 范围上限</p>	将 CD 中的 16 位带符号二进制值 (字内容或常数) 与由 LL 和 UL 定义的范围进行比较, 并将结果输出到辅助区中的算术标志。	输出	需要
带符号双字区域范围比较	ZCPSL	118	 <p>CD: 比较数据(2个字) LL: 下限首字 UL: 上限首字</p>	将 CD 和 CD+1 中的 32 位带符号二进制值 (字内容或常数) 与由 LL 和 UL 定义的范围进行比较, 并将结果输出到辅助区中的算术标志。	输出	需要

A-1-6 数据传送指令

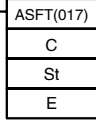
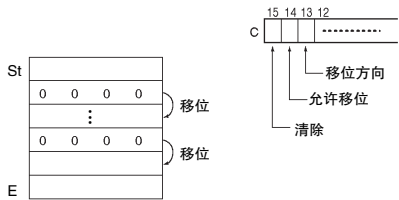
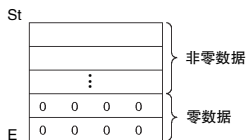
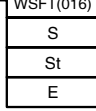
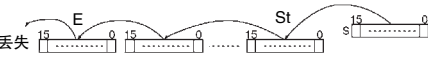

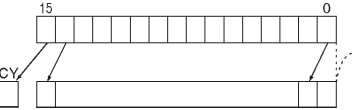

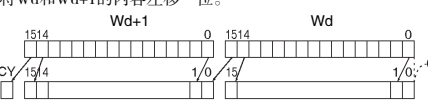

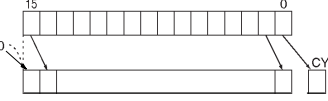

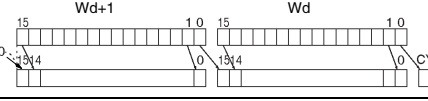

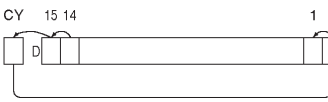

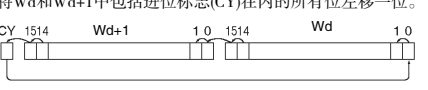
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
传送	MOV @MOV !MOV	021	 <p>S: 源 D: 目的</p>	<p>将一个字的数据传送到指定字。</p>  <p>位状态保持不变</p>	输出	需要
双字传送	MOVL @MOVL	498	 <p>S: 源首字 D: 目的首字</p>	<p>将两个字的数据传送到指定字。</p>  <p>位状态保持不变</p>	输出	需要
传送反	MVN @MVN	022	 <p>S: 源 D: 目的</p>	<p>将一个字的数据的补码传送到指定字。</p>  <p>位状态取反</p>	输出	需要
双字传送反	MVNL @MVNL	499	 <p>S: 源首字 D: 目的首字</p>	<p>将两个字的数据的补码传送到指定字。</p>  <p>位状态取反</p>	输出	需要
位传送	MOVB @MOVB	082	 <p>S: 源字或数据 C: 控制字 D: 目的字</p>	<p>传送指定的位。</p> 	输出	需要
数位传送	MOVD @MOVD	083	 <p>S: 源字或数据 C: 控制字 D: 目的字</p>	<p>传送指定的一个或多个数位。 (每个数位由4位组成。)</p> 	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
多位传送	XFRB @XFRB	062	 <p>C: 控制字 S: 源首字 D: 目的首字</p>	<p>传送指定数目的连续位。</p> 	输出	需要
块传送	XFER @XFER	070	 <p>N: 字数 S: 源首字 D: 目的首字</p>	<p>传送指定数目的连续字。</p> 	输出	需要
块设置	BSET @BSET	071	 <p>S: 源字 St: 起始字 E: 结束字</p>	<p>将同一个字的内容复制到几个连续字中。</p> 	输出	需要
数据交换	XCHG @XCHG	073	 <p>E1: 第一个交换字 E2: 第二个交换字</p>	<p>交换两个指定字的内容。</p> 	输出	需要
双字数据交换	XCGL @XCGL	562	 <p>E1: 第一个交换字 E2: 第二个交换字</p>	<p>将一对连续字的内容与另一对连续字的内容进行交换。</p> 	输出	需要
单字分配	DIST @DIST	080	 <p>S: 源字 Bs: 目的基址 Of: 偏移量</p>	<p>通过添加偏移值到基地址计算出目的字，并向其传送源字。</p> 	输出	需要

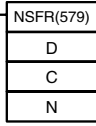
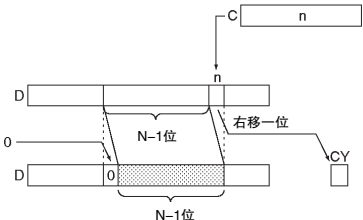

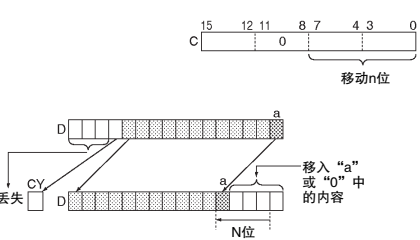

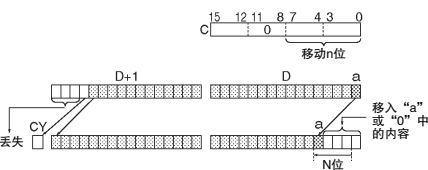

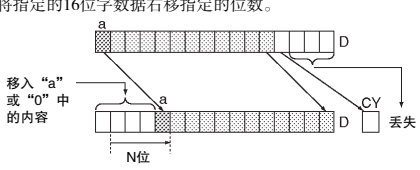

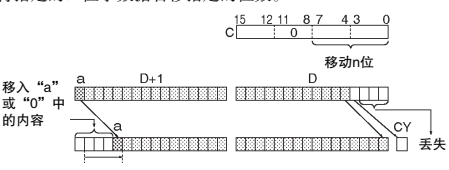
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
数据收集	COLL @COLL	081	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> COLL(081) Bs Of D </div> <p>Bs: 源基地址 Of: 偏移量 D: 目的字</p>	<p>通过添加偏移值到基地址计算出源字，并将其传送到目的字。</p>	输出	需要
传送至寄存器	MOVR @MOVR	560	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> MOVR(560) S D </div> <p>S: 源 (所需的字或位) D: 目的</p>	<p>在指定的变址寄存器中设定指定字、位或定时器/计数器完成标志的内部I/O存储器地址。(用MOVRW(561)指令将定时器/计数器PV值的内部I/O存储器地址存放到变址寄存器中。)</p>	输出	需要
传送定时器 / 计数器的当前值 (PV) 至寄存器	MOVRW @MOVRW	561	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> MOVRW(561) S D </div> <p>S: 源 (所需的TC号) D: 目的 (变址寄存器)</p>	<p>在指定的变址寄存器中设定指定定时器或计数器的PV值的内部I/O存储器地址。(用MOVR(560)指令将字、位或定时器/计数器完成标志的内部I/O存储器地址存放到变址寄存器中。)</p>	输出	需要

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	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SFT(010) St E </div> <p>St: 起始字 E: 结束字</p>	<p>操作一个移位寄存器。</p>	输出	需要
可逆移位寄存器	SFTR @SFTR	084	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SFTR(084) C St E </div> <p>C: 控制字 St: 起始字 E: 结束字</p>	<p>创建一个可实现数据左移或右移的移位寄存器。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
异步移位寄存器	ASFT @ASFT	017	 <p>C: 控制字 St: 起始字 E: 结束字</p>	<p>将指定字范围内的所有非零字数据朝S或E方向移动，以替换0000 Hex字数据。</p>  	输出	需要
字移位	WSFT @WSFT	016	 <p>S: 源字 St: 起始字 E: 结束字</p>	<p>将St和E之间的数据以字为单位移位。</p> 	输出	需要
算术左移	ASL @ASL	025	 <p>Wd: 字</p>	<p>将Wd的内容左移一位。</p> 	输出	需要
双字左移	ASLL @ASLL	570	 <p>Wd: 字</p>	<p>将Wd和Wd+1的内容左移一位。</p> 	输出	需要
算术右移	ASR @ASR	026	 <p>Wd: 字</p>	<p>将Wd的内容右移一位。</p> 	输出	需要
双字右移	ASRL @ASRL	571	 <p>Wd: 字</p>	<p>将Wd和Wd+1的内容右移一位。</p> 	输出	需要
循环左移	ROL @ROL	027	 <p>Wd: 字</p>	<p>将Wd中的包括进位标志(CY)在内的所有位左移一位。</p> 	输出	需要
双字循环左移	ROLL @ROLL	572	 <p>Wd: 字</p>	<p>将Wd和Wd+1中包括进位标志(CY)在内的所有位左移一位。</p> 	输出	需要

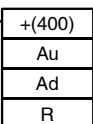

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无进位循环左移	RLNC @RLNC	574	 <p>Wd: 字</p>	将Wd中不包括进位标志(CY)在内的所有位左移一位。	输出	需要
无进位双字循环左移	RLNL @RLNL	576	 <p>Wd: 字</p>	将Wd和Wd+1中不包括进位标志(CY)在内的所有位左移一位。	输出	需要
循环右移	ROR @ROR	028	 <p>Wd: 字</p>	将Wd中的包括进位标志(CY)在内的所有位右移一位。	输出	需要
双字循环右移	RORL @RORL	573	 <p>Wd: 字</p>	将Wd和Wd+1中包括进位标志(CY)在内的所有位右移一位。	输出	需要
无进位循环右移	RRNC @RRNC	575	 <p>Wd: 字</p>	将Wd中不包括进位标志(CY)在内的所有位右移一位。 Wd最右位的内容将移至最左位和进位标志(CY)。	输出	需要
无进位双字循环右移	RRNL @RRNL	577	 <p>Wd: 字</p>	将Wd和Wd+1中不包括进位标志(CY)在内的所有位右移一位。 Wd+1最右位的内容将移至Wd的最左位和进位标志(CY)。	输出	需要
一个数位左移	SLD @SLD	074	 <p>St: 起始字 E: 结束字</p>	将数据左移一个数位(4位)。	输出	需要
一个数位右移	SRD @SRD	075	 <p>St: 起始字 E: 结束字</p>	将数据右移一个数位(4位)。	输出	需要
N 位数据左移	NSFL @NSFL	578	 <p>D: 移位起始字 C: 起始位 N: 移位数据长度</p>	将指定的位数左移。	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
N 位数据右移	NSFR @NSFR	579	 <p>D: 移位起始字 C: 起始位 N: 移位数据长度</p>	<p>将指定的位数右移。</p> 	输出	需要
左移 N 位	NASL @NASL	580	 <p>D: 移位字 C: 控制字</p>	<p>将指定的16位字数据左移指定的位数。</p> 	输出	需要
双字左移 N 位	NSLL @NSLL	582	 <p>D: 移位字 C: 控制字</p>	<p>将指定的32位字数据左移指定的位数。</p> 	输出	需要
右移 N 位	NASR @NASR	581	 <p>D: 移位字 C: 控制字</p>	<p>将指定的16位字数据右移指定的位数。</p> 	输出	需要
双字右移 N 位	NSRL @NSRL	583	 <p>D: 移位字 C: 控制字</p>	<p>将指定的32位字数据右移指定的位数。</p> 	输出	需要

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	 Wd: 字	将指定字的4位数十六进制内容递增1。 	输出	需要
双字二进制递增	++L @++L	591	 Wd: 字	将指定字的8位数十六进制内容递增1。 	输出	需要
二进制递减	-- @--	592	 Wd: 字	将指定字的4位数十六进制内容递减1。 	输出	需要
双字二进制递减	--L @--L	593	 Wd: 首字	将指定字的8位数十六进制内容递减1。 	输出	需要
BCD 递增	++B @++B	594	 Wd: 字	将指定字的4位BCD内容递增1。 	输出	需要
双字 BCD 递增	++BL @++BL	595	 Wd: 首字	将指定字的8位BCD内容递增1。 	输出	需要
BCD 递减	--B @--B	596	 Wd: 字	将指定字的4位BCD内容递减1。 	输出	需要
双字 BCD 递减	--BL @--BL	597	 Wd: 首字	将指定字的8位BCD内容递减1。 	输出	需要

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	+ @+	400	 Au: 被加数字 Ad: 加数字 R: 结果首字	4位数(单字)十六进制数据和/或常数相加。 	输出	需要

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	+L @+L	401	<p>Au: 被加数首字 Ad: 加数首字 R: 结果首字</p>	<p>8位数(双字)十六进制数据和/或常数相加。</p> <p>有进位时CY 将置ON</p>	输出	需要
有进位带符号二进制加	+C @+C	402	<p>Au: 被加数字 Ad: 加数字 R: 结果字</p>	<p>4位数(单字)十六进制数据和/或常数带进位标志(CY)相加。</p> <p>有进位时CY 将置ON</p>	输出	需要
有进位带符号双字二进制加	+CL @+CL	403	<p>Au: 被加数首字 Ad: 加数首字 R: 结果首字</p>	<p>8位数(双字)十六进制数据和/或常数带进位标志(CY)相加。</p> <p>有进位时CY 将置ON</p>	输出	需要
无进位BCD加	+B @+B	404	<p>Au: 被加数字 Ad: 加数字 R: 结果字</p>	<p>4位数(单字)BCD数据和/或常数相加。</p> <p>有进位时CY 将置ON</p>	输出	需要
无进位双字BCD加	+BL @+BL	405	<p>Au: 被加数首字 Ad: 加数首字 R: 结果首字</p>	<p>8位数(双字)BCD数据和/或常数相加。</p> <p>有进位时CY 将置ON</p>	输出	需要
有进位BCD加	+BC @+BC	406	<p>Au: 被加数字 Ad: 加数字 R: 结果字</p>	<p>4位数(单字)BCD数据和/或常数带进位标志(CY)相加。</p> <p>有进位时CY 将置ON</p>	输出	需要

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	+BCL @+BCL	407	$\begin{array}{ c } \hline +BCL(407) \\ \hline Au \\ \hline Ad \\ \hline R \\ \hline \end{array}$ <p>Au: 被加数首字 Ad: 加数首字 R: 结果首字</p>	<p>8位数(双字)BCD数据和/或常数带进位标志(CY)相加。</p> $\begin{array}{r} \begin{array}{ c c } \hline Au+1 & Au \\ \hline \end{array} \quad (BCD) \\ \begin{array}{ c c } \hline Ad+1 & Ad \\ \hline \end{array} \quad (BCD) \\ + \\ \hline \begin{array}{ c } \hline CY \\ \hline \end{array} \\ \hline \begin{array}{ c c c } \hline CY & R+1 & R \\ \hline \end{array} \quad (BCD) \end{array}$ <p>有进位时CY 将置ON</p>	输出	需要
无借位带符号二进制减	- @-	410	$\begin{array}{ c } \hline -(410) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ <p>Mi: 被减数字 Su: 减数字</p>	<p>4位数(单字)十六进制数据和/或常数相减。</p> $\begin{array}{r} \begin{array}{ c } \hline Mi \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ - \\ \begin{array}{ c } \hline Su \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ \hline \begin{array}{ c c } \hline CY & R \\ \hline \end{array} \quad (\text{带符号二进制数}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
无借位带符号双字二进制减	-L @-L	411	$\begin{array}{ c } \hline -L(411) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ <p>Mi: 被减数字 Su: 减数字 R: 结果字</p>	<p>8位数(双字)十六进制数据和/或常数相减。</p> $\begin{array}{r} \begin{array}{ c c } \hline Mi+1 & Mi \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ - \\ \begin{array}{ c c } \hline Su+1 & Su \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ \hline \begin{array}{ c c c } \hline CY & R+1 & R \\ \hline \end{array} \quad (\text{带符号二进制数}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
有借位带符号二进制减	-C @-C	412	$\begin{array}{ c } \hline -C(412) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ <p>Mi: 被减数字 Su: 减数字 R: 结果字</p>	<p>4位数(单字)十六进制数据和/或常数带借位标志(CY)相减。</p> $\begin{array}{r} \begin{array}{ c } \hline Mi \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ - \\ \begin{array}{ c } \hline Su \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ \hline \begin{array}{ c } \hline CY \\ \hline \end{array} \\ \hline \begin{array}{ c c } \hline CY & R \\ \hline \end{array} \quad (\text{带符号二进制数}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
有借位带符号双字二进制减	-CL @--CL	413	$\begin{array}{ c } \hline -CL(413) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ <p>Mi: 被减数字 Su: 减数字 R: 结果字</p>	<p>8位数(双字)十六进制数据和/或常数带借位标志(CY)相减。</p> $\begin{array}{r} \begin{array}{ c c } \hline Mi+1 & Mi \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ \begin{array}{ c c } \hline Su+1 & Su \\ \hline \end{array} \quad (\text{带符号二进制数}) \\ - \\ \hline \begin{array}{ c } \hline CY \\ \hline \end{array} \\ \hline \begin{array}{ c c c } \hline CY & R+1 & R \\ \hline \end{array} \quad (\text{带符号二进制数}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
无借位 BCD 减	-B @-B	414	$\begin{array}{ c } \hline -B(414) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ <p>Mi: 被减数字 Su: 减数字 R: 结果字</p>	<p>4位数(单字)BCD数据和/或常数相减。</p> $\begin{array}{r} \begin{array}{ c } \hline Mi \\ \hline \end{array} \quad (BCD) \\ - \\ \begin{array}{ c } \hline Su \\ \hline \end{array} \quad (BCD) \\ \hline \begin{array}{ c c } \hline CY & R \\ \hline \end{array} \quad (BCD) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要




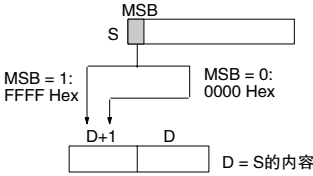
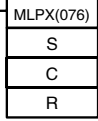
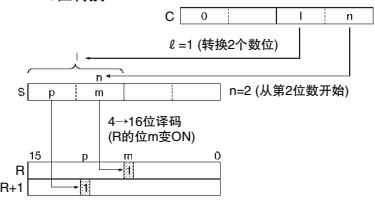
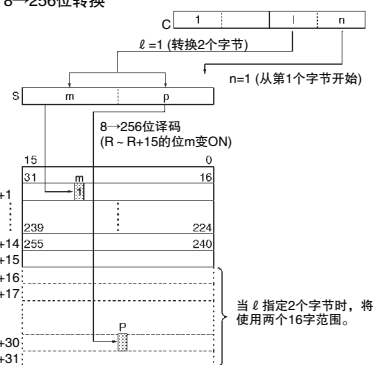
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无借位双字 BCD 减	-BL @-BL	415	$\begin{array}{ c } \hline \text{---} \\ \hline \text{-BL(415)} \\ \hline \text{Mi} \\ \hline \text{Su} \\ \hline \text{R} \\ \hline \end{array}$ <p>Mi: 被减数首字 Su: 减数首字 R: 结果首字</p>	8位数(双字)BCD数据和/或常数相减。 $\begin{array}{r} \text{Mi +1} \quad \text{Mi} \quad (\text{BCD}) \\ - \quad \text{Su+1} \quad \text{Su} \quad (\text{BCD}) \\ \hline \text{CY} \quad \text{R+1} \quad \text{R} \quad (\text{BCD}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
有借位 BCD 减	-BC @-BC	416	$\begin{array}{ c } \hline \text{---} \\ \hline \text{-BC(416)} \\ \hline \text{Mi} \\ \hline \text{Su} \\ \hline \text{R} \\ \hline \end{array}$ <p>Mi: 被减数字 Su: 减数字 R: 结果字</p>	4位数(单字)BCD数据和/或常数带借位标志(CY)相减。 $\begin{array}{r} \text{Mi} \quad (\text{BCD}) \\ - \quad \text{Su} \quad (\text{BCD}) \\ \hline \text{CY} \\ \hline \text{CY} \quad \text{R} \quad (\text{BCD}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
有借位双字 BCD 减	-BCL @-BCL	417	$\begin{array}{ c } \hline \text{---} \\ \hline \text{-BCL(417)} \\ \hline \text{Mi} \\ \hline \text{Su} \\ \hline \text{R} \\ \hline \end{array}$ <p>Mi: 被减数首字 Su: 减数首字 R: 结果首字</p>	8位数(双字)BCD数据和/或常数带借位标志(CY)相减。 $\begin{array}{r} \text{Mi +1} \quad \text{Mi} \quad (\text{BCD}) \\ - \quad \text{Su+1} \quad \text{Su} \quad (\text{BCD}) \\ \hline \text{CY} \\ \hline \text{CY} \quad \text{R+1} \quad \text{R} \quad (\text{BCD}) \end{array}$ <p>有借位时CY 将置ON</p>	输出	需要
带符号二进制乘	* @*	420	$\begin{array}{ c } \hline \text{---} \\ \hline \text{*(420)} \\ \hline \text{Md} \\ \hline \text{Mr} \\ \hline \text{R} \\ \hline \end{array}$ <p>Md: 被乘数字 Mr: 乘数字 R: 结果字</p>	4位数带符号十六进制数据和/或常数相乘。 $\begin{array}{r} \text{Md} \quad (\text{带符号二进制数}) \\ \times \quad \text{Mr} \quad (\text{带符号二进制数}) \\ \hline \text{R+1} \quad \text{R} \quad (\text{带符号二进制数}) \end{array}$	输出	需要
带符号双字二进制乘	*L @*L	421	$\begin{array}{ c } \hline \text{---} \\ \hline \text{*L(421)} \\ \hline \text{Md} \\ \hline \text{Mr} \\ \hline \text{R} \\ \hline \end{array}$ <p>Md: 被乘数首字 Mr: 乘数首字 R: 结果首字</p>	8位数带符号十六进制数据和/或常数相乘。 $\begin{array}{r} \text{Md +1} \quad \text{Md} \quad (\text{带符号二进制数}) \\ \times \quad \text{Mr +1} \quad \text{Mr} \quad (\text{带符号二进制数}) \\ \hline \text{R+3} \quad \text{R+2} \quad \text{R+1} \quad \text{R} \quad (\text{带符号二进制数}) \end{array}$	输出	需要
无符号二进制乘	*U @*U	422	$\begin{array}{ c } \hline \text{---} \\ \hline \text{*U(422)} \\ \hline \text{Md} \\ \hline \text{Mr} \\ \hline \text{R} \\ \hline \end{array}$ <p>Md: 被乘数字 Mr: 乘数字 R: 结果字</p>	4位数无符号十六进制数据和/或常数相乘。 $\begin{array}{r} \text{Md} \quad (\text{无符号二进制数}) \\ \times \quad \text{Mr} \quad (\text{无符号二进制数}) \\ \hline \text{R+1} \quad \text{R} \quad (\text{无符号二进制数}) \end{array}$	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无符号双字二进制乘	*UL @*UL	423	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *UL(423) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘数首字 Mr: 乘数首字 R: 结果首字</p>	8位数无符号十六进制数据和/或常数相乘。 $\begin{array}{r} \text{Md} + 1 \quad \text{Md} \quad (\text{无符号二进制数}) \\ \times \quad \text{Mr} + 1 \quad \text{Mr} \quad (\text{无符号二进制数}) \\ \hline \text{R} + 3 \quad \text{R} + 2 \quad \text{R} + 1 \quad \text{R} \quad (\text{无符号二进制数}) \end{array}$	输出	需要
BCD 乘	*B @*B	424	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *B(424) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘数 Mr: 乘数 R: 结果字</p>	4位数(单字)BCD数据和/或常数相乘。 $\begin{array}{r} \text{Md} \quad (\text{BCD}) \\ \times \quad \text{Mr} \quad (\text{BCD}) \\ \hline \text{R} + 1 \quad \text{R} \quad (\text{BCD}) \end{array}$	输出	需要
双字 BCD 乘	*BL @*BL	425	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *BL(425) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘数首字 Mr: 乘数首字 R: 结果首字</p>	8位数(双字)BCD数据和/或常数相乘。 $\begin{array}{r} \text{Md} + 1 \quad \text{Md} \quad (\text{BCD}) \\ \times \quad \text{Mr} + 1 \quad \text{Mr} \quad (\text{BCD}) \\ \hline \text{R} + 3 \quad \text{R} + 2 \quad \text{R} + 1 \quad \text{R} \quad (\text{BCD}) \end{array}$	输出	需要
带符号二进制除	/ @/	430	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /(430) <hr/> Dd <hr/> Dr <hr/> R </div> <p>Dd: 被除数 Dr: 除数 R: 结果字</p>	4位数(单字)带符号十六进制数据和/或常数相除。 $\begin{array}{r} \text{Dd} \quad (\text{带符号二进制数}) \\ \div \quad \text{Dr} \quad (\text{带符号二进制数}) \\ \hline \text{R} + 1 \quad \text{R} \quad (\text{带符号二进制数}) \end{array}$ <p style="text-align: center;">余数 商</p>	输出	需要
带符号双字二进制除	/L @/L	431	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /L(431) <hr/> Dd <hr/> Dr <hr/> R </div> <p>Dd: 被除数首字 Dr: 除数首字 R: 结果首字</p>	8位数(双字)带符号十六进制数据和/或常数相除。 $\begin{array}{r} \text{Dd} + 1 \quad \text{Dd} \quad (\text{带符号二进制数}) \\ \div \quad \text{Dr} + 1 \quad \text{Dr} \quad (\text{带符号二进制数}) \\ \hline \text{R} + 3 \quad \text{R} + 2 \quad \text{R} + 1 \quad \text{R} \quad (\text{带符号二进制数}) \end{array}$ <p style="text-align: center;">余数 商</p>	输出	需要
无符号二进制除	/U @/U	432	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /U(432) <hr/> Dd <hr/> Dr <hr/> R </div> <p>Dd: 被除数 Dr: 除数 R: 结果字</p>	4位数(单字)无符号十六进制数据和/或常数相除。 $\begin{array}{r} \text{Dd} \quad (\text{无符号二进制数}) \\ \div \quad \text{Dr} \quad (\text{无符号二进制数}) \\ \hline \text{R} + 1 \quad \text{R} \quad (\text{无符号二进制数}) \end{array}$ <p style="text-align: center;">余数 商</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
无符号双字二进制除	/UL @/UL	433	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /UL(433) Dd Dr R </div> <p>Dd: 被除数首字 Dr: 除数首字 R: 结果首字</p>	8位数(双字)无符号十六进制数据和/或常数相除。 $\begin{array}{r} \boxed{Dd+1} \quad \boxed{Dd} \text{ (无符号二进制数)} \\ \div \quad \boxed{Dr+1} \quad \boxed{Dr} \text{ (无符号二进制数)} \\ \hline \boxed{R+3} \quad \boxed{R+2} \quad \boxed{R+1} \quad \boxed{R} \text{ (无符号二进制数)} \\ \text{余数} \qquad \qquad \qquad \text{商} \end{array}$	输出	需要
BCD 除	/B @/B	434	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /B(434) Dd Dr R </div> <p>Dd: 被除数字 Dr: 除数字 R: 结果字</p>	4位数(单字)BCD数据和/或常数相除。 $\begin{array}{r} \boxed{Dd} \text{ (BCD)} \\ \div \quad \boxed{Dr} \text{ (BCD)} \\ \hline \boxed{R+1} \quad \boxed{R} \text{ (BCD)} \\ \text{余数} \qquad \qquad \qquad \text{商} \end{array}$	输出	需要
双字 BCD 除	/BL @/BL	435	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /BL(435) Dd Dr R </div> <p>Dd: 被除数首字 Dr: 除数首字 R: 结果首字</p>	8位数(双字)BCD数据和/或常数相除。 $\begin{array}{r} \boxed{Dd+1} \quad \boxed{Dd} \text{ (BCD)} \\ \div \quad \boxed{Dr+1} \quad \boxed{Dr} \text{ (BCD)} \\ \hline \boxed{R+3} \quad \boxed{R+2} \quad \boxed{R+1} \quad \boxed{R} \text{ (BCD)} \\ \text{余数} \qquad \qquad \qquad \text{商} \end{array}$	输出	需要

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	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BIN(023) S R </div> <p>S: 源字 R: 结果字</p>	将BCD数据转换成二进制数据。 $S \quad \boxed{\text{(BCD)}} \longrightarrow R \quad \boxed{\text{(BIN)}}$	输出	需要
双字 BCD → 双字二进制	BINL @BINL	058	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BINL(058) S R </div> <p>S: 源首字 R: 结果首字</p>	将8位BCD数据转换成8位十六进制数据(32位二进制数据)。 $\begin{array}{r} S \quad \boxed{\text{(BCD)}} \longrightarrow R \quad \boxed{\text{(BIN)}} \\ S+1 \quad \boxed{\text{(BCD)}} \longrightarrow R+1 \quad \boxed{\text{(BIN)}} \end{array}$	输出	需要
二进制 → BCD	BCD @BCD	024	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BCD(024) S R </div> <p>S: 源字 R: 结果字</p>	将一个字的二进制数据转换成一个字的BCD数据。 $S \quad \boxed{\text{(BIN)}} \longrightarrow R \quad \boxed{\text{(BCD)}}$	输出	需要
双字二进制 → 双字 BCD	BCDL @BCDL	059	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BCDL(059) S R </div> <p>S: 源首字 R: 结果首字</p>	将8位十六进制(32位二进制)数据转换成8位BCD数据。 $\begin{array}{r} S \quad \boxed{\text{(BIN)}} \longrightarrow R \quad \boxed{\text{(BCD)}} \\ S+1 \quad \boxed{\text{(BIN)}} \longrightarrow R+1 \quad \boxed{\text{(BCD)}} \end{array}$	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
二进制求补	NEG @NEG	160	 <p>S: 源字 R: 结果字</p>	计算一个字的十六进制数据的2的补码。 $\overline{(S)} \xrightarrow{2\text{的补码(补码+1)}} (R)$	输出	需要
双字二进制求补	NEGL @NEGL	161	 <p>S: 源首字 R: 结果首字</p>	计算两个字的十六进制数据的2的补码。 $\overline{(S+1, S)} \xrightarrow{2\text{的补码(补码+1)}} (R+1, R)$	输出	需要
16位→32位带符号二进制	SIGN @SIGN	600	 <p>S: 源字 R: 结果字</p>	将16位带符号二进制值扩展为32位等值数。  <p>D = S 的内容</p>	输出	需要
数据译码	MLPX @MLPX	076	 <p>S: 源字 C: 控制字 R: 结果首字</p>	读源字中指定数位(或字节)的数值, 并将结果字(或16字范围)中的相应位置ON、将所有其它位置OFF。 4→16位转换  8→256位转换  <p>当 ℓ 指定2个字节时, 将使用两个16字范围。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
数据编码	DMPX @DMPX	077	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">DMPX(077)</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">S</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">R</div> <div style="border: 1px solid black; padding: 2px;">C</div> <p>S: 源首字 R: 结果字 C: 控制字</p>	<p>寻找源字范围(或16字范围)内的第一个或最后一个ON位的位置, 并将该值写入结果字的指定数位(或字节)中。</p> <p>16→4位转换</p> <p>256→8位转换</p>	输出	需要
ASCII 转换	ASC @ASC	086	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">ASC(086)</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">S</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Di</div> <div style="border: 1px solid black; padding: 2px;">D</div> <p>S: 源字 Di: 数位定义 D: 目的首字</p>	<p>把源字中的4位十六进制数位转换成等值的8位ASCII码。</p>	输出	需要
ASCII → 十六进制	HEX @HEX	162	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">HEX(162)</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">S</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">Di</div> <div style="border: 1px solid black; padding: 2px;">D</div> <p>S: 源首字 Di: 数位定义 D: 目的字</p>	<p>把源字中的最多4个字节的ASCII数据转换成等值的十六进制数位, 并将这些数位写入指定的目的字中。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
列→行	LINE @LINE	063	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> LINE(063) S N D </div> <p>S: 源首字 N: 位号 D: 目的字</p>	<p>将16字范围中的一列位(16个连续字中的相同位号)转换成目的字的16个位。</p>	输出	需要
行→列	COLM @COLM	064	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> COLM(064) S D N </div> <p>S: 源字 D: 目的首字 N: 位号</p>	<p>将源字的16个位转换成目的字的16字范围中的一列位(16个连续字中的相同位号)。</p>	输出	需要
带符号BCD→二进制	BINS @BINS	470	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BINS(470) C S D </div> <p>C: 控制字 S: 源字 D: 目的字</p>	<p>将一个字的带符号BCD数据转换成一个字的带符号二进制数据。</p>	输出	需要
带符号双字BCD→二进制	BISL @BISL	472	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BISL(472) C S D </div> <p>C: 控制字 S: 源首字 D: 目的首字</p>	<p>将带符号双字BCD数据转换为带符号双字二进制数据。</p>	输出	需要
带符号二进制→BCD	BCDS @BCDS	471	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BCDS(471) C S D </div> <p>C: 控制字 S: 源字 D: 目的字</p>	<p>将一个字的带符号二进制数据转换成一个字的带符号BCD数据。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
带符号双字二进制→BCD	BDSL @BDSL	473	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BDSL(473) <hr/> C <hr/> S <hr/> D </div> <p>C: 控制字 S: 源首字 D: 目的首字</p>	<p>将双字带符号二进制数据转换为双字带符号BCD数据。</p> 	输出	需要
格雷码转换	GRY	474	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> GRY (474) <hr/> C <hr/> S <hr/> D </div> <p>C: 控制字 S: 源字 D: 目的首字</p>	<p>将指定字中的格雷码数据转换成二进制、BCD，或以指定分辨率转换成角度(°)数据。</p>	输出	需要
格雷码→二进制转换	GRAY _BIN @GRAY_ BIN	478	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> GRAY_BIN(478) <hr/> S <hr/> D </div> <p>S: 源(格雷码) D: 目的(二进制)</p>	<p>将 S 指定的一个字的格雷码转换成一个字的二进制数，并将其输出到 D。</p>	输出	需要
双字格雷码→二进制转换	GRAY _BINL @GRAY_ BINL	479	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> GRAY_BINL(479) <hr/> S <hr/> D </div> <p>S: 源首字(格雷码) D: 目的首字(二进制)</p>	<p>将 S 指定的两个字的格雷码转换成两个字的二进制数，并将其输出到 D。</p>	输出	需要
二进制→格雷码转换	BIN _GRAY @BIN_ GRAY	480	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BIN_GRAY(480) <hr/> S <hr/> D </div> <p>S: 源字(二进制) D: 目的字(格雷码)</p>	<p>将 S 指定的一个字的二进制数转换成一个字的格雷码，并将其输出到 D。</p>	输出	需要
双字二进制→格雷码转换	BIN _GRAYL @BIN_ GRAYL	481	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BIN_GRAYL(481) <hr/> S <hr/> D </div> <p>S: 源首字(二进制) D: 目的首字(格雷码)</p>	<p>将 S 指定的两个字的二进制数转换成两个字的格雷码，并将其输出到 D。</p>	输出	需要
4 位数→ASCII	STR4 @STR4	601	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> STR4 <hr/> S <hr/> D </div> <p>S: 数值 D: ASCII文本</p>	<p>将 4 位数的十六进制数(#0000 ~ #FFFF)转换成 ASCII 数据(4 个字符)。</p> 	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件																																												
8 位数 → ASCII	STR8 @STR8	602	 <p>S: 数值 D: ASCII文本</p>	<p>将 8 位数十六进制数 (#0000 0000 ~ #FFFF FFFF) 转换成 ASCII 数据 (8 个字符)。</p> <p>15 12 11 8 7 4 3 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>S</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>S+1</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table> <p>十六进制: #12345678</p> <p>↓ ASCII</p> <p>15 8 7 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>D</td><td>31</td><td>32</td></tr> <tr><td>D+1</td><td>33</td><td>34</td></tr> <tr><td>D+2</td><td>35</td><td>36</td></tr> <tr><td>D+3</td><td>37</td><td>38</td></tr> </table>	S	5	6	7	8	S+1	1	2	3	4	D	31	32	D+1	33	34	D+2	35	36	D+3	37	38	输出	需要																						
S	5	6	7	8																																														
S+1	1	2	3	4																																														
D	31	32																																																
D+1	33	34																																																
D+2	35	36																																																
D+3	37	38																																																
16 位数 → ASCII	STR16 @STR16	603	 <p>S: 数值 D: ASCII文本</p>	<p>将 16 位数的十六进制数 (#0000 0000 0000 0000 ~ #FFFF FFFF FFFF FFFF) 转换成 ASCII 数据 (16 个字符)。</p> <p>15 12 11 8 7 4 3 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>S</td><td>C</td><td>D</td><td>E</td><td>F</td></tr> <tr><td>S+1</td><td>8</td><td>9</td><td>A</td><td>B</td></tr> <tr><td>S+2</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>S+3</td><td>0</td><td>1</td><td>2</td><td>3</td></tr> </table> <p>十六进制: #1234567890ABCDEF</p> <p>↓ ASCII</p> <p>15 8 7 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>D</td><td>30</td><td>31</td></tr> <tr><td>D+1</td><td>32</td><td>33</td></tr> <tr><td>D+2</td><td>34</td><td>35</td></tr> <tr><td>D+3</td><td>36</td><td>37</td></tr> <tr><td>D+4</td><td>38</td><td>39</td></tr> <tr><td>D+5</td><td>41</td><td>42</td></tr> <tr><td>D+6</td><td>43</td><td>44</td></tr> <tr><td>D+7</td><td>45</td><td>46</td></tr> </table>	S	C	D	E	F	S+1	8	9	A	B	S+2	4	5	6	7	S+3	0	1	2	3	D	30	31	D+1	32	33	D+2	34	35	D+3	36	37	D+4	38	39	D+5	41	42	D+6	43	44	D+7	45	46	输出	需要
S	C	D	E	F																																														
S+1	8	9	A	B																																														
S+2	4	5	6	7																																														
S+3	0	1	2	3																																														
D	30	31																																																
D+1	32	33																																																
D+2	34	35																																																
D+3	36	37																																																
D+4	38	39																																																
D+5	41	42																																																
D+6	43	44																																																
D+7	45	46																																																
ASCII → 4 位数	NUM4 @NUM4	604	 <p>S: ASCII文本 D: 数值</p>	<p>将 4 个字符的 ASCII 数据转换成 4 位数十六进制数。</p> <p>15 8 7 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>S</td><td>31</td><td>32</td></tr> <tr><td>S+1</td><td>33</td><td>34</td></tr> </table> <p>ASCII</p> <p>↓ 十六进制</p> <p>15 12 11 8 7 4 3 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>D</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table>	S	31	32	S+1	33	34	D	1	2	3	4	输出	需要																																	
S	31	32																																																
S+1	33	34																																																
D	1	2	3	4																																														
ASCII → 8 位数	NUM8 @NUM8	605	 <p>S: ASCII文本 D: 数值</p>	<p>将 8 个字符的 ASCII 数据转换成 8 位数十六进制数。</p> <p>15 8 7 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>S</td><td>31</td><td>32</td></tr> <tr><td>S+1</td><td>33</td><td>34</td></tr> <tr><td>S+2</td><td>35</td><td>36</td></tr> <tr><td>S+3</td><td>37</td><td>38</td></tr> </table> <p>ASCII</p> <p>↓ 十六进制</p> <p>15 12 11 8 7 4 3 0</p> <table border="1" style="margin-left: 20px;"> <tr><td>D</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>D+1</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> </table>	S	31	32	S+1	33	34	S+2	35	36	S+3	37	38	D	5	6	7	8	D+1	1	2	3	4	输出	需要																						
S	31	32																																																
S+1	33	34																																																
S+2	35	36																																																
S+3	37	38																																																
D	5	6	7	8																																														
D+1	1	2	3	4																																														


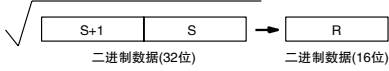

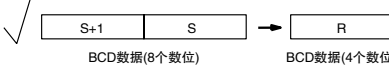
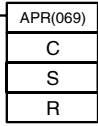
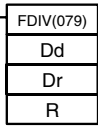
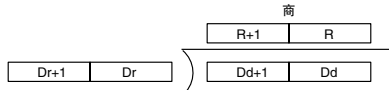
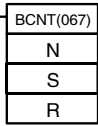
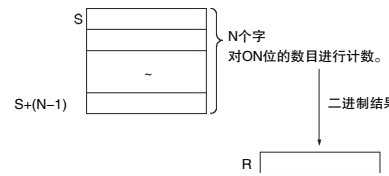
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件																																																																																										
ASCII → 16 位数	NUM16 @NUM16	606	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px;">STR16(606)</div> <div style="border: 1px solid black; padding: 2px;">S</div> <div style="border: 1px solid black; padding: 2px;">D</div> </div> <p>S: ASCII文本 D: 数值</p>	<p>将 16 个字符的 ASCII 数据转换成 16 位数的十六进制数。</p> <table style="margin-left: 20px;"> <tr> <td></td> <td style="text-align: center;">15</td> <td style="text-align: center;">8</td> <td style="text-align: center;">7</td> <td style="text-align: center;">0</td> </tr> <tr> <td>S</td> <td style="border: 1px solid black; padding: 2px;">30</td> <td style="border: 1px solid black; padding: 2px;">31</td> <td colspan="2"></td> </tr> <tr> <td>S+1</td> <td style="border: 1px solid black; padding: 2px;">32</td> <td style="border: 1px solid black; padding: 2px;">33</td> <td colspan="2"></td> </tr> <tr> <td>S+2</td> <td style="border: 1px solid black; padding: 2px;">34</td> <td style="border: 1px solid black; padding: 2px;">35</td> <td colspan="2"></td> </tr> <tr> <td>S+3</td> <td style="border: 1px solid black; padding: 2px;">36</td> <td style="border: 1px solid black; padding: 2px;">37</td> <td colspan="2"></td> </tr> <tr> <td>S+4</td> <td style="border: 1px solid black; padding: 2px;">38</td> <td style="border: 1px solid black; padding: 2px;">39</td> <td colspan="2"></td> </tr> <tr> <td>S+5</td> <td style="border: 1px solid black; padding: 2px;">41</td> <td style="border: 1px solid black; padding: 2px;">42</td> <td colspan="2"></td> </tr> <tr> <td>S+6</td> <td style="border: 1px solid black; padding: 2px;">43</td> <td style="border: 1px solid black; padding: 2px;">44</td> <td colspan="2"></td> </tr> <tr> <td>S+7</td> <td style="border: 1px solid black; padding: 2px;">45</td> <td style="border: 1px solid black; padding: 2px;">46</td> <td colspan="2"></td> </tr> </table> <p style="margin-left: 20px;">ASCII ↓ 十六进制</p> <table style="margin-left: 20px;"> <tr> <td></td> <td style="text-align: center;">15</td> <td style="text-align: center;">12</td> <td style="text-align: center;">11</td> <td style="text-align: center;">8</td> <td style="text-align: center;">7</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">0</td> </tr> <tr> <td>D</td> <td style="border: 1px solid black; padding: 2px;">C</td> <td style="border: 1px solid black; padding: 2px;">D</td> <td style="border: 1px solid black; padding: 2px;">E</td> <td style="border: 1px solid black; padding: 2px;">F</td> <td colspan="4"></td> </tr> <tr> <td>D+1</td> <td style="border: 1px solid black; padding: 2px;">8</td> <td style="border: 1px solid black; padding: 2px;">9</td> <td style="border: 1px solid black; padding: 2px;">A</td> <td style="border: 1px solid black; padding: 2px;">B</td> <td colspan="4"></td> </tr> <tr> <td>D+2</td> <td style="border: 1px solid black; padding: 2px;">4</td> <td style="border: 1px solid black; padding: 2px;">5</td> <td style="border: 1px solid black; padding: 2px;">6</td> <td style="border: 1px solid black; padding: 2px;">7</td> <td colspan="4"></td> </tr> <tr> <td>D+3</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">2</td> <td style="border: 1px solid black; padding: 2px;">3</td> <td colspan="4"></td> </tr> </table>		15	8	7	0	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				15	12	11	8	7	4	3	0	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	需要
	15	8	7	0																																																																																												
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																																																																																														
	15	12	11	8	7	4	3	0																																																																																								
D	C	D	E	F																																																																																												
D+1	8	9	A	B																																																																																												
D+2	4	5	6	7																																																																																												
D+3	0	1	2	3																																																																																												

