



# 梯形图编程手册

SmartAXIS FT2J 型

FC6A 型 MICROSmart

programmable by





# 安全上的重要注意事项


---


为了安全使用弊社的产品，请务必在使用产品前阅读各产品的用户手册中所述的安全上的重要注意事项。产品操作不当可能导致死亡或重伤。

在本手册中，将因错误操作而可能发生的危险程度分为“警告”和“注意”。以下为各含义：

 **警告** 警告提示用于强调操作不当会导致严重的人身伤亡。

 **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。

 **警告** 输入梯形图程序以及操作产品需要专业知识。如果没有专业知识，一般消费者无法操作。

 **注意** 在运行或停止梯形图程序之前，请彻底检查安全性。操作失误可能会导致机器损坏或事故。

# 关于本手册

本手册对梯形图程序编程的基本操作、各种指令以及注意事项进行说明。  
请仔细阅读本手册，了解产品的功能和性能，并正确使用。

本手册介绍了控制器的设备的分配、基本指令和高级指令、安装和操作原理。

## 第 1 章：基本操作

关于设置基本控制器系统以进行编程、启动和停止控制器操作以及简单操作步骤的一般信息。包括从在计算机上使用 WindLDR 创建用户程序到监控控制器操作的所有内容。

## 第 2 章：设备地址

基本指令或高级指令中使用的输入输出、内部继电器、寄存器、定时器和计数器等各种设备的分配描述，以及特殊内部继电器和特殊数据寄存器的分配详情。

## 第 3 章：指令参考

控制器编程的基本指令列表和高级指令列表，以及使用高级指令的一般规则。

## 第 4 章：基本指令

基本指令的编程、可用设备和样本程序。

## 第 5 章~第 28 章：

高级指令的详细描述。

## 附录

关于指令执行次数和字节大小的附加信息。

## 索引

依据关键字首字母顺序排序。

## 相关手册

可结合本手册一同参考使用。

手册名称	说明
FC6A 型 MICROSmart 用户手册	介绍产品规格、安装和接线说明、基本编程操作和特殊功能说明、设备和指令表以及 FC6A 型 MICROSmart 产品的故障排除步骤。
FC6A 型 MICROSmart 通信手册	介绍 FC6A 型 MICROSmart 的通信相关规格、功能的说明、设置方法及使用示例。
FC6A 型 MICROSmart PID 模块用户手册	介绍 PID 模块的规格和功能。
梯形图编程手册（本手册）	介绍使用梯形图编程的基本操作、监控方法、设备和指令列表以及每条指令的详细信息。
WindO/I-NV4 用户手册	介绍 SmartAXIS FT2J 型的基本性能、创建项目所需的操作步骤，以及用于配置项目的各种图形和部件。

我们的网站随时免费提供最新的产品手册 PDF。请从我们的网站下载最新的产品手册 PDF。

## 版本简历

2023 年 11 月：第 1 版

## 注意

- 本手册的所有权利均归 IDEC 公司所有。未经许可不得复制、转载、出售转让或出租。
- 本手册的内容如有更改，恕不另行通知。
- 关于产品的内容力求做到全，如有疑问或错误等发现之处，请联系购买的销售店或本公司的营业所、办事处。

## 商标

- WindO/I、WindLDR、SmartAXIS 和 MICROSmart 是 IDEC 公司在日本的注册商标。
- 本文提及的其他公司和产品名称可能是其各自所有者的商标或注册商标。

---

#### 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 控制器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。

# 本手册中使用的名称及简称

项目	说明
控制器	FT2J 型、FC6A 型的总称
FT2J型	SmartAXIS FT2J 型的简称
FC6A型	CPU模块、扩展模块、增设扩展模块、HMI模块、盒基本模块、盒的总称
CPU模块	All-in-One CPU模块、CAN J1939 All-in-One CPU模块、Plus CPU模块的总称
All-in-One CPU模块	FC6A-C****E CPU 模块的总称
16-I/O型	输入输出端子总计 16 点 All-in-One CPU 模块的的总称 (FC6A-C16****)
24-I/O型	输入输出端子总计24点的All-in-One CPU模块的总称 (FC6A-C24****)
CAN J1939 All-in-One CPU模块	FC6A-C40***EJ CPU模块的总称
Plus CPU模块	FC6A-D****CEE CPU模块的总称
Plus 16-I/O型	输入输出端子总计 16 点的 Plus CPU 模块的总称 (FC6A-D16****)
Plus 32-I/O型	输入输出端子总计 32 点的 Plus CPU 模块的总称 (FC6A-D32****)
40-I/O型	输入输出端子总计 40 点的 CPU 模块的总称 (FC6A-C40****)
AC电源类型	电源规格为交流电的CPU模块的总称 (FC6A-C****AE、FC6A-C****AEJ)
DC电源类型	直流24V电源型、直流12V电源型的CPU模块的总称
24V DC电源类型	这是电源规格为直流24V的CPU模块的统称 (FC6A-C****CE、FC6A-C****CEJ、FC6A-D****CEE)
12V DC电源类型	这是电源规格为直流12V的CPU模块的统称 (FC6A-C****DE、FC6A-C****DEJ)
继电器输出类型	输出为继电器输出的 CPU 模块的统称 (FC6A-C**R**E、FC6A-C**R**E*)
晶体管输出类型	晶体管沉型输出类型、晶体管保护源型输出类型的CPU模块的统称
晶体管沉型输出类型	这是其输出为晶体管沉型输出的 CPU 模块的总称 (FC6A-C**K**E、FC6A-C**K**E*、FC6A-D**K*CEE)
晶体管保护源型输出类型	这是其输出为晶体管保护源型输出的 CPU 模块的总称 (FC6A-C**P**E、FC6A-C**P**E*、FC6A-D**P*CEE)
扩展模块	I/O模块、通信模块、PID模块的总称
I/O模块	数字I/O模块、模拟I/O模块的总称
数字I/O模块	数字输入模块、数字输出模块、数字混合I/O模块的总称
数字输入模块	带输入端子的数字 I/O 模块的总称 (FC6A-N****)
数字输出模块	带输出端子的数字 I/O 模块的总称 (FC6A-R***、FC6A-T****)
数字混合I/O模块	带有输入和输出端子的数字 I/O 模块的总称 (FC6A-M****)
模拟I/O模块	模拟量输入模块、模拟量输出模块、模拟量混合I/O模块的总称
模拟量输入模块	带输入端子的模拟 I/O 模块的总称 (FC6A-J***、FC6A-J4CN*、FC6A-J4CH**、FC6A-J8CU*)
模拟量输出模块	带输出端子的模拟I/O 模块的总称 (FC6A-K****)
模拟量混合I/O模块	带有输入和输出端子的模拟I/O模块的总称 (FC6A-L03CN*、FC6A-L06A*)

项目		说明
	通信模块	串行通信模块的简称
	串行通信模块	FC6A-SIF52、FC6A-SIF524的总称
	PID模块	FC6A-F2M*、FC6A-F2MR* 的总称
	增设扩展模块	一体型、组合型主机、组合型从机的总称
	一体型	FC6A-EXM2、FC6A-EXM24的总称
	组合型主机	FC6A-EXM1M
	组合型从机	FC6A-EXM1S、FC6A-EXM1S4 的总称
	HMI模块	FC6A-PH1
	盒基本模块	FC6A-HPH1
	盒	I/O盒、通信盒的总称
	I/O盒	数字I/O盒、模拟I/O盒的总称
	数字I/O盒	扩展数字输入或数字输出的 I/O 盒的总称 (FC6A-PN4、FC6A-PT*4)
	模拟I/O盒	扩展模拟输入或输出的 I/O 盒的总称 (FC6A-PJ2A、FC6A-P*2**)
	通信盒	RS232C通信盒、RS485通信盒、Bluetooth通信盒的总称
	RS232C通信盒	FC6A-PC1
	RS485通信盒	FC6A-PC3
	Bluetooth通信盒	FC6A-PC4
	WindO/I-NV4	创建FT2J型画面的作图软件
	WindLDR	FT2J 型控制功能的编辑器。以及 FC6A 型的梯形图编程软件
	USB电缆	USB维护电缆 (HG9Z-XCM42)、USB-mini B 端口延长电缆 (HG9Z-XCE21) 的总称
	用户程序	梯形图编程软件 WindLDR 设置的功能设置以及梯形图程序的统合数据
	功能设置	控制器的各项设置内容 设置选项卡、模块构成编辑器的设置内容
	梯形图程序	主程序、子程序、用户定义宏的总称
	主程序	第一行是梯形图程序进入点的程序。进入点是梯形图程序执行的起点它是在梯形图程序编辑器的 [ 主程序 ] 选项卡上创建的。
	子程序	以下程序。 <ul style="list-style-type: none"> <li>主程序内的从 LABEL 指令到 LRET 指令的程序</li> <li>用 WindLDR 的子程序功能创建的程序 (梯形图程序编辑器的选项卡中显示为 [# 子程序] (# : 子程序编号))</li> </ul>
	用户定义宏	WindLDR 用户定义宏功能创建的程序。 (在梯形图程序编辑器的选项卡中, 用 [# 用户定义宏] 表示 (# : 用户定义宏的编号))
	源设备	作为计算对象的设备 (为执行高级指令数据的存储位)
	目标设备	它是一个存储操作结果的设备
	定时器指令	TML指令、TIM指令、TMH指令、TMS指令的总称
	断开延时定时器指令	TMLO指令、TIMO指令、TMHO指令、TMSO指令的总称
	计数器指令	CNT指令、CDP指令、CUD指令的总称
	双字计数器指令	CNTD指令、CDPD指令、CUDD指令的总称
	移位寄存器指令	SFR指令、SFRN指令的总称
	计数器比较指令	CC=指令、CC>=指令的总称

项目	说明
比较指令	CMP= 指令、CMP<> 指令、CMP< 指令、CMP> 指令、CMP<= 指令、CMP>= 指令的总称
脉冲输出指令	PULS指令、PWM指令、RAMP指令、RAMPL指令、ARAMP指令、ABS指令、JOG指令的总称
双/示教定时器指令	DTIM指令、DTML指令、DTMH指令、DTMS指令、TTIM指令的总称

### WindLDR 的名称

本手册中使用的名称	WindLDR 操作步骤
功能设置	“设置”选项卡>“功能设置”组
监控	选择“联机”>“监控”>“启动监控器”。
PLC 状态	选择“联机”>“PLC”>“状态”。
通信设置	选择“联机”>“通信”>“设置”。
Modbus 主机请求表	在“设置”选项卡的“功能设置”中单击“通信端口”，并在所显示的“功能设置”对话框内，从“通信端口”的“通信模式”中选择“Modbus RTU 主机”或“Modbus TCP 客户端”
应用程序按钮	显示在菜单栏左侧的按钮。单击即可显示“新建”、“保存”、“另存为”等菜单或最近使用的项目、及“WindLDR 选项”、“退出 WindLDR”按钮





# 目录

安全上的重要注意事项 .....	前言 -1
关于本手册 .....	888-2
本手册中使用的名称及简称 .....	前言 -4

<b>第1章：</b>	<b>基本操作</b>	
	启动 WindLDR .....	1-1
	创建程序 .....	1-4
	保存一个项目 .....	1-8
	模拟操作 .....	1-9
	下载程序 .....	1-10
	监控操作 .....	1-12
	退出 WindLDR .....	1-13
	关于工作区显示的窗口 .....	1-14
	确认 WindLDR 版本的方法 .....	1-17
	梯形图程序的动作 .....	1-18
	运行 / 停止用户程序 .....	1-19
<b>第2章：</b>	<b>设备地址</b>	
	FT2J 型 .....	2-1
	FC6A 型 .....	2-12
	设备名称和符号 .....	2-62
	类型 ID、状态一览 .....	2-63
<b>第3章：</b>	<b>指令参考</b>	
	基本指令表 .....	3-1
	高级指令表 .....	3-3
	关于高级指令 .....	3-7
	.....	3-12
	执行指令时设备的指定方法 .....	3-13
<b>第4章：</b>	<b>基本指令</b>	
	LOD (读取) .....	4-1
	LODN (读取非) .....	4-1
	OUT (输出) .....	4-1
	OUTN (求反输出) .....	4-1
	SET (置位) .....	4-3
	RST (复位) .....	4-3
	AND (与) .....	4-4
	ANDN (与非) .....	4-4
	OR (或) .....	4-5
	ORN (或非) .....	4-5
	AND LOD (块与) .....	4-6
	OR LOD (块或) .....	4-6
	BPS (位推入) .....	4-7
	BRD (位读取) .....	4-7
	BPP (位弹出) .....	4-7
	TML (1-s 定时器) .....	4-8
	TIM (100-ms 定时器) .....	4-8
	TMH (10-ms 定时器) .....	4-8
	TMS (1-ms 定时器) .....	4-8
	TMLO (1-s 断开延时定时器) .....	4-10
	TIMO (100-ms 断开延时定时器) .....	4-10
	TMHO (10-ms 断开延时定时器) .....	4-10
	TMSO (1-ms 断开延时定时器) .....	4-10
	CNT (加计数器) .....	4-14
	CDP (加 / 减计数器) .....	4-14

	CUD (加 / 减切换计数器) .....	4-14
	CNTD (双字加计数器) .....	4-18
	CDPD (双字加 / 减计数器) .....	4-18
	CUDD (双字加 / 减切换计数器) .....	4-18
	CC= (计数器比较 (=)) .....	4-23
	CC>= (计数器比较 (>=)) .....	4-23
	DC= (数据寄存器比较 (=)) .....	4-25
	DC>= (数据寄存器比较 (>=)) .....	4-25
	SFR (右移移位寄存器) .....	4-27
	SFRN (左移移位寄存器) .....	4-27
	SOTU (上升沿微分) .....	4-31
	SOTD (下降沿微分) .....	4-31
	MCS (主控继电器开始) .....	4-33
	MCR (主控继电器结束) .....	4-33
	JMP (跳转) .....	4-35
	JEND (跳转结束) .....	4-35
	END .....	4-37
	梯形图编程限制 .....	4-38
<b>第 5 章:</b>	<b>传送指令</b>	
	MOV (传送) .....	5-1
	MOVN (求反传送) .....	5-5
	IMOV (间接传送) .....	5-6
	IMOVN (间接求反传送) .....	5-8
	MOVC (传送字符) .....	5-9
	BMOV (块传送) .....	5-12
	IBMV (间接位传送) .....	5-13
	IBMVN (间接位求反传送) .....	5-15
	NSET (N 数据置位) .....	5-16
	NRS (N 数据重复置位) .....	5-17
	XCHG (交换) .....	5-18
	TCCST (存储定时器 / 计数器当前值) .....	5-19
<b>第 6 章:</b>	<b>数据比较指令</b>	
	CMP= (比较等于) .....	6-1
	CMP<> (比较不等于) .....	6-1
	CMP< (比较小于) .....	6-1
	CMP> (比较大于) .....	6-1
	CMP<= (比较小于或等于) .....	6-2
	CMP>= (比较大于或等于) .....	6-2
	ICMP>= (间隔比较大于或等于) .....	6-6
	LC= (接点比较等于) .....	6-8
	LC<> (接点比较不等于) .....	6-8
	LC< (接点比较小于) .....	6-8
	LC> (接点比较大于) .....	6-8
	LC<= (接点比较小于或等于) .....	6-8
	LC>= (接点比较大于或等于) .....	6-8
<b>第 7 章:</b>	<b>四则运算指令</b>	
	ADD (加法) .....	7-1
	SUB (减法) .....	7-1
	MUL (乘法) .....	7-1
	DIV (除法) .....	7-1
	INC (递增) .....	7-14
	DEC (递减) .....	7-14
	ROOT (平方根) .....	7-15
	SUM (合计) .....	7-16
	RNDM (随机) .....	7-19

<b>第 8 章:</b>	<b>逻辑运算指令</b>	
	ANDW (与) .....	8-1
	ORW (或) .....	8-1
	XORW (异或) .....	8-1
<b>第 9 章:</b>	<b>移位/循环指令</b>	
	SFTL (左移) .....	9-1
	SFTR (右移) .....	9-3
	BCDLS (BCD 码左移) .....	9-5
	WSFT (字移位) .....	9-7
	ROTL (循环左移) .....	9-8
	ROTR (循环右移) .....	9-10
<b>第 10 章:</b>	<b>数据转换指令</b>	
	HTOB (HEX → BCD 码) .....	10-1
	BTOH (BCD 码 → HEX) .....	10-3
	HTOA (HEX → ASCII 码) .....	10-5
	ATOH (ASCII 码 → HEX) .....	10-7
	BTOA (BCD 码 → ASCII 码) .....	10-9
	ATOB (ASCII 码 → BCD 码) .....	10-12
	ENCO (编码) .....	10-15
	DECO (解码) .....	10-16
	BCNT (位计数) .....	10-17
	ALT (交替输出) .....	10-18
	CVDT (转换数据类型) .....	10-19
	DTDV (数据分割) .....	10-21
	DTCB (数据组合) .....	10-22
	SWAP (数据交换) .....	10-23
<b>第 11 章:</b>	<b>周程序指令</b>	
	WEEK (周定时器) .....	11-3
	YEAR (年定时器) .....	11-16
<b>第 12 章:</b>	<b>显示指令</b>	
	MSG (消息) .....	12-1
	DISP (显示) .....	12-20
	DGRD (数字读取) .....	12-22
<b>第 13 章:</b>	<b>程序分支指令</b>	
	LABEL (标签) .....	13-1
	LJMP (标签跳转) .....	13-2
	LCAL (标签调用) .....	13-4
	LRET (标签返回) .....	13-5
	DJNZ (递减跳转非零) .....	13-7
<b>第 14 章:</b>	<b>刷新指令</b>	
	IOREF (I/O 刷新) .....	14-1
	HSCRF (高速计数器刷新) .....	14-3
	FRQRF (频率测量刷新) .....	14-4
	COMRF (通信刷新) .....	14-5
<b>第 15 章:</b>	<b>中断控制指令</b>	
	DI (禁用中断) .....	15-1
	EI (启用中断) .....	15-1
<b>第 16 章:</b>	<b>坐标转换指令</b>	
	XYFS (XY 格式设置) .....	16-1
	CVXTY (X → Y 转换) .....	16-3

	CVYTX (Y → X 转换) .....	16-4
<b>第 17 章:</b>	<b>平均指令</b>	
	AVRG (平均化) .....	17-1
<b>第 18 章:</b>	<b>脉冲输出指令</b>	
	PULS (脉冲输出) .....	18-1
	PWM (可变占空比脉冲输出) .....	18-9
	RAMP (台形控制) .....	18-17
	RAMPL (直线插补控制) .....	18-29
	ZRN (归零) .....	18-37
	ARAMP (带表的 RAMP) .....	18-47
	ABS (绝对位置设置) .....	18-69
	JOG (JOG 运行) .....	18-72
	脉冲监控 .....	18-79
	定位控制 .....	18-82
<b>第 19 章:</b>	<b>PID 控制指令</b>	
	PID (PID 控制) .....	19-1
	PIDA (PID 控制) .....	19-4
	PIDD (微分衰减的 PID) .....	19-28
	PID 监控 .....	19-42
	应用实例 .....	19-48
<b>第 20 章:</b>	<b>双/示教定时器指令</b>	
	DTML (1 秒双定时器) .....	20-1
	DTIM (100 毫秒双定时器) .....	20-1
	DTMH (10 毫秒双定时器) .....	20-1
	DTMS (1 毫秒双定时器) .....	20-1
	TTIM (示教定时器) .....	20-3
<b>第 21 章:</b>	<b>三角函数指令</b>	
	RAD (角度换算为弧度) .....	21-1
	DEG (弧度换算为角度) .....	21-2
	SIN (正弦) .....	21-3
	COS (余弦) .....	21-4
	TAN (正切) .....	21-5
	ASIN (反正弦) .....	21-6
	ACOS (反余弦) .....	21-7
	ATAN (反正切) .....	21-8
<b>第 22 章:</b>	<b>对数/幂指令</b>	
	LOGE (自然对数) .....	22-1
	LOG10 (常用对数) .....	22-2
	EXP (指数) .....	22-3
	POW (幂) .....	22-4
<b>第 23 章:</b>	<b>文件数据处理指令</b>	
	FIFO (先进先出格式) .....	23-1
	FIEX (执行先进) .....	23-3
	FOEX (执行先出) .....	23-4
	NDSRC (N 数据查找) .....	23-6
<b>第 24 章:</b>	<b>时钟指令</b>	
	TADD (时间加法) .....	24-1
	TSUB (时间减法) .....	24-5
	HOUR (小时计量器) .....	24-9
	HTOS (HMS → 秒) .....	24-11
	STOH (秒 → HMS) .....	24-12

<b>第 25 章:</b>	<b>数据日志指令</b>	
	DLOG (数据日志) .....	25-1
	TRACE (数据跟踪) .....	25-16
<b>第 26 章:</b>	<b>脚本</b>	
	SCRPT (脚本) .....	26-1
	脚本功能概述 .....	26-3
	脚本编程和管理 .....	26-5
	脚本的记述方法 .....	26-12
	脚本的记述例 .....	26-19
	注意事项 .....	26-42
	关于运算符优先级 .....	26-43
<b>第 27 章:</b>	<b>流量计算指令</b>	
	SCALE (模拟量输入转换) .....	27-1
	FLWA (模拟流量累积) .....	27-10
	FLWP (脉冲流量累积) .....	27-19
<b>第 28 章:</b>	<b>用户定义宏指令</b>	
	UMACRO (用户定义宏) .....	28-1
<b>附录</b>		
	用户定义宏与子程序的差异 .....	附录 -2
	指令的执行时间 .....	附录 -3
	指令数据长度 (字节) .....	附录 -11
	用户程序执行错误和梯形图程序执行错误 .....	附录 -19
	用户程序执行错误记录 .....	附录 -20
<b>索引</b>		
	产品保修说明 .....	前言 -1



# 1: 基本操作

## 简介

本章介绍操作 WindLDR 的基本步骤以及编程和维护软件。

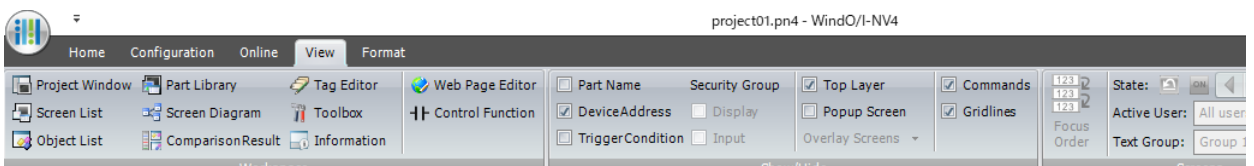
## 启动 WindLDR

FT2J FC6A

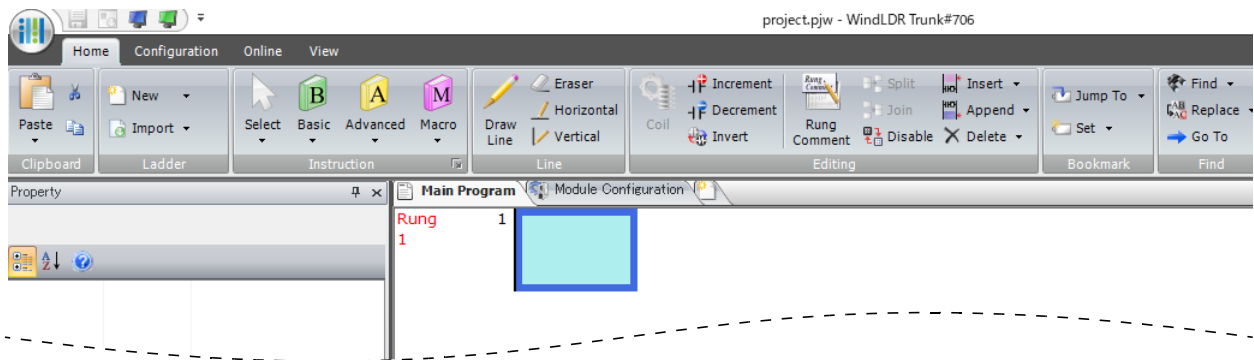
本节介绍启动 WindLDR 和创建梯形图程序的操作步骤。

### FT2J 型

WindLDR 通过 WindO/I-NV4 启动。在 WindO/I-NV4 的“视图”选项卡上，单击“工作区”下的“控制功能”。启动 WindLDR。



打开如下的梯形图编辑器（编辑梯形图程序的窗口）。



然后创建梯形图程序。详情请参见第 1-4 页上的“创建程序”。

## 1: 基本操作

### FC6A 型

1. 按以下步骤启动 WindLDR。

#### ■ Windows 11/10

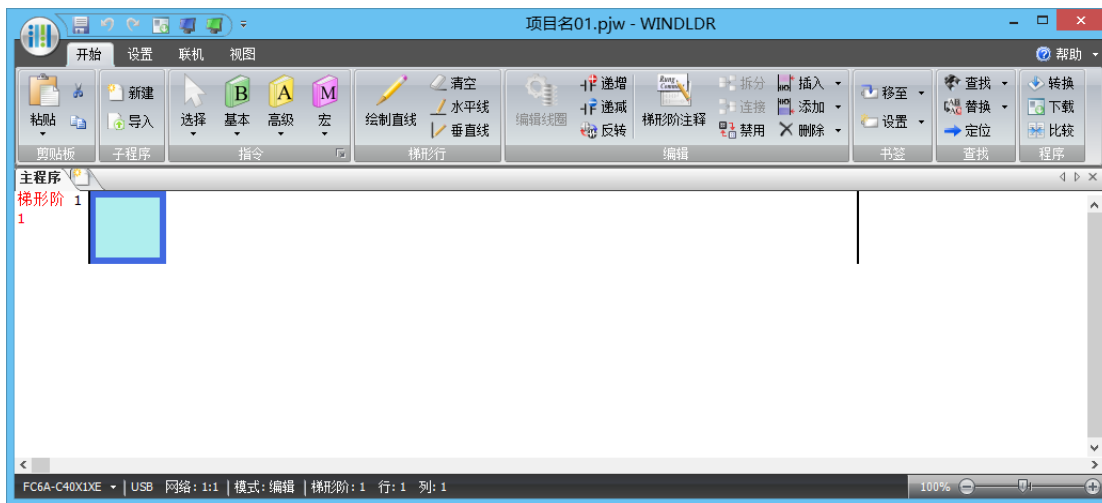
按照“开始”按钮、“所有程序”、“IDEC Automation Organizer”、“WindLDR”的顺序单击。

#### ■ Windows 8

单击开始画面磁贴中的“WindLDR”。

#### ■ Windows 7

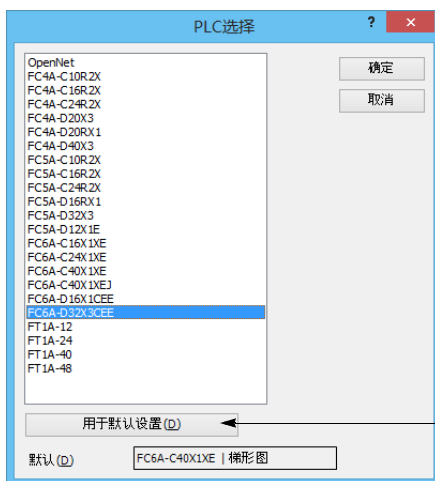
按照“开始”按钮、“程序”、“IDEC Automation Organizer”、“WindLDR”的顺序单击。



2. 在 WindLDR 菜单栏的中选择“设置”，然后选择“PLC”>“PLC 类型”。

出现“PLC 选择”对话框。

3. 在选择框中选择 PLC 类型，然后单击“确定”。

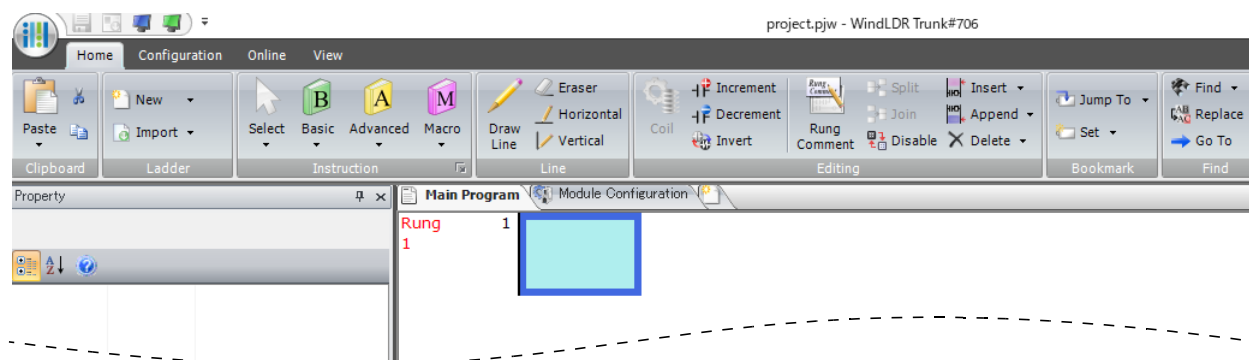


从一览中选择机型，单击“用于默认设置”按钮，  
可将所选机型设为 WindLDR 启动时的默认机型。



PLC 选择选项	型号
FC6A-C16XXXE	FC6A-C16R1AE, FC6A-C16R1CE, FC6A-C16K1CE, F6A-C16P1CE, FC6A-C16R1DE, FC6A-C16K1DE, FC6A-C16P1DE, FC6A-C16R4AE, FC6A-C16R4CE, FC6A-C16K4CE, F6A-C16P4CE, FC6A-C16R4DE, FC6A-C16K4DE, FC6A-C16P4DE
FC6A-C24XXXE	FC6A-C24R1AE, FC6A-C24R1CE, FC6A-C24P1CE, FC6A-C24K1CE, FC6A-C24R4AE, FC6A-C24R4CE, FC6A-C24P4CE, FC6A-C24K4CE
FC6A-C40XXXE	FC6A-C40R1AE, FC6A-C40R1CE, FC6A-C40P1CE, FC6A-C40K1CE, FC6A-C40R1DE, FC6A-C40P1DE, FC6A-C40K1DE, FC6A-C40R4AE, FC6A-C40R4CE, FC6A-C40P4CE, FC6A-C40K4CE, FC6A-C40R4DE, FC6A-C40P4DE, FC6A-C40K4DE
FC6A-C40XXXEJ	FC6A-C40R1AEJ, FC6A-C40R1CEJ, FC6A-C40P1CEJ, FC6A-C40K1CEJ, FC6A-C40R1DEJ, FC6A-C40P1DEJ, FC6A-C40K1DEJ, FC6A-C40R4AEJ, FC6A-C40R4CEJ, FC6A-C40P4CEJ, FC6A-C40K4CEJ, FC6A-C40R4DEJ, FC6A-C40P4DEJ, FC6A-C40K4DEJ
FC6A-D16XXCEE	FC6A-D16R1CEE, FC6A-D16R4CEE, FC6A-D16K1CEE, FC6A-D16K4CEE, FC6A-D16P1CEE, FC6A-D16P4CEE
FC6A-D32XXCEE	FC6A-D32K3CEE, FC6A-D32K4CEE, FC6A-D32P3CEE, FC6A-D32P4CEE

打开如下的梯形图编辑器（编辑梯形图程序的窗口）。



然后创建梯形图程序。详情请参见第 1-4 页上的“创建程序”。

## 1: 基本操作

### 创建程序

FT2J FC6A

#### 创建一个梯形图程序

介绍在 WindLDR 中创建梯形图程序的操作步骤。

**注释:** 有关设备的详情, 请参见第 2-1 页上的“设备地址”。

#### 示例用户程序

使用 WindLDR 创建简单程序。示例程序将执行以下操作:

只有输入 I0 打开时, 才会打开输出 Q0。

只有输入 I1 打开时, 才打开输出 Q1。

输入 I0 和 I1 都打开时, 输出 Q2 以 1 秒为增量闪烁。

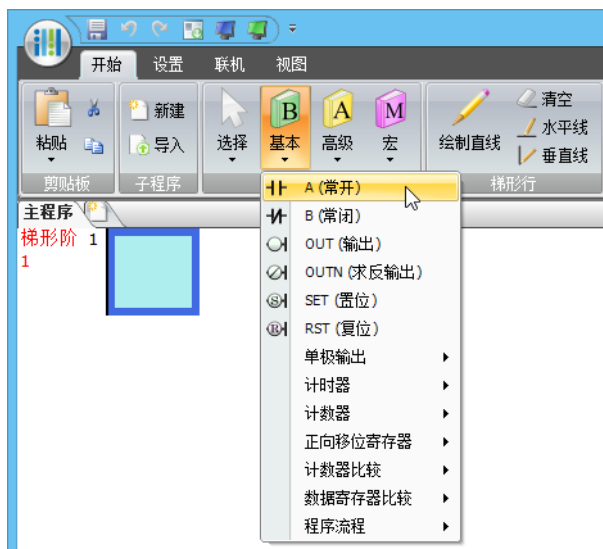
梯形阶编号	输入 I0	输入 I1	输出操作
1	开	关	输出 Q0 打开。
2	关	开	输出 Q1 打开。
3	开	开	输出 Q2 以 1 秒为增量闪烁。

**注释:** 一组控制输出的指令集或高级指令集称为梯形阶。WindLDR 管理梯形阶单元内的程序。功能说明可设置为单个梯形阶的梯形阶注释。

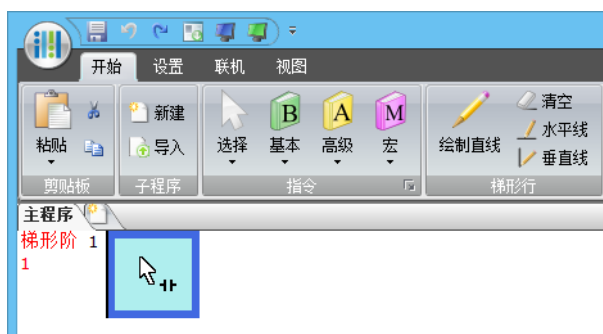
#### 按梯形阶编辑用户程序

通过插入输入 I0 的常开接点, 用 LOD 指令启动用户程序。

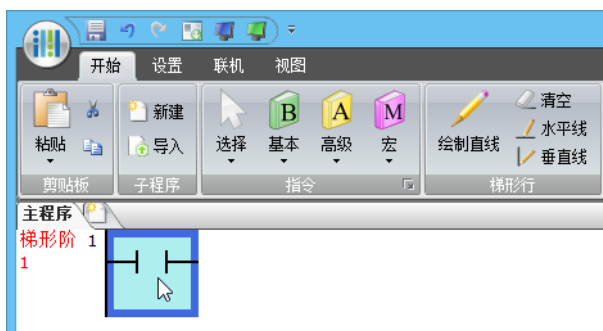
1. 从 WindLDR 菜单栏选择“开始”>“指令”>“基本”>“A (常开)”。



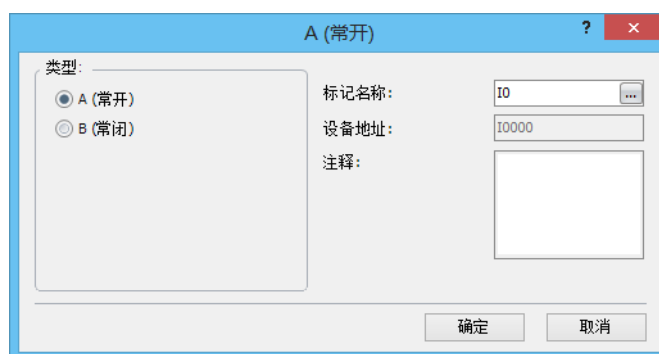
2. 将鼠标指针移动到想要插入常开接点的第一条线的第一列, 并单击左鼠标按钮。



3. 双击“A 接点”。  
将显示“A（常开）”对话框。



4. 在“标记名称”字段中输入“IO”，然后单击“确定”。



创建输入 I0 的 A 接点。



**注释:**

- 通过右击菜单输入 A 接点时，应在输入 A 接点的位置进行右击，并在右击菜单中单击“基本指令（B）”，然后再单击“A（常开）”。
- 使用键盘输入 A 接点时，应按下 [A] (A) 键，并在显示的“线圈选择”对话框中选择“A（常开）”，按下 [Enter] (Enter) 键。

输入 I0 的常开接点的编程内容位于第一个梯形行的第一列中。

下一步，通过插入输入 I1 的常闭接点以编写 ANDN 指令。

5. 从 WindLDR 菜单栏选择“开始”>“指令”>“基本”>“B（常闭）”。
6. 将鼠标指针移动到想要插入常闭接点的第一个梯形行的第二列，并单击左鼠标按钮。

## 1: 基本操作

7. 双击“B 接点”。  
将显示“B（常闭）”对话框。



8. 在标记名称字段中输入“11”，然后单击“确定”。

输入 11 的常闭接点的程序内容位于第一梯形行的第二列中。  
在第一个梯形行的末尾，通过插入输出 Q0 的常开线圈以编写 OUT 指令。

9. 从 WindLDR 菜单栏选择“开始”>“指令”>“基本”>“OUT（输出）”。

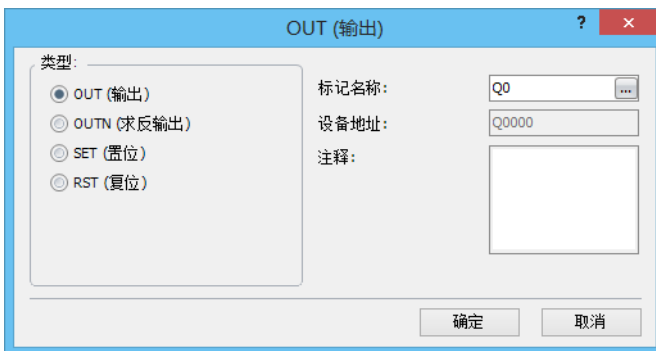
10. 将鼠标指针移动到想要插入输出线圈的第一条梯形行的第三列，并单击左鼠标按钮。

**注释：**插入指令（基本或高级）的另一个方法是在想要插入指令的位置键入指令符号 OUT。

11. 双击输出。  
将显示“OUT（输出）”对话框。



12. 在标记名称字段中输入“Q0”，然后单击“确定”。



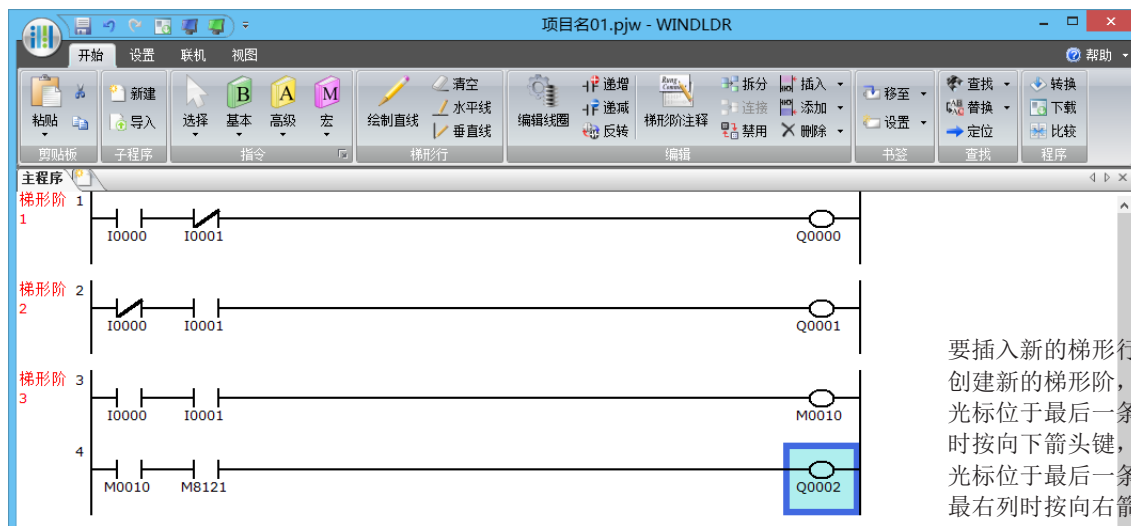
输出 Q0 的常开输出线圈的程序内容位于第一条梯形行的最右列。这就完成了梯形阶 1 的编程。



重复类似过程，继续为梯形阶 2 和 3 编程。

通过在光标位于前一个梯形阶上的同时按“Enter”键，可以插入新的梯形阶。通过选择“开始”>“编辑”>“添加”>“添加梯形阶”，也可以插入新的梯形阶。

完成后，梯形图程序的外观与下图类似。



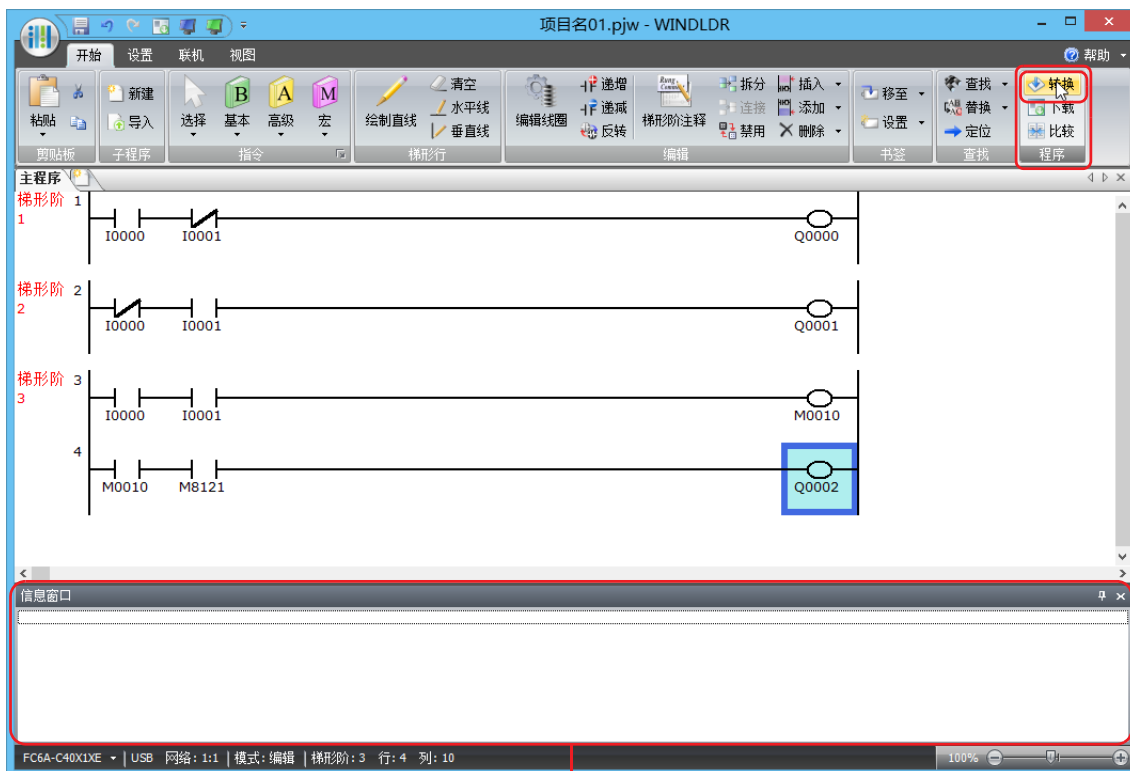
要插入新的梯形行而不创建新的梯形阶，请在光标位于最后一条行上时按向下箭头键，或当光标位于最后一条行的最右列时按向右箭头键。

## 转换程序

可检查该程序是否包含一些用户程序语法错误。

从 WindLDR 菜单栏选择“开始”>“程序”>“转换”。

正确连接指令符号时，就可成功完成程序转换。如果发现任何错误，这些错误就会被列示在信息窗口中。然后，进行一些必要的更改。



信息窗口

## 1: 基本操作

### 保存一个项目

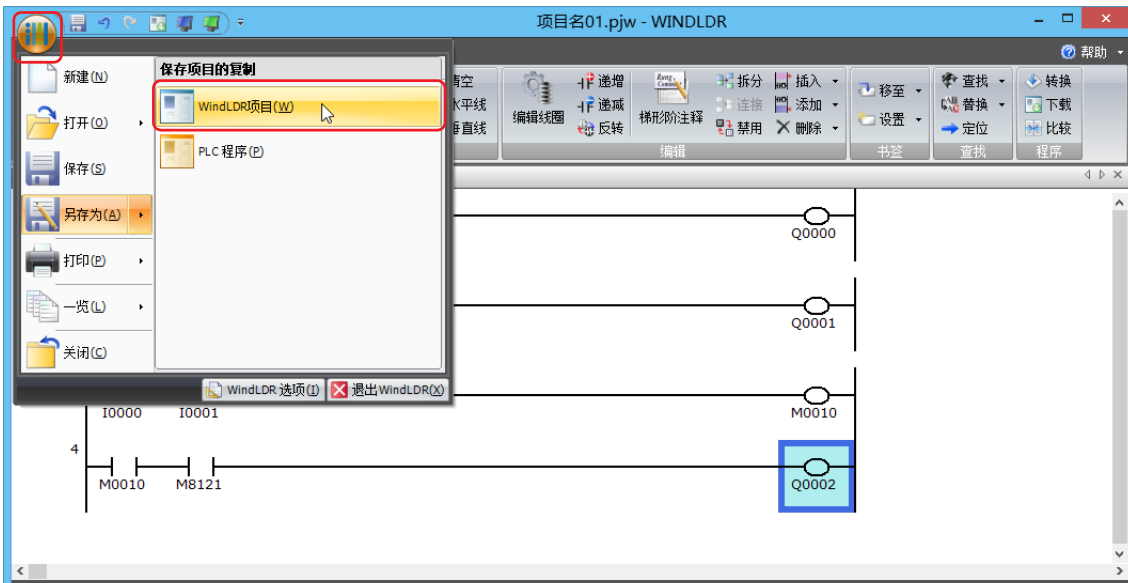
FT2J FC6A

介绍将所创建的梯形图程序保存为项目文件的操作步骤。

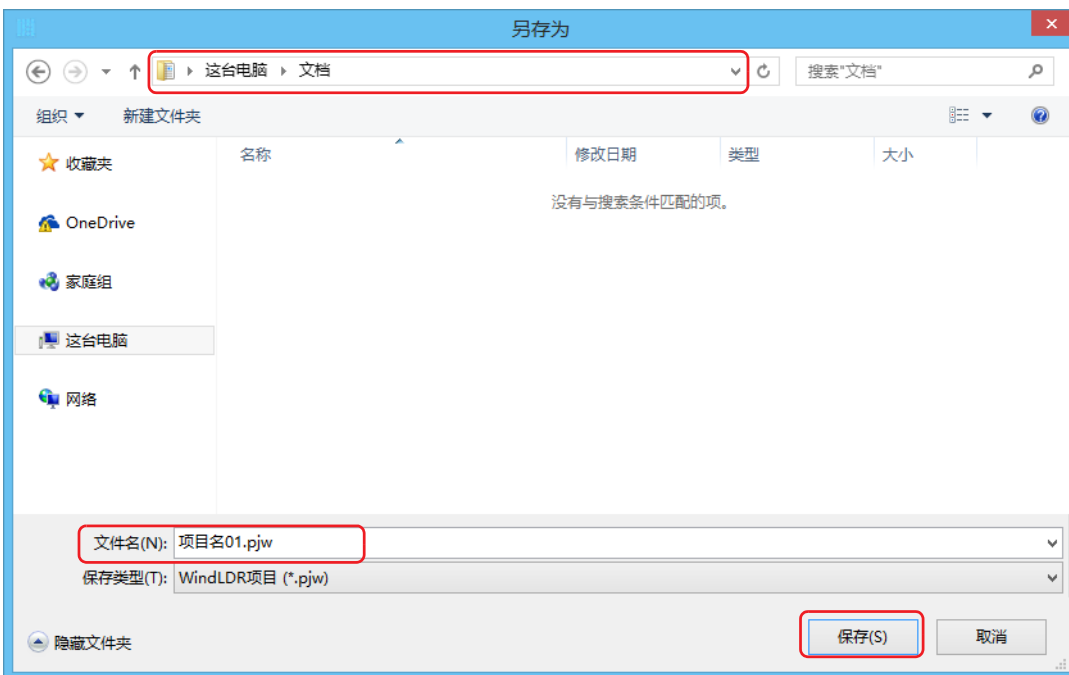
**注释：**保存 FT2J 型项目时，请参见《WindO/I-NV4 用户手册》第 4 章“1.3 保存项目数据”。

#### 1. 将项目另存。

按照 （应用程序）按钮、“另存为（A）”、“WindLDR 项目（W）”的顺序单击。



#### 2. 在“文件名”中输入文件名，并指定保存位置文件夹后单击“保存”按钮。



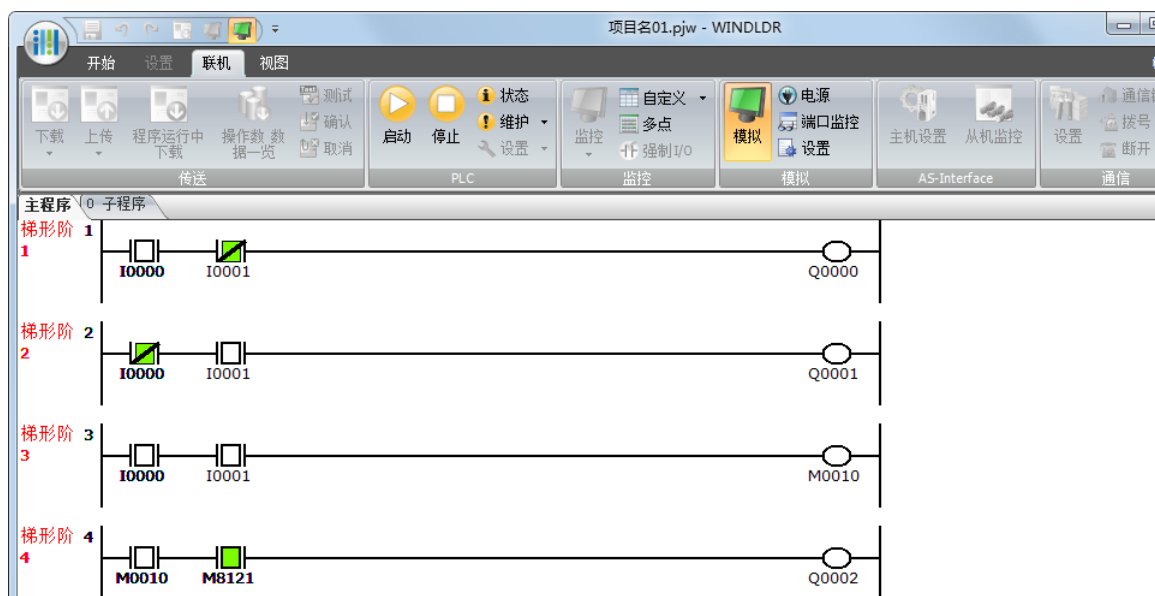
至此，项目保存在文件中。

**注释：**将保存的文件称为项目文件。扩展名为“.pjw”。

## 模拟操作

FT2J FC6A

介绍将用户程序传送到控制器之前，确认程序动作的操作步骤。  
从 WindLDR 菜单栏中选择“联机”>“模拟”，将出现模拟屏幕。



选择要更改的输入接点后进行右击，并单击右击菜单中的“置位（S）”或“复位（R）”。  
要退出模拟，从 WindLDR 菜单栏中选择“联机”>“模拟”。

### 注释：

- 也可通过选择接点后双击来变更输入接点的状态。
- 退出模拟功能时，应重新在“联机”选项卡的“模拟”中单击“模拟”。

## 1: 基本操作

### 下载程序

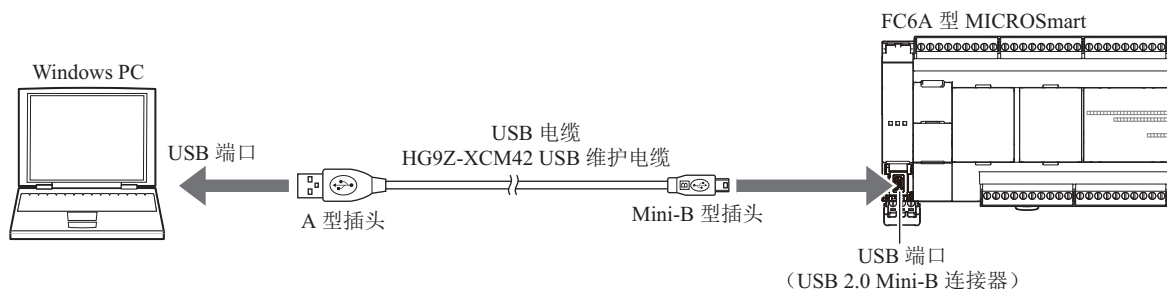
FT2J FC6A

当 PLC 上的 WindLDR 正在运行时，可以将用户程序下载到 FC6A 型。

可通过 USB 或者以太网从 WindLDR 下载用户程序到 FC6A 型。将以 USB 连接为例，对通信方法设置以及用户程序下载等操作步骤进行介绍。

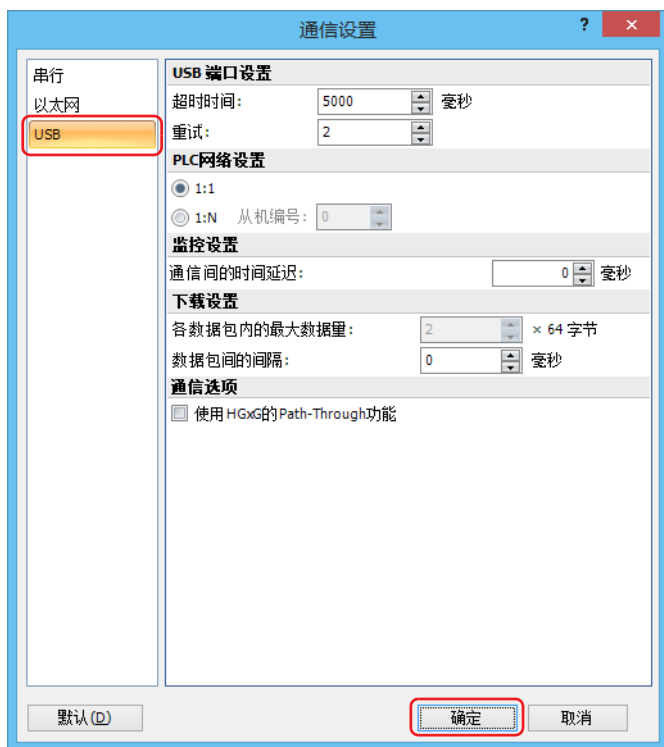
FC6A 型的 USB 端口必须通过 USB 电缆连接到 PC，这样才能使用 USB。

**注释：**将 FT2J 型的项目数据下载到本体时，请参见《WindO/I-NV4 用户手册》第 29 章“2.1 将项目数据下载到主装置”。



**注释：**PC 上必须安装专用的 USB 驱动程序，WindLDR 才能通过 USB 与 FC6A 型进行通信。有关驱动的安装步骤的详情，请参见《FC6A 型 MICROSmart 用户手册》附录“USB 驱动程序安装步骤”。

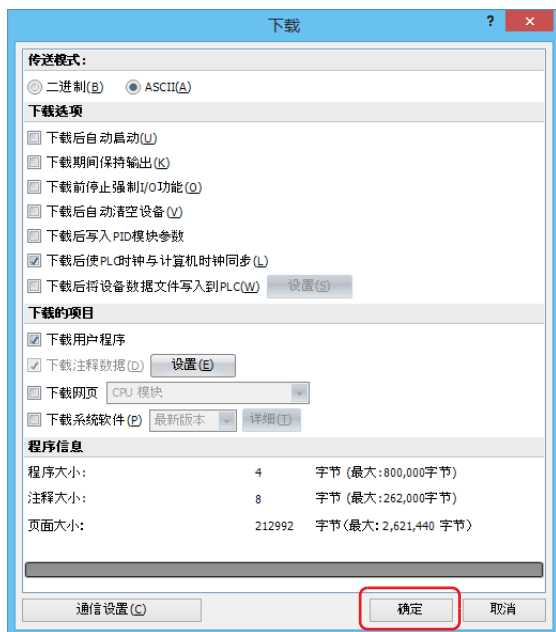
1. 从 WindLDR 菜单栏，选择“联机”>“设置”。  
将显示“通信设置”对话框。
2. 单击“USB”选项卡，并单击“确定”按钮。



此时，通信方法已设置为 USB。接下来，下载一个用户程序。



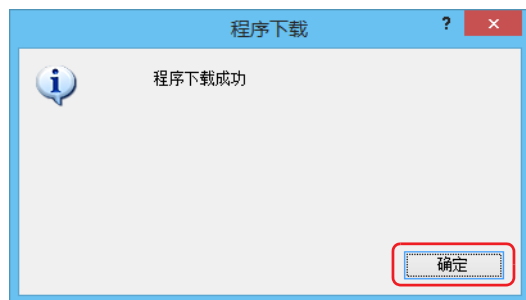
3. 在 WindLDR 菜单栏中选择“联机”>“下载程序”。将出现下载程序对话框，然后单击“下载”按钮。用户程序将下载到 FC6A 型。



**注释:** 也可选择“开始”>“程序”>“下载”，打开下载对话框。

**注释:** 下载用户程序时，“功能设置”中的所有值和选择项也将下载到 FC6A 型。有关功能设置的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章“功能和设置”。

显示以下信息后表示下载成功。单击“确定”按钮。



**注释:**

- 在 WindLDR 中启用“通过功能开关运行 / 停止 PLC”时，即使启用“下载后自动启动”下载程序，只要功能开关为 0，PLC 就不会运行。  
要使 PLC 运行，需要将功能开关设为 1。
- WindLDR 的初始设置中“通过功能开关运行 / 停止 PLC”为启用状态。

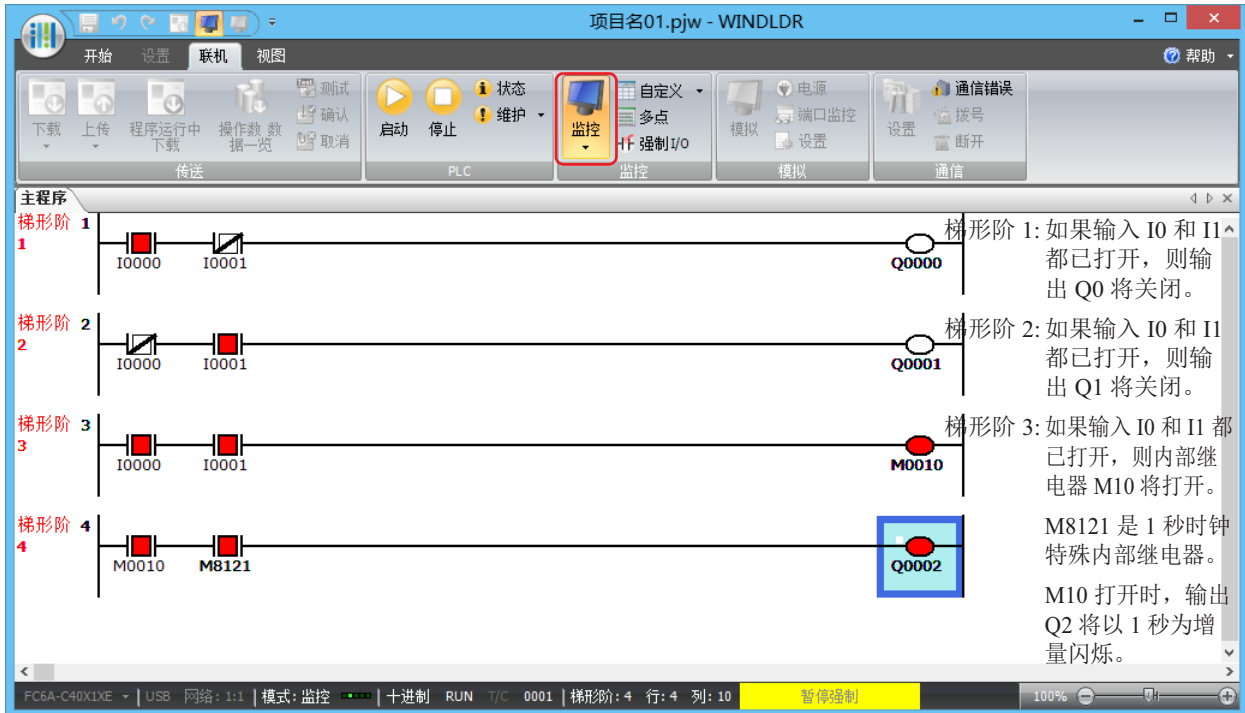
## 1: 基本操作

### 监控操作

FT2J FC6A

WindLDR 的另一个强大功能是在计算机上监控 PLC 操作。可以在梯形图中监控示例程序的输入和输出状态。在 WindLDR 菜单栏中选择“联机”>“监控”>“监控”。

如果输入 I0 和 I1 都已打开，则监控屏幕上的梯形图的外观将如下所示：

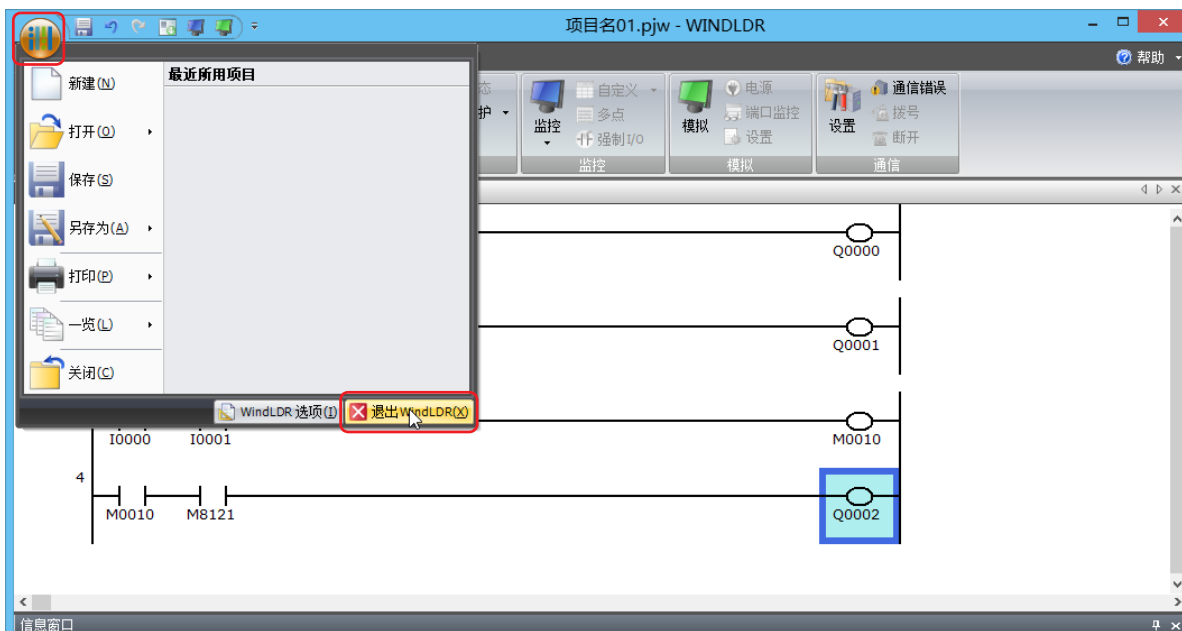


## 退出 WindLDR

FT2J FC6A

介绍退出 WindLDR 的操作步骤。

完成监控后，您可以直接从监控屏幕或编辑屏幕退出 WindLDR。无论使用哪种方法，都需要选择 WindLDR 应用程序按钮，然后单击“退出 WindLDR”。



**注释：**在退出的 WindLDR 时，也会退出 WindO/I-NV4。请参见《WindO/I-NV4 用户手册》第 2 章“2.2 WindO/I-NV4 的退出”。

## 1: 基本操作

### 关于工作区显示的窗口

FT2J FC6A

本节将对更改窗口显示位置或窗口显示的方法进行介绍。

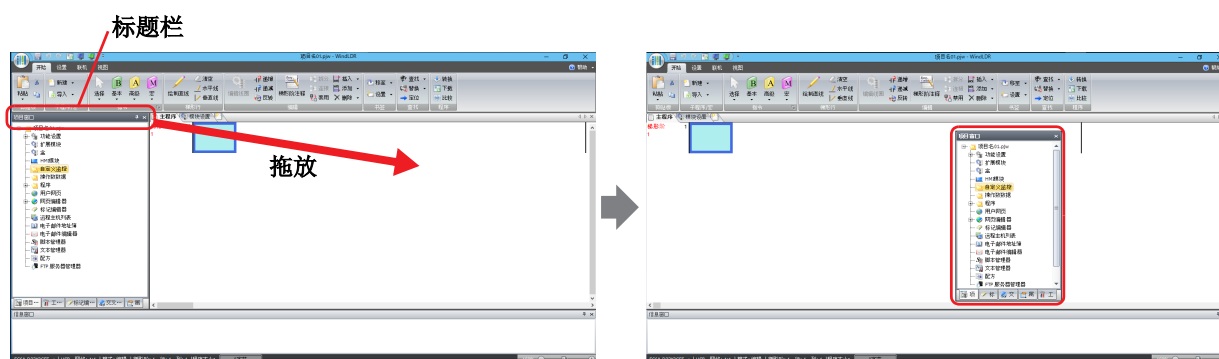
#### 变更窗口的显示位置

拖放窗口的标题栏或选项卡解除停驻后，可以任意变更窗口的显示位置。解除了停驻的窗口称为浮动窗口。

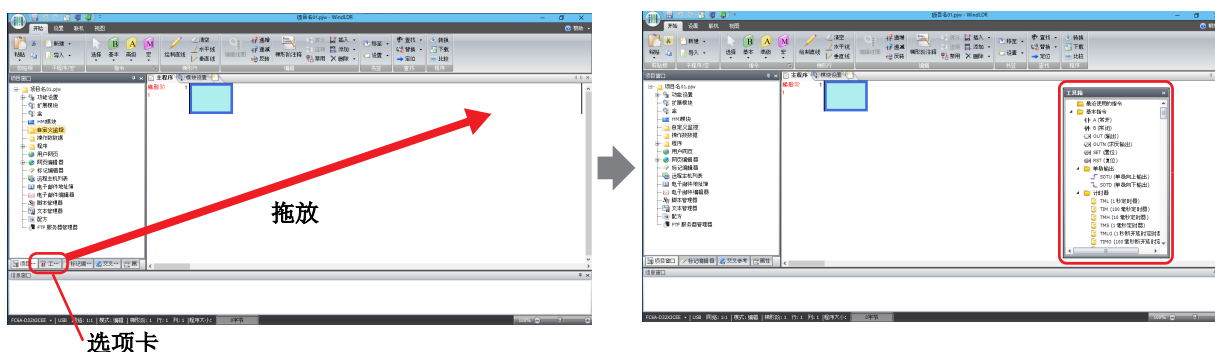
#### 解除停驻

对解除窗口标题栏或窗口选项卡停靠的方法进行介绍。

- 拖住工作区的窗口的标题栏可以一齐移动停驻着的窗口。




- 拖住工作区的窗口的选项卡只可以移动所选择的窗口。

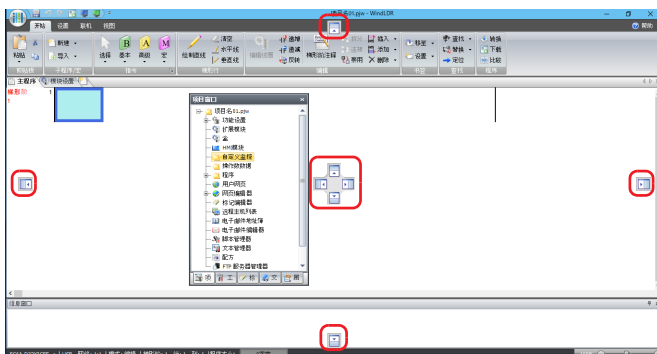


#### 停驻

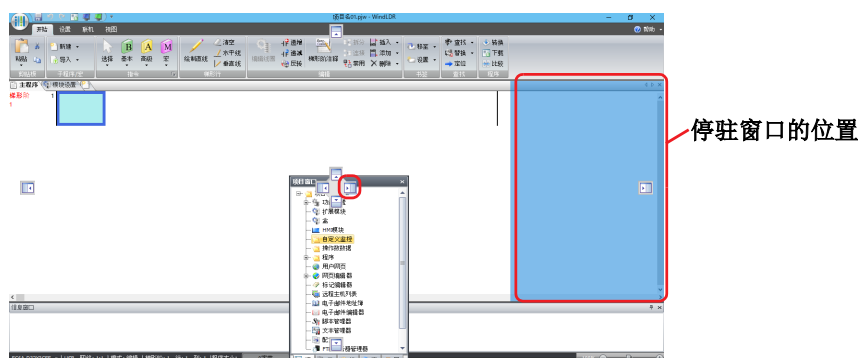
对悬浮窗停靠 WindLDR 上下左右边框或其他窗口的的方法进行介绍。

1. 拖住窗口的标题栏或选项卡。


显示  (停驻) 图标。

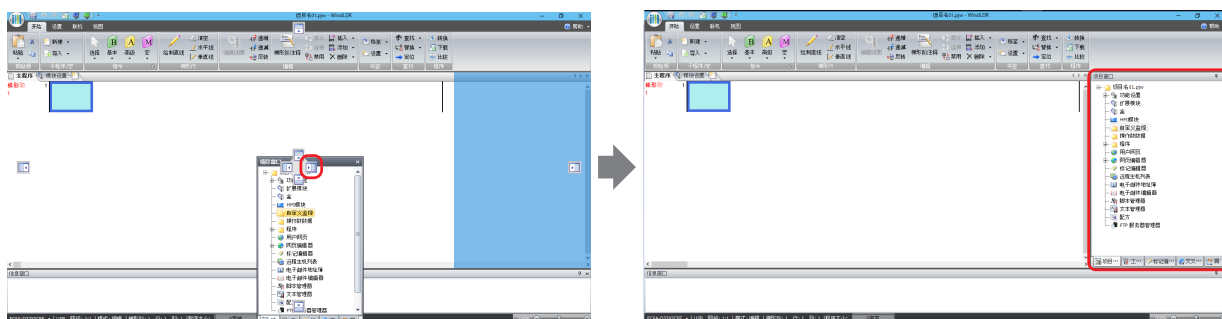



**注释:** 拖住标题栏或选项卡, 将光标靠近  (停驻) 图标, 显示停驻窗口的位置。

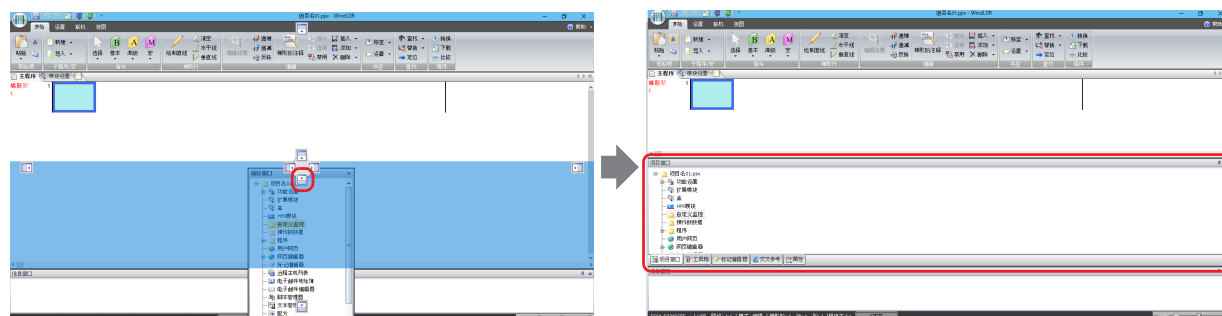




2. 拖到  (停驻) 图标上, 就可以停驻到 WinLDR 的上下左右框中或另外的窗口中。

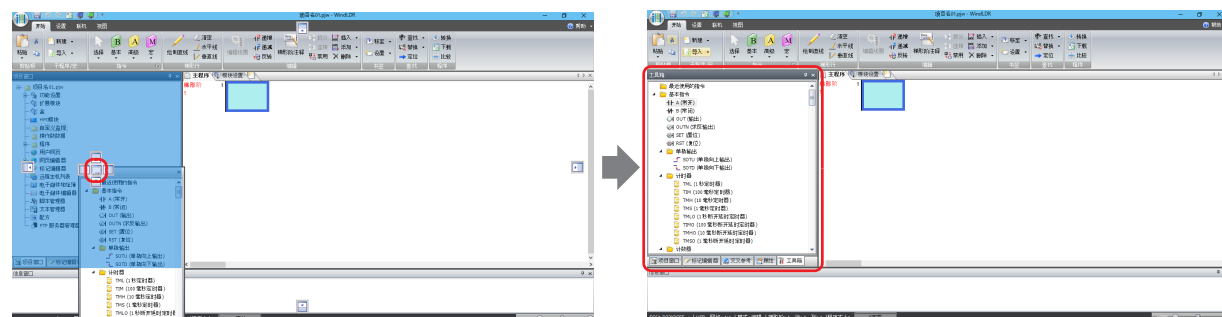
- 将工作区的窗口拖到  (停驻) 图标上时, 停驻在 WinLDR 的上下左右框中。



- 将浮动窗口拖到  (停驻) 图标上时, 停驻在 WinLDR 的上下左右框中或正在停驻的窗口中。



- 拖住浮动窗口的标题栏, 将光标重叠在另外的窗口上, 则显示  (停驻) 图标。拖到  (停驻) 图标上, 则浮动窗口停驻在该窗口上。用选项卡切换窗口显示。

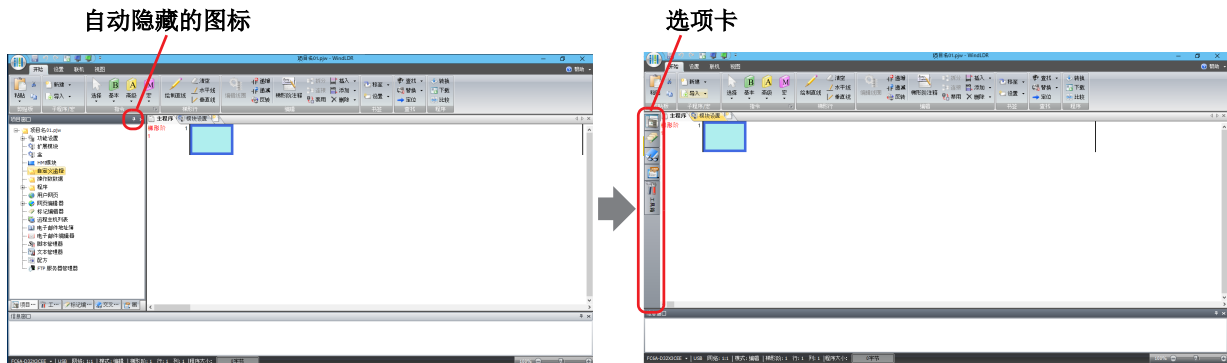


## 1: 基本操作

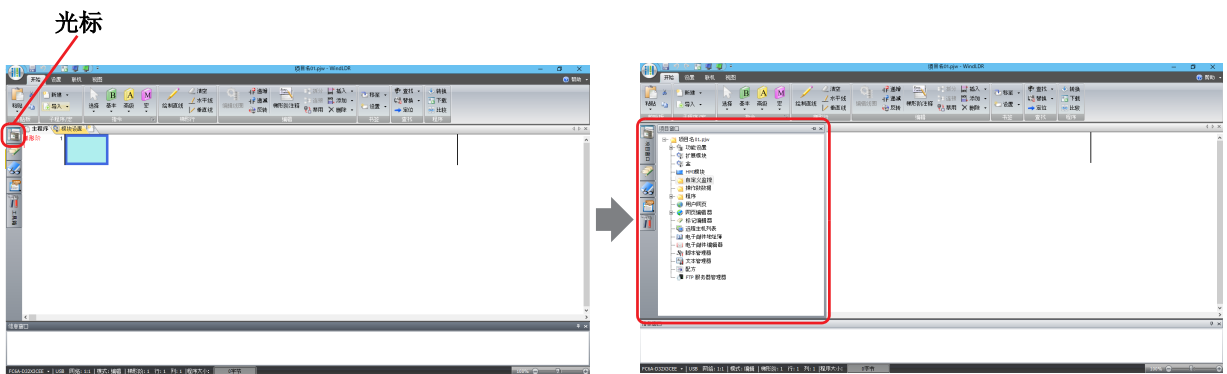
### 变更窗口的显示方法

工作区的窗口已停驻时，可以自动隐藏窗口、切换成只显示选项卡。



- 单击  (自动隐藏) 图标，即切换成只显示选项卡的窗口。



- 光标靠近选项卡就显示窗口。




### 注释:

- 单击  (自动隐藏) 则窗口被固定。
- 单击  (关闭) 则关闭窗口。

## 确认 WindLDR 版本的方法

FT2J FC6A

将对确认 WindLDR 版本的操作步骤进行介绍

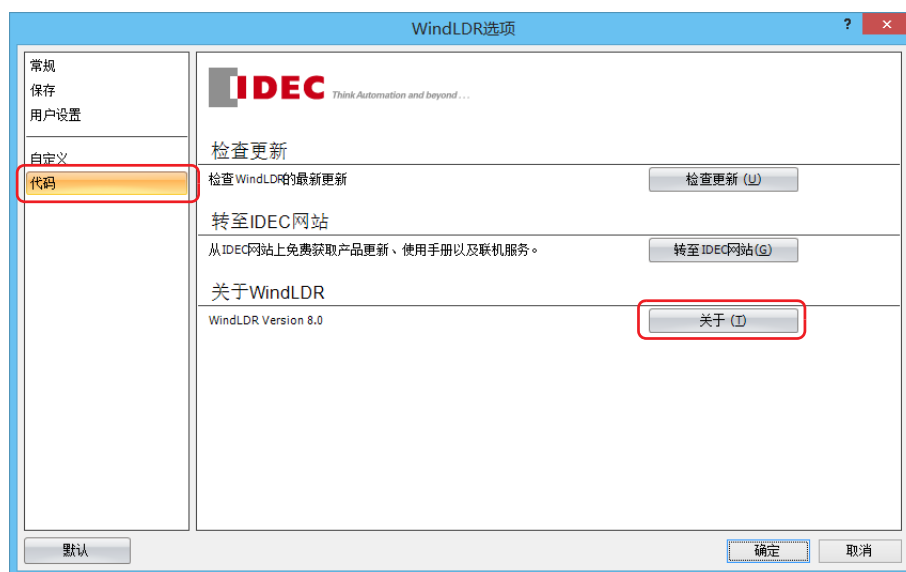
1. 按照 （应用程序）按钮、“WindLDR 选项（I）”的顺序进行单击。

将显示“WindLDR 选项”对话框。



2. 单击“代码”选项卡，然后再单击“关于”按钮。

将显示“关于 WindLDR”对话框。



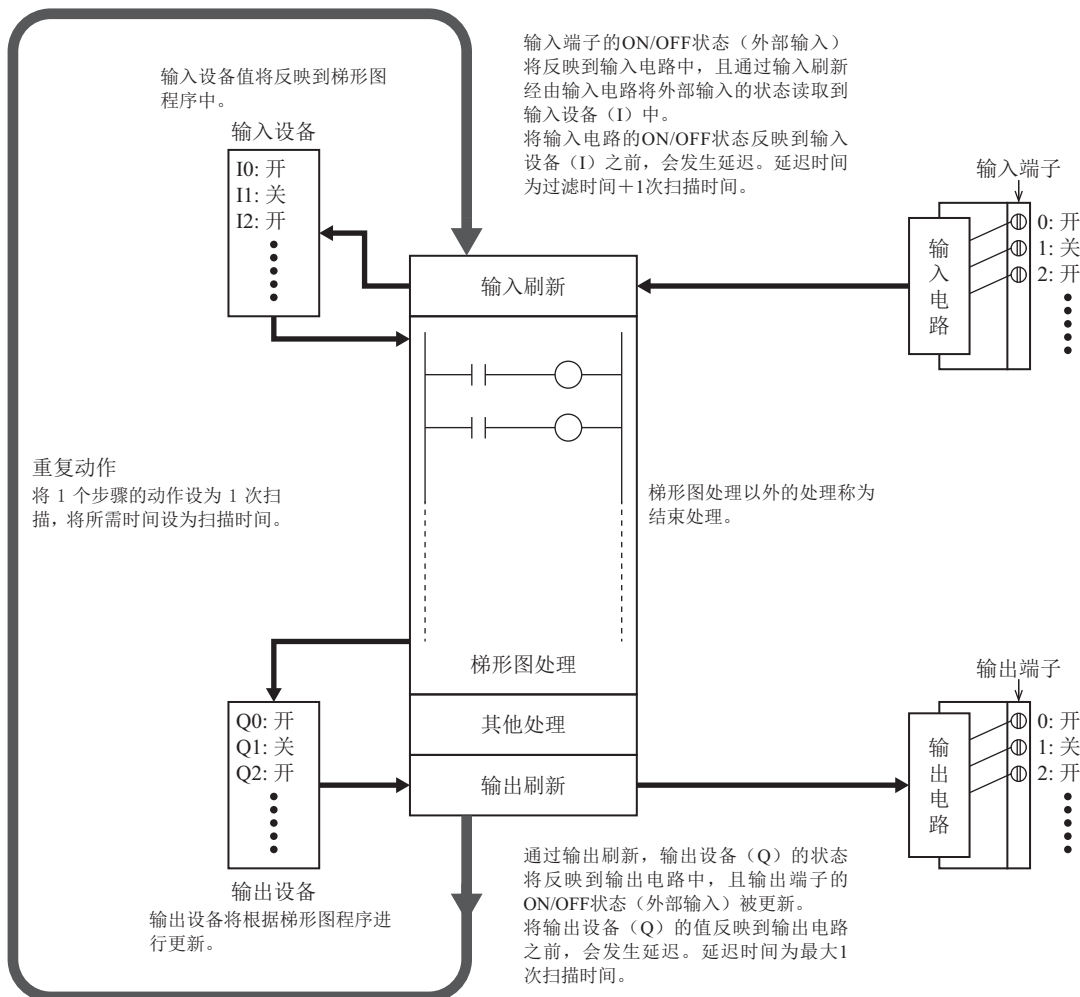
可确认 WindLDR 的版本。结束时，单击“确定”按钮。



梯形图程序的动作

FT2J FC6A

控制器通过以下动作处理梯形图程序。



注释:

- 梯形图程序操作以外的操作被称为结束处理。
- 实际输入及输出的延迟时间还会累加硬件的延迟时间。  
有关硬件的输入及输出延迟时间，请参见以下手册。  
FT2J 型：《SmartAXIS 硬件手册》第 1 章“主机规格”  
FC6A 型：《FC6A 型 MICROSmart 用户手册》第 2 章“产品规格”
- 输入过滤时间取决于所使用的输入。
  - 用于控制器输入  
在 WindLDR 的“功能设置”中进行设置。初始值为 3ms。
  - 用于扩展模块输入（扩展输入）  
没有输入过滤时间。
  - 用于盒输入（扩展输入）  
在“数字 I/O 盒设置”对话框中进行设置。初始值为 3ms。
- 控制器输入和扩展输入的范围因型号而异。详情请参见下一页。  
FT2J 型：第 2-1 页上的“FT2J 型”  
FC6A 型：第 2-12 页上的“FC6A 型”



## 运行 / 停止用户程序

FT2J FC6A

本节介绍运行 / 停止用户程序的操作步骤。

**注意** 运行 / 停止用户程序的操作请务必保证安全性。操作错误可能会导致器械损坏以及事故发生。

用户程序的运行 / 停止可以通过控制器电源打开和关闭、WindLDR、数据文件管理器的操作、停止输入、复位输入以及设备来切换。FT2J 型可以通过操作控制器本身来切换用户程序的运行 / 停止。FC6A 型可以通过操作功能开关或使用 HMI 模块操作菜单来在运行 / 停止之间切换用户程序。

### 控制器接入电源时

#### 通过控制器电源进行运行 / 停止操作

通过打开和关闭控制器电源来运行或停止用户程序

1. 将电源连接至控制器本体的电源端子部。
2. 接入电源后，根据功能设置中的“通电状态下的运行 / 停止功能”的设置运行或停止。  
电源关闭时停止。

**注释：**有关“起动时的 Run/Stop 选择”的详情，请参见以下手册。

FT2J 型：《WindO/I-NV4 用户手册》第 28 章“2.3 通电状态下的运行 / 停止功能”

FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章“通电状态下的运行 / 停止功能”

### 打开控制器电源时

由结束处理切换运行或停止。有关结束处理的详情，请参见第附录 -1 页上的“结束处理时间细分”。关于各方法的用户程序的运行 / 停止，请参见以下手册。

运行 / 停止方法	参照
WindLDR 操作	第 1-20 页上的“通过 WindLDR 的运行 / 停止操作”
控制器本体操作 *1	《WindO/I-NV4 用户手册》第 36 章“1.1 维护画面的概述”
数据文件管理器操作	《数据文件管理器用户手册》第 2 章“3 更改 PLC 运行状态”
停止输入	第 1-20 页上的“通过停止输入、复位输入或功能开关进行运行 / 停止操作”
复位输入	
功能开关 *2	
HMI 模块操作 *2	《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”

\*1 仅限 FT2J 型

\*2 仅限 FC6A 型

#### 注释：

- 如果打开 / 关闭开始控制（M8000），将运行 / 结束用户程序。有关开始控制（M8000）详情，请参见一下手册。  
FT2J 型  
FC6A 型
- 关于停止输入、复位输入或功能开关详情，请参见以下手册。

功能	参照
停止输入	FT2J 型：《WindO/I-NV4 用户手册》第 28 章“3.3 停止输入” FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“停止输入”
复位输入	FT2J 型：《WindO/I-NV4 用户手册》第 28 章“3.4 复位输入” FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“复位输入”
功能开关 *1	《FC6A 型 MICROSmart 用户手册》第 5 章中的“功能开关设置”

\*1 仅限 FC6A 型

## 1: 基本操作

### 通过 WindLDR 的运行 / 停止操作

通过 WindLDR 操作运行或停止用户程序。

1. 运行用户程序时，单击“联机”选项卡上“PLC”中的“启动”。



2. 单击“启动”按钮。
3. 将显示确认消息，并单击“是”按钮。



4. 停止用户程序时，单击“联机”选项卡上“PLC”中的“停止”。
5. 将显示确认消息，并单击“是”按钮。

### 通过停止输入、复位输入或功能开关进行运行 / 停止操作

使用停止输入、复位输入或功能开关时，用户程序将根据这些功能以及开始控制（M8000）的打开 / 关闭来运行或停止。

#### ■ 切换为运行

FT2J 型

停止输入	复位输入	切换为运行
任意设置一个		关闭复位输入以及停止输入，且开始控制（M8000）打开时运行。
未设置		开始控制（M8000）打开时运行。

FC6A 型

停止输入	复位输入	功能开关	切换为运行
设置其中一个输入		无效	关闭复位输入以及停止输入，且开始控制（M8000）打开时运行。
		启用	关闭复位输入以及停止输入时，开始控制（8000）以及功能开关开始运行*1。
未设置		无效	当启动控制 (M8000) 为关于时，运行。
		启用	通过操作开始控制（M8000）和功能开关运行*1。

\*1 有关功能开关的操作的详情，请参见《FC6A 型 MICROsmart 用户手册》第 5 章中的“功能开关设置”。

#### 注释：

- 从停止切换到运行时，将根据“功能设置”中的“内存备份”清除或保留设备状态。有关详情请参见以下手册。  
FT2J 型：《WindO/I-NV4 用户手册》第 28 章“2.4 内存备份”  
FC6A 型：《FC6A 型 MICROsmart 用户手册》第 5 章中的“内存备份”
- 从停止切换到运行时，定时器指令和断开延时定时器指令的当前值将会复位。  
有关定时器指令的详情，请参见第 4-8 页上的“TML（1-s 定时器）”。  
有关断开延时定时器指令的详情，请参见第 4-10 页上的“TMLO（1-s 断开延时定时器）”。
- 如果发生一般错误，则根据发生错误时的操作状态进行运行或停止。有关详情请参见以下手册。  
FT2J 型：《WindO/I-NV4 用户手册》第 37 章“2.3 一般错误”  
FC6A 型：《FC6A 型 MICROsmart 用户手册》第 13 章中的“一般错误”

### ■ 切换至停止

当用户程序停止时，高速计数器 / 频率测量 / 用户中断 / 捕捉输入 / 定时器中断 / 用户通信 / 脉冲输出功能也将停止。

FT2J 型

停止输入	复位输入	切换至运行
设置其中一个输入		当复位或停止输入打开时，停止。 当复位和停止输入关闭且开始控制（M8000）关闭时，停止。  打开复位输入或停止输入，且开始控制（8000）打开时运行。 关闭复位输入以及停止输入，且开始控制（M8000）打开时运行。 当复位或停止输入打开时，停止。
未设置		当开始控制（M8000）关闭时，停止。

FC6A 型

停止输入	复位输入	功能开关	切换至运行
任意设置一个		禁用	当复位和停止输入关闭且开始控制（M8000）关于时，运行。
		启用	当复位和停止输入关闭时，通过操作开始控制（M8000）和功能开关运行*1。
未设置		禁用	当启动控制 (M8000) 为关于时，运行。
		启用	通过操作开始控制（M8000）和功能开关运行*1。

\*1 有关功能开关的操作的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“功能开关设置”。

**注释：**停止时的输出可通过 M8025（停止时维持输出）选择为保持 / 清除。有关 M8025（停止时维持输出）的详情，请参见第 2-22 页上的“M8025：停止时维持输出”。



## 2: 设备地址

本章将对以基本指令或高级指令使用的输入输出、内部继电器、寄存器、定时器、计数器等各种设备的分配、特殊内部继电器及特殊数据寄存器的分配详情进行介绍。

请作为各设备的参考使用。

**注释：**输入及操作控制器的用户程序时，需要具备专业知识。  
请充分理解本手册内容及程序后，再使用控制器。

### FT2J 型

#### 设备列表

设备	符号	单位	范围 (点数)
输入 *1*2	I	位	I0 - I15 (14 点)
扩展输入 *1*3	I	位	I20 - I27 (8 点)
输出 *1*2	Q	位	Q0 - Q7 (8 点)
扩展输出 *1*3	Q	位	Q10 - Q17 (8 点)
内部继电器 *1	M	位	M0 - M7997 (6,400 点)
特殊内部继电器 *1	M	位	M8000 - M8177 (144 点)
移位寄存器	R	位	R0 - R127 (128 点)
定时器	T	位 / 字	T0 - T199 (200 点)
计数器	C	位 / 字	C0 - C199 (200 点)
数据寄存器	D	位 / 字	D0000 - D3999 (4,000 点)
特殊数据寄存器	D	位 / 字	D8000 - D8199 (200 点)
间接寄存器 *4	P	2 个字	P0 - P15 (16 点)

\*1 设备地址的下 1 位是八进制数 (0 ~ 7)。

\*2 FT2J 型主机的输入和输出。

\*3 扩展输入和扩展输出是只有在连接数字 I/O 盒时才能使用的设备。

\*4 只能使用 L (长整) 数据类型。

## 2: 设备地址

### 特殊内部继电器



**警告** 请勿改写特殊内部继电器设备地址中写为保留的区域内的数据。否则系统可能无法正常工作。

**注释:** 读 / 写为读取 / 写入的简称。

读 / 写栏的表述如下所示。

读 / 写: 可读取及写入

读: 仅可读取

写: 仅可写入

设备地址	说明	停止时	电源关闭	读 / 写	
M8000	开始控制	保持	保持	读 / 写	
M8001	1 秒时钟复位	清除	清除	读 / 写	
M8002	所有输出关闭	清除	清除	读 / 写	
M8003	进位 (Cy) 和借位 (Bw)	清除	清除	读	
M8004	梯形图程序执行错误	清除	清除	读	
M8005- M8007	— 保留 —	—	—	—	
M8010	夏时制校正	执行	清除	读	
M8011- M8021	— 保留 —	—	—	—	
M8022	HMI 功能启动完成标志	执行	清除	读	
M8023	时钟数据初始化标志	执行	清除	读	
M8024	BMOV/WSFT 指令执行标志	保持	保持	读	
M8025	CPU 停止时保持输出	保持	清除	读 / 写	
M8026 M8027	— 保留 —	—	—	—	
M8030	高速计数器 (组 1)	比较一致输出复位	清除	清除	读 / 写
M8031		门输入	保持	清除	读 / 写
M8032		复位输入	保持	清除	读 / 写
M8033		复位状态	保持	清除	读
M8034		比较一致状态	保持	清除	读
M8035		上溢出	保持	清除	读
M8036		下溢出	保持	清除	读
M8037		计数方向标记	保持	清除	读
M8040	高速计数器 (组 2)	比较一致输出复位	清除	清除	读 / 写
M8041		门输入	保持	清除	读 / 写
M8042		复位输入	保持	清除	读 / 写
M8043		比较一致状态	保持	清除	读
M8044		上溢出	保持	清除	读
M8045- M8054	— 保留 —	—	—	—	
M8055	高速计数器 (组 3)	比较一致输出复位	清除	清除	读 / 写
M8056		门输入	保持	清除	读 / 写
M8057		复位输入	保持	清除	读 / 写
M8060		比较一致状态	保持	清除	读
M8061		上溢出	保持	清除	读
M8062- M8067	— 保留 —	—	—	—	

设备地址	说明	停止时	电源关闭	读 / 写	
M8070	中断输入状态	组 1	清除	清除	读
M8071		组 2	清除	清除	读
M8072		组 3	清除	清除	读
M8073		组 4	清除	清除	读
M8074		组 5	清除	清除	读
M8075- M8077	— 保留 —	—	—	—	
M8080	中断输入边沿	组 1	清除	清除	读
M8081		组 2	清除	清除	读
M8082		组 3	清除	清除	读
M8083		组 4	清除	清除	读
M8084		组 5	清除	清除	读
M8085- M8087	— 保留 —	—	—	—	
M8090	捕捉输入状态	组 1	保持	清除	读
M8091		组 2	保持	清除	读
M8092		组 3	保持	清除	读
M8093		组 4	保持	清除	读
M8094		组 5	保持	清除	读
M8095- M8117	— 保留 —	—	—	—	
M8120	初始化脉冲	清除	清除	读	
M8121	1 秒时钟	执行	清除	读	
M8122	100 毫秒时钟	执行	清除	读	
M8123	10 毫秒时钟	执行	清除	读	
M8124	定时器 / 计数器预置值变更状态	保持	清除	读	
M8125	梯形图程序执行状态	清除	清除	读	
M8126- M8143	— 保留 —	—	—	—	
M8144	定时器中断状态	清除	清除	读	
M8145- M8147	— 保留 —	—	—	—	
M8150	比较结果	结果 1	保持	清除	读
M8151		结果 2	保持	清除	读
M8152		结果 3	保持	清除	读
M8153- M8165	— 保留 —	—	—	—	
M8166	高速计数器 (组 4)	比较一致输出复位	清除	清除	读 / 写
M8167		门输入	保持	清除	读 / 写
M8170		复位输入	保持	清除	读 / 写
M8171		比较一致状态	保持	清除	读
M8172		上溢出	保持	清除	读
M8173	高速计数器 (组 5)	比较一致输出复位	清除	清除	读 / 写
M8174		门输入	保持	清除	读 / 写
M8175		复位输入	保持	清除	读 / 写
M8176		比较一致状态	保持	清除	读
M8177		上溢出	保持	清除	读

### 特殊内部继电器说明

#### ■ M8000: 开始控制

控制梯形图程序状态（运行 / 停止）。

打开 M8000 时进入运行状态，关闭则进入停止状态。请参见第 1-19 页上的“运行 / 停止用户程序”。但是，功能开关、停止输入及复位输入皆优先于开始控制。虽然 M8000 会在电源关闭时保持其状态，但当关闭时间超过备份时间而使保持数据丢失时，将按照“功能设置”的“在保持数据错误时的运行 / 停止指定”中设置的内容（运行指定 / 停止指定）进行动作。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章中的“2.3 保持数据错误或监视定时器错误时的运行 / 停止指定”。

#### ■ M8001: 1 秒时钟复位

M8001 打开期间，M8121（1 秒时钟）将始终关闭。

#### ■ M8002: 所有输出均关闭

M8002 打开期间，所有输出将关闭。以梯形图程序创建的自保持电路也将关闭。

#### ■ M8003: 进位 (Cy) 和借位 (Bw)

执行高级指令过程中产生进位 (Cy) 或借位 (Bw) 时，M8003 将打开。有关详情，请参见第 3-11 页上的“进位和借位”。

#### ■ M8004: 梯形图程序执行错误

执行梯形图程序过程中发生错误时，M8004 将打开。有关梯形图程序执行错误的详情，请参见附录 -19 页上的“用户程序执行错误和梯形图程序执行错误”。

#### ■ M8010: 夏时制校正

夏令时调整过程中该值为打开。当禁用夏令时或未针对夏令时进行调整时，该值为关闭。

#### ■ M8022: HMI 功能启动完成标志

当 HMI 功能启动完成时，该值变为打开。当重新接通主单元的电源时，该值变为关闭。

#### ■ M8023: 时钟数据初始化标志

当时钟数据初始化发生时，该值变为打开。时钟数据重新设置时关闭。

#### ■ M8024: BMOV/WSFT 指令执行标志

执行 WSFT 指令、BMOV（块传送）指令过程中，M8024 将打开；指令执行完成（正常结束）后将关闭。

#### ■ M8025: CPU 停止时保持输出

当值为打开时，且梯形图程序置于 STOP 状态时，输出 Q 和 WQ 保留其值。当从 STOP 状态进入 RUN 状态时，该值变为关闭。

#### ■ M8030 ~ M8037: 高速计数器（组 1）

用于高速计数器的特殊内部继电器。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

#### ■ M8040 ~ M8044: 高速计数器（组 2）

用于高速计数器的特殊内部继电器。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

#### ■ M8055 ~ M8061: 高速计数器（组 3）

用于高速计数器的特殊内部继电器。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

#### ■ M8070 ~ M8074: 中断输入状态

启用相应的用户中断时，将打开。禁用用户中断时，将关闭。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章“2.7 中断输入”。

#### ■ M8080 ~ M8084: 中断输入边沿

在中断输入的上升沿发生中断时，将打开。在中断输入的下降沿发生中断时，将关闭。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章“2.7 中断输入”。

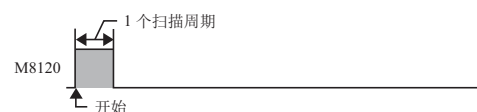
#### ■ M8090 ~ M8094: 捕捉输入状态

在 1 次扫描期间检测捕捉输入中指定的输入接点上升沿 / 下降沿输入时，将读取输入接点的状态。可检测的边沿为 1 个扫描中 1 次。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章“2.6 捕捉输入”。



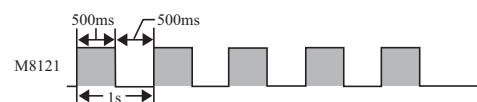
### ■ M8120: 初始化脉冲

仅打开运行（RUN）开始时的 1 次扫描。



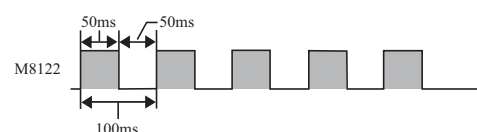
### ■ M8121: 1 秒时钟

M8001 关闭期间，M8121 将以 1s 为周期重复打开和关闭（占空比 1: 1）。



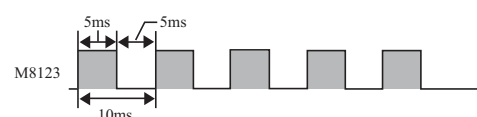
### ■ M8122: 100 毫秒时钟

M8122 以 100ms 为周期重复打开和关闭（占空比 1: 1）。



### ■ M8123: 10 毫秒时钟

M8123 以 10ms 为周期重复打开和关闭（占空比 1: 1）。



### ■ M8124: 定时器 / 计数器预置值变更状态

更改定时器 / 计数器的设置后，该值变为打开。进行下一步操作时，则值变为关闭。

- 下载项目
- 清除更改的预置值
- 打开 FT2J 型号时
- 切换为系统模式

### ■ M8125: 梯形图程序执行状态

处于运行状态期间，将打开。

### ■ M8144: 定时器中断状态

启用定时器中断时，将打开。禁用定时器中断时，将关闭。

### ■ M8150 ~ M8152: 比较结果

M8150 ~ M8152 将根据 CMP= 指令、ICMP>= 指令的比较结果打开。

有关详情，请参见第 6-6 页上的“特殊内部继电器 M8150、M8151 和 M8152 用于 ICMP>=”

### ■ M8166 ~ M8172: 高速计数器（组 4）

用于高速计数器的特殊内部继电器。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

### ■ M8173 ~ M8177: 高速计数器（组 5）

用于高速计数器的特殊内部继电器。有关详情，请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

## 2: 设备地址

### 特殊数据寄存器



**警告** 请勿改写特殊数据寄存器一览中写为保留的区域内的数据。否则系统可能无法正常工作。

**注释:** 读 / 写为读取 / 写入的简称。

读 / 写栏的表述如下所示。

读 / 写: 可读取及写入

读: 仅可读取

写: 仅可写入

设备地址	说明		刷新时间	读 / 写
D8000 D8001	— 保留 —		—	—
D8002	机型信息		通电时	读
D8003 D8004	— 保留 —		—	—
D8005	通用错误代码		发生错误时	读 / 写
D8006	程序执行错误代码		发生错误时	读
D8007	— 保留 —		—	—
D8008	内部时钟的数据 (当前值)	年	每 500ms	读
D8009		月	每 500ms	读
D8010		日	每 500ms	读
D8011		星期 0: 星期日、1: 星期一、2: 星期二、 3: 星期三、4: 星期四、5: 星期五、 6: 星期六	每 500ms	读
D8012		小时	每 500ms	读
D8013		分钟	每 500ms	读
D8014		秒	每 500ms	读
D8015- D8021		— 保留 —		—
D8022	固定扫描时间预置值		—	读 / 写
D8023	扫描时间数据 (当前值)		每次扫描	读
D8024	扫描时间数据 (最大值)		更新时	读
D8025	扫描时间数据 (最小值)		更新时	读
D8026- D8030	— 保留 —		—	—
D8031	选项连接信息		通电时	读
D8032	中断输入跳转目标标签编号	组 1	—	读 / 写
D8033		组 2	—	读 / 写
D8034		组 3	—	读 / 写
D8035		组 4	—	读 / 写
D8036	定时器中断跳转目标标签编号		—	读 / 写
D8037	中断输入跳转目标标签编号	组 5	—	读 / 写
D8038 D8039	— 保留 —		—	—
D8040	模拟量输入值	AI0	每次扫描	读
D8041		AI1	每次扫描	读
D8042	模拟量输入状态	AI0	每次扫描	读
D8043		AI1	每次扫描	读
D8044	模拟量输出值	AQ0	每次扫描	读 / 写
D8045		AQ1	每次扫描	读 / 写

设备地址	说明		刷新时间	读 / 写	
D8046	模拟量输出状态	AQ0	每次扫描	读	
D8047		AQ1	每次扫描	读	
D8048 D8049	— 保留 —		—	—	
D8050	高速计数器 (组 1)	高位字	当前值	每次扫描	读
D8051		低位字		每次扫描	读
D8052		高位字	预置值	—	读 / 写
D8053		低位字		—	读 / 写
D8054		高位字	复位值	—	读 / 写
D8055		低位字		—	读 / 写
D8056	高速计数器 / 频率测量 (组 2)	高位字	当前值 / 频率测量当前值	每次扫描	读
D8057		低位字		每次扫描	读
D8058		高位字	预置值	—	读 / 写
D8059		低位字		—	读 / 写
D8060		高位字	复位值	—	读 / 写
D8061		低位字		—	读 / 写
D8062- D8067	— 保留 —		—	—	
D8068	高速计数器 / 频率测量 (组 3)	高位字	当前值 / 频率测量当前值	每次扫描	读
D8069		低位字		每次扫描	读
D8070		高位字	预置值	—	读 / 写
D8071		低位字		—	读 / 写
D8072		高位字	复位值	—	读 / 写
D8073		低位字		—	读 / 写
D8074- D8076	— 保留 —		—	—	
D8077	模拟量输入超出范围错误状态		每次扫描	读	
D8078- D8121	— 保留 —		—	—	
D8122	盒插槽信息 (Slot1)	类型 ID/ 状态	通电时	读	
D8123		系统软件版本	通电时	读	
D8124	盒插槽信息 (Slot2)	类型 ID/ 状态	通电时	读	
D8125		系统软件版本	通电时	读	
D8126- D8133	— 保留 —		—	—	
D8134	高速计数器 / 频率测量 (组 4)	高位字	当前值 / 频率测量当前值	每次扫描	读
D8135		低位字		每次扫描	读
D8136		高位字	预置值	—	读 / 写
D8137		低位字		—	读 / 写
D8138		高位字	复位值	—	读 / 写
D8139		低位字		—	读 / 写
D8140	高速计数器 (组 5)	高位字	当前值	每次扫描	读
D8141		低位字		每次扫描	读
D8142		高位字	预置值	—	读 / 写
D8143		低位字		—	读 / 写
D8144		高位字	复位值	—	读 / 写
D8145		低位字		—	读 / 写
D8146- D8171	— 保留 —		—	—	
D8172	模拟量输入值	AI2	每次扫描	读	
D8173		AI3	每次扫描	读	

## 2: 设备地址

设备地址	说明	刷新时间	读 / 写
D8174	模拟量输入状态	AI2	读
D8175		AI3	读
D8176	模拟量 I/O 盒数据 (Slot1)	AI4/AQ2	读 / 写
D8177		AI5/AQ3	读 / 写
D8178	模拟量 I/O 盒状态 (Slot1)	AI4/AQ2	读
D8179		AI5/AQ3	读
D8180- D8185	— 保留 —	—	—
D8186	模拟量 I/O 盒数据 (Slot2)	AI6/AQ4	读 / 写
D8187		AI7/AQ5	读 / 写
D8188	模拟量 I/O 盒状态 (Slot2)	AI6/AQ4	读
D8189		AI7/AQ5	读
D8190- D8199	— 保留 —	—	—

## 特殊数据寄存器补充说明

## ■ D8002: 机型信息

写入控制器的机型信息。

3 (03h): FT2J 型

## ■ D8005: 通用错误代码

写入 FT2J 型的通用错误代码。有关通用错误代码的详情, 请参见《WindO/I-NV4 用户手册》第 37 章中的“控制功能错误”。

## ■ D8006: 程序执行错误代码

FT2J 型写入梯形图程序执行错误信息。有关梯形图程序执行错误的详情, 请参见附录 -19 页上的“用户程序执行错误和梯形图程序执行错误”。

## ■ D8008 ~ D8014: 内部时钟的数据 (当前值)

存储 FT2J 型内部时钟的日期和时间数据。

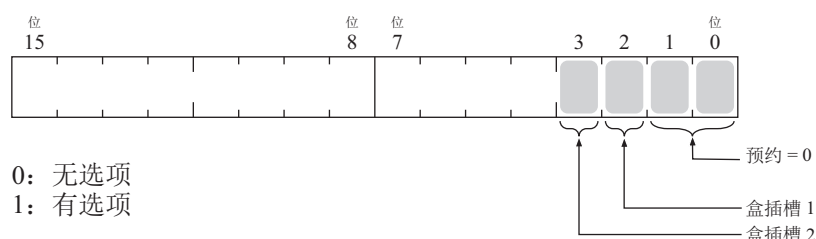
## ■ D8022 ~ D8025: 扫描时间数据

D8022 到 D8025 都是特殊数据寄存器, 用于检查扫描时间和设置常量扫描时间。有关扫描时间的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“3 固定扫描时间”。

## ■ D8031: 选项连接信息

写入选项连接信息。

设备内的分配 (位分配) 如下所示。



## ■ D8032 ~ D8035、D8037: 中断输入跳转目标标签编号

写入中断输入的跳转目标标签编号。有关中断输入的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.7 中断输入”。

## ■ D8036: 定时器中断跳转目标标签编号

写入发生定时器中断时的跳转目标标签编号。有关定时器中断的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.12 定时器中断”。

## ■ D8040、D8041: 模拟量输入值

FT2J 型模拟量输入端的模拟量值被转换为数字值并写入该寄存器。有关详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.10 模拟量 / 数字量输入”。

## ■ D8042、D8043: 模拟量输入状态

FT2J 型模拟量输入端的模拟量输入状态写入该寄存器。有关详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.10 模拟量 / 数字量输入”。

## ■ D8044、D8045: 模拟量输出值

特殊数据寄存器中存储的数字值转换为模拟量值, 并从 FT2J 型模拟量输出端子输出。有关详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.11 模拟量输出”。

## ■ D8046、D8047: 模拟量输出状态

存储 FT2J 型模拟量输出端子的模拟量输出状态。有关详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.11 模拟量输出”。

## ■ D8050 ~ D8055: 高速计数器 (组 1)

这些特殊数据寄存器用于高速计数器功能。

有关高速计数器的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

## 2: 设备地址

### ■ D8056 ~ D8061: 高速计数器 / 频率测量 (组 2)

这些特殊数据寄存器用于高速计数器功能和频率测量功能。

有关高速计数器的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”和“2.8 频率测量”。

### ■ D8068 ~ D8073: 高速计数器 / 频率测量 (组 3)

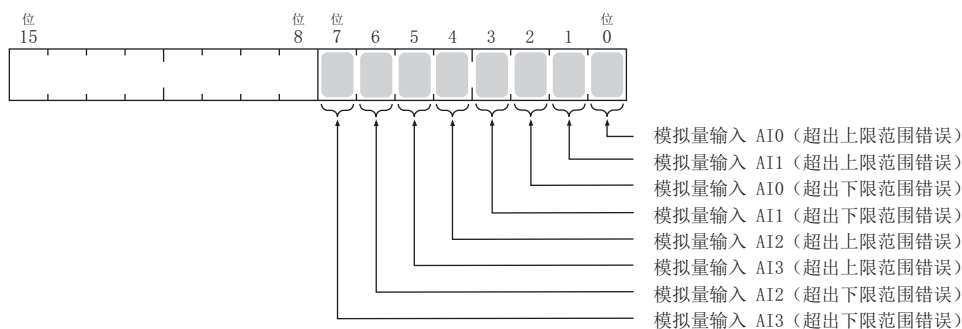
这些特殊数据寄存器用于高速计数器功能和频率测量功能。

有关高速计数器的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”和“2.8 频率测量”。

### ■ D8077: 模拟量输入超出范围错误状态

如果 FT2J 型模拟量输入端子上的模拟量输入信号超过 11V 或 21mA (超出上限范围错误) 或低于 2mA (超出下限范围错误), 则相关位的值设为 1。如果高于 2mA 且低于 11V 或 21mA, 则值设为 0。

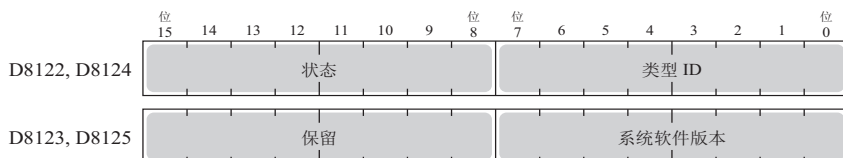
各个模拟量输入的分配如下所示。



### ■ D8122 ~ D8125: 盒插槽信息

写入盒插槽 1 和 2 的信息。

信息的分配 (位分配) 如下所示。



D8122、D8123 = 盒插槽信息 (Slot1)

D8124、D8125 = 盒插槽信息 (Slot2)

有关类型 ID 和状态的详细信息, 请参见第 2-57 页, "类型 ID 和状态列表"。

### ■ D8134 ~ D8139: 高速计数器 / 频率测量 (组 4)

这些特殊数据寄存器用于高速计数器功能和频率测量功能。

有关高速计数器的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”和“2.8 频率测量”。

### ■ D8140 ~ D8145: 高速计数器 (组 5)

这些特殊数据寄存器用于高速计数器功能。

有关高速计数器的详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”。

### ■ D8172、D8173: 模拟量输入值

FT2J 型模拟量输入端的模拟量值被转换为数字值并写入该寄存器。有关详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.10 模拟量 / 数字量输入”。

### ■ D8174、D8175: 模拟量输入状态

FT2J 型模拟量输入端的模拟量输入状态写入该寄存器。有关详情, 请参见《WindO/I-NV4 用户手册》第 28 章中的“2.10 模拟量 / 数字量输入”。

### ■ D8176、D8177: 模拟量 I/O 盒数据 (Slot1)

写入模拟 I/O 盒的模拟量输入输出值。

模拟量输入型时: 将模拟 I/O 盒中输入的模拟量值转换为数字值进行写入。

模拟量输出型时: 将存储为数字值的数字值转换为模拟量值从模拟 I/O 盒进行输出。

有关详情, 请参见《SmartAXIS 硬件手册》第 2 章中的“2 模拟量 I/O 盒”。

■ **D8178、D8179：模拟量 I/O 盒状态 (Slot1)**

写入模拟量 I/O 盒的模拟状态。

有关详情，请参见《SmartAXIS 硬件手册》第 4 章“状态”。

■ **D8186、D8187：模拟量 I/O 盒数据 (Slot2)**

写入模拟 I/O 盒的模拟量输入输出值。

模拟量输入型时：将模拟 I/O 盒中输入的模拟量值转换为数字值进行写入。

模拟量输出型时：将存储为数字值的数字值转换为模拟量值从模拟 I/O 盒进行输出。

有关详情，请参见《SmartAXIS 硬件手册》第 2 章中的“2 模拟量 I/O 盒”。

■ **D8188、D8189：模拟量 I/O 盒状态 (Slot2)**

写入模拟量 I/O 盒的模拟状态。

有关详情，请参见《SmartAXIS 硬件手册》第 4 章“状态”。

## 2: 设备地址

### FC6A 型

#### 设备列表

#### All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块

设备	符号	单位	范围 (点数)		
			16-I/O 型	24-I/O 型	40-I/O 型
输入 *1	I	位	I0 - I10 (9 点)	I0 - I15 (14 点)	I0 - I27 (24 点)
扩展输入 *1	I	位	I30 - I187 (128 点) I190 - I507*2 (256 点) I630 - I633*4 (4 点)	I30 - I307 (224 点) I310 - I627*3 (256 点) I630 - I633*4 (4 点)	I30 - I307 (224 点) I310 - I627*3 (256 点) I630 - I637*4 (8 点)
输出 *1	Q	位	Q0 - Q6 (7 点)	Q0 - Q11 (10 点)	Q0 - Q17 (16 点)
扩展输出 *1	Q	位	Q30 - Q187 (128 点) Q190 - Q507*2 (256 点) Q630 - Q633*4 (4 点)	Q30 - Q307 (224 点) Q310 - Q627*3 (256 点) Q630 - Q633*4 (4 点)	Q30 - Q307 (224 点) Q310 - Q627*3 (256 点) Q630 - Q637*4 (8 点)
内部继电器 *1	M	位	M0 - M7997 (6,400 点) M10000 - M17497 (6,000 点)		
特殊内部继电器 *1	M	位	M8000 - M8317 (256 点)		
移位寄存器	R	位	R0 - R255 (256 点)		
定时器	T	位 / 字	T0 - T1023 (1,024 点)		
计数器	C	位 / 字	C0 - C511 (512 点)		
数据寄存器	D	位 / 字	D0000 - D7999 (8,000 点) D10000 - D55999 (46,000 点)		
特殊数据寄存器	D	位 / 字	D8000 - D8499 (500 点)		

\*1 设备地址的下 1 位是八进制数 (0 ~ 7)。

\*2 I190 ~ I507 及 Q190 ~ Q507 为可用于使用增设扩展模块连接扩展模块 (增设扩展侧) 时的设备。

\*3 I310 ~ I627 及 Q310 ~ Q627 为可用于使用增设扩展模块连接扩展模块 (增设扩展侧) 时的设备。

\*4 I630 ~ I637 及 Q630 ~ Q637 为可用于连接 I/O 盒时的设备。



## Plus CPU 模块

设备	符号	单位	范围 (点数)	
			Plus 16-I/O 型	Plus 32-I/O 型
输入 *1	I	位	I0 - I7 (8 点)	I0 - I17 (16 点)
扩展输入 *1	I	位	I30 - I307 (224 点) I310 - I627*2 (256 点) I630 - I643*3 (12 点) I1000 - I10597*4 (2,016 点)	
输出 *1	Q	位	Q0 - Q7 (8 点)	Q0 - Q17 (16 点)
扩展输出 *1	Q	位	Q30 - Q307 (224 点) Q310 - Q627*2 (256 点) Q630 - Q643*3 (12 点) Q1000 - Q10597*4 (2,016 点)	
内部继电器 *1	M	位	M0 - M7997 (6,400 点) M10000 - M21247 (9,000 点)	
特殊内部继电器 *1	M	位	M8000 - M9997 (1,600 点)	
移位寄存器	R	位	R0 - R255 (256 点)	
定时器	T	位 / 字	T0 - T1999 (2,000 点)	
计数器	C	位 / 字	C0 - C511 (512 点)	
数据寄存器	D	位 / 字	D0000 - D7999 (8,000 点) D10000 - D61999 (52,000 点)	
非保持型数据寄存器	D	位 *7 / 字	D70000 - D269999*5 (200,000 点)	
特殊数据寄存器	D	位 / 字	D8000 - D8899 (900 点)	
间接寄存器 *6	P	2 个字	P0 - P15 (16 点)	

\*1 设备地址的下 1 位是八进制数 (0 ~ 7)。

\*2 I310 ~ I627 及 Q310 ~ Q627 为可用于使用增设扩展模块连接扩展模块 (增设扩展侧) 时的设备。(节点 0)

\*3 I630 ~ I643 及 Q630 ~ Q643 为可用于连接 I/O 盒时的设备。

\*4 I1000 ~ I10597 及 Q1000 ~ Q10597 为可用于使用增设扩展模块组合型主机 / 组合型从机及增设扩展模块一体型连接扩展模块时的设备。(节点 1 ~ 10)

\*5 D70000 ~ D269999 无法指定保持。停止 → 运行过程中虽然会保持, 但通电时将复位为 0。

\*6 可使用的数据类型仅为 L (长整)。

\*7 仅 SCRPT 指令执行的脚本内和 UMACRO 指令使用的参数设备可使用。

## 2: 设备地址

### 特殊内部继电器



**警告** 请勿改写特殊内部继电器设备地址中写为保留的区域内的数据。否则系统可能无法正常工作。

**注释:** 读 / 写为读取 / 写入的简称。

读 / 写栏的表述如下所示。

读 / 写: 可读取及写入

读: 仅可读取

写: 仅可写入

设备地址	说明	停止时	电源关闭	读 / 写	
M8000	开始控制	保持	保持	读 / 写	
M8001	1 秒时钟复位	清除	清除	读 / 写	
M8002	所有输出关闭	清除	清除	读 / 写	
M8003	进位 (Cy) 或借位 (Bw)	清除	清除	读 / 写	
M8004	用户程序执行错误	清除	清除	读 / 写	
M8005	通信错误	保持	清除	读 / 写	
M8006	通信禁止标记 (数据连接主机时)	保持	保持	读 / 写	
M8007	初始化标记 (数据连接主机时) / 通信停止标记 (数据连接从机时)	清除	清除	读 / 写	
M8010	状态 LED 执行	执行	清除	读 / 写	
M8011	— 保留 —	—	—	—	
M8012					
M8013	日历 / 时钟数据写入 / 调整错误标记	执行	清除	读 / 写	
M8014	日历 / 时钟数据读取错误标记	执行	清除	读 / 写	
M8015	— 保留 —	—	—	—	
M8016	日历数据写入标记	执行	清除	读 / 写	
M8017	时钟数据写入标记	执行	清除	读 / 写	
M8020	日历 / 时钟数据写入标记	执行	清除	读 / 写	
M8021	时钟数据调整标记	执行	清除	读 / 写	
M8022	用户通信接收指令取消标记 (端口 1)	清除	清除	读 / 写	
M8023	用户通信接收指令取消标记 (端口 2)	清除	清除	读 / 写	
M8024	BMOV/WSFT 执行标记	保持	保持	读 / 写	
M8025	停止时维持输出	保持	清除	读 / 写	
M8026	用户通信接收指令取消标记 (端口 3)	清除	清除	读 / 写	
M8027	高速计数器 (组 1/10)	计数方向标记	保持	清除	读 / 写
M8030		比较一致输出复位	清除	清除	读 / 写
M8031		门输入	保持	清除	读 / 写
M8032		复位输入	保持	清除	读 / 写
M8033	用户通信接收指令取消标记 (端口 4)	清除	清除	读 / 写	
M8034	高速计数器 (组 3/13)	比较一致输出复位	清除	清除	读 / 写
M8035		门输入	保持	清除	读 / 写
M8036		复位输入	保持	清除	读 / 写
M8037	— 保留 —	—	—	—	
M8040	高速计数器 (组 4/14)	比较一致输出复位	清除	清除	读 / 写
M8041		门输入	保持	清除	读 / 写
M8042		复位输入	保持	清除	读 / 写
M8043	高速计数器 (组 5/16)	计数方向标记	保持	清除	读 / 写
M8044		比较一致输出复位	清除	清除	读 / 写
M8045		门输入	保持	清除	读 / 写
M8046		复位输入	保持	清除	读 / 写
M8047	— 保留 —	—	—	—	
M8050					

设备地址	说明	停止时	电源关闭	读 / 写	
M8051	高速计数器 (组 2/I1)	比较一致输出复位	清除	清除	读 / 写
M8052		门输入	保持	清除	读 / 写
M8053		复位输入	保持	清除	读 / 写
M8054		比较打开状态	保持	清除	读
M8055		上溢出	保持	清除	读
M8056	— 保留 —	—	—	—	
M8057	高速计数器 (组 6/I7)	比较一致输出复位	清除	清除	读 / 写
M8060		门输入	保持	清除	读 / 写
M8061		复位输入	保持	清除	读 / 写
M8062		比较打开状态	保持	清除	读
M8063		上溢出	保持	清除	读
M8064- M8067	— 保留 —	—	—	—	
M8070	SD 记忆卡固定状态	保持	清除	读	
M8071	SD 记忆卡写入标记	保持	清除	读	
M8072	SD 记忆卡固定解除	执行	清除	读 / 写	
M8073	功能开关状态	执行	清除	读	
M8074	电池电压测量标记	执行	清除	读 / 写	
M8075- M8077	— 保留 —	—	—	—	
M8080	数据连接从机 1 通信完成继电器 (数据连接主机时)	执行	清除	读	
M8081	数据连接从机 2 通信完成继电器	执行	清除	读	
M8082	数据连接从机 3 通信完成继电器	执行	清除	读	
M8083	数据连接从机 4 通信完成继电器	执行	清除	读	
M8084	数据连接从机 5 通信完成继电器	执行	清除	读	
M8085	数据连接从机 6 通信完成继电器	执行	清除	读	
M8086	数据连接从机 7 通信完成继电器	执行	清除	读	
M8087	数据连接从机 8 通信完成继电器	执行	清除	读	
M8090	数据连接从机 9 通信完成继电器	执行	清除	读	
M8091	数据连接从机 10 通信完成继电器	执行	清除	读	
M8092	数据连接从机 11 通信完成继电器	执行	清除	读	
M8093	数据连接从机 12 通信完成继电器	执行	清除	读	
M8094	数据连接从机 13 通信完成继电器	执行	清除	读	
M8095	数据连接从机 14 通信完成继电器	执行	清除	读	
M8096	数据连接从机 15 通信完成继电器	执行	清除	读	
M8097	数据连接从机 16 通信完成继电器	执行	清除	读	
M8100	数据连接从机 17 通信完成继电器	执行	清除	读	
M8101	数据连接从机 18 通信完成继电器	执行	清除	读	
M8102	数据连接从机 19 通信完成继电器	执行	清除	读	
M8103	数据连接从机 20 通信完成继电器	执行	清除	读	
M8104	数据连接从机 21 通信完成继电器	执行	清除	读	
M8105	数据连接从机 22 通信完成继电器	执行	清除	读	
M8106	数据连接从机 23 通信完成继电器	执行	清除	读	
M8107	数据连接从机 24 通信完成继电器	执行	清除	读	
M8110	数据连接从机 25 通信完成继电器	执行	清除	读	
M8111	数据连接从机 26 通信完成继电器	执行	清除	读	
M8112	数据连接从机 27 通信完成继电器	执行	清除	读	
M8113	数据连接从机 28 通信完成继电器	执行	清除	读	
M8114	数据连接从机 29 通信完成继电器	执行	清除	读	
M8115	数据连接从机 30 通信完成继电器	执行	清除	读	

## 2: 设备地址

设备地址	说明		停止时	电源关闭	读 / 写
M8116	数据连接从机 31 通信完成继电器		执行	清除	读
M8117	数据连接全部从机通信完成继电器		执行	清除	读
M8120	初始化脉冲		清除	清除	读
M8121	1 秒时钟		执行	清除	读
M8122	100 毫秒时钟		执行	清除	读
M8123	10 毫秒时钟		执行	清除	读
M8124	定时器 / 计数器预置值更改		保持	清除	读
M8125	动作中输出		清除	清除	读
M8126	程序运行中下载完成后打开 1 次扫描		清除	清除	读
M8127	— 保留 —		—	—	—
M8130	高速计数器 (组 1/10)	复位状态	保持	清除	读
M8131		比较打开状态	保持	清除	读
M8132	— 保留 —		—	—	—
M8133	高速计数器 (组 3/13)	比较打开状态	保持	清除	读
M8134	高速计数器 (组 4/14)	比较打开状态	保持	清除	读
M8135	高速计数器 (组 5/16)	复位状态	保持	清除	读
M8136		比较打开状态	保持	清除	读
M8137	中断输入 I0 状态 (组 1/10)	(ON: 已允许, OFF: 已禁止)	清除	清除	读
M8140	中断输入 I1 状态 (组 2/11)		清除	清除	读
M8141	中断输入 I3 状态 (组 3/13)		清除	清除	读
M8142	中断输入 I4 状态 (组 4/14)		清除	清除	读
M8143	中断输入 I6 状态 (组 5/16)		清除	清除	读
M8144	定时器中断状态		清除	清除	读
M8145	用户通信接收指令取消标记 (端口 5)		清除	清除	读 / 写
M8146	用户通信接收指令取消标记 (端口 6)		清除	清除	读 / 写
M8147	用户通信接收指令取消标记 (端口 7)		清除	清除	读 / 写
M8150	比较结果 1		保持	清除	读
M8151	比较结果 2		保持	清除	读
M8152	比较结果 3		保持	清除	读
M8153	捕捉输入开 / 关状态	组 1/10	保持	清除	读
M8154		组 2/11	保持	清除	读
M8155		组 3/13	保持	清除	读
M8156		组 4/14	保持	清除	读
M8157		组 5/16	保持	清除	读
M8160		组 6/17	保持	清除	读
M8161	高速计数器 (组 1/10)	上溢出	保持	清除	读
M8162		下溢出	保持	清除	读
M8163	高速计数器 (组 5/16)	上溢出	保持	清除	读
M8164		下溢出	保持	清除	读
M8165	高速计数器 (组 3/13)	上溢出	保持	清除	读
M8166	高速计数器 (组 4/14)	上溢出	保持	清除	读
M8167	中断输入 I7 状态 (组 6/17)	(ON: 已允许, OFF: 已禁止)	保持	清除	读
M8170	用户通信接收指令取消标记 (端口 8)		清除	清除	读 / 写
M8171	— 保留 —		—	—	—
M8172	晶体管源型输出过电流检测	组 1	执行	清除	读
M8173		组 2	执行	清除	读
M8174		组 3	执行	清除	读
M8175		组 4	执行	清除	读
M8176	用户通信接收指令取消标记 (端口 9)		清除	清除	读 / 写

设备地址	说明	停止时	电源关闭	读 / 写
M8177- M8183	— 保留 —	—	—	—
M8184	HMI 模块 网络设置更改触发器	执行	清除	读 / 写
M8185	在夏时制时间段	执行	清除	读
M8186	以太网端口 1 自动 Ping 执行中	执行	清除	读
M8187	以太网端口 1 自动 Ping 停止标记	执行	清除	读 / 写
M8190	CPU 模块 以太网端口 1 网络设置更改触发器	执行	清除	读 / 写
M8191	SNTP 获取标记	执行	清除	读 / 写
M8192	中断输入 I0 边沿	ON: 上升沿 OFF: 下降沿	清除	读
M8193	中断输入 I3 边沿		清除	读
M8194	中断输入 I4 边沿		清除	读
M8195	中断输入 I6 边沿		清除	读
M8196	中断输入 I7 边沿		清除	读
M8197	中断输入 I1 边沿		清除	读
M8200	用户通信接收指令取消标记	连接 1	清除	读 / 写
M8201		连接 2	清除	读 / 写
M8202		连接 3	清除	读 / 写
M8203		连接 4	清除	读 / 写
M8204		连接 5	清除	读 / 写
M8205		连接 6	清除	读 / 写
M8206		连接 7	清除	读 / 写
M8207		连接 8	清除	读 / 写
M8210	— 保留 —	—	—	—
M8211	初始化 HMI 模块电子邮件发送服务器设置	执行	清除	读 / 写
M8212	连接状态 (ON: 已接通, OFF: 未接通)	连接 1	执行	读
M8213		连接 2	执行	读
M8214		连接 3	执行	读
M8215		连接 4	执行	读
M8216		连接 5	执行	读
M8217		连接 6	执行	读
M8220		连接 7	执行	读
M8221		连接 8	执行	读
M8222	断开用户通信连接	连接 1	执行	读 / 写
M8223		连接 2	执行	读 / 写
M8224		连接 3	执行	读 / 写
M8225		连接 4	执行	读 / 写
M8226		连接 5	执行	读 / 写
M8227		连接 6	执行	读 / 写
M8230		连接 7	执行	读 / 写
M8231		连接 8	执行	读 / 写
M8232	HMI 模块连接信息参考 连接状态	执行	清除	读
M8233- M8247	— 保留 —	—	—	—
M8250	从 SD 记忆卡下载执行标记	执行	清除	读 / 写
M8251	向 SD 记忆卡上传执行标记	执行	清除	读 / 写
M8252	SD 记忆卡下载执行中	执行	清除	读
M8253	SD 记忆卡上传执行中	执行	清除	读
M8254	SD 记忆卡下载 / 上传执行完成输出	执行	清除	读
M8255	SD 记忆卡下载 / 上传执行错误输出	执行	清除	读

## 2: 设备地址

设备地址	说明		停止时	电源关闭	读 / 写
M8256 M8257	— 保留 —		—	—	—
M8260	写入配方执行标记		执行	清除	读 / 写
M8261	读取配方执行标记		执行	清除	读 / 写
M8262	写入配方执行中		执行	清除	读 / 写
M8263	读取配方执行中		执行	清除	读 / 写
M8264	配方频道执行完成输出		执行	清除	读 / 写
M8265	执行配方错误输出		执行	清除	读 / 写
M8266	配方块执行完成输出		执行	清除	读 / 写
M8267	配方内存 (ROM- 区域) 读取限制		执行	清除	读
M8270	— 保留 —		—	—	—
M8271	从 SD 记忆卡进行 MQTT 基本设置下载的执行标记		执行	清除	读 / 写
M8272	从 SD 记忆卡进行 MQTT 基本设置下载的执行完成输出		执行	清除	读
M8273	从 SD 记忆卡进行 MQTT 基本设置下载的错误输出		执行	清除	读
M8274- M8297	— 保留 —		—	—	—
M8300	J1939 通信许可标记		清除	清除	读 / 写
M8301	J1939 联机状态		清除	清除	读
M8302	J1939 本机地址确认状态		清除	清除	读
M8303	J1939 通信错误输出		清除	清除	读
M8304	发生 J1939 通信总线关闭输出		清除	清除	读
M8305- M8310	— 保留 —		—	—	—
M8311	ESC+ 键输入 (上)	ESC+ 键输入 (⬆)	清除	清除	读
M8312	ESC+ 键输入 (下)	ESC+ 键输入 (⬇)	清除	清除	读
M8313	ESC+ 键输入 (左)	ESC+ 键输入 (⬅)	清除	清除	读
M8314	ESC+ 键输入 (右)	ESC+ 键输入 (➡)	清除	清除	读
M8315- M8319	— 保留 —		—	—	—
M8320	初始化增设扩展模块组合型主机 / 从机		执行	清除	读 / 写
M8321- M8330	— 保留 —		—	—	—
M8331	以太网端口 2 自动 Ping 执行中		执行	清除	读
M8332	以太网端口 2 自动 Ping 停止标记		执行	清除	读 / 写
M8333	CPU 模块 以太网端口 2 网络设置更改触发器		执行	清除	读 / 写
M8334	用户通信接收指令取消标记	连接 9	清除	清除	读 / 写
M8335		连接 10	清除	清除	读 / 写
M8336		连接 11	清除	清除	读 / 写
M8337		连接 12	清除	清除	读 / 写
M8340		连接 13	清除	清除	读 / 写
M8341		连接 14	清除	清除	读 / 写
M8342		连接 15	清除	清除	读 / 写
M8343		连接 16	清除	清除	读 / 写
M8344	以太网端口 1 初始化电子邮件发送服务器设置		执行	清除	读 / 写

设备地址	说明	停止时	电源关闭	读 / 写	
M8345	连接状态 (ON: 已接通, OFF: 未接通)	连接 9	执行	清除	读
M8346		连接 10	执行	清除	读
M8347		连接 11	执行	清除	读
M8350		连接 12	执行	清除	读
M8351		连接 13	执行	清除	读
M8352		连接 14	执行	清除	读
M8353		连接 15	执行	清除	读
M8354		连接 16	执行	清除	读
M8355	断开用户通信连接	连接 9	执行	清除	读 / 写
M8356		连接 10	执行	清除	读 / 写
M8357		连接 11	执行	清除	读 / 写
M8360		连接 12	执行	清除	读 / 写
M8361		连接 13	执行	清除	读 / 写
M8362		连接 14	执行	清除	读 / 写
M8363		连接 15	执行	清除	读 / 写
M8364		连接 16	执行	清除	读 / 写
M8365	用户通信接收指令取消标记 (端口 10)	清除	清除	读 / 写	
M8366	用户通信接收指令取消标记 (端口 11)	清除	清除	读 / 写	
M8367	用户通信接收指令取消标记 (端口 12)	清除	清除	读 / 写	
M8370	用户通信接收指令取消标记 (端口 13)	清除	清除	读 / 写	
M8371	用户通信接收指令取消标记 (端口 14)	清除	清除	读 / 写	
M8372	用户通信接收指令取消标记 (端口 15)	清除	清除	读 / 写	
M8373	用户通信接收指令取消标记 (端口 16)	清除	清除	读 / 写	
M8374	用户通信接收指令取消标记 (端口 17)	清除	清除	读 / 写	
M8375	用户通信接收指令取消标记 (端口 18)	清除	清除	读 / 写	
M8376	用户通信接收指令取消标记 (端口 19)	清除	清除	读 / 写	
M8377	用户通信接收指令取消标记 (端口 20)	清除	清除	读 / 写	
M8380	用户通信接收指令取消标记 (端口 21)	清除	清除	读 / 写	
M8381	用户通信接收指令取消标记 (端口 22)	清除	清除	读 / 写	
M8382	用户通信接收指令取消标记 (端口 23)	清除	清除	读 / 写	
M8383	用户通信接收指令取消标记 (端口 24)	清除	清除	读 / 写	
M8384	用户通信接收指令取消标记 (端口 25)	清除	清除	读 / 写	
M8385	用户通信接收指令取消标记 (端口 26)	清除	清除	读 / 写	
M8386	用户通信接收指令取消标记 (端口 27)	清除	清除	读 / 写	
M8387	用户通信接收指令取消标记 (端口 28)	清除	清除	读 / 写	
M8390	用户通信接收指令取消标记 (端口 29)	清除	清除	读 / 写	
M8391	用户通信接收指令取消标记 (端口 30)	清除	清除	读 / 写	
M8392	用户通信接收指令取消标记 (端口 31)	清除	清除	读 / 写	
M8393	用户通信接收指令取消标记 (端口 32)	清除	清除	读 / 写	
M8394	用户通信接收指令取消标记 (端口 33)	清除	清除	读 / 写	
M8395- M8447	— 保留 —	—	—	—	
M8450	BACnet 通信许可	执行	清除	读 / 写	
M8451- M8457	— 保留 —	—	—	—	
M8460	EtherNet/IP 通信许可	清除	清除	读 / 写	
M8461- M8597	— 保留 —	—	—	—	

## 2: 设备地址

设备地址	说明	停止时	电源关闭	读 / 写	
M8600	高速计数器 (组 3/I3)	复位状态	保持	清除	读
M8601		下溢出	保持	清除	读
M8602		计数方向标记	保持	清除	读
M8603- M8997	— 保留 —	—	—	—	



## 特殊内部继电器补充说明

### ■ M8000: 开始控制

控制 FC6A 型的运行 / 停止状态。打开 M8000 时进入运行状态，关闭则进入停止状态。请参见第 1-19 页上的“运行 / 停止用户程序”。但是，功能开关、停止输入及复位输入皆优先于开始控制。虽然 M8000 会在电源关闭时保持其状态，但当关闭时间超过备份时间而使保持数据丢失时，将按照“功能设置”的“在保持数据错误时的运行 / 停止指定”中设置的内容（运行指定 / 停止指定）进行动作。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“在保持数据错误时的运行 / 停止指定”。

### ■ M8001: 1 秒时钟复位

M8001 打开期间，M8121（1 秒时钟）将始终关闭。

### ■ M8002: 所有输出均关闭

M8002 打开期间，所有输出将关闭。以梯形图程序创建的自保持电路也将关闭。

### ■ M8003: 进位 (Cy)/ 借位 (Bw)

执行高级指令过程中产生进位 (Cy) 或借位 (Bw) 时，M8003 将打开。有关详情，请参见第 3-11 页上的“进位和借位”。

### ■ M8004: 用户程序执行错误

执行用户程序过程中发生错误时，M8004 将打开。有关用户程序执行错误的详情，请参见附录 -19 页上的“用户程序执行错误和梯形图程序执行错误”。

### ■ M8005: 通信错误

在数据连接通信时发生错误时，M8005 将打开。即使错误解除也将保持。

### ■ M8006: 通信禁止标记（数据连接主机时）

数据连接通信时、M8006 打开期间，将停止通信。

### ■ M8007: 初始化标记（数据连接主机时） / 通信停止标记（数据连接从机时）

数据连接主机时：在运行状态下将该标记从关闭设为打开时，为确认连接状态，数据连接将仅初始化 1 次。用于构成数据连接的从机电源接通时间晚于主机时。

数据连接从机时：来自主机的通信经过 10s 以上切断时，该标记将打开。如果可正常接收，则关闭。

### ■ M8010: 状态 LED 执行

M8010 打开期间，状态 LED [STAT] 亮起。关闭期间，状态 LED [STAT] 熄灭。

### ■ M8013: 日历 / 时钟数据写入 / 调整错误标记

时钟写入或时钟调整处理无法正常执行时，将打开。处理正常完成后将关闭。

### ■ M8014: 日历 / 时钟数据读取错误标记

从内部时钟向特殊数据寄存器（D8008 ~ D8021）的日历数据及时钟数据读取失败时，将打开。读取正常完成后将关闭。

### ■ M8016: 日历数据写入标记

在向日历数据（只写）的特殊数据寄存器（D8015 ~ D8018）中写入数据后，将 M8016 从关闭设为打开时，将向内置时钟写入日历数据（年、月、日、星期）。

### ■ M8017: 时钟数据写入标记

在向时钟数据（只写）的特殊数据寄存器（D8019 ~ D8021）中写入数据后，将 M8017 从关闭设为打开时，将向内置时钟写入时钟数据（时、分、秒）。

### ■ M8020: 日历 / 时钟数据写入标记

在向日历 / 时钟数据（只写）的特殊数据寄存器（D8015 ~ D8021）写入数据后，将 M8020 从关闭设为打开时，将向内置时钟写入日历数据（年、月、日、星期）及时钟数据（时、分、秒）。

### ■ M8021: 时钟数据调整标记

将 M8021 从关闭设为打开时，将对内置时钟的秒数进行调整。

- 秒数在 0 ~ 29 之间将 M8021 从关闭设为打开时，将秒数设为 0。

- 秒数在 30 ~ 59 之间将 M8021 从关闭设为打开时，分钟数将加 1 且将秒数设为 0。

### ■ M8022: 用户通信接收指令取消标记（端口 1）

M8022 打开期间，将取消正在端口 1 中执行的用户通信（接收指令）。

## 2: 设备地址

---

### ■ M8023: 用户通信接收指令取消标记 (端口 2)

M8023 打开期间, 将取消正在端口 2 中执行的用户通信 (接收指令)。

### ■ M8024: BMOV/WSFT 执行标记

执行 WSFT (字移位) 指令、BMOV (块传送) 指令过程中, M8024 将打开; 指令执行完成 (正常结束) 后将关闭。

### ■ M8025: 停止时维持输出

在 FC6A 型运行过程中打开 M8025, 停止时输出将保持运行时的状态。再次运行时, M8025 将自动关闭。

### ■ M8026: 用户通信接收指令取消标记 (端口 3)

M8026 打开期间, 将取消正在端口 3 中执行的用户通信 (接收指令)。

### ■ M8033、M8145 ~ M8147、M8170、M8176、M8365 ~ M8394: 用户通信接收指令取消标记 (端口 4 ~ 33)

这些标记打开期间, 将取消正在相应端口中执行的用户通信 (接收指令)。

M8033 = 用户通信接收指令取消标记 (端口 4)

M8145 = 用户通信接收指令取消标记 (端口 5)

M8146 = 用户通信接收指令取消标记 (端口 6)

M8147 = 用户通信接收指令取消标记 (端口 7)

M8170 = 用户通信接收指令取消标记 (端口 8)

M8176 = 用户通信接收指令取消标记 (端口 9)

M8365 = 用户通信接收指令取消标记 (端口 10)

M8366 = 用户通信接收指令取消标记 (端口 11)

M8367 = 用户通信接收指令取消标记 (端口 12)

M8370 = 用户通信接收指令取消标记 (端口 13)

M8371 = 用户通信接收指令取消标记 (端口 14)

M8372 = 用户通信接收指令取消标记 (端口 15)

M8373 = 用户通信接收指令取消标记 (端口 16)

M8374 = 用户通信接收指令取消标记 (端口 17)

M8375 = 用户通信接收指令取消标记 (端口 18)

M8376 = 用户通信接收指令取消标记 (端口 19)

M8377 = 用户通信接收指令取消标记 (端口 20)

M8380 = 用户通信接收指令取消标记 (端口 21)

M8381 = 用户通信接收指令取消标记 (端口 22)

M8382 = 用户通信接收指令取消标记 (端口 23)

M8383 = 用户通信接收指令取消标记 (端口 24)

M8384 = 用户通信接收指令取消标记 (端口 25)

M8385 = 用户通信接收指令取消标记 (端口 26)

M8386 = 用户通信接收指令取消标记 (端口 27)

M8387 = 用户通信接收指令取消标记 (端口 28)

M8390 = 用户通信接收指令取消标记 (端口 29)

M8391 = 用户通信接收指令取消标记 (端口 30)

M8392 = 用户通信接收指令取消标记 (端口 31)

M8393 = 用户通信接收指令取消标记 (端口 32)

M8394 = 用户通信接收指令取消标记 (端口 33)

### ■ M8027 ~ M8032、M8034 ~ M8036、M8040 ~ M8046、M8051 ~ M8055、M8057 ~ M8063、M8130、M8131、M8133 ~ M8136、M8161 ~ M8167、M8600 ~ M8602: 用于高速计数器的特殊内部继电器

用于高速计数器的特殊内部继电器。有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“高速计数器”。

M8027 ~ M8032、M8130、M8131、M8161、M8162 = 高速计数器 (组 1/I0)

M8034 ~ M8036、M8133、M8165、M8600 ~ M8602 = 高速计数器 (组 3/I3)

M8040 ~ M8042、M8134、M8166 = 高速计数器 (组 4/I4)

M8043 ~ M8046、M8135、M8136、M8163、M8164 = 高速计数器 (组 5/I6)

M8051 ~ M8055 = 高速计数器 (组 2/I1)

M8057 ~ M8063 = 高速计数器 (组 6/I7)

### ■ M8070: SD 记忆卡固定状态

在 FC6A 型中插入 SD 记忆卡，且 SD 记忆卡被识别并处于可使用状态时，将打开。未插入 SD 记忆卡或未被识别时，将关闭。

### ■ M8071: SD 记忆卡写入标记

访问 SD 记忆卡过程中，将打开。访问完成后将关闭。

### ■ M8072: SD 记忆卡固定解除

将 M8072 从关闭设为打开时，将停止访问 SD 记忆卡。为了将已停止访问的 SD 记忆卡设为可访问，请重新插入 SD 记忆卡。

### ■ M8073: 功能开关状态

表示位于 CPU 模块前面的功能开关的状态。

功能开关为 1 时，将打开。功能开关为 0 时，将关闭。

### ■ M8074: 电池电压测量标记

表示备份用电池的电池电压测量状态。

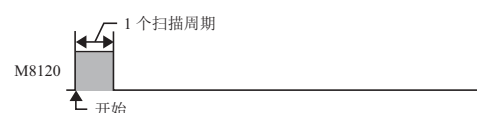
将 M8074 从关闭设为打开时，将开始测量电池电压；测量完成后将关闭。

### ■ M8080 ~ M8117: 数据连接通信完成继电器

用于数据连接通信的特殊内部继电器。有关详情，请参见《FC6A 型 MICROSmart 通信手册》第 7 章“数据连接通信”。

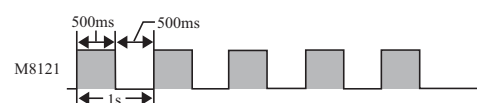
### ■ M8120: 初始化脉冲

仅打开运行（RUN）开始时的 1 次扫描。



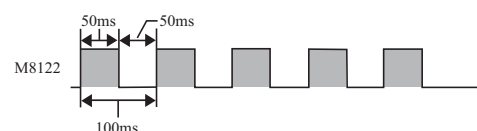
### ■ M8121: 1 秒时钟

M8001 关闭期间，M8121 将以 1s 为周期重复打开和关闭（占空比 1: 1）。



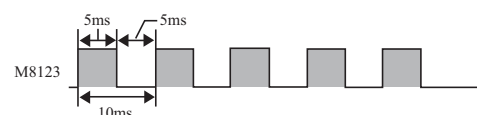
### ■ M8122: 100 毫秒时钟

M8122 以 100ms 为周期重复打开和关闭（占空比 1: 1）。



### ■ M8123: 10 毫秒时钟

M8123 以 10ms 为周期重复打开和关闭（占空比 1: 1）。



### ■ M8124: 定时器 / 计数器预置值更改

更改定时器 / 计数器的设置后，将打开。传送用户程序时或清除更改数据时，将关闭。

### ■ M8125: 运行中输出

处于运行状态期间，将打开。

### ■ M8126: 程序运行中下载完成后打开 1 次扫描

在运行状态下更改用户程序时（程序运行中下载）、下载完成后、开始运行更改后的用户程序时，将仅打开 1 次扫描。

## 2: 设备地址

---

### ■ M8137 ~ M8143、M8167: 中断输入状态

启用相应的用户中断时, 将打开。禁用用户中断时, 将关闭。

M8137 = 中断输入 I0 状态

M8140 = 中断输入 I1 状态

M8141 = 中断输入 I3 状态

M8142 = 中断输入 I4 状态

M8143 = 中断输入 I6 状态

M8167 = 中断输入 I7 状态

### ■ M8144: 定时器中断状态

启用定时器中断时, 将打开。禁用定时器中断时, 将关闭。

### ■ M8150 ~ M8152: 比较结果

M8150 ~ M8152 将根据 CMP= (比较等于) 指令、ICMP>= (间隔比较大于或等于) 指令的比较结果打开。有关详情, 请参见第 6-6 页上的“特殊内部继电器 M8150、M8151 和 M8152 用于 ICMP>=”

### ■ M8153 ~ M8160: 捕捉输入开/关状态

在 1 次扫描期间检测捕捉输入中指定的输入接点上升沿/下降沿输入时, 将读取输入接点的状态。可检测的边沿为 1 个扫描中 1 次。

M8153 = 组 1/I0 的状态

M8154 = 组 2/I1 的状态

M8155 = 组 3/I3 的状态

M8156 = 组 4/I4 的状态

M8157 = 组 5/I6 的状态

M8160 = 组 6/I7 的状态

### ■ M8172 ~ M8175: 晶体管源型输出过电流检测

在 CPU 模块的晶体管保护源型输出中产生过电流输出时, 将打开特殊内部继电器 (M8172 ~ M8175)。并将输出 4 点作为 1 个组, 分配下一个特殊内部继电器。在任意一个特殊内部继电器中产生过电流输出时, 将打开。

即使过电流输出解除, 这些特殊内部继电器也不会恢复为关闭。如需恢复关闭状态, 请通过梯形图程序编程使其关闭。

M8172 = 组 1 (Q0 ~ Q3) 的状态

M8173 = 组 2 (Q4 ~ Q7) 的状态

M8174 = 组 3 (Q10 ~ Q13) 的状态

M8175 = 组 4 (Q14 ~ Q17) 的状态

### ■ M8184: HMI 模块网络设置更改触发器

将 M8184 从关闭设为打开时, D8437 ~ D8456 中存储的值将设置为 HMI 模块的 IP 地址。

仅在 D8437 ~ D8456 的值有所更改时, 无法进行设置。有关 HMI 模块网络设置更改的详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“由 HMI 模块的特殊数据寄存器进行的网络设置”。

### ■ M8185: 在夏时制时间段

夏时制功能启用时、在夏时制时间段时将打开。在夏时制时间段之外时将关闭。

夏时制功能禁用时将关闭。

### ■ M8186: 以太网端口 1 自动 Ping 执行中

以太网端口 1 的自动 Ping 运行过程中将打开。自动 Ping 停止时将关闭。有关自动 Ping 的详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“自动 Ping 功能”。

### ■ M8187: 以太网端口 1 自动 Ping 停止标记

M8187 打开期间, 将停止以太网端口 1 的自动 Ping。M8187 关闭期间, 则执行自动 Ping。此时, 无论上次的结束状态如何, 自动 Ping 都将从远程主机列表中指定的远程主机编号的最小编号开始执行。

### ■ M8190: CPU 模块以太网端口 1 网络设置更改触发器

将 M8190 从关闭设为打开时, D8303 ~ D8323 中存储的值将设置为 CPU 模块以太网端口 1 的 IP 设置/DNS 设置。

仅在 D8303 ~ D8323 的值有所更改时, 无法进行设置。有关 CPU 模块以太网端口 1 的 IP 设置/DNS 设置更改的详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“由特殊数据寄存器进行的网络设置”。

### ■ M8191: SNTP 获取标记

将 M8191 从关闭设为打开时, 将从 SNTP 服务器获取时间信息。

### ■ M8192 ~ M8197: 中断输入边沿

在中断输入的上升沿发生中断时, 将打开。在中断输入的下降沿发生中断时, 将关闭。

M8192 = 中断输入 I0 边沿

M8193 = 中断输入 I3 边沿

M8194 = 中断输入 I4 边沿

M8195 = 中断输入 I6 边沿

M8196 = 中断输入 I7 边沿

M8197 = 中断输入 I1 边沿

### ■ M8200 ~ M8207、M8334 ~ M8343: 用户通信接收指令取消标记

将 M8200 ~ M8207 或 M8334 ~ M8343 从关闭设为打开时, 将中断正在执行的用户通信接收指令。

M8200 = 正在客户端连接 1 中执行的用户通信接收指令

M8201 = 正在客户端连接 2 中执行的用户通信接收指令

M8202 = 正在客户端连接 3 中执行的用户通信接收指令

M8203 = 正在客户端连接 4 中执行的用户通信接收指令

M8204 = 正在客户端连接 5 中执行的用户通信接收指令

M8205 = 正在客户端连接 6 中执行的用户通信接收指令

M8206 = 正在客户端连接 7 中执行的用户通信接收指令

M8207 = 正在客户端连接 8 中执行的用户通信接收指令

M8334 = 正在客户端连接 9 中执行的用户通信接收指令

M8335 = 正在客户端连接 10 中执行的用户通信接收指令

M8336 = 正在客户端连接 11 中执行的用户通信接收指令

M8337 = 正在客户端连接 12 中执行的用户通信接收指令

M8340 = 正在客户端连接 13 中执行的用户通信接收指令

M8341 = 正在客户端连接 14 中执行的用户通信接收指令

M8342 = 正在客户端连接 15 中执行的用户通信接收指令

M8343 = 正在客户端连接 16 中执行的用户通信接收指令

### ■ M8211: 初始化 HMI 模块电子邮件发送服务器设置

将 M8211 从关闭设为打开时, 将初始化 HMI- 以太网端口的电子邮件发送服务器设置。

### ■ M8212 ~ M8221、M8345 ~ M8354: 连接状态

通过维护通信服务器、用户通信服务器 / 客户端、Modbus TCP 服务器 / 客户端连接网络设备期间, 连接状态将打开。未连接期间, 将关闭。

M8212 = 连接 1

M8213 = 连接 2

M8214 = 连接 3

M8215 = 连接 4

M8216 = 连接 5

M8217 = 连接 6

M8220 = 连接 7

M8221 = 连接 8

M8345 = 连接 9

M8346 = 连接 10

M8347 = 连接 11

M8350 = 连接 12

M8351 = 连接 13

M8352 = 连接 14

M8353 = 连接 15

M8354 = 连接 16

## 2: 设备地址

---

### ■ M8222 ~ M8231、M8355 ~ M8364: 断开用户通信连接

通过用户通信进行连接时，如果将 M8222 ~ M8231 从关闭设为打开，将断开相应的连接。

M8222 = 连接 1  
M8223 = 连接 2  
M8224 = 连接 3  
M8225 = 连接 4  
M8226 = 连接 5  
M8227 = 连接 6  
M8230 = 连接 7  
M8231 = 连接 8  
M8355 = 连接 9  
M8356 = 连接 10  
M8357 = 连接 11  
M8360 = 连接 12  
M8361 = 连接 13  
M8362 = 连接 14  
M8363 = 连接 15  
M8364 = 连接 16

仅会在使用用户通信客户端期间启用。

### ■ M8232: HMI 模块连接信息参考 连接状态

在 D8429 中指定的连接编号将在有连接时打开。在无连接时，将关闭。

### ■ M8250: 从 SD 记忆卡下载执行标记

将 M8250 从关闭设为打开时，将从 SD 记忆卡中下载 ZLD 文件。要下载的文件为以 autoexec.ini 文件指定的 ZLD 文件。

### ■ M8251: 向 SD 记忆卡上传执行标记

将 M8251 从关闭设为打开时，将向 SD 记忆卡中上传 ZLD 文件。  
以通过 autoexec.ini 文件指定的文件名创建 ZLD 文件。

### ■ M8252: SD 记忆卡下载执行中

开始从 SD 记忆卡下载时将打开，完成后将关闭。

### ■ M8253: SD 记忆卡上传执行中

开始向 SD 记忆卡中上传时将打开，完成后将关闭。

### ■ M8254: SD 记忆卡下载 / 上传执行完成输出

开始从 SD 记忆卡下载或向 SD 记忆卡上传时 M8254 将关闭，下载 / 上传完成后则打开。

### ■ M8255: SD 记忆卡下载 / 上传执行错误输出

从 SD 记忆卡下载或向 SD 记忆卡上传完成时，将进行更新。D8255（下载、上传执行状态）为 0 以外数值时，将打开。

### ■ M8260: 写入配方执行标记

从关闭设为打开时，对 D8260（配方块编号）中指定的所有频道进行配方写入。

### ■ M8261: 读取配方执行标记

从关闭设为打开时，对 D8260（配方块编号）中指定的所有频道进行配方读出。

### ■ M8262: 写入配方执行中

在开始写入配方处理时将打开，完成后将关闭。在开始读取配方处理时也将关闭。

### ■ M8263: 读取配方执行中

在开始读取配方处理时将打开，完成后将关闭。在开始写入配方处理时也将关闭。

### ■ M8264: 配方频道执行完成输出

配方频道读写开始时关闭，配方频道读写完成时打开。

### ■ M8265: 执行配方错误输出

执行配方完成时，如果 D8264（配方执行状态）为 0 以外数值，将打开。有关配方的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

- **M8266: 配方块执行完成输出**  
配方块读写开始时关闭, 配方块读写完成时打开。
- **M8267: 配方内存 (ROM- 区域) 读取限制**  
向内存 (ROM- 区域) 完成配方块读取时打开。M8267 打开时, 无法将配方读取到内存 (ROM- 区域 1) 和内存 (ROM- 区域 2) 中。如需读取配方, 请关闭 M8267。
- **M8271: 从 SD 记忆卡进行 MQTT 基本设置下载的执行标记**  
将 M8271 从关闭设为打开时, 下载 MQTT 基本设置用文件。有关 MQTT 基本设置用文件的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。
- **M8272: 从 SD 记忆卡进行 MQTT 基本设置下载的执行完成输出**  
在开始下载 MQTT 基本设置用文件时关闭, 结束后打开。
- **M8273: 从 SD 记忆卡进行 MQTT 基本设置下载的错误输出**  
在开始下载 MQTT 基本设置用文件的处理时关闭。在处理过程中发生错误时打开。
- **M8300 ~ M8304: J1939 通信**  
用于 J1939 通信的特殊数据寄存器。有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 8 章“J1939 通信”。
- **M8311 ~ M8314: 键输入状态**  
同时按下 HMI 模块的 ESC 键和方向键期间, 将打开。未按下键时, 将关闭。  
M8311 = ESC 键+上 (⬆) 键  
M8312 = ESC 键+下 (⬇) 键  
M8313 = ESC 键+左 (⬅) 键  
M8314 = ESC 键+右 (➡) 键
- **M8320: 初始化增设扩展模块组合型主机 / 从机**  
将 M8320 设为打开时, 将初始化增设扩展模块组合型主机 / 从机及增设扩展模块组合型从机上所连接的扩展模块。初始化结束后, 将自动恢复关闭, 且增设扩展模块组合型从机重新开始 I/O 刷新。
- **M8331: 以太网端口 2 自动 Ping 执行中**  
以太网端口 2 的自动 Ping 运行过程中将打开。自动 Ping 停止时将关闭。有关自动 Ping 的详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“自动 Ping 功能”。
- **M8332: 以太网端口 2 自动 Ping 停止标记**  
M8332 打开期间, 将停止以太网端口 2 的自动 Ping。M8332 关闭期间, 则执行自动 Ping。此时, 无论上次的结束状态如何, 自动 Ping 都将从远程主机列表中指定的远程主机编号的最小编号开始执行。
- **M8333: CPU 模块 以太网端口 2 网络设置更改触发器**  
将 M8333 从关闭设为打开时, D8630 ~ D8650 中存储的值将设置为 CPU 模块以太网端口 2 的 IP 设置 /DNS 设置。  
仅在 D8630 ~ D8650 的值有所更改时, 无法进行设置。有关 CPU 模块以太网端口 2 的 IP 设置 /DNS 设置更改的详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“由特殊数据寄存器进行的网络设置”。
- **M8344: 以太网端口 1 初始化电子邮件发送服务器设置**  
将 M8344 从关闭设为打开时, 将初始化以太网端口 1 的电子邮件发送服务器设置。
- **M8450: BACnet 通信许可**  
M8450 打开期间, 进行 BACnet 通信。
- **M8460: EtherNet/IP 通信许可**  
启用 / 禁用 EtherNet/IP 通信。  
OFF: 禁用 EtherNet/IP 通信  
ON: 启用 EtherNet/IP 通信

## 2: 设备地址

### 特殊数据寄存器



**警告** 请勿改写特殊数据寄存器一览中写为保留的区域内的数据。否则系统可能无法正常工作。

**注释:** 读 / 写为读取 / 写入的简称。

读 / 写栏的表述如下所示。

读 / 写: 可读取及写入

读: 仅可读取

写: 仅可写入

设备地址	说明	刷新时间	读 / 写
D8000	输入点数	I/O 初始化时	读
D8001	输出点数	I/O 初始化时	读
D8002	CPU 模块类型信息	通电时	读
D8003 D8004	— 保留 —	—	—
D8005	通用错误代码	发生错误时	读 / 写
D8006	用户程序执行错误代码	发生错误时	读
D8007	— 保留 —	—	—
D8008	日历 / 时钟 当前值 (只读)	年	每 500ms 读
D8009		月	每 500ms 读
D8010		日	每 500ms 读
D8011		星期	每 500ms 读
D8012		小时	每 500ms 读
D8013		分钟	每 500ms 读
D8014		秒	每 500ms 读
D8015	日历 / 时钟 预置值 (只写)	年	— 写
D8016		月	— 写
D8017		日	— 写
D8018		星期	— 写
D8019		小时	— 写
D8020		分钟	— 写
D8021		秒	— 写
D8022	扫描时间数据	固定扫描时间预置值	— 写
D8023		扫描时间 (当前值)	每次扫描 读
D8024		扫描时间 (最大值)	出现时 读
D8025		扫描时间 (最小值)	出现时 读
D8026	通信模式信息 (端口 1 ~ 3)	每次扫描	读
D8027 D8028	— 保留 —	—	—
D8029	系统软件版本	通电时	读
D8030	通信盒信息	通电时	读
D8031	选项连接信息	通电时	读
D8032	中断输入跳转目标标签编号 (I1)	—	读 / 写
D8033	中断输入跳转目标标签编号 (I3)	—	读 / 写
D8034	中断输入跳转目标标签编号 (I4)	—	读 / 写
D8035	中断输入跳转目标标签编号 (I6)	—	读 / 写
D8036	定时器中断跳转目标标签编号	—	读 / 写
D8037	输入输出模块连接台数	I/O 初始化时	读
D8038 D8039	— 保留 —	—	—
D8040	从机编号 (端口 4)	—	读 / 写



设备地址	说明	刷新时间	读/写
D8041	从机编号 (端口 5)	—	读/写
D8042	从机编号 (端口 6)	—	读/写
D8043	从机编号 (端口 7)	—	读/写
D8044	从机编号 (端口 8)	—	读/写
D8045	从机编号 (端口 9)	—	读/写
D8046- D8051	— 保留 —	—	—
D8052	J1939 通信错误代码	每次扫描	读/写
D8053- D8055	— 保留 —	—	—
D8056	电池电压	—	读
D8057	模拟量 (AI0)	每次扫描	读
D8058	内置模拟量输入 (AI1)	每次扫描	读
D8059	模拟量输入状态 AI0	每次扫描	读
D8060	模拟量输入状态 AI1	每次扫描	读
D8061- D8066	— 保留 —	—	—
D8067	背光亮起时间	—	读/写
D8068	— 保留 —	—	—
D8069	从机 1 通信状态 / 错误 (数据连接主机模式时) 从机通信状态 / 错误 (数据连接从机模式时)	发生错误时	读
D8070	从机 2 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8071	从机 3 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8072	从机 4 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8073	从机 5 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8074	从机 6 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8075	从机 7 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8076	从机 8 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8077	从机 9 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8078	从机 10 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8079	从机 11 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8080	从机 12 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8081	从机 13 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8082	从机 14 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8083	从机 15 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8084	从机 16 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8085	从机 17 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8086	从机 18 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8087	从机 19 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8088	从机 20 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8089	从机 21 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8090	从机 22 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8091	从机 23 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8092	从机 24 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8093	从机 25 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8094	从机 26 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8095	从机 27 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8096	从机 28 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8097	从机 29 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读
D8098	从机 30 通信状态 / 错误 (数据连接主机模式时)	发生错误时	读

## 2: 设备地址

设备地址	说明		刷新时间	读 / 写	
D8099	从机 31 通信状态 / 错误 (数据连接主机模式时)		发生错误时	读	
D8100	从机编号 (端口 1)		—	读 / 写	
D8101	— 保留 —		—	—	
D8102	从机编号 (端口 2)		—	读 / 写	
D8103	从机编号 (端口 3)		—	读 / 写	
D8104	控制信号状态 (端口 1 ~ 5)		每次扫描	读	
D8105	RS232C DSR 控制信号状态 (端口 1 ~ 5)		每次扫描	读 / 写	
D8106	RS232C DTR 输出控制信号选项 (端口 1 ~ 5)		每次扫描	读 / 写	
D8107- D8119	— 保留 —		—	—	
D8120	HMI 模块信息	类型 ID/ 状态	—	读	
D8121		系统软件版本	—	读	
D8122	盒插槽 1 信息	类型 ID/ 状态	—	读	
D8123		系统软件版本	—	读	
D8124	盒插槽 2 信息	类型 ID/ 状态	—	读	
D8125		系统软件版本	—	读	
D8126	盒插槽 3 信息	类型 ID/ 状态	—	读	
D8127		系统软件版本	—	读	
D8128- D8169	— 保留 —		—	—	
D8170	模拟 I/O 盒输入输出 AI2/AQ2		每次扫描	读	
D8171	模拟 I/O 盒输入输出 AI3/AQ3		每次扫描	读	
D8172	模拟 I/O 盒状态 AI2/AQ2		每次扫描	读	
D8173	模拟 I/O 盒状态 AI3/AQ3		每次扫描	读	
D8174	模拟 I/O 盒输入输出 AI4/AQ4		每次扫描	读	
D8175	模拟 I/O 盒输入输出 AI5/AQ5		每次扫描	读	
D8176	模拟 I/O 盒状态 AI4/AQ4		每次扫描	读	
D8177	模拟 I/O 盒状态 AI5/AQ5		每次扫描	读	
D8178	模拟 I/O 盒输入输出 AI6/AQ6		每次扫描	读	
D8179	模拟 I/O 盒输入输出 AI7/AQ7		每次扫描	读	
D8180	模拟 I/O 盒状态 AI6/AQ6		每次扫描	读	
D8181	模拟 I/O 盒状态 AI7/AQ7		每次扫描	读	
D8182- D8191	— 保留 —		—	—	
D8192	高速计数器 (组 2/I1)	高位字	当前值 / 频率测量 (I1) 当前值	每次扫描	读
D8193		低位字		每次扫描	读
D8194		高位字	预置值	—	读 / 写
D8195		低位字		—	读 / 写
D8196		高位字	复位值	—	读 / 写
D8197		低位字		—	读 / 写
D8198	高速计数器 (组 6/I7)	高位字	当前值 / 频率测量 (I7) 当前值	每次扫描	读
D8199		低位字		每次扫描	读
D8200		高位字	预置值	—	读 / 写
D8201		低位字		—	读 / 写
D8202		高位字	复位值	—	读 / 写
D8203		低位字		—	读 / 写
D8204	控制信号状态 (端口 6 ~ 9)		每次扫描	读	
D8205	RS232C DSR 控制信号状态 (端口 6 ~ 9)		每次扫描	读 / 写	
D8206	RS232C DTR 输出控制信号选项 (端口 6 ~ 9)		每次扫描	读 / 写	

设备地址	说明		刷新时间	读 / 写	
D8207- D8209	— 保留 —		—	—	
D8210	高速计数器 (组 1/I0)	高位字	当前值 / 频率测量 (I0) 当前值	每次扫描	读
D8211		低位字		每次扫描	读
D8212		高位字	预置值	—	读 / 写
D8213		低位字		—	读 / 写
D8214	中断输入跳转目标标签编号 (I7)		—	读 / 写	
D8215	中断输入跳转目标标签编号 (I0)		—	读 / 写	
D8216	高速计数器 (组 1/I0)	高位字	复位值	—	读 / 写
D8217		低位字		—	读 / 写
D8218	高速计数器 (组 3/I3)	高位字	当前值 / 频率测量 (I3) 当前值	每次扫描	读
D8219		低位字		每次扫描	读
D8220		高位字	预置值	—	读 / 写
D8221		低位字		—	读 / 写
D8222	高速计数器 (组 4/I4)	高位字	当前值 / 频率测量 (I4) 当前值	每次扫描	读
D8223		低位字		每次扫描	读
D8224		高位字	预置值	—	读 / 写
D8225		低位字		—	读 / 写
D8226	高速计数器 (组 5/I6)	高位字	当前值 / 频率测量 (I6) 当前值	每次扫描	读
D8227		低位字		每次扫描	读
D8228		高位字	预置值	—	读 / 写
D8229		低位字		—	读 / 写
D8230 D8231	— 保留 —		—	—	
D8232	高速计数器 (组 5/I6)	高位字	复位值	—	读 / 写
D8233		低位字		—	读 / 写
D8234	高速计数器 (组 3/I3)	高位字	复位值	—	读 / 写
D8235		低位字		—	读 / 写
D8236	高速计数器 (组 4/I4)	高位字	复位值	—	读 / 写
D8237		低位字		—	读 / 写
D8238	— 保留 —		—	—	
D8239	绝对位置管理状态		每次扫描	读	
D8240	绝对位置计数 器 1	高位字	绝对位置	每次扫描	读
D8241		低位字		每次扫描	读
D8242	绝对位置计数 器 2	高位字	绝对位置	每次扫描	读
D8243		低位字		每次扫描	读
D8244	绝对位置计数 器 3	高位字	绝对位置	每次扫描	读
D8245		低位字		每次扫描	读
D8246	绝对位置计数 器 4	高位字	绝对位置	每次扫描	读
D8247		低位字		每次扫描	读
D8248 D8249	— 保留 —		—	—	
D8250	SD 记忆卡容量显示		每次扫描	读	
D8251	SD 记忆卡可用空间显示		每次扫描	读	
D8252 D8253	— 保留 —		—	—	
D8254	SD 记忆卡 下载 / 上传执行信息		处理完成时	读	
D8255	SD 记忆卡 下载 / 上传执行状态		处理完成时	读	
D8256	从 SD 记忆卡进行 MQTT 基本设置下载的执行信息		处理完成时	读	
D8257	从 SD 记忆卡进行 MQTT 基本设置下载的执行错误信息		处理完成时	读	

## 2: 设备地址

设备地址	说明		刷新时间	读 / 写
D8258 D8259	— 保留 —		—	—
D8260	配方块编号		—	读 / 写
D8261	执行配方块编号		执行配方完成时	读
D8262	执行配方频道编号		执行配方完成时	读
D8263	执行配方动作		执行配方完成时	读
D8264	执行配方状态		执行配方完成时	读
D8265	执行配方错误信息		执行配方完成时	读
D8266	配方内存 (ROM- 区域 1) 读取次数		执行配方完成时	读
D8267	配方内存 (ROM- 区域 2) 读取次数		执行配方完成时	读
D8268	连接 1 的远程主机号 (1 ~ 255)		—	读 / 写
D8269	连接 2 的远程主机号 (1 ~ 255)		—	读 / 写
D8270	连接 3 的远程主机号 (1 ~ 255)		—	读 / 写
D8271	连接 4 的远程主机号 (1 ~ 255)		—	读 / 写
D8272	连接 5 的远程主机号 (1 ~ 255)		—	读 / 写
D8273	连接 6 的远程主机号 (1 ~ 255)		—	读 / 写
D8274	连接 7 的远程主机号 (1 ~ 255)		—	读 / 写
D8275	连接 8 的远程主机号 (1 ~ 255)		—	读 / 写
D8276 D8277	— 保留 —		—	—
D8278	通信模式信息	连接 1 ~ 4	—	读
D8279	(客户端连接)	连接 5 ~ 8	—	读
D8280- D8283	— 保留 —		—	—
D8284	通信模式信息	HMI 连接 1 ~ 4	—	读
D8285	(HMI 连接)	HMI 连接 5 ~ 8	—	读
D8286- D8302	— 保留 —		—	—
D8303	CPU 模块以太网端口 1 IP 设置 / DNS 设置切换		—	读 / 写
D8304	CPU 模块以太网端口 1 IP 地址 (只写)		—	写
D8305			—	写
D8306			—	写
D8307			—	写
D8308			—	写
D8309	CPU 模块以太网端口 1 子网掩码 (只写)		—	写
D8310			—	写
D8311			—	写
D8312			—	写
D8313	CPU 模块以太网端口 1 默认网关 (只写)		—	写
D8314			—	写
D8315			—	写
D8316			—	写
D8317	CPU 模块以太网端口 1 首选 DNS 服务器 (只写)		—	写
D8318			—	写
D8319			—	写
D8320			—	写
D8321	CPU 模块以太网端口 1 备用 DNS 服务器 (只写)		—	写
D8322			—	写
D8323			—	写
D8323			—	写

设备地址	说明	刷新时间	读/写
D8324	CPU 模块以太网端口 1 MAC 地址 (只读当前值)	每 1 秒	读
D8325		每 1 秒	读
D8326		每 1 秒	读
D8327		每 1 秒	读
D8328		每 1 秒	读
D8329		每 1 秒	读
D8330	CPU 模块以太网端口 1 IP 地址 (只读当前值)	每 1 秒	读
D8331		每 1 秒	读
D8332		每 1 秒	读
D8333		每 1 秒	读
D8334	CPU 模块以太网端口 1 子网掩码 (只读当前值)	每 1 秒	读
D8335		每 1 秒	读
D8336		每 1 秒	读
D8337		每 1 秒	读
D8338	CPU 模块以太网端口 1 默认网关 (只读当前值)	每 1 秒	读
D8339		每 1 秒	读
D8340		每 1 秒	读
D8341		每 1 秒	读
D8342	CPU 模块以太网端口 1 首选 DNS 服务器 (只读当前值)	每 1 秒	读
D8343		每 1 秒	读
D8344		每 1 秒	读
D8345		每 1 秒	读
D8346	CPU 模块以太网端口 1 备用 DNS 服务器 (只读当前值)	每 1 秒	读
D8347		每 1 秒	读
D8348		每 1 秒	读
D8349		每 1 秒	读
D8350	连接 1 连接的 IP 地址	每 1 秒	读
D8351		每 1 秒	读
D8352		每 1 秒	读
D8353		每 1 秒	读
D8354	连接 2 连接的 IP 地址	每 1 秒	读
D8355		每 1 秒	读
D8356		每 1 秒	读
D8357		每 1 秒	读
D8358	连接 3 连接的 IP 地址	每 1 秒	读
D8359		每 1 秒	读
D8360		每 1 秒	读
D8361		每 1 秒	读
D8362	连接 4 连接的 IP 地址	每 1 秒	读
D8363		每 1 秒	读
D8364		每 1 秒	读
D8365		每 1 秒	读
D8366	连接 5 连接的 IP 地址	每 1 秒	读
D8367		每 1 秒	读
D8368		每 1 秒	读
D8369		每 1 秒	读
D8370	连接 6 连接的 IP 地址	每 1 秒	读
D8371		每 1 秒	读
D8372		每 1 秒	读
D8373		每 1 秒	读

## 2: 设备地址

设备地址	说明	刷新时间	读 / 写	
D8374	连接 7 连接的 IP 地址	每 1 秒	读	
D8375		每 1 秒	读	
D8376		每 1 秒	读	
D8377		每 1 秒	读	
D8378	连接 8 连接的 IP 地址	每 1 秒	读	
D8379		每 1 秒	读	
D8380		每 1 秒	读	
D8381		每 1 秒	读	
D8382	HMI 模块 MAC 地址 (只读当前值)	每 1 秒	读	
D8383		每 1 秒	读	
D8384		每 1 秒	读	
D8385		每 1 秒	读	
D8386		每 1 秒	读	
D8387		每 1 秒	读	
D8388	HMI 模块 IP 地址 (只读当前值)	每 1 秒	读	
D8389		每 1 秒	读	
D8390		每 1 秒	读	
D8391		每 1 秒	读	
D8392	HMI 模块 子网掩码 (只读当前值)	每 1 秒	读	
D8393		每 1 秒	读	
D8394		每 1 秒	读	
D8395		每 1 秒	读	
D8396	HMI 模块 默认网关 (只读当前值)	每 1 秒	读	
D8397		每 1 秒	读	
D8398		每 1 秒	读	
D8399		每 1 秒	读	
D8400	HMI 模块 首选 DNS 服务器 (只读当前值)	每 1 秒	读	
D8401		每 1 秒	读	
D8402		每 1 秒	读	
D8403		每 1 秒	读	
D8404	HMI 模块 备用 DNS 服务器 (只读当前值)	每 1 秒	读	
D8405		每 1 秒	读	
D8406		每 1 秒	读	
D8407		每 1 秒	读	
D8408- D8412	— 保留 —	—	—	
D8413	时区偏移量	—	读 / 写	
D8414	SNTP 运行状态	—	读	
D8415	SNTP 访问经过时间	—	读	
D8416- D8428	— 保留 —	—	—	
D8429	HMI 模块 连接信息参考	连接编号	—	
D8430		连接 IP 地址	每 1 秒	读 / 写
D8431			每 1 秒	读
D8432			每 1 秒	读
D8433			每 1 秒	读
D8434	连接端口编号	每 1 秒	读	
D8435 D8436	— 保留 —	—	—	

设备地址	说明		刷新时间	读 / 写
D8437	HMI 模块 IP 地址 (只写)		—	写
D8438			—	写
D8439			—	写
D8440			—	写
D8441	HMI 模块子网掩码 (只写)		—	写
D8442			—	写
D8443			—	写
D8444			—	写
D8445	HMI 模块默认网关 (只写)		—	写
D8446			—	写
D8447			—	写
D8448			—	写
D8449	HMI 模块首选 DNS 服务器 (只写)		—	写
D8450			—	写
D8451			—	写
D8452			—	写
D8453	HMI 模块备用 DNS 服务器 (只写)		—	写
D8454			—	写
D8455			—	写
D8456			—	写
D8457	EMAIL 指令详细错误信息 (HMI- 以太网端口)		—	读
D8458- D8469	— 保留 —		—	—
D8470	扩展模块槽 1 信息	类型 ID/ 状态	—	读
D8471		系统软件版本 / 位置信息	—	读
D8472	扩展模块槽 2 信息	类型 ID/ 状态	—	读
D8473		系统软件版本 / 位置信息	—	读
D8474	扩展模块槽 3 信息	类型 ID/ 状态	—	读
D8475		系统软件版本 / 位置信息	—	读
D8476	扩展模块槽 4 信息	类型 ID/ 状态	—	读
D8477		系统软件版本 / 位置信息	—	读
D8478	扩展模块槽 5 信息	类型 ID/ 状态	—	读
D8479		系统软件版本 / 位置信息	—	读
D8480	扩展模块槽 6 信息	类型 ID/ 状态	—	读
D8481		系统软件版本 / 位置信息	—	读
D8482	扩展模块槽 7 信息	类型 ID/ 状态	—	读
D8483		系统软件版本 / 位置信息	—	读
D8484	扩展模块槽 8 信息	类型 ID/ 状态	—	读
D8485		系统软件版本 / 位置信息	—	读
D8486	扩展模块槽 9 信息	类型 ID/ 状态	—	读
D8487		系统软件版本 / 位置信息	—	读
D8488	扩展模块槽 10 信息	类型 ID/ 状态	—	读
D8489		系统软件版本 / 位置信息	—	读
D8490	扩展模块槽 11 信息	类型 ID/ 状态	—	读
D8491		系统软件版本 / 位置信息	—	读
D8492	扩展模块槽 12 信息	类型 ID/ 状态	—	读
D8493		系统软件版本 / 位置信息	—	读
D8494	扩展模块槽 13 信息	类型 ID/ 状态	—	读
D8495		系统软件版本 / 位置信息	—	读

## 2: 设备地址

设备地址	说明	刷新时间	读 / 写	
D8496	扩展模块 槽 14 信息	类型 ID/ 状态	—	读
D8497		系统软件版本 / 位置信息	—	读
D8498	扩展模块 槽 15 信息	类型 ID/ 状态	—	读
D8499		系统软件版本 / 位置信息	—	读
D8500	扩展模块 槽 16 信息	类型 ID/ 状态	—	读
D8501		系统软件版本 / 位置信息	—	读
D8502	扩展模块 槽 17 信息	类型 ID/ 状态	—	读
D8503		系统软件版本 / 位置信息	—	读
D8504	扩展模块 槽 18 信息	类型 ID/ 状态	—	读
D8505		系统软件版本 / 位置信息	—	读
D8506	扩展模块 槽 19 信息	类型 ID/ 状态	—	读
D8507		系统软件版本 / 位置信息	—	读
D8508	扩展模块 槽 20 信息	类型 ID/ 状态	—	读
D8509		系统软件版本 / 位置信息	—	读
D8510	扩展模块 槽 21 信息	类型 ID/ 状态	—	读
D8511		系统软件版本 / 位置信息	—	读
D8512	扩展模块 槽 22 信息	类型 ID/ 状态	—	读
D8513		系统软件版本 / 位置信息	—	读
D8514	扩展模块 槽 23 信息	类型 ID/ 状态	—	读
D8515		系统软件版本 / 位置信息	—	读
D8516	扩展模块 槽 24 信息	类型 ID/ 状态	—	读
D8517		系统软件版本 / 位置信息	—	读
D8518	扩展模块 槽 25 信息	类型 ID/ 状态	—	读
D8519		系统软件版本 / 位置信息	—	读
D8520	扩展模块 槽 26 信息	类型 ID/ 状态	—	读
D8521		系统软件版本 / 位置信息	—	读
D8522	扩展模块 槽 27 信息	类型 ID/ 状态	—	读
D8523		系统软件版本 / 位置信息	—	读
D8524	扩展模块 槽 28 信息	类型 ID/ 状态	—	读
D8525		系统软件版本 / 位置信息	—	读
D8526	扩展模块 槽 29 信息	类型 ID/ 状态	—	读
D8527		系统软件版本 / 位置信息	—	读
D8528	扩展模块 槽 30 信息	类型 ID/ 状态	—	读
D8529		系统软件版本 / 位置信息	—	读
D8530	扩展模块 槽 31 信息	类型 ID/ 状态	—	读
D8531		系统软件版本 / 位置信息	—	读
D8532	扩展模块 槽 32 信息	类型 ID/ 状态	—	读
D8533		系统软件版本 / 位置信息	—	读
D8534	扩展模块 槽 33 信息	类型 ID/ 状态	—	读
D8535		系统软件版本 / 位置信息	—	读
D8536	扩展模块 槽 34 信息	类型 ID/ 状态	—	读
D8537		系统软件版本 / 位置信息	—	读
D8538	扩展模块 槽 35 信息	类型 ID/ 状态	—	读
D8539		系统软件版本 / 位置信息	—	读
D8530	扩展模块 槽 36 信息	类型 ID/ 状态	—	读
D8541		系统软件版本 / 位置信息	—	读
D8542	扩展模块 槽 37 信息	类型 ID/ 状态	—	读
D8543		系统软件版本 / 位置信息	—	读
D8544	扩展模块 槽 38 信息	类型 ID/ 状态	—	读
D8545		系统软件版本 / 位置信息	—	读



设备地址	说明	刷新时间	读 / 写	
D8546	扩展模块 槽 39 信息	类型 ID/ 状态	—	读
D8547		系统软件版本 / 位置信息	—	读
D8548	扩展模块 槽 40 信息	类型 ID/ 状态	—	读
D8549		系统软件版本 / 位置信息	—	读
D8550	扩展模块 槽 41 信息	类型 ID/ 状态	—	读
D8551		系统软件版本 / 位置信息	—	读
D8552	扩展模块 槽 42 信息	类型 ID/ 状态	—	读
D8553		系统软件版本 / 位置信息	—	读
D8554	扩展模块 槽 43 信息	类型 ID/ 状态	—	读
D8555		系统软件版本 / 位置信息	—	读
D8556	扩展模块 槽 44 信息	类型 ID/ 状态	—	读
D8557		系统软件版本 / 位置信息	—	读
D8558	扩展模块 槽 45 信息	类型 ID/ 状态	—	读
D8559		系统软件版本 / 位置信息	—	读
D8560	扩展模块 槽 46 信息	类型 ID/ 状态	—	读
D8561		系统软件版本 / 位置信息	—	读
D8562	扩展模块 槽 47 信息	类型 ID/ 状态	—	读
D8563		系统软件版本 / 位置信息	—	读
D8564	扩展模块 槽 48 信息	类型 ID/ 状态	—	读
D8565		系统软件版本 / 位置信息	—	读
D8566	扩展模块 槽 49 信息	类型 ID/ 状态	—	读
D8567		系统软件版本 / 位置信息	—	读
D8568	扩展模块 槽 50 信息	类型 ID/ 状态	—	读
D8569		系统软件版本 / 位置信息	—	读
D8570	扩展模块 槽 51 信息	类型 ID/ 状态	—	读
D8571		系统软件版本 / 位置信息	—	读
D8572	扩展模块 槽 52 信息	类型 ID/ 状态	—	读
D8573		系统软件版本 / 位置信息	—	读
D8574	扩展模块 槽 53 信息	类型 ID/ 状态	—	读
D8575		系统软件版本 / 位置信息	—	读
D8576	扩展模块 槽 54 信息	类型 ID/ 状态	—	读
D8577		系统软件版本 / 位置信息	—	读
D8578	扩展模块 槽 55 信息	类型 ID/ 状态	—	读
D8579		系统软件版本 / 位置信息	—	读
D8580	扩展模块 槽 56 信息	类型 ID/ 状态	—	读
D8581		系统软件版本 / 位置信息	—	读
D8582	扩展模块 槽 57 信息	类型 ID/ 状态	—	读
D8583		系统软件版本 / 位置信息	—	读
D8584	扩展模块 槽 58 信息	类型 ID/ 状态	—	读
D8585		系统软件版本 / 位置信息	—	读
D8586	扩展模块 槽 59 信息	类型 ID/ 状态	—	读
D8587		系统软件版本 / 位置信息	—	读
D8588	扩展模块 槽 60 信息	类型 ID/ 状态	—	读
D8589		系统软件版本 / 位置信息	—	读
D8590	扩展模块 槽 61 信息	类型 ID/ 状态	—	读
D8591		系统软件版本 / 位置信息	—	读
D8592	扩展模块 槽 62 信息	类型 ID/ 状态	—	读
D8593		系统软件版本 / 位置信息	—	读
D8594	扩展模块 槽 63 信息	类型 ID/ 状态	—	读
D8595		系统软件版本 / 位置信息	—	读

## 2: 设备地址

设备地址	说明	刷新时间	读 / 写	
D8596	增设扩展模块组合型主机槽信息	类型 ID/ 状态	—	读
D8597		系统软件版本 / 增设扩展模块组合型从机的连接台数	—	读
D8598	增设扩展模块组合型从机 (节点 1) 槽信息	类型 ID/ 状态	—	读
D8599		系统软件版本 / 扩展模块的连接信息	—	读
D8600	增设扩展模块组合型从机 (节点 2) 槽信息	类型 ID/ 状态	—	读
D8601		系统软件版本 / 扩展模块的连接信息	—	读
D8602	增设扩展模块组合型从机 (节点 3) 槽信息	类型 ID/ 状态	—	读
D8603		系统软件版本 / 扩展模块的连接信息	—	读
D8604	增设扩展模块组合型从机 (节点 4) 槽信息	类型 ID/ 状态	—	读
D8605		系统软件版本 / 扩展模块的连接信息	—	读
D8606	增设扩展模块组合型从机 (节点 5) 槽信息	类型 ID/ 状态	—	读
D8607		系统软件版本 / 扩展模块的连接信息	—	读
D8608	增设扩展模块组合型从机 (节点 6) 槽信息	类型 ID/ 状态	—	读
D8609		系统软件版本 / 扩展模块的连接信息	—	读
D8610	增设扩展模块组合型从机 (节点 7) 槽信息	类型 ID/ 状态	—	读
D8611		系统软件版本 / 扩展模块的连接信息	—	读
D8612	增设扩展模块组合型从机 (节点 8) 槽信息	类型 ID/ 状态	—	读
D8613		系统软件版本 / 扩展模块的连接信息	—	读
D8614	增设扩展模块组合型从机 (节点 9) 槽信息	类型 ID/ 状态	—	读
D8615		系统软件版本 / 扩展模块的连接信息	—	读
D8616	增设扩展模块组合型从机 (节点 10) 槽信息	类型 ID/ 状态	—	读
D8617		系统软件版本 / 扩展模块的连接信息	—	读
D8618	增设扩展模块组合型 I/O 刷新时间 当前值	每次扫描	读	
D8619	增设扩展模块组合型 I/O 刷新时间 最大值	每次扫描	读	
D8620- D8629	— 保留 —	—	—	
D8630	CPU 模块以太网端口 2 IP 设置 / DNS 设置切换	—	读 / 写	
D8631	CPU 模块 以太网端口 2 IP 地址 (只写)	—	写	
D8632		—	写	
D8633		—	写	
D8634		—	写	
D8635	CPU 模块 以太网端口 2 子网掩码 (只写)	—	写	
D8636		—	写	
D8637		—	写	
D8638		—	写	
D8639	CPU 模块 以太网端口 2 默认网关 (只写)	—	写	
D8640		—	写	
D8641		—	写	
D8642		—	写	
D8643	CPU 模块 以太网端口 2 首选 DNS 服务器 (只写)	—	写	
D8644		—	写	
D8645		—	写	
D8646		—	写	
D8647	CPU 模块 以太网端口 2 备用 DNS 服务器 (只写)	—	写	
D8648		—	写	
D8649		—	写	
D8650		—	写	

设备地址	说明	刷新时间	读 / 写
D8651	CPU 模块 以太网端口 2 MAC 地址 (只读当前值)	每 1 秒	读
D8652		每 1 秒	读
D8653		每 1 秒	读
D8654		每 1 秒	读
D8655		每 1 秒	读
D8656		每 1 秒	读
D8657	CPU 模块 以太网端口 2 IP 地址 (只读当前值)	每 1 秒	读
D8658		每 1 秒	读
D8659		每 1 秒	读
D8660		每 1 秒	读
D8661	CPU 模块 以太网端口 2 子网掩码 (只读当前值)	每 1 秒	读
D8662		每 1 秒	读
D8663		每 1 秒	读
D8664		每 1 秒	读
D8665	CPU 模块 以太网端口 2 默认网关 (只读当前值)	每 1 秒	读
D8666		每 1 秒	读
D8667		每 1 秒	读
D8668		每 1 秒	读
D8669	CPU 模块 以太网端口 2 首选 DNS 服务器 (只读当前值)	每 1 秒	读
D8670		每 1 秒	读
D8671		每 1 秒	读
D8672		每 1 秒	读
D8673	CPU 模块 以太网端口 2 备用 DNS 服务器 (只读当前值)	每 1 秒	读
D8674		每 1 秒	读
D8675		每 1 秒	读
D8676		每 1 秒	读
D8677	连接 9 连接的 IP 地址	每 1 秒	读
D8678		每 1 秒	读
D8679		每 1 秒	读
D8680		每 1 秒	读
D8681	连接 10 连接的 IP 地址	每 1 秒	读
D8682		每 1 秒	读
D8683		每 1 秒	读
D8684		每 1 秒	读
D8685	连接 11 连接的 IP 地址	每 1 秒	读
D8686		每 1 秒	读
D8687		每 1 秒	读
D8688		每 1 秒	读
D8689	连接 12 连接的 IP 地址	每 1 秒	读
D8690		每 1 秒	读
D8691		每 1 秒	读
D8692		每 1 秒	读
D8693	连接 13 连接的 IP 地址	每 1 秒	读
D8694		每 1 秒	读
D8695		每 1 秒	读
D8696		每 1 秒	读
D8697	连接 14 连接的 IP 地址	每 1 秒	读
D8698		每 1 秒	读
D8699		每 1 秒	读
D8700		每 1 秒	读

## 2: 设备地址

设备地址	说明	刷新时间	读 / 写
D8701	连接 15 连接的 IP 地址	每 1 秒	读
D8702		每 1 秒	读
D8703		每 1 秒	读
D8704		每 1 秒	读
D8705	连接 16 连接的 IP 地址	每 1 秒	读
D8706		每 1 秒	读
D8707		每 1 秒	读
D8708		每 1 秒	读
D8709- D8716	— 保留 —	—	—
D8717	控制信号状态 (端口 10 ~ 13)	每次扫描	读
D8718	RS232C DSR 控制信号状态 (端口 10 ~ 13)	每次扫描	读 / 写
D8719	RS232C DTR 输出控制信号选项 (端口 10 ~ 13)	每次扫描	读 / 写
D8720	控制信号状态 (端口 14 ~ 17)	每次扫描	读
D8721	RS232C DSR 控制信号状态 (端口 14 ~ 17)	每次扫描	读 / 写
D8722	RS232C DTR 输出控制信号选项 (端口 14 ~ 17)	每次扫描	读 / 写
D8723	控制信号状态 (端口 18 ~ 21)	每次扫描	读
D8724	RS232C DSR 控制信号状态 (端口 18 ~ 21)	每次扫描	读 / 写
D8725	RS232C DTR 输出控制信号选项 (端口 18 ~ 21)	每次扫描	读 / 写
D8726	控制信号状态 (端口 22 ~ 25)	每次扫描	读
D8727	RS232C DSR 控制信号状态 (端口 22 ~ 25)	每次扫描	读 / 写
D8728	RS232C DTR 输出控制信号选项 (端口 22 ~ 25)	每次扫描	读 / 写
D8729	控制信号状态 (端口 26 ~ 29)	每次扫描	读
D8730	RS232C DSR 控制信号状态 (端口 26 ~ 29)	每次扫描	读 / 写
D8731	RS232C DTR 输出控制信号选项 (端口 26 ~ 29)	每次扫描	读 / 写
D8732	控制信号状态 (端口 30 ~ 33)	每次扫描	读
D8733	RS232C DSR 控制信号状态 (端口 30 ~ 33)	每次扫描	读 / 写
D8734	RS232C DTR 输出控制信号选项 (端口 30 ~ 33)	每次扫描	读 / 写
D8735	从机编号 (端口 10)	—	读 / 写
D8736	从机编号 (端口 11)	—	读 / 写
D8737	从机编号 (端口 12)	—	读 / 写
D8738	从机编号 (端口 13)	—	读 / 写
D8739	从机编号 (端口 14)	—	读 / 写
D8740	从机编号 (端口 15)	—	读 / 写
D8741	从机编号 (端口 16)	—	读 / 写
D8742	从机编号 (端口 17)	—	读 / 写
D8743	从机编号 (端口 18)	—	读 / 写
D8744	从机编号 (端口 19)	—	读 / 写
D8745	从机编号 (端口 20)	—	读 / 写
D8746	从机编号 (端口 21)	—	读 / 写
D8747	从机编号 (端口 22)	—	读 / 写
D8748	从机编号 (端口 23)	—	读 / 写
D8749	从机编号 (端口 24)	—	读 / 写
D8750	从机编号 (端口 25)	—	读 / 写
D8751	从机编号 (端口 26)	—	读 / 写
D8752	从机编号 (端口 27)	—	读 / 写
D8753	从机编号 (端口 28)	—	读 / 写
D8754	从机编号 (端口 29)	—	读 / 写
D8755	从机编号 (端口 30)	—	读 / 写
D8756	从机编号 (端口 31)	—	读 / 写

设备地址	说明	刷新时间	读/写
D8757	从机编号 (端口 32)	—	读/写
D8758	从机编号 (端口 33)	—	读/写
D8759	EMAIL 指令详细错误信息 (以太网端口 1)	—	读
D8760	通信模式信息 (客户端连接)	连接 9 ~ 12	读
D8761		连接 13 ~ 16	读
D8762- D8773	— 保留 —	—	—
D8774	连接 9 的远程主机号 (1 ~ 255)	—	读/写
D8775	连接 10 的远程主机号 (1 ~ 255)	—	读/写
D8776	连接 11 的远程主机号 (1 ~ 255)	—	读/写
D8777	连接 12 的远程主机号 (1 ~ 255)	—	读/写
D8778	连接 13 的远程主机号 (1 ~ 255)	—	读/写
D8779	连接 14 的远程主机号 (1 ~ 255)	—	读/写
D8780	连接 15 的远程主机号 (1 ~ 255)	—	读/写
D8781	连接 16 的远程主机号 (1 ~ 255)	—	读/写
D8782	BACnet 运行状态	—	读
D8783	BACnet 错误信息	—	读
D8784- D8789	— 保留 —	—	—
D8790	EtherNet/IP 运行状态	—	读
D8791	EtherNet/IP 错误信息	—	读
D8792- D8819	— 保留 —	—	—
D8820	从 SD 记忆卡下载用于服务器功能文件执行信息	执行完成时	读
D8821	从 SD 记忆卡下载用于服务器功能文件执行错误信息	执行完成时	读
D8822	Web 服务器当前连接数 (Plus CPU 模块)	约每 1 秒	读
D8823- D8899	— 保留 —	—	—

## 2: 设备地址

### 特殊数据寄存器补充说明

#### ■ D8000: 输入点数

写入 FC6A 型的输入点数。  
写入 CPU 模块和扩展模块的输入点数合计。

#### ■ D8001: 输出点数

写入 FC6A 型的输出点数。  
写入 CPU 模块和扩展模块的输出点数合计值。

#### ■ D8002: CPU 模块类型信息

写入 CPU 模块的类型信息。  
0 (00h): 16-I/O 型  
1 (01h): 24-I/O 型  
2 (02h): 40-I/O 型 (All-in-One CPU 模块)  
18 (12h): 40-I/O 型 (CAN J1939 All-in-One CPU 模块)  
32 (20h): Plus 16-I/O 型  
33 (21h): Plus 32-I/O 型

#### ■ D8005: 通用错误代码

写入 FC6A 型的通用错误代码。如果出现通用错误,则会开启与该错误相对应的位。通过将“1”写入 D8005 的最高有效位,可清除通用错误和用户程序执行错误。有关通用错误代码的详情,请参见《FC6A 型 MICROSmart 用户手册》第 13 章中的“通用错误代码”。

#### ■ D8006: 用户程序执行错误代码

FC6A 型写入用户程序执行错误信息。发生用户程序执行错误时,将写入与错误内容对应的错误代码。有关用户程序执行错误的详情,请参见附录 -19 页上的“用户程序执行错误和梯形图程序执行错误”。

#### ■ D8008 ~ D8021: 日历 / 时钟数据

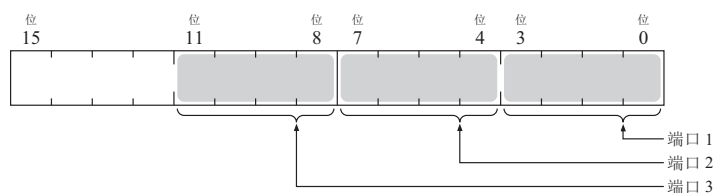
D8008 到 D8021 用于读取来自内部时钟的日历 / 时钟数据,也用于将日历 / 时钟数据写入内部时钟。有关日历 / 时钟数据的详情,请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“时钟功能”。

#### ■ D8022 ~ D8025: 扫描时间数据

D8022 到 D8025 都是特殊数据寄存器,用于检查扫描时间和设置常量扫描时间。有关扫描时间的详情,请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“固定扫描时间”。

#### ■ D8026: 通信模式信息 (端口 1 ~ 3)

表示端口 1 ~ 3 的通信模式。  
设备内各通信端口的分配 (位分配) 如下所示。



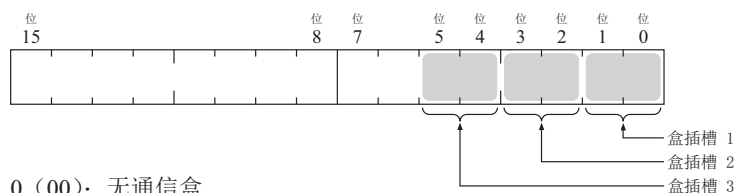
- 0 (0000): 维护通信
- 1 (0001): 用户通信
- 2 (0010): Modbus RTU 主机
- 3 (0011): Modbus RTU 从机
- 4 (0100): 数据连接通信

#### ■ D8029: 系统版本信息

写入 CPU 模块的系统软件的版本号。

### ■ D8030: 通信盒信息

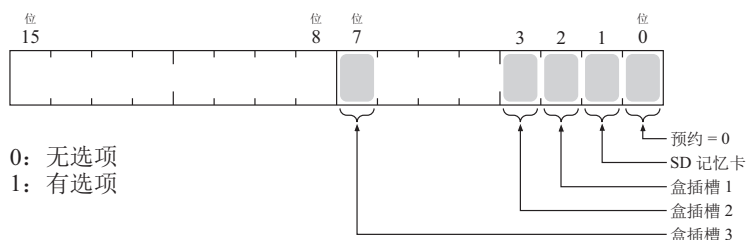
表示端口 2 及端口 3 上的通信盒连接状况。  
设备内各通信端口的分配（位分配）如下所示。



- 0 (00): 无通信盒
- 1 (01): RS232C 通信盒连接
- 2 (10): RS485 通信盒连接
- 3 (11): Bluetooth 通信盒连接

### ■ D8031: 选项连接信息

写入选项连接信息。  
设备内的分配（位分配）如下所示。



- 0: 无选项
- 1: 有选项

### ■ D8032 ~ D8035、D8214、D8215: 中断输入跳转目标标签编号

写入中断输入的跳转目标标签编号。如需使用中断输入，请写入与中断输入中分配的特殊数据寄存器相对应的标签编号。有关中断输入的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“中断输入”。

- D8032 = I1
- D8033 = I3
- D8034 = I4
- D8035 = I6
- D8214 = I7
- D8215 = I0

### ■ D8036: 定时器中断跳转目标标签编号

写入发生定时器中断时的跳转目标标签编号。如需使用定时器中断，请存储相应的标签编号。  
有关定时器中断的详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“定时器中断”。

### ■ D8037: 扩展模块连接台数

写入 CPU 模块所连接的扩展模块（I/O 模块、PID 模块及通信模块）的台数。

### ■ D8052: J1939 通信错误代码

J1939 通信中发生错误时，将写入错误代码。有关 J1939 通信错误代码的详情，请参见《FC6A 型 MICROSmart 通信手册》第 8 章中的“J1939 通信错误代码（D8052）”。

### ■ D8056: 电池电压

以 1mV 为单位写入备份用电池的电池电压测量结果。

- 0: 测量错误

### ■ D8057: 模拟量 (AI0)

将模拟量值转换为数字值进行写入。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“模拟量”。

### ■ D8058: 内置模拟量输入 (AI1)

将模拟量输入的模拟量输入值（0 ~ 10V DC）转换为数字值进行写入。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“内置模拟量输入”。

### ■ D8059: 模拟量输入状态 AI0

写入模拟量的模拟量输入状态。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“模拟量”。

## 2: 设备地址

### ■ D8060: 模拟量输入状态 AI1

写入模拟量输入的模拟量输入状态。有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“内置模拟量输入”。

### ■ D8067: 背光亮起时间

写入 HMI 模块的背光亮起时间。背光亮起时间可通过更改 D8067 的值, 在 1 ~ 65,535s 范围内进行设置。将 D8067 的值设为 0s 时, 背光将始终亮起。背光亮起时间可在 HMI 模块的环境设置中进行更改。有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 7 章中的“设置 LCD 背光亮起时间”。

### ■ D8069 ~ D8099: 从机 (1 ~ 31) 通信状态 / 错误

用于数据连接通信的特殊数据寄存器。有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 7 章中的“主机站”。

