

FC5A 系列
MICRO**Smart** _____
_____ *pentra* _____

用户手册 高级卷

安全注意事项

- 在安装、接线、操作、维护和检查 MicroSmart 前，请仔细阅读此用户手册以确保操作正确。
- 所有 MicroSmart 模块都是在 IDEC 严格的质量管理系统下制造的，万一 MicroSmart 发生故障则会发生重大事故或损害的用途中使用 MicroSmart 时，请用户务必在控制系统中做好备份或故障保护准备。
- 在本用户手册中，将安全事项归为警告和注意两类，请用户予以特别重视。



警告

警告提示用于强调操作不当会导致严重的人身伤亡。

- 在安装、拆卸、接线、维护以及检查 MicroSmart 前，请务必关闭 MicroSmart 的电源。如果不关闭电源，可能导致触电或火灾危险。
- 需要采用特殊的专门技术来安装、接线、编程和操作 MicroSmart。没有这些专门技术的人员不得使用 MicroSmart。
- 必须在 MicroSmart 的外部设置紧急停止和联锁电路。如果将这样的电路设置在 MicroSmart 的内部，那么，一旦 MicroSmart 发生故障，则可能导致控制系统混乱、损坏或意外事故。
- 请按本用户手册所描述的操作步骤安装 MicroSmart。安装不正确将导致 MicroSmart 发生跌落、故障或误动作。



注意

在疏忽会导致人身伤害或设备损坏的地方会有注意提示。

- MicroSmart 是为安装在机柜中设计的。请勿将 MicroSmart 安装在机柜的外部。
- 请在用户手册所述的环境下安装 MicroSmart。如果在使用 MicroSmart 时，MicroSmart 周围的环境为高温、高湿度、有结露或腐蚀性气体，且摇摆和震动剧烈，则会导致触电、火灾或故障发生。
- 使用 MicroSmart 的环境是“污染等级 2”。请在污染等级为 2（按照 IEC 60664-1）的环境中使用 MicroSmart。
- 要避免在移动和运输 MicroSmart 的过程中将 MicroSmart 跌落，否则会造成 MicroSmart 损坏或出现故障。
- 防止金属碎片和电缆片段落入 MicroSmart 机架内部。安装和接线时，请在 MicroSmart 模块上盖上面罩。若有碎屑进入，可能会导致火灾、损坏或故障。
- 使用额定值的电源。电源使用不当会导致火灾。
- 在 MicroSmart 外部的电源线上使用符合 IEC 60127 的保险丝。这是销往欧洲的装有 MicroSmart 的设备所必需的。
- 请在输出电路上使用经 IEC60127 认可的保险丝。这是销往欧洲的装有 MicroSmart 的设备所必需的。
- 使用经欧盟认可的断路器。这是销往欧洲的装有 MicroSmart 的设备所必需的。
- 在启动和停止 MicroSmart 前，或操作 MicroSmart 强行打开或关闭输出时，请确保安全。MicroSmart 操作不当会导致机器损坏或意外事故。
- 如果 MicroSmart 输出模块中的继电器或晶体管发生故障，输出可能持续打开或关闭的状态。为了避免输出信号造成严重事故，请在 MicroSmart 外设置监控电路。
- 请勿将地线与 MicroSmart 直接连接。请使用 M4 或更大的螺钉为装有 MicroSmart 的机柜提供保护性接地。这是销往欧洲的装有 MicroSmart 的设备所必需的。
- 请勿擅自分解、修理或改装 MicroSmart 模块。
- 当 MicroSmart 模块中的电池电量耗尽时，请按照相关规定处理。请使用专用容器存放或处置电池。这是销往欧洲的装有 MicroSmart 的设备所必需的。
- 请按工业废物处理 MicroSmart。

版本简历

下表概述了自 2006 年 4 月首版 FC9Y-B929-0 以来更改的内容。

在 FC5A MicroSmart CPU 模块中执行升级的功能和新功能如下所示。这些功能的使用取决于模块和 MicroSmart CPU 模块的系统程序版本。

要确认 MicroSmart CPU 模块的系统程序版本，请在与此 CPU 模块连接的计算机上使用 WindLDR。系统程序版本显示在 PLC 状态对话框中。请参阅第 13-1 页（基本卷）。

要确认 WindLDR 版本，请选择 WindLDR 屏幕左上角的 WindLDR 应用程序按钮，然后选择 **WindLDR 选项 > 代码**。WindLDR 版本位于关于 WindLDR 之中。

升级的功能和新功能一览

CPU 模块	集成型		超薄型	WindLDR	页面
	FC5A-C10R2 FC5A-C10R2C FC5A-C10R2D FC5A-C16R2 FC5A-C16R2C FC5A-C16R2D	FC5A-C24R2 FC5A-C24R2C FC5A-C24R2D	FC5A-D16RK1 FC5A-D16RS1 FC5A-D32K3 FC5A-D32S3 FC5A-D12K1E FC5A-D12S1E (注释 1)		
HMI 模块升级 (注释 2)	110 或更高		101 或更高	-	基本卷 5-60
FC5A-SIF2 扩展 RS232C 通信模块的兼容性 (注释 3)	-	110 或更高		5.1 或更高	基本卷 2-86 25-1
Modbus 主机升级 (注释 4)		-	110 或更高		基本卷 12-5
Modbus 从机升级 (注释 4)	110 或更高	110 或更高			基本卷 12-12
32 位数据存储设置					基本卷 5-46
强制 I/O					基本卷 5-72
RUN LED 闪烁模式					基本卷 5-49
内存盒上传功能 (注释 5)					基本卷 2-93
断开延时定时器指令 (TMLO, TIMO, TMHO, TMSO)					基本卷 7-11
双字计数器指令 (CNTD, CDPD, CUDD)					基本卷 7-15
MOV 和 IMOV 指令升级 (新数据类型 F)					3-1
N 数据置位和 N 数据重复置位指令 (NSET, NRS)					3-13 3-14
定时器 / 计数器 当前值存储指令 (TCCST)	200 或更高	200 或更高	200 或更高	5.2 或更高	3-16
CMP 指令升级					4-4
数据比较接点指令 (LC=, LC<>, LC<, LC>, LC<=, LC>=)					4-8
BTOA 和 ATOB 指令升级 (新数据类型 D)					8-9 8-12
数据分割、组合和交换指令 (DTDV, DTCB, SWAP)					8-21 8-22 8-23
用户通信指令升级 (TXD, RXD)					基本卷 10-6 基本卷 10-15
文件处理指令 (FIFO, FIEX, FOEX)					19-1 19-3

CPU 模块	集成型		超薄型	WindLDR	页面
	FC5A-C10R2 FC5A-C10R2C FC5A-C10R2D FC5A-C16R2 FC5A-C16R2C FC5A-C16R2D	FC5A-C24R2 FC5A-C24R2C FC5A-C24R2D	FC5A-D16RK1 FC5A-D16RS1 FC5A-D32K3 FC5A-D32S3 FC5A-D12K1E FC5A-D12S1E (注释 1)		
键矩阵功能 (注释 6)	-				基本卷 5-38
升级用户程序保护					基本卷 5-44
交换指令 (XCHG)					3-15
递增指令 (INC)					5-13
递减指令 (DEC)					5-13
合计指令 (SUM)					5-15
随机指令 (RNDM)					5-18
递减跳转非零 (DJNZ)					11-5
N 数据查找指令 (NDSRC)					19-5
时钟指令 (TADD, Tsub, HTOS, STOH, HOUR)					20-1
集成 12V DC 电源 CPU 模块	-	-	-		基本卷 2-1
模拟量 I/O 模块升级 (版本 200 或更高)	-	一部分	一部分	一部分	基本卷 2-56
Modbus TCP 通信					23-1
端口 1 的 Modbus 从机通信 (注释 4)	210 或更高	210 或更高	210 或更高	5.3 或更高	基本卷 12-9
通电状态下运行 / 停止选项	220 或更高				基本卷 5-4
FC5A-SIF4 扩展 RS485 通信模块的 兼容性 (注释 3)					基本卷 2-86 25-1
端口 3 至端口 7 的数据连接与 Modbus 通信 (注释 4)	-	220 或更高	220 或更高	6.2 或更高	基本卷 11-1 基本卷 12-1
端口 3 至端口 7 的通信刷新选项					基本卷 5-43
PID 功能改善 (比例范围用 支持积 分开始系数)	-	246 或更高	246 或更高 (FC5A-D16Rx1 或 FC5A-D32x3) 131 或更高 (FC5A-D12x1E)	7.2 或更高	14-9

通过使用电脑上的 WindLDR 与 CPU 模块连接，确认 MicroSmart CPU 模块的系统版本。系统程序版本显示在 PLC 状态对话框中。请参阅第 13-1 页 (基本卷)。

注释 1: 所有功能均可在系统程序版本 100 的 FC5A-D12K1E 和 FC5A-D12S1E 上使用。

注释 2: 使用该功能必须配置 HMI 模块 (FC4A-PH1)。

注释 3: 扩展 RS232C 和 RS485 通信模块 (FC5A-SIF2 与 FC5A-SIF4) 不能与 FC5A-C24R2D CPU 模块一起使用。

注释 4: Modbus 主机通信模式在端口 2 至 7 上可以使用。端口 1 至 7 上可以使用 Modbus 从机通信模式。接口 2 的使用需要用到可选通信适配器 (FC4A-PC1 或 FC4A-PC3) 或通信模块 (FC4A-HPC1 或 FC4A-HPC3)。接口 3 至 7 的使用需要用到扩展 RS232C 或 RS485 通信模块 (FC5A-SIF2 或 FC5A-SIF4)。

注释 5: 使用该功能必须配置内存盒 (FC4A-PM32, FC4A-PM64 或 FC4A-PM128)。

注释 6: 键矩阵输入不能用于 FC5A-C24R2D CPU 模块。

版本更新

日期	使用手册编号	说明
2011 年 3 月	B-1274(0)	第一版
2014 年 8 月	B-1274(1)	添加 PID 新规格，可在选择比例带时设置积分开始系数

关于本手册

本用户手册主要介绍 MicroSmart 的设置和强大的通信功能。

第 1 章：基本指令参考

设置基本指令、可用设备和示例程序。

第 2 章：高级指令

使用高级指令、术语、数据类型和高级指令所用格式的一般规则。

第 3 章～第 20 章

这 18 章详细介绍高级指令。

第 21 章～第 25 章

各种通信功能，如计算机连接、调制解调器模式、Modbus TCP、AS-Interface 和扩展 RS232C/RS485 通信。

索引

依据关键字首字母顺序排序。

重要声明

在任何情况下，IDEC 株式会社都不对由于使用或应用 IDEC PLC 组件而间接或直接导致的损坏负责（无论是单独使用，还是与其他设备结合使用）。

所有使用这些组件的人员都要自行承担选择适用于其应用程序的组件以及选择适用于这些组件的应用程序的责任（无论是单独使用，还是与其他设备结合使用）。

本手册中的所有图表和示例仅起说明作用。这些图表和示例并不保证其适用于任何特殊应用软件。在安装前，最终用户需承担测试和认可所有程序的责任。

相关手册

可提供以下有关 FC5A 系列 MicroSmart 的手册。可结合本手册一同参考使用。

代码	手册名称	说明
FC9Y-B1274	FC5A 系列 MicroSmart Pentra 用户手册 高级卷 (本手册)	介绍指令列表、传送指令、数据比较指令、四则运算指令、逻辑运算指令、移位 / 循环指令、数据转换指令、周程序指令、接口指令、程序分支指令、刷新指令、中断控制指令、坐标转换指令、平均指令、脉冲输出指令、PID 指令、双 / 示教定时器指令、智能型模块访问指令、三角函数指令、对数 / 幂指令、文件数据处理指令、时钟指令、计算机连接通信、调制解调器通信、Modbus TCP 通信、扩展 RS232C/RS485 通信模块、以及 AS - Interface 主机模块。
FC9Y-B1269	FC5A 系列 MicroSmart Pentra 用户手册 基本卷	介绍模块规格、安装说明、接线说明、基本操作、特殊功能、设备地址、指令列表、基本指令、模拟模块、用户通信、数据连接通信、Modbus ASCII/RTU 通信以及故障排除。
FC9Y-B1279	FC5A 系列 MicroSmart Pentra 用户手册 Web 服务器 CPU 模块卷	介绍 FC5A 超薄型网络服务器 CPU 模块的规格及功能。
FC9Y-B1284	FC5A 系列 PID 模块 用户手册	介绍 PID 模块的规格及功能。

目录

1:	基本指令参考	
	基本指令表	1-1
2:	高级指令	
	高级指令表	2-1
	高级指令适用的 CPU 模块	2-4
	高级指令的结构	2-7
	高级指令的输入条件	2-7
	源设备和目标设备	2-7
	使用定时器或计数器作为源设备	2-7
	使用定时器或计数器作为目标设备	2-7
	高级指令的数据类型 (整型)	2-8
	设备区域中断	2-10
	NOP (空操作)	2-10
3:	传送指令	
	MOV (传送)	3-1
	MOVN (求反传送)	3-5
	IMOV (间接传送)	3-6
	IMOVN (间接求反传送)	3-8
	BMOV (块传送)	3-9
	IBMV (间接位传送)	3-10
	IBMVN (间接位求反传送)	3-12
	NSET(N 数据置位)	3-13
	NRS (N 数据重复置位)	3-14
	XCHG (交换)	3-15
	TCCST(存储定时器 / 计数器当前值)	3-16
4:	数据比较指令	
	CMP= (比较等于)	4-1
	CMP<> (比较不等于)	4-1
	CMP< (比较小于)	4-1
	CMP> (比较大于)	4-1
	CMP<= (比较小于或等于)	4-1
	CMP>= (比较大于或等于)	4-2
	ICMP>= (间隔比较大于或等于)	4-6
	LC= (接点比较等于)	4-8
	LC<> (接点比较不等于)	4-8
	LC< (接点比较小于)	4-8
	LC> (接点比较大于)	4-8
	LC<= (接点比较小于或等于)	4-8
	LC>= (接点比较大于或等于)	4-8
5:	四则运算指令	
	ADD (加法)	5-1
	SUB (减法)	5-1
	MUL (乘法)	5-1
	DIV (除法)	5-1
	INC (递增)	5-13
	DEC (递减)	5-13

	ROOT (平方根)	5-14
	SUM (合计)	5-15
	RNDM (随机)	5-18
6:	逻辑运算指令	
	ANDW (与)	6-1
	ORW (或)	6-1
	XORW (异或)	6-1
7:	移位/循环指令	
	SFTL (左移)	7-1
	SFTR (右移)	7-3
	BCDLS (BCD 码左移)	7-5
	WSFT (字移位)	7-7
	ROTL (循环左移)	7-8
	ROTR (循环右移)	7-10
8:	数据转换指令	
	HLOB (HEX → BCD 码)	8-1
	BLOH (BCD 码 → HEX)	8-3
	HTOA (HEX → ASCII 码)	8-5
	ATOH (ASCII 码 → HEX)	8-7
	BTOA (BCD 码 → ASCII 码)	8-9
	ATOB (ASCII 码 → BCD 码)	8-12
	ENCO (编码)	8-15
	DECO (解码)	8-16
	BCNT (位计数)	8-17
	ALT (交替输出)	8-18
	CVDT (转换数据类型)	8-19
	DTDV (数据分割)	8-21
	DTCB (数据组合)	8-22
	SWAP (数据交换)	8-23
9:	周程序指令	
	WKTIM (周定时器)	9-1
	WKTBL (周表)	9-2
	使用时钟盒	9-5
	使用 WindLDR 设置日历/时钟	9-6
	使用用户程序设置日历/时钟	9-6
	使用用户程序调整时钟	9-7
	调整时钟盒准确性	9-8
10:	接口指令	
	DISP (七段译码显示)	10-1
	DGRD (数字读取)	10-3
11:	程序分支指令	
	LABEL (标签)	11-1
	LJMP (标签跳转)	11-1
	LCAL (标签调用)	11-3
	LRET (标签返回)	11-3
	DJNZ (递减跳转非零)	11-5

	DI (禁用中断)	11-7
	EI (启用中断)	11-7
	IOREF (I/O刷新)	11-9
	HSCRF (高速计数器刷新)	11-11
	FRQRF (频率测量刷新)	11-12
	COMRF (通信刷新)	11-13
12:	坐标转换指令	
	XYFS (XY格式设置)	12-1
	CVXTY (X → Y转换)	12-2
	CVYTX (Y → X转换)	12-3
	AVRG (平均化)	12-7
13:	脉冲指令	
	PULS1 (脉冲输出 1)	13-2
	PULS2 (脉冲输出 2)	13-2
	PULS3 (脉冲输出 3)	13-2
	PWM1 (脉宽调制 1)	13-8
	PWM2 (脉宽调制 2)	13-8
	PWM3 (脉宽调制 3)	13-8
	RAMP1 (台形控制 1)	13-14
	RAMP2 (台形控制 2)	13-14
	ZRN1 (零返回 1)	13-26
	ZRN2 (零返回 2)	13-26
	ZRN3 (零返回 3)	13-26
14:	PID 指令	
	PID (PID控制)	14-1
	应用程序示例	14-18
15:	双/示教定时器指令	
	DTML (1秒双定时器)	15-1
	DTIM (100毫秒双定时器)	15-1
	DTMH (10毫秒双定时器)	15-1
	DTMS (1毫秒双定时器)	15-1
	TTIM (示教定时器)	15-3
16:	智能型模块访问指令	
	RUNA READ (运行访问读取)	16-2
	RUNA WRITE (运行访问写入)	16-3
	STPA READ (停止访问读取)	16-4
	STPA WRITE (停止访问写入)	16-5
17:	三角函数指令	
	RAD (角度换算为弧度)	17-1
	DEG (弧度换算为角度)	17-2
	SIN (正弦)	17-3
	COS (余弦)	17-4
	TAN (正切)	17-5
	ASIN (反正弦)	17-6
	ACOS (反余弦)	17-7
	ATAN (反正切)	17-8

18:	对数 / 幂指令	
	LOGE (自然对数)	18-1
	LOG10 (常用对数)	18-2
	EXP (指数)	18-3
	POW (幂)	18-4
19:	文件数据处理指令	
	FIFO (先进先出格式)	19-1
	FIEX (执行先进)	19-3
	FOEX (执行先出)	19-3
	NDSRC (N 数据查找)	19-5
20:	时钟指令	
	TADD (时间加法)	20-1
	TSUB (时间减法)	20-5
	HTOS (HMS → 秒)	20-9
	STOH (秒 → HMS)	20-10
	HOUR (小时计量器)	20-11
21:	计算机连接通信	
	计算机连接系统设置 (1:N 计算机连接系统)	21-1
	编程 WindLDR	21-2
	监控 PLC 状态	21-3
22:	调制解调器模式	
	系统安装	22-1
	可用调制解调器	22-2
	调制解调器模式下使用的特殊内部继电器	22-2
	用于调制解调器模式的特殊数据寄存器	22-3
	初始模式	22-3
	断开模式	22-5
	AT 常规命令模式	22-5
	应答模式	22-6
	调制解调器模式状态数据寄存器	22-7
	初始化字符串命令	22-8
	使用调制解调器的准备	22-9
	设置数据寄存器和内部继电器	22-9
	设置 CPU 模块	22-9
	设置 WindLDR	22-10
	调制解调器模式操作步骤	22-11
	调制解调器初始模式示例程序	22-12
	调制解调器应答模式示例程序	22-13
	调制解调器通信故障排除	22-14
23:	MODBUS TCP 通信	
	Modbus TCP 通信通用信息	23-1
	Modbus TCP 主机通信	23-2
	Modbus TCP 从机通信	23-5
	使用 WindLDR 设置 Modbus TCP 从机通信	23-7
	编辑网络服务器单元 (FC4A-SX5ES1E)	23-8
	Modbus TCP 通信格式	23-9

24:	AS-INTERFACE 主机通信	
	AS-Interface	24-1
	基本操作	24-6
	按钮和 LED 指示灯	24-14
	AS-Interface 设备	24-18
	使用两个 AS-Interface 主机模块	24-32
	使用 WindLDR	24-34
	SwitchNet 数据 I/O 端口 (AS-Interface 主机模块 1)	24-39
25:	扩展 RS232C/RS485 通信	
	分配通信端口编号	25-1
	计算机连接通信	25-2
	可编程显示器通信	25-6
	用户通信	25-9
	故障排除	25-13

索引

1: 基本指令参考

简介

本章介绍基本指令列表并简要说明各项指令。

有关各项基本指令的详情，请参见《FC5A 用户手册基本卷》(FC9Y-B1269)的有关页面。

基本指令表

符号	名称	功能	参阅 页面
AND	与	串联常开接点	基本卷 7-5
AND LOD	块与	串联电路块	基本卷 7-6
ANDN	与非	串联常闭接点	基本卷 7-5
BPP	位弹出	还原临时保存的位逻辑操作的结果	基本卷 7-7
BPS	位推入	临时保存位逻辑操作的结果	基本卷 7-7
BRD	位读取	读取临时保存的位逻辑操作的结果	基本卷 7-7
CC=	计数器比较 (=)	计数器当前值的等于比较	基本卷 7-19
CC≥	计数器比较 (≥)	计数器当前值的大于或等于比较	基本卷 7-19
CDP	加 / 减计数器	加 / 减计数器 (0 - 65535)	基本卷 7-12
CDPD	双字加 / 减计数器	双字加 / 减计数器 (0 - 4,294,967,295)	基本卷 7-16
CNT	加计数器	加计数器 (0 - 65535)	基本卷 7-12
CNTD	双字加计数器	双字加计数器 (0 - 4,294,967,295)	基本卷 7-15
CUD	加 / 减切换计数器	加 / 减切换计数器 (0 - 65535)	基本卷 7-14
CUDD	双字加 / 减切换计数器	双字加 / 减切换计数器 (0 - 4,294,967,295)	基本卷 7-17
DC=	数据寄存器比较 (=)	数据寄存器值的等于比较	基本卷 7-21
DC≥	数据寄存器比较 (≥)	数据寄存器值的大于或等于比较	基本卷 7-21
END	结束	结束程序	基本卷 7-31
JEND	跳转结束	结束跳转指令	基本卷 7-30
JMP	跳转	跳转至指定的程序区域	基本卷 7-30
LOD	负载	存储中间结果，并读取接点状态	基本卷 7-3
LODN	取非	存储中间结果，并读取求反的接点状态	基本卷 7-3
MCR	主控继电器结束	结束主控继电器	基本卷 7-28
MCS	主控继电器开始	开始主控继电器	基本卷 7-28
OR	或	并联常开接点	基本卷 7-5
OR LOD	块或	并联电路块	基本卷 7-6
ORN	或非	并联常闭接点	基本卷 7-5
OUT	输出	输出位逻辑操作的结果	基本卷 7-3
OUTN	求反输出	输出位逻辑操作的求反结果	基本卷 7-3
RST	复位	将输出、内部继电器或移位寄存器位复位	基本卷 7-4
SET	置位	将输出、内部继电器、或移位寄存器位置位	基本卷 7-4
SFR	右移位寄存器	右移位寄存器	基本卷 7-23
SFRN	左移位寄存器	左移位寄存器	基本卷 7-23
SOTD	下降沿微分	下降沿微分输出	基本卷 7-27
SOTU	上升沿微分	上升沿微分输出	基本卷 7-27
TIM	100 毫秒定时器	减 100 毫秒定时器 (0 - 6553.5s)	基本卷 7-8
TIMO	100 毫秒断开延时定时器	减 100 毫秒断开延时定时器 (0 - 6553.5s)	基本卷 7-11
TMH	10 毫秒定时器	减 10 毫秒定时器 (0 - 655.35s)	基本卷 7-8
TMHO	10 毫秒断开延时定时器	减 10 毫秒断开延时定时器 (0 - 655.35s)	基本卷 7-11
TML	1 秒钟定时器	减 1 秒钟定时器 (0 - 65535s)	基本卷 7-8
TMLO	1 秒钟断开延时定时器	减 1 秒钟断开延时定时器 (0 - 65535s)	基本卷 7-11

1: 基本指令参考

符号	名称	功能	参阅 页面
TMS	1 毫秒定时器	减 1 毫秒定时器 (0 - 65.535s)	基本卷 7-8
TMSO	1 毫秒断开延时定时器	减 1 毫秒断开延时定时器 (0 - 65.535s)	基本卷 7-11

2: 高级指令

简介

本章描述高级指令的通用使用规则、术语、数据类型和用于高级指令的格式。

高级指令表

组	符号	名称	有效数据类型					参阅页面
			W	I	D	L	F	
NOP	NOP	空操作						2-10
传送	MOV	传送	X	X	X	X		3-1
	MOVN	求反传送	X	X	X	X		3-5
	IMOV	间接传送	X		X			3-6
	IMOVN	间接求反传送	X		X			3-8
	BMOV	块传送	X					3-9
	IBMV	间接位传送	X					3-10
	IBMVN	间接位求反传送	X					3-12
	NSET	N 数据置位	X	X	X	X	X	3-13
	NRS	N 数据重复置位	X	X	X	X	X	3-14
	XCHG	交换	X		X			3-15
TCCST	存储定时器 / 计数器当前值	X		X			3-16	
数据比较	CMP=	比较等于	X	X	X	X	X	4-1
	CMP<>	比较不等于	X	X	X	X	X	4-1
	CMP<	比较小于	X	X	X	X	X	4-1
	CMP>	比较大于	X	X	X	X	X	4-1
	CMP<=	比较小于或等于	X	X	X	X	X	4-1
	CMP>=	比较大于或等于	X	X	X	X	X	4-2
	ICMP>=	间隔比较大于或等于	X	X	X	X	X	4-6
	LC=	接点比较等于	X	X	X	X	X	4-8
	LC<>	接点比较不等于	X	X	X	X	X	4-8
	LC<	接点比较小于	X	X	X	X	X	4-8
	LC>	接点比较大于	X	X	X	X	X	4-8
	LC<=	接点比较小于或等于	X	X	X	X	X	4-8
LC>=	接点比较大于或等于	X	X	X	X	X	4-8	
二进制运算	ADD	加法	X	X	X	X	X	5-1
	SUB	减法	X	X	X	X	X	5-1
	MUL	乘法	X	X	X	X	X	5-1
	DIV	除法	X	X	X	X	X	5-1
	INC	递增	X	X	X	X		5-13
	DEC	递减	X	X	X	X		5-13
	ROOT	平方根	X		X		X	5-14
	SUM	合计 (加法)	X	X	X	X	X	5-15
		合计 (异或)	X					
RNDM	随机	X					5-18	
逻辑运算	ANDW	与	X		X			6-1
	ORW	或	X		X			6-1
	XORW	异或	X		X			6-1

2: 高级指令

组	符号	名称	有效数据类型					参阅页面
			W	I	D	L	F	
移位和循环	SFTL	左移						7-1
	SFTR	右移						7-3
	BCDLS	BCD 码左移			X			7-5
	WSFT	字移位	X					7-7
	ROTL	循环左移	X		X			7-8
	ROTR	循环右移	X		X			7-10
数据转换	HTOB	HEX → BCD 码	X		X			8-1
	BTOH	BCD 码 → HEX	X		X			8-3
	HTOA	HEX → ASCII 码	X					8-5
	ATOH	ASCII 码 → HEX	X					8-7
	BTOA	BCD 码 → ASCII 码	X					8-9
	ATOB	ASCII 码 → BCD 码	X					8-12
	ENCO	编码						8-14
	DECO	解码						8-16
	BCNT	位计数						8-17
	ALT	交替输出						8-18
	CVDT	转换数据类型	X	X	X	X	X	8-19
	DTDV	数据分割	X					8-21
	DTCB	数据组合	X					8-22
	SWAP	数据交换	X		X			8-23
周程序	WKTIM	周定时器						9-1
	WKTBL	周表						9-2
接口	DISP	七段译码显示						10-1
	DGRD	数字读取						10-3
用户通信	TXD1	发送 1						基本卷 10-6
	TXD2	发送 2						基本卷 10-6
	TXD3	发送 3						基本卷 10-6
	TXD4	发送 4						基本卷 10-6
	TXD5	发送 5						基本卷 10-6
	TXD6	发送 6						基本卷 10-6
	TXD7	发送 7						基本卷 10-6
	RXD1	接收 1						基本卷 10-15
	RXD2	接收 2						基本卷 10-15
	RXD3	接收 3						基本卷 10-15
	RXD4	接收 4						基本卷 10-15
	RXD5	接收 5						基本卷 10-15
	RXD6	接收 6						基本卷 10-15
	RXD7	接收 7						基本卷 10-15
程序分支	LABEL	标签						11-1
	LJMP	标签跳转						11-1
	LCAL	标签调用						11-3
	LRET	标签返回						11-3
	DJNZ	递减跳转非零						11-5
	DI	禁用中断						11-7
	EI	启用中断						11-7
	IOREF	I/O 刷新						11-9
	HSCRFB	高速计数器刷新						11-11
	FRQRF	频率测量刷新						11-12
	COMRF	通信刷新						11-13

组	符号	名称	有效数据类型					参阅页面
			W	I	D	L	F	
坐标转换	XYFS	XY 格式设置	X	X				12-1
	CVXTY	X → Y 转换	X	X				12-2
	CVYTX	Y → X 转换	X	X				12-3
	AVRG	平均化	X	X	X	X	X	12-7
脉冲	PULS1	脉冲输出 1						13-2
	PULS2	脉冲输出 2						13-2
	PULS3	脉冲输出 3						13-2
	PWM1	脉宽调制 1						13-8
	PWM2	脉宽调制 2						13-8
	PWM3	脉宽调制 3						13-8
	RAMP1	台形脉冲输出 1						13-14
	RAMP2	台形脉冲输出 2						13-14
	ZRN1	零返回 1						13-26
	ZRN2	零返回 2						13-26
ZRN3	零返回 3						13-26	
PID 指令	PID	PID 控制	X	X				14-1
双 / 示教定时器	DTML	1 秒双定时器						15-1
	DTIM	100 毫秒双定时器						15-1
	DTMH	10 毫秒双定时器						15-1
	DTMS	1 毫秒双定时器						15-1
	TTIM	示教定时器						15-3
智能型模块访问	RUNA	运行访问	X	X				16-2
	STPA	停止访问	X	X				16-4
三角函数	RAD	角度 → 弧度					X	17-1
	DEG	弧度 → 角度					X	17-2
	SIN	正弦					X	17-3
	COS	余弦					X	17-4
	TAN	正切					X	17-5
	ASIN	反正弦					X	17-6
	ACOS	反余弦					X	17-7
	ATAN	反正切					X	17-8
对数 / 幂	LOGE	常用对数					X	18-1
	LOG10	自然对数					X	18-2
	EXP	指数					X	18-3
	POW	幂					X	18-4
文件数据处理	FIFO	先进先出格式	X					19-1
	FIEX	执行先进	X					19-3
	FOEX	执行先出	X					19-3
	NDSRC	N 数据查找	X	X	X	X	X	19-5
时钟	TADD	时间加法						20-1
	TSUB	时间减法						20-5
	HTOS	HMS → 秒						20-9
	STOH	秒 → HMS						20-10
	HOUR	小时计量器						20-11
以太网指令	EMAIL	发送电子邮件						
	PING	Ping						
	ETXD	通过以太网发送						
	ERXD	通过以太网接收						

2: 高级指令

高级指令适用的 CPU 模块

适用的高级指令取决于下表列出的 CPU 模块的类型。

组	符号	集成型 CPU 模块			超薄型 CPU 模块	
		FC5A-C10R2 FC5A-C10R2C FC5A-C10R2D	FC5A-C16R2 FC5A-C16R2C FC5A-C16R2D	FC5A-C24R2 FC5A-C24R2C FC5A-C24R2D	FC5A-D16RK1 FC5A-D16RS1	FC5A-D32K3 FC5A-D32S3 FC5A-D12K1E FC5A-D12S1E
NOP	NOP	X	X	X	X	X
传送	MOV	X	X	X	X	X
	MOVN	X	X	X	X	X
	IMOV	X	X	X	X	X
	IMOVN	X	X	X	X	X
	BMOV	X	X	X	X	X
	IBMV	X	X	X	X	X
	IBMVN	X	X	X	X	X
	NSET	X	X	X	X	X
	NRS	X	X	X	X	X
	XCHG	X	X	X	X	X
TCCST	X	X	X	X	X	
数据比较	CMP=	X	X	X	X	X
	CMP<>	X	X	X	X	X
	CMP<	X	X	X	X	X
	CMP>	X	X	X	X	X
	CMP<=	X	X	X	X	X
	CMP>=	X	X	X	X	X
	ICMP>=	X	X	X	X	X
	LC=	X	X	X	X	X
	LC<>	X	X	X	X	X
	LC<	X	X	X	X	X
	LC>	X	X	X	X	X
LC<=	X	X	X	X	X	
LC>=	X	X	X	X	X	
二进制运算	ADD	X	X	X	X	X
	SUB	X	X	X	X	X
	MUL	X	X	X	X	X
	DIV	X	X	X	X	X
	INC	X	X	X	X	X
	DEC	X	X	X	X	X
	ROOT	X	X	X	X	X
	SUM	X	X	X	X	X
RNDM	X	X	X	X	X	
逻辑运算	ANDW	X	X	X	X	X
	ORW	X	X	X	X	X
	XORW	X	X	X	X	X
移位和循环	SFTL	X	X	X	X	X
	SFTR	X	X	X	X	X
	BCDLS	X	X	X	X	X
	WSFT	X	X	X	X	X
	ROTL	X	X	X	X	X
	ROTR	X	X	X	X	X

组	符号	集成型 CPU 模块			超薄型 CPU 模块	
		FC5A-C10R2 FC5A-C10R2C FC5A-C10R2D	FC5A-C16R2 FC5A-C16R2C FC5A-C16R2D	FC5A-C24R2 FC5A-C24R2C FC5A-C24R2D	FC5A-D16RK1 FC5A-D16RS1	FC5A-D32K3 FC5A-D32S3 FC5A-D12K1E FC5A-D12S1E
数据转换	HTOB	X	X	X	X	X
	BTOH	X	X	X	X	X
	HTOA	X	X	X	X	X
	ATOH	X	X	X	X	X
	BTOA	X	X	X	X	X
	ATOB	X	X	X	X	X
	ENCO	X	X	X	X	X
	DECO	X	X	X	X	X
	BCNT	X	X	X	X	X
	ALT	X	X	X	X	X
	CVDT	X	X	X	X	X
	DTDV	X	X	X	X	X
	DTCB	X	X	X	X	X
	SWAP	X	X	X	X	X
周程序	WKTIM	X	X	X	X	X
	WKTBL	X	X	X	X	X
接口	DISP			X	X	X
	DGRD			X	X	X
用户通信	TXD1	X	X	X	X	X(注释 1)
	TXD2	X	X	X	X	X
	TXD3			X(注释 2)	X	X
	TXD4			X(注释 2)	X	X
	TXD5			X(注释 2)	X	X
	TXD6				X	X
	TXD7				X	X
	RXD1	X	X	X	X	X(注释 1)
	RXD2	X	X	X	X	X
	RXD3			X(注释 2)	X	X
	RXD4			X(注释 2)	X	X
	RXD5			X(注释 2)	X	X
	RXD6				X	X
	RXD7				X	X
程序分支	LABEL	X	X	X	X	X
	LJMP	X	X	X	X	X
	LCAL	X	X	X	X	X
	LRET	X	X	X	X	X
	DJNZ	X	X	X	X	X
	DI	X	X	X	X	X
	EI	X	X	X	X	X
	IOREF	X	X	X	X	X
	HSCRFB	X	X	X	X	X
	FRQRF	X	X	X	X	X
COMRF			X(注释 2)	X	X	
坐标转换	XYFS	X	X	X	X	X
	CVXTY	X	X	X	X	X
	CVYTX	X	X	X	X	X
	AVRG	X	X	X	X	X

注释 1: 不适用于 FC5A-D12K1E/S1E。

注释 2: 不适用于 FC5A-C24R2D。

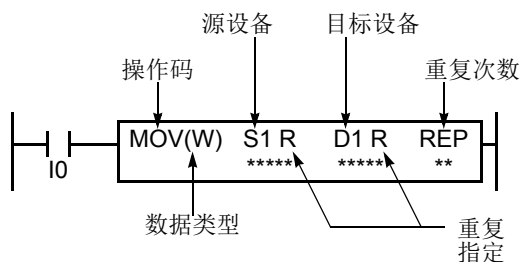
2: 高级指令

组	符号	集成型 CPU 模块			超薄型 CPU 模块	
		FC5A-C10R2 FC5A-C10R2C FC5A-C10R2D	FC5A-C16R2 FC5A-C16R2C FC5A-C16R2D	FC5A-C24R2 FC5A-C24R2C FC5A-C24R2D	FC5A-D16RK1 FC5A-D16RS1	FC5A-D32K3 FC5A-D32S3 FC5A-D12K1E FC5A-D12S1E
脉冲	PULS1				X	X
	PULS2				X	X
	PULS3					X
	PWM1				X	X
	PWM2				X	X
	PWM3					X
	RAMP1				X	X
	RAMP2					X
	ZRN1				X	X
	ZRN2				X	X
	ZRN3					X
PID 指令	PID			X	X	X
双 / 示教定时器	DTML	X	X	X	X	X
	DTIM	X	X	X	X	X
	DTMH	X	X	X	X	X
	DTMS	X	X	X	X	X
	TTIM	X	X	X	X	X
智能型模块访问	RUNA			X(注释 1)	X	X
	STPA			X(注释 1)	X	X
三角函数	RAD	X	X	X	X	X
	DEG	X	X	X	X	X
	SIN	X	X	X	X	X
	COS	X	X	X	X	X
	TAN	X	X	X	X	X
	ASIN	X	X	X	X	X
	ACOS	X	X	X	X	X
	ATAN	X	X	X	X	X
对数 / 幂	LOGE	X	X	X	X	X
	LOG10	X	X	X	X	X
	EXP	X	X	X	X	X
	POW	X	X	X	X	X
文件数据处理	FIFO	X	X	X	X	X
	FIEX	X	X	X	X	X
	FOEX	X	X	X	X	X
	NDSRC	X	X	X	X	X
时钟	TADD	X	X	X	X	X
	TSUB	X	X	X	X	X
	HTOS	X	X	X	X	X
	STOH	X	X	X	X	X
	HOUR	X	X	X	X	X
以太网指令	EMAIL					X(注释 2)
	PING					X(注释 2)
	ETXD					X(注释 2)
	ERXD					X(注释 2)

注释 1: 不适用于 FC5A-C24R2D。

注释 2: 以太网指令只能用于 FC5A-D12K1E 和 FC5A-D12S1E。

高级指令的结构



重复指定

指定是否对设备使用重复。

重复次数

指定重复周期的数量 :1 - 99。

操作码

操作码是用于标识高级指令的符号。

数据类型

指定字型 (W)、整型 (I)、双字型 (D)、长整型 (L) 或浮点型 (F) 数据类型。

源设备

源设备指定要被高级指令处理的 16 位或 32 位数据。某些高级指令需要两个源设备。

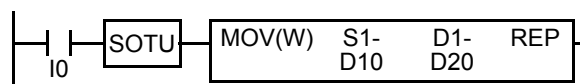
目标设备

目标设备指定用来存储高级指令结果的 16 位或 32 位数据。某些高级指令需要两个目标设备。

高级指令的输入条件

几乎所有高级指令都必须以接点开始，但 NOP（空操作）、LABEL（标签）、LRET（标签返回）和 STPA（停止访问）指令除外。可以使用位设备（例如，输入、输出、内部继电器或移位寄存器）来设置输入条件。此外，也可以将定时器和计数器用作当定时器超时或计数器计数到设定值时打开接点的输入条件。

输入条件打开时，将在每个扫描中执行高级指令。要只在输入的上升沿或下降沿执行高级指令，请使用 SOTU 或 SOTD 指令。



输入条件关闭时，将不执行高级指令，并且保持设备状态。

源设备和目标设备

源设备和目标设备根据选择的数据类型指定 16 位或 32 位数据。将位设备（例如，输入、输出、内部继电器或移位寄存器）指定为源设备或目标设备时，可将以指定数字开始的 16 点或 32 点作为源数据或目标数据来处理。将字设备（例如，定时器或计数器）指定为源设备时，可将当前值作为源数据来读取。将定时器或计数器指定为目标设备时，可将高级指令的结果设置为定时器或计数器的预置值。将数据寄存器指定为源设备或目标设备时，可从指定的数据寄存器读取数据，或将数据写入其中。

使用定时器或计数器作为源设备

由于所有定时器指令 — TML（1 秒定时器）、TIM（100 毫秒定时器）、TMH（10 毫秒定时器）和 TMS（1 毫秒定时器）— 均从预置值减去，因此，将从预置值减去当前值，并指示剩余时间。如上所述，将定时器指定为高级指令的源设备时，可将定时器的当前值（即剩余时间）作为源数据来读取。加法计数器 CNT 从 0 开始计数，并且当前值最多增加到预置值。可逆计数器 CDP 和 CUD 从预置值开始计数，并且当前值从预置值增加或减少。如果将任何计数器指定为高级指令的源设备，则将当前值作为源数据来读取。

使用定时器或计数器作为目标设备

如上所述，将定时器或计数器指定为高级指令的目标设备时，可将高级指令的结果设置为定时器或计数器的预置值。定时器和计数器的预置值可以是 0 - 65535。

使用数据寄存器指定定时器或计数器预置值时，不能将定时器或计数器指定为高级指令的目标设备。执行此类高级指令时，将导致用户程序执行错误。有关用户程序执行错误的详细信息，请参阅第 13-2 页（基本卷）。

注释：发生用户程序执行错误时，不会将结果设置为目标。

2: 高级指令

高级指令的数据类型（整型）

使用传送、数据比较、二进制运算、逻辑运算、移位/循环、数据转换和坐标转换指令时，可以从字型(W)、整型(I)、双字型(D)、长整型(L)或浮点型(F)中选择数据类型。对于其他高级指令，将以16位字为单位来处理数据。

数据类型	符号	位数	使用的数据寄存器数量	十进制的范围
字型（无符号 16 位）	W	16 位	1	0 ~ 65,535
整型（带符号 15 位）	I	16 位	1	-32,768 ~ 32,767
双字型（无符号 32 位）	D	32 位	2	0 ~ 4,294,967,295
长整型（带符号 31 位）	L	32 位	2	-2,147,483,648 ~ 2,147,483,647
浮点型（浮点）	F	32 位	2	$-3.402823 \times 10^{38} \sim 3.402823 \times 10^{38}$

十进制和十六进制存储（字型、整型、双字型和长整型）

下表显示了存储在 CPU 中的十六进制等价值，并显示了加、减十进制的结果：

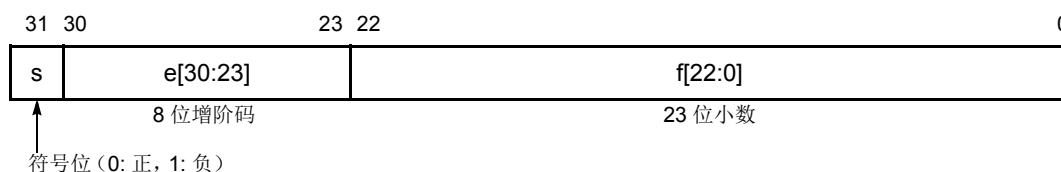
数据类型	加的结果	十六进制存储	减的结果	十六进制存储
字型	0	0000	65535	FFFF
	65535	FFFF	0	0000
	131071	(CY)FFFF	-1	(BW)FFFF
			-65535	(BW)0001
整型			-65536	(BW)0000
	65534	(CY)7FFE	65534	(BW) 7FFE
	32768	(CY)0000	32768	(BW)0000
	32767	7FFF	32767	7FFF
	0	0000	0	0000
	-1	FFFF	-1	FFFF
	-32767	8001	-32767	8001
	-32768	8000	-32768	8000
双字型	-32769	(CY)FFFF	-32769	(BW)FFFF
	-65535	(CY)8001	-65535	(BW) 8001
	0	00000000	4294967295	FFFFFFFF
	4294967295	FFFFFFFF	0	00000000
长整型	8589934591	(CY)FFFFFFFF	-1	(BW) FFFFFFFF
			-4294967295	(BW) 00000001
			-4294967296	(BW) 00000000
	4294967294	(CY) 7FFFFFFE	4294967294	(BW) 7FFFFFFE
	2147483648	(CY) 00000000	2147483648	(BW) 00000000
	2147483647	7FFFFFFF	2147483647	7FFFFFFF
	0	00000000	0	00000000
	-1	FFFFFFFF	-1	FFFFFFFF
	-2147483647	80000001	-2147483647	80000001
	-2147483648	80000000	-2147483648	80000000
-2147483649	(CY)FFFFFFFF	-2147483649	(BW) FFFFFFFF	
-4294967295	(CY)80000001	-4294967295	(BW) 80000001	

浮点型数据格式

FC5A MicroSmart 可为高级指令指定浮点型 (F) 数据类型。与双字型 (D) 和长整型 (L) 数据类型一样，浮点型数据类型也使用两个连续的数据寄存器来执行高级指令。FC5A MicroSmart 支持基于 IEEE（电气电子工程师学会）标准 754 单存储格式的浮点型数据。

单存储格式

IEEE 单存储格式由三个字段构成：一个 23 位小数 (f)；一个 8 位增阶码 (e) 和一个 1 位符号 (s)。这些字段相连存储在一个 32 位字中，如下图所示。位 0:22 包含 23 位小数 (f)，位 0 为小数的最低有效位，位 22 为最高有效位；位 23:30 包含 8 位增阶码 (e)，位 32 为增阶码的最低有效位，位 30 为最高有效位；最高阶位 31 包含符号位 (s)。



单存储格式

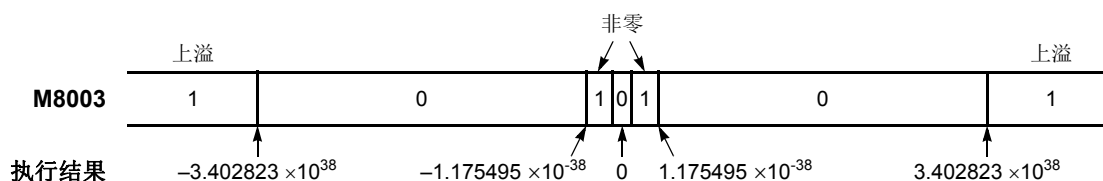
下表说明了这三个构成字段 s、e 和 f 的值与以单格式位模式表示的值之间的对应关系。在将任何超出该位模式的值输入高级指令，或在执行高级指令（例如用 0 除）的过程中生成任何超出该位模式的值时，会导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

单格式位模式	值
$0 < e < 255$	$(-1)^s \times 2^{e-127} \times 1.f$ (范数)
$e = 0; f = 0$ (f 中所有位均为 0)	$(-1)^s \times 2^{e-127} \times 0.0$ (带符号 0)

浮点型数据处理中的进位和借位

当执行含浮点型数据的高级指令时，将更新特殊内部继电器 M8003（进位和借位）。

M8003	执行结果	值
1	$\neq 0$	上溢（超出 $-3.402823 \times 10^{38} \sim 3.402823 \times 10^{38}$ 的范围）
1	0	非零（在 $-1.175495 \times 10^{-38} \sim 1.175495 \times 10^{-38}$ 的范围内）
0	0	零



2: 高级指令

数据寄存器中的双字型设备

为源设备或目标设备选择双字型数据类型时，将从两个连续数据寄存器中读取数据，或将数据存储到两个连续数据寄存器。这两个设备的顺序取决于设备的类型。

选择数据寄存器、定时器或计数器作为双字型设备时，将从选择的第一个设备读取高位字数据，或将高位字数据存储到选择的第一个设备。然后，将从后续设备读取低位字数据，或将低位字数据存储到后续设备。

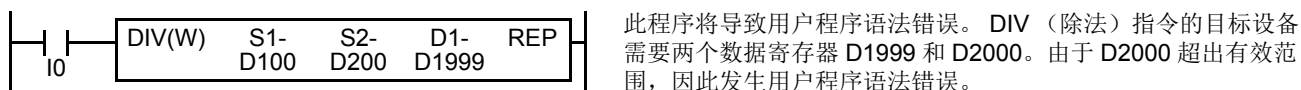
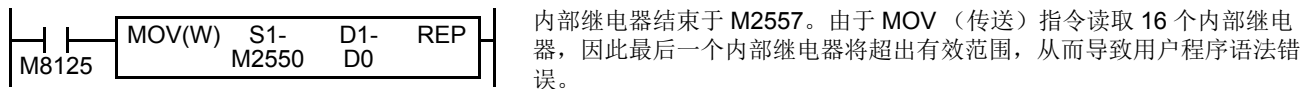
示例：在将数据寄存器 D10 指定为双字型源设备，并将数据寄存器 D20 指定为双字型目标设备时，将从两个连续的数据寄存器读取数据，或将数据存储到两个连续的数据寄存器，如下所示。



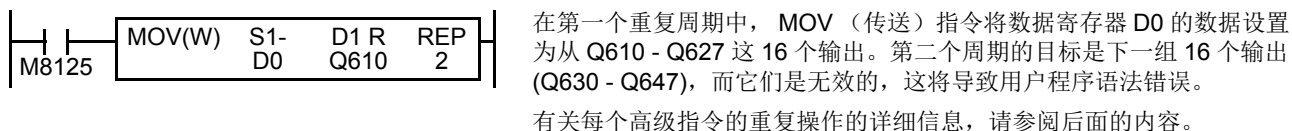
备注：上例显示的是 FC5A MicroSmart 的默认设置。在系统程序版本为 110 或更高的 CPU 模块上，这两个设备的顺序是可选的。

设备区域中断

每个设备区域都是分散和不连续的，例如，从输入到输出，或从输出到内部继电器。此外，特殊内部继电器 M8000 - M8157（集成型 CPU）或 M8317（超薄型 CPU）位于一个从内部继电器 M0 - M2557 的单独区域中。数据寄存器 D0 - D1999、扩展数据寄存器 D2000 - D7999（仅超薄型 CPU）、特殊数据寄存器 D8000 - D8199（集成型 CPU）或 D8499（超薄型 CPU）都位于单独区域中，并且相互不连续。



高级指令只对有效区域内的可用设备执行操作。如果在程序执行期间发现用户程序语法错误，WindLDR 将拒绝该程序指令，并显示错误信息。



NOP（空操作）



后面的章节将介绍所有其他高级指令的详细信息。

3: 传送指令

简介

传送数据时可以使用 MOV（传送）、MOVN（求反传送）、IMOV（间接传送）或 IMOVN（间接求反传送）指令。所传送的数据是 16 位数据，并且还可以使用重复操作来增加所传送的数据数量。在 MOV 或 MOVN 指令中，源设备和目标设备是由 S1 和 D1 直接指定的。在 IMOV 或 IMOVN 指令中，源设备和目标设备由添加到源设备 S1 和目标设备 D1 的 S2 和 D2 所指定的偏移量值来确定。

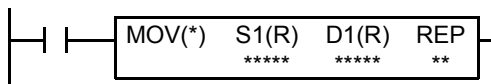
BMOV（块传送）指令用于传送连续的定时器、计数器和数据寄存器值块。

IBMV（间接位传送）和 IBMVN（间接位求反传送）指令用于将一个位数据从源设备传送到目标设备。两个设备都是通过向设备添加偏移量来确定的。使用重复操作时，可以传送连续的位所组成的数据。

NSET(N 数据置位) 和 NRS(N 数据重复置位) 指令可以用于置位设备组的值。XCHG(交换) 指令用于在两个设备间交换字与双字的数据。定时器或计数器的当前值可以使用 TCCST(存储定时器/计数器当前值) 指令进行更改。

由于传送指令是当输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

MOV（传送）



S1→D1

当输入打开时，来自由 S1 所指定的设备的 16 位或 32 位数据将传送到由 D1 所指定的设备。

浮点数据类型适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1（源 1）	要传送的起始设备编号	X	X	X	X	X	X	X	X	1-99
D1（目标 1）	要传送到的起始设备编号	—	X	▲	X	X	X	X	—	1-99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

源设备可以是内部继电器 M0 - M2557，也可以是特殊内部继电器 M8000 - M8157（集成型 CPU）或 M8317（超薄型 CPU）。

在 T（定时器）或 C（计数器）用作 S1 时，定时器/计数器当前值（TC 或 CC）将被读出。在 T（定时器）或 C（计数器）用作 D1 时，数据将作为预置值（TP 或 CP）写入，此值介于 0 到 65535 之间。

当被选择为 F（浮点型）数据类型时，只有数据寄存器和常量才能指定为 S1，数据寄存器才能指定为 D1。

当选择 F（浮点型）数据类型，并且 S1 不符合标准浮点格式时，将导致程序执行错误，并打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

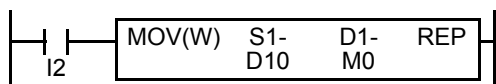
有效数据类型

W（字型）	X	如果将 I（输入）、Q（输出）、M（内部继电器）或 R（移位寄存器）等位设备指定为源或目标，则使用 16 点（字型或整型数据类型）或 32 点（双字型或长整型数据类型）。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I（整型）	X	
D（双字型）	X	如果将 T（定时器）、C（计数器）或 D（数据寄存器）等字设备指定为源或目标，则使用 1 点（字型或整型数据类型）或 2 点（双字型或长整型数据类型）。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L（长整型）	X	
F（浮点型）	X	

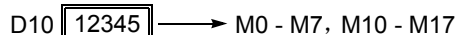
3: 传送指令

示例：MOV

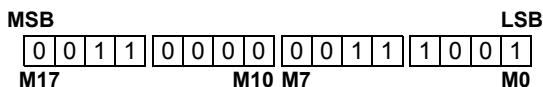
数据类型：字型



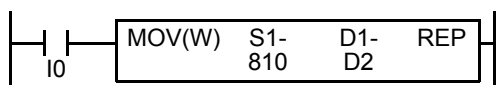
D10→M0
输入 I2 打开时，由源设备 S1 指定的数据寄存器 D10 中的数据将传送到由目标设备 D1 指定的、以 M0 开始的 16 个内部继电器。



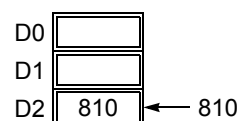
源数据寄存器中的数据将转换为 16 位二进制数据，并且 16 位的开/关状态将传送到内部继电器 M0 - M7 和 M10 - M17。M0 是 LSB（最低有效位）。M17 是 MSB（最高有效位）。



数据类型：字型

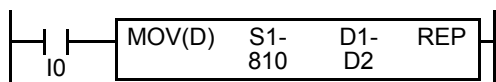


810 →D2
输入 I0 打开时，由源设备 S1 指定的常量 810 将传送到由目标设备 D1 指定的数据寄存器 D2。

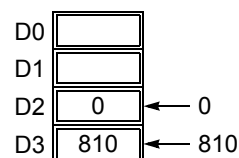


整型数据类型与字型数据类型的数据传送操作相同。

数据类型：双字型

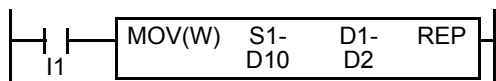


810 →D2·D3
输入 I0 打开时，由源设备 S1 指定的常量 810 将传送到由目标设备 D1 指定的数据寄存器 D2 和 D3。

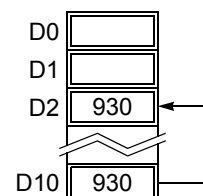


长整型数据类型与双字型数据类型的数据传送操作相同。

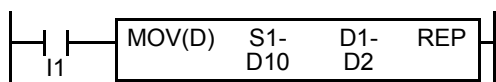
数据类型：字型



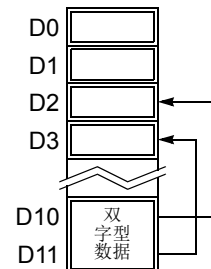
D10→D2
输入 I1 打开时，由源设备 S1 指定的数据寄存器 D10 中的数据将传送到由目标设备 D1 指定的数据寄存器 D2。



数据类型：双字型



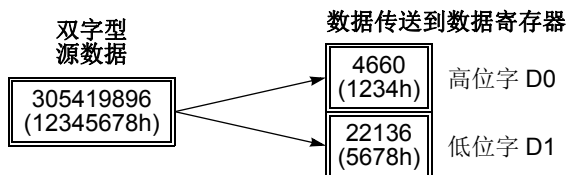
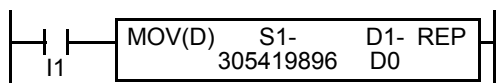
D10·D11→D2·D3
输入 I1 打开时，由源设备 S1 指定的数据寄存器 D10 和 D11 中的数据将传送到由目标设备 D1 指定的数据寄存器 D2 和 D3。



数据寄存器中的双字型数据传送

选择数据寄存器、定时器或计数器作为双字型设备时，将从选择的第一个设备读取高位字数据，或将高位字数据存储在选择的第一个设备。然后，将从后续设备读取低位字数据，或将低位字数据存储在后续设备。

双字型目标设备：数据寄存器

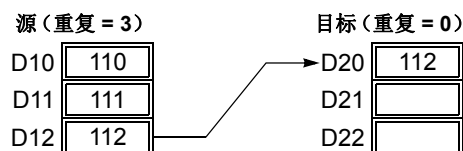
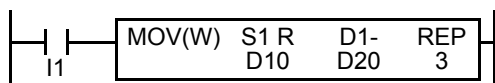


传送指令中的重复操作

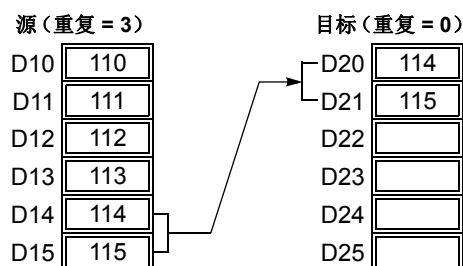
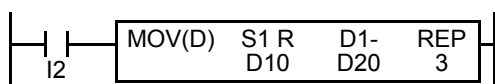
重复源设备

将 S1（源）指定为重复时，以 S1 所指定的设备开始并与重复周期一样多的设备将传送到目标。结果，只有最后一个源设备传送到目标。

·数据类型：字型



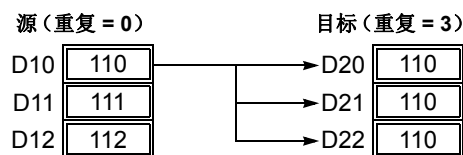
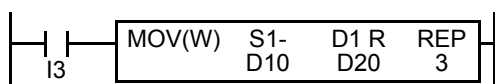
·数据类型：双字型



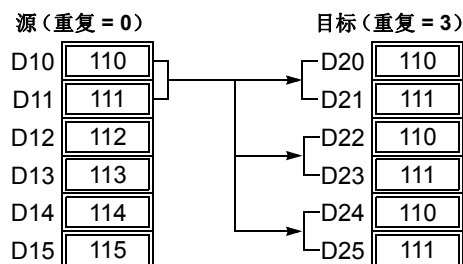
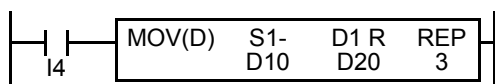
重复目标设备

将 D1（目标）指定为重复时，S1 指定的源设备将传送到以 D1 所指定的目标开始并与重复周期一样多的所有目标设备。

·数据类型：字型



·数据类型：双字型

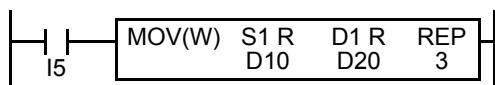


重复源设备和目标设备

将 S1（源）和 D1（目标）指定为重复时，以 S1 所指定的设备开始并与重复周期一样多的设备将传送到以 D1 所指定的设备开始的相同数量的设备。

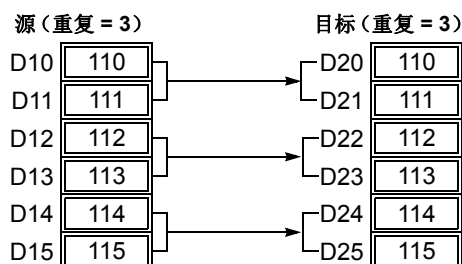
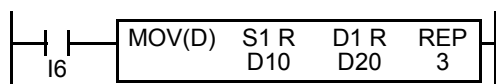
注释：对于指定为重复的源和目标，BMOV（块传送）指令与 MOV 指令的作用相同。

·数据类型：字型



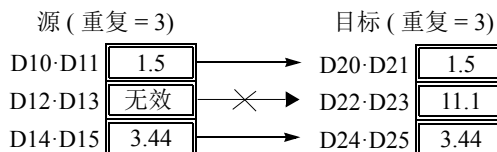
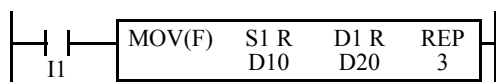
3: 传送指令

·数据类型：双字型



·数据类型：浮点型

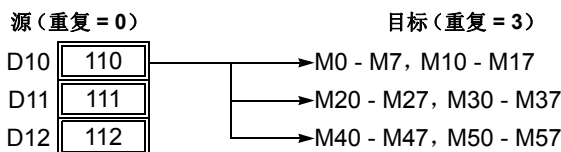
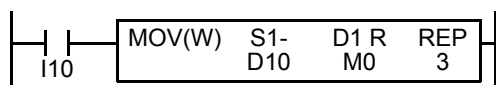
当源数据不符合重复操作中的标准浮点格式时，将发生程序执行错误，并且源数据不能传送到目标数据中。



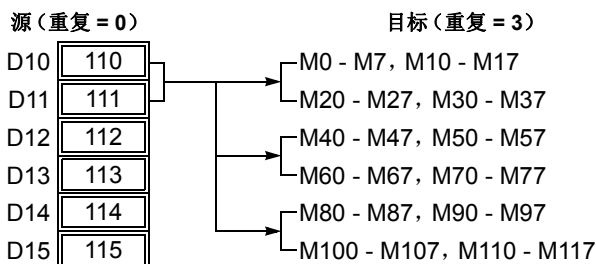
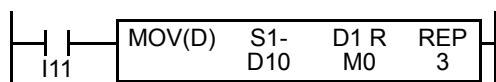
重复位设备

MOV（传送）指令传送 16 位数据（字型或整型数据类型）或 32 位数据（双字型或整型数据类型）。如果将输入、输出、内部继电器或移位寄存器等位设备指定为源设备或目标设备，则以 S1 或 D1 指定的位开始的 16 位或 32 位将成为目标数据。如果对位设备指定重复操作，则根据选择的数据类型，目标数据将以 16 位或 32 位为增量增加。

·数据类型：字型

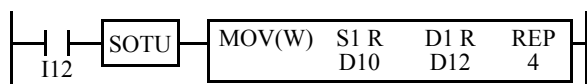


·数据类型：双字型

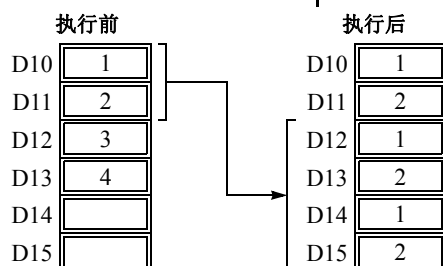


重复所导致的重叠设备

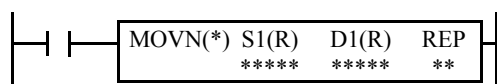
如果对源和目标都指定了重复操作，并且一部分源区域和目标区域相互重叠，则还会更改重叠区域中的源数据。



源： D10 ~ D13 (重复=4)
目标： D12 ~ D15 (重复=4)



MOVN (求反传送)



S1 NOT→D1

输入打开时，来自 S1 所指定的设备的 16 位或 32 位数据将被逐位求反，并传送到 D1 所指定的设备。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要传送的起始设备编号	X	X	X	X	X	X	X	X	1-99
D1 (目标 1)	要传送到的起始设备编号	—	X	▲	X	X	X	X	—	1-99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

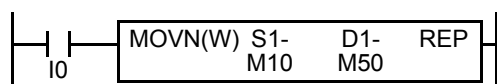
▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

有效数据类型

W (字型)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等设备指定为源或目标，则使用 16 点 (字型或整型数据类型) 或 32 点 (双字型或长整型数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整型)	X	
D (双字型)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型或整型数据类型) 或 2 点 (双字型或长整型数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整型)	X	
F (浮点型)	—	

示例 : MOVN

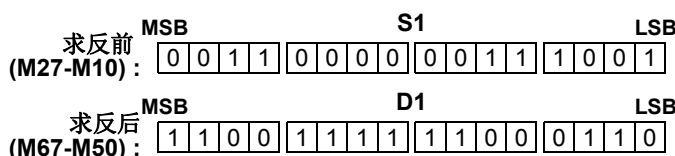


M10 NOT→M50

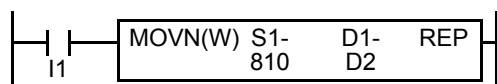
输入 I0 打开时，由源设备 S1 指定并以 M10 开始的 16 个内部继电器将被逐位求反，并传送到由目标设备 D1 指定并以 M50 开始的 16 个内部继电器。

M10 - M17, M20 - M27 NOT

→ M50 - M57, M60 - M67

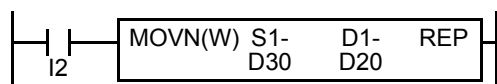
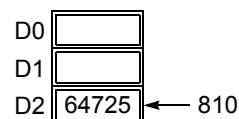
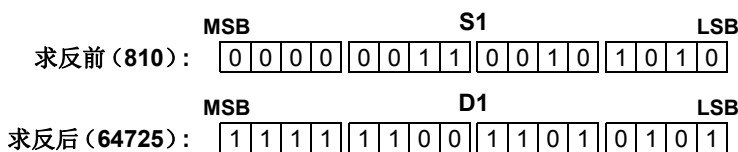


16 个内部继电器 (M10 - M17 和 M20 - M27) 的开 / 关状态将被求反，并传送到 16 个内部继电器 (M50 - M57 和 M60 - M67)。M50 是 LSB (最低有效位)，M67 是 MSB (最高有效位)。



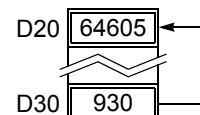
810 NOT→D2

输入 I1 打开时，由源设备 S1 指定的十进制常量 810 将转换为 16 位二进制数据，并且 16 位的开 / 关状态将被求反，并传送到由目标设备 D1 指定的数据寄存器 D2。



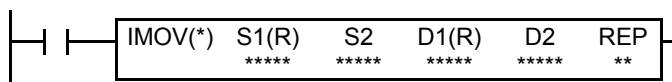
D30 NOT→D20

输入 I2 打开时，由 S1 指定的数据寄存器 D30 中的数据将被逐位求反，并传送到由 D1 指定的数据寄存器 D20。



3: 传送指令

IMOV (间接传送)



$S1 + S2 \rightarrow D1 + D2$

输入打开时，包含在由 S1 和 S2 指定的设备中的值将相加，以确定数据源。由此确定的 16 位或 32 位数据将传送到目标，而目标由包含在 D1 和 D2 所指定的设备中的值之和来确定。

浮点数据类型适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	作为传送源的基地址	X	X	X	X	X	X	X	—	1-99
S2 (源 2)	S1 的偏移量	X	X	X	X	X	X	X	—	—
D1 (目标 1)	作为传送目标的基地址	—	X	▲	X	X	X	X	—	1-99
D2 (目标 2)	D1 的偏移量	X	X	X	X	X	X	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1、S2 或 D2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

当指定为 F (浮点型) 数据类型时，只有数据寄存器能设置为 S1 或 D1。

当选择 F (浮点型) 数据类型，并且 S1 不符合标准浮点格式时，将导致程序执行错误，并打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

源设备 S2 或目标设备 D2 都不是必须指定的。如果不指定 S2 或 D2，则源设备或目标设备由没有偏移量的 S1 或 D1 确定。

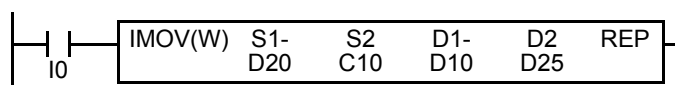
请确保由 S1 + S2 确定的源数据和由 D1 + D2 确定的目标数据在有效的设备范围以内。如果派生的源设备或目标设备超出了有效的设备范围，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

有效数据类型

W (字型)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字型数据类型) 或 32 点 (双字型数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整型)	—	
D (双字型)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型数据类型) 或 2 点 (双字型数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整型)	—	
F (浮点型)	X	

示例：IMOV

·数据类型：字型



$$D20 + C10 \rightarrow D10 + D25$$

源设备 S1 和目标设备 D1 确定设备的类型。源设备 S2 和目标设备 D2 是偏移量值，用于确定源设备和目标设备。

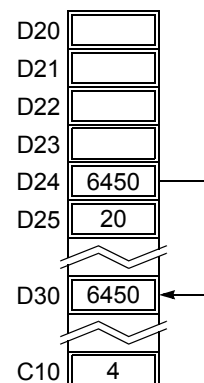
如果由源设备 S2 指定的计数器 C10 的当前值是 4，则源数据由偏移量加上源设备 S1 所指定的数据寄存器 D20 来确定：

$$D(20 + 4) = D24$$

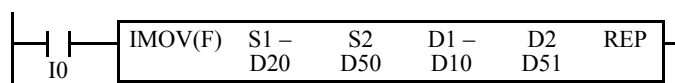
如果数据寄存器 D25 包含值 20，则通过偏移量加上目标设备 D1 所指定的数据寄存器 D10 来确定目标：

$$D(10 + 20) = D30$$

因此，当输入 I0 打开时，数据寄存器 D24 中的数据将传送到数据寄存器 D30。



·数据类型：浮点型



$$D20 + D50 \rightarrow D10 + D51$$

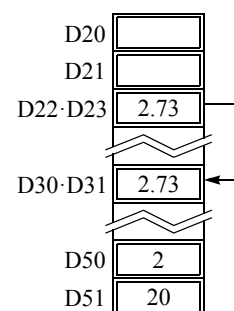
如果数据寄存器 D50 包含值 2，则通过偏移量加上源设备 S1 所指定的数据寄存器 D20 来确定：

$$D(20 + 2) = D22$$

如果数据寄存器 D51 包含值 20，则通过偏移量加上目标设备 D1 所指定的数据寄存器 D10 来确定：

$$D(10 + 20) = D30$$

因此，当输入 I0 打开时，数据寄存器 D22 · D23 中的数据将传送到数据寄存器 D30 · D31。



3: 传送指令

IMOVN (间接求反传送)



$S1 + S2 \text{ NOT} \rightarrow D1 + D2$

输入打开时，包含在由 S1 和 S2 指定的设备中的值将相加，以确定数据源。由此确定的 16 位或 32 位数据将被求反并传送到目标，而目标由包含在 D1 和 D2 所指定的设备中的值之和来确定。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	作为传送源的基地址	X	X	X	X	X	X	X	—	1-99
S2 (源 2)	S1 的偏移量	X	X	X	X	X	X	X	—	—
D1 (目标 1)	作为传送目标的基地址	—	X	▲	X	X	X	X	—	1-99
D2 (目标 2)	D1 的偏移量	X	X	X	X	X	X	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1、S2 或 D2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

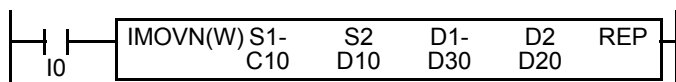
源设备 S2 或目标设备 D2 都不是必须指定的。如果不指定 S2 或 D2，则源设备或目标设备由没有偏移量的 S1 或 D1 确定。

请确保由 S1 + S2 确定的源数据和由 D1 + D2 确定的目标数据在有效的设备范围以内。如果派生的源设备或目标设备超出了有效的设备范围，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

有效数据类型

W (字型)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字型数据类型) 或 32 点 (双字型数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整型)	—	
D (双字型)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型数据类型) 或 2 点 (双字型数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整型)	—	
F (浮点型)	—	

示例 : IMOVN



$C10 + D10 \text{ NOT} \rightarrow D30 + D20$

源设备 S1 和目标设备 D1 确定设备的类型。源设备 S2 和目标设备 D2 是偏移量值，用于确定源设备和目标设备。

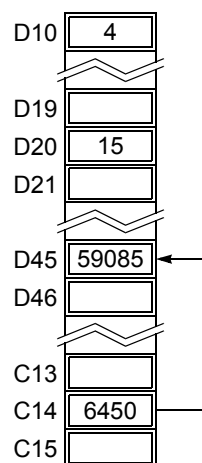
如果由源设备 S2 指定的数据寄存器 D10 的数据是 4，则通过偏移量加上由源设备 S1 指定的计数器 C10 来确定源数据：

$$C(10 + 4) = C14$$

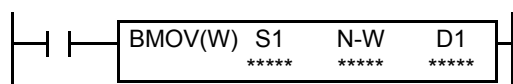
如果由目标设备 D2 指定的数据寄存器 D20 包含值 15，则通过偏移量加上由目标设备 D1 指定的数据寄存器 D30 来确定目标数据：

$$D(30 + 15) = D45$$

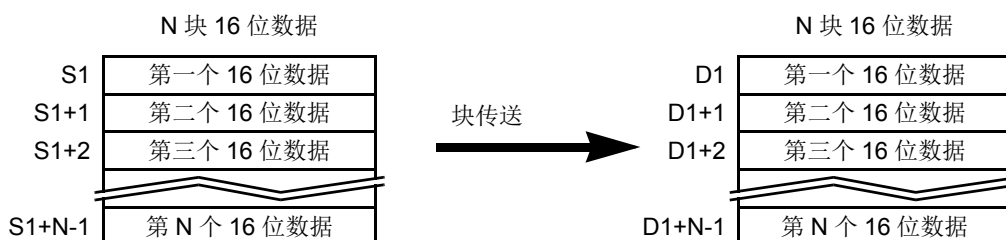
因此，当输入 I0 打开时，计数器 C14 的当前值将被求反，并传送到数据寄存器 D45。



BMOV (块传送)



$S1, S1+1, S1+2, \dots, S1+N-1 \rightarrow D1, D1+1, D1+2, \dots, D1+N-1$
 输入打开时, 以 $S1$ 指定的设备开始的 N 块 16 位字数据将传送到以 $D1$ 指定的设备开始的 N 块目标。 $N-W$ 指定要传送的块数量。



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要传送的起始设备编号	X	X	X	X	X	X	X	—	—
N-W (N 个字)	要传送的块数量	X	X	X	X	X	X	X	X	—
D1 (目标 1)	要传送到的起始设备编号	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 或 N-W 时, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时, 数据将作为预置值 (TP 或 CP) 写入, 此值介于 0 到 65535 之间。

请确保由 $S1+N-1$ 所确定的最后一个源数据和由 $D1+N-1$ 所确定的最后一个目标数据在有效的设备范围以内。如果派生的源设备或目标设备超出了有效的设备范围, 将导致用户程序执行错误, 这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

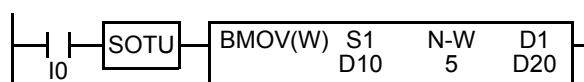
有效数据类型

W (字型)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源、N-W 或目标, 则使用 16 点 (字型数据类型)。
I (整型)	—	
D (双字型)	—	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源、N-W 或目标, 则使用 1 点 (字型数据类型)。
L (长整型)	—	
F (浮点型)	—	

特殊内部继电器 M8024: BMOV/WSFT 执行标记

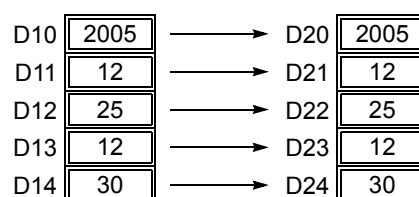
执行 BMOV 或 WSFT 时, M8024 将打开。完成后, M8024 将关闭。如果在执行 BMOV 或 WSFT 时 CPU 断电, 则当 CPU 再次通电时, M8024 将保持打开状态。

示例: BMOV



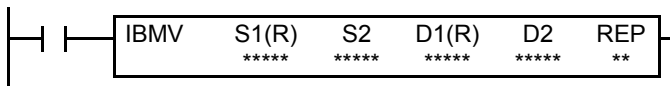
$D10 - D14 \rightarrow D20 - D24$

输入 I0 打开时, 以源设备 S1 所指定的 D10 开始的 5 个数据寄存器的数据将传送到以目标设备 D1 所指定的 D20 开始的 5 个数据寄存器。



3: 传送指令

IBMV (间接位传送)



$$S1 + S2 \rightarrow D1 + D2$$

输入打开时，包含在由 S1 和 S2 指定的设备中的值将相加，以确定数据源。由此确定的 1 位数据将传送到目标，而目标由包含在 D1 和 D2 所指定的设备中的值之和来确定。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	作为传送源的基地址	X	X	X	X	—	—	X	0 或 1	1-99
S2 (源 2)	S1 的偏移量	X	X	X	X	X	X	X	0-65535	—
D1 (目标 1)	作为传送目标的基地址	—	X	▲	X	—	—	X	—	1-99
D2 (目标 2)	D1 的偏移量	X	X	X	X	X	X	X	0-65535	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

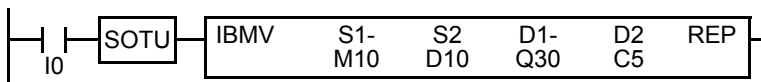
▲ 可以将内部继电器 M0 - M2557 指定为 S2。不能将特殊内部继电器指定为 S2。

当 T (定时器) 或 C (计数器) 用作 S2 或 D2 时，将读取定时器 / 计数器的当前值 (TC 或 CC)。

请确保由 S1+S2 确定的最后一个源数据和由 D1+D2 确定的最后一个目标数据在有效的设备范围以内。如果派生源设备或目标设备超出了有效的设备范围，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

源设备 S2 或目标设备 D2 都不是必须指定的。如果不指定 S2 或 D2，则源设备或目标设备由没有偏移量的 S1 或 D1 确定。

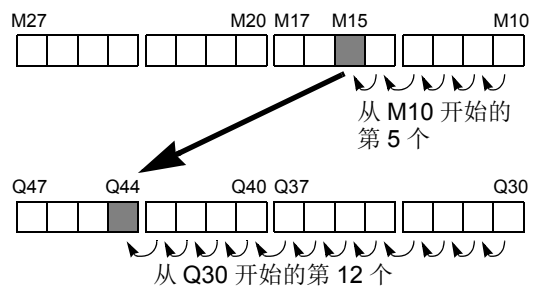
示例 : IBMV



$$M10 + D10 \rightarrow Q30 + C5$$

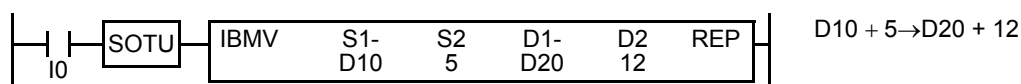
源设备 S1 和目标设备 D1 确定设备的类型。源设备 S2 和目标设备 D2 是偏移量值，用于确定源设备和目标设备。

如果由源设备 S2 指定的数据寄存器 D10 的值是 5，则通过偏移量加上源设备 S1 所指定的内部继电器 M10 来确定源数据。



如果目标设备 D2 所指定的计数器 C5 的当前值是 12，则通过偏移量加上目标设备 D1 所指定的输出 Q30 来确定目标数据。

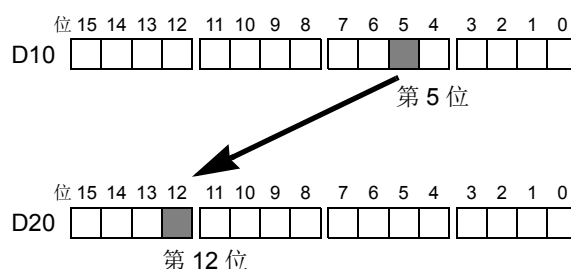
因此，当输入 I0 打开时，内部继电器 M15 的开 / 关状态将传送到输出 Q44。



由于源设备 S1 是数据寄存器，并且源设备 S2 的值是 5，因此，源数据是由源设备 S1 所指定的数据寄存器 D10 的第 5 位。

由于目标设备 D1 是数据寄存器，并且源设备 D2 的值是 12，因此目标数据是目标设备 D1 所指定的数据寄存器 D20 的第 12 位。

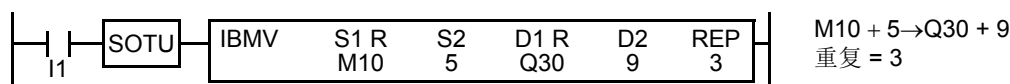
因此，当输入 I0 打开时，数据寄存器 D10 的第 5 位的开 / 关状态将传送到数据寄存器 D20 的第 12 位。



间接位传送指令中的重复操作

重复位设备（源和目标）

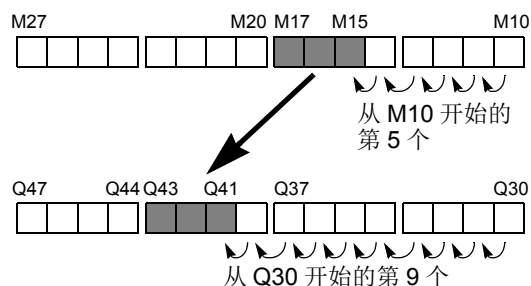
如果对位设备（例如，输入、输出、内部继电器或移位寄存器）指定了重复操作，将传送与重复周期一样多的位设备。



由于源设备 S1 是内部继电器 M10，并且源设备 S2 的值是 5，因此，源数据是以 M15 开始的 3 个内部继电器。

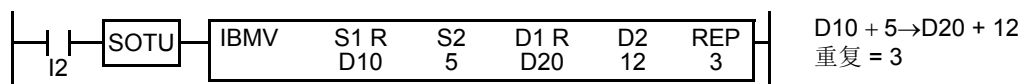
由于目标设备 D1 是输出 Q30，并且目标设备 D2 的值是 9，因此，目标数据是以 Q41 开始的 3 个输出。

因此，当输入 I1 打开时，内部继电器 M15 - M17 的开 / 关状态将传送到输出 Q41 - Q43。



重复字设备（源和目标）

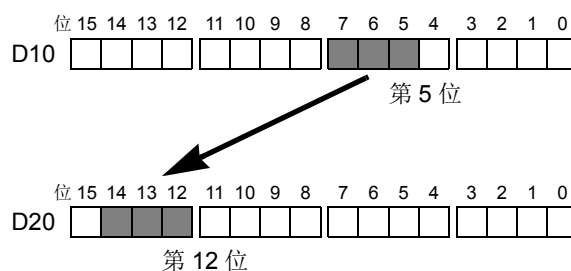
如果对字设备（例如，数据寄存器）指定了重复操作，将传送指定数据寄存器中与重复周期一样多的位状态。



由于源设备 S1 是数据寄存器 D10，并且源设备 S2 的值是 5，因此，源数据是以数据寄存器 D10 的第 5 位开始的 3 个位。

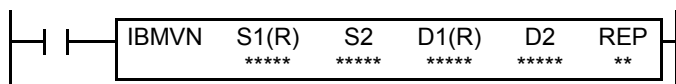
由于目标设备 D1 是数据寄存器 D20，并且目标设备 D2 的值是 12，因此，目标数据是以数据寄存器 D20 的第 12 位开始的 3 个位。

因此，当输入 I2 打开时，数据寄存器 D10 的第 5 - 7 位的开 / 关状态将传送到数据寄存器 D20 的第 12 - 14 位。



3: 传送指令

IBMVN (间接位求反传送)



$$S1 + S2 \text{ NOT} \rightarrow D1 + D2$$

输入打开时，包含在由 S1 和 S2 指定的设备中的值将相加，以确定数据源。由此确定的 1 位数据将被求反并传送到目标，而目标由包含在 D1 和 D2 所指定的设备中的值之和来确定。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	作为传送源的基地址	X	X	X	X	—	—	X	0 或 1	1-99
S2 (源 2)	S1 的偏移量	X	X	X	X	X	X	X	0-65535	—
D1 (目标 1)	作为传送目标的基地址	—	X	▲	X	—	—	X	—	1-99
D2 (目标 2)	D1 的偏移量	X	X	X	X	X	X	X	0-65535	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

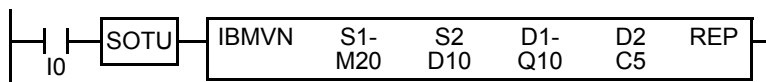
▲ 可以将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

当 T (定时器) 或 C (计数器) 用作 S2 或 D2 时，将读取定时器 / 计数器的当前值 (TC 或 CC)。

请确保由 S1+S2 确定的最后一个源数据和由 D1+D2 确定的最后一个目标数据在有效的设备范围以内。如果派生源设备或目标设备超出了有效的设备范围，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

源设备 S2 或目标设备 D2 都不是必须指定的。如果不指定 S2 或 D2，则源设备或目标设备由没有偏移量的 S1 或 D1 确定。

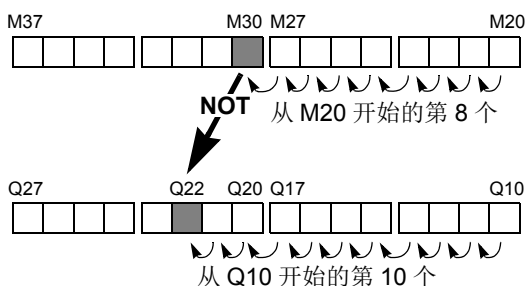
示例: IBMVN



$$M20 + D10 \text{ NOT} \rightarrow Q10 + C5$$

源设备 S1 和目标设备 D1 确定设备的类型。源设备 S2 和目标设备 D2 是偏移量值，用于确定源设备和目标设备。

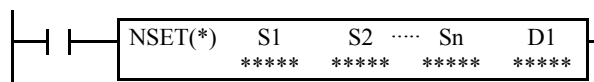
如果源设备 S2 所指定的数据寄存器 D10 的值是 8，则通过偏移量加上源设备 S1 所指定的内部继电器 M20 来确定源数据。



如果目标设备 D2 所指定的计数器 C5 的当前值是 10，则通过偏移量加上目标设备 D1 所指定的输出 Q10 来确定目标数据。

因此，当输入 I0 打开时，内部继电器 M30 的开 / 关状态将被求反，并传送到输出 Q22。

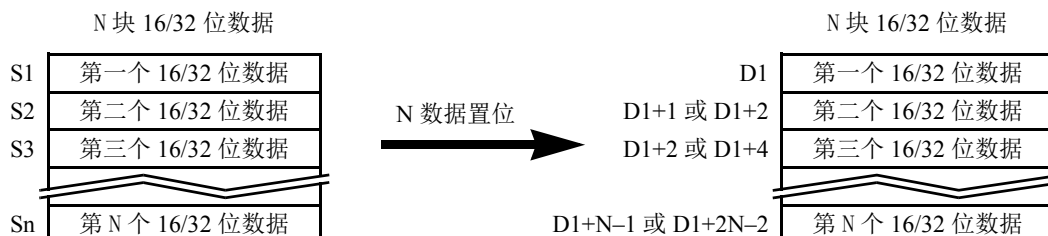
NSET(N 数据置位)



$S1, S2, S3, \dots, Sn \rightarrow D1, D2, D3, \dots, Dn$

输入打开时，以 $S1, S2, S3, \dots, Sn$ 指定的设备的 N 块 16 位或 32 位字数据将传送到以 $D1$ 指定的设备开始的 N 块目标。

此指令适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要传送的起始设备编号	X	X	X	X	X	X	X	X	-
D1 (目标 1)	要传送到的起始设备编号	-	X	▲	X	X	X	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

▲ 不能将特殊内部继电器指定为 $D1$ 。

当 T (定时器) 或 C (计数器) 用作 $S1$ 时，将读取定时器 / 计数器的当前值 (TC 或 CC)。当 T (定时器) 或 C (计数器) 用作 $D1$ ，将数据作为预置值 (TP 或 CP) 写入。

当选择为 F (浮点型) 数据类型时，只有数据寄存器和常量才能指定为 $S1$ ，数据寄存器才能指定为 $D1$ 。

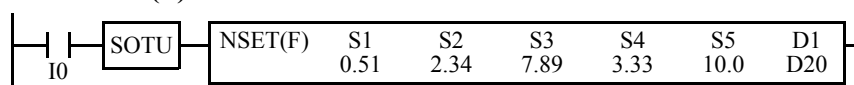
当选择 F (浮点型) 数据类型，并且 $S1$ 不符合标准浮点格式时，将导致程序执行错误，并打开 CPU 模块上的特殊继电器 $M8004$ 和 ERR LED。

请确保由 $D1+N-1$ (字型或整数型) 或 $D1+2N-2$ (双字型、长整数型或浮点型) 所确定的最后一个源数据在有效的设备范围内。如果相加的源设备超出了有效的设备范围，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 $M8004$ 和 ERR LED。

有效数据类型

W (字)	X	如果将 I (输入)、 Q (输出)、 M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字型或整数型数据类型) 或 32 点 (双字型或长整数型数据类型)。
I (整数)	X	如果将 T (定时器)、 C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型或整数型数据类型) 或 2 点 (双字型、长整数型或浮点型数据类型)。
D (双字)	X	
L (长整数)	X	
F (浮点)	X	

示例：NSET(F)



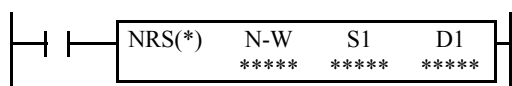
五个常量 0.51, 2.34, 7.89, 3.33 和 10.0 \rightarrow $D20 \sim D29$

输入 $I0$ 打开时，以源设备 $S1 \sim S5$ 所指定的 $D20$ 开始的 10 个数据寄存器的数据将传送到以目标设备 $D1$ 所指定的 $D20$ 开始的 10 个数据寄存器。



3: 传送指令

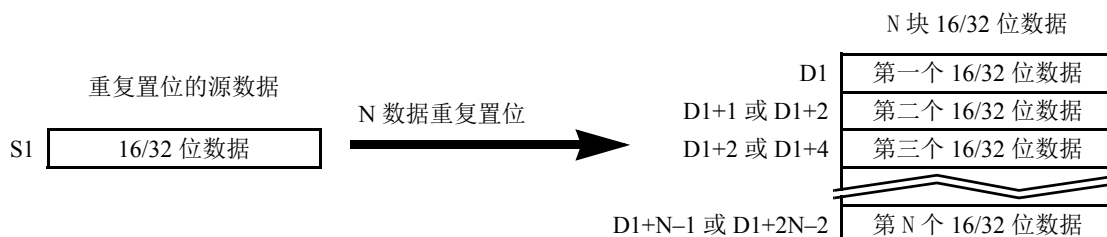
NRS (N 数据重复置位)



S1 → D1, D2, D3, ..., Dn-1

输入打开时，以 S1 指定的 16 位或 32 位字数据将置位到以 D1 指定的设备开始的 N 块目标。

此指令适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
N-W (N 块)	要传送的块数量	X	X	X	X	X	X	X	X	-
S1 (源 1)	要传送的起始设备编号	X	X	X	X	X	X	X	X	-
D1 (目标 1)	要传送到的起始设备编号	-	X	▲	X	X	X	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 或 N-W 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入。

当选择为 F (浮点型) 数据类型时，只有数据寄存器和常量才能指定为 S1，数据寄存器才能指定为 D1。

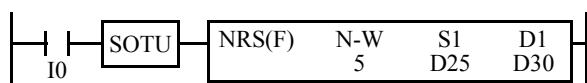
当选择 F (浮点型) 数据类型，并且 S1 不符合标准浮点格式时，将导致程序执行错误，并打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

请确保由 D1+N-1 (字型或整数型) 或 D1+2N-2 (双字型、长整数型或浮点型) 所指定的最后一个源数据在有效的设备范围内。如果相加的源设备超出了有效的设备范围，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

有效数据类型

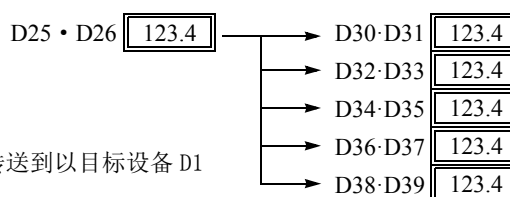
W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字型或整数型数据类型) 或 32 点 (双字型或长整数型数据类型)。
I (整数)	X	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型或整数型数据类型) 或 2 点 (双字型、长整数型或浮点型数据类型)。
L (长整数)	X	
F (浮点)	X	

示例 : NRS(F)

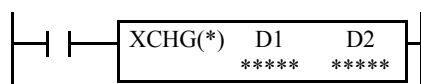


D25 · D26 → D30 ~ D39

输入 I0 打开时，以源设备 S1 所指定的数据寄存器 D25 · D26 的数据将传送到以目标设备 D1 所指定的 D30 开始的 10 个数据寄存器。



XCHG (交换)



字数据类型: D1 ↔ D2
 双字数据类型: D1·D1+1 → D2, D2+1

当输入打开, 由设备 D1 和 D2 指定的 16 位或 32 位数将相互交换。

本指令适用于升级后的系统程序 Ver. 210 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
D1 (目标 1)	用于交换的起始设备编号	-	X	▲	X	-	-	X	-	-
D2 (目标 2)	用于交换的起始设备编号	-	X	▲	X	-	-	X	-	-

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

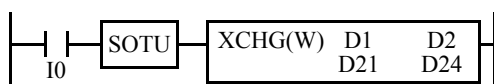
▲ 内部继电器 M0 ~ M2557 可以指定为 D1 或 D2。特殊内部继电器不能指定为 D1 或 D2。

有效数据类型

W (字)	X	如果将 Q (输出)、M (内部继电器) 或 R (移位寄存器) 等设备指定为目标, 则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	-	
D (双字)	X	如果将 D (数据寄存器) 等字设备指定为目标, 则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	-	
F (浮点)	-	

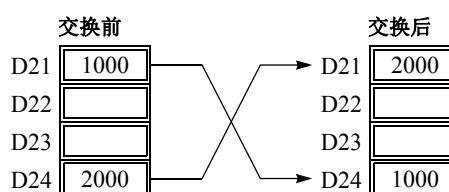
示例: XCHG

·数据类型: 字

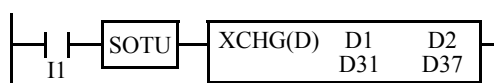


D21 ↔ D24

输入 I0 打开时, 由设备 D1 和 D2 指定的数据寄存器 D20 和 D24 的数据将相互交换。

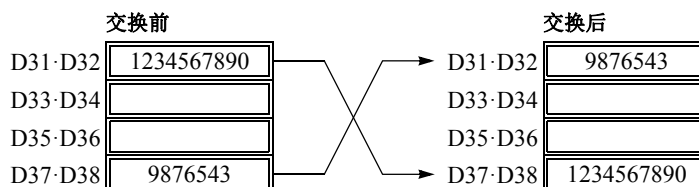


·数据类型: 双字



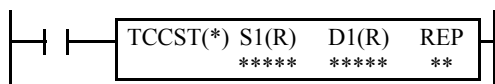
D31·D32 ↔ D37·D38

输入 I1 打开时, 由设备 D1 和 D2 指定的数据寄存器 D31·D32 和 D37·D38 的数据将相互交换。



3: 传送指令

TCCST(存储定时器 / 计数器当前值)



S1 → D1

输入打开时，以 S1 指定的 16 位或 32 位字数据将读取和存储到以 D1 指定的设备的当前值。

此指令适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要传送的起始设备编号	X	X	X	X	X	X	X	X	1 ~ 99
D1 (目标 1)	要传送到的起始设备编号	-	-	-	-	X	X	-	-	1 ~ 99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当 T (定时器) 或 C (计数器) 用作 S1 时，将读取定时器 / 计数器的当前值 (TC 或 CC)。当 T (定时器) 或 C (计数器) 用作 D1，将数据作为预置值 (TP 或 CP) 写入。

由于 TCCST 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

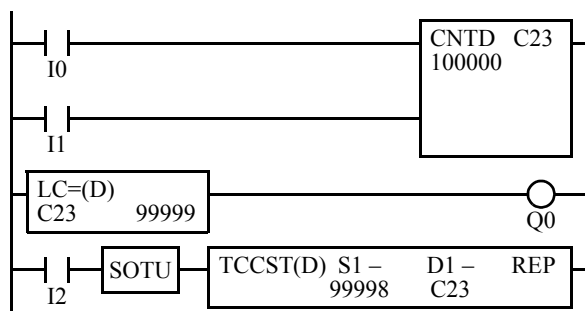
有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源，则使用 16 点 (整数型数据类型) 或 32 点 (双字型数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整数)	-	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型或整数型数据类型) 或 2 点 (双字型、长整数型或浮点型数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整数)	-	
F (浮点)	-	

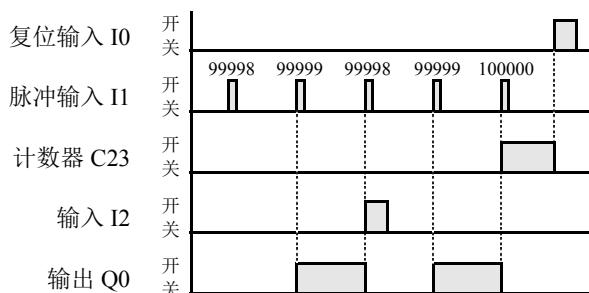
示例 : TCCST

当打开输入 I2，99998 将写入到计数器 C23 的当前值。

梯形图程序



时序图



4: 数据比较指令

简介

可以使用数据比较指令来比较数据，例如等于、不等于、小于、大于、小于或等于和大于或等于。如果比较结果一致，将打开输出或内部继电器。还可以用重复操作来比较多组数据。

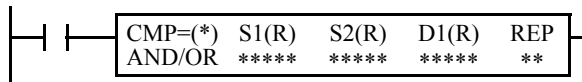
当启用使用重复操作时，新的逻辑 OR 运算选项加入到 CMP 指令。从 AND 或 OR 运算中可以选择 CMP 指令的重复比较结果，并将该结果输出到输出或内部继电器中。本选项适用于升级后的系统程序版本 200 或更高的 CPU 模块。

使用 ICMP>= 指令也可以比较三个值。

添加了接点比较指令。后跟的指令是否执行取决于比较结果。这些指令适用于升级后的系统程序版本 200 或更高的 CPU 模块。

由于数据比较指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

CMP= (比较等于)

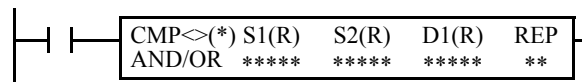


数据类型 W 或 I: $S1 = S2 \rightarrow D1$ 打开

数据类型 D、L 或 F: $S1 \cdot S1+1 = S2 \cdot S2+1 \rightarrow D1$ 打开

当输入打开时，将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据等于 S2 数据时，目标设备 D1 将打开。条件不满足时，D1 将关闭。

CMP<> (比较不等于)

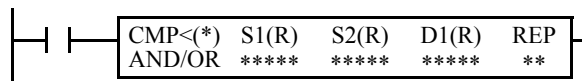


数据类型 W 或 I: $S1 \neq S2 \rightarrow D1$ 打开

数据类型 D、L 或 F: $S1 \cdot S1+1 \neq S2 \cdot S2+1 \rightarrow D1$ 打开

当输入打开时，将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据不等于 S2 数据时，目标设备 D1 将打开。条件不满足时，D1 将关闭。

CMP< (比较小于)

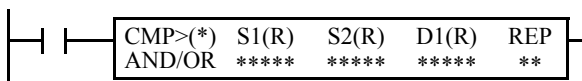


数据类型 W 或 I: $S1 < S2 \rightarrow D1$ 打开

数据类型 D、L 或 F: $S1 \cdot S1+1 < S2 \cdot S2+1 \rightarrow D1$ 打开

当输入打开时，将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据小于 S2 数据时，目标设备 D1 将打开。条件不满足时，D1 将关闭。

CMP> (比较大于)

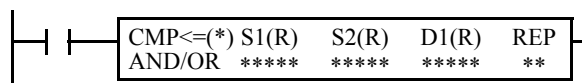


数据类型 W 或 I: $S1 > S2 \rightarrow D1$ 打开

数据类型 D、L 或 F: $S1 \cdot S1+1 > S2 \cdot S2+1 \rightarrow D1$ 打开

当输入打开时，将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据大于 S2 数据时，目标设备 D1 将打开。条件不满足时，D1 将关闭。

CMP<= (比较小于或等于)



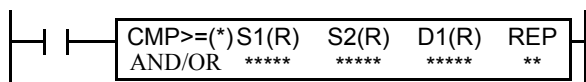
数据类型 W 或 I: $S1 \leq S2 \rightarrow D1$ 打开

数据类型 D、L 或 F: $S1 \cdot S1+1 \leq S2 \cdot S2+1 \rightarrow D1$ 打开

当输入打开时，将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据小于或等于 S2 数据时，目标设备 D1 将打开。条件不满足时，D1 将关闭。

4: 数据比较指令

CMP>= (比较大于或等于)



数据类型 W 或 I: $S1 \geq S2 \rightarrow D1$ 打开
 数据类型 D、L 或 F: $S1 \cdot S1+1 \geq S2 \cdot S2+1 \rightarrow D1$ 打开
 当输入打开时, 将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据大于或等于 S2 数据时, 目标设备 D1 将打开。
 条件不满足时, D1 将关闭。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
重复结果	逻辑 AND 或 OR 指令	-	-	-	-	-	-	-	-	-
S1 (源 1)	要比较的数据	X	X	X	X	X	X	X	X	1-99
S2 (源 2)	要比较的数据	X	X	X	X	X	X	X	X	1-99
D1 (目标 1)	比较输出	-	X	▲	-	-	-	-	-	1-99

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

当只重复 S1 和 / 或 S2 时, 逻辑运算类型可以选择 AND 或 OR。逻辑运算 OR 适用于升级后的系统程序版本 200 或更高的 CPU 模块。

▲可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

把 T (定时器) 或 C (计数器) 用作 S1 或 S2 时, 将读取定时器 / 计数器的当前值 (TC 或 CC)。

当选择 F (浮点) 数据类型时, 只有数据寄存器和常量可指定为 S1 和 S2。

当选择 F (浮点) 数据类型, 并且 S1 或 S2 不符合标准浮点格式时, 将导致程序执行错误, 打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

有效数据类型

W (字)	X
I (整数)	X
D (双字)	X
L (长整数)	X
F (浮点)	X

如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源, 则使用 16 点 (字或整数数据类型) 或 32 点 (双字、长整数数据类型)。如果对位设备指定重复, 则设备位的数量将以 16 点或 32 点为增量增加。

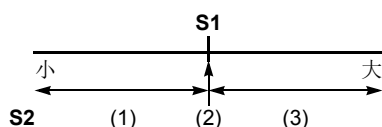
如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源, 则使用 1 点 (字或整数数据类型) 或 2 点 (双字、长整数或浮点数据类型)。如果对字设备指定重复, 则设备字的数量将以 1 点或 2 点为增量增加。

当将输出或内部继电器指定为目标时, 不论选择什么数据类型都只使用 1 点。为目标指定重复时, 将使用与重复周期一样多的输出或内部继电器。

CMP= 中的特殊内部继电器 M8150、M8151 和 M8152

有三个特殊内部继电器可以用于指示 CMP= 指令的比较结果。根据结果, 三个特殊内部继电器中将有一个会打开。

S1 > S2 时, M8150 (大于) 将打开。
 S1 = S2 时, M8151 (等于) 将打开。
 S1 < S2 时, M8152 (小于) 将打开。



S2 值	M8150	M8151	M8152	D1 状态
(1) S1 > S2	开	关	关	关
(2) S1 = S2	关	开	关	开
(3) S1 < S2	关	关	开	关

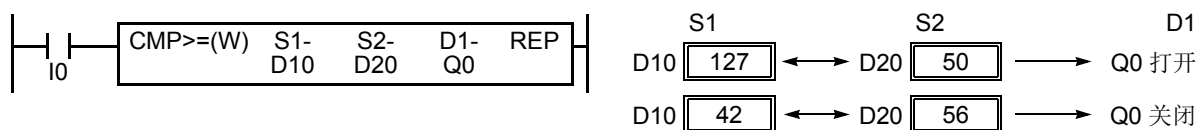
指定重复时, 最后一个重复周期的比较结果将打开三个特殊内部继电器中的某一个。

使用多个 CMP= 或 ICMP>= 指令时, M8150、M8151 或 M8152 将指示最后执行的指令的结果。

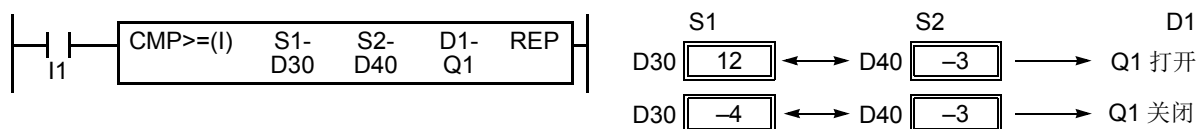
示例: **CMP>=**

以下示例描述如何使用 CMP 指令。所有其他数据比较指令的数据比较操作均与 CMP 指令相同。

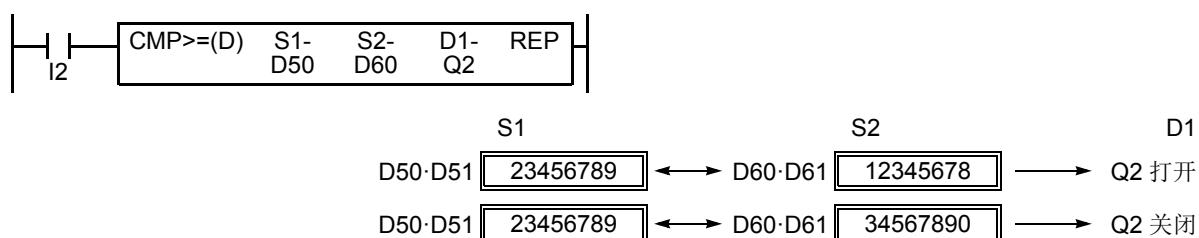
·数据类型：字



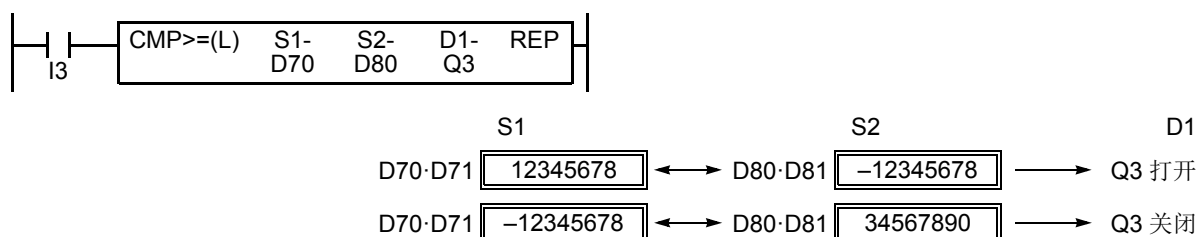
·数据类型：整数



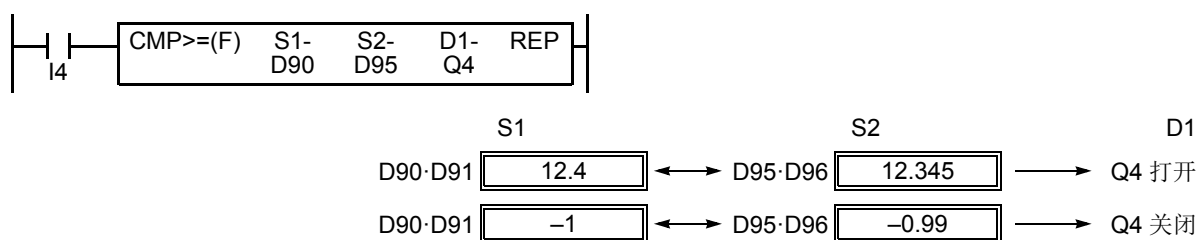
·数据类型：双字



·数据类型：长整数



·数据类型：浮点



4: 数据比较指令

数据比较指令中的重复操作

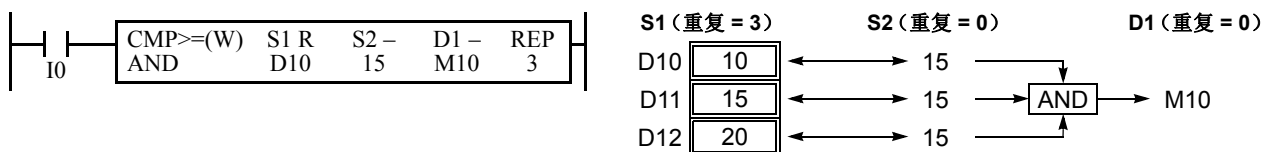
以下示例使用字和双字数据类型的 CMP \geq 指令描述。所有其他数据比较指令和其他数据类型的重复操作均与下列相同。

当启用使用重复操作时，新的逻辑 OR 运算选项加入到 CMP 指令。从 AND 或 OR 运算中可以选择 CMP 指令的重复比较结果，并将该结果输出到输出或内部继电器中。本选项适用于升级后的系统程序 Ver.200 或更高版本的 CPU 模块。

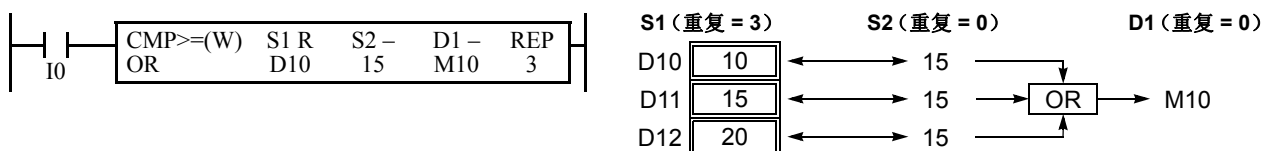
重复一个源设备

仅指定 S1（源）重复时，源设备（与重复周期一样多，并以 S1 所指定的设备起始）将与 S2 所指定的设备进行比较。比较结果将作 AND 运算，并设置到由 D1 所指定的目标设备中。

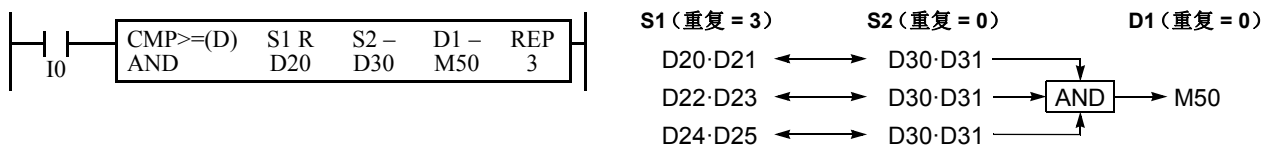
·数据类型：字（重复逻辑操作 AND）



·数据类型：字（重复逻辑操作 OR）



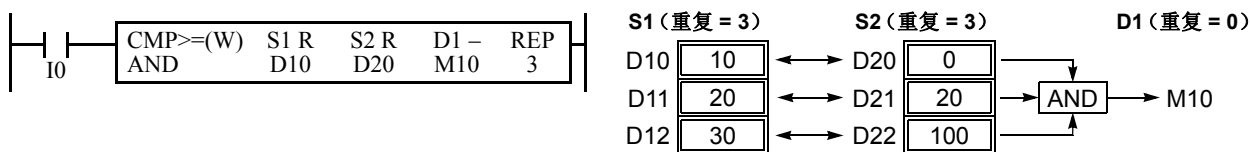
·数据类型：双字（重复逻辑操作 AND）



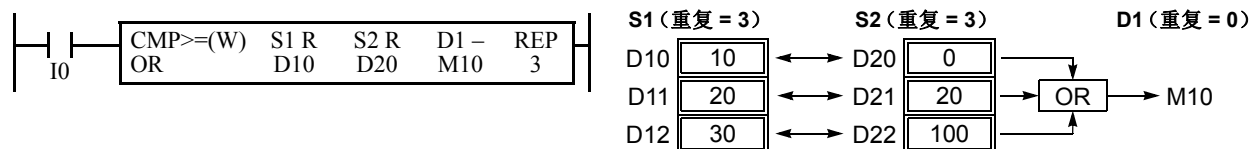
重复两个源设备

指定 S1（源）和 S2（源）重复时，源设备（与重复周期一样多，并以 S1 和 S2 所指定的设备起始）将相互进行比较。比较结果将作 AND 运算，并设置到由 D1 所指定的目标设备中。

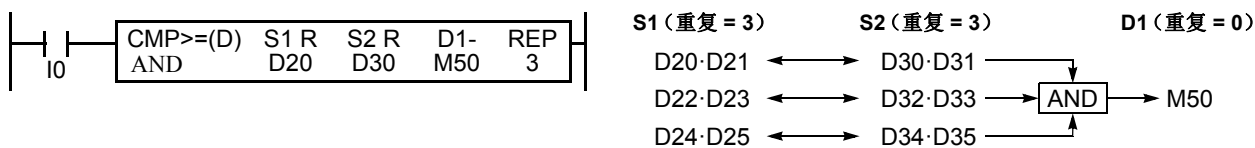
·数据类型：字（重复逻辑操作 AND）



·数据类型：字（重复逻辑操作 OR）



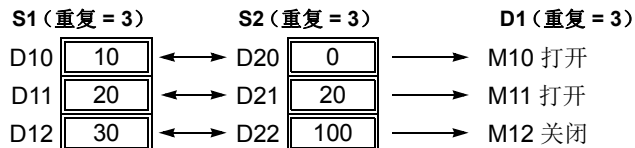
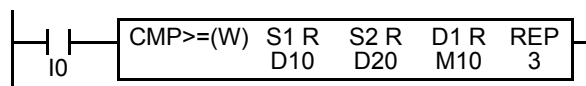
·数据类型：双字（重复逻辑操作 AND）



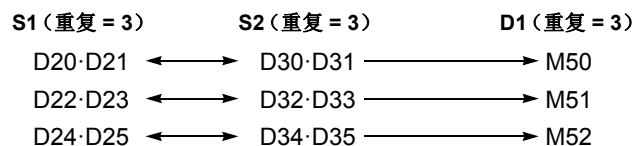
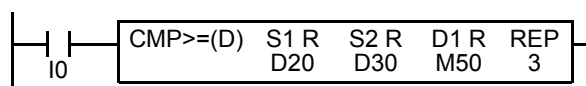
重复源设备和目标设备

指定 S1、S2（源）和 D1（目标）重复时，源设备（与重复周期一样多，并以 S1 和 S2 所指定的设备开始）将相互比较。比较结果将设置到目标设备（与重复周期一样多，并以 D1 所指定的设备开始）。

·数据类型：字

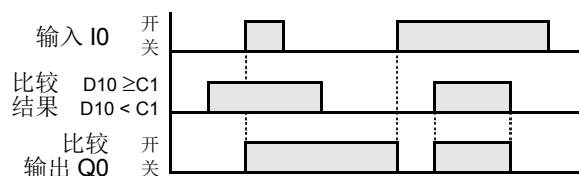
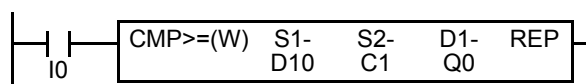