A-1-11 逻辑指令

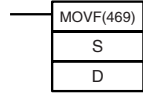
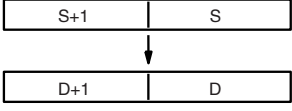
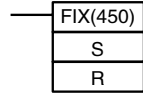
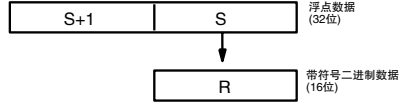
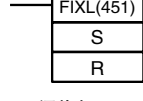
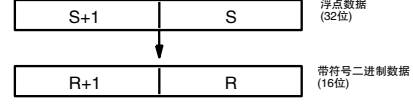
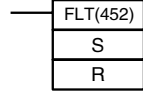
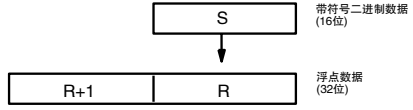
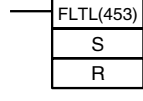
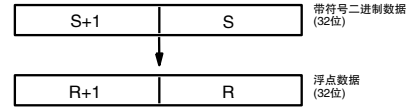
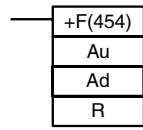
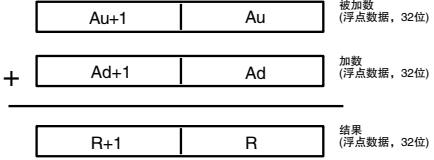
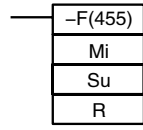

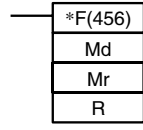
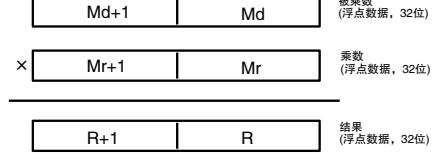
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件															
逻辑与	ANDW @ANDW	034	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px;">ANDW(034)</div> <div style="border: 1px solid black; padding: 2px;">I₁</div> <div style="border: 1px solid black; padding: 2px;">I₂</div> <div style="border: 1px solid black; padding: 2px;">R</div> </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	<p>对一个单字数据和/或常数中的相应位作逻辑与运算。</p> <p>$I_1, I_2 \rightarrow R$</p> <table style="margin-left: 20px;"> <tr> <td style="border: 1px solid black; padding: 2px;">I₁</td> <td style="border: 1px solid black; padding: 2px;">I₂</td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> </table>	I ₁	I ₂	R	1	1	1	1	0	0	0	1	0	0	0	0	输出	需要
I ₁	I ₂	R																			
1	1	1																			
1	0	0																			
0	1	0																			
0	0	0																			
双字逻辑与	ANDL @ANDL	610	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px;">ANDL(610)</div> <div style="border: 1px solid black; padding: 2px;">I₁</div> <div style="border: 1px solid black; padding: 2px;">I₂</div> <div style="border: 1px solid black; padding: 2px;">R</div> </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	<p>对一个双字数据和/或常数中的相应位作逻辑与运算。</p> <p>$(I_1, I_1+1), (I_2, I_2+1) \rightarrow (R, R+1)$</p> <table style="margin-left: 20px;"> <tr> <td style="border: 1px solid black; padding: 2px;">I_{1, I₁+1}</td> <td style="border: 1px solid black; padding: 2px;">I_{2, I₂+1}</td> <td style="border: 1px solid black; padding: 2px;">R, R+1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> </table>	I _{1, I₁+1}	I _{2, I₂+1}	R, R+1	1	1	1	1	0	0	0	1	0	0	0	0	输出	需要
I _{1, I₁+1}	I _{2, I₂+1}	R, R+1																			
1	1	1																			
1	0	0																			
0	1	0																			
0	0	0																			
逻辑或	ORW @ORW	035	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px;">ORW(035)</div> <div style="border: 1px solid black; padding: 2px;">I₁</div> <div style="border: 1px solid black; padding: 2px;">I₂</div> <div style="border: 1px solid black; padding: 2px;">R</div> </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	<p>对一个单字数据和/或常数中的相应位作逻辑或运算。</p> <p>$I_1 + I_2 \rightarrow R$</p> <table style="margin-left: 20px;"> <tr> <td style="border: 1px solid black; padding: 2px;">I₁</td> <td style="border: 1px solid black; padding: 2px;">I₂</td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> </table>	I ₁	I ₂	R	1	1	1	1	0	1	0	1	1	0	0	0	输出	需要
I ₁	I ₂	R																			
1	1	1																			
1	0	1																			
0	1	1																			
0	0	0																			
双字逻辑或	ORWL @ORWL	611	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px;">ORWL(611)</div> <div style="border: 1px solid black; padding: 2px;">I₁</div> <div style="border: 1px solid black; padding: 2px;">I₂</div> <div style="border: 1px solid black; padding: 2px;">R</div> </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	<p>对一个双字数据和/或常数中的相应位作逻辑或运算。</p> <p>$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$</p> <table style="margin-left: 20px;"> <tr> <td style="border: 1px solid black; padding: 2px;">I_{1, I₁+1}</td> <td style="border: 1px solid black; padding: 2px;">I_{2, I₂+1}</td> <td style="border: 1px solid black; padding: 2px;">R, R+1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">1</td> <td style="border: 1px solid black; padding: 2px;">1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> </tr> </table>	I _{1, I₁+1}	I _{2, I₂+1}	R, R+1	1	1	1	1	0	1	0	1	1	0	0	0	输出	需要
I _{1, I₁+1}	I _{2, I₂+1}	R, R+1																			
1	1	1																			
1	0	1																			
0	1	1																			
0	0	0																			

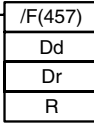
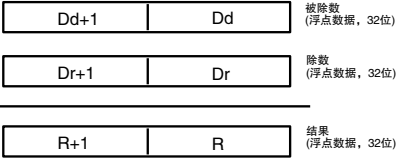

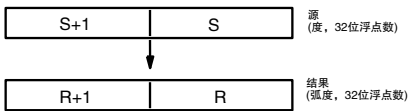

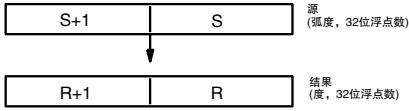

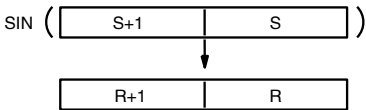

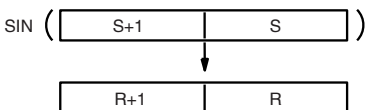



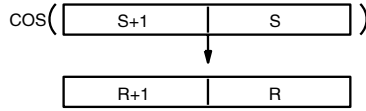

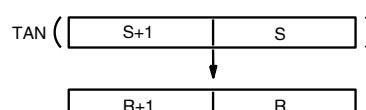
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件															
异或	XORW @XORW	036	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> XORW(036) I₁ I₂ R </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	对一个单字数据和/或常数中的相应位作逻辑异或运算。 $I_1 \cdot I_2 + I_1 \cdot \bar{I}_2 \rightarrow R$ <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>I₁</th> <th>I₂</th> <th>R</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	I ₁	I ₂	R	1	1	0	1	0	1	0	1	1	0	0	0	输出	需要
I ₁	I ₂	R																			
1	1	0																			
1	0	1																			
0	1	1																			
0	0	0																			
双字异或	XORL @XORL	612	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> XORL(612) I₁ I₂ R </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	对一个双字数据和/或常数中的相应位作逻辑异或运算。 $(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) + (\bar{I}_1 \cdot I_1 + 1) \cdot (\bar{I}_2 \cdot I_2 + 1) \rightarrow (R, R + 1)$ <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>I₁, I₁+1</th> <th>I₂, I₂+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1	1	1	0	1	0	1	0	1	1	0	0	0	输出	需要
I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1																			
1	1	0																			
1	0	1																			
0	1	1																			
0	0	0																			
异或非	XNRW @XNRW	037	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> XNRW(037) I₁ I₂ R </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	对单字数据和/或常数的相应位作逻辑异或非运算。 $I_1 \cdot I_2 + I_1 \cdot \bar{I}_2 \rightarrow R$ <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>I₁</th> <th>I₂</th> <th>R</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	I ₁	I ₂	R	1	1	1	1	0	0	0	1	0	0	0	1	输出	需要
I ₁	I ₂	R																			
1	1	1																			
1	0	0																			
0	1	0																			
0	0	1																			
双字异或非	XNRL @XNRL	613	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> XNRL(613) I₁ I₂ R </div> <p>I1: 输入1 I2: 输入2 R: 结果字</p>	对双字数据和/或常数的相应位作逻辑异或非运算。 $(I_1 \cdot I_1 + 1) \cdot (I_2 \cdot I_2 + 1) + (\bar{I}_1 \cdot I_1 + 1) \cdot (\bar{I}_2 \cdot I_2 + 1) \rightarrow (R, R + 1)$ <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>I₁, I₁+1</th> <th>I₂, I₂+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1	1	1	1	1	0	0	0	1	0	0	0	1	输出	需要
I ₁ , I ₁ +1	I ₂ , I ₂ +1	R, R+1																			
1	1	1																			
1	0	0																			
0	1	0																			
0	0	1																			
求补	COM @COM	029	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> COM(029) Wd </div> <p>Wd: 字</p>	将Wd中的所有ON位置OFF, 并将所有OFF位置ON。 $\bar{Wd} \rightarrow Wd: 1 \rightarrow 0 \text{ 和 } 0 \rightarrow 1$	输出	需要															
双字求补	COML @COML	614	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> COML(614) Wd </div> <p>Wd: 字</p>	将Wd和Wd+1中的所有ON位置OFF, 并将所有OFF位置ON。 $\overline{(Wd+1, Wd)} \rightarrow (Wd+1, Wd)$	输出	需要															

A-1-12 特殊算术指令

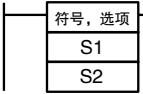
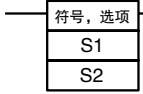

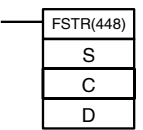
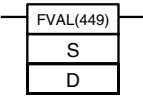
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
二进制平方根	ROTB @ROTB	620	 <p>S: 源首字 R: 结果字</p>	<p>计算指定字的32位二进制数的平方根，并将结果的整数部分输出到指定的结果字中。</p> 	输出	需要
BCD平方根	ROOT @ROOT	072	 <p>S: 源首字 R: 结果字</p>	<p>计算一个8位数的BCD数的平方根，并将结果的整数部分输出到指定的结果字中。</p> 	输出	需要
算术处理	APR @APR	069	 <p>C: 控制字 S: 源数据 R: 结果字</p>	<p>计算正弦、余弦或线性外插。 正弦或余弦计算：计算角度数据 (0° ~ 90°) 的正弦或余弦值，将结果保留到小数点后四位并以 BCD 格式输出。 线性外插：计算指定输入数据的线性外插，并以二进制格式输出。</p>	输出	需要
浮点数除	FDIV @FDIV	079	 <p>Dd: 被除数首字 Dr: 除数首字 R: 结果首字</p>	<p>将两个7位数的浮点数相除。浮点数用科学计数法表示 (7位尾数和1位指数)。</p> 	输出	需要
位计数器	BCNT @BCNT	067	 <p>N: 字数 S: 源首字 R: 结果字</p>	<p>统计指定字中的所有ON位数目。</p> 	输出	需要

A-1-13 浮点算术运算指令





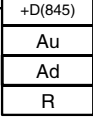
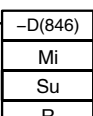
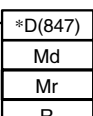
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
浮点数传送 (单精度)	MOVF	469	 <p>S: 源首字 D: 目的首字</p>	<p>将指定的 32 位浮点数传送到指定目的字中。</p> 	输出	需要
浮点数→16位	FIX @FIX	450	 <p>S: 源首字 R: 结果字</p>	<p>将一个32位浮点数据转换成一个带符号16位二进制数据, 并将结果放进指定的结果字中。</p> 	输出	需要
浮点数→32位	FIXL @FIXL	451	 <p>S: 源首字 R: 结果首字</p>	<p>将一个32位浮点数据转换成一个32位带符号二进制数据, 并将结果放进指定的结果字中。</p> 	输出	需要
16位→浮点数	FLT @FLT	452	 <p>S: 源字 R: 结果首字</p>	<p>将一个16位带符号二进制数据转换成一个32位浮点数据, 并将结果放进指定的结果字中。</p> 	输出	需要
32位→浮点数	FLTL @FLTL	453	 <p>S: 源首字 R: 结果首字</p>	<p>将一个32位带符号二进制数据转换成一个32位浮点数据, 并将结果放进指定的结果字中。</p> 	输出	需要
浮点数加	+F @+F	454	 <p>Au: 被加数首字 Ad: 加数首字 R: 结果首字</p>	<p>将两个32位浮点数相加, 并将结果放进指定的结果字中。</p> 	输出	需要
浮点数减	-F @-F	455	 <p>Mi: 被减数首字 Su: 减数首字 R: 结果首字</p>	<p>将两个32位浮点数相减, 并将结果放进指定的结果字中。</p> 	输出	需要
浮点数乘	*F @*F	456	 <p>Md: 被乘数首字 Mr: 乘数首字 R: 结果首字</p>	<p>将两个32位浮点数相乘, 并将结果放进指定的结果字中。</p> 	输出	需要

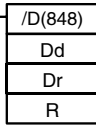








指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
浮点数除	/F @/F	457	 <p>Dd: 被除数首字 Dr: 除数首字 R: 结果首字</p>	<p>将两个32位浮点数相除并将结果放进指定的结果字中。</p>  <p>被除数 (浮点数据, 32位)</p> <p>除数 (浮点数据, 32位)</p> <p>结果 (浮点数据, 32位)</p>	输出	需要
度→弧度	RAD @RAD	458	 <p>S: 源首字 R: 结果首字</p>	<p>将一个32位浮点数从度转换成弧度, 并将结果放入指定的结果字中。</p>  <p>源 (度, 32位浮点数)</p> <p>结果 (弧度, 32位浮点数)</p>	输出	需要
弧度→度	DEG @DEG	459	 <p>S: 源首字 R: 结果首字</p>	<p>将一个32位浮点数从弧度转换成度, 并将结果放入指定的结果字中。</p>  <p>源 (弧度, 32位浮点数)</p> <p>结果 (度, 32位浮点数)</p>	输出	需要
正弦	SIN @SIN	460	 <p>S: 源首字 R: 结果首字</p>	<p>计算32位浮点数(用弧度表示)的正弦值, 并将结果放入指定的结果字中。</p>  <p>SIN ()</p>	输出	需要
高速正弦	SINQ @SINQ	475	 <p>S: 源首字 R: 结果首字</p>	<p>计算32位浮点数(用弧度表示)的正弦值, 并将结果放入指定的结果字中。</p>  <p>SIN ()</p>	输出	需要
余弦	COS @COS	461	 <p>S: 源首字 R: 结果首字</p>	<p>计算32位浮点数(用弧度表示)的余弦值, 并将结果放入指定的结果字中。</p>  <p>COS ()</p>	输出	需要
高速余弦	COSQ @COSQ	476	 <p>S: 源首字 R: 结果首字</p>	<p>计算32位浮点数(用弧度表示)的余弦值, 并将结果放入指定的结果字中。</p>  <p>COS ()</p>	输出	需要
正切	TAN @TAN	462	 <p>S: 源首字 R: 结果首字</p>	<p>计算32位浮点数(用弧度表示)的正切值, 并将结果放入指定的结果字中。</p>  <p>TAN ()</p>	输出	需要




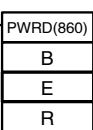



指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
高速正切	TANQ @TANQ	477	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">TANQ(477)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的(用弧度表示)的正切值, 并将结果放入指定的结果字中。 $\text{TAN} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$	输出	需要
反正弦	ASIN @ASIN	463	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">ASIN(463)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的反正弦值, 并将结果放入指定的结果字中。(反正弦函数与正弦函数正好相反, 该函数计算出-1到1之间的给定正弦值所对应的角度值。) $\text{SIN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$	输出	需要
反余弦	ACOS @ACOS	464	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">ACOS(464)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的反余弦值, 并将结果放入指定的结果字中。(反余弦函数与余弦函数正好相反, 该函数计算出-1到1之间的给定余弦值所对应的角度值。) $\text{COS}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$	输出	需要
反正切	ATAN @ATAN	465	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">ATAN(465)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的反正切值, 并将结果放入指定的结果字中。(反正切函数与正切函数正好相反, 该函数计算出给定正切值所对应的角度值。) $\text{TAN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$	输出	需要
平方根	SQRT @SQRT	466	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">SQRT(466)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的平方根, 并将结果放入指定的结果字中。 $\sqrt{\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}}$ $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$	输出	需要
指数	EXP @EXP	467	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">EXP(467)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的自然指数(底数为e), 并将结果放入指定的结果字中。 $e^{\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}}$ 源 (32位浮点数) $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ 结果 (32位浮点数)	输出	需要
对数	LOG @LOG	468	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">LOG(468)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>S: 源首字 R: 结果首字</p>	计算32位浮点数的自然对数(底数为e), 并将结果放入指定的结果字中。 $\log_e \begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}$ 源 (32位浮点数) $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ 结果 (32位浮点数)	输出	需要
指数幂	PWR @PWR	840	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">PWR(840)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">B</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px; width: 20px; text-align: center;">E</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; width: 20px; text-align: center;">R</div> <p>B: 底数首字 E: 指数首字 R: 结果首字</p>	自乘一个32位浮点数到另一个32位浮点数的幂。 $\begin{array}{ c c } \hline \text{E+1} & \text{E} \\ \hline \end{array}$ 幂 $\begin{array}{ c c } \hline \text{B+1} & \text{S} \\ \hline \end{array} \rightarrow \begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ 底数	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
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)	<p>使用LD:</p>  <p>使用AND:</p>  <p>使用OR:</p>  <p>S1: 比较数据1 S2: 比较数据2</p>	<p>比较指定的单精度数据 (32 位) 或常数并在比较结果为真时生成一个 ON 执行条件。</p> <p>有三类浮点符号比较指令供使用: LD (载入)、AND 和 OR。</p>	<p>LD: 逻辑开始</p> <p>AND 或 OR: 在梯级上连续</p>	<p>LD: 不需要</p> <p>AND 或 OR: 需要</p>
浮点数→ASCII	FSTR @FSTR	448	 <p>S: 源首字 C: 控制字 D: 目的字</p>	<p>将指定的单精度浮点数据 (32 位小数点或指数格式) 转换成字符串数据 (ASCII) 并将结果输出到目的字中。</p>	输出	需要
ASCII→浮点数	FVAL @FVAL	449	 <p>S: 源字 D: 目的首字</p>	<p>将代表单精度浮点数据 (小数点或指数格式) 的指定字符串 (ASCII) 转换成 32 位单精度浮点数据并将结果输出到目的字中。</p>	输出	需要


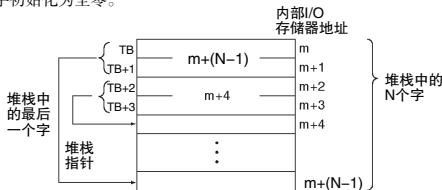
A-1-14 双精度浮点数指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双精度浮点数→ 16 位二进制数	FIXD @FIXD	841	 <p>S: 源首字 D: 目的字</p>	将指定的双精度浮点数(64 位)转换成 16 位带符号二进制数, 并将结果输出到目的字中。	输出	需要
双精度浮点数→ 32 位二进制数	FIXLD @FIXLD	842	 <p>S: 源首字 D: 目的首字</p>	将指定的双精度浮点数(64 位)转换成 32 位带符号二进制数, 并将结果输出到目的字中。	输出	需要
16 位二进制数→ 双精度浮点数	DBL @DBL	843	 <p>S: 源字 D: 目的首字</p>	将指定的 16 位带符号二进制数转换成双精度浮点数(64 位), 并将结果输出到目的字中。	输出	需要
32 位二进制数→ 双精度浮点数	DBLL @DBLL	844	 <p>S: 源首字 D: 目的首字</p>	将指定的 32 位带符号二进制数转换成双精度浮点数(64 位), 并将结果输出到目的字中。	输出	需要
双精度浮点数加	+D @+D	845	 <p>Au: 被加数首字 Ad: 加数首字 R: 结果首字</p>	将两个指定的双精度浮点数(各 64 位)相加, 并将结果输出到结果字中。	输出	需要
双精度浮点数减	-D @-D	846	 <p>Mi: 被减数首字 Su: 减数首字 R: 结果首字</p>	将两个指定的双精度浮点数(各 64 位)相减, 并将结果输出到结果字中。	输出	需要
双精度浮点数乘	*D @*D	847	 <p>Md: 被乘数首字 Mr: 乘数首字 R: 结果首字</p>	将两个指定的双精度浮点数(各 64 位)相乘, 并将结果输出到结果字中。	输出	需要

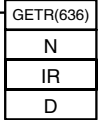
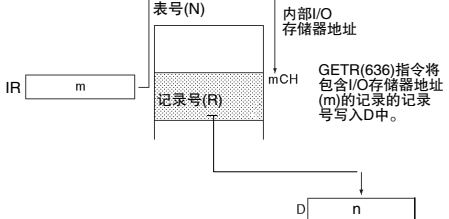
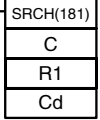
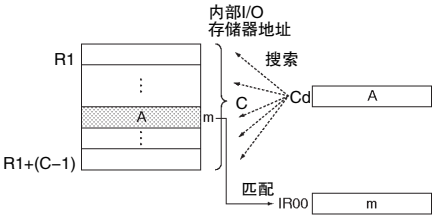
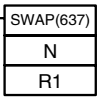
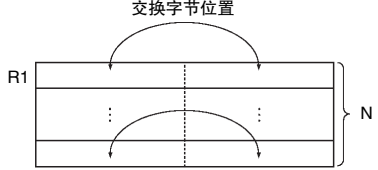
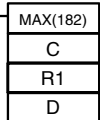
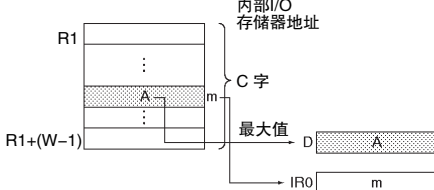
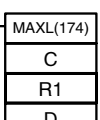
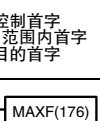
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双精度浮点数除	/D @/D	848	 <p>Dd: 被除数首字 Dr: 除数首字 R: 结果首字</p>	将两个指定的双精度浮点数 (各 64 位) 相除, 并将结果输出到结果字中。	输出	需要
双精度度→弧度	RADD @RADD	849	 <p>S: 源首字 R: 结果首字</p>	将指定的双精度浮点数 (64 位) 从度转换成弧度, 并将结果输出到结果字中。	输出	需要
双精度弧度→度	DEGD @DEGD	850	 <p>S: 源首字 R: 结果首字</p>	将指定的双精度浮点数 (64 位) 从弧度转换成度, 并将结果输出到结果字中。	输出	需要
双精度正弦	SIND @SIND	851	 <p>S: 源首字 R: 结果首字</p>	计算以指定的双精度浮点数 (64 位) 表示的角度 (弧度) 的正弦值, 并将结果输出到结果字中。	输出	需要
双精度余弦	COSD @COSD	852	 <p>S: 源首字 R: 结果首字</p>	计算以指定的双精度浮点数 (64 位) 表示的角度 (弧度) 的余弦值, 并将结果输出到结果字中。	输出	需要
双精度正切	TAND @TAND	853	 <p>S: 源首字 R: 结果首字</p>	计算以指定的双精度浮点数 (64 位) 表示的角度 (弧度) 的正切值, 并将结果输出到结果字中。	输出	需要
双精度反正弦	ASIND @ASIND	854	 <p>S: 源首字 R: 结果首字</p>	从以指定的双精度浮点数 (64 位) 表示的正弦值计算出角度值 (以弧度表示), 并将结果输出到结果字中。(反正弦函数与正弦函数正好相反, 该函数计算出 -1 到 1 之间的给定正弦值所对应的角度值。)	输出	需要
双精度反余弦	ACOSD @ACOSD	855	 <p>S: 源首字 R: 结果首字</p>	从以指定的双精度浮点数 (64 位) 表示的余弦值计算出角度值 (以弧度表示), 并将结果输出到结果字中。(反余弦函数与余弦函数正好相反, 该函数计算出 -1 到 1 之间的给定余弦值所对应的角度值。)	输出	需要
双精度反正切	ATAND @ATAND	856	 <p>S: 源首字 R: 结果首字</p>	从以指定的双精度浮点数 (64 位) 表示的正切值计算出角度值 (以弧度表示), 并将结果输出到结果字中。(反正切函数与正切函数正好相反, 该函数计算出给定正切值所对应的角度值。)	输出	需要

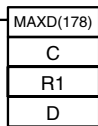
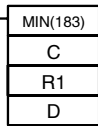
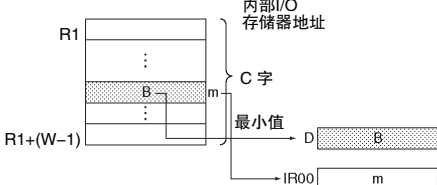
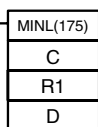
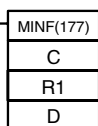
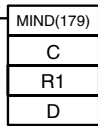
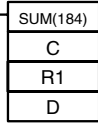
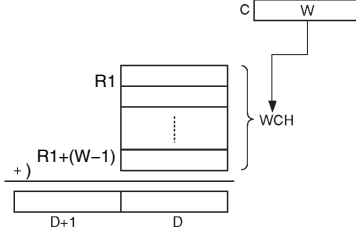
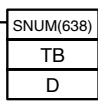
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
双精度平方根	SQRTD @SQRTD	857	 <p>S: 源首字 R: 结果首字</p>	计算指定的双精度浮点数 (64 位) 的平方根, 并将结果输出到结果字中。	输出	需要
双精度指数	EXPD @EXPD	858	 <p>S: 源首字 R: 结果首字</p>	计算指定的双精度浮点数 (64 位) 的自然 (底数为 e) 指数, 并将结果输出到结果字中。	输出	需要
双精度对数	LOGD @LOGD	859	 <p>S: 源首字 R: 结果首字</p>	计算指定的双精度浮点数 (64 位) 的自然 (底数为 e) 对数, 并将结果输出到结果字中。	输出	需要
双精度指数幂	PWRD @PWRD	860	 <p>B: 底数首字 E: 指数首字 R: 结果首字</p>	自乘一个双精度浮点数 (64 位) 到另一个双精度浮点数, 并将结果输出到结果字中。	输出	需要
双精度符号比较	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)	<p>使用LD:</p>  <p>使用AND:</p>  <p>使用OR:</p>  <p>S1: 比较数据1 S2: 比较数据2</p>	比较指定的双精度数据 (64 位), 并在比较结果为真时生成一个 ON 执行条件。 有三类浮点符号比较指令供使用: LD (载入)、AND 和 OR。	LD: 逻辑开始 AND 或 OR: 在梯级上连续	LD: 不需要 AND 或 OR: 需要

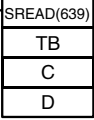
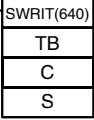

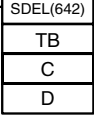
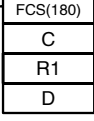
A-1-15 表格数据处理指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
设置堆栈	SSET @SSET	630	 <p>TB: 堆栈首地址 N: 字数</p>	<p>定义从指定字开始的指定长度的堆栈, 并将该数据区中的字初始化为全零。</p> 	输出	需要


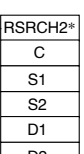
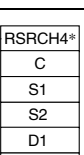
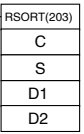
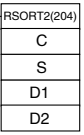
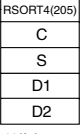
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
压入栈	PUSH @PUSH	632	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> PUSH(632) TB S </div> <p>TB: 堆栈首地址 S: 源字</p>	<p>将一个字的数据写到指定堆栈中。</p>	输出	需要
后进先出	LIFO @LIFO	634	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> LIFO(634) TB D </div> <p>TB: 堆栈首地址 D: 目的字</p>	<p>读取写入指定堆栈中的最后一个字数据(堆栈中的最新数据)。</p>	输出	需要
先进先出	FIFO @FIFO	633	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> FIFO(633) TB D </div> <p>TB: 堆栈首地址 D: 目的字</p>	<p>读取写入指定堆栈中的第一个字数据(堆栈中的最早数据)。</p>	输出	需要
定义记录表	DIM @DIM	631	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> DIM(631) N LR NR TB </div> <p>N: 表号 LR: 每条记录的长度 NR: 记录数 TB: 表首字</p>	<p>通过声明每条记录的长度和记录数的方法来定义记录表。 最多可定义16个记录表。</p>	输出	需要
设置记录位置	SETR @SETR	635	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SETR(635) N R D </div> <p>N: 表号 R: 记录号 D: 目的变址寄存器</p>	<p>将指定记录的位置(记录开头的内部I/O存储器地址)写入指定变址寄存器中。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
获取记录号	GETR @GETR	636	 <p>N: 表号 IR: 变址寄存器 D: 目的字</p>	<p>返回包含在指定变址寄存器中的内部I/O存储器地址处的记录的记录号。</p>  <p>GETR(636)指令将包含I/O存储器地址(m)的记录的记录号写入D中。</p>	输出	需要
数据搜索	SRCH @SRCH	181	 <p>C: 控制首字 R1: 范围内首字 Cd: 比较数据</p>	<p>在字范围内搜索一个字的数据。</p> 	输出	需要
交换字节	交换字节	637	 <p>N: 字数 R1: 范围内首字</p>	<p>将范围内的所有字的左字节和右字节交换。</p> <p>交换字节位置</p> 	输出	需要
寻找最大值	MAX @MAX	182	 <p>C: 控制首字 R1: 范围内首字 D: 目的字</p>	<p>寻找范围内的最大值。</p> 	输出	需要
寻找双字长最大值	MAXL @MAXL	174	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	<p>将C指定的数据项数量作为双字表数据(从R1指定的范围中的首字开始)处理,并将表中的最大值输出到D+1和D。</p>	输出	需要
寻找浮点数最大值	MAXF @MAXF	176	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	<p>将C指定的数据项数量作为单精度双字浮点数据(从R1指定的范围中的首字开始)处理,并将表中的最大值输出到D+1和D。</p>	输出	需要

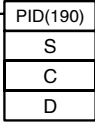
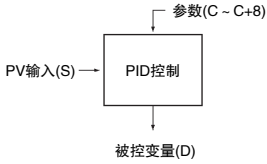
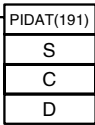
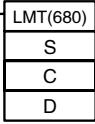
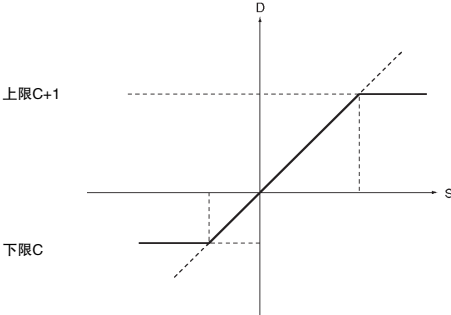
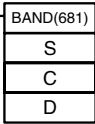
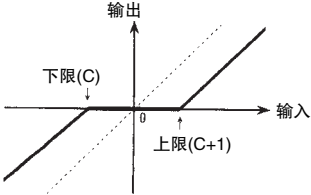
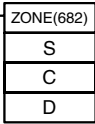
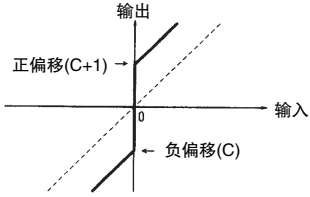
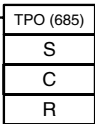
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
寻找双精度浮点数最大值	MAXD @MAXD	178	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	将 C 指定的数据项数量作为双精度双字浮点数据 (从 R1 指定的范围中的首字开始) 处理, 并将表中的最大值输出到 D+1 和 D。	输出	需要
寻找最小值	MIN @MIN	183	 <p>C: 控制首字 R1: 范围内首字 D: 目的字</p>	寻找范围内的最小值。 	输出	需要
寻找双字长最小值	MINL @MINL	175	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	将 C 指定的数据项数量作为双字表数据 (从 R1 指定的范围中的首字开始) 处理, 并将表中的最小值输出到 D+1 和 D。	输出	需要
寻找浮点数最小值	MINF @MINF	177	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	将 C 指定的数据项数量作为单精度双字浮点数据 (从 R1 指定的范围中的首字开始) 处理, 并将表中的最小值输出到 D+1 和 D。	输出	需要
寻找双精度浮点数最小值	MIND @MIND	179	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	将 C 指定的数据项数量作为双精度双字浮点数据 (从 R1 指定的范围中的首字开始) 处理, 并将表中的最小值输出到 D+1 和 D。	输出	需要
求和	SUM @SUM	184	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	将范围内的字节或字相加, 并将结果输出到两个字中。 	输出	需要
读取栈大小	SNUM @SNUM	638	 <p>TB: 堆栈首地址 D: 目的字</p>	统计指定堆栈中的堆栈数据量 (字数)。	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
读取栈数据	SREAD @SREAD	639	 <p>TB: 堆栈首地址 C: 偏移值 D: 目的字</p>	从堆栈的指定数据元素中读数据。偏移值表示所需数据元素的位置 (在当前指针位置之前的数据元素数目)。	输出	需要
覆盖栈数据	SWRIT @SWRIT	640	 <p>TB: 堆栈首地址 C: 偏移值 S: 源数据</p>	将源数据写入堆栈中的指定数据元素中 (覆盖原来的数据)。偏移值表示所需数据元素的位置 (在当前指针位置之前的数据元素数目)。	输出	需要
插入栈数据	SINS @SINS	641	 <p>TB: 堆栈首地址 C: 偏移值 S: 源数据</p>	在堆栈的指定位置处插入源数据并将堆栈中的其余数据下移。偏移值表示插入点的位置 (在当前指针位置之前的数据元素数目)。	输出	需要
删除栈数据	SDEL @SDEL	642	 <p>TB: 堆栈首地址 C: 偏移值 D: 目的字</p>	删除堆栈中指定位置处的数据元素并将堆栈中的其余数据上移。偏移值表示删除点的位置 (在当前指针位置之前的数据元素数目)。	输出	需要
帧校验和	FCS @FCS	180	 <p>C: 控制首字 R1: 范围内首字 D: 目的首字</p>	计算指定范围的 FCS 值并以 ASCII 码输出该值。	输出	需要

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
无符号单字记录搜索指令	RSRCH + <, <=, =, >, >= @RSRCH + <, <=, =, >, >=	360 (<), 361 (<=), 362 (=), 363 (>), 364 (>=) (>=)	 <p>C: 控制首字 S1: 要搜索的首记录首字 S2: 搜索数据 D1: 目的首字 D2: 目的变址寄存器</p>	从 S1 指定的表开始, 无符号单字记录搜索指令搜索 S2 指定的数据 (单字)。当找到符合指定条件的记录时, 将该记录的记录号及数据输出到前面的 D1。当指定变址寄存器用于 D2 时, 则将相匹配记录的地址输出到变址寄存器。不使用变址寄存器输出时, 将 D2 设定为 #00000000。	输出	需要
无符号双字记录搜索指令	RSRCH2 + <, <=, =, >, >= @RSRCH2 + <, <=, =, >, >=	370 (<), 371 (<=), 372 (=), 373 (>), 374 (>=) (>=)	 <p>C: 控制首字 S1: 要搜索的首记录首字 S2: 搜索数据的首字 D1: 目的首字 D2: 目的变址寄存器</p>	从 S1 指定的表开始, 无符号双字记录搜索指令搜索 S2 指定的数据 (双字)。当找到符合指定条件的记录时, 将该记录的记录号及数据输出到前面的 D1。当指定变址寄存器用于 D2 时, 则将相匹配记录的地址输出到变址寄存器。不使用变址寄存器输出时, 将 D2 设定为 #00000000。	输出	需要
无符号四字记录搜索指令	RSRCH4 + <, <=, =, >, >= @RSRCH4 + <, <=, =, >, >=	380 (<), 381 (<=), 382 (=), 383 (>), 384 (>=) (>=)	 <p>C: 控制首字 S1: 要搜索的首记录首字 S2: 搜索数据 D1: 目的首字 D2: 目的变址寄存器</p>	从 S1 指定的表开始, 无符号四字记录搜索指令搜索 S2 指定的数据 (四字)。当找到符合指定条件的记录时, 将该记录的记录号及数据输出到前面的 D1。当指定变址寄存器用于 D2 时, 则将相匹配记录的地址输出到变址寄存器。不使用变址寄存器输出时, 将 D2 设定为 #00000000。	输出	需要
无符号单字记录排序	RSORT @RSORT	203	 <p>C: 控制首字 S: 要排序的首记录的 首字 D1: 排序结果的首字 D2: 目的变址寄存器</p>	根据控制字对 S 指定的表中的记录 (单字) 进行排序。	输出	需要
无符号双字记录排序	RSORT2 @RSORT2	204	 <p>C: 控制首字 S: 要排序的首记录的 首字 D1: 排序结果的首字 D2: 目的变址寄存器</p>	根据控制字对 S 指定的表中的记录 (双字) 进行排序。	输出	需要
无符号四字记录排序	RSORT4 @RSORT4	205	 <p>C: 控制首字 S: 要排序的首记录的 首字 D1: 排序结果的首字 D2: 目的变址寄存器</p>	根据控制字对 S 指定的表中的记录 (四字) 进行排序。	输出	需要

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	 <p>S: 输入字 C: 参数首字 D: 输出字</p>	<p>根据指定参数执行PID控制。</p> 	输出	需要
带自整定的PID控制	PIDAT	191	 <p>S: 输入字 C: 参数首字 D: 输出字</p>	<p>根据指定参数执行PID控制。PID常数可用PIDAT(191)指令进行自整定。</p>	输出	需要
限位控制	LMT @LMT	680	 <p>S: 输入字 C: 限位首字 D: 输出字</p>	<p>根据输入数据是否在上、下限之内来控制输出数据。</p> 	输出	需要
静带控制	BAND @BAND	681	 <p>S: 输入字 C: 限位首字 D: 输出字</p>	<p>根据输入数据是否在静带范围内来控制输出数据。</p> 	输出	需要
静域控制	ZONE @ZONE	682	 <p>S: 输入字 C: 限位首字 D: 输出字</p>	<p>将指定的偏移值和输入数据相加并输出结果。</p> 	输出	需要
时间比例输出	TPO	685	 <p>S: 输入字 C: 参数首字 R: 脉冲输出位</p>	<p>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，并从指定输出出来输出结果。</p>	输出	需要


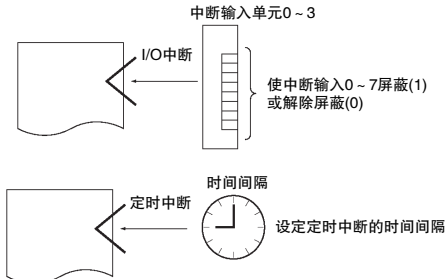


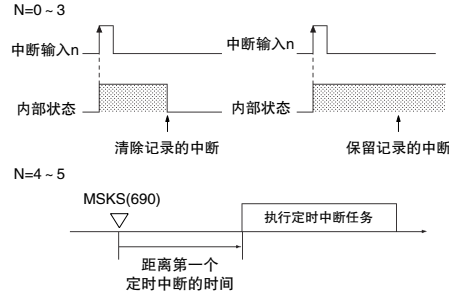

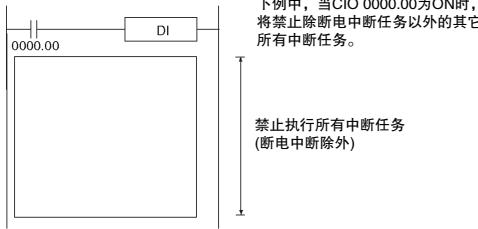
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件												
定标	SCL @SCL	194	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>SCL(194)</p> <p>S</p> <p>P1</p> <p>R</p> </div> <p>S: 源字 P1: 参数首字 R: 结果字</p>	<p>根据指定的线性函数，将无符号二进制数据转换成无符号BCD数据。</p> <p>根据由A点和B点定义的线性函数来执行定标</p> <p>R (无符号BCD)</p> <p>点 A</p> <p>点 B</p> <p>Ad</p> <p>Bd</p> <p>As</p> <p>Bs</p> <p>S (无符号二进制数)</p> <table border="1" style="margin-left: 20px;"> <tr><td>P</td><td>Ad (BCD)</td><td>转换后的值</td></tr> <tr><td>P1 + 1</td><td>As (BIN)</td><td></td></tr> <tr><td>P1 + 2</td><td>Bd (BCD)</td><td>转换后的值</td></tr> <tr><td>P1 + 3</td><td>Bs (BIN)</td><td></td></tr> </table>	P	Ad (BCD)	转换后的值	P1 + 1	As (BIN)		P1 + 2	Bd (BCD)	转换后的值	P1 + 3	Bs (BIN)		输出	需要
P	Ad (BCD)	转换后的值																
P1 + 1	As (BIN)																	
P1 + 2	Bd (BCD)	转换后的值																
P1 + 3	Bs (BIN)																	
定标 2	SCL2 @SCL2	486	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>SCL2(486)</p> <p>S</p> <p>P1</p> <p>R</p> </div> <p>S: 源字 P1: 参数首字 R: 结果字</p>	<p>根据指定的线性函数，将带符号二进制数据转换成带符号BCD数据。偏移值可在定义线性函数时输入。</p> <p>正偏移量</p> <p>R (带符号BCD)</p> <p>负偏移量</p> <p>R (带符号BCD)</p> <p>偏移量</p> <p>偏移量</p> <p>ΔY</p> <p>ΔX</p> <p>ΔY</p> <p>ΔX</p> <p>S (带符号二进制)</p> <p>S (带符号二进制)</p> <table border="1" style="margin-left: 20px;"> <tr><td>P1</td><td>偏移量</td><td>(带符号二进制)</td></tr> <tr><td>P1 + 1</td><td>ΔY</td><td>(带符号二进制)</td></tr> <tr><td>P1 + 2</td><td>ΔX</td><td>(带符号BCD)</td></tr> </table> <p>偏移量 0000</p> <p>R (带符号BCD)</p> <p>偏移量 = 0000 hex</p> <p>ΔY</p> <p>ΔX</p> <p>S (带符号二进制)</p>	P1	偏移量	(带符号二进制)	P1 + 1	ΔY	(带符号二进制)	P1 + 2	ΔX	(带符号BCD)	输出	需要			
P1	偏移量	(带符号二进制)																
P1 + 1	ΔY	(带符号二进制)																
P1 + 2	ΔX	(带符号BCD)																
定标 3	SCL3 @SCL3	487	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>SCL3(487)</p> <p>S</p> <p>P1</p> <p>R</p> </div> <p>S: 源字 P1: 参数首字 R: 结果字</p>	<p>根据指定的线性函数，将带符号BCD数据转换成带符号二进制数据。偏移值可在定义线性函数时输入。</p> <p>正偏移量</p> <p>R (带符号二进制)</p> <p>负偏移量</p> <p>R (带符号二进制)</p> <p>最大转换值</p> <p>最大转换值</p> <p>最小转换值</p> <p>最小转换值</p> <p>ΔY</p> <p>ΔX</p> <p>ΔY</p> <p>ΔX</p> <p>S (带符号BCD)</p> <p>S (带符号BCD)</p> <p>偏移量</p> <p>偏移量</p> <p>偏移量 0000</p> <p>R (带符号BCD)</p> <p>最大转换值</p> <p>最小转换值</p> <p>ΔY</p> <p>ΔX</p> <p>S (带符号BCD)</p>	输出	需要												