### ■ D8040 ~ D8045、D8100、D8102、D8103、D8735 ~ D8758: 从机编号 (端口 1 ~ 33)

端口 1 ~ 33 的通信模式为维护通信、Modbus RTU 从机或数据连接通信时, 将写入从机编号。通过在“功能设置”中更改相应设备的值, 可更改从机编号。

D8100 = 端口 1 从机编号

D8102 = 端口 2 从机编号

D8103 = 端口 3 从机编号

D8040 = 端口 4 从机编号

D8041 = 端口 5 从机编号

D8042 = 端口 6 从机编号

D8043 = 端口 7 从机编号

D8044 = 端口 8 从机编号

D8045 = 端口 9 从机编号

D8735 = 端口 10 从机编号

D8736 = 端口 11 从机编号

D8737 = 端口 12 从机编号

D8738 = 端口 13 从机编号

D8739 = 端口 14 从机编号

D8740 = 端口 15 从机编号

D8741 = 端口 16 从机编号

D8742 = 端口 17 从机编号

D8743 = 端口 18 从机编号

D8744 = 端口 19 从机编号

D8745 = 端口 20 从机编号

D8746 = 端口 21 从机编号

D8747 = 端口 22 从机编号

D8748 = 端口 23 从机编号

D8749 = 端口 24 从机编号

D8750 = 端口 25 从机编号

D8751 = 端口 26 从机编号

D8752 = 端口 27 从机编号

D8753 = 端口 28 从机编号

D8754 = 端口 29 从机编号

D8755 = 端口 30 从机编号

D8756 = 端口 31 从机编号

D8757 = 端口 32 从机编号

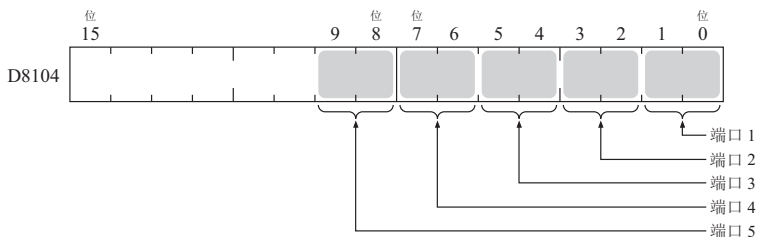
D8758 = 端口 33 从机编号

有关各通信模式的详情, 请参见《FC6A 型 MICROSmart 通信手册》的以下内容。

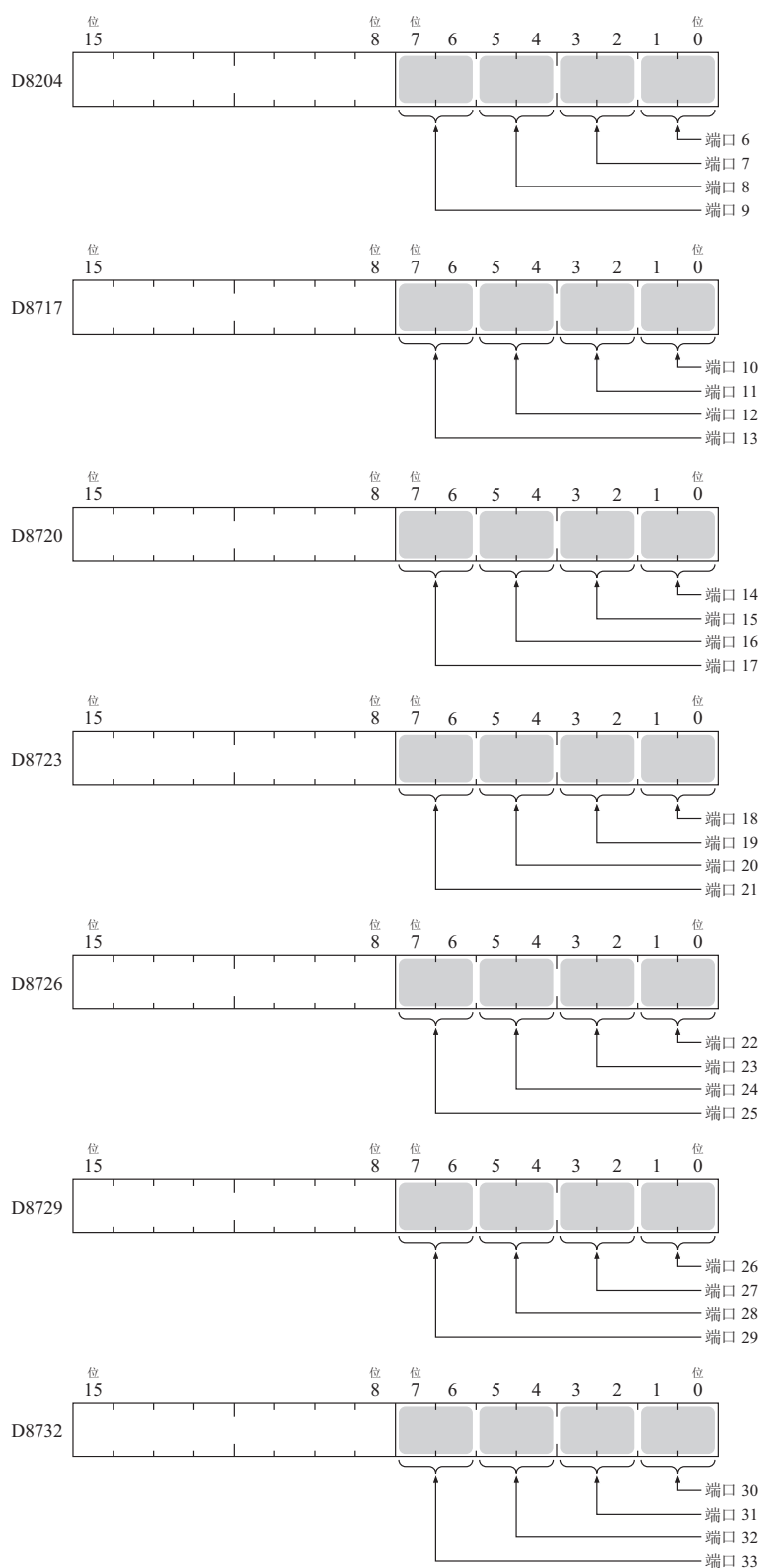
- 维护通信: 第 4 章“维护通信”
- Modbus RTU 从机: 第 6 章“Modbus 通信”
- 数据连接通信: 第 7 章“数据连接通信”

### ■ D8104、D8204、D8717、D8720、D8723、D8726、D8729、D8732: 控制信号状态 (端口 1 ~ 33)

写入 DSR、DTR 的各控制线的信号状态。以停止中及运行中的结束处理进行刷新。设备内各通信端口的分配 (位分配) 如下所示。







0 (00): DTR 和 DSR 皆为关闭。

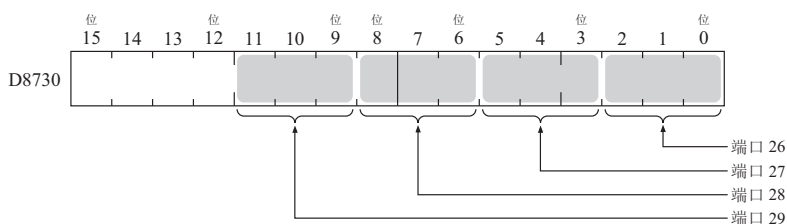
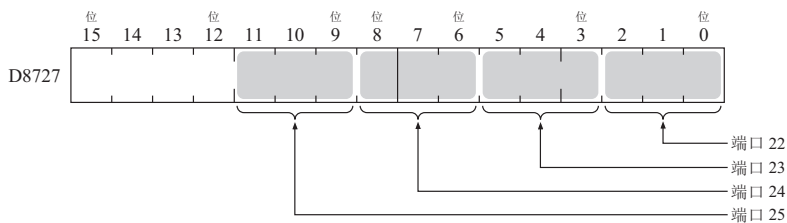
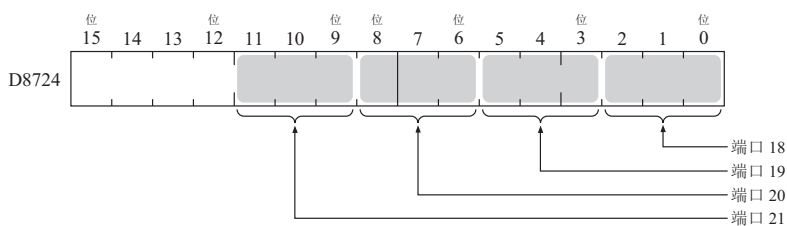
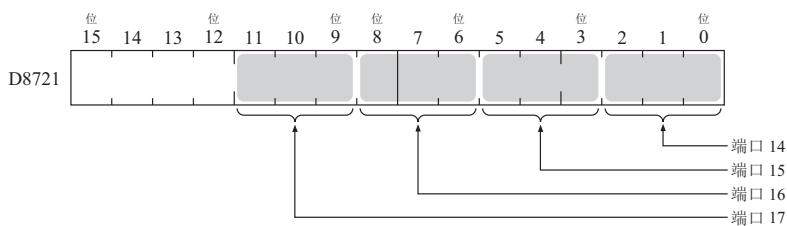
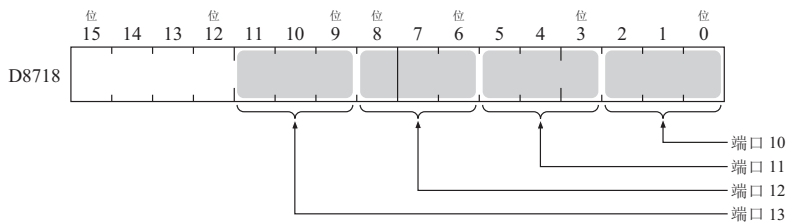
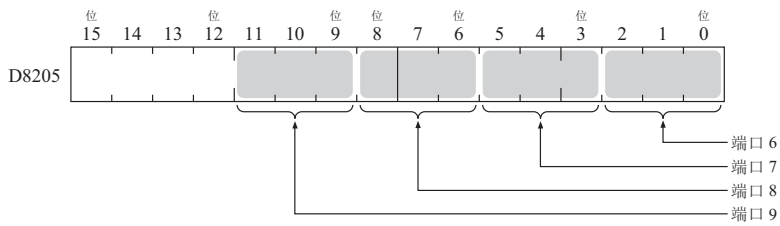
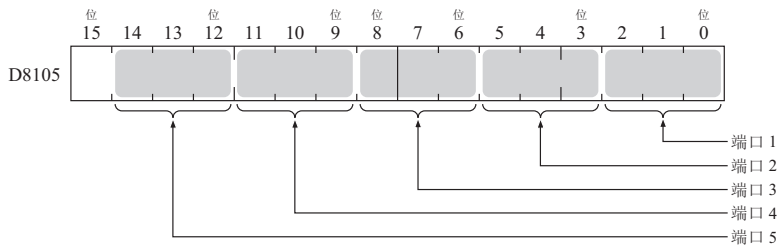
1 (01): DTR 为关闭, DSR 为打开。

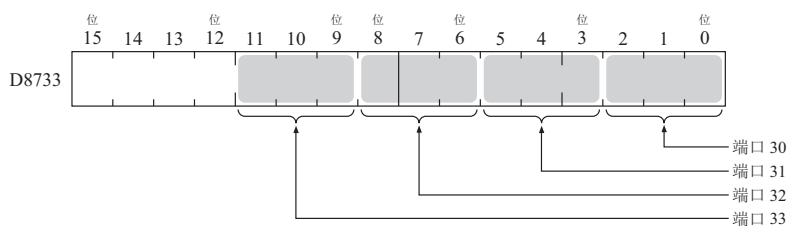
2 (10): DTR 为打开, DSR 为关闭。

3 (11): DTR 和 DSR 皆为打开。

## 2: 设备地址

- **D8105、D8205、D8718、D8721、D8724、D8727、D8730、D8733: RS232C DSR 控制信号状态 (端口 1 ~ 33)**  
 写入 DSR、DTR 的各控制线的信号状态。以停止中及运行中的结束处理进行刷新。设备内各通信端口的分配 (位分配) 如下所示。





0 (000): FC6A 型的收发控制中不会使用 DSR 信号的状态。

如无需进行 DSR 信号控制，请在该状态下使用。

1 (001): DSR 信号打开时，FC6A 型可进行收发。



2 (010): DSR 信号关闭时，FC6A 型可进行收发。



3 (011): DSR 信号打开时，可进行发送（始终可发送）。



这通常称为“Busy 控制”，用于处理速度较慢的设备（打印机等）的发送控制。

（如果以连接设备立场来看，则为输入数据的限制。）

4 (100): DSR 信号关闭时，可进行发送。



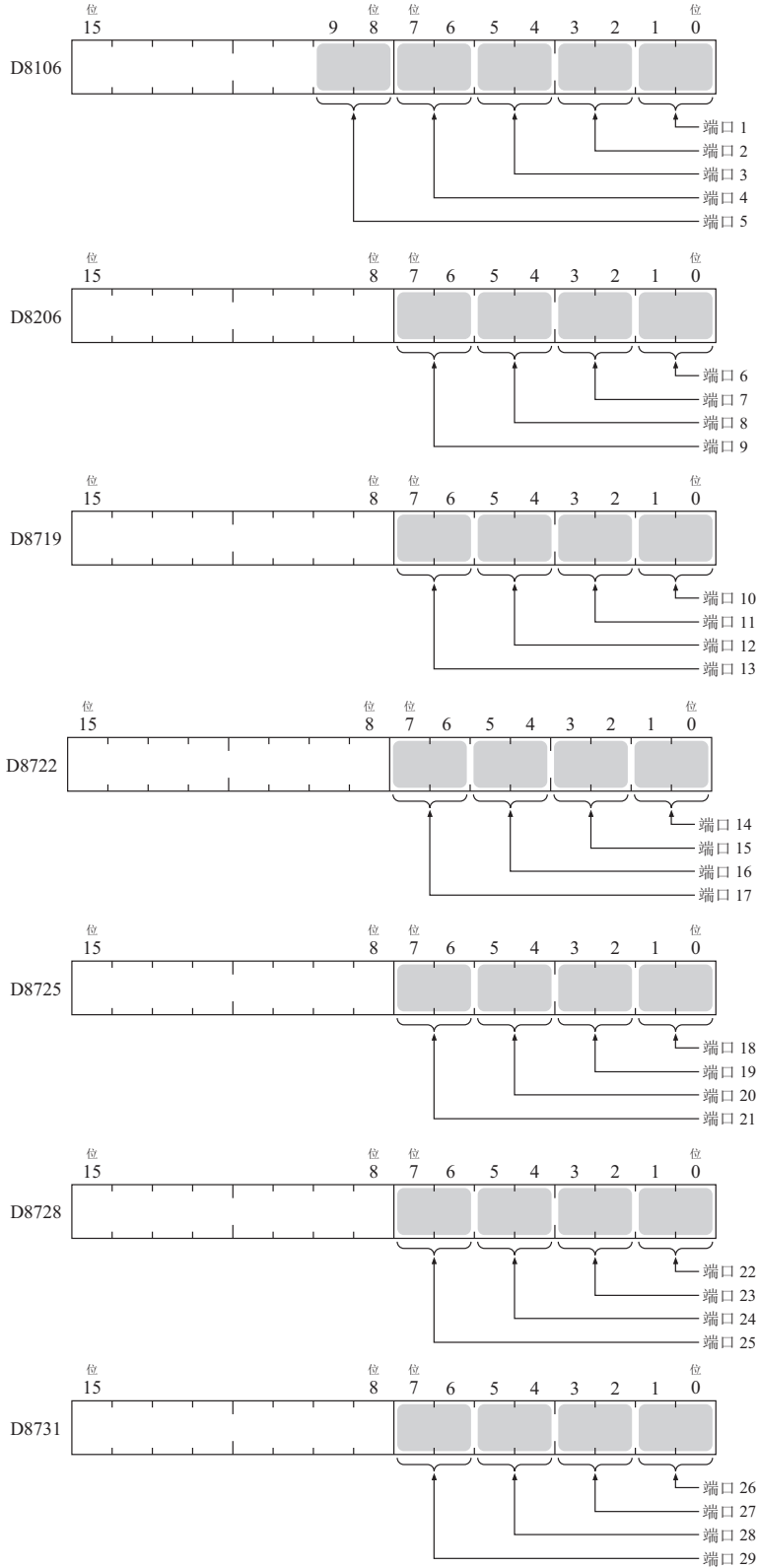
5 以上: 进行与预置值“000”相同的动作。

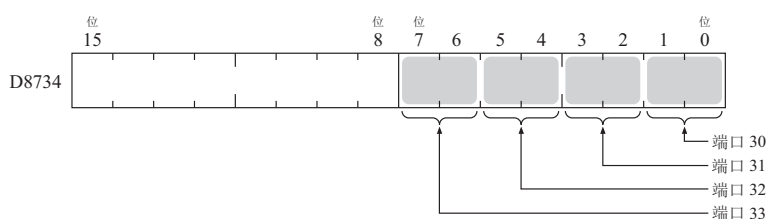
## 2: 设备地址

### ■ D8106、D8206、D8719、D8722、D8725、D8728、D8731、D8734: RS232C DTR 输出控制信号选项（端口 1 ~ 33）

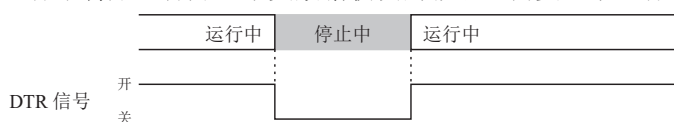
用于在目标设备中表示 FC6A 型的控制状态或收发状态时。该控制线为从 FC6A 型到目标设备的输出信号。仅在用户通信时启用。

设备内各通信端口的分配（位分配）如下所示。





- 0 (00): 在 FC6A 型为运行状态时打开, 为停止状态时关闭。  
运行中将始终打开, 不受数据收发的影响。需要显示运行状态时进行设置。



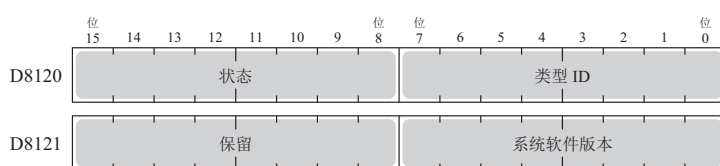
- 1 (01): 始终关闭。  
2 (10): 对接收数据进行流程控制时进行设置。可从目标设备接收数据时, 将打开。  
不可接收时, 将关闭。



- 3 (11): 进行与预置值“0”相同的动作。

#### ■ D8120、D8121: HMI 模块信息

写入 HMI 模块的类型信息。  
信息的分配 (位分配) 如下所示。

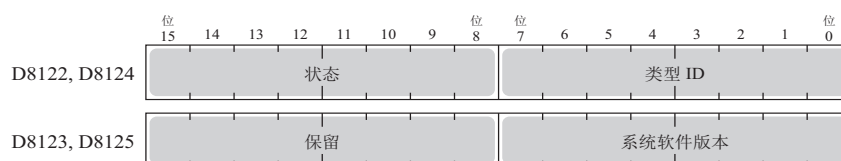


将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情, 请参见第 2-63 页上的“类型 ID、状态一览”。

#### ■ D8122 ~ D8127: 盒插槽信息

写入盒插槽 1 ~ 3 的信息。  
信息的分配 (位分配) 如下所示。



D8122、D8123 = 盒插槽 1 信息  
D8124、D8125 = 盒插槽 2 信息  
D8126、D8127 = 盒插槽 3 信息

有关类型 ID 和状态的详情, 请参见第 2-63 页上的“类型 ID、状态一览”。

## 2: 设备地址

### ■ D8170、D8171、D8174、D8175、D8178、D8179: 模拟 I/O 盒输入输出

写入模拟 I/O 盒的模拟量输入输出值。

模拟量输入型时: 将模拟 I/O 盒中输入的模拟量值转换为数字值进行写入。

模拟量输出型时: 将存储为数字值的数字值转换为模拟量值从模拟 I/O 盒进行输出。

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 10 章中的“模拟 I/O 盒”。

D8170 = AI2/AQ2

D8171 = AI3/AQ3

D8174 = AI4/AQ4

D8175 = AI5/AQ5

D8178 = AI6/AQ6

D8179 = AI7/AQ7

### ■ D8172、D8173、D8176、D8177、D8180、D8181: 模拟 I/O 盒状态

写入模拟 I/O 盒的模拟状态。

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 10 章中的“模拟 I/O 盒”。

D8172 = AI2/AQ2

D8173 = AI3/AQ3

D8176 = AI4/AQ4

D8177 = AI5/AQ5

D8180 = AI6/AQ6

D8181 = AI7/AQ7

### ■ D8192 ~ D8203、D8210 ~ D8213、D8216 ~ D8229、D8232 ~ D8237: 高速计数器

这些特殊数据寄存器用于高速计数器功能和频率测量功能。

有关高速计数器的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“高速计数器”。

D8210 ~ D8213、D8216、D8217 = 高速计数器 (组 1/I0)

D8218 ~ D8221、D8234、D8235 = 高速计数器 (组 3/I3)

D8222 ~ D8225、D8236、D8237 = 高速计数器 (组 4/I4)

D8226 ~ D8229、D8232、D8233 = 高速计数器 (组 5/I6)

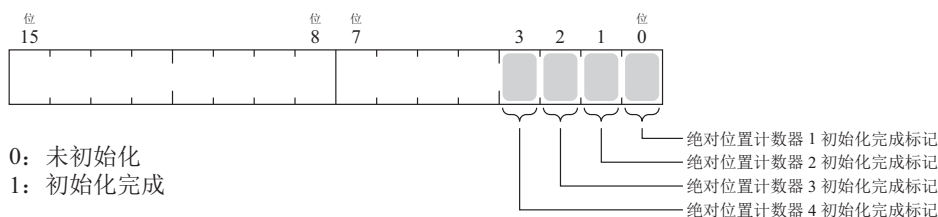
D8192 ~ D8197 = 高速计数器 (组 2/I1)

D8198 ~ D8203 = 高速计数器 (组 6/I7)

### ■ D8239: 绝对位置管理状态

表示绝对位置管理的状态。

设备内的绝对位置计数器初始化完成标记的分配 (位分配) 如下所示。有关绝对位置管理状态的详情, 请参见第 18-69 页上的“ABS (绝对位置设置)”。



### ■ D8240 ~ D8247: 绝对位置计数器 1 ~ 4

根据脉冲输出写入绝对位置。绝对位置可以 ABS 指令进行初始化。

以脉冲输出指令 (RAMP/ ARAMP/ ZRN/ JOG) 设置为有方向控制时, 将根据输出结果加减计算绝对位置。

有关绝对位置计数器的详

情, 请参见第 18-69 页上的“ABS (绝对位置设置)”。

D8240、D8241 = 绝对位置计数器 1

D8242、D8243 = 绝对位置计数器 2

D8244、D8245 = 绝对位置计数器 3

D8246、D8247 = 绝对位置计数器 4

### ■ D8250: SD 记忆卡容量显示

以 MB 为单位显示已识别 SD、SDHC (最大 32GB) 对应的 SD 记忆卡容量。

未插入 SD 记忆卡或记忆卡未被识别时, 将变为 0。

### ■ D8251: SD 记忆卡可用空间显示

以 MB 为单位显示已识别 SD 记忆卡的可用空间。  
未插入 SD 记忆卡或记忆卡未被识别时, 将变为 0。

### ■ D8254: SD 记忆卡 下载 / 上传执行信息

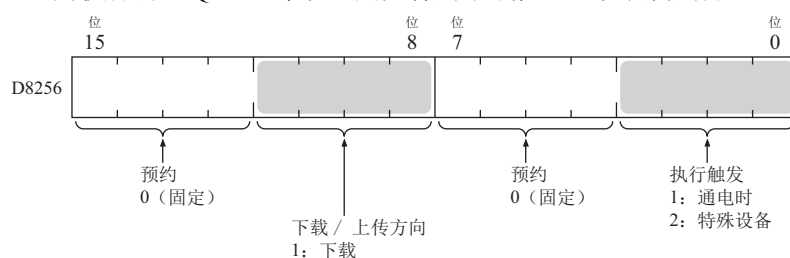
已执行的 SD 记忆卡 下载 / 上传的信息。  
有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

### ■ D8255: SD 记忆卡 下载 / 上传执行状态

已执行的 SD 记忆卡 下载 / 上传的状态。  
有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

### ■ D8256: 从 SD 记忆卡进行 MQTT 基本设置下载的执行信息

此为执行的 MQTT 基本设置用文件的下载信息。设备内的分配 (位分配) 如下所示。



### ■ D8257: 从 SD 记忆卡进行 MQTT 基本设置下载的执行错误信息

此为执行的 MQTT 基本设置用文件的下载错误信息。有关详情, 请参见《FC6A 型 MICROSmart 用户手册》中的“下载 MQTT 基本设置”。

### ■ D8260: 配方块编号

使用特殊内部继电器进行读写的配方块编号。  
已指定块编号的所有频道皆为目标。

### ■ D8261: 执行配方块编号

已执行配方的配方块编号。  
在开始配方块时进行刷新, 完成后则维持状态。

### ■ D8262: 执行配方频道编号

已执行配方的频道编号。  
在开始频道处理时进行刷新, 完成后则维持状态。

### ■ D8263: 执行配方动作

已执行配方动作的相关信息。  
有关配方的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

### ■ D8264: 执行配方状态

已执行配方的状态。  
有关配方的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

### ■ D8265: 执行配方错误信息

已执行配方的错误信息。  
有关配方的详情, 请参见《FC6A 型 MICROSmart 用户手册》第 11 章中的“配方功能”。

### ■ D8266: 配方内存 (ROM- 区域 1) 读取次数

将设备值作为配方数据读取到内存 (ROM- 区域 1) 中的次数以 100 单位进行存储。

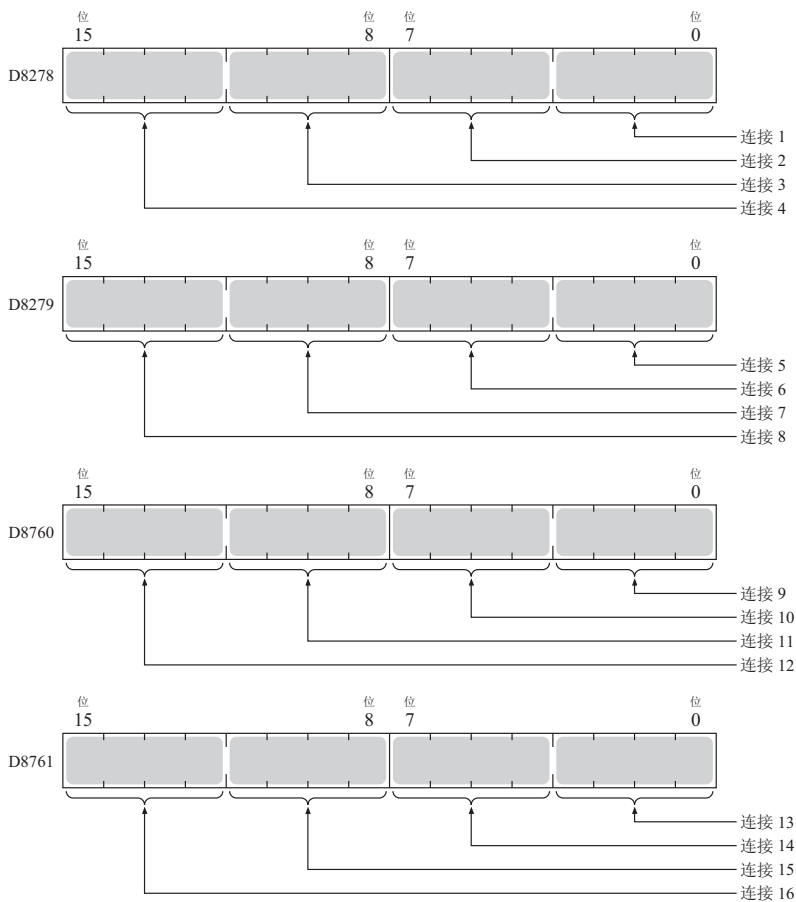
### ■ D8267: 配方内存 (ROM- 区域 2) 读取次数

将设备值作为配方数据读取到内存 (ROM- 区域 2) 中的次数以 100 单位进行存储。

### ■ D8278、D8279、D8760、D8761: 通信模式信息 (客户端连接) (连接 1 ~ 16)

D8278 = 表示连接 1 ~ 4 的通信模式。  
D8279 = 表示连接 5 ~ 8 的通信模式。  
D8760 = 表示连接 9 ~ 12 的通信模式。  
D8761 = 表示连接 13 ~ 16 的通信模式。  
设备内各连接的分配 (位分配) 如下所示。

## 2: 设备地址



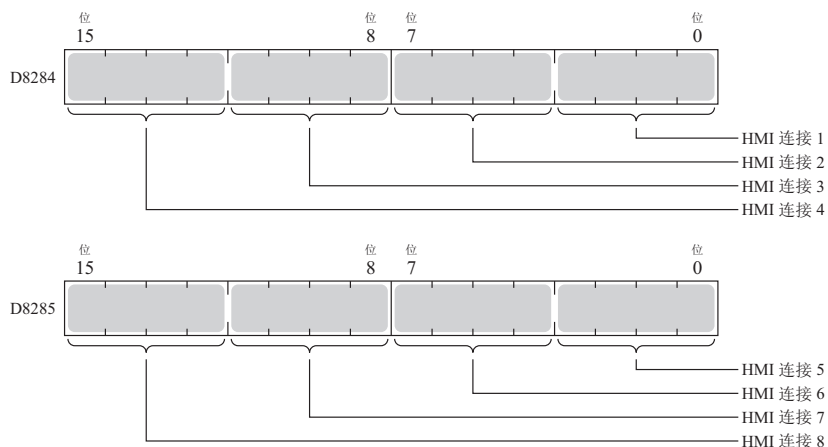
- 客户端连接（最高有效位 = 0）
  - 0000: 未使用
  - 0001: 用户通信
  - 0010: Modbus TCP 客户端
  - 0100: 用户通信 UDP
- 服务器连接（最高有效位 = 1）
  - 1000: 维护通信
  - 1001: 用户通信
  - 1010: Modbus TCP 服务器

### ■ D8284、D8285: 通信模式信息（HMI 连接）（HMI 连接 1 ~ 8）

D8284: 表示 HMI 连接 1 ~ 4 的通信模式。

D8285: 表示 HMI 连接 5 ~ 8 的通信模式。

设备内各连接的分配（位分配）如下所示。





- 客户端连接（最高有效位 = 0）  
0000: 未使用
- 服务器连接（最高有效位 = 1）  
1000: 维护通信

### ■ D8303: CPU 模块以太网端口 1 IP 设置 / DNS 设置切换

通过向 D8303 写入下页表中的预置值，并将 M8190 切换到 ON，可更改以太网端口 1 的 IP 设置 / DNS 设置。  
如需使用本功能，请在 WindLDR 的“功能设置”中启用以下项目。

- All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块  
“功能设置”的“网络设置”中“启用 D8630（IP 设置 / DNS 设置切换）”
- Plus CPU 模块  
“功能设置”的“以太网端口 1”中“启用 D8630（IP 设置 / DNS 设置切换）”

## 2: 设备地址

预置值的含义如下所示。

预置值	IP 设置 /DNS 设置
0	依照功能设置。
1	启用 DHCP。
2	依照特殊数据寄存器（D8304 ~ D8323）的设置。

- **D8304 ~ D8307: CPU 模块以太网端口 1 IP 地址（只写）**  
用于写入 CPU 模块的 IP 地址。  
IP 地址：设置为 aaa.bbb.ccc.ddd 时，将进行如下写入。  
D8304 = aaa、D8305 = bbb、D8306 = ccc、D8307 = ddd
- **D8308 ~ D8311: CPU 模块以太网端口 1 子网掩码（只写）**  
用于写入 CPU 模块的子网掩码。  
子网掩码：aaa.bbb.ccc.ddd 时，将进行如下写入。  
D8308 = aaa、D8309 = bbb、D8310 = ccc、D8311 = ddd
- **D8312 ~ D8315: CPU 模块以太网端口 1 默认网关（只写）**  
用于写入 CPU 模块的默认网关。  
默认网关：aaa.bbb.ccc.ddd 时，将进行如下写入。  
D8312 = aaa、D8313 = bbb、D8314 = ccc、D8315 = ddd
- **D8316 ~ D8319: CPU 模块以太网端口 1 首选 DNS 服务器（只写）**  
用于写入 CPU 模块的首选 DNS 服务器。  
首选 DNS 服务器：aaa.bbb.ccc.ddd 时，将进行如下写入。  
D8316 = aaa、D8317 = bbb、D8318 = ccc、D8319 = ddd
- **D8320 ~ D8323: CPU 模块以太网端口 1 备用 DNS 服务器（只写）**  
用于写入 CPU 模块的备用 DNS 服务器。  
备用 DNS 服务器：aaa.bbb.ccc.ddd 时，将进行如下写入。  
D8320 = aaa、D8321 = bbb、D8322 = ccc、D8323 = ddd
- **D8324 ~ D8329: CPU 模块以太网端口 1 MAC 地址（只读当前值）**  
CPU 模块的 MAC 地址将如下所示写入到各特殊数据寄存器中。  
例）MAC 地址：AA-BB-CC-DD-EE-FF  
D8324=AA、D8325=BB、D8326=CC、D8327=DD、D8328=EE、D8329=FF
- **D8330 ~ D8333: CPU 模块以太网端口 1 IP 地址（只读当前值）**  
CPU 模块的 IP 地址将如下所示写入到各特殊数据寄存器中。  
例）本机 IP 地址：aaa.bbb.ccc.ddd  
D8330=aaa、D8331=bbb、D8332=ccc、D8333=ddd
- **D8334 ~ D8337: CPU 模块以太网端口 1 子网掩码（只读当前值）**  
CPU 模块的子网掩码将如下所示写入到各特殊数据寄存器中。  
例）子网掩码：aaa.bbb.ccc.ddd  
D8334=aaa、D8335=bbb、D8336=ccc、D8337=ddd
- **D8338 ~ D8341: CPU 模块以太网端口 1 默认网关（只读当前值）**  
CPU 模块的默认网关将如下所示写入到各特殊数据寄存器中。  
例）默认网关：aaa.bbb.ccc.ddd  
D8338=aaa、D8339=bbb、D8340=ccc、D8341=ddd
- **D8342 ~ D8345: CPU 模块以太网端口 1 首选 DNS 服务器（只读当前值）**  
CPU 模块的首选 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。  
例）首选 DNS 服务器：aaa.bbb.ccc.ddd  
D8342=aaa、D8343=bbb、D8344=ccc、D8345=ddd
- **D8346 ~ D8349: CPU 模块以太网端口 1 备用 DNS 服务器（只读当前值）**  
CPU 模块的备用 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。  
例）备用 DNS 服务器：aaa.bbb.ccc.ddd  
D8346=aaa、D8347=bbb、D8348=ccc、D8349=ddd

### ■ D8350 ~ D8381、D8677 ~ D8708: 已接通从机的 IP 地址

如下所示写入正在访问连接的目标设备的 IP 地址。

连接 1 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8350=aaa、D8351=bbb、D8352=ccc、D8353=ddd  
 连接 2 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8354=aaa、D8355=bbb、D8356=ccc、D8357=ddd  
 连接 3 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8358=aaa、D8359=bbb、D8360=ccc、D8361=ddd  
 连接 4 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8362=aaa、D8363=bbb、D8364=ccc、D8365=ddd  
 连接 5 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8366=aaa、D8367=bbb、D8368=ccc、D8369=ddd  
 连接 6 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8370=aaa、D8371=bbb、D8372=ccc、D8373=ddd  
 连接 7 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8374=aaa、D8375=bbb、D8376=ccc、D8377=ddd  
 连接 8 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8378=aaa、D8379=bbb、D8380=ccc、D8381=ddd  
 连接 9 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8677=aaa、D8678=bbb、D8679=ccc、D8680=ddd  
 连接 10 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8681=aaa、D8682=bbb、D8683=ccc、D8684=ddd  
 连接 11 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8685=aaa、D8686=bbb、D8687=ccc、D8688=ddd  
 连接 12 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8689=aaa、D8690=bbb、D8691=ccc、D8692=ddd  
 连接 13 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8693=aaa、D8694=bbb、D8695=ccc、D8696=ddd  
 连接 14 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8697=aaa、D8698=bbb、D8699=ccc、D8700=ddd  
 连接 15 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8701=aaa、D8702=bbb、D8703=ccc、D8704=ddd  
 连接 16 连接的 IP 地址: aaa.bbb.ccc.ddd 时  
 D8705=aaa、D8706=bbb、D8707=ccc、D8708=ddd

### ■ D8382 ~ D8387: HMI 模块 MAC 地址 (只读当前值)

MAC 地址将如下所示写入到各特殊数据寄存器中。

例) MAC 地址: AA-BB-CC-DD-EE-FF

D8382=AA、D8383=BB、D8384=CC、D8385=DD、D8386=EE、D8387=FF

### ■ D8388 ~ D8391: HMI 模块 IP 地址 (只读当前值)

HMI 模块的 IP 地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块 IP 地址: aaa.bbb.ccc.ddd

D8388=aaa、D8389=bbb、D8390=ccc、D8391=ddd

### ■ D8392 ~ D8395: HMI 模块子网掩码 (只读当前值)

HMI 模块的子网掩码的值将如下所示写入到各特殊数据寄存器中。

例) HMI 模块子网掩码: aaa.bbb.ccc.ddd

D8392=aaa、D8393=bbb、D8394=ccc、D8395=ddd

### ■ D8396 ~ D8399: HMI 模块默认网关 (只读当前值)

HMI 模块的默认网关的地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块默认网关: aaa.bbb.ccc.ddd

D8396=aaa、D8397=bbb、D8398=ccc、D8399=ddd

### ■ D8400 ~ D8403: HMI 模块首选 DNS 服务器 (只读当前值)

HMI 模块的首选 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块首选 DNS 服务器: aaa.bbb.ccc.ddd

D8400=aaa、D8401=bbb、D8402=ccc、D8403=ddd

## 2: 设备地址

---

### ■ D8404 ~ D8407: HMI 模块备用 DNS 服务器 (只读当前值)

HMI 模块的备用 DNS 服务器的地址将如下所示写入到各特殊数据寄存器中。

例) HMI 模块备用 DNS 服务器: aaa.bbb.ccc.ddd

D8404=aaa、D8405=bbb、D8406=ccc、D8407=ddd

### ■ D8413: 时区偏移量

可以 15 分为单位调整功能设置中设置的时区。

有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“SNTP 设置”。

### ■ D8414: SNTP 运行状态

写入通过操作 M8191 (SNTP 时间获取标记) 获取时间信息、或自动获取时间信息时的动作状态。

有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“SNTP 设置”。

### ■ D8415: SNTP 访问经过时间

以分为单位写入自最后从 SNTP 服务器获取时间信息起的经过时间。

有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 3 章中的“SNTP 设置”。

### ■ D8429: HMI 模块连接信息参考连接编号

将已指定连接编号的连接信息反映到 D8430 ~ D8434 中。写入 0 时, 将向 D8430 ~ D8434 中写入 0。即使指定了不存在的连接编号, 也会向 D8430 ~ D8434 中写入 0。

### ■ D8430 ~ D8433: HMI 模块连接信息参考连接 IP 地址

正在访问连接的终端的 IP 地址将如下所示写入到各特殊数据寄存器中。

例) 读取目标的 IP 地址: aaa.bbb.ccc.ddd

D8430=aaa、D8431=bbb、D8432=ccc、D8433=ddd

### ■ D8434: HMI 模块连接信息参考连接端口编号

写入正在访问连接的终端的端口编号。

### ■ D8437 ~ D8440: HMI 模块 IP 地址 (只写)

用于写入 HMI 模块的 IP 地址。

HMI 模块 IP 地址: 设置为 aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8437 = aaa、D8438 = bbb、D8439 = ccc、D8440 = ddd

### ■ D8441 ~ D8444: HMI 模块子网掩码 (只写)

用于写入 HMI 模块的子网掩码。

HMI 模块子网掩码: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8441 = aaa、D8442 = bbb、D8443 = ccc、D8444 = ddd

### ■ D8445 ~ D8448: HMI 模块默认网关 (只写)

用于写入 HMI 模块的默认网关。

HMI 模块默认网关: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8445 = aaa、D8446 = bbb、D8447 = ccc、D8448 = ddd

### ■ D8449 ~ D8452: HMI 模块首选 DNS 服务器 (只写)

用于写入 HMI 模块的首选 DNS 服务器。

HMI 模块首选 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8449 = aaa、D8450 = bbb、D8451 = ccc、D8452 = ddd

### ■ D8453 ~ D8456: HMI 模块备用 DNS 服务器 (只写)

用于写入 HMI 模块的备用 DNS 服务器。

HMI 模块备用 DNS 服务器: aaa.bbb.ccc.ddd 时, 将进行如下写入。

D8453 = aaa、D8454 = bbb、D8455 = ccc、D8456 = ddd

### ■ D8457: EMAIL 指令详细错误信息 (HMI- 以太网端口)

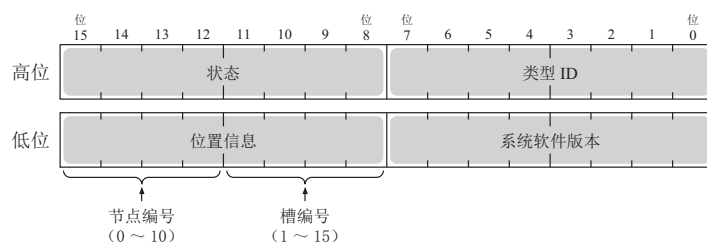
写入 EMAIL 指令的详细错误信息。

有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 12 章中的“确认 EMAIL 指令的错误详情”。

### ■ D8470 ~ D8595: 扩展模块 槽信息

写入扩展模块的类型信息。

信息的分配（位分配）如下所示。



高位	低位	说明
D8470、D8471		= 扩展模块 槽 1 信息
D8472、D8473		= 扩展模块 槽 2 信息
D8474、D8475		= 扩展模块 槽 3 信息
D8476、D8477		= 扩展模块 槽 4 信息
D8478、D8479		= 扩展模块 槽 5 信息
D8480、D8481		= 扩展模块 槽 6 信息
D8482、D8483		= 扩展模块 槽 7 信息
D8484、D8485		= 扩展模块 槽 8 信息
D8486、D8487		= 扩展模块 槽 9 信息
D8488、D8489		= 扩展模块 槽 10 信息
D8490、D8491		= 扩展模块 槽 11 信息
D8492、D8493		= 扩展模块 槽 12 信息
D8494、D8495		= 扩展模块 槽 13 信息
D8496、D8497		= 扩展模块 槽 14 信息
D8498、D8499		= 扩展模块 槽 15 信息
D8500、D8501		= 扩展模块 槽 16 信息
D8502、D8503		= 扩展模块 槽 17 信息
D8504、D8505		= 扩展模块 槽 18 信息
D8506、D8507		= 扩展模块 槽 19 信息
D8508、D8509		= 扩展模块 槽 20 信息
D8510、D8511		= 扩展模块 槽 21 信息
D8512、D8513		= 扩展模块 槽 22 信息
D8514、D8515		= 扩展模块 槽 23 信息
D8516、D8517		= 扩展模块 槽 24 信息
D8518、D8519		= 扩展模块 槽 25 信息
D8520、D8521		= 扩展模块 槽 26 信息
D8522、D8523		= 扩展模块 槽 27 信息
D8524、D8525		= 扩展模块 槽 28 信息
D8526、D8527		= 扩展模块 槽 29 信息
D8528、D8529		= 扩展模块 槽 30 信息
D8530、D8531		= 扩展模块 槽 31 信息
D8532、D8533		= 扩展模块 槽 32 信息
D8534、D8535		= 扩展模块 槽 33 信息
D8536、D8537		= 扩展模块 槽 34 信息
D8538、D8539		= 扩展模块 槽 35 信息
D8540、D8541		= 扩展模块 槽 36 信息
D8542、D8543		= 扩展模块 槽 37 信息
D8544、D8545		= 扩展模块 槽 38 信息
D8546、D8547		= 扩展模块 槽 39 信息
D8548、D8549		= 扩展模块 槽 40 信息
D8550、D8551		= 扩展模块 槽 41 信息

## 2: 设备地址

高位	低位	
D8552、D8553		=扩展模块槽 42 信息
D8554、D8555		=扩展模块槽 43 信息
D8556、D8557		=扩展模块槽 44 信息
D8558、D8559		=扩展模块槽 45 信息
D8560、D8561		=扩展模块槽 46 信息
D8562、D8563		=扩展模块槽 47 信息
D8564、D8565		=扩展模块槽 48 信息
D8566、D8567		=扩展模块槽 49 信息
D8568、D8569		=扩展模块槽 50 信息
D8570、D8571		=扩展模块槽 51 信息
D8572、D8573		=扩展模块槽 52 信息
D8574、D8575		=扩展模块槽 53 信息
D8576、D8577		=扩展模块槽 54 信息
D8578、D8579		=扩展模块槽 55 信息
D8580、D8581		=扩展模块槽 56 信息
D8582、D8583		=扩展模块槽 57 信息
D8584、D8585		=扩展模块槽 58 信息
D8586、D8587		=扩展模块槽 59 信息
D8588、D8589		=扩展模块槽 60 信息
D8590、D8591		=扩展模块槽 61 信息
D8592、D8593		=扩展模块槽 62 信息
D8594、D8595		=扩展模块槽 63 信息

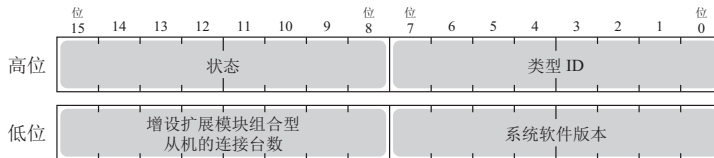
将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情，请参见第 2-63 页上的“类型 ID、状态一览”。

### ■ D8596、D8597：增设扩展模块组合型主机槽信息

写入增设扩展模块组合型主机的类型信息。

信息的分配（位分配）如下所示。



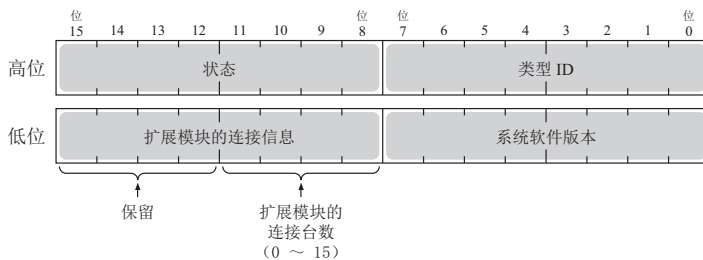
将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情，请参见第 2-63 页上的“类型 ID、状态一览”。

### ■ D8598 ~ D8617：增设扩展模块组合型从机槽信息

写入增设扩展模块组合型从机的类型信息。

信息的分配（位分配）如下所示。



高位	低位	
D8598、D8599		=增设扩展模块组合型从机（第 1 台）槽信息
D8600、D8601		=增设扩展模块组合型从机（第 2 台）槽信息

高位 低位

D8602、D8603 = 增设扩展模块组合型从机（第 3 台）槽信息  
 D8604、D8605 = 增设扩展模块组合型从机（第 4 台）槽信息  
 D8606、D8607 = 增设扩展模块组合型从机（第 5 台）槽信息  
 D8608、D8609 = 增设扩展模块组合型从机（第 6 台）槽信息  
 D8610、D8611 = 增设扩展模块组合型从机（第 7 台）槽信息  
 D8612、D8613 = 增设扩展模块组合型从机（第 8 台）槽信息  
 D8614、D8615 = 增设扩展模块组合型从机（第 9 台）槽信息  
 D8616、D8617 = 增设扩展模块组合型从机（第 10 台）槽信息

将在系统软件版本中显示向设备内写入的软件版本。

有关类型 ID 和状态的详情，请参见第 2-63 页上的“类型 ID、状态一览”。

#### ■ D8618: 增设扩展模块组合型 I/O 刷新时间 当前值

连接增设扩展模块组合型以后的扩展模块的 I/O 刷新时间的当前值会以 1ms 为单位进行存储。

#### ■ D8619: 增设扩展模块组合型 I/O 刷新时间 最大值

连接增设扩展模块组合型以后的扩展模块的 I/O 刷新时间的最大值会以 1ms 为单位进行存储。

#### ■ D8630: CPU 模块以太网端口 2 IP 设置 / DNS 设置切换

通过向 D8630 写入下表中的预置值，并将 M8333 切换到 ON，可更改以太网端口 2 的 IP 设置 / DNS 设置。如需使用本功能，请在 WindLDR 的“功能设置”中启用“以太网端口 2”的“启用 D8630（IP 设置 / DNS 设置切换）”。

预置值的含义如下所示。

预置值	IP 设置 / DNS 设置
0	依照功能设置。
1	启用 DHCP。
2	依照特殊数据寄存器（D8631 ~ D8650）的设置。

#### ■ D8631 ~ D8634: CPU 模块以太网端口 2 IP 地址（只写）

用于写入 Plus CPU 模块以太网端口 2 的 IP 地址。

IP 地址：设置为 aaa.bbb.ccc.ddd 时，将进行如下写入。

D8631 = aaa、D8632 = bbb、D8633 = ccc、D8634 = ddd

#### ■ D8635 ~ D8638: CPU 模块以太网端口 2 子网掩码（只写）

用于写入 Plus CPU 模块以太网端口 2 的子网掩码。

子网掩码：aaa.bbb.ccc.ddd 时，将进行如下写入。

D8635 = aaa、D8636 = bbb、D8637 = ccc、D8638 = ddd

#### ■ D8639 ~ D8642: CPU 模块以太网端口 2 默认网关（只写）

用于写入 Plus CPU 模块以太网端口 2 的默认网关。

默认网关：aaa.bbb.ccc.ddd 时，将进行如下写入。

D8639 = aaa、D8640 = bbb、D8641 = ccc、D8642 = ddd

#### ■ D8643 ~ D8646: CPU 模块以太网端口 2 首选 DNS 服务器（只写）

用于写入 Plus CPU 模块以太网端口 2 的首选 DNS 服务器。

首选 DNS 服务器：aaa.bbb.ccc.ddd 时，将进行如下写入。

D8643 = aaa、D8644 = bbb、D8645 = ccc、D8646 = ddd

#### ■ D8647 ~ D8650: CPU 模块以太网端口 2 备用 DNS 服务器（只写）

用于写入 Plus CPU 模块以太网端口 2 的备用 DNS 服务器。

备用 DNS 服务器：aaa.bbb.ccc.ddd 时，将进行如下写入。

D8647 = aaa、D8648 = bbb、D8649 = ccc、D8650 = ddd

## 2: 设备地址

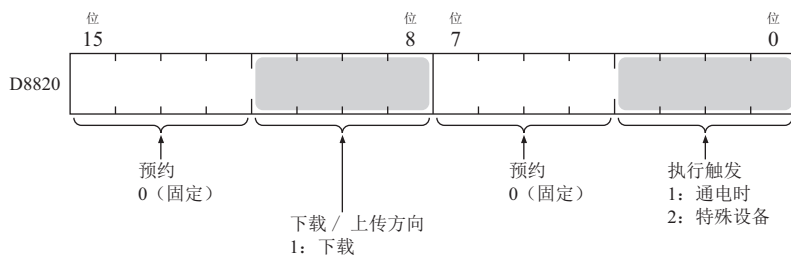
---

- **D8651 ~ D8656: CPU 模块 以太网端口 2 MAC 地址 (只读当前值)**  
Plus CPU 模块以太网端口 2 的 MAC 地址将如下所示写入到各特殊数据寄存器中。  
例) MAC 地址: AA-BB-CC-DD-EE-FF  
D8651 = AA、D8652 = BB、D8653 = CC、D8654 = DD、D8655 = EE、D8656 = FF
- **D8657 ~ D8660: CPU 模块 以太网端口 2 IP 地址 (只读当前值)**  
Plus CPU 模块以太网端口 2 的 IP 地址将如下所示写入到各特殊数据寄存器中。  
例) 本机 IP 地址: aaa.bbb.ccc.ddd  
D8657 = aaa、D8658 = bbb、D8659 = ccc、D8660 = ddd
- **D8661 ~ D8664: CPU 模块 以太网端口 2 子网掩码 (只读当前值)**  
Plus CPU 模块以太网端口 2 的子网掩码值将如下所示写入到各特殊数据寄存器中。  
例) 子网掩码: aaa.bbb.ccc.ddd  
D8661 = aaa、D8662 = bbb、D8663 = ccc、D8664 = ddd
- **D8665 ~ D8668: CPU 模块 以太网端口 2 默认网关 (只读当前值)**  
Plus CPU 模块以太网端口 2 的默认网关地址将如下所示写入到各特殊数据寄存器中。  
例) 默认网关: aaa.bbb.ccc.ddd  
D8665 = aaa、D8666 = bbb、D8667 = ccc、D8668 = ddd
- **D8669 ~ D8672: CPU 模块 以太网端口 2 首选 DNS 服务器 (只读当前值)**  
Plus CPU 模块以太网端口 2 的首选 DNS 服务器地址将如下所示写入到各特殊数据寄存器中。  
例) 首选 DNS 服务器: aaa.bbb.ccc.ddd  
D8669 = aaa、D8670 = bbb、D8671 = ccc、D8672 = ddd
- **D8673 ~ D8676: CPU 模块 以太网端口 2 备用 DNS 服务器 (只读当前值)**  
Plus CPU 模块以太网端口 2 的备用 DNS 服务器地址将如下所示写入到各特殊数据寄存器中。  
例) 备用 DNS 服务器: aaa.bbb.ccc.ddd  
D8673 = aaa、D8674 = bbb、D8675 = ccc、D8676 = ddd
- **D8759: EMAIL 指令详细错误信息 (以太网端口 1)**  
写入 EMAIL 指令 (以太网端口 1) 的详细错误信息。  
有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 12 章中的“确认 EMAIL 指令的错误详情”。
- **D8782: BACnet 运行状态**  
写入 BACnet 通信的运行状态。  
有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 15 章中的“BACnet/IP 使用的特殊设备”。
- **D8783: BACnet 错误信息**  
写入 BACnet 通信中最后发生的错误信息。  
有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 15 章中的“BACnet/IP 使用的特殊设备”。
- **D8790: EtherNet/IP 运行状态**  
写入 EtherNet/IP 通信的动作状态。  
有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 16 章中的“EtherNet/IP 通信使用的特殊设备”。
- **D8791: EtherNet/IP 错误信息**  
写入 EtherNet/IP 通信中最后发生的错误信息。  
有关详情, 请参见《FC6A 型 MICROSmart 通信手册》第 16 章中的“EtherNet/IP 通信使用的特殊设备”。



### ■ D8820: 从 SD 记忆卡下载用于服务器功能文件执行信息

执行的服务器功能用文件的下载信息。设备内的分配（位分配）如下所示。



### ■ D8821: 从 SD 记忆卡下载用于服务器功能文件执行错误信息

这是服务器功能用文件的执行错误信息。

有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

### ■ D8822: Web 服务器当前连接数（Plus CPU 模块）

这是 Web 服务器功能的当前连接数。仅支持 Plus CPU 模块。

有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

### 设备名称和符号

---

#### ■ 输入 (I)、扩展输入 (I)

将来自外部设备的 ON/OFF 信息输入至控制器的设备。

#### ■ 输出 (Q)、扩展输出 (Q)

将来自控制器的 ON/OFF 信息输出至外部设备的设备。

#### ■ 内部继电器 (M)

在控制器内部使用的位单位的设备。

#### ■ 特殊内部继电器 (M)

在控制器内部使用的位单位的设备。各位分配有特殊的功能。

#### ■ 移位寄存器 (R)

以 SFR 指令或 SFRN 指令使用的位单位的设备。根据脉冲输入对数据的位串进行移位。

#### ■ 定时器 (T)

在控制器内部使用的定时器。包括定时器位 (符号: T、单位: 位)、定时器预置值 (符号: TP、单位: 字)、定时器当前值 (符号: TC、单位: 字) 3 种设备。

可用作接通延时定时器、断开延时定时器。有关定时器 (T) 的详情, 请参见第 3-7 页上的“使用定时器或计数器作为源设备”和第 3-7 页上的“使用定时器或计数器作为目标设备”。

#### ■ 计数器 (C)

在控制器内部使用的计数器。包括计数器位 (符号: C、单位: 位)、计数器预设值 (符号: CP、单位: 字)、计数器当前值 (符号: CC、单位: 字) 3 种设备。可用作加计数器、加/减切换计数器。有关计数器 (C) 的详情, 请参见第 3-7 页上的“使用定时器或计数器作为源设备”和第 3-7 页上的“使用定时器或计数器作为目标设备”。

#### ■ 数据寄存器 (D)

在控制器内部用于写入数值数据的字单位的设备。可同时用作位单位的设备。

#### ■ 特殊数据寄存器 (D)

在控制器内部用于写入数值数据的字单位的设备。各数据寄存器分配有特殊的功能。可同时用作位单位的设备。

#### ■ 间接寄存器 (P)

用于存储间接值的 2 字单位设备, 在指定间接设备地址时使用。有关详情, 请参见第 3-13 页上的“间接指定”。

**注释:** 间接指定会使用表示基本设备及从基本设备的相对位置的间接指定寄存器 Pn, 指定源设备及目标设备。

## 类型 ID、状态一览

## ■ 类型 ID

## 扩展模块、HMI 模块

类型 ID		型号
十六进制	二进制	
0x00	0000 0000	FC6A-N16B1、FC6A-N16B4、FC6A-N16B3
0x01	0000 0001	FC6A-R161、FC6A-R164、FC6A-T16K1、FC6A-T16K4、F6A-T16P1、FC6A-T16P4、F6A-T16K3、FCA-T16P3
0x02	0000 0010	FC6A-N32B3
0x03	0000 0011	FC6A-T32K3、FC6A-T32P3
0x04	0000 0100	FC6A-N08B1、FC6A-N08B4、FC6A-N08A11、FC6A-N08A14
0x05	0000 0101	FC6A-R081、FC6A-R084、FC6A-T08K1、FC6A-T08K4、FC6A-T08P1、FC6A-T08P4
0x06	0000 0110	FC6A-M08BR1、FC6A-M08BR4
0x07	0000 0111	FC6A-M24BR1、FC6A-M24BR4
0x18	0001 1000	FC6A-PH1
0x19	0001 1001	FC6A-EXM2、FC6A-EXM24
0x1A	0001 1010	FC6A-EXM1S、FC6A-EXM1S4
0x20	0010 0000	FC6A-J2C1、FC6A-J2C4
0x21	0010 0001	FC6A-J4A1、FC6A-J4A4
0x22	0010 0010	FC6A-J8A1、FC6A-J8A4
0x23	0010 0011	FC6A-K2A1、FC6A-K2A4
0x24	0010 0100	FC6A-K4A1、FC6A-K4A4
0x25	0010 0101	FC6A-L06A1、FC6A-L06A4
0x26	0010 0110	FC6A-L03CN1、FC6A-L03CN4
0x27	0010 0111	FC6A-J4CN1、FC6A-J4CN4
0x28	0010 1000	FC6A-J8CU1、FC6A-J8CU4
0x29	0010 1001	FC6A-F2M1、FC6A-F2M4
0x2A	0010 1010	FC6A-F2MR1、FC6A-F2MR4
0x2B	0010 1011	FC6A-J4CH1Y、FC6A-J4CH4Y
0x2C	0010 1100	FC6A-EXM1M
0x2E	0010 1110	FC6A-SIF52、FC6A-SIF524
0xFF	1111 1111	未连接

## 盒

类型 ID		型号
十六进制	二进制	
0x00	0000 0000	FC6A-PJ2A
0x01	0000 0001	FC6A-PK2AV
0x02	0000 0010	FC6A-PK2AW
0x03	0000 0011	FC6A-PJ2CP
0x06	0000 0110	FC6A-PC1
0x07	0000 0111	FC6A-PC3
0x09	0000 1001	FC6A-PTS4、FC6A-PTK4
0x0A	0000 1010	FC6A-PN4
0x0C	0000 1100	FC6A-PC4
0xFF	1111 1111	未连接

## 2: 设备地址

### ■ 状态

状态		说明
十六进制	二进制	
0x00	0000 0000	正常
0x81	1000 0001	通信错误（在扩展模块、HMI 模块、盒与控制器之间进行通信时，会发生异常。）
0x82	1000 0010	检测未知设备（已连接控制器以外的设备。）
0x83	1000 0011	设备配置错误（未连接设备或已连接与用户程序配置设备不同的设备。）
0x84	1000 0100	设备写入错误（设备的动作设置失败。）
0x85	1000 0101	系统更新错误（系统更新失败。）
0x86	1000 0110	增设扩展模块组合型主机通信错误（在增设扩展模块组合型主机与增设扩展模块组合型从机之间进行通信时，会发生异常。）

## 3: 指令参考

### 简介

本章介绍控制器的指令列表与功能。

指令分为基本指令和高级指令。基本指令用于排序处理，而高级指令则用于传送、比较、逻辑运算、四则运算、位移位以及其他操作。

### 基本指令表

符号	名称	功能	参阅页面
AND	与	串联常开接点	4-4
AND LOD	块与	串联电路块	4-6
ANDN	与非	串联常闭接点	4-4
BPP	位弹出	还原临时保存的位逻辑操作的结果	4-7
BPS	位推入	临时保存位逻辑操作的结果	4-7
BRD	位读取	读取临时保存的位逻辑操作的结果	4-7
CC=	计数器比较(=)	计数器当前值的等于比较	4-23
CC>=	计数器比较(>=)	计数器当前值的大于或等于比较	4-25
CDP	加/减计数器	加/减计数器 (0 - 65,535)	4-14
CDPD	双字加/减计数器	双字加/减计数器 (0 - 4,294,967,295)	4-18
CNT	加计数器	加计数器 (0 - 65,535)	4-14
CNTD	双字加计数器	双字加计数器 (0 - 4,294,967,295)	4-18
CUD	加/减切换计数器	加/减切换计数器 (0 - 65,535)	4-14
CUDD	双字加/减切换计数器	双字加/减切换计数器 (0 - 4,294,967,295)	4-18
DC=	数据寄存器比较(=)	数据寄存器值的等于比较	4-25
DC>=	数据寄存器比较(>=)	数据寄存器值的大于或等于比较	4-25
END	结束	结束程序	4-37
JEND	跳转结束	结束跳转指令	4-35
JMP	跳转	跳转至指定的程序区域	4-35
LOD	负载	存储中间结果，并读取接点状态	4-1
LODN	取非	存储中间结果，并读取求反的接点状态	4-1
MCR	主控继电器结束	结束主控继电器	4-33
MCS	主控继电器开始	开始主控继电器	4-33
OR	或	并联常开接点	4-5
OR LOD	块或	并联电路块	4-6
ORN	或非	并联常闭接点	4-5
OUT	输出	输出位逻辑操作的结果	4-1
OUTN	求反输出	输出位逻辑操作的求反结果	4-1
RST	复位	将输出、内部继电器或移位寄存器位复位	4-3
SET	置位	将输出、内部继电器、或移位寄存器位置位	4-3
SFR	右移位寄存器	右移位寄存器	4-27
SFRN	左移位寄存器	左移位寄存器	4-27
SOTD	下降沿微分	下降沿微分输出	4-31
SOTU	上升沿微分	上升沿微分输出	4-31
TIM	100 毫秒定时器	减 100 毫秒定时器 (0 - 6,553.5s)	4-8
TIMO	100 毫秒断开延时定时器	减 100 毫秒断开延时定时器 (0 - 6,553.5s)	4-14
TMH	10 毫秒定时器	减 10 毫秒定时器 (0 - 655.35s)	4-8
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)	4-14
TML	1 秒钟定时器	减 1 秒钟定时器 (0 - 65,535s)	4-8
TMLO	1 秒钟断开延时定时器	减 1 秒钟断开延时定时器 (0 - 65,535s)	4-14

### 3: 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)	4-8
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)	4-14

## 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						3-12
传送	MOV	传送	X	X	X	X	X	5-1
	MOVN	求反传送	X	X	X	X		5-5
	IMOV	间接传送	X		X		X	5-6
	IMOVN	间接求反传送	X		X			5-8
	MOVC	传送字符						5-9
	BMOV	块传送	X					5-12
	IBMV	间接位传送	X					5-13
	IBMVN	间接位求反传送	X					5-15
	NSET	N 数据置位	X	X	X	X	X	5-16
	NRS	N 数据重复置位	X	X	X	X	X	5-17
	XCHG	交换	X		X			5-18
	TCCST	存储定时器 / 计数器当前值	X		X			5-19
	数据比较	CMP=	比较等于	X	X	X	X	X
CMP<>		比较不等于	X	X	X	X	X	6-1
CMP<		比较小于	X	X	X	X	X	6-1
CMP>		比较大于	X	X	X	X	X	6-1
CMP<=		比较小于或等于	X	X	X	X	X	6-2
CMP>=		比较大于或等于	X	X	X	X	X	6-2
ICMP>=		间隔比较大于或等于	X	X	X	X	X	6-6
LC=		接点比较等于	X	X	X	X	X	6-8
LC<>		接点比较不等于	X	X	X	X	X	6-8
LC<		接点比较小于	X	X	X	X	X	6-8
LC>		接点比较大于	X	X	X	X	X	6-8
LC<=		接点比较小于或等于	X	X	X	X	X	6-8
LC>=		接点比较大于或等于	X	X	X	X	X	6-8
二进制运算	ADD	加法	X	X	X	X	X	7-1
	SUB	减法	X	X	X	X	X	7-1
	MUL	乘法	X	X	X	X	X	7-1
	DIV	除法	X	X	X	X	X	7-1
	INC	递增	X	X	X	X		7-14
	DEC	递减	X	X	X	X		7-14
	ROOT	平方根	X		X		X	7-15
	SUM	合计 (加法)	X	X	X	X	X	7-16
		合计 (异或)	X					7-16
RNDM	随机	X					7-19	
逻辑运算	ANDW	与	X		X			8-1
	ORW	或	X		X			8-1
	XORW	异或	X		X			8-1
移位和循环	SFTL	左移						9-1
	SFTR	右移						9-3
	BCDLS	BCD 码左移			X			9-5
	WSFT	字移位	X					9-7
	ROTL	循环左移	X		X			9-8
	ROTR	循环右移	X		X			9-10

### 3: 指令参考

组	符号	名称	有效数据类型					参阅页面
			W	I	D	L	F	
数据转换	HDOB	HEX → BCD 码	X		X			10-1
	BTOH	BCD 码 → HEX	X		X			10-3
	HTOA	HEX → ASCII 码	X					10-5
	ATOH	ASCII 码 → HEX	X					10-7
	BTOA	BCD 码 → ASCII 码	X		X			10-9
	ATOB	ASCII 码 → BCD 码	X		X			10-12
	ENCO	编码						10-15
	DECO	解码						10-16
	BCNT	位计数						10-17
	ALT	交替输出						10-18
	CVDT	转换数据类型	X	X	X	X	X	10-19
	DTDV	数据分割	X					10-21
	DTCB	数据组合	X					10-22
	SWAP	数据交换	X		X			10-23
周程序	WKTIM	日历定时器						11-1
	WKTBL	周表						11-1
	WEEK	周定时器						11-3
	YEAR	年定时器						11-16
接口	MSG	信息						12-1
	DISP	显示						12-20
	DGRD	数字读取						12-22
程序分支	LABEL	标签						13-1
	LJMP	标签跳转						13-2
	LCAL	标签调用						13-4
	LRET	标签返回						13-5
	DJNZ	递减跳转非零						13-7
	DI	禁用中断						15-1
	EI	启用中断						15-1
	IOREF	I/O 刷新						14-1
	HSCRF	高速计数器刷新						14-3
	FRQRF	频率测量刷新						14-4
COMRF	通信刷新						14-5	
坐标转换	XYFS	XY 格式设置	X	X				16-1
	CVXTY	X → Y 转换	X	X				16-3
	CVYTX	Y → X 转换	X	X				16-4
	AVRG	平均化	X	X	X	X	X	17-1



组	符号	名称	有效数据类型					参阅页面
			W	I	D	L	F	
脉冲	PULS1	脉冲输出 1						18-1
	PULS2	脉冲输出 2						18-1
	PULS3	脉冲输出 3						18-1
	PULS4	脉冲输出 4						18-1
	PWM1	脉宽调制 1						18-9
	PWM2	脉宽调制 2						18-9
	PWM3	脉宽调制 3						18-9
	PWM4	脉宽调制 4						18-9
	RAMP1	台形脉冲输出 1						18-17
	RAMP2	台形脉冲输出 2						18-17
	RAMPL	直线插补控制						18-29
	ZRN1	零返回 1						18-37
	ZRN2	零返回 2						18-37
	ARAMP1	高级 RAMP1						18-47
	ARAMP2	高级 RAMP2						18-47
	ABS	绝对位置设置						18-69
	JOG	JOG 运行						18-72
PID 控制	PID	PID 控制 (兼容 FC5A)						19-1
	PIDA	PID 控制						19-4
	PIDD	微分衰减的 PID						19-28
双 / 示教定时器	DTML	1 秒双定时器						20-1
	DTIM	100 毫秒双定时器						20-1
	DTMH	10 毫秒双定时器						20-1
	DTMS	1 毫秒双定时器						20-1
	TTIM	示教定时器						20-3
三角函数	RAD	角度 → 弧度					X	21-1
	DEG	弧度 → 角度					X	21-2
	SIN	正弦					X	21-3
	COS	余弦					X	21-4
	TAN	正切					X	21-5
	ASIN	反正弦					X	21-6
	ACOS	反余弦					X	21-7
	ATAN	反正切					X	21-8
对数 / 幂	LOGE	自然对数					X	22-1
	LOG10	常用对数					X	22-2
	EXP	指数					X	22-3
	POW	幂					X	22-4
文件数据处理	FIFO	先进先出格式	X					23-1
	FIEX	执行先进	X					23-3
	FOEX	执行先出	X					23-4
	NDSRC	N 数据查找	X	X	X	X	X	23-6
时钟	TADD	时间加法						24-1
	TSUB	时间减法						24-5
	HTOS	HMS → 秒						24-11
	STOH	秒 → HMS						24-12
	HOUR	小时计量器						24-12
数据日志	DLOG	数据日志						25-1
	TRACE	数据痕迹						25-16
脚本	SCRPT	脚本	X	X	X	X	X	26-1
流量计算	SCALE	模拟量输入转换	X	X				27-1
	FLWA	模拟流量累积						27-10
	FLWP	脉冲流量累积	X		X			27-19

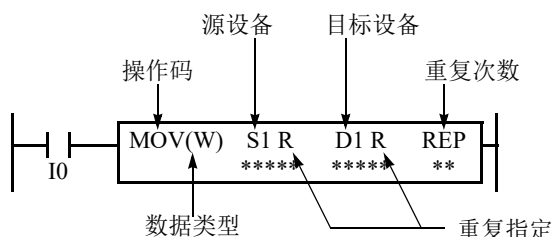
### 3: 指令参考

组	符号	名称	有效数据类型					参阅页面
			W	I	D	L	F	
用户定义宏	UMACRO	用户定义宏						28-1
用户通信	TXD	发送						*1
	ETXD	以太网发送						*1
	RXD	接收						*1
	ERXD	以太网接收						*1
PING	PING	Ping						*1
电子邮件	EMAIL	发送电子邮件						*1

\*1 有关指令的详情，请参见《FC6A 型 MICROSmart 通信手册》。

## 关于高级指令

### 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

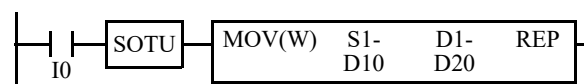
指定重复周期的数量：1 - 99。

### 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）和 LRET（标签返回）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 - 65,535。

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，请参见附录 -19 页上的“用户程序执行错误和梯形图程序执行错误”。

#### 注释：

- 发生用户程序执行错误时，不会将结果设置为目标。
- 当定时器被指定为 TCCST 指令的源设备或目标设备时，定时器的当前值将被引用。有关 TCCST 指令的详情，请参见第 5-19 页上的“TCCST（存储定时器 / 计数器当前值）”。

### 3: 指令参考

#### 高级指令的数据类型

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)<sup>\*1</sup> 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。

符号	数据类型	使用的数据寄存器数量		处理单元	数据范围	
		字	位		十进制	十六进制 <sup>*2</sup>
W	字型	1 字	16 位	无符号 16 位	0 ~ 65,535	\$0000 ~ \$FFFF
I	整型	1 字	16 位	带符号 16 位	-32,768 ~ 32,767	\$8000 ~ \$7FFF
D	双字型	2 字	32 位	无符号 32 位	0 ~ 4,294,967,295	\$00000000 ~ \$FFFFFFFF
L	长整型	2 字	32 位	带符号 32 位	-2,147,483,648 ~ 2,147,483,647	\$80000000 ~ \$7FFFFFFF

\*1 关于浮点 (F)，请参见第 3-9 页上的“浮点型数据格式”。

\*2 通过 WindLDR 设置常量时，可以十进制或十六进制进行设置。以十六进制设置时，请在起始添加“\$”。

#### 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	65,535	FFFF
	65,535	FFFF	0	0000
	131,071	(CY) FFFF	-1	(BW) FFFF
			-65,535	(BW) 0001
			-65,536	(BW) 0000
整型	65,534	(CY) 7FFE	65,534	(BW) 7FFE
	32,768	(CY) 0000	32,768	(BW) 0000
	32,767	7FFF	32,767	7FFF
	0	0000	0	0000
	-1	FFFF	-1	FFFF
	-32,767	8001	-32,767	8001
	-32,768	8000	-32,768	8000
	-32,769	(CY) FFFF	-32,769	(BW) FFFF
	-65,535	(CY) 8001	-65,535	(BW) 8001
双字型	0	00000000	4,294,967,295	FFFFFFFF
	4,294,967,295	FFFFFFFF	0	00000000
	8,589,934,591	(CY) FFFFFFFF	-1	(BW) FFFFFFFF
			-4,294,967,295	(BW) 00000001
			-4,294,967,296	(BW) 00000000
长整型	4,294,967,294	(CY) 7FFFFFFE	4,294,967,294	(BW) 7FFFFFFE
	2,147,483,648	(CY) 00000000	2,147,483,648	(BW) 00000000
	2,147,483,647	7FFFFFFF	2,147,483,647	7FFFFFFF
	0	00000000	0	00000000
	-1	FFFFFFFF	-1	FFFFFFFF
	-2,147,483,647	80000001	-2,147,483,647	80000001
	-2,147,483,648	80000000	-2,147,483,648	80000000
	-2,147,483,649	(CY) FFFFFFFF	-2,147,483,649	(BW) FFFFFFFF
	-4,294,967,295	(CY) 80000001	-4,294,967,295	(BW) 80000001



### 3: 指令参考

#### 32 位数据存储

在设备中存储 32 位数据的方法取决于控制器。

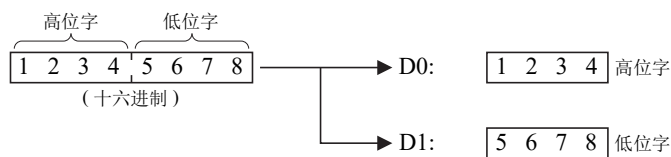
FT2J 型：高位字存储在第一个设备。

FC6A 型：32 位数据根据“功能设置”中选择的方法存储在设备中。关于适用的设备和指令，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“32 位数据存储设置”。

字设备 \*1： 型号为 FT2J 型，或在 FC6A 型“功能设置”对话框中选中“设备设置”的“从高位字开始”的情况

指定 D0 作为源设备或目标设备时，高位字存储在 D0 中，而低位字存储在 D1 中。

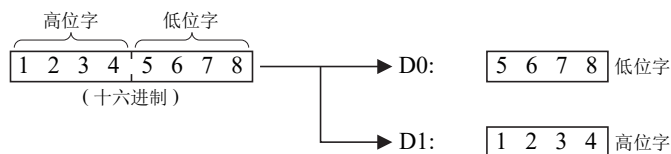
双字数据（常数）



在 FC6A 型“功能设置”对话框中选中“设备设置”的“从低位字开始”的情况

指定 D0 作为源设备或目标设备时，低位字存储在 D0 中，而高位字存储在 D1 中。

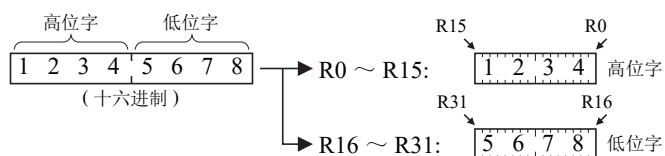
双字数据（常数）



位设备 \*1： 型号为 FT2J 型，或在 FC6A 型“功能设置”对话框中选中“设备设置”的“从高位字开始”的情况

指定 R0 作为源设备或目标设备时，高位字存储在 R0 ~ R15 中，而低位字存储在 R16 ~ R31 中。

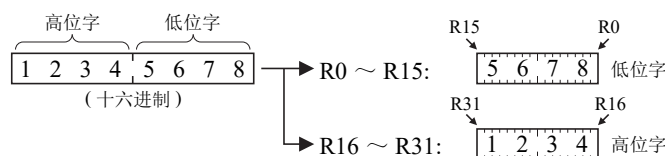
双字数据（常数）



在 FC6A 型“功能设置”对话框中选中“设备设置”的“从低位字开始”的情况

指定 R0 作为源设备或目标设备时，低位字存储在 R0 ~ R15 中，而高位字存储在 R16 ~ R31 中。

双字数据（常数）



\*1 有关位设备及字设备的详情，请参见第 2-1 页上的“设备地址”。

## 用户程序执行错误

**注释：**使用 FT2J 型时，请将“用户程序执行错误”替换为“梯形图程序执行错误”。

执行某条高级指令时，如果满足下列任何条件，则会出现用户程序执行错误。

- 高级指令的结果无效
- 高级指令中间接指定的源设备或目标设备超出有效设备范围
- 高级指令未正常运行

例如，当数据类型为浮点型（F）时，源设备中的数据不符合标准浮点格式。有关设备范围的详情，请参见第 2-1 页上的“设备地址”。

发生用户程序执行错误时，特殊内部继电器 M8004 开启并且特殊数据寄存器 D8006 中存入相应的错误代码。关于错误代码信息的详细说明，请参见附录 -19 页上的“用户程序执行错误和梯形图程序执行错误”。

发生用户程序错误时，指令运行如下：

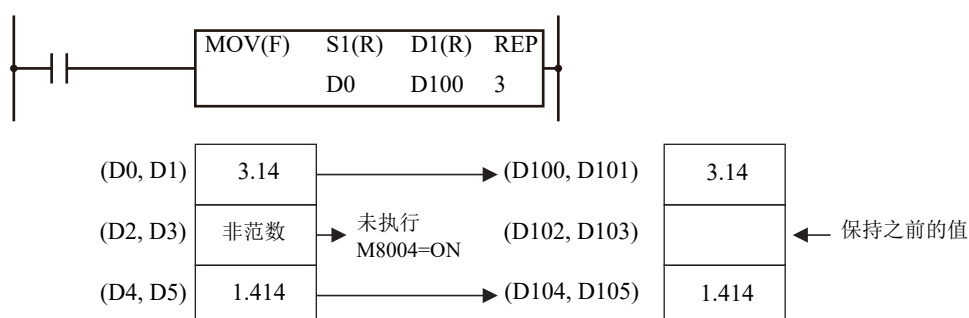
- 如果源设备的数据无效，则会取消执行高级指令，并且目标设备中的数据保持不变。
- 如果执行结果无效，则目标设备中会存入一个值。  
有关存入值的详细信息，请参见相关指令的说明。
- 如果在重复操作期间发生用户程序执行错误，则会取消该操作并执行下一重复操作。即使在接下来的重复操作中不再发生用户程序操作错误，M8004 的值仍将保持不变。

### 例如：重复操作时发生的用户程序执行错误

源数据不符合标准浮点格式时

执行第二次重复操作时，由于源数据非浮点型数据类范数，特殊内部继电器 M8004 开启。

取消第二次重复操作并执行第三次重复操作。



## 进位和借位

操作结果超出有效设备范围时，会产生一个进位 (CY) 或借位 (BW)。

根据不同的数据类型，进位或借位将在下述条件下发生：

数据类型	状态
字型	超出 0 ~ 65,535 的范围
整型	超出 -32,768 ~ 32,767 的范围
双字型	超出 0 ~ 4,294,967,295 的范围
长整型	超出 -2,147,483,648 ~ 2,147,483,647 的范围
浮点型	如果发生上溢或下溢错误 有关上溢和下溢的详情，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。

发生进位或借位错误时，特殊内部继电器 M8003（进位或借位）开启。

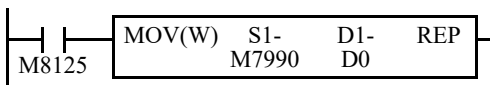
例如，如果 D0 有一个（十六进制）值 FFFF 并且使用 INC 指令进行 +1 操作，则结果为 10000（十六进制）。但是如果数据类型为字 (W)，则 D0 内将存储 0000（十六进制）并将 1 存入 M8003。

## 设备区域中断

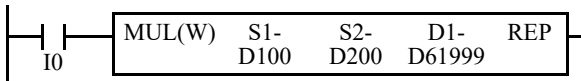
每个设备区域都是分散和不连续的，例如，从输入到输出，或从输出到内部继电器。此外，特殊内部继电器位于一个从内部继电器的单独区域中。数据寄存器、特殊数据寄存器都位于单独区域中，并且相互不连续。

### 3: 指令参考

#### 例) 使用 FC6A 型时

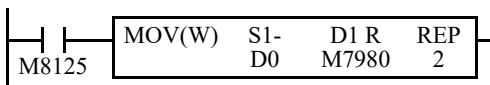


内部继电器结束于 M7997。由于 MOV（传送）指令读取 16 个内部继电器，因此最后一个内部继电器将超出有效范围，从而导致用户程序语法错误。



此程序将导致用户程序语法错误。MUL（乘法）指令的目标设备需要两个数据寄存器 D61999 和 D62000。由于 D62000 超出有效范围，因此发生用户程序语法错误。

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。



在第一个重复周期中，MOV（传送）指令将数据寄存器 D0 的数据设置为 16 个内部继电器（M7980-M7997）。第二个周期的目标是下一组 16 个内部继电器（M8000-M8017），而它们是无效的，这将导致用户程序语法错误。

有关每个高级指令的重复操作的详细信息，请参阅后面的内容。



## 执行指令时设备的指定方法

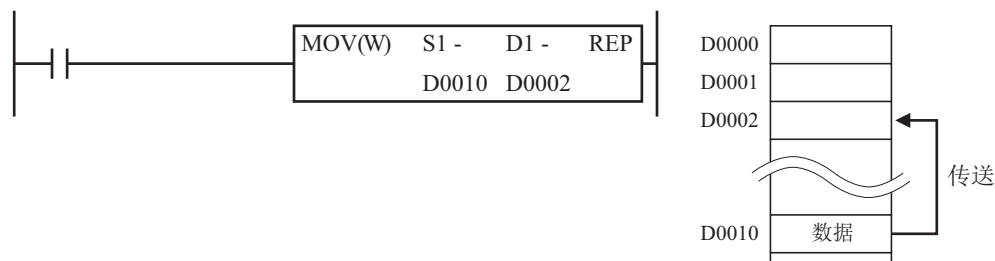
本节将对执行指令时使用的源设备及目标设备的指定方法进行介绍。指定方法包括直接指定和间接指定 2 种。

### 直接指定

直接指定可将使用的设备指定为源设备或目标设备。

#### ■直接指定的执行

例) 直接指定了 MOV(W) 指令的源设备 (D0010) 时将 D0010 的数据传送到 D0002 中。



#### ■可直接指定的设备

I	Q	M	R	T	C	D	P	常量
X	X	X	X	X	X	X	X	X

#### ■可直接指定的指令

所有基本指令和高级指令皆可直接指定。

### 间接指定

#### 间接指定的格式

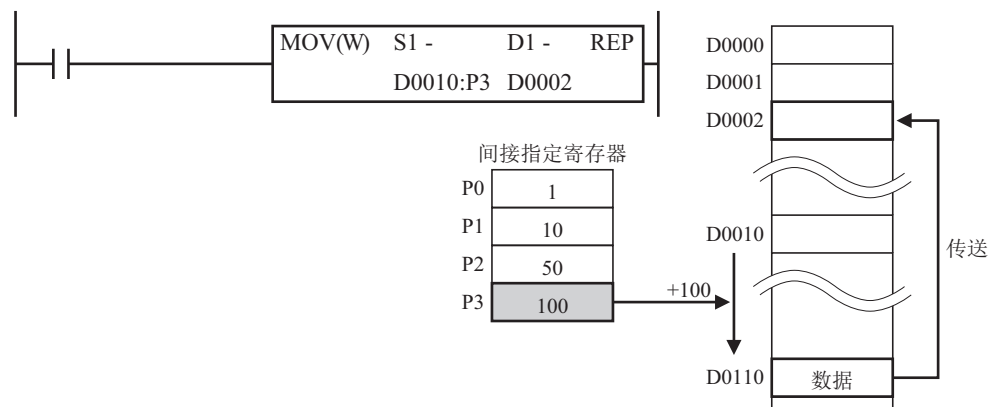
间接指定会使用表示基本设备及从基本设备的相对位置的间接指定寄存器 Pn，指定源设备及目标设备。间接指定的格式如下所示。

“基本设备”+“:”+“Pn (第 n 个间接指定寄存器)”

例) 基本设备为 D0010 且使用第 3 个间接指定寄存器时  
D0010 的后面描述 “:” 和 “P3”。  
D0010:P3

#### 间接指定的执行

例) 间接指定 (P3) 了 MOV(W) 指令的源设备 (D0010) 时将 D0010 的地址加上 P3 的数据后得出的 D (10 + (P3)) = D (10 + (100)) = D0110 数据传送到 D0002 中。



### 3: 指令参考

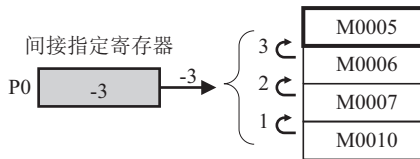
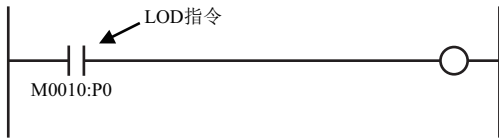
#### 间接指定寄存器

间接指定寄存器可使用 P0 ~ P15 最多 16 个，数据作为 32 位 L（长整数）型进行处理。

例) LOD M10:P0 时

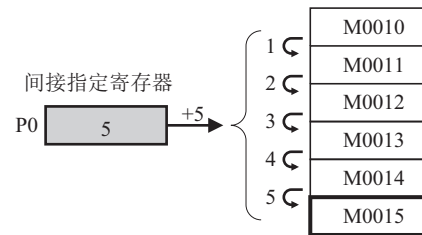
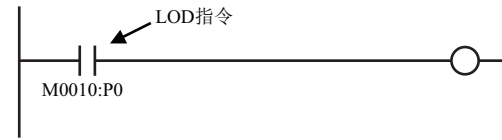
• P0=-3

表示 M0010 前 3 位处的 M0005。



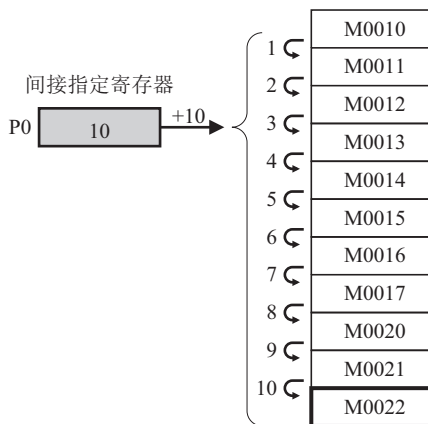
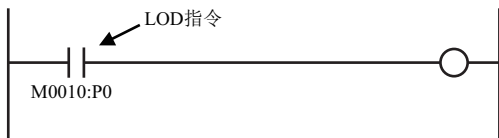
• P0=5

表示 M0010 后 5 位处的 M0015。



• P0=10

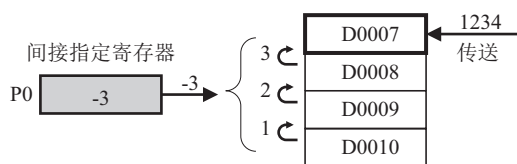
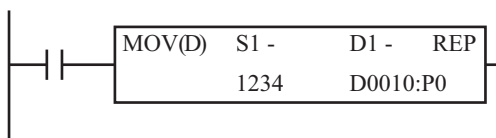
表示 M0010 后 10 位处的 M0022。



例) MOV(D) 1234 D10:P0 时

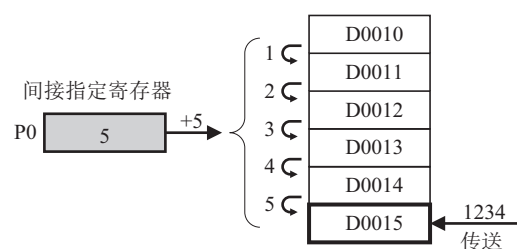
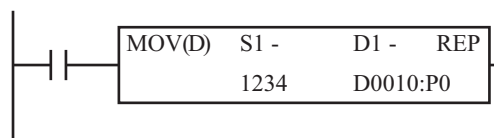
• P0=-3

表示 D0010 前 3 字处的 D0007。



• P0=5

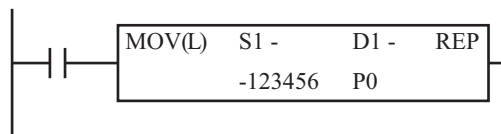
表示 D0010 后 5 字处的 D0015。



间接指定寄存器可将该数据作为 32 位长整型数据处理，可使用 MOV (L)、ADD (L)、SUB (L) 及 MUL (L) 指令将间接指定寄存器本身指定为源或目标，进行 L (长整) 数据的传送及运算。

例) MOV(L) -123456 时

- 将 -123456 传送至 P0。



注释:

- 无法在脚本中使用。
- 无法对不同的设备进行间接指定。
- 无法间接指定不同设备。  
数据寄存器在以下各范围内作为不同设备进行处理。

FT2J 型

D0000 ~ D3999、D8000 ~ D8199  
M0000 ~ M7997、M8000 ~ D8177

FC6A 型

All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块时:

D0000 ~ D7999、D8000 ~ D8499、D10000 ~ D55999  
M0000 ~ M2557、M2560 ~ M7997、M8000 ~ D8317、M10000 ~ M17497

Plus CPU 模块时:

D0000 ~ D7999、D8000 ~ D8899、D10000 ~ D61999、D70000 ~ D269999  
M0000 ~ M7997、M8000 ~ M9997、M10000 ~ M21247

- 如果以高级指令间接指定的设备超出范围，将输出用户程序执行错误 (1: 源 / 目标设备超出范围)。
- 进行间接指定时，无法以标记名称指定基本设备及间接指定寄存器。
- 无法以数据寄存器的位指定进行间接指定。
- 可指定间接指定寄存器 P 单体的指令包括 MOV(L)、ADD(L)、SUB(L)、MUL(L) 4 种。请直接指定间接指定寄存器 P 使用。

### 可间接指定的设备

I	Q	M	R	T	C	D	P	常量
X	X	X	X	X	X	X	—	—

### 可间接指定的指令

#### ■基本指令

指令	间接指定	备注
LOD、LODN	X	数据寄存器的位指定时不可
OUT、OUTN	X	数据寄存器的位指定时不可
SET、RST	X	数据寄存器的位指定时不可
AND、ANDN	X	数据寄存器的位指定时不可
OR、ORN	X	数据寄存器的位指定时不可
AND LOD	—	
OR LOD	—	
BPS、BRD、BPP	—	
TML、TIM、TMH、TIMS、TMLO、TIMO、TMHO、TIMSO	—	由于已分配独特的编号，不会进行间接指定。
CNT、CDP、CUD、CNTD、CDPD、CUDD	—	由于已分配独特的编号，不会进行间接指定。
CC=、CC>=、DC=、DC>=	—	

### 3: 指令参考

---

指令	间接指定	备注
SFR、SFRN	—	
SOTU、SOTD	—	
MCS、MCR	—	
JMP、JEND	—	
END	—	

## ■高级指令

指令	间接指定	备注
NOP	—	
MOV、MOVN	X	
IMOV、IMOVN	—	
MOVC	X	
BMOV	X	
IBMV、IBMVN	—	
NSET	X	
NRS	X	
XCHG	X	
TCCST	X	
CMP=、CMP<>、CMP<、 CMP>、CMP<=、CMP>=	X	
ICMP>=	X	
LC=、LC<>、LC<、 LC>、LC<=、LC>=	X	
ADD	X	
SUB	X	
MUL	X	
DIV	X	
INC	X	
DEC	X	
ROOT	X	
SUM	X	
RNDM	X	
ANDW	X	
ORW	X	
XORW	X	
SFTL、SFTR	X	
BCDLS	X	
WSFT	X	
ROTL、ROTR	X	
HTOB	X	
BTOH	X	
HTOA	X	
ATOH	X	
BTOA	X	
ATOB	X	
ENCO	—	
DECO	—	
BCNT	X	
ALT	X	
CVDT	X	
DTDV	X	
DTCB	X	
SWAP	X	
WKTIM	—	
WKTBL	—	
WEEK	—	
YEAR	—	

### 3: 指令参考

指令	间接指定	备注
MSG	—	
DISP	—	
DGRD	—	
LABEL	—	
LJMP	—	
LCAL	—	
LRET	—	
DJNZ	—	
IOREF	—	
HSCRF	—	
FRQRF	—	
COMRF	—	
DI	—	
EI	—	
XYFS	—	
CVXTY	—	
CVYTX	—	
AVRG	—	
PULS	—	
PWM	—	
RAMP	—	
RAMPL	—	
ZRN	—	
ARAMP	—	
ABS	—	
JOG	—	
PID	—	
PIDA	—	
PIDD	—	
DTML、DTIM、DTMH、 DTMS	—	
TTIM	—	
RAD	X	
DEG	X	
SIN	X	
COS	X	
TAN	X	
ASIN	X	
ACOS	X	
ATAN	X	
LOGE	X	
LOG10	X	
EXP	X	
POW	X	
FIFO	—	
FIEX	—	
FOEX	—	
NDSRC	—	
TADD	—	
TSUB	—	

指令	间接指定	备注
HOUR	—	
HTOS	—	
STOH	—	
DLOG	—	
TRACE	—	
SCRPT	—	脚本中不可用
SCALE	—	
FLWA	—	
FLWP	—	
UMACRO	—	可使用用户宏内的指令
TXD、RXD、ETXD、 ERXD	—	
PING	—	
EMAIL	—	





# 4: 基本指令

## 简介

本章介绍进行排序控制的基本指令。

## LOD（读取）

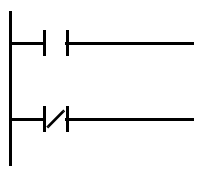
LOD 指令用于开始与 NC（常开）接点的逻辑操作。

## LODN（读取非）

LODN 指令用于开始与 NC（常闭）接点的逻辑操作。

FT2J FC6A

总共可以连续编写八个 LOD 和 / 或 LODN 指令。

梯形图	有效设备								
	指令	I	Q	M	T	C	R	D	P
	LOD	X	X	X	X	X	X	X*1	—
	LODN	X	X	X	X	X	X	X*1	—

\*1 可以指定数据寄存器中的一位。在数据寄存器编号和位位置之间加入“.”并指定。  
无法指定非保持型数据寄存器为 1 位。  
无法使用特殊数据寄存器。

有效设备范围取决于控制器类型。有关详情，请参见第 2-1 页上的“设备地址”。

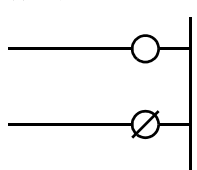
## OUT（输出）

OUT 指令用于将逻辑操作的结果输出到指定的设备。

## OUTN（求反输出）

OUTN 指令用于将逻辑操作的求反结果输出到指定的设备。

FT2J FC6A

梯形图	有效设备								
	指令	I	Q	M	T	C	R	D	P
	OUT	—	X	X	—	—	—	X*1	—
	OUTN	—	X	X	—	—	—	X*1	—

\*1 可以指定数据寄存器中的一位。在数据寄存器编号和位位置之间加入“.”并指定。  
无法指定非保持型数据寄存器为 1 位。  
无法使用特殊数据寄存器。

有效设备范围取决于控制器类型。有关详情，请参见第 2-1 页上的“设备地址”。

**注释：**关于对 OUT 和 OUTN 指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。

## 4: 基本指令

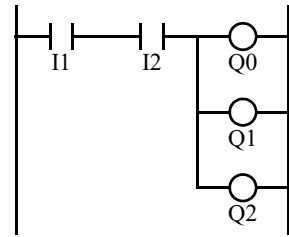
### 多个 OUT 和 OUTN

对于可以编写到一个梯形阶中的 OUT 和 OUTN 指令数，不存在限制。

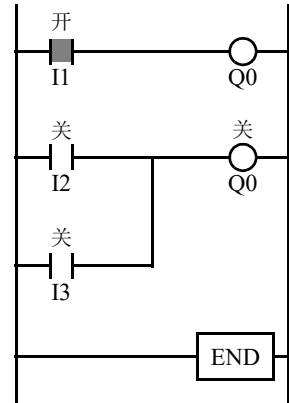
建议不要为同一个输出编号编写多个输出。但是，如果这样做，最好的做法是用 JMP/JEND 指令集或 MCS/MCR 指令集将输出分隔开。本章随后将详细介绍这些指令。

在编程中，如果在一个扫描中多次使用同一个输出编号，则最靠近 END 指令的输出将获得输出优先权。在右侧示例中，输出 Q0 被关闭。

梯形图

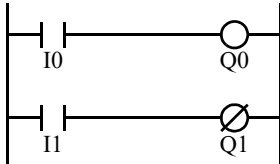


梯形图



### 示例：LOD（读取）、OUT（输出）和 NOT

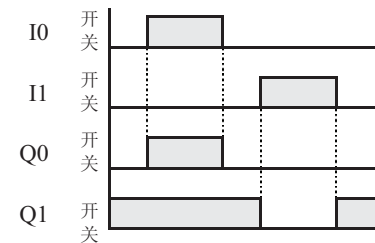
梯形图



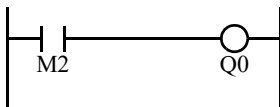
程序列表

指令	数据
LOD	I0
OUT	Q0
LOD	I1
OUTN	Q1

时序图



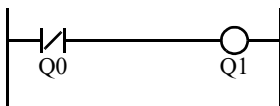
梯形图



程序列表

指令	数据
LOD	M2
OUT	Q0

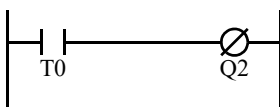
梯形图



程序列表

指令	数据
LODN	Q0
OUT	Q1

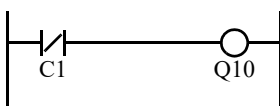
梯形图



程序列表

指令	数据
LOD	T0
OUTN	Q2

梯形图



程序列表

指令	数据
LODN	C1
OUT	Q10

## SET（置位）

SET 指令用于在运行条件为打开时，打开指定的设备。

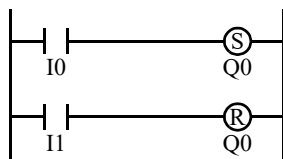
## RST（复位）

RST 指令于在运行条件为打开时，关闭指定的设备。

FT2J FC6A

SET 和 RST（复位）指令用来对输出、内部继电器和移位寄存器位进行置位（开）或复位（关）。在一个程序中，同一个输出可以置位和复位很多次。当输入已打开时，SET 和 RST 指令将在每次扫描时生效。

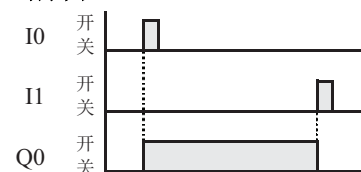
梯形图



程序列表

指令	数据
LOD	I0
SET	Q0
LOD	I1
RST	Q0

时序图



有效设备

指令	I	Q	M	T	C	R	D	P
SET	—	X	X	—	—	X	X <sup>*1</sup>	—
RST	—	X	X	—	—	X	X <sup>*1</sup>	—

\*1 可以指定数据寄存器中的一位。在数据寄存器编号和位位置之间加入“.”并指定。  
无法指定非保持型数据寄存器为 1 位。  
无法使用特殊数据寄存器。

有效设备范围取决于控制器类型。有关详情，请参见第 2-12 页上的“FC6A 型”。

**注释：**有关 SET 和 RST 指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。

## 4: 基本指令

### AND（与）

AND 指令用于串联常开接点。

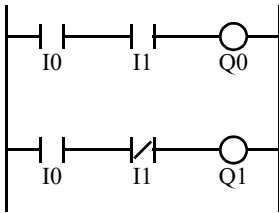
### ANDN（与非）

ANDN 指令用于串联常闭接点。

FT2J FC6A

AND 或 ANDN 指令在第一组接点之后输入。

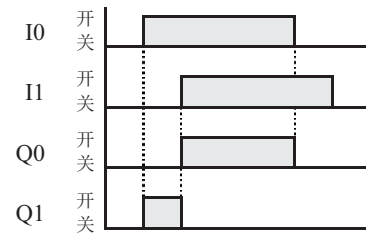
梯形图



程序列表

指令	数据
LOD	I0
AND	I1
OUT	Q0
LOD	I0
ANDN	I1
OUT	Q1

时序图



当输入 I0 和 I1 都打开时，输出 Q0 将打开。输入 I0 或 I1 都关闭时，输出 Q0 将关闭。

输入 I0 打开并且输入 I1 关闭时，输出 Q1 将打开。输入 I0 关闭或输入 I1 打开时，输出 Q1 将关闭。

### 有效设备

指令	I	Q	M	T	C	R	D	P
AND	X	X	X	X	X	X	X*1	—
ANDN	X	X	X	X	X	X	X*1	—

\*1 可以指定数据寄存器中的一位。在数据寄存器编号和位位置之间加入“.”并指定。  
无法指定非保持型数据寄存器为 1 位。  
无法使用特殊数据寄存器。

有效设备范围取决于控制器类型。有关详情，请参见第 2-12 页上的“FC6A 型”。

**OR（或）**

OR 指令用于串联常开接点。

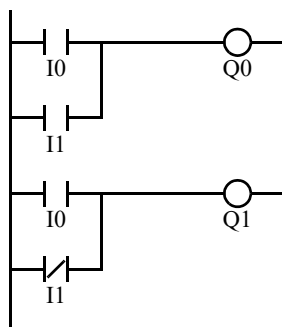
**ORN（或非）**

ORN 指令用于串联常闭接点。

FT2J FC6A

OR 或 ORN 指令在第一组接点之后输入。

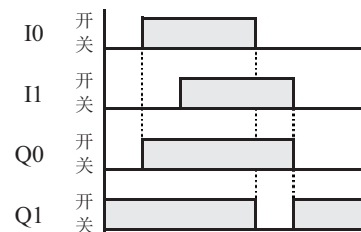
梯形图



程序列表

指令	数据
LOD	I0
OR	I1
OUT	Q0
LOD	I0
ORN	I1
OUT	Q1

时序图



输入 I0 或 I1 打开时，输出 Q0 将打开。输入 I0 和 I1 都关闭时，输出 Q0 将关闭。

输入 I0 打开或输入 I1 关闭时，输出 Q1 将打开。输入 I0 关闭并且输入 I1 打开时，输出 Q1 将关闭。

**有效设备**

指令	I	Q	M	T	C	R	D	P
OR ORN	X	X	X	X	X	X	X*1	—

\*1 可以指定数据寄存器中的一位。在数据寄存器编号和位位置之间加入“.”并指定。

无法指定非保持型数据寄存器为 1 位。

无法使用特殊数据寄存器。

有效设备范围取决于控制器类型。有关详情，请参见第 2-12 页上的“FC6A 型”。

## 4: 基本指令

### AND LOD (块与)

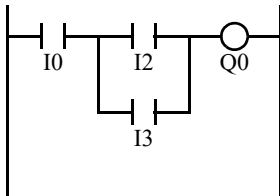
AND LOD 指令用于串联两个或更多个以 LOD 指令开始的电路。

FT2J FC6A

AND LOD 指令等同于梯形图中的“节点”。

使用 WindLDR 时，用户不需要编写 AND LOD 指令。编译梯形图时，下面显示的梯形图中的电路将转换为 AND LOD。

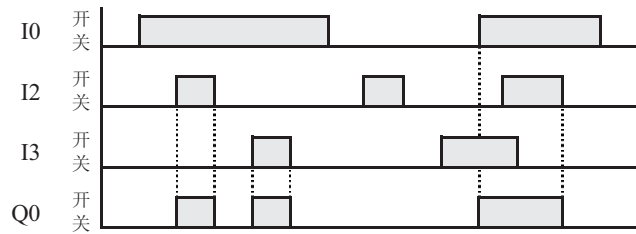
梯形图



程序列表

指令	数据
LOD	I0
LOD	I2
OR	I3
ANDLOD	
OUT	Q0

时序图



输入 I0 打开并且输入 I2 或 I3 打开时，输出 Q0 将打开。  
输入 I0 关闭或者输入 I2 和 I3 都关闭时，输出 Q0 将关闭。

### OR LOD (块或)

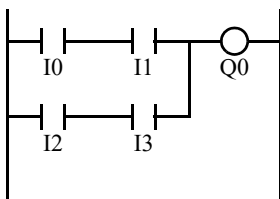
OR LOD 指令用于并联两个或更多个以 LOD 指令开始的电路。

FT2J FC6A

OR LOD 指令等同于梯形图中的“节点”。

使用 WindLDR 时，用户不需要编写 OR LOD 指令。编译梯形图时，下面显示的梯形图中的电路将转换为 OR LOD。

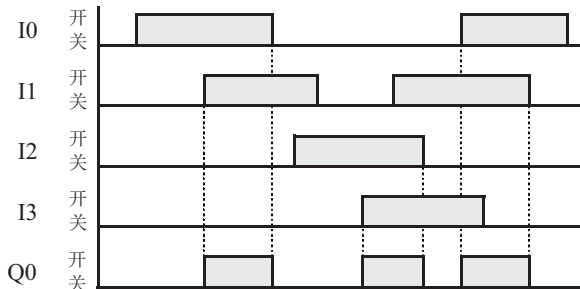
梯形图



程序列表

指令	数据
LOD	I0
AND	I1
LOD	I2
AND	I3
ORLOD	
OUT	Q0

时序图



输入 I0 和 I1 都打开，或者输入 I2 和 I3 都打开时，输出 Q0 将打开。  
输入 I0 或 I1 关闭，并且输入 I2 或 I3 关闭时，输出 Q0 将关闭。

## BPS（位推入）

BPS 指令用于临时保存逻辑操作的结果。

## BRD（位读取）

BRD 指令用于读取临时保存的位逻辑操作的结果。

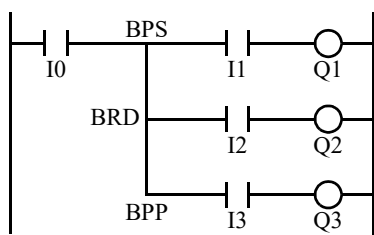
## BPP（位弹出）

BPP 指令用于还原临时保存的位逻辑操作的结果。

FT2J FC6A

使用 WindLDR 时，用户不需要编写 BPS、BRD 和 BPP 指令。编译梯形图时，下面显示的梯形图中的电路将转换为 BPS、BRD 和 BPP。

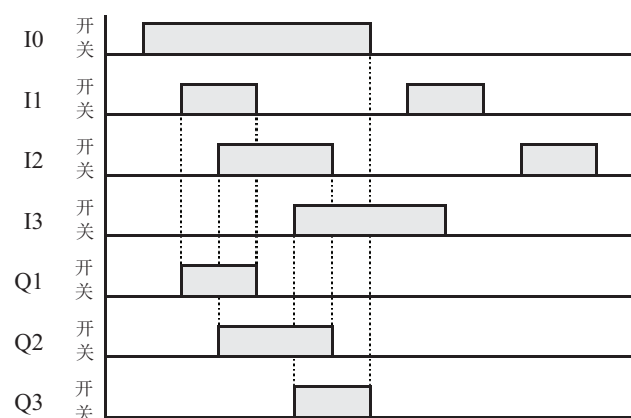
梯形图



程序列表

指令	数据
LOD	I0
BPS	
AND	I1
OUT	Q1
BRD	
AND	I2
OUT	Q2
BPP	
AND	I3
OUT	Q3

时序图



输入 I0 和 I1 都打开时，输出 Q1 将打开。

输入 I0 和 I2 都打开时，输出 Q2 将打开。

输入 I0 和 I3 都打开时，输出 Q3 将打开。

## 4: 基本指令

### TML (1-s 定时器)

TML 是以 1s 为时间间隔的定时器。

### TIM (100-ms 定时器)

TIM 是以 100ms 为时间间隔的定时器。

### TMH (10-ms 定时器)

TMH 是以 10ms 为时间间隔的定时器。

### TMS (1-ms 定时器)

TMS 是以 1ms 为时间间隔的断电延时定时器。

FT2J FC6A

每个定时器必须分配一个唯一的编号。

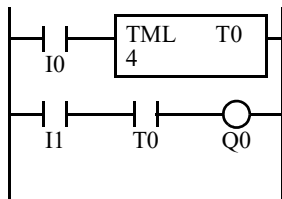
符号	设备地址	范围	递增	预置值
TML	设备: 定时器 (T)	0 ~ 65,535s	1s	常量: 0 ~ 65,535
TIM	设备地址: 有效设备范围取决于控制器类型。	0 ~ 6,553.5s	100ms	数据寄存器: 有效设备范围取决于控制器类型。
TMH		0 ~ 655.35s	10ms	
TMS		0 ~ 65.535s	1ms	

有关详情, 请参见第 2-12 页上的“FC6A 型”。

若为间接指定, 则以数据寄存器编号进行指定, 并在 0 ~ 65,535 的范围内指定数据寄存器的值。

### TML (1-s 定时器)

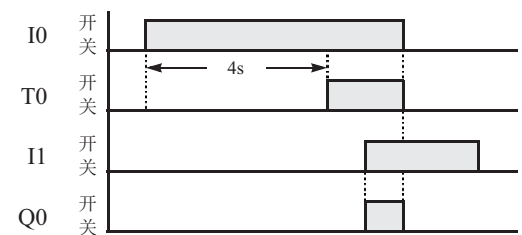
梯形图 (TML)



程序列表

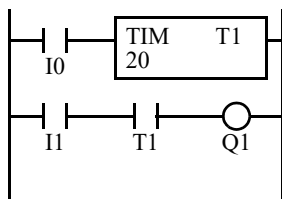
指令	数据
LOD	I0
TML	T0 4
LOD	I1
AND	T0
OUT	Q0

时序图



### TIM (100-ms 定时器)

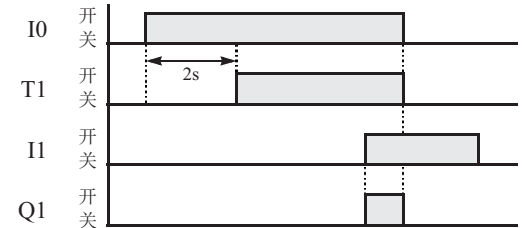
梯形图 (TIM)



程序列表

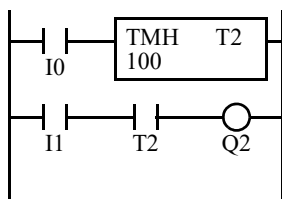
指令	数据
LOD	I0
TIM	T1 20
LOD	I1
AND	T1
OUT	Q1

时序图



### TMH (10-ms 定时器)

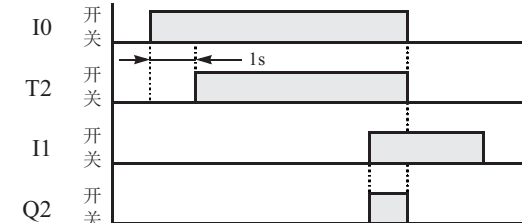
梯形图 (TMH)



程序列表

指令	数据
LOD	I0
TMH	T2 100
LOD	I1
AND	T2
OUT	Q2

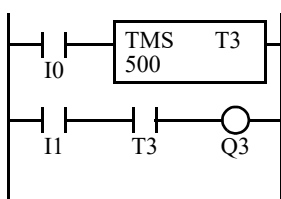
时序图





## TMS (1-ms 定时器)

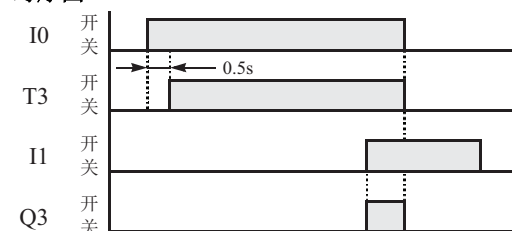
梯形图 (TMS)



程序列表

指令	数据
LOD	I0
TMS	T3 500
LOD	I1
AND	T3
OUT	Q3

时序图



从预置值开始的倒数是在紧靠定时器输入之前的操作结果为开时启动的。

当前值（计时值）达到 0 时，定时器输出将打开。

定时器输入为关时，当前值将恢复到预置值。

## 注释:

- 在 WindLDR 中，在定时器指令之后可对一些指令进行串联编程，因此不能像上面的梯形图中显示的那样自动连接到右母线。有关详情，请参见第 4-12 页上的“定时器电路”。



- 当定时器设备被指定为高级指令的源设备时，定时器的当前值或剩余时间将作为源数据被读取。当定时器设备被指定为高级指令的目标设备时，高级指令的结果将被设置为定时器的预设值。
- 有关编程和定时器指令错误的详细信息，请参见以下几页。  
第 4-13 页上的“定时器计数误差”  
第 4-13 页上的“超时输出误差”  
第 4-38 页上的“梯形图编程限制”
- 在使用程序分支指令创建的子程序和分支目标梯形图程序中使用定时器指令时，请注意以下几点。
  - 要初始化定时器指令，请在跳转后的一个或多个扫描周期内关闭定时器指令的输入。
  - 从开始计数到计时结束，每次扫描都必须执行定时器指令。
 有关子程序和分支指令的详细信息，请参见第 13-1 页上的“程序分支指令”。
- 在梯形图程序中，定时器设备地址不能重复使用。
- 定时器指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 定时器指令的预设值可通过外部设备（可编程显示器等）、WindLDR 或控制器操作进行更改。断电后，已更改的预设值将被清除，并加载原始预设值。但是，即使电源关闭，FC6A 型也可以保留更改的预设值。保留更改预设值的设置方法如下。

## 使用 WindLDR

从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”，在“PLC”组中单击“状态”，然后单击“TIM/CNT 更改状态”的“确定”。预设值一经确认，即使清除也无法重置为更改前的预设值。

## 使用 HMI 模块

有关 HMI 模块的详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”。

有关更改、确认和清除预设值时的数据移动，请参见第 4-22 页上的“更改、确认和清除定时器和计数器的预置值”。

- 如果在倒计时过程中更改了定时器预设值，定时器将保持不变，直至关闭定时器指令输入。
- 如果定时器预设值变为 0，则定时器停止运行，定时器输出立即打开。

## 4: 基本指令

### TMLO (1-s 断开延时定时器)

TMLO 是以 1s 为时间间隔的断开延时定时器。

### TIMO (100-ms 断开延时定时器)

TIMO 是以 100ms 为时间间隔的断开延时定时器。

### TMHO (10-ms 断开延时定时器)

TMHO 是以 10ms 为时间间隔的断开延时定时器。

### TMSO (1-ms 断开延时定时器)

TMSO 是以 1ms 为时间间隔的断开延时定时器。

FT2J FC6A

每个定时器必须分配一个唯一的编号。

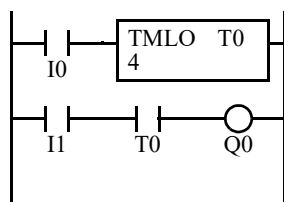
符号	设备地址	范围	递增	预置值
TMLO	设备: 定时器 (T)	0 ~ 65,535s	1s	常量: 0 ~ 65,535
TIMO	设备地址: 有效设备范围取决于控制器类型。	0 ~ 6,553.5s	100ms	数据寄存器: 有效设备范围取决于控制器类型。
TMHO		0 ~ 655.35s	10ms	
TMSO		0 ~ 65.535s	1ms	

有关详情, 请参见第 2-12 页上的“FC6A 型”。

若为间接指定, 则以数据寄存器编号进行指定, 并在 0 ~ 65,535 的范围内指定数据寄存器的值。

### TMLO (1-s 断开延时定时器)

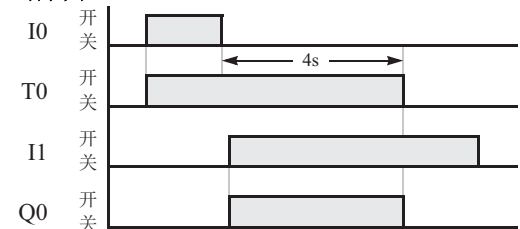
梯形图 (TMLO)



程序列表

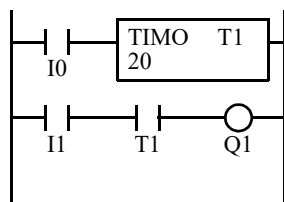
指令	数据
LOD	I0
TMLO	T0
	4
LOD	I1
AND	T0
OUT	Q0

时序图



### TIMO (100-ms 断开延时定时器)

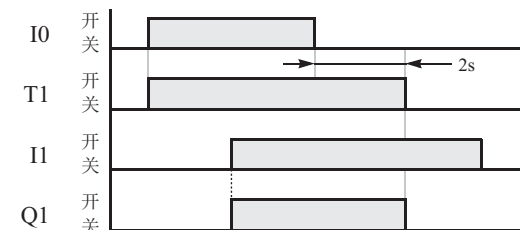
梯形图 (TIMO)



程序列表

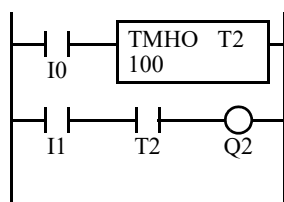
指令	数据
LOD	I0
TIMO	T1
	20
LOD	I1
AND	T1
OUT	Q1

时序图



### TMHO (10-ms 断开延时定时器)

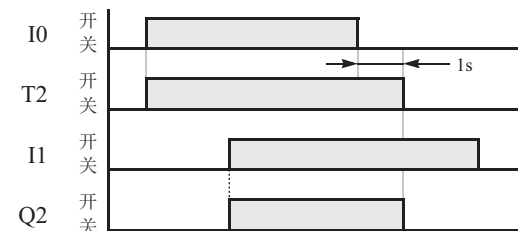
梯形图(TMHO)



程序列表

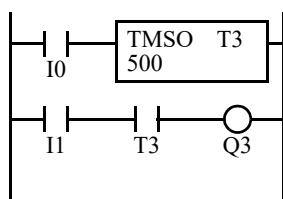
指令	数据
LOD	I0
TMHO	T2
	100
LOD	I1
AND	T2
OUT	Q2

时序图



## TMSO (1-ms 断开延时定时器)

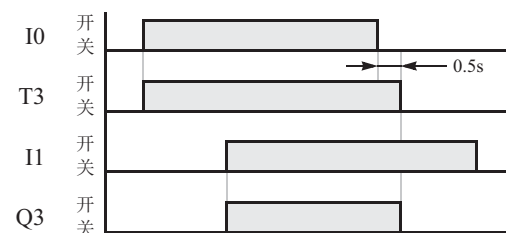
梯形图(TMSO)



程序列表

指令	数据
LOD	I0
TMSO	T3 500
LOD	I1
AND	T3
OUT	Q3

时序图



## 注释:

- 在 WindLDR 中, 在断开延时定时器指令之后可对一些指令进行串联编程, 因此不能像上面的梯形图中显示的那样自动连接到右母线。有关详情, 请参见第 4-12 页上的“定时器电路”。
- 当定时器设备被指定为高级指令的源设备时, 定时器的当前值或剩余时间将作为源数据被读取。当定时器设备被指定为高级指令的目标设备时, 高级指令的结果将被设置为定时器的预设值。
- 有关编程和断开延时定时器指令错误的详细信息, 请参见以下几页。  
第 4-13 页上的“定时器计数误差”  
第 4-13 页上的“超时输出误差”  
第 4-38 页上的“梯形图编程限制”
- 在使用程序分支指令创建的子程序和分支目标梯形图程序中使用断开延时定时器时, 请注意以下几点。
  - 要初始化断开延时定时器指令, 请在跳转后的一个或多个扫描周期内打开断开延时定时器指令的输入。
  - 从开始计数到计时结束, 每次扫描都必须执行断开延时定时器指令。
 有关子程序和分支指令的详细信息, 请参见第 13-1 页上的“程序分支指令”。
- 在梯形图程序中, 断开延时定时器设备地址不能重复使用。
- 断开延时定时器指令不能在中断程序中使用。如果在中断程序中使用, 将导致用户程序执行错误, 将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。
- 断开延时定时器指令的预设值可通过外部设备 (可编程显示器等)、WindLDR 或控制器操作进行更改。断电后, 已更改的预设值将被清除, 并加载原始预设值。但是, 即使电源关闭, FC6A 型也可以保留更改的预设值。保留更改预设值的设置方法如下。

## 使用 WindLDR

从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”, 在“PLC”组中单击“状态”, 然后单击“TIM/CNT 更改状态”的“确定”。预设值一经确认, 即使清除也无法重置为更改前的预设值。

## 使用 HMI 模块

有关 HMI 模块的详情, 请参见《FC6A 型 MICROsmart 用户手册》第 7 章“HMI 功能”。

有关更改、确认和清除预设值时的数据移动, 请参见第 4-22 页上的“更改、确认和清除定时器和计数器的预置值”。

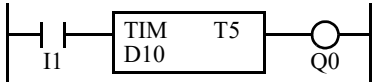
- 如果在倒计时过程中更改了断开延时定时器的预设值, 定时器将保持不变, 直到断开延时定时器指令的输入被打开。
- 如果断开延时定时器预设值变为 0, 则定时器停止运行, 断开延时定时器输出立即打开。

## 4: 基本指令

### 定时器电路

可以使用数据寄存器来指定预置值 0 ~ 65,535；然后数据寄存器的数据将成为预置值。紧接在 TML、TIM、TMH 或 TMS 指令之后，可以编写 OUT、OUTN、SET、RST、TML、TIM、TMH、TMS、TMLO、TIMO、TMHO 或 TMSO 指令。

梯形图



程序列表

指令	数据
LOD	I1
TIM	T5 D10
OUT	Q0

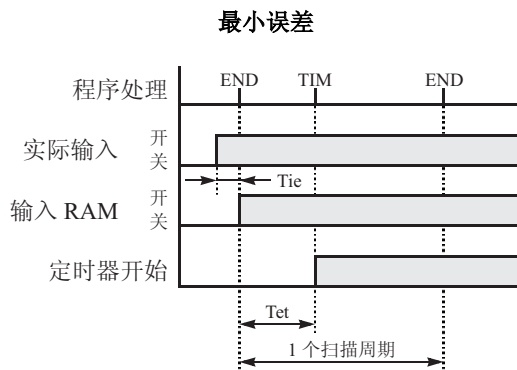
**注释：**有关定时器指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。

### 定时器精确度

由于软件设置，定时器精确度取决于三个因素：定时器输入误差、定时器计数误差和超时输出误差。这些误差不是固定不变的，而是会随着用户程序和其他原因而变化。

#### 定时器输入误差

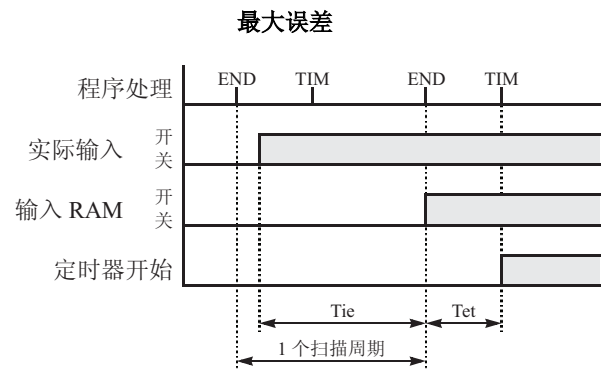
输入状态将在结束处理时被读取，并存储在输入 RAM 中。定时器输入根据从关闭状态变为打开时的扫描状态而产生误差。但是，无论是标准输入还是捕捉输入产生相同的误差。下面显示的定时器输入误差不包括由硬件导致的输入延迟。



当输入在紧靠结束处理之前打开时， $T_{ie}$  几乎为 0。然后，定时器输入误差只是  $T_{et}$ （延时误差）并且是它的最小值。

$T_{ie}$ : 从输入打开到结束处理之间的时间

$T_{et}$ : 从结束处理到定时器指令执行之间的时间



当输入在紧靠结束处理之后打开时， $T_{ie}$  几乎等于一个扫描周期。然后，定时器输入误差是  $T_{ie} + T_{et}$  = 一个扫描周期 +  $T_{et}$ （误差时），并且是它的最大值。

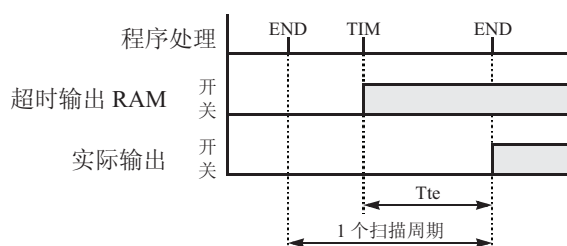
### 定时器计数误差

每个定时器指令操作均单独地基于异步 16 位参考定时器。因此，误差是否发生将取决于执行定时器指令时异步 16 位定时器的状态。

误差		TML TMLO	TIM TIMO	TMH TMHO	TMS TMSO
最大	前进误差	1,000ms	100ms	10ms	1ms
	延时误差	1 个扫描周期	1 个扫描周期	1 个扫描周期	1 个扫描周期

### 超时输出误差

当处理 END 指令时，输出 RAM 的状态将设置为实际输出。如果开启已关闭的超时输出，则会出现错误，具体情况取决于用户程序中编写定时器指令的位置。下面显示的超时输出错误不包括由硬件导致的输出延迟。



超时输出误差等于  $T_{te}$ （延时误差），并且可以是 0 和一个扫描周期之间的值。

$$0 < T_{te} < 1 \text{ 次扫描周期}$$

$T_{te}$ : 从定时器指令执行到结束处理之间的时间

### 最大和最小误差

误差		定时器输入误差	定时器计数误差	超时输出误差	最大误差
最小	前进误差	0（注释）	0	0（注释）	0
	延时误差	$T_{et} + T_{ie}$	0	$T_{te}$	0
最大	前进误差	0（注释）	增量	0（注释）	增量 - ( $T_{et} + T_{te}$ )
	延时误差	1 个扫描周期 + $T_{et}$ (1 个扫描周期)	1 个扫描周期	$T_{te}$ (1 个扫描周期)	2 个扫描周期 + ( $T_{et} + T_{te}$ )

**注释：** 定时器输入和超时输出不会发生前进误差。

$T_{et} + T_{te} = 1$  个扫描周期

增量是 1s (TML)、100ms (TIM)、10ms (TMH) 或 1ms (TMS)。

最大前进误差是：增量 - 1 个扫描周期

最大延时误差是：3 个扫描周期

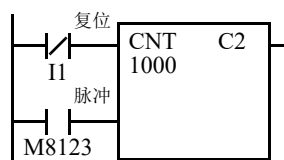
上面显示的定时器输入误差和超时输出误差不包括由硬件导致的输入响应时间（延时误差）和输出响应时间（延时误差）。

### 电源故障内存保护

定时器 TML、TIM、TMH 和 TMS 没有电源故障保护。要提供具有此保护的定时器，请使用计数器指令和特殊内部继电器 M8121（1 秒时钟）、M8122（100 毫秒时钟）或 M8123（10 毫秒时钟）。

#### 梯形图

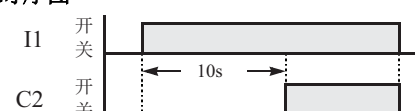
(10秒定时器)



#### 程序列表

指令	数据
LODN	I1
LOD	M8123
CNT	C2 1000

#### 时序图



**注释：** 请将此程序中使用的计数器 C2 指定为保持类型计数器。请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“内存备份”。

## 4: 基本指令

### CNT（加计数器）

CNT 是加计数器。

### CDP（加 / 减计数器）

CDP 是加 / 减计数器。

### CUD（加 / 减切换计数器）

CUD 是加 / 减切换计数器。

FT2J FC6A

每个计数器都必须分配一个唯一的编号。

符号	设备地址	预置值
CNT	设备: 计数器 (C)	常量: 0 ~ 65,535
CDP	设备地址: 有效设备范围取决于控制器类型。	数据寄存器: 有效设备范围取决于控制器类型。
CUD		

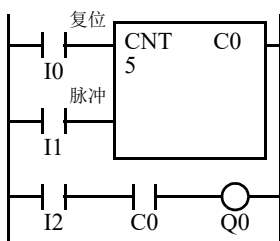
有关详情，请参见第 2-12 页上的“FC6A 型”。

若为间接指定，则以数据寄存器编号进行指定，并在 0 ~ 65,535 的范围内指定数据寄存器的值。

### CNT（加计数器）

编写计数器指令时，需要使用两个地址。加法（递增）计数器的电路必须按以下顺序编写：复位输入、脉冲输入、CNT 指令和计数器数字，后跟计数器预置值（0 ~ 65,535）。

梯形图



程序列表

指令	数据
LOD	I0
LOD	I1
CNT	C0
	5
LOD	I2
AND	C0
OUT	Q0

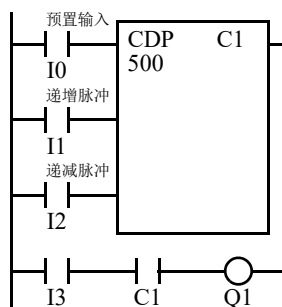
- 复位输入打开时，将忽略所有脉冲输入。
- 复位输入已关闭时，计数器将对脉冲输入的前沿进行计数，并将它们与预置值进行比较。
- 当复位输入从关被更改为开时，当前值将复位。
- 当前值达到预置值时，计数器将打开输出。输出将保持在打开状态，直到复位输入打开。
- 即使在当前值达到预设值后打开脉冲输入，当前值也不会改变。
- 更改 CNT 指令的预设值或当前值时的操作如下。

条件	计数器输出
计数器完成计数输出时，会更改当前值或预置值。	计数器保持计数输出状态。
在计数器计数输出前，会将当前值更改为比预置值大的值。	计数器输出开启。
预置值更改为 0。	无论当前值如何，计数器输出开启。
复位输入开启时，预置值更改为 0。	计数器输出未开启。

## CDP（加 / 减计数器）

加 / 减计数器 CDP 有递增和递减脉冲输入，因此需要三个输入。加 / 减计数器的电路必须按以下顺序编写：预置输入、递增脉冲输入、递减脉冲输入、CDP 指令和计数器数字，后跟计数器预置值（0 ~ 65,535）。

梯形图

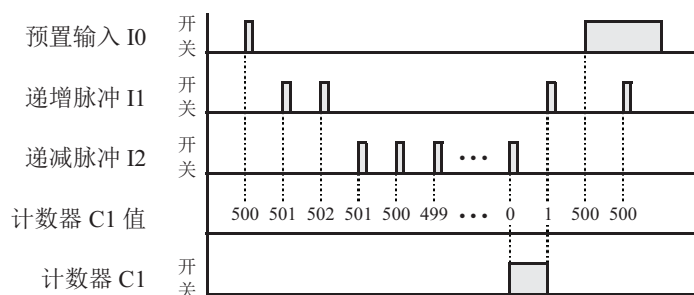


程序列表

指令	数据
LOD	I0
LOD	I1
LOD	I2
CDP	C1
	500
LOD	I3
AND	C1
OUT	Q1

- 预置输入必须在一开始就打开，以便让当前值恢复到预置值。
- 计数有可能开始之前，必须关闭预置输入。
- 当前值达到 0（减计数）之后，它将在下一次减计数时更改为 65,535。
- 当前值达到 65,535（加计数）之后，它将在下一次加计数时更改为 0。
- 计数器输出只在当前值为 0 时才会打开。
- 递增脉冲和递减脉冲同时打开时，没有任何脉冲被计数。

时序图

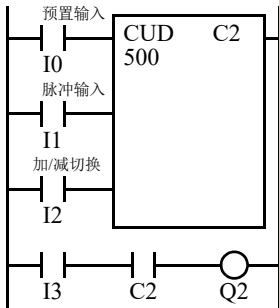


## 4: 基本指令

### CUD（加 / 减切换计数器）

加 / 减切换计数器 CUD 有一个可以切换加 / 减门的选择输入，因此需要三个输入。加 / 减切换计数器的电路必须按以下顺序编写：预置输入、脉冲输入、加 / 减切换输入、CUD 指令和计数器数字，后跟计数器预置值（0 ~ 65,535）。

梯形图

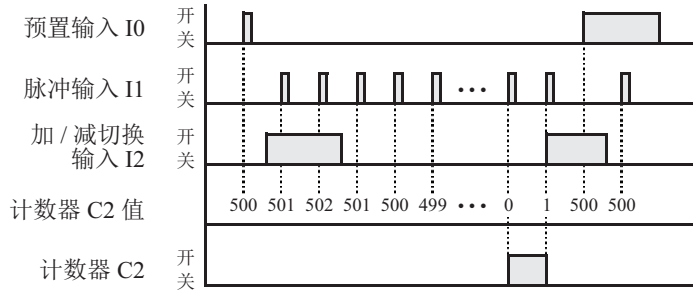


程序列表

指令	数据
LOD	I0
LOD	I1
LOD	I2
CUD	C2
	500
LOD	I3
AND	C2
OUT	Q2

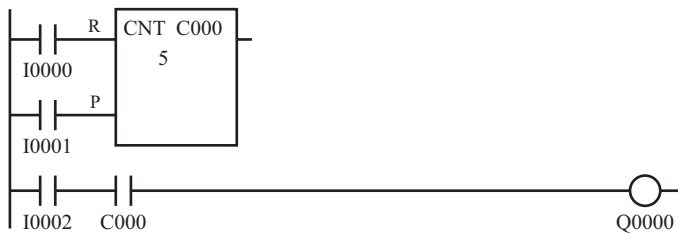
- 预置输入必须在一开始就打开，以便让当前值恢复到预置值。
- 当加 / 减切换输入打开时，将切换为加计数模式。
- 当加 / 减切换输入关闭时，将切换为减计数模式。
- 当前值达到 0（减计数）之后，它将在下一次减计数时更改为 65,535。
- 当前值达到 65,535（加计数）之后，它将在下一次加计数时更改为 0。
- 计数器输出只在当前值为 0 时才会打开。

时序图



**注释：**

- 在 WindLDR 中，在计数器指令之后可对一些指令进行串联编程，因此不能像上面的梯形图中显示的那样自动连接到右母线。有关详情，请参见第 4-17 页上的“”。



- 当计数器设备被指定为高级指令的源设备时，将读取当前值作为源数据。当计数器设备被指定为高级指令的目标设备时，高级指令的结果将被设置为计数器的预设值。
- 有关计数器指令的梯形图编程限制，请参见第 4-38 页上的“梯形图编程限制”。
- 在梯形图程序中，计数器设备地址不能使用多次。
- 计数器指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。



- 计数器指令的预设值可通过外部设备（可编程显示器等）、WindLDR 或控制器操作进行更改。断电后，已更改的预设值将被清除，并加载原始预设值。但是，即使电源关闭，FC6A 型也可以保留更改的预设值。保留更改预设值的设置方法如下。

### 使用 WindLDR

从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”，在“PLC”组中单击“状态”，然后单击“TIM/CNT 更改状态”的“确定”。预设值一经确认，即使清除也无法重置为更改前的预设值。

### 使用 HMI 模块

有关 HMI 模块的详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”。

有关更改、确认和清除预设值时的数据移动，请参见第 4-22 页上的“更改、确认和清除定时器和计数器的预置值”。

## 4: 基本指令

### CNTD (双字加计数器)

CNTD 是双字加 (递增) 计数器。

### CDPD (双字加 / 减计数器)

CDPD 是双字加 / 减计数器。

### CUDD (双字加 / 减切换计数器)

CUDD 是双字加 / 减切换计数器。

FT2J FC6A

每个双字计数器分配 2 个连续的设备，其编号范围为。在一个用户程序中，相同编号的计数器不能重复使用在不同的计数器指令。

符号	设备地址	预置值
CNT	设备: 计数器 (C)	常量: 0 ~ 4,294,967,295
CDP	设备地址: 有效设备范围取决于控制器类型。	数据寄存器: 有效设备范围取决于控制器类型。
CUD		

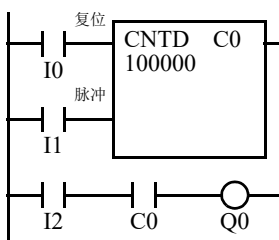
有关详情，请参见第 2-12 页上的“FC6A 型”。

预置值可以是 0 ~ 4,294,967,295，并使用十进制常量或数据寄存器来指定。当一个数据寄存器设置预置值时，将使用两个连续的数据寄存器。

### CNTD (双字加计数器)

编写计数器指令时，需要使用两个地址。加法 (递增) 计数器的电路必须按以下顺序编写：复位输入、脉冲输入、CNTD 指令和计数器数字，后跟计数器预置值 (0 ~ 4,294,967,295)。

梯形图

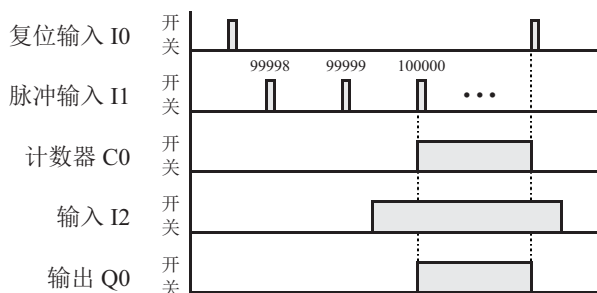


程序列表

指令	数据
LOD	I0
LOD	I1
CNTD	C0
	100000
LOD	I2
AND	C0
OUT	Q0

- 复位输入已关闭时，计数器将对脉冲输入的前沿进行计数，并将它们与预置值进行比较。
- 当前值达到预置值时，计数器将打开输出。输出将保持在打开状态，直到复位输入打开。
- 当复位输入从关被更改为开时，当前值将复位。
- 复位输入打开时，将忽略所有脉冲输入。
- 计数有可能开始之前，复位输入必须关闭。
- 即使在前值达到预设值后打开脉冲输入，当前值也不会改变。

时序图



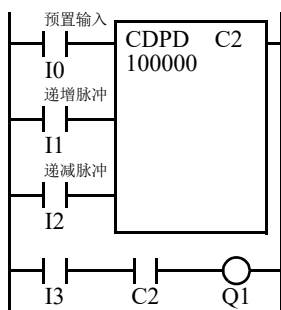
- 更改 CNT 指令的预设值或当前值时的操作如下。

条件	计数器输出
计数器完成计数输出时，会更改当前值或预置值。	计数器保持计数输出状态。
在计数器计数输出前，会将当前值更改为比预置值大的值。	计数器输出开启。
预置值更改为 0。	无论当前值如何，计数器输出开启。
复位输入开启时，预置值更改为 0。	计数器输出未开启。

### CDPD (双字加 / 减计数器)

双字加 / 减计数器 CDPD 有递增和递减脉冲输入，因此需要三个输入。双字加 / 减计数器的电路必须按以下顺序编写：预置输入、递增脉冲输入、递减脉冲输入、CDPD 指令和计数器数字，后跟计数器预置值（0 ~ 4,294,967,295）。

梯形图

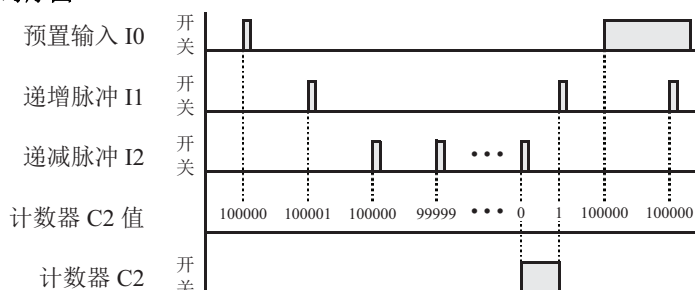


程序列表

指令	数据
LOD	I0
LOD	I1
LOD	I2
CDPD	C2
	100000
LOD	I3
AND	C2
OUT	Q1

- 预置输入必须在一开始就打开，以便让当前值恢复到预置值。
- 计数有可能开始之前，必须关闭预置输入。
- 递增脉冲和递减脉冲同时打开时，没有任何脉冲被计数。
- 当前值达到 0(减计数)之后，它将在下一次减计数时更改为 4,294,967,295。
- 当前值达到 4,294,967,295(加计数)之后，它将在下一次加计数时更改为 0。
- 计数器输出只在当前值为 0 时才会打开。

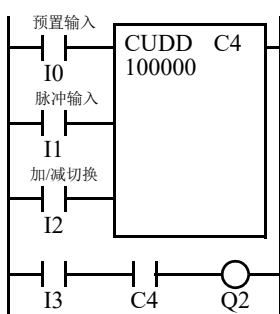
时序图



### CUDD (双字加 / 减切换计数器)

双字加 / 减切换计数器 CUDD 有一个可以切换加 / 减门的选择输入，因此需要三个输入。双字加 / 减切换计数器的电路必须按以下顺序编写：预置输入、脉冲输入、加 / 减切换输入、CUD 指令和计数器数字，后跟计数器预置值（0 ~ 4,294,967,295）。

梯形图

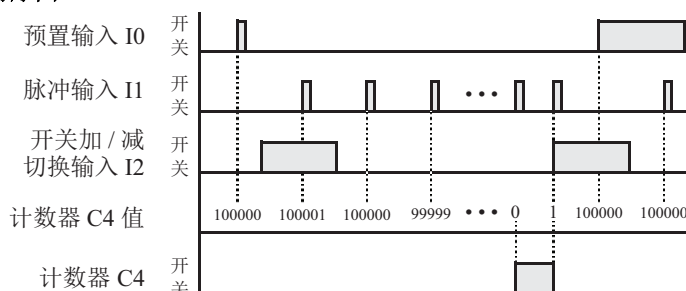


程序列表

指令	数据
LOD	I0
LOD	I1
LOD	I2
CUDD	C4
	100000
LOD	I3
AND	C4
OUT	Q2

- 预置输入必须在一开始就打开，以便让当前值恢复到预置值。
- 计数有可能开始之前，必须关闭预置输入。
- 当加 / 减切换输入打开时，将切换为加计数模式。
- 当加 / 减切换输入关闭时，将切换为减计数模式。
- 当前值达到 0(减计数)之后，它将在下一次减计数时更改为 4,294,967,295。
- 当前值达到 4,294,967,295(加计数)之后，它将在下一次加计数时更改为 0。
- 计数器输出只在当前值为 0 时才会打开。

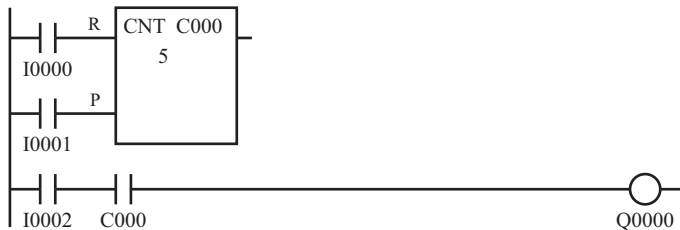
时序图



## 4: 基本指令

### 注释:

- 双字计数器指令使用两个连续的计数器，并且计数器在一个用户程序中只能被使用一次。
- 当计数器设备被指定为高级指令的源设备时，将读取当前值作为源数据。当计数器设备被指定为高级指令的目标设备时，高级指令的结果将被设置为计数器的预设值。
- 设置 32 位数据存储的方法取决于控制器类型。  
 FT2J: 高位字存储在第一个设备。  
 FC6A: 它遵循“功能设置”对话框中指定的 32 位数据存储设置。参见 FC6A 系列 MICROSmart 用户手册第 5 章“32 位数据存储设置”。请参见《FC6A 型 MICROSmart 用户手册》第 5 章“32 位数据存储设置”。
- 在 WindLDR 中，在计数器指令之后可对一些指令进行串联编程，因此不能像上面的梯形图中显示的那样自动连接到右母线。有关详情，请参见第 4-17 页上的“”。



- 当计数器设备被指定为高级指令的源设备时，将读取当前值作为源数据。当计数器设备被指定为高级指令的目标设备时，高级指令的结果将被设置为计数器的预设值。
- 有关计数器指令的梯形图编程限制，请参见第 4-38 页上的“梯形图编程限制”。
- 在梯形图程序中，计数器设备地址不能使用多次。
- 计数器指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 计数器指令的预设值可通过外部设备（可编程显示器等）、WindLDR 或控制器操作进行更改。断电后，已更改的预设值将被清除，并加载原始预设值。但是，即使电源关闭，FC6A 型也可以保留更改的预设值。保留更改预设值的设置方法如下。

### 使用 WindLDR

从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”，在“PLC”组中单击“状态”，然后单击“TIM/CNT 更改状态”的“确定”。预设值一经确认，即使清除也无法重置为更改前的预设值。

### 使用 HMI 模块

有关 HMI 模块的详情，请参见《FC6A 型 MICROSmart 用户手册》第 7 章“HMI 功能”。

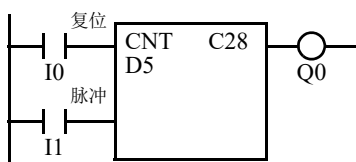
有关更改、确认和清除预设值时的数据移动，请参见第 4-22 页上的“更改、确认和清除定时器和计数器的预置值”。

- 要比较双字计数器指令的当前值，请使用 LC(D) 或 CMP(D) 指令。计数器比较指令不能比较双字计数器指令的当前值。

## 计数器电路

在计数器和双字计数器指令之后，可以直接编程 OUT、OUTN、SET、RST、TML、TIM、TMH、TMS、TMLO、TIMO、TMHO 或 TMSO 指令。计数器和双字计数器指令的计数器输出将成为连接在这些指令之后的指令的输入。

### 梯形图

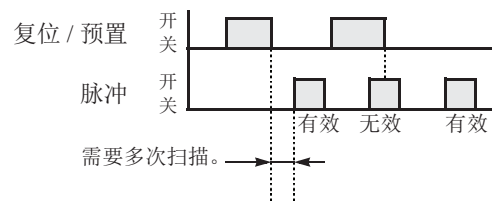


### 程序列表

指令	数据
LOD	I0
LOD	I1
CNT	C28
OUT	Q0

### 有效脉冲输入

复位或预置输入优先于脉冲输入。在复位或预置输入从开更改为关之后的一个扫描期间，计数器将开始对从关更改为开的脉冲输入进行计数。



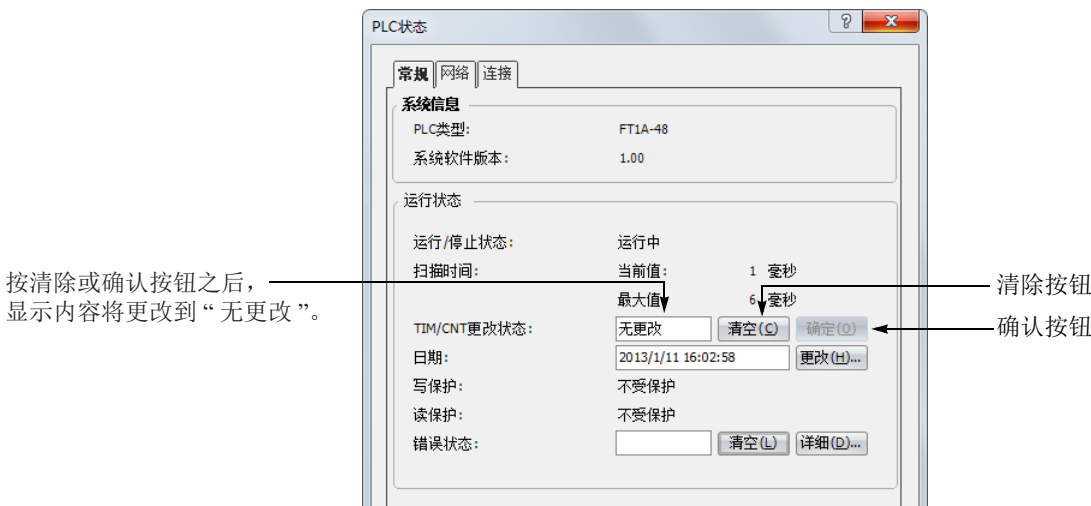
计数有可能开始之前，复位输入必须关闭。

## 4: 基本指令

### 更改、确认和清除定时器和计数器的预置值

通过在 WindLDR 上选择“联机”>“监控”>“监控”，然后选择“联机”>“自定义”>“新建自定义监控”，可以更改定时器与计数器的预置值，新值将如前文所述传送到 FC6A 型 RAM。临时更改预置值之后，即可将更改写入位于 FC6A 型 ROM 中的用户程序，或者将其从 RAM 中清除。电源关闭时，会清除 RAM 内临时更改的预置值并加载原始预置值。确认更改后的预置值，并存储到 ROM 内。仅当 FC6A 型停止时，才可确认定时器 / 计数器预置值。

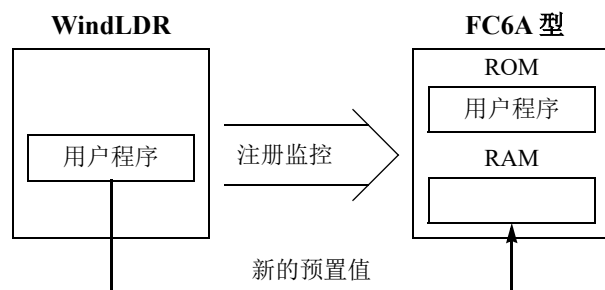
要从 WindLDR 菜单栏访问 PLC 状态对话框，请选择“联机”>“监控”>“监控”，然后选择“联机”>“状态”。



### 更改定时器 / 计数器预置值时的数据传送

在 WindLDR 中使用“注册监控”来更改定时器 / 计数器预置值时，新的预置值将写入 FC6A 型 RAM 中。ROM 中的用户程序和预置值不会被更改。

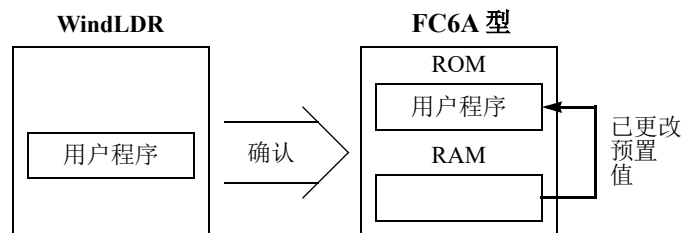
**注释：** 还可使用 LCD 屏幕和按钮来更改预置值，并确认更改后的预置值。



### 确认更改后的预置值时的数据传送

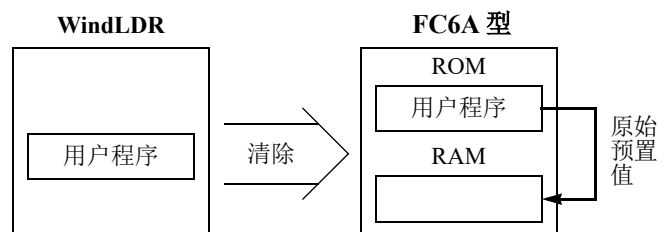
在按清除按钮之前，如果按了确认按钮，则 FC6A 型 RAM 中的更改后的定时器 / 计数器预置值将被写入 ROM。

确认之后，在上传用户程序时，其预置值已更改的用户程序将从 FC6A 型 ROM 上传到 WindLDR。



### 清除更改后的预置值以还原原始值时的数据传送

如果更改了 FC6A 型 RAM 中定时器和计数器的预置值，将不会自动更新用户内存 (ROM) 中的预置值。这对还原原始预置值是有用的。在按确认按钮之前，如果按了清除按钮，更改后的定时器 / 计数器预置值将从 RAM 中清除，并且原始预置值将从 ROM 加载到 RAM。



**CC= (计数器比较 (=))**

CC= 指令是用于计数器当前值的相等比较。

**CC>= (计数器比较 (>=))**

CC>= 指令是用于计数器当前值的等于或大于比较。

FT2J FC6A

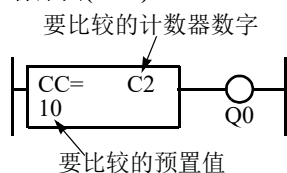
CC= 指令是用于计数器当前值的相等比较指令。通常，该指令用于将当前值与已在程序中设置的值进行比较。如果计数器值等于给定值，则启动希望的输出。

CC>= 指令是用于计数器当前值的等于或大于比较指令。通常，该指令将当前值与已在程序中设置的值进行比较。如果计数器值等于或大于给定值，则启动希望的输出。

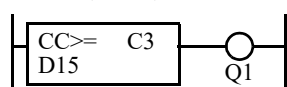
编写计数器比较指令时，需要两个地址。若为常量，则在 0 ~ 65,535 的范围内进行指定。

若为间接指定，则以数据寄存器编号进行指定，并在 0 ~ 65,535 的范围内指定数据寄存器的值。

预置值可以使用十进制常量或数据寄存器指定。使用数据寄存器时，数据寄存器的数据将成为预置值。

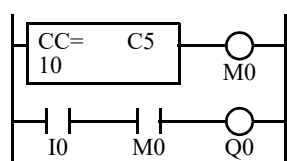
**梯形图(CC=)****程序列表**

指令	数据
CC=	C2
	10
OUT	Q0

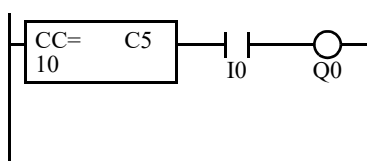
**梯形图(CC>=)****程序列表**

指令	数据
CC>=	C3
	D15
OUT	Q1

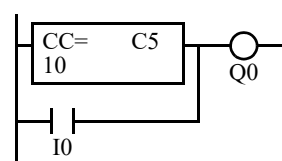
- 可以对不同的预置值重复使用 CC= 和 CC>= 指令。
- 比较指令仅比较当前值。计数器的状态不影响此功能。
- 比较指令还可充当隐式的 LOD 指令。
- 比较指令可以与内部继电器一起使用，将在单独的程序地址中对该内部继电器执行 AND 或 OR。
- 与 LOD 指令类似，比较指令可以后跟 AND 和 OR 指令。

**梯形图****程序列表**

指令	数据
CC=	C5
	10
OUT	M0
LOD	I0
AND	M0
OUT	Q0

**梯形图****程序列表**

指令	数据
CC=	C5
	10
AND	I0
OUT	Q0

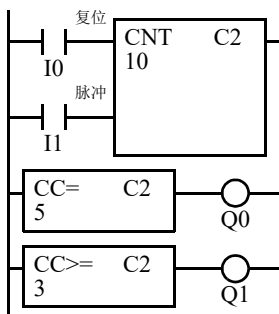
**梯形图****程序列表**

指令	数据
CC=	C5
	10
OR	I0
OUT	Q0

## 4: 基本指令

### 示例：CC= 和 CC>= (计数器比较)

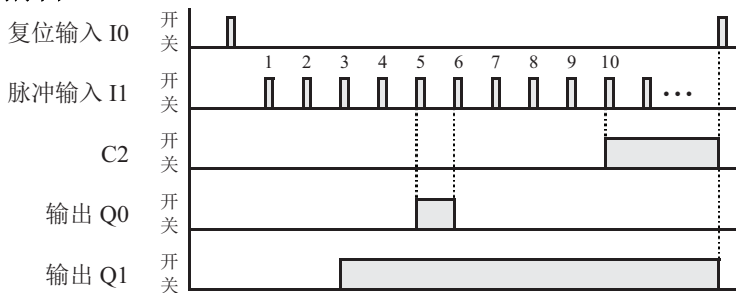
梯形图1



程序列表

指令	数据
LOD	I0
LOD	I1
CNT	C2
	10
CC=	C2
	5
OUT	Q0
CC>=	C2
	3
OUT	Q1

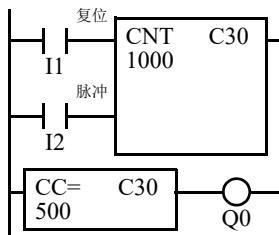
时序图



当计数器 C2 的当前值为 5 时，输出 Q0 将打开。

当计数器 C2 的当前值达到 3 时，输出 Q1 将打开，并且保持到计数器 C2 复位。

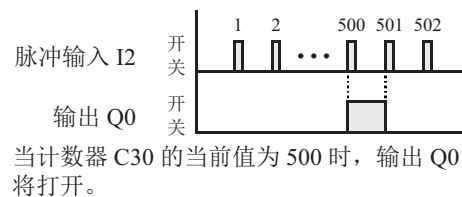
梯形图2



程序列表

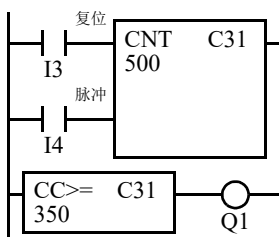
指令	数据
LOD	I1
LOD	I2
CNT	C30
	1000
CC=	C30
	500
OUT	Q0

时序图



当计数器 C30 的当前值为 500 时，输出 Q0 将打开。

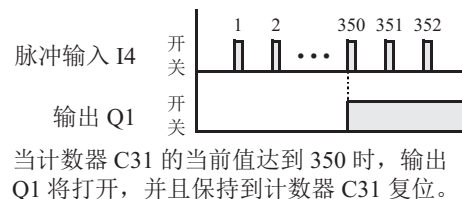
梯形图3



程序列表

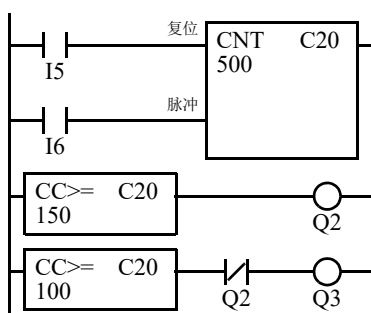
指令	数据
LOD	I3
LOD	I4
CNT	C31
	500
CC≧	C31
	350
OUT	Q1

时序图



当计数器 C31 的当前值达到 350 时，输出 Q1 将打开，并且保持到计数器 C31 复位。

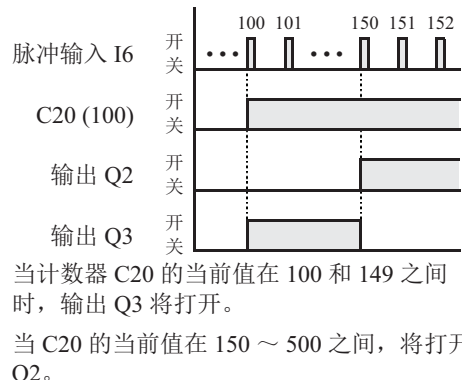
梯形图4



程序列表

指令	数据
LOD	I5
LOD	I6
CNT	C20
	500
CC>=	C20
	150
OUT	Q2
CC>=	C20
	100
ANDN	Q2
OUT	Q3

时序图



当计数器 C20 的当前值在 100 和 149 之间时，输出 Q3 将打开。

当 C20 的当前值在 150 ~ 500 之间，将打开 Q2。



**DC= (数据寄存器比较 (=))**

DC= 指令是用于数据寄存器值的相等比较。

**DC>= (数据寄存器比较 (>=))**

DC>= 指令是用于数据寄存器值的等于或大于比较。

FT2J FC6A

DC= 指令是用于数据寄存器值的相等比较指令。通常，该指令将数据寄存器值与已在程序中设置的值进行比较。如果数据寄存器值等于给定值，则启动希望的输出。

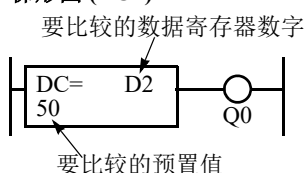
DC>= 指令是用于数据寄存器值的等于或大于比较指令。通常，该指令将数据寄存器值与已在程序中设置的值进行比较。如果数据寄存器值等于或大于给定值，则启动希望的输出。

如果在程序中使用数据寄存器比较指令，则需要两个地址。若为常量，则在 0 ~ 65,535 的范围内进行指定。若为间接指定，则以数据寄存器编号进行指定，并在 0 ~ 65,535 的范围内指定数据寄存器的值。

预置值可以使用十进制常量或数据寄存器指定。使用数据寄存器时，数据寄存器的数据将成为预置值。

有关 LC(接点比较等于) 指令，请参见第 6-8 页上的“比较两个指定数据，并根据结果打开 / 关闭。”。

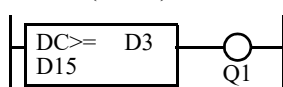
梯形图 (DC=)



程序列表

指令	数据
DC=	D2
	50
OUT	Q0

梯形图 (DC&gt;=)

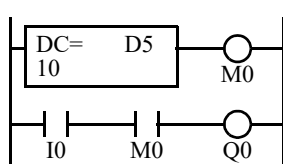


程序列表

指令	数据
DC>=	D3
	D15
OUT	Q1

- 可以对不同预置值重复使用 DC= 和 DC>= 指令。
- 比较指令还可充当隐式的 LOD 指令。
- 比较指令可以与内部继电器一起使用，将在单独的地址中对该内部继电器执行 AND 或 OR。
- 与 LOD 指令类似，比较指令可以后跟 AND 和 OR 指令。

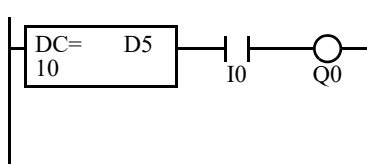
梯形图



程序列表

指令	数据
DC=	D5
	10
OUT	M0
LOD	I0
AND	M0
OUT	Q0

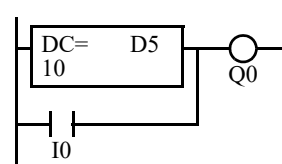
梯形图



程序列表

指令	数据
DC=	D5
	10
AND	I0
OUT	Q0

梯形图



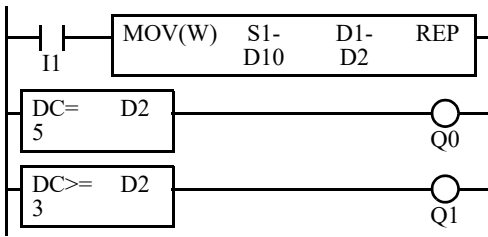
程序列表

指令	数据
DC=	D5
	10
OR	I0
OUT	Q0

## 4: 基本指令

示例：DC= 和 DC>=( 数据寄存器比较)

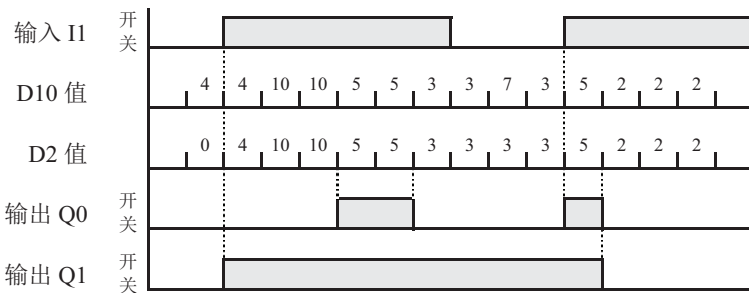
梯形图1



程序列表

指令	数据
LOD	I1
MOV(W)	D10 - D2 -
DC=	D2 5
OUT	Q0
DC <sup>3</sup>	D2 3
OUT	Q1

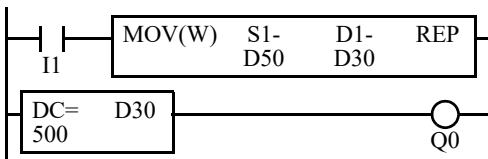
时序图



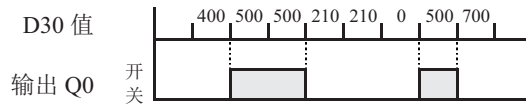
当数据寄存器 D2 的值为 5 时，输出 Q0 将打开。

当数据寄存器 D2 的值为 3 或更大数字时，输出 Q1 将打开。

梯形图2

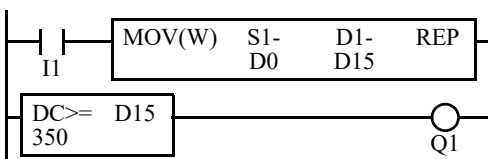


时序图

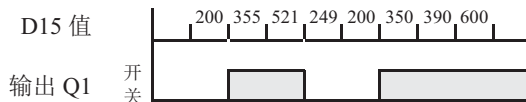


当数据寄存器 D30 的值为 500 时，输出 Q0 将打开。

梯形图3

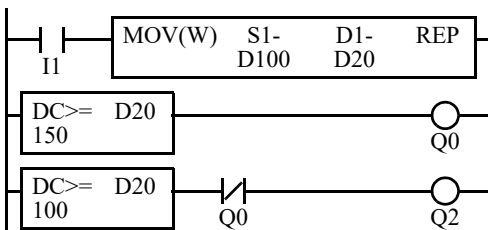


时序图

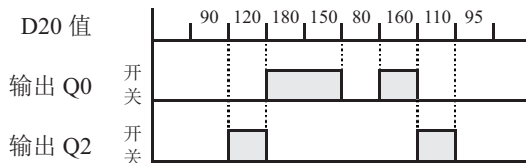


当数据寄存器 D15 的值为 350 或更大数字时，输出 Q1 将打开。

梯形图4



时序图



当数据寄存器 D20 的值在 149 和 100 之间时，输出 Q2 将打开。

当 D20 的当前值在 150 以上时，将打开输出 Q0。

## SFR（右移移位寄存器）

SFR 指令用于右移的移位寄存器。

## SFRN（左移移位寄存器）

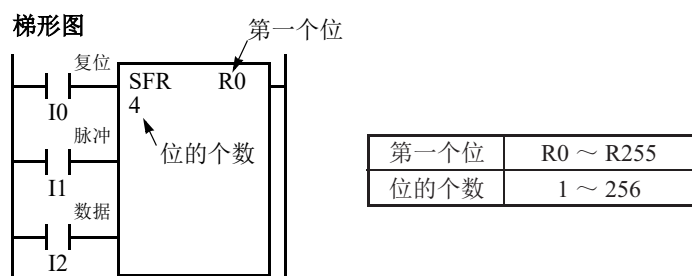
SFRN 指令用于左移的移位寄存器。

FT2J FC6A

### 右移移位寄存器（SFR）

编写 SFR 指令时，总是需要两个地址。输入 SFR 指令时，后面要跟从合适的设备数字中选择的移位寄存器数字。移位寄存器数字对应于第一个位或首位。在 SFR 指令后面，位的个数是第二个需要的地址。

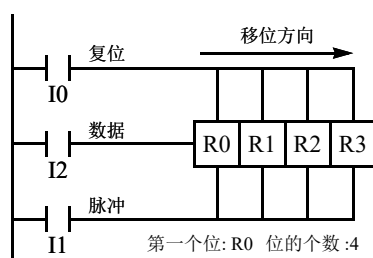
SFR 指令需要三个输入。右移移位寄存器电路必须按以下顺序编写：复位输入、脉冲输入、数据输入和 SFR 指令，后跟第一个位和位的个数。



程序列表

指令	数据
LOD	I0
LOD	I1
LOD	I2
SFR	R0 4

结构图



### 复位输入

复位输入将导致移位寄存器的每个位的值恢复到零。可以在启动时用初始化脉冲特殊内部继电器 M8120 来初始化移位寄存器。

### 脉冲输入

脉冲输入将触发数据发生移位。对于右移移位寄存器，移位方向是向前；对于左移移位寄存器，则相反。当遇到前沿脉冲（即当脉冲打开）时，将发生数据移位。如果脉冲已经打开并保持不变，则不发生数据移位。

### 数据输入

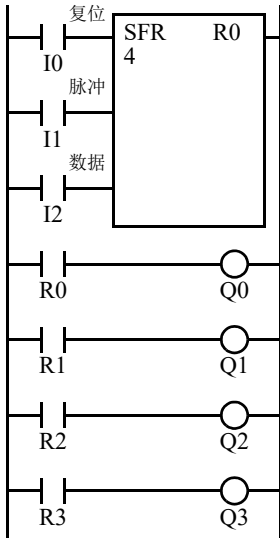
数据输入是在发生右移数据移位时移位到第一个位中的信息，或者是发生左移数据移位时移位到最后一个位中的信息。

### 注释：

- 电源关闭时，所有移位寄存器位的状态通常将被清除。根据需要，也可以使用“功能设置”来维持移位寄存器位的状态。有关详情，请参见以下手册。  
FT2J 型：《WindO/I-NV4 用户手册》第 28 章中的“2.4 内存备份”  
FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“内存备份”
- SFR 指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 关于对移位寄存器指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。

## 4: 基本指令

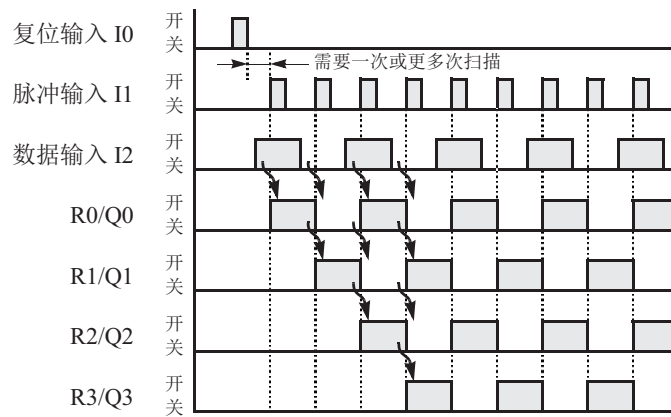
梯形图



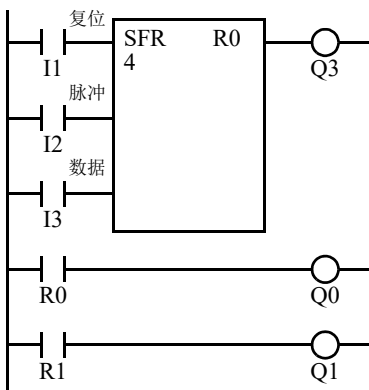
程序列表

指令	数据
LOD	I0
LOD	I1
LOD	I2
SFR	R0 4
LOD	R0
OUT	Q0
LOD	R1
OUT	Q1
LOD	R2
OUT	Q2
LOD	R3
OUT	Q3

时序图



梯形图

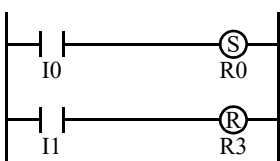


程序列表

指令	数据
LOD	I1
LOD	I2
LOD	I3
SFR	R0 4
OUT	Q3
LOD	R0
OUT	Q0
LOD	R1
OUT	Q1

- 可以直接在 SFR 指令之后编写最后一个位的状态输出。在此示例中，位 R3 的状态被读取到输出 Q3。
- 使用 LOD R# 指令，可以读取每个位。

### 置位和复位移位寄存器位

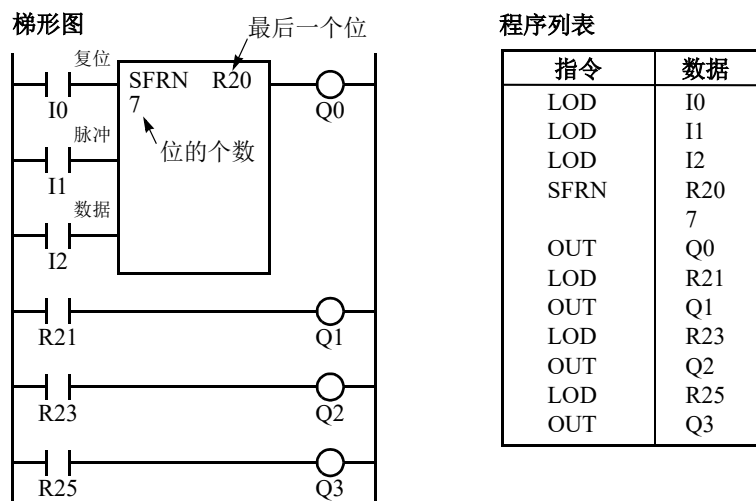


- 使用 SET 指令可以打开任何移位寄存器位。
- 使用 RST 指令可以关闭任何移位寄存器位。
- SET 或 RST 指令可以由任何输入条件启动。

### 左移移位寄存器 (SFRN)

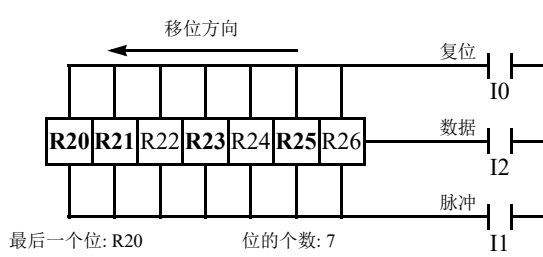
要进行左移移位，请使用 SFRN 指令。编写 SFRN 指令时，总是需要两个地址。输入 SFRN 指令时，后面要跟从合适的设备数字中选择的移位寄存器数字。移位寄存器数字对应字符串中的最低位数字。位数是 SFRN 指令之后的第二个所需地址。

SFRN 指令需要三个输入。左移移位寄存器电路必须按以下顺序编写：复位输入、脉冲输入、数据输入和 SFRN 指令，后跟最后一个位和位数。



- 最后一个位的状态输出可以直接编写在 SFRN 指令之后。在此示例中，位 R20 的状态被读取到输出 Q0。
- 可以使用 LOD R# 指令读取每个位。
- 有关复位、脉冲和数据输入的详情，请参见第 4-27 页上的“右移移位寄存器 (SFR)”。

### 结构图



只对那些粗体显示的位启动输出。

### 注释:

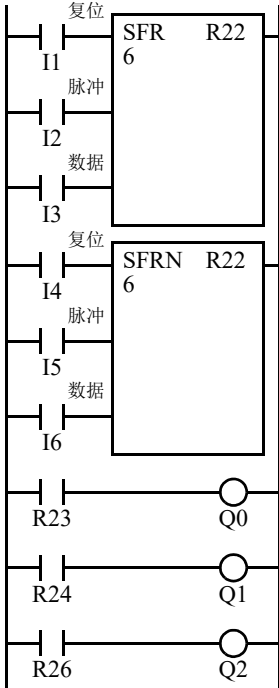
- 电源关闭时，所有移位寄存器位的状态通常将被清除。根据需要，也可以使用“功能设置”来维持移位寄存器位的状态。有关详情，请参见以下手册。  
FT2J 型：《WindO/I-NV4 用户手册》第 28 章中的“2.4 内存备份”  
FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“内存备份”
- SFRN 指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 关于对移位寄存器指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。

## 4: 基本指令

### 双向移位寄存器

要创建双向移位寄存器，首先需要编写 SFR 指令（第 4-27 页上的“右移移位寄存器（SFR）”一节对此做了详细介绍）。然后，编写 SFRN 指令（第 4-29 页上的“左移移位寄存器（SFRN）”一节对此做了详细介绍）。

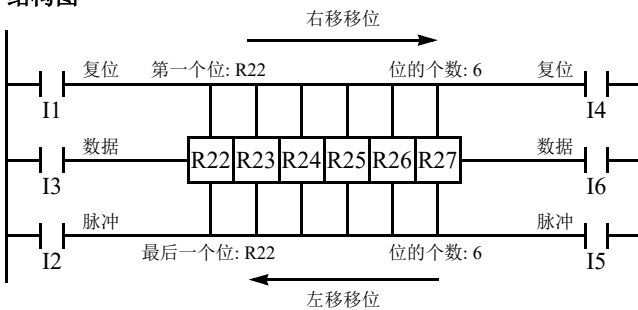
梯形图



程序列表

指令	数据
LOD	I1
LOD	I2
LOD	I3
SFR	R22 6
LOD	I4
LOD	I5
LOD	I6
SFRN	R22 6
LOD	R23
OUT	Q0
LOD	R24
OUT	Q1
LOD	R26
OUT	Q2

结构图



只对那些粗体显示的位启动输出。

## SOTU（上升沿微分）

SOTU 指令是用于在输入上升沿时，打开仅一次扫描输出。

## SOTD（下降沿微分）

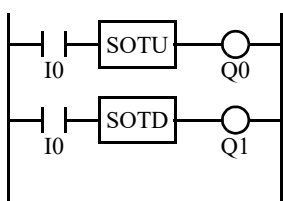
SOTD 指令是用于在输入下降沿时，打开仅一次扫描输出。

FT2J FC6A

SOTU 指令用于“查找”给定输入从关到开的转换。SOTU 指令用于“查找”给定输入从关到开的转换。发生此转换时，希望的输出将在一个扫描周期内保持打开。SOTU 或 SOTD 指令将输入信号转换为“仅一次”脉冲信号。

如果在给定输入已打开时启动操作，SOTU 输出将不打开。从关到开的转换是触发 SOTU 指令的过程。

梯形图

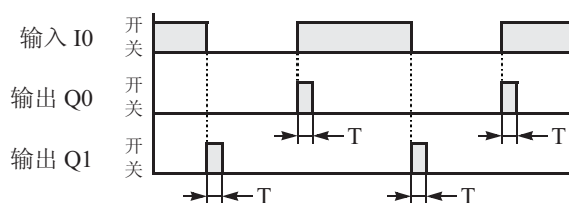


程序列表

指令	数据
LOD	I0
SOTU	
OUT	Q0
LOD	I0
SOTD	
OUT	Q1

**注释：** 关于对 SOTU 和 SOTD 指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。

时序图



“T” 等于一个扫描周期（仅一次脉冲）。

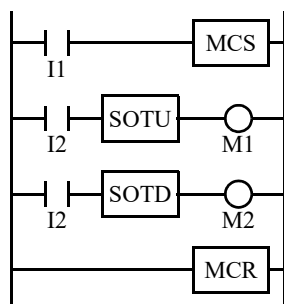
### 注释：

- 梯形图程序中可使用的射频指令总数取决于控制器类型。  
FT2J 型：最多 1024 条  
FC6A 型：最多 3027 条
- SOTU 或 SOTD 指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 关于对移位寄存器指令的梯形图编程的限制，请参见第 4-38 页上的“梯形图编程限制”。
- 在使用分支指令创建的子程序和分支目标梯形图程序中，SOTU 或 SOTD 指令的导通输出被保留，直到下一次执行相同的触发指令。

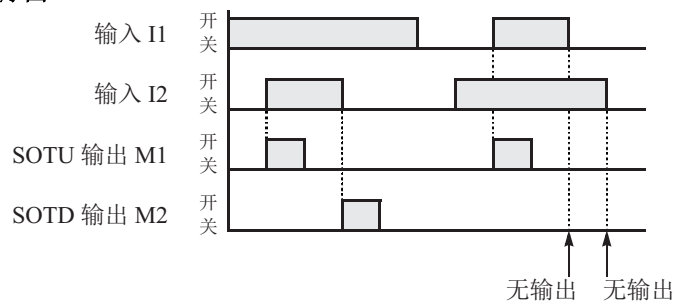
在 MCS 和 MCR 指令（有关详情，请参见第 4-33 页上的“MCS（主控继电器开始）”）之间使用 SOTU 和 SOTD 指令时，存在一种特殊情况。当 MCS 指令的输入 I1 已打开时，如果 SOTU 指令的输入 I2 打开，那么 SOTU 输出将打开。当输入 I1 已打开时，如果 SOTD 指令的输入 I2 关闭，那么 SOTD 输出将打开。当输入 I2 已打开时，如果输入 I1 打开，那么 SOTU 输出将打开。但是，当输入 I2 已打开时，如果输入 I1 关闭，那么 SOTD 输出将不打开，如下所示。

## 4: 基本指令

梯形图



时序图





## MCS（主控继电器开始）

MCS 指令是主控电路的开始点。

## MCR（主控继电器结束）

MCR 指令是主控电路的结束点。

FT2J FC6A

MCS（主控继电器开始）指令通常与 MCR（主控继电器结束）指令组合使用。MCS 指令还可以与 END 指令（而不是 MCR 指令）一起使用。

位于 MCS 指令前面的输入关闭时，将执行 MCS，以便强制关闭在 MCS 和 MCR 之间的部分的所有输入。位于 MCS 指令前面的输入打开时，将不执行 MCS，以便按照实际的输入状态执行在它后面的程序。

当 MCS 指令的输入条件为关闭并执行 MCS 时，将按如下所示执行 MCS 和 MCR 之间的其它指令：

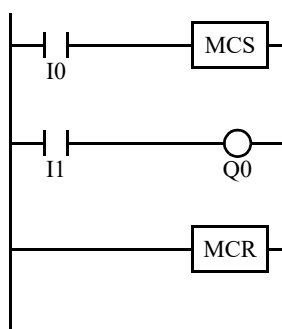
指令	状态
OUT	全部关闭。
OUTN	全部打开。
SET 和 RST	全部保持当前状态。
TML、TIM、TMH 和 TMS	当前值复位为 0。 关闭超时状态。
TMLO、TIMO、TMHO 和 TMSO	当前值开始倒计时。 当当前值为 0 时，关闭超时状态。
CNT、CDP 和 CUD	保持当前值。
CNTD、CDPD 和 CUDD	关闭脉冲输入。 关闭计数器输出状态。
SFR 和 SFRN	保持移位寄存器位状态。 关闭脉冲输入。 关闭最后一个位的输出。

无法设置 MCR 指令的输入条件。

可以与一个 MCR 指令一起使用多个 MCS 指令。

相应的 MCS/MCR 指令无法嵌套在另一对相应的 MCS/MCR 指令中。

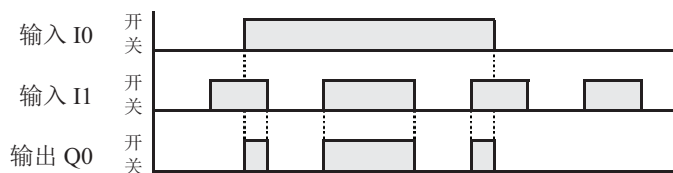
梯形图



程序列表

指令	数据
LOD	I0
MCS	
LOD	I1
OUT	Q0
MCR	

时序图



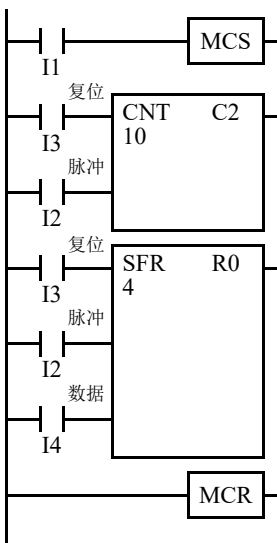
输入 I0 关闭时，将执行 MCS，以便强制关闭随后的输入。

当输入 I0 打开时，将不执行 MCS，以便按照实际的输入状态执行后面的程序。

## 4: 基本指令

### 主控电路中的计数器和移位寄存器

梯形图

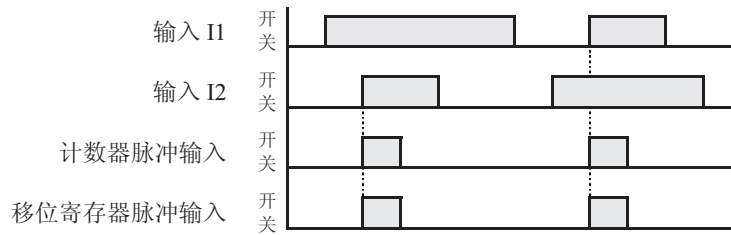


输入 I1 打开时，将不执行 MCS，以便按照随后的输入 I2 ~ I4 的实际状态执行计数器和移位寄存器。

当输入 I1 关闭时，将执行 MCS，以便强制关闭随后的输入 I2 ~ I4。

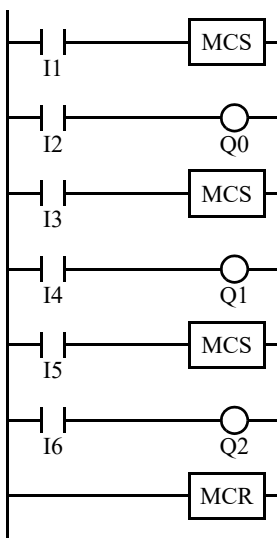
当输入 I2 已打开时，如果输入 I1 打开，则计数器和移位寄存器的脉冲输入将按如下所示打开。

时序图



### 多重使用 MCS 指令

梯形图



程序列表

指令	数据
LOD	I1
MCS	
LOD	I2
OUT	Q0
LOD	I3
MCS	
LOD	I4
OUT	Q1
LOD	I5
MCS	
LOD	I6
OUT	Q2
MCR	

此主控电路的优先级将按顺序高于 I1、I3 和 I5。

输入 I1 关闭时，将执行第一个 MCS，以便强制关闭随后的输入 I2 ~ I6。

输入 I1 打开时，将不执行第一个 MCS，以便按照 I2 ~ I6 的实际输入状态执行后面的程序。

I1 打开并且 I3 关闭时，将执行第二个 MCS，以便强制关闭随后的输入 I4 ~ I6。

I1 和 I3 都打开时，将不执行第一个和第二个 MCS，以便按照 I4 ~ I6 的实际输入状态执行后面的程序。

## JMP (跳转)

JMP 指令用于跳转到指定的梯形图程序区域。

## JEND (跳转结束)

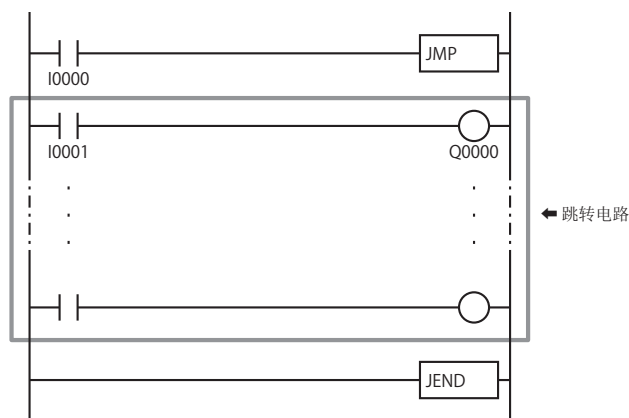
JEND 指令用于 JMP 指令指定的梯形图程序区域的结束点。

FT2J FC6A

JMP 指令通常与 JEND 指令组合使用。程序结束时，还可以与 END 指令（而不是 JEND 指令）一起使用 JMP 指令。

这些指令用来继续通过 JMP 和 JEND 之间的程序部分，而不执行这些部分。这类似于 MCS/MCR 指令，但事实要执行 MCS 和 MCR 指令之间的程序部分。

**注释：** JMP 指令到 JEND 指令（或 END 指令）之间的梯形图程序称为跳转电路。执行 JMP 指令时，不处理跳转电路。



当紧靠 JMP 指令之前的操作结果为打开时，JMP 将有效，并且不执行程序。当紧靠 JMP 指令之前的操作结果为关闭时，JMP 将无效，并且执行程序。

当 JMP 指令的输入条件为打开并执行 JMP 时，将按如下所示执行 JMP 和 JEND 之间的其它指令：

指令	状态
OUT 和 OUTN	全部保持当前状态。
SET 和 RST	全部保持当前状态。
TML、TIM、TMH 和 TMS	保持当前值。
TMLO、TIMO、TMHO 和 TMSO	保持超时状态。
CNT、CDP 和 CUD	保持当前值。
CNTD、CDPD 和 CUDD	关闭脉冲输入。 保持计数器输出状态。
SFR 和 SFRN	保持移位寄存器位状态。 关闭脉冲输入。 保持最后一个位的输出。

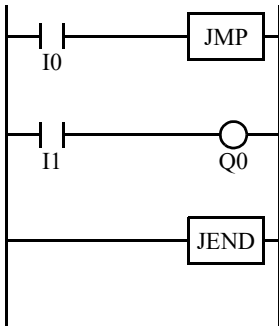
无法设置 JEND 指令的输入条件。

多个 JMP 指令可以与一个 JEND 指令一起使用。

相应的 JMP/JEND 指令无法嵌套在另一对相应的 JMP/JEND 指令中。

## 4: 基本指令

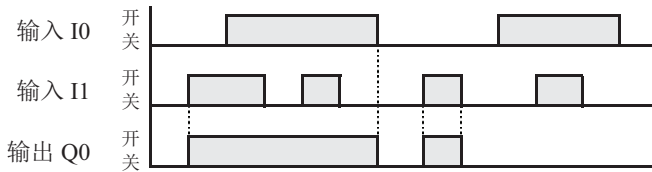
梯形图



程序列表

指令	数据
LOD	I0
JMP	
LOD	I1
OUT	Q0
JEND	

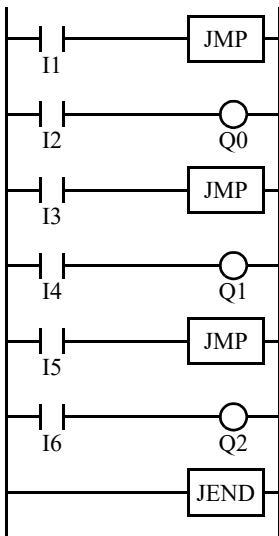
时序图



当输入 I0 打开时，将执行 JMP，以便保持随后的输出状态。

当输入 I0 关闭时，将不执行 JMP，以便按照实际的输入状态执行后面的程序。

梯形图



程序列表

指令	数据
LOD	I1
JMP	
LOD	I2
OUT	Q0
LOD	I3
JMP	
LOD	I4
OUT	Q1
LOD	I5
JMP	
LOD	I6
OUT	Q2
JEND	

此跳转电路的优先级将按顺序高于 I1、I3 和 I5。

当输入 I1 打开时，将执行第一个 JMP，以便保持 Q0 ~ Q2 的随后的输出状态。

当输入 I1 关闭时，将不执行第一个 JMP，以便按照 I2 ~ I6 的实际输入状态执行后面的程序。

当 I1 关闭并且 I3 打开时，将执行第二个 JMP，以便保持 Q1 和 Q2 的随后的输出状态。

当 I1 和 I3 都关闭时，将不执行第一个和第二个 JMP，以便按照 I4 ~ I6 的实际输入状态执行后面的程序。

## END

END 指令用于指定梯形图程序的结束点。

FT2J FC6A

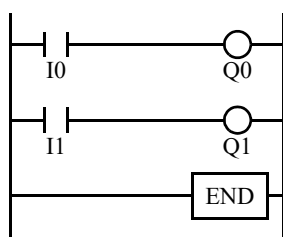
程序末尾总是需要有 END 指令；但是，没有必要在最后一个程序指令之后编写 END 指令。在程序结束时，WindLDR 会自动附加 END 指令。

扫描是执行从地址零到 END 指令的所有指令的过程。此执行过程所需的时间称为一个扫描周期。扫描周期取决于程序长度而变化，而程序长度则对应发现 END 指令的地址。

在扫描周期期间，将按顺序处理程序指令。这就是为什么最接近 END 指令的输出指令的优先级高于相同输出的上一个指令的原因。只有在扫描中的所有逻辑都已被处理之后，才会启动输出。

输出将同时发生，并且这是执行 END 指令的第一部分。执行 END 指令的第二部分是监控所有输入，这也是同时完成的。然后，程序指令就准备好再次按顺序被处理。

梯形图



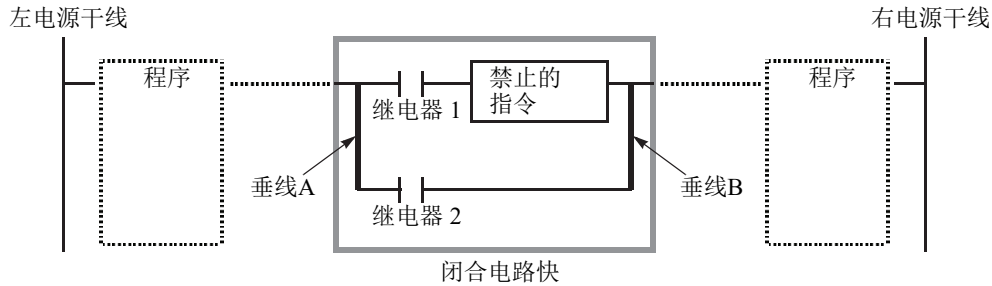
程序列表

指令	数据
LOD	I0
OUT	Q0
LOD	I1
OUT	Q1
END	

## 4: 基本指令

### 梯形图编程限制

由于 WindLDR 的结构，不能编写下列梯形图 — 由垂线构成的闭合电路块，除了左右电源干线外，闭合电路块含有一个以上下图所示的禁止的指令。

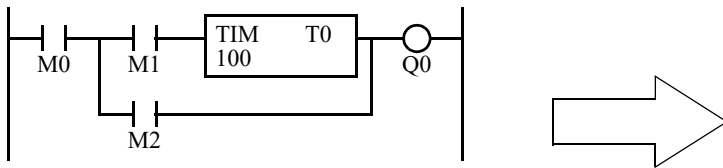


禁止指令	OUT, OUTN, SET, RST, TML, TIM, TMH, TMS, TMLO, TIMO, TMHO, TMSO, CNT, CDP, CUD, CNTD, CDPD, CUDD, SFR, SFRN, SOTU, SOTD
错误检测	当转换梯形图时，显示一条错误信息，例如“TIM 监视到一个无效设备”。转换不能创建助记符而且程序不能下载到 FC6A 型。

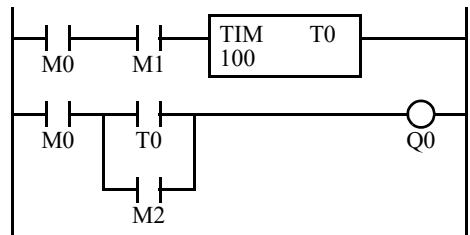
### 更改进制的梯形图程序

如下图所示更改禁止的梯形图程序可进行预定操作。

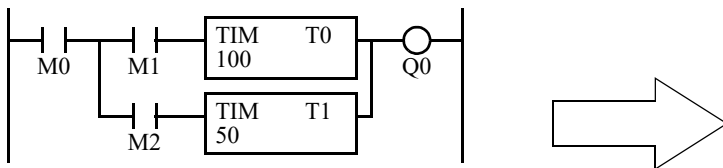
禁止的梯形图程序 1



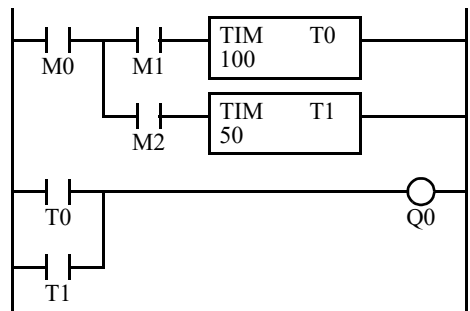
更改的梯形图程序 1



禁止的梯形图程序 2



更改的梯形图程序 2



# 5: 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 位或 32 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。

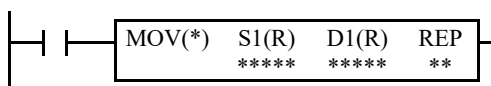
NSET(N 数据置位) 和 NRS(N 数据重复置位) 指令可以用于置位设备组的值。XCHG(交换) 指令用于在两个设备间交换字与双字的数据。定时器或计数器的当前值可以使用 TCCST(存储定时器 / 计数器当前值) 指令进行更改。

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）

直接移动数据。

FT2J FC6A



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。

### 注释:

- 当选择 F（浮点）数据类型，并且 S1 不符合标准浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 利用 MOV 指令（无重复设置）改写定时器 / 计数器预置值时，M8124 不会切换到 ON。如需改写预置值，请使用 IMOV 指令。使用 IMOV 指令改写定时器或计数器设置。

### 有效设备

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

有效设备范围取决于控制器类型。有关详情，请参见第 2-1 页上的“设备地址”。

▲ 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，将显示定时器 / 计数器当前值（TC 或 CC）。

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。

仅可于数据类型为 L（长整）时使用。

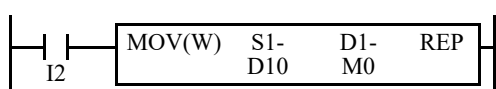
### 有效数据类型

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（字或整数数据）或 32 点（双字或长整数数据）。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	
L（长整型）	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（字或整数数据）或 2 点（双字、长整数或浮点数据）。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。
F（浮点型）	X	

## 5: 传送指令

### 示例：MOV

#### 数据类型：字型

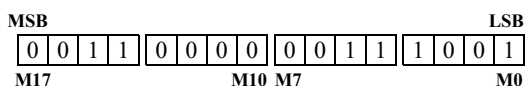


D10 → M0

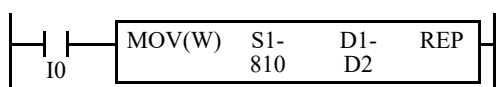
输入 I2 打开时，由源设备 S1 指定的数据寄存器 D10 中的数据将传送到由目标设备 D1 指定的、以 M0 开始的 16 个内部继电器。

D10 12345 → M0 - M7, M10 - M17

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）。

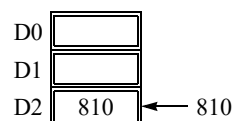


#### 数据类型：字型



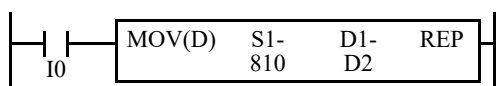
810 → D2

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。



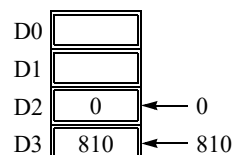
整型数据类型与字型数据类型的数据传送操作相同。

#### 数据类型：双字型



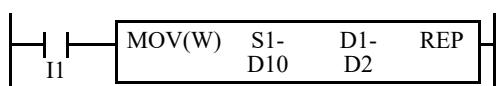
810 → D2·D3

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。



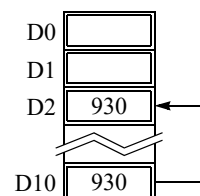
长整型数据类型与双字型数据类型的数据传送操作相同。

#### 数据类型：字型

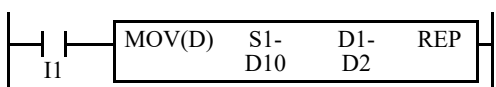


D10 → D2

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。

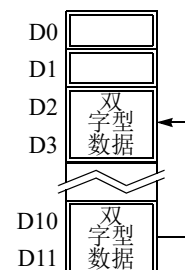


#### 数据类型：双字型



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。



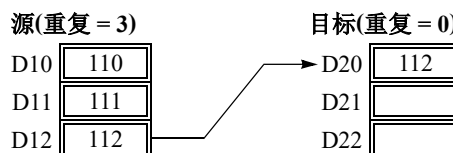
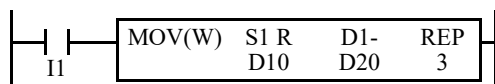


## 传送指令中的重复操作

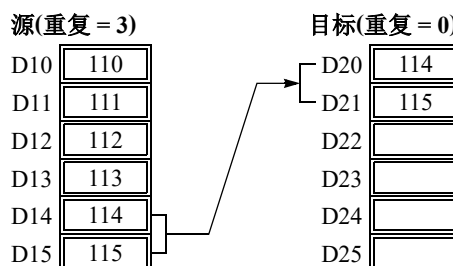
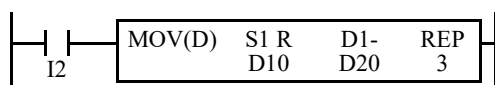
### 重复源设备

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



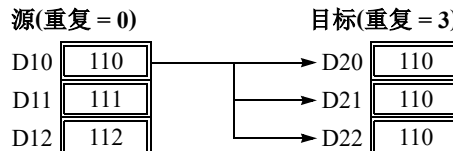
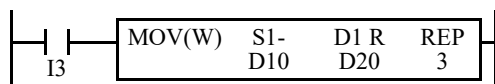
- 数据类型：双字型



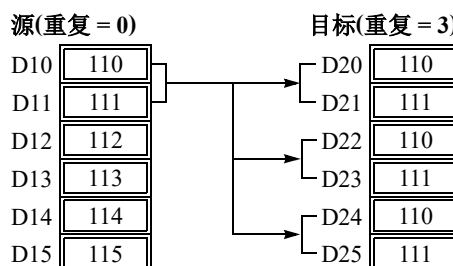
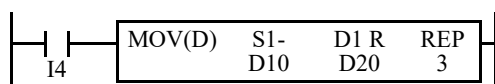
### 重复目标设备

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

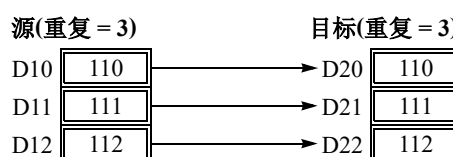
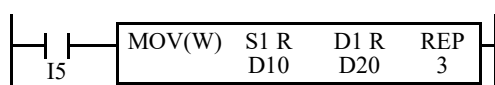


### 重复源设备和目标设备

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。

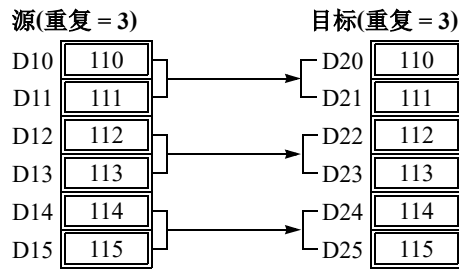
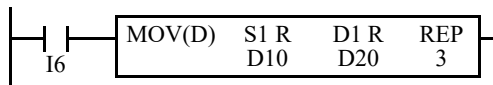
**注释：**对于指定为重复的源和目标，BMOV（块传送）指令与 MOV 指令的作用相同。

- 数据类型：字型



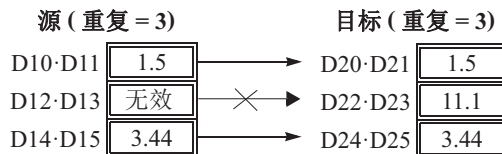
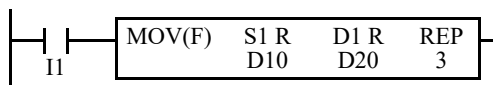
## 5: 传送指令

- 数据类型：双字型



- 数据类型：浮点型

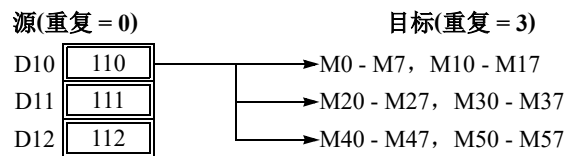
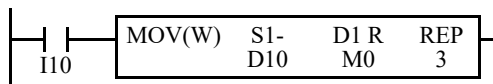
当源数据不符合重复操作中的标准浮点格式时，将发生程序执行错误，并且源数据不能传送到目标数据中。



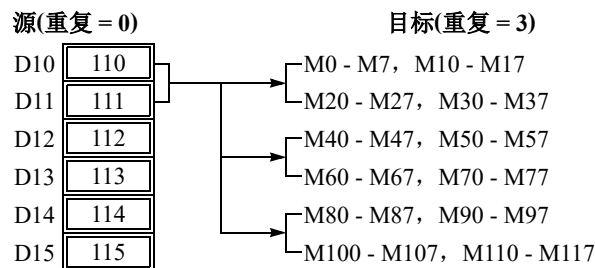
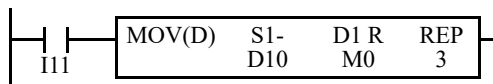
### 重复位设备

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

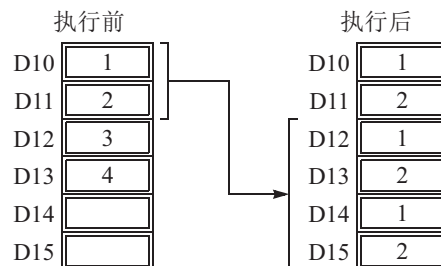
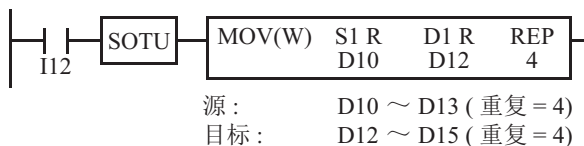


- 数据类型：双字型



### 重复所导致的重叠设备

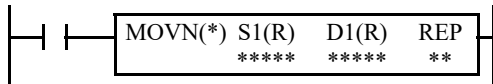
如果对源和目标都指定了重复操作，并且一部分源区域和目标区域相互重叠，则还会更改重叠区域中的源数据。



## MOVN (求反传送)

反转数据并直接移动。

FT2J FC6A



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。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

▲ 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。

### 有效数据类型

W (字型)	X
I (整型)	X
D (双字型)	X
L (长整型)	X
F (浮点型)	—

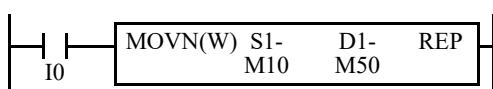
如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字型或整型数据类型) 或 32 点 (双字型或长整型数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。

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。

### 重复操作

有关设置重复次数时的操作信息，请参见 MOV 指令中的第 5-3 页上的“传送指令中的重复操作”。

### 示例: MOVN

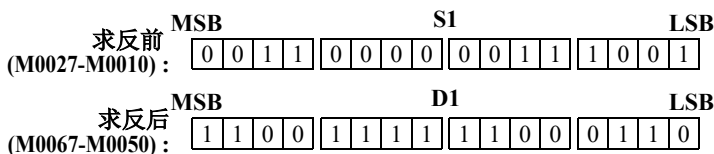


M10 NOT → M50

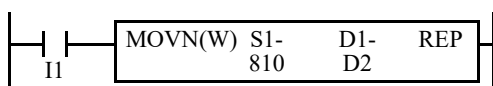
输入 I0 打开时，由源设备 S1 指定并以 M10 开始的 16 个内部继电器将被逐位求反，并传送到由目标设备 D1 指定并以 M50 开始的 16 个内部继电器。

M10 - M17, M20 - M27 NOT

→ M50 - M57, M60 - M67

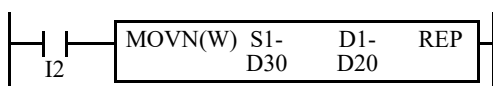
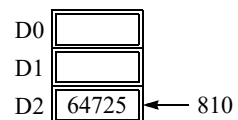
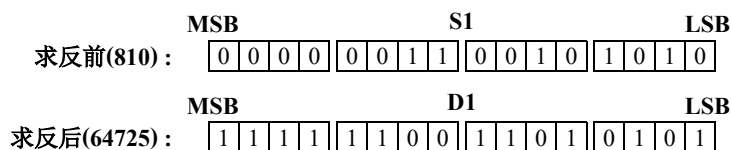