·数据类型：双字

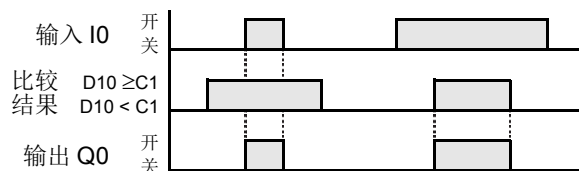
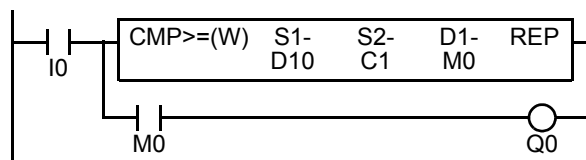


比较输出状态

当数据比较指令的输入已关闭时，比较输出通常将被保持。如果比较输出打开，那么当输入关闭时，将保持打开状态，如此程序所示。



当输入关闭时，此程序将关闭输出。



4: 数据比较指令

ICMP>= (间隔比较大于或等于)



数据类型 W 或 I: $S1 \geq S2 \geq S3 \rightarrow D1$ 打开
 数据类型 D、L 或 F: $S1 \cdot S1+1 \geq S2 \cdot S2+1 \geq S3 \cdot S3+1 \rightarrow D1$ 打开
 输入打开时, 将比较由 S1、S2 和 S3 指定的 16 位或 32 位数据。条件满足时, 目标设备 D1 将打开。条件不满足时, D1 将关闭。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要比较的数据	X	X	X	X	X	X	X	X	—
S2 (源 2)	要比较的数据	X	X	X	X	X	X	X	X	—
S3 (源 3)	要比较的数据	X	X	X	X	X	X	X	X	—
D1 (目标 1)	比较输出	—	X	▲	—	—	—	—	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1、S2 或 S3 时, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。

当选择 F (浮点) 数据类型时, 只有数据寄存器和常量可指定为 S1、S2 和 S3。

当选择 F (浮点) 数据类型, 并且 S1、S2 或 S3 不符合标准浮点格式时, 将导致程序执行错误, 打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

如果 S1 的数据小于 S3 的数据 ($S1 < S3$), 将导致用户程序执行错误, 这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

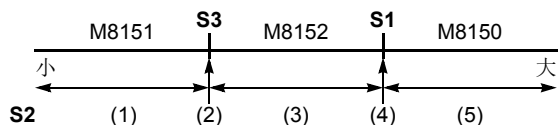
有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等设备指定为源, 则使用 16 点 (字或整数数据类型) 或 32 点 (双字或长整数数据类型)。
I (整数)	X	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源, 则使用 1 点 (字或整数数据类型) 或 2 点 (双字、长整数或浮点数据类型)。
L (长整数)	X	
F (浮点)	X	不管选择了什么数据类型, 目标只使用一个输出或内部继电器。

特殊内部继电器 M8150、M8151 和 M8152 用于 ICMP>=

有三个特殊内部继电器可以用于指示 ICMP>= 指令的比较结果。根据结果, 三个特殊内部继电器中将有一个会打开。S1 必须总是大于或等于 S3 ($S1 \geq S3$)。

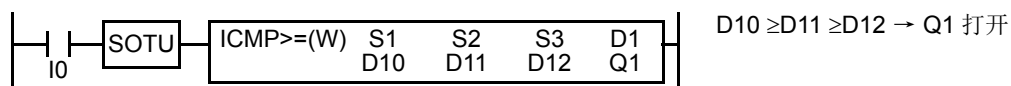
S2 > S1 时, M8150 将打开。
 S2 < S3 时, M8151 将打开。
 S1 > S2 > S3 时, M8152 将打开。



S2 值	M8150	M8151	M8152	D1 状态
(1) S2 < S3	关	开	关	关
(2) S2 = S3	关	关	关	开
(3) S3 < S2 < S1	关	关	开	开
(4) S2 = S1	关	关	关	开
(5) S2 > S1	开	关	关	关

如果使用了多个 ICMP>= 或 CMP= 指令, 则 M8150、M8151 或 M8152 将指示最后执行的指令的结果。

示例: ICMP>=

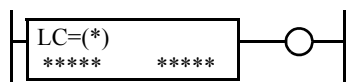


输入 I0 打开时，将比较由源设备 S1、S2 和 S3 所指定的数据寄存器 D10、D11 和 D12 的数据。当条件满足时，目标设备 D1 所指定的内部继电器 Q1 将打开。条件不满足时，Q1 将关闭。

	S1		S2		S3		D1	M8150	M8151	M8152	M8004		
D10	17	>	D11	15	=	D12	15	→	Q1 打开	关	关	关	关
D10	15	<	D11	18	<	D12	19	→	Q1 关闭	开	开	关	开

4: 数据比较指令

LC= (接点比较等于)



数据类型 W 或 I: $S1 = S2$

数据类型 D, L 或 F: $S1 \cdot S1+1 = S2 \cdot S2+1$

该指令将不断的比较由 S1 和 S2 所指定的 16 或 32 位数据。S1 数据等于 S2 数据时，打开紧跟输出的指令。条件不满足时，关闭输出。

LC<> (接点比较不等于)

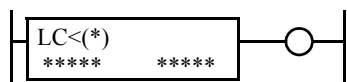


数据类型 W 或 I: $S1 \neq S2$

数据类型 D, L 或 F: $S1 \cdot S1+1 \neq S2 \cdot S2+1$

该指令将不断的比较由 S1 和 S2 所指定的 16 或 32 位数据。S1 数据不等于 S2 数据时，打开紧跟输出的指令。条件不满足时，关闭输出。

LC< (接点比较小于)

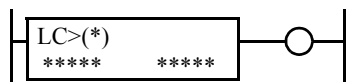


数据类型 W 或 I: $S1 < S2$

数据类型 D, L 或 F: $S1 \cdot S1+1 < S2 \cdot S2+1$

该指令将不断的比较由 S1 和 S2 所指定的 16 或 32 位数据。S1 数据小于 S2 数据时，打开紧跟输出的指令。条件不满足时，关闭输出。

LC> (接点比较大于)

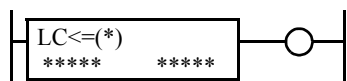


数据类型 W 或 I: $S1 > S2$

数据类型 D, L 或 F: $S1 \cdot S1+1 > S2 \cdot S2+1$

该指令将不断的比较由 S1 和 S2 所指定的 16 或 32 位数据。S1 数据大于 S2 数据时，打开紧跟输出的指令。条件不满足时，关闭输出。

LC<= (接点比较小于或等于)

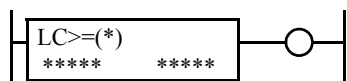


数据类型 W 或 I: $S1 \leq S2$

数据类型 D, L 或 F: $S1 \cdot S1+1 \leq S2 \cdot S2+1$

该指令将不断的比较由 S1 和 S2 所指定的 16 或 32 位数据。S1 数据小于或等于 S2 数据时，打开紧跟输出的指令。条件不满足时，关闭输出。

LC>= (接点比较大于或等于)



数据类型 W 或 I: $S1 \geq S2$

数据类型 D, L 或 F: $S1 \cdot S1+1 \geq S2 \cdot S2+1$

该指令将不断的比较由 S1 和 S2 所指定的 16 或 32 位数据。S1 数据大于或等于 S2 数据时，打开紧跟输出的指令。条件不满足时，关闭输出。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要比较的数据	X	X	X	X	X	X	X	X	-
S2 (源 2)	要比较的数据	X	X	X	X	X	X	X	X	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

在使用 T (定时器) 或 C (计数器) 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。

当被选择为 F (浮点型) 数据类型时，只能指定数据寄存器和常量。

当选择 F (浮点型) 数据类型，并且 S1 或 S2 不符合标准浮点格式时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。输出将被关闭。

有效数据类型

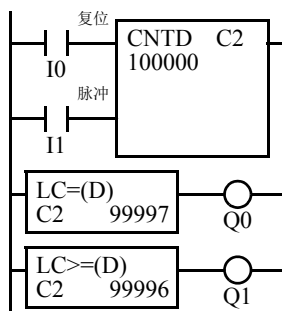
W(字)	X
I(整数)	X
D(双字)	X
L(长整数)	X
F(浮点)	X

如果 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定, 则使用 16 点 (字型或整数型数据类型) 或 32 点 (双字型或长整数型数据类型)。

如果 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备被指定, 则使用 1 点 (字型或整数型数据类型) 或 2 点 (双字型、长整数型或浮点型数据类型)。

示例: LC

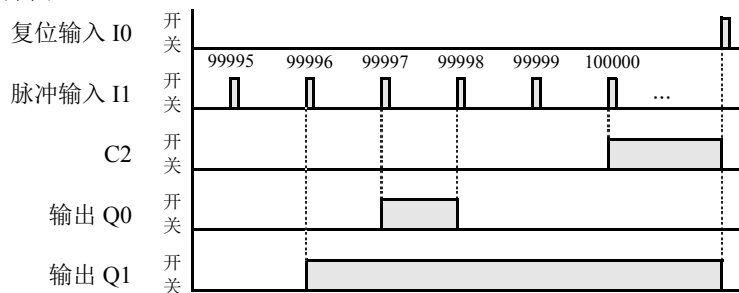
梯形图 1



程序列表

指令	数据
LOD	I0
LOD	I1
CNTD	C2 100000
LC=(D)	C2 99997
OUT	Q0
LC>=(D)	C2 99996
OUT	Q1

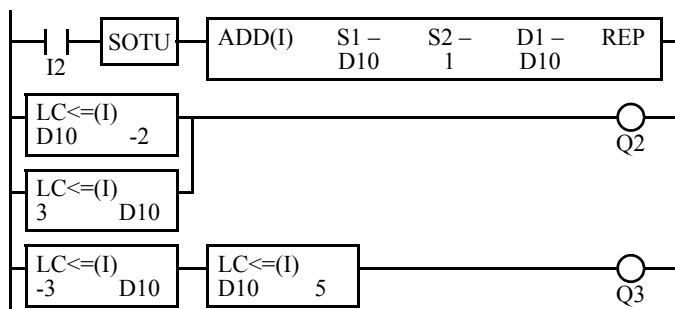
时序图



当计数器 C2 当前值为 99997, 打开输出 Q0。

当计数器 C2 当前值到达 99996 时, 打开输出 Q0, 并保留到计数器 C2 被复位时为止。

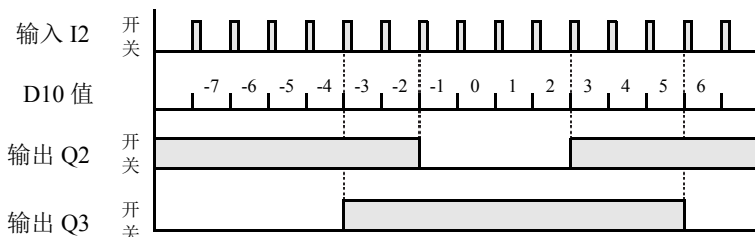
梯形图 2



程序列表

指令	数据
LOD	I12
SOTU	
ADD(I)	D10 1
LC<=(I)	D10 -2
OUT	Q2
LC<=(I)	D10 3
OUT	Q3
ANDLOD	
LC<=(I)	D10 5
OUT	Q3

时序图



当数据寄存器 D10 小于等于 -2 或大于等于 3 时, 打开输出 Q2。

数据寄存器 D10 为 -3 ~ 5 时, 打开输出 Q3。

4: 数据比较指令

5: 四则运算指令

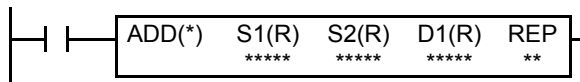
简介

四则运算指令让用户能够使用加法、减法、乘法、除法来编写运算程序。对于加和减设备，将用内部继电器 M8003 来进位或借位。

ROOT 指令可以用来计算存储在一个或两个数据寄存器中的值的平方根。

INC (递增)、DEC (递减)、SUM (合计) 和 RNDM (随机) 添加到升级后的系统程序版本 210 或更高版本的 CPU 模块。

ADD (加法)

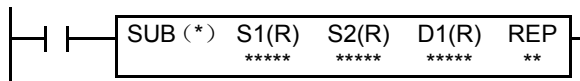


数据类型 W 或 I: $S1 + S2 \rightarrow D1, CY$

数据类型 D、L 或 F: $S1 \cdot S1+1 + S2 \cdot S2+1 \rightarrow D1 \cdot D1+1, CY$

当输入打开时，将添加由源设备 S1 和 S2 所指定的 16 位或 32 位数据。结果将被设置到目标设备 D1 和进位 (M8003)。

SUB (减法)



数据类型 W 或 I: $S1 - S2 \rightarrow D1, BW$

数据类型 D、L 或 F: $S1 \cdot S1+1 - S2 \cdot S2+1 \rightarrow D1 \cdot D1+1, BW$

输入打开时，将从源设备 S1 所指定的 16 位或 32 位数据中减去源设备 S2 所指定的 16 位或 32 位数据。结果将设置到目标设备 D1 和借位 (M8003)。

MUL (乘法)



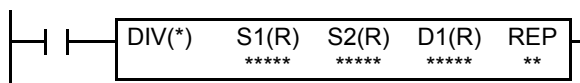
数据类型 W 或 I: $S1 \times S2 \rightarrow D1 \cdot D1+1$

数据类型 D、L 或 F: $S1 \cdot S1+1 \times S2 \cdot S2+1 \rightarrow D1 \cdot D1+1$

当输入打开时，将源设备 S1 所指定的 16 位或 32 位数据乘以源设备 S2 所指定的 16 位或 32 位数据。结果将设置到目标设备 D1。

当结果超过数据类型 D 或 L 的有效范围时，ERR LED 和特殊内部继电器 M8004 (用户程序执行错误) 将打开。

DIV (除法)



数据类型 W 或 L: $S1 \div S2 \rightarrow D1$ (商), $D1+1$ (余数)

数据类型 D 或 L:

$S1 \cdot S1+1 \div S2 \cdot S2+1 \rightarrow D1 \cdot D1+1$ (商),

$D1+2 \cdot D1+3$ (余数)

数据类型 F:

$S1 \cdot S1+1 \div S2 \cdot S2+1 \rightarrow D1 \cdot D1+1$ (商)

当输入打开时，将源设备 S1 所指定的 16 位或 32 位数据除以源设备 S2 所指定的 16 位或 32 位数据。商将设置到 16 位或 32 位目标设备 D1，余数设置到下一个 16 位或 32 位数据。数据类型 F 不产生余数。

对于 F 外的其他类型数据，当 S2 为 0 (除以 0) 时，ERR LED 和特殊内部继电器 M8004 (用户程序执行错误) 将打开。

以下除运算也会导致用户程序执行错误发生。

数据类型 I: $-32768 \div (-1)$

数据类型 I: $-2147483648 \div (-1)$

5: 四则运算指令

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要计算的数据	X	X	X	X	X	X	X	X	1-99
S2 (源 2)	要计算的数据	X	X	X	X	X	X	X	X	1-99
D1 (目标 1)	用于存储结果的目标	—	X	▲	X	X	X	X	—	1-99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 或 S2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

当选择 F (浮点) 数据类型时，只有数据寄存器和常量可指定为 S1 和 S2。

当选择 F (浮点) 数据类型，并且 S1 或 S2 不符合标准浮点格式时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于四则运算指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源，使用 16 点 (字或整数数据类型) 或 32 点 (双字、长整数数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整数)	X	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源，则使用 1 点 (字或整数数据类型) 或 2 点 (双字、长整数数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整数)	X	
F (浮点)	X	

使用进位或借位信号

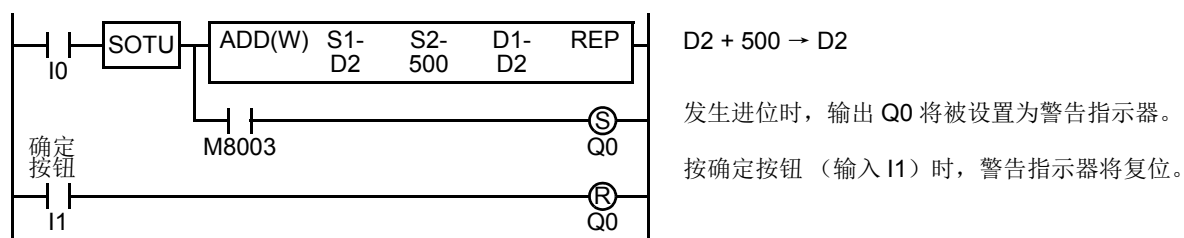
当任何四则运算结构导致 D1 (目标) 数据超出有效数据范围，将发生进位或借位，并且特殊内部继电器 M8003 将打开。

数据类型	当 D1 超出下列之间的范围时，将发生进位 / 借位
W (字)	0 和 65,535
I (整数)	-32,768 和 32,767
D (双字)	0 和 4,294,967,295
L (长整数)	-2,147,483,648 和 2,147,483,647
F (浮点)	-3.402823×10 ³⁸ 和 -1.175495×10 ⁻³⁸ 1.175495×10 ⁻³⁸ 和 3.402823×10 ³⁸

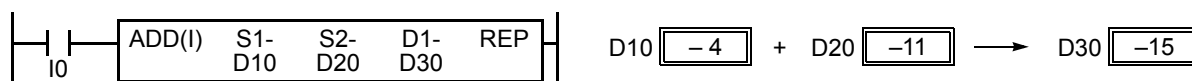
示例：ADD

·数据类型：字

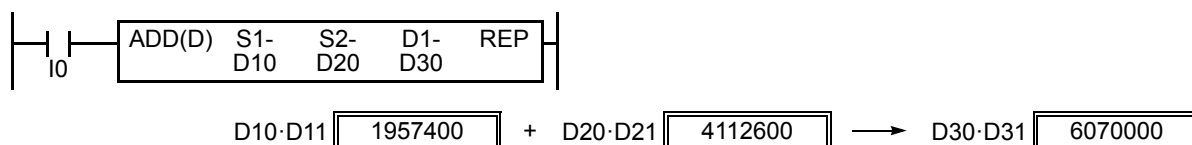
该示例演示了如何使用来自特殊内部继电器 M8003 的进位信号来设置报警信号。



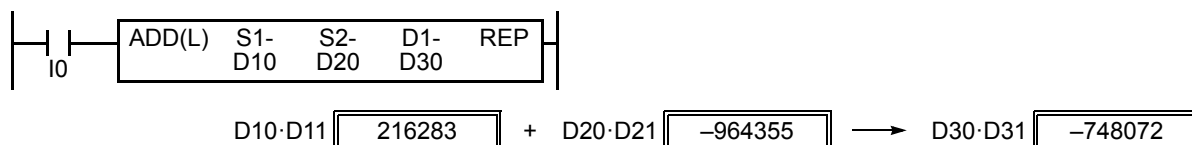
·数据类型：整数



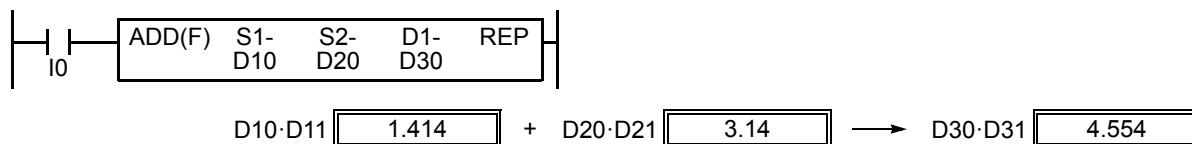
·数据类型：双字



·数据类型：长整数



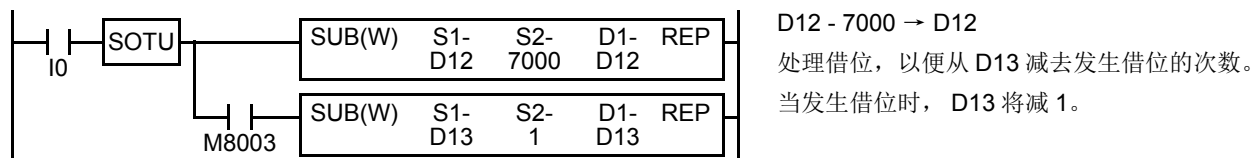
·数据类型：浮点



示例：SUB

·数据类型：字

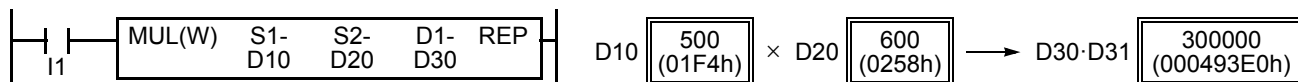
以下示例演示了如何使用特殊内部继电器 M8003 来处理借位。



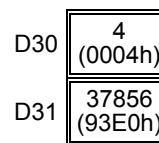
5: 四则运算指令

示例：MUL

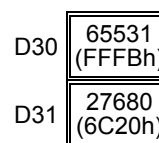
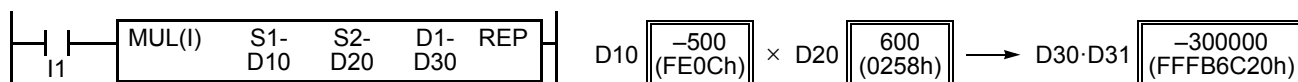
·数据类型：字



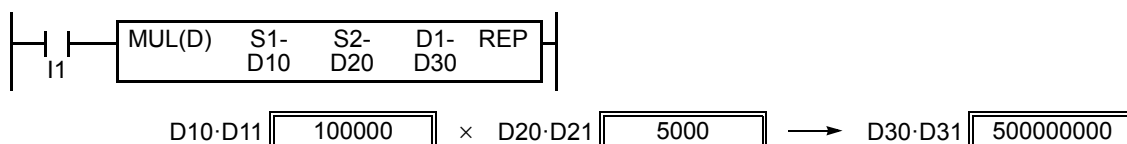
输入 I1 打开时，D10 的数据将乘以 D20 的数据，并将设置到 D30 和 D31。



·数据类型：整数

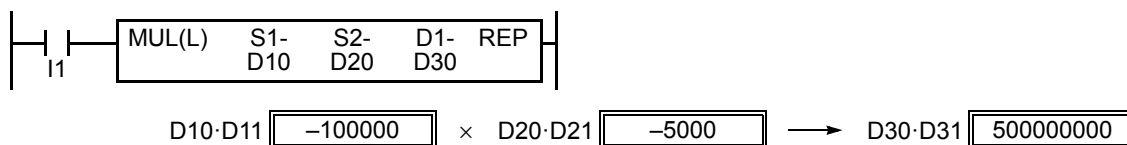


·数据类型：双字



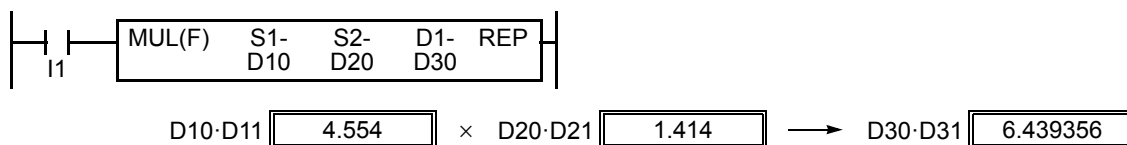
注释：在双字数据类型相乘中，结果的低位 32 位数据被设置到目标设备 D1·D1+1。

·数据类型：长整数



注释：在长整数数据类型相乘中，结果的低位 32 位数据被设置到目标设备 D1·D1+1。

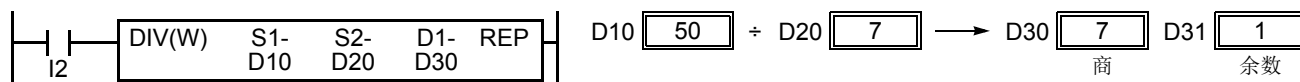
·数据类型：浮点



注释：因为在乘运算中目标使用两个字设备，所以数据寄存器 D1999 不能被用作目标设备 D1。当使用内部继电器这样的位设备作为目标时，需要 32 位内部继电器；所以内部继电器 M2521 或更大编号不能作为目标设备 D1 使用。

示例：DIV

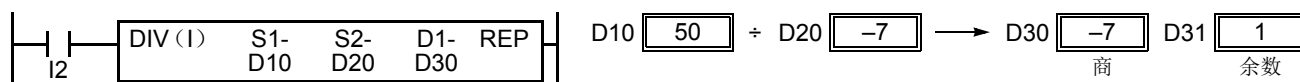
·数据类型：字



输入 I2 打开时，D10 的数据将除以 D20 的数据。商将设置到 D30，余数则设置到 D31。

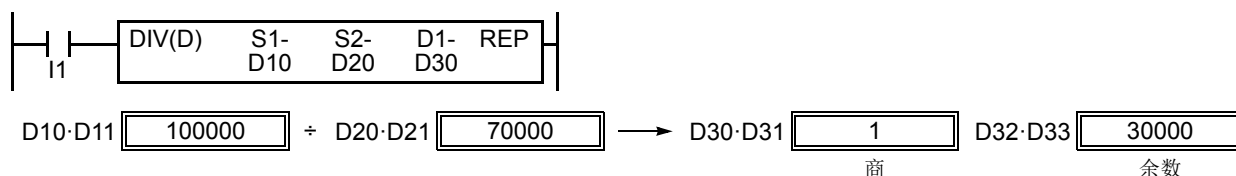
注释：因为在字数据类型除运算中目标使用两个字设备，所以数据寄存器 D1999 不能作为目标设备 D1 使用。当使用内部继电器这样的位设备作为目标时，需要 32 位内部继电器；所以 M2521 或更大编号不能作为目标设备 D1 使用。

·数据类型：整数



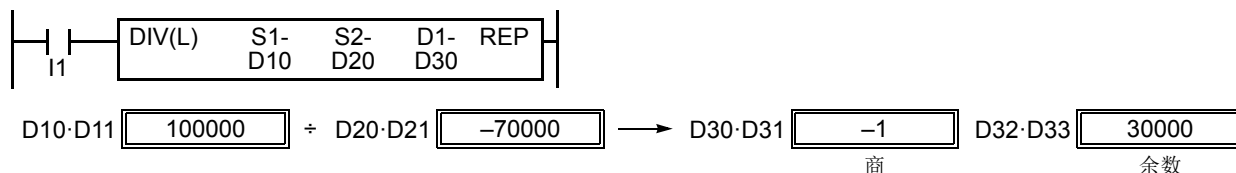
注释：因为在整数数据类型除运算中目标使用两个字设备，所以数据寄存器 D1999 不能作为目标设备 D1 使用。当使用内部继电器这样的位设备作为目标时，需要 32 位内部继电器；所以 M2521 或更大编号不能作为目标设备 D1 使用。

·数据类型：双字



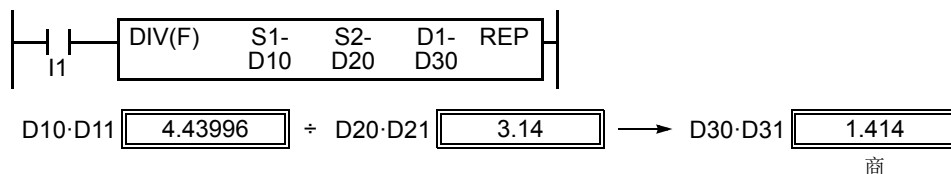
注释：因为在双字数据类型除运算中目标使用四个字设备，所以数据寄存器 D1997 - D1999 不能作为目标设备 D1 使用。当使用内部继电器这样的位设备作为目标时，需要 64 位内部继电器；所以 M2481 或更大编号不能作为目标设备 D1 使用。

·数据类型：长整数



注释：因为在长整数数据类型除运算中目标使用四个字设备，所以数据寄存器 D1997 - D1999 不能作为目标设备 D1 使用。当使用内部继电器这样的位设备作为目标时，需要 64 位内部继电器；所以 M2481 或更大编号不能作为目标设备 D1 使用。

·数据类型：浮点



注释：因为在浮点数据类型除运算中目标使用两个字设备，所以数据寄存器 D1999 不能作为目标设备 D1 使用。

5: 四则运算指令

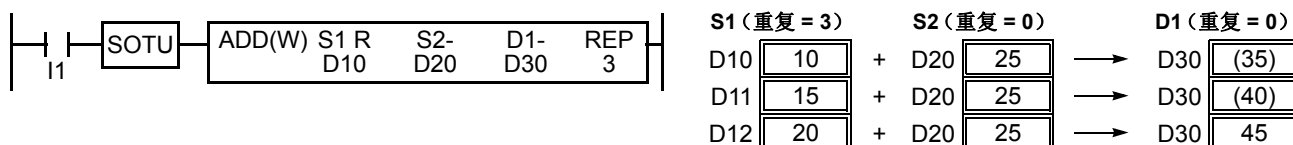
ADD 和 SUB 指令中的重复操作

可以单独或组合地将源设备 S1、S2 和目标设备 D1 指定为重复。如果不将目标设备 D1 指定为重复，则最后的结果将设置到目标设备 D1。如果指定了重复，将使用以指定的设备开始并与重复次数一样多的连续设备。因为重复运算类似于执行 ADD（加）和 SUB（减）指令，下例使用 ADD 指令进行说明。

重复一个源设备

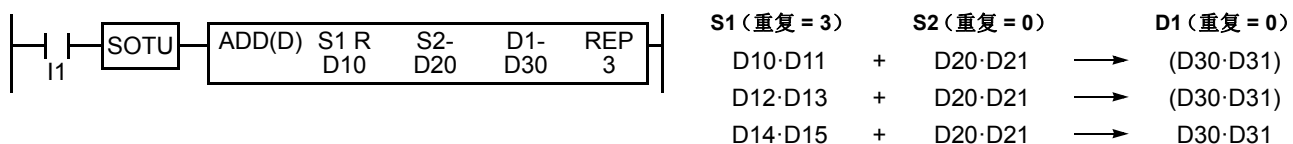
·数据类型：字和整数

仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1。



·数据类型：双字、长整数和浮点

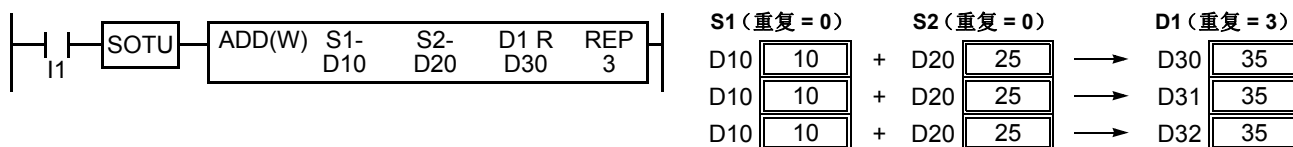
仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



仅重复目标设备

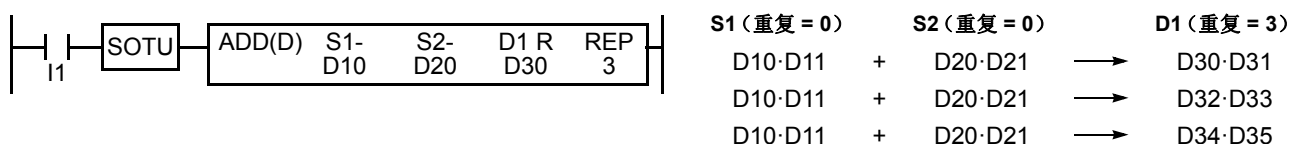
·数据类型：字和整数

仅指定 D1（目标）重复时，相同的结果将设置到以 D1 开始的 3 个设备。



·数据类型：双字、长整数和浮点

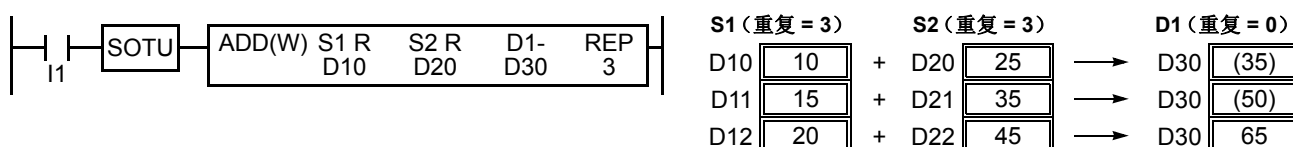
仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 3 个设备。



重复两个源设备

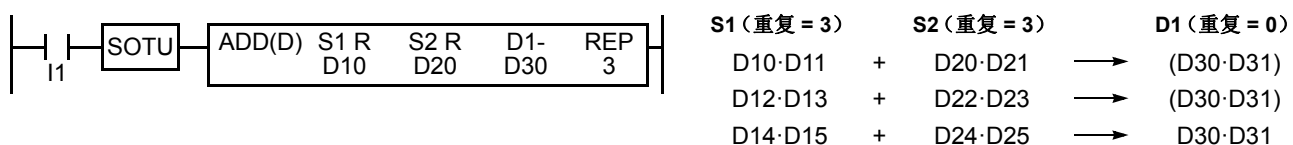
·数据类型：字和整数

指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1。



·数据类型：双字、长整数和浮点

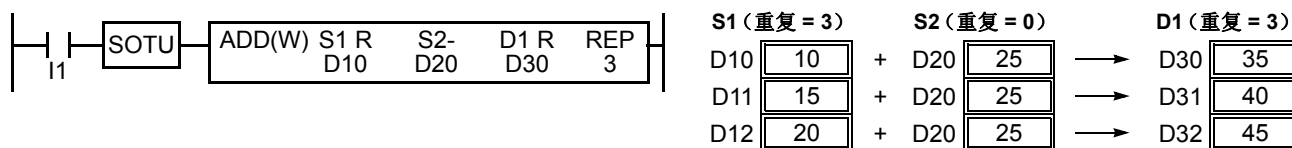
当指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



重复源设备和目标设备

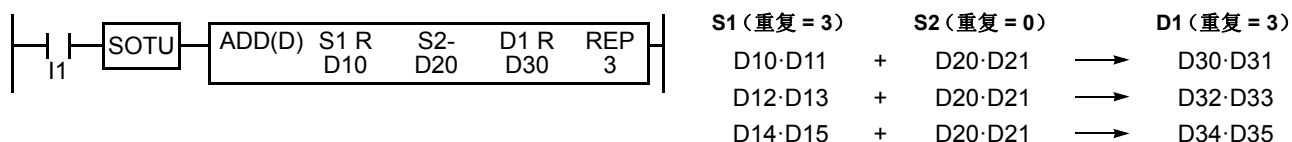
·数据类型：字和整数

指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1 开始的 3 个设备。



·数据类型：双字、长整数和浮点

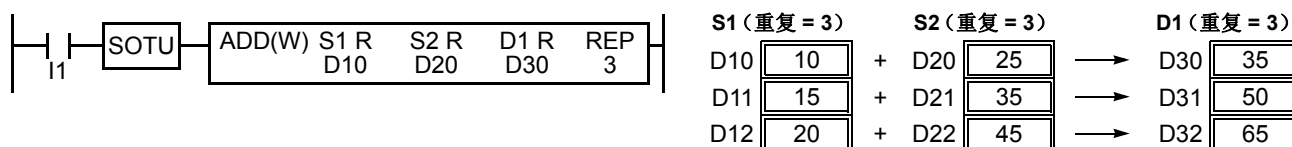
当指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



重复所有源设备和目标设备

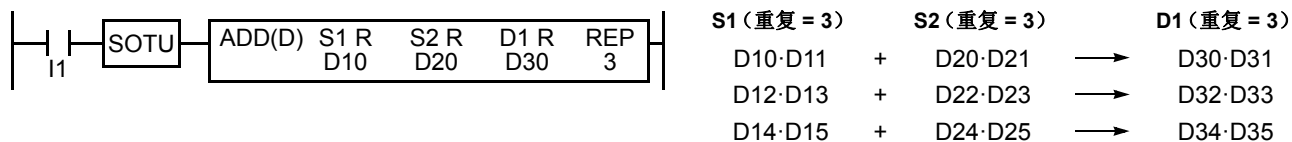
·数据类型：字和整数

指定所有设备重复时，不同的结果将设置到以 D1 开始的 3 个设备。



·数据类型：双字、长整数和浮点

当指定所有设备重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



注释：在最后一个重复操作中发生进位或借位时，特殊内部继电器 M8003（进位 / 借位）将打开。在任何重复操作中发生用户程序执行错误时，特殊内部继电器 M8004（用户程序执行错误）和 ERR LED 将打开，并在继续执行其他指令的操作时保持。

5: 四则运算指令

MUL 指令中的重复操作

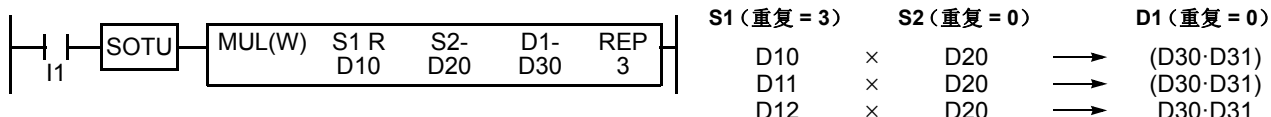
由于 MUL（乘）指令使用了两个目标设备，因此，结果将存储到下面描述的目标设备。可以单独或组合地将源设备 S1、S2 和目标设备 D1 指定为重复。未指定目标设备 D1 重复时，最后的结果将设置到目标设备 D1 和 D1+1。指定重复时，将使用与重复周期一样多并以指定设备开始的连续设备。

由于在字和整数数据类型上重复操作的模式是相似的，因此使用字数据类型来描述以下示例。

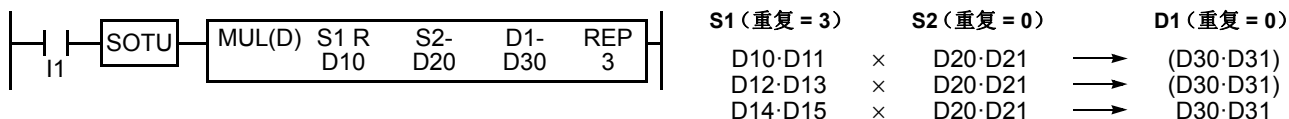
重复一个源设备

仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。

·数据类型：字与整数



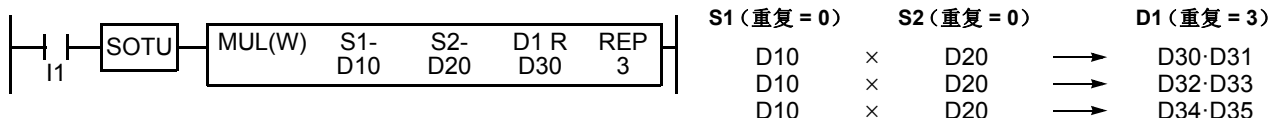
·数据类型：双字、长整数和浮点



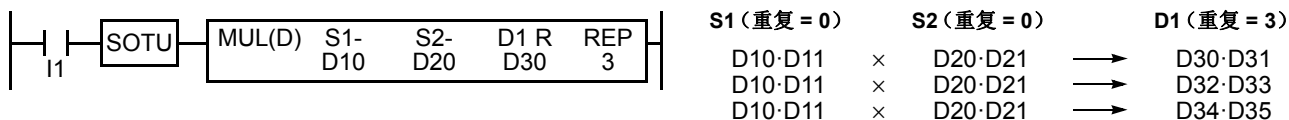
仅重复目标设备

仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 3 个设备。

·数据类型：字与整数



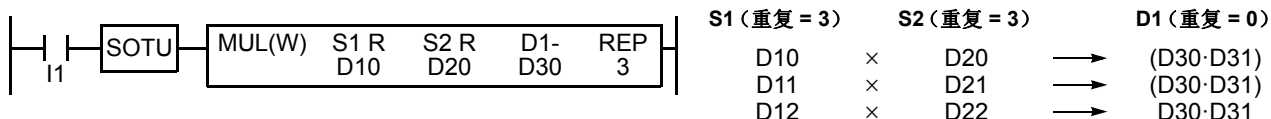
·数据类型：双字、长整数和浮点



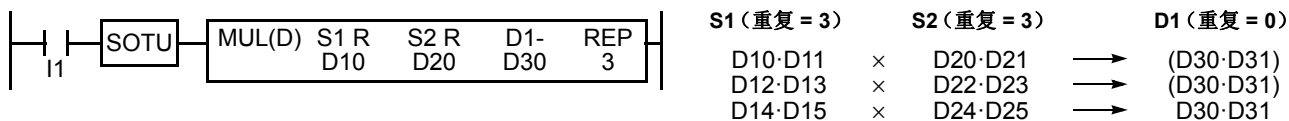
重复两个源设备

当指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1·D1+1。

·数据类型：字和整数



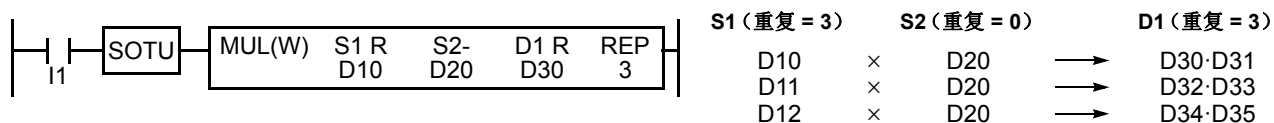
·数据类型：双字、长整数和浮点



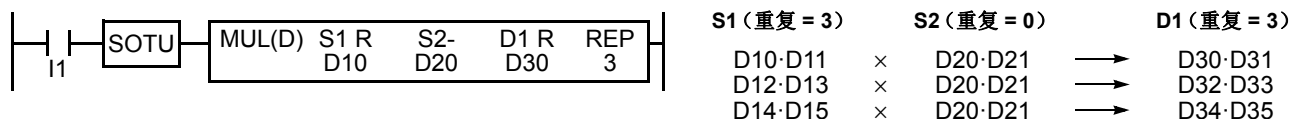
重复源设备和目标设备

当指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。

·数据类型：字与整数



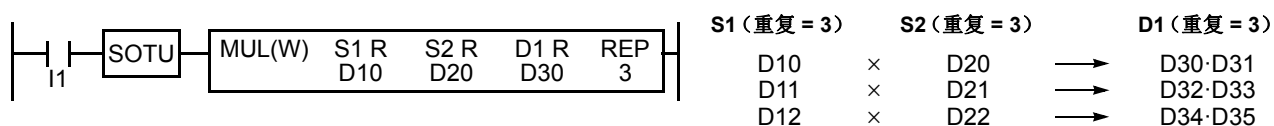
·数据类型：双字、长整数和浮点



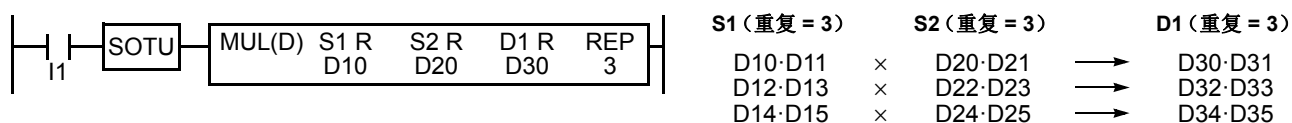
重复所有源设备和目标设备

当指定所有设备重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。

·数据类型：字与整数



·数据类型：双字、长整数和浮点



5: 四则运算指令

DIV 指令中的重复操作

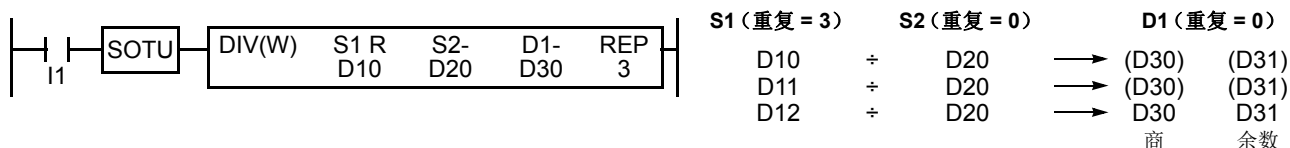
由于 DIV（除）指令（浮点数据类型除外）使用两个目标设备，因此商和余数按下面所述进行存储。可以单独或组合地将源设备 S1、S2 和目标设备 D1 指定为重复。未指定目标设备 D1 重复时，最后的结果将设置到目标设备 D1（商）和 D1+1（余数）。如果指定了重复，将使用以指定的设备开始并与重复次数一样多的连续设备。

在浮点数据类型中除指令不产生余数，并且使用两个连续的数据寄存器来存储商。当浮点数据类型的目标指定为重复时，使用与重复次数相同的连续数据寄存器。

重复一个源设备

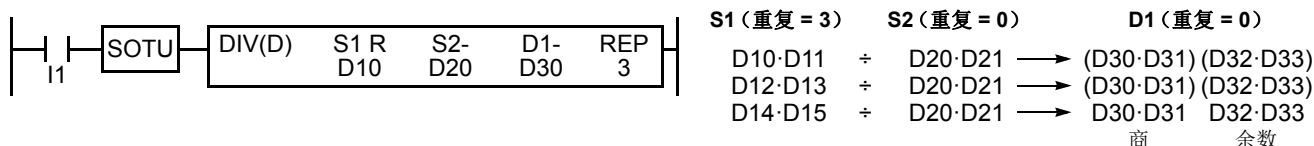
·数据类型：字和整数

仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1 和 D1+1。



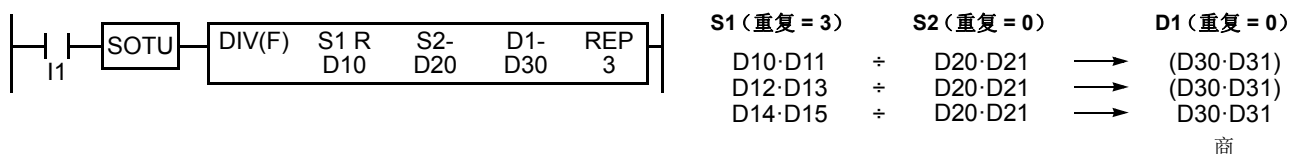
·数据类型：双字和长整数

仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1 和 D1+2·D1+3。



·数据类型：浮点

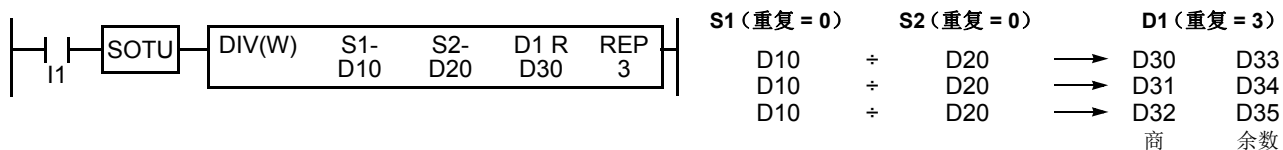
仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



仅重复目标设备

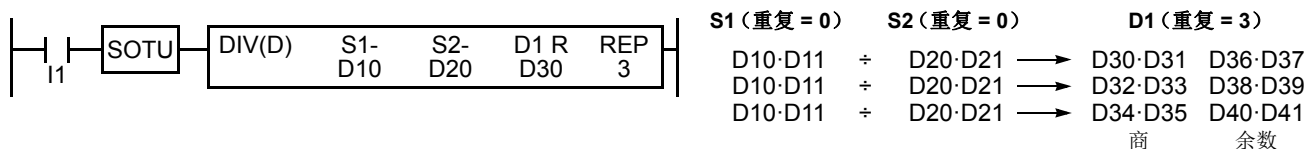
·数据类型：字和整数

仅指定 D1（目标）重复时，相同的结果将设置到以 D1 开始的 6 个设备。



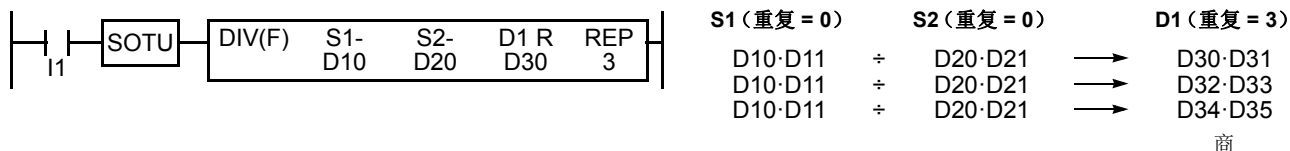
·数据类型：双字和长整数

仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 6 个设备。



·数据类型：浮点

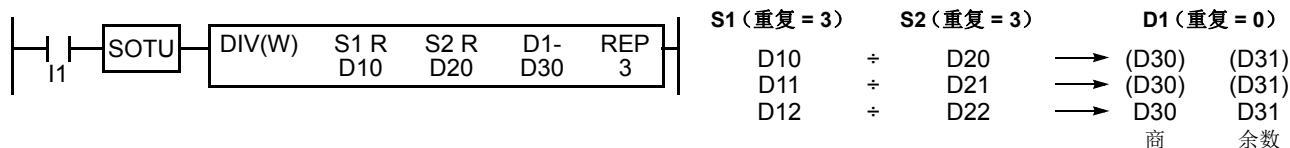
仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 3 个设备。



重复两个源设备

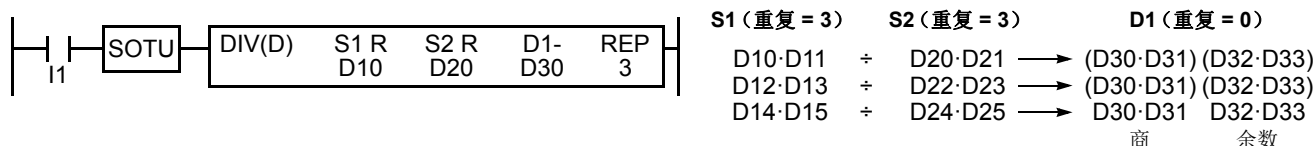
·数据类型：字和整数

指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1 和 D1+1。



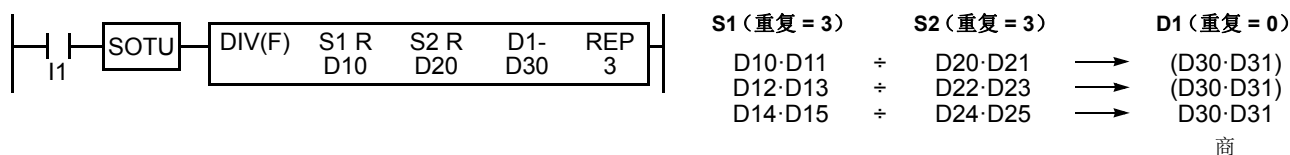
·数据类型：双字和长整数

当指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1·D1+1 和 D1+2·D1+3。



·数据类型：浮点

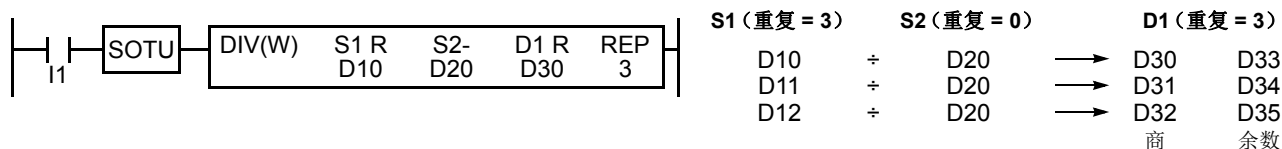
当指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



重复源设备和目标设备

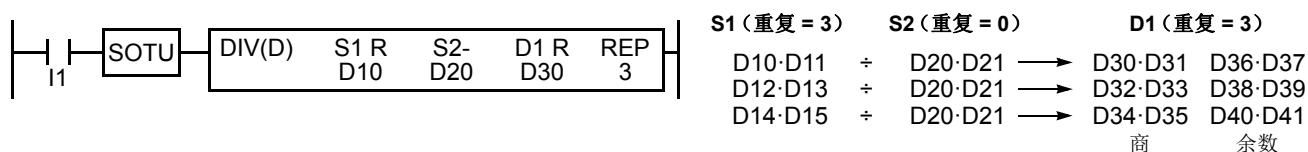
·数据类型：字和整数

指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1 开始的 6 个设备。



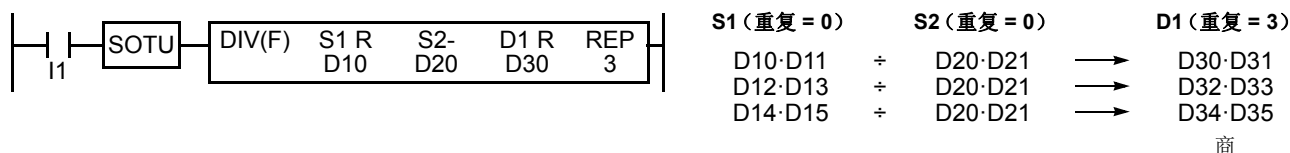
·数据类型：双字和长整数

当指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1·D1+1 开始的 6 个设备。



·数据类型：浮点

当指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。

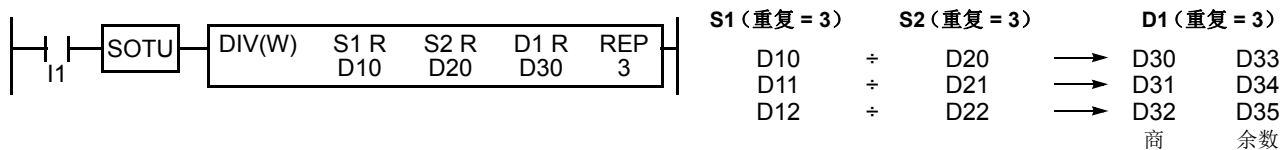


5: 四则运算指令

重复所有源设备和目标设备

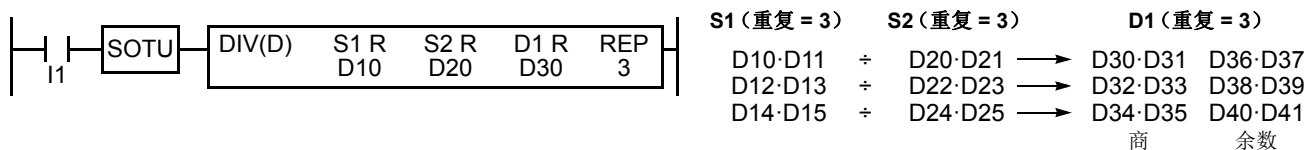
·数据类型：字和整数

指定所有设备重复时，不同的结果将设置到以 D1 开始的 6 个设备。



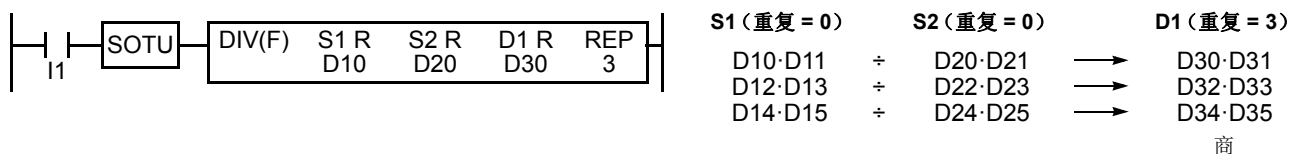
·数据类型：双字和长整数

当指定所有设备重复时，不同的结果将设置到以 D1·D1+1 开始的 6 个设备。



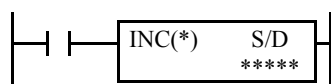
·数据类型：浮点

当指定所有设备重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



注释：在任何重复操作中发生用户程序执行错误时，特殊内部继电器 M8004（用户程序执行错误）和 ERR LED 将打开，并在继续执行其他指令的操作时保持。

INC (递增)



数据类型 W 或 I: $S/D + 1 \rightarrow S/D$
 数据类型 D 或 L: $S/D \cdot S/D+1 + 1 \rightarrow S/D \cdot S/D+1$

当输入打开时, 由设备 S/D 所指定的 16 位或 32 位数据加上 1, 并将结果存储到相同的设备中。

指令适用于升级后的系统程序 Ver. 210 或更高版本的 CPU 模块。

DEC (递减)



数据类型 W 或 I: $S/D - 1 \rightarrow S/D$
 数据类型 D 或 L: $S/D \cdot S/D+1 - 1 \rightarrow S/D \cdot S/D+1$

当输入打开时, 由设备 S/D 所指定的 16 位或 32 位数据减去 1, 并将结果存储到相同的设备中。

本指令适用于升级后的系统程序 Ver. 210 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S/D (源 / 目标)	用于递增数据的设备	-	-	-	-	-	-	X	-	-

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

由于 INC 和 DEC 指令在输入打开时的每个扫描中执行的, 因此, 应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	X	如果将 D (数据寄存器) 等字设备指定为源或目标, 将使用 1 点 (字或整数数据类型) 或 2 点 (双字或长整数数据类型)。
I (整数)	X	
D (双字)	X	
L (长整数)	X	
F (浮点)	-	

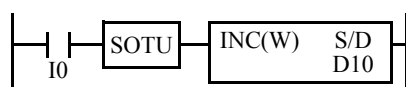
超出递增的最大值

当 S/D 的值为最大值并继续加 1 时, S/D 的值将返回到 0 并打开进位 (M8003)。

超出递减的最小值

当 S/D 的值为最小值并继续减 1 时, S/D 的值将返回到该值的最大值 (字或双字数据类型) 或减 1 (整数或长整数类型), 并打开借位 (M8003)。

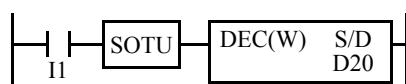
示例: INC



D10 100 + 1 \rightarrow D10 101

当输入 I0 打开时, D10 的数据将被递增 1。
 如果 SOTU 没有被编程, 在每次扫描时将递增 D10 的数据。

示例: DEC

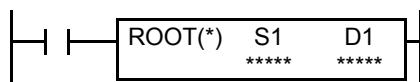


D20 100 - 1 \rightarrow D20 99

当输入 I1 打开时, D20 的数据将被递减 1。
 如果 SOTU 没有被编程, 在每次扫描时将递减 D20 的数据。

5: 四则运算指令

ROOT (平方根)



数据类型 W: $\sqrt{S1} \rightarrow D1$

输入打开时, S1 所指定的设备的平方根将被提取并存储到 D1 指定的目标。

平方根计算到两位小数, 省略小数第二位以下的数字, 并且用 100 相乘。

数据类型 D: $\sqrt{S1 \cdot S1+1} \rightarrow D1 \cdot D1+1$

当输入打开时, S1·S1+1 所指定的设备的平方根将被提取并存储到 D1·D1+1 指定的目标。

平方根计算到两位小数, 省略小数第二位以下的数字, 并且用 100 相乘。

数据类型 F: $\sqrt{S1 \cdot S1+1} \rightarrow D1 \cdot D1+1$

当输入打开时, S1·S1+1 所指定的设备的平方根将被提取并存储到 D1·D1+1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	二进制数据	—	—	—	—	—	—	X	X	—
D1 (目标 1)	用于存储结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

当选择 F (浮点) 数据类型, 并且源设备 S1 含一个负值时, 将导致程序执行错误, 打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

当选择 F (浮点) 数据类型, 并且 S1 不符合标准浮点格式时, 将导致程序执行错误, 打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

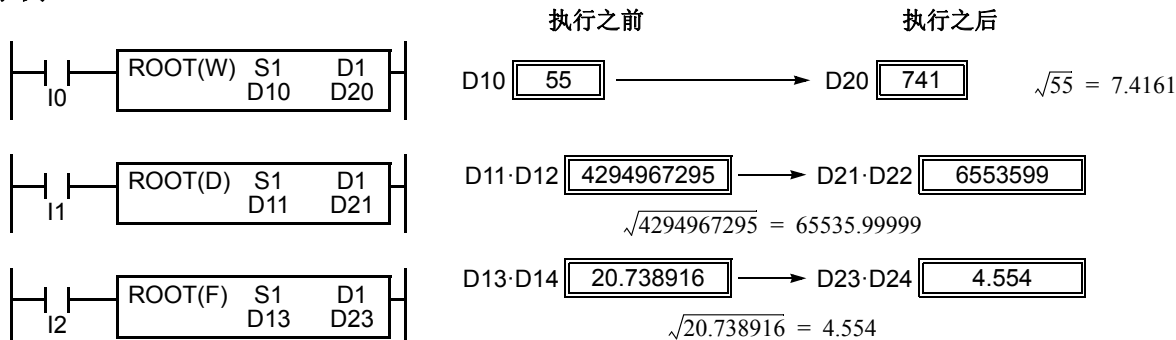
由于 ROOT 指令是在输入打开时在每个扫描中执行的, 因此, 应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

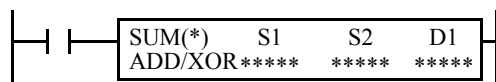
W (字)	X
I (整数)	—
D (双字)	X
L (长整数)	—
F (浮点)	X

如果将 D (数据寄存器) 这样的字设备指定为源或目标, 则使用 1 点 (字数据类型) 或 2 点 (双字数据类型或浮点数据类型)。

示例: ROOT



SUM (合计)



计算指定数据的总计，取决于计算选项。

ADD:

当输入打开时，相加由 S1 指定的设备开始的 16 位或 32 位数据的 N 块，并将其结果保存到由 D1 指定的设备。S2 指定数据块的数量。

XOR:

当输入打开时，异或由 S1 指定的设备开始的 16 位或 32 位数据的 N 块，并将其结果保存到由 D1 指定的设备。S2 指定数据块的数量。

本指令适用于升级后的系统程序 Ver. 210 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	用于计算的起始设备编号	-	-	-	-	X	X	X	-	-
S2 (源 2)	数据块的数量	-	-	-	-	-	-	X	X	-
D1 (目标 1)	用于存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

把 T(定时器)或 C(计数器)用作 S1 时，将读取定时器/计数器的当前值 (TC 或 CC)。

当选择 F(浮点)数据类型，只有数据寄存器可指定为 S1。

源 S2 无视数据类型总是使用一个字。

当选择 F(浮点)数据类型，并且 S1 不符合标准浮点格式时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

当 S2 为 0 或超出所选设备的值域时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

有效数据类型

计算类型	ADD	XOR
W(字)	X	X
I(整数)	X	-
D(双字)	X	-
L(长整数)	X	-
F(浮点)	X	-

当选择 ADD 时，可以使用所有数据类型。

当选择 XOR 时，仅可使用 W(字)数据类型。

如果将 T(定时器)、C(计数器)或 D(数据寄存器)等字设备指定为源或目标，则使用 1 点(字或整数数据类型)或 2 点(双字、长整数或浮点数据类型)

源设备和目标设备的数量

根据 ADD 或 XOR 操作的 W(字)和 I(整数)数据类型，目标将使用不同的设备数量。

运算	W(字)、I(整数)	D(双字)、L(长整数)、F(浮点)
ADD	S1, S2: 1 个字设备 D1: 2 个字设备	S1, D1: 2 个字设备 S2: 1 个字设备
XOR	S1, S2, D1: 1 个字设备	-

5: 四则运算指令

进位和借位

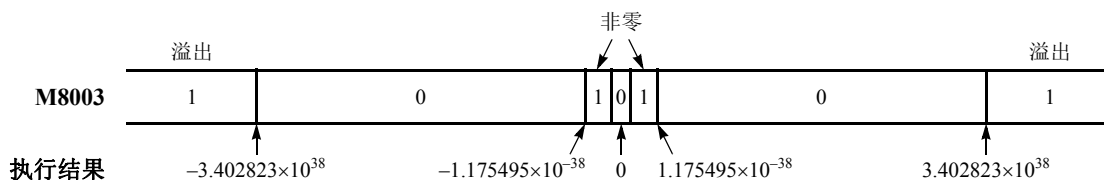
在含 D(双字)、L(长整数)或 F(浮点)数据的高级指令中,当指令的执行结果如下表所示时,将打开特殊内部继电器 M8003(进位和借位)。

数据类型	M8003	执行结果
D(双字)	1	超出值域(0 ~ 4,294,967,295)
L(长整数)	1	超出值域(-2,147,483,648 ~ 2,147,483,647)
F(浮点)	1	参照下图

浮点型数据处理中的进位和借位

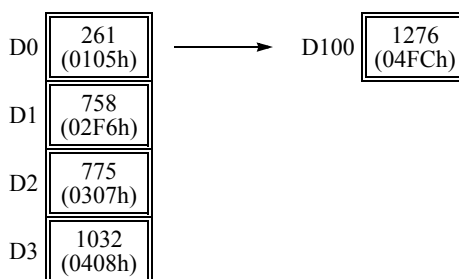
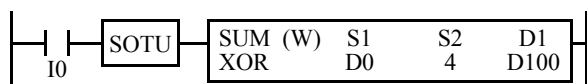
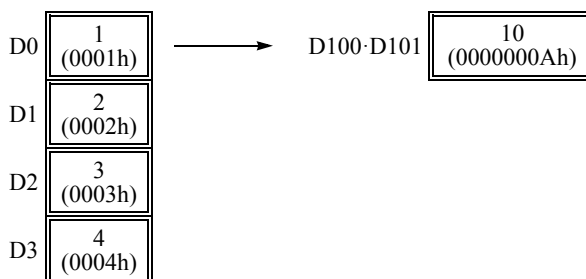
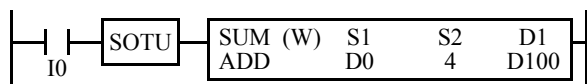
当执行含浮点型数据的高级指令时,将更新特殊内部继电器 M8003(进位和借位)。

M8003	执行结果	值
1	≠ 0	溢出(超出值域 $-3.402823 \times 10^{38} \sim 3.402823 \times 10^{38}$)
1	0	非零(在值域 $-1.175495 \times 10^{-38} \sim 1.175495 \times 10^{-38}$ 内)
0	0	零



示例: SUM

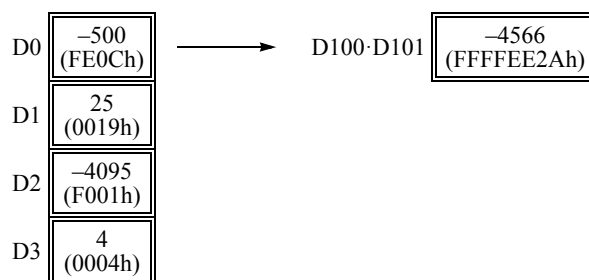
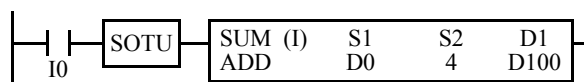
·数据类型:字



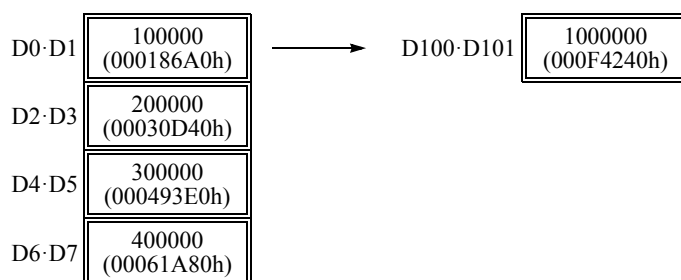
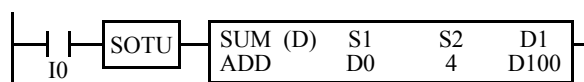
XOR 运算

D0	(0105h)	0000 0001 0000 0101
D1	(02F6h)	0000 0010 1111 0110
D2	(0307h)	0000 0011 0000 0111
XOR	D3 (0408h)	0000 0100 0000 1000
	D100 (04FCh)	0000 0100 1111 1100

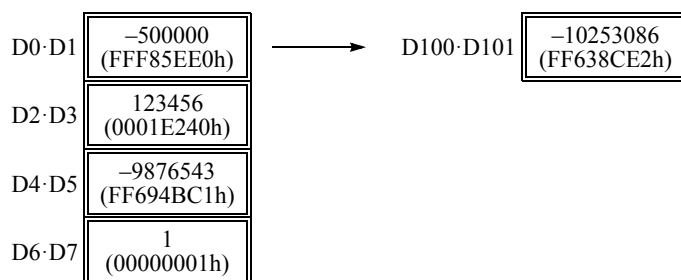
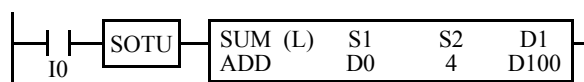
·数据类型：整数



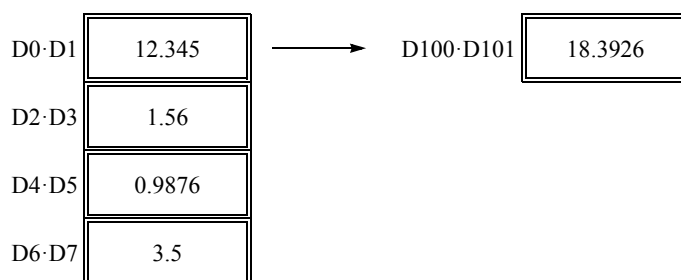
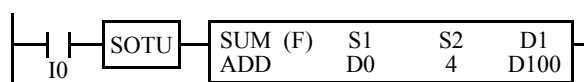
·数据类型：双字



·数据类型：长整数

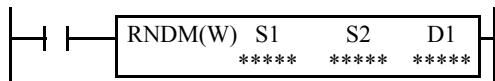


·数据类型：浮点



5: 四则运算指令

RNDM (随机)



当输入打开时，将显现为随机数值。

源设备 S1 和 S2 分别指定生成的伪随机数值的最小值和最大值。S2 的值必须大于 S1 的值。S1 和 S2 的值必须在 0 ~ 32767 的范围内。

结果存储在由设备 D1 指定的目标中。

本指令适用于升级后的系统程序 Ver. 210 或更高版本的 CPU 模块。

适用 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	伪随机数值的最小值	-	-	-	-	-	-	X	X	-
S2 (源 2)	伪随机数值的最大值	-	-	-	-	-	-	X	X	-
D1 (目标 1)	用于存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当 S1 或 S2 的值超出 32767 时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED 。

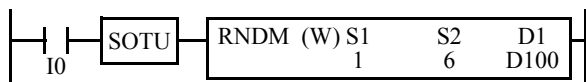
当 S1 的值大于或等于 S2 的值时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED 。

有效数据类型

W (字)	X
I (整数)	-
D (双字)	-
L (长整数)	-
F (浮点)	-

如果将 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字)。

示例 : RNDM



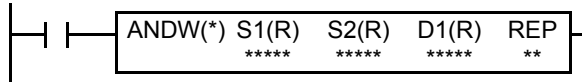
当输入 I0 打开时，RNDM 将执行生成值域为 1 ~ 6 之间的伪随机值，并将结果存储到由目标设备 D1 指定的数据寄存器 D100。

6: 逻辑运算指令

简介

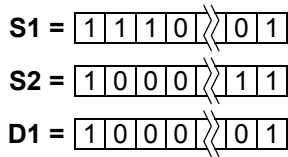
逻辑运算使用 AND、OR 和 XOR 语句，在字数据类型中则分别使用 ANDW、ORW 和 XORW 指令。

ANDW (与)



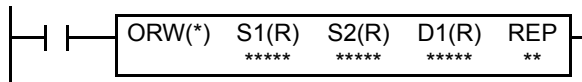
$$S1 \cdot S2 \rightarrow D1$$

当输入打开时，源设备 S1 和 S2 所指定的 16 位或 32 位数据将逐位进行 AND 运算。结果将设置到目标设备 D1。



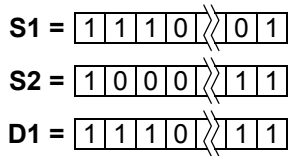
S1	S2	D1
0	0	0
0	1	0
1	0	0
1	1	1

ORW (或)



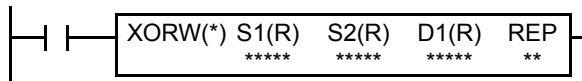
$$S1 + S2 \rightarrow D1$$

当输入打开时，源设备 S1 和 S2 所指定的 16 位或 32 位数据将逐位进行 OR 运算。结果将设置到目标设备 D1。



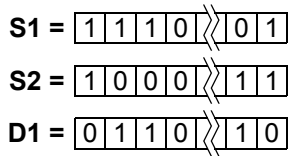
S1	S2	D1
0	0	0
0	1	1
1	0	1
1	1	1

XORW (异或)



$$S1 \oplus S2 \rightarrow D1$$

当输入打开时，源设备 S1 和 S2 所指定的 16 位或 32 位数据将逐位进行 XOR 异或运算。结果将设置到目标设备 D1。



S1	S2	D1
0	0	0
0	1	1
1	0	1
1	1	0

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

6: 逻辑运算指令

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要运算的数据	X	X	X	X	X	X	X	X	1-99
S2 (源 2)	要运算的数据	X	X	X	X	X	X	X	X	1-99
D1 (目标 1)	用于存储结果的目标	—	X	▲	X	X	X	X	—	1-99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 或 S2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

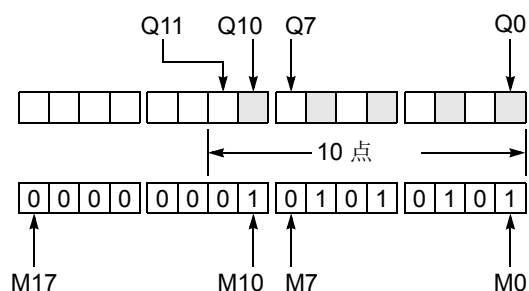
由于逻辑运算指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

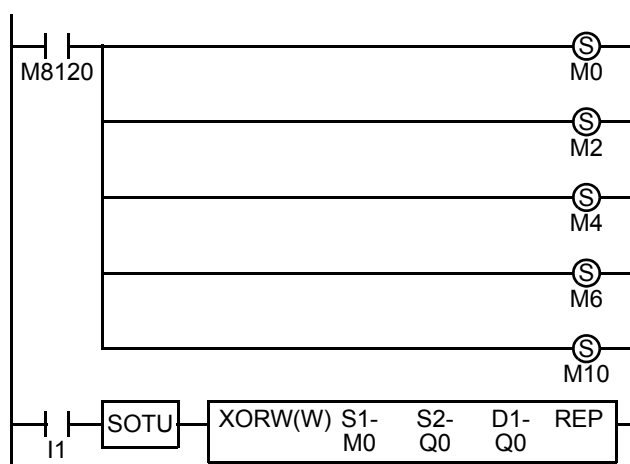
W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整数)	—	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整数)	—	
F (浮点)	—	

示例 : XORW

要在一系列 10 个输出点中转换可选输出状态，请与 10 个内部继电器点组合使用 XORW 指令。



此程序将把左侧灰色输出的状态从开取反为关，并且将那些非灰色输出的状态从关取反为开。



十个输出 (Q0 - Q11) 被分配给 10 个内部继电器 (M0 - M11)。

五个内部继电器 (M0、M2、M4、M6 和 M10) 由初始化脉冲特殊内部继电器 M8120 进行设置。

当输入 I1 打开时，将执行 XORW 指令，以便取反输出 Q0、Q2、Q4、Q6 和 Q10 的状态。

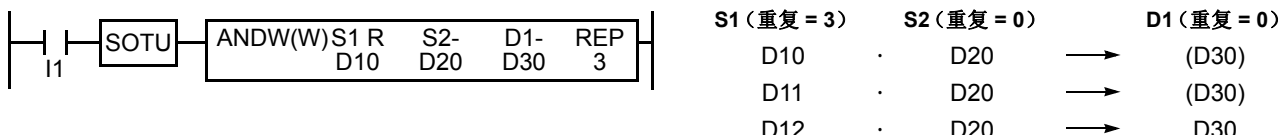
ANDW、ORW 和 XORW 指令中的重复操作

可以单独或组合地将源设备 S1、S2 和目标设备 D1 指定为重复。如果不将目标设备 D1 指定为重复，则最后的结果将设置到目标设备 D1。如果指定了重复，将使用以指定的设备开始并与重复次数一样多的连续设备。由于 ANDW（与）、ORW（或）和 XORW（异或）指令的重复运算的模式相似，因此使用 ANDW 指令来描述以下示例。

重复一个源设备

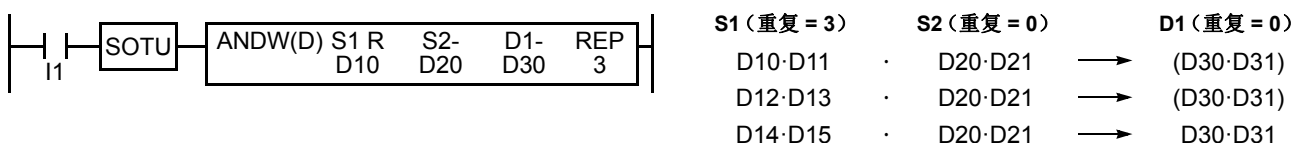
·数据类型：字

仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1。



·数据类型：双字

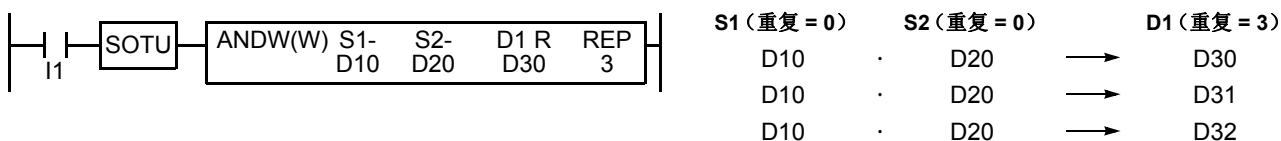
仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



仅重复目标设备

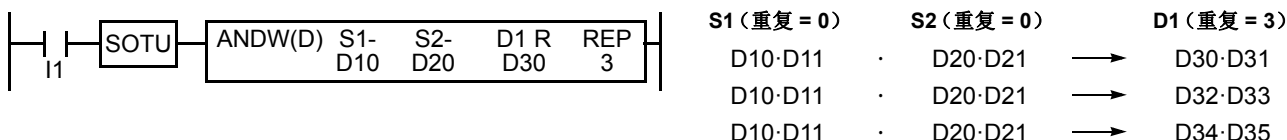
·数据类型：字

仅指定 D1（目标）重复时，相同的结果将设置到以 D1 开始的 3 个设备。



·数据类型：双字

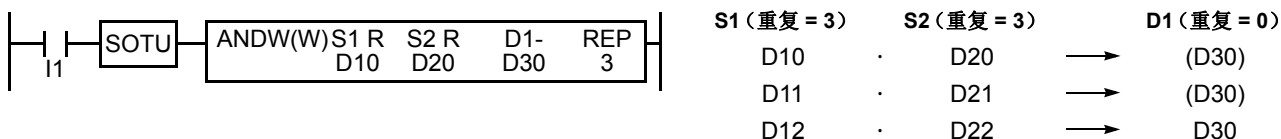
仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 3 个设备。



重复两个源设备

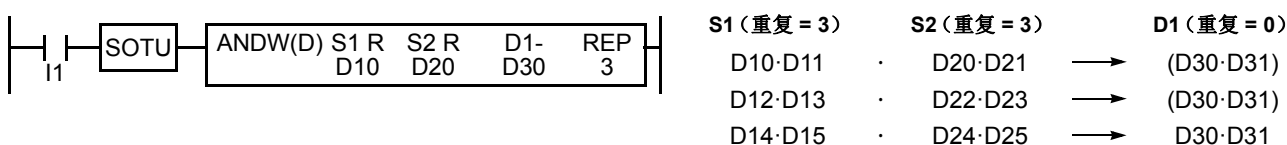
·数据类型：字

指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1。



·数据类型：双字

当指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1·D1+1。

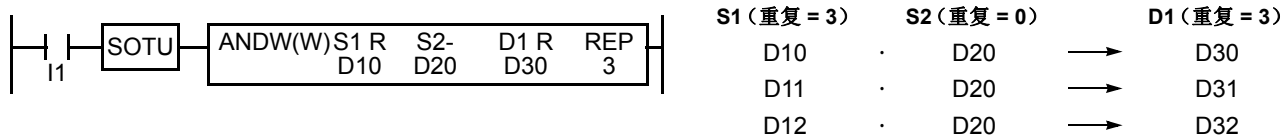


6: 逻辑运算指令

重复源设备和目标设备

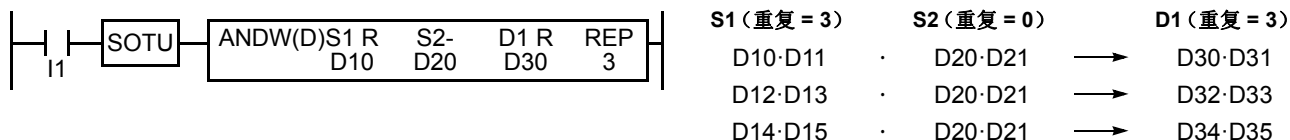
·数据类型：字

指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1 开始的 3 个设备。



·数据类型：双字

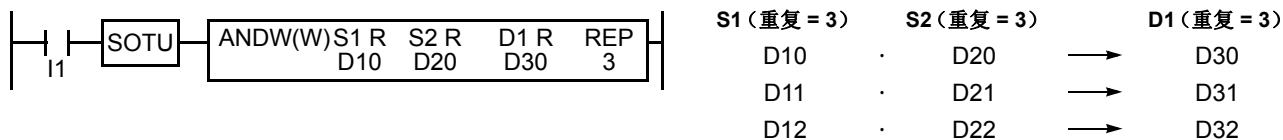
当指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



重复所有源设备和目标设备

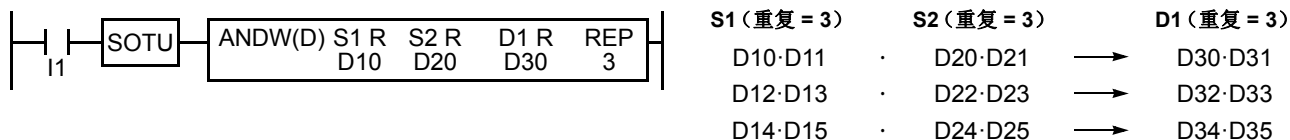
·数据类型：字

指定所有设备重复时，不同的结果将设置到以 D1 开始的 3 个设备。



·数据类型：双字

当指定所有设备重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



注释：在任何重复操作中发生用户程序错误时，特殊内部继电器 M8004（用户程序执行错误）和 ERROR LED 将打开，并在继续执行其他指令的操作时保持。对于导致用户程序执行错误的高级指令，结果不设置到任何目标。

7: 移位 / 循环指令

简介

移位指令用于将以源设备 S1 起始的数据字符串根据指定向左或向右移动 1 - 15 位。数据字符串可以是 1 - 65535 位。结果将设置到源设备 S1 和进位（特殊内部继电器 M8003）中。LSB 或 MSB 将根据指定使用 0 或 1 填充。

移位和循环指令用于将所指定的源设备 S1 中的 16 位或 32 位数据字符串向左或向右移动指定的位数。结果将设置到源设备 S1 和进位（特殊内部继电器 M8003）中。

BCD 码左移指令将两个连续的数据寄存器中的 BCD 数字向左移。

字移位指令用于将 16 位数据传送到目标数据寄存器，并将随后的数据寄存器中的数据向下移动指定的位数。

SFTL（左移）

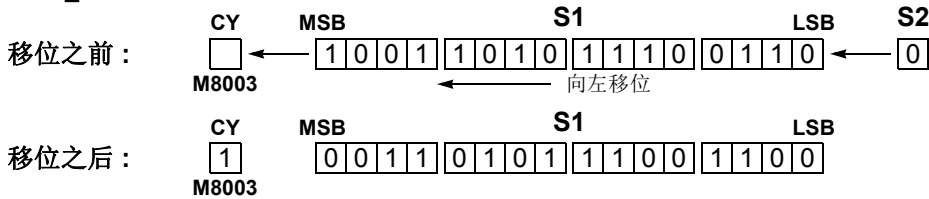


CY ← S1

当输入打开时，将以源设备 S1 起始的 N_B 数据字符串向左移动设备位所指定的位数。

结果设置到源设备 S1，并将最后一个移出的位状态设置到进位（特殊内部继电器 M8003）。源设备 S2 指定的 0 或 1 被设置到 LSB。

• S2 = 0、N_B = 16、bits = 1



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要移位的第一个数据	—	X	▲	X	—	—	X	—	—
S2 (源 2)	要移入 LSB 的数据	X	X	X	X	—	—	—	0 或 1	—
N_B	数据字符串中的位数	—	—	—	—	—	—	X	1-65535	—
bits	要移位的位数	—	—	—	—	—	—	—	1-15	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

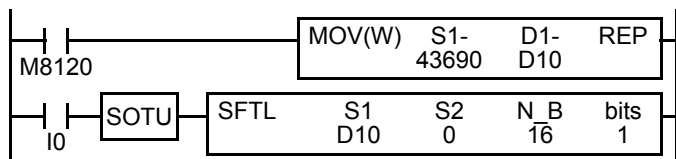
▲ 可将内部继电器 M0 - M2557 指定为 S1。无法将特殊内部继电器指定为 S1。

由于 SFTL 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

7: 移位/循环指令

示例：SFTL

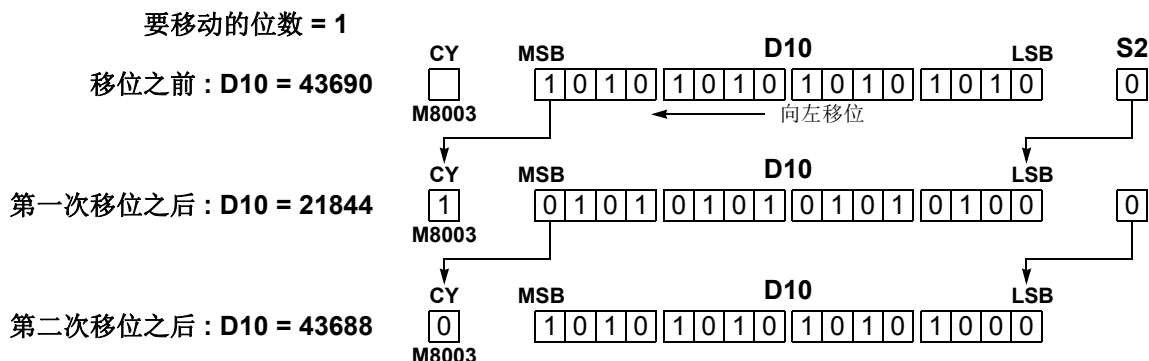
· N_B = 16 单位



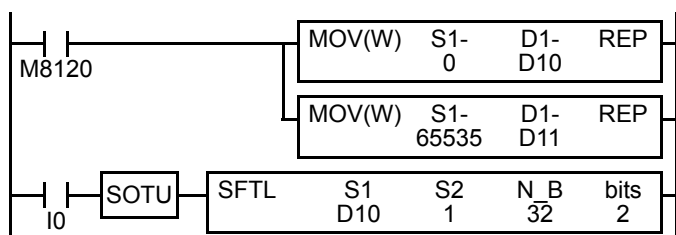
M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 43690 设置到数据寄存器 D10。

每次输入 I0 打开时，数据寄存器 D10 的 16 位数据都将向左移动设备位所指定的 1 位。最后一个移出的位状态将设置到进位（特殊内部继电器 M8003）。将零设置到 LSB。



· N_B = 32 位

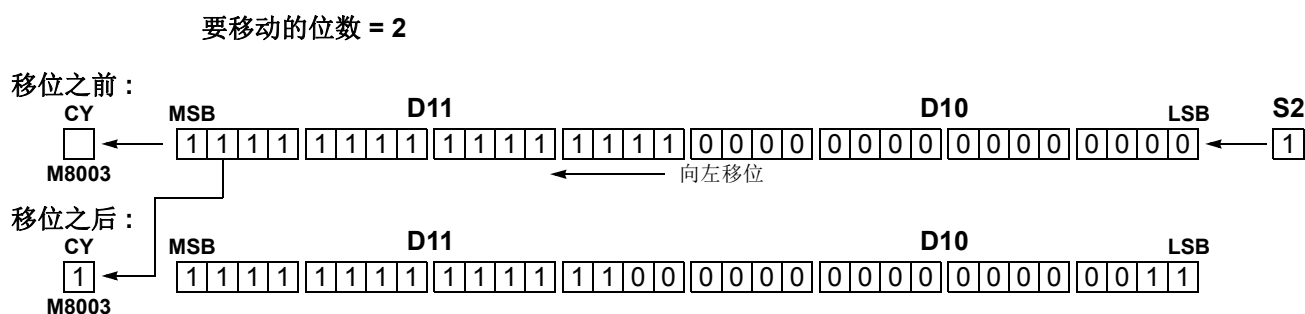


M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 0 和 65535 分别设置到数据寄存器 D10 和 D11。

每次输入 I0 打开时，数据寄存器 D10 和 D11 的 32 位数据都将向左移动设备位所指定的 2 位。D10 是低位字，D11 是高位字。

最后一个移出的位状态将设置到进位（特殊内部继电器 M8003）。将 1 设置到 LSB。



SFTR (右移)

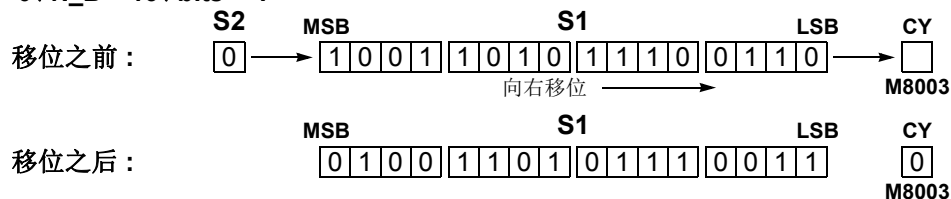


S1 → CY

当输入打开时，将以源设备 S1 起始的 N_B 数据字符串向右移动设备位所指定的位数。

结果设置到源设备 S1，并将最后一个移出的位状态设置到进位（特殊内部继电器 M8003）。源设备 S2 指定的 0 或 1 被设置到 MSB。

• S2 = 0, N_B = 16, bits = 1



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要移位的第一个数据	—	X	▲	X	—	—	X	—	—
S2 (源 2)	要移入 MSB 的数据	X	X	X	X	—	—	—	0 或 1	—
N_B	数据字符串中的位数	—	—	—	—	—	—	X	1-65535	—
bits	要移位的位数	—	—	—	—	—	—	—	1-15	—

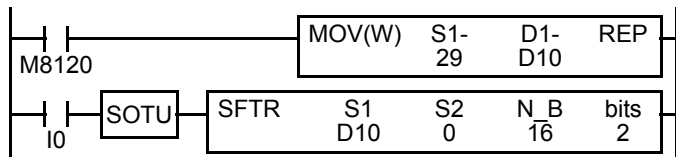
关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

由于 SFTR 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

7: 移位/循环指令

示例：SFTR

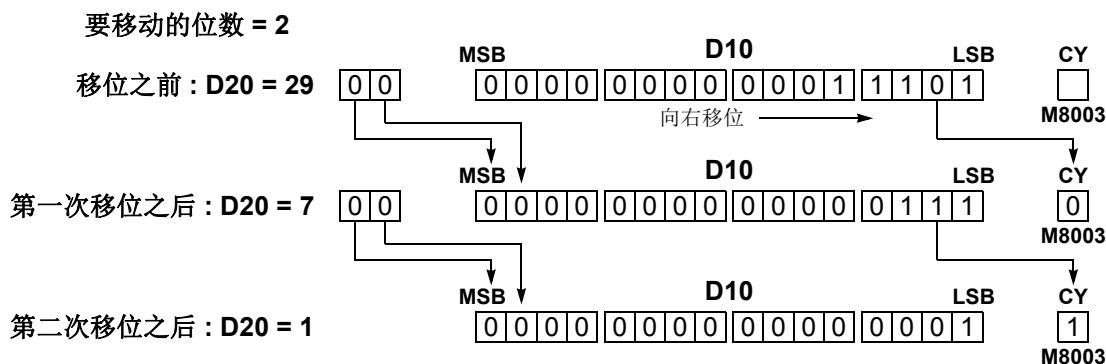
·数据类型：字



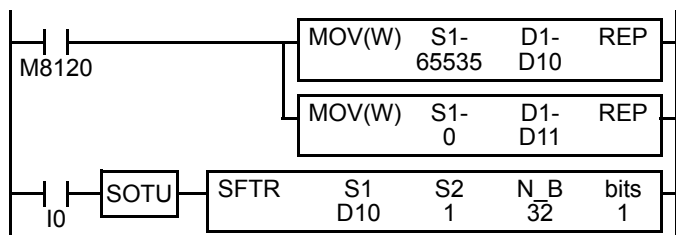
M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 29 设置到数据寄存器 D10。

每次输入 I0 打开时，数据寄存器 D10 的 16 位数据都将向右移动由设备位所指定的 2 位。最后一个移出的位状态将设置到进位（特殊内部继电器 M8003）。将零设置到 MSB。



·数据类型：双字



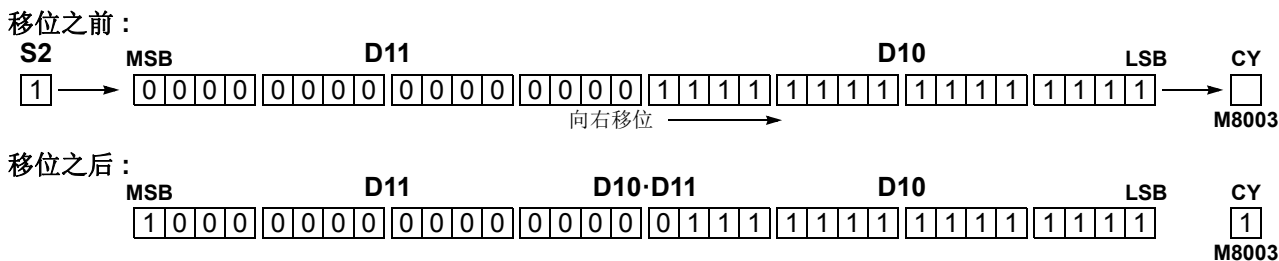
M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 65535 和 0 分别设置到数据寄存器 D10 和 D11。

每次输入 I0 打开时，数据寄存器 D10 和 D11 的 32 位数据都将向右移动设备位所指定的 1 位。D10 是低位字，D11 是高位字。

最后一个移出的位状态将设置到进位（特殊内部继电器 M8003）。将 1 设置到 MSB。

要移动的位数 = 1



BCDLS (BCD 码左移)



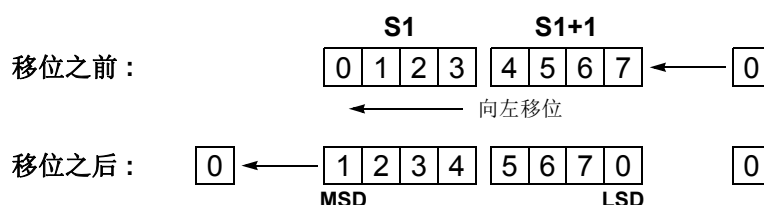
当输入打开时，S1 所指定的 32 位二进制数据将转换为 8 个 BCD 数字，并向左移动由 S2 所指定数字个数，然后转换回 32 位二进制数据。

每个 S1 和 S1+1 的有效值是 0 - 9999。

要移动的数字个数可以是 1 - 7。

与移动的数字一样多的零将设置到最低的数字。

当 S2 = 1 (要移动的位数)



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要进行 BCD 移动的数据	—	—	—	—	—	—	X	—	—
S2 (源 2)	要移动的数字个数	X	X	X	X	X	X	X	1-7	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当 T (定时器) 或 C (计数器) 用作 S2 时，将读取定时器 / 计数器的当前值 (TC 或 CC)。

由 S2 指定的要移动的数字个数可以是 1 - 7。

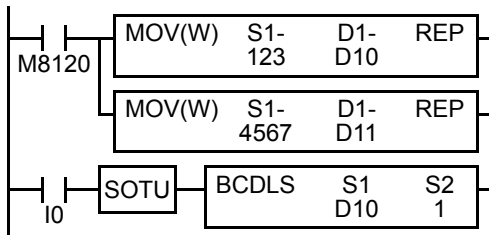
请确保对于每个数据寄存器来说由 S1 和 S1+1 所确定的源数据在 0 - 9999 之间。如果源数据超过 9999，将导致用户程序执行错误，这时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。S2 超过 7 时，也将导致用户程序执行错误。

有效数据类型

W (字)	—	如果 D (数据寄存器) 这样的字设备指定为源 S1，则使用 2 点 (双字数据类型)。
I (整数)	—	当把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源 S2 时，将使用 16 点。
D (双字)	X	
L (长整数)	—	当把 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源 S2 时，将使用 1 点。
F (浮点)	—	

7: 移位/循环指令

示例：BCDLS



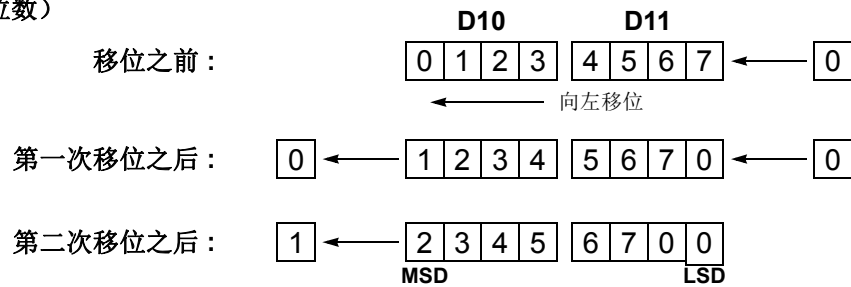
M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 123 和 4567 分别设置到数据寄存器 D10 和 D11。

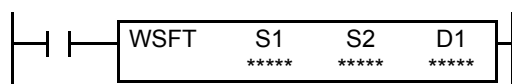
每次输入 I0 打开时，由 S1 所指定的数据寄存器 D10 和 D11 的 32 位二进制数据都将转换为 8 个 BCD 数字，并向左移动由设备 S2 所指定的 1 个数字，然后转换回 32 位二进制数据。

与移动的数字一样多的零将设置到最低的数字。

当 S2 = 1（要移动的位数）

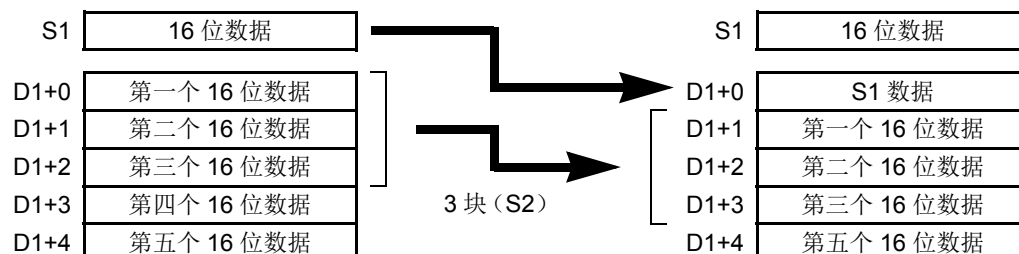


WSFT (字移位)



当输入打开时，以 D1 所指定的设备开始的 N 块 16 位字数据将上移到下一组 16 位位置。同时，设备 S1 所指定的数据将传送到 D1 所指定的设备。S2 指定要传送的块数量。

当 S2 = 3 (要移动的块数量) 时



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	字移动的源数据	X	X	X	X	X	X	X	X	—
S2 (源 2)	要移动的块数量	X	X	X	X	X	X	X	X	—
D1 (目标 1)	要移位的第一个设备编号	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

把 T (定时器) 或 C (计数器) 用作 S1 或 S2 时，将读取定时器 / 计数器的当前值 (TC 或 CC)。

有效数据类型

W (字)	X	当 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定为源 S1 或 S2 时，将使用 16 点。
I (整数)	—	
D (双字)	—	当 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备被指定为源 S1 或 S2 时，将使用 1 点。
L (长整数)	—	
F (浮点)	—	

特殊内部继电器 M8024: BMOV/WSFT 执行标记

执行 BMOV 或 WSFT 时，M8024 将打开。完成后，M8024 将关闭。在执行 BMOV 或 WSFT 时，如果 CPU 断电，那么当 CPU 再次通电时，M8024 将保持打开状态。

示例: WSFT

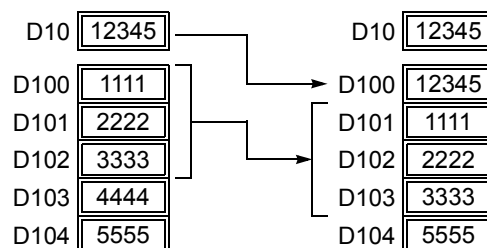


D100 ~ D102 → D101 ~ D103

D10 → D100

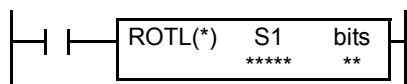
当输入 I0 打开时，以目标设备 D1 所指定的 D100 开始的 3 个数据寄存器的数据将移动到下一组数据寄存器。源设备 S1 所指定的数据寄存器 D10 的数据将传送到目标设备 D1 所指定的 D100。

移位之前: 第一次移位之后:



7: 移位/循环指令

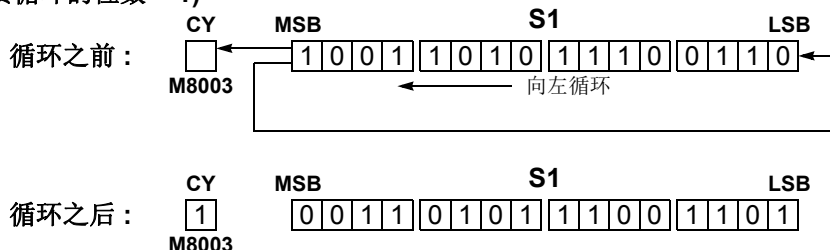
ROTL (循环左移)



当输入打开时，源设备 S1 所指定的 16 位或 32 位数据将向左循环由设备位所指定的位数。

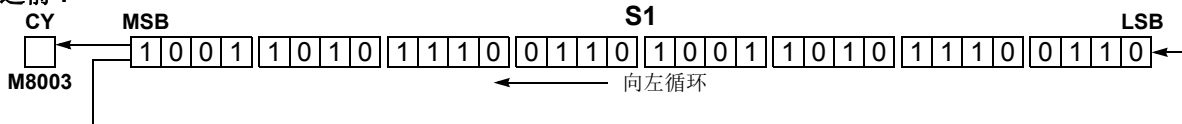
结果将设置到源设备 S1，并且循环出来的最后位状态将设置到进位（特殊内部继电器 M8003）。

- 数据类型：字 (要循环的位数 = 1)

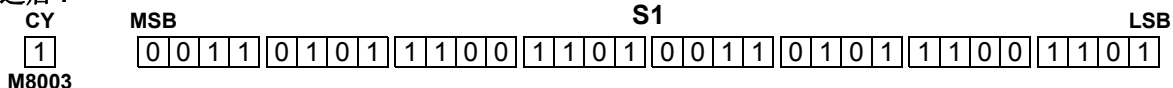


- 数据类型：双字 (要循环的位数 = 1)

循环之前：



循环之后：



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要进行位循环的数据	—	X	▲	X	—	—	X	—	—
bits	要循环的位数	—	—	—	—	—	—	—	1-15, 1-31	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 S1。无法将特殊内部继电器指定为 S1。

字数据类型要循环的位数可以是 1 - 15，或者双字数据类型要循环的位数可以是 1 - 31。

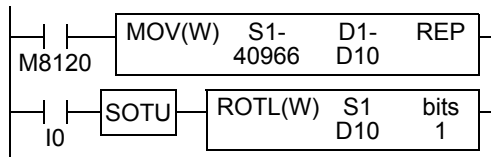
由于 ROTL 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	X	如果将 Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源，则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	—	
D (双字)	X	如果将 D (数据寄存器) 这样的字设备指定为源，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	—	
F (浮点)	—	

示例：ROTL

·数据类型：字



M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 40966 设置到数据寄存器 D10。

每次输入 I0 打开时，数据寄存器 D10 的 16 位数据都将向左循环设备位所指定的 1 位。

MSB 的状态将设置到进位（特殊内部继电器 M8003）。

要循环的位数 = 1

循环之前：D10 = 40966

第一次循环之后：D10 = 16397

第二次循环之后：D10 = 32794



·数据类型：双字



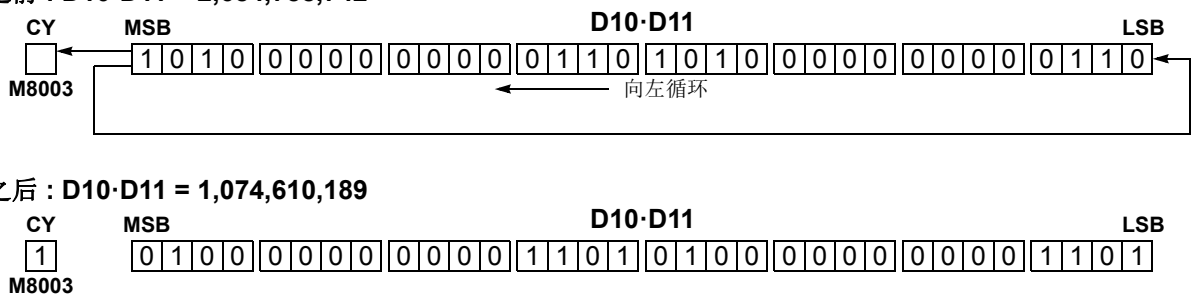
每次输入 I1 打开时，数据寄存器 D10 和 D11 的 32 位数据都将向左循环设备位所指定的 1 位。

MSB 的状态将设置到进位（特殊内部继电器 M8003）。

要循环的位数 = 1

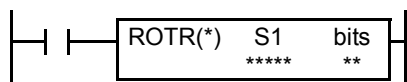
循环之前：D10·D11 = 2,684,788,742

循环之后：D10·D11 = 1,074,610,189



7: 移位/循环指令

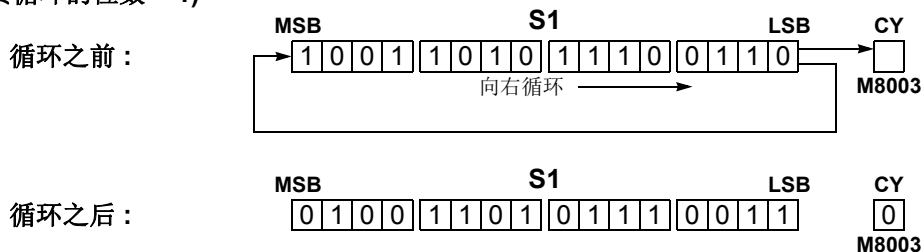
ROTR (循环右移)



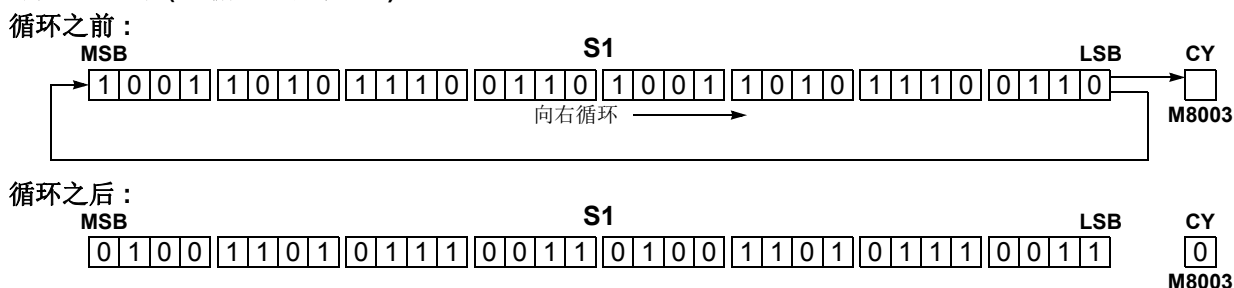
当输入打开时，源设备 S1 所指定的 16 位或 32 位数据将向右循环由设备位所指定的位数。

结果将设置到源设备 S1，并且循环出来的最后位状态将设置到进位（特殊内部继电器 M8003）。

- 数据类型：字 (要循环的位数 = 1)



- 数据类型：双字 (要循环的位数 = 1)



适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要进行位循环的数据	—	X	▲	X	—	—	X	—	—
bits	要循环的位数	—	—	—	—	—	—	—	1-15, 1-31	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 S1。无法将特殊内部继电器指定为 S1。

字数据类型要循环的位数可以是 1 - 15，或者双字数据类型要循环的位数可以是 1 - 31。

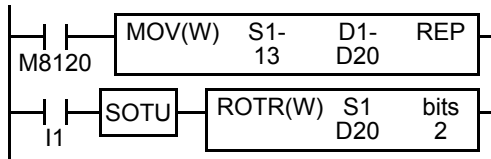
由于 ROTR 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	X	如果将 Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源，则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	—	
D (双字)	X	如果将 D (数据寄存器) 这样的字设备指定为源，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	—	
F (浮点)	—	

示例：ROTR

·数据类型：字



M8120 是初始化脉冲特殊内部继电器。

当 CPU 开始运行时，MOV（传送）指令将把 13 设置到数据寄存器 D20。

每次输入 I1 打开时，数据寄存器 D20 的 16 位数据都将向右循环由设备位所指定的 2 位。

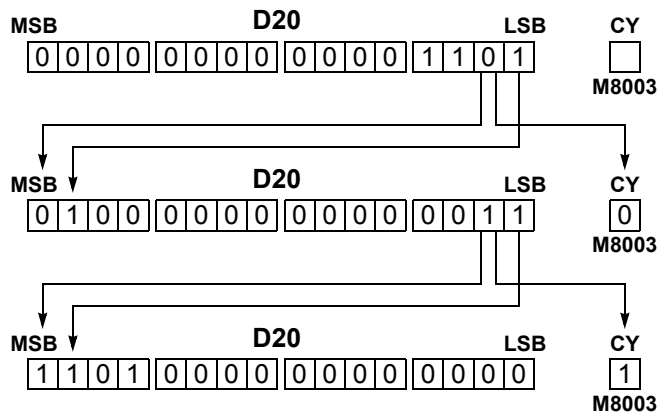
循环出来的最后位状态将设置到进位（特殊内部继电器 M8003）。

要循环的位数 = 2

循环之前：D20 = 13

第一次循环之后：D20 = 16387

第二次循环之后：D20 = 53248



·数据类型：双字



每次输入 I1 打开时，数据寄存器 D20 和 D21 的 32 位数据都将向右循环设备位所指定的 1 位。

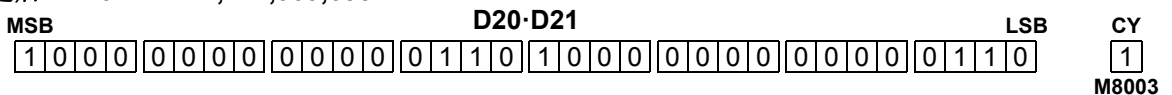
循环出来的最后位状态将设置到进位（特殊内部继电器 M8003）。

要循环的位数 = 1

循环之前：D20-D21 = 851,981



循环之后：D20-D21 = 2,147,909,638



8: 数据转换指令

简介

数据转换指令用于在二进制、BCD 和 ASCII 之间进行数据格式转换。

BTOA (BCD 码 → ASCII 码) 和 ATOB (ASCII 码 → BCD 码) 指令中添加了双字数据类型。因而 BTOA 和 ATOB 指令可以转换双字数据。BTOA 和 ATOB 指令适用于升级后的系统程序版本 200 或更高的 CPU 模块。

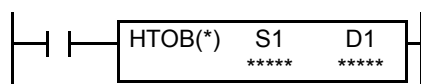
ENCO (编码)、DECO (解码) 和 BCNT (位计数) 指令用于处理位设备数据。

ALT (交替输出) 指令用于在每次按下输入按钮时打开和关闭输出。

CVDT (转换数据类型) 指令用于在字 (W)、整数 (I)、双字 (D)、长整数 (L) 或浮点 (F) 之间进行数据类型转换。

升级后的系统程序版本 200 或更高的 CPU 模块上添加了新的指令：DTDV (数据分割)，DTCB (数据组合) 和 SWAP (数据交换) 指令。DTDV 和 DTCB 指令在两个 1 字节数据和一个字数据之间进行转换。SWAP 指令将高位字节的字或双字的字数据分别与低位字节的同类型字数据交换。

HTOB (HEX → BCD 码)



S1 → D1

当输入打开时，S1 所指定的 16 位或 32 位数据将转换为 BCD，并存储到设备 D1 所指定的目标中。

字数据类型的源设备的有效值为 0 - 9999，双字数据类型的源设备的有效值为 0 - 9999 9999。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的二进制数据	X	X	X	X	X	X	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

字数据类型的源设备的有效值为 0 - 9999 (270Fh)，双字数据类型的源设备的有效值为 0 - 9999 9999 (5F5 E0FFh)。请确保 S1 所指定的源数据在有效值范围之内。如果源数据不在有效范围内，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 HTOB 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

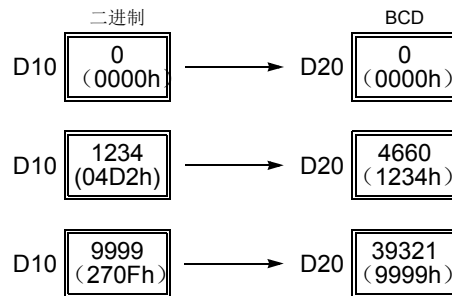
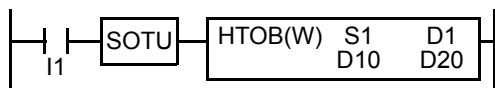
有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源，则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	—	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	—	
F (浮点)	—	

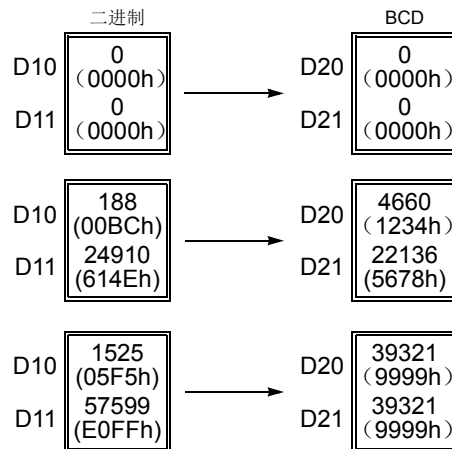
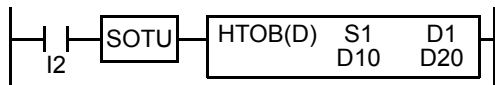
8: 数据转换指令

示例：HTOB

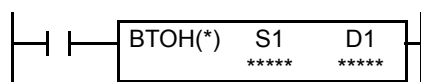
- 数据类型：字



- 数据类型：双字



BTOH (BCD 码→ HEX)



S1 → D1

输入打开时，S1 所指定的 BCD 数据将转换为 16 位或 32 位二进制数据，并存储到设备 D1 所指定的目标中。

字数据类型的源设备的有效值为 0 - 9999 (BCD)，双字数据类型的源设备的有效值为 0 - 9999 9999 (BCD)。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的 BCD 数据	X	X	X	X	X	X	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

字数据类型的源设备的有效值为 0 - 9999 (BCD)，双字数据类型的源设备的有效值为 0 - 9999 9999 (BCD)。请确保 S1 所指定的源数据的每个数字都是 0 - 9。如果源数据不在有效范围内，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 BTOH 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

