

客服热线 400-820-9595

绵密网络 专业服务

中达电通已建立了 48 个分支机构及服务网点，并塑建训练有素的专业团队，提供客户最满意的服务，公司技术人员能在 2 小时内回应您的问题，并在 48 小时内提供所需服务。

上海 电话:(021)6301-2827 传真:(021)6301-2307	南昌 电话:(0791)8625-5010 传真:(0791)8625-5102	合肥 电话:(0551)6281-6777 传真:(0551)6281-6555	南京 电话:(025)8334-6585 传真:(025)8334-6554	杭州 电话:(0571)8882-0610 传真:(0571)8882-0603
武汉 电话:(027)8544-8265 传真:(027)8544-9500	长沙 电话:(0731)8827-7881 传真:(0731)8827-7882	南宁 电话:(0771)5879-599 传真:(0771)2621-502	厦门 电话:(0592)5313-601 传真:(0592)5313-628	广州 电话:(020)3879-2175 传真:(020)3879-2178
济南 电话:(0531)8690-7277 传真:(0531)8690-7099	郑州 电话:(0371)6384-2772 传真:(0371)6384-2656	北京 电话:(010)8225-3225 传真:(010)8225-2308	天津 电话:(022)2301-5082 传真:(022)2335-5006	太原 电话:(0351)4039-475 传真:(0351)4039-047
乌鲁木齐 电话:(0991)6118-160 传真:(0991)6118-289	西安 电话:(029)8669-0780 传真:(029)86690780-8000	成都 电话:(028)8434-2075 传真:(028)8434-2073	重庆 电话:(023)8806-0306 传真:(023)8806-0776	哈尔滨 电话:(0451)5366-0643 传真:(0451)5366-0248
沈阳 电话:(024)2334-1612 传真:(024)2334-1163	长春 电话:(0431)8892-5060 传真:(0431)8892-5065			

AH Motion Controller 标准指令手册



AH Motion Controller 标准指令手册



中达电通股份有限公司

地址：上海市浦东新区民夏路238号
邮编：201209
电话：(021)5863-5678
传真：(021)5863-0003
网址：<http://www.deltagreentech.com.cn>

AH-0259710-03
2018/3/31

中达电通公司版权所有
如有改动,恕不另行通知

www.deltaww.com



AH Motion Controller 标准指令手册

版本修订一览表

版本	变更内容	发行日期
第一版	第一版发行	2016/12/09
第二版	<ol style="list-style-type: none">第 2 章删除 TR/CR/HCR 指令、增加以太网 Socket 相关指令、删除 AR/AM 相关数据、新增数据转换指令 FTODF 和 DFTOF第 3 章增加以太网 Socket 相关指令、新增数据转换指令 FTODF 和 DFTOF附录增加 SM7/SM450/SM451/SM1001/SM1196 /SM1270-SM1390、SR440-SR451/ SR1118-SR1320 相关资料；增加指令 DFTOF、FTOF、SCLOSE、SOPEN、SEND、SRCVD；错误码与故障排除部份增加 Gcode、SDO、CamIn/GearIn 相关部份	2017/08/31
第三版	<ol style="list-style-type: none">第 2 章修改 INV 的图形第 3 章修改 D/的超连结及 INT、FINT 的数据类型附录修改错误码灯号及排除方式	2018/03/31

AH Motion Controller 标准指令手册

目录

前言

P.1 简介	II
P.1.1 适用产品	II
P.1.2 相关手册	II
P.2 关于手册间的交互参照	III

第 1 章 标准指令简介

1.1 概述	1-2
1.2 标准指令分类	1-2

第 2 章 装置、变量符号和指令

2.1 装置	2-3
2.1.1 装置功能说明	2-3
2.1.2 装置列表	2-3
2.1.3 停电保持装置	2-5
2.1.4 输入继电器 (X)	2-6
2.1.5 输出继电器 (Y)	2-7
2.1.6 辅助继电器 (M)	2-7
2.1.7 特殊辅助继电器 (SM/AM)	2-7
2.1.8 数据寄存器 (D)	2-7
2.1.9 特殊数据寄存器 (SR/AR)	2-8
2.1.10 链接寄存器 (L)	2-8
2.1.11 步进点继电器 (S)	2-8
2.1.12 定时器 (T)	2-8
2.1.13 计数器 (C)	2-9
2.1.14 32 位计数器 (HC/AC)	2-9
2.1.15 变址寄存器 (E)	2-9
2.1.16 数值、常数 (K · 16#)	2-9
2.1.17 浮点数 (F · DF)	2-10
2.1.18 字符串 ("\$")	2-10

2.1.19 指标寄存器 (PR)	2-10
2.1.19.1 计时器指标寄存器 (T_Pointer)(TR)	2-10
2.1.19.2 16 位元计数器指标寄存器 (C_Pointer)(CR)	2-10
2.1.19.3 32 位元计数器指标寄存器 (HC_Pointer)(HCR)	2-11
2.2 EtherCAT 符号	2-11
2.3 变量符号	2-12
2.3.1 变量符号的作用范围	2-12
2.3.2 变量符号的类别	2-12
2.3.3 变量符号的数据类型	2-13
2.3.4 变量符号的地址配置与初始值	2-14
2.3.5 变量符号的变址操作	2-15
2.4 自定义数据类型 (DUT): ENUM	2-16
2.4 指令	2-18
2.4.1 标准指令分类	2-18
2.4.2 标准指令列表	2-19

第 3 章 标准指令

3.1 指令阅读方式	3-3
3.1.1 指令组成说明	3-3
3.1.2 指令使用限制	3-9
3.2 梯形图指令	3-11
3.3 比较操作指令	3-36
3.4 四则运算指令	3-73
3.5 数据转换指令	3-113
3.6 数据转移指令	3-159
3.7 程序跳转指令	3-185
3.8 程序执行控制指令	3-194
3.9 I/O 更新指令	3-202
3.10 便利指令	3-205
3.11 逻辑操作指令	3-241
3.12 循环指令	3-265
3.13 基本指令	3-277

3.14 位移指令	3-286
3.15 数据处理指令	3-320
3.16 结构建立指令	3-372
3.17 模块的数据读写指令	3-380
3.18 浮点数指令	3-386
3.19 万年历指令	3-432
3.20 外围设备指令	3-447
3.21 通讯指令	3-463
3.22 其它指令	3-489
3.23 字符串处理指令	3-502
3.24 以太网控制指令	3-566
3.25 储存卡读写指令	3-591
3.26 任务控制指令	3-604
3.27 SFC 控制指令	3-609

附录 A

A.1 特殊辅助继电器 (SM)	A-3
A.1.1 特殊辅助继电器 SM 的更新时间	A-34
A.2 特殊数据寄存器 (SR)	A-38
A.2.1 特殊数据寄存器 SR 的更新时间	A-61
A.3 SM/SR 补充说明	A-64
A.4 标准指令 (依英文字母排序)	A-77
A.5 错误码与故障排除	A-86
A.5.1 错误码与对应灯号状态说明	A-86
AHxxEMC-5A	A-87
对应之错误信息表	A-87
错误码和灯号指示说明	A-88
模拟 I/O 模块及温度模块	A-100
AH02HC-5A/AH04HC-5A	A-101
AH05PM-5A/AH10PM-5A/AH15PM-5A	A-102
AH20MC-5A	A-103
AH10EN-5A/AH15EN-5A	A-104
AH10SCM-5A/AH15SCM-5A	A-104

AH10DNET-5A	A-104
AH10PFBM-5A	A-105
AH10PFBS-5A.....	A-107
AH10COPM-5A.....	A-106
A.5.2 错误码与故障排除.....	A-108
AHxxEMC-5A	A-108
ERROR 灯号常亮.....	A-108
ERROR 灯号闪烁.....	A-109
BUS FAULT 灯号常亮.....	A-126
BUS FAULT 灯号闪烁.....	A-127
其它	A-127
模拟 I/O 模块及温度模块故障排除	A-138
AH02HC-5A/AH04HC-5A	A-140
AH05PM-5A/AH10PM-5A/AH15PM-5A.....	A-141
AH20MC-5A	A-142
AH10EN-5A/AH15EN-5A.....	A-144
AH10SCM-5A/AH15SCM-5A.....	A-144
AH10DNET-5A	A-144
AH10PFBM-5A	A-145
AH10PFBS-5A.....	A-146
AH10COPM-5A.....	A-146
A.5.3 极限错误排除	A-147
软极限错误排除.....	A-147
硬极限错误排除.....	A-148
A.6 数据类型：列举 (Enum)	A-149



前言

目录

P.1	简介	II
P.1.1	适用产品	II
P.1.2	相关手册	II
P.2	关于手册间的交互参照	III

P.1 简介

感谢您购买 AH Motion Controller 系列运动控制器，并采用我们为您提供的高端运动控制系统。

本手册介绍了台达提供的标准指令，包含梯形图常用的基本指令以及丰富的应用指令。请确认您对于 AH Motion Controller 运动控制系统的配置以及操作有充分的了解，以便正确地使用 AH Motion Controller 系列运动控制 CPU。

您可在 AH Motion Controller 系列手册以及其他相关手册间交互参照，以便取得配合您的系统配置所需的相关内容。

P.1.1 适用产品

本手册适用或与以下产品相关：

- AHxxEMC-5A (AH08EMC-5A / AH10EMC-5A / AH20EMC-5A)

P.1.2 相关手册

AH Motion Controller 系列运动控制器的相关手册组成如下：

1. **AH Motion Controller 硬件手册**
介绍硬件规格、电气及功能规格、外观、尺寸等等。
2. **ISPSOft 使用手册**
内容包含ISPSOft软件操作方式、程序编辑语言(梯形图、顺序功能图、ST (Structured Text) 和功能块、程序组织单元 (POU) 以及任务 (Task) 的概念、以及运动控制程序的编辑方式。
3. **AH Motion Controller 标准指令手册**
说明编辑PLC程序所需使用的装置、符号，以及标准PLC指令等等。
4. **AH Motion Controller 操作手册**
介绍运动控制系统概念、软硬件设定、软件操作简介、装置说明、运动控制程序架构、故障排除等等信息。
5. **AH Motion Controller 运动控制指令手册**
介绍运动控制程序编辑所需使用的装置、轴参数、符号以及单轴/多轴运动控制指令。
6. **AH500 运动控制模块手册**
提供关于AH500系列运动控制模块的功能、规格、配线方式以及指令。
7. **AH500 模块手册**
介绍AH500系列特殊模块的使用方式，包含网络模块、模拟输入 / 输出模块、温度量测模块等等。

P.2 关于手册间的交互参照

在开始使用本产品之前，请先详阅作为基础知识需参考的三本手册，分别是 **AH Motion Controller 硬件手册**、**ISPSOft 使用手册**、以及 **AH Motion Controller 标准指令手册**。

藉由说明基础知识的三本手册，您可以了解硬件架构配置的方式、软件操作方式、以及如何使用基本指令来使用本系统。

了解您所需的系统配置后，可依适合您的系统配置，参阅下表的指引来交互参照各手册来获得您需要的信息。详阅所有和您系统配置相关的手册，可确保正确使用本产品，并发挥 AH 系列运动控制系统的最大性能。

相关手册	AH Motion Controller 系列手册						
	基础知识			AH Motion Controller 操作手册	AH Motion Controller 运动控制指令手册	AH500 运动控制模块手册	AH500 模块手册
	AH Motion Controller 硬件手册	ISPSOft 使用手册	AH Motion Controller 标准指令手册				
操作步骤概要							
1. AH Motion Controller 系统架构及产品概观	V						
2. 系统硬件配置	V						
运动控制应用相关				V			
通讯功能相关 (如: EtherCAT)							
增加运动控制模块以扩展运动控制功能						V	
扩展 AH500 系列输入/输出模块							V
3. 开始使用软件	V						
运动控制应用相关				V			
通讯功能相关 (如: EtherCAT)							
增加运动控制模块以扩展运动控制功能						V	
扩展 AH500 系列输入/输出模块							V
4. 编辑程序	V		V				
运动控制应用相关				V	V		
通讯功能相关 (如: EtherCAT)							

P

操作步骤概要	相关手册	AH Motion Controller 系列手册						
		基础知识			AH Motion Controller 操作手册	AH Motion Controller 运动控制指令手册	AH500 运动控制模块手册	AH500 模块手册
		AH Motion Controller 硬件手册	ISPSOft 使用手册	AH Motion Controller 标准指令手册				
增加运动控制模块以扩展运动控制功能						V		
扩展 AH500 系列输入/输出模块							V	
5. 测试与故障排除								
运动控制应用相关					V*			
通讯功能相关 (如: EtherCAT)		V		V				
增加运动控制模块以扩展运动控制功能			V*			V		
扩展 AH500 系列输入/输出模块			V*				V	
6. 维护与检修	V							

***注：**错误码、指示灯以及对应的错误排除相关信息可在此手册中的附录快速查询。完整的系统错误以及故障排除内容，请参考 **AH Motion Controller 操作手册**。

第1章 标准指令简介

目录

1.1 概述.....	1-2
1.2 标准指令分类.....	1-2

1.1 概述

本手册描述标准程序设计相关的内容介绍，其中包含装置、变量符号与标准指令。

大部分的指令使用操作数作为输入与输出，操作数需连接指定的装置，如 X、Y、D、SM 及 SR 等或变量符号。而变量符号所配置的装置将根据设定时所指定的数据类型来决定。

在使用指令前，请确认已彻底了解装置、变量符号和标准指令使用方式。

你也可透过手册的附录来快速索引特殊数据寄存器（SR）、特殊辅助继电器（SM）、标准指令列表以及错误码。

1.2 标准指令分类

分类	说明
梯形图指令	基本常用的 Ladder 编译指令如 LD、LDP、AND、OR、OUT 等
比较操作指令	比较操作如=、<>、>、>=、<、<=...等
四则运算指令	BIN 或 BCD 的加法、减法、乘法或除法
数据转换指令	将 BCD 转换成 BIN 和将 BIN 转换成 BCD
数据转移指令	传送指定的数据
程序跳转指令	程序跳转
程序执行控制指令	允许或禁止中断程序
I/O 更新指令	运行局部更新
便利指令	用于以下目的的指令：计数器增加/减小、教学定时器、特殊功能定时器、旋转台最短距离控制等
逻辑操作指令	逻辑操作，如逻辑加法、逻辑乘法等
旋转指令	指定数据的旋转移位
基本指令	定时器/计数器指令
移位指令	指定数据的移位
数据处理指令	16 位数据查询，数据处理如译码和编码
结构建立指令	循环回路、调用功能块...等
模块的数据读/写指令	特殊功能模块的数据读/写
浮点数指令	浮点数运算指令
万年历指令	万年历时间（年、月、日、小时、分、秒和星期）的读取、更新与比较
外围设备指令	连接到外围设备的 I/O
通讯指令	通讯控制接口设备
其它指令	其它不符合以上范畴的指令，如通讯逾时重置指令和定时时钟指令

分类	说明
字符串处理指令	CBIN/BCD 和 ASCII 之间的转换；BIN 和字符串之间的转换；浮点十进制数和字符串之间的转换、字符串处理等等
以太网控制指令	控制以太网数据交换相关指令
储存卡读写指令	储存卡读写相关指令
任务控制指令	程序中任务控制相关指令
SFC 控制指令	控制 SFC 相关指令

MEMO

1

第2章 装置、变量符号和指令

目录

2.1	装置.....	2-3
2.1.1	装置功能说明.....	2-3
2.1.2	装置列表	2-3
2.1.3	停电保持装置.....	2-5
2.1.4	输入继电器 (X)	2-6
2.1.5	输出继电器 (Y)	2-7
2.1.6	辅助继电器 (M)	2-7
2.1.7	特殊辅助继电器 (SM)	2-7
2.1.8	数据寄存器 (D)	2-7
2.1.9	特殊数据寄存器 (SR)	2-8
2.1.10	链接寄存器 (L)	2-8
2.1.11	步进点继电器 (S)	2-8
2.1.12	定时器 (T)	2-8
2.1.13	计数器 (C)	2-9
2.1.14	32 位计数器 (HC/AC)	2-9
2.1.15	变址寄存器 (E)	2-9
2.1.16	数值、常数 (K · 16#)	2-9
2.1.17	浮点数 (F · DF)	2-10
2.1.18	字符串 ("\$")	2-10
2.1.19	指标寄存器 (PR)	2-10
2.1.19.1	计时器指标寄存器 (T_Pointer) (TR)	2-10
2.1.19.2	16 位元计数器指标寄存器 (C_Pointer) (CR)	2-10
2.1.19.3	32 位元计数器指标寄存器 (HC_Pointer) (HCR)	2-11
2.2	EtherCAT 符号.....	2-11

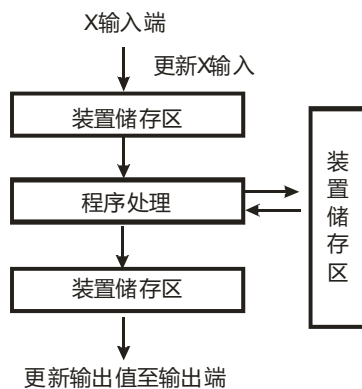
2.3 变量符号	2-12
2.3.1 变量符号的作用范围	2-12
2.3.2 变量符号的类别	2-12
2.3.3 变量符号的数据类型	2-13
2.3.4 变量符号的地址配置与初始值	2-14
2.3.5 变量符号的变址操作	2-15
2.4 自定义数据类型 (DUT) : ENUM.....	2-16
2.5 指令	2-18
2.5.1 标准指令分类	2-18
2.5.2 标准指令列表	2-19

2.1 装置

本章节针对 PLC 所处理的数值、字符串和输入、输出、辅助继电器、定时器、计数器及数据寄存器等各种装置的配置和功能做说明。关于装置更详细的说明可参考 **AH Motion Controller 操作手册**。

2.1.1 装置功能说明

PLC 对于程序的处理流程（结束再生方式）：



- 更新输入信号：
 1. PLC 在执行程序之前会将外部输入信号状态读入至输入信号储存区内。
 2. 在程序执行中若输入信号作 ON/OFF 变化，但是输入信号储存区内的状态不会改变，一直到下一次扫描开始才会再更新输入信号。
- 程序处理：

PLC 更新输入信号后，开始从程序的起始地址依序执行程序中的每一指令，其处理结果存入各装置储存区。
- 更新输出状态：

当执行到 END 指令后将装置储存区内的状态送到用户所分配的输出端。

2.1.2 装置列表

形式	装置名称		装置数	范围
位装置	输入继电器	X	8192	X0.0~X511.15 (支持强制位 ON/OFF)
	输出继电器	Y	8192	Y0.0~Y511.15 (支持强制位 ON/OFF)
	数据寄存器	D	1048576	D0.0~D65535.15
	链接寄存器	L	1048576	L0.0~L65535.15
	辅助继电器	M	8192	M0~M8191
	特殊辅助标志	SM	SM : 2048	SM0~SM2047
	步进点继电器	S	2048	S0~S2047
	定时器	T	2048	T0~T2047
	计数器	C	2048	C0~C2047
字符	32 位计数器	HC/AC	HC : 64 AC : 56 (AH10EMC)	HC0~HC63 AC0~AC55 (AH10EMC)
		输入继电器	X	512

形式	装置名称		装置数	范围
装置	输出继电器	Y	512	Y0~Y511
	数据寄存器	D	65536	D0~D65535
	特殊数据寄存器	SR	SR : 2048	SR0~SR2047
	链接寄存器	L	65536	L0~L65535
	定时器	T	2048	T0~T2047
	计数器	C	2048	C0~C2047
	32 位计数器	HC/AC	HC : 64 AC : 56 (AH10EMC)	HC0~HC63 AC0~AC55 (AH10EMC)
	变址寄存器	E	32	E0~E31
常数*	十进制	K	16 位 : -32768~32767 32 位 : -2147483648~2147483647	
	十六进制	16#	16 位 : 16#0~16#FFFF 32 位 : 16#0~16#FFFFFFFF	
	单精度浮点数	F	32位 : $\pm 1.17549435^{-38} \sim \pm 3.40282347^{+38}$	
	双精度浮点数	DF	64 位 : $\pm 2.2250738585072014^{-308} \sim \pm 1.7976931348623157^{+308}$	
字符串*	字符串	"\$"	1~31 字符 (characters)	
指针*	指针寄存器 PR	PR		

*1 : 十进制的表示方式，在第3章的指令装置表中以K来表示，但在ISPSoft中直接输入值，例如K50，请直接输入50。

*2 : 浮点数的表示方式，在第3章的指令装置表中以F/DF来表示，但在ISPSoft中是直接以小数点的方式来表示，例如要输入F500的浮点数，请直接输入500.0。

*3 : 字符串的表示方式 在第3章的指令装置表中以"\$"来表示，但在ISPSoft中是以" "方式来表示，例如要输入字符串1234，请直接输入"1234"。

2.1.3 停电保持装置

- 停电保持区的装置范围

装置	功能	装置范围	停电保持区范围
X	输入继电器	X devices (bit) : X0.0~X511.15 X devices (word) : X0~X511	固定非停电保持
Y	输出继电器	Y devices (bit) : Y0.0~Y511.15 Y devices (word) : Y0~Y511	固定非停电保持
M*	辅助继电器	M0~M8191	默认 M0~M8191
SM	特殊辅助继电器	SM: SM0~SM2047	部分停电保持并且不能被改变 详细内容请参考 SM 菜单
S	步进点继电器	S0~S2047	固定非停电保持
T*	定时器	T0~T2047	默认 T0~T2047.
C*	计数器	C0~C2047	默认 C0~C2047.
HC/AC*	32 位计数器	HC: HC0~HC63 AC: AC0~ AC55 (AH10EMC)	默认 HC0~HC63. AC 装置固定非停电保持.
D*	数据寄存器	D device (bit) : D0.0~D65535.15 D device (word) : D0~D65535	默认 D0~D32767 最多可以设定 32768 个
SR	特殊数据寄存器	SR: SR0~SR2047	部分停电保持并且不能被改变 详细内容请参考 SR 菜单
L	链接寄存器	L0~ L65535	固定非停电保持
E	变址寄存器	E0~E31	固定非停电保持

*注：M·T·C·D 装置可由用户设定停电保持区的范围。可以设定此装置不停电保持，而设定的范围最大不能超过装置范围，其中 D 装置最多只能设定 32768 个 D 装置，例如：可以设定 D50~D32817 为停电保持区或设定 D32768~D65535 为停电保持区，而其默认为 D0~D32767 为停电保持区。

- 停电保持储存方式

PLC 动作		内存类型	非停电保持区	停电保持区	Y 装置
电源 OFF=>ON			清除	保持	清除
STOP=>RUN	设定 Y 装置清除		保持	保持	清除
	设定 Y 装置保持		保持	保持	保持
	设定 Y 装置回复 STOP 前状态		保持	保持	回复 STOP 前状态
STOP=>RUN	设定非停电保持区清除		清除	保持	参照 Y 装置的设定
	设定非停电保持区保持		保持	保持	参照 Y 装置的设定
RUN=>STOP			保持	保持	保持
SM204ON (清除所有的非停电保持区域)			清除	保持	清除
SM205ON (清除所有停电保持区域)			保持	清除	保持
出厂设定值			0	0	0

2.1.4 输入继电器 (X)

- 输入接点 X 的功能：

输入接点 X 与输入设备 (按钮开关·旋钮开关·数字开关等的外部设备) 连接·读取输入信号进入 PLC。每一个输入接点 X 的 A 或 B 接点在程序中使用次数没有限制。输入接点 X 的 ON/OFF 只会跟随输入设备的 ON/OFF 做变化。

- 输入接点的编号 (以十进制编号)：

对 PLC 系列而言·输入端的编号固定从 X0.0 开始算·编号的多寡跟随 DIO 模块的输入点数大小而变化·随着与主机的连接顺序来推算出。PLC 机种最大输入点数可达 8192 点·范围如下：X0.0 ~ X511.15。

- 输入的种类：

输入有刷新输入和直接输入 2 种

1. 刷新输入：采用程序执行前的外部输入刷新时接收的 ON/OFF 数据来进行运算的输入方式 (如：LD X0.0)
2. 直接输入：采用指令执行时从外部输入接收的 ON/OFF 数据进行运算的输入方式 (如：LD DX0.0)

2.1.5 输出继电器 (Y)

- 输出接点 Y 的功能：

输出接点 Y 的任务就是送出 ON/OFF 信号来驱动连接输出接点 Y 的负载 (外部信号灯 · 数字显示器 · 电磁阀等) 。输出接点分成三种 · 一为继电器 (Relay) · 二为晶体管 (Transistor) · 三为交流硅控器 (TRIAC (Thyristors)) 。每一个输出接点 Y 的 A 或 B 接点在程序中使用次数没有限制 · 但输出 Y 的编号 · 在程序建议仅能使用一次 · 否则依 PLC 的程序扫描原理 · 其输出状态的决定权会落在程序中最后的输出 Y 的电路。

- 输出接点的编号 (以十进制编号) ：

对 PLC 系列而言 · 输出端的编号固定从 Y0.0 开始算 · 编号的多寡跟随 DIO 模块的输出点数大小而变化 · 随着与主机的连接顺序来推算出 · PLC 机种最大输出点数可达 8192 点 · 范围如下：Y0.0 ~ Y511.15 。未实际配置使用的 Y 编号可当作一般的装置用。

- 输出的种类：

输出有刷新输出和直接输出 2 种

1. 刷新输出：采用程序执行到 END 指令 · 依据 ON/OFF 数据来进行实际输出方式 (如：OUT Y0.0)
2. 直接输出：采用指令执行时 · 直接依据 ON/OFF 数据进行实际输出方式 (如：OUT DY0.0)

2.1.6 辅助继电器 (M)

辅助继电器 M 有 A、B 接点 · 而且在程序当中使用次数无限制 · 用户可利用辅助继电器 M 来组合控制回路 · 但无法直接驱动外部负载 · 依其性质可区分为下列二种：

1. 一般用：一般用辅助继电器在 PLC 运转时若遇到停电 · 其状态将全部被复位为 OFF · 再送电时其状态仍为 OFF。
2. 停电保持用：停电保持用辅助继电器在 PLC 运转时若遇到停电 · 其状态将全部被保持 · 再送电时其状态为停电前状态。

2.1.7 特殊辅助继电器 (SM)

SM：特殊辅助继电器

每一个特殊辅助继电器都有其特定的功用 · 未定义的特殊标志请勿使用。

关于 SM 装置的功能说明 · 请参考附录 A.1 特殊辅助继电器表

2.1.8 数据寄存器 (D)

用于储存数值数据 · 其数据长度为 16 位 (-32,768 ~ +32,767) · 最高位为正负号 · 可储存 -32,768 ~ +32,767 的数值数据 · 也可将两个 16 位寄存器合并成一个 32 位寄存器 (D+1 · D 编号小的为下 16 位) 使用 · 而其最高位为正负号 · 可储存 -2,147,483,648 ~ +2,147,483,647 的数值数据 · 也可将四个 16 位寄存器合并成一个 64 位寄存器 (D+3 · D+2 · D+1 · D 编号小的为下 16 位) 使用 · 而其最高位为正负号 · 可储存 -9,223,372,036,854,776 ~ +9,223,372,036,854,775,807。

寄存器依其性质可区分为下列二种：

一般用寄存器：当 PLC 由 STOP → RUN 或断电时 · 寄存器内的数值数据会被清除为 0 · 如果想要 PLC 由 STOP → RUN 时 · 数据会保持不被清除 · 请参考 ISPSOFT 使用手册的硬件组态说明 · 但断电时仍会被清除为 0。

停电保持用寄存器：当 PLC 断电时此区域的寄存器数据不会被清除，仍保持其断电前之数值。清除停电保持用寄存器的内容值，可使用 RST 或 ZRST 指令。

2.1.9 特殊数据寄存器 (SR)

SR：特殊数据寄存器。

每个特殊数据寄存器都有其特殊定义与用途，主要作为存放系统状态、错误信息、监视状态之用。此外特殊数据寄存器也被使用在运动控制上

关于特殊数据寄存器 (SR) 的内容说明，请参考附录A.2 特殊数据寄存器表

2.1.10 链接寄存器 (L)

L 装置主要用于数据交换功能，当 AH10EMC 对 AH10EMC 进行数据交换时，可以使用 L 装置作为数据交换的缓冲区，连接寄存器 L 的装置编号为 L0 ~ L65535 共 65536 个 Words，也可当作一般的辅助寄存器使用。

2.1.11 步进点继电器 (S)

步进点继电器的功能：

步进点继电器 S 在工程自动化控制中可轻易的设定程序，其为步进梯形图最基本的装置，使用在步进梯形图(或称顺序功能图，Sequential Function Chart，SFC)中，SFC 使用说明请参考 ISPSOFT 使用手册。

步进点继电器 S 的装置编号为 S0 ~ S2047 共 2048 点，各步进点继电器 S 与输出继电器 Y 一样有输出线圈及 A、B 接点，而且在程序当中使用次数无限制，但无法直接驱动外部负载。步进继电器 (S) 不用于步进梯形图时，可当作一般的辅助继电器使用。

2.1.12 定时器 (T)

- 100ms 定时器：TMR 指令所指定的 T 定时器以 100ms 为单位计时
- 1ms 定时器：TMRH 指令所指定的 T 定时器以 1ms 为单位计时。
- 子程序专用定时器为 T1920~T2047。
- 运算型 T 定时器为 ST0~ST2047，但若使用装置监控，就是监控 T0~T2047。
- 在程序中同一个 T 定时器如果重复使用 (包含使用在不同指令 TMR、TMRH 中)，则设定值以最快到达的为主。
- 在程序中同一个 T 定时器如果重复使用，其中一个条件接点 OFF 时则 T 会 OFF。
- 在程序中同一个 T 定时器如果重复使用为 T 与 ST，其中一个条件接点 OFF 时则 T 会 OFF。
- 当 T 定时器 ON->OFF 且条件式为 ON 时，T 计时值归零并重新计时。
- 当 TMR 指令执行时，其所指定的定时器线圈受电，定时器开始计时，当到达所指定的定时值 (计时值>=设定值)，其接点动作如下：

NO (Normally Open) 接点	开路
NC (Normally Closed) 接点	闭合

其余相关说明请参考AH Motion Controller操作手册 (第五章 一般装置)

2.1.13 计数器 (C)

计数器的功能：

计数器的计数脉冲输入信号由 OFF→ON 时，计数器当前值等于设定值时输出线圈导通，设定值为十进制常数，也可使用数据寄存器 D 当成设定值。

16 位计数器：

1. 16 位计数器的设定范围：0~32,767。(0 与 1 相同，在第一次计数时输出接点马上导通。)
2. 一般用计数器在 PLC 停电的时候，计数器当前值即被清除，若为停电保持型计数器会将停电前的当前值及计数器接点状态储存着，复电后会继续累计。
3. 若使用 MOV 指令，将大于设定值的数值传送到 C0 当前值寄存器时，在下次 X0.1 由 OFF→ON 时，C0 计数器接点即变成 ON，同时当前值内容变成与设定值相同。
4. 计数器的设定值可使用常数直接设定或使用寄存器 D 中的数值作间接设定。
5. 设定值可使用常数或使用数据寄存器 D 作为设定值可以是正负数。计数器当前值由 32,767 再往上累计时则变为-32,768。

其余相关说明请参考 **AH Motion Controller 操作手册** (第五章 一般装置)

2.1.14 32 位计数器 (HC/AC)

HC：32 位一般用加减法运算计数器：

1. 32 位一般用计数器的设定范围：-2,147,483,648~2,147,483,647。
2. 32 位一般用加减法运算计数器切换上下数用特殊辅助继电器，由 SM621~SM684 来决定。例：SM621=OFF 时决定 HC0 为加法运算，SM621=ON 时决定 HC0 为减法运算其余类推。
3. 设定值可使用常数或使用数据寄存器 D 作为设定值可以是正负数，若使用数据寄存器 D 则一个设定值占用两个连续的数据寄存器。
4. 一般用计数器在 PLC 停电的时候，计数器当前值即被清除，若为停电保持型计数器，则会将停电前的当前值及计数器接点状态储存着，复电后会继续累计。
5. 计数器当前值由 2,147,483,647 再往上累计时则变为-2,147,483,648。同理计数器当前值由-2,147,483,648 再往下递减时，则变为 2,147,483,647。

其余相关说明请参考 **AH Motion Controller 操作手册** (第五章 一般装置)

2.1.15 变址寄存器 (E)

变址寄存器 E 是 16 位的数据寄存器，跟一般的寄存器一样可以被读、写，但主要功能是做变址寄存器使用，使用范围为 E0~E13。

其余相关说明请参考 **AH Motion Controller 操作手册** (第五章 一般装置)

2.1.16 数值、常数 (K · 16#)

PLC 内部依据各种不同控制目的，共使用 4 种数值类型执行运算的工作，各种数值的任务及功能如下说明。

1. 二进制 (BIN)
PLC 内部的数值运算或储存均采用二进制
2. 十进制 (DEC)

十进制在 PLC 应用的时机如下：

- 作为定时器 T、计数器 C/HC 等的设定值，例：TMR C0 50。（K 常数）。
- S、M、T、C、E...等装置的编号，例：M10、T30。（装置编号）
- X、Y、D...等装置小数点前后的编号，例：X0.0、Y0.11、D10.0。（装置编号）
- 十进制常数（K）：在应用指令中作为操作数使用，例：MOV 123 D0。（K 常数）

3. BCD (Binary Code Decimal · BCD)

以一个位数或 4 个位来表示一个十进制的数值，故连续的 16 个位可以表示 4 位数的十进制数值数据。

4. 十六进制 (Hexadecimal Number · HEX)

十六进制在 PLC 应用的时机如下：

- 十六进制常数（16#）：在应用指令中作为操作数使用，例：MOV 16#1A2B D0。（十六进制常数）

其余相关说明请参考 **AH Motion Controller 操作手册**（第五章 一般装置）

2.1.17 浮点数 (F · DF)

浮点数的表示方式在 ISPSOft 中是以小数点的方式来表示，例如要输入 500 的浮点数，必须输入 500.0。

其余相关说明请参考 **AH Motion Controller 操作手册**（第五章 一般装置）

2.1.18 字符串 ("\$")

字符串可以处理的是 ASCII 编码的字 (*1)，一个完整的字符串定义为字符串的起始字符到终止符 NULL 码 (16#00) 为止。若用户直接输入的是字符串，则最多可以输入 31 个字且 ISPSOft 会自动补上终止符 16#00。若用户输入的是寄存器，则需在结束的后面补上一个终止符 16#00。

其余相关说明请参考 **AH Motion Controller 操作手册**（第五章 一般装置）

2.1.19 指标寄存器 (PR)

- ISPSOft 有支援功能块的功能，当功能块的变数类别宣告为 VAR_IN_OUT，数据类型为 POINTER (指标) 时，此变数将配置 PR 装置，PR 可以传入的装置为 (X、Y、D、L) 以及 Address 为 ISPSOft 自动配置的变数。
- 每个功能块中可以宣告的 PR 个数为 16 个：PR0~ PR15，或是使用位元 PR0.0 ~ PR15.15。

2.1.19.1 计时器指标寄存器 (T_Pointer) (TR)

- ISPSOft 有支援功能块的功能，如果要在功能块中使用计时器时，必需在功能块中宣告一个 TR 装置，并在呼叫功能块时传入 T 装置的指标。
- 当功能块的变数类别宣告为 VAR_IN_OUT，数据类型为 T_POINTER 时，此变数将配置 TR 装置，TR 可以传入的装置为 (T) 以及 ISPSOft 配置为计时器的变数。
- 每个功能块中可以宣告 TR 的个数为 8 个：TR0~TR7。
- 如果要在功能块中使用指令时，运算元区域中有支援装置 T，则必须使用 TR 指标暂存器来运算。

2.1.19.2 16 位元计数器指标寄存器 (C_Pointer) (CR)

- ISPSOft 有支援功能块的功能，如果要在功能块中使用 16 位元计数器时，必需在功能块中宣告一个 CR 装置，并在呼叫功能块时传入 C 装置的指标。

- 当功能块的变数类别宣告为 VAR_IN_OUT，资料形态为 C_POINTER 时，此变数将配置 CR 装置，CR 可以传入的装置为 (C) 以及 ISPSOft 配置为计数器的变数。
- 每个功能块中可以宣告 CR 的个数为 8 个：CR0~CR7。
- 如果要在功能块中使用指令时，运算元区域中有支援装置 C，则必须使用 CR 指标寄存器来运算。

2.1.19.3 32 位元计数器指标寄存器 (HC_Pointer) (HCR)

- ISPSOft 有支援功能块的功能，如果要在功能块中使用 32 位元计数器时，必需在功能块中宣告一个 HCR 装置，并在呼叫功能块时传入 HC 装置的指标。
- 当功能块的变数类别宣告为 VAR_IN_OUT，资料形态为 HC_POINTER 时，此变数将配置 HCR 装置，HCR 可以传入的装置为 (HC) 以及 ISPSOft 配置为计数器的变数。
- 每个功能块中可以宣告 HCR 的个数为 8 个：HCR0~HCR7。
- 如果要在功能块中使用指令时，运算元区域中有支援装置 HC，则必须使用 HCR 指标寄存器来运算。

其余相关说明请参考 **AH Motion 操作手册** (第五章 一般装置)

2.2 EtherCAT 符号

EtherCAT 符号主要用于读取 EtherCAT Slave 状态。在多数的应用中，您可以在 ISPSOft 软件内 EtherCAT 符号表中看到此系列符号。

符号名称	说明			
EtherCAT 符号名称	数据类型	功能说明	读取范围	出厂值 (默认)
eCAT_Dis.SlaveAddress[...]	WORD[199]	EtherCAT 断线的从站地址 (n=0~198)	0~9999	0
eCAT_Dis.Count	WORD	EtherCAT 断线的从站数量	0~(2 ¹⁶ -1)	0

2.3 变量符号

在传统的 PLC 程序开发过程中，往往需要花费很多的时间在管理装置地址，而在面对大型的项目时，不论在管理或除错上都会是很大的负担。有鉴于此，在 IEC 61131-3 中便导入了一般高级语言中的变量观念，除了可自行定义变量符号名称来取代 PLC 的装置编号之外，更可利用自动配置地址的功能来节省配置装置的时间，同时也提高了程序的可读性及开发效率。

*注；变量在 ISPSOft 中被称之为符号 (Symbol)，因此在手册之中，变量与符号代表的将会是相同的意义。

2.3.1 变量符号的作用范围

变量符号在使用前必须经过宣告的程序 (如下图)，且依据宣告位置的不同又可分为全局符号与区域符号；全局符号可让项目中的所有 POU 共享，而区域符号则只能在宣告该符号的 POU 当中使用。此外，在不同 POU 当中的区域符号名称是可以重复的，但若宣告的区域符号与某个全局符号的名称重复时，则在宣告该区域符号的 POU 当中，系统会自动将此名称的变量符号视为区域符号。

*注；关于 POU 说明请参考 ISPSOft 使用手册。

2.3.2 变量符号的类别

变量符号的类别依功能可分为下列 4 种，而各种类别所具有的特性则请参考下列说明。

- **VAR - 一般变量符号**

当变量符号被宣告成此类别时，代表该变量符号仅供一般的运算之用，而其所代表的意义则必须视宣告时，该变量符号被指定的数据类型或是对应的装置名称。

- **INPUT - 功能块输入端点的变量符号**

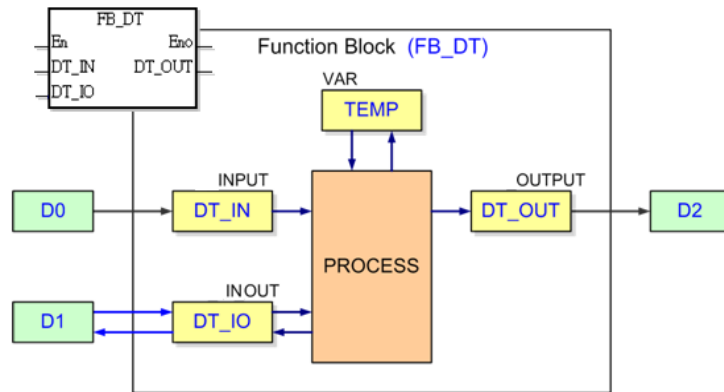
此类别的变量符号主要用做功能块的输入端点，且只有在功能块内部才可宣告；当功能块接受调用时，便可利用此类别的变量符号来接收调用方的输入值。此外，在 LD/FBD 等编程环境中，此类别的变量符号会被排列在功能块图标的左侧，并会配置一端点用以接受调用方的输入值。

- **OUTPUT - 功能块输出端点的变量符号**

此类别的变量符号主要用做功能块的输出端点，且只有在功能块内部才可宣告；当功能块执行完毕之后，其运算的结果便可经由此类别的变量符号来回传给调用方。此外，在 LD/FBD 等编程环境中，此类别的变量符号会被排列在功能块图标的右侧，并会配置一端点用以输出结果。

- **INOUT - 功能块反馈型端点的变量符号**

此类的变量符号主要用做功能块的反馈型端点，且只有在功能块当中才可宣告，其功能请参考下列。当功能块被调用时，调用方会先将 D1 的状态值输入 VAR_IN_OUT 类别的变量符号 DT_IO，而运算结束后，DT_IO 便会再将自己最后的状态值反馈输出至 D1。此外，在 LD/FBD 等编程环境中，此类别的变量符号会被排列在功能块图标的左侧，且该端点会以蓝色的粗线来与调用方进行连结。



2.3.3 变量符号的数据类型

变量符号的数据类型会决定其内容值所代表的意义，例如有两个符号 VAR_1 与 VAR_2，其中 VAR_1 的数据类型为 BOOL，而 VAR_2 则是 WORD 类型；当两者在程序当中被使用时，VAR_1 便可用来代表一个接点装置，而 VAR_2 则可用来存放 16 位大小的数值，且可进行四则运算或数据搬移。

下表为 ISPSOft 所支持的数据类型：

数据类型	名称	说明	程序	功能块
BOOL	布尔格式	其内容值可代表一个接点装置的状态。TRUE 或 FALSE.	✓	✓
WORD	16 位长度字组	可存放 16 个位大小的数据	✓	✓
DWORD	32 位长度字组	可存放 32 个位大小的数据	✓	✓
LWORD	64 位长度字组	可存放 64 个位大小的数据	✓	✓
UINT	无符号整数	16 个位大小的数据	✓	✓
UDINT	无符号双整数	32 个位大小的数据	✓	✓
INT	整数	16 个位大小的数据	✓	✓
DINT	双整数	32 个位大小的数据	✓	✓
LINT	长整数	64 个位大小的数据	✓	✓
REAL	实数	32 个位大小的数据。采单精度浮点格式数据	✓	✓
LREAL	长实数	64 个位大小的数据。采双精度浮点格式数据	✓	✓
CNT	计数	16 位或 32 位数值计数器格式，用来代表计数器装置	✓	✓
TMR	计时	16 位数值定时器格式，用来代表定时器装置。	✓	✓
ARRAY	矩阵	数组格式，而数组长度与存放的数据类型须于宣告时一并指定。（最大的数组长度为 256）	✓	✓
String	字符串	字符串格式，以 8 个位代表 1 个 ASCII 字符，且进行宣告时须一并指定字符串长度	✓	✓

2.3.4 变量符号的地址配置与初始值

每个符号都会依据其数据类型而配置一个对应的装置地址，并且可为其设定一个**初始值**，当下载项目程序且执行运动主机时的第一个扫描周期便将**初始值**一并写入该变量符号所对应的装置地址中。

以下为符号的地址配置原则。

- 全局符号与程序 POU 的区域符号可自行指定对应的装置地址或交由系统自动配置。
- 功能块的区域符号除了 VAR 形态外，仅可交由系统自动配置装置地址，无法自行指定。
- 系统自动配置的地址皆会占用可用的装置地址（用户可设定自动配置的装置范围）。
- 符号的地址、数据类型、初始值必须要相互搭配。

下表则为各种数据类型可指定或配置的装置种类。

数据类型	AH Motion Controller CPU	
	自行指定	自动配置
BOOL	M/SM 装置 或 X/Y 装置 (*3) 位类型	M/SM 装置
WORD	D 装置	W 装置
DWORD	D 装置	W 装置
LWORD	D 装置	W 装置
UINT	D 装置	W 装置
UDINT	D 装置	W 装置
INT	D 装置	W 装置
DINT	D 装置	W 装置
LINT	D 装置	W 装置
REAL	D 装置	W 装置
LREAL	D 装置	W 装置
CNT	C 装置	C 装置
TMR	T 装置	T 装置
ARRAY	ARRAY 类型的变量符号会根据指定的数组数据类型决定所配置的装置类型，且会由自行指定或自动配置的起始地址开始，连续配置符合数组长度的装置数量。 ARRAY 所指定的装置不可为 SR / SM / E / F。	
String	N/A	

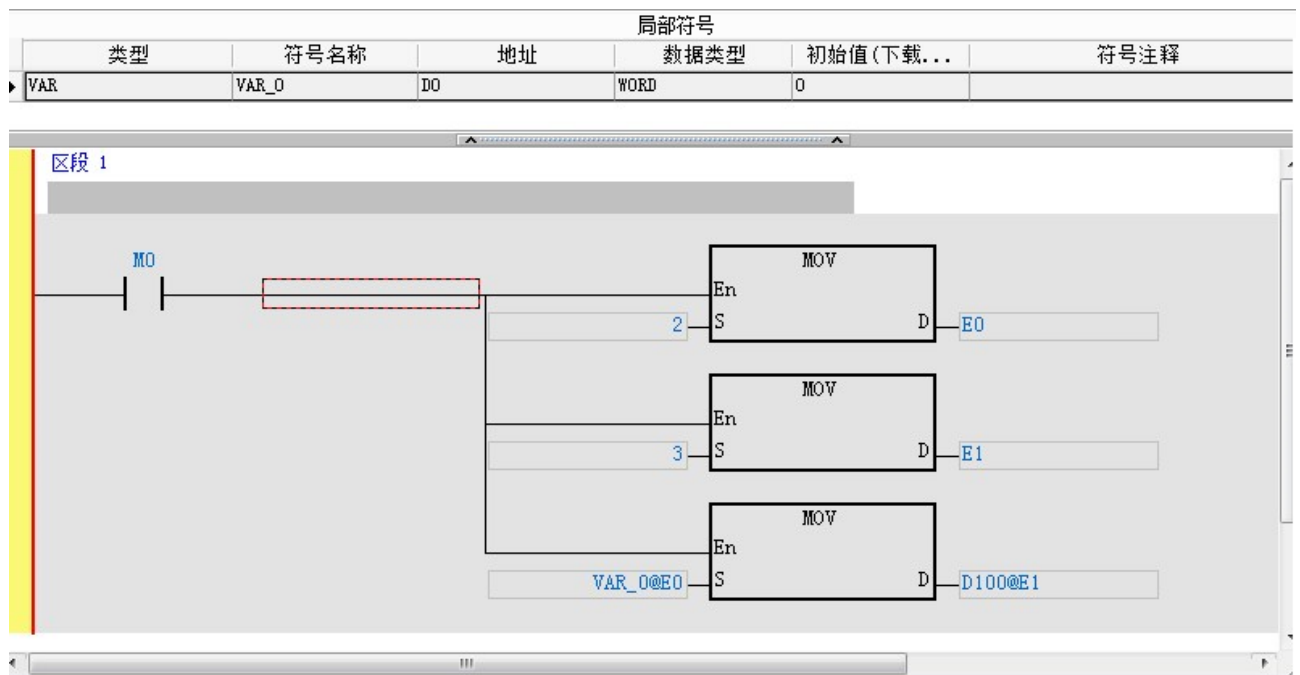
*1. 关于自动配置范围的设定方式与限制，请参考 **ISPSOft 使用手册**。

*2. 关于功能块类型的变量符号，因其具有特殊意义，因此将于 **ISPSOft 使用手册** 另行介绍..

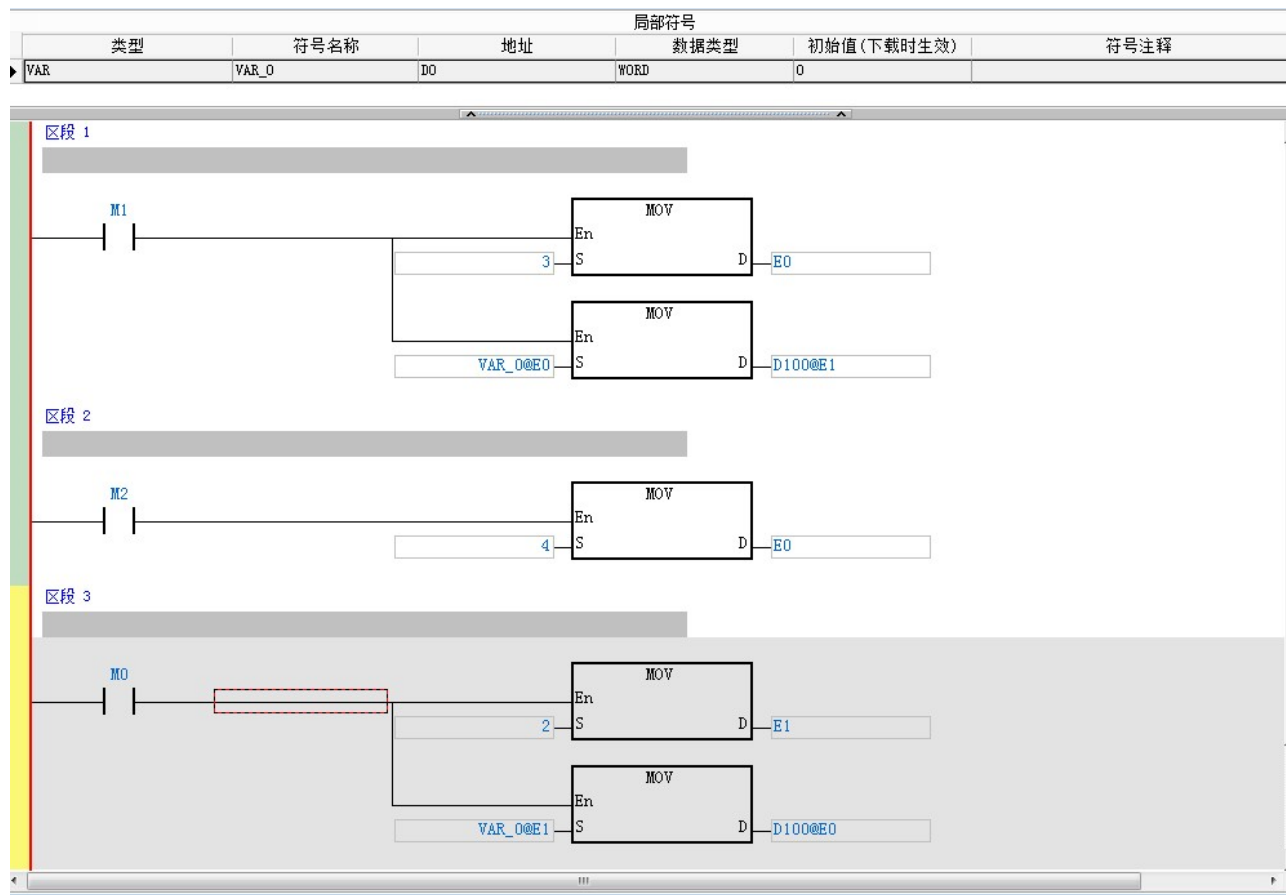
*3. 位操作指的是以 X0.0、Y0.1...等格式来表达字组中的某个位，详细说明请参考 **ISPSOft 使用手册** 的相关内容

2.3.5 变量符号的变址操作

在 ISPSOft 中允许使用变址寄存器，变址寄存器共两种类型，E 装置变址寄存器与一般的数据寄存器同为 16 位的数据寄存器，它可以自由的被写入及读出，若当一般寄存器用，仅能使用在 16 位的指令。符号与变址用法格式为 变量符号名称@装置编号，如果使用变址寄存器 E 来修饰操作数时，16 位指令及 32 位指令，皆可混用。请参考下图的程序，变量符号 VAR_0 指定的装置地址为 D0，变址寄存器内部储存的数据，其意义代表被修饰对象的偏移值；如此处的范例，当变址寄存器 E0 的值为 2 时，VAR_0@ E0 代表的便是 VAR_0 所指定的装置 (D0) 再偏移 2 个地址，即装置 D2。故程序执行的结果为当 M0 为 On 时，E0 为 2，E1 为 3，且 D2 的值被搬入 D103 中。



此外，当变址寄存器的值被变更时，实际操作的装置便会随即变更，因此若原本操作的装置没有在程序的其他位置被使用，该装置的值便会保留在最后的状况；如下图，当 E0 的值为 3 时，D2 的值会传到 D103，但当 E0 的值由 3 变成 4 时，实际操作的装置便会变更成 D104，此时 D2 的值会传到 D104，并且 D103 会继续保持之前的数值。



- *1. 变址寄存器的使用是针对某个装置进行寻址的偏移，但若所修饰的变量符号是采用自动配置的装置地址时，因用户无法得知该变量符号会被随机配置到哪个地址，因此贸然使用变址寄存器将可能造成程序执行的错乱。
- *2. 变址寄存器也可以宣告符号使用，但须指定装置地址与正确的数据类型

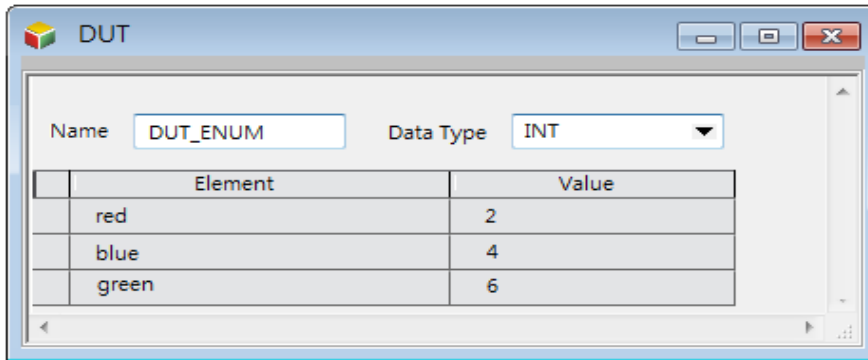
2.4 自定义数据类型 (DUT) : ENUM

用户可利用台达所支持的数据类型来建立使用自定义的ENUM格式。所谓的ENUM格式，就是变量以ENUM的元素所定义的文字来表示的衍生数据类型。在完成ENUM类型的建立后，在符号表中便可宣告此类型的变量，并可用元素名称或是数值定义初始值，若不定义则默认为元素的第一个值，可指定其中的元素，输入到此类型的变量。

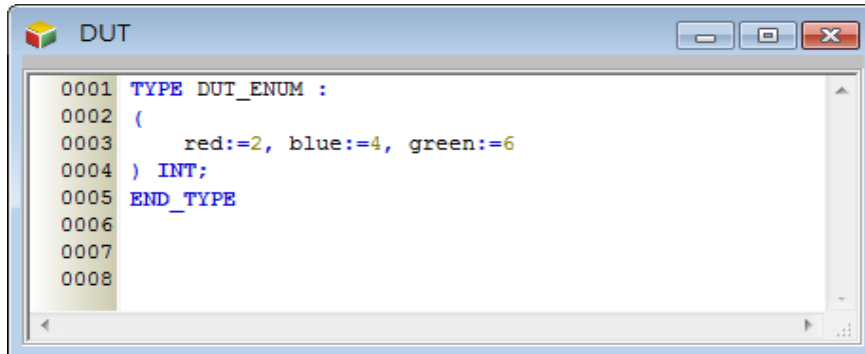
所谓的ENUM类型，可帮助研发人员更容易理解变量的意义。

ENUM的内容元素规则如下：

- 全部元素内容保持空白时，依序为数值 0、1、2...
- 部分元素内容有定义时，定义之前的元素，其数值为由 0 开始依序产生，定义之后的元素则自动加 1，例如 0、1、2、35 (用户定义)、36、37...
- 另一部分定义的例子如，0、1、2、35 (用户定义)、36、70 (用户定义)、71、...
- 设定方式提供以下两种：



使用设定表格
宣告



使用文字编辑
方式宣告

详细ENUM列表可参考本手册附录，其余相关软件设定方式与程序中操作说明，请参考*ISPSOft软件使用手册*。

2.5 指令

在AH Motion Controller 系列产品中，指令包含标准指令和运动控制指令。

关于运动控制指令的说明请参考**AH Motion Controller –运动控制指令手册**。

指令快速查询请参考附录**A.4 标准指令（依字母排序）**

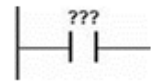
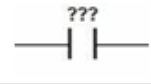
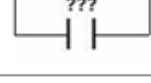
2.5.1 标准指令分类

分类	说明
Ladder 指令	基本常用的 Ladder 编译指令如 LD、LDP、AND、OR、OUT...等
比较操作指令	比较操作如=、<>、>、>=、<、<=...等
四则运算指令	BIN 或 BCD 的加法、减法、乘法或除法
数据转换指令	将 BCD 转换成 BIN 和将 BIN 转换成 BCD
数据转移指令	传送指定的数据
程序跳转指令	程序跳转
程序执行控制指令	允许或禁止中断程序
I/O 更新指令	运行局部更新
便利指令	用于以下目的的指令：计数器增加/减小，教学定时器，特殊功能定时器，旋转台最短距离控制...等
逻辑操作指令	逻辑操作，如逻辑加法、逻辑乘法等
旋转指令	指定数据的旋转移位
基本指令	定时器，计数器 指令
移位指令	指定数据的移位
数据处理指令	16 位数据查询，数据处理如译码和编码
结构建立指令	巢状回路、调用功能块...等
模块的数据读/写指令	特殊功能模块的数据读/写
浮点数指令	浮点数运算指令
万年历指令	万年历时间（年、月、日、小时、分、秒和星期）的读取、更新与比较
外围设备指令	连接到外围设备的 I/O
通讯指令	通讯控制接口设备

分类	说明
其他指令	其他不符合以上范畴的指令，如通讯逾时重置指令和定时时钟指令
字符串处理指令	CBIN/BCD 和 ASCII 之间的转换；BIN 和字符串之间的转换；浮点十进制数和字符串之间的转换、字符串处理等等
以太网控制指令	控制以太网数据交换相关指令
储存卡读写指令	储存卡读写相关指令
任务控制指令	程序中任务控制相关指令
SFC 控制指令	SFC 控制相关指令

2.5.2 标准指令列表

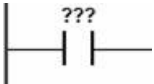
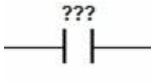
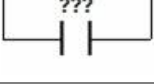
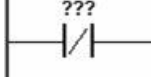
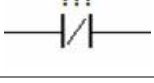
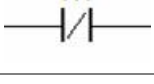
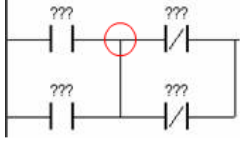
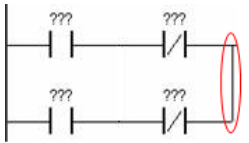
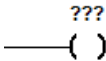

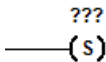
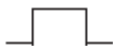
● 表格字段说明



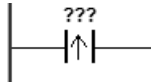

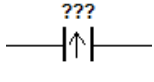
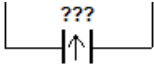
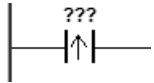

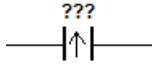
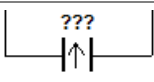
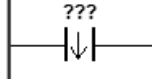

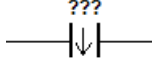
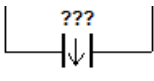
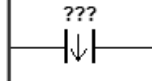

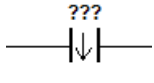
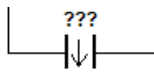
① FB/FC	② 指令码	③ 图形	④ 功能	⑤ 操作数
FC	LD		载入 A 接点	DX、X、Y、M、 SM/AM/AR、S、T、C、 HC/AC、D、L、PR
FC	AND		串联 A 接点	
FC	OR		并联 A 接点	

字段说明	
1	FB/FC FB：功能块；FC：函数 关于程序（PROG）、功能块（FB）、函数（FC）的说明请参考 ISPSOft 软件使用手册。
2	指令码 指令名称
3	图形 在 ISPSOft 梯形图所显示的图形。
4	功能 指令的功能用途说明。
5	操作数 指令所支持的装置类别

● 梯形图指令

指令详细说明参考第 3.2 节梯形图指令

FB/FC	指令码	图形	功能		操作数
FC	LD		载入 A 接点		DX、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	AND		串联 A 接点		
FC	OR		并联 A 接点		
FC	LDI		载入 B 接点		DX、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	ANI		串联回路方块		
FC	ORI		并联回路方块		
FC	ANB		存入堆栈指令		—
FC	ORB		堆栈读取指针不动		—
FC	MPS	—	读出堆栈		—
FC	MRD	—	串联回路方块		—
FC	MPP	—	并联回路方块		—
FC	OUT		驱动线圈	执行条件： 	DY、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	SET		动作保持 (ON)	执行条件： 	DY、X、Y、M、SM、S、T、C、HC/AC、D、L、PR

FB/FC	指令码	图形	功能		操作数
FC	MC		共通串联接点的连结		N
FC	MCR		共通串联接点的解除		N
FC	LDP		上升沿检出动作开始连接	执行条件： 	DX、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	ANDP		上升沿检出动作串联连接		
FC	ORP		上升沿检出动作并联连接		
FC	PED		上升沿检出动作开始连接	执行条件： 	X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	APED		上升沿检出动作串联连接		
FC	OPED		上升沿检出动作并联连接		
FC	LDF		下降沿检出动作开始连接	执行条件： 	DX、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	ANDF		下降沿检出动作串联连接		
FC	ORF		下降沿检出动作并联连接		
FC	NED		下降沿检出动作开始连接	执行条件： 	X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	ANED		下降沿检出动作串联连接		
FC	ONED		下降沿检出动作并联连接		

FB/FC	指令码	图形	功能		操作数
FC	PLS		上微分输出	执行条件： 	DY、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	PLF		下微分输出	执行条件： 	DY、X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	INV		运算结果反相		-
FC	NOP	-	无动作		-
FC	NP		上升沿触发指令		-
FC	PN		下降沿触发指令		-
FC	FB_NP		上升沿触发指令		X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	FB_PN		下降沿触发指令		X、Y、M、SM、S、T、C、HC/AC、D、L、PR
FC	PSTOP		PLC 程序停止执行		-

● 表格字段说明

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	LD=	DLD=	—		接点类型比较 S1 = S2 导通 S1 ≠ S2 不导通
FC	LD<>	DLD<>	—		接点类型比较 S1 ≠ S2 导通 S1 = S2 不导通

字段说明		
1	FB/FC	FB：功能块；FC：函数
2	指令码及类型	指令名称（16 位/32 位/64 位） 32 位：16 位的指令，如有支持 32 位，则在指令前加 D，即为 32 位指令。 64 位：32 位的浮点数指令，如有支持 64 位，则在指令前加 D，即为 64 位浮点数指令。
3	脉冲指令	是否有脉冲（P）指令，有：✓。无：—。 有脉冲时，要使用脉冲指令，只需在指令后加 P。
4	图形	在 ISPSOft 梯形图所显示的图形。
5	功能	指令的功能用途说明。

● 比较操作指令

详细说明请参考第 3.3 节比较操作指令

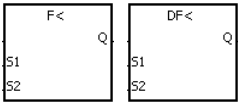
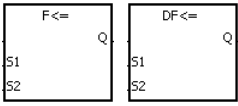
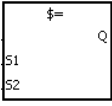
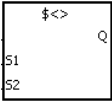
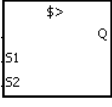
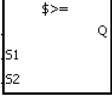
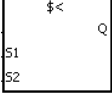
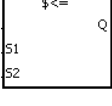
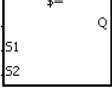
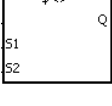
FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	LD=	DLD=	—		接点类型比较 S1 = S2 导通 S1 ≠ S2 不导通
FC	LD<>	DLD<>	—		接点类型比较 S1 ≠ S2 导通 S1 = S2 不导通
FC	LD>	DLD>	—		接点类型比较 S1 > S2 导通 S1 ≤ S2 不导通
FC	LD>=	DLD>=	—		接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通
FC	LD<	DLD<	—		接点类型比较 S1 < S2 导通 S1 ≥ S2 不导通