16 个内部继电器 (M10-M17 和 M20-M27) 的开 / 关状态将被求反，并传送到 16 个内部继电器 (M50-M57 和 M60-M67)。M50 是 LSB (最低有效位)，M67 是 MSB (最高有效位)。



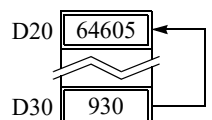
810 NOT → D2

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。

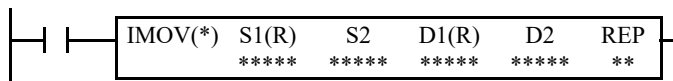


## 5: 传送指令

### IMOV (间接传送)

间接移动数据。

FT2J FC6A



$S1 + S2 \rightarrow D1 + D2$

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

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当 S1 以及 S2 指定的传输源设备或 D1 和 D2 指定的传输目标设备超出范围时，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。
- 当选择 F (浮点型) 数据类型，并且 S1 不符合标准浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

▲ 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。

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。

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。

#### 有效数据类型

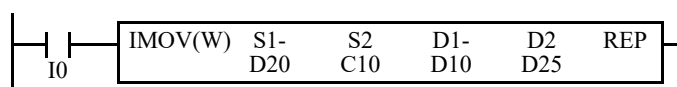
W (字型)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字数据) 或 2 点 (双字数据)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
I (整型)	—	
D (双字型)	X	
L (长整型)	—	
F (浮点型)	X	

#### 重复操作

有关设置重复次数时的操作信息，请参见 IBMV 指令中的第 5-14 页上的“间接位传送指令中的重复操作”。

## 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，则源数据由偏移值加上源设备 S1 所指定的数据寄存器 D20 来确定：

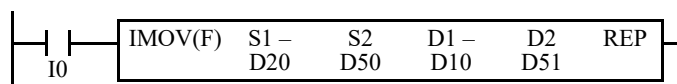
$D(20 + 4) = D24$

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

$D(10 + 20) = D30$

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

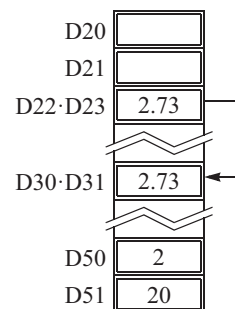
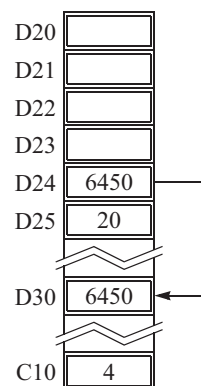


$D20 + D50 \rightarrow D10 + D51$

如果数据寄存器 D50 包含值 2，则通过偏移值加上源设备 S1 所指定的数据寄存器 D20 来确定源数据： $D(20 + 2) = D22$

如果数据寄存器 D51 包含值 20，则通过偏移值加上目标设备 D1 所指定的数据寄存器 D10 来确定目标数据： $D(10 + 20) = D30$

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。

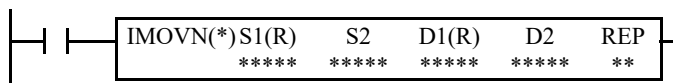


## 5: 传送指令

### IMOVN（间接求反传送）

反转数据并间接移动数据。

FT2J FC6A



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

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。

**注释：**当 S1 以及 S2 指定的传输源设备或 D1 和 D2 指定的传输目标设备超出范围时，将导致用户程序执行错误，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

▲ 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。

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。

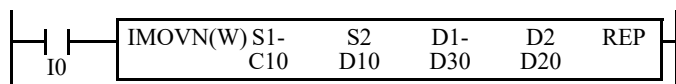
#### 有效数据类型

W (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。
I (整型)	—	
D (双字型)	X	
L (长整型)	—	
F (浮点型)	—	

#### 重复操作

有关设置重复次数时的操作信息，请参见 IBMV 指令中的第 5-14 页上的“间接位传送指令中的重复操作”。

#### 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。

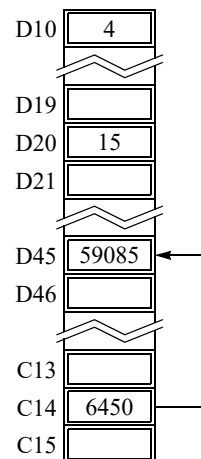
如果由源设备 S2 指定的数据寄存器 D10 的数据是 4，则通过偏移值加上由源设备 S1 指定的计数器 C10 来确定源数据：

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

如果由目标设备 D2 指定的数据寄存器 D20 包含值 15，则通过偏移值加上由目标设备 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。

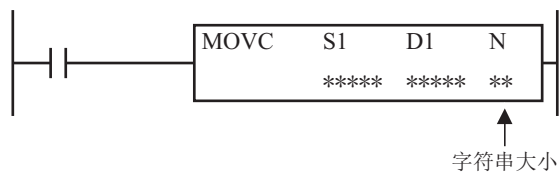


## MOVC（传送字符）

传送已指定字符集的字符串。

FT2J FC6A

### 梯形图



### 操作

输入为打开时，将在 S1 指定的字符串中附加终端字符 NULL（00h）（1 字节），并以 D1 指定的设备的高位字节为起始，连续传送该数据。

设备	存储值	
	高位字节	低位字节
D0000	'1'=31h	'2'=32h
D0001	'3'=33h	'4'=34h
D0002	00h	
D0003		

S1='1234' ⇒

### 注释:

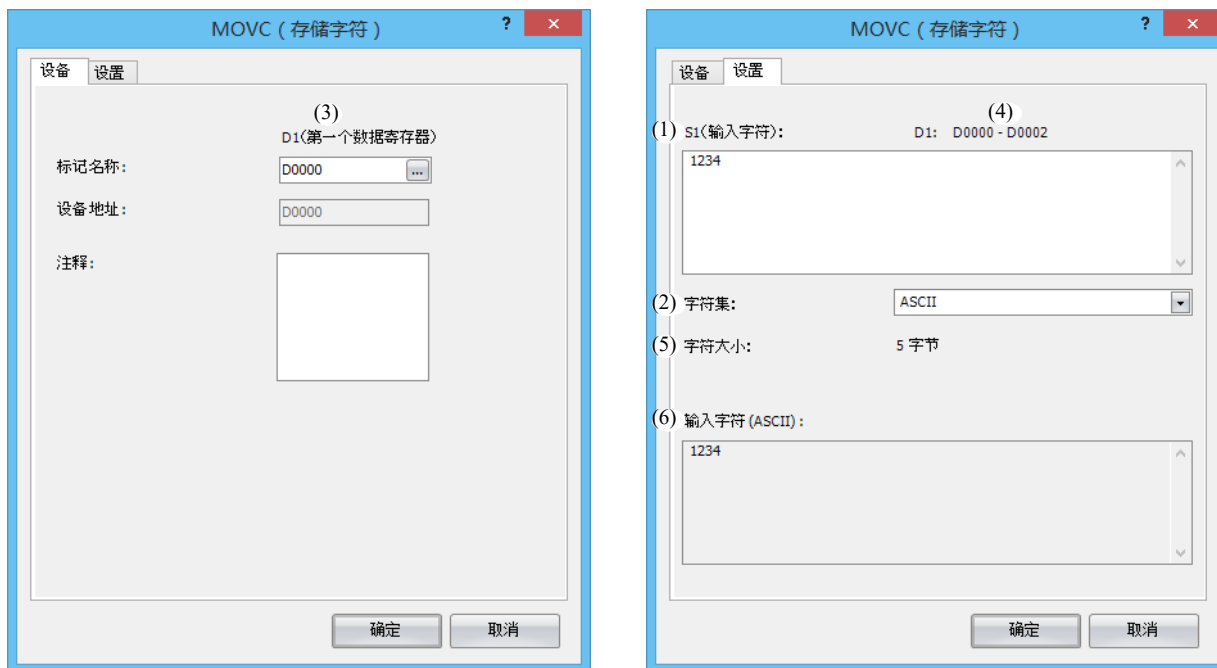
- 可在 S1 中指定最大 1,023 字节的字符串。
- 可用于向通过 EMAIL 指令的数据寄存器插入功能分配的数据寄存器设置字符串时等。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	字符串	—	—	—	—	—	—	—	—	X	—
D1（目标 1）	传送目标	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—

\*1 无法使用特殊数据寄存器。

### 设置



#### ■ S1 (源 1) 的设置

##### (1) 输入字符

以最大 1,023 字节的任意字符进行指定，全角、半角皆可。如果已换行，将插入换行代码（0D0Ah）。

##### (2) 字符集

从以下 6 种选项中选择字符串的字符集。

- ASCII
- Unicode (UTF-8)
- 日语 (ISO-2022-JP)
- 日语 (Shift-JIS)
- 中文 (GB2312)
- 西欧 (ISO 8859-1(Latin 1))

#### ■ D1 (目标 1) 的设置

##### (3) 第一个数据寄存器

指定存储字符串的数据寄存器的起始编号。

#### ■ 其他设置

##### (4) 设备范围

显示以指定的数据寄存器为起始，存储输入字符 (1) 中指定的字符串所需的数据寄存器范围。

以下情况下，不会显示数据寄存器范围：

- 输入字符 (1) 中指定的字符串字节数超过 1,023 个字节
- 第一个数据寄存器 (3) 未指定数据寄存器（特殊数据寄存器除外），或未设定第一个数据寄存器 (3)

**注释：**如果占用的数据寄存器超出范围，将显示如下信息。





**(5) 字符大小**

显示输入字符串的字节数。

显示在输入字符 (1) 中指定的字符串中附加终端字符 NULL (1 字节) 后的值。

**(6) 解码后的输入字符**

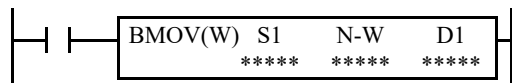
显示以字符集 (2) 中选择的字符集解码输入字符 (1) 中输入的字符串后所得的字符串。

## 5: 传送指令

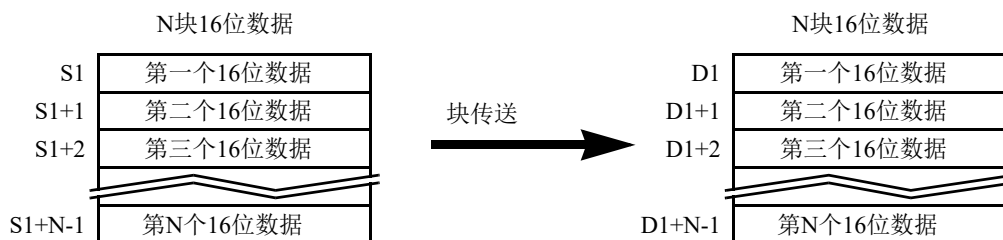
### BMOV (块传送)

批量传送数值。

FT2J FC6A



S1、S1+1、S1+2、...、S1+N-1 → D1、D1+1、D1+2、...、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。



#### 注释:

- 执行 BMOV 时，BMOV/WSFT 执行标记 (M8024) 将打开。完成后，M8024 将关闭。
- 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，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	要传送的起始设备编号	X	X	X	X	X <sup>*1</sup>	X <sup>*1</sup>	X	—	—	—
N-W (N 个字)	要传送的块数量	X	X	X	X	X <sup>*1</sup>	X <sup>*1</sup>	X	—	X <sup>*2</sup>	—
D1 (目标 1)	要传送到的起始设备编号	—	X	X <sup>*3</sup>	X	X <sup>*4</sup>	X <sup>*4</sup>	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

\*1 在 S1 或 N-W 中指定 T/C 时，将变为当前值区域。

\*2 可在 1 ~ 65,535 的范围内进行指定。

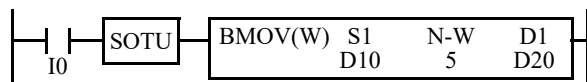
\*3 无法使用特殊内部继电器。

\*4 在 D1 中指定 T/C 时，将变为预置值区域。

#### 有效数据类型

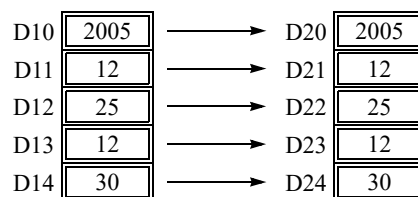
W (字型)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源、N-W 或目标，则使用 16 点 (字数据)。
I (整型)	—	
D (双字型)	—	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源、N-W 或目标，则使用 1 点 (字数据)。
L (长整型)	—	
F (浮点型)	—	

#### 示例: BMOV



D10 - D14 → D20 - D24

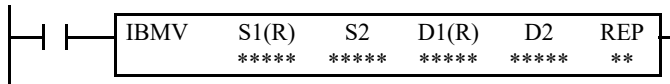
输入 I0 打开时，以源设备 S1 所指定的 D10 开始的 5 个数据寄存器的数据将传送到以目标设备 D1 所指定的 D20 开始的 5 个数据寄存器。



## IBMV (间接位传送)

按位间接传送数值。

FT2J FC6A



$S1 + S2 \rightarrow D1 + D2$

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。

**注释：**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，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	0-65,535	—
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	—	0-65,535	—

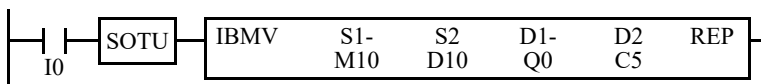
关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

▲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)。

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

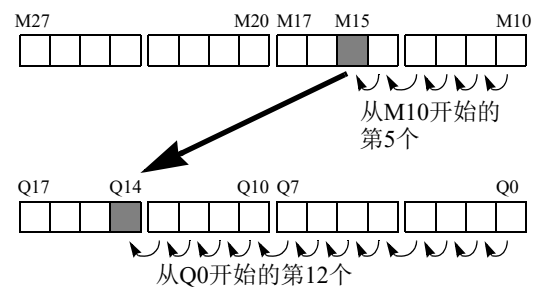


$M10 + D10 \rightarrow Q0 + C5$

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。

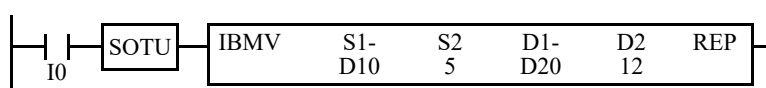
如果由源设备 S2 指定的数据寄存器 D10 的值是 5，则通过偏移值加上源设备 S1 所指定的内部继电器 M10 来确定源数据。

如果目标设备 D2 所指定的计数器 C5 的当前值是 12，则通过偏移值加上目标设备 D1 所指定的输出 Q0 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 Q14。

## 5: 传送指令

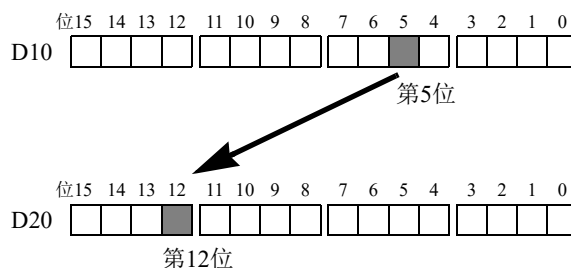


由于源设备 S1 是数据寄存器，并且源设备 S2 的值是 5，因此，源数据是由源设备 S1 所指定的数据寄存器 D10 的第 5 位。

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。

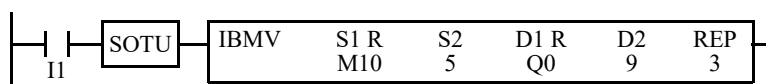
D10 + 5 → D20 + 12



### 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。



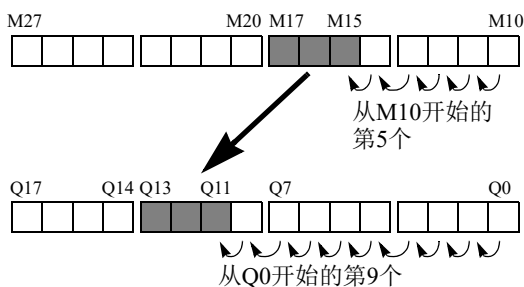
由于源设备 S1 是内部继电器 M10，并且源设备 S2 的值是 5，因此，源数据是以 M15 开始的 3 个内部继电器。

由于目标设备 D1 是输出 Q0，并且目标设备 D2 的值是 9，因此，目标数据是以 Q11 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 Q11-Q13。

M10 + 5 → Q0 + 9

重复 = 3



#### 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。



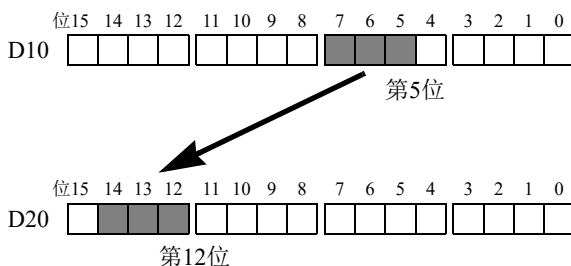
由于源设备 S1 是数据寄存器 D10，并且源设备 S2 的值是 5，因此，源数据是以数据寄存器 D10 的第 5 位开始的 3 个位。

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。

D10 + 5 → D20 + 12

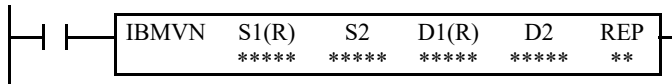
重复 = 3



## IBMVN（间接位求反传送）

数值逐位反转，间接传送。

FT2J FC6A



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

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。

**注释：**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，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	0-65,535	—
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	—	0-65,535	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

▲ 不能将特殊内部继电器指定为 D1。

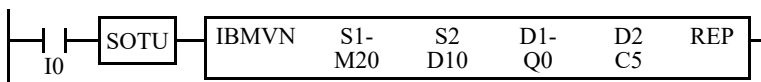
当 T（定时器）或 C（计数器）用作 S2 或 D2 时，将读取定时器 / 计数器的当前值（TC 或 CC）。

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。

### 重复操作

有关设置重复次数时的操作信息，请参见 IBMV 指令中的第 5-14 页上的“间接位传送指令中的重复操作”。

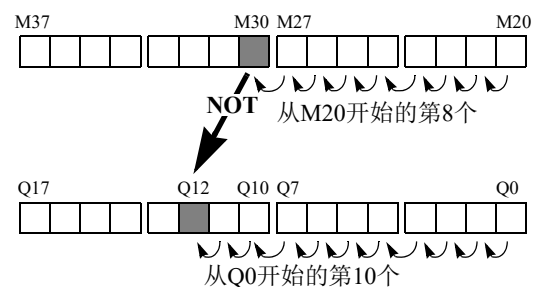
### 示例：IBMVN



$M20 + D10 \text{ NOT} \rightarrow Q0 + C5$

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 Q0 来确定目标数据。

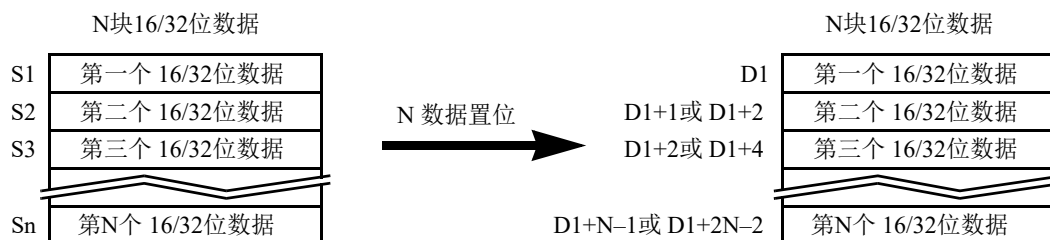
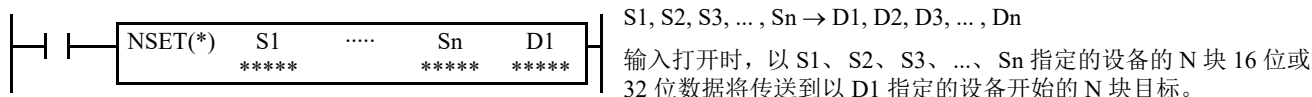
因此，当输入 I0 打开时，内部继电器 M30 的开 / 关状态将被求反，并传送到输出 Q12。

## 5: 传送指令

### NSET (N 数据置位)

分批转移单个数值。

FT2J FC6A



**注释:** 当选择 F (浮点型) 数据类型, 并且  $S_n$  不符合标准浮点格式时, 将导致用户程序执行错误, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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, 请参见第 2-1 页上的“设备地址”。

▲ 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)。

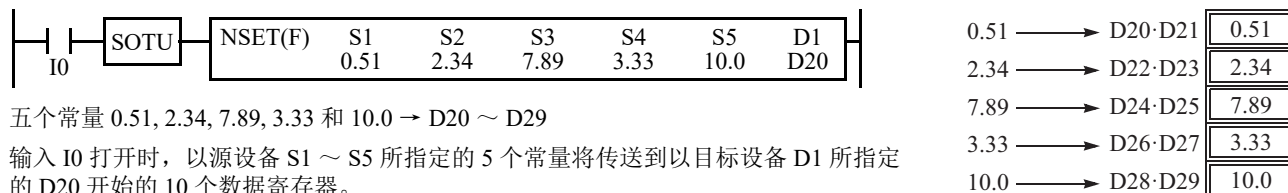
当 T (定时器) 或 C (计数器) 用作 D1, 将数据作为预置值 (TP 或 CP) 写入。

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。

#### 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: NSET(F)



五个常量 0.51, 2.34, 7.89, 3.33 和 10.0 → D20 ~ D29

输入 I0 打开时, 以源设备 S1 ~ S5 所指定的 5 个常量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。

**注释:** 在设备中存储 32 位数据的方法取决于控制器。

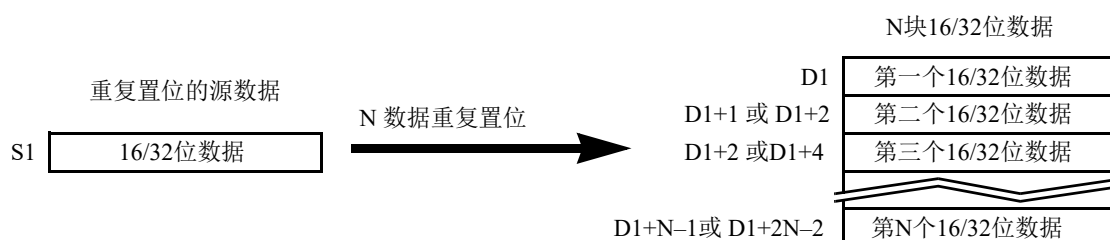
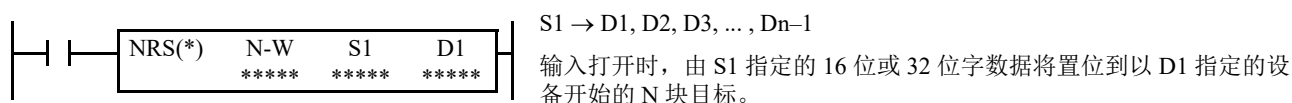
FT2J 型: 高位字存储在第一个设备。

FC6A 型: 32 位数据根据“功能设置”中选择的方法存储在设备中。关于适用的设备和指令, 请参见第 3-10 页上的“32 位数据存储”。

## NRS (N 数据重复置位)

重复传送数值。

FT2J FC6A



**注释:** 当选择 F (浮点型) 数据类型, 并且 S1 不符合标准浮点格式时, 将导致用户程序执行错误, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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, 请参见第 2-1 页上的“设备地址”。

对于 N-W, 无论是哪种数据类型, 始终使用 1 字 (16 位)。

▲ 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。

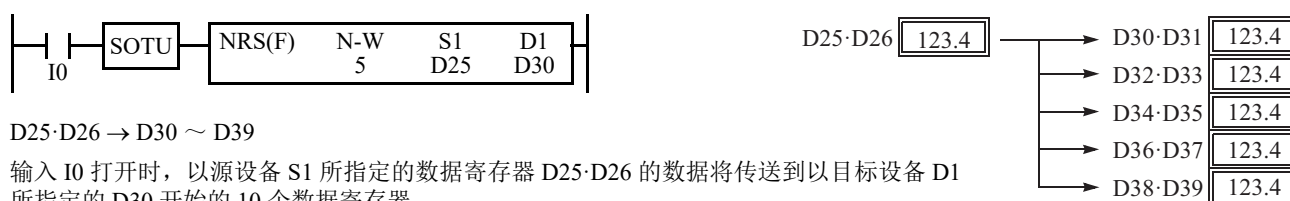
在 T (定时器) 或 C (计数器) 用作 D1 时, 数据将作为预置值 (TP 或 CP) 写入。

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。

## 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)



**注释:** 在设备中存储 32 位数据的方法取决于控制器。

FT2J 型: 高位字存储在第一个设备。

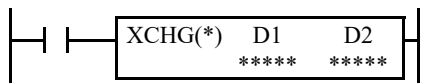
FC6A 型: 32 位数据根据“功能设置”中选择的方法存储在设备中。关于适用的设备和指令, 请参见第 3-10 页上的“32 位数据存储”。

## 5: 传送指令

### XCHG (交换)

交换两个数值。

FT2J FC6A



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

输入打开时, 由 D1 和 D2 指定的设备中的 16 位或 32 位数据将相互交换。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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, 请参见第 2-1 页上的“设备地址”。

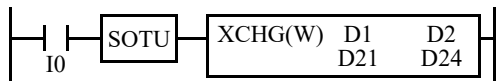
▲ 特殊内部继电器不能指定为 D1 或 D2。

#### 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) 等位设备指定为目标, 则使用 16 点 (字数据) 或 32 点 (双字数据)。
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) 或 2 点 (双字数据)。
L (长整数)	—	
F (浮点)	—	

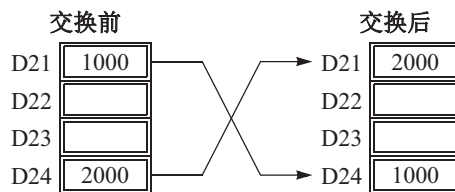
#### 示例: XCHG

##### • 数据类型: 字

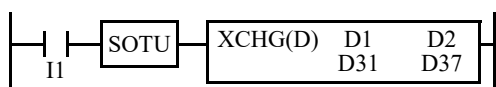


D21 ↔ D24

输入 I0 打开时, 由设备 D1 和 D2 指定的数据寄存器 D21 和 D24 的数据将相互交换。

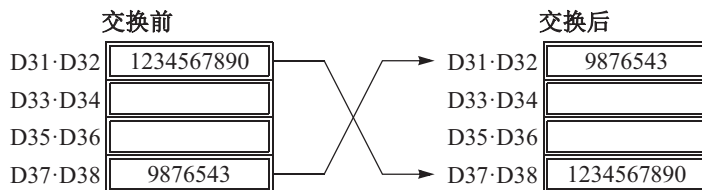


##### • 数据类型: 双字



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。



**注释:** 在设备中存储 32 位数据的方法取决于控制器。

FT2J 型: 高位字存储在第一个设备。

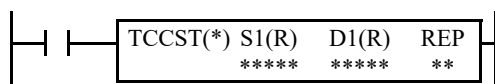
FC6A 型: 32 位数据根据“功能设置”中选择的方法存储在设备中。关于适用的设备和指令, 请参见第 3-10 页上的“32 位数据存储”。



## TCCST（存储定时器 / 计数器当前值）

将数据移至定时器或计数器的当前值。

FT2J FC6A



S1 → D1

输入打开时，将显示以 S1 指定的 16 位或 32 位数据并存储为以 D1 指定的设备的当前值。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

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) 或 32 点 (双字数据)。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	
L (长整数)	—	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) 或 2 点 (双字数据)。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。
F (浮点)	—	

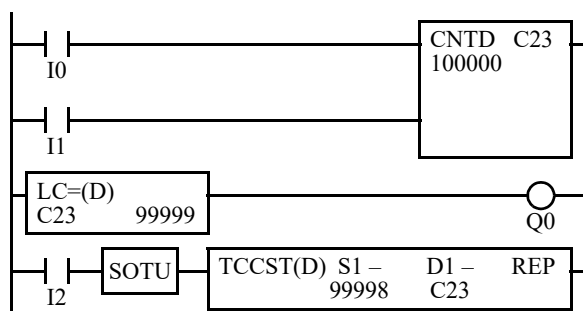
### 重复操作

有关设置重复次数时的操作信息，请参见 MOV 指令中的第 5-3 页上的“传送指令中的重复操作”。

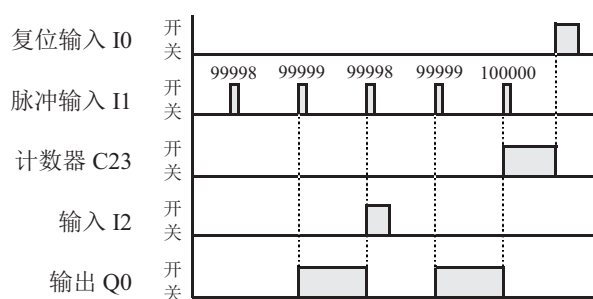
### 示例：TCCST

当打开输入 I2，99,998 将写入到计数器 C23 的当前值。

#### 梯形图程序



#### 时序图





# 6: 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。

如果启用重复操作，则可通过 AND 或 OR 运算选择重复的 CMP 指令比较结果，并且该结果会被输出至输出继电器或内部继电器。

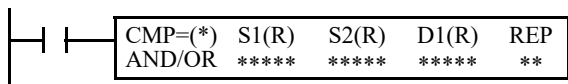
使用 ICMP>= 指令也可以比较三个值。

数据比较接点指令加载比较结果就能发出下列指令。

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)

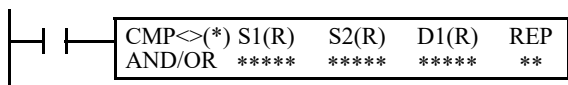
FT2J FC6A



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  
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)

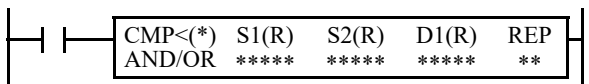
FT2J FC6A



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  
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)

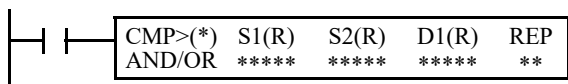
FT2J FC6A



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  
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)

FT2J FC6A

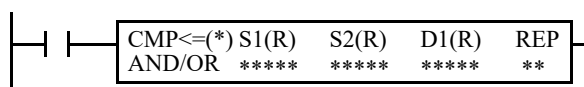


数据类型 W 或 I:  $S1 > S2 \rightarrow D1$  打开  
数据类型 D、L 或 F:  $S1 \cdot S1+1 > S2 \cdot S2+1 \rightarrow D1$  打开  
输入打开时，将比较由源设备 S1 和 S2 所指定的 16 位或 32 位数据。S1 数据大于 S2 数据时，目标设备 D1 将打开。条件不满足时，D1 将关闭。

## 6: 数据比较指令

### CMP<= (比较小于或等于)

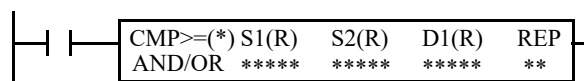
FT2J FC6A



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  
 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。

### CMP>= (比较大于或等于)

FT2J FC6A



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。

#### 注释:

- 当比较指令输入关闭时, 比较输出 (D1) 保持上次执行比较指令时的值。
- 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, 将导致用户程序执行错误, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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, 请参见第 2-1 页上的“设备地址”。

当只重复 S1 和 / 或 S2 时, 逻辑运算类型可以选择 AND 或 OR。

▲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, 将显示定时器 / 计数器的当前值 (TC 或 CC)。

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

#### 有效数据类型

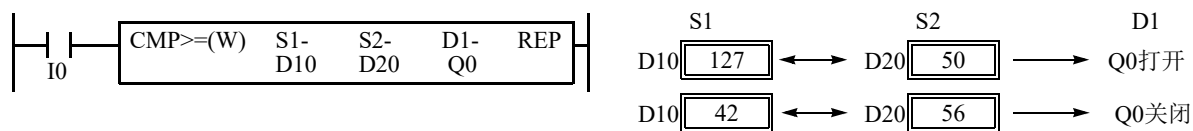
W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源, 则使用 16 点 (字或整数数据) 或 32 点 (双字或长整数数据)。如果对位设备指定重复, 则设备位的数量将以 16 点或 32 点为增量增加。
I (整数)	X	
D (双字)	X	
L (长整数)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源, 则使用 1 点 (字或整数数据) 或 2 点 (双字、长整数或浮点数据)。如果对字设备指定重复, 则设备字的数量将以 1 点或 2 点为增量增加。
F (浮点)	X	

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。

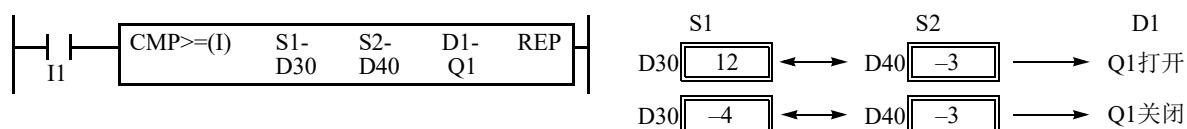
## 示例：CMP&gt;=

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。

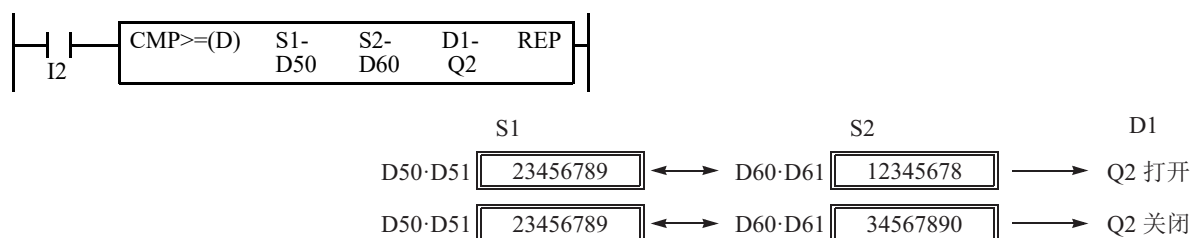
## • 数据类型：字



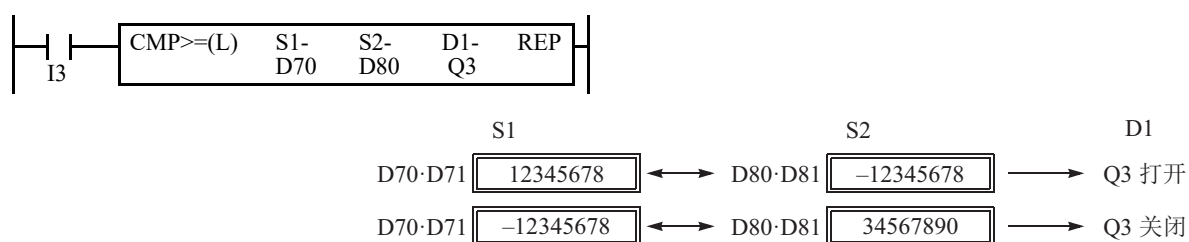
## • 数据类型：整数



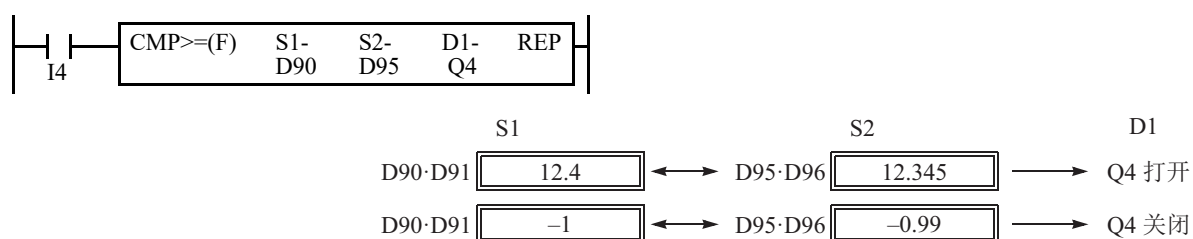
## • 数据类型：双字



## • 数据类型：长整数



## • 数据类型：浮点



## 6: 数据比较指令

### 数据比较指令中的重复操作

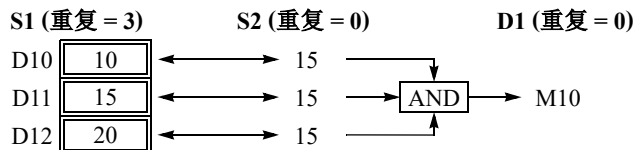
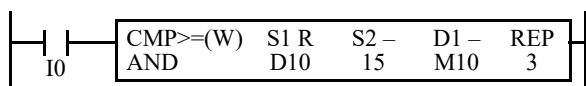
以下示例使用字型和双字型数据的 CMP>= 指令。所有其他数据比较指令和其他数据类型的重复操作均与下列相同。

如果启用重复操作，则可通过 AND 或 OR 运算选择重复的 CMP 指令比较结果，并且该结果会被输出至输出继电器或内部继电器。

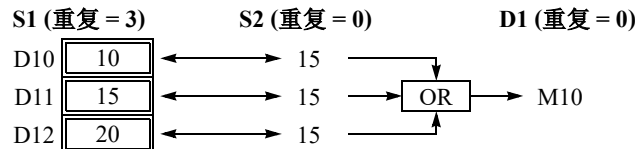
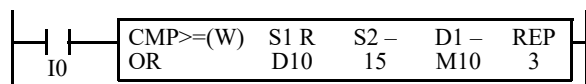
#### 重复一个源设备

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。

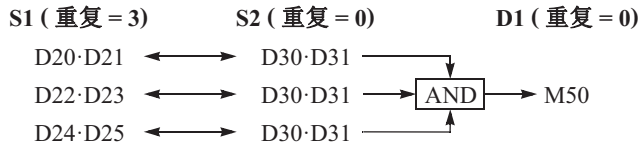
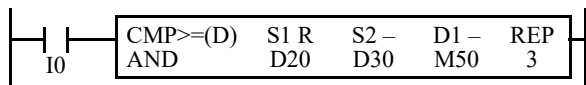
- 数据类型：字（重复逻辑操作 AND）



- 数据类型：字（重复逻辑操作 OR）



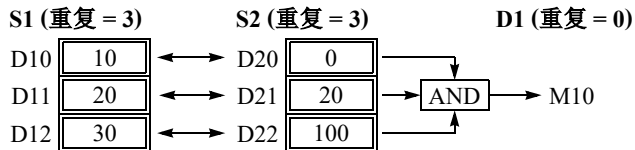
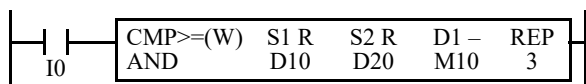
- 数据类型：双字（重复逻辑操作 AND）



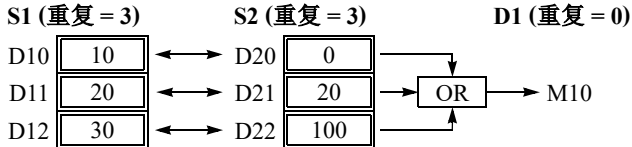
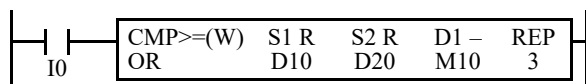
#### 重复两个源设备

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。

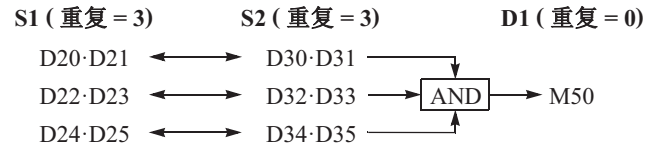
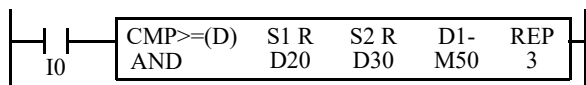
- 数据类型：字（重复逻辑操作 AND）



- 数据类型：字（重复逻辑操作 OR）



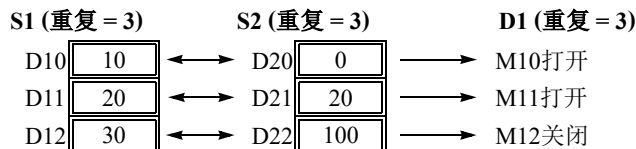
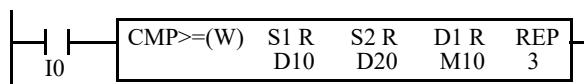
- 数据类型：双字（重复逻辑操作 AND）



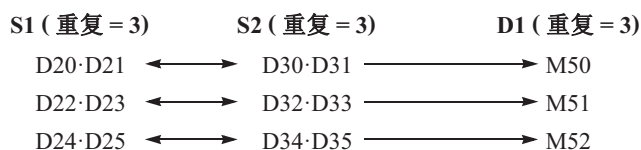
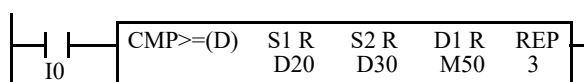
### 重复源设备和目标设备

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

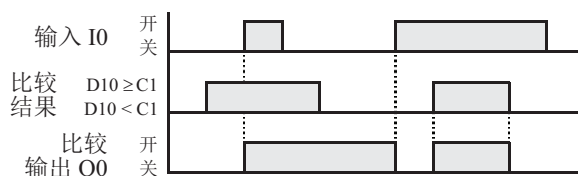
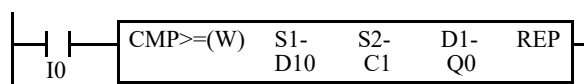


#### • 数据类型：双字

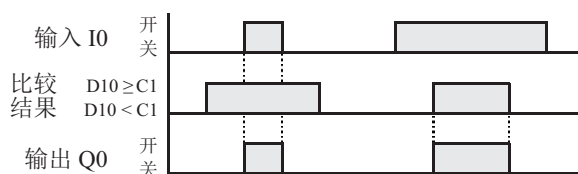
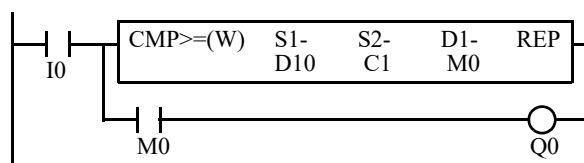


### 比较输出状态

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。



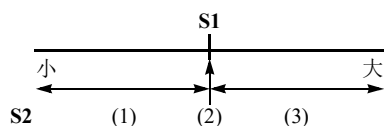
### CMP= 中的特殊内部继电器 M8150、M8151 和 M8152

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 将打开。

S1 = S2 时，M8151 将打开。

S1 < S2 时，M8152 将打开。



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

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。

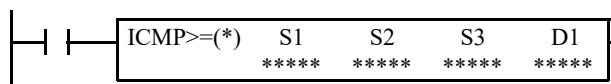
使用多个 CMP= 或 ICMP>= 指令时，M8150、M8151 或 M8152 将指示最后执行的指令的结果。

## 6: 数据比较指令

### ICMP>= (间隔比较大于或等于)

比较三个值并输出结果。

FT2J FC6A



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。

#### 注释:

- 当 ICMP>= 指令的输入关闭时, 比较输出 (D1) 保持上次执行 ICMP>= 指令时的结果。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。
- 如果 S1 的数据小于 S3 的数据 ( $S1 < S3$ ), 将存储错误代码 23 在用户程序执行错误代码 (D8006) 中。
- 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, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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, 请参见第 2-1 页上的“设备地址”。

▲ 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, 将显示定时器 / 计数器的当前值 (TC 或 CC)。

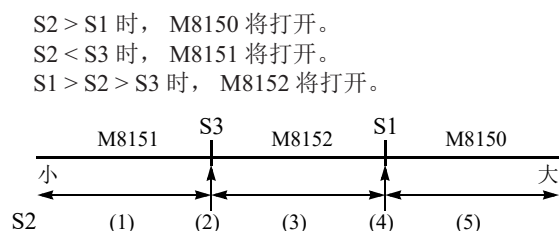
当选择 F (浮点) 数据时, 只有数据寄存器和常量可指定为 S1、S2 和 S3。

#### 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 (字或整数数据) 或 32 点 (双字或长整数数据)。
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) 或 2 点 (双字、长整数或浮点数据)。
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 ( $S1 \geq S3$ )。



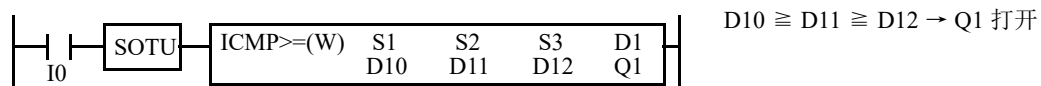
$S2 > S1$  时, M8150 将打开。  
 $S2 < S3$  时, M8151 将打开。  
 $S1 > S2 > S3$  时, M8152 将打开。

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

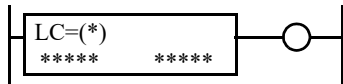
	S1		S2		S3		D1	M8150	M8151	M8152	M8004		
D10	17	>	D11	15	=	D12	15	→	Q1打开	关	关	关	关
D10	15	<	D11	18	<	D12	19	→	Q1关闭	开	开	关	开

## 6: 数据比较指令

比较两个指定数据，并根据结果打开 / 关闭。

### LC= (接点比较等于)

FT2J FC6A

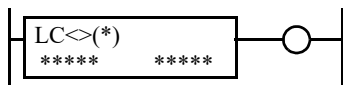


数据类型 W 或 I:  $S1 = S2$   
数据类型 D, L 或 F:  $S1 \cdot S1+1 = S2 \cdot S2+1$

该指令将不断地比较由 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)

FT2J FC6A

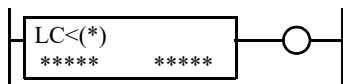


数据类型 W 或 I:  $S1 \neq S2$   
数据类型 D, L 或 F:  $S1 \cdot S1+1 \neq S2 \cdot S2+1$

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)

FT2J FC6A

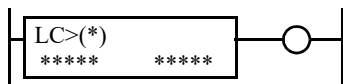


数据类型 W 或 I:  $S1 < S2$   
数据类型 D, L 或 F:  $S1 \cdot S1+1 < S2 \cdot S2+1$

该指令将不断地比较由 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)

FT2J FC6A

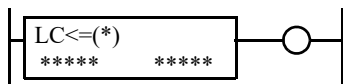


数据类型 W 或 I:  $S1 > S2$   
数据类型 D, L 或 F:  $S1 \cdot S1+1 > S2 \cdot S2+1$

该指令将不断地比较由 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)

FT2J FC6A

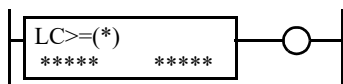


数据类型 W 或 I:  $S1 \leq S2$   
数据类型 D, L 或 F:  $S1 \cdot S1+1 \leq S2 \cdot S2+1$

该指令将不断地比较由 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)

FT2J FC6A



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

该指令将不断地比较由 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。

**注释:** 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，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

在使用 T（定时器）或 C（计数器）时，将显示定时器 / 计数器的当前值（TC 或 CC）。

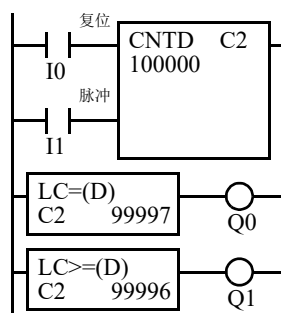
当被选择为 F（浮点型）数据类型时，只能指定数据寄存器和常量。

## 有效数据类型

W (字)	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（字或整数数据）或 32 点（双字或长整数数据）。
I (整数)	X	
D (双字)	X	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（字或整数数据）或 2 点（双字、长整数或浮点数据）。
L (长整数)	X	
F (浮点)	X	

## 示例：LC

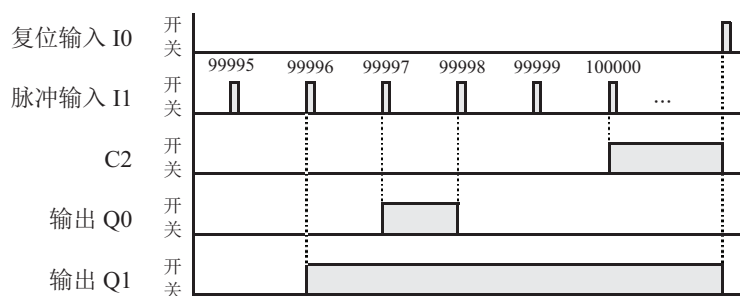
梯形图 1



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

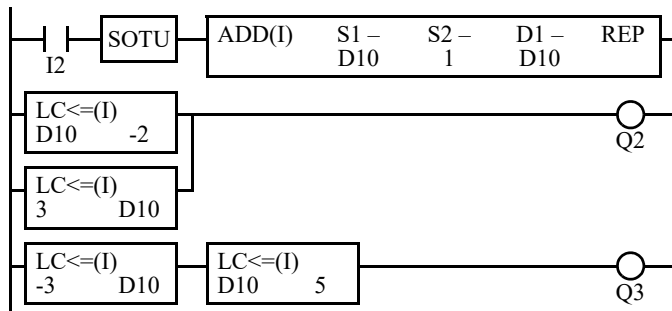


当计数器 C2 当前值为 99,997，打开输出 Q0。

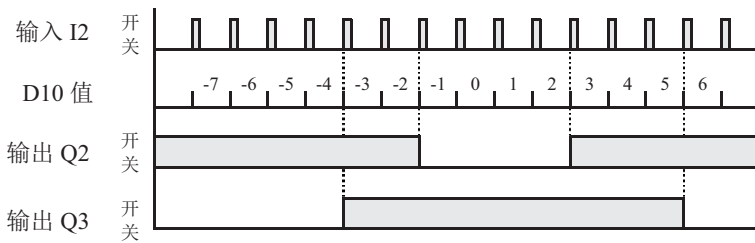
当计数器 C2 当前值到达 99,996 时，打开输入 Q1，并保留到计数器 C2 被复位时为止。

## 6: 数据比较指令

梯形图 2



时序图



程序列表

指令	数据
LOD	I2
SOTU	
ADD(I)	D10
	1
	D10
LC<=(I)	D10
	-2
LC<=(I)	3
	D10
ORLOD	
OUT	Q2
LC<=(I)	-3
	D10
LC<=(I)	D10
	5
ANDLOD	
OUT	Q3

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。

# 7: 四则运算指令

## 简介

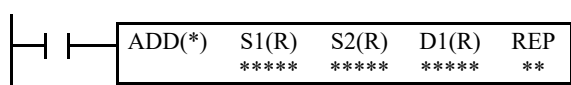
本章将对根据指定数据进行运算，并将其结果存储到设备中的算术运算指令进行介绍。对于加和减设备，将用内部继电器 M8003 来进位或借位。

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。

## ADD (加法)

添加指定值。

FT2J FC6A



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

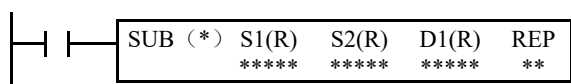
数据类型 D、L 或 F:  $S1 \cdot S1+1 + S2 \cdot S2+1 \rightarrow D1 \cdot D1+1, CY$

输入打开时，将源设备 S1 和 S2 所指定的 16 位或 32 位数据相加。结果将被设置到目标设备 D1 和内部继电器 M8003（进位或借位）。

## SUB (减法)

减去指定值。

FT2J FC6A



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

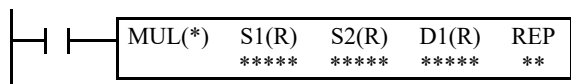
数据类型 D、L 或 F:  $S1 \cdot S1+1 - S2 \cdot S2+1 \rightarrow D1 \cdot D1+1, BW$

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 和内部继电器 M8003（进位或借位）。

## MUL (乘法)

乘以指定值。

FT2J FC6A



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

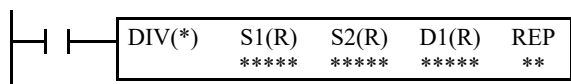
数据类型 D、L 或 F:  $S1 \cdot S1+1 \times S2 \cdot S2+1 \rightarrow D1 \cdot D1+1$

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。

## DIV (除法)

除以指定值。

FT2J FC6A



数据类型 W 或 I:  $S1 \div S2 \rightarrow D1$  (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)

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。

## 7: 四则运算指令

**注释:** 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

- 当结果超出数据类型范围时, 将存储错误代码 2 在用户程序执行错误代码 (D8006) 中。
- 当 S2 为 0 (除以 0) 时, DIV 指令的操作如下。在这种情况下, 将存储错误代码 3 在用户程序执行错误代码 (D8006) 中。
  - 当选择 F (浮点型) 数据类型时, D1 中将存储无穷大 ( $\infty$ )。
  - 如果数据类型不是 F (浮点数), S1 的值存储在 D1 中。
- 当结果超出数据类型范围时, 将存储错误代码 3 在用户程序执行错误代码 (D8006) 中。
  - 选择 I (整型) 数据类型时, S1 为 -32,768, S2 为 -1, 结果为 32,768, 超出数据类型范围。
  - 选择 L (长整) 数据类型时, S1 为 -2,147,483,648, S2 为 -1, 结果为 2,147,483,648, 超出了数据类型范围。
- 当选择 F (浮点型) 数据类型, 并且 S1 或 S2 不符合正常浮点格式时, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

### 有效设备

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

关于有效设备编号范围, 请参见第 2-1 页上的“设备地址”。

▲ 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, 将显示定时器 / 计数器的当前值 (TC 或 CC)。

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 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。

P (间接指定寄存器) 仅可于数据类型为 L (长整) 时使用。

### 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 (字或整数数据) 或 32 点 (双字、长整数或浮点数据)。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	
L (长整数)	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, 则使用 1 点 (字或整数数据) 或 2 点 (双字、长整数或浮点数据)。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。
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。

#### ADD (加法)

数据类型	进位 / 借位	D1 中存储的数据
W (字型)	进位	从计算结果中减去 65,536 后得到的值将被存储。
	借位	不发生借位。
I (整型)	进位	从计算结果中减去 32,768 后得到的值将被存储。
	借位	将计算结果加上 32,768 后得到的值存储起来。
D (双字型)	进位	从计算结果中减去 4,294,967,296 后得到的值将被存储。
	借位	不发生借位。
L (长整型)	进位	从计算结果中减去 2,147,483,648 后得到的值将被存储。
	借位	将计算结果加上 2,147,483,648 后得到的值存储起来。
F (浮点型)	进位	存储计算结果。
	借位	

## SUB (减法)

数据类型	进位 / 借位	D1 中存储的数据
W (字型)	进位	不发生进位。
	借位	将计算结果加上 65,536 后得到的值存储起来。
I (整型)	进位	从计算结果中减去 32,768 后得到的值将被存储。
	借位	将计算结果加上 32,768 后得到的值存储起来。
D (双字型)	进位	不发生进位。
	借位	将计算结果加上 4,294,967,296 后得到的值存储起来。
L (长整型)	进位	从计算结果中减去 2,147,483,648 后得到的值将被存储。
	借位	将计算结果加上 2,147,483,648 后得到的值存储起来。
F (浮点型)	进位	存储计算结果。
	借位	

## MUL (乘法)

数据类型	进位 / 借位	D1 中存储的数据
W (字型)	进位	不发生。
	借位	
I (整型)	进位	不发生。
	借位	
D (双字型)	进位	计算结果被存储，用户程序执行错误。
	借位	
L (长整型)	进位	计算结果被存储，用户程序执行错误。
	借位	
F (浮点型)	进位	存储计算结果。
	借位	

## DIV (除法)

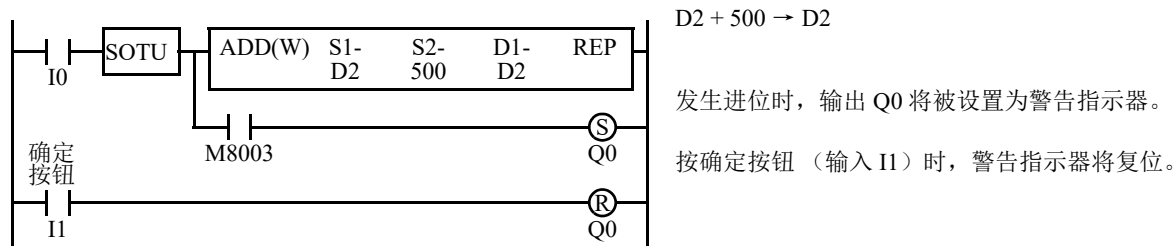
数据类型	进位 / 借位	D1 中存储的数据
W (字型)	进位	不发生。
	借位	
I (整型)	进位	不发生。
	借位	
D (双字型)	进位	不发生。
	借位	
L (长整型)	进位	不发生。
	借位	
F (浮点型)	进位	存储计算结果。
	借位	

## 7: 四则运算指令

### 示例：ADD

#### • 数据类型：字

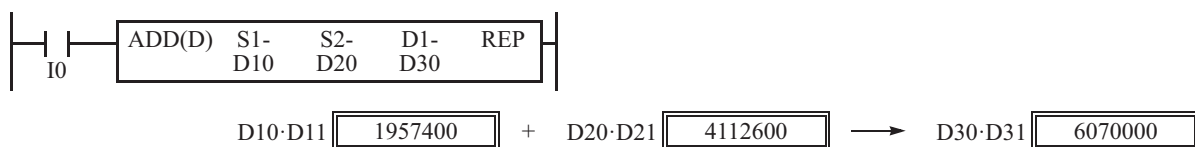
该示例演示了如何使用来自特殊内部继电器 M8003 的进位信号来设置报警信号。



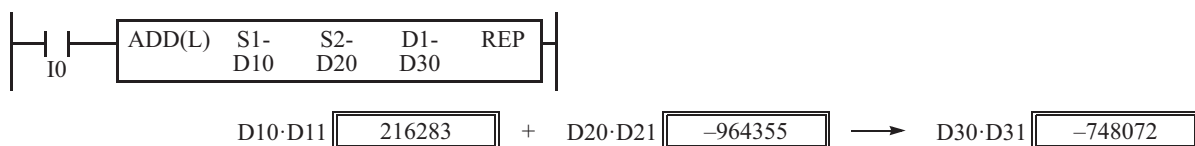
#### • 数据类型：整数



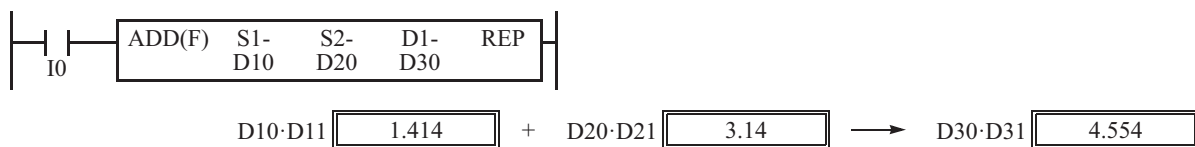
#### • 数据类型：双字



#### • 数据类型：长整数



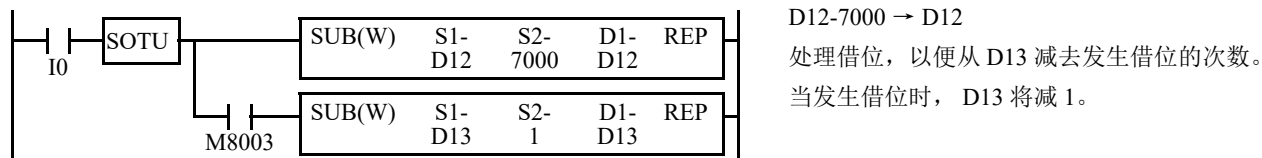
#### • 数据类型：浮点



### 示例：SUB

#### • 数据类型：字

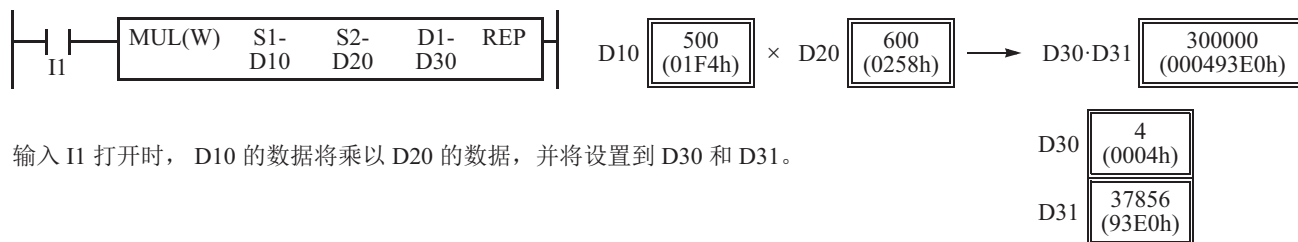
以下示例演示了如何使用特殊内部继电器 M8003 来处理借位。



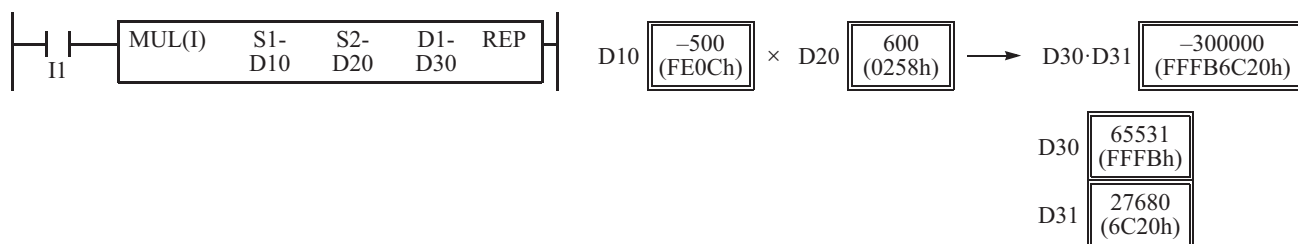


## 示例：MUL

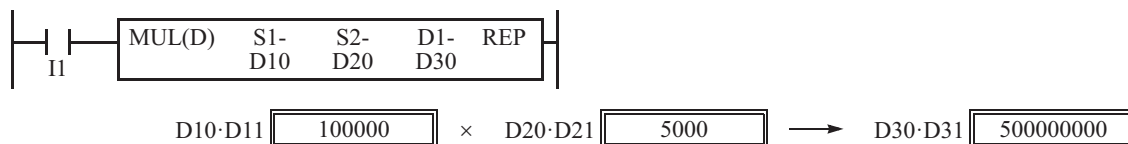
## • 数据类型：字



## • 数据类型：整数

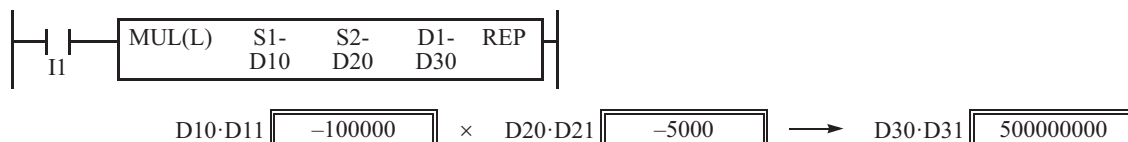


## • 数据类型：双字



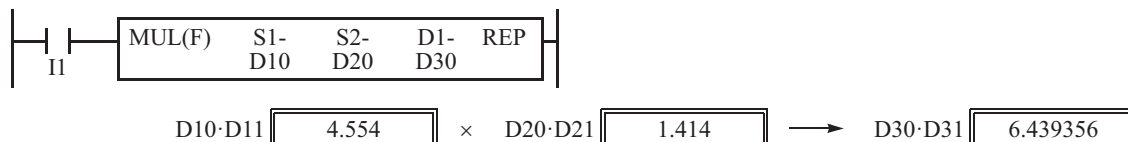
**注释：**在双字数据类型相乘中，结果的低位 32 位数据被设置到目标设备 D1·D1+1。

## • 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

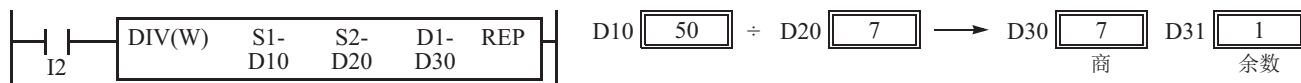


**注释：**在乘运算中目标使用两个字设备。当使用位设备（如用于目的地的内部继电器）时，需要 32 个内部继电器。设置目的设备时，应确保所使用的设备在设备范围内。

## 7: 四则运算指令

### 示例：DIV

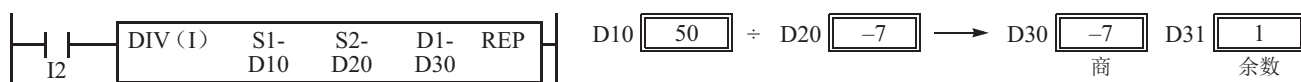
#### • 数据类型：字



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。

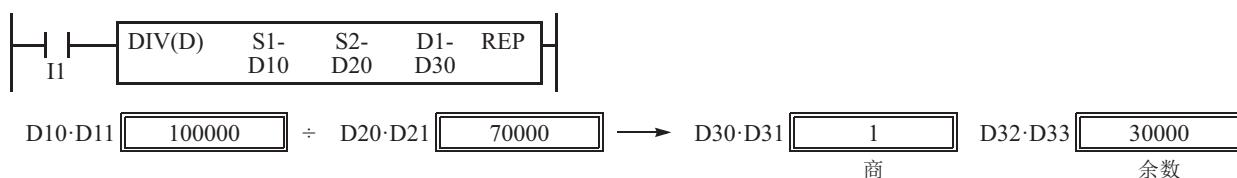
**注释：**在字数据类型除运算中目标使用两个字设备。当使用位设备（如用于目的地的内部继电器）时，需要 32 个内部继电器。设置目的设备时，应确保所使用的设备在设备范围内。

#### • 数据类型：整数



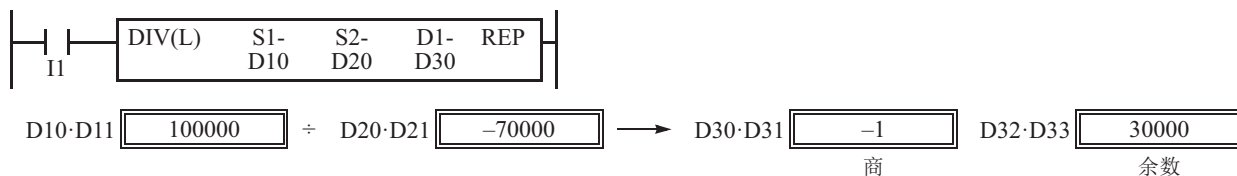
**注释：**在整数数据类型除运算中目标使用两个字设备。当使用位设备（如用于目的地的内部继电器）时，需要 32 个内部继电器。设置目的设备时，应确保所使用的设备在设备范围内。

#### • 数据类型：双字



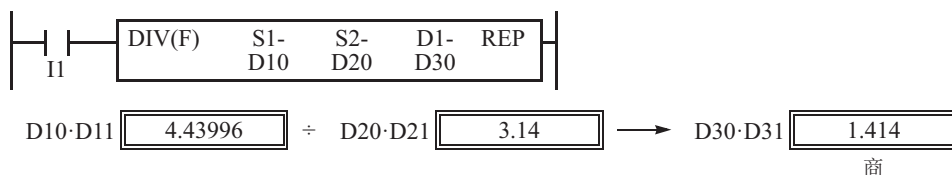
**注释：**在双字数据类型除运算中目标使用四个字设备。当使用位设备（如用于目的地的内部继电器）时，需要 64 个内部继电器。设置目的设备时，应确保所使用的设备在设备范围内。

#### • 数据类型：长整数



**注释：**在长整数数据类型除运算中目标使用四个字设备。当使用位设备（如用于目的地的内部继电器）时，需要 64 个内部继电器。设置目的设备时，应确保所使用的设备在设备范围内。

#### • 数据类型：浮点



**注释：**在浮点数据类型除运算中目标使用两个字设备。设置目的设备时，应确保所使用的设备在设备范围内。

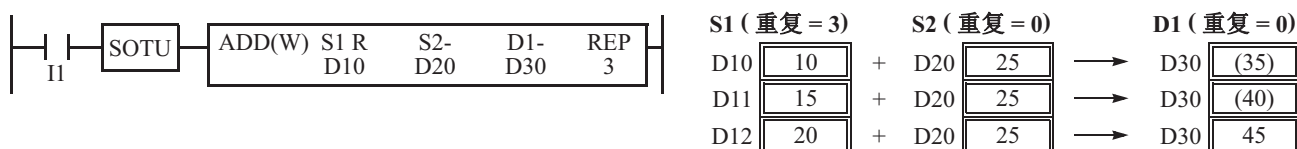
## ADD 和 SUB 指令中的重复操作

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

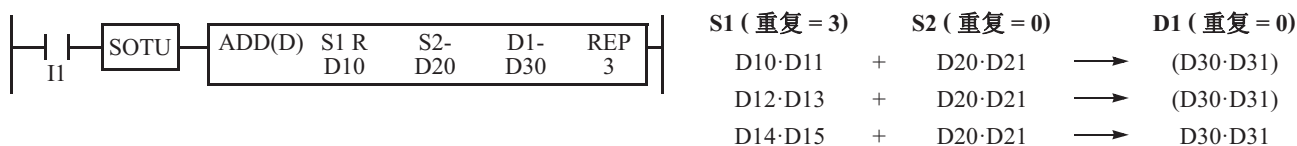
- 数据类型：字和整数

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

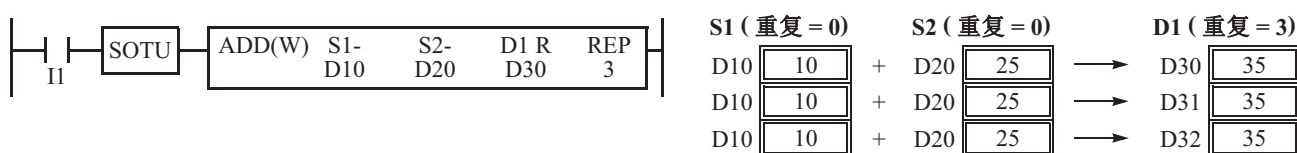
仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



### 仅重复目标设备

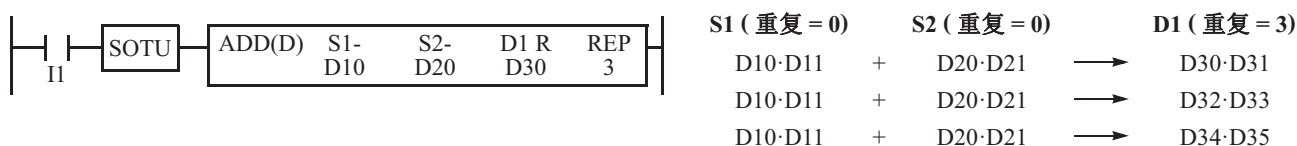
- 数据类型：字和整数

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

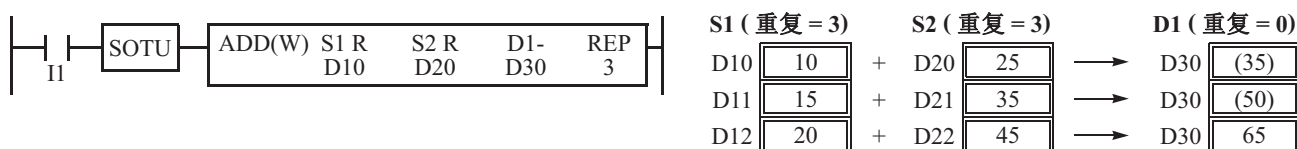
仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 3 个设备。



### 重复两个源设备

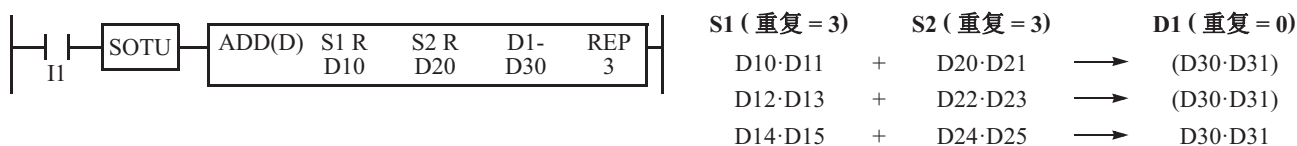
- 数据类型：字和整数

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

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。

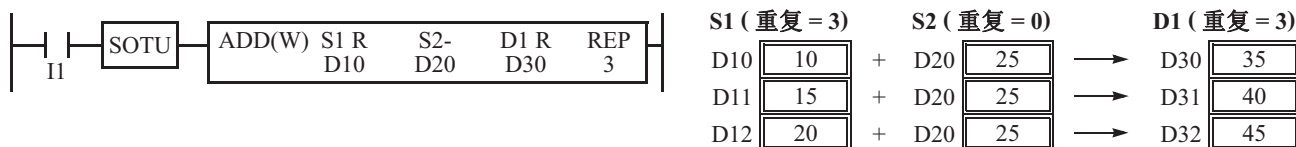


## 7: 四则运算指令

### 重复源设备和目标设备

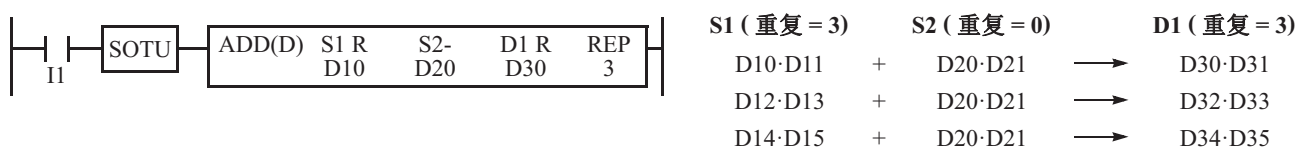
- 数据类型: 字和整数

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

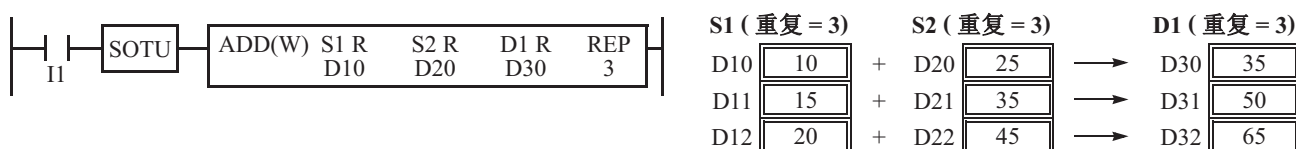
指定 S1 (源) 和 D1 (目标) 重复时, 不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



### 重复所有源设备和目标设备

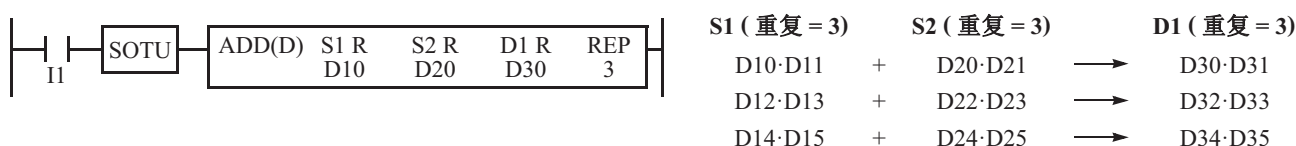
- 数据类型: 字和整数

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

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) 将打开, 并在继续执行其他指令的操作时保持。

## MUL 指令中的重复操作

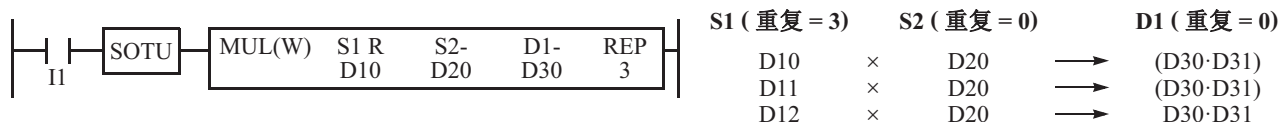
由于 MUL（乘）指令使用了两个目标设备，因此，结果将存储到下面描述的目标设备。可以单独或组合地将源设备 S1、S2 和目标设备 D1 指定为重复。未指定目标设备 D1 重复时，最后的结果将设置到目标设备 D1 和 D1+1。未指定目标设备 D1 重复时，最后的结果将设置到目标设备 D1 和 D1+1。

由于在字和整数数据上重复操作的模式是相似的，因此使用字数据来描述以下示例。

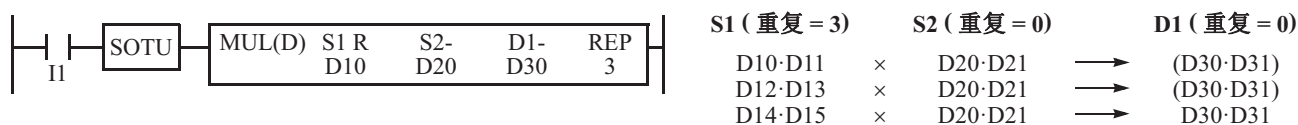
### 重复一个源设备

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



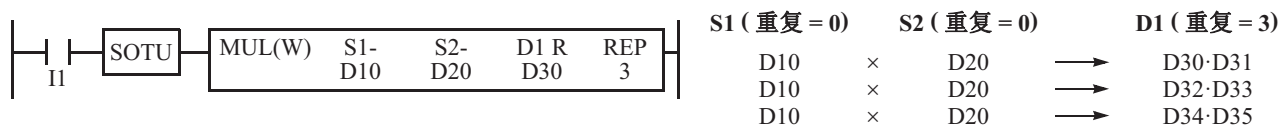
- 数据类型：双字、长整数和浮点



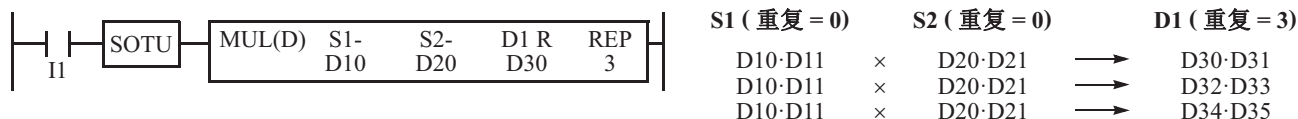
### 仅重复目标设备

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



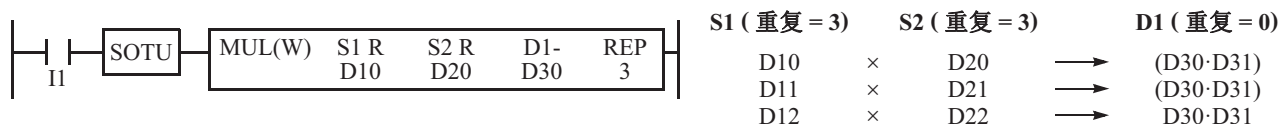
- 数据类型：双字、长整数和浮点



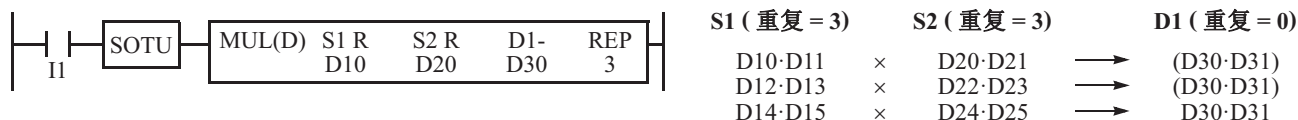
### 重复两个源设备

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

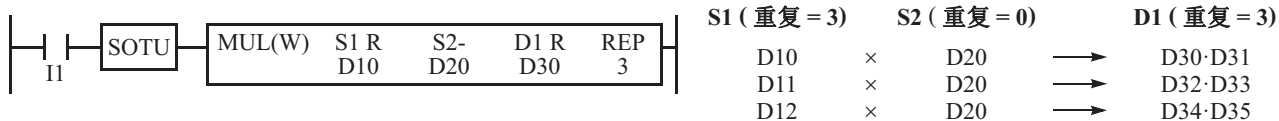


## 7: 四则运算指令

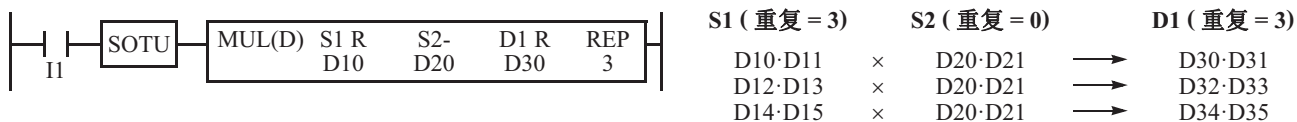
### 重复源设备和目标设备

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



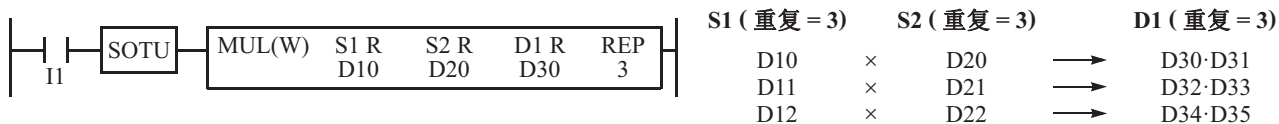
- 数据类型：双字、长整数和浮点



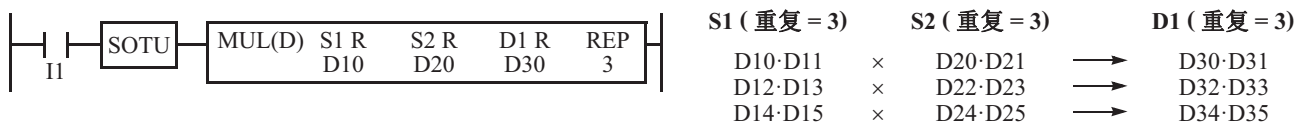
### 重复所有源设备和目标设备

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



## DIV 指令中的重复操作

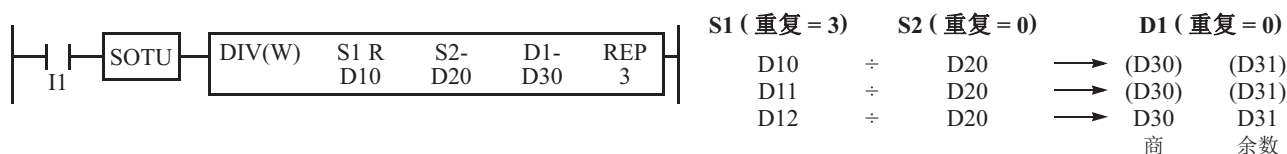
由于 DIV（除）指令（浮点数据类型除外）使用两个目标设备，因此商和余数按下面所述进行存储。可以单独或组合地将源设备 S1、S2 和目标设备 D1 指定为重复。未指定目标设备 D1 重复时，最后的结果将设置到目标设备 D1（商）和 D1+1（余数）。如果指定了重复，将使用以指定的设备开始并与重复次数一样多的连续设备。

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

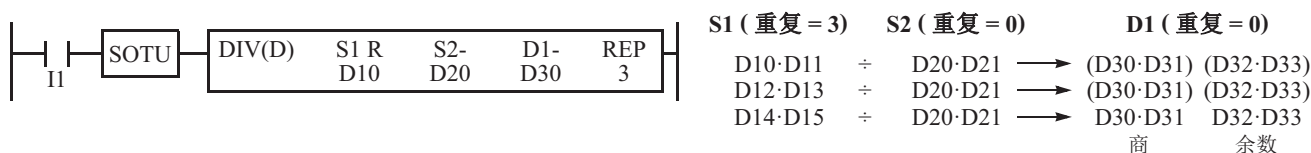
#### • 数据类型：字和整数

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。



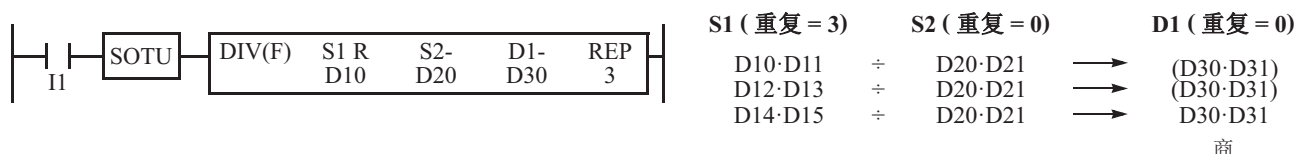
#### • 数据类型：双字和长整数

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

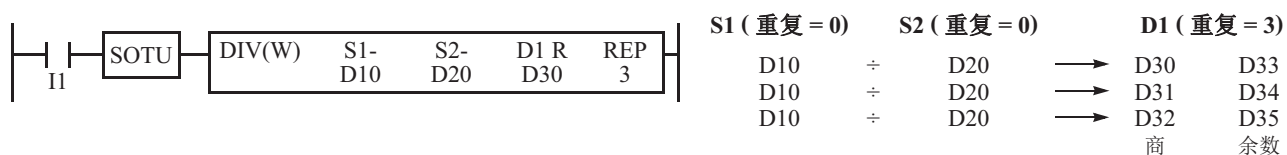
仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



### 仅重复目标设备

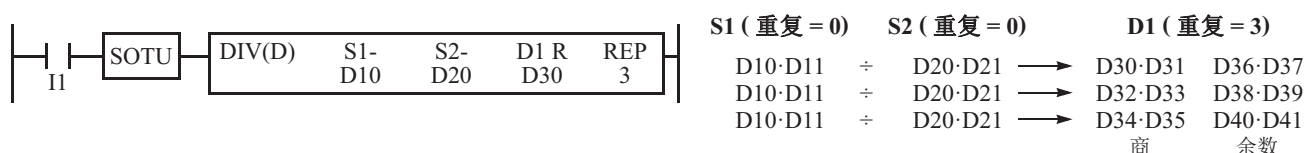
#### • 数据类型：字和整数

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。



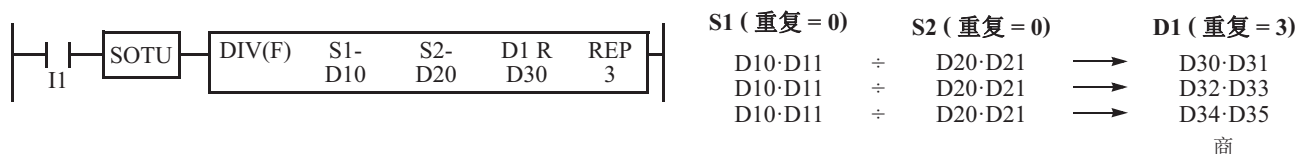
#### • 数据类型：双字和长整数

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。

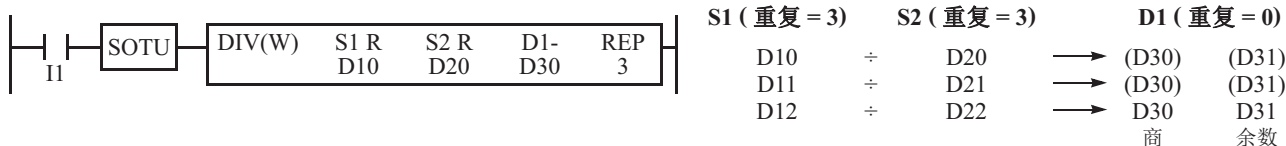


## 7: 四则运算指令

### 重复两个源设备

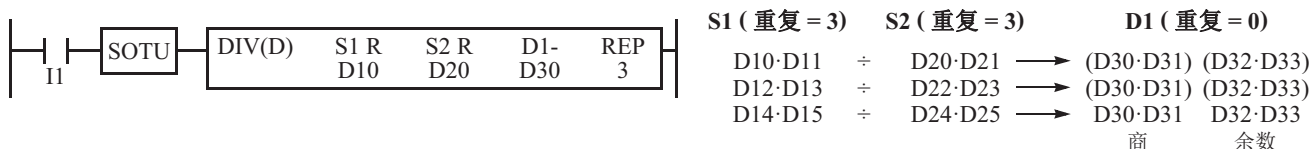
#### • 数据类型：字和整数

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。



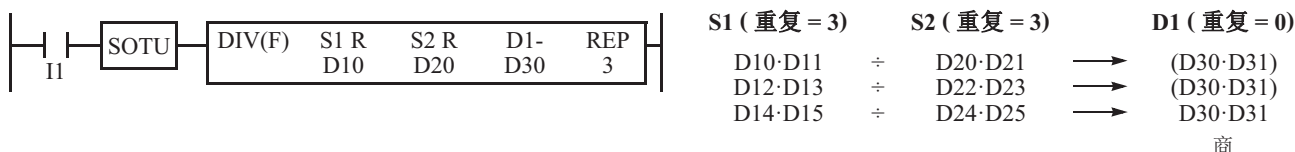
#### • 数据类型：双字和长整数

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

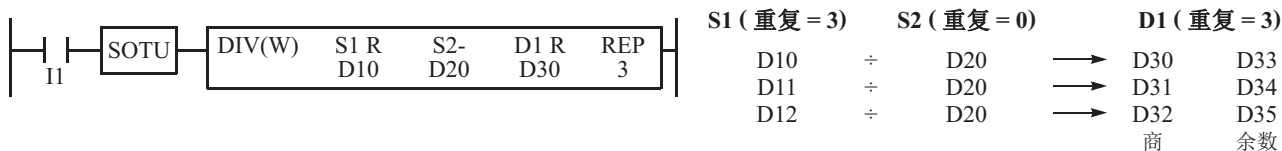
当指定 S1 和 S2（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



### 重复源设备和目标设备

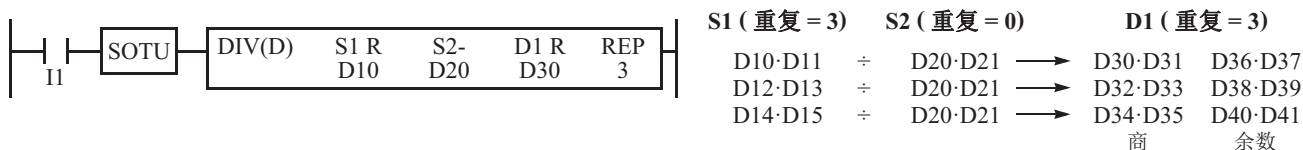
#### • 数据类型：字和整数

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。



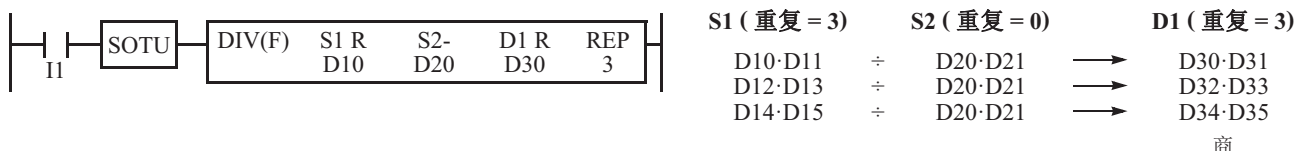
#### • 数据类型：双字和长整数

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。

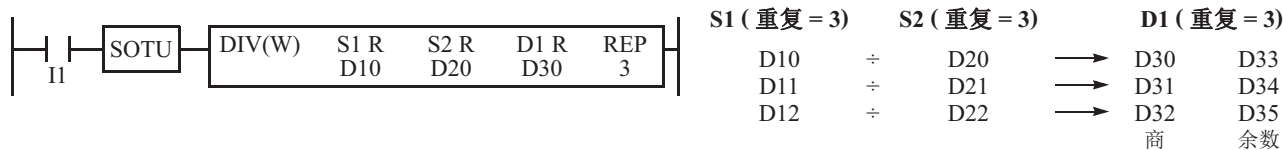




### 重复所有源设备和目标设备

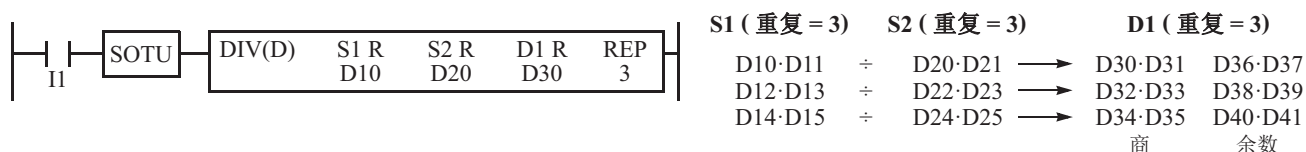
- 数据类型：字和整数

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。



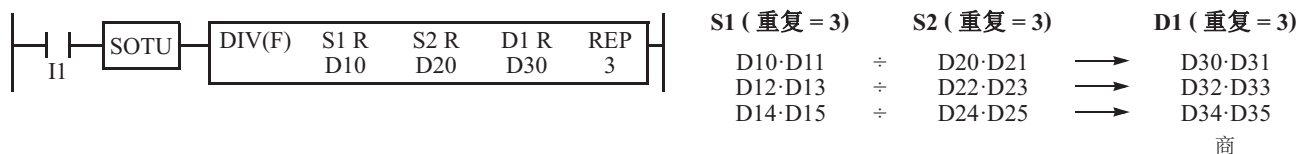
- 数据类型：双字和长整数

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。



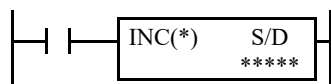
**注释：**在任何重复操作中发生用户程序执行错误时，特殊内部继电器 M8004（用户程序执行错误）将打开，并在继续执行其他指令的操作时保持。

## 7: 四则运算指令

### INC（递增）

递增 (+1) 指定设备的值。

FT2J FC6A



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。

### DEC（递减）

递减 (-1) 指定设备的值。

FT2J FC6A



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 或 32 位数据减去 1，并将结果存储到相同的设备中。

#### 有效设备

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

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

#### 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) 或 2 点 (双字或长整数数据)。
I (整数)	X	
D (双字)	X	
L (长整数)	X	
F (浮点)	—	

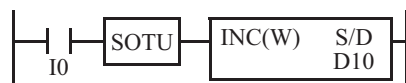
#### 超出递增的最大值

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 的值为最小值并被递减 1 时，S/D 的值将返回到该值的最大值 (字或双字数据) 或减 1 (整数或长整数数据)，并打开内部继电器 M8003 (进位或借位)。

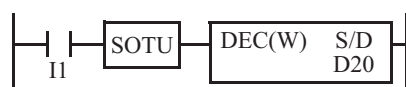
#### 示例: INC



$$D10 \boxed{100} + 1 \longrightarrow D10 \boxed{101}$$

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



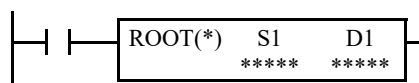
$$D20 \boxed{100} - 1 \longrightarrow D20 \boxed{99}$$

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。

## ROOT (平方根)

计算指定值的平方根。

FT2J FC6A



数据类型 W:  $\sqrt{S1} \rightarrow D1$

当输入打开时, 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。

**注释:** 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

- 当选择 F (浮点) 数据, 并且源设备 S1 含一个负值时, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。
- 当选择 F (浮点) 数据, 并且 S1 不符合标准浮点格式时, 将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

### 有效设备

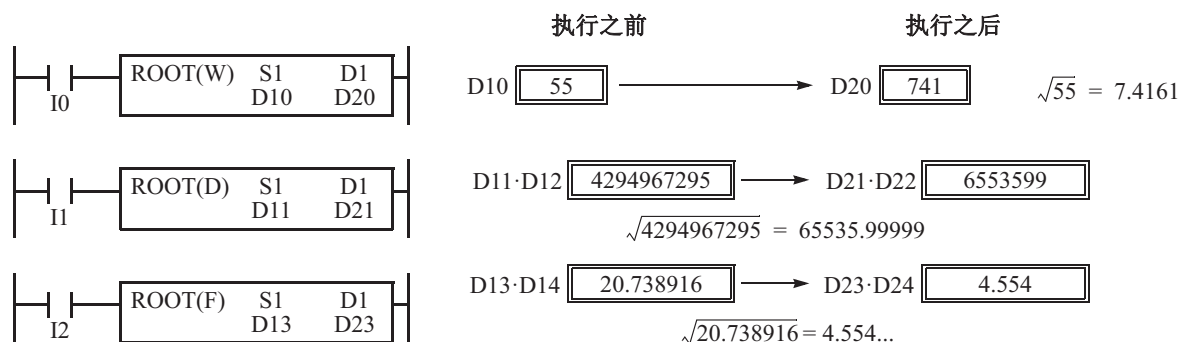
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	二进制数据	—	—	—	—	—	—	X	—	X	—
D1 (目标 1)	用于存储结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围, 请参见第 2-1 页上的“设备地址”。

### 有效数据类型

W (字)	X	如果将 D (数据寄存器) 这样的字设备指定为源或目标, 将使用 1 点 (字数据) 或 2 点 (双字或浮点数据)。
I (整数)	—	
D (双字)	X	
L (长整数)	—	
F (浮点)	X	

### 示例: ROOT

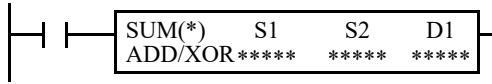


## 7: 四则运算指令

### SUM (合计)

计算指定值的合计。

FT2J FC6A



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 N 块 16 位或 32 位数据，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 N 块 16 位数据，并将其结果保存到由 D1 指定的设备。S2 指定数据块的数量。

**注释:** 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当 S2 为 0 或由 S1 + (S2) 确定的源数据超出有效设备范围时，将存储错误代码 35 在用户程序执行错误代码 (D8006) 中。
- 当选择 F (浮点) 数据，并且 S1 不符合标准浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

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，只有数据寄存器可指定为 S1。

源 S2 无视数据总是使用一个字。

#### 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。

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) 或 2 点 (双字、长整数或浮点数据)

#### 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，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	-

## 进位和借位

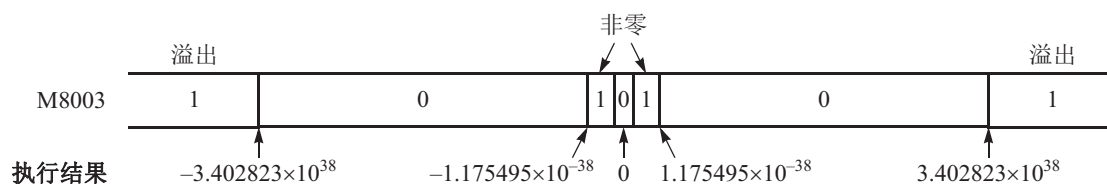
在含 D（双字）、L（长整数）或 F（浮点）数据的高级指令中，当指令的执行结果如下表所示时，将打开特殊内部继电器 M8003（进位和借位）。

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

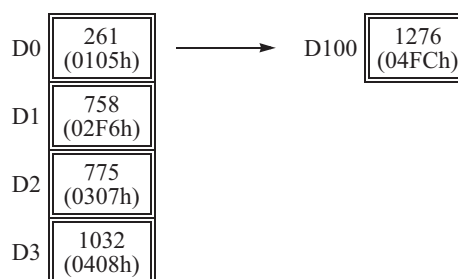
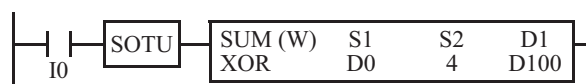
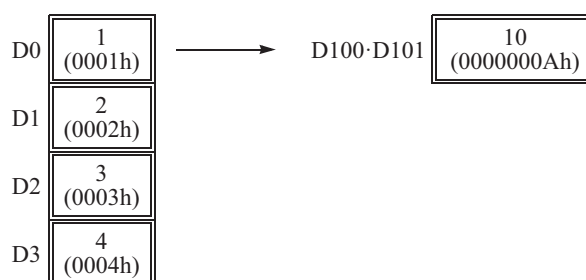
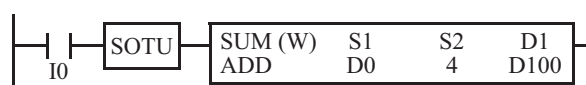
当执行含浮点型数据的高级指令时，将更新特殊内部继电器 M8003（进位和借位）。

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

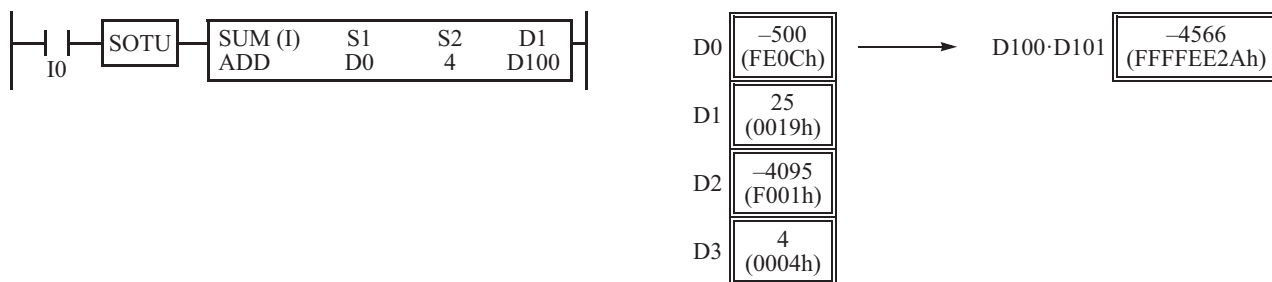


## XOR 运算

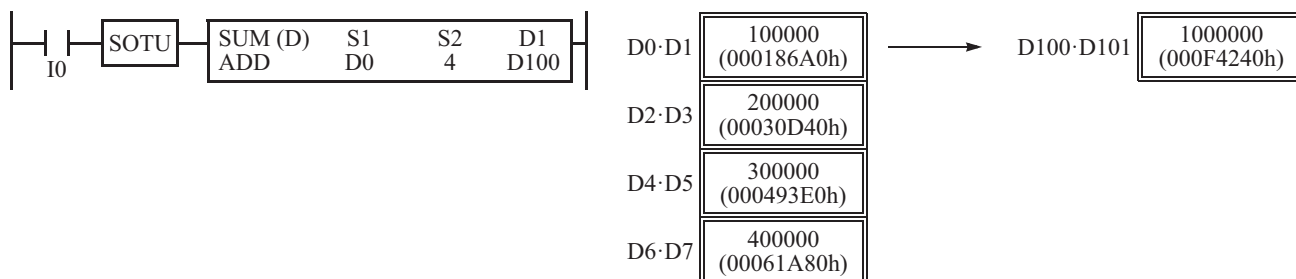
	D0	(0105h)	0000 0001 0000 0101
	D1	(02F6h)	0000 0010 1111 0110
	D2	(0307h)	0000 0011 0000 0111
XOR	D3	(0408h)	0000 0100 0000 1000
	D100	(04FCh)	0000 0100 1111 1100

## 7: 四则运算指令

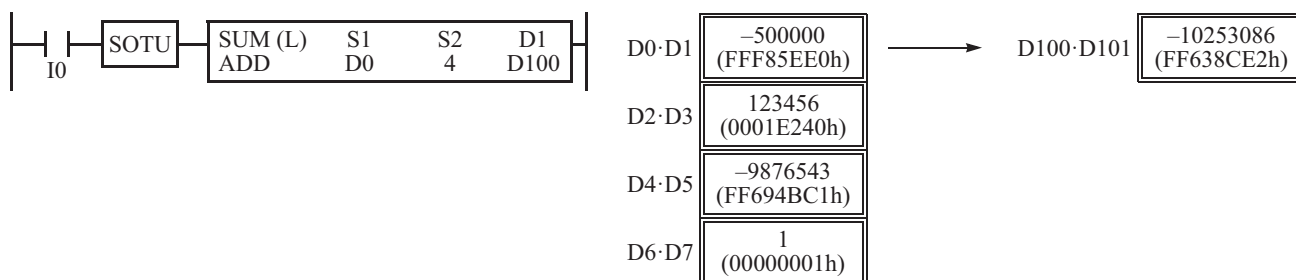
### • 数据类型：整数



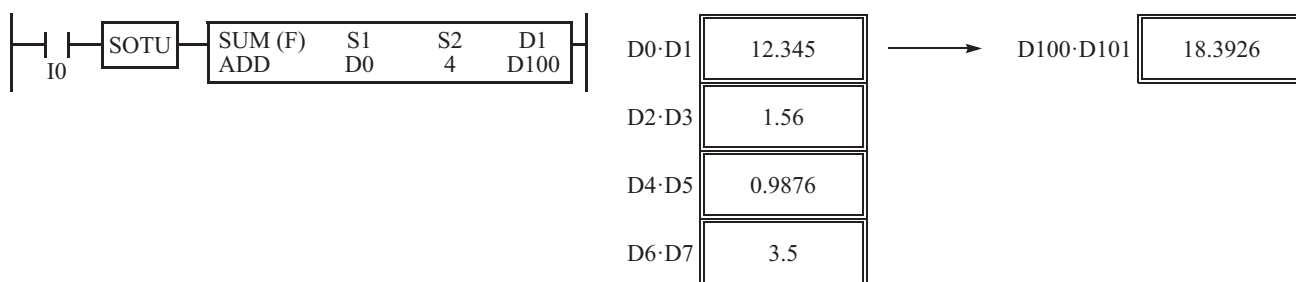
### • 数据类型：双字



### • 数据类型：长整数



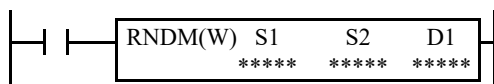
### • 数据类型：浮点



## RNDM ( 随机 )

生成伪随机数。

FT2J FC6A



生成伪随机数值。

输入打开时，将生成以 S1 中指定的数据为下限值、S2 中指定的数据为上限值的伪随机数值，并将该数值存储到 D1 中。

结果存储在由设备 D1 指定的目标中。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- S1、S2 的数据超出 0 至 32,767 范围时，将存储错误代码 33 在用户程序执行错误代码 (D8006) 中。
- S1 中指定的数据大于 S2 中指定的数据时，将存储错误代码 33 在用户程序执行错误代码 (D8006) 中。

### 有效设备

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

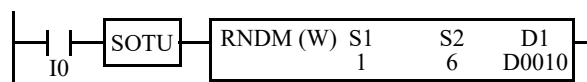
\*1 可输入 0 至 32,767 间的数值。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 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，则使用 1 点 ( 字 )。
I ( 整数 )	-	
D ( 双字 )	-	
L ( 长整数 )	-	
F ( 浮点 )	-	

### 示例 : RNDM



输入打开时，将生成 1 至 6 范围内的伪随机数值，并将结果存储到 D0010 中。





# 8: 逻辑运算指令

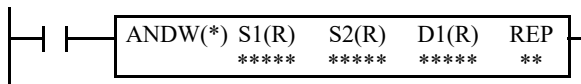
## 简介

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

## ANDW（与）

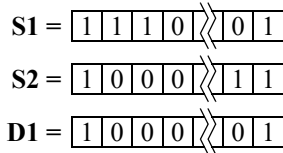
对指定值执行布尔运算。

FT2J FC6A



$$S1 \cdot S2 \rightarrow D1$$

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。



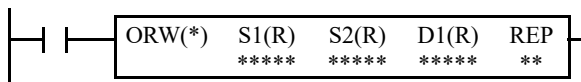
逻辑 AND 的真值表

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

## ORW（或）

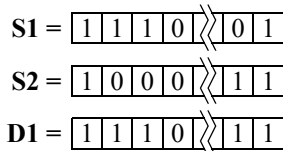
对指定值执行逻辑 OR 运算。

FT2J FC6A



$$S1 + S2 \rightarrow D1$$

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。



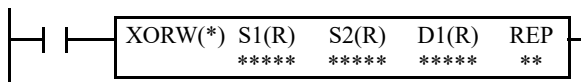
逻辑 AND 的真值表

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

## XORW（异或）

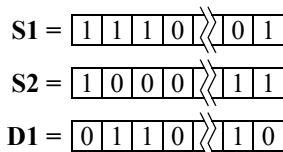
对指定值执行排他或操作。

FT2J FC6A



$$S1 \oplus S2 \rightarrow D1$$

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。



逻辑 AND 的真值表

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

## 8: 逻辑运算指令

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-1 页上的“设备地址”。

▲ 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。

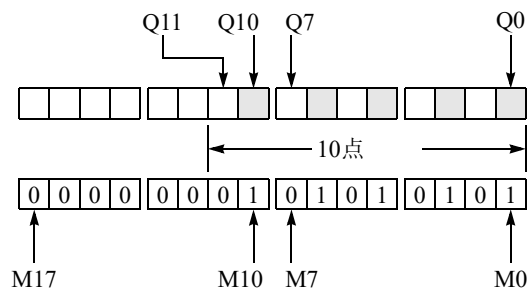
在 T (定时器) 或 C (计数器) 用作 D1 时，数据将作为预置值 (TP 或 CP) 写入，此值介于 0 到 65,535 之间。

### 有效数据类型

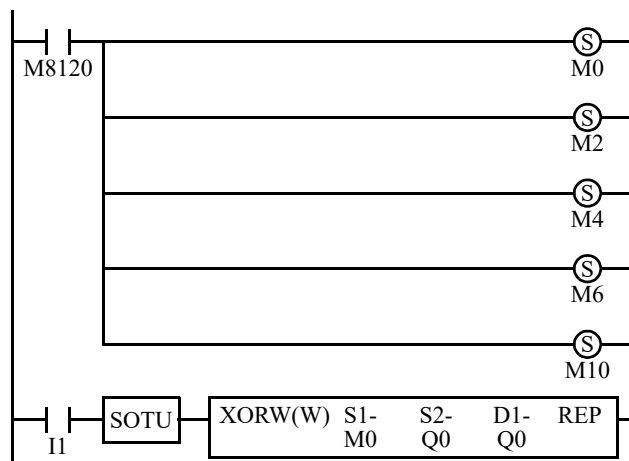
W (字)	X	如果将 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标，则使用 16 点 (字数据类型) 或 32 点 (双字数据类型)。如果对位设备指定重复，则设备位的数量将以 16 点或 32 点为增量增加。
I (整数)	—	
D (双字)	X	
L (长整数)	—	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标，则使用 1 点 (字数据类型) 或 2 点 (双字数据类型)。如果对字设备指定重复，则设备字的数量将以 1 点或 2 点为增量增加。
F (浮点)	—	

### 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。

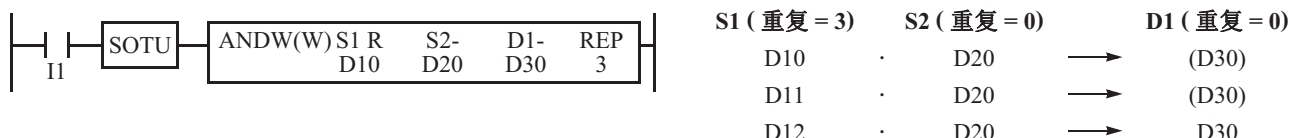
### ANDW、ORW 和 XORW 指令中的重复操作

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

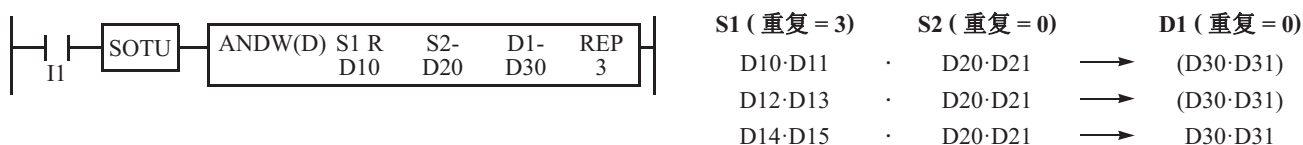
##### • 数据类型：字

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

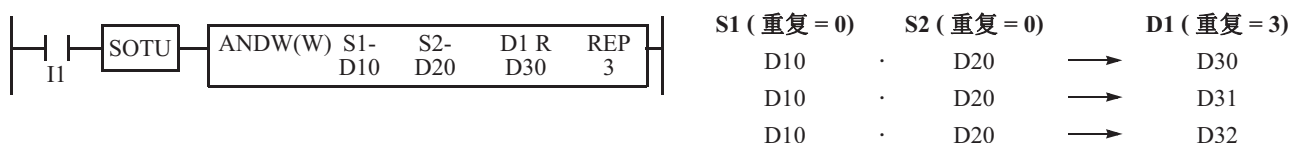
仅指定 S1（源）重复时，最后的结果将设置到目标设备 D1·D1+1。



#### 仅重复目标设备

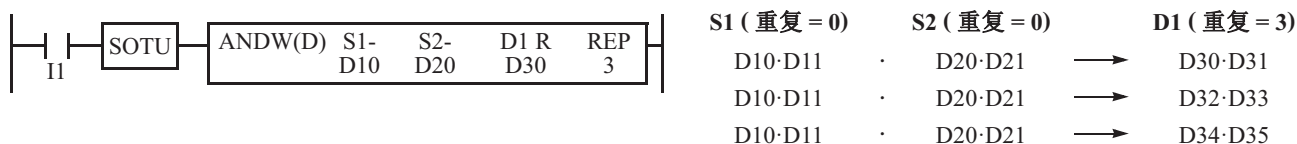
##### • 数据类型：字

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

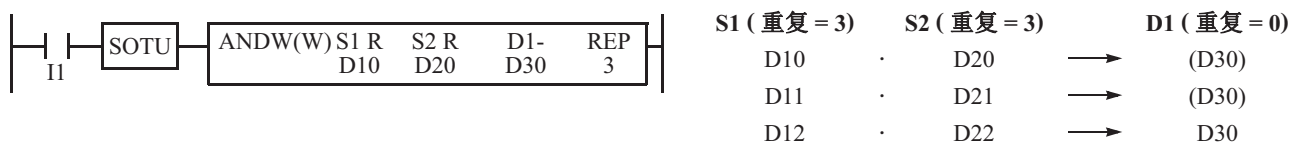
仅指定 D1（目标）重复时，相同的结果将设置到以 D1·D1+1 开始的 3 个设备。



#### 重复两个源设备

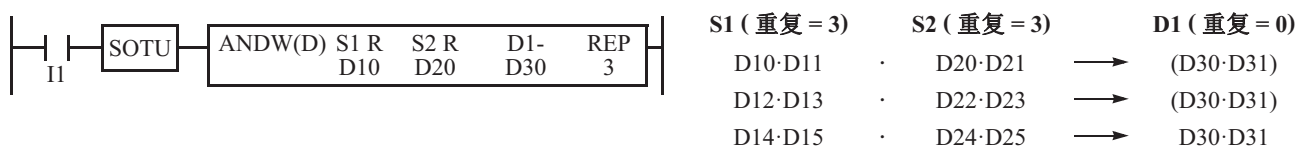
##### • 数据类型：字

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。

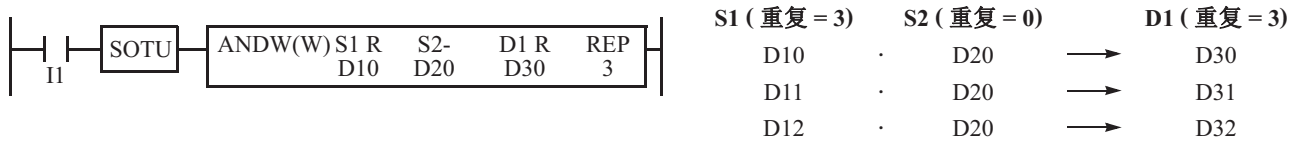


## 8: 逻辑运算指令

### 重复源设备和目标设备

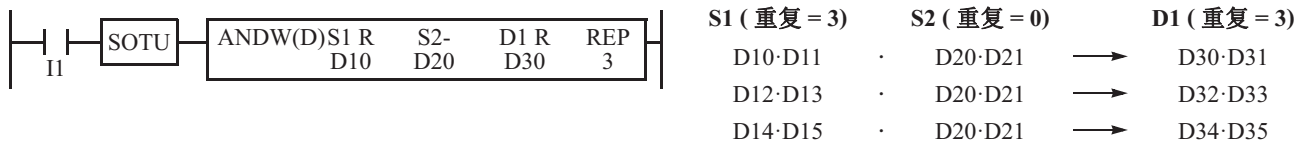
• 数据类型：字

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

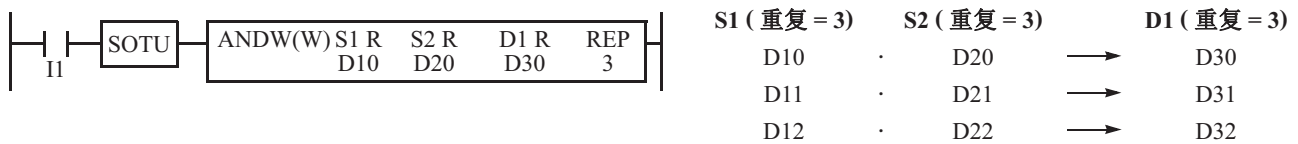
当指定 S1（源）和 D1（目标）重复时，不同的结果将设置到以 D1·D1+1 开始的 3 个设备。



### 重复所有源设备和目标设备

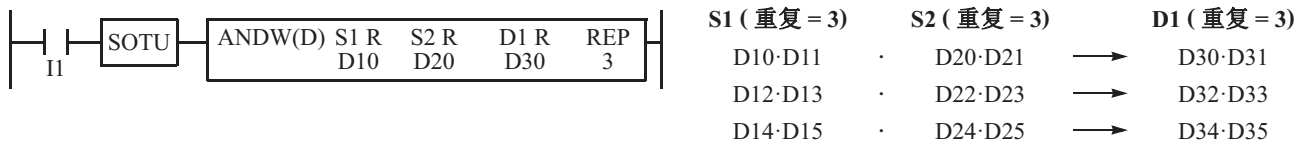
• 数据类型：字

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。



# 9: 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 1-65,535 位。结果将设置到源设备 S1 和特殊内部继电器 M8003（进位或借位）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 和特殊内部继电器 M8003（进位或借位）中。

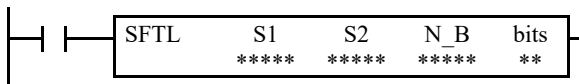
BCD 码左移指令将两个连续的数据寄存器中的 BCD 数字向左移。

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）

将数值按位左移。

FT2J FC6A

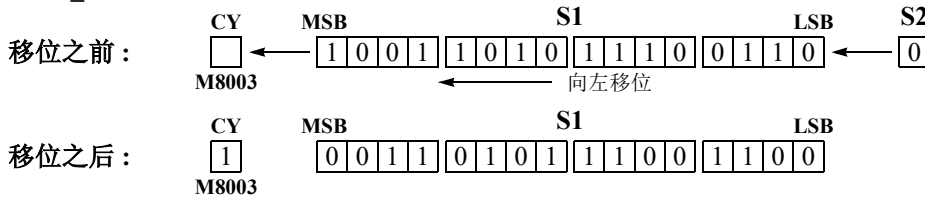


CY ← S1

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特殊内部继电器 M8003（进位或借位）。源设备 S2 指定的 0 或 1 被设置到 LSB。

• S2 = 0、N\_B = 16、Bits = 1



## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	1-65,535	—
Bits	要移位的位数	—	—	—	—	—	—	X	—	1-15	—

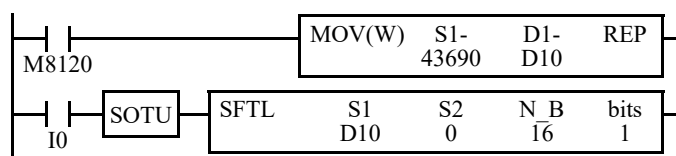
关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 无法将特殊内部继电器指定为 S1。

## 9: 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。

梯形图程序运行时，MOV（传送）指令将把 43,690 设置到数据寄存器 D10。

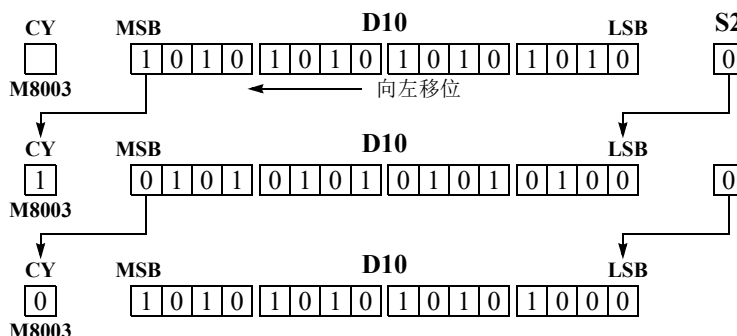
每次输入 I0 打开时，数据寄存器 D10 的 16 位数据都将向左移动设备位所指定的 1 位。最后一个移出的位状态将设置到特殊内部继电器 M8003（进位或借位）。将 0 设置到 LSB。

要移动的位数 = 1

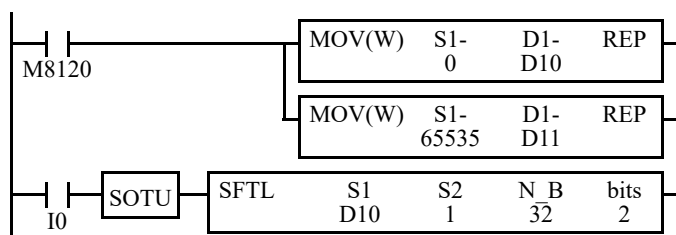
移位之前：D10 = 43690

第一次移位之后：D10 = 21844

第二次移位之后：D10 = 43688



- N\_B = 32 位



M8120 是初始化脉冲特殊内部继电器。

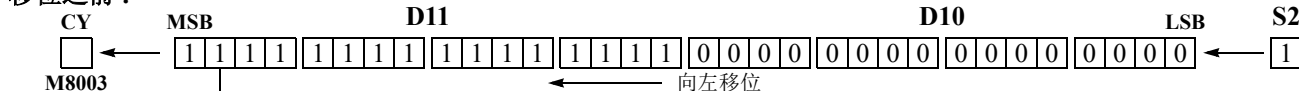
梯形图程序运行时，MOV（传送）指令将把 0 和 65,535 分别设置到数据寄存器 D10 和 D11。

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。

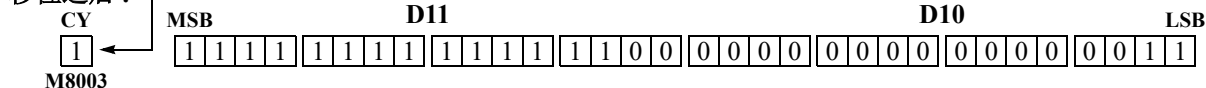
最后一个移出的位状态将设置到进位（特殊内部继电器 M8003）。将 1 设置到 LSB。

要移动的位数 = 2

移位之前：



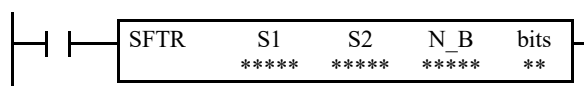
移位之后：



## SFTR (右移)

将数值按位右移。

FT2J FC6A

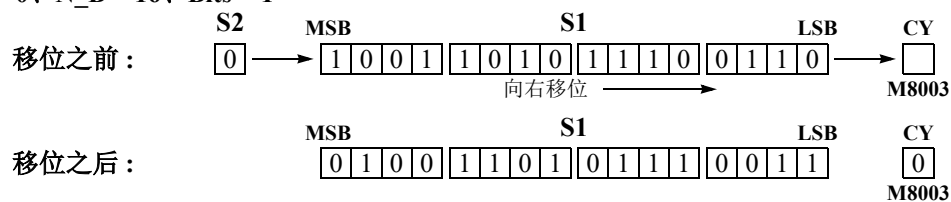


S1 → CY

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到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 将设置到 MSB。

• S2 = 0、N\_B = 16、Bits = 1



## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	1-65,535	—
Bits	要移位的位数	—	—	—	—	—	—	X	—	1-15	—

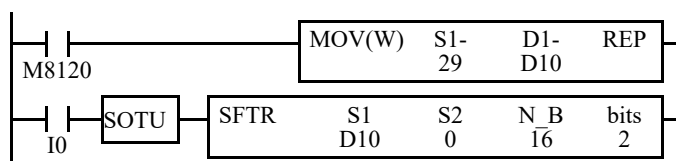
关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 无法将特殊内部继电器指定为 S1。

## 9: 移位/循环指令

### 示例：SFTR

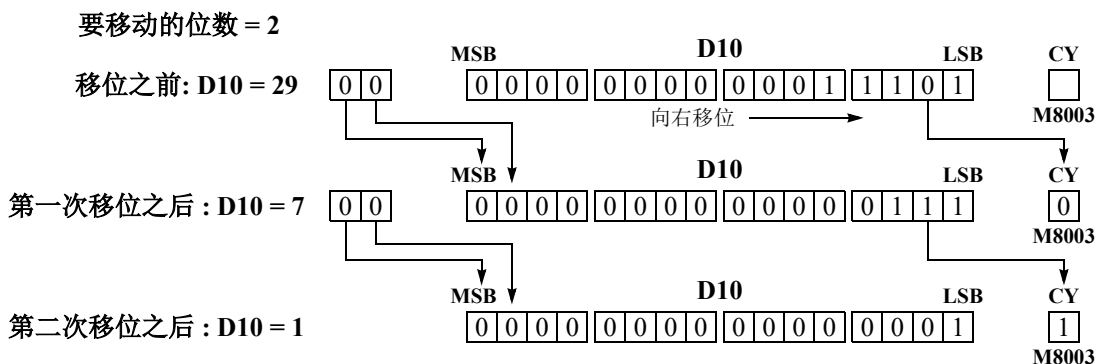
- 数据类型：字



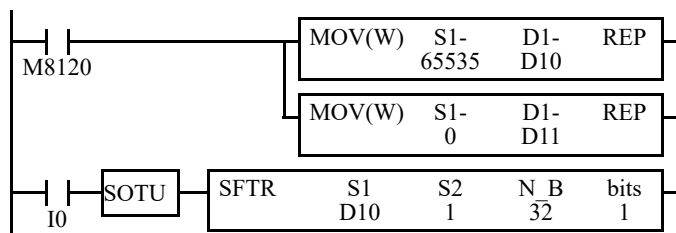
M8120 是初始化脉冲特殊内部继电器。

梯形图程序运行时，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特殊内部继电器 M8003（进位或借位）。将零设置到 MSB。



- 数据类型：双字

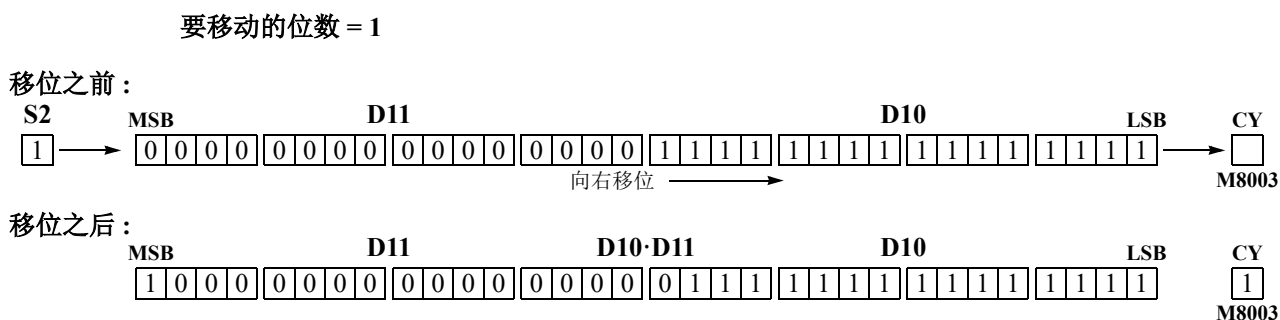


M8120 是初始化脉冲特殊内部继电器。

梯形图程序运行时，MOV（传送）指令将把 65,535 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特殊内部继电器 M8003（进位或借位）。将 1 设置到 MSB。





## BCDLS (BCD 码左移)

将 BCD 数字左移。

FT2J FC6A



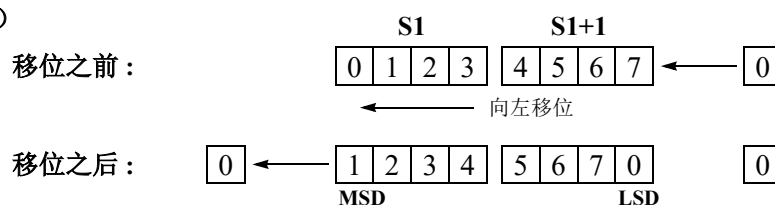
当输入打开时，S1 所指定的 32 位二进制数据将转换为 8 个 BCD 数字，并向左移动由 S2 所指定的数字个数，然后转换回 32 位二进制数据。

每个 S1 和 S1+1 的有效值是 0-9,999。

可以移动的数字个数是 1~7。

每次移动后，零将设置到最低的数字。

当 S2 = 1 (要移动的位数)



**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 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 0-9,999 之间。如果源数据超过 9,999，将存储错误代码 4 在用户程序执行错误代码 (D8006) 中。
- 如果 S2 的值不是 1 至 7，将存储错误代码 25 在用户程序执行错误代码 (D8006) 中。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

当 T (定时器) 或 C (计数器) 用作 S2 时，将显示定时器 / 计数器的当前值 (TC 或 CC)。

可以作为 S2 移动的数字个数是 1~7。

## 有效数据类型

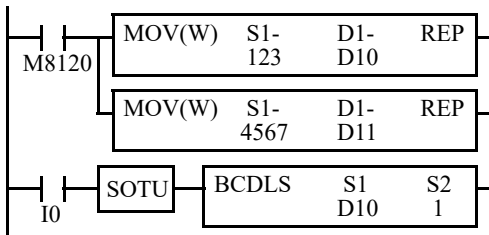
如果 D (数据寄存器) 这样的字设备指定为源 S1，则使用 2 点 (双字数据)。

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。

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。

## 9: 移位/循环指令

### 示例：BCDLS



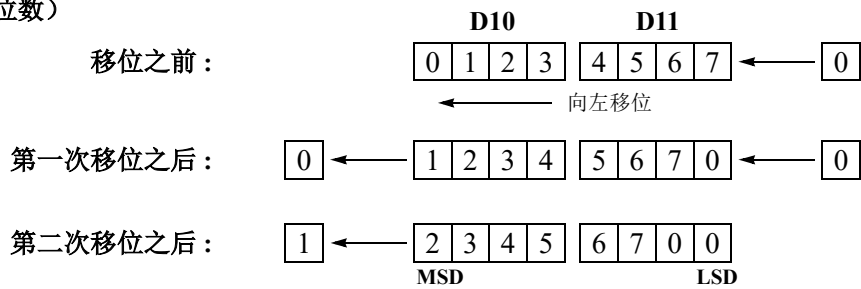
M8120 是初始化脉冲特殊内部继电器。

梯形图程序运行时，MOV（传送）指令将把 123 和 4,567 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。

每次移动后，零将设置到最低的数字。

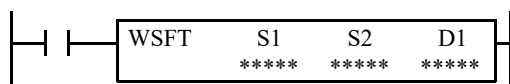
当 S2 = 1（要移动的位数）



## WSFT（字移位）

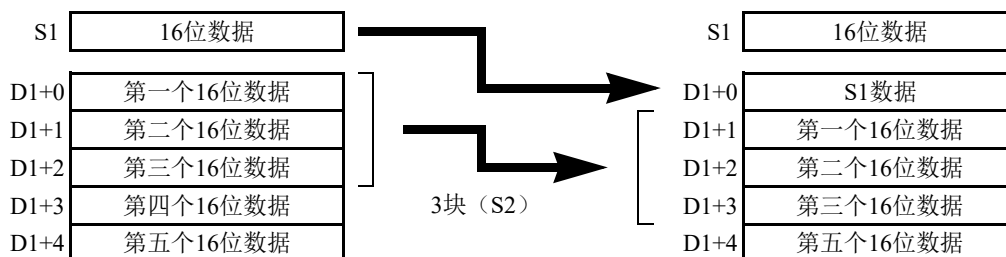
移动指定范围内的数值。

FT2J FC6A



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



注释：

- 执行 WSFT 时，BMOV/WSFT 执行标记 (M8024) 将打开。完成后，M8024 将关闭。
- 如果 D1+(S2) 超出设备范围，将导致用户程序执行错误，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

把 T（定时器）或 C（计数器）用作 S1 或 S2 时，将显示定时器 / 计数器的当前值（TC 或 CC）。

有效数据类型

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。

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。

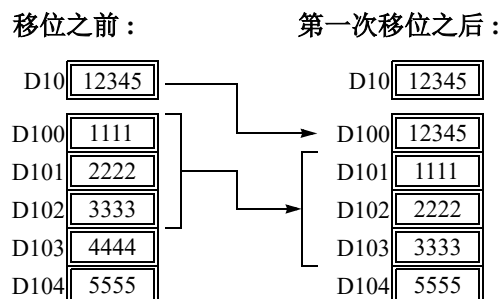
示例：WSFT



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。

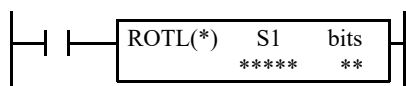


## 9: 移位/循环指令

### ROTL (循环左移)

将数值以位单位向左旋转。

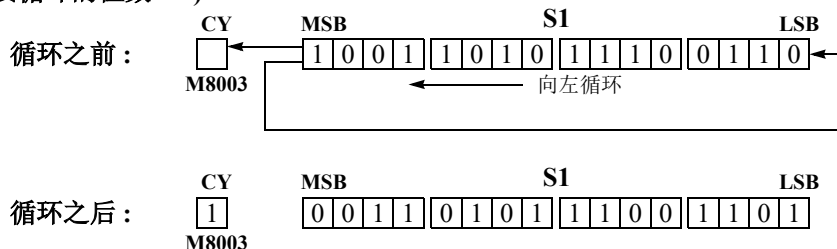
FT2J FC6A



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。

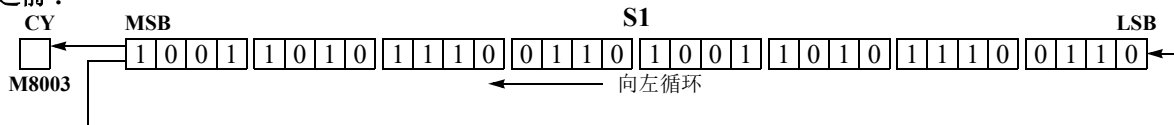
结果将设置到源设备 S1，并且循环出来的最后位状态将设置到进位（特殊内部继电器 M8003）。

- 数据类型：字(要循环的位数 = 1)

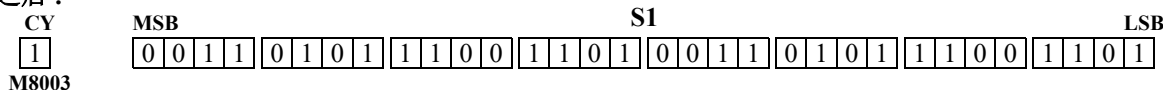


- 数据类型：双字(要循环的位数 = 1)

循环之前：



循环之后：



#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 无法将特殊内部继电器指定为 S1。

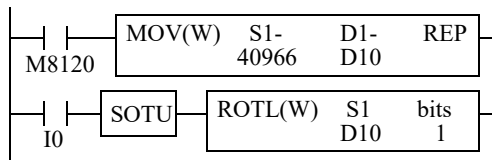
字数据要循环的位数可以是 1~15；或者双字数据要循环的位数可以是 1~31。

#### 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) 或 32 点 (双字数据)。
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) 或 2 点 (双字数据)。
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。

梯形图程序运行时，MOV（传送）指令将把 40,966 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 的状态将设置到特殊内部继电器 M8003（进位或借位）。

要循环的位数 = 1

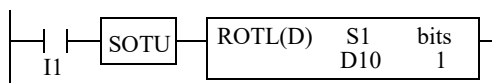
循环之前：D10 = 40966

第一次循环之后：D10 = 16397

第二次循环之后：D10 = 32794



• 数据类型：双字



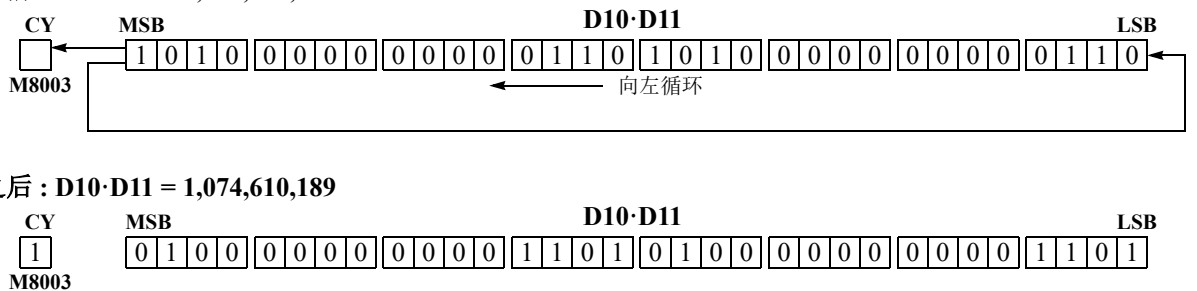
每次输入 I1 打开时，数据寄存器 D10 和 D11 的 32 位数据都将向左循环设备位所指定的 1 位。

MSB 的状态将设置到特殊内部继电器 M8003（进位或借位）。

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

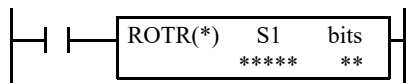


## 9: 移位/循环指令

### ROTR (循环右移)

将数值以位单位向右旋转。

FT2J FC6A

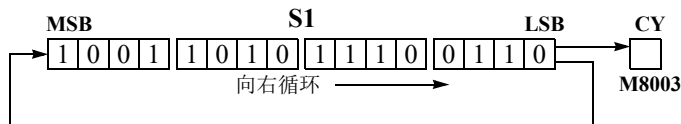


当输入打开时，源设备 S1 所指定的 16 位或 32 位数据将向右循环由设备位所指定的位数。

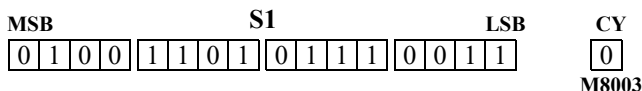
结果将设置到源设备 S1，并且循环出来的最后位状态将设置到特殊内部继电器 M8003（进位或借位）。

#### • 数据类型: 字(要循环的位数 = 1)

循环之前:

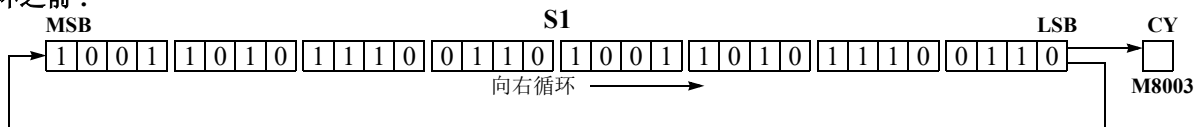


循环之后:

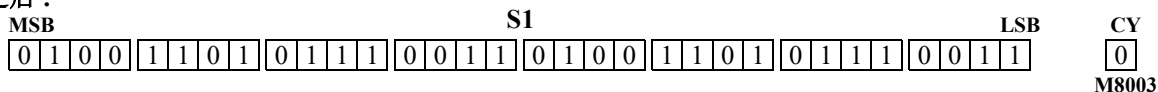


#### • 数据类型: 双字(要循环的位数 = 1)

循环之前:



循环之后:



### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 无法将特殊内部继电器指定为 S1。

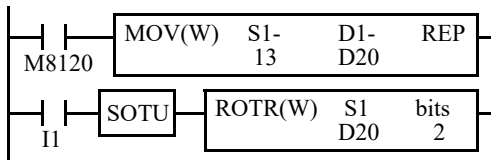
字数据可以循环的位数是 1~15；或者双字数据可以循环的位数是 1~31。

### 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) 或 32 点 (双字数据)。
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) 或 2 点 (双字数据)。
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。

梯形图程序运行时，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。

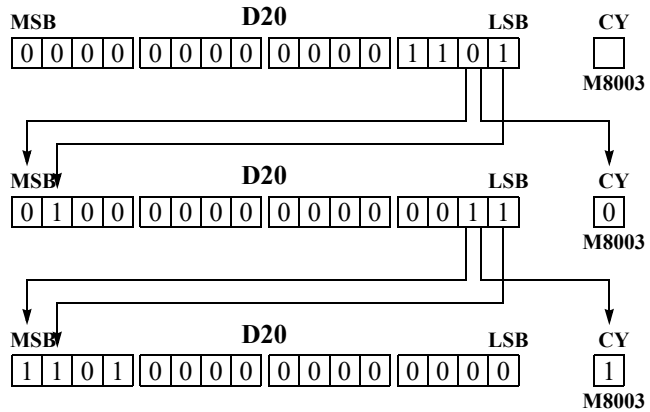
循环出来的最后位状态将设置到特殊内部继电器 M8003（进位或借位）。

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。

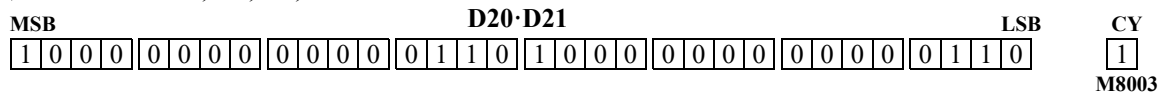
循环出来的最后位状态将设置到特殊内部继电器 M8003（进位或借位）。

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







# 10: 数据转换指令

## 简介

本章将对转换指定数据形式的数据转换指令进行介绍。

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。

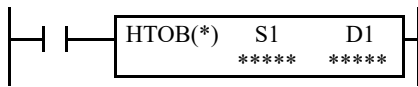
CVDT（转换数据）指令用于在字(W)、整数(I)、双字(D)、长整数(L)或浮点(F)之间进行数据类型转换。

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）

将二进制数据转换为 BCD 数据。

FT2J FC6A



S1 → D1

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。

字数据的源设备的有效值为 0~9,999，双字数据的源设备的有效值为 0~99,999,999。

**注释：**请确保 S1 指定的源在有效值范围之内。如果源数据不在有效范围内，将导致用户程序执行错误，将存储错误代码 5 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 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，将显示定时器 / 计数器当前值（TC 或 CC）。

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，可以是 0~65,535。

数据类型为 W（字）时，S1（二进制数据）的有效范围为 0 至 9,999。

数据类型为 D（双字）时，S1（二进制数据）的有效范围为 0 至 99,999,999。

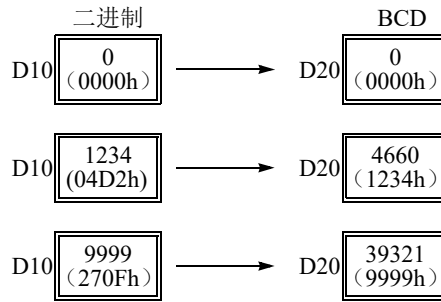
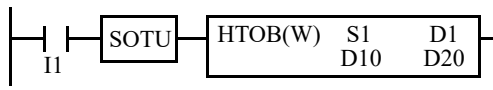
### 有效数据类型

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）或 32 点（双字数据）。
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）或 2 点（双字数据）。
L（长整数）	—	
F（浮点）	—	

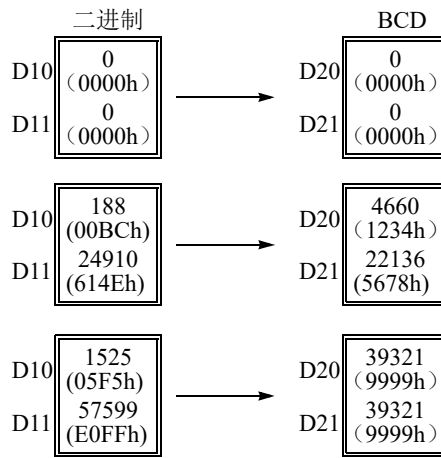
## 10: 数据转换指令

### 示例：HTOB

- 数据类型：字



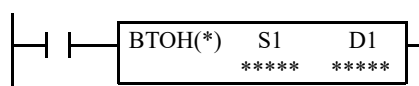
- 数据类型：双字



**BTOH (BCD 码→ HEX)**

将 BCD 数据转换为二进制数据。

FT2J FC6A



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。

字数据的源设备的有效值为 0~9,999 (BCD)，双字数据的源设备的有效值为 0~99,999,999 (BCD)。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 如果 BCD 数据的每个位数都是 0-9 以外的值，将存储错误代码 6 在用户程序执行错误代码 (D8006) 中。
- 请确保 S1 所指定的源数据的每个数字都是 0 - 9。如如果源数据不在有效范围内，将存储错误代码 6 在用户程序执行错误代码 (D8006) 中。

**有效设备**

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 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，将显示定时器 / 计数器当前值 (TC 或 CC)。

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。

数据类型为 W (字) 时，S1 (BCD 数据) 的有效范围为 0 至 9,999。

数据类型为 D (双字) 时，S1 (BCD 数据) 的有效范围为 0 至 99,999,999。

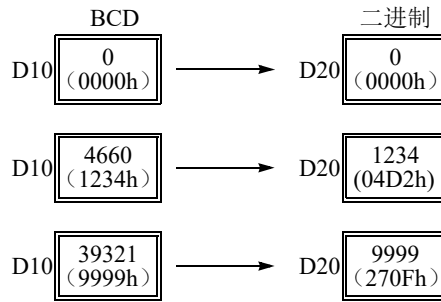
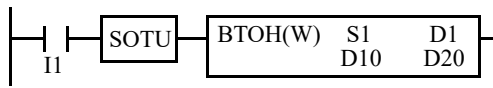
**有效数据类型**

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) 或 32 点 (双字数据)。
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) 或 2 点 (双字数据)。
L (长整数)	—	
F (浮点)	—	

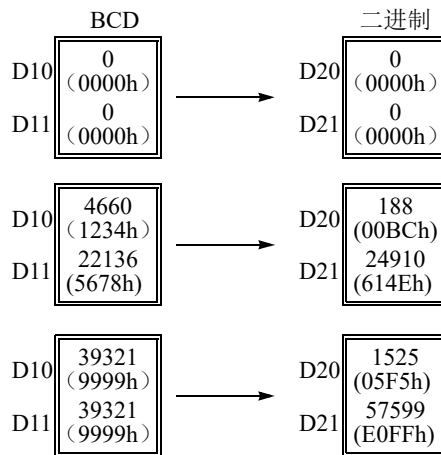
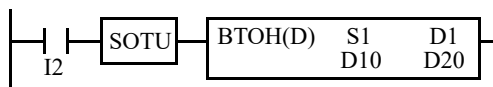
## 10: 数据转换指令

### 示例：BTOH

- 数据类型：字



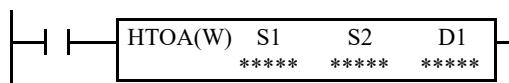
- 数据类型：双字



## HTOA (HEX → ASCII 码)

将二进制数据转换为 ASCII 数据。

FT2J FC6A



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。

**注释：**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，将存储错误代码 7 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

在 T (定时器) 或 C (计数器) 用作 S1 或 S2 时，将显示定时器 / 计数器当前值 (TC 或 CC)。

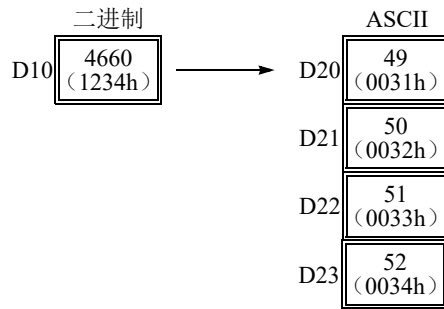
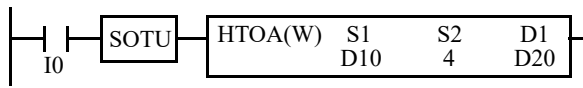
### 有效数据类型

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)。
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)。
L (长整数)	—	
F (浮点)	—	

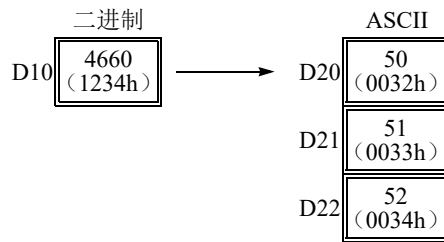
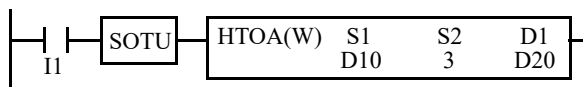
## 10: 数据转换指令

### 示例：HTOA

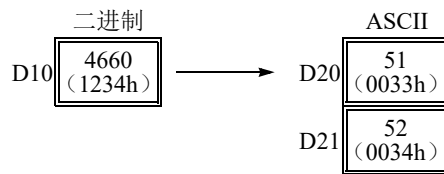
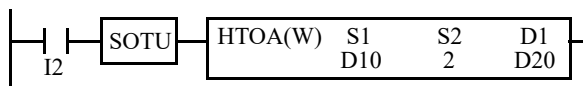
- 数字的位数：4



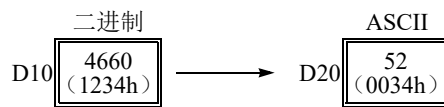
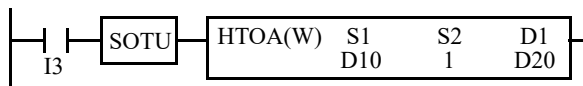
- 数字的位数：3



- 数字的位数：2



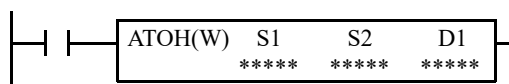
- 数字的位数：1



## ATOH (ASCII 码→HEX)

将 ASCII 数据转换为二进制数据。

FT2J FC6A



S1, S1+1, S1+2, S1+3 → D1

输入打开时，S1 所指定的 ASCII 数据将转换为 16 位二进制数据。次数由 S2 所指定的数字位数确定。然后，数据将存储到设备 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。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 要转换的源 S1 数据的有效值是 30h - 39h 和 41h - 46h。如果 S1 数据不在有效范围内，将存储错误代码 8 在用户程序执行错误代码 (D8006) 中。
- 如果 S2 的值超出范围，将存储错误代码 7 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 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，将显示定时器 / 计数器当前值 (TC 或 CC)。

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。

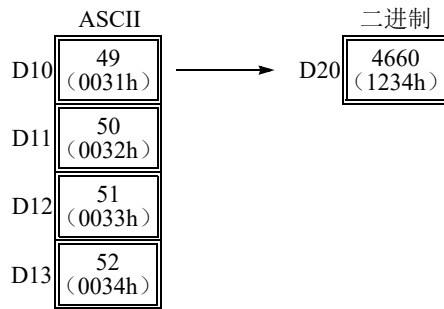
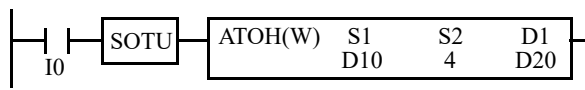
### 有效数据类型

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)。
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)。
L (长整数)	—	
F (浮点)	—	

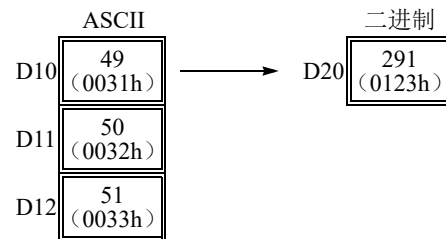
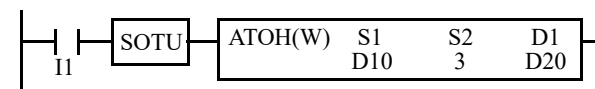
## 10: 数据转换指令

### 示例：ATOH

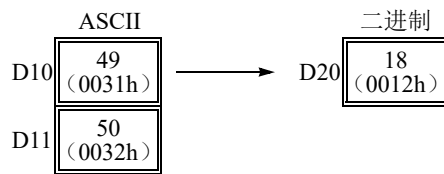
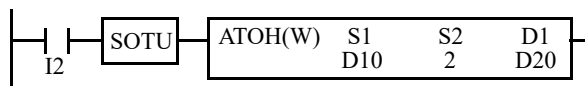
- 数字的位数：4



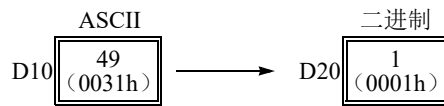
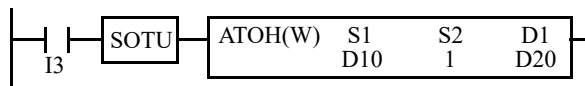
- 数字的位数：3



- 数字的位数：2



- 数字的位数：1

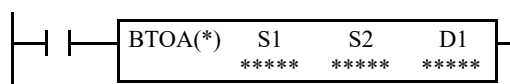




## BTOA (BCD 码→ ASCII 码)

将二进制数据转换为 BCD 数据，然后再转换为 ASCII 数据。

FT2J FC6A



字数据: S1 → D1, D1+1, D1+2, D1+3, D1+4

双字数据: 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。

字数据可转换的位数是 1~5, 双字数据可转换的位数是 1~10。

**注释:** 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, 将存储错误代码 7 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围, 请参见第 2-1 页上的“设备地址”。

在 T (定时器) 或 C (计数器) 用作 S1 或 S2 时, 将显示定时器 / 计数器当前值 (TC 或 CC)。

数据类型为 W (字) 时, S2 (位数) 的有效范围为 1 至 5。

数据类型为 D (双字) 时, S2 (位数) 的有效范围为 1 至 10。

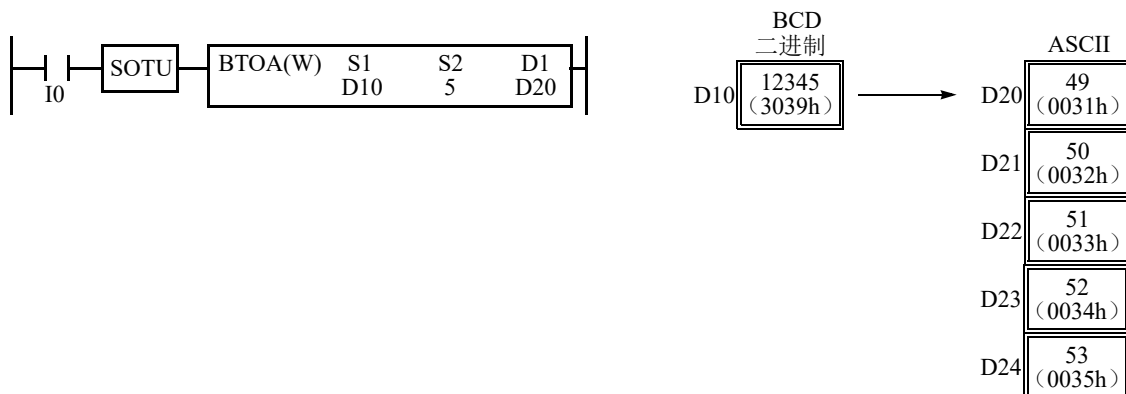
### 有效数据类型

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) 或 32 点 (双字数据)。
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) 或 2 点 (双字数据)。
L (长整数)	—	
F (浮点)	—	

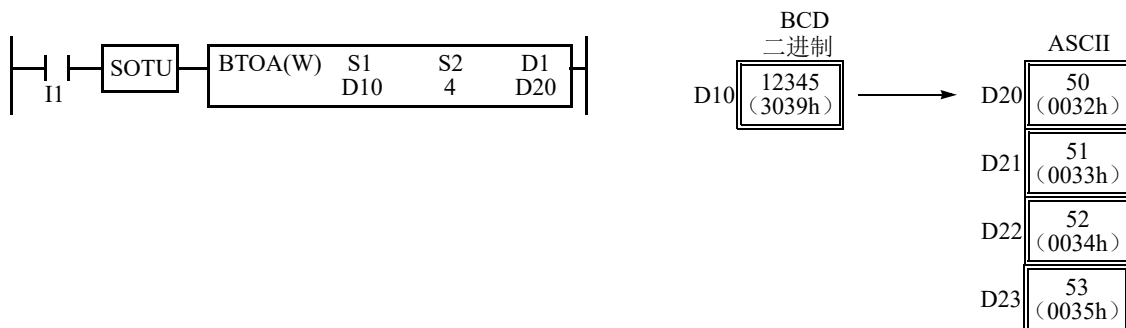
## 10: 数据转换指令

### 示例：BTOA(W)

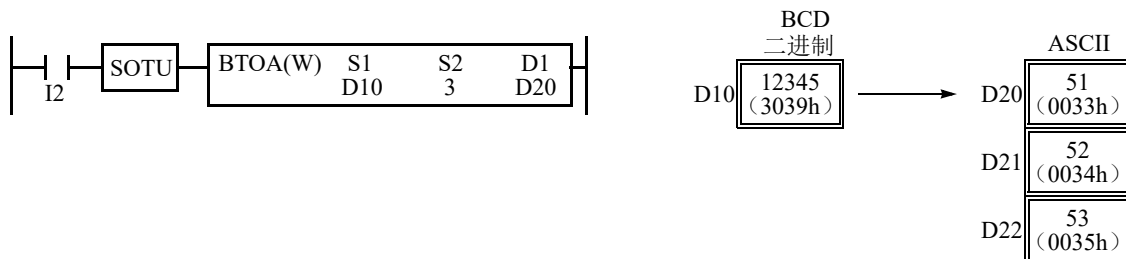
- 数字的位数：5



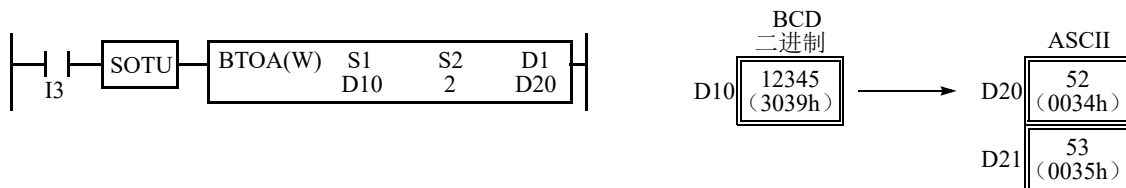
- 数字的位数：4



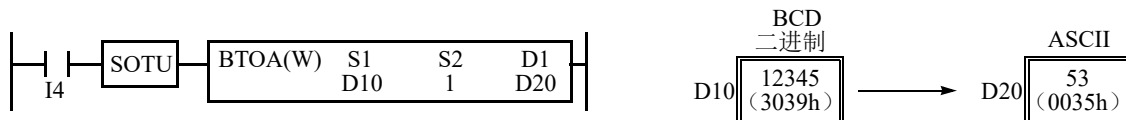
- 数字的位数：3



- 数字的位数：2

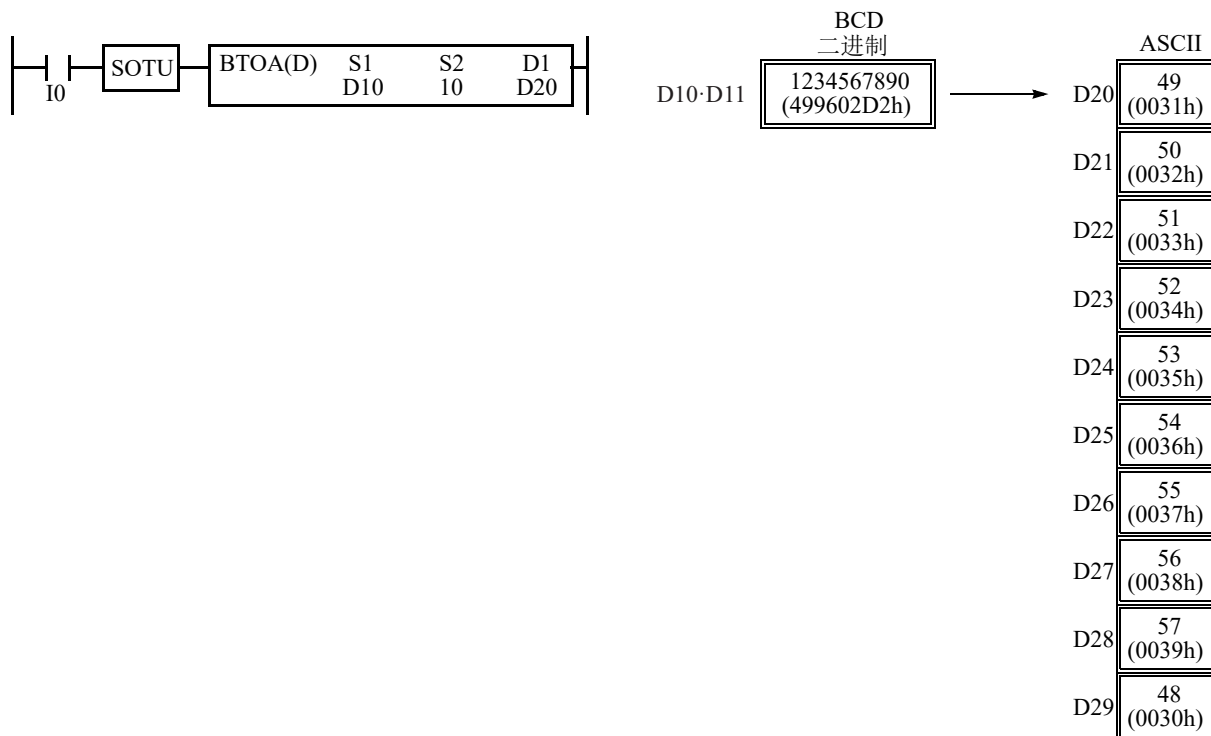


- 数字的位数：1

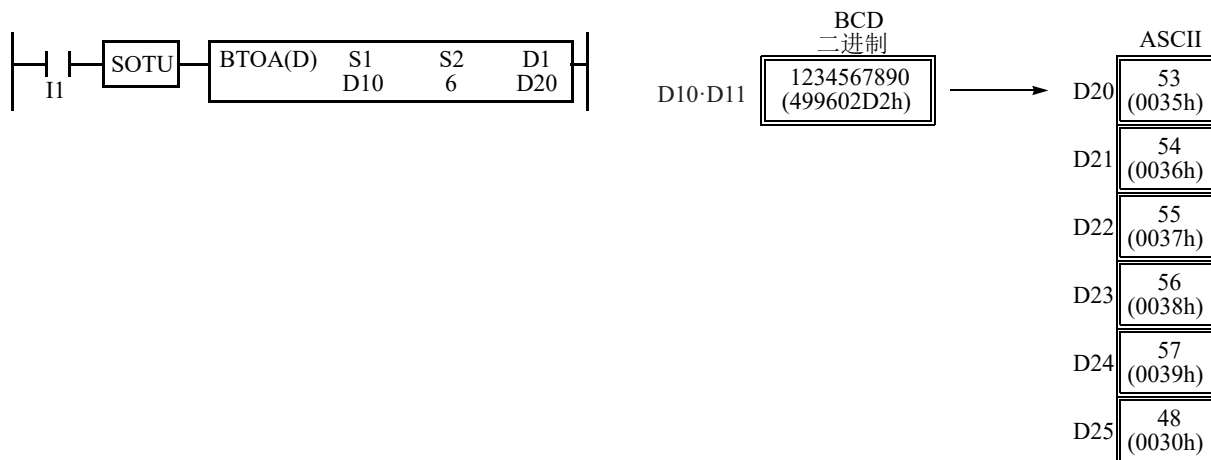


示例：BTOA(D)

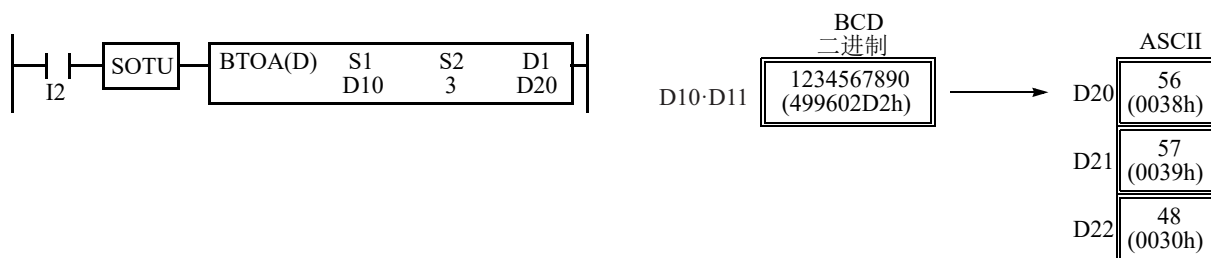
- 数字的位数：10



- 数字的位数：6



- 数字的位数：3

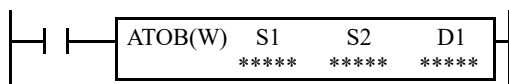


## 10: 数据转换指令

### ATOB (ASCII 码→BCD 码)

将 ASCII 数据转换为 BCD 数据，然后再转换为二进制数据。

FT2J FC6A



字数据: S1, S1+1, S1+2, S1+3, S1+4 → D1

双字数据: 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。

字数据可转换的位数是 1~5, 双字数据可转换的位数是 1~10。

**注释:** 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。

- 如果 S1 中设置的每个数字的数据不是 ASCII 数据 (30h - 39h), 将存储错误代码 8 在用户程序执行错误代码 (D8006) 中。
- 如果 S2 的值超出范围, 将存储错误代码 7 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围, 请参见第 2-1 页上的“设备地址”。

▲ 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, 将显示定时器 / 计数器当前值 (TC 或 CC)。

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。

要转换的源 S1 数据的有效值是 30h - 39h。

数据类型为 W (字) 时, S2 (位数) 的有效范围为 1 至 5。

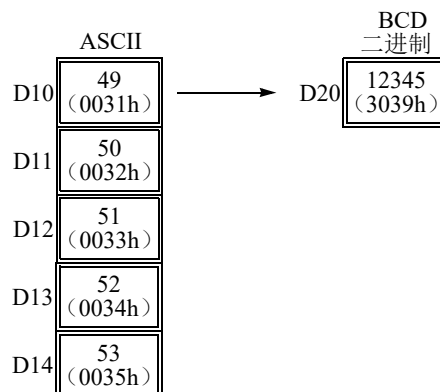
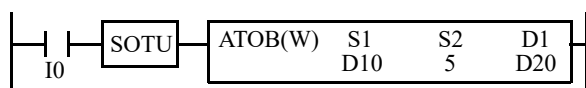
数据类型为 D (双字) 时, S2 (位数) 的有效范围为 1 至 10。

#### 有效数据类型

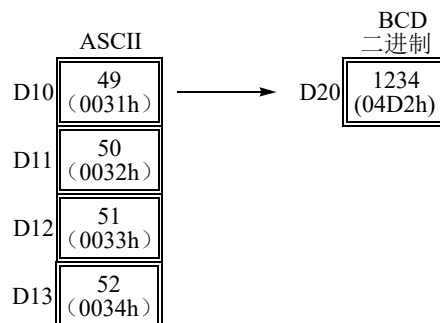
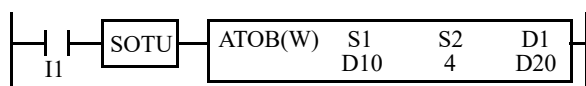
W (字)	X	把 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备指定为源或目标时, 将使用 16 点 (字数据) 或 32 点 (双字数据)。
I (整数)	—	
D (双字)	X	如果将 T (定时器)、C (计数器) 或 D (数据寄存器) 等字设备指定为源或目标, 则使用 1 点 (字数据) 或 2 点 (双字数据)。
L (长整数)	—	
F (浮点)	—	

示例：ATOB(W)

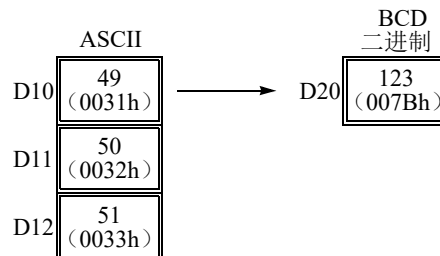
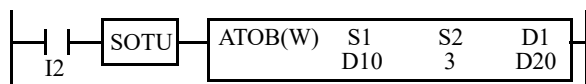
- 数字的位数：5



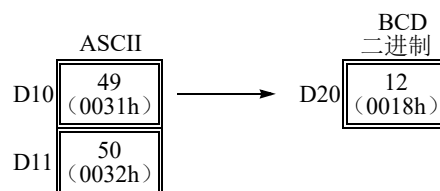
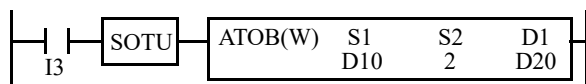
- 数字的位数：4



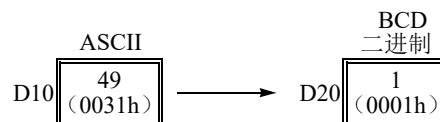
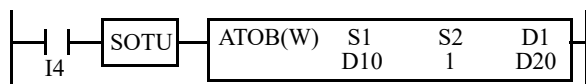
- 数字的位数：3



- 数字的位数：2



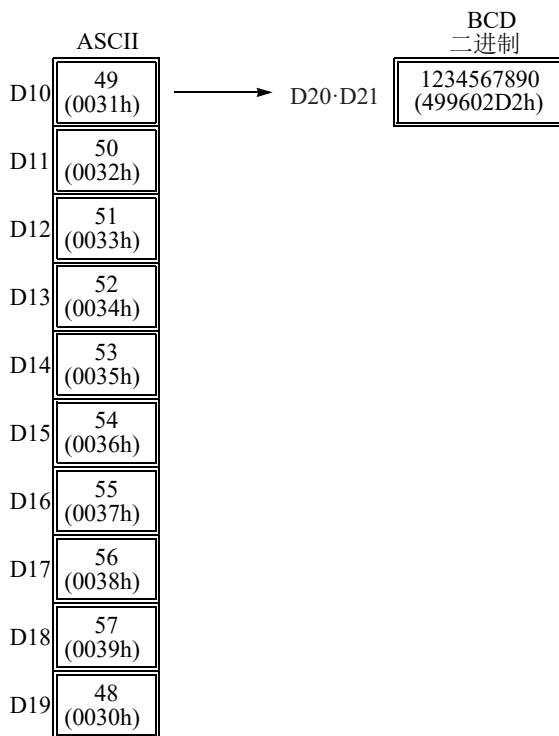
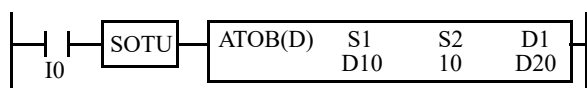
- 数字的位数：1



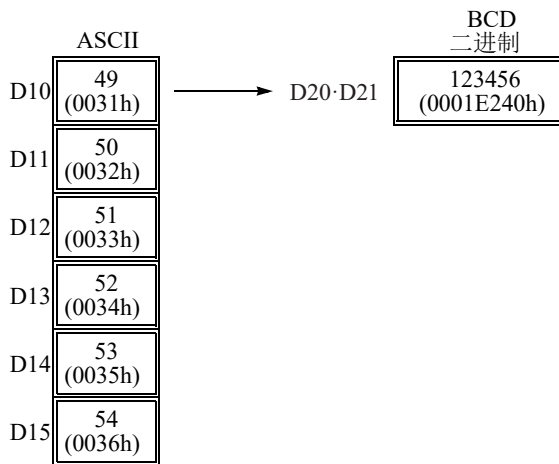
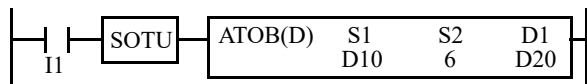
## 10: 数据转换指令

### 示例：ATOB(D)

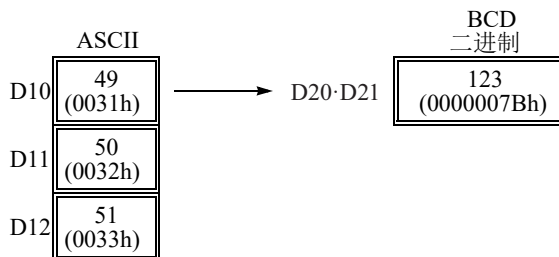
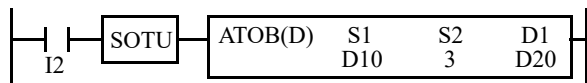
- 数字的位数：10



- 数字的位数：6



- 数字的位数：3



## ENCO (编码)

搜索指定范围内第一个设置为开的位。

FT2J FC6A



当输入打开时，将开始搜索设置为打开的第一位。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 65,535 存储到 D1。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 不能将特殊内部继电器指定为 D1。

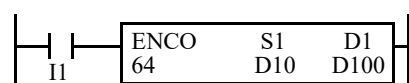
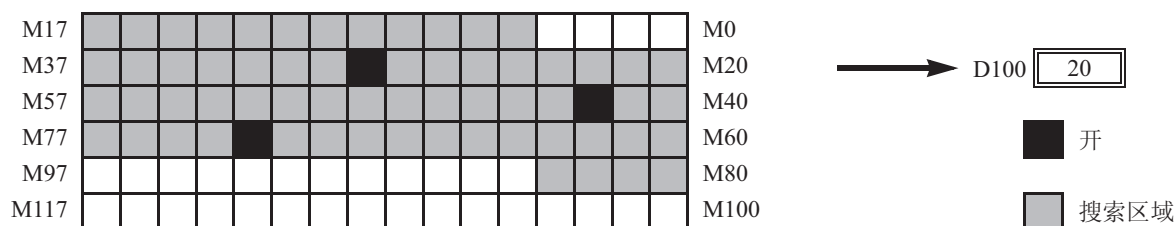
用于指定要搜索的位数有效值是 1 - 256。

## 示例：ENCO



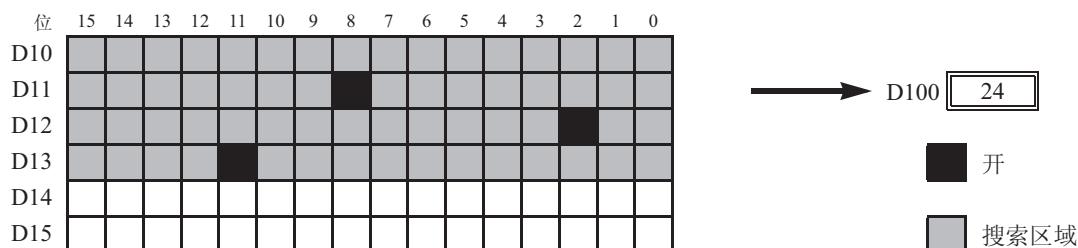
当输入 I0 打开时，将在从设备 S1 所指定的内部继电器 M4 开始的一组 64 位中搜索打开的位。

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 位开始的一组 64 位中搜索打开的位。

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。



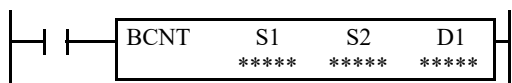




## BCNT (位计数)

对指定范围内打开的位数进行计数。

FT2J FC6A



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搜索打开的总位数。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。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 如果 S1+(S2) 超出设备范围，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。
- 如果 S2 的值超出范围，将存储错误代码 22 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	X	—
D1 (目标 1)	要存储“开”位数的目标	—	X	▲	X	X	X	X	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

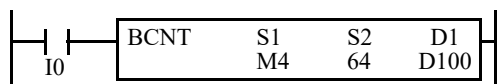
▲ 不能将特殊内部继电器指定为 D1。

在 T (定时器) 或 C (计数器) 用作 S2 时，将显示定时器 / 计数器当前值 (TC 或 CC)。

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。

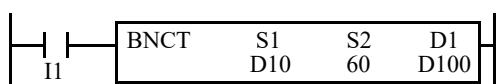
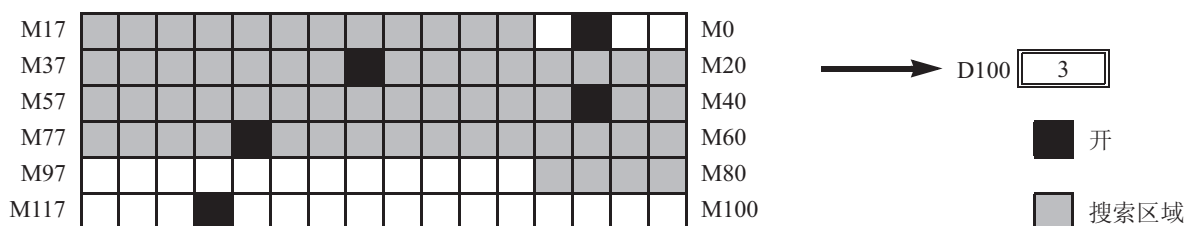
S2 (查找范围) 的有效范围为 1 至 256。

### 示例: BCNT



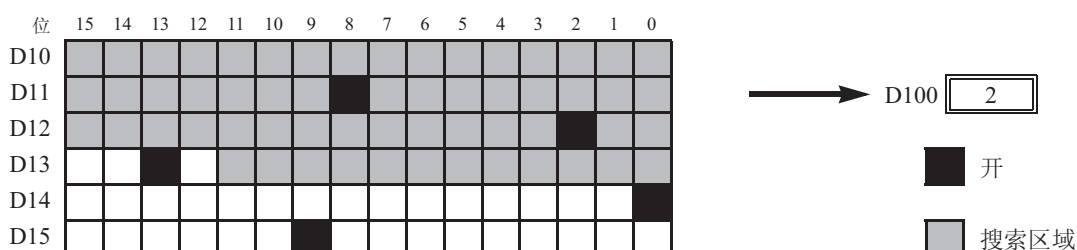
当输入打开时，将在从源设备 S1 所指定的内部继电器 M4 开始的一组 64 位中搜索打开的总位数。

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 I0 打开时，将在从设备 S1 所指定的数据寄存器 D10 的第 0 位开始的一组 60 位中搜索打开的总位数。

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。

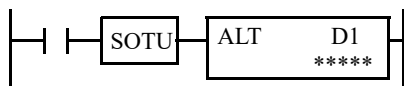


## 10: 数据转换指令

### ALT（交替输出）

打开或关闭输出。

FT2J FC6A



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。

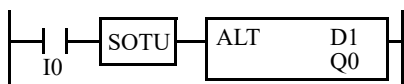
**注释：**由于 ALT 指令是在输入打开时在每个扫描中执行的，因此必须使用来自 SOTU 或 SOTD 指令的脉冲输入。有关 SOTU 指令和 SOTD 指令，请参见第 4-31 页上的“SOTU（上升沿微分）”。

#### 有效设备

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

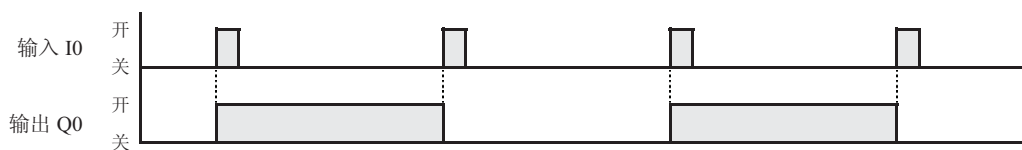
关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

#### 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。

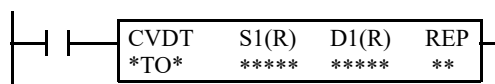
当输入 I0 再次打开时，输出 Q0 将关闭。



## CVDT (转换数据类型)

转换数据类型。

FT2J FC6A



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，只传送整数数据，省略小数。

当源数据超过了目标数据的范围时，目标在目标数据内存储一个最接近源数据的值。

## 注释:

- 当为源数据和目标数据指定了相同的数据类型时，CVDT 指令的功能与 MOV 指令相同。
- 除非源数据和目标数据都选择了 F(浮点)数据类型，否则只移动整数，而不移动分数。
- 当源数据超出目的地数据的范围时，目的地会在目的地数据中存储一个最接近源数据的值。
- 当选择 F(浮点)数据类型，并且 S1 不符合标准浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码(D8006)中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	存储转换数据类型的起始设备编号	X <sup>*1</sup>	X <sup>*1</sup>	X <sup>*1</sup>	X <sup>*1</sup>	X <sup>*1*2</sup>	X <sup>*2</sup>	X	—	X	1-99
D1 (目标 1)	存储转换数据的起始设备编号	—	X <sup>*1</sup>	X <sup>*1*3</sup>	X <sup>*1</sup>	X <sup>*1*4</sup>	X <sup>*1*4</sup>	X	—	—	1-99

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

\*1 无法在数据类型 F(浮点)中使用。

\*2 在 S1 中指定 T/C 时，将变为当前值区域。

\*3 不能将特殊内部继电器指定为 D1。

\*4 在 T(定时器)或 C(计数器)用作 S1 时，将显示定时器/计数器当前值(TC 或 CC)。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 65,535 之间。

## 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)或 32 点(双字、长整数或浮点数据)。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)或 2 点(双字、长整数或浮点数据)。如果对字设备指定重复，则设备的数量将以 1 点或 2 点为增量增加。
L(长整数)	X	
F(浮点)	X	

## 10: 数据转换指令

### 示例：CVDT

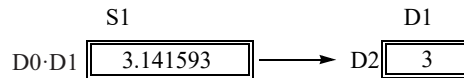
- 数据类型：S1 或 D1 不是 F(浮点)

除非是给源和目标选择了 F(浮点) 数据，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。



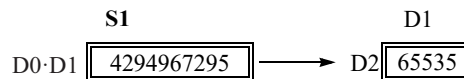
- 数据类型：S1 有一个比 D1 更大的数据范围

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	4,294,967,295
目标	W	65,535

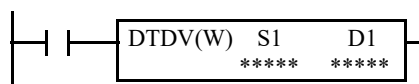
当输入 I0 打开时，65,535 将存储到数据寄存器 D2。



## DTDV (数据分割)

将字数据分成两个字节数据。

FT2J FC6A



S1 → D1, D1+1

当输入打开时，S1 所指定的 16 位二进制数据将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。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

在 T（定时器）或 C（计数器）用作 S1 时，将显示定时器 / 计数器当前值（TC 或 CC）。

目标设备 D1 使用以 D1 指定的设备开始的两个数据寄存器。

### 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)。
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)。
L (长整数)	—	
F (浮点)	—	

### 示例 : DTDV

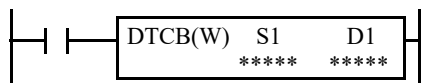


## 10: 数据转换指令

### DTCB (数据组合)

组合两个字节的的数据。

FT2J FC6A



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显示，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。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 不能将特殊内部继电器指定为 D1。

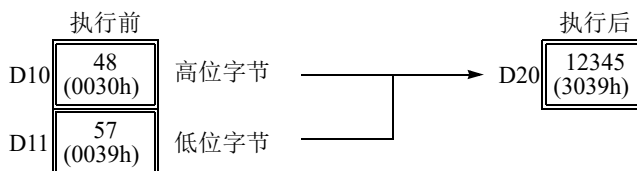
当 T (定时器) 或 C (计数器) 用作 D1，将数据作为预置值 (TP 或 CP) 写入。

源设备 S1 使用以 S1 指定的设备开始的两个数据寄存器。

#### 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 (字数据)。
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)。
L (长整数)	—	
F (浮点)	—	

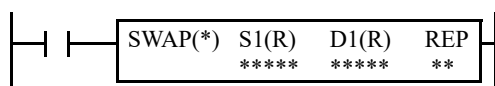
#### 示例: DTCB



## SWAP（数据交换）

交换指定数据的上下字节。

FT2J FC6A



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。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

### 有效数据类型

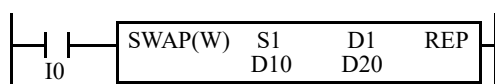
W (字)	X
I (整数)	—
D (双字)	X
L (长整数)	—
F (浮点)	—

如果将 D（数据寄存器）指定为源或目标，则使用 1 点（字数据）或 2 点（双字数据）。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。

### 示例：SWAP

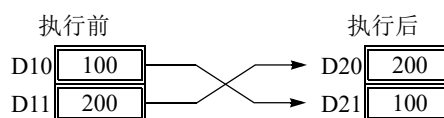
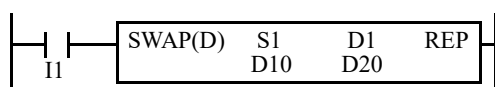
#### • 数据类型：W (字)

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。







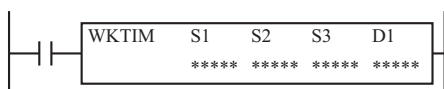
# 11: 周程序指令

本章将对在指定星期或日期时间打开或关闭输出的周程序指令进行介绍。

## WKTIM（日历定时器）

将指定星期和起始时间、结束时间与当前时间进行比较，并输出其结果。

FT2J FC6A

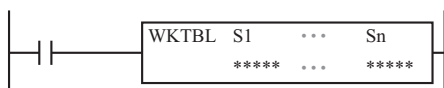


**注释：**WKTIM 指令与 FC5A 型 MICROSmart 的 WKTIM 指令通用。有关 WKTIM 指令的详情，请参见《FC5A 系列 MICROSmart Pentra 用户手册 高级卷》第 9 章“WKTIM（周定时器）”。

## WKTBL（周表）

将指定月日设置为特殊日期。

FT2J FC6A

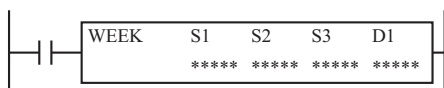


**注释：**WKTBL 指令与 FC5A 型 MICROSmart 的 WKTBL 指令通用。有关 WKTBL 指令的详情，请参见《FC5A 系列 MICROSmart Pentra 用户手册 高级卷》第 9 章“WKTBL（周表）”。

## WEEK（周定时器）

将指定星期和 ON 时间、OFF 时间与当前时间进行比较，并输出其结果。

FT2J FC6A

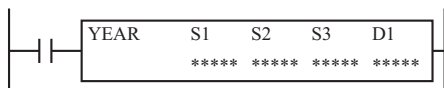


**注释：**有关 WEEK 指令的详情，请参见第 11-3 页上的“WEEK（周定时器）”。

## YEAR（年定时器）

将指定日期与当前日期进行比较，并输出其结果。可在一年期限内指定特殊日期。

FT2J FC6A



**注释：**有关 YEAR 指令的详情，请参见第 11-16 页上的“YEAR（年定时器）”。

## 11: 周程序指令

### 关于周程序指令的区别

周程序指令包括 WKTIM 指令、WKTBL 指令、WEEK 指令、YEAR 指令 4 种。WEEK 指令、YEAR 指令可在指令对话框中简单设置星期、ON 时间、OFF 时间、特别日期等的初始值。WKTIM 指令、WKTBL 指令必须使用数据寄存器设置初始值。

### WKTIM 指令、WEEK 指令的比较

项目	WKTIM 指令	WEEK 指令
1 个梯形图程序中可使用的指令数量	无限制	10 个
1 个指令中可设置的模式数量	1 个模式	20 个模式
脉冲输出功能	无	有
初始值的设置方法	使用数据寄存器设置初始值	在 WindLDR 的对话框中设置初始值
设置模式的预览功能	无	有

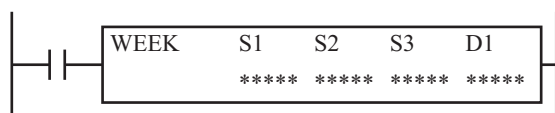
### WKTBL 指令、YEAR 指令的比较

项目	WKTBL 指令	YEAR 指令
1 个梯形图程序中可使用的指令数量	无限制	10 个
1 个指令中可设置的模式数量	20 个模式	20 个模式
脉冲输出功能	无	有
日历的指定方法	日期指定	日期指定 星期指定 月末指定
初始值的设置方法	在梯形图程序中使用数据寄存器设置初始值	在 WindLDR 的对话框中设置初始值
设置模式的预览功能	无	有

## WEEK（周定时器）

将指定星期和 ON 时间、OFF 时间与当前时间进行比较，并输出其结果。

FT2J FC6A

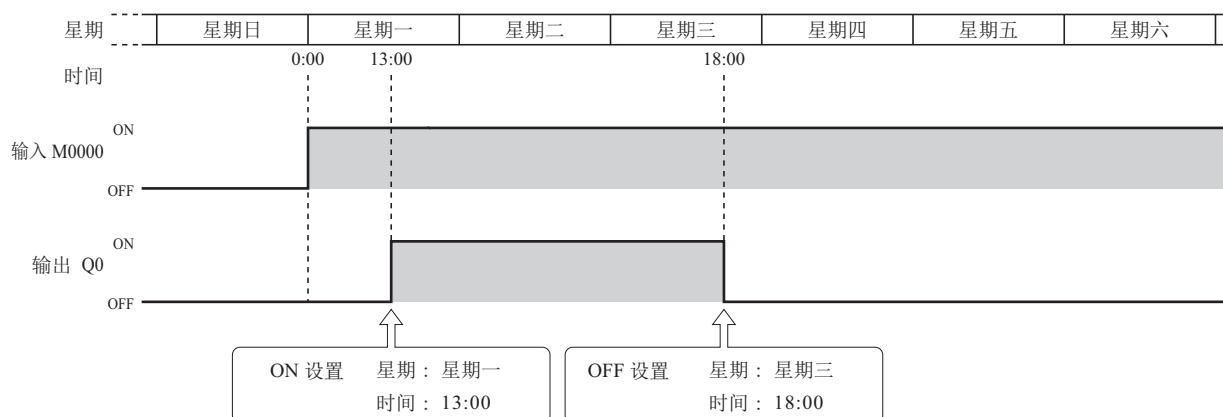


WEEK 指令将指定的星期、ON 时间和 OFF 时间与当前时间进行对比并输出该结果。

在 ON 设置指定的星期和时间与当前的星期和时间匹配时，D1 指定的输出被打开。

此外，OFF 设置的星期和时间与当前星期和时间一致时，将关闭 D1 中指定的输出。

例) ON 设置为星期一 13:00、OFF 设置为星期三 18:00 时，输出 D1 将按如下所示打开 / 关闭。  
在星期一 13:00 WEEK 指令的输入为 ON 时，在星期一 13:00 打开输出 D1，并在星期三 18:00 关闭。  
(图表示 WEEK 指令的输入时间为星期一 0:00 时)



### 注释:

- 一个用户程序中最多可以使用 10 条 WEEK 指令。
- 通常，仅在当前时间和 ON/OFF 设置的时间匹配时，系统才会更新输出，但当 WEEK 指令输入从关闭变为打开时，当前时间的输出状态将根据 ON/OFF 设置来确定，从而打开或关闭输出。有关详情，请参见第 11-10 页上的“在配置的间隔内打开输入时的时间图”。
- 如果启用脉冲输出，则系统将打开输出以便在 ON 设置时间进行一次扫描。有关脉冲输出，请参见第 11-5 页上的“6. 脉冲输出”或第 11-8 页上的“6. 脉冲输出”。
- 对于 FT2J 型，在执行 WEEK 指令之前，请检查 HMI 功能启动完成标志 (M8022) 是否为 ON。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 如果 ON 时间设置为大于 2,359 的值、OFF 时间设置为大于 2,400 的值或 ON/OFF 时间的后 2 位设置为大于 59 的值，将存储错误代码 9 在用户程序执行错误代码 (D8006) 中。
- 中断程序中不能使用 WEEK 指令。如果使用该指令，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	第一个数据寄存器	—	—	—	—	—	—	▲	—	—	—
S2 (源 2)	初始化输入	—	—	X	—	—	—	—	—	—	—
S3 (源 3)	参数选项卡数目	—	—	—	—	—	—	—	—	X	—
D1 (目标 1)	输出端口	—	X	▲	—	—	—	—	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

▲ 不能将特殊数据寄存器指定为 S1。不能将特殊内部继电器指定为 D1。

### 设置

配置 WEEK 指令的星期和时间有两种方法。在“设备”选项卡上，通过“数据寄存器设置”指定方法。

- 将星期和时间配置为固定设置

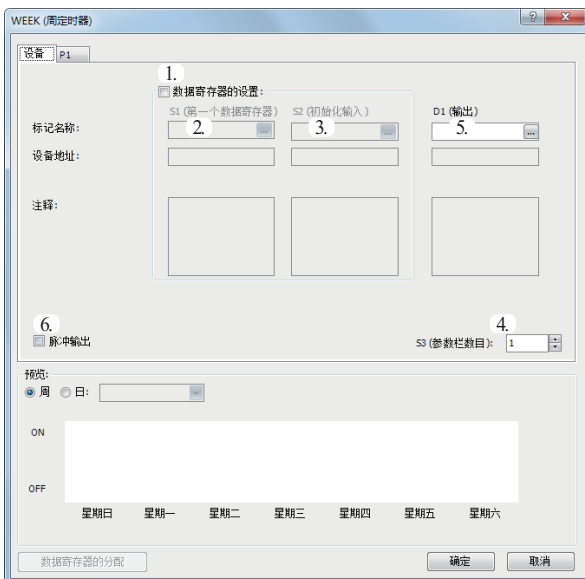
ON/OFF 设置的星期和时间单独确定。在梯形图程序运行时，无法更改 ON/OFF 设置的星期和时间。有关详情，请参见第 11-4 页上的“将星期和时间配置为固定设置”。

- 通过指定数据寄存器配置星期和时间

ON/OFF 设置的星期和时间根据指定数据寄存器中存储的值来配置。在梯形图程序运行时，可以更改 ON/OFF 设置的星期和时间。有关详情，请参见第 11-7 页上的“通过指定数据寄存器配置星期和时间”。

### 将星期和时间配置为固定设置

- “设备”选项卡



#### 1. 数据寄存器的设置

此选项用于确定是选择星期和时间的固定配置，还是选择通过数据寄存器的间接指定。

要将星期和时间配置为固定设置，请清除此复选框。

##### 未选中的复选框

星期和时间是固定设置。星期和时间在参数选项卡中设置。在梯形图程序运行时，无法更改星期和时间。有关设置，请参见第 11-5 页上的““P（参数）”选项卡”。

**注释：**选中此复选框时，星期和时间设置将通过数据寄存器间接指定。在梯形图程序运行时，可以更改星期和时间。有关数据寄存器的特定指令，请参见第 11-7 页上的“通过指定数据寄存器配置星期和时间”。

#### 2. S1（源 1）：第一个数据寄存器

将星期和时间配置为固定设置时，系统不会配置此设置。

#### 3. S2（源 2）：初始化输入

将星期和时间配置为固定设置时，系统不会配置此设置。

#### 4. S3（源 3）：参数栏数目

此设置配置参数选项卡的数目。当此值增加或减小时，对话框中显示的参数选项卡数也会增加或减小。每个参数选项卡占用 6 个字节的用户程序区域。

有关设置，请参见第 11-5 页上的““P（参数）”选项卡”。

### 5. D1（目标1）：输出

此设置用于配置的星期和时间与当前的星期和时间的对比结果的输出。

设置	说明
标记名称	指定每台设备的标记名称或设备地址。
设备地址	显示与标记名称相对应的设备地址。
备注	显示设备地址的备注。此项可进行编辑。

### 6. 脉冲输出

此设置配置 D1 的操作（输出）。此设置将应用于所有参数选项卡。

#### 已选中的复选框

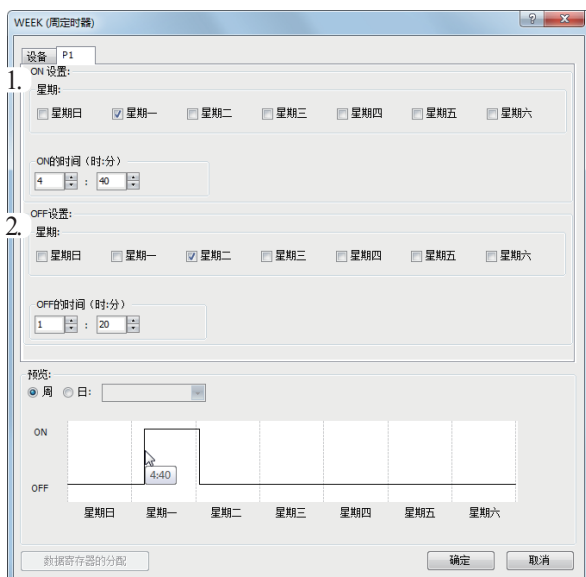
当 ON 设置中配置的星期和时间与当前的星期和时间匹配时，系统将打开输出以进行一次扫描。

#### 未选中的复选框

系统将根据 ON 设置和 OFF 设置来打开和关闭输出。

### • “P（参数）”选项卡

此选项卡配置输出的 ON/OFF 设置。一条 WEEK 指令最多可以配置 20 个参数选项卡。



#### 1. ON 设置

此部分配置要打开输出的星期和时间。系统将在指定星期的指定时间打开输出。

设置	说明	范围
星期	指定星期。	—
ON 的时间	输入时间。在 00:00 到 23:59 之间的范围内设置时间。	小时：0 到 23 分钟：0 到 59

#### 2. OFF 设置

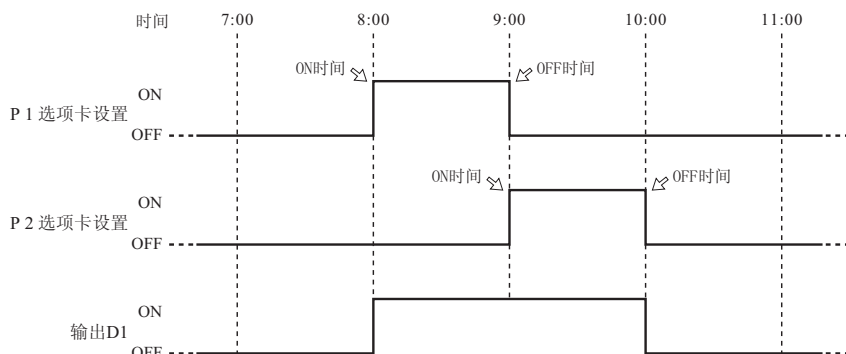
此部分配置要关闭输出的星期和时间。系统将在指定星期的指定时间关闭输出。

设置	说明	范围
星期	指定星期。	—
OFF 的时间	输入时间。在 00:00 到 24:00 之间的范围内设置时间。	小时：0 到 24 分钟：0 到 59

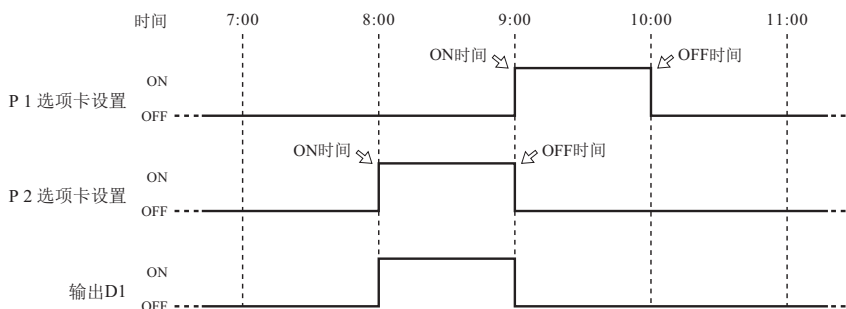
## 11: 周程序指令

**注释:** 如果时间在其他选项卡上的设置中重复, 则编号较大的选项卡上的设置有效。

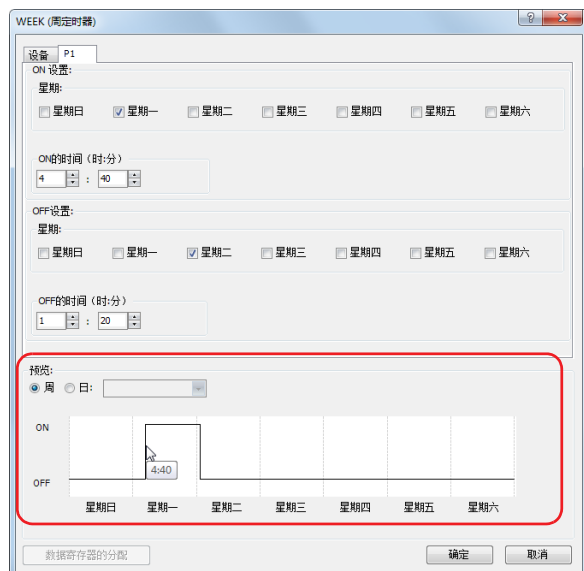
例如, 如果“P1”选项卡上的“ON 的时间”是 8:00, “OFF 的时间”是 9:00, 而“P2”选项卡上的“ON 的时间”是 9:00, “OFF 的时间”是 10:00, 则 9:00 设置在这两个选项卡上重复, 系统将禁用“P1”选项卡的“OFF 的时间”。在这种情况下, 系统将在 8:00 到 10:00 之间打开输出。



如果“P1”选项卡上的“ON 的时间”是 9:00, “OFF 的时间”是 10:00, 而“P2”选项卡上的“ON 的时间”是 8:00, “OFF 的时间”是 9:00, 则 9:00 设置在这两个选项卡上重复, 系统将禁用“P1”选项卡的“ON 的时间”。在这种情况下, 系统将在 8:00 到 9:00 之间打开输出。



### • 预览



预览将根据参数选项卡上配置的设置, 以时间图的形式显示输出的 ON/OFF 状态。预览可以按周或按日显示。

设置	说明
周	按周显示预览时, 请选择此单选按钮。
日	按日显示预览时, 请选择此单选按钮。

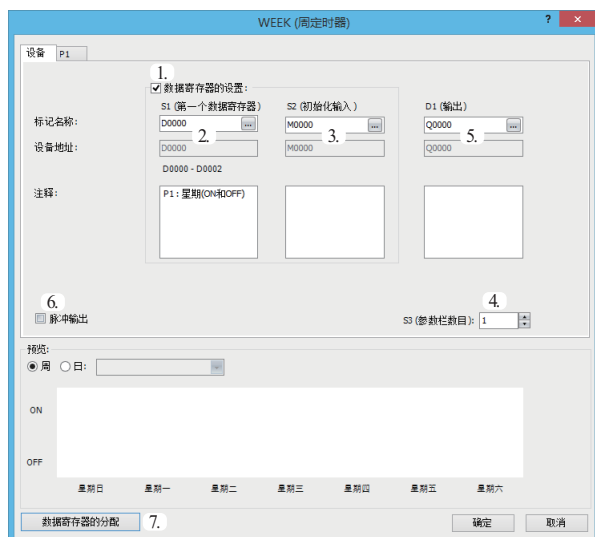
## 通过指定数据寄存器配置星期和时间

ON/OFF 设置的星期和时间根据指定数据寄存器中存储的值来配置。在梯形图程序运行时，可以更改 ON/OFF 设置的星期和时间。

**注释：**在打开 WEEK 指令输入的状态下，更改了 ON/OFF 设置的星期和时间时，当前星期和时间与更改后的 ON/OFF 设置的星期和时间达到一致之前，将不会反映至输出动作。

如需立即反映至输出动作，请在更改星期和时间后，先关闭 WEEK 指令，然后重新打开。

### • “设备”选项卡



### 1. 数据寄存器的设置

此选项用于区分是选择星期和时间的固定配置，还是选择通过数据寄存器的间接指定。

要通过指定数据寄存器配置星期和时间，请选中此复选框。

#### 已选中的复选框

星期和时间设置由数据寄存器间接指定。在梯形图程序运行时，可以更改星期和时间。有关分配数据寄存器区域，请参见第 11-9 页上的“数据寄存器分配”。

通过打开初始化输入，可以使用参数选项卡中配置的星期和时间来初始化数据寄存器。有关初始化，请参见第 11-8 页上的“3. S2 (源 2)：初始化输入”。

**注释：**如果未选中此复选框，则星期和时间为固定设置。星期和时间在参数选项卡中设置。在梯形图程序运行时，无法更改星期和时间。有关固定设置，请参见第 11-4 页上的“将星期和时间配置为固定设置”。

### 2. S1 (源 1)：第一个数据寄存器

此设备为数据寄存器区域的起始地址，用于存储 WEEK 指令的星期和时间的设置。

仅当使用数据寄存器间接指定 WEEK 指令的设置时，才会使用此设置。

设置	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示与标记名称相对应的设备地址。
使用的数据寄存器	显示用于存储设置的数据寄存器的范围。当设备地址或参数选项卡数变化时，此项也会随之变化。
备注	显示设备地址的备注。此项可进行编辑。

有关分配数据寄存器区域，请参见第 11-8 页上的““P (参数)”选项卡”。

## 11: 周程序指令

### 3. S2 (源 2): 初始化输入

此设置指定的设备用于初始化以 S1(源 1) 为起始地址的数据寄存器区域内存存储的星期和时间。

通过打开初始化输入, 可以在数据寄存器中存储参数选项卡上配置的值。

仅当使用数据寄存器间接指定 WEEK 指令的设置时, 才会使用此设置。

### 4. S3 (源 3): 参数栏数目

此设置配置参数选项卡的数目。

此设置与“将星期和时间配置为固定设置”的效果相同。请参见第 11-4 页上的“4. S3 (源 3): 参数栏数目”。

### 5. D1 (目标 1): 输出

此设置用于配置的星期和时间与当前的星期和时间的对比结果的输出。

此设置与“将星期和时间配置为固定设置”的效果相同。请参见第 11-5 页上的“5. D1 (目标 1): 输出”。

### 6. 脉冲输出

此设置配置 D1 的操作 (输出)。此设置将应用于所有参数选项卡, 与“将星期和时间配置为固定设置”的效果相同。请参见第 11-5 页上的“6. 脉冲输出”。

### 7. 数据寄存器的分配

单击此按钮可显示“设备分配”对话框。如下所示, 对话框 (8) 中将显示数据寄存器及其对应的 WEEK 指令设置的表格。单击“分配注释”(9) 以后, 您可以为与设置名称相对应的数据寄存器配置备注。