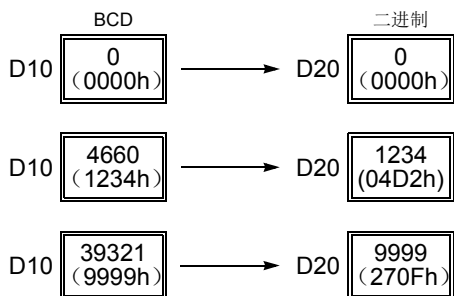
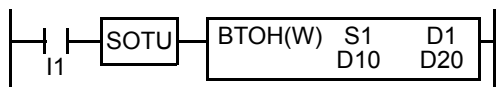
有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源，则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	—	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	—	
F (浮点)	—	

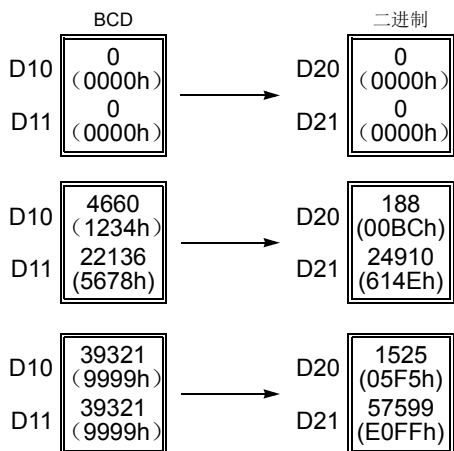
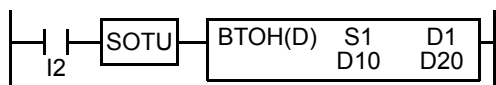
8: 数据转换指令

示例：BTOH

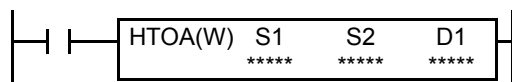
- 数据类型：字



- 数据类型：双字



HTOA (HEX → ASCII 码)



S1 → D1, D1+1, D1+2, D1+3

当输入打开时，将从最低位数字开始读取与 S2 所指定的数字的位数一样多的、由 S1 所指定的 16 位二进制数据，然后将它转换为 ASCII 数据，并存储到以 D1 所指定的设备开始的目标中。

要转换的数字的位数可以是 1 - 4。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的二进制数据	X	X	X	X	X	X	X	X	—
S2 (源 2)	要转换的数字的位数	X	X	X	X	X	X	X	1-4	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

在 T (定时器) 或 C (计数器) 用作 S1 或 S2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。

要转换的数字的位数可以是 1 - 4。请确保 S2 所指定的数字的数量在有效范围之内。如果 S2 数据不在有效范围内，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 HTOA 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

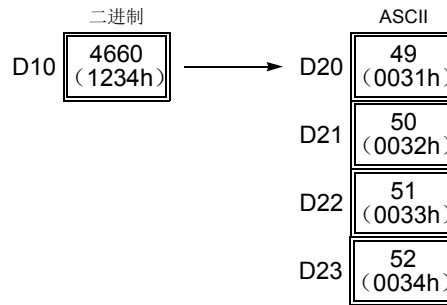
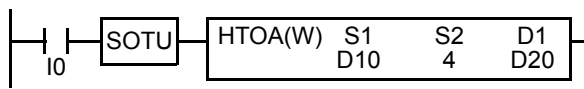
有效数据类型

W (字)	X	把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源时，将使用 16 点 (字数据类型)。
I (整数)	—	
D (双字)	—	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字数据类型)。
L (长整数)	—	
F (浮点)	—	

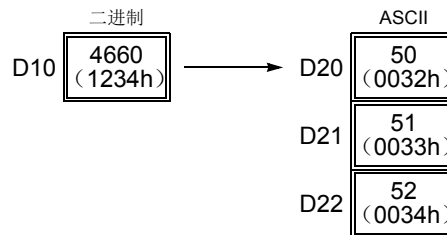
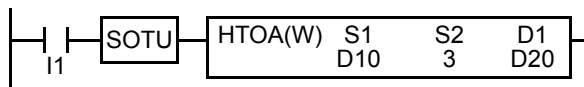
8: 数据转换指令

示例：HTOA

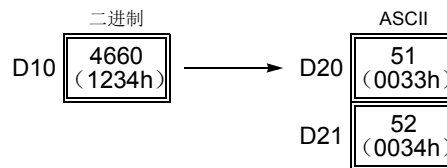
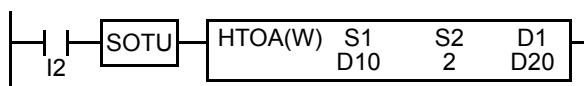
- 数字的位数：4



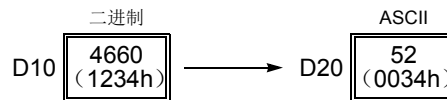
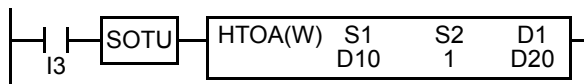
- 数字的位数：3



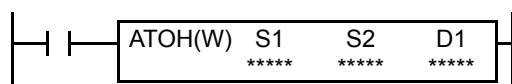
- 数字的位数：2



- 数字的位数：1



ATOH (ASCII 码→ HEX)



S1, S1+1, S1+2, S1+3 → D1

输入打开时，由 S1 所指定的、与 S2 所指定的数字的位数一样多的 ASCII 数据将转换为 16 位二进制数据，并存储到设备 D1 所指定的目标中。

要转换的源数据的有效值是 30h - 39h 和 41h - 46h。

要转换的数字的位数可以是 1 - 4。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的 ASCII 数据	—	—	—	—	—	—	X	—	—
S2 (源 2)	要转换的数字的位数	X	X	X	X	X	X	X	1-4	—
D1 (目标 1)	要存储转换结果的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

要转换的源 S1 数据的有效值是 30h - 39h 和 41h - 46h。请确保 S1 所指定的每个源数据的值和 S2 所指定的数字的位数在有效范围之内。如果 S2 或 S2 数据不在有效范围内，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 ATOH 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

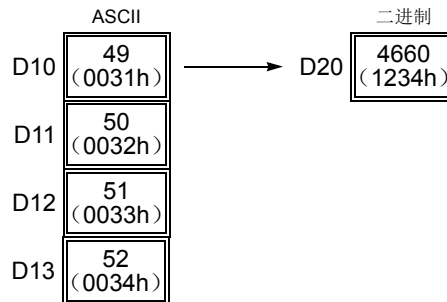
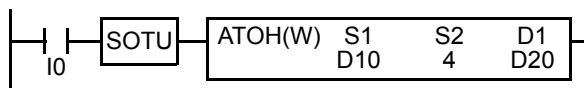
有效数据类型

W (字)	X	把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标时，将使用 16 点 (字数据类型)。
I (整数)	—	
D (双字)	—	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字数据类型)。
L (长整数)	—	
F (浮点)	—	

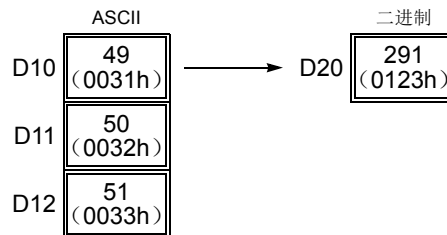
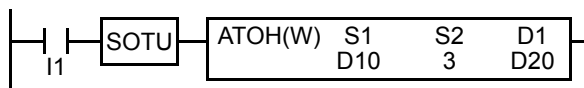
8: 数据转换指令

示例：ATOH

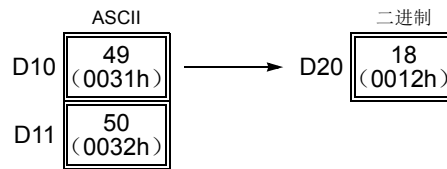
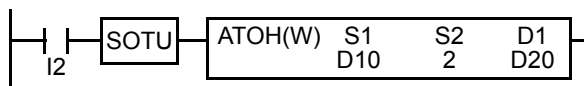
- 数字的位数：4



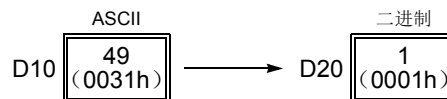
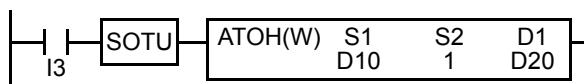
- 数字的位数：3



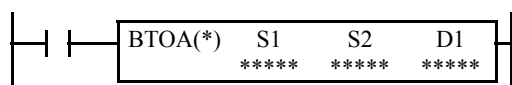
- 数字的位数：2



- 数字的位数：1



BTOA (BCD 码→ ASCII 码)



字数据类型: S1 → D1, D1+1, D1+2, D1+3, D1+4
 双字数据类型: S1·S1+1 → D1, D1+1, D1+2, ... , D1+9

输入打开时, S1 所指定的 16 位或 32 位二进制数据将转换为 BCD, 并转换为 ASCII 数据。将从最低位数字开始读取与 S2 所指定的数字的位数一样多的数据。结果存储到以 D1 所指定的设备开始的目标中。

要转换的数字的位数, 字数据类型为 1 - 5, 双字数据类型为 1 - 10。

双字数据类型适用于升级后的系统程序 Ver.200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的二进制数据	X	X	X	X	X	X	X	X	—
S2 (源 2)	要转换的数字的位数	X	X	X	X	X	X	X	1-5, 1-10	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

在 T (定时器) 或 C (计数器) 用作 S1 或 S2 时, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。

要转换的数字的位数, 字数据类型为 1 - 5, 双字数据类型为 1 - 10。请确保 S2 所指定的数字的数量在有效范围之内。如果 S2 数据不在有效范围内, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

请确保 D1+S2-1 所指定的最后一个源数据在有效范围之内。如果相加的源设备不在有效范围内, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 BTOA 指令是在输入打开时在每个扫描中执行的, 因此, 应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

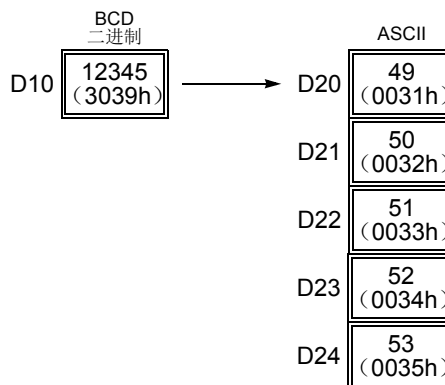
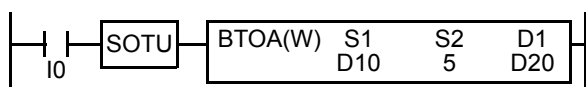
有效数据类型

W (字)	X	把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源时, 将使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	—	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标, 则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	—	
F (浮点)	—	

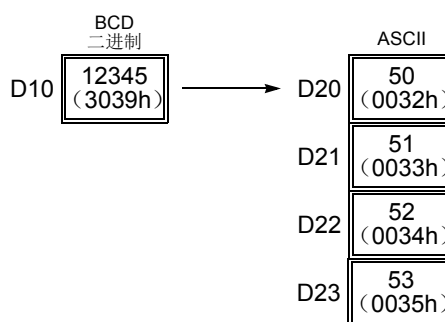
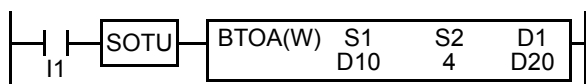
8: 数据转换指令

示例：BTOA(W)

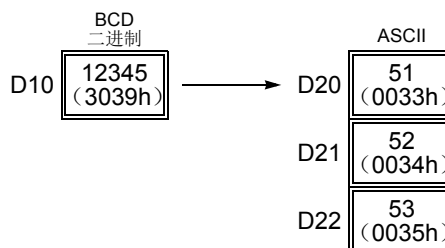
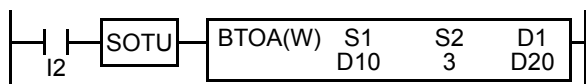
- 数字的位数：5



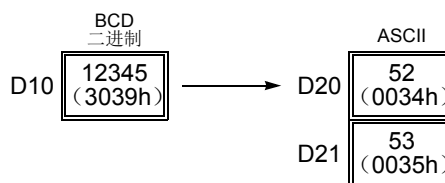
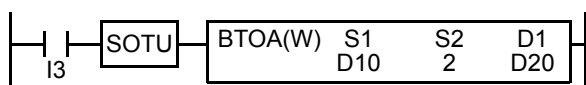
- 数字的位数：4



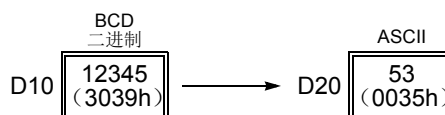
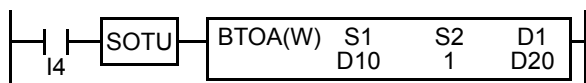
- 数字的位数：3



- 数字的位数：2

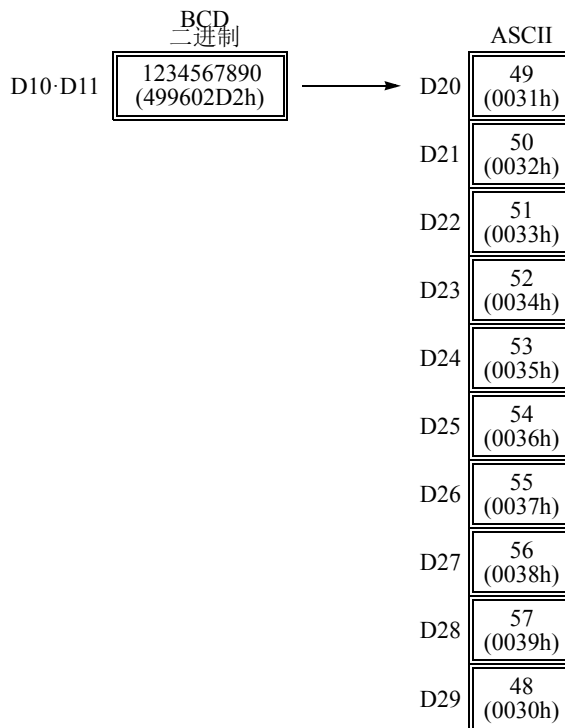
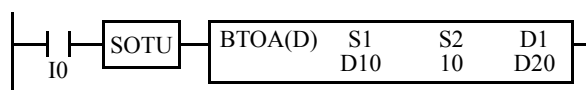


- 数字的位数：1

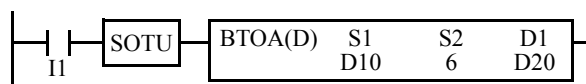


示例：BTOA(D)

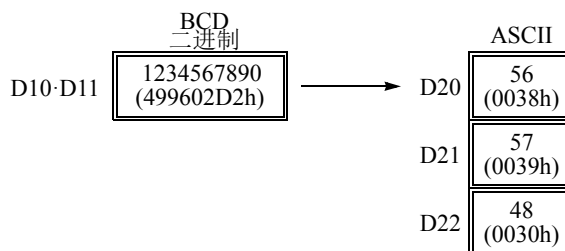
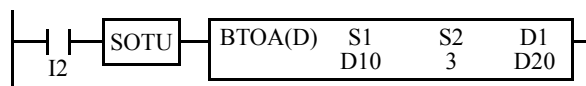
• 数字的位数：10



• 数字的位数：6

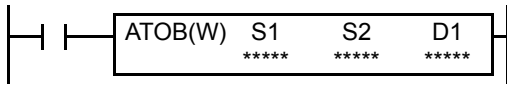


• 数字的位数：3



8: 数据转换指令

ATOB (ASCII 码→BCD 码)



字数据类型: S1, S1+1, S1+2, S1+3, S1+4 → D1
 双字数据类型: S1, S1+1, S1+2, ..., S1+9 → D1·D1+1

当输入打开时, S1 所指定的、与 S2 所指定的数字的位数一样多的 ASCII 数据将转换为 BCD, 并转换为 16 位或 32 位二进制数据。结果存储到设备 D1 所指定的目标中。

要转换的源数据的有效值是 30h - 39h。

要转换的数字的位数, 字数据类型为 1 - 5, 双字数据类型为 1 - 10。双字数据类型适用于升级后的系统程序 Ver.200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的 ASCII 数据	—	—	—	—	—	—	X	—	—
S2 (源 2)	要转换的数字的位数	X	X	X	X	X	X	X	1-5,1-10	—
D1 (目标 1)	要存储转换结果的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S2 时, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时, 数据将作为预置值 (TP 或 CP) 写入, 此值介于 0 到 65535 之间。

要转换的源 S1 数据的有效值是 30h - 39h。要转换的数字的位数: 字数据类型为 1 ~ 5, 双字数据类型为 1 ~ 10。请确保 S1 所指定的每个源数据的值和 S2 所指定的数字的位数在有效范围之内。如果 S1 或 S2 数据不在有效范围内, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

请确保 S1+S2-1 所指定的最后一个源数据在有效范围之内。如果相加的源设备不在有效范围内, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

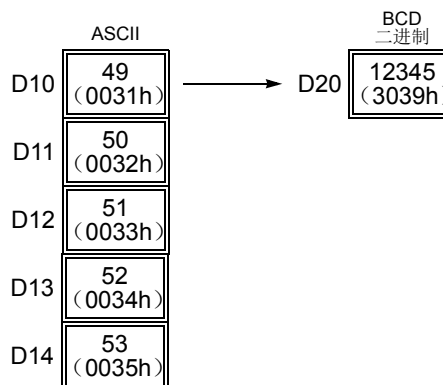
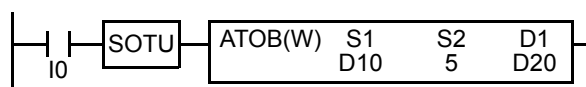
由于 ATOB 指令是在输入打开时在每个扫描中执行的, 因此, 应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

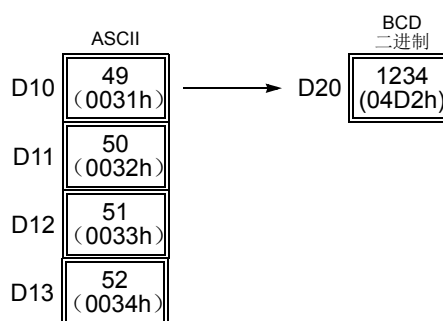
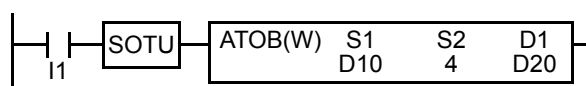
W (字)	X	把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标时, 将使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。
I (整数)	—	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标, 则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。
L (长整数)	—	
F (浮点)	—	

示例：ATOB(W)

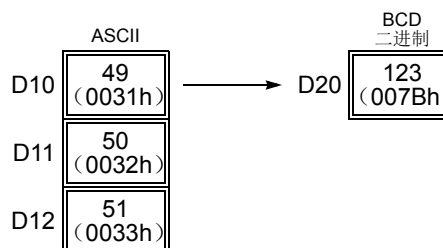
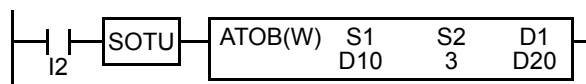
• 数字的位数：5



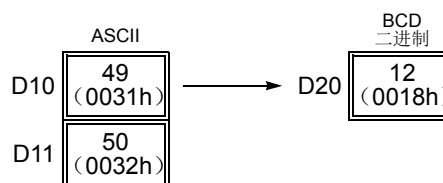
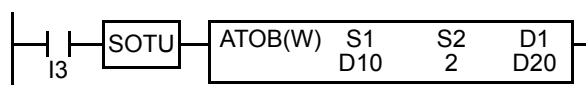
• 数字的位数：4



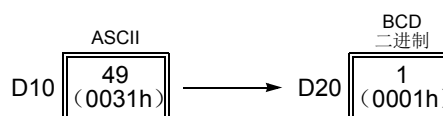
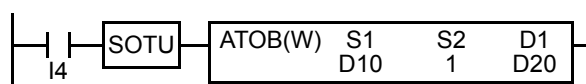
• 数字的位数：3



• 数字的位数：2



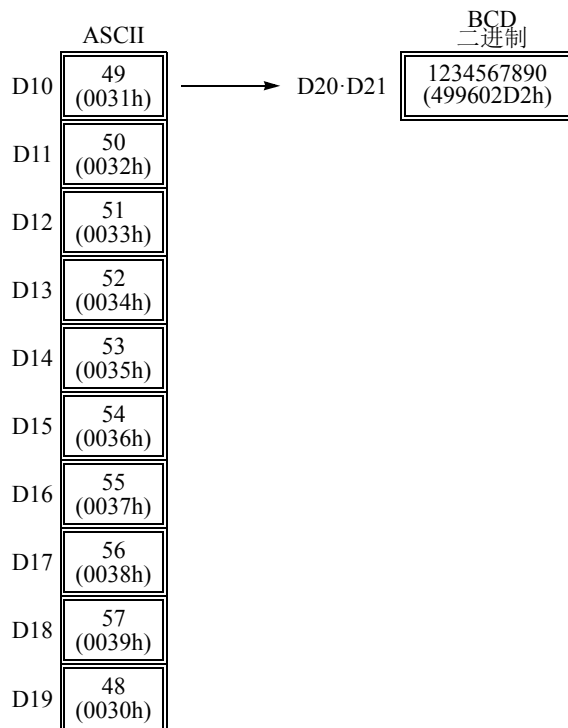
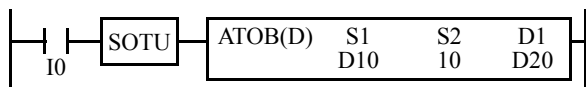
• 数字的位数：1



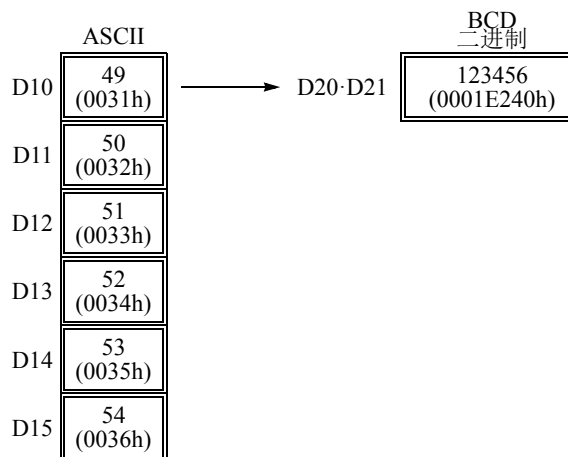
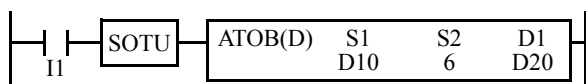
8: 数据转换指令

示例：ATOB(D)

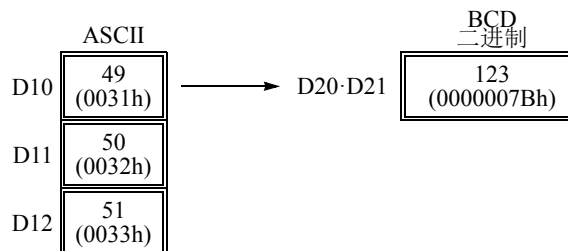
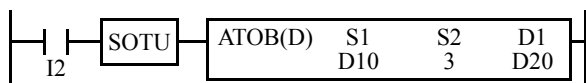
- 数字的位数：10



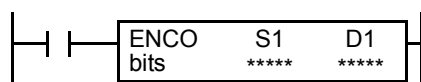
- 数字的位数：6



- 数字的位数：3



ENCO (编码)



当输入打开时，将寻找打开的位。搜索从 S1 开始，直到找到被置位（开）的第一个点。从 S1 到第一个置位点的点数（偏移量）将存储到设备 D1 所指定的目标中。

如果搜索区域内没有打开的点，则将 65535 存储到 D1。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	开始搜索的第一个位	X	X	X	X	—	—	X	—	—
D1 (目标 1)	用于存储搜索结果的目标	—	X	▲	X	—	—	X	—	—
bits	搜索的位数	—	—	—	—	—	—	—	1-256	—

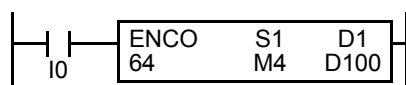
关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

用于指定要搜索的位数有效值是 1 - 256。请确保 S1 加上位数而得到的搜索区域在有效值范围之内。如果源数据不在有效范围内，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

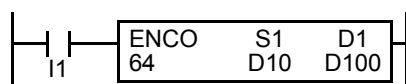
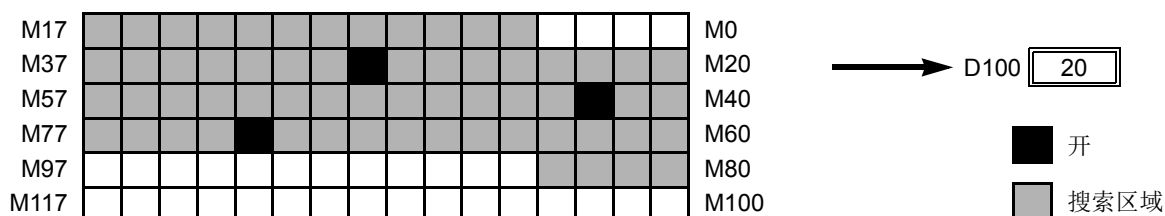
由于 ENCO 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

示例：ENCO



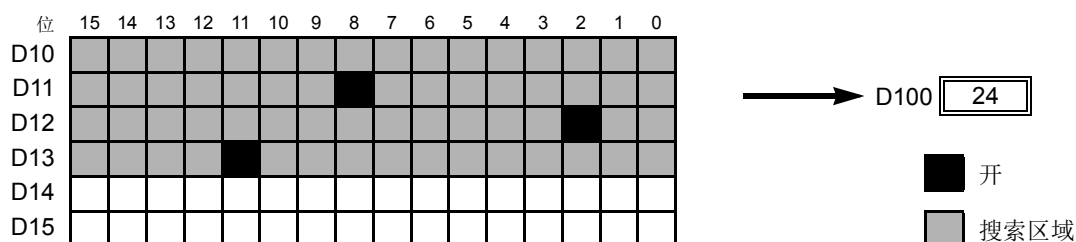
当输入 I0 打开时，将在从设备 S1 所指定的内部继电器 M4 开始的 64 个位中寻找打开的位。

由于内部继电器 M30 是第一个打开的点，因此从第一个搜索点开始的偏移量是 20，将把 20 存储到设备 D1 所指定的数据寄存器 D100 中。



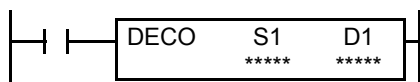
当输入 I1 打开时，将在从设备 S1 所指定的数据寄存器 D10 的第 0 位开始的 64 个位中寻找打开的位。

由于数据寄存器 D11 的第 8 位是第一个打开的位，因此从第一个搜索点开始的偏移量是 24，将把 24 存储到设备 D1 所指定的数据寄存器 D100 中。



8: 数据转换指令

DECO (解码)



当输入打开时，S1 和 D1 所指定的设备中所包含的值将相加，以确定目标，并打开由此确定的位。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	偏移量	X	X	X	X	—	—	X	0-255	—
D1 (目标 1)	要计数偏移量的第一个位	—	X	▲	X	—	—	X	—	—

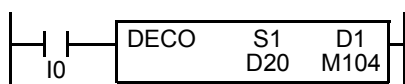
关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

源设备 S1 所指定的偏移量的有效值是 0 - 255。请确保 S1 所指定的偏移量以及 S1 和 D1 之和所确定的目标数据的最后一个位在有效值范围之内。如果偏移量或目标数据不在有效范围内，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

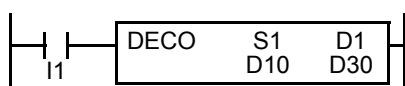
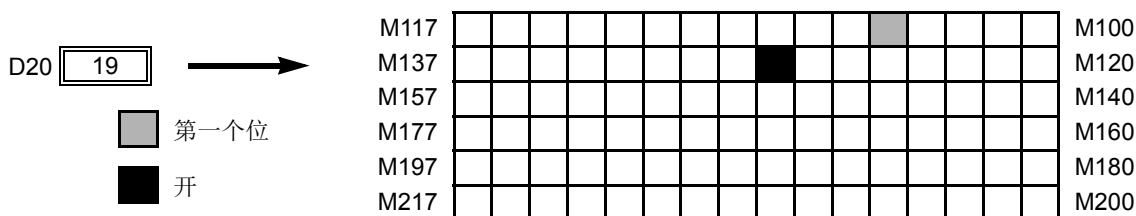
由于 DECO 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

示例 : DECO



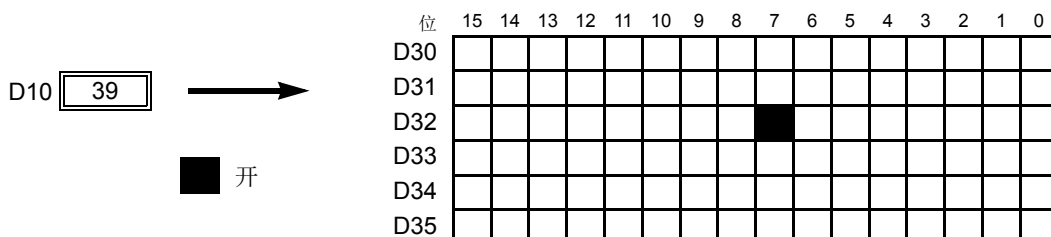
当输入 I0 打开时，将通过把设备 S1 所指定的数据寄存器 D20 所包含的值加上目标设备 D1 所指定的内部继电器 M104 来确定目标位。

由于来自内部继电器 M104 的第 19 位是内部继电器 M127，因此将打开由此确定的位。



当输入 I1 打开时，将通过把包含在设备 S1 所指定的数据寄存器 D10 中的值加上目标设备 D1 所指定的数据寄存器 D30 来确定目标位。

由于从数据寄存器 D30 的第 0 位开始的第 39 位是数据寄存器 D32 的第 7 位，由此确定的位将打开。



BCNT (位计数)



当输入打开时，将在一组从源设备 S1 所指定的点开始的连续位中寻找打开的位。源设备 S2 指定搜索的位数。打开的位的个数将存储到设备 D1 所指定的目标中。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	开始搜索的第一个位	X	X	X	X	—	—	X	—	—
S2 (源 2)	搜索的位数	X	X	X	X	X	X	X	1-256	—
D1 (目标 1)	要存储“开”位数的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

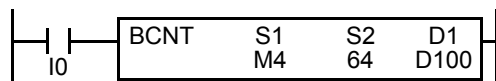
▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S2 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

用于指定搜索位数的 S2 的有效值是 1 - 256。请确保 S1 加上 S2 所指定的搜索区域在有效值范围之内。如果源数据不在有效范围内，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

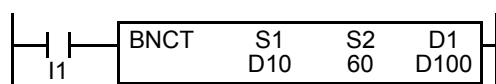
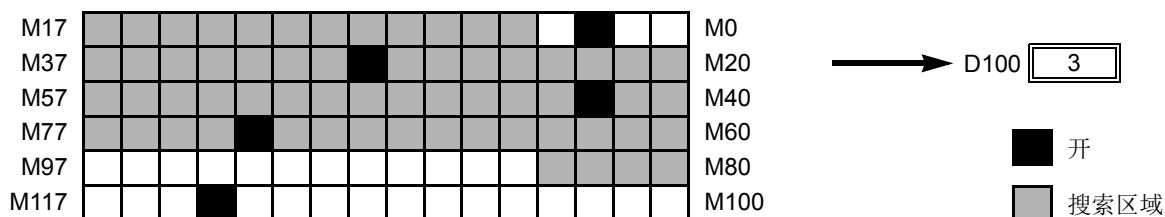
由于 BCNT 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

示例：BCNT



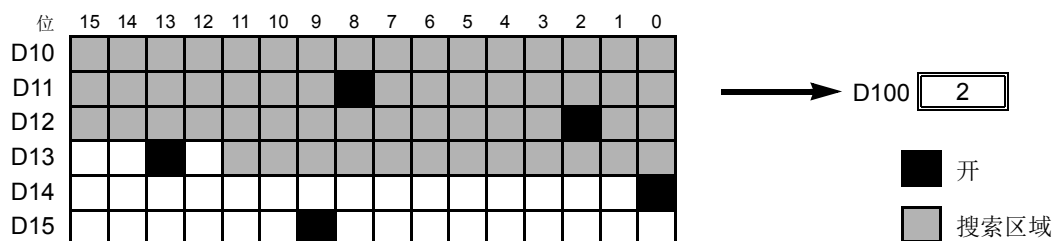
当输入 I0 打开时，将在从源设备 S1 所指定的内部继电器 M4 开始的一组 64 位中寻找打开的位。

由于 3 个位在搜索区域中是打开的，该数量将存储到由目标设备 D1 所指定的数据寄存器 D100 中。



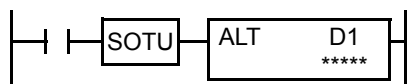
当输入 I1 打开时，将在从设备 S1 所指定的数据寄存器 D10 的第 0 位开始的 60 位中寻找打开的位。

由于在 60 位中有 2 个位是打开的，因此 2 将存储到设备 D1 所指定的数据寄存器 D100 中。



8: 数据转换指令

ALT (交替输出)



当输入打开时，D1 所指定的输出、内部继电器或移位寄存器位将打开，并在输入关闭之后保持不变。

当输入再次打开时，指定的输出、内部继电器或移位寄存器位将关闭。

ALT 指令必须与 SOTU 或 SOTD 指令一起使用，否则指定的输出、内部继电器或移位寄存器位将在每个扫描中重复打开和关闭。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

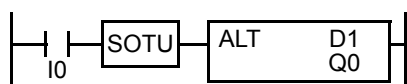
有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
D1 (目标 1)	要打开和关闭的位	—	X	X	X	—	—	—	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

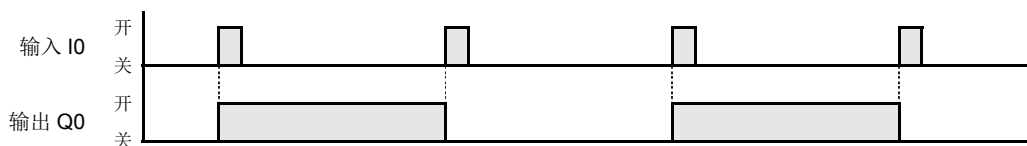
由于 ALT 指令是在输入打开时在每个扫描中执行的，因此必须使用来自 SOTU 或 SOTD 指令的脉冲输入。

示例 : ALT

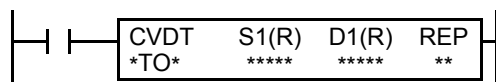


当输入 I0 打开时，设备 D1 所指定的输出 Q0 将打开，并在输入 I0 关闭之后保持不变。

当输入 I0 再次打开时，输出 Q0 将关闭。



CVDT (转换数据类型)



S1 → D1

当输入打开时，S1 所指定的 16 位或 32 位数据的数据类型将转换并存储到设备 D1 所指定的目标中。

数据类型可以分别指定为源和目标。

数据类型	W、I	D、L、F
源	S1	S1·S1+1
目标	D1	D1·D1+1

当给源和目标指定了相同的数据类型时，CVDT 指令具有与 MOV 指令相同的作用。

除非是给源和目标选择了 F(浮点)数据类型，只传送整数数据，省略小数。

当源数据超过了目标数据类型的范围时，目标在目标数据类型内存储一个最接近源数据的值。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	存储转换数据类型的起始设备编号	X	X	X	X	X	X	X	X	1-99
D1 (目标 1)	存储转换数据的起始设备编号	—	X	▲	X	X	X	X	—	1-99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S1 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65535 之间。

当选择 F (浮点) 数据类型时，只有数据寄存器和常量可指定为 S1，只有数据寄存器可指定为 D1。

当选择 F (浮点) 数据类型，并且 S1 或 D1 不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

有效数据类型

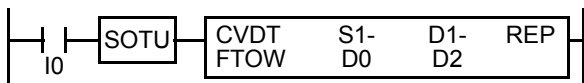
W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字或整数数据类型) 或 32 点 (双字、长整数或浮点数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整数)	X	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字或整数数据类型) 或 2 点 (双字、长整数或浮点数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
L (长整数)	X	
F (浮点)	X	

8: 数据转换指令

示例：CVDT

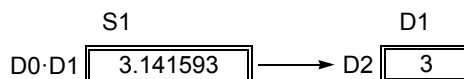
- 数据类型：S1 或 D1 不是 F(浮点)

除非是给源和目标选择了 F(浮点) 数据类型，只传送整数数据，省略小数。



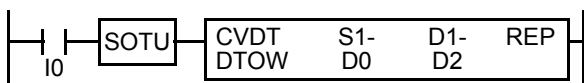
设备	数据类型	值
源	F	3.141593
目标	W	3

当输入 I0 打开时，3 将存储到数据寄存器 D2。



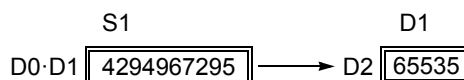
- 数据类型：S1 有一个比 D1 更大的数据范围

当源数据超过了目标数据类型的范围时，目标在目标数据类型内存储一个最接近源数据的值。

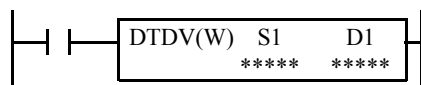


设备	数据类型	值
源	D	4294967295
目标	W	65535

当输入 I0 打开时，65535 将存储到数据寄存器 D2。



DTDV (数据分割)



S1 → D1, D1+1

当输入打开时, S1 所指定的 16 位二进制数据分割为高位字节或低位字节。高位字节数据存储到设备 D1 所指定的目标中。低位字节数据存储到 D1 之后的设备所指定的目标中。

该指令适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换的二进制数据	X	X	X	X	X	X	X	X	-
D1 (目标 1)	要存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

在 T (定时器) 或 C (计数器) 用作 S1, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。

目标设备 D1 使用以 D1 指定的设备开始的两个数据寄存器。

由于 DTDV 指令是在输入打开时在每个扫描中执行的, 因此, 应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

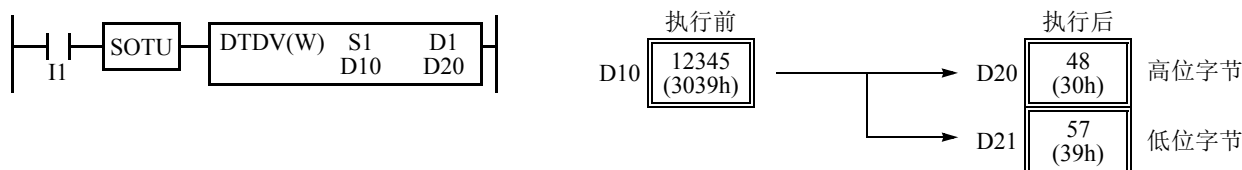
有效数据类型

W (字)	X
I (整数)	-
D (双字)	-
L (长整数)	-
F (浮点)	-

把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源时, 将使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。

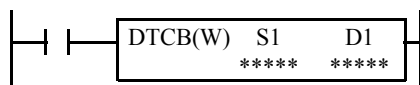
如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标, 则使用 1 点 (字数据类型)。

示例: DTDV



8: 数据转换指令

DTCB (数据组合)



S1, S1+1 → D1

输入打开时，低位字节数据从以 S1 指定的设备的两个连续源开始读取，并组合为 16 位数据。起始源设备的低位字节数据将传送到以 D1 指定的目标高位字节中，并把下一个源设备的低位字节数据传送到目标低位字节。

此指令适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要组合的二进制数据	-	-	-	-	-	-	X	-	-
D1 (目标 1)	要存储结果的目标	-	X	▲	X	X	X	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 不能将特殊内部继电器指定为 D1。

当 T (定时器) 或 C (计数器) 用作 D1，将数据作为预置值 (TP 或 CP) (0 ~ 65535) 写入。

源设备 S1 使用以 S1 指定的设备开始的两个数据寄存器。

由于 DTCB 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

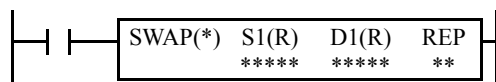
有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为目标，则使用 16 点 (字型或整数型数据类型)。
I (整数)	-	
D (双字)	-	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字型数据类型)。
L (长整数)	-	
F (浮点)	-	

示例 : DTCB



SWAP (数据交换)



S1 → D1

输入打开时，以 S1 指定的高位字节的字或双字的字数据与低位字节的同类型字数据进行交换，并将其结果保存到以 D1 指定的目标中。

此指令适用于升级后的系统程序 Ver. 200 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要交换的二进制数据	-	-	-	-	-	-	X	-	1 ~ 99
D1 (目标 1)	要存储转换结果的目标	-	-	-	-	-	-	X	-	1 ~ 99

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

由于 SWAP 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用时来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	X	如果将 D (数据寄存器) 指定为源或目标，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。如果指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
I (整数)	-	
D (双字)	X	
L (长整数)	-	
F (浮点)	-	

示例: SWAP

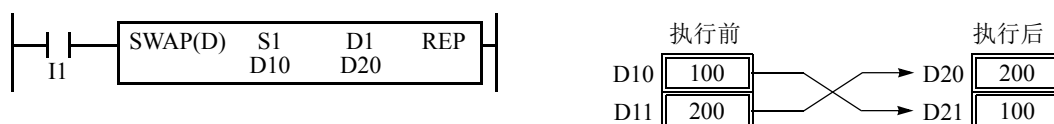
• 数据类型: W (字)

输入 I0 打开时，以源设备 S1 指定的，在数据寄存器 D10 中的 16 位数据的高位与低位字节数据进行交换，并将其结果保存到以 D1 指定的数据寄存器 D20 中。



• 数据类型: D (双字)

输入 I1 打开时，以源设备 S1 指定的在数据寄存器 D10 和 D11 中的 32 位数据的高位与低位字节数据进行交换，并将其结果保存到以 D1 指定的数据寄存器 D20 和 D21 中。



9: 周程序指令

简介

可以使用数量无限制的 WKTIM 指令以便按预定的次数和星期，打开和关闭指定的输出和内部继电器。

一旦设置了内部日历 / 时钟，WKTIM 指令就将把预定的时间与时钟盒中的时钟数据进行比较。如果达到预置时间，被指定为目标设备的内部继电器或输出就将按所安排的日程打开或关闭。关于设置日历 / 时钟的详情，请参阅第 9-6 页。

关于时钟盒规格的信息，请参阅第 2-95 页 (基本卷)。

WKTIM (周定时器)



当输入打开时，WKTIM 将把 S1 和 S2 的预置数据与当前日期和时间进行比较。

当前日期和时间达到预置值时，设备 D1 所指定的输出或内部继电器就将打开，具体取决于 MODE 所指定的周表输出控制。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
MODE	周表输出控制	—	—	—	—	—	—	—	0-2	—
S1 (源 1)	星期比较数据	—	—	—	—	—	—	X	0-127	—
S2 (源 2)	要打开的时 / 分比较数据	—	—	—	—	—	—	X	0-2359	—
S3 (源 3)	要关闭的时 / 分比较数据	—	—	—	—	—	—	X	0-2359	—
D1 (目标 1)	比较“开”输出	—	X	▲	—	—	—	—	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 ~ M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

MODE—周表输出控制 (0 ~ 2)

0: 禁用周表

当前日期和时间达到 S1、S2 和 S3 的预置值时，指定的输出或内部继电器将打开或关闭。不使用 WKTBL 时，应当将 MODE 设置为 0；即使编写了 WKTBL 指令，它也将被忽略。

1: 周表中的附加天数

当前时间达到为 S2 或 S3 所设置的、在 WKTBL 中编写的特殊日期的时 / 分比较数据时，指定的输出或内部继电器将打开 (S2) 或关闭 (S3)。

2: 周表中跳过的天数

在 WKTBL 中编写的特殊日期中，指定的输出或内部继电器将不会打开或关闭，即使当前日期和时间达到 S1、S2 和 S3 的预置值。

注释: 如果为 MODE 设置了 1 或 2，请使用 WKTBL 指令 (后跟 WKTIM 指令) 在周表中编写特殊日期。如果在 WKTIM 指令中为 MODE 设置了 1 或 2 但不编写 WKTBL 指令，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。如果在 WKTBL 指令之前执行 WKTIM 指令，还会发生相同的错误。

9: 周程序指令

S1— 星期比较数据 (0 ~ 127)

指定打开 D1 所指定的输出或内部继电器的星期。

星期	星期日	星期一	星期二	星期三	星期四	星期五	星期六
值	1	2	4	8	16	32	64

指定总值作为打开输出或内部继电器的设备 S1。

示例: 要在星期一至星期五打开输出, 请指定 62 为 S1, 因为 $2 + 4 + 8 + 16 + 32 = 62$ 。

S2— 要打开的时 / 分比较数据

S3— 要关闭的时 / 分比较数据

指定打开 (S2) 或关闭 (S3) D1 所指定的输出或内部继电器的小时和分钟。

小时	分钟	禁用比较
00 ~ 23	00 ~ 59	10000

示例: 要使用 WKTIM 指令在上午 8:30 打开输出或内部继电器, 请将 830 指定为 S2。要在下午 5:05 关闭输出或内部继电器, 请将 1705 指定为 S3。

如果将 10000 设置为时 / 分比较数据, 该比较数据将被忽略。例如, 如果将 10000 设置为要关闭的时 / 分比较数据 (S3), 则 WKTIM 指令将只比较要打开的时 / 分比较数据 (S2)。

如果要打开的时 / 分比较数据 (S2) 大于要关闭的时 / 分比较数据 (S3), 则比较“开”输出 (D1) 将在 S1 所指定日期的 S2 时候打开, 并在上午 0 时以后一直打开, 然后在下一天的 S3 时候关闭。例如, 如果 S2 是 2300, S3 是 100, 并且 S1 中包括星期一, 那么 D1 指定的输出将在星期一晚上 23 时打开, 并在星期二早上 1 时关闭。

请确保为 MODE、S1、S2 和 S3 设置的值在有效范围之内。如果任何数据超过有效值, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

WKTBL (周表)



S1, S2, S3, ..., SN → 周表 (N ≤ 20)

输入打开时, S1、S2、S3、...、SN 所指定的设备中的 N 块特殊月 / 日数据将被设置到周表。

特殊日期的数量最多可以是 20。

存储在周表中的特殊天数用于加上或跳过一定的天数, 以打开或关闭在随后的 WKTIM 指令中编写的比较输出。

WKTBL 必须位于 WKTIM 指令的前面。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	特殊月 / 日数据	—	—	—	—	—	—	X	101-1231	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

S1 ~ SN — 特殊月 / 日数据

指定要加上或跳过的月数和天数，以打开或关闭在 WKTIM 指令中编写的比较输出。

月	日
01 ~ 12	01 ~ 31

示例：要将 7 月 4 日设置为特殊天，请将 704 指定为 S1。

确保为 S1 ~ SN 所设置的值在有效范围内。如果任何数据超过有效值，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

示例 :WKTIM 和 WKTBL**• 没有特殊天 (MODE = 0)**

此示例是周程序员应用程序的基本程序，但没有使用 WKTBL (周表) 指令。CPU 运行时，WKTIM 将把 S1、S2 和 S3 的预置数据与当前日期和时间进行比较。

如果当前日期和时间达到预置值，则设备 D1 所指定的输出将打开和关闭。

M8125	WKTIM	S1	S2	S3	D1
	0	62	830	1715	Q0

M8125 是运行中输出特殊内部继电器。

S1 (62) 指定星期一~星期五。

WKTIM 在在星期一~星期五的 8:30 打开输出 Q0，并在 17:15 关闭输出 Q0。

• 周表中有附加天数 (MODE = 1)

如果当前时间达到在 WKTBL 中编写的特殊日的时 / 分预置时间，则指定的输出将打开或关闭。此外，指定的输出将按照 WKTIM 的设备 S1 的指定在每周打开和关闭。

在标准执行过程中，如果当前日期和时间与 WKTIM 的预置日期 (S1) 和时间 (S2 或 S3) 一致，则指定的输出将打开或关闭。在特殊日期执行的优先级高于在标准日期执行。

此示例演示在除正常周末以外的特殊日期的操作。输出将在每个星期六和星期日从上午 10:30 到晚上 11:10 打开。在不考虑星期的情况下，输出还将在 12 月 31 日到 1 月 3 日打开。

M8120	WKTBL	S1	S2	S3	S4
		1231	101	102	103
M8125	WKTIM	S1	S2	S3	D1
	1	65	1030	2310	Q0

M8120 是初始化脉冲特殊内部继电器。

WKTBL 指定 12 月 31 日~1 月 3 日为特殊日期。

MODE (1) 加上特殊日期。

S1 (65) 指定星期六和星期日。

WKTIM 在每个星期六、星期日和特殊日的 10:30 打开输出 Q0，并在 23:10 关闭。

• 周表中有跳过天数 (MODE = 2)

在 WKTBL 中设置的特殊日中，当指定的输出按照 WKTIM 的设备 S1 的指定在每周打开和关闭时，指定的输出不会打开或关闭。

在标准执行过程中，如果当前日期和时间与预置日期 (S1) 和时间 (S2 或 S3) 一致，则指定的输出将打开或关闭。在特殊日期执行的优先级高于在标准日期执行。

此示例演示了在特殊日期中止的操作。输出将在每个星期一~星期五的上午 10:00 ~ 晚上 8:00 打开，但从 5 月 2 日~5 月 5 日不会打开。

M8120	WKTBL	S1	S2	S3	S4
		502	503	504	505
M8125	WKTIM	S1	S2	S3	D1
	2	62	1000	2000	Q0

WKTBL 指定 5 月 2 日~5 月 5 日为特殊日。

MODE (2) 跳过特殊日。

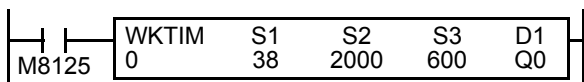
S1 (62) 指定星期一~星期五。

除特殊日以外，WKTIM 将在每个星期一~星期五的 10:00 打开输出 Q0，并在 20:00 关闭。

9: 周程序指令

• 使输出跨过晚上 0 点保持打开

如果要打开的时 / 分比较数据 (S2) 大于要关闭的时 / 分比较数据 (S3)，则比较“开”输出 (D1) 将在 S1 所指定日期的 S2 时候打开，并在上午 0 时以后一直打开，然后在下一天的 S3 时候关闭。此示例所演示的程序将使指定的输出跨过晚上 0 点保持打开，并在下一天关闭输出。



M8125 是运行中输出特殊内部继电器。

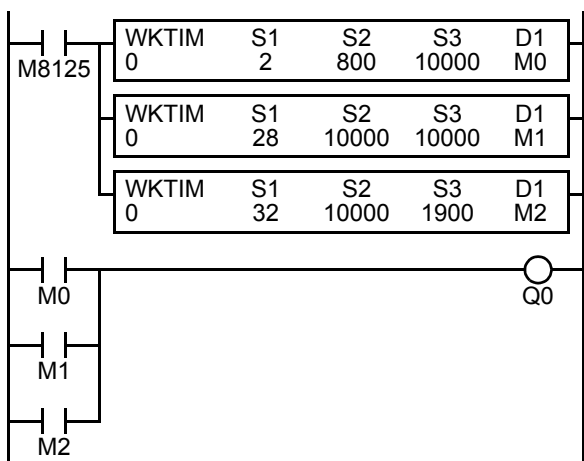
S1 (38) 指定星期一、星期二和星期五。

WKTIM 在星期一、星期二和星期五的 20:00 打开输出 Q0，并在下一天的 6:00 关闭输出 Q0。



• 使输出几天保持打开

可以用多个 WKTIM 指令来使输出在超过 24 小时的时间内保持打开。此示例演示的程序将使指定的输出从每个星期一早上 8 点~每个星期五早上 7 点保持打开。



M8125 是运行中输出特殊内部继电器。

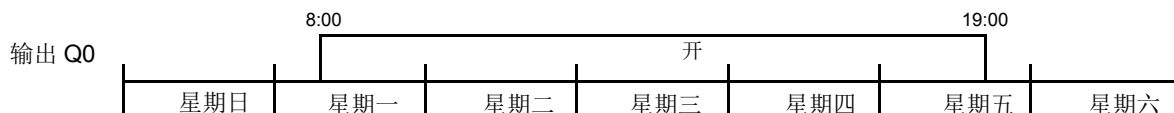
S1 (2) 指定星期一。

S1 (28) 指定星期二，星期三和星期四。

S1 (32) 指定星期五。

S2 (10000) 和 S3 (10000) 禁用小时和分钟数据的比较。

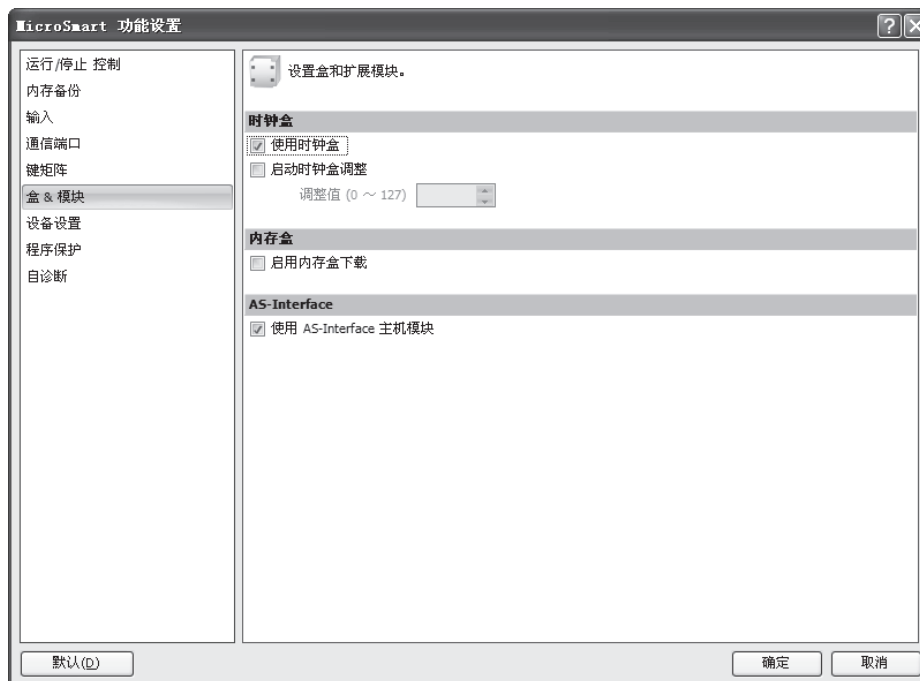
内部继电器 M0、M1 或 M2 打开时，输出 Q0 将保持打开。



使用时钟盒

当使用周程序指令时，您需要在 CPU 模块上安装时钟盒，并使用 WindLDR 启用使用时钟盒，步骤如下：

1. 在 WindLDR 菜单栏中选择 **设置 > 功能设置 > 盒 & 模块**。此时出现“盒 & 模块”的“功能设置”对话框。



2. 单击复选框启用使用时钟盒。
3. 单击**确定**按钮。
4. 将用户程序下载到 CPU 模块，并关闭然后打开 CPU 模块的电源。



注意

- 在拆卸时钟盒后，请勿运行带有“功能设置”使用时钟盒的用户程序，否则会发生 IC 错误，打开 CPU 模块上的 ERR LED。特殊数据寄存器 D8005(通用错误代码) 存储 400h(时钟 IC 错误)。

使用 WindLDR 设置日历 / 时钟

第一次使用时钟盒之前，必须先设置时钟盒中的日历 / 时钟数据，要进行该设置，可以使用 WindLDR，或执行一个用户程序，以便从分配给日历 / 时钟的特殊数据寄存器传送正确的日历 / 时钟数据。一旦日历 / 时钟数据被存储，数据将由时钟盒中的备份电池保持。

1. 从 WindLDR 菜单栏的中选择**联机**，然后选择**监控 > 监控**。屏幕显示将更改到监控窗口。
2. 在 **PLC** 菜单中选择**状态**。此时显示 MicroSmart PLC 状态对话框。将从时钟盒中读出当前日历 / 时钟数据，并在日历字段中显示出来。
3. 单击日历的**更改**按钮。此时弹出设置日期和时间对话框，其中的日期和时间值是从计算机内部时钟读取的。



4. 单击日历右侧的**向下箭头**按钮，然后显示日历，可以在其中更改年、月和日。输入或选择新的值。
5. 要更改小时和分钟，请单击**时间**框，并键入新的值，或使用上 / 下键。输入新值后，单击**确定**按钮，新的值将传送到时钟盒。

使用用户程序设置日历 / 时钟

设置日历 / 时钟数据的另一个方法是在日历和时钟专用的特殊数据寄存器中存储值，并打开特殊内部继电器 M8016、M8017 或 M8020。在执行用户程序之前，数据寄存器 D8015 ~ D8021 不会保持日历 / 时钟数据的当前值，而是保持未知值。

用于日历 / 时钟数据的特殊数据寄存器

数据寄存器编号	数据	值	读 / 写	更新
D8008	年 (当前数据)	0 ~ 99	只读	500 ms 或一个扫描周期的任何更大者
D8009	月 (当前数据)	1 ~ 12		
D8010	日 (当前数据)	1 ~ 31		
D8011	星期 (当前数据)	0 ~ 6 (注释)		
D8012	小时 (当前数据)	0 ~ 23		
D8013	分钟 (当前数据)	0 ~ 59		
D8014	秒 (当前数据)	0 ~ 59		
D8015	年 (新数据)	0 ~ 99	只写	不更新
D8016	月 (新数据)	1 ~ 12		
D8017	天 (新数据)	1 ~ 31		
D8018	星期 (新数据)	0 ~ 6 (注释)		
D8019	小时 (新数据)	0 ~ 23		
D8020	分钟 (新数据)	0 ~ 59		
D8021	秒 (新数据)	0 ~ 59		

注释：按如下所示为当前数据和新数据指定星期值：

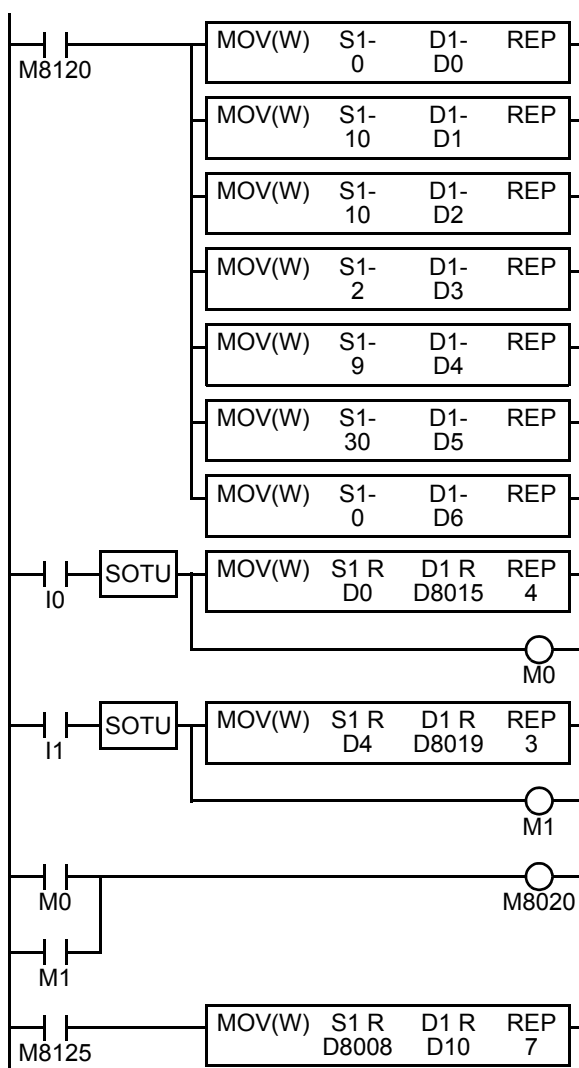
0	1	2	3	4	5	6
星期日	星期一	星期二	星期三	星期四	星期五	星期六

用于日历 / 时钟数据的特殊内部继电器

M8016	日历数据写入标记	当 M8016 打开时，数据寄存器 D8015 ~ D8018（日历新数据）中的数据将被设置到安装在 CPU 模块上的时钟盒中。
M8017	时钟数据写入标记	当 M8017 打开时，数据寄存器 D8019 ~ D8021（时钟新数据）中的数据将被设置到安装在 CPU 模块上的时钟盒中。
M8020	日历 / 时钟数据写入标记	当 M8020 打开时，数据寄存器 D8015 ~ D8021（时钟新数据）中的数据将被设置到安装在 CPU 模块上的时钟盒中。

示例：设置日历 / 时钟数据

此示例演示了如何使用梯形图程序设置日历 / 时钟数据。将新的日历 / 时钟数据存储到数据寄存器 D8015 ~ D8021 中之后，必须打开特殊内部继电器 M8020（日历 / 时钟数据写入标记），才能将新日历 / 时钟数据设置到时钟盒。



M8120 是初始化脉冲特殊内部继电器。

当 CPU 启动时，七条 MOV (W) 指令将把日历 / 时钟数据存储到数据寄存器 D0 ~ D6。

当输入 I0 打开时，新的日历数据（年、月、日和星期）将传送到数据寄存器 D8015 ~ D8018，并且内部继电器 M0 将在 1 个扫描周期以内打开。

当输入 I1 打开时，新的时钟数据（小时、分钟和秒）将传送到数据寄存器 D8019 ~ D8021，并且内部继电器 M1 将在 1 个扫描周期之内打开。

当 M0 或 M1 打开时，日历 / 时钟数据写入标记特殊内部继电器 M8020 将打开，以便将新的日历 / 时钟数据设置到时钟盒。

M8125 是运行中输出特殊内部继电器。

当 CPU 运行时，MOV (W) 将把当前日历 / 时钟数据传送到数据寄存器 D10 ~ D16。

使用用户程序调整时钟

可以通过特殊内部继电器 M8021（时钟数据调整标记）来调整时钟数据。当 M8021 打开时，将对时钟的秒进行调整。如果当前时间的秒在 0 ~ 29 之间，则将秒设置为 0，并使分钟保持不变。如果当前时间的秒在 30 ~ 59 之间，则将秒设置为 0，并将分钟加一。M8021 可用于从零秒开始的精确定时。

示例：将日历 / 时钟数据调整到 0 秒

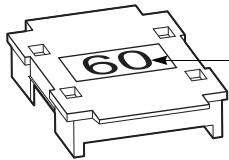


当输入 I12 打开时，时钟数据调整标记特殊内部继电器 M8021 将打开，并按秒调整时钟。

调整时钟盒准确性

可选的时钟盒（FC4A-PT1）每月会出现 ± 2 分钟的初始误差（在 25 °C 下）。通过使用“启用时钟盒调整”功能设置，可以将时钟盒的准确性提高到 ± 30 秒。

开始进行时钟盒调整之前，请确认时钟盒上所记录的调整值。该值是工厂出货之前在每个时钟盒上测量到的调整参数。



调整值

时钟盒上记录的调整值是在 25 °C 下为获得最佳准确性而测量到的。在其他温度下使用时钟盒时，时钟盒准确性可能不高。

设置 WindLDR

1. 在 WindLDR 菜单栏中选择 **设置 > 功能设置 > 盒 & 模块**。此时出现“盒 & 模块”的“功能设置”对话框。



2. 单击复选框以启用时钟盒调整，并在“调整值”字段中键入在时钟盒上找到的调整值。
3. 单击**确定**按钮。
4. 将用户程序下载到 CPU 模块，并关闭然后打开 CPU 模块的电源。

时钟盒备份持续时间

时钟盒数据是由时钟盒中的锂电池备份的，在 25 °C 下可以保存大约 30 天。如果 CPU 模块的断电时间超过备份保持时间，则时钟数据将初始化为以下值。

日历： 00/01/01
时间： 0:00:00 AM

10: 接口指令

简介

DISP（七段译码显示）指令用于在 7 段显示单元上显示 1 - 5 位数的定时器 / 计数器当前值和寄存器数据。

DGRD（数字读取）指令用于读取 1 - 5 位数的数字开关设置到寄存器。在使用数字开关更改定时器和计数器的预置值时使用该指令。

DISP（七段译码显示）



当输入打开时，由源设备指定的数据将分配给由设备 Q 指定的输出或内部继电器。该指令可用于向显示单元输出 7 段数据。

在用户程序中可以使用八个 DISP 指令。
显示数据可以为 0 - 65535（FFFFh）。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

注释: DISP 指令需要晶体管输出终端。当使用集成型 24-I/O 型 CPU 模块 FC5A-C24R2 时，请连接晶体管输出模块。

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要显示的数据	—	—	—	—	X	X	X	—	—
Q (输出)	显示数据的起始输出编号	—	X	▲	—	—	—	—	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

▲ 内部继电器 M0 - M2557 可以指定为 Q。特殊内部继电器不能指定为 Q。
当 T（定时器）或 C（计数器）用作 S1 时，将读出定时器 / 计数器的当前值（TC 或 CC）。

转换

- BCD:** 可连接 BCD（十进制）显示单元
- BIN:** 可连接 BIN（十六进制）显示单元

栓锁相和数据段

选择与显示单元的相匹配的栓锁相和数据段时，需考虑输出模块的沉型或源型输出。

输出点

所需输出点的数量为 4 加上要显示的数字位数。在显示 4 位数字，同时输出 Q0 指定为起始输出编号时，必须保存从 Q0 - Q7 共 8 个连续输出点。

显示处理时间

在打开输入至 DISP 指令后，显示一位数字需要扫描 3 次。按照下面所示时间持续输入 DISP 指令以处理显示数据的所有数字。

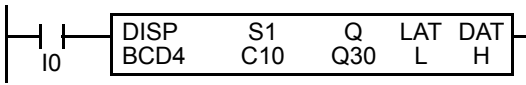
$$\text{显示处理时间} = 3 \text{ 次扫描} \times \text{数字的位数}$$

若扫描时间小于 2 ms，则无法正确显示数据。若扫描时间太短而无法确保正常显示，则将数据寄存器 D8022（固定扫描时间预置值）的值设为 3 或更大（单位是 ms）。请参阅第 5-50 页（基本卷）。

10: 接口指令

示例 : DISP

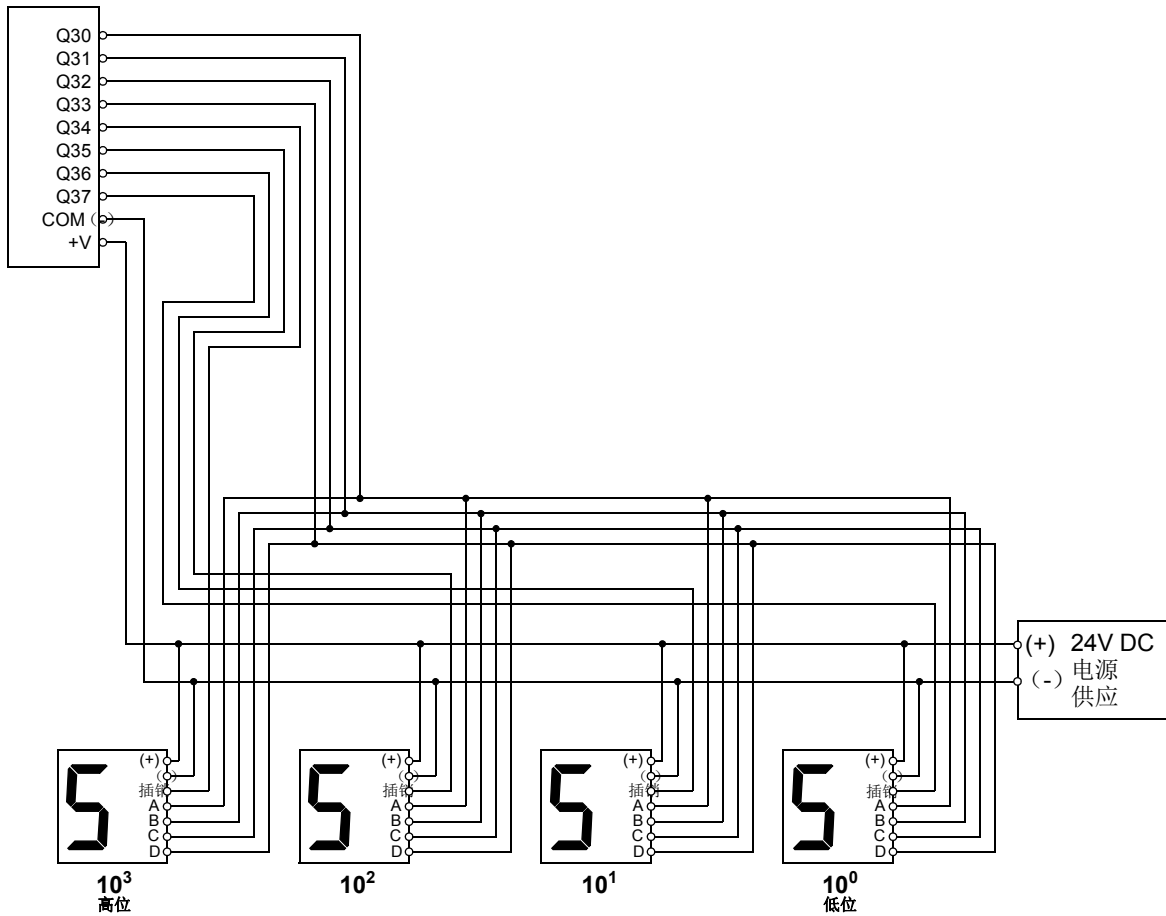
以下示例演示在已连接至晶体管沉型输出模块的 7 段显示单元 (IDEC DD3S-F31N) 上显示计数器 CNT10 的 4 位数字当前值的程序。



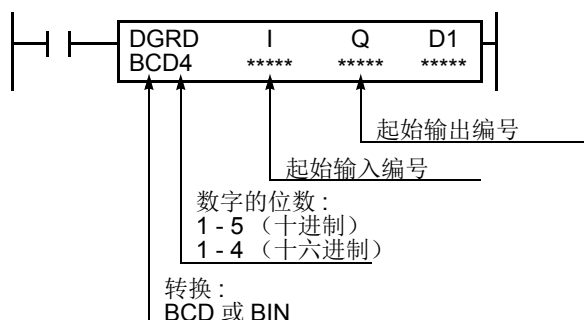
当输入 I0 打开时, 计数器 C10 的 4 位数字当前值将显示在 7 段数字显示单元上。

输出布线图

8 点晶体管沉型 输出模块 FC4A-T08K1



DGRD (数字读取)



当输入打开时，由设备 I 和 Q 指定的数据将设置在目标设备 D1 指定的寄存器。

可以用该指令在使用数字开关时更改定时器和计数器指令的预置值。用该指令读取的数据为 0 - 65535（5 位数字）或 FFFFh。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

注释：DGRD 指令需要晶体管输出终端。当使用集成型 24-I/O 型 CPU 模块 FC5A-C24R2 时，请连接晶体管输出模块。

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
I	要读取的起始输入编号	X	—	—	—	—	—	—	—	—
Q	用于数字选择的起始输出编号	—	X	—	—	—	—	—	—	—
D1 (目标 1)	用于存储结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

DGRD 指令最多可以读取 65535（5 位数字）。当数字的位数设为 5，且读取值超过 65535 时，会导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

注释：在用户程序中，DGRD 指令最多可以使用 16 次。在传送包含超过 16 个 DGRD 指令的用户程序至 CPU 时，会出现用户程序语法错误，并打开 ERR LED。无法执行用户程序。

转换

BCD: 可连接 BCD（十进制）数字开关

BIN: 可连接 BIN（十六进制）数字开关

输入点

输入用于读取数字开关中的数据。所需输入点的数量通常为 4。必须从以设备 I 指定的输入编号开始保留四个输入点。例如，当输入 I0 被指定为设备 I 时，使用输入 I0 - I3。

当使用 CPU 模块上的输入端子时，过滤值会起作用（默认值是 3 ms）。扩展输入模块上的输入端子有一个 4 ms 的固定值。有关输入过滤器，请参阅第 5-42 页（基本卷）。

输出点

输出用于选择要读取的数字。所需输出点的位数等于要读取的数字的数量。当连接最多 5 个数字开关时，必须从以设备 Q 指定的输出编号开始保留五个输出点。例如，当输出 Q0 指定为设备 Q 以读取 3 位数字时，使用输出 Q0 - Q2。

数字开关数据读取时间

在打开输入至 DGRD 指令后，需要以下时间来读取数字开关数据。按下面显示的时间持续显示输入 DGRD 指令以读取数字开关数据。例如，在从 5 个数字开关读取数据至目标设备时，需要扫描 14 次

$$\text{数字开关数据读取时间} \\ 2 \text{ 次扫描} \times (\text{数字的位数} + 2)$$

10: 接口指令

调整扫描时间

DGRD 指令所需扫描时间比过滤时间长 6 ms。

最低所需扫描时间

(扫描周期) \geq (过滤时间) + 6 ms

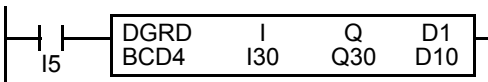
过滤时间取决于如以下所示使用的输入终端。

输入端子	过滤时间
CPU 模块上的 I0 - I7	在功能设置中所选的过滤值 (默认为 3 ms) 请参阅第 5-42 页 (基本卷) 上的输入过滤器。
CPU 模块上的 I10 - I17	3 ms (固定)
扩展输入模块上的输入	4 ms (固定)

当实际扫描时间太短而无法执行 DGRD 指令时, 请使用固定扫描功能。当输入过滤时间设置为 3 ms 时, 将特殊数据寄存器 D8022 (固定扫描时间预置值) 的值设为 9 或更多 (单位是 ms)。请参阅第 5-50 页 (基本卷)。当更改输入过滤时间时, 请为 D8022 设置正确的值以确保最低所需扫描时间为上述值。

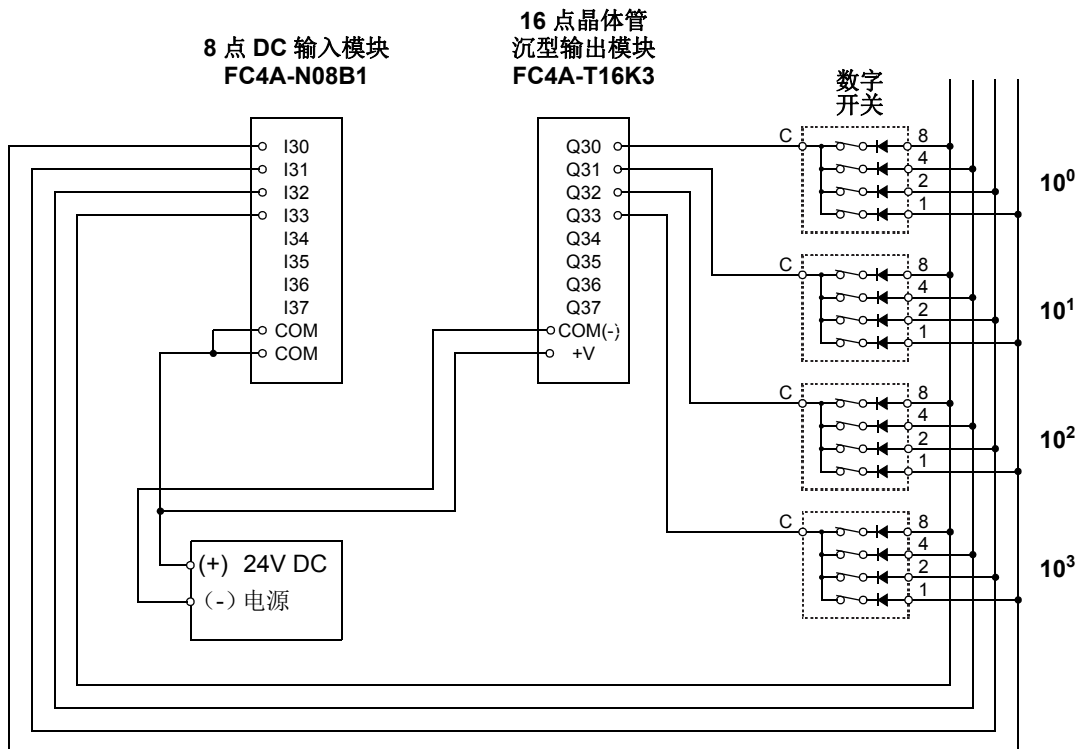
示例 : DGRD

以下示例演示使用 8 点 DC 输入模块和 16 点晶体管沉型输出模块从数字开关 (IDEC FBN-031D-B) 中读取数据至 CPU 模块的数据寄存器的程序。



当输入 I5 打开时, 将读取 BCD 数字开关的 4 位数值至数据寄存器 D10。

I/O 布线图



11: 程序分支指令

简介

当不能满足某些条件时，程序分支指令可以绕过部分程序以缩短执行时间。

LABEL 和 LJMP 是基本程序分支指令，可用于标记地址以及跳转至已标记的地址。编程工具包括在程序多个部分和调用一个子程序以使执行返回结束处的功能之间的“either/or”选项。

DI 或 EI 指令可单独禁用或启用中断输入和定时器中断。

LABEL (标签)



标签编号从 0 ~ 127 (集成型 CPU) 或 0 ~ 255 (超薄型 CPU) 用在程序分支开始执行程序指令的程序地址处。

END 指令可用于将已标记的程序部分与主程序分隔。这样，可以在满足输入条件前不执行程序分支以最小化扫描时间。

注释：同一标签编号只能使用一次。

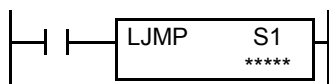
适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
标签编号	LJMP 和 LCAL 标记	—	—	—	—	—	—	—	0-127, 0-255	—

LJMP (标签跳转)



当输入打开时，将跳转至带有 S1 指定的标签 0 ~ 127 (集成型 CPU) 或 0 ~ 255 (超薄型 CPU) 的地址处。

当停止输入时，将不发生跳转，且程序会继续执行下一个指令。

LJMP 指令用作两部分程序之间的“either/or”选项。在程序分支后，程序执行不返回 LJMP 指令后的指令。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	标签编号至跳转至	—	—	—	—	—	—	X	0-127, 0-255	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

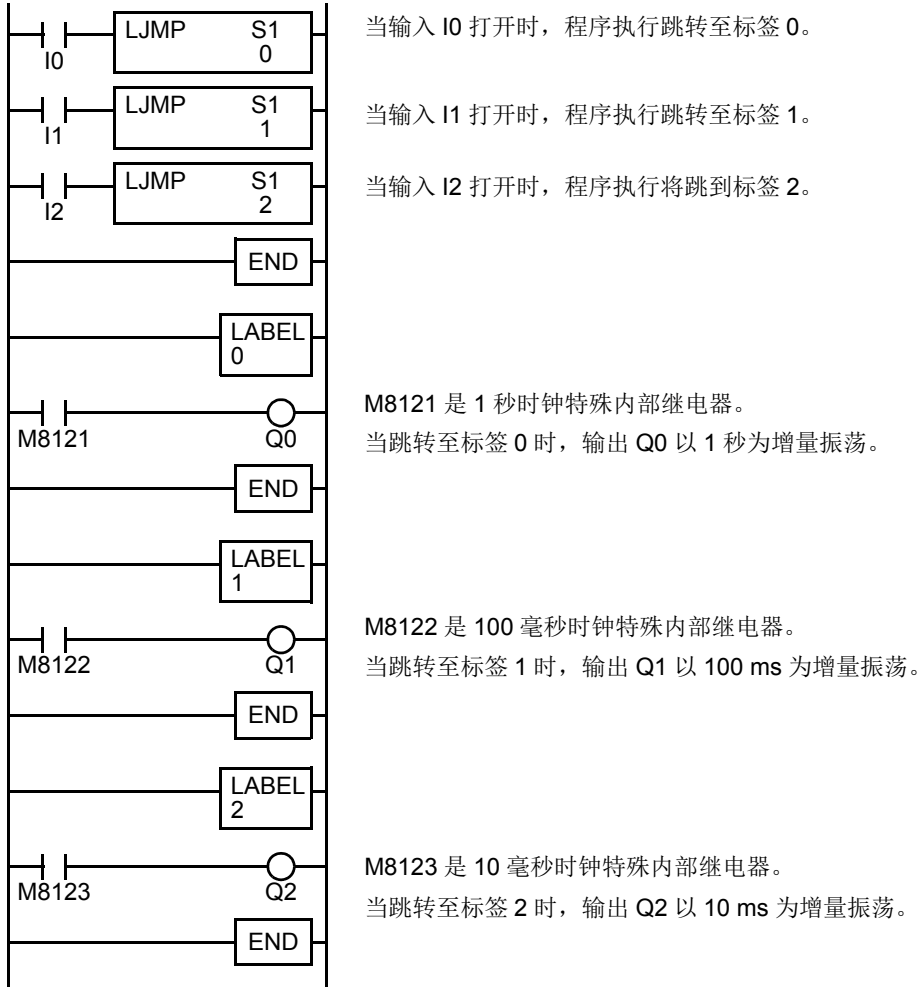
因为在打开输入时，每次扫描都要执行 LJMP 指令，所以应该按需使用 SOTU 或 SOTD 指令中的脉冲输入。

注释：确保已编写用于 LJMP 指令的标签编号的 LABEL 指令。当指定使用非常量的 S1 时，该标签值为变量。当标签为变量时，确保用户程序中包括所有可能的 LABEL 编号。如果没有匹配的标签，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

11: 程序分支指令

示例 :LJMP 和 LABEL

以下示例演示了根据输入跳转至程序三个不同部分的程序。

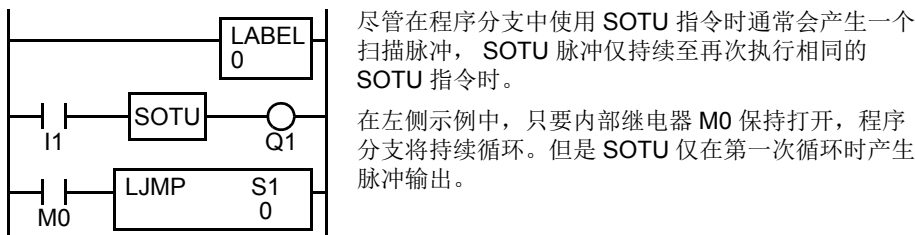


使用带程序分支的定时器指令

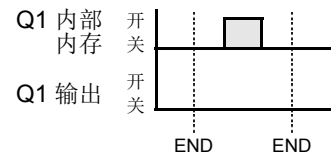
当 TML、TIM、TMH 或 TMS 指令的定时器开始输入已打开时，一跳转至以定时器当前值起始的位置立即开始计时。在使用程序分支时，确保在跳转后按需要初始化定时器很重要。如果需要在跳转后初始化定时器指令（设置为预置值），则应该在初始化之前关闭定时器开始输入以进行一次或多次扫描。否则，无法识别出定时器输入已打开。

使用带程序分支的 SOTU/SOTD 指令

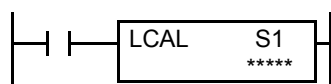
如果需要，在跳转时要检查计数器和移位寄存器的脉冲输入，以及保持单一输出（SOTU 和 SOTD）的输入。在跳转后关闭输入以进行一次或多次扫描来识别上升沿或下降沿转换。



只要 M0 保持打开便无法执行 END 指令，因此即使打开输入 I1，也无法打开输出 Q1。



LCAL（标签调用）



当输入打开时，将调用带有 S1 指定的标签 0 ~ 127（集成型 CPU）或 0 ~ 255（超薄型 CPU）的地址。当输入停止时，将不发生调用，且程序会继续执行下一个指令。

LCAL 指令调用子程序，并在执行分支后返回主程序。LRET 指令（如下所示）必须位于调用的程序分支的末尾，这样当返回 LCAL 指令后的指令时即可继续正常执行程序。

注释：必须使用 END 指令将主程序与 LCAL 指令调用的子程序分隔。

最多可嵌套四个 LCAL 指令。当嵌套多于四个 LCAL 指令时，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

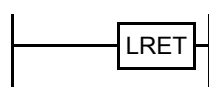
设备	功能	I	Q	M	R	T	C	D	常量	重复
S1（源 1）	要调用的标签编号	—	—	—	—	—	—	X	0-127, 0-255	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

因为在打开输入时，每次扫描都要执行 LCAL 指令，所以应该按需使用 SOTU 或 SOTD 指令中的脉冲输入。

注释：确保已编写用于 LCAL 指令的标签编号的 LABEL 指令。当指定使用非常量的 S1 时，该标签值为变量。当标签为变量时，确保用户程序中包括所有可能的 LABEL 编号。如果没有匹配的标签，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

LRET（标签返回）



该指令位于 LCAL 指令调用的子程序的末尾。当子程序完成时，返回 LCAL 指令后的指令即可继续正常执行程序。

必须将 LRET 置于以 LABEL 指令开始的子程序末尾处。若将 LRET 编写于其他位置，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

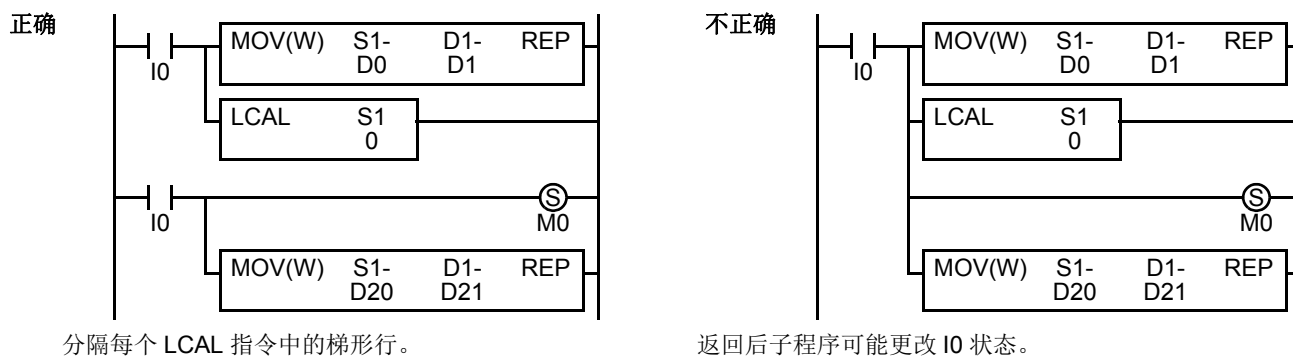
设备	功能	I	Q	M	R	T	C	D	常量	重复
—	—	—	—	—	—	—	—	—	—	—

11: 程序分支指令

调用子程序的正确结构

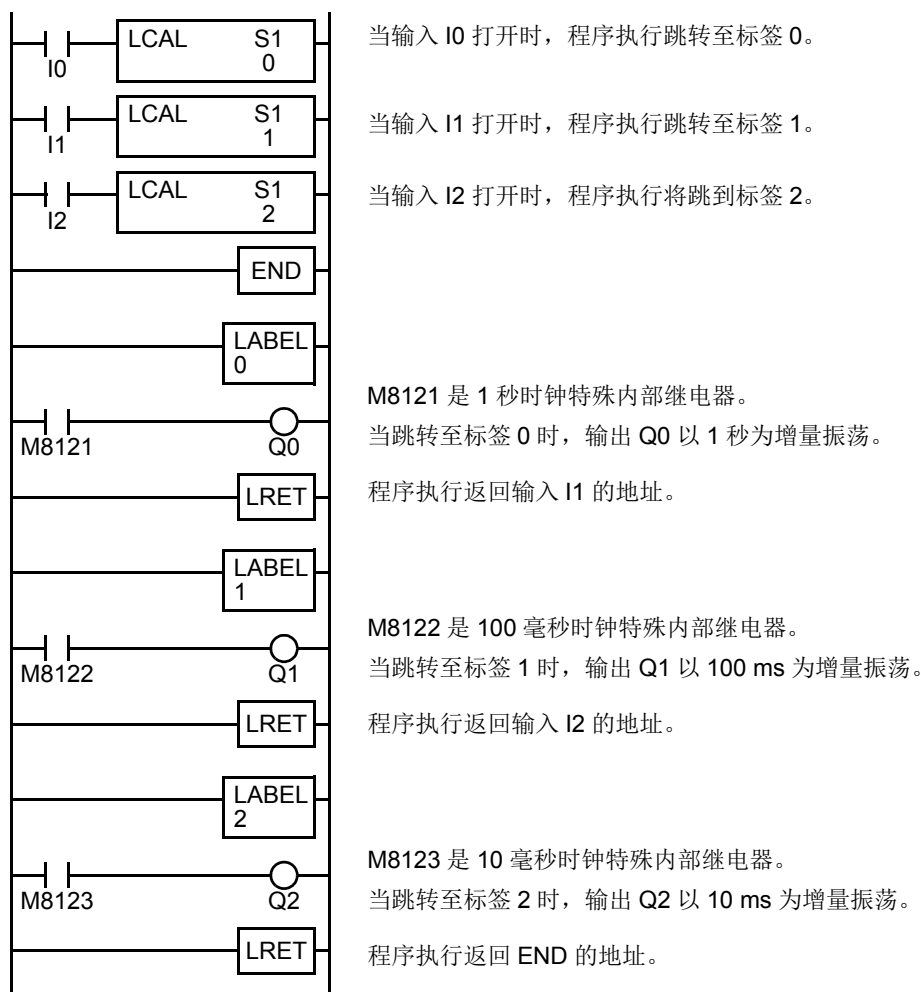
在执行 LCAL 指令时，如果子程序更改了输入条件，则相同梯形阶上的剩余程序指令在返回后可能无法执行。在子程序的 LRET 指令后，根据当前输入条件，程序开始执行以 LCAL 指令后的指令。

在调用子程序后必须执行 LCAL 指令后的指令时，确保子程序不会错误更改输入条件。另外，从 LCAL 指令中分隔的新梯形行中包括后续指令。



示例 :LCAL 和 LRET

以下示例演示了根据输入调用程序三个不同部分的程序。当子程序完成时，程序执行返回 LCAL 指令后的指令。



DJNZ (递减跳转非零)



当输入打开时，存储在由 S1 指定的数据寄存器中的值被递减 1 并进行核对。如果结果不为 0 时，程序执行跳转到由 S2 指定的标签 0 ~ 127 (集成型 CPU) 或 0 ~ 255 (超薄型 CPU) 的地址。如果递减结果为 0 时，不发生跳转，并且程序执行处理下一个指令。

本指令适用于升级后的系统程序 Ver. 210 或更高版本的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	递减值	-	-	-	-	-	-	X	-	-
S2 (源 2)	要跳转到的标签编号	-	-	-	-	-	-	X	0 ~ 127 0 ~ 255	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

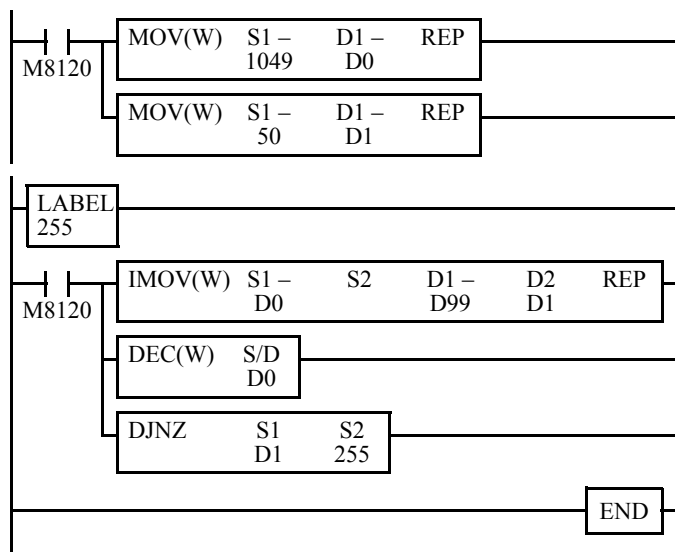
由于 DJNZ 指令在输入打开时的每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

标签编号的值域为 0 ~ 127 (集成型 CPU) 或 0 ~ 255 (超薄型 CPU)。确保已编写用于 DJNZ 指令的标签编号的 LABEL 指令。当指定 S2 使用数据寄存器时，该标签的值为变量。当标签为变量时，确保用户程序中包含所有可能的 LABEL 编号。如果没有匹配的标签，将导致程序执行错误，打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED 。

11: 程序分支指令

示例：DJNZ 和 LABEL

以下示例演示了如何分别将连续的值 1000 ~ 1049 存储到数据寄存器 D100 ~ D149 的程序。



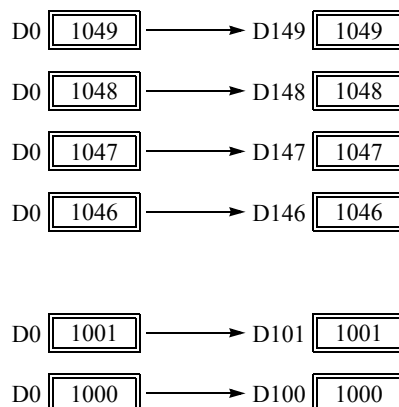
M8120 为初始化脉冲特殊内部继电器。
启动时 MOV 指令存储原始数据。
1049 → D0 用于存储第 1 次循环的值。
50 → D1 用于确定跳转循环。

在第 1 次循环中 IMOV 将 D0 数据 1049 传送至 D1049。

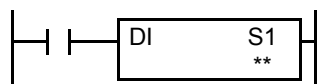
DEC 将 D0 数据递减至 1048。

DJNZ 跳转到标签 255 直到 D1 的值减至 0。

第 1 次循环：	D1	<input type="text" value="50"/>	目标：D99 + 50 = D149
第 2 次循环：	D1	<input type="text" value="49"/>	目标：D99 + 49 = D148
第 3 次循环：	D1	<input type="text" value="48"/>	目标：D99 + 48 = D147
第 4 次循环：	D1	<input type="text" value="47"/>	目标：D99 + 47 = D146
⋮			
第 49 次循环：	D1	<input type="text" value="2"/>	目标：D99 + 2 = D101
第 50 次循环：	D1	<input type="text" value="1"/>	目标：D99 + 1 = D100

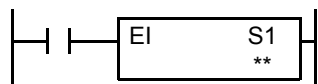


DI (禁用中断)



当输入打开时，将禁用源设备 S1 指定的中断输入和定时器中断。

EI (启用中断)



当输入打开时，将启用源设备 S1 指定的中断输入和定时器中断。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	中断输入和定时器中断	—	—	—	—	—	—	—	1-31	—

在启动 CPU 时，通常会启用功能设置中所选的中断输入 I2 ~ I5 和定时器中断。在执行 DI 指令时，即使满足 DI 指令后续用户程序区域内的中断条件，仍将禁用源设备 S1 指定的中断输入和定时器中断。在执行 EI 指令时，将再次启用 EI 指令后续用户程序区域内已禁用的源设备 S1 指定的中断输入和定时器中断。可以为 DI 和 EI 指令选择不同的设备以有选择的禁用和启用中断输入。有关中断输入和定时器中断，请参阅第 5-34 页和第 5-36 页 (基本卷)。

确保已在功能设置中选择源设备 S1 指定的中断输入和定时器中断。否则，在执行 DI 或 EI 指令时，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

不能在中断程序中使用 DI 和 EI 指令。否则，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

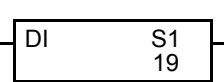
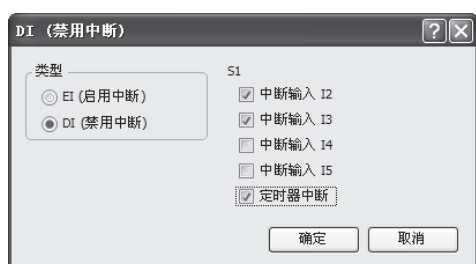
特殊内部继电器 M8140-M8144: 中断状态

特殊内部继电器 M8140 ~ M8144 以指示是否启用或禁用中断输入和定时器中断。

中断	中断已启用	中断已禁用
中断输入 I2	M8140 开	M8140 关
中断输入 I3	M8141 开	M8141 关
中断输入 I4	M8142 开	M8142 关
中断输入 I5	M8143 开	M8143 关
定时器中断	M8144 开	M8144 关

设置 WindLDR

在禁用中断 (DI) 或启用中断 (EI) 对话框中，单击中断输入 I2 ~ I5 或定时器中断左侧的复选框以选择源设备 S1。以下示例为 DI 指令选择中断输入 I2、I3 和定时器中断，源设备 S1 将显示为 19。



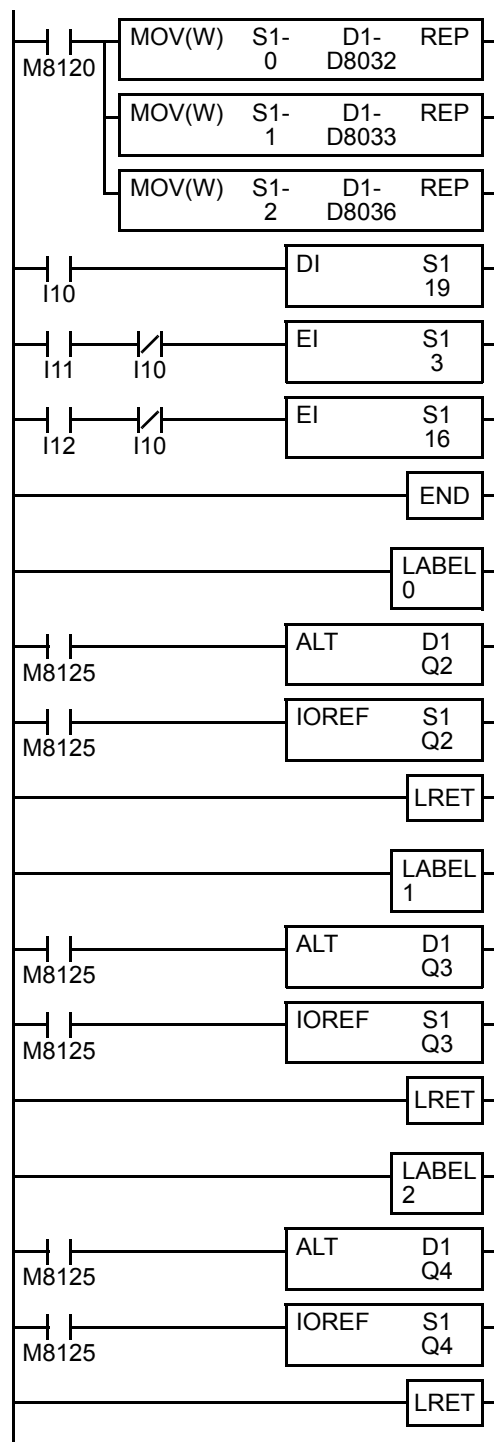
源设备 S1 显示为选择的中断输入和定时器中断总数。

中断	S1 值
中断输入 I2	1
中断输入 I3	2
中断输入 I4	4
中断输入 I5	8
定时器中断	16

11: 程序分支指令

示例 :DI 和 EI

以下示例演示了有选择的禁用和启用中断输入和定时器中断的程序。有关中断输入和定时器中断功能，请参阅第 5-34 页和第 5-36 页 (基本卷)。在此示例中，输入 I2 和 I3 将指定为中断输入，定时器中断的中断间隔时间为 100 ms。



M8120 是初始化脉冲特殊内部继电器。

D8032 存储标签编号 0 作为中断输入 I2 的跳转目标。

D8033 存储标签编号 1 作为中断输入 I3 的跳转目标。

D8036 存储标签编号 2 作为定时器中断的跳转目标。

当输入 I10 打开时，DI 将禁用中断输入 I2、I3 和定时器中断，然后关闭 M8140、M8141 和 M8144。

当输入 I11 打开而输入 I10 关闭时，EI 将启用中断输入 I2 和 I3，然后打开 M8140 和 M8141。

当输入 I12 打开而输入 I10 关闭时，EI 将启用定时器中断，然后打开 M8144。

主程序结束。

当输入 I2 打开时，程序执行将跳到标签 0。

M8125 是运行中输出特殊内部继电器。

ALT 打开或关闭输出 Q2 内存。

IOREF 立即将输出 Q2 内存状态写入实际输出 Q2。

程序执行返回主程序。

当输入 I3 打开时，程序执行跳转至标签 1。

M8125 是运行中输出特殊内部继电器。

ALT 打开或关闭输出 Q3 内存。

IOREF 立即将输出 Q3 内存状态写入实际输出 Q3。

程序执行返回主程序。

每 100 ms 出现一次定时器中断，然后程序执行跳转至标签 2。

M8125 是运行中输出特殊内部继电器。

ALT 打开或关闭输出 Q4 内存。

IOREF 立即将输出 Q4 内存状态写入实际输出 Q4。

程序执行返回主程序。

IOREF (I/O 刷新)



当输入打开时，不管扫描时间如何，都会刷新源设备 S1 指定的 1 位 I/O 数据。

当 I（输入）用作 S1 时，将立即读取实际输入状态至以 M300 开始的内部继电器，M300 已指定给 CPU 模块上的每个可用输入。

当 Q（输出）用作 S1 时，会立即将 RAM 中的输出数据写入 CPU 模块上的可用实际输出中。

当具有长扫描时间的用户程序中需要实时相应时，刷新指令是非常有用的。使用数据前在梯阶使用刷新指令时，刷新指令最有效。

IOREF 可与中断输入或定时器中断一起用于刷新数据。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	供刷新的 I/O	X	X	—	—	—	—	—	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

只可将 CPU 模块上提供的输入或输出编号指定为 S1。不能将扩展 I/O 模块的输入和输出编号指定为 S1。

输入设备编号和已指定内部继电器

输入设备	内部继电器	输入设备	内部继电器
I0	M300	I10	M310
I1	M301	I11	M311
I2	M302	I12	M312
I3	M303	I13	M313
I4	M304	I14	M314
I5	M305	I15	M315
I6	M306	I16	M316
I7	M307	I17	M317

正常执行用户程序时，在扫描结束时执行 END 指令的同时刷新 I/O 状态。当需要实时响应以执行中断时，可以使用 IOREF 指令。当打开输入至 IOREF 指令时，会立即读取或写入指定输入或输出的状态。

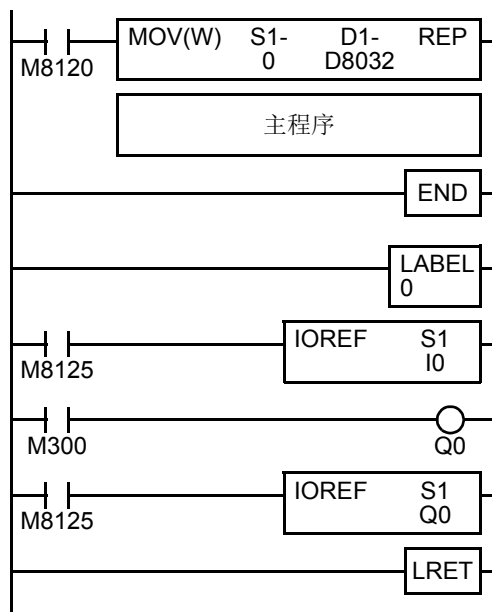
当执行 IOREF 指令以输入时，过滤器不生效，并读输入状态至相应的内部继电器中。

与在正常扫描中执行 END 指令一样，相同输入编号的实际输入状态将读取至内部输入内存中，然后筛选值将产生功能设置中指定的作用。请参阅第 5-42 页 (基本卷)。

11: 程序分支指令

示例 :IOREF

以下示例演示了使用 IOREF 指令传送输入 I0 状态至输出 Q0 的程序。输入 I2 已指定为中断输入。有关中断输入功能，请参阅第 5-34 页 (基本卷)。



M8120 是初始化脉冲特殊内部继电器。

D8032 存储 0 给中断输入 I2 指定跳转目标标签 0。

当输入 I2 打开时，程序执行将跳到标签 0。

M8125 是运行中输出特殊内部继电器。

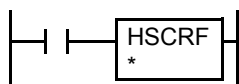
IOREF 立即将输入 I0 状态读取到内部继电器 M300。

M300 打开或关闭输出 Q0 内部内存。

另一个 IOREF 立即将输出 Q0 内部内存状态写入到实际的输出 Q0。

程序执行返回主程序。

HSCRF (高速计数器刷新)



当输入打开时，HSCRF 指令实时刷新特殊数据寄存器中的高速计数器当前值。

HSC1 ~ HSC4 四个高速计数器的当前值通常在每次扫描时进行更新。HSCRF 可以在您想读取更新的高速计数器当前值的梯形图中任意位置使用。

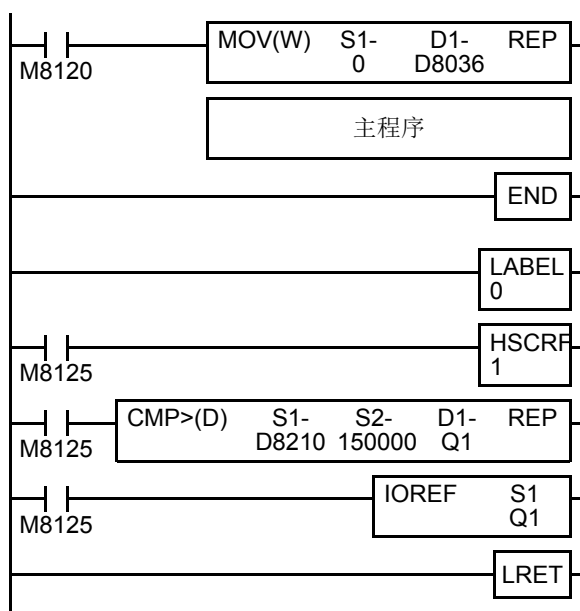
有关高速计数器功能，请参阅第 5-7 页 (基本卷)。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

示例 :HSCRF

以下示例演示了使用 HSCRF 更新高速计数器 HSC1 当前值的程序。有关定时器中断，请参阅第 5-36 页 (基本卷)。



M8120 是初始化脉冲特殊内部继电器。

D8036 将存储 0，以便将定时器中断的跳转目标标签指定为 0。

中断程序由 END 指令与主程序分隔开。

CPU 正在运行时，程序执行过程将按照在“功能设置”中选择的间隔重复跳转至标签 0。

M8125 是运行中输出特殊内部继电器。

HSCRF 更新数据寄存器 D8210 和 D8211 中的 HSC1 当前值。

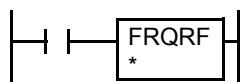
当 D8210/D8211 超过 150000，打开 Q1。

IOREF 立即将输出 Q0 内存状态写入实际输出 Q0。

每次中断程序完成后，程序执行过程都将在发生定时器中断的地址返回主程序。

11: 程序分支指令

FRQRF (频率测量刷新)



当输入打开时，FRQRF 指令实时刷新特殊数据寄存器中的频率测量值。

FRQRF 可以在您想读取频率测量值的梯形图中任意位置使用。

在测量结果存储到特殊内部继电器之前，使用最大的计算期间加一次扫描时间。不论输入频率是多少，在梯形图中使用 FRQRF 指令可在 250 ms 时间内读取频率测量最新值。

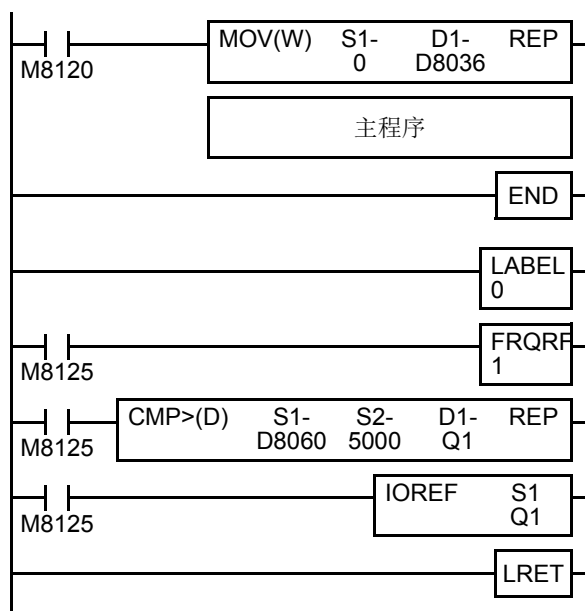
关于频率测量功能，请参阅第 5-30 页 (基本卷)。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

示例 :FRQRF

以下示例演示了使用 FRQRF 指令频率测量当前值的程序。有关定时器中断，请参阅第 5-36 页 (基本卷)。



M8120 是初始化脉冲特殊内部继电器。

D8036 将存储 0，以便将定时器中断的跳转目标标签指定为 0。

中断程序由 END 指令与主程序分隔开。

CPU 正在运行时，程序执行过程将按照在“功能设置”中选择的间隔重复跳转至标签 0。

M8125 是运行中输出特殊内部继电器。

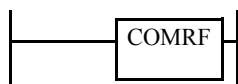
FRQRF 更新数据寄存器 D8060 和 D8061 中的 HSC1 频率测量值。

当 D8060/D8061 超过 5000 时，打开 Q1。

IOREF 立即将输出 Q0 内存状态写入实际输出 Q0。

每次中断程序完成后，程序执行过程都将在发生定时器中断的地址返回主程序。

COMRF (通信刷新)



COMRF 指令可实时刷新端口 3 至端口 7 上的扩展通信缓冲器上的发送与接收数据。缓冲器上的发送数据在 END 处理中通常被发送。缓冲器上的接收数据在 END 处理中通常被发送到 MicroSmart 设备。COMRF 指令可以用于需要立即执行 TXD 指令或更新比扫描时间更短的在节点上的接收数据的梯形图的任何位置。

注释: COMRF 指令不能用于中断程序中。如果使用，将导致用户程序执行错误，并打开 CPU 模块上的内部继电器 M8004 和 ERR LED。

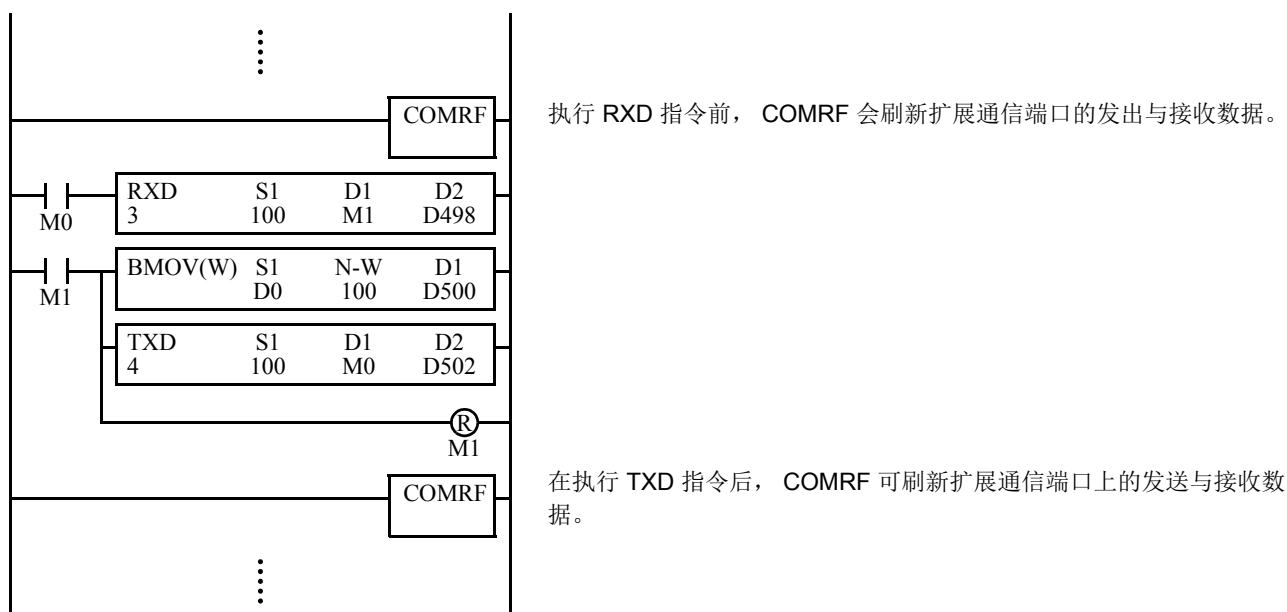
本指令适用于升级后的系统程序版本 110 或更高的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D FC5A-C24R2D	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

示例: COMRF

以下示例演示了在其他指令中执行 COMRF 指令的程序。



COMRF 执行时间

在扩展 RS232C/RS485 通信模块发送或接收通信期间，CPU 模块执行 COMRF 指令所需执行时间如下表所示。

扩展 RS232C/RS485 通信模块	COMRF 最大执行时间 (注释 1)
FC5A-SIF2	约 4 ms
FC5A-SIF4 (注释 2)	约 10 ms

注释 1: 数值为扩展 RS232C/RS485 通信模块进行通信时的最大执行时间。当多个扩展 RS232C/RS485 通信模块同时完成通信时，所需的执行时间为扩展 RS232C/RS485 通信模块数量 × COMRF 执行时间。

注释 2: 使用 FCA5-SIF4 扩展 RS485 通信模块时，CPU 模块必须使用 220 或以上版本的系统程序。

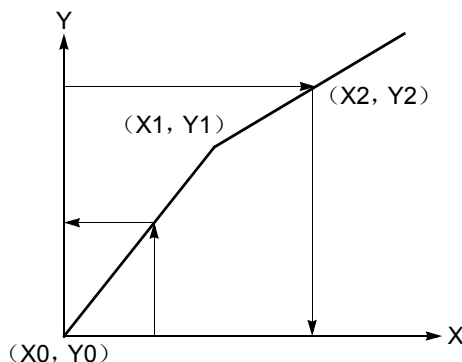
端口 3 至 7 的通信刷新

不使用 COMRF 指令时也可每 10 ms 自动对扩展通信寄存器进行更新。在“功能域设置”中的“端口 3 至端口 7 的“通信刷新”功能下选择“每 10 ms”。详情请参见第 5-43 页 (基本卷)。

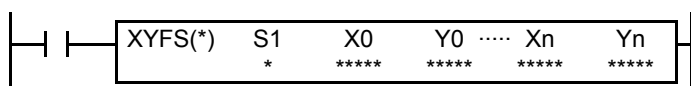
12: 坐标转换指令

简介

坐标转换指令使用 X 和 Y 的值之间的线性关系将一个数据点转换为另一个值。



XYFS (XY 格式设置)



当输入打开时，将设置 XY 转换格式。XY 坐标定义 X 和 Y 之间的线性关系。

CPU 模块	XY 坐标编号	n
集成型	2 - 5	$0 \leq n \leq 4$
超薄型	2 - 32	$0 \leq n \leq 31$

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	格式编号	—	—	—	—	—	—	—	0 - 5 (集成型 CPU) 0 - 29 (超薄型 CPU)	—
X0 - Xn	X 值	X	X	X	X	X	X	X	0 - 65535	—
Y0 - Yn	Y 值	X	X	X	X	X	X	X	0 - 65535 -32768 - 32767	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

在 T (定时器) 或 C (计数器) 用作 X0 至 Xn 或 Y0 至 Yn 中某个时，定时器 / 计数器的当前值 (TC 或 CC) 将被读出。

S1 (格式编号)