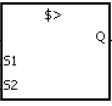
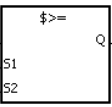
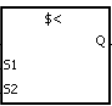
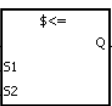
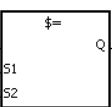
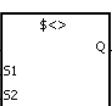
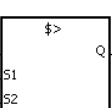
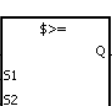
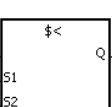
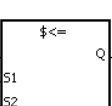
FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	LD<=	DLD<=	—		接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
FC	AND=	DAND=	—		串连接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	AND<>	DAND<>	—		串连接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
FC	AND>	DAND>	—		串连接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
FC	AND>=	DAND>=	—		串连接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
FC	AND<	DAND<	—		串连接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	AND<=	DAND<=	—		串连接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
FC	OR=	DOR=	—		并联接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	OR<>	DOR<>	—		并联接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
FC	OR>	DOR>	—		并联接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	OR>=	DOR>=	-		并联接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	OR<	DOR<	-		并联接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	OR<=	DOR<=	-		并联接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	FLD=	DFLD=	-		浮点数接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	FLD<>	DFLD<>	-		浮点数接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
FC	FLD>	DFLD>	-		浮点数接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
FC	FLD>=	DFLD>=	-		浮点数接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
FC	FLD<	DFLD<	-		浮点数接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	FLD<=	DFLD<=	-		浮点数接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	FAND=	DFAND=	—		浮点数串联接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	FAND<>	DFAND<>	—		浮点数串联接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
FC	FAND>	DFAND>	—		浮点数串联接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
FC	FAND>=	DFAND>=	—		浮点数串联接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
FC	FAND<	DFAND<	—		浮点数串联接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	FAND<=	DFAND<=	—		浮点数串联接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
FC	FOR=	DFOR=	—		浮点数并联接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	FOR<>	DFOR<>	—		浮点数并联接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
FC	FOR>	DFOR>	—		浮点数并联接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
FC	FOR>=	DFOR>=	—		浮点数并联接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	FOR<	DFOR<	—		浮点数并联接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	FOR<=	DFOR<=	—		浮点数并联接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
FC	LD\$=	—	—		字符串串联接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	LD\$<>	—	—		字符串串联接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
FC	LD\$>	—	—		字符串串联接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
FC	LD\$>=	—	—		字符串串联接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
FC	LD\$<	—	—		字符串串联接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
FC	LD\$<=	—	—		字符串串联接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
FC	AND\$=	—	—		字符串串联接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
FC	AND\$<>	—	—		字符串串联接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	AND\$>	-	-		字符串串联接点类型比较 S1 > S2 导通 S1 ≤ S2 不导通
FC	AND\$>=	-	-		字符串串联接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通
FC	AND\$<	-	-		字符串串联接点类型比较 S1 < S2 导通 S1 ≥ S2 不导通
FC	AND\$<=	-	-		字符串串联接点类型比较 S1 ≤ S2 导通 S1 > S2 不导通
FC	OR\$=	-	-		字符串并联接点类型比较 S1 = S2 导通 S1 ≠ S2 不导通
FC	OR\$<>	-	-		字符串并联接点类型比较 S1 ≠ S2 导通 S1 = S2 不导通
FC	OR\$>	-	-		字符串并联接点类型比较 S1 > S2 导通 S1 ≤ S2 不导通
FC	OR\$>=	-	-		字符串并联接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通
FC	OR\$<	-	-		字符串并联接点类型比较 S1 < S2 导通 S1 ≥ S2 不导通
FC	OR\$<=	-	-		字符串并联接点类型比较 S1 ≤ S2 导通 S1 > S2 不导通

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	CMP	DCMP	✓		比较设定输出
FC	ZCP	DZCP	✓		区域比较
FC	-	FCMP	✓		浮点数比较
FC	-	FZCP	✓		浮点数区域比较
FC	MCMP	-	✓		矩阵比较
FC	CMPT=	-	✓		表格比较, =
FC	CMPT<>	-	✓		表格比较, ≠
FC	CMPT>	-	✓		表格比较, >
FC	CMPT>=	-	✓		表格比较, ≥
FC	CMPT<	-	✓		表格比较, <

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	CMPT<=	-	✓		表格比较, \leq
FC	CHKADR	-	-		接点类型指针寄存器地址检查

● 四则运算指令

详细说明请参考第 3.4 节四则运算指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	+	D+	✓		BIN 加法 $S1+S2=D$
FC	-	D-	✓		BIN 减法 $S1-S2=D$
FC	*	D*	✓		BIN 乘法 $S1*S2=D$
FC	/	D/	✓		BIN 除法 $S1/S2=D$

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	F+	DF+	✓		浮点数加法 $S1+S2=D$
FC	F-	DF-	✓		浮点数减法 $S1-S2=D$
FC	F*	DF*	✓		浮点数乘法 $S1*S2=D$
FC	F/	DF/	✓		浮点数除法 $S1/S2=D$

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	B+	DB+	✓		BCD 加法 $S1+S2=D$
FC	B-	DB-	✓		BCD 减法 $S1-S2=D$

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	B*	DB*	✓		BCD 乘法 $S1 * S2 = D$
FC	B/	DB/	✓		BCD 除法 $S1 / S2 = D$
FC	BK+	—	✓		连续区块 BIN 加法
FC	BK-	—	✓		连续区块 BIN 减法
FC	\$+	—	✓		字符串链结
FC	INC	DINC	✓		BIN 加一
FC	DEC	DDEC	✓		BIN 减一
FC	MUL16	MUL32	✓		16 位专用 BIN 乘法/ 32 位专用 BIN 乘法
FC	DIV16	DIV32	✓		16 位专用 BIN 除法/ 32 位专用 BIN 除法

● 数据转换指令

详细说明请参考第 3.5 节数据转换指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	BCD	DBCD	✓		BIN→BCD 变换
FC	BIN	DBIN	✓		BCD→BIN 变换
FC	FLT	DFLT	✓		BIN 整数→二进浮点数变换
FC	FLTD	DFLTD	✓		BIN 整数→64 位浮点数变换
FC	INT	DINT	✓		二进浮点数→BIN 整数变换

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	FINT	DFINT	✓		64 位浮点数→BIN 整数变换
FC	FTODF	—	—		单精度浮点数→双精度浮点数转换
FC	—	DFTOF	—		双精度浮点数→单精度浮点数转换

FB/FC	指令码		P 指令	图形		功能
	16 位	32 位				
FC	MMOV	—	✓			16→32 位数值转换
FC	RMOV	—	✓			32→16 位数值转换
FC	GRY	DGRY	✓	 	 	BIN→GRY 码变换
FC	GBIN	DGBIN	✓	 	 	GRY 码→BIN 变换
FC	NEG	DNEG	✓	 	 	取负数 (取 2 的补码)
FC	—	FNEG	✓			32 位浮点正负符号反相
FC	—	FBCD	✓			二进浮点数→十进浮点数
FC	—	FBIN	✓			十进浮点数→二进浮点数
FC	BKBCD	—	✓			连续区块 BIN→BCD 变换
FC	BKBIN	—	✓			连续区块 BCD→BIN 变换
FC	SCAL	—	✓			比例运算
FC	SCLP	—	✓	 	 	参数型比例运算

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	LINE	DLINE	✓		COLUMN to LINE
FC	COLM	DCOLM	✓		LINE to COLUMN

● 数据转移指令

详细说明请参考第 3.6 节数据转移指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	MOV	DMOV	✓		数据移动 S：数据源 D：数据目的地

FB/FC	指令码		P 指令	图形	功能
	32 位	64 位			
FC	—	DFMOV	✓		64 位浮点数数据移动 S：数据源 D：数据目的地

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	\$MOV	—	✓		字符串移动
FC	CML	DCML	✓		反转传送

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	BMOV	—	✓		全部传送
FC	NMOV	DNMOV	✓		多点移动
FC	XCH	DXCH	✓		数据的交换
FC	BXCH	—	✓		全部交换
FC	SWAP	DSWAP	✓		上/下 BYTE 变换
FC	SMOV	—	✓		位数移动
FC	MOVB	—	✓		多位移动

● 程序跳转指令

详细说明请参考第 3.7 节程序跳转指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	CJ	—	✓		条件跳转
FC	JMP	—	—		无条件跳转

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	GOEND	—	—		跳转到 END

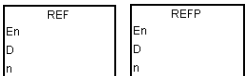
- 程序执行控制指令

详细说明请参考第 3.8 节程序执行控制指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	DI	—	—		中断插入禁止
FC	EI	—	—		中断插入允许
FC	IMASK	—	—		中断控制

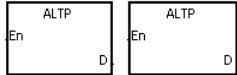

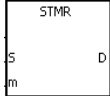
- I/O 更新指令

详细说明请参考第 3.9 节 I/O 更新指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	REF	—	✓		I/O 更新处理

- 便利指令

详细说明请参考第 3.10 节便利指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	ALT	—	✓		ON/OFF 交替
FC	TTMR	—	—		教导式定时器
FC	STMR	—	—		特殊定时器

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	RAMP	—	—		倾斜信号
FC	MTR	—	—		矩阵输入
FC	ABSD	DABSD	—		绝对方式凸轮控制
FC	INCD	—	—		相对方式凸轮控制
FC	—	DPID	—		PID 运算
FC	—	DPIDE	—		PID 运算

● 逻辑操作指令

详细说明请参考第 3.11 节逻辑操作指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	WAND	DAND	✓		逻辑与 (AND) 运算
FC	MAND	—	✓		矩阵与 (AND) 运算
FC	WOR	DOR	✓		逻辑或 (OR) 运算
FC	MOR	—	✓		矩阵或 (OR) 运算
FC	WXOR	DXOR	✓		逻辑异或 (XOR) 运算
FC	MXOR	—	✓		矩阵异或 (XOR) 运算
FC	LD&	DLD&	—		ON : $S_1 \& S_2 \neq 0$ OFF : $S_1 \& S_2 = 0$
FC	LD	DLD	—		ON : $S_1 S_2 \neq 0$ OFF : $S_1 S_2 = 0$
FC	LD^	DLD^	—		ON : $S_1 \wedge S_2 \neq 0$ OFF : $S_1 \wedge S_2 = 0$

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	AND&	DAND&	—		ON : $S_1 \& S_2 \neq 0$ OFF : $S_1 \& S_2 = 0$
FC	AND	DAND	—		ON : $S_1 S_2 \neq 0$ OFF : $S_1 S_2 = 0$
FC	AND^	DAND^	—		ON : $S_1 \wedge S_2 \neq 0$ OFF : $S_1 \wedge S_2 = 0$
FC	OR&	DOR&	—		ON : $S_1 \& S_2 \neq 0$ OFF : $S_1 \& S_2 = 0$
FC	OR	DOR	—		ON : $S_1 S_2 \neq 0$ OFF : $S_1 S_2 = 0$
FC	OR^	DOR^	—		ON : $S_1 \wedge S_2 \neq 0$ OFF : $S_1 \wedge S_2 = 0$

● 旋转指令

详细说明请参考第 3.12 节旋转指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	ROR	DROR	✓		右旋转
FC	RCR	DRCR	✓		附进位标志右旋转
FC	ROL	DROL	✓		欲旋转的装置

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	RCL	DRCL	✓		附进位标志左旋转
FC	MBR	—	✓		矩阵位旋转

● 基本指令

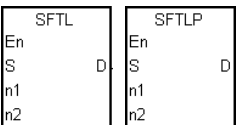


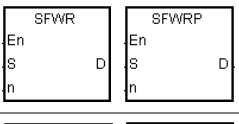



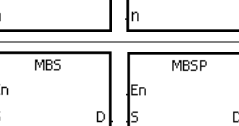

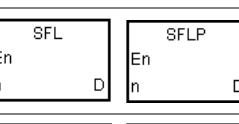


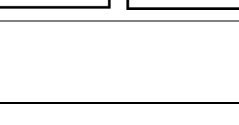

详细说明请参考第 3.13 节基本指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	RST	—	—		接点或寄存器清除
FC	TMR	—	—		16 位定时器
FC	TMRH	—	—		16 位定时器
FC	CNT	—	—		16 位计数器
FC	—	DCNT	—		32 位计数器

● 移位指令

详细说明请参考第 3.14 节移位指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	SFTR	—	✓		位右移

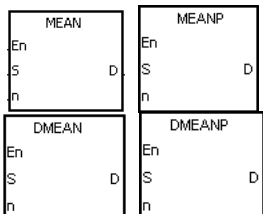
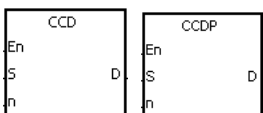
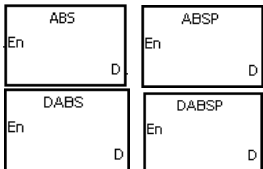
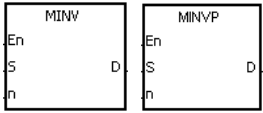

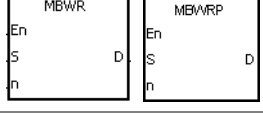
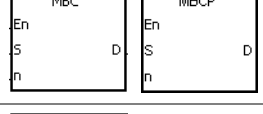
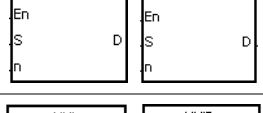

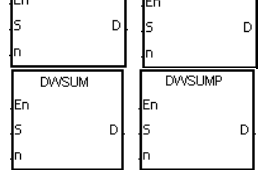
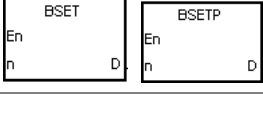
FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	SFTL	—	✓		位左移
FC	WSFR	—	✓		寄存器右移
FC	WSFL	—	✓		寄存器左移
FC	SFWR	—	✓		位移写入
FC	SFRD	—	✓		位移读出
FC	SFPO	—	✓		读出数据串行最新数据
FC	SFDEL	—	✓		删除数据串行中的数据
FC	SFINS	—	✓		插入数据到数据串行中
FC	MBS	—	✓		矩阵位位移
FC	SFR	—	✓		16 位寄存器位右移
FC	SFL	—	✓		16 位寄存器位左移
FC	BSFR	—	✓		n 个位右移 1 个位
FC	BSFL	—	✓		n 个位左移 1 个位
FC	NSFR	—	✓		n 个寄存器右移

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	NSFL	—	✓		n 个寄存器左移

● 位处理指令

详细说明请参考第 3.15 节位处理指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	SER	DSER	✓		多点比较
FC	SUM	DSUM	✓		ON 位数量
FC	DECO	—	✓		译码器
FC	ENCO	—	✓		编码器
FC	SEGD	—	✓		7 段显示器解码
FC	SORT	DSORT	-		数据排序
FC	ZRST	—	✓		区域清除
FC	BON	DBON	✓		ON 位判定

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	MEAN	DMEAN	✓		平均值
FC	CCD	—	✓		总和检查
FC	ABS	DABS	✓		绝对值
FC	MINV	—	✓		矩阵反相
FC	MBRD	—	✓		矩阵位读出
FC	MBWR	—	✓		矩阵位写入
FC	MBC	—	✓		矩阵位状态计数
FC	DIS	—	✓		16 位数据的 4 位分组
FC	UNI	—	✓		16 位数据的 4 位链结
FC	WSUM	DWSUM	✓		16 位数据的总和计算
FC	BSET	—	✓		字符件的位设定

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	BRST	—	✓		字符件的位复位
FC	BKRST	—	✓		指定区域清除
FC	LIMIT	DLIMIT	✓		BIN 16 位和 BIN 32 位数据的高低限控制
FC	BAND	DBAND	✓		BIN 16 位和 32 位死区控制
FC	ZONE	DZONE	✓		BIN 16 位和 32 位区域控制

● 结构建立指令

详细说明请参考第 3.16 节结构建立指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	FOR	—	—		循环回路起始
FC	NEXT	—	—		循环回路结束
FC	BREAK	—	✓		强制结束 FOR-NEXT 循环

● 模块的数据读/写指令

详细说明请参考第 3.17 节模块的数据读/写指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	FROM	DFROM	✓		特殊模块 CR 数据读出
FC	TO	DTO	✓		特殊模块 CR 数据写入

● 浮点数指令

详细说明请参考第 3.18 节浮点数指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	—	FSIN	✓		二进浮点数 SIN 运算
FC	—	FCOS	✓		二进浮点数 COS 运算
FC	—	FTAN	✓		二进浮点数 TAN 运算
FC	—	FASIN	✓		二进浮点数 ASIN 运算
FC	—	FACOS	✓		二进浮点数 ACOS 运算
FC	—	FATAN	✓		二进浮点数 ATAN 运算
FC	—	FSINH	✓		二进浮点数 SINH 运算

FB/FC	指令码		P 指令	图形		功能
	16 位	32 位				
FC	—	FCOSH	✓			二进浮点数 COSH 运算
FC	—	FTANH	✓			二进浮点数 TANH 运算
FC	—	FRAD	✓			角度→弧度
FC	—	FDEG	✓			弧度→角度
FC	SQR	DSQR	✓			BIN 开平方根
FC	—	FSQR	✓			浮点数开平方根
FC	—	FEXP	✓			浮点数取指数
FC	—	FLOG	✓			浮点数取对数
FC	—	FLN	✓			二进浮点数取自然对数
FC	—	FPOW	✓			浮点数权值指令
FC	RAND	—	✓			随机数值
FC	BSQR	DBSQR	✓			BCD 开平方根
FC	—	BSIN	✓			BCD SIN 运算
FC	—	BCOS	✓			BCD COS 运算
FC	—	BTAN	✓			BCD TAN 运算

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	—	BASIN	✓	 	BCD ASIN 运算
FC	—	BACOS	✓	 	BCD ACOS 运算
FC	—	BATAN	✓	 	BCD ATAN 运算

● 万年历指令

详细说明请参考第 3.19 节万年历指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	TRD	—	✓	 	万年历数据读出
FC	TWR	—	✓	 	万年历数据写入
FC	T+	—	✓	 	万年历数据加法运算
FC	T-	—	✓	 	万年历数据减法运算
FC	HOUR	DHOUR	—	 	运转定时器
FC	TCMP	—	✓	 	万年历数据比较
FC	TZCP	—	✓	 	万年历数据区域比较

● 外围设备指令

详细说明请参考第 3.20 节外围设备指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	TKY	DTKY	—		10 键键盘输入
FC	HKY	DHKY	—		16 键键盘输入
FC	DSW	—	—		指拨开关输入
FC	ARWS	—	—		箭头键盘输入
FC	SEGL	—	—		七段显示器扫描输出

● 通讯指令

详细说明请参考第 3.21 节通讯指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	RS	—	—		串行数据传输
FC	LRC	—	✓		和检查 LRC 模式
FC	CRC	—	✓		和检查 CRC 模式

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	MODRW	—	—		MODBUS 数据读写
FC	COMRS	—	—		通讯传送与接收指令

● 其他指令


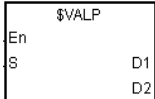
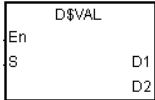



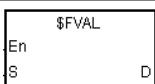
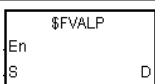
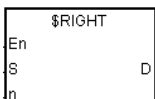

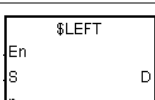
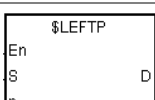



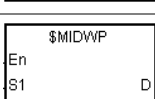
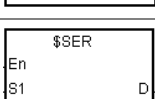
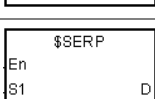
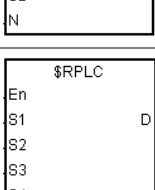
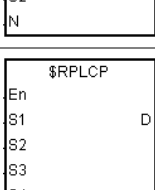
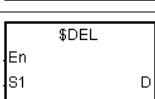
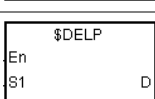

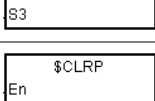
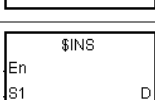
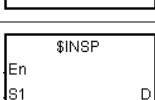
详细说明请参考第 3.22 节其他指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	WDT	—	✓		逾时监视定时器
FC	DELAY	—	✓		延迟指令
FC	GPWM	—	—		一般用脉冲波宽调变
FC	TIMCHK	—	—		时间检查
FC	EPUSH	—	✓		指针寄存器存入
FC	EPOP	—	✓		指针寄存器读出

● 字符串处理指令

详细说明请参考第 3.23 节字符串处理指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	BINDA	DBINDA	✓		有号数十进制→ASCII 变换
FC	BINHA	DBINHA	✓		BIN 十六进制→十六进制 ASCII 变换
FC	BCDDA	DBCDDA	✓		BCD→ASCII 变换
FC	DABIN	DDABIN	✓		有号数十进制 ASCII→有号数 十进制 BIN 变换
FC	HABIN	DHABIN	✓		十六进制 ASCII→十六进制 BIN 变换
FC	DABCD	DDABCD	✓		ASCII→BCD 变换
FC	\$LEN	—	✓		计算字符串长度
FC	\$STR	\$DSTR	✓		BIN→String

FB/FC	指令码		P 指令	图形		功能
	16 位	32 位				
FC	\$VAL	\$DVAL	✓			String→BIN
						
FC	\$FSTR	—	✓			Float→String
FC	\$FVAL	—	✓			String→Float
FC	\$RIGHT	—	✓			从右边获取字符串
FC	\$LEFT	—	✓			从左边获取字符串
FC	\$MIDR	—	✓			区段获取字符串
FC	\$MIDW	—	✓			区段字符串取代
FC	\$SER	—	✓			字符串搜寻
FC	\$RPLC	—	✓			字符串取代
FC	\$DEL	—	✓			指定字符串删除
FC	\$CLR	—	✓			字符串清除
FC	\$INS	—	✓			字符串插入

FB/FC	指令码		P 指令	图形		功能
	16 位	32 位				
FC	\$FMODE	—	✓			浮点数转 BCD 浮点数
FC	\$FREXP	—	✓			BCD 浮点数转浮点数

● 以太网控制指令

详细说明请参考第 3.24 节以太网控制指令

FB/FC	指令码		P 指令	图形		功能
	16 位	32 位				
FC	EMDRW	—	✓			MODBUS TCP 数据读写
FC	—	DINTOA	✓			IP→字符串变换
FC	—	DIATON	✓			字符串→IP 变换
FC	SOPEN	—	✓			开启 Socket
FC	SSEND	—	✓			通过已开启的 Socket 传送数据
FC	SRCVD	—	✓			通过已开启的 Socket 接收数据
FC	SCLOSE	—	✓			关闭 Socket

● 储存卡读写指令

详细说明请参考第 3.25 节储存卡读写指令

2

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	MWRIT	—	✓		写入储存卡数据
FC	MREAD	—	✓		读取储存卡数据
FC	MTWRIT	—	✓		写入字符串至储存卡

● 任务控制指令

详细说明请参考第 3.26 节任务控制指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	TKON	—	✓		Cyclic 任务 Task 启动
FC	TKOFF	—	✓		Cyclic 任务 Task 关闭

● SFC 控制指令

详细说明请参考第 3.27 节 SFC 控制指令

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	SFCRUN	—	—		SFC 启动

FB/FC	指令码		P 指令	图形	功能
	16 位	32 位			
FC	SFCPSE	—	—		SFC 暂停
FC	SFCSTP	—	—		SFC 停止

MEMO

2

第3章 标准指令

目录

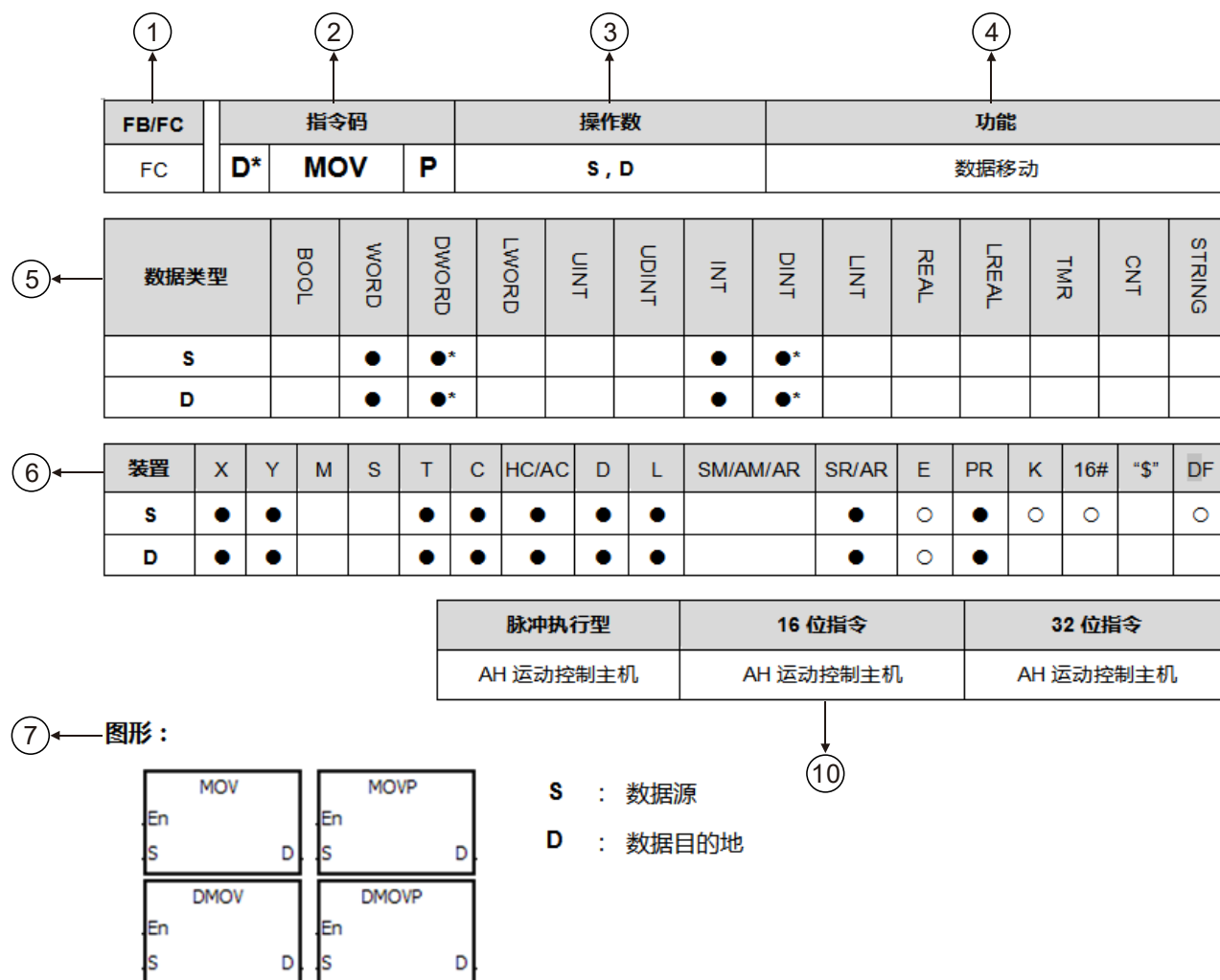
3.1	指令阅读方式	3-3
3.1.1	指令组成说明	3-3
3.1.2	指令使用限制	3-9
3.2	梯形图指令	3-11
3.3	比较操作指令	3-36
3.4	四则运算指令	3-73
3.5	数据转换指令	3-113
3.6	数据转移指令	3-159
3.7	程序跳转指令	3-185
3.8	程序执行控制指令	3-194
3.9	I/O 更新指令	3-202
3.10	便利指令	3-205
3.11	逻辑操作指令	3-241
3.12	循环指令	3-265
3.13	基本指令	3-277
3.14	位移指令	3-286
3.15	数据处理指令	3-320
3.16	结构建立指令	3-372
3.17	模块的数据读写指令	3-380
3.18	浮点数指令	3-386

3.19	万年历指令	3-432
3.20	外围设备指令	3-447
3.21	通讯指令	3-463
3.22	其它指令	3-489
3.23	字符串处理指令	3-502
3.24	以太网控制指令	3-566
3.25	储存卡读写指令	3-591
3.26	任务控制指令	3-604
3.27	SFC 控制指令	3-609

3.1 指令阅读方式

为了能够充分了解各指令的功能、说明以及限制，本节介绍各指令的表格字段，以及用来说明指令属性的各种标记方式。

3.1.1 指令组成说明



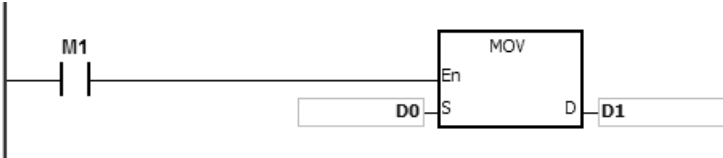
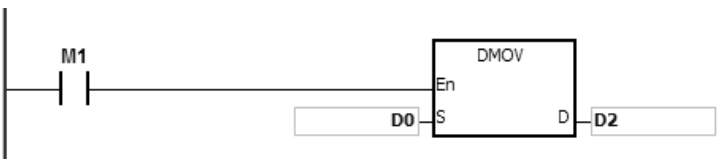
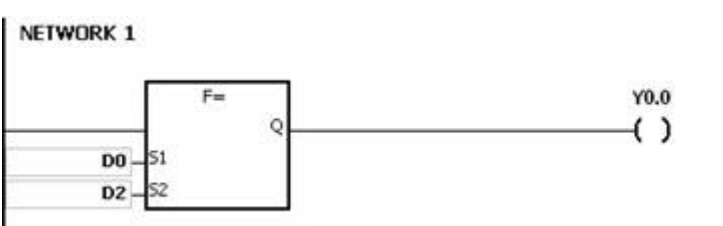
⑧ 指令说明：

1. 当该指令执行时，将 S 的内容直接搬移至 D 内。当指令不执行时，D 内容不会变化。
2. 32 位指令 S 才可使用 F 浮点数。
3. 32 位指令才可使用 HC 装置。

⑨ 程序范例：

1. 16 位数据搬移，须使用 MOV 指令。

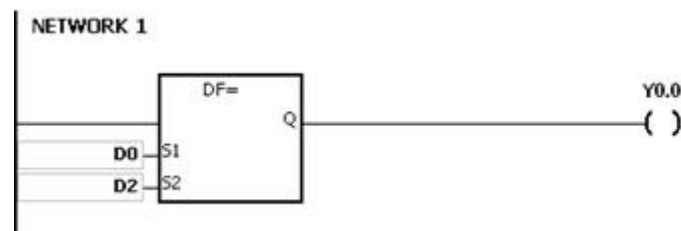
指令表字段说明

1	FB/FC	<p>FB：功能块指令；FC：函数指令</p> <p>FC型的指令可使用在程序，功能块，和函数。</p> <p>FB型的指令仅可使用在程序和功能块。</p>
2	指令码	<p>指令码：指令的名称</p> <p>16位的指令，如有支持32位，则在指令前加D*，即为32位指令。32位的浮点数指令（指令码带有前缀字F，如：FMOV），如有支持64位，则在指令前加D*，即为64位浮点数指令。有脉冲执行型的指令则会在指令名称后有P。</p> <p>● D*：区别16位、32位或64位指令</p> <p>操作数的数值内容，其长度可分为16位及32位，因此部分指令处理不同长度的数据则分为16及32位的指令，用以区分32位的指令只需要在16位指令前加上“D”来表示即可。</p> <p>注：D 右上角的星号[*]可对应至32或64位指令所对应的数据类型，细节请参照「字段5:数据类型」的说明。</p> <p>16位MOV指令</p>  <p>当 M1=ON，D0 被传送至 D1</p> <p>32位DMOV指令</p>  <p>当 M1=ON 时，(D1, D0) 的内容被传送至 (D3, D2)</p> <p>32位[F=]指令</p> 

3

指令表字段说明

64 位「DF=」指令



Note：部分指令在前缀除了 D*还增加 W 以区分 16 位指令 (WXOR) 或 32 位指令 (DXOR)

例：

FB/FC	指令码		
FC	W D*	XOR	P

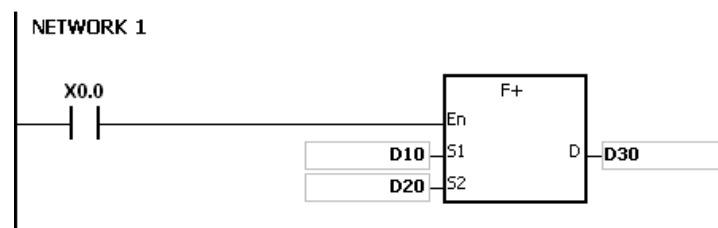
● F/DF：浮点数指令

浮点数指令分为 32 位浮点数指令跟 64 位浮点数指令，指令中 32 位浮点数指令为单精度浮点数指令，64 位浮点数指令为双精度浮点数指令。详细的浮点数说明可以参考第二章的浮点数内容说明。

浮点数指令操作数的数值内容表示方式，其长度可分为 32 位及 64 位，因此部份指令处理不同长度的数据则分为 32 及 64 位的指令，浮点数指令的开头都会以“F”来表示，而用以区分 64 位的指令只需要在 32 位浮点数指令前加上“D”来表示即可。

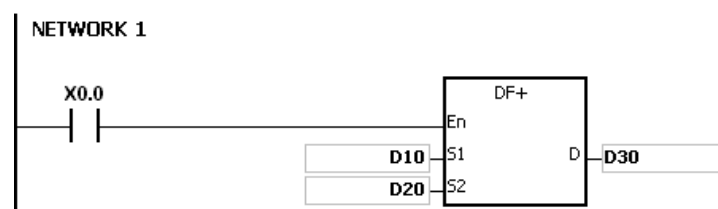
2 指令码

32 位单精度浮点数指令 F+



当 X0.0=ON 时，单精度浮点数 (D11, D10) + 单精度浮点数 (D21, D20) 的内容被传送至 (D31, D30)

64 位单精度浮点数指令 DF+



当 X0.0=ON 时，双精度浮点数 (D13, D12, D11, D10) + 双精度浮点数 (D23, D22, D21, D20) 的内容被传送至 (D33, D32, D31, D30)

指令表字段说明

● P：脉冲执行指令

连续执行型和脉冲执行型：

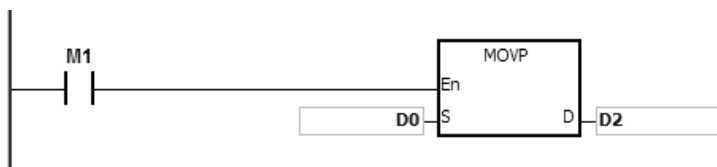
1. 以指令的执行方式来说也可分成「连续执行型」及「脉冲执行型」2种。由于指令不被周期性连续执行时，所需的程序扫描时间比较短，因此程序中尽可能的使用脉冲执行型指令可减少扫描周期。

“脉冲”功能可让指定的指令由上升沿触发，而该指令在一个扫描周期内被执行一次。

2. 后续若此触发输入的接点保持为 ON，则该指令不会执行。需重新执行指令，则触发输入的接点必须被再次触发（从 OFF 到 ON）。

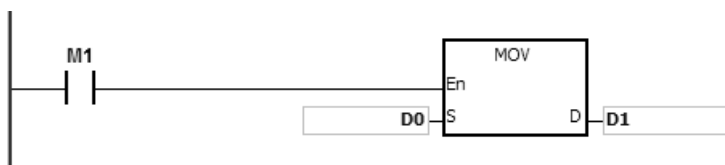
范例：

脉冲执行型



当 M1 由 OFF→ON 变化时，MOV 指令被执行一次，该次扫描指令不再被执行，因此称为脉冲执行型指令。

连续执行型



当 M1=ON 的每次扫描周期，MOV 指令均被执行一次，因此称为连续执行型指令。上图的条件接点 M1=OFF 时，指令不被执行，目的地操作数 D 的内容没有变化。

指令所支持的操作数

● 操作数区域会列出各种指令所用到的操作数。S，D，n，m 等操作数分别代表不同的功能如下：

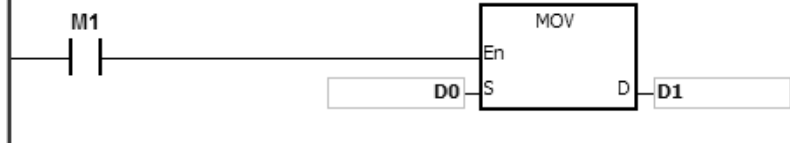
S	数据源 如果不止一个操作数并且功能相同时，会附加编号使用，如 S ₁ ，S ₂ ...
D	数据目的地 如果不止一个操作数并且功能相同时，会附加编号使用，如 D ₁ ，D ₂ ...
N	循环程序结构的层数
若操作数只可指定常数 K/H 或寄存器时，那么则以 m，m ₁ ，m ₂ ，n，n ₁ ，n ₂ 表示。	

以 MOV（数据移动）为例，操作数 S 表示数据移动的来源，操作数 D 表示数据移动的目的地。

3 指令码

3 操作数

指令表字段说明

3	操作数	 <ul style="list-style-type: none"> 应用指令中有些指令仅有指令部份（指令码）构成，例如：EI，DI...或 WDT 等等，但是大部份都是指令部份再加上数个操作数所组合而成。 <p>如果要在功能块中使用指令时，操作数区域中有支持装置 T，C，HC/AC，则必须使用 TR，CR，HCR 指针寄存器来运算，详细说明请参考第二章的装置说明内容。</p>																																																																								
4	功能	指令的功能用途说明																																																																								
5	数据类型	<p>实心圆●表示此指令支持的数据类型</p> <ul style="list-style-type: none"> 操作数为矩阵数据类型时，在操作数后会以括号表示此矩阵大小。例如，操作数 D (array, 3) 在“Word”数据类型以实心圆表示支持时，表示此操作数的数据类型为 3 个 Word。 <table border="1" data-bbox="475 963 805 1164"> <thead> <tr> <th>数据类型</th> <th>BOOL</th> <th>WORD</th> <th>DWORD</th> </tr> </thead> <tbody> <tr> <td>S₁, S₂</td> <td></td> <td>●</td> <td>●*</td> </tr> <tr> <td>D (array, 3)</td> <td>●</td> <td></td> <td></td> </tr> </tbody> </table> <ul style="list-style-type: none"> WORD/UINT/INT 数据类型仅支持 16 位指令，DWORD/UDINT/DINT 数据类型仅支持 32 位指令，实心圆上用星号来作为区分的符号。 <table border="1" data-bbox="475 1288 805 1377"> <thead> <tr> <th>FB/FC</th> <th colspan="3">指令码</th> </tr> </thead> <tbody> <tr> <td>FC</td> <td>D*</td> <td>MOV</td> <td>P</td> </tr> </tbody> </table> <table border="1" data-bbox="475 1400 805 1579"> <thead> <tr> <th>数据类型</th> <th>BOOL</th> <th>WORD</th> <th>DWORD</th> </tr> </thead> <tbody> <tr> <td>S</td> <td></td> <td>●</td> <td>●*</td> </tr> <tr> <td>D</td> <td></td> <td>●</td> <td>●*</td> </tr> </tbody> </table> <ul style="list-style-type: none"> REAL 数据类型仅支持 32 位指令，LWORD/LINT/LREALT 数据类型仅支持 64 位指令，实心圆上用星号来作为区分的标记： <table border="1" data-bbox="475 1713 1449 1803"> <thead> <tr> <th>FB/FC</th> <th colspan="2">指令码</th> <th>操作数</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>FC</td> <td>D*</td> <td>FLD※</td> <td>S₁, S₂</td> <td>浮点数接点类型比较 LD※</td> </tr> </tbody> </table> <table border="1" data-bbox="475 1814 1449 1971"> <thead> <tr> <th>数据类型</th> <th>BOOL</th> <th>WORD</th> <th>DWORD</th> <th>LWORD</th> <th>UINT</th> <th>UDINT</th> <th>INT</th> <th>DINT</th> <th>LINT</th> <th>REAL</th> <th>LREAL</th> <th>TMR</th> <th>CNT</th> <th>STRING</th> </tr> </thead> <tbody> <tr> <td>S₁, S₂</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>●</td> <td>●*</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	数据类型	BOOL	WORD	DWORD	S ₁ , S ₂		●	●*	D (array, 3)	●			FB/FC	指令码			FC	D*	MOV	P	数据类型	BOOL	WORD	DWORD	S		●	●*	D		●	●*	FB/FC	指令码		操作数	功能	FC	D*	FLD※	S ₁ , S ₂	浮点数接点类型比较 LD※	数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING	S ₁ , S ₂										●	●*			
数据类型	BOOL	WORD	DWORD																																																																							
S ₁ , S ₂		●	●*																																																																							
D (array, 3)	●																																																																									
FB/FC	指令码																																																																									
FC	D*	MOV	P																																																																							
数据类型	BOOL	WORD	DWORD																																																																							
S		●	●*																																																																							
D		●	●*																																																																							
FB/FC	指令码		操作数	功能																																																																						
FC	D*	FLD※	S ₁ , S ₂	浮点数接点类型比较 LD※																																																																						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING																																																												
S ₁ , S ₂										●	●*																																																															

指令表字段说明

<p>3</p> <p>6</p>	<p>装置</p>	<p>表示指令每个操作数所支持的装置 空心圆○表示此装置不支持 E 装置修饰；实心圆●表示此装置支持 E 装置修饰</p> <p>● 操作数的指定对象：</p> <p>X：输入继电器：X0.0 ~ X127.15 或 X0 ~ X0~X127 Y：输出继电器：Y0.0 ~ Y127.15 或 Y0 ~ Y127 M：内部继电器：M0 ~ M4095 S：步进标志：S0 ~ S2047 T：定时器：T0 ~ T255 C：16 位计数器：C0~C199 HC/AC：32 位计数器：HC0~HC63，AC0~AC55 D：数据寄存器：D0~D65535 或 D0.0~D65535.15 L：Link 寄存器：L0~L65535 或 L0.0~D65535.15 SM：特殊继电器：SM0~SM2047 SR：特殊寄存器：SR0~SR2047 E：变址寄存器：E0~E15 PR：指针寄存器：PR0~PR15 定时器指针寄存器：TR0~TR7 16 位计数器指针寄存器：CR0~CR7 32 位计数器指针寄存器：HCR0~HCR7 K：十进制常数 16#：十六进制常数 “\$”：字符串 DF：单精度浮点数：F；双精度位浮点数：DF</p> <p>● 说明：</p> <ol style="list-style-type: none"> 1. 寄存器一般为 16 位长度，也就是 1 个寄存器，若指定 32 位长度的数据寄存器时，是指定连续号码的 2 个寄存器。 2. 若 32 位指令的操作数指定 D0，则 (D1, D0) 所组成的 32 位数据寄存器被占用，D1 为上位 16 位，而 D0 为下位 16 位。定时器 T，及 16 位计数器 C 被使用的规则也相同。 3. 32 位计数器 HC/AC 若是当数据寄存器来使用时，只有 32 位指令的操作数可指定。 <p>有关装置的介绍请参考第二章装置说明内容。</p>
<p>7</p>	<p>图形</p>	<p>指令在 ISPSOft 中梯形图显示的图形 图形包含的一般脚位：</p> <p>En：表示使能 S：数据源</p>

指令表字段说明						
7	图形	D：数据目的地 Q：状态输出，可用于驱动输出点或另外串联接点				
8	指令说明	指令功能详细说明				
9	范例	指令功能操作范例				
10	适用机种	说明该指令适用机种，以及适用脉冲执行型、16 位、32 位或 64 位指令。				
		<table border="1"> <thead> <tr> <th>适用机种</th> <th>机种名称</th> </tr> </thead> <tbody> <tr> <td>AH 运动控制主机</td> <td>AH10EMC-5A</td> </tr> </tbody> </table>	适用机种	机种名称	AH 运动控制主机	AH10EMC-5A
		适用机种	机种名称			
AH 运动控制主机	AH10EMC-5A					

3.1.2 指令使用限制

- 只支持功能块中使用的指令
CHKADR、FB_NP、FB_PN、NED、ANED、ONED、PED、APED 及 OPED
- 不支持使用在中断程序的指令
GOEND
- 功能块（Function Block）中不支持的指令
LDP、ANDP、ORP、LDF、ANDF、ORF、PLS、PLF、NP、PN、MC/MCR、GOEND 以及应用命令中所有脉冲执行型指令

上述部分指令若还是要使用在功能块中，可以替代的指令如下

功能块中不可使用指令	功能块中替代指令
LDP/ANDP/ORP	PED/APED/OPED
LDF/ANDF/ORF	NED/ANED/ONED
PLS	-
PLF	-
NP	FB_NP
PN	FB_PN
MC	-
MCR	-
应用命令中所有脉冲执行型指令	注一

注一：脉冲执行型指令，不可使用在功能块中，如果要在 FB 功能块中达到脉冲执行型指令功能，可参考下列范例。

范例：

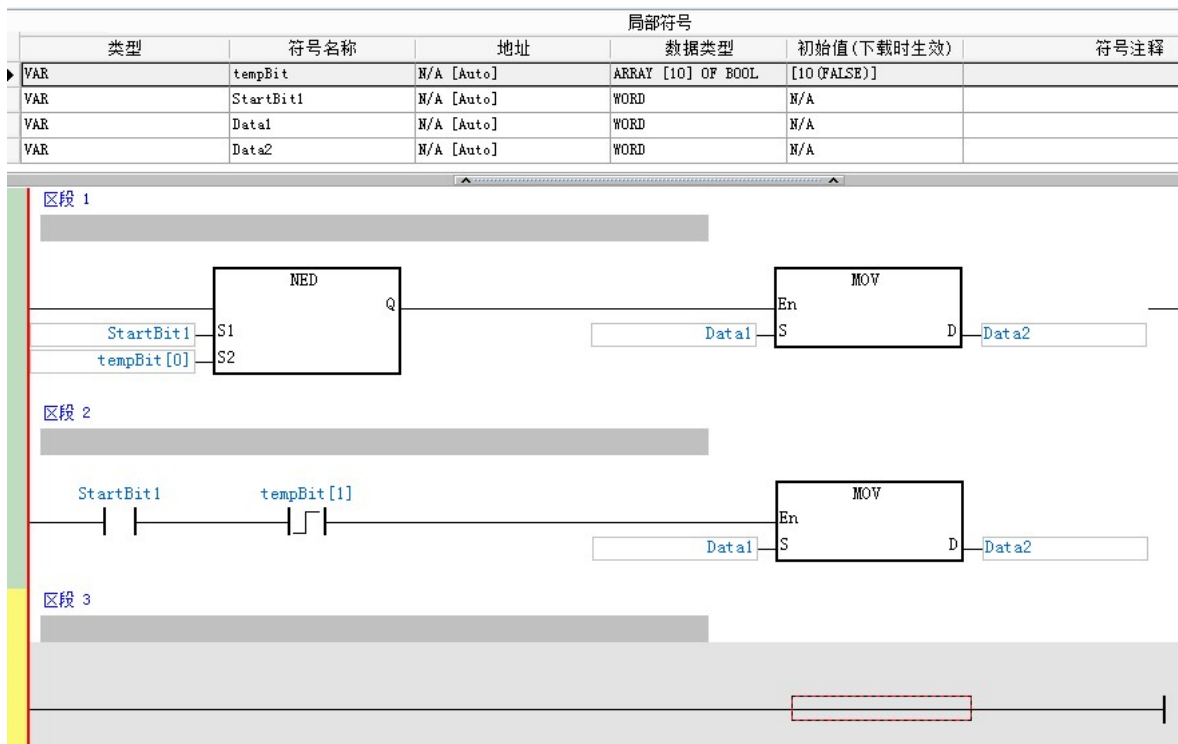
1. 先宣告系统用的 10 个 Bit 变量 tempBit[10]。

2. 当 StartBit1 由 ON=>OFF 时，方法一（NEXTWORK1）执行一次 MOV 指令。

当 StartBit1 由 OFF=>ON 时，方法二（NEXTWORK2）执行一次 MOV 指令。

方法一（NEXTWORK1）与方法二（NEXTWORK2）皆仅会执行一次。

3. 系统用的 tempBit 变量，不可重复使用。



3

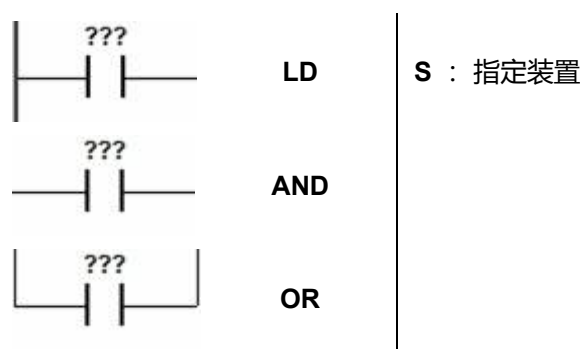
3.2 梯形图指令

FB/FC	指令码	功能	操作数	Step
FC	<u>LD/AND/OR</u>	载入/串联/并联 A 接点	DX、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>LDI/ANI/ORI</u>	载入/串联/并联 B 接点	DX、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>ANB/ORB</u>	串联/并联回路方块	-	1
FC	<u>MPS/MRD/MPP</u>	存入/读出/读出堆栈	-	1
FC	<u>OUT</u>	驱动线圈	DY、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>SET</u>	动作保持 (ON)	DY、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>MC/MCR</u>	共通串联接点的连结/解除	N	1
FC	<u>LDP/ANDP/ORP</u>	上升沿检出动作开始/串联/并联连接	DX、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>LDF/ANDF/ORF</u>	下降沿检出动作开始/串联/并联连接	DX、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>PED/APED/OPED</u>	上升沿检出动作开始/串联/并联连接	X、Y、M、SM、S、T、C、 HC/AC、D、L、PR	5
FC	<u>NED/ANED/ONED</u>	下降沿检出动作开始/串联/并联连接	X、Y、M、SM、S、T、C、 HC/AC、D、L、PR	5
FC	<u>PLS</u>	上升沿检出动作开始/串联/并联连接	DY、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>PLF</u>	下降沿检出动作开始/串联/并联连接	DY、X、Y、M、SM、S、 T、C、HC/AC、D、L、 PR	1-2
FC	<u>INV</u>	运算结果反相	-	1

FB/FC	指令码	功能	操作数	Step
FC	<u>NOP</u>	无动作	-	1
FC	<u>NP</u>	上升沿触发指令	-	1
FC	<u>PN</u>	下降沿触发指令	-	1
FC	<u>FB_NP</u>	上升沿触发指令	X、Y、M、SM、S、T、C、 HC/AC、D、L、PR	1-2
FC	<u>FB_PN</u>	下降沿触发指令	X、Y、M、SM、S、T、C、 HC/AC、D、L	1-2
FC	<u>PSTOP</u>	PLC程序停止执行	-	1

FB/FC	指令码		操作数				功能							
FC	LD/AND/OR		S				载入/串联/并联 A 接点							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													
装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR	
S	●		●	●	●	●	●	●	●	●	●	●	●	

图形：

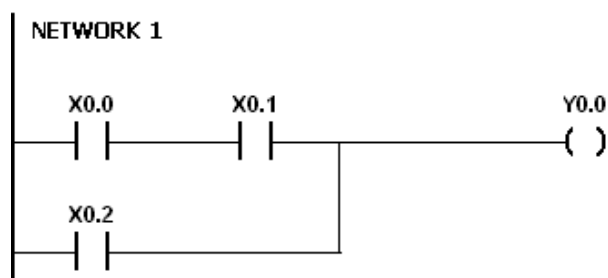


指令说明：

- LD 指令用于左母线开始的 A 接点或一个接点回路块开始的 A 接点，它的作用是把当前内容保存，同时把取来的接点状态存入累积寄存器内。
- AND 指令用于 A 接点的串联连接，先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“与”（AND）的运算，并将结果存入累积寄存器内。
- OR 指令用于 A 接点的并联连接，它的作用是先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累积寄存器内。

程序范例：

- 载入 X0.0 的 A 接点，串联 X0.1 的 A 接点，并联 X0.2 的 A 接点，驱动 Y0.0 线圈。
- 当 X0.0 和 X0.1=ON 或 X0.2=ON 时，Y0.0=ON。



FB/FC	指令码	操作数	功能
FC	LDI/ANI/ORI	S	载入/串联/并联 B 接点

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													

装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR
S	●		●	●	●	●	●	●	●	●	●	●	●

图形：

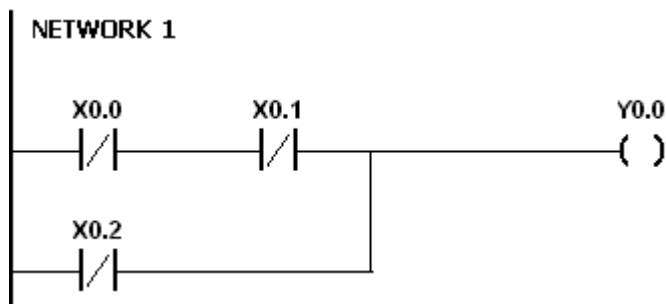


指令说明：

- LD 指令用于左母线开始的 B 接点或一个接点回路块开始的 B 接点，它的作用是把当前内容保存，同时把取来的接点状态存入累积寄存器内。
- AND 指令用于 B 接点的串联连接，先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“与”（AND）的运算，并将结果存入累积寄存器内。
- OR 指令用于 B 接点的并联连接，它的作用是先读取目前所指定串联接点的状态再与接点之前逻辑运算结果作“或”（OR）的运算，并将结果存入累积寄存器内。

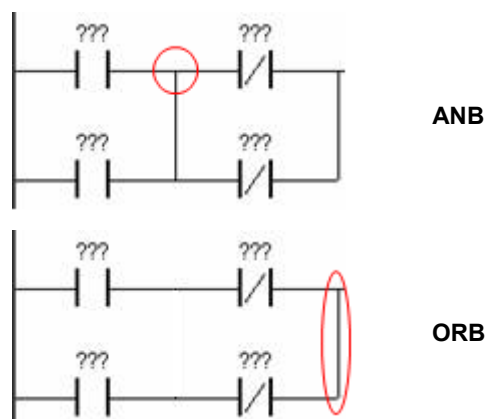
程序范例：

- 载入 X0.0 的 B 接点，串联 X0.1 的 B 接点，并联 X0.2 的 B 接点，驱动 Y0.0 线圈。
- 当 X0.0 和 X0.1=OFF 或 X0.2=OFF 时，Y0.0=ON。



FB/FC	指令码	操作数	功能
FC	ANB/ORB	-	串联/并联回路方块

图形：

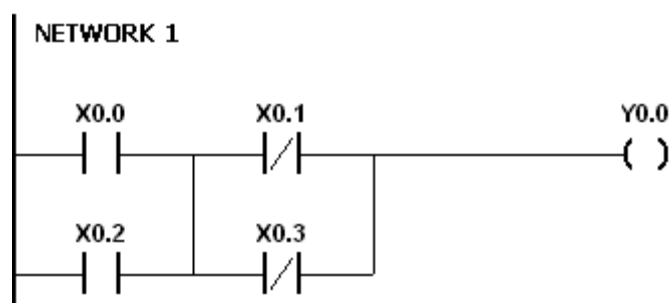


指令说明：

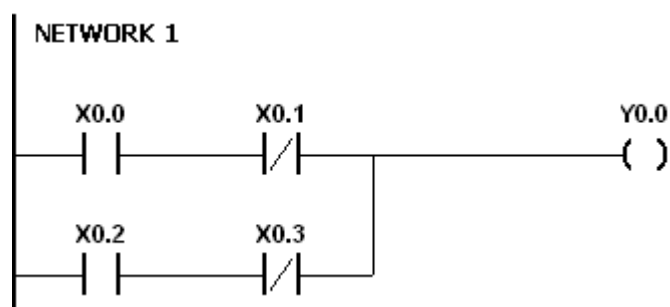
1. ANB 是将前一保存的逻辑结果与目前累积寄存器的内容作“与”（AND）的运算。
2. ORB 是将前一保存的逻辑结果与目前累积寄存器的内容作“或”（OR）的运算。

程序范例：

1. 载入 A 接点 X0.0，并联 A 接点 X0.2，载入 B 接点 X0.1，并联 B 接点 X0.3，串联回路方块，驱动 Y0.0 线圈。



2. 载入 A 接点 X0.0，串联 B 接点 X0.1，载入 A 接点 X0.2，串联 B 接点 X0.3，并联回路方块，驱动 Y0.0 线圈。



FB/FC	指令码	操作数	功能
FC	MPS/MRD/MPP	-	存入/读出/读出堆栈

指令说明：

1. MPS 存入堆栈指令，将目前累积寄存器的内容存入堆栈（堆栈指针加一）。
2. MRD 读出指令，读取堆栈内容存入累积寄存器（堆栈指针不动）。
3. MPP 读出指令，自堆栈取回前一保存的逻辑运算结果，存入累积寄存器（堆栈指针减一）。

程序范例：

1. 加载 X0 的 A 接点，存入堆栈。
2. 串联 X1 的 A 接点，驱动 Y1 线圈，读出堆栈（指针不动）。
3. 串联 X2 的 A 接点，驱动 M0 线圈，读出堆栈。

指令： **操作说明：**

LD X0 载入 X0 的 A 接点

MPS 存入堆栈

AND X1 串联X1的A接点

OUT Y1 驱动Y1线圈

MRD 读出堆栈（指针不动）

AND X2 串联X2的A接点

OUT M0 驱动M0线圈

MPP 读出堆栈

OUT Y2 驱动Y2线圈

END 程序结束

备注：

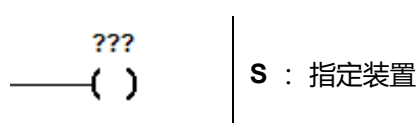
1. MPS 与 MPP 要一一对应。
2. MPS 指令最多可以连续使用 31 次。

FB/FC	指令码	操作数	功能
FC	OUT	S	驱动线圈

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													

装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR
S		●	●	●	●	●	●	●	●	●	●	●	●

图形：



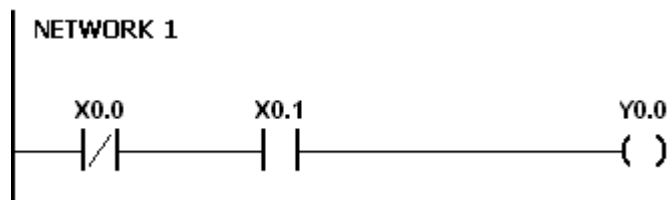
指令说明：

1. 将 OUT 指令之前的逻辑运算结果输出至指定的元件。
2. 线圈接点动作：

运算结果	OUT 指令		
	线圈	接点	
		A 接点 (常开)	B 接点 (常闭)
False	OFF	不导通	导通
True	ON	导通	不导通

程序范例：

1. 加载 X0.0 的 B 接点，串联 X0.1 的 A 接点，驱动 Y0.0 线圈。
2. 当 X0.0=OFF 且 X0.1=ON 时，Y0.0=ON。

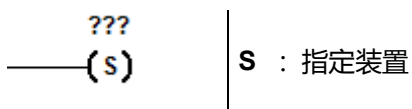


FB/FC	指令码	操作数	功能
FC	SET	S	动作保持 (ON)

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													

装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR
S		●	●	●	●	●	●	●	●	●	●	●	●

图形：



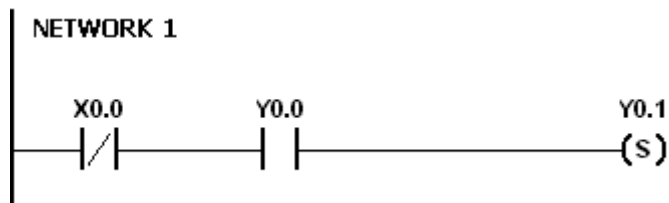
3

指令说明：

当 SET 指令被驱动，其指定的元件被设定为 ON，且被设定的元件会维持 ON，不管 SET 指令是否仍被驱动。可利用 RST 指令将该元件设为 OFF。

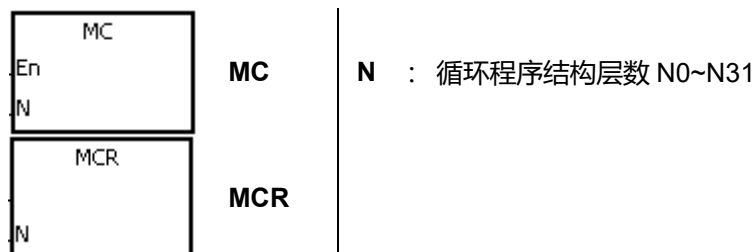
程序范例：

1. 载入 X0.0 的 B 接点，串联 Y0.0 的 A 接点，Y0.1 动作保持 (ON)。
2. 当 X0.0=OFF 且 Y0.0=ON 时，Y0.1=ON 且即使运算结果改变，Y0.1 也保持 ON 的状态。



FB/FC	指令码	操作数	功能
FC	MC/MCR	N	共通串联接点的连结/解除

图形：



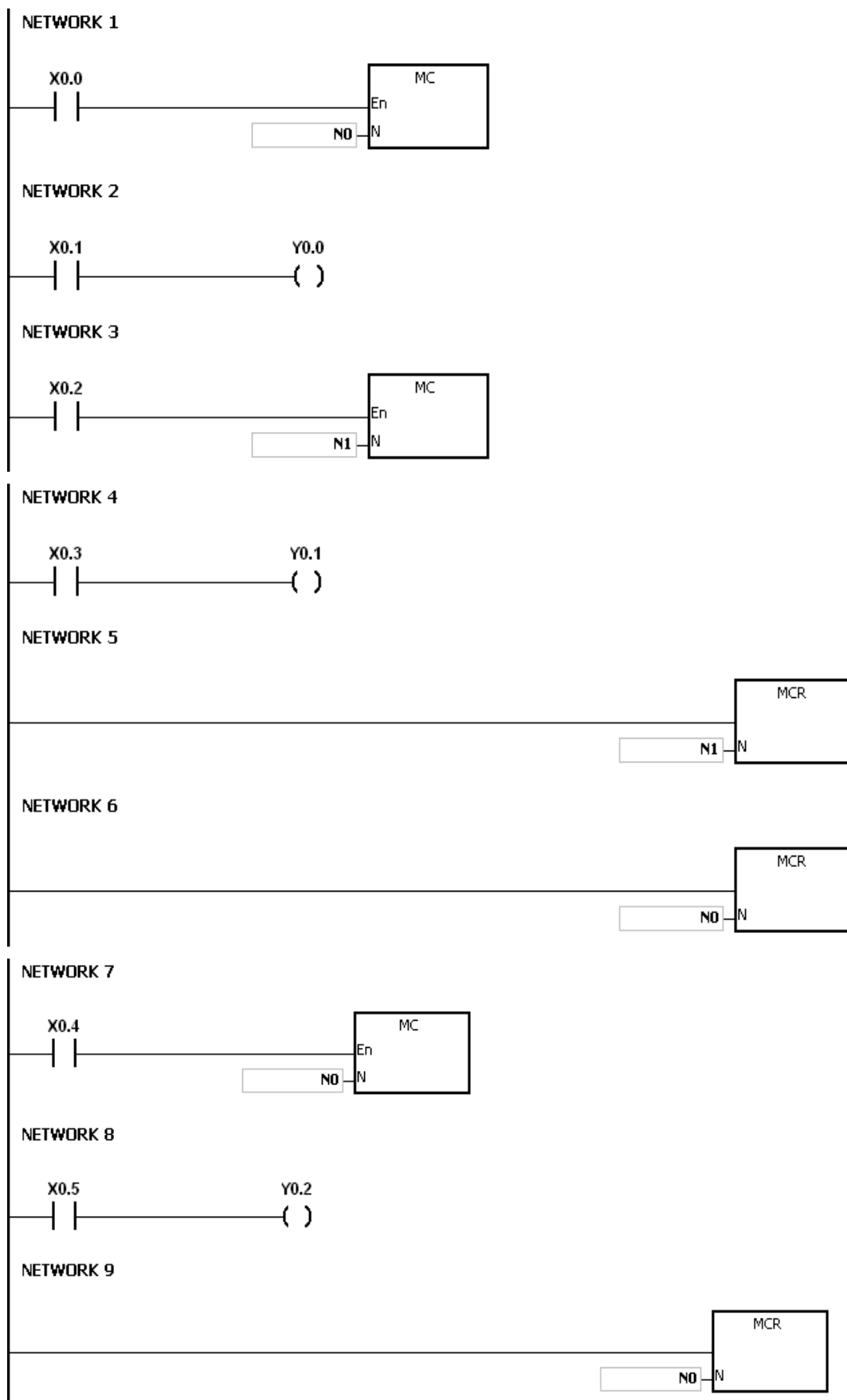
指令说明：

1. MC 为主控起始指令，当 MC 指令执行时，位于 MC 与 MCR 指令之间的指令照常执行。当 MC 指令 OFF 时，位于 MC 与 MCR 指令之间的指令动作如下所示：

指令区分	说明
一般定时器	计时值归零，线圈失电，接点不动作
功能块用定时器	计时值归零，线圈失电，接点不动作
运算型定时器	线圈失电，计时值及接点保持目前状态
计数器	线圈失电，计数值及接点保持目前状态
OUT 指令驱动的线圈	全部不受电
SET, RST 指令驱动的元件	保持目前状态
应用指令	全部不动作，但 FOR-NEXT 循环回路仍会来回执行 N 次，但 FOR-NEXT 间的任何指令依 MC-MCR 之间其它指令相同动作