仅当使用数据寄存器间接指定 WEEK 指令的设置时, 才会使用此按钮。

#### • 设备分配对话框



#### • “P (参数)”选项卡

此选项卡配置输出的 ON/OFF 设置。1 条 WEEK 指令最多可以配置 20 个参数选项卡。

如果使用数据寄存器间接指定 WEEK 指令的设置, 则在打开初始化输入后, 参数选项卡上配置的设置将存储到数据寄存器中。

此设置与“将星期和时间配置为固定设置”的效果相同。请参见第 11-5 页上的““P (参数)”选项卡”。

#### • 预览

预览将根据参数选项卡上配置的设置, 以时间图的形式显示输出的 ON/OFF 状态。

此功能与“将星期和时间配置为固定设置”的效果相同。请参见第 11-6 页上的“预览”。



## 数据寄存器分配

如果使用数据寄存器间接指定 WEEK 指令的设置，则设置会按如下所示分配到数据寄存器。

存储目的地	设置	
起始地址 +0	P 1 选项卡	星期
起始地址 +1		ON 时间
起始地址 +2		OFF 时间
起始地址 +3	P 2 选项卡	星期
起始地址 +4		ON 时间
起始地址 +5		OFF 时间
⋮		⋮
起始地址 +57	P 20 选项卡	星期
起始地址 +58		ON 时间
起始地址 +59		OFF 时间

### • 星期数据寄存器分配

ON 设置的星期和 OFF 设置的星期将按位在 1 个数据寄存器中进行分配，如下所示。

ON 设置星期								OFF 设置星期										
位	15	14	13	12	11	10	9	位	8	7	6	5	4	3	2	1	位	0
保留	星期六	星期五	星期四	星期三	星期二	星期一	星期日	保留	星期六	星期五	星期四	星期三	星期二	星期一	星期日			

0: 禁用 (未选中复选框)

1: 启用 (选中复选框)

数据寄存器 (1 字)

### 星期设置示例

[ 将输出配置为星期一打开，星期五关闭 ]

在 ON 设置中启用“星期一”，在 OFF 设置中启用“星期五”

ON 设置星期								OFF 设置星期										
位	15	14	13	12	11	10	9	位	8	7	6	5	4	3	2	1	位	0
保留	0	0	0	0	0	0	1	0	保留	0	1	0	0	0	0	0		

星期设置 (ON): 0000010

星期设置 (OFF): 0100000

数据寄存器的值为 1000100000 (二进制) = 544 (十进制)。

[ 将输出配置为星期一和星期四打开，星期二和星期六关闭 ]

在 ON 设置中启用“星期一”和“星期四”，在 OFF 设置中启用“星期二”和“星期六”

ON 设置星期								OFF 设置星期										
位	15	14	13	12	11	10	9	位	8	7	6	5	4	3	2	1	位	0
保留	0	0	1	0	0	0	1	0	保留	1	0	0	0	0	1	0		

星期设置 (ON): 0010010

星期设置 (OFF): 1000100

数据寄存器的值为 1001001000100 (二进制) = 4676 (十进制)。

数据寄存器将按如下所示存储 ON 时间和 OFF 时间。



## 11: 周程序指令

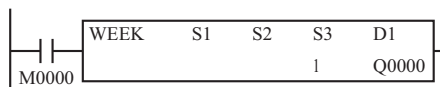
### 在配置的间隔内打开输入时的时间图

在 ON 设置和 OFF 设置之间的间隔内打开或关闭输入时，以及当脉冲输出有效时，在 ON 设置时并在已设置好日期的 0:00 以后打开或关闭输入时，时间图如下所示。

#### ■ 禁用脉冲输出时

##### 设置

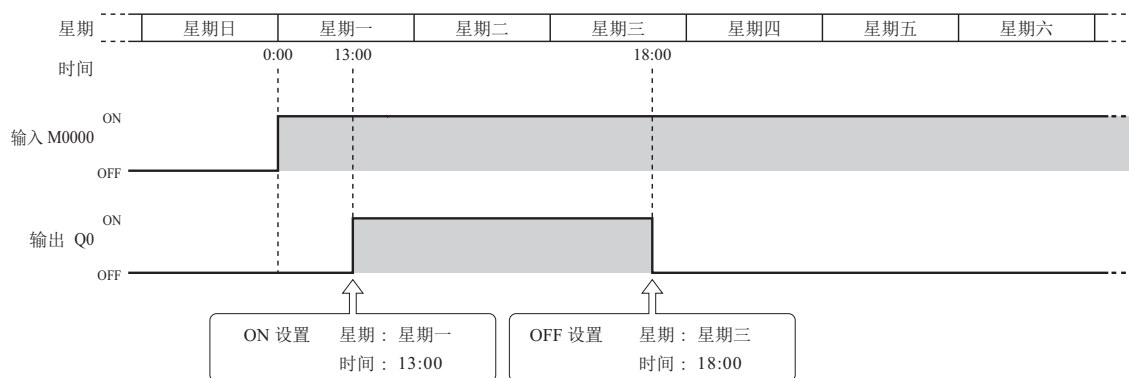
ON 设置	星期一 13:00
OFF 设置	星期三 18:00
输出端口	Q0



#### [ 当输入在 ON 设置指定的星期和时间前打开时 ]

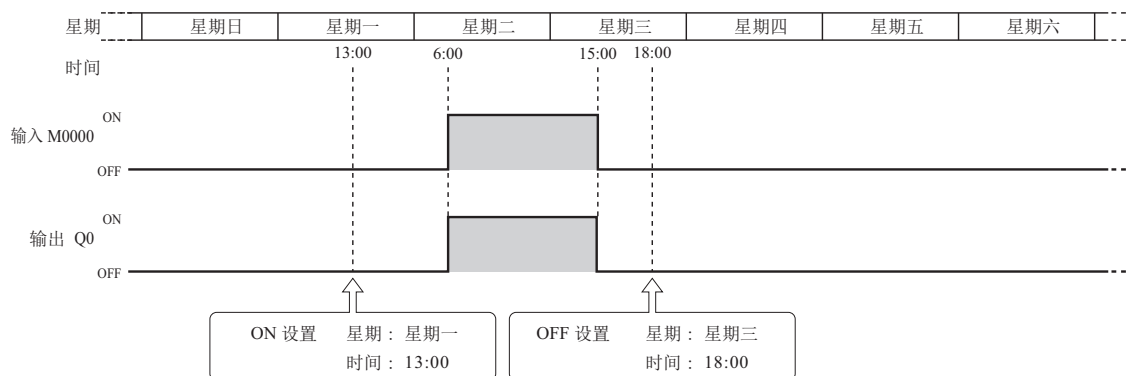
输入 M0000 在星期一 0:00 打开时与 ON 设置、OFF 设置进行比较的结果未处于 ON 设置和 OFF 设置之间（星期一 13:00 至星期三 18:00），因此，输出会保持关闭状态。

之后，将在星期一 13:00 打开输出 Q0，并在星期三 18:00 关闭。



#### [ 当输入在 ON 和 OFF 设置之间的间隔内打开或关闭时 ]

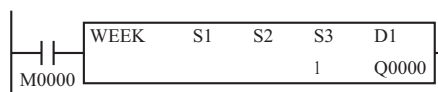
输入 M0000 在星期二 6:00 打开时与 ON 设置、OFF 设置进行比较的结果未处于 ON 设置和 OFF 设置之间（星期一 13:00 至星期三 18:00），因此输出变为关闭。之后，如果输入 M0000 在 OFF 设置之前关闭，则输出也同时关闭。



## ■ 启用脉冲输出时

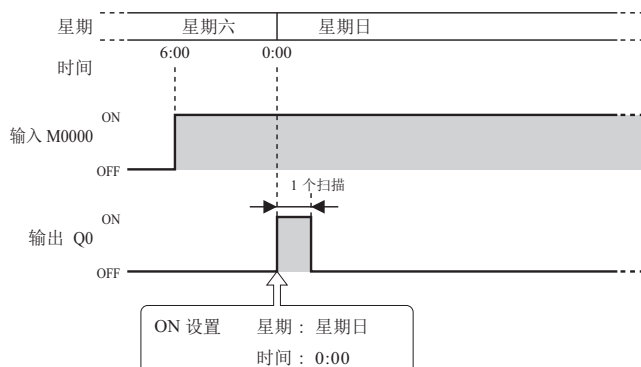
### 设置

ON 设置	星期日 0:00
输出端口	Q0



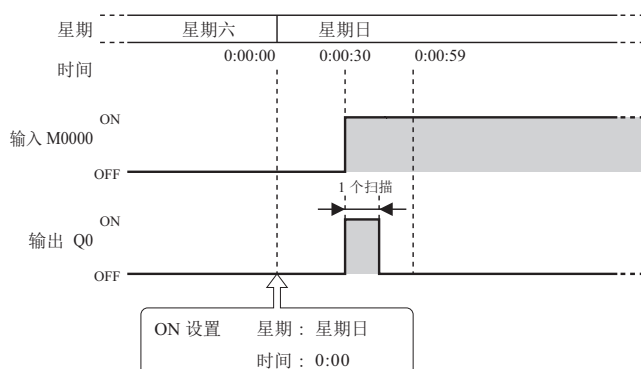
### [ 当输入在 ON 设置指定的星期和时间前打开时 ]

输入 M0000 将在星期日 0:00 打开，因此仅 1 个扫描会在星期日 0:00 打开输出 Q0。



### [ 当输入在 ON 设置指定的星期和时间中打开时 ]

输入 M0000 在星期日 0:00:30（星期日 0:00:00 至 0:00:59 之间）从关闭变为打开时，仅 1 个扫描会打开输出 Q1。

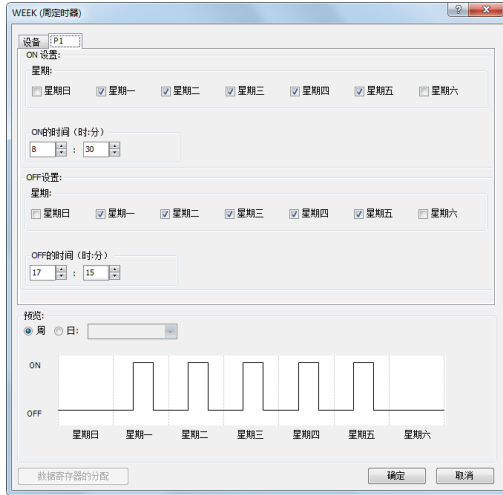


## 11: 周程序指令

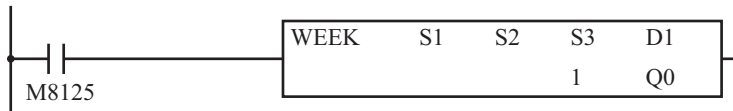
### 示例: WEEK

[ 每周星期一到星期五 8:30 到 17:15 打开输出 Q0 ]

#### “P (参数)” 选项卡

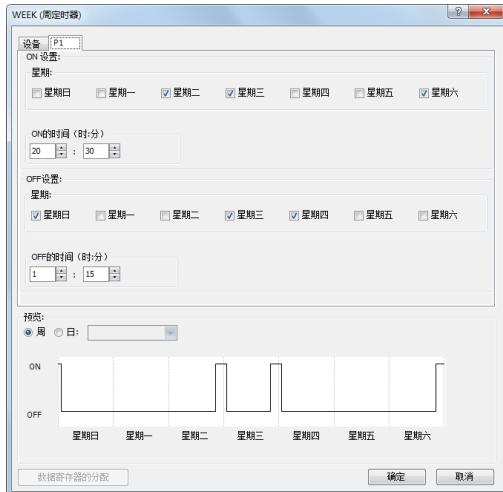


按如上所示配置选项卡，并将“D1” 设置为 Q0。

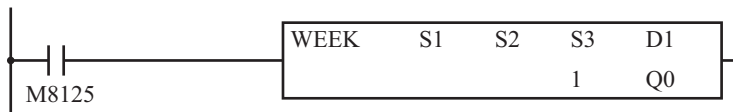


[ 每周星期二、星期三和星期六 20:30 到第二天 1:15 打开输出 Q0 ]

#### “P (参数)” 选项卡



按如上所示配置选项卡，并将“D1” 设置为 Q0。

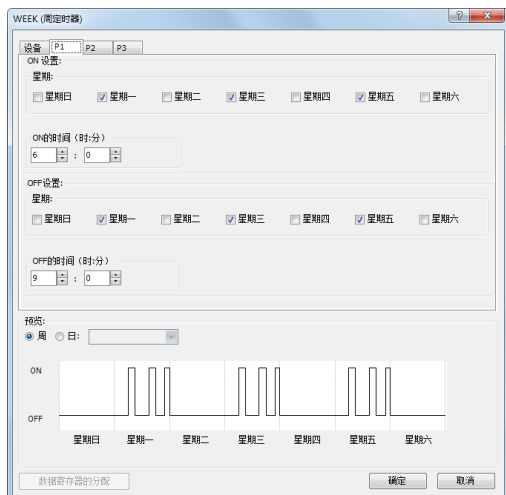


[每周星期一、星期三和星期五 6:00 到 9:00、15:00 到 18:00 以及 22:00 到第二天 0:00 打开输出 Q0]

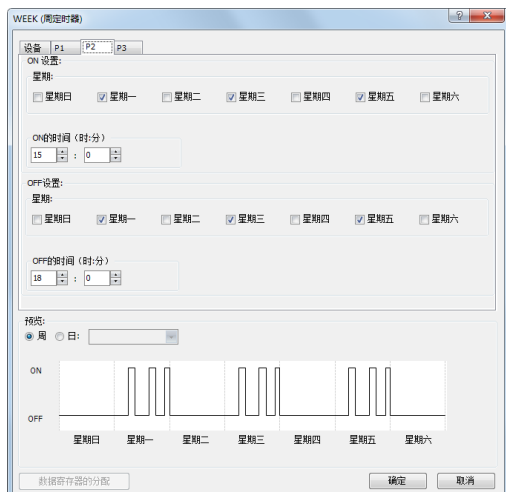
### “P（参数）”选项卡

使用三个选项卡配置设置。

在 P1 选项卡中，将输出配置为星期一、星期三和星期五 6:00 到 9:00 打开。

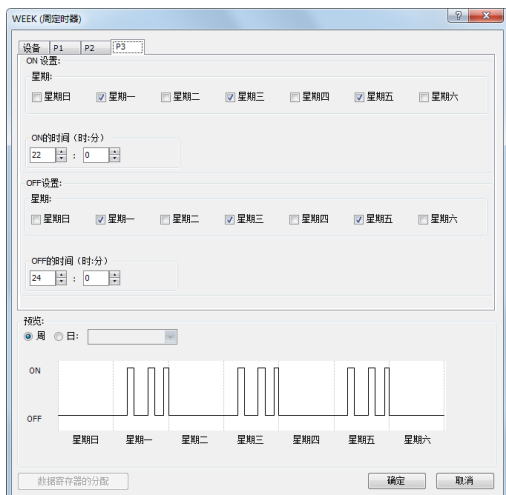


在 P2 选项卡中，将输出配置为星期一、星期三和星期五 15:00 到 18:00 打开。



## 11: 周程序指令

在 P3 选项卡中，将输出配置为星期一、星期三和星期五 22:00 到第二天 0:00 打开。



按如上所示配置选项卡，并将“D1”设置为 Q0。

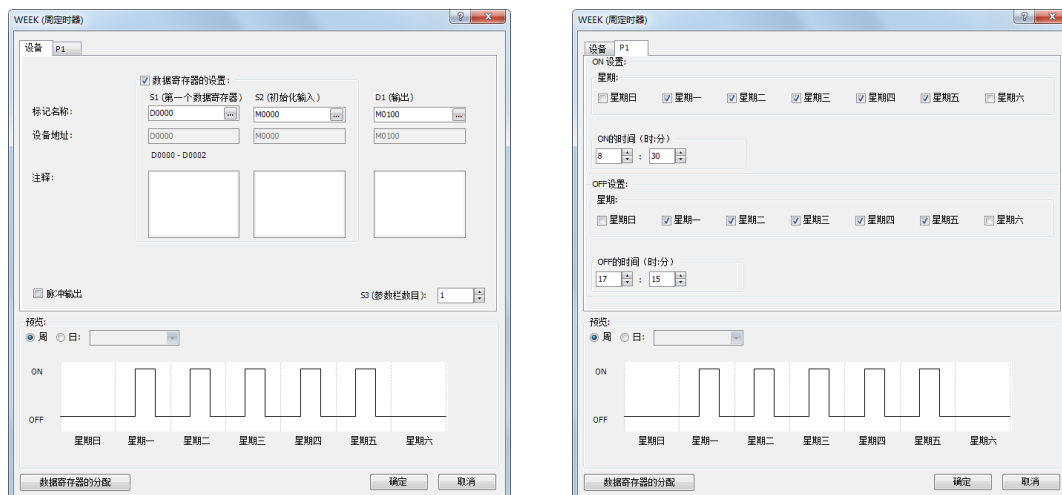


## [ 使用数据寄存器间接指定设置 ]

以下示例以每周星期一到星期五 8:30 到 17:15 打开输出 M0100 为例。

选中“数据寄存器的设置”复选框，并将 S1 设置为 D0000，将 S2 设置为 M0000。

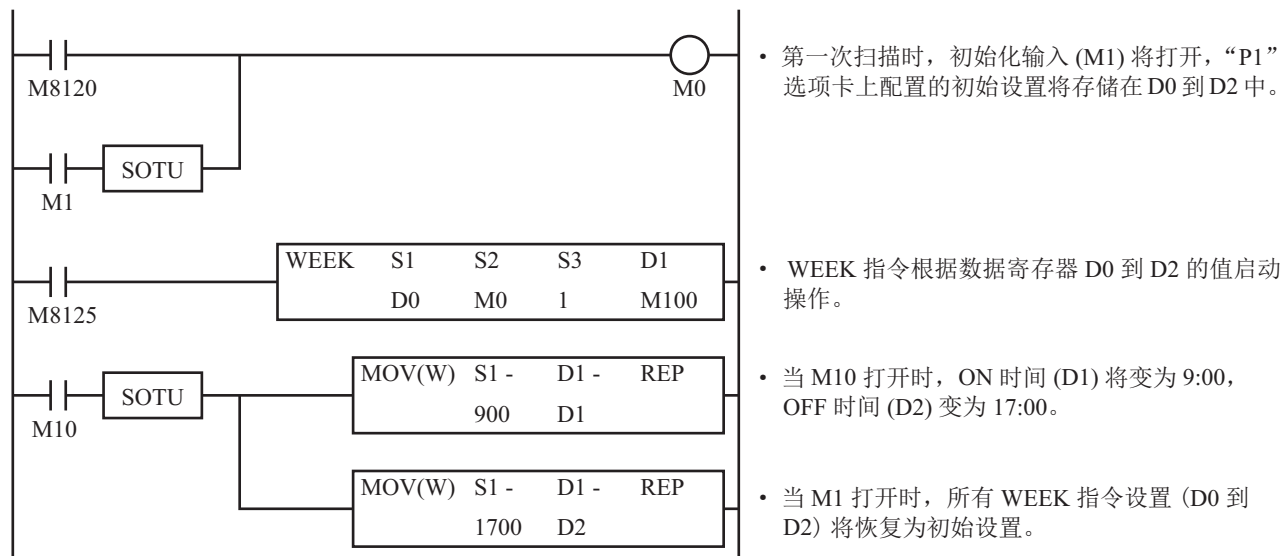
## “P1”选项卡



## 数据寄存器分配

“P1”选项卡上的设置将分配到数据寄存器 D0 到 D2 中，如下表所示。通过打开初始化输入 S2，可以把参数选项卡上的设置存储到 D0 到 D2。

数据寄存器	设置	初始设置	
D0	P1 选项卡	15934 (星期一到星期五, 包括 ON 设置和 OFF 设置)	
D1			ON 时间
D2			OFF 时间

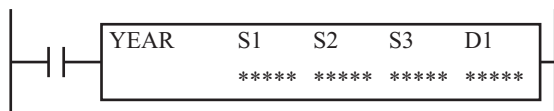


## 11: 周程序指令

### YEAR（年定时器）

将指定日期与当前日期进行比较，并输出其结果。可在一年期限内指定特殊日期。

FT2J FC6A

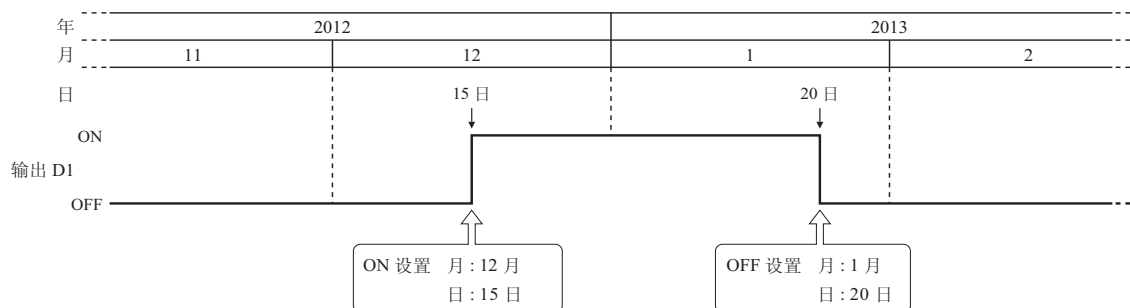


YEAR 指令将指定的日期与当前日期进行对比并输出该结果。使用此指令，您可以在一年期限内指定特殊日期（“特别日期”是指使用 YEAR 指令指定的 ON/OFF 设置日期）。

当输入打开时，D1 指定的输出将在 ON 设置指定的日期与当前日期匹配时打开。

在 OFF 设置指定的日期与当前日期匹配时，系统将关闭输出。

如果 ON 设置为 2012 年 12 月 15 日，OFF 设置为 2013 年 1 月 20 日，则输出 D1 将按如下所示打开和关闭。



#### 注释:

- 一个用户程序中最多可以使用 10 条 YEAR 指令。
- 通常，仅在当前日期和 ON/OFF 设置的日期匹配时，系统才会更新输出，但当 YEAR 指令输入从关闭变为打开时，当前日期的输出状态将根据 ON/OFF 设置来确定，从而打开或关闭输出。有关详情，请参见第 11-25 页上的“在配置的间隔内打开输入时的时间图”。
- 如果启用脉冲输出，则系统将打开输出以便在日期变为 ON 的一瞬间 (0:00) 进行一次扫描。当 YEAR 指令输入从关闭变为打开时，系统将根据 ON/OFF 设置判断当前日期的输出状态，并且仅打开输出进行一次扫描。有关脉冲输出，请参见第 11-18 页上的“6. 脉冲输出”或第 11-22 页上的“6. 脉冲输出”。
- 如果启用了年份设置或月份设置，并且根据月份或年份不存在的某个日期设置了 ON 或 OFF 设置，则会在下个月的第一天打开或关闭输出。
- 对于 FT2J 型，在执行 YEAR 指令之前，请检查 HMI 功能启动完成标志 (M8022) 是否为 ON。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 如果年数据在 2000 到 2099 范围之外、月数据在 1 到 12 范围之外、日数据在 1 到 31 范围之外、周数据在 1 到 5 以及最后范围之外、星期数据在 0 到 6 范围之外，将存储错误代码 10 在用户程序执行错误代码 (D8006) 中。
- 中断程序中不能使用 YEAR 指令。如果使用该指令，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	第一个数据寄存器	—	—	—	—	—	—	▲	—	—	—
S2 (源 2)	初始化输入	—	—	X	—	—	—	—	—	—	—
S3 (源 3)	参数选项卡数目	—	—	—	—	—	—	—	—	X	—
D1 (目标 1)	输出端口	—	X	▲	—	—	—	—	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

▲ 不能将特殊数据寄存器指定为 S1。不能将特殊内部继电器指定为 D1。



## 设置

配置 YEAR 指令的日期有两种方法。在“设备”选项卡上，通过“数据寄存器设置”指定方法。

- 将日期配置为固定设置

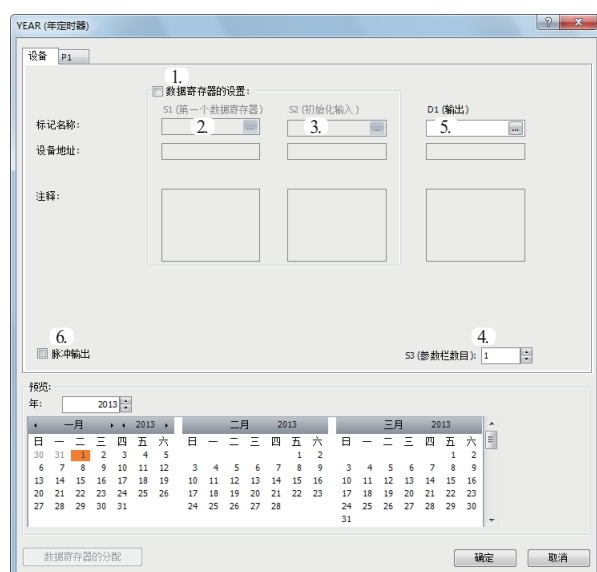
ON/OFF 设置的日期单独确定。在梯形图程序运行时，无法更改 ON/OFF 设置的日期。有关详情，请参见第 11-17 页上的“将日期配置为固定设置”。

- 通过指定数据寄存器配置日期

ON/OFF 设置的日期根据指定数据寄存器中存储的值来配置。在梯形图程序运行时，可以更改 ON/OFF 设置的日期。有关详情，请参见第 11-21 页上的“通过指定数据寄存器配置日期”。

## 将日期配置为固定设置

- “设备”选项卡



### 1. 数据寄存器的设置

此选项用于区分是选择日期的固定配置，还是选择通过数据寄存器的间接指定。

要将日期配置为固定设置，请清除此复选框。

#### 未选中的复选框

日期为固定设置。日期在参数选项卡上配置。在梯形图程序运行时，无法更改日期。有关设置，请参见第 11-5 页上的““P（参数）”选项卡”。

**注释：**选中此复选框时，日期设置将通过数据寄存器间接指定。使用数据寄存器配置日期。在梯形图程序运行时，可以更改日期。有关通过数据寄存器进行的间接指定，请参见第 11-21 页上的“通过指定数据寄存器配置日期”。

### 2. S1（源 1）：第一个数据寄存器

将日期配置为固定设置时，系统不会配置此设置。

### 3. S2（源 2）：初始化输入

将日期配置为固定设置时，系统不会配置此设置。

### 4. S3（源 3）：参数栏数目

此设置配置参数选项卡的数目。当此值增加或减小时，对话框中显示的参数选项卡数也会增加或减小。每个“P（参数）”选项卡占用 10 个字节的用户程序区域。

有关设置，请参见第 11-18 页上的“参数选项卡”。

## 11: 周程序指令

### 5. D1（目标 1）：输出

此设置用于配置对比配置的日期与当前的日期时的结果输出。

设置	说明
标记名称	指定每台设备的标记名称或设备地址。
设备地址	显示与标记名称相对应的设备地址。
备注	显示设备地址的备注。此项可进行编辑。

### 6. 脉冲输出

此设置配置 D1 的操作（输出）。此设置将应用于所有参数选项卡。

#### 已选中的复选框

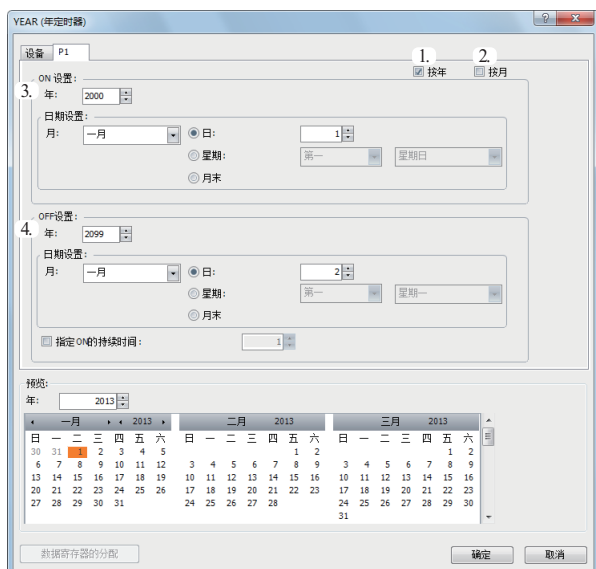
在当前日期变为 ON 设置中的日期时，系统将打开输出进行一次扫描。

#### 未选中的复选框

系统将根据 ON 设置和 OFF 设置来打开和关闭输出。

### • 参数选项卡

此选项卡配置输出的设置。1 条 YEAR 指令最多可以配置 20 个参数选项卡。



#### 1. 按年

选择“按年”时，“日期设置”每年都有效。

此时，如果要使“日期设置”从某年到某年的期间内都有效或无效的话，可以通过“年”来设定。

#### 2. 按月

所选选项卡的设置每月都有效。此时，“月”被禁用。

#### 3. ON 设置

此部分配置要打开输出的日期。系统将在所配置日期的 0:00 打开输出。

设置	说明	范围	
年	指定要打开输出的年份。	2000 到 2099	
日期设置	月	指定要打开输出的月份。	一月到十二月
	日	指定要打开输出的日期。	1 到 31
	星期	指定要打开输出的星期。在第一周到第五周以及最后一周的范围内设置该星期。	周：第一到第五，最后 星期：星期日到星期六
	月末	希望在指定月份的最后一天打开输出时选择此项。	—

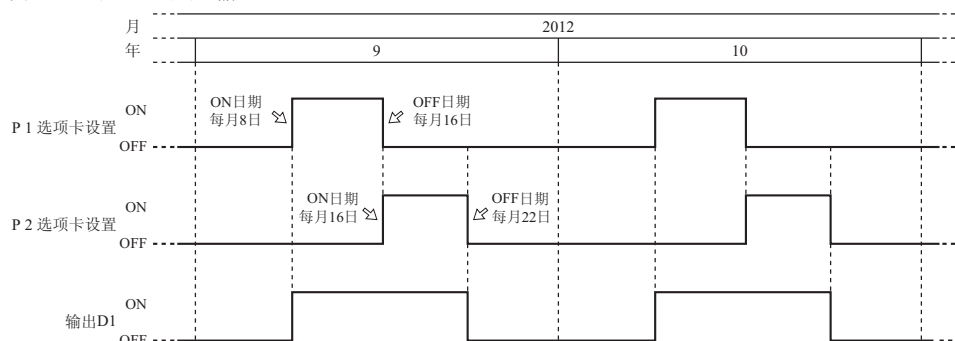
## 4. OFF 设置

此部分配置要关闭输出的日期。系统将在所配置日期的 0:00 关闭输出。

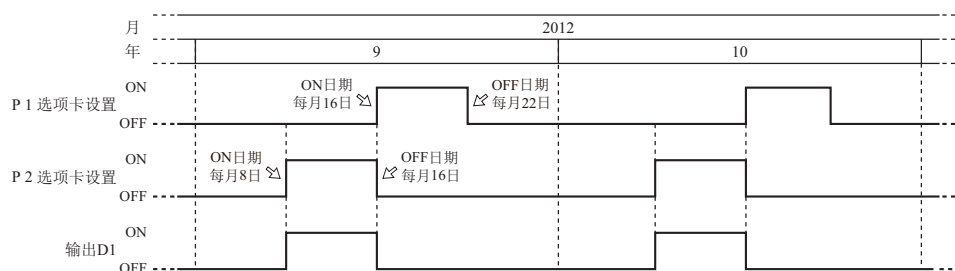
设置	说明	范围
年	指定要关闭输出的年份。	2000 到 2099
日期设置	月	指定要关闭输出的月份。
	日	指定要关闭输出的日期。
	星期	指定要关闭输出的星期。在第 1 周到第 5 周以及最后一周的范围 内设置该星期。
	月末	希望在指定月份的最后一天关闭输出时选择此项。
指定 ON 的持续时间	此项指定自输出打开之日起经过指定的天数后是否关闭输出。 如果启用此设置，则系统将禁用 OFF 设置。此设置可在 1 天到 30 天的范围内设置。	1 到 30

**注释：**如果日期在其他选项卡上的设置中重复，则编号较大的选项卡上的设置有效。

例如，如果“P1”选项卡上的 ON 日期为每月 8 日，OFF 日期为每月 16 日，而“P2”选项卡上的 ON 日期为每月 16 日，OFF 日期为每月 22 日，则每月 16 日设置在这两个选项卡上重复，“P2”选项卡的 ON 设置将有效。在这种情况下，系统将在每月 8 日到 22 日打开输出。

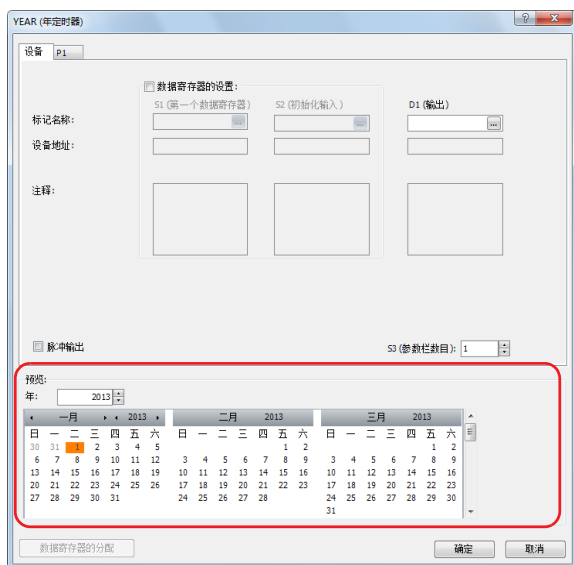


如果“P1”选项卡上的 ON 日期为每月 16 日，OFF 日期为每月 22 日，而“P2”选项卡上的 ON 日期为每月 8 日，OFF 日期为每月 16 日，则每月 16 日设置在这两个选项卡上重复，系统将禁用“P1”选项卡的 ON 设置。在这种情况下，系统将在每月 8 日到 16 日打开输出。



## 11: 周程序指令

### • 预览



预览将根据参数选项卡上配置的设置，以日历的形式显示输出的 ON/OFF 状态。设置为 ON 的日期将以橙色突出显示。一次显示三个月。

设置	说明
年	指定要在预览中显示的年份。
滚动条	您可以通过移动滚动条来更改预览中显示的月份。

## 通过指定数据寄存器配置日期

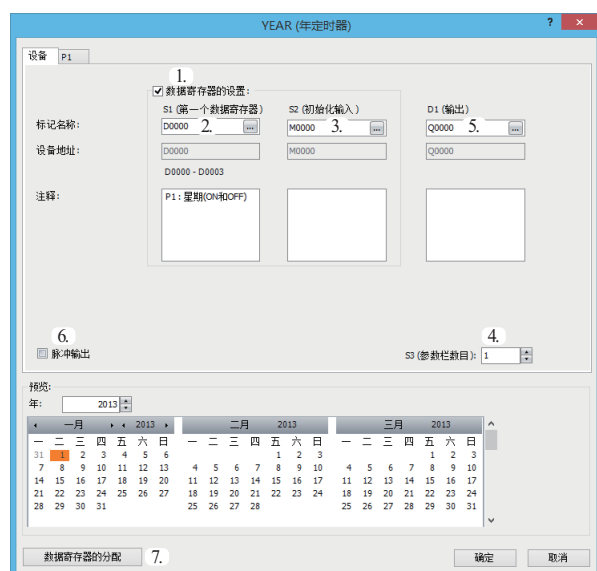
ON/OFF 设置的日期根据指定数据寄存器中存储的值来配置。

在梯形图程序运行时，可以更改 ON/OFF 设置的日期。

**注释：**在打开 YEAR 指令输入的状态下，更改了 ON/OFF 设置的特别日期时，当前日期时间与更改后的特别日期达到一致之前，将不会反映至输出动作。

如需立即反映至输出动作，请在更改特别日期后，先关闭 YEAR 指令，然后重新打开。

### • “设备”选项卡



### 1. 数据寄存器的设置

选择固定设置 YEAR 指令的特别日期或通过数据寄存器间接指定 YEAR 指令的特别日期。

要通过指定数据寄存器配置日期，请选中此复选框。

#### 已选中的复选框

日期设置由数据寄存器间接指定。使用数据寄存器配置日期。在梯形图程序运行时，可以更改日期。有关分配数据寄存器区域，请参见第 11-23 页上的“数据寄存器分配”。

通过打开初始化输入，可以使用参数选项卡中配置的星期和时间来初始化数据寄存器。有关初始化，请参见第 11-21 页上的“3. S2 (源 2)：初始化输入”。

**注释：**如果未选中此复选框，则日期为固定设置。日期在参数选项卡上配置。在梯形图程序运行时，无法更改日期。有关固定设置，请参见第 11-17 页上的“将日期配置为固定设置”。

### 2. S1 (源 1)：第一个数据寄存器

此设备为数据寄存器区域的起始地址，用于存储 YEAR 指令的日期的设置。

仅当使用数据寄存器间接指定日期时，才会使用此设置。

设置	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示与标记名称相对应的设备地址。
使用的数据寄存器	显示用于存储设置的数据寄存器的范围。当设备地址或参数选项卡数变化时，此项也会随之变化。
备注	显示设备地址的备注。此项可进行编辑。

有关分配数据寄存器区域，请参见第 11-23 页上的“数据寄存器分配”。

### 3. S2 (源 2)：初始化输入

此设置指定的设备用于初始化以 S1(源 1) 为起始地址的数据寄存器区域内存储的日期。

通过打开初始化输入，可以在数据寄存器中存储参数选项卡上配置的值。

仅当使用数据寄存器间接指定 YEAR 指令的设置时，才会使用此设置。

## 11: 周程序指令

### 4. S3（源 3）：参数栏数目

此设置配置参数选项卡的数目。

此设置与“将日期配置为固定设置”的效果相同。请参见第 11-17 页上的“4. S3（源 3）：参数栏数目”。

### 5. D1（目标 1）：输出

此设置用于配置与当前的日期对比结果的输出。

此设置与“将日期配置为固定设置”的效果相同。请参见第 11-18 页上的“5. D1（目标 1）：输出”。

### 6. 脉冲输出

此设置配置 D1 的操作（输出）。此设置将应用于所有参数选项卡。

此设置与“将日期配置为固定设置”的效果相同。请参见第 11-18 页上的“6. 脉冲输出”。

### 7. 数据寄存器的分配

单击此按钮可显示“设备分配”对话框。如下所示，对话框 (8) 中将显示数据寄存器及其对应的 YEAR 指令设置的表格。单击“分配注释”(9) 以后，您可以为与设置名称相对应的数据寄存器配置备注。

仅当使用数据寄存器间接指定 YEAR 指令的设置时，才会使用此按钮。

- “设备分配”对话框

- 参数选项卡



- “P（参数）”选项卡

此选项卡配置输出的设置。1 条 YEAR 指令最多可以配置 20 个参数选项卡。

如果使用数据寄存器间接指定 YEAR 指令的设置，则在打开初始化输入后，参数选项卡上配置的设置将存储到数据寄存器中。

此设置与“将日期配置为固定设置”的效果相同。请参见第 11-18 页上的“参数选项卡”。

- 预览

预览将根据参数选项卡上配置的设置，以日历的形式显示输出的 ON/OFF 状态。

此功能与“将日期配置为固定设置”的效果相同。请参见第 11-20 页上的“预览”。

## 数据寄存器分配

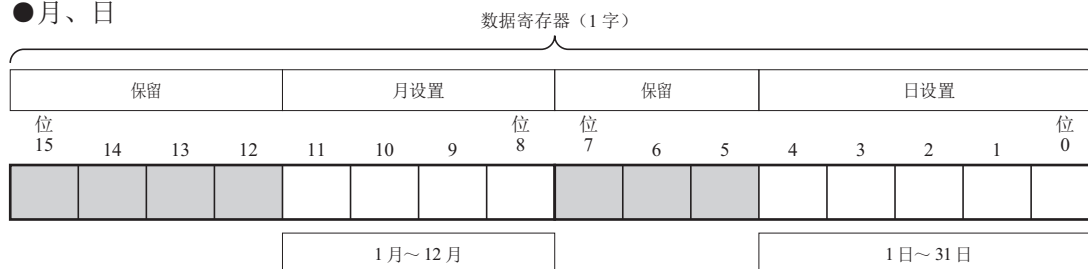
“参数”选项卡上配置的设置将按如下所示分配到数据寄存器中。

存储目的地	设置		
起始地址 +0	P 1 选项卡	ON 设置	年
起始地址 +1			月、日或星期
起始地址 +2		OFF 设置	年
起始地址 +3			月、日或星期（指定保持打开的天数）
起始地址 +4	P 2 选项卡	ON 设置	年
起始地址 +5			月、日或星期
起始地址 +6		OFF 设置	年
起始地址 +7			月、日或星期（指定保持打开的天数）
⋮			⋮
起始地址 +76	P 20 选项卡	ON 设置	年
起始地址 +77			月、日或星期
起始地址 +78		OFF 设置	年
起始地址 +79			月、日或星期（指定保持打开的天数）

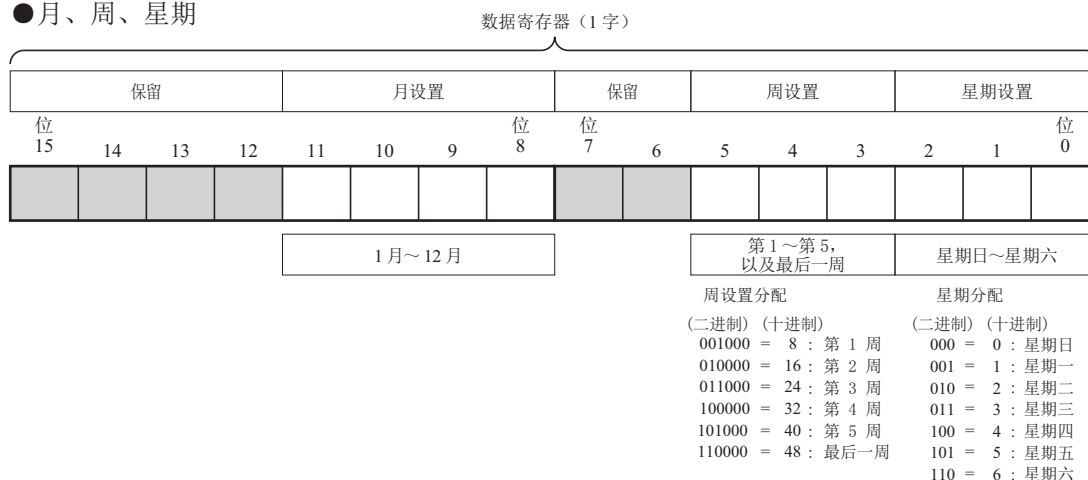
• 月、日或星期数据寄存器分配

月、日或星期将按位在 1 个数据寄存器中进行分配，如下所示。

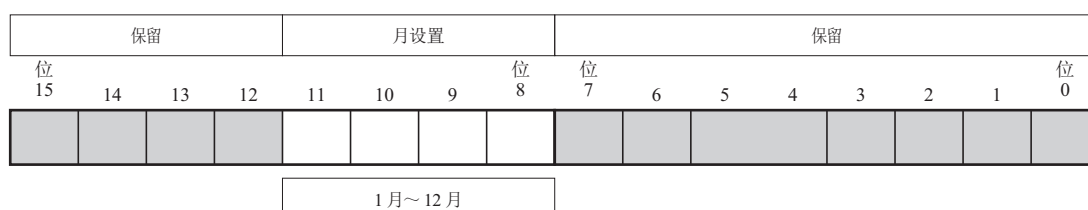
● 月、日



● 月、周、星期



● 月、月末



## 11: 周程序指令

### 星期设置示例

#### [ 当配置为 1 月 1 日打开输出时 ]

1 月 1 日

保留				月设置				保留			日设置								
位	15	14	13	12	11	10	9	位	8	位	7	6	5	4	3	2	1	位	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	

月设置: 0001 = 1  
 日设置: 00001 = 1  
 数据寄存器的值为100000001 (二进制) = 257 (十进制)。

#### [ 当配置为 12 月 31 日打开输出时 ]

12 月 31 日

保留				月设置				保留			日设置								
位	15	14	13	12	11	10	9	位	8	位	7	6	5	4	3	2	1	位	0
	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	

月设置: 1100 = 12  
 日设置: 11111 = 31  
 数据寄存器的值为110000011111 (二进制) = 3103 (十进制)。

#### [ 当配置为 1 月的第 1 个星期一打开输出时 ]

1 月的第 1 个星期一

保留				月设置				保留			周设置		星期设置						
位	15	14	13	12	11	10	9	位	8	位	7	6	5	4	3	2	1	位	0
	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	

月设置: 0001 = 1 (1 月)  
 周设置: 001 = 1 (第 1)  
 星期设置: 001 = 1 (星期一)  
 数据寄存器的值为100001001 (二进制) = 265 (十进制)。

#### [ 当配置为 6 月的第 4 个星期四打开输出时 ]

6 月的第 4 个星期四

保留				月设置				保留			周设置		星期设置						
位	15	14	13	12	11	10	9	位	8	位	7	6	5	4	3	2	1	位	0
	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	

月设置: 0110 = 6 (6 月)  
 周设置: 100 = 4 (第 4)  
 星期设置: 100 = 4 (星期四)  
 数据寄存器的值为11000100100 (二进制) = 1572 (十进制)。

#### [ 当配置为最后打开输出时 ]

最后

保留				月设置				保留			周设置		星期设置						
位	15	14	13	12	11	10	9	位	8	位	7	6	5	4	3	2	1	位	0
	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0	

月设置: 0110 = 6 (6 月)  
 周设置: 110 = 6 (最后)  
 星期设置: 100 = 4 (星期四)  
 数据寄存器的值为11000110100 (二进制) = 1588 (十进制)。



## 在配置的间隔内打开输入时的时间图

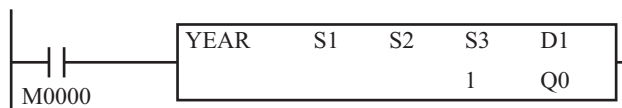
在 ON 设置和 OFF 设置之间的间隔内打开或关闭输入时，以及当脉冲输出有效时，在 ON 设置时并在已设置好日期的 0:00 以后打开或关闭输入时，时间图如下所示。

### • 禁用脉冲输出时

当输入打开时，系统会将当前的日期与 ON 和 OFF 设置进行对比，并打开或关闭输出。

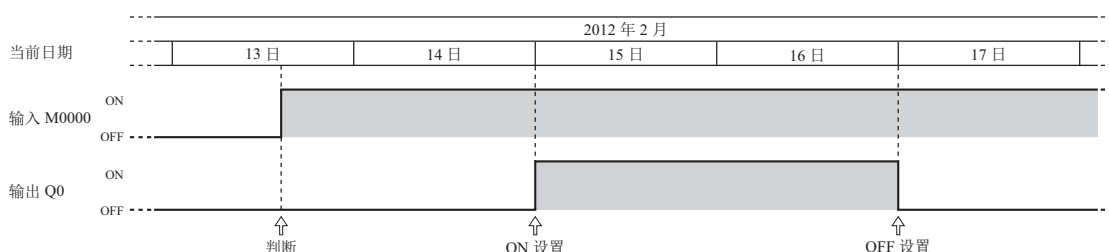
#### 设置

ON 设置	2012 年 2 月 15 日
OFF 设置	2012 年 2 月 17 日
输出端口	Q0



### [ 当输入在 ON 设置指定的日期前打开时 ]

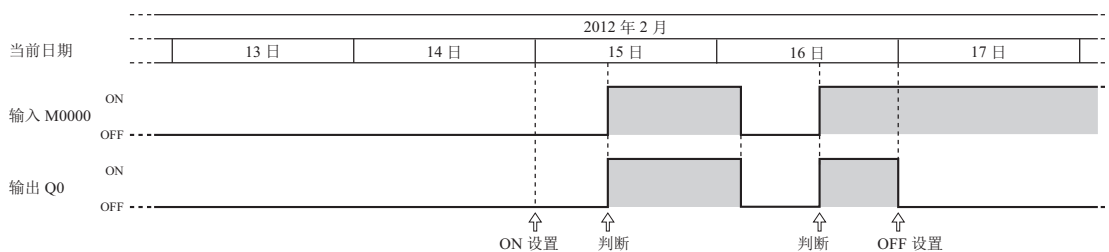
即使在 2012 年 2 月 13 日输入被打开，但当前日期与 ON、OFF 设置的对比结果并没有落在 ON、OFF 的设置范围 (2012 年 2 月 15 日到 2 月 17 日) 内，因此，输出会保持关闭状态。



### [ 当输入在 ON 和 OFF 设置之间的间隔内打开或关闭时 ]

在 2012 年 2 月 15 日打开输入时，当前日期与 ON 和 OFF 设置进行对比的结果在 ON 和 OFF 设置的范围 (2012 年 2 月 15 日到 2 月 17 日) 内。因此，输出会打开。当输入关闭时，输出状态会被关闭。

在 2012 年 2 月 16 日打开输入时，当前日期与 ON 和 OFF 设置进行对比的结果是输出会被打开。



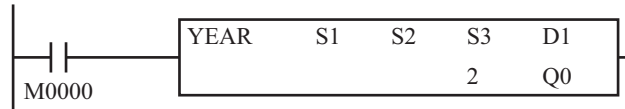
## 11: 周程序指令

### • 启用脉冲输出时

在 ON 设置的日期的 0:00 时确定打开或关闭输入，并且打开输出。当输入打开时，当前日期不与 ON 设置进行对比。

#### 设置

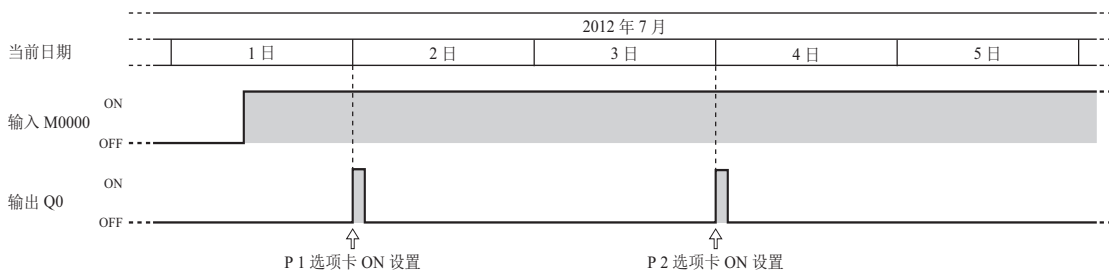
P 1 选项卡 ON 设置	2012 年 7 月 2 日
P 2 选项卡 ON 设置	2012 年 7 月 4 日
输出端口	Q0



#### [ 当输入在 ON 设置指定的日期前打开时 ]

在 2012 年 7 月 2 日 0:00 确定的输入结果是打开输出以进行一次扫描。

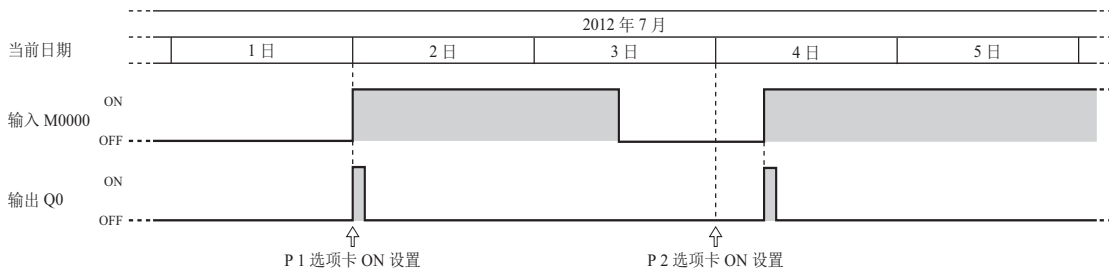
在 2012 年 7 月 4 日 0:00 确定的输入结果是打开输出以进行一次扫描。



#### [ 当输入正好在 ON 设置指定的日期打开时 ]

当输入在 2012 年 7 月 2 日 0:00 打开时，系统将打开输出以进行一次扫描。

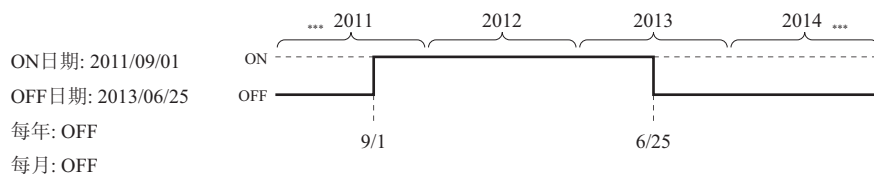
当输入在 2012 年 7 月 4 日 0:00 后打开时，系统将打开输出以进行一次扫描。



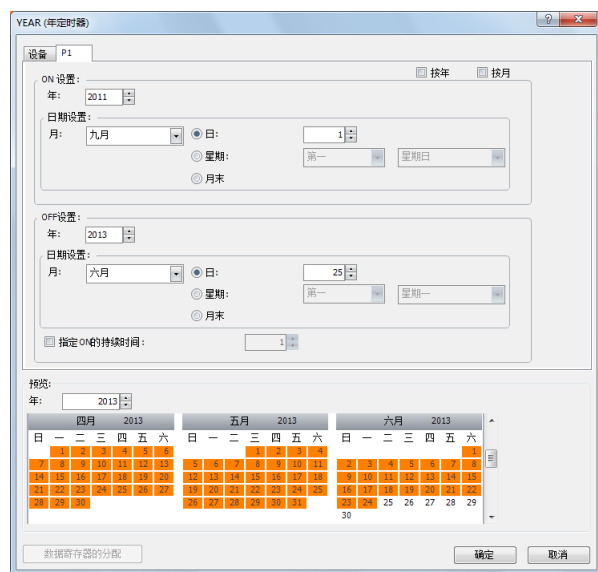
## 示例：YEAR

- 将日期配置为固定设置

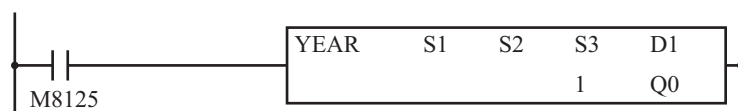
[ 从 2011 年 9 月 1 日 0:00 到 2013 年 6 25 日 0:00 打开 Q0 ]



## “P（参数）”选项卡



按如上所示配置选项卡，并将“D1”设置为 Q0。



## 11: 周程序指令

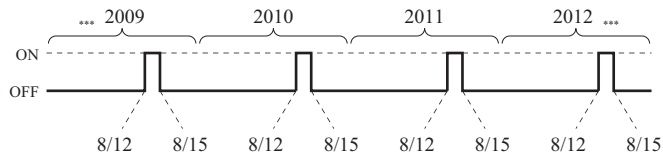
[从每年8月12日0:00到8月15日0:00打开输出Q0]

ON日期: 2000/08/12

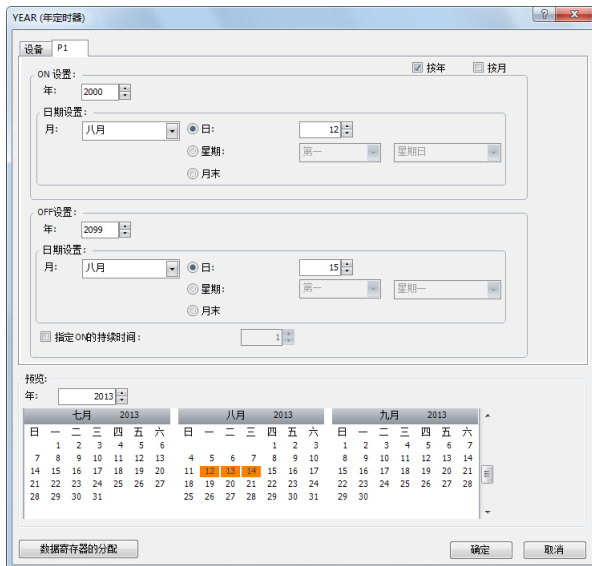
OFF日期: 2009/08/15

每年: ON

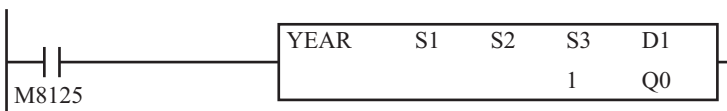
每月: OFF



“P（参数）”选项卡

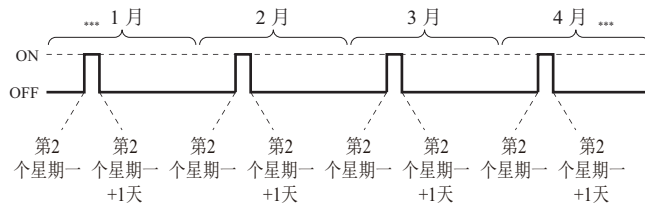


按如上所示配置选项卡，并将“D1”设置为Q0。

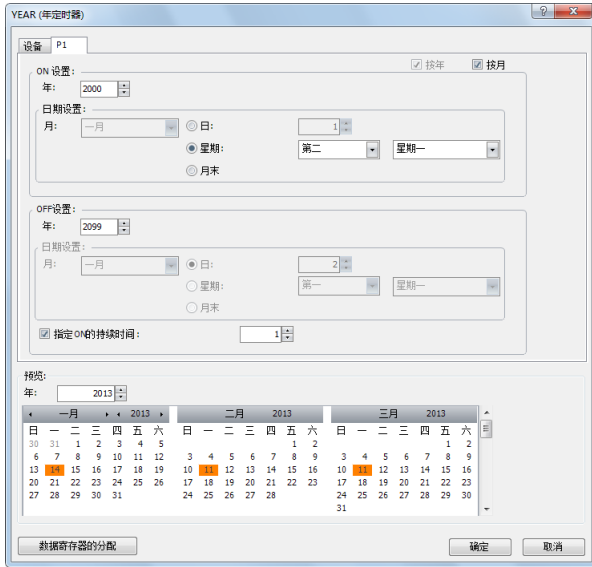


[ 从 2000 年到 2099 年 仅在每月的第 2 个星期一打开输出 Q0 ]

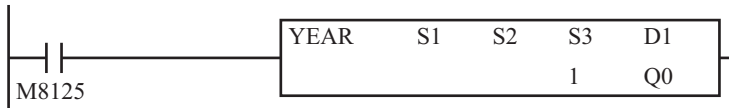
ON日期: 2000/\*\*/第2个星期一  
 OFF日期: 2099/\*\*/第2个星期一+1天  
 每年: ON  
 每月: ON



“P (参数)” 选项卡



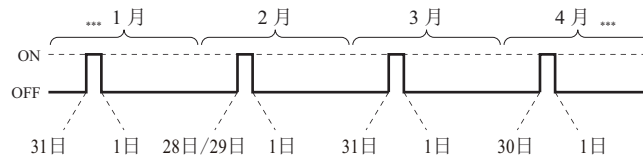
按如上所示配置选项卡，并将“D1”设置为 Q0。



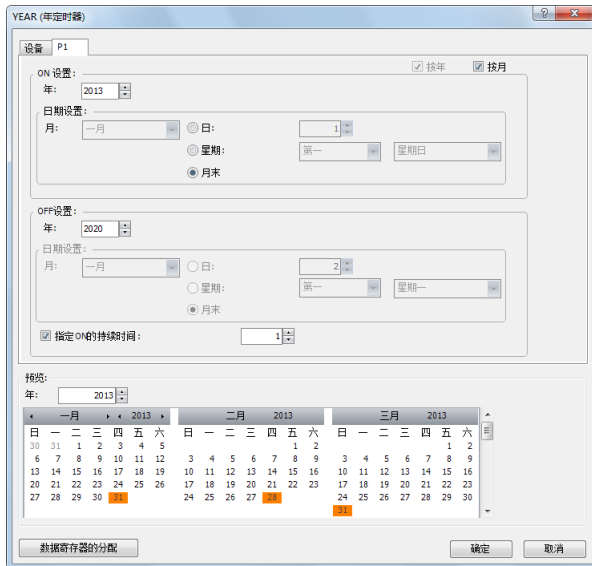
## 11: 周程序指令

[2013 年至 2020 年期间，仅在月末打开输出 Q0 时]

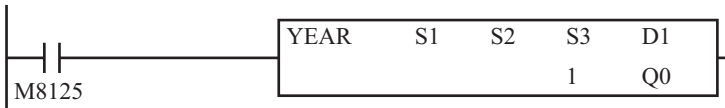
ON年月日: 2013/\*\*/月末  
 OFF年月日: 2020/\*\*/月末+1日  
 每年: ON  
 每月: ON



### “P（参数）”选项卡

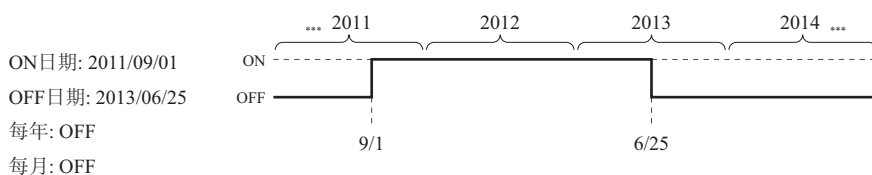


按如上所示配置选项卡，并将“D1”设置为 Q0。

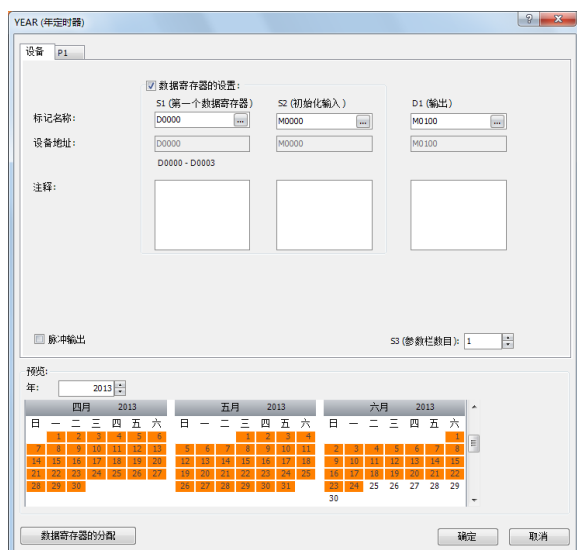


- 通过指定数据寄存器配置日期

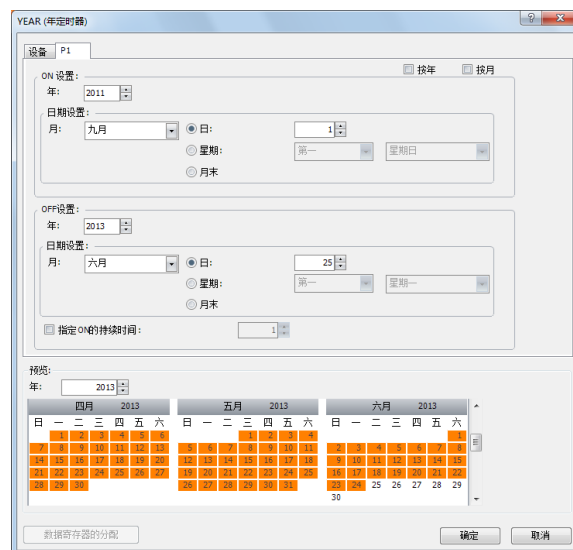
[从 2011 年 9 月 1 日 0:00 到 2013 年 6 25 日 0:00 打开 M100]



## “设备”选项卡



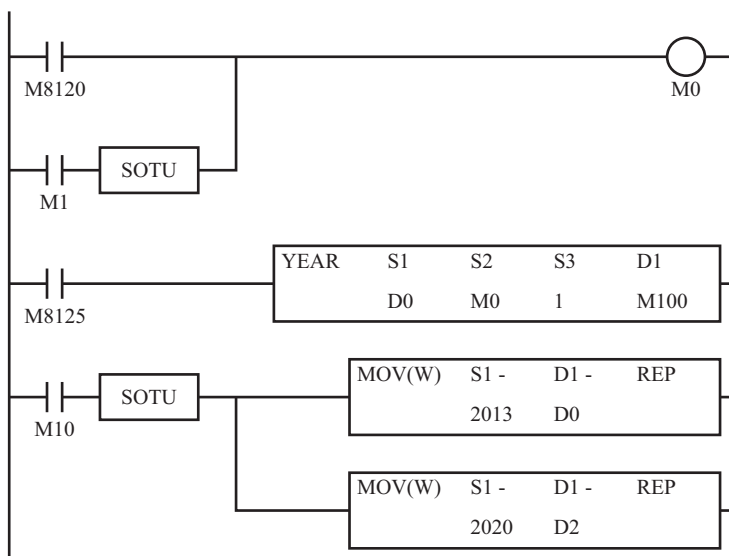
## “P (参数)”选项卡



## 数据寄存器分配

“P1”选项卡上的设置将分配到数据寄存器 D0 到 D3 中，如下表所示。通过打开始化输入 S2，可以将“P1”选项卡上的设置存储到 D0 到 D2 上。

数据寄存器	设置	初始设置
D0	P1 选项卡	ON 设置
D1		年
D2	OFF 设置	月、日
D3		年
		月、日



- 第一次扫描时，“P1”选项卡上配置的初始设置将存储在 D0 到 D3 中。
- 当 M1 打开时，所有 YEAR 指令设置 (D0 到 D3) 将恢复为初始设置。
- YEAR 指令根据数据寄存器 D0 到 D3 的值启动操作。
- 当 M10 打开时，ON 设置年份将变为 2013 年 (D0)，OFF 设置年份将变为 2020 年 (D2)。





# 12: 显示指令

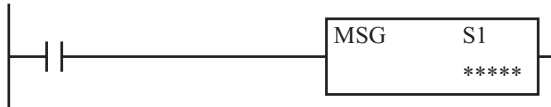
## 简介

本章将对将指定数据显示于 HMI 模块的 LCD 和外部设备的显示指令进行介绍。

## MSG（消息）

将指定数据显示于 HMI 模块的 LCD。

FT2J FC6A



输入打开时，按照“MSG（信息）”对话框中设置的内容，在 HMI 模块的 LCD 上显示信息。MSG 指令的功能如下所示。

### 可以显示设备值。

- 根据指定的数据类型，可以数值形式显示设备值。  
有关详情，请参见第 12-4 页上的“插入设备”。
- 可以条形图形式显示设备值。  
有关详情，请参见第 12-8 页上的“插入条形图”。
- 根据位设备（输入 / 输出、内部继电器、移位寄存器、定时器触点或计数器触点）的值，可以显示和切换文本。  
有关详情，请参见第 12-6 页上的“插入位设备”。

### 可以显示任意值。

- 可以显示指定的文本。  
有关详情，请参见第 12-2 页上的““MSG（信息）”对话框”。

### 可以配置文本显示效果。

- 可以滚动、闪烁或反转文本。  
有关详情，请参见第 12-7 页上的“插入带效果的文本”。

### 可以显示日期 / 时间数据。

- LCD 上可以显示当前日期 / 时间和 MSG 指令输入的开启日期 / 时间。  
有关详情，请参见第 12-3 页上的“7. 特殊数据”。

### 可从 9 种语言中选择显示文本的语言。

- 文本可以使用 4 种类型的字符集以 9 种语言显示。  
有关详情，请参见第 12-9 页上的“MSG 指令常规设置”。

### 可以配置文本显示设置。

- 可以配置滚动单位、滚动速度和闪烁速度。  
有关详情，请参见第 12-9 页上的“MSG 指令常规设置”。

### 可以修改设备值。

- 使用 MSG 指令显示的设备的值，可通过 HMI 模块进行更改。  
有关详情，请参见第 12-17 页上的“修改 HMI 模块上的设备值”。

### 注释：

- 用户程序内最多可输入 50 个 MSG 指令。
- 中断程序期间不能使用 MSG 指令。如果在中断程序期间使用该指令，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	优先级	—	—	—	—	—	—	—	—	X	—

有效设备范围取决于控制器类型。有关详情，请参见第 2-1 页上的“设备地址”。

## 12: 显示指令

### 设置

有关 MSG 指令设置，有适用于个别 MSG 指令的设置和所有 MSG 指令通用的设置。

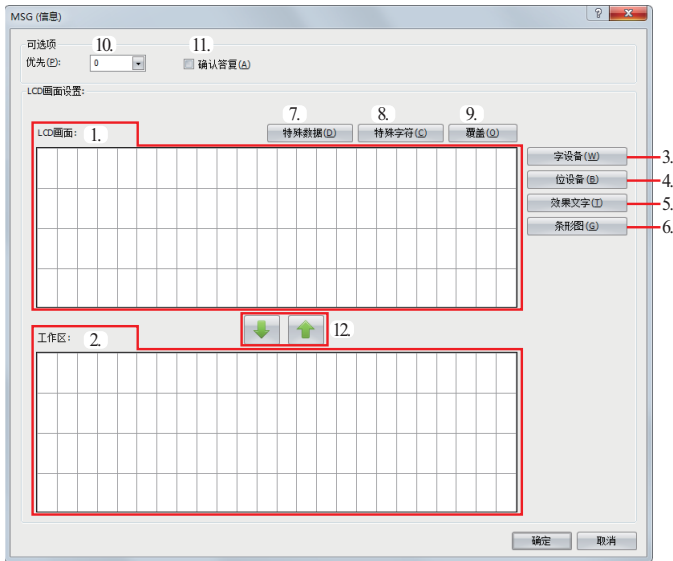
**注释：**在 WindLDR 的“功能设置”对话框中可修改所有 MSG 指令通用的设置。有关详情，请参见第 12-9 页上的“MSG 指令常规设置”。

#### ■ MSG 指令个别设置

个别设置可设置要显示的文本和设备以及优先级等。

个别设置在“MSG（信息）”对话框中进行设置。

#### “MSG（信息）”对话框



#### 1. LCD 画面

通过在使用鼠标光标选择的区域中配置字符或设备，可设置在 HMI 模块的 LCD 上显示的画面。如需输入字符，请通过键盘在光标位置输入。通过单击“插入”(9)可更改字符输入法（插入/覆盖）。

**注释：**使用键盘在 LCD 画面 (1) 上输入的文本不能设置为滚动、闪烁或反转。要滚动、闪烁或反转文本，请使用“效果文字”(5) 输入文本并配置显示选项。有关插入带效果的文本的详情，请参见第 12-7 页上的“插入带效果的文本”。

#### 2. 工作区

编辑 LCD 画面 (1) 屏幕时将使用此区域。此区域用于临时重新定位文本和设备数据。

光标位置上的字符或设备等数据，可使用  或  按钮 (12)，在 LCD 画面与工作区之间进行移动。

关闭对话框时，不会保存工作区数据。

#### 3. 字设备

在光标位置插入字设备。HMI 模块 LCD 上将显示指定字设备的值。有关详情，请参见第 12-4 页上的“插入字设备”。

#### 4. 位设备

在光标指定的区域插入位设备。根据指定位设备的值，可在 HMI 模块 LCD 上切换两个不同的文本项目并显示它们。有关详情，请参见第 12-6 页上的“插入位设备”。

#### 5. 效果文字

在光标指定的区域插入文本。HMI 模块 LCD 上将显示指定的文本。有关详情，请参见第 12-7 页上的“插入带效果的文本”。

#### 6. 条形图

在光标指定的区域插入条形图。在 HMI 模块 LCD 上，指定设备的值以条形图形式显示。有关详情，请参见第 12-8 页上的“插入条形图”。

## 7. 特殊数据

可在光标位置输入当前日期和时间等特殊数据。按下“特殊数据”时，在弹出的特殊数据列表窗口中选择要输入的数据。LCD 画面 (1) 屏幕上使用的区域大小因所选的特殊数据而异。

特殊数据	显示		占用的区域 (行 x 列)
	显示类型	显示示例 (2015 年 12 月 1 日, 13:30)	
当前日期	年 / 月 / 日	2015/12/01	1 x 10
当前时间	小时 : 分钟	13:30	1 x 5
开启 MSG 指令输入的日期	年 / 月 / 日	2015/12/01	1 x 10
开启 MSG 指令输入的时间	小时 : 分钟	13:30	1 x 5

## 8. 特殊字符

可在光标位置输入特殊字符。按下“特殊字符”时，在弹出的特殊字符列表窗口中选择要输入的字符。可使用的特殊字符如下所示。

特殊字符列表							
▼	▲	◀	▶	°C	°F	°	±



## 9. 插入 / 覆盖

选择插入或覆盖作为字符输入模式。单击此按钮可切换输入模式。

## 10. 优先

MSG 指令的优先级可以在 0 到 49 之间配置。0 是最高优先级，49 是最低优先级。

- 不能为多个 MSG 指令设置同一优先级。
- 当两个或多个 MSG 指令的输入开启时，在所有输入开启的 MSG 指令中将显示优先级最高的 MSG 指令的消息。
- 当优先级最高的 MSG 指令输入从开启变为关闭时，将显示下一最高优先级的 MSG 指令的消息（输入变化时检查优先级）。


**注释：**按下 HMI 模块上的 （向上）或 （向下）按钮时，将在输入开启的 MSG 指令消息之间切换。当 MSG 指令的“确认”启用后，也将在消息之间切换。


电动机 5 停止在 10:12 !! 行动 !!	示例：优先级 30 的信息输出
-----------------------------------	-----------------

电动机 2 3000 小时 !! 维护 !!	示例：优先级 10 的信息输出
---------------------------------	-----------------

运行中 2015/12/01/(二) 09:00:12	日期和当前时间
-----------------------------------	---------

## 11. 确认答复

启用确认答复后，即使 MSG 指令的输入关闭，也将持续显示信息。按下 HMI 模块的 （确定）键后，将隐藏信息，并同时显示输入打开的所有 MSG 指令中最高优先级的 MSG 指令内容。

启用了确认答复的 MSG 指令的输入为打开时，即使按下 （向上）按钮，信息也不会隐藏。

**注释：**按下 HMI 模块的确定键将会不再显示信息，并会显示该时刻输入为打开的 MSG 指令中优先级最高的 MSG 指令的内容。

## 12: 显示指令

### 12. ↓ / ↑按钮

在 LCD 画面 (1) 与工作区 (2) 之间移动光标位置上的字符或设备等数据。

当光标同时位于 LCD 画面 (1) 和工作区 (2) 上时, 按向下按钮可将光标在 LCD 画面 (1) 上选择的字符或设备移至工作区 (2)。按向上按钮可将光标在工作区 (2) 中选择的字符或设备移至 LCD 画面 (1)。

### 插入字设备

HMI 模块 LCD 上可显示指定字设备的值。

#### 1. 设备

输入要显示的设备。

有效设备

W (字)	TC、TP、CC、CP、D
I (整数)	D
D (双字)	CC、CP、D
L (长整数)	D
F (浮点)	D

#### 2. 数据类型和转换类型

选择指定设备的显示类型。LCD 屏幕上使用的区域大小因指定的数据类型和转换类型而异。

数据类型	转换类型	占用的区域	LCD 上的示例
W (字)	十进制	5	65535
	十六进制	4	FFFF
I (整数)	十进制	6	-32768
D (双字)	十进制	10	4294967295
	十六进制	8	FFFFFFFF
L (长整数)	十进制	11	-2147483648
F (浮点)	十进制	13	1.234567E-01

#### 3. 选项

设置指定设备的值闪烁显示或反转显示, 以及允许 / 禁止使用 HMI 模块编辑值。有关闪烁速度, 请参见第 12-9 页上的“MSG 指令常规设置”。

显示选项	说明
闪烁	闪烁指定设备的值。
反转	反转显示指定的值。
禁止编辑值	使用 HMI 模块, 禁止编辑指定设备的值。

#### 4. 占用区域

显示 LCD 屏幕上使用的区域大小（行：1，列：4 到 13）。区域大小由所选的数据类型和转换类型决定。

#### F（浮点）的 LCD 标记

根据 IEEE754 的单精度浮点数定义和 CPU 模块系统中包含的函数，决定使用小数点标记或指数标记来标记 F（浮点）的 LCD 标记。关于 IEEE754 中的单精度浮点数，请参见第 3-9 页上的“单存储格式”。

下表说明了这二个构成字段 e 和 f 的值与以单存储格式位模式表示的值之间的对应关系。值为  $\pm 0$ 、非范数以及范数时，将根据 CPU 模块中所包含的函数变为小数点标记。

值	指数字段 e	小数字段 f	WindLDR 中的表现形式
$\pm 0$	e=0	f=0	0.0
非范数	e=0	f $\neq$ 0	<ul style="list-style-type: none"> <li>小数标记时  <math>0.0001 \leq  \text{值}  &lt; 9,999,999.5</math>            （有效位数为 7 位，对第 8 位进行四舍五入）</li> <li>指数标记时  <math> \text{值}  &lt; 0.0001</math>  <math> \text{值}  \geq 9,999,999.5</math></li> </ul>
范数	$0 < e < 255$	任意值	
$\pm \infty$ （正负无穷）	e=255	f=0	INF
无效值		f $\neq$ 0	NAN

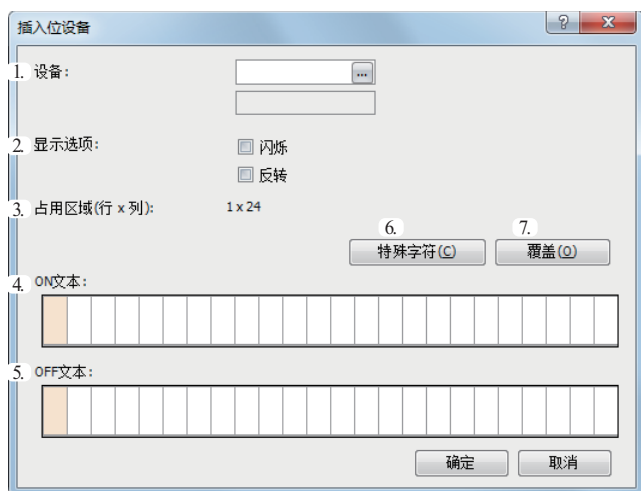
例）以下列值相对应的 LCD 标记为例进行显示。

值	LCD 的标记
1234567	1234567
12345678	1.2345678E+07
1234567.8	1234568
0.0001	0.0001
0.00001	1E-05
0.000001	1E-06
0.0000001	1E-07
0.123456	0.123456
0.1234567	0.1234567
0.12345678	0.1234568
0.0000012	1.2E-06
1.2345678	1.234568
0	0
0.0001234568	0.0001234568

## 12: 显示指令

### 插入位设备

根据指定位设备的值（开启时 / 关闭时），可在 HMI 模块 LCD 上切换显示两个不同的文本项目。



#### 1. 设备

输入要显示的设备。

有效设备

I	Q	M	R	T	C	D	P
X	X	X	X	X	X	—	—

#### 2. 显示选项

配置文本闪烁或反转的选项。有关闪烁速度，请参见第 12-9 页上的“MSG 指令常规设置”。

显示选项	说明
闪烁	闪烁指定的文本。
反转	反转显示指定的文本。

#### 3. 占用区域

显示 LCD 屏幕上使用的区域大小（行：1，列：1 到 24）。显示尺寸由 LCD 屏幕区域（或工作区）上所选的范围决定。如果选择了多行，则使用所选范围中最前面一行的区域。

#### 4. ON 文本

输入当指定的设备开启时要显示的文本。最多可输入 24 个单字节字符。一个空格也将计为一个字符。

#### 5. OFF 文本

输入当指定的设备关闭时要显示的文本。最多可输入 24 个单字节字符。一个空格也将计为一个字符。

#### 6. 特殊字符

在光标位置输入一个特殊字符。按下“特殊字符”时，在弹出的特殊字符列表窗口中选择要输入的字符。有关特殊字符，请参见第 12-3 页上的“8. 特殊字符”。

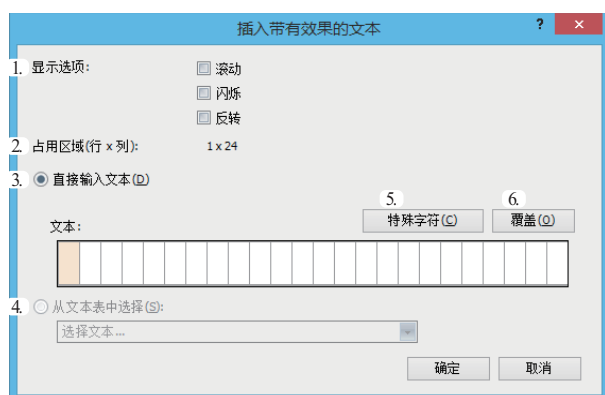
#### 7. 插入 / 覆盖

选择输入新字符时要插入还是覆盖字符。

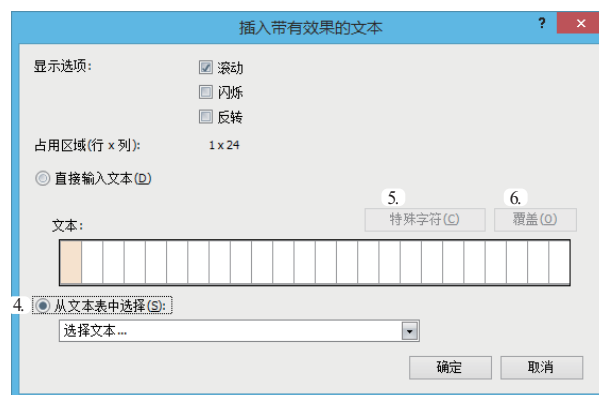
## 插入带效果的文本

HMI 模块 LCD 上可显示指定的文本。

[ 禁用滚动时 ]



[ 启用滚动时 ]



### 1. 显示选项

配置滚动、闪烁或反转指定文本的选项。有关滚动单位、滚动速度和闪烁速度，请参见第 12-9 页上的“MSG 指令常规设置”。

显示选项	说明
滚动	滚动指定的文本。
闪烁	闪烁指定的文本。
反转	反转显示指定的文本。

### 2. 占用区域

显示 LCD 画面（第 12-2 页上的“1. LCD 画面”）上使用的区域大小（行：1，列：1 到 24）。占用的区域由 LCD 画面区域（或工作区（第 12-2 页上的“2. 工作区”））上所选的范围决定。如果选择了多行，则使用所选范围中最前面一行的区域。

### 3. 直接输入文本

直接输入要显示的字符。一个空格也将计为一个字符。如果禁用滚动，则消息中最多可输入 24 个单字节字符。但是，输入的字符数不能超过占用的区域。如果启用滚动，则消息中最多可输入 48 个单字节字符。

### 4. 从文本表中选择

从文本管理器中选择要显示的文本。仅当启用了滚动时，才能进行此种选择。

### 5. 特殊字符

在光标位置输入一个特殊字符。按下“特殊字符”时，在弹出的特殊字符列表窗口中选择要输入的字符。有关特殊字符，请参见第 12-3 页上的“8. 特殊字符”。

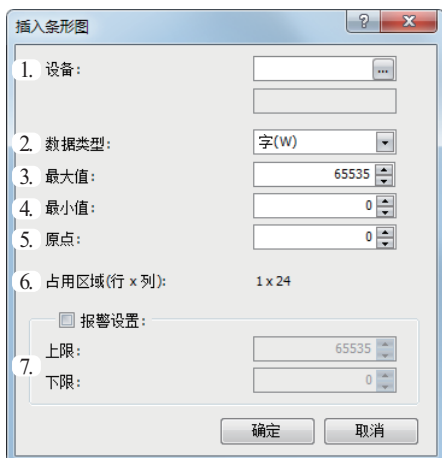
### 6. 插入 / 覆盖

选择输入新字符时要插入还是覆盖字符。

## 12: 显示指令

### 插入条形图

在 HMI 模块 LCD 上，可以条形图形式显示指定设备的值。



#### 1. 设备

输入要以条形图形式显示的设备。

##### 有效设备

W (字)	TC、TP、CC、CP、D
I (整数)	D
D (双字)	CC、CP、D
L (长整数)	D
F (浮点)	—

#### 2. 数据类型

选择指定设备的数据类型。

##### 数据类型

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

有关数据类型的详情，请参见第 3-8 页上的“高级指令的数据类型”。

#### 3. 最大值

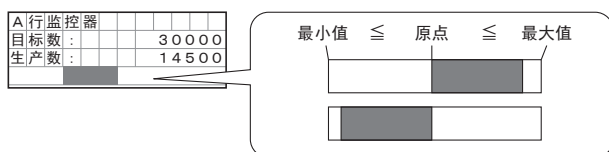
输入条形图的最大值。如果设备值大于最大值，则条形图将显示为最大值。有效范围因数据类型而异。有关数据类型的详情，请参见第 3-8 页上的“高级指令的数据类型”。

#### 4. 最小值

输入条形图的最小值。如果设备值小于最小值，则条形图将显示为最小值。有效范围因数据类型而异。有关数据类型的详情，请参见第 3-8 页上的“高级指令的数据类型”。

#### 5. 原点

输入作为条形图原点的值。如果设备值大于原点值，将在原点值的右侧显示条形图。如果设备值小于原点值，将在原点值的左侧显示条形图。有效范围因数据类型而异。有关数据类型的详情，请参见第 3-8 页上的“高级指令的数据类型”。原点值必须满足最小值、原点和最大值的条件。



#### 6. 占用区域

占用的区域由 LCD 画面（第 12-2 页上的“1. LCD 画面”）区域（或工作区（第 12-2 页上的“2. 工作区”））上所选择的范围决定。如果选择了多行，则使用所选范围中最前面一行的区域（行：1，列：1 到 24）。

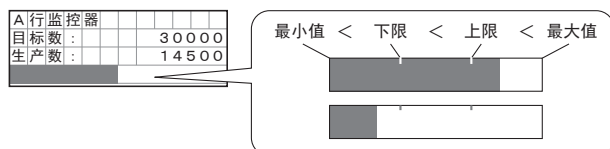


## 7. 报警设置

当指定设备的值超过上限或下限时，条形图将闪烁。  
有关闪烁速度，请参见第 12-9 页上的“MSG 指令常规设置”。

报警设置	说明
上限	当指定设备的值大于上限时，条形图将闪烁。
下限	当指定设备的值小于下限时，条形图将闪烁。

最大值、最小值、上限和下限值必须满足以下条件。

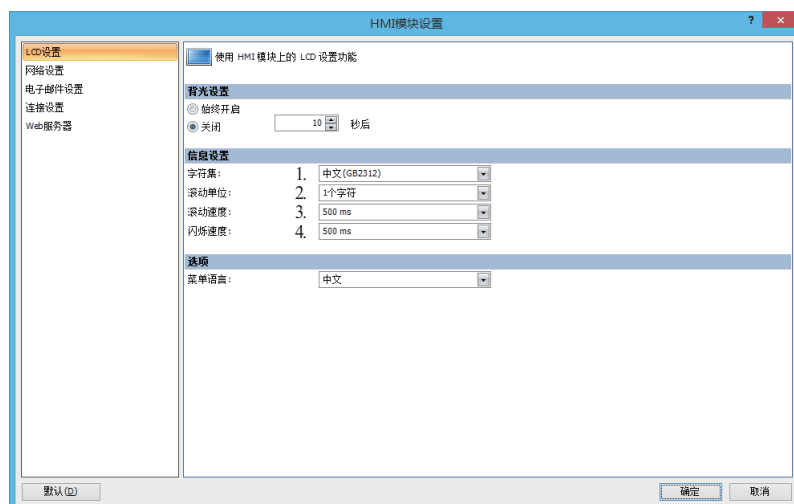


### ■ MSG 指令常规设置

可以配置要显示的消息的常规设置、字符集、滚动单位、滚动速度和闪烁速度。常规设置将在 WindLDR 的“扩展模块设置”中显示“HMI 模块设置”对话框，然后在“LCD”选项卡中进行设置。

**注释：**MSG 指令的常规设置适用于梯形图程序中的所有 MSG（消息）指令。有关 MSG 指令个别设置，请参见第 12-2 页上的“MSG 指令个别设置”。

### “HMI 模块设置”对话框的“LCD 设置”选项卡



#### 1. 字符集

从以下选项中可配置消息使用的字符集。

字符集	MSG 指令中可使用的语言
西欧 (ISO-8859-1 (Latin 1))	意大利语、英语、荷兰语、西班牙语、德语、法语
日文 (Shift-JIS)	日文
中文 (GB2312)	中文（简体）
西里尔语 (ANSI-1251)	俄语

#### 2. 滚动单位

可配置滚动文本的单位。

1 个字符	以 1 个字符为单位滚动文本。
1 个点	以 1 个点为单位滚动文本。

#### 3. 滚动速度

可配置滚动文本的速度。设置范围为 500ms 到 1,000ms。

#### 4. 闪烁速度

可配置闪烁文本的速度。设置范围为 500ms 到 1,000ms。

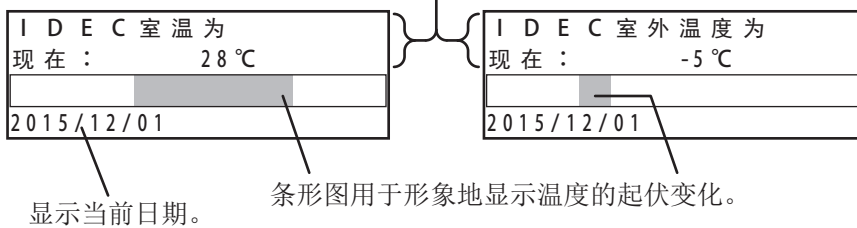
**注释：**无论使用哪种 HMI 模块系统菜单语言，均可配置 MSG 指令字符集。

## 12: 显示指令

### MSG 指令配置示例

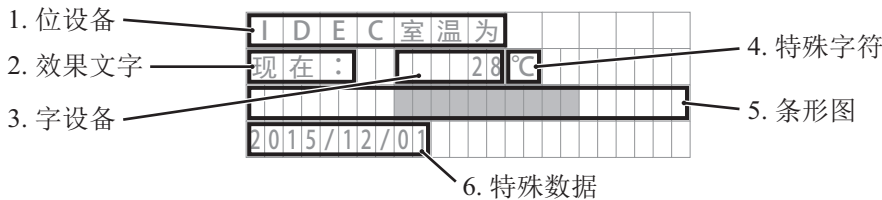
[ 当 M0000 开启时，显示室温；当 M0000 关闭时，显示室外温度 ]

室温 / 室外温度值存储在 D0002 中。创建梯形图程序，以便在 M0000 为打开时存储室内温度值、为关闭时存储室外温度值。



### ■ 设置

配置以下项目。



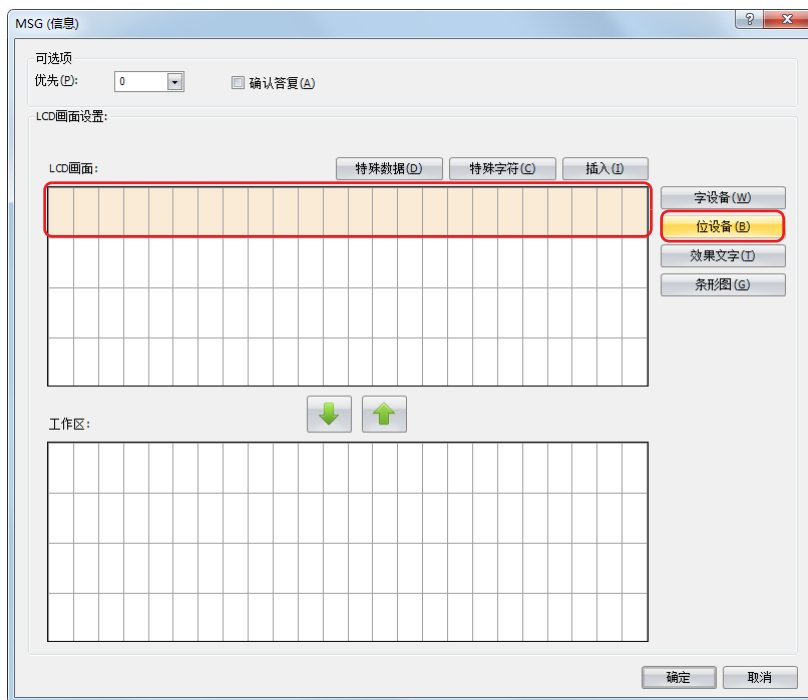
设置项目		设置详细信息
1. 位设备	设备	M0000
	显示选项	全部禁用 (滚动、闪烁、反转)
	开启文本	IDEC 室温为
	关闭文本	IDEC 室外温度为
2. 效果文字	文本	现在:
	显示选项	全部禁用 (滚动、闪烁、反转)
3. 字设备	设备	D0002
	数据类型	I (整数)
	转换类型	十进制
4. 特殊字符		°C
5. 条形图	设备	D0002
	数据类型	I (整数)
	最大值	50
	最小值	-20
	原点	0
6. 特殊数据		禁用
6. 特殊数据		当前日期

## ■ 操作程序

1. 在输入 MSG 指令的位置右击，然后在右击菜单中单击“高级指令”>“显示”>“MSG（信息）”。出现“MSG（信息）”对话框。

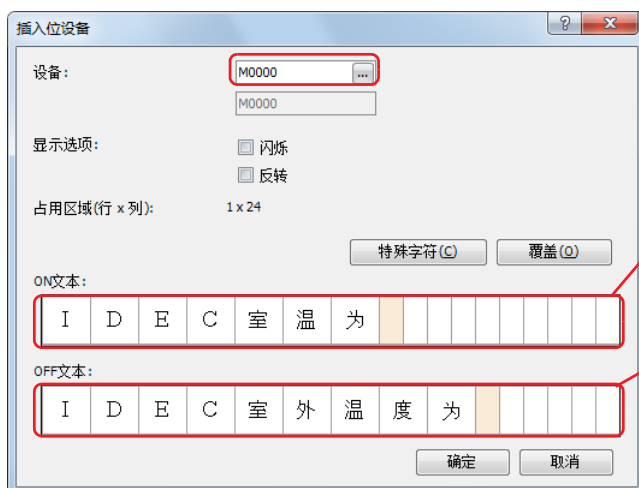
## 配置位设备

2. 选择要插入参数的区域，然后单击“位设备”。



将打开“插入位设备”对话框。

3. 配置参数，以便在 M0000 开启时显示室温，而当 M0000 关闭时显示室外温度。将“设备”设置为“M0000”。使用键盘输入“IDEC 室温为”作为开启文本，输入“IDEC 室外温度为”作为关闭文本。在“显示选项”下，禁用所有选项。



当 M0000 开启时显示

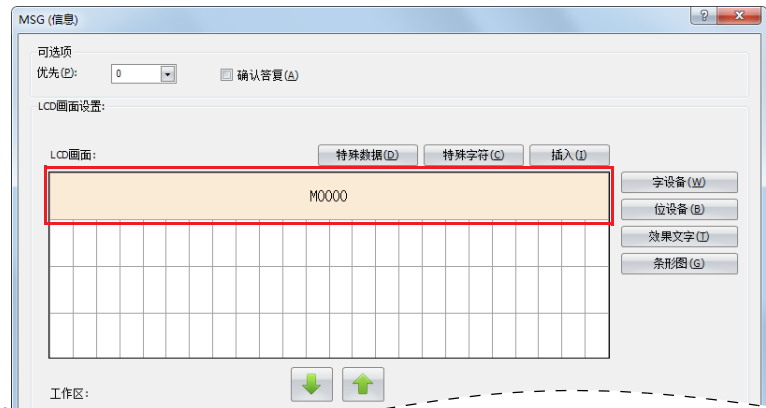


当 M0000 关闭时显示



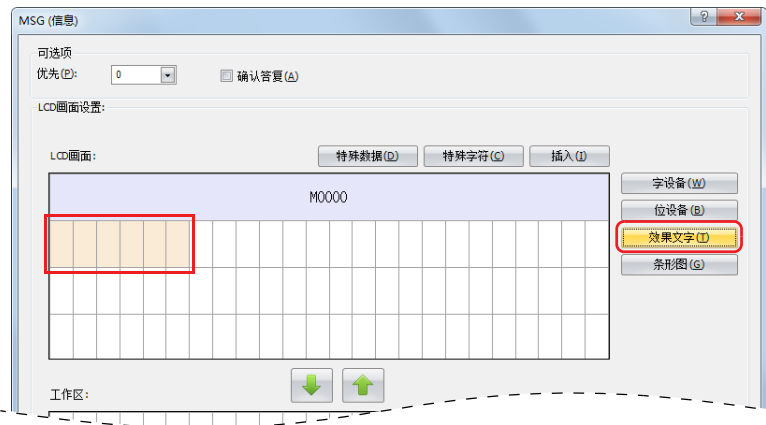
## 12: 显示指令

- 配置好设置后，单击“确定”。  
LCD 屏幕区域将显示配置的内容。



### 配置带效果的文本

- 从第二行开始选择六列区域，然后单击“效果文字”。

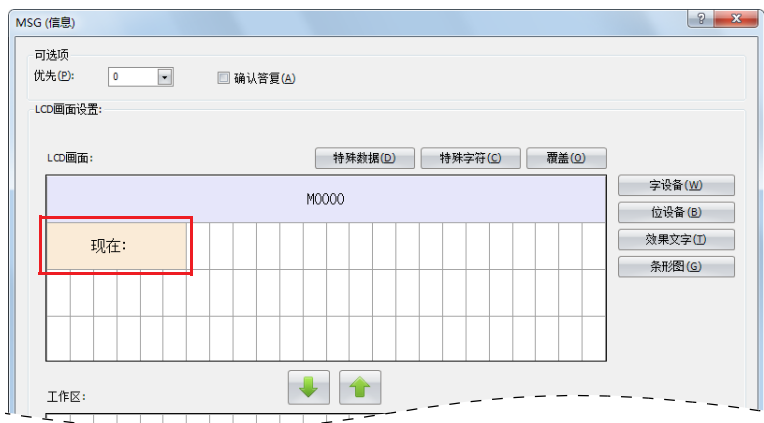


**注释：**也可以在 LCD 屏幕区域上直接输入文本。在上例中，将光标移到第二行的开始位置，使用键盘输入“现在：”。直接输入文本时，请继续第 12-13 页上的“配置字设备”的步骤 8。  
出现“插入带有效果的文本”对话框。

- 在文本中，使用键盘输入“现在：”。在“显示选项”下，禁用所有选项。

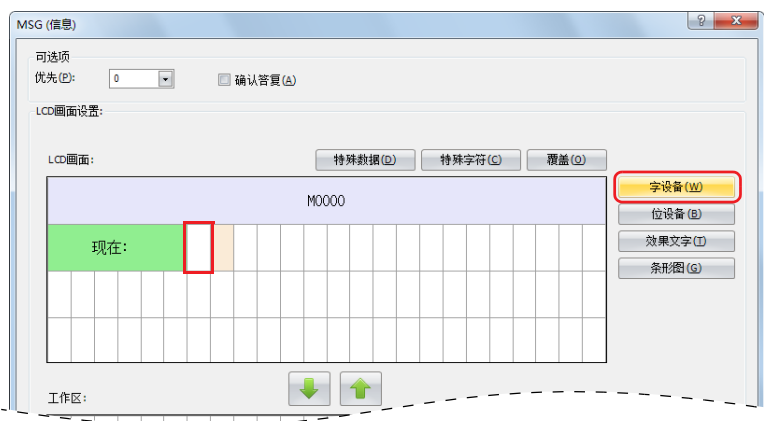


7. 配置好设置后，单击“确定”。  
LCD 屏幕区域将显示配置的内容。



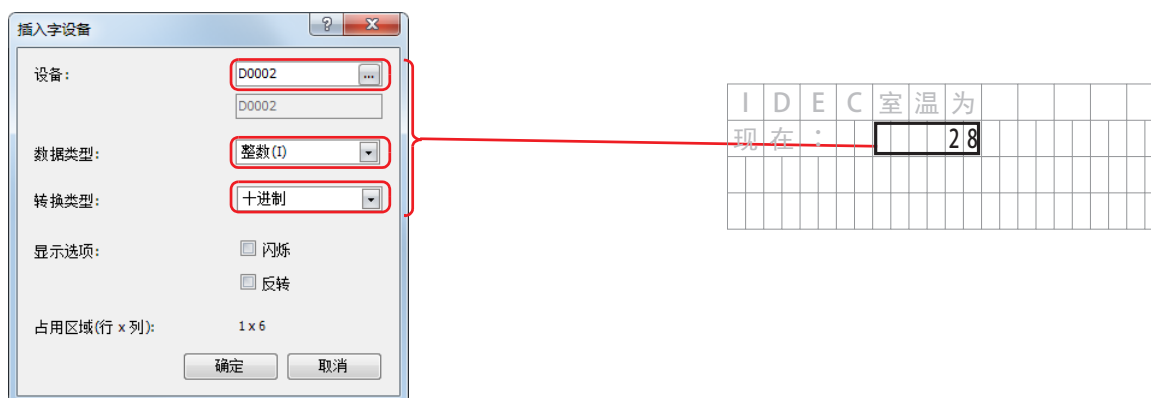
### 配置字设备

8. 选择第二行第九列的区域，然后单击“字设备”。



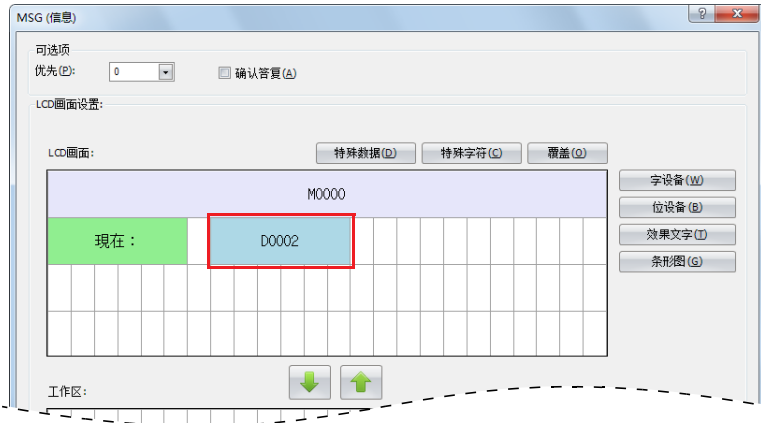
出现“插入字设备”对话框。

9. 将“设备”设置为“D0002”、“数据类型”设置为“整数 (I)”、“转换类型”设置为“十进制”。  
在“显示选项”下，禁用所有选项。



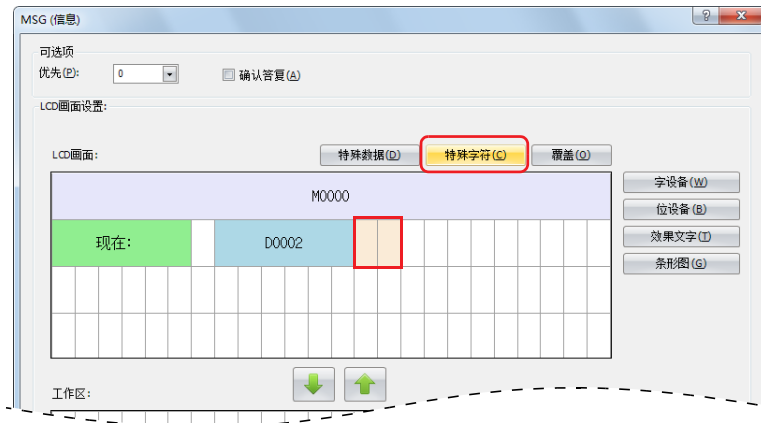
## 12: 显示指令

- 配置好设置后，单击“确定”。  
LCD 屏幕区域将显示配置的内容。



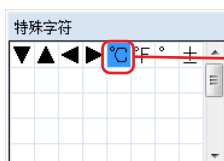
### 配置特殊字符

- 选择第二行第 15 列的区域，然后单击“特殊字符”。

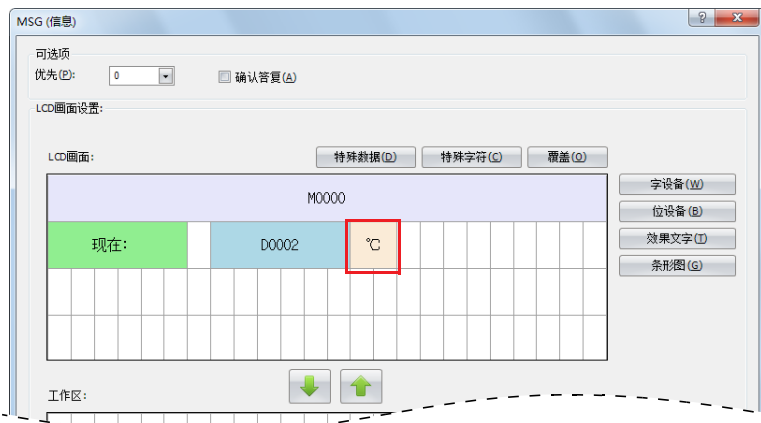


将弹出“特殊字符”窗口。

- 双击 °C。

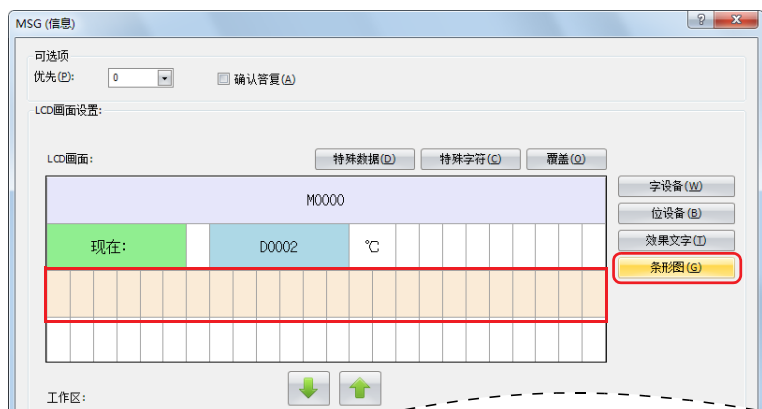


LCD 屏幕区域将显示配置的内容。



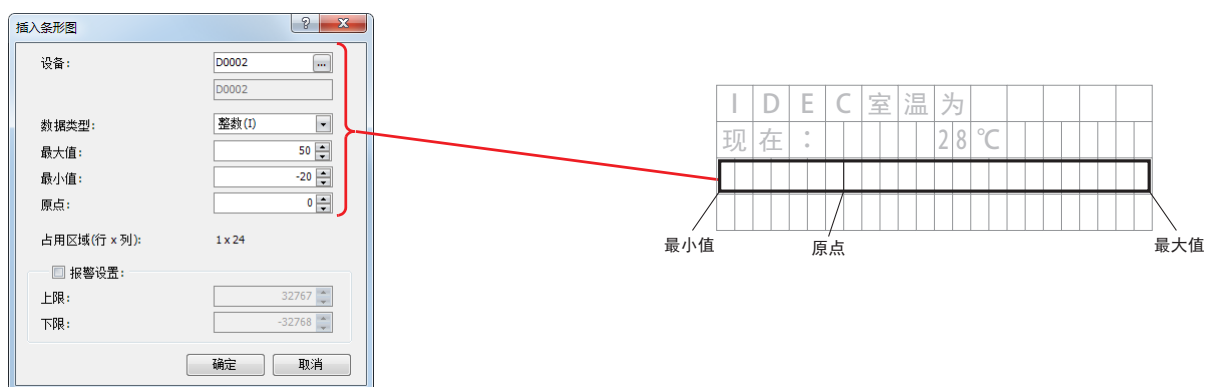
## 配置条形图

13. 选择整个第三行区域，然后单击“条形图”。

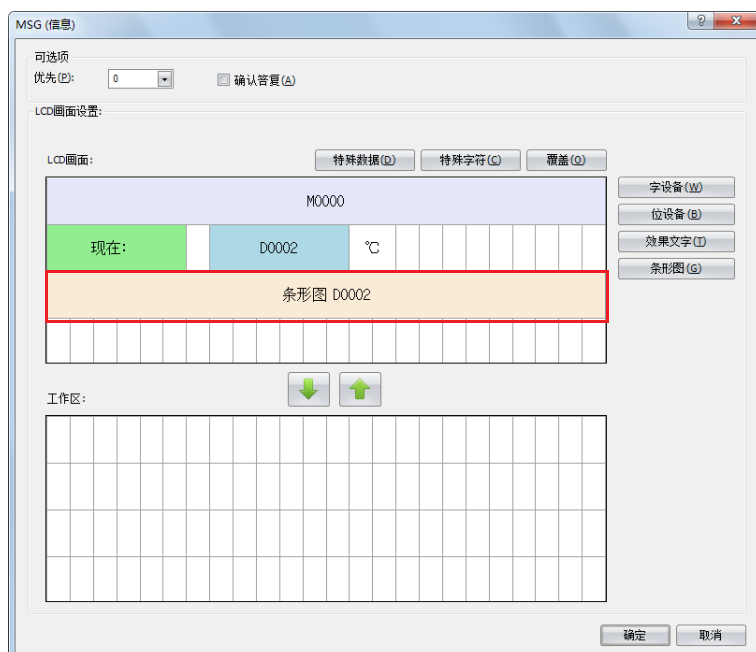


将弹出“插入条形图”窗口。

14. 将“设备”设置为“D0002”、“数据类型”设置为“整数 (I)”、“最大值”设置为“50”、“最小值”设置为“-20”、“原点”设置为“0”。禁用闪烁设置。



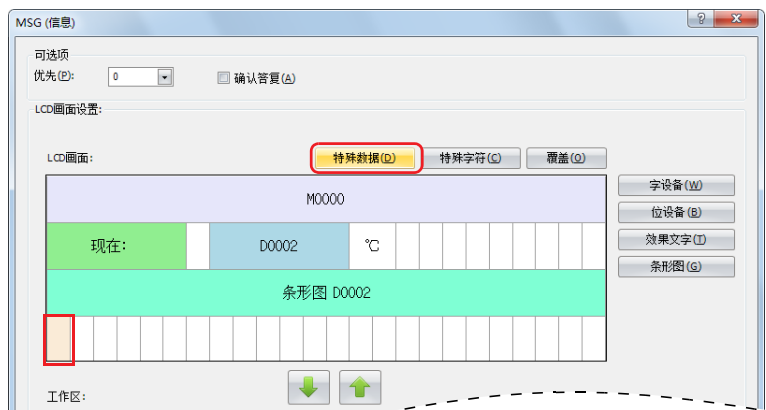
15. 配置好设置后，单击“确定”。  
LCD 屏幕区域将显示配置的内容。



## 12: 显示指令

### 配置特殊数据

16. 选择第四行的左边，然后单击“特殊数据”。

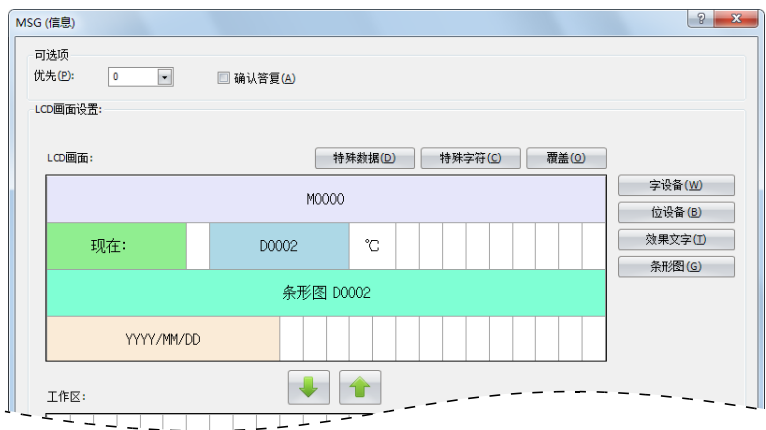


将弹出“特殊数据”窗口。

17. 双击“当前日期”。



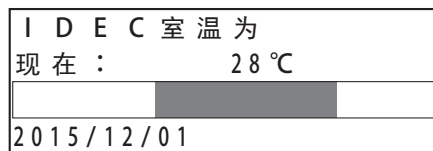
18. 配置好设置后，单击“确定”。  
LCD 屏幕区域将显示配置的内容。



配置即完成。

### LCD 显示器

开启 M0000 时，D0002 中存储的室温  
将以数字值 (°C) 和条形图形式显示。



关闭 M0000 时，D0002 中存储的室外温度  
将以数字值 (°C) 和条形图形式显示。





### 修改 HMI 模块上的设备值

使用 HMI 模块操作按钮可修改 HMI 模块 LCD 上显示的字设备的值。  
当用户程序停止时，无法修改值。

[ 修改字设备 CP0 的值 ]

A 行 监 控 器									
计 划 值 :	CP0	实 际 值 :	CC0						
差 值 :	D0								
条形图 CC0									

当用户程序运行并且 MSG 指令输入开启时，LCD 上将显示以下屏幕。

A 行 监 控 器									
计 划 值 :	60000	实 际 值 :	20000						
差 值 :	40000								

显示以上消息时，按住  $\square$  (确定) 按钮，可修改的设备上将显示光标。

A 行 监 控 器									
计 划 值 :	<u>6</u> 0000	实 际 值 :	20000						
差 值 :	40000								

使用  $\blacktriangle$  (向上)、 $\blacktriangledown$  (向下)、 $\blacktriangleleft$  (向左)、 $\blacktriangleright$  (向右) 按钮将光标移到想要编辑的设备上，按  $\square$  (确定) 按钮更改处于可编辑状态的设备。

A 行 监 控 器									
计 划 值 :	6 <u>0</u> 000	实 际 值 :	20000						
差 值 :	40000								

按下  $\blacktriangleright$  (向右) 按钮将光标移到第四位数，然后使用  $\blacktriangle$  (向上) 按钮将目标值改为“65000”。

A 行 监 控 器									
计 划 值 :	65000	实 际 值 :	20000						
差 值 :	40000								

请按下  $\square$  (确定) 按钮确认修改。

A 行 监 控 器									
计 划 值 :	65000	实 际 值 :	20000						
差 值 :	45000								

## 12: 显示指令

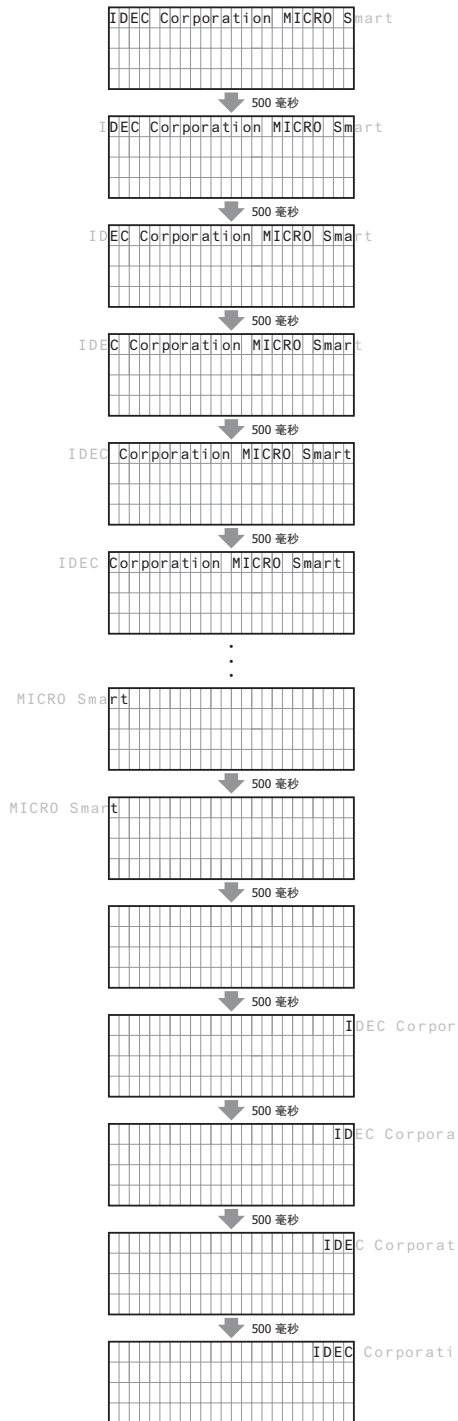
### 滚动文本示例

[字符集: 欧洲语言, 滚动速度: 500ms]

当带效果的文本设置为滚动时, HMI 模块 LCD 上显示的文本如下所示。

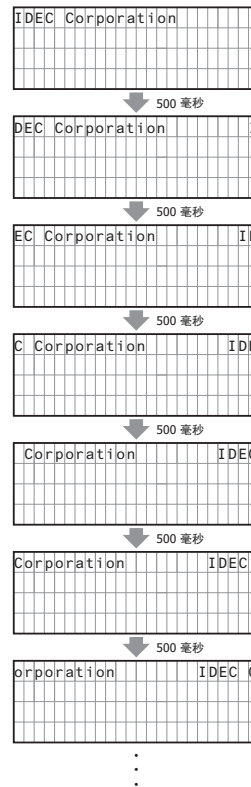
#### 滚动长度超过 LCD 上的指定区域时

文本: “IDEC Corporation MICRO Smart”



#### 滚动长度小于或等于 LCD 上的指定区域时

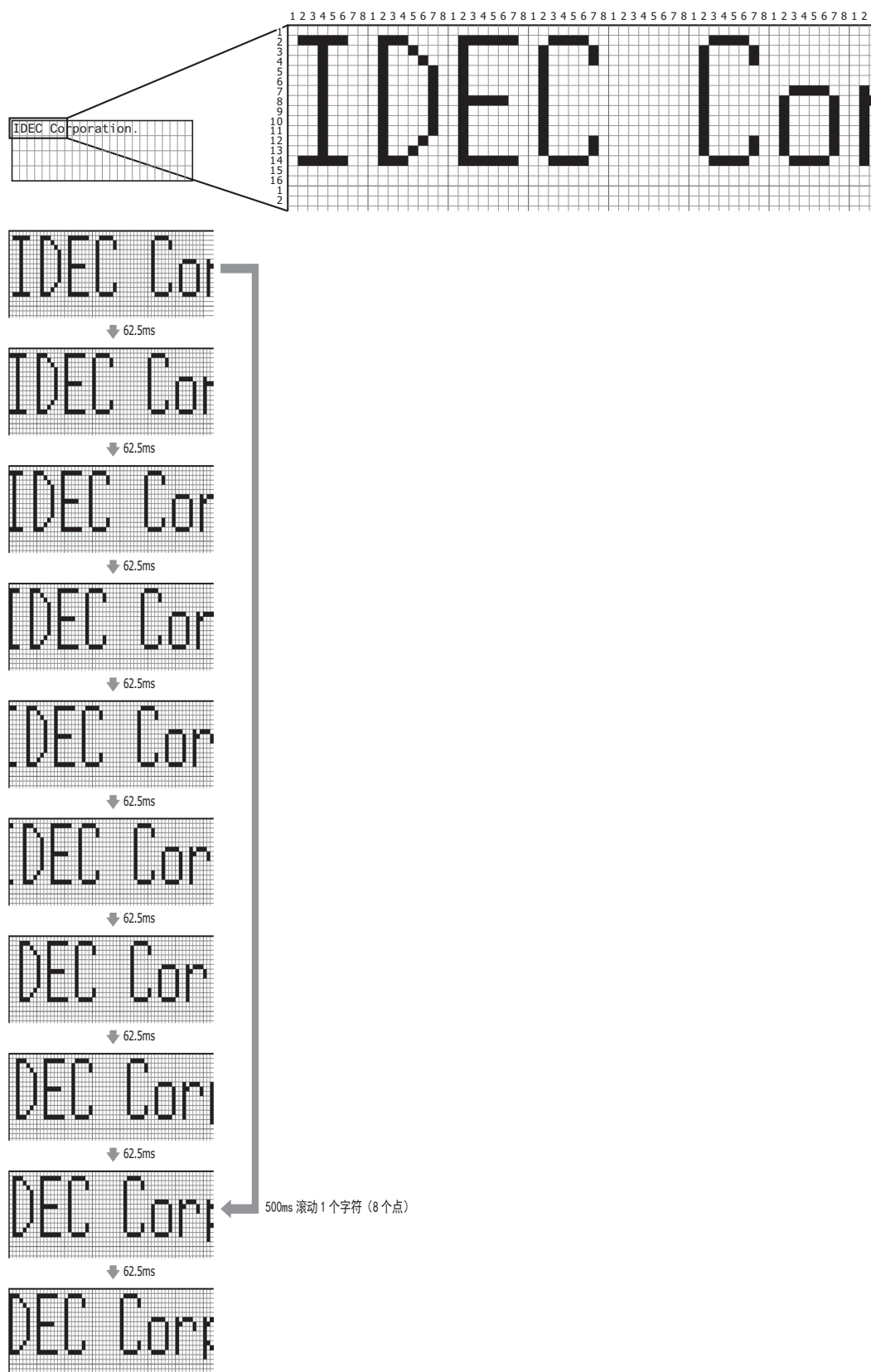
文本: “IDEC Corporation”



[字符集: 欧洲语言, 滚动速度: 500ms, 滚动单位: 1个点]

当带效果的文本设置为滚动时, HMI 模块的 LCD 上显示的文本如下所示。

文本: “IDEC Corporation.”



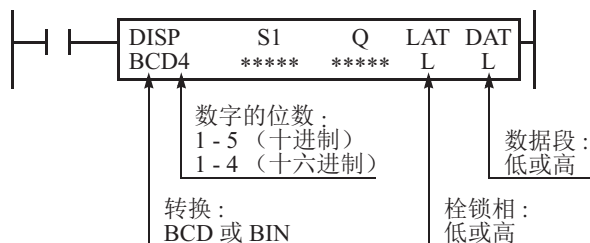
## 12: 显示指令

### DISP (显示)

在 7 段显示单元上显示指定数据。

FT2J FC6A

※ 仅晶体管输出类型



输入为打开时，S1 中指定的数据将显示在 7 段显示单元上。将指定数据显示于 7 段显示单元。

显示数据可以为 0 - 65535 (FFFFh)。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	要显示的数据	—	—	—	—	X <sup>*1</sup>	X <sup>*1</sup>	X	—	—	—
Q (输出)	显示数据的输出目标	—	X	X <sup>*2</sup>	—	—	—	—	—	—	—

关于有效设备编号范围，请参见第 2-12 页上的“FC6A 型”。

\*1 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)。

\*2 特殊内部继电器不能指定为 Q。

#### 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<sup>\*1</sup>和数据段<sup>\*1</sup>

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。

\*1 7 段显示单元的沉型输出规格、源型输出规格不同，LAT (栓锁相)、DAT (数据段) 的设置也有所不同。请根据 7 段显示单元的规格进行设置。

在 S1 中指定想要显示在 7 段显示单元上的数据 (显示数据)。在 Q 中指定显示数据的输出目标。显示数据和显示位数中指定的数据 (4 点 + 位数) 将以 Q 中指定的设备为起始，连续进行分配。

例如，显示位数设置为 4 位、显示数据的输出目标设置为 Q0 时，将分配 Q0 至 Q7 (Q0 至 Q3 中为发送至显示单元的数据信号、Q4 至 Q7 中为位选择信号)。

#### 注释:

- 用户程序内最多可输入 8 个 DISP 指令。
- 可显示范围为 0 至 65535 (FFFFh)。
- 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。

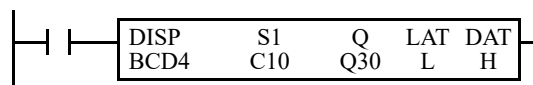
##### 显示处理时间

3 次扫描 × 数字的位数

若扫描时间小于 2ms，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)。

**示例：DISP**

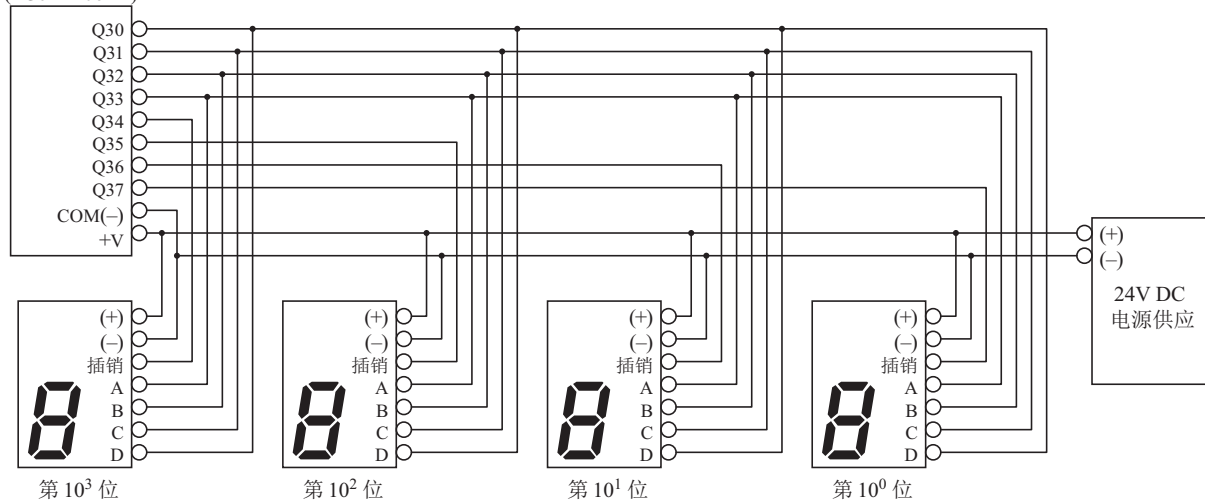
在晶体管沉型输出模块上连接 7 段显示单元（IDEC 制 DD3S-F31N），并以 4 位数字显示计数器 C10 的数据时。



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  
(FC6A-T08K1)

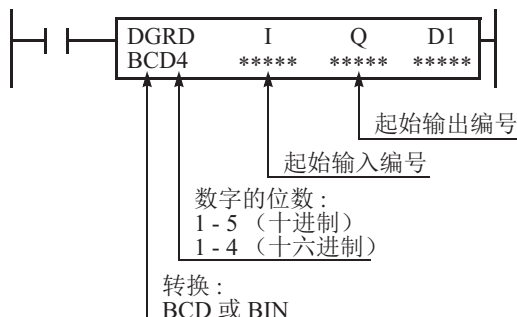


## DGRD（数字读取）

将所连接数字开关的设置值存储到指定的设备中。

FT2J FC6A

※ 仅晶体管输出类型



输入为打开时，所连接的数字开关的预置值存储在 D1 中指定的设备中。

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。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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，请参见第 2-12 页上的“FC6A 型”。

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 65,535 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

从 I 中指定的输入开始的 4 点分配为读取输入。

例如，在 I 中设置了 I0 时，将分配 I0 至 I3。

从 Q 中指定的输出开始，显示位数中指定的位数量将分配为位选择输出。

例如，显示位数为 3 位、在 Q 中设置了 Q0 时，将分配 Q0 至 Q2。

**注释：**

- 用户程序内最多可输入 16 个 DGRD 指令。
- 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  
数字开关数据读取时间  
2 次扫描 × (数字的位数 + 2)
- DGRD 指令所需扫描时间比过滤时间长 6ms。  
最低所需扫描时间  
(扫描周期) ≥ (过滤时间) + 6ms

请根据需要，使用固定扫描时间（D8022）功能调整扫描时间。过滤时间因读取输入中使用的输入而异。

**在读取输入中指定“CPU 模块”的输入时（I0 至 I27）**

请参照 WindLDR 的“功能设置”中设置的输入过滤时间。初始值为 3ms。

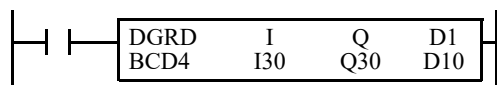
**在读取输入中指定“输入模块”的输入时（I30 之后）**

请将输入延迟时间替换为过滤时间后进行计算。

- 用该指令读取的数据为 0 - 65535（5 位数字）或 FFFFh。如果 10 进制代码、5 位指定为 65,536 以上，将导致用户程序执行错误，且不执行处理。  
有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

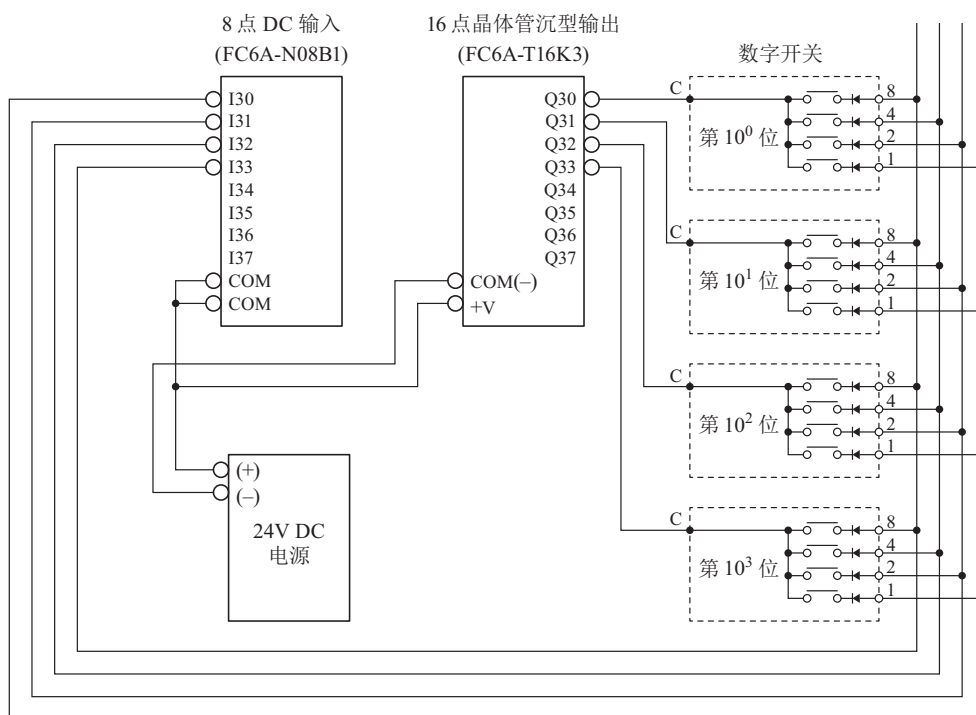
### 示例：DGRD

以下程序示例和连接示例表示在 16-I/O 型晶体管沉型输出模块上连接数字开关（IDEC 制 DFBN-031D-B），并将该预置值读取到 D0010 时的情况。



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







# 13: 程序分支指令

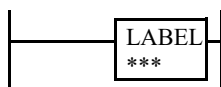
## 简介

本章将对梯形图程序执行的分支、梯形图程序的重复执行及梯形图程序的子程序化的程序分支指令进行介绍。通过 L JMP（标签跳转）指令或 DJNZ（递减跳转非零）指令，将梯形图程序的指令运行程序分支到已输入 LABEL 指令的位置。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。有关用户定义宏与子程序的差异，请参见附录 -2 页上的“用户定义宏与子程序的差异”。

## LABEL（标签）

设置标签编号。

FT2J FC6A



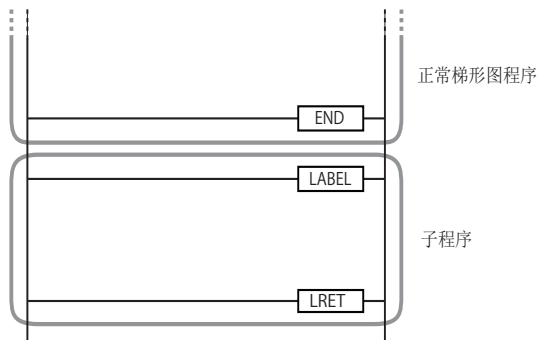
可指定的标签编号为 0 至 255。

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：**一个标签编号只能使用一次。

### 注释：

- 子程序可通过梯形图程序的 LCAL 指令调用。
- 使用 WindLDR 子程序功能创建子程序时，WindLDR 会自动生成 LABEL 指令和 LRET 指令。无需放置 LABEL 和 LRET 指令即可轻松创建子程序。
- 在不使用 WindLDR 子程序函数的情况下创建子程序时，将 END 指令放在主程序的末尾，并在 LABEL 和 LRET 指令之间创建梯形图程序，如下所示。



## 有效设备

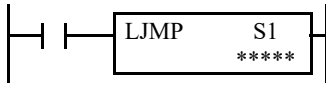
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
标签编号	LJMP 和 LCAL 标记	—	—	—	—	—	—	—	—	0-255	—

## 13: 程序分支指令

### LJMP (标签跳转)

跳转到指定的 LABEL 指令。

FT2J FC6A



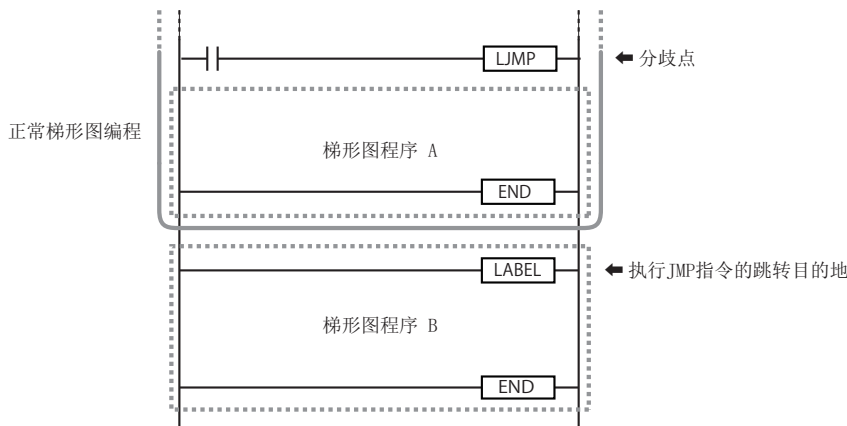
当输入打开时，将跳转到带有 S1 指定的标签 0 ~ 255 的地址处。

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。

#### 注释:

- LJMP 指令用于跳转到放置 LABEL 指令的行。在执行 END 指令时，运行梯形图程序 B 以结束程序。在这种情况下，不执行梯形图程序 A。在主程序中创建梯形图程序 B。



- 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，将存储错误代码 14 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	标签编号	—	—	—	—	—	—	X	—	0-255	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 执行 JMP 和 LJMP 指令时每条指令的状态

当 JMP 指令输入打开时，在跳转电路中编程的每条指令的状态如下；当 LJMP 指令输入打开时，在 LJMP 和 END 指令之间编程的每条指令的状态如下。

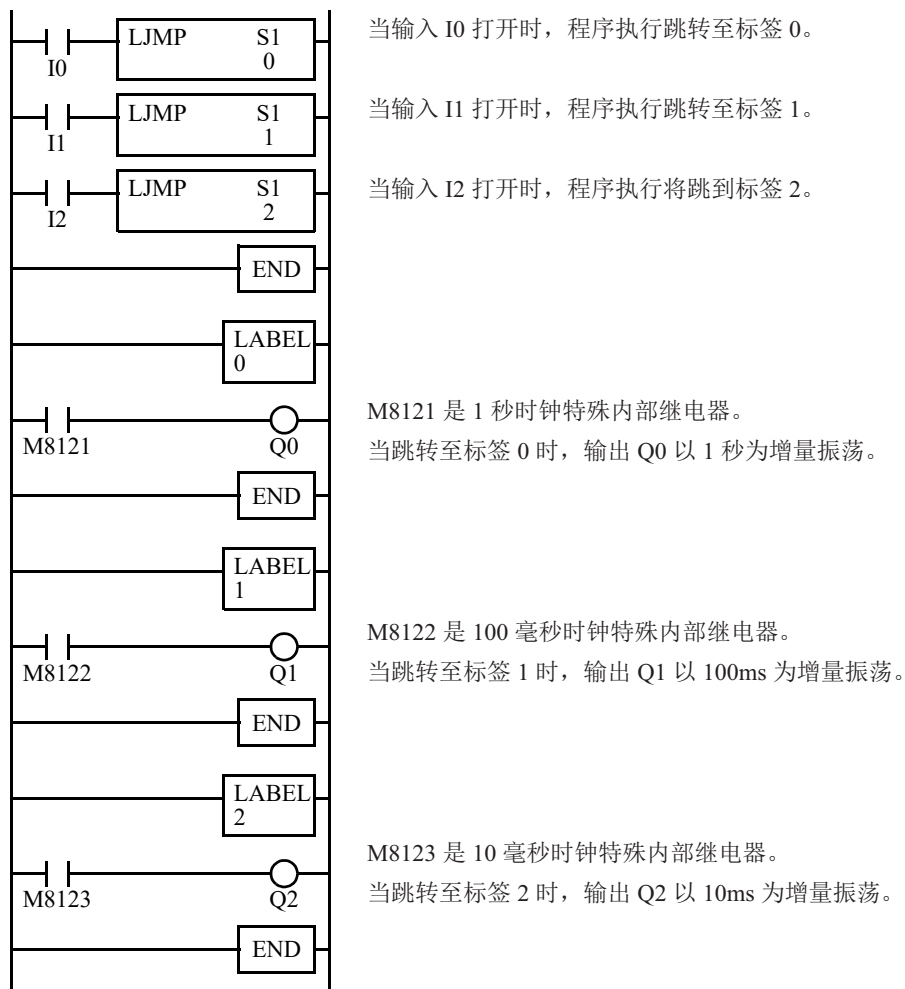
指令	当 JMP 指令*1 输入打开时	当 LJMP 指令输入打开时
OUT 和 OUTN	全部保持当前状态。	
SET 和 RST	全部保持当前状态。	
TML、TIM、TMH 和 TMS	保持当前值。	计数继续进行。 但是，超时状态不会开启*2。
TMLO、TIMO、TMHO 和 TMSO	保持超时状态。	
CNT、CDP 和 CUD	关闭脉冲输入。	保持当前值。保持计数器输出状态。
CNTD、CDPD 和 CUDD	保持当前值。	
SFR 和 SFRN	保持移位寄存器位状态。 关闭脉冲输入。	

\*1 有关 JMP 指令的详情，请参见第 4-35 页上的“JMP (跳转)”。

\*2 超时状态会在 LJMP 指令输入关闭后反映出来。

### 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。



## 13: 程序分支指令

### LCAL（标签调用）

保存当前程序执行并调用子程序。

FT2J FC6A



当输入打开时，将调用带有 S1 指定的标签 0 ~ 255 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：**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，将存储错误代码 14 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

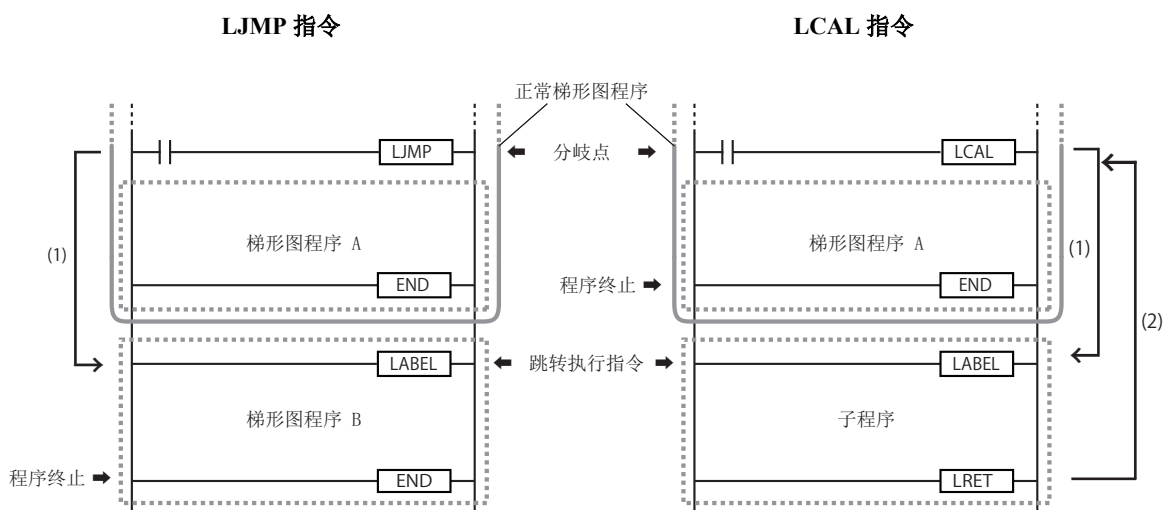
#### 有效设备

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

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### LJMP 和 LCAL 指令的区别

LJMP 和 LCAL 指令在执行梯形图程序后的行为有所不同，每条指令都被执行并跳转到梯形图程序。



程序执行跳转到 LJMP 指令指定标签号的 LABEL 指令所在行 (1)。

跳转后，不能执行 LRET 指令。  
END 指令终止程序执行。

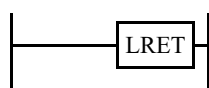
程序执行跳转到带有 LCAL 指令指定的标签号的 LABEL 指令所在行 (1)。

子程序被执行后，LRET 指令将程序执行跳转到调用子程序的 LCAL 指令的下一行 (2)。  
(程序执行也可以通过在 LRET 指令的位置放置一条 END 指令来终止)。

## LRET (标签返回)

将程序执行返回到执行 LCAL 指令的行。

FT2J FC6A



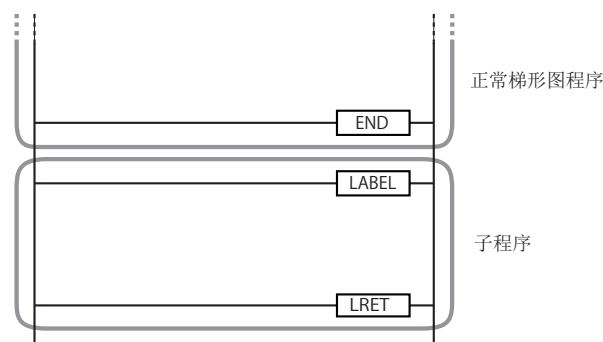
返回到 LCAL (标签调用) 指令的地址。

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 FC6A 型 MICROSmart 上的特殊内部继电器 M8004 和 ERR LED。

### 注释:

- 使用 WindLDR 子程序功能创建子程序时, WindLDR 会自动生成 LABEL 指令和 LRET 指令。无需放置 LABEL 和 LRET 指令即可轻松创建子程序。
- 在不使用 WindLDR 子程序函数的情况下创建子程序时, 将 END 指令放在主程序的末尾, 并在 LABEL 和 LRET 指令之间创建梯形图程序, 如下所示。



- 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, 将存储错误代码 14 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情, 请参阅第 3-10 页上的 "用户程序执行错误"。

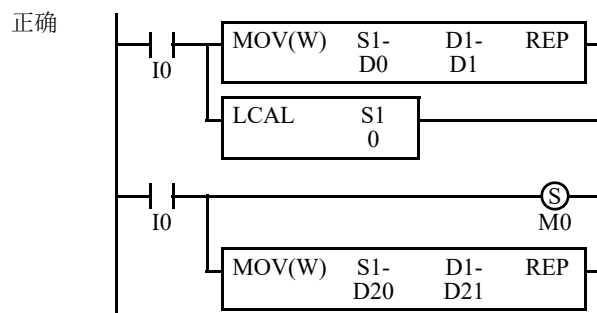
### 有效设备

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

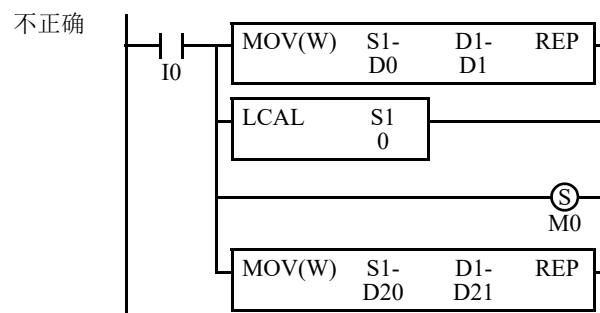
### 调用子程序的正确结构

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。



分隔每个 LCAL 指令中的梯形行。

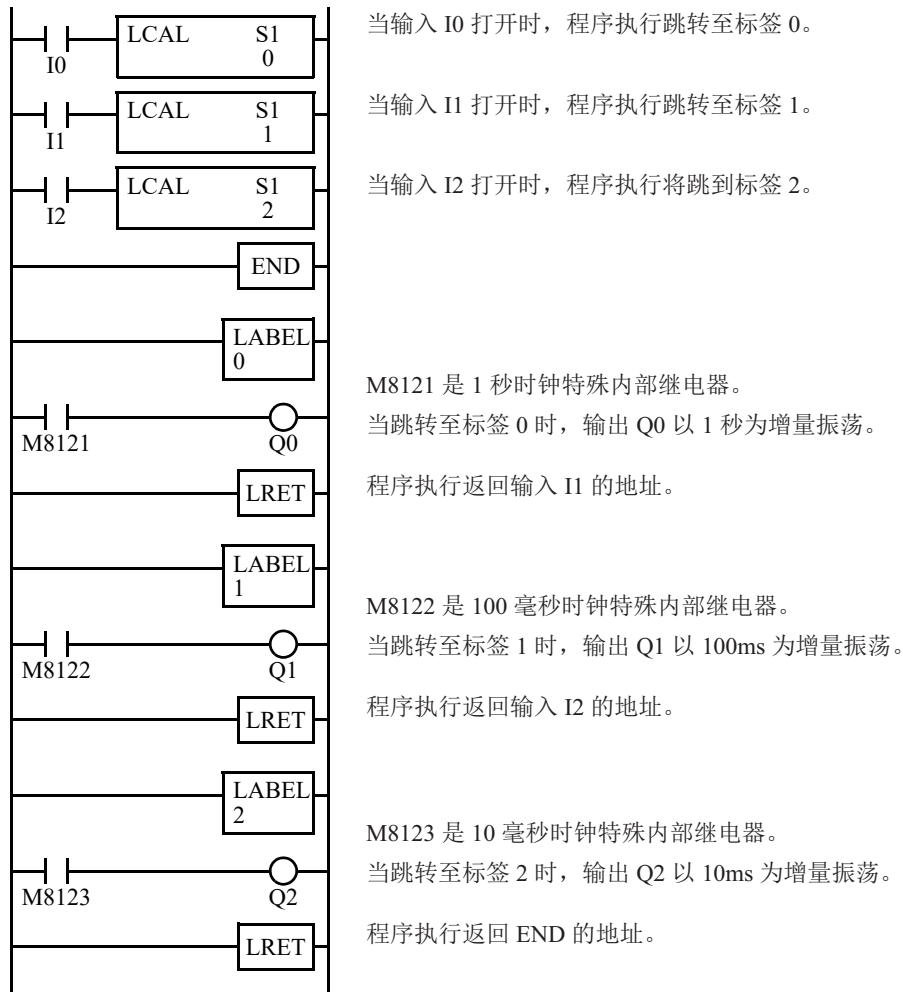


返回后子程序可能更改 I0 状态。

### 13: 程序分支指令

#### 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 LRET 指令后的指令。



## DJNZ（递减跳转非零）

条件判断后，跳转到指定的 LABEL 指令。

FT2J FC6A

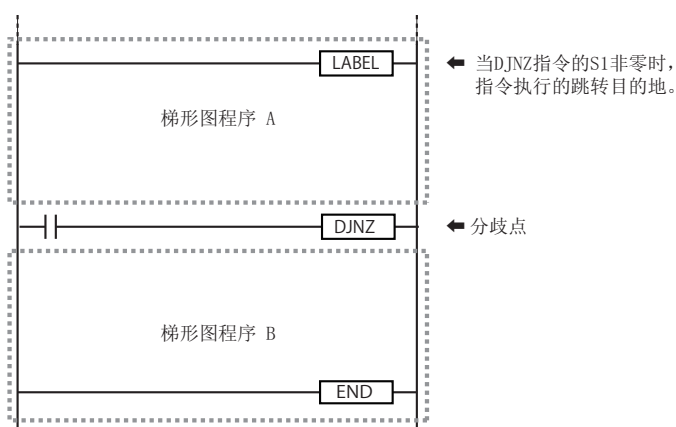


当输入打开时，存储在由 S1 指定的数据寄存器中的值被递减 1 并进行核对。

如果结果值不是 0，程序执行跳转到由 S2 指定的标签 0 ~ 255 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。

### 注释：

- 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，将导致用户程序执行错误，将存储错误代码 14 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 如果结果值不是 0，程序执行将跳转到放置 LABEL 指令的行，并执行梯形图程序 A。如果递减结果为 0，程序执行将执行梯形图程序 B，并以 END 指令结束。



### 有效设备

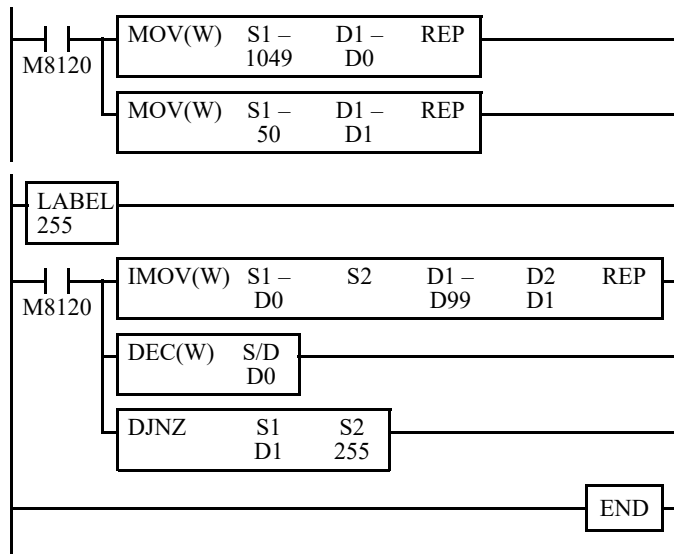
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	递减值	—	—	—	—	—	—	X	—	—	—
S2 (源 2)	要跳转到的标签编号	—	—	—	—	—	—	X	—	0 ~ 255	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 13: 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 1,000 ~ 1,049 存储到数据寄存器 D100 ~ D149 的程序。



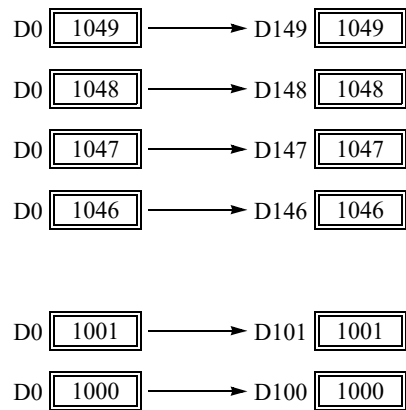
M8120 为初始化脉冲特殊内部继电器。  
启动时 MOV 指令存储原始数据。  
1,049 → D0 用于存储第1次循环的值。  
50 → D1 用于确定跳转循环。

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。

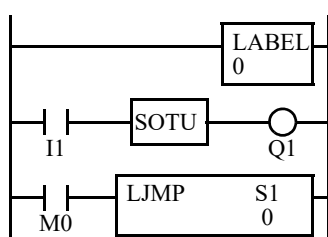
第1次循环:	D1	<input type="text" value="50"/>	目标: D99 + 50 = D149
第2次循环:	D1	<input type="text" value="49"/>	目标: D99 + 49 = D148
第3次循环:	D1	<input type="text" value="48"/>	目标: D99 + 48 = D147
第4次循环:	D1	<input type="text" value="47"/>	目标: D99 + 47 = D146
		⋮	
第49次循环:	D1	<input type="text" value="2"/>	目标: D99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





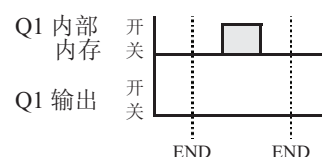
### 程序分支指令说明

- LJMPL 和 LCALL 指令将在每次扫描时输入打开时执行，因此必须使用 SOTU 或 SOTD 指令的脉冲输入；有关 SOTU 和 SOTD 指令，请参见第 4-31 页上的“SOTU（上升沿微分）”。
- 当梯形图程序在跳转目标处使用以下指令时，请注意以下几点。  
 定时器、计数器、双字计数器、移位寄存器和 SOTU 指令  
 跳转后，指令输入应在一个或多个扫描周期内保持关闭。  
 要使定时器指令正常工作，必须从计数开始到计时结束的每个扫描周期都执行该指令。  
 断开延时定时器和 SOTD 指令  
 跳转后，指令输入应保持接通一个或多个扫描周期。  
 要正确执行定时器指令，必须从计数开始到计时结束的每个扫描周期都执行该指令。
- 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。



尽管在程序分支中使用 SOTU 指令时通常会产一个扫描脉冲，SOTU 脉冲仅持续至再次执行相同的 SOTU 指令时。

在左侧示例中，只要内部继电器 M0 保持打开，程序分支将持续循环。但是 SOTU 仅在第一次循环时产生脉冲输出。



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。

- LCALL 指令中的子程序调用层次（在被调用的子程序中调用另一个子程序的嵌套深度）最多为 4。调用 5 级或 5 级以上子程序时，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 使用 LJMPL 或 DJNZ 指令时，请为指定标签号的跳转目标创建梯形图程序。
- 使用 LCALL 指令时，请创建一个具有指定编号的子程序或一个具有指定标签编号的跳转目标的梯形图程序。
- 分支指令会产生无限循环。当扫描处理时间超过允许时间时，监视定时器将发生错误并复位系统。创建梯形图程序时请格外注意。有关监视定时器的详情，请参见以下手册。  
 FT2J 型：《WindO/I-NV4 用户手册》第 28 章中的“2.13 监视定时器设置”  
 FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“监视定时器设置”



# 14: 刷新指令

## 简介

本章中所述的刷新指令，用于更新输入触点和外部输出的状态，以及将高速计数器的当前值更新为梯形扫描期间的最新值。

## IOREF (I/O 刷新)

在梯形图程序过程的任何位置执行输入和输出刷新。

FT2J FC6A



输入打开时，不管扫描时间如何，都会立即刷新源设备 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 已指定给控制器上的每个可用输入。

当 Q（输出）用作 S1 时，会立即将 RAM 中的输出数据写入控制器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。

### 注释:

- 如果为 S1 指定了输入设备，则在使用该输入设备的值之前的行上执行 IOREF 指令可获得最高效果。
- 如果为 S1 指定了输出设备，则在改变输出设备后立即在行上执行 IOREF 指令，效果最高。

### 输入设备编号和已指定内部继电器

要存储的内部继电器取决于输入设备。例如，如果已将 I10 指定给 S1，则最新的 I10 值会被存入 M310。

输入设备	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	I20	M320
I1	M301	I11	M311	I21	M321
I2	M302	I12	M312	I22	M322
I3	M303	I13	M313	I23	M323
I4	M304	I14	M314	I24	M324
I5	M305	I15	M315	I25	M325
I6	M306	I16	M316	I26	M326
I7	M307	I17	M317	I27	M327

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。

未使用 IOREF 指令的内部继电器可作为普通内部继电器使用。

### 注释:

- 可为 S1 设置的输入或输出设备取决于控制器类型。
- FT2J 型的 I12 ~ I15 无法指定为 S1。
- Plus CPU 模块的 I14 ~ I17、Q14 ~ Q17 无法指定为 S1。
- 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。
- IOREF 可与中断输入或定时器中断一起用于刷新数据。

## 14: 刷新指令

### 有效设备

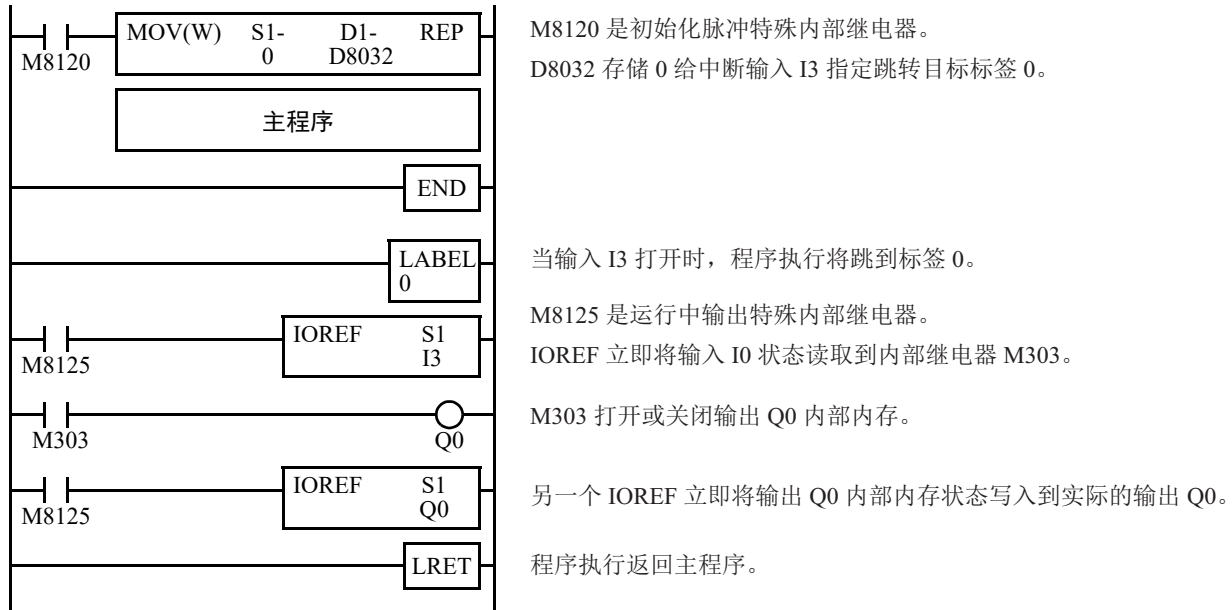
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	供刷新的 I/O	X	X	—	—	—	—	—	—	—	—

IOREF 指令的 S1 中，仅可指定控制器上存在的输入 / 输出。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 示例 :IOREF

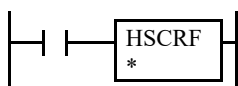
以下示例演示了使用 IOREF 指令传送输入 I0 状态至输出 Q0 的程序。输入 I3 已指定为中断输入。



## HSCRFB (高速计数器刷新)

将高速计数器的当前值更新为最新值。

FT2J FC6A



输入为打开时，特殊数据寄存器中分配的高速计数器当前值，将刷新为最新值。（高级指令标记的下方将显示已选高速计数器的组编号。）

高速计数器的当前值通常在每次扫描时进行更新。HSCRFB 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，请参见以下手册。

FT2J 型：《WindO/I-NV4 用户手册》第 28 章中的“2.5 高速计数器”

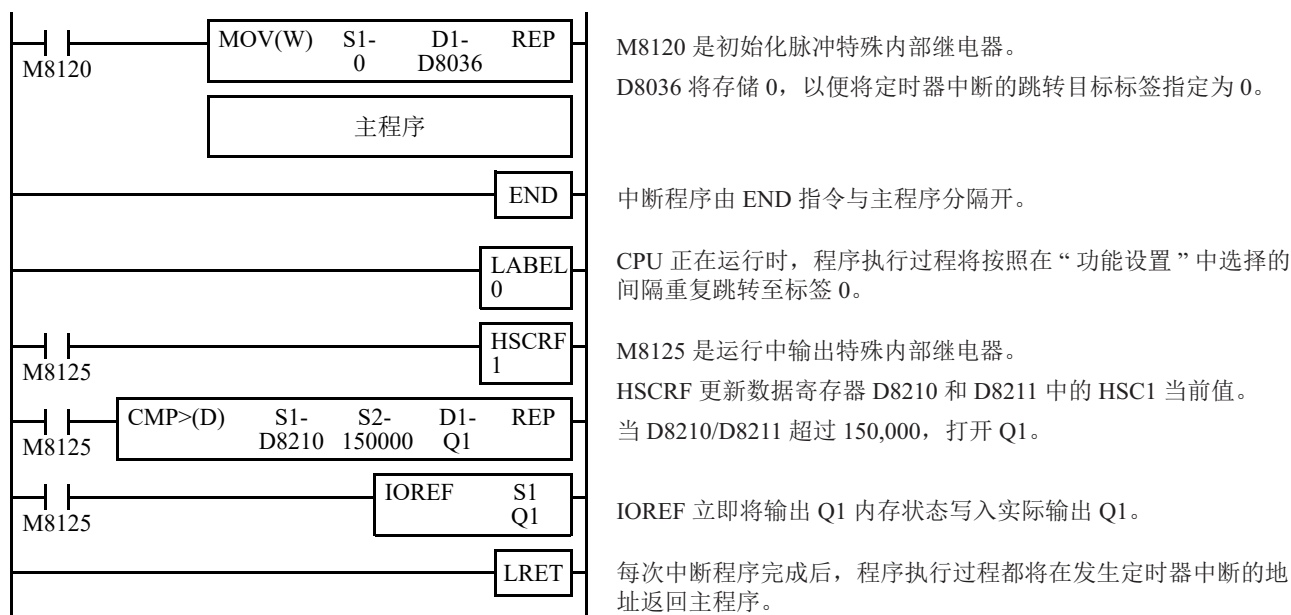
FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“高速计数器”

### 注释：

- 如果指定了不使用高速计数器的组，将导致用户程序执行错误，错误代码 19 将存储在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 在使用高速计数器当前值之前的行上执行 HSCRFB 指令时，效果最佳。

### 示例 :HSCRFB

以下示例演示了使用 HSCRFB 更新高速计数器 HSC1 当前值的程序。有关定时器中断，请参见《FC6A 型 MICROSmart 用户手册》第 5 章中的“定时器中断”。



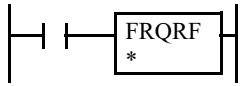
存储高速计数器当前值的特殊数据寄存器取决于控制器类型。请参见第 2-1 页上的“设备地址”。

## 14: 刷新指令

### FRQRF (频率测量刷新)

将频率测量值更新为最新值。

FT2J FC6A



输入为 ON 时，特殊数据寄存器中分配的频率测量值当前值，将刷新为最新值。（高级指令标记的下方将显示已选高速计数器的组编号。）

在测量结果存储到特殊数据寄存器之前，最多需要 250ms + 扫描时间，但通过使用 FRQRF 指令，可使用用户程序内任意位置的最新频率测量值的当前值进行控制处理。

有关频率测量功能，请参见以下手册。

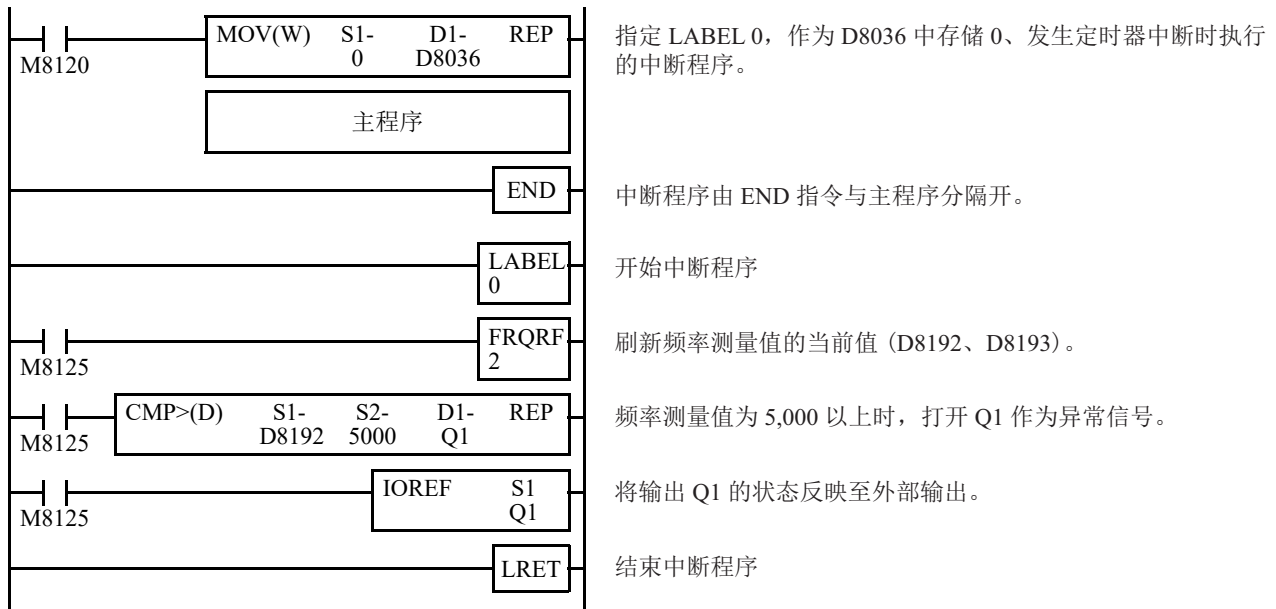
FT2J 型：《WindO/I-NV4 用户手册》第 28 章中的“2.8 频率测量”

FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“频率测量”

#### 示例:FRQRF

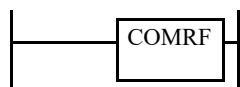
在中断程序内使用 FRQRF 指令，定期刷新频率测量值的当前值。

频率测量值为 5,000 以上时，打开异常信号（Q1）。



存储频率测量当前值的特殊数据寄存器取决于控制器类型。请参见第 2-1 页上的“设备地址”。

## COMRF (通信刷新)



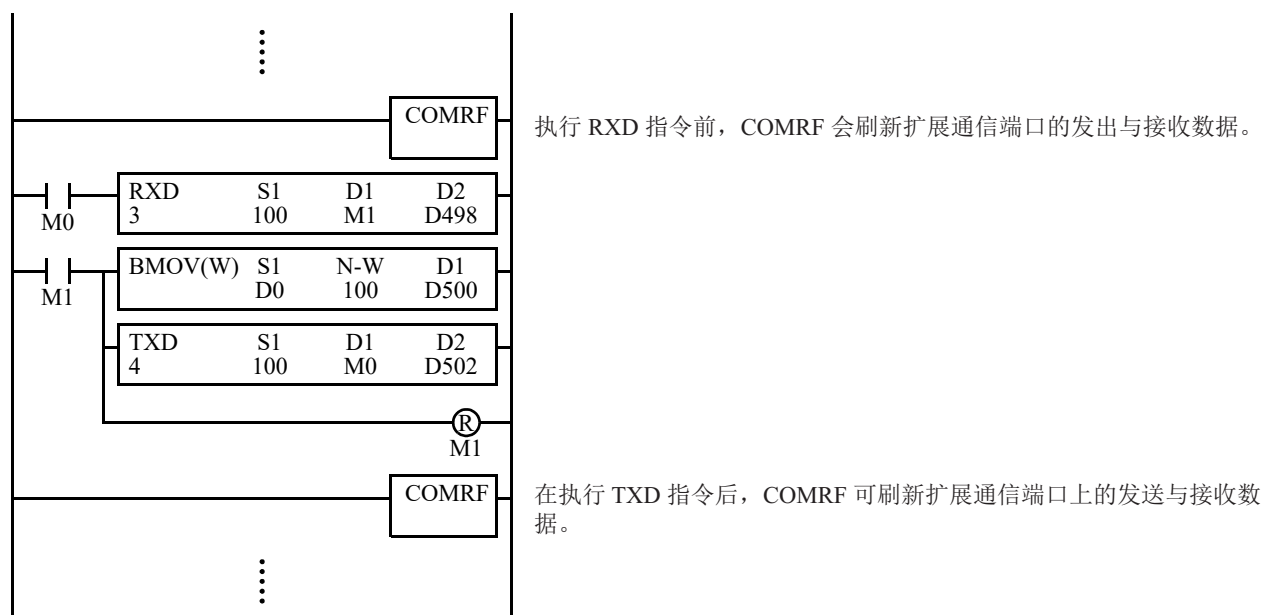
在用户程序处理中，针对端口 2 及 3 读取接收数据或写入发送数据（通信端口的刷新）。如需实现必须以短于 FC6A 型扫描时间的时间间隔响应通信盒的通信时，请编写 COMRF 指令。无论有无 COMRF 指令，都可通过结束处理，针对扩展的通信端口读取接收数据或写入发送数据。

## 注释：

- 中断程序无法使用本指令。如果在中断程序中使用本指令，将导致用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 对通信盒的通信响应性能要求实时性时启用，但 FC6A 型的扫描时间会在每个 COMRF 指令中最多延迟 2ms，因此请根据应用进行使用。

## 示例：COMRF

以下示例演示了在其他指令中执行 COMRF 指令的程序。







# 15: 中断控制指令

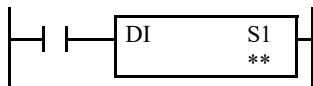
## 简介

本章介绍禁用 / 启用用户中断（中断输入、定时器中断）动作的中断控制指令。

## DI（禁用中断）

禁用中断输入和定时器中断。

FT2J FC6A

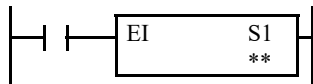


输入打开时，将禁用源设备 S1 指定的中断输入和定时器中断。

## EI（启用中断）

启用中断输入和定时器中断。

FT2J FC6A



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。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	中断输入和定时器中断	—	—	—	—	—	—	—	—	X <sup>*1</sup>	—

\*1 可设置中断输入或定时器中断。详情请参见第 2-1 页上的“设备地址”。

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 指令选择不同的中断，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的详情，请参见以下手册。

FT2J 型：《WindO/I-NV4 用户手册》第 28 章中的“2.7 中断输入”和“2.12 定时器中断”

FC6A 型：《FC6A 型 MICROSmart 用户手册》第 5 章中的“中断输入”和“定时器中断”

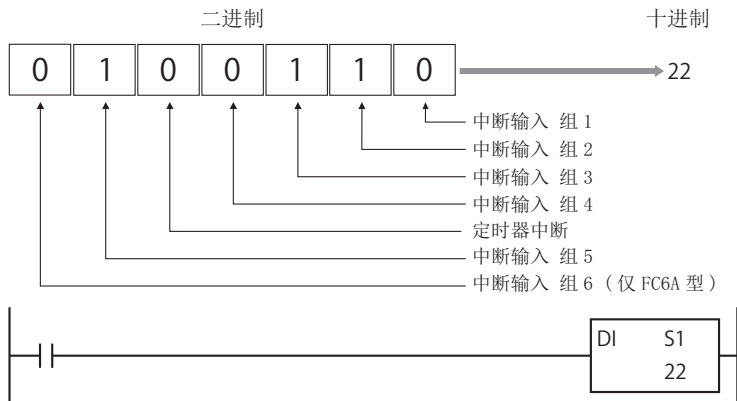
## 注释：

- 在启动 CPU 时，通常会启用功能设置中所选的中断输入 I0 ~ I7 和定时器中断。
- 在执行 EI 指令之前，DI 指令禁用的中断输入和定时器中断不会被启用。编写与 DI 指令相对应的 EI 指令。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 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，将存储错误代码 26 在用户程序执行错误代码 (D8006) 中。
- 中断程序中不能使用 DI 和 EI 指令。如果使用该指令，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

## 15: 中断控制指令

### 用户中断

要禁用或启用的用户中断由 DI 或 EI 指令的 S1 指定。与 DI 或 EI 指令对话框中指定的用户中断相对应的位序列被转换为十进制数并存储在 S1 中。下面的示例为 DI 指令选择了中断组 2、3 和定时器中断，源设备 S1 将显示为 22。



### 用户中断状态

提供特殊的内部继电器来指示中断输入和定时器中断是启用还是禁用。(启用：ON，禁用：OFF)。

#### FT2J 型

用户中断	用户中断状态
组 1	M8070
组 2	M8071
组 3	M8072
组 4	M8073
组 5	M8074
定时器中断	M8144

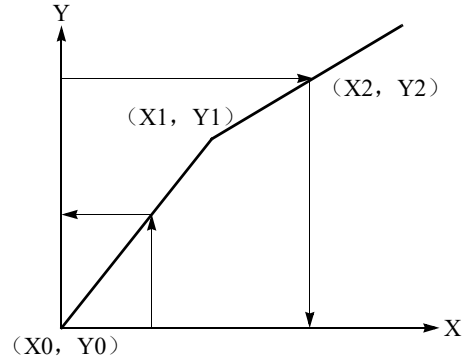
#### FC6A 型

用户中断	用户中断状态
组 1	M8137
组 2	M8140
组 3	M8141
组 4	M8142
组 5	M8143
组 6	M8167
定时器中断	M8144

# 16: 坐标转换指令

## 简介

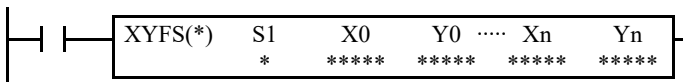
本章介绍一系列的坐标转换指令，以指定的 2 个以上的点为基准，计算 X-Y 平面上的连续直线，并求出任意 X 坐标相对应直线上的 Y 坐标或任意 Y 坐标相对应直线上的 X 坐标。



## XYFS (XY 格式设置)

根据两个或多个指定点，计算 X-Y 平面上的连续线段，并将其登记为 XY 格式。

FT2J FC6A



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。

XY 坐标编号	n
2 - 32	$0 \leq n \leq 31$

### 注释:

- XYFS 指令与 CVXTY 和 CVYTX 指令结合使用。
- 梯形图程序中可使用的 XYFS 指令数量取决于控制器类型。  
FT2J 型: 最多 8 条  
FC6A 型: 最多 30 条
- XYFS 指令不能在中断程序中使用。如果在中断程序中使用，将导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	格式编号	—	—	—	—	—	—	—	—	0-29	—
X0 - Xn	X 值	X	X	X	X	X	X	X	—	0 - 65,535	—
Y0 - Yn	Y 值	X	X	X	X	X	X	X	—	0 - 65,535 -32,768 - 32,767	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

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)

可设置的格式编号取决于控制器类型。

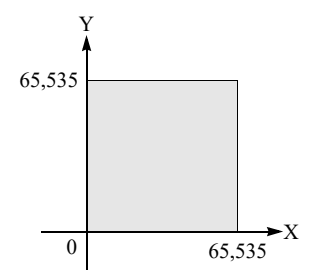
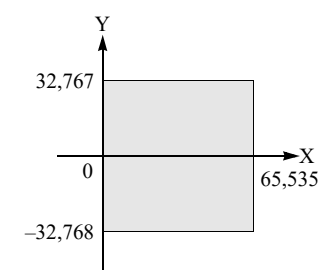
FT2J: 0 至 7

FC6A: 0 至 29

## 16: 坐标转换指令

### 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 - 65,535	0 - 65,535
Yn (Y 值)	0 - 65,535	-32,768 - 32,767
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)	—

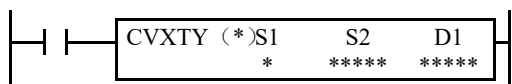
当 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定为 Xn 或 Yn 时, 将使用 16 点。

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。

## CVXTY (X → Y 转换)

根据指定格式编号的连续行计算与 X 值转换为相应的 Y 值。

FT2J FC6A



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 8 个 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。

### 注释:

- CVXTY 指令与 XYFS 指令结合使用。
- 转换结果的小数点后第一位四舍五入。
- 当 X 值存在多个 Y 值候选值时，XYFS 指令设置的坐标 (X<sub>n</sub>, Y<sub>n</sub>) 和 (X<sub>n+1</sub>, Y<sub>n+1</sub>) 连接线中 "n" 较小的线计算出的 Y 值就是 CVXTY 指令的转换结果。详情请参见第 16-7 页上的“示例: 重叠坐标”。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 如果未编入具有相应格式编号的 XYFS 指令，或者具有相同格式编号的 XYFS 和 CVXTY 指令的数据类型指定不同，将存储错误代码 12 在用户程序执行错误代码 (D8006) 中。
- 如果无法根据 S1 和 S2 计算出转换结果 D1，将存储错误代码 13 在用户程序执行错误代码 (D8006) 中。
- CVXTY 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	格式编号	—	—	—	—	—	—	—	—	0-29	—
S2 (源 2)	X 值	X	X	X	X	X	X	X	—	0 - 65,535	—
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，请参见第 2-1 页上的“设备地址”。

▲ 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。

### S1 (格式编号)

选择一个已经使用 XYFS 指令设置的格式编号。

### 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 - 65,535	0 - 65,535
D1 (Y 值)	0 - 65,535	-32,768 - 32,767

有效坐标	字	整数

## 16: 坐标转换指令

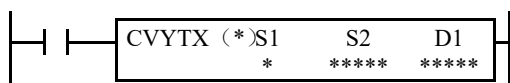
### 有效数据类型

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 S2 或 D1 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 S2 或 D1 时, 将使用 1 点。
L (长整数)	—	
F (浮点)	—	

### CVYTX (Y → X 转换)

根据指定格式编号的连续行计算与 Y 值转换为相应的 X 值。

FT2J FC6A



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 8 个 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。

#### 注释:

- CVYTX 指令与 XYFS 指令结合使用。
- 转换结果的小数点后第一位四舍五入。
- 当 Y 值存在多个 X 值候选值时, XYFS 指令设置的坐标 (X<sub>n</sub>, Y<sub>n</sub>) 和 (X<sub>n+1</sub>, Y<sub>n+1</sub>) 连接中 "n" 较小的线计算出的 X 值就是 CVYTX 指令的转换结果。详情请参见第 16-7 页上的“示例: 重叠坐标”。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情, 请参见第 3-11 页上的“用户程序执行错误”。
- 如果未编入具有相应格式编号的 XYFS 指令, 或者具有相同格式编号的 XYFS 和 CVYTX 指令的数据类型指定不同, 将存储错误代码 12 在用户程序执行错误代码 (D8006) 中。
- 如果无法根据 S1 和 S2 计算出转换结果 D1, 将存储错误代码 13 在用户程序执行错误代码 (D8006) 中。
- CVYTX 指令不能在中断程序中使用。如果在中断程序中使用, 将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	格式编号	—	—	—	—	—	—	—	—	0-29	—
S2 (源 2)	Y 值	X	X	X	X	X	X	X	—	0 - 65,535 -32,768 - 32,767	—
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, 请参见第 2-1 页上的“设备地址”。

▲ 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。

#### S1 (格式编号)

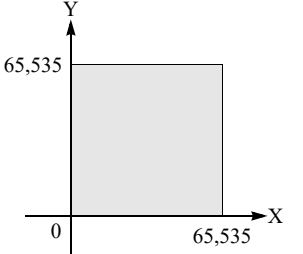
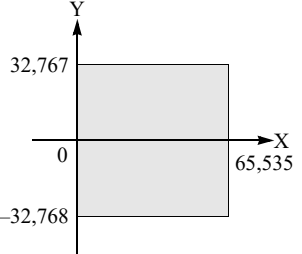
选择一个已经使用 XYFS 指令设置的格式编号。

#### 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 - 65,535	-32,768 - 32,767
D1 (X 值)	0 - 65,535	0 - 65,535
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)	—

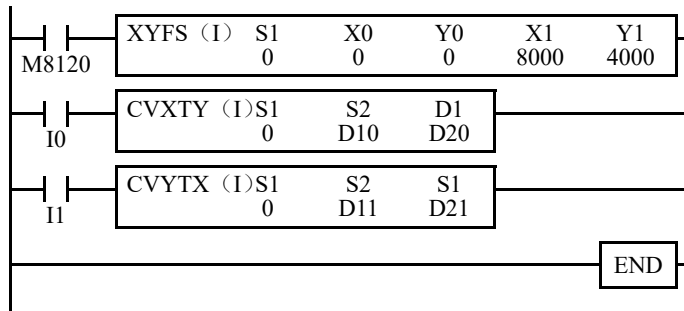
当 I (输入)、Q (输出)、M (内部继电器) 或 R (移位寄存器) 等位设备被指定为 S2 或 D1 时, 将使用 16 点。

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)。

## 16: 坐标转换指令

### 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  $(X1, Y1) = (8,000, 4,000)$ 。设置好后，将有  $X \rightarrow Y$  转换以及  $Y \rightarrow X$  转换。

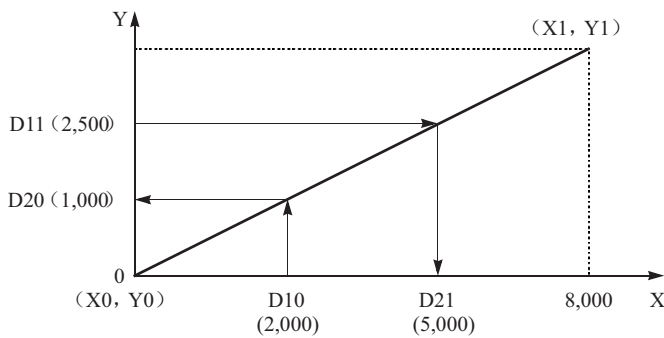


M8120 是初始化脉冲特殊内部继电器。

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 2,000，则指定给 D20 的值为 1,000。

对于  $Y \rightarrow X$  转换，将使用以下等式：

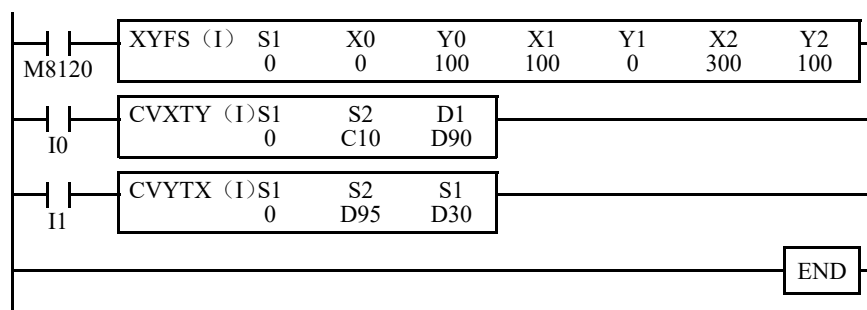
$$X = 2Y$$

如果寄存器 D11 中的值为 2,500，则指定给 D21 的值为 5,000。

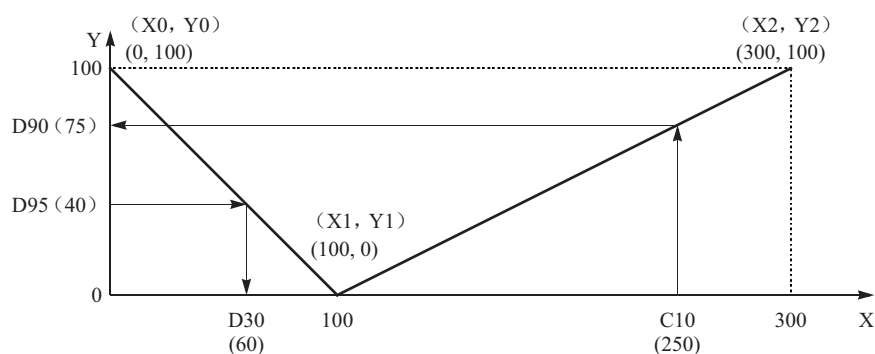


### 示例：重叠坐标

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。



M8120 是初始化脉冲特殊内部继电器。启动后，XYFS 会指定三个点。  
CVXTY 转换 C10 中的当前值，并将结果存储在 D90 中。  
CVYTX 转换 D95 中的值，并将结果存储在 D30 中。



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。



# 17: 平均指令

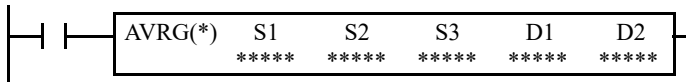
## 简介

本章介绍对指定数据进行平均化的平均指令。

## AVRG（平均化）

计算指定数据的平均值、最大值和最小值。

FT2J FC6A



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。

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

### 注释:

- 梯形图程序中可使用的 AVRG 指令数量取决于控制器类型。  
FT2J 型：最多 8 条  
FC6A 型：最多 32 条
- 当 S3（取样周期）设置为 0 时，如果 S2（取样结束输入）未打开，当采样周期超过 65,535 且发生溢出时，将输出结果。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当选择 F（浮点）数据类型，并且 S1 不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。
- AVRG 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	0-65,535	—
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，请参见第 2-1 页上的“设备地址”。

▲ 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 - 65,535 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。

## 17: 平均指令

---

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。

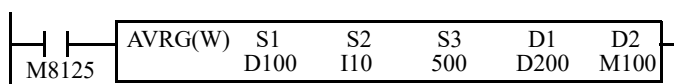
当没有使用取样结束输入时，指定一个内部继电器或其他有效设备作为源设备 S2 的空位。

### 有效数据类型

<b>W (字)</b>	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) 或 32 点 (双字或长整数数据)。
<b>I (整数)</b>	X	
<b>D (双字)</b>	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) 或 2 点 (双字、长整数或浮点数据)。
<b>L (长整数)</b>	X	
<b>F (浮点)</b>	X	

**示例：AVRG**

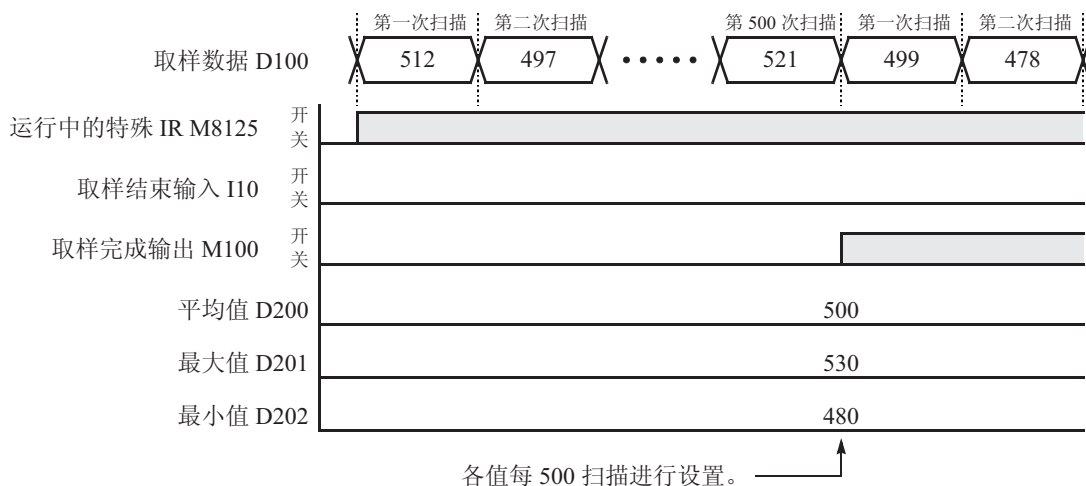
下面的示例演示的是程序在每 500 次扫描中计算数据寄存器 D100 的平均值并将结果存储到数据寄存器 D200。



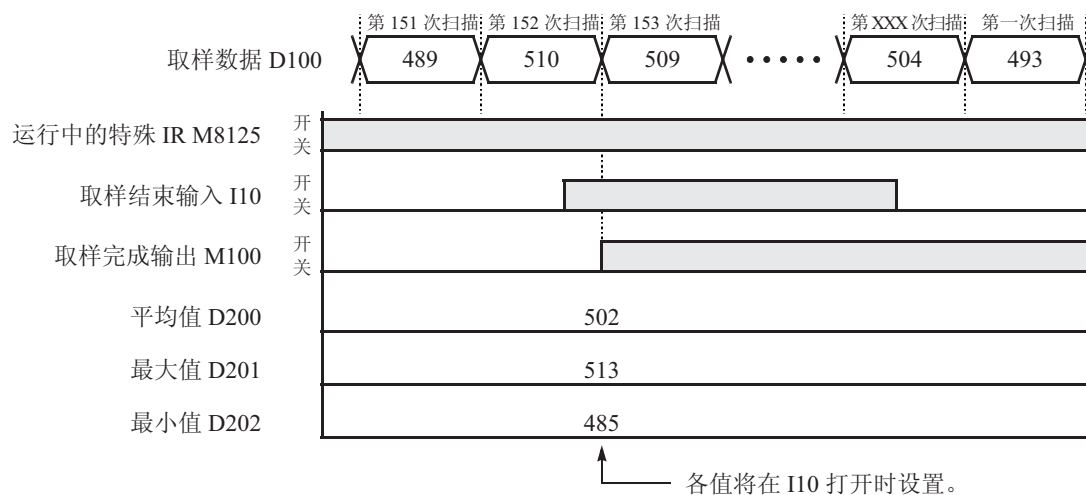
M8125 是运行中输出特殊内部继电器。

**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。





# 18: 脉冲输出指令

## 简介

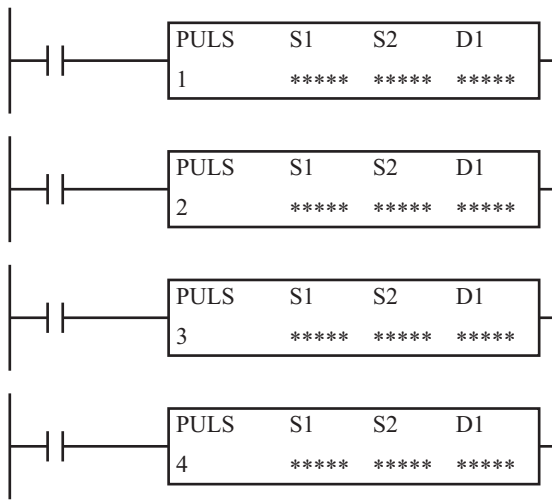
通过脉冲输出输出指定频率和占空比的脉冲。

## PULS（脉冲输出）

根据指定格式编号的连续行计算与 X 值转换为相应的 Y 值。

FT2J\* FC6A\*

※ 仅晶体管输出类型



从脉冲输出以固定占空比输出指定频率的脉冲。

当输入打开后，系统将根据 S1 指定的控制寄存器设置，以固定占空比输出脉冲。

当 S2 指定的初始化输入打开后，“设置”选项卡上 WindLDR “PULS 指令”对话框中配置的初始值将存储到控制寄存器中。

当 S2 指定的初始化输入打开后，设置选项卡上 WindLDR PWM 指令对话框中配置的初始值将存储到控制寄存器中。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- PULS 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 PULS 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
D1 (目标 1)	操作状态	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1~7。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 18: 脉冲输出指令

### 设置

#### ■ “设备”选项卡



#### 1. 选择模式\*1

选择配置模式。可选择“FC6A 标准模式”或“FC5A 兼容模式”。

希望在 FC5A 型 MICROSmart 的 PULS 指令规格下使用时，选择“FC5A 兼容模式”。

希望变更 FC5A/FC4A 型 MICRO Smart 类型时，可自动选择“FC5A 兼容模式”。

另外，“FC5A 兼容模式”可使用的最低频率为 20Hz。

#### 注释：

- 有关 FC5A 兼容模式下的配置详情，请参见《FC5A 系列 MICROSmart Pentra 用户手册 高级卷》第 15 章“脉冲指令”的 PULS 指令。
- FC5A 兼容模式可输出脉冲的最低频率为 20Hz。无法输出慢于、低于 20Hz 频率的脉冲。指定后将导致脉冲频率配置错误。
- FC5A 兼容模式的 PULS3 可输出的最高脉冲频率为 5kHz。无法输出高于 5kHz 的脉冲。指定后将导致脉冲频率错误。

\*1 仅 FC6A 型

以下记录了选择“FC6A 标准模式”的情况。

#### 2. 选择指令

此项选择要使用的 PULS 指令（“PULS1”、“PULS2”、“PULS3”或“PULS4”）。

脉冲输出位置和可配置的频率因指令和控制器的类型而异。

#### FT2J 型

指令	脉冲输出	可配置频率
PULS1 (Q0000)	Q0	15Hz~20kHz（增量为1Hz）
PULS2 (Q0001)	Q1	
PULS3 (Q0002)	Q2	
PULS4 (Q0003)	Q3	

#### FC6A 型

##### All-in-One CPU 模块

指令	脉冲输出	可配置频率
PULS1 (Q0000)	Q0	15Hz ~ 100kHz（增量为 1Hz）
PULS2 (Q0001)	Q1	
PULS3 (Q0002)	Q2	15Hz ~ 5kHz（增量为 1Hz）
PULS4 (Q0003)	Q3	



## CAN J1939 All-in-One CPU 模块 /Plus CPU 模块

指令	脉冲输出	可配置频率
PULS1 (Q0000)	Q0	15Hz~100kHz (增量为1Hz)
PULS2 (Q0002)	Q2	
PULS3 (Q0004)	Q4	
PULS4 (Q0006)	Q6	

## 3. S1 (源 1): 控制寄存器

S1 指定用于 PULS1、PULS2、PULS3 或 PULS4 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始, 使用数据寄存器的 8 个连续字。

请指定设备范围内的第一数据寄存器。

操作模式和脉冲频率配置的设置因控制器的类型和指令而异。

## FT2J 型

存储目的地	功能	设置	参考
起始编号+0	输出脉冲频率 (高位字)	PULS1 ~ PULS4: 15~20,000 (增量为1Hz)	第18-4页上的“6. 输出脉冲频率”
起始编号+1	输出脉冲频率 (低位字)		
起始编号+2	预置值 (高位字)	1~100,000,000次脉冲	第18-4页上的“8. 预置值”
起始编号+3	预置值 (低位字)		
起始编号+4	当前值 (高位字)	1~100,000,000次脉冲	第18-4页上的“9. 当前值”
起始编号+5	当前值 (低位字)		
起始编号+6	错误状态	当设置中出现错误时, 存储与错误内容相对应的错误代码。	第18-5页上的“10. 错误状态”
起始编号+7	保留		

## FC6A 型

存储目的地	功能	设置		参考
		All-in-One CPU 模块	CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	
起始编号 +0	输出脉冲频率*1 (高位字)	PULS1、PULS2: 15 ~ 100,000 (增量为 1Hz)	PULS1 ~ PULS4: 15 ~ 100,000 (增量为 1Hz)	第 18-4 页上的 “6. 输出脉冲频率”
起始编号 +1	输出脉冲频率*1 (低位字)	PULS3、PULS4: 15 ~ 5,000 (增量为 1Hz)		
起始编号 +2	预置值 (高位字)*1	1 ~ 100,000,000 次脉冲		第 18-4 页上的 “8. 预置值”
起始编号 +3	预置值 (低位字)*1			
起始编号 +4	当前值 (高位字)*1	1 ~ 100,000,000 次脉冲		第 18-4 页上的 “9. 当前值”
起始编号 +5	当前值 (低位字)*1			
起始编号 +6	错误状态	当设置中出现错误时, 存储与错误内容相对应的错误代码。		第 18-5 页上的 “10. 错误状态”
起始编号 +7	保留			

\*1 FC6A 型根据指定的 32 位数据存储方法更改高位和低位数据寄存器。  
有关详情, 请参见第 3-10 页上的 “32 位数据存储”。

## 4. S2 (源 2): 初始化输入

S2 指定初始化输入。当初始化输入打开后, “设置”选项卡上 WindLDR “PULS”对话框中配置的初始值将存储到控制寄存器中。可以指定外部输入或内部继电器。

当初始化输入打开后, 初始值将在每次扫描时写入数据寄存器中。(即使在不执行 PULS 指令 (未打开) 的状态下, 若打开初始化输入, 初始值也将存储在数据寄存器中。)要对值仅初始化一次, 请将初始化输入与 SOTU (上升沿微分) 指令或 SOTD (下降沿微分) 指令结合使用。

## 5. D1 (目标 1): 运行状态

D1 指定用于存储操作状态的内部继电器的起始编号。

从指定的内部继电器开始, 使用 3 个连续的内部继电器。

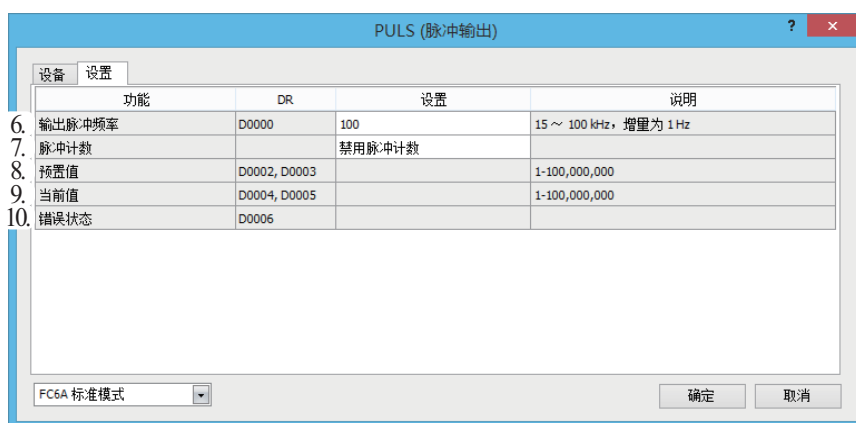
请指定设备范围内的起始编号。

## 18: 脉冲输出指令

存储目的地	功能	设置	
起始编号 +0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	此继电器在脉冲输出过程中打开。 此继电器在脉冲输出停止后关闭。 此继电器在指定次数的脉冲输出结束后关闭。
起始编号 +1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	此继电器在脉冲输出完成后打开。 此继电器在脉冲输出开始后关闭。
起始编号 +2	上溢出	0: 无 1: 出现上溢出	启用脉冲计数后, 此继电器将在输出的脉冲超过配置的预置值时打开。

### ■ “设置”选项卡

设置 PULS 指令的操作。



### 6. 输出脉冲频率

指定脉冲频率。输出频率误差限于  $\pm 5\%$ 。

配置的设置因控制器的类型和指令而异。

#### FT2J型

指令	可配置范围	
	设置值	频率
PULS1 ~ PULS4	15 ~ 20,000	15Hz ~ 20kHz (增量为 1Hz)

输出频率误差为  $\pm 2\%$ 。

#### FC6A 型

CPU 模块的类型	指令	可配置范围	
		设置值	频率
All-in-One CPU 模块	PULS1、PULS2	15 ~ 100,000	15Hz ~ 100kHz (增量为 1Hz)
	PULS3、PULS4	15 ~ 5,000	15Hz ~ 5kHz (增量为 1Hz)
CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	PULS1 ~ PULS4	15 ~ 100,000	15Hz ~ 100kHz (增量为 1Hz)

输出频率误差为  $\pm 5\%$ 。

### 7. 脉冲计数

指定启用、禁用脉冲计数。

指定为“启用脉冲计数”时, 输出预置值 (8.) 中指定的脉冲数。

指定为“禁用脉冲计数”时, 在 PULS 指令输入打开期间, 连续输出脉冲。

### 8. 预置值

7. 脉冲计数配置为“启用脉冲计数”时, 指定输出的脉冲数。

### 9. 当前值

指定存储输出脉冲数的设备 (数据寄存器)。

执行 PULS 指令时, 每次扫描都会更新当前值。

## 10. 错误状态

若配置的设置有误，则输出有误设置对应的错误代码。如果在 PULS 指令输入从关闭变为打开时出现配置错误，将导致用户程序执行错误，在 D8006 中存储错误代码 20。

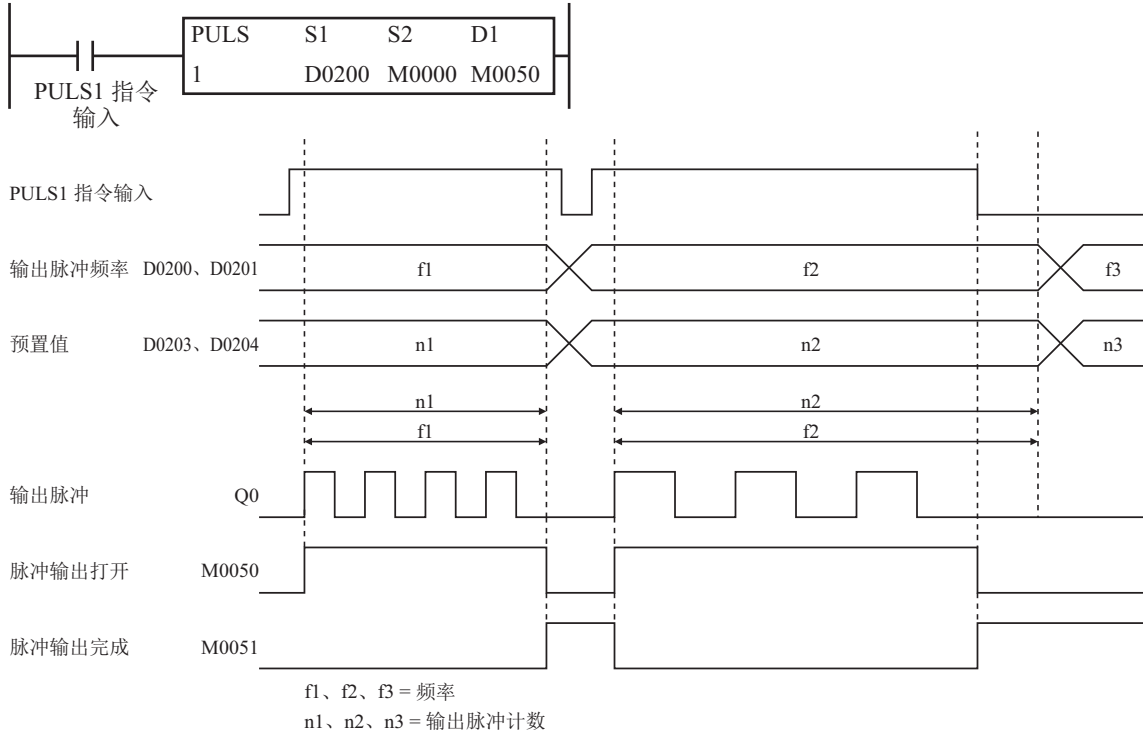
错误代码	状态	说明	
0	正常	—	
2	脉冲频率指定错误	All-in-One CPU 模块	PULS1、PULS2: 脉冲频率未设置在 15 到 100,000 之间。 PULS3、PULS4: 脉冲频率未设置在 15 到 5,000 之间。
		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	PULS1 ~ PULS4: 脉冲频率未设置在 15 到 100,000 之间。
4	预置值指定错误	指定为“启用脉冲计数”时，预置值配置范围在 1 ~ 100,000,000 之外。	

## 18: 脉冲输出指令

### 示例: PULS

#### PULS1 指令（启用脉冲计数）时间图

[PULS1 指令, S1 指定为 D0200, D1 指定为内部继电器 M0050]



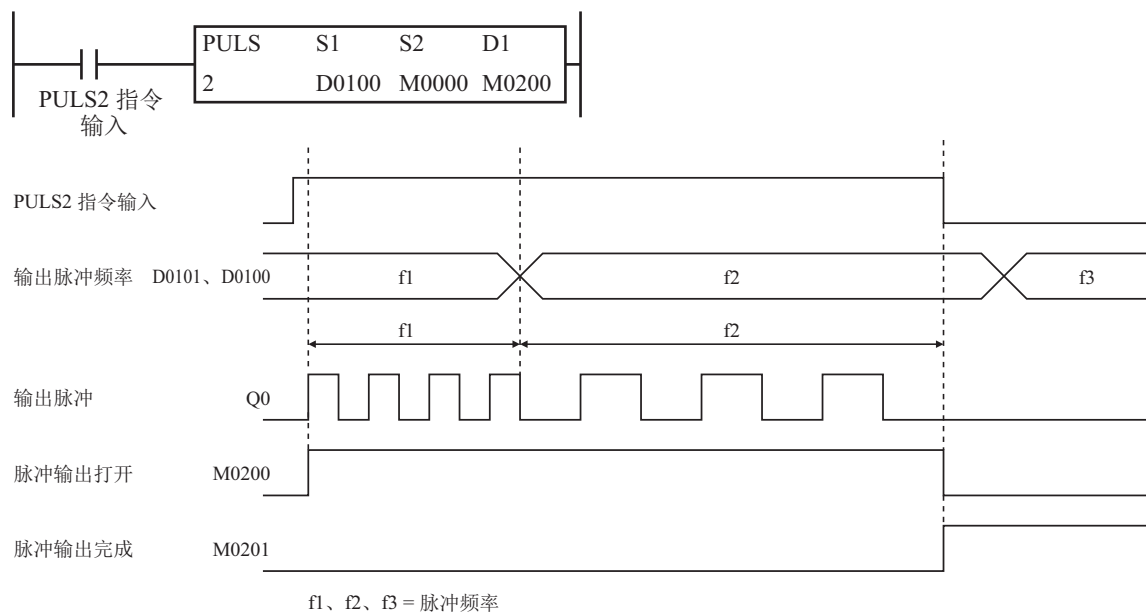
当 PULS1 指令输入从关闭变为打开时, M0050 打开, 并且输出 D0200 和 D0201 配置的频率的脉冲。输出 D0202 和 D0203 配置的脉冲数时, 脉冲输出停止。如果 D0200 和 D0201 的值在脉冲输出期间发生变化, 则系统将输出基于该值的频率的脉冲。

当 PULS1 指令输入从打开变为关闭时, M0050 关闭, 同时 M0051 会打开。

PULS1 指令输入打开时, 系统不会反映初始化输入的变化。如果要将数据寄存器初始化为初始化输入, 请在关闭输入后打开初始化输入。

## PULS2 指令（禁用脉冲计数）时间图

[PULS2 指令，S1 指定为 D0100，D1 指定为内部继电器 M0200]



当 PULS2 指令输入从关闭变为打开时，M0200 打开，并且输出 D0101 和 D0100 配置的频率的脉冲。如果 D0101 和 D0100 的值在脉冲输出期间发生变化，则系统将输出基于该值的频率的脉冲。