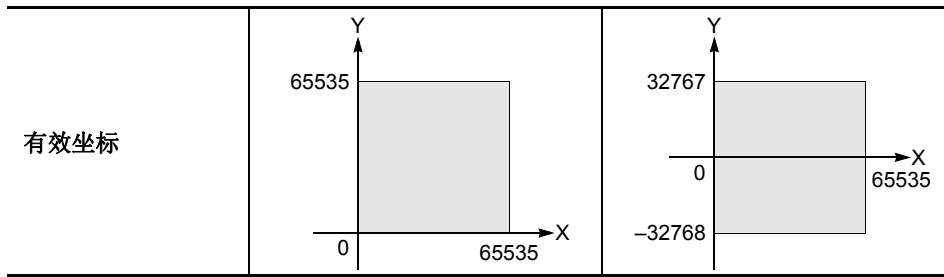
选择一个格式编号 0 - 5 (集成型 CPU) 或 0 - 29 (超薄型 CPU)。XY 转换最多可以设置 6 个或 30 个格式。

Xn (X 值), Yn (Y 值)

输入 X 和 Y 的坐标值。根据数据的类型，可以使用不同的数据范围。

数据类型	字	整数
Xn (X 值)	0 - 65535	0 - 65535
Yn (Y 值)	0 - 65535	-32768 - 32767

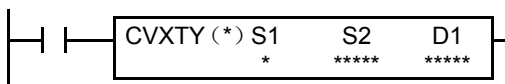
12: 坐标转换指令



有效数据类型

W (字)	X	当 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定为 Xn 或 Yn 时, 将使用 16 点。
I (整数)	X	
D (双字)	—	当 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备被指定为 Xn 或 Yn 时, 将使用 1 点。
L (长整数)	—	
F (浮点)	—	

CVXTY (X → Y 转换)



当输入打开时, 根据在 XYFS 指令中定义的线性关系将设备 S2 指定的 X 值转换为相应的 Y 值。设备 S1 会从最多 6 个 (集成型 CPU) 或 30 个 (超薄型 CPU) XY 转换格式中选择一个格式。转换结果设置为 D1 指定的设备。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	格式编号	—	—	—	—	—	—	—	0 - 5 (集成型 CPU) 0 - 29 (超薄型 CPU)	—
S2 (源 2)	X 值	X	X	X	X	X	X	X	0 - 65535	—
D1 (目标 1)	用于存储结果的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S2 时, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时, 数据将作为预置值 (TP 或 CP) 写入, 此值介于 0 到 65535 之间。

S1 (格式编号)

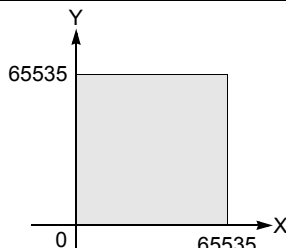
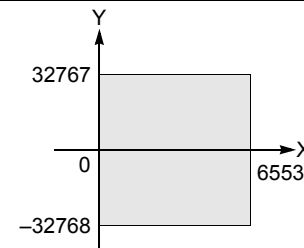
选择一个已经使用 XYFS 指令设置的格式编号 0 - 5 (集成型 CPU) 或 0 - 29 (超薄型 CPU)。若没有编写带相应格式编号的 XYFS 指令, 或同一格式编号的 XYFS 和 CVXTY 指令有不同的数据类型指定, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

S2 (X 值)

在 XYFS 指令指定的范围内, 输入要转换的 X 坐标值。

D1 (存储结果的目标)

Y 值的转换结果储存至该目标。

数据类型	字	整数
S2 (X 值)	0 - 65535	0 - 65535
D1 (Y 值)	0 - 65535	-32768 - 32767
有效坐标		

有效数据类型

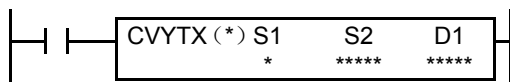
W (字)	X
I (整数)	X
D (双字)	—
L (长整数)	—
F (浮点)	—

当 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定为 S2 或 D1 时, 将使用 16 点。

当 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备被指定为 S2 或 D1 时, 将使用 1 点。

数据转换错误

数据转换错误为 ± 0.5 。

CVYTX (Y → X 转换)

当输入打开时, 根据在 XYFS 指令中定义的线性关系将设备 S2 指定的 Y 值转换为相应的 X 值。设备 S1 会从最多 6 个 (集成型 CPU) 或 30 个 (超薄型 CPU) XY 转换格式中选择一个格式。转换结果设置为 D1 指定的设备。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	格式编号	—	—	—	—	—	—	—	0 - 5 (集成型 CPU) 0 - 29 (超薄型 CPU)	—
S2 (源 2)	Y 值	X	X	X	X	X	X	X	0 - 65535 -32768 - 32767	—
D1 (目标 1)	用于存储结果的目标	—	X	▲	X	X	X	X	—	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S2 时, 定时器 / 计数器当前值 (TC 或 CC) 将被读出。在 T (定时器) 或 C (计数器) 用作 D1 时, 数据将作为预置值 (TP 或 CP) 写入, 此值介于 0 到 65535 之间。

S1 (格式编号)

选择一个已经使用 XYFS 指令设置的格式编号 0 - 5 (集成型 CPU) 或 0 - 29 (超薄型 CPU)。若没有编写带相应格式编号的 XYFS 指令, 或同一格式编号的 XYFS 和 CVYTX 指令有不同的数据类型指定, 将导致用户程序执行错误, 此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

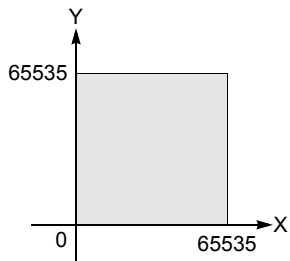
12: 坐标转换指令

S2 (Y 值)

在 XYFS 指令指定的范围内，输入要转换的 Y 坐标值。根据数据的类型，可以使用不同的数据范围。

D1 (存储结果的目标)

X 值的转换结果存储至该目标。

数据类型	字	整数
S2 (Y 值)	0 - 65535	-32768 - 32767
D1 (X 值)	0 - 65535	0 - 65535
有效坐标		

有效数据类型

W (字)	X
I (整数)	X
D (双字)	—
L (长整数)	—
F (浮点)	—

当 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定为 S2 或 D1 时，将使用 16 点。

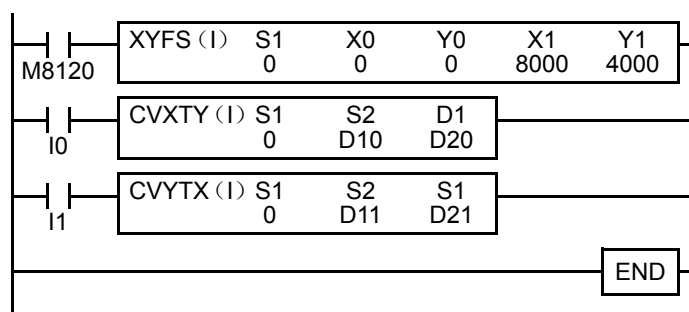
当 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备被指定为 S2 或 D1 时，将使用 1 点 (整数数据类型)。

数据转换错误

数据转换错误为 ± 0.5 。

示例：线性转换

以下示例演示了如何设置两个坐标点以定义 X 和 Y 之间的线性关系。两个点为 $(X0, Y0) = (0, 0)$ 和 $(X1, Y1) = (8000, 4000)$ 。设置好后，将有 $X \rightarrow Y$ 转换以及 $Y \rightarrow X$ 转换。

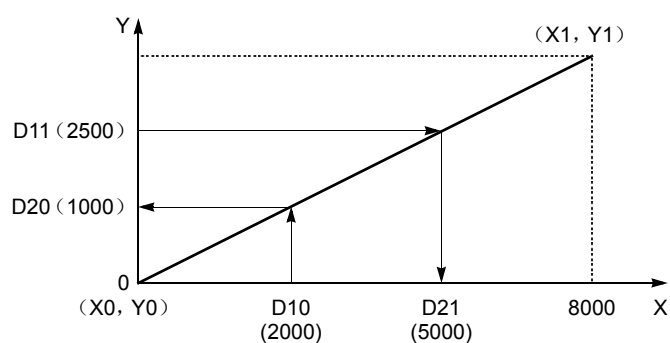


M8120 是初始化脉冲特殊内部继电器。

启动后，XYFS 会指定两个点。

当输入 I0 打开时，CVXTY 将转换 D10 中的值，并将结果存储至 D20 中。

当输入 I1 打开时，CVYTX 将转换 D11 中的值，并将结果存储至 D21 中。



该图显示两个点定义的线性关系：

$$Y = \frac{1}{2}X$$

如果寄存器 D10 中的值为 2000，则指定给 D20 的值为 1000。

对于 $Y \rightarrow X$ 转换，将使用以下等式：

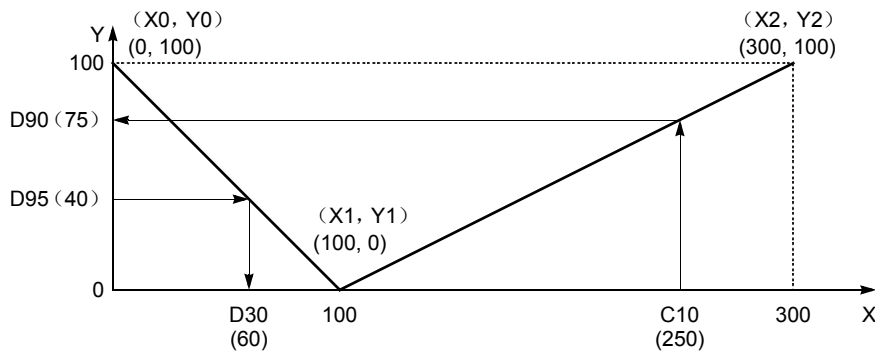
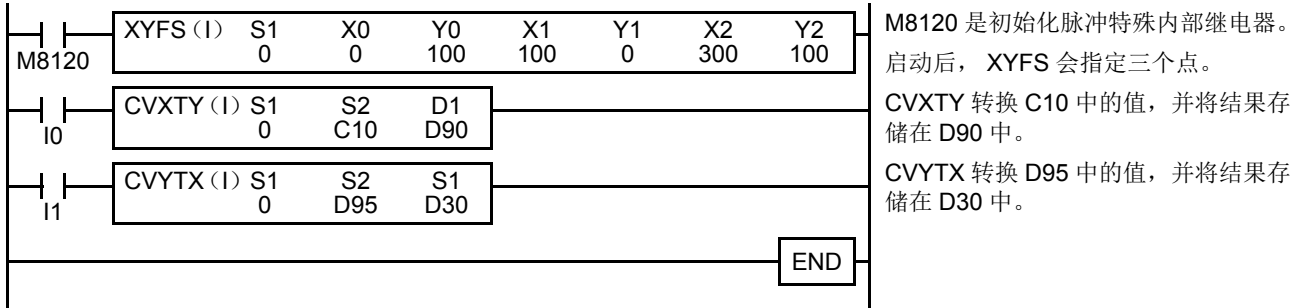
$$X = 2Y$$

如果寄存器 D11 中的值为 2500，则指定给 D21 的值为 5000。

12: 坐标转换指令

示例：重叠坐标

在该示例中，XYFS 指令设置三个坐标点，以定义 X 和 Y 之间的两种不同线性关系。这三个点为：(X0, Y0) = (0, 100)、(X1, Y1) = (100, 0) 和 (X2, Y2) = (300, 100)。这两个线段定义 X 的重叠坐标。即，对于指定范围内的每个 Y 值，都有两个对应的 X 值。



第一个线段定义 X → Y 转换的以下关系：

$$Y = -X + 100$$

第二个线段定义 X → Y 转换的另一个关系：

$$Y = \frac{1}{2}X - 50$$

对于 X → Y 转换，每个 X 值仅有一个相应的 Y 值。如果计数器 C10 的当前值为 250，则指定给 D90 的值为 75。

对于 Y → X 转换，XYFS 指令为每个 Y 值指定两个可能的 X 值。前两个点定义的关系在这些示例中有优先性。点 (X0, Y0) 和 (X1, Y1) 之间的线，即 (0, 100) 和 (100, 0) 之间的线可优先定义 Y → X 转换的关系 (X = -Y + 100)。

因此，如果寄存器 D95 中的值为 40，则指定给 D30 的值为 60，而不是 180。

XYFS 指令还可以定义同样两条线段，除了先指定点 (300, 100) 作为 (X0, Y0)，再定义点 (100, 0) 作为 (X1, Y1)。这时，该线性关系有优先性。

此时，如果寄存器 D95 中的值为 40，则指定给 D30 的值为 180，而不是 60。

AVRG (平均化)



当输入打开时，将根据设备 S2 和 S3 指定的取样条件对设备 S1 指定的取样数据进行处理。

当取样完成时，将平均值、最大值和最小值存储到以 D1 指定的设备起始的 3 个连续设备中，然后设备 D2 指定的取样完成输出将打开。

数据类型	W、I	D、L、F
平均化	D1	D1·D1+1
最大值	D1+1	D1+2·D1+3
最小值	D1+2	D1+4·D1+5

该指令对模拟量输入值的数据处理有效。在一个用户程序中最多可以编写 32 个 AVRG 指令。

适用的 CPU 模块

FC5A-C10R2/C	FC5A-C16R2/C	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	取样数据	X	X	X	X	X	X	X	—	—
S2 (源 2)	取样结束输入	X	X	X	X	—	—	—	—	—
S3 (源 3)	取样周期 (扫描时间)	X	X	X	X	X	X	X	0-65535	—
D1 (目标 1)	用来存储结果的第一个设备编号	—	—	—	—	—	—	X	—	—
D2 (目标 2)	抽样完成输出	—	X	▲	—	—	—	—	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可以将内部继电器 M0 - M2557 指定为 D2。特殊内部继电器不能分配给 D2。

在 T (定时器) 或 C (计数器) 用作 S1 或 S3 时，定时器 / 计数器当前值 (TC 或 CC) 将被读出。

当选择 F (浮点) 数据类型时，只有数据寄存器可指定为 S1 和 D1。

当输入打开时，将在每个扫描中执行 AVRG 指令。当设备 S3 指定的取样周期 (扫描时间) 数量是 1 - 65535 时，在每次扫描中对设备 S1 指定的取样数据进行处理。当指定的取样周期完成时，取样数据平均值设置到 D1 (W 或 I 数据类型) 或 D1·D1+1 (D、L 或 F 数据类型)。取样数据的最大值设置到下一个设备 D1+1 (W 或 I 数据类型) 或 D1+2·D1+3 (D、L 或 F 数据类型)。取样数据的最小值设置到下一个设备 D1+2 (W 或 I 数据类型) 或 D1+4·D1+5 (D、L 或 F 数据类型)。设备 D2 指定的取样完成输出打开。

当设备 S3 指定的取样周期 (扫描时间) 数量是 0 时，取样在 AVRG 指令输入打开时开始，在设备 S2 指定的取样结束输入打开时停止。然后，平均值、最大值和最小值将被设置到以 D1 指定的设备起始的 3 个设备。

当取样超过 65535 次时，在此点的平均值、最大值和最小值将被设置到以 D1 指定的设备起始的 3 个设备，并且继续取样。

当设备 S3 指定的取样次数没有完成之前取样结束输入打开时，取样会结束，并且在此点的结果将设置到以 D1 指定的设备起始的 3 个设备。

平均值计算到最小的整数，对小数部分四舍五入。

当没有使用取样结束输入时，指定一个内部继电器或其他有效设备作为源设备 S2 的空位。

当选择 F (浮点) 数据类型，并且 S1 不符合标准浮点格式时，将导致程序执行错误，打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。当发生错误时，将跳过错误的 S1 数据。根据正确的 S1 数据计算平均值、最大值和最小值，并设置到以 D1 指定的设备起始的 3 个设备。

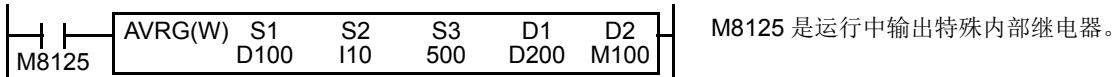
12: 坐标转换指令

有效数据类型

W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源, 使用 16 点 (字或整数数据类型) 或 32 点 (双字或长整数数据类型)。
I (整数)	X	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源, 使用 1 点 (字或整数数据类型) 或 2 点 (双字、长整数或浮点数据类型)。
L (长整数)	X	
F (浮点)	X	

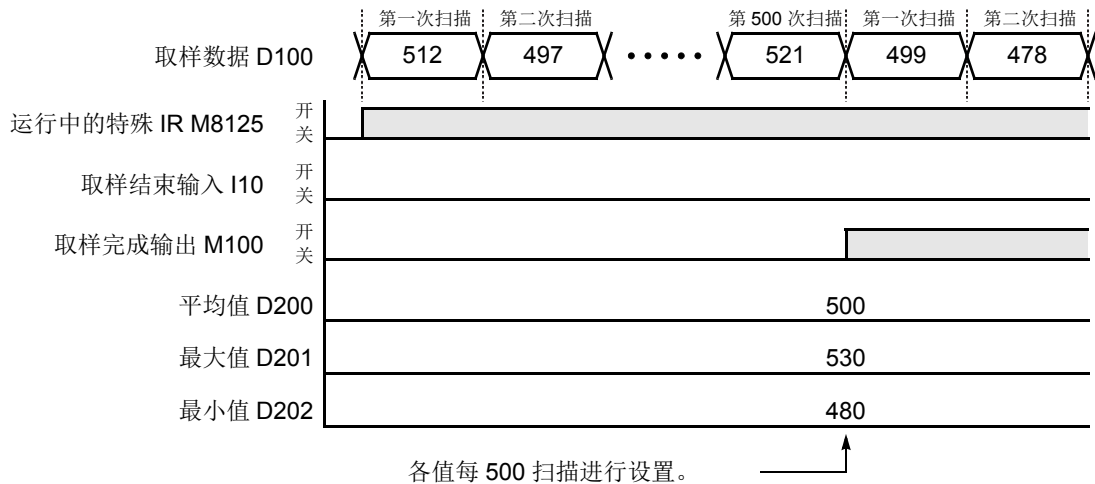
示例 : AVRГ

下面的示例演示的是程序在每 500 次扫描中计算数据寄存器 D100 的平均值并将结果存储到数据寄存器 D200。



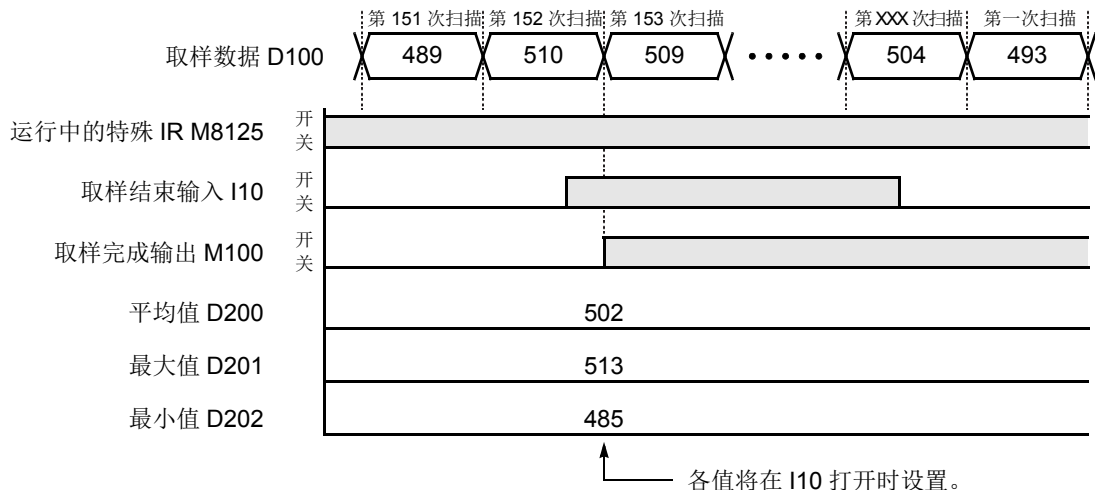
当取样结束输入没有打开时

当取样结束输入 I10 关闭时, 在每 500 次扫描中计算平均值、最大值和最小值, 并分别将结果存储到数据寄存器 D200、D201 和 D202。每隔 500 次扫描设置取样完成输出 M100。



当取样结束打开时

当取样结束输入 I10 打开时, 此点的平均值、最大值和最小值将分别存储到数据寄存器 D200、D201 和 D202。同时设置取样完成输出 M100。当取样结束输入 I10 关闭时, 在第一次扫描开始时恢复取样。



13: 脉冲指令

简介

PULS（脉冲输出）指令用于生成 10 Hz - 100 kHz 的脉冲输出，以控制简单位置控制应用程序的脉冲电机。

PWM（脉宽调制）指令用于生成 14.49、45.96 或 367.65 Hz 的脉冲输出，脉宽比率在 0% 和 100% 之间变化以进行照明控制。

RAMP 指令用于台形控制。

ZRN 指令用于零返回控制。

除了 PULS3、PWM3、RAMP2 和 ZRN3 指令不能在 FC5A-D16RK1 和 FC5A-D16RS1 上使用外，PULS、PWM、RAMP 和 ZRN 指令可以在所有超薄型 CPU 模块上使用。

指令		PULS	PWM	RAMP	ZRN
脉冲输出端口	Q0	PULS1	PWM1	RAMP1	ZRN1
	Q1	PULS2	PWM2		ZRN2
	Q2	PULS3	PWM3	RAMP2	ZRN3
	Q3	—	—		—
输出频率		10 Hz - 100 kHz	14.49 Hz、 45.96 Hz、 367.65 Hz	10 Hz - 100 kHz	10 Hz - 100 kHz
脉宽比率		50%	0 - 100%	50%	50%
脉冲计数		PULS1 PULS3	PWM1 PWM3	RAMP2	—
预置值		1 - 100,000,000	1 - 100,000,000	1 - 100,000,000	—
频率更改时间		—	—	10 - 10,000 ms	—
减速输入	高速	—	—	—	I2、I3、I4、I5
	正常	—	—	—	I0、I1、I6 - I627、 M0 - M2557

13: 脉冲指令

PULS1 (脉冲输出 1)



当输入打开时，PULS1 指令发出来自 Q0 的脉冲输出。输出脉冲频率由源设备 S1 决定。输出脉宽比率固定在 50%。

可以编写 PULS1 以生成预定数量的输出脉冲。若禁用脉冲计数，当 PULS1 指令的起始输入仍然打开时，PULS1 会生成输出脉冲。

PULS2 (脉冲输出 2)



当输入打开时，PULS2 指令发出来自 Q1 的脉冲输出。输出脉冲频率由源设备 S1 决定。输出脉宽比率固定在 50%。

当 PULS2 指令的起始输入仍然打开时，PULS2 将生成输出脉冲。无法编写 PULS2 以生成预定数量的输出脉冲。

PULS3 (脉冲输出 3)



当输入打开时，PULS3 指令发出来自 Q2 的脉冲输出。输出脉冲频率由源设备 S1 决定。输出脉宽比率固定在 50%。

可编写 PULS3 生成预定数量的输出脉冲。当禁用脉冲计数时，PULS3 生成输出脉冲，期间 PULS3 指令的起始输入保持打开。

不适用于 FC5A-16RK1/RS1

注释: PULS1、PULS2 和 PULS3 指令在用户程序中只能使用一次。当没有使用 PULS1、PULS2 或 PULS3 时，未使用的输出 Q0、Q1 或 Q2 可用于另一个脉冲指令或普通输出。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	—	X (PULS1 和 PULS2)	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X	—	—
D1 (目标 1)	状态继电器	—	—	X	—	—	—	—	—	—

源设备 S1 (控制寄存器) 使用设备指定以 S1 为起始编号的 8 个数据寄存器。可以将数据寄存器 D0 - D1992、D2000 - D7992 和 D10000 - D49992 指定为 S1。有关详细信息，请参阅下文。

目标设备 D1 (状态继电器) 使用以设备以 D1 为起始编号的 3 个内部继电器。可以将内部继电器 M0 - M2550 指定为 D1。指定为 D1 的内部继电器编号的最低有效数字必须是 0，否则 PULS 指令不能正确工作。不能将特殊内部继电器指定为 D1。有关详细信息，请参阅第 6-2 页 (基本卷)。

源设备 S1 (控制继电器)

根据需要，在执行 PULS 指令前，将合适的值存储至以 S1 指定的设备开始的数据寄存器中，并确保该值在有效范围内。设备 S1+5 ~ S1+7 为只读。

设备	功能	说明	R/W
S1+0	动作模式	0: 10 Hz - 1 kHz 1: 100 Hz - 10 kHz 2: 1 kHz - 100 kHz 3: 200 Hz - 100 kHz (注释 2)	R/W
S1+1	输出脉冲频率	当 S1+0 (动作模式) = 0 - 2 时: 1 - 100 (%) (所选模式 S1+0 的最高频率的 1% - 100%) 当 S1+0 (动作模式) = 3 时: 20 - 10,000 (× 10 Hz) (注释 3)	R/W
S1+2	脉冲计数	0: 禁用脉冲计数 1: 启用脉冲计数 (仅 PULS1/PULS3)	R/W

设备	功能	说明	R/W
S1+3	预置值 (高位字)	1 - 100,000,000 (05F5 E100h) (仅 PULS1/PULS3)	R/W
S1+4	预置值 (低位字)		
S1+5	当前值 (高位字)	1 - 100,000,000 (05F5 E100h) (仅 PULS1/PULS3)	R
S1+6	当前值 (低位字)		
S1+7	错误状态	0 - 5	R

注释 1: 设备的高位和低位字可以在升级后的系统程序版本 110 或更高的 CPU 模块中进行交换。请参阅第 5-46 页 (基本卷)。

注释 2: 对于 FC5A-D12K1E 和 FC5A-D12S1E, 模式 3 的频率范围为 250Hz - 100 kHz。

注释 3: 对于 FC5A-D12K1E 和 FC5A-D12S1E, 模式 3 的频率范围为 25 - 10,000 (x10 Hz)。

S1+0 动作模式

存储在设备 S1+0 指定的数据寄存器中的值可确定脉冲输出的频率范围。

- 0: 10 Hz - 1 kHz
- 1: 100 Hz - 10 kHz
- 2: 1 kHz - 100 kHz
- 3: 200 Hz - 100 kHz

S1+1 输出脉冲频率

当 S1+0 设为 0 - 2 时, 存储在设备 S1+1 指定的数据寄存器中的值将以 S1+0 选定的频率范围最大值的百分比指定脉冲输出频率。设备 S1+1 的有效值为 1 - 100, 因此输出脉冲频率可以分别为 10 Hz - 1 kHz (动作模式 0)、100 Hz - 10 kHz (动作模式 1) 或 1 kHz - 100 kHz (动作模式 2)。

当 S1+0 设为 3 (200 Hz - 100 kHz) 时, 设备 S1+1 的有效值为 20 - 10,000, 将 S1+1 值乘以 10 可确定输出脉冲频率, 因此输出脉冲频率可以按 10 Hz 的增量进行设置。输出频率错误最大为 ±5%。

动作模式 (S1+0)	S1+1	输出脉冲频率 (Hz)
0 - 2	1 - 100	S1+0 选择的最大频率 × S1+1 值 (%)
3	20 - 10,000	S1+1 值 × 10

S1+2 脉冲计数

脉冲计数仅可用于 PULS1 和 PULS3 指令。当启用脉冲计数时, PULS1 或 PULS3 将生成由设备 S1+3 和 S1+4 指定的预定数量的输出脉冲。禁用脉冲计数时, 在 PULS 指令的起始输入保持打开时, PULS1、PULS2 或 PULS3 将生成输出脉冲。

- 0: 禁用脉冲计数
- 1: 启用脉冲计数 (仅 PULS1/PULS3)

编写 PULS2 时, 将 0 存储至 S1+2 指定的数据寄存器中。

S1+3 预置值 (高位字)

S1+4 预置值 (低位字)

按如上所述启用脉冲计数时, PULS1 或 PULS3 将生成由设备 S1+3 和 S1+4 指定的预定数量的输出脉冲。预置值可以为 1 - 100,000,000 (05F5 E100h), 这些值将存储在由 S1+3 (高位字) 和 S1+4 (低位字) 指定的两个连续数据寄存器中。

当禁用 PULS1 或 PULS3 脉冲计数或编写 PULS2 时, 会将 0 存储至 S1+3 和 S1+4 指定的数据寄存器中。

S1+5 当前值 (高位字)

S1+6 当前值 (低位字)

在禁用脉冲计数时执行 PULS1 或 PULS3 指令, 输出脉冲计数将存储在由设备 S1+5 (高位字) 和 S1+6 (低位字) 指定的两个连续数据寄存器中。当前值可以为 1 - 100,000,000 (05F5 E100h), 并在每次扫描时进行更新。

13: 脉冲指令

S1+7 错误状态

当打开 PULS 指令的起始输入时，将校验设备值。若在设备值中发现错误，则错误代码将存储在设备 S1+7 指定的数据寄存器中。

错误代码	说明
0	正常
1	动作模式设置错误 (S1+0 存储 0 - 3 以外的值)
2	输出脉冲频率设置错误 (S1+1 的存储值不在输出脉冲频率范围内。)
3	脉冲计数设置错误 (S1+2 存储 0 和 1 以外的值)
4	预置值设置错误 (S1+3 和 S1+4 存储 1 - 100,000,000 以外的值)
5	PULS2 的无效脉冲计数设置 (S1+2 存储 1)

目标设备 D1 (状态继电器)

以 D1 指定的设备开始的三个内部继电器指示 PULS 指令的状态。这些设备为只读。

设备	功能	说明	R/W
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	R
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	R
D1+2	脉冲输出溢出	0: 未出现溢出 1: 出现溢出 (仅 PULS1/PULS3)	R

D1+0 脉冲输出打开

在 PULS 指令生成输出脉冲时，设备 D1+0 指定的内部继电器保持打开。当 PULS 指令的起始输入关闭，或者 PULS1 或 PULS3 指令完全生成预定数量的输出脉冲时，设备 D1+0 指定的内部继电器关闭。

D1+1 脉冲输出完成

当 PULS1 或 PULS3 指令完全生成预定数量的输出脉冲，或 PULS 指令停止生成输出脉冲时，设备 D1+1 指定的内部继电器打开。当 PULS 指令的起始输入打开时，设备 D1+1 指定的内部继电器关闭。

D1+2 脉冲输出溢出

当 PULS1 或 PULS3 指令生成超过预定数量的输出脉冲时，设备 D1+2 指定的内部继电器打开。当 PULS 指令的起始输入打开时，设备 D1+2 指定的内部继电器关闭。

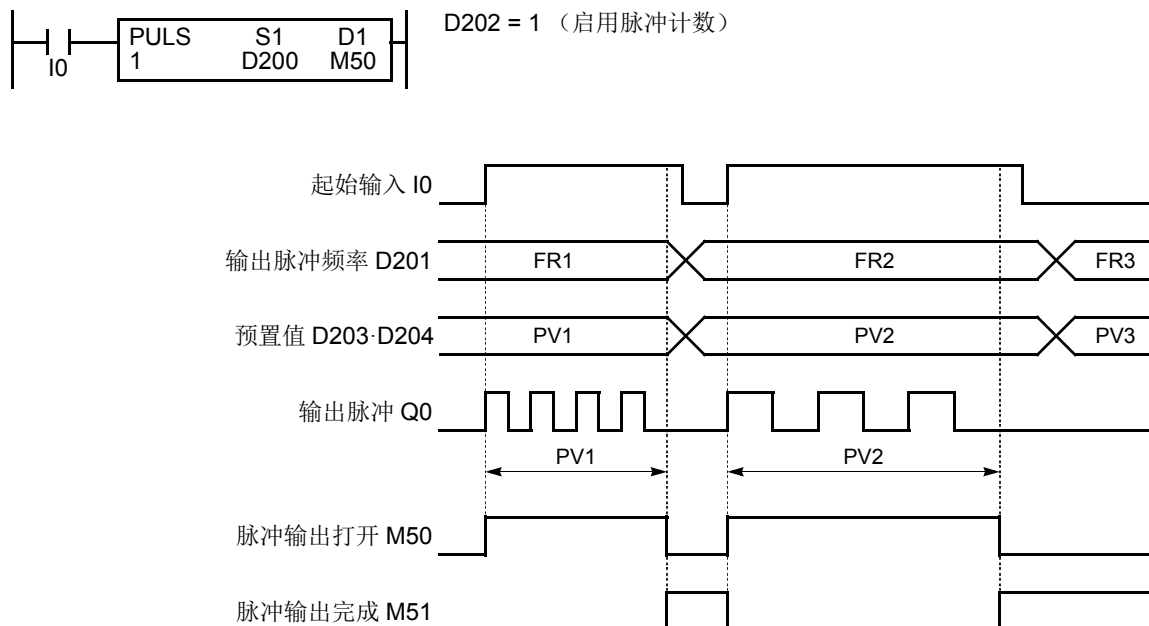
用于脉冲输出的特殊数据寄存器

三个附加特殊数据寄存器可存储脉冲输出的当前频率。

设备地址	功能	说明
D8055	当前脉冲频率 PULS1 或 RAMP1 (Q0)	在执行 PULS1 或 RAMP1 指令时，D8055 存储输出 Q0 的当前脉冲频率。 每次扫描时都会更新该值。
D8056	当前脉冲频率 PULS2 或 RAMP1 (Q1)	在执行 PULS2 或 RAMP1 (方向控制双脉冲输出) 指令时，D8056 存储输出 Q1 的当前脉冲频率。 每次扫描时都会更新该值。
D8059	当前脉冲频率 PULS3 或 RAMP2 (Q2)	在执行 PULS3 或 RAMP2 指令时，D8059 存储输出 Q2 的当前脉冲频率。 每次扫描时都会更新该值。

启用脉冲计数时序

该程序演示启用脉冲计数时 PULS1 指令的时序。

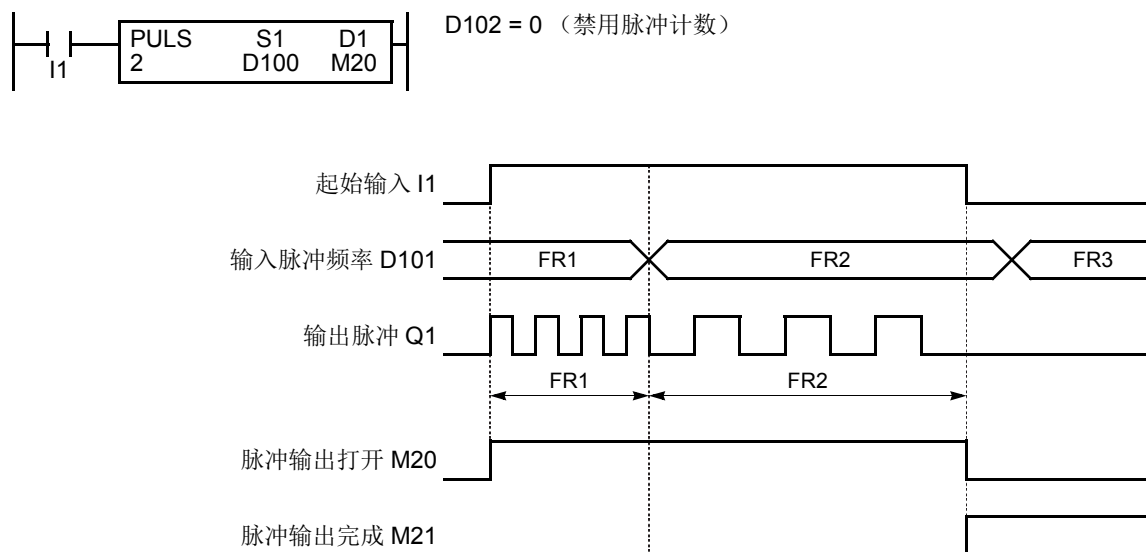


- 当输入 I0 打开时，PULS1 开始生成数据寄存器 D201 中存储的值指定频率的输出脉冲。当输出 Q0 发出输出脉冲时，内部继电器 M50 保持打开。
- 当生成的输出脉冲数量达到数据寄存器 D203 和 D204 指定的预置值时，PULS1 将停止生成输出脉冲。然后内部继电器 M50 关闭，内部继电器 M51 打开。
- 如果在生成输出脉冲时 D201 中的输出脉冲频率发生变更，则该变更会影响下一个扫描。当更改脉冲频率时，确保变更定时大大慢于输出脉冲频率，以使脉冲频率变更成功。
- 如果在达到预置值之前关闭输入 I0，则 PULS1 会立即停止生成输出脉冲，然后将关闭内部继电器 M50，并打开内部继电器 M51。

13: 脉冲指令

禁用脉冲计数时序

该程序演示未启用脉冲计数时 PULS2 指令的时序。



- 当输入 I1 打开时，PULS2 开始生成数据寄存器 D101 中存储的值指定频率的输出脉冲。当输出 Q1 发出输出脉冲时，内部继电器 M20 保持打开。
- 当输入 I1 关闭时，PULS2 会立即停止生成输出脉冲，然后将关闭内部继电器 M20，并打开内部继电器 M21。
- 如果在生成输出脉冲时 D101 中的输出脉冲频率发生变更，则该变更会影响下一个扫描。当更改脉冲频率时，确保变更定时大大慢于输出脉冲频率，以使脉冲频率变更成功。

示例程序 : PULS1

该程序演示 PULS1 指令的用户程序，可生成 5,000 个来自输出 Q0 的 200 Hz 频率的脉冲，然后再生成 60,000 个 500 Hz 频率的脉冲。

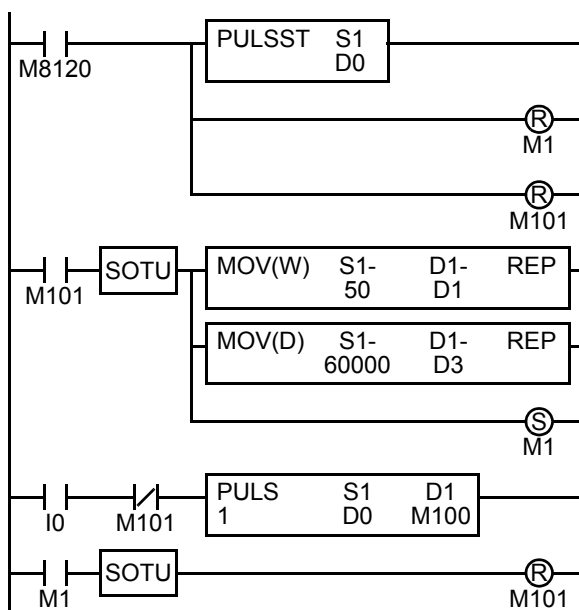
设置 WindLDR

在 WindLDR 编辑屏幕上，将光标插入您想插入脉冲指令宏的地方，并键入 **PULSST**。如下所示输入参数。



设备设置

设备	功能	说明	设备地址 (值)
S1+0	动作模式	频率范围 200 Hz - 100 kHz	D0 (3)
S1+1	输出脉冲频率	200 Hz	D1 (20)
S1+2	脉冲计数	启用脉冲计数	D2 (1)
S1+3	预置值 (高位字)	5,000	D3/D4 (5000)
S1+4	预置值 (低位字)		
S1+5	当前值 (高位字)	0 ~ 60,000	D5/D6
S1+6	当前值 (低位字)		



M8120 是初始化脉冲特殊内部继电器。

当启动 CPU 时，PULSST 宏在第一级指定脉冲输出的参数。

脉冲数据更新标志 M1 复位 (脉冲数据未更新)。

脉冲输出完成标志 M101 已关闭。

当 M101 打开时，两个 MOV 指令将第二级参数存储至数据寄存器 D1、D3 和 D4。

D1 (输出脉冲频率): 50 (500 Hz)

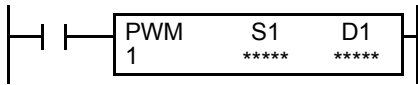
D3/D4 (预置值) 60,000

脉冲数据更新标志 M1 设置 (脉冲数据未更新)。

当起始输入 I0 打开时，PULS1 开始在第一级 200Hz 生成 5,000 输出脉冲。

脉冲输出完成 M101 关闭。

PWM1 (脉宽调制 1)



当输入打开时， PWM1 指令将生成脉冲输出。输出脉冲频率可以为 14.49、45.96 或 367.65 Hz，输出脉宽比率由源设备 S1 确定。

PWM1 从输出 Q0 发出输出脉冲。

可以编写 PWM1 以生成预定数量的输出脉冲。当禁用脉冲计数，而 PWM1 指令的起始输入仍然打开时， PWM1 会生成输出脉冲。

PWM2 (脉宽调制 2)

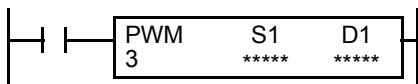


当输入打开时， PWM2 指令将生成脉冲输出。输出脉冲频率可以为 14.49、45.96 或 367.65 Hz，输出脉宽比率由源设备 S1 确定。

PWM2 从输出 Q1 发出输出脉冲。

当 PWM2 指令的起始输入仍然打开时， PWM2 将生成输出脉冲。可以编写 PWM2 以生成预定数量的输出脉冲。

PWM3 (脉宽调制 3)



当输入打开时， PWM3 指令将生成脉冲输出。输出脉冲频率可以为 14.49、45.96 或 367.65 Hz，输出脉宽比率由源设备 S1 确定。

PWM3 从输出 Q2 发出输出脉冲。

可以编写 PWM3 以生成预定数量的输出脉冲。当禁用脉冲计数，而 PWM3 指令的起始输入仍然打开时， PWM3 会生成输出脉冲。

不适用于 FC5A-16RK1/RS1

注释: PWM1、 PWM2 和 PWM3 指令在用户程序中只能使用一次。当没有使用 PWM1、 PWM2 或 PWM3 时，未使用的输出 Q0、 Q1 或 Q2 可用于另一个脉冲指令或普通输出。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	—	X (PWM1 和 PWM2)	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X	—	—
D1 (目标 1)	状态继电器	—	—	X	—	—	—	—	—	—

源设备 S1 (控制寄存器) 使用设备指定以 S1 为起始编号的 8 个数据寄存器。可以将数据寄存器 D0 - D1992、 D2000 - D7992 和 D10000 - D49992 指定为 S1。有关详细信息，请参阅下文。

目标设备 D1 (状态继电器) 使用以设备以 D1 为起始编号的 3 个内部继电器。可以将内部继电器 M0 - M2550 指定为 D1。指定为 D1 的内部继电器编号的最低有效数字必须是 0，否则 PWM 指令不能正确工作。不能将特殊内部继电器指定为 D1。有关详细信息，请参阅第 6-2 页 (基本卷)。

源设备 S1（控制继电器）

根据需要，在执行 PMW 指令前，将合适的值存储至以 S1 指定的设备开始的数据寄存器中，并确保该值在有效范围内。设备 S1+5 - S1+7 为只读。

设备	功能	说明	R/W
S1+0	输出脉冲频率	FC5A-D16RK1/RS1, FC5A-D32K3/S3 0 : 11.44 Hz 1 : 45.78 Hz 2 : 366.2 Hz FC5A-D12K1E/S1E 0 : 15.26 Hz 1 : 61.04 Hz 2 : 488.3 Hz	R/W
S1+1	脉宽比率	1 - 100 (由输出脉冲频率 S1+0 决定的周期的 1% - 100%)	R/W
S1+2	脉冲计数	0: 禁用脉冲计数 1: 启用脉冲计数 (仅 PWM1/PWM3)	R/W
S1+3	预置值 (高位字)	1 - 100,000,000 (05F5 E100h) (仅 PWM1/PWM3)	R/W
S1+4	预置值 (低位字)		
S1+5	当前值 (高位字)	1 - 100,000,000 (05F5 E100h) (仅 PWM1/PWM3)	R
S1+6	当前值 (低位字)		
S1+7	错误状态	0 - 5	R

注释：设备的高位和低位字可以在升级后的系统程序版本 110 或更高版本的 CPU 模块中进行交换。请参阅第 5-46 页（基本卷）。

S1+0 输出脉冲频率

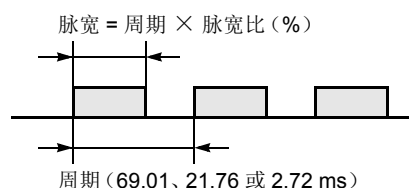
存储在设备 S1+0 指定的数据寄存器中的值可指定脉冲输出频率。

FC5A-D16RK1/RS1, FC5A-D32K3/S3	FC5A-D12K1E/S1E
0: 11.44 Hz (69.01 ms 周期)	0: 15.26 Hz
1: 45.78 Hz (21.76 ms 周期)	1: 61.04 Hz
2: 366.2 Hz (2.72 ms 周期)	2: 488.3 Hz

S1+1 脉宽比率

存储在设备 S1+1 指定的数据寄存器中的值将以 S1+0 选定的输出脉冲频率决定的周期百分比指定脉冲输出的脉宽比率。设备 S1+1 的有效值为 1 - 100。

$$\begin{aligned} \text{脉宽} &= \text{周期} \times \text{脉宽比} / 100 \\ &= 1 / \text{输出脉冲频率} \times \text{脉宽比} / 100 \end{aligned}$$



S1+2 脉冲计数

脉冲计数仅可用于 PWM1 和 PWM3 指令。当启用脉冲计数时，PWM1 或 PWM3 将生成由设备 S1+3 和 S1+4 指定的预定数量的输出脉冲。当禁用脉冲计数，而 PWM 指令的起始输入保持打开时，PWM 将生成输出脉冲。

- 0: 禁用脉冲计数
- 1: 启用脉冲计数 (仅 PWM1/PWM3)

编写 PWM2 时，将 0 存储至 S1+2 指定的数据寄存器中。

S1+3 预置值 (高位字)

S1+4 预置值 (低位字)

按如上所述启用脉冲计数时，PWM1 或 PWM3 将生成由设备 S1+3 和 S1+4 指定的预定数量的输出脉冲。预置值可以为 1 - 100,000,000 (05F5 E100h)，这些值将存储在由 S1+3 (高位字) 和 S1+4 (低位字) 指定的两个连续数据寄存器中。

当禁用 PWM1 或 PWM3 脉冲计数或编写 PWM2 时，会将 0 存储至 S1+3 和 S1+4 指定的数据寄存器中。

13: 脉冲指令

S1+5 当前值 (高位字)

S1+6 当前值 (低位字)

在执行 PWM1 或 PWM3 指令时，输出脉冲计数将存储在由设备 S1+5 (高位字) 和 S1+6 (低位字) 指定的两个连续数据寄存器中。当前值可以为 1 - 100,000,000 (05F5 E100h)，并在每次扫描时进行更新。

S1+7 错误状态

当打开 PWM 指令的起始输入时，将校验设备值。若在设备值中发现错误，则错误代码将存储在设备 S1+7 指定的数据寄存器中。

错误代码	说明
0	正常
1	输出脉冲频率设置错误 (S1+0 存储 0 - 2 以外的值)
2	脉宽比率设置错误 (S1+1 存储 1 - 100 以外的值)
3	脉冲计数设置错误 (S1+2 存储 0 和 1 以外的值)
4	预置值设置错误 (S1+3 和 S1+4 存储 1 - 100,000,000 以外的值)
5	PWM2 的无效脉冲计数设置 (S1+2 存储 1)

目标设备 D1 (状态继电器)

以 D1 指定的设备开始的三个内部继电器指示 PWM 指令的状态。这些设备为只读。

设备	功能	说明	R/W
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	R
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	R
D1+2	脉冲输出溢出	0: 未出现溢出 1: 出现溢出 (仅 PWM1/PWM3)	R

D1+0 脉冲输出打开

在 PWM 指令生成输出脉冲时，设备 D1+0 指定的内部继电器保持打开。当 PWM 指令的起始输入关闭，或者 PWM1 或 PWM3 指令生成预定数量的输出脉冲时，设备 D1+0 指定的内部继电器关闭。

D1+1 脉冲输出完成

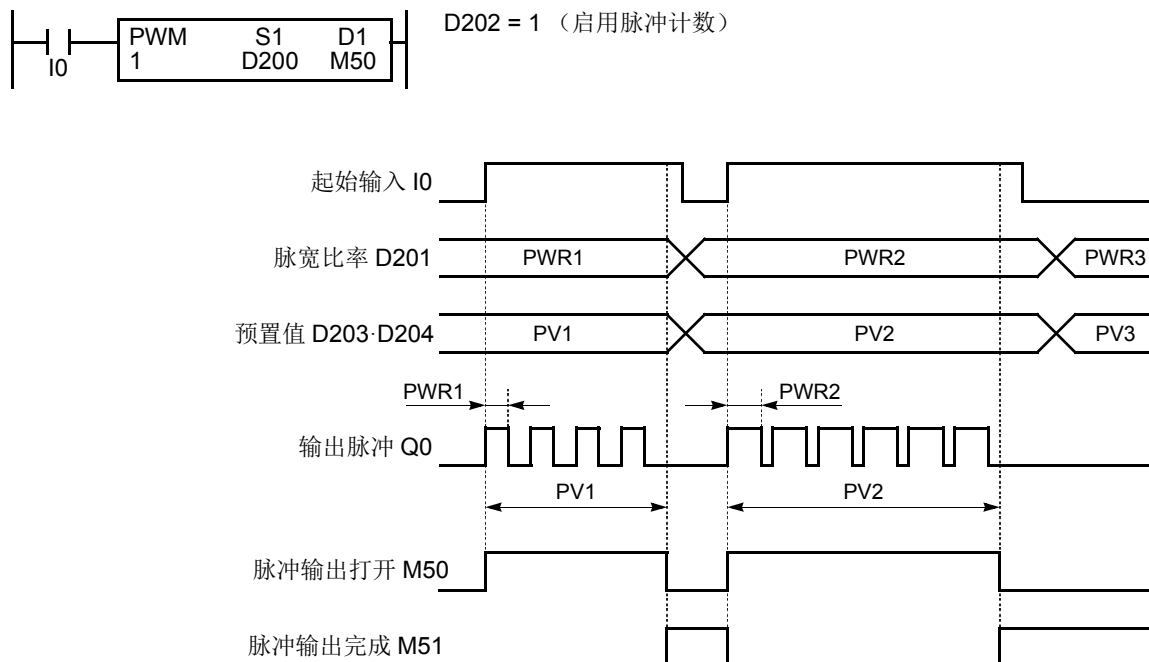
当 PWM1 或 PWM3 指令完全生成预定数量的输出脉冲，或 PWM 指令停止生成输出脉冲时，设备 D1+1 指定的内部继电器打开。当 PWM 指令的起始输入打开时，设备 D1+1 指定的内部继电器关闭。

D1+2 脉冲输出溢出

当 PWM1 或 PWM3 指令生成超过预定数量的输出脉冲时，设备 D1+2 指定内部继电器打开。当 PWM 指令的起始输入打开时，设备 D1+2 指定的内部继电器关闭。

启用脉冲计数时序

该程序演示启用脉冲计数时 PWM1 指令的时序。

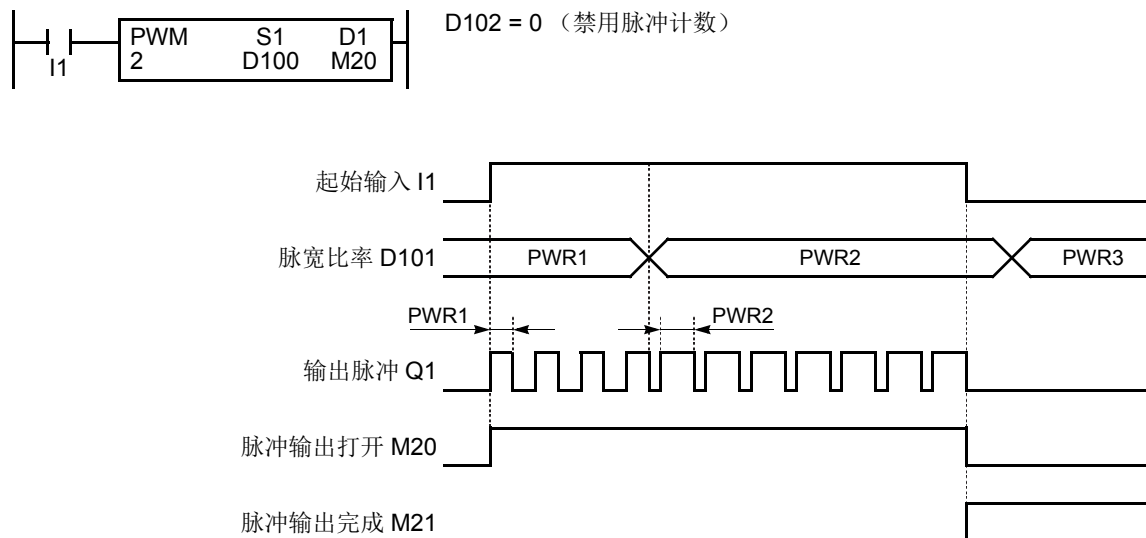


- 当输入 I0 打开时，PWM1 开始生成数据寄存器 D200 中存储的值指定频率的输出脉冲。脉宽由数据寄存器 D201 中存储的值决定。当输出 Q0 发出输出脉冲时，内部继电器 M50 保持打开。
- 当生成的输出脉冲数量达到数据寄存器 D203 和 D204 指定的预置值时，PWM1 将停止生成输出脉冲。然后内部继电器 M50 关闭，内部继电器 M51 打开。
- 如果在生成输出脉冲时 D201 中的脉宽比率值发生变更，则该变更会影响下一个扫描。当更改脉宽比率时，确保变更定时大大慢于输出脉冲频率，以使脉宽比率变更成功。
- 如果在达到预置值之前关闭输入 I0，则 PWM1 会立即停止生成输出脉冲，然后将关闭内部继电器 M50，并打开内部继电器 M51。

13: 脉冲指令

禁用脉冲计数时序

该程序演示未启用脉冲计数时 PWM2 指令的时序。



- 当输入 I1 打开时，PWM2 开始生成数据寄存器 D100 中存储的值指定频率的输出脉冲。脉宽由数据寄存器 D101 中存储的值决定。当输出 Q1 发出输出脉冲时，内部继电器 M20 保持打开。
- 当输入 I1 关闭时，PWM2 会立即停止生成输出脉冲，然后将关闭内部继电器 M20，并打开内部继电器 M21。
- 如果在生成输出脉冲时 D101 中的脉宽比率值发生变更，则该变更会影响下一个扫描。当更改脉宽比率时，确保变更定时大大慢于输出脉冲频率，以使脉宽比率变更成功。

示例程序 : PWM2

该程序演示 PWM2 指令的用户程序从输出 Q1 生成脉冲，当输入 I0 关闭时，开 / 关比率为 30%，当输入 I0 打开时，开 / 关比率为 60%。

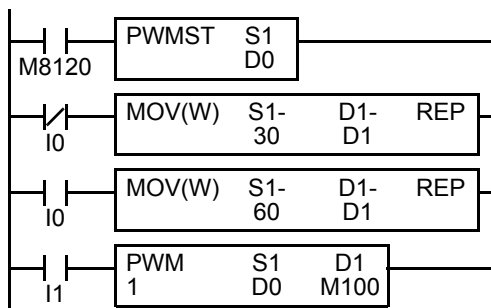
设置 WindLDR

在 WindLDR 编辑屏幕上，将光标插入您想插入脉冲指令宏的地方，并键入 **PWMST**。如下所示输入参数。



设备设置

设备	功能	说明	设备地址 (值)
S1+0	输出脉冲频率	367.65 Hz	D0 (2)
S1+1	脉宽比率	30%	D1 (30)
S1+2	脉冲计数	禁用脉冲计数	D2 (0)
S1+3	预置值 (高位字)	未使用	D3
S1+4	预置值 (低位字)		D4
S1+5	当前值 (高位字)	未使用	D5
S1+6	当前值 (低位字)		D6
S1+7	错误状态		D7
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	M100
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	M101
D1+2	脉冲输出溢出	0: 未出现溢出 1: 出现溢出 (仅 PWM1/PWM3)	M102



M8120 是初始化脉冲特殊内部继电器。

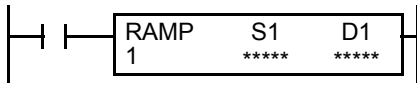
当启动 CPU 时，PWMST 宏在第一级指定脉冲输出的参数。

当输入 I0 关闭时，D1 (脉宽比率) 存储 30 (30%)。

当输入 I0 打开时，D1 (脉宽比率) 存储 60 (60%)。

当输入 I1 打开时，PWM2 分别根据输入 I0 是打开还是关闭，以从输出 Q1 生成 30% 或 60% 脉宽比率的输出脉冲。

RAMP1 (台形控制 1)



当输入打开时，RAMP1 指令从输出 Q0 发出预定数量的输出脉冲。输出脉冲以源设备 S1 决定的梯形图变化。在启动 RAMP1 指令后，输出脉冲频率会线性增加至预定常量值，并保持常量为该值一段时间，然后线性减少至初始值。

可以选择频率更改比率或频率更改时间以加快和降低变化速度。

当输入关闭时，脉冲输出保持关闭。当再次打开输入时，RAMP1 指令将开始生成新的输出脉冲循环。

RAMP1 也可以用于方向控制来生成方向控制输出或自输出 Q1 的反向输出脉冲。

注释：RAMP1 指令在用户程序中只能使用一次。当使用 RAMP1，而禁用方向控制时，未使用的输出 Q1 可用于另一个脉冲指令 PULS2、PWM2 或 ZRN2 或普通输出。

RAMP2 (台形控制 2)



不适用于 FC5A-16RK1/RS1

当输入打开时，RAMP2 指令从输出 Q2 发出预定数量的输出脉冲。输出脉冲以源设备 S1 决定的梯形图变化。在启动 RAMP2 指令后，输出脉冲频率会线性增加至预定常量值，并保持常量为该值一段时间，然后线性减少至初始值。

可以选择频率更改比率或频率更改时间以加快和降低变化速度。

当输入关闭时，脉冲输出保持关闭。当再次打开输入时，RAMP2 指令将开始生成新的输出脉冲循环。

RAMP2 不能用于可逆控制。

注释：RAMP2 指令在用户程序中只能使用一次。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	—	X (RAMP1)	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X	—	—
D1 (目标 1)	状态继电器	—	—	X	—	—	—	—	—	—

源设备 S1 (控制寄存器) 使用设备指定以 S1 为起始编号的 11 个数据寄存器。可以将数据寄存器 D0-D1989、D2000-D7989 和 D10000-D49989 指定为 S1。有关详细信息，请参阅下文。

目标设备 D1 (状态继电器) 使用以设备以 D1 为起始编号的 4 个内部继电器。可以将内部继电器 M0 - M2550 指定为 D1。指定为 D1 的内部继电器编号的最低有效数字必须是 0，否则 RAMP 指令不能正确工作。不能将特殊内部继电器指定为 D1。有关详细信息，请参阅第 6-2 页 (基本卷)。

源设备 S1（控制继电器）

根据需要，在执行 RAMP 指令前，将合适的值存储至以指定 S1 的设备开始的数据寄存器中，并确保该值在有效范围内。设备 S1+8 - S1+10 为只读。

设备	功能	说明	R/W
S1+0	动作模式	0: 10 Hz - 1 kHz 1: 100 Hz - 10 kHz 2: 1 kHz - 100 kHz 3: 200 Hz - 100 kHz（注释 2）	R/W
S1+1	目标频率	当 S1+0（动作模式）= 0 - 2 时：1 - 100（%） （所选模式 S1+0 的最高频率的 1% - 100%） 当 S1+0（动作模式）= 3 时：20 - 10,000（×10 Hz）（注释 3）	R/W
S1+2	启动频率	当 S1+0（动作模式）= 0 - 2 时：1 - 100（%） （所选模式 S1+0 的最高频率的 1% - 100%） 当 S1+0（动作模式）= 3 时：20 - 10,000（×10 Hz）（注释 3）	R/W
S1+3	频率更改比率	当 S1+0（动作模式）= 0 - 2 时：1 - 100（%） （所选模式 S1+0 的最高频率的 1% - 100%）	R/W
	频率更改时间	当 S1+0（动作模式）= 3 时：10 - 10,000 ms （指定以 10 为增量）	
S1+4	启用方向控制	0: 禁用方向控制 1: 方向控制（单脉冲输出） 2: 方向控制（双脉冲输出）（仅 RAMP1）	R/W
S1+5	控制方向	0: 正向 1: 反向	R/W
S1+6	预置值（高位字）	1 - 100,000,000（05F5 E100h）	R/W
S1+7	预置值（低位字）		
S1+8	当前值（高位字）	1 - 100,000,000（05F5 E100h）	R
S1+9	当前值（低位字）		
S1+10	错误状态	当 S1+0（动作模式）= 0 - 2 时：0 - 10 当 S1+0（动作模式）= 3 或 4 时：0 - 9	R

注释 1: 设备的高位和低位字可以在升级后的系统程序版本 110 或更高的 CPU 模块中进行交换。请参阅第 5-46 页（基本卷）。

注释 2: 对于 FC5A-D12K1E 和 FC5A-D12S1E，模式 3 的频率范围为 250Hz - 100 kHz。

注释 3: 对于 FC5A-D12K1E 和 FC5A-D12S1E，模式 3 的频率范围为 25 - 10,000（x10 Hz）。

S1+0 动作模式

存储在设备 S1+0 指定的数据寄存器中的值可确定脉冲输出的频率范围。

- 0: 10 Hz - 1 kHz
- 1: 100 Hz - 10 kHz
- 2: 1 kHz - 100 kHz
- 3: 200 Hz - 100 kHz

S1+1 目标频率

当 S1+0 设为 0 - 2 时，存储在设备 S1+1 指定的数据寄存器中的值将以 S1+0 选定的频率范围最大值的百分比指定目标输出频率。设备 S1+1 的有效值为 1 - 100，因此输出脉冲频率可以分别为 10 Hz - 1 kHz（动作模式 0）、100 Hz - 10 kHz（动作模式 1）或 1 kHz - 100 kHz（动作模式 2）。

当 S1+0 设为 3（200 Hz - 100 kHz）时，设备 S1+1 的有效值为 20 - 10,000（以 10 为增量），将 S1+1 值乘以 10 可确定目标脉冲频率，因此输出脉冲频率可以按 10 Hz 的增量进行设置。输出频率错误最大为 ±5%。

动作模式（S1+0）	S1+1	目标频率（Hz）
0 - 2	1 - 100	S1+0 选择的最大频率 × S1+1 值（%）
3	20 - 10,000	S1+1 值 × 10

13: 脉冲指令

S1+2 启动频率

当 S1+0 设为 0 - 2 时，存储在设备 S1+2 指定的数据寄存器中的值将以 S1+0 选定的频率范围最大值的百分比指定启动脉冲输出频率。设备 S1+2 的有效值为 1 - 100，因此启动脉冲频率可以分别为 10 Hz - 1 kHz（动作模式 0）、100 Hz - 10 kHz（动作模式 1）或 1 kHz - 100 kHz（动作模式 2）。

当 S1+0 设为 3（200 Hz - 100 kHz）时，设备 S1+2 的有效值为 20 - 10,000（以 10 为增量），将 S1+2 值乘以 10 可确定启动脉冲频率，因此启动脉冲频率可以按 10 Hz 的增量进行设置。输出频率错误最大为 ±5%。

动作模式 (S1+0)	S1+2	启动频率 (Hz)
0 - 2	1 - 100	S1+0 选择的最大频率 × S1+2 值 (%)
3	20 - 10,000	S1+2 值 × 10

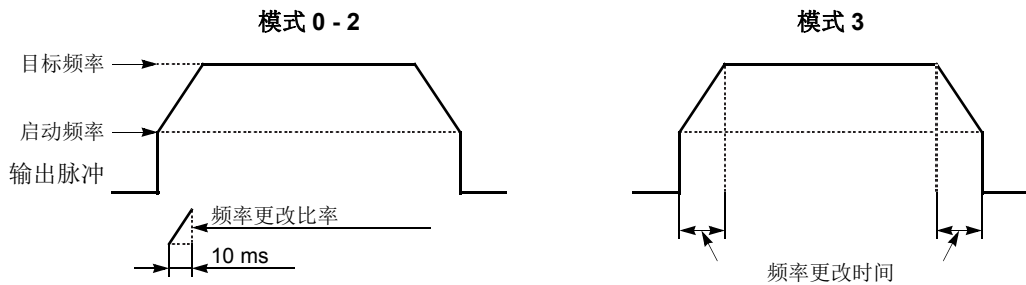
S1+3 频率更改比率 / 频率更改时间

当 S1+0 设为 0 - 2 时，存储在设备 S1+3 指定的数据寄存器中的值将在 10 ms 期间以 S1+0 选定的频率范围最大值的百分比确定脉冲输出频率的更改比率作数 S1+3 的有效值为 1 - 100，因此频率的更改比率可以分别为 10 Hz - 1 kHz（动作模式 0）、100 Hz - 10 kHz（动作模式 1）或 1 kHz - 100 kHz（动作模式 2）。

当 S1+0 设为 3 时，存储在设备 S1+3 指定的数据寄存器中的值可直接确定频率更改时间。有效值为 10 - 10,000，增量为 10，因此频率更改时间可以为 10 - 10,000 ms。最低位数字将被省略。

动作模式	频率更改比率 / 频率更改时间	
0 - 2	10 ms (Hz) 的频率更改比率	S1+0 选择的最大频率 (Hz) × S1+3 值 (%)
3	频率更改时间 (ms)	S1+3 选择的频率更改时间 (ms)



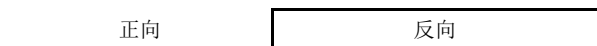


对于梯形频率更改图的加速和减速周期可以使用相同的频率更改比率和频率更改时间。



S1+4 启用方向控制

存储在设备 S1+4 指定的数据寄存器中的值可指定一个输出模式。

RAMP1 可为 S1+4 指定 0 - 2，而 RAMP2 可以指定 0 和 1。

S1+4 值	方向控制	说明
0	方向控制 禁用	<p>输出 Q0 或 Q2 生成输出脉冲；用于单向控制。</p> <p>输出 Q0/Q2 </p> <p>输出 Q1 可用于 PULS2、PWM2、ZRN2 或普通输出。 当使用 RAMP2 时，输出 Q3 可用于普通输出。</p>
1	方向控制 (单脉冲输出)	<p>输出 Q0 或 Q2 生成输出脉冲，而输出 Q1 或 Q3 生成方向控制信号。</p> <p>输出 Q0/Q2 </p> <p>输出 Q1/Q3 </p> <p>根据设备 S1+5（控制方向）指定的数据寄存器中存储的值打开或关闭输出 Q1 或 Q3: 0 为正向或 1 为反向。</p>
2 (仅 RAMP1)	方向控制 (双脉冲输出)	<p>输出 Q0 生成正向输出脉冲，而输出 Q1 生成反向输出脉冲。</p> <p>输出 Q0 (正向) </p> <p>输出 Q1 (反向) </p> <p>根据设备 S1+5（控制方向）指定的数据寄存器中存储的值，输出 Q0 或 Q1 交替生成输出脉冲: 0 为正向或 1 为反向。</p>

如果 RAMP 指令的起始输入打开后，设备 S1+4 指定的数据寄存器中存储的值更改，则该更改仅在 CPU 重启后生效。

S1+5 控制方向

当 S1+4 设为 1 或 2 以启用方向控制时，设备 S1+5 指定的数据寄存器中存储的值指定控制方向。

0: 正向

1: 反向

S1+6 预置值 (高位字)**S1+7 预置值 (低位字)**

RAMP1 或 RAMP2 指令生成由设备 S1+6 和 S1+7 预定数量的输出脉冲。预置值可以为 1 - 100,000,000 (05F5 E100h)，这些值将存储在由 S1+6 (高位字) 和 S1+7 (低位字) 指定的两个连续数据寄存器中。

S1+8 当前值 (高位字)**S1+9 当前值 (低位字)**

在执行 RAMP1 或 RAMP2 令生成输出脉冲时，输出脉冲计数将存储在由设备 S1+8 (高位字) 和 S1+9 (低位字) 指定的两个连续数据寄存器中。当前值可以为 1 - 100,000,000 (05F5 E100h)，并在每次扫描时进行更新。

13: 脉冲指令

S1+10 错误状态

当打开 RAMP 指令的起始输入时，将校验设备值。若在设备值中发现错误，则错误代码将存储在设备 S1+10 指定的数据寄存器中。

错误代码	说明
0	正常
1	操作模式指定错误 (S1+0 存储 0 - 3 以外的值)
2	启动频率设置错误 (S1+2 的存储值不在启动脉冲输出频率范围内。)
3	预置值设置错误 (S1+6 和 S1+7 存储 1 - 100,000,000 以外的值)
4	稳定频率设置错误 (S1+1 的存储值不在目标脉冲输出频率范围内。)
5	频率更改比率设置错误 模式 0 - 2: S1+3 存储 1 - 100 以外的值 模式 3: S1+3 存储 10 - 10,000 以外的值
6	方向控制启用设置错误 (S1+4 存储 0 - 2 以外的值)
7	控制方向设置错误 (S1+5 存储 0 和 1 以外的值)
8	在频率更改区域利用目标频率 (S1+1)、启动频率 (S1+2) 和频率更改比率 / 次数 (S1+3) 计算出的脉冲数量超过了总输出脉冲的预置值 (S1+6/7)。 要更正此错误，请减少目标频率 (S1+1) 或启动频率 (S1+2) 的值，或增加频率更改比率 / 次数 (S1+3)。
9	启动频率 (S1+2) 大于或等于目标频率 (S1+1)。 降低启动频率 (S1+2) 至小于目标频率 (S1+1) 的值。
10	模式 0 - 2: 频率更改比率 (S1+3) 大于启动频率 (S1+2) 和目标频率 (S1+1) 之间的差。降低频率更改比率 (S1+3) 或启动频率 (S1+2)。

目标设备 D1（状态继电器）

以 D1 指定的设备开始的四个内部继电器指示 RAMP 指令的状态。这些设备为只读。

设备	功能	说明	R/W
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	R
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	R
D1+2	脉冲输出状态	0: 目标脉冲输出 1: 更改输出脉冲频率	R
D1+3	脉冲输出溢出	0: 未出现溢出 1: 出现溢出	R

D1+0 脉冲输出打开

在 RAMP 指令生成输出脉冲时，设备 D1+0 指定的内部继电器保持打开。当 RAMP 指令的起始输入关闭或 RAMP 指令生成预定数量的输出脉冲时，设备 D1+0 指定的内部继电器关闭。

D1+1 脉冲输出完成

当 RAMP 指令生成预定数量的输出脉冲，或 RAMP 指令停止生成输出脉冲时，设备 D1+1 指定的内部继电器打开。当 RAMP 指令的起始输入打开时，设备 D1+1 指定的内部继电器关闭。

D1+2 脉冲输出状态

当输出脉冲频率增加或降低时，设备 D1+2 指定的内部继电器打开，当输出脉冲频率达到目标频率（S1+1）时，该内部继电器关闭。当脉冲输出关闭时，设备 D1+2 指定的内部继电器保持关闭。

D1+3 脉冲输出溢出

当 RAMP 指令生成超过预定数量的输出脉冲（S1+6/7）时，设备 D1+3 指定的内部继电器打开。当出现溢出时，当前值（S1+8/9）停止在预置值（S1+6/7）处。当 RAMP 指令的起始输入打开时，设备 D1+3 指定的内部继电器关闭。

用于脉冲输出的特殊数据寄存器

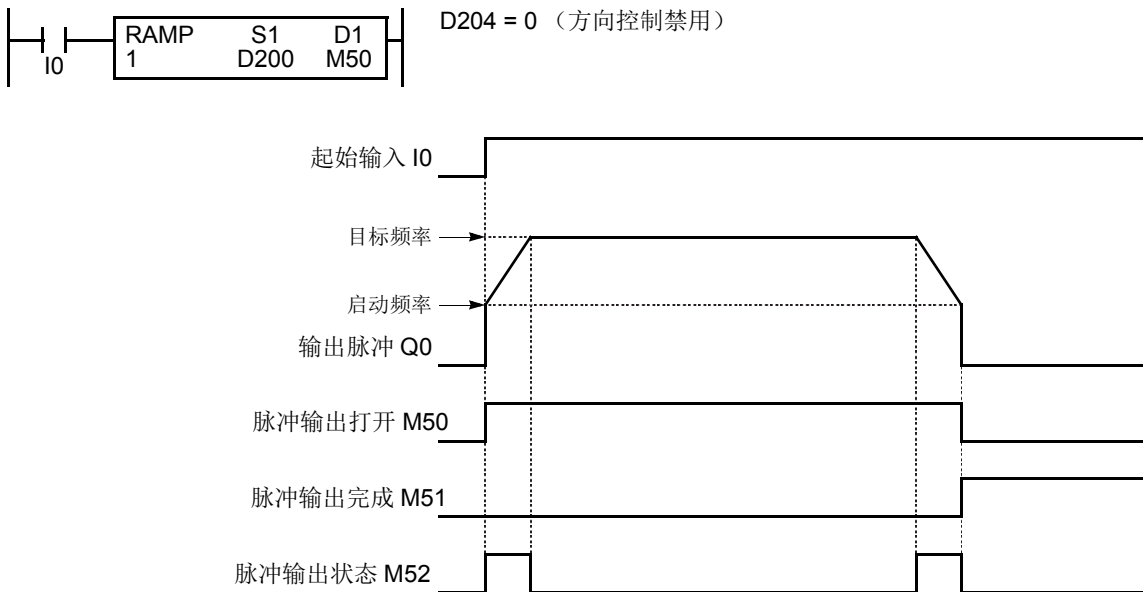
三个附加特殊数据寄存器可存储脉冲输出的当前频率。

设备地址	功能	说明
D8055	当前脉冲频率 PULS1 或 RAMP1 (Q0)	在执行 PULS1 或 RAMP1 指令时，D8055 存储输出 Q0 的当前脉冲频率。每次扫描时都会更新该值。
D8056	当前脉冲频率 PULS2 或 RAMP1 (Q1)	在执行 PULS2 或 RAMP1（方向控制双脉冲输出）指令时，D8056 存储输出 Q1 的当前脉冲频率。每次扫描时都会更新该值。
D8059	当前脉冲频率 PULS3 或 RAMP2 (Q2)	在执行 PULS3 或 RAMP2 指令时，D8059 存储输出 Q2 的当前脉冲频率。每次扫描时都会更新该值。

13: 脉冲指令

方向控制禁用时序

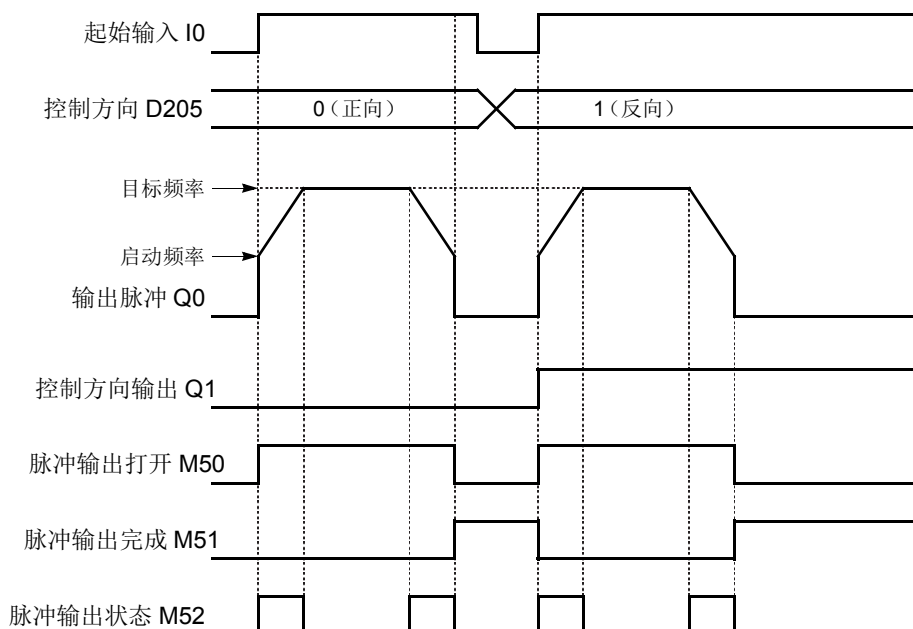
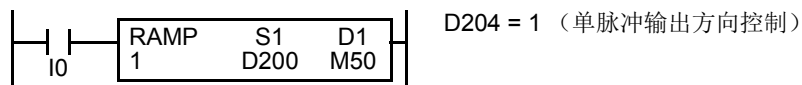
该程序演示禁用方向控制时 RAMP1 指令的时序。



- 当输入 I0 打开时，RAMP1 生成以数据寄存器 D202 中存储的值指定的启动频率开始的输出脉冲。当输出 Q0 发出输出脉冲时，内部继电器 M50 保持打开。
- 动作模式 0 - 2: 脉冲频率按照数据寄存器 D203 中存储的频率更改比率增加。
- 动作模式 3: 脉冲频率会在数据寄存器 D203 中存储的频率更改时间内一直增加。
- 在输出脉冲频率增加时，内部继电器 M52 保持打开。
- 当输出脉冲频率达到数据寄存器 D201 中存储的值指定的目标频率时，内部继电器 M52 关闭。当输出脉冲频率开始降低时，内部继电器 M52 再次打开。
- 当生成的输出脉冲数量达到数据寄存器 D206 和 D207 指定的预置值时，RAMP1 将停止生成输出脉冲。然后内部继电器 M50 和 M52 关闭，内部继电器 M51 打开。
- 如果在生成输出脉冲时 D200 - D207 (除了 D204) 中的参数值更改，当起始输入 I0 开始下一个循环时更改生效。
- 如果在起始输入 I0 打开时 D204 中存储的值更改，则仅在 CPU 重启后更改才会生效。
- 如果在达到预置值之前关闭起始输入 I0，则 RAMP1 会立即停止生成输出脉冲，然后将关闭内部继电器 M50，并打开内部继电器 M51。当再次打开输入 I0 时，RAMP1 重新启动以生成下一循环的输出脉冲，该循环从启动频率开始。

单脉冲输出方向控制时序

该程序演示启用单脉冲输出方向控制时 RAMP1 指令的时序。

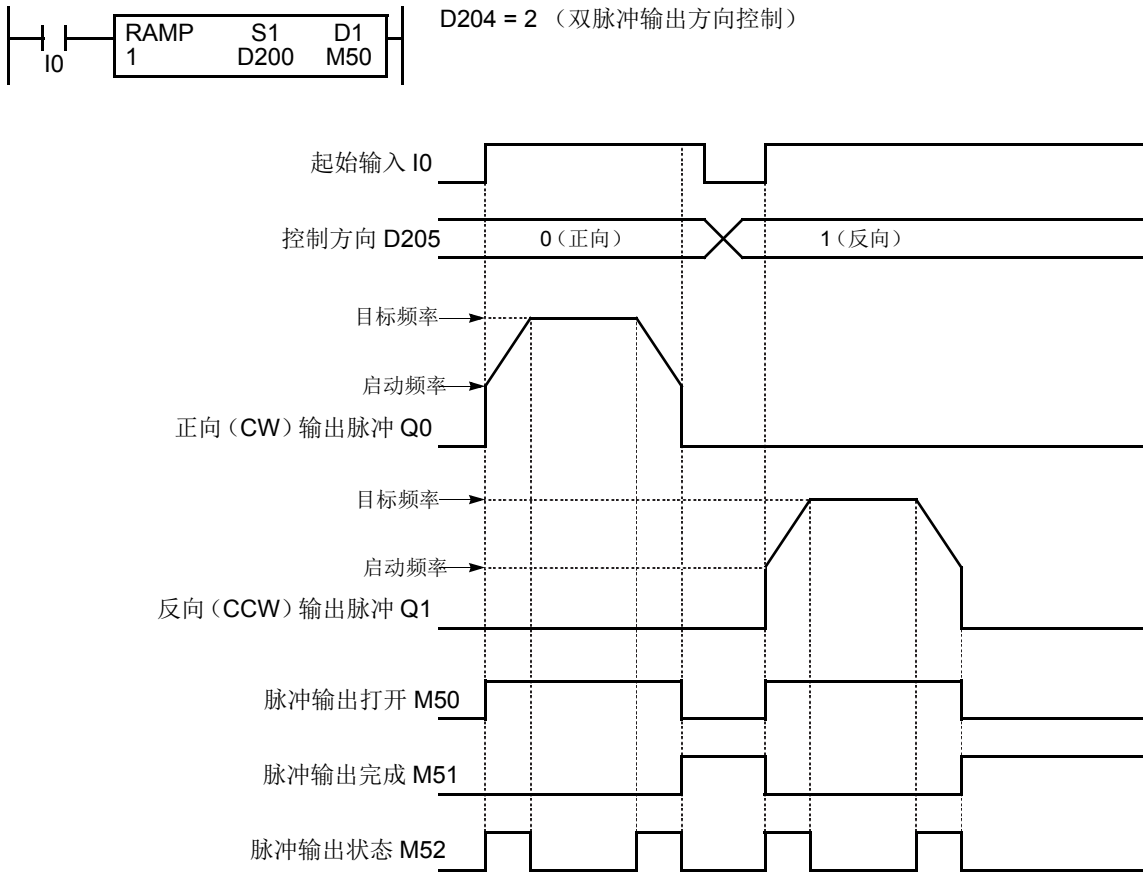


- 当输入 I0 打开时，RAMP1 生成以数据寄存器 D202 中存储的值指定的启动频率开始的输出脉冲。当输出 Q0 发出输出脉冲时，内部继电器 M50 保持打开。
- 动作模式 0 - 2: 脉冲频率按照数据寄存器 D203 中存储的频率更改比率增加。
- 动作模式 3: 脉冲频率会在数据寄存器 D203 中存储的频率更改时间内一直增加。
- 在输出脉冲频率增加时，内部继电器 M52 保持打开。
- 根据数据寄存器 D205 中存储的值指定的控制方向，当 D205 存储 0（正向）或 1（反向）时分别关闭或打开控制方向输出 Q1。
- 当输出脉冲频率达到数据寄存器 D201 中存储的值指定的目标频率时，内部继电器 M52 关闭。当输出脉冲频率开始降低时，内部继电器 M52 再次打开。
- 当生成的输出脉冲数量达到数据寄存器 D206 和 D207 指定的预置值时，RAMP1 将停止生成输出脉冲。然后内部继电器 M50 和 M52 关闭，内部继电器 M51 打开。
- 如果在生成输出脉冲时 D200 - D207（除了 D204）中的参数值更改，当起始输入 I0 开始下一个循环时更改生效。
- 如果在起始输入 I0 打开时 D204 中存储的值更改，则仅在 CPU 重启后更改才会生效。
- 如果在达到预置值之前关闭起始输入 I0，则 RAMP1 会立即停止生成输出脉冲，然后将关闭内部继电器 M50，并打开内部继电器 M51。当再次打开输入 I0 时，RAMP1 重新启动以生成下一循环的输出脉冲，该循环从启动频率开始。

13: 脉冲指令

双脉冲输出方向控制时序

该程序演示启用双脉冲输出方向控制时 RAMP1 指令的时序。



- 当输入 I0 打开时，RAMP1 生成以数据寄存器 D202 中存储的值指定的启动频率开始的输出脉冲。当输出 Q0 或 Q1 发出输出脉冲时，内部继电器 M50 保持打开。
- 动作模式 0 - 2: 脉冲频率按照数据寄存器 D203 中存储的频率更改比率增加。
- 动作模式 3: 脉冲频率会在数据寄存器 D203 中存储的频率更改时间内一直增加。
- 在输出脉冲频率增加时，内部继电器 M52 保持打开。
- 根据数据寄存器 D205 中存储的值指定的控制方向，当 D205 存储 0 (正向) 或 1 (反向) 时，输出 Q0 或 Q1 分别发出输出脉冲。
- 当输出脉冲频率达到数据寄存器 D201 中存储的值指定的目标频率时，内部继电器 M52 关闭。当输出脉冲频率开始降低时，内部继电器 M52 再次打开。
- 当生成的输出脉冲数量达到数据寄存器 D206 和 D207 指定的预置值时，RAMP1 将停止生成输出脉冲。然后内部继电器 M50 和 M52 关闭，内部继电器 M51 打开。
- 如果在生成输出脉冲时 D200 - D207 (除了 D204) 中的参数值更改，当起始输入 I0 开始下一个循环时更改生效。
- 如果在起始输入 I0 打开时 D204 中存储的值更改，则仅在 CPU 重启后更改才会生效。
- 如果在达到预置值之前关闭起始输入 I0，则 RAMP1 会立即停止生成输出脉冲，然后将关闭内部继电器 M50，并打开内部继电器 M51。当再次打开输入 I0 时，RAMP1 重新启动以生成下一循环的输出脉冲，该循环从启动频率开始。

示例程序：RAMP1— 方向控制禁用

该程序演示从输出 Q0 生成 48,000 个脉冲的 RAMP1 指令的用户程序。

目标频率：	6 kHz
启动频率：	300 Hz
频率更改时间：	2,000 (ms)
方向控制启用：	方向控制禁用
预置值：	总共 48,000 个脉冲

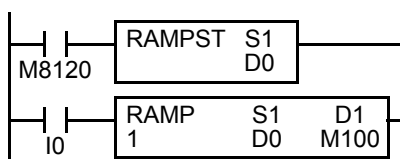
设置 WindLDR

在 WindLDR 编辑屏幕上，将光标插入您想插入脉冲指令宏的地方，并键入 **RAMPST**。如下所示输入参数。



设备设置

设备	功能	说明	设备地址 (值)
S1+0	动作模式	频率范围 200 Hz - 100 kHz	D0 (3)
S1+1	目标频率	6 kHz	D1 (600)
S1+2	启动频率	300 Hz	D2 (30)
S1+3	频率更改时间	2,000 ms	D3 (2000)
S1+4	启用方向控制	禁用方向控制	D4 (0)
S1+5	控制方向	未使用 (无影响)	D5
S1+6	预置值 (高位字)	48,000	D6/D7 (48000)
S1+7	预置值 (低位字)		
S1+8	当前值 (高位字)	0 - 48,000	D8/D9
S1+9	当前值 (低位字)		
S1+10	错误状态		D10
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	M100
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	M101
D1+2	脉冲输出状态	0: 目标脉冲输出 1: 更改输出脉冲频率	M102
D1+3	脉冲输出溢出	0: 未出现溢出 1: 出现溢出	M103



M8120 是初始化脉冲特殊内部继电器。

当启动 CPU 时，RAMPST 宏指定脉冲输出的参数。

当起始输入 I0 打开时，RAMP1 开始生成 48,000 个输出脉冲。

13: 脉冲指令

示例程序：RAMP1—单脉冲输出方向控制

该程序演示从输出 Q0 生成 100,000 个脉冲的 RAMP1 指令的用户程序。当输入 I1 关闭或打开时，控制方向输出 Q1 关闭或打开以分别指示正向或反向。

目标频率：10 kHz
 启动频率：500 Hz
 频率更改时间：2,000 (ms)
 方向控制启用：单脉冲输出方向控制
 预置值：总共 100,000 个脉冲

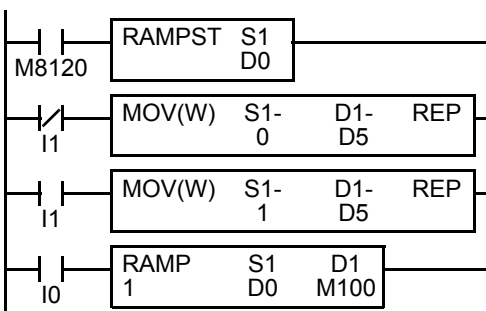
设置 WindLDR

在 WindLDR 编辑屏幕上，将光标插入您想插入脉冲指令宏的地方，并键入 **RAMPST**。如下所示输入参数。



设备设置

设备	功能	说明	设备地址 (值)
S1+0	动作模式	频率范围 200 Hz - 100 kHz	D0 (3)
S1+1	目标频率	10 kHz	D1 (1000)
S1+2	启动频率	500 Hz	D2 (50)
S1+3	频率更改时间	2,000 ms	D3 (2000)
S1+4	启用方向控制	单输出方向控制	D4 (1)
S1+5	控制方向	正向	D5 (0)
S1+6	预置值 (高位字)	100,000	D6/D7 (100000)
S1+7	预置值 (低位字)		
S1+8	当前值 (高位字)	0 - 100,000	D8/D9
S1+9	当前值 (低位字)		
S1+10	错误状态		D10



M8120 是初始化脉冲特殊内部继电器。

当启动 CPU 时，RAMPST 宏指定脉冲输出的参数。

当输入 I1 关闭时，D5 (控制方向) 存储 0 (正向)。

当输入 I1 打开时，D5 (控制方向) 存储 1 (反向)。

当起始输入 I0 打开时，RAMP1 开始生成 100,000 个输出脉冲。

示例程序：RAMP1— 双脉冲输出方向控制

该程序演示 RAMP1 指令的用户程序，可在输入 I1 关闭或打开时分别从输出 Q0（正向脉冲）或 Q1（反向脉冲）生成 1,000,000 个脉冲。

目标频率：30 kHz
 启动频率：10 kHz
 频率更改时间：2,000 (ms)
 方向控制启用：双脉冲输出方向控制
 预置值：总共 1,000,000 个脉冲

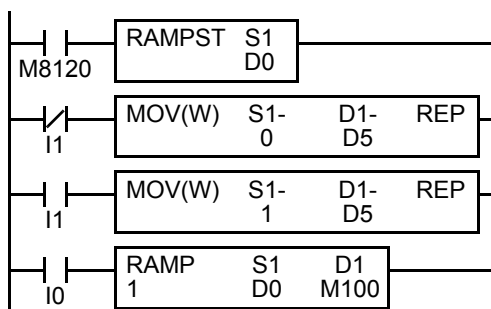
设置 WindLDR

在 WindLDR 编辑屏幕上，将光标插入您想插入脉冲指令宏的地方，并键入 **RAMPST**。如下所示输入参数。



设备设置

设备	功能	说明	设备地址 (值)
S1+0	动作模式	频率范围 200 Hz - 100 kHz	D0 (3)
S1+1	目标频率	30 kHz	D1 (3000)
S1+2	启动频率	10 kHz	D2 (1000)
S1+3	频率更改时间	2,000 ms	D3 (2000)
S1+4	启用方向控制	双输出方向控制	D4 (2)
S1+5	控制方向	正向	D5 (0)
S1+6	预置值 (高位字)	1,000,000	D6/D7 (1000000)
S1+7	预置值 (低位字)		
S1+8	当前值 (高位字)	0 - 1,000,000	D8/D9
S1+9	当前值 (低位字)		
S1+10	错误状态		D10



M8120 是初始化脉冲特殊内部继电器。

当启动 CPU 时，RAMPST 宏指定脉冲输出的参数。

当输入 I1 关闭时，D5（控制方向）存储 0（正向）。

当输入 I1 打开时，D5（控制方向）存储 1（反向）。

当起始输入 I0 打开时，RAMP1 开始生成 1,000,000 个输出脉冲。

13: 脉冲指令

ZRN1（零返回 1）



当输入打开时，ZRN1 指令从输出 Q0 发出预定的高频率脉冲输出。当减速输入打开时，输出频率降低至蠕变频率。当减速输入关闭时，ZRN1 指令停止生成输出脉冲。

输出脉宽比率固定在 50%。

ZRN2（零返回 2）



当输入打开时，ZRN2 指令从输出 Q1 发出预定的高频率脉冲输出。当减速输入打开时，输出频率降低至蠕变频率。当减速输入关闭时，ZRN2 指令停止生成输出脉冲。

输出脉宽比率固定在 50%。

ZRN3（零返回 3）



当输入打开时，ZRN3 指令从输出 Q2 发出预定的高频率脉冲输出。当减速输入打开时，输出频率降低至蠕变频率。当减速输入关闭时，ZRN2 指令停止生成输出脉冲。

输出脉宽比率固定在 50%。

不适用于 FC5A-16RK1/RS1

注释：ZRN1、ZRN2 和 ZRN3 指令在用户程序中只能使用一次。当没有使用 ZRN1、ZRN2 和 ZRN3 时，未使用的输出 Q0、Q1 或 Q2 可用于另一个脉冲指令或普通输出。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	—	X (ZRN1 和 ZRN2)	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X	—	—
S2 (源 2)	减速输入	X	—	▲	—	—	—	—	—	—
D1 (目标 1)	状态继电器	—	—	▲	—	—	—	—	—	—

源设备 S1（控制寄存器）使用设备指定以 S1 为起始编号的 5 个数据寄存器。可以将数据寄存器 D0 - D1995、D2000 - D7995 和 D10000 - D49995 指定为 S1。有关详细信息，请参阅下文。

源设备 S2（减速输入）可指定输入 I0 - I627 和内部继电器 M0 - M2557。特殊内部继电器不能指定为 S2。

目标设备 D1（状态继电器）使用以设备以 D1 为起始编号的 2 个内部继电器。可以将内部继电器 M0 - M2550 指定为 D1。指定为 D1 的内部继电器编号的最低有效数字必须是 0，否则 ZRN 指令不能正常工作。不能将特殊内部继电器指定为 D1。有关详细信息，请参阅第 6-2 页（基本卷）。

源设备 S1（控制寄存器）

根据需要，在执行 ZRN 指令前，将合适的值存储至以 S1 指定的设备开始的数据寄存器中，并确保该值在有效范围内。设备 S1+4 为只读。

设备	功能	说明	R/W
S1+0	归零速度动作模式	0: 10 Hz - 1 kHz 1: 100 Hz - 10 kHz 2: 1 kHz - 100 kHz 3: 200 Hz - 100 kHz（注释 1）	R/W
S1+1	启动频率	当 S1+0（动作模式）= 0 - 2 时：1 - 100（%） （所选模式 S1+0 的最高频率的 1% - 100%） 当 S1+0（动作模式）= 3 时：20 - 10,000（×10 Hz）（注释 2）	R/W
S1+2	蠕变动作模式	0: 10 Hz - 1 kHz 1: 100 Hz - 10 kHz 2: 1 kHz - 100 kHz 3: 200 Hz - 100 kHz（注释 1）	R/W
S1+3	蠕变脉冲频率	当 S1+2（动作模式）= 0 - 2 时：1 - 100（%） （所选模式 S1+2 的最高频率的 1% - 100%） 当 S1+2（动作模式）= 3 时：20 - 10,000（×10 Hz）（注释 2）	R/W
S1+4	错误状态	0 - 2	R

注释 1: 对于 FC5A-D12K1E 和 FC5A-D12S1E，模式 3 的频率范围为 250Hz - 100 kHz。

注释 2: 对于 FC5A-D12K1E 和 FC5A-D12S1E，模式 3 的频率范围为 25 - 10,000（×10 Hz）。

S1+0 归零速度动作模式

存储在设备 S1+0 指定的数据寄存器中的值可确定高频启动脉冲输出的频率范围。

- 0: 10 Hz - 1 kHz
- 1: 100 Hz - 10 kHz
- 2: 1 kHz - 100 kHz
- 3: 200 Hz - 100 kHz

S1+1 启动频率

当 S1+0 设为 0 - 2 时，存储在设备 S1+1 指定的数据寄存器中的值将以 S1+0 选定的频率范围最大值的百分比指定启动脉冲输出频率。设备 S1+1 的有效值为 1 - 100，因此启动脉冲频率可以分别为 10 Hz - 1 kHz（动作模式 0）、100 Hz - 10 kHz（动作模式 1）或 1 kHz - 100 kHz（动作模式 2）。

当 S1+0 设为 3（200 Hz - 100 kHz）时，设备 S1+1 的有效值为 20 - 10,000，将 S1+1 值乘以 10 可确定输出脉冲频率，因此输出脉冲频率可以按 10 Hz 的增量进行设置。脉冲频率错误最大为 ±5%。

归零速度动作模式（S1+0）	S1+1	启动频率（Hz）
0 - 2	1 - 100	S1+0 选择的最大频率 × S1+1 值（%）
3	20 - 10,000	S1+1 值 × 10

S1+2 蠕变动作模式

存储在设备 S1+2 指定的数据寄存器中的值可确定低频蠕变脉冲输出的频率范围。

- 0: 10 Hz - 1 kHz
- 1: 100 Hz - 10 kHz
- 2: 1 kHz - 100 kHz
- 3: 200 Hz - 100 kHz

13: 脉冲指令

S1+3 蠕变脉冲频率

当 S1+2 设为 0 - 2 时，存储在设备 S1+3 指定的数据寄存器中的值将以 S1+2 选定的频率范围最大值的百分比指定蠕变脉冲输出频率。设备 S1+3 的有效值为 1 - 100，因此启动频率可以分别为 10 Hz - 1 kHz（动作模式 0）、100 Hz - 10 kHz（动作模式 1）或 1 kHz - 100 kHz（动作模式 2）。

当 S1+2 设为 3（200 Hz - 100 kHz）时，设备 S1+3 的有效值为 20 - 10,000（以 10 为增量），将 S1+3 值乘以 10 可确定蠕变脉冲频率，因此蠕变脉冲频率可以按 10 Hz 的增量进行设置。脉冲频率错误最大为 ±5%。

蠕变动作模式 (S1+2)	S1+3	蠕变脉冲频率 (Hz)
0 - 2	1 - 100	S1+2 选择的最大频率 × S1+3 值 (%)
3	20 - 10,000	S1+3 值 × 10

S1+4 错误状态

当打开 ZRN 指令的起始输入时，将校验设备值。若在设备值中发现错误，则错误代码将存储在设备 S1+4 指定的数据寄存器中。

错误代码	说明
0	正常
1	动作模式设置错误 (S1+0 或 S1+2 存储 0 - 3 以外的值)
2	输出脉冲频率设置错误 (S1+1 的存储值不在启动脉冲输出频率范围内，或 S1+3 的存储值不在蠕变脉冲输出频率内。)

源设备 S2（减速输入）

若 ZRN 指令生成启动频率的输出脉冲时减速输入已打开，则脉冲频率会更改至蠕变脉冲频率。当减速输入关闭时，ZRN 指令停止生成输出脉冲。

当使用 ZRN1、ZRN2 和 ZRN3 指令时，要为 ZRN1、ZRN2 和 ZRN3 指令的减速输入分配不同的输入或内部继电器编号。如果使用相同的减速输入，且同时执行 ZRN1、ZRN2 和 ZRN3 指令，则当减速输入打开时，脉冲输出可能不关闭。

根据指定的设备编号有两种减速输入可用。

设备	功能	说明
S2	高速减速输入	I2、I3、I4、I5
	正常减速输入	I0、I1、I6 - I627、M0 - M2557

高速减速输入（I2、I3、I4、I5）

不管扫描时间如何，高速减速输入使用中断处理来读取减速输入信号。

当 I2 - I5 用于 ZRN 指令的减速输入时，请在功能设置中指定这些输入编号为正常输入。如果在功能设置中将用作减速输入的 I2 - I5 指定为中断输入、捕捉输入或高速计数器输入，则这些输入将用作 ZRN 指令的减速输入；功能设置中的设置无效。

当使用高速减速输入时，确保输入接点重复。如果输入信号收到干扰，则脉冲输出将立即停止。

正常减速输入（I0、I1、I6 - I627、M0 - M2557）

当在 END 处理时更新输入数据，若正常减速输入读取减速输入信号，则接受减速输入的定时视扫描时间而定。

目标设备 D1（状态继电器）

以 D1 指定的设备开始的两个内部继电器指示 ZRN 指令的状态。这些设备为只读。

设备	功能	说明	R/W
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	R
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	R

D1+0 脉冲输出打开

在 ZRN 指令生成输出脉冲时，设备 D1+0 指定的内部继电器保持打开。当 ZRN 指令的起始输入或减速输入关闭以停止生成输出脉冲时，设备 D1+0 指定的内部继电器关闭。

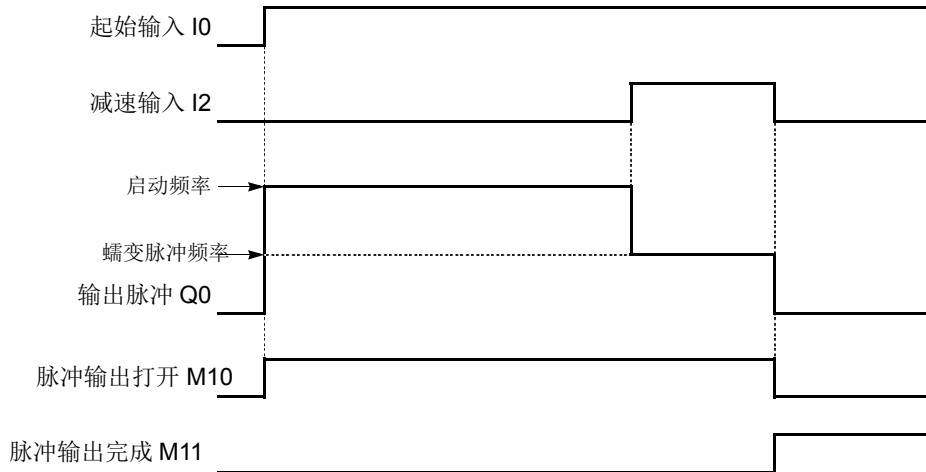
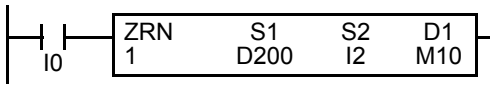
D1+1 脉冲输出完成

当 ZRN 指令的减速关闭以停止生成输出脉冲时，设备 D1+1 指定的内部继电器打开。当 ZRN 指令的起始输入打开时，设备 D1+1 指定的内部继电器关闭。

13: 脉冲指令

归零操作时序

该程序演示当输入 I2 用作高速减速输入时 ZRN1 指令的时序。



- 当输入 I0 打开时，ZRN1 开始生成数据寄存器 D201 中存储的值指定启动频率的输出脉冲。当输出 Q0 发出输出脉冲时，内部继电器 M10 保持打开。
- 当减速输入 I2 打开时，输出脉冲频率立即降低至数据寄存器 D203 中存储的值指定的蠕变脉冲频率。
- 当减速输入 I2 关闭时，ZRN1 指令立即停止生成输出脉冲。然后内部继电器 M10 关闭，内部继电器 M11 打开。
- 如果在生成输出脉冲时 D200 - D203 中的参数值更改，当起始输入 I0 开始下一个循环时更改生效。
- 如果在生成启动频率或蠕变频率的输出脉冲时，起始输入 I0 关闭，ZRN1 将停止生成输出脉冲，然后内部继电器 M10 关闭，内部继电器 M11 打开。当再次打开输入 I0 时，ZRN1 重新启动以生成下一循环的输出脉冲，该循环从启动频率开始。
- 如果在起始输入 I0 已打开时，减速输入 I2 已打开，则 ZRN1 开始生成蠕变脉冲频率的脉冲输出。

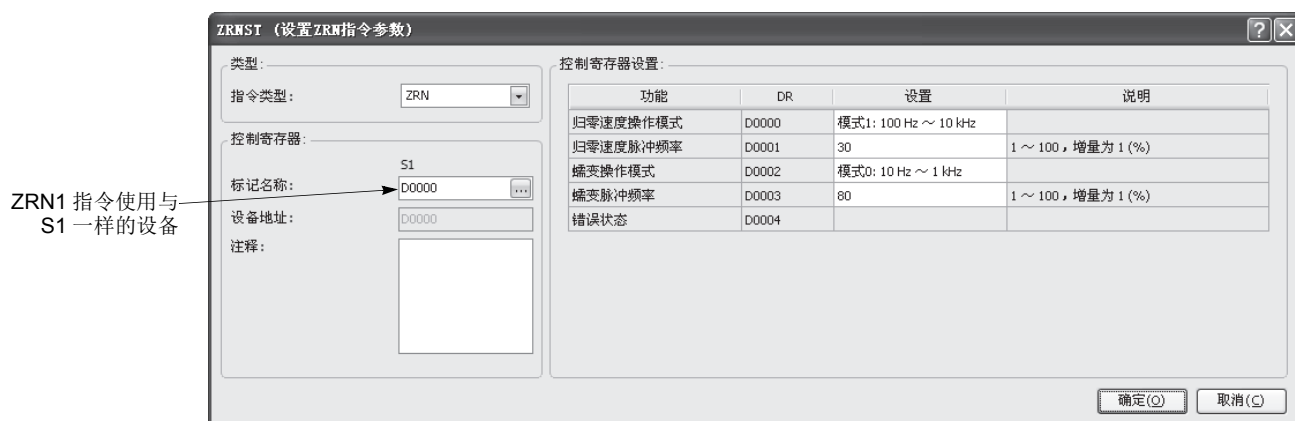
示例程序 : ZRN1

该程序演示使用 ZRN1 指令的用户程序进行归零操作以在输入 I1 打开时从输出 Q0 生成 3 kHz 启动频率的输出脉冲。当减速输入 I3 打开时，输出脉冲频率降至 800 Hz 蠕变脉冲频率。当减速输入 I3 关闭时 ZRN1 指令将停止生成输出脉冲。

启动频率： 3 kHz
 蠕变脉冲频率： 800 Hz
 减速输入： I3（高速减速输入）

设置 WindLDR

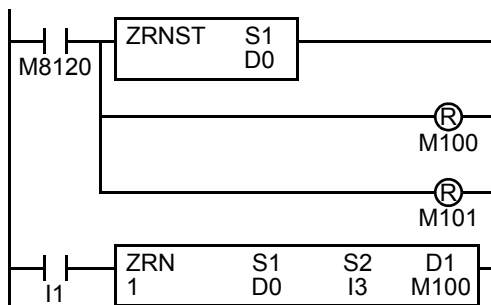
在 WindLDR 编辑屏幕上，将光标插入您想插入脉冲指令宏的地方，并键入 **ZRNST**。如下所示输入参数。



ZRN1 指令使用与 S1 一样的设备

设备设置

设备	功能	说明	设备地址 (值)
S1+0	归零速度动作模式	频率范围 100 Hz - 10 kHz	D0 (1)
S1+1	启动频率	$10 \text{ kHz} \times 30\% = 3 \text{ kHz}$	D1 (30)
S1+2	蠕变动作模式	频率范围 10 Hz - 1 kHz	D2 (0)
S1+3	蠕变脉冲频率	$1 \text{ kHz} \times 80\% = 800 \text{ Hz}$	D3 (80)
S1+4	错误状态		D4
S2	减速输入	高速减速输入	I3
D1+0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	M100
D1+1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	M101



M8120 是初始化脉冲特殊内部继电器。

当启动 CPU 时，ZRNST 宏指定脉冲输出的参数。

脉冲输出打开标志 M100 已关闭。

脉冲输出完成标志 M101 已关闭。

当起始输入 I1 打开时，ZRN1 开始从输出 Q0 生成输出脉冲。

14: PID 指令

简介

PID 指令使用内置自动调节实现 PID（比例、积分和微分）算法以自动确定 PID 参数，如比例增益、积分时间、微分时间和控制动作。此外，提前自动调节不需要指定自动调节参数即可自动确定 PID 参数。

PID 指令主要设计用于与模拟量 I/O 模块一起使用以读取模拟量输入数据，打开和关闭指定输出以在应用程序中执行 PID 控制，如第 14-18 页上的应用程序示例描述的温度控制。此外，PID 指令也会生成模拟量输出模块的输出操作变量。当把该设备值传送到模拟量输出模块时，可生成电压 (0 ~ 10V DC) 或电流 (4 ~ 20 mA DC) 输出。



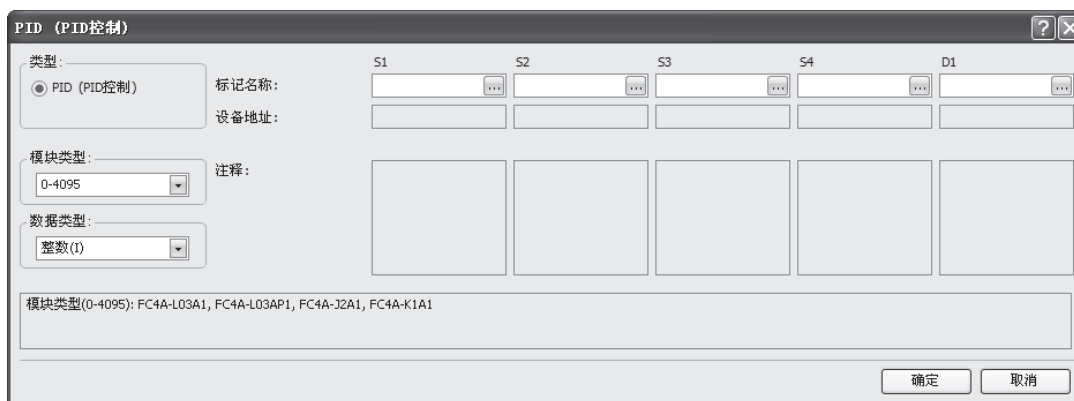
警告

- 要使用 Micro Smart 的 PID 功能，则需要有关 PID 控制的专业技术知识。使用 PID 功能而不理解 PID 控制可能会使 MicroSmart 执行误动作，导致控制系统发生异常、损坏或故障。
- 当使用 PID 指令进行反馈控制时，必须在 MicroSmart 外部设置紧急停止和联锁电路。如果在 MicroSmart 内部设置这些电路，则输入进程变量失败便会导致设备损坏或故障。

PID (PID 控制)



当输入打开时，将根据为动作模式指定的数据寄存器设备中存储的值 (0 ~ 4) 执行自动调节或 PID 动作。



可用 CPU 模块和 PID 指令的数量

根据 CPU 模块类型，在用户程序中最多可以使用 32 或 56 个 PID 指令。

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X (32)	X (56)	X (56)	X (56)

14: PID 指令

有效设备

设备	功能	I	Q	M	R	T	C	D	常量
S1 (源 1)	控制寄存器	—	—	—	—	—	—	D0-D7973 D10000-D49973	—
S2 (源 2)	控制继电器	—	Q0- Q620	M0-M2550	—	—	—	—	—
S3 (源 3)	设置点	—	—	—	—	—	—	D0-D7999 D10000-D49999	0-4095 0-50000
S4 (源 4)	进程变量 (转换前)	—	—	—	—	—	—	D0-D7999 D10000-D49999	—
D1 (目标 1)	操作变量	—	—	—	—	—	—	D0-D7999 D10000-D49999	—
模块类型	根据模拟量 I/O 模块, 选择 0 ~ 4095 或 0 ~ 50000。								
数据类型	当使用模拟量模块类型 0 ~ 50000 时, 请从字型 (W) 或整型 (I) 中选择数据类型。								

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

源设备 S1 (控制寄存器) 使用以 S1 指定的设备开始的 27 个数据寄存器。S1 可以指定数据寄存器 D0 ~ D1973、D2000 ~ D7973 和 D10000 ~ D49973。有关详细信息, 请参阅下文。

源设备 S2 (控制继电器) 使用输出 8 点或由 S2 指定的设备开始的内部继电器。S2 可以指定输出 Q0 ~ Q620 和内部继电器 M0 ~ M2550。有关详细信息, 请参阅第 14-14 页。

源设备 S3 (设置点): 当禁用线性变换 (S1+4 设为 0 或 2) 时, 设置点 (S3) 的有效范围为 0 ~ 4095 或 50,000, 可以使用数据寄存器或常量指定。当启用线性变换 (S1+4 设为 1 或 3) 时, 有效范围为 0 ~ 65535 (字型数据类型) 或 -32768 ~ 32767 (整型数据类型), 这是线性变换后的值。当使用线性变换时, 请使用数据寄存器指定设置点的负值。有关详细信息, 请参阅第 14-16 页。

使用作为已连接的模拟量 I/O 模块 (如 D760 或 D766) 的模拟量输入数据分配的数据寄存器指定源设备 S4 (进程变量)。要从模拟量 I/O 模块中读取输入数据, 请根据模拟量 I/O 模块的插槽位置和连接至模拟量输入源的模拟量输入频道号指定正确的数据寄存器编号。有关详细信息, 请参阅第 14-16 页。

源设备 S4 与模拟量输入模块的二进制数据类型兼容, 范围从 0 ~ 60,000。

目标设备 D1 (操作变量) 存储 -32768 ~ 32767, 即 PID 动作的计算结果。有关详细信息, 请参阅第 14-16 页。

模块类型: 根据模拟量 I/O 模块, 选择 0 ~ 4095 或 0 ~ 50000。这些值确定进程变量 (S4) 的数据范围和模拟量输出模块 (S1+24) 输出操作变量。

模块类型	模拟量 I/O 模块
0-4095	FC4A-L03A1 FC4A-L03AP1 FC4A-J2A1 FC4A-K1A1 FC4A-K4A1
0-50000	FC4A-J4CN1 FC4A-J8C1 FC4A-J8AT1 (注释) FC4A-K2C1

注释: PTC 和 NTC 热敏电阻的 FC4A-J8AT1 模拟量输出模块只能在支持热敏电阻线性的范围内使用。

数据类型: 当使用 0-50000 的模拟量 I/O 模块类型时, 请从字型 (W) 整型 (I) 中选择数据类型。当使用 0-4095 的模拟量 I/O 模块类型时, 数据类型固定在整型 (I)。数据类型选择影响设置点 (S4), 线性变换最大值和最小值 (S1+5 和 S1+6), 上限报警值和下限报警值 (S1+14 和 S1+15), AT 设置点 (S1+21) 和模拟量输出模式 (S1+24) 的输出操作值。

源设备 S1（控制寄存器）

根据需要，在执行 PID 指令前，将合适的值存储至以分配给 S1 的设备开始的数据寄存器中，并确保该值在有效范围内。设备 S1+0 ~ S1+2、S1+23 和 S1+24 为只读。有关使用宏编写设备，请参阅第 14-21 页。