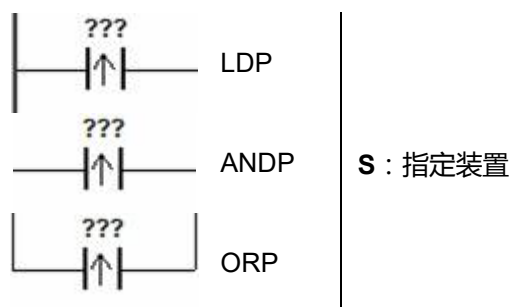
2. MCR 为主控结束指令，置于主控程序最后，在 MCR 指令之前不可有接点指令。
3. MC-MCR 主控程序指令支持循环程序结构，最多可 32 层，使用时依 N0~N31 的顺序，请参考程序范例所示。

程序范例：



FB/FC	指令码			操作数					功能					
FC	LDP/ANDP/ORP			S					上升沿检出动作开始/串联/并联连接					
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													
装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR	
S	●		●	●	●	●	●	●	●	●	●	●	●	

图形：

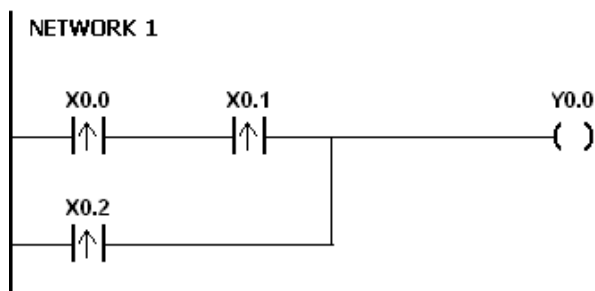


指令说明：

1. LDP 指令用法上与 LD 相同，但动作不同，它的作用是指当前内容保存，同时把取来的接点上升沿检出状态存入累积寄存器内。
2. ANDP 指令用于接点上升沿检出的串联连接。
3. ORP 指令用于接点上升沿检出的并联连接。
4. 上升沿检出动作，必须在指令扫描到的时候才会得知装置目前的状态，下一次扫描到指令才会判断装置状态是否有变化。
5. 子程序中请使用 PED、APED、OPED 指令。

程序范例：

1. X0.0 上升沿检出动作开始，串联 X0.1 的上升沿检出，并联 X0.2 的上升沿检出，驱动 Y0.0 线圈。
2. 当 X0.0 和 X0.1 同时由 OFF 到 ON 或 X0.2 由 OFF 到 ON 时，Y0.0 会 ON 一个扫描周期。

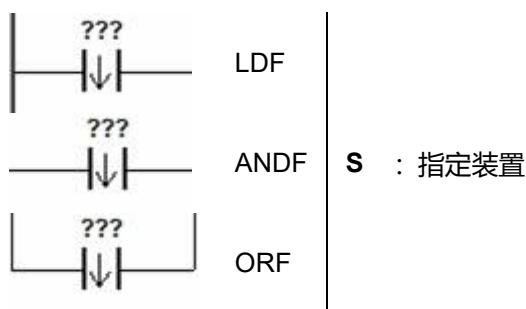


FB/FC	指令码	操作数	功能
FC	LDF/ANDF/ORF	S	下降沿检出动作开始/串联/并联连接

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													

装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR
S	●		●	●	●	●	●	●	●	●	●	●	●

图形：

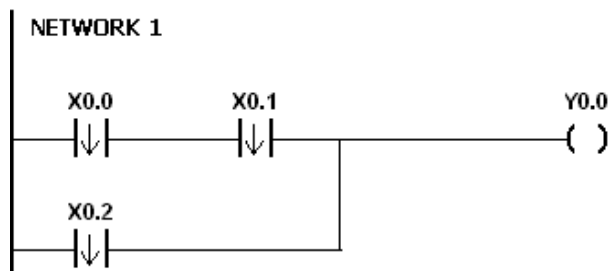


指令说明：

1. LDF 指令用法上与 LD 相同，但动作不同，它的作用是指当前内容保存，同时把取来的接点下降沿检出状态存入累积寄存器内。
2. ANDF 指令用于接点下降沿检出的串联连接。
3. ORF 指令用于接点下降沿检出的并联连接。
4. 下降沿检出动作，必须在指令扫描到的时候才会得知装置目前的状态，下一次扫描到指令才会判断装置状态是否有变化
5. 子程序中请使用 NED、ANED、ONED 指令。

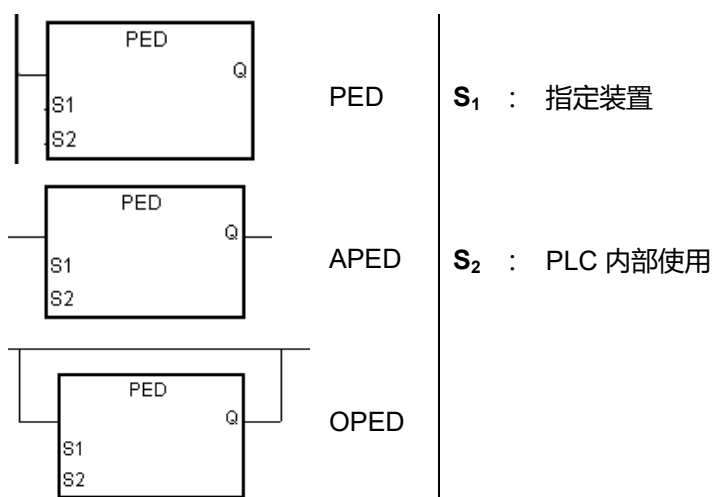
程序范例：

1. X0.0 下降沿检出动作开始，串联 X0.1 的下降沿检出，并联 X0.2 的下降沿检出，驱动 Y0.0 线圈。
2. 当 X0.0 和 X0.1 同时由 ON 到 OFF 或 X0.2 由 ON 到 OFF 时，Y0.0 会 ON 一个扫描周期。



FB/FC	指令码		操作数					功能						
FC	PED/APED/OPED		S ₁ , S ₂					上升沿检出动作开始/串联/并联连接						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂	●													
装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR	
S ₁ , S ₂			●	●	●	●	●	●	●	●	●	●	●	

图形：

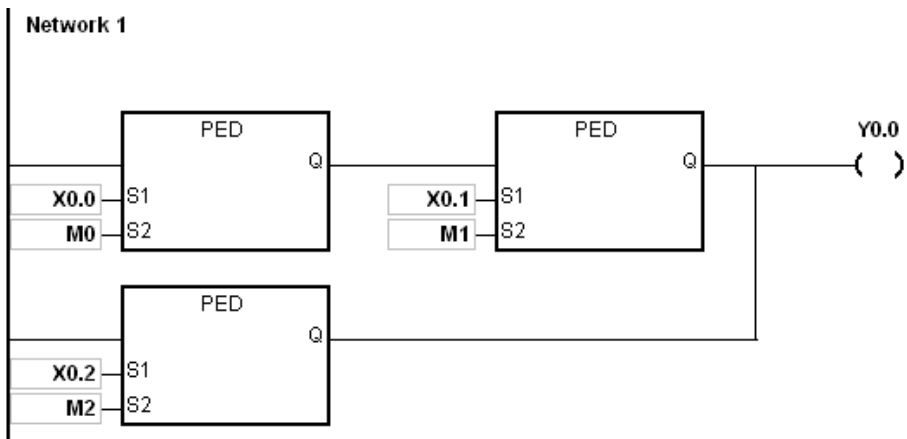


指令说明：

1. PED 指令用法上与 LDP 相同、APED 指令用法上与 ANDP 相同、OPED 指令用法上与 ORP 相同，不同之处在于需多指定一个 S₂ 的 Bit 装置，让 PLC 内部记住该接点的上一次状态，此 S₂ 装置在程序中的其他地方请勿重复使用，否则可能会造成执行结果错误。
2. APED 指令用于接点上升沿检出的串联连接。
3. OPED 指令用于接点上升沿检出的并联连接。
4. 上升沿检出动作，必须在指令扫描到的时候才会得知装置目前的状态，下一次扫描到指令才会判断装置状态是否有变化。
5. PED/APED/OPED 指令只能在功能块中使用。

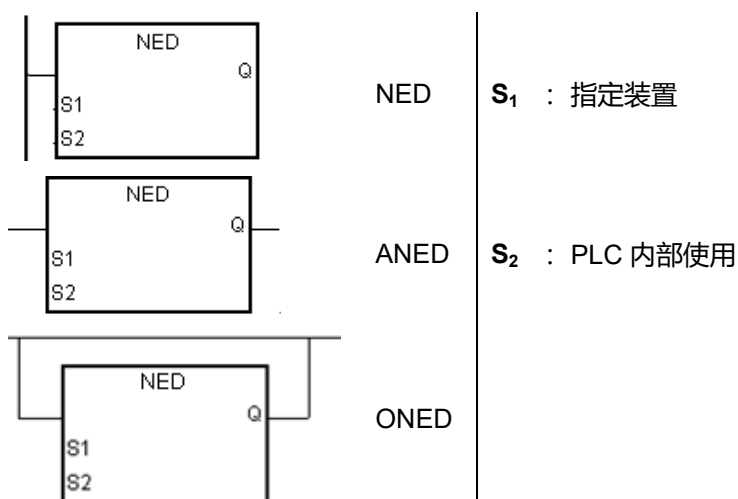
程序范例：

1. X0.0 上升沿检出动作开始，串联 X0.1 的上升沿检出，并联 X0.2 的上升沿检出，驱动 Y0.0 线圈。
2. 当 X0.0 和 X0.1 同时由 OFF 到 ON 或 X0.2 由 OFF 到 ON 时，Y0.0 会 ON 一个扫描周期。



FB/FC	指令码		操作数				功能							
FC	NED/ANED/ONED		S ₁ , S ₂				下降沿检出动作开始/串联/并联连接							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂	●													
装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR	
S ₁ , S ₂		●	●	●	●	●	●	●	●	●	●	●	●	

图形：

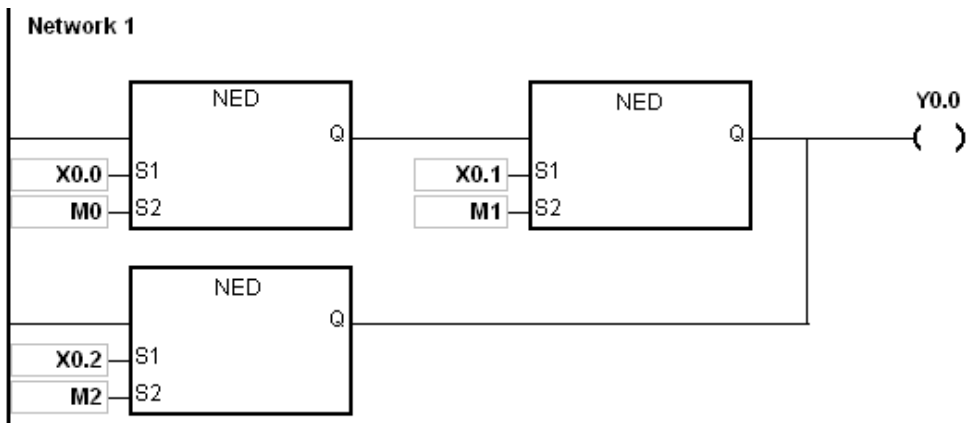


指令说明：

1. NED 指令用法上与 LDF 相同、ANED 指令用法上与 ANDF 相同、ONED 指令用法上与 ORF 相同，不同之处在于需多指定一个 S₂ 的 Bit 装置，让 PLC 内部记住该接点的上一次状态，此 S₂ 装置在程序中的其它地方请勿重复使用，否则可能会造成执行结果错误。
2. ANED 指令用于接点下降沿检出的串联连接。
3. ONED 指令用于接点下降沿检出的并联连接。
4. 下降沿检出动作，必须在指令扫描到的时候才会得知装置目前的状态，下一次扫描到指令才会判断装置状态是否有变化。
5. NED/ANED/ONED 指令只能在功能块中使用。

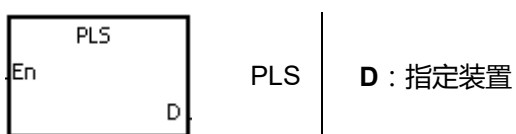
程序范例：

1. X0.0 下降沿检出动作开始，串联 X0.1 的下降沿检出，并联 X0.2 的下降沿检出，驱动 Y0.0 线圈。
2. 当 X0.0 和 X0.1 同时由 ON 到 OFF 或 X0.2 由 ON 到 OFF 时，Y0.0 会 ON 一个扫描周期。



FB/FC	指令码		操作数					功能						
FC	PLS		D					上微分输出						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●													
装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR	
D		●	●	●	●	●	●	●	●	●	●	●	●	

图形：

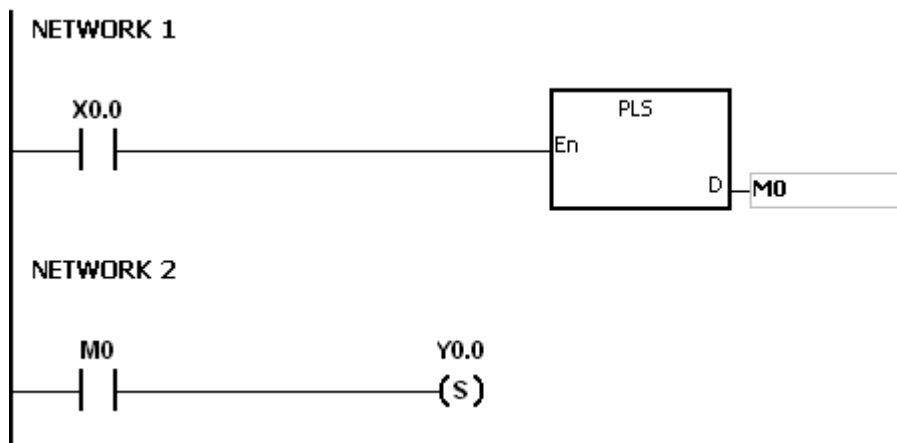


指令说明：

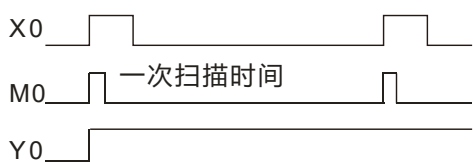
1. 上微分输出指令。当条件接点由 OFF 到 ON (上升沿触发) 时，PLS 指令被执行，D 送出一脉冲，脉冲长度为一次扫描时间。
2. 功能块中请勿使用。

程序范例：

当 X0.0=ON 时，M0 ON 一个脉冲的时间；M0=ON 时，Y0.0 Set 为 ON。



时序图：



FB/FC	指令码	操作数	功能
FC	PLF	D	下微分输出

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●													

装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR
D		●	●	●	●	●	●	●	●	●	●	●	●

图形：

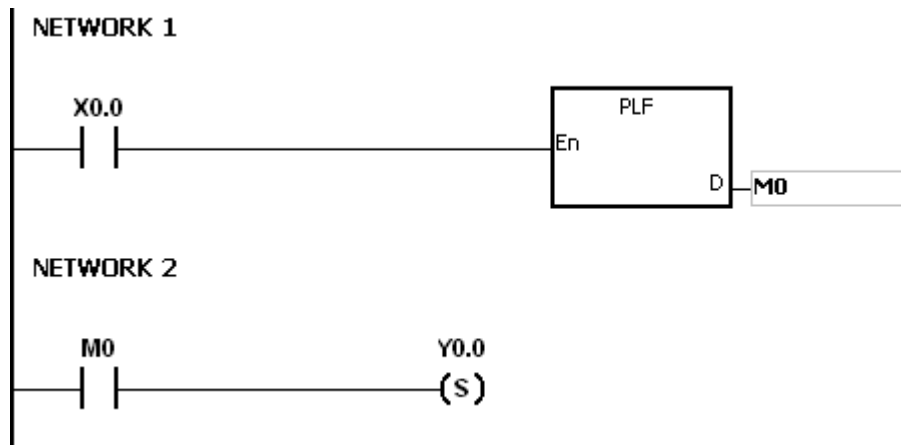


指令说明：

1. 下微分输出指令。当条件接点由 ON 到 OFF (下降沿触发) 时, PLF 指令被执行, D 送出一脉冲, 脉冲长度为一次扫描时间。
2. 功能块中请勿使用。

程序范例：

当 X0.0=ON 时, M0 ON 一个脉冲的时间; M0=ON 时, Y0.0 Set 为 ON。



时序图：



FB/FC	指令码	操作数	功能
FC	INV	-	运算结果反相

图形：

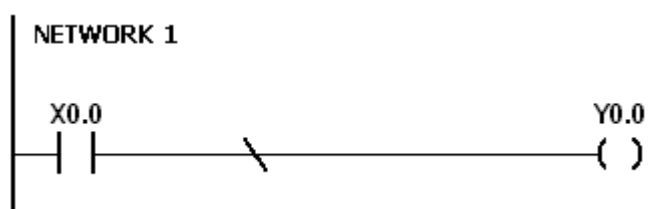


指令说明：

将 INV 指令之前的逻辑运算结果反相存入累积寄存器内。

程序范例：

当 X0.0=ON 时，Y0.0=OFF；X0.0=OFF 时，Y0.0=ON。



FB/FC	指令码	操作数	功能
FC	NOP	-	无动作

指令说明：

指令 NOP 在程序不做任何运算，因此执行后仍会保持原逻辑运算结果，使用时机如下：想要删除某一指令，而又不想改变程序长度，则可以 NOP 指令取代。

本指令仅支持 ISPSOft 的指令列表（Instruction List）功能编辑，不支持梯形图编辑。

程序范例：

ISPSOft 的指令列表中

指令：	操作说明：
LD X0.0	载入 X0 的 A 接点
NOP	无动作
OUT Y1.0	驱动线圈 Y1.0

FB/FC	指令码	操作数	功能
FC	NP	-	上升沿触发指令

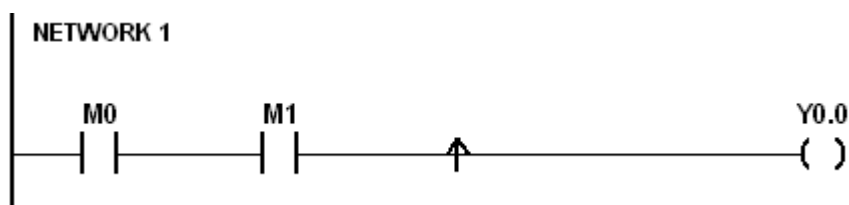
图形：



指令说明：

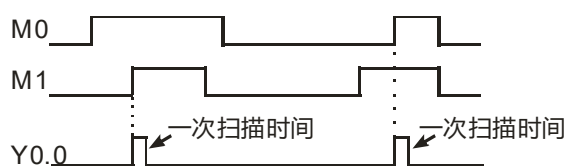
1. 当累积寄存器由 0 变为 1 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. 在 Function Block 中请使用 FB_NP 指令。

程序范例：



指令：	操作说明：
LD M0	载入 M0 的 A 接点
AND M1	串入 M1 之 A 接点
NP	上升沿触发指令
OUT Y0.0	驱动线圈 Y0.0

时序图：



FB/FC	指令码	操作数	功能
FC	PN	-	下降沿触发指令

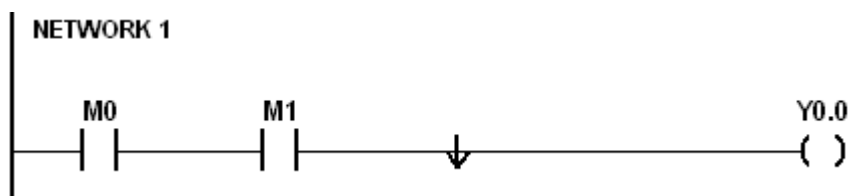
图形：



指令说明：

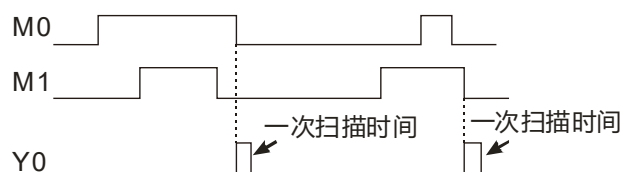
1. 当累积寄存器由 1 变为 0 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. 在 Function Block 中请使用 FB_P N 指令。

程序范例：



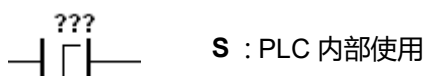
指令：	操作说明：
LD M0	载入 M0 的 A 接点
AND M1	串入 M1 之 A 接点
PN	下降沿触发指令
OUT Y0.0	驱动线圈 Y0.0

时序图：



FB/FC	指令码		操作数					功能						
FC	FB_NP		S					上升沿触发指令						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													
装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR	
S			●	●	●	●	●	●	●	●	●	●	●	

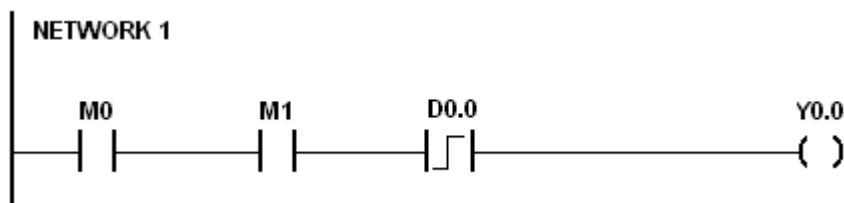
图形：



指令说明：

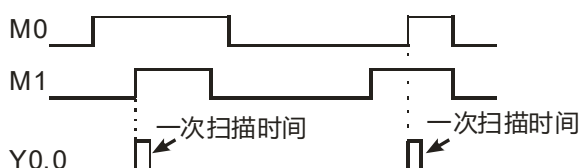
1. 当累积寄存器由 0 变为 1 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. **S** 装置，让 PLC 内部记住该接点的上一次状态，此 S 装置在程序中的其它地方请勿重复使用，否则可能会造成执行结果错误。
3. **本指令只能在 Function Block 中使用。**

程序范例：



指令：		操作说明：
LD	M0	载入 M0 的 A 接点
AND	M1	串入 M1 之 A 接点
FB_NP	D0.0	上升沿触发指令
OUT	Y0.0	驱动线圈 Y0.0

时序图：

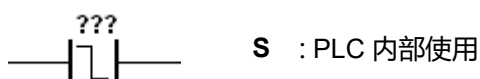


FB/FC	指令码	操作数	功能
FC	FB_PN	S	下降沿触发指令

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													

装置	DX	DY	X	Y	M	SM	S	T	C	HC/AC	D	L	PR
S			●	●	●	●	●	●	●	●	●	●	●

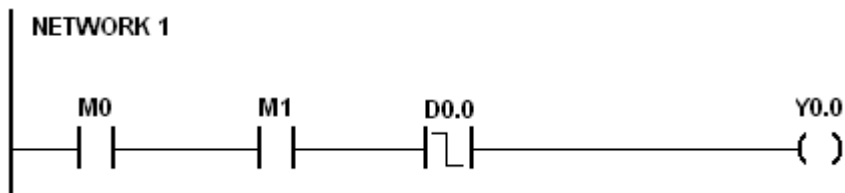
图形：



3 指令说明：

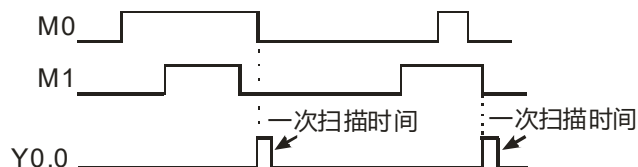
1. 当累积寄存器由 1 变为 0 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. **S** 装置，让 PLC 内部记住该接点的上一次状态，此 S 装置在程序中的其它地方请勿重复使用，否则可能会造成执行结果错误。
3. 本指令只能在 Function Block 中使用。

程序范例：



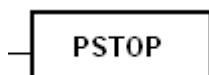
指令：		操作说明：
LD	M0	载入 M0 的 A 接点
AND	M1	串入 M1 之 A 接点
FB_PN	D0.0	下降沿触发指令
OUT	Y0.0	驱动线圈 Y0.0

时序图：



FB/FC	指令码	操作数	功能
FC	PSTOP	-	PLC 程序停止执行

图形：

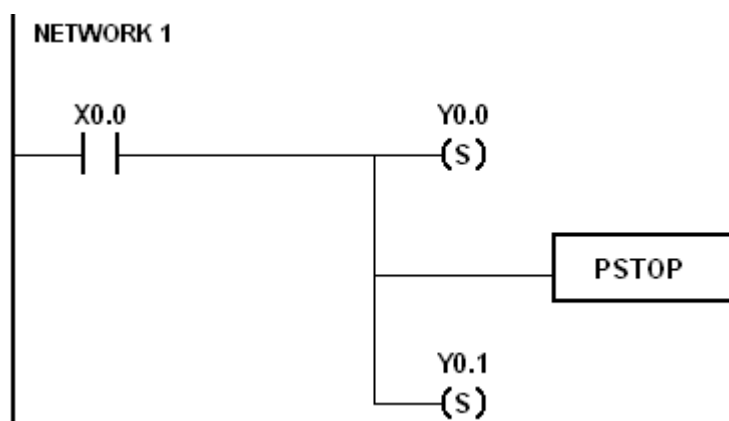


指令说明：

当条件接点启动时，指令执行 END 动作，并且将 PLC 的状态由 RUN 变成 STOP。

程序范例：

当 X0.0=ON 时，Y0.0 被设定为 ON，且 Y0.1 保持 OFF 的状态，且 PLC 的状态变成 STOP。



3.3 比较操作指令

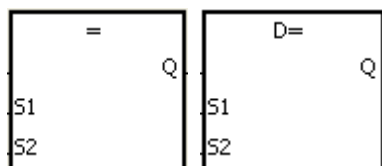
FB/FC	指令码 (位)			P 指令	功能	Step
	16	32	64			
FC	<u>LD=</u>	<u>DLD=</u>	—	—	$S_1 = S_2$	5
FC	<u>LD<></u>	<u>DLD<></u>	—	—	$S_1 \neq S_2$	5
FC	<u>LD></u>	<u>DLD></u>	—	—	$S_1 > S_2$	5
FC	<u>LD>=</u>	<u>DLD>=</u>	—	—	$S_1 \geq S_2$	5
FC	<u>LD<</u>	<u>DLD<</u>	—	—	$S_1 < S_2$	5
FC	<u>LD<=</u>	<u>DLD<=</u>	—	—	$S_1 \leq S_2$	5
FC	<u>AND=</u>	<u>DAND=</u>	—	—	$S_1 = S_2$	5
FC	<u>AND<></u>	<u>DAND<></u>	—	—	$S_1 \neq S_2$	5
FC	<u>AND></u>	<u>DAND></u>	—	—	$S_1 > S_2$	5
FC	<u>AND>=</u>	<u>DAND>=</u>	—	—	$S_1 \geq S_2$	5
FC	<u>AND<</u>	<u>DAND<</u>	—	—	$S_1 < S_2$	5
FC	<u>AND<=</u>	<u>DAND<=</u>	—	—	$S_1 \leq S_2$	5
FC	<u>OR=</u>	<u>DOR=</u>	—	—	$S_1 = S_2$	5
FC	<u>OR<></u>	<u>DOR<></u>	—	—	$S_1 \neq S_2$	5
FC	<u>OR></u>	<u>DOR></u>	—	—	$S_1 > S_2$	5
FC	<u>OR>=</u>	<u>DOR>=</u>	—	—	$S_1 \geq S_2$	5
FC	<u>OR<</u>	<u>DOR<</u>	—	—	$S_1 < S_2$	5
FC	<u>OR<=</u>	<u>DOR<=</u>	—	—	$S_1 \leq S_2$	5
FC	—	<u>FLD=</u>	<u>DFLD=</u>	—	$S_1 = S_2$	5-7
FC	—	<u>FLD<></u>	<u>DFLD<></u>	—	$S_1 \neq S_2$	5-7
FC	—	<u>FLD></u>	<u>DFLD></u>	—	$S_1 > S_2$	5-7
FC	—	<u>FLD>=</u>	<u>DFLD>=</u>	—	$S_1 \geq S_2$	5-7
FC	—	<u>FLD<</u>	<u>DFLD<</u>	—	$S_1 < S_2$	5-7
FC	—	<u>FLD<=</u>	<u>DFLD<=</u>	—	$S_1 \leq S_2$	5-7
FC	—	<u>FAND=</u>	<u>DFAND=</u>	—	$S_1 = S_2$	5-7
FC	—	<u>FAND<></u>	<u>DFAND<></u>	—	$S_1 \neq S_2$	5-7

FB/FC	指令码 (位)			P 指令	功能	Step
	16	32	64			
FC	–	<u>FAND</u> >	<u>DFAND</u> >	–	$S_1 > S_2$	5-7
FC	–	<u>FAND</u> >=	<u>DFAND</u> >=	–	$S_1 \geq S_2$	5-7
FC	–	<u>FAND</u> <	<u>DFAND</u> <	–	$S_1 < S_2$	5-7
FC	–	<u>FAND</u> <=	<u>DFAND</u> <=	–	$S_1 \leq S_2$	5-7
FC	–	<u>FOR</u> =	<u>DFOR</u> =	–	$S_1 = S_2$	5-7
FC	–	<u>FOR</u> <>	<u>DFOR</u> <>	–	$S_1 \neq S_2$	5-7
FC	–	<u>FOR</u> >	<u>DFOR</u> >	–	$S_1 > S_2$	5-7
FC	–	<u>FOR</u> >=	<u>DFOR</u> >=	–	$S_1 \geq S_2$	5-7
FC	–	<u>FOR</u> <	<u>DFOR</u> <	–	$S_1 < S_2$	5-7
FC	–	<u>FOR</u> <=	<u>DFOR</u> <=	–	$S_1 \leq S_2$	5-7
FC	<u>LD\$</u> =	–	–	–	$S_1 = S_2$	5-17
FC	<u>LD\$</u> <>	–	–	–	$S_1 \neq S_2$	5-17
FC	<u>LD\$</u> >	–	–	–	$S_1 > S_2$	5-17
FC	<u>LD\$</u> >=	–	–	–	$S_1 \geq S_2$	5-17
FC	<u>LD\$</u> <	–	–	–	$S_1 < S_2$	5-17
FC	<u>LD\$</u> <=	–	–	–	$S_1 \leq S_2$	5-17
FC	<u>AND\$</u> =	–	–	–	$S_1 = S_2$	5-17
FC	<u>AND\$</u> <>	–	–	–	$S_1 \neq S_2$	5-17
FC	<u>AND\$</u> >	–	–	–	$S_1 > S_2$	5-17
FC	<u>AND\$</u> >=	–	–	–	$S_1 \geq S_2$	5-17
FC	<u>AND\$</u> <	–	–	–	$S_1 < S_2$	5-17
FC	<u>AND\$</u> <=	–	–	–	$S_1 \leq S_2$	5-17
FC	<u>OR\$</u> =	–	–	–	$S_1 = S_2$	5-17
FC	<u>OR\$</u> <>	–	–	–	$S_1 \neq S_2$	5-17
FC	<u>OR\$</u> >	–	–	–	$S_1 > S_2$	5-17
FC	<u>OR\$</u> >=	–	–	–	$S_1 \geq S_2$	5-17
FC	<u>OR\$</u> <	–	–	–	$S_1 < S_2$	5-17

FB/FC	指令码 (位)			P 指令	功能	Step
	16	32	64			
FC	<u>OR\$<=</u>	—	—	—	$S_1 \leq S_2$	5-17
FC	<u>CMP</u>	<u>DCMP</u>	—	✓	比较设定输出	7
FC	<u>ZCP</u>	<u>DZCP</u>	—	✓	区域比较	9
FC	—	<u>FCMP</u>	—	✓	浮点数比较	7-9
FC	—	<u>FZCP</u>	—	✓	浮点数区域比较	9-12
FC	<u>MCMP</u>	—	—	✓	矩阵比较	9
FC	<u>CMPT=</u>	—	—	✓	表格比较, =	9
FC	<u>CMPT<></u>	—	—	✓	表格比较, <>	9
FC	<u>CMPT></u>	—	—	✓	表格比较, >	9
FC	<u>CMPT>=</u>	—	—	✓	表格比较, >=	9
FC	<u>CMPT<</u>	—	—	✓	表格比较, <	9
FC	<u>CMPT<=</u>	—	—	✓	表格比较, <=	9
FC	<u>CHKADR</u>	—	—	—	接点类型指针寄存器地址检查	7

FB/FC	指令码				操作数					功能								
FC	D*	LD※			S ₁ , S ₂					接点类型比较 LD※								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○			
					脉冲执行型			16 位指令				32 位指令						
					-			AH 运动控制主机				AH 运动控制主机						

图形：



以 LD=跟 DLD=为例

S₁ : 数据源 1S₂ : 数据源 2

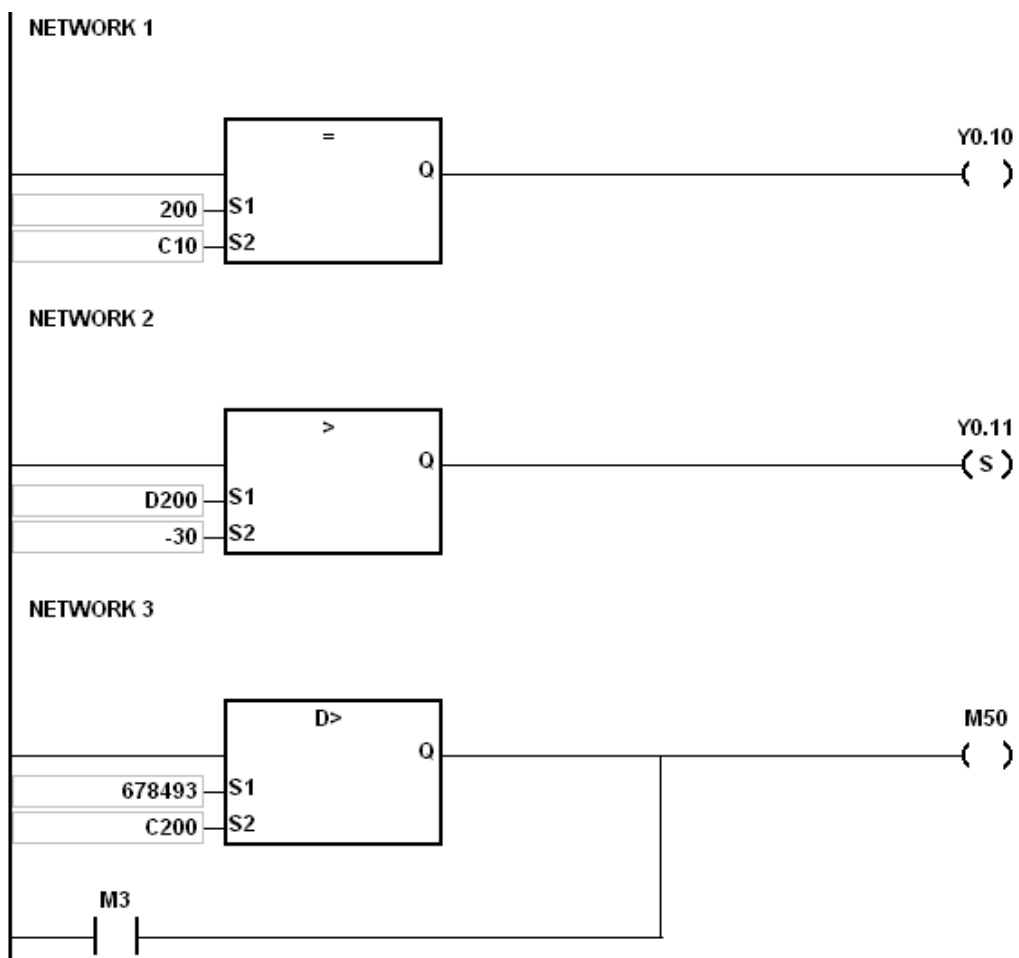
指令说明：

- S₁ 与 S₂ 的内容做比较的指令。以“LD=”做为例子，比较结果为“等于”时，该指令导通；“不等于”时，该指令不导通。
- 32 位指令才可以使用 HC 装置。

16 位指令	32 位指令	导通条件	非导通条件
LD =	DLD =	S ₁ = S ₂	S ₁ ≠ S ₂
LD < >	DLD < >	S ₁ ≠ S ₂	S ₁ = S ₂
LD >	DLD >	S ₁ > S ₂	S ₁ ≤ S ₂
LD > =	DLD > =	S ₁ ≥ S ₂	S ₁ < S ₂
LD <	DLD <	S ₁ < S ₂	S ₁ ≥ S ₂
LD < =	DLD < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

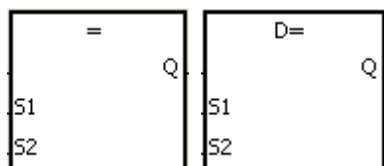
- C10 的内容等于 200 时，Y0.10=ON。
- 当 D200 的内容大于-30，Y0.11=ON 并保持住。
- (C201, C200) 的内容小于 678,493 或者是 M3=ON 时，M50=ON。



3

FB/FC	指令码				操作数					功能								
FC	D*	AND※			S ₁ , S ₂					接点类型比较 AND※								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○			
					脉冲执行型			16 位指令				32 位指令						
					-			AH 运动控制主机				AH 运动控制主机						

图形：

S₁ : 数据源 1S₂ : 数据源 2

以 AND=跟 DAND=为例

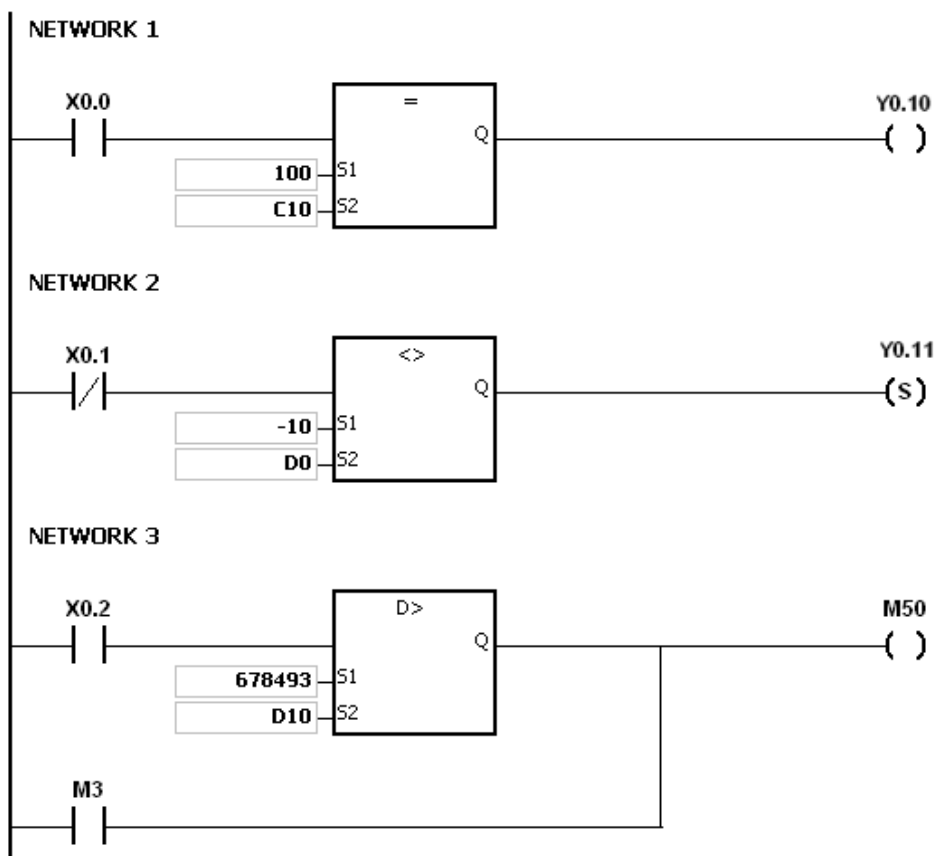
指令说明：

1. S₁ 与 S₂ 的内容做比较的指令。以“AND=”做为例子，比较结果为“等于”时，该指令导通；“不等于”时，该指令不导通。
2. 32 位指令才可以使用 HC/AC 装置。

16 位指令	32 位指令	导通条件	非导通条件
AND =	DAND =	S ₁ = S ₂	S ₁ ≠ S ₂
AND < >	DAND < >	S ₁ ≠ S ₂	S ₁ = S ₂
AND >	DAND >	S ₁ > S ₂	S ₁ ≤ S ₂
AND > =	DAND > =	S ₁ ≥ S ₂	S ₁ < S ₂
AND <	DAND <	S ₁ < S ₂	S ₁ ≥ S ₂
AND < =	DAND < =	S ₁ ≤ S ₂	S ₁ > S ₂

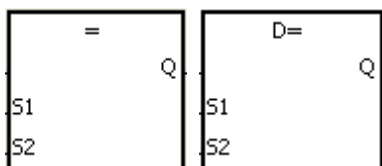
程序范例：

1. 当 X0.0=ON 时且 C10 的当前值又等于 100 时，Y0.10=ON。
2. 当 X0.1=OFF 而寄存器 D0 的内容又不等于-10 时，Y0.11=ON 并保持住。
3. 当 X0.2=ON 而且 32 位寄存器 (D11, D0) 的内容又小于 678,493 时或 M3=ON 时，M50=ON。



FB/FC	指令码			操作数						功能								
FC	D*	OR※		S ₁ , S ₂						接点类型比较 OR※								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○			
							脉冲执行型			16 位指令				32 位指令				
							-			AH 运动控制主机				AH 运动控制主机				

图形：



S₁ : 数据源 1

S₂ : 数据源 2

以 OR=跟 DOR=为例

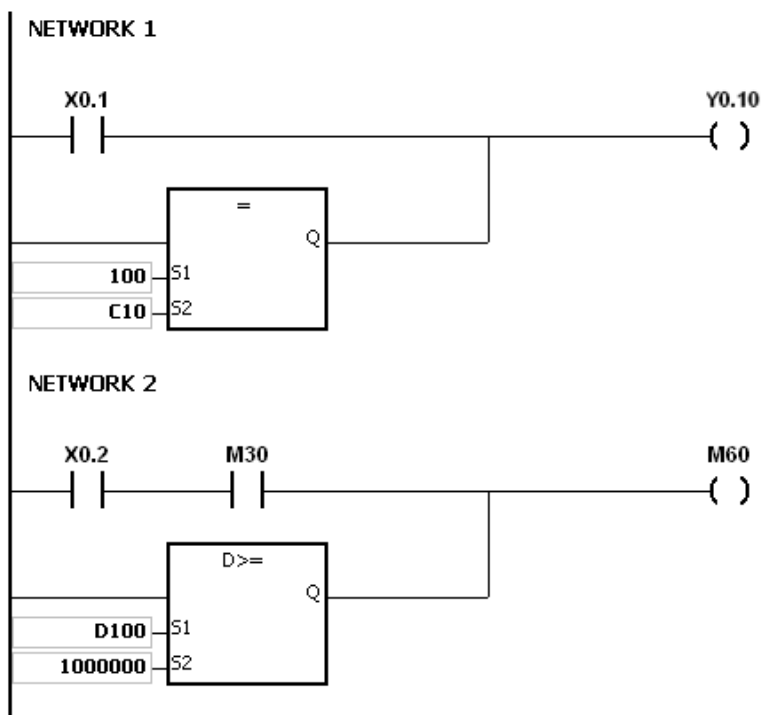
指令说明：

1. S₁与 S₂的内容做比较的指令。以“OR=”做为例子，比较结果为“等于”时，该指令导通；“不等于”时，该指令不导通。
2. 32 位指令才可以使用 HC 装置。

16 位指令	32 位指令	导通条件	非导通条件
OR =	DOR =	S ₁ = S ₂	S ₁ ≠ S ₂
OR < >	DOR < >	S ₁ ≠ S ₂	S ₁ = S ₂
OR >	DOR >	S ₁ > S ₂	S ₁ ≤ S ₂
OR > =	DOR > =	S ₁ ≥ S ₂	S ₁ < S ₂
OR <	DOR <	S ₁ < S ₂	S ₁ ≥ S ₂
OR < =	DOR < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

1. 当 X0.1=ON 时，或者是 C10 的当前值等于 100 时，Y0.10=ON。
2. 当 X0.2 及 M30 都等于 ON 时，或者是 32 位寄存器 (D101, D100) 的内容大于或等于 1,000,000 时，M60=ON。



3

FB/FC	指令码				操作数						功能						
FC	D*	FLD※			S ₁ , S ₂						浮点数接点类型比较 LD※						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂										●	●*						
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
脉冲执行型				32 位指令				64 位指令									
-				AH 运动控制主机				AH 运动控制主机									

图形：



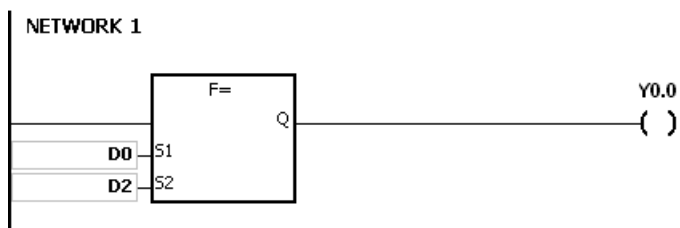
指令说明：

S₁与 S₂的内容以浮点数类型做比较的指令。以“FLD=”做为例子，比较结果为“等于”时，该指令导通；“不等于”时，该指令不导通。

32 位指令	64 位指令	导通条件	非导通条件
FLD =	DFLD =	S ₁ = S ₂	S ₁ ≠ S ₂
FLD < >	DFLD < >	S ₁ ≠ S ₂	S ₁ = S ₂
FLD >	DFLD >	S ₁ > S ₂	S ₁ ≤ S ₂
FLD > =	DFLD > =	S ₁ ≥ S ₂	S ₁ < S ₂
FLD <	DFLD <	S ₁ < S ₂	S ₁ ≥ S ₂
FLD < =	DFLD < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

以单精度浮点数指令为例，D0 的值等于 D2 的值时，Y0.0 ON。

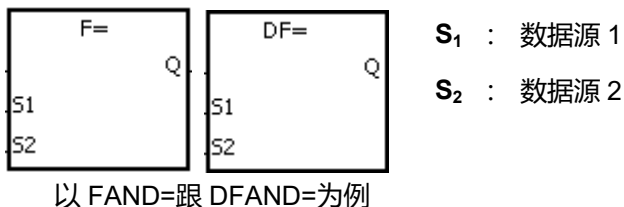


补充说明：

当 S_1 或 S_2 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码				操作数						功能							
FC	D*	FAND※			S ₁ , S ₂						浮点数接点类型比较 AND※							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂										●	●*							
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○	
脉冲执行型						32 位指令						64 位指令						
-						AH 运动控制主机						AH 运动控制主机						

图形：



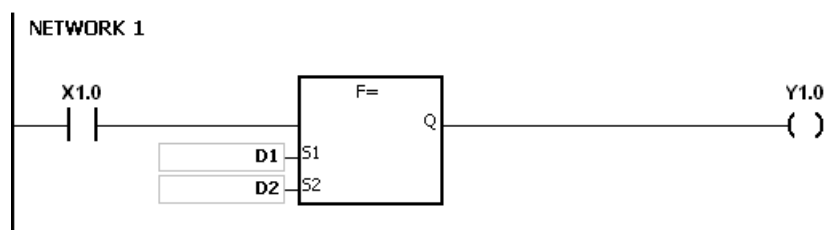
指令说明：

S₁与 S₂的内容以浮点数类型做比较的指令。以“FAND=”做为例子，比较结果为“等于”时，该指令导通；“不等于”时，该指令不导通。

32 位指令	64 位指令	导通条件	非导通条件
FAND =	DFAND =	S ₁ = S ₂	S ₁ ≠ S ₂
FAND < >	DFAND < >	S ₁ ≠ S ₂	S ₁ = S ₂
FAND >	DFAND >	S ₁ > S ₂	S ₁ ≤ S ₂
FAND > =	DFAND > =	S ₁ ≥ S ₂	S ₁ < S ₂
FAND <	DFAND <	S ₁ < S ₂	S ₁ ≥ S ₂
FAND < =	DFAND < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

以单精度浮点数指令为例，当 X1.0=ON 且 D1 的值等于 D2 的值时，Y1.0 ON。

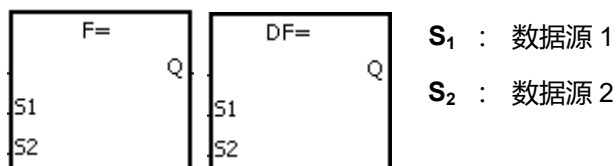


补充说明：

当 S_1 或 S_2 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码				操作数						功能						
FC	D*	FOR※			S ₁ , S ₂						浮点数接点类型比较 OR※						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂										●	●*						
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
						脉冲执行型			32 位指令			64 位指令					
						-			AH 运动控制主机			AH 运动控制主机					

图形：



以 FOR=跟 DFOR=为例

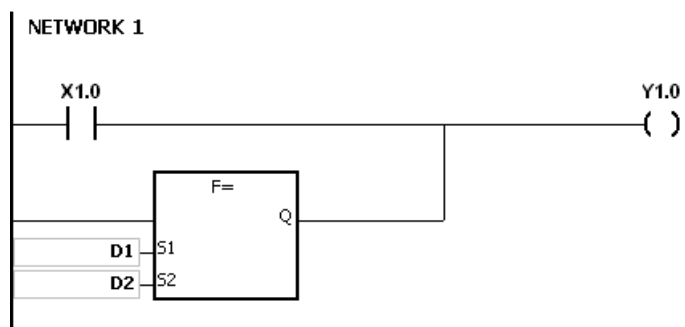
指令说明：

S₁与 S₂的内容以浮点数类型做比较的指令。以“FOR=”做为例子，比较结果为“等于”时，该指令导通；“不等于”时，该指令不导通。

32 位指令	64 位指令	导通条件	非导通条件
FOR =	DFOR =	S ₁ = S ₂	S ₁ ≠ S ₂
FOR < >	DFOR < >	S ₁ ≠ S ₂	S ₁ = S ₂
FOR >	DFOR >	S ₁ > S ₂	S ₁ ≤ S ₂
FOR > =	DFOR > =	S ₁ ≥ S ₂	S ₁ < S ₂
FOR <	DFOR <	S ₁ < S ₂	S ₁ ≥ S ₂
FOR < =	DFOR < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

X1.0=ON 或者 D1 的值等于 D2 时，Y1.0=ON。

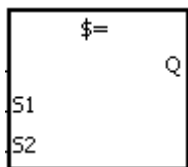


补充说明：

当 S_1 或 S_2 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码				操作数						功能							
FC		LD\$※			S ₁ , S ₂						字符串类型比较 LD\$※							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂														●				
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●			●	●		●	●		●	○	●			○		
				脉冲执行型				16 位指令				32 位指令						
				-				AH 运动控制主机				-						

图形：



以 LD\$=为例

S₁ : 数据源 1

S₂ : 数据源 2

指令说明：

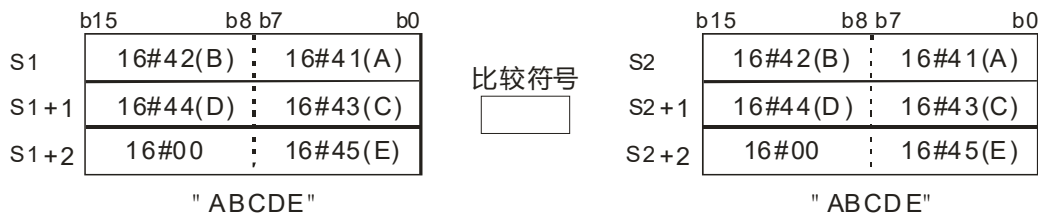
- S₁ 与 S₂ 的内容以字符串类型做比较的指令。以“LD\$=”做为例子，比较结果为“等于”时，该接点导通；“不等于”时，该接点不导通。

指令	导通条件	非导通条件
LD\$ =	S ₁ = S ₂	S ₁ ≠ S ₂
LD\$ < >	S ₁ ≠ S ₂	S ₁ = S ₂
LD\$ >	S ₁ > S ₂	S ₁ ≤ S ₂
LD\$ > =	S ₁ ≥ S ₂	S ₁ < S ₂
LD\$ <	S ₁ < S ₂	S ₁ ≥ S ₂
LD\$ < =	S ₁ ≤ S ₂	S ₁ > S ₂

- S~S+n (这里的 n 表示第 n 个装置，没有限制) 中有包含 16#00 时，才能判断为一个完整字符串，例如：

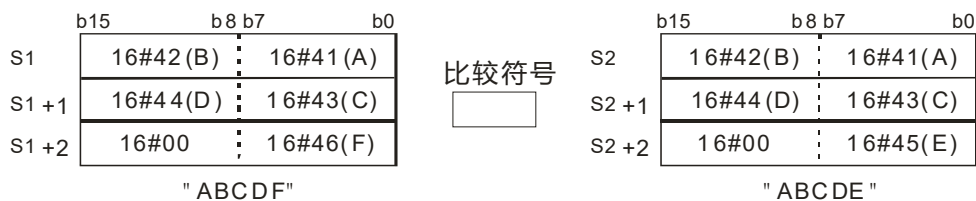
	b15	b8	b7	b0
S	16#32(2)		16#31(1)	
S+1	16#34(4)		16#33(3)	
S+2	16#00		16#35(5)	
" 12345"				

- 字符串完全相同时，如下范例 S₁ 与 S₂ 的字符串都是“ABCDE”，各指令执行结果如下表所列：



比较符号	比较结果
\$ =	导通
\$ < >	不导通
\$ >	不导通
\$ > =	导通
\$ <	不导通
\$ < =	导通

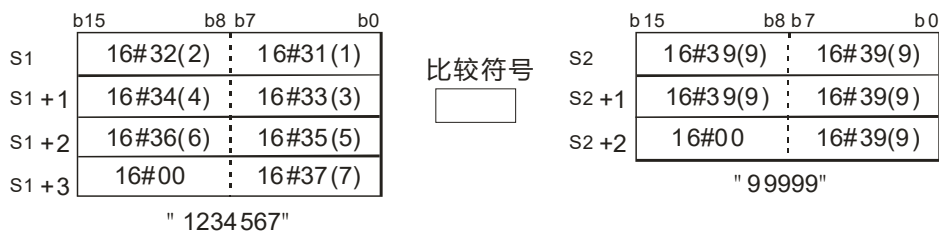
4. 字符串长度相同，但内容不相同，以遇到的第一个不相同字的字符值（ASCII 值）来比较哪个字符串较大；如下范例 S_1 的字符串是“ABCDF”而 S_2 的字符串是“ABCDE”，彼此遇到的第一个不同字为“F”（16#46）跟“E”（16#45），因为 16#46>16#45，所以认定 S_1 的字符串> S_2 的字符串，各指令执行结果如下表所列：



如上范例

比较符号	比较结果
\$ =	不导通
\$ < >	导通
\$ >	导通
\$ > =	导通
\$ <	不导通
\$ < =	不导通

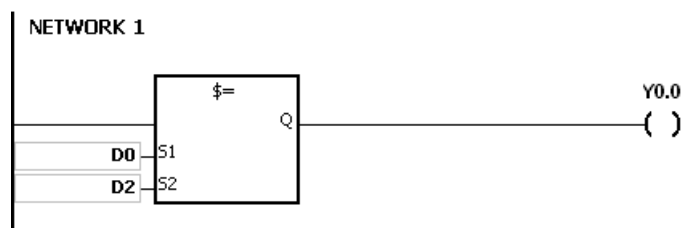
5. 字符串长度不相等时，以字符串长度较长的为大；如下范例 S_1 的字符串是“1234567”而 S_2 的字符串是“99999”，因为 S_1 的字符串长度是 7 而 S_2 的字符串是 5，所以认定 S_1 的字符串> S_2 的字符串，各指令执行结果如下表所列：



比较符号	比较结果
\$ =	不导通
\$ < >	导通
\$ >	导通
\$ > =	导通
\$ <	不导通
\$ < =	不导通

程序范例：

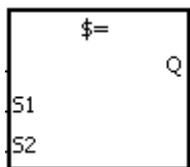
D0~16#00 的字符串等于 D2~16#00 的字符串时，Y0.0 ON。

**补充说明：**

字符串的字尾无 16#00 当做结束符号，指令不执行，SM0=ON，错误码 SR0=16#200E。

FB/FC	指令码			操作数							功能						
FC		AND\$※		S₁, S₂							字符串类型比较 AND\$※						
数据类型				BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S₁, S₂																	●
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S₁, S₂	●	●			●	●		●			●	○	●			○	
脉冲执行型							16 位指令					32 位指令					
-							AH 运动控制主机										

3 图形：



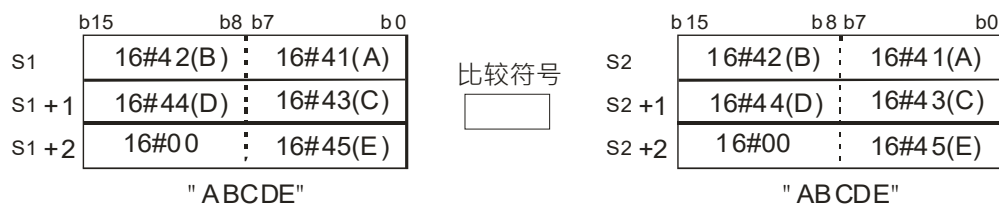
S₁ : 数据源 1

S₂ : 数据源 2

以 AND\$=为例

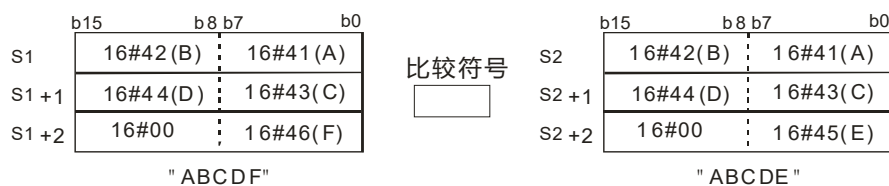
指令说明：

1. S₁ 与 S₂ 的内容以字符串类型做比较的指令。以“AND\$=”做为例子，比较结果为“等于”时，该接点导通，“不等于”时，该接点不导通。
2. S 一直到 S+n (这里的 n 表示第 n 个装置，没有限制) 中有包含 16#00 时，才能当做一个完整的字符串。
3. 字符串完全相同时，如下范例 S₁ 与 S₂ 的字符串都是“ABCDE”，各指令执行结果如下表所列：



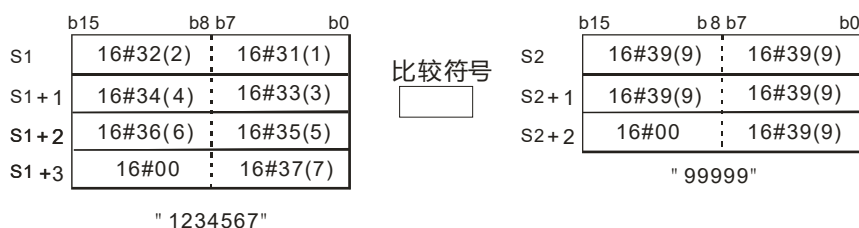
比较符号	比较结果
\$ =	导通
\$ < >	不导通
\$ >	不导通
\$ > =	导通
\$ <	不导通
\$ < =	导通

4. 字符串长度相同，但内容不相同时，以遇到的第一个不相同字的字符值 (ASCII 值) 来比较哪个字符串较大；如下范例 S_1 的字符串是“ABCDF”而 S_2 的字符串是“ABCDE”，彼此遇到的第一个不同字为“F”(16#46)跟“E”(16#45)，因为 $16\#46 > 16\#45$ ，所以认定 S_1 的字符串 $>$ S_2 的字符串，各指令执行结果如下表所列：



比较符号	比较结果
\$ =	不导通
\$ < >	导通
\$ >	导通
\$ > =	导通
\$ <	不导通
\$ < =	不导通

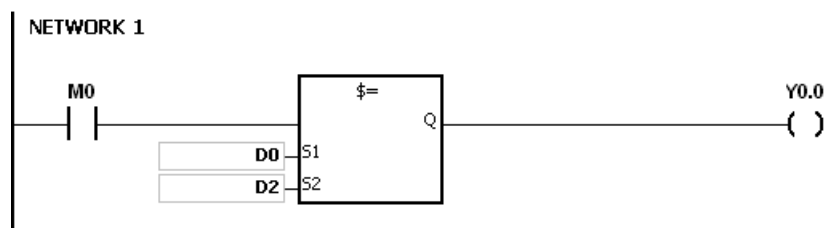
5. 字符串长度不相等时，以字符串长度较长的为大；如下范例 S_1 的字符串是“1234567”而 S_2 的字符串是“99999”，因为 S_1 的字符串长度是 7 而 S_2 的字符串是 5，所以认定 S_1 的字符串 $>$ S_2 的字符串，各指令执行结果如下表所列：



比较符号	比较结果
\$ =	不导通
\$ < >	导通
\$ >	导通
\$ > =	导通
\$ <	不导通
\$ < =	不导通

程序范例：

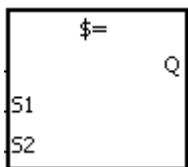
当 $M0=ON$ 且 $D0\sim 16\#00$ 的字符串等于 $D2\sim 16\#00$ 的字符串，则 $Y0.0 ON$ 。



FB/FC	指令码			操作数						功能							
FC		OR\$※		S₁, S₂						字符串类型比较 OR\$※							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂														●			
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●		●	●		●	○	●			○	
脉冲执行型						16 位指令						32 位指令					
-						AH 运动控制主机						-					

3

图形：



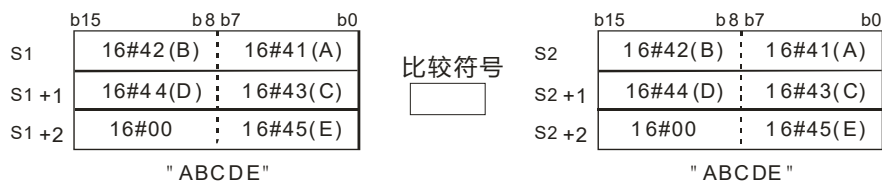
S₁ : 数据源 1

S₂ : 数据源 2

以 OR\$=为例

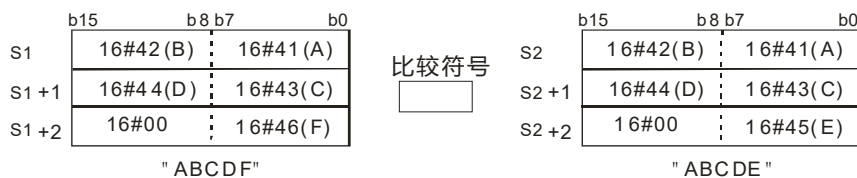
指令说明：

- S₁ 与 S₂ 的内容以字符串类型做比较的指令。以“OR\$=”做为例子，比较结果为“等于”时，该接点导通；“不等于”时，该接点不导通。
- S 一直到 S+n (这里的 n 表示第 n 个装置，没有限制) 中有包含 16#00 时，才能当做一个完整的字符串。
- 字符串完全相同时，如下范例 S₁ 与 S₂ 的字符串都是“ABCDE”，各指令执行结果如下表所列：