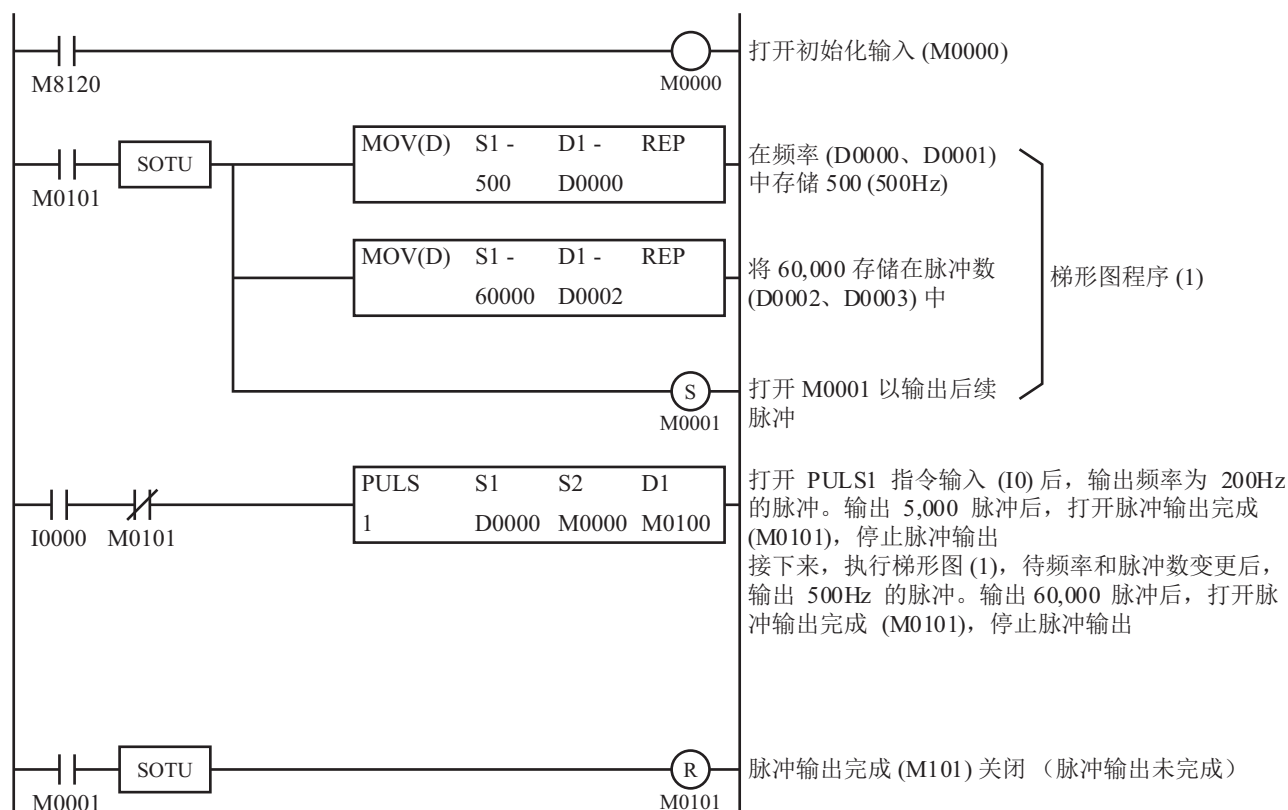
当 PULS2 指令输入从打开变为关闭时，M0200 关闭，同时 M0201 会打开。

PULS2 指令输入打开时，系统不会反映初始化输入的变化。如果要将数据寄存器初始化为初始化输入，请在关闭输入后打开初始化输入。

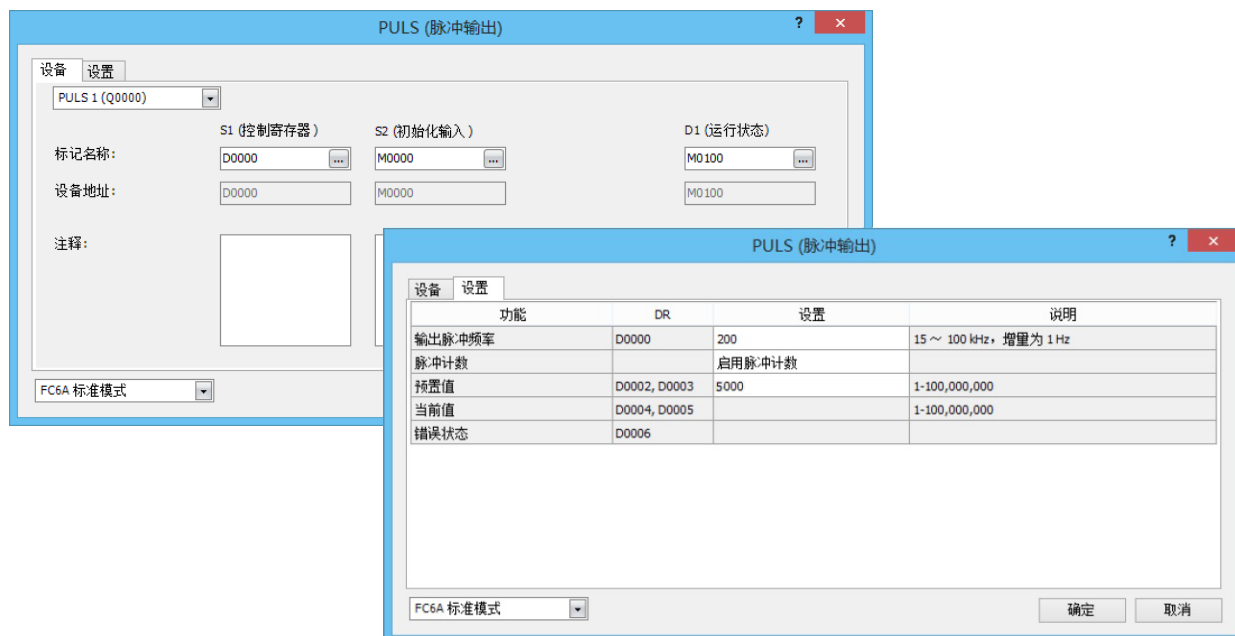
## 18: 脉冲输出指令

### 示例程序

本节介绍一个示例程序，它从输出 Q0 先后输出 5,000 个脉冲（每次 200Hz）和 60,000 个脉冲（每次 500Hz）



### 设置



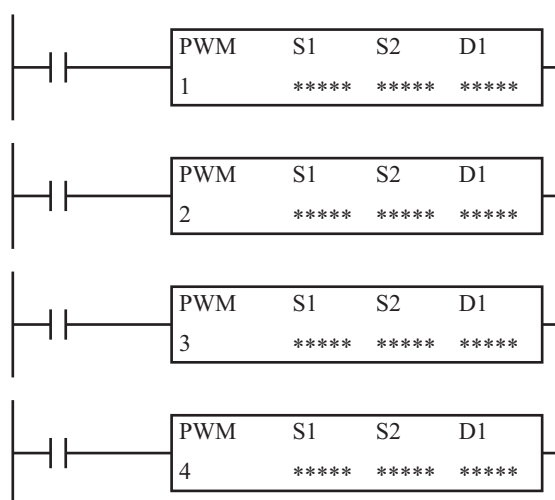
功能	设备地址	设置值	详情
输出脉冲频率	D0000	200	200Hz
脉冲计数	—	1	启用脉冲计数
预置值	D0002、D0003	5000	预置值 =5,000

## PWM（可变占空比脉冲输出）

根据指定格式编号的连续行计算与 X 值转换为相应的 Y 值。

FT2J<sup>※</sup> FC6A<sup>※</sup>

※ 仅晶体管输出类型



从脉冲输出端口输出指定频率和占空比的脉冲。

当输入打开后，系统将根据 S1 指定的控制寄存器设置输出脉冲。

脉冲控制信息（输出打开 / 输出完成 / 错误）作为操作状态存储在 D1 指定的内部继电器中。

当 S2 指定的初始化输入打开后，“设置”选项卡上 WindLDR “PWM (脉宽调制)”对话框中配置的初始值将存储到控制寄存器中。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- PWM 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 PWM 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
D1 (目标 1)	操作状态	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1 ~ 7。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 18: 脉冲输出指令

### 设置

#### ■ “设备”选项卡



#### 1. 选择模式

选择配置模式。可从“FC6A 标准模式”、“FC4A 兼容模式”、“FC5A（FC5A-D12X1E 除外）兼容模式”、“FC5A-D12X1E 兼容模式”中选择。

希望以 FC4A 型 MICROSmart、FC5A 型 MICROSmart、FC5A-D12 的 PWM 指令规格使用时，请选择任意一项 FC 兼容模式。希望变更 FC4A 型 MICROSmart 类型时，自动选择“FC4A 兼容模式”。变更 FC5A、FC5A-D12 类型时，分别选择“FC5A（FC5A-D12X1E 除外）兼容模式”、“FC5A-D12X1E 兼容模式”。

可在 FC 兼容模式下指定的频率为 FC4A、FC5A、FC5A-D12 各自 PWM 指令频率的近似值。

#### 注释：

- 有关 FC4A 兼容模式下的配置详情，请参见《FC4A 系列 MICROSmart Pentra 用户手册》第 2 章“指令”的 PWM 指令。
- 有关 FC5A（FC5A-D12X1E 除外）兼容模式、FC5A-D12X1E 兼容模式下的配置详情，请参见《FC5A 系列 MICROSmart 用户手册 高级卷》“第 13 章 脉冲指令”的 PWM 指令。

注释：可在 FC 兼容模式下输出的脉冲频率如下所示。

动作模式	FC4A 兼容模式	FC5A（FC5A-D12X1E 除外）兼容模式	FC5A-D12X1E 兼容模式
动作模式 0	15Hz 固定	15Hz 固定	15Hz 固定
动作模式 1	27Hz 固定	46Hz 固定	61Hz 固定
动作模式 2	218Hz 固定	366Hz 固定	488Hz 固定

以下记录了选择“FC6A 标准模式”的情况。

#### 2. 选择指令

此项选择要使用的 PWM 指令（“PWM1”、“PWM2”、“PWM3”或“PWM4”）。

脉冲输出位置因指令和控制器的类型而异。

##### FT2J 型

指令	脉冲输出	可配置范围	
		频率	占空比
PWM1 (Q0000)	Q0	30Hz ~ 1kHz（增量为 1Hz）	0.1 ~ 100.0（增量为 0.1%）
PWM2 (Q0001)	Q1		
PWM3 (Q0002)	Q2		
PWM4 (Q0003)	Q3		

配置输出脉冲频率和脉冲占空比，使 ON 时间和 OFF 时间大于等于 25μs。

有关各输出的输出延迟时间详情，请参见《SmartAXIS 硬件手册》第 1 章“主机规格”。



## FC6A 型

## All-in-One CPU 模块

指令	脉冲输出	可配置范围	
		频率	占空比
PWM1	Q0	15Hz ~ 5kHz (增量为 1Hz)	0.1 ~ 100.0 (增量为 0.1%)
PWM2	Q1		
PWM3	Q2		
PWM4	Q3		

当输出为 Q0 或 Q1 且计算出的 OFF 时间短于 15 $\mu$ s 时，将调整脉冲占空比，使 OFF 时间变为 15 $\mu$ s，然后输出脉冲。

当输出为 Q2 或 Q3 时，配置输出脉冲频率和脉冲占空比，使 ON 时间和 OFF 时间大于或等于 100 $\mu$ s。

有关各输出的输出延迟时间详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“产品规格”。

## CAN J1939 All-in-One CPU 模块 /Plus CPU 模块

指令	脉冲输出	可配置范围	
		频率	占空比
PWM1	Q0	15Hz ~ 5kHz (增量为 1Hz)	0.1 ~ 100.0 (增量为 0.1%)
PWM2	Q2		
PWM3	Q4		
PWM4	Q6		

当输出为 Q0 或 Q1 且计算出的 OFF 时间短于 15 $\mu$ s 时，将调整脉冲占空比，使 OFF 时间变为 15 $\mu$ s，然后输出脉冲。

有关各输出的输出延迟时间详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“产品规格”。

## 3. S1 (源 1): 控制寄存器

S1 指定用于 PWM1、PWM2、PWM3 或 PWM4 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始，使用 8 个连续的数据寄存器。

请指定设备范围内的第一数据寄存器。

存储目的地	功能	设置	参考
起始编号 +0	输出脉冲频率	15 ~ 5,000 (增量为 1Hz)	第 18-12 页上的“6. 输出脉冲频率”
起始编号 +1	脉冲占空比 (打开比率)	1 ~ 1,000 (增量为 0.1%)	第 18-12 页上的“7. 脉宽比率”
		1 ~ 100 (增量为 1%)	
起始编号 +2	预置值 (高位字) *1	1 ~ 100,000,000 次脉冲	第 18-12 页上的“9. 预置值”
起始编号 +3	预置值 (低位字) *1		
起始编号 +4	当前值 (高位字) *1	1 ~ 100,000,000 次脉冲	第 18-12 页上的“10. 当前值”
起始编号 +5	当前值 (低位字) *1		
起始编号 +6	错误状态	0 ~ 4	第 18-13 页上的“11. 错误状态”
起始编号 +7	保留		

\*1 FC6A 型根据指定的 32 位数据存储方法更改高位和低位数据寄存器。

有关详情，请参见第 3-10 页上的“32 位数据存储”。

## 4. S2 (源 2): 初始化输入

S2 指定初始化输入。当初始化输入 S2 打开后，“设置”选项卡上 WindLDR “PWM (脉宽调制)”对话框中配置的初始值将存储到控制寄存器中。可以指定外部输入或内部继电器。

当初始化输入打开后，初始值将在每次扫描时写入数据寄存器中。要对值仅初始化一次，请将初始化输入与 SOTU (上升沿微分) 指令或 SOTD (下降沿微分) 指令结合使用。

## 18: 脉冲输出指令

### 5. D1（目标 1）：运行状态

D1 指定用于 PWM 指令的内部继电器的第一内部继电器。  
从指定的内部继电器开始，使用 3 个连续的内部继电器。  
请指定设备范围内的第一内部继电器。

存储目的地	功能	设置	
起始编号 +0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	此继电器在脉冲输出过程中打开。 此继电器在脉冲输出停止后关闭。 此继电器在指定次数的脉冲输出结束后关闭。
起始编号 +1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	此继电器在脉冲输出完成后打开。 此继电器在脉冲输出停止后关闭。
起始编号 +2	上溢出	0: 无 1: 出现上溢出	启用脉冲计数后，此继电器将在输出的脉冲超过配置的预置值时打开。

### ■ “设置”选项卡



### 6. 输出脉冲频率

指定输出的脉冲频率。输出脉冲频率的可配置范围取决于控制器类型。

控制器	可配置范围	
	输出脉冲频率	输出脉冲频率误差
FT2J 型	30Hz ~ 1kHz（增量为 1Hz）	±2% 以内
FC6A 型	15Hz ~ 5kHz（增量为 1Hz）	±5% 以内

### 7. 脉宽比率

此设置指定要输出的脉冲频率的打开比率（占空比）。

配置模式为“FC6A 标准模式”时，指定为 0.1% ~ 100.0%（增量为 0.1%）之间。为“FC4A 兼容模式”、“FC5A（FC5A-D12X1E 除外）兼容模式”、“FC5A-D12X1E 兼容模式”时，指定为 1% ~ 100%（增量为 1%）之间。

### 8. 脉冲计数

指定启用、禁用脉冲计数。

指定为“启用脉冲计数”时，输出预置值（9.）中指定的脉冲数。

指定为“禁用脉冲计数”时，在 PWM 指令输入打开期间，连续输出脉冲。

### 9. 预置值

8. 脉冲计数配置为“启用脉冲计数”时，指定输出的脉冲数。

### 10. 当前值

指定存储输出脉冲数的设备（数据寄存器）。

执行 PWM 指令时，每次扫描都会更新当前值。

### 11. 错误状态

若配置的设置有误，则输出有误设置对应的错误代码。如果在 PWM 指令输入从关闭变为打开时出现配置错误，将导致用户程序执行错误，在 D8006 中存储错误代码 20。

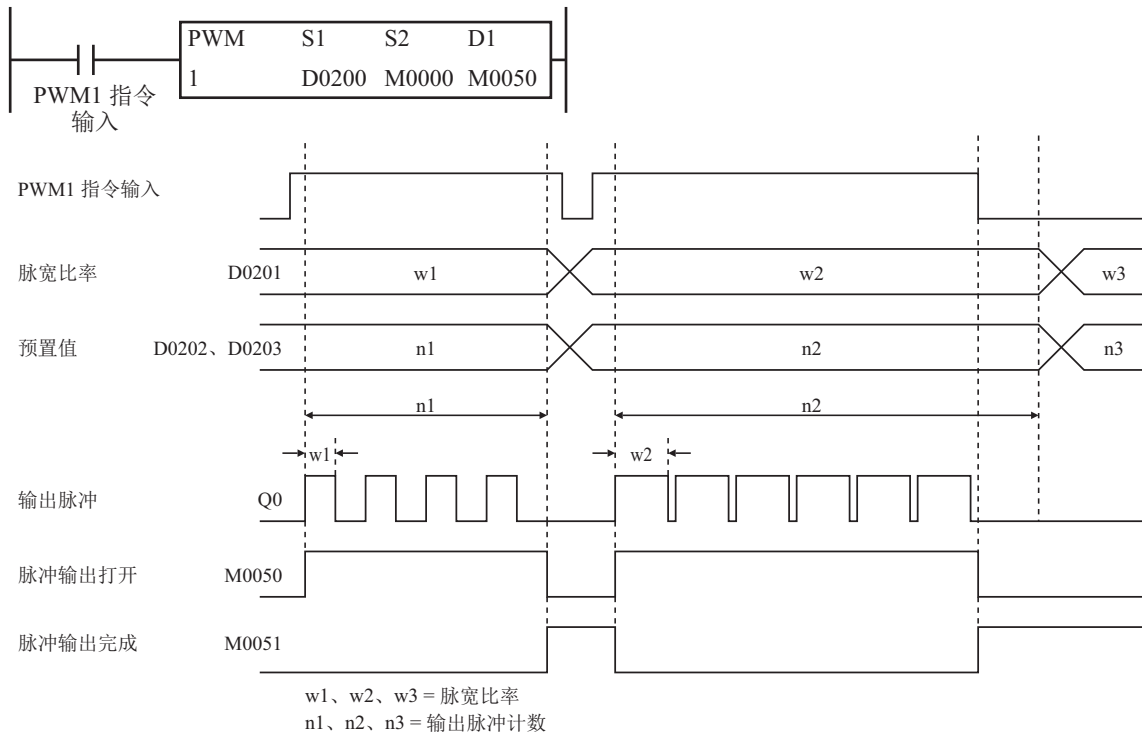
错误代码	状态	说明
0	正常	—
1	输出频率配置错误	脉冲频率未设置在 15 到 5,000 之间。
2	脉宽比率指定错误	脉宽比率未设置在 1 到 1,000 之间。
4	预置值指定错误	指定为“启用脉冲计数”时，预置值配置范围在 1 ~ 100,000,000 之外。

## 18: 脉冲输出指令

### 示例: PWM

#### PWM1 指令（启用脉冲计数）时间图

[PWM1 指令, S1 指定为 D200, D1 指定为内部继电器 M0050]



当 PWM1 指令输入从关闭变为打开时, M0050 打开, 并且输出 D0201 配置的宽比的脉冲。输出 D0202 和 D0203 配置的脉冲数时, 脉冲输出停止。如果 D0201 的值在脉冲输出期间发生变化, 则系统将输出基于该值的宽比的脉冲。

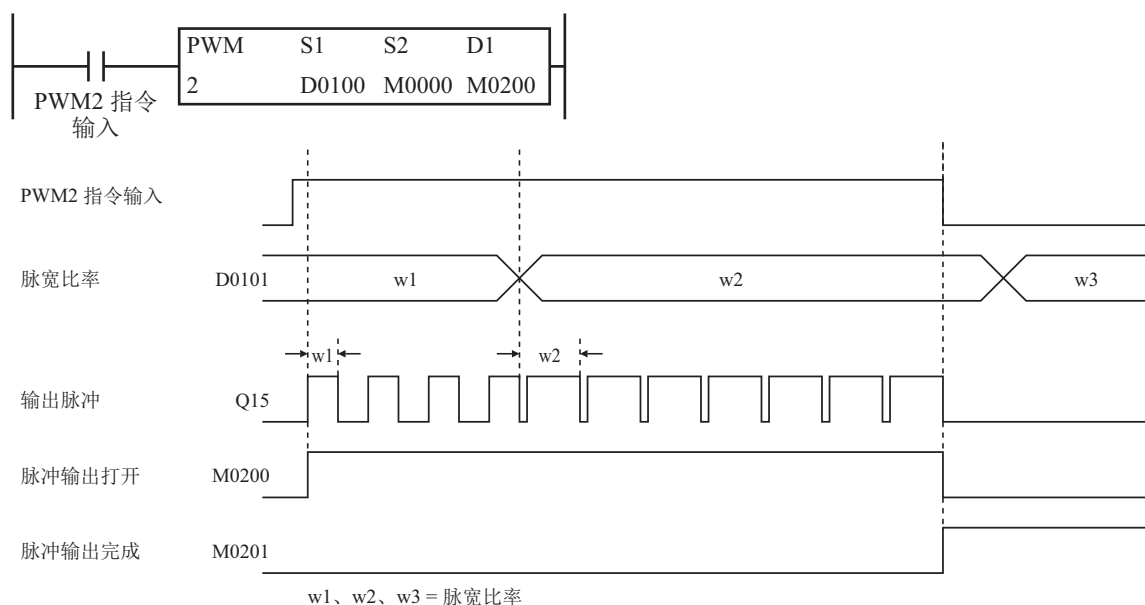
设置宽比变化的间隔 (周期), 使其相对于输出脉冲频率足够长。

当 PWM1 指令输入从打开变为关闭时, M0050 关闭, 同时 M0051 会打开。

PWM1 指令输入打开时, 系统不会反映初始化输入的变化。如果要将数据寄存器初始化为初始化输入, 请在关闭输入后打开初始化输入。

## PWM2 指令（禁用脉冲计数）时间图

[PWM2 指令，S1 指定为 D0100，D1 指定为内部继电器 M0200]



当 PWM2 指令输入从关闭变为打开时，M0200 打开，并且输出 D0101 配置的宽比的脉冲。如果 D0101 的值在脉冲输出期间发生变化，则系统将输出基于该值的宽比的脉冲。设置宽比变化的间隔（周期），使其相对于输出脉冲频率足够长。

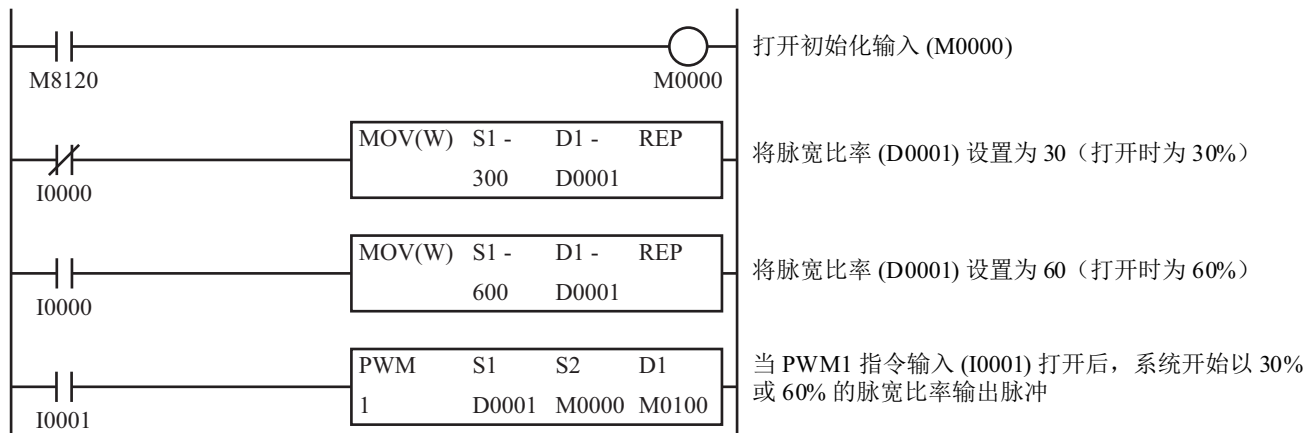
当 PWM2 指令输入从打开变为关闭时，M0200 关闭，同时 M0201 会打开。

PWM2 指令输入打开时，系统不会反映初始化输入的变化。如果要将数据寄存器初始化为初始化输入，请在关闭输入后打开初始化输入。

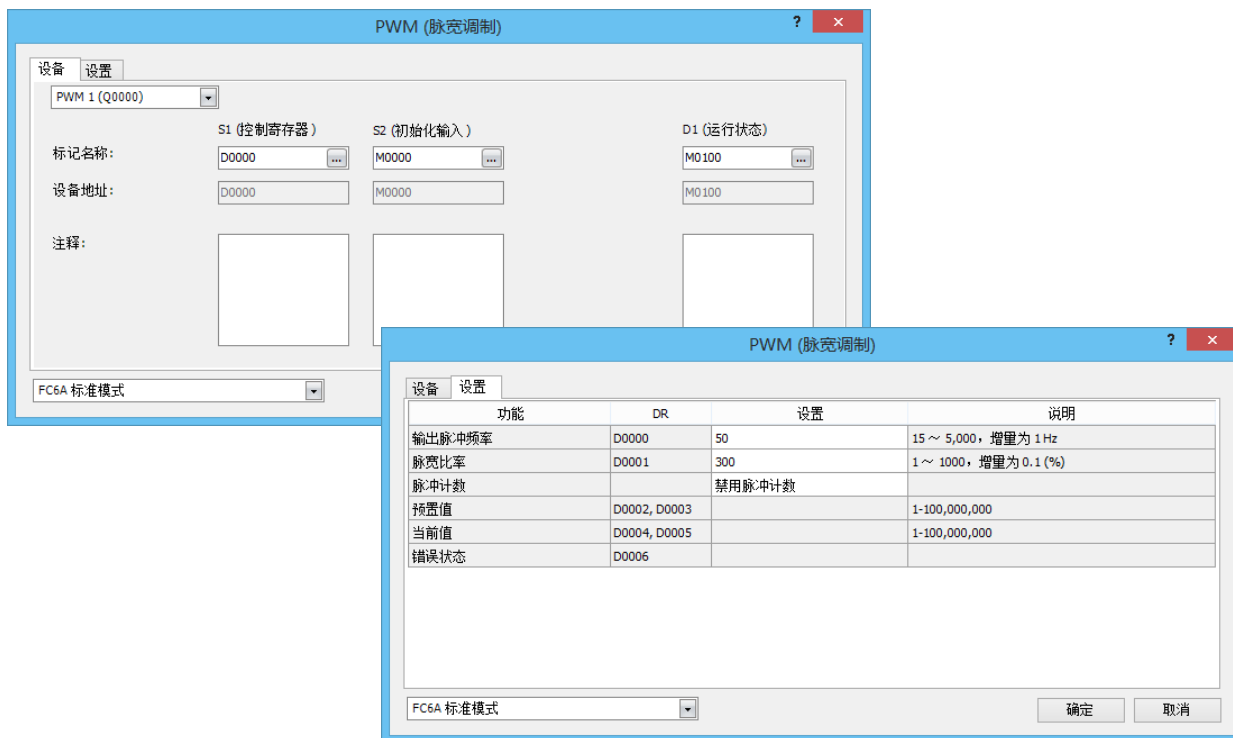
## 18: 脉冲输出指令

### 示例程序

本节介绍一个示例程序，它在 I0 关闭和打开时分别以 30% 和 60% 的脉宽比率输出脉冲。



### 设置



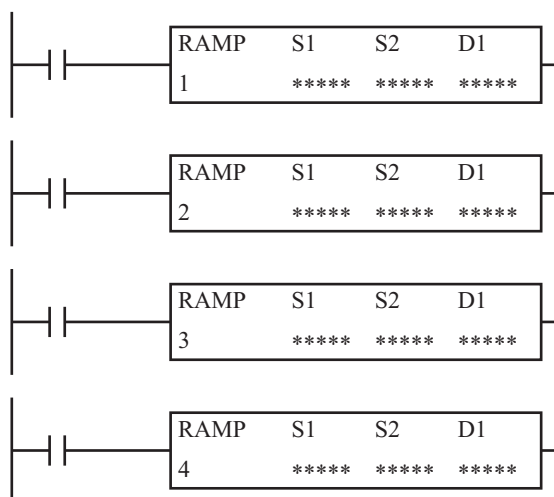
功能	设备地址	设置值	详情
输出脉冲频率	D0000	50	50Hz
脉宽比率	D0001	300	30%
脉冲计数	—	禁用脉冲计数	—
预置值	D0002、D0003	—	—

## RAMP（台形控制）

输出具有频率变化功能的脉冲。

FT2J FC6A\*

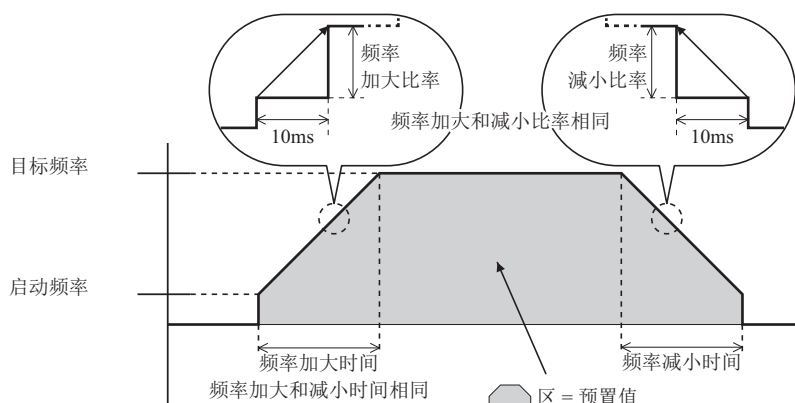
※ 仅晶体管输出类型



RAMP 指令利用频率变化功能输出脉冲。

当输入打开时，系统将输出 S1 指定的启动频率的脉冲，然后脉冲频率将增加固定的比率，直到达到目标频率。

当脉冲输出按目标频率输出后，在脉冲计数达到 S1 指定的预置值之前，脉冲频率将减小，达到预置值后脉冲输出停止。



S2 中指定的初始化输入打开时，将“设置”选项卡上 WindLDR “RAMP（台形控制）”对话框中配置的初始值存储到控制寄存器中。

脉冲控制信息（输出打开 / 输出完成 / 错误）作为操作状态存储在 D1 指定的内部继电器中。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- RAMP 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 RAMP 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2（源 2）	初始化输入	X	—	X	—	—	—	—	—	—	—
D1（目标 1）	操作状态	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1～7。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 设置

- “设备”选项卡



### 1. 选择模式

选择配置模式。可选择“FC6A 标准模式”或“FC5A（FC5A-D12X1E 除外）兼容模式”。

希望在 FC5A 型 MICROSmart 的 PULS 指令规格下使用时，选择“FC5A（FC5A-D12X1E 除外）兼容模式”。

希望变更 FC5A/FC4A 型 MICRO Smart 类型时，可自动选择“FC5A（FC5A-D12X1E 除外）兼容模式”。

另外，“FC5A（FC5A-D12X1E 除外）兼容模式”可使用的最低频率为 20Hz\*。

\*1 FC5A 型 MICROSmart 可输出的频率下限值为 10Hz，但兼容模式下为 20Hz。

#### 注释：

- FC5A（FC5A-D12X1E 除外）兼容模式可输出脉冲的最低频率为 20Hz。无法输出慢于、低于 20Hz 频率的脉冲。指定后将导致脉冲频率配置错误。
- FC5A（FC5A-D12X1E 除外）兼容模式的 RAMP 可输出的最高脉冲频率为 5kHz。无法输出高于 5kHz 的脉冲。指定后将导致脉冲频率错误。
- FC5A 兼容模式不支持 S 形频率变化曲线。

以下记录了选择“FC6A 标准模式”的情况。

**注释：**有关 FC5A（FC5A-D12X1E 除外）兼容模式下的配置详情，请参见《FC5A 系列 MICROSmart Pentra 用户手册 高级卷》第 13 章“脉冲指令”的 RAMP 指令。

### 2. 选择指令

此项选择要使用的 RAMP 指令（“RAMP1”、“RAMP2”、“RAMP3”或“RAMP4”）。

输出、可配置的方向控制模式和频率因指令和 CPU 模块的类型而异。

有关指令组合、方向控制模式和脉冲输出模式导致的限制，请参见第 18-21 页上的“10. 启用方向控制”。



### 3. S1（源 1）：控制寄存器

S1 指定用于 RAMP1、RAMP2、RAMP3 或 RAMP4 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始，使用 12 个连续的数据寄存器。

请指定设备范围内的第一数据寄存器。

存储目的地	功能	设置		参考
		All-in-One CPU 模块	CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	
起始编号 +0	目标频率 (高位字) *1	RAMP1、RAMP2: 15-100,000 (增量为 1Hz)	PULS1-PULS4: 15-100,000 (增量为 1Hz)	第 18-20 页上的“6. 目标频率”
起始编号 +1	目标频率 (低位字) *1	RAMP3、RAMP4: 15-5,000 (增量为 1Hz)		
起始编号 +2	启动频率 (高位字) *1	PULS1、PULS2: 15-100,000 (增量为 1Hz)	PULS1-PULS4: 15-100,000 (增量为 1Hz)	第 18-20 页上的“7. 归零速度脉冲频率”
起始编号 +3	启动频率 (低位字) *1	PULS3、PULS4: 15-5,000 (增量为 1Hz)		
起始编号 +4	频率更改时间	10-10,000ms		第 18-20 页上的“8. 频率时间”
起始编号 +5	控制方向	0: 正向 1: 反向		第 18-21 页上的“11. 控制方向”
起始编号 +6	预置值 (高位字) *1	禁用绝对位置指定模式时 1 ~ 100,000,000 (脉冲)		第 18-22 页上的“13. 预置值”
起始编号 +7	预置值 (低位字) *1	启用绝对位置指定模式时 -2,147,483,648 ~ 2,147,483,647 (脉冲)		
起始编号 +8	当前值 (高位字) *1	1-100,000,000 次脉冲 *2		第 18-22 页上的“14. 当前值”
起始编号 +9	当前值 (低位字) *1			
起始编号 +10	错误状态	0-4		第 18-22 页上的“15. 错误状态”
起始编号 +11	保留			

\*1 高位和低位数据寄存器根据指定的 32 位数据存储方法而变化。

有关详情，请参见第 3-10 页上的“32 位数据存储”。

\*2 无论启用 / 禁用绝对位置指定模式，都将存储输出的脉冲数。有关绝对位置指定模式的详情，请参见第 18-69 页上的“ABS（绝对位置设置）”。

### 4. S2（源 2）：初始化输入

S2 指定初始化输入。

当初始化输入 S2 打开后，“设置”选项卡上 WindLDR “Ramp 脉冲输出”对话框中配置的初始值将存储到控制寄存器中。

可以指定外部输入或内部继电器。

当初始化输入打开后，初始值将在每次扫描时写入数据寄存器中。要对值仅初始化一次，请将初始化输入与 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令结合使用。

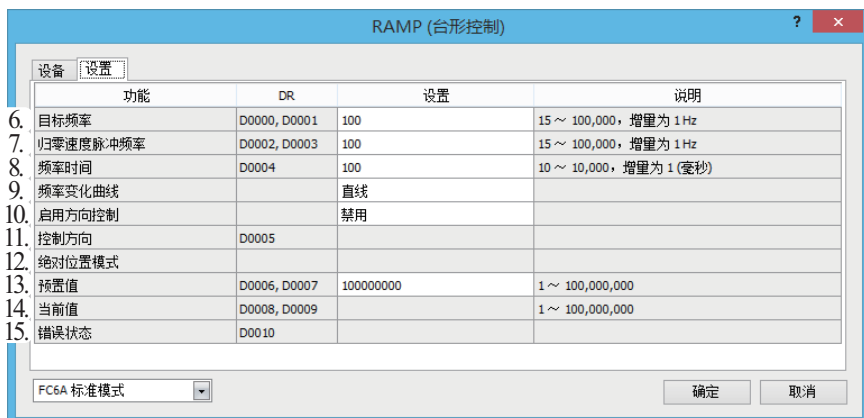
### 5. D1（目标 1）：运行状态

D1 指定用于 RAMP 指令的内部继电器的第一内部继电器。从指定的内部继电器开始，使用 4 个连续的内部继电器。请指定设备范围内的第一内部继电器。

存储目的地	功能	设置	
起始编号 +0	脉冲输出 打开	0: 脉冲输出关闭 1: 脉冲输出打开	此继电器在脉冲输出过程中打开。 此继电器在脉冲输出停止后关闭。 此继电器在指定次数的脉冲输出结束后关闭。
起始编号 +1	脉冲输出 完成	0: 脉冲输出未完成 1: 脉冲输出完成	此继电器在脉冲输出完成后打开。 此继电器在脉冲输出开始后关闭。
起始编号 +2	脉冲输出 状态	0: 目标脉冲输出 1: 更改输出脉冲频率	此继电器在脉冲输出状态稳定后关闭。 此继电器在脉冲输出变化时打开。
起始编号 +3	上溢出	0: 无 1: 出现上溢出	启用脉冲计数后，此继电器将在输出的脉冲超过配置的预置值时打开。在稳定输出过程中出现上溢出或脉冲频率变化时，脉冲输出将保持平稳。但是，在出现上溢出时脉冲计数将暂停。

## 18: 脉冲输出指令

### ■ “设置”选项卡



#### 6. 目标频率

此设置指定脉冲频率加大后的目标频率。输出频率误差限于  $\pm 5\%$ 。  
对应指令、频率因 CPU 模块的类型而异。

CPU 模块的类型	指令	可配置范围	
		设置值	频率
All-in-One CPU 模块	RAMP1、 RAMP2	15-100,000	15Hz-100kHz (增量为 1Hz)
	RAMP3、 RAMP4	15-50,000	15Hz-5kHz (增量为 1Hz)
CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	RAMP1-RAMP4	15-100,000	15Hz-100kHz (增量为 1Hz)

#### 7. 归零速度脉冲频率

此设置指定脉冲输出开始的频率。输出频率误差限于  $\pm 5\%$ 。  
对应指令、频率因 CPU 模块的类型而异。

CPU 模块的类型	指令	可配置范围	
		设置值	频率
All-in-One CPU 模块	RAMP1、 RAMP2	15-100,000	15Hz-100kHz (增量为 1Hz)
	RAMP3、 RAMP4	15-50,000	15Hz-5kHz (增量为 1Hz)
CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	RAMP1-RAMP4	15-100,000	15Hz-100kHz (增量为 1Hz)

#### 8. 频率时间

此设置指定脉冲频率加大和减小的时间。

将该时间设置为 10 到 10,000ms (增量为 1ms) 之间。预置值的个位作为 0 处理。例如, 输入 144 时, 预置值作为 140ms 处理。

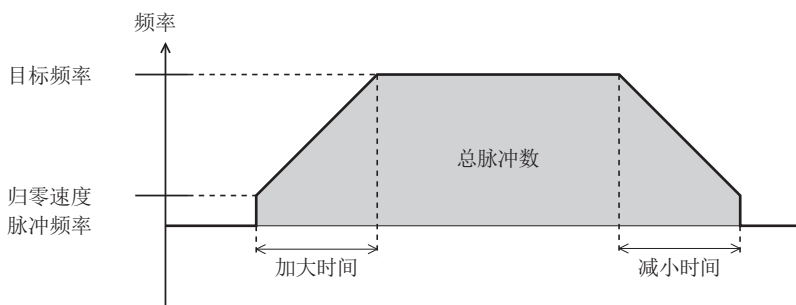
#### 9. 频率变化曲线

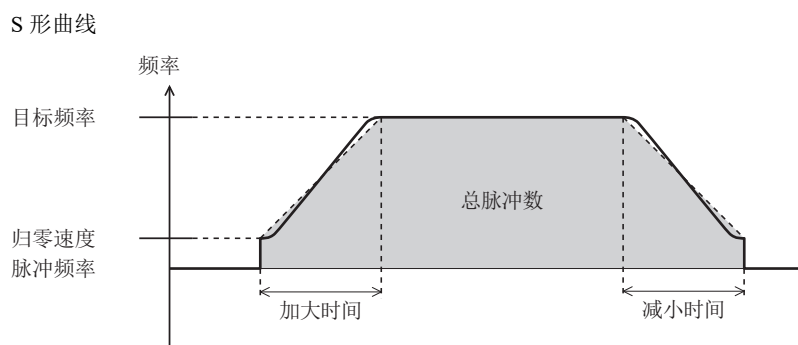
将频率变化曲线选择为“直线”或“S 形曲线”。

如果选择了“S 形曲线”, 由于会减少初期加速, 相较于“直线”, 可抑制冲击或振动。

仅支持 Plus CPU 模块的晶体管输出类型。

直线



**注释:**

- S 形频率变化曲线会根据设置值变为 3 次函数中近似的曲线。近似公式的预置值无法更改。
- 预置值在以下情况下，频率变化曲线会变为直线。
  - 频率更改时间不足 100ms 时
  - 归零速度脉冲频率不足 100Hz 时

**10. 启用方向控制**

此设置启用或禁用方向控制，并从以下方向控制模式中选择方向控制方法。脉冲输出模式分为两种：单脉冲和双脉冲。它们可以与方向控制结合使用，如下所示。（以 All-in-One CPU 模块使用 RAMP1 时为例。）

启用方向控制	动作	模式
禁用	当单向使用脉冲输出时选择此选项。脉冲 A 和 B 可独立使用。	Q0
方向控制“单脉冲输出”	脉冲 A 用作脉冲输出。脉冲 B 打开 / 关闭用作方向控制。	Q0 Q2
方向控制“双脉冲输出”	脉冲 A 用作正向脉冲 (CW) 输出。 脉冲 B 用作反向脉冲 (CCW) 输出。	Q0 Q1

FC6A 型上使用的输出因使用的指令、脉冲输出模式和方向控制结合以及使用的型号而异。

指令	操作条件	使用的输出			
		All-in-One CPU 模块		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	
		脉冲输出	方向控制输出	脉冲输出	方向控制输出
RAMP1	禁用方向控制	Q0	—	Q0	—
	方向控制（单脉冲输出）	Q0	Q2 <sup>*1</sup>	Q0	Q1
	方向控制（双脉冲输出）	Q0、Q1 <sup>*2</sup>	—	Q0、Q1	—
RAMP2	禁用方向控制	Q1	—	Q2	—
	方向控制（单脉冲输出）	Q1	Q3 <sup>*1</sup>	Q2	Q3
	方向控制（双脉冲输出）	—	—	Q2、Q3	—
RAMP3	禁用方向控制	Q2	—	Q4	—
	方向控制（单脉冲输出）	—	—	Q4	Q5
	方向控制（双脉冲输出）	—	—	Q4、Q5	—
RAMP4	禁用方向控制	Q3	—	Q6	—
	方向控制（单脉冲输出）	—	—	Q6	Q7
	方向控制（双脉冲输出）	—	—	Q6、Q7	—

\*1 使用 All-in-One CPU 模块、单脉冲输出模式时，因使用 Q2 或 Q3，不可同时执行使用同一输出的指令。

\*2 使用 All-in-One CPU 模块、双脉冲输出模式时，因使用 Q1，不可同时执行使用同一输出的指令。

**11. 控制方向**

启用方向控制后，系统将在用于正向操作的数据寄存器中存储 0，在用于反向操作的数据寄存器中存储 1。

“启用”绝对位置指定模式时可忽略。用目标位置减去绝对位置计数器之值，若为正数则自动选择“正向”，若为负数则自动选择“反向”，输出脉冲。

## 18: 脉冲输出指令

### 12. 绝对位置模式

以脉冲数（13.）指定目标位置时，将通过存储在绝对位置计数器（D8240 ~ D8247）的当前位置与目标位置之差自动计算脉冲数及方向，输出脉冲。

绝对位置指定	概要
禁用	指定“控制方向”、“预置值”执行 RAMP 指令。 输出“预置值”中指定的脉冲数。
启用	对“预置值”指定作为目标的绝对位置（目标位置），执行 RAMP 指令。 通过绝对位置计数器管理的绝对位置和“预置值”指定的目标位置，计算并执行输出的脉冲数和方向。 禁用“控制方向”配置。

#### 注释:

- 在方向控制模式下选择“禁用”时，绝对位置指定模式无效。
- D8239（绝对位置管理状态）对应的绝对位置计数器初始化已完成标记为 0（未初始化）时，即使对绝对位置指定模式执行“启用”指令，也将导致用户程序执行错误。梯形图程序运行开始后，请执行 1 次目标输出对应的 ABS 指令。

### 13. 预置值

绝对位置指定模式为“禁用”时，总输出脉冲数配置在 1 ~ 100,000,000 之间。

绝对位置指定模式为“启用”时，在 -2,147,483,648 ~ 2,147,483,647 范围内指定目标位置。输出脉冲数为用目标位置减去绝对位置计数器之值的绝对值。

### 14. 当前值

无论启用 / 禁用绝对位置指定模式，脉冲输出数都将存储在数据寄存器中。

执行 RAMP 指令时，每次扫描都会更新当前值。

### 15. 错误状态

如果在 RAMP 指令输入从关闭变为打开时出现配置错误，M8004（用户程序执行错误）将打开并且保存该错误代码。

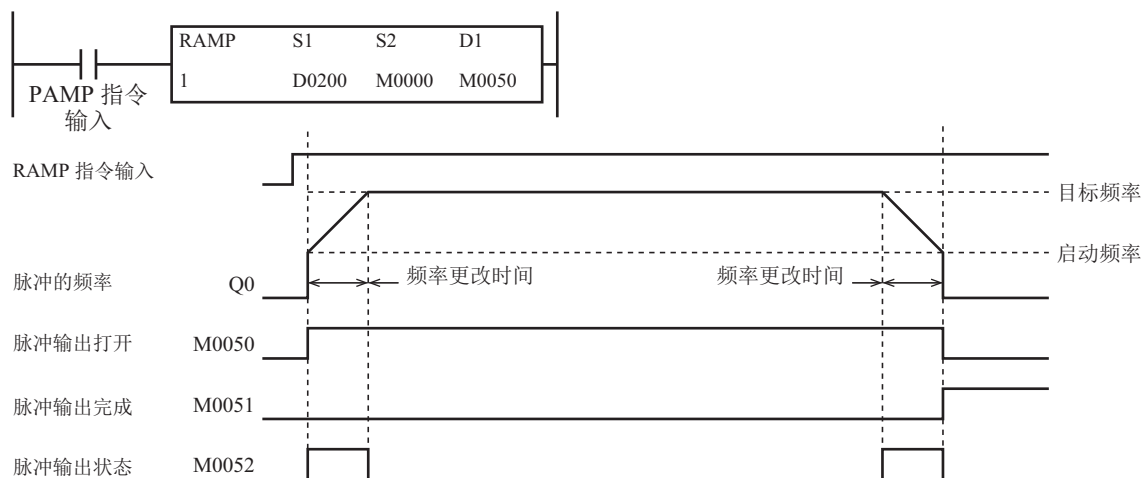
错误代码	说明	详情		
0	正常	—		
2	启动频率指定错误	All-in-One CPU 模块	RAMP1、RAMP2	脉冲频率设置在 15 到 100,000 之间。
			RAMP3、RAMP4	脉冲频率未设置在 15 到 5,000 之间。
		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	RAMP1-RAMP4	脉冲频率未设置在 15 到 100,000 之间。
3	预置值指定错误	禁用绝对位置指定模式时：预置值未设置在 1 到 100,000,000 之间。 启用绝对位置指定模式时：预置值未设置在 -2,147,483,648 到 2,147,483,647 之间。或者所设置的脉冲数与绝对位置计数器之差大于 100,000,000 脉冲。		
4	目标频率指定错误	All-in-One CPU 模块	RAMP1、RAMP2	脉冲频率未设置在 15 到 100,000 之间。
			RAMP3、RAMP4	脉冲频率未设置在 15 到 5,000 之间。
		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	RAMP1-RAMP4	脉冲频率未设置在 15 到 100,000 之间。
5	频率更改时间指定错误	频率更改时间未设置在 10 到 10,000 之间。		
7	控制方向指定错误	控制方向未设置为 0 或 1。		
8	超出脉冲计数错误	频率变化脉冲数超出输出脉冲的总数*1。		
9	启动频率设置为与目标频率相同，或设置为大于目标频率的值*2。			

\*1 是指由启动频率、目标频率和频率更改时间而算出的频率变化区域的脉冲数超出输出脉冲的总数。此时，可以减小目标频率或启动频率或者缩短频率更改时间进行设置调整。

\*2 设置启动频率，使其低于目标频率。

## RAMP1 指令（禁用方向控制）时间图

RAMP1 指令，S1 指定为 D0200，D1 指定为内部继电器 M0050



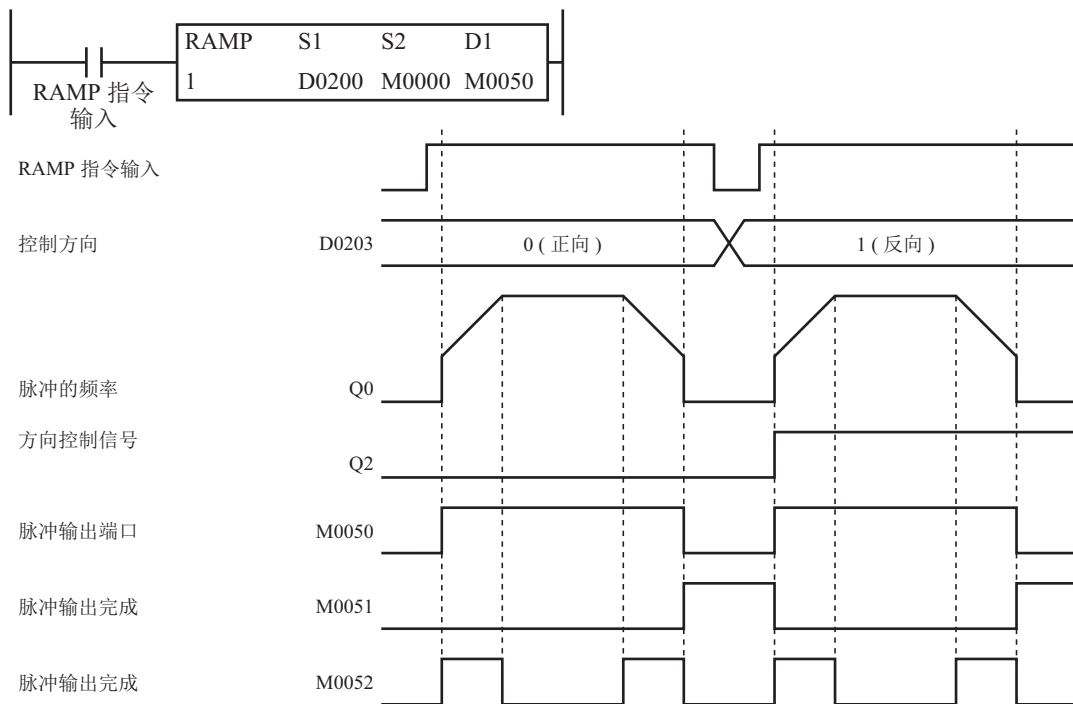
当 RAMP 指令输入打开后，系统将根据控制寄存器配置的设置输出脉冲。M0050 在脉冲输出过程中打开。M0052 在脉冲频率加大或减小时打开或关闭。脉冲根据频率变化时间输出，直到（从初始脉冲频率）达到稳定脉冲频率。将频率变化时间配置指定为 100 时，以 10ms 为单位进行频率变化，直至在 100ms 后达到目标频率。当输出的脉冲达到预置值配置的值后，脉冲输出停止。（频率变化的脉冲数也将计数。）此时，M0050 关闭，M0051 打开。

如果 RAMP 指令输入在脉冲输出过程中关闭，则脉冲输出将取消。如果再次打开 RAMP 指令，则脉冲计数将复位并开始脉冲计数。即使控制寄存器的内容在脉冲输出过程中发生变化，脉冲输出操作中也不会反映这种变化。变化的内容会在下次执行 RAMP 指令时反映出来。

## 18: 脉冲输出指令

### RAMP1 指令（启用方向控制、单脉冲输出模式）时间图

RAMP1 指令，S1 指定为 D0200，D1 指定为内部继电器 M0050



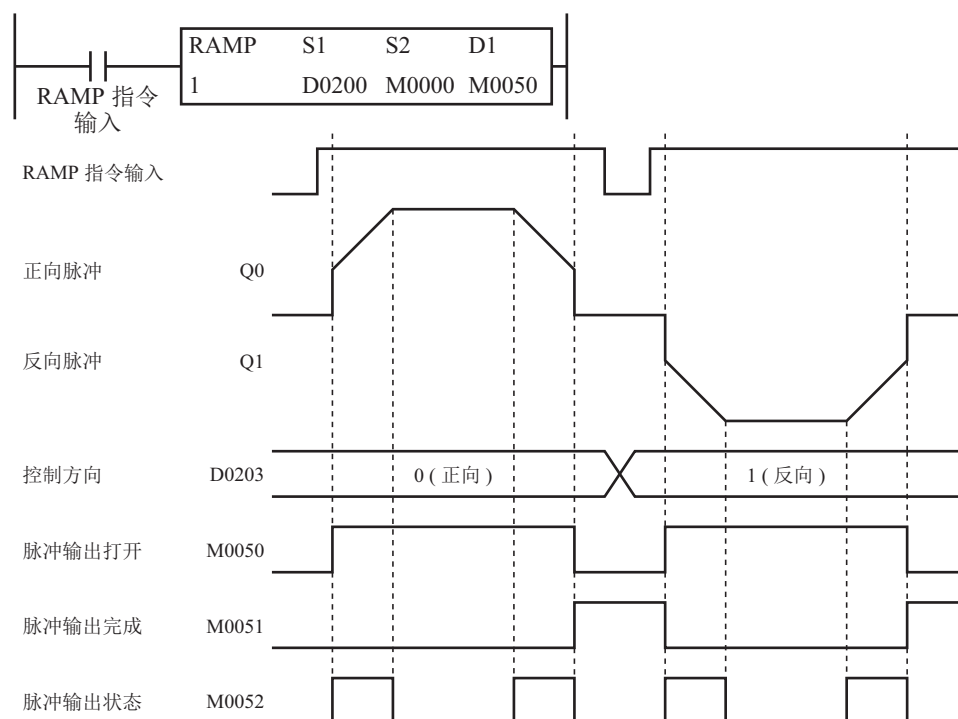
当 RAMP 指令输入打开后，系统将根据控制寄存器配置的设置输出脉冲。方向控制信号从 Q2 中输出。脉冲输出开始后，M0050 将打开。当脉冲频率加大或减小时，M0052 打开。以在频率变化时间内从启动频率达到目标频率的方式输出脉冲。将频率变化时间配置指定为 100 时，以 10ms 为单位进行频率变化，直至在 100ms 后达到目标频率。

输出脉冲数中预置的脉冲时，脉冲输出停止。（频率变化的脉冲数也将计数。）此时，M0050 关闭，M0051 打开。

如果 RAMP 指令输入在脉冲输出过程中关闭，则脉冲输出将取消。如果此输入再次打开，则操作将从头开始。即使数据寄存器的内容在脉冲输出过程中发生变化，脉冲输出操作中也不会反映这种变化。变化的内容会在下次启动 RAMP 指令时反映出来。

## RAMP1 指令（启用方向控制、双脉冲输出模式）时间图

RAMP1 指令，S1 指定为 D0200，D1 指定为内部继电器 M0050



当 RAMP 指令输入从关闭变为打开时，脉冲将根据控制寄存器配置的设置从 Q0 或 Q1 中输出。

脉冲输出开始后，M0050 打开。在加大或减小过程中，M0052 打开。

以在频率变化时间内从启动频率达到目标频率的方式输出脉冲。将频率变化时间配置指定为 100 时，以 10ms 为单位进行频率变化，直至在 100ms 后达到目标频率。

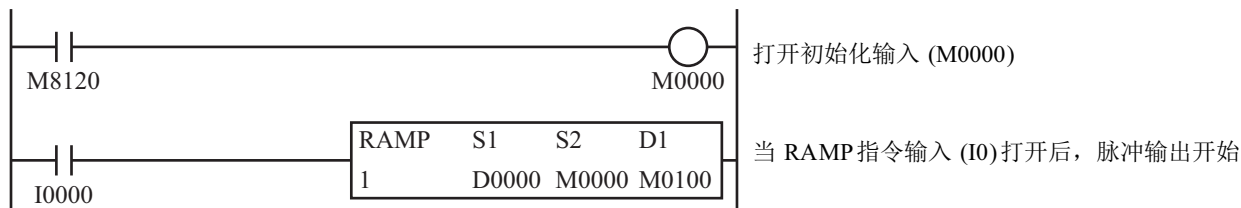
输出脉冲数中预置的脉冲时，脉冲输出停止。（频率变化的脉冲数也将计数。）此时，M0050 关闭，M0051 打开。

如果 RAMP 指令输入在脉冲输出过程中关闭，则脉冲输出将取消。如果此输入再次打开，则操作将从头开始。即使数据寄存器的内容在脉冲输出过程中发生变化，脉冲输出操作中也不会反映这种变化。变化的内容会在下次启动 RAMP 指令时反映出来。

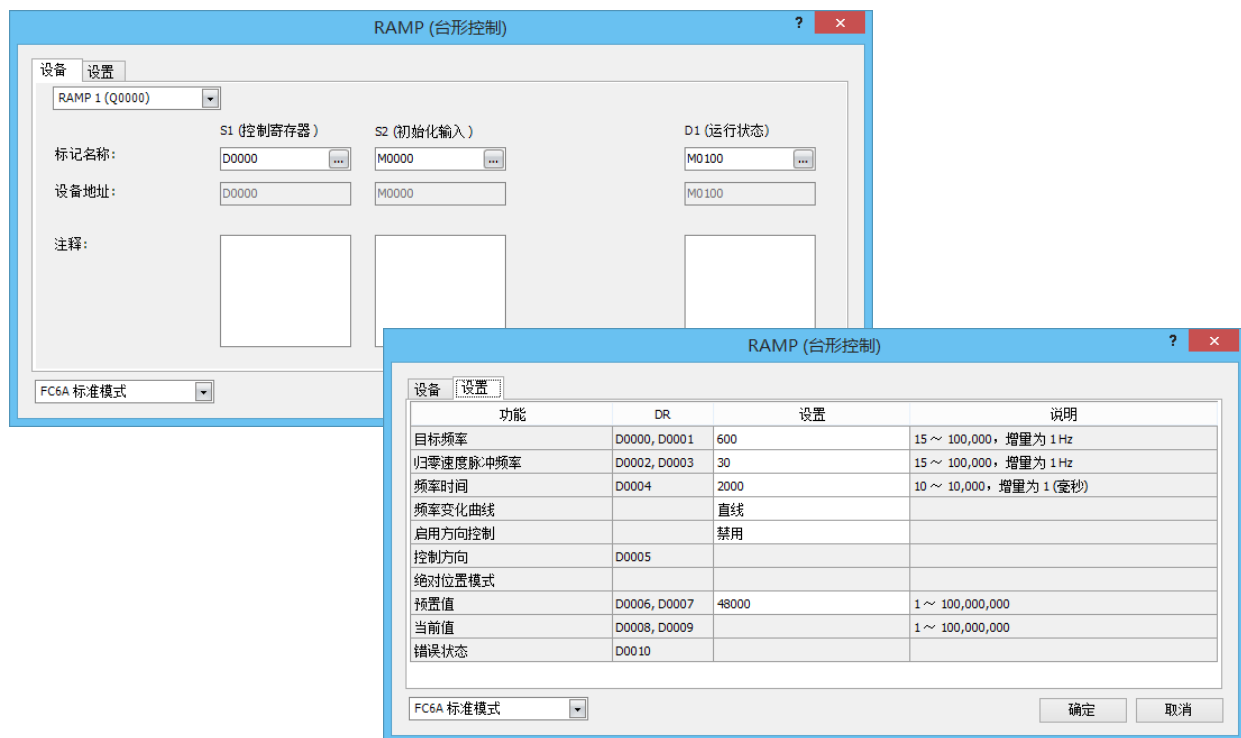
## 18: 脉冲输出指令

示例：

要利用频率变化功能（禁用方向控制）从 Q0 输出 48,000 次脉冲



设置

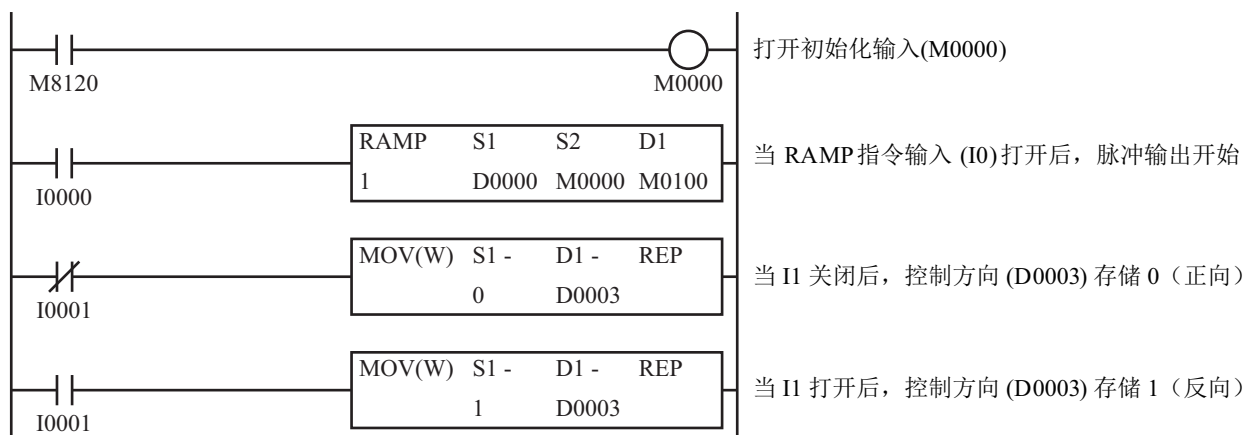


功能	设备地址	设置值	详情
目标频率	D0000、 D0001	600	600Hz
归零速度脉冲频率	D0002、 D0003	30	30Hz
频率时间	D0004	2000	2,000ms
启用方向控制	—	禁用	—
频率变化曲线	—	直线	—
控制方向	D0005	—	—
绝对位置模式	—	—	—
预置值	D0006、 D0007	48000	预置值 =48,000

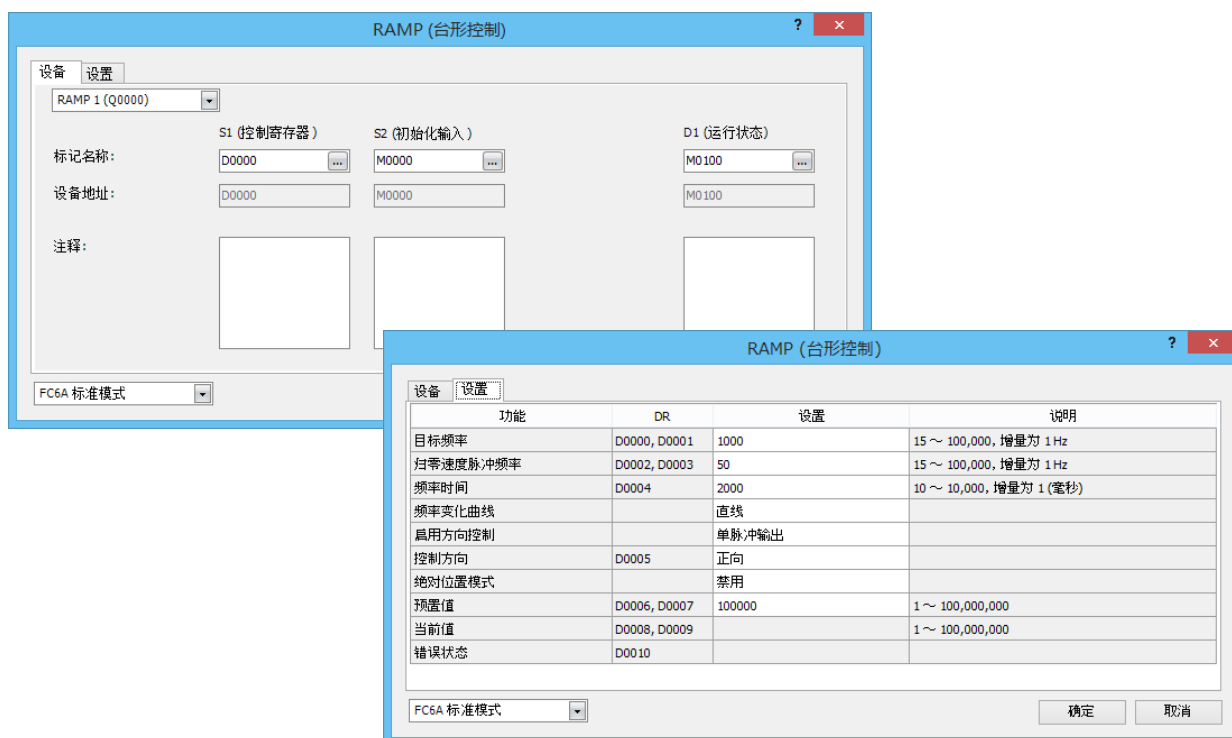


要利用频率变化功能（方向控制为单脉冲输出）从 Q0 输出 100,000 次脉冲

当 RAMP 指令输入 (I0) 从关闭变为打开时，脉冲输出开始。当 I1 关闭时，方向控制信号 (Q2) 关闭（正向）。当 I1 打开时，方向控制信号 (Q2) 打开（反向）。



## 设置



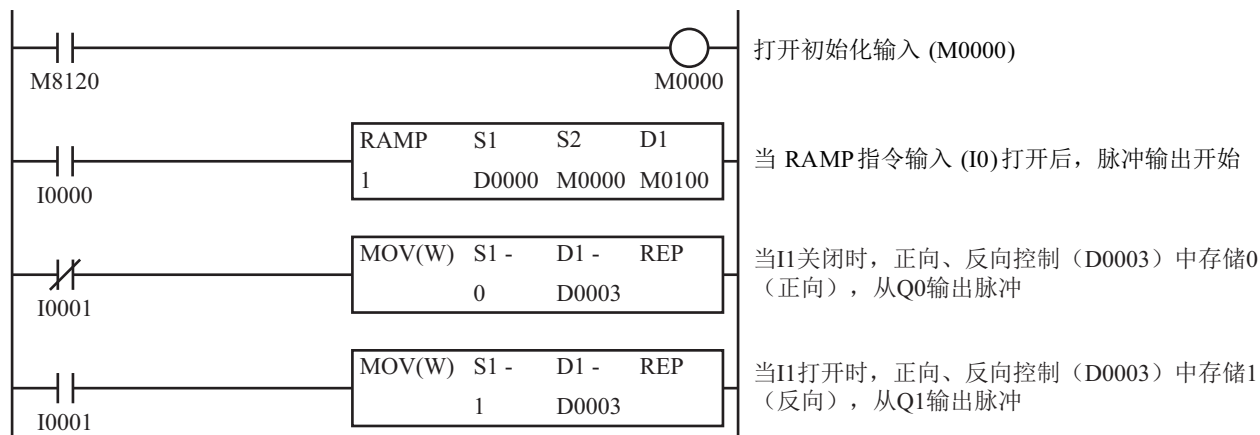
功能	设备地址	设置值	详情
目标频率	D0000、 D0001	1000	1,000Hz
归零速度脉冲频率	D0002、 D0003	50	50Hz
频率时间	D0004	2000	2,000ms
频率变化曲线	—	直线	—
启用方向控制	—	单脉冲输出	—
控制方向	D0005	正向	正向 =0
绝对位置模式	—	禁用	—
预置值	D0006、 D0007	100000	预置值 =100,000

## 18: 脉冲输出指令

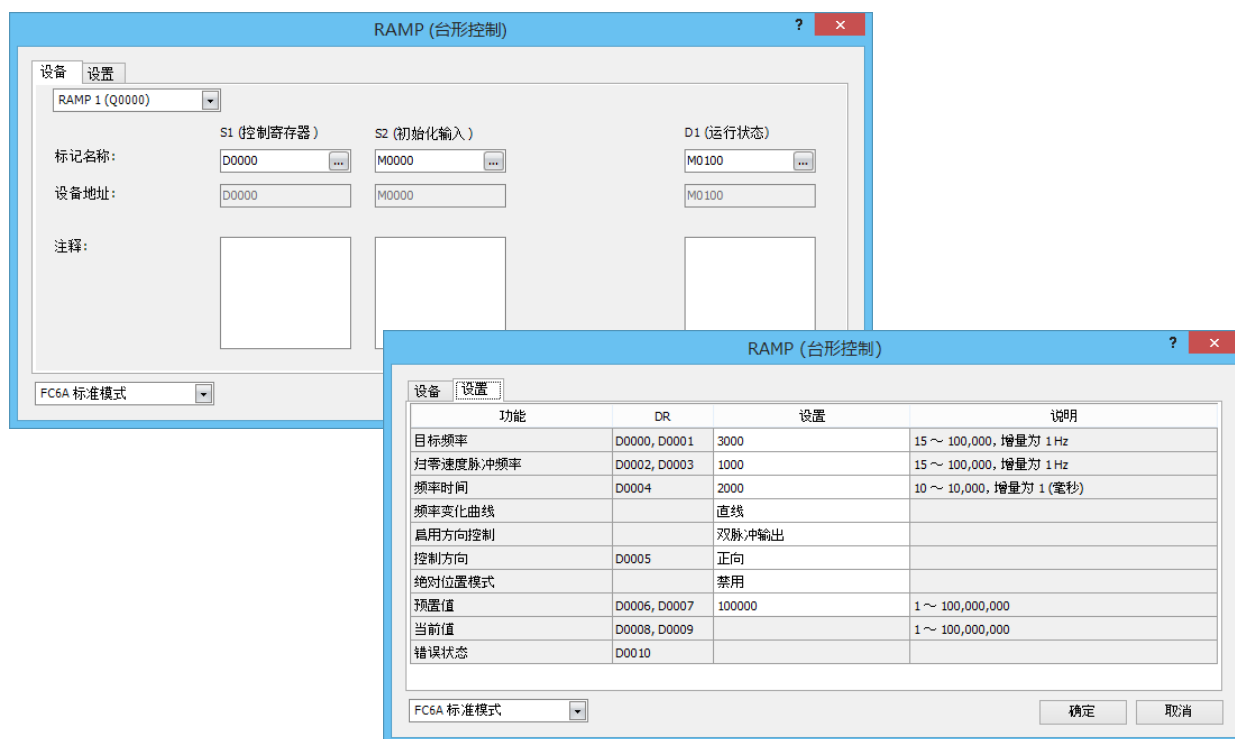
要利用频率变化功能（方向控制为双脉冲输出）输出 1,000,000 次脉冲

当 RAMP 指令输入 (I0) 从关闭变为打开时，脉冲输出开始。对于正向，在 I1 关闭时，脉冲 (CW) 将从 Q0 输出。

对于反向，在 I1 打开时，脉冲 (CCW) 将从 Q1 输出。



### 设置



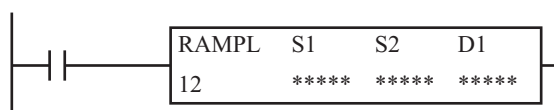
功能	设备地址	设置值	详情
目标频率	D0000、 D0001	3000	3,000Hz
归零速度脉冲频率	D0002、 D0003	1000	1,000Hz
频率时间	D0004	2000	2,000ms
频率变化曲线	—	直线	—
启用方向控制	—	双脉冲输出	—
控制方向	D0005	正向	正向 =0
绝对位置模式	—	禁用	—
预置值	D0006、 D0007	100000	预置值 =100,000

## RAMPL（直线插补控制）

利用频率变化功能，通过 2 个输出同时输出脉冲，以使移动轨迹呈直线。

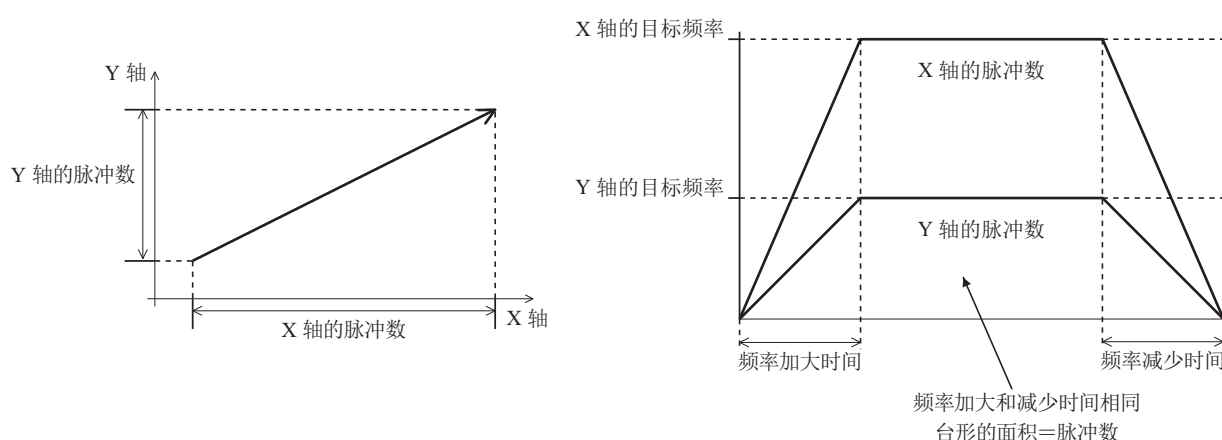
FT2J FC6A

※ 可使用 Plus CPU 模块晶体管输出类型及 CAN J1939 All-in-One CPU 模块晶体管输出类型



### 操作

输入为打开时，根据 S1 指定的脉冲数（目标位置）和合成启动频率以及合成目标频率，计算可使移动轨迹呈直线的各轴启动频率与目标频率。之后，将通过指定的 2 个输出同时输出，并以固定比率加大脉冲直至达到各轴的目标频率。以目标频率输出固定速度的脉冲后，减小脉冲直至达到 S1 指定的脉冲数，并在达到脉冲数时停止脉冲输出。



S2 中指定的初始化输入打开时，将 WindLDR“RAMPL（具直线插补控制的台形脉冲输出）”对话框的“常规设置”选项卡中配置的初始值存储到控制寄存器中。

脉冲输出状态（正在输出 / 输出方向 / 输出完成）等控制状态将存储到 D1 指定的运行状态中。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- RAMPL 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 RAMPL 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。
- RAMPL 指令仅在伴随方向控制的绝对位置模式下有效。梯形图程序运行开始后，请执行指定输出对应的 ABS 指令，并初始化绝对位置计数器。D8239（绝对位置管理状态）对应的绝对位置计数器初始化已完成标记为 0（未初始化）时，执行指令时将导致用户程序执行错误。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2（源 2）	初始化输入	X	—	X	—	—	—	—	—	—	—
D1（目标 1）	运行状态	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 18: 脉冲输出指令

### 设置

#### ■ “设备”选项卡



#### (1) 选择指令

可从“RAMPL12”、“RAMPL13”、“RAMPL14”、“RAMPL23”、“RAMPL24”、“RAMPL34”中选择使用的RAMPL指令。

脉冲输出的组合因指令而异。

有关详情，请参见第 18-33 页上的“(11) 方向控制”。

#### (2) S1（源 1）：控制寄存器

S1 指定用于 RAMPL12、RAMPL13、RAMPL14、RAMPL23、RAMPL24、RAMPL34 指令的数据寄存器的第一数据寄存器。

从第一数据寄存器开始，使用连续的 30 位数据寄存器。请指定设备范围内的第一数据寄存器。

存储目的地	功能	设置内容	参考	
起始编号 +0	合成目标频率（高位字）*1	15 ~ 100,000（增量为 1Hz）	第 18-32 页上的“(5) 合成目标频率”	
起始编号 +1	合成目标频率（低位字）*1			
起始编号 +2	合成启动频率（高位字）*1	15 ~ 100,000（增量为 1Hz）	第 18-32 页上的“(6) 合成启动频率”	
起始编号 +3	合成启动频率（低位字）*1			
起始编号 +4	频率更改时间	10 ~ 10,000 (ms)	第 18-32 页上的“(7) 频率更改时间”	
起始编号 +5		— 保留 —		
起始编号 +6		— 保留 —		
起始编号 +7		— 保留 —		
起始编号 +8		— 保留 —		
起始编号 +9	错误状态		第 18-32 页上的“(8) 错误状态”	
起始编号 +10	X 轴	目标频率（高位字）*1*2	15 ~ 100,000（增量为 1Hz）	第 18-33 页上的“(9) 目标频率”
起始编号 +11		目标频率（低位字）*1*2		
起始编号 +12	X 轴	启动频率（高位字）*1*2	15 ~ 100,000（增量为 1Hz）	第 18-33 页上的“(10) 归零速度脉冲频率”
起始编号 +13		启动频率（低位字）*1*2		
起始编号 +14		— 保留 —		
起始编号 +15		— 保留 —		
起始编号 +16	X 轴	脉冲数（高位字）*1	绝对位置模式 -2147483648 ~ 2147483647（脉冲）	第 18-34 页上的“(13) 脉冲数”
起始编号 +17		脉冲数（低位字）*1		
起始编号 +18	X 轴	当前值（高位字）*1	1 ~ 100,000,000（脉冲）*3	第 18-34 页上的“(14) 当前值”
起始编号 +19		当前值（低位字）*1		

存储目的地	功能	设置内容	参考	
起始编号 +20	X 轴	目标频率（高位字）*1*2	15 ~ 100,000（增量为 1Hz）	
起始编号 +21		目标频率（低位字）*1*2		
起始编号 +22		启动频率（高位字）*1*2	15 ~ 100,000（增量为 1Hz）	
起始编号 +23		启动频率（低位字）*1*2		
起始编号 +24		— 保留 —		
起始编号 +25		— 保留 —		
起始编号 +26		脉冲数（高位字）*1	绝对位置模式	第 18-34 页上的“(13) 脉冲数”
起始编号 +27		脉冲数（低位字）*1	-2147483648 ~ 2147483647（脉冲）	
起始编号 +28	当前值（高位字）*1	1 ~ 100,000,000（脉冲）*3	第 18-34 页上的“(14) 当前值”	
起始编号 +29	当前值（低位字）*1			

\*1 高位和低位数据寄存器根据 32 位数据存储设置而变化。有关详情，请参见第 3-10 页上的“32 位数据存储”。

\*2 指令的输入打开时，将存储自动计算的值。

\*3 无论绝对位置计数器的值为何，都将存储输出的脉冲数。

### (3) S2（源 2）：初始化输入

S2 指定初始化输入。

初始化输入打开时，将 WindLDR“RAMPL（具直线插补控制的台形脉冲输出）”对话框的“常规设置”选项卡中配置的初始值存储到控制寄存器中。可指定外部输入或内部继电器。

初始化输入为打开时，每次扫描都会将初始值存储到数据寄存器中。（即使在不执行 RAMPL 指令（未打开）的状态下，若打开初始化输入，初始值也将存储在数据寄存器中。）要对值仅初始化一次，请将初始化输入与 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令结合使用。

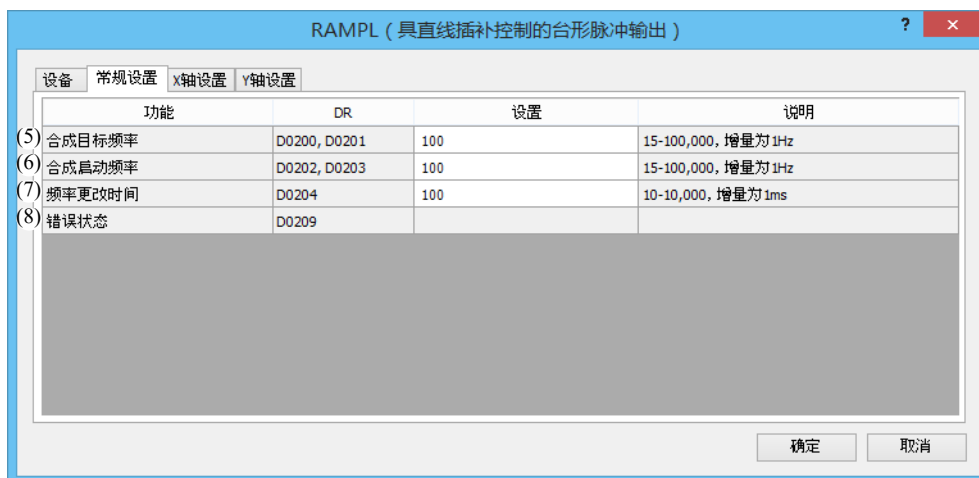
### (4) D1（目标 1）：运行状态

D1 指定用于 RAMPL 指令的内部继电器的第一内部继电器。以指定的内部继电器为起始，使用 4 个连续的内部继电器。请指定设备范围内的第一内部继电器。

存储目的地	功能	设置内容	
起始编号 +0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	此继电器在脉冲输出期间打开。 脉冲输出停止后关闭。 输出完指定的脉冲数后关闭。
起始编号 +1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	脉冲输出完成后打开。 脉冲输出开始后关闭。
起始编号 +2	脉冲输出状态	0: 恒速状态 1: 频率变化状态	脉冲输出状态为恒速状态时关闭。 脉冲输出状态为频率变化状态时打开。
起始编号 +3	上溢出	0: 无 1: 发生上溢出	此继电器在启用脉冲计数、超过设置脉冲数输出脉冲时打开。 加速或恒速动作时，即使发生上溢出也会继续输出脉冲。 但是，当前值会在发生上溢出时中断计数。

## 18: 脉冲输出指令

### ■ “常规设置”选项卡



#### (5) 合成目标频率

指定变化后恒速状态的频率。输出频率误差限于  $\pm 5\%$ 。

根据合成目标频率计算出的 X 轴、Y 轴各自的目标频率小于合成目标频率。X 轴、Y 轴各自的目标频率请设置为大于 15Hz。

#### (6) 合成启动频率

指定开始输出脉冲时的频率。输出频率误差限于  $\pm 5\%$ 。

根据合成启动频率计算出的 X 轴、Y 轴各自的启动频率小于合成启动频率。X 轴、Y 轴各自的启动频率请设置为大于 15Hz。

#### (7) 频率更改时间

指定加大或减小脉冲的时间。设于 10 ~ 10,000ms（增量为 1ms）之间。预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。

#### (8) 错误状态

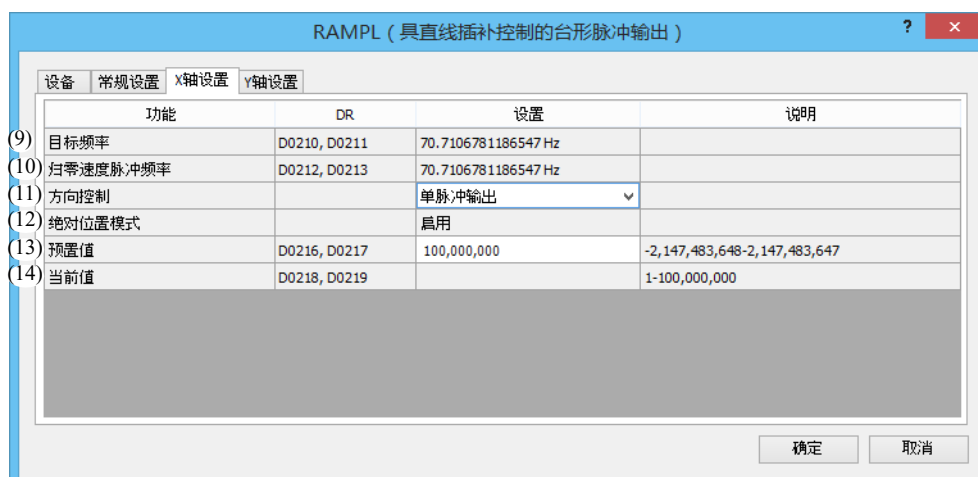
若配置的设置有误，则输出有误设置对应的错误代码。如果在 RAMPL 指令输入从关闭变为打开时出现配置错误，M8004（用户程序执行错误）将打开并且存储错误代码。

错误代码	内容	说明
0	正常	—
2	合成启动频率指定错误	合成启动频率未设置在 15 ~ 100,000 之间。
4	合成目标频率指定错误	合成目标频率未设置在 15 ~ 100,000 之间。
5	频率更改时间指定错误	频率更改时间未设置在 10 ~ 10,000 之间。
9	频率指定错误	将合成启动频率设为与合成目标频率相同或大于合成目标频率的值。 <sup>*1</sup>
12	X 轴启动频率指定错误	根据预置值计算出的 X 轴启动频率未在 15 ~ 100,000 之间。
13	X 轴脉冲数指定错误	X 轴脉冲数未设置在 -2,147,483,648 ~ 2,147,483,647 之间。或者所设置的脉冲数与绝对位置计数器之差大于 100,000,000 脉冲。
14	X 轴目标频率指定错误	根据预置值计算出的 X 轴目标频率未在 15 ~ 100,000 之间。
18	X 轴脉冲数超过错误	X 轴的频率变化脉冲数超出总输出脉冲数。 <sup>*2</sup>
22	Y 轴启动频率指定错误	根据预置值计算出的 Y 轴启动频率未在 15 ~ 100,000 之间。
23	Y 轴脉冲数指定错误	Y 轴脉冲数未设置在 -2,147,483,648 ~ 2,147,483,647 之间。或者所设置的脉冲数与绝对位置计数器之差大于 100,000,000 脉冲。
24	Y 轴目标频率指定错误	根据预置值计算出的 Y 轴目标频率未在 15 ~ 100,000 之间。
28	Y 轴脉冲数超过错误	Y 轴的频率变化脉冲数超出总输出脉冲数。 <sup>*2</sup>

\*1 请将合成启动频率设为低于合成目标频率。

\*2 根据目标频率、启动频率及频率更改时间计算出的频率变化范围的脉冲数超出总输出脉冲数。请减小目标频率、启动频率，或缩短频率更改时间。

## ■ “X 轴设置” 选项卡、“Y 轴设置” 选项卡



## (9) 目标频率

将根据向目标位置的移动量与合成目标频率计算出的各轴目标频率存储到数据寄存器中。指令打开时，计算并更新目标频率。“设置”中将显示各轴分解合成目标频率后的值。

## (10) 归零速度脉冲频率

将根据向目标位置的移动量与合成启动频率计算出的各轴归零速度脉冲频率存储到数据寄存器中。指令打开时，计算并更新归零速度脉冲频率。“设置”中将显示各轴分解合成启动频率后的值。

## (11) 方向控制

从以下方向控制模式中选择 X 轴、Y 轴各自的方向控制方法。脉冲输出模式包括单脉冲输出模式和双脉冲输出模式。无法选择禁用方向控制。

方向控制模式	动作	模式
方向控制“单脉冲输出”	脉冲 A 用作脉冲输出。脉冲 B 打开 / 关闭用作方向控制。	
方向控制“双脉冲输出”	脉冲 A 用作正向脉冲 (CW) 输出。脉冲 B 用作反向脉冲 (CCW) 输出。	

Plus CPU 模块上使用的输出因使用的指令而异。

## 18: 脉冲输出指令

指令	操作条件	使用的输出			
		X 轴		Y 轴	
		脉冲输出	方向控制输出	脉冲输出	方向控制输出
RAMPL12	方向控制（单脉冲输出模式）	Q0	Q1	Q2	Q3
	方向控制（双脉冲输出模式）	Q0、Q1	—	Q2、Q3	—
RAMPL13	方向控制（单脉冲输出模式）	Q0	Q1	Q4	Q5
	方向控制（双脉冲输出模式）	Q0、Q1	—	Q4、Q5	—
RAMPL14	方向控制（单脉冲输出模式）	Q0	Q1	Q6	Q7
	方向控制（双脉冲输出模式）	Q0、Q1	—	Q6、Q7	—
RAMPL23	方向控制（单脉冲输出模式）	Q2	Q3	Q4	Q5
	方向控制（双脉冲输出模式）	Q2、Q3	—	Q4、Q5	—
RAMPL24	方向控制（单脉冲输出模式）	Q2	Q3	Q6	Q7
	方向控制（双脉冲输出模式）	Q2、Q3	—	Q6、Q7	—
RAMPL34	方向控制（单脉冲输出模式）	Q4	Q5	Q6	Q7
	方向控制（双脉冲输出模式）	Q4、Q5	—	Q6、Q7	—

**注释：**用于 RAMPL 指令的输出（脉冲输出、方向控制输出）重复时，无法同时执行使用相同输出的指令。

### (12) 绝对位置模式

绝对位置模式始终为“启用”。请以 X 轴、Y 轴各自的脉冲数 (13) 指定目标位置。

**注释：**指定脉冲输出对应的 D8239（绝对位置管理状态）的绝对位置计数器初始化已完成标记为 0（未初始化）时，将导致用户程序执行错误。梯形图程序运行开始后，请执行 1 次目标输出对应的 ABS 指令。

### (13) 脉冲数

指定目标位置。输出脉冲数为用目标位置减去绝对位置计数器之值的绝对值。此时，将根据减算值的正负，自动切换正为正向、负为反向输出脉冲。

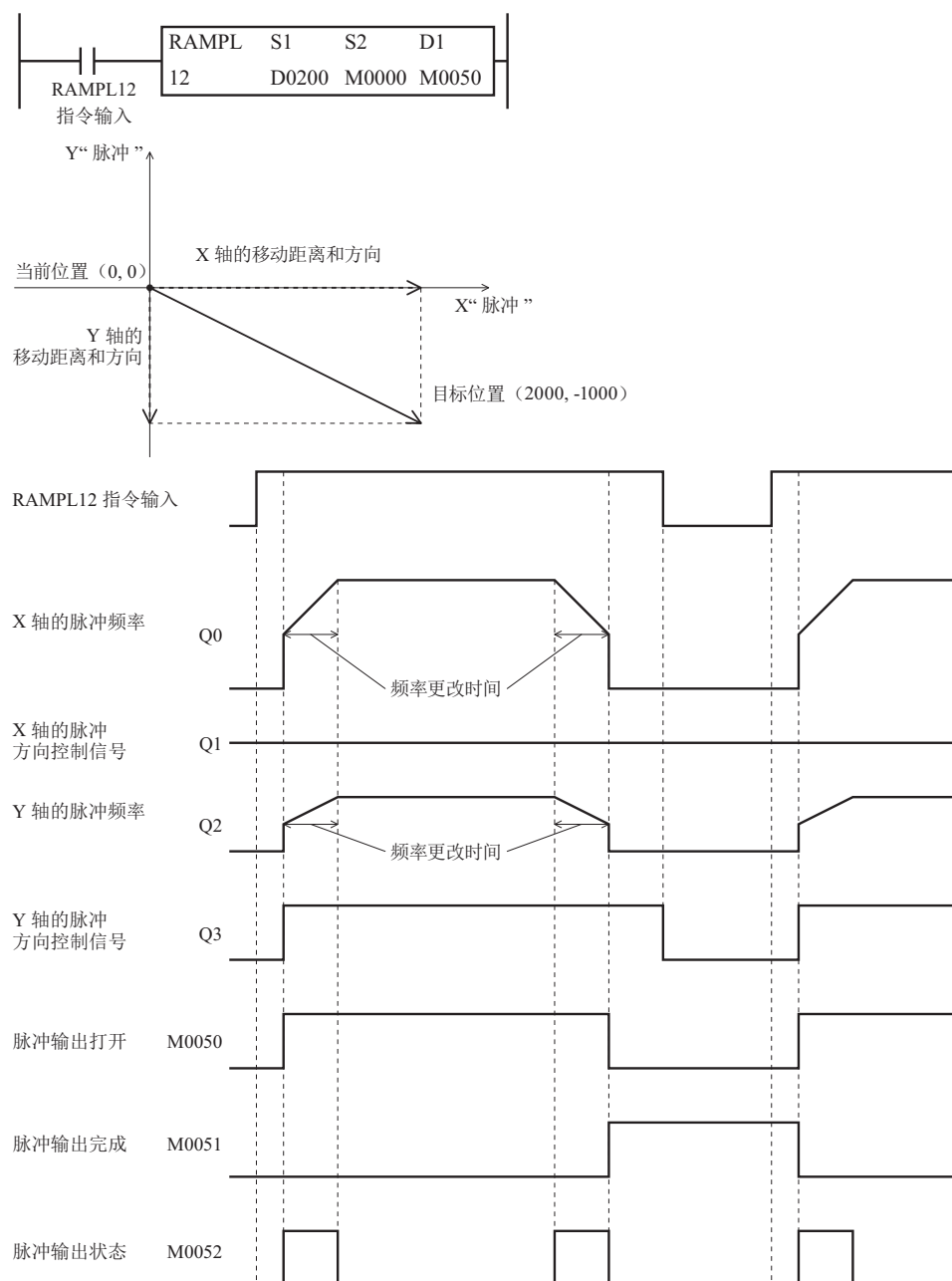
### (14) 当前值

脉冲输出数将存储在数据寄存器中。执行 RAMPL 指令时，每次扫描都会更新当前值。



## RAMPL12 指令（启用方向控制、单脉冲输出模式）时间图

将 RAMPL12 指令的 S1 指定为数据寄存器 D0200、D1 指定为内部继电器 M0050，以从当前位置（0,0）移动至目标位置（2000,-1000）时

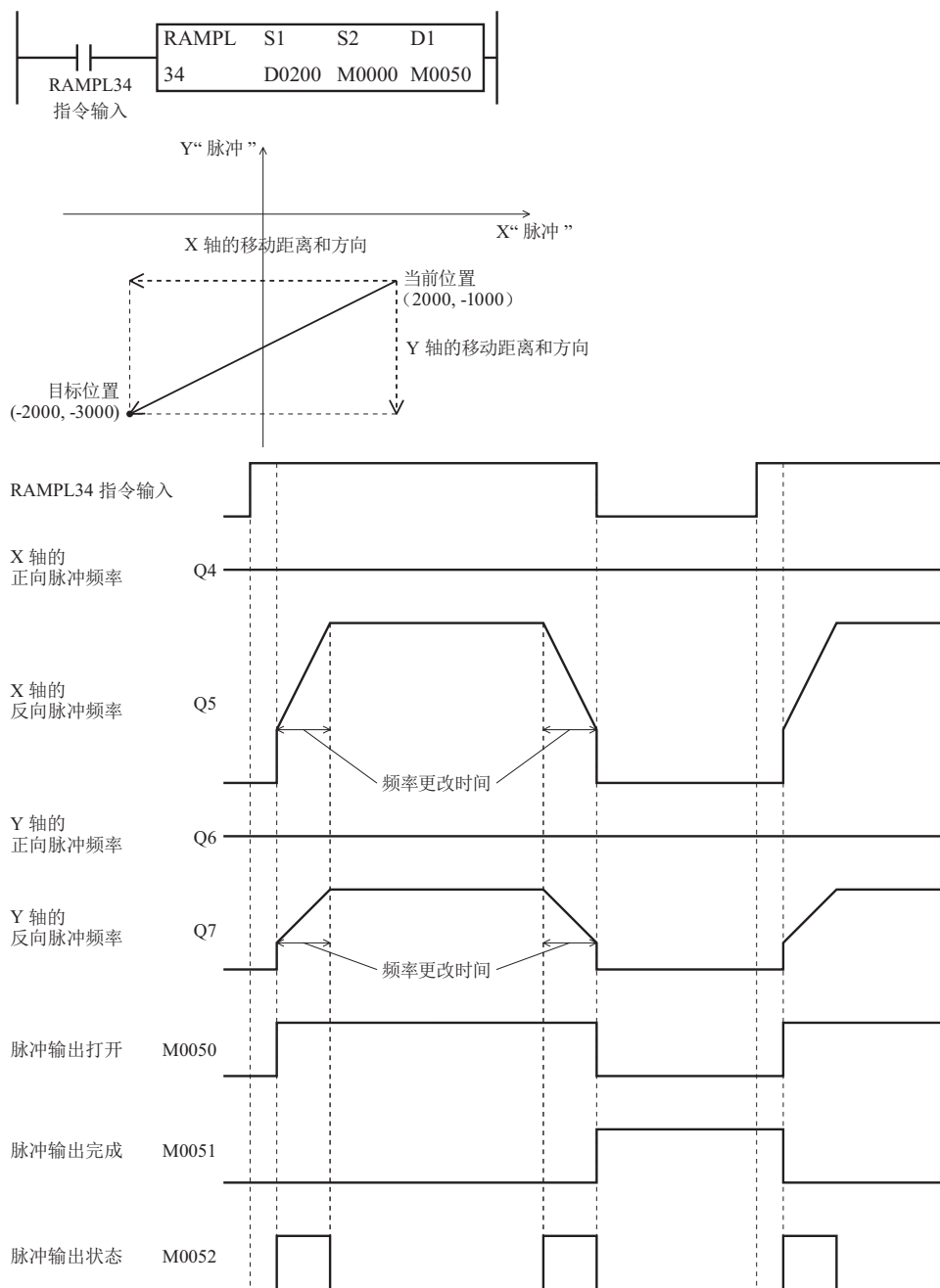


- RAMPL12 指令的输入打开时，将根据控制寄存器配置的设置从 Q0 或 Q2 中输出脉冲。由于 X 轴向正向，Q1 关闭。由于 Y 轴向负向，Q3 打开。
- 开始输出脉冲后，M0050 打开。在加大或减小过程中，M0052 打开。
- 根据各轴的脉冲数、合成目标频率、合成启动频率计算各轴的目标频率和启动频率，并在频率更改时间内从启动频率达到目标频率的方式输出各轴的脉冲。将频率变化时间设置指定为 100 时，以 10ms 为单位进行频率变化，直至在 100ms 后到达目标频率。
- 输出根据当前位置与目标位置之差计算出的脉冲后，脉冲输出停止。（频率变化的脉冲数也将计数。）此时，M0050 关闭，M0051 打开。
- 在脉冲输出过程中关闭 RAMPL12 指令输入后，脉冲输出中断。再次打开该输入，将从起始开始操作。
- 即使在脉冲输出中变更数据寄存器的设置，也不会反映在脉冲输出操作上。变更内容将在下一次启动 RAMPL12 指令时反映出来。

## 18: 脉冲输出指令

### RAMPL34 指令（启用方向控制、双脉冲输出模式）时间图

将 RAMPL34 指令的 S1 指定为数据寄存器 D0200、D1 指定为内部继电器 M0050，以从当前位置（2000, -1000）移动至目标位置（-2000, -3000）时



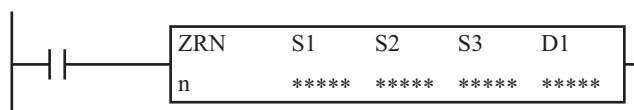
- RAMPL34 指令的输入打开时，将根据控制寄存器配置的设置输出脉冲。由于 X 轴、Y 轴各自向负方向前进，将输出从 Q5 至 X 轴、从 Q7 至 Y 轴的反向脉冲。
- 开始输出脉冲后，M0050 打开。在加大或减小过程中，M0052 打开。
- 根据各轴的脉冲数、合成目标频率、合成启动频率计算各轴的目标频率和启动频率，并在频率更改时间内从启动频率达到目标频率的方式输出各轴的脉冲。将频率变化时间设置指定为 100 时，以 10ms 为单位进行频率变化，直至在 100ms 后到达目标频率。
- 输出根据当前位置与目标位置之差计算出的脉冲后，脉冲输出停止。（频率变化的脉冲数也将计数。）此时，M0050 关闭，M0051 打开。
- 在脉冲输出过程中关闭 RAMPL34 指令输入后，脉冲输出中断。再次打开该输入，将从起始开始操作。
- 即使在脉冲输出中变更数据寄存器的设置，也不会反映在脉冲输出操作上。变更内容将在下一次启动 RAMPL34 指令时反映出来。

## ZRN (归零)

监视进行归零的多个信号的同时，输出脉冲。

FT2J FC6A※

※ 仅晶体管输出类型



当输入打开后，系统将根据存储在 S1 指定的控制寄存器中的频率变化配置，输出脉冲。

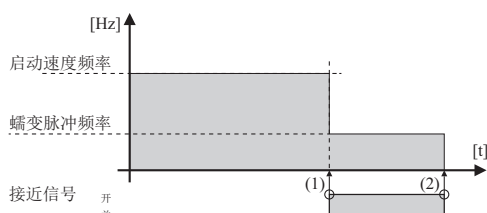
脉冲控制信息（正在输出 / 输出完成 / 错误）将其作为操作状态存储在 D2 指定的内部继电器中。

在 S2 中指定的初始化输入打开时，将 WindLDR“ZRN（归零）”对话框中配置的初始值存储到控制寄存器中。

可从“ZRN 模式 0”和“ZRN 模式 1”2 种中选择归零方法。

### ZRN 模式 0

仅监视在 S3 中指定的接近信号，进行归零。



(1) 当检测到接近信号上升时，将启动速度频率切换为蠕变速度频率。

(2) 当检测到接近信号下降时，脉冲输出停止。

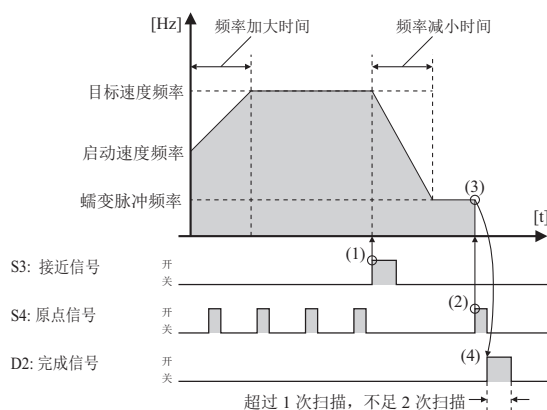
### ZRN 模式 1

监视在 S3 中指定的接近信号（减小时机）和 S4 中指定的原点信号（停止时机），进行归零。

通过原点信号停止脉冲输出后，可使用 D2 输出指定的完成信号。

ZRN 模式 1 可从以下 2 种中选择开始监视原点信号的时间。

接近信号上升时开始监视原点信号



(1) 检测到接近信号上升，开始减小，同时开始监视原点信号。

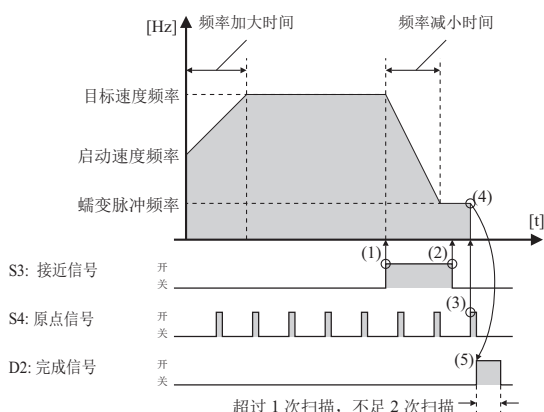
(2) 当检测到接近信号上升时，脉冲输出停止。

(3) 停止脉冲输出的同时，打开完成信号。

(4) 完成信号打开期间为超过 1 次扫描，不足 2 次扫描。

## 18: 脉冲输出指令

接近信号下降时开始监视原点信号



- (1) 当检测到接近信号上升时，开始减速
- (2) 当检测到接近信号下降时，开始监视原点信号。
- (3) 当检测到接近信号上升时，脉冲输出停止。
- (4) 停止脉冲输出的同时，打开完成信号。
- (5) 完成信号打开期间为超过 1 次扫描，不足 2 次扫描。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- ZRN 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 ZRN 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
S3 (源 3)	接近信号	X	—	X	—	—	—	—	—	—	—
S4 (源 4)	原点信号	X	—	X	—	—	—	—	—	—	—
D1 (目标 1)	操作状态	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—
D2 (目标 2)	完成信号	—	X	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1~7。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 设置

### ■ 设备选项卡



#### 1. 选择模式

选择配置模式。可选择“FC6A 标准模式”或“FC5A（FC5A-D12X1E 除外）兼容模式”。

希望在 FC5A 型 MICROSmart 的 ZRN 指令规格下使用时，选择“FC5A（FC5A-D12X1E 除外）兼容模式”。

希望变更 FC5A/FC4A 型 MICRO Smart 类型时，可自动选择“FC5A（FC5A-D12X1E 除外）兼容模式”。

另外，“FC5A（FC5A-D12X1E 除外）兼容模式”可使用的最低频率为 20Hz。

**注释：**FC5A（FC5A-D12X1E 除外）兼容模式可输出脉冲的最低频率为 20Hz。无法输出低于 20Hz 的脉冲。指定后将导致脉冲频率错误。

以下记录了选择“FC6A 标准模式”的情况。

**注释：**有关 FC5A（FC5A-D12X1E 除外）兼容模式下的配置详情，请参见《FC5A 系列 MICROSmart Pentra 用户手册 高级卷》第 13 章“脉冲指令”的 ZRN 指令。

#### 2. 选择指令

此项选择要使用的 ZRN 指令（“ZRN1”、“ZRN2”、“ZRN3”或“ZRN4”）。

输出、可配置的方向控制模式和频率因指令和 CPU 模块的类型而异。

有关指令、方向控制模式和脉冲输出模式的组合限制，请参见第 18-43 页上的“17. 启用方向控制”。

## 18: 脉冲输出指令

### 3. S1（源 1）：控制寄存器

S1 指定用于 ZRN1、ZRN2、ZRN3 或 ZRN4 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始，使用 14 个连续的数据寄存器。

请指定设备范围内的第一数据寄存器。

存储目的地	功能	设置		参考
		All-in-One CPU 模块	CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	
起始编号 +0	启动速度频率 (高位字) <sup>*1</sup>	ZRN1、ZRN2: 15-100,000 (增量为 1Hz)	ZRN1-ZRN4: 15-100,000 (增量为 1Hz)	第 18-42 页上的“10. 归零速度 脉冲频率”
起始编号 +1	启动速度频率 (低位字) <sup>*1</sup>	ZRN3、ZRN4: 15-5,000 (增量为 1Hz)		
起始编号 +2	蠕变脉冲频率 (高位字) <sup>*1</sup>	ZRN1、ZRN2: 15-100,000 (增量为 1Hz)	ZRN1-ZRN4: 15-100,000 (增量为 1Hz)	第 18-42 页上的“11. 蠕变脉冲 频率”
起始编号 +3	蠕变脉冲频率 (低位字) <sup>*1</sup>	ZRN3、ZRN4: 15-5,000 (增量为 1Hz)		
起始编号 +4	错误状态			第 18-42 页上的“12. 错误状态”
起始编号 +5	保留			
起始编号 +6	目标速度频率 (高位字) <sup>*1*2</sup>	ZRN1、ZRN2: 15-100,000 (增量为 1Hz)	ZRN1-ZRN4: 15-100,000 (增量为 1Hz)	第 18-42 页上的“13. 目标频率”
起始编号 +7	目标速度频率 (低位字) <sup>*1*2</sup>	ZRN3、ZRN4: 15-5,000 (增量为 1Hz)		
起始编号 +8	频率加大时间 <sup>*2</sup>	10-10,000ms		第 18-42 页上的“14. 加速时间”
起始编号 +9	频率减小时间 <sup>*2</sup>	10-10,000ms		第 18-43 页上的“15. 减速时间”
起始编号 +10	控制方向 <sup>*2*3</sup>	0: 正向 1: 反向		第 18-43 页上的“18. 控制方向”
起始编号 +11	保留			
起始编号 +12	保留			
起始编号 +13	保留			

\*1 高位和低位数据寄存器根据指定的 32 位数据存储方法而变化。有关详情，请参见第 3-10 页上的“32 位数据存储”。

\*2 将归零方式指定为 ZRN 模式 0 时，禁用起始编号 +6 ~ +10。

\*3 仅在方向控制模式选择“启用方向控制‘单脉冲输出’”或“启用方向控制‘双脉冲输出’”时有效。

### 4. S2（源 2）：初始化输入

S2 指定初始化输入。当初始化输入 S2 打开后，“设置”选项卡上 WindLDR “ZRN（归零）”对话框中配置的初始值将存储到控制寄存器中。可以指定外部输入或内部继电器。

当初始化输入打开后，初始值将在每次扫描时写入数据寄存器中。（即使在不执行 ZRN 指令（未打开）的状态下，若打开初始化输入，初始值也将存储在数据寄存器中。）要对值仅初始化一次，请将初始化输入与 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令结合使用。

### 5. S3（源 3）：接近信号输入

S3 指定接近信号。接近信号上升时，开始减速。

可通过外部输入或内部继电器指定接近信号。

检测速度	输入设备	说明
高速	I0、I1、I3、I4、I6、I7	接近信号的读取取决于中断。读取接近信号不受用户程序扫描影响。
正常	I0、I1、I3、I4、I6、I7 以外的外部输入及内部 继电器	结束处理中更新的信息将作为接近信号读取。它们会受用户程序扫描影响。

注释：

- 切勿将 ZRN1、ZRN2、ZRN3 和 ZRN4 指令与接近信号使用相同的输入或内部继电器。如果它们同时运行，则即使接近信号从打开变为关闭，脉冲输出也不会停止。
- 要使用高速接近信号，请将“功能设置”上的“特定输入”设为“标准输入”。切勿设为中断输入、捕捉输入、高速计数器或频率测量。
- 使用高速接近信号时，确保接近信号中不会出现反弹。

**6. S4（源 4）：原始输入**

S4 指定原点信号。检测到原点信号上升时，脉冲输出停止。可指定外部输入或内部继电器。9. 将 ZRN 模式选择为“ZRN 模式 0”时，禁用指定的原点信号。

检测速度	输入设备	说明
高速	I0、I1、I3、I4、I6、I7	中断原点信号的读取取决于中断。 读取原点信号不受用户程序扫描影响。
正常	I0、I1、I3、I4、I6、I7 以外的外部输入及内部继电器	结束处理中更新的信息将作为原点信号读取。 它们会受用户程序扫描影响。

**注释：**

- 切勿通过 ZRN1、ZRN2、ZRN3、ZRN4 指令进行相同输入或将内部继电器作为原点信号使用。若同时操作，即使原点信号从打开变为关闭，脉冲输出仍不会停止。
- 要使用高速原点信号，请在“功能设置”上的“特定输入”将相应输入设为“标准输入”。切勿设为中断输入、捕捉输入、高速计数器或频率测量。
- 使用高速原点信号时，确保原点信号不会出现反弹。

**7. D1（目标 1）：运行状态**

D1 指定用于存储操作状态的内部继电器的第一内部继电器。

从指定的继电器开始，使用 4 个连续的继电器。

请指定设备范围内的第一内部继电器。

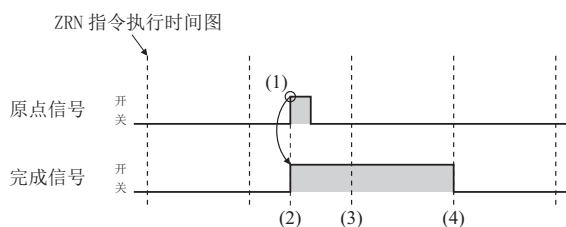
存储目的地	功能	设置	
起始编号 +0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	此继电器在脉冲输出过程中打开。此继电器将在 ZRN 指令输入关闭或脉冲输出结束后关闭。
起始编号 +1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	脉冲输出完成后打开。 脉冲输出开始后关闭。
起始编号 +2	脉冲输出状态	0: 恒速状态 1: 频率变化状态	脉冲输出为频率变化状态时打开。 脉冲输出为恒速状态时关闭。
起始编号 +3	归零完成	0: 归零未完成 1: 归零完成	归零操作完成后打开。 归零操作开始后关闭。

**8. D2（目标 2）：完成输出**

D2 指定完成信号。检测到原点信号上升，停止脉冲输出时，可输出完成信号。可指定外部输出或内部继电器。

仅在归零方式指定为 ZRN 模式 1 时可进行设置。不使用完成信号时可省略。指定为 ZRN 模式 0 时，禁用指定的完成信号。

**注释：**指定外部输出（Q0 ~ Q17）时，不受梯形图的 ZRN 指令执行时间或 I/O 刷新时间影响，完成信号均打开。完成信号打开后，在执行第 2 次 ZRN 指令时恢复关闭状态。



## 18: 脉冲输出指令

### ■ “设置”选项卡

设置 ZRN 指令各功能的操作。



### 9. ZRN 模式

根据目标系统，可从“ZRN 模式 0”和“ZRN 模式 1”2 种中选择归零方法。

设置	说明
ZRN 模式 0	仅监视接近信号，进行归零。
ZRN 模式 1	监视接近信号和原点信号，进行归零。

### 10. 归零速度脉冲频率

指定开始输出脉冲时的频率。

设于 15Hz ~ 100,000Hz（增量为 1Hz）之间。All-in-One CPU 模块使用 ZRN3 或 ZRN4 指令时，可设于 15Hz ~ 5,000Hz（增量为 1Hz）之间。输出频率误差限于 ±5%。

### 11. 蠕变脉冲频率

指定脉冲减速结束变为恒速状态时的频率。

设于 15Hz ~ 100,000Hz（增量为 1Hz）之间。All-in-One CPU 模块使用 ZRN3 或 ZRN4 指令时，可设于 15Hz ~ 5,000Hz（增量为 1Hz）之间。输出频率误差限于 ±5%。

### 12. 错误状态

若配置的设置有误，则输出有误设置对应的错误代码。如果在 ZRN 指令执行时出现配置错误，M8004（用户程序执行错误）将打开并存储错误代码。

错误代码	说明	说明
0	正常	—
2	脉冲频率指定错误	将启动速度频率、蠕变脉冲频率、目标速度频率指定为 15 ~ 100000Hz 以外的值。
5	频率更改时间指定错误	频率更改时间未设置在 10 到 10,000 之间。
7	控制方向指定错误	控制方向未设置为 0 或 1。
9	启动速度频率指定错误	将启动速度频率设为与目标速度频率相同或大于目标速度频率的值。 <sup>*1</sup>
10	蠕变脉冲频率指定错误	将蠕变脉冲频率设为与目标速度频率相同或大于目标速度频率的值。 <sup>*2</sup>

\*1 请将启动速度频率设为低于目标速度频率。

\*2 请将蠕变脉冲频率设为低于目标速度频率。

### 13. 目标频率

指定脉冲加速结束变为恒速状态时的频率。

设于 15Hz ~ 100,000Hz（增量为 1Hz）之间。All-in-One CPU 模块使用 ZRN3 或 ZRN4 指令时，可设于 15Hz ~ 5,000Hz（增量为 1Hz）之间。输出频率误差限于 ±5%。

### 14. 加速时间

指定加大脉冲的时间。

将该时间设置为 10 到 10,000 ms（增量为 1ms）之间。预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。



**15. 减速时间**

指定减小脉冲的时间。

将该时间设置为 10 到 10,000 ms（增量为 1ms）之间。预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。





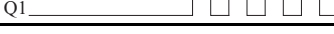
**16. 触发以开始监控原始输入**

指定原点信号的监视开始时间。

时间	说明
减速输入已打开	接近信号从关闭变为打开后，监视原点信号从关闭到打开的变化。
减速输入已关闭	接近信号从关闭变为打开，下一次从打开变为关闭后，监视原点信号从关闭到打开的变化。

**17. 启用方向控制**

9. 将 ZRN 模式选择为“ZRN 模式 1”时，从以下方向控制模式中选择方向控制的方法。（以 All-in-One CPU 模块使用 ZRN1 时为例。）

启用方向控制	动作	模式
禁用	当单向使用脉冲输出时选择此选项。脉冲 A 和 B 可独立使用。	Q0 
方向控制“单脉冲输出”	脉冲 A 用作脉冲输出。脉冲 B 打开 / 关闭用作方向控制。	Q0  Q2 
方向控制“双脉冲输出”	脉冲 A 用作正向脉冲 (CW) 输出。脉冲 B 用作反向脉冲 (CCW) 输出。	Q0  Q1 

FC6A 型上使用的输出因使用的指令、脉冲输出模式和方向控制结合以及使用的型号而异。

指令	操作条件	使用的输出			
		All-in-One CPU 模块		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	
		脉冲输出	方向控制输出	脉冲输出	方向控制输出
ZRN1	禁用方向控制	Q0	—	Q0	—
	方向控制（单脉冲输出模式）	Q0	Q2	Q0	Q1
	方向控制（双脉冲输出模式）	Q0、Q1	—	Q0、Q1	—
ZRN2	禁用方向控制	Q1	—	Q2	—
	方向控制（单脉冲输出模式）	Q1	Q3	Q2	Q3
	方向控制（双脉冲输出模式）	—	—	Q2、Q3	—
ZRN3	禁用方向控制	Q2	—	Q4	—
	方向控制（单脉冲输出模式）	—	—	Q4	Q5
	方向控制（双脉冲输出模式）	—	—	Q4、Q5	—
ZRN4	禁用方向控制	Q3	—	Q6	—
	方向控制（单脉冲输出模式）	—	—	Q6	Q7
	方向控制（双脉冲输出模式）	—	—	Q6、Q7	—

**注释：**用于 ZRN1 指令的输出（脉冲输出、方向控制输出）重复时，无法同时执行使用相同输出的指令。

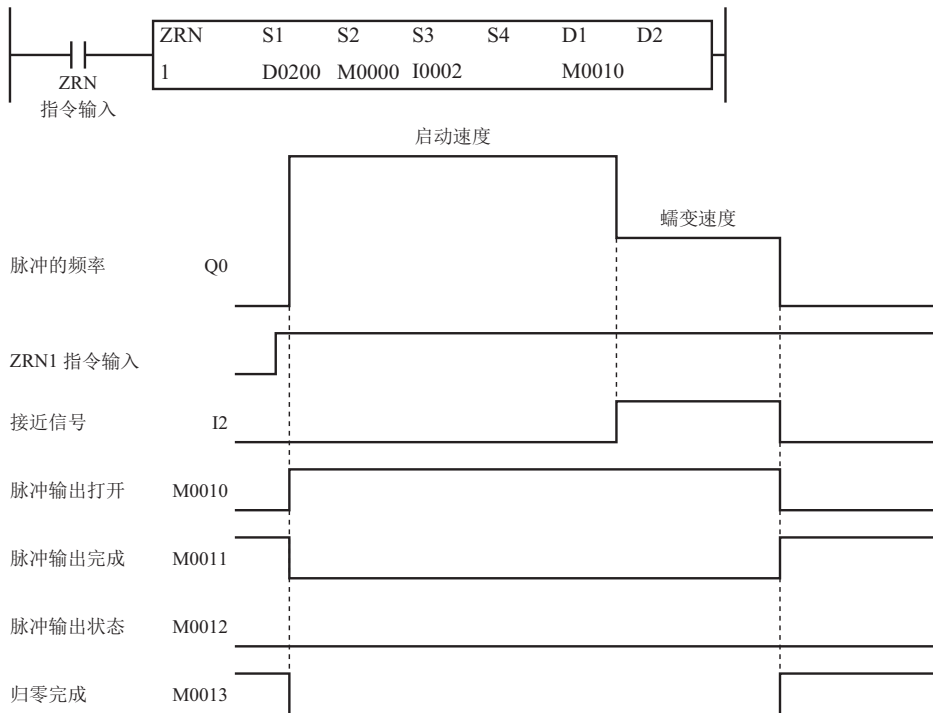
**18. 控制方向**

启用方向控制后，系统将在用于正向操作的数据寄存器中存储 0，在用于反向操作的数据寄存器中存储 1。

## 18: 脉冲输出指令

### ZRN 模式 0（仅使用接近信号）的时间图

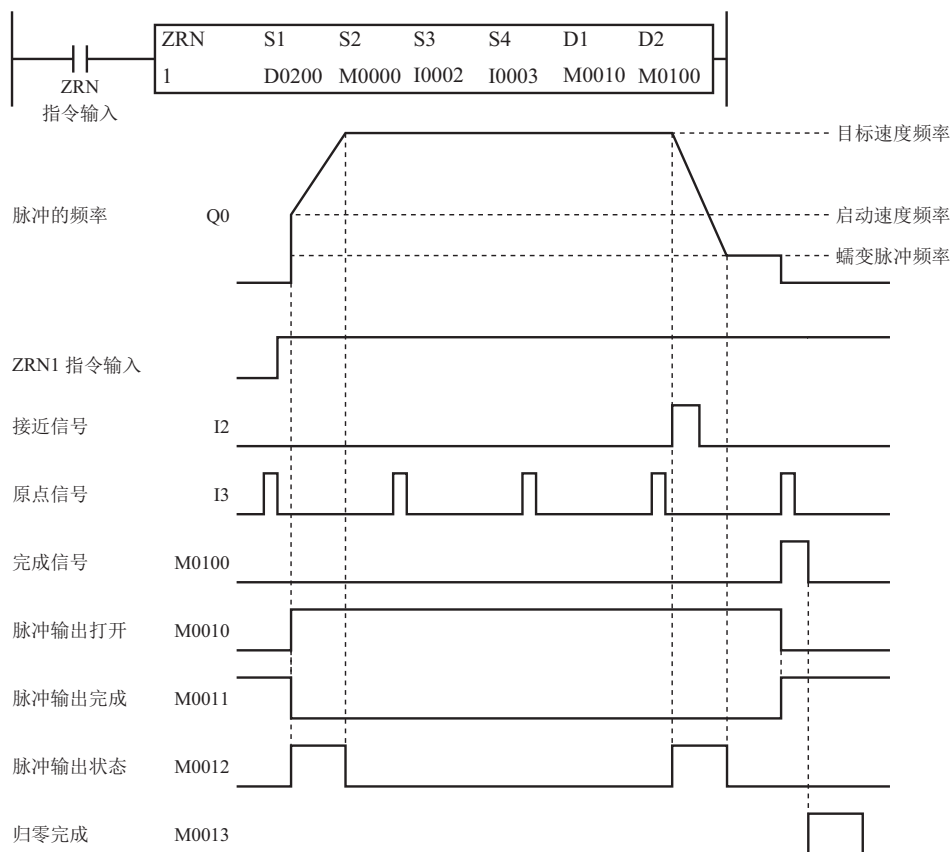
ZRN1 指令，S1 指定为数据寄存器 D0200，S3 指定为外部输入 I2，D1 指定为内部继电器 M0010



- 当 ZRN1 指令输入从关闭变为打开时，脉冲将按启动速度频率输出。
- 脉冲输出开始后，M0010 打开，M0011 和 M0013 关闭。
- 因脉冲为恒速状态，M0012 关闭。
- 当 I2 从关闭变为打开后，脉冲将按蠕变脉冲频率输出。
- 当 I2 从打开变为关闭后，脉冲输出停止。
- 脉冲输出停止后，M0010 关闭，M0011 和 M0013 打开。
- 在脉冲输出过程中关闭 ZRN1 指令输入后，脉冲输出停止。再次打开输入后，从起始开始操作。
- 即使在脉冲输出中变更数据寄存器的设置，也不会反映在脉冲输出操作上。变更内容将在下一次启动 ZRN1 指令时反映出来。

## ZRN 模式 1（使用接近信号、原点信号）的时间图

ZRN1 指令，S1 指定为数据寄存器 D0200，S3 指定为外部输入 I2，D1 指定为内部继电器 M0010



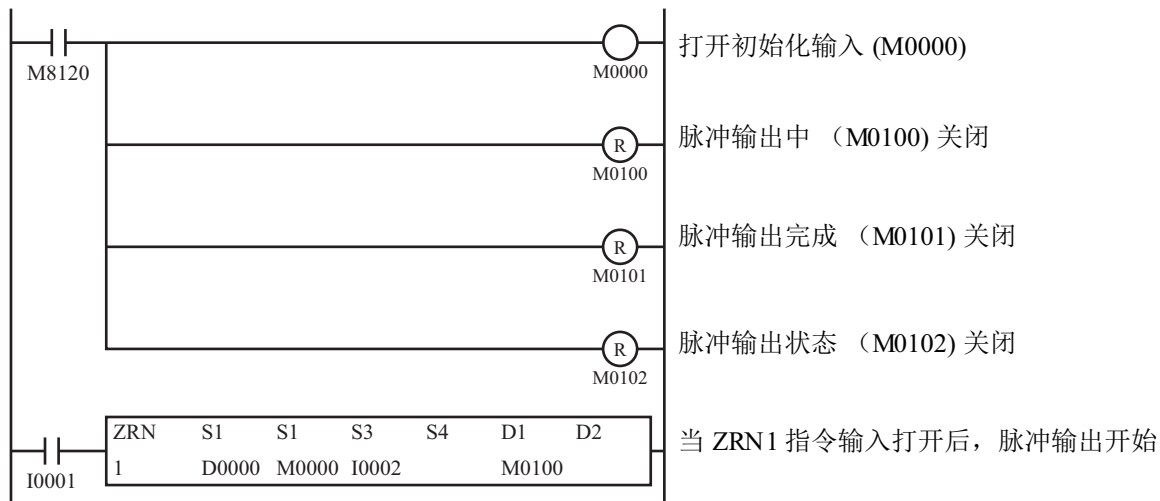
- ZRN1 指令输入从关闭变为打开后，将输出启动速度频率的脉冲，以在频率变化时间内从启动频率到达目标频率的方式输出脉冲。将频率变化时间设置指定为 100 时，以 10ms 为单位进行频率变化，直至在 100ms 后到达目标频率。
- 脉冲输出开始后，M0010 打开，M0011 关闭。
- 当脉冲频率加大或减小的过程中，M0012 打开。
- 当 I2 从关闭变为打开后，脉冲速度开始减小，直至到达蠕变脉冲频率。
- 检测到 I3 从关闭变为打开时，脉冲输出停止。
- 脉冲输出停止后，M0100 打开。M0100 在超过 1 次扫描，不足 2 次扫描时打开。
- M0100 从打开变为关闭，归零操作完成后，M0013 打开。
- 且脉冲输出停止后，M0010 关闭，M0011 打开。
- 在脉冲输出过程中关闭 ZRN1 指令输入后，脉冲输出停止。再次打开输入后，从起始开始操作。
- 即使在脉冲输出中变更数据寄存器的设置，也不会反映在脉冲输出操作上。变更内容将在下一次启动 ZRN1 指令时反映出来。

## 18: 脉冲输出指令

### 操作示例

使用接近信号 I2、启动速度频率 3kHz 和蠕变脉冲频率 800Hz 执行归零操作

#### 梯形图



#### 设置



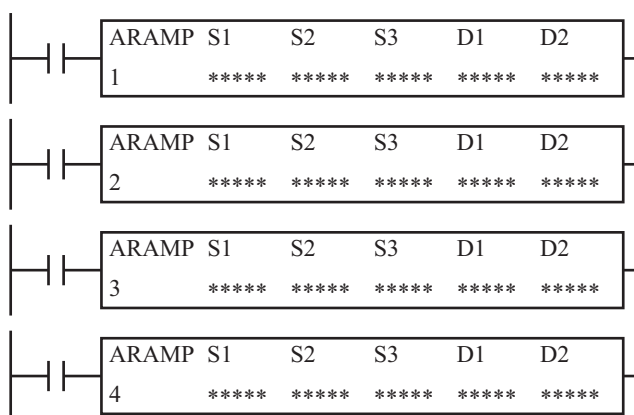
功能	设备地址	预置值	说明
归零速度脉冲频率	D0000、 D0001	3000	3,000Hz
蠕变脉冲频率	D0002、 D0003	800	800Hz

## ARAMP (带表的 RAMP)

根据频率表中的信息利用频率变化功能输出脉冲。

FT2J FC6A\*

※ 仅晶体管输出类型

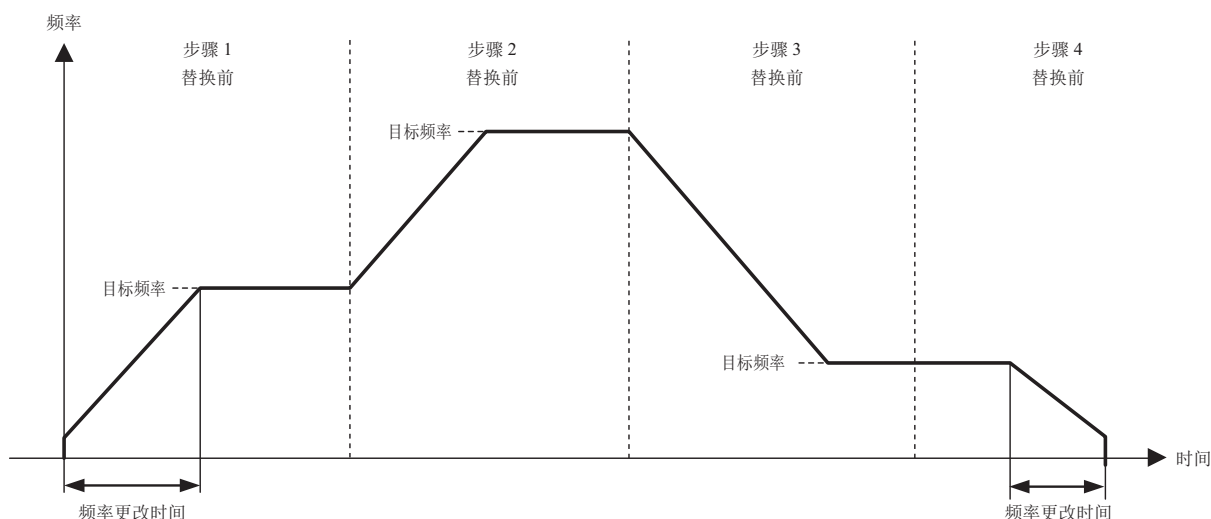


根据步骤配置频率变化和目標频率，经与步骤结合，控制脉冲频率。

当输出的脉冲数达到脉冲数后，将执行各步骤指定的下一步骤。您最多可以设置 18 个步骤。

脉冲输出操作可通过步骤选项配置，从下列 2 个操作中选择。

- 在到达目标频率前以固定比率进行频率变化，并以目标频率输出固定频率的脉冲。  
(下图步骤 1~3 的操作)
- 维持前一步骤的频率输出脉冲后，以固定比率变化直至达到目标频率。  
(下图步骤 4 的操作)



当 S2 指定的初始化输入打开后，WindLDR “ARAMP (Ramp 脉冲输出表)” 中配置的初始值将存储到控制寄存器中。当 S3 指定的中断输入打开后，系统将中止正在执行的步骤并执行中断步骤。

正在执行的步骤的脉冲数和目标频率等将存储到 D1 指定的监控寄存器中。

脉冲输出状态 (正在输出 / 输出方向 / 输出完成) 等控制状态将存储到 D2 指定的操作状态中。

**注释:** 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- ARAMP 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 ARAMP 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

## 18: 脉冲输出指令

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
S3 (源 3)	中断输入	X	—	X	—	—	—	—	—	—	—
D1 (目标 1)	监控寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
D2 (目标 2)	操作状态	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1~7。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 设置

#### • 设备选项卡

#### 1. 选择指令

此项选择要使用的 ARAMP 指令（“ARAMP1”、“ARAMP2”、“ARAMP3”或“ARAMP4”）。

输出、可选择的方向控制模式和操作模式，因指令和 CPU 模块的类型而异。

有关指令组合、方向控制模式和脉冲输出模式导致的限制，请参见第 18-53 页上的“8. 启用方向控制”。

## 2. S1（源 1）：控制寄存器

S1 指定用于 ARAMP1、ARAMP2、ARAMP3 或 ARAMP4 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始，使用“ $2+8 \times N$ （N：步骤数）”个连续的数据寄存器。

可指定的数据寄存器取决于步骤数。

步骤数为 1 时需要 10 个字。

步骤数为 18（最大步骤数）时需要 146 个字。

请指定设备范围内的第一数据寄存器。

步骤将按开始时配置的设置运行。如果步骤设置在开始运行后发生变化，则这些变化不会反映在正在运行的步骤中。

存储目的地	功能	设置		参考
		All-in-One CPU 模块	CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	
起始编号 +0	中断编号	0-18		第 18-53 页上的“11. 中断步骤编号”
起始编号 +1	保留			
步骤 1（8 个字）				
起始编号 +2、 起始编号 +3	目标频率	ARAMP1、ARAMP2: 15-100,000（增量为 1Hz） ARAMP3、ARAMP4: 15-5,000（增量为 1Hz）	ARAMP1-ARAMP4: 15-100,000（增量为 1Hz）	第 18-54 页上的“13. 目标频率”
起始编号 +4	保留			
起始编号 +5	频率更改时间	10-10,000ms		第 18-54 页上的“14. 频率更改时间”
起始编号 +6	预置值（高位字）*1	1-100,000,000 次脉冲		第 18-54 页上的“15. 预置值”
起始编号 +7	预置值（低位字）*1			
起始编号 +8	步骤选项	0-3		第 18-54 页上的“16. 步骤选项”
起始编号 +9	下一步骤编号	1-18		第 18-55 页上的“17. 下一步骤编号”
步骤 2（8 个字）				
起始编号 +10、 起始编号 +11	目标频率	ARAMP1、 ARAMP2: 15-100,000 （增量为 1Hz） ARAMP3、 ARAMP4: 15-5,000 （增量为 1Hz）	ARAMP1-ARAMP4: 15-100,000（增量为 1Hz）	第 18-54 页上的“13. 目标频率”
:	:	:		:
起始编号 +17	下一步骤编号	1-18		第 18-55 页上的“17. 下一步骤编号”
:				
步骤 N（8 个字）				
起始编号 +2+N x 8- 8 起始编号 +3+N x 8- 8	目标频率	ARAMP1、 ARAMP2: 15-100,000 （增量为 1Hz） ARAMP3、 ARAMP4: 15-5,000 （增量为 1Hz）	ARAMP1-ARAMP4: 15-100,000（增量为 1Hz）	第 18-54 页上的“13. 目标频率”
:	:	:		:
起始编号 +9+N x 8- 8	下一步骤编号	1-18		第 18-55 页上的“17. 下一步骤编号”

\*1 高位和低位数据寄存器根据指定的 32 位数据存储方法而变化。

有关详情，请参见第 3-10 页上的“32 位数据存储”。

## 18: 脉冲输出指令

### 3. S2（源 2）：初始化输入

S2 指定初始化输入。

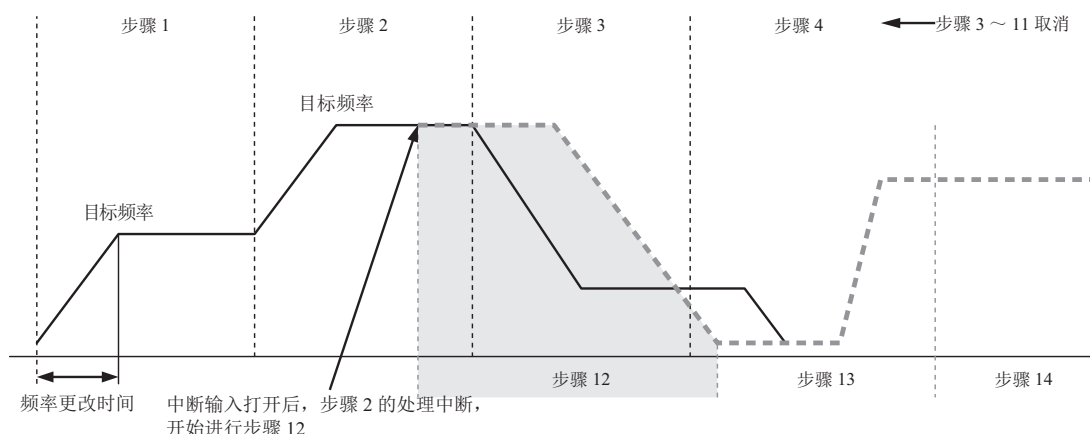
当初始化输入打开后，将“设置”选项卡上 WindLDR 的“ARAMP(Ramp 脉冲输出表)”对话框中配置的初始值将存储到控制寄存器中。可指定外部输入或内部继电器。

初始化输入为打开时，每次扫描都会将初始值存储到数据寄存器中。（即使在不执行 ARAMP 指令（未打开）的状态下，若打开初始化输入，初始值也将存储在数据寄存器中。）要对值仅初始化一次，请将初始化输入与 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令结合使用。

### 4. S3（源 3）：中断输入

S3 指定中断输入。

当中断输入从关闭变为打开时，将中止正在运行的步骤的脉冲输出流程，并且脉冲输出将按中断步骤编号（10.）配置的步骤开始。当中断步骤的脉冲输出完成后，按下一步骤编号（16.）开始下一步骤的脉冲输出。配置为按步骤 1～步骤 18 依次进行脉冲输出，在步骤 2 中打开中断输入的示例如下所示。



当中断输入打开时，将中止正在输出脉冲的步骤 2，按中断步骤编号配置的步骤编号（步骤 12）开始脉冲输出。当步骤 12 完成后，按配置的步骤顺序开始下一步骤的脉冲输出。

可通过外部输入或内部继电器指定中断输入。

检测速度因使用设备而异。

检测速度	设备编号	说明
高速	I0、I1、I3、I4、I6、I7	中断输入的读取取决于中断 读取中断输入不受用户程序扫描影响。
正常	I0、I1、I3、I4、I6、I7 以外的外部输入及内部继电器	结束处理中更新的信息将作为中断输入读取。 它们会受用户程序扫描影响。

不使用中断输入时可省略 S3。

#### 注释：

- 切勿通过 ARAMP1、ARAMP2、ARAMP3、ARAMP4 指令进行相同输入或将内部继电器作为中断输入信号使用。
- 要使用高速中断输入信号时，请将“功能设置”上的“特定输入”将相应输入设为“标准输入”。切勿设为中断输入、捕捉输入、高速计数器或频率测量。
- 使用高速中断输入时，确保中断输入中不会出现反弹。
- 当中断输入打开时，无论中断步骤的控制方向如何，控制方向（正向或反向）都保持不变。



### 5. D1（目标 1）：监控寄存器

D1 指定用于 ARAMP1、ARAMP2、ARAMP3 或 ARAMP4 的数据寄存器的第一数据寄存器。从指定的数据寄存器开始，使用 11 个连续的数据寄存器。请指定设备范围内的第一数据寄存器。监控寄存器的内容为只读。

地址	功能	设置		参考
		All-in-One CPU 模块	CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	
起始编号 +0	下一步骤编号	0-18		第 18-51 页上的“下一步骤编号”
起始编号 +1	正在运行的步骤编号	1-18		第 18-51 页上的“正在运行的步骤编号”
起始编号 +2	目标频率监控 (高位字) *1	ARAMP1、ARAMP2: 15-100,000 (增量为 1Hz) ARAMP3、ARAMP4: 15-5,000 (增量为 1Hz)	ARAMP1-ARAMP4: 15-100,000 (增量为 1Hz)	第 18-51 页上的“目标频率监控”
起始编号 +3	目标频率监控 (低位字) *1			
起始编号 +4	保留			
起始编号 +5	频率更改时间监控	10-10,000 (增量为 1Hz) 预置值的个位作为 0 处理。		第 18-51 页上的“频率更改时间监控”
起始编号 +6	预置值监控 (高位字) *1	1-100,000,000 次脉冲		第 18-51 页上的“预置值监控”
起始编号 +7	预置值监控 (低位字) *1			
起始编号 +8	当前值 (高位字) *1	1-100,000,000 次脉冲		第 18-51 页上的“当前值”
起始编号 +9	当前值 (低位字) *1			
起始编号 +10	错误状态	0-9		第 18-52 页上的“错误状态”

\*1 高位和低位数据寄存器根据指定的 32 位数据存储方法而变化。  
有关详情，请参见第 3-10 页上的“32 位数据存储”。

#### 下一步骤编号

此寄存器存储下一步要执行的步骤编号。

如果下一步为 0，脉冲输出将在当前执行的步骤完成后结束。

#### 正在运行的步骤编号

此寄存器存储当前正在执行的步骤的编号。

#### 目标频率监控

此寄存器存储当前正在执行的步骤的目标频率。

#### 频率更改时间监控

此寄存器存储当前正在执行的步骤的频率更改时间。

将该时间设置为 10 到 10,000ms（增量为 1ms）之间。预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。

#### 预置值监控

存储执行中的步骤输出的脉冲数。

#### 当前值

此寄存器存储当前正在执行的步骤已输出的脉冲数。每次扫描执行 ARAMP 指令时，当前值都会进行更新。

## 18: 脉冲输出指令

### 错误状态

若配置的设置有误，则输出有误设置对应的错误代码。如果各步骤开始执行时，执行的步骤出现配置错误，将导致用户程序执行错误，在 D8006 中存储错误代码 20。

错误代码	状态	说明		
0	正常	—		
3	预置值指定错误	预置值未设置在 1 到 100,000,000 之间。		
4	目标频率指定错误	All-in-One CPU 模块	ARAMP1、ARAMP2	目标频率未设置在 15 到 100,000 之间。
			ARAMP3、ARAMP4	目标频率未设置在 15 到 5,000 之间。
		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	ARAMP1 - ARAMP4	目标频率未设置在 15 到 100,000 之间。
5	频率更改时间指定错误	频率更改时间未设置在 10 到 10,000 之间。		
7	步骤选项指定错误	步骤选项未设置为有效值。		
8	下一步骤编号指定错误	下一步骤编号未设置在 0 到 18 之间。		
9	中断编号指定错误	中断编号未设置在 1 到 18 之间。		

### 6. D2（目标 2）：运行状态

D2 指定用于 ARAMP1、ARAMP2、ARAMP3 或 ARAMP4 指令的内部继电器的起始编号。

从指定的内部继电器开始，使用 5 个连续的内部继电器。

请指定设备范围内的起始编号。

地址		说明	
起始编号 +0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	此继电器在脉冲输出过程中打开。 此继电器在 ARAMP 指令输出停止后关闭。 此继电器在指定次数的脉冲输出结束后关闭。
起始编号 +1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	此继电器在脉冲输出完成后打开。 此继电器在当前正在执行的步骤编号为 0 时打开。 此继电器在 ARAMP 指令输出开始后关闭。
起始编号 +2	脉冲输出状态	0: 目标脉冲输出 1: 更改输出脉冲频率	此继电器在脉冲输出状态稳定后关闭。 此继电器在脉冲输出变化时打开。
起始编号 +3	上溢出	0: 无 1: 出现上溢出	此继电器在脉冲输出超过配置的预置值后打开。 在稳定输出过程中出现上溢出或脉冲频率变化时，脉冲输出将保持平稳。但是，在出现上溢出时脉冲计数（当前值数据寄存器）将暂停。
起始编号 +4	脉冲输出方向	0: 正向 1: 反向	此继电器当正在输出的脉冲的输出方向为正向时关闭。 此继电器当正在输出的脉冲的输出方向为反向时打开。

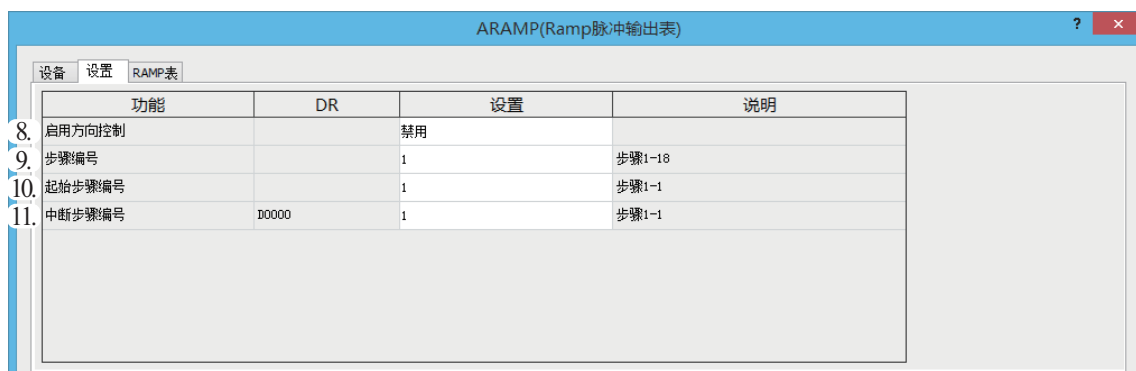
### 7. 预览

预览显示配置的 ARAMP 指令的操作。可确认脉冲输出的频率变化、正向 / 反向操作、步骤的执行顺序。

纵轴表示脉冲频率，横轴表示时间。

因各步骤宽度固定，实际的横轴比率不正确。

## ■ 设置选项卡



### 8. 启用方向控制

此设置启用或禁用方向控制，并从以下模式中选择方向控制方法。脉冲输出模式分为两种：单脉冲输出模式和双脉冲输出模式。它们可以与方向控制结合使用，如下所示。（以 All-in-One CPU 模块使用 ARAMP 时为例。）

启用方向控制	动作	模式
禁用方向控制	当单向使用脉冲输出时选择此选项。脉冲 A 和 B 可独立使用。	Q0
方向控制 “单脉冲输出”	脉冲 A 用作脉冲输出。脉冲 B 打开 / 关闭用作方向控制。	Q0
方向控制 “双脉冲输出”	脉冲 A 用作正向脉冲 (CW) 输出。 脉冲 B 用作反向脉冲 (CCW) 输出。	Q0

FC6A 型上使用的输出因使用的指令、脉冲输出模式和方向控制结合以及使用的型号而异。

指令	操作条件	使用的输出			
		All-in-One CPU 模块		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	
		脉冲输出	方向控制输出	脉冲输出	方向控制输出
ARAMP1	禁用方向控制	Q0	—	Q0	—
	方向控制（单脉冲输出）	Q0	Q2*1	Q0	Q1
	方向控制（双脉冲输出）	Q0、Q1*2	—	Q0、Q1	—
ARAMP2	禁用方向控制	Q1	—	Q2	—
	方向控制（单脉冲输出）	Q1	Q3*1	Q2	Q3
	方向控制（双脉冲输出）	—	—	Q2、Q3	—
ARAMP3	禁用方向控制	Q2	—	Q4	—
	方向控制（单脉冲输出）	—	—	Q4	Q5
	方向控制（双脉冲输出）	—	—	Q4、Q5	—
ARAMP4	禁用方向控制	Q3	—	Q6	—
	方向控制（单脉冲输出）	—	—	Q6	Q7
	方向控制（双脉冲输出）	—	—	Q6、Q7	—

\*1 使用单脉冲输出模式时，因使用 Q2 或 Q3，不可使用 Q2 或 Q3 的脉冲输出指令。

\*2 使用双脉冲输出模式时，因使用 Q1，不可使用 Q1 的脉冲输出指令

### 9. 步骤编号

此设置指定步骤的数目。最多为 18 步。

### 10. 起始步骤编号

当输入从关闭变为打开时，脉冲输出将以配置为起始编号的步骤的设置开始。

### 11. 中断步骤编号

当中断输入从关闭变为打开时，将中止正在运行的步骤的脉冲输出流程，并且脉冲输出将按中断编号配置的步骤设置重新开始。

## 18: 脉冲输出指令

### ■ “RAMP 表” 选项卡

步骤	功能	DR	设置	说明
13.	目标频率	D0002, D0003	100	15 ~ 100,000, 增量为 1 Hz
14.	频率更改时间	D0005	100	10 ~ 10,000, 增量为 1 (毫秒)
15.	预置值	D0006, D0007	100000000	1-100,000,000
16.	控制方向	D0008		
16.	变化时间	D0008	替换前	
17.	下一步骤编号	D0009	1	步骤1-1

#### 12. 步骤编号

此选项选择要配置的步骤编号。输出频率误差限于  $\pm 5\%$ 。

#### 13. 目标频率

此设置指定频率变化前后处于稳定输出状态的频率。

CPU 模块的类型	命令	目标频率
All-in-One CPU 模块	ARAMP1、 ARAMP2	在 15-100,000 (增量为 1Hz) 的范围内设置此值。
	ARAMP3、 ARAMP4	在 15-5,000 (增量为 1Hz) 的范围内设置此值。
CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	ARAMP1-ARAMP4	在 15-100,000 (增量为 1Hz) 的范围内设置此值。

#### 14. 频率更改时间

此设置指定脉冲变化的时间。

在 10-10,000 (增量为 1ms) 的范围内设置此值。

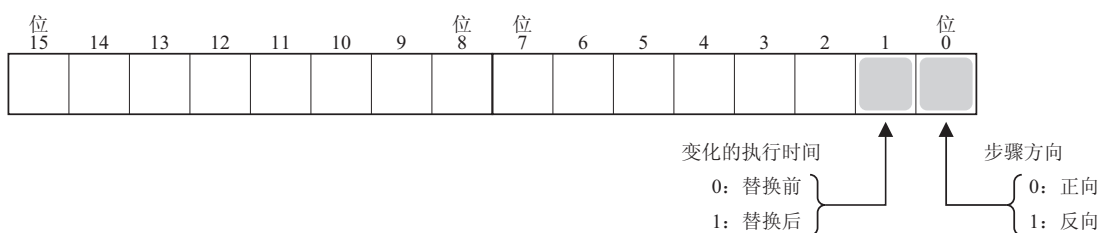
预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。

#### 15. 预置值

此设置配置输出的脉冲数，范围介于 1-100,000,000 之间。

#### 16. 步骤选项

此设置配置步骤方向和脉冲变化的执行时间。

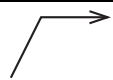
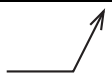

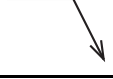


#### 控制方向

启用方向控制后，系统将在用于正向操作的数据寄存器中存储 0，在用于反向操作的数据寄存器中存储 1。

### 变化时间

频率根据变化执行时间的设置变化，如下图所示。在“替换前”列中，频率先发生变化，然后变得稳定。当输出的脉冲数达到预置值后，该指令将过渡到下一步骤。在“替换后”列中，脉冲频率保持稳定，接下来频率变化，该指令过渡到下一步骤。

	替换前	替换后
加大		
减小		

**注释：**依据步骤组合和中断输入的时间，可以通过高频率切换正向 / 反向。请进行编程，以便根据使用的应用程序规格进行切换。

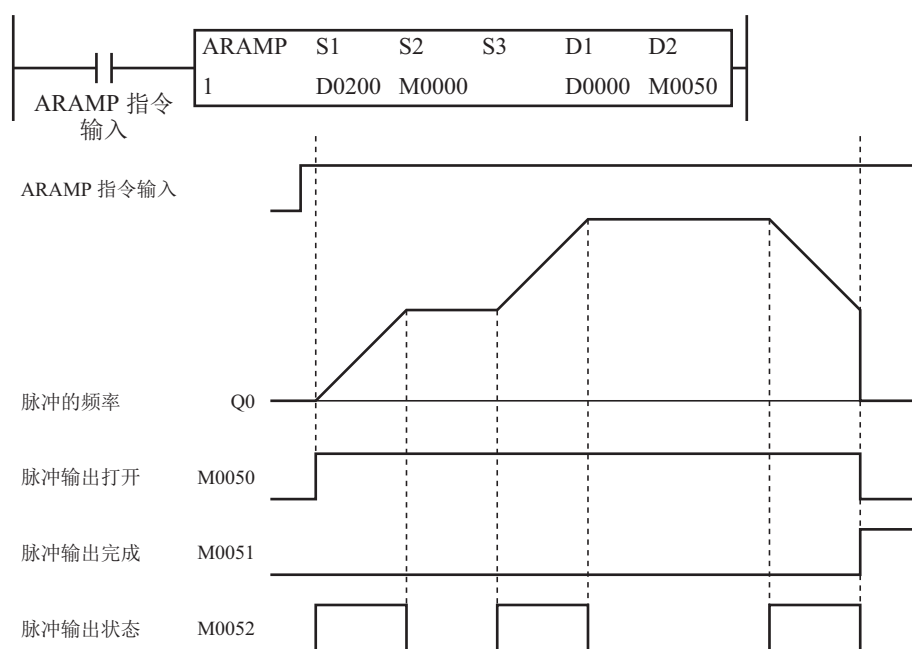
### 17. 下一步骤编号

此设置指定，在当前所执行步骤的输出完成后要执行的下一步骤的编号。

将下一执行步骤编号配置为 0 后，则该步骤为最终步骤，在该步骤的脉冲输出完成后，结束脉冲输出。

### ARAMP1 指令（禁用方向控制）时间图

**ARAMP1 指令**，S1 指定为数据寄存器 D0200，S2 指定为内部继电器 M0000，S3 禁用，D1 指定为数据寄存器 D0000，D2 指定为内部继电器 M0050



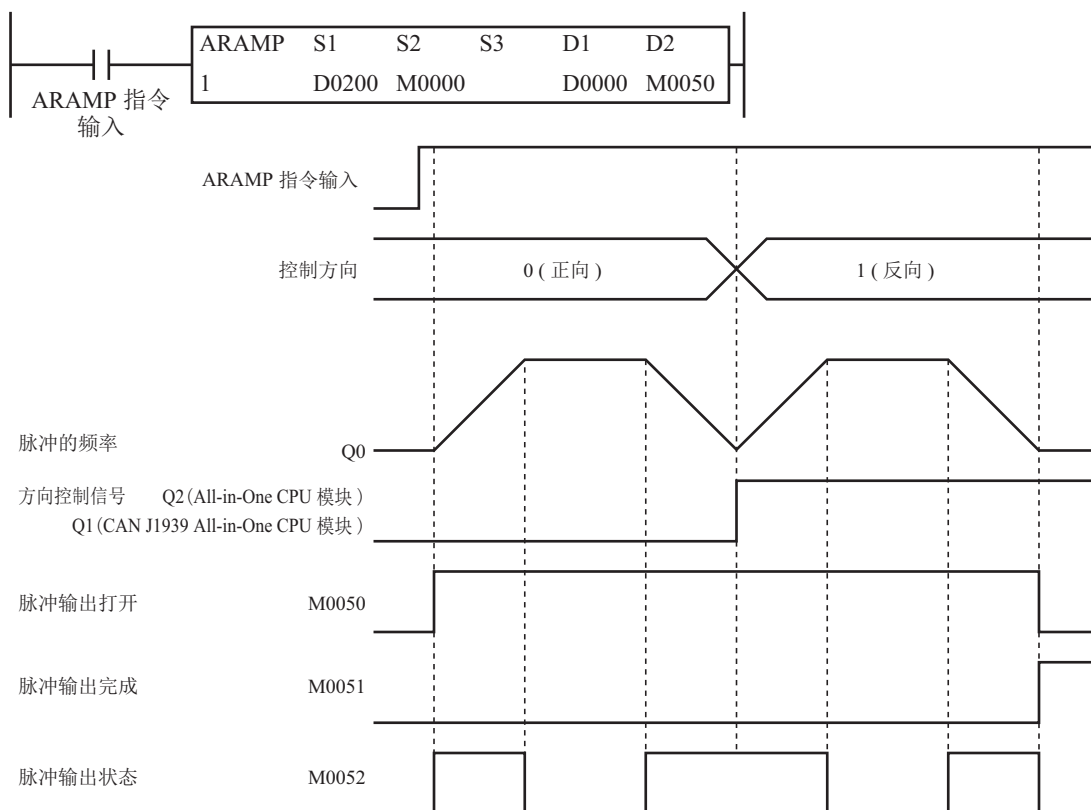
当 ARAMP 指令从关闭变为打开时，脉冲将根据数据寄存器配置的设置输出。脉冲输出开始后，M0050 将打开。当脉冲频率加大或减小时，M0052 打开。各步骤输出的脉冲数达到脉冲数后，将执行下一步骤。将下一执行步骤编号配置为 0 后，在该步骤的脉冲输出完成后停止脉冲。此时，M0050 关闭，M0051 打开。

如果 ARAMP 指令输入在脉冲输出过程中关闭，脉冲输出将结束。如果 ARAMP 指令输入再次打开，操作将从头开始。即使数据寄存器的内容在脉冲输出过程中发生变化，脉冲输出操作中也不会反映这种变化。变化的内容会在下次启动 ARAMP 指令时反映出来。

## 18: 脉冲输出指令

### ARAMP1 指令（启用单脉冲输出方向控制）时间图

ARAMP1 指令，S1 指定为数据寄存器 D0200，S2 指定为内部继电器 M0000，S3 禁用，D1 指定为数据寄存器 D0000，D2 指定为内部继电器 M0050



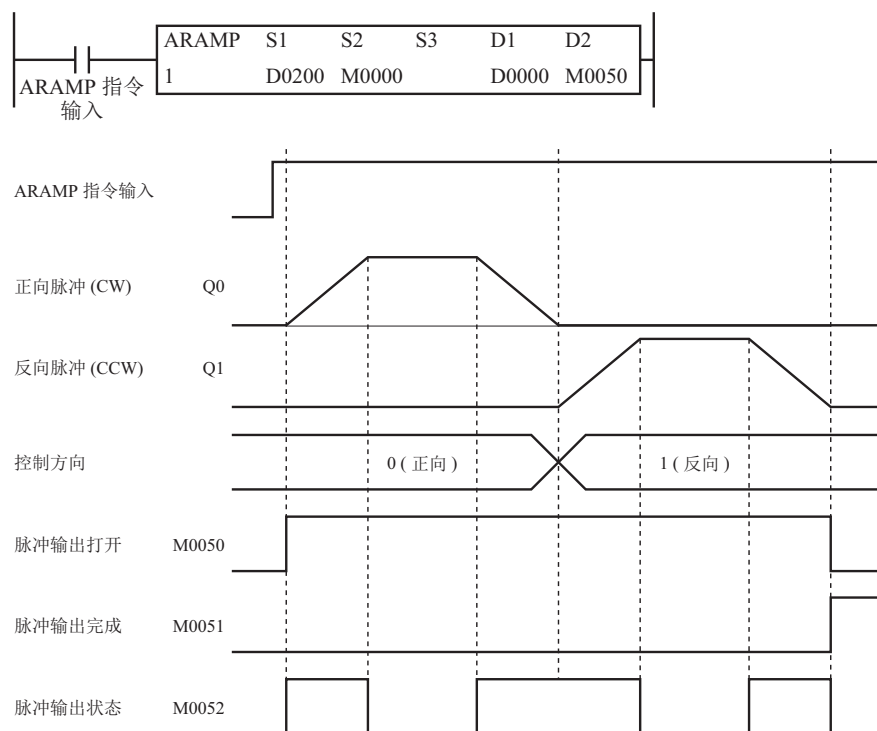
当 ARAMP 指令从关闭变为打开时，脉冲将根据数据寄存器配置的设置从 Q0 中输出。方向控制信号从 Q2 或 Q1 中输出。脉冲输出开始后，M0050 将打开。当脉冲频率加大或减小时，M0052 打开。各步骤输出的脉冲数达到脉冲数后，将执行下一步骤。将下一执行步骤编号配置为 0 后，在该步骤的脉冲输出完成后停止脉冲。此时，M0050 关闭，M0051 打开。

如果 ARAMP 指令输入在脉冲输出过程中关闭，脉冲输出将结束。如果此输入再次打开，则操作将从头开始。即使数据寄存器的内容在脉冲输出过程中发生变化，脉冲输出操作中也不会反映这种变化。变化的内容会在下次启动 ARAMP 指令时反映出来。

应用程序在切换正向 / 反向时，必须将脉冲频率降到最小值后方可执行，因此在反向前插入降低频率的步骤。

## ARAMP1 指令（启用双脉冲输出方向控制）时间图

All-in-One CPU 模块，ARAMP1 指令，S1 指定为数据寄存器 D0200，S2 指定为内部继电器 M0000，S3 禁用，D1 指定为数据寄存器 D0000，D2 指定为内部继电器 M0050



当 ARAMP 指令从关闭变为打开时，脉冲将根据数据寄存器配置的设置从 Q0 中输出。方向控制信号从 Q0 或 Q1 中输出。脉冲输出开始后，M0050 将打开。当脉冲频率加大或减小时，M0052 打开。脉冲根据频率更改时间加大和减小，直到从当前的脉冲频率达到目标频率。输出配置的脉冲数时，脉冲停止。此时，M0050 关闭，M0051 打开。

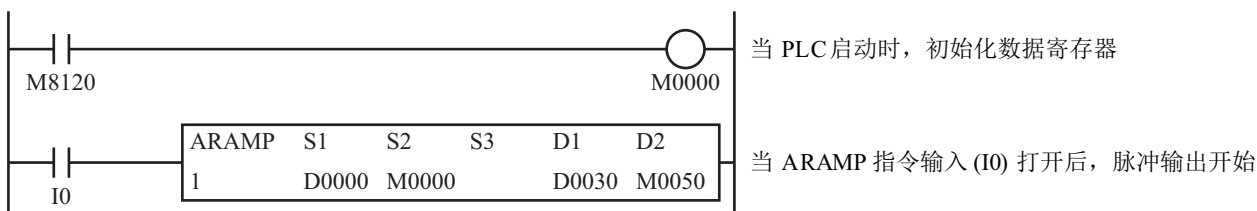
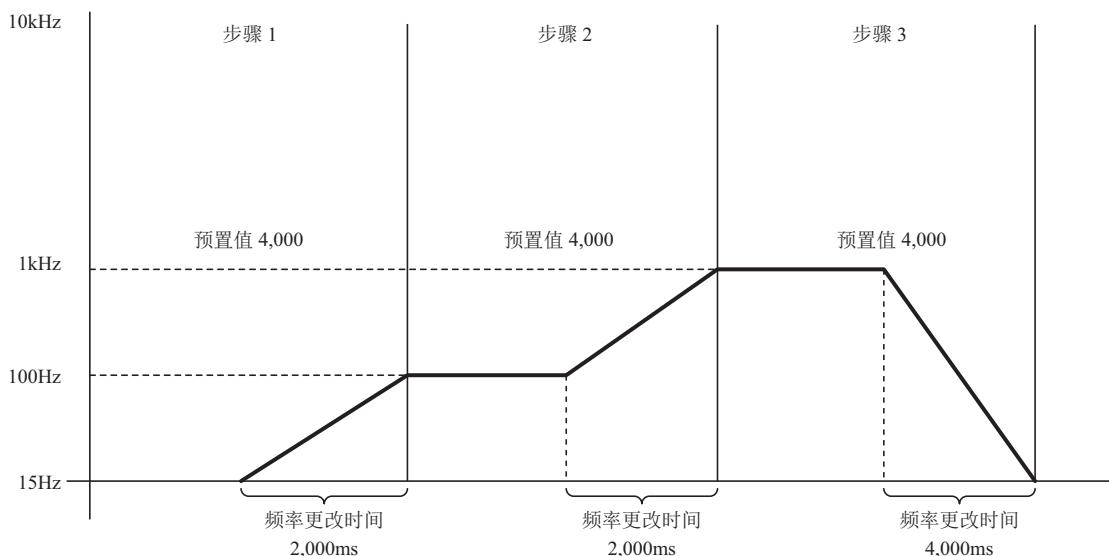
如果 ARAMP 指令输入在脉冲输出过程中关闭，脉冲输出将结束。如果 ARAMP 指令输入再次打开，操作将从头开始。即使数据寄存器的内容在脉冲输出过程中发生变化，脉冲输出操作中也不会反映这种变化。变化的内容会在下次启动 ARAMP 指令时反映出来。

应用程序在切换正向 / 反向时，必须将脉冲频率降到最小值后方可执行，因此在反向前插入降低频率的步骤。

## 18: 脉冲输出指令

### 示例程序

使用以下设置利用频率变化功能（禁用方向控制）如下图所示输出脉冲时这些脉冲从 Q0 中输出。



### 基本设置



功能	设备地址	设置值	详情
启用方向控制	—	禁用	—
步骤编号	—	3	—
起始步骤编号	—	1	步骤 1
中断步骤编号	D0000	1	—



## 步骤 1 设置

ARAMP(Ramp脉冲输出表)

功能	DR	设置	说明
目标频率	D0002, D0003	100	15 ~ 100,000, 增量为 1 Hz
频率更改时间	D0005	2000	10 ~ 10,000, 增量为 1 (毫秒)
预置值	D0006, D0007	4000	1-100,000,000
控制方向	D0008		
变化时间	D0008	替换后	
下一步骤编号	D0009	2	步骤1-3

功能	设备地址	设置值	详情
目标频率	D0002、D0003	100	100Hz
频率更改时间	D0005	2000	2,000ms
预置值	D0006、D0007	4000	预置值 =4,000
控制方向	D0008	—	—
变化时间	D0008	替换后	替换后 =2
下一步骤编号	D0009	2	步骤 2

## 步骤 2 设置

ARAMP(Ramp脉冲输出表)

功能	DR	设置	说明
目标频率	D0010, D0011	1000	15 ~ 100,000, 增量为 1 Hz
频率更改时间	D0013	2000	10 ~ 10,000, 增量为 1 (毫秒)
预置值	D0014, D0015	4000	1-100,000,000
控制方向	D0016		
变化时间	D0016	替换后	
下一步骤编号	D0017	3	步骤 1-3

功能	设备地址	设置值	详情
目标频率	D0010、D0011	1000	1,000Hz
频率更改时间	D0013	2000	2,000ms
预置值	D0014、D0015	4000	预置值 =4,000
控制方向	D0016	—	—
变化时间	D0016	替换后	替换后 =2
下一步骤编号	D0017	3	步骤 3

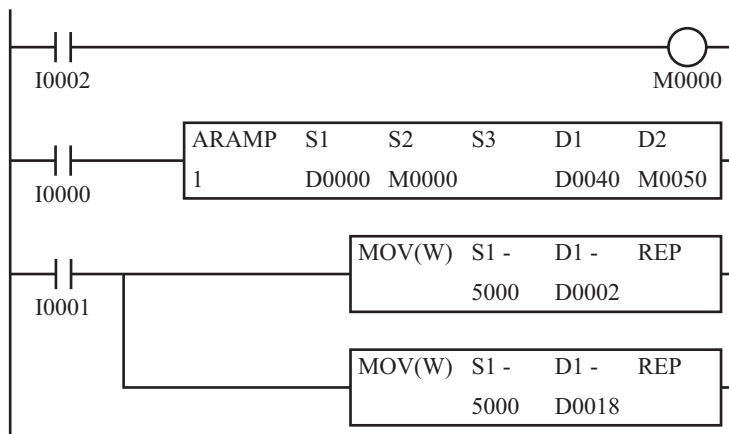
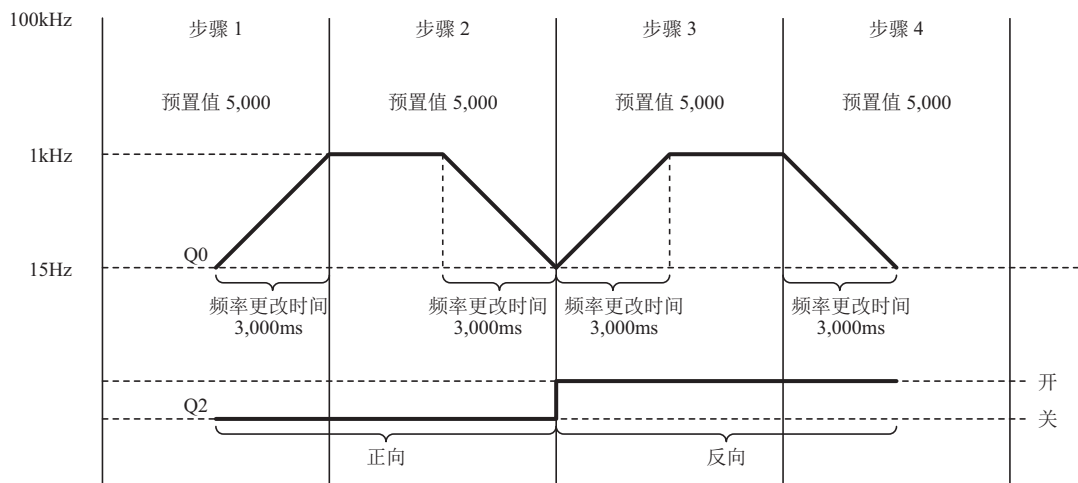
## 18: 脉冲输出指令

### 步骤 3 设置

步骤	功能	DR	设置	说明
步骤 1	目标频率	D0018, D0019	15	15 ~ 100,000, 增量为 1 Hz
步骤 2	频率更改时间	D0021	4000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 3	预置值	D0022, D0023	4000	1~100,000,000
	控制方向	D0024		
	变化时间	D0024	替换后	
	下一步骤编号	D0025	0	步骤 1-3

功能	设备地址	设置值	详情
目标频率	D0018、D0019	15	15Hz
频率更改时间	D0021	4000	4,000ms
预置值	D0022、D0023	4000	预置值 =4,000
控制方向	D0024	—	—
变化时间	D0024	替换后	替换后 =2
下一步骤编号	D0025	0	0= 结束输出

使用以下设置利用频率变化功能（启用单脉冲输出的方向控制）如下图所示输出脉冲时从 Q0 输出脉冲，从 Q2 输出方向控制信号。

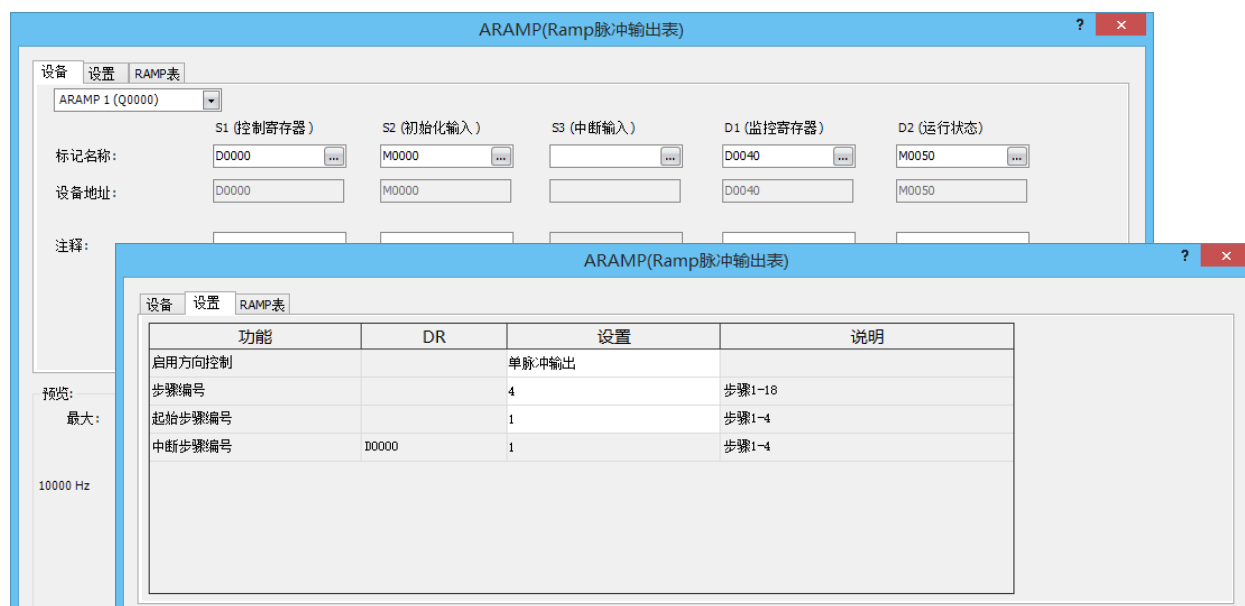


当 I2 打开后，初始化输入 (M0000) 打开，ARAMP 指令控制寄存器值将进行初始化

当 ARAMP 指令输入 (I0) 打开后，脉冲输出开始

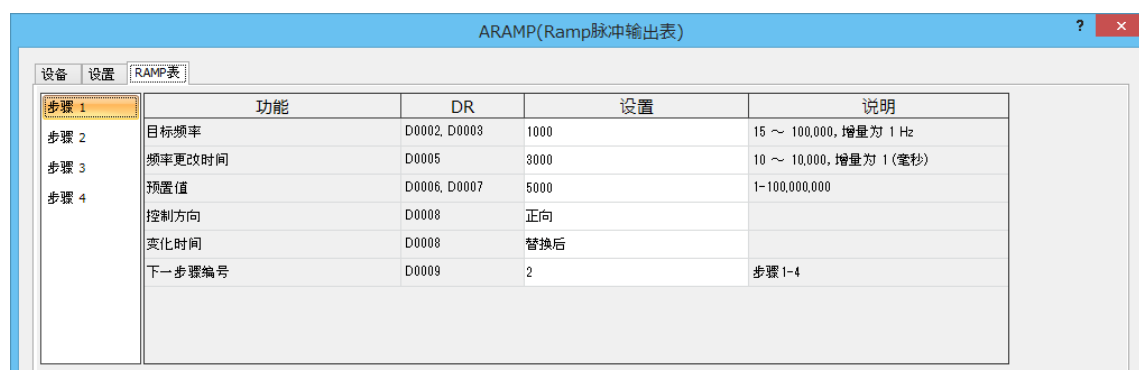
当 MOV 指令输入 (I1) 打开后，步骤 2 和 4 (D0002、D0003 和 D0018、D0019) 的目标频率变为 5kHz。

## 基本设置



功能	设备地址	设置值	详情
启用方向控制	—	单脉冲输出	—
步骤编号	—	4	—
起始步骤编号	—	1	步骤 1
中断步骤编号	D0000	1	—

## 步骤 1 设置



功能	设备地址	设置值	详情
目标频率	D0002、 D0003	1000	1,000Hz
频率更改时间	D0005	3000	3,000ms
预置值	D0006、 D0007	5000	预置值 =5,000
控制方向	D0008	正向	正向
变化时间	D0008	替换后	替换后 =2
下一步骤编号	D0009	2	步骤 2

## 18: 脉冲输出指令

### 步骤 2 设置

步骤	功能	DR	设置	说明
步骤 1				
步骤 2	目标频率	D0010, D0011	15	15 ~ 100,000, 增量为 1 Hz
步骤 3	频率更改时间	D0013	3000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 4	预置值	D0014, D0015	5000	1~100,000,000
	控制方向	D0016	正向	
	变化时间	D0016	替换后	
	下一步骤编号	D0017	3	步骤1~4

功能	设备地址	设置值	详情
目标频率	D0010、D0011	15	15Hz
频率更改时间	D0013	3000	3,000ms
预置值	D0014、D0015	5000	预置值 =5,000
控制方向	D0016	正向	正向
变化时间	D0016	替换后	替换后 =2
下一步骤编号	D0017	3	步骤 3

### 步骤 3 设置

步骤	功能	DR	设置	说明
步骤 1				
步骤 2	目标频率	D0018, D0019	1000	15 ~ 100,000, 增量为 1 Hz
步骤 3	频率更改时间	D0021	3000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 4	预置值	D0022, D0023	5000	1~100,000,000
	控制方向	D0024	反向	
	变化时间	D0024	替换前	
	下一步骤编号	D0025	4	步骤 1~4

功能	设备地址	设置值	详情
目标频率	D0018、D0019	1000	1,000Hz
频率更改时间	D0021	3000	3,000ms
预置值	D0022、D0023	5000	预置值 =5,000
控制方向	D0024	反向	反向
变化时间	D0024	替换前	替换前 =0
下一步骤编号	D0025	4	步骤 4

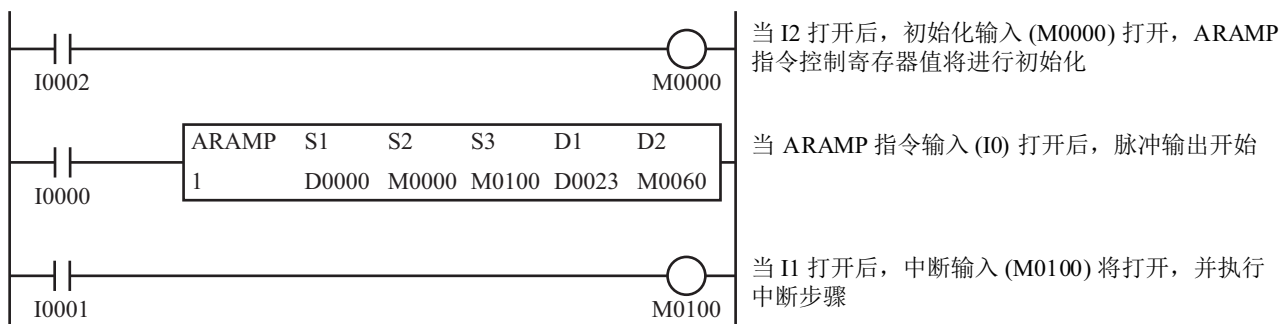
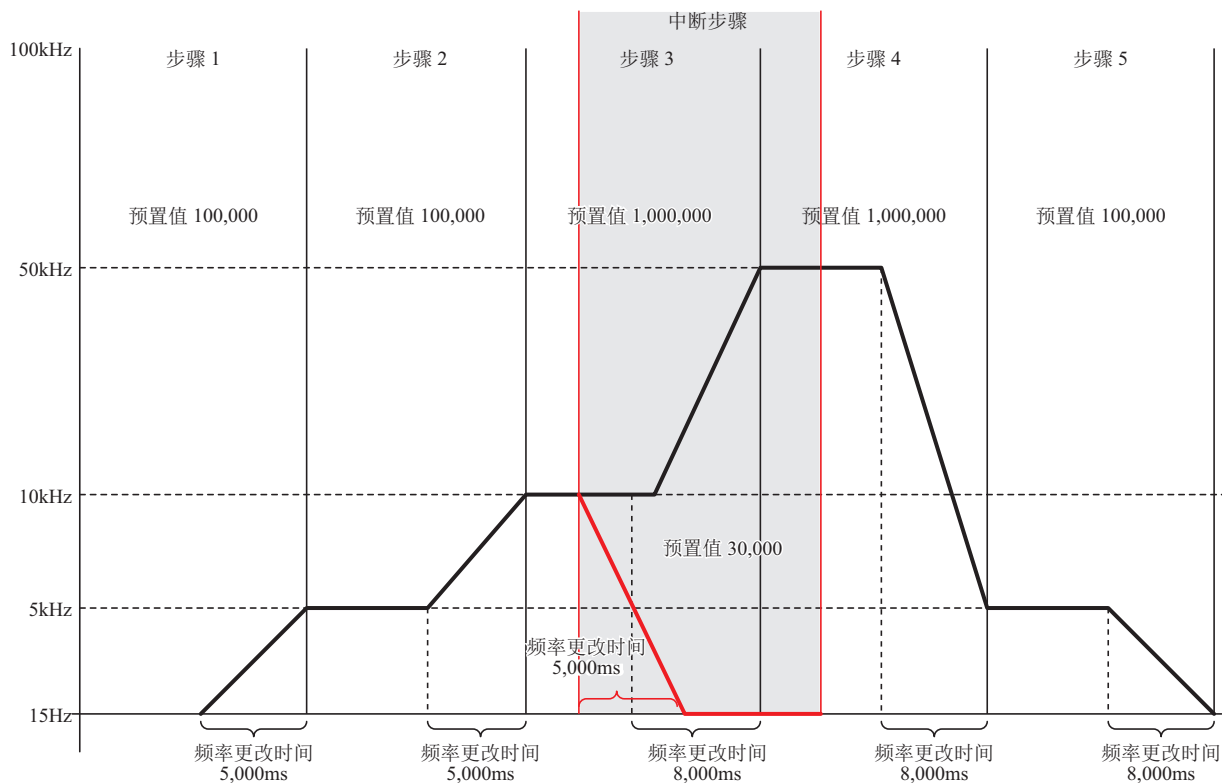
## 步骤 4 设置

步骤	功能	DR	设置	说明
步骤 1	目标频率	D0026, D0027	15	15 ~ 100,000, 增量为 1 Hz
步骤 2	频率更改时间	D0029	3000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 3	预置值	D0030, D0031	5000	1-100,000,000
步骤 4	控制方向	D0032	反向	
	变化时间	D0032	替换前	
	下一步骤编号	D0033	0	步骤1-4

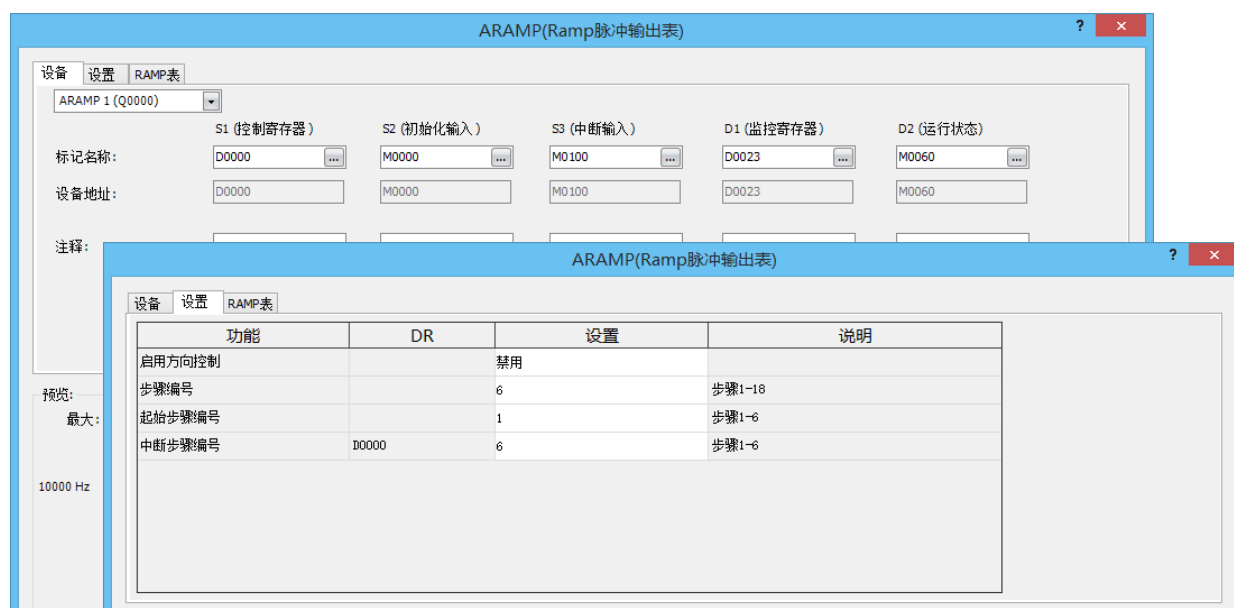
功能	设备地址	设置值	详情
目标频率	D0026、 D0027	15	15Hz
频率更改时间	D0029	3000	3,000ms
预置值	D0030、 D0031	5000	预置值 =5,000
控制方向	D0032	反向	反向
变化时间	D0032	替换前	替换前 =0
下一步骤编号	D0033	0	步骤 0 (结束)

## 18: 脉冲输出指令

使用以下设置利用频率变化功能（禁用单脉冲输出的方向控制）如下图所示输出脉冲时这些脉冲从 Q0 中输出。

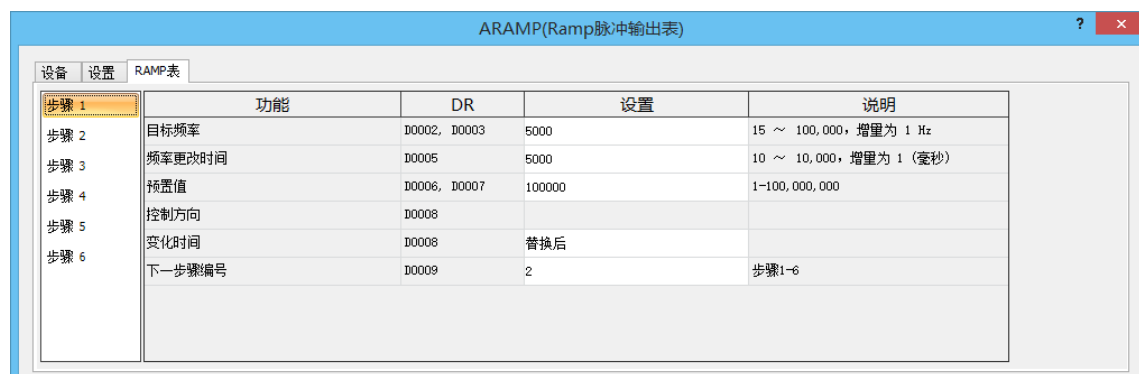


## 基本设置



功能	设备地址	设置值	详情
启用方向控制	—	禁用	—
步骤编号	—	6	—
起始步骤编号	—	1	步骤 1
中断步骤编号	D0000	6	步骤 6

## 步骤 1 设置



功能	设备地址	设置值	详情
目标频率	D0002、 D0003	5000	5,000Hz
频率更改时间	D0005	5000	5,000ms
预置值	D0006、 D0007	100000	预置值 =100,000
控制方向	D0008	—	—
变化时间	D0008	替换后	替换后 =2
下一步骤编号	D0009	2	步骤 2

## 18: 脉冲输出指令

### 步骤 2 设置

步骤	功能	DR	设置	说明
步骤 1				
步骤 2	目标频率	D0010, D0011	10000	15 ~ 100,000, 增量为 1 Hz
步骤 3	频率更改时间	D0013	5000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 4	预置值	D0014, D0015	1000000	1-100,000,000
步骤 5	控制方向	D0016		
步骤 6	变化时间	D0016	替换后	
	下一步骤编号	D0017	3	步骤1-6

功能	设备地址	设置值	详情
目标频率	D0010、D0011	10000	10,000Hz
频率更改时间	D0013	5000	5,000ms
预置值	D0014、D0015	100000	预置值=100,000
控制方向	D0016	—	—
变化时间	D0016	替换后	替换后=2
下一步骤编号	D0017	3	步骤 3

### 步骤 3 设置

步骤	功能	DR	设置	说明
步骤 1				
步骤 2	目标频率	D0018, D0019	50000	15 ~ 100,000, 增量为 1 Hz
步骤 3	频率更改时间	D0021	8000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 4	预置值	D0022, D0023	1000000	1-100,000,000
步骤 5	控制方向	D0024		
步骤 6	变化时间	D0024	替换后	
	下一步骤编号	D0025	4	步骤1-6

功能	设备地址	设置值	详情
目标频率	D0018、D0019	50000	50,000Hz
频率更改时间	D0021	8000	8,000ms
预置值	D0022、D0023	1000000	预置值=1,000,000
控制方向	D0024	—	—
变化时间	D0024	替换后	替换后=2
下一步骤编号	D0025	4	步骤 4



## 步骤 4 设置

步骤	功能	DR	设置	说明
步骤 1	目标频率	D0026, D0027	5000	15 ~ 100,000, 增量为 1 Hz
步骤 2	频率更改时间	D0029	8000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 3	预置值	D0030, D0031	1000000	1-100,000,000
步骤 4	控制方向	D0032		
步骤 5	变化时间	D0032	替换后	
步骤 6	下一步骤编号	D0033	5	步骤 1-6

功能	设备地址	设置值	详情
目标频率	D0026、D0027	5000	5,000Hz
频率更改时间	D0029	8000	8,000ms
预置值	D0030、D0031	1000000	预置值 =1,000,000
控制方向	D0032	—	—
变化时间	D0032	替换后	替换后 =2
下一步骤编号	D0033	5	步骤 5

## 步骤 5 设置

步骤	功能	DR	设置	说明
步骤 1	目标频率	D0034, D0035	15	15 ~ 100,000, 增量为 1 Hz
步骤 2	频率更改时间	D0037	8000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 3	预置值	D0038, D0039	100000	1-100,000,000
步骤 4	控制方向	D0040		
步骤 5	变化时间	D0040	替换后	
步骤 6	下一步骤编号	D0041	0	步骤 1-6

功能	设备地址	设置值	详情
目标频率	D0034、D0035	15	15Hz
频率更改时间	D0037	8000	8,000ms
预置值	D0038、D0039	100000	预置值 =100,000
控制方向	D0040	—	—
变化时间	D0040	替换后	替换后 =2
下一步骤编号	D0041	0	0= 结束输出

## 18: 脉冲输出指令

### 步骤 6 设置

ARAMP(Ramp脉冲输出表)

步骤	功能	DR	设置	说明
步骤 1	目标频率	D0042, D0043	15	15 ~ 100,000, 增量为 1 Hz
步骤 2	频率更改时间	D0045	5000	10 ~ 10,000, 增量为 1 (毫秒)
步骤 3	预置值	D0046, D0047	30000	1~100,000,000
步骤 4	控制方向	D0048		
步骤 5	变化时间	D0048	替换前	
步骤 6	下一步骤编号	D0049	0	步骤 1-6

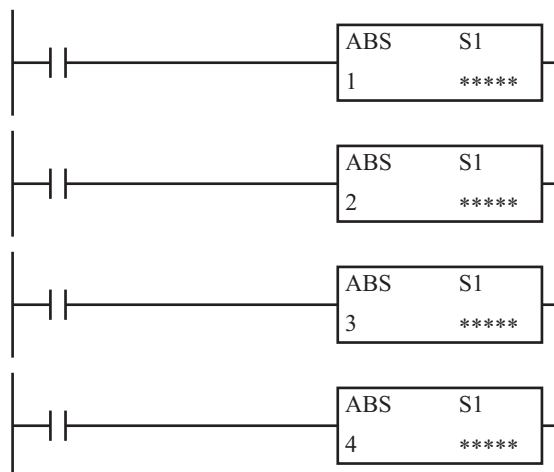
功能	设备地址	设置值	详情
目标频率	D0042、 D0043	15	15Hz
频率更改时间	D0045	5000	5000ms
预置值	D0046、 D0047	30000	预置值 =30,000
控制方向	D0048	—	—
变化时间	D0048	替换前	替换前 =0
下一步骤编号	D0049	0	步骤 0 (结束)

## ABS（绝对位置设置）

初始化脉冲输出用的绝对位置计数器。

FT2J FC6A\*

※ 仅晶体管输出类型



### 操作

当输入打开时，将特殊数据寄存器 D8240 ~ D8247（绝对位置计数器）的值更新为 S1 指定的初始值。更新后，将 1 存储在特殊数据寄存器 D8239（绝对位置管理状态）的绝对位置计数器初始化已完成标记中。

### 注释：

- 执行 RAMP、ARAMP、ZRN、JOG 指令的过程中，切勿执行 ABS 指令。绝对位置计数器可能无法初始化预期值。
- 用户程序执行开始后，请进行编程，以便在执行指定绝对位置的 RAMP（台形控制）指令前执行 ABS 指令。
- 在同一次扫描内执行多个 ABS 指令时，最后执行的 ABS 指令设置将反映在绝对位置计数器中。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- ABS 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 ABS 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	初始值	—	—	—	—	—	—	X	—	X	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 18: 脉冲输出指令

### 设置

#### ■ “设备”选项卡



#### 1. 选择指令

可从“ABS1”、“ABS2”、“ABS3”、“ABS4”中选择使用的 ABS 指令。

初始化目标的绝对位置计数器因指令而异。

指令	绝对位置计数器	特殊数据寄存器 *1*2*3	说明	更新绝对位置计数器的指令 *4
ABS1	绝对位置计数器 1	高位字 D8240	-2147483648 ~ 2147483647	ABS1 / RAMP1 *5 / ARAMP1 / ZRN1 / JOG1
		低位字 D8241		
ABS2	绝对位置计数器 2	高位字 D8242		
		低位字 D8243		
ABS3	绝对位置计数器 3	高位字 D8244		
		低位字 D8245		
ABS4	绝对位置计数器 4	高位字 D8246		
		低位字 D8247		

\*1 高位和低位数据寄存器根据 32 位数据存储设置而变化。有关详情，请参见第 3-10 页上的“32 位数据存储”。

\*2 即使电源关闭，也可通过备份电池保持绝对位置计数器的值。产生保持数据错误时，初始化为 0。

\*3 特殊数据寄存器为只读。变更 D8240 ~ D8247 的值时，请使用 ABS 指令。

\*4 在“禁用”下使用 RAMP、ARAMP、ZRN、JOG 指令时，不更新绝对位置计数器。

\*5 使用 RAMP 指令启用绝对位置指定模式时，可通过绝对位置指定目标位置。要通过 RAMP 指令使用绝对位置指定模式时，请先使用 ABS 指令对相应的绝对位置计数器进行初始化。有关详情，请参见第 18-17 页上的“RAMP（台形控制）”。

有关 D8240 ~ D8247 的初始化状态，可在定义为 D8239（绝对位置管理状态）的绝对位置计数器初始化已完成标记中确认。

FC6A 型从停止切换为运行时，将 0（未初始化）存储在绝对位置初始化已完成标记中。接着，使用 ABS 指令对绝对位置计数器进行初始化后，存储 1（初始化已完成）。

设备编号	参数名称	位位置	说明
D8239	绝对位置管理状态	bit0: 绝对位置计数器 1 初始化完成标记	0: 未初始化 1: 初始化完成
		bit1: 绝对位置计数器 2 初始化完成标记	
		bit2: 绝对位置计数器 3 初始化完成标记	
		bit3: 绝对位置计数器 4 初始化完成标记	
		bit4 ~ bit15	保留

## 2. S1（源 1）：初始值

指定用于存储初始值的设备的第一数据寄存器或常量。

从指定的数据寄存器开始，使用连续的 2 位数据寄存器。

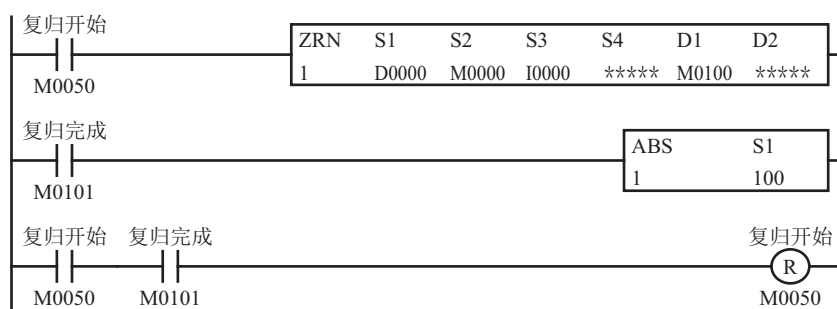
请指定设备范围内的第一数据寄存器。数据类型为 L（长整）。

存储目的地	设置	
起始编号 +0	初始值（高位字）*1	-2147483648 ~ 2147483647
起始编号 +1	初始值（低位字）*1	

\*1 高位和低位数据寄存器根据 32 位数据存储设置而变化。有关详情，请参见第 3-10 页上的“32 位数据存储”。

## 操作示例

### 梯形图



电源打开后启动 ZRN1 指令，进行归零。

归零操作完成后，通过 ABS1 指令将绝对位置计数器 1 初始化为 100。

### 设置



功能	预置值
选择指令	ABS1
标记名称	100（常量）

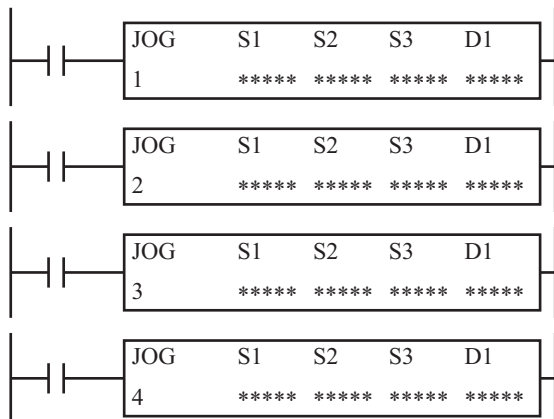
## 18: 脉冲输出指令

### JOG (JOG 运行)

输出频率变化的脉冲。

FT2J FC6A\*

※ 仅晶体管输出类型



#### 操作

输入打开时，若 S3 指定的 JOG 输入打开，将输出 S1 指定的启动频率的脉冲，并以固定比率加大脉冲直至达到目标频率。达到目标频率后，输出一定频率的脉冲。

S3 指定的 JOG 输入关闭后，以固定比率减小直至达到启动频率，达到启动频率后停止输出脉冲。

将脉冲控制信息（正在输出 / 输出结束 / 错误）在 D1 指定的内部继电器中存储为操作状态。

在 S2 中指定的初始化输入打开时，将“设置”选项卡上 WindLDR“JOG（JOG 控制）”对话框中配置的初始值存储到控制寄存器中。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当执行与正在执行的脉冲输出指令使用相同输出的另一条脉冲输出指令时，随后执行的脉冲输出指令将被取消。在这种情况下，将存储错误代码 48 在用户程序执行错误代码 (D8006) 中。
- JOG 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。
- 继电器输出型执行 JOG 指令时，将存储错误代码 19 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	▲	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
S3 (源 3)	JOG 输入	X	—	X	—	—	—	—	—	—	—
D1 (目标 1)	操作状态	—	—	▲	—	—	—	—	—	—	—

▲无法使用特殊数据寄存器。无法使用特殊内部继电器。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 设置

## ■ “设备”选项卡



## 1. 选择指令

可从“JOG1”、“JOG2”、“JOG3”、“JOG4”中选择使用的 JOG 指令。

输出、可选择的方向控制模式因指令和 CPU 模块类型而异。

有关指令、方向控制模式和脉冲输出模式的组合限制，请参见第 18-75 页上的“10. 启用方向控制”。

## 2. S1（源 1）：控制寄存器

S1 指定用于 JOG1、JOG2、JOG3、JOG4 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始，使用连续的 12 位数据寄存器。

请指定设备范围内的第一数据寄存器。

存储目的地	功能	设置		参考
		All-in-One CPU 模块	CAN J1939 All-in-One CPU 模块 /Plus CPU 模块	
起始编号 +0	目标频率（高位字）*1	JOG1、JOG2: 15 ~ 10,000 （增量为 1Hz）	JOG1 ~ JOG4: 15-10,000 （增量为 1Hz）	第 18-75 页上的“6. 稳定脉冲频率”
起始编号 +1	目标频率（低位字）*1	JOG3、JOG4: 15-5,000 （增量为 1Hz）		
起始编号 +2	启动频率（高位字）*1	JOG1、JOG2: 15-10,000 （增量为 1Hz）	JOG1 ~ JOG4: 15-10,000 （增量为 1Hz）	第 18-75 页上的“7. 归零速度脉冲频率”
起始编号 +3	启动频率（低位字）*1	JOG3、JOG4: 15-5000 （增量为 1Hz）		
起始编号 +4	加大时间	10-10,000 (ms)		第 18-75 页上的“8. 加速时间”
起始编号 +5	频率减小时间	10-10,000 (ms)		第 18-75 页上的“9. 减速时间”
起始编号 +6	控制方向	0: 正向 1: 反向		第 18-76 页上的“11. 控制方向”
起始编号 +7	保留			
起始编号 +8	当前值（高位字）*1	1-100,000,000（脉冲）*2		第 18-76 页上的“12. 当前值”
起始编号 +9	当前值（低位字）*1			
起始编号 +10	错误状态	0 ~ 3		第 18-76 页上的“13. 错误状态”
起始编号 +11	保留			

\*1 高位和低位数据寄存器根据 32 位数据存储设置而变化。有关详情，请参见第 3-10 页上的“32 位数据存储”。

\*2 脉冲输出开始时，当前值复位为 0。

## 18: 脉冲输出指令

### 3. S2（源 2）：初始化输入

S2 指定初始化输入。

当初始化输入打开后，“设置”选项卡上 WindLDR 的“JOG（JOG）”对话框中配置的初始值将存储到控制寄存器中。可指定外部输入或内部继电器。

当初始化输入打开时，初始值将在每次扫描时存储到数据寄存器中。（即使在不执行 JOG 指令（未打开）的状态下，若打开初始化输入，初始值也将存储在数据寄存器中。）要对值仅初始化一次，请将初始化输入与 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令结合使用。

### 4. S3（源 3）：JOG 输入

S3 指定 JOG 输入。

打开 JOG 输入后，输出启动频率脉冲并加大，直至达到目标频率。

当 JOG 输入关闭，减小脉冲直至达到启动频率后，停止输出脉冲。

可指定外部输入或内部继电器。

在加大过程中关闭 JOG 输入，减小脉冲直至达到启动频率后，停止输出脉冲。

在减小过程中打开 JOG 输入，加大脉冲直至达到目标频率。

### 5. D1（目标 1）：运行状态

D1 指定用于 JOG 指令的内部继电器的第一内部继电器。

从指定的继电器开始，使用 4 个连续的继电器。

请指定设备范围内的第一内部继电器。

存储目的地	功能	设置	
起始编号 +0	脉冲输出打开	0: 脉冲输出关闭 1: 脉冲输出打开	在脉冲输出过程中打开。 脉冲输出停止后关闭。
起始编号 +1	脉冲输出完成	0: 脉冲输出未完成 1: 脉冲输出完成	脉冲输出完成后打开。 脉冲输出开始后关闭。
起始编号 +2	脉冲输出状态	0: 恒速状态 1: 频率变化状态	脉冲输出状态为恒速状态时打开。 脉冲输出状态为变化状态时关闭。
起始编号 +3	保留		



## ■ “设置”选项卡



### 6. 稳定脉冲频率

指定变化后恒速状态的频率。设于 15Hz ~ 100kHz（增量为 1Hz）之间。输出频率误差限于 ±5%。  
All-in-One CPU 模块的 JOG3、JOG4 设于 15Hz ~ 5kHz（增量为 1Hz）之间。

### 7. 归零速度脉冲频率

指定开始输出脉冲时的频率。设于 15Hz ~ 100kHz（增量为 1Hz）之间。输出频率误差限于 ±5%。  
All-in-One CPU 模块的 JOG3、JOG4 设于 15Hz ~ 5kHz（增量为 1Hz）之间。

### 8. 加速时间

指定加大脉冲的时间。设于 10 ~ 10,000ms（增量为 1ms）之间。预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。

### 9. 减速时间

指定减小脉冲的时间。设于 10 ~ 10,000ms（增量为 1ms）之间。预置值的个位作为 0 处理。例如，输入 144 时，预置值作为 140ms 处理。

### 10. 启用方向控制

从以下方向控制模式中选择有 / 无方向控制，及方向控制的方法。脉冲输出模式包括单脉冲输出模式和双脉冲输出模式，与有 / 无方向控制的组合使用如下所示。（以 All-in-One CPU 模块使用 JOG1 时为例。）

启用方向控制	动作	模式
禁用	当单向使用脉冲输出时选择此选项。脉冲 A 和 B 可独立使用。	Q0
方向控制“单脉冲输出”	脉冲 A 用作脉冲输出。脉冲 B 打开 / 关闭用作方向控制。	Q0 Q2
方向控制“双脉冲输出”	脉冲 A 用作正向脉冲 (CW) 输出。脉冲 B 用作反向脉冲 (CCW) 输出。	Q0 Q1

使用的输出因使用的指令、脉冲输出模式和方向控制结合以及使用的型号而异。

指令	操作条件	使用的输出			
		All-in-One CPU 模块		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	
		脉冲输出	方向控制输出	脉冲输出	方向控制输出
JOG1	禁用方向控制	Q0	—	Q0	—
	方向控制“单脉冲输出”	Q0	Q2 <sup>*1</sup>	Q0	Q1
	方向控制“双脉冲输出”	Q0, Q1 <sup>*2</sup>	—	Q0、Q1	—
JOG2	禁用方向控制	Q1	—	Q2	—
	方向控制“单脉冲输出”	Q1	Q3 <sup>*1</sup>	Q2	Q3
	方向控制“双脉冲输出”	—	—	Q2、Q3	—

## 18: 脉冲输出指令

指令	操作条件	使用的输出			
		All-in-One CPU 模块		CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	
		脉冲输出	方向控制输出	脉冲输出	方向控制输出
JOG3	禁用方向控制	Q2	—	Q4	—
	方向控制“单脉冲输出”	—	—	Q4	Q5
	方向控制“双脉冲输出”	—	—	Q4、Q5	—
JOG4	禁用方向控制	Q3	—	Q6	—
	方向控制“单脉冲输出”	—	—	Q6	Q7
	方向控制“双脉冲输出”	—	—	Q6、Q7	—

\*1 使用 All-in-One CPU 模块的单脉冲输出模式时，因使用 Q2 或 Q3，不可同时执行使用同一输出的指令。

\*2 使用 All-in-One CPU 模块的双脉冲输出模式时，因使用 Q1，不可同时执行使用同一输出的指令。

### 11. 控制方向

启用方向控制后，系统将在用于正向操作的数据寄存器中存储 0，在用于反向操作的数据寄存器中存储 1。

### 12. 当前值

脉冲输出数将存储在数据寄存器中。

执行 JOG 指令时，每次扫描都会更新当前值。

脉冲输出开始时，当前值复位为 0。

### 13. 错误状态

若配置的设置有误，则输出有误设置对应的错误代码。如果各步骤开始执行时，执行的步骤出现配置错误，将导致用户程序执行错误，在 D8006 中存储错误代码 20。

#### 错误代码一览

错误代码	状态	说明		
0	正常	—		
2	启动频率指定错误	All-in-One CPU 模块	JOG1、JOG2	启动频率未设置在 15 到 100,000 之间。
			JOG3、JOG4	启动频率未设置在 15 到 5,000 之间。
4	目标频率指定错误	CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	JOG1 ~ JOG4	启动频率未设置在 15 到 100,000 之间。
		All-in-One CPU 模块	JOG1、JOG2	目标频率未设置在 15 到 100,000 之间。
			JOG3、JOG4	目标频率未设置在 15 到 5,000 之间。
5	频率更改时间指定错误	CAN J1939 All-in-One CPU 模块 / Plus CPU 模块	JOG1 ~ JOG4	目标频率未设置在 15 到 100,000 之间。
7	控制方向指定错误	频率更改时间未设置在 10 ~ 10,000 之间。		
9	频率指定错误	控制方向未设置为 0 或 1。		
		将启动频率设为与目标频率相同或大于目标频率的值。*1		

\*1 请将启动频率设为低于目标频率。



## 18: 脉冲输出指令

### 设置

?
×

设备
设置

JOG 1 (Q0000)
S1 (控制寄存器)
S2 (初始化输入)
S3 (JOG 输入)
D1 (运行状态)

标记名称:
D0000
M0000
I0000
M0010

设备地址:
D0000
M0000
I0000
M0010

?
×

设备
设置

功能	DR	设置	说明
稳定脉冲频率	D0000, D0001	300	15 ~ 100,000 Hz, 增量为 1 Hz
归零速度脉冲频率	D0002, D0003	50	15 ~ 100,000 Hz, 增量为 1 Hz
加速时间	D0004	100	10 ~ 10,000 ms, 增量为 10 ms
减速时间	D0005	100	10 ~ 10,000 ms, 增量为 10 ms
启用方向控制		单脉冲输出	
控制方向	D0006	正向	
当前值	D0008, D0009		1 ~ 100,000,000
错误状态	D0010		