设备	功能	说明	R/W
S1+0	进程变量 (转换后)	当 S1+4(控制模式) = 1 或 3(启用线性变换): 存储转换后的进程变量 当 S1+4(控制模式) = 0 或 2(禁用线性变换): 存储未转换的进程变量	R
S1+1	输出操作变量	存储以百分比表示的输出操作变量 (手动模式输出变量和 AT 输出操作变量)。 0 ~ 100 (0% ~ 100%)	R
S1+2	运行状态	存储 PID 指令的运行状态或错误状态。	R
S1+3	动作模式	0: PID 动作 1: AT (自动调节) + PID 动作 2: AT (自动调节) 3: 高级 AT + PID 动作 4: 高级 AT	R/W
S1+4	控制模式 (线性转换和比例项)	0: 禁用线性变换, 比例增益 1: 启用线性变换, 比例增益 2: 禁用线性变换, 比例带 3: 启用线性变换, 比例带	R/W
S1+5	线性变换最大值	字数据类型: 0 ~ 65535 整型数据类型: -32768 ~ +32767	R/W
S1+6	线性变换最小值	字数据类型: 0 ~ 65535 整型数据类型: -32768 ~ +32767	R/W
S1+7	比例项	当 S1+4(控制模式) = 0 或 1(比例增益): 1 ~ 10000(0.01% ~ 100.00%) 0 指定 0.01%, ≥ 10001 指定 100.00% 当 S1+4(控制模式) = 2 或 3(比例带): 1 ~ 10000 (±0.01% ~ ±100.00%) 0 指定 ±0.01%, ≥ 10001 指定 ±100.00%	R/W
S1+8	积分时间	1 ~ 65535 (0.1s ~ 6553.5s), 0 禁用积分运算	R/W
S1+9	微分时间	1 ~ 65535 (0.1s ~ 6553.5s), 0 禁用微分运算	R/W
S1+10	积分开始系数	S1 + 4 为 0、1: 1 ~ 100 (1% ~ 100%) 0 或 101 以上作为 100% 执行 S1 + 4 为 2、3: 10001 ~ 10100 (1% ~ 100%) 10000 以下或 10101 以上作为 100% 执行	R/W
S1+11	输入过滤器系数	0 ~ 99 (0% ~ 99%), ≥ 100 指定 99%	R/W
S1+12	取样周期	1 ~ 10000 (0.01s ~ 100.00s) 0 指定 0.01s, ≥ 10001 指定 100.00s	R/W
S1+13	控制周期	1 ~ 500 (0.1s ~ 50.0s) 0 指定 0.1s, ≥ 501 指定 50.0s	R/W
S1+14	上限报警值	当 S1+4(控制模式) = 0 或 2(禁用线性变换): 0 ~ 4095 (≥ 4096 指定 4095) 0 ~ 50000 (≥ 50001 指定 50000) 当 S1+4(控制模式) = 1 或 3(启用线性变换): 线性变换最小值 ≤ 上限报警值 ≤ 线性变换最大值 当 S1+14 < S1+6 (线性变换时间)、S1+6 变为上限报警时。 当 S1+14 > S1+5 (线性变换最大值)、S1+5 变为上限报警时。	R/W
S1+15	下限报警值	当 S1+4(控制模式) = 0 或 2(禁用线性变换): 0 ~ 4095 (≥ 4096 指定 4095) 0 ~ 50000 (≥ 50001 指定 50000) 当 S1+4(控制模式) = 1 或 3(启用线性变换): 线性变换最小值 ≤ 下限报警值 ≤ 线性变换最大值 当 S1+15 < S1+6 (线性变换时间)、S1+6 变为下限报警时。 当 S1+15 > S1+5 (线性变换最大值)、S1+5 变为下限报警时。	R/W

14: PID 指令

设备	功能	说明	R/W
S1+16	输出操作变量上限	0 ~ 100、10001 ~ 10099 (其他值指定 100)	R/W
S1+17	输出操作变量下限	0 ~ 100 (≥ 101 指定 100)	R/W
S1+18	手动模式输出操作变量	0 ~ 100 (≥ 101 指定 100)	R/W
S1+19	AT 取样周期	1 ~ 10000 (0.01s ~ 100.00s) 0 指定 0.01s, ≥ 10001 指定 100.00s	R/W
S1+20	AT 控制周期	1 ~ 500 (0.1s ~ 50.0s) 0 指定 0.1s, ≥ 501 指定 50.0s	R/W
S1+21	AT 设置点	当 S1+4 (控制模式) = 0 或 2 时: 0 ~ 4095 (≥ 4096 指定 4095) 0 ~ 50000 (≥ 50001 指定 50000) 当 S1+4 (控制模式) = 1 或 3 时: 线性变换最小值 ≤ AT 设置点 ≤ 线性变换最大值 当 S1+21 < S1+6 (最小线性变换)、S1+6 变为 AT 设置点时。 当 S1+21 > S1+5 (最大线性变换)、S1+5 变为 AT 设置点时。	R/W
S1+22	AT 输出操作变量	0 ~ 100 (≥ 101 指定 100)	R/W
S1+23	输出操作变量 %	-32768 ~ 32767 (-327.68% ~ 327.67%)	R
S1+24	模拟量输出模块的输出操作变量	根据模拟量输出模块类型, 从输出操作变量 (S1+1) 转换。 0 ~ 4095(0% ~ 100%) 0 ~ 50000 (0% ~ 100%)	R
S1+25	比例带补偿值	-100 ~ 100(-100% ~ 100%) ≤ -101 指定 -100%, ≥ 101 指定 100%	R/W
S1+26	微分增益	0 ~ 100(0% ~ 100%) ≥ 101 指定 100%	R/W

注释: 仅当 PID 指令的起始输入打开时才会校验 S1+3 (动作模式) 指定的数据寄存器中存储的值。每次扫描时都会刷新所有其他控制寄存器中的值。

S1+0 进程变量 (转换后)

当启用线性变换 (S1+4 设为 1 或 3) 时, S1+0 指定的数据寄存器将存储进程变量 (S4) 的线性变换结果。进程变量 (S1+0) 会提取线性变换最小值 (S1+6) 和线性变换最大值 (S1+5) 之间的一个值。

当禁用线性变换 (S1+4 设为 0 或 2) 时, S1+0 指定的数据寄存器将存储与进程变量 (S4) 相同的值。

S1+1 输出操作变量

在进行 PID 动作时, 由 S1+1 指定的数据寄存器会保留从操作变量中读取的 0 ~ 100, 以及在 D1 指定的数据寄存器中存储的 -32768 ~ 32767, 并忽略小于 0 和大于 100 的值。S1+1 中的百分比值可按照与控制周期 (S1+13) 的比例确定输出控制 (S2+6) 的打开时间。

当启用手动模式, 且自动 / 手动模式控制继电器 (S2+1) 设为开时, S1+1 存储从手动模式输出操作变量 (S1+18) 中读取的 0 ~ 100。

在进行自动调节 (AT) 时, S1+1 会存储从 AT 输出操作变量 (S1+22) 中读取的 0 ~ 100。

S1+2 运行状态

由 S1+2 指定的数据寄存器会存储 PID 指令的运行状态或错误状态。

状态代码 1X ~ 6X 中包含启动自动调节或 PID 动作后的已用时间。X 从 0 ~ 9 (以 10 分钟为增量) 更改为代表 0 ~ 90 分钟。当时间过去 90 分钟后, 时间代码仍然为 9。当动作模式 (S1+3) 设为 1 (AT+PID) 时, 时间代码复位为 0, 并且从 AT 转为 PID。

状态代码为 100 及更大的值时表示出错, 并停止自动调节或 PID 动作。当出现这些错误时, 将导致用户程序执行错误, 并打开 ERR LED 和特殊内部继电器 M8004 (用户程序执行错误)。要继续操作, 请输入正确的参数, 并打开 PID 指令的起始输入。

状态代码	说明	操作
1X	AT 正在进行	AT 正常。
2X	AT 已完成	
5X	PID 动作正在进行	PID 动作正常
6X	已达到 PID 设置点 (S3)。当达到 PID 设置点时, 状态代码从 5X 更改为 6X。	

状态代码	说明	操作
100	动作模式 (S1+3) 设为大于 4 的值。	由于参数设置错误, PID 动作或 AT 停止。
101	控制模式 (S1+4) 设为大于 3 的值。	
102	当启用线性变换 (S1+4 设为 1 或 3) 时, 线性变换最大值 (S1+5) 和线性变换最小值 (S1+6) 设为相同的值。	
103	输出操作变量上限 (S1+16) 设为小于输出操作变量下限 (S1+17) 的值。	
104	当启用线性变换 (S1+4 设为 1 或 3) 时, AT 设置点 (S1+21) 设为大于线性变换最大值 (S1+5) 或小于线性变换最小值 (S1+6) 的值。	
105	当禁用线性变换 (S1+4 设为 0 或 2) 时, 根据模拟量模块的类型, AT 设置点 (S1+21) 设为大于 4095 或 50000 的值。	
106	当启用线性变换 (S1+4 设为 1 或 3) 时, AT 设置点 (S1+21) 设为大于线性变换最大值 (S1+5) 或小于线性变换最小值 (S1+6) 的值。	
107	当禁用线性变换 (S1+4 设为 0 或 2) 时, 根据模拟量模块的类型, AT 设置点 (S3) 设为大于 4095 或 50000 的值。	由于 AT 执行错误, AT 停止。
108	当 AT + PID 动作执行 (S1+3 设置为 1 或 3) 时, 进程变量 (S1+0) 不能达到 AT 设置点 (S1+21)。 • 在直接控制动作 (S2+0 打开) 中, 当进程变量有下列关系时将启动 AT + PID 动作: 设置点 (S3) \leq 进程变量 (S1+0) \leq AT 设置点 (S1+21) 要解决此问题, 将 AT 设置点设置为小于进程变量的值。 • 在反向控制动作 (S2+0 关闭) 中, 当进程变量有下列关系时将启动 AT + PID 动作: 设置点 (S1+21) \leq 进程变量 (S1+0) \leq AT 设置点 (S3) 要解决此问题, 将 AT 设置点设置为大于进程变量的值。	
200	当前控制动作 (S2+0) 不同于在 AT 开始时确定的控制动作。要重新启动 AT, 请根据下列可能原因设置正确的参数: • 操作变量 (D1) 或控制输出 (S2+6) 没有输出至正确的控制目标。 • 进程变量没有存储至 S4 指定的设备中。 • AT 输出操作变量 (S1+22) 没有设为足够大的值以使进程变量 (S4) 发生显著更改。 • 出现严重干扰。	
201	由于进程变量 (S4) 波动过度, 导致 AT 无法正常完成。要重新启动 AT, 请设置 AT 取样周期 (S1+19) 或输入过滤器系数 (S1+11) 为一个较大的值。	
202	因为 AT 取样周期数量太小, AT 不能产生正确的结果。当 AT 取样周期 (S1+19) 过长或 AT 取样开始和 AT 取样点 (S1+21) 的进程变量之间差异过小时, 将发生此错误。将 AT 取样周期 (S1+21) 设置到可执行足够阶跃作用的值, 并将 AT 取样周期 (S1+19) 设置到能够进行 10 次以上取样的值。	

S1+3 动作模式

当打开 PID 指令的起始输入时, CPU 模块将校验 S1+3 指定的数据寄存器中存储的值, 并执行所选操作。在执行 PID 指令时不能更改选择。

0: PID 动作

根据指定的 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9)、取样周期 (S1+12)、控制周期 (S1+13) 和控制动作 (S2+0)] 执行 PID 动作。

1: AT (自动调节) + PID 动作

根据指定的 AT 参数 [如 AT 取样周期 (S1+19)、AT 控制周期 (S1+20)、AT 设置点 (S1+21) 和 AT 输出操作变量 (S1+22)] 首先执行自动调节。作为自动调节的结果, 确定 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9) 和控制动作 (S2+0)], 然后根据除了指定的 PID 参数 [如取样周期 (S1+12) 和控制周期 (S1+13)] 外的微分参数执行 PID 动作。

2: AT (自动调节)

根据指定的 AT 参数执行自动调节, 以确定 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9) 和控制动作 (S2+0)]; 不执行 PID 动作。

14: PID 指令

3: 高级 AT(自动调节)+ PID 动作

根据指定的 AT 参数 [如 AT 取样周期 (S1+19)、AT 控制周期 (S1+20)、AT 设置点 (S1+21) 和 AT 输出操作变量 (S1+22)] 首先执行自动调节。作为自动调节的结果, 确定 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9)、取样周期 (S1+12)、控制周期和控制方向 (S2+0)], 然后根据微分 PID 参数执行 PID 动作。

4: 高级 AT(自动调节)

根据自动指定的 AT 参数执行自动调节, AT 设置点 (S1+21) 除外, 以确定 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9)、取样时间 (S1+12) 和控制周期 (S2+0)]; 不执行 PID 动作。

动作模式的设备指定 (S1+3)

下表总结了需要为各个动作模式指定的设备。当使用自动调节时, 自动确定几个设备, 不需要指定。

动作模式 (S1+3)		0	1	2	3	4
		PID 动作	AT (自动调谐) + PID 动作	AT (自动调谐)	高级 AT + PID 动作	高级 AT
S1+7	比例项	指定 *	自动 *	自动	自动 *	自动
S1+8	积分时间	指定 *	自动 *	自动	自动 *	自动
S1+9	微分时间	指定 *	自动 *	自动	自动 *	自动
S1+12	取样周期	指定 *	指定 *	—	自动 *	自动
S1+13	控制周期	指定 *	指定 *	—	自动 *	自动
S1+19	AT 取样周期	—	指定	指定	自动	自动
S1+20	AT 控制周期	—	指定	指定	自动	自动
S1+21	AT 设置点	—	指定	指定	自动	指定
S1+22	AT 输出操作变量	—	指定	指定	自动	自动
S2+0	控制动作	指定	自动	自动	自动	自动

* 当进行 PID 动作 (动作状态 S1+1 是 5X 或 6X) 时, 这些值因微调改变。注意不恰当的改变会导致非所求操作。

S1+4 控制模式 (线性变换和比例项)

控制模式指定是否禁用或启用线性转换以及比例项是否使用比例增益或比例带。

控制模式 (S1+4)	线性转换	比例项
0	禁用线性变换	比例增益
1	启用线性变换	比例增益
2	禁用线性变换	比例带
3	启用线性变换	比例带
其它	错误状态 101	

注释: 当正在进行 PID 动作 (运算状态 S1+1 是 5X 或 6X) 时, 请勿更改控制模式 (S1+4), 否则 PID 动作会导致错误并停止运算。

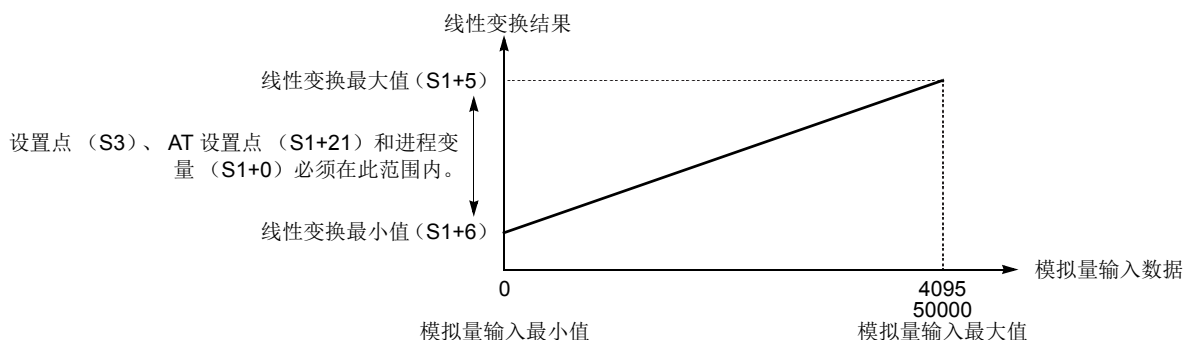
禁用线性变换

不执行线性变换。当禁用线性变换 (S1+4 设为 0 或 2) 时, 来自模拟量 I/O 模块的模拟量输入数据 (0 ~ 4095 或 50000, 根据模拟量 I/O 模块类型) 将存储至进程变量 (S4), 且相同的值不经转换便存储至进程变量 (S1+0)。

启用线性变换

线性变换功能用于将工程单元中的进程变量缩放为实际测量值。

当启用线性变换 (S1+4 设为 1 或 3) 时, 来自模拟量 I/O 模块的模拟量输入数据 (0 ~ 4095 或 50000, 根据模拟量 I/O 模块类型) 将进行线性变换, 结果会存储至进程变量 (S1+0) 中。当使用线性变换时, 请设置正确的线性变换最大值 (S1+5) 和线性变换最小值 (S1+6) 以指定线性变换的输出范围。当在温度控制应用程序中使用线性变换功能时, 温度值可用于指定设置点 (S3)、上限报警值 (S1+14)、下限报警值 (S1+15) 和 AT 设置点 (S1+21), 还可用于读取进程变量 (S1+0)。



比例增益或比例带

可以从比例增益 (S1+4 设为 0 或 1) 或比例带 (S1+4 设为 2 或 3) 中选择比例项 (S1+7)。

S1+5 线性变换最大值

当启用线性变换 (S1+4 设为 1 或 3) 时, 设置线性变换最大值至 S1+5 指定的数据寄存器。有效值为 0 ~ 65535(字型数据类型) 或 -32768 ~ 32767(整型数据类型), 且线性变换最大值必须大于线性变换最小值 (S1+6)。选择合适的线性变换最大值作为模拟量 I/O 模块输入信号的最大值。

当禁用线性变换 (S1+4 设为 0 或 2) 时, 不必设置线性变换最大值 (S1+5)。

S1+6 线性变换最小值

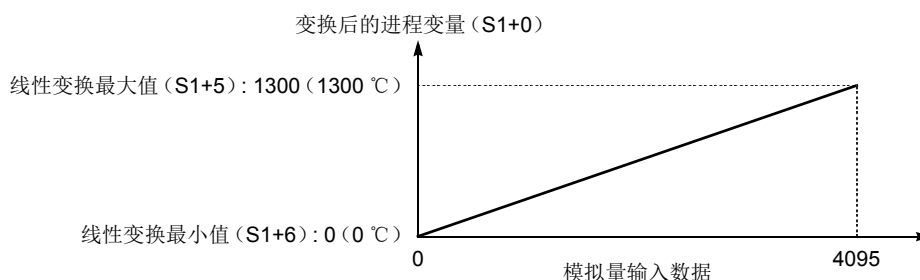
当启用线性变换 (S1+4 设为 1 或 3) 时, 设置线性变换最小值至 S1+6 指定的数据寄存器。有效值为 0 ~ 65535(字型数据类型) 或 -32768 ~ 32767(整型数据类型), 且线性变换最小值必须大于线性变换最大值 (S1+5)。选择合适的线性变换最小值作为模拟量 I/O 模块输入信号的最小值。

当禁用线性变换 (S1+4 设为 0 或 2) 时, 不必设置线性变换最小值 (S1+5)。

示例:

当连接了 K 型热电偶时, 模拟量输入数据的范围为 0 ~ 4095。要转换模拟量输入数据至实际测量的温度值, 请设置以下参数。

线性变换 (S1+4):	1 或 3 (启用线性变换)
线性变换最大值 (S1+5):	1300 (1300°C)
线性变换最小值 (S1+6):	0 (0°C)



S1+7 比例项

比例项是根据控制模式 (S1+4) 确定比例增益或比例带中比例动作数量的参数。

当通过设置动作模式 (S1+3) 到 1 (AT+PID)、2 (AT)、3 (高级 AT+PID), 或 4 (高级 AT) 使用自动调节或高级自动调节时, 会自动确定比例项, 并且不必由用户指定。

当设置动作模式 (S1+3) 为 0 (PID) 时未使用自动调节, 从 1 ~ 10000 中选取所需值以指定 0.01% ~ 100.00% 的比例增益或 $\pm 0.01\%$ ~ $\pm 100.00\%$ 的比例带给 S1+7 指定的数据寄存器。当 S1+7 存储 0 时, 比例增益设为 0.01% 或比例带设置为 $\pm 0.01\%$ 。当 S1+7 存储大于 10000 的值时, 比例增益设置为 100.00%, 或比例带设置为 $\pm 100.00\%$ 。

当选择比例增益时, 从设置点 (S3) 和进程变量 (S4) 之间的变差计算输出操作变量 (S1+1)。当比例增益设为较大的值时, 比例带变小, 响应速度加快, 但是会导致脉冲跳增和异常。相反, 当比例增益设为较小的值时, 则会抑止脉冲跳增和异常, 但是对干扰的响应速度会变慢。

14: PID 指令

比例带是输出操作变量 (S1+1) 从 0% 变化到 100% 需要的输入范围 (设置点和进程变量之间的偏差)。从有关比例带的电流输入中计算比例项的输出操作变量 (S1+1)。当选择比例带时, 只有进程变量 (S1+0) 在比例带内时启用积分运算, 即是输出操作变量的计算值在 0% 和 100% 之间。当进程变量 (S1+0) 超出比例带时, 将禁用积分运算。

当进行 PID 动作时, 用户可以更改比例项值。

S1+8 积分时间

当仅使用比例运算时, 设置点 (S3) 和进程变量 (S1+0) 之间的差量 (偏移量) 会持续到控制目标达到稳定状态后。需要进行积分运算将偏移量降低至零。积分时间是确定积分运算量的参数。

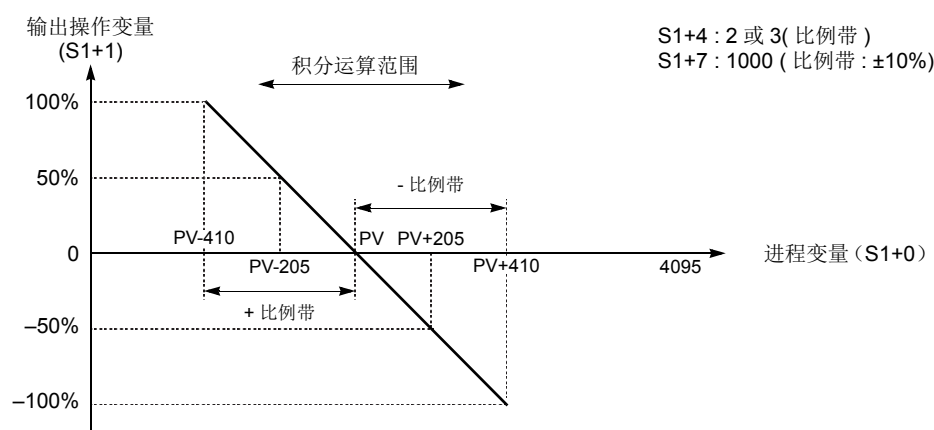
当通过设置动作模式 (S1+3) 到 1 (AT+PID)、2 (AT)、3 (高级 AT+PID), 或 4 (高级 AT) 使用自动调节或高级自动调节时, 会自动确定积分时间, 并且不必由用户指定。

若设置动作模式 (S1+3) 为 0 (PID) 时未使用自动调节, 则从 1 ~ 65535 中选取所需值以指定 0.1s ~ 6553.5s 的积分时间给 S1+8 指定的数据寄存器。当 S1+8 设为 0 时, 将禁用积分运算。

若积分时间太短, 则积分运算会变大, 导致出现长周期异常。相反, 若积分时间太长, 则进程变量 (S1+0) 达到设置点 (S3) 的时间将变长。

在加比例带和减比例带之间的范围内执行积分运算。当由于外部噪音或或设置点改变导致进程变量 (S1+0) 超出比例带时, 将禁用积分运算。结果, 操作变量紧随设置点, 超调量和不足调量较小。

当 PID 动作在进行中时, 用户可以更改积分时间值。



S1+9 微分时间

当设置点 (S3) 更改或进程变量 (S1+0) 和设置点 (S3) 之间的差由于干扰而增加时, 可使用微分运算增加操作变量 (D1) 调整进程变量 (S1+0) 至设置点 (S3)。微分时间是确定微分运算量的参数。

当通过设置动作模式 (S1+3) 到 1 (AT+PID)、2 (AT)、3 (高级 AT+PID), 或 4 (高级 AT) 使用自动调节时, 会自动确定微分时间, 并且不必由用户指定。

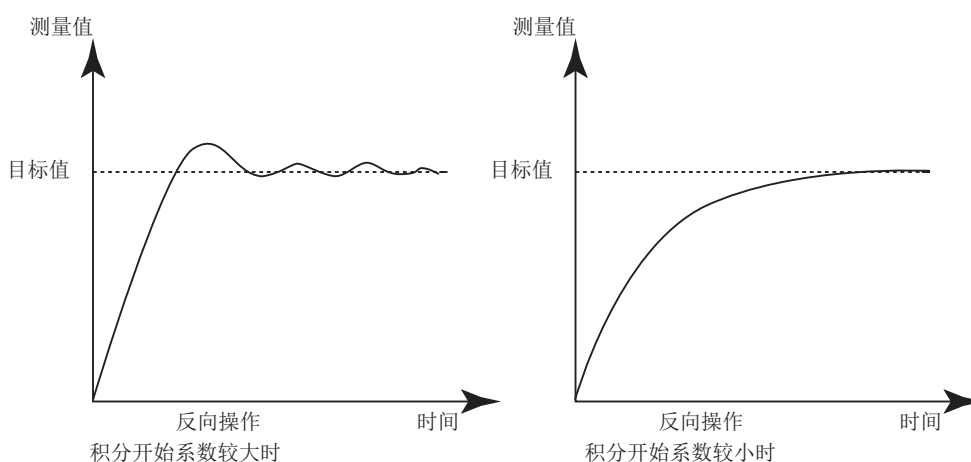
若设置动作模式 (S1+3) 为 0 (PID) 时未使用自动调节, 则从 1 ~ 65535 中选取所需值以指定 0.1s ~ 6553.5s 的微分时间给 S1+9 指定的数据寄存器。当 S1+9 设为 0 时, 将禁用微分运算。

当微分时间设为较大的值时, 微分运算变大。若微分运算太大, 则会导致短周期异常。

当 PID 动作在进行中时, 用户可以更改微分时间值。

S1+10 积分开始系数

该参数可设置开始积分操作的阈值。如果从开始执行 PID 指令时启用积分操作, 将引起过冲。通过使积分开始系数与比例项联动进行延迟, 可抑制过冲。如果过于缩小积分开始系数, 虽然不存在过冲, 但目标值和测量值之间会产生一定的偏差 (偏移量)。如果过于增大积分开始系数, 虽然偏移量会变少, 但会发生过冲。启用积分开始系数时, 请关闭控制继电器的积分开始系数设置继电器 (S2+3)。禁用积分开始系数时, 请打开控制继电器的积分开始系数设置继电器 (S2+3)。



控制模式 (S1 + 4) 的比例项为比例增益时:

请指定 1 ~ 100 (1% ~ 100%) 的值。0 或 101 以上作为 100% 执行。

控制模式 (S1 + 4) 的比例项为比例带时:

如下表所示, 系统软件版本不同, 预置值所对应的操作也有所不同。

型号	系统软件版本	运行
FC5A-C10R2x FC5A-C16R2x FC5A-C24R2x	245 以下	不使用该预置值。积分开始系数始终以 100% 运行。
FC5A-D16Rx1 FC5A-D32x3	246 以上	请指定 10001 ~ 10100 (1% ~ 100%) 的值。 10000 以下或 10101 以上以 100% 运行。
FC5A-D12x1E	130 以下	不使用该预置值。积分开始系数始终以 100% 运行。
	131 以上	请指定 10001 ~ 10100 (1% ~ 100%) 的值。 10000 以下或 10101 以上以 100% 运行。

S1+11 输入过滤器系数

输入过滤器可以缓和进程变量 (S4) 的波动。从 0 ~ 99 中选取所需值以指定输入过滤器系数的 0% ~ 99% 给 S1+11 指定的数据寄存器。当 S1+11 存储的值大于 99 时, 输入过滤器系数设为 99%。系数越大, 输入过滤器效果越明显。

在取样时若值更改, 输入过滤器可以有效地读取进程变量 (S4), 如温度数据。输入过滤器系数在自动调节和 PID 动作时生效。

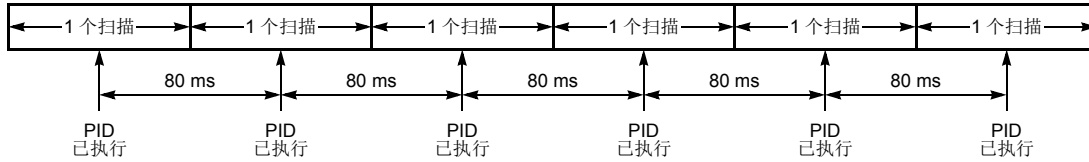
14: PID 指令

S1+12 取样周期

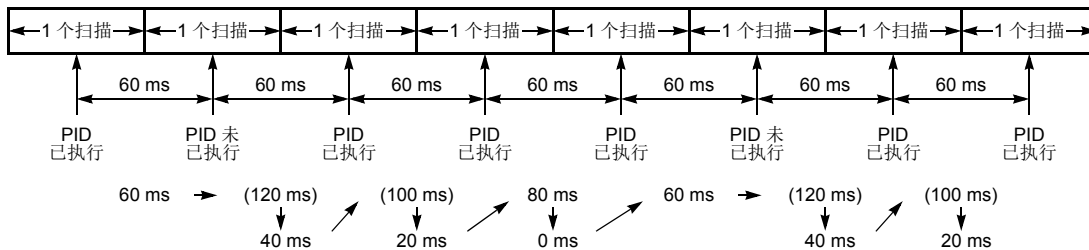
取样周期确定执行 PID 指令的时间间隔。从 1 ~ 10000 中选取所需值以从 0.01s ~ 100.00s 中选取取样周期，并将其指定给 S1+12 指定的数据寄存器。当 S1+12 存储 0 时，取样周期设为 0.01s。当 S1+12 存储的值大于 10000 时，取样周期设为 100.00s。

当取样周期的值小于扫描时间时，每次扫描都会执行 PID 指令。

示例 - 取样周期：40 ms，扫描时间：80 ms（取样周期 ≤ 扫描时间）



示例 - 取样周期：80 ms，扫描时间：60 ms（取样周期 > 扫描时间）



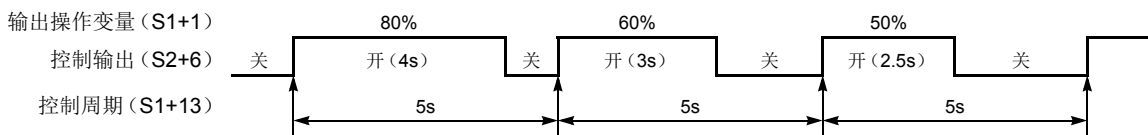
注释：当进行 PID 动作（动作状态 S1+2 是 5X 或 6X）时，可随时更改取样时间。取样周期以及积分时间（S1+8）和微分时间（S1+9）影响积分操作变量和微分操作变量。在 PID 动作中更改取样时间时，使用在 PID 动作开始时确定的取样时间来计算积分操作变量和微分操作变量。

S1+13 控制周期

控制周期可以决定控制输出（S2+6）的开/关循环时间，即根据 PID 操作计算出或从手动模式输出操作变量（S1+18）得出的输出操作变量（S1+1）打开或关闭输出。从 1 ~ 500 中选取所需值以从 0.1s ~ 50.0s 中选取控制周期，并将其指定给 S1+13 指定的数据寄存器。当 S1+13 存储 0 时，控制周期设为 0.1s。当 S1+13 存储的值大于 500 时，控制周期设为 50.0s。

控制输出（S2+6）的 ON 脉冲周期由控制周期（S1+13）和输出操作变量（S1+1）的乘积确定。

示例 - 控制周期：5s（S1+13 设为 50）



S1+14 上限报警值

上限报警值是进程变量（S1+0）的上限，超过此值即会报警。当进程变量的值大于或等于上限报警值时，将打开上限报警输出控制继电器（S2+4）。当进程变量的值小于上限报警值时，将关闭上限报警输出控制继电器（S2+4）。

当禁用线性变换（S1+4 设为 0 或 2）时，根据模拟量 I/O 模块类型，将 0 ~ 4095 或 50000 的所需上限报警值设置到 S1+14 指定的数据寄存器。当 S1+14 存储的值大于 4095 或 50000 时，上限报警值分别设为 4095 或 50000。

当启用线性变换（S1+4 设为 1 或 3）时，将 0 ~ 65535（字型数据类型）或 -32768 ~ 32767（整型数据类型）的所需上限报警值设置到 S1+14 指定的数据寄存器。上限报警值必须大于或等于线性变换最小值（S1+6），且小于或等于线性变换最大值（S1+5）。若上限报警值小于线性变换最小值（S1+6），则线性变换最小值将成为上限报警值。若上限报警值大于线性变换最大值（S1+5），则线性变换最大值将成为上限报警值。

S1+15 下限报警值

下限报警值是进程变量 (S1+0) 的下限, 低于此值即会报警。当进程变量的值小于或等于下限报警值时, 将打开下限报警输出控制继电器 (S2+5)。当进程变量的值大于下限报警值时, 将关闭下限报警输出控制继电器 (S2+5)。

当禁用线性变换 (S1+4 设为 0 或 2) 时, 根据模拟量 I/O 类型, 将 0 ~ 4095 或 50000 的所需下限报警值设置到 S1+15 指定的数据寄存器。当 S1+15 存储的值大于 4095 或 50000 时, 下限报警值分别设为 4095 或 50000。

当启用线性变换 (S1+4 设为 1 或 3) 时, 将 0 ~ 65535(字型数据类型) 或 -32768 ~ 32767(整型数据类型) 的所需下限报警值设置到 S1+15 指定的数据寄存器。下限报警值必须大于或等于线性变换最小值 (S1+6), 且小于或等于线性变换最大值 (S1+5)。若下限报警值小于线性变换最小值 (S1+6), 则线性变换最小值将成为下限报警值。若下限报警值大于线性变换最大值 (S1+5), 则线性变换最大值将成为下限报警值。

S1+16 输出操作变量上限

S1+16 指定的数据寄存器中包含的值可以两种方式指定输出操作变量 (S1+1) 的上限: 直接和成比例

S1+16 值为 0 ~ 100

当 S1+16 包含的值在 0 ~ 100 范围内时, 该值直接确定输出操作变量 (S1+1) 的上限。如果操作变量 (D1) 大于或等于上限值 (S1+16), 那么输出该上限值给输出操作变量 (S1+1)。从 0 ~ 100 中选取所需值作为 S1+16 指定的数据寄存器的输出操作变量上限。当 S1+16 存储的值大于 100 (10001 ~ 10099 除外) 时, 输出操作变量上限 (S1+16) 设为 100。输出操作变量上限 (S1+16) 必须大于输出操作变量下限 (S1+17)。

要启用操作变量上限, 请打开输出操作变量极限值启用控制继电器 (S2+2)。当 S2+2 关闭时, 输出操作变量上限 (S1+16) 无效。

S1+16 值 10001 ~ 10099 (禁用输出操作变量下限 S1+17)

当 S1+16 包含的值在 10001 ~ 10099 范围内时, 用该值减 10000 可确定输出操作变量 (S1+1) 与操作变量 (D1) (0 ~ 100) 的比率。可以通过以下等式计算输出操作变量 (S1+1):

$$\text{输出操作变量 (S1+1)} = \text{操作变量 (D1)} \times (\text{N}-10000)$$

N 是存储在输出操作变量上限 (S1+16) (10001 ~ 10099) 中的值。

如果操作变量 (D1) 大于或等于 100, 100 与 (N-10000) 的乘积将指定给输出操作变量 (S1+1)。如果 D1 小于或等于 0, 0 将指定给 S1+1。

要启用操作变量上限, 请打开输出操作变量极限值启用控制继电器 (S2+2)。当 S2+2 关闭时, 输出操作变量上限 (S1+16) 无效。

当 S1+16 设为 10001 ~ 10099 内的值时, 将禁用输出操作变量下限 (S1+17)。

S1+17 输出操作变量下限

S1+17 指定的数据寄存器中包含的值可指定输出操作变量 (S1+1) 的下限。从 0 ~ 100 中选取所需值作为 S1+17 指定的数据寄存器的输出操作变量下限。当 S1+17 存储的值大于 100 时, 输出操作变量下限设为 100。输出操作变量下限 (S1+17) 必须小于输出操作变量上限 (S1+16)。

要启用输出操作变量下限, 请打开输出操作变量极限值启用控制继电器 (S2+2), 并设置输出操作变量上限 (S1+16) 为 10001 ~ 10099 以外的值。当操作变量 (D1) 小于或等于指定的下限时, 该下限值将指定给输出操作变量 (S1+1)。

当输出操作变量极限值启用控制继电器 (S2+2) 关闭时, 输出操作变量下限 (S1+17) 无效。

14: PID 指令

S1+18 手动模式输出操作变量

手动模式输出操作变量指定手动模式下的输出操作变量 (0 ~ 100)。从 0 ~ 100 中选择所需值作为手动模式输出操作变量指定给 S1+18 指定的数据寄存器。当 S1+18 存储的值大于 100 时, 手动模式输出操作变量设为 100。

要启用手动模式, 请打开自动 / 手动模式控制继电器 (S2+1)。在手动模式下, 将禁用 PID 动作。输出手动模式输出操作变量 (S1+18) 到输出操作变量 (S1+1) 和模拟量输出模块 (S1+24) 的输出操作变量。根据控制周期 (S1+13) 和手动模式输出操作变量 (S1+18) 打开和关闭控制输出 (S2+6)。

S1+18 值不影响操作值 (D1) 和输出操作变量 % (S1+23)。

自动调节 (AT) 和高级自动调节 (高级 AT)

当选择自动调节, 且动作模式 (S1+3) 设为 1 (AT+PID) 或 2 (AT) 时, 在启动 PID 控制以自动确定 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9) 和控制动作 (S2+0)] 后执行自动调节。

MicroSmart 使用步进响应方式执行自动调节。要启用自动调节, 请在执行 PID 指令 [如 AT 取样周期 (S1+19)、AT 控制周期 (S1+20)、AT 设置点 (S1+21) 和 AT 输出操作变量 (S1+22)] 前设置自动调节的四个参数。

当选择高级自动调节并且动作模式 (S1+3) 设置为 3(高级 AT+PID) 或 4(高级 AT) 时, 将自动确定最大 AT 参数, 不需要用户设置。只有在使用高级自动调节并且 S1+3 设置到 4(高级 AT) 时, 用户需要指定 AT 设置点 (S1+21)。

AT 参数

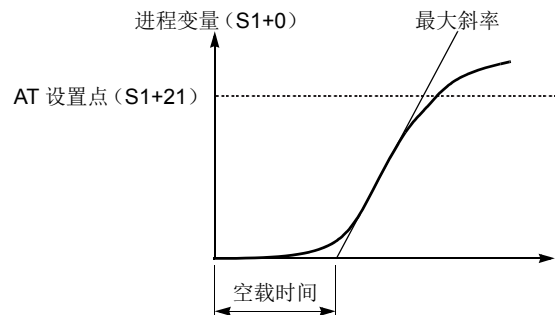
在指定自动调节前, 用户必须根据下表总结指定 AT 参数。

动作模式 (S1+3)	AT 取样周期 (S1+19)	AT 控制周期 (S1+20)	AT 设置点 (S1+21)	AT 输出操作变量 (S1+22)
0: PID 动作	—	—	—	—
1: AT (自动调节) + PID 动作	用户指定	用户指定	用户指定	用户指定 (0 ~ 100)
2: AT (自动调谐)				
3: 高级 AT + PID 动作	自动	自动	自动	自动
4: 高级 AT	自动	自动	用户指定	自动

步进响应方式

MicroSmart 使用布局响应方式执行自动调节, 并自动确定 PID 参数, 如比例项 (S1+7)、积分时间 (S1+8)、微分时间 (S1+9) 和控制动作 (S2+0)。按以下步骤执行自动调节:

1. 在进程变量达到 AT 设置点 (S1+21) 之前计算进程变量 (S1+0) 的最大斜率。
2. 基于得出的最大斜率计算空载时间。
3. 基于最大斜率和空载时间计算四个 PID 参数。



S1+19 AT 取样周期

AT 取样周期可决定在自动调节中取样的时间间隔。当使用自动调节并且动作模式 (S1+3) 设置到 1 (AT+PID) 或 2(AT) 时, 将指定 AT 取样周期 0.01s ~ 100.00s 的 1 ~ 10000 所需值给 S1+19 指定的数据寄存器。当 S1+19 存储 0 时, AT 取样周期设为 0.01s。当 S1+19 存储的值大于 10000 时, AT 取样周期设为 100.00s。

设置 AT 取样周期为一个较大的值以确保在正向控制动作 (S2+0 打开) 时当前进程变量小于或等于先前进程变量, 或在反向控制动作 (S2+0 关闭) 时当前进程变量大于或等于先前进程变量。

当使用高级自动调节并且动作模式 (S1+3) 设置为 3(高级 AT+PID) 或 4(高级 AT) 时, 将自动确定 AT 取样周期, 不需要用户设置。

S1+20 AT 控制周期

AT 控制周期可决定在自动调节时控制输出 (S2+6) 的开 / 关循环时间。有关控制输出操作, 请参阅第 14-10 页上的“控制周期”。

当使用自动调节并且动作模式 (S1+3) 设置到 1 (AT+PID) 或 2(AT) 时, 将指定 AT 控制周期 0.1s ~ 50.0s 的 1 ~ 500 所需值给 S1+20 指定的数据寄存器。当 S1+20 存储 0 时, AT 控制周期设为 0.1s。当 S1+20 存储的值大于 500 时, AT 控制周期设为 50.0s。

当使用高级自动调节并且动作模式 (S1+3) 设置为 3(高级 AT+PID) 或 4(高级 AT) 时, 将自动确定 AT 控制周期, 不需要用户设置。

S1+21 AT 设置点

在执行自动调节时, AT 输出操作变量 (S1+22) 将指定给输出操作变量 (S1+1) 直至进程变量 (S1+0) 达到 AT 设置点 (S1+21)。当进程变量 (S1+0) 达到 AT 设置点 (S1+21) 时, 自动调节完成, 输出操作变量 (S1+1) 降为零。当选择 PID 动作, 同时动作模式 (S1+3) 设为 1(AT+PID) 或 3(高级 AT+PID) 时, PID 操作将立即执行。

当动作模式 (S1+3) 设置到 1(AT+PID)、2(AT) 或 4(高级 AT) 时, 将所需 AT 设置点设置到 S1+21 指定的数据寄存器。当动作模式 (S1+3) 设置到 3(高级 AT+PID) 时, 将自动确定设置点, 不需要用户设置。

当禁用线性变换 (S1+4 设为 0 或 2) 时, 根据模拟量 I/O 模块类型, 将 0 ~ 4095 或 50000 的所需 AT 设置点设置到 S1+21 指定的数据寄存器。当 S1+21 存储的值大于 4095 或 50000 时, AT 设置点设为 4095 或 50000。

当启用线性变换 (S1+4 设为 1 或 3) 时, 将 0 ~ 65535(字型数据类型) 或 -32768 ~ 32767(整型数据类型) 的所需 AT 设置点设置到 S1+21 指定的数据寄存器。AT 设置点必须大于或等于线性变换最小值 (S1+6), 且小于或等于线性变换最大值 (S1+5)。

在正向控制动作 (请参阅第 14-15 页) 中, 设置 AT 设置点 (S1+21) 为明显小于自动调节开始处的进程变量 (S4) 的值。在反向控制动作 (S2+0 关闭) 中, 设置 AT 设置点 (S1+21) 为明显大于自动调节开始处的进程变量 (S4) 的值, 否则进程变量 (S1+0) 不能达到 AT 设置点 (S1+21), 并且不能确定 AT 参数。

S1+22 AT 输出操作变量

AT 输出操作变量指定在自动调节时输出操作变量 (0 ~ 100) 的量。在使用自动调节时, 从 0 ~ 100 中选取所需 AT 输出操作变量以指定给 S1+22 指定的数据寄存器。当 S1+22 存储大于 100 的值时, AT 输出操作变量设为 100。

在执行自动调节时, AT 输出操作变量 (S1+22) 的指定值将指定给输出操作变量 (S1+1), 并根据 AT 控制周期 (S1+20) 和 AT 输出操作变量 (S1+22) 打开和关闭控制输出 (S2+6)。要在自动调节时保持控制输出 (S2+6), 请设置 S1+22 的值为 100。

当使用高级自动调节并且动作模式 (S1+3) 设置为 3(高级 AT+PID) 或 4(高级 AT) 时, 将自动确定 AT 输出操作变量, 不需要用户设置。

14: PID 指令

S1+23 输出操作变量 %

当进行 PID 动作时，S1+23 指定的数据寄存器保持操作变量 -32768 ~ 32767 (-327.68% ~ 327.67%)，值保留两位小数。

当启用手动模式，且自动 / 手动模式控制继电器 (S2+1) 设为开时，S1+23 存储不定值。

当进行自动调节或高级自动调节时，S1+23 存储不定值。

S1+24 模拟量输出模块的输出操作变量

当进行 PID 动作时，根据模拟量 I/O 模块类型，S1+24 指定的数据寄存器保持 0 ~ 4095 或 50000 的一个值。从存储在 S1+1 中 0 ~ 100 的值转换值代表 0% ~ 100% 的输出操作变量。

当启用手动模式，且自动 / 手动模式控制继电器 (S2+1) 设为开时，S1+24 保持从手动模式输出操作变量 (S1+18) 转换的 0 ~ 4095 或 50000。

当进行自动调节或高级自动调节时，S1+24 保持从 AT 输出操作变量 (S1+22) 读取的 0 ~ 4095 或 50000 中的一个值。

S1+25 比例带补偿值

当选择比例带 (S1+4 设为 2 或 3) 时，-100% ~ 100% 的补偿可移位 0% ~ 100% 的输出操作变量 (S1+1)。在执行自动调节前将所需的 -100 ~ 100 补偿值设置到 S1+25 指定的数据寄存器。

当选择比例增益 (S1+4 设为 0 或 1) 时，比例带补偿值 (S1+25) 无效。

S1+26 微分增益

可从 0% ~ 100% 中选择微分增益。当微分增益设为小值时，输出操作变量 (S1+1) 易受外部噪音或设置点变化影响。当微分增益设为较大值时，输出操作变量 (S1+1) 不易受外部噪音或设置点变化影响，但是在正常操作中影响稳定性。在执行自动调节前将所需的 0 ~ 100 微分增益设置到 S1+26 指定的数据寄存器。

当进程变量波动或易受噪音影响时，建议值为 20% ~ 30%。

源设备 S2 (控制继电器)

在根据需要执行 PID 指令之前，打开或关闭以 S2 指定的设备开始的输出或内部继电器。设备 S2+4 ~ S2+7 为只读，可反映 PID 和自动调节状态。

设备	功能	说明	R/W
S2+0	控制动作	ON: 正向控制动作 OFF: 反向控制动作	R/W
S2+1	自动 / 手动模式	ON: 手动模式 OFF: 自动模式	R/W
S2+2	输出操作变量极限值启用	ON: 启用输出操作变量上限和下限 (S1+16 和 S1+17) OFF: 禁用输出操作变量上限和下限 (S1+16 和 S1+17)	R/W
S2+3	积分开始系数禁用	ON: 禁用积分开始系数 (S1+10) OFF: 启用积分开始系数 (S1+10)	R/W
S2+4	上限报警输出	ON: 当进程变量 (S1+0) \geq 上限报警值 (S1+14) 时 OFF: 当进程变量 (S1+0) $<$ 上限报警值 (S1+14) 时	R
S2+5	下限报警输出	ON: 当进程变量 (S1+0) \leq 下限报警值 (S1+15) 时 OFF: 当进程变量 (S1+0) $>$ 下限报警值 (S1+15) 时	R
S2+6	控制输出	根据 AT 参数或 PID 计算结果打开和关闭	R
S2+7	AT 完成输出	当 AT 完成或失败时打开，并保持至复位	R

S2+0 控制动作

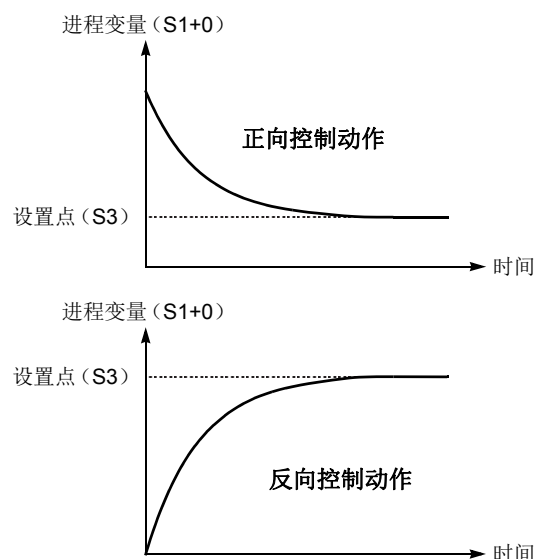
当执行自动调节，并且动作模式 (S1+3) 已设为 1 (AT+PID)、2 (AT)、3 (高级 AT+PID) 或 4 (高级 AT) 时，将自动确定控制动作。当自动调节导致正向控制动作时，S2+0 指定的控制动作控制继电器打开。当自动调节导致反向控制动作时，S2+0 指定的控制动作控制继电器关闭。根据得到的控制动作（在 PID 动作中该动作持续生效）执行 PID 动作。

在没有执行自动调节时，若动作模式 (S1+3) 已设为 0 (PID)，则在执行 PID 指令前打开或关闭控制动作控制继电器 (S2+0) 可分别选择正向或反向控制动作。

在正向控制动作中，若进程变量 (S1+0) 大于设置点 (S3)，则进程变量 (D1) 增大。在正向控制动作中执行温度控制以进行冷却。

在反向控制动作中，若进程变量 (S1+0) 小于设置点 (S3)，则操作变量 (D1) 增大。在反向控制动作中执行温度控制以进行加热。

在正向或反向控制动作中，若进程变量 (S1+0) 和设置点 (S3) 之间的差增大，则操作变量 (D1) 增大。

**S2+1 自动 / 手动模式**

要选择自动模式，请在启动 PID 指令前或后关闭 S2+1 指定的自动 / 手动模式控制继电器。在自动模式下，执行 PID 动作，且操作变量 (D1) 将存储 PID 计算结果。根据控制周期 (S1+13) 和输出操作变量 (S1+1) 打开和关闭控制输出 (S2+6)。

要选择手动模式，请打开自动 / 手动模式控制继电器 (S2+1)。使用手动模式时，在启用手动模式之前为手动模式输出操作变量 (S1+18) 设置所需值。在手动模式下，输出操作变量 (S1+1) 存储手动模式输出操作变量 (S1+18)，模拟量输出模块 (S1+24) 的输出操作变量存储从手动模块输出操作变量 (S1+18) 转换的 0 ~ 4095 或 50000 的一个值。根据控制周期 (S1+13) 和手动模式输出操作变量 (S1+18) 打开和关闭控制输出 (S2+6)。

S1+18 值不影响操作值 (D1) 和输出操作变量 % (S1+23)。

当自动调节在进行中时，不能启用手动模式。仅当自动调节完成后，才能启用自动或手动模式。还可以在执行 PID 指令时切换自动 / 手动模式。

S2+2 输出操作变量极限值启用

使用输出操作变量极限值启用控制继电器 (S2+2) 启用或禁用输出操作变量上限 (S1+16) 和输出操作变量下限 (S1+17)。

要启用输出操作变量上 / 下限，请打开 S2+2。

要禁用输出操作变量上 / 下限，请关闭 S2+2。

S2+3 积分开始系数禁用

使用积分开始系数禁用控制继电器启用或禁用积分开始系数 (S1+10)。

要启用积分开始系数 (S1+10)，请关闭 S2+3；可通过积分开始系数 (S1+10) 的指定启用积分项。

要禁用积分开始系数 (S1+10)，请打开 S2+3；将在 PID 动作开始时启用积分项。

S2+4 上限报警输出

当进程变量 (S1+0) 大于或等于上限报警值 (S1+14) 时，将打开上限报警输出控制继电器 (S2+4)。当 S1+0 小于 S1+14 时，S2+4 将关闭。

S2+5 下限报警输出

当进程变量 (S1+0) 小于或等于下限报警值 (S1+15) 时，将打开下限报警输出控制继电器 (S2+5)。当 S1+0 大于 S1+15 时，S2+5 将关闭。

14: PID 指令

S2+6 控制输出

在自动模式下自动调节时，若自动 / 手动模式控制继电器 (S2+1) 已关闭，将根据 AT 控制周期 (S1+20) 和 AT 输出操作变量 (S1+22) 打开和关闭控制输出 (S2+6)。

在自动模式下执行 PID 动作时，若自动 / 手动模式控制继电器 (S2+1) 已关闭，将根据 PID 动作计算的控制周期 (S1+13) 和输出操作变量 (S1+1) 打开和关闭控制输出 (S2+6)。

当进行高级自动调节时，控制输出 (S2+6) 保持打开。

在手动模式下，若自动 / 手动模式控制继电器 (S2+1) 已关闭，将根据控制周期 (S1+13) 和手动模式输出操作变量 (S1+18) 打开和关闭控制输出 (S2+6)。

S2+7 AT 完成输出

当自动调节完成或失败时，AT 完成输出控制继电器 (S2+7) 会持续打开直至复位。运行状态代码存储在运行状态控制寄存器 (S1+2) 中。请参阅第 14-4 页。

源设备 S3 (设置点)

执行 PID 动作以调整进程变量 (S1+0) 至设置点 (S3)。

当禁用线性变换 (S1+4 设为 0 或 2) 时，根据模拟量 I/O 模块类型，将 0 ~ 4095 或 50000 的所需设置点设置到 S3 指定的设备。有效设备为寄存器和常量。

当启用线性变换 (S1+4 设为 1 或 3) 时，将寄存器分配给设备 S3，并从 0 ~ 65535(字型数据类型) 或 -32768 ~ 32767(整型数据类型) 中选取所需设置点的值设置到 S3 指定的数据寄存器。设置点值 (S3) 必须大于或等于线性变换最小值 (S1+6)，并小于或等于线性变换最大值 (S1+5)。

当指定给设置点的值无效时，PID 动作将停止，错误代码会存储至 S1+2 指定的数据寄存器中。请参阅第 14-4 页上的运行状态。

源设备 S4 (转换前的进程变量)

PID 指令设计用于使用模拟量 I/O 模块的模拟量输入数据作为进程变量。模拟量 I/O 模块将输入信号转换为 0 ~ 4095 或 50000 范围内的数字值，并根据模拟量 I/O 模块的安装位置和连接模拟量输入源的模拟量输入频道将该数字值存储至寄存器中。分配寄存器给设备 S4 以存储进程变量。

有关要指定为源设备 S4 的数据寄存器，请参阅第 9-3 页 (基本卷)。把“设置参数”对话框中数据下显示的数据寄存器编号指定为 PID 指令的源设备 (进程变量)。所选数据寄存器中的模拟量输入数据用作 PID 指令的进程变量。

目标设备 D1 (操作变量)

目标设备 D1 指定的数据寄存器将存储 PID 动作计算的操作变量 (-32768 ~ 32767)。如果计算结果小于 -32768，则 D1 将存储 -32768。若计算结果大于 32767，则 D1 存储 32767。如果计算结果小于 -32768 或大于 32767，则 PID 动作继续。

在进行 PID 动作时，若禁用输出操作变量 (S2+2 设为关闭)，由 S1+1 指定的数据寄存器会保留操作变量 (D1) 的值 (0 ~ 100)，并忽略小于 0 和大于 100 的值。S1+1 中的百分比值可按照与控制周期 (S1+13) 的比例确定输出控制 (S2+6) 的打开时间。

当启用输出操作变量极限值 (S2+2 设为打开) 时，根据下表中归纳的输出操作变量上限 (S1+16) 和输出操作变量下限 (S1+17) 操作变量 (D1) 将存储至输出操作变量 (S1+1) 中。

当启用手动模式，且自动 / 手动模式控制继电器 (S2+1) 设为开时，S1+1 存储从手动模式输出操作变量 (S1+18) 中读取的 0 ~ 100，而 D1 存储不定值，并且 D1 存储与 S1+18 值无关的不定值。

在进行自动调节 (AT) 时，S1+1 会存储从 AT 输出动作变量 (S1+22) 中读取的 0 ~ 100，而 D1 存储不定值。当进行高级自动调节时，S1+1 和 D1 存储不定值。

输出操作变量值示例

输出操作变量极限值 启用 (S2+2)	输出操作变量上限 (S1+16)	输出操作变量下限 (S1+17)	操作变量 (D1)	输出操作变量 (S1+1)
关 (禁用)	—	—	≥ 100	100
			1 ~ 99	1 ~ 99
			≤ 0	0
开 (启用)	50	25	≥ 50	50
			26 ~ 49	26 ~ 49
			≤ 25	25
	10050	—	≥ 100	50
			1 ~ 99	$(1 \sim 99) \times 0.5$
			≤ 0	0

重要信息

根据控制周期 (S1+13) 和输出操作变量 (S1+1) 打开和关闭控制输出 (S2+6)。当反馈系统包含控制输出 (S2+6) 时, 对某些控制对象可能不能实现最佳控制, 建议使用操作变量 (D1) 的计算结果编写反馈控制系统。

使用 PID 指令注意事项:

- 因为 PID 指令需要持续运行, 所以要保持 PID 指令的起始输入。
- 当 PID 指令的起始输入打开时, 上限报警输出 (S2+4) 和下限报警输出 (S2+5) 工作。但是当由于控制寄存器 (S1+0 ~ S1+26) 中出现数据错误, 或 PID 指令的起始输入关闭, 而导致 PID 指令执行错误 (S1+2 存储 100 或更多) 时, 这些报警输出不工作。请使用单独的程序以监控进程变量 (S4)。
- 当出现 PID 执行错误 (S1+2 存储 100 或更多) 或自动调节完成时, 操作变量 (D1) 存储 0, 且控制输出 (S2+6) 关闭。
- 请勿在程序分支指令中使用 PID 指令: LABEL、LJMP、LCAL、LRET、JMP、JEND、MCS 和 MCR。在这些指令中 PID 指令不能正确工作。
- PID 指令使用设置点 (S3) 和进程变量 (S4) 之间的差作为输入, 并根据 PID 参数 [如比例项 (S1+7)、积分时间 (S1+8) 和微分时间 (S1+9)] 计算操作变量 (D1)。当由于干扰而使设置点 (S3) 或进程变量 (S4) 发生变化时, 将导致脉冲跳增或负脉冲信号。在将 PID 控制输入实际应用程序中之前, 请预期设置点和进程变量值的变动 (干扰), 进行充分的模拟测试。
- PID 参数 [如比例项 (S1+7)、积分时间 (S1+8) 和微分时间 (S1+9)] 由自动调节决定, 可能不总是适用于实际应用的最佳值。调整这些参数以确保获得最佳效果。在确定了最佳 PID 参数后, 除非控制对象更改, 在一般运行中请尽量仅执行 PID 动作。

14: PID 指令

应用程序示例

下列两个应用程序示例演示高级自动调节和 PID 动作保持加热器温度在 200 °C。

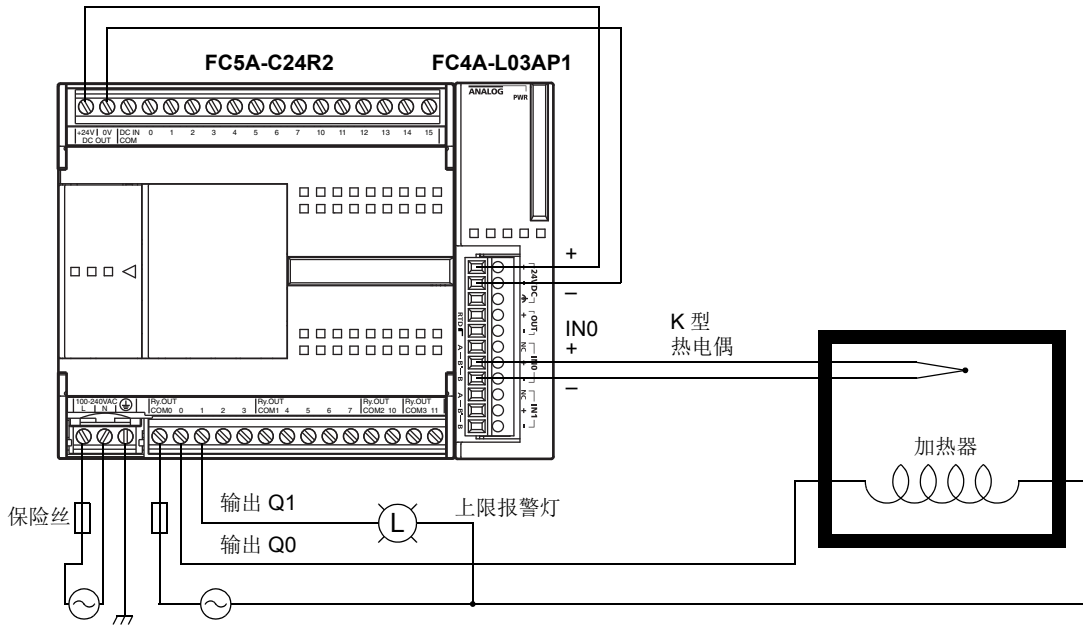
在两个示例中，当启动程序时，使用输入模拟量输入模块的温度数据，PID 指令首先执行高级自动调节确定 AT 参数，如 AT 取样周期、AT 控制周期、AT 设置点和 AT 输出操作变量，然后执行自动调节确定 PID 参数，如比例项、积分时间、微分时间、取样周期、控制周期和控制动作。当自动调节完成时，PID 动作将使用得出的 PID 参数开始控制温度到 200 °C。

示例 1: 使用继电器输出的开 / 关控制

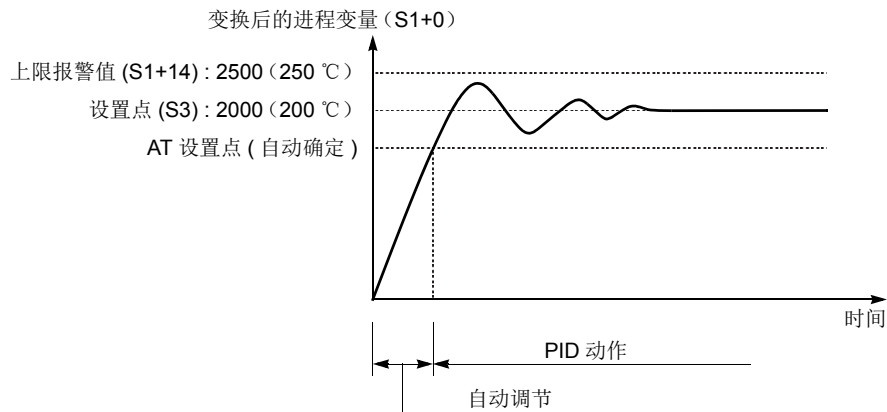
根据 PID 动作计算出的输出操作变量打开和关闭加热器。当加热器温度高于或等于 250 °C 时，上限报警输出将打开报警灯。

还会监控模拟量输入操作状态以强制关闭加热器电源及强制打开上限报警灯。

系统设置



通过自动调节和 PID 动作控制温度

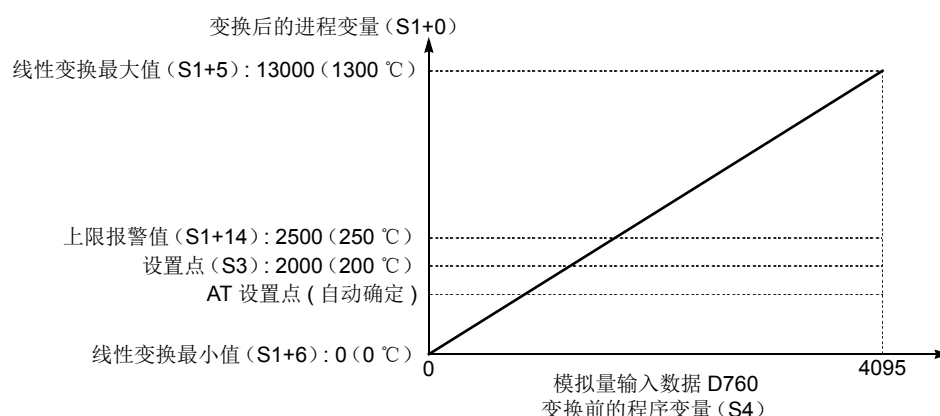


设备设置

设备	功能	说明	设备地址 (值)
S1+3	动作模式	高级 AT (自动调节) + PID 动作	D3 (3)
S1+4	控制模式	启用线性变换, 比例带	D4 (3)
S1+5	线性变换最大值	1300 °C	D5 (13000)
S1+6	线性变换最小值	0 °C	D6 (0)
S1+10	积分开始系数	100%	D10 (100)
S1+11	输入过滤器系数	70%	D11 (70)
S1+14	上限报警值	250 °C	D14 (2500)
S1+15	下限报警值	0 °C	D15 (0)
S1+25	比例带补偿值	0%	D25 (0)
S1+26	微分增益	0%	D26 (0)
S2+2	输出操作变量极限值启用	禁用输出操作变量极限值	M2 (关)
S2+3	积分开始系数禁用	启用积分开始系数 (S1+10)	M3 (关)
S2+4	上限报警输出	ON: 当温度 ≥ 250 °C 时 OFF: 当温度 < 250 °C 时	M4
S2+6	控制输出	高级自动调节时保持打开; 在执行 PID 操作时, 根据控制周期 (S1+13) 和输出操作变量 (S1+1) 继续和停止。	M6
S3	设置点	200 °C	D100 (2000)
S4	进程变量	模拟量 I/O 模块 1 的模拟量输入数据, 模拟量 输入频道 0; 存储 0 ~ 4095	D760
	模拟量输入操作状态	存储 0 ~ 5	D761
	模拟量输入信号类型	类型 K 温度计	D762 (2)
	模拟量输入数据类型	12 位数据 (0 ~ 4095)(注意)	D763 (0)
D1	操作变量	存储 PID 计算结果	D50
	PID 起始输入	开始执行 PID 指令	I0
	加热器电源开关	通过控制输出 M6 打开和关闭	Q0
	上限报警灯	通过上限报警输出 M4 和模拟量输入错误 M11 打开和关闭	Q1
	模拟量输入错误	当模拟量输入操作状态 D761 为 3 以上时打开	M11

注释: 当 PID 指令使用模拟量 I/O 模块 FC4A-L03AP1 时, 选择二进制数据确保进程变量选取 0 ~ 4095 中的值。

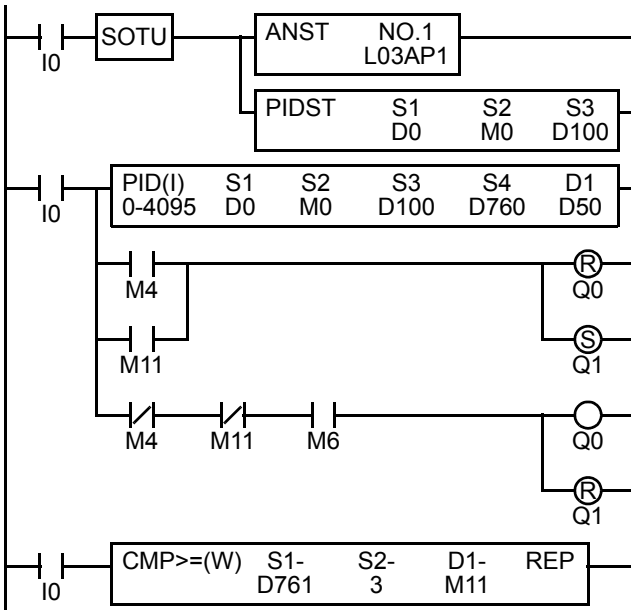
模拟量输入数据与变换后的进程变量



14: PID 指令

梯形图程序

如下所示的梯形图描述使用 PID 指令的示例。必须根据应用修改用户程序，并在实际操作前执行模拟量演示。



当 I0 打开时，ANST(模拟量宏)指令存储模拟量 I/O 模块功能参数。

PIDST(PID 宏)指令也存储 PID 功能参数。

D760 是模拟量 I/O 模块 1 的模拟量输入数据，模拟量输入频道 0；存储 0 ~ 4095

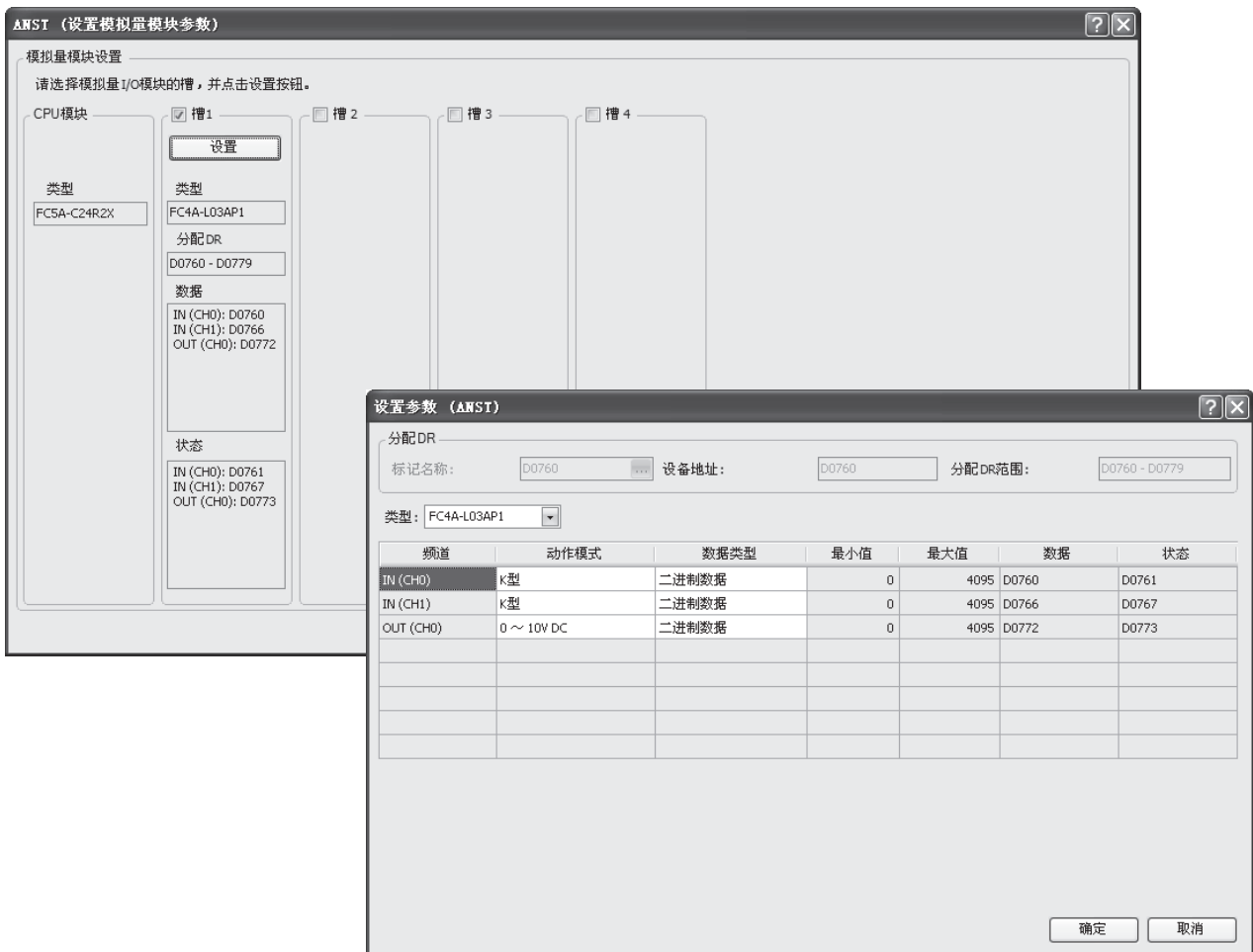
当打开内部继电器 M4(上限报警输出)或 M11(模拟量输入操作状态为 3 以上)时，将关闭 Q0(加热器输出)并打开输出 Q1(上限报警灯)。

当 M4 和 M11 关闭并且打开 M6(控制输出)时，将打开 Q0(加热器输出)并关闭输出 Q1(上限报警灯)。

当 D761(模拟量输入操作状态)存储 3 以上时，将打开内部继电器 M11。

设置模拟量模块参数 (ANST) 对话框

WindLDR 有一个为模拟量 I/O 模块编写参数的宏。将光标放在要插入 ANST 指令的位置，单击鼠标右键并选择宏指令 >ANST(设置模拟量模块参数)。在 ANST 对话框中按下槽 1 下的设置按钮，然后按如下方式编写。



设置 PID 参数 (PIDST) 对话框

将光标放在要插入 PIDST 指令的位置，单击鼠标右键并选择宏指令 >PIDST(设置 PID 参数)。在 PIDST 对话框中按如下方式编写。

跟使用 PID 指令一样选择选项和设备。

The screenshot shows the 'PIDST (设置PID参数)' dialog box. It includes fields for '模块类型' (Module Type) set to '0-4095' and '数据类型' (Data Type) set to '整数(I)'. A table lists devices: S1 (D0000), S2 (M0000), and S3 (D0100). The 'PID参数' section contains settings for '操作模式' (Operation Mode) set to '扩展 AT + PID 操作', '控制操作' (Control Operation) set to '正向控制操作', '积分运算' (Integral Operation) set to '启用' with a coefficient of '100', and '比例' (Proportional) set to '比例带'. The 'PID动作参数' section includes '设置点' (Setpoint) at '2000', '取样周期' (Sampling Period) at '1', '控制周期' (Control Period) at '1', '比例带' (Proportional Band) at '1', '积分时间' (Integral Time) at '1', '微分时间' (Derivative Time) at '1', and '微分增益' (Derivative Gain) at '0'. The 'AT参数' section includes '设置点' (Setpoint) at '4095', '取样周期' (Sampling Period) at '10000', '控制周期' (Control Period) at '500', and '输出操作变量' (Output Operation Variable) at '100'. The '输入设置' section includes '线性转换' (Linear Conversion) set to '启用', '最大值' (Maximum) at '13000', '最小值' (Minimum) at '0', '输入过滤器系数' (Input Filter Coefficient) at '70', '报警上限值' (Alarm Upper Limit) at '2500', and '报警下限值' (Alarm Lower Limit) at '0'. The '输入设置' section also includes '操纵变量极限值' (Manipulation Variable Limit) set to '禁用', '上限' (Upper Limit) at '100', '下限' (Lower Limit) at '0', and '比例带偏差值' (Proportional Band Deviation) at '0'. Callouts S1+3 through S1+26 point to various fields in the dialog.

S1+3	操作模式	S1+14	报警上限值
S1+4	控制模式	S1+15	报警下限值
S1+5	线性变换最大值	S1+25	比例带偏差值
S1+6	线性变换最小值	S1+26	微分增益
S1+10	积分启用系数	S2+2	操作变量极限值
S1+11	输入过滤器系数	S2+3	积分运算
		S3	设置点

PID 控制 (PID) 对话框

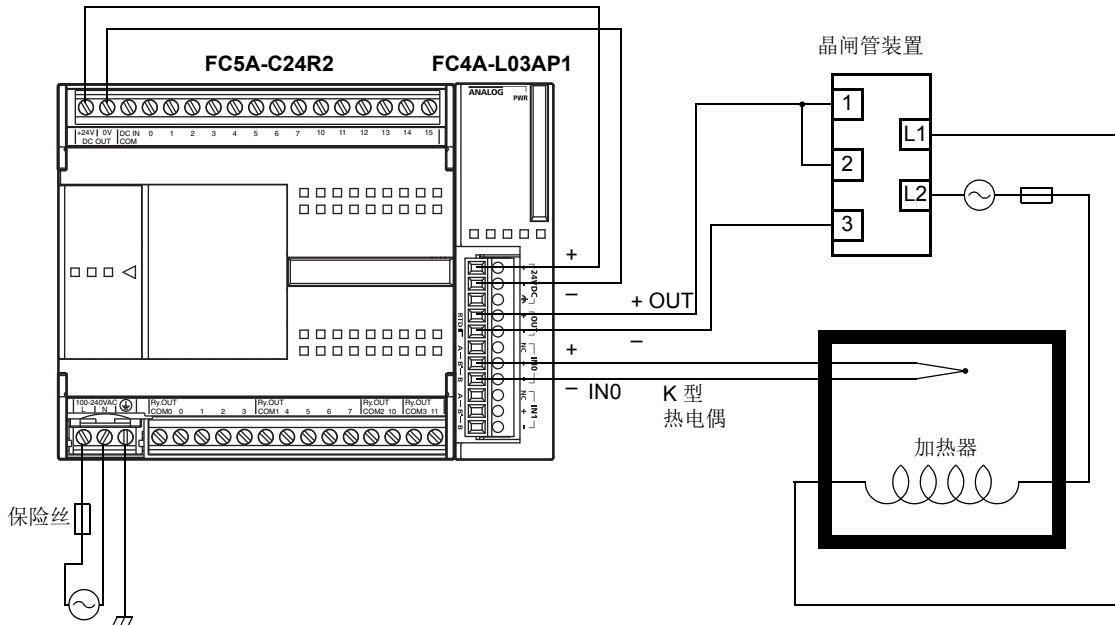
The screenshot shows the 'PID (PID控制)' dialog box. It includes a '类型' (Type) dropdown set to 'PID (PID控制)'. The '标记名称' (Tag Name) and '设备地址' (Device Address) fields are populated with values for devices S1, S2, S3, S4, and D1. The '模块类型' (Module Type) is set to '0-4095' and '数据类型' (Data Type) is set to '整数(I)'. The '注释' (Comment) field is empty. The '模块类型(0-4095): FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, FC4A-K1A1' is listed at the bottom. Callouts S1 through D1 point to the respective device selection fields.

14: PID 指令

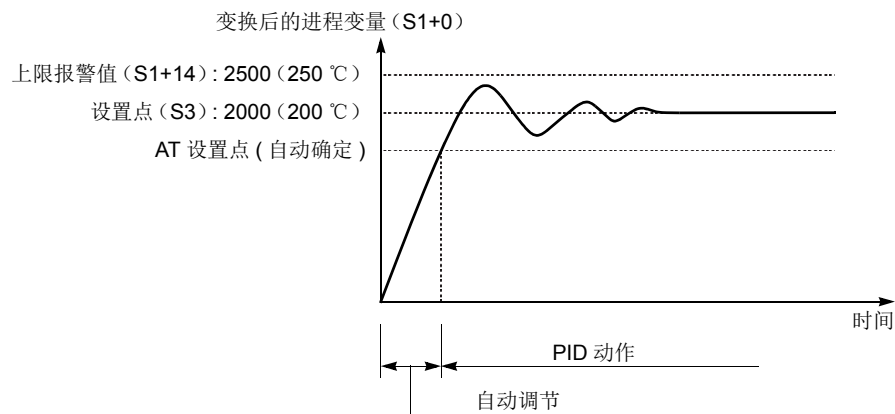
示例 2: 使用模拟量输出的开 / 关控制

当 PID 指令的模拟量输出模块 (S1+24) 的输出操作变量传送到模拟量输出数据 (D772)，并且模拟量 I/O 模块发送 0 ~ 10V DC 的电压输出。模拟量输出然后连接到使用相控制控制 AC 电源的晶闸管装置。

系统设置



通过自动调节和 PID 动作控制温度



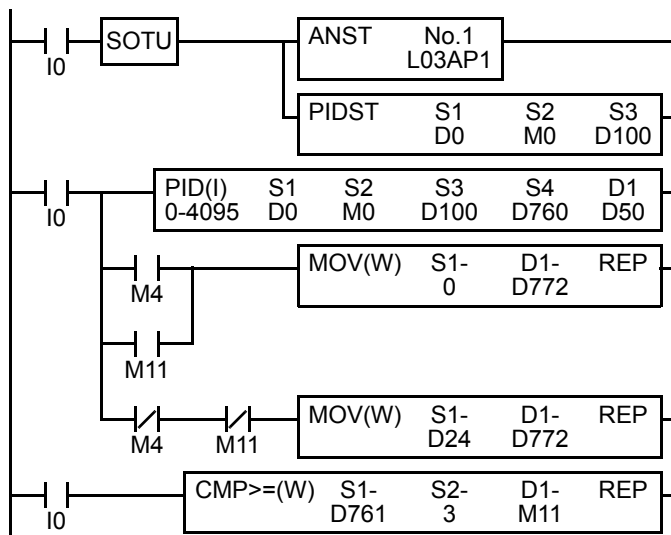
模拟量 I/O 模块的设备设置

模拟量频道	功能	说明	设备地址 (值)
输入频道 0	模拟量输入数据	模拟量 I/O 模块 1 的模拟量输入数据, 模拟量输入频道 0; 存储 0 ~ 4095	D760
	模拟量输入操作状态	存储 0 ~ 5	D761
	模拟量输入信号类型	类型 K 温度计	D762 (2)
	模拟量输入数据类型	12 位数据 (0 ~ 4095)	D763 (0)
输出	模拟量输出数据	0 ~ 4095	D772
	模拟量输出操作状态	存储 0 ~ 4	D773
	模拟量输出信号类型	电压输出 (0 ~ 10V DC)	D774 (0)
	模拟量输出数据类型	12 位数据 (0 ~ 4095)	D775 (0)

梯形图程序

如下所示的梯形图描述使用 PID 指令的示例。必须根据应用修改用户程序，并在实际操作前执行模拟量演示。

在 ANST(设置模拟量模块参数)、PIDST(设置 PID 参数) 和 PID(PID 控制) 的对话框中编写指令与前面的示例相同。



当 I0 打开时，ANST(模拟量宏) 指令存储模拟量 I/O 模块功能参数。

PIDST(PID 宏) 指令也存储 PID 功能参数。

D760 是模拟量 I/O 模块 1 的模拟量输入数据，模拟量输入频道 0；存储 0 ~ 4095

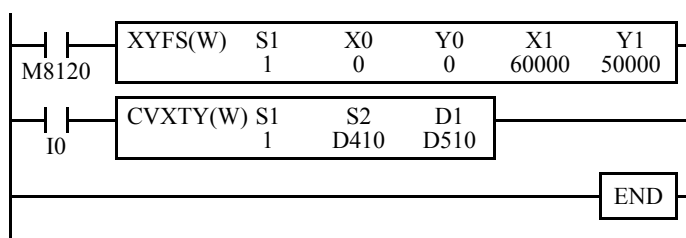
当打开内部继电器 M4(上限报警输出) 或 M11(模拟量输入操作状态为 3 以上) 时，把 0 设置到 D772(模拟量输出数据)，关闭加热器电源。

当 M4 和 M11 关闭时，PID 指令的 D24(模拟量输出模块 S1+24 的输出操作变量) 将传送到 D772(模拟量输出数据)。

当 D761(模拟量输入操作状态) 存储 3 以上时，将打开内部继电器 M11。

使用梯形图刷新模拟量输入模块的注意事项：

- 当使用带 Pt100 或 Ni100 输入的模拟量输入模块 FC4A-J4CN1 时，使用 XYFS 和 CVXTY 指令把 0-6,000 输入转换为 0-50,000 输入，并把结果存储到 PID 指令的进程变量 (S4)。
- 当使用带 Pt1000 或 Ni1000 输入的模拟量输入模块 FC4A-J4CN1 时，使用 XYFS 和 CVXTY 指令把 0-60,000 输入转换为 0-50,000 输入，并把结果存储到 PID 指令的进程变量 (S4)。
- 当使用模拟量输入模块 FC4A-J8AT1 时，保持在热敏电阻显示线性特点的温度范围内操作。
- 当使用模拟量输入模块 FC4A-J8AT1 时，使用 XYFS 和 CVXTY 指令把 0-4,000 输入转换为 0-50,000 输入，并把结果存储到 PID 指令的进程变量 (S4)。
- 当使用带电压输出的模拟量输出模块 FC4A-K2C1 时，使用 XYFS 和 CVXTY 指令转换模拟量输出模块 (S1+24) 的输出操作变量，并把结果存储到指定为模拟量输出模块的模拟量输出数据的数据寄存器。
- 下面示例演示模拟量输入模块 FC4A-J4CN1 在 D410 中把 Pt1000 或 Ni1000 模拟量输入数据转换为 0 和 50,000 之间的一个值并把结果存储到 D510 的一个程序。



M8120 是初始化脉冲特殊内部继电器。

在启动时 XYFS 指定两个点。

当输入 I0 打开时，CVXTY 在 D410 中转换值并把结果存储到 D510。

15: 双 / 示教定时器指令

简介

双定时器指令从指定输出、内部继电器或移位寄存器位中生成所需时间段的开 / 关脉冲。提供了四个双定时器，可以从 1 ms - 65535s 范围内选择开 / 关时间段。

示教定时器指令测量该指令起始输入的打开时间段，并将测量数据存储在设计数据寄存器中，可用作定时器指令的预置值。

DTML (1 秒双定时器)



当输入打开时，目标设备 D1 在设备 S1 和 S2 指定的时间段分别打开和关闭。

时间范围为 0 - 65535s。

DTIM (100 毫秒双定时器)



当输入打开时，目标设备 D1 在设备 S1 和 S2 指定的时间段分别打开和关闭。

时间范围为 0 - 6553.5s。

DTMH (10 毫秒双定时器)



当输入打开时，目标设备 D1 在设备 S1 和 S2 指定的时间段分别打开和关闭。

时间范围为 0 - 655.35s。

DTMS (1 毫秒双定时器)



当输入打开时，目标设备 D1 在设备 S1 和 S2 指定的时间段分别打开和关闭。

时间范围为 0 - 65.535s。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

15: 双/示教定时器指令

有效设备

设备	功能	I	Q	M	R	T	C	D	常量
S1 (源 1)	打开时间	—	—	—	—	—	—	X	0-65535
S2 (源 2)	关闭时间	—	—	—	—	—	—	X	0-65535
D1 (目标 1)	双定时器输出	—	X	▲	X	—	—	—	—
D2 (目标 2)	系统工作区	—	—	—	—	—	—	X	—

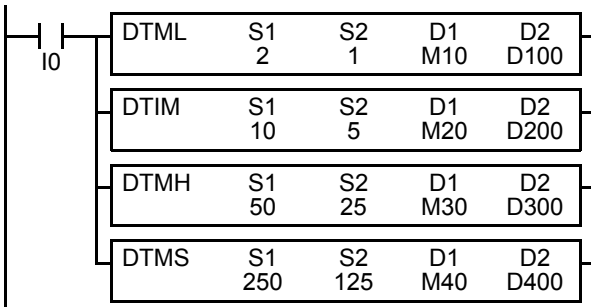
关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 可将内部继电器 M0 - M2557 指定为 D1。不能将特殊内部继电器指定为 D1。

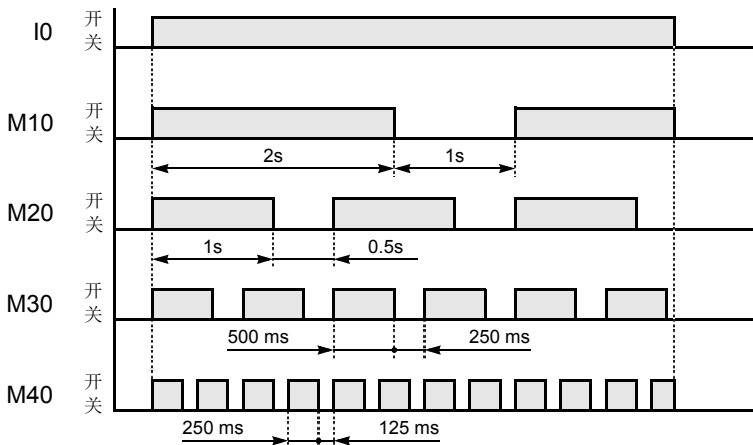
目标设备 D2 (系统工作区) 使用以分配给 D2 的设备起始的 2 个数据寄存器。可以将数据寄存器 D0 - D1998、D2000 - D7998 和 D10000 - D49998 指定为 D2。这两个数据寄存器用于系统工作区。请勿使用这些数据寄存器作为其他高级指令的目标，以及使用 WindLDR 上的点写入功能更改这些数据寄存器的值。如果这些数据寄存器中的数据被更改，则双定时器不能正确操作。

不能在中断程序中使用双定时器指令。否则，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

示例：DTML、DTIM、DTMH、DTMS



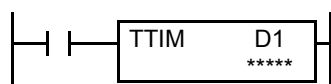
当输入 I0 打开时，四个双定时器指令会根据源设备 S1 和 S2 指定的打开和关闭时间段打开和关闭目标设备。



指令	递增	S1	打开时间	S2	关闭时间
DTML	1s	2	$1s \times 2 = 2s$	1	$1s \times 1 = 1s$
DTIM	100 ms	10	$100\text{ ms} \times 10 = 1s$	5	$100\text{ ms} \times 5 = 0.5s$
DTMH	10 ms	50	$10\text{ ms} \times 50 = 500\text{ ms}$	25	$10\text{ ms} \times 25 = 250\text{ ms}$
DTMS	1 ms	250	$1\text{ ms} \times 250 = 250\text{ ms}$	125	$1\text{ ms} \times 125 = 125\text{ ms}$

有关定时器指令的定时器精确度，请参阅第 7-9 页 (基本卷)。

TTIM（示教定时器）



当输入打开时，将以 100 ms 为单位测量打开时间，测量值将存储至目标设备 D1 指定的数据寄存器中。

测量时间范围为 0 - 6553.5s。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

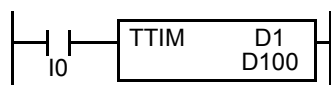
设备	功能	I	Q	M	R	T	C	D	常量
D1（目标 1）	测量值	—	—	—	—	—	—	X	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

目标设备 D1（测量值）使用以分配给 D1 的设备起始的 3 个数据寄存器。可以将数据寄存器 D0 - D1997、D2000 - D7997 和 D10000 - D49997 指定为 D1。以目标设备 D1+1 起始的两个后续数据寄存器用于系统工作区。请勿使用这两个数据寄存器作为其他高级指令的目标，以及使用 WindLDR 上的点写入功能更改这些数据寄存器的值。如果这些数据寄存器中的数据被更改，则示教定时器不能正确操作。

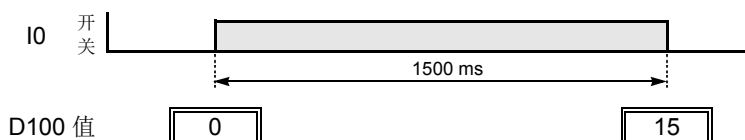
不能在中断程序中使用示教定时器指令。否则，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

示例：TTIM

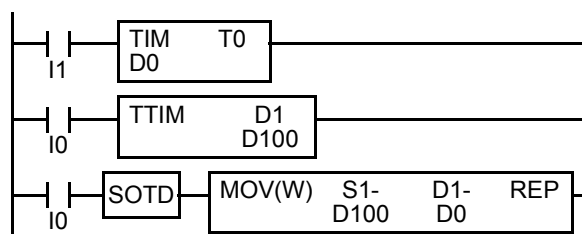


当输入 I0 打开时，TTIM 将数据寄存器 D100 复位为零，并开始将输入 I0 的打开时间存储至数据寄存器 D100 中（测量单位为 100 ms）。

当输入 I0 关闭时，TTIM 停止测量，且数据寄存器 D100 将维持打开时间段的测量值。



以下示例演示可测量输入 I0 的打开时段，并使用打开时间作为 100 毫秒定时器指令 TIM 预置值的程序。



当输入 I1 打开时，100 毫秒定时器 T0 开始使用存储在数据寄存器 D0 中的预置值工作。

当输入 I0 打开时，TTIM 测量输入 I0 的打开时间，并将测量值以 100 ms 为单位存储至数据寄存器 D100 中。

当输入 I0 关闭时，MOV(W) 存储 D100 值至数据寄存器 D0 中，作为定时器 T0 的预置值。

15: 双 / 示教定时器指令

16: 智能型模块访问指令

简介

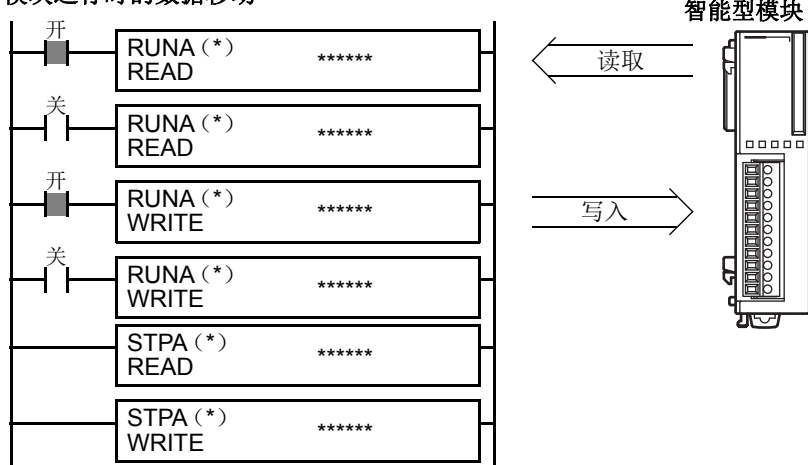
智能型模块访问指令用于在 CPU 模块运行或停止时，读取 CPU 模块和最多七个智能型模块之间的数据或写入数据至其中。

智能型模块访问列表

运行访问读取指令可读取智能型模块中指定地址的数据，并在 CPU 模块运行时将读取的数据存储至指定设备。运行访问读取指令可在 CPU 模块运行时，将指定设备中的数据写入智能型模块中的指定地址。

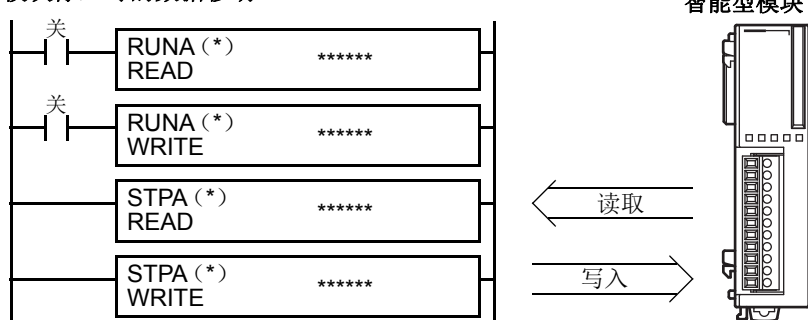
停止访问读取指令可读取智能型模块中指定地址的数据，并在 CPU 模块停止时将读取的数据存储至指定设备。停止访问读取指令可在 CPU 模块停止时，将指定设备中的数据写入智能型模块中的指定地址。

CPU 模块运行时的数据移动



当 CPU 模块正在运行，且输入打开时，可执行 RUNA READ 以读取智能型模块中的数据，或执行 RUNA WRITE 将数据写入智能型模块。

CPU 模块停止时的数据移动



当 CPU 模块停止时，可执行 STPA READ 以读取智能型模块中的数据，或执行 STPA WRITE 将数据写入智能型模块。

16: 智能型模块访问指令

RUNA READ (运行访问读取)



当输入打开时，将从以 SLOT 指定的智能型模块中的 ADDRESS 起始的区域读取数据，并将这些数据存储至 DATA 指定的设备中。

BYTE 指定要读取的数据量。

STATUS 存储运行状态代码。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D FC5A-C24R2D	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

有效设备 (运行访问读取)

设备	功能	I	Q	M	R	T	C	D	常量	重复
DATA	存储读取数据的起始设备编号	—	X	▲	X	X	X	X	—	—
STATUS	运行状态代码	—	—	—	—	—	—	X	—	—
SLOT	智能型模块插槽编号	—	—	—	—	—	—	—	1-7	—
ADDRESS	可从中读取数据的智能型模块起始地址	—	—	—	—	—	—	—	0-127	—
BYTE	要读取的数据大小	—	—	—	—	—	—	—	1-127	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

DATA : 指定起始设备编号以存储从智能型模块中读取的数据。

▲ 可将内部继电器 M0 - M2557 指定为 DATA。特殊内部继电器不能分配给 DATA。

当 T (定时器) 或 C (计数器) 用作 DATA 以运行访问读取时，可以将从智能型模块中读取的数据存储为预置值 (TP 或 CP) (0 - 65535)。

包括特殊数据寄存器、扩展数据寄存器的所有寄存器都可以指定为 DATA。

STATUS : 指定存储运行状态代码的数据寄存器。可以将数据寄存器 D0 - D1999 和 D10000 - D49999 指定为 STATUS。不能指定特殊数据寄存器和扩展数据寄存器。有关状态代码描述，请参阅第 16-6 页。

SLOT : 在安装智能模块的地方输入插槽编号。最多可以使用七个智能型模块。

ADDRESS : 指定要从中读取数据的智能型模块的起始地址。

BYTE : 以字节为单位指定要读取的数据量。

不能在中断程序中使用 RUNA READ 指令。否则，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

有效数据类型

W (字)	X	当分配位设备 [如 I (输入)、Q (输出)、M (内部继电器) 或 R (移位数据寄存器)] 给 DATA 时，将使用 16 点。
I (整数)	X	
D (双字)	—	当分配字设备 [如 T (定时器)、C (定时器) 或 D (数据寄存器)] 给 DATA 时，将使用 1 点。
L (长整数)	—	
F (浮点)	—	

RUNA WRITE (运行访问写入)



当输入打开时，以 DATA 指定的设备起始的区域中的数据将写入 SLOT 指定的智能型模块的 ADDRESS 中。

BYTE 指定要写入的数据量。

STATUS 存储运行状态代码。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D FC5A-C24R2D	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

有效设备 (运行访问写入)

设备	功能	I	Q	M	R	T	C	D	常量	重复
DATA	要从中提取数据的起始设备编号	X	X	X	X	X	X	X	X	X
STATUS	运行状态代码	—	—	—	—	—	—	X	—	—
SLOT	智能型模块插槽编号	—	—	—	—	—	—	—	1-7	—
ADDRESS	要写入数据的智能型模块起始地址	—	—	—	—	—	—	—	0-127	—
BYTE	要写入的数据大小	—	—	—	—	—	—	—	1-127	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

DATA : 指定起始设备编号以提取要写入智能型模块的数据。

当 T (定时器) 或 C (计数器) 用作 DATA 以运行访问写入时，定时器 / 计数器当前值将 (TC 或 CC) 写入智能型模块。

包括特殊数据寄存器、扩展数据寄存器的所有数据寄存器都可以指定为 DATA。

当分配常量给 DATA 时，不能选择重复。有关重复或不重复数据移动的详细信息，请参阅第 16-7 页。

STATUS : 指定存储运行状态代码的数据寄存器。可以将数据寄存器 D0 - D1999 和 D10000 - D49999 指定为 STATUS。不能指定特殊数据寄存器和扩展数据寄存器。有关状态代码描述，请参阅第 16-6 页。

SLOT : 在安装智能模块的地方输入插槽编号。最多可以使用七个智能型模块。

ADDRESS : 指定要存储数据的智能型模块的起始地址。

BYTE : 以字节为单位指定要写入的数据量。

不能在中断程序中使用 RUNA WRITE 指令。否则，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

有效数据类型

W (字)	X	当分配位设备 [如 I (输入)、Q (输出)、M (内部继电器) 或 R (移位数据寄存器)] 给 DATA 时，将使用 16 点。
I (整数)	X	
D (双字)	—	当分配字设备 [如 T (定时器)、C (定时器) 或 D (数据寄存器)] 给 DATA 时，将使用 1 点。
L (长整数)	—	
F (浮点)	—	

16: 智能型模块访问指令

STPA READ（停止访问读取）

STPA (*) READ	DATA *****	STATUS *****	SLOT *	ADDRESS ***	BYTE ***
------------------	---------------	-----------------	-----------	----------------	-------------

该指令不要使用起始输入。

当 CPU 模块停止时，将从以 SLOT 指定的智能型模块中的 ADDRESS 开始的区域读取数据，并将这些数据存储至 DATA 指定的设备中。

BYTE 指定要读取的数据量。

STATUS 存储运行状态代码。

注释：STPA READ 和 STPA WRITE 指令在用户程序中可使用 64 次。当在用户程序中使用多于 64 个 STPA READ 和 STPA WRITE 指令时，不会执行超出的指令，且错误代码 7 会存储至分配给 STATUS 的数据寄存器中。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D FC5A-C24R2D	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

有效设备（停止访问读取）

设备	功能	I	Q	M	R	T	C	D	常量	重复
DATA	存储读取数据的起始设备编号	—	X	▲	X	X	X	X	—	—
STATUS	运行状态代码	—	—	—	—	—	—	X	—	—
SLOT	智能型模块插槽编号	—	—	—	—	—	—	—	1-7	—
ADDRESS	可从中读取数据的智能型模块起始地址	—	—	—	—	—	—	—	0-127	—
BYTE	要读取的数据大小	—	—	—	—	—	—	—	1-127	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

DATA： 指定起始设备编号以存储从智能型模块中读取的数据。

▲ 可将内部继电器 M0 - M2557 指定为 DATA。特殊内部继电器不能分配给 DATA。

在 T（定时器）或 C（计数器）用作 DATA for Run Access Read（运行访问读取的数据）时，从智能型模块读取的数据将作为预置值（TP 或 CP）存储，此值介于 0 到 65535 之间。

包括特殊数据寄存器、扩展数据寄存器的所有数据寄存器都可以指定为 DATA。

STATUS： 指定存储运行状态代码的数据寄存器。可以将数据寄存器 D0 - D1999 和 D10000 - D49999 指定为 STATUS。不能指定特殊数据寄存器和扩展数据寄存器。有关状态代码描述，请参阅第 16-6 页。

SLOT： 在安装智能模块的地方输入插槽编号。最多可以使用七个智能型模块。

ADDRESS： 指定要从中读取数据的智能型模块的起始地址。

BYTE： 以字节为单位指定要读取的数据量。

不能在中断程序中使用 STPA READ 指令。否则，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

如果将 STPA READ 指令写入 MCS 和 MCR 指令之间，则当 CPU 模块停止时，无论 MCS 指令的输入条件是否满足，都将执行 STPA READ 指令。有关 MCS 和 MCR 指令，请参阅第 7-28 页（基本卷）。

有效数据类型

W（字）	X	当分配位设备 [如 I（输入）、Q（输出）、M（内部继电器）或 R（移位数据寄存器）] 给 DATA 时，将使用 16 点。
I（整数）	X	当分配字设备 [如 T（定时器）、C（计数器）或 D（数据寄存器）] 给 DATA 时，将使用 1 点。
D（双字）	—	
L（长整数）	—	
F（浮点）	—	

STPA WRITE (停止访问写入)

STPA(*)	DATA (R)	STATUS	SLOT	ADDRESS	BYTE
WRITE	*****	*****	*	***	***

该指令不要使用起始输入。

当 CPU 模块停止时, 以 DATA 指定的设备开始的区域中的数据将写入 SLOT 指定的智能型模块的 ADDRESS 中。

BYTE 指定要写入的数据量。

STATUS 存储运行状态代码。

注释: STPA READ 和 STPA WRITE 指令在用户程序中可使用 64 次。当在用户程序中使用多于 64 个 STPA READ 和 STPA WRITE 指令时, 不会执行超出的指令, 且错误代码 7 会存储至分配给 STATUS 的数据寄存器中。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D FC5A-C24R2D	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	X	X	X	X

有效设备 (运行访问写入)

设备	功能	I	Q	M	R	T	C	D	常量	重复
DATA	要从中提取数据的起始设备编号	X	X	X	X	X	X	X	X	X
STATUS	运行状态代码	—	—	—	—	—	—	X	—	—
SLOT	智能型模块插槽编号	—	—	—	—	—	—	—	1-7	—
ADDRESS	要写入数据的智能型模块起始地址	—	—	—	—	—	—	—	0-127	—
BYTE	要写入的数据大小	—	—	—	—	—	—	—	1-127	—

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

DATA: 指定起始设备编号以提取要写入智能型模块的数据。

在 T (定时器) 或 C (计数器) 用作 DATA for Run Access Write (运行访问写入的数据) 时, 定时器 / 计数器当前值 (TC 或 CC) 将写入智能型模块。

包括特殊数据寄存器、扩展数据寄存器的所有寄存器都可以指定为 DATA。

当分配常量给 DATA 时, 不能选择重复。有关重复或不重复数据移动的详细信息, 请参阅第 16-7 页。

STATUS: 指定存储运行状态代码的寄存器。可以将数据寄存器 D0 - D1999 和 D10000 - D49999 指定为 STATUS。不能指定特殊数据寄存器和扩展数据寄存器。有关状态代码描述, 请参阅第 16-6 页。

SLOT: 在安装智能模块的地方输入插槽编号。最多可以使用七个智能型模块。

ADDRESS: 指定要存储数据的智能型模块的起始地址。

BYTE: 以字节为单位指定要写入的数据量。

不能在中断程序中使用 STPA WRITE 指令。否则, 将导致用户程序执行错误, 这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

如果将 STPA WRITE 指令写入 MCS 和 MCR 指令之间, 则当 CPU 模块停止时, 无论 MCS 指令的输入条件是否满足, 都将执行 STPA WRITE 指令。有关 MCS 和 MCR 指令, 请参阅第 7-28 页 (基本卷)。

有效数据类型

W (字)	X	当分配位设备 [如 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器)] 给 DATA 时, 将使用 16 点。
I (整数)	X	
D (双字)	—	当分配字设备 [如 T (定时器)、C (定时器) 或 D (数据寄存器)] 给 DATA 时, 将使用 1 点。
L (长整数)	—	
F (浮点)	—	

16: 智能型模块访问指令

智能型模块访问状态代码

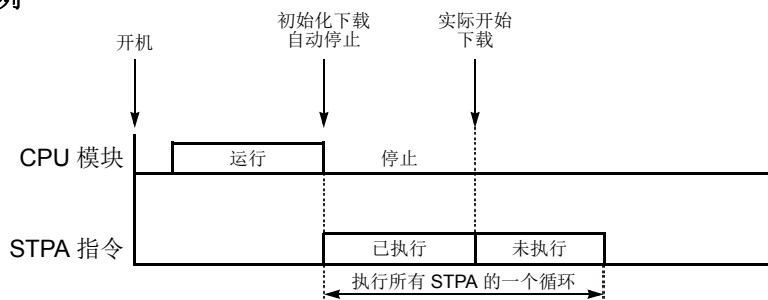
分配给 STATUS 的数据寄存器存储状态代码以指示运行状态和智能型模块访问运行的错误。当存储状态代码 1、3 或 7 时，请按下表所述进行修正测量：

状态代码	状态	说明	RUNA	STPA
0	正常	智能型模块访问正常。	X	X
1	总线错误	智能型模块安装错误。 断开 MicroSmart 模块的电源，并重新正确安装智能型模块。	X	X
3	无效模块编号	没找到指定的模块编号。 确认智能型模块编号并修正程序。	X	X
7	指令使用过多	使用的 STPA READ 和 STPA WRITE 指令超过 64 个。删除超出的指令。	—	X

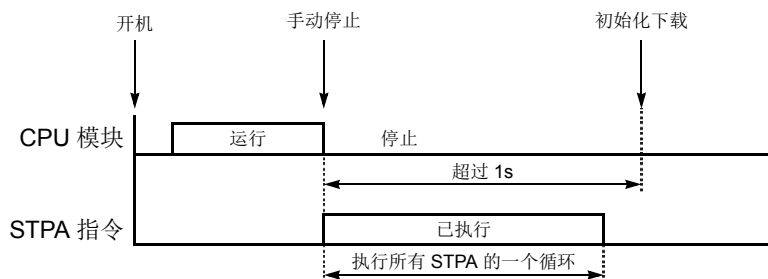
在下载程序时执行 STPA

在下载用户程序时，默认情况下，CPU 模块自动停止。根据下载初始化时间和执行所有 STPA READ 和 STPA WRITE 指令的总时间，可能无法执行某些 STPA 指令。如果出现这种情况，手动停止 CPU 模块。在 1s 后，如下表所示初始化用户程序下载。

自动停止序列



手动停止序列

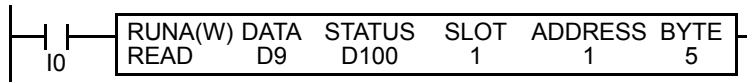


MCS 和 MCR 指令之间的 STPA 执行

当 CPU 模块停止时，不论 MCS 指令的输入打开还是关闭，都将执行 MCS 和 MCR 指令之间编写的 STPA 指令。有关 MCS 和 MCR 指令，请参阅第 7-28 页 (基本卷)。

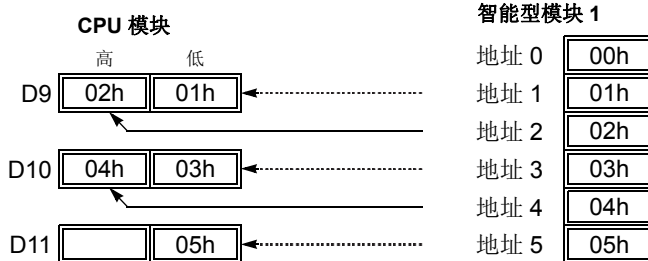
示例 :RUNA READ

以下示例描述 RUNA READ 指令的数据移动。STPA READ 的数据移动与 RUNA READ 指令的相同。



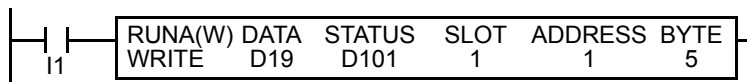
当输入 I0 打开时，从智能型模块 1 中以地址 1 开始的区域读取 5 字节数据，然后将该数据存储至以 D9 开始的数据寄存器中。

状态代码存储在数据寄存器 D100 中。



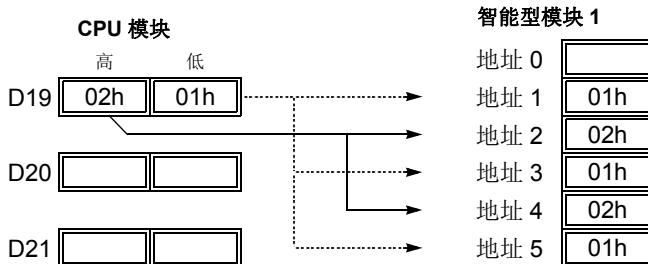
示例 :不重复的 RUNA WRITE

以下示例描述没有重复设置的 RUNA WRITE 指令的数据移动。STPA WRITE 的数据移动与 RUNA WRITE 指令的相同。



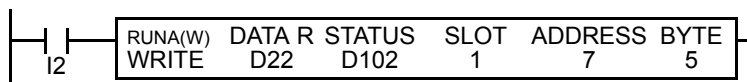
当输入 I1 打开时，数据寄存器 D19 中的数据将写入智能型模块 1 内以地址 1 开始的 5 字节区域内。

状态代码存储在数据寄存器 D101 中。



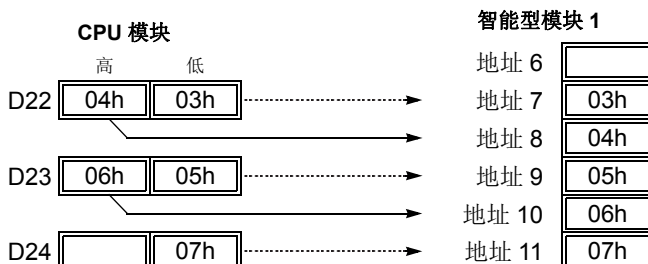
示例 :重复的 RUNA WRITE

以下示例描述有重复设置的 RUNA WRITE 指令的数据移动。STPA WRITE 的数据移动与 RUNA WRITE 指令的相同



当输入 I2 打开时，以数据寄存器 D22 开始的 5 字节区域中的数据将写入智能型模块 1 内以地址 7 开始的 5 字节区域内。

状态代码存储在数据寄存器 D102 中。



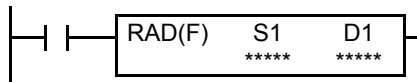
16: 智能型模块访问指令

17: 三角函数指令

简介

三角函数指令用于转换弧度和角度值，从弧度转换为正弦、余弦、正切，也用于计算反正弦、反余弦和反正切值。

RAD (角度换算为弧度)



$$S1 \cdot S1 + 1^\circ \times \pi / 180 \rightarrow D1 \cdot D1 + 1 \text{ rad}$$

当输入打开时，源设备 S1 所指定的角度值将转换为弧度值，并存储到设备 D1 所指定的目标中。

适用的 CPU 模块

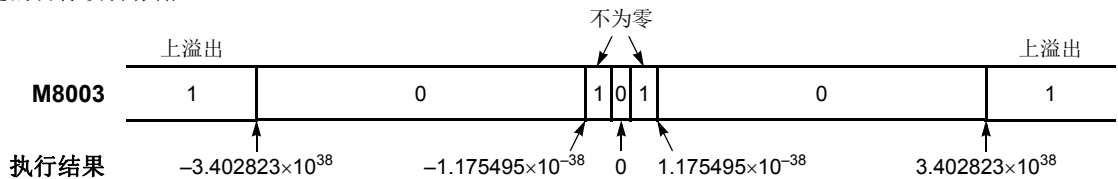
FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为弧度的角度值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当转换结果不在 -3.402823×10^{38} 与 $-1.175495 \times 10^{-38}$ 或 1.175495×10^{-38} 与 3.402823×10^{38} 之间的范围内时，除了转换结果是 0 外，将打开特殊内部继电器 M8003 (进位或借位)。当转换结果在 $-1.175495 \times 10^{-38}$ 与 1.175495×10^{-38} 之间时，D1 指定的目标设备存储 0。



当 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 RAD 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

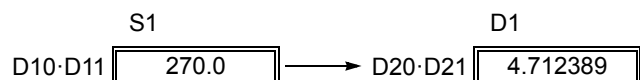
W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

示例 :RAD



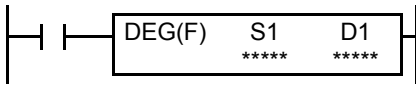
当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的角度值转换为弧度值，并且存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$$270^\circ \times \pi / 180 \rightarrow 4.712389 \text{ rad}$$



17: 三角函数指令

DEG (弧度换算为角度)



$$S1 \cdot S1+1 \text{ rad} \times 180 / \pi \rightarrow D1 \cdot D1+1^\circ$$

当输入打开时，源设备 S1 所指定的弧度值将转换为角度值，并存储到设备 D1 所指定的目标中。

适用的 CPU 模块

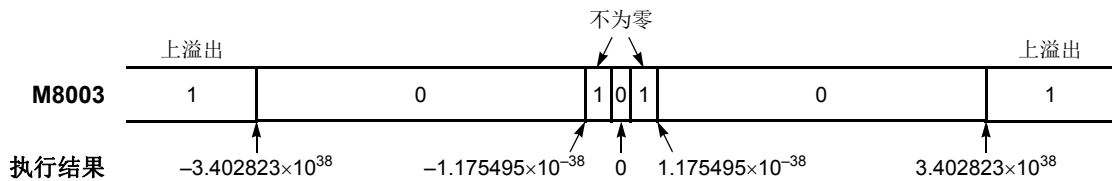
FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为角度的弧度值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当转换结果不在 -3.402823×10^{38} 与 $-1.175495 \times 10^{-38}$ 或 1.175495×10^{-38} 与 3.402823×10^{38} 的范围内时，除了转换结果是 0 外，将打开特殊内部继电器 M8003 (进位或借位)。当转换结果低于 -3.402823×10^{38} 或高于 3.402823×10^{38} 时，将导致溢出，D1 指定的目标设备存储负无穷大或正无穷大。



当 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 DEG 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

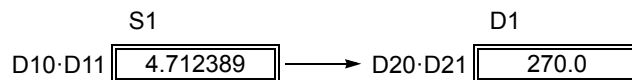
W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

示例 :DEG

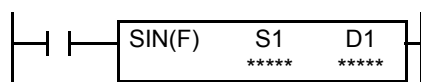


当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的弧度值转换为角度值，并且存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$$4.712389 \text{ rad} \times 180 / \pi \rightarrow 270.0^\circ$$



SIN (正弦)


 $\sin S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$

输入打开时，将源设备 S1 所指定的弧度值的正弦存储到设备 D1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为正弦值的弧度值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

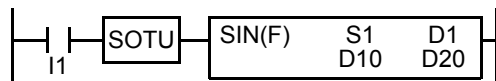
由于 SIN 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

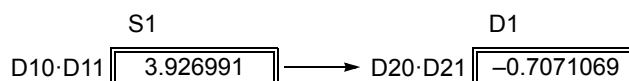
W (字)	—
I (整数)	—
D (双字)	—
L (长整数)	—
F (浮点)	X