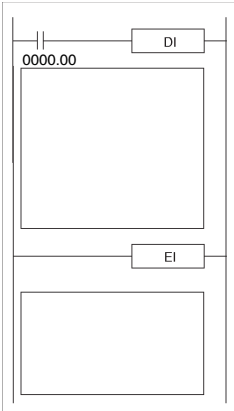
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
平均值	AVG	195	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">AVG(195)</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">S</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">N</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">R</div> </div> <p>S: 源字 N: 循环数 R: 结果字</p>	<p>计算指定循环次数的输入字的平均值。</p>	输出	需要

A-1-18 子程序指令

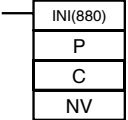
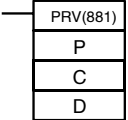
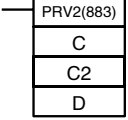
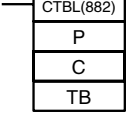
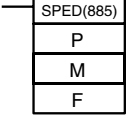
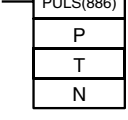
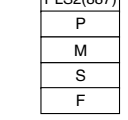
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
子程序调用	SBS @SBS	091	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">SBS(091)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">N</div> <p>N: 子程序号</p>	<p>调用指定子程序号的子程序并执行该程序。</p>	输出	需要
宏	MCRO @MCRO	099	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">MCRO(099)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">N</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">S</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">D</div> <p>N: 子程序号 S: 输入参数首字 D: 输出参数首字</p>	<p>调用指定子程序号的子程序并用S~S+3中的输入参数和D~D+3中的输出参数来执行该程序。</p>	输出	需要
子程序入口	SBN	092	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">SBN(092)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">N</div> <p>N: 子程序号</p>	<p>表示指定子程序号的子程序的开始。</p>	输出	需要
子程序返回	RET	093	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">RET(093)</div>	<p>表示子程序的结束。</p>	输出	不需要
全局子程序调用	GSBS @GSBS	750	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">GSBS(750)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">N</div> <p>N: 子程序号</p>	<p>调用指定子程序号的子程序并执行该程序。</p>	输出	需要
全局子程序入口	GSBN	751	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">GSBN(751)</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">N</div> <p>N: 子程序号</p>	<p>表示指定子程序号的子程序的开始。</p>	输出	不需要
全局子程序返回	GRET	752	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">GRET(752)</div>	<p>表示子程序的结束。</p>	输出	不需要

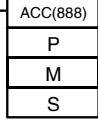

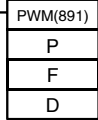
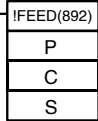
A-1-19 中断控制指令

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	 <p>N: 中断号 C: 控制数据</p>	<p>设置I/O中断、输入中断或定时中断的中断处理。当首次接通PLC的电源时，对I/O中断、输入中断和定时中断进行屏蔽(禁止)。MSKS(690)可用于解除屏蔽或屏蔽中断和设定定时中断的时间间隔。</p> 	输出	需要
读取中断屏蔽	MSKR @MSKR	692	 <p>N: 中断号 D: 目的字</p>	<p>读取由 MSKS(690) 所设置的当前中断处理设定值。</p>	输出	需要
清除中断	CLI @CLI	691	 <p>N: 中断号 C: 控制数据</p>	<p>清除或保留为I/O中断所记录的中断输入，或为定时中断设定首次定时中断的时间。</p> 	输出	需要
禁止中断	DI @DI	693		<p>禁止执行所有中断任务(断电中断除外)。</p>  <p>下例中，当CIO 0000.00为ON时，将禁止除断电中断任务以外的其它所有中断任务。</p> <p>禁止执行所有中断任务(断电中断除外)</p>	输出	需要

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)	<p>允许执行被DI(693)禁止的所有中断任务。</p>  <p>下例中，当CIO 0000.00为ON时，EI(694)指令允许执行被DI(693)指令所禁止的所有中断任务。</p> <p>禁止执行所有中断任务 (断电中断除外)。</p> <p>允许执行所有被禁止的中断任务</p>	输出	不需要

A-1-20 高速计数器 / 脉冲输出指令


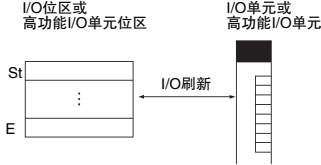


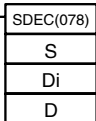
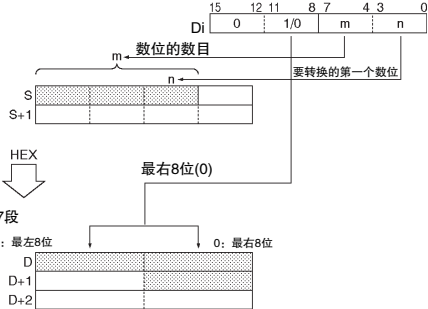
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
模式控制	INI @INI	880	 <p>P: 端口说明符 C: 控制数据 NV: 新值首字</p>	INI(880) 指令用于启动和停止比较表的比较、改变高速计数器的当前值 (PV)、改变计数器模式下的输入中断的当前值 (PV)、改变环形计数器的最大值、改变脉冲输出的当前值 (PV)(例如变为 0 以建立原点)、停止脉冲输出, 或者改变原点搜索 / 返回的设定。 (仅 CJ2M 单元支持 INI(880) 指令。)	输出	需要
读高速计数器的当前值 (PV)	PRV @PRV	881	 <p>P: 端口说明符 C: 控制数据 D: 目的首字</p>	PRV(881) 指令用于读取高速计数器、脉冲输出或中断输入 (计数器模式) 的当前值 (PV)。(仅 CJ2M 单元支持 PRV(881) 指令。)	输出	需要
计数器频率转换	PRV2 @PRV2	883	 <p>C: 控制数据 C2: 每转的脉冲数 D: 目的首字</p>	从高速计数器读取脉冲频率输入, 并将频率转换成旋转速度 (转数) 或者将计数器 PV 转换成总转数。结果作为 8 位数十六进制数输出至目的字。(仅 CJ2M 单元支持 PRV2(883) 指令。)	输出	需要
寄存器比较表	CTBL @CTBL	882	 <p>P: 端口说明符 C: 控制数据 TB: 比较表首字</p>	CTBL(882) 指令用于对高速计数器的当前值 (PV) 执行目标值或范围比较。(仅 CJ2M 单元支持 CTBL(882) 指令。)	输出	需要
速度输出	SPED @SPED	885	 <p>P: 端口说明符 M: 输出模式 F: 脉冲频率首字</p>	SPED(885) 用于指定频率并执行无加速或减速的脉冲输出。(仅 CJ2M 单元支持 SPED(885) 指令。)	输出	需要
设置脉冲	PULS @PULS	886	 <p>P: 端口说明符 T: 脉冲类型 N: 脉冲数</p>	PULS(886) 用于设置脉冲输出的脉冲数。(仅 CJ2M 单元支持 PULS(886) 指令。)	输出	需要
脉冲输出	PLS2 @PLS2	887	 <p>P: 端口说明符 M: 输出模式 S: 设定表首字 F: 启动频率首字</p>	PLS2(887) 用于设置脉冲频率和加速度 / 减速度, 从而 (以不同的加速度 / 减速度) 执行加速 / 减速脉冲输出。(仅 CJ2M 单元支持 PLS2(887) 指令。)	输出	需要

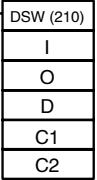
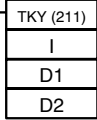


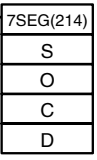
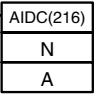
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
加速控制	ACC @ACC	888	 <p>P: 端口说明符 M: 输出模式 S: 设定表首字</p>	ACC(888) 用于设置脉冲频率和加速率 / 减速率, 从而 (以相同的加速率/减速率) 执行加速/减速率脉冲输出。(仅 CJ2M 单元支持 ACC(888) 指令。)	输出	需要
原点搜索	ORG @ORG	889	 <p>P: 端口说明符 C: 控制数据</p>	ORG(889) 用于执行原点搜索和返回。(仅 CJ2M 单元支持 ORG(889) 指令。)	输出	需要
占空比可变脉冲	PWM @PWM	891	 <p>P: 端口说明符 F: 频率 D: 占空比</p>	PWM(891) 用于输出占空比可变的脉冲。(仅 CJ2M 单元支持 PWM(891) 指令。)	输出	需要
中断进给	IFEED @IFEED	892	 <p>P: 端口说明符 C: 控制数据 S: 设定表首字</p>	IFEED(892) 使用输入中断作为从速度控制切换至位置控制的触发器, 并移动指定数目的脉冲。 (仅 CJ2M 单元支持 IFEED(892) 指令。)	输出	需要

A-1-21 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	 B: 位	STEP(008) 以下列 2 种方式工作, 具体视其位置以及是否已指定控制位而异。 (1) 开始一个指定的步。 (2) 结束该步程序区 (即步执行)。 步程序区从第一条 STEP(008) 指令 (总是带一个控制位) 开始, 到最后一条 STEP(008) 指令 (不带控制位) 结束。	输出	需要
步启动	SNXT	009	 B: 位	SNXT(009) 用于下列三种情况: (1) 开始步程序的执行。 (2) 继续到下一个步控制位。 (3) 结束步程序的执行。	输出	需要

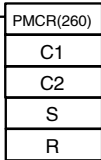
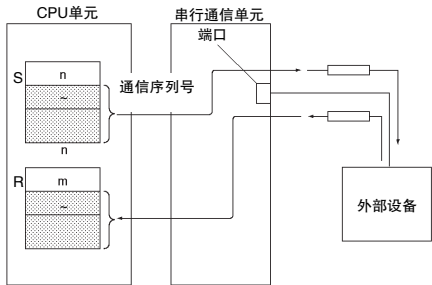
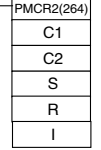
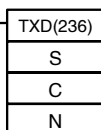
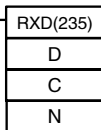
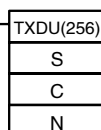
A-1-22 基本 I/O 单元指令

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	 St: 起始字 E: 结束字	刷新指定的 I/O 字。 	输出	需要
高功能 I/O 单元 I/O 刷新	FIORF @FIORF	225	 N: 单元号	即时刷新指定单元号的高功能 I/O 单元分配到的 I/O 字。	输出	需要
CPU 总线单元 I/O 刷新	DLNK @DLNK	226	 N: 单元号	即时刷新指定单元号的 CPU 总线单元分配到的 I/O 字。	输出	需要
7 段译码	SDEC @SDEC	078	 S: 源字 Di: 数位定义 D: 目的首字	将指定数位中的十六进制内容转换成相应的 8 位 7 段显示码, 并将其存入指定目的字中的高或低 8 位。 	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
数字开关输入	DSW	210	 <p>I: 数据输入字 (D0 ~ D3) O: 输出字 D: 结果首字 C1: 数位的数目 C2: 系统字</p>	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) 上设定的值并将 4 位或 8 位 BCD 数据存储到指定字中。	输出	需要
十进制数按键输入	TKY	211	 <p>I: 数据输入字 D1: 寄存器首字 D2: 按键输入字</p>	从与输入单元相连的十进制键盘读入数字值, 并将 8 位 BCD 数据存储到指定的字中。	输出	需要
十六进制数按键输入	HKY	212	 <p>I: 数据输入字 O: 输出字 D: 寄存器首字 C: 系统字</p>	从与输入单元和输出单元相连的十六进制键盘读入数字值, 并将 8 个数位的十六进制数存储到指定的字中。	输出	需要
矩阵输入	MTR	213	 <p>I: 数据输入字 O: 输出字 D: 目的首字 C: 系统字</p>	从与输入单元和输出单元 (使用 8 点输入和 8 点输出) 相连的 8 × 8 矩阵输入最多 64 个信号, 并将该 64 位数据存储到 4 个目的字中。	输出	需要
7 段显示输出	7SEG	214	 <p>S: 源首字 O: 输出字 C: 控制数据 D: 系统字</p>	将源数据 (4 位或 8 位 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	需要
模拟量输入直接转换 (适用于 CJ1W-AD042)	AIDC @AIDC	216	 <p>N: 单元号 A: 模拟量输入数</p>	以直接转换模式从 CJ1W-AD042 模拟输入量单元读取指定模拟输入号的输入转换值。	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
模拟量输出直接转换 (适用于 CJ1W-DA042V)	AODC @AODC	217	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> AODC(217) <hr/> N <hr/> A </div> <p>N: 单元号 A: 模拟量输出数</p>	以直接转换模式将指定模拟输出号的输出设置值输出至 CJ1W-DA042V 模拟量输出单元。	输出	需要
PCU 高速定位 (仅限 CJ1W-NC□□4 或 CJ1W-NC□81)	NCDMV @NCDMV	218	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> NCDMV(218) <hr/> C <hr/> A </div> <p>C: 控制数据 A: 直接操作命令区的首字</p>	NCDMV(218) 指令启动对 CJ1W-NC□□4 或 CJ1W-NC□81 位置控制单元的指定轴的绝对或相对高速点到点位置控制。	输出	需要
PCU 定位触发器 (仅限 CJ1W-NC□81)	NCDTR @NCDTR	219	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> NCDTR(219) <hr/> C </div> <p>C: 控制数据</p>	当对 CJ1W-NC□81 位置控制单元的存储器操作的启动条件为正在等待来自 NCDTR(219) 指令的命令时, NCDTR(219) 指令用于启动该序列。	输出	需要
智能 I/O 读	IORD @IORD	222	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> IORD(222) <hr/> C <hr/> S <hr/> D </div> <p>C: 控制数据 S: 传送源和字数 D: 传送目的和字数</p>	<p>读取高性能I/O单元或CPU总线单元的存储区内容(见注)。</p>	输出	需要
智能 I/O 写	IOWR @IOWR	223	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> IOWR(223) <hr/> C <hr/> S <hr/> D </div> <p>C: 控制数据 S: 传送源和字数 D: 传送目的和字数</p>	<p>将CPU单元的I/O/存储区内容输出至高性能I/O单元或CPU总线单元(见注)。</p>	输出	需要

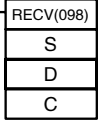
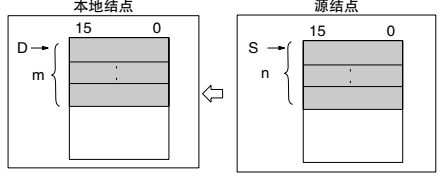
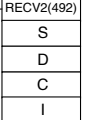
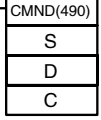
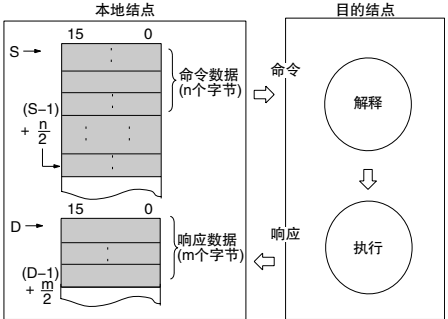
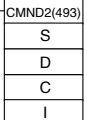
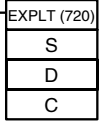
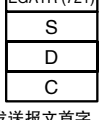
A-1-23 串行通信指令


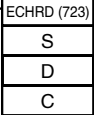
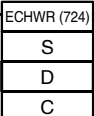
指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
协议宏	PMCR @PMCR	260	 <p>C1: 控制字1 C2: 控制字2 S: 发送首字 R: 接收首字</p>	<p>调用并执行在串行通信选件板(仅限CS系列)或串行通信单元中注册的通信序列。</p> 	输出	需要
协议宏 2	PMCR2 @PMCR2	264	 <p>C1: 控制字1 C2: 控制字2 S: 发送首字 R: 接收首字 I: 通信信息首字</p>	最多可同时使用 64 条 PMCR2 指令, 使用方法同 PMCR 指令。	输出	需要
发送	TXD @TXD	236	 <p>S: 源首字 C: 控制字 N: 字节数 0000 ~ 0100 Hex (十进制的0 ~ 256)</p>	根据 PLC 设置中无协议模式所指定的起始码 / 结束码, 从 CPU 单元的内置 RS-232C 端口 (无协议模式)、串行通信板的串行端口或 1.2 版或更高版本的单元 (无协议模式) 输出指定字节数的数据 (无需转换)。	输出	需要
接收	RXD @RXD	235	 <p>D: 目的首字 C: 控制字 N: 要存储的字节数 0000 ~ 0100 Hex (十进制的0 ~ 256)</p>	根据 PLC 设置中无协议模式所指定的起始码 / 结束码, 从 CPU 单元的内置 RS-232C 端口 (无协议模式)、串行通信板的串行端口或 1.2 版或更高版本的单元 (无协议模式) 读取指定字节数的数据 (从指定的首字开始)。	输出	需要
通过串行通信单元发送	TXDU @TXDU	256	 <p>S: 源首字 C: 控制首字 N: 字节数 (0000 ~ 0256 BCD)</p>	从 1.2 版或更高版本的串行通信单元的串行端口输出指定字节数的数据 (无需转换)。数据以无协议模式输出, 其起始代码和结束代码 (若有) 在所分配的 DM 区中指定。	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
通过串行通信单元接收数据	RXDU @RXDU	255	<p>D: 目的首字 C: 控制首字 N: 要存储的字节数 (0000 ~ 0100 Hex)</p>	从 1.2 版或更高版本的串行通信单元的串行端口读取指定字节数的数据 (从指定的首字开始)。数据以无协议模式读取, 其起始码和结束码 (若有) 在所分配的 DM 设置区中指定。	输出	需要
通过串行通信单元 / 串行端口直接接收 (仅限 CJ1W-SCU22、CJ1W-SCU32 或 CJ1W-SCU42)	DRXDU @DRXDU	261	<p>D: 目的首字 C: 控制首字 N: 字节数 0000 ~ 0100 Hex (0 ~ 256)</p>	从 CJ1W-SCU22/SCU32/SCU42 串行通信单元的串行端口将指定字节数的数据读取到 CPU 单元以指定的首字起始的存储器中。数据以无协议模式读取, 其起始码和结束码 (若有) 在所分配的 DM 设置区中指定。 该指令一执行时, 即从串行通信单元读取数据, 以实现高速数据接收。	输出	需要
通过串行通信单元 / 串行端口直接传送 (仅限 CJ1W-SCU22、CJ1W-SCU32 或 CJ1W-SCU42)	DTXDU @DTXDI	262	<p>S: 源首字 C: 控制首字 N: 字节数 0000 ~ 0100 Hex (0 ~ 256)</p>	从 CJ1W-SCU22/SCU32/SCU42 串行通信单元的串行端口输出指定字节数的数据。数据以无协议模式从指定的首字开始输出, 首字的起始码和结束码 (若有) 在所分配的 DM 设置区中指定。 该指令一执行时, 即将数据发送到串行通信单元, 以实现高速数据传送。	输出	需要
修改串行端口设定	STUP @STUP	237	<p>C: 控制字 (端口) S: 源首字</p>	修改 CPU 单元、串行通信单元 (CPU 总线单元) 或串行选件板上的串行端口的通信参数。STUP(237) 指令允许在 PLC 运行过程中修改协议模式。	输出	需要

A-1-24 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	<p>S: 源首字 D: 目的首字 C: 控制首字</p>	<p>将数据传送到网络的一个结点上。</p>	输出	需要
网络发送 2	SEND2 @DEND2	491	<p>S: 源首字 D: 目的首字 C: 控制首字 I: 通信信息首字</p>	最多可同时使用 64 条 SEND2 指令, 使用方法同 SEND 指令。	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
网络接收	RECV @RECV	098	 <p>S: 源首字 D: 目的首字 C: 控制首字</p>	<p>请求从网络的一个结点发送数据并接收该数据。</p> 	输出	需要
网络接收 2	RECV2 @RECV2	492	 <p>S: 源首字 D: 目的首字 C: 控制首字 I: 通信信息首字</p>	<p>最多可同时使用 64 条 RECV2 指令，使用方法同 RECV 指令。</p>	输出	需要
发布命令	CMND @CMND	490	 <p>S: 命令首字 D: 响应首字 C: 控制首字</p>	<p>发送FINS命令并接收响应</p> 	输出	需要
发布命令 2	CMND2 @CMND2	493	 <p>S: 命令首字 D: 响应首字 C: 控制首字 I: 通信信息首字</p>	<p>最多可同时使用 64 条 CMND2 指令，使用方法同 CMND 指令。</p>	输出	需要
发送显式报文	EXPLT	720	 <p>S: 发送报文首字 D: 接收报文首字 C: 控制首字</p>	<p>以任意服务代码发送一条显式报文。</p>	输出	需要
显式获取属性	EGATR	721	 <p>S: 发送报文首字 D: 接收报文首字 C: 控制首字信息</p>	<p>读取显式报文中包含的状态信息 (获取单个属性，服务代码: 0E Hex)。</p>	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
显式设定属性	ESATR	722	 <p>S: 发送报文首字 C: 控制首字</p>	写入显式报文中包含的状态信息 (设定单个属性, 服务代码: 0E Hex)。	输出	需要
显式字读取	ECHRD	723	 <p>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</p>	从网络中的一台远程 CPU 单元将数据读入本地 CPU 单元。(远程 CPU 单元必须支持显式报文。)	输出	需要
显式字写入	ECHWR	724	 <p>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</p>	从本地 CPU 单元将数据写入网络中的一台远程 CPU 单元。(远程 CPU 单元必须支持显式报文。)	输出	需要

A-1-25 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	<table border="1"> <tr><td>FREAD(700)</td></tr> <tr><td>C</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制字 S1: 源首字 S2: 文件名 D: 目的首字</p>	FREAD(700)	C	S1	S2	D	<p>将文件存储器中指定数据文件的指定数据或数据量读入CPU单元的指定数据区中。</p> <p>在S1+2和S1+3中指定的起始读地址</p> <p>在S2中指定的文件</p> <p>CPU单元</p> <p>在S1和S1+1中指定的字数</p> <p>存储卡或EM文件存储器 (由C的第4个数位指定)</p> <p>写入D和D+1的字</p> <p>字数 →</p> <p>在S2中指定的文件</p> <p>存储卡或EM文件存储器 (由C的第4个数位指定)</p>	输出	需要
FREAD(700)											
C											
S1											
S2											
D											
写数据文件	FWRIT @FWRIT	701	<table border="1"> <tr><td>FWRIT(701)</td></tr> <tr><td>C</td></tr> <tr><td>D1</td></tr> <tr><td>D2</td></tr> <tr><td>S</td></tr> </table> <p>C: 控制字 D1: 目的首字 D2: 文件名 S: 源首字</p>	FWRIT(701)	C	D1	D2	S	<p>用来自CPU单元中数据区的指定数据来覆写或追加文件存储器的指定数据文件中的数据。若指定的文件不存在, 则用该文件名来创建一个新文件。</p> <p>CPU单元</p> <p>在S中指定的起始地址</p> <p>在D1+2和D1+3中指定的起始字</p> <p>在D2中指定的文件</p> <p>在D1和D1+1中指定的字数</p> <p>覆写</p> <p>存储卡或EM文件存储器 (由C的第4个数位指定)</p> <p>CPU单元</p> <p>在S中指定的起始地址</p> <p>文件结尾</p> <p>在D1和D1+1中指定的字数</p> <p>在D2中指定的文件</p> <p>现有数据</p> <p>追加</p> <p>存储卡或EM文件存储器 (由C的第4个数位指定)</p> <p>CPU单元</p> <p>在S中指定的起始地址</p> <p>文件开头</p> <p>在D1和D1+1中指定的字数</p> <p>在D2中指定的文件</p> <p>创建的新文件</p> <p>存储卡或EM文件存储器 (由C的第4个数位指定)</p>	输出	需要
FWRIT(701)											
C											
D1											
D2											
S											

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	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> TWRIT C S1 S2 S3 S4 </div> <p>C: 控制字 S1: 要写入的 字节数 S2: 目录和文件名 S3: 写数据 S4: 分隔符</p>	<p>从 I/O 存储器中读取 ASCII 数据并将该数据作为文本文件存储到存储卡中 (写入一个新文件或对已有文件进行追加)。数据以 TXT 格式存储。</p>	输出	需要

A-1-26 显示指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
显示信息	MSG @MSG	046	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> MSG(046) N M </div> <p>N: 信息号 M: 信息首字</p>	<p>读取指定的 16 个扩展 ASCII 字, 并在外围设备 (如编程器) 上显示该信息。</p>	输出	需要

A-1-27 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	CADD @CADD	730	<div style="border: 1px solid black; padding: 2px; display: inline-block;">CADD(730)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">C</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">T</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">R</div> C: 日历首字 T: 时间首字 R: 结果首字	在指定字的日历数据中增加时间。 	输出	需要
日历减	CSUB @CSUB	731	<div style="border: 1px solid black; padding: 2px; display: inline-block;">CSUB(731)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">C</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">T</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">R</div> C: 日历首字 T: 时间首字 R: 结果首字	在指定字的日历数据中减去时间。 	输出	需要
小时→秒	SEC @SEC	065	<div style="border: 1px solid black; padding: 2px; display: inline-block;">SEC(065)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">D</div> S: 源首字 D: 目的首字	将以时/分/秒格式表示的时间数据转换成仅以秒表示的等值时间。 	输出	需要
秒→小时	HMS @HMS	066	<div style="border: 1px solid black; padding: 2px; display: inline-block;">HMS(066)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">D</div> S: 源首字 D: 目的首字	将以秒表示的时间数据转换成以时/分/秒格式表示的等值时间。 	输出	需要
时钟调整	DATE @DATE	735	<div style="border: 1px solid black; padding: 2px; display: inline-block;">DATE(735)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin: 2px;">S</div> S: 源首字	将内部时钟设定改为指定源字中的设定。 	输出	需要









A-1-28 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)	执行 TRSM(045) 指令时, 对预先选择的位或字的状态进行采样并将该状态存储到跟踪存储器中。TRSM(045) 指令在程序中的使用位置和次数不限。	输出	不需要

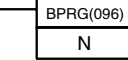
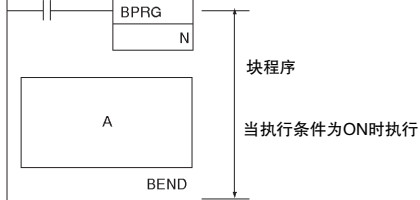
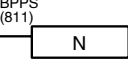
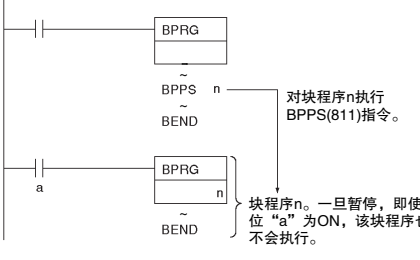
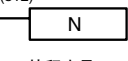
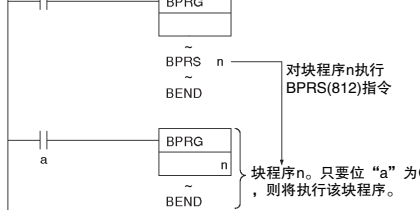
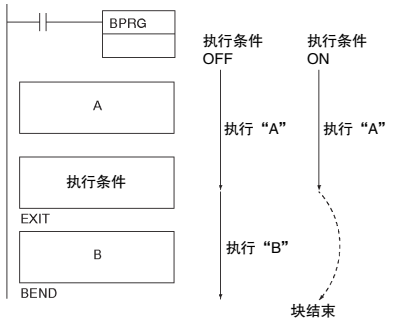
A-1-29 故障诊断指令

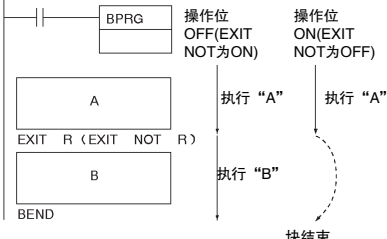
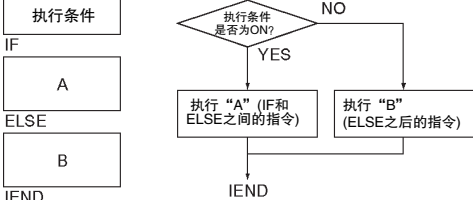

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
故障报警	FAL @FAL	006	— FAL(006) N S	<p>产生或清除用户定义的非致命错误。非致命错误不会使PLC停止运行。另外还可产生系统非致命错误。</p> <p>时间监控功能: 当执行条件A变为ON时的起始定时。如果在监控时间范围内, 输出B不变为ON, 则产生一个非致命错误。</p> <p>下一个指令块</p> <p>逻辑诊断块*</p> <p>逻辑诊断 执行条件C</p> <p>诊断输出B</p> <p>逻辑诊断功能 确定阻止输出B变为ON的输入项C。</p>	输出	需要
严重故障报警	FALS	007	— FALS(007) N S	<p>产生用户定义的致命错误。致命错误将使PLC停止运行。另外还可产生系统致命错误。</p> <p>下一个指令块</p> <p>逻辑诊断块*</p> <p>逻辑诊断 执行条件C</p> <p>诊断输出B</p> <p>逻辑诊断功能 确定阻止输出B变为ON的输入项C。</p>	输出	需要
故障点检测	FPD	269	— FPD(269) C T R	<p>通过监控执行的FPD(269)指令和诊断输出之间的时间, 并找到阻止输出变为ON的输入项, 从而来诊断一个指令块中的错误。</p> <p>时间监控功能: 当执行条件A变为ON时的起始定时。如果在监控时间范围内, 输出B不变为ON, 则产生一个非致命错误。</p> <p>下一个指令块</p> <p>逻辑诊断块*</p> <p>逻辑诊断 执行条件C</p> <p>诊断输出B</p> <p>逻辑诊断功能 确定阻止输出B变为ON的输入项C。</p>	输出	需要

A-1-30 其它指令

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
置进位	STC @STC	040	— 	置进位标志 (CY)。	输出	需要
清除进位	CLC @CLC	041	— 	将进位标志 (CY) 置 OFF。	输出	需要
选择 EM 存储 区块	EMBC @EMBC	281	—  N N: EM存储区块号	改变当前的 EM 存储区块。	输出	需要
延长 最大循环时间	WDT @WDT	094	—  T T: 定时器设定	仅在此指令执行的循环内延长最大循环时间。	输出	需要
保存条件标志	CCS @CCS	282	— 	保存条件标志的状态。	输出	需要
载入条件标志	CCL @CCL	283	— 	读取所保存的条件标志的状态。	输出	需要
将 CV 值转换 成地址值	FRMCV @FRMCV	284	—  S D S: 包含CV系列 存储器地址的字 D: 目的变址寄存器	将一个 CV 系列 PLC 的存储器地址转换成其等同的 CS/CJ 系列 PLC 的存储器地址。	输出	需要
将地址值转换 成 CV 值	TOCV @TOCV	285	—  S D S: 包含CS系列 存储器地址的 变址寄存器 D: 目的字	将一个 CS/CJ 系列 PLC 的存储器地址转换成其等同的 CVM1/CV 系列 PLC 的存储器地址。	输出	需要

A-1-31 块编程指令


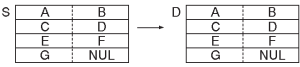
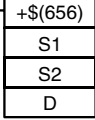
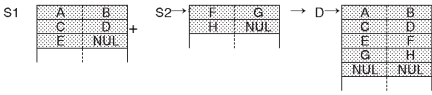
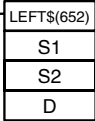
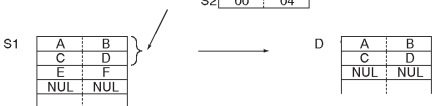
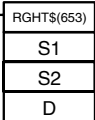
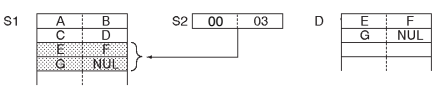
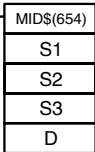
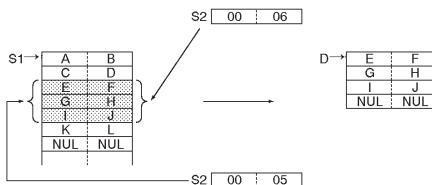
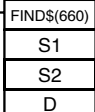
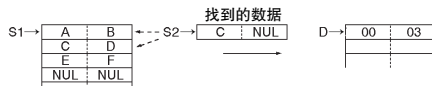


指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
块程序开始	BPRG	096	 <p>N: 块程序号</p>	<p>定义一个块编程区。每个BPRG(096)指令均须有一个对应的BEND(801)指令。</p> 	输出	需要
块程序结束	BEND	801	---	定义一个块编程区。每个 BPRG(096) 指令均须有一个对应的 BEND (801) 指令。	块程序	需要
块程序暂停	BPPS	811	 <p>N: 块程序号</p>	<p>暂停并从另一个块程序重新启动指定的块程序。</p> 	块程序	需要
块程序重新启动	BPRS	812	 <p>N: 块程序号</p>	<p>暂停并从另一个块程序重新启动指定的块程序。</p> 	块程序	需要
条件块退出	EXIT	806	<p>EXIT(806)</p> <p>B: 位操作数</p>	<p>如果执行条件为ON, 则无操作位的EXIT(806)指令将使程序退出。</p> 	块程序	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
条件块退出	EXIT	806	EXIT(806)B B: 位操作数	如果执行条件为ON, 则无操作位的EXIT(806)指令将使程序退出。 	块程序	需要
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, 则无操作位的 EXIT(806) 指令将使程序退出。	块程序	需要
条件块分支	IF	802	IF(802)	如果执行条件为ON, 将执行IF(802)和ELSE(803)之间的指令; 而如果执行条件为OFF时, 则将执行ELSE(803)和IEND(804)之间的指令。 	块程序	需要
条件块分支	IF	802	IF(802) B B: 位操作数	如果操作位为ON, 则将执行IF(802)和ELSE(803)之间的指令。如果操作位为OFF, 则将执行ELSE(803)和IEND(804)之间的指令。 	块程序	需要
条件块分支 (非)	IF NOT	802	IF(802)NOT B B: 位操作数	如果操作位为 ON, 将执行 IF(802) 和 ELSE(803) 之间的指令; 而如果操作位为 OFF, 则将执行 ELSE(803) 和 IEND(804) 之间的指令。	块程序	需要
条件块分支 (ELSE)	ELSE	803	---	如果 ELSE(803) 指令被忽略且操作位为 ON, 则将执行 IF(802) 和 IEND(804) 之间的指令。	块程序	需要
条件块分支结束	IEND	804	---	如果操作位为 OFF, 则仅执行 IEND(804) 之后的指令。	块程序	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
一个循环并等待	WAIT	805	WAIT(805)	<p>如果WAIT(805)的执行条件为ON, 则将跳过块程序中的其余指令。</p>	块程序	需要
一个循环并等待	WAIT	805	WAIT(805) B	<p>如果操作位为 OFF(对于 WAIT NOT(805) 为 ON), 则将跳过块程序中的其余指令。在下一个循环, 除 WAIT(805) 或 WAIT(805)NOT 指令的执行条件之外, 将不执行块程序中的任何指令。当执行条件变为 ON(对于 WAIT(805)NOT 为 OFF) 时, 将执行从 WAIT(805) 或 WAIT(805)NOT 到程序结尾的指令。</p>	块程序	需要
一个循环与等待(非)	WAIT NOT	805	WAIT(805) NOT B	<p>如果操作位为 OFF(对于 WAIT NOT(805) 为 ON), 则将跳过块程序中的其余指令。在下一个循环, 除 WAIT(805) 或 WAIT(805)NOT 指令的执行条件之外, 将不执行块程序中的任何指令。当执行条件变为 ON(对于 WAIT(805)NOT 为 OFF) 时, 将执行从 WAIT(805) 或 WAIT(805)NOT 到程序结尾的指令。</p>	块程序	需要
定时器等待	TIMW (BCD)	813	TIMW(813) N SV	<p>延迟块程序的执行, 直到指定的时间已过。当定时器的定时已到时, 将从TIMW(813)/TIMWX(816)之后的下一条指令开始继续执行。 SV: 0 ~ 999.9s (BCD格式) 和0 ~ 6.553.5s (二进制格式)</p>	块程序	需要
	TIMWX (二进制)	816	TIMWX(816) N SV			
计数器等待	CNTW (BCD)	814	CNTW(814) N SV	<p>延迟块程序中其余指令的执行, 直到到达指定的计数值。当计数器完成计数时, 将从CNTW(814)/CNTWX(817)之后的下一条指令开始继续执行。 SV: 0 ~ 9,999次(BCD格式)和0 ~ 65,535次(二进制格式)</p>	块程序	需要
	CNTWX (二进制)	818	CNTWX(818) N SV I			

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
高速定时器等待	TMHW (BCD)	815	TMHW(815) N SV N: 定时器号 SV: 设定值	<p>延迟块程序中其余指令的执行, 直到指定的时间已过。当定时器的定时已到时, 将从TMHW(815)之后的下一条指令开始继续执行。 SV: 0 ~ 99.99s(BCD格式)和0 ~ 655.35s(二进制格式)</p>	块程序	需要
	TMHWX (二进制)	817	TMHW(817) N SV N: 定时器号 SV: 设定值			
循环	LOOP	809	---	<p>LOOP(809)指定循环程序的开始。</p>	块程序	需要
LEND	LEND	810	LEND(810)	<p>LEND(810) 或 LEND(810)NOT 指令指定循环的结束。当执行到LEND(810) 或 LEND(810)NOT 指令时, 程序的执行将循环回到下一条先前的 LOOP(809) 指令, 直到 LEND(810) 或 LEND(810)NOT 的操作位分别变为 ON 或 OFF, 或者直到LEND(810)的执行条件变为 ON。</p>	块程序	需要
LEND	LEND	810	LEND(810) B B: 位操作数	<p>如果LEND(810)指令的操作位为OFF(或者LEND(810)NOT指令的操作位为ON), 则从LOOP(809)之后的下一条指令开始重复执行循环。如果LEND(810)指令的操作位为ON(或者LEND(810)NOT指令的操作位为OFF), 则循环结束, 并继续执行到LEND(810)或LEND(810)NOT之后的下一条指令。</p> <p>注: LEND(810)NOT指令的操作位的状态相反</p>	块程序	需要
LEND NOT	LEND NOT	810	LEND(810)NOT B: 位操作数	<p>LEND(810) 或 LEND(810)NOT 指令指定循环的结束。当执行到LEND(810) 或 LEND(810)NOT 指令时, 程序的执行将循环回到下一条先前的 LOOP(809) 指令, 直到 LEND(810) 或 LEND(810)NOT 的操作位分别变为 ON 或 OFF, 或者直到LEND(810)的执行条件变为 ON。</p>	块程序	需要

A-1-32 文本字符串处理指令

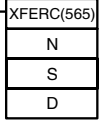
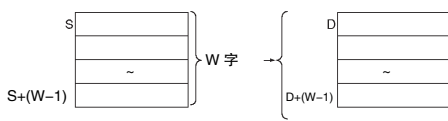
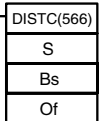
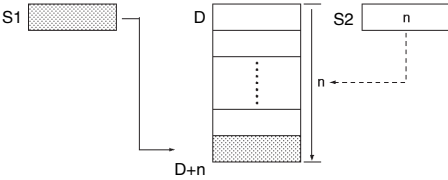
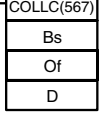
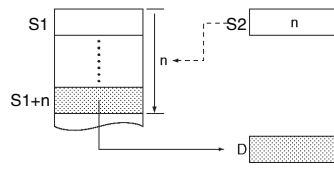
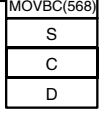
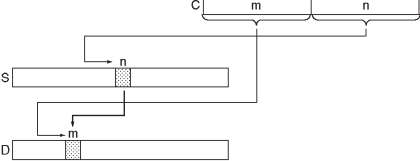
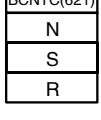

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
移动串	MOV\$ @MOV\$	664	 <p>S: 源首字 D: 目的首字</p>	传送一个文本串。 	输出	需要
链接串	+\$ @+\$	656	 <p>S1: 文本串1 S2: 文本串2 D: 目的首字</p>	将一个文本串链接到另一个文本串。 	输出	需要
取左串	LEFT\$ @LEFT\$	652	 <p>S1: 文本串首字 S2: 字符数 D: 目的首字</p>	从一个文本串的左边(起始处)开始取指定数目的字符。 	输出	需要
取右串	RGHT\$ @RGHT\$	653	 <p>S1: 文本串首字 S2: 字符数 D: 目的首字</p>	从一个文本串的右边(结尾处)开始读取指定数目的字符。 	输出	需要
取中间串	MID\$ @MID\$	654	 <p>S1: 文本串首字 S2: 字符数 S3: 起始位置 D: 目的首字</p>	从一个文本串的中间任意位置处开始读取指定数目的字符。 	输出	需要
查找串	FIND @FIND\$	660	 <p>S1: 文本串源首字 S2: 查找的文本串首字 D: 目的首字</p>	在一个文本串内查找指定的文本串。 	输出	需要
计算串长度	LENS\$ @LENS\$	650	 <p>S: 文本串首字 D: 目的首字</p>	计算一个文本串的长度。 	输出	需要

指令	助记符	代码	符号 / 操作数	功能	位置	执行条件
取代串	RPLC\$ @RPLC\$	661	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> RPLC\$(654) <hr/> S1 <hr/> S2 <hr/> S3 <hr/> S4 <hr/> D </div> <p>S1: 文本串首字 S2: 替换的文本串首字 S3: 字符数 S4: 起始位置 D: 目的首字</p>	<p>用一个指定的文本串从指定位置开始取代另一个文本串。</p>	输出	需要
删除串	DELS\$ @DELS\$	658	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> DELS\$(658) <hr/> S1 <hr/> S2 <hr/> S3 <hr/> D </div> <p>S1: 文本串首字 S2: 字符数 S3: 起始位置 D: 目的首字</p>	<p>从一个文本串的中间开始删除一个指定的文本串。</p> <p>要删除的字符数(由S2指定)</p>	输出	需要
交换串	XCHG\$ @XCHG\$	665	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> XCHG\$(665) <hr/> Ex1 <hr/> Ex2 </div> <p>Ex1: 交换首字1 Ex2: 交换首字2</p>	<p>用一个指定的文本串来取代另一个指定的文本串。</p>	输出	需要
清除串	CLRS\$ @CLRS\$	666	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> CLRS\$(666) <hr/> S </div> <p>S: 文本串首字</p>	<p>用NUL(00 Hex)来对整个文本串清零。</p>	输出	需要
插入串	INSS\$ @INSS\$	667	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> INSS\$(657) <hr/> S1 <hr/> S2 <hr/> S3 <hr/> D </div> <p>S1: 文本串基址首字 S2: 插入的文本串首字 S3: 起始位置 D: 目的首字</p>	<p>从一个文本串的中间开始插入一个指定的文本串。</p>	输出	需要
比较串	LD、AND、OR + = \$, < \$, < \$, <= \$, > \$, >=\$	670 (= \$) 671 (< \$) 672 (< \$) 673 (<= \$) 674 (> \$) 675 (>= \$)	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> 符号 <hr/> S1 <hr/> S2 </div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-top: 10px;"> 符号 <hr/> S1 <hr/> S2 </div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-top: 10px;"> 符号 <hr/> S1 <hr/> S2 </div> <p>S1: 文本串1 S2: 文本串2</p>	<p>串比较指令 (= \$, < \$, < \$, <= \$, > \$, >= \$) 根据 ASCII 代码的值从两个文本串的起始处开始比较这两个文本串的大小。如果比较的结果为真, 则为 LOAD、AND 或 OR 指令生成 ON 执行条件。</p>	LD: 逻辑开始 AND、OR: 在梯级上连续	需要