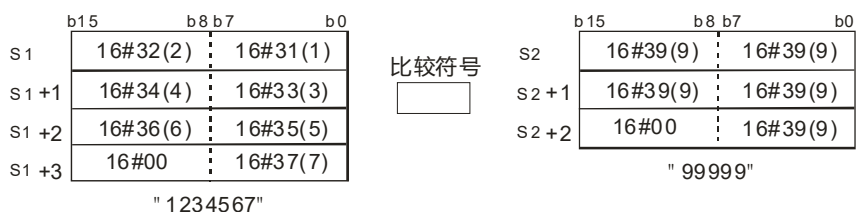
比较符号	比较结果
\$ =	导通
\$ < >	不导通
\$ >	不导通
\$ > =	导通
\$ <	不导通
\$ < =	导通

4. 字符串长度相同，但内容不相同时，以遇到的第一个不相同字的字符值（ASCII 值）来比较哪个字符串较大；如下范例 S_1 的字符串是“ABCDF”而 S_2 的字符串是“ABCDE”，彼此遇到的第一个不同字为“F”（16#46）跟“E”（16#45），因为 $16\#46 > 16\#45$ ，所以认定 S_1 的字符串 $>$ S_2 的字符串，各指令执行结果如下表所列：



比较符号	比较结果
\$ =	不导通
\$ < >	导通
\$ >	导通
\$ > =	导通
\$ <	不导通
\$ < =	不导通

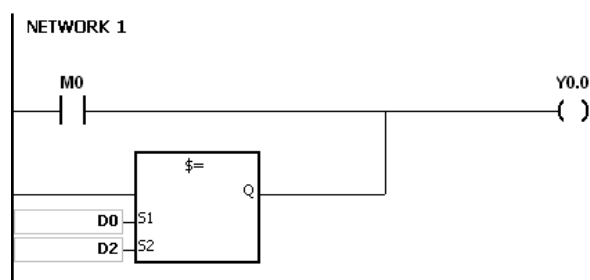
5. 字符串长度不相等时，以字符串长度较长的为大；如下范例 S_1 的字符串是“1234567”而 S_2 的字符串是“99999”，因为 S_1 的字符串长度是 7 而 S_2 的字符串是 5，所以认定 S_1 的字符串 $>$ S_2 的字符串，各指令执行结果如下表所列：



比较符号	比较结果
\$ =	不导通
\$ < >	导通
\$ >	导通
\$ > =	导通
\$ <	不导通
\$ < =	不导通

程序范例：

当 $M0=ON$ 或者是 $D0\sim 16\#00$ 的字符串等于 $D2\sim 16\#00$ 的字符串时， $Y0.0$ ON。



FB/FC	指令码			操作数				功能						
FC	D*	CMP	P	S ₁ , S ₂ , D				比较设定输出						

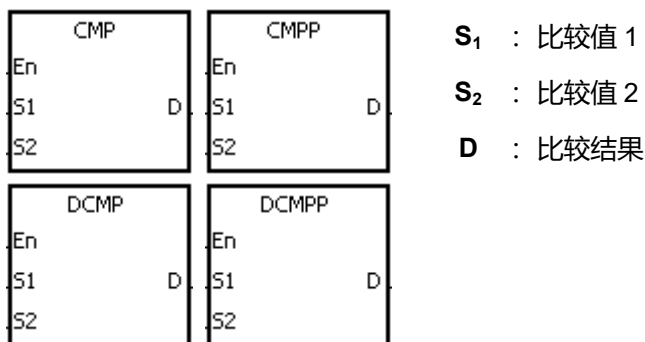
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D (array , 3)	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：

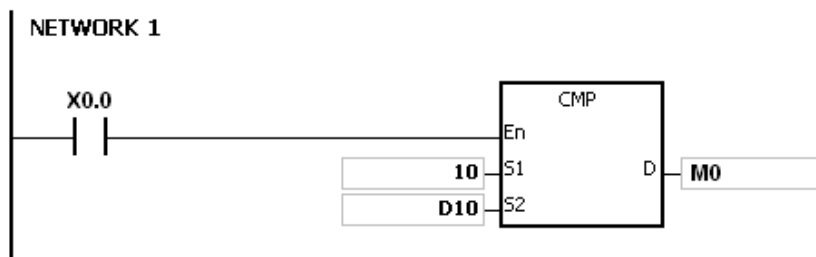


指令说明：

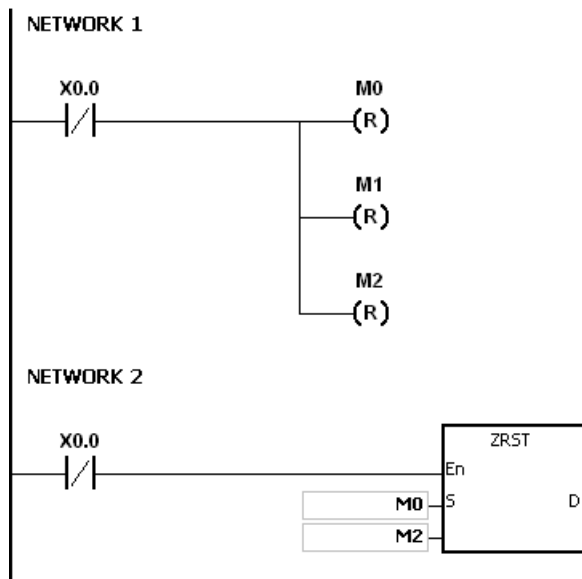
1. 将操作数 S₁ 和 S₂ 的内容以有号数十进制作大小比较，其比较结果在 D 作表示。
2. 操作数 D 占用 3 个连续装置。D, D+1, D+2 用于储存比较结果，如果 S₁ > S₂, D=ON；如果 S₁= S₂, D+1=ON；如果 S₁ < S₂, D+2=ON。
3. DCMP、DCMPP 指令才可使用 32 位计数器。

程序范例：

1. 如果操作数 D 设为 M0，那么比较结果会显示在 M0, M1, M2 中，如下所示。
2. 当 X0.0=ON, CMP 指令执行，M0, M1, M2 其中的某个装置会 ON。当 X0.0=ON, CMP 指令停止执行并且 M0, M1, M2 保持它们的当前值。



若要清除其比较结果请使用 RST 或 ZRST 指令。



补充说明：

1. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of BOOL。
2. D+2 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数				功能						
FC	D*	ZCP	P	S ₁ , S ₂ , S, D				区域比较						

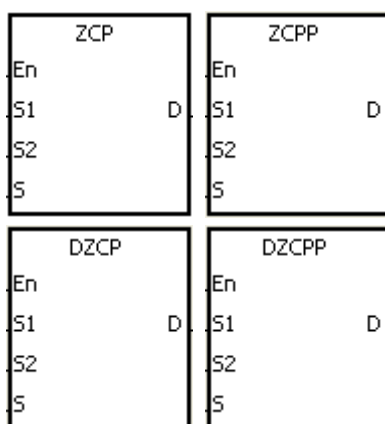
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂ , S		●	●*				●	●*						
D (array, 3)	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂ , S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



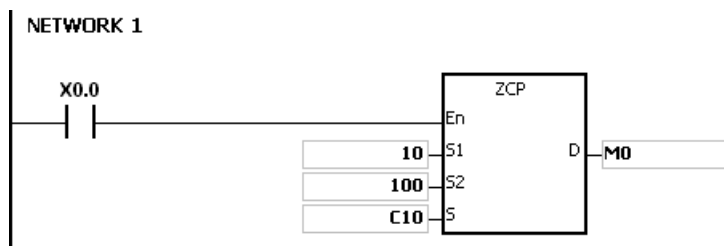
S₁ : 区间比较的下限值
 S₂ : 区间比较的上限值
 S : 比较值
 D : 比较结果

指令说明：

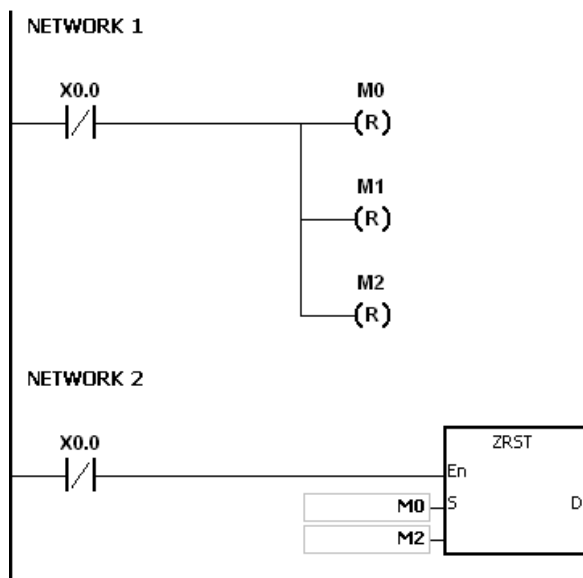
1. 比较值 S 与下限 S₁ 及上限 S₂ 以有号数十进制数值做比较，其比较结果在 D 作表示。
2. 操作数 S₁ 必须比 S₂ 小。当 S₁>S₂，指令执行时把 S₁ 作为上/下限值进行比较。
3. 操作数 D 占用 3 个连续的装置，D，D+1，D+2 储存比较结果。如果 S₁>S，D=ON；如果 S₁≤S≤S₂，D+1=ON；如果 S₂<S，D+2=ON。
4. DZCP、DZCPP 指令才可使用 HC 装置。

程序范例：

1. 比较结果指定装置为 M0，则自动占有 M0，M1 及 M2。
2. 当 X0.0=ON 时，ZCP 指令执行，M0，M1 及 M2 其中之一会 ON，当 X0.0=OFF 时，ZCP 指令不执行，M0，M1 及 M2 状态保持在 X0.0=OFF 之前的状态。



若要清除其结果请使用 RST 或 ZRST 指令。



补充说明：

1. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of BOOL。
2. D+2 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数						功能				
FC		FCMP	P	S₁ , S₂ , D						浮点数比较				

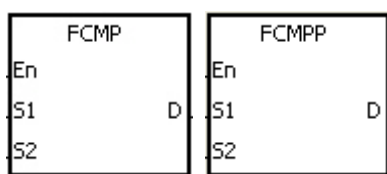
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂										●				
D (array , 3)	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●	●	●				●	●	●			●				

脉冲执行型	32 位指令	64 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



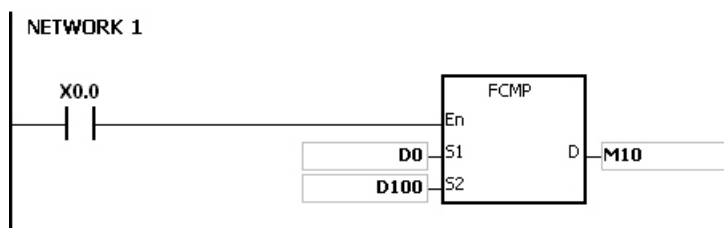
S₁ : 浮点数比较值 1
 S₂ : 浮点数比较值 2
 D : 比较结果

指令说明：

1. 浮点数比较值 S₁与浮点数比较值 S₂进行比较，比较的结果 (> , = , <) 在 D 作表示。
2. 操作数 D 占用 3 个连续装置。D , D+1 , D+2 用于储存比较结果 如果 S₁>S₂ , D=ON 如果 S₁=S₂ , D+1=ON ; 如果 S₁<S₂ , D+2=ON。

程序范例：

1. 若比较结果指定装置为 M10 则自动占有 M10~M12。
2. 当 X0.0=ON 时 , FCMP 指令执行 , M10~M12 其中之一会 ON。当 X0.0=OFF , FCMP 指令不执行 , M10~M12 状态保持在 X0.0=OFF 之前的状态。
3. 若需得到 ≥ , ≤ , # 的结果时 , 可将 M10~M12 串并联即可取得。
4. 若要清除其结果请使用 RST 或 ZRST 指令。



补充说明：

1. 当 S_1 或 S_2 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。
2. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of BOOL。
3. $D+2$ 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数						功能					
FC		FZCP	P	S₁, S₂, S, D						浮点数区域比较					

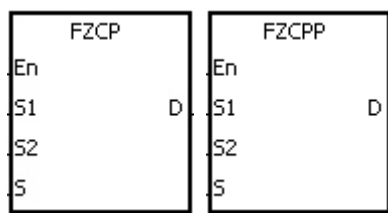
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂ , S										●				
D (array, 3)	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂ , S	●	●			●	●	●	●	●		●	○	●				○
D	○	●	●	●				●	●	●			●				

脉冲执行型	32 位指令	64 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



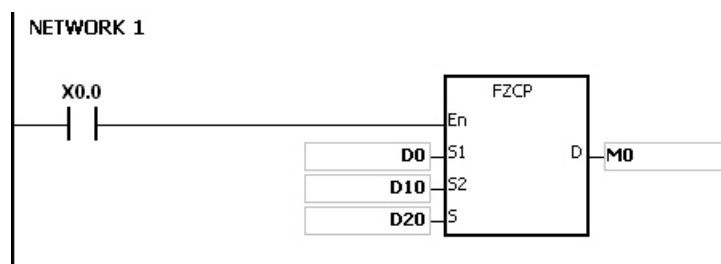
- S₁ : 浮点数区域比较
- S₂ : 区域比较之浮点数上限值
- S : 浮点数比较值
- D : 比较结果

指令说明：

1. 浮点数比较值 S 与浮点数下限值 S₁ 及浮点数上限值 S₂ 做比较，其比较结果在 D 做表示。
2. 操作数 S₁ 必须比 S₂ 小。当 S₁>S₂，指令执行时把 S₁ 作为上/下限值进行比较。
3. 操作数 D 占用 3 个连续的装置，D，D+1，D+2 储存比较结果。如果 S₁>S，D=ON；如果 S₁≤S≤S₂，D+1=ON；如果 S₂<S，D+2=ON。

程序范例：

1. 若比较结果指定装置为 M0，则自动占有 M0~M2。
2. 当 X0.0=ON 时，FZCP 指令执行，M0~M2 其中之一会 ON。当 X0.0=OFF，FZCP 指令不执行，M0~M2 状态保持在 X0.0=OFF 之前的状态。
3. 若要清除其结果请使用 RST 或 ZRST 指令。

**补充说明：**

1. 当 S_1 或 S_2 或 S 的内容值超出浮点数可以表示的范围时,接点不导通,SM0=ON,错误码 SR0=16#2013。
2. D 操作数,若使用 ISPSOft 宣告,则数据类型为 ARRAY [3] of BOOL。
3. $D+2$ 超过装置范围时,指令不执行,SM0=ON,错误码 SR0=16#2003。

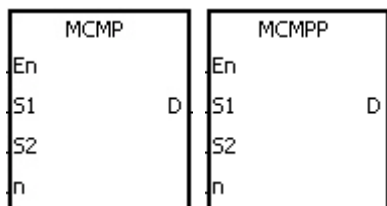
FB/FC	指令码		操作数				功能						
FC	MCMP	P	S_1, S_2, n, D				矩阵比较						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●		●	○	○		
D	●	●			●	●		●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



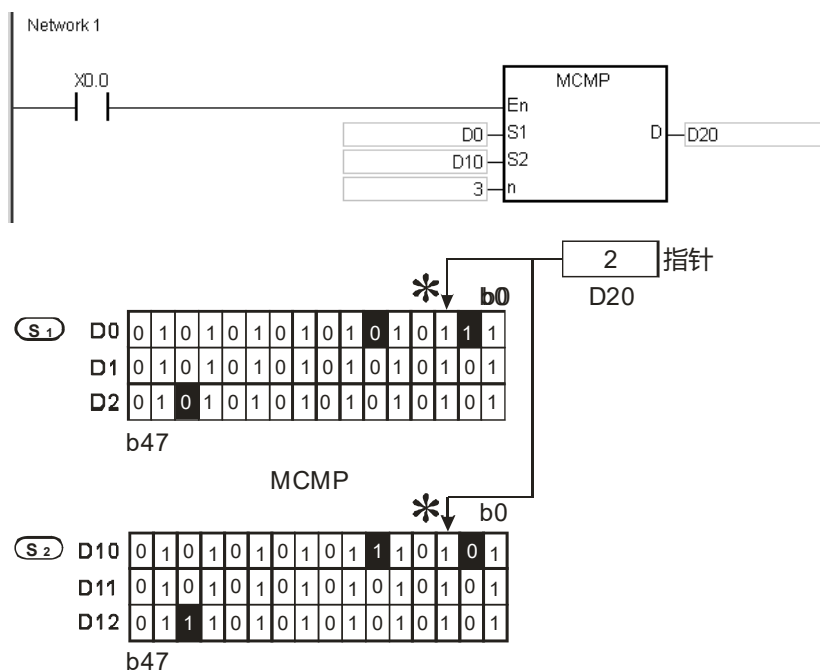
- S_1 : 矩阵来源装置 1
- S_2 : 矩阵来源装置 2
- n : 数组长度
- D : 指针 Pr 用以存放目标的地址值

指令说明：

1. 每次比较依两个矩阵来源指针 Pr 地址。将 S_1 或 S_2 两个矩阵中的每一个 bit 从地址 D+1 开始做比较，找出值不同的地址，再将此地址值存到 D 中，完成此次比较。
2. n 操作数的范围为 1~256。
3. 由矩阵比较标志 SM607 决定比较相同值 (SM607=1) 或不同值 (SM607=0)，当比较到达时立即停止比较动作，矩阵位寻找标志 SM610=1。当比较到最后一个 bit 时，矩阵搜寻结束标志 SM608=ON 比较到达的编号存于 D 中，下一次扫描周期时，再由第 0 个 bit 开始比较，同时矩阵搜寻起始标志 SM609=1。当 D 的值超过范围时指针错误标志 SM611=1。
4. 在矩阵指令运作中，通常需要有一个 16 位寄存器来指定矩阵中 16n 个单点的某个单点当作运算对象。此寄存器称为矩阵的指针 Pr (Pointer)，由用户于指令中指定，其有效范围为 0~16n-1，分别对应至矩阵中的位 b0~b16n-1。在运作中应避免更动到 Pr 值，以免影响其正确的比较找寻，若 Pr 值超出此范围则矩阵指针错误标志 SM611 设为 1，且本指令不执行。
5. 若矩阵搜寻结束标志 SM608 与矩阵位寻找标志 SM610 同时发生则会同时=1。

程序范例：

- 当 X0.0 由 OFF→ON，矩阵搜寻起始标志 SM609=0，故由指针当时值加 1 的 bit 地址（标注 * 处）开始往下比较找寻位状态不同（SM607=0 为找不同）者。
- 设指针当时值 D20=2，当 X0.0 由 OFF→ON 时动作 4 次，可得到如（①，②，③，④）四个执行结果。
 - D20=5，矩阵位寻找标志 SM610=1，矩阵搜寻结束标志 SM608=0。
 - D20=45，矩阵位寻找标志 SM610=1，矩阵搜寻结束标志 SM608=0。
 - D20=47，矩阵位寻找标志 SM610=0，矩阵搜寻结束标志 SM608=1。
 - D20=1，矩阵位寻找标志 SM610=1，矩阵搜寻结束标志 SM608=0。



补充说明：

- 运算错误码说明：

当 S_1+n-1 或 S_2+n-1 的装置超过范围时，该指令不执行，SM0=ON，错误码 SR0=16#2003。

若 n 操作数的值不在 1~256 之间，该指令不执行，且 SM0=ON，错误码 SR0=16#200B。

- 标志信号说明：

SM607：	矩阵比较标志，比较相同值 SM607=1，比较不同值 SM607=0
SM608：	矩阵搜寻结束标志，当比较到最后一个 bit 时，SM608=1
SM609：	矩阵搜寻起始标志，由第一个 bit 开始比较，SM609=1
SM610：	矩阵位寻找标志，比较到达时立即停止比较动作 SM610=1
SM611：	矩阵指针错误标志，指针 Pr 值超出此范围则 SM611=1

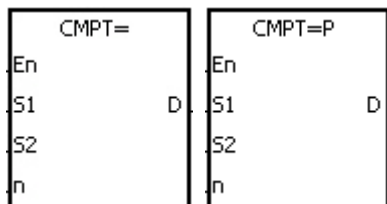
FB/FC	指令码			操作数					功能					
FC		CMPT※	P	S ₁ , S ₂ , n, D					表格比较指令					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							
n		●					●							
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●		●	○	●	○	○		○
S ₂	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

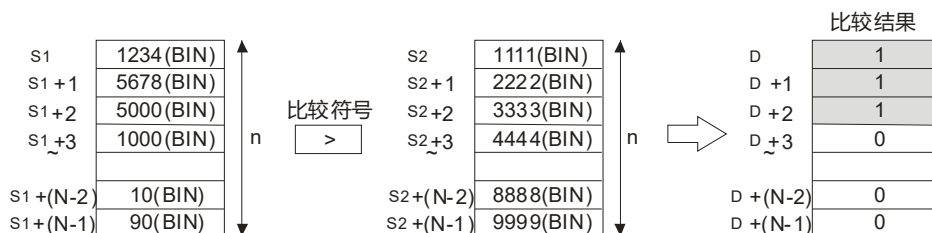
图形：



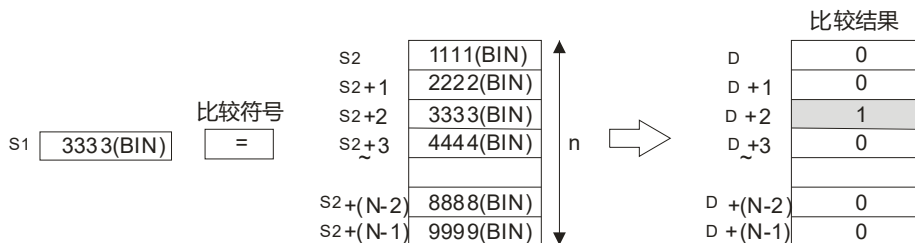
- S₁ : 来源起始装置 1
- S₂ : 来源起始装置 2
- n : 比较数据长度
- D : 目标装置。

指令说明：

- S₁ 与 S₂ 的内容以有号数十进制做 n 笔比较，将结果存放于 D。
- n 操作数的范围为 1~256。
- D 操作数写入值都以 1 位写入，未被对应之 bit 值不变。
- 表格比较指令 CMPT#全部输出标志，若比较结果为全都输出（全都 ON），则 SM620=ON，反之 SM620=OFF。
- 如果 S₁ 的来源指定为装置，则比较方式如下：



6. 如果 S_1 的来源直接指定为 -32768~32767，则比较方式如下：

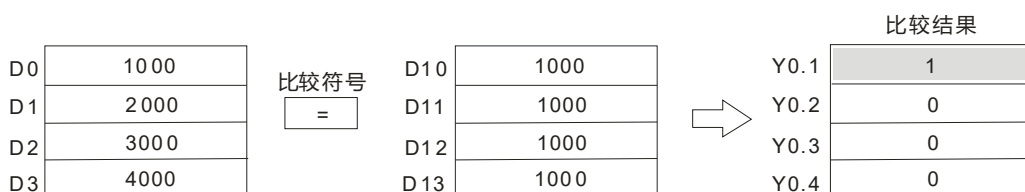
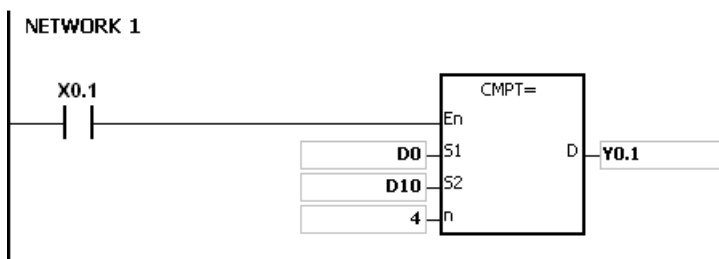


7. 各指令比较结果如下：

16 位指令	比较结果	
	1	0
CMPT =	$S_1 = S_2$	$S_1 \neq S_2$
CMPT < >	$S_1 \neq S_2$	$S_1 = S_2$
CMPT >	$S_1 > S_2$	$S_1 \leq S_2$
CMPT > =	$S_1 \geq S_2$	$S_1 < S_2$
CMPT <	$S_1 < S_2$	$S_1 \geq S_2$
CMPT < =	$S_1 \leq S_2$	$S_1 > S_2$

程序范例：

D0~D3 与 D10~D13 数据比对，若比较结果相同，对应的 Y0.1~Y0.4=ON。



补充说明：

1. 若 n 不在 1~256 之间，指令不执行，SM0=ON，SR0 错误码=16#200B。
2. $S_1 \sim S_1+n$ 与 $S_2 \sim S_2+n$ 与 D 操作数的装置如果不足，则指令不执行，SM0=ON，错误码 SR0=16#2003。

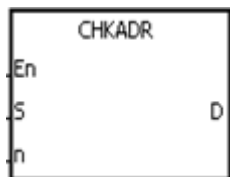
FB/FC	指令码			操作数				功能						
FC	CHKADR			S, n, D				接点类型指针寄存器地址检查						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														
n		●					●							
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S													●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



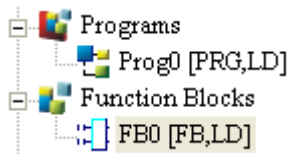
S : 指针寄存器
n : 装置个数
D : 检查结果

指令说明：

1. 检查 S 指针寄存器的内容值~ (S 的内容值+n-1) 是否超过 S 指针寄存器所指定的装置类别范围，检查结果没有超出装置范围时，D 装置为 ON，否则 D 装置为 OFF。
2. S 支持 PR、TR、CR、HCR 指针寄存器。
3. n 的范围 1~1024。
4. 本指令只可以在功能块中使用。

程序范例：

1. 使用 ISPSOft 建立一个 Program，及一个 Function Block。



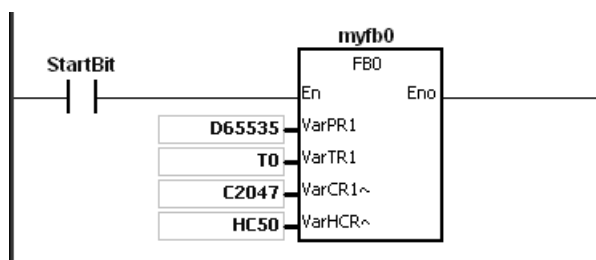
并且在 Program 中宣告两个变量如下：

Local Symbols				
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR	myfb0	N/A [Auto]	FB0	N/A
VAR	StartBit	N/A [Auto]	BOOL	FALSE

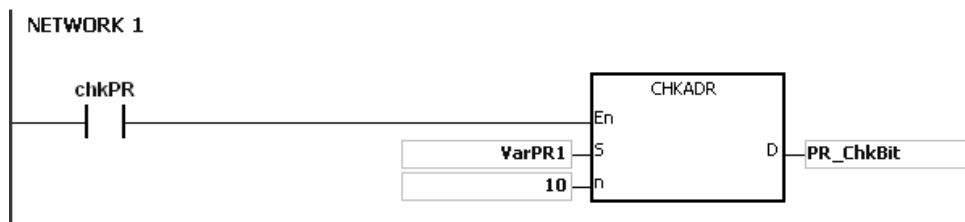
2. 在已建立的 Function Block 中宣告 VarPR1、VarTR1、VarCR1 以及 VarHCR1，Type 分别指定为 POINTER、T_POINTER、C_POINTER、HC_POINTER。

Local Symbols				
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR_IN_OUT	VarPR1	N/A [Auto]	POINTER	N/A
VAR_IN_OUT	VarTR1	N/A [Auto]	T_POINTER	N/A
VAR_IN_OUT	VarCR1	N/A [Auto]	C_POINTER	N/A
VAR_IN_OUT	VarHCR1	N/A [Auto]	HC_POINTER	N/A
VAR	PR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	TR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	CR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	HCR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	chkPR	N/A [Auto]	BOOL	N/A
VAR	chkTR	N/A [Auto]	BOOL	N/A
VAR	chkCR	N/A [Auto]	BOOL	N/A
VAR	chkHCR	N/A [Auto]	BOOL	N/A

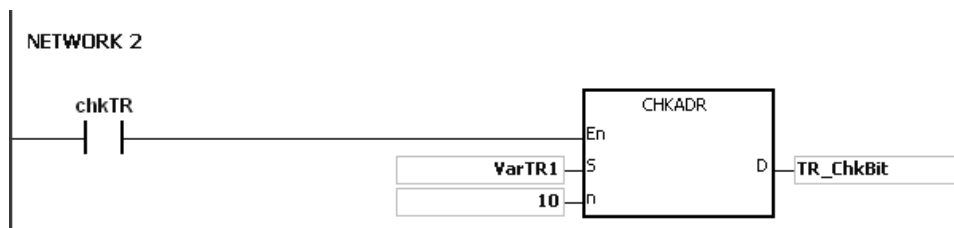
3. 在 Program 中调用已建立好的 Function Block FB0，并且将 D65535、T0、C2047、HC50 指定给 FB0 中的 VarPR1、VarTR1、VarCR1、VarHCR1。



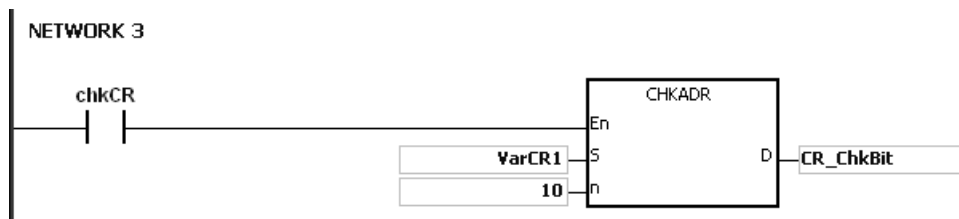
4. 在 FB0 中使用 CHKADR 来检查 VarPR1、VarTR1、VarCR1、VarHCR1 是否超出范围。
 5. 当 chkPR=ON，VarPR1 所代表的实际装置为 D65535，而 D 装置的合法范围为 D0~D65535，所以 $D65535+10-1=D65544$ 已超出范围，PR_ChkBit=OFF。



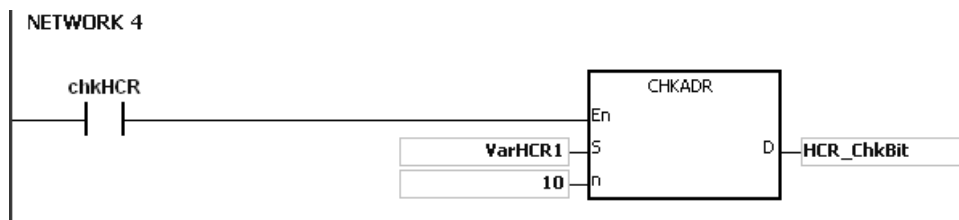
6. 当 chkTR=ON，VarTR1 所代表的实际装置为 T0，而 T 装置的合法范围为 T0~T2047，所以 $T0+10-1=T9$ 没有超出范围，TR_ChkBit=ON。



7. 当 $chkCR=ON$, $VarCR1$ 所代表的实际装置为 C2047 , 而 C 装置的合法范围为 C0~C2047 , 所以 $C2047+10-1=C2056$ 已超出范围 , $CR_ChkBit=OFF$ 。



8. 当 $chkHCR=ON$, $VarHCR1$ 所代表的实际装置为 HC50 , 而 HC 装置的合法范围为 HC0~HC63 , 所以 $HC50+10-1=HC59$ 没有超出范围 , $HCR_ChkBit=ON$ 。



补充说明：

1. **S** 的内容值 (实际装置地址) 如果一开始就不合法 , 超出装置范围时 , 指令不执行 , $SM0=ON$, 错误码 $SR0=16\#2003$ 。
2. 若 **n** 不在 1~1024 之间 , 指令不执行 , $SM0=ON$, 错误码 $SR0=16\#200B$ 。

3.4 四则运算指令

FB/FC	指令码 (位)			P 指令	功能	Step
	16	32	64			
FC	<u>+</u>	<u>D+</u>	-	✓	BIN 加法	7
FC	<u>-</u>	<u>D-</u>	-	✓	BIN 减法	7
FC	<u>*</u>	<u>D*</u>	-	✓	BIN 乘法	7
FC	<u>/</u>	<u>D/</u>	-	✓	BIN 除法	7
FC	-	<u>F+</u>	<u>DF+</u>	✓	浮点数加法	7-9
FC	-	<u>F-</u>	<u>DF-</u>	✓	浮点数减法	7-9
FC	-	<u>F*</u>	<u>DF*</u>	✓	浮点数乘法	7-9
FC	-	<u>F/</u>	<u>DF/</u>	✓	浮点数除法	7-9
FC	<u>B+</u>	<u>DB+</u>	-	✓	BCD 加法	7
FC	<u>B-</u>	<u>DB-</u>	-	✓	BCD 减法	7
FC	<u>B*</u>	<u>DB*</u>	-	✓	BCD 乘法	7
FC	<u>B/</u>	<u>DB/</u>	-	✓	BCD 除法	7
FC	<u>BK+</u>	-	-	✓	连续区块 BIN 加法	9
FC	<u>BK-</u>	-	-	✓	连续区块 BIN 减法	9
FC	<u>\$+</u>	-	-	✓	字符串链结	7-19
FC	<u>INC</u>	<u>DINC</u>	-	✓	BIN 加一	3
FC	<u>DEC</u>	<u>DDEC</u>	-	✓	BIN 减一	3
FC	<u>MUL16</u>	<u>MUL32</u>	-	✓	16/32 位专用 BIN 乘法	7
FC	<u>DIV16</u>	<u>DIV32</u>	-	✓	16/32 位专用 BIN 除法	7

FB/FC	指令码			操作数				功能						
FC	D*	+	P	S ₁ , S ₂ , D				BIN 加法						

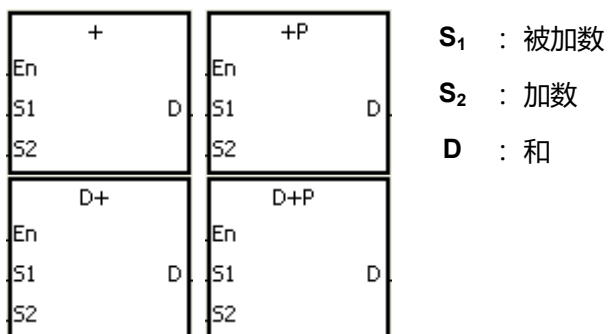
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：

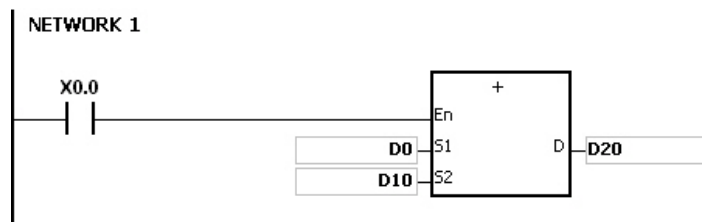


指令说明：

- S₁ 及 S₂ 以 BIN 方式相加的结果存于 D。
- 32 位指令才可以使用 HC 装置。
- 标志：SM600 零标号 (Zero flag)，SM601 借位标号 (Borrow flag)，SM602 进位标号 (Carry flag)。
- 运算结果为 0 时，零标号 (Zero flag) SM600 为 ON，否则为 OFF。
- 16 位 BIN 加法：
运算结果大于 16 位 BIN 可表示范围时，进位标号 (Carry flag) SM602 为 ON，否则为 OFF。
- 32 位 BIN 加法：
运算结果大于 32 位 BIN 可表示范围时，进位标号 (Carry flag) SM602 为 ON，否则为 OFF。

程序范例一：

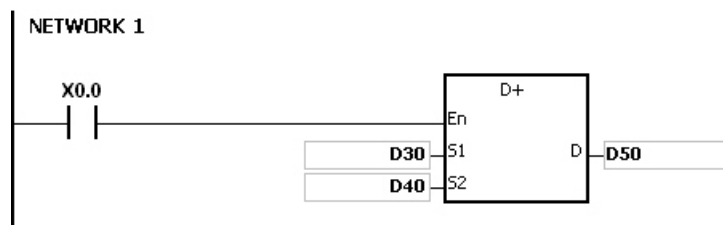
16 位 BIN 加法：当 X0.0=ON 时，被加数 D0 内容加上加数 D10 之内容将结果存在 D20 之内容当中。



- 当 $D0=100$, $D10=10$, $D0+D10=110$, $D20=110$ 。
- 当 $D0=16\#7FFF$, $D10=16\#1$, $D0+D10=16\#8000$, $D20=16\#8000$ 。
- 当 $D0=16\#FFFF$, $D10=16\#1$, $D0+D10=16\#10000$, 此时运算结果超出 16 位 BIN 可表示范围, 则进位标号 $SM602=ON$, $D20=16\#0$, 因为运算结果为 $16\#0$, 所以零标号 $SM600=ON$ 。

程序范例二：

32 位 BIN 加法：当 $X0.0=ON$ 时，被加数（ $D31$ 、 $D30$ ）内容加上加数（ $D41$ 、 $D40$ ）之内容将结果存在（ $D51$ 、 $D50$ ）之中。（其中 $D30$ 、 $D40$ 、 $D50$ 为低 16 位数据， $D31$ 、 $D41$ 、 $D51$ 为高 16 位数据）。



- 当 $(D31、D30)=11111111$, $(D41、D40)=44444444$, $(D31、D30)+(D41、D40)=55555555$, 此时运算结果 , $(D51、D50)=55555555$ 。
- 当 $(D31、D30)=16\#80000000$, $(D41、D40)=16\#FFFFFFFF$, $(D31、D30)+(D41、D40)=16\#17FFFFFFF$, 此时运算结果超出 32 位 BIN 可表示范围, 则进位标号 $SM602=ON$, $(D51、D50)=16\#7FFFFFFF$ 。

补充说明：

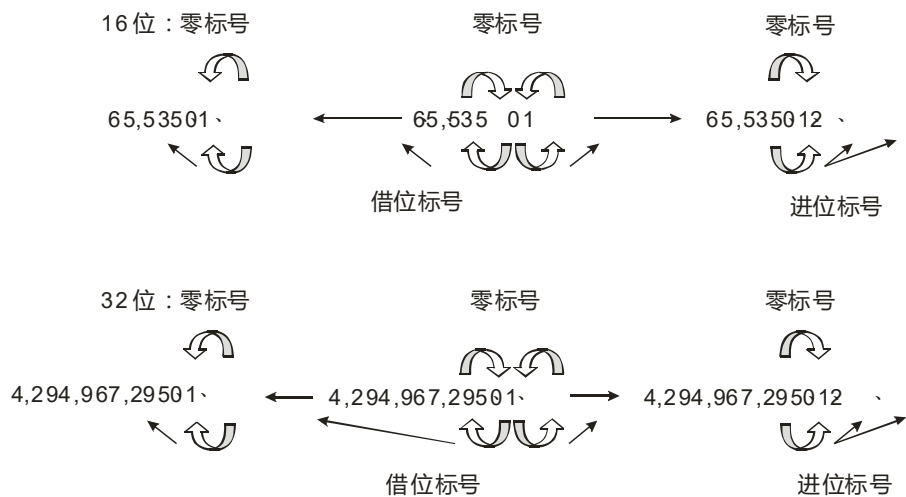
标志操作：

1. 16 位指令：

- 如果操作结果是“0”，零标志 $SM600$ 被设定成 ON。
- 如果操作结果超出 65,535，进位标志 $SM602$ 被设定成 ON。

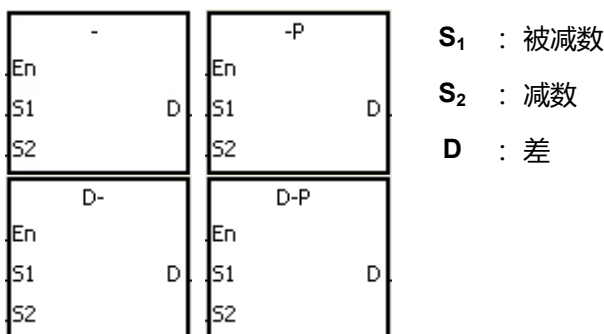
2. 32 位指令：

- 如果操作结果是“0”，零标志 $SM600$ 被设定成 ON。
- 如果操作结果超出 4,294,967,295，进位标志 $SM602$ 被设定成 ON。



FB/FC	指令码			操作数							功能							
FC	D*	-	P	S ₁ , S ₂ , D							BIN 减法							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂		●	●*				●	●*										
D		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●			●	●	●	●			●	○		○	○			
D	●	●			●	●	●	●			●	○						
脉冲执行型							16 位指令					32 位指令						
AH 运动控制主机							AH 运动控制主机					AH 运动控制主机						

图形：

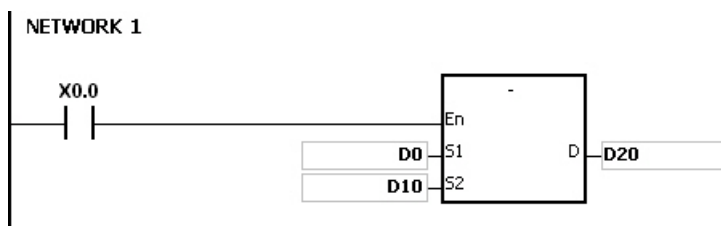


指令说明：

1. S₁ 及 S₂ 以 BIN 方式相减的结果存于 D。
2. 32 位指令才可以使用 HC 装置。
3. 标志：SM600 零标号 (Zero flag)，SM601 借位标号 (Borrow flag)，SM602 进位标号 (Carry flag)。
4. 运算结果为 0 时，零标号 (Zero flag) SM600 为 ON，否则为 OFF。
5. 运算发生借位时，借位标号 (Borrow flag) SM601 为 ON，否则为 OFF。

程序范例一：

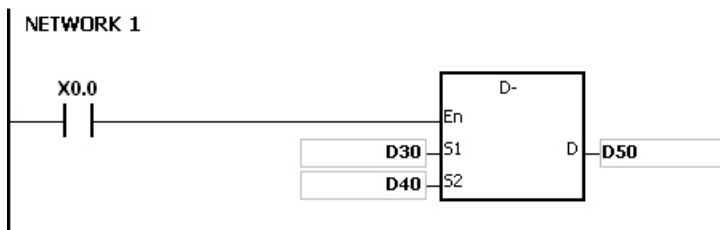
16 位 BIN 减法：当 X0.0=ON 时，将 D0 内容减掉 D10 内容将差存在 D20 之内容中，如下说明：



- 当 $D0=100$, $D10=10$, $D0-D10=90$, $D20=90$ 。
- 当 $D0=16\#8000$, $D10=16\#1$, $D0-D10=16\#7FFF$, $D20=16\#7FFF$ 。
- 当 $D0=16\#1$, $D10=16\#2$, $D0-D10=16\#FFFF$, 运算发生借位 , 借位标号 $SM601=ON$, $D20=16\#FFFF$ 。
- 当 $D0=16\#0$, $D10=16\#FFFF$, $D0-D10=16\#F0001$, 运算发生借位 , 借位标号 $SM601=ON$, $D20=16\#1$ 。

程序范例二：

32 位 BIN 减法：当 $X0.0=ON$ 时，($D31$ 、 $D30$) 内容减掉 ($D41$ 、 $D40$) 的内容将差存在 ($D51$ 、 $D50$) 之中。（其中 $D30$ 、 $D40$ 、 $D50$ 为低 16 位数据， $D31$ 、 $D41$ 、 $D51$ 为高 16 位数据），如下说明：



- 当 ($D31$ 、 $D30$) = 55555555 , ($D41$ 、 $D40$) = 11111111 , ($D31$ 、 $D30$) - ($D41$ 、 $D40$) = 44444444 , ($D51$ 、 $D50$) = 44444444 。
- 当 ($D31$ 、 $D30$) = $16\#80000000$, ($D41$ 、 $D40$) = $16\#FFFFFFFF$, ($D31$ 、 $D30$) - ($D41$ 、 $D40$) = $16\#F80000001$, 运算发生借位 , 借位标号 $SM601=ON$, ($D51$ 、 $D50$) = $16\#80000001$ 。

3

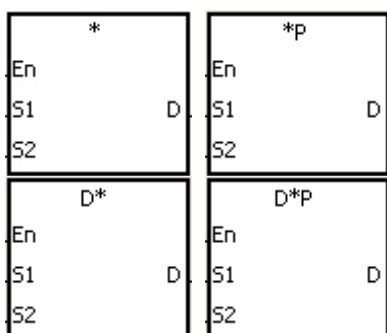
FB/FC	指令码			操作数				功能						
FC	D*	*	P	S ₁ , S ₂ , D				BIN 乘法						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		○
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



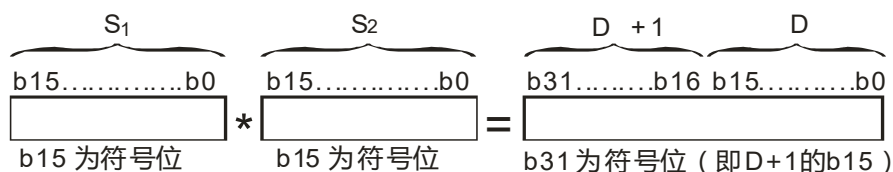
S₁ : 被乘数

S₂ : 乘数

D : 积

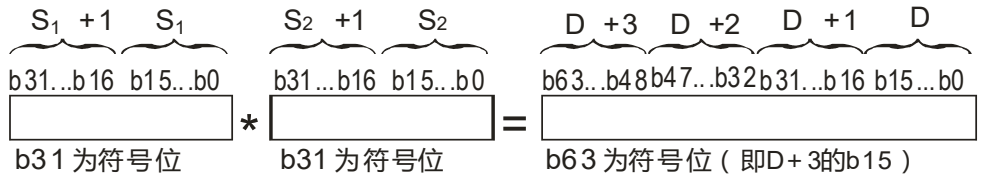
指令说明：

- S₁ 及 S₂ 以有号数二进制方式相乘后的积存于 D。
- D*指令才可以使用 HC 装置。
- 16 位 BIN 乘法运算：



积为 32 位数据，储存在 (D+1, D) 组成的 32 位寄存器中，且符号位 b31=0 为正数，符号位 b31=1 为负数。

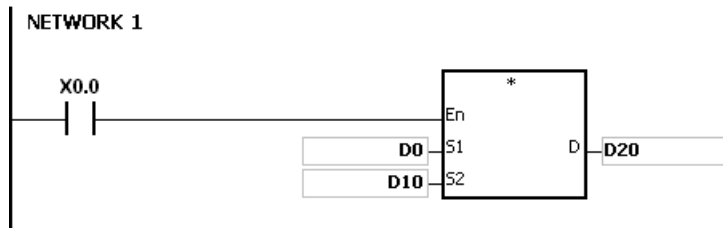
- 32 位 BIN 乘法运算：



积为 64 位数据，储存在 (D+3 , D+2 , D+1 , D) 组成的 64 位寄存器中，且符号位 b63=0 为正数，符号位 b63=1 为负数。

程序范例：

16 位 D0 乘上 16 位 D10 得到一个 32 位的积，结果存在 (D21 , D20)。高 16 位存于 D21，低 16 位存于 D20，结果的正负由最高位 (b31) 的 OFF/ON 来指示。OFF 表示正的 (0)，同时 ON 表示负的 (1)。



D0×D10= (D21 , D20)

16 位×16 位=32 位

3

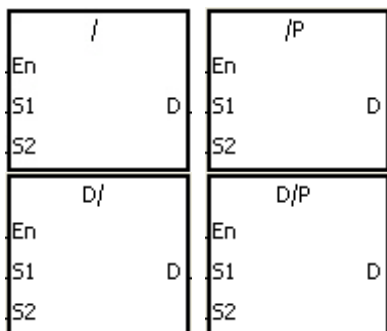
FB/FC	指令码			操作数					功能				
FC	D*	/	P	S ₁ , S ₂ , D					BIN 除法				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D (array , 2)		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

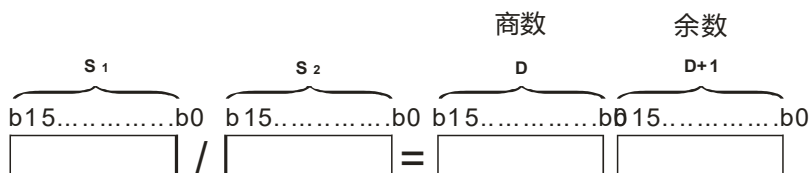
图形：



S₁ : 被除数
 S₂ : 除数
 D : 商及余数

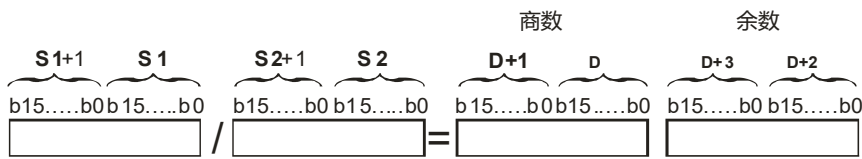
指令说明：

- S₁ 及 S₂ 以有号数二进制方式相除后的商及余数存于 D。
- 32 位指令才可以使用 HC 装置。
- 符号位=0 为正数，符号位=1 为负数。
- 16 位 BIN 除法运算：



D 操作数连续占用两个，D 储存商，D+1 储存余数。

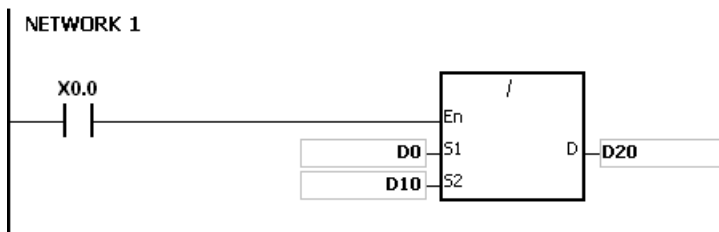
- 32 位 BIN 除法运算：



D 操作数连续占用两个，(D+1, D) 储存商，(D+3, D+2) 储存余数。

程序范例：

当 X0.0=ON 时，被除数 D0 除以除数 D10 而结果商被指定放于 D20，余数指定放于 D21 内。所得结果的正负由最高位位的 OFF/ON 来代表正或负值。



补充说明：

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若除数为零，指令不执行，SM0=ON，错误码 SR0=16#2012。
3. 16 位指令的 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
4. 32 位指令的 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of DWORD/DINT。

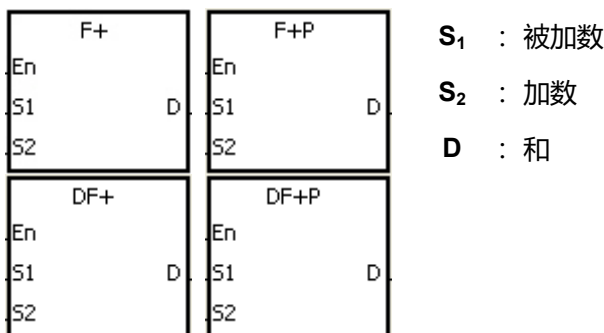
FB/FC	指令码			操作数					功能						
FC	D*	F+	P	S ₁ , S ₂ , D					浮点数加法						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂										●	●*			
D										●	●*			

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：

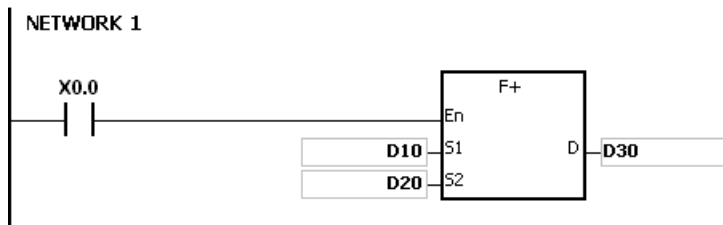


指令说明：

- S₁ 及 S₂ 以浮点数方式相加的结果存于 D。
- 32 位单精度浮点数加法：
 - 运算结果为 0 时，零标号 (Zero flag) SM600 为 ON。
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FF7FFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，则 D=16#7F7FFFFFFF。
- 64 位双精度浮点数加法：
 - 运算结果为 0 时，零标号 (Zero flag) SM600 为 ON。
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

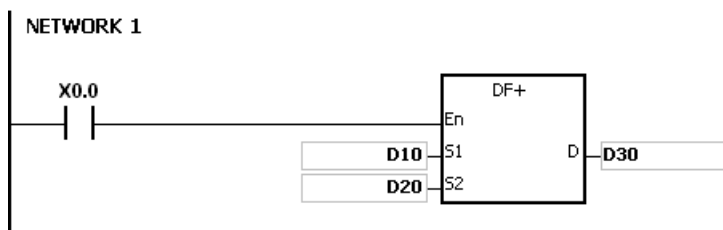
程序范例：

单精度浮点数加法：当 X0.0=ON 时，被加数(D11 , D10)=16#3FB9999A(表浮点值 1.450)加上加数(D21 , D20)=16#4046B852 (表浮点值 3.105) 将结果存在 (D31 , D30)=16#4091C28F (表浮点值 4.555) 的内容当中。



双精度浮点数加法：当 X0.0=ON 时，被加数 (D13 , D12 , D11 , D10)=16#3FF7333333333333 (表浮点值 1.450) 内容加上加数 (D23 , D22 , D21 , D20)=16#4008D70A3D70A3D7 的内容将结果存在 (D33 , D32 , D31 , D30)=16# 40123851EB851EB8 (表浮点值 4.555) 的内容当中。

3

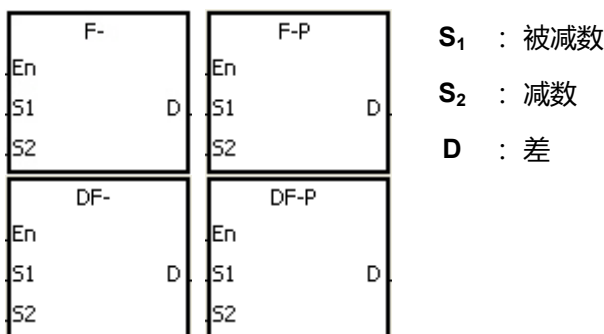


补充说明：

当 S₁ 或 S₂ 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数						功能							
FC	D*	F-	P	S ₁ , S ₂ , D						浮点数减法							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂										●	●*						
D										●	●*						
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				
				脉冲执行型				32 位指令				64 位指令					
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机					

图形：

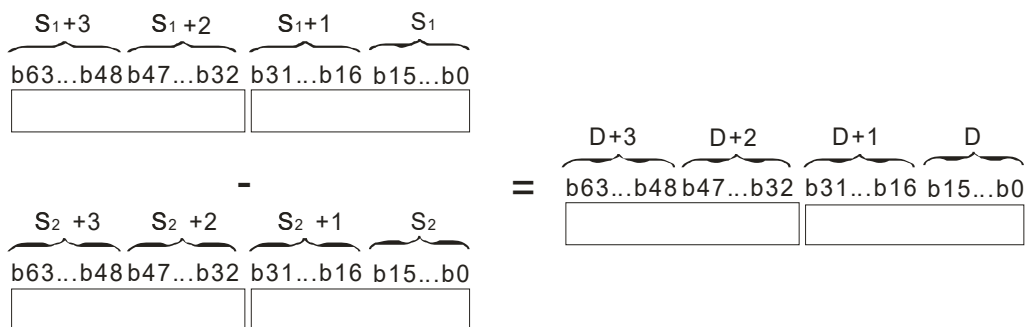


指令说明：

- S₁ 及 S₂ 以浮点数类型相减的结果存于 D。
- 若运算结果为 0，则零标号 SM600=ON。
- 32 位单精度浮点数减法：
 - 若运算结果的绝对值小于可表示的最小浮点值，则 D=16#FF7FFFFFFF。
 - 若运算结果的绝对值大于可表示的最大浮点值，则 D=16#7F7FFFFFFF。

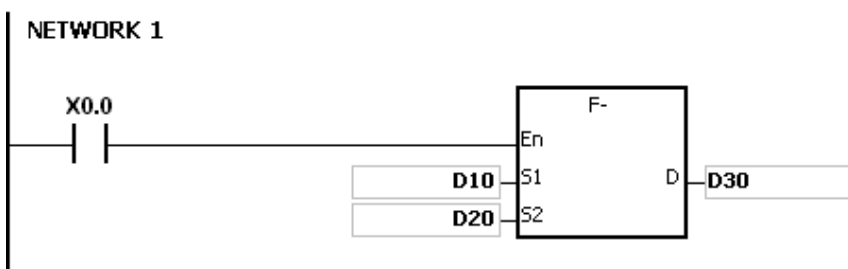
$$\begin{array}{c}
 \overbrace{b31 \dots b16}^{S_1 + 1} \quad \overbrace{b15 \dots b0}^{S_1} \\
 \boxed{ }
 \end{array}
 -
 \begin{array}{c}
 \overbrace{b31 \dots b16}^{S_2 + 1} \quad \overbrace{b15 \dots b0}^{S_2} \\
 \boxed{ }
 \end{array}
 =
 \begin{array}{c}
 \overbrace{b31 \dots b16}^{D+1} \quad \overbrace{b15 \dots b0}^D \\
 \boxed{ }
 \end{array}$$

- 64 位双精度浮点数减法：
 - 若运算结果的绝对值小于可表示的最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示的最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

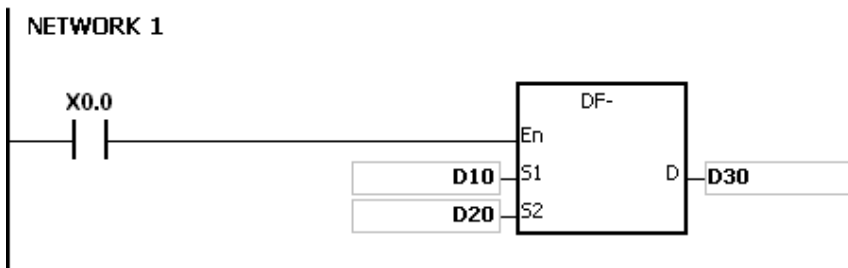


程序范例：

1. 32 位单精度浮点数减法：当 X0.0=ON 时，被减数 (D11 , D10) 内容减去减数 (D21 , D20) 的内容将结果存在 (D31 , D30) 的内容当中。



2. 64 位双精度浮点数减法：当 X0.0=ON 时，被减数 (D13 , D12 , D11 , D10) 内容减去减数 (D23 , D22 , D21 , D20) 的内容将结果存在 (D33 , D32 , D31 , D30) 的内容当中。



补充说明：

当 S₁ 或 S₂ 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

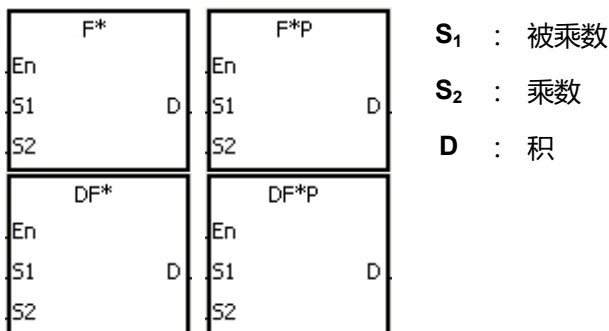
FB/FC	指令码			操作数						功能					
FC	D*	F*	P	S ₁ , S ₂ , D						浮点数乘法					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂										●	●*			
D										●	●*			

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

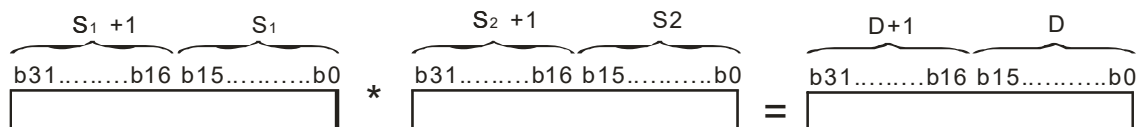
脉冲执行型	32 位指令	64 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：

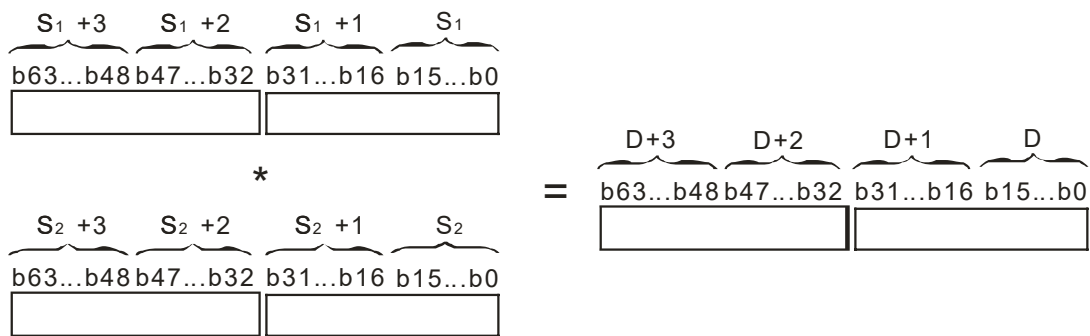


指令说明：

- S₁ 及 S₂ 以浮点数类型相乘的结果存于 D。
- 若运算结果为 0，则零标号 SM600=ON。
- 32 位单精度浮点数乘法：
 - 若运算结果的绝对值小于可表示的最小浮点值，则 D=16#FF7FFFFFFF。
 - 若运算结果的绝对值大于可表示的最大浮点值，则 D=16#7F7FFFFFFF。

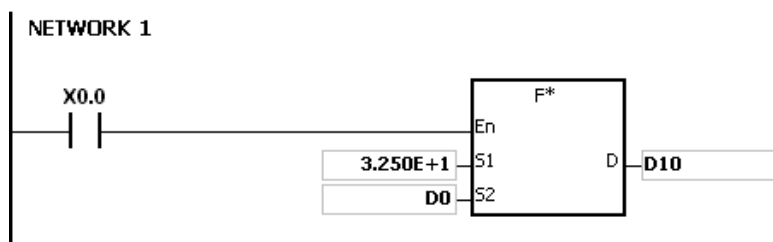


- 64 位双精度浮点数乘法：
 - 若运算结果的绝对值小于可表示的最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示的最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

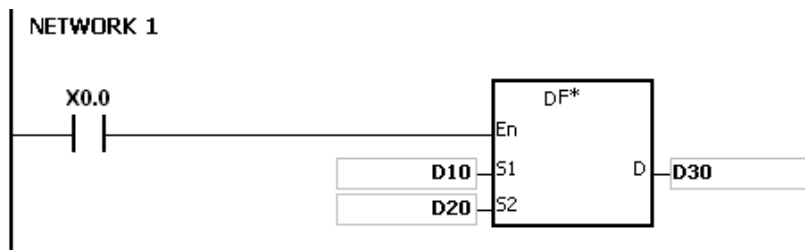


程序范例：

1. 32 位单精度浮点数乘法：当 X0.0=ON 时，被乘数常数 32.5 乘上乘数 (D1 , D0) 的内容，并将结果存在 (D11 , D10) 的内容当中。



2. 64 位双精度浮点数乘法：当 X0.0=ON 时，被乘数 (D13 , D12 , D11 , D10) 的内容乘上乘数 (D23 , D22 , D21 , D20) 之内容，并将结果存在 (D33 , D32 , D31 , D30) 的内容当中。

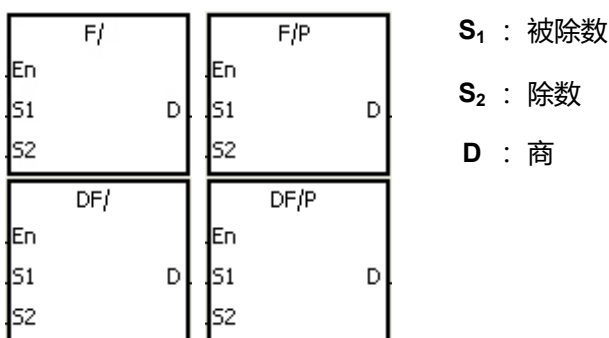


补充说明：

当 S₁ 或 S₂ 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数						功能							
FC	D*	F/	P	S ₁ , S ₂ , D						浮点数除法							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂										●	●*						
D										●	●*						
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				
				脉冲执行型				32 位指令				64 位指令					
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机					