由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。

示例 :SIN

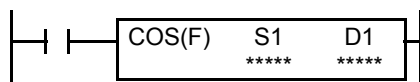


当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的弧度值的正弦存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

 $3.926991 \text{ rad} = 5\pi/4 \text{ rad}$
 $\sin 5\pi/4 \rightarrow -0.7071069$


17: 三角函数指令

COS (余弦)



$\cos S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$

输入打开时，将源设备 S1 所指定的弧度值的余弦存储到设备 D1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为余弦值的弧度值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 COS 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

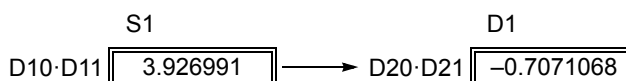
示例 :COS



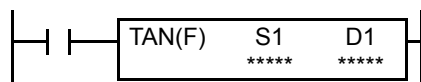
当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的弧度值的余弦存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$3.926991 \text{ rad} = 5\pi / 4 \text{ rad}$

$\cos 5\pi / 4 \rightarrow -0.7071068$



TAN (正切)


 $\tan S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$

输入打开时，将源设备 S1 所指定的弧度值的正切存储到设备 D1 指定的目标。

适用的 CPU 模块

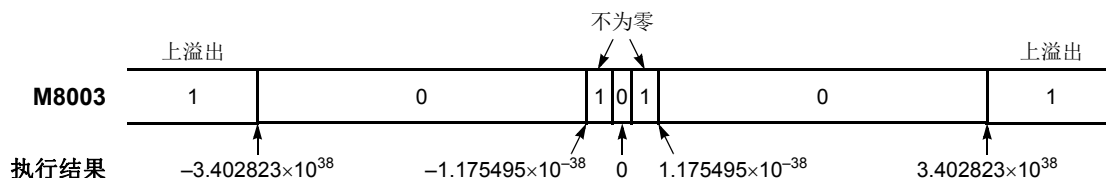
FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为正切值的弧度值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当转换结果不在 -3.402823×10^{38} 与 $-1.175495 \times 10^{-38}$ 或 1.175495×10^{-38} 与 3.402823×10^{38} 的范围内时，除了转换结果是 0 外，将打开特殊内部继电器 M8003 (进位或借位)。当转换结果低于 -3.402823×10^{38} 或高于 3.402823×10^{38} 时，将导致溢出，D1 指定的目标设备存储负无穷大或正无穷大。



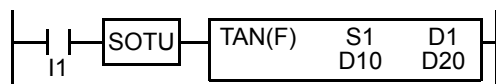
当 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 TAN 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

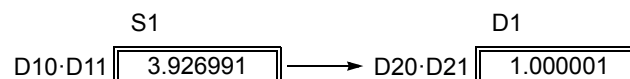
有效数据类型

W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

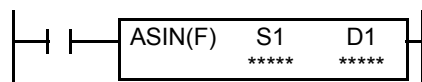
示例:TAN



当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的弧度值的正切存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

 $3.926991 \text{ rad} = 5 \pi / 4 \text{ rad}$
 $\tan 5 \pi / 4 \rightarrow 0.9999998$


ASIN (反正弦)



asin S1·S1+1 → D1·D1+1 rad

当输入打开时，将源设备 S1 所指定的值的反正弦以弧度存储到设备 D1 指定的目标。

S1·S1+1 值必须在下列范围内：

$$-1.0 \leq S1 \cdot S1+1 \leq 1.0$$

如果 S1·S1+1 值不在该范围内，将在 D1·D1+1 中存储无穷值。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为弧度的反正弦值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

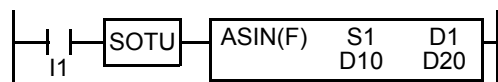
当源设备 S1 指定的数据不 -1.0 和 1.0 之间或不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 ASIN 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

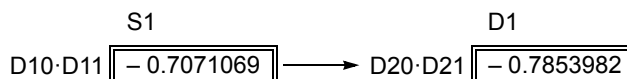
示例 :ASIN



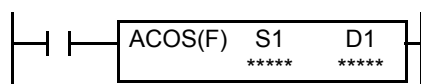
当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的值的反正弦存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$$\text{asin } -0.7071069 \rightarrow -0.7853982 \text{ rad}$$

$$-0.7853982 \text{ rad} = -\pi / 4 \text{ rad}$$



ACOS (反余弦)



$$\text{acos } S1 \cdot S1+1 \rightarrow D1 \cdot D1+1 \text{ rad}$$

当输入打开时，将源设备 S1 所指定的值的反余弦以弧度存储到设备 D1 指定的目标。

S1·S1+1 值必须在下列范围内：

$$-1.0 \leq S1 \cdot S1+1 \leq 1.0$$

如果 S1·S1+1 值不在该范围内，将在 D1·D1+1 中存储无穷值。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为弧度的反余弦值	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

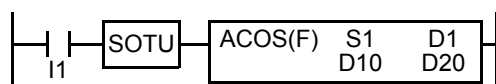
当源设备 S1 指定的数据不在 -1.0 和 1.0 之间或不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 ACOS 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

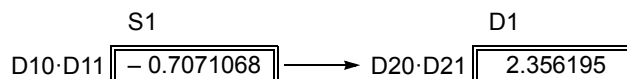
示例 :ACOS



当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的值的反余弦存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

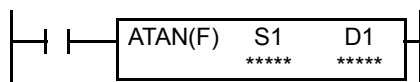
$$\text{acos } -0.7071068 \rightarrow 2.356195 \text{ rad}$$

$$2.356195 \text{ rad} = 3\pi/4 \text{ rad}$$



17: 三角函数指令

ATAN（反正切）



atan S1·S1+1 → D1·D1+1 rad

当输入打开时，将源设备 S1 所指定的值的反正切以弧度存储到设备 D1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1（源 1）	要转换为弧度的反正切值	—	—	—	—	—	—	X	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

当源设备 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 ATAN 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W（字）	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I（整数）	—	
D（双字）	—	
L（长整数）	—	
F（浮点）	X	

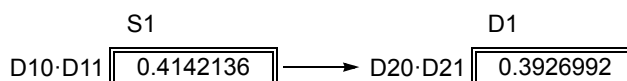
示例 :ATAN



当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的值的反正切存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

atan 0.4142136 → 0.3926992 rad

0.3926992 rad = $\pi/8$ rad

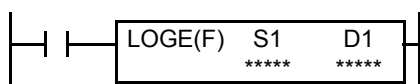


18: 对数 / 幂指令

简介

本章描述用来计算源设备的对数和幂值的对数和幂指令。

LOGE (自然对数)



$\log_e S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$

输入打开时，将源设备 S1 所指定的二进制数据的自然对数存储到设备 D1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	要转换为自然对数的二进制数据	—	—	—	—	—	—	X	X	—
D1 (目标 1)	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当源设备 S1 指定的数据小于或等于 0，或者不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 LOGE 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

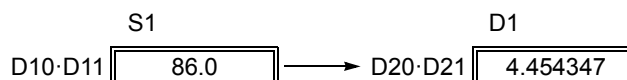
W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

示例: LOGE



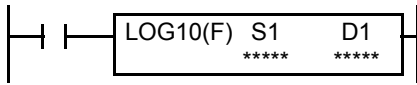
当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的二进制数据的自然对数存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$\log_e 86 \rightarrow 4.454347$



18: 对数/幂指令

LOG10（常用对数）



$$\log_{10} S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$$

输入打开时，将源设备 S1 所指定的二进制数据的常用对数存储到设备 D1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1（源 1）	要转换为常用对数的二进制数据	—	—	—	—	—	—	X	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

当源设备 S1 指定的数据小于或等于 0，或者不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

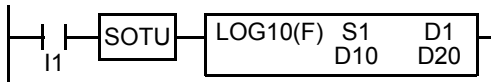
由于 LOG10 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W（字）	—
I（整数）	—
D（双字）	—
L（长整数）	—
F（浮点）	X

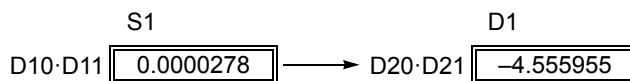
由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。

示例：LOG10

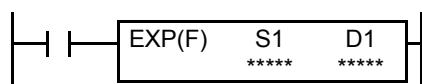


当输入 I1 打开时，将源设备 S1 指定的数据寄存器 D10 和 D11 的二进制数据的常用对数存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$$\log_{10} 0.0000278 \rightarrow -4.555955$$



EXP (指数)



$$e^{S1 \cdot S1+1} \rightarrow D1 \cdot D1+1$$

输入打开时，将计算源设备 S1 所指定的 e 的 S1·S1+1 次幂并存储到设备 D1 指定的目标。

e (自然对数的底数) = 2.7182818

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	指数的二进制数据	—	—	—	—	—	—	X	X	—
D1 (目标 1)	用于存储结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当运算结果不在 $-3.402823 \times 10^{38} \times 10^{38}$ 和 $-1.175495 \times 10^{-38}$ 或 1.175495×10^{-38} 和 3.402823×10^{38} 的范围内时，除了结果为 0 外，特殊内部继电器 M8003 (进位/借位) 将打开。

当运算结果在 $-1.175495 \times 10^{-38}$ 和 1.175495×10^{-38} 之间时，D1 指定的目标设备存储 0。当运算结果小于 -3.402823×10^{38} 或大于 3.402823×10^{38} 时，将导致溢出，D1 指定的目标设备存储负无穷大或正无穷大。

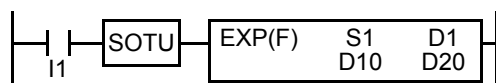
当源设备 S1 指定的数据不符合标准浮点格式时，将导致程序执行错误，此时将打开 CPU 模块上的特殊继电器 M8004 和 ERR LED。

由于 EXP 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

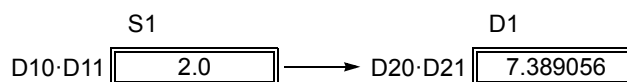
W (字)	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I (整数)	—	
D (双字)	—	
L (长整数)	—	
F (浮点)	X	

示例: EXP



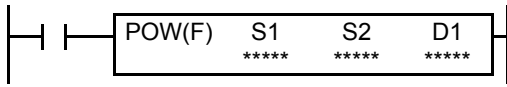
当输入 I1 打开时，将计算 e 的源设备 S1 指定的数据寄存器 D10 和 D11 的数据次幂，并且将运算结果存储到目标设备 D1 指定的数据寄存器 D20 和 D21。

$$e^2 = 2.7182818^2 \rightarrow 7.389056$$



18: 对数/幂指令

POW (幂)



$$S1 \cdot S1+1^{S2 \cdot S2+1} \rightarrow D1 \cdot D1+1$$

当输入打开时，将计算源设备 S1 指定的二进制数据由源设备 S2 指定的 S2·S2+1 次幂，并把运算结果存储到设备 D1 指定的目标。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	底数的二进制数据	—	—	—	—	—	—	X	X	—
S2 (源 2)	指数的二进制数据	—	—	—	—	—	—	X	X	—
D1 (目标 1)	用于存储结果的目标	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当运算结果不在 -3.402823×10^{38} 和 $-1.175495 \times 10^{-38}$ 或 1.175495×10^{-38} 至 3.402823×10^{38} 的范围内时，除了结果为 0 外，特殊内部继电器 M8003 (进位/借位) 将打开。

当运算结果在 $-1.175495 \times 10^{-38}$ 和 1.175495×10^{-38} 之间时，D1 指定的目标设备存储 0。当运算结果小于 -3.402823×10^{38} 或大于 3.402823×10^{38} 时，将导致溢出，D1 指定的目标设备存储负无穷大或正无穷大。

当发生下列情况时，将导致用户程序执行错误，这将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

- 源设备 S1 定的数据小于 0 并且源设备 S2 指定的数据不是整数。
- 源设备 S1 定的数据为 0 并且源设备 S2 指定的数据小于或等于 0。
- 源设备 S1 或 S2 指定的数据不符合标准浮点格式。

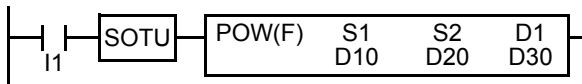
由于 POW 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	—
I (整数)	—
D (双字)	—
L (长整数)	—
F (浮点)	X

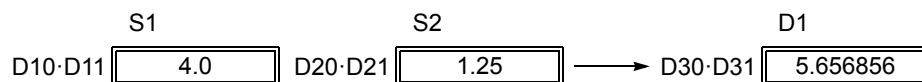
由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。

示例：POW



当输入 I1 打开时，将计算源设备 S1 指定的数据寄存器 D10 和 D11 的数据由源设备 S2 指定的 D20·D20+1 次幂，并且将运算结果存储到目标设备 D1 指定的数据寄存器 D30 和 D31。

$$4^{1.25} \rightarrow 5.656856$$



19: 文件数据处理指令

简介

文件数据处理指令实现先进先出 (FIFO) 数据结构。FIFO (先进先出格式) 指令将初始化 FIFO 数据文件存储的数据。FIEX (执行先进) 指令存储新数据到 FIFO 数据文件, FOEX (执行先出) 指令从 FIFO 数据文件取得存储的数据。被 FIEX 指令存储到 FIFO 数据文件的第一个数据为被 FOEX 指令检索的第一个数据。

NDSRC (N 数据查找) 指令在指定范围内查找指定值。

FIFO (先进先出格式)



输入打开时, FIFO 指令初始化 FIFO 数据文件。每个数据文件拥有各自的编号 0 ~ 9。一个用户程序中最多可以使用 10 个数据文件。

本指令适用于升级后的系统程序版本 200 或更高的 CPU 模块。

适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
N (文件编号)	文件编号	-	-	-	-	-	-	-	0 ~ 9	-
S1 (源 1)	数据寄存器每个记录的数量	-	-	-	-	-	-	-	1 ~ 255	-
S2 (源 2)	记录数量	-	-	-	-	-	-	-	2 ~ 255	-
D1 (目标 1)	要存储 FIFO 数据文件的起始数据寄存器	-	-	-	-	-	-	X	-	-
D2 (目标 2)	FIFO 状态输出	-	-	▲	-	-	-	-	-	-

关于有效设备编号范围, 请参阅第 6-1 页和第 6-2 页 (基本卷)。

▲ 不能将特殊内部继电器指定为 D2。

由于 FIFO 是在输入打开时在每个扫描中执行的, 因此, 应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

有效数据类型

W (字)	X
I (整数)	-
D (双字)	-
L (长整数)	-
F (浮点)	-

当 M (内部继电器) 指定为 D2 时, 将使用从以 D2 指定的设备开始的三个内部继电器。

当 D (数据寄存器) 指定为 D1 时, 将使用从以 D1 指定的设备开始的 S1xS2+2 数据寄存器。

19: 文件数据处理指令

目标设备 D1 (FIFO 数据文件)

当执行对应的 FIFO 指令时，将初始化 FIFO 数据文件。FIFO 数据文件被放置在以 D1 指定的设备开始的区域，并且占用与 $S1 \times S2 + 2$ 数据寄存器一样多的区域。每个记录的大小等于 S1 个数据。使用 FIEX 指令可以将 S2 个记录存储到一个 FIFO 数据文件中。存储的数据可以通过使用 FOEX 指令从 FIFO 数据文件中取出。