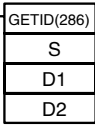








A-1-33 任务控制指令

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	<div style="border: 1px solid black; padding: 2px; display: inline-block;">TKON(820)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">N</div> N: 任务号	<p>使指定的任务成为可执行状态</p> <p>指定任务的任务号大于本地任务的 任务号(m<n)。</p> <p>指定任务的任务号小于本地任务的 任务号(m>n)。</p>	输出	需要
任务 OFF	TKOF @TKOF	821	<div style="border: 1px solid black; padding: 2px; display: inline-block;">TKOF(821)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">N</div> N: 任务号	<p>将指定任务置于待机状态。</p> <p>指定任务的任务号大于本地任务的 任务号(m<n)。</p> <p>指定任务的任务号小于本地任务的 任务号(m>n)。</p>	输出	需要

A-1-34 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	 <p>N: 字数 S: 源首字 D: 目的首字</p>	传送指定数目的连续字。 	输出	需要
单字分配	DISTC @DISTC	566	 <p>S: 源字 Bs: 目的基址 Of: 偏移量</p>	通过添加偏移值到基地址计算出目的字，并向其传送源字。  <p>也可写入堆栈中(压入栈操作)。</p>	输出	需要
数据收集	COLLC @COLLC	567	 <p>Bs: 源基地址 Of: 偏移量(BCD) D: 目的字</p>	通过添加偏移值到基地址计算出源字，并将其传送到目的字。  <p>也可从堆栈读取数据(栈读取操作)。</p>	输出	需要
位传送	MOVBC @MOVBC	568	 <p>S: 源字或数据 C: 控制字(BCD) D: 目的字</p>	传送指定的位。 	输出	需要
位计数器	BCNTC @BCNTC	621	 <p>N: 字数(BCD) S: 源首字 R: 结果字</p>	统计指定字中的所有ON位数目。 	输出	需要

A-1-35 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
获取变量 ID	GETID @GETID	286	 S: 变量或地址 D1: ID代码 D2: 目的字	输出指定变量或地址的 FINS 命令变量类型(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的已分配地址。	输出	需要
步激活	SA @SA	784		激活子图中指定的步, 以开始执行相应的操作。	输出	需要
步停止	SE @SE	785		停止子图中指定的步, 以结束执行相应的操作。	输出	需要
读设定定时器	TSR @TSR	780		由 S 指定的步定时器的当前值存储在从 D 开始的位置。	输出	需要
设定步定时器	TSW @TSW	781		将由 S 指定的步定时器的当前值改为由从 D 开始的位置所指定的值。	输出	需要
SFC ON	SFCON	789	 N: SFC任务号	重新开始执行被另一个 SFC 任务控制指令之一结束或暂停的 SFC 任务。	输出	需要
SFC OFF	SFCOFF	790	 N: SFC任务号	结束 SFC 任务的执行, 保持所有输出的状态。当重新开始执行 SFC 任务时, 将从第一步开始执行。	输出	需要
SFC 带复位暂停	SFCPR	793	 N: SFC任务号	暂停 SFC 任务的执行, 将使所有输出的状态复位。	输出	需要
SFC 不带复位暂停	SFCPRN	791	 N: SFC任务号	暂停 SFC 任务的执行, 保持所有输出的状态。	输出	需要

A-2 指令执行时间和步数

下表列出了 CPU 单元所支持的所有指令的执行时间。

一个用户程序中指令的总执行时间为计算循环时间时的程序执行的处理时间 (*1)。

*1 将可在循环任务中和满足中断条件的中断任务中执行的任务分配给用户程序。

大多数指令的执行时间因所使用的 CPU 单元和执行指令时的条件而异。

当执行条件为 OFF 时，执行时间也会有所不同。

下表还在长度 (步数) 栏中列出了每条指令的长度。各指令在用户程序区中所需的步数取决于指令和指令中所使用的操作数。

程序中的步数不等于指令的条数。

注 1 支持大多数指令的微分形式 (用 ↑、↓、@ 和 % 表示)。
指定为微分形式将会使执行时间延迟下表中列出的量。

(单位: μs)

符号	CJ2 CPU 单元	
	CJ2H-CPU6□(-EIP)	CJ2M-CPU□□
↑或↓	+0.24	+0.32
@或%	+0.24	+0.32

2 不执行指令时，请使用下列时间作为参考。

(单位: μs)

CJ2 CPU 单元	
CJ2H-CPU6□(-EIP)	CJ2M-CPU□□
0.016	0.020

执行时间因指令的执行顺序和指令执行的条件而异。请用以下公式计算最差的假设情况下的最大偏差量。

CJ2H CPU 单元: 指令步数 × 0.016(μs)

CJ2M CPU 单元: 指令步数 × 0.020(μs)

A-2-1 顺序输入指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
载入	LD	---	1 ~ 4*	0.016	0.040	---
	!LD	---	2 ~ 16*	0.99	1.260	---
载入非	LD NOT	---	1 ~ 4*	0.016	0.040	---
	!LD NOT	---	2 ~ 16*	0.99	1.260	---
与	AND	---	1 ~ 4*	0.016	0.040	---
	!AND	---	2 ~ 16*	0.99	1.260	---
与非	AND NOT	---	1 ~ 4*	0.016	0.040	---
	!AND NOT	---	2 ~ 16*	0.99	1.260	---
或	OR	---	1 ~ 4*	0.016	0.040	---
	!OR	---	2 ~ 16*	0.99	1.260	---
或非	OR NOT	---	1 ~ 4*	0.016	0.040	---
	!OR NOT	---	2 ~ 16*	0.99	1.260	---
逻辑块与	AND LD	---	1	0.016	0.040	---
逻辑块或	OR LD	---	1	0.016	0.040	---
非	NOT	520	1	0.016	0.040	---
条件 ON	UP	521	3	0.26	0.36	---
条件 OFF	DOWN	522	4	0.27	0.40	---
载入位测试	LD TST	350	4	0.11	0.16	---
载入位测试取反	LD TSTN	351	4	0.11	0.16	---
与位测试	AND TST	350	4	0.11	0.16	---
与位测试取反	AND TSTN	351	4	0.11	0.16	---
或位测试	OR TST	350	4	0.11	0.16	---
或位测试取反	OR TSTN	351	4	0.11	0.16	---

* 除考虑下列偏差的情况外，上述指令的长度均为 1 步。

- 微分指令：+2 步
- 指定 EM 区：+1 步
- 指定即时刷新：+1 步
- 指定即时刷新和 CIO 0.00 ~ CIO 999.15：+12 步

A-2-2 顺序输出指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
输出	OUT	---	1 ~ 2	0.016	0.040	---
	!OUT	---	2 ~ 14	0.99	1.320	---
反相输出	OUT NOT	---	1 ~ 2	0.016	0.040	---
	!OUT NOT	---	2 ~ 14	0.99	1.320	---
保持	KEEP	011	1 ~ 2	0.048	0.060	---
	!KEEP	011	16	0.99	1.340	---
上升沿微分	DIFU	013	2 ~ 2	0.28	0.30	---
下降沿微分	DIFD	014	2 ~ 2	0.24	0.30	---
置位	SET	---	1 ~ 2	0.016	0.040	---
	!SET	---	2 ~ 14	0.99	1.360	---
复位	RSET	---	1 ~ 2	0.016	0.040	---
	!RSET	---	2 ~ 14	0.99	1.360	---
多个位置位	SETA	530	4	3.68	4.12	对 1 个位置位
				15.5	24.4	对 1,000 个位置位
多个位复位	RSTA	531	4	3.7	4.1	对 1 个位复位
				15.5	24.4	对 1,000 个位复位
单个位置位	SETB	532	2	0.19	0.280	---
	!SETB	---	16	0.99	1.120	---
单个位复位	RSTB	534	2	0.19	0.280	---
	!RSTB	---	16	0.99	1.120	---
单个位输出	OUTB	534	2	0.19	0.280	---
	!OUTB	---	16	0.99	1.180	---

A-2-3 顺序控制指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
结束	END	001	1	2.6	3.5	---
空操作	NOP	000	1	0.016	0.040	---
互锁	IL	002	1	0.048	0.060	---
互锁清除	ILC	003	1	0.048	0.060	---
多路互锁微分保持	MILH	517	3	2.3	3.3	不满足互锁条件(输入条件 ON)
				3.4	4.6	满足互锁条件(输入条件 OFF)
				3.8	5.2	互锁期间再次满足互锁条件(输入条件 OFF)
多路互锁微分释放	MILR	518	3	2.3	3.1	不满足互锁条件(输入条件 ON)
				3.4	4.5	满足互锁条件(输入条件 OFF)
				3.8	5.1	互锁期间再次满足互锁条件(输入条件 OFF)
多路互锁清除	MILC	519	2	1.2	1.7	未处于互锁期间
				1.6	2.2	处于互锁期间
跳转	JMP	004	2	0.31	0.34	---
跳转结束	JME	005	2	---	---	---
条件跳转	CJP	510	2	0.31	0.34	满足跳转条件(输入条件 ON)
条件跳转非	CJPN	511	2	0.31	0.34	满足跳转条件(输入条件 OFF)
多路跳转	JMPO	515	1	0.048	0.060	---
多路跳转结束	JME0	516	1	0.048	0.060	---
FOR 循环	FOR	512	2	0.27	0.42	指定一个常数
循环中断	BREAK	514	1	0.048	0.060	---
NEXT 循环	NEXT	513	1	0.14	0.16	继续循环时
				0.18	0.18	结束循环时

A-2-4 定时器和计数器指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
100ms 定时器	TIM	---	3	0.67	0.84	---
	TIMX	550		0.67	0.84	
10ms 定时器	TIMH	015	3	0.67	0.84	---
	TIMHX	551		0.67	0.84	
1ms 定时器	TMHH	540	3	0.67	0.84	---
	TMHHX	552		0.67	0.84	
0.1ms 定时器	TIMU	541	3	0.67	0.84	---
	TIMUX	556		0.67	0.84	
0.01ms 定时器	TMUH	544	3	0.67	0.84	---
	TMUHX	557		0.67	0.84	
累加定时器	TTIM	087	3	9.2	12.1	---
				6.9	8.4	复位时
				5.0	6.5	互锁时
	TTIMX	555	3	8.8	11.7	---
				6.8	8.5	复位时
				5.0	6.5	互锁时
长定时器	TIML	542	4 ~ 5	5.8	7.0	---
				3.9	4.1	互锁时
	TIMLX	553	4 ~ 5	5.7	7.0	---
				3.6	3.7	互锁时
多路输出定时器	MTIM	543	4	6.4	7.2	---
				3.7	4.3	复位时
	MTIMX	554	4	5.5	6.4	---
				3.4	3.8	复位时
定时器复位	TRSET	549	2	0.58	0.8	---
计数器	CNT	---	3	0.51	0.58	---
	CNTX	546		0.51	0.58	
可逆计数器	CNTR	012	3	9.1	11.8	---
	CNTRX	548		8.0	10.3	
复位定时器 / 计数器	CNR	545	3	4.8	5.4	复位 1 个字时
				2839	2555	复位 1,000 个字时
	CNRX	547	3	4.7	5.5	复位 1 个字时
				2839	2555	复位 1,000 个字时

A-2-5 比较指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
输入比较指令 (无符号)	=	300	4	0.08	0.16	---
	<>	305				
	<	310				
	<=	315				
	>	320				
	>=	325				
输入比较指令 (双字、无符号)	=L	301	4 ~ 6	0.08	0.24	---
	<>L	306				
	<L	311				
	<=L	316				
	>L	321				
	>=L	326				
输入比较指令 (带符号)	=S	302	4	0.08	0.16	---
	<>S	307				
	<S	312				
	<=S	317				
	>S	322				
	>=S	327				
输入比较指令 (双字、带符号)	=SL	303	4 ~ 6	0.08	0.24	---
	<>SL	308				
	<SL	313				
	<=SL	318				
	>SL	323				
	>=SL	328				
时间比较指令	=DT	341	4	16.300	27.9	---
	<>DT	342				
	<DT	343				
	<=DT	344				
	>DT	345				
	>=DT	346				
比较	CMP	020	3	0.06	0.080	---
	!CMP	020	30	2.06	2.6	---
双字比较	CMPL	060	3 ~ 5	0.064	0.120	---
带符号二进制比较	CPS	114	3 ~ 5	0.064	0.080	---
	!CPS	114	30	2.06	2.6	---
带符号双字二进制比较	CPSL	115	3 ~ 5	0.064	0.120	---
表比较	TCMP	085	4	10.3	12.5	---
多路比较	MCMP	019	4	15.2	20.3	---
无符号块比较	BCMP	068	4	16.3	20.5	---
扩展块比较	BCMP2	502	4	5.0	5.1	数据的字数: 1
				217.2	278	数据的字数: 255
区域范围比较	ZCP	088	3	0.14	0.400	---

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
双字区域范围比较	ZCPL	116	3 ~ 5	0.14	0.640	---
带符号区域范围比较	ZCPS	117	3	0.14	0.400	---
带符号双字区域范围比较	ZCPSL	118	3 ~ 5	0.14	0.640	---

A-2-6 数据传送指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
传送	MOV	021	3	0.05	0.12	---
	!MOV	021	30	1.98	2.6	---
双字传送	MOVL	498	3 ~ 4	0.05	0.20	---
传送反	MVN	022	3	0.05	0.12	---
双字传送反	MVNL	499	3 ~ 4	0.05	0.20	---
位传送	MOVB	082	4	0.19	0.32	---
数位传送	MOVD	083	4	0.19	0.32	---
多位传送	XFRB	062	4	6.6	9.4	传送 1 位
				85.8	119	传送 255 位
块传送	XFER	070	4	0.29	0.28	传送 1 个字
				240.1	220	传送 1,000 个字
块设置	BSET	071	4	0.21	0.20	设置 1 个字
				142.2	140	设置 1,000 个字
数据交换	XCHG	073	3	0.32	0.48	---
双字数据交换	XCGL	562	3 ~ 4	0.12	0.29	---
单字分配	DIST	080	4	4.5	4.7	---
数据收集	COLL	081	4	4.6	4.7	---
传送至寄存器	MOVR	560	3	0.064	0.200	---
传送定时器 / 计数器的当前值 (PV) 至寄存器	MOVRW	561	3	0.064	0.200	---

A-2-7 数据移位指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
移位寄存器	SFT	010	3	2.86	3.47	使 1 个字移位
				315	422	使 1,000 个字移位
可逆移位寄存器	SFTR	084	4	6.22	6.38	使 1 个字移位
				319	422	使 1,000 个字移位
异步移位寄存器	ASFT	017	4	5.3	6.3	使 1 个字移位
				948	1285	使 1,000 个字移位 ^{*1}
字移位	WSFT	016	4	2.3	3.1	使 1 个字移位
				233	187	使 1,000 个字移位
算术左移	ASL	025	2	0.18	0.260	---
双字左移	ASLL	570	2	0.32	0.420	---
算术右移	ASR	026	2	0.18	0.260	---
双字右移	ASRL	571	2	0.32	0.420	---
循环左移	ROL	027	2	0.18	0.260	---
双字循环左移	ROLL	572	2	0.32	0.420	---
无进位循环左移	RLNC	574	2	0.18	0.260	---
无进位双字循环左移	RLNL	576	2	0.32	0.420	---
循环右移	ROR	028	2	0.18	0.260	---
双字循环右移	RORL	573	2	0.32	0.420	---
无进位循环右移	RRNC	575	2	0.18	0.260	---
无进位双字循环右移	RRNL	577	2	0.32	0.420	---
一个数位左移	SLD	074	3	3.7	4.4	使 1 个字移位
				317.8	429	使 1,000 个字移位
一个数位右移	SRD	075	3	4.5	5.4	使 1 个字移位
				479.5	656	使 1,000 个字移位
N 位数据左移	NSFL	578	4	4.6	5.2	使 1 个位移位
				31.5	36.1	使 1,000 个位移位
N 位数据右移	NSFR	579	4	4.5	5.2	使 1 个位移位
				39.0	50.2	使 1,000 个位移位
左移 N 位	NASL	580	3	0.18	0.38	---
双字左移 N 位	NSLL	582	3	0.32	0.54	---
右移 N 位	NASR	581	3	0.18	0.38	---
双字右移 N 位	NSRL	583	3	0.32	0.54	---

*1 指令执行时间受数据量的影响极大。这将影响循环时间。为减小对循环时间的影响，可指定后台执行。详情请参考“10-2-5 后台执行”。

A-2-8 递增 / 递减指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
二进制递增	++	590	2	0.18	0.24	---
双字二进制递增	++L	591	2	0.18	0.24	---
二进制递减	--	592	2	0.18	0.24	---
双字二进制递减	--L	593	2	0.18	0.24	---
BCD 递增	++B	594	2	3.0	3.4	---
双字 BCD 递增	++BL	595	2	3.2	3.5	---
BCD 递减	--B	596	2	3.0	3.5	---
双字 BCD 递减	--BL	597	2	3.2	3.5	---

A-2-9 四则运算指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
无进位带符号二进制加	+	400	4	0.18	0.34	---
无进位带符号双字二进制加	+L	401	4 ~ 6	0.18	0.24	---
有进位带符号二进制加	+C	402	4	0.18	0.34	---
有进位带符号双字二进制加	+CL	403	4 ~ 6	0.18	0.24	---
无进位 BCD 加	+B	404	4	4.0	4.8	---
无进位双字 BCD 加	+BL	405	4 ~ 6	4.9	6.0	---
有进位 BCD 加	+BC	406	4	4.4	5.2	---
有进位双字 BCD 加	+BCL	407	4 ~ 6	5.2	6.6	---
无借位带符号二进制减	-	410	4	0.18	0.340	---
无借位带符号双字二进制减	-L	411	4 ~ 6	0.18	0.24	---
有借位带符号二进制减	-C	412	4	0.18	0.340	---
有借位带符号双字二进制减	-CL	413	4 ~ 6	0.18	0.24	---
无借位 BCD 减	-B	414	4	4.1	4.9	---
无借位双字 BCD 减	-BL	415	4 ~ 6	4.9	5.9	---
有借位 BCD 减	-BC	416	4	4.5	5.2	---
有借位双字 BCD 减	-BCL	417	4 ~ 6	5.2	6.3	---
带符号二进制乘	*	420	4	0.26	0.520	---
带符号双字二进制乘	*L	421	4 ~ 6	3.6	3.9	---
无符号二进制乘	*U	422	4	0.26	0.26	---
无符号双字二进制乘	*UL	423	4 ~ 6	3.6	3.9	---
BCD 乘	*B	424	4	3.6	4.6	---
双字 BCD 乘	*BL	425	4 ~ 6	4.9	6.2	---
带符号二进制除	/	430	4	0.29	0.540	---
带符号双字二进制除	/L	431	4 ~ 6	4.2	4.8	---
无符号二进制除	/U	432	4	0.29	0.540	---

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
无符号双字二进制除	/UL	433	4 ~ 6	3.8	4.2	---
BCD 除	/B	434	4	5.0	5.9	---
双字 BCD 除	/BL	435	4 ~ 6	4.8	5.9	---

A-2-10 转换指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
BCD → 二进制	BIN	023	3	0.18	0.280	---
双字 BCD → 双字二进制	BINL	058	3 ~ 4	3.3	3.5	---
二进制 → BCD	BCD	024	3	0.19	0.300	---
双字二进制 → 双字 BCD	BCDL	059	3 ~ 4	3.3	3.7	---
2 进制求补	NEG	160	3	0.14	0.240	---
双字 2 进制求补	NEGL	161	3 ~ 4	0.26	0.440	---
16 位 → 32 位带符号二进制	SIGN	600	3	0.26	0.340	---
数据译码	MLPX	076	4	0.17	0.280	对 1 个数位译码 (4 ~ 16)
				0.42	0.770	对 4 个数位译码 (4 ~ 16)
				1.14	1.760	对 1 个数位译码 (8 ~ 256)
				2.17	3.370	对 4 个数位译码 (8 ~ 256)
数据编码	DMPX	077	4	3.3	4.6	对 1 个数位编码 (16 ~ 4)
				3.7	5.2	对 4 个数位编码 (16 ~ 4)
				17.3	26.3	对 1 个数位编码 (256 ~ 8)
				35	47	对 2 个数位编码 (256 ~ 8)
ASCII 转换	ASC	086	4	4.0	4.5	将 1 个数位转换成 ASCII 码
				4.6	5.2	将 4 个数位转换成 ASCII 码
ASCII → 十六进制	HEX	162	4	3.3	3.8	转换 1 个数位
列 → 行	LINE	063	4	10.5	13.1	---
行 → 列	COLM	064	4	13.8	17.6	---
带符号 BCD → 二进制	BINS	470	4	3.6	4.0	0 号数据格式设定
				3.6	4.0	1 号数据格式设定
				3.6	4.0	2 号数据格式设定
				3.6	4.0	3 号数据格式设定

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
带符号双字 BCD→二进制	BISL	472	4 ~ 5	3.7	4.1	0 号数据格式设定
				3.6	4.1	1 号数据格式设定
				3.7	4.2	2 号数据格式设定
				3.7	4.2	3 号数据格式设定
带符号二进制→BCD	BCDS	471	4	3.7	4.0	0 号数据格式设定
				3.7	4.1	1 号数据格式设定
				3.7	4.2	2 号数据格式设定
				3.7	4.2	3 号数据格式设定
带符号双字二进制→BCD	BDSL	473	4 ~ 5	4.0	4.5	0 号数据格式设定
				4.0	4.6	1 号数据格式设定
				4.0	4.6	2 号数据格式设定
				4.1	4.6	3 号数据格式设定
格雷码转换	GRY	474	4	26.5	49.1	8 位二进制
				27.6	51.1	8 位 BCD
				30.9	57.2	8 位角度
				35.3	66.0	15 位二进制
				36.3	68.0	15 位 BCD
				39.6	74.0	15 位角度
格雷码→二进制转换	GRAY_BIN	478	3	0.1	0.3	---
双字格雷码→二进制	GRAY_BINL	479	3 ~ 4	0.1	0.4	---
二进制→格雷码转换	BIN_GRAY	480	3	0.1	0.3	---
双字二进制→格雷码转换	BIN_GRAYL	481	3 ~ 4	0.1	0.4	---
4 位数→ASCII	STR4	601	3	8.4	14.2	---
8 位数→ASCII	STR8	602	3 ~ 4	10.2	16.4	---
16 位数→ASCII	STR16	603	3	15.8	28.2	---
ASCII→4 位数	NUM4	604	3 ~ 4	10.5	18.5	---
ASCII→8 位数	NUM8	605	3	14.8	27.1	---
ASCII→16 位数	NUM16	606	3	27.4	52.0	---

A-2-11 逻辑指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
逻辑与	ANDW	034	4	0.14	0.340	---
双字逻辑与	ANDL	610	4 ~ 6	0.26	0.640	---
逻辑或	ORW	035	4	0.18	0.340	---
双字逻辑或	ORWL	611	4 ~ 6	0.26	0.640	---
异或	XORW	036	4	0.18	0.340	---
双字异或	XORL	612	4 ~ 6	0.26	0.640	---
异或非	XNRW	037	4	0.18	0.340	---
双字异或非	XNRL	613	4 ~ 6	0.26	0.640	---
求补	COM	029	2	0.18	0.240	---
双字求补	COML	614	2	0.32	0.440	---

A-2-12 特殊算术指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
二进制平方根	ROTB	620	3	15.4	24.2	---
BCD 平方根	ROOT	072	3	17.1	25.3	---
算术处理	APR	069	4	4.6	5.3	指定 SIN 和 COS
				5.7	6.9	指定线段接近
浮点数除	FDIV	079	4	76	149	---
位计数器	BCNT	067	4	0.24	0.360	对 1 个字计数

A-2-13 浮点算术运算指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
浮点数→16位	FIX	450	3 ~ 4	0.13	0.24	---
浮点数→32位	FIXL	451	3 ~ 4	0.13	0.32	---
16位→浮点数	FLT	452	3 ~ 4	0.13	0.30	---
32位→浮点数	FLTL	453	3 ~ 4	0.13	0.32	---
浮点数加	+F	454	4 ~ 6	0.24	0.66	---
浮点数减	-F	455	4 ~ 6	0.24	0.66	---
浮点数除	/F	457	4 ~ 6	0.4	0.9	---
浮点数乘	*F	456	4 ~ 6	0.24	0.66	---
度→弧度	RAD	458	3 ~ 4	2.7	3.3	---
弧度→度	DEG	459	3 ~ 4	3.0	3.2	---
正弦	SIN	460	3 ~ 4	3.8	4.3	指定了 0°
				4.5	5.4	指定了 45°
				5.0	6.0	指定了 90°
高速正弦	SINQ	475	8 ~ 9	0.59	0.86	指定了 0°、45° 或 90°
余弦	COS	461	3 ~ 4	3.7	4.3	指定了 0°
				4.4	5.2	指定了 45°
				5.3	6.7	指定了 90°
高速余弦	COSQ	476	8 ~ 9	0.59	0.86	指定了 0°、45° 或 90°
正切	TAN	462	3 ~ 4	3.9	4.5	指定了 0°
				6.1	8.2	指定了 45°
高速正切	TANQ	477	15 ~ 16	1.2	1.7	指定了 0°、45° 或 90°
反正弦	ASIN	463	3 ~ 4	5.8	7.1	指定了 0°
				24.8	33.0	指定了 45°
				5.6	7.0	指定了 90°
反余弦	ACOS	464	3 ~ 4	5.3	6.8	指定了 0°
				27.2	34.6	指定了 45°
				6.4	7.1	指定了 90°
反正切	ATAN	465	3 ~ 4	4.0	5.0	指定了 0°
				5.6	7.0	指定了 45°
平方根	SQRT	466	3 ~ 4	0.42	0.66	---
指数	EXP	467	3 ~ 4	3.8	4.5	---
对数	LOG	468	3 ~ 4	5.8	6.5	---
指数幂	PWR	840	4 ~ 6	35.7	56.6	---

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
浮点符号比较	=F	329	3 ~ 5	0.13	0.26	---
	<>F	330				
	<F	331				
	<=F	332				
	>F	333				
	>=F	334				
浮点数→ASCII	FSTR	448	4 ~ 5	15.6	23.9	---
ASCII→浮点数	FVAL	449	3	21.2	31.4	---
浮点数传送 (单精度)	MOVF	469	3 ~ 4	0.18	0.20	---

A-2-14 双精度浮点数指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
双精度符号比较	=D	335	3	5.1	6.7	---
	<>D	336				
	<D	337				
	<=D	338				
	>D	339				
	>=D	340				
双精度浮点数→16位二进制数	FIXD	841	3	5.1	5.4	---
双精度浮点数→32位二进制数	FIXLD	842	3	5.1	5.4	---
16位二进制数→双精度浮点数	DBL	843	3	3.5	4.3	---
32位二进制数→双精度浮点数	DBLL	844	3	3.5	4.3	---
双精度浮点数加	+D	845	4	6.0	7.1	---
双精度浮点数减	-D	846	4	6.1	7.1	---
双精度浮点数乘	*D	847	4	6.1	7.1	---
双精度浮点数除	/D	848	4	6.4	7.5	---
双精度度→弧度	RADD	849	3	6.1	6.5	---
双精度弧度→度	DEGD	850	3	6.0	6.4	---
双精度正弦	SIND	851	3	14.7	21.5	指定了 0°
				20.4	35.4	指定了 45°
				18.5	35.0	指定了 90°
双精度余弦	COSD	852	3	14.1	20.6	指定了 0°
				19.6	29.9	指定了 45°
				19.1	29.8	指定了 90°
双精度正切	TAND	853	3	7.3	9.4	指定了 0°
				27.4	50.3	指定了 45°

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
双精度反正弦	ASIND	854	3	7.5	9.8	指定了 0°
				55.0	75.2	指定了 45°
				6.1	8.3	指定了 90°
双精度反余弦	ACOSD	855	3	8.3	10.9	指定了 0°
				55.9	72.8	指定了 45°
				43.7	72.8	指定了 90°
双精度反正切	ATAND	856	3	6.1	7.4	指定了 0°
				29.7	36.5	指定了 45°
双精度平方根	SQRTD	857	3	16.6	23.4	---
双精度指数	EXPD	858	3	39.7	58.4	---
双精度对数	LOGD	859	3	35.5	52.2	---
双精度指数幂	PWRD	860	4	66	99	---

A-2-15 表格数据处理指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
设置堆栈	SSET	630	3	7.6	9.4	指定堆栈区中的 5 个字
				107	65	指定堆栈区中的 1,000 个字
压入栈	PUSH	632	3	4.9	5.9	---
先进先出	FIFO	633	3	4.8	5.0	指定堆栈区中的 5 个字
				231	167	指定堆栈区中的 1,000 个字
后进先出	LIFO	634	3	5.3	7.1	---
定义记录表	DIM	631	5	11.1	19.7	---
设置记录位置	SETR	635	4	3.8	5.5	---
获取记录号	GETR	636	4	4.6	7.9	---
数据搜索	SRCH	181	4	13.9	25.0	搜索 1 个字
				1940	3257	搜索 1,000 个字*1
交换字节	SWAP	637	3	10.1	17.5	交换 1 个字
				1421	2098	交换 1,000 个字*1

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
寻找最大值	MAX	182	4 ~ 5	4.8	5.8	要搜索的值的个数: 1
				465	672	要搜索的值的个数: 1,000 ^{*1}
寻找双字长最大值	MAXL		4 ~ 5	4.8	5.8	要搜索的值的个数: 1
				465	773	要搜索的值的个数: 1,000 ^{*1}
寻找浮点数最大值	MAXF		4 ~ 5	5.2	6.5	要搜索的值的个数: 1
				682	1090	要搜索的值的个数: 1,000 ^{*1}
寻找双精度浮点数 最小值	MAXD		4 ~ 5	5.4	6.4	要搜索的值的个数: 1
				1435	2333	要搜索的值的个数: 1,000 ^{*1}
寻找最小值	MIN	183	4 ~ 5	4.8	5.8	要搜索的值的个数: 1
				465	677	要搜索的值的个数: 1,000 ^{*1}
寻找双字长最小值	MINL		4 ~ 5	4.8	5.9	要搜索的值的个数: 1
				189	774	要搜索的值的个数: 1,000 ^{*1}
寻找浮点数最小值	MINF		4 ~ 5	5.2	6.5	要搜索的值的个数: 1
				683	1091	要搜索的值的个数: 1,000 ^{*1}
寻找双精度浮点数 最小值	MIND		4 ~ 5	5.2	6.4	要搜索的值的个数: 1
				1402	2303	要搜索的值的个数: 1,000 ^{*1}
SUM	SUM	184	4	17.5	31.3	加 1 个字
				900	1696	加 1,000 个字 ^{*1}
帧校验和	FCS	180	4	14.1	25.2	对于 1 个字的表长度
				1235	2089	对于 1,000 个字的表长 度 ^{*1}
读取栈大小	SNUM	638	3	4.5	5.3	---
读取栈数据	SREAD	639	4	4.6	5.4	---
覆盖栈数据	SWRIT	640	4	4.3	5.0	---
插入栈数据	SINS	641	4	8.2	9.3	---
				275	256	对于 1,000 个字的表
删除栈数据	SDEL	642	4	6.1	7.8	---
				247	180	对于 1,000 个字的表

*1 指令执行时间受数据量的影响极大。这将影响循环时间。为减小对循环时间的影响，可指定后台执行。详情请参考“10-2-5 后台执行”。

A-2-16 跟踪指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
无符号单字记录搜索指令	RSRCH	360 ~ 364	6	13.9	15.9	记录数: 1
				504	585	记录数: 1,000
无符号双字记录搜索指令	RSRCH2	370 ~ 374	6	14.7	17.6	记录数: 1
				838	932	记录数: 1,000
无符号四字记录搜索指令	RSRCH4	380 ~ 384	6	17.0	19.1	记录数: 1
				1544	1684	记录数: 1,000
无符号单字记录排序	RSORT	203	5	149	156	100条记录, 禁止拆分排序, 将“99, 98, 97...0”排序成“0, 1, 2...99”(最糟糕的情况)
无符号双字记录排序	RSORT2	204	5	250	249	
无符号四字记录排序	RSORT4	205	5	457	440	

A-2-17 数据控制指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
PID 控制	PID	190	4	297	526	初次执行
				234	423	输入 ON 并采样
				71	117	输入 ON 但不采样
				7.2	10.5	输入 OFF
带自整定的 PID 控制	PIDAT	191	4	302	600	初次执行
				237	428	输入 ON 并采样
				73	118	输入 ON 但不采样
				7.3	10.5	输入 OFF
				120	203	初次执行自整定
限位控制	LMT	680	4 ~ 5	10.8	18.3	---
静带控制	BAND	681	4 ~ 5	11.2	19.2	---
静域控制	ZONE	682	4 ~ 5	10.9	17.7	---
时间比例输出	TPO	685	4	6.9	10.2	输入 OFF
				37	65	输入 ON 且指定了占空比或者禁止了输出限位
定标	SCL	194	4	7.6	9.3	---
定标 2	SCL2	486	4	6.8	9.2	---
定标 3	SCL3	487	4	7.8	9.9	---
平均值	AVG	195	4	22	40	1 次运算的平均值
				212	351	64 次运算的平均值

A-2-18 子程序指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
子程序调用	SBS	091	2	0.90	2.8	---
子程序入口	SBN	092	2	2.8	4.1	---
子程序返回	RET	093	1	0.43	2.0	---
宏	MCRO	099	4	16.8	21.7	---
全局子程序返回	GSBS	750	2	0.90	2.8	---
全局子程序调用	GSBN	751	2	2.7	3.6	---
全局子程序入口	GRET	752	1	0.43	2.0	---

A-2-19 中断控制指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
设置中断屏蔽	MSKS	690	3	10.6	22.1	---
读中断屏蔽	MSKR	692	3	9.6	14.8	---
清除中断	CLI	691	3	10.1	21.6	---
禁止中断	DI	693	1	10.3	20.4	---
允许中断	EI	694	1	9.3	16.0	---

A-2-20 高速计数器 / 脉冲输出指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
模式控制	INI	880	4	---	7.4	启动高速计数器比较
					4.0	停止高速计数器比较
					9.9	变更脉冲输出当前值 (PV)
					8.0	变更高速计数器当前值 (PV)
					5.0	在计数器模式下变更中断输入的当前值 (PV)
					9.2	停止脉冲输出
					5.2	停止 PWM 输出
					48.2	变更原点搜索 / 返回设定

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
读取脉冲 PV 值	PRV	881	4	---	7.6	读入脉冲输出当前值 (PV)
					6.3	读入高速计数器状态
					3.7	在计数器模式下读入中断输入的当前值 (PV)
					7.1	读入脉冲输出状态
					6.3	读入高速计数器状态
					4.1	读入 PWM 状态
					17.5	读入 8 个范围的高速计数器范围比较结果
					34.6	读入 32 个范围的高速计数器范围比较结果
					4.9	读入高速计数器 0 的频率
计数器频率转换	PRV2	883	4	---	5.2	---
寄存器比较表	CTBL	882	4	---	31.7	注册目标值表并启动 1 个目标值的比较
					1528	注册目标值表并启动 48 个目标值的比较
					45.0	注册范围表并启动 8 个范围的比较
					150.8	注册范围表并启动 32 个范围的比较
					26.2	只为 1 个目标值注册目标值表, 但不启动比较
					1520	只为 48 个目标值注册目标值表, 但不启动比较
					41.8	注册 8 个范围的范围比较表, 但不启动比较
					150.0	注册 32 个范围的范围比较表, 但不启动比较
速度输出	SPED	885	4	---	23.5	连续模式
					24.6	单独模式
设置脉冲	PULS	886	4	---	8.2	---
定位	PLS2	887	5	---	39.4	---

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
加速控制	ACC	888	4	---	35.0	连续模式
					44.6	单独模式
原点搜索	ORG	889	3	---	28.1	原点搜索
					23.6	原点返回
PWM 输出	PWM	891	4	---	7.8	---
中断进给	!FEED	892	4	---	42.4	---

A-2-21 步指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
步定义	STEP	008	2	8.7	10.6	步控制位 ON
				8.7	9.8	步控制位 OFF
步启动	SNXT	009	2	2.2	2.8	---

A-2-22 基本 I/O 单元指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
I/O 刷新	IORF	097	3	10.1	12.2	对基本 I/O 单元进行 1 个字刷新 (IN)
				10.5	13.0	对基本 I/O 单元进行 1 个字刷新 (OUT)
高性能 I/O 单元 I/O 刷新	FIORF	225	2	*1	*1	---
CPU 总线 I/O 刷新	DLNK	226	4	234	256	分配了 1 个字
7 段译码	SDEC	078	4	2.5	3.3	---
数字开关输入	DSW	210	6	24.8	39.6	4 个数位, 数据输入值: 0
				24.8	40.2	8 个数位, 数据输入值: 00
十进制数按键输入	TKY	211	4	7.2	9.7	数据输入值: 00
				6.5	8.6	数据输入值: FF
十六进制数按键输入	HKY	212	5	25.9	40.8	数据输入值: 00
				25.9	41.0	数据输入值: FF
矩阵输入	MTR	213	5	25.0	38.5	数据输入值: 00
				25.0	38.5	数据输入值: FF
7 段显示输出	7SEG	214	5	31.4	51.9	4 个数位
				34.6	59.4	8 个数位
模拟量输入直接转换 (适用于 CJ1W-AD042)	AIDC	216	3	25.0	27.0	模拟量输入个数: 1, 使用的模拟量输入个数: 4
				38.8	41.6	模拟量输入个数: 0, 使用的模拟量输入个数: 4
模拟量输出直接转换 (适用于 CJ1W-DA042V)	AODC	217	3	23.1	24.4	模拟量输出个数: 1, 使用的模拟量输出个数: 4
				44.1	45.3	模拟量输出个数: 0, 使用的模拟量输出个数: 4

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
PCU 高速定位 (仅限 CJ1W-NC□□4 或 CJ1W-NC□81)	NCDMV	218	4	81.7	95.3	---
PCU 定位触发器 (仅限 CJ1W-NC□81)	NCDTR	219	3	22.9	25.5	---
智能 I/O 读	IORD	222	4	*1	*1	---
智能 I/O 写	IOWR	223	4	*1	*1	---

*1 IORD、IOWR 和 FIORF 指令的执行情况取决于执行这些指令的高功能 I/O 单元。

A-2-23 串行通信指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
协议宏	PMCR	260	5	57.8	97.8	直接指定
				77	132	操作数规定, 发送 1 个字, 接收 1 个字
协议宏 2	PMCR2	264	6	49.5	96.0	直接指定
				69	129	操作数规定, 发送 1 个字, 接收 1 个字
发送	TXD	236	4	57.5	93.8	发送 1 个字节
				517	947	发送 256 个字节
接收	RXD	235	4	79	128	存储 1 个字节
				570	1033	存储 256 个字节
通过串行通信单元 / 串行端口直接传送 (仅限 CJ1W-SCU22、CJ1W-SCU32 或 CJ1W-SCU42)	TXDU	256	4	75	130	发送 1 个字节
通过串行通信单元 / 串行端口直接接收 (仅限 CJ1W-SCU22、CJ1W-SCU32 或 CJ1W-SCU42)	RXDU	255	4	74	128	存储 1 个字节
通过串行通信单元直接发送	DTXDU	262	4	25.8	37.0	发送 1 个字节
				179	203	发送 256 个字节
通过串行通信单元直接接收	DRXDU	261	4	27.8	39.7	存储 1 个字节
				188	205	存储 256 个字节
修改串行端口设定	STUP	237	3	233	276	寻址到 CPU 单元上的 COM 端口

A-2-24 网络指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
网络发送	SEND	090	4	44.3	79.4	---
网络发送 2	SEND2	491	5	43.4	82.8	---
网络接收	RECV	098	4	43.9	79.9	---
网络接收 2	RECV2	492	5	44.5	82.8	---
发布命令	CMND	490	4	52.7	95.1	---
发布命令 2	CMND2	493	5	53.0	98.1	---
发送显式报文	EXPLT	720	4	78	134	---
显式获取属性	EGATR	721	4	74	127	---
显式设定属性	ESATR	722	3	69	117	---
显式字读取	ECHRD	723	4	65	110	---
显式字写入	ECHWR	724	4	64	110	---

A-2-25 文件存储指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
读数据文件	FREAD	700	5	217	372	二进制格式
写数据文件	FWRIT	701	5	216	366	二进制格式
写文本文件	TWRIT	704	5	205	370	---

A-2-26 显示指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
显示信息	MSG	046	3	6.9	10.5	显示信息
				6.6	9.5	删除显示的信息

A-2-27 时钟指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
日历加	CADD	730	4	15.6	22.5	---
日历减	CSUB	731	4	16.4	24.9	---
小时→秒	SEC	065	3	3.6	4.1	---
秒→小时	HMS	066	3	3.5	4.0	---
时钟调整	DATE	735	2	29.6	53.2	---

A-2-28 调试指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
跟踪存储器采样	TRSM	045	1	8.9	12.6	对 1 个位和 0 个字进行采样
				31.6	33.1	对 31 个位和 6 个字进行采样
				38.8	39.2	对 31 个位和 16 个字进行采样

A-2-29 故障诊断指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
故障报警	FAL	006	3	7.9	14.7	记录错误
				14.7	22.3	删除错误 (按优先级的顺序)
				12.9	22.5	删除错误 (所有错误)
				117	210	删除错误 (单个删除)
严重故障报警	FALS	007	3	---	---	---
故障点检测	FPD	269	4	111	188	位地址输出, 时间受监控
				107	202	位地址输出, 检测第一个错误
				129	242	报文字符输出, 时间受监控
				159	244	报文字符输出, 检测第一个错误

A-2-30 其它指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
置进位	STC	040	1	0.048	0.060	---
清除进位	CLC	041	1	0.048	0.060	---
选择 EM 存储区块	EMBC	281	2	7.6	14.6	---
延长最大循环时间	WDT	094	2	7.6	17.1	---
保存条件标志	CCS	282	1	5.8	8.3	---
载入条件标志	CCL	283	1	6.4	9.9	---
将 CV 值转换成地址值	FRMCV	284	3	9.4	15.7	---
将地址值转换成 CV 值	TOCV	285	3 ~ 4	10.3	18.2	---
禁止外设服务	IOSP	287	1	---	---	---
允许外设服务	IORS	288	1	---	---	---