图形：

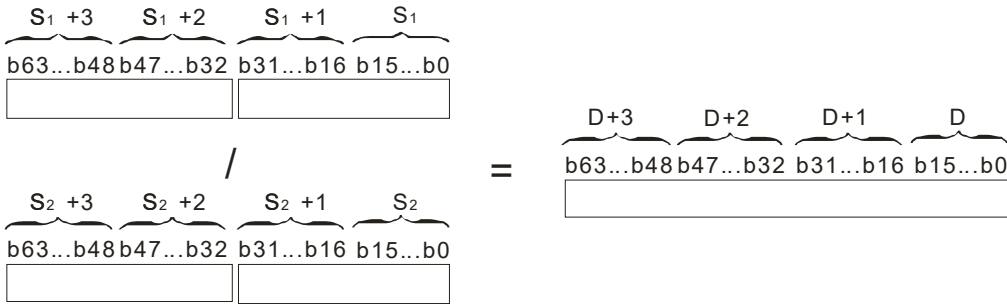


指令说明：

- S₁ 及 S₂ 以单精度浮点数类型相除的结果存于 D。
- 若运算结果为 0，则零标号 SM600=ON。
- 32 位单精度浮点数除法：
 - 若运算结果的绝对值小于可表示的最小浮点值，则 D=16#FF7FFFFFFF。
 - 若运算结果的绝对值大于可表示的最大浮点值，则 D=16#7F7FFFFFFF。

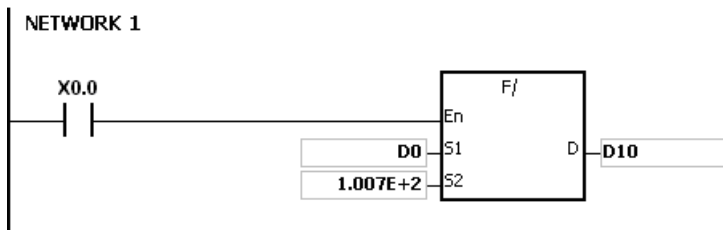
$$\begin{array}{c}
 \overbrace{\text{b31} \dots \text{b16}}^{\text{S}_1 + 1} \quad \overbrace{\text{b15} \dots \text{b0}}^{\text{S}_1} \\
 \boxed{\phantom{\text{b31} \dots \text{b16} \text{ b15} \dots \text{b0}}}
 \end{array}
 /
 \begin{array}{c}
 \overbrace{\text{b31} \dots \text{b16}}^{\text{S}_2 + 1} \quad \overbrace{\text{b15} \dots \text{b0}}^{\text{S}_2} \\
 \boxed{\phantom{\text{b31} \dots \text{b16} \text{ b15} \dots \text{b0}}}
 \end{array}
 =
 \begin{array}{c}
 \overbrace{\text{b31} \dots \text{b16}}^{\text{D} + 1} \quad \overbrace{\text{b15} \dots \text{b0}}^{\text{D}} \\
 \boxed{\phantom{\text{b31} \dots \text{b16} \text{ b15} \dots \text{b0}}}
 \end{array}$$

- 64 位双精度浮点数除法：
 - 若运算结果的绝对值小于可表示的最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示的最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

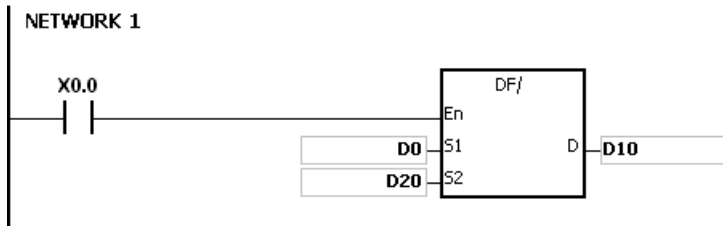


程序范例：

- 32 位单精度浮点数除法：当 X0.0=ON 时，被除数(D1 ,D0)的内容除以除数 100.7 ,并将结果存在(D11 , D10) 的内容当中。



- 64 位双精度浮点数除法：当 X0.0=ON 时，被除数 (D3 , D2 , D1 , D0) 的内容除以除数 (D23 , D22 , D21 , D20) 的内容，并将结果存在 (D13 , D12 , D11 , D10) 的内容当中。



补充说明：

- 除数为 0 时，指令不执行，SM0=ON，错误码 SR0=16#2012。
- 当 S1 或 S2 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

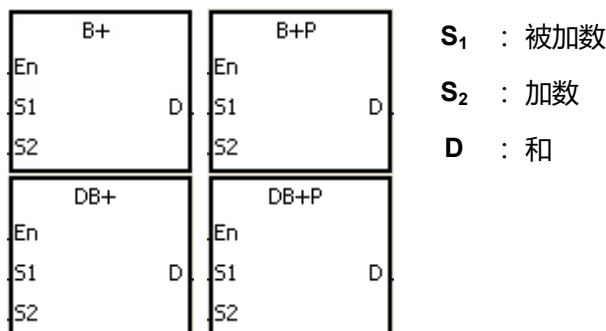
FB/FC	指令码			操作数				功能						
FC	D*	B+	P	S ₁ , S ₂ , D				BCD 加法						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



指令说明：

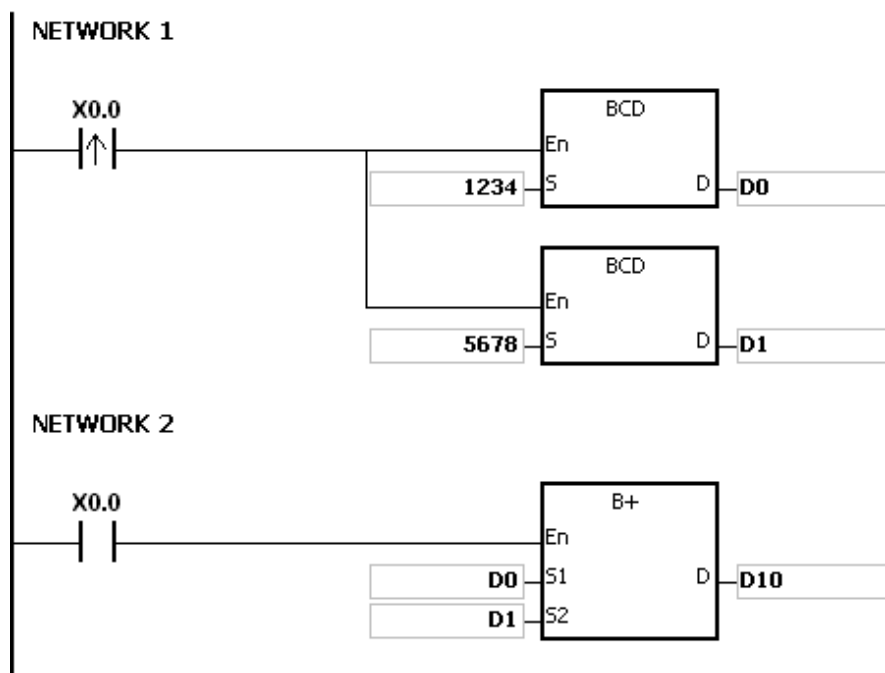
- S₁ 及 S₂ 以 BCD 方式相加的结果存于 D。
- DB+指令才可以使用 HC 装置。
- BCD 值以 HEX 表示且每一个位是都要在 0~9 之间。
- 16 位 BCD 加法：

当 S₁=9999 (BCD)，S₂=0002 (BCD)，则 S₁+S₂=10001 (BCD)，则进位被忽略，D=0001 (BCD)。
- 32 位 BCD 加法：

当 S₁=99999999 (BCD)，S₂=00000002 (BCD)，则 S₁+S₂=100000001 (BCD)，则进位被忽略，D=00000001 (BCD)。

程序范例：

当 X0.0=ON 时，分别将常数 1234 和 5678 作 BCD 转换后储存至 D0 与 D1，接着 D0 与 D1 作 BCD 加法后将结果储存至 D10 当中。



补充说明：

1. B+指令时当 S_1 或 S_2 超出 BCD 可表示范围时 0~9999，指令不执行，SM0=ON，错误码 SR0=16#200D（BCD 值以 Hex 表示有任一位数不在 0~9 的范围内）。
2. DB+指令时当 S_1 或 S_2 超出 BCD 可表示范围时 0~99999999，指令不执行，SM0=ON，错误码 SR0=16#200D（BCD 值以 Hex 表示有任一位数不在 0~9 的范围内）。
3. 此指令不支持零标志 SM600、借位标志 SM601 以及进位标志 SM602。

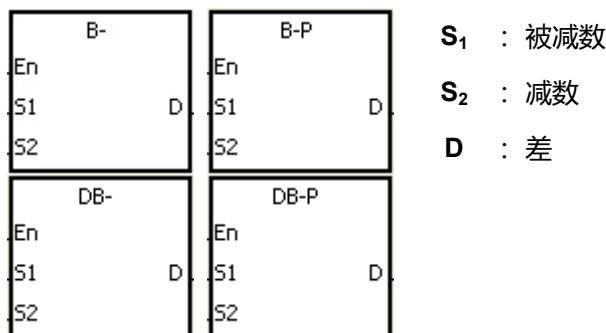
FB/FC	指令码			操作数				功能						
FC	D*	B-	P	S ₁ , S ₂ , D				BCD 减法						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		○
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



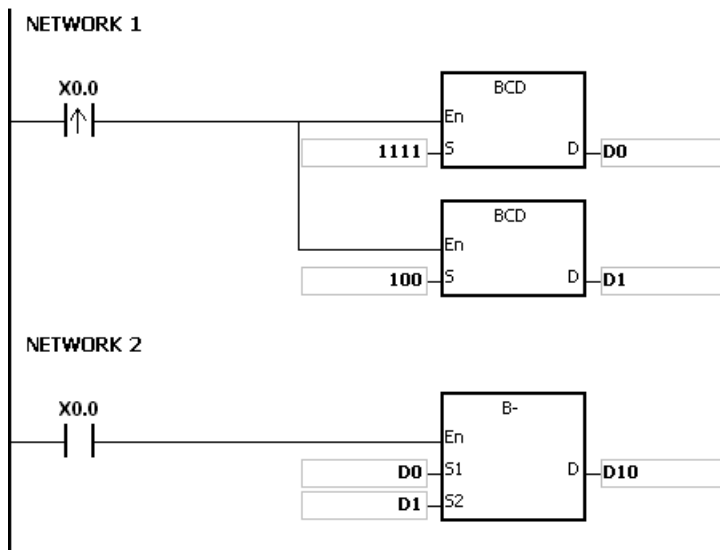
指令说明：

- S₁ 及 S₂ 以 BCD 方式相减的结果存于 D。
- DB-指令才可以使用 HC 装置。
- BCD 值以 HEX 表示且每一个位是都要在 0~9 之间。
- 16 位 BCD 减法：
 - 当 S₁=9999 (BCD)，S₂=9998 (BCD)，则 S₁-S₂=0001 (BCD)，则 D=0001 (BCD)。
 - 当 S₁=0001 (BCD)，S₂=9999 (BCD)，则 S₁-S₂=-9998 (BCD)，则 D=0002 (BCD)。
 - 当 S₁=0001 (BCD)，S₂=0004 (BCD)，则 S₁-S₂=-0003 (BCD)，则 D=9997 (BCD)。
- 32 位 BCD 减法：
 - 当 S₁=99999999 (BCD)，S₂=99999998 (BCD)，则 S₁-S₂=00000001 (BCD)，D=00000001 (BCD)。

- 当 $S_1=00000001$ (BCD) , $S_2=99999999$ (BCD) , 则 $S_1-S_2=-99999998$ (BCD) , 则 $D=00000002$ (BCD) 。
- 当 $S_1=00000001$ (BCD) , $S_2=00000004$ (BCD) , 则 $S_1-S_2=-00000003$ (BCD) , 则 $D=99999997$ (BCD) 。

程序范例：

当 X0.0=ON 时，分别将常数 1111 和 100 作 BCD 转换后储存至 D0 与 D1，接着 D0 与 D1 作 BCD 减法后将结果储存至 D10 当中。

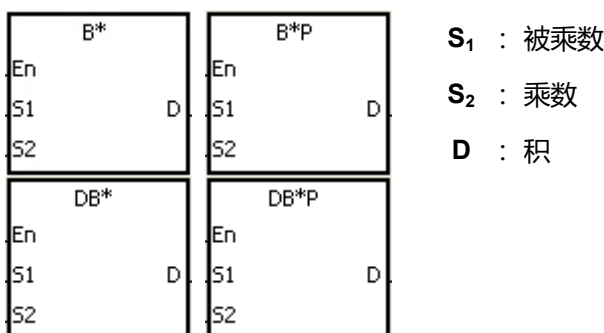


补充说明：

1. B-指令时当 S_1 或 S_2 超出 BCD 可表示范围时 0~9999，指令不执行，SM0=ON，错误码 SR0=16#200D (BCD 值以 Hex 表示有任一位数不在 0~9 的范围内)。
2. DB-指令时当 S_1 或 S_2 超出 BCD 可表示范围时 0~99999999，指令不执行，SM0=ON，错误码 SR0=16#200D (BCD 值以 Hex 表示有任一位数不在 0~9 的范围内)。
3. 此指令不支持零标志 SM600、借位标志 SM601 以及进位标志 SM602。

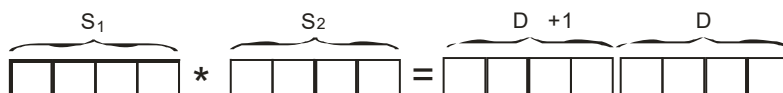
FB/FC	指令码			操作数						功能							
FC	D*	B*	P	S ₁ , S ₂ , D						BCD 乘法							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂		●	●*				●	●*									
D			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				
			脉冲执行型			16 位指令			32 位指令								
			AH 运动控制主机			AH 运动控制主机			AH 运动控制主机								

图形：



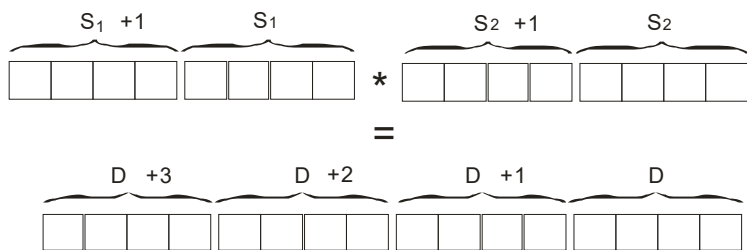
指令说明：

- S₁ 及 S₂ 以 BCD 方式相乘的结果存于 D。
- DB*指令才可以使用 HC 装置。
- BCD 值以 HEX 表示且每一个位是都要在 0~9 之间。
- 16 位 BCD 乘法：
 - 当 S₁=1234 (BCD)，S₂=5678 (BCD)，则 D=07006652 (BCD)。



积为 32 位数据，储存在 (D+1, D) 组成的 32 位寄存器中。

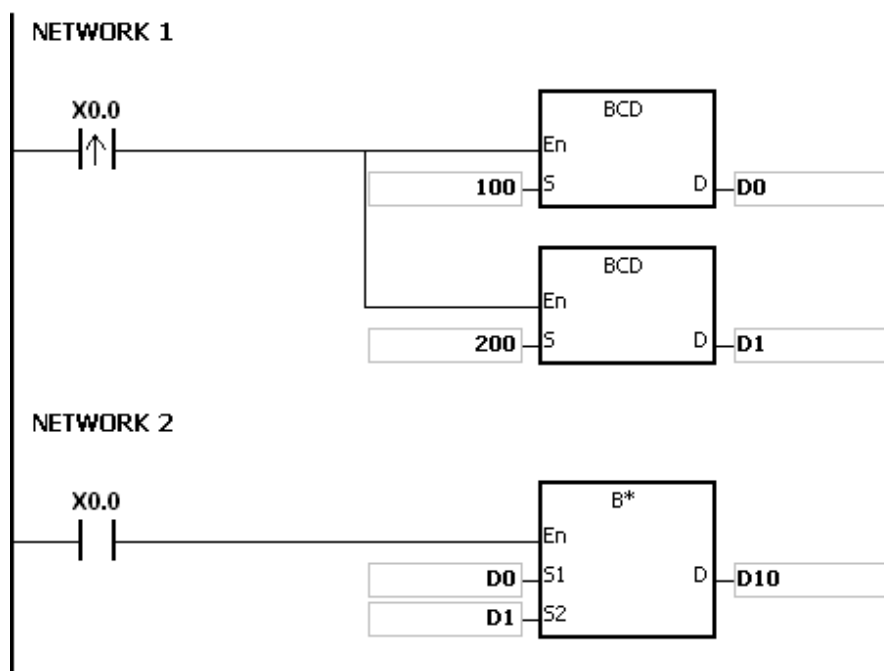
- 32 位 BCD 乘法：
 - 当 S₁=99999999 (BCD)，S₂=99999998 (BCD)，则 D=9999999700000002 (BCD)。



积为 64 位数据，储存在 (D+3, D+2, D+1, D) 组成的 64 位寄存器中。

程序范例：

1. 当 X0.0=ON 时，分别将常数 100 和 200 作 BCD 转换后储存至 D0 与 D1，接着 D0 与 D1 作 BCD 乘法后将结果储存至 D10~D11 当中。



补充说明：

1. B*指令时当 S₁ 或 S₂ 超出 BCD 可表示范围时 0~9999，指令不执行，SM0=ON，错误码=16#200D (BCD 值以 Hex 表示有任一位数不在 0~9 的范围内)。
2. DB*指令时当 S₁ 或 S₂ 超出 BCD 可表示范围时 0~99999999，指令不执行，SM0=ON，错误码=16#200D (BCD 值以 Hex 表示有任一位数不在 0~9 的范围内)。
3. 此指令不支持零标志 SM600、借位标志 SM601 以及进位标志 SM602。

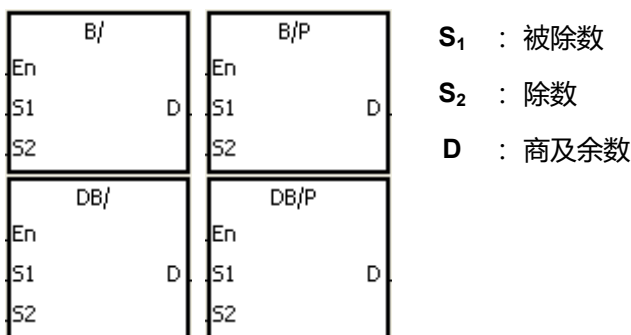
FB/FC	指令码			操作数						功能								
FC	D*	B/	P	S ₁ , S ₂ , D						BCD 除法								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂		●	●*				●	●*										
D			●	●*				●	●*									

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

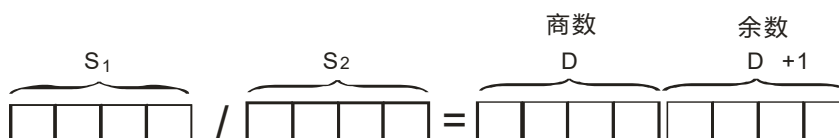
3

图形：



指令说明：

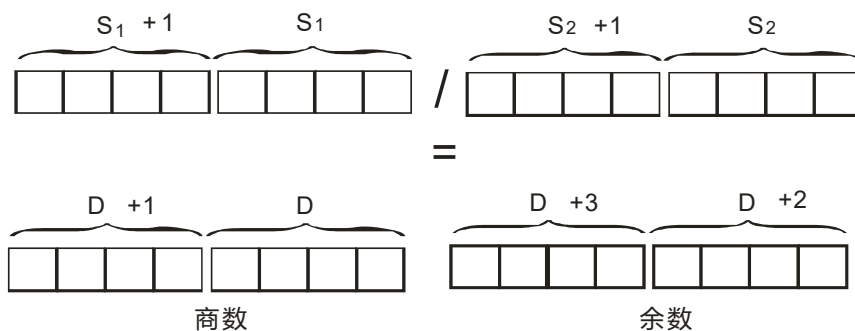
- S₁ 及 S₂ 以 BCD 方式相除的结果存于 D。
- DB/指令才可以使用 HC 装置。
- BCD 值以 HEX 表示且每一个位是都要在 0~9 之间。
- 16 位 BCD 除法：
 - 当 S₁=5678 (BCD)，S₂=1234 (BCD)，则 D=0004，D+1=0742 (BCD)。



D 操作数连续占用两个，D 储存商，D+1 储存余数。

- 32 位 BCD 除法：

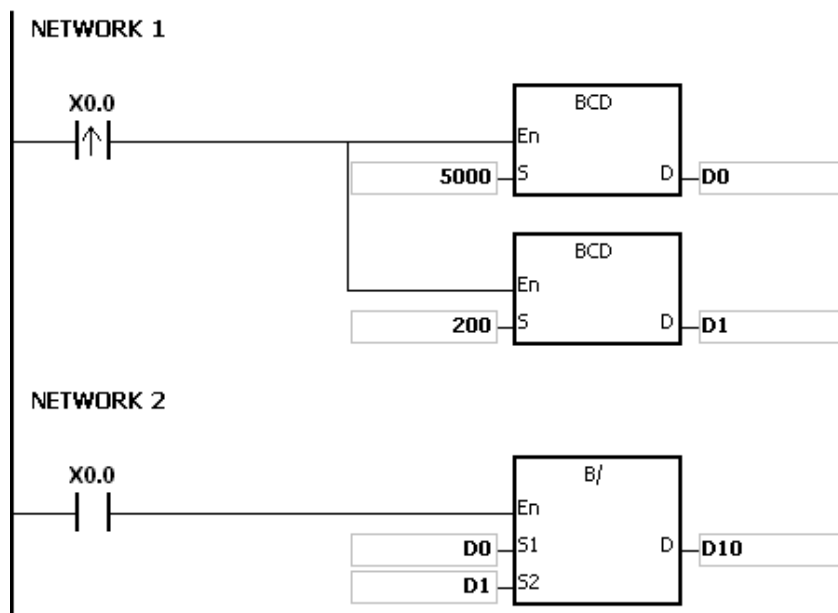
- 当 $S_1=87654321$ (BCD) , $S_2=12345678$ (BCD) , 则 $(D+1, D) = 00000007$ (BCD) , $(D+3, D+2) = 01234575$ 。



- D 操作数连续占用两个, $(D+1, D)$ 储存商, $(D+3, D+2)$ 储存余数

程序范例：

当 X0.0=ON 时, 分别将常数 5000 和 200 作 BCD 转换后储存至 D0 与 D1, 接着 D0 与 D1 作 BCD 除法后将商与余数储存至 D10~D11 当中。



补充说明：

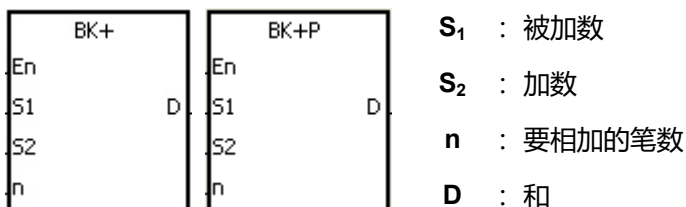
1. 除数为 0 时, 指令不执行, SM0=ON, 错误码=16#2012。
2. B/指令时当 S_1 或 S_2 超出 BCD 可表示范围时 0~9999, 指令不执行, SM0=ON, 错误码=16#200D (BCD 值以 Hex 表示有任一位数不在 0~9 的范围内)。
3. DB/指令时当 S_1 或 S_2 超出 BCD 可表示范围时 0~99999999, 指令不执行, SM0=ON, 错误码=16#200D (BCD 值以 Hex 表示有任一位数不在 0~9 的范围内)。
4. 此指令不支持零标志 SM600、借位标志 SM601 以及进位标志 SM602。
5. 16 位指令的 D 操作数, 若使用 ISPSOft 宣告, 则数据类型为 ARRAY [2] of WORD/INT。
6. 32 位指令的 D 操作数, 若使用 ISPSOft 宣告, 则数据类型为 ARRAY [2] of DWORD/DINT。

FB/FC	指令码			操作数						功能						
FC		BK+	P	S_1, S_2, n, D						连续区块 BIN 加法						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING		
S_1		●					●									
S_2		●					●									
n		●					●									
D		●					●									

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●	●	●	●		●	○	●				○
S_2	●	●			●	●	●	●	●		●	○	●	○	○		
n	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

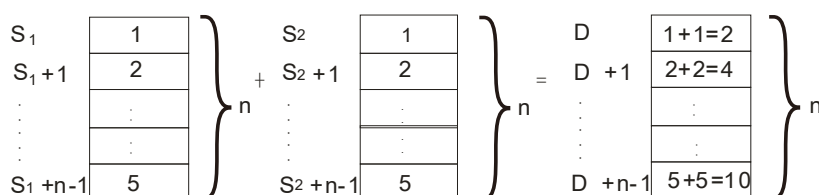
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

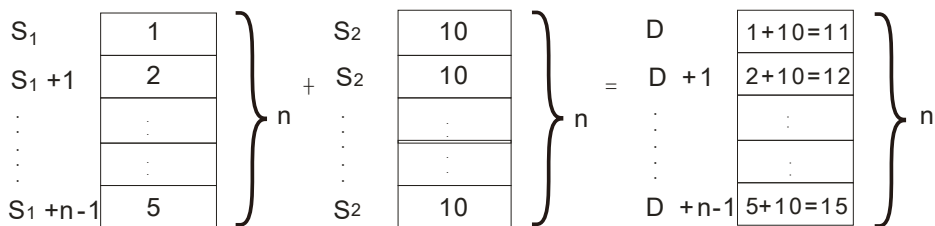


指令说明：

1. 连续 n 个 S_1 及 S_2 以 BIN 方式相加的结果存于 D 。
2. n 操作数的范围为 1~256。
3. 运算结果为 0 时，零标号 (Zero flag) SM600 为 ON。
4. 运算结果小于 -32,768 时，借位标号 (Borrow flag) SM601 为 ON。
5. 运算结果大于 32,767 时，进位标号 (Carry flag) SM602 为 ON。
6. 当 S_2 是装置 (非 K/16#) 的时候：

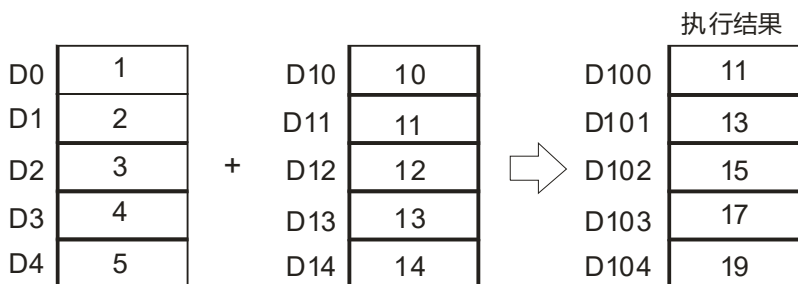
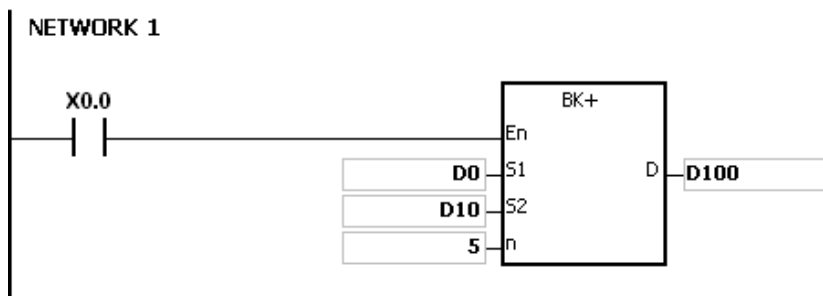


7. 当 S₂是常数 K/16#的时候：



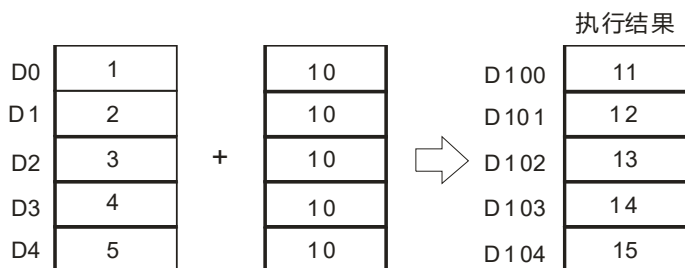
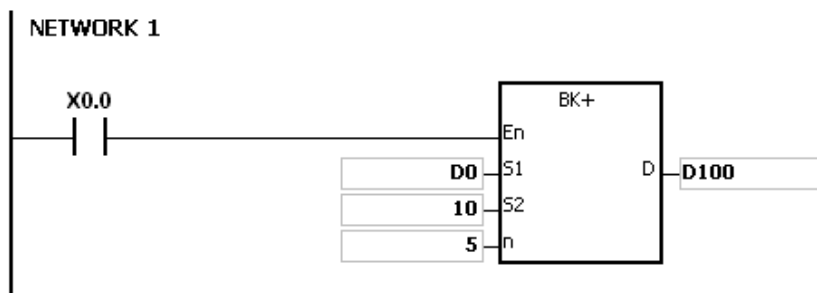
程序范例一：

X0.0=ON 时，将被加数 D0~D4 内容与加数 D10~D14 内容作 BIN 相加后储存至 D100~D104 当中。



程序范例二：

X0.0=ON 时，将被加数 D0~D4 的内容与加数 10 的内容作 BIN 相加后储存至 D100~D104 当中。



补充说明：

1. S_1 、 S_2 或 D 装置的起始地址 $\sim n-1$ 的地址 超出装置范围时 指令不执行 $SM0=ON$ 错误码 $SR0=16\#2003$ 。
2. 若 $n < 1$ 或 $n > 256$ 时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200B$ 。
3. 若 $S_1 \sim S_1+n-1$ 的装置与 $D \sim D+n-1$ 的装置有重叠时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200C$ 。
4. 若 $S_2 \sim S_2+n-1$ 的装置与 $D \sim D+n-1$ 的装置有重叠时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200C$ 。
5. 若 $S_1 \sim S_1+n-1$ 的装置与 $S_2 \sim S_2+n-1$ 的装置有重叠时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200C$ 。

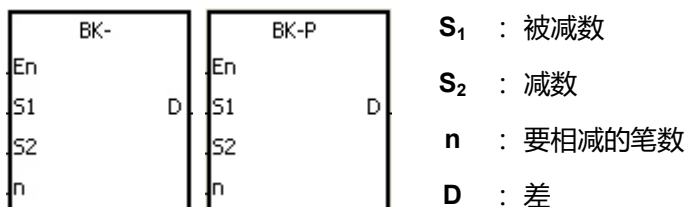
FB/FC	指令码		操作数				功能				
FC		BK-	P	S_1, S_2, n, D				连续区块 BIN 减法			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1			●					●						
S_2			●					●						
n			●					●						
D			●					●						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●	●	●	●		●	○	●				○
S_2	●	●			●	●	●	●	●		●	○	●	○	○		
n	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

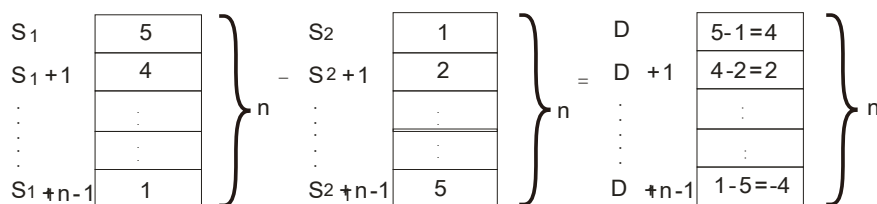
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

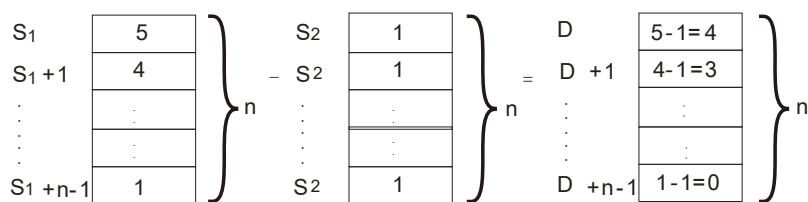


指令说明：

1. 连续 n 个 S_1 及 S_2 以 BIN 方式相减的结果存于 D 。
2. n 操作数的范围为 1~256。
3. 运算结果为 0 时，零标号 (Zero flag) SM600 为 ON。
4. 运算结果小于 -32,768 时，借位标号 (Borrow flag) SM601 为 ON。
5. 运算结果大于 32,767 时，进位标号 (Carry flag) SM602 为 ON。
6. 当 S_2 是装置 (非 K/16#) 的时候：

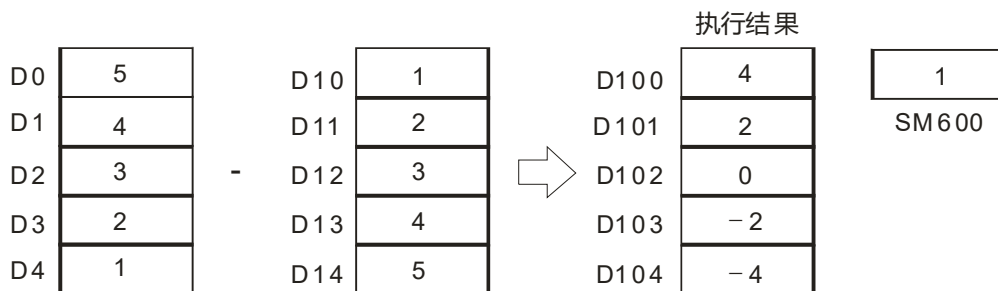
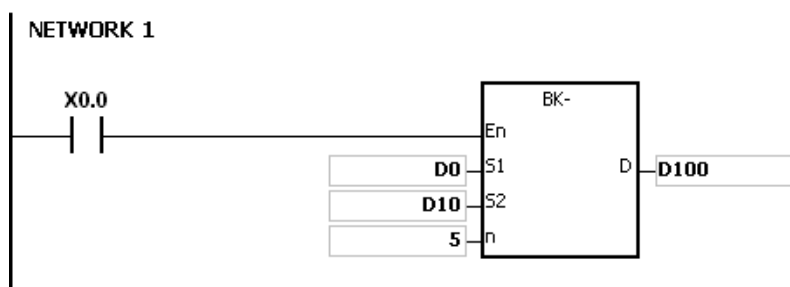


7. 当 S_2 是常数 K/16# 的时候：



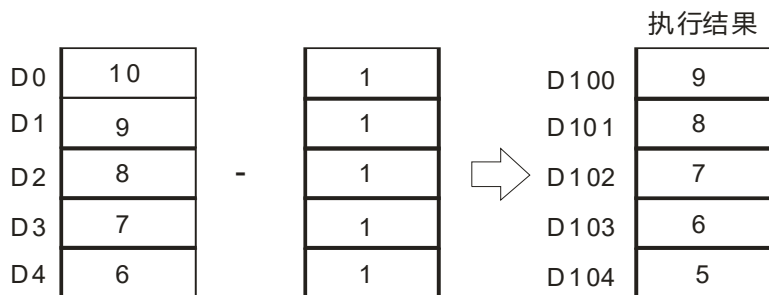
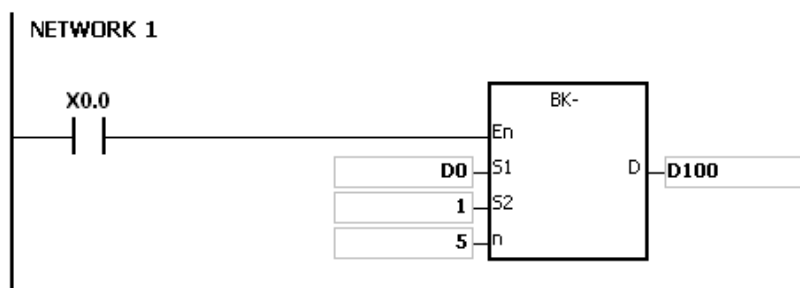
程序范例一：

X0.0=ON 时，将被减数 D0~D4 内容与减数 D10~D14 内容作 BIN 相减后储存至 D100~D104 当中。



程序范例二：

X0.0=ON 时，将被加数 D0~D4 的内容与减数 1 的内容作 BIN 相减后储存至 D100~D104 当中。



补充说明：

1. S_1 、 S_2 或 D 装置的起始地址 $\sim n-1$ 的地址，超出装置范围时，指令不执行 $SM0=ON$ ，错误码=16#2003。
2. 若 $n < 1$ 或 $n > 256$ 时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200B$ 。
3. 若 $S_1 \sim S_1+n-1$ 的装置与 $D \sim D+n-1$ 的装置有重叠时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200C$ 。
4. 若 $S_2 \sim S_2+n-1$ 的装置与 $D \sim D+n-1$ 的装置有重叠时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200C$ 。
5. 若 $S_1 \sim S_1+n-1$ 的装置与 $S_2 \sim S_2+n-1$ 的装置有重叠时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200C$ 。

FB/FC	指令码			操作数						功能					
FC		\$+	P	S ₁ , S ₂ , D						字符串链接					

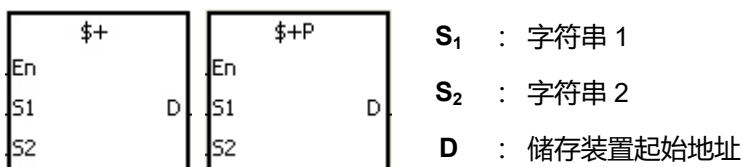
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂														●
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●		●	●		●	○	●			○	
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

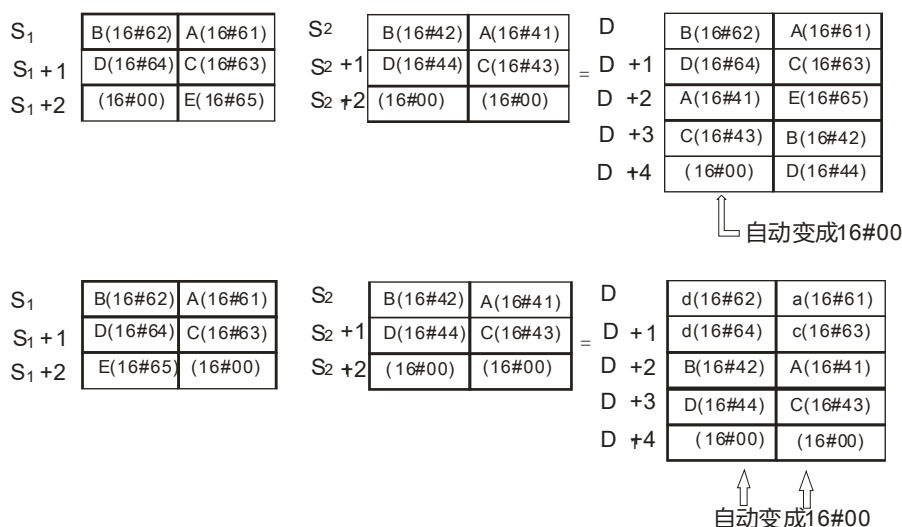
3

图形：



指令说明：

- 该指令执行时，将 S_1 操作数 $S_1 \sim 16\#00$ （不搬 $16\#00$ ）与 S_2 操作数 $S_2 \sim 16\#00$ （不搬 $16\#00$ ）之间的字符串链接后搬移至 D 操作数内且结尾会补上结后缀（ $16\#00$ ）。当指令不执行时， D 内容不会变化。
- S_1 操作数及 S_2 操作数两个字符串链接的结果存于 D 操作数的说明如下：

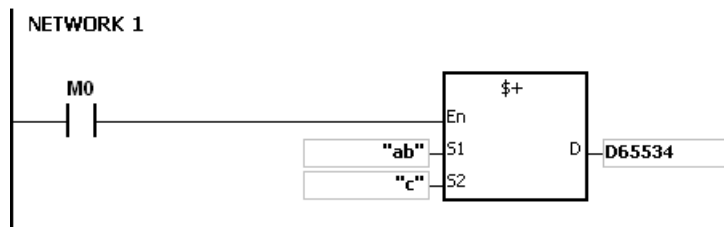


- 若 S_1 或 S_2 操作数的来源不是字符串时，就要在数据结尾后面有一个结后缀（ $16\#00$ ）。

4. 若 S_1 或 S_2 操作数不是字符串且该指令执行时，第一个字符就是结后缀 (16#00)，则还是会搬移结后缀 (16#00) 做链接。

程序范例：

给定 S_1 ="ab"， S_2 ="C"，启动条件接点 M0 后，D65534=16#4241，D65535=16#0043



补充说明：

1. 若 S_1 或 S_2 操作数为字符串时， S_1 或 S_2 操作数各可以搬移 1~31 个字符；且字符串的 Step 数计算方式为 Step 数=1+ (字符数+1) /4 (若不能整除则无条件进位)。Step 数计算表格如下：

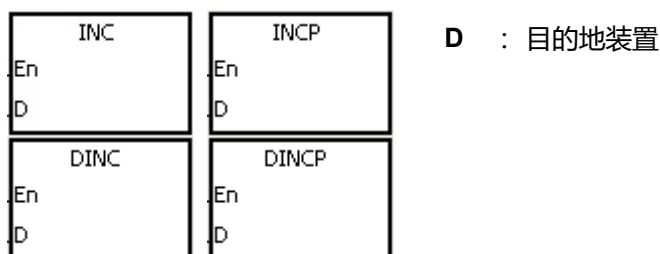
字符数	1~3	4~7	8~11	12~15	16~19	20~23	24~27	28~31
Step 数	2	3	4	5	6	7	8	9

范例：字符串+"ABCDE" D0 D100 则 Step 数=1 (指令) +3 (字符串) +2 (D0) +2 (D100) =8

2. 当 D 装置的空间不足以存放 S_1+S_2 字符串时，指令不执行，SM0=ON，错误码 SR0=16#2003。
 3. S_1 或 S_2 装置与 D 装置有重叠时，指令不执行，SM0=ON，错误码 SR0=16#200C。
 4. S_1 或 S_2 没有 16#00 当结尾时，指令不执行，SM0=ON，错误码 SR0=16#200E。

FB/FC	指令码			操作数							功能							
FC	D*	INC	P	D							BIN 加 1							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
D		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
D	●	●			●	●	●	●	●		●	○	●					
脉冲执行型							16 位指令							32 位指令				
AH 运动控制主机							AH 运动控制主机							AH 运动控制主机				

图形：



指令说明：

1. D 内容加 1。
2. DINC 指令才可以使用 HC 装置。
3. 16 位运算时 ,32,767 再加 1 则变为-32,768。32 位运算时 ,2,147,483,647 再加 1 则变为-2,147,483,648。

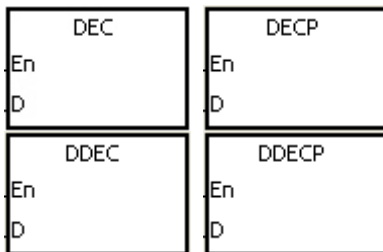
程序范例：

当 X0.0=OFF→ON 时，D0 内容自动加 1。



FB/FC		指令码			操作数						功能						
FC		D*	DEC	P	D						BIN 减 1						
数据类型		BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING		
D			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型						16 位指令						32 位指令					
AH 运动控制主机						AH 运动控制主机						AH 运动控制主机					

图形：



D : 目的地装置

指令说明：

1. D 内容减 1。
2. DDEC 指令才可以使用 HC 装置。
3. 16 位运算时，-32,768 再减 1 则变为 32,767。32 位运算时，-2,147,483,648 再减 1 则变为 2,147,483,647。

程序范例：

当 X0.0=OFF→ON 时，D0 内容自动减 1。



FB/FC	指令码			操作数			功能		
FC	MUL16 MUL32*	P		S ₁ , S ₂ , D			16 位专用 BIN 乘法 32 位专用 BIN 乘法		

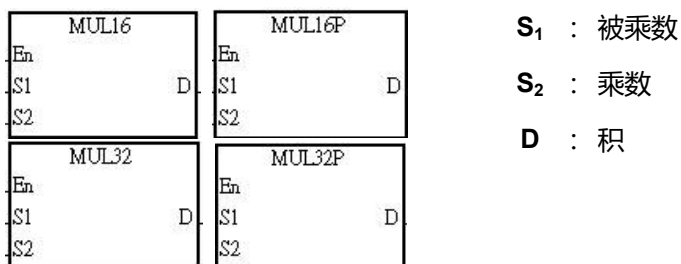
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

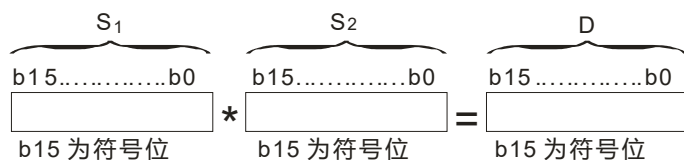
3

图形：



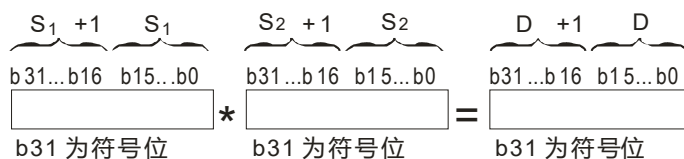
指令说明：

- S₁ 及 S₂ 以有号数二进制方式相乘后的积存于 D。
- MUL32 指令才可以使用 HC 装置。
- 16 位 BIN 乘法运算：



积为 16 位数据，储存在 D 的 16 位寄存器中，且符号位 b15=0 为正数，符号位 b15=1 为负数。

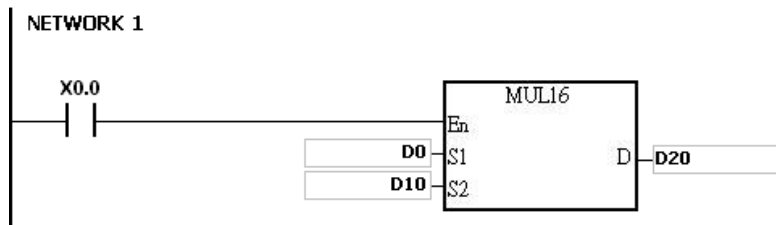
- 32 位 BIN 乘法运算：



积为 32 位数据，储存在 (D+1 , D) 组成的 32 位寄存器中，且符号位 b31=0 为正数，符号位 b31=1 为负数。

程序范例：

16 位 D0 乘上 16 位 D10 得到一个 16 位的积，结果存在 D20。结果的正负由最高位 (b15) 之 OFF/ON 来指示。OFF 表示正的 (0) ，同时 ON 表示负的 (1) 。



$D0 \times D10 = D20$

16 位 \times 16 位 = 16 位

3

补充说明：

1. 当 16 位乘法的积超出 16 位有号数可表示范围时，则数值比 16 位最大正数 (K32767) 还大或者数值比最小负数 (K-32768) 还小时，设定 SM602 进位标志为 ON；并只写入低 16 位的数值
2. 若 16 位指令相乘结果需要得到完整的数值 (纪录为 32 位) ，请改用 API0102 * / *P 指令，详细说明请参考该指令。
3. 当 32 位乘法的积超出 32 位有号数可表示范围时，则数值比 32 位最大正数 (K2147483647) 还大或者数值比最小负数 (K-2147483648) 还小时，设定 SM602 进位标志为 ON；并只写入低 32 位的数值。
4. 若 32 位指令相乘结果需要得到完整的数值 (纪录为 64 位) ，请改用 API0102 D* / D*P 指令，详细说明请参考该指令。

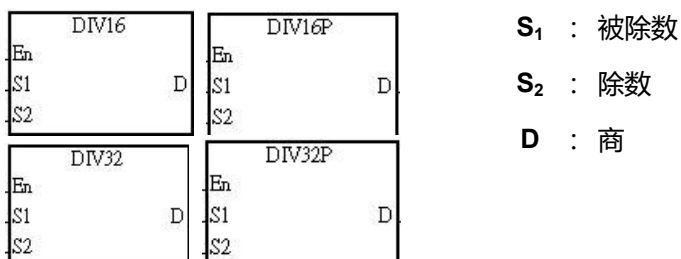
FB/FC	指令码			操作数			功能		
FC	DIV16	DIV32*	P	S₁, S₂, D			16 位专用 BIN 除法 32 位专用 BIN 除法		

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S₁, S₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S₁, S₂	●	●			●	●	●	●	●		●	○	●	●	●		
D	●	●			●	●	●	●	●		●	○	●				

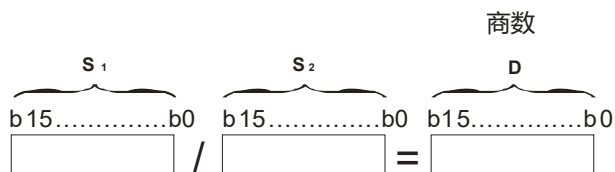
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



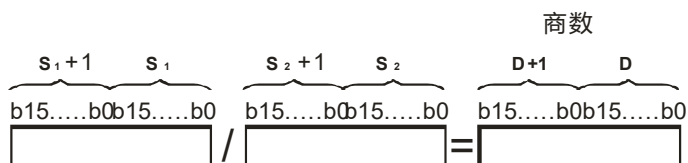
指令说明：

1. S₁ 及 S₂ 以有号数二进制方式相除后的商存于 D。
2. 32 位指令才可以使用 HC 装置。
3. 符号位=0 为正数，符号位=1 为负数。
4. 16 位 BIN 除法运算：



D 操作数储存商。

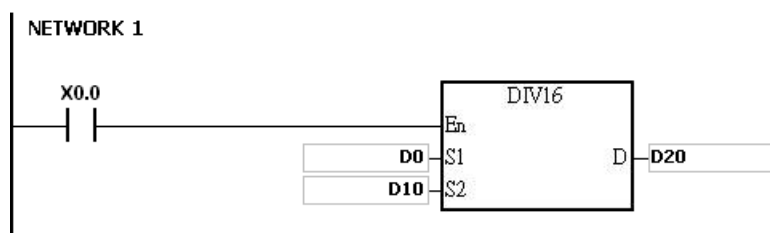
5. 32 位 BIN 除法运算：



D 操作数续占用两个，(D+1, D) 储存商。

程序范例：

当 X0.0=ON 时 被除数 D0 除以除数 D10 而结果商被指定放于 D20。所得结果的正负由最高位位的 OFF/ON 来代表正或负值。



3

补充说明：

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若除数为零，指令不执行，SM0=ON，错误码 SR0=16#2012。
3. 若需要记录余数，请改用 API0103，详细说明请参考该指令。

3.5 数据转换指令

FB/FC	指令码 (位)			P 指令	功能	Step
	16	32	64			
FC	<u>BCD</u>	<u>DBC</u> D	—	✓	BIN→BCD 变换	5
FC	<u>BIN</u>	<u>DBIN</u>	—	✓	BCD→BIN 变换	5
FC	<u>FLT</u>	<u>DFLT</u>	—	✓	BIN 整数→单精度浮点数变换	5
FC	<u>FLTD</u>	<u>DFLTD</u>	—	✓	BIN 整数→双精度浮点数变换	5
FC	<u>INT</u>	<u>DINT</u>	—	✓	单精度浮点数→BIN 整数变换	5
FC	—	<u>FINT</u>	<u>DFINT</u>	✓	双精度浮点数→BIN 整数变换	5
FC	—	<u>FTODF</u>	—	—	单精度浮点数→双精度浮点数转换	5
FC	—	—	<u>DFTOF</u>	—	双精度浮点数→单精度浮点数转换	5
FC	<u>MMOV</u>	—	—	✓	16→32 位数值转换	5
FC	<u>RMOV</u>	—	—	✓	32→16 位数值转换	5
FC	<u>GRY</u>	<u>DGRY</u>	—	✓	BIN→GRY 码变换	5
FC	<u>GBIN</u>	<u>DGBIN</u>	—	✓	GRY 码→BIN 变换	5
FC	<u>NEG</u>	<u>DNEG</u>	—	✓	取负数 (取 2 的补码)	3
FC	—	<u>FNEG</u>	—	✓	单精度浮点数正负符号反相	3
FC	—	<u>FBCD</u>	—	✓	单精度浮点数→十进浮点数	5
FC	—	<u>FBIN</u>	—	✓	十进浮点数→单精度浮点数	5
FC	<u>BKBCD</u>	—	—	✓	连续区块 BIN→BCD 变换	7
FC	<u>BKBIN</u>	—	—	✓	连续区块 BCD→BIN 变换	7
FC	<u>SCAL</u>	—	—	✓	比例运算	9
FC	<u>SCLP</u>	<u>DSCLP</u>	—	✓	参数型比例运算	9
FC	<u>LINE</u>	<u>DLINE</u>	—	✓	COLUMN to LINE	7
FC	<u>COLM</u>	<u>DCOLM</u>	—	✓	LINE to COLUMN	7

FB/FC	指令码			操作数				功能						
FC	D*	BCD	P	S, D				BIN→BCD 变换						

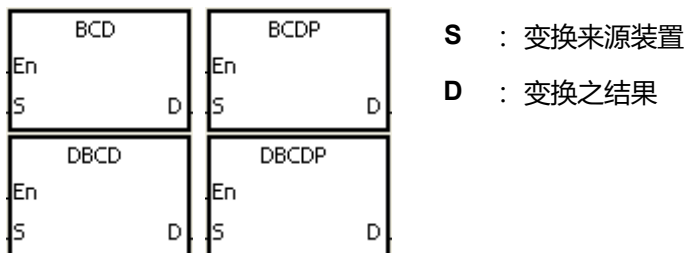
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●			○	
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：

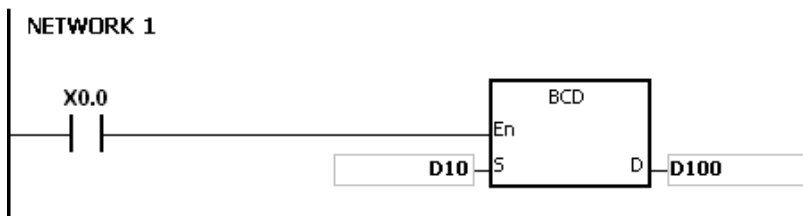


指令说明：

1. 数据源 S 的内容 (BIN 值) 作 BCD 的转换, 存于 D。
2. DBCD 才可以使用 HC 装置。
3. PLC 内的四则运算及 INC、DEC 指令都是以 BIN 方式来执行。所以在应用方面, 当要看到 10 进制数值的显示器时, 用 BCD 转换即可将 BIN 值变为 BCD 值输出。

程序范例：

当 X0.0=ON 时, D10 的 BIN 值被转换成 BCD 值后, 将结果存于 D100 当中。



若 D10=16#04D2=1234, 则执行结果 D100=16#1234.

补充说明：

1. 在 BCD 转换结果若超过 0~9,999，指令不执行，SM0=ON，错误码 SR0=16#200D。（BCD 值以 HEX 表示有任一位数不在 0~9 的范围内）。
2. 在 DBCD 转换结果若超过 0~99,999,999，指令不执行，SM0=ON，错误码 SR0=16#200D。（BCD 值以 HEX 表示有任一位数不在 0~9 的范围内）。

FB/FC	指令码			操作数				功能						
FC	D*	BIN	P	S, D				BCD→BIN 变换						

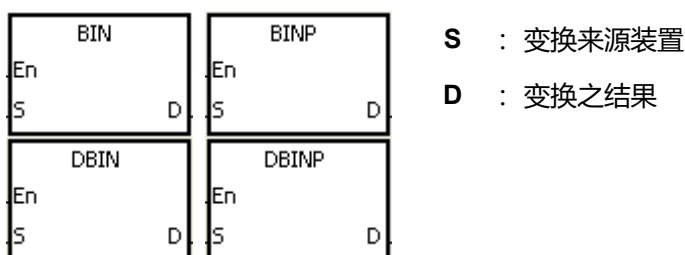
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●			○	
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：

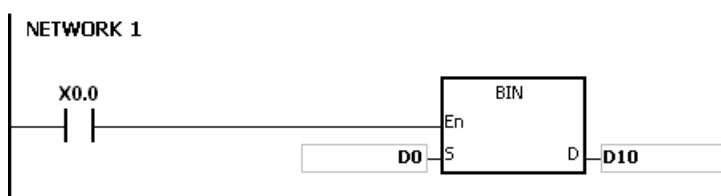


指令说明：

1. 数据源 S 的内容 (BCD : 0~9,999) , DBCD (0~99,999,999) 作 BIN 的转换, 存于 D。
2. 数据源 S 的内容有效数值范围 : BCD (0~9,999) , DBCD (0~99,999,999) 。
3. 32 位指令才可以使用 HC 装置。
4. 常数 K、16#会自动转换成 BIN, 故不需运用此指令。

程序范例：

当 X0.0=ON 时, D0 的 BCD 值被转换成 BIN 值后, 将结果存于 D10 中。

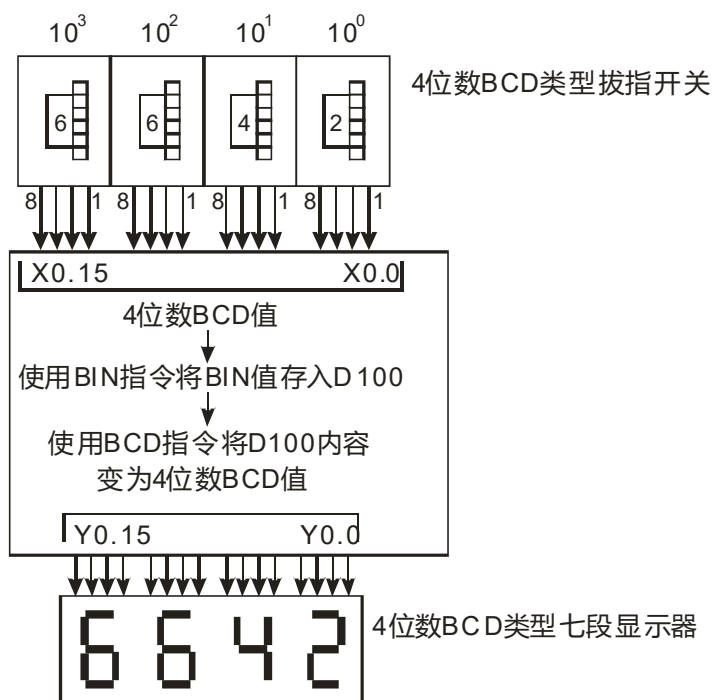
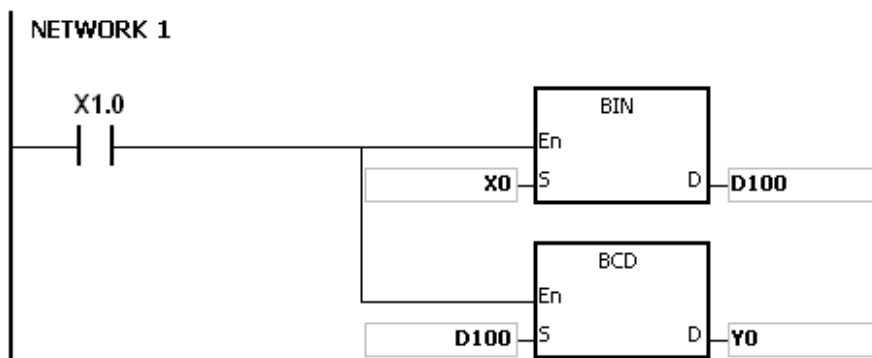


补充说明：

1. 当 S 的数据内容并非为 BCD 值 (以 HEX 表示有任一数位不在 0~9 的范围内) , 则将会产生运算错误, SM0=ON, 错误码 SR0=16#200D。

2. BCD 与 BIN 指令应用说明：

- 当 PLC 要从外界读取一个 BCD 类型指拨开关时,就必须使用 BIN 指令先将读取到的数据转换成 BIN 值再储存在 PLC 内。
- 当 PLC 要将内部储存的数据经由外界一个 BCD 类型的 7 段显示器显示出来时,就必须使用 BCD 指令先将要显示的内部数据转换成 BCD 值再送到 7 段显示器。
- 当 X1.0=ON 时,将 X0.0~X0.15 BCD 值转换成 BIN 值传送到 D100,再将 D100 的 BIN 值转换成 BCD 值传送到 Y0.0~Y0.15。



FB/FC	指令码			操作数				功能						
FC	D*	FLT	P	S, D				BIN 整数→单精度浮点数变换						

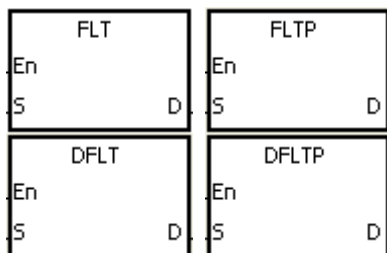
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D			●/●*					●/●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



S : 变换来源装置
D : 存放变换结果的装置

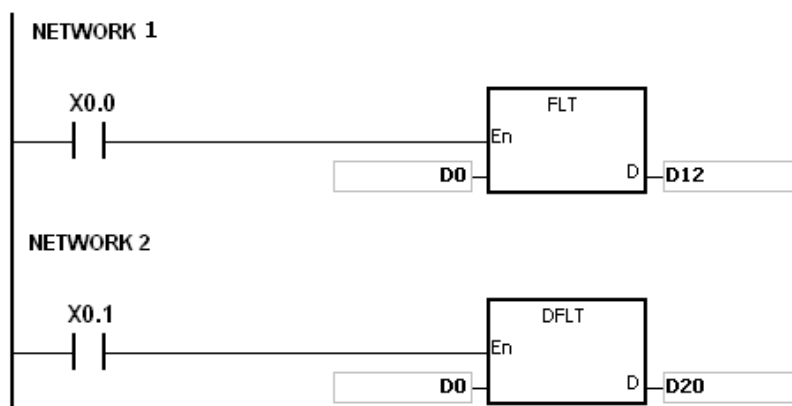
指令说明：

1. 将 BIN 整数变换成单精度浮点数。
2. FLT 中 S 不可使用 HC 装置。
3. FLT 中 S 变换来源装置占用 1 个寄存器，D 存放变换结果的装置占用 2 个寄存器。
4. DFLT 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果的装置占用 2 个寄存器。
 - 若转换结果的绝对值大于可表示的最大浮点值，则进位标号 SM602=ON，D 存放最大浮点值。
 - 若转换结果的绝对值小于可表示的最小浮点值，则借位标号 SM601=ON，D 存放最小浮点值。
 - 若转换结果为 0，则零标号 SM600=ON。

程序范例一：

1. 当 X0.0=ON 时，将 D0 (内为 BIN 整数) 变换成单精度浮点数存放于 (D13、D12)。
2. 当 X0.1=ON 时，将 D1、D0 (内为 BIN 整数) 变换成单精度浮点数存放于 (D21、D20)。
3. 若 D0=10，则 X0.0=ON，转换后单精度浮点数为 16#41200000，存于 32 位寄存器 (D13、D12) 内。

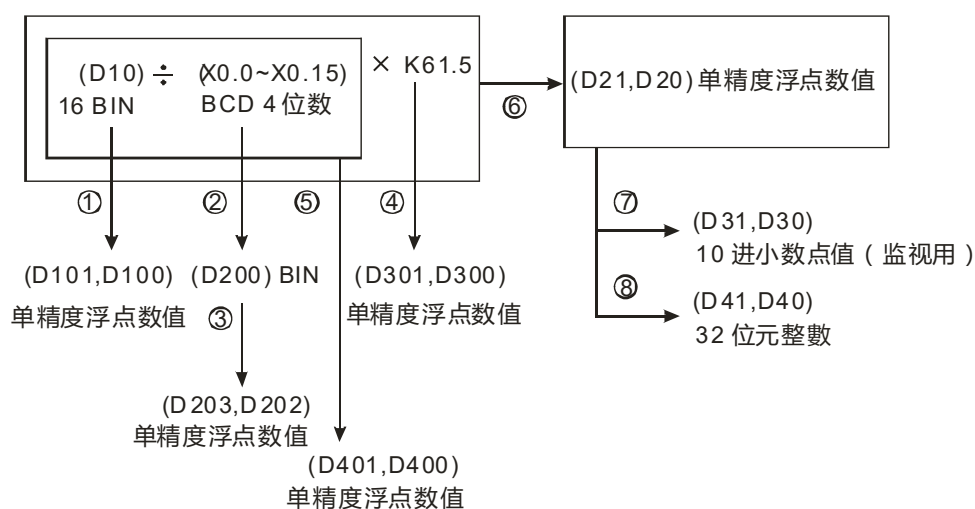
4. 若 32 位寄存器 (D1、D0) =100,000，则 X0.1=ON，转换后单精度浮点数为 16#47C35000，存于 32 位寄存器 (D21、D20) 内。