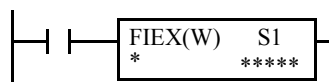
设备	功能	说明
D1+0	FI 指针	FI 指针显示存储新数据到 FIFO 数据文件的位置。当执行 FIEX 指令时，FIEX 指令的、从以 S1 指定的设备开始的数据寄存器中的新数据，存储在以 FI 指针指定的位置。 当 FI 指针显示 FIFO 数据文件的最后记录，并执行 FIEX 指令时，FI 将返回到 0。
D1+1	FO 指针	FO 指针显示并从 FIFO 数据文件中取出已存储的数据的位置。当执行 FOEX 指令时，取出以 FIFO 指针指定的位置的数据，并存储到 FOEX 指令的以 D1 指定的设备开始的数据寄存器中，FO 指针递增 1 以显示取出下一个数据的位置。 当 FO 指针指定 FIFO 数据文件的最后记录，并执行 FOEX 指令时，FO 将返回到 0。
D1+2	记录 0	要存储数据的第一个记录。
...		
D1+(S1+1)		
D1+(S1+2)	记录 1	要存储数据的第二个记录。
...		
D1+(S1x2+1)		
~~~~~		
D1+(S1x(S2-1)+2)	记录 S2-1	要存储数据的最后一个记录。
...		
D1+(S1xS2+1)		

### 目标设备 D2 (FIFO 状态输出)

当执行 FIEX 或 FOEX 指令时，按照执行状态打开 / 关闭以下的内部继电器。

设备	功能	说明
D2+0	数据文件满输出	当存储在 FI 指针 (D1+0) 的值等于 FO 指针 (D1+1) - 1 的值时，FIFO 数据文件已满，不能再存储数据。当 FIFO 数据文件已满时，如果执行 FIEX 指令，将不执行任何操作，并打开数据文件已满输出 (D2+0)。
D2+1	数据文件空输出	当存储在 FI 指针 (D1+0) 的值等于 FO 指针 (D1+1) 的值时，FIFO 数据文件为空。当 FIFO 数据文件为空时，如果执行 FOEX 指令，将不执行任何操作，并打开数据文件为空输出 (D2+1)。
D2+2	指针超出值域输出	存储在 FI 或 FO 指针的值可以是 0 ~ S2-1。在 FI 或 FO 指针的值超出值域期间，当执行 FIEX 或 FOEX 指令时，将不执行任何操作，并打开指针超出值域输出 (D2+2)。

## FIEX ( 执行先进 )



当打开输入时，存储在以 S1 指定的数据寄存器开始的设备中的数据将存储到相应的 FIFO 数据文件中。

本指令适用于升级后的系统程序版本 200 或更高的 CPU 模块。

### 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

### 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
N (文件编号)	文件编号	-	-	-	-	-	-	-	0 ~ 9	-
S1 (源 1)	要存储到 FIFO 数据文件的起始数据存储器	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

请确保相应的 FIFO 指令初始化 FIFO 数据文件后执行 FIEX 指令。如果没有执行相应的 FIFO 指令就执行 FIEX 指令，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 FIEX 指令是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

### 有效数据类型

W (字)	X
I (整数)	-
D (双字)	-
L (长整数)	-
F (浮点)	-

当 D (数据寄存器) 指定为源时，将使用与存储在相应的 FIFO 指令的设备中的值一样多的数据寄存器。

## FOEX ( 执行先出 )



当打开输入时，将从相应的 FIFO 数据文件中取得数据，并存储到以 D1 指定的从数据寄存器开始的设备。

本指令适用于升级后的系统程序版本 200 或更高的 CPU 模块。

### 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

### 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
N (文件编号)	文件编号	-	-	-	-	-	-	-	0 ~ 9	-
D1 (目标 1)	要存储数据的起始数据寄存器编号	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

请确保相应的 FIFO 指令初始化 FIFO 数据文件后执行 FOEX 指令。如果没有执行相应的 FIFO 指令就执行 FOEX 指令，将导致用户程序执行错误，此时将打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 FOEX 是在输入打开时在每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

## 19: 文件数据处理指令

### 有效数据类型

W (字)	X
I (整数)	-
D (双字)	-
L (长整数)	-
F (浮点)	-

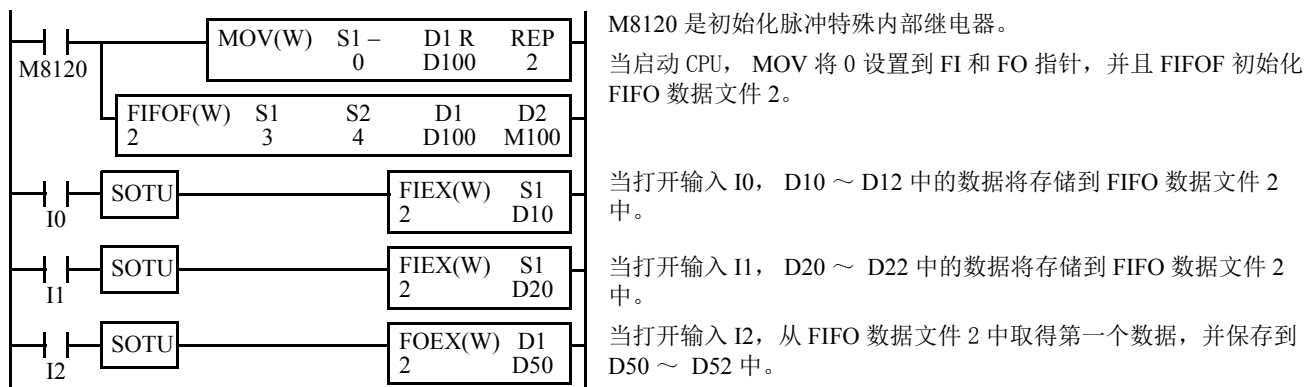
当 D (数据寄存器) 指定为目标时, 将使用与存储在相应的 FIFO 指令的设备 S1 中的值一样多的数据寄存器。

### 示例: FIFO, FIEX, FOEX

本程序描述了 FIFO, FIEX, FOEX 指令的用户程序使用 FIFO 数据文件。

文件编号:	2
数据寄存器每个记录的数量:	3
记录的数量:	4
FIFO 数据文件:	D100 ~ D113 (3x4+2 数据寄存器)
FIFO 状态输出:	M100 ~ M102

### 梯形图



### FIFO 数据文件

当按序打开输入 I0, I1 和 I2, 如下表所示数据将存储在 FIFO 数据文件中。下标仅显示由 FIFO, FIEX 和 FOEX 指令管理的有效数据。

功能	设备地址	输入 I0	输入 I1	输入 I2
FI 指针	D100	1	2	2
FO 指针	D101	0	0	1
记录 0	D102 ~ D104	D10, D11, D12	D10, D11, D12	-
记录 1	D105 ~ D107	-	D20, D21, D22	D20, D21, D22
记录 2	D108 ~ D110	-	-	-
记录 3	D111 ~ D113	-	-	-

## NDSRC (N 数据查找)



当输入打开，被设备 S1 指定的值将被查找。由设备 S2 指定的数据寄存器开始的数据寄存器进行查找。设备 S3 指定数据寄存器的 1 字或 2 字块的数量来查找，此数量取决于数据类型。

发生的第一个匹配的数据寄存器的偏离值存储在由设备 D1 指定的数据寄存器中。与值相匹配的次数将存储在下一个数据寄存器。当查找结果不匹配时，65535 被存储在设备 D1 并将 0 存储到设备 D1+1。

本指令适用于升级后的系统程序版本 210 或更高的 CPU 模块。

### 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

### 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	用于被查找的值	-	-	-	-	-	-	X	X	-
S2 (源 2)	用于查找的起始数据寄存器编号	-	-	-	-	-	-	X	-	-
S3 (源 3)	用于查找的块的数量	-	-	-	-	-	-	X	X	-
D1 (目标 1)	查找结果	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

源 S1 数据所指定的用于查找的值取决于数据类型的有效范围。

查找范围不能跨越数据寄存器、扩展数据寄存器和特殊数据寄存器。请确保由 S1 和 S2 指定的数据寄存器编号的合计不生成不同数据寄存器值域。

源 S3 和目标 D1 无视数据类型总是使用一个字。

目标 D1 占用由 D1 指定的设备开始的两个连续的数据寄存器。数据寄存器 D0 ~ D1998、D2000 ~ D7998 和 D10000 ~ 49998 可以被指定为目标 D1。

当选择 F (浮点) 数据类型并且 S1 或 S2 不是正确的浮点格式时，将导致用户程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

当 S3 为无效编号或 S2 和 S3 的合计不在数据寄存器的值域内，将导致用户程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

由于 NDSRC 是在输入打开时在每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

### 有效数据类型

W (字)	X	如果 D (数据寄存器) 等字设备指定为源，则使用 1 点 (字或整数数据类型) 或 2 点 (双字、长数整或浮点数据类型)。
I (整数)	X	
D (双字)	X	
L (长整数)	X	
F (浮点)	X	

### 源设备和目标设备的数量

根据数据类型，源设备 S1 和 S2 使用不同的设备数量。源设备 S3 和目标设备 D1 无视数据类型总是使用 1 个字。

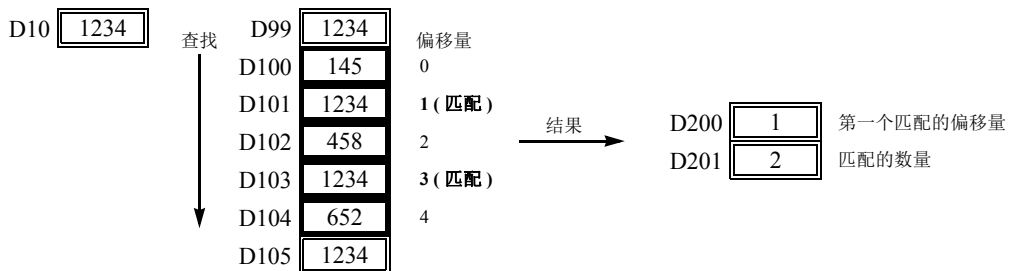
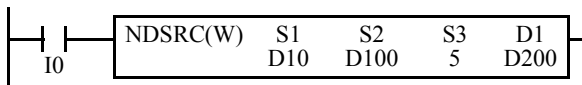
设备	W (字)、I (整数)	D (双字)、L (长整数)、F (浮点)
S1、S2	1 个字设备	2 个字设备
S3、D1	1 个字设备	1 个字设备

## 19: 文件数据处理指令

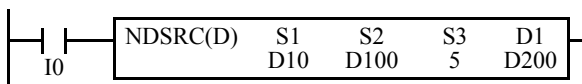
### 示例：NDSRC

以下示例描述了如何使用 NDSRC 指令查找三种数据类型的数据。

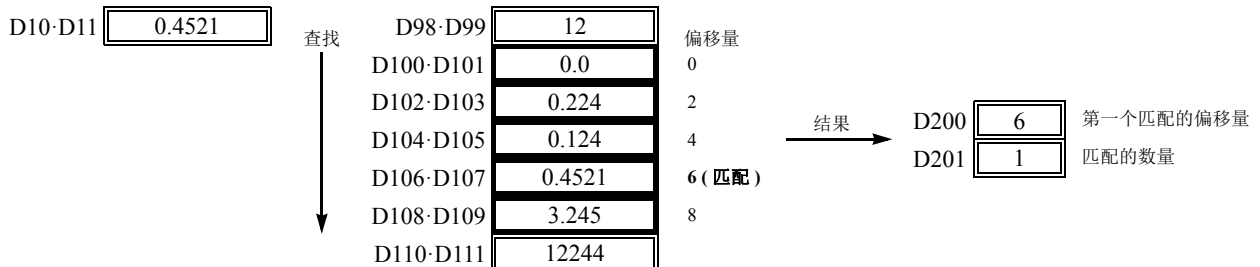
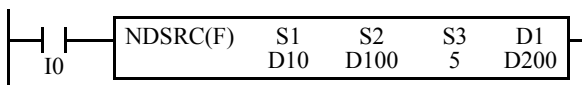
#### • 数据类型：字



#### • 数据类型：双字



#### • 数据类型：浮点



# 20: 时钟指令

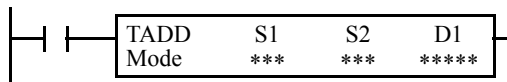
## 简介

TADD（时间加法）和 TSUB（时间减法）指令分别执行两个时间数据的加法或减法。该数据可以从时间（小时、分和秒）或日期/时间（年、月、日、星期、小时、分和秒）中选择。

HTOS（HMS → 秒）和 STO H（秒 → HMS）指令执行时间数据在小时、分、秒与秒之间的转换。

HOUR（小时计量器）指令测量输入的打开期间并将合计期间与预置值比较。当到达预置值将打开一个输出或内部继电器。

## TADD( 时间加法 )



$S1 + S2 \rightarrow D1, CY$

当输入打开时，根据所选模式，由源设备 S2 指定的时间数据被加到由源设备 S1 指定的日期/时间数据中。结果被保存到指定设备 D1 并进位 (M8003)。

该指令适用于升级后的系统程序版本 210 或更高的 CPU 模块。

## 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

## 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
模式	S1 数据范围的选择	-	-	-	-	-	-	-	0, 1	-
S1 (源 1)	要加到的日期/时间数据	-	-	-	-	-	-	X	-	-
S2 (源 2)	要加的时间数据	-	-	-	-	-	-	X	-	-
D1 (目标 1)	要存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页（基本卷）。

当选择模式 0，源设备 S1、S2 和目标设备 D1 占用由设备指定开始的 3 个连续数据寄存器。数据寄存器 D0 ~ D1997、D2000 ~ D7997 和 D10000 ~ D49997 可以指定为那些设备。

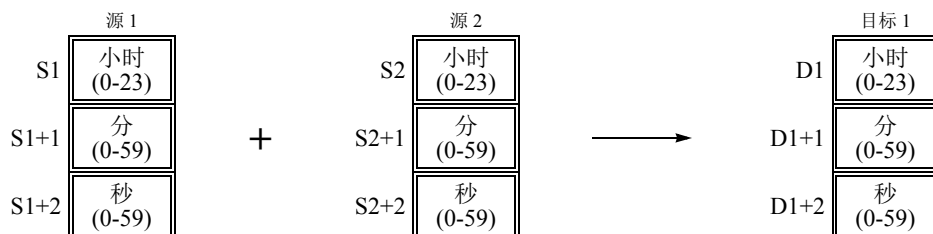
当选择模式 1，源设备 S1 和目标设备 D1 占用由设备指定开始的 7 个连续数据寄存器。数据寄存器 D0 ~ D1993、D2000 ~ D7993 和 D10000 ~ D49993 可以指定为那些设备。源设备 S2 占用由设备指定开始的 3 个连续数据寄存器。数据寄存器 D0 ~ D1997、D2000 ~ D7997 和 D10000 ~ D49997 可以指定为源设备 S2。

由于 TADD 指令是在输入打开时的每个扫描中执行的，因此，应当根据需要来自 SOTU 或 SOTD 指令的脉冲输入。

## 20: 时钟指令

### 模拟 0

当选择模式 0, 存储在从源设备 S2 开始的 3 个连续的数据寄存器中的时间数据 (小时、分和秒) 被加到存储在从源设备 S1 开始的 3 个连续的数据寄存器中的时间数据 (小时、分和秒)。该结果被存储到从目标设备 D1 开始的 3 个连续的数据存储器。



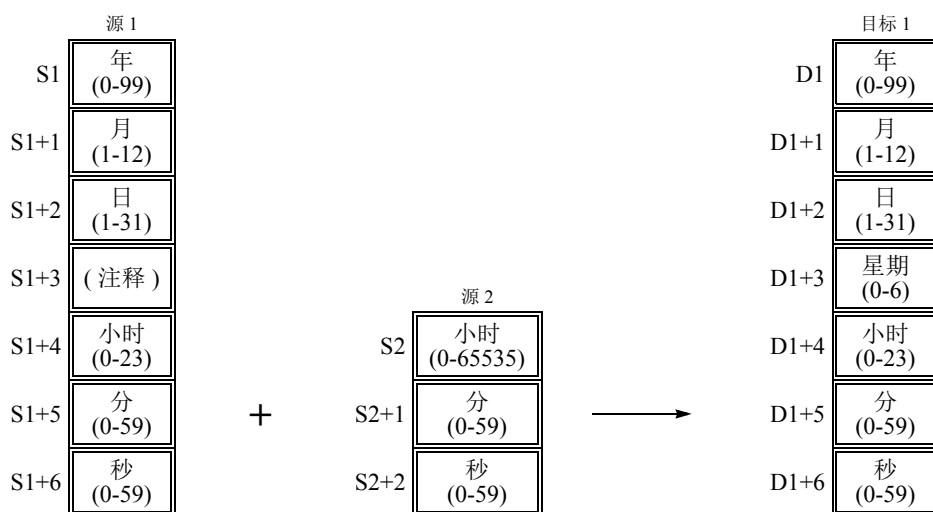
小时数据可以为 0 ~ 23。分和秒数据可以为 0 ~ 59。

当执行结果超出 23:59:59, 其结果将被减去 24 小时后存储到由目标设备 D1 指定的数据寄存器, 并且打开特殊内部继电器 M8003 (进位)。

如果小时、分和秒数据超出有效范围, 将导致程序执行错误, 并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

### 模式 1

当选择模式 1, 存储在从源设备 S2 开始的 3 个连续的数据寄存器中的时间数据 (小时、分和秒) 被加到存储在从源设备 S1 开始的 7 个连续的数据寄存器中的时间数据 (年、月、日、星期、小时、分和秒)。该结果被存储到从目标设备 D1 开始的 7 个连续的数据存储器。



**注释:** 源 1 中的设备 S1+3 不执行使用并且无需被指定。

源 1 数据与闰年兼容。

关于源 1: 年数据可以为 0 ~ 99。月数据可以为 1 ~ 12。日数据可以为 1 ~ 31。小时数据可以为 0 ~ 23。分和秒数据可以为 0 ~ 59。

年数据 0 ~ 99 将被处理为年 2000 ~ 2099。当设备 S1 包含 3 位或更多位数时, 最低的 2 位数将被用作年数据。

有关源 2: 小时数据可以为 0 ~ 65535。分和秒数据可以为 0 ~ 59。

目标 1: 自动从年、月和日的合计计算星期, 其结果将保存到设备 D1+3。

星期数据指的是: 0 (星期日), 1 (星期一), 2 (星期二), 3 (星期三), 4 (星期四), 5 (星期五) 和 6 (星期六)。

如果源 1 包含无效的日 / 时间数据, 将导致程序执行错误, 并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

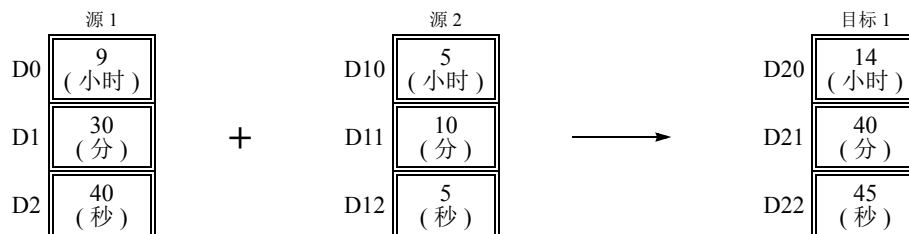
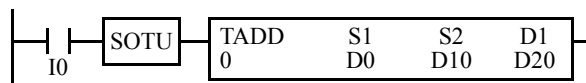
如果执行结果超出 99 年 12 月 31 日 23:59:59, 将导致程序执行错误, 并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。



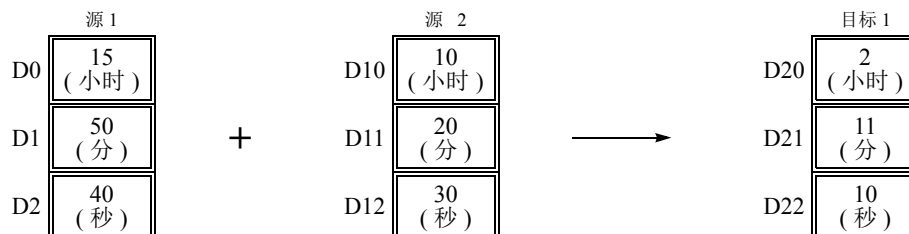
### 示例：TADD

该示例演示了在两个不同模式中如何使用 TADD 指令来加时间数据。

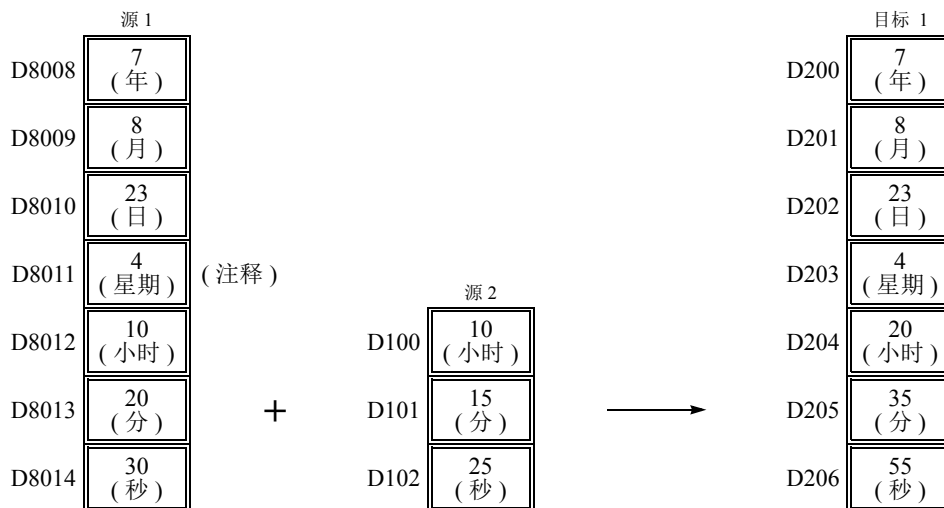
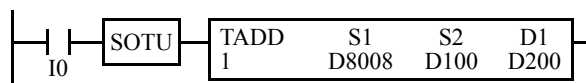
#### • 模式 0



当结果超出 23:59:59，生成的小时数据将被减去 24，并打开特殊内部继电器 M8003（进位）。



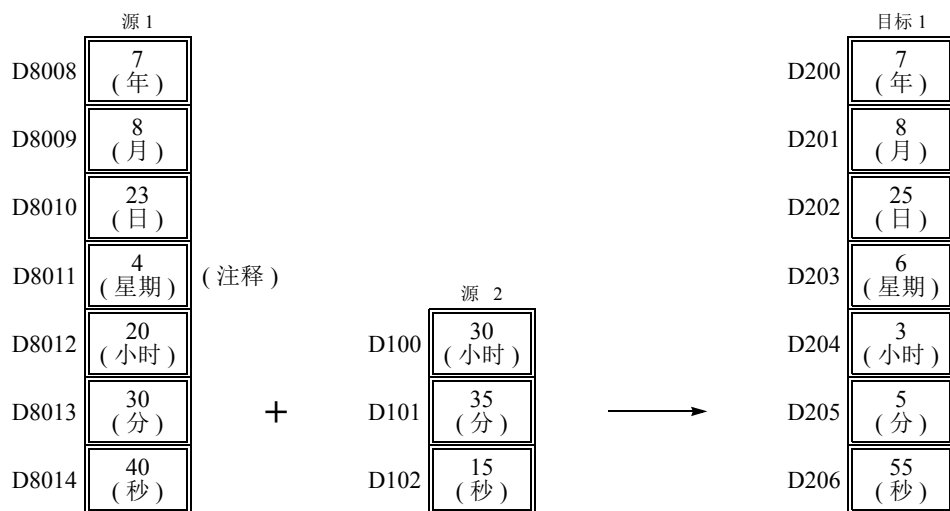
#### • 模式 1



**注释：**源 1 中的 D8011 没有被指定使用，并且无需指定。星期将自动从年、月和日的合计中被计算，并存储在目标 1 的 D203。

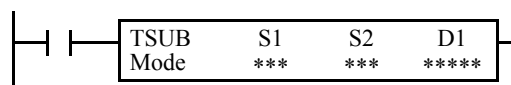
## 20: 时钟指令

当结果超出 23:59:59，合计的小时数据将被减去 24 的倍数，并递增日数据。



**注释：**源 1 中的 D8011 没有被指定使用，并且无需指定。星期将自动从年、月和日的合计中被计算，并存储在目标 1 的 D203。

## TSUB( 时间减法 )



$S1 - S2 \rightarrow D1, CY$

当输入打开时，根据所选模式，从由源设备 S1 指定的日期 / 时间数据中减去由源设备 S2 指定的时间数据。结果被保存到指定设备 D1 并借位 (M8003)。

该指令适用于升级后的 CPU 模块的系统程序版本 210 或更高版本。

### 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

### 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
模式	S1 数据范围的选择	-	-	-	-	-	-	-	0, 1	-
S1 (源 1)	需要从中减去的日期 / 时间数据	-	-	-	-	-	-	X	-	-
S2 (源 2)	要减去的时间数据	-	-	-	-	-	-	X	-	-
D1 (目标 1)	要存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

当选择模式 0，源设备 S1、S2 和目标设备 D1 占用由设备指定开始的 3 个连续的数据寄存器。数据寄存器 D0 ~ D1997、D2000 ~ D7997 和 D10000 ~ D49997 可以指定为那些设备。

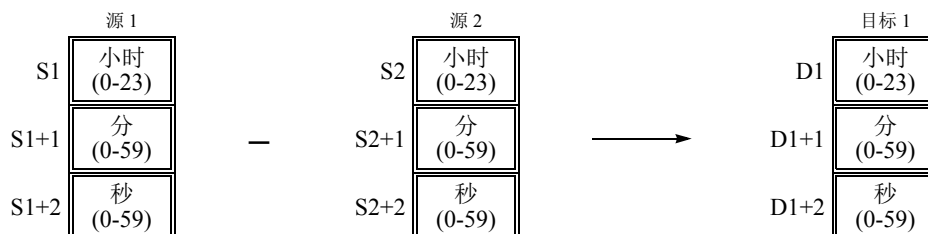
当选择模式 1，源设备 S1 和目标设备 D1 占用由设备指定开始的 7 个连续数据寄存器。数据寄存器 D0 ~ D1993、D2000 ~ D7993 和 D10000 ~ D49993 可以指定为那些设备。源设备 S2 占用由设备指定开始的 3 个连续数据寄存器。数据寄存器 D0 ~ D1997、D2000 ~ D7997 和 D10000 ~ D49997 可以指定为源设备 S2。

由于 TSUB 指令是在输入打开时的每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

## 20: 时钟指令

### 模式 0

当选择模式 0，从存储在从源设备 S1 开始的 3 个连续的数据寄存器中的时间数据 (小时、分和秒) 减去存储在从源设备 S2 开始的 3 个连续的数据寄存器中的时间数据 (小时、分和秒)。该结果被存储到从目标设备 D1 开始的 3 个连续的数据寄存器。



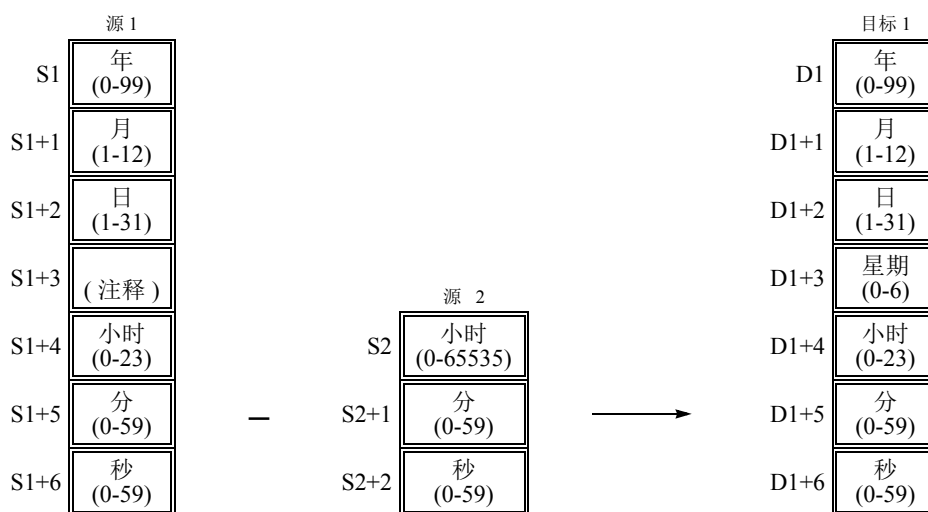
小时数据可以为 0 ~ 23。分和秒数据可以为 0 ~ 59。

当执行 TADD 指令后的结果小于 00:00:00，其结果将被加上 24 小时后存储到由目标设备 D1 指定的数据寄存器，并且打开特殊内部继电器 M8003 (借位)。

如果小时、分和秒数据超出有效范围，将导致程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

### 模式 1

当选择模式 1，从存储在从源设备 S1 开始的 7 个连续的数据寄存器中的时间数据 (年、月、日、星期、小时、分和秒) 减去存储在从源设备 S2 开始的 3 个连续的数据寄存器中的时间数据 (小时、分和秒)。该结果被存储到从目标设备 D1 开始的 7 个连续的数据寄存器。



**注释：** 源 1 中的设备 S1+3 没有被指定使用，并且无需被指定。

源 1 数据与闰年兼容。

关于源 1：年数据可以为 0 ~ 99。月数据可以为 1 ~ 12。日数据可以为 1 ~ 31。小时数据可以为 0 ~ 23。分和秒数据可以为 0 ~ 59。

年数据 0 ~ 99 将被处理为年 2000 ~ 2099。当设备 S1 包含 3 位或更多位数时，最低的 2 位数将被用作年数据。

有关源 2：小时数据可以为 0 ~ 65535。分和秒数据可以为 0 ~ 59。

目标 1：自动从年、月和日的合计计算星期，其结果将保存到设备 D1+3。

星期数据指的是：0 (星期日), 1 (星期一), 2 (星期二), 3 (星期三), 4 (星期四), 5 (星期五) 和 6 (星期六)

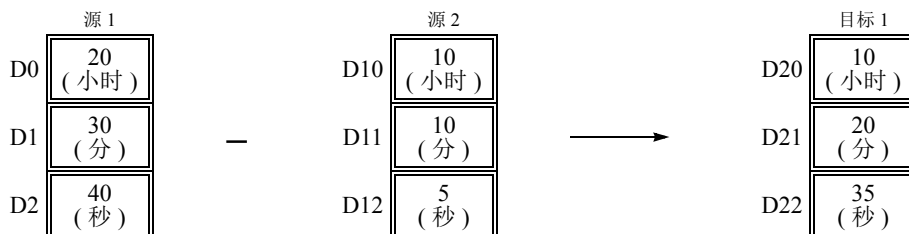
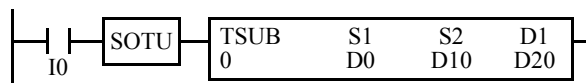
如果源 1 包含无效的日 / 时间数据，将导致程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

如果执行结果超出 00 年 1 月 1 日 00:00:00，将导致程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。

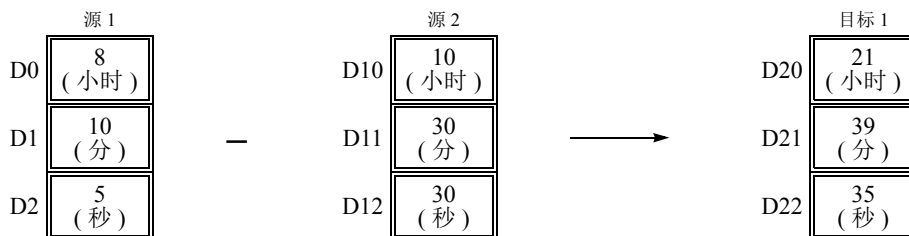
示例：TSUB

该示例演示了在两个不同模式中如何使用 TSUB 指令来减时间数据。

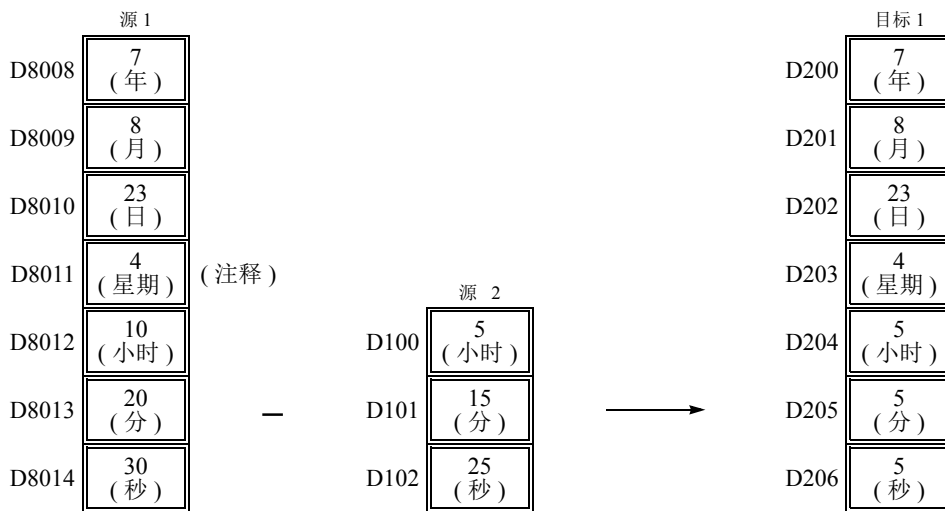
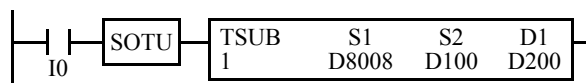
• 模式 0



当结果小于 00:00:00，合计的小时数据被加上 24，并打开特殊内部继电器 M8003 (借位)。



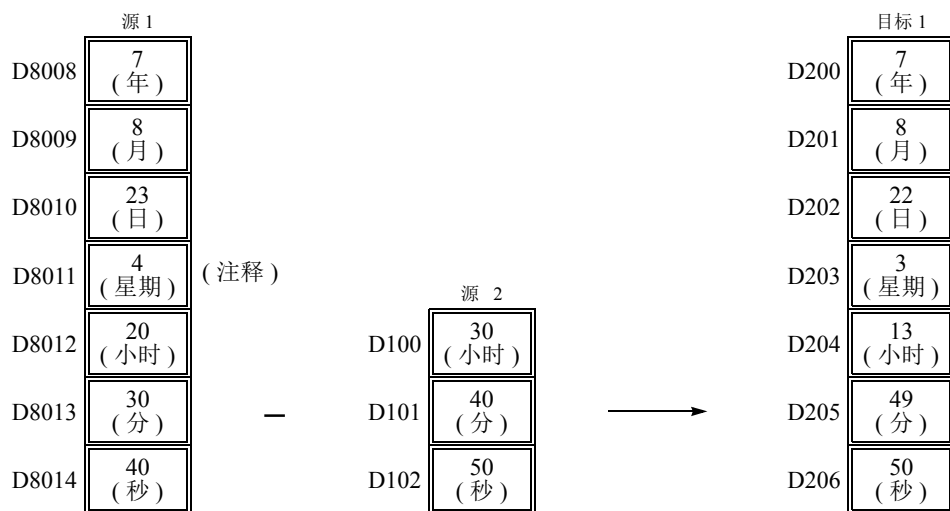
• 模式 1



**注释：**源 1 中的 D8011 没有被指定使用，并且无需指定。星期将自动从年、月和日的合计中被计算，并存储在目标 1 的 D203。

## 20: 时钟指令

当结果小于 00:00:00，合计的小时数据将被加上 24 的倍数，并递减日数据。



**注释：**源 1 中的 D8011 没有被指定使用，并且无需指定。星期将自动从年、月和日的合计中被计算，并存储在目标 1 的 D203。

## HTOS(HMS → 秒)



小时、分、秒 → 秒

当输入打开，由源设备 S1 指定的小时、分和秒中的时间数据转换为秒。此结果存储到目标设备 D1。

该指令适用于升级后的系统程序版本 210 或更高的 CPU 模块。

## 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

## 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	小时、分和秒中的时间数据	-	-	-	-	-	-	X	-	-
D1 (目标 1)	用于存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

源设备 S1 占用由设备指定开始的用来存储双字数据的 3 个连续的数据寄存器。数据寄存器 D0 ~ D1997、D2000 ~ D7997、D10000 ~ D49997 可以指定为源设备 S1。

目标设备 D1 占用由设备指定开始的用来存储双字数据的 2 个连续的数据寄存器。数据寄存器 D0 ~ D1998、D2000 ~ D7998 和 D10000 ~ D49998 可以指定为目标设备 D1。

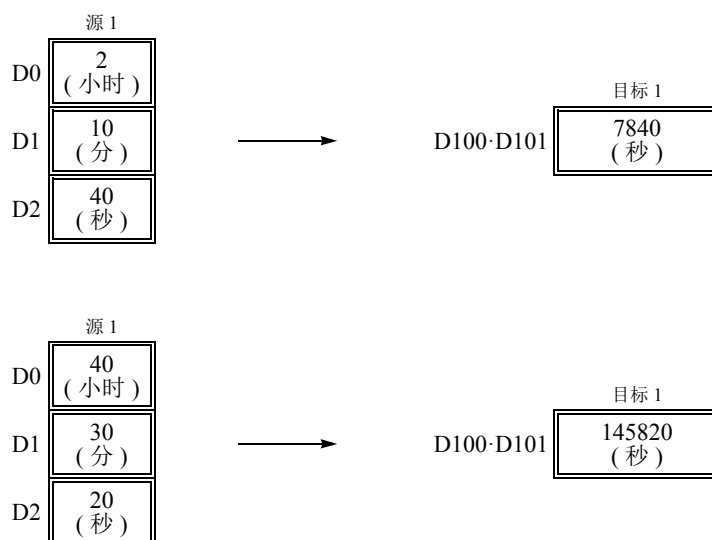
小时数据可以为 0 ~ 65535。分和秒数据可以为 0 ~ 59。

如果小时、分和秒数据超出有效范围将导致程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED。该指令将不被执行。

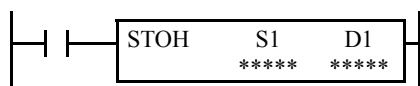
由于 HTOS 指令是在输入打开时的每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

## 示例：HTOS

该示例演示了 HTOS 指令如何将小时、分和秒中的时间数据转换为秒，并将其结果存储到 2 个连续的数据寄存器。



### STOH (秒 → HMS)



秒 → 小时、分、秒

当输入打开，由源设备 S1 指定的秒的时间数据转换为小时、分和秒。此结果存储到目标设备 D1。

该指令适用于升级后的系统程序版本 210 或更高的 CPU 模块。

#### 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 (源 1)	秒的时间数据	-	-	-	-	-	-	X	X	-
D1 (秒 1)	要存储结果的目标	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 (基本卷)。

源设备 S1 占用由设备指定开始的用来存储双字数据的 2 个连续的数据寄存器。数据寄存器 D0 ~ D1998、D2000 ~ D7998 和 D10000 ~ D49998 可以指定为源设备 S1。

目标设备 D1 占用由设备指定开始的 3 个连续数据寄存器。数据寄存器 D0 ~ D1997、D2000 ~ D7997 和 D10000 ~ D49997 可以指定为目标设备 D1。

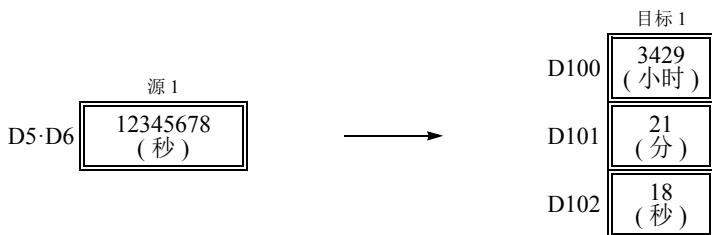
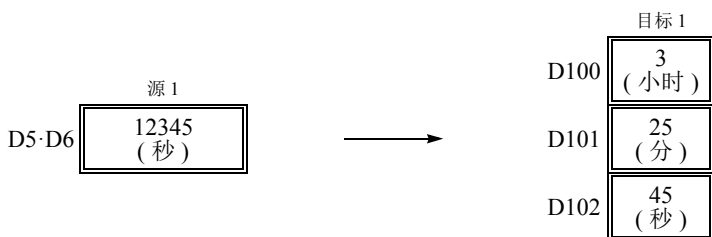
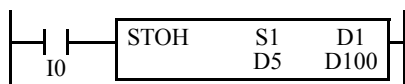
源设备 S1 的秒数据可以为 0 ~ 4,294,967,295。

如果转换结果超出 65535 小时 59 分 59 秒，将打开特殊内部继电器 M8003 (进位)。例如，当转换结果为 65537 小时 0 分 0 秒时，目标 1 存储 1 小时 0 分 0 秒，并打开特殊内部继电器 M8003 (进位)。

由于 SROH 指令是在输入打开时的每个扫描中执行的，因此，应当根据需要使用来自 SOTU 或 SOTD 指令的脉冲输入。

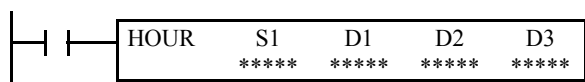
#### 示例 : STO H

该示例演示了 HTOS 指令如何将秒的时间数据转换为小时、分和秒，并将其结果存储到 3 个连续的数据寄存器。





## HOUR( 小时计量器 )



S1 ↔ D1 → D2

输入打开时，测量 ON 期间。测量到的时间值 ( 小时、分和秒 ) 被存储到由目标设备 D1 指定的 3 个连续的数据寄存器，并与由源设备 S1 指定的预置值比较。

当 D1 值到达 S1 值，将打开一个输出或由目标设备 D2 指定的内部继电器。

由目标设备 D3 指定开始的两个数据寄存器将预留为系统工作区。

该指令适用于升级后的 CPU 模块的系统程序版本 210 或更高版本。

### 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D	FC5A-C24R2/C/D	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
X	X	X	X	X	X

### 有效设备

设备	功能	I	Q	M	R	T	C	D	常量	重复
S1 ( 源 1 )	预置值	-	-	-	-	-	-	X	0 ~ 65535	-
D1 ( 目标 1 )	测量输入 ON 期间	-	-	-	-	-	-	X	-	-
D2 ( 目标 2 )	比较输出	-	X	▲	-	-	-	-	-	-
D3 ( 目标 3 )	系统工作区	-	-	-	-	-	-	X	-	-

关于有效设备编号范围，请参阅第 6-1 页和第 6-2 页 ( 基本卷 )。

源设备 S1 可以指定为数据寄存器或常量。

源设备 S1, 当指定为数据寄存器并且目标设备 D1 包含由存储小时、分和秒数据的设备指定开始的 3 个连续的数据寄存器。D0 ~ D1997、D2000 ~ D7997 和 D10000 ~ D49997 可以指定为这些设备。

当源设备 S1 被指定为常量时，小时中的预置值可以为 0 ~ 65535，分和秒可以设置为 0。

▲ 特殊内部继电器不可被指定为目标设备 D2。

目标设备 D3 要求 1 个数据寄存器预留为系统工作区。

小时数据可以为 0 ~ 65535。分和秒数据可以为 0 ~ 59。

当在目标设备 D1 测量到的输入 ON 期间值到达由源设备 S1 指定的预置值时，将打开由目标设备 D2 指定的比较输出。只要输入还存在，测量到的 ON 期间值将继续递增。当测量到的 ON 期间值超出 65535 小时 59 分 59 秒时，该值将返回到 0 小时 0 分 0 秒并重复另一个测量周期，比较输出保持打开。

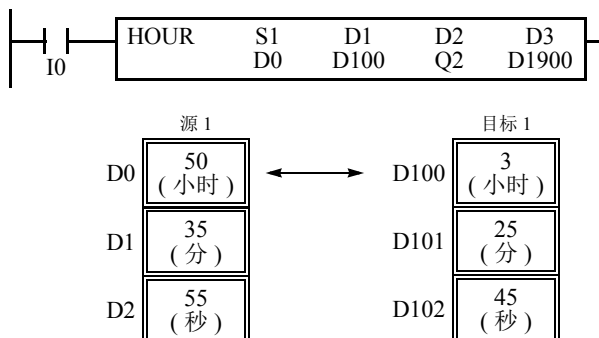
如果源设备 S1 的小时、分或秒数据超出有效范围将导致用户程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED，但输入 ON 期间将被测量。

当打开比较输出后，源设备 S1 的小时、分或秒数据被更改为无效数据时，关闭比较输出，这将导致用户程序执行错误，并打开 CPU 模块上的特殊内部继电器 M8004 和 ERR LED，但输入 ON 期间测量仍将继续。

示例: HOUR

该示例演示了 HOUR 指令如何测量小时、分和秒中的输入 ON 期间值并用两种不同的方法比较该值。

• 源设备 S1: 数据寄存器



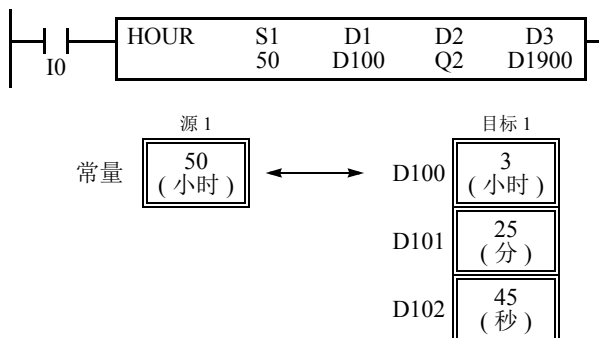
D0·D1·D2 ↔ D100·D101·D102 → Q2

打开输入 I0, 将测量 ON 期间。测量到的时间值 (小时、分和秒) 将存储到由目标设备 D1 指定的数据寄存器 D100 D101 D102, 并与存储在由源设备 S1 指定的数据寄存器中的预置值进行比较。

当测量到的值到达预置值, 将打开由目标设备 D2 指定的输出 Q2。

由目标设备 D3 指定的数据寄存器 D1900 和 D1901 将预留为系统工作区。

• 源设备 S1: 常量



50 ↔ D100·D101·D102 → Q2

打开输入 I0, 将测量 ON 期间。测量到的时间值 (小时、分和秒) 将存储到由目标设备 D1 指定的数据寄存器 D100 D101 D102, 并与由源设备 S1 指定 50 小时进行比较。

当测量到的值到达 50 小时, 将打开由目标设备 D2 指定的输出 Q2。

由目标设备 D3 指定的数据寄存器 D1900 和 D1901 将预留为系统工作区。

# 21: 计算机连接通信

## 简介

当 MicroSmart CPU 模块连接至计算机时，可以在计算机上监控运行状态和 I/O 状态，监控或更新 CPU 模块中的数据，以及下载和上传用户程序。还可以从计算机上启动和停止 CPU 模块。在 1:N 计算机连接系统中一台计算机最多可以连接 32 个 CPU 模块。

在 1:1 或 1:N 计算机连接系统中最大通信速度为 57,600 bps。

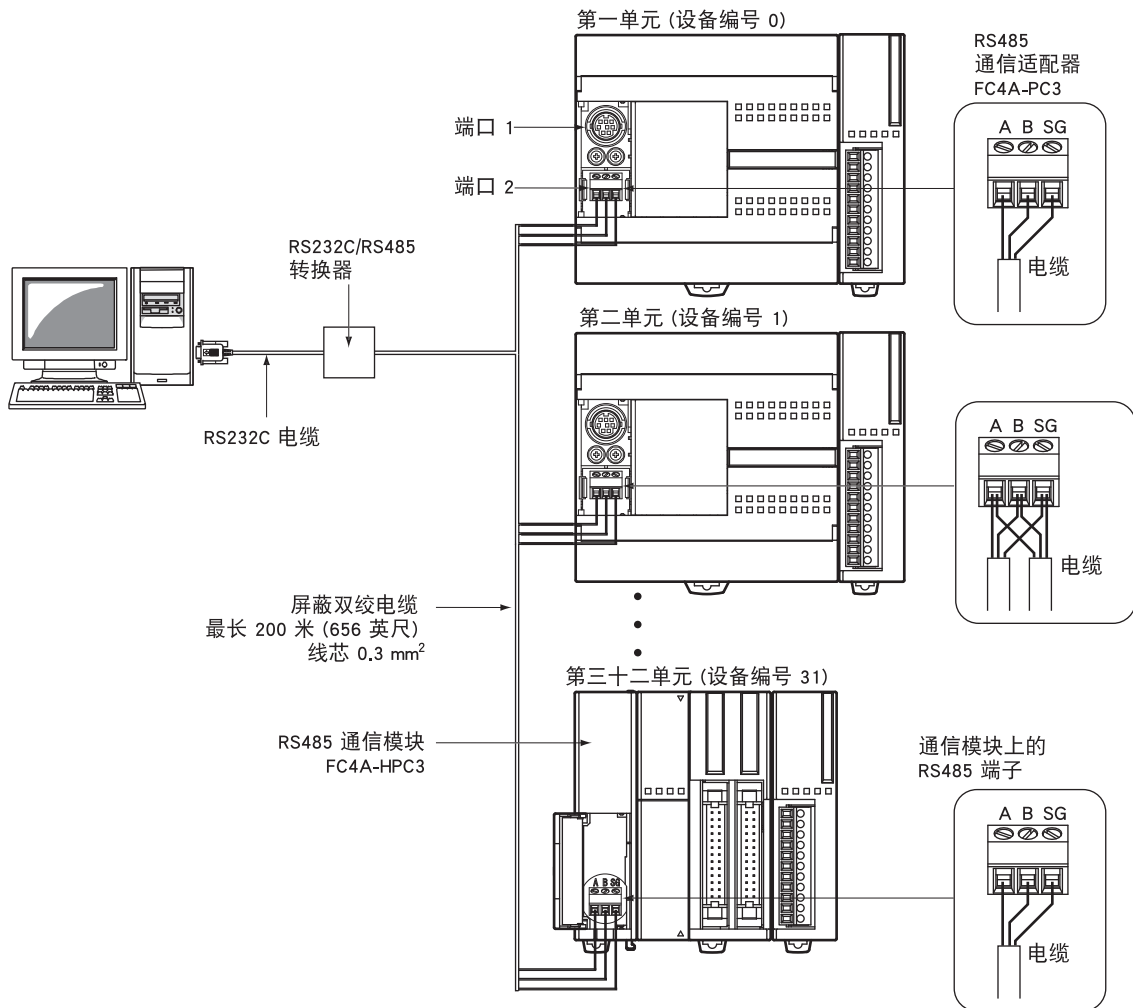
本章介绍 1:N 计算机连接系统。有关 1:1 计算机连接系统，请参阅第 4-1 页（基本卷）。有关使用扩展 RS232C/RS485 通信模块通过端口 3 ~ 7 进行连接通信，请参阅第 25-1 页。

## 计算机连接系统设置 (1:N 计算机连接系统)

要安装 1:N 通信计算机连接系统，将 RS485 通信适配器 (FC4A-PC3) 安装至集成型 CPU 模块的端口 2 连接器，或将 RS485 通信模块 (FC4A-HPC3) 安装在超薄型 CPU 模块上。使用如下所示的屏蔽双绞电缆将 RS232C/RS485 转换器连接至 CPU 模块上的 RS485 端子 A、B 和 SG。用于计算机连接系统的电缆总长度可以延伸至 200m (656ft.)。

使用 RS232C 电缆将计算机 RS232C 端口与 RS232C/RS485 转换器进行连接。RS232C 电缆有 D-sub 25 针插座型连接器可与计算机进行连接。

FC4A MicroSmart、OpenNet Controllers、MICRO³ 和 MICRO³C 可以连接至同一 1:N 计算机连接系统。



**注释：**使用第三方提供的 USB/RS485 转换器进行 1:N 通信计算机连接系统设置时，选择不会重复接收字符的 USB/RS485 转换器。

### 编程 WindLDR

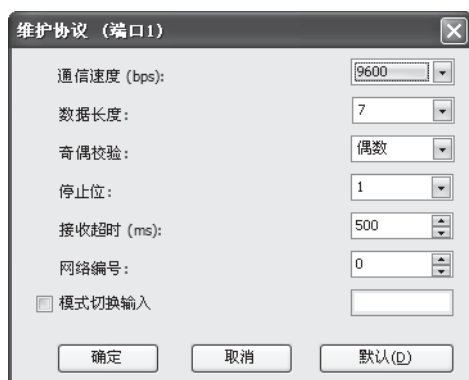
在 1:N 计算机连接系统中，必须有一台计算机连接到 CPU 模块上的端口 2，并且每个 CPU 模块都必须具有介于 0 到 31 之间的唯一网络编号。必须设置计算机连接系统中每台机器功能设置中的通信选项。若有必要，还要更改通信设置。

由于这些设置与用户程序相关，所以必须在更改设置后将用户程序下载到 MicroSmart。

1. 在 WindLDR 菜单栏中选择 **设置 > 功能设置 > 通信端口**。此时出现通信端口的“功能设置”对话框。
2. 在“端口 1”或“端口 2”的“通信模式”下拉列表中，选择**维护协议**。



3. 单击**设置**按钮。出现通信设置对话框。若有必要，请更改设置。



通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
数据长度	7 或 8
奇偶校验	无、奇数、偶数
停止位	1 或 2
接收超时时间 (ms):	10 ~ 2540( 增量为 10 ms) ( 当选择 2550 时禁用接收超时。 )
网络编号	0 ~ 31
模式选择输入	任一输入编号

**注释：**当已指定模式选择输入并且打开选择模式时，将启用选择的通信参数。没有指定模式选择输入更改通信参数的情况下，当下载用户程序时，更改的用户参数立即生效。

4. 单击**确定**按钮。

### 指定网络编号

在为 1:N 计算机连接网络中的每个 CPU 模块指定 0 到 31 间的唯一网络编号时，下载包含 1:1 计算机连接系统中每个 CPU 模块的网络编号设置的用户程序，新的网络编号将指定给 CPU 模块。确保在 1:N 计算机连接网络中没有重复的网络编号。

### 通信设置

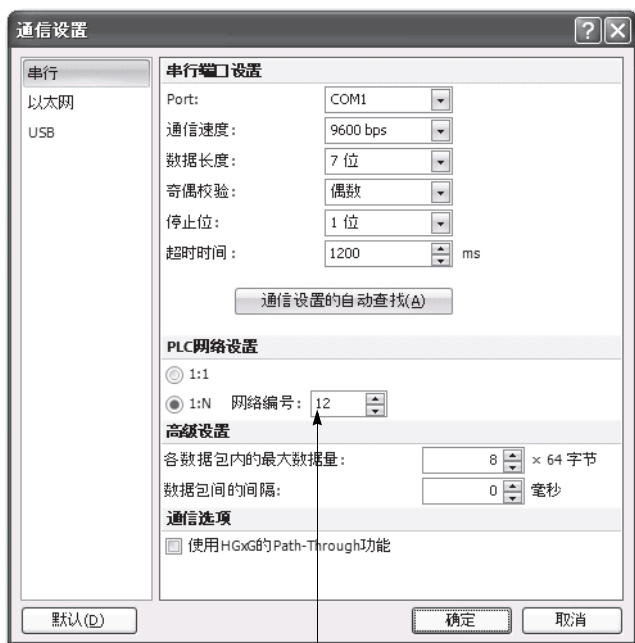
使用 WindLDR 监控 MicroSmart 操作或下载用户程序时，请确保 CPU 模块与 WindLDR 选择了相同的通信设置，以保证计算机能够与 1:1 或 1:N 计算机连接系统中的 MicroSmart 进行通信。要更改 WindLDR 的通信设置，请按如下所示打开设置菜单的通信设置对话框。

在 1:N 计算机连接系统中为实现监控或下载而进行通信时，也可以在“通信设置”对话框中选择 CPU 模块的网络编号。

## 监控 PLC 状态

下例介绍监控在 1:N 通信计算机连接系统中，网络编号指定为 12 的 MicroSmart 的运行状态的步骤。

1. 在 WindLDR 菜单栏中选择**联机 > 通信 > 设置**。出现通信设置对话框。
2. 在“PLC 网络设置”下面，单击“1:N”按钮以选择 1:N 通信，并在“网络编号”字段中选择 12。
3. 在 WindLDR 菜单栏中选择**联机 > 监控 > 启动监控**。在屏幕上的梯形图转换到监控模式。
4. 在 WindLDR 菜单栏中选择**联机 > PLC > 状态**。出现“PLC 状态”对话框。



**网络编号：**  
输入 12 以选择要与之通信的网络编号。





# 22: 调制解调器模式

## 简介

本章介绍在 MicroSmart 和另一个 MicroSmart 之间，或通过电话线与任一数据终端设备进行通信所用的调制解调器模式。使用调制解调器模式，MicroSmart 可初始化调制解调器，拨打电话号码，发送 AT 命令，启用应答模式以等待来电和断开电话线路。通过打开各个操作专用的启动内部继电器可以简单地执行这些操作。

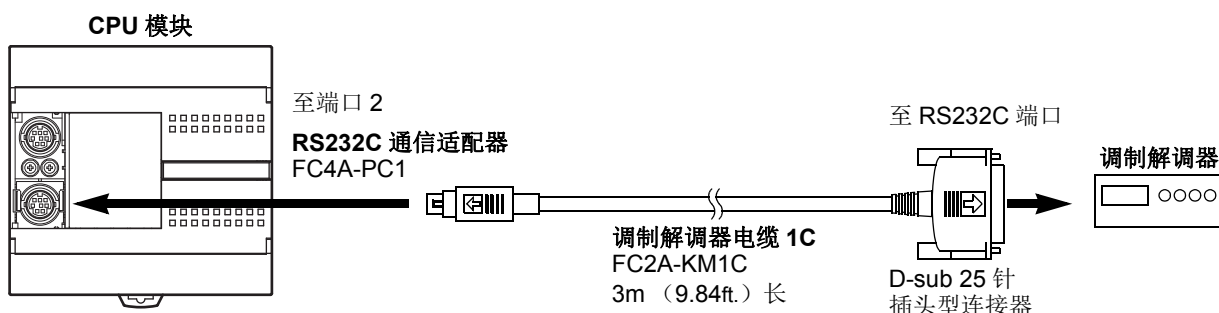


### 注意

- 调制解调器模式提供了简单的调制解调器控制功能，使 MicroSmart 可以初始化调制解调器、拨打目标电话号码或应答来电。使用调制解调器通信模式的调制解调器通信的性能视调制解调器的功能和电话线路情况而定。调制解调器模式无法避免其他系统受到干扰或出现故障。有关实际应用情况，请使用实际系统设置确认通信功能以及安全注意事项。
- 当通过调制解调器通信时，电话线路可能会意外断开或出现数据接收错误。用户程序中必须包括针对这些错误的解决方案。

## 系统安装

要连接调制解调器至 MicroSmart，将 RS232C 通信适配器（FC4A-PC1）安装至集成型 CPU 模块的端口 2 连接器，或将 RS232C 通信模块（FC4A-HPC1）安装在超薄型 CPU 模块旁边，并使用调制解调器电缆 1C（FC2A-KM1C）。要启用调制解调器模式，使用 WindLDR 选择端口 2 的调制解调器通信（设置 > 功能设置 > 通信端口）。



微型 DIN 连接器插针

说明	插针编号
屏蔽	盖
RTS 请求发送	1
DTR 数据终端准备就绪	2
TXD 发送数据	3
RXD 接收数据	4
DSR 数据设置就绪	5
SG 信号接地	6
SG 信号接地	7
NC 没有连接	8

D-sub 25 针连接器插针

插针编号	说明
1	FG 帧接地
2	TXD 发送数据
3	RXD 接收数据
4	RTS 请求发送
5	NC 没有连接
6	NC 没有连接
7	SG 信号接地
8	DCD 数据载体探测
20	DTR 数据终端准备就绪



### 注意

- 请勿将 NC（没有连接）插针编号与任何线缆相连；否则可能损坏 MicroSmart 或调制解调器。
- Apple Macintosh 计算机的调制解调器电缆不能用于 MicroSmart。
- 请勿用电缆连接端口 1 或端口 2（RS485）；否则可能损坏 MicroSmart 或调制解调器。

## 22: 调制解调器模式

### 可用调制解调器

可以使用 Hayes 兼容调制解调器。建议使用通信速度大于或等于 9600 bps 的调制解调器。在通信线路两端使用制造厂商和型号相同的调制解调器。

### 调制解调器模式下使用的特殊内部继电器

特殊内部继电器 M8050 ~ M8077 可分配给调制解调器模式。M8050 ~ M8056 用于发送 AT 命令或断开电话线路。打开 M8060 ~ M8066 和 M8070 ~ M8076 可指示命令结果。M8057、M8067 和 M8077 用于指示调制解调器模式的状态。

当打开另一个开始内部继电器时，将关闭所有完成和失败内部继电器。

#### 开始和结果内部继电器

模式	命令	开始 IR	完成 IR	失败 IR	数据寄存器
初始模式	初始化字符串	M8050	M8060	M8070	D8145-D8169
	ATZ	M8051	M8061	M8071	—
	拨号	M8052	M8062	M8072	D8170-D8199
断开模式	断开线路	M8053	M8063	M8073	—
AT 常规命令模式	AT 命令	M8054	M8064	M8074	D8130-D8144
应答模式	初始化字符串	M8055	M8065	M8075	D8145-D8169
	ATZ	M8056	M8066	M8076	—

当打开一个开始内部继电器 M8050 ~ M8056 时，将执行一次相应的命令。要重复该命令，复位开始内部继电器，然后再次打开内部继电器。

如下所示确定命令完成或失败：

**完成：** 命令重复发送的次数与在数据寄存器 D8109 中指定的重试次数相同。当命令成功完成时，完成 IR 将打开，且该命令在剩下的循环中不再执行。

**失败：** 命令重复发送，但是重试失败的次数与在数据寄存器 D8109 中指定的重试次数相同。

#### 状态内部继电器

状态 IR	状态	说明
M8057	AT 命令执行	开： AT 命令正在执行（开始 IR 已打开） 关： AT 命令未在执行（完成或失败 IR 已打开）
M8067	运行状态	开： 命令模式 关： 联机模式
M8077	线路连接	开： 电话线路已连接（注释） 关： 电话线路已断开

**注释：**当 M8077（线路连接）关闭时，MicroSmart 无法通过端口 2 发送和接收维护协议和用户协议。当 M8077 打开时，根据数据寄存器 D8103（联机模式协议选择）中存储的值启用维护协议或用户协议。



## 用于调制解调器模式的特殊数据寄存器

特殊数据寄存器 D8103 和 D8109 ~ D8199 分配给调制解调器模式。当 **MicroSmart** 开始运行时，D8109 和 D8110 存储默认值，D8145 ~ D8169 存储默认初始化字符串。

数据寄存器	存储的数据	说明
D8103	联机模式 协议选择	在电话线路连接后，D8103 值选择用于 RS232C 端口 2 的协议。 0 (1 除外)： 维护协议 1: 用户协议
D8109	重试次数 (默认值 = 3)	D8109 值选择在开始内部继电器 M8050 ~ M8056 完成初始化操作之前的重试次数。 0: 不重试 1-65535: 执行指定次数的重试
D8110	重试间隔 (默认值 = 90s)	当拨号失败，而设置的重试次数大于 1 时，D8110 值指定重试拨号的时间间隔。(按重试次数设置，连续重复其他开始命令。) 有效值： 0 ~ 65535 (s) 如果在重试间隔内电话线路没有连接，则 <b>MicroSmart</b> 将开始重试。最终，如果重试间隔设置得太小，则电话线路无法正确连接。
D8111	调制解调器模式 状态	存储调制解调器模式状态 (请参阅第 22-7 页)。当不在调制解调器模式下时，D8111 存储 0。
D8115-D8129	AT 命令结果代码	存储从调制解调器返回的 AT 命令结果。当结果代码超过 30 字节时，将存储前 30 个字节。
D8130-D8144	AT 命令字符串	存储 AT 常规命令的 AT 命令字符串。打开 M8054 输入 AT 命令字符串至这些数据寄存器中进行发送 (AT 命令开始内部继电器)。“AT”和 LF (0Ah) 会自动添加。
D8145-D8169	初始化字符串	存储初始模式和应答模式的初始化字符串。 要更改初始化字符串，请向这些数据寄存器中输入新值。可以通过打开 M8050 或 M8055 发送新值。“AT”和 LF (0Ah) 会自动添加。
D8170-D8199	电话号码	存储初始模式下的拨号电话号码。“ATD”和 LF (0Ah) 会自动添加。

## 初始模式

初始模式用于向调制解调器发送初始化字符串，发出 ATZ 命令以复位调制解调器和拨打电话号码。要执行命令，请打开一个开始内部继电器 M8050 ~ M8052。如果同时打开两个或更多开始内部继电器，将出现错误，且错误代码 61 会存储在调制解调器模式状态数据寄存器 D8111 中 (请参阅第 22-7 页)。当开始内部继电器打开时，将按如下所述执行一次相应的命令序列。当启动命令失败时，将按 D8109 指定的重试次数重复执行该命令。

- M8050: 发送初始化字符串，发送 ATZ 命令和拨打电话号码
- M8051: 发送 ATZ 命令和拨打电话号码
- M8052: 拨打电话号码

### 初始模式下的初始化字符串

当如第 22-1 页所述启用调制解调器模式，且 **MicroSmart** 开始运行时，默认初始化字符串将在第一次扫描的 END 处理阶段存储至数据寄存器 D8145 ~ D8169 中。要将 **MicroSmart** 中的初始化字符串发送至调制解调器，请打开 M8050；然后发送 ATZ 命令以连续拨打电话号码。

默认初始化字符串：ATE0Q0V1&D2&C1V0X4&K3VA0N5S0=2&W CR LF

## 22: 调制解调器模式

AT 和  $\boxed{\text{LF}}$  由系统程序自动添加到初始化字符串的开始和结尾，而不存储在数据寄存器中。

DR 8145 8146 8147 8148 8149 8150 8151 8152 8153 8154 8155 8156 8157 8158 8159 8160 8161  
 AT  $\boxed{\text{E0}} \boxed{\text{Q0}} \boxed{\text{V1}} \boxed{\&\text{D}} \boxed{\&\text{2}} \boxed{\text{C1}} \boxed{\text{W}} \boxed{\text{0X}} \boxed{\&\text{4}} \boxed{\text{K3}} \boxed{\text{VA}} \boxed{\text{0V}} \boxed{\text{N5}} \boxed{\text{S0}} \boxed{=\text{2}} \boxed{\&\text{W}} \boxed{\text{0D00}} \boxed{\text{LF}}$

根据调制解调器和电话线路，可能必须修改初始化字符串。请参考调制解调器的用户手册。

向数据寄存器 D8145 ~ D8169 中输入所需值进行更改。在一个数据寄存器中存储两个字符：第一个字符位于数据寄存器的高位字节，第二个字符位于数据寄存器的低位字节。AT 和  $\boxed{\text{LF}}$  无需存储在数据寄存器中。使用 WindLDR 的 MOV（移动）指令在结尾处设置初始化字符串字符和 CR 的 ASCII 值 0Dh  $\boxed{\text{CR}}$  编译 MOV 指令替换第一次扫描中存储的 D8145 ~ D8169 中的默认值，然后在后续扫描中执行 MOV 指令。有关初始化字符串中必须包含的关键命令，请参阅第 22-8 页。新值存储后，打开 M8050 以向调制解调器发送新的初始化字符串。

当初始化字符串成功发送后，内部继电器 M8060 将打开。如果初始化字符串失败，则内部继电器 M8070 打开。当 ATZ 的后续命令和拨号也成功完成时，M8061 和 M8062 也将打开。

存储在 D8145 ~ D8169 中的默认初始化字符串或修改的初始化字符串还用于应答模式的初始化。

### 初始模式下的 ATZ（复位调制解调器）

使用 &W 命令指定存储在调制解调器的非暂时性内存中的默认初始化字符串。当调制解调器开机或发送 ATZ 命令时，初始化字符串恢复。当 M8050 打开时，MicroSmart 先向调制解调器发送初始化字符串，然后发送 ATZ 命令。还可以通过打开 M8050 单独发送 ATZ 命令，然后自动执行拨号命令。

ATZ 命令： $\text{ATZ} \boxed{\text{CR}} \boxed{\text{LF}}$

当 ATZ 命令成功完成后，内部继电器 M8061 将打开。如果 ATZ 命令失败，则内部继电器 M8071 打开。当后续拨号也成功完成时，M8062 也将打开。

如果初始化字符串已存储在调制解调器的非暂时性内存中，可能将跳过 M8050。启动 M8051 以发送 ATZ 命令。

### 拨打电话号码

给电话号码分配数据寄存器 D8170 ~ D8199。在打开一个开始内部继电器 M8050 ~ M8052 以用于初始模式前，将电话号码存储至以 D8170 开始的数据寄存器中。一个数据寄存器存储两个字符：第一个字符位于数据寄存器的高位字节，第二个字符位于数据寄存器的低位字节。给电话号码分配了 30 个数据寄存器，受调制解调器功能限制，最多能存储 60 个字符。使用 WindLDR 的 MOV（移动）指令设置电话号码，并在打开开始内部继电器 M8050 ~ M8052 之前执行 MOV 指令。

拨号命令示例： $\text{ATD1234} \boxed{\text{CR}} \boxed{\text{LF}}$

ATD 和  $\boxed{\text{LF}}$  由系统程序自动添加到拨号命令的开始和结尾，而不是存储在数据寄存器中。要编译以上示例的电话号码，请存储电话号码和 CR 的 ASCII 值 0Dh  $\boxed{\text{CR}}$  至以 D8170 开始的数据寄存器中。还可以存储单键拨号电话的字符 T 或脉冲电话的字符 P。

D8170	$\boxed{\text{3132h}}$	31h = "1"	32h = "2"
D8171	$\boxed{\text{3334h}}$	33h = "3"	34h = "4"
D8172	$\boxed{\text{0D00h}}$	0Dh = $\boxed{\text{CR}}$	所有后续字符 $\boxed{\text{CR}}$ 都将忽略。

如上所述，当打开开始内部继电器 M8050 时，将依次发送初始化字符串、ATZ 命令和拨号命令。当开始内部继电器 M8051 打开时，将依次发送 ATZ 命令和拨号命令。还可以打开开始内部继电器 M8052 单独发送拨号命令。

如果数据寄存器 D8109 中设置了重试次数，则可以按照 D8110 指定的重试间隔（默认 90s）重复拨号命令（默认 3 次）直至建立电话线路连接。

当拨号命令成功完成后，内部继电器 M8062 将打开。如果拨号命令失败，则内部继电器 M8072 打开。

当 DCD 信号打开时，拨号命令已成功完成。

**注释：**若在电话线路已连接时断开 MicroSmart 电源，则由于 DTR 信号关闭而断开电话线路。不能使用该方法断开电话线路。始终如下所述使用 M8053 断开电话线路。

### RS232C 端口通信协议

在开机后调制解调器模式下连接电话线路之前，通过打开开始内部继电器 M8050 ~ M8056，RS232C 端口 2 只能发出 AT 命令。在电话线路连接后 RS232C 端口 2 的通信协议由数据寄存器 D8103 中存储的值确定。

D8103 值	联机模式下 RS232C 端口 2 通信协议
0 (1 除外)	维护协议
1	用户协议

电话线路连接后，无论 D8103 设为 0 还是 1，RS232C 端口 2 的状态将恢复至电话线路连接前。

在电话线路连接后，若在用户通信模式下使用 TXD 或 RXD 指令，可插入内部继电器 M8077（线路连接）作为 TXD 或 RXD 指令的输入条件。在电话线路连接后，确保在执行 TXD 或 RXD 指令之前有大约 1 秒钟的间隔直至电话线路连接稳定下来。

**注释：**当电话线路连接期间停止 MicroSmart 时，即使 D8103 设为 1(联机模式用户通信协议)，RS232C 端口 2 协议也会更改为维护协议；然后电话线路将保持连接。当 MicroSmart 重新启动后，将再次启用用户协议。

### 断开模式

断开模式仅包括一个断开电话线路的命令。要断开电话线路，请打开内部继电器 M8053。由于初始化字符串中包括 &D2 命令，则关闭 DTR 信号即可断开电话线路。

在执行调制解调器命令时，不能执行另一个命令。如果同时打开两个或更多开始内部继电器，将出现错误，且错误代码 61 会存储在调制解调器模式状态数据寄存器 D8111 中（请参阅第 22-7 页）。

当断开命令成功完成后，内部继电器 M8063 将打开。如果断开命令失败，则内部继电器 M8073 打开。

当 DCD 信号关闭时，断开命令已成功完成。

在电话线路断开后，无论 D8103 设为 0 还是 1，RS232C 端口 2 都将恢复至电话线路连接前的状态，这时通过打开开始内部继电器 M8050 ~ M8056 可以控制 RS232C 端口 2。

### AT 常规命令模式

给 AT 命令字符串分配数据寄存器 D8130 ~ D8144。在打开开始内部继电器 M8054 以用于 AT 常规命令模式之前，将 AT 命令字符串存储至以 D8130 开始的数据寄存器中。一个数据寄存器存储两个字符：第一个字符位于数据寄存器的高位字节，第二个字符位于数据寄存器的低位字节。使用 WindLDR 的 MOV（移动）指令设置 AT 命令字符串，并在打开 M8054 之前执行 MOV 指令。

AT 命令示例：     ATE0Q0V1 $\boxed{\text{CR}}$  $\boxed{\text{LF}}$

AT 和 $\boxed{\text{LF}}$ 由系统程序自动添加到 AT 常规命令的开始和结尾，而不是存储在数据寄存器中。要编译以上示例的 AT 命令字符串，请存储命令字符和 CR 的 ASCII 值 0Dh $\boxed{\text{CR}}$ 至以 D8130 开始的数据寄存器中。

D8130	$\boxed{4530\text{h}}$	45h = "E"	30h = "0"
D8131	$\boxed{5130\text{h}}$	51h = "Q"	30h = "0"
D8132	$\boxed{5631\text{h}}$	56h = "V"	31h = "1"
D8133	$\boxed{0D00\text{h}}$	0Dh = $\boxed{\text{CR}}$	所有后续字符 $\boxed{\text{CR}}$ 都将忽略。

当 ATZ 常规命令成功完成后，内部继电器 M8064 将打开。如果 ATZ 常规命令失败，则内部继电器 M8074 打开。

当收到从调制解调器返回的结果代码时， $\boxed{\text{CR}}$  $\boxed{\text{LF}}$ 确定 $\boxed{\text{CR}}$  $\boxed{\text{LF}}$ AT 常规命令成功完成。

### 应答模式

应答模式用于向调制解调器发送初始化字符串，以及发出 ATZ 命令以复位调制解调器。要执行命令，请打开一个开始内部继电器 M8055 或 M8056。如果同时打开两个或更多开始内部继电器，将出现错误，且错误代码 61 会存储在调制解调器模式状态数据寄存器 D8111 中（请参阅第 22-7 页）。当开始内部继电器打开时，将按如下所述执行一次相应的命令序列。

**M8055:** 发送初始化字符串和 ATZ 命令

**M8056:** 发送 ATZ 命令

#### 应答模式下的初始化字符串

当如第 22-1 页所述启用调制解调器模式，且 MicroSmart 开始运行时，默认初始化字符串将在第一次扫描的 END 处理阶段存储至数据寄存器 D8145 ~ D8169 中。要从数据寄存器向调制解调器发送初始化字符串，请打开 M8055；然后发送 ATZ 命令。

默认初始化字符串：ATE0Q0V1&D2&C1\VOX4&K3\A0\N5S0=2&W CR LF

如在初始模式中所述，可以修改初始化字符串与调制解调器相匹配。有关修改初始化字符串的详细信息，请参阅第 22-3 页。

当初始化字符串成功发送后，内部继电器 M8065 将打开。如果初始化字符串失败，则内部继电器 M8075 打开。当后续 AT 命令也成功完成时，M8066 也将打开。

#### 应答模式下的 ATZ（复位调制解调器）

使用 &W 命令指定存储在调制解调器的非暂时性内存中的默认初始化字符串。当调制解调器开机或发送 ATZ 命令时，初始化字符串恢复。当 M8055 打开时，MicroSmart 先向调制解调器发送初始化字符串，然后发送 ATZ 命令。通过打开 M8056 也可单独发送 ATZ 命令。

ATZ 命令： ATZ CR LF

当 ATZ 命令成功完成后，内部继电器 M8066 将打开。如果 ATZ 命令失败，则内部继电器 M8076 打开。

如果初始化字符串已存储在调制解调器的非暂时性内存中，可能将跳过 M8055。启动 M8056 以发送 ATZ 命令。

## 调制解调器模式状态数据寄存器

当启用调制解调器模式时，数据寄存器 D8111 存储调制解调器模式状态或错误代码。

D8111 值	状态	说明
0	不在调制解调器模式下	调制解调器模式未启用。
10	准备连接线路	可以打开开始内部继电器（断开线路除外）。
20	发送初始化字符串 (初始模式)	开始内部继电器正在操作，在第一次重试或后续重试中。
21	发送 ATZ (初始模式)	
22	拨号	
23	断开线路	
24	发送 AT 命令	
25	发送初始化字符串 (应答模式)	
26	发送 ATZ (应答模式)	
30	等待重新发送初始化字符串 (初始模式)	
31	等待重新发送 ATZ (初始模式)	
32	等待重新拨号	
33	等待重新断开线路	
34	等待重新发送 AT 命令	
35	等待重新发送初始化字符串 (应答模式)	
36	等待重新发送 ATZ (应答模式)	
40	线路已连接	电话线路已连接。只能打开 M8053 (断开线路)。
50	AT 命令成功完成	M8054 ~ M8056 启动的命令已成功完成。
60	AT 命令程序错误	初始化字符串、拨号号码或 AT 命令字符串中包括无效字符。 修正程序以在 AT 命令中包括 0Dh。
61	同时启动多个命令	打开两个或更多开始内部继电器。 修正用户程序以使一次仅打开一个开始内部继电器。
62	联机模式下的无效命令	当电话线路已连接时，打开 M8053 (断开线路) 以外的其他启动 IR。 修正该程序以在线路连接后仅发送断开命令。
63	AT 命令执行错误	经过第一次和所有重试后命令失败。

## 22: 调制解调器模式

### 初始化字符串命令

内置初始化字符串（请参阅第 22-3 页）包括以下命令。有关调制解调器命令的详细信息，请参阅调制解调器的用户手册。当编写可选初始化字符串时，可修改初始化字符串以与使用的调制解调器匹配。

<b>E0</b>	字符无载波。 MicroSmart 的调制解调器模式操作没有载波。没有 E0 命令，MicroSmart 将误解结果代码的载波。尽管命令执行正确也会出现错误。 初始化字符串中必须包括该命令。
<b>Q0</b>	显示结果代码。 设置 MicroSmart 的调制解调器模式以使用结果代码。没有 Q0 命令，尽管命令执行正确也会出现超时错误。 初始化字符串中必须包括该命令。
<b>V1</b>	字结果代码。 设置 MicroSmart 的调制解调器模式以使用字结果代码。没有 V1 命令，结果代码将视为无效，尽管命令执行正确也会出现超时错误。 初始化字符串中必须包括该命令。
<b>&amp;D2</b>	在 DTR 检测时挂起和禁用自动应答。 当 DTR 信号关闭时，电话线路将断开。MicroSmart 使用此功能断开电话线路。 初始化字符串中必须包括该命令。
<b>&amp;C1</b>	DCD 打开带有来自远程调制解调器的载波。 DCD 跟踪来自远程调制解调器的数据载波的状态。DCD 的打开条件表示载波存在。 初始化字符串中必须包括该命令。
<b>\V0</b>	禁用 MNP 结果代码。 使用常规结果代码，不使用可靠连接结果代码。
<b>X4</b>	启用拨号音和忙音检测。
<b>&amp;K3</b>	启用硬件流控制。 软件流控制（XON/XOFF）不能用在 MicroSmart 调制解调器模式下。 初始化字符串中必须包括该命令。
<b>\A0</b>	设置 MNP 块大小最大为 64 字节。
<b>\N5</b>	MNP 自动可靠模式
<b>S0=2</b>	响铃应答打开。 指定响铃时调制解调器接通电话线路。S0=2 指定调制解调器检测到 2 个响铃时应答来电。S0=0 禁用自动应答功能。
<b>&amp;W</b>	写入当前设置文件。 将当前设置文件存储至调制解调器的非暂时性内存中。

## 使用调制解调器的准备

使用调制解调器之前，请阅读调制解调器的用户手册。

所需初始化字符串取决于调制解调器的型号和性能。当 MicroSmart 开始运行用户程序时，默认调制解调器初始化字符串将存储至 D8145 ~ D8169。请参阅第 22-3 页。

默认初始化字符串：ATE0Q0V1&D2&C1\VOX4&K3\A0\N5S0=2&W CR LF

## 设置数据寄存器和内部继电器

要启用调制解调器模式和通过电话线路通信，需要进行以下设置。

1. 如果默认初始化字符串与使用的调制解调器不匹配，请编译正确的初始化字符串并将 ASCII 值输入以 D8145（初始化字符串）开始的数据寄存器中。要发送新的初始化字符串，请在将新值存储到数据寄存器中后，打开内部继电器 M8050（初始化字符串开始 IR）。
2. 在电话线路已连接后，将 0 或 1 移至数据寄存器 D8103（联机模式协议选择）中以选择维护协议或用户协议供 RS232C 端口 2 使用。
3. 如果需要拨号，请输入目标电话号码。将电话号码的 ASCII 值输入以 D8170（电话号码）开始的数据寄存器中。一个数据寄存器存储两个字符。在电话号码末尾输入 0Dh。请参阅第 22-4 页。
4. 如果要更改 3 次重试的默认值，请将所需值移至数据寄存器 D8109 中。
5. 根据需要将内部继电器 M8050 ~ M8077 加入用户程序以控制调制解调器通信。

## 设置 CPU 模块

1. 将 RS232C 通信适配器（FC4A-PC1）安装至集成型 CPU 模块的端口 2 连接器。

在使用超薄型 CPU 模块时，请将 RS232C 通信模块（FC4A-HPC1）安装在超薄型 CPU 模块上，并使用 RS232C 通信模块的端口 2。

将 HMI 主机模块与超薄型 CPU 模块一起使用时，请在 HMI 主机模块的端口 2 连接器上安装 RS232C 通信适配器（FC4A-PC1）。

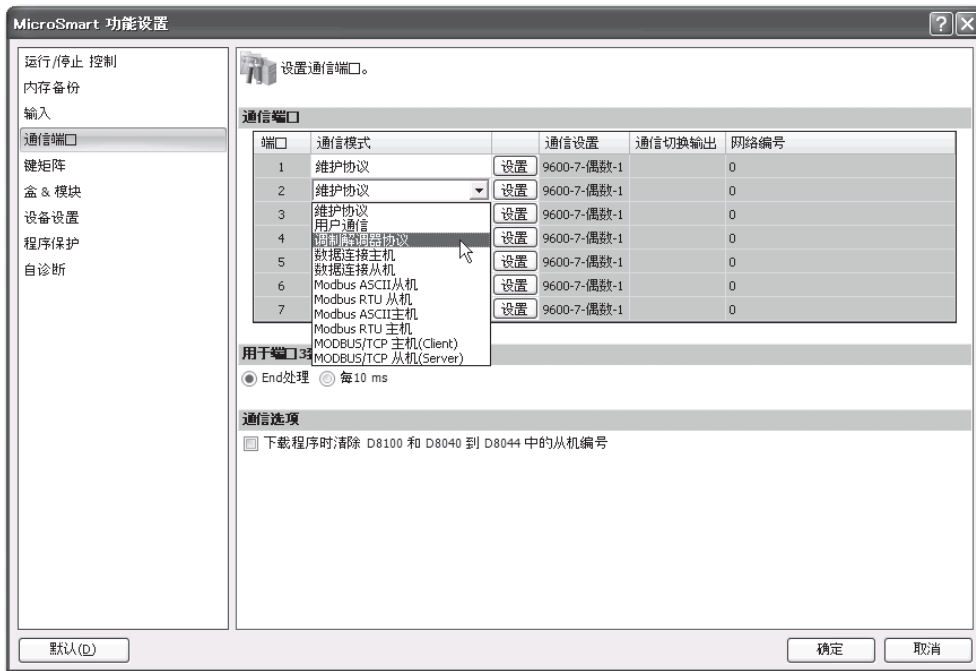
2. 如第 22-1 页上所示，使用调制解调器电缆 1C（FC2A-KM1C）将 MicroSmart CPU 模块端口 2 与调制解调器相连。

## 设置 WindLDR

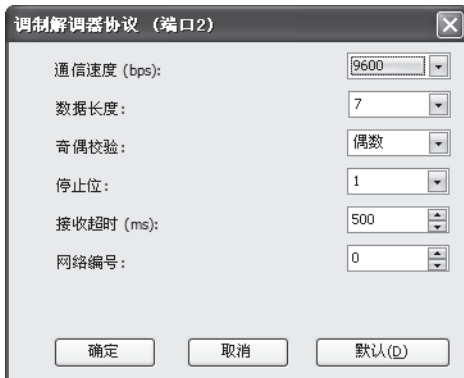
必须设置功能设置中的通信选项以启用端口 2 的调制解调器模式。若有必要，还可以更改 CPU 模块端口 2 的通信设置。

由于这些设置与用户程序相关，所以必须在更改设置后将用户程序下载到 MicroSmart。

1. 在 WindLDR 菜单栏中选择**设置 > 功能设置 > 通信端口**。此时出现通信端口的“功能设置”对话框。
2. 在“端口 2”的“通信模式”下拉列表中，选择**调制解调器协议**。



3. 单击**设置**按钮。出现通信设置对话框。若有必要，请更改设置。



通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600
数据长度	7 或 8
奇偶校验	无、奇数、偶数
停止位	1 或 2
接收超时时间 (ms):	10 ~ 2540 (增量为 10 ms) (当选择 2550 时禁用接收超时。)
网络编号	0 ~ 31

建议使用如下所示的默认通信设置。

通信速度	9600 bps
开始位	1
数据长度	7
奇偶校验	偶数
停止位	1
总计	10 位

仅当通信线路连接的调制解调器使用与 MicroSmart 的默认值不同的通信设置时，才设置匹配的通信参数。由于调制解调器通信参数总计 10 位，请设置该值总计 10 位。

4. 单击**确定**按钮。



### 调制解调器模式操作步骤

1. 完成包括功能设置的用户程序后，从运行 WindLDR 的计算机下载用户程序至 MicroSmart 中。
2. 启动 MicroSmart 以运行用户程序。
3. 打开开始内部继电器 M8050 或 M8055 以初始化调制解调器。

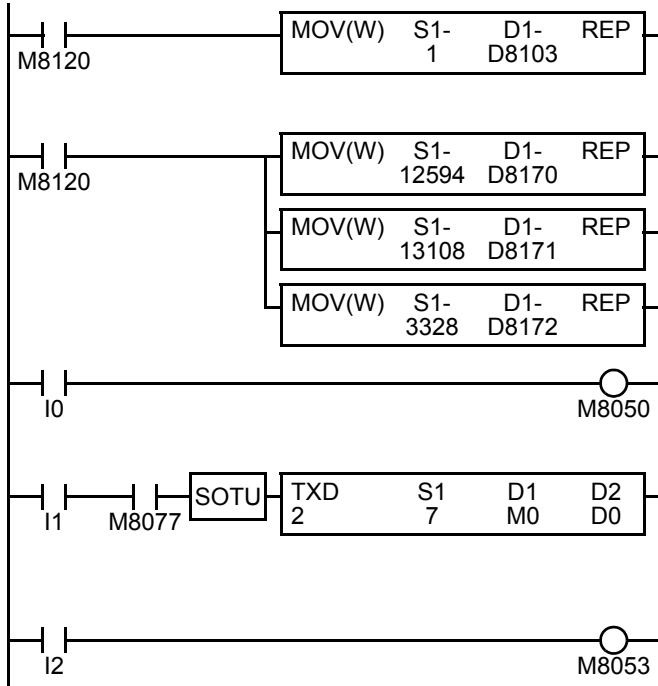
在初始化调制解调器通信时，打开 M8050 以发送初始化字符串、ATZ 命令和拨号命令。如果初始化字符串已存储在调制解调器的非暂时性内存中，在发送拨号命令后打开以 ATZ 命令开始的 M8051。

当应答来电时，打开 M8055 以发送初始化字符串和 ATZ 命令。如果初始化字符串已存储在调制解调器的非暂时性内存中，打开 M8056 仅发送 ATZ 命令。

4. 通过调制解调器发送或接收通信。
5. 打开开始内部继电器 M8053 以断开电话线路。

### 调制解调器初始模式示例程序

该程序演示在调制解调器初始模式下将值移至分配给调制解调器模式的数据寄存器中、初始化调制解调器、拨打电话号码和断开电话线路的用户程序。当电话线路已连接时，用户通信指令 TXD2 发送字符串“Connect”。



M8120 是初始化脉冲特殊内部继电器。

在电话线路连接后，MOV 指令存储 1 至 D8103 以启用用户协议。

MOV 指令设置拨号命令 ATD1234 **CR** **LF**。

“12” (3132h = 12594) → D8170

“34” (3334h = 13108) → D8171

“CR” (0D00h = 3328) → D8172 在电话号码末尾输入 **CR**。

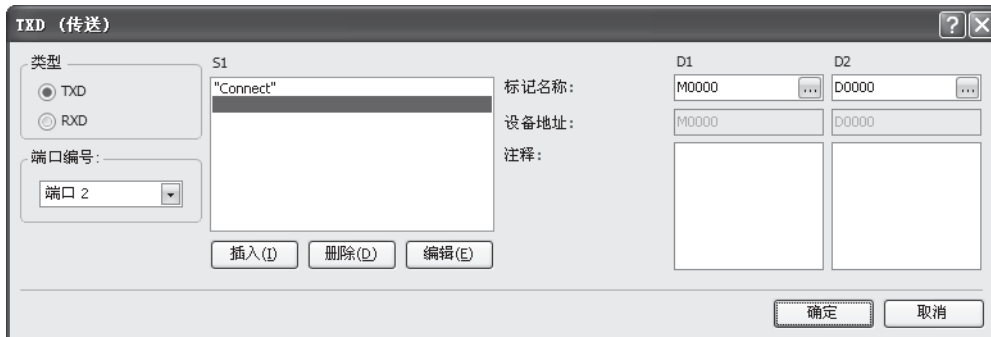
当输入 I0 打开时，将打开 M8050（初始化字符串）以发送初始化字符串、ATZ 和拨号命令至调制解调器。

在电话线路连接后，打开 M8077（线路连接状态）。

当 I1 打开时，TXD2 发送七个字符“Connect”。请参阅如下所示的“WindLDR”对话框。

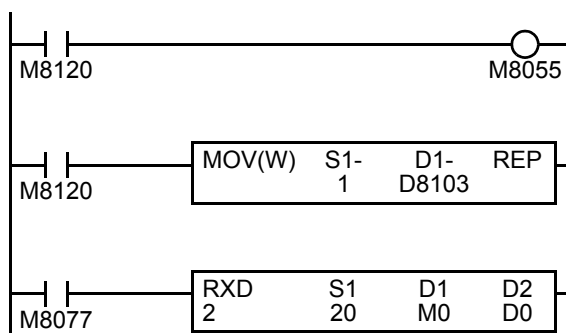
当输入 I2 打开时，打开 M8053（断开线路）以断开电话线路。

在调制解调器初始模式下，使用带有如下参数的 WindLDR 编写示例程序中的 TXD2 指令：



## 调制解调器应答模式示例程序

该程序演示用在调制解调器应答模式下，移动某值至数据寄存器并初始化调制解调器的用户程序。当电话线路连接后，执行用户通信指令 RXD2 以接收输入通信。



M8120 是初始化脉冲特殊内部继电器。

当 MicroSmart 运行开始时，打开 M8055 以在调制解调器应答模式下发送初始化字符串。

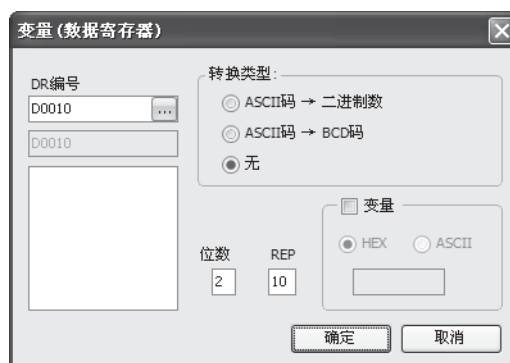
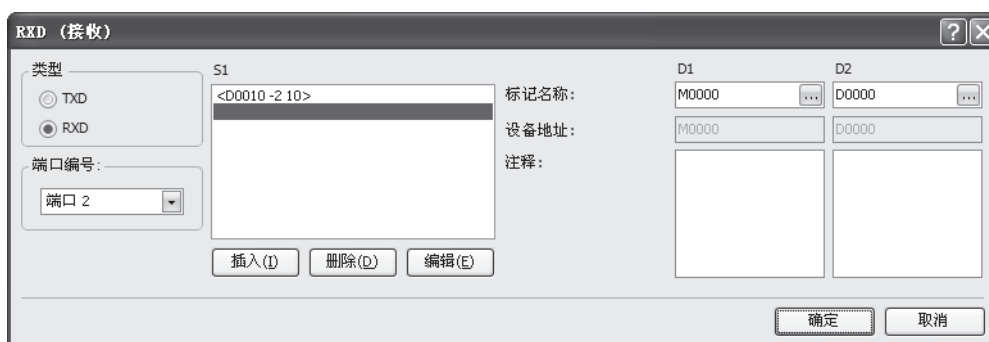
在电话线路连接后，MOV 指令存储 1 至 D8103 以启用用户协议。

在电话线路连接后，打开 M8077（线路连接状态）。

RXD2 接收输入通信，并将接收到的数据存储在以 D10 开始的数据寄存器中。

使用 WindLDR 以及以下所示参数编译 RXD2 指令：

源 S1: 数据寄存器 D10、不转换、2 位数、重复 10



### 调制解调器通信故障排除

当开始内部继电器打开时，**D8111**（调制解调器模式状态）的数据更改，但是调制解调器不工作。

**原因：** 使用了错误的电缆或接线不正确。

**解决方案：** 使用调制解调器电缆 1C（FC2A-KM1C）。

调制解调器上的 **DTR** 或 **ER** 指示灯未打开。

**原因：** 使用了错误的电缆或接线不正确。

**解决方案：** 使用调制解调器电缆 1C（FC2A-KM1C）。

当开始内部继电器打开时，**D8111**（调制解调器模式状态）的数据未更改。

**原因：** 未设置端口 2 的调制解调器协议。

**解决方案：** 使用 WindLDR 选择端口 2 的调制解调器协议（**设置 > 功能设置 > 通信端口**），然后将用户程序下载至 CPU 模块中。

当初始化字符串已发送时，出现失败，但是 **ATZ** 已成功发送。

**原因：** 初始化字符串对调制解调器无效。

**解决方案：** 请参阅调制解调器的用户手册修正初始化字符串。

当拨号命令已发送时，返回结果代码“**NO DIALTONE**”，电话线路未连接。

**原因 1：** 未连接模块电缆。

**解决方案 1：** 将模块电缆连接至调制解调器。

**原因 2：** 调制解调器用在 PBX 环境中。

**解决方案 2：** 将 X0 或 X3 添加至存储在数据寄存器 D8145 ~ D8169 中的初始化字符串中，然后重试初始化。

拨号成功完成，但是电话线路不久便断开。

**原因 1：** 线路两端的调制解调器设置不同。

**解决方案 1：** 对线路两端的调制解调器进行相同设置。

**原因 2：** 线路两端的调制解调器的型号不同。

**解决方案 2：** 在线路两端使用相同型号的调制解调器。

**原因 3：** 电话线路质量太差。

**解决方案 3：** 降低 MicroSmart 的通信速度至 9600 bps 以下。

# 23: MODBUS TCP 通信

---

---

## 简介

本章描述了 FC5A Micro-Smart CPU 模块的 Modbus TCP 主机 (client) 和从机 (server) 通信功能。

系统程序版本 210 或更高版本的所有 FC5A MicroSmart CPU 模块可以使用 Modbus TCP 通信，通过以太网发送和接收 Modbus 设备的数据。

## Modbus TCP 通信通用信息

当与网络服务器单元 (FC4A-SX5ES1E) 连接时，FC5A MicroSmart CPU 模块可以用作 Modbus TCP 通信的主机 (client) 或从机 (server)。

使用 Modbus TCP 主机通信，CPU 模块可以通过通信端口 2 对 Modbus 服务器设备中的数据进行更改或监控。

在 Modbus TCP 从机通信中，Modbus 主机设备可以通过通信端口 1 或端口 2 对 Micro-Smart CPU 模块中的设备值进行更改或监控。

有关 Modbus TCP 主机通信功能和构造，请参阅第 23-2 页；有关 Modbus TCP 从机通信通信功能和构造，请参阅第 23-5 页。

在系统程序版本 210 或更高版本的 FC5A MicroSmart CPU 模块和 WindLDR 版本 5.3 或更高版本中，才能使用 Modbus TCP 通信功能。使用 WindLDR 确认系统程序版本，详细信息请参阅第 13-1 页 (基本卷)。

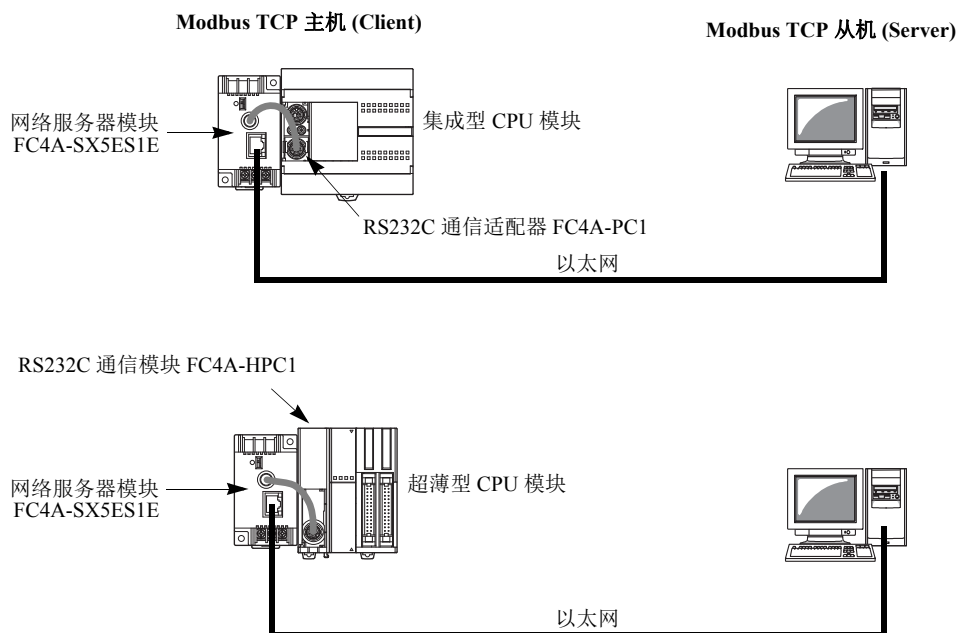
当 CPU 模块系统程序版本低于 210，请使用 WindLDR 版本 5.1 或更高版本下载最新系统程序。详细信息请参阅第 A-9 页 (基本卷)。

适用的 CPU 模块	所有 FC5A CPU 模块
系统程序版本	210 或更高
WindLDR 版本	5.3 或更高

### Modbus TCP 主机通信

Modbus TCP 主机通信的基本功能和规格，除仅能连接一个从机以外，与其他的 Modbus 主机通信相同。

WindLDR 功能设置用于设置 Modbus TCP 主机的通信设置和创建发送到 Modbus TCP 从机的请求。Modbus TCP 主机通信与用户程序执行进行异步处理。请求通过网络服务器单元 (FC4A-SX5ES1E) 发送到 Modbus TCP 从机。



注释：一个 Modbus TCP 主机只能连接一个 Modbus TCP 从机。

### Modbus TCP 主机通信规格

模式	Modbus TCP 主机通信
适用的通信端口	端口 2
通信速度 *a	9600, 19200, 38400, 57600 bps
数据长度 *a	8 位 (固定)
奇偶检验 *a	无、奇数、偶数
停止位 *a	1、2
从机编号	1 ~ 247 (0: 广播从机编号)
从机最大编号	1
接收超时* ^b	10 ~ 2550 ms (以 10 ms 为增量)
发送等待时间* ^c	1 ~ 5000 ms (以 1 ms 为增量)
重试次数	1 ~ 10

*a: 设置与网络服务器单元 (FC4A-SX5ES1E) 同样的值。

*b: 从从机接收一个响应帧之前，指定最大时间。

*c: D8054 是一个用于 Modbus 通信发送等待时间的特殊数据寄存器 (×1 ms)。使用 D8054 可以延迟从 MicroSmart 的发送。

## 使用 WindLDR 设置 Modbus TCP 主机通信

使用 WindLDR 功能设置可以编辑用于 Modbus 从机站的 Modbus TCP 主机通信设置和请求表。这些设置涉及到用户程序，因此这些设置更改后必须将用户程序下载到 MicroSmart。

1. 从 WindLDR 菜单栏选择**设置** > **功能设置** > **通信端口**。此时出现通信端口的“功能设置”对话框。
2. 在“端口 2”的“通信模式”下拉列表中，选择**Modbus TCP 主机**。



显示 Modbus TCP 主机请求表 (端口 2) 对话框。

也可以通过点击通信选项卡端口 2 的**设置**按钮打开 Modbus TCP 主机请求表 (端口 2) 对话框。



## 23: MODBUS TCP 通信

3. 点击**通信设置**按钮。  
显示通信设置对话框。请按需进行设置。



通信速度 (bps)*	9600, 19200, 38400, 57600
奇偶校验*	无、奇数、偶数
停止位*	1、2
重试次数	1 ~ 10
接收超时	1 ~ 255 (×10ms)

* 设置与网络服务器单元 (FC4A-SX5ES1E) 同样的值。

4. 点击**确定**，将返回 Modbus TCP 请求表 (端口 2)。在功能代码中指定请求。在一个请求表中最大可以输入 2040 个请求。  
选择使用请求执行设备和错误状态。当使用请求执行设备和错误状态，请输入设备的起始编号。



### 编辑请求表的注释：

请求执行内部继电器和错误状态数据寄存器按照请求编号顺序分配。当删除一个请求或更改请求的顺序时，更改了请求与请求执行内部继电器和错误状态数据寄存器的关系。如果在用户程序中使用了内部继电器或数据寄存器，必须相应地更改设备编号。在完成更改后，重新下载用户程序。

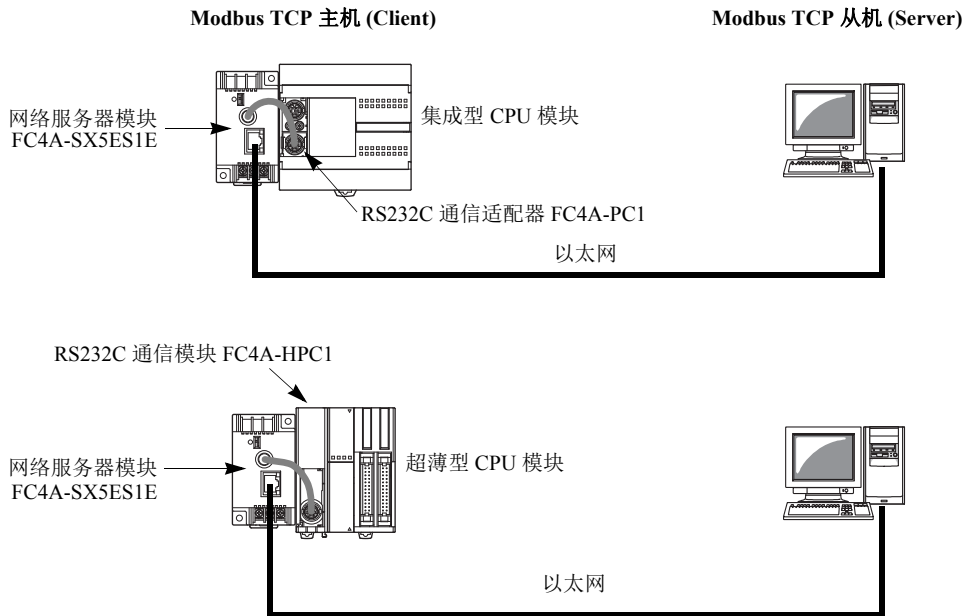
5. 当完成 Modbus TCP 主机请求表的编辑后，点击**确定**按钮保存更改。
6. 将用户程序下载到 CPU 模块。

完成编辑 Modbus TCP 主机的编辑。参数和有效值与 Modbus 主机通信相同。有关详细信息，请参阅第 12-6 页～第 12-7 页 (基本卷)。



## Modbus TCP 从机通信

使用 WindLDR 功能设置，通过 MicroSmart CPU 模块的通信端口 1 或端口 2 为 Modbus TCP 从机通信进行通信设置。在 Modbus TCP 通信中，当 Modbus TCP 从机从 Modbus TCP 主机接收到一个请求，Modbus TCP 从机根据请求读取或写入设备。请求处理至用户程序中的 END 处理。使用网络服务器单元 (FC4ASX5ES1E) 为 Modbus TCP 从机进行设置。



**注释：** 一个 Modbus TCP 主机只能连接一个 Modbus TCP 从机。

## 23: MODBUS TCP 通信

### Modbus TCP 从机通信规格

模式	Modbus TCP 主机通信	
	端口 1	端口 2
适用的通信端口	端口 1	端口 2
通信速度 *a	9600, 19200, 38400, 57600 bps	
数据长度 *a	8 位 (固定)	
奇偶检验 *a	无、奇数、偶数	
停止位 *a	1、2	
从机编号	1 ~ 247	
响应时间	1.5 ms	1 ~ 5000 ms (以 1 ms 为增量)*b
接收超时*c	10 ~ 2550 ms (以 10 ms 为增量)	
特殊内部继电器	-	M8005: 通信错误 M8080: 通信完成继电器
特殊数据寄存器	-	D8053: 错误代码 D8054: 发送等待时间 (以 1 ms 为增量)
适用的功能代码	01 读取线圈状态 02 读取输入状态 03 读取保持寄存器 04 读取输入寄存器 05 强制单线圈 06 预置单寄存器 15 强制多线圈 16 预置多位数据寄存器	

*a: 从 WindLDR > 功能设置 > 通信进行设置。

*b: Modbus 通信发送等待时间 (×1 ms) 的特殊数据寄存器 D8054 用于设置响应时间。0 时设置为 1ms, 5000 或以上时设置为 5000 ms。

*c: 当出现超时, MicroSmart 丢弃接收数据并等待下一个有效通信的第一帧。

### 通信完成和通信错误

当成功完成一个读取或写入处理, 或当出现一个通信错误时, 完成 Modbus TCP 通信。完成一个请求通信后, Modbus 通信完成继电器 M8080 将在 1 个扫描周期之内立即打开。当出现通信错误时, 错误代码将被存储到特殊寄存器 D8053 中。当 M8080 打开时, D8053 中的数据仅在 1 个扫描周期之内有效。

当出现通信错误时, 通信错误特殊内部继电器 M8005 也将在错误后的 1 个扫描周期之内立即打开。

## 使用 WindLDR 设置 Modbus TCP 从机通信

使用 WindLDR 功能设置可以编辑 Modbus TCP 从机 (server) 通信设置。这些设置涉及到用户程序，因此这些设置更改后必须将用户程序下载到 MicroSmart。

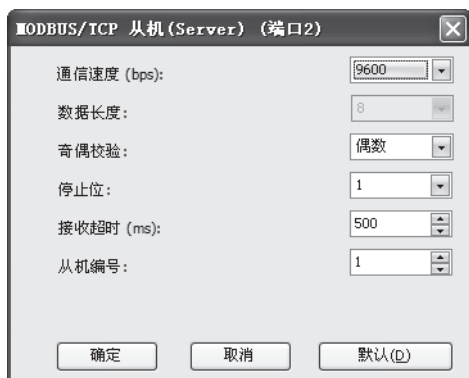
1. 从 WindLDR 菜单栏选择**设置 > 功能设置 > 通信端口**。此时出现通信端口的“功能设置”对话框。
2. 在“端口 1”或“端口 2”的“通信模式”下拉列表中，选择 **Modbus TCP 从机**。



显示 Modbus TCP 从机对话框。

也可以通过点击通信选项卡端口 1 或端口 2 的**设置**按钮打开 Modbus TCP 从机对话框。

3. 请按需更改通信设置。



通信速度 (bps)	9600, 19200, 38400, 57600
奇偶校验	无、奇数、偶数
停止位	1、2
接收超时	1 ~ 5000 ms
从机编号	1 ~ 247

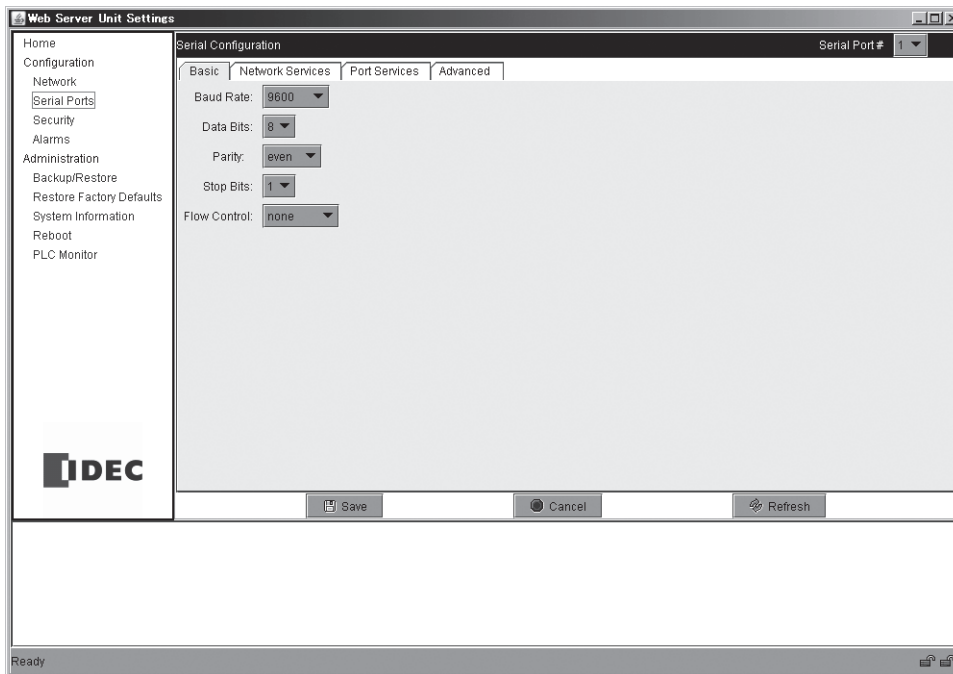
4. 点击**确认**按钮返回通信选项卡。
5. 点击**确认**按钮保存更改。
6. 将用户程序下载到 CPU 模块。

完成编辑 Modbus TCP 从机的编辑。参数和有效值与 Modbus 从机通信相同。有关详细信息，请参阅第 12-13 页 ~ 第 12-20 页 (基本卷)。

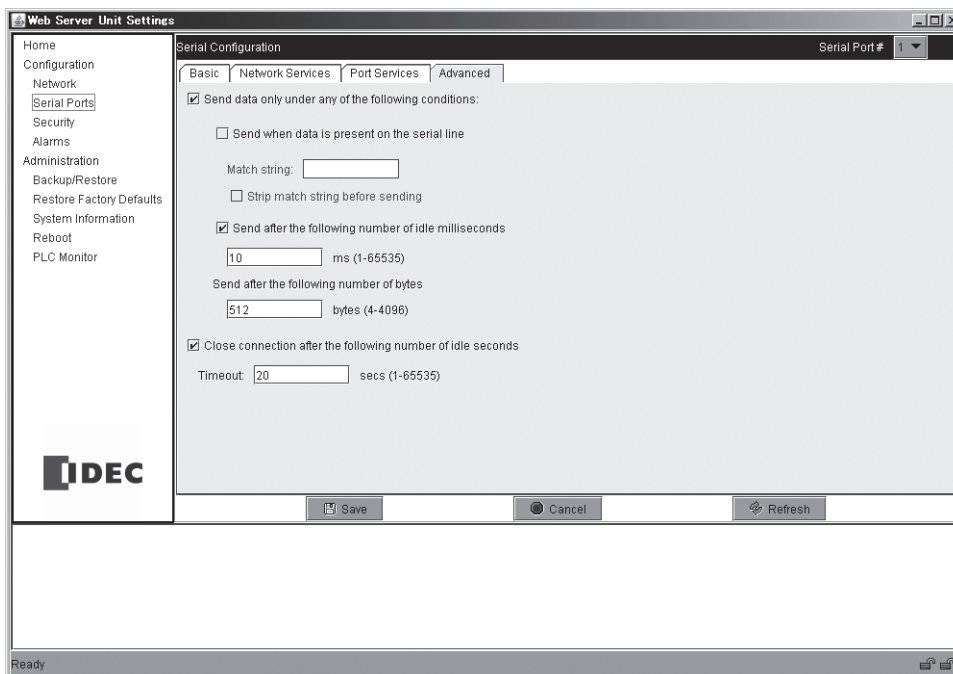
### 编辑网络服务器单元 (FC4A-SX5ES1E)

有关网络服务器单元的详细信息，请参阅网络服务器模块使用手册 (FC9Y-B921)。

1. 将网络服务器单元上的功能选择开关拨至 **USER**。
2. 在通信设置中设置相同的参数值。



3. 在高级选项卡，选中“Send after the following number of idle milliseconds.”并输入 10 ms 或更大的值。



## Modbus TCP 通信格式

本节描述了用于 Modbus TCP 主机和从机通信的通信格式。以 Modbus TCP 首部为 Modbus TCP 从机通信格式起始，后续除 CRC 和空闲 3.5 字符以外的 RTU 模式通信格式，如下所示。

### Modbus TCP 通信格式

发送 ID	协议 ID	信息长度 (字节)	单元 ID	功能代码	数据		
2 字节	2 字节	2 字节	1 字节	1 字节	N 字节		
Modbus TCP 首部							
RTU 模式 通信格式		空闲 3.5 字符	从机编号 1 字节	功能代码 1 字节	数据 N 字节	CRC 2 字节	空闲 3.5 字符

#### 发送 ID

Modbus TCP 从机 (server) 反馈来自主机 (client) 请求 ID。主机可以通过接收反馈请求 ID，确定是由哪个请求反馈的响应。当不需要确认时，发送 ID 设置为 0。

#### 协议 ID

将 0 设置为 Modbus TCP 协议。

#### 信息长度

设置后续信息的长度 (字节)。

#### 单元 ID

设置 Modbus TCP 从机地址 (1 ~ 247) 以确定单元。

#### 功能代码

设置功能代码，如 01 (读取线圈状态) 和 02 (读取输入状态)。详细信息请参阅第 23-6 页。

#### 数据

为每个功能设置要求数据。



# 24: AS-Interface 主机通信

## 简介

本章描述了关于 Actuator-Sensor-Interface(缩写为 AS-Interface) 的相关内容, 以及有关使用 AS-Interface 主机模块的详细信息。

## AS-Interface

AS-Interface 是一种控制总线, 其主要设计用途是为了控制传感器和致动器。AS-Interface 是一种与 IEC62026 标准兼容并且不归任何一个制造商私有的网络系统。通过使用在 AS-Interface 总线上传输的数字量和模拟量信号, 主机设备可以与从机设备 (例如传感器、致动器和远程 I/O) 通信。

AS-Interface 系统由以下三个主要组件组成:

- 一个主机, 例如 MicroSmart AS-Interface 主机模块 (FC4A-AS62M)
- 一个或多个从机设备, 例如传感器、致动器、开关和指示器
- 专用 30V DC AS-Interface 电源 (26.5 ~ 31.6V DC)

这些组件使用可同时进行数据传输和为 AS-Interface 供电的双芯电缆连接起来。AS-Interface 利用简单且有效的接线系统, 具有自动分配从机地址的功能, 同时它的安装和维护也非常容易。

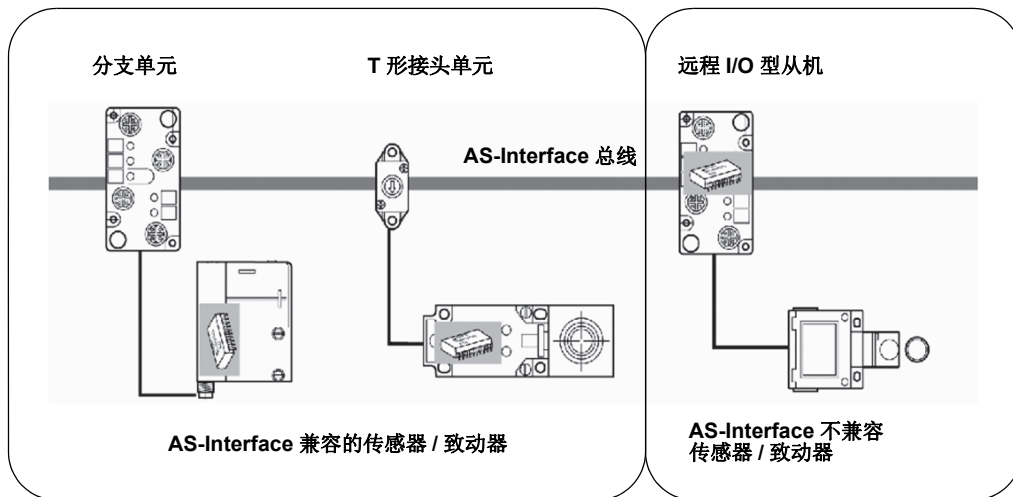
## AS-Interface 适用的传感器和致动器

### AS-Interface 兼容的传感器和致动器

AS-Interface 兼容的传感器和致动器使用内置的 AS-Interface 功能进行通信, 并在通过分支单元或 T 形接头单元直接与 AS-Interface 总线连接时作为 AS-Interface 从机使用。

### AS-Interface 不兼容的传感器 / 致动器

AS-Interface 不兼容的传感器和致动器也可以使用远程 I/O 从机连接到 AS-Interface 总线, 并可以跟 AS-Interface 兼容的设备一样操作。



使用一个或两个 AS-Interface 主机模块时的最大 I/O 点数

AS-Interface 主机模块	1 个模块	2 个模块
最大从机数	62 个从机	124 个从机
最大 I/O 点数	434 (248 点输入 / 186 点输出)	868 (496 点输入 / 372 点输出)
最大通信距离	没有转发器: 100m 有 2 转发器: 300m	没有转发器: 100m 有 2 转发器: 300m

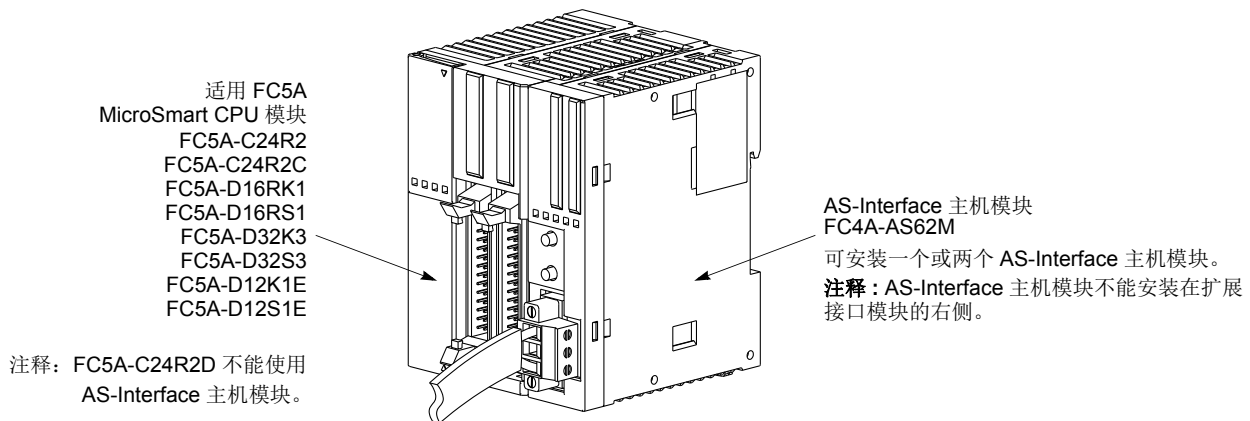
### AS-Interface 系统要求

#### 主机

AS-Interface 主机控制和监控连接到 AS-Interface 总线上的从机设备状态。

通常 AS-Interface 主机连接到 PLC (有时称为“主机”) 网关。例如 MicroSmart AS-Interface 主机模块连接到 MicroSmart CPU 模块。

FC5A MicroSmart CPU 模块可与一个或两个 AS-Interface 主机模块使用，因此可以建立两个独立的 AS-Interface 网络。



AS-Interface 主机模块最多可连接 62 个数字量 I/O 从机。AS-Interface 主机模块最多可连接七个模拟量 I/O 从机 (符合 AS-Interface 版本 2.1 和模拟量从机设置 7.3)。

#### 注意

- AS-Interface 主机模块不能连接到集成型 10-I/O 和 16-I/O 型 CPU 模块以及扩展接口模块。
- CPU 模块可连接一个或两个 AS-Interface 主机模块。如果连接了两个以上的 AS-Interface 主机模块，会发生错误并且特殊数据寄存器 D8037(扩展 I/O 模块数量) 存储错误代码 40(十六进制)。
- 通常，集成型 24-I/O 型 CPU 模块最多可以连接 4 个扩展接口模块。但在连接了一个或两个 AS-Interface 主机模块时，总共只能连接三个扩展模块，包括 AS-Interface 主机模块。由于产生热量，请勿连接三个以上的扩展模块。如果连接了三个以上的扩展模块，包括 AS-Interface 主机模块，会发生错误并且特殊数据寄存器 D8037(扩展 I/O 模块数量) 存储错误代码 20(十六进制)。
- 同样，超薄型 CPU 模块通常最多可以连接七个扩展 I/O 模块，但最多只能连接包括一个或两个 AS-Interface 主机模块的六个扩展模块。如果连接了六个以上的扩展模块，包括 AS-Interface 主机模块，会发生错误并且特殊数据寄存器 D8037(扩展 I/O 模块数量) 存储错误代码 20(十六进制)。
- AS-Interface 主机模块最多可连接七个模拟量 I/O 从机。当连接了七个以上的模拟量 I/O 从机时，AS-Interface 系统将不能正常工作。

#### 从机数

AS-Interface 总线上可以连接多种类型的从机设备，包括传感器、致动器和远程 I/O 设备。也可以连接模拟量从机处理模拟量数据。

可选用的从机有标准从机和 A/B 从机。标准从机有一个标准地址范围内 1 ~ 31 的地址。A/B 从机有一个标准地址范围内 1A ~ 31A 或扩展地址范围内 1B ~ 31B 的地址。在 A/B 从机中，具有一个 1A ~ 31A 地址的从机称为 A 从机，具有一个 1B ~ 31B 地址的从机称为 B 从机。



**AS-Interface 电源**

AS-Interface 总线使用专用的 30V DC 电源 (AS-Interface 电源)。并使用 AS-Interface 标记指示。通用电源单元不能用于 AS-Interface 总线。

当使用两个 AS-Interface 主机模块时，需要两个 AS-Interface 电源。因为 AS-Interface 电缆传输信号和电力，每个网络需要一个独立的电源。



AS-Interface 标记

**注意**

- 使用 VLSV( 超低安全电压 ) 给 AS-Interface 总线通电。  
AS-Interface 电源的标准输出电压是 30V DC。

**推荐的 IDEC AS-Interface 电源**

输入电压	输出电压	输出功率	型号
100 ~ 240V AC	30.5V DC	73W	PS2R-Q30ABL
		145W	PS2R-F30ABL

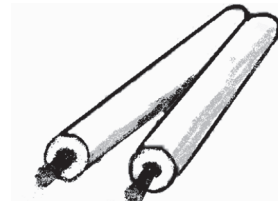
**电缆**

AS-Interface 总线只使用一条电缆传输信号和电力。使用下面其中一个电缆类型 ( 电线不需要多股绞合 )。

- 标准黄色无屏蔽 AS-Interface 电缆 ( 有极性 )
- 普通双芯扁平电缆



AS-Interface 电缆



扁形双芯电缆

**适用电缆规格**

电缆类型	电缆大小 / 制造商	横断面图
<b>AS-Interface 标准电缆</b>	电缆外皮颜色：黄色 导线截面：1.5 mm ² LAPP 电缆 型号：2170228 ( 外皮材料 EPDM ) 型号：2170230 ( 外皮材料 TPE )	
<b>扁形双芯电缆 或 单线 ( 请参阅注释 )</b>	导线截面 绞合线：0.5 ~ 1.0 mm ² 单芯线：0.75 ~ 1.5 mm ² AWG: 20 ~ 16	

**注释：**当使用单线时，最大电缆长度时 200mm。请参阅 第 24-1 页上的“最大通信距离”。

### 具有从机扩展能力的 AS-Interface V2 的主要功能

AS-Interface 是一个可靠的总线管理系统，其中的一个主机按照顺序周期性的监控连接到 AS-Interface 总线上的各个从机设备。除了从机地址外，主机还管理各个从机的 I/O 数据、参数和识别码。管理数据根据下面的从机类型：

#### 标准从机

- 每个从机最多有四个输入和四个输出。
- 设置从机动作模式的四个参数 (P3, P2, P1, P0)
- 四个识别码 (ID 码、I/O 码、ID2 码和 ID1 码)

#### A/B 从机

- 每个从机最多有四个输入和三个输出。
- 设置从机动作模式的三个参数 (P2, P1, P0)
- 四个识别码 (ID 码、I/O 码、ID2 码和 ID1 码)

**注释 1:** 参数 P3 ~ P0 用于设置从机的动作模式 有关详细信息，请参阅从机用户手册。

**注释 2:** 连接到 AS-Interface 总线上的从机根据各个从机中的 ID 码和 I/O 码相互区分。有些从机有 ID2 码和 ID1 码指示从机的内部功能。例如，模拟量从机使用 ID2 码表示从机频道数。

**注释 3:** MicroSmart AS-Interface 主机模块也和 AS-Interface 版本 2.1 及以前的从机兼容。

### 从机地址

连接到 AS-Interface 总线的每个标准从机可分配 1 ~ 31 中的一个地址，每个 A/B 从机可分配 1A ~ 31A 或 1B ~ 31B 中的一个地址。所有的从机的地址出厂设置为 0。从机的地址可以使用“编址工具”更改。使用 WindLDR 也可以更改连接到 AS-Interface 主机模块 1 和 2 的从机地址 ( 请参阅第 24-35 页 )。

当从机在运行中失效并且需要更换时，如果主机模块上启用了自动编址功能，只需要更换一个新的从机 ( 具有地址 0 和相同的识别码 )。拆卸的从机地址将自从分配给新的从机，您不需要重新设置地址。有关 ASI 命令启用自动寻址的详细信息，请参阅第 24-30 页。

### 从机识别

从机具有系列四个识别码。主机检查识别码确定连接到 AS-Interface 总线的从机类型和功能。

#### ID 码

ID 码由 4 位组成指示从机类型，例如传感器、致动器、标准从机或 A/B 从机。例如，一个标准远程 I/O 的 ID 码是 0，并且 A/B 从机的 ID 码是 A( 十六进制 )。

#### I/O 码

I/O 码由 4 位组成，指示从机上 I/O 点的数量和分配。

I/O 码	分配	I/O 码	分配	I/O 码	分配	I/O 码	分配
0h	I, I, I, I	4h	I, I, B, B	8h	O, O, O, O	Ch	O, O, B, B
1h	I, I, I, O	5h	I, O, O, O	9h	O, O, O, I	Dh	O, I, I, I
2h	I, I, I, B	6h	I, B, B, B	Ah	O, O, O, B	Eh	O, B, B, B
3h	I, I, O, O	7h	B, B, B, B	Bh	O, O, I, I	Fh	( 保留 )

I: 输入, 0: 输出 B: 输入和输出

#### ID2 码

ID2 码由 4 位组成，指示从机的内部功能。

#### ID1 码

ID2 码由 4 位组成，指示从机的附加识别。标准从机可有一个 0000 ~ 1111( 二进制 ) 的 ID1 码。A/B 从机使用 MSB 指示 A 或 B 从机，并且可有低三位的唯一值。A 从机的 MSB 设置为 0，B 从机的 MSB 设置为 1。

## 从机数量和 I/O 点数

可连接到一个 AS-Interface 主机模块的从机数量如下。

- 标准从机 : 最多 31 个
- A/B 从机 : 最多 62 个

当从机全部是标准从机或 A/B 从机时, 使用上面给出的从机数量限制。

当连接了 62 个 A/B 从机 (具有四个输入和三个输出) 时, AS-Interface 主机模块最多可控制 434 个 I/O 点 (248 个输入和 186 个输出)。

当标准从机和 A/B 从机一起混合使用时, 标准从机只能使用地址 1(A) ~ 31(A)。同样, 当标准从机使用了某个地址时, 同样编号的 B 地址不能用于 A/B 从机。

## AS-Interface 总线布局和最大长度

AS-Interface 总线布局灵活, 您可以根据需要自由安装总线。

当不使用中继器或扩展电路时, 最大总线长度可以是 100m(328ft.)。

FC4A-AS62M AS-Interface 主机模块可使用两个中继器将总线延长到 300m。

## AS-Interface 总线周期时间

AS-Interface 总线周期时间是主机循环总线上的各个从机需要的时间。

各个从机的信息按照顺序连续传输, 因此 AS-Interface 总线周期时间取决于可用从机的数量。

- 当达到 19 个可用从机时, 总线的周期时间是 3 ms。
- 当 20 ~ 62 个从机可用时, 总线周期时间是  $0.156 \times (1+N)$  ms, 其中 N 是从机数。

当 A 从机和 B 从机有相同的地址编号 (例如 12A 和 12B) 时, 每个周期两个从机交替更新。所以, 当系统包括 31 个 A 从机和 31 个 B 从机时, AS-Interface 总线周期时间将是 10 ms。

## 最大 AS-Interface 总线周期时间

- 当连接了 31 个从机时, 最大总线周期时间是 5 ms。
- 当连接了 62 个从机时, 最大总线周期时间是 10 ms。

## 高可靠性和安全性

AS-Interface 使用高可靠性和安全性的传输进程。主机监控总线上传输的 AS-Interface 电源电压和数据, 并且监控从机故障和数据错误。

在运行过程中如果更换了一个从机或添加了一个新从机, AS-Interface 主机模块不需要关闭并且可继续不间断地与总线上其他可用的从机进行通信。

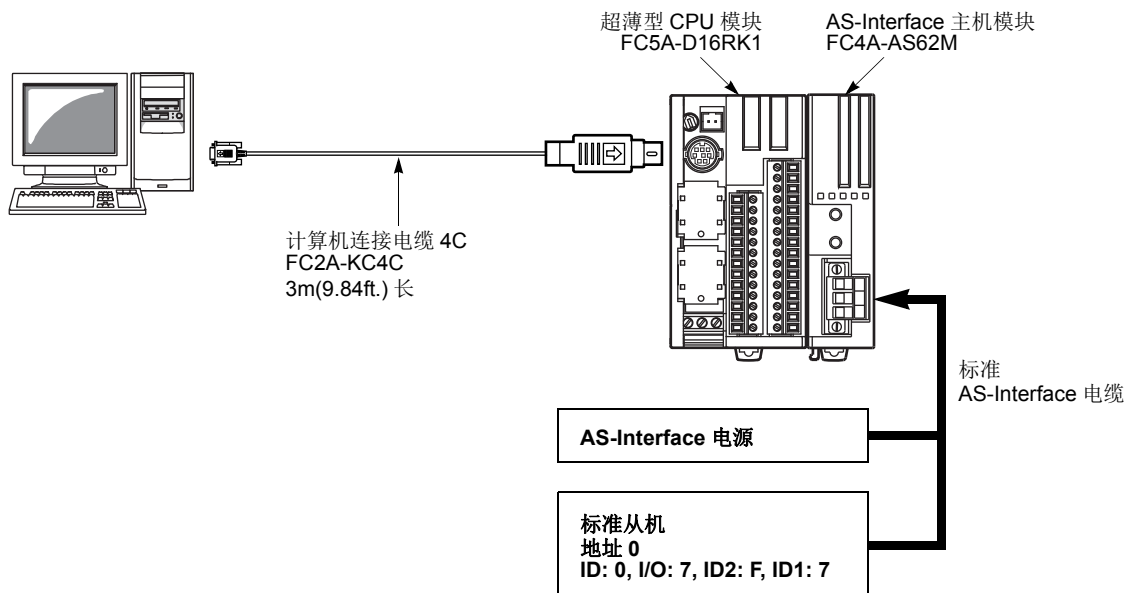
### 基本操作

本章说明在计算机上编写 WindLDR 监控从机运行的基本 AS-Interface 系统的简单操作步骤。

### AS-Interface 系统安装

示例 AS-Interface 包括下列设备：

名称	型号	说明
FC5A MicroSmart 超薄型 CPU 模块	FC5A-D16RK1	—
MicroSmart AS-Interface 主机模块	FC4A-AS62M	—
WindLDR	SW1A-W1C	5.0 或更高版本
AS-Interface 标准从机	—	1 单元 地址 0 ID: 0, I/O: 7, ID2: F, ID1: 7
AS-Interface 电源	PS2R-Q30ABL	输出 30.5V DC, 2.4A (73W)

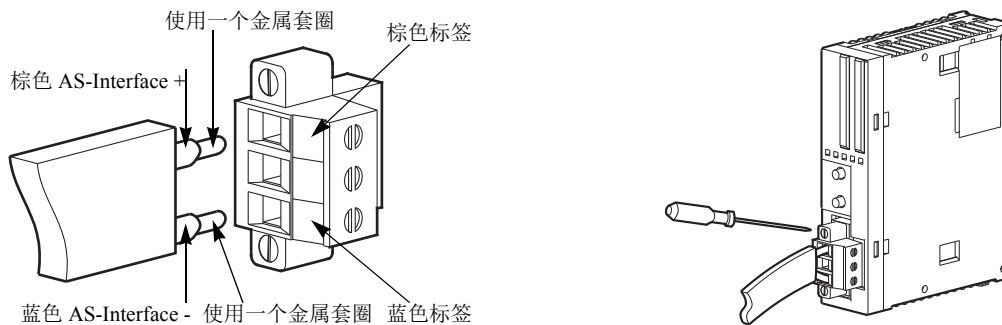


### AS-Interface 电缆接线


在进行 AS-Interface 电缆接线之前，从 AS-Interface 主机模块 AS-Interface 电缆连接器上拆下 AS-Interface 电缆端子台。

AS-Interface 规定 AS-Interface + 线使用棕色电缆，AS-Interface - 线使用蓝色电缆。连接电缆使其与端子台上的颜色标签匹配。拧紧端子螺丝到 0.5 ~ 0.6 N·m 的扭矩。

将端子台插入 AS-Interface 主机模块上的连接器，然后将安装螺丝拧紧到 0.3 ~ 0.5 N·m 的扭矩。



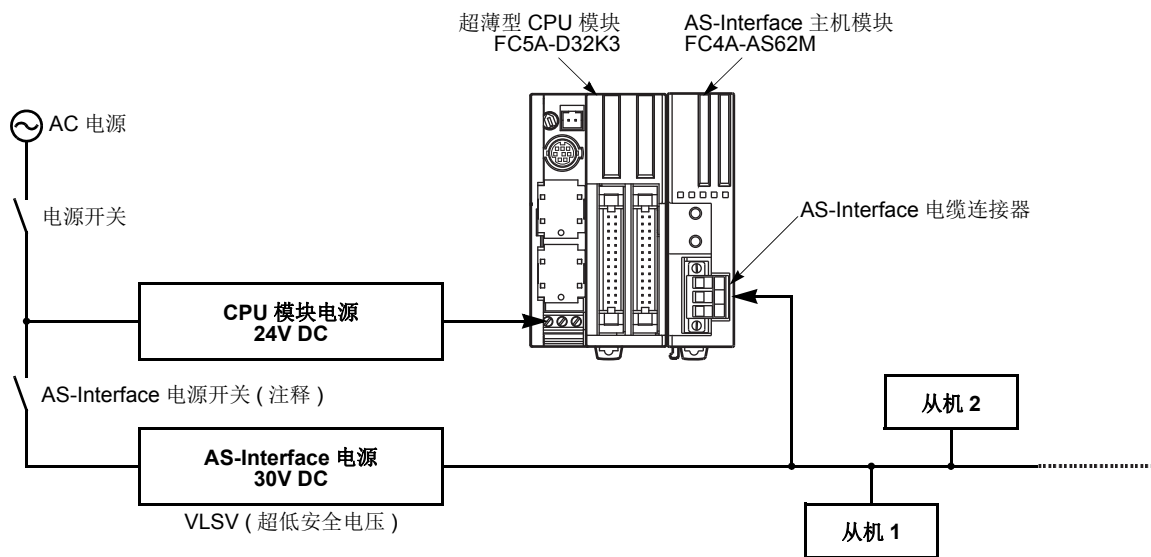
## 电源

 注意

- 当关闭 CPU 模块的电源时，也要关闭 AS-Interface 电源。如果 CPU 模块停止供电，然后恢复供电，而 AS-Interface 保持一直供电，AS-Interface 通信可能由于设置错误而停止，导致通信错误。
- 打开 AS-Interface 电源不应迟于打开 CPU 模块电源，当网络上存在从机地址 0 时除外。两个电源可按照任意顺序关闭。
- 刚打开电源时，CPU 模块不能访问 AS-Interface 主机模块中的从机 I/O 数据。编写用户程序，以便打开特殊内部继电器 M1945(Normal_Operation_Active) 后访问从机 I/O 数据。请参阅第 24-25 页。

## 电源接线图

推荐的电源接线图如下所示：给 CPU 模块电源和 AS-Interface 电源使用一个共用电源开关，确保两个电源同时打开和关闭。



**注释：**不关闭 CPU 模块和 AS-Interface 线的电源就可以使用带有地址 0 的新从机更换故障从机。但是，如果在更换从机前关闭了电源，安装带有地址 0 的新从机并执行下面其中的一个步骤，因为需要初始化 AS-Interface 主机模块来启用通信。

- 断开 AS-Interface 电缆连接器，并且打开两个电源。五秒钟后，连接 AS-Interface 电缆连接器。
- 首先打开 CPU 模块电源。五秒钟后，打开 AS-Interface 电源。

### 选择 PLC 类型

在计算机上启动 WindLDR。

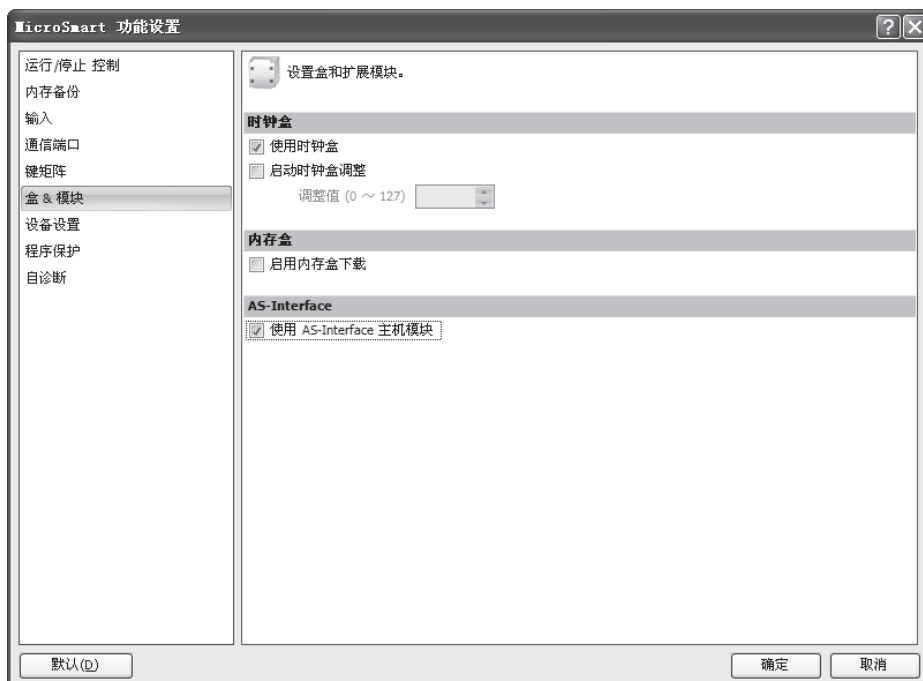
1. 从 WindLDR 菜单栏中，选择**设置 > PLC > PLC 类型**。出现“PLC 选择”对话框。
2. 选择 **FC5A-D16RX1**。
3. 单击**确定**保存更改并返回梯形图编辑屏幕。



### 功能设置

必须在“功能设置”对话框中选择使用 AS-Interface 主机模块。

1. 从 WindLDR 菜单栏中，选择**设置 > 功能设置 > 盒 & 模块**。此时出现“盒 & 模块”的“功能设置”对话框。



2. 确保选中**使用 AS-Interface 主机模块**左侧的复选框。

默认设置选择此复选框。如果在 PLC 选择项下选择 FC5A-D12K1E 或 FC5A-D12S1E，则不会显示复选框，因为此设置永久启用。因为此设置与用户程序有关，更改任何设置后将用户程序下载到 CPU 模块。

当连接了 AS-Interface 主机模块时，如果 CPU 模块上的 ERR LED 点亮，进行以上设置后将用户程序下载到 CPU 模块。

## 分配一个从机地址

出厂时，AS-Interface 兼容的从机设备设置为地址 0。如第 24-6 页所示，将从机连接到 AS-Interface 主机模块。请勿连接两个以上从机地址为 0 的从机，否则 AS-Interface 主机模块不能正确识别从机地址。

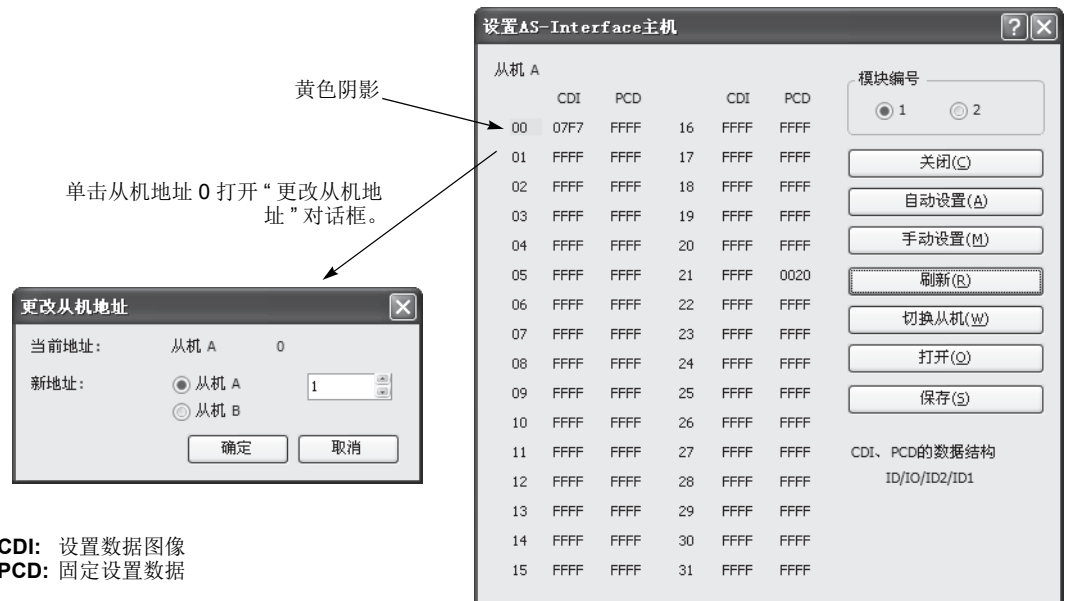
1. 首先给 MicroSmart CPU 模块通电。大约五秒钟后，打开 AS-Interface 电源。

**注释：**当 AS-Interface 总线上没有安装从机地址 0 时，可同时打开 CPU 模块电源和 AS-Interface 电源。请参阅第 24-7 页。

2. 请在 WindLDR 菜单栏中选择 **联机 > AS-Interface > 主机设置**。来打开“设置 AS-Interface 主机”对话框。按 **刷新 (R)** 收集从机信息并更新屏幕显示。(当完成主机模块中的设置时，您不需要按 **刷新 (R)**，因为屏幕显示自动更新)。

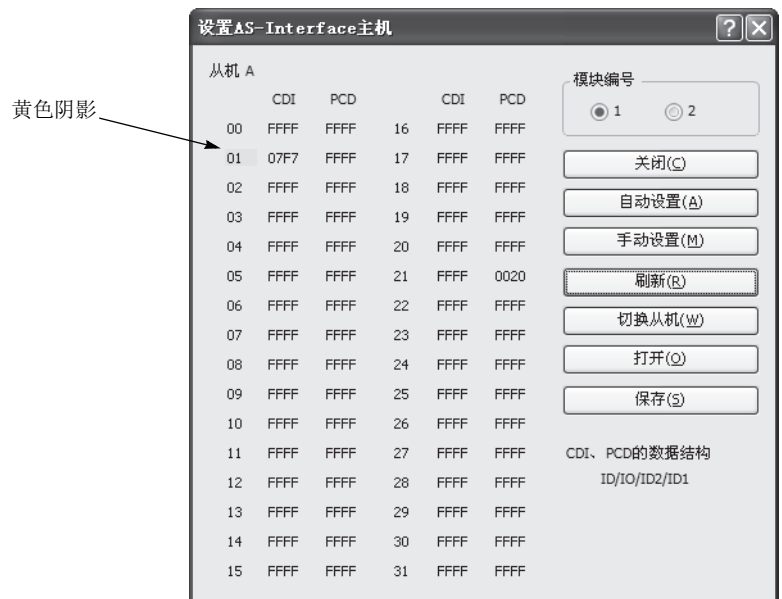
在“设置 AS-Interface 主机”对话框上，从机地址零呈黄色。这表示主机模块已经发现 AS-Interface 总线上的从机地址 0。从机地址 0 的 CDI 显示 07F7(ID: 0, I/O: 7, ID2: F, ID1: 7)。

3. 单击从机地址“00”打开从机 0 的“更改从机地址”对话框。要给从机分配从机地址 1，在新地址字段键入 1 并单击 **确定**。



新地址“01”呈黄色，表示地址分配已完成。

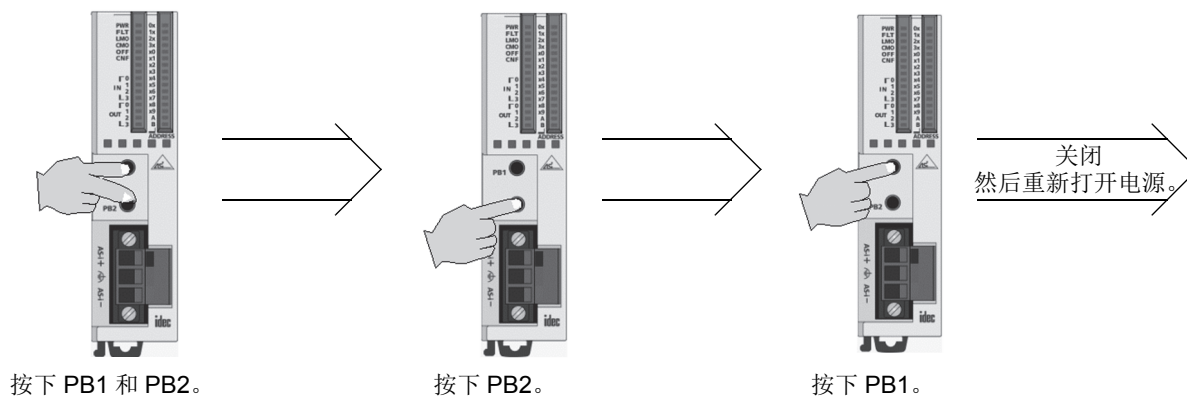
4. 当更改其他从机上的从机地址时，如果可以不用关闭电源给从机接线，请从步骤 3 继续，如果 CPU 模块电源关闭，从步骤 1 继续。



### 设置从机

然后，您需要使用 AS-Interface 主机模块上的按钮 PB1 和 PB2 或 WindLDR 在 AS-Interface 主机模块中设置从机设置。

#### 使用按钮 PB1 和 PB2 设置



1. 检查 AS-Interface 主机模块上的 PWR LED 和 CMO LED 点亮 (标准保护模式)。
2. 同时按下按钮 PB1 和 PB2 三秒钟 CMO LED 关闭，并且 LMO LED 点亮 (保护模式)。
3. 按下按钮 PB2 三秒钟 CNF LED 闪烁 (设置模式)。
4. 约 5 秒钟后，按下按钮 PB2 三秒钟。所有 I/O LED 闪烁一次完成设置。
5. 关闭 CPU 模块和 AS-Interface 主机模块，然后重新通电。检查 FLT LED 关闭，这表示设置完成。
6. 使用 WindLDR 查看“设置 AS-Interface 主机”对话框上的从机信息，并检查正确识别所有从机信息。



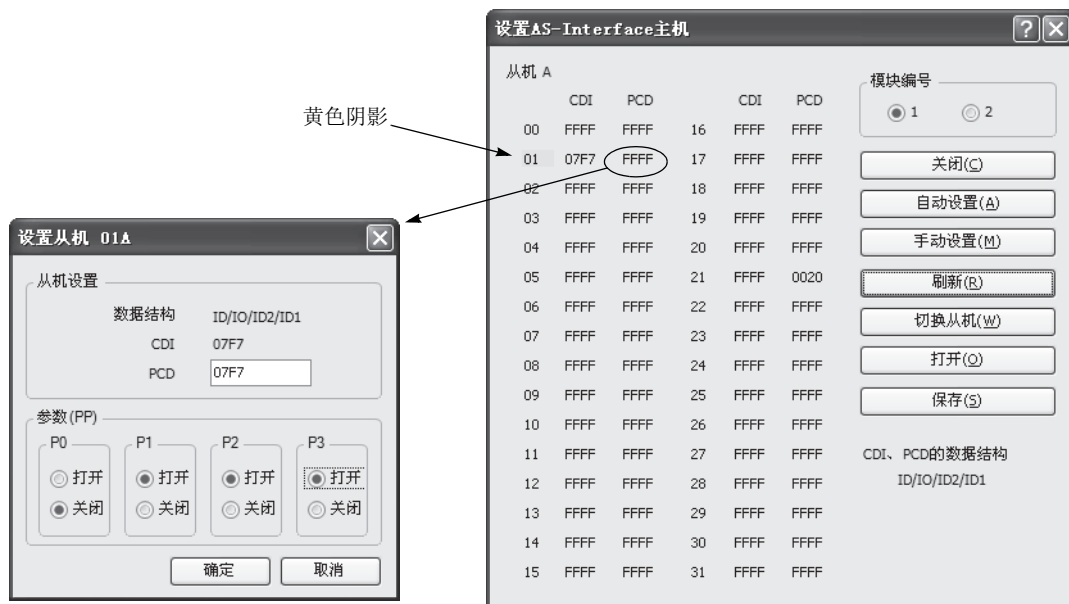
## 使用 WindLDR 设置

有两种方式使用 WindLDR 进行从机设置。使用“设置 AS-Interface 主机”对话框上的**自动设置**或**手动设置**。

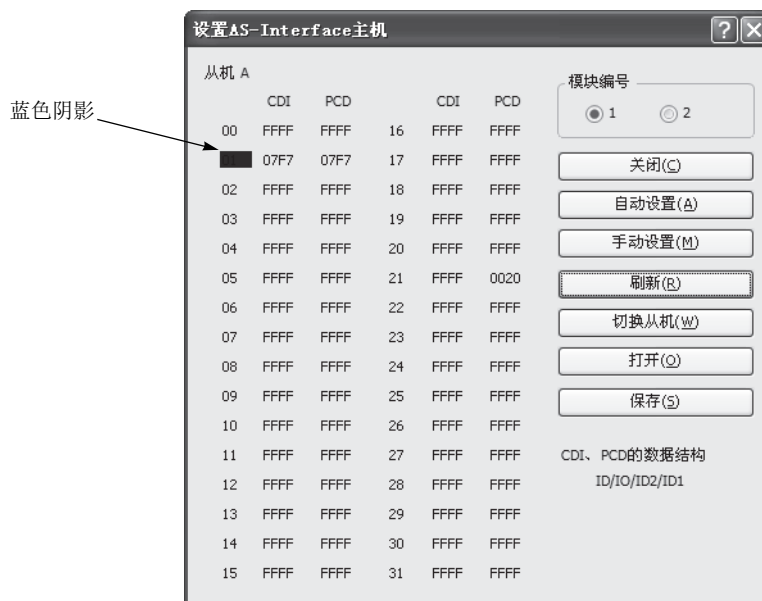
1. 单击**自动设置**按钮，将连接从机的设置信息 (LDS, CDI, PI) 存储到 AS-Interface 主机模块中的 EEPROM。有关详细信息，请参阅第 24-36 页。

自动设置把 AS-Interface 总线上发现的从机信息自动存储到 AS-Interface 主机模块中的 EEPROM，并完成设置。另一种设置方法是下面的手动设置。

2. 单击从机地址 01 的 PCD 值“FFFF”打开“设置从机 01A”对话框。
3. 输入与 PCD 字段中 CDI“07F7”相同的值。(将所有未使用的从机 PCD 值设置为 FFFF。)
4. 若有必要，选择参数 (PP)P0 ~ P3 的初始设置。



5. 单击**手动设置**按钮将选择的 PCD 和参数值存储到主机模块。
6. 检查从机地址 01 出现蓝色阴影，现在设置完成。



## 24: AS-INTERFACE 主机通信

### 监控数字量 I/O、更改输出状态和参数

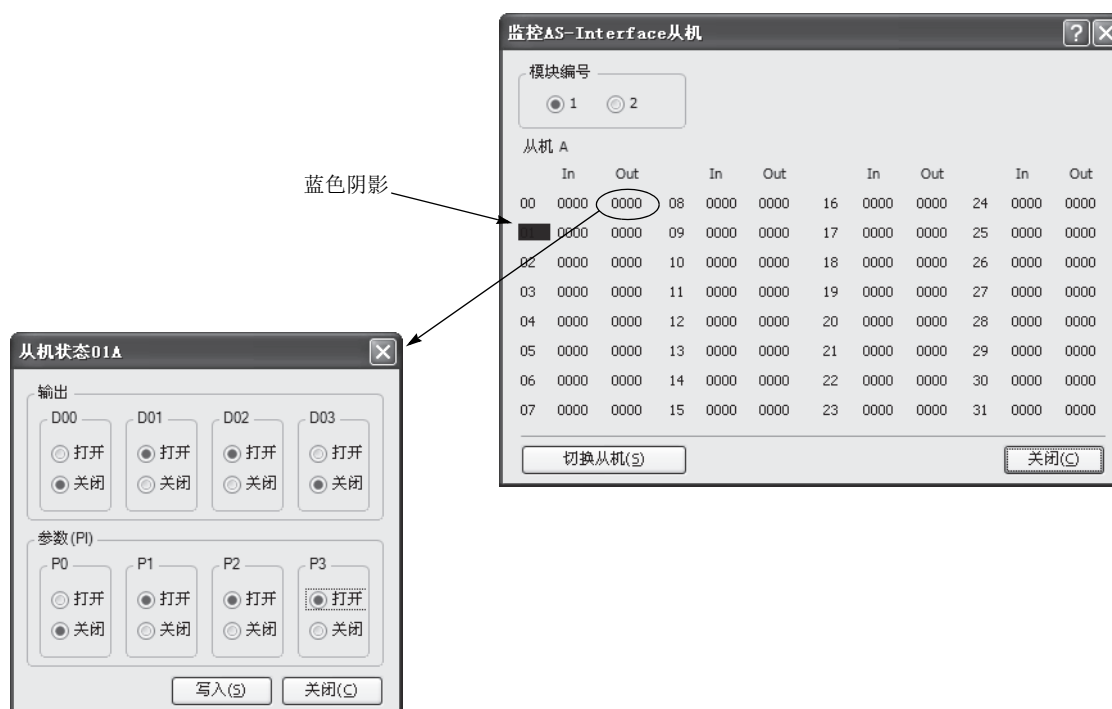
在 MicroSmart 通过 AS-Interface 总线与 AS-Interface 从机通信时，可使用计算机上的 WindLDR 监控 AS-Interface 从机的运行状态。连接到 AS-Interface 主机模块的从机输出状态和参数图像 (PI) 也可以使用 WindLDR 更改。

1. 从 WindLDR 菜单栏中，选择**联机 > 监控 > 监控**。从 WindLDR 菜单栏中，选择**联机 > AS-Interface > 从机监控**。“监控 AS-Interface 从机”对话框出现。

可用从机用蓝色阴影指示。

下一步更改可用从机的输出状态。

2. 单击从机地址 01 的输出打开“从机状态 01A”对话框。
3. 根据需要单击打开或关闭按钮更改输出 00 ~ 03 和参数 P0 ~ P3 的状态。



选择的参数 (PI) 在 CPU 模块关闭前一直有效。当 CPU 模块再次通电时，从机设置步骤 (第 24-10 页) 中选择的参数值 (PP) 将生效。要把更改的参数值存储到 AS-Interface 主机模块 EEPROM，通过把 0306、0100、0000、0000、0001 存储到数据寄存器 D1941 ~ D1945 执行复制 PI 到 PP 命令。请参阅第 24-30 页。

## 系统启动时的故障

下表概述了系统启动时可能发生的故障，可能原因以及采取的措施。

故障	原因和措施
PWR LED 关闭。 (电源)	<ul style="list-style-type: none"> <li>AS-Interface 主机模块的 AS-Interface 电源没有供电。检查接线正确并且 AS-Interface 电源已经供电。</li> <li>没有从 CPU 模块向 AS-Interface 主机模块供电。检查 CPU 模块与 AS-Interface 主机模块之间的连接。</li> </ul>
FLT LED 点亮。 (故障)	<ul style="list-style-type: none"> <li>总线上的从机设置不正确。使用 WindLDR 从机监控功能检查从机正确连接。若有必要，进行设置。有关设置方法，请参阅第 24-34 页。</li> </ul> <p>尽管从机正确连接并且设置完成，如果 FLT LED 保持点亮，请断开然后重新连接 AS-Interface 连接器，或者关闭然后打开 AS-Interface 电源。</p>
LMO LED 点亮。 (本地模式)	<p>CPU 模块不能与 AS-Interface 主机模块进行通信。检查以下几点。</p> <ul style="list-style-type: none"> <li>CPU 模块是否与 AS-Interface 兼容？检查 CPU 模块的型号</li> <li>在 WindLDR“功能设置”中的“使用 AS-Interface 主机模块”复选框是否有选中标记？默认设置选择此复选框。如果没有，请选中复选框并将用户程序下载到 CPU 模块。</li> </ul>
OFF LED 点亮。 (脱机)	<ul style="list-style-type: none"> <li>在连接地址 0 的从机时电源打开。更改从机地址后重新打开电源。有关地址更改方法，请参阅第 24-35 页。</li> </ul>
从机运行不稳定。	<ul style="list-style-type: none"> <li>检查是否有两个以上地址相同的从机？每个从机必须有唯一的地址。如果两个从机具有相同的地址和相同的识别码 (ID, I/O, ID2, ID1)，AS-Interface 主机模块会检测不到错误。当使用 WindLDR 更改重复从机地址时，从总线上拆下一个从机。</li> </ul>

## 按钮和 LED 指示灯

这一节描述操作 AS-Interface 主机模块上的按钮 PB1 和 PB2 更改动作模式，也说明了地址和 I/O LED 指示灯的功能。

### 按钮操作

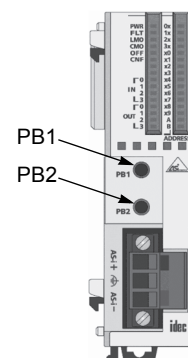
通过 AS-Interface 主机模块前部的按钮 PB1 和 PB2 进行的操作取决于按下的时间。“长按”切换动作模式，而“短按”切换 I/O LED 上监控的从机。如果按下 PB1 和 PB2 的时间与这些都不相符，AS-Interface 主机模块状态不会改变。

#### 长按

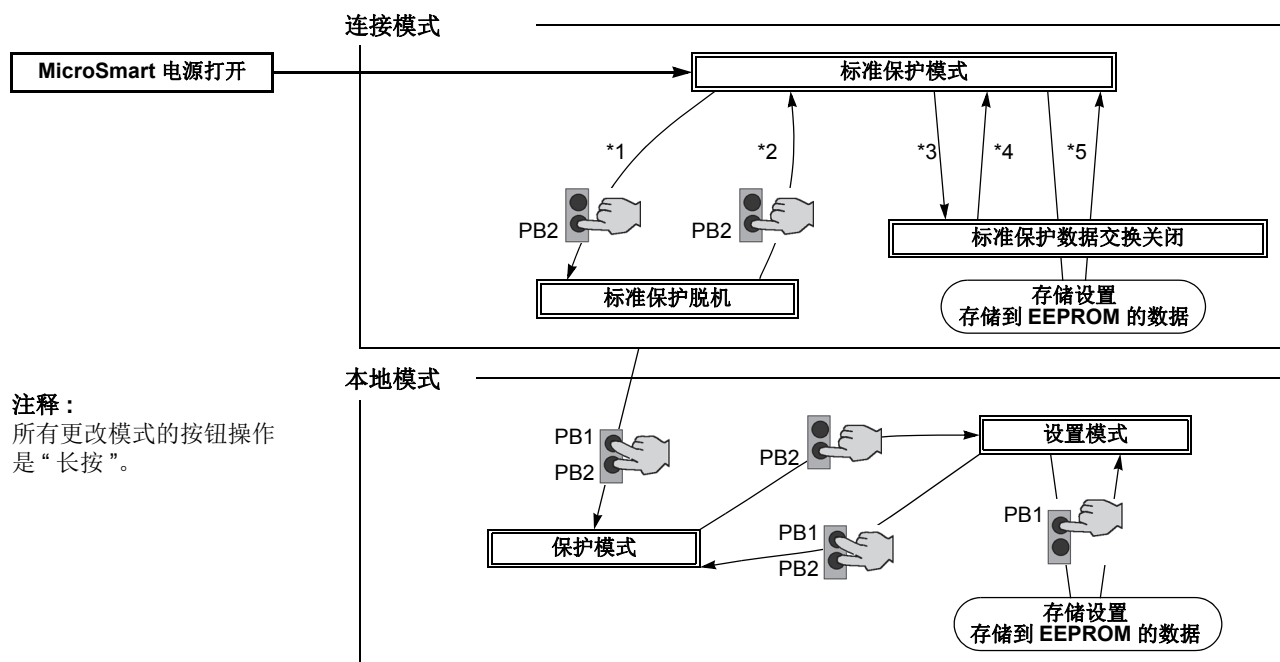
当您按下 PB1 或 PB2 其中的一个按钮或者两个都按下超过 3 秒钟时，“长按”生效。使用长按更改 AS-Interface 主机模块的动作模式，或将设置数据保存到 AS-Interface 主机模块 EEPROM。

#### 短按

当您按下 PB1 或 PB2 其中的一个按钮或者两个都按下不超过 0.5 秒钟时，“短按”生效。在 AS-Interface 主机模块 LED 指示灯上监控从机 I/O 状态时，使用短按更改从机地址。



### 使用按钮转换 AS-Interface 主机模块模式



注释：  
所有更改模式的按钮操作是“长按”。

- *1 按钮操作或执行 ASI 命令“转至标准保护脱机”。
- *2 按钮操作或执行 ASI 命令“转至标准保护模式”。
- *3 执行 ASI 命令“禁用数据交换”。
- *4 执行 ASI 命令“启用数据交换”。
- *5 通过单击 WindLDR 中的“自动设置”或“手动设置”按钮完成设置。设置数据保存到 AS-Interface 主机模块 EEPROM。

## AS-Interface 主机模块动作模式

AS-Interface 主机模块有两个操作模式：连接模式用于实际操作，本地模式模式用于维护。

### 连接模式

在连接模式中，CPU 模块与 AS-Interface 主机模块进行通信来监控和控制各个从机。连接模式又以下三种模式构成。

- **标准保护模式**

当 CPU 模块通电时，如果没有错误发生，AS-Interface 主机模块最初进入连接模式的标准保护模式。这是 AS-Interface 主机模块与连接的从机进行数据通信的标准动作模式。

如果存储在 AS-Interface 主机模块中的设置数据与当前连接的从机设置不匹配，AS-Interface 主机模块前部的 FLT LED 点亮。使用 AS-Interface 主机模块上的按钮进行设置。也可以使用 WindLDR 进行设置。请参阅第 24-36 页。

- **标准保护脱机**

AS-Interface 主机模块停止与所有从机的通信并启用脱机运行（主机模块初始化）。在这个模式中，CPU 模块不能监控从机状态。

要从标准保护模式进入标准保护脱机，长按 PB2 按钮或执行 ASI 命令“转至标准保护脱机”。要返回标准保护模式并恢复数据通信，再次长按 PB2 按钮或执行 ASI 命令“转至标准保护模式”。有关 ASI 命令的详细信息，请参阅第 24-30 页。

- **标准保护数据交换关闭**

禁用与所有从机的数据通信。要进入此模式，请执行 ASI 命令“禁用数据通信”。要返回标准保护模式并恢复数据通信，请执行 ASI 命令“启用数据交换”。有关 ASI 命令的详细信息，请参阅第 24-30 页。

在 WindLDR 上进行自动设置或手动设置时，AS-Interface 主机模块在设置过程中进入此模式。

### 本地模式

在本地模式中，CPU 模块不与 AS-Interface 主机模块进行通信。本地模式用于进行维护操作，例如检查设置和从机输入。在操作过程中使用输入 LED 检查从机输入数据。

当 CPU 模块通电时，如果没有错误发生，AS-Interface 主机模块最初进入连接模式的标准保护模式。要从任何连接模式切换到本地模式（保护模式），请同时长按 PB1 和 PB2 按钮。使用按钮不能从本地模式切换回连接模式。要返回连接模式，关闭 CPU 模块然后重新通电。

本地模式由两个模式构成：保护模式和设置模式。

- **保护模式**

此模式根据存储在 AS-Interface 主机模块中的从机设置数据操作从机。如果存储在 AS-Interface 主机模块中的设置数据与当前连接的从机设置不匹配，AS-Interface 主机模块前部的 FLT LED 点亮，并且从机不能正常运行。

要从任何连接模式进入保护模式，请同时长按 PB1 和 PB2 按钮。

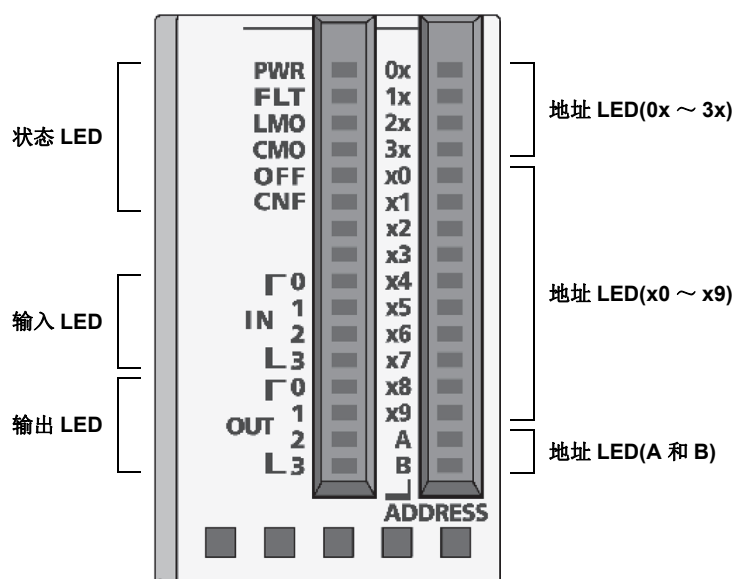
- **设置模式**

不论 AS-Interface 主机模块中存储的从机设置如何，此模式能够把所有当前连接的从机切换到可用状态。要把当前从机设置数据存储到 AS-Interface 主机模块 EEPROM，请长按 PB1 按钮。这样进行设置。

要从保护模式进入设置模式，请长按 PB2 按钮。要返回保护模式，同时长按 PB1 和 PB2 按钮。

### LED 指示灯

AS-Interface 主机模块上的 LED 指示灯包括状态 LED、I/O LED 和地址 LED。



LED 指示灯	说明
<b>状态 LED</b> <b>PWR</b> (AS-Interface 电源)  <b>FLT</b> (故障)  <b>LMO</b> (本地模式)  <b>CMO</b> (连接模式)  <b>OFF</b> (脱机)  <b>CNF</b> (设置)	指示 AS-Interface 主机模块的 AS-Interface 电源状态。 当 AS-Interface 供电充足时点亮。
	指示 AS-Interface 设置状态。 当存储在 AS-Interface 主机模块 EEPROM 中的固定设置数据 (PCD) 与当前从机设置或设置数据图像 (CDI) 不匹配时点亮。这时，没有完成设置或在 AS-Interface 总线发现错误。
	指示 AS-Interface 主机模块的模式。 当 AS-Interface 主机模块在本地模式中时点亮。 当 AS-Interface 主机模块在连接模式中时关闭。
	指示 AS-Interface 主机模块的模式。 当 AS-Interface 主机模块在连接模式中时点亮。 当 AS-Interface 主机模块在本地模式中时关闭。
	指示 AS-Interface 主机模块的运行状态。 当 AS-Interface 主机模块在标准保护脱机时点亮。
	指示 AS-Interface 主机模块的设置状态。 当 AS-Interface 主机模块在设置模式中时点亮。
<b>输入 LED</b> <b>IN0-IN3</b>	在地址 LED 指示的地址指示四个输入的运行状态。 在指示的地址相应的输入打开时点亮。
<b>输出 LED</b> <b>OUT0-OUT3</b>	在地址 LED 指示的地址指示四个输出的运行状态。 在指示的地址相应的输出打开时点亮。
<b>地址 LED</b> <b>0x-3x</b> (位置 10) <b>x0-x9</b> (位置 1) <b>A, B</b> (A 或 B 从机)	指示 0A ~ 31B 的从机地址。 当选择的地址存在时点亮。 当选择的地址不存在时闪烁。

## 状态 LED

AS-Interface 主机模块的动作模式可以通过按 AS-Interface 主机模块前部的按钮或执行 ASI 命令进行更改。动作模式可以通过 AS-Interface 主机模块上的六个状态 LED 确认。有关 ASI 命令的详细信息，请参阅第 24-30 页。

### 状态 LED 指示

状态 LED		PWR	FLT	LMO	CMO	OFF	CNF
连接模式	标准保护模式	开 ^{*1}	关 ^{*2}	关	开	关	关
	标准保护脱机	开 ^{*1}	开	关	开	开	关
	标准保护 数据交换关闭	开 ^{*1}	开	关	开	关	关
本地模式	保护模式	开 ^{*1}	关 ^{*2}	开	关	关	关
	设置模式	开 ^{*1}	关 ^{*2}	开	关	关	闪烁

*1: 当 AS-Interface 没有供电时关闭。

*2: 当 AS-Interface 总线上发现错误时点亮。

## 地址 LED 和 I/O LED

可在 AS-Interface 主机模块前部的地址 LED 和 I/O LED 上监控各个从机的运行状态和 I/O 状态。

### 从机运行状态

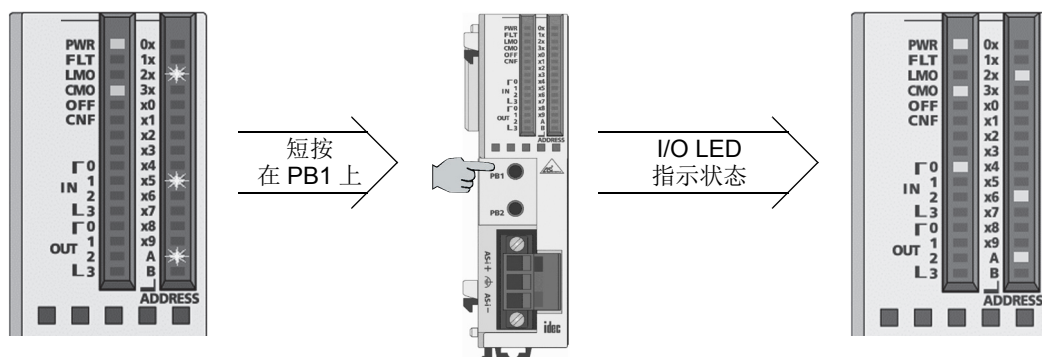
可通过查看地址 LED 和 I/O LED 确定各个从机的运行状态。

地址 LED	I/O LED	说明
开	开或关	该地址的从机可用。
开	闪烁	该地址的从机可用，但是有一个错误。
闪烁	关	此地址没有分配从机。
关	关	因为 AS-Interface 没有供电或 AS-Interface 主机模块标准保护脱机，禁用了 AS-Interface 总线通信。

### 从机 I/O 状态

可在地址 LED 和 I/O LED 上监控各个从机的 I/O 状态。在 AS-Interface 主机模块上监控从机 I/O 状态时，使用短按更改从机地址。短按 PB1 递增地址。在最后一个地址 (31B) 时再短按一次将返回第一个地址 (0A)。短按 PB2 递减地址。在第一个地址 (0A) 时再短按一次将返回最后一个地址 (31B)。

下图说明当您在地址 LED 指示 25A 时按下 PB1 按钮的情况。地址 LED 递增到分配了从机的 26A。请注意如果没有分配从机，地址 LED 闪烁。



监控从机地址 25A

地址 LED 因为没有分配从机而闪烁。

监控从机地址 26A

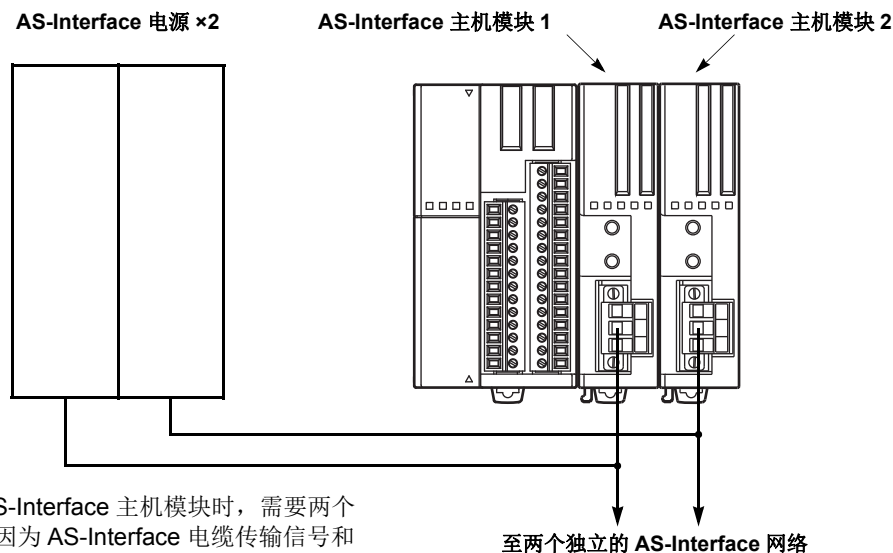
地址 LED 点亮，并且 I/O LED 指示状态。

### AS-Interface 设备

这一节描述 CPU 模块中分配的设备控制和监控 AS-Interface 主机模块，以及用于更新 CPU 模块中的 AS-Interface 设备或控制 AS-Interface 主机模块的 ASI 命令。

FC5A MicroSmart CPU 模块可与一个或两个 AS-Interface 主机模块一起使用。对于靠近 CPU 模块安装的第一个 AS-Interface 主机模块，可通过 AS-Interface 设备访问 AS-Interface 对象，如第 24-19 页上所示的内部继电器 M1300 ~ M1997 和数据寄存器 D1700 ~ D1999。

以下页面描述了有关 AS-Interface 主机模块 2 的 AS-Interface 对象详细信息。



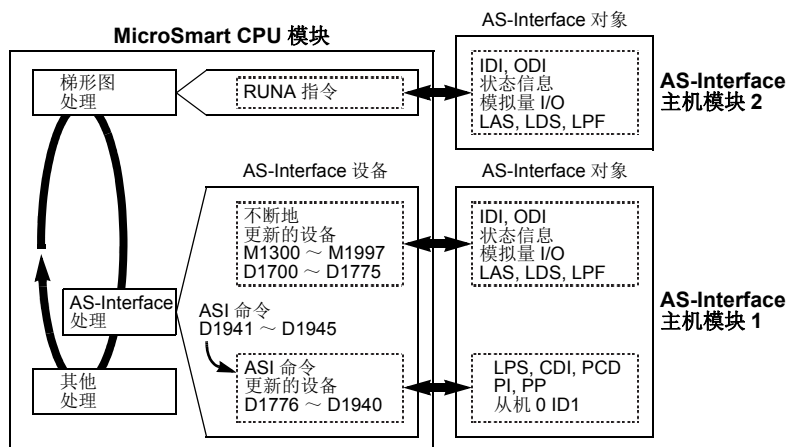
**注释：**当使用两个 AS-Interface 主机模块时，需要两个 AS-Interface 电源。因为 AS-Interface 电缆传输信号和电力，每个网络需要一个独立的电源。

### 处理时间

对于 AS-Interface 主机模块 1，用于数字量 I/O 和状态信息的 AS-Interface 内部继电器及用于 LAS、LDS 和 LPF 的数据寄存器在每次扫描中更新。只有在模拟量 I/O 连接到 AS-Interface 总线上时，模拟量 I/O 设备的数据寄存器才在每次扫描中更新。第 24-19 页上的表中说明了这些 AS-Interface 设备的处理时间。

当 CPU 模块中执行 ASI 命令时，更新其他 AS-Interface 数据寄存器。有关 ASI 命令的处理时间，请参阅第 24-30 页。

对于 AS-Interface 主机模块 1，使用 RUNA 指令更新 AS-Interface 对象。





## 访问 AS-Interface 主机模块 1 的 AS-Interface 对象

AS-Interface 总线上从机的 I/O 数据和参数，AS-Interface 总线的状态和从机的各种列表信息分配到 AS-Interface 主机模块 EEPROM。此信息称为 AS-Interface 对象，可通过 AS-Interface 设备访问，如内部继电器 M1300 ~ M1997 和数据寄存器 D1700 ~ D1999。AS-Interface 主机模块 1 的分配如下表所示。

MicroSmart CPU 模块设备地址		处理时间 (ms) *1	读 / 写	AS-Interface 主机模块 EEPROM	设备更新
设备	AS-Interface 主机模块 1			AS-Interface 对象	
AS-Interface 内部继电器	M1300-M1617	3.0	R *2	数字量输入 (IDI: 输入数据图像)	每次扫描
	M1620-M1937	3.0	W *2	数字量输出 (ODI: 输出数据图像)	
	M1940-M1997	1.0	R	状态信息	
AS-Interface 数据寄存器	D1700-D1731	5.2	R	模拟量输入 *4	
	D1732-D1763	5.2	W	模拟量输出 *4	
	D1764-D1767	1.0	R *2	可用从机列表 (LAS)	
	D1768-D1771	1.0	R *2	已检从机列表 (LDS)	
	D1772-D1775	1.0	R *2	外部故障从机列表 (LPF)	
	D1776-D1779	1.0	R/W *2*3	预定从机列表 (LPS)	
	D1780-D1811	5.2	R *2	设置数据图像 A (CDI)	
	D1812-D1843	5.2	R *2	设置数据图像 B (CDI)	
	D1844-D1875	5.2	R/W *2*3	固定设置数据 A (PCD)	
	D1876-D1907	5.2	R/W *2*3	固定设置数据 B (PCD)	
	D1908-D1923	3.0	R *2	参数图像 (PI)	
	D1924-D1939	3.0	R/W *2*3	固定参数 (PP)	
D1940	0.7	R/W	从机 0 ID1 码	每次执行 ASI 命令。	
D1941-D1945	—	R/W	ASI 命令说明		
D1946-D1999	—	—	(保留)		

*1: CPU 模块更新设备数据需要的时间。当使用 AS-Interface 主机模块 1 时，扫描时间最少增加 10 ms。有关 AS-Interface 主机模块 2，请参阅第 24-32 页。

*2: 可使用 WindLDR 读取或写入 AS-Interface 设备数据。有关详细信息，请参阅第 24-34 页。

*3: 使用 WindLDR 设置 LPS、PCD 和 PP 并下载到 CPU 模块。有关详细信息，请参阅第 24-36 页。

*4: 只有在模拟量从机连接到 AS-Interface 总线上时，才更新模拟量 I/O 数据。

## 访问 AS-Interface 主机模块 2 的 AS-Interface 对象

当使用两个 AS-Interface 主机模块时，第二个 AS-Interface 主机模块的 AS-Interface 对象可分配到任何内部继电器和数据寄存器，并可以使用 RUNA 或 STPA 指令访问。请参阅第 24-32 页。

### AS-Interface 主机模块的 I/O 数据

AS-Interface 主机模块可处理数字量 I/O 数据和模拟量 I/O 数据。每个从机的数字量 I/O 数据最多是 4 个数字量输入和 4 个数字量输出。每个从机的模拟量 I/O 数据包括 4 个频道 16 位模拟量输入或输出数据。

#### 标准从机和扩展从机的数字量 I/O 数据

有关 AS-Interface 主机模块，AS-Interface 总线上标准从机和 A/B 从机（传感器和致动器）的数字量 I/O 数据分配到 AS-Interface 内部继电器，自从机 0 开始按照升序排列。各个从机的输入数据图象 (IDI) 分配给 M1300 ~ M1617，输出数据图象 (ODI) 分配给 M1620 ~ M1937。例如，就从机 3A 而言，输入数据分配到 M1314(DI0) ~ M1317(DI3)，输出数据分配到 M1634(DO0) ~ M1637(DO3)。

对于 AS-Interface 主机模块 2，可使用 RUNA 或 STPA 指令访问数字量 I/O 数据。

#### • 数字量输入数据图像 (IDI)

设备地址		数据格式							
AS-Interface 主机 模块 1	AS-Interface 主机 模块 2	7 (DI3)	6 (DI2)	5 (DI1)	4 (DI0)	3 (DI3)	2 (DI2)	1 (DI1)	0 (DI0)
M1300	+0(低位字节)	从机 1(A)				(从机 0)			
M1310	+0(高位字节)	从机 3(A)				从机 2(A)			
M1320	+1(低位字节)	从机 5(A)				从机 4(A)			
M1330	+1(高位字节)	从机 7(A)				从机 6(A)			
M1340	+2(低位字节)	从机 9(A)				从机 8(A)			
M1350	+2(高位字节)	从机 11(A)				从机 10(A)			
M1360	+3(低位字节)	从机 13(A)				从机 12(A)			
M1370	+3(高位字节)	从机 15(A)				从机 14(A)			
M1380	+4(低位字节)	从机 17(A)				从机 16(A)			
M1390	+4(高位字节)	从机 19(A)				从机 18(A)			
M1400	+5(低位字节)	从机 21(A)				从机 20(A)			
M1410	+5(高位字节)	从机 23(A)				从机 22(A)			
M1420	+6(低位字节)	从机 25(A)				从机 24(A)			
M1430	+6(高位字节)	从机 27(A)				从机 26(A)			
M1440	+7(低位字节)	从机 29(A)				从机 28(A)			
M1450	+7(高位字节)	从机 31(A)				从机 30(A)			
M1460	+8(低位字节)	从机 1B				—			
M1470	+8(高位字节)	从机 3B				从机 2B			
M1480	+9(低位字节)	从机 5B				从机 4B			
M1490	+9(高位字节)	从机 7B				从机 6B			
M1500	+10(低位字节)	从机 9B				从机 8B			
M1510	+10(高位字节)	从机 11B				从机 10B			
M1520	+11(低位字节)	从机 13B				从机 12B			
M1530	+11(高位字节)	从机 15B				从机 14B			
M1540	+12(低位字节)	从机 17B				从机 16B			
M1550	+12(高位字节)	从机 19B				从机 18B			
M1560	+13(低位字节)	从机 21B				从机 20B			
M1570	+13(高位字节)	从机 23B				从机 22B			
M1580	+14(低位字节)	从机 25B				从机 24B			
M1590	+14(高位字节)	从机 27B				从机 26B			
M1600	+15(低位字节)	从机 29B				从机 28B			
M1610	+15(高位字节)	从机 31B				从机 30B			

* 数据地址表示 RUNA 或 STPA 指令对话框中指定的数据地址的偏移量。

## • 数字量输出数据图像 (ODI)

设备地址		数据格式							
AS-Interface 主机 模块 1	AS-Interface 主机 模块 2	7 (D03)	6 (D02)	5 (D01)	4 (D00)	3 (D03)	2 (D02)	1 (D01)	0 (D00)
M1620	+0(低位字节)	从机 1(A)				(从机 0)			
M1630	+0(高位字节)	从机 3(A)				从机 2(A)			
M1640	+1(低位字节)	从机 5(A)				从机 4(A)			
M1650	+1(高位字节)	从机 7(A)				从机 6(A)			
M1660	+2(低位字节)	从机 9(A)				从机 8(A)			
M1670	+2(高位字节)	从机 11(A)				从机 10(A)			
M1680	+3(低位字节)	从机 13(A)				从机 12(A)			
M1690	+3(高位字节)	从机 15(A)				从机 14(A)			
M1700	+4(低位字节)	从机 17(A)				从机 16(A)			
M1710	+4(高位字节)	从机 19(A)				从机 18(A)			
M1720	+5(低位字节)	从机 21(A)				从机 20(A)			
M1730	+5(高位字节)	从机 23(A)				从机 22(A)			
M1740	+6(低位字节)	从机 25(A)				从机 24(A)			
M1750	+6(高位字节)	从机 27(A)				从机 26(A)			
M1760	+7(低位字节)	从机 29(A)				从机 28(A)			
M1770	+7(高位字节)	从机 31(A)				从机 30(A)			
M1780	+8(低位字节)	从机 1B				—			
M1790	+8(高位字节)	从机 3B				从机 2B			
M1800	+9(低位字节)	从机 5B				从机 4B			
M1810	+9(高位字节)	从机 7B				从机 6B			
M1820	+10(低位字节)	从机 9B				从机 8B			
M1830	+10(高位字节)	从机 11B				从机 10B			
M1840	+11(低位字节)	从机 13B				从机 12B			
M1850	+11(高位字节)	从机 15B				从机 14B			
M1860	+12(低位字节)	从机 17B				从机 16B			
M1870	+12(高位字节)	从机 19B				从机 18B			
M1880	+13(低位字节)	从机 21B				从机 20B			
M1890	+13(高位字节)	从机 23B				从机 22B			
M1900	+14(低位字节)	从机 25B				从机 24B			
M1910	+14(高位字节)	从机 27B				从机 26B			
M1920	+15(低位字节)	从机 29B				从机 28B			
M1930	+15(高位字节)	从机 31B				从机 30B			

* 数据地址表示 RUNA 或 STPA 指令对话框中指定的数据地址的偏移量。



**注意**

- 刚打开电源时，不能访问标准从机和扩展从机的数字量 I/O 数据。当特殊内部继电器 M1945(Normal Operation Active) 打开时，CPU 模块和连接的从机之间开始数据通信。在访问从机 I/O 数据之前，确认 M1945 打开。


## 24: AS-INTERFACE 主机通信

### 模拟量从机的模拟量 I/O 数据

对于 AS-Interface 主机模块 1，AS-Interface 总线上最多七个模拟量从机（每个从机四个频道）的 I/O 数据存储到 CPU 模块中的 AS-Interface 数据寄存器。模拟量从机地址（1 ~ 31）以升序排列。各个模拟量从机的输入数据分配到数据寄存器 D1700 ~ D1731，输出数据分配到 D1732 ~ D1763。

对于 AS-Interface 主机模块 2，可使用 RUNA 或 STPA 指令访问模拟量 I/O 数据。

AS-Interface 主机模块支持模拟量从机设置文件 7.3。

 <b>注意</b>	<ul style="list-style-type: none"> <li>AS-Interface 总线上最多可以连接七个模拟量从机。请勿在一条总线上连接八个以上的模拟量从机，否则从机不能正常工作。</li> <li>当数据寄存器 D1700 ~ D1731 分配给包括 7FFF 的模拟量输入时，请勿使用此数据编程，因为此值保留用于下列特殊内部继电器： <ul style="list-style-type: none"> <li>一个从机上分配给模拟量从机的未使用频道。（对于一个从机上没有分配给模拟量从机的频道，相应的数据寄存器保持一个不定值。）</li> <li>数据溢出。</li> <li>主机和模拟量从机之间的通信不同步。</li> </ul> </li> <li>当使用模拟量从机时，请阅读模拟量从机用户手册正确处理数据。</li> </ul>
---------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### • 模拟量输入数据

设备地址		频道编号	数据格式
AS-Interface 主机模块 1	AS-Interface 主机模块 2		
D1700	+0	频道 1	第一个数据 (AI0)
D1701	+1	频道 2	
D1702	+2	频道 3	
D1703	+3	频道 4	
D1704	+4	频道 1	第二个数据 (AI1)
D1705	+5	频道 2	
D1706	+6	频道 3	
D1707	+7	频道 4	
D1708	+8	频道 1	第三个数据 (AI2)
D1709	+9	频道 2	
D1710	+10	频道 3	
D1711	+11	频道 4	
D1712	+12	频道 1	第四个数据 (AI3)
D1713	+13	频道 2	
D1714	+14	频道 3	
D1715	+15	频道 4	
D1716	+16	频道 1	第五个数据 (AI4)
D1717	+17	频道 2	
D1718	+18	频道 3	
D1719	+19	频道 4	
D1720	+20	频道 1	第六个数据 (AI5)
D1721	+21	频道 2	
D1722	+22	频道 3	
D1723	+23	频道 4	
D1724	+24	频道 1	第七个数据 (AI6)
D1725	+25	频道 2	
D1726	+26	频道 3	
D1727	+27	频道 4	
D1728	+28	—	(保留)
D1729	+29	—	
D1730	+30	—	
D1731	+31	—	

## • 模拟量输出数据

设备地址		频道编号	数据格式
AS-Interface 主机模块 1	AS-Interface 主机模块 2		
D1732	+0	频道 1	第一个数据 (AO0)
D1733	+1	频道 2	
D1734	+2	频道 3	
D1735	+3	频道 4	
D1736	+4	频道 1	第二个数据 (AO1)
D1737	+5	频道 2	
D1738	+6	频道 3	
D1739	+7	频道 4	
D1740	+8	频道 1	第三个数据 (AO2)
D1741	+9	频道 2	
D1742	+10	频道 3	
D1743	+11	频道 4	
D1744	+12	频道 1	第四个数据 (AO3)
D1745	+13	频道 2	
D1746	+14	频道 3	
D1747	+15	频道 4	
D1748	+16	频道 1	第五个数据 (AO4)
D1749	+17	频道 2	
D1750	+18	频道 3	
D1751	+19	频道 4	
D1752	+20	频道 1	第六个数据 (AO5)
D1753	+21	频道 2	
D1754	+22	频道 3	
D1755	+23	频道 4	
D1756	+24	频道 1	第七个数据 (AO6)
D1757	+25	频道 2	
D1758	+26	频道 3	
D1759	+27	频道 4	
D1760	+28	—	(保留)
D1761	+29	—	
D1762	+30	—	
D1763	+31	—	

* 数据地址表示 RUNA 或 STPA 指令对话框中指定的数据地址的偏移量。

例如，当使用模拟量从机 1、13 和 20，模拟量输出从机 5 和 25，以及模拟量 I/O 从机 14 和 21 时，设置将如下分配模拟量 I/O 从机，并保持到执行下一个设置。总是给各个从机保留四个频道 (8 字节)。

AS-Interface 模块 2 的数据范围相同。

模拟量从机模块	AS-Interface 主机模块 1 数据存储	模拟量输入从机	AS-Interface 主机模块 1 数据存储	模拟量输出从机
第一个	D1700-D1703	从机 1	D1732-D1735	未使用
第二个	D1704-D1707	未使用	D1736-D1739	从机 5
第三个	D1708-D1711	从机 13	D1740-D1743	未使用
第四个	D1712-D1715	从机 14	D1744-D1747	从机 14
第五个	D1716-D1719	从机 20	D1748-D1751	未使用
第六个	D1720-D1723	从机 21	D1752-D1755	从机 21
第七个	D1724-D1727	未使用	D1756-D1759	从机 25
(第八个)	(D1728-D1731)	(保留)	(D1760-D1763)	(保留)

## 24: AS-INTERFACE 主机通信

### 状态信息

对于 AS-Interface 主机模块 1，状态信息分配到 AS-Interface 内部继电器 M1940 ~ M1997。这些内部继电器用于监控 AS-Interface 总线的状态。如果总线上发生错误，除了这些状态继电器之外，您还可个通过 AS-Interface 主机模块前部的状态 LED 确认错误。

对于 AS-Interface 主机模块 2，可使用 RUNA 或 STPA 指令访问状态信息。

#### • 状态信息内部继电器

设备地址		状态	说明		
AS-Interface 主机模块 1	AS-Interface 主机模块 2		开	关	
M1940	+0 (低位字节)	Config_OK	设置完成。	设置没有完成。	
M1941		LDS.0	AS-Interface 总线上检测到从机地址 0。	AS-Interface 总线上没有检测到从机地址 0。	
M1942		Auto_Address_Assign	启用了自动寻址。	禁用了自动寻址。	
M1943		Auto_Address_Available	自动寻址就绪。	自动寻址未就绪。	
M1944		设置	启用了设置模式。	设置模式除外。	
M1945		Normal_Operation_Active	启用了标准保护模式。	标准保护模式除外。	
M1946		APF/not APO	AS-Interface 电源故障。	AS-Interface 电源正常。	
M1947		Offline_Ready	启用了标准保护脱机。	标准保护脱机除外。	
M1950		+0	Periphery_OK	外部设备正常。	外部设备异常。
M1951-M1957		(高位字节)	(保留)	—	—
M1960	+1 (低位字节)	Data_Exchange_Active	启用了数据交换。	禁用了数据交换。	
M1961		脱机	通过按钮或 WindLDR 发出转至标准保护脱机命令。	没有发出转至标准保护脱机命令。	
M1962		连接模式	启用了连接模式。	启用了本地模式。	
M1963-M1967	+1 (高位字节)	(保留)	—	—	
M1970-M1997	+2	(保留)	—	—	

* 数据地址表示 RUNA 或 STPA 指令对话框中指定的数据地址的偏移量。

#### M1940 Config_OK

M1940 指示设置状态。当存储在 AS-Interface 主机模块 EEPROM 中的固定设置数据 (PCD) 与设置数据图像 (CDI) 匹配时，M1940 打开。当设置更改时，M1940 关闭，例如增加了新从机或从机发生故障。然后，FLT LED 点亮。

#### M1941 LDS.0

M1941 用于检查 AS-Interface 总线上是否存在地址为 0 的从机。在标准保护模式或保护模式中 AS-Interface 总线上检测到地址为 0 的从机时，或者 AS-Interface 主机模块在标准保护模式中一个从机地址更改为 0 时，M1941 打开。

#### M1942 Auto_Address_Assign

M1942 指示启用了自动寻址功能。默认设置是“启用”，并且 M1942 通常打开。此设置可以使用 ASI“启用自动寻址”和“禁用自动寻址”更改。

**注释：**在 AS-Interface 主机模块上启用了自动寻址时，您可以更换一个具有相同识别码的新从机，而不需要关闭 AS-Interface 总线。

- 如果更换的从机分配了与故障从机相同的地址，并具有相同的识别码，更换的从机将自动添加到 LDS(已检从机列表) 继续操作。如果更换的从机分配的地址或识别码不同于故障从机，FLT LED 将点亮。
- 当使用分配了地址 0(出厂设置) 并且具有相同识别码的新从机更换故障从机时，新从机将分配到故障从机的地址并添加到 LDS 和 LAS(可用从机列表)。如果更换从机的识别码不同于故障从机，FLT LED 将点亮。
- 只有在在一个从机发生故障时，更换从机的自动寻址功能才工作。此功能不能用于更换多个从机。

**M1943 Auto_Address_Available**

M1943 指示是否满足自动寻址的条件。当启用了自动寻址功能并且 AS-Interface 总线上只有一个故障从机 (一个 AS-Interface 主机模块不能识别的从机) 时, M1943 打开。

**M1944 设置**

M1944 指示 AS-Interface 主机模块处于设置模式 (打开) 还是其他模式 (关闭)。启用了设置模式时, M1944 保持打开, 并且 CNF LED 闪烁。

**M1945 Normal_Operation_Active**

当 AS-Interface 主机模块在标准保护模式时, M1945 保持打开。在其他模式时, M1945 关闭。当 M1945 打开时, CPU 模块开始与连接的从机交换数据通信。

**M1946 APF/not APO**

AS-Interface 电源发生故障时, M1946 打开, 同时 PWR LED 关闭。

**M1947 Offline_Ready**

M1947 指示 AS-Interface 主机模块处于标准保护脱机。在标准模式脱机时, M1947 保持打开, 并且 OFF LED 也保持点亮。

**M1950 Periphery_OK**

AS-Interface 主机模块检测到外部设备故障时, M1950 保持打开。当发现故障时, M1950 关闭。

**M1960 Data_Exchange_Active**

M1960 指示启用了数据交换。M1960 打开时, AS-Interface 主机模块在标准保护模式, 并且启用了 AS-Interface 主机模块与从机之间的数据交换。使用 ASI 命令“启用数据交换”和“禁用数据交换”可启用或禁用数据交换。

**M1961 脱机**

当发出切换到标准保护脱机时, M1961 打开。要从标准保护模式切换到标准保护脱机, 按 AS-Interface 主机模块上的 PB2 按钮, 或发出 ASI 命令“转至标准保护脱机”。M1961 保持打开, 直到退出标准保护脱机。

**M1962 连接模式**

M1962 指示 AS-Interface 主机模块处于连接模式。在连接模式时, M1962 保持打开。这时, LMO LED 保持关闭, 而 CMO LED 保持点亮。

## 24: AS-INTERFACE 主机通信

### 从机列表信息

对于 AS-Interface 主机模块 1，数据寄存器 D1764 ~ D1779 分配到从机列表信息，以确定从机的运行状态。从机列表信息分成四个列表。可用从机列表 (LAS) 显示正在运行的从机。已检从机列表 (LDS) 显示在 AS-Interface 总线上检测到的从机。外部故障从机列表 (LPF) 显示故障从机。预定从机列表 (LPS) 显示存储在 AS-Interface 主机模块中的从机设置。

对于 AS-Interface 主机模块 2，可使用 RUNA 或 STPA 指令访问从机列表信息。

### 可用从机列表 (LAS)

对于 AS-Interface 主机模块 1，分配数据寄存器 D1764 ~ D1767 读取 LAS。您可以检查寄存器位确定各个从机的运行状态。当一个位打开时，表示相应的从机可用。

设备地址		数据格式	
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 8	位 7 ~ 0
D1764	+0	从机 15(A) ~ 8(A)	从机 7(A) ~ 0
D1765	+1	从机 31(A) ~ 24(A)	从机 23(A) ~ 16(A)
D1766	+2	从机 15B ~ 8B	从机 7B ~ (0B)
D1767	+3	从机 31B ~ 24B	从机 23B ~ 16B

### 已检从机列表 (LDS)

对于 AS-Interface 主机模块 1，分配数据寄存器 D1768 ~ D1771 读取 LDS。您可以检查寄存器位确定各个从机的检测状态。当一个位打开时，表示主机已经检测到相应的从机。

设备地址		数据格式	
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 8	位 7 ~ 0
D1768	+4	从机 15(A) ~ 8(A)	从机 7(A) ~ 0
D1769	+5	从机 31(A) ~ 24(A)	从机 23(A) ~ 16(A)
D1770	+6	从机 15B ~ 8B	从机 7B ~ (0B)
D1771	+7	从机 31B ~ 24B	从机 23B ~ 16B

### 外部故障从机列表 (LPF)

对于 AS-Interface 主机模块 1，分配数据寄存器 D1772 ~ D1775 读取 LPF。您可以检查寄存器位确定各个从机的故障状态。当一个位打开时，表示相应的从机有故障。

设备地址		数据格式	
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 8	位 7 ~ 0
D1772	+8	从机 15(A) ~ 8(A)	从机 7(A) ~ 0
D1773	+9	从机 31(A) ~ 24(A)	从机 23(A) ~ 16(A)
D1774	+10	从机 15B ~ 8B	从机 7B ~ (0B)
D1775	+11	从机 31B ~ 24B	从机 23B ~ 16B

* 数据地址表示 RUNA 或 STPA 指令对话框中指定的数据地址的偏移量。



**预定从机列表 (LPS)**

对于 AS-Interface 主机模块 1，分配 D1776 ~ D1779 读取和写入 LPS。在 WindLDR 上执行“自动设置”或“手动设置”时，LPS 设置存储到 AS-Interface 主机模块。可使用 ASI 命令“读取 LPS”将 LPS 数据读取到数据寄存器 D1776 ~ D1779。然后，您可以检查寄存器确定从机预定。当一个位打开时，表示相应的从机是预定从机。在更改 LPS 设置后，执行 ASI 命令“读取 LPS”，然后您可以使用更新的数据执行程序。

对于 AS-Interface 主机模块 2，不能使用 RUNA 或 STPA 指令访问预定从机列表。

设备地址		数据格式	
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 8	位 7 ~ 0
D1776	—	从机 15(A) ~ 8(A)	从机 7(A) ~ 0
D1777	—	从机 31(A) ~ 24(A)	从机 23(A) ~ 16(A)
D1778	—	从机 15B ~ 8B	从机 7B ~ (0B)
D1779	—	从机 31B ~ 24B	从机 23B ~ 16B

**从机识别信息 (从机设置文件)**

对于 AS-Interface 主机模块 1，数据寄存器 D1780 ~ D1940 分配到从机识别信息或从机设置文件。从机设置信息包括分别指示从机类型和从机运行的设置数据和参数。

对于 AS-Interface 主机模块 2，不能使用 RUNA 或 STPA 指令访问从机识别信息。

**设置数据图像 (CDI)**

对于 AS-Interface 主机模块 1，分配数据寄存器 D1780 ~ D1843 读取各个从机的 CDI。CDI 是 AS-Interface 主机模块在通电时收集并存储到 AS-Interface 主机模块的当前从机设置数据。

CDI 由四个代码构成：ID 码、I/O 码、ID2 码和 ID1 码。没有连接到 AS-Interface 总线的从机 CDI 是 FFFFh。

可使用 ASI 命令“读取 CDI”将 CDI 数据读取到数据寄存器 D1780 ~ D1843。在使用 CDI 执行程序前，执行 ASI 命令“读取 CDI”。

设备地址		数据格式			
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 12 ID 码	位 11 ~ 8 I/O 码	位 7 ~ 4 ID2 码	位 3 ~ 0 ID1 码
D1780	—	从机 0			
D1781	—	从机 1(A)			
D1782	—	从机 2(A)			
D(1780+N)	—	从机 N(A)			
D1811	—	从机 31(A)			
D1812	—	(未使用)			
D1813	—	从机 1B			
D(1812+N)	—	从机 NB			
D1843	—	从机 31B			

## 24: AS-INTERFACE 主机通信

### 固定设置数据 (PCD)

对于 AS-Interface 主机模块 1，分配数据寄存器 D1844 ~ D1907 读取和写入各个从机的 PCD。跟 CDI 一样，PCD 由四个代码构成：ID 码、I/O 码、ID2 码和 ID1 码

当执行自动设置时，CDI 复制到 PCD 并存储到 AS-Interface 主机模块的 ROM。当您执行手动设置时，可以使用 WindLDR 上的“设置从机”对话框设置 PCD。将各个从机的 PCD 设置为与其 CDI 相同的值。如果一个从机的 PCD 与 CDI 不同，那么该从机不能正常工作。将空缺从机编号的 PCD 设置为 FFFFh。

可使用 ASI 命令“读取 PCD”将 PCD 数据读取到数据寄存器 D1844 ~ D1907。在使用 PCD 执行程序前，执行 ASI 命令“读取 PCD”。

设备地址		数据格式			
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 12 ID 码	位 11 ~ 8 I/O 码	位 7 ~ 4 ID2 码	位 3 ~ 0 ID1 码
D1844	—	从机 0			
D1845	—	从机 1(A)			
D1846	—	从机 2(A)			
D(1844+N)	—	从机 N(A)			
D1875	—	从机 31(A)			
D1876	—	(未使用)			
D1877	—	从机 1B			
D(1876+N)	—	从机 NB			
D1907	—	从机 31B			

### 参数图像 (PI)

对于 AS-Interface 主机模块 1，分配数据寄存器 D1908 ~ D1923 读取各个从机的 PI。PI 由四个参数构成：P3、P2、P1 和 P0。PI 是 AS-Interface 主机模块在通电时收集并存储到 AS-Interface 主机模块的当前从机参数数据。要更改 PI 设置，请使用 WindLDR(从机状态对话框) 或执行 ASI 命令“更改从机 PI”。

可使用 ASI 命令“读取 PI”将 PI 数据读取到数据寄存器 D1908 ~ D1923。在更改 PI 设置后，执行 ASI 命令“读取 PI”，然后您可以使用更新的 PI 数据执行程序。

设备地址		数据格式			
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 12 P3/P2/P1/P0	位 11 ~ 8 P3/P2/P1/P0	位 7 ~ 4 P3/P2/P1/P0	位 3 ~ 0 P3/P2/P1/P0
D1908	—	从机 3(A)	从机 2(A)	从机 1(A)	从机 0
D1909	—	从机 7(A)	从机 6(A)	从机 5(A)	从机 4(A)
D1910	—	从机 11(A)	从机 10(A)	从机 9(A)	从机 8(A)
D(1908+N/4)	—	从机 (N+3)(A)	从机 (N+2)(A)	从机 (N+1)(A)	从机 N(A)
D1915	—	从机 31(A)	从机 30(A)	从机 29(A)	从机 28(A)
D1916	—	从机 3B	从机 2B	从机 1B	(未使用)
D1917	—	从机 7B	从机 6B	从机 5B	从机 4B
D(1916+N/4)	—	从机 (N+3)B	从机 (N+2)B	从机 (N+1)B	从机 NB
D1923	—	从机 31B	从机 30B	从机 29B	从机 28B

**固定参数 (PP)**

对于 AS-Interface 主机模块 1，分配数据寄存器 D1924 ~ D1939 读取和写入各个从机的 PP。跟 PI 一样，PP 由四个参数构成：P3、P2、P1 和 P0。当执行自动设置时，PI 复制到 PP 并存储到 AS-Interface 主机模块的 ROM。当您执行手动设置时，可以使用 WindLDR 上的“设置从机”对话框设置 PP。

可使用 ASI 命令“读取 PP”将 PP 数据读取到数据寄存器 D1924 ~ D1939。在更改 PP 设置后，执行 ASI 命令“读取 PP”，然后您可以使用更新的 PP 数据执行程序。

设备地址		数据格式			
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 12 P3/P2/P1/P0	位 11 ~ 8 P3/P2/P1/P0	位 7 ~ 4 P3/P2/P1/P0	位 3 ~ 0 P3/P2/P1/P0
D1924	—	从机 3(A)	从机 2(A)	从机 1(A)	从机 0
D1925	—	从机 7(A)	从机 6(A)	从机 5(A)	从机 4(A)
D1926	—	从机 11(A)	从机 10(A)	从机 9(A)	从机 8(A)
D(1924+N/4)	—	从机 (N+3)(A)	从机 (N+2)(A)	从机 (N+1)(A)	从机 N(A)
D1931	—	从机 31(A)	从机 30(A)	从机 29(A)	从机 28(A)
D1932	—	从机 3B	从机 2B	从机 1B	(未使用)
D1933	—	从机 7B	从机 6B	从机 5B	从机 4B
D(1932+N/4)	—	从机 (N+3)B	从机 (N+2)B	从机 (N+1)B	从机 NB
D1939	—	从机 31B	从机 30B	从机 29B	从机 28B

**更改从机 0 的 ID1 码**

对于 AS-Interface 主机模块 1，分配数据寄存器 D1940 读取和写入从机 0 的 ID1 码，要更改从机 0 的 ID1 设置，请在 D1940 中存储需要的值并执行 ASI 命令写入从机 ID1。可使用 ASI 命令“写入从机 0 ID1”将从机 0 ID1 数据读取到数据寄存器 D1940。在更改从机 0 ID1 设置后，执行 ASI 命令“读取从机 0 ID1”，然后您可以使用更新的从机 0 ID1 数据执行程序。

设备地址		数据格式			
AS-Interface 主机模块 1	AS-Interface 主机模块 2	位 15 ~ 12	位 11 ~ 8	位 7 ~ 4	位 3 ~ 0
D1940	—	—	—	—	ID1 码

## 24: AS-INTERFACE 主机通信

### ASI 命令 (AS-Interface 主机模块 1)

ASI 命令用于更新 CPU 模块中的 AS-Interface 设备或控制 AS-Interface 主机模块 1。数据寄存器 D1941 ~ D1944 用于存储命令数据。D1945 用于在执行命令前存储一个请求代码。在执行命令期间，D1945 状态和结果代码。

#### ASI 命令格式

命令部分 (8 字节)				请求 / 结果
D1941	D1942	D1943	D1944	D1945

#### ASI 命令数据

要执行 ASI 命令，将所需值存储到数据寄存器 D1941 ~ D1945，如下表所列。

ASI 命令	处理时间 (ms)	说明	命令数据 (十六进制)				
			D1941	D1942	D1943	D1944	D1945
读取 LPS	1.0 ^{*3}	将 LPS 读取到 D1776 ~ D1779	010B	084C	0000	0000	0001
读取 CDI	10.4 ^{*3}	将 CDI 读取到 D1780 ~ D1843	010C	4050	0000	0000	0001
读取 PCD	10.4 ^{*3}	将 PCD 读取到 D1844 ~ D1907	010E	4090	0000	0000	0001
读取 PI	3.0 ^{*3}	将 PI 读取到 D1908 ~ D1923	0107	20D0	0000	0000	0001
读取 PP	3.0 ^{*3}	将 PP 读取到 D1924 ~ D1939	0108	20E0	0000	0000	0001
读取从机 0 ID1	0.7 ^{*3}	将从机 0 ID1 读取到 D1940	0109	02F0	0000	0000	0001
写入从机 0 ID1	0.7 ^{*3}	将 D1940 写入从机 0 ID1	0209	02F0	0000	0000	0001
将 PI 复制到 PP	0.8 ^{*4}	将参数图像复制到固定参数	0306	0100	0000	0000	0001
更改从机 PI ^{*1}	0.8 ^{*4}	将 PI(*) 写入 (**)(注释)	0306	0102	000*	00**	0001
转至标准保护脱机	0.8 ^{*4}	从标准保护模式转至标准保护脱机	0306	0301	0000	0000	0001
转至标准保护模式	0.8 ^{*4}	从标准保护脱机转至标准保护模式	0306	0300	0000	0000	0001
禁用数据交换	0.8 ^{*4}	从标准保护模式到标准保护数据交换关闭	0306	0401	0000	0000	0001
启用数据交换	0.8 ^{*4}	从标准保护数据交换关闭到标准保护模式	0306	0400	0000	0000	0001
更改从机地址 ^{*2}	0.8 ^{*4}	将从机地址 (**) 更改为新地址 (++) (注释)	0306	0500	00**	00++	0001
启用自动寻址	0.8 ^{*4}	启用自动寻址赋值 (默认)	0306	0800	0000	0000	0001
禁用自动寻址	0.8 ^{*4}	禁用自动寻址赋值	0306	0801	0000	0000	0001

*1: WindLDR 具有“从机状态”对话框执行此命令，将 PI 值写入指定的从机。请参阅第 24-31 页上的样本程序。

*2: WindLDR 具有“更改从机地址”对话框执行此命令。

*3: 当五个数据寄存器存储相应的值时，在一次扫描中完成。当完成时，D1945 存储 4。请参阅第 24-31 页上的请求和结果代码。其他命令执行完成需要几次扫描。

*4: 每次扫描延长 0.8 ms，到 ASI 命令生效最少需要 1 秒钟。

注释：如下表所示在数据寄存器中指定从机地址：

从机地址	数据寄存器值		从机地址	数据寄存器值	
	十六进制	十进制		十六进制	十进制
0(A)	00h	0	—	—	—
1(A)	01h	1	1B	21h	33
2(A)	02h	2	2B	22h	34
31(A)	1Fh	31	31B	3Fh	63

请求和结果代码

D1945 值低位字节	说明	注释
00h	通电时的初始值	当 D1945 值低位字节存储 01h、02h 或 08h 时，请勿将任何值写入 D1945，否则 ASI 命令不能正确执行。CPU 模块自动存储所有的值，01h 除外。
01h	请求	
02h	处理 ASI 命令	
04h	正常完成	
08h	(执行设置)	
14h	外部设备故障	
24h	ASI 命令错误	
74h	不能执行	
84h	执行导致错误	

示例程序：更改从机 PI

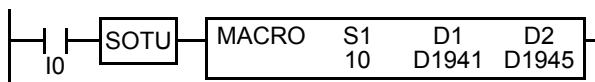
此示例程序更改从机 1A ~ 3 的 PI 值。要使用 ASI 命令“更改从机 PI”，使用 WindLDR 上的 MACRO 指令将新参数值 3 ~ 1943 和 1 ~ 1944 存储到指定的从机地址。

程序	命令数据 (十六进制)				
	D1941	D1942	D1943	D1944	D1945
将 PI 参数“3”写入从机 1A	0306	0102	0003	0001	0001

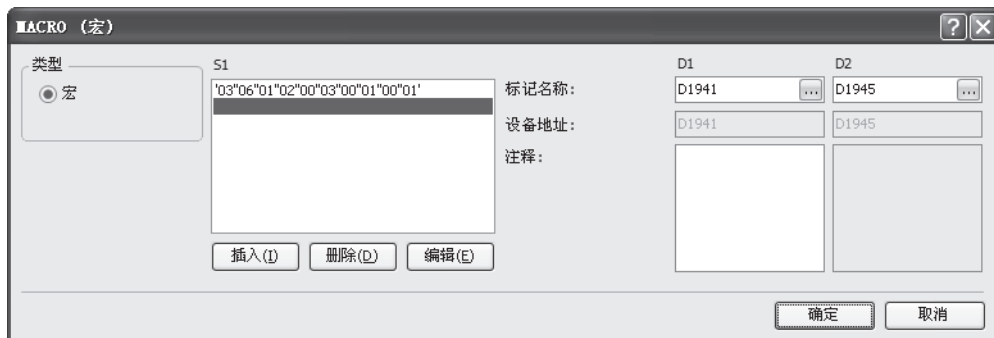
要指定从机 31A，将 001F 设置到 D1944。对于从机 1B，设置 0021。

如下表所示，衡量参数 P3 ~ P0 的重要性。当 PI 参数设置为 3，P3 和 P2 关闭时，关闭了 P3 和 P2，并且打开了 P1 和 P0。

参数	P3	P2	P1	P0
重量	8	4	2	1
开/关	关	关	开	开




当输入 I0 打开时，MACRO 指令将十六进制 0306、0102、0003、0001 和 0001 存储到五个数据寄存器 D1941 ~ D1945。



### 使用两个 AS-Interface 主机模块

FC5A MicroSmart CPU 模块可与一个或两个 AS-Interface 主机模块一起使用。对于靠近 CPU 模块安装的第一个 AS-Interface 主机模块，可通过 AS-Interface 设备访问 AS-Interface 对象，如第 24-19 页上所示的内部继电器 M1300 ~ M1997 和数据寄存器 D1700 ~ D1999。

 <b>注意</b>	<ul style="list-style-type: none"> <li>当使用 WindLDR 对 AS-Interface 主机模块 2 进行主机设置或从机监控时，不能执行 AS-Interface 主机模块 2 的 RUNA 和 STPA 指令。</li> </ul>
	<ul style="list-style-type: none"> <li>ASI 不能用于 AS-Interface 主机模块 2。使用 AS-Interface 主机模块 2 上的按钮 PB1 和 PB2 转至“标准保护模式”、“标准保护脱机”和“标准保护数据交换关闭”。</li> </ul>
	<ul style="list-style-type: none"> <li>AS-Interface 主机模块 2 没有更改从机 1 的 ID1 码和启用 / 禁用自动寻址的功能。一直启用自动寻址。</li> </ul>

### 访问 AS-Interface 主机模块 2 的 AS-Interface 对象。

AS-Interface 总线上从机的 I/O 数据和参数，AS-Interface 总线的状态和从机的各种列表信息分配到 AS-Interface 主机模块 EEPROM。此信息称为 AS-Interface 对象。第二个 AS-Interface 主机模块的 AS-Interface 对象可分配到任何内部继电器和数据寄存器，并且可使用 RUNA 或 STPA 指令访问。

AS-Interface 主机模块 2 的数据地址如下表所示。

AS-Interface 主机模块 2		处理 时间 (ms) *1	读 / 写	AS-Interface 主机模块 EEPROM
数据地址	数据大小 (字节)			AS-Interface 对象
0	32	3.0	R	数字量输入 (IDI: 输入数据图像)
3	32	3.0	W	数字量输出 (ODI: 输出数据图像)
2	6	1.0	R	状态信息
1	64	5.2	R	模拟量输入
4	64	5.2	W	模拟量输出
9	24	3.0	R	可用从机列表 (LAS) 已检从机列表 (LDS) 外部故障从机列表 (LPF)
—	—	—	—	预定从机列表 (LPS) *2
—	—	—	—	设置数据图像 A (CDI) *2
—	—	—	—	设置数据图像 B (CDI) *2
—	—	—	—	固定设置数据 A (PCD) *2
—	—	—	—	固定设置数据 B (PCD) *2
—	—	—	—	参数图像 (PI) *2
—	—	—	—	固定参数 (PP) *2
—	—	—	—	从机 0 ID1 码
—	—	—	—	ASI 命令说明

*1: CPU 模块为 RUNA 或 STPA 指令更新设备数据需要的时间。例如，当每次扫描中连续读取 IDI、ODI、状态信息、LAS、LDS 和 LPF 时，扫描时间增加 10 ms。

*2: 可使用 WindLDR 读取或写入 AS-Interface 设备数据。有关详细信息，请参阅第 24-34 页。

## 访问 AS-Interface 主机模块 2 的 AS-Interface 对象的 WindLDR 程序

下列示例演示使用 RUNA 指令把 AS-Interface 对象分配到内部继电器。数字量输入 (IDI)、数字量输出 (ODI) 和状态信息读取到内部继电器，或从内部继电器写入。

虽然不包括在示例程序中，模拟量输入和模拟量输出也可以使用 RUNA 或 STPA 指令分配到数据寄存器。

跟 AS-Interface 主机模块 1 一样，使用 WindLDR 上的“设置 AS-Interface 主机”对话框可访问其他 AS-Interface 对象，例如可用从机列表 (LAS)，已检从机列表 (LDS)，外部故障从机列表 (LPF)，预定从机列表 (LPS)，设置数据图像 (CDI)，固定设置数据 (PCD)，参数图像 (PI) 和固定参数 (PP)。

### 设置步骤

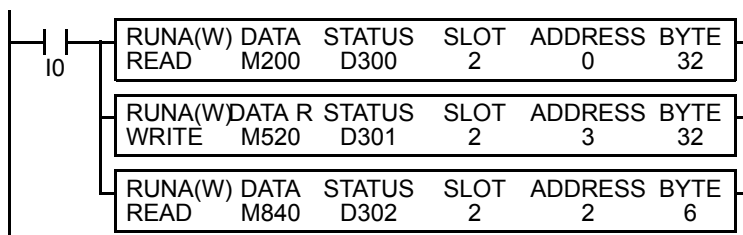
1. 确定要访问的 AS-Interface 对象和分配 AS-Interface 对象的 MicroSmart 设备。

AS-Interface 主机模块 2		读 / 写	MicroSmart 设备	AS-Interface 主机模块 EEPROM
数据地址	数据大小 (字节)			AS-Interface 对象
0	32	R	M200-M517	数字量输入 (IDI)
3	32	W	M520-M837	数字量输出 (ODI)
2	6	R	M840-M897	状态信息

2. 确认安装 AS-Interface 模块 2 的插槽编号。

有关此示例程序的系统设置，请参阅第 24-18 页。

插槽从 1 开始编号，按照与 CPU 模块距离增加的顺序。插槽编号中包括了所有扩展模块，例如数字量 I/O 模块，模拟量 I/O 模块和 AS-Interface 模块。



当打开 I/O 时，执行 RUNA 指令读取和写入指定的数据。

第二行的 RUNA WRITE 指令编写如下所示。



## 使用 WindLDR

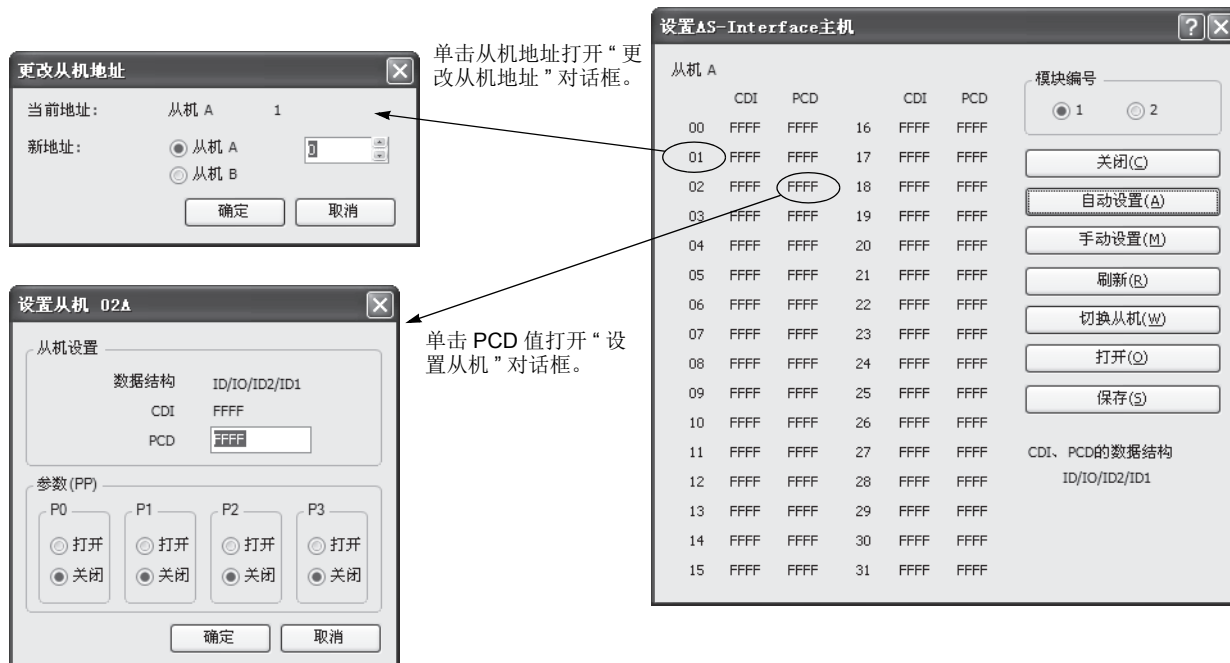
这一节描述 AS-Interface 系统使用 WindLDR 的步骤。WindLDR 包含设置从机和更改从机地址的“设置 AS-Interface 主机”对话框，和监控从机运行的“监控 AS-Interface 从机”对话框。

有关选择 PLC 类型和“功能设置”的步骤，请参阅第 24-8 页。

### 设置 AS-Interface 主机

AS-Interface 兼容从机设备出厂设置为地址 0，必须分配唯一的从机地址，以便主机和从机能正常通信。

从 WindLDR 菜单栏中选择**联机 > AS-Interface > 主机设置**。“设置 AS-Interface 主机”对话框出现。



对话框	按钮	说明
设置 AS-Interface 主机	自动设置	把当前连接的 AS-Interface 从机设置 (LDS、CDI、PI) 信息写入 AS-Interface 主机模块 ROM (LPS、PCD、PP)。
	手动设置	把从机 PCD 和用户设置的参数写入 AS-Interface 主机模块 ROM (LPS、PCD、PP)。
	刷新	刷新屏幕显示。
	切换从机	切换设置从机 A 或从机 B 的对话框。
	文件打开	打开设置 (LPS、PCD、PP) 文件
	文件保存	保存设置 (LPS、PCD、PP) 文件
	帮助	显示屏幕上的功能说明。
更改从机地址	确定	更改从机地址
	取消	放弃更改并关闭窗口。
设置从机	确定	更新 PCD 和 PP。还没有写入主机模块。
	取消	放弃更改并关闭窗口。



### 保存地址阴影颜色

通过查看“设置 AS-Interface 主机”对话框上从机地址的阴影颜色可确认从机的运行状态。单击**刷新**按钮可更新屏幕显示。

地址阴影	说明	LAS 可用从机列表	LDS 已检从机列表	LPF 外部故障从机 列表	LPS 预定从机列表
没有阴影	主机没有识别该从机。	关	关	关	开 / 关
蓝色阴影	从机可用。	开	开	关	开
黄色阴影	已识别从机，但没有启用。	关	开	关	关
红色阴影	从机中发现错误。	开 / 关	开 / 关	开	开 / 关

### 更改从机地址

当一个从机连接到 AS-Interface 主机模块时，可使用 WindLDR 更改从机地址。



**警告**

- 重复从机地址

每个从机必须有唯一的地址。请勿连接两个以上地址相同的从机，否则 AS-Interface 主机模块不能正确找到从机。如果两个从机具有相同的地址和不同的识别码 (ID、I/O、ID2、ID1)，AS-Interface 主机模块会检测到错误。如果两个从机具有相同的地址和相同的识别码，AS-Interface 主机模块不能检测到错误。不遵守此警告会造成严重的人身伤害或严重的财产损失。



**注意**

- 当地址为 0 的从机连接到 AS-Interface 主机模块时，首先给 MicroSmart CPU 模块通电。约五秒钟后，打开 AS-Interface 电源。如果同时打开 CPU 模块和 AS-Interface 的电源，AS-Interface 主机模块进入标准保护脱机。在此模式中，可更改从机地址，但不能在 WindLDR 确认从机状态。

要更改从机地址，从 WindLDR 菜单栏中选择**联机 > AS-Interface > 主机设置**。“设置 AS-Interface 主机”对话框出现。

单击从机地址打开“更改从机地址”对话框。选择从机 A 或从机 B，在新地址字段输入需要的地址并单击**确定**。“更改从机地址”对话框关闭。新从机地址存储到从机模块非易失性存储器。

如果没有正确处理命令，将出现错误信息“AS-Interface 主机错误”和一个错误代码。请参阅第 24-38 页。

在下列情况下不能更改地址。



错误代码	说明
1	• 扩展 I/O 总线上发现错误。
7	• 当 AS-Interface 主机模块处于本地模式。
8	<ul style="list-style-type: none"> <li>• 您试图更改的从机不存在。</li> <li>• 指定新地址的从机已经存在。</li> <li>• 当 A 地址设置了标准从机时，尝试在相同编号的 B 地址设置一个 A/B 从机。</li> <li>• 当 B 地址设置了 A/B 从机时，尝试在相同编号的 A 地址设置一个标准从机。</li> </ul>

### 设置

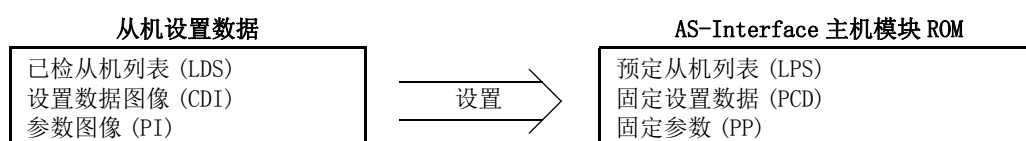
在试运行 AS-Interface 主机模块之前，必须使用 WindLDR 或 AS-Interface 主机模块前部的按钮进行设置。这一节描述使用 WindLDR 设置的方法。有关使用按钮设置，请参阅第 24-10 页。设置就是将下列信息存储到 AS-Interface 主机模块 ROM 的步骤。

- 要使用的从机地址列表
- 指定从机类型或识别码 (ID、I/O、ID2、ID1) 的设置数据
- 在通电时指定从机运行的参数 (P3、P2、P1、P0)

WindLDR 提供两种设置选项：指定自动设置的自动设置和根据用户选择的数据执行设置的手动设置。

### 自动设置

自动设置把当前从机设置数据 (LDS、CDI、PI) 存储到 AS-Interface 主机模块 ROM(LPS、PCD、PP)。要执行自动设置，请在“设置 AS-Interface 主机”对话框中按**自动设置**。自动设置跟使用 AS-Interface 主机模块上的按钮有相同的作用。



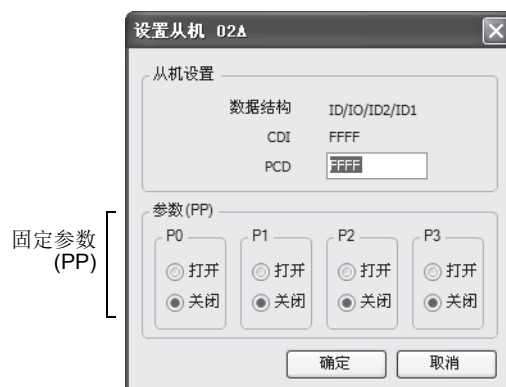
### 手动设置

手动设置 WindLDR 上指定的 LPS、PCD 和 PP 写入 AS-Interface 主机模块 ROM 的步骤。WindLDR 根据 PCD 的值自动生成 LPS。

PCD	LPS
FFFFh	0
其他值	1

要更改 PCD 和 PP，请使用“设置从机”对话框。将各个从机的 PCD 设置为与其 CDI 相同的值。如果一个从机的 PCD 与 CDI 不同，那么该从机不能正常工作。将空缺从机编号的 PCD 设置为 FFFFh。

在输入 PCD 值并选择参数状态后，单击**确定**。此时，设置数据没有存储到 AS-Interface 主机模块 ROM。要存储更改，请在“设置 AS-Interface 主机”对话框中单击**手动设置**。“设置 AS-Interface 主机”对话框的屏幕显示可使用**刷新**更新。



如果您把设置数据保存到一个文件中，您可以打开文件使用相同的数据设置其他 AS-Interface 主机模块。要保存打开的设置文件，单击**文件保存**或**文件打开**。

如果没有正确处理设置命令，将出现错误信息“AS-Interface 主机错误”和一个错误代码。请参阅第 24-38 页。

如果出现“设置失败，请确认从机设置并重新进行设置”错误信息，并且 FLT LED 点亮，请参阅第 24-13 页去除错误原因并重复设置。

在下列情况下不能进行设置。

错误代码	说明
1	●扩展 I/O 总线上发现错误。
2	●当 AS-Interface 主机模块处于脱机模式时，尝试执行了自动设置或手动设置。
7	●当总线上存在从机地址 0 时，尝试执行了自动设置或手动设置。 ●当 AS-Interface 主机模块处于本地模式。

## 监控 AS-Interface 从机

在 MicroSmart 通过 AS-Interface 总线与 AS-Interface 从机通信时，可使用计算机上的 WindLDR 监控 AS-Interface 从机的运行状态。使用 WindLDR 也可以更改输出状态和参数图像 (PI)。

要打开“监控 AS-Interface 从机”对话框，从 WindLDR 菜单栏中选择**联机 > 监控 > 监控**。从 WindLDR 菜单栏中，选择**联机 > AS-Interface > 从机监控**。



对话框	按钮	说明
监控 AS-Interface 从机	切换从机	在从机 A 屏幕和从机 B 屏幕之间切换。
	关闭	关闭窗口。
	帮助	显示屏幕上的功能说明。
从机状态	存储	把输出状态和参数存储到从机。
	关闭	关闭窗口。

## 更改从机输出状态和参数

可更改连接到 AS-Interface 主机模块的从机输出状态和参数图像 (PI)。要打开“从机状态”对话框，在“监控 AS-Interface 从机”对话框中单击所需从机地址的输出。然后，根据需要单击打开或关闭按钮更改输出 DO0 ~ DO3 和参数 P0 ~ P3 的状态。单击**存储**保存对从机模块的更改。

如果没有正确处理命令，将出现错误信息“AS-Interface 主机错误”和一个错误代码。请参阅第 24-38 页。

在下列情况下不能更改输出状态和参数。

错误代码	说明
1	• 扩展 I/O 总线上发现错误。
7	• 当 AS-Interface 主机模块处于本地模式。
8	• 尝试了更改不存在从机的参数。

## 24: AS-INTERFACE 主机通信

### 错误信息

当 AS-Interface 主机模块返回一个错误时，WindLDR 将显示一个错误信息。错误代码及其含义如下所示。



错误代码	说明
1	<ul style="list-style-type: none"><li>• 扩展 I/O 总线上发现错误。</li></ul>
2	<ul style="list-style-type: none"><li>• 当 AS-Interface 主机模块处于脱机模式时，尝试执行了自动设置或手动设置。</li><li>• 发送了一个错误命令。</li></ul>
7	<ul style="list-style-type: none"><li>• 当总线上存在从机地址 0 时，尝试执行了自动设置或手动设置。</li><li>• 当 AS-Interface 主机模块处于本地模式。</li></ul>
8	<ul style="list-style-type: none"><li>• 您试图更改的从机不存在。</li><li>• 指定新地址的从机已经存在。</li><li>• 当 A 地址设置了标准从机时，尝试在相同编号的 B 地址设置一个 A/B 从机。</li><li>• 当 B 地址设置了 A/B 从机时，尝试在相同编号的 A 地址设置一个标准从机。</li><li>• 尝试了更改不存在从机的参数。</li></ul>

当 AS-Interface 主机模块没有返回回复信息时，将显示下列错误。



## SwitchNet 数据 I/O 端口 (AS-Interface 主机模块 1)

SwitchNet 控制单元可用作 AS-Interface 网络中的从机，有  $\phi 16\text{mmL6}$  系列和  $\phi 22\text{mmHW}$  系列供选择。MicroSmartAS-Interface 主机模块的输入信号读入分配给从机编号和 DI 编号指定输入点的内部继电器。同样，MicroSmartAS-Interface 主机模块的输出信号写入分配给从机编号和 DO 编号指定输出点的内部继电器。当为 MicroSmart 编写梯形图时，使用分配给 SwitchNet 控制单元输入信号和输出信号的内部继电器。

L6 系列和 HW 系列 SwitchNet 控制单元的模拟量 I/O 数据分配稍微不同。

### L6 系列数字量 I/O 数据分配

输入数据自从机发送到 AS-Interface 主机。输出数据自 AS-Interface 主机发送到从机。

SwitchNet L6 系列 从机单元	使用 I/O	输入数据 (从机发送数据)				输出数据 (从机接收数据)			
		DI3	DI2	DI1	DI0	DO3	DO2	DO1	DO0
按钮	1 输入	0	X1	1	1	*	—	—	—
前导灯	1 输出	0	0	1	1	*	—	—	X1
发光按钮	1 输入 / 1 输出	0	X1	1	1	*	—	—	X1
选择器、键控选择器、杠杆：2-位置	1 输入	0	X2	1	1	*	—	—	—
选择器、键控选择器、杠杆：3-位置	2 输入	X3	X3	1	1	*	—	—	—
发光选择器：2-位置	1 输入 / 1 输出	0	X2	1	1	*	—	—	X1
发光选择器：3-位置	2 输入 / 1 输出	X3	X3	1	1	*	—	—	X1

#### 注释：

- * AS-Interface 主机模块使用位 DO3 对 A/B 从机寻址。
- 上表中，标记 X1、X2 和 X3 的位用于 SwitchNet I/O 数据。
- X1: 当按下按钮时，输入数据是 1(开)。没有按下时，输入数据是 0(关)。当输出数据是 1(开)时，LED 点亮。当输出数据是 0(关)时，LED 关闭。
- X2: 2-位置选择器、键控选择器、发光选择器开关，2-位置杠杆开关的输入数据取决于如下所示的操作器位置。
- X3: 3-位置选择器、键控选择器、发光选择器开关，3-位置杠杆开关的输入数据取决于如下所示的操作器位置。

2-位置操作器	选择器		杠杆	
	左	右	上	下
操作器位置	左 / 下		右 / 上	
DI2	0		1	

3-位置操作器	选择器			杠杆		
	左	中	右	上	中	下
操作器位置	左 / 下	中	右 / 上			
DI3	0	0	1			
DI2	1	0	0			

- 未使用的输入位 DI3 和 DI2 是 0(关)，未使用的输入位 DI1 和 DI0 是 1(开)。从机忽略主机发送的未使用的输出数据(—)。

#### • 写入参数命令

0	0	A4	A3	A2	A1	A0	1	Sel P3	P2	P1	P0	PB	1
---	---	----	----	----	----	----	---	-----------	----	----	----	----	---

#### • 写入参数设置

LED 亮度	设置			备注
	输出选择	控制数据		
	P2	P1	P0	
100%	1: DO0 0: DO1	1	1	默认设置
50%		0	1	
25%		1	0	
12.5%		0	0	

## 24: AS-INTERFACE 主机通信

### HW 系列数字量 I/O 数据分配

输入数据自从机发送到 AS-Interface 主机。输出数据自 AS-Interface 主机发送到从机。

SwitchNet HW 系列 从机单元	使用 I/O	通信信息块 安装位置	输入数据 (从机发送数据)				输出数据 (从机接收数据)			
			DI3	DI2	DI1	DI0	DO3	DO2	DO1	DO0
按钮	1 输入	?	0	X1	1	1	*	—	—	—
前导灯	1 输出	?	0	0	1	1	*	—	—	X1
发光按钮	1 输入 / 1 输出	?	0	X1	1	1	*	—	—	X1
选择器、键控选择器：2- 位置	1 输入	?	0	X2	1	1	*	—	—	—
选择器、键控选择器：3- 位置	1 输入	?	0	X3	1	1	*	—	—	—
	1 输入	?	0	X3	1	1	*	—	—	—
发光选择器：2- 位置	1 输入 / 1 输出	?	0	X2	1	1	*	—	—	X1
发光选择器：3- 位置	1 输入	?	0	X3	1	1	*	—	—	—
	1 输入 / 1 输出	?	0	X3	1	1	*	—	—	X1

#### 注释：

- *AS-Interface 主机模块使用位 DO3 对 A/B 从机寻址。
- 上表中，标记 X1、X2 和 X3 的位用于 SwitchNet I/O 数据。
- X1: 当按下按钮时，输入数据是 1(开)。没有按下时，输入数据是 0(关)。当输出数据是 1(开)时，LED 点亮。当输出数据是 0(关)时，LED 关闭。
- X2: 2- 位置选择器、键控选择器和发光选择器开关的输入数据取决于如下所示的操作器位置。

2- 位置操作器	选择器	
	左	右
操作器位置	左	右
DI2	0	1

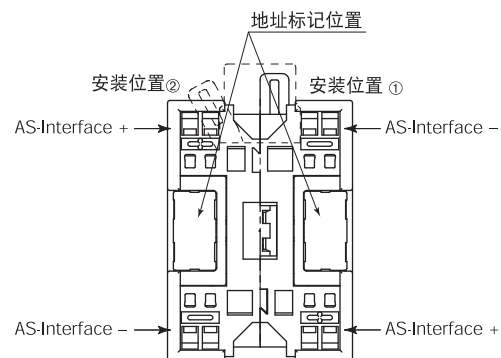
- X3: 3- 位置选择器、键控选择器和发光选择器开关的输入数据取决于如下所示的操作器位置。

3- 位置操作器	选择器			
	左	中	右	
操作器位置	左	中	右	
通信信息块安装位置	左	中	右	
?	DI2	1	0	0
?	DI2	0	0	1

如表和图所示，3- 位置选择器、键控选择器和发光选择器开关使用两个通信信息块。每个通信信息块必须有唯一的地址，因此 3- 位置选择器需要 2 个从机地址。

- 未使用的输入位 DI3 和 DI2 是 0(关)，未使用的输入位 DI1 和 DI0 是 1(开)。从机忽略主机发送的未使用的输出数据(—)。

#### 通信信息块安装位置 (背视图)



3- 位置选择器、键控选择器、发光选择器开关和通信信息块 ? 和 ? 安装在上面所示的位置。

#### ● 写入参数命令

0	0	A4	A3	A2	A1	A0	1	Sel P3	P2	P1	P0	PB	1
---	---	----	----	----	----	----	---	-----------	----	----	----	----	---

#### ● 写入参数设置

LED 亮度	设置			备注
	输出选择	控制数据		
	P2	P1	P0	
100%	1: DO0 0: DO1	1	1	默认设置
50%		0	1	
25%		1	0	
12.5%		0	0	

## •SwitchNet 从机的内部继电器 (AS-Interface 主机模块 1)

## • L6 系列

从机编号	按钮	前导灯	发光按钮		选择器、键控选择器、杠杆: 2-位置
	输入 DI2	输出 DO0	输入 DI2	输出 DO0	输入 DI2
(从机 0)	M1302	M1620	M1302	M1620	M1302
从机 1(A)	M1306	M1624	M1306	M1624	M1306
从机 2(A)	M1312	M1630	M1312	M1630	M1312
从机 3(A)	M1316	M1634	M1316	M1634	M1316
从机 4(A)	M1322	M1640	M1322	M1640	M1322
从机 5(A)	M1326	M1644	M1326	M1644	M1326
从机 6(A)	M1332	M1650	M1332	M1650	M1332
从机 7(A)	M1336	M1654	M1336	M1654	M1336
从机 8(A)	M1342	M1660	M1342	M1660	M1342
从机 9(A)	M1346	M1664	M1346	M1664	M1346
从机 10(A)	M1352	M1670	M1352	M1670	M1352
从机 11(A)	M1356	M1674	M1356	M1674	M1356
从机 12(A)	M1362	M1680	M1362	M1680	M1362
从机 13(A)	M1366	M1684	M1366	M1684	M1366
从机 14(A)	M1372	M1690	M1372	M1690	M1372
从机 15(A)	M1376	M1694	M1376	M1694	M1376
从机 16(A)	M1382	M1700	M1382	M1700	M1382
从机 17(A)	M1386	M1704	M1386	M1704	M1386
从机 18(A)	M1392	M1710	M1392	M1710	M1392
从机 19(A)	M1396	M1714	M1396	M1714	M1396
从机 20(A)	M1402	M1720	M1402	M1720	M1402
从机 21(A)	M1406	M1724	M1406	M1724	M1406
从机 22(A)	M1412	M1730	M1412	M1730	M1412
从机 23(A)	M1416	M1734	M1416	M1734	M1416
从机 24(A)	M1422	M1740	M1422	M1740	M1422
从机 25(A)	M1426	M1744	M1426	M1744	M1426
从机 26(A)	M1432	M1750	M1432	M1750	M1432
从机 27(A)	M1436	M1754	M1436	M1754	M1436
从机 28(A)	M1442	M1760	M1442	M1760	M1442
从机 29(A)	M1446	M1764	M1446	M1764	M1446
从机 30(A)	M1452	M1770	M1452	M1770	M1452
从机 31(A)	M1456	M1774	M1456	M1774	M1456
从机 1B	M1466	M1784	M1466	M1784	M1466
从机 2B	M1472	M1790	M1472	M1790	M1472
从机 3B	M1476	M1794	M1476	M1794	M1476
从机 4B	M1482	M1800	M1482	M1800	M1482
从机 5B	M1486	M1804	M1486	M1804	M1486
从机 6B	M1492	M1810	M1492	M1810	M1492
从机 7B	M1496	M1814	M1496	M1814	M1496
从机 8B	M1502	M1820	M1502	M1820	M1502
从机 9B	M1506	M1824	M1506	M1824	M1506
从机 10B	M1512	M1830	M1512	M1830	M1512
从机 11B	M1516	M1834	M1516	M1834	M1516
从机 12B	M1522	M1840	M1522	M1840	M1522
从机 13B	M1526	M1844	M1526	M1844	M1526
从机 14B	M1532	M1850	M1532	M1850	M1532
从机 15B	M1536	M1854	M1536	M1854	M1536
从机 16B	M1542	M1860	M1542	M1860	M1542
从机 17B	M1546	M1864	M1546	M1864	M1546
从机 18B	M1552	M1870	M1552	M1870	M1552
从机 19B	M1556	M1874	M1556	M1874	M1556
从机 20B	M1562	M1880	M1562	M1880	M1562

## 24: AS-INTERFACE 主机通信

从机编号	按钮	前导灯	发光按钮		选择器、键控选择器、杠杆：2-位置
	输入 DI2	输出 DO0	输入 DI2	输出 DO0	输入 DI2
从机 21B	M1566	M1884	M1566	M1884	M1566
从机 22B	M1572	M1890	M1572	M1890	M1572
从机 23B	M1576	M1894	M1576	M1894	M1576
从机 24B	M1582	M1900	M1582	M1900	M1582
从机 25B	M1586	M1904	M1586	M1904	M1586
从机 26B	M1592	M1910	M1592	M1910	M1592
从机 27B	M1596	M1914	M1596	M1914	M1596
从机 28B	M1602	M1920	M1602	M1920	M1602
从机 29B	M1606	M1924	M1606	M1924	M1606
从机 30B	M1612	M1930	M1612	M1930	M1612
从机 31B	M1616	M1934	M1616	M1934	M1616

### • L6 系列 (继续)

从机编号	选择器、键控选择器、杠杆：3-位置		发光选择器：2-位置		发光选择器：3-位置		
	输入 DI3	输入 DI2	输入 DI2	输出 DO0	输入 DI3	输入 DI2	输出 DO0
(从机 0)	M1303	M1302	M1302	M1620	M1303	M1302	M1620
从机 1(A)	M1307	M1306	M1306	M1624	M1307	M1306	M1624
从机 2(A)	M1313	M1312	M1312	M1630	M1313	M1312	M1630
从机 3(A)	M1317	M1316	M1316	M1634	M1317	M1316	M1634
从机 4(A)	M1323	M1322	M1322	M1640	M1323	M1322	M1640
从机 5(A)	M1327	M1326	M1326	M1644	M1327	M1326	M1644
从机 6(A)	M1333	M1332	M1332	M1650	M1333	M1332	M1650
从机 7(A)	M1337	M1336	M1336	M1654	M1337	M1336	M1654
从机 8(A)	M1343	M1342	M1342	M1660	M1343	M1342	M1660
从机 9(A)	M1347	M1346	M1346	M1664	M1347	M1346	M1664
从机 10(A)	M1353	M1352	M1352	M1670	M1353	M1352	M1670
从机 11(A)	M1357	M1356	M1356	M1674	M1357	M1356	M1674
从机 12(A)	M1363	M1362	M1362	M1680	M1363	M1362	M1680
从机 13(A)	M1367	M1366	M1366	M1684	M1367	M1366	M1684
从机 14(A)	M1373	M1372	M1372	M1690	M1373	M1372	M1690
从机 15(A)	M1377	M1376	M1376	M1694	M1377	M1376	M1694
从机 16(A)	M1383	M1382	M1382	M1700	M1383	M1382	M1700
从机 17(A)	M1387	M1386	M1386	M1704	M1387	M1386	M1704
从机 18(A)	M1393	M1392	M1392	M1710	M1393	M1392	M1710
从机 19(A)	M1397	M1396	M1396	M1714	M1397	M1396	M1714
从机 20(A)	M1403	M1402	M1402	M1720	M1403	M1402	M1720
从机 21(A)	M1407	M1406	M1406	M1724	M1407	M1406	M1724
从机 22(A)	M1413	M1412	M1412	M1730	M1413	M1412	M1730
从机 23(A)	M1417	M1416	M1416	M1734	M1417	M1416	M1734
从机 24(A)	M1423	M1422	M1422	M1740	M1423	M1422	M1740
从机 25(A)	M1427	M1426	M1426	M1744	M1427	M1426	M1744
从机 26(A)	M1433	M1432	M1432	M1750	M1433	M1432	M1750
从机 27(A)	M1437	M1436	M1436	M1754	M1437	M1436	M1754
从机 28(A)	M1443	M1442	M1442	M1760	M1443	M1442	M1760
从机 29(A)	M1447	M1446	M1446	M1764	M1447	M1446	M1764
从机 30(A)	M1453	M1452	M1452	M1770	M1453	M1452	M1770
从机 31(A)	M1457	M1456	M1456	M1774	M1457	M1456	M1774
从机 1B	M1467	M1466	M1466	M1784	M1467	M1466	M1784
从机 2B	M1473	M1472	M1472	M1790	M1473	M1472	M1790
从机 3B	M1477	M1476	M1476	M1794	M1477	M1476	M1794
从机 4B	M1483	M1482	M1482	M1800	M1483	M1482	M1800
从机 5B	M1487	M1486	M1486	M1804	M1487	M1486	M1804
从机 6B	M1493	M1492	M1492	M1810	M1493	M1492	M1810



从机编号	选择器、键控选择器、杠杆： 3- 位置		发光选择器：2- 位置		发光选择器：3- 位置		
	输入 DI3	输入 DI2	输入 DI2	输出 DO0	输入 DI3	输入 DI2	输出 DO0
从机 7B	M1497	M1496	M1496	M1814	M1497	M1496	M1814
从机 8B	M1503	M1502	M1502	M1820	M1503	M1502	M1820
从机 9B	M1507	M1506	M1506	M1824	M1507	M1506	M1824
从机 10B	M1513	M1512	M1512	M1830	M1513	M1512	M1830
从机 11B	M1517	M1516	M1516	M1834	M1517	M1516	M1834
从机 12B	M1523	M1522	M1522	M1840	M1523	M1522	M1840
从机 13B	M1527	M1526	M1526	M1844	M1527	M1526	M1844
从机 14B	M1533	M1532	M1532	M1850	M1533	M1532	M1850
从机 15B	M1537	M1536	M1536	M1854	M1537	M1536	M1854
从机 16B	M1543	M1542	M1542	M1860	M1543	M1542	M1860
从机 17B	M1547	M1546	M1546	M1864	M1547	M1546	M1864
从机 18B	M1553	M1552	M1552	M1870	M1553	M1552	M1870
从机 19B	M1557	M1556	M1556	M1874	M1557	M1556	M1874
从机 20B	M1563	M1562	M1562	M1880	M1563	M1562	M1880
从机 21B	M1567	M1566	M1566	M1884	M1567	M1566	M1884
从机 22B	M1573	M1572	M1572	M1890	M1573	M1572	M1890
从机 23B	M1577	M1576	M1576	M1894	M1577	M1576	M1894
从机 24B	M1583	M1582	M1582	M1900	M1583	M1582	M1900
从机 25B	M1587	M1586	M1586	M1904	M1587	M1586	M1904
从机 26B	M1593	M1592	M1592	M1910	M1593	M1592	M1910
从机 27B	M1597	M1596	M1596	M1914	M1597	M1596	M1914
从机 28B	M1603	M1602	M1602	M1920	M1603	M1602	M1920
从机 29B	M1607	M1606	M1606	M1924	M1607	M1606	M1924
从机 30B	M1613	M1612	M1612	M1930	M1613	M1612	M1930
从机 31B	M1617	M1616	M1616	M1934	M1617	M1616	M1934

## • HW 系列

从机编号	按钮	前导灯	发光按钮		选择器、键控选择器：2- 位置
	输入 DI2	输出 DO0	输入 DI2	输出 DO0	输入 DI2
(从机 0)	M1302	M1620	M1302	M1620	M1302
从机 1(A)	M1306	M1624	M1306	M1624	M1306
从机 2(A)	M1312	M1630	M1312	M1630	M1312
从机 3(A)	M1316	M1634	M1316	M1634	M1316
从机 4(A)	M1322	M1640	M1322	M1640	M1322
从机 5(A)	M1326	M1644	M1326	M1644	M1326
从机 6(A)	M1332	M1650	M1332	M1650	M1332
从机 7(A)	M1336	M1654	M1336	M1654	M1336
从机 8(A)	M1342	M1660	M1342	M1660	M1342
从机 9(A)	M1346	M1664	M1346	M1664	M1346
从机 10(A)	M1352	M1670	M1352	M1670	M1352
从机 11(A)	M1356	M1674	M1356	M1674	M1356
从机 12(A)	M1362	M1680	M1362	M1680	M1362
从机 13(A)	M1366	M1684	M1366	M1684	M1366
从机 14(A)	M1372	M1690	M1372	M1690	M1372
从机 15(A)	M1376	M1694	M1376	M1694	M1376
从机 16(A)	M1382	M1700	M1382	M1700	M1382
从机 17(A)	M1386	M1704	M1386	M1704	M1386
从机 18(A)	M1392	M1710	M1392	M1710	M1392
从机 19(A)	M1396	M1714	M1396	M1714	M1396
从机 20(A)	M1402	M1720	M1402	M1720	M1402
从机 21(A)	M1406	M1724	M1406	M1724	M1406
从机 22(A)	M1412	M1730	M1412	M1730	M1412
从机 23(A)	M1416	M1734	M1416	M1734	M1416
从机 24(A)	M1422	M1740	M1422	M1740	M1422

## 24: AS-INTERFACE 主机通信

从机编号	按钮	前导灯	发光按钮		选择器、键控选择器：2-位置
	输入 DI2	输出 DO0	输入 DI2	输出 DO0	输入 DI2
从机 25(A)	M1426	M1744	M1426	M1744	M1426
从机 26(A)	M1432	M1750	M1432	M1750	M1432
从机 27(A)	M1436	M1754	M1436	M1754	M1436
从机 28(A)	M1442	M1760	M1442	M1760	M1442
从机 29(A)	M1446	M1764	M1446	M1764	M1446
从机 30(A)	M1452	M1770	M1452	M1770	M1452
从机 31(A)	M1456	M1774	M1456	M1774	M1456
从机 1B	M1466	M1784	M1466	M1784	M1466
从机 2B	M1472	M1790	M1472	M1790	M1472
从机 3B	M1476	M1794	M1476	M1794	M1476
从机 4B	M1482	M1800	M1482	M1800	M1482
从机 5B	M1486	M1804	M1486	M1804	M1486
从机 6B	M1492	M1810	M1492	M1810	M1492
从机 7B	M1496	M1814	M1496	M1814	M1496
从机 8B	M1502	M1820	M1502	M1820	M1502
从机 9B	M1506	M1824	M1506	M1824	M1506
从机 10B	M1512	M1830	M1512	M1830	M1512
从机 11B	M1516	M1834	M1516	M1834	M1516
从机 12B	M1522	M1840	M1522	M1840	M1522
从机 13B	M1526	M1844	M1526	M1844	M1526
从机 14B	M1532	M1850	M1532	M1850	M1532
从机 15B	M1536	M1854	M1536	M1854	M1536
从机 16B	M1542	M1860	M1542	M1860	M1542
从机 17B	M1546	M1864	M1546	M1864	M1546
从机 18B	M1552	M1870	M1552	M1870	M1552
从机 19B	M1556	M1874	M1556	M1874	M1556
从机 20B	M1562	M1880	M1562	M1880	M1562
从机 21B	M1566	M1884	M1566	M1884	M1566
从机 22B	M1572	M1890	M1572	M1890	M1572
从机 23B	M1576	M1894	M1576	M1894	M1576
从机 24B	M1582	M1900	M1582	M1900	M1582
从机 25B	M1586	M1904	M1586	M1904	M1586
从机 26B	M1592	M1910	M1592	M1910	M1592
从机 27B	M1596	M1914	M1596	M1914	M1596
从机 28B	M1602	M1920	M1602	M1920	M1602
从机 29B	M1606	M1924	M1606	M1924	M1606
从机 30B	M1612	M1930	M1612	M1930	M1612
从机 31B	M1616	M1934	M1616	M1934	M1616

### • HW 系列 (继续)

从机编号	选择器、键控选择器：3-位置	发光选择器：2-位置		发光选择器：3-位置	
	输入 DI2 (通信信息块①②)	输入 DI2	输出 DO0	输入 DI2 (通信信息块①②)	输出 DO0 (通信信息块②)
(从机 0)	M1302	M1302	M1620	M1302	M1620
从机 1(A)	M1306	M1306	M1624	M1306	M1624
从机 2(A)	M1312	M1312	M1630	M1312	M1630
从机 3(A)	M1316	M1316	M1634	M1316	M1634
从机 4(A)	M1322	M1322	M1640	M1322	M1640
从机 5(A)	M1326	M1326	M1644	M1326	M1644
从机 6(A)	M1332	M1332	M1650	M1332	M1650
从机 7(A)	M1336	M1336	M1654	M1336	M1654
从机 8(A)	M1342	M1342	M1660	M1342	M1660
从机 9(A)	M1346	M1346	M1664	M1346	M1664
从机 10(A)	M1352	M1352	M1670	M1352	M1670
从机 11(A)	M1356	M1356	M1674	M1356	M1674

从机编号	选择器、键控选择器：3- 位置		发光选择器：2- 位置		发光选择器：3- 位置	
	输入 DI2 (通信信息块①②)		输入 DI2	输出 DOO	输入 DI2 (通信信息块①②)	输出 DOO (通信信息块②)
从机 12(A)	M1362	M1362	M1680	M1362	M1680	
从机 13(A)	M1366	M1366	M1684	M1366	M1684	
从机 14(A)	M1372	M1372	M1690	M1372	M1690	
从机 15(A)	M1376	M1376	M1694	M1376	M1694	
从机 16(A)	M1382	M1382	M1700	M1382	M1700	
从机 17(A)	M1386	M1386	M1704	M1386	M1704	
从机 18(A)	M1392	M1392	M1710	M1392	M1710	
从机 19(A)	M1396	M1396	M1714	M1396	M1714	
从机 20(A)	M1402	M1402	M1720	M1402	M1720	
从机 21(A)	M1406	M1406	M1724	M1406	M1724	
从机 22(A)	M1412	M1412	M1730	M1412	M1730	
从机 23(A)	M1416	M1416	M1734	M1416	M1734	
从机 24(A)	M1422	M1422	M1740	M1422	M1740	
从机 25(A)	M1426	M1426	M1744	M1426	M1744	
从机 26(A)	M1432	M1432	M1750	M1432	M1750	
从机 27(A)	M1436	M1436	M1754	M1436	M1754	
从机 28(A)	M1442	M1442	M1760	M1442	M1760	
从机 29(A)	M1446	M1446	M1764	M1446	M1764	
从机 30(A)	M1452	M1452	M1770	M1452	M1770	
从机 31(A)	M1456	M1456	M1774	M1456	M1774	
从机 1B	M1466	M1466	M1784	M1466	M1784	
从机 2B	M1472	M1472	M1790	M1472	M1790	
从机 3B	M1476	M1476	M1794	M1476	M1794	
从机 4B	M1482	M1482	M1800	M1482	M1800	
从机 5B	M1486	M1486	M1804	M1486	M1804	
从机 6B	M1492	M1492	M1810	M1492	M1810	
从机 7B	M1496	M1496	M1814	M1496	M1814	
从机 8B	M1502	M1502	M1820	M1502	M1820	
从机 9B	M1506	M1506	M1824	M1506	M1824	
从机 10B	M1512	M1512	M1830	M1512	M1830	
从机 11B	M1516	M1516	M1834	M1516	M1834	
从机 12B	M1522	M1522	M1840	M1522	M1840	
从机 13B	M1526	M1526	M1844	M1526	M1844	
从机 14B	M1532	M1532	M1850	M1532	M1850	
从机 15B	M1536	M1536	M1854	M1536	M1854	
从机 16B	M1542	M1542	M1860	M1542	M1860	
从机 17B	M1546	M1546	M1864	M1546	M1864	
从机 18B	M1552	M1552	M1870	M1552	M1870	
从机 19B	M1556	M1556	M1874	M1556	M1874	
从机 20B	M1562	M1562	M1880	M1562	M1880	
从机 21B	M1566	M1566	M1884	M1566	M1884	
从机 22B	M1572	M1572	M1890	M1572	M1890	
从机 23B	M1576	M1576	M1894	M1576	M1894	
从机 24B	M1582	M1582	M1900	M1582	M1900	
从机 25B	M1586	M1586	M1904	M1586	M1904	
从机 26B	M1592	M1592	M1910	M1592	M1910	
从机 27B	M1596	M1596	M1914	M1596	M1914	
从机 28B	M1602	M1602	M1920	M1602	M1920	
从机 29B	M1606	M1606	M1924	M1606	M1924	
从机 30B	M1612	M1612	M1930	M1612	M1930	
从机 31B	M1616	M1616	M1934	M1616	M1934	

注释：3- 位置选择器、键控选择器和发光选择器开关使用两个通信信息块，因此需要 2 个从机地址。有关通信信息快的安装位置，请参阅第 24-40 页。



# 25: 扩展 RS232C/RS485 通信

## 简介

本章介绍采用 FC5A-SIF2 扩展 RS232C 通信模块及 FC5A-SIF4 扩展 RS485 通信模块的通信实例。

有关扩展 RS232C/RS485 通信模块的规范, 请参见第 2-86 页 (基本卷)。

## 适用的 CPU 模块

FC5A-C10R2/C/D	FC5A-C16R2/C/D FC5A-C24R2D	FC5A-C24R2/C	FC5A-D16RK1/RS1	FC5A-D32K3/S3	FC5A-D12K1E/S1E
—	—	3 (注释)	5	5	5

FC5A-SIF2 扩展 RS232C 通信模块适用于升级后的系统程序版本 110 或更高的 CPU 模块。使用 FC5A-SIF4 扩展 RS485 通信模块时必须使用 220 或以上版本系统程序的 CPU 模块。系统程序版本可以使用 WindLDR 进行确认。请参阅第 13-2 页 (基本卷)。

如果 CPU 模块系统程序版本低于所要求的版本, 则可用 WindLDR 下载最新系统程序。详见第 A-9 页 (基本卷)。

**注释:** 集成 24-I/O 型 CPU 模块不能使用与下表所示的功能模块组合使用的扩展 RS232C/RS485 通信模块。如需使用扩展 RS232C/RS485 通信模块和这些功能模块时, 请使用超薄型 CPU 模块。

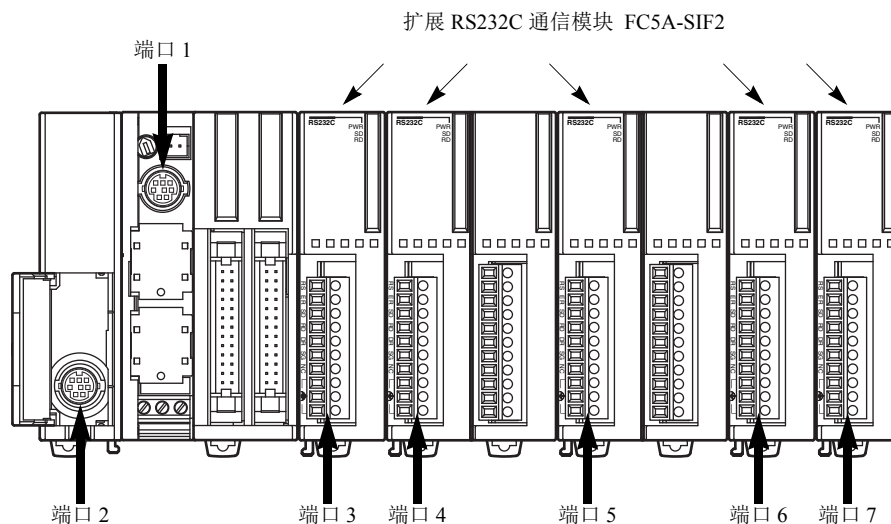
功能模块	型号
模拟量 I/O 模块	FC4A-L03A1, FC4A-L03AP1, FC4A-J2A1, FC4A-J4CN1, FC4A-J8C1, FC4A-J8AT1, FC4A-K1A1, FC4A-K2C1, FC4A-K4A1
AS-Interface 主机模块	FC4A-AS62M

## 分配通信端口编号

当安装扩展 RS232C/RS485 通信模块时, 如果安装了五个扩展 RS232C/RS485 通信模块, 端口编号将从端口 3 开始并在端口 7 结束。

### 示例:

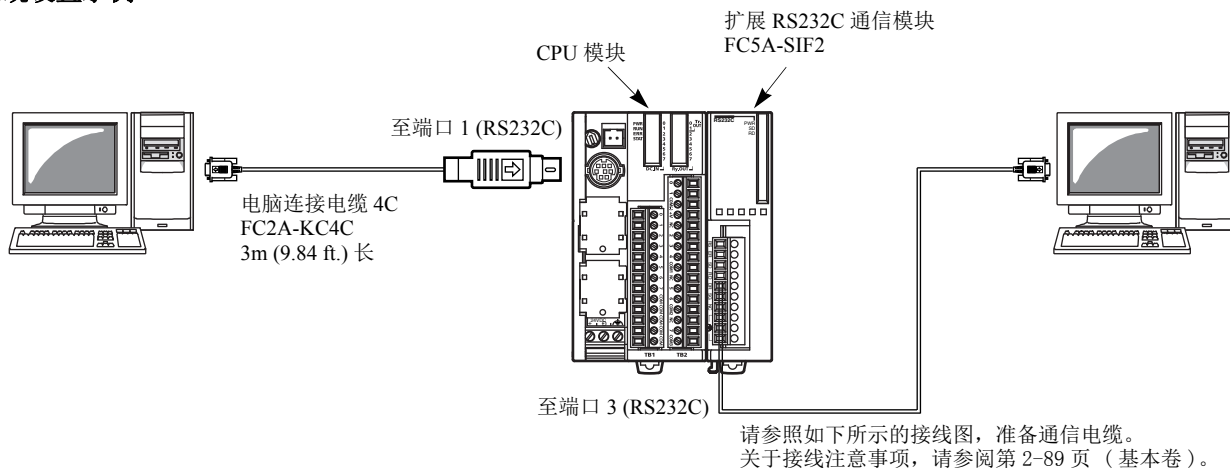
当安装了五个扩展 RS232C/RS485 通信模块和两个 I/O 模块, 通信端口编号的分配如下所示。



### 计算机连接通信

计算机连接通信可以使用电脑上的 WindLDR 与 CPU 模块连接，执行维护操作，例如上传 / 下载用户程序、启动 / 停止 PLC、监控 PLC 状态和读 / 写设备值。当在计算机连接系统的 CPU 模块上安装扩展 RS 232C/RS 485 通信模块时，必须使用计算机连接通信功能，运行时间程序下载除外。有关计算机连接通信的详细信息，请参阅第 4-1 页 (基本卷) 和第 21-1 页。

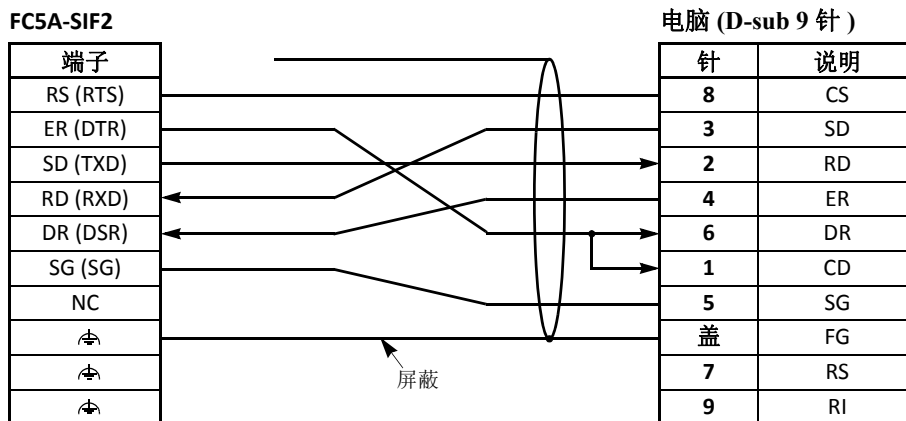
#### 系统设置示例



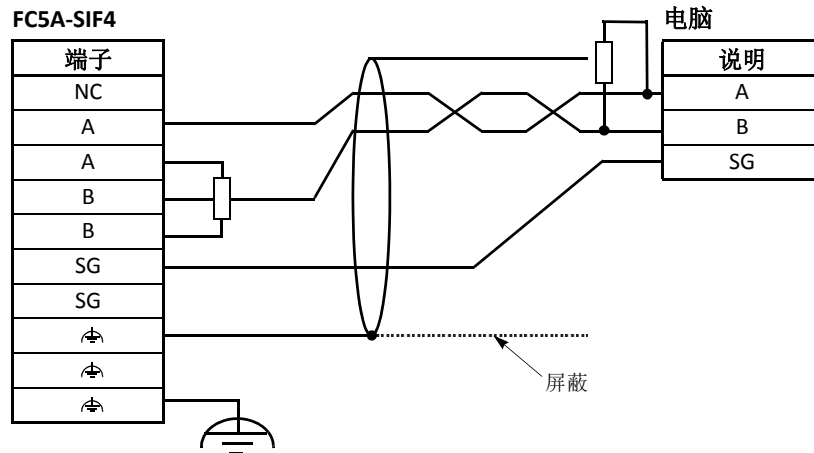
**注释：** 扩展 RS232C/RS485 通信模块上不能使用运行时程序下载功能。

**注释：** 使用扩展 RS485 通信模块时，应采用第三方提供的 USB/RS485 转换器连接电脑与 CPU 模块。

#### 电缆连接和插针 (FC5A-SIF2 扩展 RS232C 通信模块)



## 电缆连接和插针 (FC5A-SIF4 扩展 RS485 通信模块)



## 扩展 RS232C/RS485 通信模块通信参数范围

参数	可选范围	默认
通信模式	维护通信	
通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 (注释)	9600
数据长度	7 或 8	7
奇偶校验	无、奇数、偶数	偶数
停止位	1 或 2	1
接收超时 (ms)	10 ~ 2550 (10-ms 递增) (当选择 0 时, 接收超时无效。)	500
网络编号	0 ~ 31	0
通信切换输入	任何输入编号	无效

注释：使用 57600 或 115200 bps 的速度时，需要系统程序版本在 220 或更高的 CPU 模块及 FC5A-SIF2（200 版本或更高）或 FC5A-SIF4。

## 注释：

- 下载或上传用户程序时，请将传输模式设为 ASCII。
- 下载或上传用户程序时，CPU 模块的系统程序版本必须在 220 及以上，同时必须使用 FC5A-SIF4 通信模块。
- 端口 3 至端口 7 上不能使用运行时程序下载功能。

## 25: 扩展 RS232C/RS485 通信

### 通过扩展 RS232C/RS485 通信模块的计算机连接通信

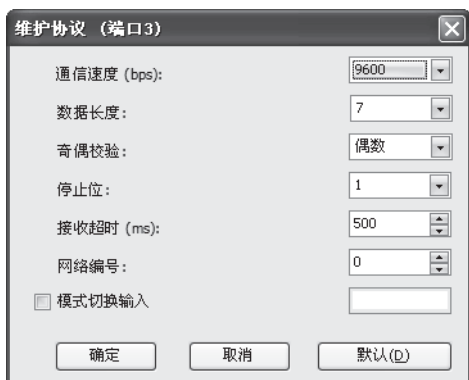
通过扩展 RS232C/RS485 通信模块不能上传或下载用户程序，因此，在 1:1 计算机连接系统中通过端口 1 或 2 下载用户程序（请参阅第 4-1 页（基本卷）），以实现使用扩展 RS232C/RS485 通信模块执行计算机连接通信。将用户程序下载到 CPU 模块后，可以使用与扩展 RS232C/RS485 通信模块连接的电脑上的 WindLDR 对 CPU 模块上的操作状态进行监控。

使用扩展 RS232C/RS485 通信模块进行计算机连接通信的操作程序如下：

1. 使用计算机连接电缆 4C (FC2A-KC4C) 连接电脑到 CPU 模块上的通信端口 1 或 2 (RS232C)。
2. 从 WindLDR 菜单栏选择**设置** > **功能设置** > **通信端口**，此时出现通信端口的“功能设置”对话框。
3. 在“端口 1”到“端口 7”的“通信模式”下拉列表中，选择**维护协议**。



4. 在显示的通信参数对话框中更改设置。



通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
数据长度	7 或 8
奇偶校验	无、奇数、偶数
停止位	1 或 2
接收超时 (ms)	10 ~ 2550 (10-ms 递增) (当选择 0 时，接收超时无效。)
网络编号	0 ~ 31
通信切换输入	任何输入编号

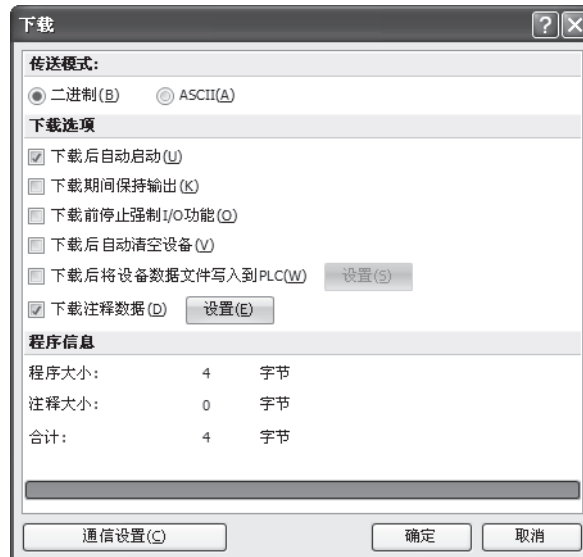
**注释：**当指定并打开通信切换输入时，将激活所选通信参数。当更改通信参数并未指定通信切换输入时，在下载用户程序时更改后的通信参数将立即生效。

5. 点击**确定**按钮保存更改。



此时，“通信设置”对话框关闭，“通信端口”页面处于活动状态。

6. 点击**确定**按钮保存更改后的功能设置。  
关闭功能设置对话框并激活梯形图编辑画面。  
然后，通过端口 1 或 2 下载用户程序到 CPU 模块。
7. 从 WindLDR 菜单栏选择**联机 > 传送 > 下载**。  
此时显示“下载”对话框。



8. 点击**确定**按钮。  
用户程序将被下载到 CPU 模块。  
**注释：** 下载程序时，功能设置中的所有值和选项也将被下载到 CPU 模块。
9. 将电脑与扩展 RS232C/RS485 通信模块上的通信端口 1 ~ 3 连接。  
有关端子排列和接线图，请参阅第 2-89 页和第 2-90 页（基本卷）。
10. 打开电脑上的 WindLDR 并连接扩展 RS232C/RS485 通信模块。
11. 从 WindLDR 菜单栏选择**联机 > 监控 > 监控**。  
显示监控画面，在该画面中可以监控 MicroSmart 运行状态和更改设备值。

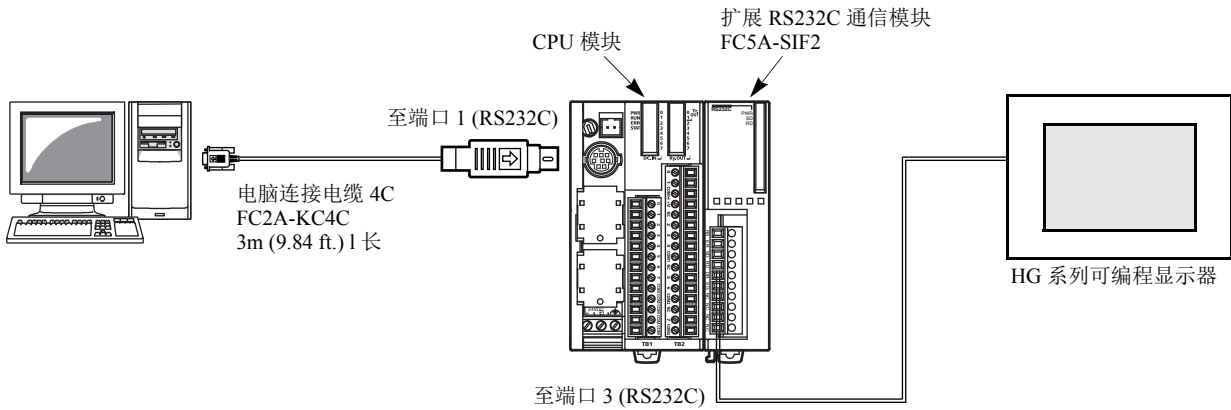
## 25: 扩展 RS232C/RS485 通信

### 可编程显示器通信

使用扩展 RS232C/RS485 通信模块，MicroSmart 可以与 IDEC 的 HG 系列可编程显示器进行通信。使用用户准备好的通信电缆将 HG 系列可编程显示器与扩展 RS232C/RS485 通信模块连接。

有关通信设置和规格，请参阅 HG 系列可编程显示器的用户手册。

#### 系统设置示例

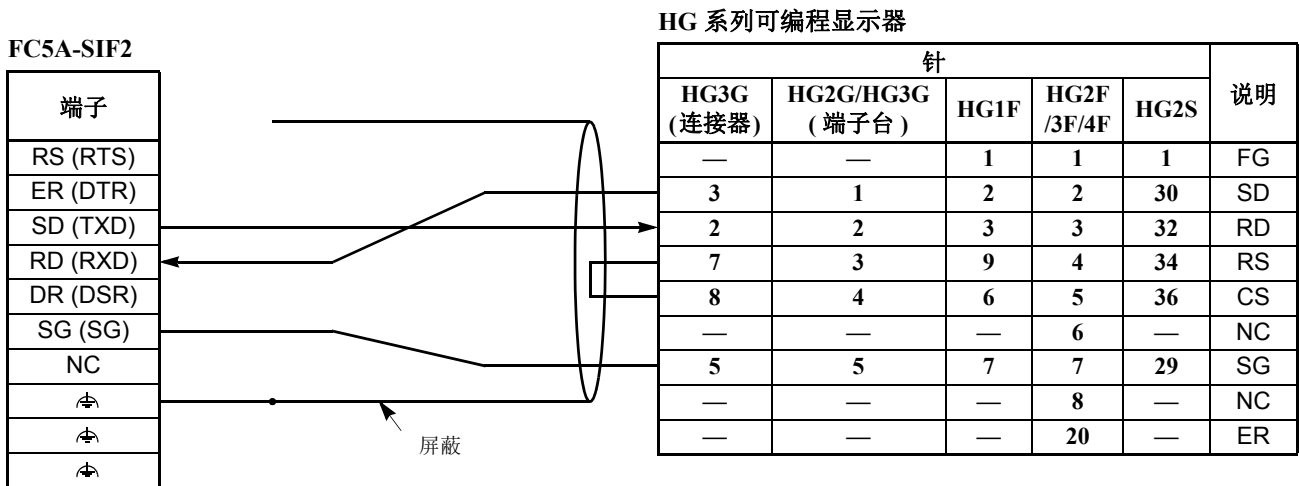


请参照如下所示的接线图，准备通信电缆。  
关于接线注意事项，请参阅第 2-89 页（基本卷）。

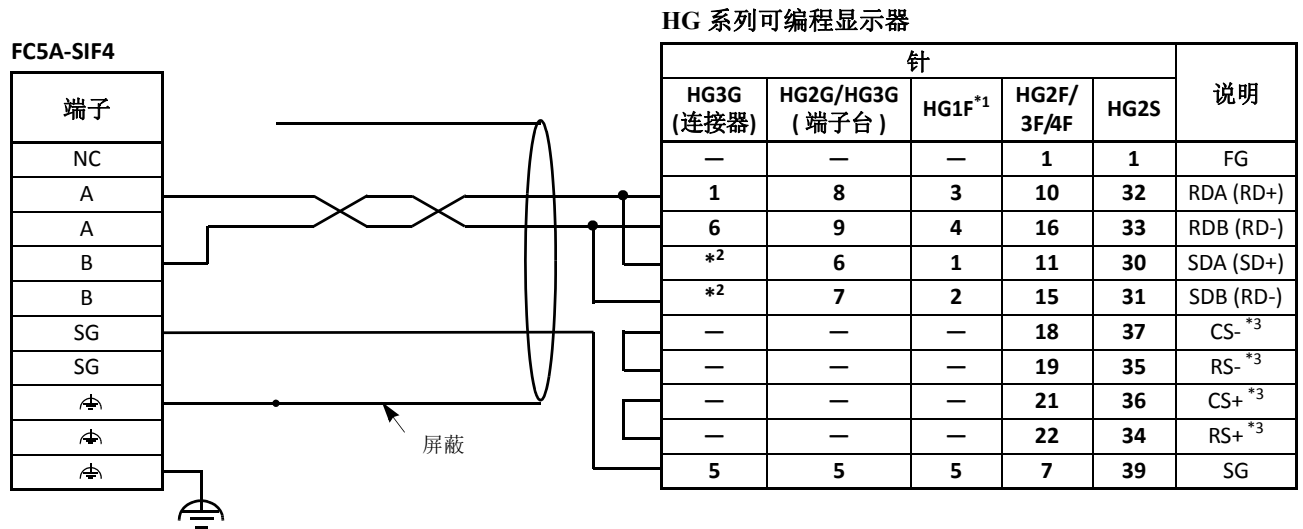
**注释：**用户程序不能通过扩展 RS232C 通信模块上传和下载。

适用的可编程显示器	HG2G, HG1F, HG2F, HG3F, HG4F, HG2S (可以使用适用于端口 1 ~ 7 的 HG 系列可编程显示器。)
-----------	------------------------------------------------------------------------

#### 电缆连接和插针 (RS232C)



## 电缆连接和插针 (RS485)



*1: 接线端编号介绍。

*2: 由于 HG3G 仅使用 RDA 和 RDB 进行 RS-485 (422) 两线制方式通信, 因此无需连接 SDA 和 SDB。

*3: 当使用 HG2G 或 HG1F 时, CS-、RS-、CS+ 与 RS+ 均不用接线。禁用所连接的 HG 系列操作界面上的硬件流量控制 (功能)。

## 适用于 HG 系列可编程显示器的电缆连接器

可编程显示器	连接器电缆
HG3G	D-sub 9 针插头型连接器 螺旋式接线盒套管
HG2G	螺旋式接线盒套管
HG1F	D-sub 9 针插头型连接器 (RS232C) 螺旋式接线盒套管 (RS485)
HG2F, HG3F, HG4F	D-sub 25 针插头型连接器
HG2S	D-sub 37 针插头型连接器

## 25: 扩展 RS232C/RS485 通信

扩展 RS232C/RS485 通信模块通信参数范围

参数	可选范围	默认
通信模式	维护通信	
通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 (注释)	9600
数据长度	7 或 8	7
奇偶校验	无、奇数、偶数	偶数
停止位	1 或 2	1
接收超时 (ms)	10 ~ 2550 (10-ms 递增) (当选择 0 时, 接收超时无效。)	500
网络编号	0 ~ 31	0
通信切换输入	任何输入编号	无效

**注释:** 使用 57600 或 115200 bps 的速度时, 需要系统程序版本在 220 或更高的 CPU 模块及 FC5A-SIF2 (200 版本或更高) 或 FC5A-SIF4。

### 通过扩展 RS232C/RS485 通信模块的可编程显示器通信

不能通过扩展 RS232C/RS485 通信模块上传或下载用户程序, 因此, 在 1:1 计算机连接系统中通过端口 1 或 2 下载用户程序 (请参阅第 25-6 页), 以实现使用扩展 RS232C/RS485 通信模块执行可编程显示器通信。将用户程序下载到 CPU 模块后, CPU 模块可以通过扩展 RS232C/RS485 通信模块与可编程显示器进行通信。

可编程显示器通信的操作步骤如下所示:

1. 更改功能设置, 并通过 CPU 模块上的通信端口 1 或 2 (RS232C) 下载用户程序。请参阅第 25-4 页和第 25-5 页上的操作步骤 1 ~ 8。
2. 连接可编程显示器与扩展 RS232C/RS485 通信模块上的通信端口 3 ~ 7。  
有关端子参数和接线图, 请参阅第 2-89 页和第 2-90 页 (基本卷)。

**注释:** 当可编程显示器上显示数据刷新周期缓慢时, 请参阅第 25-13 页上的“通信反应缓慢”。

## 用户通信

通过扩展 RS232C/RS485 通信模块，用户通信功能可以用于与电脑、打印机和条形码读取器间的通信。有关用户通信功能的详细信息，请参阅第 10-1 页（基本卷）。

### 扩展 RS232C 通信模块通信参数范围

参数	可选范围	默认
通信模式	用户通信	
通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 (注释)	9600
数据长度	7 或 8	7
奇偶校验	无、奇数、偶数	偶数
停止位	1 或 2	1
接收超时 (ms)	10 ~ 2550 (10-ms 递增) (当选择 0 时，接收超时无效。)	500

**注释：**使用 57600 或 115200 bps 的速度时，需要系统程序版本在 220 或更高的 CPU 模块及 FC5A-SIF2（200 版本或更高）或 FC5A-SIF4。

### 通过扩展 RS232C 通信模块的用户通信

本节描述通过扩展 RS232C 通信模块发送数据到打印机的用户通信示例。将用户程序下载到 CPU 模块后，CPU 模块可以通过扩展 RS232C 通信模块与打印机进行通信。

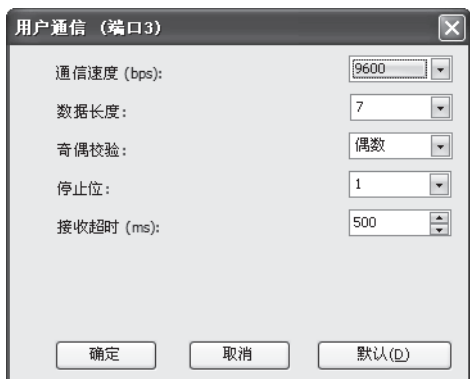
用户通信的操作步骤如下所示：

1. 更改功能设置，并通过 CPU 模块上的通信端口 1 或 2 (RS232C) 下载用户程序。请参阅第 25-4 页和第 25-5 页上的操作步骤 1 ~ 8。
2. 连接打印机与扩展 RS232C 通信模块上的通信端口 3 ~ 7。  
有关端子参数和接线图，请参阅第 2-89 页（基本卷）。
3. 从 WindLDR 菜单栏选择**设置 > 功能设置 > 通信端口**，此时出现通信端口的“功能设置”对话框。
4. 在“端口 3”到“端口 7”的“通信模式”下拉列表中，选择**用户协议**。



## 25: 扩展 RS232C/RS485 通信

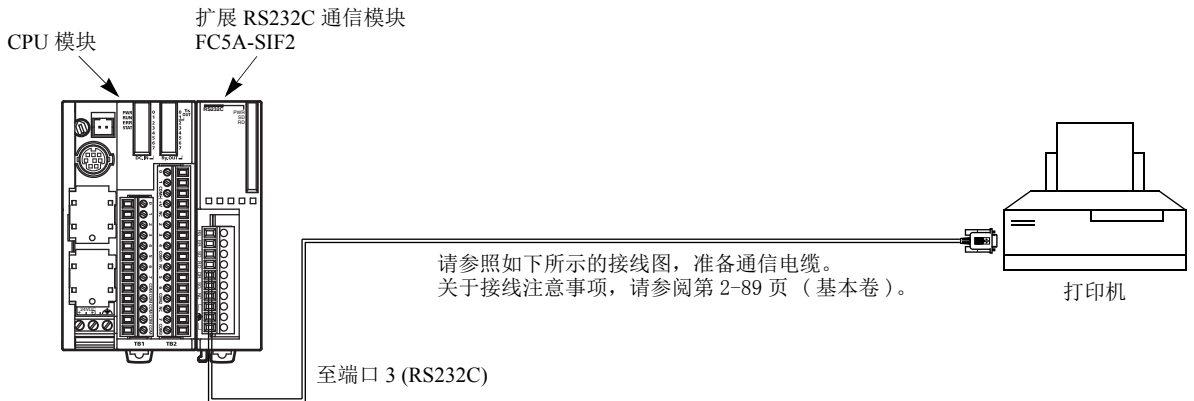
5. 显示通信参数对话框。更改设置适于远程设备的通信参数。请参阅远程设备的用户手册。



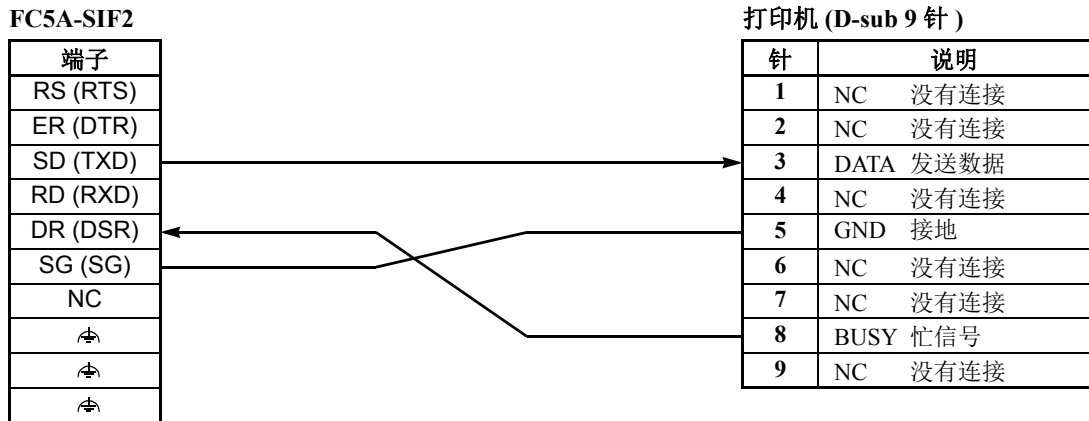
通信速度 (bps)	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
数据长度	7 或 8
奇偶校验	无、奇数、偶数
停止位	1 或 2
接收超时 (ms)	10 ~ 2540 (10-ms 递增) (当选择 0 或 2550, 接收超时无效。)

6. 点击**确定**按钮保存更改。  
关闭通信参数对话框并激活通信页面。
7. 点击**确定**按钮保存更改后的功能设置。  
关闭功能设置对话框并激活梯形图编辑画面。
8. 通过 CPU 模块上的通信端口 1 或 2(RS232C) 下载用户程序到 CPU 模块。  
有关梯形图程序控制打印机, 请参阅第 25-12 页。

连接打印机的系统设置



电缆连接和插针



BUSY 终端的名称因打印机而异, 如 DTR。该终端的功能是无论打印机是否已准备好打印数据都将信号发送至远程设备。因为该信号的操作根据打印机的不同而不同, 所以请在连接电缆前确认操作。

**注意** • 请勿将任何电线连接至NC(没有连接)针脚; 否则MicroSmart和打印机可能无法正常工作并受到损坏。

操作说明

每分钟都会打印计数器 C2 的数据和寄存器 D30。打印输出示例显示在右侧。

编写特殊数据寄存器

特殊数据寄存器 D8105 用于监控 BUSY 信号和控制打印数据的发送。

特殊 DR	值	说明
D8105	3 (011)	当 DSR 打开 (不忙) 时, CPU 将发送数据。 当 DSR 关闭 (忙) 时, CPU 将停止发送数据。 如果关闭时间超过限制 (约 5 秒钟), 将出现发送忙超时错误, 且不再发送剩余数据。发送状态寄存器存储错误代码。请参阅第 10-11 页和第 10-32 页 (基本卷)。

打印输出示例