A-2-31 块编程指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
块程序开始	BPRG	096	2	7.8	14.1	---
块程序结束	BEND	801	1	8.8	13.4	---
块程序暂停	BPPS	811	2	5.4	8.4	---
块程序重新启动	BPRS	812	2	3.6	4.8	---
条件块退出	(执行条件) EXIT	806	1	8.6	13.2	块退出 (输入条件 ON)
				2.0	2.6	块不退出 (输入条件 OFF)
条件块退出	EXIT(位地址)	806	2	9.8	14.8	块退出 (位 ON)
				3.6	4.2	块不退出 (位 OFF)
条件块退出 (非)	EXIT NOT (位地址)	806	2	3.6	4.3	块退出 (位 OFF)
				8.9	14.9	块不退出 (位 ON)
分支	IF(执行条件)	802	1	1.9	2.4	IF 真 (输入条件 ON)
				3.8	6.4	IF 假 (输入条件 OFF)
	IF(位地址)	802	2	3.2	4.0	IF 真 (位 ON)
				5.1	8.0	IF 假 (位 OFF)
分支 (非)	IF NOT (位地址)	802	2	5.1	8.2	IF 真 (位 OFF)
				3.2	4.1	IF 假 (位 ON)
分支	ELSE	803	1	3.5	5.7	IF 真
				5.3	7.3	IF 假
分支	IEND	804	1	5.3	8.5	IF 真
				2.0	2.4	IF 假
一个循环并等待	WAIT (执行条件)	805	1	10.0	15.9	不等待 (输入条件 ON)
				1.4	1.9	等待 (输入条件 OFF)
	WAIT(位地址)	805	2	9.2	13.5	不等待 (位 ON)
				2.6	3.7	等待 (位 OFF)
一个循环与等待 (非)	WAIT NOT (位地址)	805	2	9.2	13.5	不等待 (位 OFF)
				2.8	3.7	等待 (位 ON)
100ms 定时器等待	TIMW	813	3	15.6	22.9	默认设定
				16.0	23.2	正常执行
	TIMWX	816	3	15.1	21.7	默认设定
				16.0	23.2	正常执行
10ms 定时器等待	TMHW	815	3	15.7	22.6	默认设定
				17.5	24.9	正常执行
	TMHWX	817	3	15.2	22.1	默认设定
				16.4	23.4	正常执行
计数器等待	CNTW	814	4	13.7	20.5	默认设定
				13.4	19.8	正常执行
	CNTWX	818	4	13.1	19.5	默认设定
				13.5	19.7	正常执行
循环控制	LOOP	809	1	4.6	9.1	---

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
循环控制	LEND (执行条件)	810	1	4.2	8.6	不循环 (输入条件 ON)
				3.9	6.5	循环 (输入条件 OFF)
	LEND(位地址)	810	2	6.7	10.4	不循环 (位 ON)
				6.6	8.2	循环 (位 OFF)
循环控制 (非)	LEND NOT (位地址)	810	2	6.7	10.9	不循环 (位 OFF)
				6.6	8.2	循环 (位 ON)

A-2-32 文本字符串处理指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
移动串	MOV\$	664	3	31.5	58.3	传送 1 个字符*1
字符串链接	+\$	656	4	56	104	1 个字符 + 1 个字符*1
取左串	LEFT\$	652	4	33.5	62.2	从 2 个字符中检索 1 个字符*1
取右串	RGHT\$	653	4	33.4	62.1	从 2 个字符中检索 1 个字符*1
取中间串	MID\$	654	5	32.3	60.8	从 3 个字符中检索 1 个字符*1
查找串	FIND\$	660	4	30.3	56.3	从 2 个字符中搜索 1 个字符*1
计算串长度	LEN\$	650	3	14.0	24.9	检测 1 个字符*1
取代串	RPLC\$	661	6	110	213	用 1 个字符取代 2 个字符中的第 1 个字符*1
删除串	DEL\$	658	5	45.6	86.8	删除 2 个字符中的第 1 个字符*1
交换串	XCHG\$	665	3	40.3	75.4	将 1 个字符与另 1 个字符交换*1
清除串	CLR\$	666	2	15.9	28.3	清除 1 个字符*1
插入串	INS\$	657	5	85	162	在 2 个字符中的第 1 个字符后面插入 1 个字符*1
串比较指令	= \$	670	4	27.0	50.9	将 1 个字符与另 1 个字符进行比较
	<> \$	671				
	< \$	672				
	<= \$	673				
	> \$	674				
	= \$	675				

*1 指令执行时间受数据量的影响极大。这将影响循环时间。为减小对循环时间的影响，可指定后台执行。详情请参考“10-2-5 后台执行”。

A-2-33 任务控制指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
任务 ON	TKON	820	2	12.5	19.0	指定了循环任务
				13.6	22.7	指定了扩充任务
任务 OFF	TKOF	821	2	240	393	指定了循环任务
				15.5	25.8	指定了扩充任务

A-2-34 型号转换指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
块传送	XFERC	565	4	6.7	8.2	传送 1 个字
				362	409	传送 1,000 个字
单字分配	DISTC	566	4	4.6	5.3	数据分配
				6.0	7.3	栈操作
数据收集	COLLC	567	4	5.3	6.5	数据分配
				4.5	13.0	栈操作
				5.8	6.1	堆栈操作 以 FIFO 方式读入 1 个字
				42	142	堆栈操作 以 FIFO 方式读入 1,000 个字
位传送	MOVBC	568	4	4.9	5.7	---
位计数器	BCNTC	621	4	5.5	6.4	对 1 个字计数
				873	974	对 1,000 个字计数

A-2-35 特殊功能块指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
获取变量 ID	GETID	286	4	7.6	12.5	---

A-2-36 SFC 指令

指令	助记符	功能编号	长度 (步数)	ON 执行时间 (μs)		条件
				CJ2H CPU6□(-EIP)	CJ2M- CPU□□	
步激活	SA	784	2	9.1	11.8	---
步停止	SE	785	2	9.1	11.9	---
读设定定时器	TSR	780	3	3.9	4.2	---
设定步定时器	TSW	781	3	5.7	8.2	---
SFC ON	SFCON	789	2	14.0	20.4	---
SFC OFF	SFCOFF	790	2	249	402	---
SFC 带复位暂停	SFCPR	793	2	249	405	---
SFC 不带复位暂停	SFCPRN	791	2	249	405	---

A-2-37 功能块实例执行时间

在已创建功能块定义且已将实例复制到用户程序中时，使用下述公式来计算实例执行对循环时间的影响。

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)

下表中列出了 A、B 和 C 的时间长度。

操作			CPU 单元型号	
			CJ2H- CPU6□(-EIP)	CJ2M- CPU□□
A	启动时间	不包括 I/O 参数传送的启动时间	3.3 μs	7.4 μs
B	I/O 参数传送处理时间 数据类型如括号中所示。	1 位 (BOOL) 输入符号或输出符号	0.24 μs	0.88 μs
		1 个字 (INT、UINT、WORD) 输入符号或 输出符号	0.19 μs	0.88 μs
		2 个字 (DINT、UDINT、DWORD、REAL) 输入符号或输出符号	0.19 μs	1.2 μs
		4 个字 (LINT、ULINT、LWORD、LREAL) 输入符号或输出符号	0.38 μs	2.96 μs
		I/O 符号	0.114 μs	0.4 μs
C	功能块定义中的指令执行时间	指令的总处理时间 (与标准用户程序相同)		

例：CJ2H-CPU67-EIP

输入数据类型为 1 个字 (INT) 的变量：3

输出数据类型为 1 个字 (INT) 的变量：2

指令在功能块定义程序段中的总处理时间：10μs

1 个实例的执行时间 = 3.3μs + (3 + 2) × 0.19μs + 10μs = 14.25μs

注 当已将同一个功能块定义复制到多个位置时，根据实例的个数而定，执行时间可能会延长。



附加信息

功能块程序的步数

在已创建功能块定义且已将实例复制到用户程序中时，使用下述公式来计算程序的步数。

$$\text{步数} = \text{实例数} \times (\text{调用部分的大小 } m + \text{I/O 参数传送部分的大小 } n \times \text{参数个数}) + \text{功能块定义中的指令步数 } p (\text{见注})$$

注 当将同一个功能块定义复制到多个位置 (即用于多个实例) 时, 功能块定义中的指令步数 (p) 在后续实例中将会减少。因此, 在上述公式中, 不将实例数乘以功能块定义中的指令步数 (p)。

内容		CJ2 CPU 单元	
m	调用部分	57 步	
n	I/O 参数传送部分 数据类型如括号中所示。	1 位 (BOOL) 输入符号或输出符号	6 步
		1 个字 (INT、UINT、WORD) 输入符号或输出符号	6 步
		2 个字 (DINT、UDINT、DWORD、REAL) 输入符号或输出符号	6 步
		4 个字 (LINT、ULINT、LWORD、LREAL) 输入符号或输出符号	18 步
		I/O 符号	6 步
p	在功能块定义中的指令步数	指令的总步数 (与标准用户程序相同) + 27 步	

例:

输入数据类型为 1 个字 (INT) 的变量: 5

输出数据类型为 1 个字 (INT) 的变量: 5

功能块定义程序段: 100 步

1 个实例中的步数 = 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 (由系统设定)

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A0	---	10ms 增量自由运行定时器	该字中包含电源接通后使用的系统定时器。 当电源接通时将 A0 设为 0000 Hex, 随后每 10ms 都会自动递增 1。 当该值达到 FFFF Hex(655,350ms) 后将恢复为 0000 Hex, 然后继续每 10ms 自动递增 1。 当运行模式切换为 RUN 模式时, 定时器将继续递增。 例: 可不用定时器指令来计算处理 A 和处理 B 之间的时间间隔。这是由计算处理 A 的 A000 值和处理 B 的 A000 值的差值来实现的。该间隔以 10ms 为单位计算。	保持	清除	电源接通后每 10ms
A1	---	100ms 增量自由运行定时器	该字中包含电源接通后使用的系统定时器。 当电源接通时将 A1 设为 0000 Hex, 随后每 100ms 都会自动递增 1。 当该值达到 FFFF Hex(6,553,500ms) 后将恢复为 0000 Hex, 然后继续每 100ms 自动递增 1。 当运行模式切换为 RUN 模式时, 定时器将继续递增。	保持	清除	电源接通后每 100ms
A2	---	1s 增量自由运行定时器	该字中包含电源接通后使用的系统定时器。 当电源接通时将 A2 设为 0000 Hex, 随后每 1s 都会自动递增 1。当该值达到 FFFF Hex(65,535s) 后将恢复为 0000 Hex, 然后继续每 1s 自动递增 1。 当运行模式切换为 RUN 模式时, 定时器将继续递增。	保持	清除	电源接通后每 1s
A50	A50.00 ~ A50.07	基本 I/O 单元信息, 机架 0 插槽 0	若某一位置 ON, 则表示已产生了负载短路保护功能报警输出。 其中, CJ1W-OD202(每位对应 2 点) 占用低 4 位, CJ1W-OD212、OD204 和 MD232 占用最低位, CJ1W-OD232 占用低 2 位。 ON: 短路 OFF: 正常	---	---	每循环刷新
	A50.08 ~ A50.15	基本 I/O 单元信息, 机架 0 插槽 1		---	---	
A51 ~ A69	A51.00 ~ A69.15	基本 I/O 单元信息, 机架 0 插槽 2 ~ 机架 3 插槽 9		---	---	
A90 ~ A93	---	用户程序时间	这些字中包含用户程序的最近覆写日期和时间 (BCD 格式)。 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)	保持	保持	---
A94 ~ A97	---	参数时间	这些字中包含参数的最近覆写日期和时间 (BCD 格式), 格式与上述指令相同。	保持	保持	---

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A99	00	UM 读保护状态	表示是否对 PLC 中的整个用户程序设定读保护。 OFF: 不对 UM 进行读保护 ON: 对 UM 进行读保护	保持	保持	当设置或清除保护时
	01	任务读保护状态	表示是否为单独的任务设定读保护。 OFF: 不对任务进行读保护 ON: 对任务进行读保护			
	02	程序写保护状态 (在设定了读保护的情况下)	表示是否对程序进行写保护。 OFF: 允许写操作 ON: 进行写保护			
	03	允许 / 禁止将程序备份至存储卡时	表示允许 / 禁止创建备份程序文件 (.OBJ)。 OFF: 允许 ON: 禁止			
	14	多任务之间的 IR/DR 操作	将该位置 ON, 使所有任务共享变址寄存器 and 数据寄存器; 将该位置 OFF, 使每项任务单独使用变址寄存器和数据寄存器。 OFF: 单独 ON: 共享 (默认)			
A100 ~ A199	---	出错日志区	发生错误时, 出错代码、错误内容和错误的时间与日期将存储在出错日志区内。可存储 20 条最新的错误。 每条错误记录占用 5 个字; 这 5 个字的功能如下: 第一个字: 出错代码 (位 0 ~ 15) 第二个字: 出错内容 (位 0 ~ 15) 出错内容: 含有出错详情或 0000 的辅助区地址 第三个字: 分 (位 8 ~ 15), 秒 (位 0 ~ 7) 秒: 00 ~ 59, BCD 分: 00 ~ 59, BCD 第四个字: 日 (位 8 ~ 15), 时 (位 0 ~ 7) 时: 00 ~ 23, BCD 日: 01 ~ 31, BCD 第五个字: 年 (位 8 ~ 15), 月 (位 0 ~ 7) 年: 00 ~ 99, BCD 月: 01 ~ 12, BCD 由 FAL(006) 和 FALS(007) 产生的错误也将存储在该出错日志区内。 可通过 CX-Programmer 对出错日志区进行复位。 如果出错日志区写满(20条记录)并且此时又出现一个错误, 则 A100 ~ A104 中的最早记录将被清除, 其它 19 条记录向下移动, 并将新记录存储在 A195 ~ A199 中。	保持	保持	发生错误时刷新 A50014 A300 A400
A200	A200.11	首循环标志	PLC 运行开始后 (从 PROGRAM 模式切换到 RUN 或 MONITOR 模式后) 的一个循环内, 此标志为 ON。 第一次循环为 ON	---	---	---
	A200.12	步标志	当从 STEP(008) 启动步执行时的一个循环内为 ON。该标志用于步开始时的初始化处理。 执行 STEP(008) 后的第一个循环为 ON	清除	---	---
	A200.14	任务启动标志	当一个任务从 STANDBY 或 DISABLED 状态切换到 READY 状态时, 仅在该任务的第一个循环内此标志置 ON。 ON: 第一个循环为 ON (包括从 STANDBY 和 DISABLED 状态切换) OFF: 其它 该标志和 A200.15 唯一的区别是当任务从 STANDBY 状态切换到 READY 状态时该标志也将置 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。该标志用于检查当前任务是否是第一次执行, 以便必要时进行初始化处理。 ON: 第一次执行 OFF: 首次不可执行或不在执行中	清除	---	---

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A201	A201.10	联机编辑等待标志	当等待联机编辑过程时为 ON。 ON: 等待联机编辑 OFF: 不等待联机编辑 如果在等待时接收到其它联机编辑命令, 则不会记录该指令, 且会导致错误发生。	清除	清除	A527
	A201.11	联机编辑标志	当执行联机编辑过程时为 ON。 ON: 联机编辑执行中 OFF: 联机编辑未执行	清除	清除	A527
A202	A202.00 ~ A202.07	通信端口允许标志	当对相应的端口号执行网络指令 (SEND、RECV、CMND、PMCR、TXDU 或 RXDU) 或后台处理时为 ON。位 00 ~ 07 分别对应通信端口 0 ~ 7。 ON: 网络指令未执行 OFF: 网络指令执行中 (端口忙碌) 当对同一个端口号进行编程时, 若使用两条或两条以上的网络指令, 则应将相应的标志作为执行条件, 以防止这些指令被同时执行。 (执行给定端口的网络指令时, 该端口对应的标志置 OFF。) 执行指令时清除该标志。	清除	---	---
	A202.08	CJ2 指令允许标志	可使用 CJ2 指令时为 ON。该标志默认为 ON, 且仅可与下列指令组合使用: SEND2、CMND2、PMCR2 和 RECV2。	---	根据内部状态刷新 (清除)	---
	A202.15	网络通信端口分配允许标志	当执行网络指令 (SEND、RECV、CMND、PMCR、TXDU 或 RXDU) 且存在可自动分配的通信端口时为 ON。 ON: 通信端口可用 OFF: 通信端口不可用 执行通信指令前, 该标志用于确认是否存在可自动分配的通信端口 (在同时使用 9 条或更多通信指令的情况下)。	清除	---	---
A203 ~ A210	---	通信端口完成代码	这些字中包含网络指令 (SEND、RECV、CMND 或 PMCR) 执行或后台处理完成后对应端口的完成代码。 (后台执行完成后, 对应的字将被清为 0000 Hex。) 字 A203 ~ A210 分别对应通信端口 0 ~ 7。 非零数据: 出错代码 0000: 正常情况 显式报文指令 (EXPLT、EGATR、ESATR、ECHRD 或 ECHWR) 执行完成后, 将存储下列代码。 若显式通信出错标志置 OFF, 则存储 0000 Hex。 若显式通信出错标志和网络通信出错标志位均为 ON, 则存储 FINS 结束代码。 若显式通信出错标志为 ON, 且网络通信出错标志位均为 OFF, 则存储显式报文结束代码。 在通信过程中, 会在指令执行完成后存储 0000 Hex 及相应的代码。 当开始运行时, 该代码会被清除。 (当执行给定端口号对应的网络指令时, 该端口的完成代码将被清为 0000。) 执行指令时清除该标志。	清除	---	---
A211	---	CJ2 网络通信指令可用的端口数	当可用的 CJ2 网络通信指令数为 0 时, A202.08 置 OFF。该字可与下列指令组合使用: SEND2、CMND2、PMCR2 和 RECV2。 可通过字中的内容检查通信量。	---	根据内部状态刷新 (清除)	---
A213	A213.00 ~ A213.07	显式通信出错标志	当显式报文指令 (EXPLT、EGATR、ESATR、ECHRD 或 ECHWR) 执行出错时置 ON。 位 00 ~ 07 分别对应通信端口 0 ~ 7。 ON: 出错结束 OFF: 正常结束 在无法发送显式报文或返回显式报文出错响应的情况下, 相应位将会置 ON。 位状态将被保持直至执行下一次显式报文通信。当执行下一条显式报文指令时, 该位将始终保持 OFF 状态。	清除	---	A219.00 ~ A219.07 A203 ~ A210

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A214	A214.00 ~ A214.07	完成网络通信后的首个循环标志	在完成通信后的首个循环内, 各标志都会变为 ON。位 00 ~ 07 对应通信端口 0 ~ 7。已用通信端口的端口号被存储在 A218 中, 用于确定需要访问的标志。 1: 仅限完成通信后的首个循环 2: 其它状态 这些标志在执行通信指令后的下一个循环有效。访问这些标志时应至少延迟一个循环。 通过 A218(所使用的通信端口号)中指定的端口号访问所用端口对应的位。	保持	清除	---
A215	A215.00 ~ A215.07	网络通信出错后的首个循环标志	在通信出错后的首个循环内, 各标志将会变为 ON。位 00 ~ 07 对应通信端口 0 ~ 7。已用通信端口的端口号被存储在 A218 中, 用于确定需要访问的标志。根据 A203 ~ A210 中存储的通信端口完成代码确定错误成因。 ON: 仅限通信出错后的首个循环 OFF: 其它状态 这些标志在执行通信指令后的下一个循环有效。访问这些标志时应至少延迟一个循环。 通过 A218(所使用的通信端口号)中指定的端口号访问所用端口对应的位。	保持	清除	---
A216 和 A217	---	网络通信完成码存储地址	通信指令的完成代码将自动存储在这些字给出的 I/O 存储器地址中。将该地址置于变址寄存器中, 通过变址寄存器进行间接寻址来读取通信完成代码。	保持	清除	---
A218	---	已用通信端口号	当通过通信端口自动分配功能执行通信指令时, 存储所用的通信端口号。 0000 ~ 0007 Hex: 通信端口 0 ~ 7。	保持	清除	---
A219	A219.00 ~ A219.07	通信端口出错标志	在网络指令 (SEND、RECV、CMND 或 PMCR) 执行期间发生错误时 ON。 位 00 ~ 07 分别对应通信端口 0 ~ 7。 ON: 出错 OFF: 正常情况	保持	---	---
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 ~ 17 Hex 若在 PLC 处于 PROGRAM 模式下时更改 PLC 设置中的基本 I/O 单元输入响应时间, 则会导致 PLC 设置中的设定与基本 I/O 单元中的实际值不相符, 除非进行断电后再次上电操作。在这一情况下, 仅可通过这些字监控实际值。	保持	详见功能栏。	PLC 设置 (基本 I/O 单元输入响应时间设定)
A260	---	I/O 分配状态	表示当前的 I/O 分配状态 (即自动 I/O 分配或用户自定义 I/O 分配) 0000 Hex: 自动 I/O 分配 BBBB Hex: 用户自定义 I/O 分配	保持	保持	---

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定	
字	位						
A261	A261.00	I/O 表 创建出错详情	ON: CPU 总线单元设置错误 当正常生成 I/O 表时变为 OFF ON: CPU 总线单元设置错误 OFF: 正常生成 I/O 表	保持	清除	生成 I/O 表时	
	A261.02		ON: 最大 I/O 点数发生溢出 当正常生成 I/O 表时变为 OFF ON: 最大 I/O 点数发生溢出 OFF: 正常生成 I/O 表				A401.11 (I/O 点数过多)
	A261.03		ON: 同一个单元号使用了一次以上。 当正常生成 I/O 表时变为 OFF ON: 同一个单元号使用了一次以上。 OFF: 正常生成 I/O 表				A401.13(编号重复)
	A261.04		ON: I/O 总线错误 当正常生成 I/O 表时变为 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		ON: 因无法检测到 SYSMAC 总线从站导致 I/O 表错误 当正常生成 I/O 表时变为 OFF ON: SYSMAC BUS 从站缺失 OFF: 正常生成 I/O 表				---
	A261.07		ON: 高性能 I/O 单元中出错 当正常生成 I/O 表时变为 OFF ON: 高性能 I/O 单元中出错 OFF: 正常生成 I/O 表				---
	A261.09		ON: I/O 检测尚未完成。 当正常生成 I/O 表时变为 OFF ON: I/O 检测尚未完成。 OFF: 正常生成 I/O 表				---
A262 和 A263	---	最大循环时间 (以 0.1ms 为增量)	这些字中包含从 PLC 运行启动以来的最大循环时间。循环时间 (8 位 16 进制格式) 的高 4 位记录在 A263 中, 低 4 位记录在 A262 中。 0 ~ FFFFFFFF; 0 ~ 429,496,729.5ms (以 0.1ms 为增量)	清除	清除	---	
A264 和 A265	---	当前循环时间 (以 0.1ms 为增量)	这些字中包含以 0.1ms 为增量的当前循环时间。每循环一次, 当前循环时间 (8 位 16 进制格式) 的高 4 位和低 4 位均会记录在 A264 和 A265 中。 00000000 ~ FFFFFFFF(0.0 ~ 429,496,729.5ms)	清除	清除	---	
A266 和 A267	---	当前循环时间 (以 0.01ms 为增量)	这些字中包含以 0.01ms 为增量的当前循环时间。每循环一次, 当前循环时间 (8 位 16 进制格式) 的高 4 位和低 4 位均会记录在 A266 和 A267 中。 00000000 ~ FFFFFFFF(0.0 ~ 42,949,672.95ms)	清除	清除	---	
A270 和 A271	---	高速计数器 0 的当前值 (PV)	包含高速计数器 0 的当前值 (PV)。 低 4 位: A270, 高 4 位: A271	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行 PRV(881) 指令以读取当前值 (PV) 或状态时刷新。 当执行 PRV2(883) 指令 (用于将高速计数器的当前值 (PV) 转换为脉冲总数) 时刷新。 当执行 INI(880) 指令以改变当前值 (PV) 或环形计数器最大值时刷新。 	

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A272 和 A273	---	高速计数器1的当前值(PV)	包含高速计数器1的当前值(PV)。低4位: A272, 高4位: A273	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行 PRV(881) 指令以读取当前值(PV) 或状态时刷新。 当执行 PRV2(883) 指令 (用于将高速计数器的当前值(PV) 转换为脉冲总数) 时刷新。 当执行 IN(880) 指令以改变当前值(PV) 或环形计数器最大值时刷新。
A274	A274.00	高速计数器0范围比较条件1的范围内标志	这些标志表示当高速计数器0在上下限范围比较模式下运行时, 当前值(PV) 是否在指定的8个范围内。但是在达到 / 未达到指定范围之内, 范围内标志均为置 ON。 OFF: 不在范围内, ON: 在范围内	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行 PRV(881) 指令以读取范围比较结果时刷新。 当执行 IN(880) 指令以改变当前值(PV) 或环形计数器最大值时刷新。 复位时刷新。
	A274.01	高速计数器0范围比较条件2的范围内标志				
	A274.02	高速计数器0范围比较条件3的范围内标志				
	A274.03	高速计数器0范围比较条件4的范围内标志				
	A274.04	高速计数器0范围比较条件5的范围内标志				
	A274.05	高速计数器0范围比较条件6的范围内标志				
	A274.06	高速计数器0范围比较条件7的范围内标志				
	A274.07	高速计数器0范围比较条件8的范围内标志				
	A274.08	高速计数器0比较操作	该标志表示是否正在对高速计数器0执行比较操作。 OFF: 已停止, ON: 比较中	清除	清除	比较操作开始或停止时刷新。
	A274.09	高速计数器0上溢 / 下溢标志	该标志表示高速计数器0的当前值(PV) 是否已发生上溢或下溢 (仅可用于线性模式计数范围)。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 变更当前值(PV) 时清除。 发生上溢或下溢时刷新。
	A274.10	高速计数器0的计数方向	该标志表示高速计数器当前正在递增还是递减。将当前循环的计数器当前值(PV) 与上一循环的当前值(PV) 进行比较以确定方向。 OFF: 递减, ON: 递增	---	---	<ul style="list-style-type: none"> 用于高速计数器的设定, 在计数器运行期间有效。 在监控处理期间每个循环均刷新。 当执行 PRV(881) 指令以读取当前值(PV) 或状态时刷新。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定				
字	位									
A275	A275.00	高速计数器1范围比较条件1的范围内标志	这些标志表示当高速计数器1在上下限范围比较模式下运行时, 当前值(PV)是否在指定的8个范围内。但是在达到/未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置ON。 OFF: 不在范围内, ON: 在范围内	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行PRV(881)指令以读取范围比较结果时刷新。 当执行INI(880)指令以改变当前值(PV)或环形计数器最大值时刷新。 复位时刷新。 				
	A275.01	高速计数器1范围比较条件2的范围内标志								
	A275.02	高速计数器1范围比较条件3的范围内标志								
	A275.03	高速计数器1范围比较条件4的范围内标志								
	A275.04	高速计数器1范围比较条件5的范围内标志								
	A275.05	高速计数器1范围比较条件6的范围内标志								
	A275.06	高速计数器1范围比较条件7的范围内标志								
	A275.07	高速计数器1范围比较条件8的范围内标志								
	A275.08	高速计数器1比较执行中标志					该标志表示是否正在对高速计数器1执行比较操作。 OFF: 已停止, ON: 比较中	清除	清除	比较操作开始或停止时刷新。
	A275.09	高速计数器1上溢/下溢标志					该标志表示高速计数器1的当前值(PV)是否已发生上溢或下溢(仅可用于线性模式计数范围)。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 变更当前值(PV)时清除。 发生上溢或下溢时刷新。
A275.10	高速计数器1的计数方向	该标志表示高速计数器1当前正在递增还是递减。将当前循环的计数器当前值(PV)与上一循环的当前值(PV)进行比较以确定方向。 OFF: 递减, ON: 递增	---	---	<ul style="list-style-type: none"> 用于高速计数器的设定, 在计数器运行期间有效。 在监控处理期间每个循环均刷新。 当执行PRV(881)指令以读取当前值(PV)或状态时刷新。 					
A276和A277	脉冲输出0的当前值(PV)	包含对应脉冲输出端口的脉冲输出数。 8000 0000 ~ 7FFF FFFF Hex(-2,147,483,648 ~ 2,147,483,647) 当脉冲在CW方向输出时, 每个脉冲的当前值(PV)递增1。	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行INI(880)指令以更改当前值(PV)时刷新。 脉冲输出重置位置ON时清除。 开始脉冲输出时清除(未定义原点的情况下)。 当执行PRV(881)指令以读取当前值(PV)或状态时刷新。 					
A278和A279	脉冲输出1的当前值(PV)	当脉冲在CCW方向输出时, 每个脉冲的当前值(PV)递减1。 上溢后的当前值(PV): 7FFF FFFF Hex 下溢后的当前值(PV): 8000 0000 Hex 低4位: A276/A278/A322/A324 高4位: A277/A279/A323/A325								

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A280	A280.00	脉冲输出0的脉冲输出状态标志	正在按照 ORG(889)、ACC(888)、PLS2(887) 或 IFEED(892) 指令从脉冲输出 0 输出脉冲并逐步变更输出频率 (加速或减速) 时该标志为 ON。 OFF: 恒速, ON: 加速 / 减速	清除	清除	· 在监控处理期间每个循环均刷新。
	A280.01	脉冲输出 0 上溢 / 下溢标志	该标志表示脉冲输出 0 的当前值是否已发生上溢或下溢。 OFF: 正常, ON: 出错	清除	清除	· 当执行 INI(880) 指令以更改当前值 (PV) 时清除。 · 发生上溢或下溢时刷新。
	A280.02	脉冲输出0的脉冲数设定标志	已通过 PULS(886) 指令设定脉冲输出 0 的输出脉冲数为 ON。 OFF: 不设定, ON: 设定	清除	清除	· 当执行 PULS(886) 指令时刷新。 · 脉冲输出停止时清除。
	A280.03	脉冲输出0的输出完成标志	当由 PULS(886)、PLS2(887) 或 IFEED(892) 指令设置的输出脉冲数已通过脉冲输出 0 输出时为 ON。 OFF: 输出未完成, ON: 输出已完成	清除	清除	当脉冲输出在独立模式下开始或完成时刷新。
	A280.04	脉冲输出0的脉冲输出标志	当脉冲正从脉冲输出 0 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
	A280.05	脉冲输出0无原点标志	脉冲输出 0 的原点未确定时置 ON, 原点确定时置 OFF。 OFF: 已确定原点, ON: 未确定原点	运行开始时置 ON。	运行开始时置 ON。	· 脉冲输出复位时置 ON。 · 开始原点搜索时置 ON。 · 接收到限位输入及设置了清除时置 ON。 · 发生上溢或下溢时置 ON。 · 完成原点搜索时置 OFF。 · 当执行 INI(880) 指令以更改当前值 (PV) 时置 OFF。
	A280.06	脉冲输出0的停止在原点标志	当脉冲输出 0 的当前值 (PV) 与原点 (0) 一致时置 ON。 OFF: 未停止在原点, ON: 已停止在原点	保持	清除	· 当停止在原点时置 ON。 · 当离开原点时置 OFF。
	A280.07	脉冲输出0的输出已停止出错标志	在脉冲输出0原点搜索功能下输出脉冲的过程中发生错误时置 ON。 脉冲输出 0 输出停止出错代码被写入 A444。 0: 未出错, ON: 停止出错	清除	清除	· 开始原点搜索时清除。 · 在原点搜索期间发生致命的脉冲输出错误时刷新。 · 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 · 当限位输入被禁用且已存储致命脉冲输出错误时清除。
	A280.08	脉冲输出0的中断进给执行中标志	通过 IFEED(892) 指令启动脉冲输出 0 的输出后, 若接收到中断输入, 则该标志置 ON。 OFF: 未处于中断进给中 ON: 中断进给中	清除	清除	· 完成中断进给后, 在监控处理期间清除。 · 通过 IFEED(892) 指令启动脉冲输出后, 在中断输入变为 ON 时置 ON。
A280.09	脉冲输出0的中断进给出错标志	通过 IFEED(892) 指令启动脉冲输出 0 的输出后, 若接收到中断输入时发生上溢或下溢或传送了指定数量的脉冲, 则标志置 ON。 ON: 未出错 OFF: 上溢 / 下溢或已传送指定数量的脉冲	清除	清除	· 开始 IFEED(892) 指令处理时清除。 · 通过 IFEED(892) 指令及预定义的原点启动操作后, 若接收到中断输入或传送指定数量的脉冲时发生上溢或下溢, 则该标志置 ON。	

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A281	A281.00	脉冲输出1的脉冲输出状态标志	正在按照 ORG(889)、ACC(888)、PLS2(887) 或 IFEEED(892) 指令从脉冲输出1 输出脉冲并逐步变更输出频率(加速或减速)时该标志为 ON。 OFF: 恒速, ON: 加速/减速	清除	清除	在监控处理期间每个循环均刷新。
	A281.01	脉冲输出1上溢/下溢标志	该标志表示脉冲输出1的当前值是否已发生上溢或下溢。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 当执行 INI(880) 指令以更改当前值(PV)时清除。 发生上溢或下溢时刷新。
	A281.02	脉冲输出1的脉冲数设定标志	已通过 PULS(886) 指令设定脉冲输出1 的输出脉冲数时为 ON。 OFF: 不设定, ON: 设定	清除	清除	<ul style="list-style-type: none"> 当执行 PULS(886) 指令时刷新。 脉冲输出停止时清除。
	A281.03	脉冲输出1的输出完成标志	当由 PULS(886)、PLS2(887) 或 IFEEED(892) 指令设置的输出脉冲数已通过脉冲输出1 输出时为 ON。 OFF: 输出未完成, ON: 输出已完成	清除	清除	当脉冲输出在独立模式下开始或完成时刷新。
	A281.04	脉冲输出1的脉冲输出标志	当脉冲正从脉冲输出1 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
	A281.05	脉冲输出1无原点标志	脉冲输出1 的原点未确定时置 ON, 原点确定时置 OFF。 OFF: 已确定原点, ON: 未确定原点	运行开始时置 ON。	电源接通时置 ON。	<ul style="list-style-type: none"> 脉冲输出复位时置 ON。 开始原点搜索时置 ON。 接收到限位输入及设置了清除时置 ON。 发生上溢或下溢时置 ON。 完成原点搜索时置 OFF。 当执行 INI(880) 指令以更改当前值(PV)时置 OFF。
	A281.06	脉冲输出1的停止在原点标志	当脉冲输出1 的当前值(PV)与原点(0)一致时置 ON。 OFF: 未停止在原点, ON: 已停止在原点	保持	清除	<ul style="list-style-type: none"> 当停止在原点时置 ON。 当离开原点时置 OFF。
	A281.07	脉冲输出1的输出已停止出错标志	在脉冲输出1 原点搜索功能下输出脉冲的过程中发生错误时置 ON。脉冲输出1 输出停止出错代码将被写入 A445。 0: 未出错, ON: 停止出错	保持	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。
	A281.08	脉冲输出1的中断进给执行中标志	通过 IFEEED(892) 指令启动脉冲输出1 的输出后, 若接收到中断输入, 则该标志置 ON。 OFF: 未处于中断进给中 ON: 中断进给中	清除	清除	<ul style="list-style-type: none"> 完成中断进给后, 在监控处理期间清除。 通过 IFEEED(892) 指令启动脉冲输出后, 在中断输入变为 ON 时置 ON。
A281.09	脉冲输出1的中断进给出错标志	通过 IFEEED(892) 指令启动脉冲输出1 的输出后, 若接收到中断输入时发生上溢或下溢或传送了指定数量的脉冲, 则标志置 ON。 ON: 未出错 OFF: 上溢/下溢或已传送指定数量的脉冲	清除	清除	<ul style="list-style-type: none"> 开始 IFEEED(892) 指令处理时清除。 通过 IFEEED(892) 指令及预定义的原点启动操作后, 若接收到中断输入或传送指定数量的脉冲时发生上溢或下溢, 则该标志置 ON。 	

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A283	A283.00	PWM 输出 0 的输出执行中标志	当脉冲正从 PWM 输出 0 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
	A283.08	PWM 输出 1 的输出执行中标志	当脉冲正从 PWM 输出 1 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
A293	---	版本出错信息	当传送的用户程序中包含当前版本 CPU 单元不支持的功能时, 将设定此处的值。 0000 Hex: 正常 0001 Hex: 出错	清除	清除	运行开始时写入
A294	---	程序停止时的任务编号	该字中包含程序因错误而停止执行时正在执行的任务号。 正常任务: 0000 ~ 007F Hex (任务 0 ~ 127) 中断任务: 8000 ~ 80FF Hex (任务 0 ~ 255) A298 和 A299 中包含程序执行停止时的程序地址。	清除	清除	A298/A299
A295	A295.08	指令处理出错标志	当发生指令处理错误且 PLC 设置已设为在发生指令错误时停止运行时, 该标志和出错标志 (ER) 将置 ON。此标志为 ON 时, CPU 单元将停止运行且 ERR/ALM 指示灯将点亮。 ON: 出错标志 ON OFF: 出错标志 OFF	清除	清除	PLC 设置 (发生指令错误时的操作) 错误发生处的任务号被存储在 A294 中, 程序地址被存储在 A298 和 A299 中。
	A295.09	间接 DM/EM BCD 出错标志	当发生间接 DM/EM BCD 错误且 PLC 设置已设为在发生间接 DM/EM BCD 错误的情况下停止运行时, 该标志和访问出错标志 (AER) 将置 ON。(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 单元将停止运行且 ERR/ALM 指示灯将点亮。 ON: 非 BCD OFF: 正常			
	A295.10	非法访问出错标志	当发生非法访问错误且 PLC 设置已设为在非法访问时停止运行时, 此标志和访问出错标志 (AER) 将置 ON。(当存储区发生非法访问时将出现该错误。)此标志为 ON 时, CPU 单元将停止运行且 ERR/ALM 指示灯将点亮。 将下列操作视作非法访问: 读 / 写系统区 读 / 写 EM 文件存储器 写入写保护的区域 间接 DM/EM BCD 错误 (在 BCD 模式下) ON: 发生非法访问 OFF: 正常情况			
	A295.11	无 END 出错标志	当任务中的各程序没有 END(001) 指令时为 ON。 此标志为 ON 时, CPU 单元将停止运行且 ERR/ALM 指示灯将点亮。 ON: 无 END OFF: 正常情况			
	A295.12	任务出错标志	一个任务发生错误时 ON。下列条件下会产生任务错误。 不存在可执行 (启动) 的常规任务。 未将程序分配至任务。 ON: 出错 OFF: 正常			
	A295.13	微分上溢出出错标志	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 单元将停止运行且 ERR/ALM 指示灯将点亮。 ON: 出错 OFF: 正常			
A295.14	非法指令出错标志	存储了无法执行的程序时 ON。此标志为 ON 时, CPU 单元将停止运行且 ERR/ALM 指示灯将点亮。 ON: 出错 OFF: 正常				
A295.15	UM 上溢出出错标志	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 单元将停止运行且 ERR/ALM 指示灯将点亮。 ON: 出错 OFF: 正常				

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
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位二进制程序地址。	清除	清除	(A294 中包含程序停止执行时的任务号。)
A300	---	出错日志指针	当发生错误时, 出错日志指针递增1以指向下一条出错记录的记录位置, 该位置以距离出错日志区(A100 ~ A199)的起始位置的偏移量来表示。 00 ~ 14 Hex 通过将 A500.14(出错日志复位位)从 OFF 转为 ON, 可将出错日志指针清为 00。 当该指针到达 14 Hex(十进制的 20)时, 下一条出错记录将存储在 A195 ~ A199 中。	保持	保持	发生错误时刷新 A500.14
A301	---	当前 EM 存储区块 ---	该字中包含4位十六进制的当前EM存储区块号。可通过 EMBC(281)指令修改当前存储区块号。 0000 ~ 0018 Hex	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 变为 ON 或接通电源后, 当对相应的 CPU 总线单元执行初始化时, 这些标志将置 ON。 位 00 ~ 15 对应单元 0 ~ 15。 在程序中使用这些标志可防止在对CPU总线单元执行初始化时占用其刷新数据。当对 CPU 总线单元执行初始化时, 将无法使用 IORF(097)和 FIORF(225)指令(仅限 CJ2 和 CJ1H-R CPU 单元)。 OFF: 未在执行初始化 ON: 正在执行初始化 (初始化完成后自动复位为 0。) 初始化完成后, 这些位将自动置 OFF。	保持	清除	在初始化期间写入 A501.00
A315	A315.12	标签存储器出错标志(非致命错误) (仅限 CJ2H-CPU6□-EIP)	当存储了网络符号的标签存储器发生错误时为 ON。	保持	清除	---
	A315.13	选件板出错标志	若在通电时拆下选件板或安装了不支持的选件板, 该标志置 ON。 OFF: 正常, ON: 出错	清除	清除	· 电源接通时写入。 · 在监控处理期间每个循环均刷新。
	A315.15	备份存储器出错标志	当写入备份数据区和源存储区或内部闪存中的注释存储区失败时置 ON。 该位在写入成功完成时置 OFF。	保持	清除	---
A316 和 A317	---	高速计数器2的当前值(PV)	包含高速计数器2的当前值(PV)。 低4位: A316, 高4位: A317	清除	清除	---
A318 和 A319	---	高速计数器3的当前值(PV)	包含高速计数器3的当前值(PV)。 低4位: A318, 高4位: A319	清除	清除	---

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A320	A320.00	高速计数器2范围比较条件1的范围内标志	这些标志表示当高速计数器2在上下限范围比较模式下运行时, 当前值(PV)是否在指定的8个范围内。但是在达到/未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置ON。 OFF: 不在范围内, ON: 在范围内	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行PRV(881)指令以读取范围比较结果时刷新。 当执行INI(880)指令以改变当前值(PV)或环形计数器最大值时刷新。 复位时刷新。
	A320.01	高速计数器2范围比较条件2的范围内标志				
	A320.02	高速计数器2范围比较条件3的范围内标志				
	A320.03	高速计数器2范围比较条件4的范围内标志				
	A320.04	高速计数器2范围比较条件5的范围内标志				
	A320.05	高速计数器2范围比较条件6的范围内标志				
	A320.06	高速计数器2范围比较条件7的范围内标志				
	A320.07	高速计数器2范围比较条件8的范围内标志				
	A320.08	高速计数器2比较执行中标志	该标志表示是否正在对高速计数器2执行比较操作。 OFF: 已停止, ON: 比较中	清除	清除	比较操作开始或停止时刷新。
	A320.09	高速计数器2上溢/下溢标志	该标志表示高速计数器2的当前值(PV)是否已发生上溢或下溢(仅可用于线性模式计数范围)。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 变更当前值(PV)时清除。 发生上溢或下溢时刷新。
A320.10	高速计数器2的计数方向	该标志表示高速计数器2当前正在递增还是递减。将当前循环的计数器当前值(PV)与上一循环的当前值(PV)进行比较以确定方向。 OFF: 递减, ON: 递增	---	---	<ul style="list-style-type: none"> 用于高速计数器的设定, 在计数器运行期间有效。 在监控处理期间每个循环均刷新。 当执行PRV(881)指令以读取当前值(PV)或状态时刷新。 	