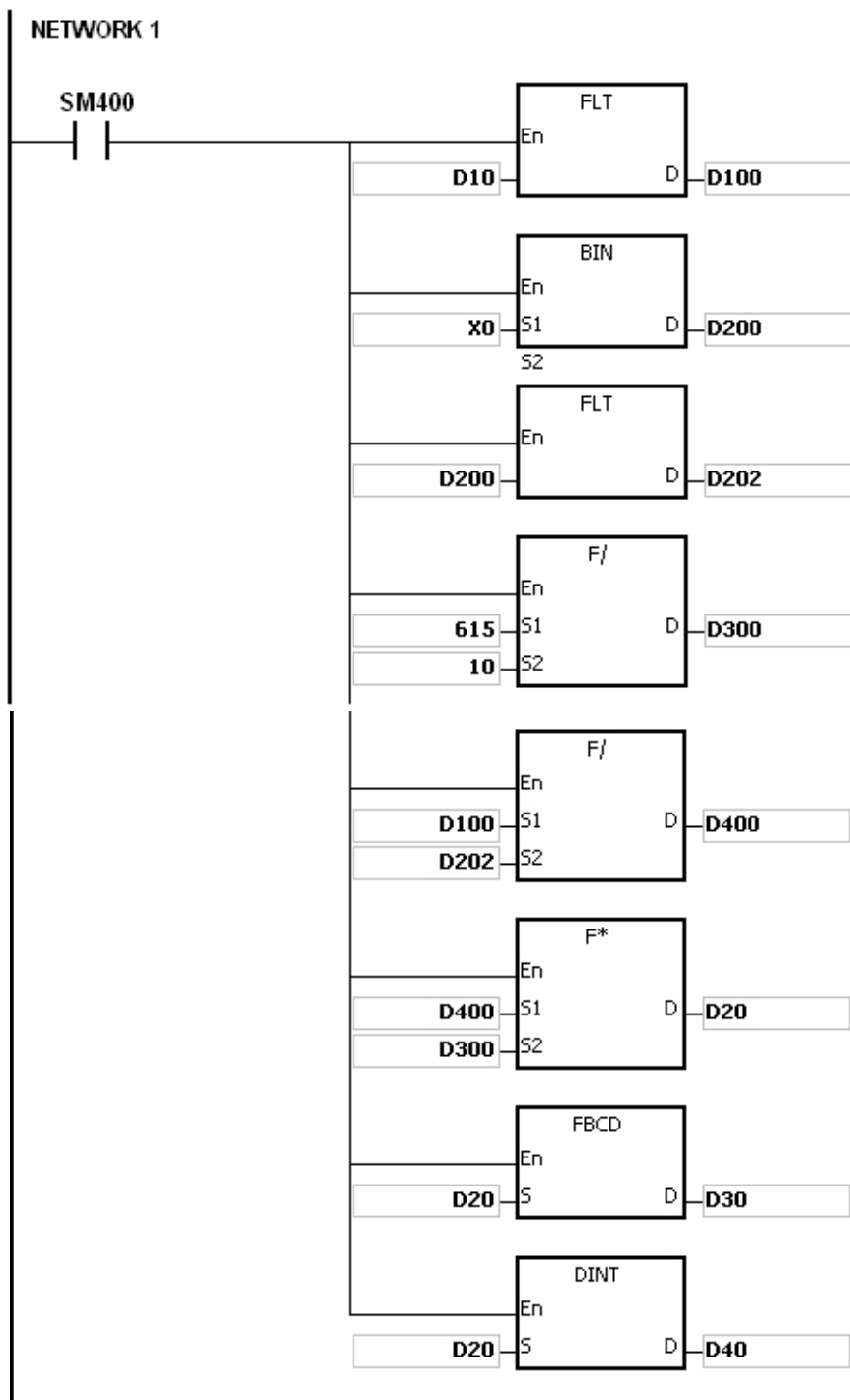


程序范例二：

使用应用指令来完成下列的算式。

1. 将 D10 (内为 BIN 整数) 变换成单精度浮点数存放于 (D101、D100) 。
2. 将 X0.0~X0.15 (BCD 值) 变换成 D200 (BIN 值) 。
3. 将 D200 (内为 BIN 整数) 变换成单精度浮点数存放于 (D203、D202) 。
4. 将常数 615÷10 结果存于 D301、D300 (单精度浮点数) 。
5. 单精度浮点数除法 (D101、D100) ÷ (D203、D202) 结果存于 D401、D400 (单精度浮点数) 。
6. 单精度浮点数乘法 (D401、D400) × (D301、D300) 结果存于 D21、D20 (单精度浮点数) 。
7. 单精度浮点数 D21、D20 变换成 10 进小数点值 D31、D30。
8. 单精度浮点数 D21、D20 变换成 BIN 整数 D41、D40。

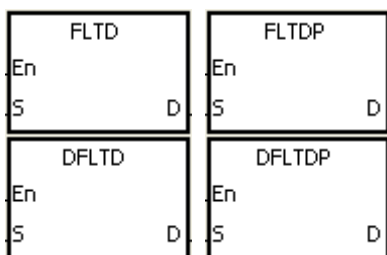




3

FB/FC	指令码			操作数							功能							
FC	D*	FLTD	P	S, D							BIN 整数转换双精度浮点数							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S		●	●*				●	●*										
D				●/●*					●/●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●	○	●				○	
D	●	●			●	●	●	●	●		●	○	●					
				脉冲执行型				16 位指令				32 位指令						
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机						

图形：



S : 变换来源装置

D : 存放变换结果的装置

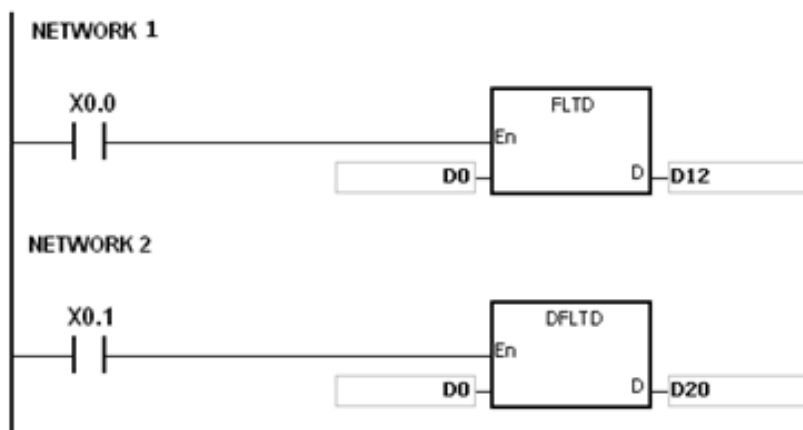
指令说明：

1. 当该指令执行时，将 BIN 整数变换成双精度浮点数。
2. FLTD 中 S 不可使用 HC 装置。
3. FLTD 中 S 变换来源装置占用 1 个寄存器，D 存放变换结果的装置占用 4 个寄存器。
4. DFLTD 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果的装置占用 4 个寄存器。
5. 若转换结果的绝对值大于可表示的最大浮点值，则进位标号 SM602=ON，D 存放最大浮点值。
6. 若转换结果的绝对值小于可表示的最小浮点值，则借位标号 SM601=ON，D 存放最小浮点值。
7. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

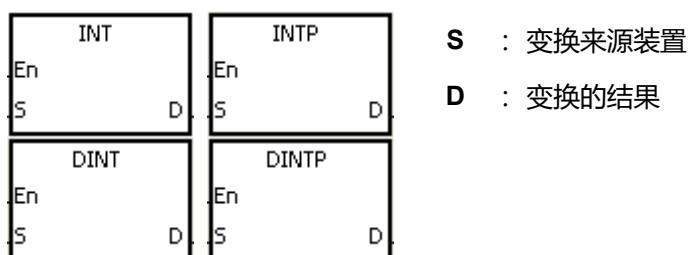
1. 当 X0.0=ON 时，将 16 位 BIN 整数 D0 变换成双精度浮点数存放于 (D15、D14、D13、D12)。
2. 当 X0.1=ON 时，将 32 位 BIN 整数 (D1、D0) 变换成双精度浮点数存放于 (D23、D22、D21、D20)。
3. 若 D0=10，则 X0.0=ON，转换后双精度浮点数为 16#4024000000000000，存于 64 位寄存器 (D15、D14、D13、D12)。

4. 若 32 位寄存器 (D1、 D0) =100,000 , 则 X0.1=ON , 转换后双精度浮点数为 16#40F86A0000000000 , 存于 64 位寄存器 (D23、 D22、 D21、 D20) 。



FB/FC	指令码			操作数				功能									
FC	D*	INT	P	S, D				单精度浮点数→BIN 整数变换									
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S			●/●*							●/●*							
D		●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●			●	○	●				○
D	●	●			●	●	●	●			●	○	●				
				脉冲执行型				16 位指令				32 位指令					
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机					

图形：

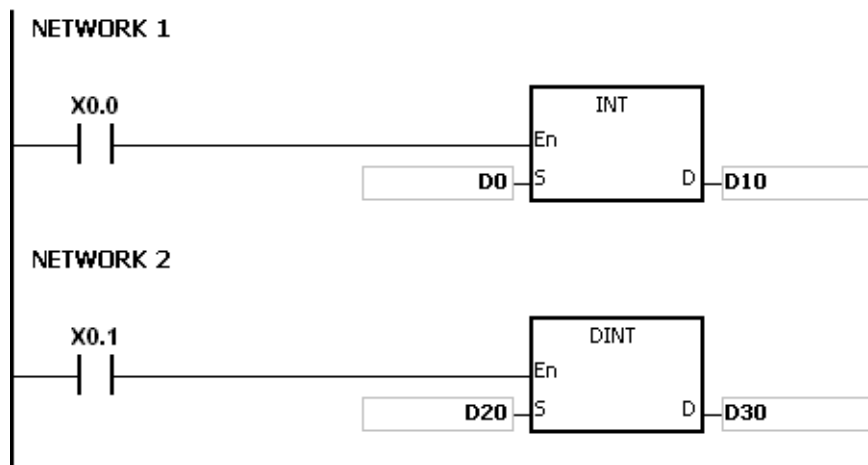


指令说明：

1. S 所指定的寄存器内容以单精度浮点数类型被变换成 BIN 整数暂存于 D 所指定的寄存器当中，BIN 整数浮点数被舍弃。
2. INT 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果的装置占用 1 个寄存器。
3. DINT 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果的装置占用 2 个寄存器。
4. INT 中 D 不可使用 HC 装置。
5. 本指令的动作与 FLT 指令刚好相反。
6. 变换结果若为 0 时，零标志 SM600=ON。
7. 变换结果有浮点数被舍弃时，借位标志 SM601=ON。
8. 变换结果若超出下列范围时（溢位），进位标志 SM602=ON。
9. 16 位指令：-32,768~32,767
10. 32 位指令：-2,147,483,648~2,147,483,647

程序范例：

1. 当 X0.0=ON 时，将单精度浮点数 (D1 , D0) 转换成 BIN 整数将结果存放至 D10 当中，BIN 整数浮点数被舍弃。
2. 当 X0.1=ON 时，将单精度浮点数 (D21 , D20) 转换成 BIN 整数将结果存放至 (D31 , D30) 当中，BIN 整数浮点数被舍弃。

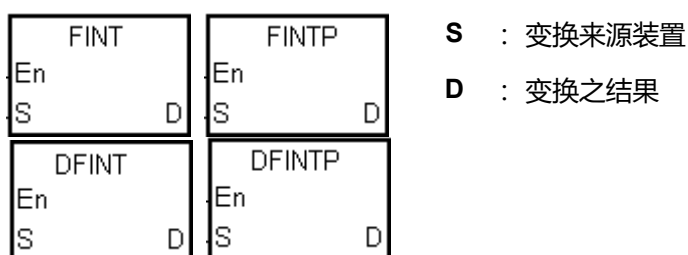


补充说明：

当 S 内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数					功能									
FC	D*	FINT	P	S, D					双精度浮点数转换 BIN 整数									
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S				●/●*							●/●*							
D		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●	○	●				○	
D	●	●			●	●	●	●	●		●	○	●					
				脉冲执行型				32 位指令				64 位指令						
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机						

图形：

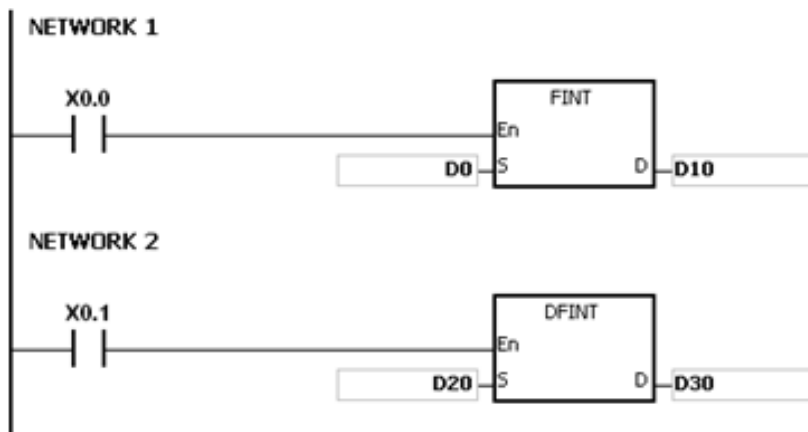


指令说明：

1. S 所指定的寄存器内容以双精度浮点数类型被转换成 BIN 整数暂存于 D 所指定的寄存器当中，BIN 整数浮点数被舍弃。
2. FINT 中 S 变换来源装置占用 4 个寄存器，D 存放变换结果的装置占用 1 个寄存器。
3. DFINT 中 S 变换来源装置占用 4 个寄存器，D 存放变换结果的装置占用 2 个寄存器。
4. FINT 跟 FLTP 中 D 不可使用 HC 装置。
5. 本指令的动作与 FLTD 指令刚好相反。
6. 变换结果若为 0 时，零标志 SM600=ON。
7. 变换结果有浮点数被舍弃时，借位标志 SM601=ON。
8. 变换结果若超出下列范围时（溢位），进位标志 SM602=ON。
32 位浮点数指令：-32,768~32,767
64 位浮点数指令：-2,147,483,648~2,147,483,647

程序范例：

1. 当 X0.0=ON 时，将双精度浮点数 (D3 , D2 , D1 , D0) 变换成 BIN 整数将结果存放至 D10 当中，BIN 整数浮点数被舍弃。
2. 当 X0.1=ON 时，将双精度浮点数 (D23 , D22 , D21 , D20) 变换成 BIN 整数将结果存放至 (D31 , D30) 当中，BIN 整数浮点数被舍弃。



3

FB/FC	脚本		操作数							功能								
FC	FTODF		S · D							单精度浮点数→双精度浮点数转换								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S										●								
D											●							
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●	○	●				●	
D	●	●			●	●	●	●	●		●	○	●					
							脉波执行型	32 位指令					64 位指令					
							-	AH 运动控制主机										

图形：

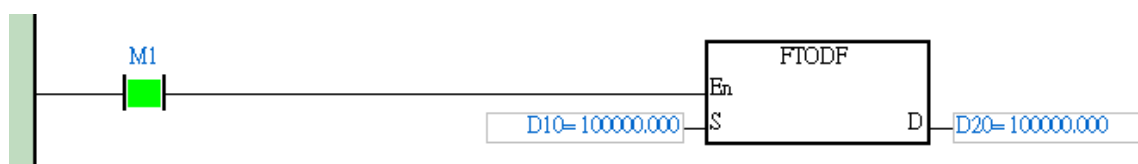


指令说明：

1. 将单精度浮点数变换 D 成双精度浮点数。
2. FTODF 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果的装置占用 4 个寄存器。
 - 若转换结果为 0，则零旗号 SM600=ON。

程序范例：

1. 当 M1=ON 时，将单精度浮点数 (D10、D11) 变换成双精度浮点数存放于 (D20、D21、D22、D23)。
2. 若单精度浮点数 32 位缓存器 (D10、D11) =100000.0，则 M1=ON，转换后双精度浮点数存于 64 位缓存器 (D20、D21、D22、D23)。



FB/FC	脚本		操作数				功能					
FC	DFTOF		S · D				双精度浮点数→单精度浮点数转换					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S											●			
D										●				

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●				
D	●	●			●	●	●	●	●		●	○	●				

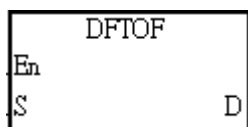
脉波执行型	32 位指令	64 位指令
-	-	AH 运动控制主机

3

图形：

S : 变换来源装置

D : 变换的结果

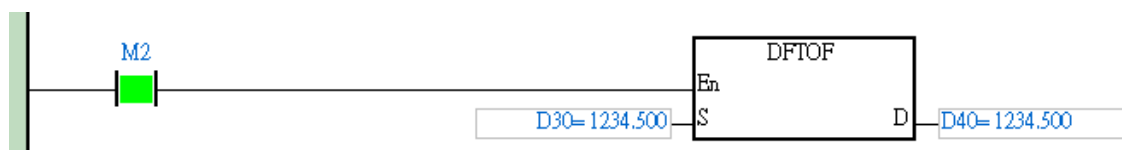


指令说明：

1. 将双精度浮点数变换成单精度浮点数。
2. DFTOF 中 **S** 变换来源装置占用 4 个寄存器，**D** 存放变换结果之装置占用 2 个寄存器。
 - 若转换结果的绝对值大于可表示的最大浮点值，则进位旗号 SM602=ON，**D** 存放最大浮点值。
 - 若转换结果的绝对值小于可表示的最小浮点值，则借位旗号 SM601=ON，**D** 存放最小浮点值。
 - 若转换结果为 0，则零旗号 SM600=ON。

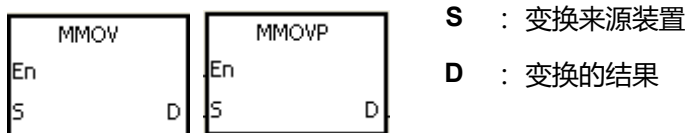
程序范例：

1. 当 M1=ON 时，将双精度浮点数 (D40、D41、42、43) 变换成单精度浮点数存放于 (D30、D31)。
2. 若双精度浮点数 64 位寄存器 (D40、D41、42、43) =1234.5，则 M1=ON，转换后单精度浮点数存于 32 位寄存器 (D30、D31)。



FB/FC	指令码				操作数						功能						
FC		MMOV	P		S, D						16→32 位数值转换						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●					●										
D			●					●									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型				16 位指令				32 位指令									
AH 运动控制主机				AH 运动控制主机				-									

图形：

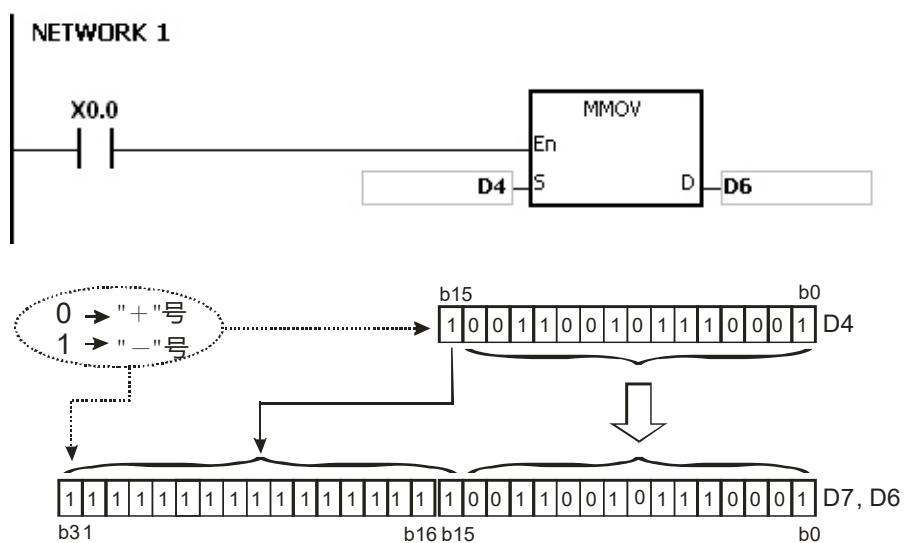


指令说明：

将 16 位装置 **S** 中的数据传送到 32 位的装置 **D** 中，其中指定的符号位被重复的复制存放在目的地。

程序范例：

当 X0.0 为 0N 时，D4 的 b15 位数据传送到 (D7, D6) 的 b15 到 b31 位，变成负数 (和 D4 的一样)。



FB/FC	指令码		操作数				功能								
FC		RMOV	P	S, D				32→16 位数值转换							

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S			●					●						
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



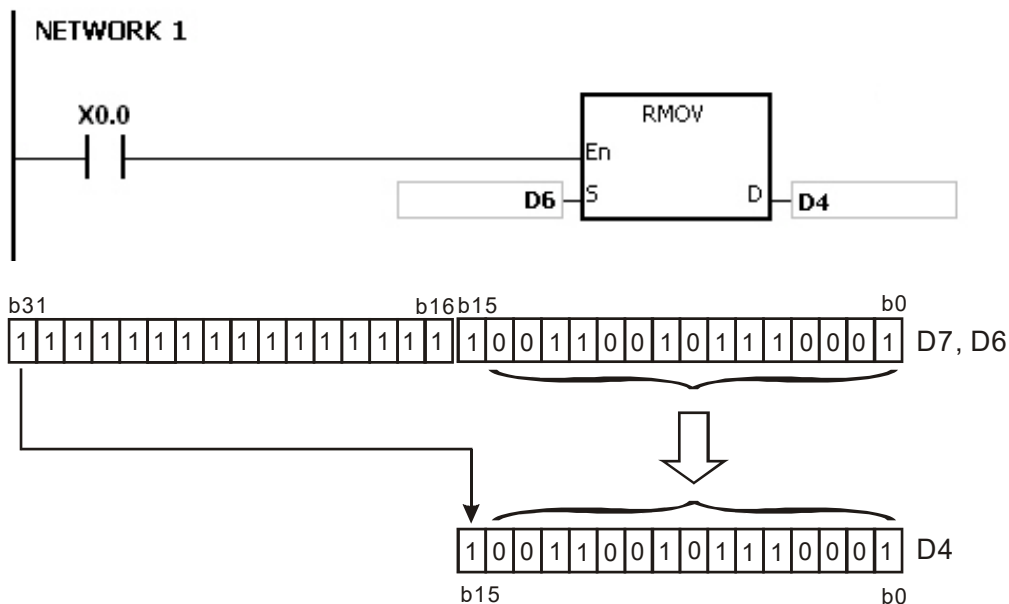
S : 变换来源装置
D : 变换的结果

指令说明：

将 32 位装置 S 中的数据传送到 16 位的装置 D 中，其中指定的符号位被保留。

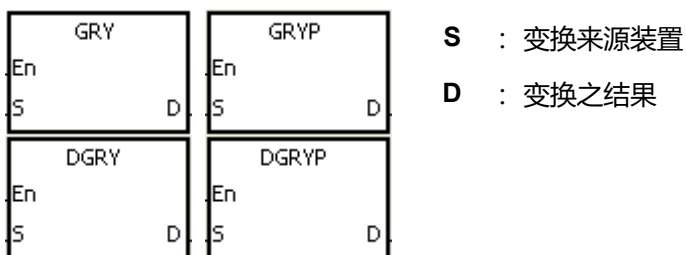
程序范例：

当 X0.0 为 ON 时，S 中最高位 (D7 : b31) 被传送到 D 中最高位 (D4 : b15) 中，其它位 (b0~b14) 则对应传送，而 b15~b30 被忽略未被传送。



FB/FC	指令码			操作数					功能								
FC	D*	GRY	P	S, D					BIN→GRAY 码变换								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●	●*				●	●*									
D		●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●			●	○	●	○	○		
D	●	●			●	●	●	●			●	○	●				
脉冲执行型		16 位指令				32 位指令											
AH 运动控制主机		AH 运动控制主机				AH 运动控制主机											

图形：

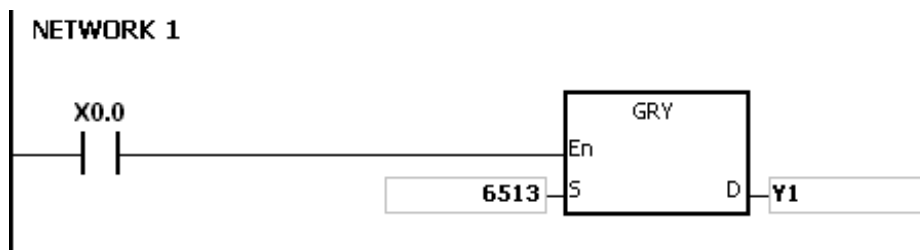


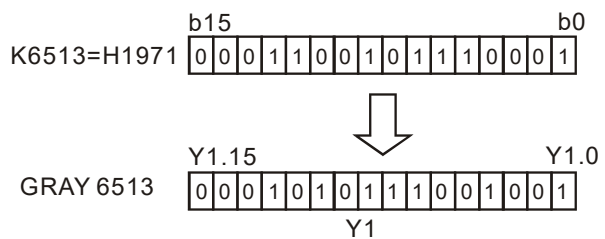
指令说明：

1. 将 S 所指定装置的内容值 (BIN 值) 变换格雷码 (GRAY CODE) 后存放于 D 所指定的装置中。
2. DGRY 才可使用 HC 装置。
3. S 的有效范围如下所示：
16 位指令：0~32,767
32 位指令：0~2,147,483,647

程序范例：

当 X0.0=ON 时，将常数 6513 变换格雷码 (GRAY CODE) 后存放于 Y1.0~Y1.15。



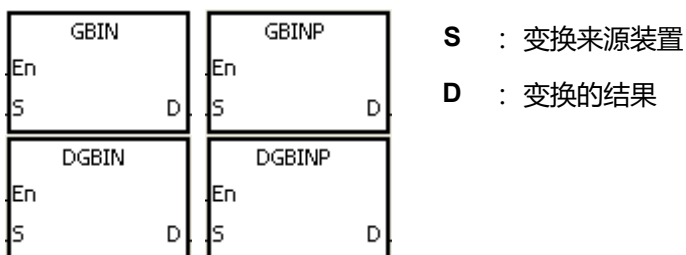


补充说明：

S 的内容值小于 0，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数					功能								
FC	D	GBIN	P	S, D					GRAY 码→BIN 变换								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●	●*				●	●*									
D		●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				
				脉冲执行型			16 位指令				32 位指令						
				AH 运动控制主机			AH 运动控制主机				AH 运动控制主机						

图形：

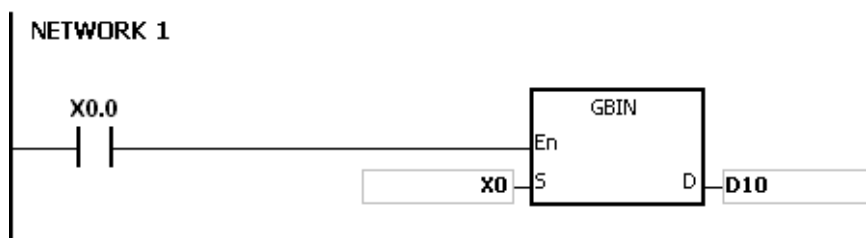


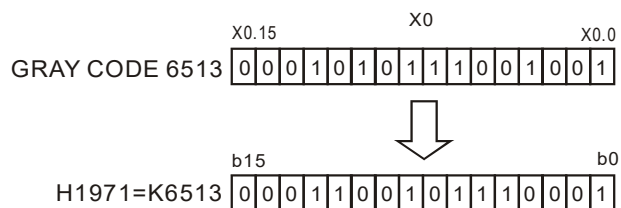
指令说明：

1. 将 S 所指定装置的内容值（格雷码（GRAY CODE））变换成 BIN 值后存放到 D 所指定之装置中。
2. 本指令将连接于 PLC 输入端的绝对位置型编码器（此编码器的输出值通常是格雷码）的内容变换成 BIN 值存放到指定的寄存器当中。
3. S 的有效范围如下所示：
16 位指令：0~32,767
32 位指令：0~2,147,483,647

程序范例：

当 X0.0 时，将 X0.0~X0.15 输入点所连接的绝对位置型编码器其格雷码（GRAY CODE）变换成 BIN 值后存放到 D10 中。





补充说明：

S 的内容值小于 0，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数						功能								
FC	D*	NEG	P	D						2 的补码								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
D		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
D	●	●			●	●	●	●	●		●	○	●					

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



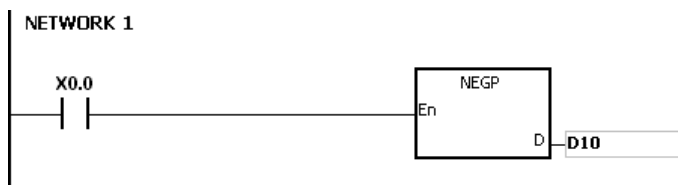
D：欲取 2 的补码的装置

指令说明：

1. 本指令可将负数的 BIN 值转换成绝对值。
2. DNEG 才可使用 HC 装置。
3. 本指令一般都是使用脉冲执行型指令（NEGP、DNEGP）。

程序范例一：

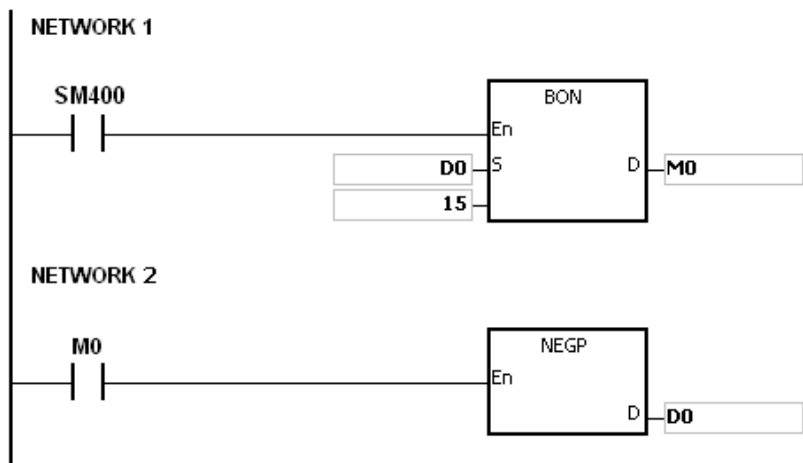
当 X0.0=OFF→ON 时，D10 内容的各位全部反相（0→1、1→0）后再加 1 存放于原寄存器 D10 当中。



程序范例二：

求负数的绝对值：

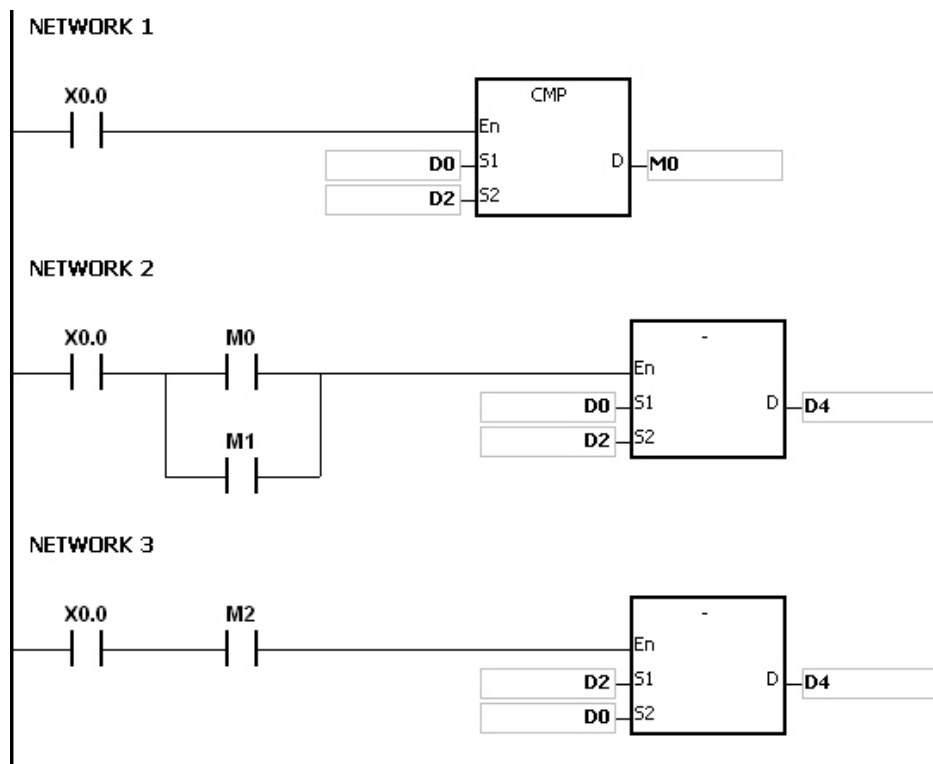
1. 当 D0 的第 15 个位为“1”时，M0=ON。（D0 表示为负数）
2. M0=ON 时，用 NEG 指令将 D0 取 2 的补码可得到其绝对值。



程序范例三：

减法运算的差取绝对值，当 X0.0=ON 时：

1. 若 $D0 > D2$ 时， $M0 = ON$ 。
2. 若 $D0 = D2$ 时， $M1 = ON$ 。
3. 若 $D0 < D2$ 时， $M2 = ON$ 。
4. 此可得 $D4$ 保持为正值。



补充说明：

数的表现及绝对值：

1. 正负数是以寄存器最上位（最左边）的位内容来表现，为“0”时，为正数、为“1”时，为负数。

2. 遇到负数时，可使用 NEG 指令将它转成绝对值。

(D0)=2

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0)=1

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0)=0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

FB/FC	指令码			操作数						功能					
FC		FNEG	P	D						单精度浮点数正负符号反相					

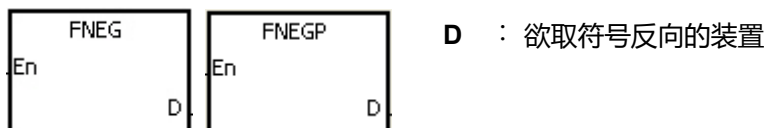
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D										●				

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	32 位指令	64 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

3



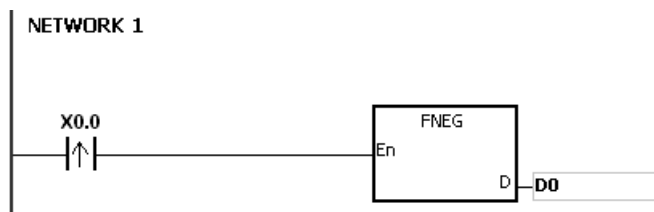
指令说明：

将装置 D 中的内容，以单精度浮点数的类型做正负符号反相。

程序范例：

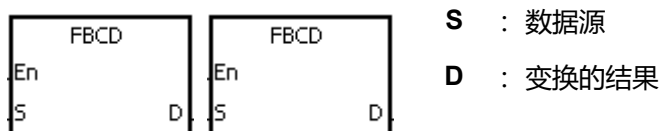
执行前 (D1、D0) =16#AE0F9000 (负值)，当 X0.0 从 OFF→ON 时，(D1、D0) 的数据以单精度浮点数做正负符号的反相运作，因此执行后 (D1、D0) =16#2E0F9000 (正值)。

执行前 (D1、D0) =16#2E0F9000 (正值)，当 X0.0 从 OFF→ON 时，(D1、D0) 的数据以单精度浮点数做正负符号的反相运作，因此执行后 (D1、D0) =16#AE0F9000 (负值)。



FB/FC	指令码			操作数						功能							
FC		FBCD	P	S, D						单精度浮点数→十进浮点数							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S										●							
D			●														
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				
				脉冲执行型			32 位指令				64 位指令						
				AH 运动控制主机			AH 运动控制主机				-						

图形：

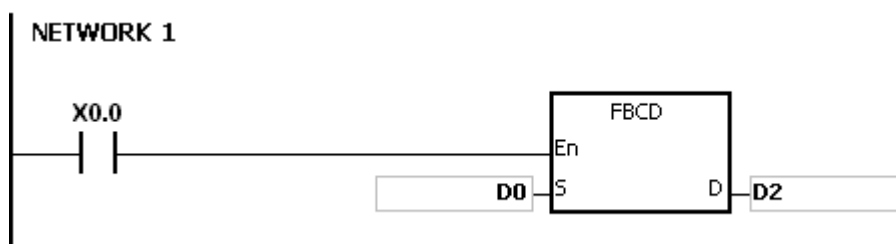


指令说明：

1. 将 **S** 所指定的寄存器中的单精度浮点数转换成十进制浮点数寄存于 **D** 所指定的寄存器当中。
2. PLC 是以单精度浮点数类型作浮点数运算的依据。FBCD 指令就是用来将单精度浮点数转换成十进制浮点数类型的专用指令。
3. 标志：SM600 为零标志，SM601 为借位标志，SM602 为进位标志。
若转换结果的绝对值大于可表示的最大浮点值，则进位标志 SM602=ON。
若转换结果的绝对值小于可表示的最小浮点值，则借位标志 SM601=ON。
若转换结果为 0，则零标志 SM600=ON。

程序范例：

当 X0.0= ON 时 (D1、D0) 内的单精度浮点数被转换成十进制浮点数存于 (D3、D2)。



2 进小数点

D 1	D 0
-----	-----

 实数23个位，指数8个位，符号1个位 1



指数 实数

10 进小数点

D 3	D 2
-----	-----

 数学式表示 $\Leftrightarrow [D2] \times 10^{[D3]}$

补充说明：

当 **S** 内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数						功能								
FC		FBIN	P	S, D						十进浮点数→单精度浮点数								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S			●															
D										●								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●	○	●				○	
D	●	●			●	●	●	●	●		●	○	●					
				脉冲执行型				32 位指令				64 位指令						
				AH 运动控制主机				AH 运动控制主机				-						

图形：

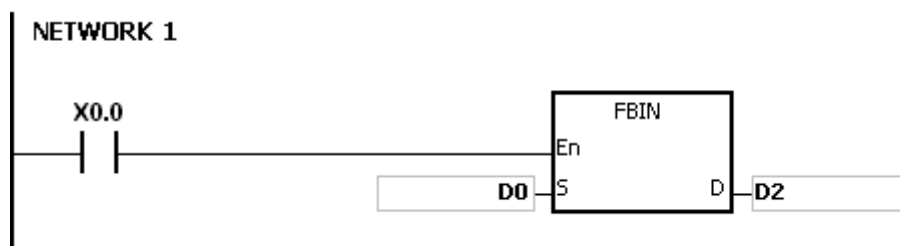


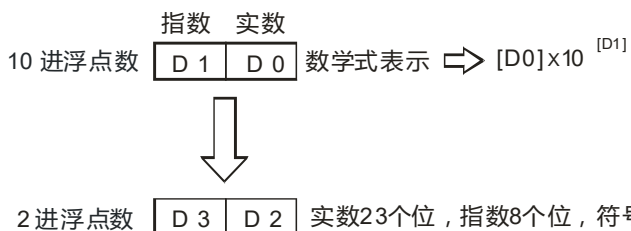
指令说明：

1. 将 **S** 所指定的寄存器中的十进制浮点数转换成单精度浮点数并寄存于 **D** 所指定的寄存器当中。
2. 例如：**S**=1234，**S**+1=3 将变换成 **S**=1.234×10⁶。
3. **D** 必须是单精度浮点数形式，**S** 和 **S**+1 中分别用十进制表示实数和指数。
4. FBIN 指令就是用来将十进浮点数转换成单精度浮点数类型的专用指令。
5. 十进浮点数实数范围为-9,999~+9,999 指数范围为-41~+35 实际PLC十进浮点数的范围为±1175×10⁻⁴¹到±3402×10⁺³⁵。若运算结果为 0，则零标志 SM600=ON。

程序范例一：

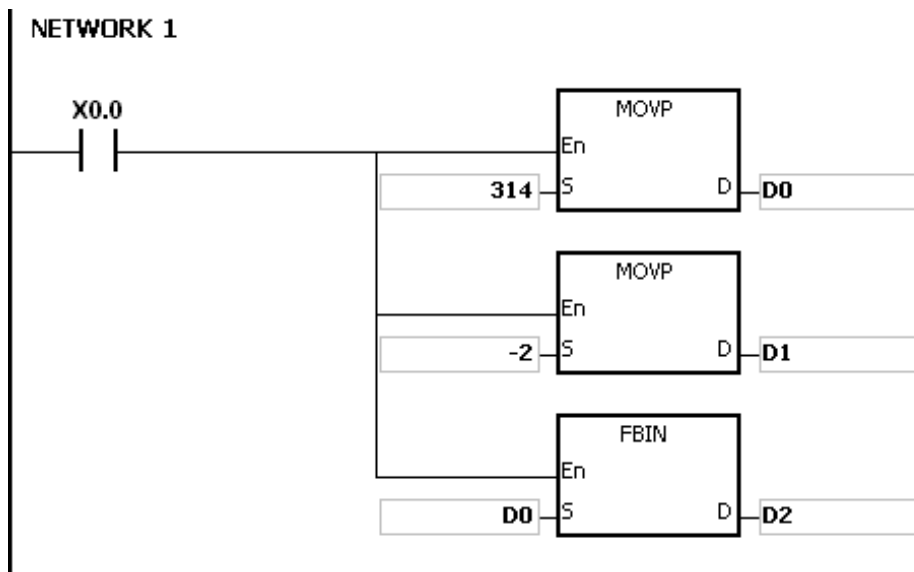
当 X0.0=ON 时，指定寄存器（D1、D0）中的十进制浮点数转换成单精度浮点数并寄存于（D3、D2）中。





程序范例二：

1. 在进行浮点数运算前必须适用 FLT 指令 BIN 整数变换成单精度浮点数，变换的前提是被变换值必须是 BIN 整数，然而，FBIN 指令可将浮点数变换成单精度浮点数。
2. 当 X0.0=ON 时，将 K314 搬移到 D0，将 K-2 搬移到 D1，组成十进制浮点数类型 ($3.14=314 \times 10^{-2}$)。



补充说明：

若 S 操作数的十进浮点数实数范围不在-9,999~+9,999 之间或指数范围不在-41~+35 之间，指令不执行，SM0=ON，错误码 SR0=16#2013。

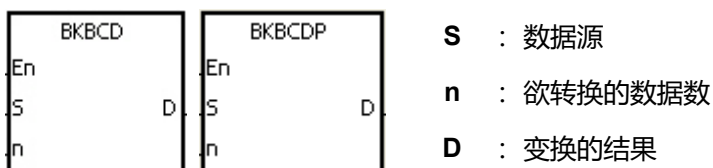
FB/FC	指令码			操作数				功能						
FC		BKBCD	P	S, n, D				连续区块 BIN→BCD 变换						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

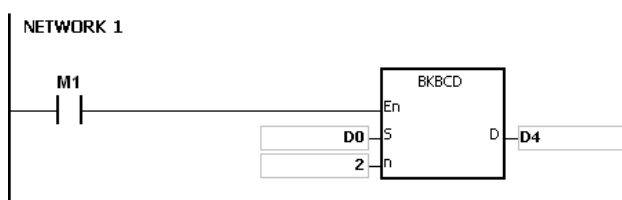


指令说明：

将 n 笔的数据源 S 的内容 (BIN 值) 作 BCD 的转换, 存于 D。n=1~256。

程序范例：

当 M1=ON 时, 将 D0、D1 连续两笔数据中的 BIN 转换成 BCD 值并存于 D4、D5 中。



补充说明：

1. 若 $n < 1$ 或 $n > 256$ 时, 指令不执行, $SM0=ON$, 错误码 $SR0=16\#200B$ 。
2. $S+n-1, D+n-1$, 装置超出范围时, 指令不执行, $SM0=ON$, 错误码 $SR0=16\#2003$ 。
3. 在 BCD 变换结果若超过 0~9,999, 指令不执行, $SM0=ON$, $SR0$ 记录错误码 16#200D。(BCD 值以 HEX 表示有任一位数不在 0~9 的范围内)。
4. 若 $S \sim S+n-1$ 的装置与 $D \sim D+n-1$ 的装置有重叠时, 指令不执行, $SM0=ON$, 错误码 $SR0=16\#200C$ 。

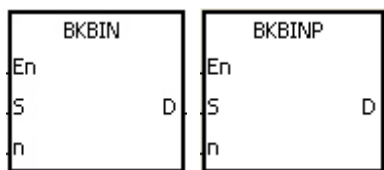
FB/FC	指令码			操作数				功能						
FC		BKBIN	P	S, n, D				连续区块 BCD→BIN 变换						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



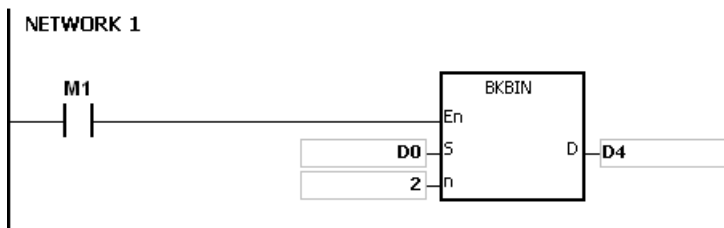
S : 数据源
n : 欲转换的数据数
D : 变换的结果

指令说明：

1. 将 n 笔的数据源 S 的内容 (BCD : 0~9,999) 作 BIN 的转换, 存于 D。
2. 数据源 S 的内容有效数值范围 : BCD (0~9,999)。
3. n=1~256。

程序范例：

当 M1=ON 时, 将 D0、D1 连续两笔数据中的 BCD 转换成 BIN 并存于 D4、D5 中。



补充说明：

1. 若 n<1 或 n>256 时, 指令不执行, SM0=ON, 错误码 SR0=16#200B。

2. **S+n-1** , **D+n-1** , 装置超出范围时, 指令不执行, SM0=ON, 错误码 SR0=16#2003。
3. 当 **S** 的数据内容并非为 BCD 值 (以 Hex 表示有任一位数不在 0~9 的范围内), 则将会产生运算错误, 指令不执行, SM0=ON, SR0 记录错误码 16#200D。
4. 若 **S~S+n-1** 的装置与 **D~D+n-1** 的装置有重叠时, 指令不执行, SM0=ON, 错误码 SR0=16#200C。

FB/FC	指令码			操作数						功能				
FC		SCAL	P	S_1, S_2, S_3, D						比例值运算				

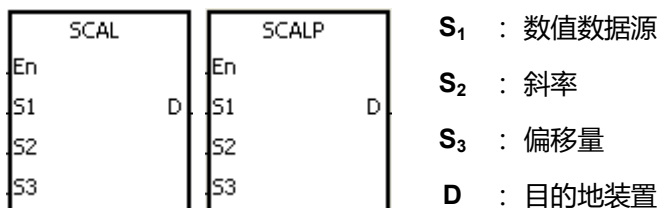
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2, S_3		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2, S_3	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



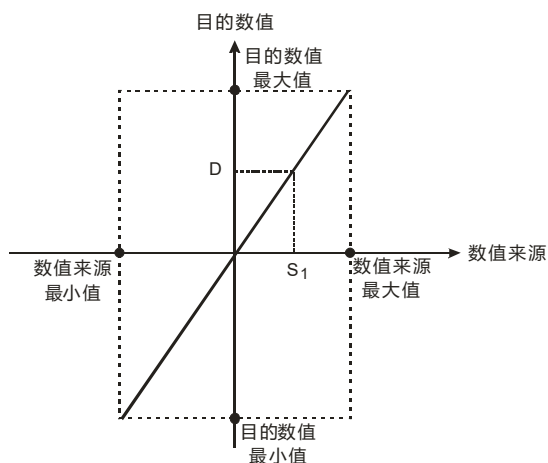
指令说明：

- 指令内部运算公式为 $D = (S_1 \times S_2) \div 1,000 + S_3$ 。
- 其中 S_2 与 S_3 的数值须由用户依下列斜率与偏移量公式先行运算，然后将小数点 4 舍 5 入后，再取 16 位的整数值输入。

斜率公式为 $S_2 = [(\text{目的数值最大值} - \text{目的数值最小值}) \div (\text{数值来源最大值} - \text{数值来源最小值})] \times 1,000$

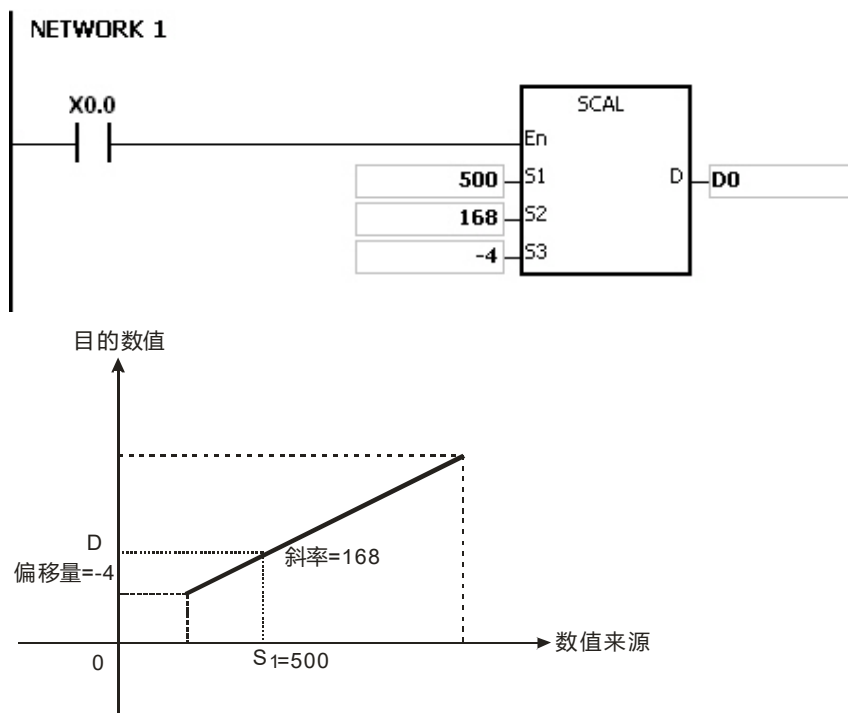
偏移量公式为 $S_3 = \text{目的数值最小值} - \text{数值来源最小值} \times S_2 \div 1,000$

其输出曲线将如下图所示：



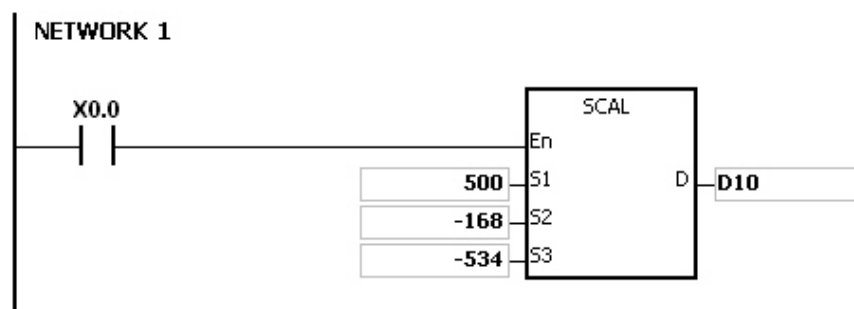
程序范例一：

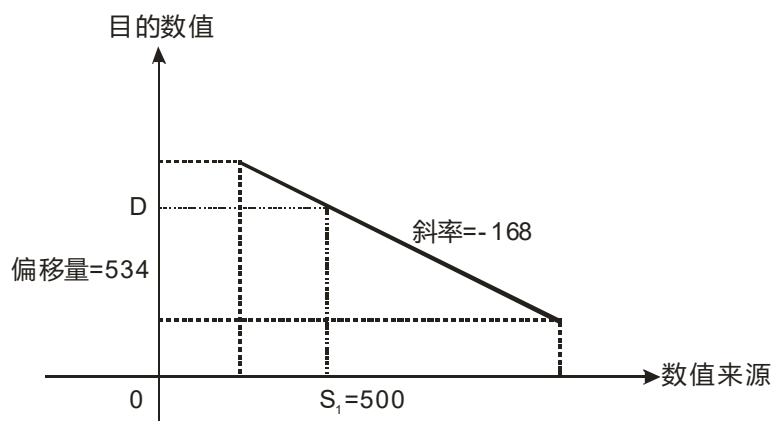
1. 已知 S_1 数值数据源为 500， S_2 斜率为 168， S_3 偏移量为 -4，当 X0.0=ON 时，SCAL 指令执行，可在 D0 得到所要求的比例值。
2. 运算方式： $D0 = (500 \times 168) \div 1,000 + (-4) = 80$



程序范例二：

1. 已知 S_1 数值数据源为 500， S_2 斜率为 -168， S_3 偏移量为 534，当 X0.0=ON 时，SCAL 指令执行，可在 D10 得到所要求的比例值。
2. 运算方式： $D10 = (500 \times -168) \div 1,000 + 534 = 450$



**补充说明：**

1. 此 SCAL 指令适用于已知斜率与偏移量，若不知斜率与偏移量建议使用 SCLP 指令来做运算。
2. 输入参数 S_2 时，其输入数值必须为 $-32,768 \sim 32,767$ 之间的数值（实际数值为 $-32,768 \sim 32,767$ ），若是 S_2 实际数值超过范围时，请改用 SCLP 指令运算。
3. 用户运用斜率换算公式时，须注意数值来源最大值，必须大于数值来源最小值，而目的数值最大值，并不限制大于目的数值最小值。
4. 若 D 的值 $> 32,767$ 则 $D = 32,767$ ，若 D 的值 $< -32,768$ 则 $D = -32,768$ 。

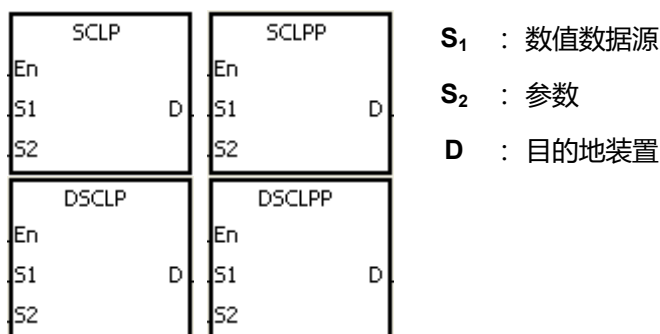
FB/FC	指令码			操作数					功能						
FC	D*	SCLP	P	S ₁ , S ₂ , D					参数型比例值运算						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



指令说明：

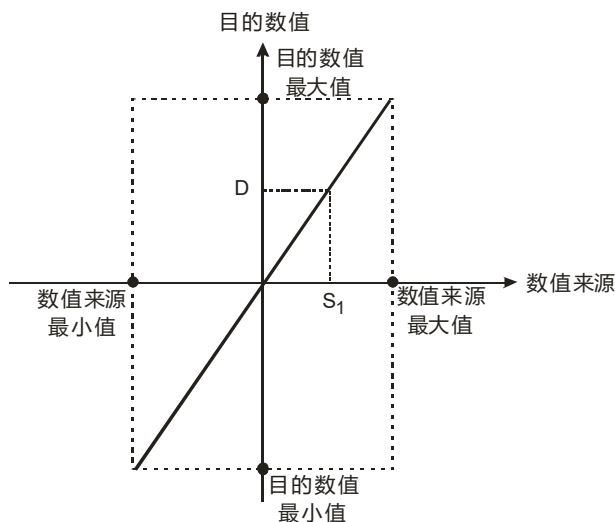
- 32 位指令才可以使用 HC 装置。
- 16 位指令 S₂ 参数设定内容如下。

装置编号	参数名称与说明	设定范围
S ₂	数值来源最大值	-32,768~32,767
S ₂ +1	数值来源最小值	-32,768~32,767
S ₂ +2	目的数值最大值	-32,768~32,767
S ₂ +3	目的数值最小值	-32,768~32,767

- 16 位指令 S₂ 操作数将连续占用 4 个装置。
- 32 位指令 S₂ 参数设定内容如下。

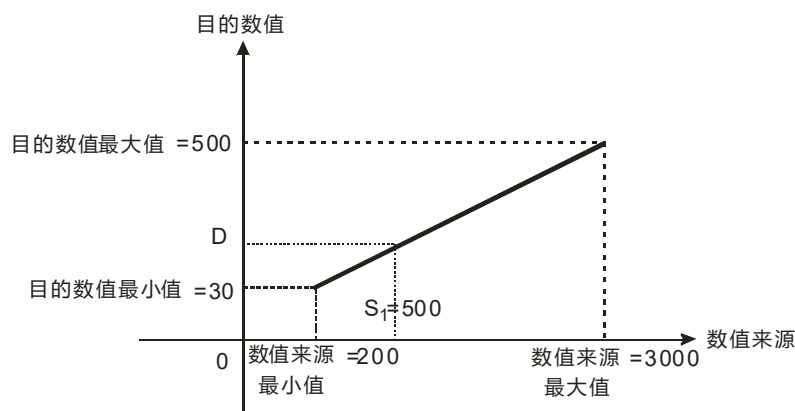
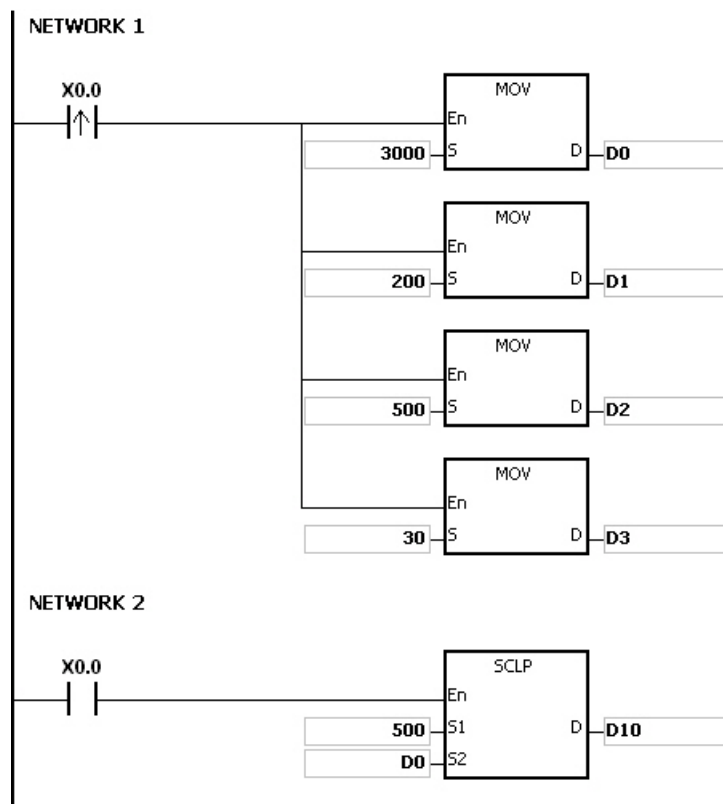
装置编号	参数名称与说明	设定范围	
		整数	整数
S_2 、 S_2+1	数值来源最大值	-2,147,483,648~ 2,147,483,647	-2,147,483,648~ 2,147,483,647
S_2+2 、 3	数值来源最小值		
S_2+4 、 5	目的数值最大值		
S_2+6 、 7	目的数值最小值		

5. 32 位指令 S_2 操作数将连续占用 8 个装置。
6. 标志 SM685 为十进制整数与单精度浮点数使用标志。在 32 位指令中若要使用浮点数做运算可以设定 SM685=ON，若要使用十进制整数做运算可以设定 SM685=OFF。
7. 指令内部运算公式为 $D=[(S_1-数值来源最小值) \times (目的数值最大值-目的数值最小值)] \div (数值来源最大值-数值来源最小值) + 目的数值最小值$ 。
8. 来源数值和目的数值运算关系：
 $y=kx+b$
 $y=目的数值(D)$
 $k=斜率 = (目的数值最大值-最小值) \div (来源数值最大值-最小值)$
 $x=来源数值(S_1)$
 $b=偏移量=目的数值最小值-来源数值最小值 \times 斜率$
 将上面的各参数带入公式 $y=kx+b$ ，即可推导出指令内部运算公式：
 $y=kx+b=D=k S_1+b=斜率 \times S_1 + 偏移量 = 斜率 \times S_1 + 目的数值最小值 - 来源数值最小值 \times 斜率 = 斜率 \times (S_1 - 来源数值最小值) + 目的数值最小值 = (S_1 - 来源数值最小值) \times (目的数值最大值 - 目的数值最小值) \div (来源数值最大值 - 来源数值最小值) + 目的数值最小值$ 。
9. 假如 $S_1 > 数值来源最大值$ ，则 $S_1 = 数值来源最大值$ ；假如 $S_1 < 数值来源最小值$ ，则 $S_1 = 数值来源最小值$ ；当输入数值与参数设定完成后，则其输出曲线将如下图所示。



程序范例一：

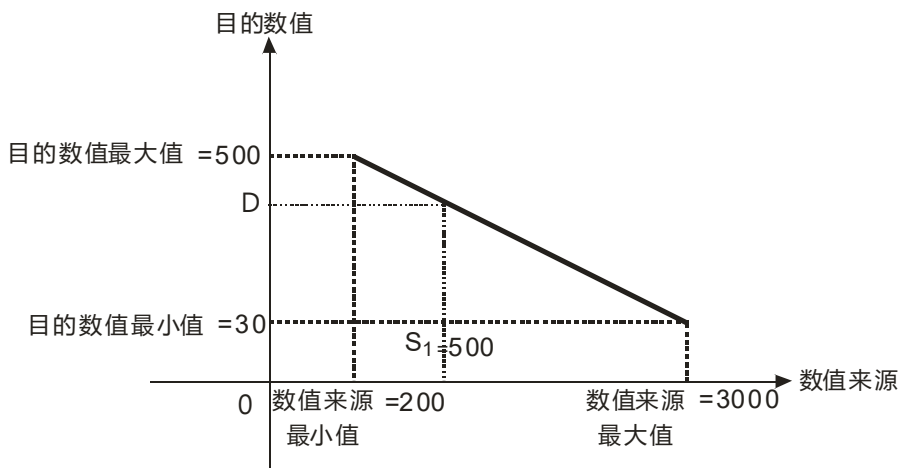
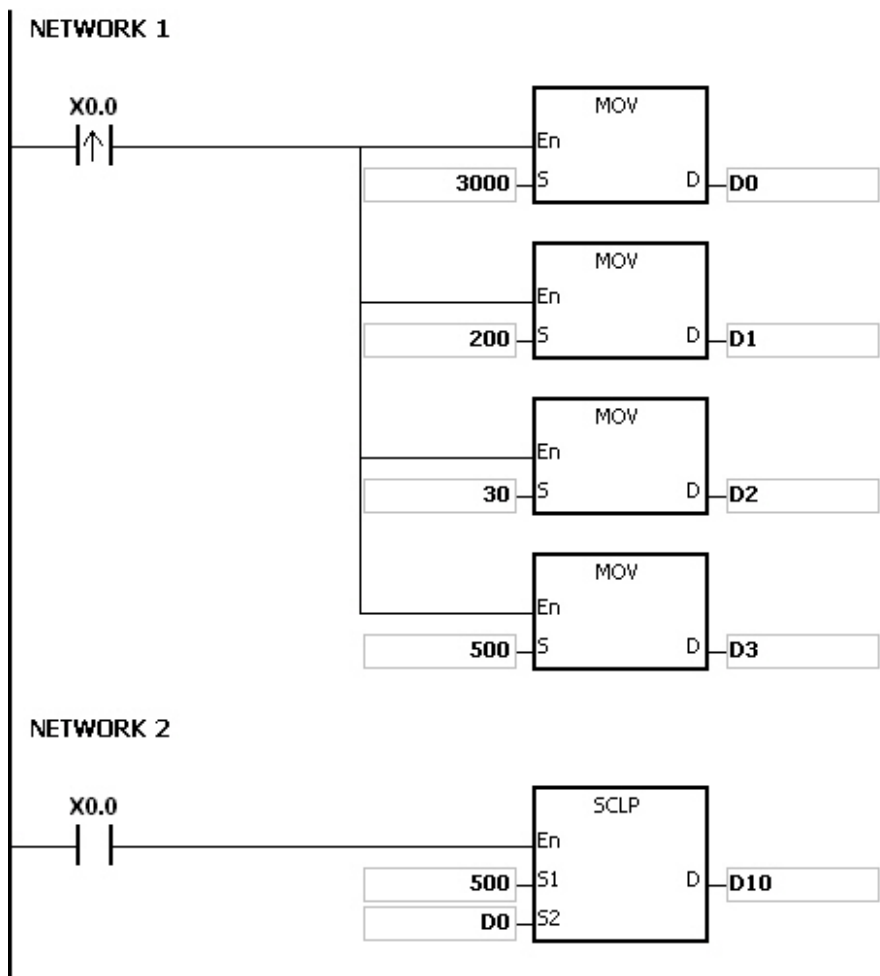
1. 已知 S_1 数值数据源为 500，数值来源最大值 $D0=3,000$ ，数值来源最小值 $D1=200$ ，目的数值最大值 $D2=500$ ，目的数值最小值 $D3=30$ 当 $X0.0=ON$ 时，SCLP 指令执行，可在 $D10$ 得到所要求的比例值。
2. 运算方式： $D10=[(500-200) \times (500-30)] \div (3,000-200) + 30=80.35$ ，取整数， $D10=80$ 。