确定
取消

## 脉冲监控

输出具有频率变化功能的脉冲。

FT2J FC6A

※ 仅 Plus CPU 模块

### 功能说明

脉冲监控是 FC6A 型以脉冲输出指令监控从各输出端口输出的脉冲，将轨迹及各轴的波形进行图显示的功能。

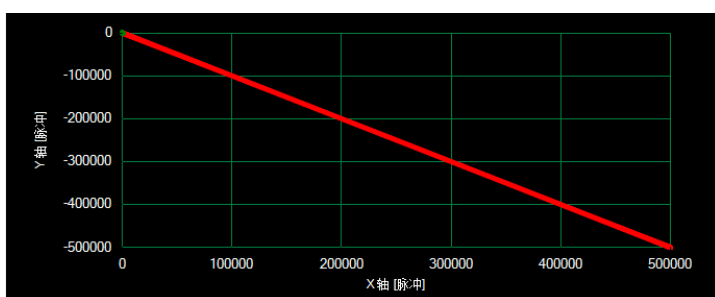
#### 注释:

- 由于 WindLDR 会以固定周期获取数据，因此需要在 PLC 内部累积数据。相较于 WindLDR 的读取速度，PLC 的累积速度更快，如果内存的数据被覆盖，则会中断脉冲监控。
- 取决于脉冲监控的取样周期或通信环境，绘制图表的时间可能慢于 PLC 动作。

#### 轨迹

根据指定的输出端口对应的脉冲数显示平面上的轨迹。

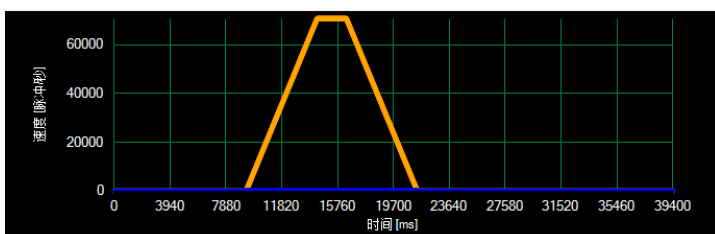
仅在绝对位置指定模式下动作时可显示轨迹图。



#### 波形

指定输出端口脉冲输出速度的显示图表。

横轴表示脉冲输出时间（开始脉冲监控后的相对时间），纵轴表示输出时间对应的脉冲数。



### 启动脉冲监控

#### 操作过程

1. 在“联机”选项卡的“监控”中单击“监控”。

监控启动。

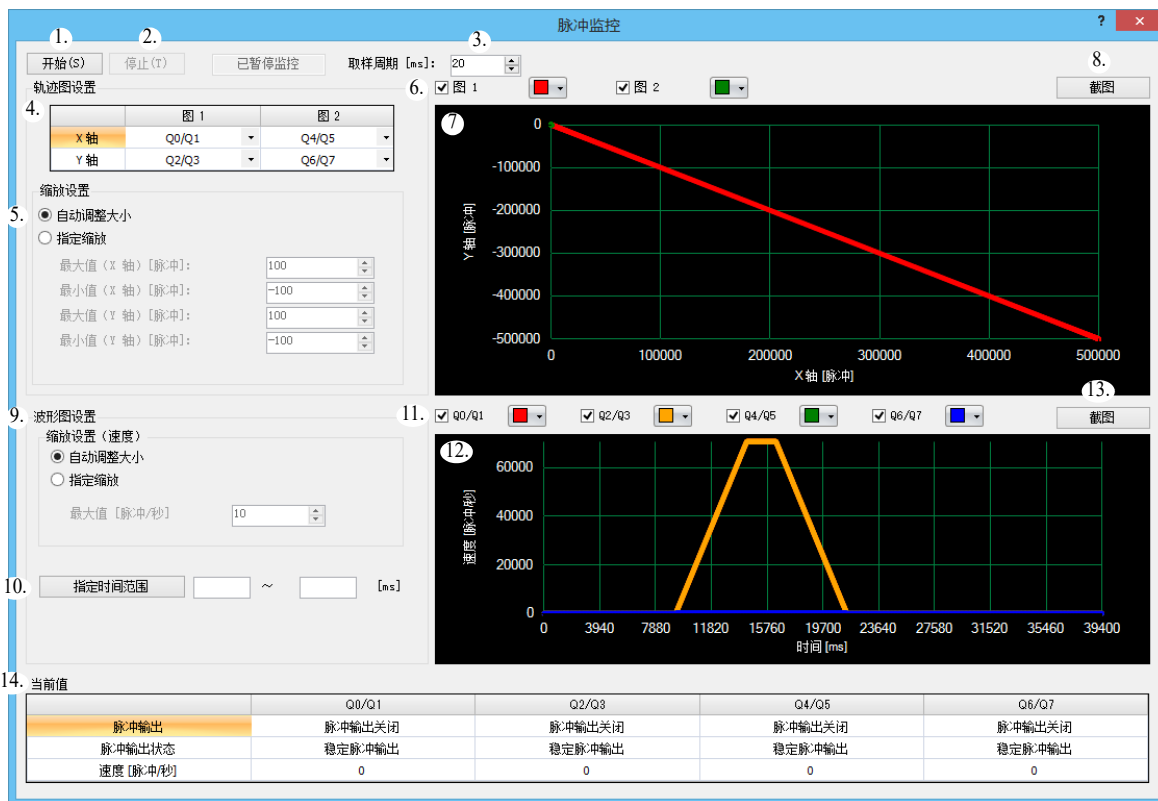


## 18: 脉冲输出指令

- 在“联机”选项卡的“监控”中单击“脉冲监控”。
- 将显示“脉冲监控”对话框。



### “脉冲监控”对话框



#### 1. 开始

单击此按钮，开始脉冲监控。WindLDR 会以固定周期通过 PLC 获取数据。

#### 2. 停止

单击此按钮，停止脉冲监控。

#### 3. 取样频率

指定取样频率。指定为 5 ~ 255ms（增量为 1ms）之间。

#### 4. 输出端口设置

从以下选项中选择显示在轨迹图中的 X 轴及 Y 轴的输出端口。所选的端口也可在开始脉冲监控后更改。

X 轴：“Q0/Q1”、“Q2/Q3”、“Q4/Q5”、“Q6/Q7”

Y 轴：“Q0/Q1”、“Q2/Q3”、“Q4/Q5”、“Q6/Q7”

#### 5. 缩放设置

将轨迹图的 X 轴及 Y 轴的范围选择为“自动调整大小”或“指定缩放”。

自动调整大小：根据数据的最大值自动刷新图表的最大值。

指定缩放：手动输入各轴的最大值及最小值。如果数值大于最大值、小于最小值，则无法显示图表。

#### 6. 图表设置

选中显示在轨迹图中的图表的复选框，选择图表的颜色。

### 7. 轨迹图

可绘制轨迹图。光标会显示在图表上，并可与波形图连动显示。

### 8. 截图

将轨迹图保存为图像。

单击“截图”按钮，将显示“文件选择”对话框，即可保存为位图文件。

### 9. 缩放设置（速度）

将波形图的速度（Y轴）刻度选择为“自动调整大小”或“指定缩放”。

自动调整大小：根据数据的最大值自动刷新图表的最大值。

指定缩放：手动输入速度（Y轴）的最大值。如果数值大于最大值，则无法显示图表。

### 10. 指定时间范围（时间）

以开始监控后的相对时间设置想要显示的图表范围。

图表将按照设置放大或缩小显示。

### 11. 图表设置

选中显示在波形图中的输出端口的复选框，选择图表的颜色。

### 12. 波形图

可绘制波形图。显示滚动条，并显示一定期间的数据。移动滚动条，可显示过去的的数据。

### 13. 截图

将波形图保存为图像。

单击“截图”按钮，将显示“文件选择”对话框，即可保存为位图文件。

### 14. 当前值

显示波形图的当前值。

脉冲输出：显示“脉冲输出打开”或“脉冲输出关闭”。

脉冲输出状态：显示“稳定脉冲输出”或“频率变化状态”。

速度：显示输出的脉冲速度。

定位控制

FT2J FC6A

本节将对定位控制的概要、WindLDR 设置、配线图及实用示例进行介绍。

定位控制的概要

通过脉冲输出指令可进行定位控制。

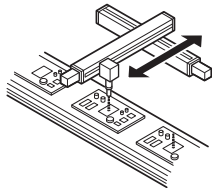
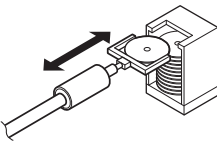
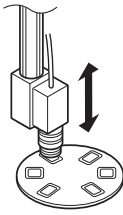
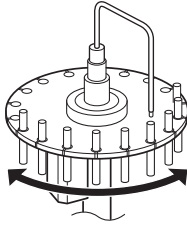
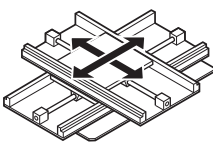
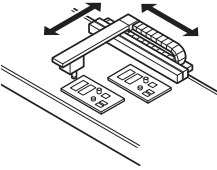
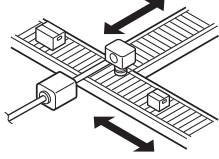
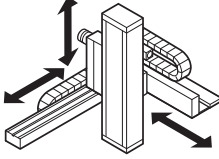
All-in-One CPU 模块最多可同时控制 2 轴。

Plus CPU 模块、CAN J1939 All-in-One CPU 模块最多可同时控制 4 轴。

脉冲输出指令示例

指令	运行内容
RAMP	以固定频率更改比率进行台形控制的定位指令。
RAMPL	针对目标位置同时控制 2 轴的定位指令。
ZRN	返回原点使其在下次从原点开始运行的指令。
ARAMP	可进行多级速度控制，并以中断输入为契机更改速度的指令。
ABS	绝对位置计数器的初始化指令。

应用示例

概要	说明			
轴数	1 轴			
应用示例	拾放 	收纳部件 	调整对焦位置 	旋转工作台 
适用指令	RAMP、ARAMP、ZRN			
概要	说明			
轴数	2 轴		3 轴	
应用示例	XY 轴平台 	检查图像 	分拣搬运品 	XYZ 轴平台 
适用指令	RAMPL+ABS ZRN		RAMP+ABS ZRN	
			RAMPL+RAMP+ABS RAMP+ABS ZRN	



## 脉冲输出指令设置项目

## 启用方向控制

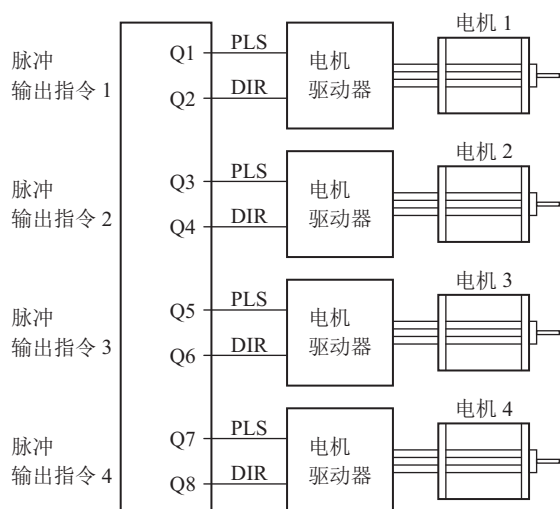
方向控制模式是控制电机旋转方向的模式。可通过脉冲输出指令（RAMP/RAMPL/ZRN/ARAMP）使用。方向控制模式包括禁用 / 单脉冲输出模式 / 双脉冲输出模式 3 种模式。

有关运行内容，请参见各指令的说明。

**注释：**如果 PLC 与电机驱动器中的设置不同，则无法正确控制电机，如电机只朝一个方向动作等。

## Plus CPU 模块 / CAN J1939 All-in-One CPU 模块示例

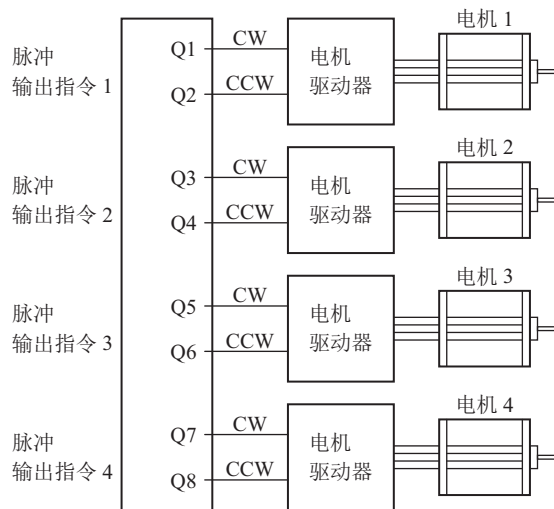
## 单脉冲输出



PLS: 脉冲输出

DIR: 方向制御

## 双脉冲输出



CW: 正向脉冲

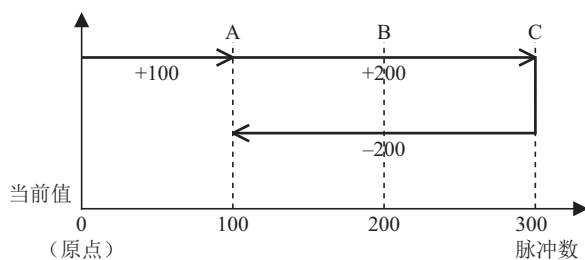
CCW: 反向脉冲

## 绝对位置模式

绝对位置模式是以绝对位置指定脉冲位置的模式。可通过脉冲输出指令（RAMP/RAMPL）使用。启用绝对位置模式后，能够以绝对位置指定脉冲输出指令中设置的目标位置。有关运行内容，请参见各指令的说明。

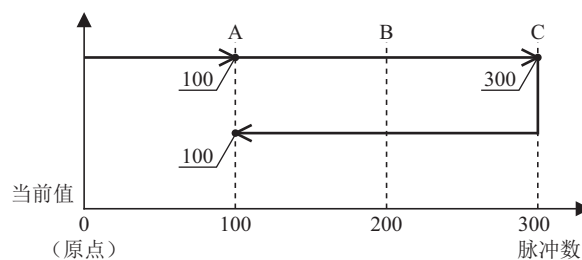
## 禁用

动作方向	脉冲数	正向、反向
点 A → 点 B	100	正向
点 B → 点 C	200	正向
点 C → 点 B	200	反向



## 启用

动作方向	脉冲数 (目标位置)
点 A → 点 B	100
点 B → 点 C	300
点 C → 点 B	100

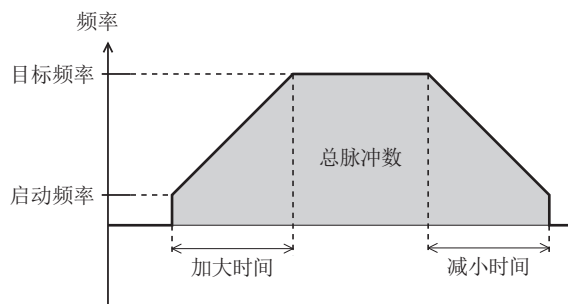


## 18: 脉冲输出指令

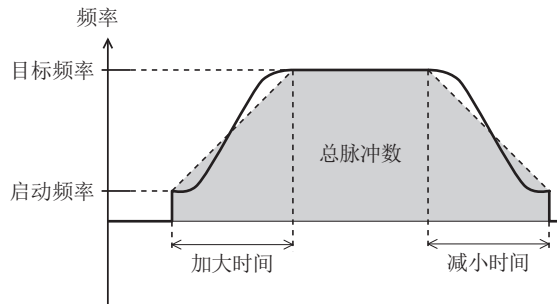
### 频率变化曲线

频率变化曲线是控制频率变化频率的功能。可通过脉冲输出指令（RAMP）使用。包括直线/S形曲线2种设置。直线会在启动和停止时以固定比率进行加减速，以吸收启动和停止时的冲击。S形曲线会在启动和停止时减小加速度，藉此减小冲击，减轻振动。有关运行内容，请参见各指令的说明。

#### 直线

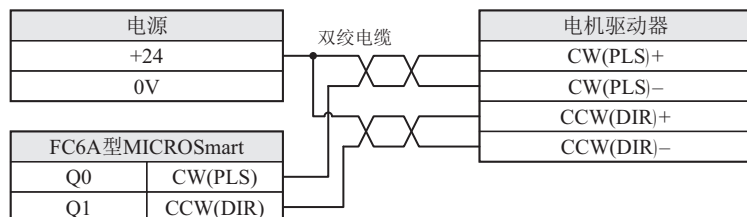


#### S形曲线

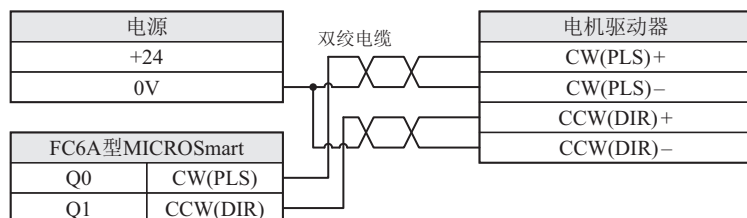


### 配线图（FC6A型 ↔ 电机驱动器）

#### FC6A型：晶体管沉型输出类型



#### FC6A型：晶体管保护源型输出类型



**注意**

请绝对不要逆接电源。否则可能导致破损。  
配线时，请仔细阅读电机驱动器手册后再行作业。

## 实用示例

## 使用 RAMP 指令的 1 轴控制示例

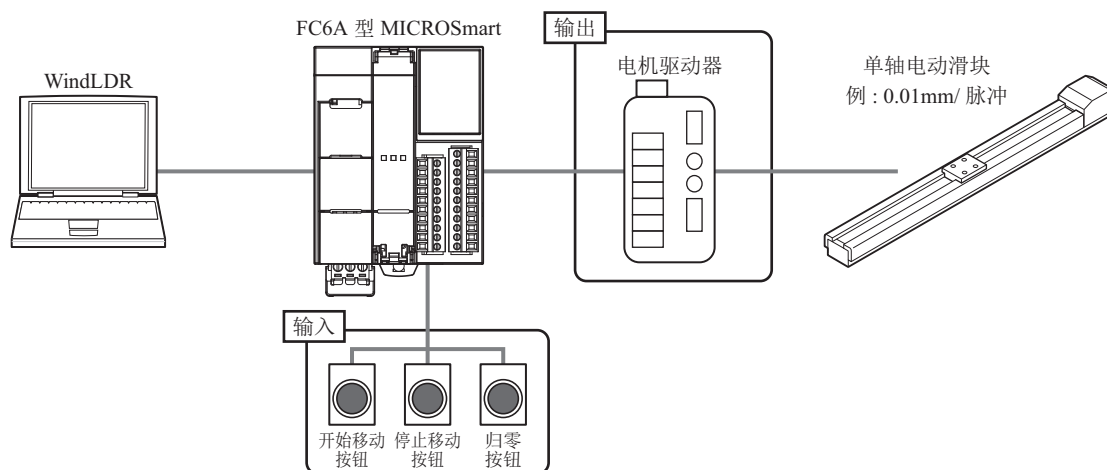
## ■ 用途

单轴电动滑块

## ■ 系统构成图

此处假设电动滑块的规格为“0.01mm/ 脉冲”。

如需了解您所使用的电动滑块移动距离，请参见电动滑块的规格（每个脉冲的移动量）。



## ■ 动作说明

长按停止移动按钮 3 秒钟，执行 ABS 指令，确定原点。

按下开始移动按钮，按照点 A → 点 C → 点 B → 点 A 的方向移动。第 2 次到达点 A 时停止。

按下停止移动按钮，停止移动中的电动滑块。

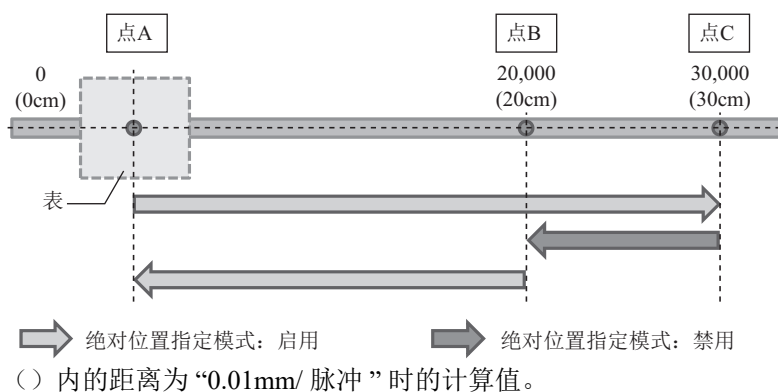
按下归零按钮，从停止位置开始向原点（点 A）返回。

通过启用绝对位置模式执行点 A → 点 C 之间、点 B → 点 A 之间的移动。

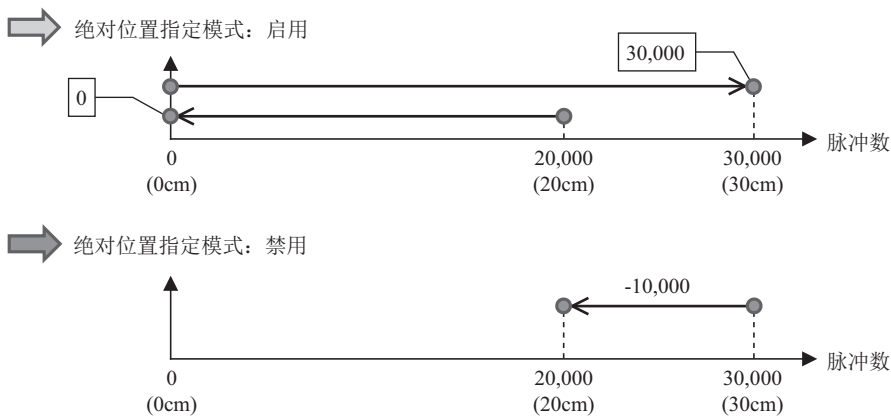
通过禁用绝对位置模式执行点 C → 点 B 之间的移动。

由于将频率变化曲线设为“S 形曲线”后会减少初期加速，因此可抑制冲击或振动。

位置	X (cm)
点 A	0
点 B	20
点 C	30



# 18: 脉冲输出指令

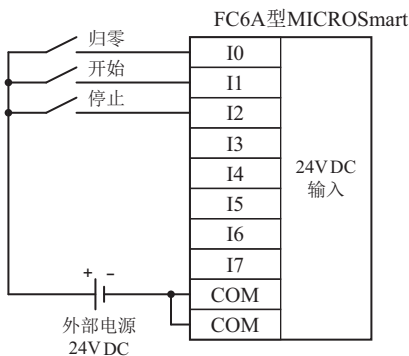


( ) 内的距离为“0.01mm/脉冲”时的计算值。

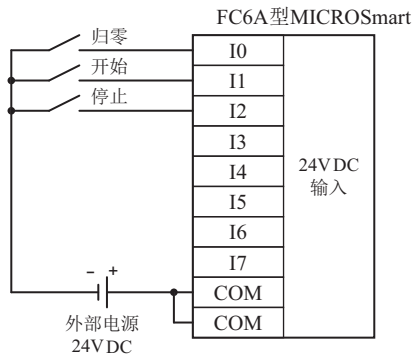
## ■ 配线图

### 输入端子

DC 沉型输入

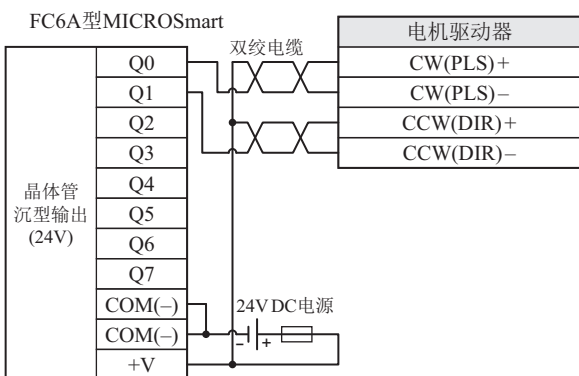


DC 源型输入

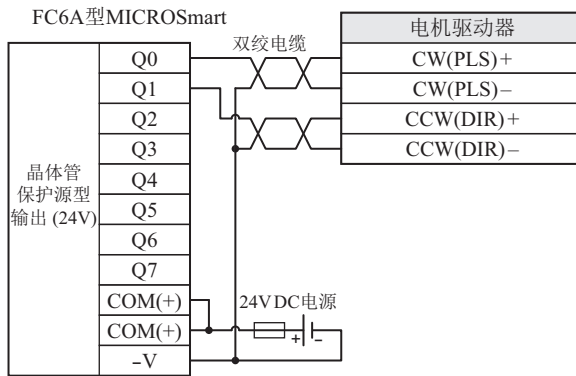


### 输出端子

晶体管沉型输出类型



晶体管保护源型输出类型

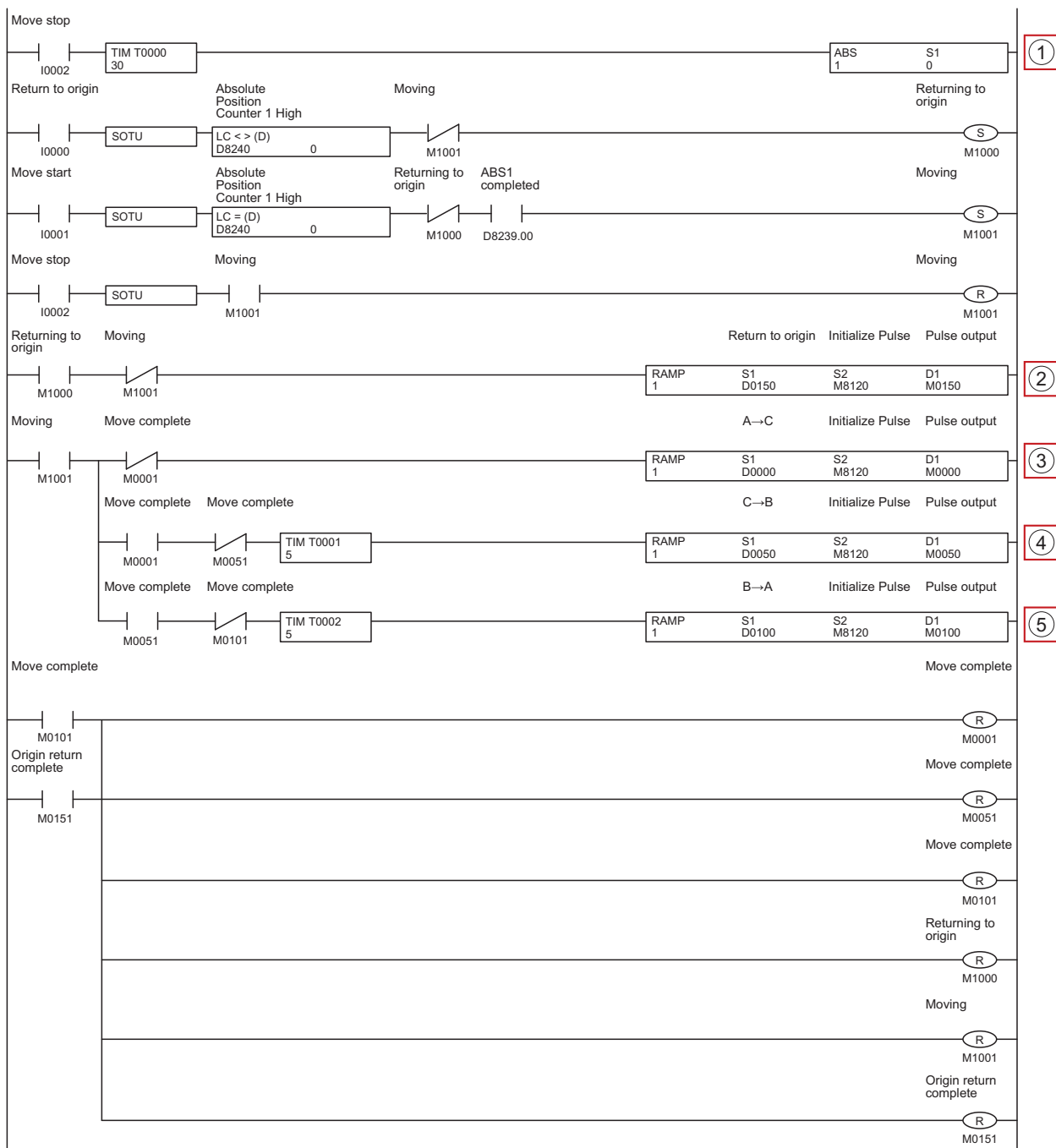


## ■ 分配地址

设备名称	单位	范围	说明
输入	位	I0	归零按钮
		I1	开始移动按钮
		I2	停止移动按钮
内部继电器	位	M10-M13	RAMP 指令运行状态
		M50-M53	
		M100-M103	
		M150-M153	
		M1000	归零位
		M1001	开始移动位
		M1002	停止移动位
特殊内部继电器		M8120	初始化脉冲
数据寄存器	位字	D10-D20	RAMP 指令控制寄存器
		D50-D60	
		D100-D110	
		D150-D160	
定时器		T0	定时器预设值
		T1	
计数器		C0	计数器预设值

# 18: 脉冲输出指令

## ■ 梯形图



编号	内容
(1)	原点定位
(2)	归零
(3)	点 A → 点 C 移动
(4)	点 C → 点 B 移动
(5)	点 B → 点 A 移动

## ■ 设置画面

## 点 A → 点 C

功能	DR	设置	
目标频率	D0000, D0001	10000	15
扫零速度脉冲频率	D0002, D0003	1000	15
频率时间	D0004	1000	10
频率变化曲线		S形曲线	
启用方向控制		单脉冲输出	
控制方向	D0005	正向	
绝对位置模式		启用	
预置值	D0006, D0007	30000	-2,
当前值	D0008, D0009		-2,
错误状态	D0010		

## 点 C → 点 B

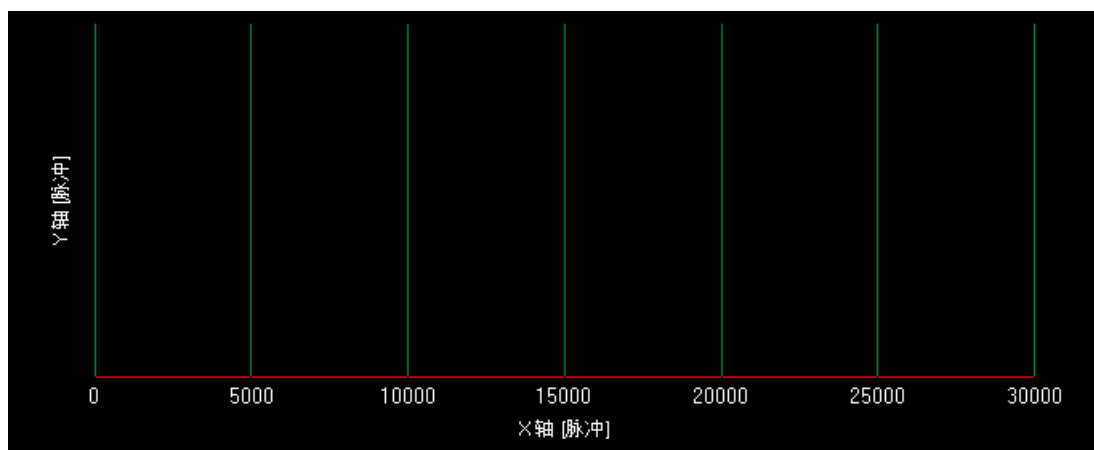
功能	DR	设置	
目标频率	D0050, D0051	6000	15
扫零速度脉冲频率	D0052, D0053	1000	15
频率时间	D0054	1000	10
频率变化曲线		S形曲线	
启用方向控制		单脉冲输出	
控制方向	D0055	反向	
绝对位置模式		禁用	
预置值	D0056, D0057	10000	1-
当前值	D0058, D0059		1-
错误状态	D0060		

## 点 B → 点 A

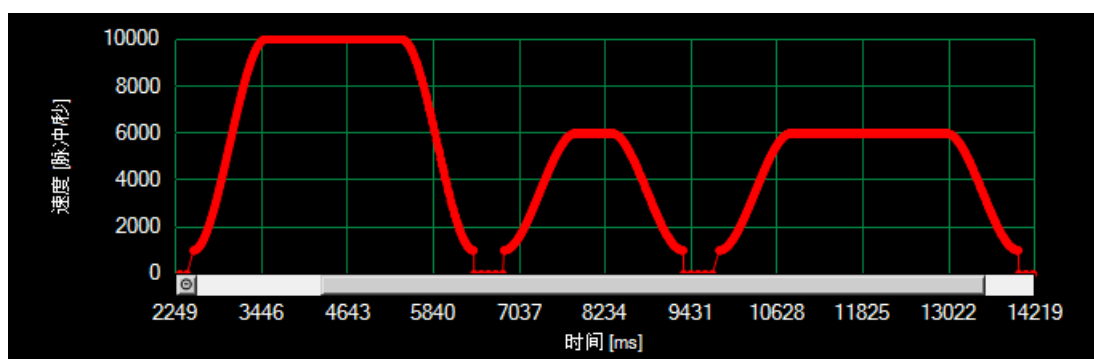
功能	DR	设置	
目标频率	D0100, D0101	6000	15
扫零速度脉冲频率	D0102, D0103	1000	15
频率时间	D0104	1000	10
频率变化曲线		S形曲线	
启用方向控制		单脉冲输出	
控制方向	D0105	正向	
绝对位置模式		启用	
预置值	D0106, D0107	0	-2,
当前值	D0108, D0109		-2,
错误状态	D0110		

## ■ 脉冲监控画面

## • 轨迹



## • 波形



## 18: 脉冲输出指令

### 使用 RAMPL 指令的 2 轴同时控制示例

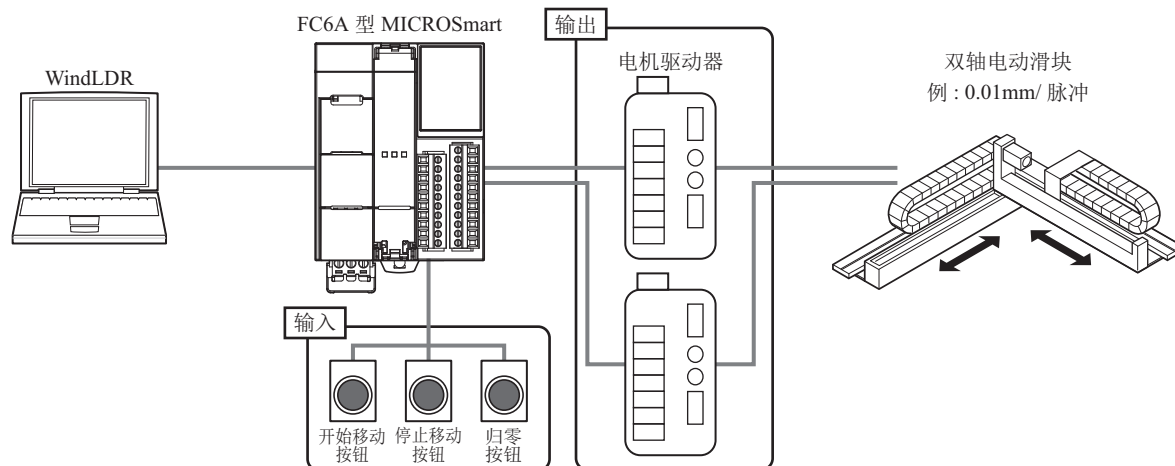
#### ■ 用途

双轴电动滑块

#### ■ 系统构成图

此处假设电动滑块的规格为“0.01mm/ 脉冲”。

如需了解您所使用的电动滑块移动距离，请参见电动滑块的规格（每个脉冲的移动量）。



#### ■ 动作说明

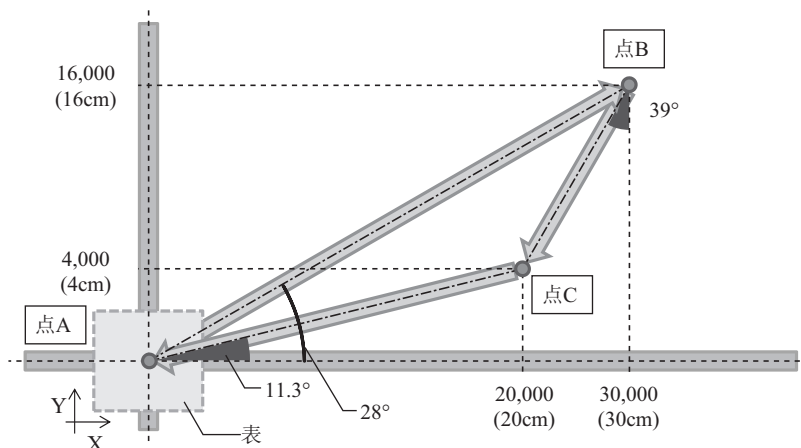
长按停止移动按钮 3 秒钟，执行 ABS 指令，确定原点。

按下开始移动按钮，按照点 A → 点 B → 点 C → 点 A 的方向移动。第 2 次到达点 A 时停止。

按下停止移动按钮，停止移动中的电动滑块。

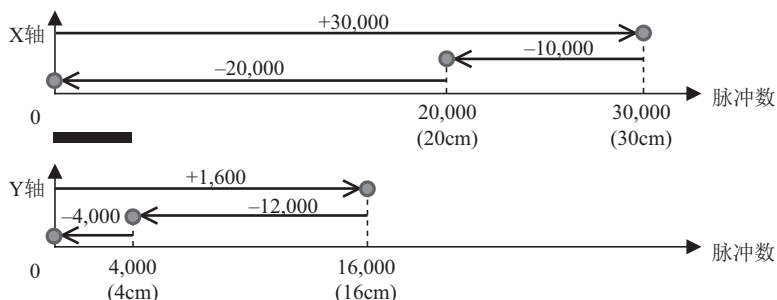
按下归零按钮，从停止位置开始向原点（点 A）返回。

位置	X (cm)	Y (cm)
点 A	0	0
点 B	30	16
点 C	20	4



( ) 内的距离为“0.01mm/ 脉冲”时的计算值。

➡ 绝对位置指定模式: 启用



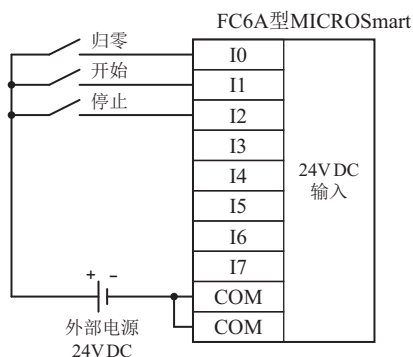
( ) 内的距离为“0.01mm/ 脉冲”时的计算值。



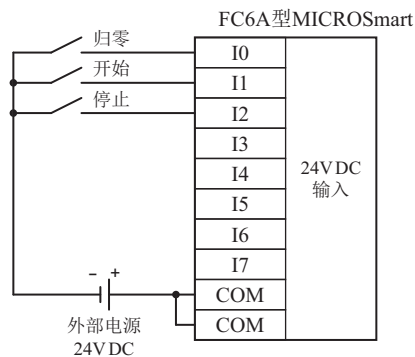
## ■ 配线图

### 输入端子

DC 沉型输入

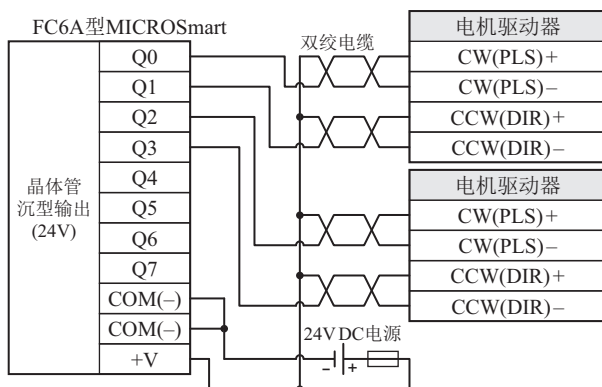


DC 源型输入

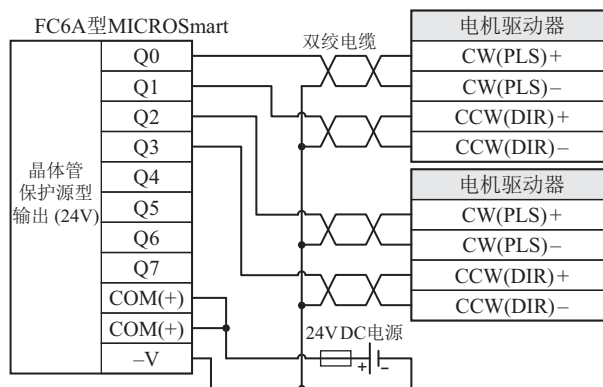


### 输出端子

晶体管沉型输出类型



晶体管保护源型输出类型



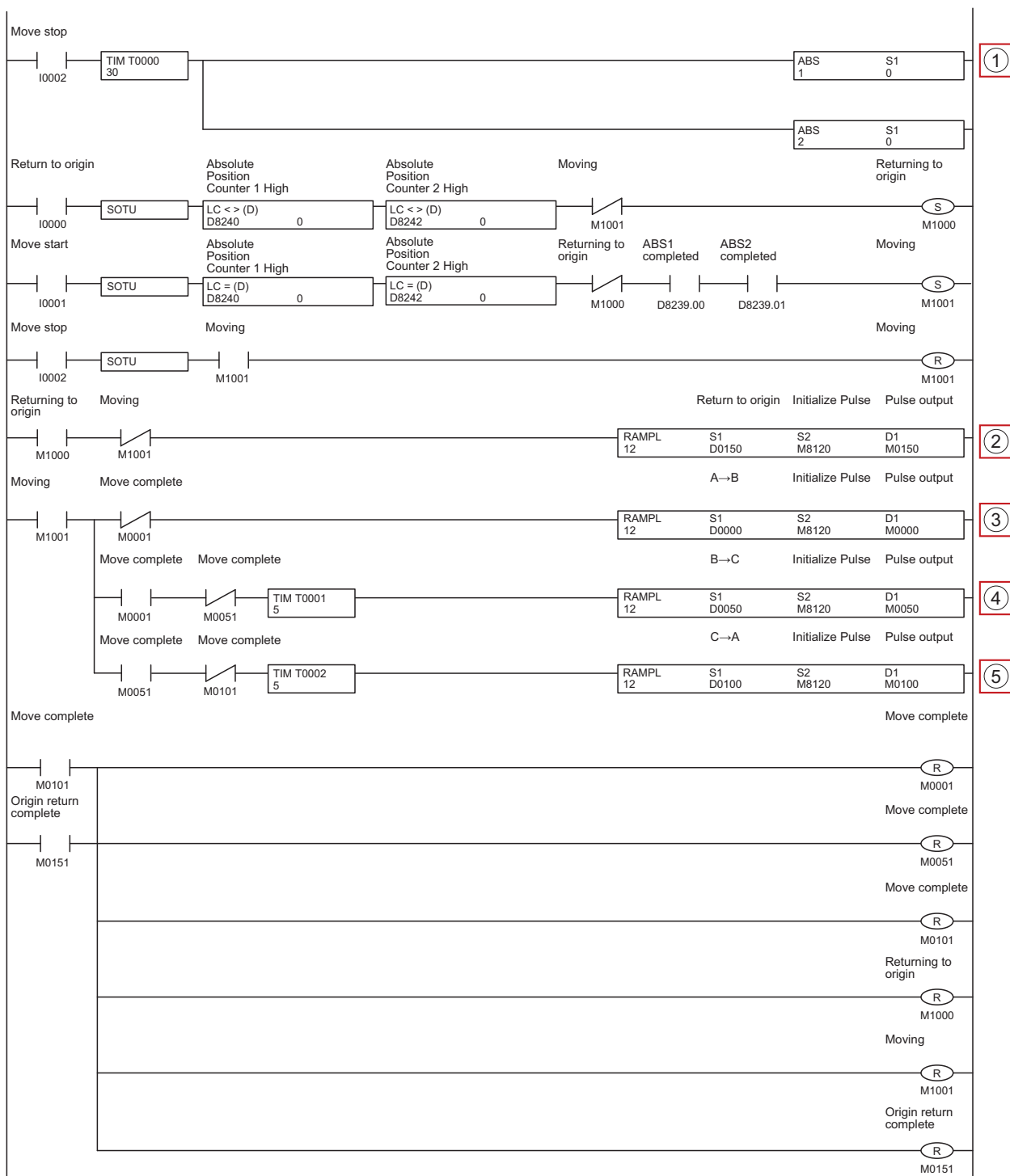
## ■ 分配地址

设备名称	单位	范围	说明
输入	位	I0	归零按钮
		I1	开始移动按钮
		I2	停止移动按钮
内部继电器	位	M10-M13	RAMPL 指令运行状态
		M50-M53	
		M100-M103	
		M150-M153	
		M1000	归零位
M1001	开始移动位		
M1002	停止移动位		
特殊内部继电器		M8120	初始化脉冲
数据寄存器	位字	D10-D39	RAMPL 指令控制寄存器
		D50-D79	
		D100-D129	
		D150-D179	
定时器		T0	定时器预设值
		T1	
计数器		C0	计数器预设值

## ■ 梯形图

# 18: 脉冲输出指令

## ■ 梯形图



编号	内容
(1)	原点定位
(2)	归零
(3)	点 A → 点 B 移动
(4)	点 B → 点 C 移动
(5)	点 C → 点 A 移动

## ■ 设置画面

### 常规设置

功能	DR	设置
合成目标频率	D0010, D0011	6000
合成启动频率	D0012, D0013	1000
频率更改时间	D0014	500
错误状态	D0019	

### 点 A → 点 B

#### X 轴

功能	DR	设置
目标频率	D0020, D0021	5294.11764705882 Hz
扫零速度脉冲频率	D0022, D0023	882.352941176471 Hz
方向控制		双脉冲输出
绝对位置模式		启用
预置值	D0026, D0027	30000
当前值	D0028, D0029	

#### Y 轴

功能	DR	设置
目标频率	D0030, D0031	2823.52941176471 Hz
扫零速度脉冲频率	D0032, D0033	470.588235294118 Hz
方向控制		双脉冲输出
绝对位置模式		启用
预置值	D0036, D0037	16000
当前值	D0038, D0039	

### 点 B → 点 C

#### X 轴

功能	DR	设置
目标频率	D0060, D0061	5883.48405414552 Hz
扫零速度脉冲频率	D0062, D0063	980.58067569092 Hz
方向控制		双脉冲输出
绝对位置模式		启用
预置值	D0066, D0067	20000
当前值	D0068, D0069	

#### Y 轴

功能	DR	设置
目标频率	D0070, D0071	1176.6968108291 Hz
扫零速度脉冲频率	D0072, D0073	196.116135138184 Hz
方向控制		双脉冲输出
绝对位置模式		启用
预置值	D0076, D0077	4000
当前值	D0078, D0079	

### 点 C → 点 A

#### X 轴

功能	DR	设置
目标频率	D0110, D0111	0 Hz
扫零速度脉冲频率	D0112, D0113	0 Hz
方向控制		双脉冲输出
绝对位置模式		启用
预置值	D0116, D0117	0
当前值	D0118, D0119	

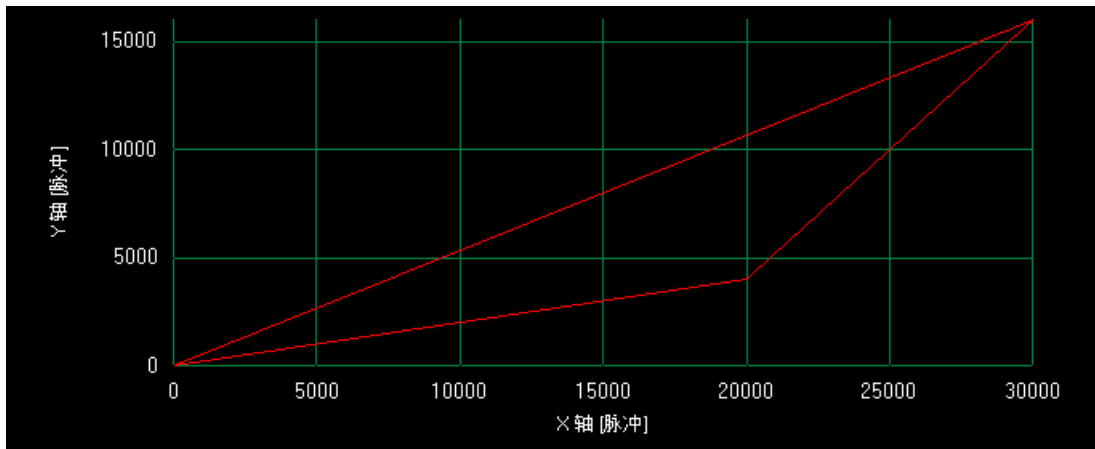
#### Y 轴

功能	DR	设置
目标频率	D0120, D0121	0 Hz
扫零速度脉冲频率	D0122, D0123	0 Hz
方向控制		双脉冲输出
绝对位置模式		启用
预置值	D0126, D0127	0
当前值	D0128, D0129	

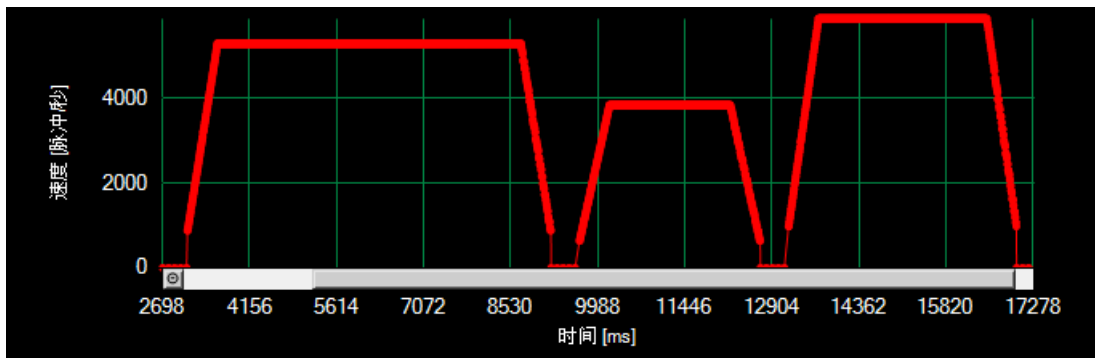
## 18: 脉冲输出指令

### ■ 脉冲监控画面

#### • 轨迹



#### • 波形



# 19: PID 控制指令

## 简介

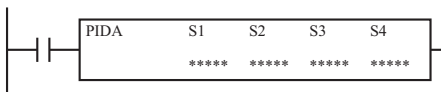
本章介绍使用温度控制等进行自动调整或 PID 控制的 PID 控制指令。

## PID (PID 控制)

执行 PID 控制，并输出其结果。

FT2J FC6A

进行自动调整时，会自动计算最佳的 PID 参数（比例增益、积分时间、微分时间）及动作方向。该指令与 FC5A 型的 PID 指令通用。使用 FC5A 型的 PID 指令进行温度控制等时，可在置换为 FC6A 型和 FC6A 型的 PID 指令后实现相同的温度控制。



### 注释:

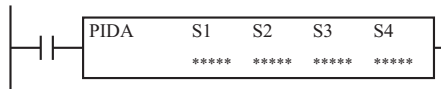
- 可在 1 个程序中混合使用 PID 指令和 PIDA 指令，总计最多可使用 32 个。
- 有关 PID 指令的详情，请参见《FC5A 系列 MICROSmart Pentra 用户手册 高级卷》第 14 章“PID 指令”。

## PIDA (PID 控制)

执行 PID 控制，并输出其结果。

FT2J FC6A

进行自动调整时，计算最佳的 PID 常量。切换自动 / 手动模式时，可自动运行无平衡无忧动功能，防止输出操作变量的剧烈变化。可从 9 种报警动作中设置最多 4 个，因此可针对多种条件检测测量值的异常。



### 注释:

- 梯形图程序中可使用的 PIDA 指令总数取决于控制器类型。  
FT2J 型：最多 6 个  
FC6A 型：可在 1 个梯形图程序中混合使用 PID 指令和 PIDA 指令，总计最多可使用 32 个。
- 有关 PIDA 指令的详情，请参见第 19-4 页上的“PIDA (PID 控制)”。

## PIDD (微分衰减的 PID)

执行 PID 控制，并输出其结果。

FT2J FC6A

存储在控制寄存器中的 PID 控制的参数以数据类型 F（浮点）进行处理，因此可对比例增益、积分增益及微分增益等控制参数进行详细调整。也可用于组合多个 PIDD 指令的串级控制。



### 注释:

- 梯形图程序中可使用的 PIDD 指令总数取决于控制器类型。  
FT2J 型：最多 6 个  
FC6A 型：最多 32 个
- 有关 PIDD 指令的详情，请参见第 19-28 页上的“PIDD (微分衰减的 PID)”。

## 19: PID控制指令

### 关于 PID 指令、PIDA 指令、PIDD 指令、PID 模块的差异

PID 控制指令包括 PID 指令、PIDA 指令和 PIDD 指令 3 种。此外，使用 PID 模块还可进行 PID 控制。各 PID 指令和 PID 模块的差异如下所示。

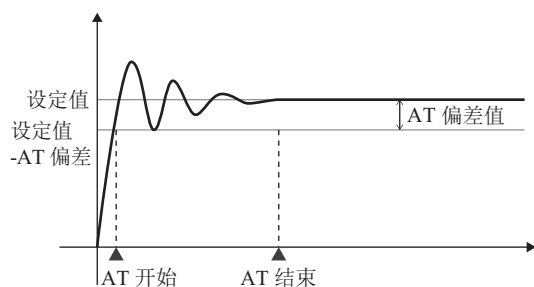
项目		PID 指令 <sup>*1</sup>	PIDA 指令	PIDD 指令	PID 模块 <sup>*2</sup>
指令数限制	FT2J 型	否	6	6	否
	FC6A 型	加上 PID 指令和 PIDA 指令后合计 32 个		32 个	—
参数数据类型		W (字型) / I (整型)		F (浮点型)	W (字型) / I (整型) (至小数第 1 位)
串级控制功能		否	否	是	是
加热 / 冷却控制		否	否	否	是
自动 / 手动切换		是	是	是	是
自动调整	方式	阶跃响应法	极限循环法	否	临界灵敏度法
	执行时间	在启动 PID 指令时执行	在任意时间执行		可在任意时间执行
自调谐		否	否	是	否
ARW (防自动复位)		是	是	是	是
无平衡无扰动功能		否	是	是	是
报警种类	上限报警动作 下限报警动作	是	是	否	是
	上 / 下限报警动作 上 / 下限范围报警动作 进程上限报警动作 进程下限报警动作 上限报警待命动作 下限报警待命动作 上 / 下限报警待命动作	否	是	否	是
参数设置		PIDST 指令、PID 指令	PIDA 指令	PIDD 指令	PID 模块设置
WindLDR 的监控	参数的确认和更改	使用设备监视器	使用专用的监控画面		
	趋势图显示	无	有		
具有兼容性的本公司指令		FC5A 型 MICROSmart 的 PID 指令	FT1A 型 Touch 的 PID 指令	无	无

\*1 仅 FC6A 型

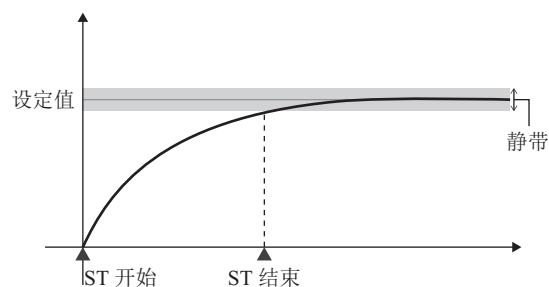
\*2 仅可连接 FC6A 型。

## 自动调整 and 自调谐的区别

自动调整



自调谐



自动调整 (AT) 和自调谐 (ST) 的区别如下所示。

项目	自动调整 (AT)	自调谐 (ST)
基本动作	根据打开 / 关闭控制的响应计算出 PID 常量并进行设置。	监控控制目标, 根据控制目标的特性, 按照一定周期计算出 PID 常量并进行设置。
计算结果	通过 1 次调整计算最佳 PID 值。	通过 1 次调整有时无法计算出最佳 PID 常量值。
执行调整	设定值到达 AT 开始点时执行 AT。	ST 启用时, 按照一定周期执行 ST。
调整时的响应	由于进行打开 / 关闭控制, 控制会混乱。	由于不进行打开 / 关闭控制, 控制不会混乱。
控制目标的特性变动时	需要再次执行自动调整。	自动计算出 PID 常量并进行设置。

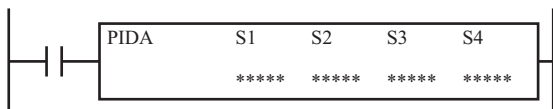
**注释:** 由于执行自调谐, PIDD 指令不会对控制目标带来较大变动, 始终执行最适当控制。如果重新考虑 PID 控制, 推荐使用 PIDD 指令。

## PIDA (PID 控制)

执行 PID 控制，并输出其结果。

FT2J FC6A

如果进行自动调整，将计算出最佳 PID 常量。  
 切换自动 / 手动模式时，可自动运行无平衡无扰动功能，防止输出操作变量的突变。  
 而且，可从 9 种报警动作中任意设置 4 个，并针对多种条件检测测量值的异常。



### 操作

输入打开时，将进行自动调整或 PID 控制。



#### 警告

- 要使用 PIDA 功能，则需要有关 PID 控制的专业技术知识。使用 PIDA 功能而不理解 PID 控制可能会使控制器执行误动作，导致控制系统发生异常、损坏或故障。
- 当使用 PIDA 指令进行反馈控制时，必须在控制器外部设置紧急停止和联锁电路。在内部电路中设置以上内容时，如果导致测量值无法正常输入，则可能导致无法进行正常反馈控制或连接设备损坏或故障。有关 PID 控制的详情，请参见第 19-25 页上的“PID 控制”。

### 注释:

- 梯形图程序中可使用的 PIDA 指令总数取决于控制器类型。详情请参见第 19-2 页上的“关于 PID 指令、PIDA 指令、PIDD 指令、PID 模块的差异”。
- PIDA 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
S3 (源 3)	控制继电器	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—
S4 (源 4)	设定值	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1 ~ 7。

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

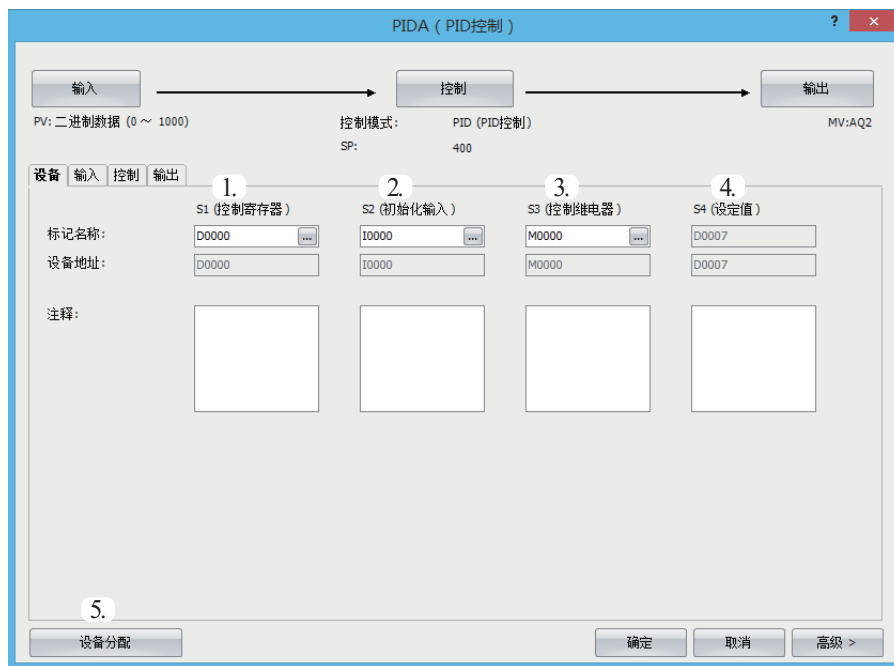


## 设置

“PIDA (PID 控制)”对话框中包括 4 个选项卡，分别为“设备”选项卡、“输入”选项卡、“控制”选项卡及“输出”选项卡。

在“设备”选项卡中，通过 PIDA 指令设置所使用的设备。在“输入”选项卡、“控制”选项卡以及“输出”选项卡中，设置 PIDA 指令的各参数初始值。

### ■ 设备选项卡



#### 1. S1 (源 1): 控制寄存器

指定存储 PIDA 指令各项目的起始设备。

可指定的设备为数据寄存器。以指定的数据寄存器为起始，占用 40 个字。

打开初始化输入后，可通过“输入”选项卡、“控制”选项卡及“输出”选项卡中所设置的值，对控制寄存器进行初始化。

有关初始化的详情，请参见第 19-5 页上的“2. S2 (源 2): 初始化输入”。

控制寄存器的说明，请参见第 19-19 页上的“S1: 控制寄存器”。

#### 2. S2 (源 2): 初始化输入

指定初始化控制寄存器的设备。

初始化输入打开时，会将“输入”选项卡、“控制”选项卡以及“输出”选项卡中所设置的值，存储到控制寄存器、控制继电器中<sup>\*1</sup>。可指定外部输入或内部继电器。

<sup>\*1</sup> 希望仅执行 1 次初始化时，请将 SOTU (上升沿微分) 指令或 SOTD (下降沿微分) 指令添加到输入条件中。

#### 3. S3 (源 3): 控制继电器

指定 PID 控制的控制和输出 PIDA 指令控制结果的设备。

以指定的设备为起始，占用 16 个字。

可指定的设备为内部继电器。无法指定特殊内部继电器。

每个位的作用均有不同，通过打开 / 关闭位，可切换 PIDA 指令的自动 / 手动模式或执行自动调整。而且，会将 PIDA 指令的控制结果或测量值的异常作为警报输出。

有关标记名称、设备地址、注释，请参见第 19-23 页上的“S3: 控制继电器”。

#### 4. S4 (源 4): 设定值

显示存储着 PID 控制目标值的设备 (S1+7)。

设定值 (S1+7) 是控制寄存器 (1) 所占用的 40 个字数据寄存器的其中 1 个。通过控制寄存器 (1) 指定起始设备时，将会自动显示。

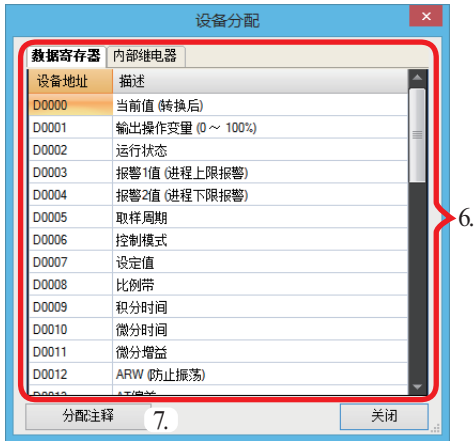
## 19: PID控制指令

### 5. 设备分配

单击此按钮，将显示“设备分配”对话框。对话框中将显示 PIDA 指令的各设置内容和数据寄存器、内部继电器的对应表 (6)。

单击“分配注释”按钮 (7) 后，对应表的内容可作为各设备的注释进行设置。

#### 设备分配对话框



## ■ “输入”选项卡

设置 PID 指令的输入参数。



### 1. 当前值 (S1+0)

设置 PID 控制中的输入。线性转换测量值后，将其用作输入值。可选择“模拟量”或“数据寄存器”。

#### “模拟量”时

指定拥有希望指定为测量值模拟量输入的模块和其模拟量输入编号，并设为 PID 控制的输入。

可以指定控制器中的内置模拟输入端、模拟输入盒、模拟量输入模块或模拟量混合 I/O 模块的模拟输入端。

项目	说明
节点编号*1	选择模拟输入设备的节点编号。选择节点 0 ~ 10。
设置*2	选择带模拟输入的设备。 CPU/HMI 模块：如果要指定控制器中的内置模拟输入或模拟输入盒的模拟输入，请选择 CPU/HMI 模块。 扩展模块 1 ~ 15：如果要指定模拟量输入模块或模拟量混合 I/O 模块的模拟输入，请选择扩展模块 1 ~ 15。
模拟量输入编号	在配置的设备上选择模拟输入编号*3。

\*1 仅 Plus CPU 模块。有关节点的详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“增设扩展模块组合型”。

\*2 仅 FC6A 型。

\*3 在 FT2J 型中，选择 FT2J 型内置模拟输入或模拟输入盒的模拟输入编号。

模拟量输入值也作为只读存储于控制寄存器的 (S1+0) 中。

有关控制寄存器的详情，请参见第 19-5 页上的“1. S1 (源 1): 控制寄存器”。

#### 注释:

- 使用模拟量输入时，需要事先设置模拟量输入。有关模拟量输入的设置详情，请参见以下手册。  
FT2J 型：《WindO/I-NV4 用户手册》第 28 章“2.10 模拟量 / 数字量输入”  
《SmartAXIS 硬件手册》第 2 章“2 模拟 I/O 盒”  
FC6A 型：《FC6A 型 MICROSmart 用户手册》第 9 章“模拟 I/O 模块”或第 10 章“I/O 盒”
- 指定了“模拟量”时，模拟量输入设置的最小值、最大值的设置将变为输入的最小值、输入的最大值。
- 为 FC6A 型指定模拟输入时，模拟量值 (AI0) 将无法作为测量值使用。

## 19: PID 控制指令

### “数据寄存器”时

将存储在控制寄存器的 S1+0 中的值设为 PID 控制输入。

存储值的范围为最小值 (5)  $\leq$  测量值  $\leq$  最大值 (4)。有关控制寄存器的详情，请参见第 19-5 页上的“1. S1 (源 1): 控制寄存器”。

在测量值中选择“数据寄存器”时设置项目 (2) 设备地址 ~ (5) 最小值。

### 2. 设备地址

显示“设备”选项卡中所设置的控制寄存器的第一个数据寄存器 (S1+0)。

### 3. 数据类型

从 W (字型) 或 I (整型) 中选择测量值 (1) 的数据类型。

### 4. 最大值、5. 最小值

以最大值、最小值设置当前值 (1) 的范围。在最大值、最小值的范围内设置 PID 控制中的各输入设置 (当前值、报警 1 值 (进程上限报警)、报警 2 值 (进程下限报警)) 及设定值。可在数据类型 (3) 的范围内设置最大值 / 最小值。

- 数据类型为 I (整型) 时: -32,768 ~ +32,767
- 数据类型为 W (字型) 时: 0 ~ 65,535

**注释:** 当测量值 (1) 比最大值 (4) 大或比最小值 (5) 小时，将状态代码 109 存储到运行状态 (S1+2) 中，停止 PID 控制。有关状态代码的详情，请参见第 19-21 页上的“运行状态 (S1+2)”。

### 6. 取样周期 (S1+5)

设置执行 PID 控制的周期。

仅在完成扫描时执行 PID 控制。

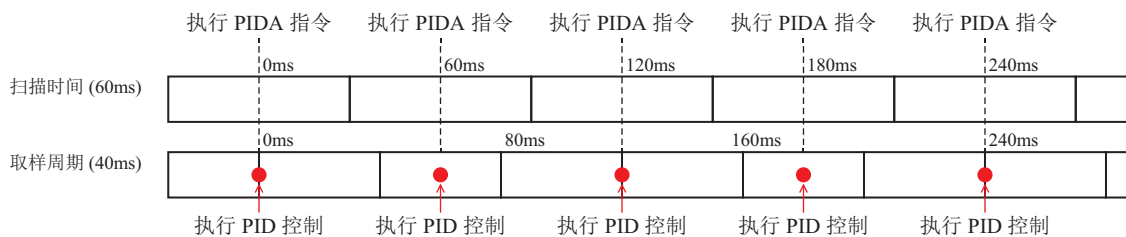
将取样周期设置的比扫描时间短时，无法在设置的取样周期内执行 PID 控制，而是与扫描时间在相同周期内执行。

将取样周期设置的比扫描时间长时，如果经过取样周期时仍未完成扫描，则将等待执行，直到 PID 控制完成扫描时为止。

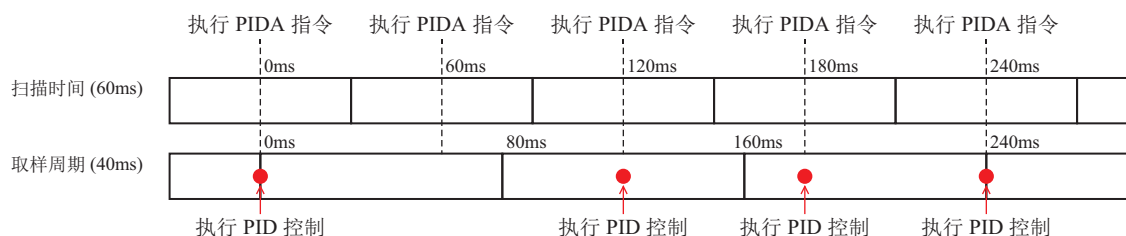
通过执行以上操作，会在取样周期与实际的 PID 控制执行时发生误差。

取样时间可在 0.01 ~ 100.00 秒的范围内以 0.01 秒为单位进行设置。

#### 取样周期 $\leq$ 扫描时间



#### 取样周期 $>$ 扫描时间



### 7. 高级 / 基本

单击此按钮，可切换详细设置的显示 / 隐藏。

可在“输入”选项卡中，将取样周期 (6) 作为详细设置进行设置。

### 8. 警报 1 预置值 (进程上限警报) (S1+3)

设置测量值的上限值 (1)。

可在警告 2 预置值 (进程下限警报) (9) ~ 最大值 (4) 的范围内设置警报 1 预置值。

如果测量值超过警报 1 预置值，将打开警报 1 输出 (S3+3)。

如果测量值小于警报 1 预置值，将关闭警报 1 输出 (S3+3)。

9. 警报 2 设定值 (进程下限警报) (S1+4)

设置 PID 控制中测量值的下限值 (1)。

可在最小值 (5) ~ 警报 1 预置值 (进程上限警报) (8) 的范围内设置警报 2 预置值。

如果测量值小于警报 2 预置值, 将打开警报 2 输出 (S3+4)。

如果测量值大于警报 2 预置值, 将关闭警报 2 输出 (S3+4)。

可在警报 3 ~ 6 的最大 4 个范围内设置 PID 控制中的测量值警报。

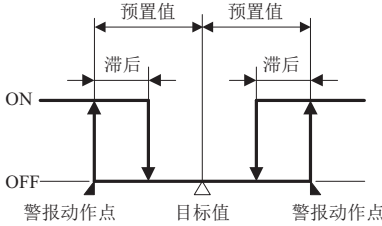
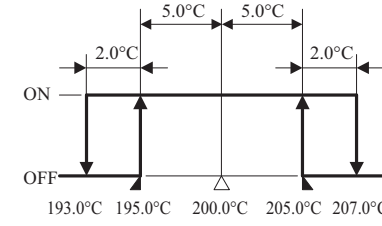
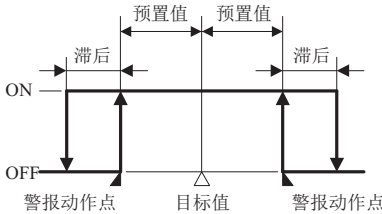
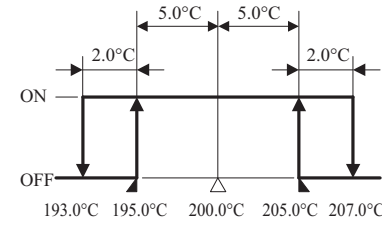
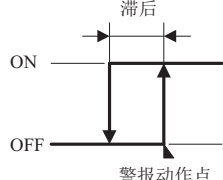
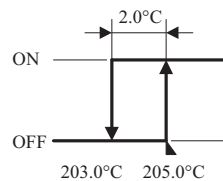
对警报动作 (10)、预置值 (11)、滞后 (12)、延时定时器 (13) 进行个别设置。

10. 警报类型

从以下选项中选择警报的动作种类。

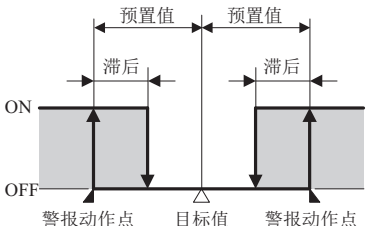
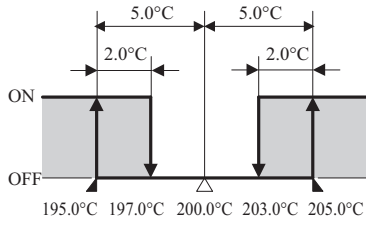
种类	动作	例
无报警动作	不输出警报。	—
上限报警	<p>测量值 <math>\geq</math> (目标值 + 预置值) 时, 将打开警报输出。                      测量值 <math>\leq</math> (目标值 + 预置值 - 滞后) 时, 将关闭警报输出。                      (目标值 + 预置值 - 滞后) &lt; 测量值 &lt; (目标值 + 预置值) 时, 警报输出将维持 1 次扫描前的状态。</p>	<p>目标值: 200.0°C                      预置值: 5.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\geq</math> 205.0°C 时, 将打开警报输出。                      测量值 <math>\leq</math> 203.0°C 时, 将关闭警报输出。</p>
	<p>测量值 <math>\leq</math> (目标值 + 预置值) 时, 将打开警报输出。                      测量值 <math>\geq</math> (目标值 + 预置值 + 滞后) 时, 将关闭警报输出。                      (目标值 + 预置值) &lt; 测量值 &lt; (目标值 + 预置值 + 滞后) 时, 警报输出将维持 1 次扫描前的状态。</p>	<p>目标值: 200.0°C                      预置值: -5.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\geq</math> 195.0°C 时, 将打开警报输出。                      测量值 <math>\leq</math> 193.0°C 时, 将关闭警报输出。</p>
下限报警	<p>测量值 <math>\leq</math> (目标值 + 预置值) 时, 将打开警报输出。                      测量值 <math>\geq</math> (目标值 + 预置值 + 滞后) 时, 将关闭警报输出。                      (目标值 + 预置值) &lt; 测量值 &lt; (目标值 + 预置值 + 滞后) 时, 警报输出将维持 1 次扫描前的状态。</p>	<p>目标值: 200.0°C                      预置值: 5.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\leq</math> 205.0°C 时, 将打开警报输出。                      测量值 <math>\geq</math> 207.0°C 时, 将关闭警报输出。</p>
	<p>测量值 <math>\geq</math> (目标值 + 预置值) 时, 将打开警报输出。                      测量值 <math>\leq</math> (目标值 + 预置值 - 滞后) 时, 将关闭警报输出。                      (目标值 + 预置值 - 滞后) &lt; 测量值 &lt; (目标值 + 预置值) 时, 警报输出将维持 1 次扫描前的状态。</p>	<p>目标值: 200.0°C                      预置值: -5.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\leq</math> 195.0°C 时, 将打开警报输出。                      测量值 <math>\geq</math> 197.0°C 时, 将关闭警报输出。</p>

# 19: PID控制指令

种类	动作	例
上/下限报警	<p>测量值 <math>\geq</math> (目标值 + 预置值) 时, 将打开报警输出。                      测量值 <math>\leq</math> (目标值 - 预置值) 时, 将打开报警输出。                      (目标值 - 预置值 + 滞后) <math>\leq</math> 测量值 <math>\leq</math> (目标值 + 预置值 - 滞后) 时, 将关闭报警输出。                      (目标值 + 预置值 - 滞后) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 + 预置值) 时, 报警输出将维持 1 次扫描前的状态。                      (目标值 - 预置值) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 - 预置值 - 滞后) 时, 报警输出将维持 1 次扫描前的状态。</p>  <p>报警动作点      目标值      报警动作点</p>	<p>目标值: 200.0°C                      预置值: 5.0°C                      滞后: 2.0°C</p>  <p>193.0°C 195.0°C 200.0°C 205.0°C 207.0°C</p> <p>测量值 <math>\geq</math> 205.0°C 时, 将打开报警输出。                      测量值 <math>\leq</math> 195.0°C 时, 将打开报警输出。                      197.0°C <math>\leq</math> 测量值 <math>\leq</math> 203.0°C 时, 将关闭报警输出。</p>
上/下限范围报警	<p>(目标值 - 预置值) <math>\leq</math> 测量值 <math>\leq</math> (目标值 + 预置值) 时, 将打开报警输出。                      测量值 <math>\geq</math> (目标值 + 预置值 + 滞后) 时, 将关闭报警输出。                      测量值 <math>\leq</math> (目标值 - 预置值 - 滞后) 时, 将关闭报警输出。                      (目标值 + 预置值) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 + 预置值 + 滞后) 时, 报警输出将维持 1 次扫描前的状态。                      (目标值 - 预置值 - 滞后) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 - 预置值) 时, 报警输出将维持 1 次扫描前的状态。</p>  <p>报警动作点      目标值      报警动作点</p>	<p>目标值: 200.0°C                      预置值: 5.0°C                      滞后: 2.0°C</p>  <p>193.0°C 195.0°C 200.0°C 205.0°C 207.0°C</p> <p>195.0°C <math>\leq</math> 测量值 <math>\leq</math> 205.0°C 时, 将打开报警输出。                      测量值 <math>\geq</math> 207.0°C 时, 将关闭报警输出。                      测量值 <math>\leq</math> 193.0°C 时, 将关闭报警输出。</p>
进程上限报警	<p>测量值 <math>\geq</math> 预置值时, 将打开报警输出。                      测量值 <math>\leq</math> (目标值 - 滞后) 时, 将关闭报警输出。                      (预置值 - 滞后) <math>&lt;</math> 测量值 <math>&lt;</math> 预置值时, 报警输出将维持 1 次扫描前的状态。</p>  <p>报警动作点</p>	<p>预置值: 205.0°C                      滞后: 2.0°C</p>  <p>203.0°C 205.0°C</p> <p>测量值 <math>\geq</math> 205.0°C 时, 将打开报警输出。                      测量值 <math>\leq</math> 203.0°C 时, 将关闭报警输出。</p>

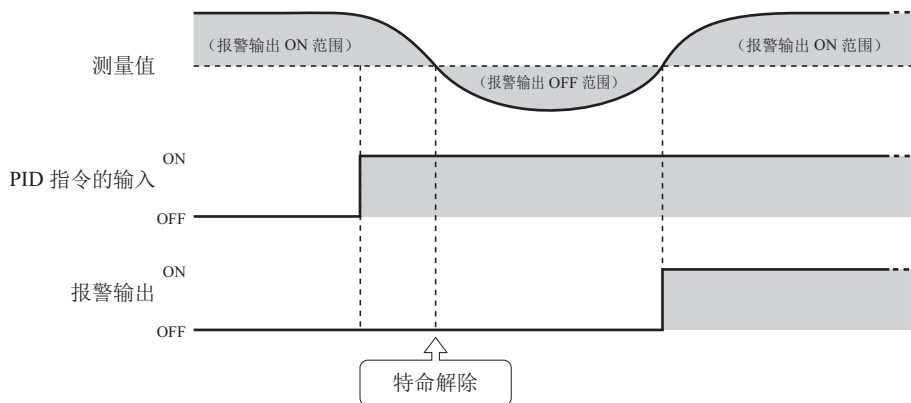
种类	动作	例
进程下限报警	<p>测量值 <math>\leq</math> 预置值时，将打开警报输出。                      测量值 <math>\geq</math> (目标值 + 滞后) 时，将关闭警报输出。                      预置值 <math>&lt;</math> 测量值 <math>&lt;</math> (预置值 + 滞后) 时，警报输出将维持 1 次扫描前的状态。</p> <p>报警动作点</p>	<p>预置值: 195.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\leq</math> 195.0°C 时，将打开警报输出。                      测量值 <math>\geq</math> 197.0°C 时，将关闭警报输出。</p>
上限报警待命	<p>测量值 <math>\geq</math> (目标值 + 预置值) 时，将打开警报输出。                      测量值 <math>\leq</math> (目标值 + 预置值 - 滞后) 时，将关闭警报输出。                      (目标值 + 预置值 - 滞后) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 + 预置值) 时，警报输出将维持 1 次扫描前的状态。                      部分待命功能有效。</p> <p>- 报警动作点      目标值      + 报警动作点</p>	<p>目标值: 200.0°C                      预置值: 5.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\geq</math> 205.0°C 时，将打开警报输出。                      测量值 <math>\leq</math> 203.0°C 时，将关闭警报输出。</p>
下限报警待命	<p>测量值 <math>\leq</math> (目标值 + 预置值) 时，将打开警报输出。                      测量值 <math>\leq</math> (目标值 + 预置值 + 滞后) 时，将关闭警报输出。                      (目标值 + 预置值) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 + 预置值 + 滞后) 时，警报输出将维持 1 次扫描前的状态。                      部分待命功能有效。</p> <p>- 报警动作点      目标值      + 报警动作点</p>	<p>目标值: 200.0°C                      预置值: -5.0°C                      滞后: 2.0°C</p> <p>测量值 <math>\leq</math> 195.0°C 时，将打开警报输出。                      测量值 <math>\geq</math> 197.0°C 时，将关闭警报输出。</p>

## 19: PID控制指令

种类	动作	例
上/下限报警待命	<p>测量值 <math>\geq</math> (目标值 + 预置值) 时, 将打开报警输出。                      测量值 <math>\leq</math> (目标值 - 预置值) 时, 将打开报警输出。                      (目标值 - 预置值 + 滞后) <math>\leq</math> 测量值 <math>\leq</math> (目标值 + 预置值 - 滞后) 时, 将关闭报警输出。                      (目标值 + 预置值 - 滞后) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 + 预置值) 时, 报警输出将维持 1 次扫描前的状态。                      (目标值 - 预置值) <math>&lt;</math> 测量值 <math>&lt;</math> (目标值 - 预置值 + 滞后) 时, 报警输出将维持 1 次扫描前的状态。                      部分待命功能有效。</p> 	<p>目标值: 200.0°C                      预置值: 5.0°C                      滞后: 2.0°C</p>  <p>测量值 <math>\geq</math> 205.0°C 时, 将打开报警输出。                      测量值 <math>\leq</math> 195.0°C 时, 将打开报警输出。                      197.0°C <math>\leq</math> 测量值 <math>\leq</math> 203.0°C 时, 将关闭报警输出。</p>

### 注释:

- 警报动作点  
表示报警输出从 OFF 切换为 ON 的地点。
- 待命功能  
该功能在开始执行 PIDA 指令时, 即使测量值处于报警输出 ON 的范围, 仍无法立即打开报警输出。  
如果测量值暂时处于报警输出 OFF 的范围, 将会解除待命功能, 当报警输出再次处于 ON 的范围时, 将打开报警输出。  
变更目标值后, 待命功能再次有效。





### 11. 预置值

设置警报动作（10）的动作条件值。要设置的值因警报动作而异。  
预置值的内容如下所示。

警报动作	预置值	设置范围
上限报警	设置与目标值的偏差。	测量值为电流、电压或数据寄存器时： -32,768 ~ 32,767* <sup>1</sup>
下限报警		测量值通过热电偶或电阻温度计，将摄氏或华氏设为数据类型时： -3,276.8 ~ 3,276.7°C/°F* <sup>1</sup>
上/下限报警		测量值为电流、电压或数据寄存器时： 0 ~ 65,535* <sup>1</sup>
上/下限范围报警		测量值通过热电偶或电阻温度计，将摄氏或华氏设为数据类型时： 0.0 ~ 6,553.5°C/°F* <sup>1</sup>
进程上限报警	通过进程设置警报所执行的值。	最小值~最大值* <sup>1</sup>
进程下限报警		
上限报警待命	设置与目标值的偏差。	测量值为电流、电压或数据寄存器时： -3,276.8 ~ 3,276.7* <sup>1</sup>
下限报警待命		测量值通过热电偶或电阻温度计，将摄氏或华氏设为数据类型时： -3,276.8 ~ 3,276.7°C/°F* <sup>1</sup>
上/下限报警待命	设置与目标值的偏差。	测量值为电流、电压或数据寄存器时： 0 ~ 65,535* <sup>1</sup> 测量值通过热电偶或电阻温度计，将摄氏或华氏设为数据类型时： 0.0 ~ 6,553.5°C/°F* <sup>1</sup>

\*<sup>1</sup> 预置值为 0 或 0.0 时，变为无警报动作。

### 12. 滞后

从警报动作点开始，设置警报输出从 ON 切换为 OFF 的宽度。

通过增加滞后，警报输出因细微变化而导致无法切换。缩小滞后时，即使警报动作点附近发生细微变化，警报输出仍可切换，但有时会对所连接的装置产生不良影响。

设置范围为 0.1 ~ 6,553.5（°C/°F）或 1 ~ 65,535（电压、电流输入）。

但是，警报动作为“上/下限报警”以及“上/下限报警待命”时，需要设置满足  $1 \leq \text{滞后} < \text{预置值（11）}$  的滞后。

### 13. 延时定时器

在满足打开警报输出的条件后，设置警报输出打开之前的时间。

可防止因噪音等影响导致的测量值变动，及警报输出在非预期状态下打开的情况。

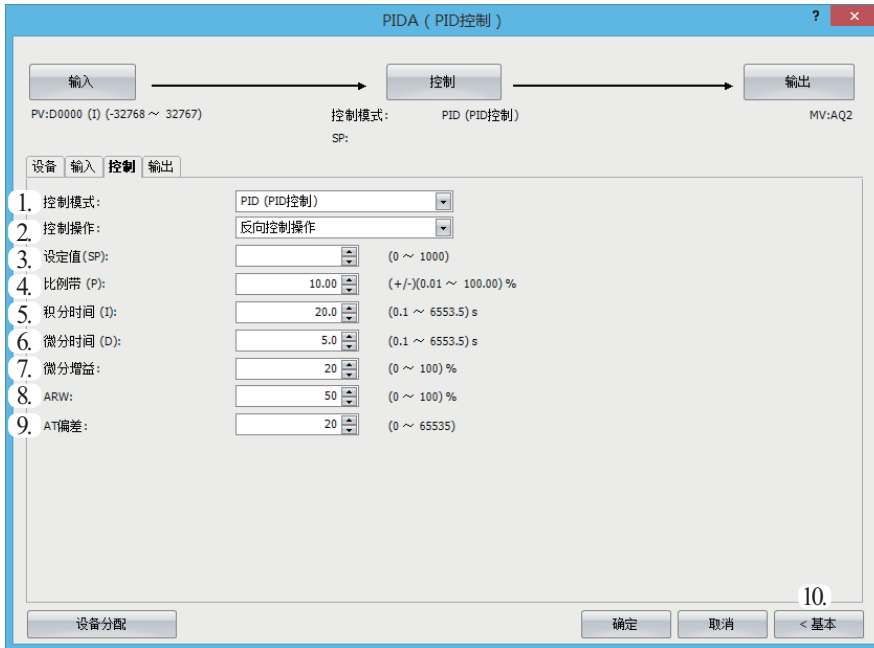
在满足打开警报输出的条件后，如果在超过延时时间之前仍无法满足条件，将无法打开警报输出。

设置范围为 0 ~ 10,000 秒。0 时，禁用延时时间。

## 19: PID控制指令

### ■ “控制”选项卡

设置 PIDA 指令的控制参数。



#### 1. 控制模式 (S1+6)

根据控制对象的特性，在“PID（PID 控制）”、“P（比例控制）”、“PI（PI 控制）”、“PD（PD 控制）”中进行选择设置。

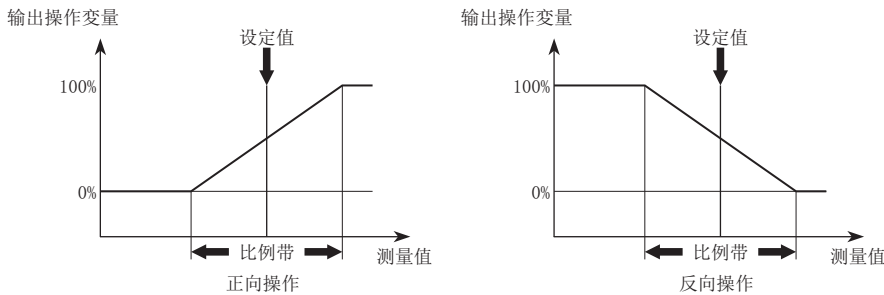
有关各控制，请参见第 19-25 页上的“PID 控制”。

#### 2. 控制操作 (S3+0)

设置控制操作。可选择“正向控制操作”或“反向控制操作”。

正操作是指测量值高于目标值时增加输出操作变量的操作。冷却装置等适用于该操作。

逆操作是指测量值低于目标值时增加输出操作变量的操作。加热炉等适用于该操作。(S3+0)为只读。



#### 3. 设定值 (S1+7)

可在测量值的最小值~最大值的范围内设置目标值。目标值的设置单位所下所示。

在测量值中设置了“模拟量输入”的热电偶输入或电阻温度计输入时，可以  $0.1^{\circ}\text{C} / ^{\circ}\text{F}$  为单位进行设置。

在测量值中设置了“数据寄存器”或“模拟量输入”的电压输入、电流输入、热敏电阻或电阻测量时，可以 1 为单位进行设置。

#### 4. 比例带 (S1+8)

比例操作是指这样一类操作，其输出变化的幅度与目标值、测量值间的偏差成正比。

测量值在比例范围内时，与偏差成正比后，打开 / 关闭控制输出。测量值在比例范围外时，控制输出 (S3+6) 将始终为打开或关闭状态。

比例范围扩大时，由于控制输出 (S3+6) 将从偏差较大的状态进行打开 / 关闭，因此过冲（测量值高于目标值）或下冲（测量值低于目标值）、速度偏差（测量值不稳定，起伏状态）将变少，但测量值达到目标值所花费的时间将延长，目标值与测量值的偏移量也将增大。

比例范围缩小时，由于控制输出 (S3+6) 将由目标值附近进行打开 / 关闭，因此尽管测量值达到目标值所花费的时间变短，且偏移量也将变小，但速度偏差将变大。如果将比例范围缩小到极限，则变为与打开 / 关闭操作相同的控制。

如果使用自动调整功能，则可针对控制对象自动设置合适的比例范围。有关详情，请参见第 19-26 页上的“自动调整 (AT)”。

比例范围可在  $\pm 0.01 \sim \pm 100.00\%$  的范围内以  $0.01\%$  为单位进行设置。

### 5. 积分时间 (S1+9)

如果仅限于比例操作，即使控制对象达到稳定状态，目标值与测量值间也将产生一定的偏差（偏移量）。由于该差接近 0，因此需要积分动作。积分时间是由积分操作而决定操作量的系数，如果积分时间过短，则积分操作变强，将引起长周期速度偏差。相反，如果积分时间过长，则达到目标值期间的处理将较为费时。积分执行范围为 - 比例范围 ~ + 比例范围，因变更目标值或出现干扰而导致测量值偏离比例范围时，将停止积分运算。该结果将提升目标值操作量的跟踪性，过冲、下冲都可进行较少的控制。

如果使用自动调整功能，则可针对控制对象自动设置合适的积分时间。有关详情，请参见第 19-26 页上的“自动调整 (AT)”。

积分时间可在 0.1 ~ 6,553.5 秒的范围内以 0.1 秒为单位进行设置。

### 6. 微分时间 (S1+10)

因变更目标值或出现干扰而导致目标值和测量值之差变大时，加大操作量，快速将测量值接近目标值的操作称为微分操作。如果缩小微分时间，则微分操作变弱，针对温度急剧变化的响应变慢。此外，由于控制温度急剧上升的功能变弱，虽然目标值的升温时间变快，但在此期间容易引起过冲。如果增加微分时间，则微分操作变强，针对温度急剧变化的响应变快。此外，由于控制温度急剧上升的功能变强，虽然目标值的升温时间变慢，但在此期间难以引起过冲。

如果使用自动调整功能，则可针对控制对象自动设置合适的微分时间。有关详情，请参见第 19-26 页上的“自动调整 (AT)”。

微分时间可在 0.1 ~ 6,553.5 秒的范围内以 0.1 秒为单位进行设置。

### 7. 微分增益 (S1+11)

将微分增益设小时，输出操作变量将易于受到干扰或目标值变化的影响。将微分增益设大时，虽然输出操作变量将不易受到干扰或目标值变化的影响，但会降低正常时的稳定性。引起干扰或测量值的变化时，正常设为 20 ~ 30%。

可在 0 ~ 100% 的范围内以 1% 为单位设置微分增益。

### 8. ARW (Anti Reset Windup) (S1+12)

设置开始积分操作的点。如果从开始执行 PIDA 指令时启用积分项，将引起过冲。可通过使 ARW 随比例范围联动进行延迟，从而抑制过冲。一般情况下，可实现 ARW 为 100% 且过冲较少的合理控制。如果过于缩小 ARW，虽然不存在过冲，但会产生偏移量。

可在 0 ~ 100% 的范围内设置 ARW。

例如，ARW=50%、比例范围=20.00% 时，从偏差低于 10% 时开始进行积分操作。

### 9. AT (Auto Tuning) 偏差 (S1+13)

设置自动调整时的偏压值。

可在 0 ~ 100% 的范围内以 1% 为单位设置 ARW。

测量值的值	AT 的开始点
测量值 $\leq$ (目标值 - AT 偏压值) 时	目标值 - AT 偏压值
测量值 $\geq$ (目标值 + AT 偏压值) 时	目标值 + AT 偏压值
(目标值 - AT 偏压值) < 测量值 < (目标值 + AT 偏压值) 时	目标值

有关详情，请参见第 19-26 页上的“自动调整 (AT)”。

可在测量值的最小值 < (目标值 - AT 偏压预置值)、或测量值的最大值 > (目标值 + AT 偏压预置值) 的范围内，以 1 为单位设置 AT 偏压。

### 10. 高级 / 基本

单击此按钮，可切换详细设置的显示 / 隐藏。

可在“控制”选项卡中，将微分增益 (7) 作为详细设置进行设置。

## ■ “输出”选项卡

设置 PIDA 指令的输出参数。



### 1. 输出操作变量（数字值）（S1+19）

设置 PID 控制中的输出。可选择“模拟输出”或“数据寄存器”。是根据输出操作变量限制（6）的设置，计算通过 PIDA 指令计算出的操作量的值。

#### “模拟输出”时

设置模块及其节点和其模拟编号，该模块拥有希望设置为输出操作变量（模拟值）输出端口的模拟量输出。可以指定控制器中的内置模拟输出端、模拟输出盒、模拟输出模块或模拟混合 I/O 模块的模拟输出端。

项目	说明
节点编号*1	选择模拟输出设备的节点编号。选择节点 0 ~ 10。
设置*2	选择带模拟输出的设备。 CPU/HMI 模块：如果要指定控制器中的内置模拟输出或模拟输出盒的模拟输出，请选择 CPU/HMI 模块。 扩展模块 1 ~ 15：如果要指定模拟量输出模块或模拟量混合 I/O 模块的模拟输出，请选择扩展模块 1 ~ 15。
模拟量输出编号	在配置的设备上选择模拟输出编号*3。

\*1 仅 Plus CPU 模块。有关节点的详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“增设扩展模块组合型”。

\*2 仅 FC6A 型。

\*3 在 FT2J 型中，选择 FT2J 型内置模拟输出或模拟输出盒的模拟输出编号。

- 自动模式时，是将输出操作变量（S1+19）中存储的模拟值，线性转换为模拟量输出的最大值和最小值范围后进行输出。
- 手动模式时，是将手动模式输出操作变量（S1+17）中存储的值，线性转换为模拟量输出的最大值和最小值范围后进行输出。

有关自动 / 手动模式的切换的详情，请参见第 19-23 页上的“自动 / 手动模式（S3+1）”。

而且，输出操作变量的值会自动存储到模拟量输出所对应的特殊数据寄存器中。

有关特殊数据寄存器的分配的详情，请参见第 2-14 页上的“”。

**注释:**

- 使用模拟量输出时，需要事先设置模拟量输出。有关模拟量输出的设置的详情，请参见以下手册。  
 FT2J 型：《WindO/I-NV4 用户手册》第 28 章 “2.11 模拟量输出”  
           《SmartAXIS 硬件手册》第 2 章 “2 模拟 I/O 盒”  
 FC6A 型：《FC6A 型 MICROSmart 用户手册》第 9 章 “模拟 I/O 模块”  
           《FC6A 型 MICROSmart 用户手册》第 10 章 “I/O 盒”

**“数据寄存器”时**

指定存储输出操作变量的设备。

- 自动模式时，PID 控制的输出操作变量会存储到控制寄存器的输出操作变量（模拟值）（S1+19）中。  
 在最小值 (5)  $\leq$  输出操作变量  $\leq$  最大值 (4) 的范围内进行存储。

• 手动模式时，是将手动模式输出操作变量（S1+17）中存储的值，线性转换为最小值 (5)、最大值 (4) 范围后的值。  
 在测量值中选择数据寄存器时设置项目 (2) 设备地址 ~ (5) 最小值。

**2. 设备地址**

在输出操作变量 (1) 中设置了“数据寄存器”时，将显示设备选项卡中所设置的控制寄存器的 S1+19。

**3. 数据类型**

从 W（字型）或 I（整型）中选择输出操作变量 (1) 的数据类型。

**4. 最大值、5. 最小值**

已在输出操作变量中设置了“模拟量输出”时  
 本设置无效。

已在输出操作变量中设置了“数据寄存器”时

通过线性转换为最大值 (4) 和最小值 (5) 范围后的值，将 PIDA 指令的输出操作变量存储到所设置的数据寄存器中。

**6. 输出操作变量限制**

限制 PIDA 指令计算出的操作量，并设为输出操作变量。

**“禁用”时**

将通过 PIDA 指令计算出的操作量设为输出操作变量。

**“启用（上限和下限）”时：**

通过 PIDA 指令计算出的操作量超过输出操作变量的上限或下限时，会将上限值或下限值设为输出操作变量。

**“启用（百分比）”时**

在通过 PIDA 指令计算出的操作量中，将乘以 1 ~ 99% 的结果设为输出操作变量。此时，输出操作变量下限值 (8) 无效。

**7. 输出操作变量上限 / 百分比 (S1+14)****“启用 (上限和下限)”时**

设置控制输出的输出操作变量的上限值。通过 PIDA 指令计算出的操作量超过输出操作变量的上限值时，会将已设置的上限值设为控制输出的输出操作变量。设置输出操作变量上限值时，需要设置满足输出操作变量下限值 < 输出操作变量上限值的输出操作变量上限值。

可在 0 ~ 100% 的范围内以 1% 为单位设置输出操作变量限制下限值。

**“启用（百分比）”时**

设置输出操作变量的百分比。在通过 PIDA 指令计算出的操作量中，将乘以 1 ~ 99% 的结果设为输出操作变量。可在 1 ~ 99% 的范围内以 1% 为单位设置输出操作变量（百分比）。

输出操作变量 = 通过 PIDA 指令计算出的操作量  $\times$  输出操作变量（倍率）

**8. 输出操作变量下限 (S1+15)****“启用（上限和下限）”时**

设置控制输出的输出操作变量的下限值。通过 PIDA 指令计算出的操作量低于输出操作变量的下限值时，会将已设置的下限值设为控制输出的输出操作变量。设置输出操作变量下限值时，需要设置满足输出操作变量下限值 < 输出操作变量上限值的输出操作变量下限值。可在 0 ~ 100% 的范围内以 1% 为单位设置输出操作变量限制下限值。

**“启用（百分比）”时：**

输出操作变量下限值无效。

### 9. 手动模式输出 MV (S1+17)

设置手动模式时的输出操作变量。启用该功能时，需要打开自动 / 手动模式 (S3+1)。手动模式输出操作变量针对输出操作变量 (数字值)、控制输出有效。

有关自动 / 手动模式的切换，请参见第 19-23 页上的“自动 / 手动模式 (S3+1)”。

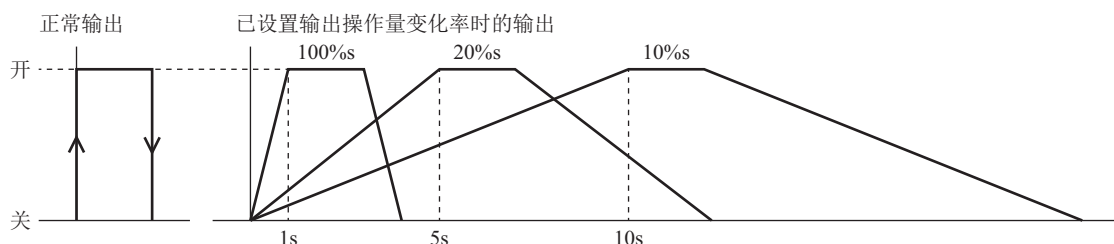
输出操作变量限制为“启用 (上限和下限)”时，需要在输出操作变量下限值~输出操作变量上限值的范围内设置手动模式输出操作变量。可在 0.00 ~ 100.00% 的范围内以 0.01% 为单位设置手动模式输出操作变量。

### 10. 输出 MV 变化率 (S1+16)

设置 1 秒间变化的输出操作变量。输出操作变量变化率为 0%/ 秒时，本功能无效。

目标值与测量值之差较大时，如下图所示，虽然通常的输出由关闭变为打开，但如果设置输出操作变量变化率，则如下图所示，可改变输出操作变量的变化率。

可在 0 ~ 100% / 秒的范围内以 1% 为单位设置输出操作变量变化率。



适用于高速通电后突然断电的情况，如控制高温用加热器 (该装置由钼、钨、铂等材料构成，可在约 1,500 ~ 1,800°C 环境下使用)。

### 11. 比例带偏差值 (S1+20)

设置比例带的偏差值。以比例带偏差值设置的值可增减输出操作变量 (S1+1)。

例如，将比例带偏差值设为 20% 时，输出操作变量 (S1+1) 会增加 20%。

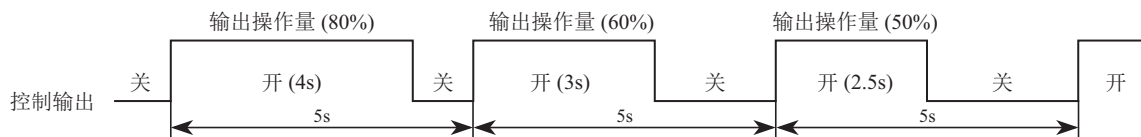
可在 -100 ~ 100% 的范围内以 1% 为单位设置比率范围偏移值。

### 12. 控制周期 (S1+21)

根据输出操作变量，设置对控制输出 (S3+5) 进行打开 / 关闭控制的周期。控制周期的打开脉宽随输出操作变量而发生变化。

控制周期可在 0.1 ~ 50.0 秒的范围内以 0.1 秒为单位进行设置。

**控制周期: 5 秒 (预置值 50) 时**



### 13. 高级 / 基本

单击此按钮，可切换详细设置的显示 / 隐藏。可在“输出”选项卡中，将输出操作变量限制 (6)、输出操作变量限制上限值 / 百分比 (7)、输出操作变量限制下限值 (8)、手动模式输出 MV (9)、输出 MV 变化率 (10)、比例带偏差值 (11)、控制周期 (12) 作为详细设置进行设置。

## S1: 控制寄存器

存储目的地	功能	设置内容	读/写
S1+0	当前值	<ul style="list-style-type: none"> <li>“模拟量”时可设置模拟输入编号。此时，S1+0 为只读。</li> <li>“数据寄存器”时请存储满足输入最小值<math>\leq</math>测量值<math>\leq</math>输入最大值的值。</li> </ul>	读/写
S1+1	输出操作变量	在 0 ~ 100 (0% ~ 100%) 的范围内存储。	读
S1+2	运行状态	存储 PIDA 指令的执行状态或错误状态。 有关详情，请参见第 19-21 页上的“运行状态 (S1+2)”。	读
S1+3	警报 1 预置值 (进程上限警报)	目标值 $<$ 测量值的最小值、测量值的最大值 $<$ 目标值时，将变为目标值设置错误，并保持之前的目标值。 警报 1 预置值 $\leq$ 警报 2 预置值 (进程下限警报)、警报 1 预置值 $>$ 测量值的最大值时，将测量值的最大值作为警报 1 预置值执行。	读/写
S1+4	警报 2 设定值 (进程下限警报)	设置测量值的下限值。请设置满足测量值的最小值 $\leq$ 警报 2 预置值 $<$ 警报 1 预置值 (进程上限警报) 的值。 警报 2 预置值 $<$ 测量值的最小值、警报 1 预置值 (进程上限警报) $\leq$ 警报 2 预置值时，将测量值的最小值作为警报 2 预置值执行。	读/写
S1+5	取样周期	请在 1 ~ 10,000 (0.01 ~ 100.00 秒) 的范围内设置。 0 作为 0.01 秒执行、10,001 以上则作为 100.00 秒执行。	读/写
S1+6	控制模式	设置常值控制操作。 0: PID 控制 1: P 控制 2: PI 控制 3: PD 控制 如果为上述以外的值，将作为 0 (PID 控制) 执行。	读/写
S1+7	设定值	请设置满足测量值最小值 $\leq$ 目标值 $\leq$ 测量值最大值的值。 目标值 $<$ 测量值的最小值、测量值的最大值 $<$ 目标值时，将变为目标值设置错误，并保持之前的目标值。	读/写
S1+8	比例带	请在 1 ~ 10,000 ( $\pm 0.01$ ~ $\pm 100.00\%$ ) 的范围内设置。 0 作为 $\pm 0.01\%$ 执行、10,001 以上则作为 $\pm 100.00\%$ 执行。	读/写
S1+9	积分时间	请在 1 ~ 65,535 (0.1 ~ 6,553.5 秒) 的范围内设置。 0 时的积分操作无效。	读/写
S1+10	微分时间	请在 1 ~ 65,535 (0.1 ~ 6,553.5 秒) 的范围内设置。 0 时的微分操作无效。	读/写
S1+11	微分增益	请在 0 ~ 100 (0 ~ 100%) 的范围内设置。 0 作为 0% 执行、101 以上则作为 100% 执行。	读/写
S1+12	ARW	请在 0 ~ 100 (0 ~ 100%) 的范围内设置。 101 以上则作为 100% 执行。	读/写
S1+13	AT 偏差	请设置满足测量值的最小值 $<$ (目标值 - AT 偏压预置值)、或测量值的最大值 $>$ (目标值 + AT 偏压预置值) 的值。 已设置未满足上述两个条件的值时，将 133 存储到操作状态 (S1+2) 中。	读/写
S1+14	输出操作变量上限 / 百分比	<ul style="list-style-type: none"> <li>“启用 (上限和下限)”时 请在 0 ~ 100 (0 ~ 100%) 的范围内设置。 0 ~ 100 以外时，作为 100 执行。</li> <li>“启用 (百分比)”时 请在 10001 ~ 10099 (1 ~ 99%) 的范围内设置。 10001 ~ 10099 以外将作为输出操作变量限制上限值执行。 输出操作变量 = 通过 PIDA 指令计算出的操作量 <math>\times</math> 倍率</li> </ul>	读/写
S1+15	输出操作变量下限	请在 0 ~ 100 (0 ~ 100%) 的范围内，设置满足输出操作变量限制下限值 $<$ 输出操作变量限制上限值的值。 0 ~ 100 以外时，作为 0 执行。	读/写
S1+16	输出操作变量变化率	请在 0 ~ 100 (0 ~ 100%/秒) 的范围内设置。 0 ~ 100 以外时，禁用输出操作变量变化率。	读/写

## 19: PID控制指令

存储目的地	功能	设置内容	读/写
S1+17	手动模式输出操作变量	请设置输出操作变量下限值 $\leq$ 手动模式输出操作变量 $\leq$ 输出操作变量上限值。 在启用输出操作变量上/下限设置 (S3+2) 的状态下, 未满足输出操作变量限制下限值 $\sim$ 输出操作变量限制上限值时, 输出操作变量限制下限值以下将作为输出操作变量限制下限值执行, 输出操作变量限制上限值以上将作为输出操作变量限制上限值执行。	读/写
S1+18	输出操作变量 (% 值)	在 -32768 $\sim$ +32767 (-327.68 $\sim$ +327.67%) 范围内保存。 正在执行自动调整的输出操作变量 (% 值) 不固定。 手动模式时, 不会反映手动模式输出操作变量的值。	读
S1+19	输出操作变量 (模拟量值)	<ul style="list-style-type: none"> <li>在输出操作变量中指定了“模拟输出”时 在模拟量输出的最小值<math>\sim</math>最大值范围内, 存储 S1+1 输出操作变量 (%) 线性转换后的值。</li> <li>在输出操作变量中指定了“数据寄存器”时 在输出最小值<math>\sim</math>输出最大值范围内, 保存 S1+1 输出操作变量 (%) 线性转换后的值。</li> </ul>	读
S1+20	比例带偏差值	请在 -100 $\sim$ 100 (-100 $\sim$ 100%) 的范围内设置。 -101 以下时作为 -100% 执行、101 以上时作为 100% 执行。	读/写
S1+21	控制周期	请在 1 $\sim$ 500 (0.1 $\sim$ 50.0 秒) 的范围内设置。 0 作为 0.1 秒进行操作、501 以上则作为 50.0 秒进行操作。	读/写
S1+22	警报 3 动作	设置警报 3 动作的种类。 0: 不动作 1: 上限警报 2: 下限警报 3: 上/下限警报 4: 上/下限范围警报 5: 进程上限警报 6: 进程下限警报 7: 上限警报待命 8: 下限警报待命 9: 上/下限警报待命 如果为上述以外的值, 将作为 0 (不动作) 执行。	读/写
S1+23	警报 3 预置值	设置警报 3 动作的动作条件值。 有关详情, 请参见第 19-22 页上的“警报 3 预置值 (S1+23)”。	读/写
S1+24	警报 3 滞后	从警报动作点开始, 设置警报输出从 ON 切换为 OFF 的宽度。 请在 1 $\sim$ 65,535 (0.1 $\sim$ 6,553.5 ( $^{\circ}$ C/ $^{\circ}$ F)、1 $\sim$ 65,535 (电压、电流输入)) 的范围内, 设置满足以下条件的值。 <ul style="list-style-type: none"> <li>已在警报 3 动作中设置了“上/下限警报”或“上/下限警报待命”时 1 <math>\leq</math> 警报 3 滞后 <math>&lt;</math> 警报 3 预置值</li> <li>已在警报 3 动作中设置了“上/下限警报”或“上/下限警报待命”以外时 1 <math>\leq</math> 警报 3 滞后 <math>\leq</math> 65,535</li> </ul> 0 时, 将滞后作为 1 (0.1 ( $^{\circ}$ C/ $^{\circ}$ F) 或 1 (电压、电流输入)) 执行。	读/写
S1+25	警报 3 延时时间	在满足打开警报输出的条件后, 设置警报输出打开之前的时间。 请在 0 (禁用延时时间) 或 1 $\sim$ 10,000 (1 $\sim$ 10,000 秒) 的范围内进行设置。 10,001 以上时, 将作为 10,000 秒执行。	读/写
S1+26	警报 4 动作	设置警报 4 动作的种类。 设置内容与警报 3 动作 (S1+22) 相同。	读/写
S1+27	警报 4 预置值	设置警报 4 动作的动作条件值。 设置内容与警报 3 预置值 (S1+23) 相同。	读/写
S1+28	警报 4 滞后	从警报动作点开始, 设置警报输出从 ON 切换为 OFF 的宽度。 设置内容与警报 3 滞后 (S1+24) 相同。	读/写
S1+29	警报 4 延时时间	在满足打开警报输出的条件后, 设置警报输出打开之前的时间。 设置内容与警报 3 延时时间 (S1+25) 相同。	读/写
S1+30	警报 5 动作	设置警报 5 动作的种类。 设置内容与警报 3 动作 (S1+22) 相同。	读/写
S1+31	警报 5 预置值	设置警报 5 动作的动作条件值。 设置内容与警报 3 预置值 (S1+23) 相同。	读/写



存储目的地	功能	设置内容	读/写
S1+32	警报 5 滞后	从警报动作点开始, 设置警报输出从 ON 切换为 OFF 的宽度。 设置内容与警报 3 滞后 (S1+24) 相同。	读/写
S1+33	警报 5 延时时间	在满足打开警报输出的条件后, 设置警报输出打开之前的时间。 设置内容与警报 3 延时时间 (S1+25) 相同。	读/写
S1+34	警报 6 动作	设置警报 6 动作的种类。 设置内容与警报 3 动作 (S1+22) 相同。	读/写
S1+35	警报 6 预置值	设置警报 6 动作的动作条件值。 设置内容与警报 3 预置值 (S1+23) 相同。	读/写
S1+36	警报 6 滞后	从警报动作点开始, 设置警报输出从 ON 切换为 OFF 的宽度。 设置内容与警报 3 滞后 (S1+24) 相同。	读/写
S1+37	警报 6 延时时间	在满足打开警报输出的条件后, 设置警报输出打开之前的时间。 设置内容与警报 3 延时时间 (S1+25) 相同。	读/写
S1+38 S1+39	— 保留 —	—	—

### 运行状态 (S1+2)

表示 PIDA 指令的执行状态或错误状态。存储状态代码。

- 表中的“X”表示从开始自动调整时到结束自动调整的经过时间。“X”以 10 分钟为单位逐一变化。经过时间在 90 分钟以上时为 9。
- 如果正在执行 PID, 表中的“x”表示从 PID 开始时至达到目标值为止的的经过时间。“x”以 10 分钟为单位逐一变化。经过时间 90 分钟以上时则为 9。
- 操作状态为错误代码 (状态代码为 100 以上且未满 120 的值) 时, 将停止执行 PID 控制。请在设置常规参数后, 重新打开 PIDA 指令的输入。

状态代码	状态说明	状态分类
1x	正在执行自动调整	正在正常执行 AT
2x	自动调整结束	
5x	正在执行 P/PI/PID 控制	正在正常执行 PID 控制
6x	达到目标值 (即使只有一次达到目标值, 也会由 5x 变为 6x。)	
103	已设置输出操作变量限制上限值 < 输出操作变量限制下限值。	
104	— 保留 —	—
106	已在目标值中设置了未满足测量值最小值 $\leq$ 目标值 $\leq$ 测量值最大值的值。	PID 控制执行停止
109	已设置测量值 > 测量值最大值、或测量值 < 测量值最小值。	PID 控制执行停止
120	已在控制模式中设置了 0 ~ 3 以外的值。	PID 控制执行继续
121	已在取样时间中设置了 0 或 10,001 以上的值。	PID 控制执行继续
122	已在比例范围中设置了 0 或 10,001 以上的值。	PID 控制执行继续
123	已在微分增益中设置了 0 或 101 以上的值。	PID 控制执行继续
124	已在积分开始系数 (ARW) 中设置了 101 以上的值。	PID 控制执行继续
125	在警报 1 预置值 (进程上限警报) 中设置了未满足警报 2 预置值 (进程下限警报) < 警报 1 预置值 (进程上限警报) $\leq$ 测量值的最大值的值。	PID 控制执行继续
126	在警报 2 预置值 (进程下限警报) 中设置了未满足测量值的最小值 $\leq$ 警报 2 预置值 (进程下限警报) < 警报 1 预置值 (进程上限警报) 的值。	PID 控制执行继续
127	已在输出操作变量限制上限值中设置了 101 ~ 10,000 或 10,100 ~ 65,535 的值。	PID 控制执行继续
128	已在输出操作变量限制下限值中设置了 101 以上的值。	PID 控制执行继续
129	已在输出操作变量变化率中设置了 101 以上的值。	PID 控制执行继续
130	已在手动模式输出操作变量中设置了未满足输出操作变量限制下限值 $\leq$ 手动模式输出操作变量 $\leq$ 输出操作变量限制上限值的值。	PID 控制执行继续
131	已在比率范围偏移值中设置了 -100 ~ 100 以外的值。	PID 控制执行继续
132	已在控制周期中设置了 0 或 501 以上的值。	PID 控制执行继续

## 19: PID 控制指令

状态代码	状态说明	状态分类
133	请在 AT 偏压中设置未满足测量值的最小值 < (目标值 - AT 偏压预置值)、且测量值的最大值 > (目标值 + AT 偏压预置值) 的值。	PID 控制执行继续
134	在执行 PIDA 指令第 2 次扫描后, 在目标值中设置了不满足最小值 $\leq$ 目标值 $\leq$ 最大值的值。	PID 控制执行继续 (保持到上次扫描为止的正常范围内的目标值。)
135 ~ 199	— 保留 —	—
200	在警报 3 的警报动作中设置了 10 以上的值。	PID 控制执行继续
201	在警报 4 的警报动作中设置了 10 以上的值。	PID 控制执行继续
202	在警报 5 的警报动作中设置了 10 以上的值。	PID 控制执行继续
203	在警报 6 的警报动作中设置了 10 以上的值。	PID 控制执行继续
204 ~ 209	— 保留 —	—
210	设置预置值的警报 3 警报动作点已超过了测量值的最小值或最大值。	PID 控制执行继续
211	设置预置值的警报 4 警报动作点已超过了测量值的最小值或最大值。	PID 控制执行继续
212	设置预置值的警报 5 警报动作点已超过了测量值的最小值或最大值。	PID 控制执行继续
213	设置预置值的警报 6 警报动作点已超过了测量值的最小值或最大值。	PID 控制执行继续
214 ~ 219	— 保留 —	—
220	设置了滞后的警报 3 警报动作点已超过测量值的最小值或最大值。	PID 控制执行继续
221	设置了滞后的警报 4 警报动作点已超过测量值的最小值或最大值。	PID 控制执行继续
222	设置了滞后的警报 5 警报动作点已超过测量值的最小值或最大值。	PID 控制执行继续
223	设置了滞后的警报 6 警报动作点已超过测量值的最小值或最大值。	PID 控制执行继续
224 ~ 229	— 保留 —	—
230	已在警报 3 的警报延时时间中设置了 10,001 以上的值。	PID 控制执行继续
231	已在警报 4 的警报延时时间中设置了 10,001 以上的值。	PID 控制执行继续
232	已在警报 5 的警报延时时间中设置了 10,001 以上的值。	PID 控制执行继续
233	已在警报 6 的警报延时时间中设置了 10,001 以上的值。	PID 控制执行继续
234 ~ 239	— 保留 —	—
240	已在警报 3 的警报动作中设置了上 / 下限警报或上 / 下限警报待命时, 在警报 3 的滞后中设置了滞后 < 未满足预置值的值。	PID 控制执行继续
241	已在警报 4 的警报动作中设置了上 / 下限警报或上 / 下限警报待命时, 在警报 4 的滞后中设置了滞后 < 未满足预置值的值。	PID 控制执行继续
242	已在警报 5 的警报动作中设置了上 / 下限警报或上 / 下限警报待命时, 在警报 5 的滞后中设置了滞后 < 未满足预置值的值。	PID 控制执行继续
243	已在警报 6 的警报动作中设置了上 / 下限警报或上 / 下限警报待命时, 在警报 6 的滞后中设置了滞后 < 未满足预置值的值。	PID 控制执行继续
244 ~ 65535	— 保留 —	—

### 警报 3 预置值 (S1+23)

设置警报动作的动作条件值。要设置的值因警报动作而异。  
预置值的内容如下所示。

警报动作	预置值	设置范围
上限报警	设置与目标值偏差的值。	-32,768 ~ 32,767* <sup>1</sup>
下限报警		
上 / 下限报警		0 ~ 65,535* <sup>1</sup>
上 / 下限范围报警		
进程上限报警	通过进程设置警报所执行的值。	最小值 ~ 最大值
进程下限报警		

警报动作	预置值	设置范围
上限报警待命	设置与目标值偏差的值。	-32,768 ~ 32,767*1
下限报警待命		
上 / 下限报警待命		0 ~ 65,535*1

\*1 预置值为 0 时，变为无警报动作。

## S2: 初始化输入

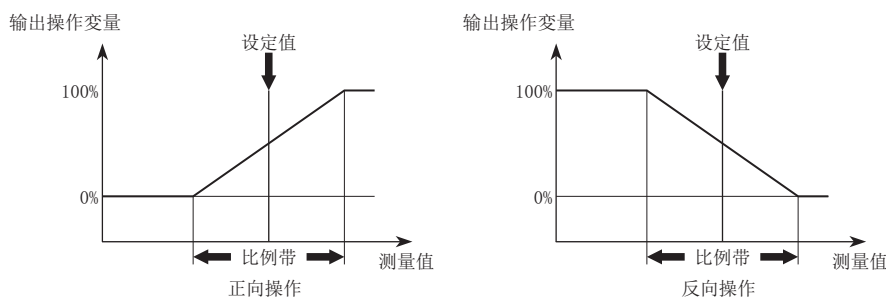
初始化输入为 ON 时，将以 WindLDR 的输入选项卡、控制选项卡以及输出选项卡中所设置的值覆盖控制寄存器（数据寄存器）。希望仅执行 1 次初始化时，请将 SOTU（上升）指令或 SOTD（下降）指令添加到输入条件中。

## S3: 控制继电器

存储目的地	功能	设置内容	读 / 写
S3+0	控制操作	0 (关): 反向操作 1 (开): 正向操作	读
S3+1	自动 / 手动模式	0 (关): 自动 1 (开): 手动	读 / 写
S3+2	输出操作变量限制启用	0 (关): 禁用 1 (开): 启用 (以 S1+14、15 的设置进行操作)	读 / 写
S3+3	警报 1 输出	测量值 (S1+0) $\geq$ 警报 1 预置值 (进程上限警报) (S1+3) 时, 处于打开状态。	读
S3+4	警报 2 输出	测量值 (S1+0) $\leq$ 警报 2 预置值 (进程下限警报) (S1+4) 时, 处于打开状态。	读
S3+5	控制输出	按照控制周期和输出操作变量进行打开 / 关闭。	读
S3+6	AT 执行	打开后开始自动调整。 如果正在执行自动调整时关闭, 将停止自动调整。	读 / 写
S3+7	AT 完成输出	自动调整结束后打开	读
S3+8	警报 3 输出	测量值 (S1+0) 超出警报 3 动作 (S1+22) 的范围时, 处于打开状态。 测量值 (S1+0) 在警报 3 动作 (S1+22) 的范围内时, 处于关闭状态。	读
S3+9	警报 4 输出	测量值 (S1+0) 超出警报 4 动作 (S1+26) 的范围时, 处于打开状态。 测量值 (S1+0) 在警报 4 动作 (S1+26) 的范围内时, 处于关闭状态。	读
S3+10	警报 5 输出	测量值 (S1+0) 超出警报 5 动作 (S1+30) 的范围时, 处于打开状态。 测量值 (S1+0) 在警报 5 动作 (S1+30) 的范围内时, 处于关闭状态。	读
S3+11	警报 6 输出	测量值 (S1+0) 超出警报 6 动作 (S1+34) 的范围时, 处于打开状态。 测量值 (S1+0) 在警报 6 动作 (S1+34) 的范围内时, 处于关闭状态。	读
S3+12 ~ S3+15	— 保留 —	—	—

### 控制操作 (S3+0)

正操作是指测量值高于目标值时增加输出操作变量的操作。冷却装置等适用于该操作。  
逆操作是指测量值低于目标值时增加输出操作变量的操作。加热炉等适用于该操作。



### 自动 / 手动模式 (S3+1)

自动模式是根据 PIDA 指令计算出的操作量进行控制的模式。手动模式是将手动模式输出操作变量 (S1+17) 输出为操作量的模式。使用手动模式时, 需要实现设置手动模式输出操作变量。

### 输出操作变量限制启用 (S3+2)

## 19: PID控制指令

---

设置启用 / 禁用输出操作变量上限值 (S1+14)、输出操作变量下限值 (S1+15) 的输出操作变量限制。

### 警报 1 输出 (S3+3)

如果测量值 (S1+0) 超过警报 1 预置值 (进程上限警报) (S1+3)，将处于打开状态。只读。

### 警报 2 输出 (S3+4)

如果测量值 (S1+0) 小于警报 2 预置值 (进程下限警报) (S1+4)，将处于打开状态。只读。

### 控制输出 (S3+5)

在自动模式下，根据 PIDA 指令计算出的操作量以及已设置的控制周期 (S1+21) 进行打开 / 关闭。在手动模式下，根据手动模式输出操作变量 (S1+17)、以及已设置的控制周期进行打开 / 关闭。

### AT 执行 (S3+6)

打开时将执行自动调整，结束后将自动关闭。如果正在执行自动调整时关闭，将停止自动调整。

### AT 完成输出 (S3+7)

自动调整结束后打开。此外，自动调整异常结束时也将打开。

### 警报 3 输出 (S3+8)

测量值 (S1+0) 超出警报 3 动作 (S1+22) 的范围时，处于打开状态。

测量值 (S1+0) 在警报 3 动作 (S1+22) 的范围内时，处于关闭状态。只读。

### 警报 4 输出 (S3+9)

测量值 (S1+0) 超出警报 4 动作 (S1+26) 的范围时，处于打开状态。

测量值 (S1+0) 在警报 4 动作 (S1+26) 的范围内时，处于关闭状态。只读。

### 警报 5 输出 (S3+10)

测量值 (S1+0) 超出警报 5 动作 (S1+30) 的范围时，处于打开状态。

测量值 (S1+0) 在警报 5 动作 (S1+30) 的范围内时，处于关闭状态。只读。

### 警报 6 输出 (S3+11)

测量值 (S1+0) 超出警报 6 动作 (S1+34) 的范围时，处于打开状态。

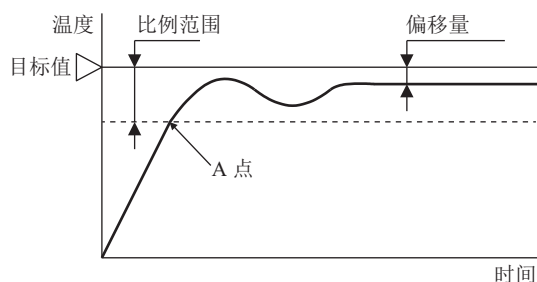
测量值 (S1+0) 在警报 6 动作 (S1+34) 的范围内时，处于关闭状态。只读。

## PID 控制

PID 控制为普通的温控操作，它主要负责实施调节操作，以消除单一的目标值与测量值间的偏差。可在控制器中使用的 PID 控制，如下所示。

### ■ P 控制

P 控制是指这一类控制，在比例范围内其输出操作变量的程度与目标值和测量值的偏差成正比。测量值达到 A 点（比例范围）时将打开输出，如果超过 A 点（进入比例范围），则开始以控制周期打开 / 关闭控制输出，如果超过目标值，则控制输出将完全变为关闭状态。随着从 A 点升温到目标值，控制输出的打开时间将缩短，而关闭时间将延长。P 控制必然会产生偏移量。P 操作适用于气体压力控制或级别控制等高效型程序。如果设置 P 控制，将自动忽略积分时间和微分时间。

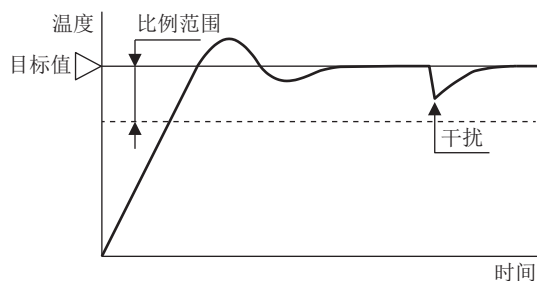


- 比例范围缩小时，由于控制输出将由目标值附近进行打开 / 关闭，因此尽管测量值升温至目标值所花费的时间缩短，且偏移量也将变小，但速度偏差将变大。
- 比例范围扩大时，由于控制输出将由大大低于目标值的温度进行打开 / 关闭，因此尽管过冲或速度偏差消失，测量值升温至目标值期间也将花费大量时间，且目标值和测量值的偏移量也将增大。

### ■ PI 控制

I 操作将自动修正由 P 控制引起的偏移量。但是，针对因干扰导致的温度急剧变化，稳定温度需要花费时间。PI 操作适用于变化速度较慢的温度控制。

如果设置 PI 控制，则微分时间自动变为 0。

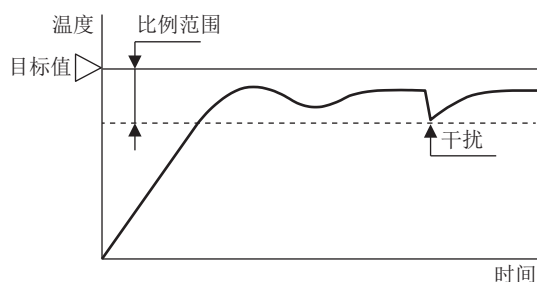


- 如果积分时间变短，则 I 控制增强，虽然可在短时间内修正偏移量，但将引起长周期速度偏差。
- 如果积分时间变长，则 I 控制减弱，修正偏移量需要花费时间。

### ■ PD 控制

与 P 控制相比，PD 控制针对因干扰导致的温度急剧变化，也力求快速响应，并在短时间内稳定控制，提高瞬态响应特性。PD 控制适用于变化速度较快的温度控制。

如果设置 PD 控制，则积分时间自动变为 0。

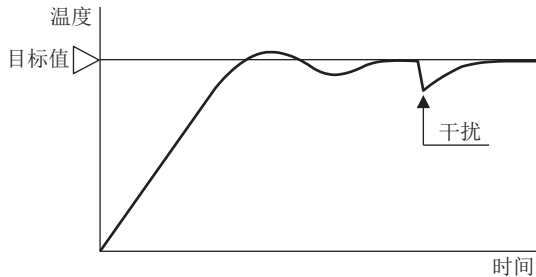


- 如果缩短微分时间，则 D 控制变弱，针对温度急剧变化的响应变慢。此外，由于控制温度急剧上升的功能变弱，虽然目标值的升温时间变快，但在此期间容易引起过冲。
- 如果延长微分时间，则 D 控制增强，针对温度急剧变化的响应变快。此外，由于控制温度急剧上升的功能变强，虽然目标值的升温时间变慢，但在此期间难以引起过冲。

## 19: PID控制指令

### ■ PID 控制

P 控制可抑制过冲或速度偏差，I 控制可修正偏移量，D 控制可在短时间内抑制因干扰导致的温度急剧变化。通过使用 PID 控制，可实现理想的温度控制。可通过自动调整，自动设置 PID 控制的比例带、积分时间、微分时间的各参数。



### 自动调整 (AT)

最佳温度控制的参数因控制对象的特性而异。进行 PID 控制时，通过执行自动调整，可自动设置比例范围 (P)、积分时间 (I)、微分时间 (D)。

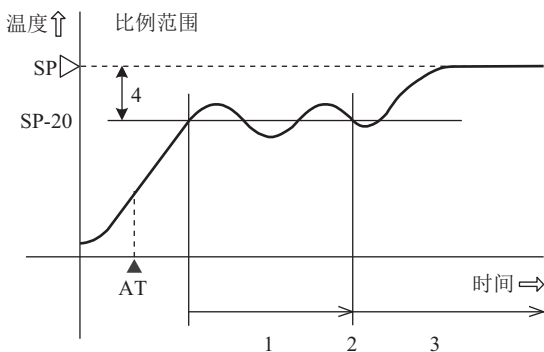
- 请在试运行中进行自动调整。
- 在常温附近执行自动调整时，无法进行相应温度变动，因此自动调整有时不会正常结束。此时，请手动设置 P、I、D 的各值。
- 执行一次自动调整后，只要程序不变，则无需重新执行自动调整。

### ■ 自动调整 (AT)

将自动设置比例范围 (P)、积分时间 (I)、微分时间 (D) 各值，因此需强制改变控制对象，将各值设置为最佳值。为进行最佳自动调整，需要在测量值达到目标值附近时，进行相应变动。通过设置 AT 偏压，可在测量值接近目标值时，进行相应变动。目标值、AT 偏压、自动调整开始点以及变动开始点的关系，如下所示。

#### 测量值 (PV) ≤ 目标值 (SP) - AT 偏压设置值

将 AT 偏压设置设为 20°C 时，如果测量值 (PV) 达到低于目标值 (SP) 20°C 的温度后，则开始变动。

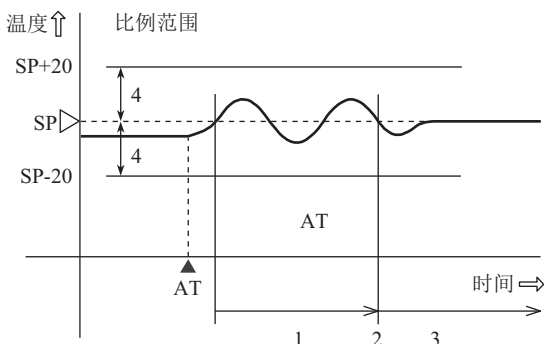


1. 正在测量 PID 常量
2. 计算 PID 常量
3. 以 AT 中设置的 PID 常量进行控制
4. AT 偏压设置值 (20)

▲ AT: AT 执行位打开地点

#### 目标值 (SP) - AT 偏压设置值 < 测量值 (PV) < 目标值 (SP) + AT 偏压设置值

测量值 (PV) 达到目标值 (SP) 时开始变动。

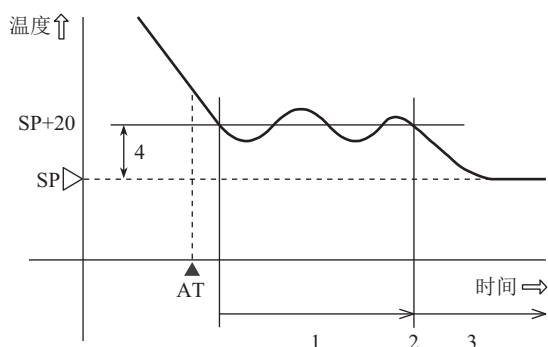


1. 正在测量 PID 常量
2. 计算 PID 常量
3. 以 AT 中设置的 PID 常量进行控制
4. AT 偏压设置值 (20)

▲ AT: AT 执行位打开地点

**测量值 (PV)  $\cong$  目标值 (SP) + AT 偏压设置值**

将 AT 偏压设置设为 20°C 时，如果测量值 (PV) 达到高于目标值 (SP) 20°C 的温度后，则开始变动。



1. 正在测量 PID 常量
2. 计算 PID 常量
3. 以 AT 中设置的 PID 常量进行控制
4. AT 偏压设置值 (20)

▲ AT: AT 执行位打开地点

**■ 执行自动调整 (AT)**

要执行自动调整时，需在 PIDA 指令输入为打开的状态下，打开自动调整执行标记 (S3+6)。P、I、D 的各值将自动进行设置。正在执行自动调整时，操作状态 (S1+2) 中将存入 1。

自动调整结束后，自动调整执行标记 (S3+6) 将自动关闭，并在操作状态 (S1+2) 中存入 2，打开自动调整完成输出 (S3+7)。

**■ 解除执行自动调整 (AT)**

如果要在执行自动调整的过程中解除自动调整，则需关闭自动调整执行标记 (S3+6)。关闭自动调整执行标记 (S3+6) 后，将停止执行自动调整，并在操作状态 (S1+2) 中存入 2，打开自动调整完成输出 (S3+7)。

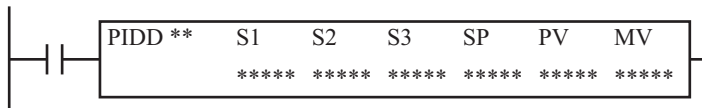
如果在中途解除自动调整，则 P、I、D 的各值将恢复至执行自动调整前的值。

## PIDD（微分衰减的 PID）

执行 PID 控制，并输出其结果。

FT2J FC6A

存储在控制寄存器中的 PID 控制的参数以数据类型 F（浮点）进行处理，因此可对比例增益、积分增益及微分增益等控制参数进行详细调整。也可用于组合多个 PIDD 指令的串级控制。



### 操作

输入为打开时进行 PID 控制。



#### 警告

要使用 PIDD 指令时，需要掌握 PID 控制的相关知识。如果在不理解 PID 控制的情况下使用，可能会变为用户非预期的控制，因此请在充分理解 PID 控制及 PIDD 指令后予以使用。  
要进行使用了 PIDD 指令的反馈控制时，请根据用户应用程序在控制器的外部电路中设置紧急停止电路或连锁电路等。在内部电路中设置以上内容时，如果导致测量值无法正常输入，则可能导致无法进行正常反馈控制或连接设备损坏或故障。

### 注释：

- 梯形图程序中可使用的 PIDD 指令总数取决于控制器类型。详情请参见第 19-2 页上的“关于 PID 指令、PIDA 指令、PIDD 指令、PID 模块的差异”。
- PIDD 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

控制寄存器为 100 个字，控制继电器占用 32 个内部继电器。

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	控制寄存器	—	—	—	—	—	—	X <sup>*1</sup>	—	—	—
S2 (源 2)	初始化输入	X	—	X	—	—	—	—	—	—	—
S3 (源 3)	控制继电器	—	—	X <sup>*2</sup>	—	—	—	—	—	—	—

\*1 无法使用特殊数据寄存器。

\*2 无法使用特殊内部继电器。此外，内部继电器编号的第一位只能指定为 0。不能指定为 1~7。  
关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。



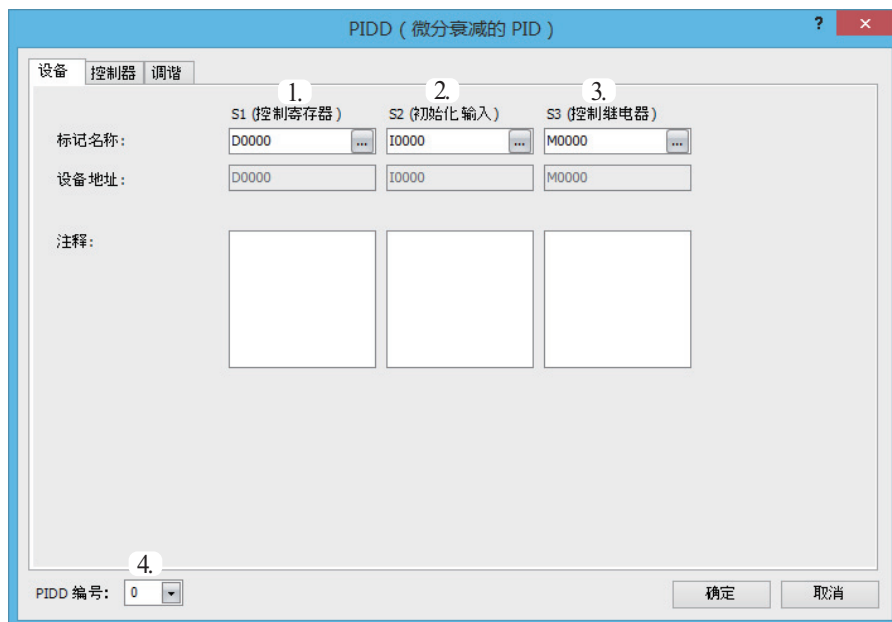
## 设置

“PIDD（微分衰减的PID）”对话框中包括3个选项卡，分别为“设备”选项卡、“控制器”选项卡及“调谐”选项卡。

在“设备”选项卡中设置PIDD指令占用的设备。

在“控制器”选项卡及“调谐”选项卡中设置PIDD指令的各参数初始值。

### ■ “设备”选项卡



#### 1. S1（源1）：控制寄存器

指定存储PIDD指令的控制参数的数据寄存器区域的起始设备。

以指定的数据寄存器为起始，占用100个字的数据寄存器。可在控制寄存器中指定的设备为数据寄存器。

打开PIDD指令的初始化输入后，可以“控制器”选项卡及“调谐”选项卡中设置的值，对控制寄存器进行初始化。

有关控制寄存器的说明，请参见第19-37页上的“S1：控制寄存器”。

#### 2. S2（源2）：初始化输入

指定初始化PIDD指令的控制寄存器、控制继电器的设备。

可在初始化输入中指定的设备为外部输入或内部继电器。

PIDD指令对话框的“控制器”选项卡及“调谐”选项卡中设置的参数为PIDD指令的控制寄存器、控制继电器的初始值。这些初始值会作为用户程序下载到控制器中，并存储在控制器的ROM内。当初始化输入打开时，将ROM内的PIDD指令的初始值存储到控制寄存器、控制继电器中。

在初始化输入打开期间，每次扫描都会存储初始值。要对值仅初始化一次，请将初始化输入与SOTU（上升沿微分）指令或SOTD（下降沿微分）指令结合使用。

#### 3. S3（源3）：控制继电器

指定控制PID控制及存储PIDD指令状态的位设备。

以指定的内部继电器为起始，占用32个内部继电器。可指定的设备为内部继电器。无法指定特殊内部继电器。

控制继电器每个位的作用均不相同，通过打开/关闭位，即可进行PIDD指令的正向控制操作/反向控制操作、微分操作的启用/禁用、自调谐的启用/禁用及自动模式/手动模式/串级控制模式的切换等。

有关控制继电器的说明，请参见第19-41页上的“S3：控制继电器”。

## 19: PID控制指令

### 4. PIDD 编号

分配用于识别各 PIDD 指令的固有编号。  
进行串级控制时，以 PIDD 编号指定主机 PIDD 指令。

**注释：** PIDD 编号可在以下范围内指定。

FT2J 型：0 ~ 5

FC6A 型：0 ~ 31

#### ■ “控制器”选项卡

设置 PIDD 指令的控制参数。

为了将“控制器”选项卡中设置的 PIDD 指令的初始设置存储到控制寄存器、控制继电器中，请将用户程序下载到本体后，打开相应 PIDD 指令的初始化输入。

#### 1. 当前值 (PV) (S1+0、S1+1)

从“模拟量”或“数据寄存器”中选择 PID 控制的当前值 (PV)。

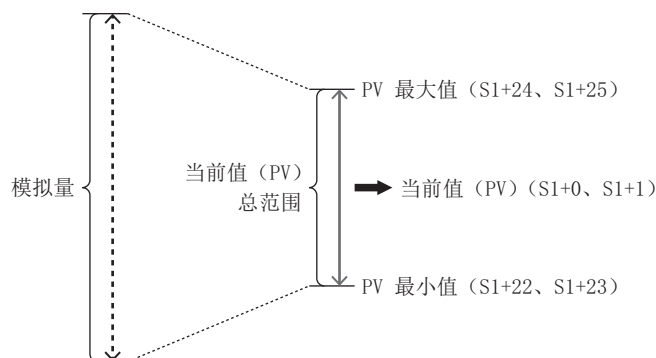
##### 选择了“模拟量”时

作为 PIDD 指令的当前值 (PV)，可指定控制器本体的内置模拟量、模拟输入盒、模拟输入模块及模拟量混合 I/O 模块的模拟量。

项目	说明
节点编号*1	选择模拟输入设备的节点编号。选择节点 0 ~ 10。
设置	选择带模拟输入的设备。 控制器 / CPU/HMI 模块：可以指定控制器中的内置模拟输入端或模拟输入盒的模拟输入端。
	扩展模块 1 ~ 15：如果要指定模拟量输入模块或模拟量混合 I/O 模块的模拟输入，请选择扩展模块 1 ~ 15。
模拟量输入编号	在配置的设备上选择模拟输入编号。

\*1 仅 Plus CPU 模块。有关节点的详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“增设扩展模块组合型”。

当前值 (PV) (S1+0、S1+1) 为, 在 PV 最小值 (S1+22、S1+23) ~ PV 最大值 (S1+24、S1+25) 的范围内将指定模拟量输入的输入值转换为总范围的值。

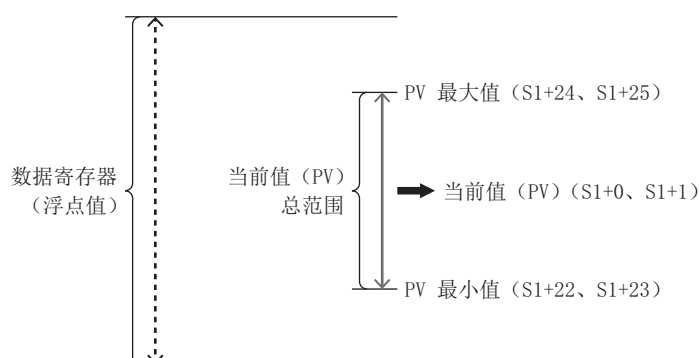


**注释:**

- 使用模拟量输入时, 需要事先设置模拟量输入。有关模拟量输入的设置详情, 请参见以下手册。  
 FT2J 型: 《WindO/I-NV4 用户手册》第 28 章 “2.10 模拟量 / 数字量输入”  
 《SmartAXIS 硬件手册》第 2 章 “2 模拟 I/O 盒”  
 FC6A 型: 《FC6A 型 MICROSmart 用户手册》第 5 章 “内置模拟量输入”、第 9 章 “模拟 I/O 模块” 或第 10 章 “I/O 盒”
- 指定了 “模拟量” 时, 模拟量输入设置的最小值、最大值的设置将变为输入的最小值、输入的最大值。
- 为 FC6A 型指定模拟输入时, 模拟量值 (AI0) 将无法作为测量值使用。

**选择了 “数据寄存器” 时**

将存储在 PIDD 指令指定的数据寄存器为起始的 2 个数据寄存器的值 (数据类型 F (浮点)) 导入为当前值。如果数据寄存器的值高于 PV 最大值, 则 PV 最大值为当前值, 如果低于 PV 最小值, 则 PV 最小值为当前值。当前值会存储到当前值 (PV) (S1+0、S1+1) 中。



**2. PV 最大值 (URV) (S1+24、S1+25)、PV 最小值 (LRV) (S1+22、S1+23)**

以最大值、最小值设置当前值 (PV) (S1+0、S1+1) 的范围 (总范围)。

如果当前值高于 PV 最大值 (S1+24、S1+25), 则 PV 最大值为当前值 (PV) (S1+0、S1+1), 如果低于 PV 最小值 (S1+22、S1+23), 则 PV 最小值为当前值 (PV) (S1+0、S1+1)。此时, 错误代码 109 会存储到错误状态 (S1+34、S1+35) 中。

在 -32,768.0 ~ 65,535.0 的范围内, 以数据类型 F (浮点) 设置 PV 最大值 (S1+24、S1+25) 及 PV 最小值 (S1+22、S1+23)。

## 19: PID控制指令

### 3. 控制模式和设置值 (S3+1 ~ S3+3)

指定 PID 控制模式和设置值的浏览位置。

选择“自动模式 - LSP (S1+2、S1+3)”或“串级控制 - RSP (S1+4、S1+5)”。

#### 选择了“自动模式 - LSP (S1+2、S1+3)”时

PIDD 指令为自动模式。自动模式指示 (S3+2) 打开，手动模式指示 (S3+1) 和串级控制模式指示 (S3+3) 关闭。

请在 SP 下限 ~ SP 上限的范围内将设定值存储到设定值 (SP) (S1+2、S1+3) 中。

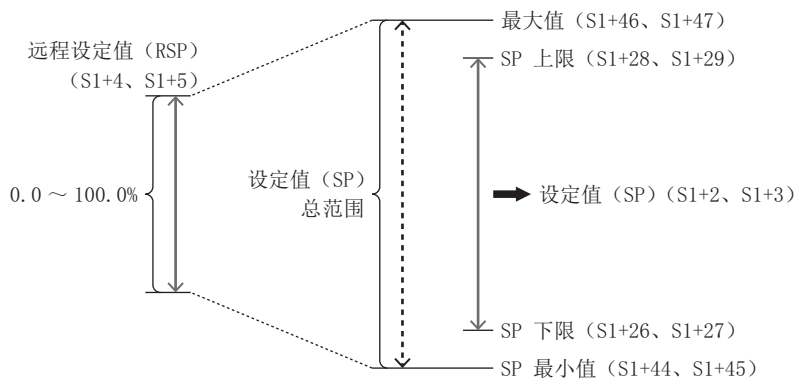
如果设定值 (SP) (S1+2、S1+3) 的值高于 SP 上限，则 SP 上限为 PID 控制的设定值，如果低于 SP 下限，则 SP 下限为 PID 控制的设定值。设定值超出上限、下限范围时，错误代码 106 会存储到错误状态 (S1+34、S1+35) 中。

#### 选择了“串级控制 - RSP (S1+4、S1+5)”时

PIDD 指令为串级控制模式。串级控制模式指示 (S3+3) 打开，手动模式指示 (S3+1) 和自动模式指示 (S3+2) 关闭。

请在 0.0 ~ 100.0% 的范围内将设定值存储到远程设定值 (RSP) (S1+4、S1+5) 中。以主机 PIDD 编号指定了其他 PIDD 指令时，指定的 PIDD 指令的输出操作变量 (MV) (S1+16、S1+17) 会自动导入到远程设定值 (RSP) (S1+4、S1+5) 中。

远程设定值 (RSP) (S1+4、S1+5) 可转换为 SP 最小值 (S1+44、S1+45) ~ SP 最大值 (S1+46、S1+47) 的总范围。如果转换后的设定值高于 SP 上限 (S1+28、S1+29)，则 SP 上限为设定值，如果低于 SP 下限 (S1+26、S1+27)，则 SP 下限为设定值。转换后的设定值会存储到设定值 (SP) (S1+2、S1+3) 中。



### 4. 控制操作 (S3+0)

选择“反向控制操作”或“正向控制操作”。选择了“反向控制操作”时，控制操作 (S3+0) 关闭。选择了“正向控制操作”时，控制操作 (S3+0) 打开。

反向控制操作时，一旦 PIDD 指令的当前值 (PV) 大于设定值 (SP)，输出操作变量 (MV) 就会变小。反向控制操作可用于加热控制等。

正向控制操作时，一旦 PIDD 指令的当前值 (PV) 大于设定值 (SP)，输出操作变量 (MV) 也会随之变大。正向控制操作可用于冷却控制等。

### 5. Kp 依存 (S3+8)

选择“依存”或“独立”。选择了“依存”时，Kp 依存 (S3+8) 打开。选择了“独立”时，Kp 依存 (S3+8) 关闭。Kp 依存时，积分操作及微分操作会与 Kp (增益) (S1+6、S1+7) 成正比变大。Kp 独立时，积分操作及微分操作均不会受 Kp (增益) (S1+6、S1+7) 的影响。

**注释：**自调谐启用时，自动确定比例增益依存 / 独立 (S3+8)。

比例增益  $\geq 1$  时，S3+8 打开。

比例增益  $< 1$  时，S3+8 关闭。

### 6. SP 上限 (S1+28、S1+29)、SP 下限 (S1+26、S1+27)

设置当前值 (SP) (S1+2、S1+3) 的上限、下限。

如果设定值 (SP) (S1+2、S1+3) 的值高于 SP 上限 (S1+28、S1+29)，则 SP 上限为 PIDD 控制的设定值，如果低于 SP 下限 (S1+26、S1+27)，则 SP 下限为 PIDD 控制的设定值。

在 SP 最小值 (S1+44、S1+45) ~ SP 最大值 (S1+46、S1+47) 的范围内，以数据类型 F (浮点) 设置 SP 上限 (S1+28、S1+29) 及 SP 下限 (S1+26、S1+27)。

### 7. 设定值 (SP) (S1+2、 S1+3)

设置 PID 控制的设定值。

在 SP 下限 (S1+26、 S1+27) ~ SP 上限 (S1+28、 S1+29) 的范围内，以数据类型 F (浮点) 设置设定值 (SP) (S1+2、 S1+3)。

### 8. MV 上限 (S1+32、 S1+33)、 MV 下限 (S1+30、 S1+31)

设置输出操作变量 (MV) (S1+16、 S1+17) 的上限、下限。

如果以 PID 控制计算出的输出操作变量 (0 ~ 100%) 高于 MV 上限 (S1+32、 S1+33)，则 MV 上限为输出操作变量 (MV) (S1+16、 S1+17)，如果低于 MV 下限 (S1+30、 S1+31)，则 MV 下限为输出操作变量 (MV) (S1+16、 S1+17)。

在 0.0 ~ 100.0 (0.0 ~ 100.0%) 的范围内，以数据类型 F (浮点) 设置 MV 上限 (S1+32、 S1+33) 及 MV 下限 (S1+30、 S1+31)。

### 9. 输出操作变量 (MV) (S1+16、 S1+17)

选中“模拟值”复选框，即可从“模拟输出”或“数据寄存器”中选择模拟值进行输出。

取消“模拟值”复选框，则不输出模拟值。

选中“数字值”复选框，即可打开 / 关闭控制输出 (S3+14)。

取消“数字值”复选框，则不打开 / 关闭控制输出 (S3+14)。

#### “模拟输出”时

设置模块及其节点和其模拟编号，该模块拥有希望设置为输出操作变量 (模拟值) 输出端口的模拟量输出。可以指定控制器中的内置模拟输出端、模拟输出盒、模拟输出模块或模拟混合 I/O 模块的模拟输出端。

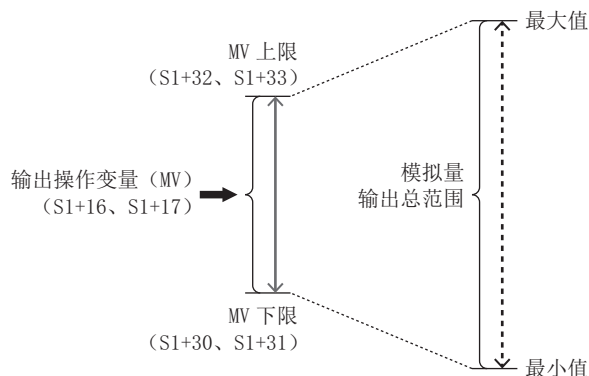
项目	说明
节点编号*1	选择模拟输出设备的节点编号。选择节点 0 ~ 10。
设置	选择带模拟输出的设备。 控制器 / CPU/HMI 模块：可以指定控制器中的内置模拟输出端或模拟输出盒的模拟输出端。 扩展模块 1 ~ 15：如果要指定模拟量输出模块或模拟量混合 I/O 模块的模拟输出，请选择扩展模块 1 ~ 15。
模拟量输出编号	在配置的设备上选择模拟输出编号。

\*1 仅 Plus CPU 模块。有关节点的详情，请参见《FC6A 型 MICROSmart 用户手册》第 2 章“增设扩展模块组合型”。

- 自动模式时，在模拟量输出的最小值~最大值范围内将输出操作变量 (S1+16、 S1+17) 转换为总范围，并通过已指定的模拟量输出编号进行输出。
- 手动模式时，在模拟量输出的最小值~最大值范围内将手动模式输出操作变量 (S1+14、 S1+15) 转换为总范围，并通过模拟量输出编号进行输出。

此外，转换为总范围的模拟量输出值可通过输出操作变量 (模拟值) (S1+66、 S1+67) 进行确认。

有关自动 / 手动模式的切换，请参见第 19-41 页上的“S3：控制继电器”。



## 19: PID控制指令

**注释:** 使用模拟量输出时, 需要事先设置模拟量输出。有关模拟量输出的设置的详情, 请参见以下手册。

FT2J 型: 《WindO/I-NV4 用户手册》第 28 章 “2.11 模拟量输出”

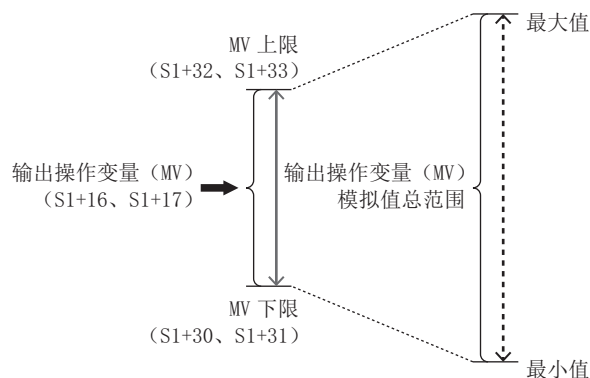
《SmartAXIS 硬件手册》第 2 章 “2 模拟 I/O 盒”

FC6A 型: 《FC6A 型 MICROSmart 用户手册》第 9 章 “模拟 I/O 模块” 或第 10 章 “I/O 盒”

### “数据寄存器”时

- 自动模式时, 在最小值~最大值范围内将输出操作变量 (MV) (S1+16、S1+17) 转换为总范围后的值, 将存储在输出操作变量 (MV) 模拟值 (S1+66、S1+67) 中。
- 手动模式时, 在最小值~最大值范围内将手动模式输出操作变量 (S1+14、S1+15) 转换为总范围后的值, 将存储在输出操作变量 (MV) 模拟值 (S1+66、S1+67) 中。

有关自动 / 手动模式的切换, 请参见第 19-41 页上的 “S3: 控制继电器”。



### “数字值”时

- 自动模式时, 通过输出操作变量 (MV) (S1+16、S1+17)、控制周期 (S1+64、S1+65) 打开 / 关闭控制输出 (S3+14)。控制周期对应的打开脉宽根据输出操作变量而变化。
  - 手动模式时, 通过手动模式输出操作变量 (S1+14、S1+15)、控制周期 (S1+64、S1+65) 打开 / 关闭控制输出 (S3+14)。控制周期对应的打开脉宽根据手动模式输出操作变量 (S1+14、S1+15) 而变化。
- 输出操作变量 (MV) 选择为数据寄存器时, 可设置设备地址 (10)、最大值、最小值 (11)。

### 10. 设备地址

显示存储输出操作变量 (MV) 模拟值 (S1+66、S1+67) 的设备地址。

### 11. 最大值、最小值

最大值、最小值设置用于从输出操作变量 (MV) (S1+16、S1+17) 总范围转换为输出操作变量 (MV) 模拟值 (S1+66、S1+67)。在 -32768.0 ~ 65535.0 范围内进行设置。

输出操作变量 (MV) 选择为数字值时, 可设置设备地址 (12)、控制周期 (S1+64、S1+65) (13)。

### 12. 设备地址

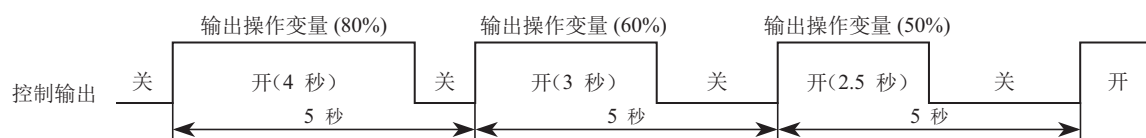
显示控制输出 (S3+14) 的设备地址。

### 13. 控制周期 (S1+64、S1+65)

设置通过输出操作变量 (MV) (S1+16、S1+17) 对控制输出 (S3+14) 进行打开 / 关闭控制的周期。控制周期的打开脉宽随输出操作变量而发生变化。

在 0.1 ~ 50.0 秒的范围内, 通过数据类型 F (浮点) 以 0.1 秒为单位设置控制周期。

控制周期: 5.0 秒 (预置值 5.0) 时



## ■ “调谐”选项卡

设置 PID 指令的调谐参数。

为了将“调谐”选项卡中设置的 PID 指令的初始设置存储到控制寄存器、控制继电器中，请在将用户程序下载到本体后打开相应 PID 指令的初始化输入。

### 1. Kp（增益）（S1+6、S1+7）

设置 PID 控制的比例增益。比例操作是指这一类操作，其输出变化的幅度与设定值（SP）和当前值（PV）的差（以后为偏移量）成正比。当前值在比例带的范围内时，输出与偏差成正比的操作变量。

缩小比例增益时，过冲（当前值高于设定值）、下冲（当前值低于设定值）及速度偏差（当前值不稳定，处于起伏状态）会变少，但当前值达到设定值会较为费时，而且设定值和当前值的偏移量还会变大。扩大比例增益时，当前值达到设定值的时间会变短，偏移量也会变小，但速度偏差会变大。

在 0.00001 ~ 100.0000（0.00001 ~ 100.0000%）的范围内，以数据类型 F（浮点）设置比例增益（S1+6、S1+7）。

### 2. Ki（积分）（S1+8、S1+9）

设置 PID 控制的积分增益。积分增益是由积分操作决定输出操作变量的系数。如果只有比例操作，即使控制对象达到稳定状态，设定值和当前值之间仍会产生一定的偏移量。为了使该差接近 0，需要进行积分操作。

如果扩大积分增益，则积分操作就会变强从而快速达到设定值，但会引起长周期的速度偏差。一旦积分增益变小，达到设定值就会非常费时。

在 0.00001 ~ 100.0000（0.00001 ~ 100.0000 回/分）的范围内，以数据类型 F（浮点）设置积分增益（S1+8、S1+9）。

### 3. Kd（微分）（S1+10、S1+11）

设置 PID 控制的微分增益。微分增益是由微分操作决定输出操作变量的系数。如果因干扰等变为当前值，则微分操作就会使输出操作变量发生变化，是一种快速使当前值稳定的操作。

如果缩小微分增益，则微分操作变弱，针对当前值急剧变化的响应就会变慢。如果扩大微分增益，则微分操作变强，针对急剧的当前值变化的响应就会变快，但当前值的变化变得敏感，输出操作变量的变化变得激烈。

在 0.00001 ~ 100.0000（0.00001 ~ 100.0000 秒）的范围内，以数据类型 F（浮点）设置微分增益（S1+10、S1+11）。

**注释：**自调谐启用时（自调谐 启用 / 禁用（S3+15）打开时）微分增益固定为 0.00001。

### 4. 禁用 Kd (微分) (S3+9)

选中“禁用 Kd (微分)”复选框，即可禁用微分操作。微分操作 (S3+9) 打开。  
取消“禁用 Kd (微分)”复选框，即可启用微分操作。微分操作 (S3+9) 关闭。

### 5. 禁用 Kd (微分) 衰减 (S3+10)

选中“禁用 Kd (微分) 衰减”复选框，即可禁用 Kd (微分) 衰减。Kd (微分) 衰减 (S3+10) 打开。  
取消“禁用 Kd (微分) 衰减”复选框，即可启用 Kd (微分) 衰减。Kd (微分) 衰减是针对微分操作的过滤器，可防止输出操作变量的急剧变动。Kd (微分) 衰减 (S3+10) 关闭。

### 6. 启用 PV 跟踪 (S3+7)

选中“启用 PV 跟踪”复选框，即可启用 PV 跟踪。PV 跟踪 (S3+7) 打开。PID 指令在手动模式下启用 PV 跟踪时，将当前值 (PV) (S1+0、S1+1) 的值复制到设定值 (SP) (S1+2、S1+3) 中。要从手动模式切换为自动模式时，为了当前值 (PV) 与设定值 (SP) 间不产生偏移量，将维持输出操作变量 (MV)。

取消“启用 PV 跟踪”复选框，即可禁用 PV 跟踪。PV 跟踪 (S3+7) 关闭。

### 7. 启用自调谐 (S3+15)

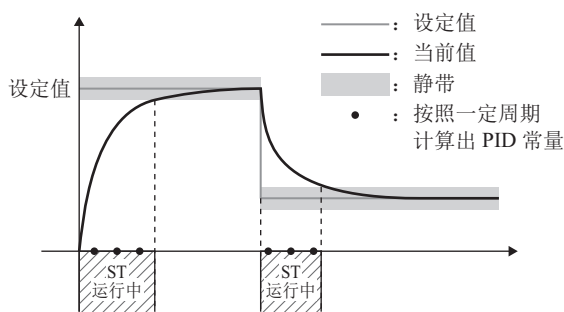
选中“启用自调谐”复选框后，执行自调谐。取消勾选“启用自调谐”复选框后，不执行自调谐。

执行自调谐时，自调谐 启用 / 禁用 (S3+15) 打开，不执行自调谐时关闭。

仅在自动模式下可执行自调谐。手动模式下无法执行自调谐。

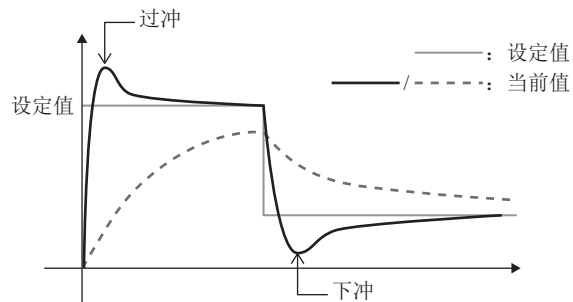
自调谐 (ST) 不会对控制目标带来较大变动，它是可以始终执行最适当控制的一种调整功能。始终监控控制目标，根据控制目标的特性，按照一定周期计算出比例增益 (Kp)、积分增益 (Ki)、微分增益 (Kd) 并进行设置。

自调谐启用时



可进行最适当控制。

自调谐禁用时



PID 常量会导致发生过冲和下冲，在达到设定值之前需要一定时间。

**注释：**自调谐启用时，自调谐 启用 / 禁用 (S3+15) 打开。

### 8. ST 静带 (S1+52、S1+53)

设置 ST 静带。请使用数据类型 F (浮点) 在当前值 (PV) 总范围的 0.2000000 ~ 20.00000 (0.2000000 ~ 20.00000%) 的这一范围内进行设置。

ST 静带是指启用自调谐时不执行自调谐的区域。依据自调谐 启用 / 禁用 (S3+15) 的状态判断自调谐的启用 / 禁用。

设定值与当前值的差称为偏差。下面介绍偏差与 ST 静带的关系。

#### | 偏差 | > ST 静带

执行自调谐，在执行自调谐 执行中 (S3+17)。比例增益或积分增益更改时，仅 1 次扫描打开自调谐 增益更改 (S3+16)。

#### | 偏差 | ≤ ST 静带

不执行自调谐。在执行自调谐 增益更改 (S3+16) 和自调谐 执行中 (S3+17) 不打开。



**9. ST 周期 (S1+68、S1+69)**

自调谐启用时，设置执行自调谐的周期。请使用数据类型 F（浮点）在 8.0 ~ 300.0（8.0 ~ 300.0 秒）的范围内进行设置。

需要快速结束时请设置较小的值，需要平稳结束时请设置较大的值。

**10. ST 更新周期 (S1+54、S1+55)**

自调谐启用时，执行自调谐，设置更新比例增益和积分增益的周期。请使用数据类型 F（浮点）在 2.0 ~ 8.0（2.0 ~ 8.0 秒）的范围内进行设置。

**11. Kp 上限 (S1+58、S1+59)、Kp 下限 (S1+56、S1+57)**

设置自调谐过程中比例增益的最大值和最小值。

请使用数据类型 F（浮点）在 0.0100000 ~ 12.00000（0.0100000 ~ 12.00000%）的范围内进行设置。

**12. Ki 上限 (S1+62、S1+63)、Ki 下限 (S1+60、S1+61)**

设置自调谐过程中积分增益的最大值和最小值。

请使用数据类型 F（浮点）在 0.1000000 ~ 40.00000（0.1000000 ~ 40.00000 回/分）的范围内进行设置。

**13. 默认值**

将 ST 静带、ST 周期、ST 更新周期、Kp 上限、Kp 下限、Ki 上限和 Ki 下限恢复为默认值。

参数	默认值
ST 静带	2.0000 (EU)
ST 周期	15 (秒)
ST 更新周期	4 (秒)
Kp 上限	6.00 (%)
Kp 下限	0.01 (%)
Ki 上限	15.0 (回/分)
Ki 下限	0.1 (回/分)

**S1: 控制寄存器**

存储目的地	功能	说明	读/写	执行中更改
S1+0、S1+1	当前值 (PV)	可存储 PID 控制的当前值。 只读。	读	—
S1+2、S1+3	设定值 (SP)	设置 PID 控制的设定值。请在 SP 下限 ~ SP 上限的范围内设置设定值。 如果设定值高于 SP 上限，则会以 SP 上限进行操作，如果低于 SP 下限，则会以 SP 下限进行操作。设定值超出范围时，错误代码 106 会存储到错误状态 (S1+34、S1+35) 中。	读/写	是
S1+4、S1+5	远程设定值 (RSP)	请在 0.0 ~ 100.0% 的范围内设置主机 PID 指令的输出操作值。 串级控制模式时，远程设定值 (RSP) 的值可转换为 SP 最小值 (S1+44、S1+45) ~ SP 最大值 (S1+46、S1+47) 的总范围。如果转换后的设定值高于 SP 上限 (S1+28、S1+29)，则 SP 上限为设定值，如果低于 SP 下限 (S1+26、S1+27)，则 SP 下限为设定值。转换后的设定值会存储到设定值 (SP) (S1+2、S1+3) 中。	读/写	是
S1+6、S1+7	Kp (增益)	请在 0.00001 ~ 100.0 (0.00001 ~ 100.0%) 的范围内设置比例增益。 如果比例增益的值低于 0.00001，则以 0.00001% 进行操作，如果高于 100.0，则以 100.0% 进行操作。超出范围时，错误代码 122 会存储到错误状态 (S1+34、S1+35) 中。	读/写	是
S1+8、S1+9	Ki (积分)	请在 0.00001 ~ 100.0 (0.00001 ~ 100.0 回/分) 的范围内设置积分增益。 如果积分增益的值低于 0.00001，则以 0.00001 回/分进行操作，如果高于 100.0，则以 100.0 回/分进行操作。	读/写	是

## 19: PID控制指令

存储目的地	功能	说明	读/写	执行中更改
S1+10、 S1+11	Kd (微分)	请在 0.00001 ~ 100.0 (0.00001 ~ 100.0 秒) 的范围内设置微分增益。 如果微分增益的值低于 0.00001, 则以 0.00001 秒进行操作, 如果高于 100.0, 则以 100.0 秒进行操作。	读/写	是
S1+12、 S1+13	— 保留 —	—	—	—
S1+14、 S1+15	手动输出操作变量	PIDD 指令为手动模式时, 请在 0.0 ~ 100.0% 的范围内设置输出操作变量。 如果手动输出操作变量的值低于 0.0%, 则作为 0.0 进行操作, 如果高于 100.0, 则作为 100.0 进行操作。超出范围时, 错误代码 130 会存储到错误状态 (S1+34、S1+35) 中。	读/写	是
S1+16、 S1+17	输出操作变量 (MV)	可存储 PIDD 指令的输出操作变量。 可存储 MV 下限 (S1+30、S1+31) ~ MV 上限 (S1+32、S1+33) 的范围内的值。	读	—
S1+18 ~ S1+21	— 保留 —	—	—	—
S1+22、 S1+23	PV 最小值 (LRV)	请在 -32,768.0 ~ 65,535.0 的范围内设置 PV 最小值。 设置的 PV 最小值超出 -32,768.0 ~ 65,535.0 范围时或 PV 最大值 $\leq$ PV 最小值时, 会恢复为更改前的 PV 最小值, 并继续执行 PID 控制。设置的 PV 最小值超出范围时, 错误代码 102 会存储到错误状态 (S1+34、S1+35) 中。 但在执行 PIDD 指令第 1 次扫描时, PV 最小值超出范围或 PV 最大值 $\leq$ PV 最小值时, PID 控制的执行会停止。变为范围内的值不发生错误时, 开始执行 PID 控制。	读/写	是
S1+24、 S1+25	PV 最大值 (URV)	请在 -32,768.0 ~ 65,535.0 的范围内设置 PV 最大值。 设置的 PV 最大值超出 -32,768.0 ~ 65,535.0 范围时或 PV 最大值 $\leq$ PV 最小值时, 会恢复为更改前的 PV 最大值, 并继续执行 PID 控制。设置的 PV 最大值超出范围时, 错误代码 102 会存储到错误状态 (S1+34、S1+35) 中。 但在执行 PIDD 指令第 1 次扫描时, PV 最大值超出范围或 PV 最大值 $\leq$ PV 最小值时, PID 控制的执行会停止。变为范围内的值不发生错误时, 开始执行 PID 控制。	读/写	是
S1+26、 S1+27	SP 下限	请在 SP 最小值 (S1+44、S1+45) ~ SP 最大值 (S1+46、S1+47) 的范围内设置 SP 下限。 如果 SP 下限低于 SP 最小值 (S1+44、S1+45), 则会作为 SP 最小值进行操作, 如果高于 SP 最大值 (S1+46、S1+47), 则会作为 SP 最大值进行操作。SP 上限 $\leq$ SP 下限时, 会作为 SP 最小值进行操作。此时, 错误代码 106 会存储到错误状态 (S1+34、S1+35) 中。	读/写	是
S1+28、 S1+29	SP 上限	请在 SP 最小值 (S1+44、S1+45) ~ SP 最大值 (S1+46、S1+47) 的范围内设置 SP 上限。 如果 SP 上限低于 SP 最小值 (S1+44、S1+45), 则会作为 SP 最小值进行操作, 如果高于 SP 最大值 (S1+46、S1+47), 则会作为 SP 最大值进行操作。SP 上限 $\leq$ SP 下限时, 会作为 SP 最大值进行操作。此时, 错误代码 106 会存储到错误状态 (S1+34、S1+35) 中。	读/写	是
S1+30、 S1+31	MV 下限	请在 0.0 ~ 100.0 (0.0 ~ 100.0%) 的范围内设置 MV 下限。 如果 MV 下限的值低于 0.0 或高于 100.0, 则均会作为 0.0 进行操作。MV 上限 $\leq$ MV 下限时, MV 下限会作为 0.0 进行操作, 而 MV 上限会作为 100.0 进行操作。此时, 错误代码 103 会存储到错误状态 (S1+34、S1+35) 中。	读/写	是

存储目的地	功能	说明	读 / 写	执行中更改
S1+32、 S1+33	MV 上限	请在 0.0 ~ 100.0 (0.0 ~ 100.0%) 的范围内设置 MV 上限。如果 MV 上限的值低于 0.0 或高于 100.0, 则均会作为 100.0 进行操作。MV 上限 $\leq$ MV 下限时, MV 下限会作为 0.0 进行操作, 而 MV 上限会作为 100.0 进行操作。此时, 错误代码 103 会存储到错误状态 (S1+34、S1+35) 中。	读 / 写	是
S1+34、 S1+35	错误状态	可以数据类型 D (双字) 存储 PID 指令的错误状态。有关错误代码的详情, 请参见第 19-40 页上的“错误状态 (S1+34、S1+35)”。	读	—
S1+36 ~ S1+43	— 保留 —	—	—	—
S1+44、 S1+45	SP 最小值	可存储与 PV 最小值 (S1+22、S1+23) 相同的值。	读	—
S1+46、 S1+47	SP 最大值	可存储与 PV 最大值 (S1+24、S1+25) 相同的值。	读	—
S1+48、 S1+49	PID 控制停止时的输出操作变量	请打开 PID 控制停止 (S3+11), 在 0.0 ~ 100.0 (0.0 ~ 100.0%) 的范围内, 以数据类型 F (浮点) 设置停止 PID 控制期间的输出操作变量。PID 控制停止 (S3+11) 关闭期间, PID 指令的输出操作变量 (S1+16、S1+17) 的值会复制到“PID 控制停止时的输出操作变量 (S1+48、S1+49)”中。	读 / 写	是
S1+50、 S1+51	偏差 (偏移量)	可存储针对设定值 (SP) 与当前值 (MV) 的差 (偏移量) 的当前值总范围的比率 (%)。	读	—
S1+52、 S1+53	ST 静带	在当前值 (PV) 总范围的 0.2~20.0 (0.2~20.0%) 的范围内设置 ST 静带。ST 静带的值低于 0.2 时按照 0.2% 运行, 高于 20.0 时按照 20.0% 运行。不在范围内时, 错误代码 110 会存储到错误状态 (S1+34、S1+35) 中。	读 / 写	是
S1+54、 S1+55	ST 更新周期	在 2~8 (2~8 秒) 的范围内设置 ST 更新周期。ST 更新周期的值低于 2 时按照 2 秒运行, 高于 8 时按照 8 秒运行。	读 / 写	是
S1+56、 S1+57	Kp 下限	在 0.01000~12.00000 (0.01000~12.00000%) 的范围内设置 Kp 下限。Kp 下限低于 0.01000 时按照 0.01000% 运行, 高于 12.00000 时按照 12.00000% 运行。	读 / 写	是
S1+58、 S1+59	Kp 上限	在 0.01000~12.00000 (0.01000~12.00000%) 的范围内设置 Kp 上限。Kp 上限低于 0.01000 时按照 0.01000% 运行, 高于 12.00000 时按照 12.00000% 运行。	读 / 写	是
S1+60、 S1+61	Ki 下限	在 0.10000~40.00000 (0.10000~40.00000 回 / 分) 的范围内设置 Ki 下限。Ki 下限低于 0.10000 时按照 0.10000 回 / 分运行, 高于 40.00000 时按照 40.00000 回 / 分运行。	读 / 写	是
S1+62、 S1+63	Ki 上限	在 0.10000~40.00000 (0.10000~40.00000 回 / 分) 的范围内设置 Ki 上限。Ki 上限低于 0.10000 时按照 0.10000 回 / 分运行, 高于 40.00000 时按照 40.00000 回 / 分运行。	读 / 写	是
S1+64、 S1+65	控制周期	设置通过输出操作变量对控制输出 (S3+14) 进行打开 / 关闭控制的周期。请在 0.1 ~ 50.0 的范围内, 通过数据类型 F (浮点) 进行设置。小于 0.1 时作为 0.1 秒、大于 50.1 时作为 50.0 秒动作。此时, 错误代码 132 会存储到错误状态 (S1+34、S1+35) 中。	读 / 写	—

## 19: PID控制指令

存储目的地	功能	说明	读/写	执行中更改
S1+66、 S1+67	输出操作变量 (MV) 模拟值	<ul style="list-style-type: none"> <li>在输出操作变量中指定“模拟量输出”时存储在模拟量输出的最大值~最小值范围内从输出操作变量 (S1+16、S1+17) 转换为总范围后的值。</li> <li>在输出操作变量中指定“数据寄存器”时存储在最大值~最小值范围内从输出操作变量 (S1+16、S1+17) 转换为总范围后的值。</li> </ul>	读	—
S1+68、 S1+69	ST 周期	在 8~300 (8~300 秒) 的范围内设置 ST 周期。ST 周期低于 8 时按照 8 秒运行, 高于 300 时按照 300 秒运行。	读/写	是
S1+70 ~ S1+99	— 保留 —	—	—	—

### 错误状态 (S1+34、S1+35)

表示 PID 指令的错误状态。

错误代码	状态说明	对策	状态分类	WindLDR 的错误检查
101	未设置作为“模拟量”指定的模拟量。	请确认模拟 I/O 模块的设置。	PID 控制执行停止	有
102	启动 PID 指令时, 在 PV 最大值、PV 最小值的设置中设置了超出范围的值。	请确认 PV 最大值、PV 最小值是否进行了超出范围的设置, 或是否 PV 最小值 $\geq$ PV 最大值。	PID 控制执行停止	有
	PID 指令执行中, 在 PV 最大值、PV 最小值中设置了超出范围的值。		PID 控制执行继续	
103	在 MV 上限、MV 下限中设置了超出范围的值。	请确认 MV 上限、MV 下限是否进行了超出范围的设置, 或是否 MV 下限 $\geq$ MV 上限。	PID 控制执行继续	有
106	在 SP 上限、SP 下限中设置了超出范围的值。	请确认 SP 上限、SP 下限是否进行了超出范围的设置, 或是否 SP 下限 $\geq$ SP 上限。	PID 控制执行继续	有
109	在当前值 (PV) 中设置了超出 PV 最小值~PV 最大值范围的值。	请确认是否对当前值进行了超出 PV 最大值、PV 最小值范围的设置。	PID 控制执行继续	无
110	在 ST 静带上设置了超出范围的值。	请确认 ST 静带的设置。	PID 控制执行继续	有
111	“模拟量输出”中设置的模拟 I/O 盒设置错误。	请确认模拟 I/O 模块的设置。	PID 控制执行停止	有
122	在 Kp (增益)、Kp 下限或 Kp 上限上设置了超出范围的值。	请确认 Kp (增益)、Kp 下限和 Kp 上限的设置。	PID 控制执行继续	有
123	在 Kd (微分) 上设置了超出范围的值。	请确认 Kd (微分) 的设置。	PID 控制执行继续	有
124	在 Ki (积分)、Ki 下限或 Ki 上限上设置了超出范围的值。	请确认 Ki (积分)、Ki 下限和 Ki 上限的设置。	PID 控制执行继续	有
130	在手动模式输出 MV 中设置了超出 0.0 ~ 100.0 范围的值。	请确认手动模式输出 MV 的值。	PID 控制执行继续	无
132	在控制周期中设置了超出 0.1 ~ 50.0 范围的值。	请控制周期的设置。	PID 控制执行继续	有
上述以外	— 保留 —	—	—	—

## S3: 控制继电器

存储目的地	功能	说明	读 / 写	执行中更改
S3+0	控制操作	0 (OFF): 反向控制操作 1 (ON): 正向控制操作	读 / 写	是
S3+1	手动模式指示	请将 PIDD 指令转移至手动模式时设为 1 (ON)。S3+1 会自动变为 0 (OFF)。	读 / 写	是
S3+2	自动模式指示	请将 PIDD 指令转移至自动模式时设为 1 (ON)。S3+2 会自动变为 0 (OFF)。	读 / 写	是
S3+3	串级控制模式指示	请将 PIDD 指令转移至串级控制模式时设为 1 (ON)。S3+3 会自动变为 0 (OFF)。	读 / 写	是
S3+4	手动模式监控	手动模式时为 1 (ON)。 手动模式以外时为 0 (OFF)。	读	—
S3+5	自动模式监控	自动模式时为 1 (ON)。 自动模式以外时为 0 (OFF)。	读	—
S3+6	串级控制模式监控	串级控制模式时为 1 (ON)。 串级控制模式以外时为 0 (OFF)。	读	—
S3+7	PV 跟踪	0 (OFF): 禁用 PV 跟踪 1 (ON): 启用 PV 跟踪	读 / 写	是
S3+8	Kp 依存	0 (OFF): Kp 依存 1 (ON): Kp 独立	读 / 写	是
S3+9	微分操作	0 (OFF): 启用微分操作 1 (ON): 禁用微分操作	读 / 写	是
S3+10	Kd (微分) 衰减	0 (OFF): 启用 Kd (微分) 衰减 1 (ON): 禁用 Kd (微分) 衰减	读 / 写	是
S3+11	PID 控制停止	0 (OFF): PID 控制运行 1 (ON): PID 控制停止	读 / 写	是
S3+12	PID 控制停止监控	PID 控制的参数更改时、PID 控制停止时等, PID 控制会在暂停时为 1 (ON)。	读	—
S3+13	微分操作执行中监控	正在执行微分操作期间为 1 (ON)。	读	—
S3+14	控制输出 (输出操作变量 (MV) 数字值)	根据控制周期和输出操作变量切换 1 (ON) / 0 (OFF)。	读	—
S3+15	自调谐 启用 / 禁用	打开时, 自调谐启用。 关闭时, 自调谐禁用。执行自调谐时, 如果关闭则停止自调谐。	读 / 写	是
S3+16	自调谐 增益更改	执行自调谐过程中, 比例增益或积分增益更改时, 仅仅 1 次扫描打开。	读	—
S3+17	自调谐 执行中	自调谐 执行中打开。自调谐 启用 / 禁用 (S3+15) 打开, 当前值超出 ST 静带范围时, 执行自调谐。	读	—
S3+18 ~ S3+31	— 保留 —	—	—	—

## PID 监控

FT2J FC6A

### 功能说明

可以图表形式监控 PID 控制执行中的 PIDA/PIDD 指令的各参数。此外，可通过“PID 监控”对话框直接更改各 PIDA/PIDD 指令所使用的数据寄存器、内部继电器的值，并能在调整 PID 控制参数的同时确认各 PIDA/PIDD 指令的操作。

**注释：** PID 监控并非按照控制器内部的设置进行操作，而是按照 WindLDR 画面上所显示的梯形图程序上的 PIDA/PIDD 指令设置进行操作。控制器内部的 PIDA/PIDD 指令与 WindLDR 的 PIDA/PIDD 指令分配为不同设备时，将无法正确监控。

### 启动 PID 监控


#### 操作程序

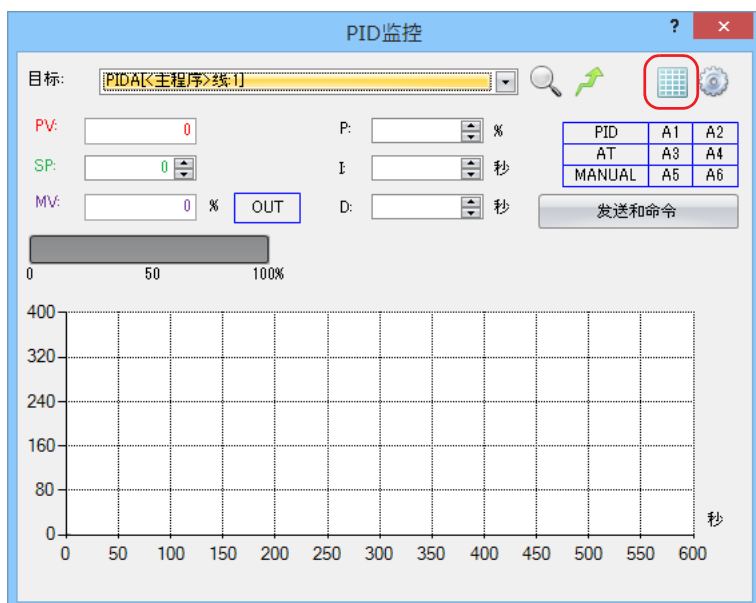
1. 从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”。  
将开始监控。



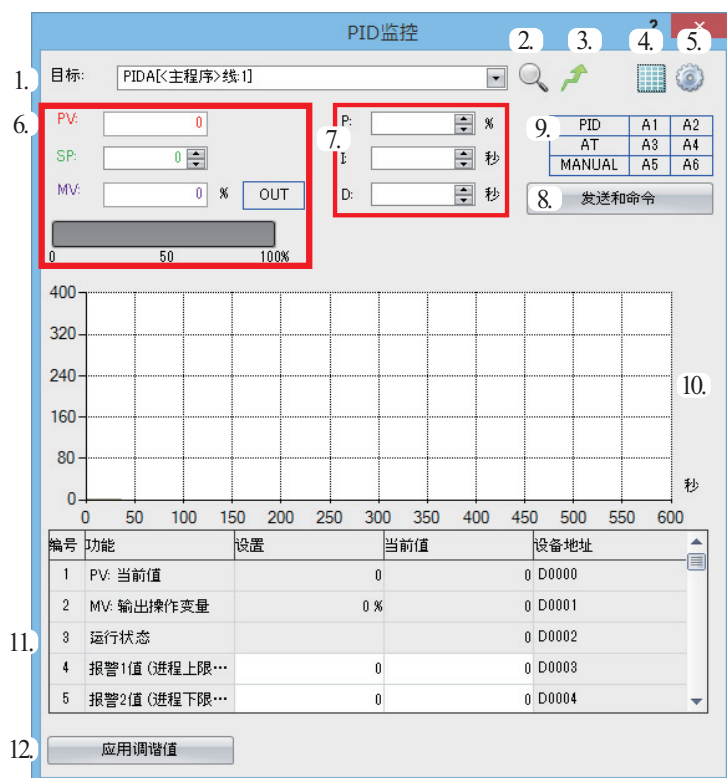
2. 从 WindLDR 菜单栏中选择“联机”>“监控”>“PID 监控”。  
将显示“PID 监控”对话框。



3. 单击“PID 监控”对话框的  (显示 / 隐藏 PID 监控表) 按钮  
将显示 PID 监控表。



## “PID 监控”对话框



## 1. 目标

选择监控对象的 PIDA/PIDD 指令。将显示 WindLDR 中打开的梯形图程序上的所有 PIDA/PIDD 指令。各 PIDA/PIDD 指令可以指令所使用的主程序、子程序等的程序名称和行编号进行识别。

## 2. 参见 PID 设置

可打开正在监控的 PIDA/PIDD 指令对话框，并确认设置。

## 3. 跳转至对象 PID 指令

将梯形图编辑光标移动至正在监控的 PIDA/PIDD 指令位置。

## 4. 显示 / 隐藏 PID 监控表

可显示或隐藏“PID 监控表”（11）、“应用调谐值”按钮（12）。

## 5. PID 监控设置

打开“PID 监控设置”对话框。有关详情，请参见第 19-47 页上的““PID 监控设置”对话框”。

## 6. 主要参数控制板

可确认当前值（PV）、设定值（SP）、操作变量（MV）、ST 静带（DB）的值。此外，可通过横向条形图确认 MV 相关的模拟量值，通过 OUT 指示灯颜色确认 MV 相关的数字值（打开 / 关闭）。OUT 指示灯显示白色时，关闭输出，显示绿色时，打开输出。

测量值（PV）、目标值（SP）、操作量（MV）的值的颜色，可在“PID 监控设置”对话框中进行更改。有关详情，请参见第 19-47 页上的““PID 监控设置”对话框”。

选择 PIDA 指令时，数字值（ON/OFF）可以 OUT 指示器的颜色进行确认。OUT 指示器颜色在输出关闭时为白色，打开时为绿色。

选择 PIDD 指令时，当手动模式打开时即可编辑操作变量（MV）的文本框，并能设置手动输出操作变量（S1+14、S1+15）。单位为%，可在 0.0 ~ 100.0 的范围内进行设置。

## 7. PID 参数控制板

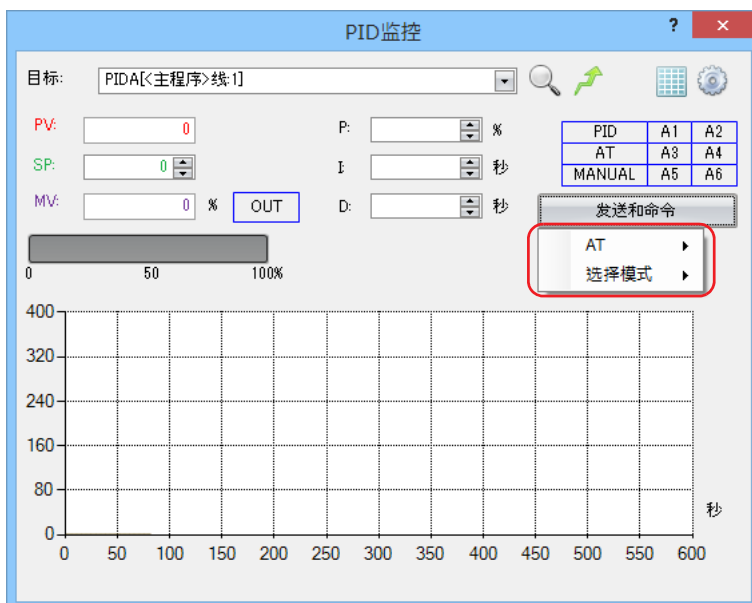
选择 PIDA 指令时，可确认比例带（P）、积分时间（I）、微分时间（D）的值。

选择 PIDD 指令时，可确认 Kp（增益）、Ki（积分）及 Kd（微分）的值。

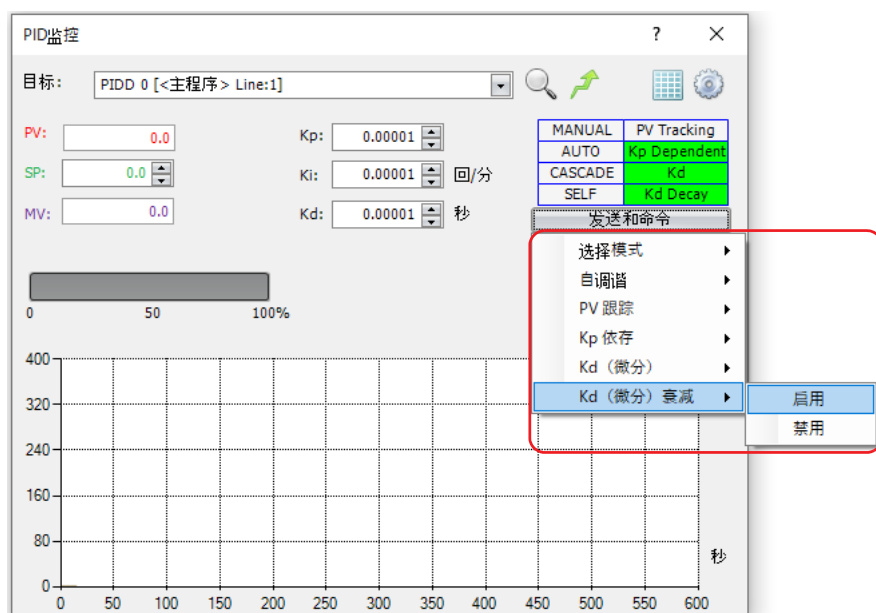
## 19: PID控制指令

### 8. 发送和命令

选择 PIDA 指令时，可将 PID 控制的 AT 的执行 / 停止、手动 / 自动模式切换的指令发送到控制器中。单击该按钮，将显示下拉菜单，单击菜单即可发送指令。



选择 PIDD 指令时，可将 PID 控制的控制模式（手动 / 自动 / 串级）、自调谐的启用 / 禁用、PV 跟踪的启用 / 禁用、比例增益依存 / 独立、微分控制的启用 / 禁用、Kd（微分）衰减的启用 / 禁用切换的指令发送到控制器中。单击此按钮，将显示弹出式菜单，通过单击相应项目即可发送指令。





## 9. 状态指示灯

可按各指示灯的颜色确认 PID 控制的主要状态。

选择 PIDA 指令时，将显示下表的指示器。

指示灯名称	背景颜色	状态
PID	灰色	PID 控制停止中
	绿色	正在执行 PID 控制
AT	灰色	AT 停止中
	绿色	正在执行 AT
MANUAL	灰色	自动模式
	绿色	手动模式
A1	灰色	正常操作
	红色	打开警报 1 输出
A2	灰色	正常操作
	红色	打开警报 2 输出
A3	灰色	正常操作
	红色	打开警报 3 输出
A4	灰色	正常操作
	红色	打开警报 4 输出
A5	灰色	正常操作
	红色	打开警报 5 输出
A6	灰色	正常操作
	红色	打开警报 6 输出

选择 PIDD 指令时，将显示下表的指示器。

指示灯名称	背景颜色	状态
MANUAL	灰色	其他模式
	绿色	手动模式
AUTO	灰色	其他模式
	绿色	自动模式
CASCADE	灰色	其他模式
	绿色	串级控制模式
SELF	灰色	自调谐禁用
	绿色	自调谐启用
PV Tracking	灰色	禁用 PV 跟踪
	绿色	启用 PV 跟踪
Kp Dependent	灰色	Kp 独立
	绿色	Kp 依存
Kd	灰色	禁用微分操作
	绿色	启用微分操作
Kd Decay	灰色	禁用 Kd (微分) 衰减
	绿色	启用 Kd (微分) 衰减

## 10. 趋势图

选择 PIDA 指令时，可使用趋势图监控当前值 (PV)、设定值 (SP)、操作变量 (MV)、报警 1 值 (进程上限报警)、报警 2 值 (进程下限报警)。

选择 PIDD 指令时，可使用趋势图监控当前值 (PV)、设定值 (SP)、操作变量 (MV)、ST 静带 (DB)。各值的显示/隐藏、色彩、上限、下限可在“PID 监控设置”对话框中进行更改。有关详情，请参见第 19-47 页上的““PID 监控设置”对话框”。

**注释：**趋势图上的曲线到达图表右端时，图表中心将移动至图表左端，新的记录信息将显示于图表中心。以往的记录信息可通过移动滚动条进行确认。此外，记录信息最多可保持 10,000 次。记录信息达到 10,000 次时，最早的记录信息将被删除，新记录信息将被记录。关闭对话框或更改监控对象的 PIDA/PIDD 指令时，趋势图上显示的记录信息将被删除。

## 19: PID 控制指令

### 11. PID 监控表

可通过 PIDA/PIDD 指令确认正在使用的数据寄存器、内部继电器的功能名称、当前值、设备地址。以白色单元显示的当前值可进行更改。各项目的显示 / 隐藏可在“PID 监控设置”对话框中进行更改。有关详情，请参见第 19-47 页上的““PID 监控设置”对话框”。

### 12. 应用调谐值

单击“应用调谐值”按钮，可将“PID 监控表”（11）中调整的 PID 控制各参数，反映为“目标”（1）中正在选择的 PIDA/PIDD 指令初始设置值。为了将反映的初始报警值存储到 FC6A 型的 ROM 中，需要退出监控，将用户程序下载到控制器中。

## 更改 PID 监控的设置


### 操作程序

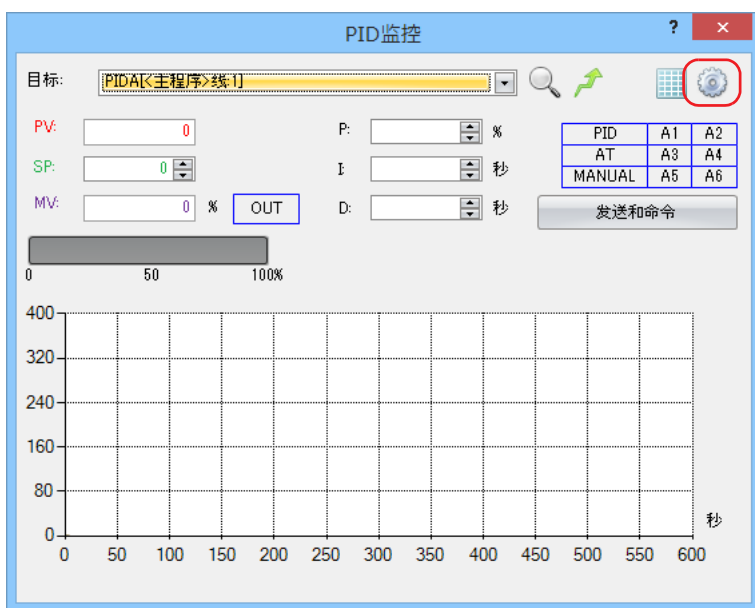
1. 从 WindLDR 菜单栏中选择“联机”>“监控”>“监控”。  
将开始监控。



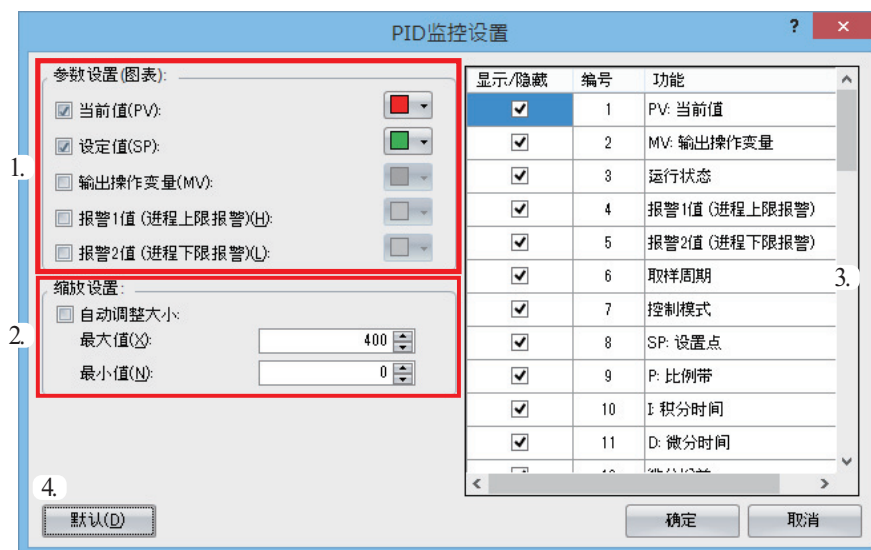
2. 从 WindLDR 菜单栏中选择“联机”>“监控”>“PID 监控”。  
将显示“PID 监控”对话框。



3. 单击“PID 监控”对话框的 （PID 监控设置）按钮  
将显示“PID 监控设置”对话框。



## “PID 监控设置”对话框



## 1. 参数设置（图表）

可切换显示在“PID 监控”对话框趋势图上的参数的显示 / 隐藏及指定色彩。选中各项目复选框则显示，未选中则隐藏。各项目右侧按钮的颜色为项目显示颜色。按下▼按钮，将显示颜色列表，可通过选择颜色更改显示颜色。选择的颜色也可适用于“PID 监控”对话框的主要参数控制板。

选择 PIDA 指令时，可更改当前值（PV）、设定值（SP）、输出操作变量（MV）、报警 1 值（进程上限报警）、报警 2 值（进程下限报警）的设置。

选择 PIDD 指令时，可更改当前值（PV）、设定值（SP）、输出操作变量（MV）、ST 静带（DB）的设置。

## 2. 缩放设置

可设置“PID 监控”对话框的趋势图纵轴缩放比例。选中自动缩放复选框时，将针对趋势图上显示的值，自动调整纵轴的上限值、下限值。未选中复选框时，上限值、下限值为输入的值。

## 3. PID 监控表设置

可设置 PID 监控表项目的显示 / 隐藏。选中各项目复选框则显示，未选中则隐藏。

## 4. 默认

所有设置均可设置初始值。

## 应用实例

以下将对 FC6A 型中使用 PIDA 指令的应用实例进行介绍。

**注释：**根据实际的系统构成及运用状况等应用，需要变更各设置。

关于将控制目标的温度目标值设为 200°C 并进行 PID 控制的应用，将以下列 2 种系统构成为例进行介绍。

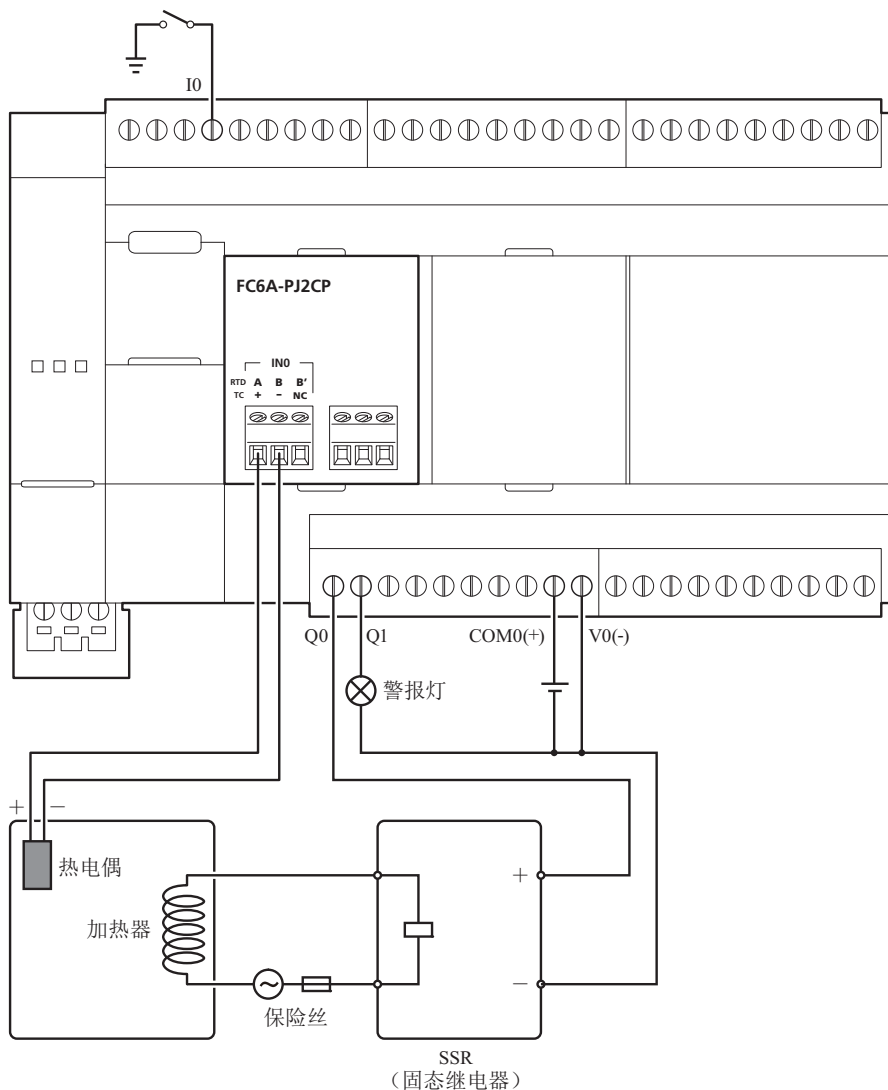
- 通过数字输出的 ON/OFF 进行 PID 控制
- 通过模拟输出进行 PID 控制