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定				
字	位									
A321	A321.00	高速计数器3范围比较条件1的范围内标志	这些标志表示当高速计数器3在上下限范围比较模式下运行时, 当前值(PV)是否在指定的8个范围内。但是在达到/未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置ON。 OFF: 不在范围内, ON: 在范围内	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行PRV(881)指令以读取范围比较结果时刷新。 当执行INI(880)指令以改变当前值(PV)或环形计数器最大值时刷新。 复位时刷新。 				
	A321.01	高速计数器3范围比较条件2的范围内标志								
	A321.02	高速计数器3范围比较条件3的范围内标志								
	A321.03	高速计数器3范围比较条件4的范围内标志								
	A321.04	高速计数器3范围比较条件5的范围内标志								
	A321.05	高速计数器3范围比较条件6的范围内标志								
	A321.06	高速计数器3范围比较条件7的范围内标志								
	A321.07	高速计数器3范围比较条件8的范围内标志								
	A321.08	高速计数器3比较执行中标志					该标志表示是否正在对高速计数器3执行比较操作。 OFF: 已停止, ON: 比较中	清除	清除	比较操作开始或停止时刷新。
	A321.09	高速计数器3上溢/下溢标志					该标志表示高速计数器3的当前值(PV)是否已发生上溢或下溢(仅可用于线性模式计数范围)。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 变更当前值(PV)时清除。 发生上溢或下溢时刷新。
A321.10	高速计数器3的计数方向	该标志表示高速计数器3当前正在递增还是递减。将当前循环的计数器当前值(PV)与上一循环的当前值(PV)进行比较以确定方向。 OFF: 递减, ON: 递增	---	---	<ul style="list-style-type: none"> 用于高速计数器的设定, 在计数器运行期间有效。 在监控处理期间每个循环均刷新。 当执行PRV(881)指令以读取当前值(PV)或状态时刷新。 					
A322和A323	---	脉冲输出2的当前值(PV)	包含对应脉冲输出端口的脉冲输出数。 包含对应脉冲输出端口的脉冲输出数。 8000 0000 ~ 7FFF FFFF Hex(-2,147,483,648 ~ 2,147,483,647)	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 当执行INI(880)指令以更改相应计数器的当前值(PV)时刷新。 脉冲输出重置位置ON时清除。 				
A324和A325	---	脉冲输出3的当前值(PV)	当脉冲在CW方向输出时, 每个脉冲的当前值(PV)递增1。 当脉冲在CCW方向输出时, 每个脉冲的当前值(PV)递减1。 上溢后的当前值(PV): 7FFF FFFF Hex 下溢后的当前值(PV): 8000 0000 Hex 低4位: A276/A278/A322/A324 高4位: A277/A279/A323/A325	清除	清除	<ul style="list-style-type: none"> 开始脉冲输出时清除(未定义原点的情况下)。 当执行PRV(881)指令以读取当前值(PV)或状态时刷新。 				

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A326	A326.00	脉冲输出2的脉冲输出状态标志	正在按照 ORG(889)、ACC(888)、PLS2(887) 或 IFEED(892) 指令从脉冲输出 2 输出脉冲并逐步变更输出频率 (加速或减速) 时该标志为 ON。 OFF: 恒速, ON: 加速 / 减速	清除	清除	在监控处理期间每个循环均刷新。
	A326.01	脉冲输出 2 上溢 / 下溢标志	该标志表示脉冲输出 2 的当前值是否已发生上溢或下溢。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 当执行 INI(880) 指令以更改当前值 (PV) 时清除。 发生上溢或下溢时刷新。
	A326.02	脉冲输出2的脉冲数设定标志	已通过 PULS(886) 指令设定脉冲输出 2 的输出脉冲数为 ON。 OFF: 不设定, ON: 设定	清除	清除	<ul style="list-style-type: none"> 当执行 PULS(886) 指令时刷新。 脉冲输出停止时清除。
	A326.03	脉冲输出2的输出完成标志	当由 PULS(886)、PLS2(887) 或 IFEED(892) 指令设置的输出脉冲数已通过脉冲输出 2 输出时为 ON。 OFF: 输出未完成, ON: 输出已完成	清除	清除	当脉冲输出在独立模式下开始或完成时刷新。
	A326.04	脉冲输出2的脉冲输出标志	当脉冲正从脉冲输出 2 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
	A326.05	脉冲输出2无原点标志	脉冲输出 2 的原点未确定时置 ON, 原点确定时置 OFF。 OFF: 已确定原点, ON: 未确定原点	运行开始时置 ON。	电源接通时置 ON。	在监控处理期间每个循环均刷新。
	A326.06	脉冲输出2的停止在原点标志	当脉冲输出 2 的当前值 (PV) 与原点 (0) 一致时置 ON。 OFF: 未停止在原点, ON: 已停止在原点	清除	清除	在监控处理期间每个循环均刷新。
	A326.07	脉冲输出2的输出已停止出错标志	在脉冲输出2原点搜索功能下输出脉冲的过程中发生错误时置 ON。脉冲输出 2 输出停止出错代码将被写入 A438。 0: 未出错, ON: 停止出错	清除	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。
	A326.08	脉冲输出2的中断进给执行中标志	通过 IFEED(892) 指令启动脉冲输出 2 ~ 3 的输出后, 若接收到中断输入, 则这些标志置 ON。 OFF: 未处于中断进给中 ON: 中断进给中	清除	清除	<ul style="list-style-type: none"> 完成中断进给后, 在监控处理期间清除。 通过 IFEED(892) 指令启动脉冲输出后, 在接收到中断输入时刷新。
A326.09	脉冲输出2的中断进给出错标志	通过 IFEED(892) 指令启动脉冲输出 2 的输出后, 若接收到中断输入时发生上溢或下溢或传送了指定数量的脉冲, 则标志置 ON。 OFF: 上溢 / 下溢或已传送指定数量的脉冲 ON: 正常	清除	清除	<ul style="list-style-type: none"> 开始 IFEED(892) 指令处理时清除。 通过 IFEED(892) 指令及预定义的原点启动操作后, 若接收到中断输入或传送指定数量的脉冲时发生上溢或下溢, 则该标志置 ON。 	
A327	A327.00	脉冲输出3的脉冲输出状态标志	正在按照 ORG(889)、ACC(888)、PLS2(887) 或 IFEED(892) 指令从脉冲输出 3 输出脉冲并逐步变更输出频率 (加速或减速) 时该标志为 ON。 OFF: 恒速, ON: 加速 / 减速	清除	清除	在监控处理期间每个循环均刷新。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A327	A327.01	脉冲输出 3 上溢 / 下溢标志	该标志表示脉冲输出 3 的当前值是否已发生上溢或下溢。 OFF: 正常, ON: 出错	清除	清除	<ul style="list-style-type: none"> 当执行 INI(880) 指令以更改当前值 (PV) 时清除。 发生上溢或下溢时刷新。
	A327.02	脉冲输出 3 的脉冲数设定标志	已通过 PULS(886) 指令设定脉冲输出 3 的输出脉冲数为 ON。 OFF: 不设定, ON: 设定	清除	清除	<ul style="list-style-type: none"> 当执行 PULS(886) 指令时刷新。 脉冲输出停止时清除。
	A327.03	脉冲输出 3 的输出完成标志	当由 PULS(886)、PLS2(887) 或 IFEED(892) 指令设置的输出脉冲数已通过脉冲输出 3 输出时为 ON。 OFF: 输出未完成, ON: 输出已完成	清除	清除	当脉冲输出在独立模式下开始或完成时刷新。
	A327.04	脉冲输出 3 的脉冲输出标志	当脉冲正从脉冲输出 3 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
	A327.05	脉冲输出 3 无原点标志	脉冲输出 3 的原点未确定时置 ON, 原点确定时置 OFF。 OFF: 已确定原点, ON: 未确定原点	运行开始时置 ON。	电源接通时置 ON。	<ul style="list-style-type: none"> 脉冲输出复位时置 ON。 开始原点搜索时置 ON。 接收到限位输入及设置了清除时置 ON。 发生上溢或下溢时置 ON。 完成原点搜索时置 OFF。 当执行 INI(880) 指令以更改当前值 (PV) 时置 OFF。
	A327.06	脉冲输出 3 的停止在原点标志	当脉冲输出 3 的当前值 (PV) 与原点 (0) 一致时置 ON。 OFF: 未停止在原点, ON: 已停止在原点	清除	清除	<ul style="list-style-type: none"> 当停止在原点时置 ON。 当离开原点时置 OFF。
	A327.07	脉冲输出 3 的输出已停止出错标志	在脉冲输出 3 原点搜索功能下输出脉冲的过程中发生错误时置 ON。脉冲输出 3 输出停止出错代码将被写入 A439。 0: 未出错, ON: 停止出错	清除	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。
	A327.08	脉冲输出 3 的中断进给执行中标志	通过 IFEED(892) 指令启动脉冲输出 3 的输出后, 若接收到中断输入, 则这些标志置 ON。 OFF: 未处于中断进给中 ON: 中断进给中	清除	清除	<ul style="list-style-type: none"> 完成中断进给后, 在监控处理期间清除。 通过 IFEED(892) 指令启动脉冲输出后, 在接收到中断输入时刷新。
	A327.09	脉冲输出 3 的中断进给出错标志	通过 IFEED(892) 指令启动脉冲输出 3 的输出后, 若接收到中断输入时发生上溢或下溢或传送了指定数量的脉冲, 则标志置 ON。 ON: 未出错 OFF: 上溢 / 下溢或已传送指定数量的脉冲	清除	清除	<ul style="list-style-type: none"> 开始 IFEED(892) 指令处理时清除。 通过 IFEED(892) 指令及预定义的原点启动操作后, 若接收到中断输入或传送指定数量的脉冲时发生上溢或下溢, 则该标志置 ON。
A329	A329.00	PWM 输出 2 的输出执行中标志	当脉冲正从 PWM 输出 2 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。
	A329.08	PWM 输出 3 的输出执行中标志	当脉冲正从 PWM 输出 3 输出时为 ON。 OFF: 已停止, ON: 输出中	清除	清除	在脉冲输出开始或停止时刷新。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A330 ~ A335	A330.00 ~ A335.15	高性能 I/O 单元初始化标志	高性能 I/O 单元重启位 (A502.00 ~ A507.15) 从 OFF 变为 ON 或接通电源后, 当对相应的高性能 I/O 单元执行初始化时, 这些标志将置 ON。 这些字中的位对应单元 0 ~ 95: 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可防止在对高性能 I/O 单元执行初始化时占用其刷新数据。同样, 对高性能 I/O 单元执行初始化时, 将无法使用 IORF(097) 和 FIORF(225) 指令。 OFF: 未在执行初始化 ON: 正在执行初始化 (初始化完成后自动复位为 0。) 初始化完成后, 这些位将自动置 OFF。	保持	清除	A502.00 ~ A507.15
A336	A336.00 ~ A336.15	启动时检测到的单元 (机架 0 ~ 3)	在各机架上检测到的单元数以 1 位十六进制形式存储 (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 示例: 若机架 0、机架 1、机架 2 和机架 3 分别含有 1 个、4 个、8 个及 10 个单元, 则将存储下列内容: A336 = A 8 4 1	保持	清除	---
A339 和 A340	---	最大微分标志数	这些字中包含微分指令正在使用的微分标志的最大数目。	见“功能”栏	清除	运行开始时写入 A295.13
A343	A343.00 ~ A343.02	存储卡类型	表示安装的存储卡的类型。 0: 无 4: 闪存 当接通 PLC 电源或打开存储卡电源开关时, 将记录该信息。	保持	见“功能”栏	见“功能”栏
	A343.06	EM 文件存储器格式化出错标志	当文件存储器占用的首个 EM 存储区块发生格式化错误时为 ON。 ON: 格式化错误 OFF: 格式化正常 该标志在格式化成功完成后置 OFF。	保持	清除	---
	A343.07	存储卡格式化出错标志	当存储卡未执行格式化操作或发生格式化错误时为 ON。(该标志在格式化成功完成后置 OFF。) ON: 格式化错误 OFF: 格式化正常 当接通 PLC 电源或打开存储卡电源开关时, 将写入该标志。	保持	见“功能”栏	见“功能”栏
	A343.08	文件传输出错标志	将数据写入文件存储器期间发生错误时为 ON。 ON: 出错 OFF: 正常	保持	清除	写入文件数据时刷新。
	A343.09	写文件出错标志	由于写保护或数据大小超出文件存储器容量导致数据无法写入文件存储器时为 ON。 ON: 无法写入 OFF: 正常情况	保持	清除	写入文件数据时刷新。
	A343.10	读文件错误	由于故障 (文件或数据损坏) 导致无法读取文件时为 ON。 ON: 无法读取 OFF: 正常情况或正在执行读处理	保持	清除	读取文件数据时刷新。
	A343.11	文件丢失标志	当试图读取的文件不存在或指定写入文件的目录不存在时为 ON。 ON: 指定文件或目录丢失 OFF: 正常情况或正在执行读处理	保持	清除	读取文件数据时刷新。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A343	A343.13	文件存储器操作标志	在执行下列任意操作时为 ON, 不执行时为 OFF。 CMND 指令向本地 CPU 单元发送 FINS 命令。 执行文件存储器指令。 通过辅助区中的控制位进行程序替换。 进行简易备份操作。 ON: 指令执行中 OFF: 指令未执行	保持	清除	执行文件存储器指令时刷新。
	A343.14	访问文件数据标志	当访问文件数据时为 ON。 ON: 文件正被访问 OFF: 文件未被访问 通过该标志可防止两条文件存储器指令被同时执行。	保持	清除	---
	A343.15	存储卡检测标志	检测到存储卡时为 ON, 未检测到存储卡时为 OFF。 ON: 检测到存储卡 OFF: 未检测到存储卡	保持	清除	插入存储卡或电源接通时刷新。
A344	A344.00 ~ A344.07	EM 文件存储器的起始存储区块	包含 EM 跟踪存储器或 EM 文件存储器的起始存储区块号。将 EM 区中的所有 EM 存储区块 (从起始存储区块到末尾存储区块) 格式化, 以作为文件存储器或跟踪存储器使用。如果未将 EM 区存储区块格式化为文件存储器或跟踪存储器, 则 A344 中的内容为 FFFF Hex。 若要将 EM 区的一部分转换为文件存储器或跟踪存储器, 可在 CX-Programmer 中选择 “PLC” – “Memory Allocation” (存储器分配) – “EM Memory Settings” (EM 存储器设定), 然后选择 “File Memory” (文件存储器) 或 “Trace Memory” (跟踪存储器)。	保持	保持	PLC 设置 (EM 文件设定启用参数及 EM 起始文件号参数)
	A344.14	EM 跟踪存储器标志	当 A344 中的内容不是 FFFF Hex, 且该标志为 ON 时, EM 区中相应的存储区块 (从 A344.00 ~ A344.07 中指定的存储区块到 EM 区末尾) 将被格式化为跟踪存储器。	保持	保持	
	A344.15	EM 文件存储器标志	当 A344 中的内容不是 FFFF Hex, 且该标志为 ON 时, EM 区中相应的存储区块 (从 A344.00 ~ A344.07 中指定的存储区块到 EM 区末尾) 将被格式化为文件存储器。	保持	保持	
A345	A345.00	FB 程序源信息标志	当源代码 / 注释存储器中包含 FB 程序源信息时为 ON。 ON: 包含信息, OFF: 不包含信息	保持	内部状态更新	从 CX-Programmer 传送时
	A345.01	符号表信息标志	当源代码 / 注释存储器中包含符号表信息时为 ON。 ON: 包含信息, OFF: 不包含信息	保持	内部状态更新	
	A345.02	注释信息标志	当源代码 / 注释存储器中包含注释信息时为 ON。 ON: 包含信息, OFF: 不包含信息	保持	内部状态更新	
	A345.03	程序索引信息标志	当源代码 / 注释存储器中包含程序索引信息时为 ON。 ON: 包含信息, OFF: 不包含信息	保持	内部状态更新	
	A345.05	SFC 程序源信息标志	当源代码 / 注释存储器中包含 SFC 程序源信息时为 ON。 ON: 包含信息, OFF: 不包含信息	保持	内部状态更新	
	A345.07	网络符号 (标签) 信息标记	当标签存储器中包含网络符号 (标签) 信息时为 ON。 ON: 包含信息, OFF: 不包含信息	保持	内部状态更新	
A346 和 A347	---	需传送的剩余字数	这些字中包含需由 FREAD(700) 或 FWRTIT(701) 指令传送的剩余字数 (8 位十六进制格式)。执行其中的一条指令时, 需传送的字数将被写入 A346 和 A347 中。 在数据传送过程中, 这些字中的值递减。 A346 和 A347 中分别存储低 4 位和高 4 位。 检查这些字的内容, 以确定是否可以成功传送指定数量的字。	保持	清除	执行 FREAD 或 FWRTIT 指令时写入实际传送数据时递减

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A351 ~ A354	---	日历 / 时钟区	这些字中包含以 BCD 格式表示的 CPU 单元的内部时钟数据。该时钟可通过 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	保持	保持	每个循环均写入
A360 ~ A391	A360.01 ~ A391.15	已执行的 FAL 编号标志	在执行 FAL(006) 时, 和指定的 FAL 号对应的标志将置 ON。位 A360.01 ~ A391.15 对应 FAL 编号 001 ~ 511。 ON: 已执行该 FAL OFF: 未执行该 FAL 错误被清除时该标志将置 OFF。	保持	清除	发生错误时刷新 A402.15
A392	A392.04	串行端口出错标志	当 RS-232C 端口发生错误时为 ON。(请勿在外设总线模式、1:N NT 链接模式或串行 PLC 链接主站 / 从站单元模式下访问该位。) ON: 出错 OFF: 正常	保持	清除	发生错误时刷新
	A392.05	串行端口发送就绪标志 (无协议模式)	当 RS-232C 端口可以在无协议模式下发送数据时为 ON。 ON: 能够发送 OFF: 无法发送	保持	清除	发送后写入
	A392.06	串行端口接收完成标志 (无协议模式)	当 RS-232C 端口在无协议模式下完成接收时为 ON。 · 指定字节数时: 接收到指定字节数时为 ON。 · 指定结束码时: 接收到结束码或接收到 256 个字节时为 ON。	保持	清除	接收后写入
	A392.07	串行端口接收上溢标志 (无协议模式)	当内置 RS-232C 端口在无协议模式下接收期间发生数据上溢时为 ON。 · 在指定了字节数的情况下: 当接收完成, 但没有执行 RXD(235) 指令前接收到更多数据时为 ON。 · 在指定了结束代码的情况下: 当接收到结束代码, 但执行 RXD(235) 指令前接收到更多数据时为 ON。 在结束代码前接收到 257 个字节时为 ON。 ON: 上溢 OFF: 正常	保持	清除	
A393	A393.00 ~ A393.07	串行端口 PT 通信标志	当串行端口在 NT 链接模式或串行 PLC 链接模式下通信时, 相应位将置 ON。 位 0 ~ 7 对应单元 0 ~ 7。 ON: 通信中 OFF: 未通信	保持	清除	对令牌有正常响应时刷新。
	A393.08 ~ A393.15	串行端口 PT 优先级注册标志	当 RS-232C 端口在 NT 链接模式下通信时, 相应优先级的 PT 对应的位将会置 ON。 位 0 ~ 7 对应单元 0 ~ 7。 接收到优先级注册命令时, 将写入这些标志。 ON: 已注册优先级 OFF: 未注册优先级	保持	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 端口处于无协议模式时接收到的数据的字节数 (二进制)。	保持	清除	接收数据时刷新。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A395	A395.06	文件已删除标志	如果在存储卡文件的更新过程中发生断电, 系统将会删除该文件的剩余部分。 ON: 文件已删除 OFF: 无文件删除	清除	清除	当系统删除文件时刷新。
	A395.07		如果在 EM 存储器文件的更新过程中发生断电, 系统将会删除该文件的剩余部分。 ON: 文件已删除 OFF: 无文件删除	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	在后台处理期间发生指令处理错误或非访区访问错误时置 ON。 ON: 出错。电源接通或运行启动时置 OFF(0)。 OFF: 正常。启动后台处理时为 OFF(0)。	清除	清除	---
	A395.11	存储器损坏检测标志	若在通电时检测到存储器损坏, 则该标志置 ON。 ON: 存储器损坏 OFF: 正常	保持	见“功能”栏	电源接通时刷新。
	A395.12	DIP 开关第 6 位的状态标志	每循环一次, CPU 单元前部的 DIP 开关第 6 位的状态标志均会被写入该标志。 ON: 第 6 位 ON OFF: 第 6 位 OFF	保持	见“功能”栏	每个循环均写入
A400	---	出错代码	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, 将 4 位十六进制出错代码写入该字中。(请参考第 A-146 页上的“A-3-3 辅助区操作的详细信息”。) 同时发生两个或两个以上的错误时, 记录最高出错代码。	清除	清除	发生错误时刷新

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A401	A401.03	存储卡传送出错标志	<ul style="list-style-type: none"> · 通电时, 若存储卡中的内容未被正确写入 CPU 单元, 则该标志置 ON。 · 通电时, 若传送的程序文件 (.OBJ) 中包含网络符号, 则该标志置 ON(仅限 CJ2H-CPU6□ 和 CJ2M-CPU1□)。 	执行错误清除操作时清除。	清除	---
	A401.05	版本出错标志	CPU 单元的版本不支持所传送的用户程序时为 ON。	执行错误清除操作时清除。	清除	---
	A401.06	FALS 出错标志 (致命错误)	<p>由 FALS(006) 指令产生一个非致命错误时置 ON。CPU 单元将继续运行且 ERR/ALM 指示灯将闪烁。</p> <p>相应出错代码将写入到 A400 中。出错代码 C101 ~ C2FF 对应 FALS 编号 001 ~ 511。</p> <p>ON: 已执行 FALS(006)</p> <p>OFF: 未执行 FALS(006)</p> <p>当清除 FALS 错误时该标志置 OFF。</p>	清除	清除	发生错误时刷新 A400
	A401.08	循环时间超出标志 (致命错误)	<p>循环时间超出在 PLC 设置中设定的最大循环时间 (循环时间的监控时间) 时置 ON。CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。</p> <p>OFF: 未超出最大循环时间</p> <p>ON: 超出最大循环时间</p> <p>错误被清除后该标志将置 OFF。</p>	清除	清除	超出最大循环时间时刷新。 PLC 设置 (循环时间的监控时间)
	A401.09	程序出错标志 (致命错误)	<p>程序内容不正确时置 ON。</p> <p>CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。错误发生处的任务编号将被存储在 A294 中, 程序地址将被存储在 A298 和 A299 中。</p> <p>所发生的程序错误类别将存储在 A295.08 ~ A295.15 中。更多有关程序错误的详情, 请参考 A295 和《编程手册》中的说明。</p> <p>ON: 出错</p> <p>OFF: 正常</p> <p>错误被清除后该标志将置 OFF。</p>	清除	清除	A294, A295, A298 和 A299
	A401.10	I/O 设定出错标志 (致命错误)	<ul style="list-style-type: none"> · 当注册的 I/O 表与实际 I/O 表不相符 (即注册的单元与实际连接的单元不相符) 时为 ON。 · 对于 CJ2H-CPU6□-EIP, 当中断输入单元安装在下述插槽以外的插槽时为 ON。 CJ2H-CPU6□-EIP: CPU 机架插槽 0 ~ 3 CJ2H-CPU6□ 或 CJ2M-CPU□□: CPU 机架插槽 0 ~ 4 · 将 CJ2H-CPU6□-EIP 或 CJ2M-CPU1□ CPU 单元的注册 I/O 表下载至 CJ2H-CPU6□ 或 CJ2M-CPU1□ CPU 单元或将 CJ2H-CPU6□ CPU 单元的注册 I/O 表下载至 CJ2H-CPU6□-EIP 或 CJ2M-CPU3□ CPU 单元时为 ON。CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 <p>ON: 出错</p> <p>OFF: 正常</p> <p>错误被清除后该标志将置 OFF。</p>	清除	清除	A405.08
	A401.11	I/O 点数过多标志 (致命错误)	<p>当基本 I/O 单元所使用的 I/O 点数超出 PLC 允许的最大点数或同一个机架上连接了超过 11 个单元时为 ON。</p> <p>CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。</p> <p>ON: 出错</p> <p>OFF: 正常</p> <p>错误被清除后该标志将置 OFF。</p>	清除	清除	A407

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A401	A401.13	重复出错标志 (致命错误)	下述情况下为 ON: · 两个 CPU 总线单元占用了同一个单元号。 · 两个高功能 I/O 单元占用了同一个单元号。 · 两个基本 I/O 单元占用了同一个数据区字。 CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 A409 ~ A416 中指定的单元号重复。 ON: 重复错误 OFF: 正常 错误被清除后该标志将置 OFF。	清除	清除	A410 ~ A416
	A401.14	I/O 总线出错标志 (致命错误)	当 CPU 单元和另一个安装在插槽中的单元之间进行数据传送或未将端板连接到 CPU 机架或扩展机架时为 ON。 CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 将发生 I/O 总线错误的插槽号 (00 ~ 09) 以二进制格式写入 A404.00 ~ A404.07。如果未连接端板, 则将会存储 0E Hex。 将发生 I/O 总线错误的机架号 (00 ~ 07) 以二进制格式写入 A404.08 ~ A404.15。若 CJ2HCPU6□-EIP 或 CJ2M-CPU3□ 的内置网络中发生 I/O 总线错误, 则这些位中将包含 0B Hex。如果未连接端板, 则将会存储 0E Hex。	清除	清除	A404
	A401.15	存储器出错标志 (致命错误)	在通电状态下, 若存储器中发生错误或从存储卡进行自动传送时出错, 则该标志置 ON。 CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 ON: 出错 OFF: 正常 A403.00 ~ A403.08 中包含发生错误的位置信息。 错误被清除时该标志将置 OFF。	清除	清除	A403
A402	A402.00	其它非致命出错标志 (系统工作存储器错误)	· 发生 A402.01 ~ A402.15 中未定义的非致命错误时置 ON。(如用于联机编辑的存储器发生错误时) · 其它非致命错误的详情被存储在 A315 中。	执行错误清除操作时清除	清除	---
	A402.02	高功能 I/O 单元设定出错标志 (非致命错误)	安装的高功能 I/O 单元与 I/O 表中注册的高功能 I/O 单元不一致时为 ON。CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 发生设定错误的单元号存储在 A428 ~ A433 中。 当单元同步运转设定中注册的单元发生下述任一情况时为 ON。 · 该单元不支持同步单元运转功能。 · PLC 中未连接该单元。 · CPU 机架上未安装该单元 (即安装在了扩展机架上)。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除且将正确的 PLC 设置传送到 CPU 单元时该标志将置 OFF。 发生设定错误的单元号存储在 A428 ~ A433 中。	清除	清除	A428 ~ A433
	A402.03	CPU 总线单元设定出错标志 (非致命错误)	安装的 CPU 总线单元与 I/O 表中注册的 CPU 总线单元不一致时为 ON。CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 发生设定错误的单元号被写入 A427 中。	清除	清除	A427

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A402	A402.04	电池出错标志 (非致命错误)	如果 CPU 单元的电池未连接或电压过低且在 PLC 设置中设定了检测电池错误时置 ON。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 该标志可用于控制外部报警灯或其它指示灯, 以表示电池需要更换。	清除	清除	PLC 设置 (检测电池错误)
	A402.06	高性能 I/O 单元出错标志 (非致命错误)	当 CPU 单元和高性能 I/O 单元之间的数据交换出现错误 (包括高性能 I/O 单元本身的错误) 时为 ON。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。发生错误的高性能 I/O 单元停止运转。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 发生数据交换错误的单元号存储在 A418 ~ A423 中。	清除	清除	A418 ~ A423
	A402.07	CPU 总线单元出错标志 (非致命错误)	当 CPU 单元和 CPU 总线单元之间的数据交换出现错误 (包括 CPU 总线单元本身的错误) 时为 ON。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。发生错误的 CPU 总线单元停止运转。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 发生数据交换错误的单元号存储在 A417 中。	清除	清除	A417
	A402.10	PLC 设置出错标志 (非致命错误)	当在 PLC 设置中发生设定错误时置 ON。CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。出错位置被写入 A406。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 发生单元同步运转设定错误后, 若将正确的设定传送到同步单元, 则该标志也会置 OFF。	清除	清除	A406
	A402.12	基本 I/O 单元出错标志 (非致命错误)	基本 I/O 单元发生错误时为 ON。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除后该标志将置 OFF。 出错位置被写入 A408。	清除	清除	A408
	A402.13	重复刷新出错标志 (非致命错误)	当 PLC 设置中的 “Detect Duplicate Refresh Errors” (检测重复刷新错误) 参数设为 “检测重复刷新错误” 且同一个高性能 I/O 单元中发生以下任何一种情况时置 ON。 · 循环任务中的 FIORF(225)、IORF(097)、IORD(222) 或 IOWR(223) 与中断任务中的 FIORF(225)、IORF(097)、IORD(222) 或 IOWR(223) 发生竞争冲突。 · I/O 刷新时, 在中断任务中执行 FIORF(225)、IORF(097)、IORD(222) 或 IOWR(223)。 如果没有在高性能 I/O 单元的 PLC 设置中禁用循环刷新, 并在中断任务中对同一个高性能 I/O 单元执行 FIORF(225)、IORF(097)、IORD(222) 或 IOWR(223), 将发生重复刷新错误。	清除	清除	A426、PLC 设置 (当 PLC 设置中 “Detect Duplicate Refresh Errors” (检测重复刷新错误) 参数设为 “Detect” (检测) 时。)
	A402.15	FAL 出错标志 (非致命错误)	FAL(006) 执行过程中产生一个非致命错误时置 ON。CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 与在 FALS(006) 中指定的 FAL 编号对应的 A360 ~ A391 中的位将置 ON 且相应的出错代码将写入 A400 中。出错代码 4101 ~ 42FF 对应 FAL 号 001 ~ 2FF(0 ~ 511)。 ON: 发生 FALS(006) 错误 OFF: 未执行 FALS(006) 错误被清除后该标志将置 OFF。	清除	清除	A360 ~ A391, A400

地址		名称	功能	模式变更后 的状态	启动时的 状态	写入时间 / 相关标 志, 设定
字	位					
A403	A403.00 ~ A403.08	存储器错误位置	当发生存储器错误时, 存储器出错标志 (A401.15) 将置 ON, 同时下列其中一个标志将置 ON 以指示发生错误的存储区: A403.00: 用户程序 A403.04: PLC 设置 A403.05: 注册 I/O 表 A403.07: 路由表 A403.08: CPU 总线单元设定 发生存储器错误时 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除后相应标志将置 OFF。	清除	清除	A401.15
A404	A404.00 ~ A404.07	I/O 总线错误插槽号	包含发生 I/O 总线错误的插槽号 (00 ~ 09, 8 位二进制格式)。若 CJ2H-CPU6□-EIP 的内置 EtherNet/IP 部分发生 I/O 总线错误, 则将会存储 0B Hex。如果未将端板连接到 CPU 机架或扩展机架, 则将会存储 F Hex。 CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 00 ~ 09(插槽 00 ~ 09) I/O 总线出错标志 (A401.14) 将会置 ON。 错误被清除后该标志将置 OFF。	清除	清除	A401.14
	A404.08 ~ A404.15	I/O 总线错误机架号	包含发生 I/O 总线错误的机架号 (00 ~ 03, 8 位二进制格式)。若 CJ2H-CPU6□-EIP 的内置 EtherNet/IP 部分发生 I/O 总线错误, 则将会存储 0B Hex。如果未将端板连接到 CPU 机架或扩展机架, 则将会存储 F Hex。 CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 00 ~ 03(机架 00 ~ 03) I/O 总线出错标志 (A401.14) 将会置 ON。 错误被清除后该标志将置 OFF。	清除	清除	A401.14
A405	A405.08	中断输入单元位置出错标志	当中断输入单元未连接在下列范围中时为 ON。如果该标志置 ON, 则会发生 I/O 设定错误 (致命错误) 并导致运转停止。 · CJ2H-CPU6□-EIP: CPU 机架插槽 0 ~ 3(CPU 单元右侧的 4 个单元) · CJ2H-CPU6□: CPU 机架插槽 0 ~ 4(CPU 单元右侧的 5 个单元) 如果单元被安装在上述范围中, 但未使用虚拟单元注册方式将这一范围中的单元分配至 I/O 表, 则也会发生错误。	清除	清除	A401.10
A407	A407.00 ~ A407.12	I/O 点数过多, 详情 1	当下列任意一种数值出现上溢或发生重复错误时, 将以二进制形式保存相应的值。CPU 单元将会停止运转。CPU 单元前部的 ERR/ALM 指示灯也会点亮。有关出错原因的信息, 请参见 “I/O 点数过多, 详情 2” (A407.13 ~ A407.15) 中的说明。 1. 当 I/O 表中设定的 I/O 点总数 (不含从站机架) 超过 CPU 单元允许的最大数量时, 将在此处写入 I/O 点数。 2. 当中断输入点数超过 32 时, 将写入中断输入点数。 3. 当扩展机架数超过最大值时, 将在此处写入机架数。 当发生错误时, 将在此处 (A407.00 ~ A407.12) 写入相关值。当错误被清除时, 这些位也将被清除。	清除	清除	A401.11, A407.13 ~ A407.15
	A407.13 ~ A407.15	I/O 点数过多, 详情 2	这些位对应的 3 位二进制值表示发生 I/O 点数过多错误的原因以及写入位 A407.00 ~ A407.12 中的值的含义。 000 ~ 101(0 ~ 5) 对应上文 “I/O 点数过多, 原因 1” 中所述的原因 1 ~ 6。 000: I/O 点总数过多 001: 中断输入点数过多 100: 脉冲 I/O 模块数过多 101: 机架数过多 111: 机架上的单元数过多 当错误被清除时, 这些位也将被清除。	清除	清除	---

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A408	A408.00 ~ A408.07	基本 I/O 单元错误, 插槽号	当基本 I/O 单元发生错误时, A402.12 将会置 ON, 且在此处以二进制形式写入发生错误的插槽号。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 00 ~ 09 Hex (插槽 0 ~ 9) 当错误被清除时, 这些位也将被清除。	清除	清除	A402.12
	A408.08 ~ A408.15	基本 I/O 单元错误, 机架号	当基本 I/O 单元发生错误时, A402.12 将会置 ON, 且在此处以二进制形式写入发生错误的机架号。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 00 ~ 03 Hex (机架 0 ~ 3) 当错误被清除时, 这些位也将被清除。	清除	清除	A402.12
A409	A409.00 ~ A409.03	扩展机架号重复标志	当通过 CX-Programmer 设定扩展机架的起始地址时, 若两个机架所分配到的字是相同的或机架的起始地址超出了 CIO 0901, 相应的标志将会置 ON。位 00 ~ 03 对应机架 0 ~ 3。 ON: 两个不同的机架分配到了相同的字或机架的起始地址超出了 CIO 0901 OFF: 正常 当错误被清除时, 相应的标志也将被清除。	清除	清除	---
A410	A410.00 ~ A410.15	CPU 总线单元号重复标志	当 CPU 总线单元的单元号发生重复时, 重复出错标志 (A401.13) 和 A410 中的相应标志将会置 ON。位 00 ~ 15 对应单元 0 ~ F。 CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 ON: 检测到重复 OFF: 正常	清除	清除	A401.13
A411 ~ A416	A411.00 ~ A416.15	高性能 I/O 单元号重复标志	当高性能 I/O 单元的单元号发生重复时, 重复出错标志 (A401.13) 和 A411 ~ A416 中的相应标志将会置 ON。 位 00 ~ 15 对应单元 0 ~ F。 (位 A411.00 ~ A416.15 对应单元 000 ~ 05F(0 ~ 95)。) CPU 单元将停止运行且 CPU 单元前部的 ERR/ALM 指示灯将点亮。 若由于扩展机架的起始字设定导致扩展机架上的基本 I/O 单元占用了高性能 I/O 单元字, 则相应的位也将置 ON。 ON: 检测到重复 OFF: 正常	清除	清除	A401.13
A417	A417.00 ~ A417.15	CPU 总线单元和 CPU 总线单元之间的数据交换出现错误时, CPU 总线单元出错标志 (A402.07) 和发生错误的单元号对应的 A417 位均会置 ON。位 00 ~ 15 对应单元 0 ~ F。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常	清除	清除	A402.07	
A418 ~ A423	A418.00 ~ A423.15	高性能 I/O 单元错误, 单元号标志	当 CPU 单元和高性能 I/O 单元之间的数据交换出现错误时, 高性能 I/O 单元出错标志 (A402.06) 将会置 ON。 每个位对应一个单元号。A418.00 ~ A423.15 对应单元 0 ~ 95。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 出错 OFF: 正常 错误被清除时该标志将置 OFF。 如果无法确定当前的单元号, 则这些标志都不会置 ON。	清除	清除	A402.06
A426	A426.00 ~ A426.11	重复刷新出错单元号	当 A426.13 为 ON 时, 这些位将会包含执行了重复刷新的高性能 I/O 单元的单元号。 当错误被清除时, 这些位也将被清除。 单元号: 000 ~ 05F Hex(0 ~ 95) 错误被清除时该标志将置 OFF。	清除	清除	A402.13 A426.15
	A426.15	重复刷新出错原因	当 A402.13(重复刷新出错标志) 为 ON 时, 该标志将用于表示出错原因。CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 重复刷新	清除	清除	A402.13, A426.00 ~ A426.11

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A427	A427.00 ~ A427.15	CPU 总线单元设定错误, 单元号标志	当发生 CPU 总线单元设定错误时, A402.03 以及当前单元号对应的字中的位将会置 ON。位 00 ~ 15 对应单元 0 ~ F。 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。 ON: 设定错误 OFF: 正常	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	高功能 I/O 单元设定错误, 单元号标志	当发生高功能 I/O 单元设定错误时, A402.02 以及当前单元号对应的字中的位将会置 ON。位 00 ~ 15 对应单元 0 ~ F。 ON: 设定错误 OFF: 正常 CPU 单元将继续运行且 CPU 单元前部的 ERR/ALM 指示灯将闪烁。	清除	清除	接通电源或识别出 I/O 时刷新。 A402.02
A438	---	脉冲输出 2 停止出错代码	如果脉冲输出 2 发生脉冲输出停止错误, 则将出错代码写入该字。	保持	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。
A439	---	脉冲输出 3 停止出错代码	如果脉冲输出 3 发生脉冲输出停止错误, 则将出错代码写入该字。	保持	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。
A440	---	中断任务最长处理时间	包含中断任务的最长处理时间 (单位为 0.1ms) (执行完并清除处理时间最长的中断任务后, 在 PLC 开始运行时写入该值。) 当在 PLC 设置中启用了高速中断功能时被禁止。	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任务编号。8000 ~ 80FF Hex 对应任务 00 ~ FF。在发生中断时位 15 置 ON。 (执行完并清除处理时间最长的中断任务后, 在 PLC 开始运行时写入该值。) 当在 PLC 设置中启用了高速中断功能时被禁止。	清除	清除	见“功能”栏
A444	---	脉冲输出 0 停止出错代码	如果脉冲输出 0 发生脉冲输出停止错误, 则将出错代码写入该字。	保持	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A445	---	脉冲输出 1 停止出错代码	如果脉冲输出 1 发生脉冲输出停止错误, 则将出错代码写入该字。	保持	清除	<ul style="list-style-type: none"> 开始原点搜索时清除。 在原点搜索期间发生致命的脉冲输出错误时刷新。 当 PLC 设置中的脉冲输出的限位输入信号被设定为始终启用且脉冲输出因限位输入停止时刷新。 当限位输入被禁用且已存储致命脉冲输出错误时清除。
A446	A446.00 ~ A446.07	保护功能被禁用的次数	<ul style="list-style-type: none"> 每次当禁用保护功能失败 (由于在禁用密码保护时输入了错误的密码) 时进行计数。 显示 UM 保护及任务保护被禁用的总次数。 当该数值达到 255 (十进制) 时, 计数器停止计数。 当禁用了所有保护时, 计数器将被置为 00 Hex。 	---	---	---
A450	---	CIO 区指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A451	---	字区域指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A452	---	保持区指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A459	---	变址寄存器区指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A460	---	DM 区指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A461	---	EM 存储区块 0 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A462	---	EM 存储区块 1 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A463	---	EM 存储区块 2 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A464	---	EM 存储区块 3 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A465	---	EM 存储区块 4 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A466	---	EM 存储区块 5 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A467	---	EM 存储区块 6 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A468	---	EM 存储区块 7 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A469	---	EM 存储区块 8 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A470	---	EM 存储区块 9 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A471	---	EM 存储区块 A 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A472	---	EM 存储区块 B 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A473	---	EM 存储区块 C 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A474	---	EM 存储区块 D 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A475	---	EM 存储区块 E 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A476	---	EM 存储区块 F 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A477	---	EM 存储区块 10 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A478	---	EM 存储区块 11 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A479	---	EM 存储区块 12 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A480	---	EM 存储区块 13 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A481	---	EM 存储区块 14 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A482	---	EM 存储区块 15 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A483	---	EM 存储区块 16 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A484	---	EM 存储区块 17 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A485	---	EM 存储区块 18 指定	将某个地址指定为用于功能块的参数 (而非 I/O 变量 (VER_IN_OUT)) 时, 该字中将会存储一个用于实现区域指定的固定代码。该代码仅可被 OMRON 功能块库调用。请勿修改该字中的内容。	设定了固定值。	设定了固定值。	---
A10001 ~ A10003	---	生产批号	CPU 单元中存储了 BCD 数据格式的生产批号。可参照该区域来实现硬件识别。 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	---	---	---
A10100	A10100.00	单元同步运转服务标志	执行单元同步运转时为 ON。 ON: 单元同步运转执行中。 OFF: 未执行单元同步运转。	清除	清除	接通电源或单元重启后, 同步信号第二次置 ON 时刷新。
A10101	---	同步输入数据刷新出错代码	当 CPU 单元无法在指定时间内接收到来自同步单元的同步输入数据时, 将 0001 Hex 存储至该字。 0001 Hex: 出错 0000 Hex: 正常	清除	清除	刷新同步数据时
A10102	---	同步运转周期时间	这些字包含 PLC 设置中设定的同步运转周期时间。	保持	保持	当电源接通或单元重启时