程序范例二：

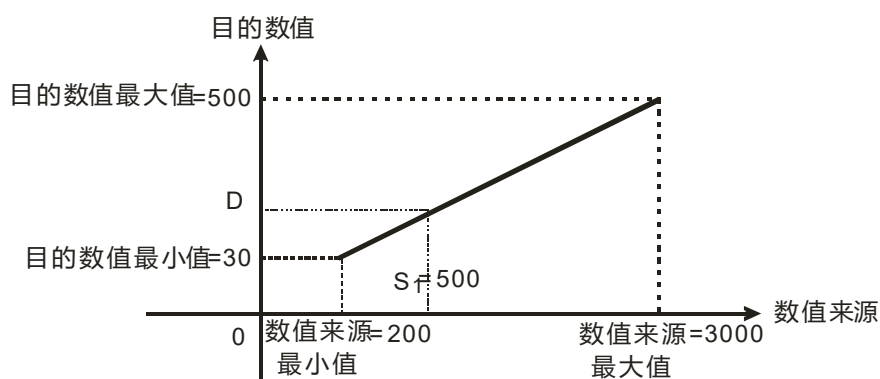
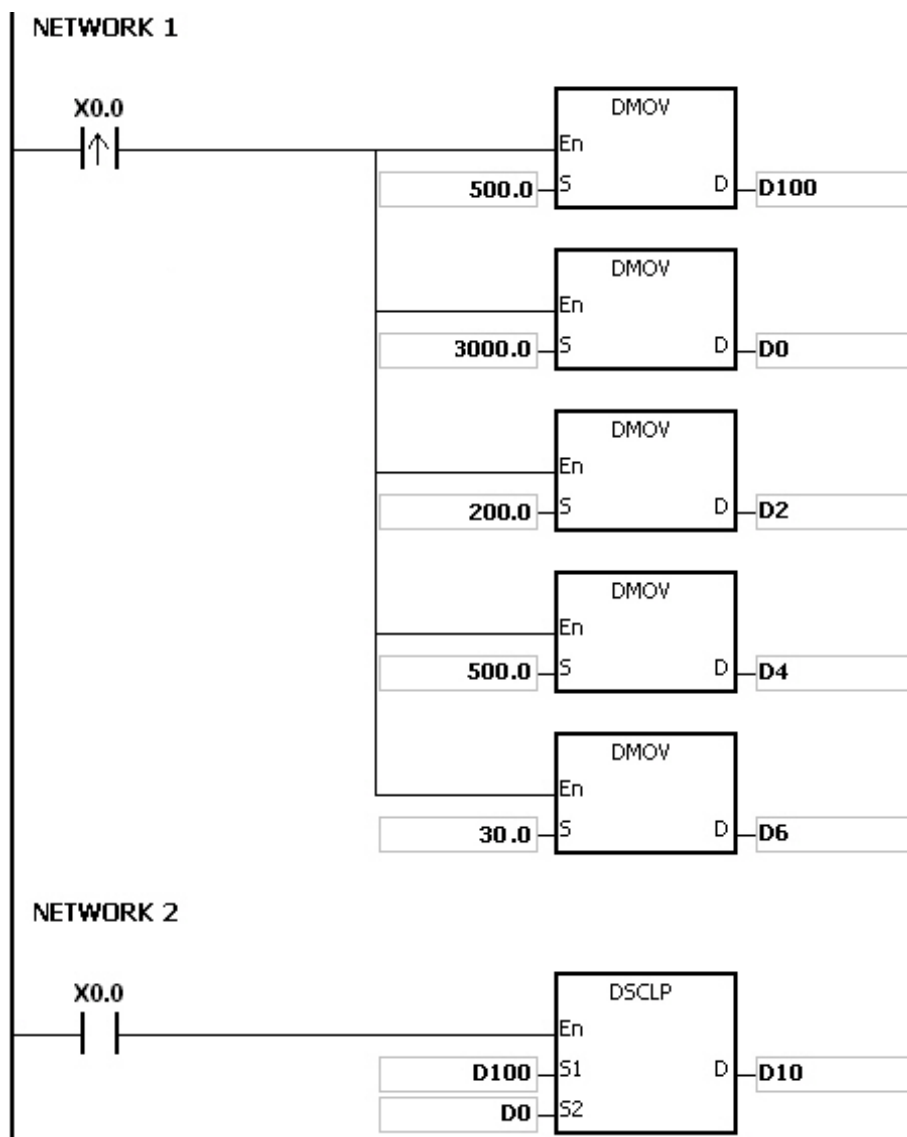
1. 已知 S_1 数值数据源为 500，数值来源最大值= $D0=3,000$ ，数值来源最小值 $D1=200$ ，目的数值最大值 $D2=30$ ，目的数值最小值 $D3=500$ 当 $X0.0=ON$ 时，SCLP 指令执行，可在 $D10$ 得到所要求的比例值。
2. 运算方式： $D10=[(500-200) \times (30-500)] \div (3,000-200) + 500=449.64$ 。四舍五入取整数， $D10=450$ 。

3



程序范例三：

1. 已知 S_1 数值数据源 D100 为 500.0，数值来源最大值 D0 为 3000.0，数值来源最小值 D2 为 200.0，目的数值最大值 D4 为 500.0，目的数值最小值 D6 为 30.0 当 X0.0=ON 时，SET SM685，使用浮点数运算且 DSCLP 指令执行，可在 D10 得到所要求的比例值。
2. 运算方式： $D10 = [(500.0 - 200.0) \times (500.0 - 30.0)] \div (3000.0 - 200.0) + 30.0 = 80.35$ ，取整数，D10=80.0。

**补充说明：**

- 16 位 S_1 操作数数值设定范围：数值来源最大值 $\geq S_1 \geq$ 数值来源最小值，-32,768~32,767 超出边界值以边界值运算。
- 32 位 S_1 整数操作数数值设定范围：数值来源最大值 $\geq S_1 \geq$ 数值来源最小值，-2,147,483,648~2,147,483,647 超出边界值以边界值运算。

3. 32 位 S_1 浮点数操作数数值设定范围：数值来源最大值 $\geq S_1 \geq$ 数值来源最小值，依单精度浮点数范围，超出边界值以边界值运算。
4. 用户运用时，须注意数值来源最大值，必须大于数值来源最小值，而目的数值最大值，并不限制大于目的数值最小值。
5. 数值来源最大值等于数值来源最小值，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2012。
6. 16 位指令 S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of WORD。
7. 32 位指令 S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of DWORD。

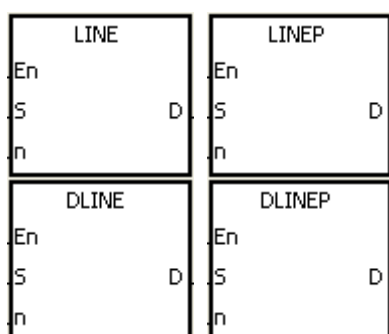
FB/FC	指令码			操作数	功能
FC	D*	LINE	P	S, n, D	COLUMN to LINE

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
n		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



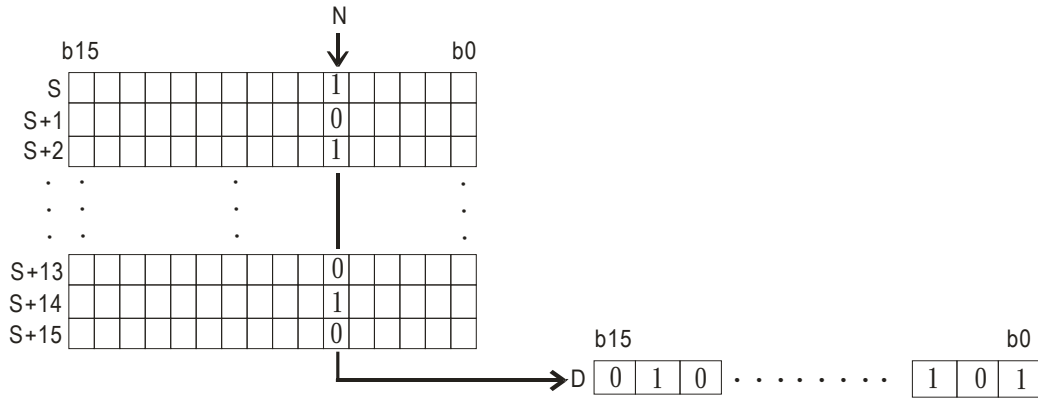
S : 数值数据源

n : 欲转换的数据数

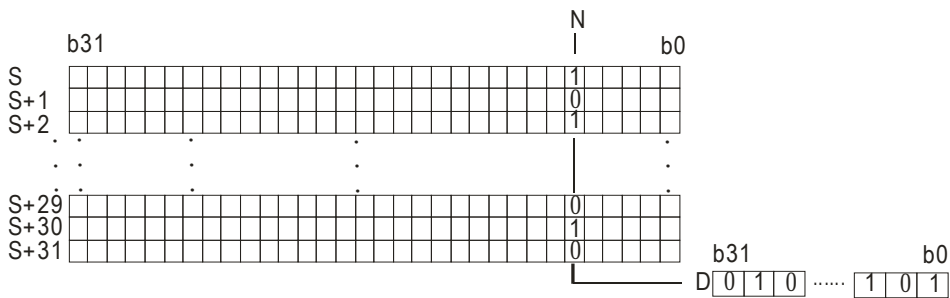
D : 目的地装置

指令说明：

- 16 位指令 S 操作数固定占用连续 16 笔数据源 S~S+15，其中每一笔为 16 位寄存器。
- 32 位指令 S 操作数固定占用连续 32 笔数据源 S~S+31，其中每一笔为 32 位寄存器。
- n 操作数表示要获取 S 操作数中每一笔的第 n 个 Bit。并在 16 位指令中限制 n=0~15，在 32 位指令中限制 n=0~31。
- 16 位指令将 16 笔的数据源 S~S+15 的第 n 个 Bit，依 S~S+15 的顺序存于 D 操作数的 b0~b15。
- 32 位指令将 32 笔的数据源 S~S+31 的第 n 个 Bit，依 S~S+31 的顺序存于 D 操作数的 b0~b31。
- 32 位指令才可使用 HC 装置。
- 以 16 位指令说明：

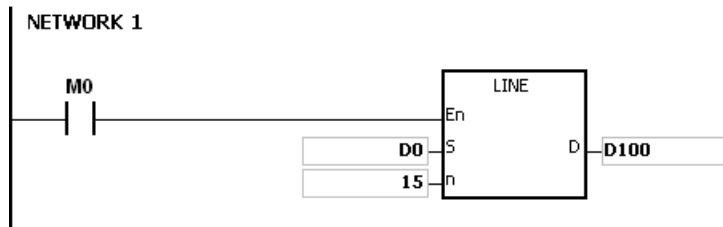


8. 以 32 位指令说明：



程序范例：

M0=ON 时，D0~D14 的第 15 个 Bit 储存于 D100 的 b0~b15 当中。



补充说明：

1. 16 位指令 S+15 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 32 位指令 S+31 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

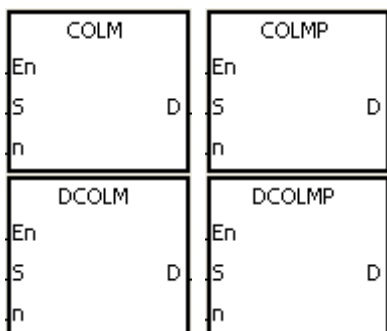
FB/FC	指令码			操作数				功能						
FC	D*	COLM	P	S, n, D				LINE to COLUMN						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
n		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

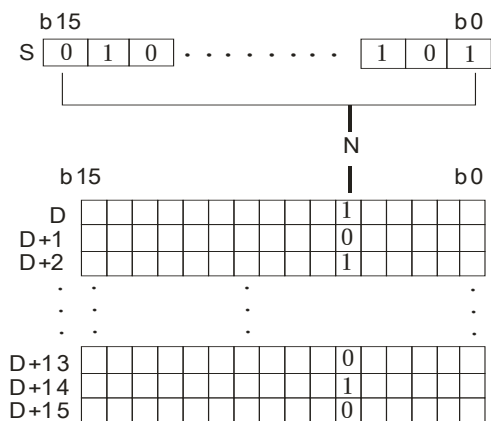
图形：



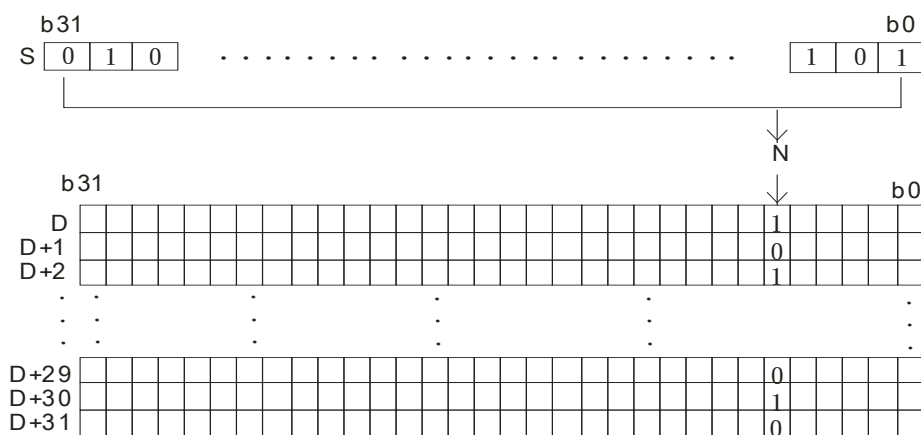
S : 数值数据源
n : 欲转换的数据数
D : 目的地装置

指令说明：

- 16 位指令 D 操作数固定占用连续 16 笔数据源 D~D+15，其中每一笔为 16 位寄存器。
- 32 位指令 D 操作数固定占用连续 32 笔数据源 D~D+31，其中每一笔为 32 位寄存器。
- n 操作数表示要填入 D 操作数中每一笔的第 n 个 Bit。并在 16 位指令中限制 n=0~15，在 32 位指令中限制 n=0~31。
- 16 位指令将数据源 S 依 b0~b15 的顺序，依序存于 D~D+15 中每一笔的第 n 个 Bit。
- 32 位指令将数据源 S 依 b0~b31 的顺序，依序存于 D~D+31 中每一笔的第 n 个 Bit。
- 32 位指令才可使用 HC 装置。
- 以 16 位指令说明：

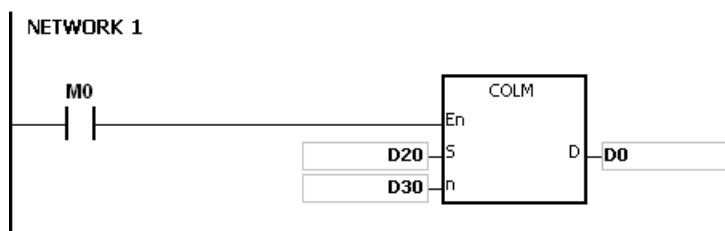


8. 以 32 位指令说明：



程序范例：

当 M0=ON 时，给定 D30=3，依序将 D20 的 b0~b15 存放到 D0~D15 的第 3 个位。



补充说明：

1. 16 位指令 **D+15** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 32 位指令 **D+31** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. **n** 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

3.6 数据转移指令

FB/FC	指令码 (位)			P 指令	功能	Step
	16	32	64			
FC	<u>MOV</u>	<u>DMOV</u>	—	✓	数据移动	5
FC	—	—	<u>DFMOV</u>	✓	双精度浮点数数据移动	5-6
FC	<u>\$MOV</u>	—	—	✓	字符串移动	5-11
FC	<u>CML</u>	<u>DCML</u>	—	✓	反转传送	5
FC	<u>BMOV</u>	—	—	✓	全部传送	7
FC	<u>NMOV</u>	<u>DNMOV</u>	—	✓	多点移动	7
FC	<u>XCH</u>	<u>DXCH</u>	—	✓	数据的交换	5
FC	<u>BXCH</u>	—	—	✓	全部交换	7
FC	<u>SWAP</u>	<u>DSWAP</u>	—	✓	上/下 BYTE 变换	3
FC	<u>SMOV</u>	—	—	✓	位数移动	11
FC	<u>MOVB</u>	—	—	✓	多位移动	7

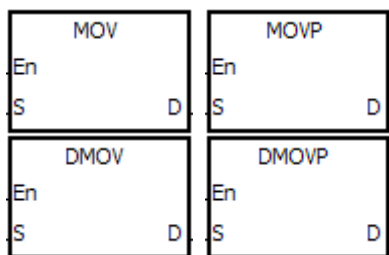
FB/FC	指令码			操作数	功能
FC	D*	MOV	P	S, D	数据移动

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		○
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



S : 数据源
D : 数据目的地

指令说明：

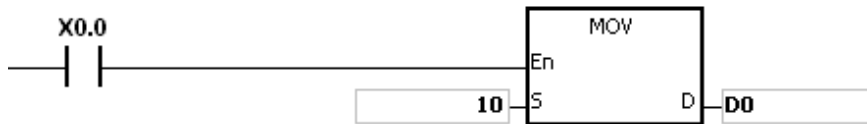
1. 当该指令执行时，将 S 的内容直接搬移至 D 内。当指令不执行时，D 内容不会变化。
2. 32 位指令 S 才可使用 F 浮点数。
3. 32 位指令才可使用 HC 装置。

程序范例：

1. 16 位数据搬移，须使用 MOV 指令。
 - 当 X0.0=OFF 时，D0 内容没有变化，若 X0.0=ON 时，将数值 10 传送至 D0 数据寄存器内。
 - 当 X0.1=OFF 时，D10 内容没有变化，若 X0.1=ON 时，将 T0 当前值传送至 D10 数据寄存器内。
2. 32 位数据搬移，须使用 DMOV 指令。
 - 当 X0.2=OFF 时，(D31、D30)、(D41、D40) 内容没有变化，若 X0.2=ON 时，将 (D21、D20) 当前值传送至 (D31、D30) 数据寄存器内。同时，将 HC0 当前值传送至 (D41、D40) 数据寄存器内。
3. F 浮点数搬移，须使用 DMOV 指令。

- 当 X0.3=OFF 时，(D51、D50) 内容没有变化，若 X0.3=ON 时，将浮点数 3.450 转换为二进浮点值传送至 (D51、D50) 数据寄存器内。

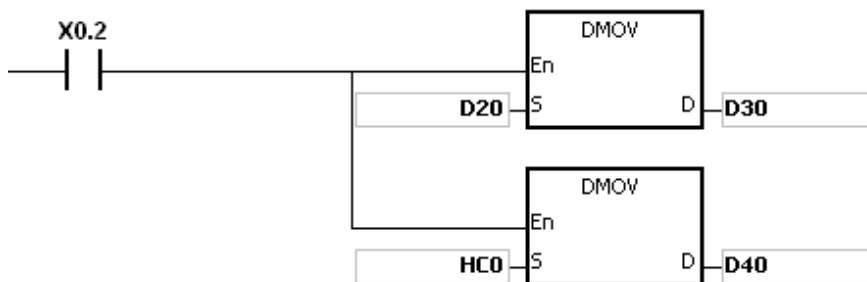
NETWORK 1



NETWORK 2



NETWORK 3



NETWORK 4



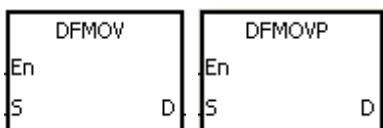
FB/FC	指令码			操作数	功能
FC	D*	FMOV	P	S, D	双精度浮点数数据移动

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S											●*			
D											●*			

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●			●	○					○
D	●	●			●	●	●	●			●	○					

脉冲执行型	32 位指令	64 位指令
AH 运动控制主机	-	AH 运动控制主机

图形：



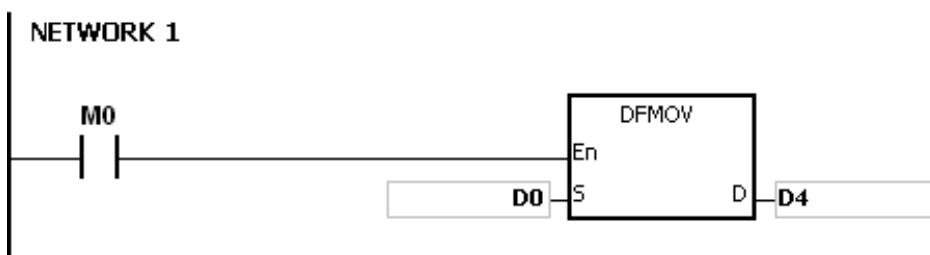
S : 数据源
D : 数据目的地

指令说明：

1. 当该指令执行时，将 S 的内容直接搬移至 D 内。当指令不执行时，D 内容不会变化。
2. 仅支持 64 位指令。
3. DFMOV、DFMOVP 指令为双精度搬移指令。

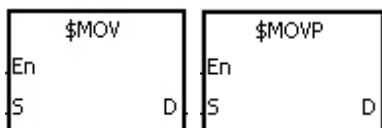
程序范例：

当 M0=ON 时，将 D0~D3 的数值传送到 D4~D7 当中。



FB/FC	指令码			操作数					功能									
FC		\$MOV	P	S, D					字符串移动									
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S																		●
D																		●
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●		●	●		●	○	●				○	
D	●	●			●	●		●	●		●	○	●					
				脉冲执行型				16 位指令				32 位指令						
				AH 运动控制主机				AH 运动控制主机				-						

图形：

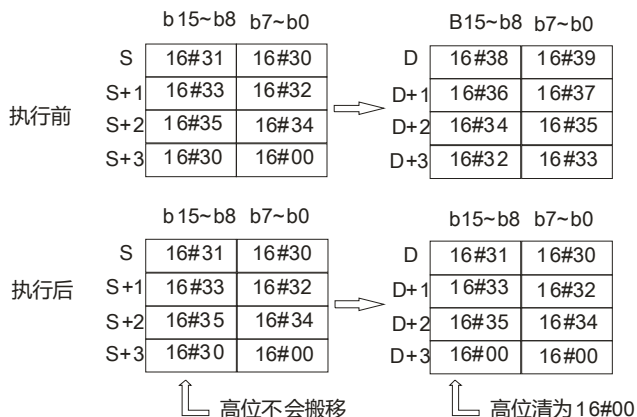
**S** : 数据源**D** : 数据搬移目的地

指令说明：

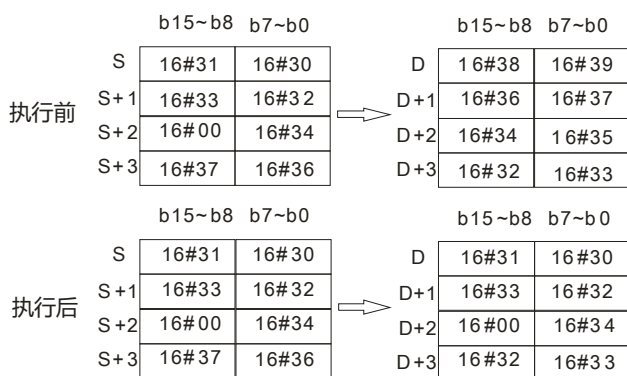
1. 若 **S** 操作数为字符串时，可以搬移 1~31 个字符；且字符串的 Step 数计算方式为 Step 数=1+(字符串数+1)/4 (若不能整除则无条件进位)。Step 数计算表格如下：

字符串数	1~3	4~7	8~11	12~15	16~19	20~23	24~27	28~31
Step 数	2	3	4	5	6	7	8	9

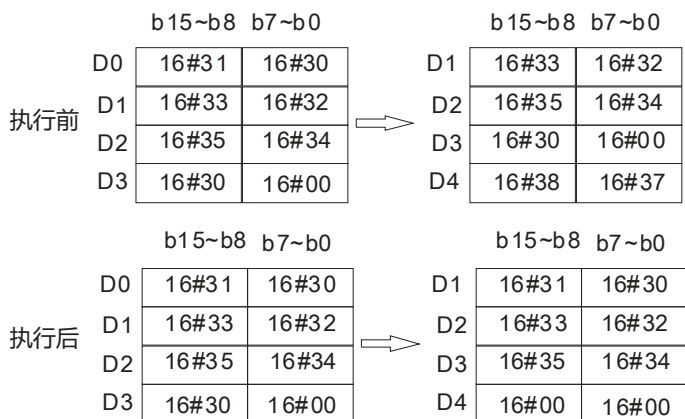
2. 若 **S** 操作数为字符串且该指令执行时，将 **S** 操作数的字符串直接搬移至 **D** 操作数内且结尾会补上结后缀 (16#00)。
3. 若 **S** 操作数要做搬移的不是字符串时，就要在结尾后面有一个结后缀 (16#00)。
4. 若 **S** 操作数不是字符串且该指令执行时 将 **S**~16#00 之间的字符串直接搬移至 **D** 内(含结后缀 16#00)。当指令不执行时，**D** 内容不会变化。
5. 若 **S** 操作数不是字符串且该指令执行时，第一个字符就是结后缀 (16#00)，则还是会搬移结后缀。
6. 当 16#00 字符串结尾在低位时，执行结果如下：



7. 当 16#00 字符串结尾在高位时，执行结果如下：

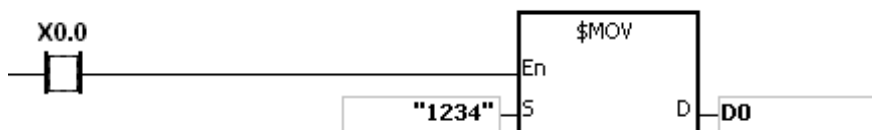


8. 当 S 跟 D 装置重叠且 S<D 时，搬移的顺序从结尾 16#00 开始搬移，一直搬到 D，不会发生 S 被覆盖的情形。



程序范例一：

若 S 操作数为偶数字符串“1234”时，当条件接点 X0.0 启动后，D0~D3 的值如下：



The operand **S** :

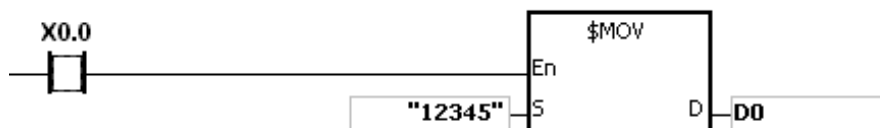
字符串	'1'	'2'	'3'	'4'
HEX	16#31	16#32	16#33	16#34

执行后, **D** 操作数

装置	上 8 位	下 8 位	备注
D0	16#32	16#31	'1'=16#31, '2'=16#32
D1	16#34	16#33	'3'=16#33, '4'=16#34
D2	16#00	16#00	下 8 位 16#00 为终止符, 上 8 位 16#00 为自动补零
D3	不变	不变	

程序范例二 :

若 **S** 操作数为奇数字符串“12345”时, 当条件接点 X0.0 启动后, D0~D3 的值如下 :



S 操作数 :

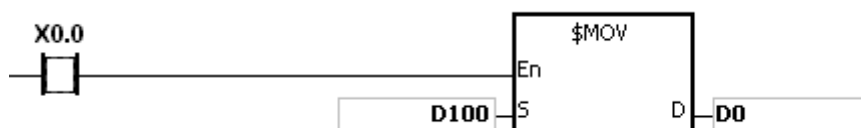
字符串	'1'	'2'	'3'	'4'	'5'
HEX	16#31	16#32	16#33	16#34	16#35

执行后, **D** 操作数

装置	上 8 位	下 8 位	备注
D0	16#32	16#31	'1'=16#31, '2'=16#32
D1	16#34	16#33	'3'=16#33, '4'=16#34
D2	16#00	16#35	上 8 位 16#00 为终止符
D3	不变	不变	

程序范例三 :

若 **S** 操作数非字符串且结尾字符 16#00 出现在低位时, 执行结果如下 :



S 操作数

装置	上 8 位	下 8 位	备注
D100	16#31	16#30	'1'=16#31, '0'=16#30
D101	16#33	16#32	'3'=16#33, '2'=16#32
D102	16#35	16#34	'5'=16#35, '4'=16#34
D103	16#30	16#00	'0'=16#30, 16#00 为终止符

执行后, **D 操作数**

装置	上 8 位	下 8 位	备注
D0	16#31	16#30	'1'=16#31, '0'=16#30
D1	16#33	16#32	'3'=16#33, '2'=16#32
D2	16#35	16#34	'5'=16#35, '4'=16#34
D3	16#00	16#00	下 8 位 16#00 为终止符, 上 8 位 16#00 为自动补零
D4	不变	不变	

程序范例四：

若 **S 操作数**非字符串且结尾字符 16#00 出现在高位时, 执行结果如下：



S 操作数：

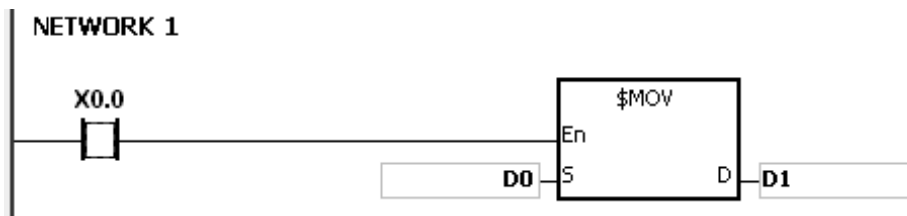
装置	上 8 位	下 8 位	备注
D100	16#31	16#30	'1'=16#31, '0'=16#30
D101	16#33	16#32	'3'=16#33, '2'=16#32
D102	16#00	16#34	16#00 为终止符, '4'=16#34
D103	16#37	16#36	'7'=16#37, '6'=16#36

执行后, **D 操作数**

装置	上 8 位	下 8 位	备注
D0	16#31	16#30	'1'=16#31, '0'=16#30
D1	16#33	16#32	'3'=16#33, '2'=16#32
D2	16#00	16#34	16#00 为终止符, '4'=16#34
D3	不变	不变	

程序范例五：

当 S 跟 D 装置重叠且 S<D 时，搬移的顺序从结尾 16#00 开始搬移，一直搬到 D，不会发生 S 被覆盖的情形。



S 操作数

装置	上 8 位	下 8 位	备注
D0	16#31	16#30	'1'=16#31, '0'=16#30
D1	16#33	16#32	'3'=16#33, '2'=16#32
D2	16#35	16#34	'5'=16#35, '4'=16#34
D3	16#30	16#00	'0'=16#30, 16#00 为终止符
D4	16#38	16#37	'8'=16#38, '7'=16#37

执行后，D 操作数

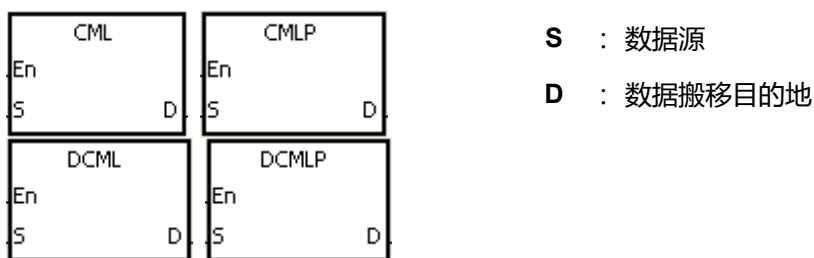
装置	上 8 位	下 8 位	备注
D1	16#31	16#30	'1'=16#31, '0'=16#30
D2	16#33	16#32	'3'=16#33, '2'=16#32
D3	16#35	16#34	'5'=16#35, '4'=16#34
D4	16#00	16#00	下 8 位 16#00 为终止符， 上 8 位 16#00 为自动补零
D5	不变	不变	

补充说明：

1. 当 S 中没有 16#00 当结尾时，指令不执行。SM0=ON，错误码 SR0=16#200E。
2. 当 D 装置空间不足以存放 S 的字符串时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC		指令码			操作数					功能							
FC		D*	CML	P	S, D					指令码							
数据类型		BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING		
S			●	●*				●	●*								
D			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型					16 位指令					32 位指令							
AH 运动控制主机					AH 运动控制主机					AH 运动控制主机							

图形：

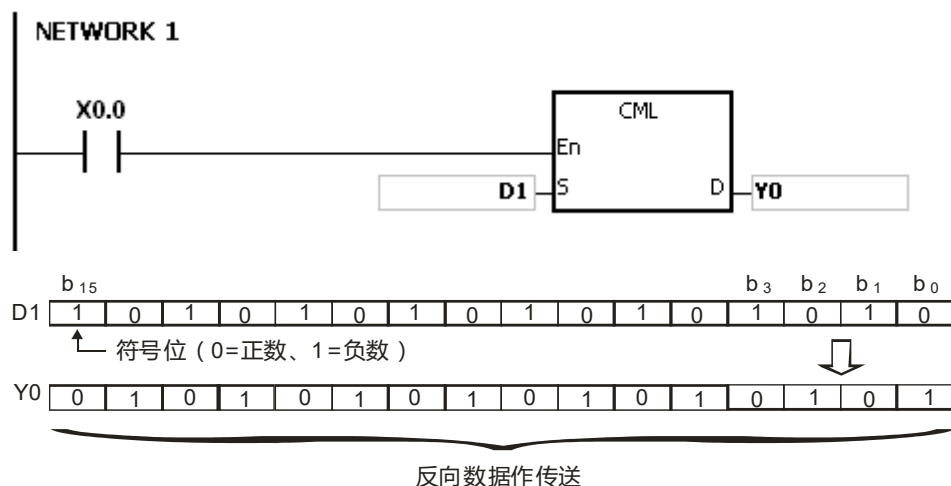


指令说明：

1. 希望作反相输出时，使用本指令。将 S 的内容全部反相 (0→1、1→0) 传送至 D 当中。如果内容为常数时，此常数自动被转换成 BIN 值。
2. 32 位指令才可使用 HC 装置。

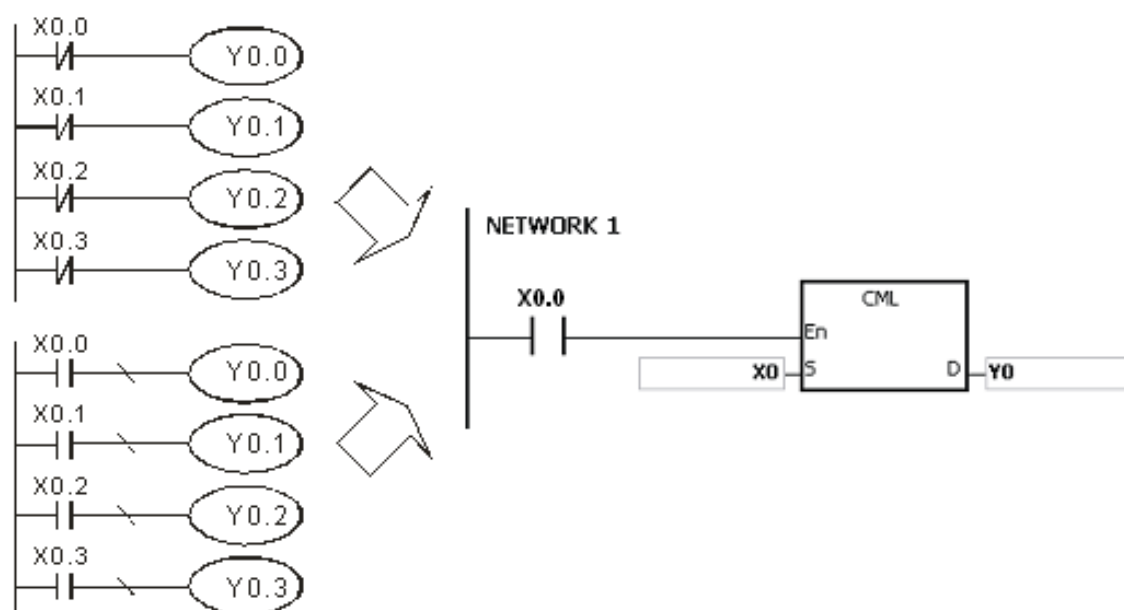
程序范例一：

当 X0.0=ON 时，将 D1 的 b0~b15 内容反相后传送到 Y0.0~Y0.15。



程序范例二：

下图左边的回路也可以使用 CML 指令来表现，如下图所示



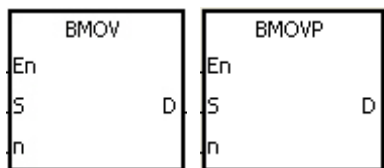
FB/FC	指令码		操作数				功能			
FC	BMOV	P	S, D, n				全部传送			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	

图形：

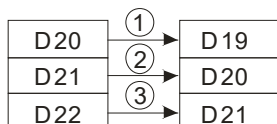


S : 数据源
D : 数据搬移目的地
n : 传送长度

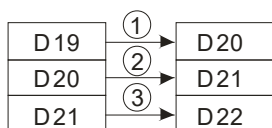
指令说明：

1. 所指定的装置起始号码开始算 **n** 个寄存器的内容被传送至 **D** 所指定的装置起始号码开始算 **n** 个寄存器当中。
2. **n**=1~256。
3. 为了防止两个操作数所指定传送的号码重叠时，所造成的混乱，请注意两个操作数所指定号码大小的安排，如下所示：

当 **S>D** 时，以①→②→③的顺序传送，

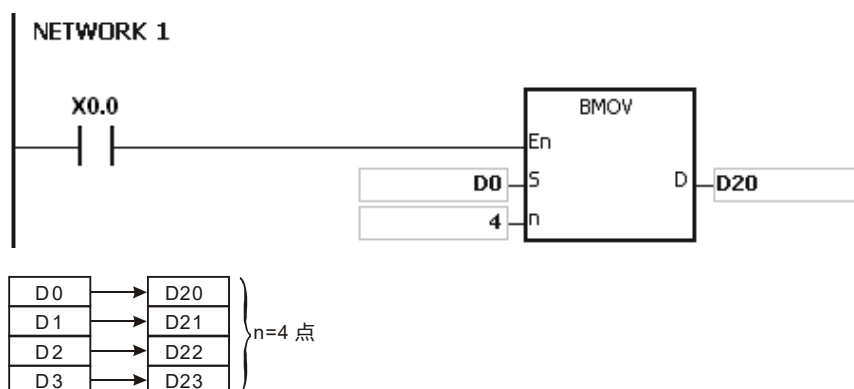


当 **S<D** 时，是以③→②→①的顺序传送。



程序范例一：

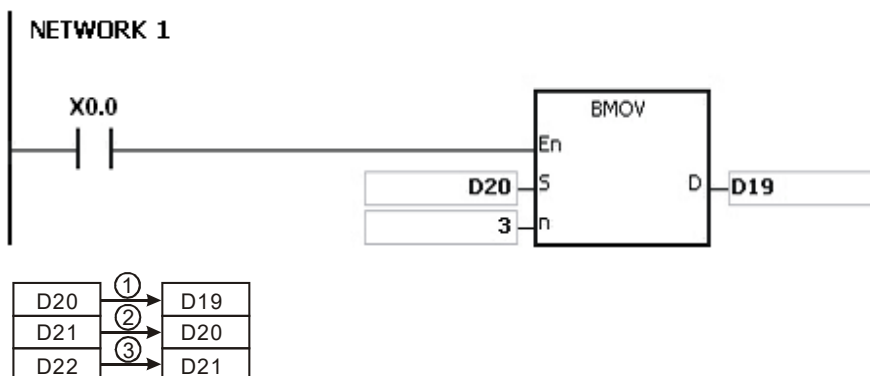
当 X0.0=ON 时，D0~D3 的 4 个寄存器的内容被传送至 D20~D23 的 4 个寄存器内。



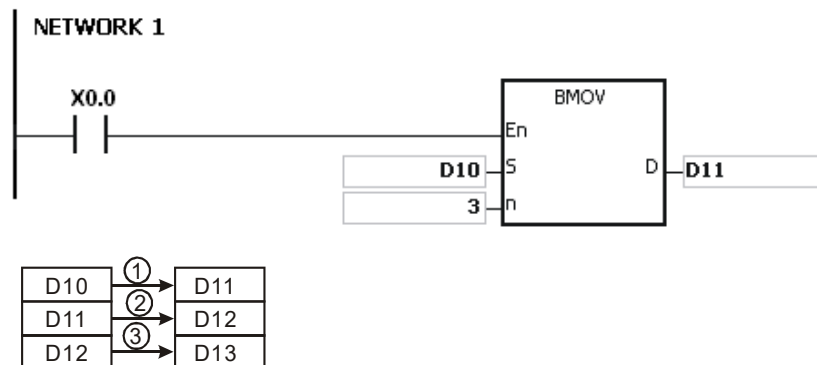
程序范例二：

为了防止两个操作数所指定传送的号码重叠时，所造成的混乱，请注意两个操作数所指定号码大小的安排，如下所示：

- 当 $S > D$ 时，以①→②→③的顺序传送



- 当 $S < D$ 时，其执行结果以③→②→①的顺序传送。



补充说明：

1. 当 $D+n-1$ 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 $S+n-1$ 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 当 $n > 256$ 或 $n < 1$ 时，指令不会执行，SM0=ON，错误码 SR0=16#200B。

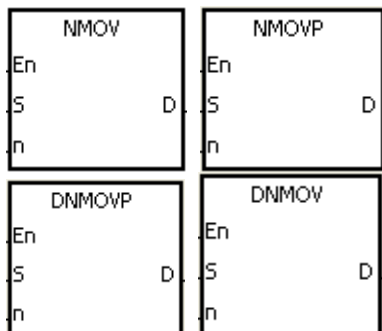
FB/FC	指令码			操作数				功能						
FC	D*	NMOV	P	S, D, n				多点移动						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

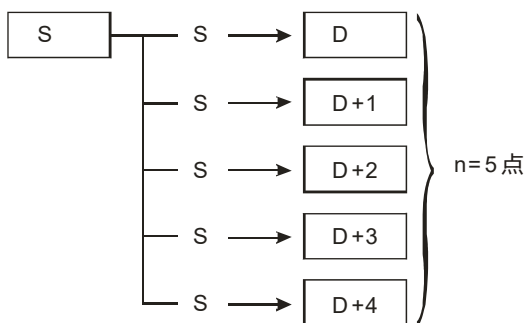
图形：



S : 数据源
 D : 数据搬移目的地
 n : 传送长度

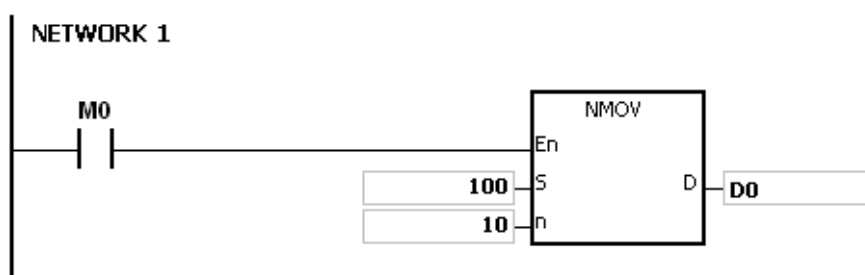
指令说明：

1. 当该指令执行时，将 S 的内容直接搬移至 D 开始算起的 n 个装置中，当指令不执行时，D 内容不会变化。
2. HC 装置只可在 32 位指令时使用。
3. NMOV 时 n=1~256，DNMOV 时 n=1~128。



程序范例：

M0 为 ON 时，D0~D9 的值为 100。

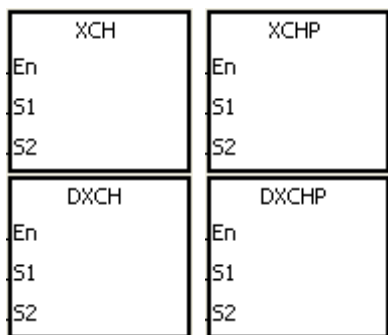


补充说明：

1. 当 $D \sim D+n-1$ 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 16 位指令中，当 $n > 256$ 或 $n < 1$ 时，指令不会执行，SM0=ON，错误码 SR0=16#200B。
3. 32 位指令中，当 $n > 128$ 或 $n < 1$ 时，指令不会执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数							功能							
FC	D*	XCH	P	S ₁ , S ₂							数据的交换							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁		●	●*				●	●*										
S ₂		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁	●	●			●	●	●	●	●		●	○	●					
S ₂	●	●			●	●	●	●	●		●	○	●					
				脉冲执行型				16 位指令				32 位指令						
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机						

图形：



S₁ : 要交换的数据 1

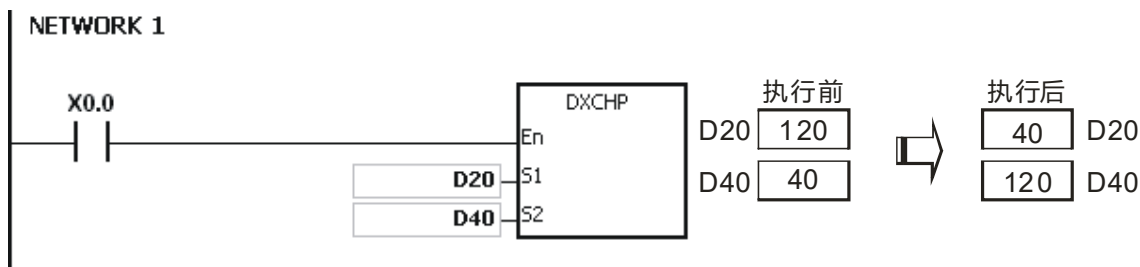
S₂ : 要交换的数据 2

指令说明：

1. S₁ 及 S₂ 所指定之装置内容值互相交换。
2. 32 位指令才可以使用 32 位计数装置。

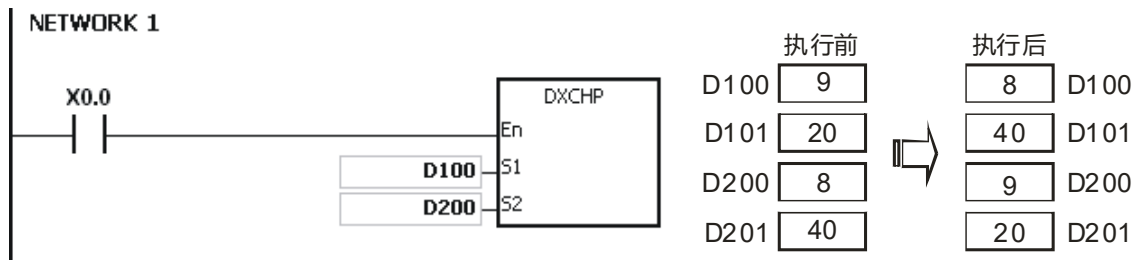
程序范例一：

X0.0=OFF→ON 时，D20 与 D40 的内容互相交换。



程序范例二：

X0.0=OFF→ON 时，D100 与 D200 的内容互相交换。



FB/FC	指令码		操作数					功能							
FC		BXCH	P	S_1, S_2, n					全部交换						

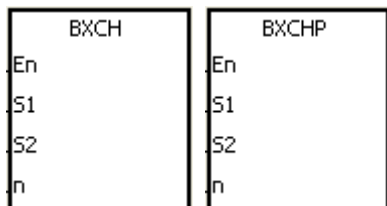
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

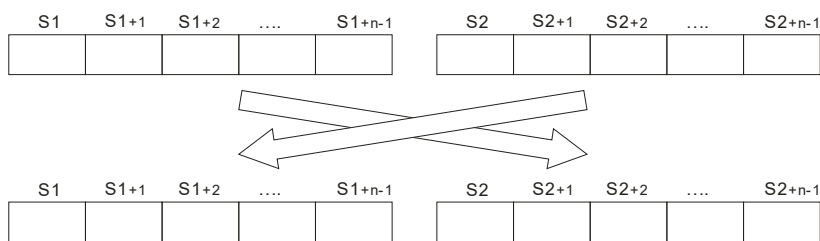
图形：



S_1 : 要交换的数据 1
 S_2 : 要交换的数据 2
 n : 长度

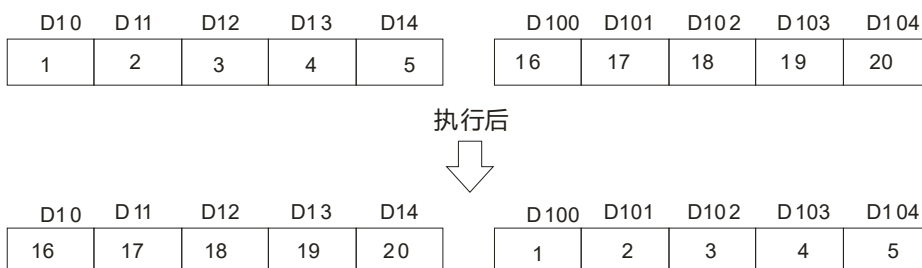
指令说明：

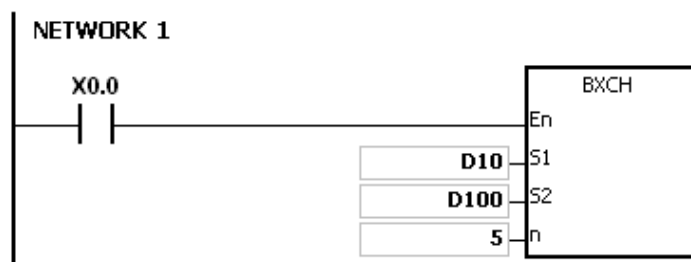
- $S_1 \sim S_1+n-1$ 及 $S_2 \sim S_2+n-1$ 所指定的装置内容值互相交换。
- $n=1 \sim 256$ 。



程序范例：

X0.0=ON 时，D10~D14 的内容与 D100~D104 的内容互相交换。

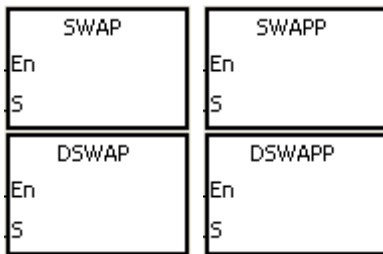


**补充说明：**

1. 当 S_1+n-1 的范围超出 S_1 装置时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 S_2+n-1 的范围超出 S_2 装置时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 当 $n>256$ 或 $n<1$ 时，指令不会执行，SM0=ON，错误码 SR0=16#200B。

FB/FC		指令码			操作数						功能						
FC		D*	SWAP	P	S						上/下 BYTE 变换						
数据类型		BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING		
S			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●			○	
脉冲执行型						16 位指令						32 位指令					
AH 运动控制主机						AH 运动控制主机						AH 运动控制主机					

图形：



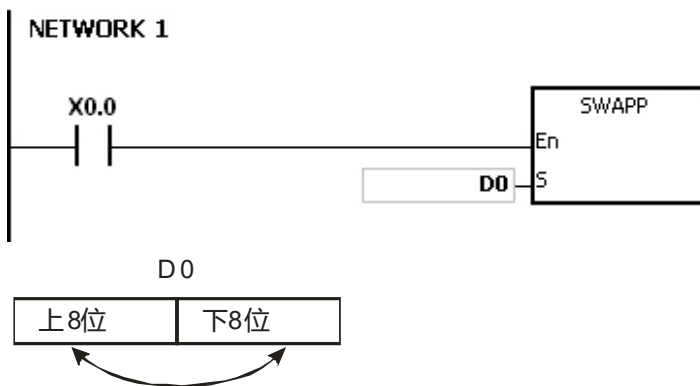
S : 数据源

指令说明：

1. 16 位指令，将 S 的上下 8 位的内容互相交换。
2. 32 位指令将 S 的上 16 位的上下 8 位内容互相交换，及下 16 位的上下 8 位内容互相交换。
3. 32 位指令才可以使用 32 位计数装置。

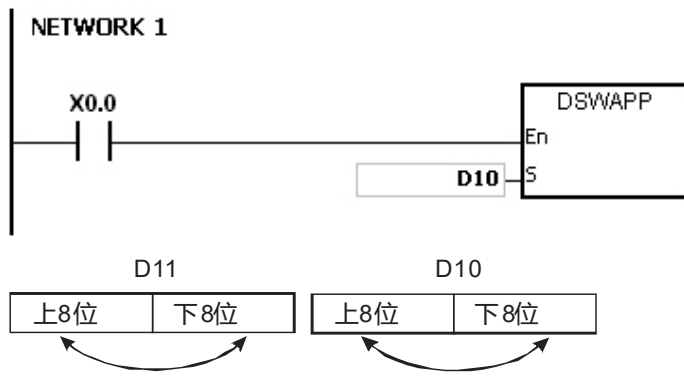
程序范例一：

当 X0.0=ON 时，将 D0 的上位 8 位与下位 8 位的内容互相交换。



程序范例二：

当 X0.0=ON 时，将 D11 的上位 8 位与下位 8 位的内容互相交换，D10 的上位 8 位与下位 8 位的内容互相交换。



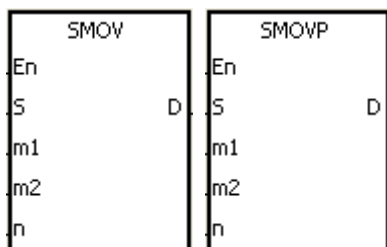
FB/FC	指令码		操作数				功能				
FC		SMOV	P	S , m₁ , m₂ , D , n				位数移动			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
m₁ , m₂		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
m₁ , m₂	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

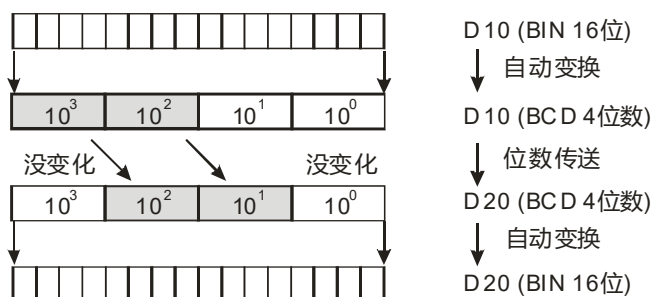
图形：



- S** : 要移动的数据源
- m₁** : 数据源要传送的起始位数
- m₂** : 数据源要传送的位数的个数
- D** : 传送之目的地装置
- n** : 目的地装置要存放数据源的起始位数

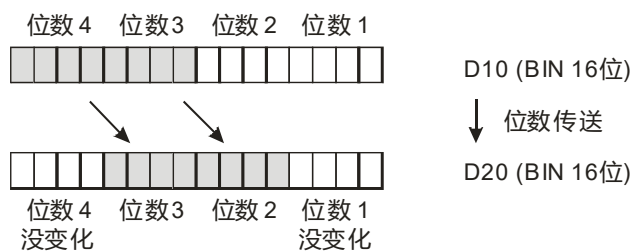
指令说明：

- 此指令可将数据重新分配或合成。当该指令执行时，指定 **S** 的第 **m₁** 位数开始往低位计算的 **m₂** 位数内容传送至 **D** 的第 **n** 位数开始往低位计算的 **m₂** 位数中。
- m₁**=1~4，**m₂**=1~**m₁**，**n**=**m₂**~4。（4 个 bit 为一个单位）
- SM605=OFF 时，为 BCD 模式，指令会自动将 **S** 的内容当成 BCD 的模式来执行。



若 S=K1234，D=K5678，则执行后 S=1234，D=5128

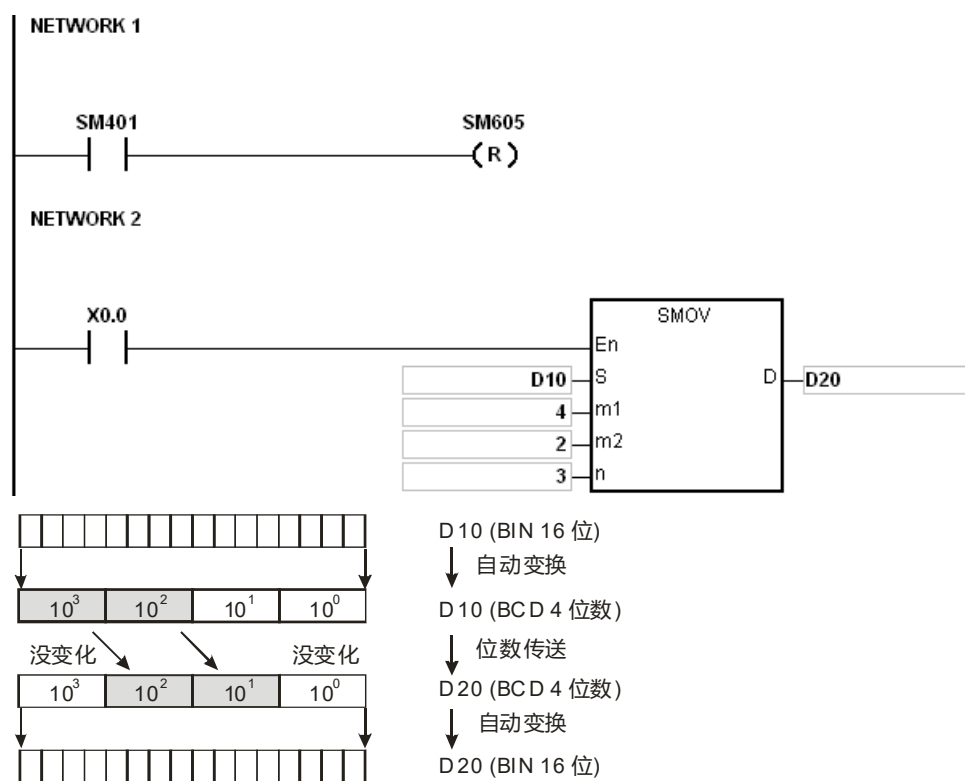
4. SM605=ON 时，为 BIN 模式。



若 S=16#1234，D=16#5678，则执行后 S=16#1234，D=16#5128。

程序范例一：

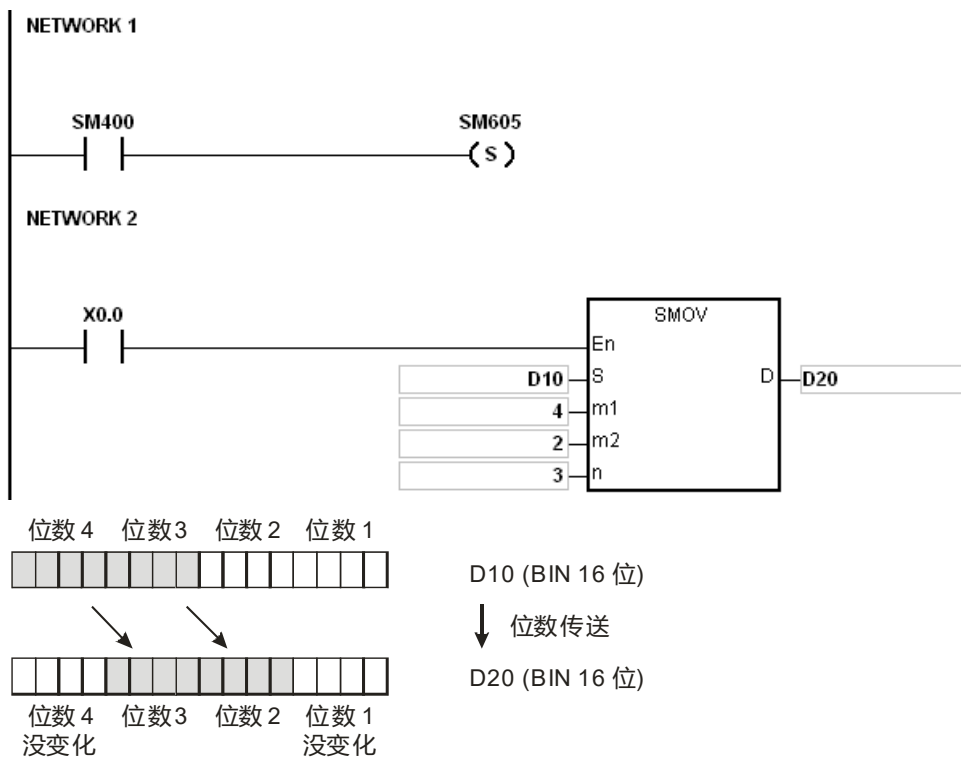
1. 当 SM605=OFF 时 (BCD 模式)，X0.0=ON，指定 D10 的 10 进制数值的第 4 位数 (亦即千位数) 开始往低位计算的 2 位数内容传送至 D20 的 10 进制数值的第 3 位数 (亦即百位数) 开始往低位计算的 2 位数中。而 D20 的 10^3 及 10^0 于本指令被执行后内容没有变化。
2. 当 BCD 值超过 0~9,999 的范围时, PLC 判定为运算错误 指令不执行, SM0=ON, 错误码 SR0=16#200D。



若执行前 D10=1234，D20=5678，执行完毕后，D10 不变，D20=5128。

程序范例二：

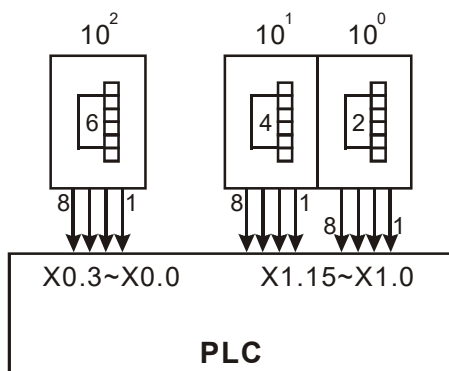
当 SM605=ON (BIN 模式) 时，使用 SMOV 指令的话，D10、D20 并不会作 BCD 变换，而是以 BIN 类型 4 个位为一个单位作传送。

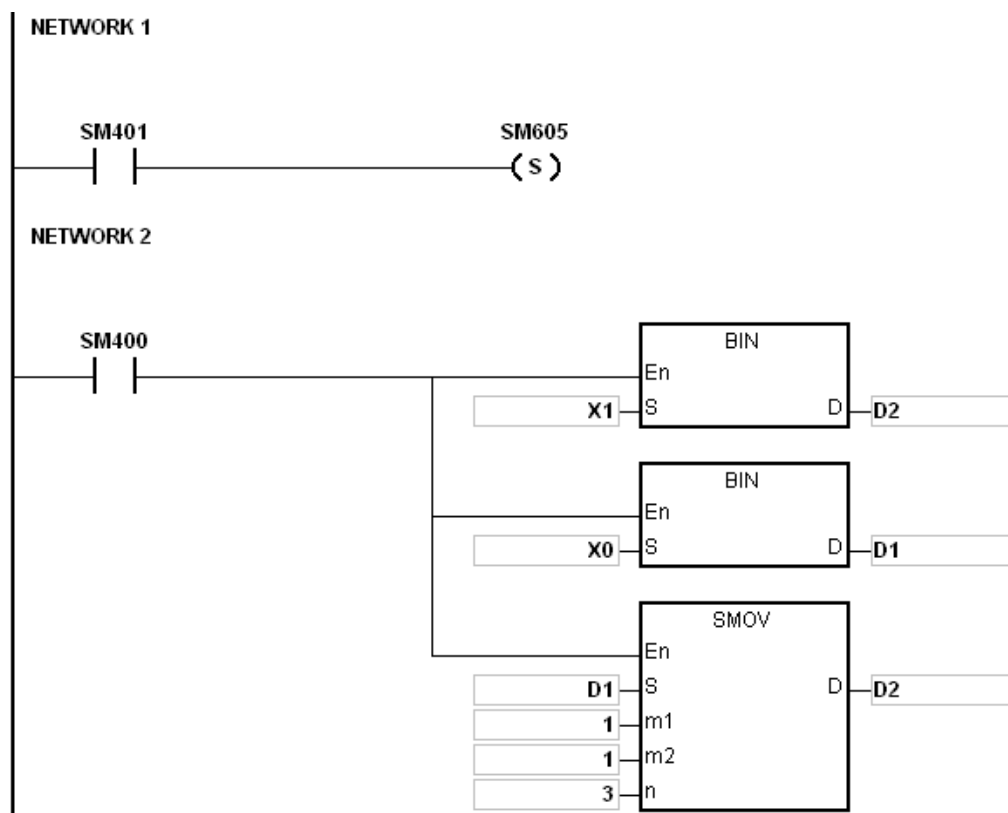


若执行前 D10=16#1234，D20=16#5678，执行完毕后，D10 不变，D20=16#5128。

程序范例三：

1. 连接于非连续编号输入端的指拨开关可使用本指令来合成。
2. 将右 2 位指拨开关传送至 D2 的右 2 位，左 1 位指拨开关传送至 D1 的右 1 位数当中。
3. 使用 SMOV 指令将 D1 的第 1 位传送至 D2 的第 3 位数将两组指拨开关合成 1 组。