## 动作说明

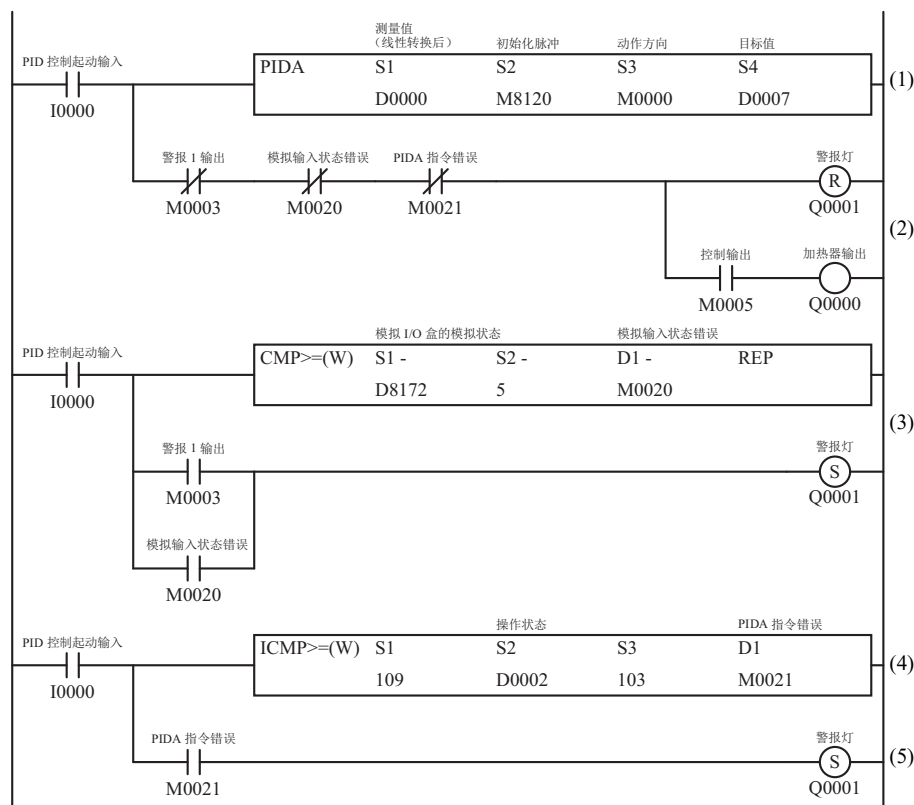
- 以模拟 I/O 盒中输入的温度为基础进行 PID 控制，输出操作变量。
- 目标值为 200°C。
- 将控制模式设为 PID（PID 控制）、动作方向设为反向控制动作。
- 如果测量值变为 250°C，将打开警报 1 输出（S3+3），并停止加热器或进行电力调整。

## 通过数字输出的 ON/OFF 进行 PID 控制

### 系统构成



## 梯形图程序



设备地址	备注
D0000	测量值（线性转换后）
D0002	操作状态
D0007	目标值
D8172	模拟 I/O 盒的模拟状态
I0000	PID 控制起动输入
Q0000	加热器输出
Q0001	警报灯

设备地址	备注
M0000	动作方向
M0003	警报 1 输出
M0005	控制输出
M0020	模拟输入状态错误
M0021	PIDA 指令错误
M8120	初始化脉冲

- (1) I0 为 ON 时，将执行 PIDA 指令。
- (2) M0003（警报 1 输出（进程上限警报））为 OFF 且 M0020 为 OFF（模拟 I/O 盒的模拟输入状态错误为 0～2）时，根据使用 PID 控制计算出的操作量和控制周期，打开和关闭 Q0（加热器输出）。发生异常时，关闭加热器输出。
- (3) 模拟 I/O 盒的模拟输入状态错误为 5、6、8～11 时，打开 M0020。
- (4) M0003（警报 1 输出（进程上限警报））为 ON 或 M0020 为 ON（模拟 I/O 盒的模拟输入状态错误为 5、6、8～11 时，打开 Q1（警报灯）。
- (5) 判断 PIDA 指令的动作状况，并在发生异常时，打开 M0021，且 Q1（警报灯）处于打开状态。

**注释：**模拟 I/O 盒的模拟状态如下所示。

- 模拟输入状态错误
  - 0：正常动作中
  - 1：数据转换中
  - 2：初始化中
  - 5、6：配线异常
  - 8～11：模拟 I/O 盒的相关错误

有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 10 章中的“模拟 I/O 盒”。

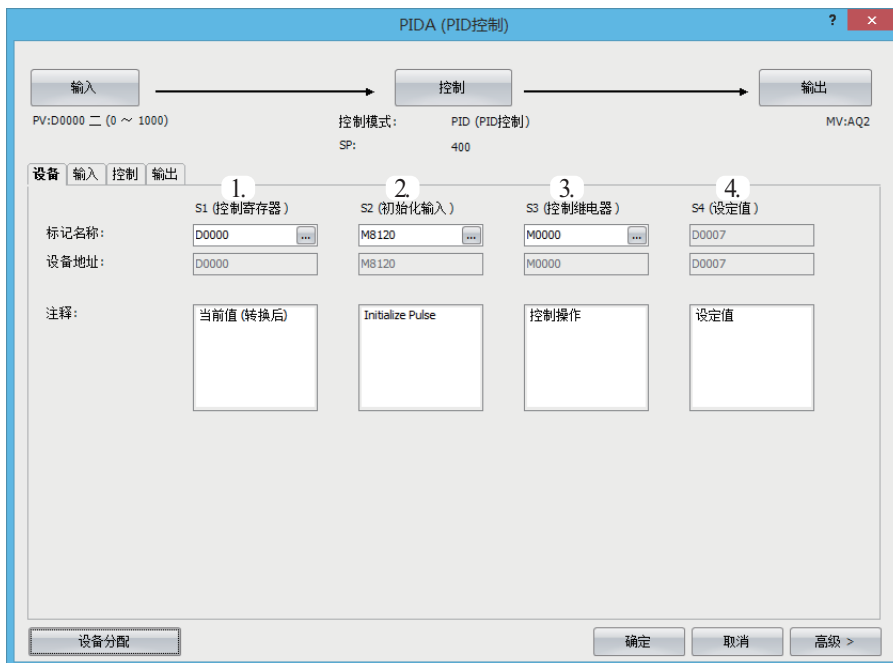
## 19: PID控制指令

### “PIDA (PID 控制)”对话框设置步骤

步骤中未记载的设置项目已使用默认值。

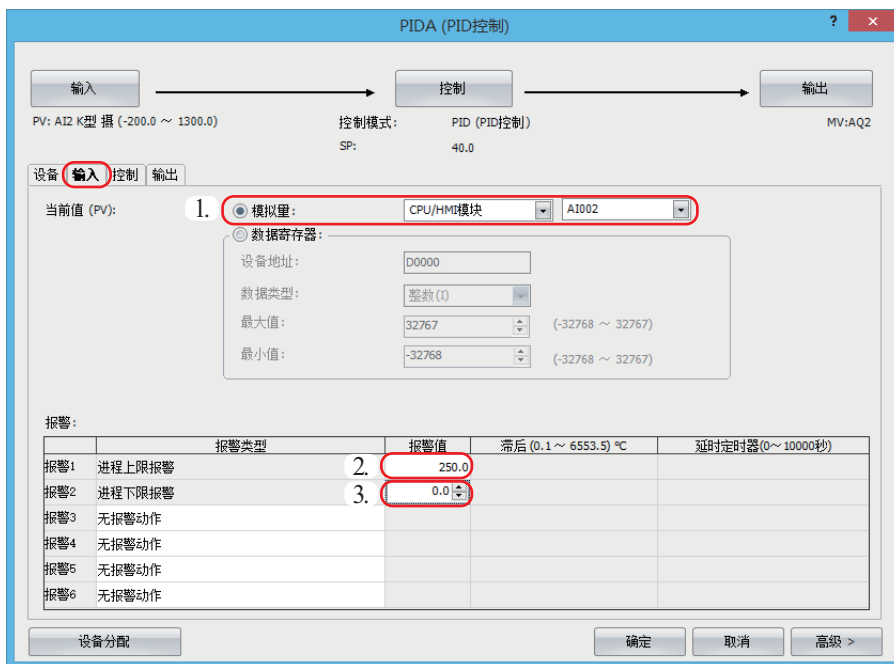
1. 在“设备”选项卡中，通过 PIDA 指令设置所使用的设备。

- 将 D0000 设为 S1 (控制寄存器) (1)。
- 将 M8120 设为 S2 (初始化输入) (2)。
- 将 M0000 设为 S3 (控制继电器) (3)。
- D0007 可自动设为 S4 (设定值) (4)。



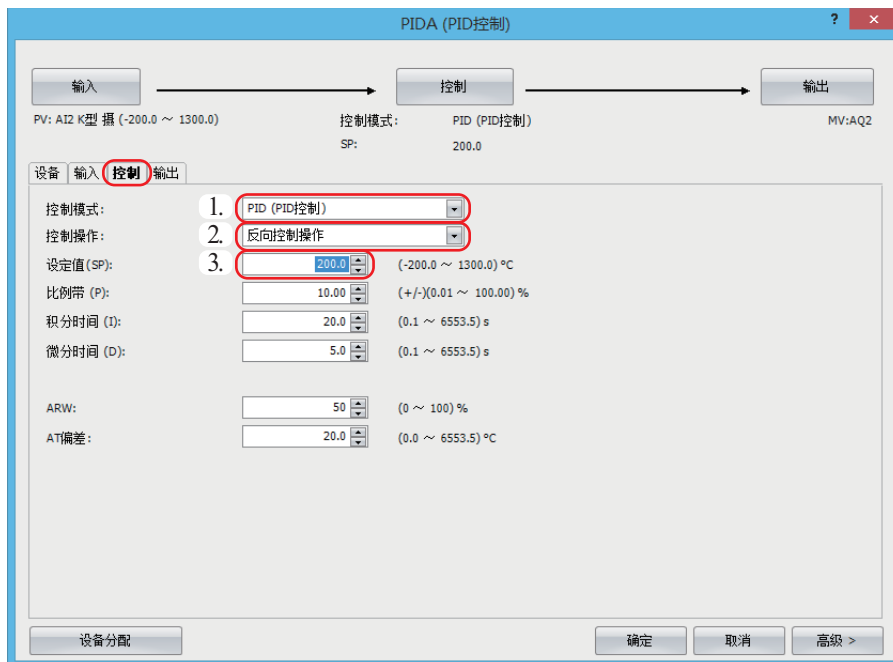
2. 单击“输入”选项卡，设置各项目。

- 在当前值 (PV) 中选择“模拟量”，并选择“CPU/HMI 模块”和“A1002”(1)。
- 通过报警 1，在进程上限报警的预置值中设置“250.0” (2)。
- 通过报警 2，在进程下限报警的预置值中设置“0.0” (3)。



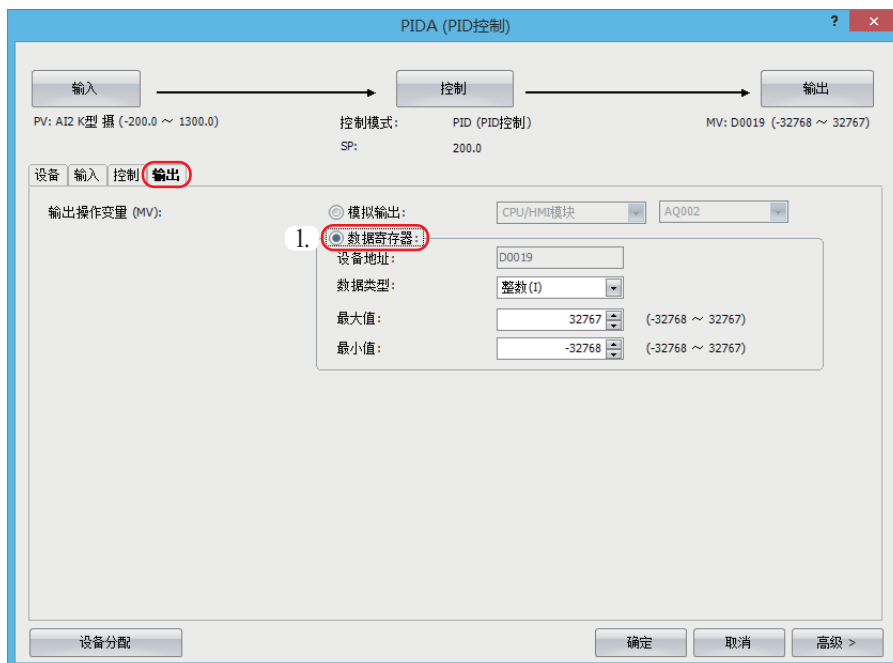
## 3. 单击“控制”选项卡，设置各项目。

- 在控制模式中选择“PID（PID控制）”（1）。
- 在控制操作中选择“反向控制操作”（2）。
- 在设定值（SP）中设置“200.0”（3）。



## 4. 单击“输出”选项卡，设置各项目。

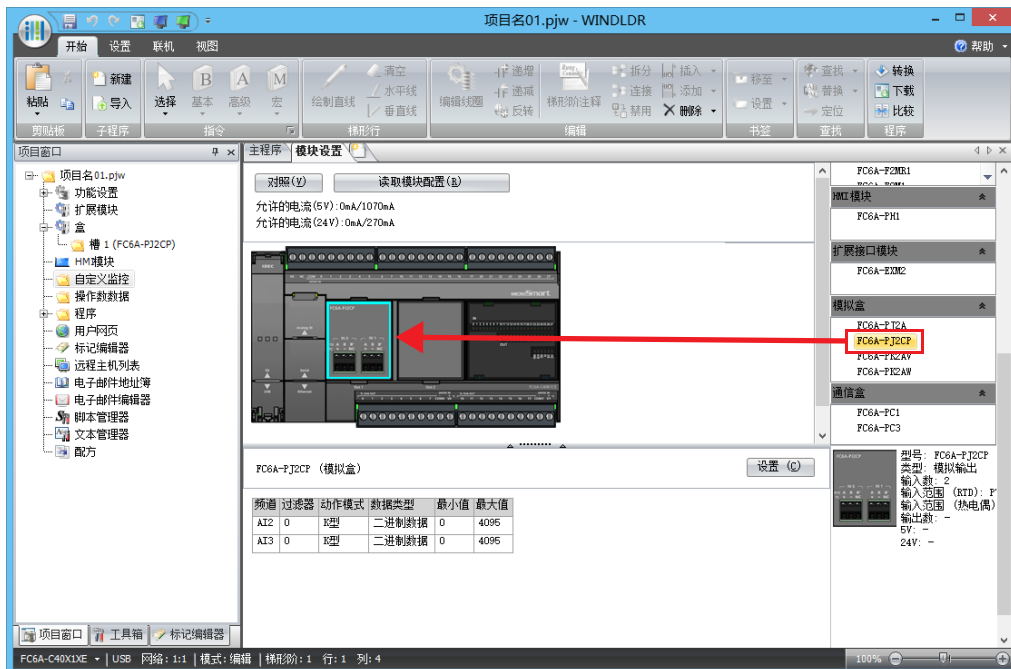
- 在输出操作变量（MV）中选择“数据寄存器”（1）。



## 5. 单击“确定”按钮，关闭“PID（PID控制）”对话框。

## 19: PID控制指令

- 通过模块构成编辑器进行模拟 I/O 盒的模拟量输入。  
在“设置”选项卡的“PLC”中单击“扩展模块”。
- 从扩展模块 / 盒一览中，将“FC6A-PJ2CP”拖放到模块构成区域中。



- 单击模块构成区域中插入的模拟 I/O 盒“FC6A-PJ2CP”，然后单击“设置”按钮。将显示“模拟量参数配置（盒槽 1）”对话框。
- 设置各项目。
  - 频道 IN0：在 AI2 的“数据类型”中选择“摄氏（℃）”（1）。

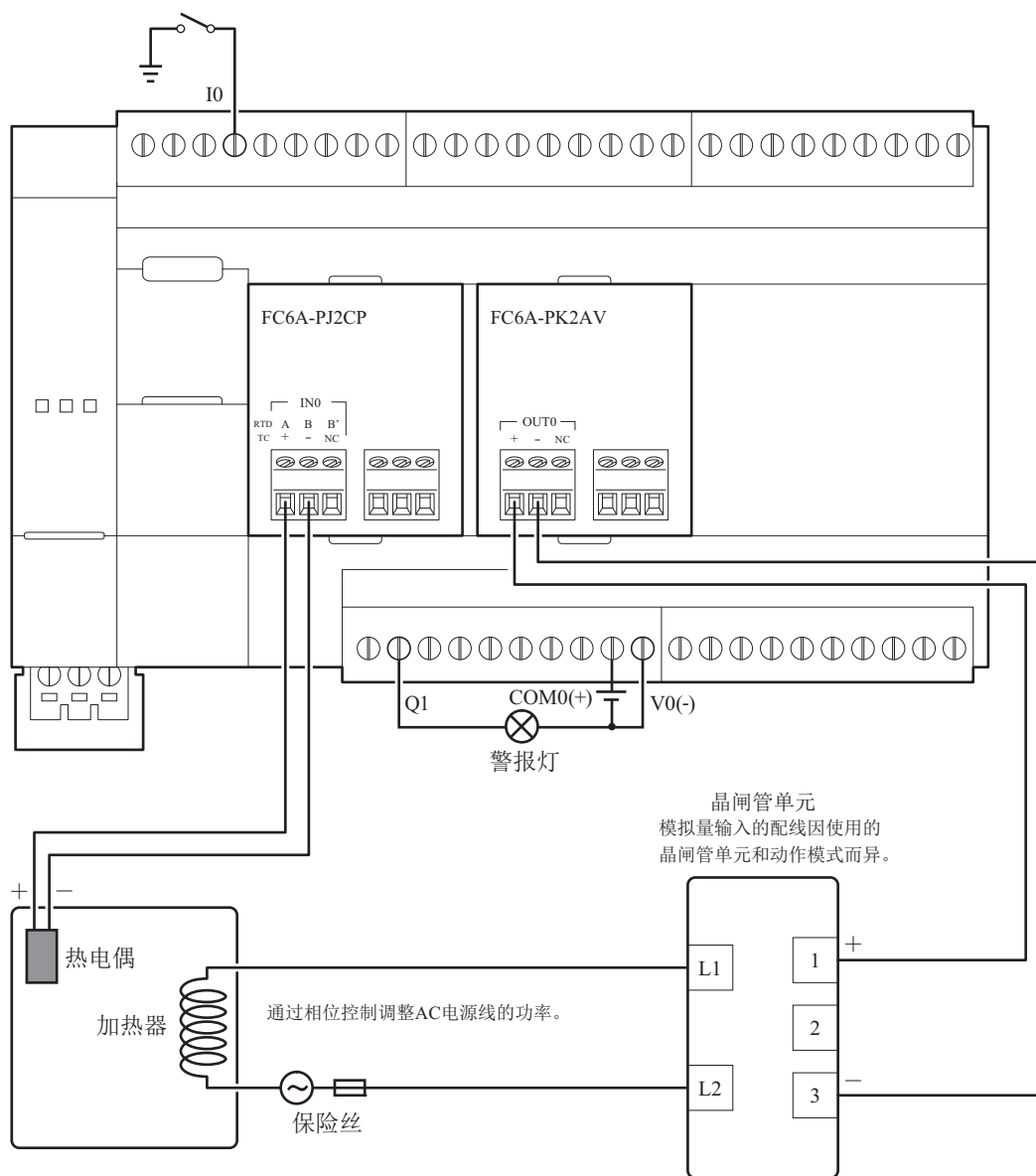


- 单击“确定”按钮，关闭“模拟量参数配置（盒槽 1）”对话框。



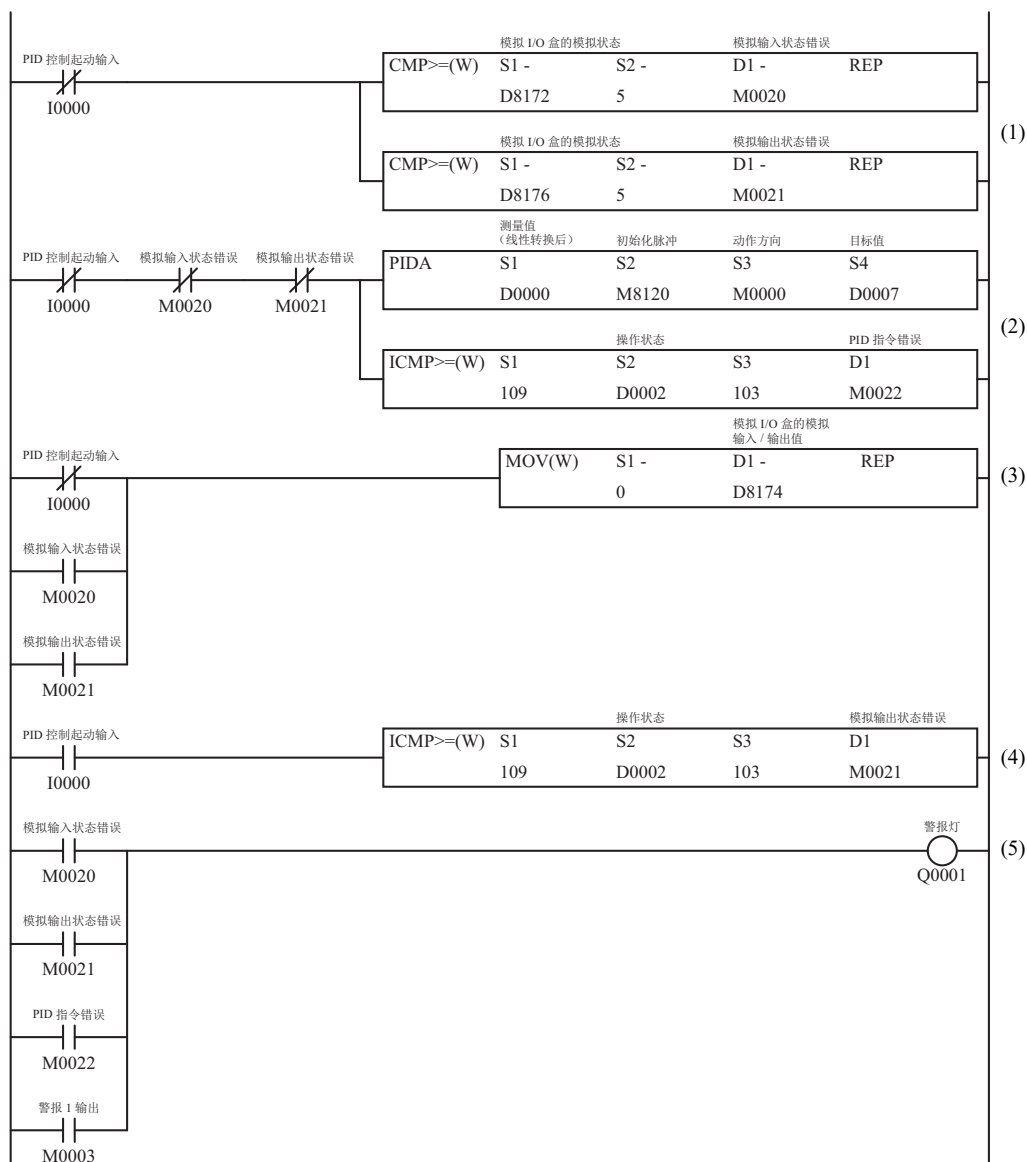
通过模拟输出进行PID控制

系统构成



## 19: PID控制指令

### 梯形图程序



设备地址	备注
D0000	测量值（线性转换后）
D0002	操作状态
D0007	目标值
D8178	模拟 I/O 盒的模拟状态
D8186	模拟 I/O 盒的模拟输入 / 输出值
D8188	模拟 I/O 盒的模拟状态
I0000	PID 控制起动作输入
Q0001	报警灯

设备地址	备注
M0000	动作方向
M0003	报警 1 输出
M0020	模拟输入状态错误
M0021	模拟输出状态错误
M0022	PID 指令错误
M8120	初始化脉冲

- (1) 在 I0000 为 ON 的状态下，模拟 I/O 盒的模拟输入状态错误为 5、6、8 ~ 11 时，打开 M0020。  
此外，模拟 I/O 盒的模拟输出状态错误为 8 ~ 11 时，打开 M0021。
- (2) 在 I0000 为 ON 的状态下，关闭 M0020 和 M0021 时，将执行 PID 指令。  
此外，判断 PID 指令的动作状况，并在发生异常时，打开 M0022。
- (3) M0020 或 M0021 打开时、或 I0000 关闭（PID 指令停止）时，将 0 存储到 D8186 中，并从 AQ4 输出 0V。
- (4) 判断 PID 指令的动作状况，并在发生异常时，打开 M0021。
- (5) M0020、M0021、M0022、M0003（报警 1 输出（进程上限报警））中的任意一个为 ON 时，将打开 Q0001（报警灯）。

**注释:**

模拟 I/O 盒的模拟状态如下所示。

- 模拟输入状态错误
  - 0: 正常动作中
  - 1: 数据转换中
  - 2: 初始化中
  - 5 ~ 6: 配线异常
  - 8 ~ 11: 模拟 I/O 盒的相关错误
- 模拟输出状态错误
  - 0: 正常动作中
  - 2: 初始化中
  - 3: 参数设置错误
  - 5 ~ 6: 配线异常
  - 8 ~ 11: 模拟 I/O 盒的相关错误

有关详情, 请参见《FC6A 型 MICROSmart 用户手册》第 10 章中的“模拟 I/O 盒”。

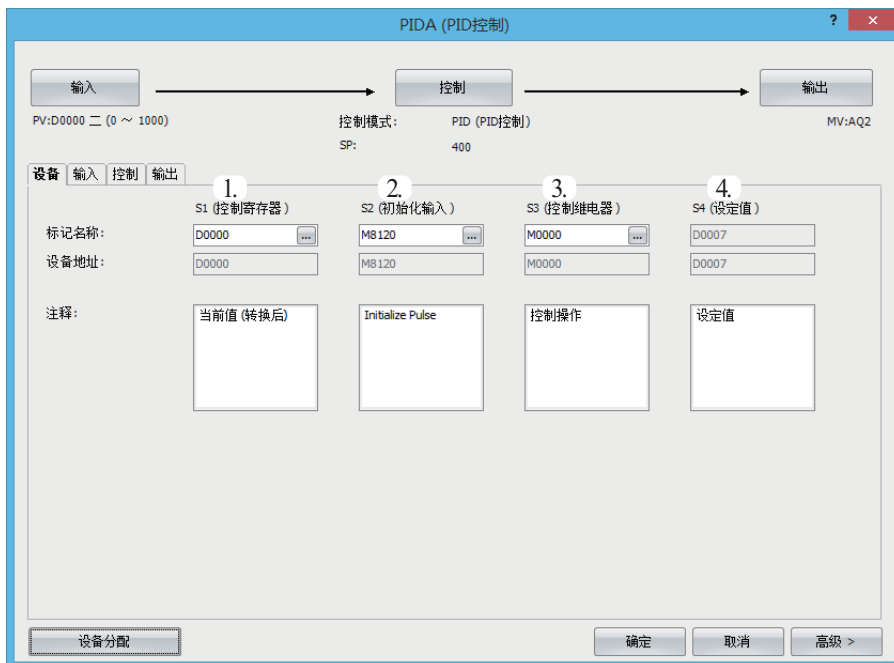
## 19: PID控制指令

### PID 指令的对话框设置步骤

步骤中未记载的设置项目已使用默认值。

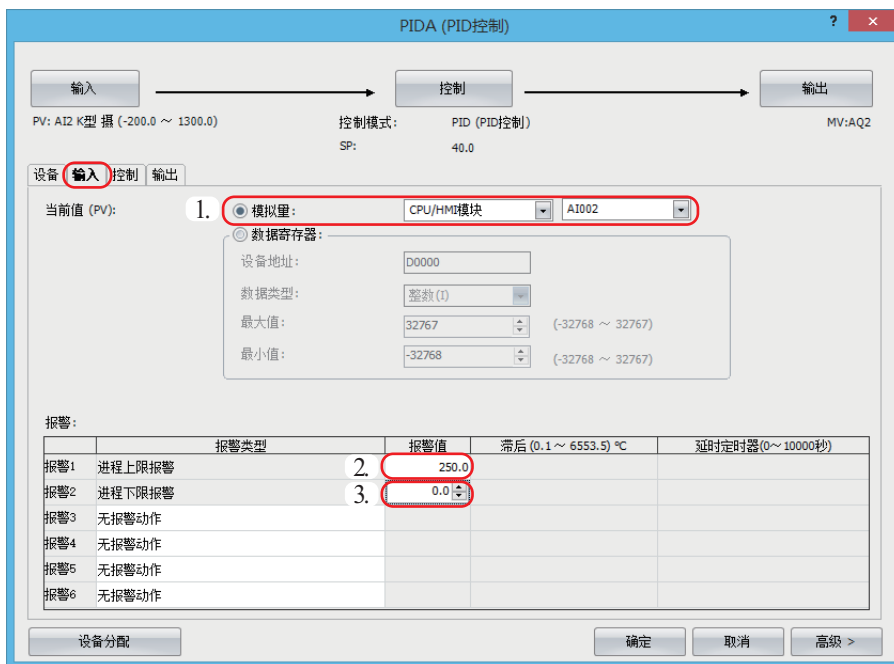
1. 在“设备”选项卡中，通过 PID 指令设置所使用的设备。

- 将 D0000 设为 S1（控制寄存器）（1）。
- 将 M8120 设为 S2（初始化输入）（2）。
- 将 M0000 设为 S3（控制继电器）（3）。
- D0007 可自动设为 S4（设定值）（4）。



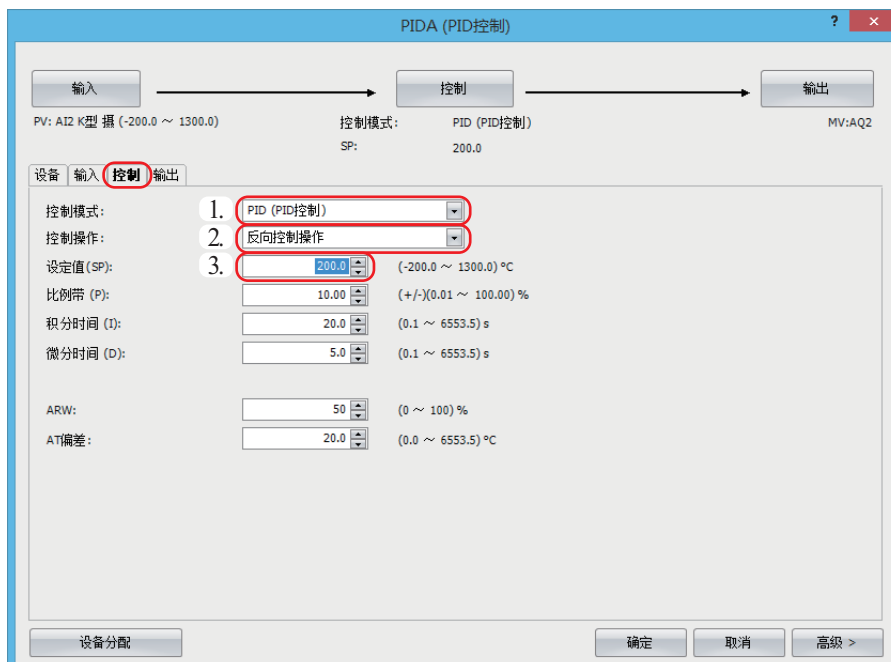
2. 单击“输入”选项卡，设置各项目。

- 在当前值（PV）中选择“模拟量”，并选择“AI002”（1）。
- 通过警报 1，在进程上限报警的预置值中设置“250.0”（2）。
- 通过警报 2，在进程下限报警的预置值中设置“0.0”（3）。



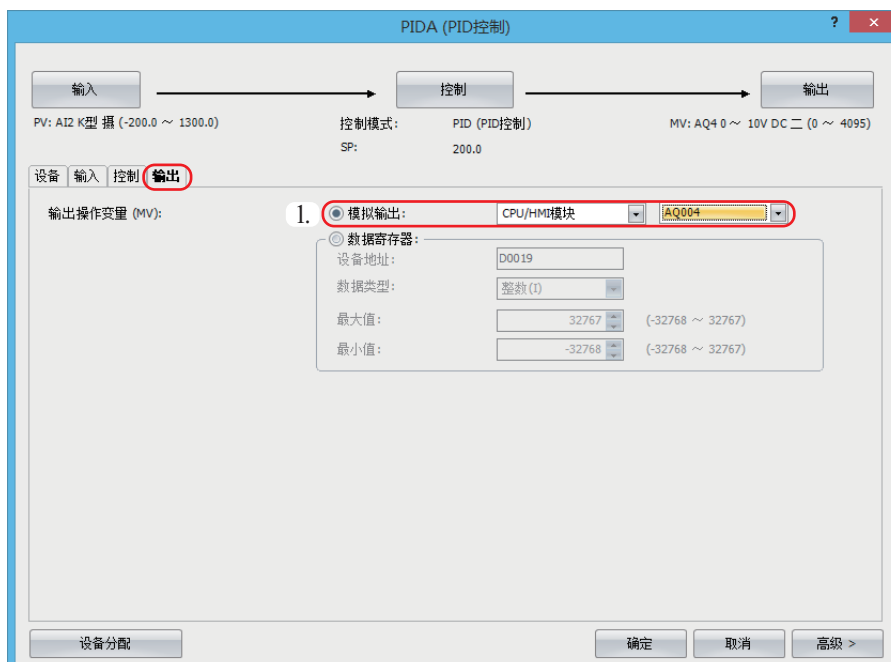
## 3. 单击“控制”选项卡，设置各项目。

- 在控制模式中选择“PID（PID控制）”（1）。
- 在控制操作中选择“反向控制操作”（2）。
- 在设定值（SP）中设置“200.0”（3）。



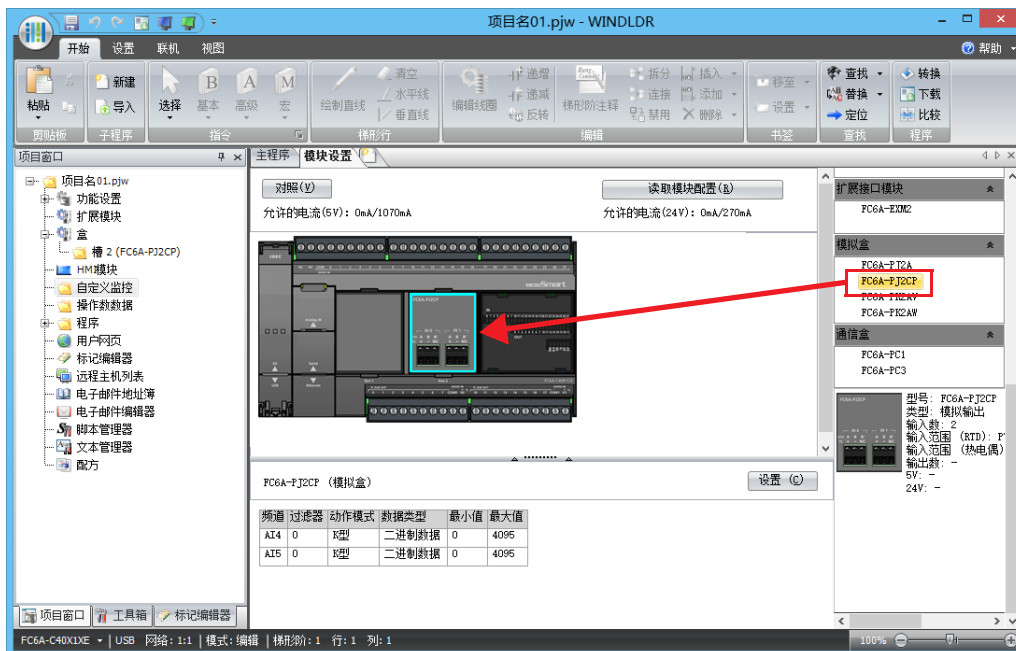
## 4. 单击“输出”选项卡，设置各项目。

- 在输出操作变量（MV）中选择“模拟输出”，并选择“AQ004”（1）。



## 19: PID控制指令

5. 通过模块构成编辑器进行模拟 I/O 盒的模拟量输入。  
在“设置”选项卡的“PLC”中单击“扩展模块”。
6. 从扩展模块/盒一览中，将“FC6A-PJ2CP”拖放到模块构成区域中。

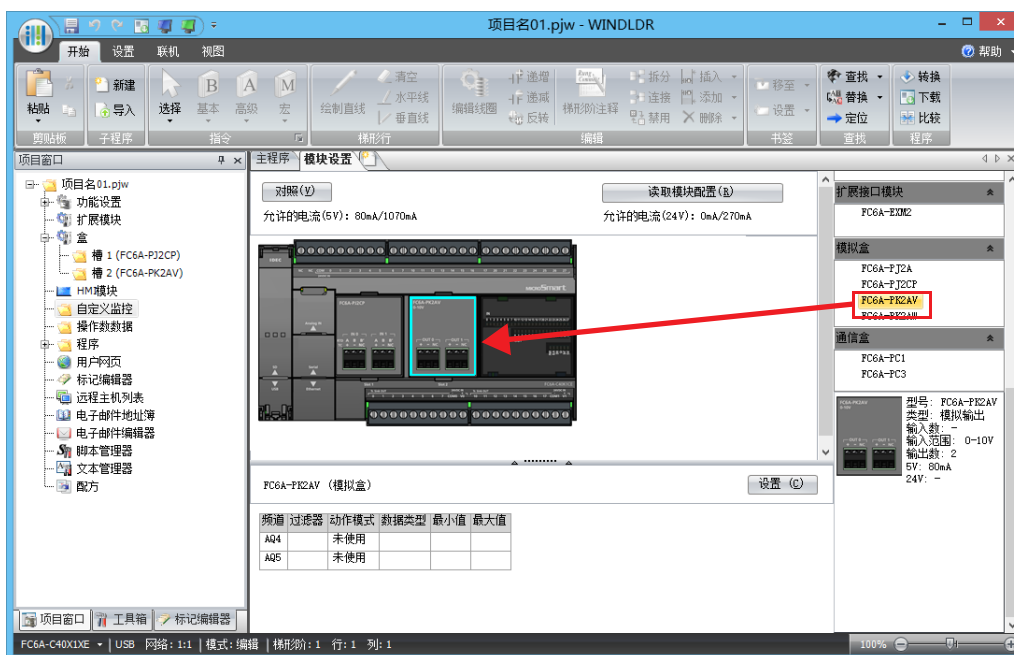


7. 单击模块构成区域中插入的模拟 I/O 盒“FC6A-PJ2CP”，然后单击“设置”按钮。  
将显示“模拟量参数配置（盒槽 1）”对话框。
8. 设置各项目。
  - 频道 IN0: 在 AI2 的“数据类型”中选择“摄氏（℃）”（1）。



9. 单击“确定”按钮，关闭“模拟量参数配置（盒槽 1）”对话框。

10. 从扩展模块 / 盒一览中，将“FC6A-PK2AV”拖放到模块构成区域中。



11. 单击模块构成区域中插入的模拟 I/O 盒“FC6A-PK2AV”，然后单击“设置”按钮。将显示“模拟量参数配置（盒槽 2）”对话框。

12. 设置各项目。

- 频道 OUT0: 在 AQ4 的“动作模式”中选择“0 ~ 10V DC”。



13. 单击“确定”按钮，关闭“模拟量参数配置（盒槽 2）”对话框。





# 20: 双 / 示教定时器指令

## 简介

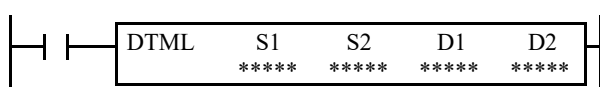
本章介绍以指定的时间间隔对输出进行周期性打开 / 关闭的双 / 示教定时器指令。提供了四个双定时器，可以从 1ms-65,535s 范围内选择开 / 关时间段。

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指定数据寄存器中，可用作定时器指令的预置值。

## DTML (1 秒双定时器)

DTML 指令是一个开 / 关时间设置定时器，定时器基数为 1 秒。

FT2J FC6A



当输入打开时，目标设备 D1 将在设备 S1 和 S2 指定的时间段分别重复打开和关闭。

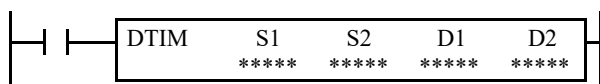
如果关闭输出，则会关闭 D1。

时间范围为 0-65,535s。

## DTIM (100 毫秒双定时器)

DTIM 指令是一个开 / 关时间设置定时器，定时器基数为 100 毫秒。

FT2J FC6A



当输入打开时，目标设备 D1 将在设备 S1 和 S2 指定的时间段分别重复打开和关闭。

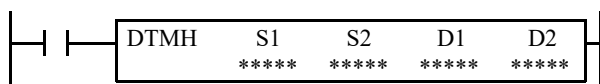
如果关闭输出，则会关闭 D1。

时间范围为 0-6,553.5s。

## DTMH (10 毫秒双定时器)

DTMH 指令是一个开 / 关时间设置定时器，定时器基数为 10 毫秒。

FT2J FC6A



当输入打开时，目标设备 D1 将在设备 S1 和 S2 指定的时间段分别重复打开和关闭。

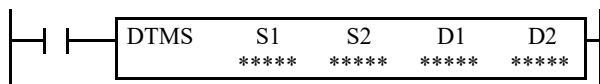
如果关闭输出，则会关闭 D1。

时间范围为 0-655.35s。

## DTMS (1 毫秒双定时器)

DTMS 指令是一个开 / 关时间设置定时器，定时器基数为 1 毫秒。

FT2J FC6A



当输入打开时，目标设备 D1 将在设备 S1 和 S2 指定的时间段分别重复打开和关闭。

如果关闭输出，则会关闭 D1。

时间范围为 0-65,535s。

### 注释:

- 有关双定时器指令的定时器精度，请参见第 4-13 页上的“定时器计数误差”。
- 双定时器指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

## 20: 双/示教定时器指令

### 有效设备

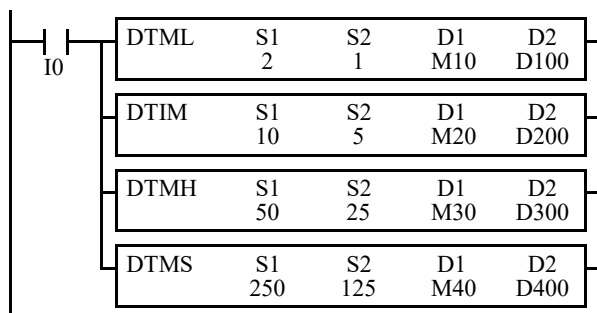
设备	功能	I	Q	M	R	T	C	D	P	常量
S1 (源 1)	打开时间	—	—	—	—	—	—	X	—	0-65,535
S2 (源 2)	关闭时间	—	—	—	—	—	—	X	—	0-65,535
D1 (目标 1)	双定时器输出	—	X	▲	X	—	—	—	—	—
D2 (目标 2)	系统工作区	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

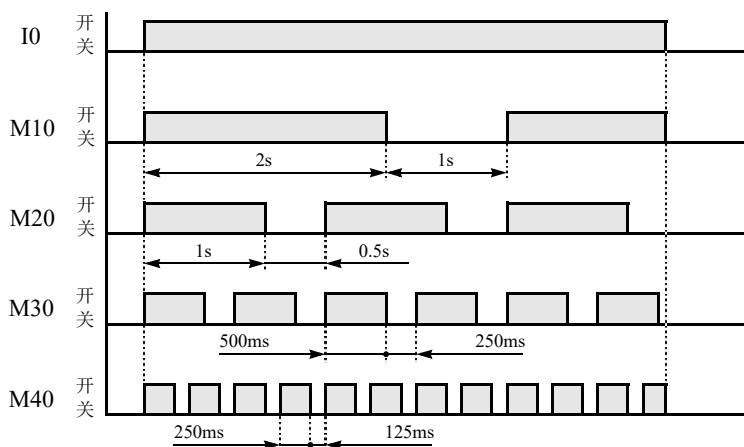
▲ 不能将特殊内部继电器指定为 D1。

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。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 WinLDR 上的监控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。

### 示例：DTML、DTIM、DTMH、DTMS



当输入 I0 打开时，四个双定时器指令会根据 I0 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。

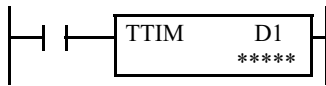


指令	递增	S1	打开时间	S2	关闭时间
DTML	1s	2	$1s \times 2 = 2s$	1	$1s \times 1 = 1s$
DTIM	100ms	10	$100ms \times 10 = 1s$	5	$100ms \times 5 = 0.5s$
DTMH	10ms	50	$10ms \times 50 = 500ms$	25	$10ms \times 25 = 250ms$
DTMS	1ms	250	$1ms \times 250 = 250ms$	125	$1ms \times 125 = 125ms$

## TTIM（示教定时器）

测量输入的打开时间。

FT2J FC6A



当输入打开时，将以 100ms 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。

如果关闭输出，则会关闭 D1。

测量时间范围为 0-6,553.5s。

**注释：**TTIM 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量
D1（目标 1）	测量值	—	—	—	—	—	—	X	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

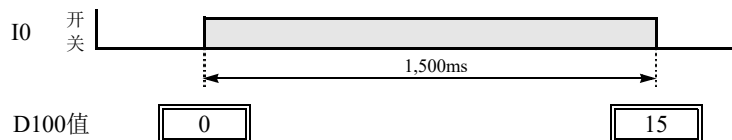
目标设备 D1（测量值）使用以分配给 D1 的设备起始的 3 个数据寄存器。以目标设备 D1+1 起始的两个后续数据寄存器用于系统工作区。请勿使用这两个数据寄存器作为其他高级指令的目标，以及使用 WindLDR 上的监控功能更改这些数据寄存器的值。如果这些数据寄存器中的数据被更改，则示教定时器不能正确操作。

### 示例：TTIM

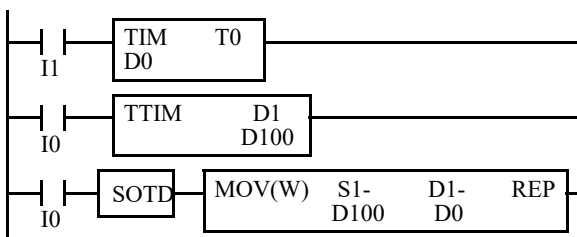


当输入 I0 打开时，TTIM 将数据寄存器 D100 复位为零，并开始将输入 I0 的打开时间存储至数据寄存器 D100 中（测量单位为 100ms）。

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 100ms 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。



# 21: 三角函数指令

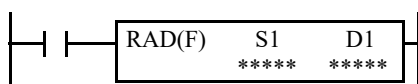
## 简介

本章将对计算从弧度转换为正弦、余弦和正切的三角函数指令进行介绍。

## RAD（角度换算为弧度）

将角度值转换为弧度值。

FT2J FC6A



$$S1 \cdot S1 + 1^\circ \times \pi / 180 \rightarrow D1 \cdot D1 + 1 \text{rad}$$

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。

### 注释：

- 当转换结果不在  $-3.402823 \times 10^{38}$  与  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  与  $3.402823 \times 10^{38}$  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）。有关进位或借位，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
- 当 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

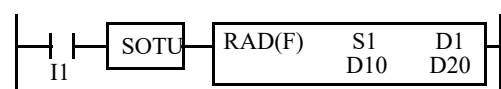
关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 有效数据类型

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

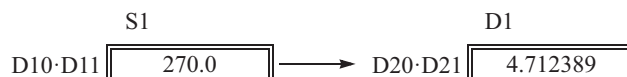
由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。

### 示例 :RAD



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。

$$270^\circ \times \pi / 180 \rightarrow 4.712389 \text{ rad}$$

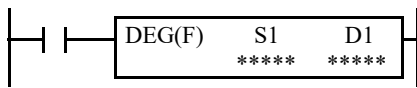


## 21: 三角函数指令

### DEG（弧度换算为角度）

将弧度值转换为角度值。

FT2J FC6A



$$S1 \cdot S1 + 1 \text{rad} \times 180/\pi \rightarrow D1 \cdot D1 + 1^\circ$$

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。

#### 注释：

- 当转换结果不在  $-3.402823 \times 10^{38}$  与  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  与  $3.402823 \times 10^{38}$  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 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

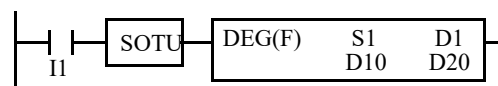
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为角度的弧度值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 有效数据类型

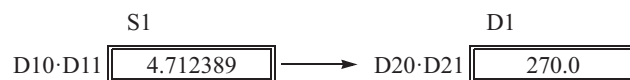
W（字）	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I（整数）	—	
D（双字）	—	
L（长整数）	—	
F（浮点）	X	

#### 示例 :DEG



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。

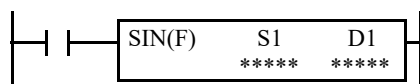
$$4.712389 \text{ rad} \times 180/\pi \rightarrow 270.0^\circ$$



## SIN（正弦）

计算弧度值的正弦值。

FT2J FC6A



$\sin S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$

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。

**注释：**当 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

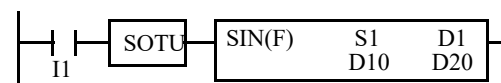
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为正弦值的弧度值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 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 :SIN



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。

$3.926991\text{rad} = 5\pi/4\text{rad}$

$\sin 5\pi/4 \rightarrow -0.7071069$

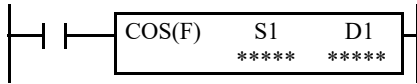


## 21: 三角函数指令

### COS（余弦）

计算弧度值的余弦值。

FT2J FC6A



$$\cos S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$$

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。

**注释：**当 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

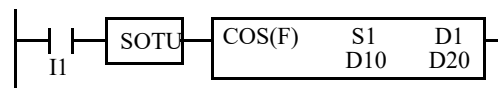
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为余弦值的弧度值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 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	

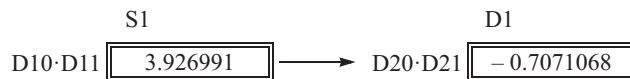
#### 示例 :COS



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。

$$3.926991\text{rad} = 5\pi/4\text{rad}$$

$$\cos 5\pi/4 \rightarrow -0.7071068$$

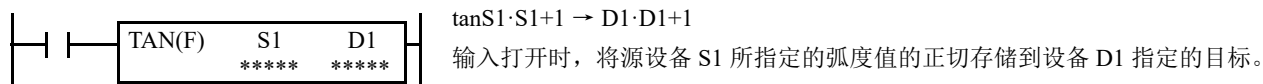




## TAN（正切）

计算弧度值的正切值。

FT2J FC6A



## 注释:

- 当转换结果不在  $-3.402823 \times 10^{38}$  与  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  与  $3.402823 \times 10^{38}$  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）。有关进位或借位，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
- 当 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

## 有效设备

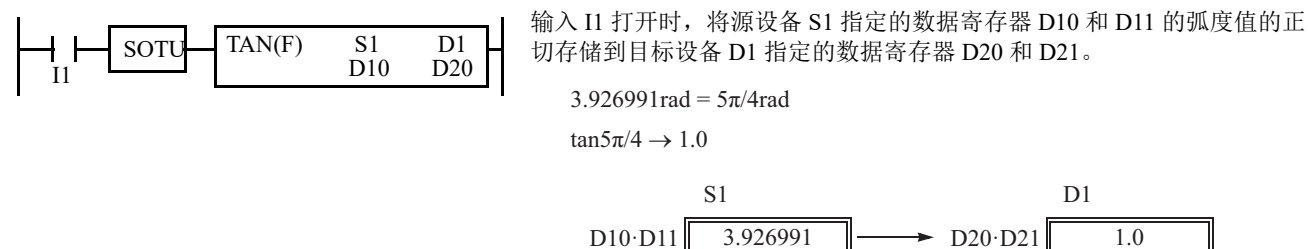
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为正切值的弧度值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 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

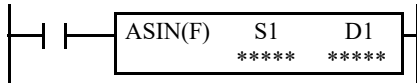


## 21: 三角函数指令

### ASIN（反正弦）

计算反正弦的主值。

FT2J FC6A



$\text{asin}S1 \cdot S1+1 \rightarrow D1 \cdot D1+1 \text{rad}$

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 值不在此范围内，将在 D1·D1+1 中存储 0。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当源设备 S1 指定的数据不 -1.0 和 1.0 之间时，将存储错误代码 29 在用户程序执行错误代码 (D8006) 中。
- 当 S1 指定的数据不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

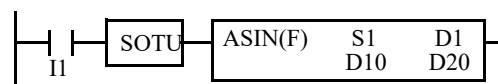
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为弧度的反正弦值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 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	

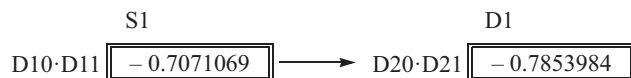
#### 示例 :ASIN



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。

$$\text{asin} - 0.7071069 \rightarrow - 0.7853984 \text{rad}$$

$$- 0.7853984 \text{rad} = - \pi/4 \text{rad}$$



## ACOS（反余弦）

计算反余弦的主值。

FT2J FC6A



$$\text{acos}S1 \cdot S1+1 \rightarrow D1 \cdot D1+1 \text{rad}$$

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 值不在此范围内，将在 D1·D1+1 中存储 0。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当源设备 S1 指定的数据不 -1.0 和 1.0 之间时，将存储错误代码 29 在用户程序执行错误代码 (D8006) 中。
- 当 S1 指定的数据不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

## 有效设备

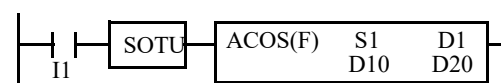
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为弧度的反余弦值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 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



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。

$$\text{acos} - 0.7071068 \rightarrow 2.356194 \text{rad}$$

$$2.356194 \text{rad} = 3\pi/4 \text{rad}$$

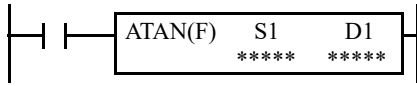


## 21: 三角函数指令

### ATAN（反正切）

计算反正切的主值。

FT2J FC6A



$\text{atan}S1 \cdot S1+1 \rightarrow D1 \cdot D1+1\text{rad}$

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。

**注释：**当 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

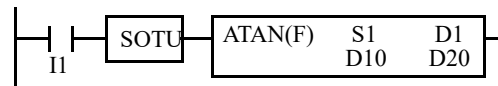
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为弧度的反正切值	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 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	

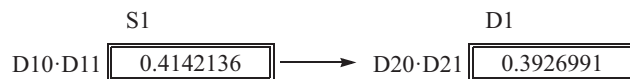
#### 示例 :ATAN



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。

$\text{atan } 0.4142136 \rightarrow 0.3926991\text{rad}$

$0.3926991\text{rad} = \pi/8\text{rad}$



# 22: 对数 / 幂指令

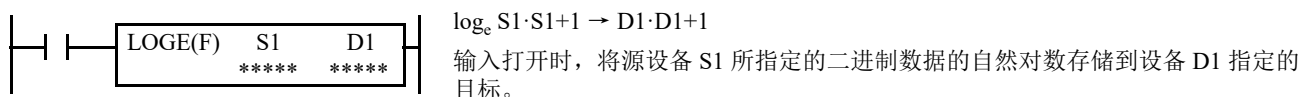
## 简介

本章将对计算指定数据的幂和对数的幂函数和对数函数指令进行介绍。

## LOGE（自然对数）

计算二进制数据的自然对数。

FT2J FC6A



### 注释:

- 当运算结果不在  $-3.402823 \times 10^{38} \times 10^{38}$  和  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  和  $3.402823 \times 10^{38}$  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。有关进位或借位，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当 S1 指定的数据为 0 或小于 0 时，将存储错误代码 29 在用户程序执行错误代码 (D8006) 中。
- 当 S1 指定的数据不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

### 有效设备

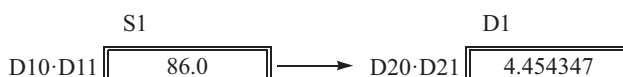
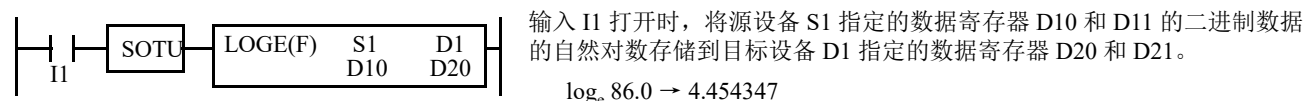
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为自然对数的二进制数据	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 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

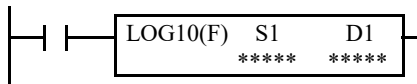


## 22: 对数/幂指令

### LOG10（常用对数）

计算二进制数据的常用对数。

FT2J FC6A



$\log_{10} S1 \cdot S1+1 \rightarrow D1 \cdot D1+1$

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。

#### 注释：

- 当运算结果不在  $-3.402823 \times 10^{38} \times 10^{38}$  和  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  和  $3.402823 \times 10^{38}$  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。有关进位或借位，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当 S1 指定的数据为 0 或小于 0 时，将存储错误代码 29 在用户程序执行错误代码 (D8006) 中。
- 当 S1 指定的数据不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

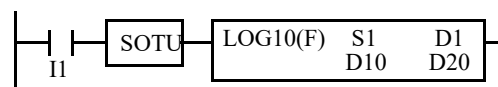
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	要转换为常用对数的二进制数据	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	要存储转换结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 有效数据类型

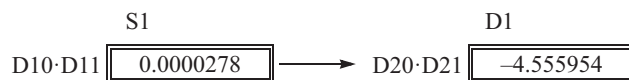
W（字）	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I（整数）	—	
D（双字）	—	
L（长整数）	—	
F（浮点）	X	

#### 示例：LOG10



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。

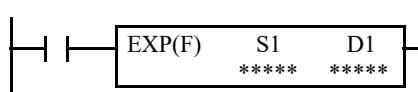
$\log_{10} 0.0000278 \rightarrow -4.555954$



## EXP（指数）

计算 e（自然对数的底数）的幂次。

FT2J FC6A



$$e^{S1 \cdot S1+1} \rightarrow D1 \cdot D1+1$$

输入打开时，将计算 e 的源设备 S1 指定的 S1·S1+1 次幂并存储到设备 D1 指定的目标。

$$e \text{（自然对数的底数）} = 2.7182818$$

### 注释：

- 当运算结果不在  $-3.402823 \times 10^{38} \times 10^{38}$  和  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  和  $3.402823 \times 10^{38}$  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。有关进位或借位，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
- 当 S1 指定的数据不符合正常浮点格式时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

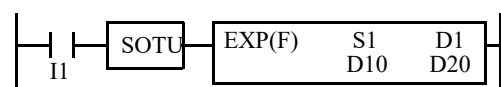
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	指数的二进制数据	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	用于存储结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

### 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



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。

$$e^2 = 2.7182818^2 \rightarrow 7.389056$$

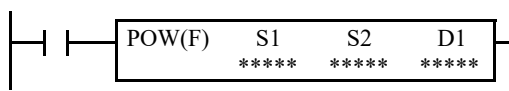


## 22: 对数/幂指令

### POW（幂）

计算二进制数据的幂。

FT2J FC6A



$$S1 \cdot S1+1^{S2 \cdot S2+1} \rightarrow D1 \cdot D1+1$$

输入打开时，将计算源设备 S1 指定的数据由源设备 S2 指定的 S2·S2+1 次幂，并将运算结果存储到设备 D1 指定的目标。

#### 注释:

- 当运算结果不在  $-3.402823 \times 10^{38}$  和  $-1.175495 \times 10^{-38}$  或  $1.175495 \times 10^{-38}$  至  $3.402823 \times 10^{38}$  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。有关进位或借位，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当 S1 指定的数据小于 0 且 S2 指定的数据不是整数时，将存储错误代码 29 在用户程序执行错误代码 (D8006) 中。
- 当 S1 指定的数据为 0 且 S2 指定的数据小于或等于 0 时，将存储错误代码 29 在用户程序执行错误代码 (D8006) 中。
- 当 S1 或 S2 指定的数据不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

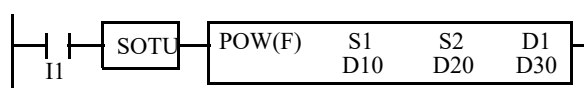
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	底数的二进制数据	—	—	—	—	—	—	X	—	X	—
S2（源 2）	指数的二进制数据	—	—	—	—	—	—	X	—	X	—
D1（目标 1）	用于存储结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

#### 有效数据类型

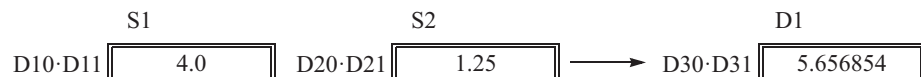
W（字）	—	由于使用了浮点数据类型，源和目标设备使用两个连续的数据寄存器。
I（整数）	—	
D（双字）	—	
L（长整数）	—	
F（浮点）	X	

#### 示例：POW



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。

$$4^{1.25} \rightarrow 5.656854$$





# 23: 文件数据处理指令

## 简介

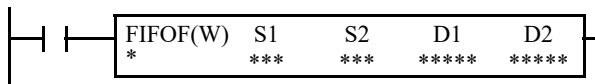
本章将对处理 FIFO（先进先出）数据结构的文件数据处理指令进行介绍。FIFO（先进先出格式）指令将初始化 FIFO 数据文件存储的数据。FIEX（执行先进）指令存储新数据到 FIFO 数据文件，FOEX（执行先出）指令从 FIFO 数据文件取得存储的数据。被 FIEX 指令存储到 FIFO 数据文件的第一个数据为被 FOEX 指令检索的第一个数据。

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）

记录 FIFO 数据文件的格式。

FT2J FC6A



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。

### 注释：

- 先进先出数据文件中最多可存储 ((S2)-1) 个数据。
- 控制器开始运行后，必须至少执行一次 FIFO 指令。
- 在执行 FIFO 指令之前，将 FI 指针和 FO 指针初始化为 0。有关 FI 和 FO 指针的详情，请参见第 23-2 页上的“目标设备 D1（FIFO 数据文件）”。
- FIFO 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
N（格式编号）	格式编号	—	—	—	—	—	—	—	—	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	—	—	▲	—	—	—	—	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

▲ 不能将特殊内部继电器指定为 D2。

### 有效数据类型

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

## 23: 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 S1-1 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)

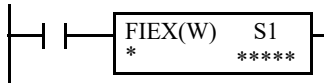
当执行 FIEX 或 FOEX 指令时，按照执行状态打开 / 关闭以下的内部继电器。

D2 的分配	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）

将记录数据存储到相应的 FIFO 数据文件中。

FT2J FC6A



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。

### 注释：

- 当 FI 指针等于 FO 指针时，FIFO 数据文件为空。如果在 FIFO 数据文件为空时执行 FOEX 指令，则不会执行任何操作，FIFO 指令的数据文件空输出 (D2+1) 将被打开。
- 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，将存储错误代码 31 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
N（格式编号）	格式编号	—	—	—	—	—	—	—	—	0~9	—
S1（源 1）	要存储到 FIFO 数据文件的起始数据存储器	—	—	—	—	—	—	X	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

### 有效数据类型

W（字）	X	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 S1 中的值一样多的数据寄存器。
I（整数）	—	
D（双字）	—	
L（长整数）	—	
F（浮点）	—	

## 23: 文件数据处理指令

### FOEX（执行先出）

从相应的 FIFO 数据文件中取得数据。

FT2J FC6A



当输入打开时，将从相应的 FIFO 数据文件中取得数据，并存储到以 D1 指定的设备开始的数寄存器中。

#### 注释：

- 当 FI 指针等于 FO 指针时，FIFO 数据文件为空。如果在 FIFO 数据文件为空时执行 FOEX 指令，则不会执行任何操作，FIFO 指令的数据文件空输出 (D2+1) 将被打开。
- 请确保相应的 FIFOF 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 FIFOF 指令就执行 FOEX 指令，将导致用户程序执行错误，将存储错误代码 31 在用户程序执行错误代码 (D8006) 中。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
N（格式编号）	格式编号	—	—	—	—	—	—	—	—	0~9	—
D1（目标 1）	要存储数据的起始数据寄存器编号	—	—	—	—	—	—	X	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

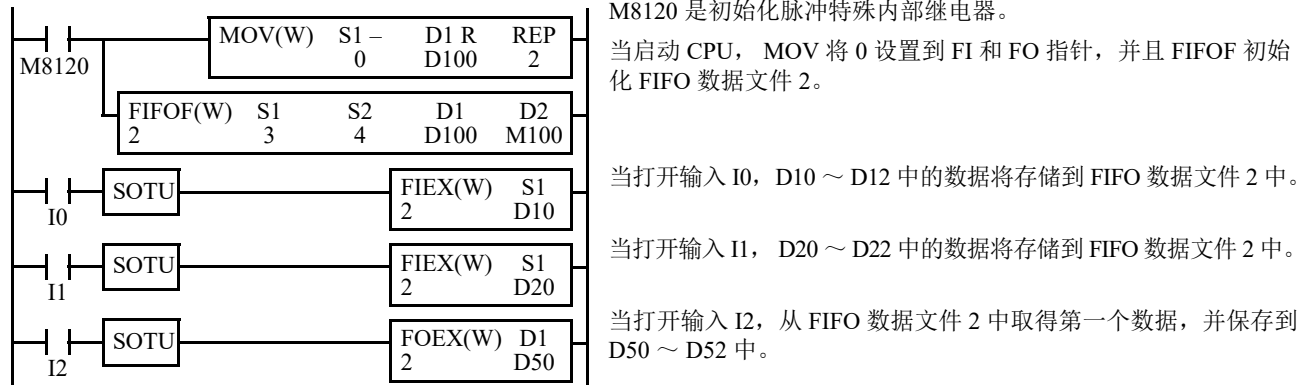
#### 有效数据类型

W（字）	X	当 D（数据寄存器）指定为目标时，将使用与存储在相应的 FIFOF 指令的设备 S1 中的值一样多的数据寄存器。
I（整数）	—	
D（双字）	—	
L（长整数）	—	
F（浮点）	—	

**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。

格式编号 :	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	—	—	—

## 23: 文件数据处理指令

### NDSRC (N 数据查找)

从指定数据寄存器区域搜索指定值。

FT2J FC6A



当输入打开，被设备 S1 指定的值将被查找。从设备 S2 指定的数据寄存器开始，查找数据寄存器。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。匹配项的数量将存储在下一个数据寄存器中。当查找结果不匹配时，65,535 被存储在设备 D1 并将 0 存储到设备 D1+1。

**注释：**在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

- 当 S3 为零或 S2 和 S3 的合计不在数据寄存器的有效范围内，将存储错误代码 1 在用户程序执行错误代码 (D8006) 中。
- 当选择 F (浮点) 数据类型，并且 S1 不符合标准浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。
- 当选择 F (浮点) 数据类型，并且 S2 到 S2+(S3) 中的数据不符合浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。跳过错误数据继续搜索，搜索结果存储在 D1 和 D1+1 中。
- NDSRC 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

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。

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。

#### 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) 或 2 点 (双字、长整数或浮点数据)。
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

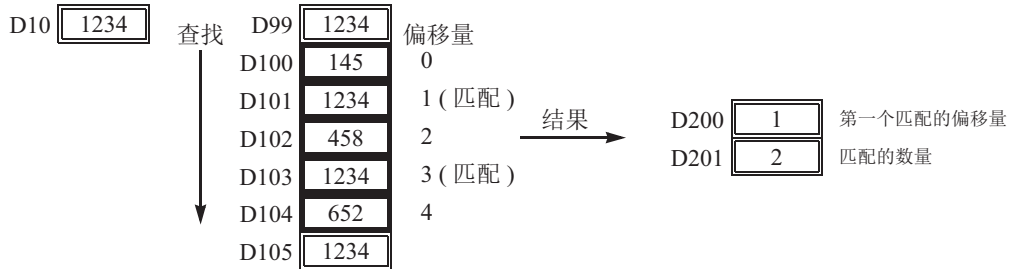
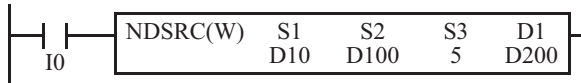
根据数据类型，源设备 S1 和 S2 使用不同的设备数量。源设备 S3 和目标设备 D1 无视数据类型总是使用 1 个字。

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

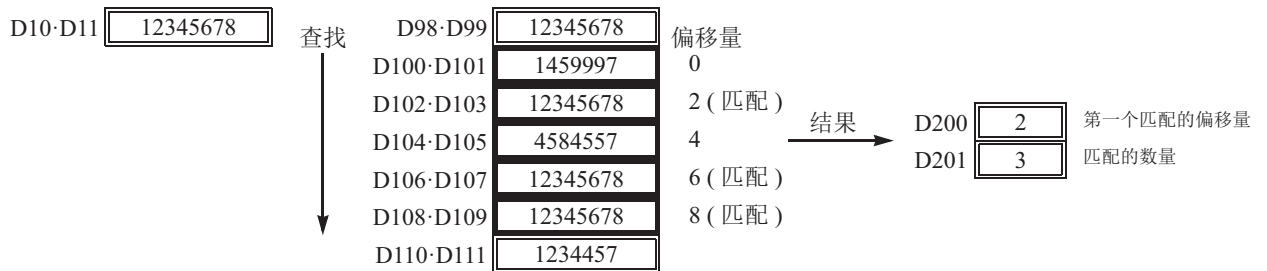
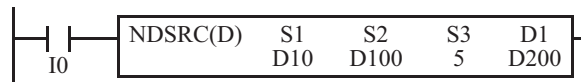
**示例：NDSRC**

以下示例描述了如何使用 NDSRC 指令查找三种不同类型的数据。

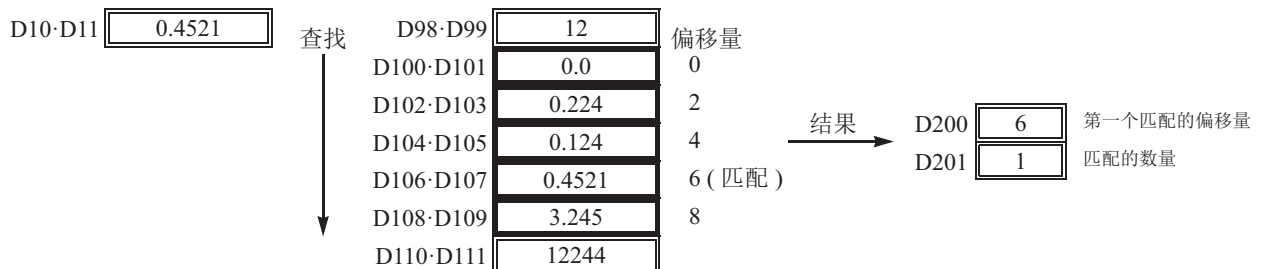
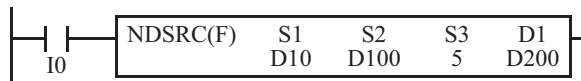
• 数据类型：字



• 数据类型：双字



• 数据类型：浮点







# 24: 时钟指令

## 简介

TADD（时间加法）和 TSUB（时间减法）指令在两个不同模式中加减时间数据。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。

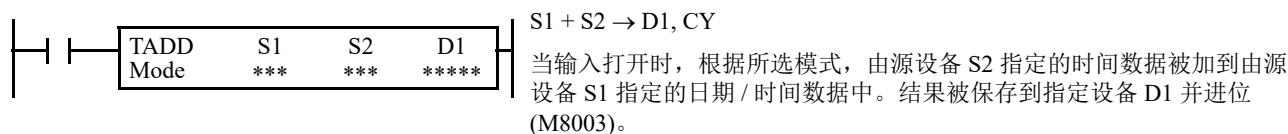
HTOS（HMS → 秒）和 STOH（秒 → HMS）指令执行时间数据在小时、分、秒与秒之间的转换。

HOURL（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）

将时间数据添加到日期/时间数据中。

FT2J FC6A

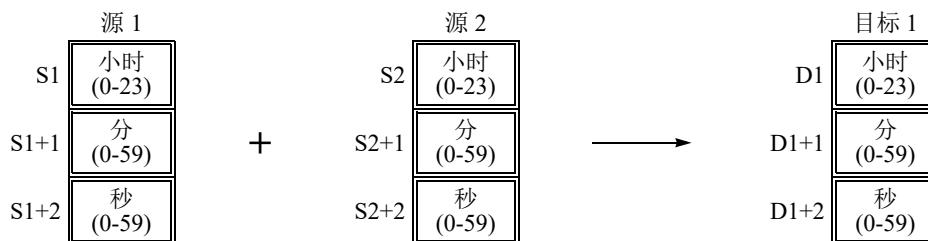


当选择模式 0，源设备 S1、S2 和目标设备 D1 占用由设备指定开始的 3 个连续数据寄存器。

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。源设备 S2 占用由设备指定开始的 3 个连续数据寄存器。

## 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 个连续的数据寄存器。小时数据可以为 0 ~ 23。分和秒数据可以为 0 ~ 59。



## 注释:

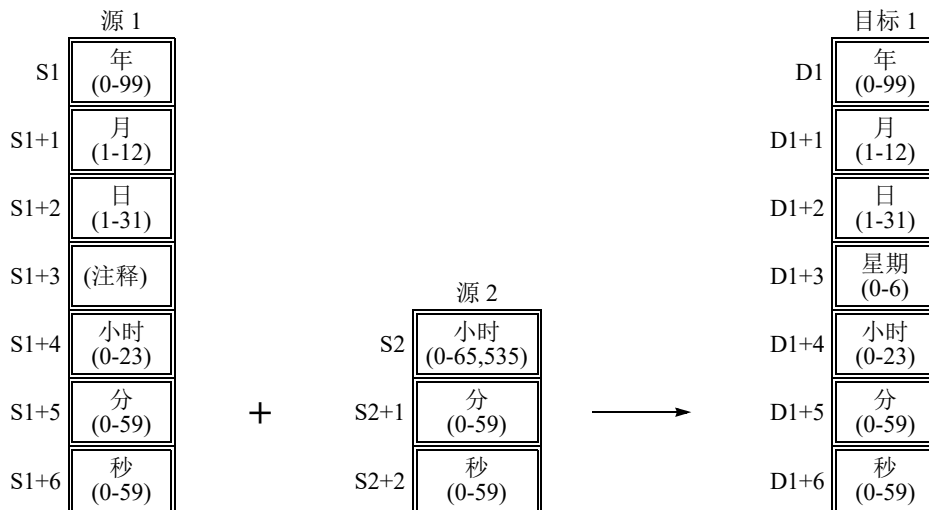
- 如果小时、分和秒数据超出有效范围，将导致用户程序执行错误，将存储错误代码 32 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当执行结果超出 23:59:59，其结果将被减去 24 小时后存储到由目标设备 D1 指定的数据寄存器，并且打开特殊内部继电器 M8003（进位）。

## 24: 时钟指令

### 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。日期 / 时间和时间数据的设置范围如下。

项目	设置范围
日期 / 时间数据	年: 0-99 (2000 至 2099 年)
	月: 1-12
	日: 1-31
	星期: -
	小时: 0-23
	分: 0-59
	秒: 0-59
时间数据	小时: 0-65535
	分: 0-59
	秒: 0-59



**注释:** S1+3 不用于计算，无需指定。D1+3 从计算结果的日期开始计算。

#### 注释:

- TADD 指令支持闰年。
- 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)。
- 在以下情况下会发生用户程序执行错误，将存储错误代码 32 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
  - 任何小时、分或秒数据超出有效范围。
  - 源 1 包含无效的日 / 时间数据。
  - 执行结果超出 99 年 12 月 31 日 23:59:59。

#### 有效设备

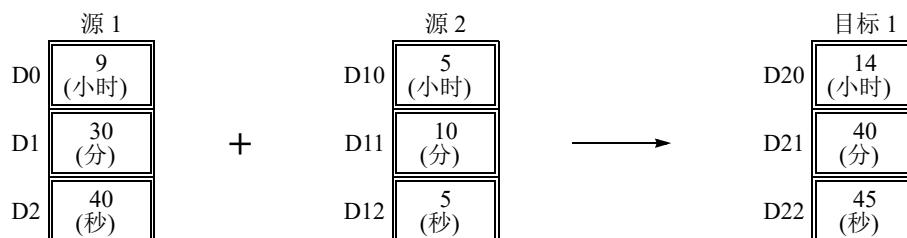
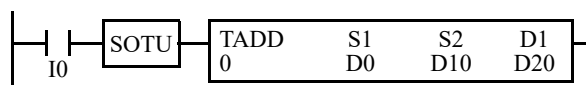
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
模式	S1 数据范围的选择	—	—	—	—	—	—	—	—	0, 1	—
S1 (源 1)	要加到的日期 / 时间数据	—	—	—	—	—	—	X	—	—	—
S2 (源 2)	要加的时间数据	—	—	—	—	—	—	X	—	—	—
D1 (目标 1)	要存储结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

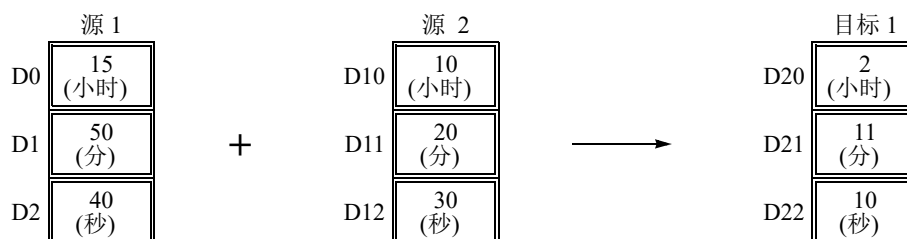
### 示例：TADD

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。

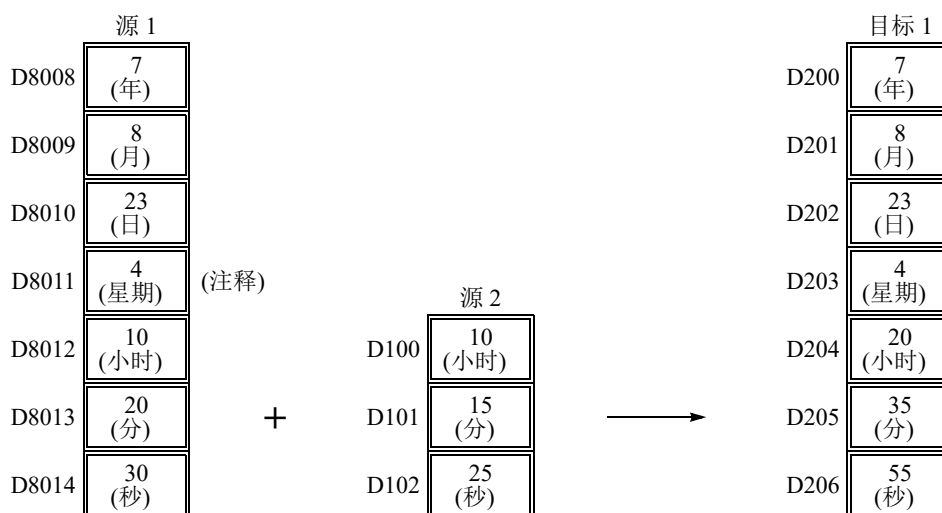
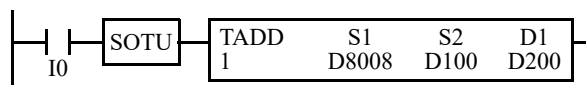
#### • 模式 0



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 )。



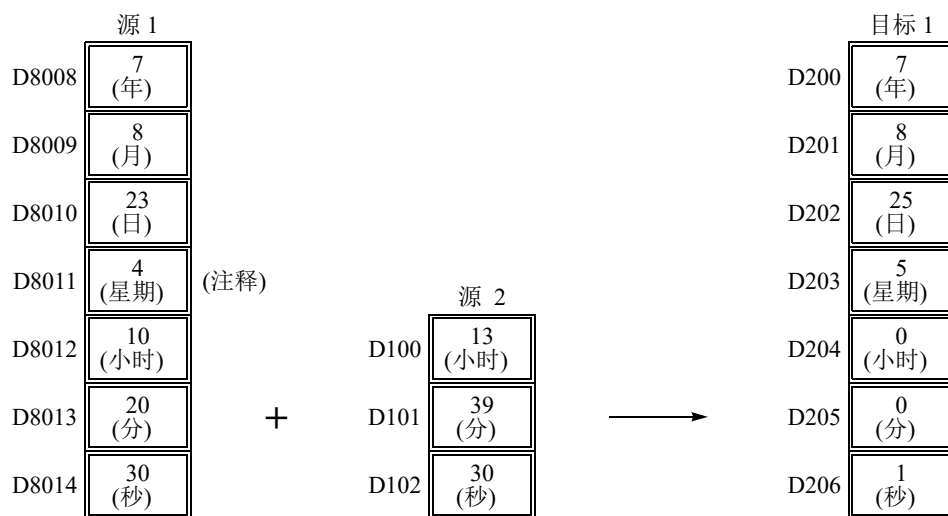
#### • 模式 1



**注释：** S1+3 (D8011) 不用于计算，无需指定。自动从年、月和日的合计计算星期，其结果将保存到设备 D203。

## 24: 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。

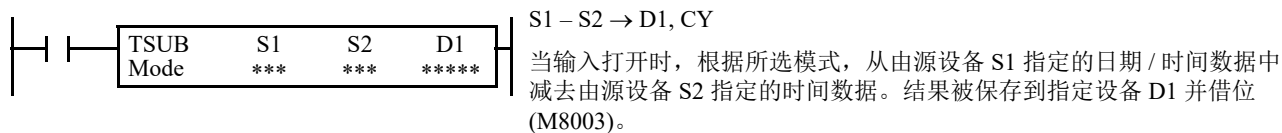


**注释:** S1+3 (D8011) 不用于计算，无需指定。自动从年、月和日的合计计算星期，其结果将保存到设备 D203。

## TSUB（时间减法）

将时间数据减去日期 / 时间数据。

FT2J FC6A

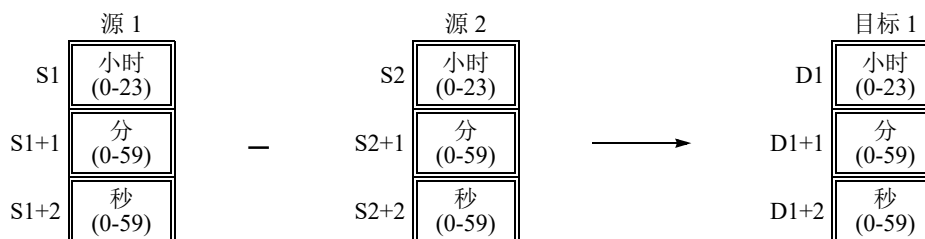


当选择模式 0，源设备 S1、S2 和目标设备 D1 占用由设备指定开始的 3 个连续的数据寄存器。

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。源设备 S2 占用由设备指定开始的 3 个连续数据寄存器。

### 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。



#### 注释:

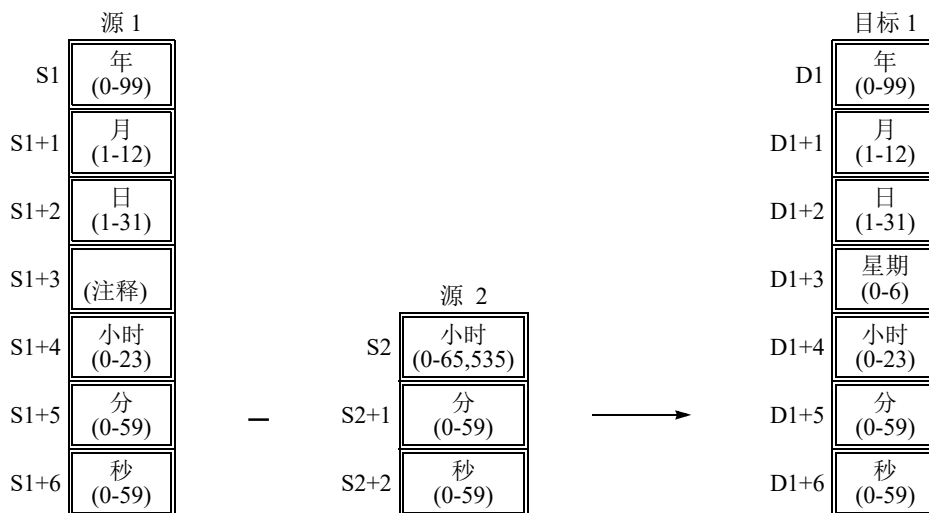
- 如果小时、分和秒数据超出有效范围，将导致用户程序执行错误，将存储错误代码 32 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当执行结果小于 00:00:00，其结果将被加上 24 小时后存储到由目标设备 D1 指定的数据寄存器，并且打开特殊内部继电器 M8003（借位）。

## 24: 时钟指令

### 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 个连续的数据存储器。日期/时间和时间数据的设置范围如下。

项目	设置范围
日期 / 时间数据	年: 0-99 (2000 至 2099 年)
	月: 1-12
	日: 1-31
	星期: -
	小时: 0-23
	分: 0-59
	秒: 0-59
时间数据	小时: 0-65535
	分: 0-59
	秒: 0-59



**注释:** S1+3 不用于计算，无需指定。D1+3 从计算结果的日期开始计算。

#### 注释:

- TSUB 指令支持闰年。
- 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)。
- 在以下情况下会发生用户程序执行错误，将存储错误代码 32 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
  - 任何小时、分或秒数据超出有效范围。
  - 源 1 包含无效的日 / 时间数据。
  - 执行结果超出 00 年 1 月 1 日 00:00:00。

#### 有效设备

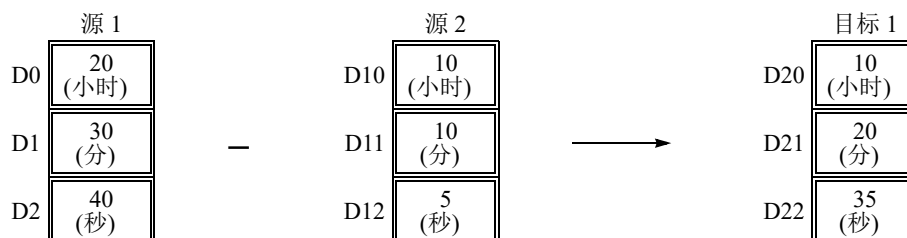
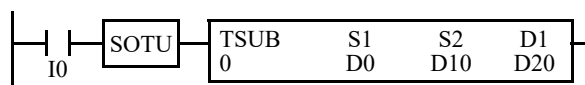
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
模式	S1 数据范围的选择	—	—	—	—	—	—	—	—	0, 1	—
S1 (源 1)	需要从中减去的日期 / 时间数据	—	—	—	—	—	—	X	—	—	—
S2 (源 2)	要减去的时间数据	—	—	—	—	—	—	X	—	—	—
D1 (目标 1)	要存储结果的目标	—	—	—	—	—	—	X	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

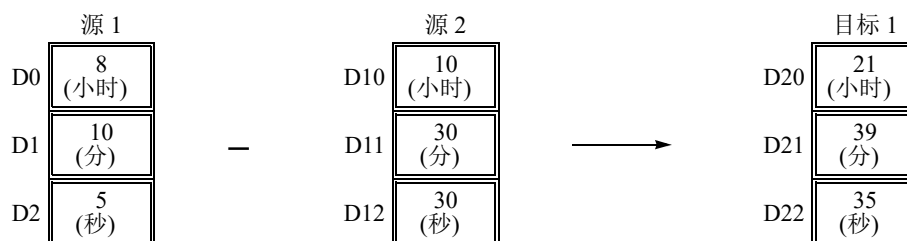
### 示例：TSUB

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。

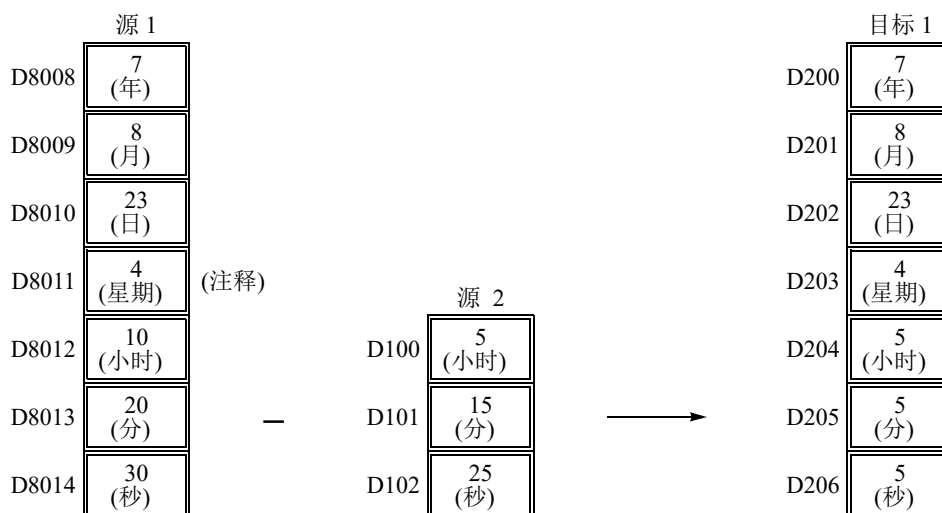
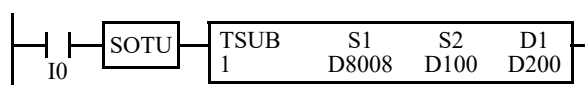
#### • 模式 0



当结果小于 00:00:00，结果将被加上 24，并打开特殊内部继电器 M8003（借位）。



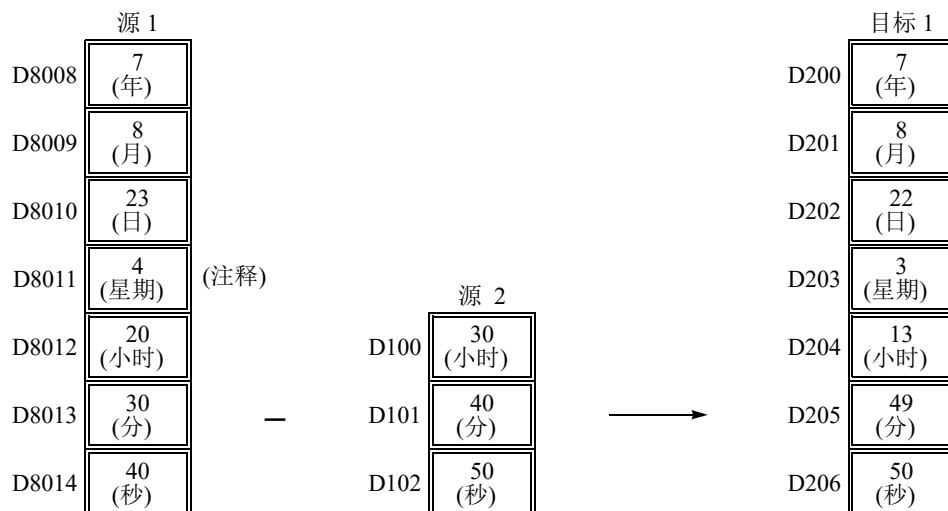
#### • 模式 1



**注释：** S1+3 (D8011) 不用于计算，无需指定。自动从年、月和日的合计计算星期，其结果将保存到设备 D203。

## 24: 时钟指令

当结果小于 00:00:00，结果将被加上 24，并递减日数据。



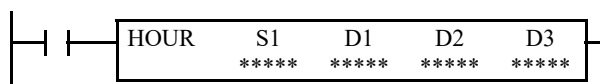
**注释:** 源 1 中的 D8011 没有被指定使用，并且无需指定。星期将自动从年、月和日的合计中被计算，并存储在目标 1 的 D203。



## HOUR（小时计量器）

测量输入的 ON 持续时间。

FT2J FC6A



S1 ↔ D1 → D2

输入打开时，将测量输入的 ON 持续时间。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。

将由 D3, D3+1 指定的数据寄存器作为系统工作区使用。

小时数据可以为 0 ~ 65,535。分和秒数据可以为 0 ~ 59。

### 注释:

- 当测量到的 ON 期间值超出 65,535 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。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 如果源设备 S1 的小时、分或秒数据超出有效范围，将存储错误代码 32 在用户程序执行错误代码 (D8006) 中。D2 关闭。但输入 ON 期间测量仍将继续。
- HOUR 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	预置值	—	—	—	—	—	—	X	—	0 ~ 65,535	—
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	—	—	—

当源设备 S1 被指定为常量时，小时中的预置值可以为 0 ~ 65,535，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。

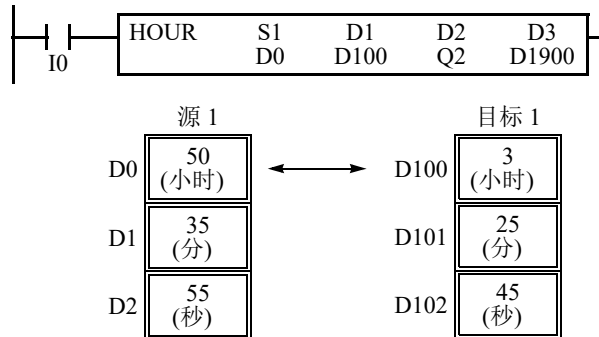
关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

## 24: 时钟指令

### 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



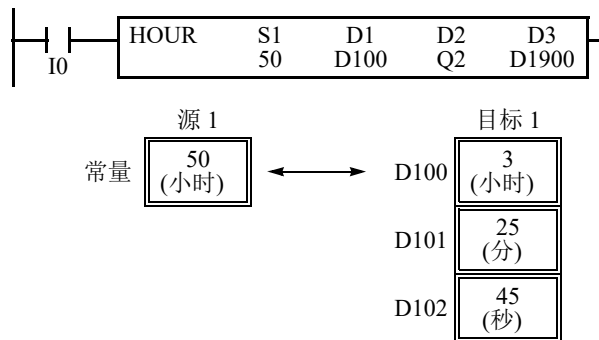
$D0 \cdot D1 \cdot D2 \leftrightarrow D100 \cdot D101 \cdot D102 \rightarrow Q2$

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 D0、D1、D2 中的预置值进行比较。

当测量到的值达到预置值，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



$50 \leftrightarrow D100 \cdot D101 \cdot D102 \rightarrow Q2$

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 指定的 50 小时进行比较。

当测量到的值达到 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。

## HTOS (HMS → 秒)

将小时、分、秒数据转换为秒。

FT2J FC6A



小时、分、秒 → 秒

当输入打开时，由源设备 S1 指定的小时、分和秒中的时间数据将转换为秒。此结果存储到目标设备 D1。

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。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。

小时数据可以为 0 ~ 65,535。分和秒数据可以为 0 ~ 59。

**注释：**如果小时、分和秒数据超出有效范围，将导致用户程序执行错误，将存储错误代码 32 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

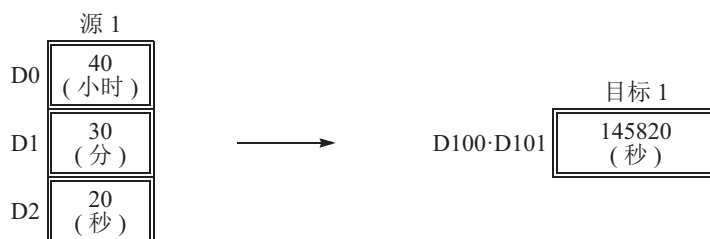
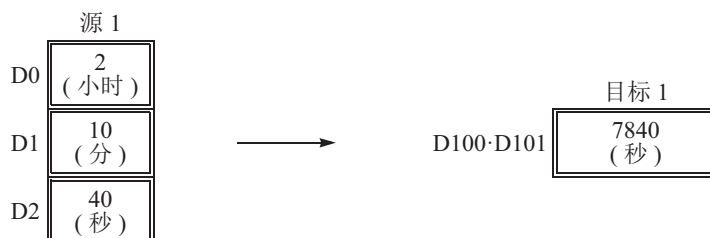
当数据寄存器分配给 S1 时，使用 1 个数据寄存器。

数据寄存器分配给 D1 时，使用 2 个的数据寄存器（双字数据）。

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

### 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。

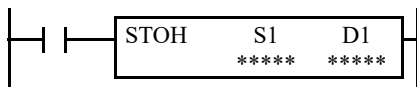


## 24: 时钟指令

### STOH (秒 → HMS)

将以秒的时间数据转换为小时、分和秒。

FT2J FC6A



秒 → 小时、分、秒

当输入打开，由源设备 S1 指定的秒的时间数据将转换为小时、分和秒。此结果存储到目标设备 D1。

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。

目标设备 D1 占用由设备指定开始的 3 个连续数据寄存器。

源设备 S1 的秒数据可以为 0 ~ 4,294,967,295。

**注释：**如果转换结果超出 65,535 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 65,537 小时 0 分 0 秒时，目标 1 存储 1 小时 0 分 0 秒，并打开特殊的内部继电器 M8003 (进位)。

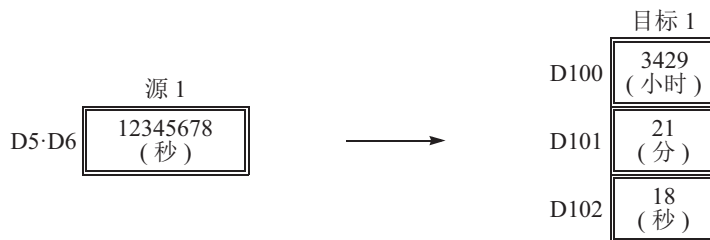
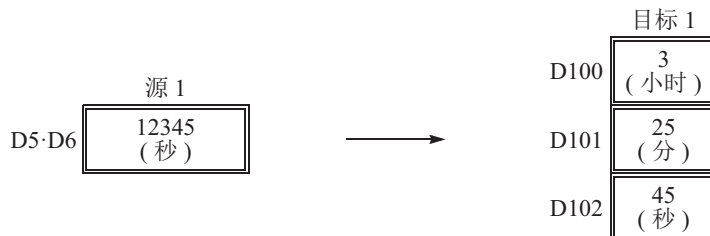
#### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	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	—	—	—

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

#### 示例：STOH

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。



# 25: 数据日志指令

## 简介

本章对可将指定的设备日志数据保存到 SD 记忆卡的数据日志指令进行介绍。

Plus CPU 模块可切换基本模式和高级模式 2 种保存方法。基本模式下将自动确定文件名。

高级模式下则可自行设置文件名，且可在任意时间切换。

## DLOG（数据日志）

DLOG 指令以指定的数据格式将指定设备的值保存为 SD 记忆卡上的 CSV 文件。

FT2J FC6A



当输入开启时，日期和时间以及指定设备的值将输出到 S1 指定文件夹中的 CSV 文件。指令执行完成后，D1 指定的设备将开启，执行状态被存储到 D2 指定的设备。

如果 SD 记忆卡中不存在 S1 指定的文件夹，将创建该文件夹。文件夹路径为“FCDATA01\DATALOG\用户指定的文件夹”。

所生成 CSV 文件的文件名如下所示。

### • All-in-One CPU 模块时

CSV 文件的文件名在 DLOG 指令打开时为“日期.csv”。

例) 日期为 2011 年 9 月 30 日时，为“20110930.csv”。

### • Plus CPU 模块时

通过 WindLDR 设置任意文件名作为 CSV 文件的文件名。另外，可对所设置文件名附加设备值及年月日。

例) 文件名为“LOG”、值为“12345”（零限点位数：5 位）且日期为 2017 年 4 月 1 日 23 时 59 分 01 秒时，CSV 文件名为“LOG\_12345\_170401\_235901.csv”。

如果 S1 指定的文件夹中不存在相同日期的文件，则创建 CSV 文件并输出标题和日志数据。

### 输出示例

Time	D0010	← 标题
2015/12/30 08:30:23	12345	← 日志数据

如果 S1 指定的文件夹中已存在相同日期的文件，则只会向该 CSV 文件中附加日志数据。

### 输出示例

Time	D0010	
2015/12/30 08:30:23	12345	
2015/12/30 17:30:23	1212	← 附加日志数据

DLOG 指令执行完成后，D1 指定的设备将开启，状态代码将根据执行结果存储到 D2 指定的设备。有关状态代码的详情，请参见第 25-4 页上的“3. D2（目标 2）：执行状态”。

## 25: 数据日志指令

### 注释:

- 停止 FC6A 型，再开始运行后，系统会在运行开始后初次执行 DLOG 指令时对 CSV 文件添加标题进行输出。Plus CPU 模块可选择是否输出标题。

### 输出示例

Time	D0010	← 标题
2015/12/30 08:30:23	12345	← 日志数据
Time	D0020	← 附加标题
2015/12/30 17:30:23	1212	← 日志数据

- 有关 SD 记忆卡规格的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。
- 有关日志数据文件大小的上限设置，请参见《FC6A 型 MICROSmart 用户手册》第 5 章“功能和设置”。

### 注释:

- 用户程序中可以编写 48 条 DLOG 指令。但请注意避免以 DLOG 指令指定的文件夹或文件重复。如果文件夹相同，将向同一 CSV 文件输出混合格式的日志数据。
- DLOG 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- DLOG 指令输入开启时，系统将向 CSV 文件重复输出日志数据。如果您只希望输出一次日志数据，请在输入条件中添加 SOTU（上升沿微分指令）或 SOTD（下降沿微分指令）。有关 SOTU 或 SOTD 的详情，请参见第 4-31 页上的“SOTU（上升沿微分）”。
- DLOG 指令向 SD 记忆卡写入数据的流程包括若干次扫描。一旦执行 DLOG 指令，该流程将一直继续到日志数据传输完成，不论指令输入发生任何变更。正在向 SD 记忆卡写入日志数据时，系统不会执行指令，即使开启 DLOG 指令输入也是如此。要再次执行 DLOG 指令，请确认之前的数据传输流程是否完成，然后再执行指令。
- 如果在将日志数据保存到 SD 记忆卡之前关闭 Plus CPU 模块的电源，日志数据将被清除。

### 有效设备

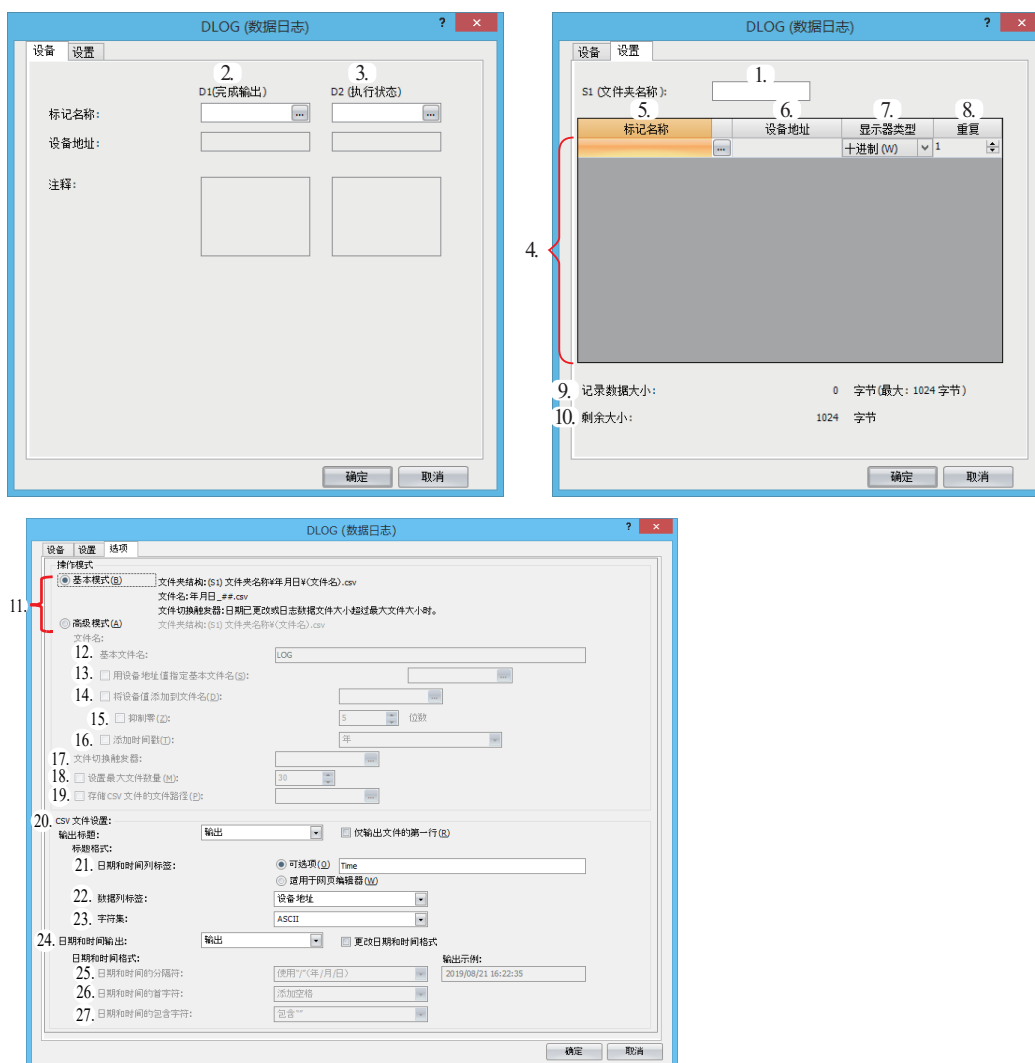
设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	文件夹名称（注释）	—	—	—	—	—	—	—	—	—	—
D1（目标 1）	完成输出	—	X	▲	—	—	—	—	—	—	—
D2（目标 2）	执行状态	—	—	—	—	—	—	▲	—	—	—

**注释:** 通过输入字符，指定文件夹名称。

▲无法将特殊内部继电器指定为 D1。无法将特殊数据寄存器指定为 D2。

关于有效设备地址范围，请参见第 2-12 页上的“FC6A 型”。

## 设置



## 注释:

- “选项”选项卡仅在 Plus CPU 模块中显示。
- 12. ~ 19. 中的项目仅在高级模式下可进行设置。
- 20. ~ 27. 在基本模式及高级模式下均可进行设置。

## 1. S1 (源 1): 文件夹名称

以所需的文本（最多为 8 个单字节字母数字字符）指定 SD 记忆卡中用来存储日志数据的文件夹名称。

## 注释:

- 文件夹名称中不能使用以下单字节字符：  
/ \ : \* ? " < > | # { } % & ~
- 文件夹名称中不能使用连续句号。
- 文件夹名称的开头或末尾不能使用句号。
- 文件夹名称开头或末尾的单字节空格将被忽略。

## 2. D1 (目标 1): 完成输出

指定向 SD 卡传输日志数据时以及 DLOG 指令执行完成后开启的设备。无论向 SD 记忆卡传输日志数据成功与否，此设备都会开启。

## 25: 数据日志指令

### 3. D2（目标 2）：执行状态

D2 指定用于 DLOG 指令的数据寄存器的第一数据寄存器。

从指定的数据寄存器开始，使用连续的 2 字数据寄存器。请指定数据寄存器范围内的第一数据寄存器。

存储目的地	功能	设置内容	参考
起始编号 +0	执行状态	存储状态代码。	第 25-4 页上的“执行状态”
起始编号 +1	文件容量率 *1	文件容量率 = “CSV 文件” / “外部存储器所设置的文件大小” *2 *100 (小数点以下进位)	

\*1 存储根据文件大小计算出的值。小数点以下将进位。

例) 将 D2 指定为 “D0000”、“外部存储器所设置的文件大小” 指定为 “5MB(5120KB)”，且 “CSV 文件” 为 “520KB” 时，  
文件容量率 = 520/5120 × 100 = 10.2 → 11  
将存储 “11”。

\*2 有关外部存储器的文件大小的设置方法，请参见《FC6A 型 MICROSmart 用户手册》第 5 章 “功能和设置”。

### 执行状态

指定存储状态代码的设备。将根据 DLOG 指令执行状态和结果存储以下状态代码。

#### All-in-One CPU 模块

状态代码	状态	说明
0	正常	—
1	SD 记忆卡插入错误	未插入 SD 记忆卡
2	SD 记忆卡容量错误	SD 记忆卡已满
3	SD 记忆卡写入错误	向 SD 记忆卡写入日志数据失败
4	CSV 文件容量错误	CSV 文件超过日志数据大小的上限
5	SD 记忆卡保护错误	SD 记忆卡已写保护
6	SD 记忆卡访问错误	在另一 DLOG 指令或 TRACE 指令执行时，系统会执行 DLOG 指令
8	文件夹创建错误	无法创建文件夹
9	CSV 文件打开错误	无法打开 CSV 文件
32	正在执行 DLOG 指令	正在向 SD 记忆卡写入日志数据

#### Plus CPU 模块

状态代码	状态	说明
0	正常	—
2	SD 记忆卡容量错误	SD 记忆卡已满
3	SD 记忆卡写入错误	向 SD 记忆卡写入日志数据失败
4	CSV 文件容量错误	1 个文件夹内 (1 日内) 的 CSV 文件数超过 99 个文件 (YYYYMMDD_01.csv ~ YYYYMMDD_99.csv)
5	SD 记忆卡保护错误	SD 记忆卡已写保护
6	SD 记忆卡访问错误	在另一 DLOG 指令或 TRACE 指令执行时，系统会执行 DLOG 指令
7	字符串转换错误	将日志数据转换为数字字符失败
8	文件夹创建错误	无法创建文件夹
9	CSV 文件打开错误	无法打开 CSV 文件
10	正常 (挥发性内存存储)	由于未插入 SD 记忆卡，已保存至 Plus CPU 模块的挥发性内存
11	挥发性内存上溢出	将日志数据存储到挥发性内存中因挥发性内存溢出而失败
12	路径存储设置错误	存储路径的数据寄存器已超出最大数据寄存器 (D7999、D61999、D269999)
13	CSV 文件大小错误	高级模式下 CSV 文件超出外部存储设备所设置的文件大小，因此不会保存日志数据
32	正在执行 DLOG 指令	正在向 SD 记忆卡写入日志数据



### 文件容量率

存储根据文件大小计算出的值。对小数点进行进位。

使用 CSV 文件和外部存储设备所设置的文件大小。

有关外部存储器的文件大小的设置方法，请参见《FC6A 型 MICROSmart 用户手册》第 5 章“功能和设置”。

文件容量率 = “CSV 文件” / “外部存储器所设置的文件大小”

例) 将 D2 指定为“D0000”、“外部存储器所设置的文件大小”指定为“5MB (5000KB)”，且“CSV 文件”为“520KB”时

将对计算结果“10.4”进行进位，存储“11”。

文件容量率  $520/5000 = 10.4 \rightarrow 11$

### 4. 设置

下列是可设置为要输出到 CSV 文件的数据的设备和保存格式列表。

显示类型	I	Q	M	R	T	TC	TP	C	CC	CP	D	P
十进制 (W)	—	—	—	—	—	X	X	—	X	X	X	—
十进制 (I)	—	—	—	—	—	—	—	—	—	—	X	—
十进制 (D)	—	—	—	—	—	—	—	—	X	X	X	—
十进制 (L)	—	—	—	—	—	—	—	—	—	—	X	—
十进制 (F)	—	—	—	—	—	—	—	—	—	—	X	—
十六进制 (W)	—	—	—	—	—	X	X	—	X	X	X	—
十六进制 (D)	—	—	—	—	—	—	—	—	X	X	X	—
二进制 (B)	X	X	X	X	X	—	—	X	—	—	—	—
字符串 (S)	—	—	—	—	—	—	—	—	—	—	X <sup>*1</sup>	—

\*1 仅 Plus CPU 模块

### 5. 标记名称

输入标记名称或设备地址，以指定要将值输出到 CSV 文件的设备。

### 6. 设备地址

将设备指定为标记名称时，系统将会显示相应的设备地址。

### 7. 显示器类型

从以下每台设备的表中选择显示类型，以了解何时将设备值输出到 CSV 文件。

显示类型	范围	最多字符数
十进制 (W)	0 到 65,535	5
十进制 (I)	-32,768 到 32,767	6
十进制 (D)	0 到 4,294,967,295	10
十进制 (L)	-2,147,483,648 到 2,147,483,647	11
十进制 (F)	-3.402823E+38 到 3.402823E+38	13
十六进制 (W)	0000 到 FFFF	4
十六进制 (D)	00000000 到 FFFFFFFF	8
二进制 (B)	0 或 1	1
字符串 (S)	“带 BOM 的 Unicode (UTF-8)”、“无 BOM 的 Unicode (UTF-8)”、“ASCII”、“日语 (Shift-JIS)”、“中文 (GB2312)”、“西欧 (ISO8859-1)”	100

### 注释:

- 将“显示器类型”设为“字符串 (S)”时，系统会将设备中存储的值输出到 CSV 文件。  
例如，使用 MOVC 指令将字符串存储到设备时，输出到 CSV 文件中的字符串将变为以 MOVC 指令选择的字符集。
- 将“显示器类型”设为“字符串 (S)”，且字符串内存在 ASCII 的控制字符时，系统将输出“?”。  
例) 0123[0x0D][0x0A]456[0x00] → 0123??456
- 将“显示器类型”设为“字符串 (S)”时，无论重复次数多少，系统都将注册为 1 个设备。
- 如需将“显示器类型”设为“字符串 (S)”，且在字符串内使用分隔符，请以双引号 (") 括起字符串。双引号 (") 属于 1 种转义排序。CSV 文件格式的规则是，将以双引号 (") 括起的值识别为 1 个字符串。如果不以 "" 括起字符串，系统会将分隔符之前的值识别为 1 个字符串。

## 25: 数据日志指令

### 8. 重复

从指定的设备地址开始，系统会向 SD 记忆卡输出与指定重复次数同样多的连续设备的数据。

例如，如果 D10 的显示类型为十进制 (W)，重复次数设为 5，则系统会按如下所示向 SD 记忆卡输出数据。

Time	D0010	D0011	D0012	D0013	D0014	
2015/12/30 15:40:00	12345	1	5	12	111	← 标题
2015/12/30 15:41:00	1212	3	7	35	222	← 日志数据 1
2015/12/30 15:42:00	345	4	99	79	333	← 日志数据 2
						← 日志数据 3

**注释：**将“显示器类型”设为“字符串 (S)”时，1 次重复 2 个字符。50 次重复最多可输出 100 个字符的字符串。

#### 设置示例 1

将“字符串”指定为“ABC,DEF”，“分隔符”指定为“逗号 (,)”，重复次数指定为“7”时

- 十六进制

设备	存储值			
	高位字节		低位字节	
	ASCII	十六进制	ASCII	十六进制
D1000	A	41h	B	42h
D1001	C	43h	逗号 (,)	2Ch
D1002	D	44h	E	45h
D1003	F	46h	NULL	00h

- 字符串

Time	D 1000	
2018/04/06 14:59:40	ABC	DEF
2018/04/06 14:59:50	ABC	DEF

#### 设置示例 2

将“字符串”指定为“”ABC,DEF””，“分隔符”指定为“逗号 (,)”，重复次数指定为“9”时

- 十六进制

设备	存储值			
	高位字节		低位字节	
	ASCII	十六进制	ASCII	十六进制
D1000	双引号 (")	22h	A	41h
D1001	B	42h	C	43h
D1002	逗号 (,)	2Ch	D	44h
D1003	E	45h	F	46h
D1004	双引号 (")	22h	NULL	00h

- 字符串

Time	D 1000	
2018/04/06 14:59:40	ABC,DEF	
2018/04/06 14:59:50	ABC,DEF	

## 设置示例 3

将“字符串”指定为“””””ABC,DEF””””，“分隔符”指定为“逗号（,）”，重复次数指定为“13”时

- 十六进制

设备	存储值			
	高位字节		低位字节	
	ASCII	十六进制	ASCII	十六进制
D1000	双引号 (")	22h	双引号 (")	22h
D1001	双引号 (")	22h	A	41h
D1002	B	42h	C	43h
D1003	逗号 (,)	2Ch	D	44h
D1004	E	45h	F	46h
D1005	双引号 (")	22h	双引号 (")	22h
D1006	双引号 (")	22h	NULL	00h

- 字符串

Time	D 1000	
2018/04/06 14:59:40	"ABC,DEF"	
2018/04/06 14:59:50	"ABC,DEF"	

## 9. 记录数据大小

显示 DLOG 指令用于当前日志设置的内存量。添加记录数据的设备时，所用的内存量会增加。您最多可以注册 64 台设备（内存总量必须小于或等于 1,024 字节）。每个字符需要占用 1 字节的内存区域。

## 10. 剩余大小

显示可用的内存量（即记录数据大小与 1,024 字节之差）。

## 11. 操作模式

从“基本模式”或“高级模式”中选择“操作模式”。

以下的 12. ~ 19. 中的项目仅在高级模式下可进行设置。

## 12. 基本文件名

输入所输出数据的文件名。

默认为“LOG”。最多可输入 116 个半角字符（不含扩展名）。

## 13. 用设备地址值指定基本文件名

选择设置基本文件名。

选中复选框后，可指定数据寄存器。

未选中复选框时，文件名保持 12. “基本文件名”不变。

指定用作基本文件名的数据寄存器。最多可使用 40 点，并设置最多 80 个字符（不含扩展名）。

以设置的设备为起始依序读取数值，将 NULL（00h）之前的对象或最多 80 个字符作为字符数据进行处理，然后作为基本文件名。

例）设备地址指定“D0100”时，固定值将变为“IDEC”。

设备	存储值			
	高位字节		低位字节	
	ASCII	十六进制	ASCII	十六进制
D1000	I	48h	D	44h
D1001	E	45h	C	43h
D1002	NULL	00h	NULL	00h

## 注释:

- 无法在“基本文件名”或“用设备地址值指定基本文件名”所设置的文件名中使用以下字符。  
 \ : \* ? " < > | # { } % & ~

## 25: 数据日志指令

- 如果在“用设备地址值指定基本文件名”(13.)中设置了无法使用的字符，文件名如下所示。
  - 设置无法使用的字符时，则文件名为该字符之前的字符。
  - 如果起始字符为无法使用的字符，文件名将变为“基本文件名”中所设置的字符。
  - 文件名开头的半角空格将被删除。

### 14. 将设备值添加到文件名

选择将设备值附加至文件名。

选中复选框后，可指定数据寄存器。

未选中复选框时，无法指定。

指定要附加至文件名的值的读取源数据寄存器。使用 2 个字。

将按照基本文件名、“\_”（半角下划线）、设备值的顺序进行附加。

请在 0 ~ 999999999（十进制）的范围内设置设备值。

数据寄存器中存储的数值超过 999999999 时，仅附加 999999999。

### 15. 抑制零

指定不附加“0”的零限值。选中复选框后，将执行零限值处理。未选中复选框时，可指定位数。设备值中附加“0”，以补足指定位数。如果设备值的位数大于设置的位数，则设备值将保持不变输出。

此外，如果设备值的位数大于配置的位数，将直接输出设备值。

例) “基本文件名”(12.)为“LOG”，选中“将设备值添加到文件名”(14.)复选框，设备地址为 D0200。此外，如果清除了“抑制零”复选框，且位数设置如下，则文件名如下。

位数	CSV 文件名
8 位数	LOG_00123456.CSV
6 位数	LOG_123456.CSV
4 位数	LOG_123456.CSV

设备	存储值	
	十进制 (字)	十进制 (双字)
D0200	1	123456
D0201	57920	

### 16. 添加时间戳

选择将输出日期时间附加至文件名。

选中复选框后，将附加输出日期时间。可以选择输出日期时间的格式。

未选中复选框时，不会附加输出日期时间。

从以下 6 种选项中选择要附加的输出日期时间格式。

“年”、“年+月”、“年+月+日”、“年+月+日+时”、“年+月+日+时+分”、

“年+月+日+时+分+秒”

格式为 YYMMDD\_hhmmss (YY: 年、MM: 月、DD: 日、hh: 时、mm: 分、ss: 秒)。

将按照基本文件名、“\_”（半角下划线）、输出日期时间的顺序进行附加。

例) “文件名”为“LOG”，输出日期时间为 2013 年 9 月 15 日 23 时 30 分 50 秒时

“年”： LOG\_13  
“年+月”： LOG\_1309  
“年+月+日”： LOG\_130915  
“年+月+日+时”： LOG\_130915\_23  
“年+月+日+时+分”： LOG\_130915\_2330  
“年+月+日+时+分+秒”： LOG\_130915\_233050

### 17. 文件切换触发器

指定用作切换文件触发器的设备。当开启指定设备并执行 DLOG 指令时，系统将切换输出数据的文件。

以 12. ~ 16. 中指定的文件名生成新文件。进行切换时，设备将自动关闭。如果已存在同名文件，则不会生成新文件，而是将数据添加到相应文件中。

### 18. 设置最大文件数量

限制 SD 记忆卡内 CSV 保存文件夹中所保存的 CSV 文件数。

选中复选框后，可设置限制。可在 1 ~ 100 的范围内设置上限文件数。文件数达到上限后，系统将删除最旧的时间戳文件，并生成新文件。

未选中复选框时，不会限制文件数。系统将持续生成 CSV 文件，直到超出 SD 记忆卡容量为止。

**注释：**如果 SD 记忆卡内保存的文件数在开始运行时已经超过上限，当时的文件数即为上限。

### 19. 存储 CSV 文件的文件路径

要存储 CSV 文件的路径时，应选中复选框。

指定执行存储的起始数据寄存器。通过在网页编辑器的趋势棒状图等中设置本数据寄存器，可在网页编辑器中显示 CSV 文件。

将在切换 CSV 文件的文件名时指定的数据寄存器中存储 CSV 文件的路径。

CSV 文件的路径以“D:\ 文件夹名称 \CSV 文件名”的格式存储在数据寄存器内。有关详情，请参见《FC6A 型 MICROSmart 使用手册 通信篇》第 13 章“网络服务器”。

20. ~ 23. 在基本模式及高级模式下均可进行设置。

### 20. 输出标题

指定是否对 CSV 文件输出标题。

仅将文件的起始行输出为标题时，应选中“仅输出文件的第一行”复选框。

- 不输出标题时

2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64
2018/04/06 15:00:00	31	35	27	32	62	70	54	64
2018/04/06 15:00:10	31	36	27	32	62	72	54	64
2018/04/06 15:00:20	32	34	28	31	64	68	56	62
2018/04/06 15:00:30	31	35	27	32	62	70	70	64

- 输出标题时

Time	D 1000	D 1001	D 1002	D 1003	D 1004	D 1005	D 1006	D 1007
2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64
Time	D 1000	D 1001	D 1002	D 1003	D 1004	D 1005	D 1006	D 1007
2018/04/06 15:00:20	32	34	28	31	64	68	56	62
2018/04/06 15:00:30	31	35	27	32	62	70	70	64

- 仅将文件的起始行输出为标题时

Time	D 1000	D 1001	D 1002	D 1003	D 1004	D 1005	D 1006	D 1007
2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64
2018/04/06 15:00:00	31	35	27	32	62	70	54	64
2018/04/06 15:00:10	31	36	27	32	62	72	54	64
2018/04/06 15:00:20	32	34	28	31	64	68	56	62

### 21. 日期和时间列标签

指定 CSV 文件标题的执行日期和时间列标签。从“可选项”或“适用于网页编辑器”中选择。

“可选项”的默认值为“Time”。

如果选择“可选项”，系统将输出文本框中输入的字符串。最多可设置 120 个字节（字符集为 UTF-8 时，为 40 个字符左右）。

选择“适用于网页编辑器”时，可在网页编辑器中使用趋势棒状图部件。

**注释：**有关趋势棒状图部件的详情，请参见《FC6A 型 MICROSmart 通信手册》第 13 章“Web 服务器”。

- 可选项（预置值：Time）

Time	D 1000	D 1001	D 1002	D 1003	D 1004	D 1005	D 1006	D 1007
2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64

## 25: 数据日志指令

- 适用于网页编辑器

#_date_time	D 1000	D 1001	D 1002	D 1003	D 1004	D 1005	D 1006	D 1007
2018-04-06 14:59:40	30	34	28	30	60	68	56	60
2018-04-06 14:59:50	32	35	27	32	64	70	54	64

**注释:** 选择“适用于网页编辑器”时, 执行年月日的字符串格式将变为 YYYY-MM-DD。

在 Microsoft Internet Explorer 中, 当使用网页编辑器的趋势图工具栏时, 请在勾选“更改日期和时间格式”后, 在“日期和时间的分隔符”中设置“使用‘/’ (年/月/日)”。

### 22. 数据列标签

指定 CSV 文件标题的数据列标签。从“设备地址”、“标记名称”或“注释”中选择

- 设备地址

Time	D 1000	D 1001	D 1002	D 1003	D 1004	D 1005	D 1006	D 1007
2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64

- 标记

Time	气温 1	气温 2	气温 3	气温 4	湿度 1	湿度 2	湿度 3	湿度 4
2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64

- 注释

Time	气温_大阪	气温_京都	气温_滋贺	气温_兵库	湿度_大阪	湿度_京都	湿度_滋贺	湿度_兵库
2018/04/06 14:59:40	30	34	28	30	60	68	56	60
2018/04/06 14:59:50	32	35	27	32	64	70	54	64

### 23. 字符集

从以下选项中选择 CSV 文件的字符集。

“带 BOM 的 Unicode (UTF-8)”、“无 BOM 的 Unicode (UTF-8)”、“ASCII”、“日语 (Shift-JIS)”、“中文 (GB2312)”、“西欧 (ISO8859-1)”

如需在网页编辑器中使用趋势图栏部件, 请选择“带 BOM 的 Unicode (UTF-8)”或“无 BOM 的 Unicode (UTF-8)”之一。

**注释:**

- 在 7. “显示器类型”中选择了“字符串(S)”时, 请对设备中存储的字符串和本设置选择相同字符集。
- “带 BOM 的 Unicode (UTF-8)”时, BOM 值 (0xEF 0xBB 0xBF) 会输出为开始 3 字节。
- “无 BOM 的 Unicode (UTF-8)”时, 无法以“UTF-8”打开, 可能会出现乱码。

### 24. 日期和时间输出

指定是否向 CSV 文件输出日期和时间。

- 输出日期和时间时

Time	D 1000
2018/04/06 14:59:40	30

- 不输出日期和时间时

D 1000
30

(25) ~ (27) 的项目可在勾选“更改日期和时间格式”后进行设置。当取消勾选“更改日期和时间格式”时, 会变为以下日期和时间格式。

- 在“日期和时间列标签”中选择“可选项”时

Time	D 1000
2018/04/06 14:59:40	30

- 在“日期和时间列标签”中选择“适用于网页编辑器”时

#_date_time_	D 1000
2018/04/06 14:59:40	30

**25. 日期和时间的分隔符**

以 CSV 文件的日期和时间格式指定年月日的分隔符。

可选择“使用“/”（年/月/日）”或“使用“-”（年-月-日）”。

- 使用“/”（年/月/日）

Time	D 1000
2018/04/06 14:59:40	30

- 使用“-”（年-月-日）

Time	D 1000
2018-04-06 14:59:40	30

**26. 日期和时间的首字符**

以 CSV 文件的日期和时间格式指定首字符是否有空格。

可选择“添加空格”或“请勿添加空格”。

- 添加空格

Time	D 1000
2018/04/06 14:59:40	30

- 请勿添加空格

Time	D 1000
2018-04-06 14:59:40	30

**27. 日期和时间的包含字符**

以 CSV 文件的日期和时间格式指定是否使用双引号。

可选择“包含“””或“请勿包含“””。

- 包含“”

Time	D 1000
2018/04/06 14:59:40	30

- 请勿包含“”

Time	D 1000
2018-04-06 14:59:40	30

**注释：**在“日期和时间输出”中选择了“输出”的 CSV 文件用表格计算软件打开后，将根据表格计算软件的显示格式显示日期和时间。

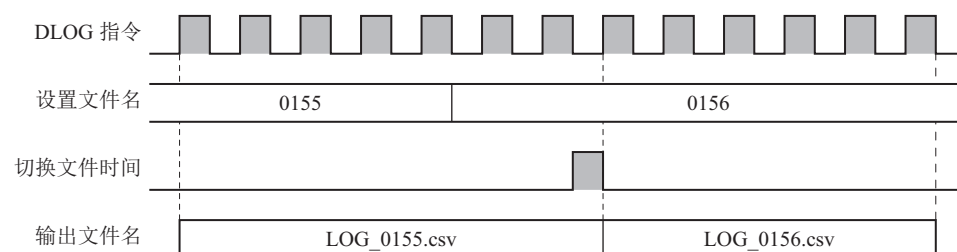




### 高级模式

如果在执行 DLOG 指令时，S1 指定的文件夹中不存在名称相同的 CSV 文件，则系统会创建新的文件并按以上输出格式示例所示输出标题和日志数据 1。如果在同一名称又执行了一次 DLOG 指令，则 CSV 文件中会附加日志数据 2。同理，如果在同一名称又执行了一次 DLOG 指令，CSV 文件中会附加日志数据 3。

当开启切换文件时间触发器，且文件名变更时，系统会用一个新文件名创建新的 CSV 文件并输出标题和日志数据。切换文件时间触发器将在执行 DLOG 指令时自动关闭。



#### • 文件格式

可在功能设置中指定 CSV 文件的十进制符号和分隔符。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

#### • 文件名

文件名输出顺序如下所示。

[ 文件名 ] [ 设备地址的值 ] [YYMMDD] [hhmmss].CSV

文件名：在“基本文件名”中输入的字符或“用设备地址值指定基本文件名”所指定的设备地址值字符

设备地址的值：在“将设备值添加到文件名”中指定的设备地址值

YYMMDD：在“添加时间戳”中设置的日期时间的年、月、日

hhmmss：在“添加时间戳”中设置的日期时间的时、分、秒

#### • 设置示例 1

项目	设置	
基本文件名	LOG	
将设备值添加到文件名	设备地址：D0200	D0200: 123456
添加时间戳	年 + 月	输出数据的日期时间：2013 年 9 月

文件名变为“LOG\_123456\_1309.CSV”。

#### • 设置示例 2

项目	设置			
基本文件名	LOG			
用设备地址值指定基本文件名	设备地址：D0100 设置的字符为“IDEC”	设备	存储值	
			高位字节	低位字节
		D0100	1	D
		D0101	E	C
		D0102	NULL	NULL
将设备值添加到文件名	设备地址：D0200	D0200: 123456		
添加时间戳	年 + 月 + 日 + 分 + 秒	输出数据的日期时间：2013 年 9 月 15 日、23:30:50		

文件名变为“IDEC\_123456\_130915\_233050.CSV”。

## 25: 数据日志指令

### 示例：DLOG

M0 开启时，D0 到 D5 的十进制值（数据类型 W（字））以及 D10 的十进制值（数据类型 F（浮点））每 10 秒保存到 SD 记忆卡的“RESULT”文件夹中的 CSV 文件一次。

### 输出示例

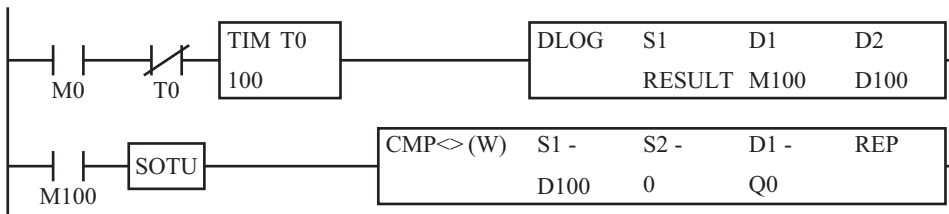
Time	D0000	D0001	D0002	D0003	D0004	D0005	D0010
2012/02/06 10:20:30	12345	0	0	56789	0	0	-3.402823E+38
2012/02/06 10:20:40	12345	0	0	56789	0	0	-3.402823E+38
2012/02/06 10:20:50	12345	0	0	56789	0	0	-3.402823E+38

下面描述的示例用户程序运行如下所示。

- 向 SD 记忆卡传输日志数据完成时，系统会开启 M100。
- DLOG 指令状态代码被存储到 D100 中。
- 检查 D100 中保存的状态代码，如果出现错误，则开启 Q0。

### • 设置流程

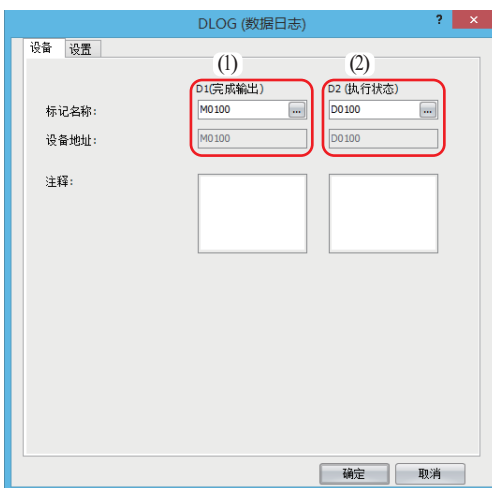
#### 1. 创建梯形图程序。



#### 2. 设置 DLOG 指令。

设置设备选项卡。

- (1) 将 M0100 指定为 D1（完成输出）。
- (2) 将 D0100 指定为 D2（执行状态）。



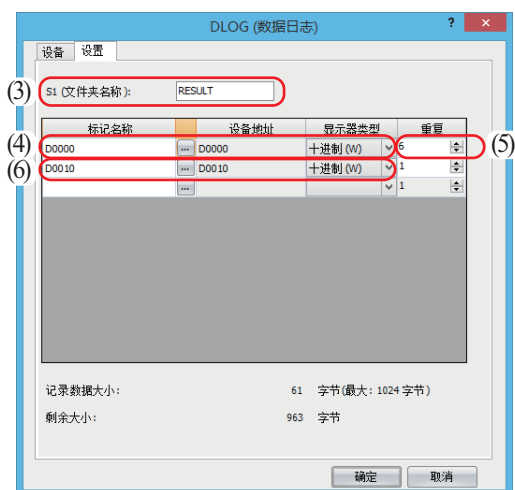
设置设置选项卡。

(3) 在“S1 (文件夹名称)”中输入“RESULT”。

(4) 将 D0000 设置为 D0000 的十进制值以数据类型 W (字) 形式输出到 CSV 文件。

(5) 将重复次数设为 6 次，以将 D0000 到 D0005 的值输出到 CSV 文件。

(6) 将 D0010 设置为 D0010 的十进制值以数据类型 F (浮点) 形式输出到 CSV 文件。



此时即完成配置。

#### • 操作说明

M0 开启 10 秒后，系统会执行一次 DLOG 指令。执行 DLOG 指令时，D0 ~ D5 和 D10 的数据将以十进制值格式输出到 SD 记忆卡上的 CSV 文件中，附带当前的日期和时间。

CSV 文件的保存位置为 FCDATA01\DATALOG\RESULT。最早的数据保存到日志数据的顶部，最新的数据保存到日志数据的底部。

DLOG 指令执行完成后，将开启完成输出 M100，并执行一次 CMP 指令。CMP 指令将存储于执行状态 D100 中的状态代码与 0 进行对比，并打开或关闭 Q0。如果 DLOG 指令中出现错误，将开启 Q0。

当 M0 开启时，日志数据将每 10 秒添加到 CSV 文件一次。

#### 输出结果

Time	D0000	D0001	D0002	D0003	D0004	D0005	D0010
2012/02/06 10:20:30	12345	0	0	56789	0	0	-3.402823E+38
2012/02/06 10:20:40	12345	0	0	56789	0	0	-3.402823E+38
2012/02/06 10:20:50	12345	0	0	56789	0	0	-3.402823E+38

## TRACE（数据跟踪）

TRACE 指令以指定的数据格式将指定设备上上次扫描的值保存为 SD 记忆卡上的 CSV 文件。

FT2J FC6A



当开启输入时，日期和时间以及指定设备上上次扫描的值将输出到 S1 指定文件夹中的 CSV 文件。

指令执行完成后，D1 指定的设备将开启，执行状态被存储到 D2 指定的设备。

如果 SD 记忆卡中不存在 S1 指定的文件夹，将创建该文件夹。文件夹路径为“FCDATA01\TRACE\用户指定的文件夹”。有关文件夹结构的详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

所生成 CSV 文件的文件名如下所示。

- All-in-One CPU 模块时

CSV 文件的文件名在 TRACE 指令打开时为“日期.csv”。

例) 日期为 2011 年 9 月 30 日时，为“20110930.csv”。

- Plus CPU 模块时

以年月日（YYYYMMDD）为名称创建文件夹，在该文件夹内保存文件。文件名中附加“\_”+“2 位数字（00～99）”。文件大小超过日志数据文件大小时，将生成递增“2 位数字（00～99）”的新文件。

例) 日期为 2017 年 4 月 1 日的最初文件时，为“20170401\20170401\_00.csv”。

如果 S1 指定的 CSV 文件夹中不存在相同日期的文件，则创建 CSV 文件并输出跟踪数据。最早的数据输出到跟踪数据的顶部，最新的数据输出到跟踪数据的底部。

### 输出示例

Triggered at:	2012/02/06 08:30:23	← 标题行 1
Scan	D0010	← 标题行 2
Old	12345	← 上次数据 2 扫描
	12345	← 上次数据 1 扫描
New	12345	← 最新数据

如果 S1 指定的文件夹中已存在相同日期的 CSV 文件，则系统会向该 CSV 文件附加标题和跟踪数据。

### 输出示例

Triggered at:	2012/02/06 08:30:23	
Scan	D0010	
Old	12345	
	12345	
New	12345	
Triggered at:	2012/02/06 17:16:15	← 附加标题行 1
Scan	D0010	← 附加标题行 2
Old	1212	← 附加上次数据 2 扫描
	1212	← 附加上次数据 1 扫描
New	1212	← 附加最新数据

TRACE 指令执行完成后，D1 指定的设备将开启，状态代码将根据执行结果存储到 D2 指定的设备。有关状态代码的详情，请参见第 25-18 页上的“3. D2（目标 2）：执行状态”。

**注释:**

- TRACE 指令在 FC6A 型运行时累积数据，而在 FC6A 型停止时则不会累积数据。
- 即使 TRACE 指令输入关闭，系统也会在 FC6A 型运行时累积数据。
- 当 TRACE 指令输入开启时，累积的数据将输出到 CSV 文件。
- 即使 MCS（主控继电器开始）指令开启，也将累积跟踪数据。
- 有关日志数据文件大小的上限设置，请参见《FC6A 型 MICROSmart 用户手册》第 5 章“功能和设置”。
- 用户程序中最多可以编写 3 条 TRACE 指令。请确保该 TRACE 指令指定的文件夹名称与其他 TRACE 指令指定的文件夹名称不同。如果文件夹相同，将向同一 CSV 文件输出混合格式的跟踪数据。
- 执行一次 TRACE 指令时，可向 CSV 文件保存的跟踪数据的扫描次数取决于指定跟踪的设备数以及每台设备的显示类型。有关详情，请参见第 25-19 页上的“9. 跟踪数据大小”。
- TRACE 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 如如果 TRACE 指令被 JMP（跳转）指令跳过，则系统不会累积跟踪数据。有关 JMP（跳转）指令的详情，请参见第 4-35 页上的“JMP（跳转）”。
- TRACE 指令输入开启时，系统将向 CSV 文件重复输出跟踪数据。如果您只希望输出一次跟踪数据，请在输入条件中添加 SOTU（上升沿微分指令）或 SOTD（下降沿微分指令）。有关 SOTU（上升沿微分指令）或 SOTD（下降沿微分指令）的详情，请参见第 4-31 页上的“SOTU（上升沿微分）”。
- TRACE 指令向 SD 记忆卡写入数据的流程包括若干次扫描。一旦执行 TRACE 指令，该流程将一直继续到跟踪数据传输完成，不论指令输入发生任何变更。正在向 SD 记忆卡写入跟踪数据时，系统不会执行指令，即使开启 TRACE 指令输入也是如此。要再次执行 TRACE 指令，请确认之前的数据写入流程是否完成，然后再执行指令。

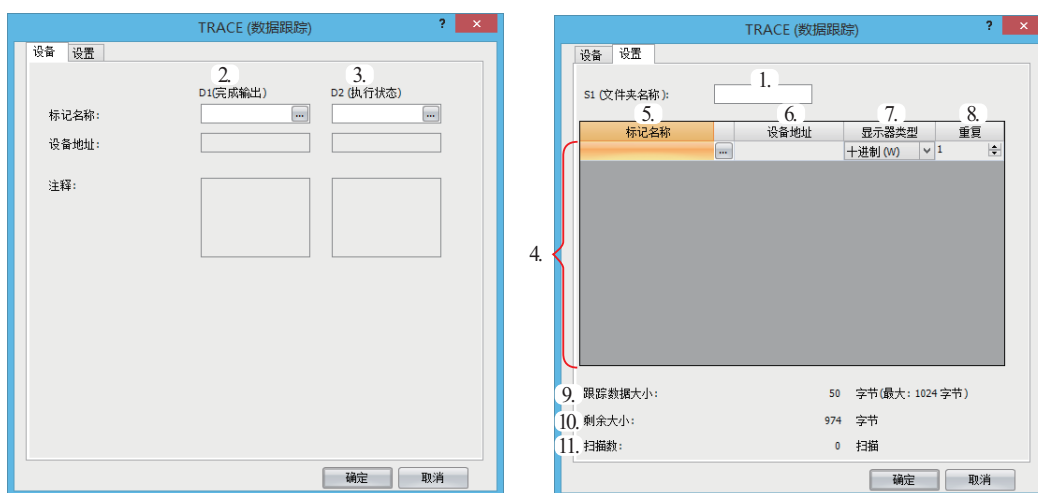
**有效设备**

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	文件夹名称（注释）	—	—	—	—	—	—	—	—	—	—
D1（目标 1）	完成输出	—	X	▲	—	—	—	—	—	—	—
D2（目标 2）	执行状态	—	—	—	—	—	—	▲	—	—	—

**注释:** 通过输入字符，指定文件夹名称。

▲无法将特殊内部继电器指定为 D1。无法将特殊数据寄存器指定为 D2。

关于有效设备地址范围，请参见第 2-12 页上的“FC6A 型”。

**设置****1. S1（源 1）：文件夹名称**

使用最多 8 个单字节字母数字字符指定 SD 记忆卡中用来存储跟踪数据的文件夹名称。

**注释:**

- 文件夹名称中不能使用以下单字节字符：  
/ \ : \* ? " < > | # { } % & ~
- 文件夹名称中不能使用连续句号。
- 文件夹名称的开头或末尾不能使用句号。
- 文件夹名称开头或末尾的单字节空格将被忽略。

## 25: 数据日志指令

### 2. D1（目标 1）：完成输出

指定向 SD 卡传输跟踪数据时以及 TRACE 指令执行完成后开启的设备。无论向 SD 记忆卡传输跟踪数据成功与否，此设备都会开启。

### 3. D2（目标 2）：执行状态

指定存储状态代码的数据寄存器。将根据 TRACE 指令执行状态和结果存储以下状态代码。

#### All-in-One CPU 模块

状态代码	说明	详情
0	正常	—
1	SD 记忆卡插入错误	未插入 SD 记忆卡
2	SD 记忆卡容量错误	SD 记忆卡已满
3	SD 记忆卡写入错误	向 SD 记忆卡写入跟踪数据失败
4	CSV 文件容量错误	CSV 文件超过日志数据大小的上限
5	SD 记忆卡保护错误	SD 记忆卡已写保护
6	SD 记忆卡访问错误	在另一 DLOG 指令或 TRACE 指令执行时，系统会执行 TRACE 指令
8	文件夹创建错误	无法创建文件夹
9	CSV 文件打开错误	无法打开 CSV 文件
32	正在执行 TRACE 指令	正在写入 SD 记忆卡

#### Plus CPU 模块

状态代码	说明	详情
0	正常	—
2	SD 记忆卡容量错误	SD 记忆卡已满
3	SD 记忆卡写入错误	向 SD 记忆卡写入跟踪数据失败
4	CSV 文件容量错误	1 个文件夹内（1 日内）的 CSV 文件数超过 99 个文件（YYYYMMDD_01.csv ~ YYYYMMDD_99.csv）
5	SD 记忆卡保护错误	SD 记忆卡已写保护
6	SD 记忆卡访问错误	在另一 DLOG 指令或 TRACE 指令执行时，系统会执行 TRACE 指令
7	字符串转换错误	将跟踪数据转换为数字字符失败
8	文件夹创建错误	无法创建文件夹
9	CSV 文件打开错误	无法打开 CSV 文件
10	正常（挥发性内存存储）	由于未插入 SD 记忆卡，已保存至 Plus CPU 模块的挥发性内存
11	挥发性内存上溢出	将日志数据存储到挥发性内存中因挥发性内存溢出而失败
32	正在执行 TRACE 指令	正在向 SD 记忆卡写入跟踪数据

## 4. 设置

下列是可设置为要输出到 CSV 文件的数据的设备和保存格式列表。

显示类型	I	Q	M	R	T	TC	TP	C	CC	CP	D	P
十进制 (W)	—	—	—	—	—	X	X	—	X	X	X	—
十进制 (I)	—	—	—	—	—	—	—	—	—	—	X	—
十进制 (D)	—	—	—	—	—	—	—	—	X	X	X	—
十进制 (L)	—	—	—	—	—	—	—	—	—	—	X	—
十进制 (F)	—	—	—	—	—	—	—	—	—	—	X	—
十六进制 (W)	—	—	—	—	—	X	X	—	X	X	X	—
十六进制 (D)	—	—	—	—	—	—	—	—	X	X	X	—
二进制 (B)	X	X	X	X	X	—	—	X	—	—	—	—
字符串 (S)	—	—	—	—	—	—	—	—	—	—	X <sup>*1</sup>	—

\*1 仅 Plus CPU 模块

### 5. 标记名称

输入标记名称或设备地址，以指定要输出到 CSV 文件的设备。

### 6. 设备地址

将设备指定为标记名称时，系统将会显示相应的设备地址。

### 7. 显示器类型

从以下每台设备的表中选择显示类型，以了解何时将设备值输出到 CSV 文件。

显示类型	范围	最多字符数
十进制 (W)	0 到 65,535	5
十进制 (I)	-32,768 到 32,767	6
十进制 (D)	0 到 4,294,967,295	10
十进制 (L)	-2,147,483,648 到 2,147,483,647	11
十进制 (F)	-3.402823E+38 到 3.402823E+38	13
十六进制 (W)	0000 到 FFFF	4
十六进制 (D)	00000000 到 FFFFFFFF	8
二进制 (B)	0 或 1	1

### 8. 重复

从指定的设备地址开始，系统会向 SD 记忆卡输出与指定重复次数同样多的连续设备的数据。

例如，如果 D10 的显示类型为十进制 (W)，重复次数设为 8，则系统会按如下所示向 SD 记忆卡输出数据。

Triggered at:	2015/12/30 15:40:30								← 标题行 1
Scan	D0010	D0011	D0012	D0013	D0014	D0015	D0016	D0017	← 标题行 2
Old	1	9	17	25	33	41	49	57	← 上次数据 7 扫描
	2	10	18	26	34	42	50	58	← 上次数据 6 扫描
	3	11	19	27	35	43	51	59	← 上次数据 5 扫描
	4	12	20	28	36	44	52	60	← 上次数据 4 扫描
	5	13	21	29	37	45	53	61	← 上次数据 3 扫描
	6	14	22	30	38	46	54	62	← 上次数据 2 扫描
	7	15	23	31	39	47	55	63	← 上次数据 1 扫描
New	8	16	24	32	40	48	56	64	← 最新数据

### 9. 跟踪数据大小

显示 TRACE 指令用于当前跟踪设置的内存量。添加跟踪数据的设备时，所用的内存量会增加。您最多可以注册 64 台设备（内存总量必须小于或等于 1,024 字节）。每个字符需要占用 1 字节的内存区域。

### 10. 剩余大小

显示可用的内存量（即跟踪数据大小与 1,024 字节之差）。

### 11. 扫描数

此项显示使用当前的跟踪设置可以累积多少次跟踪数据扫描。可以累积的数据扫描数取决于输出的跟踪数据格式。如果每次扫描要输出的数据很少，则可以累积多次扫描的数据。

## 25: 数据日志指令

### CSV 文件输出格式和文件格式设置

#### • 输出格式

CSV 文件输出格式如下所示。

```
Triggered at:,2015/12/30 15:40:30
Scan,D0010,D0020,D0030,D0040,D0050,D0060,D0070,D0080
Old,1,9,17,25,33,41,49,57
,2,10,18,26,34,42,50,58
,3,11,19,27,35,43,51,59
,4,12,20,28,36,44,52,60
,5,13,21,29,37,45,53,61
,6,14,22,30,38,46,54,62
,7,15,23,31,39,47,55,63
New,8,16,24,32,40,48,56,64
:
:
```

如果在执行 TRACE 指令时，S1 指定的文件夹中不存在日期相同的 CSV 文件，则系统会创建新的 CSV 文件并按以上输出格式示例所示输出跟踪数据。如果日期发生变化且 TRACE 指令执行，则系统会使用新的文件名创建一个新 CSV 文件。

#### • 文件格式

可在功能设置中指定 CSV 文件的十进制符号和分隔符。有关详情，请参见《FC6A 型 MICROSmart 用户手册》第 11 章“SD 记忆卡”。

### 示例：TRACE

M0 开启时，D0 到 D5 的累积数据（数据类型 W（字））以及 D10 的累积数据（数据类型 F（浮点））以十进制值的形式保存到 SD 记忆卡的“RESULT”文件夹中的 CSV 文件。

#### 输出示例

Triggered at:	2012/02/06 10:20:30						
Scan	D0000	D0001	D0002	D0003	D0004	D0005	D0010
Old	12345	2	12345	56789	1	56789	-3.402823E+38
	12345	2	12347	56789	1	56788	-3.402823E+38
	12345	2	12349	56789	1	56787	-3.402823E+38
:	:	:	:	:	:	:	:
	12345	2	12379	56789	1	56772	-3.402823E+38
	12345	2	12381	56789	1	56771	-3.402823E+38
New	12345	2	12383	56789	1	56770	-3.402823E+38

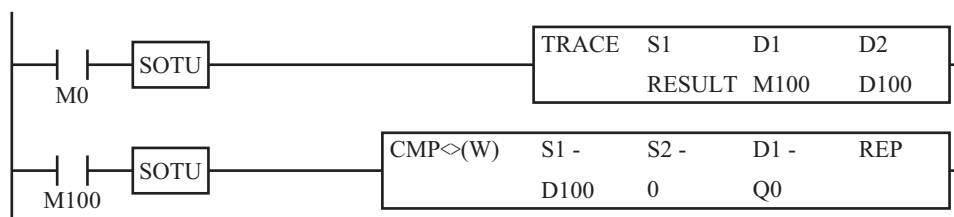
下面描述的示例用户程序运行如下所示。

- 向 SD 记忆卡写入跟踪数据完成时，系统会开启 M100。
- TRACE 指令状态代码被存储到 D100 中。
- 检查 D100 中保存的执行状态，如果出现错误，则开启 Q0。



- 设置流程

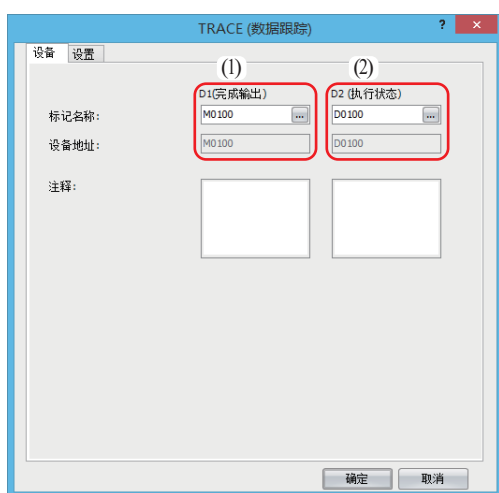
1. 创建梯形图程序。



2. 设置 TRACE 指令。

设置设备选项卡。

- (1) 将 M0100 指定为 D1（完成输出）。
- (2) 将 D0100 指定为 D2（执行状态）。



设置设置选项卡。

- (3) 在 S1 (文件夹名称) 中输入“RESULT”。
- (4) 将 D0000 设置为 D0000 的十进制值以数据类型 W（字）形式输出到 CSV 文件。
- (5) 将重复次数设为 6 次，以将 D0000 到 D0005 的值输出到 CSV 文件。
- (6) 将 D0010 设置为 D0010 的十进制值以数据类型 F（浮点）形式输出到 CSV 文件。



此时即完成配置。

## 25: 数据日志指令

### • 操作说明

当 M0 开启时，执行一次 TRACE 指令。执行 TRACE 指令时，前 17 次扫描的 D0 到 D5 和 D10 的数据将以十进制值格式输出到 SD 记忆卡上的 CSV 文件中，附带执行的日期和时间。

CSV 文件的保存位置为 FCDATA01\TRACE\RESULT。最早的数据保存到跟踪数据的顶部，最新的数据保存到跟踪数据的底部。

TRACE 指令执行完成后，将开启完成输出 M100，并执行一次 CMP 指令。CMP 指令将存储于执行状态 D100 中的状态代码与 0 进行对比，并打开或关闭 Q0。如果 TRACE 指令中出现错误，则开启 Q0。

### 输出结果

Triggered at:	2012/02/06 10:20:30						
Scan	D0000	D0001	D0002	D0003	D0004	D0005	D0010
Old	12345	2	12345	56789	1	56789	-3.402823E+38
	12345	2	12347	56789	1	56788	-3.402823E+38
	12345	2	12349	56789	1	56787	-3.402823E+38
	12345	2	12351	56789	1	56786	-3.402823E+38
	12345	2	12353	56789	1	56785	-3.402823E+38
	12345	2	12355	56789	1	56784	-3.402823E+38
	12345	2	12357	56789	1	56783	-3.402823E+38
	12345	2	12359	56789	1	56782	-3.402823E+38
	12345	2	12361	56789	1	56781	-3.402823E+38
	12345	2	12363	56789	1	56780	-3.402823E+38
	12345	2	12365	56789	1	56779	-3.402823E+38
	12345	2	12367	56789	1	56778	-3.402823E+38
	12345	2	12369	56789	1	56777	-3.402823E+38
	12345	2	12371	56789	1	56776	-3.402823E+38
	12345	2	12373	56789	1	56775	-3.402823E+38
	12345	2	12375	56789	1	56774	-3.402823E+38
New	12345	2	12377	56789	1	56773	-3.402823E+38

# 26: 脚本

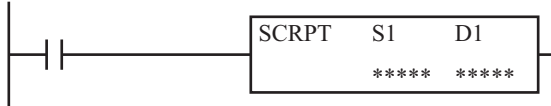
## 简介

本章将对通过梯形图程序调用并执行脚本的 SCRPT 指令进行介绍。

## SCRPT (脚本)

执行指定的脚本。

FT2J FC6A



**注释：**有关 SCRPT 指令执行的脚本中可使用的设备详情，请参见第 26-4 页上的“设备列表”。要使用 FT2J 型的内部 HMI 设备和连接到 FT2J 型的外部设备的内部设备时，请使用 WindO/I-NV4 的脚本功能。请参见《WindO/I-NV4 用户手册》第 25 章“脚本”。

## 操作

输入打开时，执行对应于 S1 指定的脚本 ID 的脚本。脚本执行完成后，执行状态和执行时间存储在 D1 和 D1+1 中。要使用 SCRPT 指令，必须事先创建一个可在“脚本管理器”对话框中执行的脚本。

有关脚本功能概述的详情，请参见第 26-3 页上的“脚本功能概述”。

有关脚本编辑的详情，请参见第 26-5 页上的“脚本编程和管理”。

## 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	脚本 ID	—	—	—	—	—	—	X	—	1 ~ 255	—
D1 (目标 1)	执行结果	—	—	—	—	—	—	▲	—	—	—

▲ 不能将特殊内部继电器指定为 D1。

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

## 设置

### S1 (源 1): 脚本 ID

指定脚本 ID。可以指定常量或数据寄存器。

### D1 (目标 1): 执行结果

指定存储执行结果的数据寄存器。使用以指定数据寄存器开始的两个数据寄存器。脚本执行状态（脚本完成时的错误代码）存储在 D1 中。从脚本开始执行到完成所用的执行时间以 100μs 增量存储在 D1+1 中。

## 26: 脚本

### 执行结果

执行结果 (D1, D1+1) 存储了执行状态和执行时间。

### 执行状态

数字值	说明	错误原因
0	正常终止	—
1	运算错误	除以 0, 浮点格式错误
2	脚本 ID 错误	指定的脚本不存在
3	设备访问错误	指定的设备无效, 超出设备边界

如果出现除脚本 ID 以外的任何错误, 则会取消脚本处理, 并终止 SCRPT 指令的执行。如果出现脚本 ID 错误, 则会更新 D1 和 D1+1, 并终止 SCRPT 指令的执行。

### 执行时间

从指定脚本开始执行到完成所用的执行时间以 100 $\mu$ s 增量存储。例如, 如果花费 1.45ms 的时间执行指定的脚本, 则会存储 15 作为执行时间。如果花费 6,553.5ms 或更长的时间执行指定的脚本, 则会存储 65535 作为执行时间。SCRPT 指令执行时间受中断处理或执行过程中发生的其他处理影响。

如果出现除脚本 ID 以外的任何错误, 则会把从脚本开始执行到发生错误的脚本执行时间存储为执行时间。如果发生脚本 ID 错误, 则将 0 存储为执行时间。

### 脚本选择

要在 WindLDR 中为 S1 指定已注册的脚本 ID, 单击“SCRPT (脚本)”对话框上的“参考”按钮打开“脚本管理器”对话框。选择可执行的脚本 ID, 然后单击“脚本管理器”对话框上的“选择”按钮。所选的 ID 就输入到 S1 中。

## 脚本功能概述

脚本是使用条件分支、逻辑操作、运算操作以及函数将复杂处理过程编程为文本的功能。可在梯形图程序中执行已编程的脚本。

例如，可按照如下方法写入逻辑 AND 操作。



### ■ 脚本编程和管理

使用 WindLDR 中的“脚本编辑器”对话框编程脚本并通过“脚本管理器”对话框来管理。

#### 注释：

- 在脚本编辑器中，通过选择列表中的条件表达式、操作符和函数来编程脚本，还可检查它们的错误。也可将脚本导出为文本文件，以便可使用像 Notepad 这样的文本编辑器对它们进行编辑。如果将编辑的脚本保存为文本文件，也可将其导入到脚本编辑器中。有关详情，请参见第 26-8 页上的“脚本编辑器”。
- 在脚本管理器中，可在脚本编辑器中添加创建的脚本或删除脚本，以便按组管理脚本。有关详情，请参见第 26-7 页上的“脚本管理器”。

## 脚本数据类型

必须考虑在脚本中处理的详情（包括将处理的数据的最大值和最小值以及是否需要负数和实数）来配置在脚本中处理的数据范围的数据类型。

#### 注释：

- 在脚本编辑器中配置数据类型。有关如何配置数据类型的相关内容，请参见第 26-8 页上的“脚本编辑器”。
- 还可记述对中括号“{ }”内的处理进行数据类型指定的脚本。
- 还可记述以执行式单位进行数据类型指定的脚本。请参见第 26-12 页上的“脚本的记述方法”。

## 数据类型

可在脚本中处理以下五种数据类型。有关梯形图中数据类型的详情，请参见第 3-8 页上的“高级指令的数据类型”。

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 16 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 32 位）	L	32 位	2	-2,147,483,648 ~ 2,147,483,647
浮点型（F）	F	32 位	2	-3.402823×10 <sup>38</sup> ~ 3.402823×10 <sup>38</sup>

**注释：**如果数据类型不同，则某些函数无法使用。检查说明列表中的函数。请参见第 26-12 页上的“脚本的记述方法”。

## 26: 脚本

### 设备列表

本节给出了可在脚本编辑器中使用的设备以及设备说明。本节介绍了可用的设备及其可在“脚本编辑器”对话框中使用的说明。

**注释：**有效设备范围取决于控制器类型。在控制器的设备范围内指定设备。有关设备范围，请参见以下页面。

FT2J 型：第 2-1 页上的“FT2J 型”

FC6A 型：第 2-12 页上的“FC6A 型”

#### FT2J 型

	设备	脚本编辑器中的说明
位设备	I (输入)	I0 ~ I15
	I (扩展输入)	I20 ~ I27
	Q (输出)	Q0 ~ Q7
	Q (扩展输出)	Q10 ~ Q17
	M (内部继电器)	M0000 ~ M7997
	M (特殊内部继电器)	M8000 ~ M8177
	R (移位寄存器)	R000 ~ R127
	T (定时器触点)	T000 ~ T199
	C (计数器触点)	C000 ~ C199
	D (数据寄存器位)	D0000.0 ... D0000.15 ~ D3999.0 ... D3999.15
D (特殊数据寄存器的位)	DD8000.0 ... D8000.15 ~ D8199.0 ... D8199.15	
字设备	D (数据寄存器)	D0000 ~ D3999
	D (特殊数据寄存器)	D8000 ~ D8199
	TC (定时器当前值)	TC000 ~ TC199
	TP (定时器预置值)	TP000 ~ TP199
	CC (计数器当前值)	CC000 ~ CC199
	CP (计数器预置值)	CP000 ~ CP199

#### FC6A 型

	设备	脚本编辑器中的说明
位设备	I (输入)	I0 ~ I27
	I (扩展输入)	I30 ~ I10597
	Q (输出)	Q0 ~ Q17
	Q (扩展输出)	Q30 ~ Q10597
	M (内部继电器)	M0000 ~ M7997, M10000 ~ M21247
	M (特殊内部继电器)	M8000 ~ M9997
	R (移位寄存器)	R000 ~ R255
	T (定时器触点)	T000 ~ T1999
	C (计数器触点)	C000 ~ C511
	D (数据寄存器位)	D0000.0 ... D0000.15 ~ D7999.0 ... D7999.15 D10000.0 ... D10000.15 ~ D61999.0 ... D61999.15
D (特殊数据寄存器的位)	DD8000.0 ... D8000.15 ~ D8899.0 ... D8899.15	
D (非保持型数据寄存器*1 的位)	D70000.0 ... D70000.15 ~ D269999.0 ... D269999.15	
字设备	D (数据寄存器位)	D0000 ~ D7999, D10000 ~ D61999
	D (特殊数据寄存器的位)	D8000 ~ D8899
	D (非保持型数据寄存器*1 的位)	D70000 ~ D269999
	TC (定时器当前值)	TC000 ~ TC1999
	TP (定时器预置值)	TP000 ~ TP1999
	CC (计数器当前值)	CC000 ~ CC511
CP (计数器预置值)	CP000 ~ CP511	

\*1 仅 Plus CPU 模块

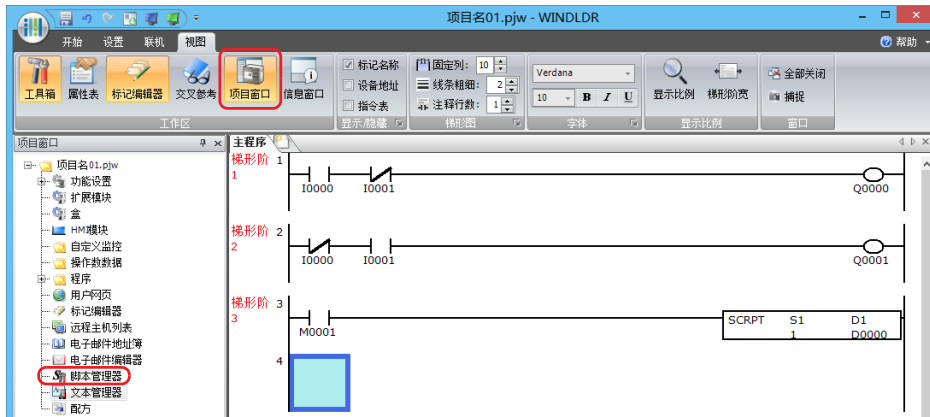
不允许同时使用位设备和字设备进行计算。位设备始终按位来处理，这些设备的值为 0 (OFF) 或 1 (ON)。

## 脚本编程和管理

### 脚本记录步骤

本节介绍创建和记录脚本的步骤。可通过在 SCRPT 指令中指定脚本 ID 来执行已注册脚本。

1. 在“项目窗口”中，双击“脚本管理器”。



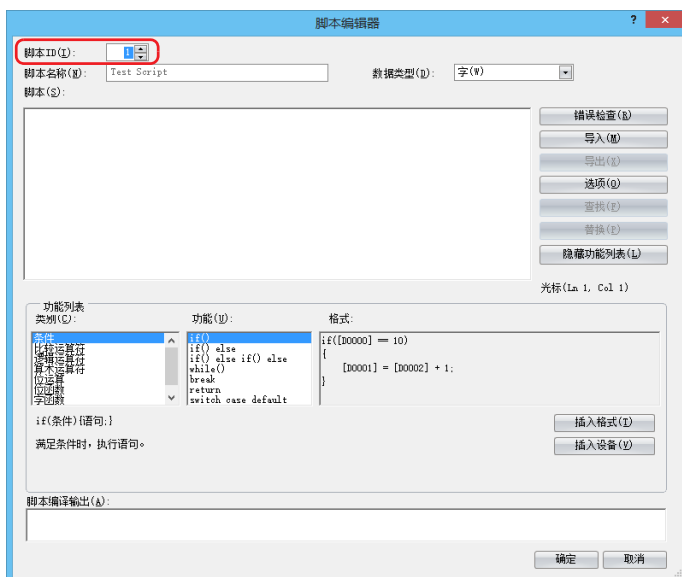
“脚本管理器”对话框打开。

2. 单击“添加”。



“脚本编辑器”对话框打开。

3. 指定“脚本 ID”。  
创建新脚本时，输入脚本 ID (1 到 255)。



## 26: 脚本

4. 输入“脚本名称”。  
脚本名称中可输入最多 40 个单字节字母数字字符。  
但是，无法使用以下半角字符。  
/\:\*?"<>|

5. 选择“数据类型”。

**注释：**使用所选的数据类型执行脚本。

6. 在“脚本”中，写入程序。

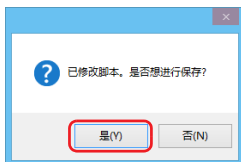
**注释：**要使用 WindLDR 提供的示例代码创建脚本，在“功能列表”下，选择“类别”和“功能”，然后单击“插入格式”。在“脚本”光标处插入“格式”中显示的示例代码。

7. 完成脚本创建后，单击“确定”。  
您将返回“脚本管理器”对话框，创建的脚本会显示在“脚本列表”中。
8. 单击“关闭”。



出现保存确认信息。

9. 单击“确定”。



保存脚本，并关闭“脚本管理器”对话框。

**注释：**

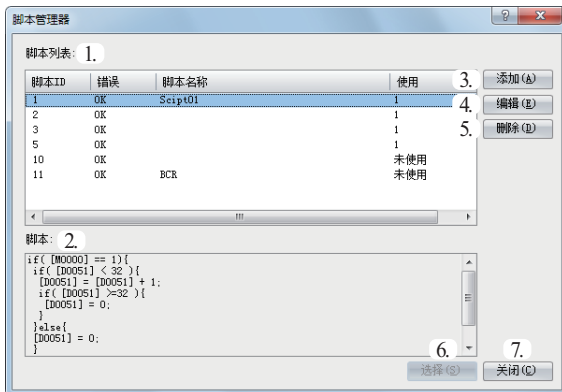
- 如果单击“否”，则不保存脚本，并关闭“脚本管理器”对话框。
- 也可通“SCRPT（脚本）”框打开脚本管理器。  
单击“请参考”按钮，脚本管理器。





## 脚本管理器

在“脚本管理器”对话框中，可添加在“脚本编辑器”对话框中创建的脚本或者删除已注册的脚本。



### 1. 脚本列表

给出已注册脚本的列表。

**脚本 ID:** 给出已注册脚本的脚本 ID（1 到 255）。

**错误:** 如果已注册脚本没有任何错误，显示“确定”。如果有些错误，则显示“NG”。

**脚本名称:** 显示已注册脚本的名称。

**使用:** 显示是否在此项目中使用了已注册脚本。如果未使用，则显示“未使用”。

**注释:** 当“使用”单元格不是“未使用”时，双击“使用”单元格，“信息窗口”将列出使用 SCRPT 指令的位置。双击“信息窗口”中显示的位置，光标将移动到使用 SCRPT 指令的梯形图行。

### 2. 脚本

显示在“脚本列表”中所选脚本的内容。

### 3. 添加

单击此按钮可显示“脚本编辑器”对话框，以便创建和添加脚本。

有关详情，请参见第 26-8 页上的“脚本编辑器”。

### 4. 编辑

单击此按钮可打开“脚本编辑器”对话框，以便编辑在“脚本列表”中选择的脚本。

有关详情，请参见第 26-8 页上的“脚本编辑器”。

### 5. 删除

单击此按钮可删除在“脚本列表”中选择的脚本。

### 6. 选择

仅在通过 SCRPT 指令对话框打开脚本管理器时使用。

在脚本一览中选择脚本，并单击该按钮，即可在对话框的 S1 中输入所选的脚本 ID。

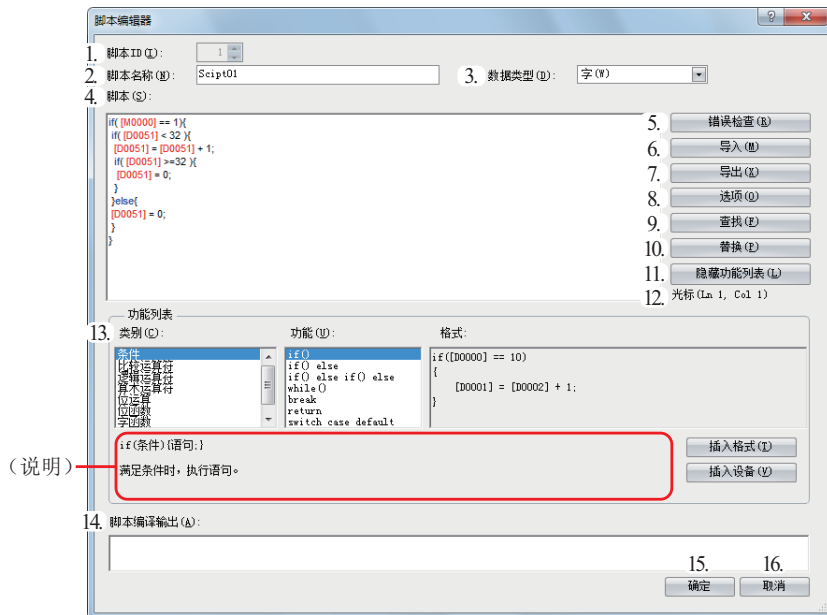
### 7. 关闭

单击此按钮关闭脚本管理器。

**注释:** 如果在修改脚本列表后单击“关闭”，则出现保存确认信息。单击确认信息上的“是”可保存更改。单击“否”可放弃更改，并关闭脚本管理器。

### 脚本编辑器

脚本编辑器是用于创建新脚本以及编辑在脚本管理器中选择的脚本的工具。



#### 1. 脚本 ID

创建新脚本时，输入脚本 ID（1 到 255）。编辑现有脚本时，这会显示已设置的脚本 ID。

#### 2. 脚本名称

输入脚本名称。脚本名称中最多可输入 40 个字符。  
但是，无法使用以下半角字符。  
/\: \* ? " < > |

#### 3. 数据类型

选择要在脚本中处理的数据类型。有关数据类型的详情，请参见第 26-3 页上的“脚本数据类型”。

#### 4. 脚本

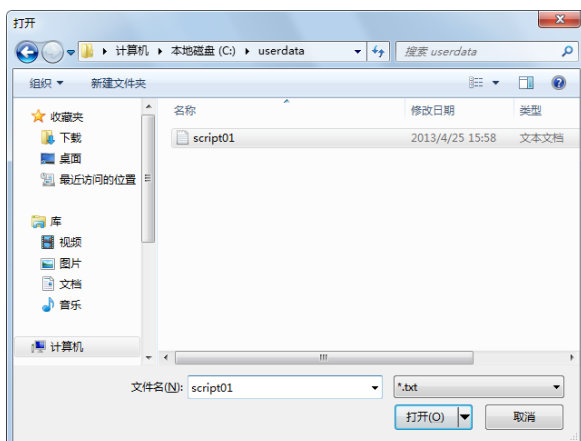
输入脚本。  
对某个脚本的限制为每行最多 240 个字符，每个脚本最多 1,024 行。

#### 5. 错误检查

单击此按钮可检查当前脚本中的错误。

#### 6. 导入

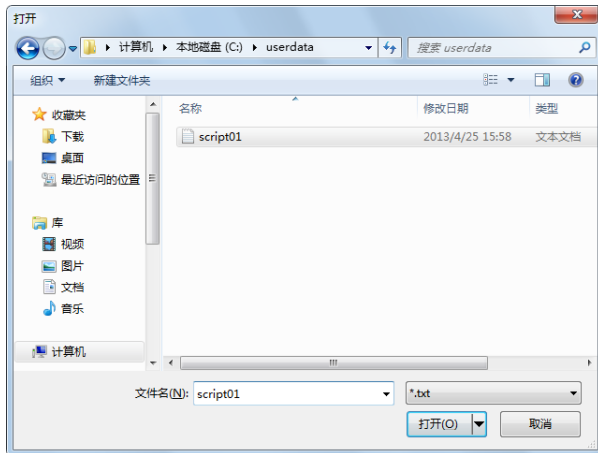
单击此按钮显示“打开”对话框，以便您导入脚本程序。  
如果选择已保存（导出）为文本文件 (\*.txt) 的脚本并单击“打开”，则会在当前脚本的当前光标处插入所选的脚本。



## 7. 导出

单击此按钮显示“打开”对话框以便您导出脚本程序。

如果选择要保存脚本的位置，请输入文件名的位置，然后单击“保存”，则当前脚本会被保存为文本文件 (\*.txt)。可使用“导入”插入保存的脚本。



## 8. 选项

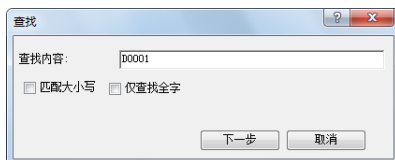
单击此按钮可显示“选项”对话框。

在“选项”对话框中，可配置要在“脚本”文本框中使用的文本的字体、颜色、制表符缩进以及其他设置。有关详情，请参见第 26-11 页上的““选项对”对话框”。

## 9. 查找

单击此按钮可显示“查找”对话框。

您可搜索在脚本中输入的文本。

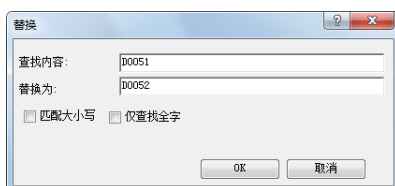


**注释：**如果您首先在“脚本”中选择了范围，然后单击“查找”，则仅可在所选的范围中搜索。

## 10. 替换

单击此按钮可显示“替换”对话框。

在“查找内容”输入要在脚本中搜索的文本，然后在“替换为”中输入替换文本。



**注释：**

- 替换设备地址时这是一个有效的工具。
- 如果您首先在“脚本”中选择了范围，然后单击“替换”，则仅可在所选的范围中搜索并替换文本。

## 11. 显示 / 隐藏功能列表

此按钮可显示和隐藏“函数列表”和“输出”。

**注释：**单击并拖动脚本编辑器的右下角，可更改用于编辑脚本的对话框的大小。通过隐藏“函数列表”和“输出”，可增大脚本编辑区域（文本框）的大小，更便于编辑脚本。

## 12. 光标

按照行数和列数显示“脚本”文本框中当前光标的位置。

### 13. 功能列表

类别：	显示函数类表列表。
功能：	显示所选类别的函数列表。
格式：	显示所选函数的编程示例。
说明：	显示所选函数的说明。
“插入格式”：	单击此按钮可在光标位置插入“格式”中显示的内容。
“插入设备”：	单击此按钮可打开“标记编辑器”对话框。 指定设备地址，单击“确定”以在光标位置插入指定的设备地址。

### 14. 脚本编译输出

如果在错误检查中脚本存在错误，则显示错误详情。

如果双击“脚本编译输出”中显示的注释，对应于错误的部分会在“脚本”中突出显示。

**注释：**根据错误的不同，错误存在的行可不同于“输出”中显示的行，或者可显示多个错误。

### 15. 确定

单击此按钮可在当前脚本中执行错误检查，然后在保存脚本后返回到脚本管理器。

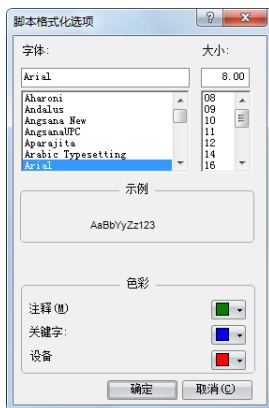
**注释：**如果当前脚本中存在任何错误，则出现保存确认信息。可保存包含错误的脚本。

### 16. 取消

单击此按钮可返回脚本管理器而不保存当前的脚本。

## “选项对”对话框

此对话框允许您指定在脚本编辑器“脚本”文本框中使用的“字体”、“大小”、“制表符缩进”和“色彩”。



### ■ 字体

输入或选择“脚本”中显示的文本的字体名称。

### ■ 大小

输入或选择“脚本”中显示的文本的大小（像素）。

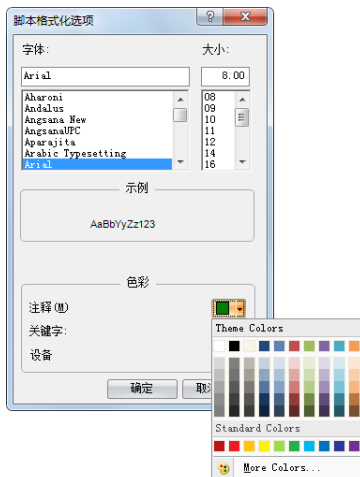
### ■ 示例

显示“脚本”文本框中显示的文本示例，字符字体和字符大小由“字体”和“大小”指定。

### ■ 色彩

显示“注释”、“关键字”和“设备”的颜色。

单击彩色按钮可显示“颜色”对话框。



非注释、关键字或设备的字符都显示为黑色。

## 脚本的记述方法

### 表述一览

以下说明控制语句、运算符、函数、数据类型指定等的表述及动作。

注解除外，命令必须用半角输入。有关具体的记述示例，请参见第 26-19 页上的“脚本的记述例”。

### 控制语句

条件式在这里记为 (条件式)、(条件式1)、(条件式2)。

执行式记为 (执行式)、(执行式1)、(执行式2)...

#### ■ 条件分支

	表述	说明
	<b>if</b> (条件式) { (执行式); }	当条件式成立时，执行执行式。
if else else if	<b>if</b> (条件式) { (执行式1); } <b>else</b> { (执行式2); }	当条件式成立时，执行执行式 1。 当条件不成立时，执行执行式 2。
	<b>if</b> (条件式1) { (执行式1); } <b>else if</b> (条件式2) { (执行式2); } <b>else</b> { (执行式3); }	当条件式 1 成立时，执行执行式 1。 当条件式 1 不成立时，判定条件式 2，如果条件式 2 成立时，执行执行式 2。 当条件式 2 也不成立时，执行执行式 3。
switch case default	<b>switch</b> (条件式) { <b>case</b> 定数 1: (执行式1); <b>break</b> ; <b>case</b> 定数 2: (执行式2); <b>break</b> ; <b>default</b> : (执行式3); <b>break</b> ; }	当条件式的值与常数 1 一致时，执行执行式 1。 当条件式的值与常数 2 一致时，执行执行式 2。 当条件式的值与常数 1、常数 2 都不一致时，执行执行式 3。

#### ■ 重复

	表述	说明
while	<b>while</b> (条件式) { (执行式); }	在条件式成立期间，反复执行执行式。 条件式总是成立时，变成无限循环，所以请不要把固定值或值不变化的设备设置在条件式中。

## ■ 中断和结束

	表述	说明
break	<pre>while ((条件式1) {   if ((条件式2)   {     (执行式1);     break;   }   (执行式2); } (执行式3);</pre>	<p>在条件式 1 成立期间，如下处理。          在条件式 2 不成立期间，继续执行执行式 2。          条件式 2 一成立，就中断循环（不执行执行式 2），执行执行式 3。</p>
break	<pre>switch ((条件式) {   case 定数 1:     (执行式1);     break;   case 定数 2:     (执行式2);     break; } (执行式3);</pre>	<p>当条件式等于常数 1 时，执行过执行式 1 后，由 break 中断常数 2 的判定，处理转移到执行式 3。</p>
return	return;	将会退出脚本，然后在用户程序中执行下一个指令。

## 运算符

在此，设备、常数、暂存设备记为  $\boxed{a}$ 、 $\boxed{b}$ ，算式记为  $(式)$ 、 $(式1)$ 、 $(式2)$ 。

## ■ 相关运算符

运算符	表述	说明
==	$\boxed{a} == \boxed{b}$	比较 $\boxed{a}$ 比较 $\boxed{b}$ 是否相等。
!=	$\boxed{a} != \boxed{b}$	比较 $\boxed{a}$ 比较 $\boxed{b}$ 是否不相等。
<	$\boxed{a} < \boxed{b}$	比较 $\boxed{a}$ 是否小于 $\boxed{b}$ 。
<=	$\boxed{a} <= \boxed{b}$	比较 $\boxed{a}$ 是否小于等于 $\boxed{b}$ 。
>	$\boxed{a} > \boxed{b}$	比较 $\boxed{a}$ 是否大于 $\boxed{b}$ 。
>=	$\boxed{a} >= \boxed{b}$	比较 $\boxed{a}$ 是否大于等于 $\boxed{b}$ 。

## ■ 逻辑运算符

运算符	表述	说明
&&	$((式1) \ \&\& \ (式2))$	运算 $(式1)$ 和 $(式2)$ 的逻辑积 (AND)。
	$((式1) \    \ (式2))$	运算 $(式1)$ 和 $(式2)$ 的逻辑和 (OR)。
!	$!((式))$	反转 $(式1)$ 的逻辑。

## ■ 算术运算符

运算符	表述	说明
+	$\boxed{a} + \boxed{b}$	$\boxed{a}$ 和 $\boxed{b}$ 相加。
-	$\boxed{a} - \boxed{b}$	从 $\boxed{a}$ 减去 $\boxed{b}$ 。
*	$\boxed{a} * \boxed{b}$	$\boxed{a}$ 和 $\boxed{b}$ 相乘。
/	$\boxed{a} / \boxed{b}$	把 $\boxed{a}$ 用 $\boxed{b}$ 除。
%	$\boxed{a} \% \boxed{b}$	求把 $\boxed{a}$ 用 $\boxed{b}$ 除后的余数。
=	$\boxed{a} = \boxed{b}$	用 $\boxed{b}$ 替换 $\boxed{a}$ 。

## ■ 位运算符

运算符	表述	说明
&	$\boxed{a} \ \& \ \boxed{b}$	运算 $\boxed{a}$ 和 $\boxed{b}$ 的各位的逻辑积 (AND)。 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。无法使用 F (浮点)。
	$\boxed{a} \ \mid \ \boxed{b}$	运算 $\boxed{a}$ 和 $\boxed{b}$ 的各位的逻辑和 (OR)。 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。无法使用 F (浮点)。
^	$\boxed{a} \ \wedge \ \boxed{b}$	运算 $\boxed{a}$ 和 $\boxed{b}$ 的各位的排他性逻辑和 (XOR)。 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。无法使用 F (浮点)。
~	$\sim \boxed{a}$	$\boxed{a}$ 反转的各位。 如果是字设备和固定值, 0 变为 65,535、65,535 变为 0。 如果是位设备, 0 变为 1、1 变为 0。 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。无法使用 F (浮点)。
<<	$\boxed{a} \ \ll \ \boxed{b}$	把 $\boxed{a}$ 的各位向左移动 $\boxed{b}$ 位。 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。无法使用 F (浮点)。
>>	$\boxed{a} \ \gg \ \boxed{b}$	把 $\boxed{a}$ 的各位向右移动 $\boxed{b}$ 位。 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。无法使用 F (浮点)。

## 函数

在此, 设备、常数、暂存设备记为  $\boxed{a}$ 、 $\boxed{b}$ 、 $\boxed{c}$ 、 $\boxed{d}$ ...

## ■ 位函数

函数	表述	说明
置位	SET ( $\boxed{a}$ );	将位设备 $\boxed{a}$ 1。 变成与 $\boxed{a} = 1$ ; 相同的结果。
复位	RST ( $\boxed{a}$ );	将位设备 $\boxed{a}$ 0。 变成与 $\boxed{a} = 0$ ; 相同的结果。
位反转	REV ( $\boxed{a}$ );	反转位设备 $\boxed{a}$ 的 1 和 0。 变成与 $\boxed{a} = \sim \boxed{a}$ ; 相同的结果。

## ■ 字函数

## 算术运算

函数	表述	说明
最大值	MAX ( $\boxed{a}$ , $\boxed{b}$ , $\boxed{c}$ )	返回 $\boxed{a}$ 、 $\boxed{b}$ 、 $\boxed{c}$ 中的最大值。 • 可在所有数据类型中使用。 • 可以记述 15 个以内的参数。
最小值	MIN ( $\boxed{a}$ , $\boxed{b}$ , $\boxed{c}$ )	返回 $\boxed{a}$ 、 $\boxed{b}$ 、 $\boxed{c}$ 中的最小值。 • 可在所有数据类型中使用。 • 可以记述 15 个以内的参数。
指数函数	EXP ( $\boxed{a}$ )	返回 $\boxed{a}$ 的指数函数。 • 只能够在数据类型 F (浮点) 中使用。
自然对数 (底: e)	LOGE ( $\boxed{a}$ )	返回 $\boxed{a}$ 的自然对数 (底为 e)。 • 只能够在数据类型 F (浮点) 中使用。 • 参数请设置大于 0 的值。
常用对数 (底: 10)	LOG10 ( $\boxed{a}$ )	返回 $\boxed{a}$ 的常用对数 (底为 10)。 • 只能够在数据类型 F (浮点) 中使用。 • 参数请设置大于 0 的值。
乘方	POW ( $\boxed{a}$ , $\boxed{b}$ )	返回 $\boxed{a}$ 的 $\boxed{b}$ 乘方。 • 只能够在数据类型 F (浮点) 中使用。



函数	表述	说明
平方根	ROOT ( [a] )	返回 [a] 的平方根。 • 只能在数据类型 F (浮点) 中使用。
正弦	SIN ( [a] )	返回 [a] 的正弦 (-1 ~ +1)。 参数 [a] 要指定表示角度的任意数字 (单位为弧度)。 • 只能在数据类型 F (浮点) 中使用。
余弦	COS ( [a] )	返回 [a] 的余弦 (-1 ~ +1)。 参数 [a] 要指定表示角度的任意数字 (单位为弧度)。 • 只能在数据类型 F (浮点) 中使用。
正切	TAN ( [a] )	返回 [a] 的正切 (-1 ~ +1)。 参数 [a] 要指定表示角度的任意数字 (单位为弧度)。 • 只能在数据类型 F (浮点) 中使用。
反正弦	ASIN ( [a] )	把 [a] 的反正弦 (-1 ~ +1) 用弧度值 ( $-\pi/2 \sim +\pi/2$ ) 返回。 参数 [a] 指定任意数字。 • 只能在数据类型 F (浮点) 中使用。
反余弦	ACOS ( [a] )	把 [a] 的反余弦 (-1 ~ +1) 用弧度值 ( $0 \sim +\pi$ ) 返回。 参数 [a] 指定任意数字。 • 只能在数据类型 F (浮点) 中使用。
反正切	ATAN ( [a] ) ;	把 [a] 的反正切 (-1 ~ +1) 用弧度值 ( $-\pi/2 \sim +\pi/2$ ) 返回。 参数 [a] 指定任意数字。 • 只能在数据类型 F (浮点) 中使用。
角度 ↓ 弧度	RAD ( [a] ) ;	把 [a] 的值从度 (°) 转换成弧度返回。 • 只能在数据类型 F (浮点) 中使用。
弧度 ↓ 角度	DEG ( [a] ) ;	把 [a] 的值从弧度转换成度 (°) 返回。 • 只能在数据类型 F (浮点) 中使用。

## 数据类型转换

函数	表述	说明
BCD ↓ 二进制	BCD2BIN ( [a] )	把 [a] 的 BCD 值以二进制值返回。 • 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。 无法使用 F (浮点)。
二进制 ↓ BCD	BIN2BCD ( [a] )	把 [a] 的二进制值以 BCD 值返回。 • 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。 无法使用 F (浮点)。
浮点 ↓ 二进制	FLOAT2BIN ( [a] )	把 [a] 的浮点值以二进制值返回。 小数点以下的值舍去。 • 可使用的数据类型为 D (双字)、L (长整)。
二进制 ↓ 浮点	BIN2FLOAT ( [a] )	把 [a] 的二进制值以浮点值返回。 • 可在数据类型 D (双字) 和 L (长整数) 中使用。
十进制 ↓ 字符串	DEC2ASCII ( [a], [b] )	把十进制的值 [b] 转换成字符串, 把 [a] 作为开头设备按照顺序储存。 • 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。 无法使用 F (浮点)。
字符串 ↓ 十进制	ASCII2DEC ( [a] )	把字符串 [a] 以十进制的值返回。 • 可使用的数据类型为 W (字)、I (整数)、D (双字)、L (长整)。 无法使用 F (浮点)。

数据的比较和复制

函数	表述	说明
数据比较	MEMCMP ( [a], [b], [c] )	<p>[a]: 比较对象 1 的开头设备                      [b]: 比较对象 2 的开头设备                      [c]: 比较的范围 (字数)                      比较从 [a] 起 [c] 字符的范围和 [b] 起 [c] 字符范围的设备的值。                      设备的值全部一致则返回 1、即使有 1 点不一致也返回 0。</p> <ul style="list-style-type: none"> <li>以字符为单位比较设置的范围, 返回结果。</li> <li>最大可比较 64 字符。</li> </ul>
数据复制	MEMCPY ( [a], [b], [c] )	<p>[a]: 复制目标的起始设备                      [b]: 复制源的起始设备                      [c]: 复制的长度范围 (字数: 1 ~ 64)                      把以 [b] 为起始位置开始的 [c] 个字长度范围内的设备地址中存储的值, 分别复制到以 [a] 为起始位置开始的 [c] 个字长度范围内的设备中去。</p> <ul style="list-style-type: none"> <li>以字为单位复制设置的范围。</li> </ul>
位设备 (1 个字长) ↓ 位设备 (1 个字长)	BITS2BITS ( [a], [b] );	<p>[a]: 复制目标的起始设备 (位设备)                      [b]: 复制源的起始设备 (位设备)                      从 [b] 复制一个字长的数据到 [a]。</p> <ul style="list-style-type: none"> <li>从起始位设备按一个字来处理 16 位。</li> </ul>
位设备 (1 个字长) ↓ 字设备	BITS2WORD ( [a], [b] );	<p>[a]: 复制目标的起始设备 (字设备)                      [b]: 复制源的起始设备 (位设备)                      从 [b] 复制一个字长的数据到 [a]。</p> <ul style="list-style-type: none"> <li>从起始位设备按一个字来处理 16 位。</li> </ul>
字设备 ↓ 位设备 (1 个字长)	WORD2BITS ( [a], [b] );	<p>[a]: 复制目标的起始设备 (位设备)                      [b]: 复制源的起始设备 (字设备)                      从 [b] 复制一个字长的数据到 [a]。</p> <ul style="list-style-type: none"> <li>从起始位设备按一个字来处理 16 位。</li> </ul>

Offset

函数	表述	说明
间接指定	OFFSET ( [a], [b] )	<p>[a]: 基准的设备                      [b]: 储存间接值 (0 ~ 32767) 的设备                      指定从 [a] 起 [b] 字符前的设备。                      间接读取                      在代入式的右边记述 OFFSET 函数。                      表述例: [c] = OFFSET ( [a], [b] )                      动作: 把从 [a] 起 [b] 字符之前的设备的值储存到 [c]。</p> <p>间接写入                      在代入式的左边记述 OFFSET 函数。                      表述例: OFFSET ( [a], [b] ) = [c]                      动作: 把 [c] 的值储存到从 [a] 起 [b] 字符之前的设备上。</p> <ul style="list-style-type: none"> <li>采用间接值时, 请储存符合数据类型的值。                          例如, 脚本的数据类型是 I (整数) 时, 请将 I (整数) 的值储存到偏差值的设备上。</li> </ul>

### 字符串操作

处理字符串时，将终端字符 NULL (00h) 设为字符串的末尾。此外，终端字符 NULL 不包含在字符串的长度中。

函数	表述	说明
字符串复制	STRCUT ([a], [b], [c], [d])	[a]: 复制对象的开头设备 [b]: 存储复制源字符串的起始设备 [c]: 复制的开始位置 (0 至 127) [d]: 要复制的字符数 (1 至 128) 从 [b] 起始的字符串的 [c] 字节位置开始，将 [d] 字符的字符串以 [a] 为起始存储到 [d] 字符中。
字符数计数	STRLEN ([a])	返回 [a] 起始的字符串的字符数。
字符串连接	STRCAT ([a], [b]);	将 [b] 起始的字符串连接到 [a] 起始的字符串，并以 [a] 为起始进行存储。
字符串搜索	STRSTR ([a], [b]);	从 [a] 起始的字符串开始，搜索 [b] 起始的字符串，并返回到发现位置 (起始的字符数 -1)。 • 要搜索的字符串最大字符数为 128 个半角字符。

### ■ 数据类型指定

可记述对中括号“{ }”内的处理进行数据类型指定的脚本。

执行式记为(执行式)、(执行式1)、(执行式2)、(执行式3)...