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A10120 和 A10121	---	脉冲输出 0 频率	包含通过数据跟踪功能跟踪脉冲输出 0 时得出的脉冲输出 0 频率。 仅在设定了数据跟踪参数时有效。	---	清除	---
A10122 和 A10123	---	脉冲输出 1 频率	包含通过数据跟踪功能跟踪脉冲输出 1 时得出的脉冲输出 1 频率。 仅在设定了数据跟踪参数时有效。	---	清除	---
A10124 和 A10125	---	脉冲输出 2 频率	包含通过数据跟踪功能跟踪脉冲输出 2 时得出的脉冲输出 2 频率。 仅在设定了数据跟踪参数时有效。	---	清除	---
A10126 和 A10127	---	脉冲输出 3 频率	包含通过数据跟踪功能跟踪脉冲输出 3 时得出的脉冲输出 3 频率。 仅在设定了数据跟踪参数时有效。	---	清除	---
A10128 和 A10129	---	高速计数器 0 范围比较条件 1 ~ 32 的范围内标志	这些标志表示当高速计数器 0 在上下限范围比较模式下运行时, 当前值 (PV) 是否在范围 1 ~ 32 内。但是在达到 / 未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置 ON。 OFF: 不在范围内, ON: 在范围内 低位字中的位 00 ~ 15 对应范围 1 ~ 16。高位字中的位 00 ~ 15 对应范围 17 ~ 32。	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 针对 1 ~ 32 个范围执行比较时刷新。 当执行 PRV(881) 指令以读取范围比较结果时刷新。 当执行 INI(880) 指令以改变当前值 (PV) 或环形计数器最大值时刷新。 复位
A10130 和 A10131	---	高速计数器 1 范围比较条件 1 ~ 32 的范围内标志	这些标志表示当高速计数器 1 在上下限范围比较模式下运行时, 当前值 (PV) 是否在范围 1 ~ 32 内。但是在达到 / 未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置 ON。 OFF: 不在范围内, ON: 在范围内 低位字中的位 00 ~ 15 对应范围 1 ~ 16。高位字中的位 00 ~ 15 对应范围 17 ~ 32。	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 针对 1 ~ 32 个范围执行比较时刷新。 当执行 PRV(881) 指令以读取范围比较结果时刷新。 当执行 INI(880) 指令以改变当前值 (PV) 或环形计数器最大值时刷新。 复位
A10132 和 A10133	---	高速计数器 2 范围比较条件 1 ~ 32 的范围内标志	这些标志表示当高速计数器 2 在上下限范围比较模式下运行时, 当前值 (PV) 是否在范围 1 ~ 32 内。但是在达到 / 未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置 ON。 OFF: 不在范围内, ON: 在范围内 低位字中的位 00 ~ 15 对应范围 1 ~ 16。高位字中的位 00 ~ 15 对应范围 17 ~ 32。	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 针对 1 ~ 32 个范围执行比较时刷新。 当执行 PRV(881) 指令以读取范围比较结果时刷新。 当执行 INI(880) 指令以改变当前值 (PV) 或环形计数器最大值时刷新。 复位
A10134 和 A10135	---	高速计数器 3 范围比较条件 1 ~ 32 的范围内标志	这些标志表示当高速计数器 3 在上下限范围比较模式下运行时, 当前值 (PV) 是否在范围 1 ~ 32 内。但是在达到 / 未达到指定范围时, 无论是否设定了高速计数器来执行中断任务, 只要比较值在范围之内, 范围内标志均为置 ON。 OFF: 不在范围内, ON: 在范围内 低位字中的位 00 ~ 15 对应范围 1 ~ 16。高位字中的位 00 ~ 15 对应范围 17 ~ 32。	清除	清除	<ul style="list-style-type: none"> 在监控处理期间每个循环均刷新。 针对 1 ~ 32 个范围执行比较时刷新。 当执行 PRV(881) 指令以读取范围比较结果时刷新。 当执行 INI(880) 指令以改变当前值 (PV) 或环形计数器最大值时刷新。 复位
A10136 和 A10137	---	高速计数器 0 的环形计数最大值	包含高速计数器 0 用作环形计数器时得出的环形计数最大值。若使用线性模式, 则将这些值清为 0。 低 4 位: A10136, 高 4 位: A10137	清除	清除	当执行 INI(880) 指令以改变环形计数器最大值时刷新。
A10138 和 A10139	---	高速计数器 1 的环形计数最大值	包含高速计数器 1 用作环形计数器时得出的环形计数最大值。若使用线性模式, 则将这些值清为 0。 低 4 位: A10138, 高 4 位: A10139	清除	清除	当执行 INI(880) 指令以改变环形计数器最大值时刷新。

地址		名称	功能	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位					
A10140 和 A10141	---	高速计数器2的环形计数最大值	包含高速计数器2用作环形计数器时得出的环形计数最大值。若使用线性模式, 则将这些值清为0。 低4位: A10140, 高4位: A10141	清除	清除	当执行 INI(880) 指令以改变环形计数器最大值时刷新。
A10142 和 A10143	---	高速计数器3的环形计数最大值	包含高速计数器3用作环形计数器时得出的环形计数最大值。若使用线性模式, 则将这些值清为0。 低4位: A10142, 高4位: A10143	清除	清除	当执行 INI(880) 指令以改变环形计数器最大值时刷新。
A10144 和 A10145	---	中断输入0锁存的当前值 (PV)	当中断输入0进行输入时, 将存储脉冲输出0的当前值 (PV) 或高速计数器输入0的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10144, 高4位: A10145	清除	清除	发生输入中断时
A10146 和 A10147	---	中断输入1锁存的当前值 (PV)	当中断输入1进行输入时, 将存储脉冲输出1的当前值 (PV) 或高速计数器输入1的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10146, 高4位: A10147	清除	清除	发生输入中断时
A10148 和 A10149	---	中断输入2锁存的当前值 (PV)	当中断输入2进行输入时, 将存储脉冲输出2的当前值 (PV) 或高速计数器输入2的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10148, 高4位: A10149	清除	清除	发生输入中断时
A10150 和 A10151	---	中断输入3锁存的当前值 (PV)	当中断输入3进行输入时, 将存储脉冲输出3的当前值 (PV) 或高速计数器输入3的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10150, 高4位: A10151	清除	清除	发生输入中断时
A10152 和 A10153	---	中断输入4锁存的当前值 (PV)	当中断输入4进行输入时, 将存储脉冲输出4的当前值 (PV) 或高速计数器输入4的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10152, 高4位: A10153	清除	清除	发生输入中断时
A10154 和 A10155	---	中断输入5锁存的当前值 (PV)	当中断输入5进行输入时, 将存储脉冲输出5的当前值 (PV) 或高速计数器输入5的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10154, 高4位: A10155	清除	清除	发生输入中断时
A10156 和 A10157	---	中断输入6锁存的当前值 (PV)	当中断输入6进行输入时, 将存储脉冲输出6的当前值 (PV) 或高速计数器输入6的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10156, 高4位: A10157	清除	清除	发生输入中断时
A10158 和 A10159	---	中断输入7锁存的当前值 (PV)	当中断输入7进行输入时, 将存储脉冲输出7的当前值 (PV) 或高速计数器输入7的当前值 (PV)。启动中断输入任务前的当前值 (PV) 将被读取和存储。 低4位: A10158, 高4位: A10159	清除	清除	发生输入中断时

注 CJ 系列 PLC 的特殊只读区中提供了下列标志, 可根据表中的说明进行指定。但辅助区中未包含这些标志。详情请参考 “6-21 条件标志” 和 “6-22 时钟脉冲”。

出错标志	条件标志区
访问出错标志	
进位标志	
大于标志	
等于标志	
小于标志	
负标志	
上溢标志	
下溢标志	
大于或等于标志	
不等于标志	
小于或等于标志	
常 ON 标志	
常 OFF 标志	
0.02s 时钟脉冲	时钟脉冲区
0.1s 时钟脉冲	
0.2s 时钟脉冲	
1s 时钟脉冲	
1min 时钟脉冲	

A-3-2 读 / 写区 (由用户设定)

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位						
A500	A500.12	IOM 保持位	将此位置 ON, 以便在 PROGRAM 模式和 RUN/MONITOR 模式之间切换时保持 I/O 存储器的状态。(如果在 PLC 设置中保存了 I/O 保持位本身的状态 (IOM 保持位状态), 则将在接通 PLC 电源或断电时保持 I/O 存储区的状态。)	ON: 保持 OFF: 不保持	保持	清除 (可在 PLC 设置中设定为保持状态)	见“功能”栏 PLC 设置 (IOM 保持位状态设定)
	A500.13	强制状态保持位	在 PROGRAM 模式和 MONITOR 模式之间进行切换或接通电源时, 将该位置 ON 可保持强制置位或强制复位的位状态。请务必与 IOM 保持位 (A500.12) 组合使用。将 IOM 保持位 (A500.12) 与该位同时置 ON, 即可在断电后保持相关的状态。 若要在 PLC 通电时保持该位的状态, 则必须将 PLC 设置中的“ Forced Status Hold Bit Startup Hold Setting ”(强制状态保持位启动保持设定) 参数设定为启用。	ON: 保持 OFF: 不保持	保持	清除 (可在 PLC 设置中设定为保持状态)	见“功能”栏 PLC 设置 (强制状态保持位状态设定)
	A500.14	出错日志复位位	此位置 ON 以将出错日志指针 (A300) 复位为 00。出错日志区自身的内容 (A100 ~ A199) 不会被清除。在出错日志指针复位后, 该位将自动复位为 0。	OFF → ON: 清除	保持	清除	A100 ~ A199, A300
	A500.15	输出 OFF 位	将该位置 ON 时, 基本 I/O 单元的所有输出都将变为 OFF, 而与 I/O 存储器中的输出位状态无关 (与发生致命错误时的情况相同)。此时, CPU 单元前部的 INH 指示灯将亮起。 注: 该位的状态将在断电后保持。 CIO 区中分配到高性能 I/O 单元的输出位状态 (如外部模拟量输出) 取决于高性能 I/O 单元本身的功能。	---	保持	保持	---
A501	A501.00 ~ A501.15	CPU 总线单元重启位	将这些位置 ON, 以重新启动 (初始化) 相应单元号的 CPU 总线单元。位 00 ~ 15 对应单元 0 ~ F。 当重启位置 ON 时, 相应的 CPU 总线单元初始化标志 (A302.00 ~ A302.15) 将会置 ON。初始化完成后, 重启位和初始化标志都会自动置 OFF。	OFF → ON: 重启 ON → OFF: 重启完成单元重启后通过系统置 OFF。	保持	清除	A302.00 ~ A302.15
A502 ~ A507	A502.00 ~ A507.15	高性能 I/O 单元重启位	将这些位置 ON, 以重新启动 (初始化) 相应单元号的高性能 I/O 单元。位 A502.00 ~ A507.15 对应单元 0 ~ 95。 当重启位置 ON 时, 相应的高性能 I/O 总线单元初始化标志 (A330.00 ~ A335.15) 将会置 ON。初始化完成后, 重启位和初始化标志都会自动置 OFF。	OFF → ON: 重启 ON → OFF: 重启完成单元重启后通过系统置 OFF。	保持	清除	A330.00 ~ A335.15
A508	A508.09	微分监控完成标志	在微分监控执行期间建立了微分监控条件时置 ON。 在微分监控启动时此标志将清零。	ON: 监控条件已建立 OFF: 尚未建立	保持	清除	---
	A508.11	跟踪触发器监控标志	通过跟踪起始位 (A508.14) 建立触发条件时为 ON。通过采样起始位 (A508.15) 启动下一次数据跟踪时为 OFF。	ON: 触发条件已建立 OFF: 尚未建立或不执行跟踪	保持	清除	---
	A508.12	跟踪已完成标志	在执行跟踪的过程中完成了某一跟踪存储区的采样后为 ON。 当下一次采样起始位 (A508.15) 从 OFF 变为 ON 时为 OFF。	ON: 跟踪已完成 OFF: 未执行跟踪或跟踪执行中	保持	清除	---
	A508.13	跟踪忙碌标志	当采样起始位 (A508.15) 从 OFF 变为 ON 时为 ON。当跟踪完成后为 OFF。	ON: 跟踪执行中 OFF: 未执行跟踪 (未采样)	保持	清除	---
	A508.14	跟踪起始位	将该位从 OFF 变为 ON, 以建立触发条件。延迟 (正值或负值) 中指定的偏移量用于确定哪些数据采样是有效的。	ON: 跟踪触发条件已建立 OFF: 尚未建立	保持	清除	---
	A508.15	采样起始位	通过 CX-Programmer 将该位从 OFF 变为 ON 以启动数据跟踪时, PLC 将以下列任意一种方式将数据存储至跟踪存储器。 1. 以固定间隔 (10 ~ 2,550ms) 进行数据采样。 2. 在程序中执行 TRSM(045) 时进行数据采样。 3. 在每个循环的末尾进行数据采样。 仅可通过 CX-Programmer 来控制 A50815 的操作。	OFF → ON: 开始数据跟踪 (采样) 通过编程设备置 ON。	保持	清除	---
A510 和 A511	---	启动时间	这些字中包含电源接通的时间。每次电源接通时均更新该内容。数据以 BCD 码格式存储。 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	保持	见“功能”栏	电源接通时刷新。

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位						
A512 和 A513	---	电源中断时间	这些字中包含电源中断时间。每次断电时均更新该内容。数据以 BCD 码格式存储。 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) 启动时不清除这些字中的内容。	见“功能”栏	保持	保持	断电时写入
A514	---	电源中断次数	包含从电源第一次接通时起的断电次数。数据以二进制格式存储。若要将该值复位, 请用 0000 Hex 覆盖当前值。启动时不会清除该字, 但在存储器损坏检测标志 (A395.11) 置 ON 时清除。	0000 ~ FFFF Hex	保持	保持	电源接通时刷新。 A395.11
A515 ~ A517	---	运行开始时间	此处以 BCD 格式存储将运行模式切换至 RUN 或 MONITOR 模式以开始运行的时间。 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) 上次运行开始的时间会在电源接通至开始运行的时间段内存储。	见“功能”栏	保持	保持	见“功能”栏
A518 ~ A520	---	运行结束时间	此处以 BCD 格式存储将运行模式切换至 PROGRAM 模式以停止运行的时间。 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) 如果在运行过程中发生错误, 则将存储出错时间。如果之后运行模式切换至 PROGRAM 模式, 则将存储进入 PROGRAM 模式的时间。	见“功能”栏	保持	保持	见“功能”栏
A523	---	总通电时间	包含 PLC 的总通电时间 (以 10 小时为单位)。该数据以二进制形式存储并每隔 10 小时更新一次。若要将该值复位, 请用 0000 Hex 覆盖当前值。 启动时不会清除该字, 但在存储器损坏检测标志 (A395.11) 置 ON 时清为 0000。	0000 ~ FFFF Hex	保持	保持	---
A526	A526.00	串行端口重启位	将该位置 ON 以重启 RS-232C 端口。(请勿在端口运行于外设总线模式时使用该位。) 重启处理完成后, 此位自动置 OFF。	OFF → ON: 重启	保持	清除	---
A527	A527.00 ~ A527.07	联机编辑禁止位确认标志	联机编辑禁止位 (A527.09) 仅在该字节包含 5A 的情况下有效。 若要通过 CX-Programmer 禁用联机编辑, 则应将该字节设定为 5A 并将 A527.09 置 ON。 (联机编辑指当 PLC 在 MONITOR 模式下运行时进行的程序修改和补充操作。)	5A: 启用 A527.09 其它值: 禁用 A527.09	保持	清除	A527.09
	A527.09	联机编辑禁止位	将该位置 ON 以禁用联机编辑。该位的设定仅在 A527.00 ~ A527.07 被设为 5A 的情况下有效。	ON: 禁用 OFF: 未禁用	保持	清除	A527.00 ~ A527.07
A528	A528.00 ~ A528.07	串行端口出错标志	这些标志表示 RS-232C 发生的错误类型。当 RS-232C 端口重启时, 这些标志将自动置 OFF。 (这些标志在串行网关模式下有效, 而在外设总线模式下无效。标志中的位 5 在 NT 链接模式下有效。) 仅下列位在串行 PLC 链接模式下有效。 主站单元: 位 5: 发生超时错误时置 ON 从站单元: 位 3: 发生成帧错误时置 ON 位 4: 发生超限错误时置 ON 位 5: 发生超时错误时置 ON 这些位可由 CX-Programmer 来清除。	位 0 和 1: 不使用 位 2: 发生校验错误时置 ON 位 3: 发生成帧错误时置 ON 位 4: 发生超限错误时置 ON 位 5: 发生超时错误时置 ON 位 6 和 7: 不使用	保持	清除	---

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位						
A529	---	用于模拟系统错误的 FAL/FALS 号	设置一个虚拟的 FAL/FALS 号以使用 FAL(006) 或 FALS(007) 指令来模拟系统错误。 在执行 FAL(006) 或 FALS(007) 时, 如果 A529 中的编号与指令操作数中指定的编号相同, 则将生成指令操作数中所给出的系统错误而非用户定义的错误。	0001 ~ 01FF Hex: FAL/FALS 号 1 ~ 511 0000 或 0200 ~ FFFF Hex: 无用于模拟系统错误的 FAL/FALS 号。(将不产生错误。)	保持	清除	---
A530	---	电源中断禁用区设定	将其设定为 A5A5 Hex 以在 DI(693) 和 EI(694) 指令之间禁用电源中断 (断电中断任务除外)。	A5A5 Hex: 禁用电源中断处理功能 其它: 启用电源中断处理功能	清除	清除	---
A531	A531.00	高速计数器 0 复位位	如果复位方式设为 Z 相信号 + 软件复位, 则在此位置 ON 后接收到 Z 相信号时, 对应高速计数器的当前值 (PV) 将被复位。 如果复位方式设为软件复位, 则在其中的一位位置 ON 后, 对应高速计数器的当前值 (PV) 将在循环中被复位。	OFF → ON: 复位 OFF → ON: 保持当前值 (PV) ON → OFF: 更新当前值 (PV)	保持	清除	---
	A531.01	高速计数器 1 复位位					
	A531.02	高速计数器 2 复位位					
	A531.03	高速计数器 3 复位位					
	A531.08	高速计数器 0 门控位	当其中的一位位置 ON 时, 将不会进行脉冲输入计数, 也不会更改相应的高速计数器的当前值 (PV)。				
	A531.09	高速计数器 1 门控位	将某一位置 OFF 时, 将再次开始计数并更新高速计数器的当前值 (PV)。若将复位方式设定为 Z 相信号 + 软件复位且相应的高速计数器复位位 (A531.00 ~ A531.03) 为 ON, 则门控位无效。				
	A531.10	高速计数器 2 门控位					
	A531.11	高速计数器 3 门控位					
A532	---	中断计数器 0 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 0 对此处设定的脉冲进行计数时, 将执行中断任务 140。	0000 ~ FFFF Hex	保持	清除	---
A533	---	中断计数器 1 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 1 对此处设定的脉冲进行计数时, 将执行中断任务 141。	0000 ~ FFFF Hex	保持	清除	---
A534	---	中断计数器 2 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 2 对此处设定的脉冲进行计数时, 将执行中断任务 142。	0000 ~ FFFF Hex	保持	清除	---
A535	---	中断计数器 3 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 3 对此处设定的脉冲进行计数时, 将执行中断任务 143。	0000 ~ FFFF Hex	保持	清除	---
A536	---	中断计数器 0 的计数当前值	包含计数器模式下输入中断用中断计数器的当前值 (PV)。在递增模式下, 计数值每次递增 1 直至达到计数设定值, 然后计数值归零; 在递减模式下, 计数值 (从设定值开始) 每次递减 1 直至计数值变为 0, 然后再回到预设值。	0000 ~ FFFF Hex	保持	清除	---
A537	---	中断计数器 1 的计数当前值					
A538	---	中断计数器 2 的计数当前值					
A539	---	中断计数器 3 的计数当前值					
A540	A540.00	脉冲输出 0 复位位	当此位置 ON 时, 脉冲输出 0 的当前值 (PV) (包含在 A276 和 A277 中) 将被清除。	---	保持	清除	---
	A540.08	脉冲输出 0 CW 限位输入信号标志	这是用于原点搜索的脉冲输出 0 的 CW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
	A540.09	脉冲输出 0 CCW 限位输入信号标志	这是用于原点搜索的脉冲输出 0 的 CCW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位						
A541	A541.00	脉冲输出 1 复位位	当此位置 ON 时, 脉冲输出 1 的当前值 (PV)(包含在 A278 和 A279 中) 将被清除。	---	保持	清除	---
	A541.08	脉冲输出 1 CW 限位输入信号标志	这是用于原点搜索的脉冲输出 1 的 CW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
	A541.09	脉冲输出 1 CCW 限位输入信号标志	这是用于原点搜索的脉冲输出 1 的 CCW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
A542	A542.00	脉冲输出 2 复位位	当此位置 ON 时, 脉冲输出 2 的当前值 (PV)(包含在 A322 和 A323 中) 将被清除。	---	保持	清除	---
	A542.08	脉冲输出 2 CW 限位输入信号标志	这是用于原点搜索的脉冲输出 2 的 CW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
	A542.09	脉冲输出 2 CCW 限位输入信号标志	这是用于原点搜索的脉冲输出 2 的 CCW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
A543	A543.00	脉冲输出 3 复位位	当此位置 ON 时, 脉冲输出 3 的当前值 (PV)(包含在 A324 和 A325 中) 将被清除。	---	保持	清除	---
	A543.08	脉冲输出 3 CW 限位输入信号标志	这是用于原点搜索的脉冲输出 3 的 CW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
	A543.09	脉冲输出 3 CCW 限位输入信号标志	这是用于原点搜索的脉冲输出 3 的 CCW 限位输入信号。若要使用此信号, 请将来自实际传感器的输入作为输入条件写入梯形图程序, 并将结果输出到此标志。				
A544	---	中断计数器 4 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 4 对此处设定的脉冲进行计数时, 将执行中断任务 144。	0000 ~ FFFF Hex	保持	清除	---
A545	---	中断计数器 5 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 5 对此处设定的脉冲进行计数时, 将执行中断任务 145。	0000 ~ FFFF Hex	保持	清除	---
A546	---	中断计数器 6 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 6 对此处设定的脉冲进行计数时, 将执行中断任务 146。	0000 ~ FFFF Hex	保持	清除	---
A547	---	中断计数器 7 的计数设定值	该字用于计数器模式下的输入中断。请在执行中断任务前设定计数值。当中断计数器 7 对此处设定的脉冲进行计数时, 将执行中断任务 147。	0000 ~ FFFF Hex	保持	清除	---
A548	---	中断计数器 4 的计数当前值	包含计数器模式下输入中断用中断计数器的当前值 (PV)。在递增模式下, 计数值每次递增 1 直至达到计数设定值, 然后计数值归零; 在递减模式下, 计数值 (从设定值开始) 每次递减 1 直至计数值变为 0, 然后再回到预设值。	0000 ~ FFFF Hex	保持	清除	---
A549	---	中断计数器 5 的计数当前值					
A550	---	中断计数器 6 的计数当前值					
A551	---	中断计数器 7 的计数当前值					
A580	A580.00 ~ A580.03	FB 通信指令重试次数	可自动存储 PLC 设置中设定的 FB 通信指令重试次数。用户无法写入这些辅助区字 / 位。尤其在通过 OMRON FB 库中的功能块执行 FINS 报文或 DeviceNet 显式报文通信时, 用户必须在 PLC 设置的 FB 通信指令设定中设定重新发送次数和响应监控时间。在 PLC 设置的 OMRON FB 库设定项中设定的值将会被自动存储到相关的辅助区字 (A580 ~ A582) 中, 并可由 OMRON FB 库中的功能块进行调用。	0 ~ F Hex	根据 PLC 设置中的设定	清除	运行开始时写入
A581	---	FB 通信指令响应监控时间	可自动存储 PLC 设置中设定的 FB 通信指令响应监控时间。用户无法写入这些辅助区字 / 位。尤其在通过 OMRON FB 库中的功能块执行 FINS 报文或 DeviceNet 显式报文通信时, 用户必须在 PLC 设置的 FB 通信指令设定中设定重新发送次数和响应监控时间。在 PLC 设置的 OMRON FB 库设定项中设定的值将会被自动存储到相关的辅助区字 (A580 ~ A582) 中, 并可由 OMRON FB 库中的功能块进行调用。	0001 ~ FFFF Hex (单位: 0.1s, 范围: 0.1 ~ 6553.5) 0000 Hex: 2s	根据 PLC 设置中的设定	清除	运行开始时写入

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位						
A582	---	FB DeviceNet 通信指令响应监控时间	可自动存储 PLC 设置中设定的 FB DeviceNet 通信指令响应监控时间。 在执行 FAL(006) 或 FALS(007) 时, 如果 A529 中的编号与指令操作数中指定的编号相同, 则将生成指令操作数中所给出的系统错误而非用户定义的错误。	0001 ~ FFFF Hex (单位: 0.1s, 范围: 0.1 ~ 6553.5) 0000 Hex: 2s	根据 PLC 设置中的设定	清除	运行开始时写入
A595 和 A596	---	后台执行的 IR00 输出	当为某个后台执行的指令指定了一个变址寄存器作为输出时, A595 和 A596 将取代 IR00 接收输出。	0000 0000 ~ FFFF FFFF Hex (A596 包含最高数位。)	清除	清除	---
A597	---	后台执行的 DR00 输出	当为某个后台执行的指令指定了一个数据寄存器作为输出时, A597 将取代 DR00 接收输出。	0000 ~ FFFF Hex	清除	清除	---
A598	A598.00	FPD 学习位	将该位置 ON 以通过学习功能自动设定监控时间。 当 A598.00 为 ON 时, FPD(269) 将对执行条件变为 ON 后诊断输出所需的置 ON 时间进行测量。如果测量时间超过监控时间, 测量时间将在乘以 1.5 倍后进行存储, 以作为新的监控时间。 学习功能仅在为监控时间操作数指定了字地址的情况下使用。	ON: 学习监控时间 OFF: 学习功能关闭	清除	清除	---
	A598.01	后台执行的等于标志	当后台执行的 SRCH(181) 指令找到了匹配的数据时置 ON。	ON: 在表中找到了搜索数据 OFF: 未找到搜索数据	清除	清除	---
A600 ~ A603	---	宏区输入字	执行 MCRO(099) 指令时, 将复制此处的四个指定源字 (从输入参数首字开始) 中的内容, 并将其传送到子程序中。	输入数据: 4 字	清除	清除	---
A604 ~ A607	---	宏区输出字	执行了 MCRO(099) 指令中指定的子程序之后, 子程序的结果将从 A604 ~ A607 传送到指定的目的字。(输出参数数字)	输出数据: 4 字	清除	清除	---
A619	A619.02	串行端口设定修改标志	当正在对 RS-232C 端口的通信设定进行更改时为 ON。执行 STUP(237) 时, 该标志将会置 ON; 而在设定更改完成后, 该标志又将置 OFF。	ON: 修改执行中 OFF: 未执行修改	保持	清除	---

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间/相关标志, 设定															
字	位																					
A620	A620.01	通信单元 0, 端口 1 设定更改标志	当正在更改特定端口的设定时, 相应的标志将会置 ON。执行 STUP(237) 时, 该标志将会置 ON; 而在设定更改完成后, 串行通信单元发出的事件又会使该标志置 OFF。 通过将这些标志置 ON, 用户可识别出针对串行端口设定的更改。	ON: 修改执行中 OFF: 未执行修改	保持	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 设定更改标志					---															
A650	A650.00 ~ A650.07	程序替换已完成标志	<ul style="list-style-type: none"> 正常完成 (即 A650.14 为 OFF) <ul style="list-style-type: none"> 01 Hex: 已替换程序文件 (OBJ)。 出错 (即 A650.14 为 ON) <ul style="list-style-type: none"> 00 Hex: 发生了致命错误。 01 Hex: 发生了存储器错误。 11 Hex: 存储器写保护。 12 Hex: 程序替换验证密码错误。 21 Hex: 未插入存储卡。 22 Hex: 指定文件不存在。 23 Hex: 指定文件的大小超出了当前型号允许的容量 (存储器错误)。 31 Hex: 正在执行下列任意操作时。 <ul style="list-style-type: none"> 正在执行与文件存储器相关的操作。 正在写入用户程序。 正在切换运行模式。 	---	保持	清除	---															
	A650.14	替换出错标志	当替换起始位 (A650.15) 为 ON 时, 该标志置 ON 表示替换程序时发生了错误。当替换起始位再次变为 ON 时, 替换出错标志将会置 OFF。	ON: 替换出错 OFF: 替换操作正常或替换起始位 (A650.15) 为 ON	保持	清除	---															
	A650.15	替换起始位	在程序密码 (A651) 有效 (A5A5 Hex) 的情况下, 程序替换操作将在替换起始位置 ON 后开始。在替换程序的过程中, 请勿将该位置 OFF。 当接通电源或程序替换完成时, 替换起始位将会置 OFF, 而与替换过程是否正常完成或出错无关。 可使用 CX-Programmer、PT 或上位计算机确认是否正在通过读取替换起始位执行程序替换过程。	ON: 程序已替换 OFF: 替换已完成或处于电源接通后的状态	保持	清除	---															
A651	---	程序密码	输入替换程序时所需的验证密码。 A5A5 Hex: 启用替换起始位 (A650.15)。 任何其它值: 禁用替换起始位 (A650.15)。 当接通电源或程序替换完成时, 替换起始位将会置 OFF, 而与替换过程是否正常完成或出错无关。	---	保持	清除	---															
A654 ~ 657	---	程序文件名	开始替换程序时, 将以 ASCII 形式存储程序文件名。指定的文件名长度可达 8 个字符 (不含扩展名)。 文件名的存储顺序为: 从 A654 ~ A657 (低位字到高位字) 的高位字节到低位字节。如果文件名长度小于 8 个字符, 则应将空格代码 (20 Hex) 填入低位剩余字节及高位位的剩余字中。文件名中不可使用无效字符或空格字符。 例如: 当文件名为 ABC.OBJ 时 <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td style="text-align: center;">15</td> <td style="text-align: center;">0</td> </tr> <tr> <td>A654</td> <td style="text-align: center;">41</td> <td style="text-align: center;">42</td> </tr> <tr> <td>A655</td> <td style="text-align: center;">43</td> <td style="text-align: center;">20</td> </tr> <tr> <td>A656</td> <td style="text-align: center;">20</td> <td style="text-align: center;">20</td> </tr> <tr> <td>A657</td> <td style="text-align: center;">20</td> <td style="text-align: center;">20</td> </tr> </table>		15	0	A654	41	42	A655	43	20	A656	20	20	A657	20	20	---	保持	清除	---
	15	0																				
A654	41	42																				
A655	43	20																				
A656	20	20																				
A657	20	20																				

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间 / 相关标志, 设定
字	位						
A720 ~ A722	---	通电时钟数据 1	这些字中包含与字 A510 ~ A511 中存储的启动时间相同的时间数据以及月份和年份信息。 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	保持	保持	电源接通时写入。
A723 ~ A725	---	通电时钟数据 2	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第一次接通的时间。 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	保持	保持	电源接通时写入。
A726 ~ A728	---	通电时钟数据 3	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第二次接通的时间。 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	保持	保持	电源接通时写入。
A729 ~ A731	---	通电时钟数据 4	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第三次接通的时间。 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	保持	保持	电源接通时写入。
A732 ~ A734	---	通电时钟数据 5	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第四次接通的时间。 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	保持	保持	电源接通时写入。
A735 ~ A737	---	通电时钟数据 6	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第五次接通的时间。 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	保持	保持	电源接通时写入。
A738 ~ A740	---	通电时钟数据 7	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第六次接通的时间。 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	保持	保持	电源接通时写入。

地址		名称	功能	设定	模式变更后的状态	启动时的状态	写入时间/相关标志, 设定
字	位						
A741 ~ A743	---	通电时钟数据 8	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第七次接通的时间。 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	保持	保持	电源接通时写入。
A744 ~ A746	---	通电时钟数据 9	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第八次接通的时间。 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	保持	保持	电源接通时写入。
A747 ~ A749	---	通电时钟数据 10	这些字中包含在字 A510 ~ A511 中存储的启动时间之前电源第九次接通的时间。 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	保持	保持	电源接通时写入。

A-3-3 辅助区操作的详细信息

A100 ~ A199: 出错日志区

A100	出错代码	
A101	出错标志内容	
A102	分	秒
A103	日	时
A104	年	分
⋮		
A195	出错代码	
A196	出错标志内容	
A197	分	秒
A198	日	时
A199	年	分

出错记录

出错记录

如果 1998 年 4 月 1 日的 17:10:30 在 PLC 设置中发生存储器错误 (出错代码: 80F1), 出错记录中将会产生下列数据 (04 Hex)。

80	F1
00	04
10	30
01	17
98	04

如果 1997 年 5 月 2 日的 8:30:15 发生编号为 001 的 FALS 错误, 出错记录中将会产生下列数据。

C1	01
00	00
30	15
02	08
97	05

出错代码和出错标志

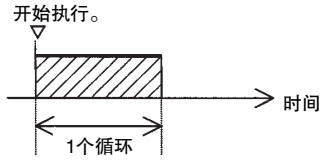
数据类型	出错代码	错误名称	出错标志
系统产生的致命错误	0x809F	循环时间超长错误	---
	80C0 ~ 80C7、80CE、80CF	I/O 总线错误	A404
	80E0	I/O 设定错误	---
	80E1	I/O 点数过多错误	A407
	80E9	单元号重复错误	A410、A411 ~ A416 ^{*1}
	80EA	机架号重复错误	A409
	80F0	程序错误	A295 ~ A299 ^{*2}
	80F1	存储器错误	A403
	80F2	版本错误	---
	80F4	存储卡传送错误	---
用户自定义的致命错误	C101 ~ C2FF	FALS 指令执行 ^{*3}	---
系统产生的非致命错误	008B	重复刷新错误	A426
	009A	基本 I/O 单元错误	A408
	009B	PLC 设置错误	A406
	00D1	选件板错误	---
	00F1	备份存储器错误	A315.15
	00F2	标签存储器错误 ^{*4}	A315.12
	00F3	系统工作存储器错误	A402.00
	00F7	电池错误	---
	0200 ~ 020F	CPU 总线单元错误	A417
	0300 ~ 035F	高性能 I/O 单元	A418 ~ A423 ^{*5}
	0400 ~ 040F	CPU 总线单元设置错误	A427
	0500 ~ 055F	高性能 I/O 单元设置错误	A428 ~ A433 ^{*5}
	用户自定义的非致命错误	4101 ~ 42FF	FAL 指令执行 ^{*6}

注 1 编号重复错误对应的出错标志内容如下:

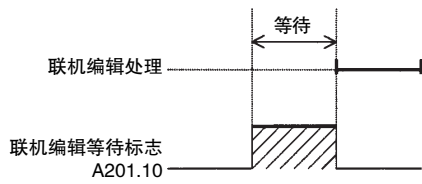
- 位 0 ~ 7: 单元号 (二进制), 高性能 I/O 单元为 00 ~ 5F Hex, CPU 总线单元为 00 ~ 0F Hex
- 位 8 ~ 14: 全零
- 位 15: 单元类型 OFF: CPU 总线单元
 ON: 高性能 I/O 单元

- 2 只有 A295 中的内容被存储为程序错误对应的出错标志内容。
- 3 存储的 C101 ~ C2FF 对应 FALS 号 001 ~ 511。
- 4 仅 CJ2H-CPU6□-EIP 和 CJ2M-CPU3□ 支持此项功能。
- 5 0000 Hex 将被存储为出错标志内容。
- 6 存储的 4101 ~ 42FF 数值对应 FAL 号 001 ~ 511。

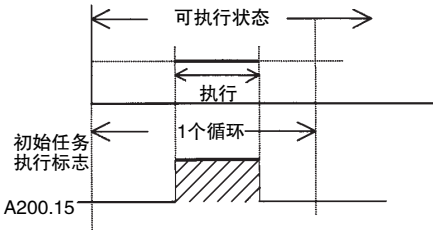
A200.11: 首循环标志



A201.10: 联机编辑等待标志

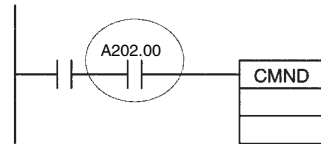
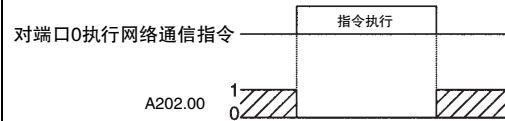
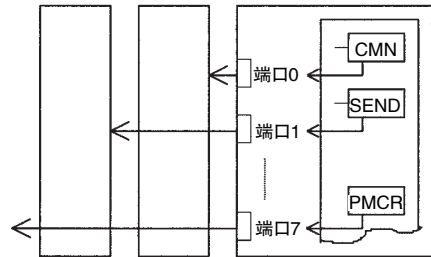


A200.15: 初始任务标志



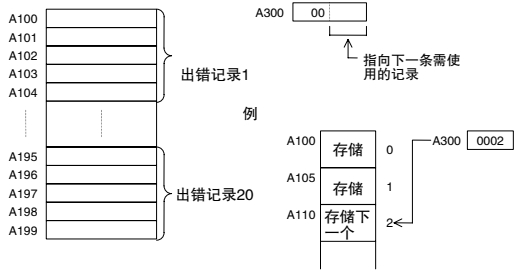
在首次执行任务的情况下，当任务达到可执行状态时，A200.15 将会置 ON。该标志仅在任务执行过程中保持 ON 状态，而在后续循环中将不再保持 ON 状态。

A202.00 ~ A202.07: 通信端口允许标志

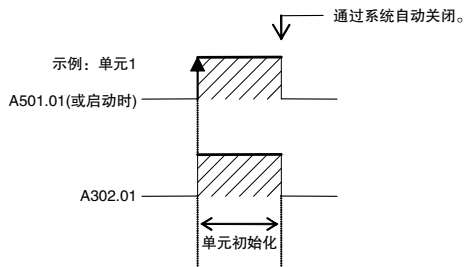


编程时，应确保仅在A202.00为ON的情况下执行CMND(490)。

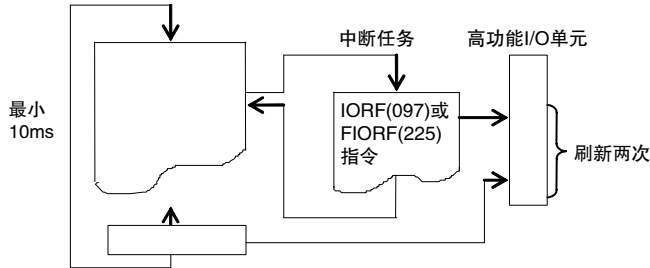
A300: 出错记录指针



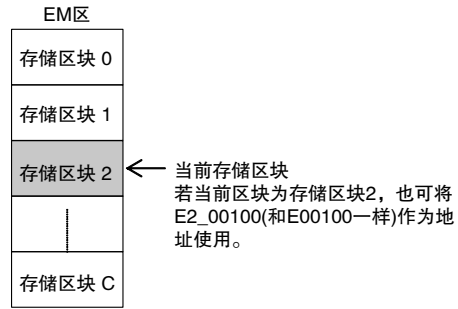
A501.01 ~ A501.15: CPU 总线单元重启位



A426.15: 重复刷新出错原因



A301: 当前 EM 存储区块



A401.09: 程序错误

	错误	地址
当 A401.09 为 ON 时的程序错误	UM 上溢出错标志	A295.15
	非法指令标志	A295.14
	分配上溢出错标志	A295.13
	任务出错标志	A259.12
	无 END(001) 出错标志	A295.11
	非法区访问出错标志	A295.10
	间接 DM/EM 寻址出错标志	A295.09

A-4 PLC 存储器地址的存储器映射

A-4-1 PLC 存储器地址

变址寄存器 (IR00 ~ IR15) 中设定了用于对 I/O 存储器进行间接寻址的 PLC 存储器地址。正常情况下, 可使用传送至寄存器 (MOVR(560)) 和传送定时器 / 计数器的当前值 (PV) 至寄存器指令 (MOVRW(561)) 将 PLC 存储器地址置入变址寄存器中。

数据搜索 (SRCH(181))、寻找最大值 (MAX(182)) 和寻找最小值 (MIN(183)) 等部分指令可将处理结果输出到变址寄存器, 以表示 PLC 存储器地址。

此外, 还有部分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。这些指令包括双字传送 (MOVL(498))、一些符号比较指令 (=L、<>L、<L、>L、<=L 和 >=L)、双字比较 (CMPL(060))、双字数据交换 (XCGL(562))、双字二进制递增 (++L(591))、双字二进制递减 (--L(593))、双字带符号无进位二进制加 (+L(401))、双字带符号无进位二进制减 (-L(411))、设定记录位置 (SETR(635)) 和获取记录位置 (GETR(636))。

所有 PLC 存储器地址是连续的且用户必须知道存储区的顺序和边界。作为参考, 本附录末尾的表中列出了 PLC 存储器地址。



正确使用注意事项

应尽量避免在程序中直接设定 PLC 存储器地址。若在程序中设定了 PLC 存储器地址, 程序与新 CPU 单元型号或存储器布局已更改的 CPU 单元的兼容性将会降低。



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。

- 当使用偏移值间接指定地址时。
- 当在数组变量中间接指定元素号时。

编程时应确保不超出数据区范围或考虑存储器映射中的数据区分配顺序。

A-4-2 存储器映射

下表所示为 I/O 存储区的 PLC 存储器地址。I/O 存储区通过电池进行数据备份。如果未安装电池，则可能会损坏 I/O 存储区中的数据。

注 切勿访问为系统保留的区域。

类型	PLC 存储器地址 (Hex)	用户地址	区位
I/O 存储区	0000 ~ 0B7FF		为系统保留
	0B800 ~ 0B801	TK0000 ~ TK0031	任务标志区
	0B802 ~ 0B83F		为系统保留
	0B840 ~ 0B9FF	A000 ~ A447	只读辅助区
	0BA00 ~ 0BDFE	A448 ~ A1471	读 / 写辅助区
		A10000 ~ A11535	只读辅助区
	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	定时器当前值 (PV)
	0F000 ~ 0FFFF	C0000 ~ C4095	计数器当前值 (PV)
	10000 ~ 17FFF	D00000 ~ D32767	DM 区
	18000 ~ 1FFFF	E0_00000 ~ E0_32767	EM 存储区块 0
	20000 ~ 27FFF	E1_00000 ~ E1_32767	EM 存储区块 1(Hex)
	:	:	:
	D8000 ~ DFFFF	E18_00000 ~ E18_32767	EM 存储区块 18(Hex)
:	:		
F8000 ~ FFFFF	E_00000 ~ E_32767	EM 区, 当前存储区块 (见注)	

注 当前在程序中指定的 EM 存储区块内容被存储在这些地址中。例如, 如果指定了存储区块 1, 则会将与 20000 ~ 27FFF 中相同的内容存储到 F8000 ~ FFFFF 中。

A-5 断电时的操作

A-5-1 断电操作

若 CPU 单元断电，将执行下列处理。当 CPU 单元在 RUN 模式或 MONITOR 模式下时，如果电源电压降至额定电压的 85%(CJ1W-PD025 直流电源单元：80%，CJ1W-PD022 直流电源单元：90%) 以下，将会执行断电处理。

1. CPU 单元停止运行。
2. 输出单元上的所有输出均置 OFF。

注 1 无论 PLC 设置中的 IOM 保持位的状态或启动时的 IOM 保持位设定如何，所有输出均会置 OFF。

- 2 交流电源单元额定电压的 85%：
100VAC：85VAC
200VAC：170VAC
100 ~ 240V 输入(宽幅)：85VAC
- 3 直流电源单元额定电压的 80%/90%：
CJ1W-PD025：19.2VDC
CJ1W-PD022：21.6VDC

电源瞬时中断

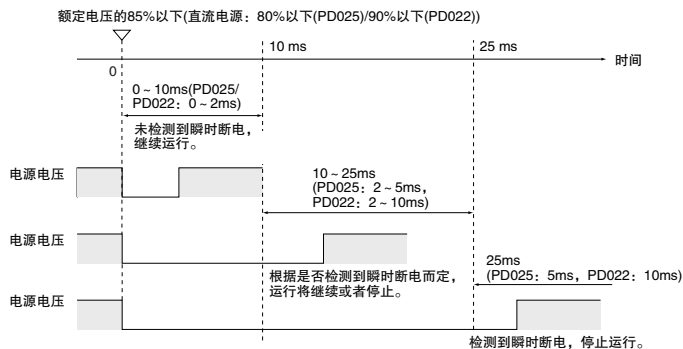
若电源仅发生了瞬时掉压，将执行以下处理(电源瞬时中断)。

- 若瞬时电源中断时间小于 10ms，系统将无条件持续运行。也就是说，当电源电压从最小额定电压的 85% 以下恢复到 85% 或以上所需的时间不超过 10ms。

注 使用 DC 电源时，CJ1W-PD025 的电压从额定电压的 80% 以下恢复到 80% 或以上或 CJ1W-PD022 的电压从额定电压的 90% 以下恢复到 90% 或以上所需的时间不超过 2ms。