```

--- PRINT TEST ---

11H 00M

CNT2...0050
D030...3854

--- PRINT TEST ---

11H 01M

CNT2...0110
D030...2124
    
```

MicroSmart 监控 DSR 信号以避免打印机的接收缓冲区溢出。特殊数据寄存器编号和值 (如上所示) 从端口 3。有关 DSR 信号, 请参阅第 10-36 页 (基本卷)。

## 25: 扩展RS232C/RS485 通信

### 设置通信参数

设置通信参数与打印机的参数匹配。请参阅第 25-10 页。有关打印机通信设置的详细信息，请参阅打印机的用户手册。示例如下：

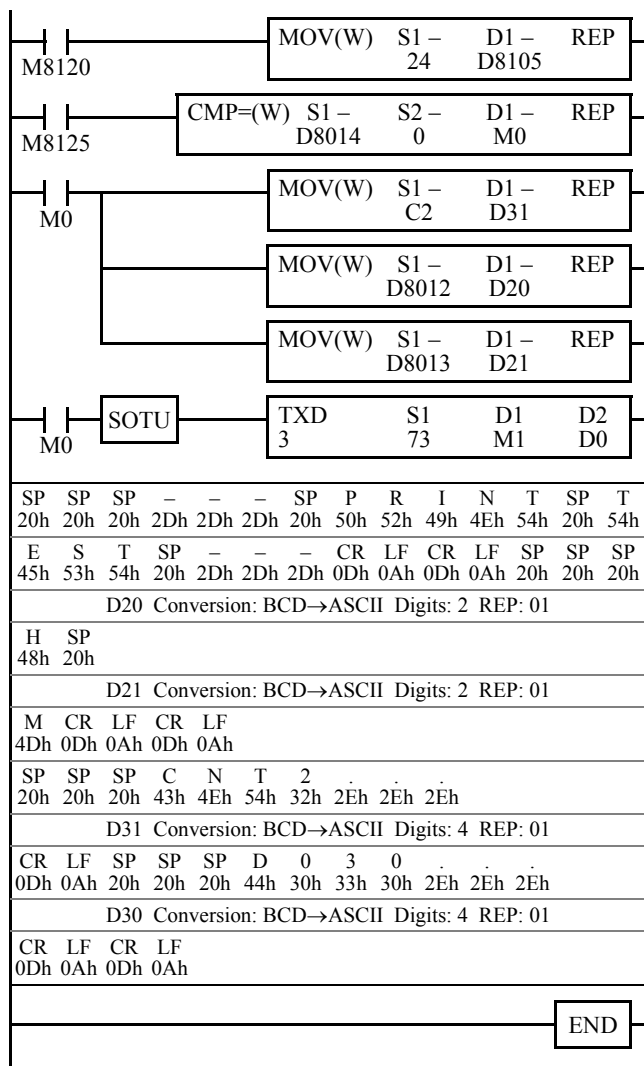
#### 通信参数：

通信速度            9600 bps  
 数据长度            8  
 奇偶校验            无  
 停止位               1

**注释：**接收超时值用于用户通信模式下的 RXD 指令中。因为该示例仅使用 TXD 指令，所以接收超时值不受影响。

### 梯形图

使用 CMP=(比较等于) 指令将特殊数据寄存器 D8014 中的第二个数据与 0 进行比较。每次条件满足时，将执行 TXD3 指令以发送 C2 和 D30 数据至打印机。此示例程序中省略了计数器 C2 的计数电路。



M8120 是初始化脉冲特殊内部继电器。  
 24 → D8105 可启用 DSR 选项进行忙控制。  
 M8125 是运行中输出特殊内部继电器。  
 CMP=(W) 将 D8014 第二个数据与 0 进行比较。  
 当 D8014 的第二个数据等于 0 时，M0 将打开。  
 计数器 C2 当前值移至 D31。  
 D8012 小时数据移至 D20。  
 D8013 分钟数据移至 D21。  
 当 M0 被打开，执行 TXD3 以通过 RS232C 端口 2 发送 73 字节至打印机。

对 D20 小时数据进行 BCD 码 → ASCII 码转换，并发送 2 位数字。  
 对 D21 分数据进行 BCD 码 → ASCII 码转换，并发送 2 位数字。  
 对 D31 计数器 C2 数据进行 BCD 码 → ASCII 码转换，并发送 4 位数字。  
 对 D30 数据进行 BCD 码 → ASCII 码转换，并发送 4 位数字。



## 故障排除

本节描述了确定故障原因的步骤，以及操作扩展 RS232C/RS485 通信模块遇到问题时要采取的措施。当遇到问题时，确认问题点和解决方案，如果不能解决请致电 IDEC 请求帮助。

### PWR ( 电源 ) LED 不能持续

问题点	解决方案	页面
CPU 模块是否供电？	供电给 CPU 模块。	基本卷 3-18, 基本卷 3-19
电压是否正确？	提供额定电压。 集成型： 100-240V AC 或 24V DC 超薄型： 24V DC	基本卷 3-18, 基本卷 3-19

### 扩展 RS232C/RS485 通信模块不能与 WindLDR 通信

问题点	解决方案	页面
通信电缆连接是否正确？	正确连接通信电缆。	基本卷 2-89, 25-3
CPU 模块上的 PWR LED 是否打开？	请参阅“PWR ( 电源 ) LED 不能持续”	25-13
FC5A-SIF4 模块上的电源指示灯是否闪烁？	CPU 模块上通上额定电压	基本卷 3-1
WindLDR 的通信设置是否与扩展通信端口一致？	为 WindLDR 和扩展通信模块端口设置相同的通信参数。	25-4
CPU 模块系统程序版本是否适用于扩展 RS232C/RS485 通信模块？	将 CPU 模块系统程序版本升级到 110 或以上版本以使用 FC5A-SIF2 通信模块，或升级到 220 或以上版本以使用 FC5A-SIF4 通信模块。下载或上传用户程序时，需将系统程序版本升级到 220 或以上版本。	基本卷 A-9

### 扩展 RS232C/RS485 通信模块不能与可编程显示器通信

问题点	解决方案	页面
通信电缆连接是否正确？	正确连接通信电缆。	基本卷 2-89, 25-7
CPU 模块上的 PWR LED 是否打开？	请参阅“PWR ( 电源 ) LED 不能持续”	25-13
FC5A-SIF4 模块上的电源指示灯是否闪烁？	CPU 模块上通上额定电压	基本卷 3-1
WindLDR 的通信设置是否与扩展通信端口一致？	为 WindLDR 设置与扩展通信模块端口相同的通信参数。	25-6
CPU 模块系统程序版本是否适用于扩展 RS232C/RS485 通信模块？	将 CPU 模块系统程序版本升级到 110 或以上版本以使用 FC5A-SIF2 通信模块，或升级到 220 或以上版本以使用 FC5A-SIF4 通信模块。	基本卷 A-9

### 通信反应缓慢

问题点	解决方案	页面
通信速度是否设置为固有值？	将通信速度设置为求定值。	25-3, 25-8, 25-9
在用户程序中是否使用 COMRF 指令？	在用户程序中使用 COMRF 指令。	11-13

### 在用户通信中数据完全不能被发送

问题点	解决方案	页面
通信电缆连接是否正确？	确保正确接线。	基本卷 2-89, 25-9
远程端子的通信设置是否与扩展 RS232C/RS485 通信端口一致？	将扩展通信端口的通信参数设为远程终端一样。	25-9
CPU 模块系统程序版本是否适用于扩展 RS232C/RS485 通信模块？	将 CPU 模块系统程序版本升级到 110 或以上版本以使用 FC5A-SIF2 通信模块，或升级到 220 或以上版本以使用 FC5A-SIF4 通信模块。	基本卷 A-9
在 TXD 指令中是否指定正确的端子编号？	在 TXD 指令中指定正确的端子编号	基本卷 10-7
TXD 指令的起始输入是否打开？	打开 TXD 指令的起始输入。	25-12
CPU 模块上的 PWR LED 是否打开？	请参阅“PWR ( 电源 ) LED 不能持续”	25-13
FC5A-SIF4 模块上的电源指示灯是否闪烁？	CPU 模块上通上额定电压	基本卷 3-1

## 25: 扩展RS232C/RS485 通信

### 在用户程序中数据不能被正确发送

问题点	解决方案	页面
外部设备和扩展通信端口的通信设置是否一致?	为扩展通信端口设置与外部设备相同的通信参数。	25-10
一个数据寄存器是否被重复指定为目标设备 D2 ( 传送状态 )?	将重复的设备更改到其他数据寄存器。	基本卷 10-6
是否同时打开输入至 5 个以上 TXD 指令?	修改程序确保不再继续同时打开输入至 5 个以上 TXD 指令。	基本卷 10-6
远程端子的忙信号的持续时间是否低于 5 秒?	确保远程端子的忙信号不超过 5 秒。	基本卷 10-38
是否确认 TXD 指令的源 1 设备?	确保指定为源 1 设备发送数据正确。	基本卷 10-7
FC5A-SIF4 模块上的电源指示灯是否闪烁?	CPU 模块上通上额定电压	基本卷 3-1

### 在用户程序中数据完全不能被接收

问题点	解决方案	页面
通信电缆连接是否正确?	确保正确接线。	基本卷 2-89, 25-11
远程端子的通信设置是否与扩展 RS232C/RS485 通信端口一致?	为扩展 RS232C 通信端口设置与远程端子相同的通信参数。	25-9
CPU 模块系统程序版本是否适用于扩展 RS232C/RS485 通信模块?	将 CPU 模块系统程序版本升级到 110 或以上版本以使用 FC5A-SIF2 通信模块, 或升级到 220 或以上版本以使用 FC5A-SIF4 通信模块。	基本卷 A-9
在 RXD 指令中是否指定正确的端子编号?	在 RXD 指令中指定正确的端子编号。	基本卷 10-16
RXD 指令的起始输入是否打开?	打开 RXD 指令的起始输入。	基本卷 10-39
CPU 模块上的 PWR LED 是否打开?	请参阅“PWR ( 电源 ) LED 不能持续”	25-13
FC5A-SIF4 模块上的电源指示灯是否闪烁?	CPU 模块上通上额定电压	基本卷 3-1

### 在用户通信中数据不能被正确接收

问题点	解决方案	页面
外部设备和扩展通信端口的通信设置是否一致?	为扩展通信端口设置与外部设备相同的通信参数。	25-9
一个数据寄存器是否被重复指定为目标设备 D2 ( 接收状态 )?	将重复的设备更改到其他数据寄存器。	基本卷 10-15
是否在 RXD 指令中指定起始符? 是否同时打开输入至 5 个以上 RXD 指令?	修改程序确保不再继续同时打开输入至 5 个以上 RXD 指令。	基本卷 10-15
是否确认输入数据的格式?	确保 RXD 指令的接收格式与输入数据的格式相匹配。	基本卷 10-16
使用 WindLDR 设置的接收超时值是否正确?	确保接收超时值大于输入数据的字符间隔。	基本卷 10-5
是否确认 RXD 指令的源 1 设备?	确保指定为源 1 设备接收数据正确。	基本卷 10-16
FC5A-SIF4 模块上的电源指示灯是否闪烁?	CPU 模块上通上额定电压	基本卷 3-1

如果在完成上述故障排除步骤后通信仍不稳定, 则应检查并调整以下各点, 并确认通信稳定性是否有改进。

- 使用较低的通信速率。
- 增加传输等待时间。
- 增加重试次数。
- 增加超时设定。
- 在阶梯程序中添加 COMRF 指令。
- 在“功能域设置”中, 在端口 3 至端口 7 的“通信刷新”下选择“每 10 ms”。

# 索引

- #
  - 1:N 计算机连接 21-1
  - 100 毫秒
    - 双定时器 15-1
  - 10 毫秒
    - 双定时器 15-1
  - 1 毫秒双定时器 15-1
  - 1 秒
    - 双定时器 15-1
- A**
  - A/B 从机 24-4
  - ACOS 17-7
  - ADD 5-1
  - advanced instruction
    - DEC 5-13
    - RNDM 5-18
    - SUM 5-15
  - ALT 8-18
  - ANDW 6-1
  - APF/ 不是 APO 24-25
  - ASCII
    - 到 BCD 8-12
    - 到 HEX 8-7
  - ASI 命令 24-30
  - ASIN 17-6
  - AS-Interface 24-1
    - 标准电缆 24-3
    - 电缆接线 24-6
    - 电源 24-3
    - 设备 24-18
    - 设备设备地址 24-19
    - 系统安装 24-6
    - 总线
      - 布局 and 最大长度 24-5
      - 周期时间 24-5
  - AT 14-12
    - 常规命令模式 22-2, 22-5
    - 命令
      - 结果代码 22-3
      - 执行 22-2
      - 字符串 22-3
  - ATAN 17-8
  - ATOB 8-12
  - ATOH 8-7
  - ATZ 22-2, 22-4, 22-6
  - AVRG 12-7
  - Auto_Address_Assign 24-24
  - Auto_Address_Available 24-25
  - 按钮操作 23-6, 24-14
  - 按钮和 LED 指示灯 24-14
  - 安装
    - 位置通信信息块 24-40
- B**
  - BCD
    - 到 ASCII 8-9
    - 到 HEX 8-3
    - 左移 7-5
  - BCDLS 7-5
  - BCNT 8-17
  - BMOV 3-9
  - BMOV/WSFT 执行标记 M8024 3-9, 7-7
  - BTOA 8-9
  - BTOH 8-3
  - 保护模式 24-15
  - 备份
    - 持续时间时钟盒 9-8
  - 本地模式 24-15
  - 比较
    - 不等于 4-1
    - 大于 4-1
    - 大于或等于 4-2
    - 等于 4-1
    - 小于 4-1
    - 小于或等于 4-1
  - 比较结果
    - M8150、M8151、M8152 4-2, 4-6
  - 编码 8-15
  - 扁平电缆 24-3
  - 编写
    - 特殊数据寄存器 25-11
  - 编址工具 24-4
  - 表
    - 高级指令 2-1
  - 标签 11-1
    - 调用 11-3
    - 返回 11-3
    - 跳转 11-1
  - 标准
    - 从机 24-4
    - 电缆 24-3
  - 标准保护
    - 模式 24-15
    - 数据交换关闭 24-15
    - 脱机 24-15
  - 拨号 22-2
    - 电话号码 22-4
  - 步进响应方式 14-12
  - 布局 24-5
- C**
  - CDI 24-27
  - CMP< 4-1
  - CMP<= 4-1
  - CMP<> 4-1
  - CMP= 4-1
  - CMP> 4-1
  - CMP>= 4-2
  - COMRF 11-13
    - execution time 11-13
  - Config_OK 24-24
  - COS 17-4
  - CVDT 8-19
  - CVXTY 12-2
  - CVYTX 12-3
  - 参数 24-4
    - 图像 (PI) 24-28
  - 操作
    - 步骤

- 调制解调器模式 22-11
    - 基本 24-6
    - 模式 24-15
  - 操作变量 14-16
  - 操作码 2-7
  - 插针 22-1
  - 长按 24-14
  - 常用对数 18-2
  - 超低安全电压 24-3, 24-7
  - 乘 5-1
  - 程序分支
    - 使用定时器指令 11-2
    - 使用 SOTU/SOTD 指令 11-2
    - 指令 11-1
  - 程序分支使用的 SOTU/SOTD 指令 11-2
  - 除 5-1
  - 处理时间 24-18, 24-30
  - 初始化字符串 22-2, 22-3, 22-6
    - 命令 22-8
  - 初始模式 22-2, 22-3
  - 从机
    - 地址 24-4
    - 发送数据 24-39, 24-40
    - 接收数据 24-39, 24-40
    - 扩展能力 24-4
    - 列表信息 24-26
    - 配置文件 24-27
      - 模拟量 24-22
    - 识别 24-4
      - 信息 24-27
  - 从机数量和 I/O 点数 24-5
  - 存储定时器 / 计数器当前值 3-16
  - 错误 24-13
    - 代码 24-35, 24-36, 24-37, 24-38
    - 信息 24-38
- D**
- Data_Exchange_Active 24-25
  - DEC 5-13
  - DECO 8-16
  - decrement 5-13
  - DEG 17-2
  - DGRD 10-3
  - DI 11-7
  - DISP 10-1
  - DIV 5-1
  - DTCB
    - 数据组合 8-22
  - DTDV (数据分割) 8-21
  - DTIM 15-1
  - DTMH 15-1
  - DTML 15-1
  - DTMS 15-1
  - 打印机 25-11
  - 地址 LED 24-16
    - 和 I/O LED 24-17
  - 电话号码 22-3, 22-4
  - 电缆 24-3, 25-2, 25-3, 25-6, 25-7, 25-11
    - 调制解调器 1C 22-1
    - 计算机连接 4C 24-6
    - 扩展通信
      - RS232C 25-2
      - RS485 25-3
      - RS232C 21-1
  - 电源 18-4, 24-3, 24-7
    - 接线图 24-7
  - 调整
    - 扫描时间 10-4
    - 使用用户程序的时钟 9-7
    - 时钟盒准确性 9-8
  - 调制解调器
    - 电缆 1C 22-1
    - 模式 22-1
      - 状态 22-3
      - 状态数据寄存器 22-7
    - 通信 22-10
  - 定时器
    - 程序分支使用的指令 11-2
    - 或计数器
      - 作为目标设备 2-7
      - 作为源设备 2-7
  - 读取
    - 时间数字开关数据 10-3
  - 短按 24-14
  - 断开
    - 模式 22-2, 22-5
    - 线路 22-2
  - 对数 / 乘方
    - 指令 18-1
- E**
- EI 11-7
  - ENCO 8-15
  - execution
    - time COMRF 11-13
  - EXP 18-3
- F**
- FIEX 19-3
  - FIFO 格式 19-1
  - FIFO 19-1
  - FRQF 11-12
  - 发送数据 24-39, 24-40
  - 反向
    - 控制动作 14-14
  - 反余弦 17-7
  - 反正切 17-8
  - 反正弦 17-6
  - 分配通信端口编号 25-1
  - 分配一个从机地址 24-9
  - 复位
    - 调制解调器 22-4, 22-6
- G**
- 高级指令 2-1
    - ACOS 17-7
    - ADD 5-1
    - ALT 8-18
    - ANDW 6-1
    - ASIN 17-6
    - ATAN 17-8
    - ATOB 8-12
    - ATOH 8-7
    - AVRG 12-7
    - BCDLS 7-5
    - BCNT 8-17
    - BMOV 3-9
    - BTOA 8-9
    - BTOH 8-3
    - 表 2-1

- CMP< 4-1  
 CMP<= 4-1  
 CMP<> 4-1  
 CMP= 4-1  
 CMP> 4-1  
 CMP>= 4-2  
 COMRF 11-13  
 COS 17-4  
 CVDT 8-19  
 CVXTY 12-2  
 CVYTX 12-3  
 DECO 8-16  
 DEG 17-2  
 DGRD 10-3  
 DI 11-7  
 DISP 10-1  
 DIV 5-1  
 DTCB 8-22  
 DTIM 15-1  
 DTMH 15-1  
 DTML 15-1  
 DTMS 15-1  
 EI 11-7  
 ENCO 8-15  
 EXP 18-3  
 FIEX 19-3  
 FIFO 19-1  
 FRQRF 11-12  
 HOUR 20-11  
 HSCRF 11-11  
 HTOA 8-5  
 HTOB 8-1  
 HTOS 20-9  
 IBMV 3-10  
 IBMVN 3-12  
 ICMP>= 4-6  
 IMOV 3-6  
 IMOVN 3-8  
 IOREF 11-9  
 结构 2-7  
 LABEL 11-1  
 LC< 4-8  
 LC<= 4-8  
 LC<> 4-8  
 LC= 4-8  
 LC> 4-8  
 LC>= 4-8  
 LCAL 11-3  
 LJMP 11-1  
 LOG10 18-2  
 LOGE 18-1  
 LRET 11-3  
 MOV 3-1  
 MOVN 3-5  
 MUL 5-1  
 NRS 3-14  
 NSET 3-13  
 ORW 6-1  
 PID 14-1  
 POW 18-4  
 PULS1 13-2  
 PULS2 13-2  
 PULS3 13-2  
 PWM1 13-8  
 PWM2 13-8  
 PWM3 13-8  
 RAD 17-1  
 RAMP1 13-14  
 RAMP2 13-14  
 ROOT 5-14  
 ROTL 7-8  
 ROTR 7-10  
 RUNA READ 16-2  
 RUNA WRITE 16-3  
 SFTL 7-1  
 SFTR 7-3  
 SIN 17-3  
 STO 20-10  
 STPA READ 16-4  
 STPA WRITE 16-5  
 SWAP 8-23  
 SUB 5-1  
 适用的 CPU 模块 2-4  
 数据类型 2-8  
 输入条件 2-7  
 TADD 20-1  
 TAN 17-5  
 TCCST 3-16  
 TTIM 15-3  
 WKTBL 9-2  
 WKTIM 9-1  
 WSFT 7-7  
 XORW 6-1  
 XYFS 12-1  
 ZRN1 13-26  
 ZRN2 13-26  
 ZRN3 13-26  
 高级指令的结构 2-7  
 高速计数器  
   刷新 11-11  
 格式编号 12-1, 12-2, 12-3  
 更改从机 0 的 ID1 码 24-29  
 功能设置 24-8  
 固定  
   参数 (PP) 24-29  
   配置数据 (PCD) 24-28  
 故障排除 25-13  
   调制解调器通信 22-14  
 规格  
   Modbus TCP 从机通信 23-6  
   Modbus TCP 主机通信 23-2
- ## H
- HEX  
   到 ASCII 8-5  
   到 BCD 8-1  
 HMS → 秒 20-9  
 HOUR 20-11  
 HSCRF 11-11  
 HTOA 8-5  
 HTOB 8-1  
 HTOS 20-9  
 HW 系列数字量 I/O 数据分配 24-40  
 弧度 17-1  
 或 6-1

- J**
- I/O
    - 代码 24-4
    - 数据 24-20
    - 刷新 11-9
  - IBMV 3-10
  - IBMVN 3-12
  - 基本
    - 指令 1-1
  - ICMP>= 4-6
  - ID1 码 24-4
    - 从机 0 的 ID1 码 24-29
  - ID2 码 24-4
  - IDI 24-20
  - ID 码 24-4
  - IMOV 3-6
  - IMOVN 3-8
  - IOREF 11-9
  - 计算机连接
    - 通信 21-1, 25-2
  - 计算机连接电缆 4C 24-6
  - 加 5-1
  - 减 5-1
  - 间隔比较大于或等于 4-6
  - 间接
    - 求反传送 3-8
    - 位求反传送 3-12
    - 位传送 3-10
    - 传送 3-6
  - 监控
    - PLC 状态 21-3
  - 监控 AS-Interface 从机 24-37
  - 监控数字量 I/O 和更改输出状态 24-12
  - 减速输入 13-28
  - 角度 17-2
  - 交换 3-15
    - XCHG 3-15
  - 交替输出 8-18
  - 接点比较
    - 不等于 4-8
    - 大于 4-8
    - 大于或等于 4-8
    - 等于 4-8
    - 小于 4-8
    - 小于或等于 4-8
  - 解码 8-16
  - 接收
    - 超时 21-2, 25-3, 25-4, 25-8, 25-9, 25-10
  - 接收数据 24-39, 24-40
  - 进位或借位信号 5-2
  - 禁用
    - 中断 11-7
- K**
- 开始
    - 和结果内部继电器 22-2
  - 可编程显示
    - 通信 25-6
  - 可用从机列表 (LAS) 24-26
  - 控制
    - 寄存器 13-2, 13-9, 13-15, 13-27, 14-3
    - 继电器 14-14
  - 块传送 3-9
  - 扩展能力 24-4
  - 扩展 RS232C 通信 25-1
- L**
- L6 系列数字量 I/O 数据分配 24-39
  - LABEL 11-1
  - LAPP 电缆 24-3
  - LAS 24-26
  - LC< 4-8
  - LC<= 4-8
  - LC<> 4-8
  - LC= 4-8
  - LC> 4-8
  - LC>= 4-8
  - LCAL 11-3
  - LDS 24-26
  - LDS.0 24-24
  - LED 指示灯 24-14, 24-16
  - LJMP 11-1
  - LOG10 18-2
  - LOGE 18-1
  - LPF 24-26
  - LPS 24-27
  - LRET 11-3
  - 连接模式 24-15, 24-25
  - 连接模式协议选择 22-3
  - 连接器插针 22-1
    - 列表
      - 基本指令 1-1
  - 零返回 13-26
  - 逻辑运算指令 6-1
- M**
- Modbus TCP
    - 通信 23-1
    - 主机请求表 23-3
  - Modbus TCP 通信格式 23-9
  - Modbus TCP 主机通信 23-2, 23-5
  - MOV 3-1
  - MOVN 3-5
  - MUL 5-1
  - 脉冲
    - 宽调制 13-8
    - 输出 13-2
    - 指令 13-1
  - 忙
    - 信号 25-11
  - 秒→HMS 20-10
  - 模拟量
    - 从机配置文件 24-22
    - I/O 数据 24-22
    - 输出数据 24-23
    - 输入数据 24-22
  - 模式 9-1
  - 目标设备 2-7
- N**
- N 数据置位 3-13
  - N 数据重复置位 3-14
  - Normal_Operation_Active 24-25
  - NRS 3-14
  - NSET 3-13
- O**
- ODI 24-21
  - Offline_Ready 24-25
  - ORW 6-1
- P**
- PCD 24-28
  - Periphery_OK 24-25
  - PI 24-28

- PID
    - 控制 14-1
    - 指令 14-1
    - 使用注意事项 14-17
  - PLC 状态
    - 监控 21-3
  - POW 18-4
  - PP 24-29
  - PULS1 13-2
  - PULS2 13-2
  - PULS3 13-2
  - PWM1 13-8
  - PWM2 13-8
  - PWM3 13-8
  - 配置 24-25, 24-36
    - 模式 24-15
  - 配置 AS-Interface 主机 24-34
  - 配置从机 24-10
  - 配置文件 24-27
    - 模拟量从机 24-22
  - 频率测量
    - 刷新 11-12
  - 平均化 12-7
- Q**
- 启用
    - 时钟盒调整 9-8
    - 中断 11-7
  - 请求
    - 表 23-4
  - 请求和结果代码 24-31
- R**
- RAD 17-1
  - RAMP1 13-14
  - RAMP2 13-14
  - random 5-18
  - RNDM 5-18
  - ROOT 5-14
  - ROTL 7-8
  - ROTR 7-10
  - RS232C
    - 电缆 21-1
    - 端口
      - 通信协议 22-5
      - 通信适配器 22-1
  - RS232C/RS485 转换器 21-1
  - RS485
    - 通信模块 21-1
    - 通信适配器 21-1
  - RUNA READ 16-2
  - RUNA WRITE 16-3
  - 日历 / 时钟
    - 设置使用
      - WindLDR 9-6
      - 用户程序 9-6
- S**
- SFTL 7-1
  - SFTR 7-3
  - SIN 17-3
  - STOH 20-10
  - STPA READ 16-4
  - STPA WRITE 16-5
  - SWAP
    - 数据交换 8-23
  - SUB 5-1
  - SwitchNet
    - 从机内部继电器 24-41
    - 数据 I/O 端口 24-39
  - SwitchNet 从机的内部继电器 24-41
  - SUM 5-15
  - sum 5-15
  - 三角函数
    - 指令 17-1
  - 扫描时间
    - 调整 10-4
  - 设备 24-18
    - 区域中断 2-10
  - 设备地址 24-19
  - 设备区域中断 2-10
  - 设置
    - 编写计算机连接 21-2
    - DI 或 EI 用 WindLDR 11-7
    - 寄存器和内部继电器 22-9
    - 日历 / 时钟
      - 使用 WindLDR 9-6
      - 使用用户程序 9-6
    - 设置调制解调器模式 22-10
    - 使用 WindLDR Modbus TCP 主机 23-3
    - 使用 WindLDR 设置 Modbus TCP 从机 23-7
    - 使用 WindLDR 设置时钟盒准确性 9-8
    - 数据图像 (CDI) 24-27
  - 设置点 14-16
  - 时间加法 20-1
  - 示教定时器 15-3
  - 十进制值和十六进制存储 2-8
  - 示例程序 24-31
    - 调制解调器初始模式 22-12
    - 调制解调器应答模式 22-13
  - 十六进制存储十进制值 2-8
  - 时序
    - 方向控制
      - 单脉冲输出 13-21
      - 禁用 13-20
      - 双脉冲输出 13-22
    - 归零操作 13-30
    - 禁用脉冲计数 13-6, 13-12
    - 启用脉冲计数 13-5, 13-11
  - 适用的
    - 传感器和致动器 24-1
  - 使用 WindLDR 24-34
  - 时钟
    - 盒
      - 备份持续时间 9-8
      - 调整准确性 9-8
      - 启用调整 9-8
      - 使用用户程序调整 9-7
  - 时钟指令 20-1
  - 输出
    - 点 10-1, 10-3
    - LED 24-16
    - 数据 24-39, 24-40
  - 数据
    - 比较指令 4-1
    - 高级指令的类型 2-8
    - 类型 2-7
    - 相 10-1

- 转换错误 12-3, 12-4
  - 转换指令 8-1
  - 数据寄存器
    - 双字型
      - 设备 2-10
      - 数据传送 3-2
  - 输入
    - 点 10-3
    - 高级指令的条件 2-7
    - LED 24-16
    - 数据 24-39, 24-40
  - 数字
    - 读取 10-3
    - 开关数据读取时间 10-3
  - 数字量
    - I/O 数据分配 24-39
    - 输出数据图像 24-21
    - 输入数据图像 24-20
  - 数字量 I/O 数据分配 24-40
  - 栓锁相 10-1
  - 双 / 示教定时器指令 15-1
  - 双字型
    - 数据寄存器中的设备 2-10
    - 数据寄存器中的数据传送 3-2
  - 四则运算法指令 5-1
- T**
- TADD 20-1
  - TAN 17-5
  - TCCST 3-16
  - TTIM 15-3
  - 台形控制 13-14
  - 特殊内部继电器
    - 的中断状态 11-7
    - 特殊数据寄存器 22-2
    - 用于日历 / 时钟数据 9-7
  - 特殊数据寄存器
    - 特殊数据寄存器 13-4, 13-19, 22-3
    - 用于日历 / 时钟数据 9-6
  - 停止访问
    - 读取 16-4
    - 写入 16-5
  - 通信
    - 参数 25-4, 25-10, 25-12
    - 端口
      - 编号 25-1
      - 设置 21-2, 21-3, 22-10, 23-4
      - 信息块安装位置 24-40
      - 选项卡 21-2, 22-10, 25-9
  - 通信设置 23-4
  - 通信刷新 11-13
  - 脱机 24-25
- W**
- WindLDR
    - 设置
      - DI 或 EI 11-7
      - 调制解调器模式 22-10
      - 计算机连接 21-2
    - Modbus TCP
      - 从机 23-7
      - 主机 23-3
    - 时钟盒准确性 9-8
    - 设置日历 / 时钟 9-6
  - WKTBL 9-2
  - WKTIM 9-1
  - VLSV 24-3, 24-7
  - WSFT 7-7
  - 外部故障从机列表 (LPF) 24-26
  - 维护
    - 协议 25-4
  - 维护通信 21-2
  - 位计数 8-17
  - 文件数据处理指令 19-1
- X**
- XCHG 3-15
  - XORW 6-1
  - 系统
    - 安装 24-6
      - 调制解调器模式 22-1
    - 设置
      - Modbus 通信 23-1
    - 要求 24-2
  - 系统程序
    - 版本 888-II
  - 系统启动时的故障 24-13
  - XYFS 12-1
  - XY 格式设置 12-1
  - 线路
    - 连接 22-2
  - 显示 10-1
    - 处理时间 10-1
  - 线性转换 12-5
  - 小时计量器 20-11
  - 选择 PLC 类型 24-8
  - 循环
    - 右 7-10
    - 左侧 7-8
- Y**
- 异或 6-1
  - 已检从机列表 (LDS) 24-26
  - 移位
    - 右 7-3
    - 左侧 7-1
  - 移位 / 循环指令 7-1
  - 应答模式 22-2, 22-6
  - 用户
    - 通信 25-9
    - 协议 25-9
  - 用户程序
    - 调整时钟 9-7
    - 设置日历 / 时钟 9-6
  - 与 6-1
  - 预定从机列表 (LPS) 24-27
  - 余弦 17-4
  - 源
    - 和目标设备 2-7
    - 设备 2-7
  - 运行访问
    - 读取 16-2
    - 写入 16-3
  - 运行状态 22-2
- Z**
- ZRN1 13-26
  - ZRN2 13-26
  - ZRN3 13-26
  - 正切 17-5
  - 正弦 17-3



- 识别 24-4
- 指定网络编号 21-3
- 直接
  - 控制动作 14-14
- 指令
  - 程序分支 11-1
  - 对数 / 乘方 18-1
  - 接口 10-1
  - 逻辑运算 6-1
  - 脉冲 13-1
  - PID 14-1
  - 三角函数 17-1
  - 数据比较 4-1
  - 数据转换 8-1
  - 双 / 示教定时器 15-1
  - 四则运算 5-1
  - 移位 / 循环 7-1
  - 智能型模块访问 16-1
  - 周程序 9-1
  - 传送 3-1
  - 坐标转换 12-1
- 智能型模块访问
  - 指令 16-1
  - 状态代码 16-6
- 指数 18-3
- 执行先进 19-3
- 重叠坐标 12-6
- 重复
  - 操作
    - ADD 和 SUB 指令 5-6
    - ANDW、ORW 和 XORW 指令 6-3
    - DIV 指令 5-10
    - 间接位传送指令 3-11
    - MUL 指令 5-8
    - 数据比较指令 4-4
    - 传送指令 3-3
  - 次数 2-7
  - 指定 2-7
- 重试
  - 次数 22-3
  - 间隔 22-3
- 周
  - 表 9-2
  - 程序指令 9-1
  - 定时器 9-1
- 周期时间 24-5
- 转换 10-1, 10-3
  - X ~ Y 12-2
  - 线性 12-5
  - Y ~ X 12-3
- 转换 AS-Interface 主机模块模式 24-14
- 转换前的进程变量 14-16
- 转换数据类型 8-19
- 传送 3-1
  - 求反 3-5
- 状态
  - 代码
    - 智能型模块访问 16-6
    - 寄存器调制解调器模式 22-7
    - 继电器 13-4, 13-10, 13-19, 13-29
    - LED 24-16, 24-17
    - 内部继电器 22-2
  - 信息 24-24
    - 内部继电器 24-24
  - 准备使用调制解调器 22-9
  - 子程序 11-4
  - 自动调节 14-12
  - 自然对数 18-1
  - 字移位 7-7
  - 最大
    - AS-Interface 总线周期时间 24-5
  - 坐标转换指令 12-1





#### 美国

IDEC CORPORATION  
1175 Elko Drive, Sunnyvale, CA 94089-2209, USA  
Tel: +1-408-747-0550  
Toll Free: (800) 262-IDEC  
Fax: +1-408-744-9055  
Toll Free Fax: (800) 635-6246  
E-mail: [opencontact@idec.com](mailto:opencontact@idec.com)

#### 加拿大

IDEC CANADA LIMITED  
3155 Pepper Mill Court, Unit 4,  
Mississauga, Ontario, L5L 4X7, Canada  
Tel: +1-905-890-8561  
Toll Free: (888) 317-4332  
Fax: +1-905-890-8562  
E-mail: [sales@ca.idec.com](mailto:sales@ca.idec.com)

#### 澳大利亚

IDEC AUSTRALIA PTY. LTD.  
17/104 Ferntree Gully Road, Oakleigh, Victoria 3166, Australia  
Tel: +61-3-8523-5900  
Toll Free: 1800-68-4332  
Fax: +61-3-8523-5999  
E-mail: [sales@au.idec.com](mailto:sales@au.idec.com)

#### 英国

IDEC ELECTRONICS LIMITED  
Unit 2, Beechwood, Chineham Business Park,  
Basingstoke, Hampshire RG24 8WA, UK  
Tel: +44-1256-321000  
Fax: +44-1256-327755  
E-mail: [sales@uk.idec.com](mailto:sales@uk.idec.com)

#### 德国

IDEC ELEKTROTECHNIK GmbH  
Wendenstrasse 331, 20537 Hamburg, Germany  
Tel: +49-40-25 30 54 - 0  
Fax: +49-40-25 30 54-24  
E-mail: [service@idec.de](mailto:service@idec.de)

#### 日本

IDEC CORPORATION  
6-64, Nishi-Miyahara 2-Chome,  
Yodogawa-ku, Osaka 532-0004, Japan  
Tel: +81-6-6398-2527  
Fax: +81-6-6398-2547  
E-mail: [marketing@idec.co.jp](mailto:marketing@idec.co.jp)

#### 中国

IDEC (SHANGHAI) CORPORATION  
Room 701-702 Chong Hing Finance Center, No.288  
Nanjing Road West, Shanghai 200003, P.R.C.  
Tel: +86-21-6135-1515  
Fax: +86-21-6135-6225  
E-mail: [idec@cn.idec.com](mailto:idec@cn.idec.com)

IDEC (BEIJING) CORPORATION  
Room 211B, Tower B, The Grand Pacific Building, 8A  
Guanghua Road, Chaoyang District, Beijing 100026, PRC  
TEL: +86-10-6581-6131  
FAX: +86-10-6581-5119

IDEC (SHENZHEN) CORPORATION  
Unit AB-3B2, Tian Xiang Building, Tian'an Cyber Park,  
Fu Tian District, Shenzhen, Guang Dong 518040, PRC  
Tel: +86-755-8356-2977  
Fax: +86-755-8356-2944

#### 香港

IDEC IZUMI (H.K.) CO., LTD.  
Unit G & H, 26/F., MG Tower,  
No. 133 Hoi Bun Road, Kwun Tong, Kowloon, Hong Kong  
Tel: +852-2803-8989  
Fax: +852-2565-0171  
E-mail: [info@hk.idec.com](mailto:info@hk.idec.com)

#### 台湾

IDEC TAIWAN CORPORATION  
8F-1, No.79, Hsin Tai Wu Road, Sec.1, Hsi-Chih District,  
22101 New Taipei City, Taiwan  
Tel: +886-2-2698-3929  
Fax: +886-2-2698-3931  
E-mail: [service@tw.idec.com](mailto:service@tw.idec.com)

#### 新加坡

IDEC IZUMI ASIA PTE. LTD.  
No. 31, Tannery Lane #05-01  
HB Centre 2, Singapore 347788  
Tel: +65-6746-1155  
Fax: +65-6844-5995  
E-mail: [info@sg.idec.com](mailto:info@sg.idec.com)

#### 泰国

IDEC ASIA (THAILAND) CO.,LTD.  
20th Fl., Sorachai Bldg., No.23/78, Soi Sukhumvit 63,  
Sukhumvit Rd., Klongton-Nua, Wattana, Bangkok 10110  
Tel: +66-2-392-9765  
Fax: +66-2-392-9768  
E-mail: [sales@th.idec.com](mailto:sales@th.idec.com)