	表述	说明
asword	asword { (执行式1); } (执行式2);	以数据类型 W (字) 执行执行式 1。 以脚本编辑器的“数据类型”中设置的数据类型执行执行式 2。
asinterger	asinterger { (执行式1); } (执行式2);	以数据类型 I (整数) 执行执行式 1。 以脚本编辑器的“数据类型”中设置的数据类型执行执行式 2。
asdobleword	asdobleword { (执行式1); } (执行式2);	以数据类型 D (双字) 执行执行式 1。 以脚本编辑器的“数据类型”中设置的数据类型执行执行式 2。
aslong	aslong { (执行式1); } (执行式2);	以数据类型 L (长整) 执行执行式 1。 以脚本编辑器的“数据类型”中设置的数据类型执行执行式 2。
asfloat	asfloat { (执行式1); } (执行式2);	以数据类型 F (浮点) 执行执行式 1。 以脚本编辑器的“数据类型”中设置的数据类型执行执行式 2。

**注释:** 数据类型指定的表述还可省略部分关键字进行记述。

表述	省略部分关键字后的表述
asword	asw
asinterger	asi
asdobleword	asd
aslong	asl
asfloat	asf

无论使用何种表述，数据类型指定的动作均相同。

### 其它

以下说明常数、设备、暂存设备、注释的记述。

#### ■ 常数

常数中可以记述十进制数和十六进制数。

##### 十进制的记述例

1234	直接记述数值。
-1234	负数在开头记述“-”（负的意思）。
12.34	如果是实数（浮点），还可以记述小数。 整数和小数之间记述“.”（小数点）。

十六进制有二种记述方法。

##### 十六进制的记述例

0x12AB	在值的开头加“0”（零）和“x”（小写）。
12ABh	在值的末尾加“h”。

#### ■ 设备

设备的设备标志和地址用“[]”和“[]”框起来记述。

##### 设备的记述

[ 设备符号和地址 ]	（设备符号和地址之间不需要有空格。）
-------------	--------------------

##### 记述例

[D0100]
---------

#### ■ 暂存设备

暂存设备只能在脚本中使用。可以储存值，作为变数使用。

在设备标志“@”后，记述地址（1 ~ 32）。

##### 暂存设备的记述

@address	（设备标志“@”和地址之间不要空格。）
----------	---------------------

##### 记述例

@2	暂存设备 2 号
----	----------

##### 注释:

- 开始执行脚本时，所有的暂存设备的值变为“0”。
- 数据类型指定的中括号“{}”内的处理无法使用暂存设备。

#### ■ 备注

这是在脚本中记述的注释。在行头记述“//”，该行就变成注释。“//”用半角来记述。“//”之后也可以用全角文字记述。

##### 注释的记述

// 任意的注释
----------

##### 记述例

// 在处理 A 的运算数据 [D0100] 上保存初期 [D0100] = 1234; : :	← 该行不执行。
-----------------------------------------------------------	----------

##### 注释:

- 记述注释时写清楚动作的内容，脚本编辑器改变了时或长时间后再编辑时，会有助于理解脚本的动作。
- 执行脚本时注释被忽略（不执行），所以不用考虑运行时间，可任意记述。

## 脚本的记述例

以下就控制语句、运算和函数等各脚本记述及其运行内容分别进行说明。

### 1. 控制语句

#### 例 1.1 条件分支

脚本

```
if ([D0100])
{
    [D0102] = 100;
}
```

运行内容

如果 D0100 的值非 0，100 存储在 D0102 中。

#### 例 1.2 条件分支

脚本

```
if ([D0100])
{
    [D0102] = [D0103] + [D0104] + [D0105];
}
```

运行内容

如果 D0100 的值非于 0，则 D0103、D0104、D0105 相加的值存储在 D0102。

#### 例 1.3 条件分支

脚本

```
if (0 != [D0100])
{
    if (0 != [D0102])
    {
        [D0103] = 0x1234;
    }
}
```

运行内容

如果 D0100 的值非 0，且 D0102 的值也非 0，1234h 存储在 D0103 中。

如果 D0100 的值非 0，且 D0102 的值为 0 的话，不执行任何指令。

如果 D0100 的值为 0，不执行任何指令，与 D0102 的值无关。

#### 例 1.4 条件分支

脚本

```
if ((0 != [D0100]) || (0 != [D0102]))
{
    [D0103] = 100;
}
else
{
    [D0104] = [D0105] + 100;
}
```

运行内容

如果 D0100 的值与 D0102 的值其中之一非 0，100 存储在 D0103 中。

如果 D0100 的值与 D0102 的值二者均为 0，D0105 与 100 相加的值存储在 D0104 中。

## 26: 脚本

### 例 1.5 条件分支

#### 脚本

```
if ([D0100] == 0)
{
    [D0102] = 0x1234;
}
else if ([D0100] == 1)
{
    [D0102] = 0x5678;
}
else
{
    [D0102] = 0x9999;
}
```

#### 运行内容

如果 D0100 的值为 0，1234h 存储在 D0102 中。

如果 D0100 的值为 1，5678h 存储在 D0102 中。

如果 D0100 的值既非 0 也非 1 时，9999h 存储在 D0102 中。

### 例 1.6 条件分支

#### 脚本

```
if ([D0100])
{
    if ([D0102])
    {
        if ([D0103])
        {
            [D0104] = 100;
        }
        else
        {
            [D0104] = 200;
        }
    }
}
```

#### 运行内容

如果 D0100、D0102、D0103 的值均非 0，100 存储在 D0104 中。

如果 D0100 的值与 D0102 的值非 0，而 D0103 的值为 0，200 存储在 D0104 中。

如果 D0100 的值与 D0102 的值其中之一为 0，不执行任何指令，与 D0103 的值无关。

### 例 1.7 重复

#### 脚本

```
[D0100] = 10;
[D0102] = 10;

while (0 < [D0100])
{
    [D0102] = [D0102] + 1;
    [D0100] = [D0100] - 1;
}
```

#### 运行内容

如果 D0100 的值大于 0，重复执行 D0102 的值上加 1、从 D0100 的值中减 1 的操作。

在上述的脚本示例中，若重复执行 10 次 while 语句，则 D0100 的值变为 0，结束 while 语句。

同时，该脚本执行后，D0100 的值变为 0，D0102 的值变为 20。

**例 1.8 重复****脚本**

```
[D0100] = 0;
[D0102] = 3;
[D0103] = 5;

while ([D0100] == 0)
{
    [D0102] = [D0102] + 1;

    if ([D0103] == [D0102])
    {
        SET([M0000]);
        break;
    }
}
```

**运行内容**

在 D0100 的值为 0 期间，重复执行 while 语句。

在 while 语句中，如果 D0102 的值与 D0103 的值已经一致，则中止 while 语句，[M0000] 切换为 1 后，退出 while 语句。在上述脚本示例中，若重复执行 2 次 while 语句，则 D0102 的值与 D0103 的值一致，M0000 切换为 1 后，退出 while 语句的循环。同时，上述示例执行后，D0100 的值变为 0，D0102 的值变为 5，D0103 的值变为 5，M0000 变为 1。

**例 1.9 使用重复（while 语句）进行间接写入与间接读取****脚本**

```
// 将 D0010 ~ D0019 传送到 D0100 ~ D0109 中

// 对间接值初始化
[D0000] = 0;

// 10 次循环
while ([D0000] < 10)
{
    // 利用间接指定，传送 1 字
    OFFSET([D0100], [D0000]) = OFFSET([D0010],
[D0000]);
    // 对间接值进行递增
    [D0000] = [D0000] + 1;
}
```

**运行内容**

将 D0010 到 D0019 的值，分别存储在 D0100 至 D0109 的脚本。

其动作如下所示。

首先将间接值 D0000 的值初始化，变为 0。

重复（循环）第 1 次：因 D0000 的值为 0，条件“[D0000] < 10”成立，执行 while 内的执行语句。

- 将 D0010 起始第 0 字的 D0010 值，存储到 D0100 起始第 0 字的 D0100 中。
- 间接值 D0000 的值与 1 相加，D0000 的值变为 1。

重复（循环）第 2 次：因 D0000 的值为 1，条件“[D0000] < 10”成立，执行 while 内的执行语句。

- 将 D0010 起始第 1 字的 D0011 值存储到 D0100 起始第 1 字的 D0101 中。
- 间接值 D0000 的值与 1 相加，D0000 的值变为 2。

：

（同样，第 3 ~ 9 次也重复执行）

：

重复（循环）第 10 次：因 D0000 的值为 9，条件“[D0000] < 10”成立，执行 while 内的执行语句。

- 将 D0010 起始第 9 字的 D0019 值存储到 D0100 起始第 9 字的 D0109 中。
- 间接值 D0000 的值与 1 相加，D0000 的值变为 10。

因 D0000 的值变为 10，条件“[D0000] < 10”不成立，退出 while 的循环。

执行后，从 D0100 到 D0109 的值，分别成为从 D0010 到 D0019 的值。

## 26: 脚本

### 例 1.10 使用 while 语句的十进制→八进制转换

#### 脚本

```
// 将 10 进制的值转换为 8 进制的处理
// - 例如, 将 10 (十进制) 转换为 12 (八进制), 将 16 (十进制) 转换为 20 (八进制)
// - 最大转换为 4 位八进制数

@1 = 0;           // while 的计数器
@2 = [D0100];    // 获得源数据
@3 = 1;          // 10 进制的基数
@4 = 0;          // 运算结果

// 重复 4 次
while (@1 < 4)
{
    // 从源数据中抽取八进制的第 1 位。将存储运算的中间结果存储在 @10 中。
    @10 = @2 % 8;
    // 将抽取结果转换为十进制数, 并将其与结果相加
    @4 = @4 + (@10 * @3);

    // 将十进制的基数增加 1 位
    @3 = @3 * 10;
    // 将源数据减少 1 位
    @2 = @2 / 8;
    // 若 @2 变为 0, 则退出 while 语句
    if (0 == @2)
    {
        break;
    }

    // while 计数器增加 1
    @1 = @1 + 1;
}

// 将运算结果存储到 LDR200 中
[D0200] = @4;
```

#### 运行内容

以上是使用 while 语句将十进制的值转换为八进制的示例。  
将十进制的源数据除以 8, 然后逐位转换为八进制数, 通过用 while 语句重复进行处理, 最多可实现 4 位的转换。  
十进制转换源的值已存储在 D0100 中, 执行脚本后, 将转换后的八进制值存储到 D0200 中。

### 例 1.11 使用 switch 的条件分支

#### 脚本

```
switch ([D0100])
{
    case 10:
        [D0200] = 0x1234;
        break;
    case 999:
        [D0200] = 0x5678;
        SET([D0000.01]);
        break;
}
```

#### 运行内容

如果 D0100 的值为 10, 1234h 存储在 D0200 中。  
如果 D0100 的值为 999, 5678h 存储在 D0200 中, 并将 D0000.01 切换为 1。  
如果 D0100 的值既非 10 也非 999 时, 不执行任何指令。



**例 1.12** 使用 default 语句的 switch 条件分支**脚本**

```
switch ([D0100])
{
    case 0:
        [D0102] = 0x1234;
        break;
    case 1:
        [D0102] = 0x5678;
        break;
    default:
        [D0102] = 0x9999;
        break;
}
```

**运行内容**

如果 D0100 的值为 0，1234h 存储在 D0102 中。

如果 D0100 的值为 1，5678h 存储在 D0102 中。

如果 D0100 的值既非 0 也非 1 时，9999h 存储在 D0102 中。

**例 1.13** 使用 return 语句的脚本结束**脚本**

```
if (0x1234 == [D0100])
{
    [D0102] = 0x5678;
    return;
}
[D0103] = 0;
```

**运行内容**

如果 D0100 的值非 1234h，则将 0 存储在 D0103 中。

如果 D0100 的值为 1234h，则 5678h 存储在 D0102 中，结束脚本。

return 语句并非与 break 语句一样用于退出函数，而是用于结束脚本执行。

**例 1.14** 使用 break 语句从循环中退出**脚本**

```
[D0100] = 0;
[D0102] = 3;
[D0103] = 5;

while ([D0100] == 0)
{
    [D0102] = [D0102] + 1;

    if ([D0102] == [D0103])
    {
        SET([D0000.01]);
        break;
    }
}
```

**运行内容**

在 D0100 的值为 0 期间，D0102 与 D0103 达到一致之前，会一直重复 while 语句。

在 while 语句中，如果 D0102 的值与 D0103 的值已经一致，则中止 while 语句，退出 while 语句。

在上述脚本示例中，若重复执行 2 次 while 语句，则 D0102 的值与 D0103 的值一致，D0000.01 切换为 1 后，while 语句结束。同时，上述示例执行后，D0100 的值变为 0，D0102 的值变为 5，D0103 的值变为 5，D0000.01 的值变为 1。

### 2. 比较运算

#### 例 2.1 等于

##### 脚本

```
if ([D0100] == [D0102])
{
    [D0103] = 0x100;
}
```

##### 运行内容

如果 D0100 的值等于 D0102 的值，100h 存储在 D0103 中。

#### 例 2.2 不等于

##### 脚本

```
if ([D0100] != [D0102])
{
    [D0103] = 0x100;
}
```

##### 运行内容

如果 D0100 的值不等于 D0102 的值，100h 存储在 D0103 中。

#### 例 2.3 小于

##### 脚本

```
if ([D0100] < [D0102])
{
    [D0103] = 0x100;
}
```

##### 运行内容

如果 D0100 的值小于 D0102 的值，100h 存储在 D0103 中。

#### 例 2.4 等于或小于

##### 脚本

```
if ([D0100] <= [D0102])
{
    [D0103] = 0x100;
}
```

##### 运行内容

如果 D0100 的值等于或小于 D0102 的值，100h 存储在 D0103 中。

#### 例 2.5 大于

##### 脚本

```
if ([D0100] > [D0102])
{
    [D0103] = 0x100;
}
```

##### 运行内容

如果 D0100 的值大于 D0102 的值，100h 存储在 D0103 中。

#### 例 2.6 等于或大于

##### 脚本

```
if ([D0100] >= [D0102])
{
    [D0103] = 0x100;
}
```

##### 运行内容

如果 D0100 的值等于或大于 D0102 的值，100h 存储在 D0103 中。

### 3. 逻辑运算

#### 例 3.1 逻辑与

##### 脚本

```
if (([D0100] == [D0200]) && ([D0300] == [D0400] +
[D0500]))
{
    [D0600] = 100;
}
```

##### 运行内容

如果 D0100 的值与 D0200 的值相等，且 D0400 的值与 D0500 的值相加后的值等于 D0300 的值，则 100 存储在 D0600 中。

如果  $([D0100] == [D0200])$  与  $([D0300] == [D0400] + [D0500])$  其中一个不成立时，则不执行中括号“{}”内的处理。

#### 例 3.2 逻辑或

##### 脚本

```
if ((0 != [D0100]) || (0 != [D0200]))
{
    [D0300] = 100;
}
```

##### 运行内容

如果 D0100 的值非 0，或者 D0200 的值非 0，100 存储在 D0300 中。

只要有一个表达式成立，将执行中括号“{}”内的处理。

#### 例 3.3 逻辑反转

##### 脚本

```
if (!( [D0100] == 0x1234 ))
{
    [D0300] = 100;
}
```

##### 运行内容

如果 D0100 的值不等于 1234h，100 存储在 D0300 中。

#### 例 3.4 逻辑反转

##### 脚本

```
if (!(0 != [D0100]))
{
    [D0300] = 100 ;
}
```

##### 运行内容

如果 D0100 的值为 0，100 存储在 D0300 中。

与记载  $if (0 == [D0100])$  时的处理相同。

### 4. 算术运算

#### 例 4.1 加

##### 脚本

```
[D0300] = [D0100] + [D0200];
```

##### 运行内容

将 D0100 的值与 D0200 的值相加，结果存储在 D0300 中。

#### 例 4.2 减

##### 脚本

```
[D0300] = [D0100] - [D0200];
```

##### 运行内容

从 D0100 的值中减去 D0200 的值，结果存储在 D0300 中。

#### 例 4.3 乘

##### 脚本

```
[D0300] = [D0100] * [D0200];
```

##### 运行内容

将 D0100 的值与 D0200 的值相乘，结果存储在 D0300 中。

#### 例 4.4 除

##### 脚本

```
[D0300] = [D0100] / [D0200];
```

##### 运行内容

用 D0100 的值除以 D0200 的值，结果存储在 D0300 中。

#### 例 4.5 余数

##### 脚本

```
[D0300] = [D0100] % [D0200];
```

##### 运行内容

用 D0100 的值除以 D0200 的值，其余数存储在 D0300 中。

## 5. 位运算

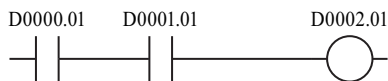
### 例 5.1 逻辑与

#### 脚本

```
if ([D0000.01] & [D0001.01])
{
    SET([D0002.01]);
}
else
{
    RST([D0002.01]);
}
```

#### 运行内容

如果 D0000.01 值和 D0001.01 值的位逻辑“与”为 1，则将 D0002.01 切换为 1。  
如果 D0000.01 值和 D0001.01 值的位逻辑“与”为 0，则将 D0002.01 切换为 0。  
动作与以下梯形图相同。



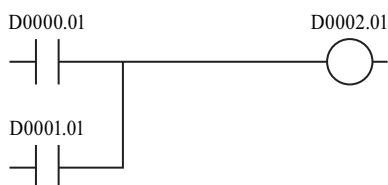
### 例 5.2 逻辑或

#### 脚本

```
if ([D0000.01] | [D0001.01])
{
    SET([D0002.01]);
}
else
{
    RST([D0002.01]);
}
```

#### 运行内容

如果 D0000.01 值和 D0001.01 值的位逻辑“或”为 1，则将 D0002.01 切换为 1。  
如果 D0000.01 值和 D0001.01 值的位逻辑“或”为 0，则将 D0002.01 切换为 0。  
动作与以下梯形图相同。



### 例 5.3 逻辑异或

#### 脚本

```
[D0200] = [D0100] ^ 0xFF;
```

#### 运行内容

将 D0100 的值和 FFh 各位的逻辑“异或”值存储在 D0200 中。  
例如，如果 D0100 的值为 15 (0Fh)，则 D0200 变为 240 (F0h)。

### 例 5.4 反转

#### 脚本

```
[D0200] = ~[D0100];
```

#### 运行内容

对 D0100 的值执行位反转，存储在 D0200 中。  
例如，如果 D0100 的值为 0，D0200 将变为 65535。

## 26: 脚本

### 例 5.5 反转

#### 脚本

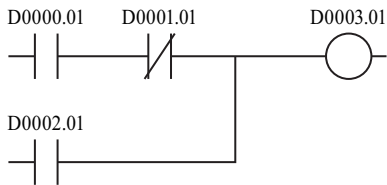
```
if (([D0000.01] & ~[D0001.01]) | [D0002.01])
{
    SET([D0003.01]);
}
else
{
    RST([D0003.01]);
}
```

#### 运行内容

如果 D0000.01 的值、D0001.01 取反值的位逻辑“与”，以及与 D0002.01 值的位逻辑“或”为 1，则 D0003.01 切换为 1。

如果 D0000.01 的值、D0001.01 取反值的位逻辑“与”，以及与 D0002.01 值的位逻辑“或”为 0，则 D0003.01 切换为 0。

动作与以下梯形图相同。



### 例 5.6 左移

#### 脚本

```
[D0300] = [D0100] << [D0200];
```

#### 运行内容

将 D0100 中的值向左移 D0200 中值的位数，结果存储在 D0300 中。

例如，假设 D0100 的值为 1，D0200 的值为 3，则将“1”左移“3”位，结果“8”存储在 D0300 中。

### 例 5.7 右移

#### 脚本

```
[D0300] = [D0100] >> [D0200];
```

#### 运行内容

将 D0100 中的值向右移 D0200 中值的位数，结果存储在 D0300 中。

例如，假设 D0100 的值为 8，D0200 的值为 3，则将“8”右移“3”位，结果“1”存储在 D0300 中。

## 6. 位函数

### 例 6.1 位置位

#### 脚本

```
SET([D0000.01]);
```

#### 运行内容

将 D0000.01 切换为 1。与 [D0000.01] = 1 的结果相同。

### 例 6.2 位复位

#### 脚本

```
RST([D0000.01]);
```

#### 运行内容

将 D0000.01 切换为 0。与 [D0000.01] = 0 的结果相同。

### 例 6.3 位反转

#### 脚本

```
REV([D0000.01]);
```

#### 运行内容

对 D0000.01 的 1 和 0 进行反转。与 [D0000.01] = ~[D0000.01] 的结果相同。

## 7. 字函数

### 算术运算

#### 例 7.1 最大值

#### 脚本

```
[D0200] = MAX([D0100], [D0110], [D0120], [D0130],  
[D0140]);
```

#### 运行内容

将存储在 D0100、D0110、D0120、D0130、D0140 中的最大值存储到 D0200 中。  
最多可记载 15 点参数。

#### 例 7.2 最小值

#### 脚本

```
[D0200] = MIN([D0100], [D0110], [D0120], [D0130],  
[D0140]);
```

#### 运行内容

将存储在 D0100、D0110、D0120、D0130、D0140 中的最小值存储到 D0200 中。  
最多可记载 15 点参数。

#### 例 7.3 指数函数

#### 脚本

```
[D0010] = EXP([D0020]);
```

#### 运行内容

运算 D0020 值的指数函数，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

## 26: 脚本

---

### 例 7.4 自然对数

#### 脚本

```
[D0010] = LOGE([D0020]);
```

#### 运行内容

运算 D0020 值的自然对数，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

### 例 7.5 常用对数

#### 脚本

```
[D0010] = LOG10([D0020]);
```

#### 运行内容

运算以 D0020 的值 10 为底的对数，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

### 例 7.6 乘方

#### 脚本

```
[D0010] = POW([D0020],[D0030]);
```

#### 运行内容

运算乘方。

例如，当 D0020 的值为 10、D0030 的值为 5 时，运算 10 的 5 次方，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

### 例 7.7 平方根

#### 脚本

```
[D0010] = ROOT([D0020]);
```

#### 运行内容

运算 [D0020] 值的平方根，结果存储到 [D0010] 中。

只能使用数据类型 F（浮点）。

### 例 7.8 正弦

#### 脚本

```
[D0010] = SIN([D0020]);
```

#### 运行内容

运算 D0020 弧度值的正弦，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

### 例 7.9 余弦

#### 脚本

```
[D0010] = COS([D0020]);
```

#### 运行内容

运算 D0020 弧度值的余弦，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。



**例 7.10 正切****脚本**

```
[D0010] = TAN([D0020]);
```

**运行内容**

运算 D0020 弧度值的正切，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

**例 7.11 反正弦****脚本**

```
[D0010] = ASIN([D0020]);
```

**运行内容**

运算 D0020 值的反正弦，结果以弧度存储到 D0010 中。

只能使用数据类型 F（浮点）。

**例 7.12 反余弦****脚本**

```
[D0010] = ACOS([D0020]);
```

**运行内容**

运算 D0020 值的反余弦，结果以弧度存储到 D0010 中。

只能使用数据类型 F（浮点）。

**例 7.13 反正切****脚本**

```
[D0010] = ATAN([D0020]);
```

**运行内容**

运算 D0020 值的反正切，结果以弧度存储到 D0010 中。

只能使用数据类型 F（浮点）。

**例 7.14 角度→弧度变换****脚本**

```
[D0010] = RAD([D0020]);
```

**运行内容**

将 D0020 的值由度（°）变换成弧度，结果存储到 D0010 中。

只能使用数据类型 F（浮点）。

**例 7.15 弧度→角度变换****脚本**

```
[D0010] = DEG([D0020]);
```

**运行内容**

将 D0020 的值由弧度变换成度（°），结果存储到 D0010。

只能使用数据类型 F（浮点）。

## 26: 脚本

### 数据类型变换

例 7.16 BCD → 二进制变换

脚本

```
[D0200] = BCD2BIN([D0100]);
```

运行内容

将 D0100 的 BCD 值变换成二进制值，存储到 D0200 中。

例如，如果将 BCD 值 10（二进制值为 16）存储到 D0100，则将 10（二进制值）存储到 D0200 中。

例 7.17 二进制 → BCD 变换

脚本

```
[D0200] = BIN2BCD([D0100]);
```

运行内容

将 D0100 的二进制值变换成 BCD 值，存储到 D0200 中。

例如，如果将二进制值 16（BCD 值为 10）存储到 D0100，则将 16（BCD 值）存储到 D0200 中。

例 7.18 浮点 → 二进制变换

脚本

```
[D0200] = FLOAT2BIN([D0100]);
```

运行内容

将 D0100 的浮点值变换成二进制值，存储到 D0200 中。

例如，如果数据类型 F（浮点）1234.0（二进制值为 449A4000h）存储在 D0100 中，则将 1234（二进制值）存储在 D0200 中。同时，如果数据类型 F（浮点）1234.56（二进制值为 449A51ECh）存储在 D0100 中，则舍去小数点以下，将 1234（二进制值）存储在 D0200 中。

例 7.19 二进制 → 浮点变换

脚本

```
[D0200] = BIN2FLOAT([D0100]);
```

运行内容

将 D0100 的二进制值变换成浮点值，存储在 D0200 中。

例如，如果二进制值 1234 存储在 D0100 中，则将浮点值 1234.0（二进制值为 449A4000h）存储在 D0200 中。

例 7.20 十进制 → 字符串变换

脚本

```
DEC2ASCII([D0100], [D0200]);
```

运行内容

将 D0200 的十进制数值变化成字符串，以 D0100 为起始地址按顺序存储。

注释：

- 可在数据类型 W（字）、I（整数）、D（双字）和 L（长整数）中使用此函数。
- 在字符串的末尾，将附带终端字符 NULL（00h）。

对 1234 进行变换（数据类型为 W（字）时）

设备	存储值
D0200	1234



设备	存储值	
	高位字节	低位字节
D0100	'1' = 31h	'2' = 32h
D0101	'3' = 33h	'4' = 34h
D0102	00h	00h

终端字符

对 -12345 进行变换（数据类型为 I（整数）时）

设备	存储值
D0200	-12345

⇒

设备	存储值	
	高位字节	低位字节
D0100	'1' = 2Dh	'1' = 31h
D0101	'2' = 32h	'3' = 33h
D0102	'4' = 34h	'5' = 35h
D0103	00h	00h

终端字符

对 1234567890 进行变换（数据类型为 D（双字）时）

设备	存储值
D0200	1234567890
D0201	

⇒

设备	存储值	
	高位字节	低位字节
D0100	'1' = 31h	'2' = 32h
D0101	'3' = 33h	'4' = 34h
D0102	'5' = 35h	'6' = 36h
D0103	'7' = 37h	'8' = 38h
D0104	'9' = 39h	'0' = 30h
D0105	00h	00h

终端字符

对 -1234567890 进行变换（数据类型为 L（长整数）时）

设备	存储值
D0200	-1234567890
D0201	

⇒

设备	存储值	
	高位字节	低位字节
D0100	'1' = 2Dh	'1' = 31h
D0101	'2' = 32h	'3' = 33h
D0102	'4' = 34h	'5' = 35h
D0103	'6' = 36h	'7' = 37h
D0104	'8' = 38h	'9' = 39h
D0105	'0' = 30h	00h

终端字符

例 7.21 字符串→十进制变换

脚本

```
[D0100] = ASCII2DEC([D0200]);
```

运行内容

将以 D0200 为起始地址存储的字符串变换为十进制，结果存储到 D0100。

可变换的位数为各数据类型的最大位数加上符号的值。

如果用于变换的字符串中含有 NULL 或无法变换为数值的字符时，只变换到该字符为止。

注释：

- 可在数据类型 W（字）、I（整数）、D（双字）和 L（长整数）中使用此函数。
- 在字符串的末尾，将附带终端字符 NULL（0x00）。

## 26: 脚本

指定字符串“1234”（数据类型为W（字）时）

设备	存储值	
	高位字节	低位字节
D0200	'1' = 31h	'2' = 32h
D0201	'3' = 33h	'4' = 34h
D0202	00h	00h

终端字符



设备	存储值
D0100	1234

对字符串“1234567”进行设置（数据类型为D（双字）时）

设备	存储值	
	高位字节	低位字节
D0200	'1' = 31h	'2' = 32h
D0201	'3' = 33h	'4' = 34h
D0202	'5' = 35h	'6' = 36h
D0203	'7' = 37h	00h

终端字符



设备	存储值
D0100	12345

指定字符串“-12345”（数据类型为I（整数）时）

设备	存储值	
	高位字节	低位字节
D0200	'1' = 2Dh	'1' = 31h
D0201	'2' = 32h	'3' = 33h
D0202	'4' = 34h	'5' = 35h
D0203	00h	00h

终端字符



设备	存储值
D0100	-12345

指定字符串“1234567890”（数据类型为L（长整数）时）

设备	存储值	
	高位字节	低位字节
D0200	'1' = 31h	'2' = 32h
D0201	'3' = 33h	'4' = 34h
D0202	'5' = 35h	'6' = 36h
D0203	'7' = 37h	'8' = 38h
D0204	'9' = 39h	'0' = 30h
D0205	00h	00h

终端字符



设备	存储值
D0100	1234567890

## 数据的比较与复制

### 例 7.22 字单位的数据比较

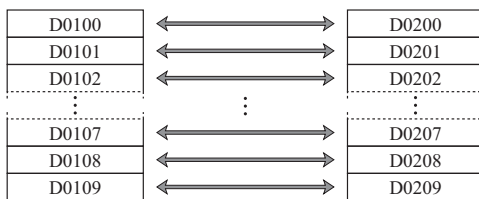
#### 脚本

```
[D0000] = MEMCMP([D0100], [D0200], 10);
```

#### 运行内容

将 D0100 起始的 10 字（最大 D0109）与 D0200 起始的 10 字（最大 D0209）的值进行比较。如果各值全部一致，在 D0000 中存储 1。即使有 1 个不一致便存储 0。

#### 分别比较



**注释：**即使将数据类型设置为 D（双字）、L（长整数）或 F（浮点），也会以字为单位从起始设备开始进行比较。

### 例 7.23 位单位的数据比较

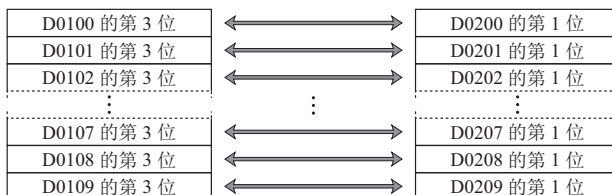
#### 脚本

```
[D0000] = MEMCMP([D0100.02], [D0200.00], 10);
```

#### 运行内容

将 D0100 的第 3 位～D0109 的第 3 位、与 D0200 的第 1 位～D0209 的第 1 位的位状态进行比较。如果各值全部一致，在 D0000 中存储 1。即使有 1 个不一致便存储 0。

#### 分别比较



**注释：**即使将数据类型设置为 D（双字）、L（长整数）或 F（浮点），也会以字为单位从起始设备开始进行比较。

### 例 7.24 字单位的数据复制

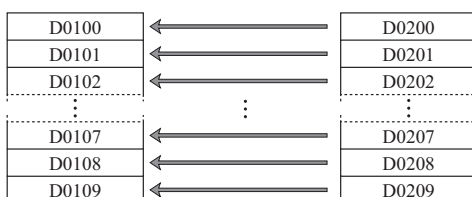
#### 脚本

```
MEMCPY([D0100], [D0200], 10);
```

#### 运行内容

将 D0200 起始的 10 字（最大 D0209）设备值复制到 D0100 起始的 10 字（最大 D0109）设备中。

#### 分别复制



**注释：**即使将数据类型设置为 D（双字）、L（长整数）或 F（浮点），也会以字为单位从起始设备开始进行复制。

## 26: 脚本

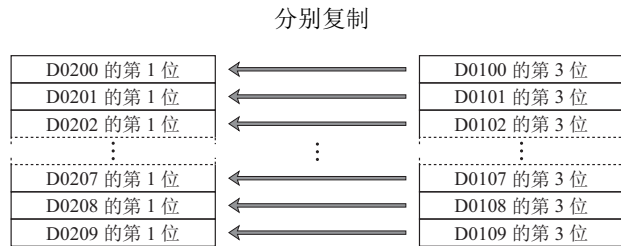
### 例 7.25 位单位的数据复制

#### 脚本

```
MEMCPY([D0200.00], [D0100.02], 10);
```

#### 运行内容

将从 D0100 起始的 10 字（最大 D0109）的第 3 位，复制到从 D0200 起始的 10 位设备（最大 D0209）对应的位状态中。



**注释：**即使将数据类型设置为 D（双字）、L（长整数）或 F（浮点），也会以位为单位从起始设备开始复制位。

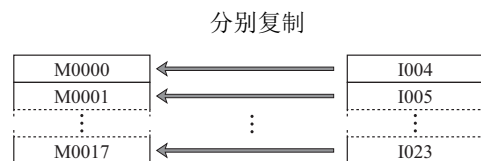
### 例 7.26 从位设备复制 1 个字到位设备

#### 脚本

```
BITS2BITS([M0000], [I004]);
```

#### 运行内容

将从 I004（I004 到 I023）开始的 1 个字长的值复制到从 M0000（M0000 到 M0017）开始的 1 个字区域的位状态中。



### 例 7.27 从位设备复制 1 个字到字设备

#### 脚本

```
BITS2WORD([D0000], [I004]);
```

#### 运行内容

将从 I004（到 I023）开始的 1 个字长的值复制到设备 D0000 的值中。  
这与 BITS2BITS(I004, D0000.0); 相同。

### 例 7.28 从字设备复制 1 个字到位设备中

#### 脚本

```
WORD2BITS([M0000], [D0100]);
```

#### 运行内容

将设备 D0100 的值复制到从 M0000（M0000 到 M0017）开始的 1 个字长值的位状态中。  
这与 BITS2BITS(D0100.0, M0000); 相同。

## 字符串操作

以 1 个字以上的连续数据寄存器设置字符串。将 2 字节的数据作为 1 个字，按高位字节→低位字节的顺序进行设置。请在字符串的末尾设置 00h。

### 例 7.29 复制字符串

#### 脚本

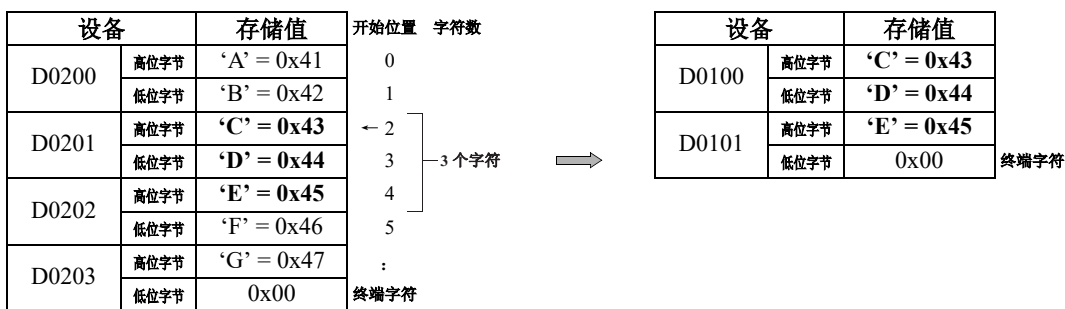
```
STRCUT([D0100], [D0200], 2, 3);
```

#### 运行内容

从 D0200 起始的字符串“ABCDEFGH”的开始位置 2（由于从 0 开始，因此为第 3 个字符）开始，将字符数 3（3 个字符）按从 D0100 开始的顺序进行存储。

**注释：**可在开始位置为 0 至 127、字符数为 1 至 128 的范围内进行指定。

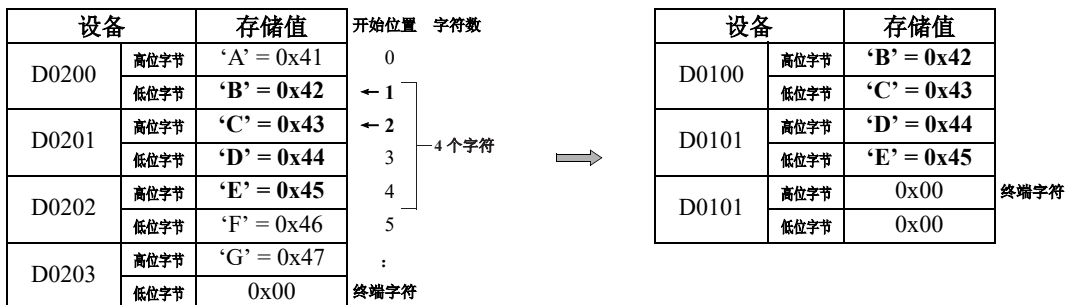
从字符串“ABCDEFGH”开始，以开始位置 2、字符数 3 进行复制



#### 脚本

```
STRCUT([D0100], [D0200], 1, 4);
```

从字符串“ABCDEFGH”开始，以开始位置 1、字符数 4 进行复制



## 26: 脚本

### 例 7.30 计数字符数

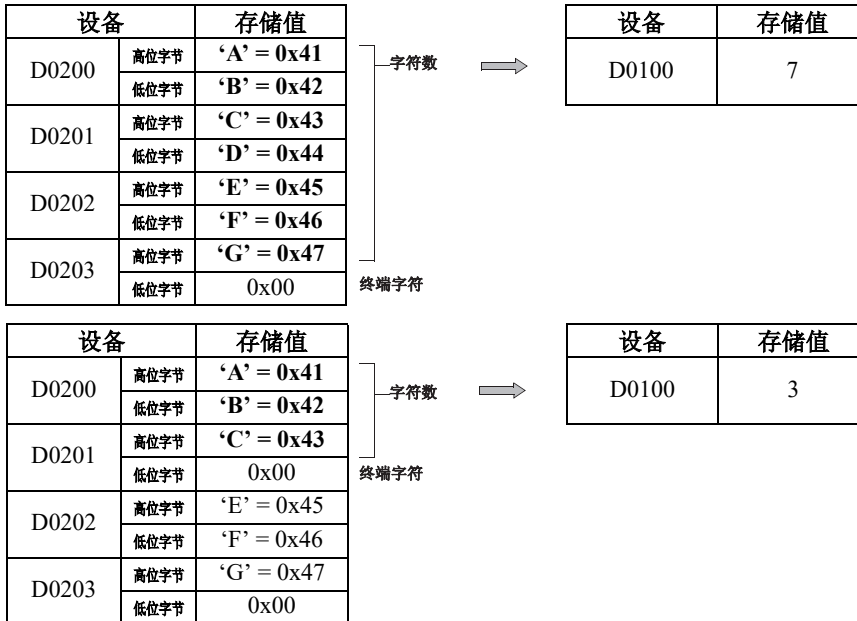
#### 脚本

```
[D0100] = STRLEN([D0200]);
```

#### 运行内容

调查 D0200 的起始字符串长度（字符数），存储到 D0100 中。

**注释：**将终端字符 NULL（00h）设为字符串的末尾。（终端字符不包含在字符串的长度中。）



### 例 7.31 连接字符串

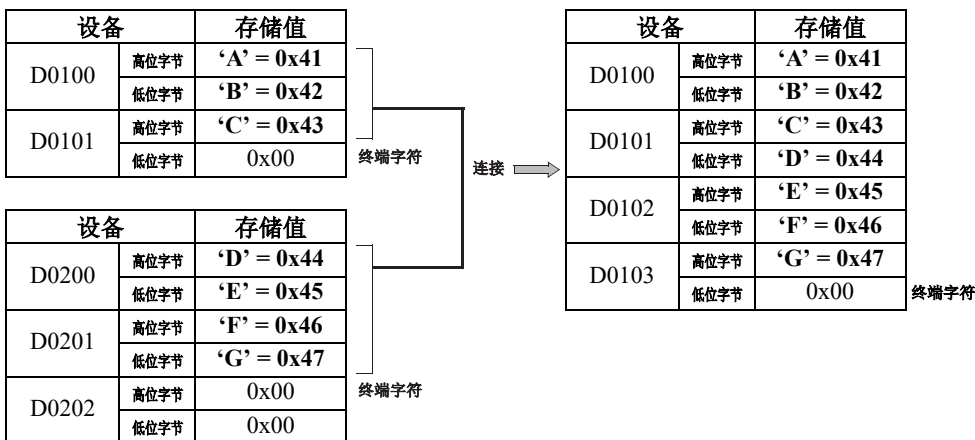
#### 脚本

```
STRCAT([D0100], [D0200]);
```

#### 运行内容

将 D0200 起始的字符串连接到 D0100 起始的字符串中。

**注释：**将终端字符 NULL（00h）设为字符串的末尾。（终端字符不包含在字符串的长度中。）





## 例 7.32 搜索字符串

## 脚本

```
[D0000] = STRSTR([D0100], [D0200]);
```

## 运行内容

从 D0100 起始的“查找对象的字符串”“ABCDEFGHJKLMNO”开始，搜索 D0200 起始的“查找内容”“DEFG”，并将字符串的出现位置存储到 D0000 中。若未发现时，将 -1 存储到 D0000 中。

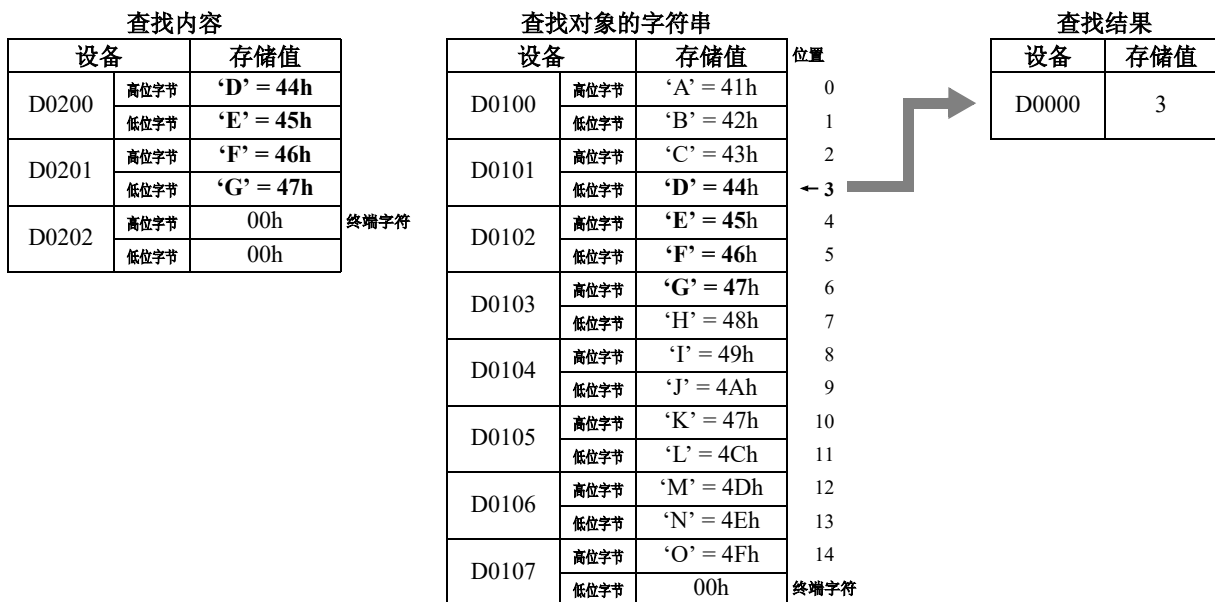
此外，如果在要搜索的字符中指定“?”，则将作为任意的 1 字节字符进行处理。

指定作为字符的“?”（3Fh）时，以“~?”（7E3Fh）和 2 字节进行指定。

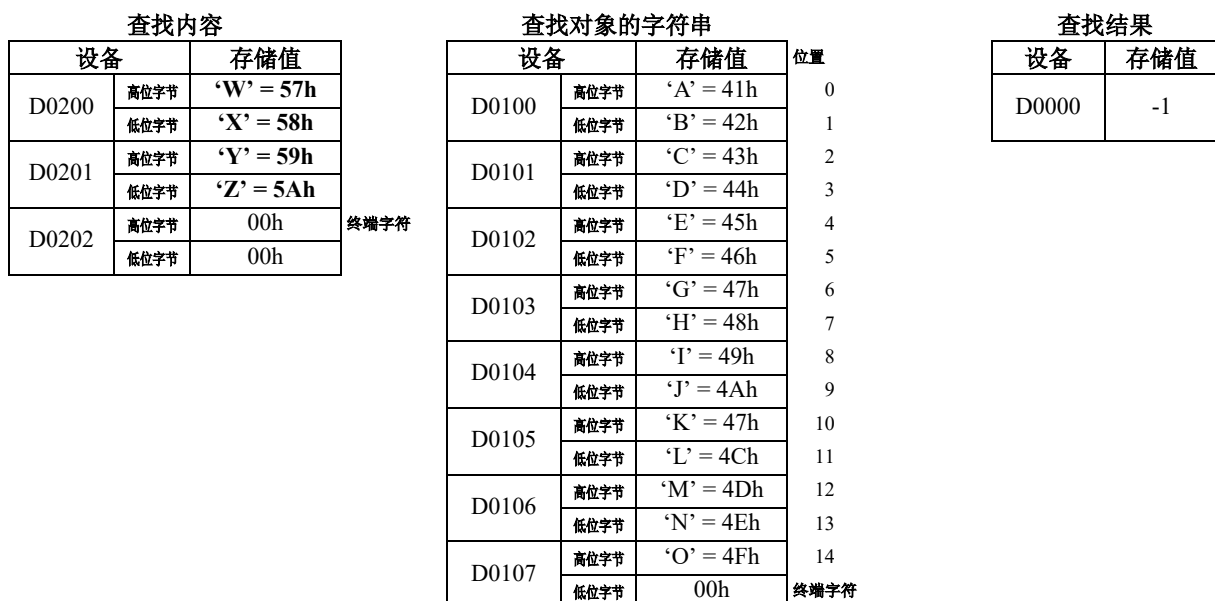
指定作为字符的“~”（7Eh）时，以“~~”（7E7Eh）和 2 字节进行指定。

**注释：**要搜索的字符串最大字符数为 128 个半角字符。

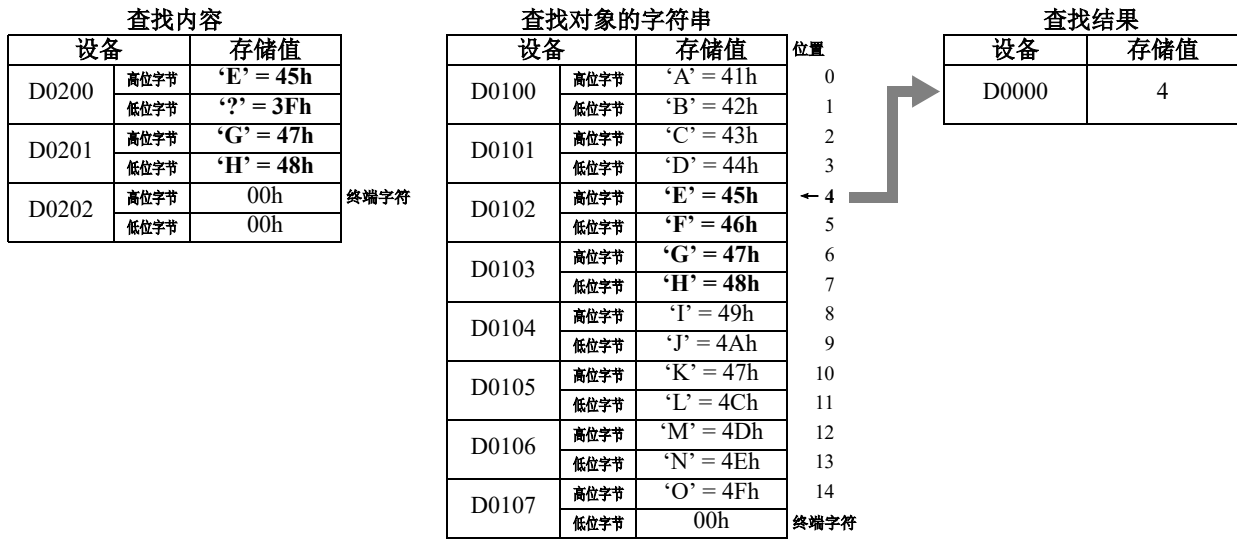
## 以“DEFG”进行搜索，未发现字符串时



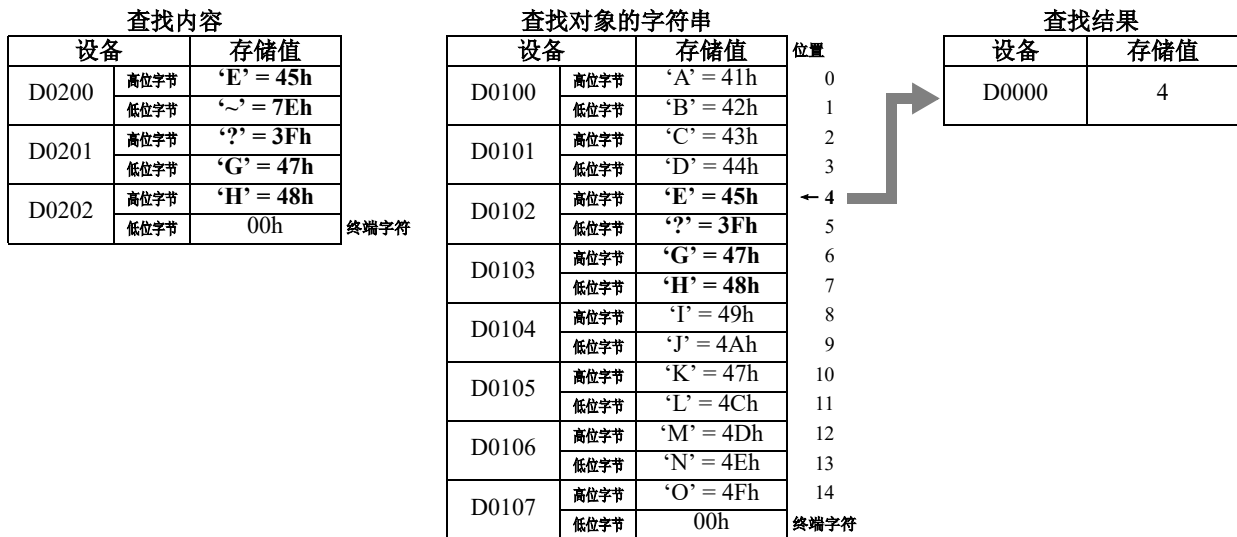
## 以“WXYZ”进行搜索，未发现字符串时



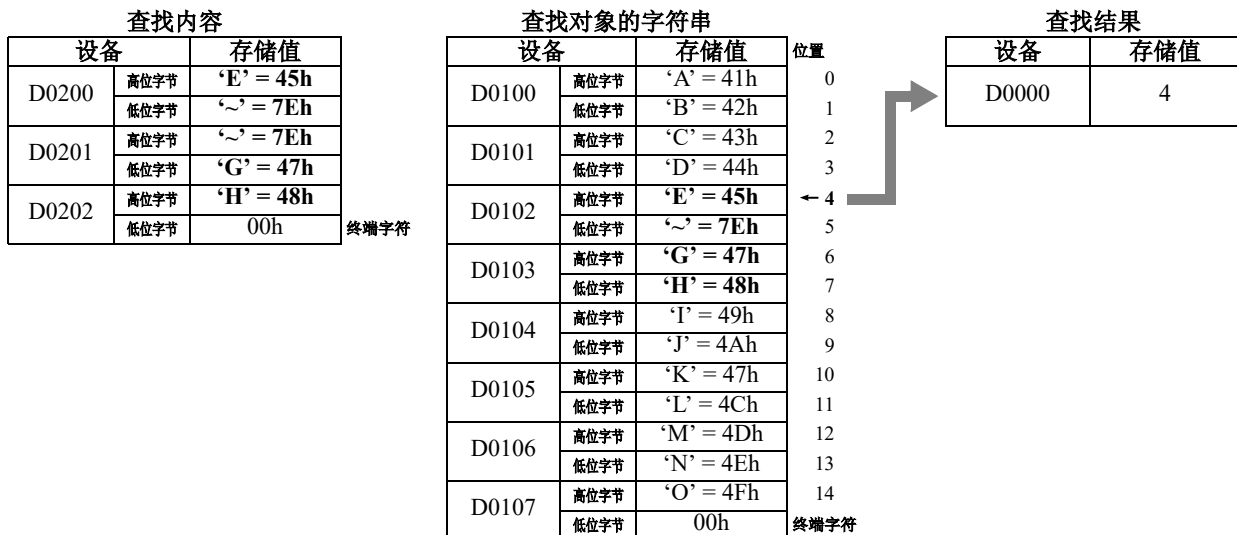
作为任意的 1 字节字符，搜索“?” 时



搜索作为字符的“~” 时



搜索作为字符的“~” 时



## 间接指定

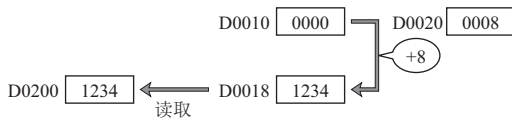
### 例 7.33 间接读取

#### 脚本

```
[D0200] = OFFSET([D0010], [D0020]);
```

#### 运行内容

D0020 的值为 8 时，读取 D0010 起始第 8 字的设备 D0018 值，存储到 D0200 中。



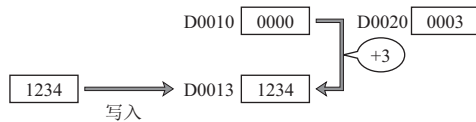
### 例 7.34 间接写入

#### 脚本

```
OFFSET([D0010], [D0020]) = 1234;
```

#### 运行内容

D0020 的值为 3 时，将常数 1234 存储到 D0010 起始第 3 字的设备 D0013 中。



## 8. 数据类型指定

例 8.1 使用了数据类型指定的 F（浮点）、D（双字）、W（字）数据类型混合存在的处理

#### 脚本

```
[D0100] = 1.5;
[D0110] = 10.5;
asd
{
    [D0200] = FLOAT2BIN([D0100]);
    [D0210] = FLOAT2BIN([D0110]);
    [D0300] = [D0200]^ [D0210];
}
asword
{
    [D0400] = [D0301];
}
```

#### 运行内容

将数据类型 F（浮点）的 D0100 的值转换为数据类型 D（双字），存储到 D0200 和 D0201 中。

将数据类型 F（浮点）的 D0110 的值转换为数据类型 D（双字），存储到 D0210 和 D0211 中。

进行数据类型 D（双字）的 D0200 和 D0210 各位逻辑异或运算，并将结果存储到 D0300 和 D0301 中。

将数据类型 D（双字）的 D0300 的低位字 D0301 的值存储到 D0400 中。

上述脚本示例中，脚本编辑器的数据类型选择为 F（浮点），“设备设置”选择为“从高位字开始”。

将 32 位数据存储在设备的方法，遵照功能设置中指定的数据存储方法。

有关详情，请参见第 3-10 页上的“32 位数据存储”。

### 注意事项

以下介绍记载脚本时的注意事项。

#### 关于 while 语句的注意事项

■ 记载时，请务必避免形成无限循环。

在 while 语句条件式成立期间，重复执行执行式。

但是，如果像下述示例所示，条件式始终成立时，则形成无限循环。

```
[D0100] = 10;

while (0 != [D0100])
{
    [D0200] = [D0200] + 1;
}
```

在 while 语句的条件式中记载有：如果 D0100 的值为 0，则退出 while 语句，然而在脚本的第 1 行中，将 10 存储在 D0100 后未进行变更，因而成为无限循环。

使用 while 语句时，请在记载中避免形成无限循环。

如果执行变成无限循环，则扫描时间将超过监视定时器设置，将会发生监视定时器错误。

在以下的示例中，若重复执行 10 次 while 语句，则 D0100 的值将变为 0，退出 while 语句。

```
[D0100] = 10;

while (0 != [D0100])
{
    [D0200] = [D0200] + 1;
    [D0100] = [D0100] - 1;
}
```

#### 数据类型指定相关注意事项

■ 请进行记述，避免在数据类型指定期间执行其他数据类型指定。

数据类型指定的调用层级最多为 1。

如下例所示，如果在数据类型指定期间执行了其他数据类型指定，通过脚本编辑器添加脚本时将发生错误。

```
[D0100] = 1.5;
[D0110] = 10.5;
asd
{
    [D0200] = FLOAT2BIN([D0100]);
    [D0210] = FLOAT2BIN([D0110]);
    [D0300] = [D0200]^ [D0210];

    asword
    {
        [D0400] = [D0301];
    }
}
```

■ 请进行记述，避免在数据类型指定期间使用暂存设备。

数据类型指定的中括号“{}”内的处理无法使用暂存设备。


暂存设备仅以脚本编辑器的“数据类型”中设置的数据类型执行动作。

如下例所示，如果在数据类型指定期间使用了暂存设备，通过脚本编辑器添加脚本时将发生错误。

```
@1 = 1.5;
asd
{
    [D0200] = FLOAT2BIN(@1);
}
```

## 关于运算符优先级

通常，是从行的左侧开始按顺序进行式的运算，但在组合使用若干运算时，按以下优先级执行运算。

优先级	运算符
高  低	()
	! ~ - (负数)
	* / %
	+ - (减法)
	<< >>
	&
	^
	< <= > >=
	== !=
	&&
	=



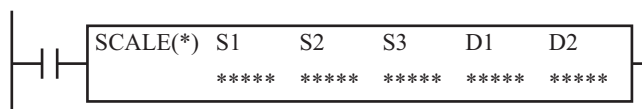
# 27: 流量计算指令

本章将对输出流量和流量累积量的流量计算指令进行介绍。

## SCALE（模拟量输入转换）

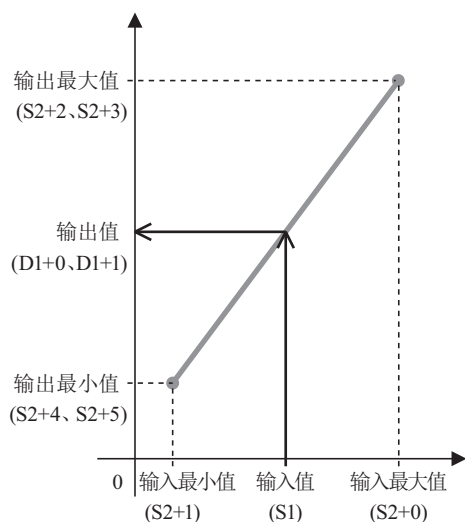
根据指定的 2 点间的坐标缩放模拟量输入值，并输出其结果。

FT2J FC6A



输入为打开时，按照 S2 中指定的数据寄存器内的设置，对 S1 中指定的数据寄存器的值进行缩放，并将运算结果存储到输出值（D1+0、D1+1）中。

如下图所示，将输入值代入到由 S2 设置的输出最大值、输出最小值、输入最大值及输入最小值的 4 个预置值所计算出的一次函数中，可计算出缩放。



### 注释:

- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 在以下情况下，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。
  - S1 或 S2 指定的数据不符合正常浮点格式时
  - 输出最大值、输出最小值、静带 (S2+6、S2+7) 为 INF (无穷) 或 NAN (无效值) 时
- 在以下情况下，将存储错误代码 46 在用户程序执行错误代码 (D8006) 中。
  - 输入最小值 (S2+1) 大于等于输入最大值 (S2+0) (输入最小值 (S2+1)  $\geq$  输入最大值 (S2+0)) 时
  - 输出最小值 (S2+4、S2+5) 大于等于输出最大值 (S2+2、S2+3) (输出最小值 (S2+4、S2+5)  $\geq$  输出最大值 (S2+2、S2+3)) 时
  - 静带 (S2+6、S2+7) 为负数时
- SCALE 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

## 27: 流量计算指令

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1 (源 1)	输入值	—	—	—	—	—	—	X	—	—	—
S2 (源 2)	控制寄存器	—	—	—	—	—	—	▲	—	—	—
S3 (源 3)	初始化输入	X	—	X	—	—	—	—	—	—	—
D1 (目标 1)	输出寄存器	—	—	—	—	—	—	▲	—	—	—
D2 (目标 2)	输出继电器	—	X	▲	—	—	—	—	—	—	—

▲不能将特殊数据寄存器指定为 S2 和 D1。不能将特殊内部继电器指定为 D2。

关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

### 有效数据类型

数据类型	W (字)	I (整数)	D (双字)	L (长整数)	F (浮点)
可指定	X	X	—	—	—

数据类型的设置仅会反映至输入值。



## 设置项目

“SCALE（扩展模块值）”对话框中包括 2 个选项卡，分别为“设备”选项卡、“设置”选项卡。

在“设备”选项卡中设置要通过 SCALE 指令使用的设备。在“设置”选项卡中设置 SCALE 指令的各项初始值。

### ■ “设备”选项卡



#### (1) S1（源 1）：输入值

指定数据寄存器，作为存储转换目标输入值的设备。

输入值大于输入最大值（S2+0）时（输入值 > 输入最大值（S2+0）），缩放输入最大值（S2+0）。

输入值小于输入最小值（S2+1）时（输入最小值（S2+1） > 输入值），缩放输入最小值（S2+1）。

#### (2) S2（源 2）：控制寄存器

指定存储输入最大值、输入最小值、输出最大值、输出最小值、静带的数据寄存器。

以指定的数据寄存器为起始，连续占用 8 个字。

有关控制寄存器的详情，请参见第 27-5 页上的“S2：控制寄存器”。

#### (3) S3（源 3）：初始化输入

指定输入或内部继电器，作为初始化控制寄存器的设备。

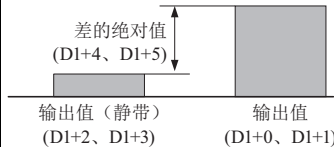
初始化输入为打开时，会将“SCALE（扩展模块值）”对话框的“设置”选项卡中设置的初始值存储到控制寄存器中。

初始化输入为打开时，每次扫描都会将初始值存储到数据寄存器中。希望仅执行 1 次初始化时，请将 SOTU（上升沿微分）指令或 SOTD（下降沿微分）指令添加到输入条件中。有关初始值设置方法的详情，请参见第 27-5 页上的“(2) 初始值”。

## 27: 流量计算指令

### (4) D1（目标 1）：输出寄存器

指定存储输出值、输出值（静带）、输出变化量的数据寄存器。  
以指定的数据寄存器为起始，连续占用 6 个字。

存储目的地	功能	设置	数据类型
D1+0 D1+1	输出值	存储每次扫描缩放后的值。	
D1+2 D1+3	输出值（静带）	禁用静带时（静带（S2+6、S2+7）为 0），每次扫描都会将输出值（D1+0、D1+1）存储到输出值（静带）中。 仅在启用静带时（静带（S2+6、S2+7）为 0 以外数值）或输出值（D1+0、D1+1）与输出值（静带）之差的绝对值大于等于静带（S2+6、S2+7）时（差 $\geq$ 静带（S2+6、S2+7）），会打开输出值（静带）更改通知（D2+0），并在输出值（静带）中存储输出值（D1+0、D1+1）。 有关详情，请参见第 27-3 页上的“设置项目”。	F（浮点）
D1+4 D1+5	输出变化量	存储输出值（D1+0、D1+1）与输出值（静带）（D1+2、D1+3）之差的绝对值。 	F（浮点）*1

\*1 数据范围为 0、1.175494E-38 ~ 3.402823E+38。

### (5) D2（目标 2）：输出继电器

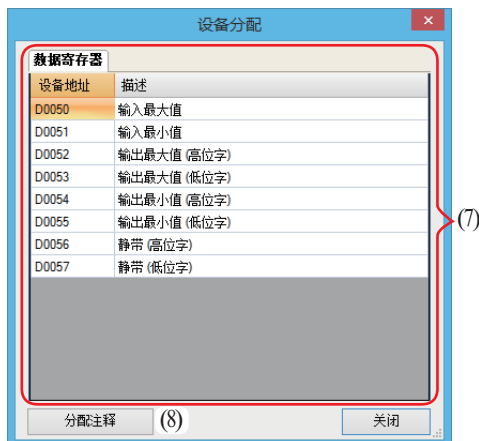
指定存储输出值（静带）更改通知、输入值报警输出的输出或内部继电器。以指定的设备为起始，连续占用 2 个字。

存储目的地	功能	设置
D2+0	输出值（静带）更改通知	输出值（D1+0、D1+1）与输出值（静带）（D1+2、D1+3）之差的绝对值大于静带（S2+6、S2+7）时，会在 1 次扫描中打开。此时，会将输出值（D1+0、D1+1）存储到输出值（静带）（D1+2、D1+3）中。
D2+1	输入值报警输出	输入值（S1）大于输入最大值（S2+0）或输入值小于输入最小值（S2+1）时打开。此时，如果输入值（S1）大于输入最大值（S2+0），则以输入最大值（S2+0）进行缩放。如果输入值（S1）小于输入最小值（S2+1），则以输入最小值（S2+1）进行缩放。

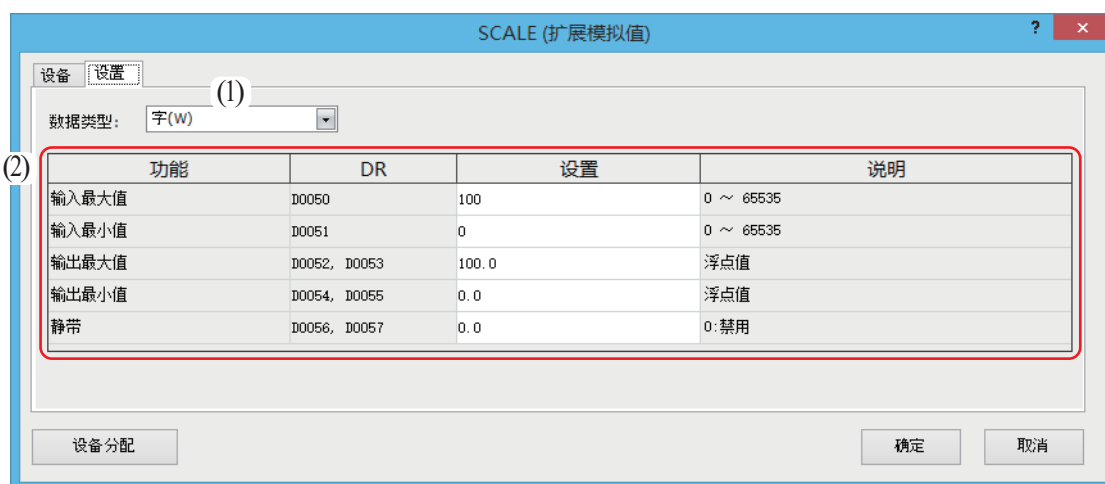
### (6) 设备分配

单击此按钮，将显示“设备分配”对话框。如下所示，将在对话框中显示 SCALE 指令的各设置内容和数据寄存器、内部继电器的对应表 (7)。而且，通过单击“分配注释”按钮 (8)，可将各设置内容设置到对应的数据寄存器、内部继电器的注释中。

“设备分配”对话框



## ■ “设置”选项卡



### (1) 数据类型

从“W（字）”或“I（整数）”中选择输入值（S1）的数据类型。

### (2) 初始值

初始化输入为打开时，设置控制寄存器中存储的各功能的初始值。有关初始值范围的详情，请参见第 27-5 页上的“S2：控制寄存器”。

## S2: 控制寄存器

存储目的地	功能	设置	数据类型
S2+0	输入最大值	请设置为输入最大值 > 输入最小值（S2+0）。 初始值为 100。	W（字） I（整数）
S2+1	输入最小值	请设置为输入最小值 < 输入最大值（S2+1）。 初始值为 0。	
S2+2 S2+3	输出最大值	请设置为输出最大值 > 输出最小值（S2+4、S2+5）。 初始值为 100.0。	F（浮点）
S2+4 S2+5	输出最小值	请设置为输出最小值 < 输出最大值（S2+2、S2+3）。 初始值为 0.0。	
S2+6 S2+7	静带	取样输出变化量*1（D1+4、D1+5），仅在该输出变化量大于等于静带时， 打开输出值（静带）更改通知（D2+0）。此时，将在输出值（D1+0、 D1+1）中存储输出值（静带）（D1+2、D1+3）。 静带为 0 时，静带功能为禁用状态，因此每次扫描都会将输出值（D1+0、 D1+1）存储到输出值（静带）（D1+2、D1+3）中。 初始值为 0。	F（浮点）*2

\*1 有关输出变化量的详情，请参见第 27-4 页上的“(4)D1（目标 1）：输出寄存器”

\*2 数据范围为 0、1.175494E-38 ~ 3.402823E+38。

### 静带功能

静带功能是指过滤 SCALE 指令中所输入值的细微变化的功能。

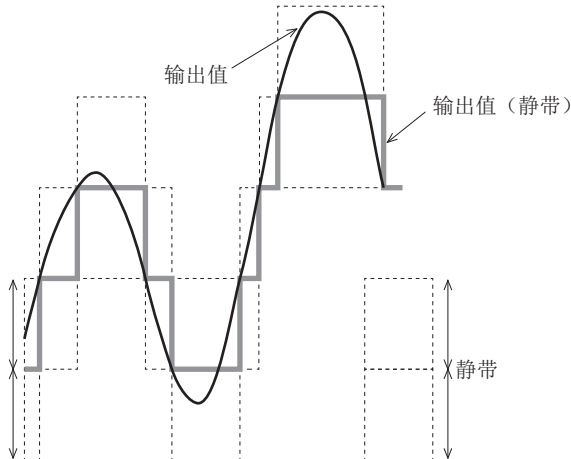
SCALE 指令会始终保持以下 2 个值。

输出值： 每次扫描、缩放输入值的结果

输出值（静带）： 以静带过滤输入值的结果

将过滤的变动范围设为静带。

当输出值的变动范围比静带小时，将无法刷新输出值（静带）。



执行 SCALE 指令时，SCALE 指令会将第 1 次扫描缩放的值存储到输出值（D1+0、D1+1）中，并刷新输出值（静带）（D1+2、D1+3）的值。

此时，动作因静带的启用 / 禁用而异。

#### 启用静带时

比较输出值（D1+0、D1+1）与输出值（静带）（D1+2、D1+3），并根据其差的绝对值（D1+4、D1+5）和静带（S2+6、S2+7）的大小，进行如下存储。

- 输出变化量（D1+4、D1+5）（输出值（D1+0、D1+1）与输出值（静带）（D1+2、D1+3）之差的绝对值）小于静带（S2+6、S2+7）时  
不会将输出值（D1+0、D1+1）存储到输出值（静带）（D1+2、D1+3）中。
- 输出变化量（D1+4、D1+5）（输出值（D1+0、D1+1）与输出值（静带）（D1+2、D1+3）之差的绝对值）大于等于静带（S2+6、S2+7）时  
将输出值（D1+0、D1+1）存储到输出值（静带）（D1+2、D1+3）中，并在 1 次扫描中打开输出值（静带）更改通知（D2+0）。

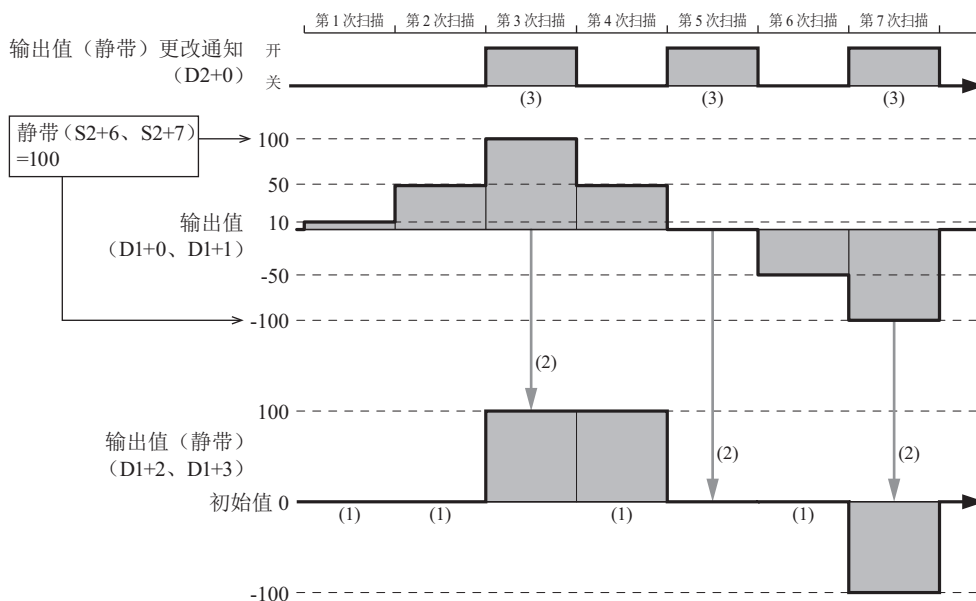
#### 禁用静带时（静带（S2+6、S2+7）=0）

每次扫描都会将输出值（D1+0、D1+1）存储到输出值（静带）（D1+2、D1+3）中。

## 例 1) 启用静带时

静带 (S2+6、S2+7) 为 100、输出值 (静带) (D1+2、D1+3) 的初始值为 0 时, 输出值 (D1+0、D1+1) 的值的对应动作如下所示。

周期	输出值 (D1+0、D1+1)	输出值 (静带) (D1+2、D1+3)	输出变化量 (D1+4、D1+5)	输出值 (静带) 更改通知 (D2+0)
第 1 次扫描	10	0	10	关闭
第 2 次扫描	50	0	50	关闭
第 3 次扫描	100	100	100	打开
第 4 次扫描	50	100	50	关闭
第 5 次扫描	0	0	100	打开
第 6 次扫描	-50	0	50	关闭
第 7 次扫描	-100	-100	100	打开



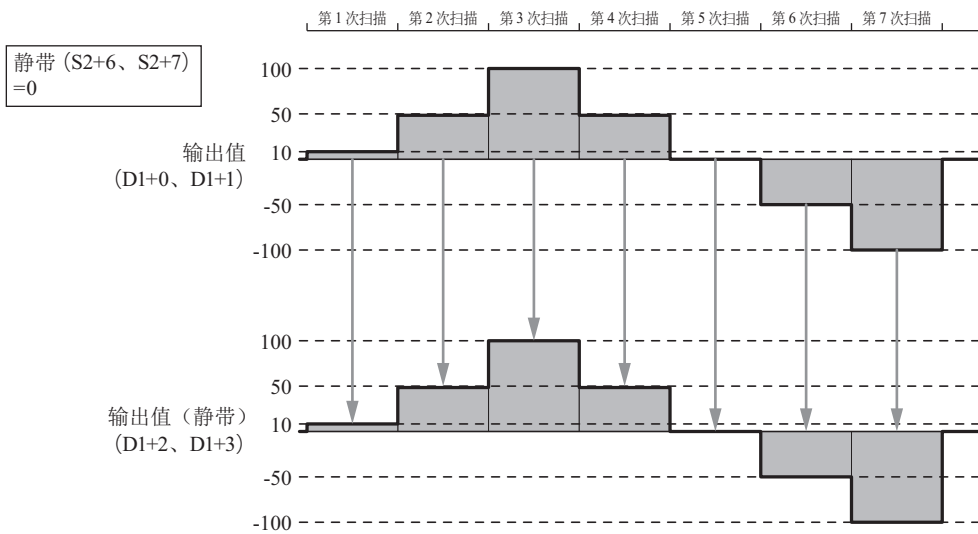
- (1) 由于输出变化量 (D1+4、D1+5) 小于静带, 因此输出值 (D1+0、D1+1) 不会存储到出力值 (静带) (D1+2、D1+3) 中。
- (2) 由于输出变化量 (D1+4、D1+5) 大于等于静带, 因此输出值 (D1+0、D1+1) 将会存储到出力值 (静带) (D1+2、D1+3) 中。
- (3) (2) 时会在 1 次扫描中打开输出值 (静带) 更改通知 (D2+0)。

## 27: 流量计算指令

### 例 2) 禁用静带时

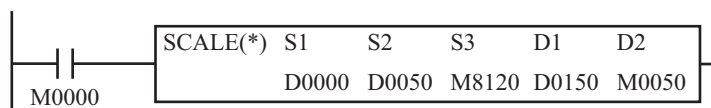
静带 (S2+6、S2+7) 为 0 且输出值 (静带) (D1+2、D1+3) 的初始值为 0 时, 每次扫描都会将输出值 (D1+0、D1+1) 存储到输出值 (静带) (D1+2、D1+3) 中。

周期	输出值 (D1+0、D1+1)	输出值 (静带) (D1+2、D1+3)
第 1 次扫描	10	10
第 2 次扫描	50	50
第 3 次扫描	100	100
第 4 次扫描	50	50
第 5 次扫描	0	0
第 6 次扫描	-50	-50
第 7 次扫描	-100	-100



## 示例 : SCALE

将模拟量输入值由 0 ~ 4,095 转换为 0 ~ 65,535 时  
创建如下梯形图程序。



- (1) 指定在 S1 中存储模拟量输入值的 D0000
- (2) 以初始化脉冲进行初始化，且 M0000 打开时，使用 SCALE 指令进行运算。

## SCALE 指令的设置



## “设备”选项卡

设置项目	说明
S1 (输入值)	D0000
S2 (控制寄存器)	D0050
S3 (初始化输入)	M8120
D1 (输出寄存器)	D0150
D2 (输出继电器)	M0050

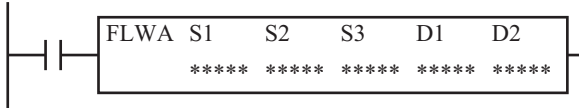
## “设置”选项卡

设置项目	说明
数据类型	字 (W)
输入最大值	4,095
输入最小值	0
输出最大值	65,535
输出最小值	0
静带	10

### FLWA（模拟流量累积）

取样瞬时流量（单位时间内的流量），并将累积流量（任意期间内流通的量）存储到日志中。

FT2J FC6A



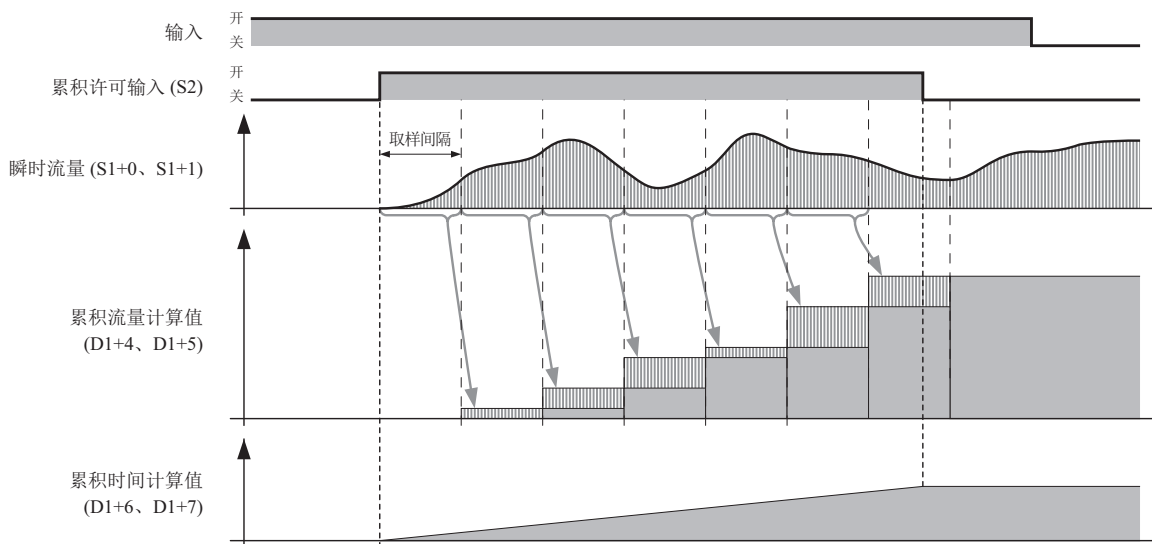
以固定周期取样瞬时流量，并计算累积流量。将其结果存储到 D2（日志数据）中指定的设备中。FLWA 指令具有取样瞬时流量进行累积的累积功能和暂存任意时间点的累积量的日志输出功能。

**注释:**

- 使用模拟量输出型流量计时，通过使用 SCALE 指令将流量计输出的模拟量值转换为瞬时流量，即可使用 FLWA 指令监视流量。  
有关 SCALE 指令的详情，请参见第 27-1 页上的“SCALE（模拟量输入转换）”。
- 使用脉冲输出型流量计时，可使用 FLWP 指令监视流量。有关 FLWP 指令的详情，请参见第 27-19 页上的“FLWP（脉冲流量累积）”。
- 对于 FT2J 型，请确保在执行 FLWA 指令之前 HMI 功能启动完成标志 (M8022) 已亮起。
- 在以下情况下会发生用户程序执行错误。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。
- 当 S1 指定的数据不符合正常浮点格式时，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。
- FLWA 指令不能在中断程序中使用。如果在中断程序中使用，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。

**累积功能的动作概要**

输入为打开时，如果打开累积许可输入（S2），则开始取样瞬时流量（S1+0、S1+1）。  
累积许可输入为打开时，进行累积处理（累积流量计算值、累积时间计算值的刷新）。  
累积许可输入为关闭时，暂时中断累积处理。（保持累积流量计算值和累积时间计算值。）



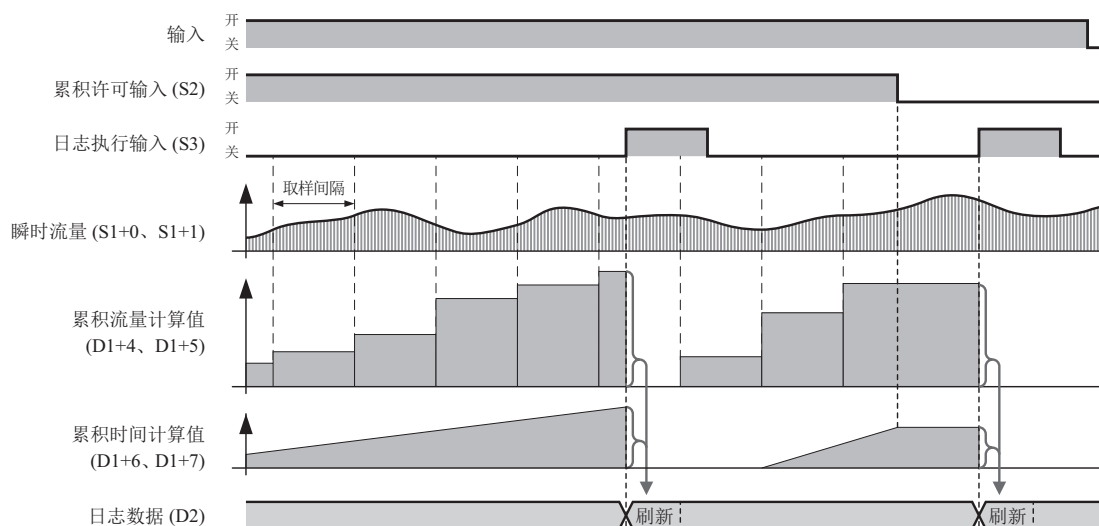
**注释:** 有关输入打开后的动作、累积许可输入（S2）重新打开时的动作等累积功能的详情，请参见第 27-12 页上的“累积功能”。



### 日志输出功能的动作概要

输入为打开时，无论累积许可输入（S2）是否打开，一旦打开日志执行输入（S3），就会刷新日志数据（D2）。日志个数最多为 35 个。

累积许可输入为关闭时，暂时中断累积处理。（保持累积流量计算值和累积时间计算值。）



**注释：**有关输入打开后的动作、累积许可输入（S2）重新打开时的动作、日志执行输入（S3）打开时的日志数据刷新处理等日志输出功能的详情，请参见第 27-14 页上的“日志输出功能”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	瞬时流量	—	—	—	—	—	—	X	—	—	—
S2（源 2）	累积许可输入	X	—	X	—	—	—	—	—	—	—
S3（源 3）	日志执行输入	X	—	X	—	—	—	—	—	—	—
D1（目标 1）	状态	—	—	—	—	—	—	▲	—	—	—
D2（目标 2）	日志数据	—	—	—	—	—	—	▲	—	—	—

▲无法使用特殊数据寄存器。

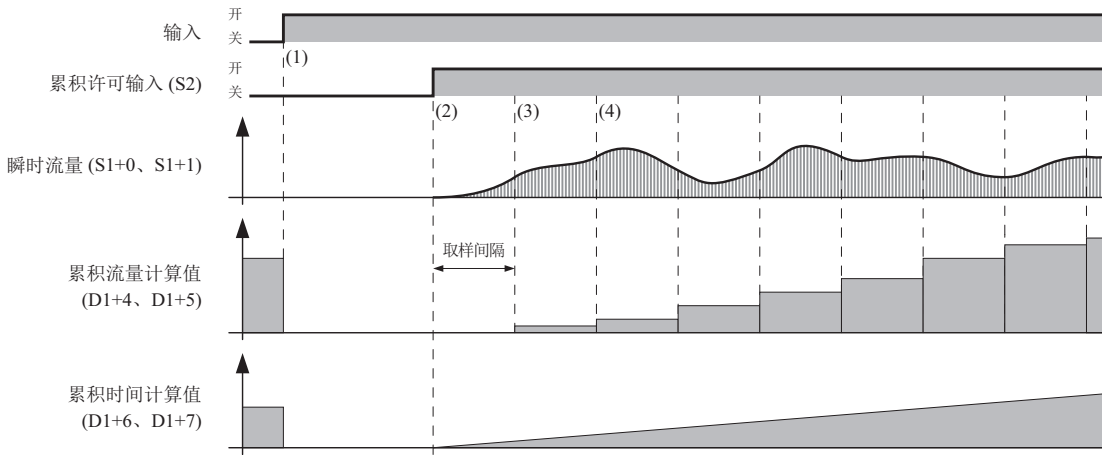
关于有效设备地址范围，请参见第 2-1 页上的“设备地址”。

功能说明

■ 累积功能

累积处理的开始

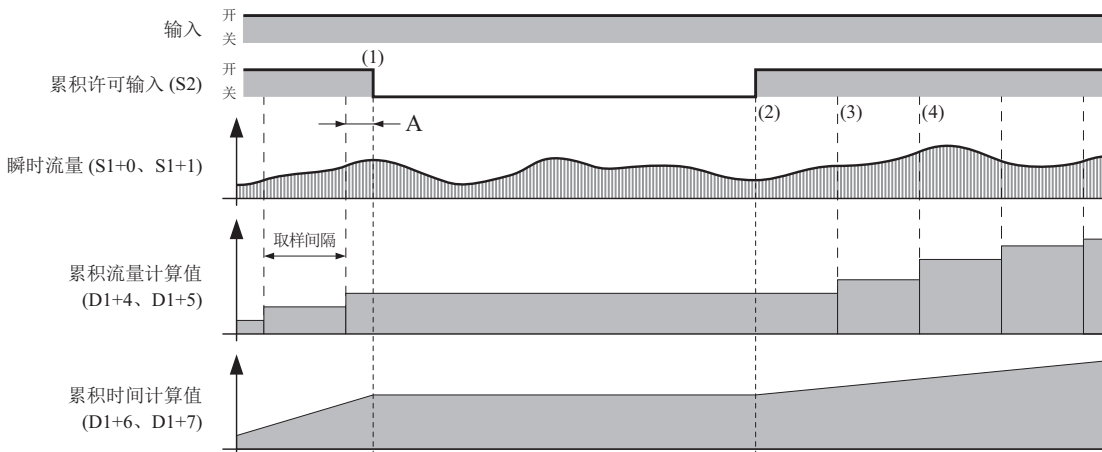
本节将对输入打开之后，到开始累积处理之前的动作进行介绍。



- (1) 输入打开时，将运行状态 (D1+0)、日志个数 (D2+0)、累积流量计算值、累积时间计算值、日志数据 (D2+2 ~ D2+2+ (6× 最大日志个数)) 初始化为“0”。
- (2) 累积许可输入 (S2) 打开时，开始进行取样。同时开始计数累积时间计算值 (D1+6、D1+7)。
- (3) 将根据取样时采集的瞬时流量 (S1+0、S1+1) 与经过时间计算的累加量与累积流量计算值 (D1+4、D1+5) 进行累加。
- (4) 之后重复 (3) 的动作。

累积处理的中断和重新开始

本节将对通过累积许可输入 (S2) 的打开 / 关闭控制累积处理的中断 / 重新开始的动作进行介绍。

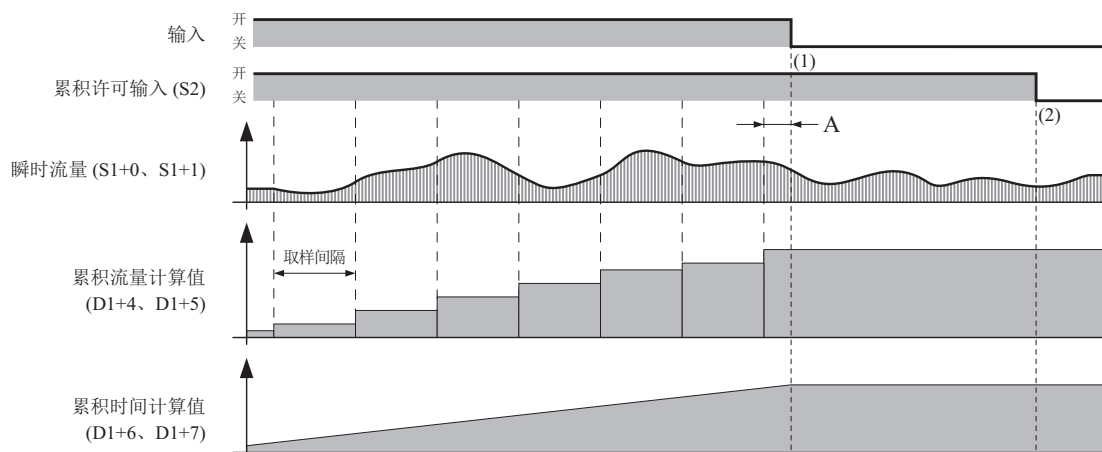


- (1) 执行累积处理过程中，如果累积许可输入关闭，则会中断取样，并停止刷新累积流量计算值 (D1+4、D1+5) 和累积时间计算值 (D1+6、D1+7)。保持累积流量计算值和累积时间计算值。  
从上次取样开始到 (1) 期间 (A) 即使瞬时流量大于“0”，也不会累加到累积流量计算值中。
- (2) 累积许可输入再次从关闭变为打开时，重新开始累积处理。  
处理与第 27-12 页上的“累积处理的开始”的 (2) 相同。
- (3) 计算取样间隔的累积流量、累积时间。  
处理与第 27-12 页上的“累积处理的开始”的 (3) 相同。
- (4) 之后重复 (3) 的动作。

## 累积处理的结束

### 关闭输入结束累积处理时

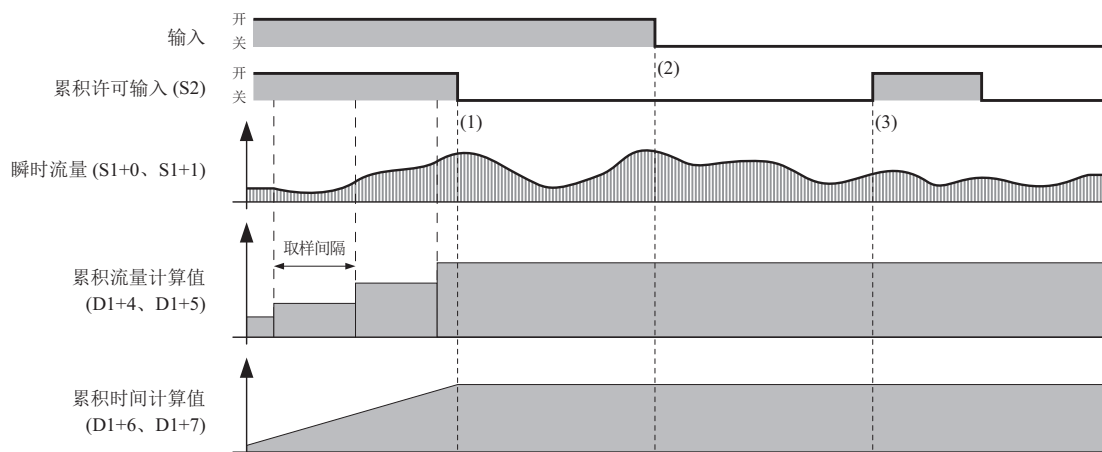
本节将对关闭输入结束累积处理时结束累积处理的动作进行介绍。



- (1) 输入为关闭时，停止刷新累积流量计算值、累积时间计算值。  
保持累积流量计算值和累积时间计算值。  
从上次取样开始到 (1) 期间 (A) 即使瞬时流量大于“0”，也不会累加到累积流量计算值中。
- (2) 输入为关闭时，即使累积许可输入关闭，也会保持累积流量计算值、累积时间计算值。

### 关闭累积许可输入 (S2) 结束累积处理时

本节将对关闭累积许可输入 (S2) 结束累积处理时结束累积处理的动作进行介绍。



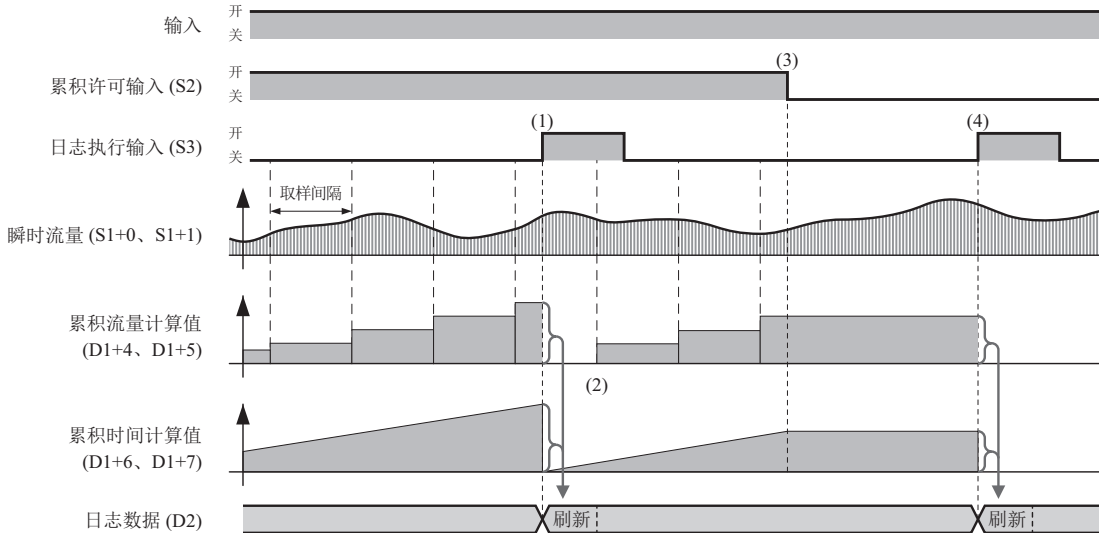
- (1) 累积许可输入为关闭时，停止累积处理（累积流量计算值、累积时间计算值的刷新）。  
保持累积流量计算值和累积时间计算值。
- (2) 即使输入关闭，(1) 的累积处理也处于停止状态，因此不会进行任何处理。  
保持累积流量计算值和累积时间计算值。
- (3) 输入为关闭时，即使累积许可输入打开，也不会进行任何处理。  
保持累积流量计算值和累积时间计算值。

■ 日志输出功能

日志输出

输入为打开时

本节将对输入为打开时的日志输出处理动作进行介绍。



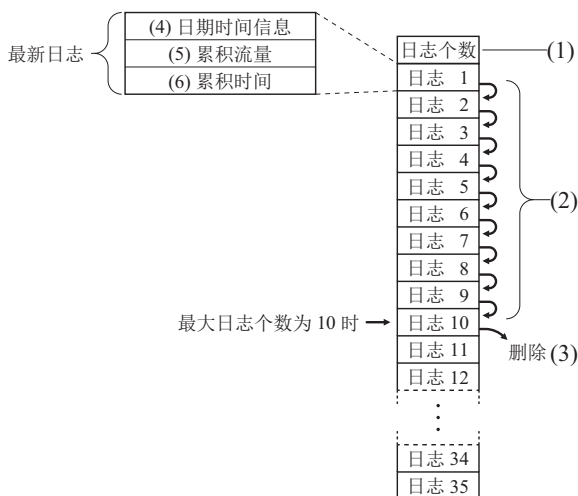
- (1) 累积许可输入 (S2) 为打开时，如果日志执行输入 (S3) 打开，则会刷新日志，并将累积流量计算值、累积时间计算值初始化为“0”。
- (2) 将根据瞬时流量 (S1+0、S1+1) 与经过时间计算的累加量与累积流量计算值进行累加。累积许可输入为打开时，即使在日志刷新处理过程中，也会保持取样间隔。
- (3) 累积许可输入为关闭时，中断取样，并停止刷新累积流量计算值和累积时间计算值。保持累积流量计算值和累积时间计算值。
- (4) 即使在累积许可输入为关闭时，只要输入打开，就会进行与 (1) 相同的日志输出处理。

输入为关闭时

输入为关闭时，即使日志执行输入 (S3) 打开，也不会进行任何处理。

日志刷新处理

日志数据可记录日志 1 ~ 日志 35。日志 1 为最新日志数据。  
 每次刷新日志，都会对日志个数 (1) 累加 1，且旧日志数据逐个向后排列 (2)。  
 已达到最大日志个数时，删除最旧日志数据 (3)。  
 将在日志 1 中存储日期时间信息 (4)、累积流量 (5)。  
 在日志 1 中将累积时间 (6) 的单位从“ms”转换为“s”进行存储。



使用的最大日志个数，在“FLWA (模拟流量累积)”对话框的“设置”选项卡中进行指定。

**注释:** 有关日志中记录的日期时间信息、累积流量、累积时间的详情，请参见第 27-16 页上的“(5) D2 (目标 2): 记录的数据”。

## 设置项目

“FLWA（模拟流量积算仪）”对话框中包括 2 个选项卡，分别为“设备”选项卡、“设置”选项卡。

### ■ “设备”选项卡

设置项目	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示标记名称对应的设备地址。
注释	显示设备地址的注释。可编辑。

#### (1) S1（源 1）：瞬时

指定存储瞬时流量的设备。以设置的设备为起始，连续占用 2 个字（S1+0、S1+1）。有效的瞬时流量范围为 0 或  $1.175494E-38 \sim 3.402823E+38$ 。

##### 注释：

- 在设备中存储 32 位数据的方法取决于控制器类型。有关详情，请参见第 3-10 页上的“32 位数据存储”。
- 瞬时流量的值超出有效值范围时，将导致用户程序执行错误，将存储错误代码 28 在用户程序执行错误代码 (D8006) 中。而且同时在状态 (D1) 中存储 10。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

#### (2) S2（源 2）：启用积算仪输入

指定允许瞬时流量的累积处理的设备。  
累积许可输入为关闭时中断累积处理，为打开时则执行累积处理。  
有关累积许可输入的详情，请参见第 27-12 页上的“功能说明”。

#### (3) S3（源 3）：日志执行输入

指定刷新日志、执行 / 停止将累积流量计算值、累积时间计算值初始化为“0”的日志处理的设备。  
日志执行输入从关闭变为打开时，执行日志处理。  
有关日志执行输入的详情，请参见第 27-14 页上的“日志输出功能”。

#### (4) D1（目标 1）：状态

指定存储执行 FLWA 指令时的错误等状态的设备。以设置的设备为起始，连续占用 10 个字。

存储目的地	功能	说明
D1+0	运行状态	存储执行 FLWA 指令时的错误等状态。 0：正常 10：取样中的瞬时流量超出范围
D1+1 ⋮ D1+9	系统工作区	

## 27: 流量计算指令

### (5) D2（目标 2）：记录的数据

指定存储日志处理后数据的设备。以设置的设备为起始，连续占用最多 212 个字。

**注释：** 日志数据占用的数据寄存器个数根据最大日志个数而变化。

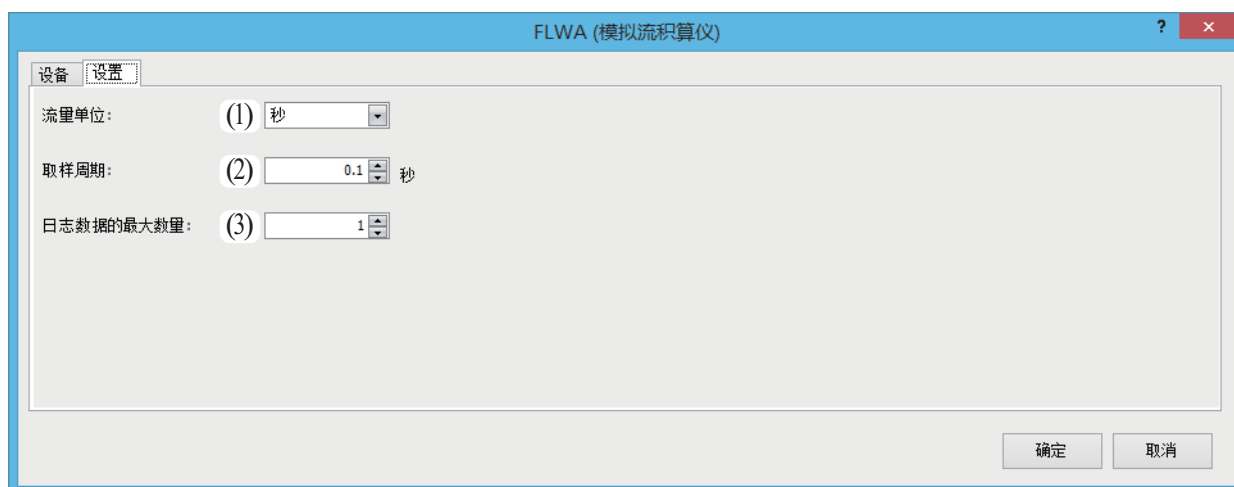
日志数据占用的数据寄存器数为  $2 + 6 \times$  最大日志个数。

存储目的地	功能		说明	数据范围
D2+0	日志个数		存储记录完的日志个数。	0 ~ 35: W (字)
D2+1	保留		—	—
D2+2 D2+3	日志 1	日志时间	日志执行输入 (S3) 从关闭变为打开时 (日志刷新处理时), 将特殊数据寄存器 D8008 ~ D8014 (日历/时钟 当前值) 的内容转换为 UNIX 时间 *1 进行存储。	0 ~ 4,294,967,295*2 : D (双字)
D2+4 D2+5		累积流量	日志执行输入 (S3) 从关闭变为打开时 (日志刷新处理时), 存储累积流量。 有关累积流量上溢出或下溢出时的处理详情, 请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。	0、 1.175494E-38 ~ 3.402823E+38*2 : F (浮点)
D2+6 D2+7		累积时间	日志执行输入 (S3) 从关闭变为打开时 (日志刷新处理时), 存储累积时间。 单位为“s”。	0 ~ 4,294,967,295*2 : D (双字)
⋮	⋮	⋮	⋮	⋮
D2+206 D2+207	日志 35	日志时间	与日志 1 相同。	与日志 1 相同。
D2+208 D2+209		累积流量		
D2+210 D2+211		累积时间		

\*1 自 1970 年 1 月 1 日 (UTC+0) 起不含闰秒的经过时间 (秒)

\*2 在设备中存储 32 位数据的方法取决于控制器类型。有关详情, 请参见第 3-10 页上的“32 位数据存储”。

## ■ “设置”选项卡

**(1) 流量单位**

瞬时流量（S1+0、S1+1）中存储的瞬时流量时间单位从以下内容中进行选择。

“秒”、“分钟”、“小时”、“日”

**(2) 取样间隔**

以 0.1s 为单位指定取样瞬时流量的时间间隔。

设置范围为 0.1 ~ 6,553.5。

**(3) 日志数据的最大数量**

指定可记录日志个数的最大值。

日志数据（D2）占用的数据寄存器个数根据最大日志个数而变化。

日志数据（D2）占用的数据寄存器数为  $2 + 6 \times$  最大日志个数。

设置范围为 1 ~ 35。

## 27: 流量计算指令

### 示例：FLWA

将 FC6A 型的模拟量输入值转换为瞬时流量。对瞬时流量进行采样和累积。每天 0:00 时临时保存累计体积。

### 设置示例



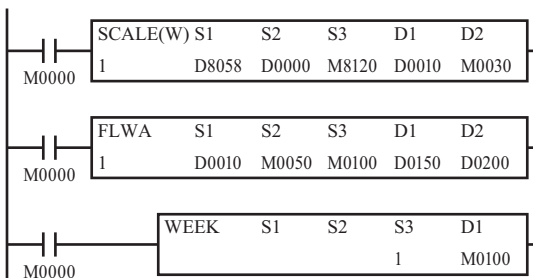
#### “设备”选项卡

项目	说明	备注
S1 (流量)	D0010	由模拟量输入值转换为瞬时流量的值。
S2 (启用积算仪输入)	M0050	
S3 (日志执行输入)	M0100	WEEK 指令的输出。设置为每天 00:00 打开 1 次扫描。
D1 (状态)	D0150	
D2 (记录的数据)	D0200	

#### “设置”选项卡

项目	说明	备注
流量单位	秒	
取样间隔	0.1s	
日志数据的最大数量	10	

### 程序示例



- 打开输入 M0000 时，将模拟量输入值 D8058\*1 转换为瞬时流量的结果，并在每次扫描时存储到 D0010 中。
- 通过打开输入 M0000，并打开累积许可输入 M0050，开始取样瞬时流量 D0010 并进行累积处理。
- 动作状况会存储到 D1 (状态) 中。
- 通过使用 WEEK 指令在每天 00:00 打开 1 次扫描 M0100，来打开 S3 (日志执行输入)，并将当前日志信息存储到 D2 (日志数据) 中。

\*1 存储内置模拟输入的特殊数据寄存器取决于控制器类型。有关详情，请参见第 2-1 页上的“设备地址”。



## FLWP（脉冲流量累积）

监视测量脉冲数的计数器，并以固定周期计算瞬时流量。而且将累积流量（任意期间内流通的量）存储到日志中。

FT2J FC6A



通过使用计数器读取脉冲输出型流量计输出的脉冲信号，然后监视计数器的值，来计算瞬时流量及累积流量。将其结果存储到 D1（状态）、D2（日志数据）中指定的设备中。

FLWP 指令具有通过脉冲计算瞬时流量的瞬时流量计算功能、累积流量的累积功能和暂存任意时间点的累积流量的日志输出功能。

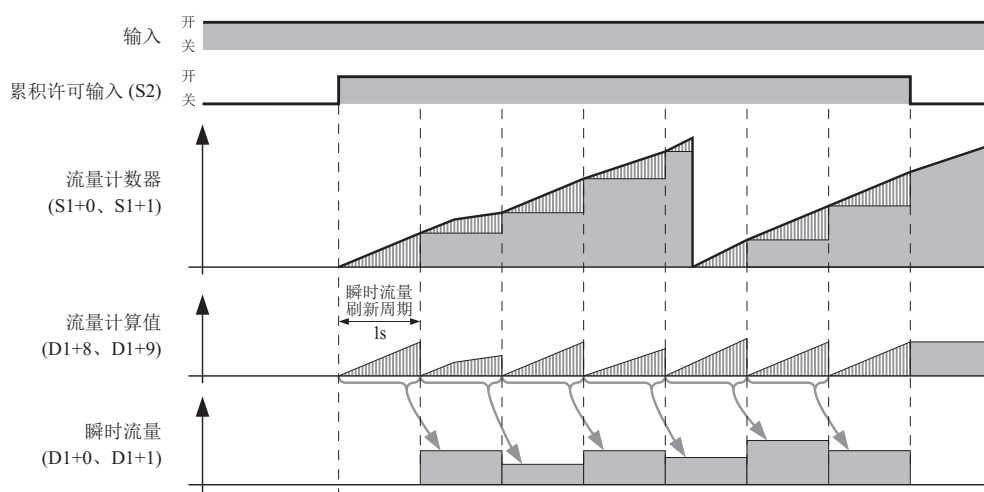
### 注释：

- 使用模拟量输出型流量计时，可使用 FLWA 指令监视流量。有关 FLWA 指令的详情，请参见第 27-10 页上的“FLWA（模拟流量累积）”。
- FLWP 指令支持的脉冲最大频率为 10kHz。
- 请以 119 小时以内的周期执行日志输出。
- 对于 FT2J 型，请确保在执行 FLWP 指令之前 HMI 功能启动完成标志 (M8022) 已亮起。
- FLWP 指令不能在中断程序中使用。如果在中断程序中使用，将会导致用户程序执行错误，将存储错误代码 18 在用户程序执行错误代码 (D8006) 中。有关用户程序执行错误的详情，请参见第 3-11 页上的“用户程序执行错误”。

### 瞬时流量计算功能的动作概要

输入为打开时，如果累积许可输入（S2）打开，则每次扫描都会将流量计数器（S1+0、S1+1）的增加量累加到流量计算值中。而且会根据每 1s 通过流量计算值测量出的结果刷新瞬时流量（D1+0、D1+1）。

累积许可输入关闭时，将瞬时流量初始化为“0”，并停止刷新瞬时流量。



**注释：**有关输入打开后的动作、累积许可输入重新打开时的动作等流量计算功能的详情，请参见第 27-21 页上的“瞬时流量计算功能”。

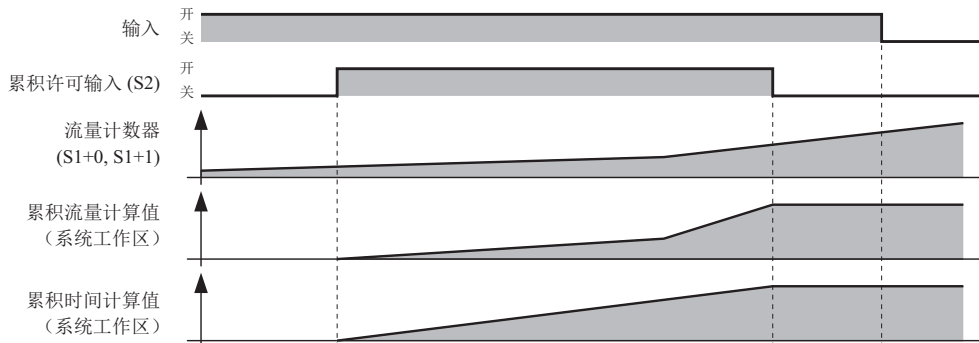
## 27: 流量计算指令

### 累积功能的动作概要

输入为打开时，如果累积许可输入（S2）打开，则开始进行累积处理。

累积许可输入为打开时，每次扫描都会将流量计数器（S1+0、S1+1）的增加量累加到累积流量计算值中。同时开始计数累积时间计算值。

累积许可输入为关闭时，暂时中断累积处理。（保持累积流量计算值和累积时间计算值。）

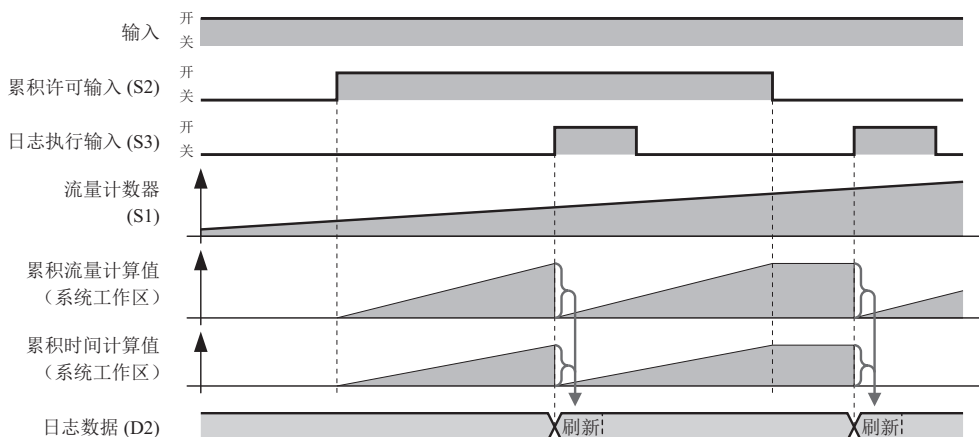


**注释：**有关输入打开后的动作、累积许可输入重新打开时的动作等累积功能的详情，请参见第 27-23 页上的“累积功能”。

### 日志输出功能的动作概要

输入为打开时，无论累积许可输入（S2）是否打开，一旦打开日志执行输入（S3），就会刷新日志数据（D2）。日志个数最多为 35 个。

累积许可输入为关闭时，暂时中断累积处理。保持累积流量计算值和累积时间计算值。



**注释：**有关输入打开后的动作、累积许可输入（S2）重新打开时的动作、日志执行输入（S3）打开时的日志数据刷新处理等日志输出功能的详情，请参见第 27-24 页上的“日志输出功能”。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	流量计数器	—	—	—	—	—	—	X	—	—	—
S2（源 2）	累积许可输入	X	—	X	—	—	—	—	—	—	—
S3（源 3）	日志执行输入	X	—	X	—	—	—	—	—	—	—
D1（目标 1）	状态	—	—	—	—	—	—	▲	—	—	—
D2（目标 2）	日志数据	—	—	—	—	—	—	▲	—	—	—

▲无法使用特殊数据寄存器。

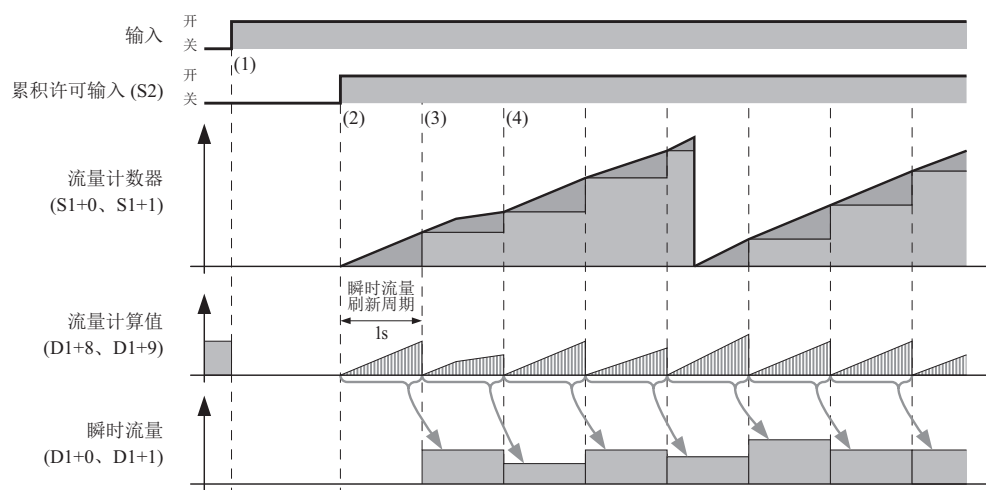
关于有效设备编号范围，请参见第 2-1 页上的“设备地址”。

## 功能说明

### ■ 瞬时流量计算功能

#### 瞬时流量计算的开始

本节将对输入打开之后，开始瞬时流量计算之前的动作进行介绍。



- (1) 输入打开时，将流量计算值及瞬时流量（D1+0、D1+1）初始化为“0”。
- (2) 打开累积许可输入（S2）时，开始对流量计数器（S1+0、S1+1）进行取样处理，且每次扫描都会将流量计数器的增加量累加到流量计算值中。
- (3) 累积许可输入为打开时，通过流量计算值计算每 1s 的流量，并存储到瞬时流量中。无论日志执行输入（S3）是否打开，都会以 1s 为周期执行瞬时流量计算。
- (4) 之后重复 (2) 及 (3) 的动作。

**注释：** 瞬时流量计算公式如下所示。

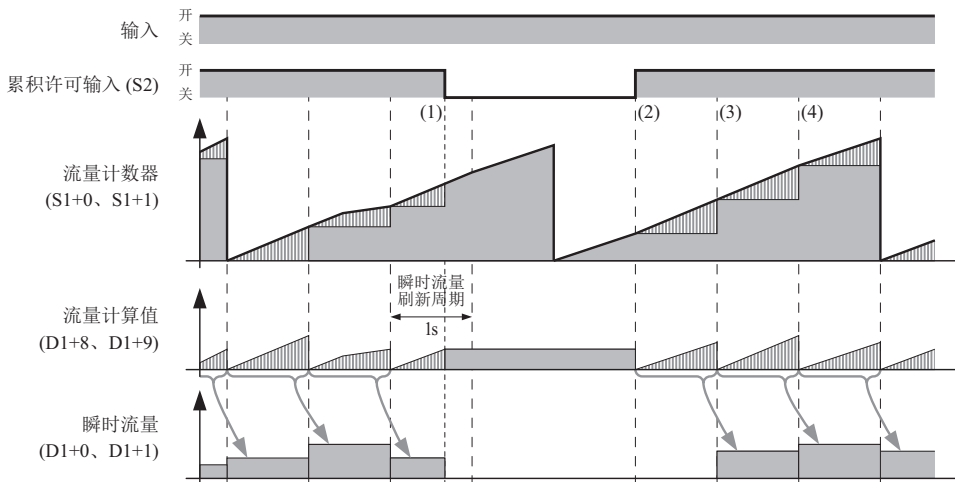
$$\text{瞬时流量} = \frac{\text{流量计算值}}{\text{K 因子}} \times T$$

T 的值根据设置的瞬时流量单位如下所示。

T (秒)	= 1
T (分)	= 60
T (小时)	= 60×60 = 360
T (天)	= 60×60×24 = 8,640

### 瞬时流量计算的中断

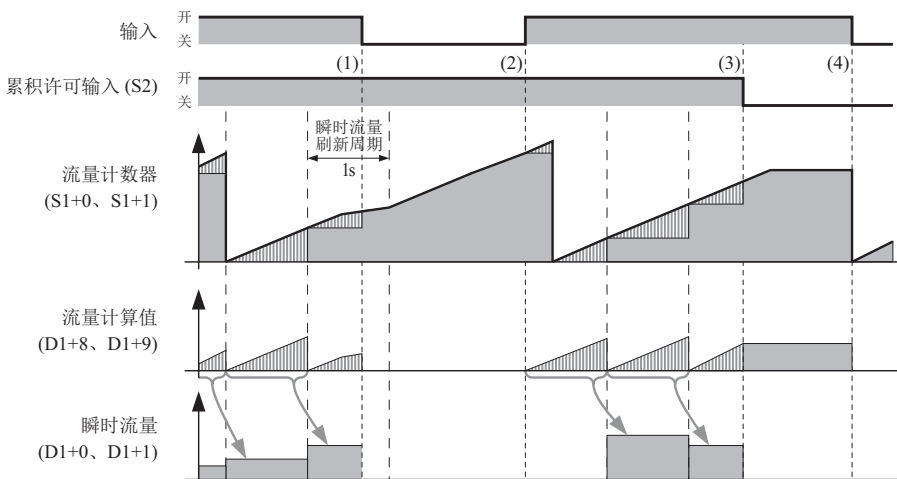
本节将对输入打开之后，中断瞬时流量计算之前的动作进行介绍。



- (1) 关闭累积许可输入 (S2) 时，将瞬时流量初始化为“0”。
- (2) 打开累积许可输入 (S2) 时，开始对流量计数器 (S1+0、S1+1) 进行取样处理，且每次扫描都会将流量计数器 (S1) 的增加量累加到流量计算值中。处理与第 27-21 页上的“瞬时流量计算的开始”的 (2) 相同。
- (3) 累积许可输入打开后通过流量计算值计算 1s 后的瞬时流量，并存储到瞬时流量 (D1+0、D1+1) 中。无论日志执行输入 (S3) 是否打开，都会以 1 秒为周期计算瞬时流量 (处理与第 27-21 页上的“瞬时流量计算的开始”的 (3) 相同)。
- (4) 之后重复 (2) 及 (3) 的动作。

### 瞬时流量计算的结束

本节将对结束瞬时流量计算之前的动作进行介绍。

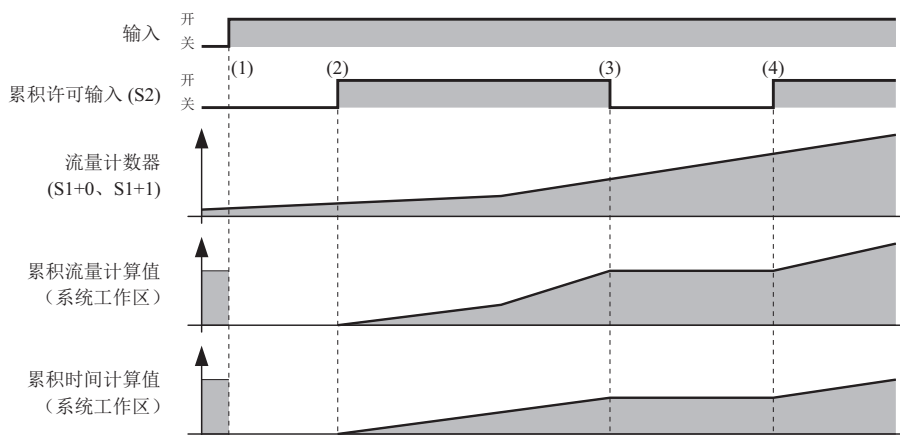


- (1) 在累积许可输入 (S2) 为打开的状态下输入关闭时，将流量计算值和瞬时流量初始化为“0”，并结束指令。
- (2) 在累积许可输入为打开的状态下输入打开时，开始瞬时流量计算。
- (3) 累积许可输入关闭时，将瞬时流量初始化为“0”，并中断瞬时流量计算 (处理与第 27-22 页上的“瞬时流量计算的中断”的 (1) 相同)。
- (4) 输入关闭时，将流量计算值和瞬时流量 (D1+0、D1+1) 初始化为“0”，并结束指令。

## ■ 累积功能

### 累积处理的开始和中断

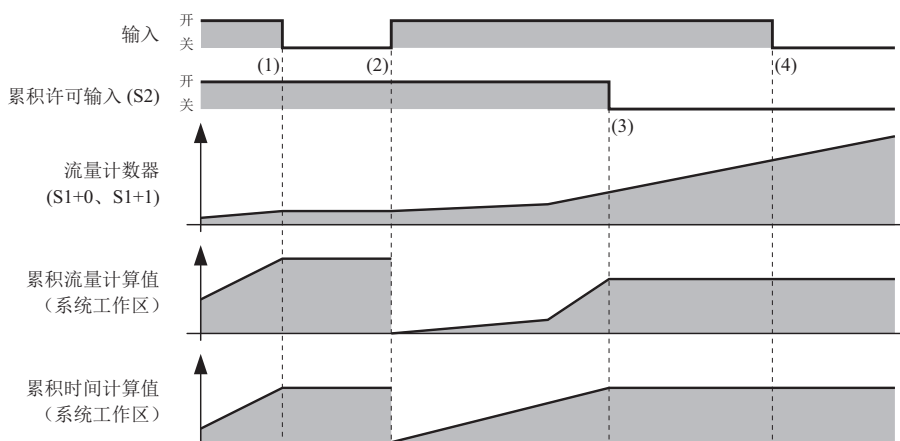
本节将对输入打开之后，开始累积处理然后中断之前的动作进行介绍。



- (1) 输入打开时，将运行状态（D1+2）、日志个数（D2）、累积流量计算值、累积时间计算值、日志数据（D2+2 ~ D2+2+（6×最大日志个数））初始化为“0”。
- (2) 在输入为打开的状态下打开累积许可输入（S2）时，开始对流量计数器进行取样处理，且每次扫描都会将流量计数器的增加量累加到累积流量计算值中。同时开始测量累积时间计算值。
- (3) 关闭累积许可输入时，暂停刷新累积流量计算值和累积时间计算值。
- (4) 累积许可输入重新从关闭变为打开时，不会初始化累积流量计算值和累积时间计算值，而是进行与(2)相同的动作。

### 累积处理的结束

本节将对结束累积处理的动作进行介绍。



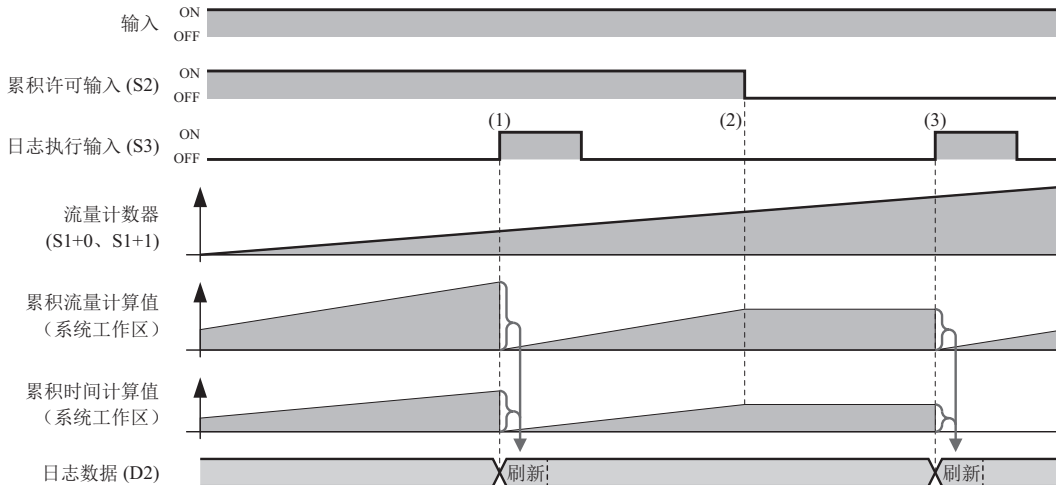
- (1) 即使累积许可输入（S2）为打开状态，一旦输入关闭，就会停止刷新累积流量计算值和累积时间计算值。保持累积流量计算值和累积时间计算值。
- (2) 打开输入时，将累积流量计算值和累积时间计算值初始化为“0”。  
此时，如果累积许可输入处于打开状态，将开始刷新累积流量计算值和累积时间计算值。
- (3) 关闭累积许可输入时，暂停刷新累积流量计算值和累积时间计算值。  
处理与第 27-23 页上的“累积处理的开始和中断”的(3)相同。
- (4) 关闭输入时，停止累积处理。保持累积流量计算值和累积时间计算值。

■ 日志输出功能

日志输出

输入为打开时

本节将对输入为打开时的日志输出处理动作进行介绍。



- (1) 累积许可输入 (S2) 为打开时，如果日志执行输入 (S3) 打开，则会刷新日志，并将累积流量计算值、累积时间计算值初始化为“0”。
- (2) 累积许可输入关闭时，中断取样，并停止刷新累积流量计算值和累积时间计算值。保持累积流量计算值和累积时间计算值。
- (3) 即使累积许可输入为关闭时，只要输入打开，就会进行与 (1) 相同的日志输出处理。

输入为关闭时

输入为关闭时，即使日志执行输入 (S3) 打开，也不会进行任何处理。

日志刷新处理

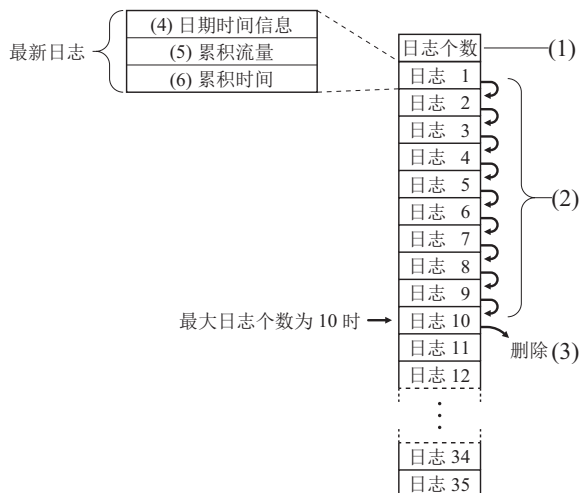
日志数据可记录日志 1 ~ 日志 35。日志 1 为最新日志数据。

每次刷新日志，都会对日志个数 (1) 累加 1，且旧日志数据逐个向后排列 (2)。

已达到最大日志个数时，删除最旧日志数据 (3)。

将在日志 1 中存储日期时间信息 (4)、累积流量 (5)。

在日志 1 中将累积时间 (6) 的单位从“ms”转换为“s”进行存储。



使用的最大日志个数，在“FLWP（脉冲流量累积）”对话框的“设置”选项卡中进行指定。

**注释：**有关日志中记录的日期时间信息、累积流量、累积时间的详情，请参见第 27-26 页上的“(5) D2（目标 2）：记录的数据”。

## 设置项目

“FLWP（脉冲流积算仪）”对话框中包括 2 个选项卡，分别为“设备”选项卡、“设置”选项卡。

### ■ “设备”选项卡



设置项目	说明
标记名称	指定设备的标记名称或设备地址。
设备地址	显示标记名称对应的设备地址。
注释	显示设备地址的注释。可编辑。

#### (1) S1（源 1）：流量计器

流量计数器根据第 27-27 页上的“(2) 计数器类型”的动作类型不同，设置内容也有所不同。以设置的设备为起始，连续占用对应数据类型数量的设备。

##### “计数器类型”为“自由振荡”时

指定存储持续累加脉冲的计数器值的设备。要求取每次扫描的计数数量时，根据上一次扫描的流量计数器与最新流量计数器之差进行计算。

##### “计数器类型”为“差分”时

指定存储每次扫描的脉冲计数数量的设备。

流量计器的设置范围和使用的数据寄存器数量取决于在“设置”选项卡中选择的\*1，如下所示。

数据类型	设置范围	使用的数据寄存器
字 (W)	0 ~ 65,535	1
双字 (D)	0 ~ 4,294,967,295	2

\*1 有关流量计器的计数器类型和数据类型的详情，请参见第 27-27 页上的“(2) 计数器类型”和第 27-27 页上的“(1) 数据类型”。

\*2 在设备中存储 32 位数据的方法取决于控制器类型。有关详情，请参见第 3-10 页上的“32 位数据存储”。

#### (2) S2（源 2）：启用积算仪输入

设置允许累积流量的设备。

累积许可输入为关闭时中断累积处理，为打开时则执行累积处理。

有关累积许可输入的详情，请参见第 27-21 页上的“功能说明”。

#### (3) S3（源 3）：日志执行输入

指定刷新日志、执行 / 停止将累积流量计算值、累积时间计算值初始化为“0”的日志处理的设备。

日志执行输入从关闭变为打开时，执行日志处理。

执行 FLWP 指令过程中，请在执行上次日志处理后的 119 小时以内，执行下一次的日志处理。

有关日志执行输入的详情，请参见第 27-24 页上的“日志输出功能”。

## 27: 流量计算指令

### (4) D1 (目标 1): 状态

存储执行瞬时流量或 FLWP 指令时的错误等状态。以设置的设备为起始，连续占用 16 个字。

存储目的地	功能	说明	存储值范围
D1+0 D1+1	瞬时流量	存储流量单位（第 27-27 页上的“(3) 流量单位”）中设置的每个时间单位的计算瞬时流量。 瞬时流量的单位将遵照 K 因子及流量单位的设置。	1.175494E-38 ~ 3.402823E+38*1: F (浮点)
D1+2	运行状态	存储执行 FLWP 指令时的错误等状态。	
D1+3 ⋮ D1+15	系统工作区		

\*1 在设备中存储 32 位数据的方法取决于控制器类型。有关详情，请参见第 3-10 页上的“32 位数据存储”。

### (5) D2 (目标 2): 记录的数据

指定存储日志处理后数据的设备。以设置的设备为起始，连续占用最多 212 个字。

**注释:** 日志数据占用的数据寄存器个数根据最大日志个数而变化。

日志数据占用的数据寄存器数为  $2 + 6 \times$  最大日志个数。

存储目的地	功能	说明	存储值范围
D2+0	日志个数	存储记录完的日志个数。	0 ~ 35: W (字)
D2+1	保留	—	—
D2+2 D2+3	日志 1	日志时间	日志执行输入 (S3) 从关闭变为打开时 (日志刷新处理时)，将特殊数据寄存器 D8008 ~ D8014 (日历/时钟 当前值) 的内容转换为 UNIX 时间*1 进行存储。
D2+4 D2+5		累积流量	日志执行输入 (S3) 从关闭变为打开时 (日志刷新处理时)，存储累积流量。 有关累积流量上溢出或下溢出时的处理详情，请参见第 3-9 页上的“浮点型数据处理中的进位和借位”。
D2+6 D2+7		累积时间	日志执行输入 (S3) 从关闭变为打开时 (日志刷新处理时)，存储累积时间。 单位为“s”。
⋮	⋮	⋮	⋮
D2+206 D2+207	日志 35	日志时间	与日志 1 相同。
D2+208 D2+209		累积流量	
D2+210 D2+211		累积时间	
			与日志 1 相同。

\*1 自 1970 年 1 月 1 日 (UTC+0) 起不含闰秒的经过时间 (秒)

\*2 在设备中存储 32 位数据的方法取决于控制器类型。有关详情，请参见第 3-10 页上的“32 位数据存储”。



## ■ 设置选项卡



### (1) 数据类型

从“字 (W)”或“双字 (D)”中选择流量计数器的数据类型。初始值为“双字 (D)”。

字 (W): 将 S1+0 作为流量计数器进行操作。

双字 (D): 将 S1+0、S1+1 作为流量计数器进行操作。

### (2) 计数器类型

从“自由振荡”或“差分”中选择要取样的流量计数器的动作类型。

自由振荡: 流量计数器不进行初始化, 而是进行只有累加的自由振荡动作。自由振荡动作是指超过数据类型的上限 (65,535 (W)、4,294,967,295 (D)) 后返回 0 持续计数的动作。

差分: 将每次扫描的脉冲计数数量存储到流量计数器中。

例如, 用于通过 2 种不同的计数器计数脉冲, 并使用梯形图等计算其差值, 然后直接存储到流量计数器。

### (3) 流量单位

瞬时流量 (D1+0、D1+1) 中存储的瞬时流量时间单位从以下内容中进行选择。

“秒”、“分钟”、“小时”、“日”

### (4) K 系数

指定表示流量计输出的每个单位量的脉冲数的常量。

单位因流量计而异 (升、加仑等), 但 FLWP 指令中动作不受该单位的影响。

例如, 流量计为流通 1 升后输出 10 脉冲的类型时, K 因子指定 10“脉冲 / 升”。

设置范围为  $1.175494\text{E}-38 \sim 3.402823\text{E}+38$ 。

### (5) 日志数据的最大数量

指定可记录日志个数的最大值。

日志数据 (D2) 占用的数据寄存器个数根据最大日志个数而变化。

日志数据 (D2) 占用的数据寄存器数为  $2 + 6 \times \text{最大日志个数}$ 。

设置范围为 1 ~ 35。

## 27: 流量计算指令

### 示例：FLWP

使用 FC6A 型累加流速，并在每天 0:00 临时保存累加流速。

### 设置示例



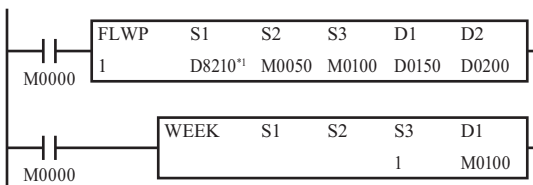
#### “设备”选项卡

项目	说明	备注
S1 (计数器)	D8210 <sup>*1</sup>	G1 高速计数器的当前值。
S2 (启用积算仪输入)	M0050	
S3 (日志执行输入)	M0100	WEEK 指令的输出。设置为每天 00:00 打开 1 次扫描。
D1 (状态)	D0150	
D2 (记录的数据)	D0200	

#### “设置”选项卡

项目	说明	备注
数据类型	双字 (D)	
计数器类型	自由振荡	
流量单位	秒	
K 系数	5.5344	
日志数据的最大数量	10	

### 程序示例



- 通过打开输入 M0000，并打开 S2 (累积许可输入)，开始进行累积。
- 将累积的值与流量计算的动作状况一同存储到 D1 (状态) 中。
- 通过使用 WEEK 指令在每天 00:00 打开 1 次扫描 M0100，来打开 S3 (日志执行输入)，并将当前日志信息存储到 D2 (日志数据) 中。

\*1 存储高速计数器当前值的特殊数据寄存器取决于控制器类型。请参见第 2-1 页上的“设备地址”。

# 28: 用户定义宏指令

本章将对执行已注册用户定义宏的指令进行介绍。

对主程序内多次使用的梯形图程序附加任意编号并命名进行注册的宏称为用户定义宏。有关创建用户定义宏的详情，请参见「用户定义宏的注册步骤」（28-6 页）。

有关用户定义宏与子程序的差异，请参见附录 -2 页上的“用户定义宏与子程序的差异”。

## UMACRO（用户定义宏）

执行指定编号的用户定义宏。

FT2J FC6A

### 符号



### 操作

输入为 ON 时，将执行指定用户定义宏名称对应的用户定义宏。此外，输入为 OFF 时，不会执行指定用户定义宏名称对应的用户定义宏。

UMACRO 指令可设置用户定义宏中所使用的参数设备。参数设备为执行 UMACRO 指令时用户定义宏实际所使用的设备。例如，A1 设置为 I0001 时，用户定义宏中 A0001 将作为 I0001 进行动作。

**注释：**用户定义宏中不能使用 UMACRO、JMP、JEND、MCR、MCS、LCAL、LJMP、LRET、LABEL、DJNZ 和 END 指令。

### 有效设备

设备	功能	I	Q	M	R	T	C	D	P	常量	重复
S1（源 1）	用户定义宏名称	—	—	—	—	—	—	—	—	X <sup>*1</sup>	—
A1 ... An <sup>*2</sup>	参数设备 1 ~ n (n 需满足 1 ≤ n ≤ 100)	X	X	X	X	X	X	X	—	X	—

\*1 S1（用户定义宏名称）的有效范围为 0 ~ 255。

\*2 A1 ~ A100 为用户定义宏所使用的参数设备。所使用的参数设备可通过第 28-4 页上的““参数设置”对话框”进行设置，并能使用 0 ~ 100 个参数设备。

例如，在第 28-4 页上的““参数设置”对话框”中将“参数数量”选择为 5 时，可使用 A1 ~ A5 的参数设备，若在第 28-4 页上的““参数设置”对话框”中将“参数数量”选择为 0 时，则无法使用参数设备。

## 28: 用户定义宏指令

### 本地设备

本地设备是指仅可在用户定义宏指令内使用的字设备。通过使用本地设备，将易于重新利用至其他项目，无需更改宏内使用的设备。

设备	符号	单位	范围（点数）
本地设备	@	字	@0 ~ @127 (128 点)

#### 注释:

- @0 ~ @127 将在执行用户定义宏指令时全部初始化为 0。
- 本地设备不支持 WindLDR 的监控、模拟。

可使用本地设备的指令如下所示。

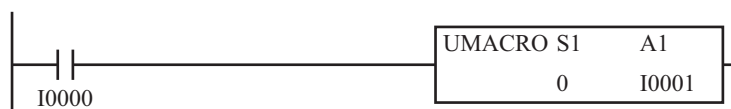
指令	支持	指令	支持	指令	支持	指令	支持	指令	支持
LOD	—	MCS	—	ROOT	X	SWAP	X	ABS	—
LODN	—	MCR	—	SUM	X	TXD	—	JOG	—
AND	—	JMP	—	RNDM	X	RXD	—	DI	—
ANDN	—	JEND	—	RAD	X	ETXD	—	EI	—
OR	—	END	—	DEG	X	ERXD	—	XYFS	—
ORN	—	NOP	—	SIN	X	LABEL	—	CVXTY	—
AND LOD	—	MOV	X	COS	X	LJMP	—	CVYTX	—
OR LOD	—	MOVN	X	TAN	X	LCAL	—	PID	—
BPS	—	IMOV	X	ASIN	X	LRET	—	PIDA	—
BRD	—	IMOVN	X	ACOS	X	DJNZ	—	PIDD	—
BPP	—	IBMV	X	ATAN	X	DISP	—	AVRG	—
OUT	—	IBMVN	X	LOGE	X	DGRD	—	RUNA	—
OUTN	—	BMOV	X	LOG10	X	MSG	—	STPA	—
SET	—	MOVC	X	EXP	X	IOREF	—	FIFO	—
RST	—	NSET	X	POW	X	HSCRFB	—	FIEX	X
TML	—	NRS	X	ANDW	X	FRQRF	—	FOEX	X
TIM	—	XCHG	X	ORW	X	COMRF	—	NDSRC	—
TMH	—	TCCST	X	XORW	X	WKTBL	—	EMAIL	—
TMS	—	CMP=	X	SFTL	X	WKTIM	—	PING	—
TMLO	—	CMP<>	X	SFTR	X	WEEK	—	SCRPT	—
TIMO	—	CMP<	X	ROTL	X	YEAR	—	DLOG	—
TMHO	—	CMP>	X	ROTR	X	TADD	X	TRACE	—
TMSO	—	CMP<=	X	BCDLS	X	TSUB	X	SCALE	—
CNT	—	CMP>=	X	WSFT	X	HOUR	—	FLWP	—
CDP	—	ICMP>=	X	HTOB	X	HTOS	X	FLWA	—
CUD	—	LC=	X	BTOH	X	STOH	X		
CNTD	—	LC<>	X	HTOA	X	DTML	—		
CDPD	—	LC<	X	ATOH	X	DTIM	—		
CUDD	—	LC>	X	BTOA	X	DTMH	—		
CC=	X	LC<=	X	ATOB	X	DTMS	—		
CC ≧	X	LC>=	X	ENCO	X	TTIM	—		
DC=	X	ADD	X	DECO	X	PULS	—		
DC ≧	X	SUB	X	BCNT	X	PWM	—		
SFR	—	MUL	X	ALT	—	RAMP	—		
SFRN	—	DIV	X	CVDT	X	RAMPL	—		
SOTU	—	INC	X	DTDV	X	ZRN	—		
SOTD	—	DEC	X	DTCB	X	ARAMP	—		

### 操作示例

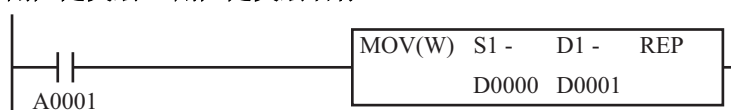
下述梯形图程序会在主程序的输入 I0000 为 ON 时执行 UMACRO 指令。执行 UMACRO 指令后，将执行用户定义宏名称 0 的用户定义宏。

由于 UMACRO 指令的 A1 设置为 I0001，用户定义宏中 A0001 将作为 I0001 进行动作。因此，I0001 为 ON 时执行 MOV 指令。

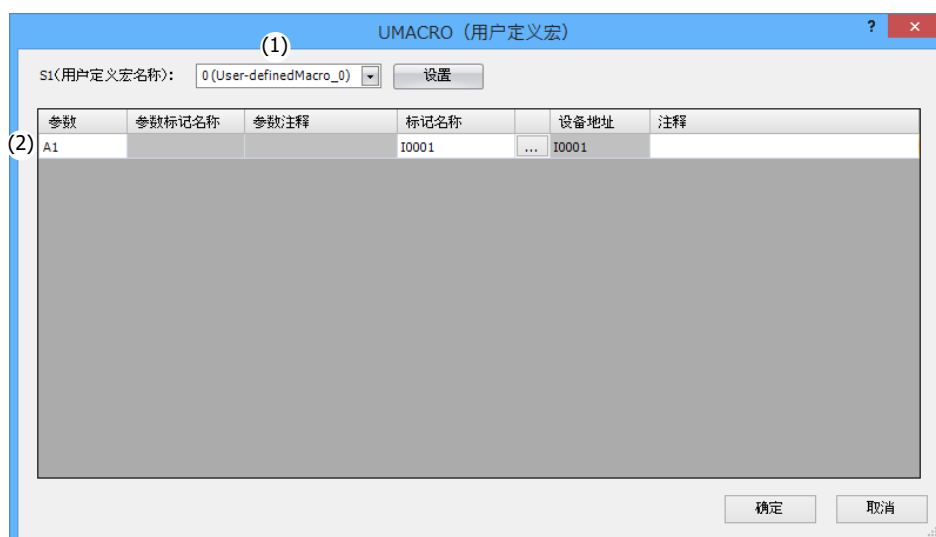
### 主程序



### 用户定义宏（用户定义宏名称 S1: 0）



### 设置项目



#### (1) S1（源 1）：用户定义宏名称

选择要运行的用户定义宏。

点击“设置”按钮后，显示“参数设置”对话框。

在“参数设置”对话框中，可针对各用户定义宏编号设置参数。

有关详情，请参见「“参数设置”对话框」（28-4 页）。

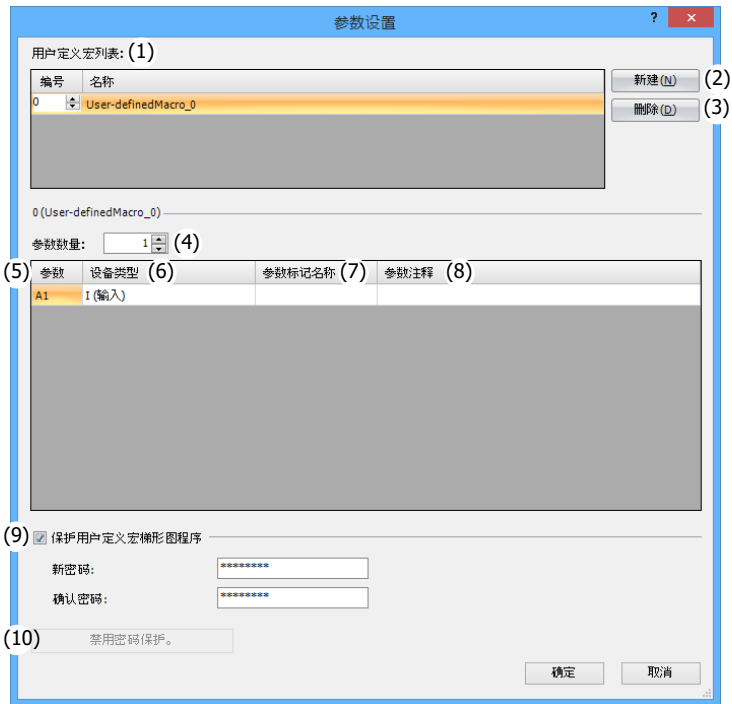
#### (2) 参数设置

设置用于提交至用户定义宏名称（1）所设置的用户定义宏的设备。根据“标记名称”列中显示的各参数的设备类型，设置设备地址或标记名称。参数数量及设备类型可在“参数设置”对话框中更改。

## 28: 用户定义宏指令

### “参数设置”对话框

此处可选择用户定义宏，或者编辑已选编号的用户定义宏。



#### (1) 用户定义宏列表

设置已列出。

编号：显示标识用户定义宏的数字（1-100）。

名称：显示用户定义宏的名称。

#### (2) 新建

添加新的用户定义宏编号。

#### (3) 删除

删除已注册的用户定义宏。

#### (4) 参数数量

选择要使用的参数数量。可在 0 ~ 100 的范围内进行指定。

#### (5) 参数

参数设备已列出。

**(6) 设备类型**

选择各参数的设备类型。在用户定义宏内使用参数设备时，以此处指定的设备类型进行动作。

设备类型	单位
I (输入)	位
Q (输出)	
M (内部继电器)	
R (移位寄存器)	
T (定时器位)	
TC (定时器当前值)	字
TP (定时器预置值)	
C (计数器位)	位
CC (计数器位)	字
CP (计数器预设值)	
D (数据寄存器)	
D (数据寄存器位)	位
常量 (W)	字
常量 (I)	
常量 (D)	
常量 (L)	2 个字
常量 (F)	

**注释：**如果参数的设备类型是常量，也要设置常量的数据类型。在用户定义宏中使用该参数的指令的数据类型应设置为与参数相同的数据类型。

**(7) 参数标记名称**

设置参数的标记名称。

**(8) 参数注释**

设置参数的注释。

**(9) 保护用户定义宏梯形图程序**

选中“保护用户定义宏梯形图程序”复选框后，可对用户定义宏设置密码。启用保护后，显示用户定义宏的梯形图程序时，将要求输入密码。

例如，允许使用已创建用户定义宏，但希望隐藏用户定义宏的梯形图程序时，需启用保护。

**(10) 禁用密码保护。**

启用保护时，无法更改参数设置。如需更改参数设置，请单击该按钮输入密码，以解除保护。

## 28: 用户定义宏指令

### 用户定义宏的注册步骤

本节将对用户定义宏的创建、注册步骤进行介绍。  
已注册的用户定义宏可用 UMACRO 指令进行设置和运行。

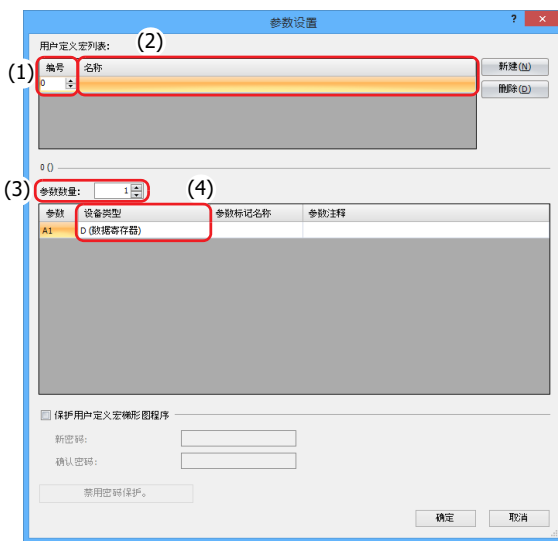
1. 右击项目窗口的“用户定义宏”，然后单击“新建”。

将显示“参数设置”对话框。



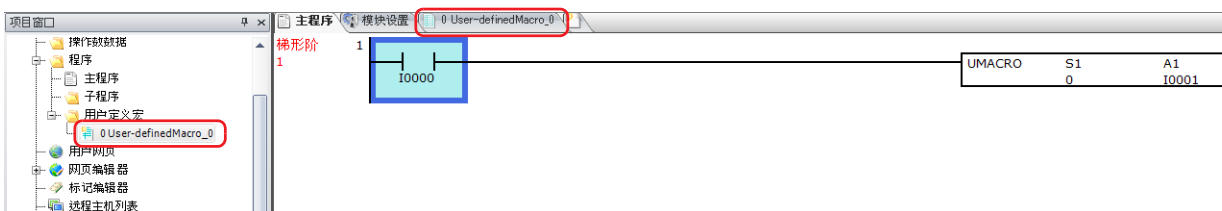
2. 设置用户定义宏。

- 在“编号”中设置用户定义宏的编号（1）。
- 在“书签名称”中输入用户定义宏编号的名称（2）。
- 在“参数数量”中选择用户定义宏所使用的参数数量（3）。
- 在“设备类型”中选择各参数的设备类型（4）。



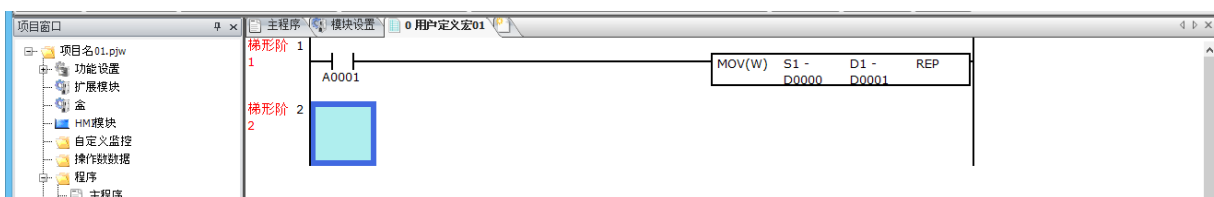
**注释：**选中“保护用户定义宏梯形图程序”复选框并设置密码，可防止无意中更改“参数设置”对话框的设置。

3. 单击“确定”按钮，关闭“参数设置”对话框。  
将显示创建用户定义宏的梯形图程序的编辑器。





4. 与主程序和子程序相同，通过用户定义宏编辑器创建梯形图程序。在“参数设置”对话框中设置的参数设备（A1 ~ A100）的内容可在“参数列表的用户定义宏分配”窗口中进行确认，并根据需要用于梯形图程序。已创建的用户定义宏可通过 UMACRO 指令执行。



**注释：**用户定义宏无法使用 UMACRO/JMP/JEND/MCR/MCS/LCAL/LJMP/LRET/LABEL/DJNZ/END 指令。

**注释：**用户定义宏可将已于“参数设置”对话框中设置的参数用作设备。例如，在“参数设置”对话框中将 A1 的设备类型设置为数据寄存器时，同样可将 A1 用作数据寄存器。

至此，完成用户定义宏的注册。

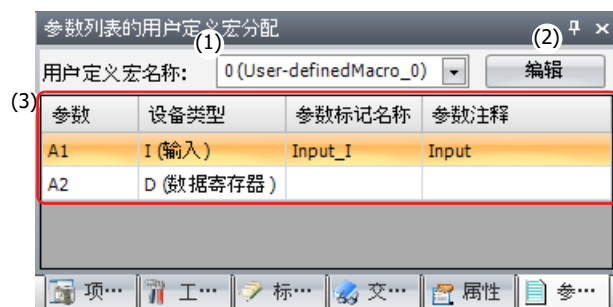
### “参数列表的用户定义宏分配”窗口

显示在用户定义宏中设置的参数内容。

1. 在“视图”选项卡的“工作区”中单击“参数列表的用户定义宏分配”。将显示“参数列表的用户定义宏分配”窗口。



2. 可在“参数设置”对话框中确认设置的内容。



**(1) 用户定义宏名称**

显示在“参数设置”对话框中注册的用户定义宏名称。

**(2) 编辑**

打开“参数设置”对话框。可编辑用户定义宏的设置。

**(3) 参数信息**

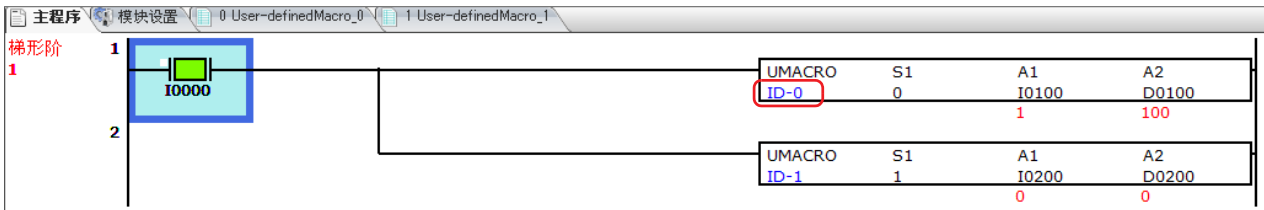
显示在已选择的用户定义宏中设置的参数设置内容。

## 28: 用户定义宏指令

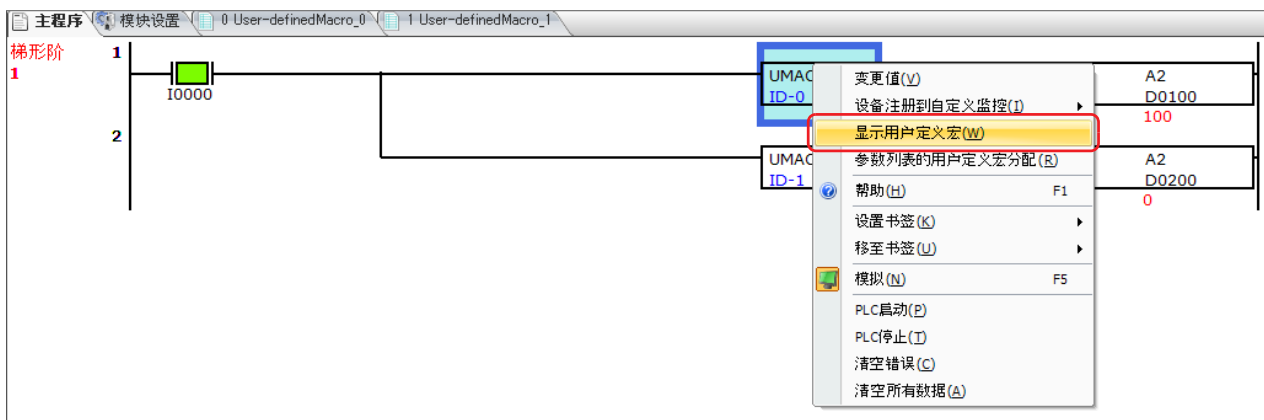
### 用户定义宏的模拟与监控

将对在 UMACRO 指令中设置的用户定义宏的模拟与监控进行说明。有关各操作方法，请参见第 1-9 页上的“模拟操作”、第 1-12 页上的“监控操作”。

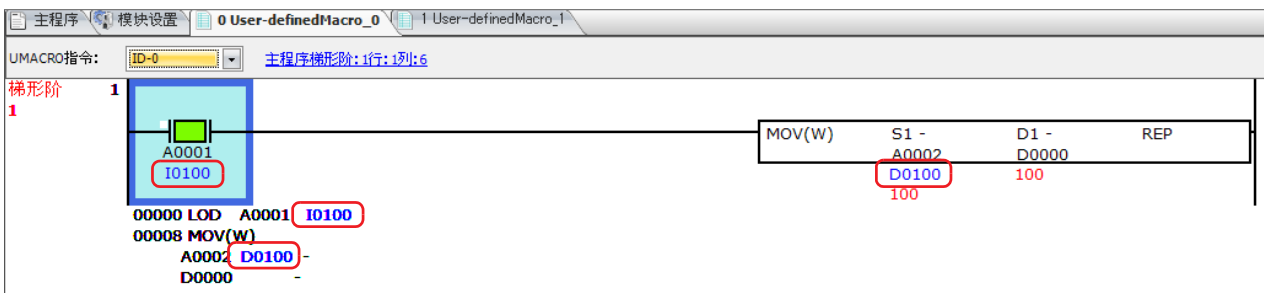
1. 运行模拟与监控后，在 UMACRO 指令上显示 UMACRO 指令的 ID。为了识别已设置的 UMACRO 指令，会被分配 UMACRO 指令的 ID。



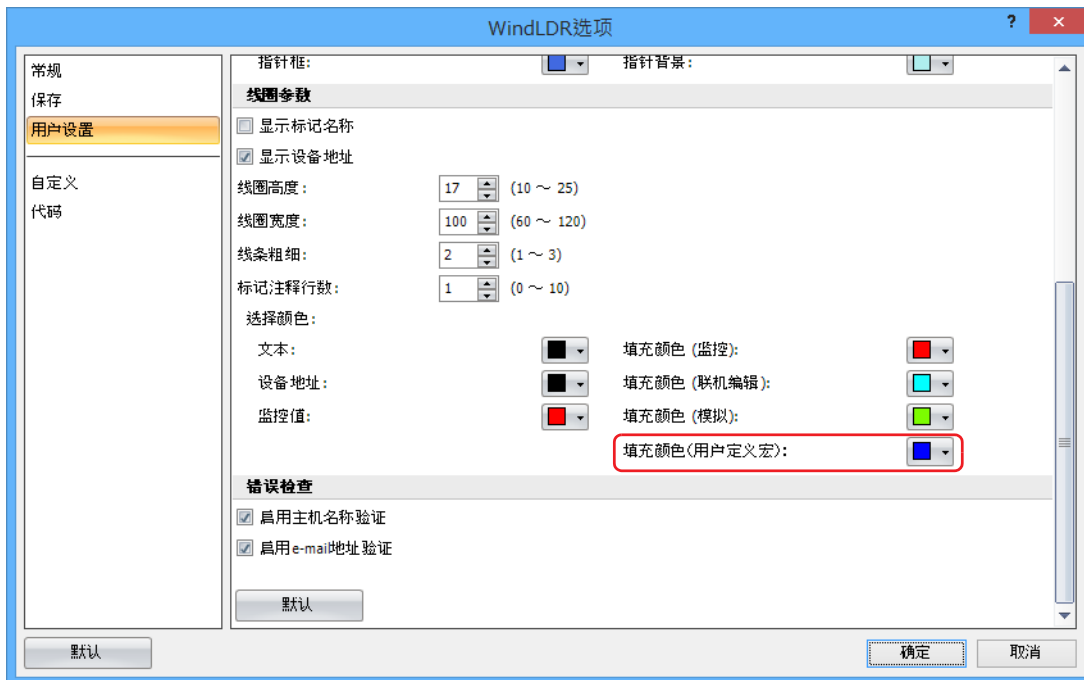
2. 选择 UMACRO 指令后点击右键，并在右键菜单中点击“显示用户定义宏”。



3. 打开在 UMACRO 指令中设置的编号所对应的用户定义宏。根据调用源的 UMACRO 指令中设置的设备地址或标记名称的情况，对已设置的参数设备 (A1 ~ A100) 进行模拟与监视。



在 UMACRO 指令上显示的 ID 以及在用户定义宏的梯形图程序上设置的参数的调用源设备地址、标记名称、注释等的颜色，可在“WindLDR 选项”对话框中更改。





# 附录

---

---

## 在一次扫描中处理

当控制器运行时，控制器重复执行输入刷新、梯形图处理和错误检查运算。

扫描是执行从地址零到 END 指令的所有指令的过程。此执行过程所需的时间称为一个扫描周期。扫描周期取决于程序长度而变化。

扫描时间的当前值将存储到特殊数据寄存器 D8023 (扫描时间当前值)，扫描时间的最大值将存储到特殊数据寄存器 D8024 (扫描时间最大值)。在 PC 上监控时，这些值可以通过 WindLDR “PLC 状态屏幕”对话框查看。

## 执行程序指令

在扫描周期期间，将从第一个梯形图行按顺序处理程序指令，中断程序的执行除外。梯形图一次扫描时间约等于各个指令执行时间总和。有关各指令的执行时间，请参见第附录 -3 页上的“指令的执行时间”。

## 结束处理时间细分

结束处理时间取决于控制器设置和系统设置。结束处理时间的最大值如下。

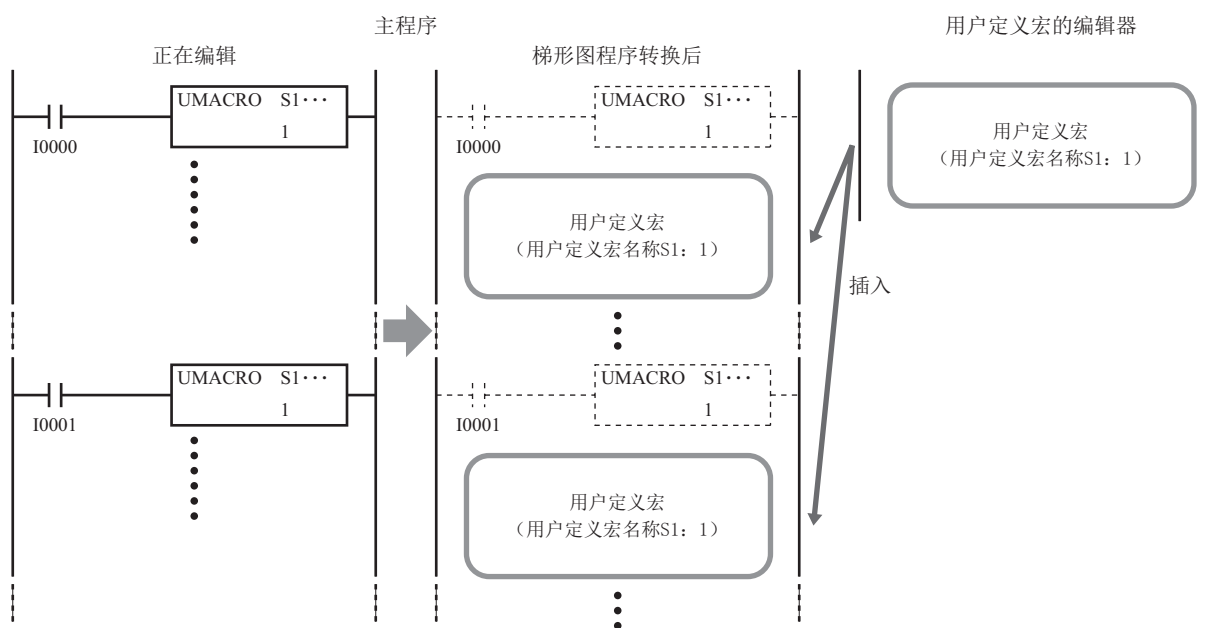
控制器	处理时间
FT2J 型	900 $\mu$ s
FC6A 型	640 $\mu$ s

## 用户定义宏与子程序的差异

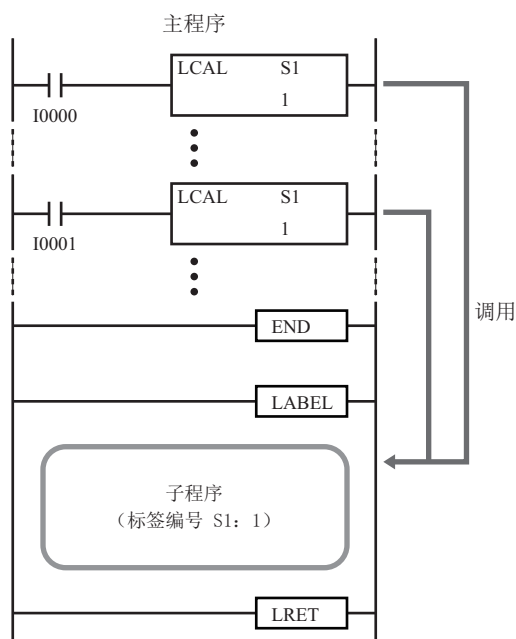
用户定义宏和子程序具有将多次使用的处理汇总成 1 个并通过调出指令执行的功能，存在以下差异。

项目	用户定义宏	子程序
创建位置和调用方法	用户定义宏在用户定义宏的编辑器中进行创建。可在 UMACRO 指令的位置插入相应用户定义宏名称的用户定义宏的内容。	子程序应创建在梯形图程序 END 指令的后面。每次执行 LCAL 指令时，均可调用相应标签编号的子程序。
用户程序大小	如果多次使用相同 UMACRO 指令的用户定义宏名称，则会在每次使用时增加用户程序大小。	即使多次使用相同 LCAL 指令的标签编号，用户程序大小仍不会增加。
参数设备	可使用。	无法使用。
多级调用	无法从用户定义宏调用其他用户定义宏。	可从子程序调用其他子程序。

### 用户定义宏



### 子程序



## 指令的执行时间

下面列出了基本指令和高级指令的执行时间。对所有设备均不指定重复。

### FT2J 型

指令	设备和条件	执行时间 (μs)
LOD	—	0.13
	使用数据寄存器	0.31
LODN	—	0.13
	使用数据寄存器	0.53
OUT	—	0.13
	使用数据寄存器	0.4
OUTN	—	0.13
	使用数据寄存器	0.4
SET、RST	—	0.13
	使用数据寄存器	0.4
AND	—	0.09
	使用数据寄存器	0.27
ANDN	—	0.13
	使用数据寄存器	0.53
OR	—	0.09
	使用数据寄存器	0.49
ORN	—	0.13
	使用数据寄存器	0.53
AND LOD、OR LOD	—	0.09
BPS	—	0.04
BRD	—	0.04
BPP	—	0.04
TML、TIM、TMH、TMS	—	7.69
TMLO、TIMO、TMHO、TMSO	—	7.64
CNT	—	7.24
CDP	—	7.73
CUD	—	7.51
CNTD、CDPD、CUDD	—	42.22
CC=、CC>=	—	4.13
DC=、DC>=	—	4.98
SFR、SFRN	N 位	$15.64 + 0.09n$
SOTU、SOTD	—	4.53
MCS、MCR、JMP、JEND	—	—
END	—	—
NOP	—	—
MOV (W、I)	M→M	9.73
	D→D	0.4
MOVN (W、I)	M→M	9.91
	D→D	0.44
MOV (D、L)	M→M	32.44
	D→D	0.8
MOVN (D、L)	M→M	32.89
	D→D	0.89
MOV (F)	—	0.84
IMOV、IMOVN (W)	M+D→M+D	18.67
	D+D→D+D	19.02
IMOV、IMOVN (D)	D+D→D+D	41.78
IMOV (F)	—	31.11
MOVC	—	—
BMOV	D→D	$36.0 + 2.67n$
IBMV、IBMVN	M+D→M+D	29.78
	D+D→D+D	29.78

指令	设备和条件	执行时间 (μs)
NSET (W、I)	D→D	5.07+4.89n
NSET (D、L)	D→D	4.89 + 28.0n
NSET (F)	D→D	4.89 + 28.0n
NRS (W、I)	D、D→D	12.89 + 1.42n
NRS (D、L)	D、D→D	24.89 + 12.89n
NRS (F)	D、D→D	24.44 + 12.89n
XCHG (W)	D↔D	11.82
XCHG (D)	D↔D	57.78
TCCST (W)	D→T	10.4
TCCST (D)	D→T	33.33
CMP (=、<>、<、>、<=、>=) (W、I)	D↔D→M	61.33
CMP (=、<>、<、>、<=、>=) (D、L)	D↔D→M	84.44
CMP (=、<>、<、>、<=、>=) (F)	D↔D→M	84.89
ICMP (>=)	D↔D↔D→M	64.44
ICMP (D、L、F)	D↔D↔D→M	99.11
LC (=、<>、<、>、<=、>=) (W、I)	D↔D	10.76
LC (=、<>、<、>、<=、>=) (D、L)	D↔D	33.78
LC (=、<>、<、>、<=、>=) (F)	D↔D	34.22
ADD (W、I)	M+M→D	25.78
	D+D→D	26.22
ADD (D、L)	M+M→D	60.44
	D+D→D	60.44
ADD (F)	D+D→D	61.78
SUB (W、I)	M-M→D	25.78
	D-D→D	26.22
SUB (D、L)	M-M→D	60.44
	D-D→D	61.78
SUB (F)	D-D→D	60.89
MUL (W、I)	M×M→D	24.89
	D×D→D	25.33
MUL (D、L)	M×M→D	60.44
	D×D→D	60.44
MUL (F)	D×D→D	60.89
DIV (W、I)	M÷M→D	15.56
	D÷D→D	15.47
DIV (D、L)	M÷M→D	62.22
	D÷D→D	62.22
DIV (F)	D÷D→D	62.22
INC (W、I)	—	19.82
INC (D、L)	—	42.89
DEC (W、I)	—	19.82
DEC (D、L)	—	42.67
ROOT (W)	$\sqrt{D}$ →D	13.87
ROOT (D)	$\sqrt{D}$ →D	40.0
ROOT (F)	$\sqrt{D}$ →D	43.11
SUM (W、I)	D、D→D	25.33 + 1.78n
SUM (D、L)	D、D→D	36.89 + 13.33n
SUM (F)	D、D→D	37.33 + 15.56n
RNDM	D、D→D	15.20
ANDW、ORW、XORW (W)	M·M→D	13.51
	D·D→D	13.78
ANDW、ORW、XORW (D)	D·D→D	48.0
SFTL、SFTR	N_B=100	31.33
BCDLS	D→D、S1=1	23.02
WSFT	D→D	34.67 + 3.11n
ROTL、ROTR (W)	D、bits=1	20.44
HTOB (W)	D→D	10.98



指令	设备和条件	执行时间 (μs)
BTOH (W)	D→D	10.98
HTOA (W)	D→D	16.00
ATOH (W)	D→D	12.18 + 1.96n
BTOA (W)	D→D	13.07 + 1.6n
ATOB (W)	D→D	12.98 + 1.69n
ENCO (W)	M→D、 bits=16	8.67 + 1.78n
DECO	D→M	21.33
BCNT	M→D、 bits=16	—
ALT	—	17.78
CVDT	W、 I→F	22.67
	D、 L→F	34.22
	F→W、 I	23.11
	F→D、 L	32.44
DTDV	D→D	10.62
DTCB	D→D	11.02
SWAP (W)	—	9.91
SWAP (D)	—	33.33
WEEK	—	—
YEAR	—	—
LABEL	—	—
LJMP	—	—
LCAL	—	—
LRET	—	—
DJNZ	—	—
IOREF	—	21.07
HSCRF	—	18.49
FRQRF	—	—
DI	—	16.53
EI	—	4.62
XYFS	—	—
CVXTY	—	—
CVYTX	—	—
AVRG (W、 I)	S3=10	17.24
AVRG (D、 L)	S3=10	30.22
AVRG (F)	S3=10	32.0
PULS	—	—
PWM	—	—
PIDA	—	—
PIDD	—	—
DTML、 DTIM、 DTMH	—	32.44
DTMS	—	30.67
TTIM	—	10.13
RAD	F→F	63.11
DEG	F→F	63.11
SIN	F→F	40.44
COS	F→F	40.89
TAN	F→F	39.11
ASIN	F→F	54.22
ACOS	F→F	54.67
ATAN	F→F	50.67
LOGE	F→F	42.22
LOG10	F→F	43.11
EXP	F→F	49.33
POW	F→F	75.56
FIFOF	—	51.56
FIEX	—	48.0
FOEX	—	48.0

指令	设备和条件	执行时间 (μs)
NDSRC (W、I)	D、D、D→D	26.89
NDSRC (D、L)	D、D、D→D	93.33
NDSRC (F)	D、D、D→D	96.89
TADD	—	28.44
TSUB	—	28.44
HOUR	D→D、Y、D	17.78
HTOS	D→D	22.22
STOH	D→D	34.22
SCRPT	—	—
SCALE	—	—
FLWA	—	—
FLWP	—	—
UMACRO	—	—

## FC6A 型

指令	设备和条件	执行时间 (μs)	
		All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
LOD	—	0.05	0.03
	使用数据寄存器	0.13	0.07
LODN	—	0.06	0.03
	使用数据寄存器	0.23	0.12
OUT	—	0.05	0.03
	使用数据寄存器	0.17	0.09
OUTN	—	0.05	0.03
	使用数据寄存器	0.17	0.09
SET、RST	—	0.05	0.03
	使用数据寄存器	0.17	0.09
AND	—	0.04	0.02
	使用数据寄存器	0.12	0.06
ANDN	—	0.05	0.03
	使用数据寄存器	0.23	0.12
OR	—	0.04	0.02
	使用数据寄存器	0.21	0.11
ORN	—	0.05	0.03
	使用数据寄存器	0.23	0.12
AND LOD、OR LOD	—	0.03	0.02
BPS	—	0.02	0.01
BRD	—	0.02	0.01
BPP	—	0.02	0.01
TML、TIM、TMH、TMS	—	3.46	1.73
TMLO、TIMO、TMHO、TMSO	—	3.44	1.72
CNT	—	3.26	1.63
CDP	—	3.48	1.74
CUD	—	3.38	1.69
CNTD、CDPD、CUDD	—	18.9	9.5
CC=、CC>=	—	1.85	0.93
DC=、DC>=	—	2.24	1.12
SFR、SFRN	N 位	$7.04 + 0.04n$	$3.52 + 0.02n$
SOTU、SOTD	—	2.04	1.02
MCS、MCR、JMP、JEND	—	—	—
END	—	—	—
NOP	—	—	—
MOV (W、I)	M→M	4.38	2.19
	D→D	0.18	0.09
MOVN (W、I)	M→M	4.46	2.23
	D→D	0.19	0.10
MOV (D、L)	M→M	14.6	7.3
	D→D	0.36	0.18
MOVN (D、L)	M→M	14.7	7.4
	D→D	0.39	0.20
MOV (F)	—	0.37	0.19
IMOV、IMOVN (W)	M+D→M+D	8.4	4.2
	D+D→D+D	8.56	4.28
IMOV、IMOVN (D)	D+D→D+D	18.8	9.4
IMOV (F)	—	13.9	7.00
MOVC	—	—	—
BMOV	D→D	$16.1 + 1.2n$	$8.1 + 0.6n$
IBMV、IBMVN	M+D→M+D	13.4	6.7
	D+D→D+D	13.4	6.7

指令	设备和条件	执行时间 (μs)	
		All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
NSET (W、 I)	D→D	2.28 + 2.2n	1.14 + 1.1n
NSET (D、 L)	D→D	2.2 + 12.6n	1.1 + 6.3n
NSET (F)	D→D	2.2 + 12.6n	1.1 + 6.3n
NRS (W、 I)	D、 D→D	5.84 + 0.64n	2.92 + 0.32n
NRS (D、 L)	D、 D→D	11.1 + 5.7n	5.6 + 2.9n
NRS (F)	D、 D→D	11.0 + 5.8n	5.5 + 2.9n
XCHG (W)	D⇔D	5.32	2.66
XCHG (D)	D⇔D	26.0	13.0
TCCST (W)	D→T	4.68	2.34
TCCST (D)	D→T	15.0	7.5
CMP (=、 <>、 <、 >、 <=、 >=) (W、 I)	D⇔D→M	27.6	13.8
CMP (=、 <>、 <、 >、 <=、 >=) (D、 L)	D⇔D→M	38.0	19.0
CMP (=、 <>、 <、 >、 <=、 >=) (F)	D⇔D→M	38.2	19.1
ICMP (>=)	D⇔D⇔D→M	29.0	14.5
ICMP (D、 L、 F)	D⇔D⇔D→M	44.6	22.3
LC (=、 <>、 <、 >、 <=、 >=) (W、 I)	D⇔D	4.84	2.42
LC (=、 <>、 <、 >、 <=、 >=) (D、 L)	D⇔D	15.2	7.6
LC (=、 <>、 <、 >、 <=、 >=) (F)	D⇔D	15.3	7.7
ADD (W、 I)	M+M→D	11.6	5.8
	D+D→D	11.7	5.9
ADD (D、 L)	M+M→D	27.2	13.6
	D+D→D	27.2	13.6
ADD (F)	D+D→D	27.8	13.9
SUB (W、 I)	M-M→D	11.6	5.8
	D-D→D	11.7	5.9
SUB (D、 L)	M-M→D	27.2	13.6
	D-D→D	27.2	13.6
SUB (F)	D-D→D	27.4	13.7
MUL (W、 I)	M×M→D	11.2	5.6
	D×D→D	11.3	5.7
MUL (D、 L)	M×M→D	27.2	13.6
	D×D→D	27.2	13.6
MUL (F)	D×D→D	27.4	13.7
DIV (W、 I)	M÷M→D	7.0	3.5
	D÷D→D	6.96	3.48
DIV (D、 L)	M÷M→D	28.0	14.0
	D÷D→D	28.0	14.0
DIV (F)	D÷D→D	28.0	14.0
INC (W、 I)	—	8.92	4.46
INC (D、 L)	—	19.3	9.65
DEC (W、 I)	—	8.92	4.46
DEC (D、 L)	—	19.2	9.6
ROOT (W)	$\sqrt{D}$ →D	6.24	3.12
ROOT (D)	$\sqrt{D}$ →D	18.0	9.0
ROOT (F)	$\sqrt{D}$ →D	19.4	9.7
SUM (W、 I)	D、 D→D	11.3 + 0.8n	5.7 + 0.4n
SUM (D、 L)	D、 D→D	16.6 + 6.0n	8.3 + 3.0n
SUM (F)	D、 D→D	16.8 + 7.0n	8.4 + 3.5n
RNDM	D、 D→D	6.84	3.42
ANDW、 ORW、 XORW (W)	M·M→D	6.08	3.04
	D·D→D	6.2	3.1
ANDW、 ORW、 XORW (D)	D·D→D	21.6	10.8
SFTL、 SFTR	N_B=100	14.10	7.05
BCDLS	D→D、 S1=1	10.36	5.18

指令	设备和条件	执行时间 (μs)	
		All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
WSFT	D→D	15.6 + 1.4n	7.8 + 0.7n
ROTL、ROTR (W)	D、bits=1	9.1	4.6
HTOB (W)	D→D	4.94	2.47
BTOH (W)	D→D	4.94	2.47
HTOA (W)	D→D	7.20	3.6
ATOH (W)	D→D	5.48 + 0.88n	2.74 + 0.44n
BTOA (W)	D→D	5.88 + 0.72n	2.94 + 0.36n
ATOB (W)	D→D	5.84 + 0.76n	2.92 + 0.38n
ENCO (W)	M→D、bits=16	3.9 + 0.8n	1.95 + 0.4n
DECO	D→M	9.6	4.8
BCNT	M→D、bits=16	—	—
ALT	—	8.00	4.00
CVDT	W、I→F	10.2	5.1
	D、L→F	15.3	7.7
	F→W、I	10.3	5.2
	F→D、L	15.5	7.3
DTDV	D→D	4.78	2.39
DTCB	D→D	4.96	2.48
SWAP (W)	—	4.46	2.23
SWAP (D)	—	14.9	7.5
WKTIM	—	—	—
WKTBL	—	—	—
WEEK	—	—	—
YEAR	—	—	—
MSG	—	—	—
DISP	—	—	—
DGRD	—	—	—
LABEL	—	—	—
LJMP	—	—	—
LCAL	—	—	—
LRET	—	—	—
DJNZ	—	—	—
IREF	—	9.48	4.74
HSCRF	—	8.32	4.16
FRQRF	—	—	—
COMRF	—	—	—
DI	—	7.44	3.72
EI	—	2.07	1.04
XYFS	—	—	—
CVXTY	—	—	—
CVYTX	—	—	—
AVRG (W、I)	S3=10	7.76	3.88
AVRG (D、L)	S3=10	13.5	6.8
AVRG (F)	S3=10	14.4	7.2
PULS	—	—	—
PWM	—	—	—
RAMP	—	—	—
RAMPL	—	—	—
ZRN	—	—	—
ARAMP	—	—	—
ABS	—	—	—
JOG	—	—	—
PID	—	—	—
PIDA	—	—	—

指令	设备和条件	执行时间 (μs)	
		All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
PIDD	—	—	—
DTML、DTIM、DTMH	—	14.6	7.3
DTMS	—	13.7	6.9
TTIM	—	4.56	2.28
RAD	F→F	28.4	14.2
DEG	F→F	28.4	14.2
SIN	F→F	18.2	9.1
COS	F→F	18.4	9.2
TAN	F→F	17.6	8.8
ASIN	F→F	24.4	12.2
ACOS	F→F	24.6	12.3
ATAN	F→F	22.8	11.4
LOGE	F→F	19.00	9.5
LOG10	F→F	19.40	9.7
EXP	F→F	22.1	11.1
POW	F→F	34.0	17.0
FIFO	—	23.2	11.6
FIEX	—	21.6	10.8
FOEX	—	21.6	10.8
NDSRC (W、I)	D、D、D→D	12.10	6.05
NDSRC (D、L)	D、D、D→D	43.0	21.0
NDSRC (F)	D、D、D→D	43.6	21.8
TADD	—	12.8	6.4
TSUB	—	12.8	6.4
HOUR	D→D、Y、D	8.0	4.0
HTOS	D→D	10.0	5.0
STOH	D→D	15.4	7.7
DLOG	—	—	—
TRACE	—	—	—
SCRPT	—	—	—
SCALE	—	—	—
FLWA	—	—	—
FLWP	—	—	—
UMACRO	—	—	—
TXD	—	—	—
RXD	—	—	—
ETXD	—	—	—
ERXD	—	—	—
PING	—	—	—
EMAIL	—	—	—

## 指令数据长度（字节）

下面列出了基本指令和高级指令的字节数。

## FT2J 型

基本指令	占用数据长度（字节）	
	使用数据寄存器位指定时	使用位设备时
LOD	12	8
LODN	12	12
OUT	8	8
OUTN	8	8
SET, RST	8	8
AND	12	8
ANDN	12	12
OR	12	8
ORN	12	12
AND LOD	—	8
OR LOD	—	8
BPS	—	4
BRD	—	4
BPP	—	4
TML	—	12
TIM	—	12
TMH	—	12
TMS	—	12
TMLO	—	12
TIMO	—	12
TMHO	—	12
TMSO	—	12
CNT	—	12
CDP	—	12
CUD	—	12
CNTD	—	12
CDPD	—	12
CUDD	—	12
CC=	—	12
CC>=	—	12
DC=	—	12-16
DC>=	—	12-16
SFR	—	12
SFRN	—	12
SOTU	—	8
SOTD	—	8
MCS	—	4
MCR	—	4
JMP	—	12
JEND	—	4
END	—	4

高级指令	占用数据长度（字节）
NOP	8

高级指令	占用数据长度 (字节)
MOV (W, I)	16-20
MOVN (W, I)	16-20
MOV (D, L)	16-20
MOVN (D, L)	16-20
MOV (F)	16-20
IMOV, IMOVN (W)	20-28
IMOV, IMOVN (D)	20-28
IMOV (F)	20-28
MOVC	16-1040
BMOV	16-20
IBMV, IBMVN	20-28
NSET (W, I)	16-1544
NSET (D, L)	16-1544
NSET (F)	16-1544
NRS (W, I)	16-24
NRS (D, L)	16-24
NRS (F)	16-24
XCHG (W)	12-16
XCHG (D)	12-16
TCCST (W)	16-20
TCCST (D)	16-20
CMP (=, <, <=, >, >=) (W, I)	20-28
CMP (=, <, <=, >, >=) (D, L)	20-28
CMP (=, <, <=, >, >=) (F)	20-28
ICMP (>=)	16-32
ICMP (D, L, F)	16-32
LC (=, <, <=, >, >=) (W, I)	12-20
LC (=, <, <=, >, >=) (D, L)	12-20
LC (=, <, <=, >, >=) (F)	12-20
ADD (W, I)	16-28
ADD (D, L)	16-28
ADD (F)	16-28
SUB (W, I)	16-28
SUB (D, L)	16-28
SUB (F)	16-28
MUL (W, I)	16-28
MUL (D, L)	16-28
MUL (F)	16-28
DIV (W, I)	16-28
DIV (D, L)	16-28
DIV (F)	16-28
INC (W, I)	12
INC (D, L)	12
DEC (W, I)	12
DEC (D, L)	12
ROOT (W)	12-20
ROOT (D)	12-20
ROOT (F)	12-20
SUM (W, I)	16-24
SUM (D, L)	16-24



高级指令	占用数据长度 (字节)
SUM (F)	16-24
RNDM	16-20
ANDW, ORW, XORW (W)	16-28
ANDW, ORW, XORW (D)	16-28
SFTL, SFTR	16-24
BCDLS	12-16
WSFT	16-20
ROTL, ROTR (W)	12-16
HTOB (W)	12-20
BTOH (W)	12-20
HTOA (W)	16-20
ATOH (W)	16-20
BTOA (W)	16-20
ATOB (W)	16-20
ENCO (W)	16-20
DECO	16-20
BCNT	16-20
ALT	12
CVDT	16-20
DTDV	12-16
DTCB	12-16
SWAP (W)	16-20
SWAP (D)	16-20
WEEK	20-140
YEAR	28-220
LABEL	12
LJMP	12
LCAL	12
LRET	8
DJNZ	12-16
IOREF	12
HSCRF	8
FRQRF	8
DI	12
EI	12
XYFS	24-272
CVXTY	16-20
CVYTX	16-20
AVRG (W, I)	20-24
AVRG (D, L)	20-24
AVRG (F)	20-24
PULS	12-28
PWM	12-28
PIDA	112-116
PIDD	118
DTML, DTIM, DTMH	16-24
DTMS	16-24
TTIM	12
RAD	12-20
DEG	12-20

高级指令	占用数据长度 (字节)
SIN	12-20
COS	12-20
TAN	12-20
ASIN	12-20
ACOS	12-20
ATAN	12-20
LOGE	20
LOG10	20
EXP	20
POW	16-24
FIFO	24
FIEX	12-16
FOEX	12-16
NDSRC (W, I)	16-28
NDSRC (D, L)	16-28
NDSRC (F)	16-28
TADD	16-24
TSUB	16-24
HOUR	20-24
HTOS	12-16
STOH	12-20
SCRPT	16-20
SCALE	36-44
FLWA	24-32
FLWP	28-36
UMACRO	24-624

## FC6A 型

基本指令	占用数据长度 (字节)			
	All-in-One CPU 模块 /CAN J1939 All-in-One CPU 模块		Plus CPU 模块	
	使用数据寄存器位指定时	使用位设备时	使用数据寄存器位指定时	使用位设备时
LOD	12	8	12	8
LODN	12	12	12	12
OUT	8	8	8	8
OUTN	8	8	8	8
SET、RST	8	8	8	8
AND	12	8	12	8
ANDN	12	12	12	12
OR	12	8	12	8
ORN	12	12	12	12
AND LOD	—	8	—	8
OR LOD	—	8	—	8
BPS	—	4	—	4
BRD	—	4	—	4
BPP	—	4	—	4
TML	—	12	—	12-16
TIM	—	12	—	12-16
TMH	—	12	—	12-16
TMS	—	12	—	12-16
TMLO	—	12	—	12-16
TIMO	—	12	—	12-16
TMHO	—	12	—	12-16
TMSO	—	12	—	12-16
CNT	—	12	—	12-16
CDP	—	12	—	12-16
CUD	—	12	—	12-16
CNTD	—	12	—	12-16
CDPD	—	12	—	12-16
CUDD	—	12	—	12-16
CC=	—	12	—	12-16
CC>=	—	12	—	12-16
DC=	—	12-16	—	12-24
DC>=	—	12-16	—	12-24
SFR	—	12	—	12
SFRN	—	12	—	12
SOTU	—	8	—	8
SOTD	—	8	—	8
MCS	—	4	—	4
MCR	—	4	—	4
JMP	—	12	—	12
JEND	—	4	—	4
END	—	4	—	4

高级指令	占用数据长度 (字节)	
	All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
NOP	8	8
MOV (W、I)	16-20	16-28
MOVN (W、I)	16-20	16-28
MOV (D、L)	16-20	16-28
MOVN (D、L)	16-20	16-28
MOV (F)	16-20	16-28
IMOV、IMOVN (W)	20-28	20-44
IMOV、IMOVN (D)	20-28	20-44
IMOV (F)	20-28	20-44
MOVC	16-1040	16-1044
BMOV	16-20	16-32
IBMV、IBMVN	20-28	20-44
NSET (W、I)	16-1544	16-2060
NSET (D、L)	16-1544	16-2060
NSET (F)	16-1544	16-2060
NRS (W、I)	16-24	16-32
NRS (D、L)	16-24	16-32
NRS (F)	16-24	16-32
XCHG (W)	12-16	12-24
XCHG (D)	12-16	12-24
TCCST (W)	16-20	12-24
TCCST (D)	16-20	12-24
CMP (=、<>、<、>、<=、>=) (W、I)	20-28	16-32
CMP (=、<>、<、>、<=、>=) (D、L)	20-28	16-32
CMP (=、<>、<、>、<=、>=) (F)	20-28	16-32
ICMP (>=)	16-32	16-36
ICMP (D、L、F)	16-32	16-36
LC (=、<>、<、>、<=、>=) (W、I)	12-20	12-24
LC (=、<>、<、>、<=、>=) (D、L)	12-20	12-24
LC (=、<>、<、>、<=、>=) (F)	12-20	12-24
ADD (W、I)	16-28	16-36
ADD (D、L)	16-28	16-36
ADD (F)	16-28	16-36
SUB (W、I)	16-28	16-36
SUB (D、L)	16-28	16-36
SUB (F)	16-28	16-36
MUL (W、I)	16-28	16-36
MUL (D、L)	16-28	16-36
MUL (F)	16-28	16-36
DIV (W、I)	16-28	16-36
DIV (D、L)	16-28	16-36
DIV (F)	16-28	16-36
INC (W、I)	12	12-16
INC (D、L)	12	12-16
DEC (W、I)	12	12-16
DEC (D、L)	12	12-16
ROOT (W)	12-20	12-24
ROOT (D)	12-20	12-24
ROOT (F)	12-20	12-24
SUM (W、I)	16-24	16-36
SUM (D、L)	16-24	16-36
SUM (F)	16-24	16-36
RNDM	16-20	16-32
ANDW、ORW、XORW (W)	16-28	16-28

高级指令	占用数据长度 (字节)	
	All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
ANDW、ORW、XORW (D)	16-28	16-28
SFTL、SFTR	16-24	16-40
BCDLS	12-16	12-24
WSFT	16-20	16-32
ROTL、ROTR (W)	12-16	12-20
HTOB (W)	12-20	12-24
BTOH (W)	12-20	12-24
HTOA (W)	16-20	16-32
ATOH (W)	16-20	16-32
BTOA (W)	16-20	16-32
ATOB (W)	16-20	16-32
ENCO (W)	16-20	16-28
DECO	16-20	16-28
BCNT	16-20	16-32
ALT	12	12
CVDT	16-20	16-28
DTDV	12-16	12-24
DTCB	12-16	12-24
SWAP (W)	16-20	16-28
SWAP (D)	16-20	16-28
WKTIM	20-24	20-40
WKTBL	12-92	12-172
WEEK	20-140	24-148
YEAR	28-220	28-228
MSG	20-	20-
DISP	16	16-24
DGRD	16-20	16-28
LABEL	12	12
LJMP	12	12-16
LCAL	12	12-16
LRET	8	8
DJNZ	12-16	12-24
IOREF	12	12
HSCRF	8	8
FRQRF	8	8
COMRF	8	8
DI	12	12
EI	12	12
XYFS	24-272	24 - 528
CVXTY	16-20	16 - 28
CVYTX	16-20	16 - 28
AVRG (W、 I)	20-24	20-40
AVRG (D、 L)	20-24	20-40
AVRG (F)	20-24	20-40
PULS	12-28	12-36
PWM	12-28	12-36
RAMP	12-36	12-44
RAMPL	36-48	36-48
ZRN	16, 52	16-68
ARAMP	40-316	40-332
ABS	12-16	12-20
JOG	36	36-48
PID	20-28	20-44
PIDA	112-116	112-124

高级指令	占用数据长度 (字节)	
	All-in-One CPU 模块 / CAN J1939 All-in-One CPU 模块	Plus CPU 模块
PIDD	272-276	272-288
DTML、DTIM、DTMH	16-24	16-36
DTMS	16-24	16-36
TTIM	12	12-16
RAD	12-20	12-24
DEG	12-20	12-24
SIN	12-20	12-24
COS	12-20	12-24
TAN	12-20	12-24
ASIN	12-20	12-24
ACOS	12-20	12-24
ATAN	12-20	12-24
LOGE	20	12-24
LOG10	20	12-24
EXP	20	12-24
POW	16-24	16-32
FIFO	24	24-32
FIEX	12-16	12-20
FOEX	12-16	12-20
NDSRC (W、I)	16-28	16-40
NDSRC (D、L)	16-28	16-40
NDSRC (F)	16-28	16-40
TADD	16-24	16-36
TSUB	16-24	16-36
HOURL	20-24	20-40
HTOS	12-16	12-24
STOH	12-20	12-24
DLOG	28-408	28-668
TRACE	28-408	28-668
SCRPT	16-20	16-28
SCALE	36-44	36-60
FLWA	24-32	24-48
FLWP	28-36	28-52
UMACRO	24-624	12-812
TXD	20-	20-
RXD	20-	20-
ETXD	20-	20-
ERXD	20-	20-
PING	16-20	16-28
EMAIL	16-20	16-28

## 用户程序执行错误和梯形图程序执行错误

如果出现用户程序执行错误或梯形图程序执行错误，错误的详细信息（错误代码）将存储在 D8006 中。

错误代码	错误详细信息
1	源 / 目标设备超出范围
2	MUL 结果超出数据类型范围。
3	DIV 结果超出数据类型范围，或被 0 除。
4	BCDLS 使 S1 或 S1+1 超过 9999。
5	S1 在 HTOB (W) 指令中为 10,000 或更大，或者 S1 在 HTOB (D) 指令中为 100,000,000 或更大
6	BTOH 使 S1 的任意位超过 9。
7	HTOA/ATOH/BTOA/ATOB 使要转换的数字的位数超出范围。
8	ATOH/ATOB 有 S1 ~ S1+4 非 ASCII 数据。
9	在 WEEK 指令中，没有指定星期，ON 时间数据大于 2359，OFF 时间数据大于 2400，ON 时间数据或 OFF 时间数据的最后两位数大于 59。
10	在 YEAR 指令中，月和日数据超出范围（年数据为 2000-2099 除外，月数据为 13 或更大，日数据为 32 或更大）。
11	在 DGRD 指令中，BCD 且 5 位指定，65,535 或更大
12	执行的 CVXTY/CVYTX 与 XYFS 不匹配。 XYFS 和 CVXTY/CVYTX 有相同的 S1，但是有不同的数据类型。
13	CVXTY/CVYTX 使 S2 超过在 XYFS 中指定的值。
14	LJMP、LCAL 或 DJNZ 指令未指定标签编号。 或者中断输入中的跳转目标标签编号没有指定标签编号。
16	PID/PIDA/PIDD 指令执行错误
18	已尝试执行不能用在中断程序中的指令（请参见第 3-1 页上的“指令参考”）。
19	已尝试执行 PLC 中不可用的指令。
20	脉冲输出指令中动作参数的设置错误
21	DECO 使 S1 超过 255。
22	BCNT 使 S2 超过 256。
23	ICMP>= 有 S1<S3。
25	BCDLS 使 S2 超过 7。
26	当功能设置中未编译中断输入或定时器中断时，将执行 DI 或 EI。
27	当使用 DTML、DTIM、DTMH、DTMS 或 TTIM 时，工作区被破坏。
28	指定浮点型的处理单位时，源设备数据超出所定义的范围
29	F(浮点型)数据类型指令的结果超出数据类型范围。
30	在 SFTL/SFTR 指令中，移位数据大小或移位位数超出所定义的范围
31	在 FIFO 指令中，FIEX 指令或 FOEX 指令在注册 FIFO 数据文件之前已执行。
32	在 TADD、TSUB、HOUR 或 HTOS 指令中，源设备 S1 指定为无效值。
33	在 RNDM 指令中，S1 的数据大于或等于 S2 的数据或者 S1、S2 的数据超过 32,767。
35	在 SUM 指令中，执行结果超出指定的数据类型的有效范围，或 S2 数据为 0。
36	在 DLOG/TRACE 指令中，执行指令时超过了 CSV 文件 1 天的容量限制或文件数限制
41	SD 记忆卡为写保护。
42	使用 SCRPT 指令执行的结果为表示除正常完成之外的任何值。
46	在 SCALE 指令中，输入最小值比输入最大值大，输出最小值比输出最大值大，或静带数据超出所定义的范围
48	已通过脉冲输出指令同时启动使用相同输出端口的指令
49	在 ABS 指令中，未初始化绝对位置计数器，即执行参照绝对位置计数器的脉冲输出指令。









# 索引

- #
  - 100 毫秒
    - 双定时器 20-1
  - 10 毫秒
    - 双定时器 20-1
  - 1 毫秒双定时器 20-1
  - 1 秒
    - 双定时器 20-1
  - 32 位数据存储 3-10
- A
  - ACOS 21-7
  - ADD 7-1
  - advanced instruction
    - DEC 7-14
    - RNDM 7-19
    - SUM 7-16
  - ALT 10-18
  - AND LOD 指令 4-6
  - AND 和 ANDN 指令 4-4
  - ANDW 8-1
  - ASCII
    - 到 BCD 10-12
    - 到 HEX 10-7
  - ASIN 21-6
  - ATAN 21-8
  - ATOB 10-12
  - ATOH 10-7
  - AVRG 17-1
- B
  - BCD
    - 到 ASCII 10-9
    - 到 HEX 10-3
    - 左移 9-5
  - BCDLS 9-5
  - BCNT 10-17
  - 比例转换指令 (SCALE) 27-1
  - BMOV 5-12
  - BPS、BRD 和 BPP 指令 4-7
  - BTOA 10-9
  - BTOH 10-3
  - 备注 26-18
  - 比较
    - 不等于 6-1
    - 大于 6-1
    - 大于或等于 6-2
    - 等于 6-1
    - 小于 6-1
    - 小于或等于 6-2
  - 比较结果
    - M8150、M8151、M8152 6-5, 6-6
  - 比较运算 26-24
  - 编辑用户程序 1-4
  - 编码 10-15
  - 表
    - 高级指令 3-3
  - 标签 13-1
    - 调用 13-4
    - 返回 13-5
    - 跳转 13-2
- 表述一览 26-12
- 步骤 附录-11
- C
  - CC= 和 CC>= 指令 4-23
  - CMP< 6-1
  - CMP<= 6-2
  - CMP<> 6-1
  - CMP= 6-1
  - CMP> 6-1
  - CMP>= 6-2
  - CNT、CDP 和 CUD 指令 4-14
  - CNTD,CDPD,CUDD( 双字计数器 ) 4-18
  - COMRF 14-5
  - COS 21-4
  - CSV 文件 25-12, 25-20
  - CVDT 10-19
  - CVXTY 16-3
  - CVYTX 16-4
  - 操作码 3-7
  - 常数 26-18
  - 常用对数 22-2
  - 乘 7-1
  - 程序分支
    - 指令 13-1, 14-1
  - 除 7-1
  - 存储定时器 / 计数器当前值 5-19
  - 错误
    - 代码
      - 用户程序执行 附录-19
- D
  - DC= 和 DC>= 指令 4-25
  - DEC 7-14
  - DECO 10-16
  - decrement 7-14
  - DEG 21-2
  - DIV 7-1
  - DTCB
    - 数据组合 10-22
  - DTDV ( 数据分割 ) 10-21
  - DTIM 20-1
  - DTMH 20-1
  - DTML 20-1
  - DTMS 20-1
  - 点写入 4-22
  - 电源 22-4
    - 故障
      - 内存保护 4-13
  - 定时器
    - 100-ms 关断延时 4-10
    - 10-ms 关断延时 4-10
    - 1-ms 关断延时 4-11
    - 1-s 关断延时 4-10
  - 或计数器
    - 作为目标设备 3-7
    - 作为源设备 3-7
  - 准确性 4-12
  - 对数 / 乘方
    - 指令 22-1

- 多重
  - OUT 和 OUTN 4-2
  - 使用 MCS 指令 4-34
- E** ENCO 10-15
- END
  - 指令 4-37
- EXP 22-3
- F** 方向控制 18-33
- FIEX 23-3
- FIFO 格式 23-1
- FIFO 23-1
- 反向
  - 移位寄存器 4-29
- 反余弦 21-7
- 反正切 21-8
- 反正弦 21-6
- 方向控制 18-21, 18-53
- 复位
  - 输入 4-27
- G** 高级指令
  - ACOS 21-7
  - ADD 7-1
  - ALT 10-18
  - ANDW 8-1
  - ARAMP 18-47
  - ASIN 21-6
  - ATAN 21-8
  - ATOB 10-12
  - ATOH 10-7
  - AVRG 17-1
  - BCDLS 9-5
  - BCNT 10-17
  - BMOV 5-12
  - BTOA 10-9
  - BTOH 10-3
  - 表 3-3
  - CMP< 6-1
  - CMP<= 6-2
  - CMP<> 6-1
  - CMP= 6-1
  - CMP> 6-1
  - CMP>= 6-2
  - COMRF 14-5
  - COS 21-4
  - CVDT 10-19
  - CVXTY 16-3
  - CVYTX 16-4
  - DECO 10-16
  - DEG 21-2
  - DIV 7-1
  - DLOG 25-1
  - DTCB 10-22
  - DTIM 20-1
  - DTMH 20-1
  - DTML 20-1
  - DTMS 20-1
  - ENCO 10-15
  - EXP 22-3
  - FIEX 23-3
  - FIFO 23-1
  - HOUR 24-12
  - HSCRFB 14-3
  - HTOA 10-5
  - HTOB 10-1
  - HTOS 24-11
  - IBMV 5-13
  - IBMVN 5-15
  - ICMP>= 6-6
  - IMOV 5-6
  - IMOVN 5-8
  - 结构 3-7
  - LABEL 13-1
  - LC< 6-8
  - LC<= 6-8
  - LC<> 6-8
  - LC= 6-8
  - LC> 6-8
  - LC>= 6-8
  - LCAL 13-4
  - LJMP 13-2
  - LOG10 22-2
  - LOGE 22-1
  - LRET 13-5
  - MOV 5-1
  - MOVC 5-9
  - MOVN 5-5
  - MUL 7-1
  - NRS 5-17
  - NSET 5-16
  - ORW 8-1
  - POW 22-4
  - PULS 18-1
  - PWM 18-9
  - RAD 21-1
  - RAMP 18-17
  - RAMPL 18-29
  - ROOT 7-15
  - ROTL 9-8
  - ROTR 9-10
  - SFTL 9-1
  - SFTR 9-3
  - SIN 21-3
  - STOH 24-12
  - SWAP 10-23
  - SUB 7-1
  - 数据类型 3-8
  - 输入条件 3-7
  - TADD 24-1
  - TAN 21-5
  - TCCST 5-19
  - TRACE 25-16
  - TTIM 20-3
  - WEEK 11-3
  - WSFT 9-7
  - XORW 8-1
  - XYFS 16-1
  - YEAR 11-16
  - ZRN 18-37
  - 高级指令的结构 3-7
  - 高速计数器
    - 刷新 14-3
  - 格式编号 16-1, 16-3, 16-4

更改  
 计时器和计数器的预置值 4-22  
 功能  
 列表 26-10  
 关断延时  
 100-ms 定时器 4-10  
 10-ms 定时器 4-10  
 1-ms 定时器 4-11  
 1-s 定时器 4-10  
 关于工作区显示的窗口 1-14  
 关于运算符优先级 26-43

**H**

HEX  
 到 ASCII 10-5  
 到 BCD 10-1  
 HMS → 秒 24-11  
 HOUR 24-12  
 HSCRF 14-3  
 HTOA 10-5  
 HTOB 10-1  
 HTOS 24-11  
 还原定时器 / 计数器预置值 4-22  
 函数 26-14  
 弧度 21-1  
 或 8-1

**J**

IO  
 刷新 14-1  
 IBMV 5-13  
 IBMVN 5-15  
 基本  
 指令 3-1, 4-1  
 基本操作 1-1  
 基本指令  
 TIMO 4-10  
 TMHO 4-10  
 TMLO 4-10  
 TMSO 4-11  
 ICMP>= 6-6  
 寄存器  
 比较指令 4-25  
 借位 2-4, 2-21  
 进位 2-4, 2-21  
 进位 / 借位 2-4, 2-21  
 IMOV 5-6  
 IMOVN 5-8  
 JMP 和 JEND 指令 4-35  
 JOG 运行 (JOG) 18-72  
 计数器  
 比较指令 4-23  
 和主控电路中的移位寄存器 4-34  
 加 / 减 4-15  
 加 / 减切换 4-16  
 双字加 / 减计数器 4-19  
 双字加 / 减切换计数器 4-19  
 双字加计数器 4-18  
 添加 (上) 计数器 4-14  
 加 7-1  
 加 / 减计数器 CDP 4-15  
 加 / 减切换计数器 CUD 4-16  
 加计数器 CNT 4-14  
 减 7-1  
 间隔比较大于或等于 6-6

间接  
 求反传送 5-8  
 位求反传送 5-15  
 位传送 5-13  
 传送 5-6  
 间接指定 3-13  
 监控操作 1-12  
 脚本 (SCRPT) 26-1  
 脚本编辑器 26-8  
 脚本的记述方法 26-12  
 脚本的记述例 26-19  
 脚本管理器 26-7  
 角度 21-2  
 交换 5-18  
 XCHG 5-18  
 交替输出 10-18  
 接点比较  
 不等于 6-8  
 大于 6-8  
 大于或等于 6-8  
 等于 6-8  
 小于 6-8  
 小于或等于 6-8  
 解码 10-16  
 结束

处理时间, 细分 附录-1  
 结束处理时间细分 附录-1  
 借位 3-11  
 进位 3-11  
 进位或借位信号 7-2  
 禁止  
 指令 4-38

**K**

控制信号状态 2-44  
 开始  
 WindLDR 1-1, 1-4, 1-8  
 控制语句 26-12, 26-19  
 块传送 5-12

**L**

LABEL 13-1  
 LC< 6-8  
 LC<= 6-8  
 LC<> 6-8  
 LC= 6-8  
 LC> 6-8  
 LC>= 6-8  
 LCAL 13-4  
 LJMP 13-2  
 LOD 和 LODN 指令 1-1, 1-4, 1-8, 4-1, 4-3, 4-4, 4-5, 4-6, 4-7, 4-8, 4-10  
 LOG10 22-2  
 LOGE 22-1  
 LRET 13-5  
 列表  
 基本指令 3-1  
 逻辑运算 26-25  
 逻辑运算符 26-13  
 逻辑运算指令 8-1

**M**

脉冲流量累积 (FLWP) 27-19  
 MCS 和 MCR 指令 4-33  
 模拟流量累积 (FLWA) 27-10  
 MOV 5-1

- MOVN 5-5  
 MUL 7-1  
 脉冲  
   输入 4-27  
 脉冲监控 18-79  
 秒→HMS 24-12  
 目标设备 3-7
- N**
- N 数据置位 5-16  
 N 数据重复置位 5-17  
 NRS 5-17  
 NSET 5-16
- O**
- Offset 26-16  
 OR LOD 指令 4-6  
 OR 和 ORN 指令 4-5  
 ORW 8-1  
 OUT 和 OUTN  
   多重 4-2  
   指令 4-1
- P**
- PIDA (PID 控制) 19-4  
 PIDD (微分衰减的 PID) 19-28  
 PLC 选择 1-2  
 PLC 状态 4-22  
 POW 22-4  
 平均化 17-1
- Q**
- 启动 / 停止  
   操作 1-19  
 清除  
   更改后的预置值 4-22  
   清除按钮 4-22  
 确认  
   按钮 4-22  
 确认更改  
   预置值 4-22
- R**
- RAD 21-1  
 RAMPL 18-29  
 random 7-19  
 日历定时器 (WKTIM) 11-1  
 RNDM 7-19  
 ROOT 7-15  
 ROTL 9-8  
 ROTR 9-10  
 RS232C DSR 控制信号状态 2-46  
 RS232C DTR 输出控制信号选项 2-48
- S**
- SET 和 RST 指令 4-3  
 SFR 和 SFRN 指令 4-27  
 SFTL 9-1  
 SFTR 9-3  
 设备地址 2-1  
 SIN 21-3  
 SOTU 和 SOTD 指令 4-31  
 special internal relay  
   for interrupt status 15-2  
 STOH 24-12  
 SWAP  
   数据交换 10-23  
 SUB 7-1  
 SUM 7-16
- sum 7-16  
 三角函数  
   指令 21-1  
 上升沿微分指令 4-31  
 设备 26-18  
   区域中断 3-11  
 设备区域中断 3-11  
 时间加法 24-1  
 示教定时器 20-3  
 十进制值和十六进制存储 3-8  
 十六进制存储十进制值 3-8  
 时钟指令 24-1  
 数据  
   比较指令 6-1  
   高级指令的类型 3-8  
   类型 3-7  
   输入 4-27  
   相 12-20  
   转换指令 10-1  
   传送  
     定时器 / 计数器预置值 4-22  
 数据的比较和复制 26-16  
 数据类型指定 26-17, 26-41  
 数据类型转换 26-15  
 输入  
   高级指令的条件 3-7  
 栓锁相 12-20  
 双 / 示教定时器指令 20-1  
 双向移位寄存器 4-30  
 双字加 / 减计数器 CDPD 4-19  
 双字加 / 减切换计数器 CUDD 4-19  
 双字加计数器 CNTD 4-18  
 四则运算法指令 7-1  
 算术运算 26-14  
 算术运算符 26-13
- T**
- TADD 24-1  
 TAN 21-5  
 TCCST 5-19  
 特殊内部继电器设备地址 2-2, 2-14  
 特殊数据寄存器一览 2-6, 2-28  
 TML、TIM、TMH 和 TMS 指令 4-8  
 TMLO, TIMO, TMHO, TMSO 指令 4-10, 4-14  
 TTIM 20-3  
 特殊日期 11-16  
 梯形阶 1-4  
 梯形图编写的限制 4-38  
 条件分支 26-12  
 跳转指令 4-35  
 通信盒信息 2-43  
 通信刷新 14-5
- W**
- WindLDR  
   开始 1-1, 1-4, 1-8  
 UMACRO 28-1  
 USB 1-10  
 WSFT 9-7  
 位函数 26-14, 26-29  
 位计数 10-17  
 位运算 26-27  
 位运算符 26-14  
 文件数据处理指令 23-1

- X**
- XCHG 5-18
  - XORW 8-1
  - 选项连接信息 2-9, 2-43
  - XYFS 16-1
  - XY 格式设置 16-1
  - 下载
    - 程序 1-10
  - 线性转换 16-6
  - 相关运算符 26-13
  - 小时计量器 24-12
  - 消息 (MSG) 12-1
  - 循环
    - 右 9-10
    - 左侧 9-8
- Y**
- 原点初始化 (ABS) 18-69
  - 异或 8-1
  - 移位
    - 右 9-3
    - 左侧 9-1
  - 移位 / 循环指令 9-1
  - 移位寄存器
    - 指令 4-27
  - 用户程序
    - 执行错误 附录-19
  - 用户程序执行错误 3-11
  - 用户程序执行错误记录 附录-20
  - 用户定义宏 28-1
  - 右移移位寄存器 4-27
  - 与 8-1
  - 余弦 21-4
  - 预置
    - 值
      - 更改 4-22
      - 还原 4-22
  - 源
    - 和目标设备 3-7
    - 设备 3-7
  - 运算符 26-13
- Z**
- 直线插补控制 (RAMPL) 18-29
  - 周表 (WKTBL) 11-1
  - 暂存设备 26-18
  - 正切 21-5
  - 正弦 21-3
  - 直接指定 3-13
  - 指令
    - 程序分支 13-1, 14-1
    - 对数 / 乘方 22-1
    - 禁止 4-38
    - 逻辑运算 8-1
    - 平均 17-1
    - 三角函数 21-1
    - 数据比较 6-1
    - 数据转换 10-1
    - 双 / 示教定时器 20-1
    - 四则运算 7-1
    - 移位 / 循环 9-1
    - 传送 5-1
    - 坐标转换 16-1
  - 指令步骤数 附录-11
  - 指令的执行时间 附录-3
  - 指数 22-3
  - 执行先进 23-3
  - 重叠坐标 16-7
  - 中断
    - 程序适用性 附录-11
    - 中断程序中的适用性 附录-11
    - 中断和结束 26-13
  - 重复 26-12
    - 操作
      - ADD 和 SUB 指令 7-7
      - ANDW、ORW 和 XORW 指令 8-3
      - DIV 指令 7-11
      - 间接位传送指令 5-14
      - MUL 指令 7-9
      - 数据比较指令 6-4
      - 传送指令 5-3
    - 次数 3-7
    - 指定 3-7
  - 主机
    - 控制指令 4-33
  - 转换 12-20, 12-22
    - X ~ Y 16-3
    - 线性 16-6
    - Y ~ X 16-4
  - 转换数据类型 10-19
  - 传送 5-1
    - 求反 5-5
  - 传送字符 (MOVC) 5-9
  - 子程序 13-5
  - 字函数 26-14, 26-29
  - 自然对数 22-1
  - 字移位 9-7
  - 坐标转换指令 16-1







# 产品保修说明

## (1) 保修期限

客户所购产品的保修期限，自原始购买日，或产品送达指定地点之日起享有 3 年保修期。

\* 超出保修范围的情形

电池已达到使用寿命，以及继电器开闭次数超过限定值（10 万次）

## (2) 保修范围

在上述保修期限内发生非用户原因造成的故障，本公司将负责对相应的产品故障部分进行免费更换，或提供免费维修。

万一出现故障时，请您携带能够直接证明您购入本产品日期的材料，至销售门店或本公司申请保修服务。

\* 因更换产品所产生的安装及工程费用，需由客户自行承担。

## (3) 保修免责条款

以下情况所导致的故障不在保修范围内。

1) 在超出样本、性能规格书、使用说明书中记载的条件以及环境下使用产品的情况

2) 故障并非是由所购产品导致的情况

3) 产品经过非本公司人员改造或修理的情况

4) 将产品用于原设计用途以外用途的情况

5) 因火灾、地震、水灾、闪电及其他自然灾害，或异常电压（电压频率）等，非本公司原因导致的故障及损坏等

6) 购入后因移机、搬运、掉落等导致的故障及损坏

7) 因安装不当导致的故障及损坏

8) 因未遵照使用说明书中规定的维护与检修所导致的故障及损坏

\* 客户有责任检查任何可编程产品的操作，风险自担。在任何情况下，本公司不对客户所设定的程序操作或因操作而造成的损失负责。

此外，本条款中涉及的保修，仅针对产品个体本身，对于因产品故障而间接产生的损失，请恕本公司概不负责。

## (4) 有偿服务项目

产品价格中，不含技术人员外派等服务费用，下列费用需由客户自行承担。

1) 安装调试指导及试运转跟进服务（含应用程序编程、操作试验等）

2) 维护检修、调整及维修

3) 技术指导及技术教育

4) 应客户要求实施的产品试验及检查

## IDEC 株式会社

日本大阪府大阪市淀川区西宫原 2-6-64



IDEC China Apps

爱德克电气贸易（上海）有限公司

北京分公司

广州分公司

香港和泉电气有限公司

 idecchina.cn

200070 上海市静安区共和路 209 号 企业中心第二座 8 楼  
电话：021-6135-1515 传真：021-6135-6225/6226 E-mail: idec@cn.idec.com

100026 北京市朝阳区光华路甲 8 号 和乔大厦 B 座 310 室  
电话：010-6581-6131 传真：010-6581-5119

510610 广州市天河区林和西路 157 号 保利中汇广场 A 栋 907 号  
电话：020-8362-2394 传真：020-8362-2394

香港九龙观塘观塘道 370 号 创纪之城 3 期 16 楼 01 室  
电话：852-2803-8989 传真：852-2565-0171/2561-8732 E-mail: info@hk.idec.com

- 本手册内所记载的公司名称以及商品名称，为各公司的注册商标。
- 本手册中的规格及其他说明若有改变，恕不另行通知。

B-2343 (0) 本资料记载内容为 2023 年 11 月的信息。