- 由于电源瞬时中断一般持续 10ms ~ 25ms(使用 DC 电源时，CJ1W-PD025 和 CJ1W-PD022 分别为 2 ~ 5ms 和 2 ~ 10ms) 之间，因此很难进行测定，会为电源中断的检测带来不确定性。
- 若电源瞬时中断时间大于 25ms(使用 DC 电源时，CJ1W-PD025 和 CJ1W-PD022 分别为“大于 5ms”和“大于 10ms”)，则系统将无条件停止。

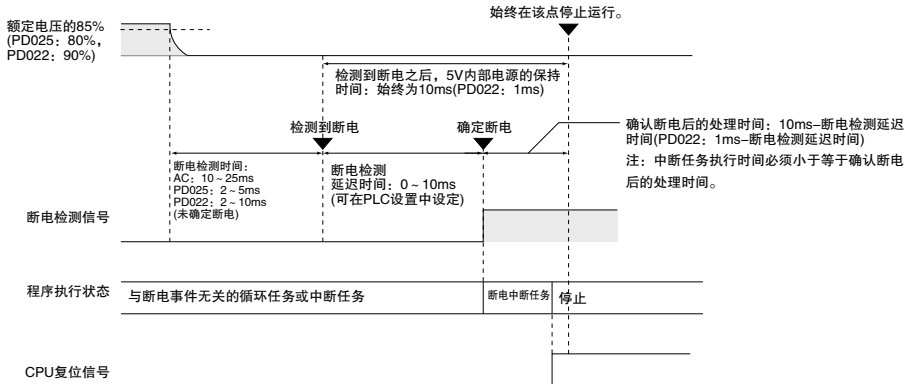
因此，若在上述第 2 和第 3 条中给出的条件下停止运行，则检测电源中断所需的时间为 10 ~ 25ms(使用 DC 电源时，CJ1W-PD025 和 CJ1W-PD022 分别为 2 ~ 5ms 和 2 ~ 10ms)，停止运行所需的时间(或开始执行断电中断任务所用的时间)可通过在 PLC 设置中设定“Power OFF Detection Delay Time”(断电检测延迟时间)(0 ~ 10ms)来进行延时。发生断电时，CJ1W-PD022 电源单元 5VDC 输出的保持时间为 1ms，因此将无法使用断电中断任务禁止和断电检测时间参数。



注 以上时序图为断电检测时间被设定为 0ms 时的一个示例。

断电时序图

下列时序图对 CPU 单元断电时的操作进行了更详细的说明。



断电检测时间： 当电源电压降至最小额定电压值的 85% (CJ1W-PD025 DC 电源单元和 CJ1W-PD022 DC 电源单元分别为 80% 和 90%) 以下时，检测到断电所需的时间。

断电检测延迟时间： 指从检测到断电到确认断电之间的延迟时间。在 PLC 设置中，该时间可设定为 0 ~ 10ms。对于因为电源不稳定引起的电源中断，则可在 PLC 设置中设定较长的断电检测延迟时间 (最大 10ms)。CJ1W-PD022 电源单元仅支持 1ms 的保持时间，因此该设定不可用。

电源保持时间： 电源掉电后 5V 内部电源的最大保持时间 (固定为 10ms)。断电中断任务的执行时间不能超过 10ms 减去断电检测延迟时间所得的值 (确认断电后的处理时间)。超过这段时间后，即使断电中断任务没有完成也会被结束。CJ1W-PD022 电源单元仅支持 1ms 的保持时间，因此该设定不可用。

操作说明

- 在断电检测时间 (AC 电源单元: 10 ~ 25ms, CJ1W-PD025 DC 电源单元: 2 ~ 5ms, CJ1W-PD022 DC 电源单元: 2 ~ 10ms) 内，若 100 ~ 240VDC 电源的电压降至最小额定电压的 85% (CJ1W-PD025 DC 电源单元: 80%, CJ1W-PD022 DC 电源单元: 90%) 以下，将检测到断电。
- 如果在 PLC 设置中设定了断电检测延迟时间 (0 ~ 10ms)，则在超过此时间后执行以下操作：
 - 若断电中断任务 (默认的 PLC 设置设定) 被禁止，CPU 复位信号将置 ON，且 CPU 单元将立即复位。
 - 若电源掉电中断任务 (在 PLC 设置中) 被启用，则 CPU 复位信号将在断电中断任务执行后置 ON，并复位 CPU 单元。确保断电中断任务在 (10ms- 断电检测延迟时间 = 电源掉电后的处理时间) 内执行完毕。检测到断电后，5V 内部电源将仅保持 10ms。发生断电时，CJ1W-PD022 电源单元内部 5V 电源的保持时间为 1ms，因此将无法使用断电检测延迟时间和断电中断任务禁止功能。

A-5-2 断电的指令执行

如果在 CPU 单元运行在 RUN 或 MONITOR 模式下时发生电源中断并且中断得到确认，则先完成当前正在执行的指令*1，然后再执行下述电源中断处理。

- 如果未启用电源 OFF 中断任务，则将立即复位 CPU 单元。
- 如果已启用电源 OFF 中断任务，则将执行该任务，然后立即复位 CPU 单元。

可在 PLC 设置中启用和禁用断电中断任务。

*1 仅当完成执行所需的时间小于等于断电检测后的处理时间 (10ms- 断电检测延迟时间) 时，才可完成当前指令的执行。若指令未在这段时间内执行完毕，则指令将被中断并执行上述处理。

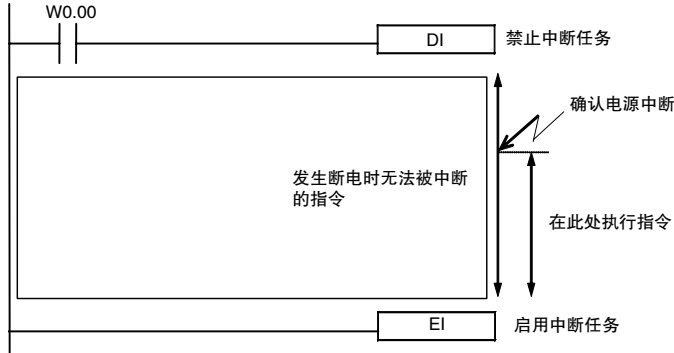
注 安装了 CJ1W-PD022 时，检测到断电后的处理时间为 1ms。

在程序中禁止电源中断处理

如果禁止了断电中断任务，可确保程序区不受电源中断的影响。因此，即使发生断电，指令仍将在 CPU 单元执行电源中断处理之前被执行。该功能可通过禁止中断 (DI(693)) 和允许中断 (EI(694)) 指令实现。

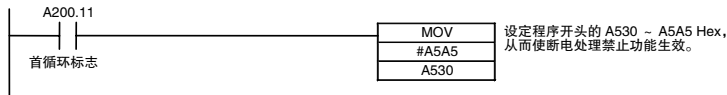
请按照下列步骤使用。

1. 在需要保护的程序段前插入 DI(693) 禁止中断，并在其后使用 EI(694) 指令来启用中断。



2. 在 A530 ~ A5A5 Hex 中设置断电中断禁止设定，以启用断电处理禁止功能。

注 A530 在断电后通常会被清除。为避免这种情况，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，或在程序开头包括以下类型的指令以将 A530 设定为 A5A5 Hex。



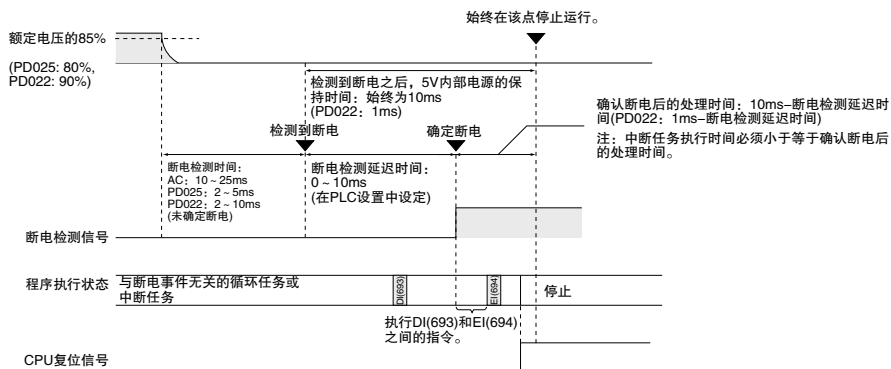
3. 在 PLC 设置中禁用断电中断任务。

利用上述步骤，在 DI(693) 和 EI(694) 之间执行指令时，即使发生电源中断，也将在执行电源 OFF 中断之前完成 *1DI(693) 和 EI(694)(或 END) 之间的所有指令。

*1 仅当完成执行所需的时间小于等于断电检测后的处理时间 (10ms- 断电检测延迟时间) 时，才可完成受保护指令的执行。若指令未在这段时间内执行完毕，则指令将被中断并执行上述处理。

注 1 安装了 CJ1W-PD022 时，检测到断电后的处理时间为 1ms。

- 2 若未在 PLC 设置中禁用断电中断任务，将会执行断电中断任务。一旦检测到断电，CPU 将会复位而不执行受保护指令。
- 3 若在执行 DI(693) 时检测到电源中断，CPU 单元将会复位而不执行受保护指令。



若将 A530(电源中断禁止区设定) 设定为除 A5A5 Hex 以外的其它值以启用屏蔽电源中断处理, 则在当前指令执行完成后将启动电源中断处理。



附加信息

中断处理将根据下表所示的 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(启用屏蔽电源中断处理)	其它值(禁用屏蔽电源中断处理)
断电中断任务 (PLC 设置)	禁用	执行 DI(693) 和 EI(694) 之间的所有指令并复位 CPU 单元。	当前指令执行完成并复位 CPU 单元。
	启用	当前指令执行完成后执行断电中断任务, 并复位 CPU 单元。	

A-6 在 Windows XP(SP2 或以上)、Windows Vista 或 Windows 7 中进行 EtherNet/IP 连接

Windows XP(SP2 及以上)、Windows Vista 和 Windows 7 系统采用了更严密的防火墙安全机制，其中增加了对 Ethernet 端口数据通信的限制。通过计算机上的 EtherNet 端口对 CPU 单元进行 EtherNet/IP 连接时，必须修改 Windows 防火墙的设置才能启用 CX-Programmer 通信。



正确使用注意事项

Windows 防火墙主要用于保护系统避免受到外部设备（如通过因特网）的非法访问。本节中对如何更改 Windows 防火墙的设定进行了说明，目的是帮助用户启用 CX-Programmer 所需的 EtherNet/IP 连接。若在公司网络中或其它网络中使用同一台计算机，则应事先确认所做的更改不会影响网络安全。本节所述的更改操作仅适用于需通过 Ethernet 端口进行 EtherNet/IP 连接的情况。若通过其它端口进行连接（如 USB 端口），则无需更改设定。

A-6-1 更改 Windows 防火墙设定

Windows XP

1. 尝试通过 Ethernet 端口将 CX-Programmer 连接到 EtherNet/IP 网络中的 PLC 时，将显示 Windows 安全警报对话框。
2. 点击“Unblock”（取消阻止）按钮。

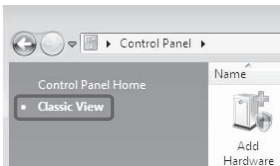


将接受来自 CX-Programmer 的 EtherNet/IP 连接，EtherNet/IP 连接会在随后启用。

Windows Vista 或 Windows 7

尝试从 CX-Programmer 进行连接前，请按照下述步骤更改设定。此时可能会显示用户帐号控制对话框。请点击“Continue”（继续）按钮并转至下一个步骤。

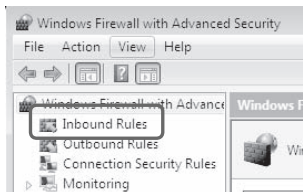
1. 在 Windows 开始菜单中选择“Control Panel”（控制面板），并将显示切换为“Classic View”（经典视图）。



2. 打开管理工具并从下图所示的对话框中选择“Windows Firewall with Advanced Security”（高级安全 Windows 防火墙）。



3. 在本地计算机的“Windows Firewall with Advanced Security”（高级安全 Windows 防火墙）下方左侧的设定框中选择“Inbound Rules”（入站规则）。



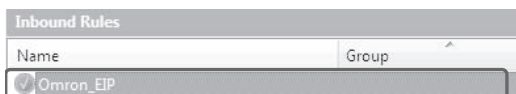
4. 在对话框右侧的动作区中选择“New Rule”（新建规则）。



5. 在“New Inbound Rule”（新建入站规则）向导对话框中每一步中进行下述设定，然后点击“Next”（下一步）按钮来切换至下一步。

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。 <div style="border: 1px solid gray; padding: 2px; width: fit-content;"> Protocol type: <input type="text" value="ICMPv4"/> Protocol number: <input type="text" value="1"/> </div>
范围	一律选择“Any IP address”（任意 IP 地址）。
措施	选择“Allow the connection”（允许连接）。
情况	选择“Domain”（域）、“Private”（专用）和“Public”（公开）。
名称	输入任意名称，如 Omron_EIP。

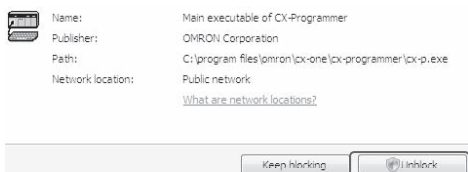
6. 点击“Finish”（完成）按钮。您所定义的规则将注册在“Inbound Rules”（入站规则）中（如 Omron_EIP）。



7. 关闭“Windows Firewall with Advanced Security”（高级安全 Windows 防火墙）对话框。

8. 尝试通过 Ethernet 端口将 CX-Programmer 连接到 EtherNet/IP 网络中的 PLC 时，将显示 Windows 安全警报对话框。

9. 点击“Unblock”（取消阻止）按钮。



将接受来自 CX-Programmer 的 EtherNet/IP 连接，EtherNet/IP 连接会在随后启用。

A-7 PLC比较表：CJ系列和CS系列PLC

下表列出了CJ2 CPU单元和其它CS/CJ系列CPU单元之间的功能差异。

功能比较

项目	CJ系列					CS系列	
	CJ2 CPU单元	CJ2M 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	外形尺寸 (高×深×宽)	CJ2H-CPU6□-EIP: 90×65×80mm CJ2H-CPU6□: 90×65×40mm	CJ2H-CPU1□: 90×75×31mm CJ2M-CPU3□: 90×75×62mm	90×65×62mm		90×65×31mm 90×65×49mm	130×101×71mm
	重量	CJ2H-CPU6□-EIP: 280g以下 CJ2H-CPU6□: 190g以下	CJ2H-CPU1□: 130g以下 CJ2M-CPU3□: 190g以下(不带串行 选件板)	200g以下		120g以下 170g以下	350g以下
	5VDC时的电流 消耗	CJ2H-CPU6□-EIP: 0.82A CJ2H-CPU6□: 0.42A	CJ2H-CPU1□: 0.5A CJ2M-CPU3□: 0.7A	0.99A		0.58A 0.64A	0.82A
	抗振性	符合JIS 60068-2-6	符合JIS 60068-2-6	符合JIS C0040			
	抗冲击性	符合JIS 60068-2-27	符合JIS 60068-2-27	符合JIS C0041			
性能规格	最大程序容量	400K步	60K步	250K步		20K步	250K步
	最大I/O容量	2,560点				640点	5,120点
	系统开销时间	CJ2H-CPU6□-EIP: 200μs(含EtherNet/IP 端口开销时间) CJ2H-CPU6□: 100μs	CJ2M-CPU3□: 270μs CJ2M-CPU1□: 160μs*1	130μs	300μs	500μs	300μs
	基本指令执行时间 (LD)	0.016μs	0.04μs	0.016μs	0.02μs	0.1μs	0.02μs
	特殊指令执行时间 (MOV)	0.048μs	0.06μs	0.14μs	0.18μs	0.3μs	0.18μs
	即时刷新指令执行 时间(LLD)	0.99μs	1.26μs	21μs		24μs, 28μs	21μs
	中断任务启动时间	26μs(1.0版: 30μs)	31μs	40μs	124μs	169μs	124μs
	基本I/O单元刷新 (16点单元)	1.4μs	3.9μs		3μs		
高性能I/O单元刷新 (8路模拟量输入)	50μs	70μs		120μs	160μs	120μs	

*1 在装有脉冲I/O模块的情况下，应增加下列时间：10μs×脉冲I/O模块数。为CJ2M-CPU3□CPU单元使用EtherNet/IP标签数据链接时，应增加下列时间：100μs+传送字数×1.8μs。

项目	CJ 系列					CS 系列	
	CJ2 CPU 单元	CJ2M CPU 单元	CJ1-H-R CPU 单元	CJ1-H CPU 单元	CJ1-M CPU 单元	CS1-H CPU 单元	
性能规格	CPU 总线单元刷新 (通过 EtherNet/IP 单元进行标签数据链接)	100 μs+(传送字数 × 0.33μs)	100 μs+(传送字数 × 0.7μs)	200 μs+(传送字数 × 0.7μs)		270 μs+(传送字数 × 0.7μs)	200 μs+(传送字数 × 0.7μs)
	最大 EM 区容量	32K 字 × 25 存储区块	32K 字 × 4 存储区块	32K 字 × 13 存储区块			
	源代码/注释存储器	3.5MB	1,024KB	2MB			
	功能块定义、动作及转换程序	2,048	CJ2M-CPU□1/CPU□2/ CPU□3; 256 CJ2M-CPU□4/CPU□5; 2,048	1,024*2	128	1,024*2	
	适用存储卡	30、64、128、256 或 512MB	128MB 256MB 512MB	30、64 或 128MB			
	循环任务数	128 个		32 个			
	通信端口(内部逻辑端口)	8+64 个		8 个			
	因联机编辑导致的循环时间延长量	约 1ms		约 8ms	约 14ms	约 8ms	
	扩展机架的最大连接数	3 个				7 个	
	中断输入单元的最大连接数	2 个				4 个	
	CPU 总线单元的最大连接数	CJ2H-CPU6□-EIP; 15 个 CJ2H-CPU6□; 16 个	CJ2M-CPU1□; 16 个 CJ2M-CPU3□; 15 个	16 个			
	内部板	不适用				适用	

*2 CPU□3H 或 CPU□2H: 128。

项目	CJ 系列					CS 系列
	CJ2 CPU 单元	CJ2M 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		正常模式或并行处理模式		
	单元同步运转	支持 (1.1 版或更高版本)	不支持	不支持		
	特定 CPU 总线单元的特殊指令	支持 (1.1 版或更高版本)	支持	不支持		
	最小循环时间	0.2 ~ 32,000ms (以 0.1ms 为增量单位) (可在运行过程中修改最小循环时间 (1.1 版或更高版本))。	0.2 ~ 32,000ms (以 0.1ms 为增量单位) (可在运行过程中修改最小循环时间 (仅限 CJ2M))。	1 ~ 32,000ms (以 1ms 为增量单位)		
	监控循环时间单位	0.01ms		0.1ms		
	用于设定定时中断的时间单位	0.1、1 或 10ms		1 或 10ms	0.1、1 或 10ms	1 或 10ms
	通信端口	USB、RS-232C 和 EtherNet/IP (仅限 CJ2H-CPU6□-EIP)	USB、RS-232C 和 RS-422 A / 485 (CJ2M-CPU3□ 需使用串行选件板) EtherNet/IP (仅限 CJ2M-CPU3□)	外设和 RS-232C		
	串行 PLC 链接	不支持	支持	不支持	支持	不支持
	标签名称服务器	CJ2H-CPU6□-EIP: 支持 CJ2H-CPU6□: 不支持	CJ2M-CPU3□: 支持 CJ2M-CPU1□: 不支持	无		
	外部中断插槽	CJ2H-CPU6□-EIP: 插槽 0 ~ 3 CJ2H-CPU6□: 插槽 0 ~ 4	插槽 0 ~ 4	插槽 0 ~ 4		插槽 0 ~ 9
	数据跟踪	32K 字以下, 采样时间: 1 ~ 2,550ms, 支持连续跟踪	8K 字, 采样时间: 1 ~ 2,550ms, 支持连续跟踪	4K 字, 采样时间: 10 ~ 2,550ms		
	通过产品批号实现写保护	支持	支持	不支持		
	PLC 名称	支持	支持	不支持		
	对 DM 和 EM 区中的位进行寻址	支持	支持	不支持		
	EM 区强制置位 / 复位	如果在 1.2 版或更高版本的单元中使用自动地址分配, 则支持 EM 区强制置位 / 复位功能。	支持 EM 区强制置位 / 复位功能	不支持		
	将 EM 区的一部分转换为跟踪存储器	支持	支持	不支持		
	指定地址偏移	支持	支持	不支持		
	对同一个程序中的定时器指令进行刷新 (BCD 和二进制)	支持	支持	不支持		

项目	CJ 系列					CS 系列
	CJ2 CPU 单元	CJ2M 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	中继网络表(路由表的一种)中的入口数		最多 128 个		最多 32 个	
时钟脉冲	0.1ms、1ms、0.01s、0.02s、0.1s、0.2s、1s 和 1min			0.02s、0.1s、0.2s、1s 和 1min		
脉冲 I/O 执行方式	CPU 单元本身不支持	CJ2M-MD211 和 CJ2M-MD212 脉冲 I/O 模块 (带 40 针 MIL 连接器, 最多 2 个模块) 支持漏型或源型输出 (取决于脉冲 I/O 模块型号)	CPU 单元本身不支持		支持 (仅限带内置 I/O 的型号) 仅限漏型输出	CPU 单元本身不支持
普通输入	不支持	最多 20 点 (每个脉冲 I/O 模块 10 点)	不支持		10 点	不支持
输入中断	不支持	最多 8 点 (每个脉冲 I/O 模块 4 点) 最小输入信号脉冲宽度: 30μs	不支持		最多 4 点 最小输入信号脉冲宽度: 30μs	不支持
快速响应输入	不支持	最多 8 点 (必须包含快速响应、高速计数器及普通输入的总和) (每个脉冲 I/O 模块 4 点)	不支持		最多 4 点 (必须包含快速响应、高速计数器及普通输入的总和)	不支持
高速计数器输入	不支持	最多 4 点 (高速计数器 0/1/2/3) (每个脉冲 I/O 模块 2 点)	不支持		最多 2 点 (高速计数器 0/1)	不支持
普通输出	不支持	最多 12 点 (每个脉冲 I/O 模块 6 点)	不支持		6 点	不支持
脉冲输出	不支持	最多 4 点 (每个脉冲 I/O 模块 2 点)	不支持		2 点	不支持
PWM 输出	不支持	最多 4 点 (每个脉冲 I/O 模块 2 点)	不支持		CJ1M-CPU22/23: 2 点 CJ1M-CPU21: 1 点	不支持
原点搜索/原点返回	不支持	最多 4 轴 (每个脉冲 I/O 模块 2 轴)	不支持		2 轴	不支持

A-8 各单元版本支持的功能

A-8-1 CJ2H CPU 单元

1.3 版或更高版本所支持的功能

若要使用 1.3 版单元新增的功能，则需使用 CX-Programmer 9.1 版或更高版本。

单元		CJ2H CPU 单元			
型号		CJ2H-CPU6□-EIP 和 CJ2H-CPU6□			
单元版本		1.3 版或更高版本	1.2 版	1.1 版	1.0 版
部分高功能 I/O 单元的特殊指令	CJ1W-NC281/NC481/NC881 位置控制单元: PCU 高速定位 (NCDMV(218))	支持	不支持	不支持	不支持
	CJ1W-NC281/NC481/NC881 位置控制单元: PCU 定位触发器 (NCDTR(219))	支持	不支持	不支持	不支持
新增特殊指令	带符号区域范围比较: ZCPS(088)	支持	不支持	不支持	不支持
	双字带符号区域范围比较: ZCPSL(116)	支持	不支持	不支持	不支持

1.2 版或更高版本

若要使用 1.2 版单元新增的功能，则需使用 CX-Programmer 8.3 版或更高版本。

单元		CJ2H CPU 单元	
型号		CJ2H-CPU6□-EIP CJ2H-CPU6□	
单元版本		1.2 版或更高版本	1.2 版以前的版本
EM 区强制置位 / 复位		支持	不支持

注 使用 1.2 版或更高版本 CJ2H CPU 单元功能的用户程序无法兼容 1.1 版或更早版本的 CJ2H CPU 单元。如果尝试通过 CX-Programmer 向 1.1 版或更早版本的 CPU 单元传送包含这些功能的程序，将发生错误，导致程序无法下载至 CPU 单元。

1.1 版或更高版本所支持的功能

若要使用为 1.1 版单元新增的功能，则需使用 CX-Programmer 8.1 版* 或更高版本。

单元	CJ2H CPU 单元		
型号	CJ2H-CPU6□-EIP 和 CJ2H-CPU6□		
功能	单元版本	1.1 版	1.0 版
高速中断功能 缩短中断任务的开销处理时间 执行定时中断任务的最小间隔时间: 0.1ms		支持	不支持
在运行过程中改变最小循环时间设定		支持	不支持
单元同步运转		支持	不支持
为部分高性能 I/O 单元增设的即时刷新指令 CJ1W-AD042 高速模拟量输入单元: 模拟量输入直接转换 (AIDC(216)) CJ1W-DA042V 高速模拟量输出单元: 模拟量输出直接转换 (AODC(217))		支持	不支持
高速串行通信单元 CJ1W-SCU22/SCU32/SCU42 高速串行通信单元: 通过串行通信单元直接接收 (DRXDU(261)) 和通过串行通信单元直接传送 (DTXDU(262))		支持	不支持

* 若要使用高速中断功能及在运行过程中改变最小循环时间设定，需采用 CX-Programmer 8.02 版或更高版本。

注 使用 1.1 或更高版本 CJ2H CPU 单元功能的用户程序无法兼容 1.0 或更早版本的 CJ2H CPU 单元。如果尝试通过 CX-Programmer 向 1.0 版或更早版本的 CPU 单元传送包含这些功能的程序，将发生错误，导致程序无法下载至 CPU 单元。如果将包含这些功能的程序文件 (扩展名: .OBJ) 传送到 1.0 版的 CPU 单元，将在运行开始或启用功能时发生程序错误，从而导致 CPU 单元停止运行。

A-8-2 CJ2M CPU 单元

2.0 版或更高版本所支持的功能

若要使用 2.0 版单元新增的功能，则需使用 CX-Programmer 9.12 版或更高版本。

单元	CJ2M CPU 单元		
型号	CJ2M-CPU□□		
功能	单元版本	2.0 版或更高版本	2.0 版以前的版本
脉冲输出		支持	不支持

索引

A

安全功能	10-46
通过 DIP 开关实现写保护	10-46
通过密码实现读保护	10-46
针对 FINS 命令的写保护	10-52
安全注意事项	21

B

保持区	6-2, 6-20
备份存储器	10-36
备份服务	3-5
备份恢复操作	7-11
本地网络表	4-11
编程语言	5-4
变址寄存器	5-86, 6-2, 6-34
标签	5-51
不检测低电池电压	9-8
步数	A-79
步梯形图段	5-110

C

CIO 区	6-2, 6-7
CPU 总线单元区	6-2
DeviceNet 区	6-2
高性能 I/O 单元区	6-2
I/O 区	6-2
内部 I/O 区	6-2
数据链接区	6-2
CPU 处理模式	9-30
CPU 单元的设置	9-8
CPU 总线单元	
I/O 分配	6-14, 6-25
CPU 总线单元的事件服务	3-5
CPU 总线单元区	6-2, 6-14
CPU 总线单元设置	2-3
CPU 总线单元设置区	4-5, 4-15
参数区	2-3
参数时间	10-5
参数文件	7-8
测试输入	10-60
插槽起始地址设定	8-11
程序 / 网络符号文件	7-7
程序错误	5-104
程序段	
步梯形图段	5-110
FOR-NEXT 循环	5-110
IL-ILC 段	5-110
JMPO-JME0 段	5-110
块程序段	5-110
子程序	5-110
程序容量	5-5
程序索引	2-3
程序索引文件	7-9
程序执行	3-4
重试计数	9-13
串行端口设定	9-23

串行 PLC 链接	11-21
串行 PLC 链接的响应性能	12-14
串行 PLC 链接区	6-2, 6-17
存储出错停止位	10-70
存储器	
参数区	2-3
CPU 总线单元设置	2-3
I/O 表	2-3
路由表	2-3
PLC 名称	2-3
PLC 设置	2-3
I/O 存储区	2-3
网络符号 (标签)	2-3
用户程序区	2-3
源和注释区	2-3
程序索引	2-3
符号表	2-3
源代码	2-3
注释	2-3
存储器管理功能	10-36
存储器中存储的时间数据	
电源中断时间	10-4
通电时钟数据	10-4
用户程序和参数时间	10-5
运行开始和结束时间	10-5
总通电时间	10-5
存储卡	7-2, 7-5
存储卡识别时间	7-5
错误	
程序错误	5-104
非法指令错误	5-103
用户程序区溢出错误	5-103
错误仿真功能	5-101

D

DeviceNet 区	6-2, 6-18
DIP 开关	4-2
单元备份文件	7-9
单元管理功能	10-32
单元同步运转	10-75
当前存储区块	6-28
地址偏移	5-95
电源 OFF 检测延时	9-17
电源 OFF 中断任务	5-22, 9-18
电源中断时的处理	3-7
电源中断时间	10-4
掉电检测延迟设定	10-22
定时器区	6-2, 6-30
定时器当前值 (PV) 区	6-2
定时器完成标志区	6-2
定时中断间隔	9-16
定时中断任务	5-24
段	5-40
断电时的操作	A-151

E

EM 文件存储器	7-2
EM 文件存储器功能	10-37

F

FALS (007) 指令	10-72
FAL (006) 指令	10-71
FB 程序区	2-3
FB 中的 DeviceNet 通信指令设定	9-13
FB 中的通信指令设定	9-13
FINS 保护	9-31
FOR-NEXT 循环	5-110
FPD (269) 指令	10-73
非法指令错误	5-103
符号	5-47
符号表	2-3, 5-47
符号表文件	7-9
辅助区	6-2, 6-21, A-108

G

高功能 I/O 单元	
I/O 分配	6-25
高功能 I/O 单元的事件服务	3-5
高功能 I/O 单元的循环刷新	9-19
高功能 I/O 单元区	6-2, 6-15
高速输入	10-9
高速中断功能	10-18
跟踪存储器	6-27
功能块	5-42
工作区	6-2, 6-19
故障报警指令	10-71

H

恒定循环时间	9-15
后台执行	10-10
后台执行设定	9-12

I

I/O 表	2-3, 4-4, 8-3
手动分配	4-9
自动分配	4-9
I/O 存储区	2-3, 6-2
I/O 分配	8-2
手动分配	8-8
自动分配	8-4
I/O 模块	9-32
I/O 区	6-2
I/O 刷新	3-6, 6-7, 6-9
I/O 刷新时间	12-7
I/O 刷新时序	5-81
I/O 响应时间	12-12
I/O 中断任务	5-26
IL-ILC 段	5-110
IOM 保持位的启动保持设定	9-6
IORF (097) 刷新	6-8, 6-10

J

JMP0-JME0 段	5-110
基本 I/O 单元的机架响应时间	9-21
集成仿真	5-100
即时刷新	5-82, 6-8, 6-9
计数器区	6-2, 6-32
计数器当前值 (PV)	6-32
计数器当前值 (PV) 区	6-2
计数器完成标志	6-32
计数器完成标志区	6-2
检查	3-4
检查程序	5-98
监视循环时间	9-14, 10-9
简易备份操作	7-11
简易备份文件	7-13
将时间设定为所有事件	9-30
禁止电源掉电中断	10-23
禁止电源 OFF 中断	9-18
禁止 SIOU 循环刷新	9-19
经由网络的 FINS 写保护设定	9-31
局部符号	5-48, 5-50

K

块程序段	5-110
扩展数据存储区	6-2, 6-26

L

联机编辑	10-61
联机编辑服务	3-5
链接区	6-12
路由表	2-3, 4-5, 4-10

M

MONITOR 模式	3-8
脉冲 I/O 区	6-2, 6-16
模拟系统错误	10-72

N

NT 链接系统	11-18
内部 I/O 区	6-2
内置 RAM	2-2
内置闪存存储器	2-2

P

PLC 备份工具	7-13
PLC 存储器地址的存储器映射	A-149
PLC 名称	2-3, 4-4, 10-56
PLC 设置	2-3, 4-4, 4-7, 9-2
CPU 处理模式	9-30
CPU 单元的设置	9-8
操作模式	9-7
串行端口设定	9-23
电源 OFF 检测延时	9-17
定时中断间隔	9-16
FB 中的通信指令设定	9-13
FINS 保护	9-31

高性能 I/O 单元的循环刷新	9-19
后台执行设定	9-12
I/O 模块	9-32
基本 I/O 单元的机架响应时间	9-21
机架响应时间	9-21
IOM 保持位的启动保持设定	9-6
监视循环时间	9-14
将“Time”（时间）设定为“All Events” （所有事件）	9-30
禁止电源 OFF 中断	9-18
禁止 SIOU 循环刷新	9-19
经由网络的 FINS 写保护设定	9-31
启动保持设定	9-5
启动运行设定	9-5
强制状态保持位的启动保持设定	9-5
时间和中断设定	9-14
通信设定	9-23
外设服务	9-30
循环时间	9-15
执行过程	9-8
执行设定	9-7
PROGRAM 模式	3-8

Q

启动保持设定	9-5
启动初始化	3-2
启动设定和维护	10-20
启动时的自动传送	7-11
启动时自动传送	10-25
启动时自动传送的文件	7-13
启动运行设定	9-5
强制置位 / 复位	10-59
强制状态保持位的启动保持设定	9-5
全局符号	5-48, 5-50

R

RUN 模式	3-8
RUN 输出	10-24
任务	5-11
任务标志	6-33
任务标志区	6-2
任务控制指令	5-17

S

SFC 语言	5-9
ST 语言	5-8
上位链接系统	11-16
设定 CPU 总线单元和高功能 I/O 单元 时间和中断设定	8-18
使用产品批号实现程序操作保护	9-14
时钟功能	10-51
时钟脉冲	10-3
时钟脉冲	6-2, 6-43
输出位	6-9
输出置 OFF	10-62
数据存储区	6-2, 6-23
数据格式	5-77
数据跟踪	10-63
数据寄存器	6-2, 6-39
数据结构	5-58
数据类型	5-56

数据链接区	6-2, 6-12
数据文件	7-8
输入位	6-7
输入中断的响应时间	12-13
输入中断任务	5-21, 5-29

T

特殊程序段	5-110
替换整个程序	10-39
梯形图	5-6
调试	10-59
条件标志	5-105, 6-2, 6-41
同步数据刷新区	6-13
同步运转周期	10-75
通电时钟数据	10-4
通过 DIP 开关实现写保护	10-46
通过仿真器进行调试	5-99
通过密码实现读保护	10-46
通信端口服务	3-5
通信设定	9-23
通用文件	7-14

W

外部中断任务	5-28
外设服务	3-5, 9-30
备份服务	3-5
CPU 总线单元的事件服务	3-5
串行端口服务	3-5
高性能 I/O 单元的事件服务	3-5
联机编辑服务	3-5
通信端口服务	3-5
外设 USB 端口服务	3-5
文件访问服务	3-5
网络符号	5-51
网络符号（标签）	2-3
为 CPU 总线单元设定 DM 区分配字	4-14
微分监控	10-60
为高性能 I/O 单元设定 DM 区分配字	4-14
文件存储器	6-27, 7-2
参数文件	7-8
程序 / 网络符号文件	7-7
程序索引文件	7-9
单元备份文件	7-9
符号表文件	7-9
数据文件	7-8
注释文件	7-9
文件存储器初始化	7-3
文件访问服务	3-5
无电池操作	9-9
无协议通信系统	11-17

X

协议宏	11-20
旋转开关	4-2
循环任务	5-14
循环时间	3-4
计算循环时间	12-4
监视循环时间	12-2
循环时间延长	12-12
循环刷新	5-81

Y

用户程序区	2-3
用户程序区溢出错误	5-103
用户程序时间	10-5
用户自定义的数据类型	5-58
与任务相关的标志	5-33
源代码	2-3
源和注释区	2-3
运行结束时间	10-5
运行开始时间	10-5
运行模式	3-8, 9-7
MONITOR 模式	3-8
PROGRAM 模式	3-8
RUN 模式	3-8
运行模式变更和启动时的保持设定	10-20
允许 / 禁止存储到存储卡	10-49
允许和禁止程序覆写	10-50

Z

暂存继电器区	6-2, 6-22
针对 FINS 命令的写保护	10-52
指定数组	5-57
指令变化	5-65
指令的结构	
标志	5-63
操作数	5-64
能流	5-62
指令条件	5-63
指令功能	A-3
指令执行时间	A-79
指令执行条件	5-65
执行过程	9-8
执行设定	9-7
中断开销时间	5-22
中断任务	5-20
电源 OFF 中断任务	5-20
定时中断任务	5-20
附加循环任务	5-20
I/O 中断任务	5-20
外部中断任务	5-20
中继网络表	4-11
注释	2-3
注释存储器	10-38
注释文件	7-9
自保持位	6-20
子程序	5-110
自动备份	10-36
自动地址分配	5-61, 6-26
自动联机	11-3, 11-11
自由运行定时器	10-6
总通电时间	10-5
最大循环时间	10-8
最小循环时间	10-7

修订记录

手册封面上样本编号的后缀部分即为修订号。

样本编号：W473-CN5-09



下表对手册每次修订后发生的变更进行了概述。具体页码参见上一版手册。

修订号	日期	修订内容
01	2008年7月	首次出版
02	2008年12月	增加 CJ2H-CPU6□。 增加对 1.1 版 CJ2 CPU 单元的说明。 增加下列基本 I/O 单元：CJ1W-ID212、CJ1W-ID233、CJ1W-OD213 和 CJ1W-OD234。
03	2009年2月	增加对单元同步运转功能的说明。 增加下列位置控制单元：CJ1W-NC214、CJ1W-NC234、CJ1W-NC414 和 CJ1W-NC434。
04	2009年7月	增加 CJ1W-AD042 模拟量输入单元。 增加 CJ1W-DA042V 模拟量输出单元。
05	2009年10月	增加 CJ1W-SCU22/32/42 串行通信单元。 增加 DRXDU(261) 和 DTXDU(262) 指令。 增加对 EM 区强制置位 / 复位功能 (1.2 版中的新功能) 的说明。
06	2010年2月	增加 CJ2M-CPU□□。
07	2010年7月	增加对 CJ2M-MD21□ 脉冲 I/O 模块 (CJ2M CPU 单元用) 的说明。 更正部分错误。
08	2010年10月	增加对 CJ2M-CPU3□ CPU 单元内置 EtherNet/IP 端口 2.1 版的说明。 更正部分错误。
09	2023年4月	增加安全对策的说明。

承诺事项

承蒙对欧姆龙株式会社(以下简称“本公司”)产品的一贯厚爱和支持,藉此机会再次深表谢意。

如果未特别约定,无论贵司从何处购买的产品,都将适用本承诺事项中记载的事项。

请在充分了解这些注意事项基础上订购。

1. 定义

本承诺事项中的术语定义如下。

- (1)“本公司产品”:是指“本公司”的FA系统机器、通用控制器、传感器、电子/结构部件。
- (2)“产品目录等”:是指与“本公司产品”有关的欧姆龙综合产品目录、FA系统设备综合产品目录、安全组件综合产品目录、电子/机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等,包括以电子数据方式提供的资料。
- (3)“使用条件等”:是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。
- (4)“客户用途”:是指客户使用“本公司产品”的方法,包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。
- (5)“适用性等”:是指在“客户用途”中“本公司产品”的(a)适用性、(b)动作、(c)不侵害第三方知识产权、(d)法规法令的遵守以及(e)满足各种规格标准。

2. 关于记载事项的注意事项

对“产品目录等”中的记载内容,请理解如下要点。

- (1)额定值及性能值是在单项试验中分别在各条件下获得的值,并不构成对各额定值及性能值的综合条件下获得值的承诺。
- (2)提供的参考数据仅作参考,并非可在该范围内一直正常运行的保证。
- (3)应用示例仅作参考,不构成对“适用性等”的保证。
- (4)如果因技术改进等原因,“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

- (1)除了额定值、性能指标外,使用时还必须遵守“使用条件等”。
- (2)客户应事先确认“适用性等”,进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。
- (3)对于“本公司产品”在客户的整个系统中的设计用途,客户应负责事先确认是否已进行了适当配电、安装等事项。
- (4)使用“本公司产品”时,客户必须采取如下措施:(i)相对额定值及性能指标,必须在留有余量的前提下使用“本公司产品”,并采用冗余设计等安全设计(ii)所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii)构建随时提示使用者危险的完整安全体系、(iv)针对“本公司产品”及“客户用途”定期实施各项维护保养。
- (5)因DDoS攻击(分布式DoS攻击)、计算机病毒以及其他技术性有害程序、非法侵入,即使导致“本公司产品”、所安装软件、或者所有的计算机器材、计算机程序、网络、数据库受到感染,对于由此而引起的直接或间接损失、损害以及其他费用,“本公司”将不承担任何责任。
对于(i)杀毒保护、(ii)数据输入输出、(iii)丢失数据的恢复、(iv)防止“本公司产品”或者所安装软件感染计算机病毒、(v)防止对“本公司产品”的非法侵入,请客户自行负责采取充分措施。
- (6)“本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。如果客户将“本公司产品”用于以下所列用途,则本公司对产品不作任何保证。但“本公司”已表明可用于特殊用途,或已与客户有特殊约定时,另行处理。
 - (a)必须具备很高安全性的用途(例:核能控制设备、燃烧设备、航空/宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途)
 - (b)必须具备很高可靠性的用途(例:燃气、自来水、电力等供应系统、24小时连续运行系统、结算系统、以及其他处理权利、财产的用途等)
 - (c)具有苛刻条件或严酷环境的用途(例:安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等)
 - (d)“产品目录等”资料中未记载的条件或环境下的用途
- (7)除了不适用于上述3.(6)(a)至(d)中记载的用途外,“本产品目录等资料中记载的产品”也不适用于汽车(含二轮车,下同)。请勿配置到汽车上使用。关于汽车配置用产品,请咨询本公司销售人员。

4. 保修条件

“本公司产品”的保修条件如下。

- (1)保修期限 自购买之日起1年。(但是,“产品目录等”资料中有明确说明时除外。)
- (2)保修内容 对于发生故障的“本公司产品”,由“本公司”判断并可选择以下其中之一方式进行保修。
 - (a)在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理(但是对于电子、结构部件不提供修理服务。)
 - (b)对发生故障的“本公司产品”免费提供同等数量的替代品
- (3)当故障因以下任何一种情形引起时,不属于保修的范围。
 - (a)将“本公司产品”用于原本设计用途以外的用途
 - (b)超过“使用条件等”范围的使用
 - (c)违反本注意事项“3.使用时的注意事项”的使用
 - (d)非因“本公司”进行的改装、修理导致故障时
 - (e)非因“本公司”出品的软件导致故障时
 - (f)“本公司”生产时的科学、技术水平无法预见的原因
 - (g)除上述情形外的其它原因,如“本公司”或“本公司产品”以外的原因(包括天灾等不可抗力)

5. 责任限制

本承诺事项中记载的保修是关于“本公司产品”的全部保证。对于因“本公司产品”而发生的其他损害,“本公司”及“本公司产品”的经销商不负任何责任。

6. 出口管理

客户若将“本公司产品”或技术资料出口或向境外提供时,请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则,“本公司”有权不予提供“本公司产品”或技术资料。

IC320GC-zh

202304

注:规格如有变更,恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化(中国)有限公司

http://www.fa.omron.com.cn 咨询热线:400-820-4535