**补充说明：**

1. 当指令为 BCD 模式时，如果 $S < 0$ 或 $S > 9999$ 或 $D < 0$ 或 $D > 9999$ ，指令不执行，SM0=ON，错误码 SR0=16#200D。
2. $m_1 < 1$ 或 $m_1 > 4$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 当 $m_2 < 1$ 或 $m_2 > m_1$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 当 $n < m_2$ 或 $n > 4$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

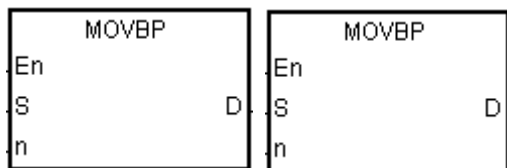
FB/FC	指令码		操作数				功能			
FC	MOVB	P	S, n, D				多位移动			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													
n		●					●							
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●	●	●	●	●	●	●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●	●	●	●	●	●	●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



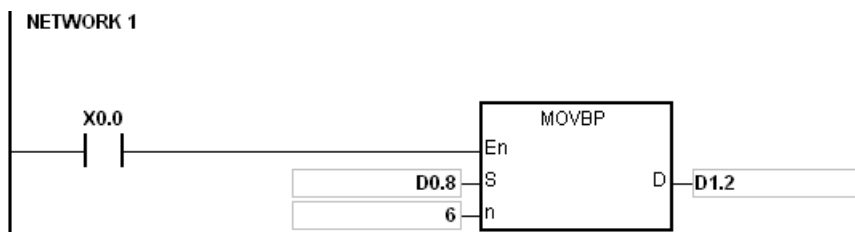
- S : 要移动的数据源
- n : 传送笔数
- D : 目的地装置要存放数据源的起始位数

指令说明：

1. 当该指令执行时，指定 S 起始位数内容传送至 D 开始的 n 个位数中。
2. S、D 为 T、C、HC/AC 装置时，只传送装置的状态，不会传送装置的当前值。
3. n=1~256，n<1 或 n>256 时指令不执行，SM0=ON，错误码=16#200B。

程序范例：

X0.0 为 ON 时，PLC RUN 后，D0.8~D0.13 的内容传送至 D1.2~D1.7。



补充说明：

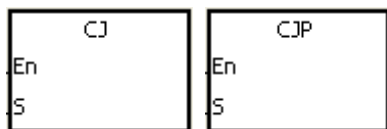
1. 当 D+n-1 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 S+n-1 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

3.7 程序跳转指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>CJ</u>	—	✓	条件跳转	3
FC	<u>JMP</u>	—	—	无条件跳转	3
FC	<u>GOEND</u>	—	—	跳转到 END	1

FB/FC	指令码					操作数					功能						
FC		CJ	P	S					条件跳转								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S																	
						脉冲执行型			16 位指令			32 位指令					
						AH 运动控制主机			AH 运动控制主机			-					

图形：



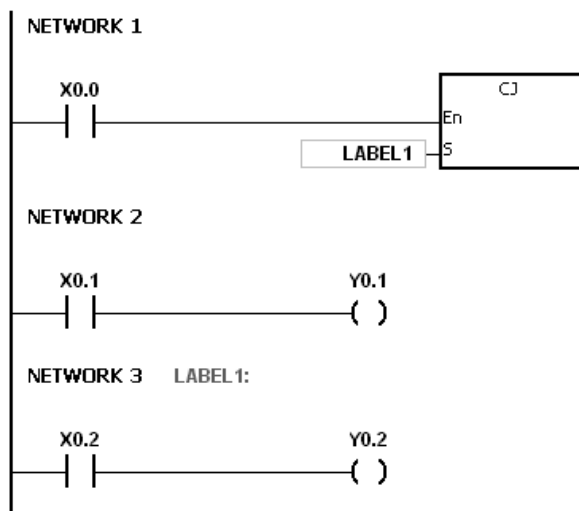
S : 条件跳转之目的指针

指令说明：

1. 当用户希望 PLC 程序中的某一部份不需要执行时，以缩短扫描时间，以及使用于双重输出时，可使用 CJ 或 CJP 指令。
2. 指针 P 所指之程序若在 CJ 指令之前，需注意会发生 WDT 逾时的错误，PLC 停止运转，请注意使用。
3. CJ 指令可重复指定同一指针 P。
4. 跳转执行中各种装置动作情形说明：
 - Y、M、S 保持跳转发生前的状态。
 - 执行计时中定时器会暂停计时。
 - 定时器之清除指令若在跳转前被驱动，则在跳转执行中该装置仍处于清除状态。
 - 一般应用指令不会被执行。

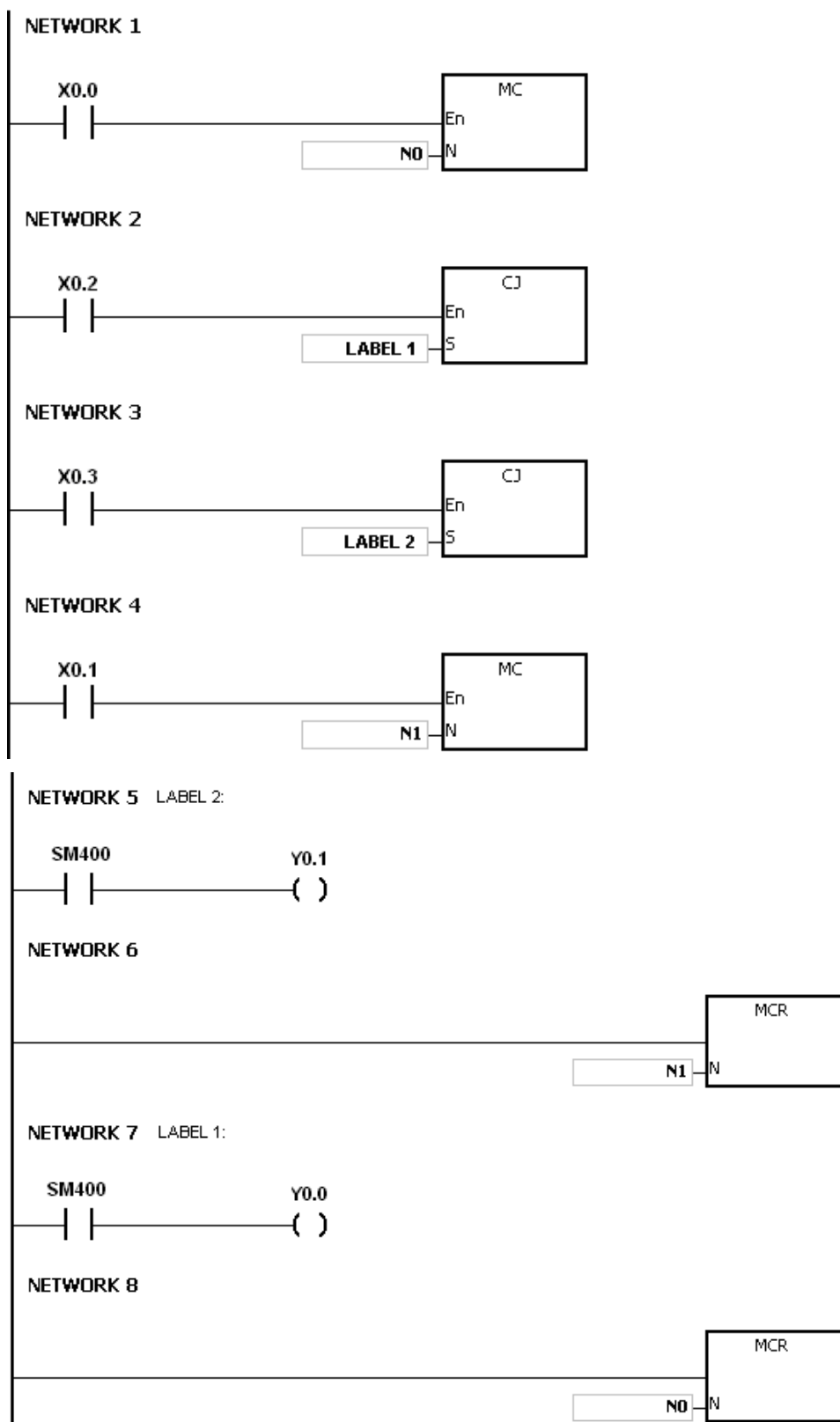
程序范例一：

1. 当 X0.0=ON 时，程序自动从地址 0 跳转至地址 N (即指定之标签 LABEL1 :) 继续执行，中间地址跳可不执行。
2. 当 X0.0=OFF 时，程序如同一般程序由地址 0 继续往下执行，此时 CJ 指令不被执行。



程序范例二：

1. CJ 指令在 MC、MCR 指令间可使用在下列五种状况：
 - (a) 在 MC~MCR 外。
 - (b) 在 MC 外至 MC 内，如下图 LABEL1：以下回路有效。
 - (c) 同一 N 层 MC 内至 MC 内。
 - (d) 在 MC 内至 MCR 外。
 - (e) 自 MC~MCR 内跳至另一 MC~MCR 内。
2. 执行 MC 指令时，PLC 会将之前开关接点的状态推入 PLC 内部自定义的堆栈中，而此堆栈由 PLC 自行控制，用户无法改变；而后当执行到 MCR 指令时，会由堆栈的最上层取出之前的开关接点状态，当上面 (b)、(d)、(e) 的状况下时，则有可能发生推入 PLC 内部堆栈与取出堆栈的次数不相同的情况，遇到这种状况时，堆栈最多能堆入 32 层，而另外取出堆栈的值最多取到堆栈为空时则不再取出，所以在搭配 CJ 等跳转指令时须注意堆栈的堆入和取出。



3

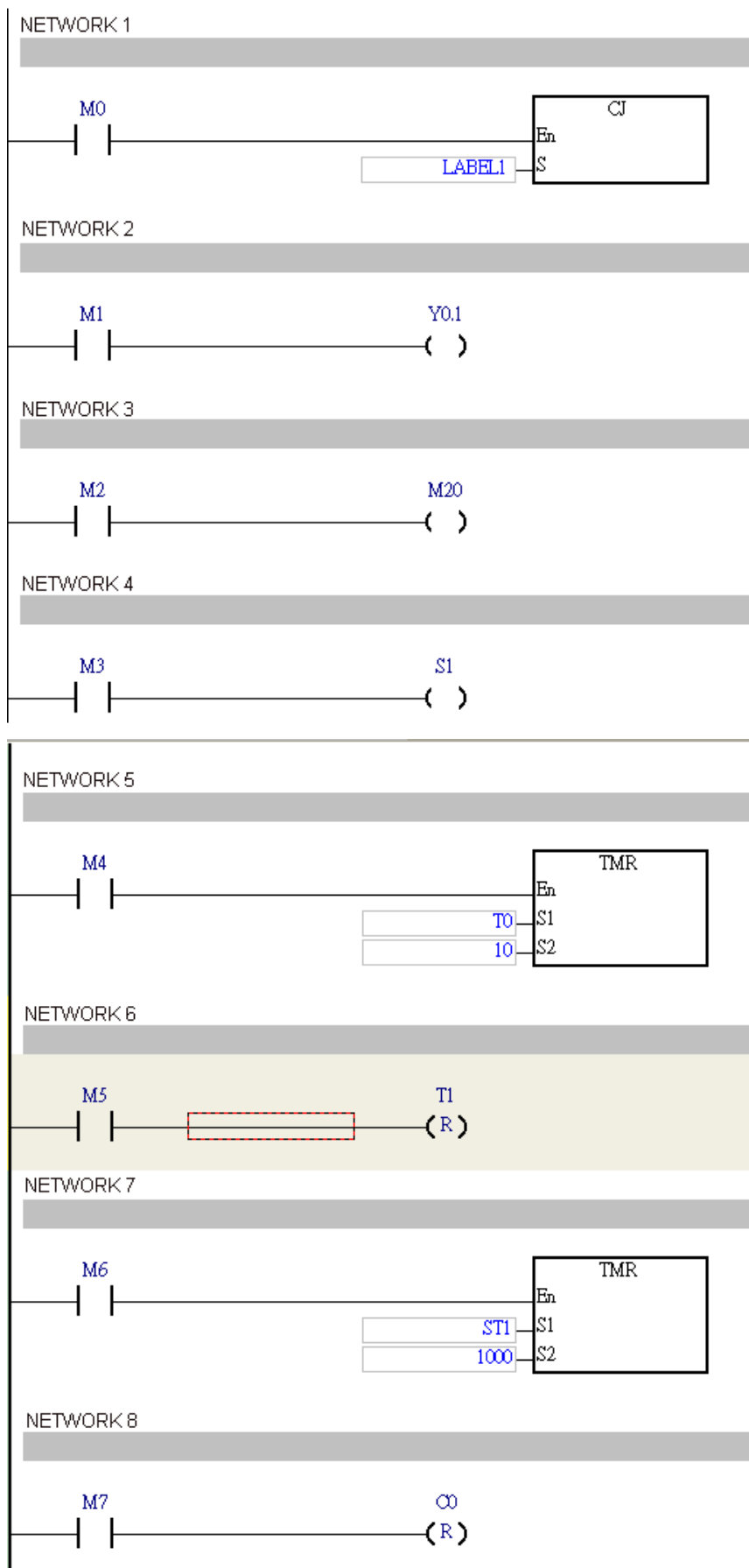
程序范例三：

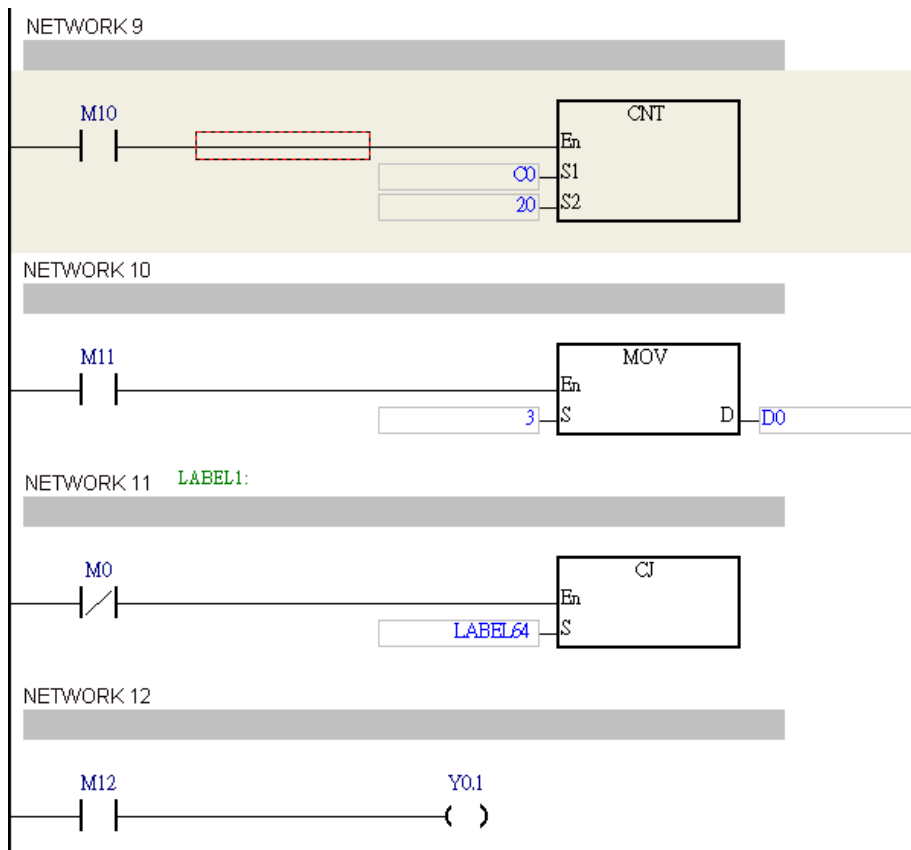
底下表格为下列程序中，各个装置状态变化。

装置	CJ 执行前 接点状态	CJ 执行中 接点状态	CJ 执行中输出线圈状态
Y、M、S	M1、M2、M3 OFF	M1、M2、M3 由 OFF→ON	Y0.1 ^{注1} 、M20、S1 OFF
	M1、M2、M3 ON	M1、M2、M3 由 ON→OFF	Y0.1 ^{注1} 、M20、S1 ON
定时器	M4 OFF	M4 由 OFF→ON	定时器不作计时动作
	M4 ON	M4 由 ON→OFF	定时器立即停止计时并保持，M0 由 ON 到 OFF，定时器清零。
运算型定时器	M6 OFF	M6 OFF→ON	定时器 ST1 不作计时动作
	M6 ON	M6 ON→OFF	运算型定时器一旦计时动作被启动，若遇到 CJ 指令时，则计时动作停止但保持。 M0 由 ON→OFF，ST1 仍保持
计数器	M7，M10 OFF	M10 ON/OFF 触发	计数器不计数
	M7 OFF，M10 ON/OFF 触发	M10 ON/OFF 触发	计数器 C0 停止计数并保持，M0 OFF 后，C0 继续计数
应用指令	M11 OFF	M11 OFF→ON	应用指令不执行
	M11 ON	M11 ON→OFF	被跳过的应用指令不执行

注 1：Y0.1 为双重输出，M0 为 OFF 时，由 M1 控制，M0 为 ON 时，由 M12 控制。

3

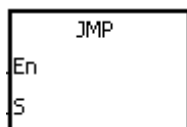


**补充说明：**

指针 P 在 ISPSOft 中的使用方式，请参考 **ISPSOft 使用手册** 说明。

FB/FC	指令码			操作数						功能							
FC		JMP		S						无条件跳转							
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S																	
						脉冲执行型			16 位指令			32 位指令					
						-			AH 运动控制主机			-					

图形：



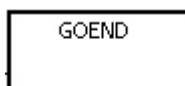
S : 跳转之目的指针

指令说明：

1. 无条件跳转到程序中的某个 P 指针。
2. 指针 P 所指的程序若在 JMP 指令之前，需注意会发生 WDT 逾时的错误，PLC 停止运转，请注意使用。
3. 跳转执行中各种装置动作情形说明：
 - Y、M、S 保持跳转发生前的状态。
 - 执行计时中定时器会暂停计时。
 - 定时器之清除指令若在跳转前被驱动，则在跳转执行中该装置仍处于清除状态。
 - 一般应用指令不会被执行。

FB/FC	指令码		操作数	功能	
FC		GOEND	-	跳转到 END	
			脉冲执行型	16 位指令	32 位指令
			-	AH 运动控制主机	-

图形：



指令说明：

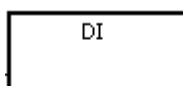
1. 条件成立时跳转到程序中的 END。
2. 功能块跟中断工作 (TASK) 与 FOR NEXT 中不支持此指令。
3. GOEND 执行时所略过的指令皆不动作，所有装置元件内容值与状态保持不变。

3.8 程序执行控制指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>DI</u>	—	—	中断插入禁止	1
FC	<u>EI</u>	—	—	中断插入使能	1
FC	<u>IMASK</u>	—	—	中断控制	3

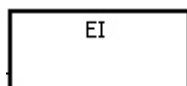
FB/FC	指令码		操作数	功能	
FC		DI	-	中断插入禁止	
			脉冲执行型	16 位指令	32 位指令
			-	AH 运动控制主机	-

图形：



FB/FC	指令码			操作数						功能							
FC		EI		-						中断插入使能							
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S																	
									脉冲执行型			16 位指令			32 位指令		
									-			AH 运动控制主机			-		

图形：



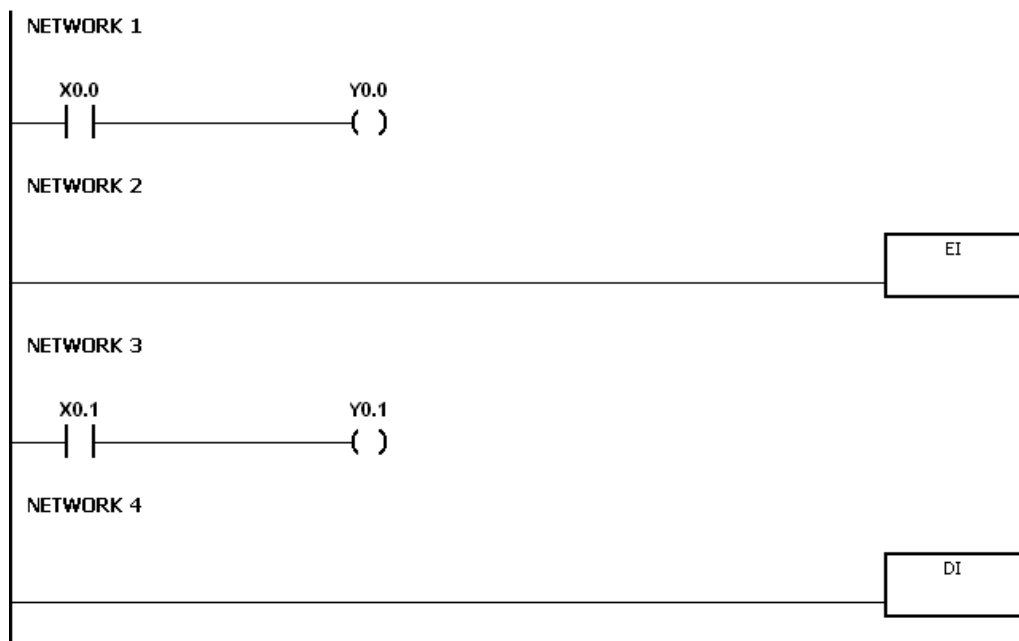
指令说明：

1. EI 表示程序中允许使用中断工作 (TASK) (关于 TASK 使用方式请参考 **ISPSOFT 软件手册**)。
2. 程序中在 EI 指令到 DI 指令间允许使用中断工作 (TASK) , 在程序中若无中断插入禁止之区间时, 则可以不使用 DI 指令。
3. 中断工作 (TASK) 执行中, 不会再执行其它产生的中断 (但会记住有中断产生) , 会先执行完目前的中断工作 (TASK) , 才去执行下一个中断工作 (TASK) 。
4. 当多数中断发生时, 以执行者优先, 同时发生以指针编号较小者优先。
5. 在 DI~EI 指令之间发生的中断要求无法立即执行, 此要求会被记忆一次, 并在中断许可范围内时, 才去执行中断程序。
6. 当中断处理中要实时 I/O 动作时, 可在程序中写入 REF 指令更新 I/O 状态。 .

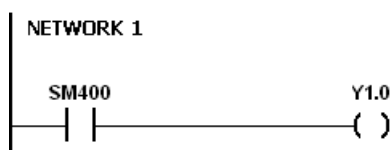
程序范例：

PLC 执行时, 当 Cyclic_0 程序扫描到 EI 指令到 DI 指令间, 如果有中断工作 (TASK) 被启动则会执行中断工作 (TASK) , 而当中断工作 (TASK) 执行完毕后, 则返回主程序并继续往下执行。

Cyclic_0 程序：



中断工作 (TASK)：



补充说明：

中断工作 (TASK) 0~255 共 256 个。

1. I/O 中断 (I0~I31)

特殊高速模块使用，模块经由 HWCONFIG 设定好中断条件以及中断编号，并经由 ISPSOft 下载中断程序到 PLC，PLC 运转时，当模块设定的中断条件成立，就会执行所对应的中断程序。

2. 通讯中断 (I32, I33)

当 RS 指令，特定字符通讯接收中断请求，也可当一般中断使用。详细说明请参考 RS 指令。

COM1 : I32。

COM2 : I33。

3. 24V 低电压检测中断

能藉由 AHPS05 上的 LV2 (VS+/VS-) 检测点，检测外部 24V 的电压是否正常，当外部 24V 不正常时，让用户可藉由中断子程序 (I34) 执行相对应的流程。

说明：每一块背板发生低电压时，将 SR731 里相对应的 bit 设定为 ON，低电压恢复后该 bit 会设定为 OFF。SR731 的 Bit8~15 为保留位。

SR731



例：

(a) 主背板发生 24V 低电压时，SR731 的第 0 个 bit 会被设为 ON。

(b) 第一块扩展背板发生 24V 低电压时，SR731 的第 1 个 bit 会被设为 ON，以此类推。

4. 外部中断 (I40~I251)

当有周边装置 (例：特殊 I/O 模块) 发出请求中断时，PLC 会去执行特定的中断工作 (Task)。

5. 定时中断 (I252~I255)

定时中断 0 (I252)：默认值 100ms (1~1000ms)

定时中断 1 (I253)：默认值 40ms (1~1000ms)

定时中断 2 (I254)：默认值 20ms (1~1000ms)

定时中断 3 (I255)：默认值 10ms (1~1000ms)

在一定的间隔时间内，执行此定时中断工作 (Task)。例如：每隔 10ms 执行此定时中断工作 (Task)。

● 优先级

中断编号	说明	优先级
I0	I/O 中断 0	1
I1	I/O 中断 1	2
I2	I/O 中断 2	3
I3	I/O 中断 3	4
I4	I/O 中断 4	5
I5	I/O 中断 5	6
I6	I/O 中断 6	7
I7	I/O 中断 7	8
I8	I/O 中断 8	9
I9	I/O 中断 9	10
I10	I/O 中断 10	11
I11	I/O 中断 11	12
I12	I/O 中断 12	13
I13	I/O 中断 13	14
I14	I/O 中断 14	15
I15	I/O 中断 15	16
I16	I/O 中断 16	17

中断编号	说明	优先级
I17	I/O 中断 17	18
I18	I/O 中断 18	19
I19	I/O 中断 19	20
I20	I/O 中断 20	21
I21	I/O 中断 21	22
I22	I/O 中断 22	23
I23	I/O 中断 23	24
I24	I/O 中断 24	25
I25	I/O 中断 25	26
I26	I/O 中断 26	27
I27	I/O 中断 27	28
I28	I/O 中断 28	29
I29	I/O 中断 29	30
I30	I/O 中断 30	31
I31	I/O 中断 31	32
I32	保留	33
I33	保留	34
I34	保留	35
I35~I39	保留	36~40
I40~I251	外部中断	41~252
I252	定时中断 0, 默认值: 100ms (1ms~1000ms)	253
I253	定时中断 1, 默认值: 40ms (1ms~1000ms)	254
I254	定时中断 2, 默认值: 20ms (1ms~1000ms)	255
I255	定时中断 3, 默认值: 10ms (1ms~1000ms)	256

FB/FC	指令码			操作数						功能					
FC	IMASK			S						中断控制					

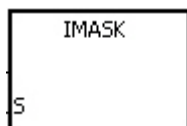
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：

3



S : 数据源

指令说明：

1. S~S+15 的内容作中断允许或禁止的控制,当 bit=1 ,且 EI 指令有执行时,相对应的中断程序才可以执行, bit=0 时,相对应的中断程序无法执行。
2. 执行此指令时会将 S~S+15 的内容值搬移至 SR623~SR638。
3. 如果没有执行此指令,则中断允许或禁止的控制 bit 会以 SR623~SR638 的内容来决定。

S	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
S+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
S+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
S+3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
S+4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
S+5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
S+6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
S+7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112
S+8	I143	I142	I141	I140	I139	I138	I137	I136	I135	I134	I133	I132	I131	I130	I129	I128
S+9	I159	I158	I157	I156	I155	I154	I153	I152	I151	I150	I149	I148	I147	I146	I145	I144
S+10	I175	I174	I173	I172	I171	I170	I169	I168	I167	I166	I165	I164	I163	I162	I161	I160
S+11	I191	I190	I189	I188	I187	I186	I185	I184	I183	I182	I181	I180	I179	I178	I177	I176
S+12	I207	I206	I205	I204	I203	I202	I201	I200	I199	I198	I197	I196	I195	I194	I193	I192
S+13	I223	I222	I221	I220	I219	I218	I217	I216	I215	I214	I213	I212	I211	I210	I209	I208
S+14	I239	I238	I237	I236	I235	I234	I233	I232	I231	I230	I229	I228	I227	I226	I225	I224
S+15	I255	I254	I253	I252	I251	I250	I249	I248	I247	I246	I245	I244	I243	I242	I241	I240

补充说明：

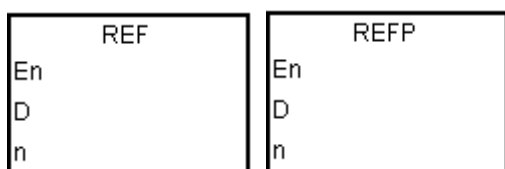
S~S+15 装置超出范围时，指令不执行，SM0=ON，SR0 错误码=16#2003。

3.9 I/O更新指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>REF</u>	—	✓	I/O 更新处理	5

FB/FC	指令码			操作数							功能						
FC		REF	P	D, n							I/O 更新处理						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
D	●																
n																	
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	○	○						○					○				
n	●	●						●	●		●		●	○	○		
							脉冲执行型			16 位指令				32 位指令			
							AH 运动控制主机			AH 运动控制主机				-			

图形：



D : 欲 I/O 更新处理的起始装置

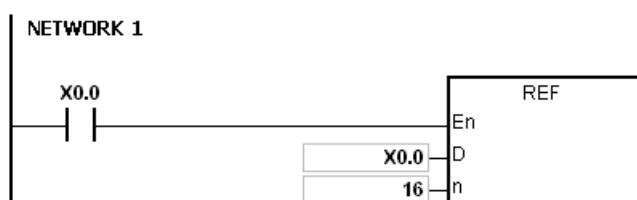
n : I/O 更新处理的数目

指令说明：

PLC 的 I/O 更新状态全部为程序扫描至 END 后,才作状态的更新,其中输入点的状态是在程序开始扫描时,自外部输入点的状态读入存在输入点内存中,而输出端子在 END 指令后,才将输出点内存内容送至输出装置。因此在运算过程中需要最新的 I/O 数据,则可利用本指令。

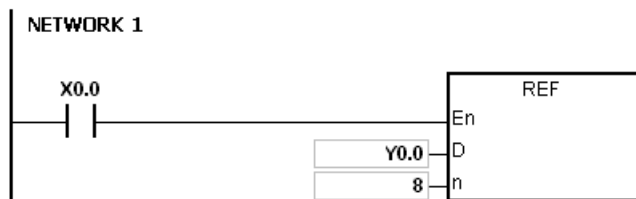
程序范例一：

当 X0.0=ON 时, PLC 会立即读取 X0.0~X0.15 之输入点状态, 输入信号更新, 并没有输入延迟。



程序范例二：

当 X0.0=ON 时, Y0.0~Y0.7 的 8 点输出信号实时被送至输出端, 输出信号立即更新, 不必到 END 指令才输出。



补充说明：

1. D+n-1 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 n>256 或 n<1 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

3.10 便利指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>ALT</u>	—	✓	ON/OFF 交替	3
FC	<u>TTMR</u>	—	—	教导式定时器	5
FC	<u>STMR</u>	—	—	特殊定时器	7
FC	<u>RAMP</u>	—	—	倾斜信号	9
FC	<u>MTR</u>	—	—	矩阵输入	9
FC	<u>ABSD</u>	<u>DABSD</u>	—	绝对方式凸轮控制	9
FC	<u>INCD</u>	—	—	相对方式凸轮控制	9
FC	—	<u>DPID</u>	—	PID 运算	35
FC	—	<u>DPIDE</u>	—	PID 运算	43

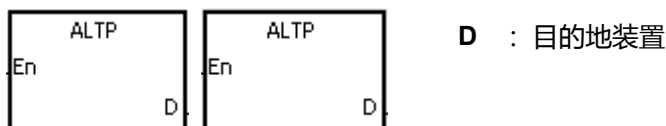
FB/FC	指令码			操作数			功能							
FC		ALT	P	D			ON/OFF 交替							

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

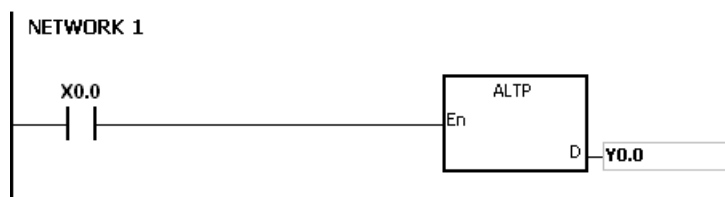


指令说明：

1. ALT 指令执行时，D ON/OFF 交换。
2. 本指令一般都是使用脉冲执行型指令（ALTP）。

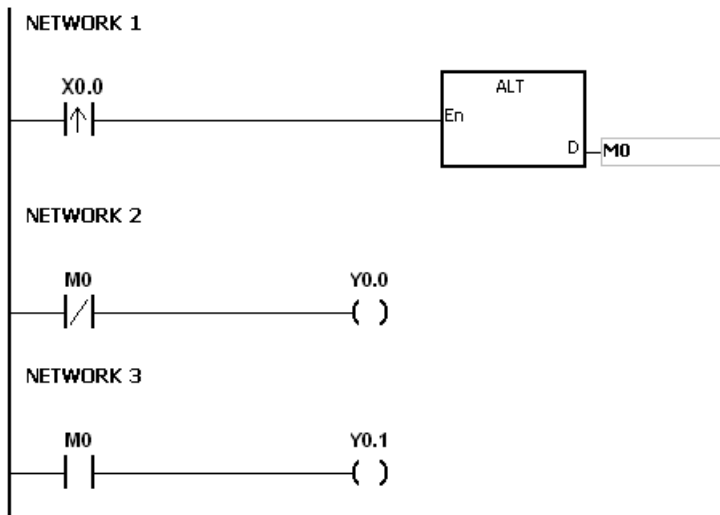
程序范例一：

当第一次 X0.0 从 OFF→ON 时，Y0.0=ON。第二次 X0.0 从 OFF→ON 时，Y0.0=OFF。



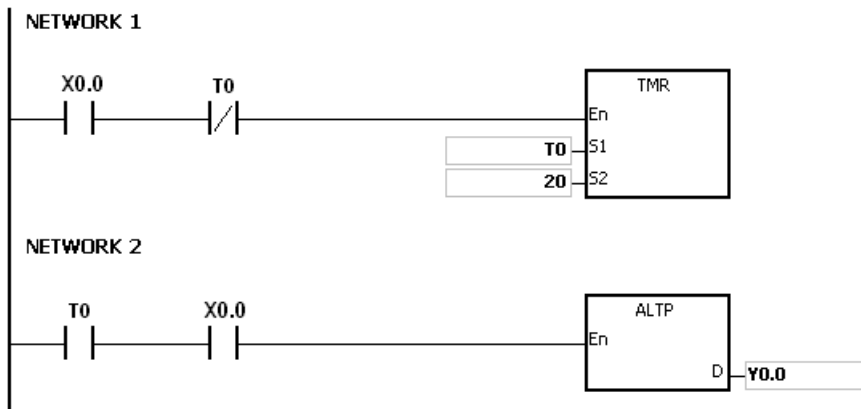
程序范例二：

使用单一开关控制启动与停止。一开始时，M0=OFF 故 Y0.0=ON、Y0.1=OFF，当 X0.0 作第一次 ON/OFF 时，M0=ON 故 Y0.1=ON、Y0.0=OFF，第二次 ON/OFF 时，M0=OFF 故 Y0.0=ON 而 Y0.1=OFF。



程序范例三：

产生闪烁的动作。当 X0.0=ON 时，T0 每隔 2 秒产生一个脉冲，Y0.0 输出会依 T0 脉冲产生 ON/OFF 交替。



FB/FC	指令码			操作数				功能						
FC		TTMR		n, D				教导式定时器						

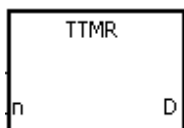
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
n	●	●						●	●		●		●	○	○		
D	●	●						●	●		●		●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

3

图形：



D : 储存按钮开关 ON 时间的装置编号

n : 倍数设定

指令说明：

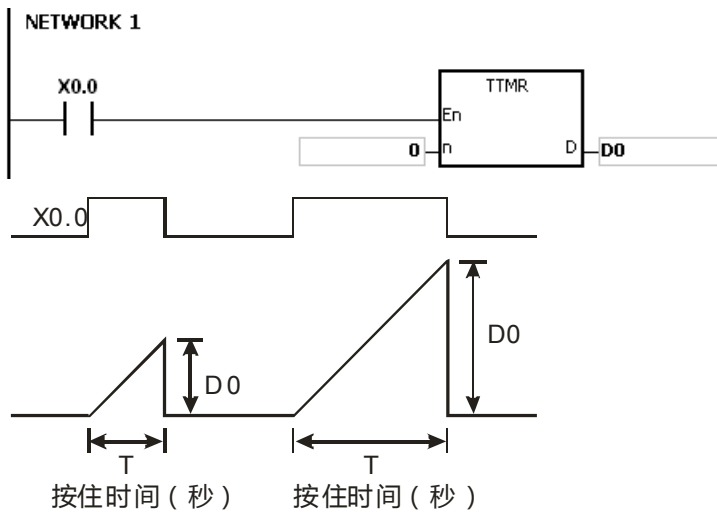
1. 将开关 ON 时间以秒为单位乘以 n 倍数后存放于 D 内，D+1 为系统内部使用，指令执行时不可更改，否则会造成时间计算错误。
2. D 的内容值在条件接点为 ON 的时候会先清为 0。
3. 倍数设定：n=0 时，D 以秒为单位，n=1 时，D 乘以 10 倍以 100ms 为单位，n=2 时，D 乘以 100 倍以 10ms 为单位。

n	D
K0 (单位：s)	1×T
K1 (单位：100 ms)	10×T
K2 (单位：10 ms)	100×T

4. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。
5. n 操作数的范围为 0~2。

程序范例一：

1. 按钮开关 X0.0 被按住时间 (X0.0 的 ON 时间)，由 n 来指定该时间的倍数，并将位数时间存入于 D0 当中。由此，可使用按钮开关来调整定时器的设定值。
2. 当 X0.0 变成 OFF 时，D0 内容没有变化。

**补充说明：**

1. $D+1$ 超出装置范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
2. $n < 0$ 或 $n > 2$ 时，指令不执行， $SM0=ON$ ， $SR0$ 记录错误码= $16\#200B$ 。
3. D 操作数，若使用 ISPSOft 宣告，则数据类型为 $ARRAY [2] \text{ of } WORD/INT$ 。

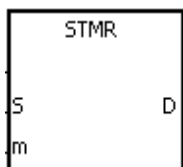
FB/FC	指令码		操作数				功能				
FC		STMR	S, D, m				特殊定时器				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S											●			
D	●													
m		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S					○												
D	●	●	●	●				●	●	●			●				
m	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



- S** : 定时器编号
- m** : 定时器设定值
- D** : 输出装置的起始编号

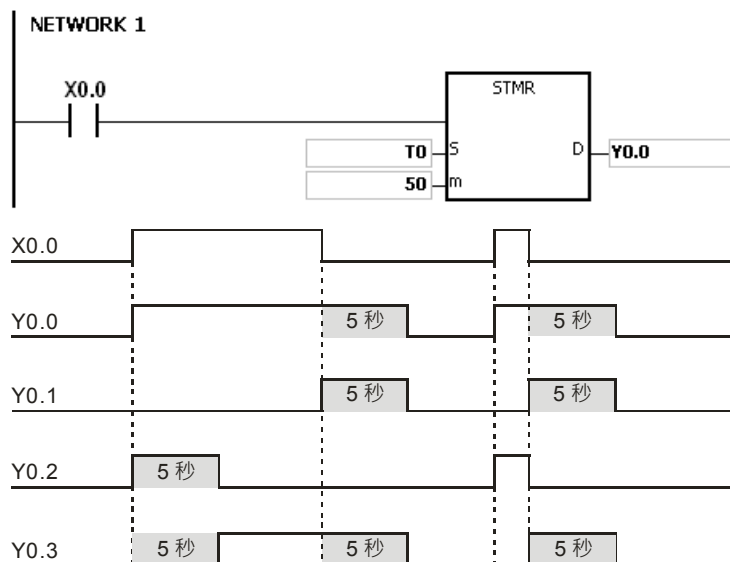
指令说明：

1. STMR 指令，用来产生 OFF 延迟，一次触发及闪烁回路的专用指令。
2. STMR 指令所指定的 T 定时器以 100ms 为单位。
3. STMR 指令所指定的定时器号码不可重复使用。
4. D 连续占用 4 个 bit。
5. 执行此指令前，请先 RST D~D+3 的状态。
6. 当条件接点未启动时，如果装置内容为以下两种，则 D、D+1、D+3 的状态会 ON m 秒之后再 OFF，除了以下两种情形之外，D~D+3 的状态会保持 OFF。
 - T 定时器内容值如果小于或等于 m，且 D 的状态为 ON，D+1 的状态为 OFF。
 - T 定时器内容值小于 m，且 D、D+1、D+3 的状态为 ON，D+2 的状态为 OFF。
7. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。
8. m 操作数的范围为 1~32767

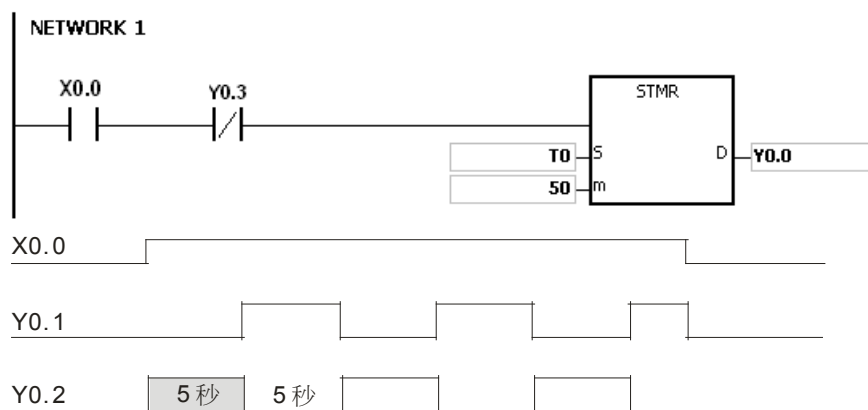
程序范例：

1. 当 X0.0=ON 时，STMR 指令指定定时器 T0，T0 的设定值为 5 秒。

2. Y0.0 为 OFF 延迟接点：当 X0.0 由 OFF 变 ON 时，Y0.0=ON，到 X0.0 由 ON 变 OFF 时，延迟 5 秒后 Y0.0=OFF。
3. Y0.1 于 X0.0 由 ON 变 OFF 时，作一次 5 秒 Y0.1=ON 输出。
4. Y0.2 于 X0.0 由 OFF 变 ON 时，作一次 5 秒 Y0.2=ON 输出。
5. Y0.3 于 X0.0 由 OFF 变 ON 时 延迟 5 秒后 Y0.3=ON 到 X0.0 由 ON 变 OFF 时 延迟 5 秒后 Y0.3=OFF。



6. 在条件接点 X0.0 后面加一个 Y0.3 的 b 接点，则 Y0.1、Y0.2 可作闪烁回路输出。当 X0.0 变成 OFF 时，Y0.0、Y0.1 及 Y0.3 变成 OFF，T0 的内容被复位为 0。



补充说明：

1. D+3 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. m≤0，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。

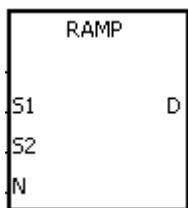
FB/FC	指令码			操作数				功能						
FC		RAMP		S_1, S_2, D, N				倾斜信号						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
D		●					●							
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●						●	●		●		●				
D	●	●						●	●		●		●				
N	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



- S_1 : 倾斜信号之起点设定
- S_2 : 倾斜信号之终点设定
- D : 倾斜信号的经过时间值
- n : 扫描次数

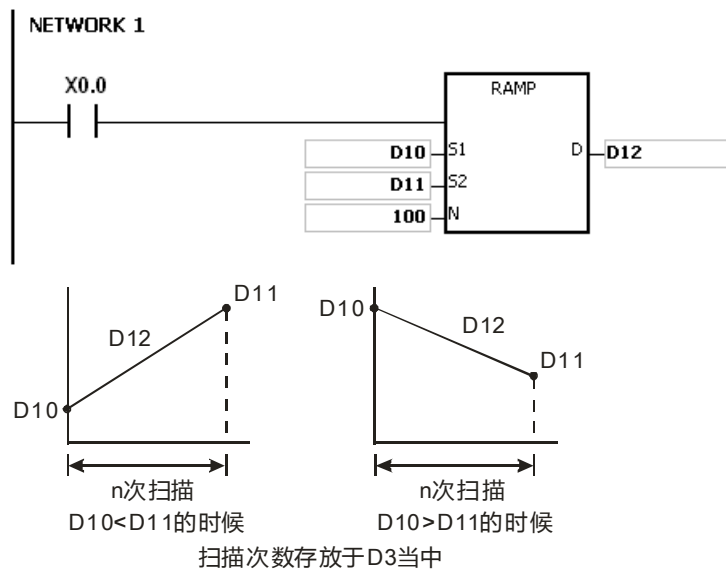
指令说明：

1. 本指令是一个求斜率的指令，斜率是线性与扫描时间有绝对的关系，因此使用本指令时，通常必须预先将扫描时间加以固定。
2. 预先将倾斜信号之起点设定值写入 S_1 及倾斜信号的终点设定值写入 S_2 内，当条件接点=ON 时，D 从 S_1 所设定的值朝 S_2 迈进（增加），扫描次数存放于 D+1 当中，当 $D = S_2$ 或 $D+1=n$ （已到达扫描次数）时，SM687 完成标志=ON。
3. 当条件接点未启动时，D、D+1 的内容值为 0，SM687 完成标志=OFF。
4. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。
5. 若需要固定扫描时间，请参考 ISPSOFT 使用手册中的硬件组态工具说明。
6. n 操作数的范围为 1~32767。

程序范例：

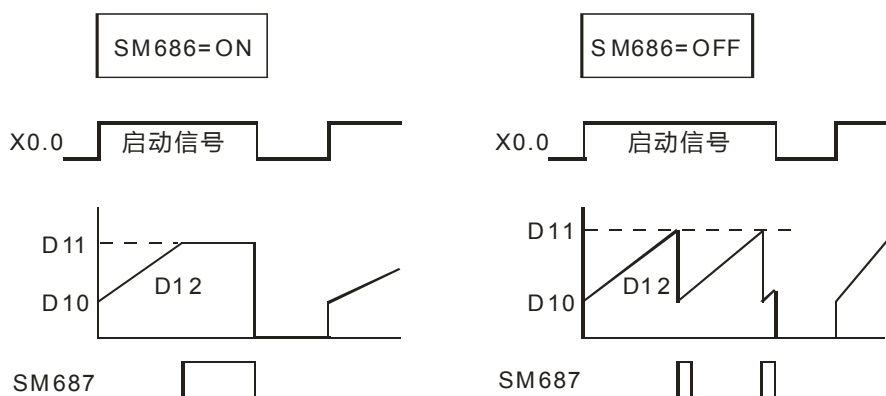
本指令若是与模拟信号输出搭配使用时，可执行缓冲启动/停止的动作。

1. 指令执行中,启动信号 X0.0 变成 OFF 时,指令停止执行,当 X0.0 再度 ON 的时候,SM687=OFF, D12 的内容被复位为 D10 的设定值, D13 的内容被清除为 0,再重新计算。
2. SM686=OFF 时, SM687=ON, D12 的内容被复位成 D10 的设定值, D13 的内容被清除为 0。



补充说明：

1. D+1 超出装置范围,指令不执行, SM0=ON, 错误码 SR0=16#2003。
2. $n \leq 0$, 指令不执行, SM0=ON, 错误码 SR0=16#200B。
3. D 操作数,若使用 ISPSOft 宣告,则数据类型为 ARRAY [2] of WORD/INT。
4. 启动模态标志信号 SM686 的 ON/OFF, D12 的内容变化如下：



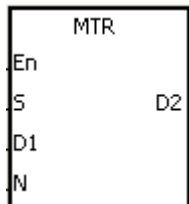
FB/FC	指令码		操作数				功能			
FC		MTR	S, D₁, D₂, N				矩阵输入			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●													
D₁	●													
D₂	●													
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	○																
D₁		○															
D₂	○	○	○	○				○	○				○				
N	●	●	●	●				●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



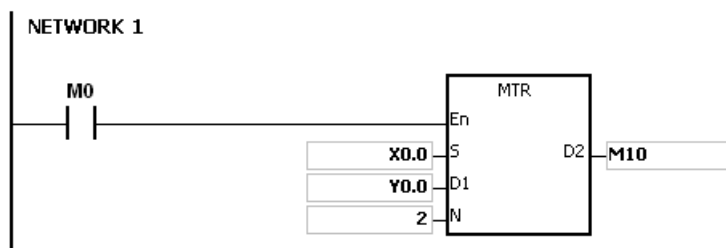
- S** : 矩阵扫描输入起始装置
- D₁** : 矩阵扫描输出起始装置
- D₂** : 矩阵扫描所对应起始装置
- n** : 矩阵扫描的列数

指令说明：

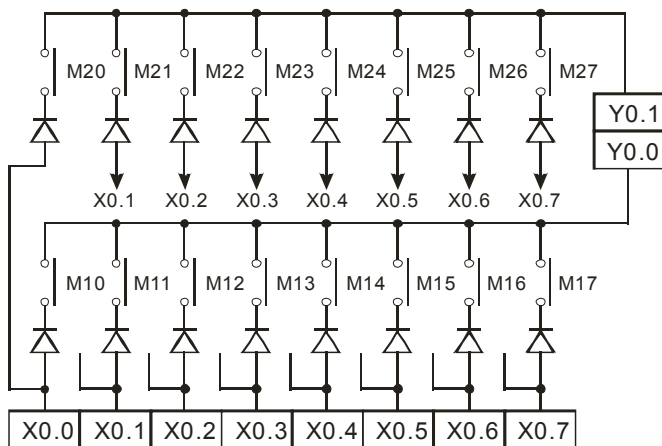
1. **S** 指定矩形所有连接输入端的起始号码，一旦指定后，该号码开始算连续 8 点为矩阵输入端。
2. **D₁** 指定那一个晶体管输出 Y 为矩阵扫描的起始号码，条件接点 OFF 时，**D₁** 起始的 **n** 个装置状态会保持 OFF。
3. 每一次扫描周期可以更新一列的输入点，每一列为 16 点，从 1~**n** 列反复的扫描。
4. 本指令由 **S** 起始的连续 8 个输入端，以 **D₁** 起始的 **n** 个外部输出点用矩阵扫描的方式读取 **n** 列的 8 个开关，变成 8×**n** 的多点矩阵输入点。并将扫描读取的开关状态反应在 **D₂** 起始之装置。
5. 使用本指令时，最大可将 8 个输入开关并接 8 列可得 64 个输入点 (8×8=64)。
6. 本指令的执行间隔必须比模块 I/O 更新的时间长，否则无法读到正确的输入点。
7. 本指令的条件接点一般都使用常 ON 接点 SM400。
8. **n** 操作数的范围 2~8。

程序范例一：

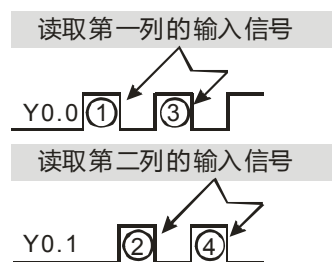
1. 当 M0=ON ,MTR 指令开始执行 ,外部 2 列共 16 个开关的状态被顺序读取并存放在内部继电器 M10~M17 , M20~M27。



2. 下图由 X0.0~X0.7 及 Y0.0~Y0.7 构成 2 列矩阵输入回路的外部接线图 ,16 个开关对应到内部继电器 M10~M17 , M20~M27。搭配 MTR 指令使用。



3. 以上图为例 ,X0.0 开始算的 8 点由 Y0.0~Y0.1 (n=2) 作矩阵扫描 ,D₂ 指定读入结果的起始号码为 M10 ,代表第一列的开始被读入至 M10~M17 ,第二列被读入至 M20~M27。



补充说明 :

1. 本指令被执行时 ,当扫描周期太长或太短都可能造成击键不良因此可运用下列技巧来克服。
2. 当扫描周期太短时 ,可能造成 I/O 来不及反应而无法读取正确之击键 ,此时 ,可将扫描时间加以固定。
3. 当扫描周期太长时 ,可能会使按键反应变得迟钝 ,可将此指令写在时间中断工作 (TASK) 内 ,固定时间执行此指令。
4. S+7、D₁+n-1、D₂+ (n*8) -1 超出装置范围 ,指令不执行 ,SM0=ON ,错误码 SR0=16#2003。
5. n<2 或 n>8 时 ,指令不执行 ,SM0=ON ,错误码 SR0=16#200B。
6. S 操作数 ,若使用 ISPSOft 宣告 ,则数据类型为 ARRAY [8] of BOOL。

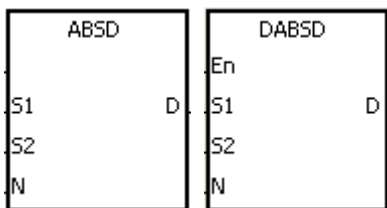
FB/FC	指令码			操作数					功能						
FC	D*	ABSD		S ₁ , S ₂ , D, N					绝对方式凸轮控制						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D	●/●*													
N		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●		●				
D	●	●	●	●				●	●	●			●				
N	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	AH 运动控制主机

图形：



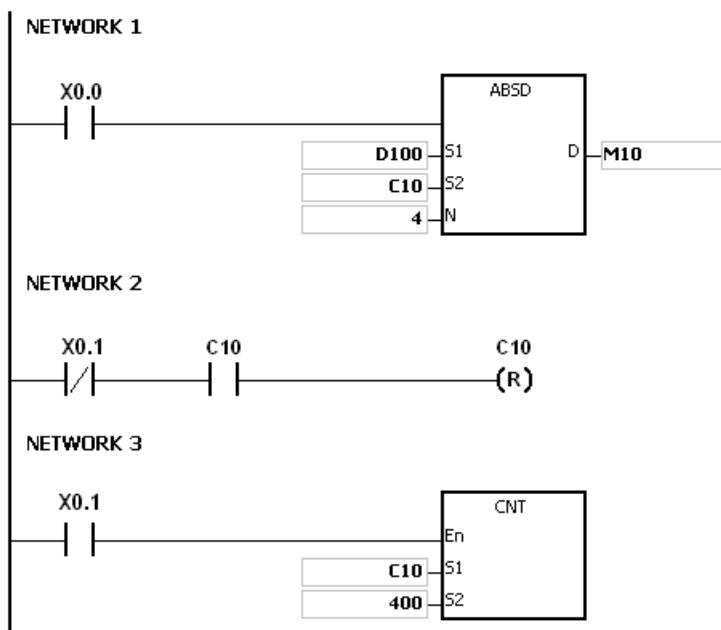
- S₁ : 指比较表起始装置
- S₂ : 比较值
- D : 比较结果起始编号
- n : 多段比较的组数

指令说明：

1. ABSD 指令为为对应计数器的当前值产生多个输出波形的指令，通常用来做绝对方式凸轮控制。
2. DABSD 指令才可以使用 HC 装置。
3. ABSD 时 n=1~256，DABSD 时 n=1~128。

程序范例一：

1. 于 ABSD 指令被执行前使用 MOV 指令预先将各设定值写入至 D100~D107。偶数 D 号码的内容为下限值，奇数 D 号码的内容为上限值。
2. 当 X0.0=ON 时，计数器 C10 的当前值与 D100~D107 等 4 组上下限值作区域比较，结果分别反应在 M10~M13。
3. X0.0=OFF 时，原 M10~M13 的 ON/OFF 状态不会变化。

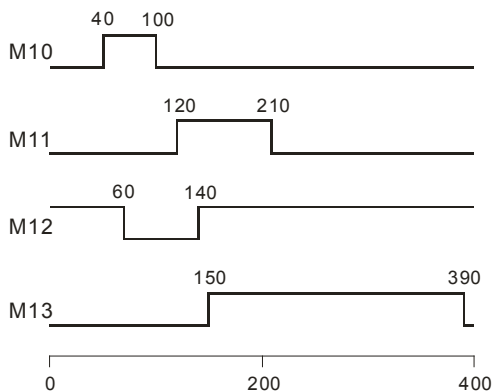


4. 在大于等于下限值且小于等于上限值范围内对应的 M10~M13 会 ON，否则为 OFF。

下限值	上限值	C10 当前值	输出
D100=40	D101=100	$40 \leq C10 \leq 100$	M10=ON
D102=120	D103=210	$120 \leq C10 \leq 210$	M11=ON
D104=140	D105=170	$140 \leq C10 \leq 170$	M12=ON
D106=150	D107=390	$150 \leq C10 \leq 390$	M13=ON

5. 若下限值大于上限值时，则小于上限值 ($C10 < 60$) 或大于下限值 ($C10 > 140$) 时，M12=ON，否则为 OFF。

下限值	上限值	C10 当前值	输出
D100=40	D101=100	$40 \leq C10 \leq 100$	M10=ON
D102=120	D103=210	$120 \leq C10 \leq 210$	M11=ON
D104=140	D105=60	$60 \leq C10 \leq 140$	M12=OFF
D106=150	D107=390	$150 \leq C10 \leq 390$	M13=ON



补充说明：

1. ABSD 指令中， $S+2*n-1$ 超出装置范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
2. DABSD 指令中， $S+4*n-1$ 超出装置范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
3. ABSD 指令中， $D+n-1$ 超出装置范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
4. DABSD 指令中， $D+2*n-1$ 超出装置范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
5. ABSD 指令中，若 $n<1$ 或 $n>256$ 时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200B$ 。
6. DABSD 指令中，若 $n<1$ 或 $n>128$ 时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#200B$ 。

1.

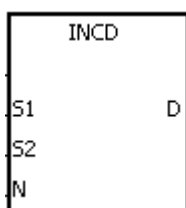
FB/FC	指令码		操作数				功能				
FC		INCD	S_1, S_2, D, N				相对方式凸轮控制				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
D	●													
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●		●	●		●		●				
D	●	●	●	●				●	●	●			●				
N	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：

 S_1 : 指比较表起始装置 S_2 : 计数器编号

D : 比较结果起始编号

n : 多段比较的组数

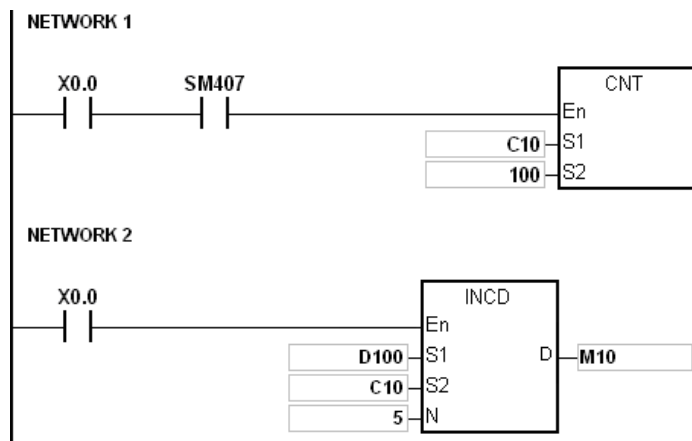
指令说明：

- INCD 指令为用一对计数器产生多个输出波形的指令，通常用来作相对方式凸轮控制。
- S_2 的当前值与 S_1 的设定值作比较，每到达一个设定值， S_2 的当前值被复位为 0 重新计数。而目前执行之段数被暂存于 S_2+1 当中。
- n 的组数比较完成时，指令执行完毕标志 SM688 会 ON 一次扫描周期。
- 当条件接点未启动时， S_2 、 S_2+1 的内容值为 0， $D-D+n-1$ 状态为 OFF，SM688 为 OFF。
- 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。
- n 操作数的范围 1~256。

程序范例：

- 于 INCD 指令被执行前，使用 MOV 指令预先将各设定值写入至 D100~D104 当中，D100=15、D101=30、D102=10、D103=40、D104=25。
- 计数器 C10 的当前值与 D100~D104 的设定值作比较，每到达一个设定值，C10 的当前值被复位为 0 重新计数。

3. 而目前执行之段数被暂存于 C11 当中。
4. C11 的内容每变动 1 时，M10~M14 相对应动作，请参考下列时序图。
5. 5 组数比较完成时，指令执行完毕标志 SM688 会 ON 一次扫描周期。
6. 当 X0.0 由 ON 变成 OFF 时，C10 及 C11 全部被复位为 0，M10~M14 亦全部变成 OFF，当 X0.0 再度 ON 时，本指令被从头执行起。



补充说明：

1. S_2+1 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. S_1+n-1 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. $D+n-1$ 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
5. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。

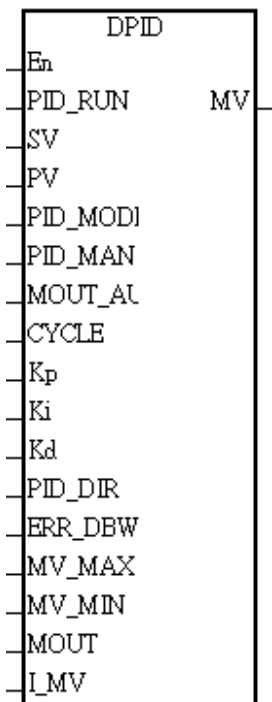
FB/FC	指令码			操作数						功能				
FC	D*	PID		参照下表						PID 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
PID_RUN	●*													
SV										●*				
PV										●*				
PID_MODE			●*					●*						
PID_MAN	●*													
MOUT_AUTO	●*													
CYCLE			●*					●*						
Kp										●*				
Ki										●*				
Kd										●*				
PID_DIR	●*													
ERR_DBW										●*				
MV_MAX										●*				
MV_MIN										●*				
MOUT										●*				
I_MV										●*				
MV										●*				

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
PID_RUN	●	●	●					●	●	●			●				
SV								●	●				●				
PV								●	●				●				
PID_MODE								●	●				●				
PID_MAN	●	●	●					●	●	●			●				
MOUT_AUTO	●	●	●					●	●	●			●				
CYCLE								●	●				●				
Kp								●	●				●				
Ki								●	●				●				
Kd								●	●				●				
PID_DIR	●	●	●					●	●	●			●				
ERR_DBW								●	●				●				
MV_MAX								●	●				●				
MV_MIN								●	●				●				
MOUT								●	●				●				
I_MV								●	●				●				
MV								●	●				●				

脉冲执行型	16 位指令	32 位指令
-	-	AH 运动控制主机

图形：



- PID_RUN** : 启动 PID 运算
- SV** : 目标值 (SV)
- PV** : 当前值 (PV)
- PID_MODE** : PID 控制模式
- PID_MAN** : PID A/M 模式 (PID_MAN)
- MOUT_AUTO** : MOUT_AUTO
- CYCLE** : 取样时间 (CYCLE)
- Kp** : P 计算值系数 (Kp)
- Ki** : I 计算值系数 (Ki)
- Kd** : D 计算值系数 (Kd)
- PID_DIR** : PID 正反向 (PID_DIR)
- ERR_DBW** : ERR 的不作用范围 (ERR_DBW)
- MV_MAX** : MV 上限值 (MV_MAX)
- MV_MIN** : MV 下限值 (MV_MIN)
- MOUT** : MV 手动值 (MOUT)
- I_MV** : 累计积分项的数值 (I_MV)
- MV** : MV 输出值 (MV)

指令说明：

- PID 运算控制的专用指令，于取样时间到达后的该次扫描才执行 PID 运算动作。PID 表示“比例、积分和微分”。PID 控制在机械设备、气动设备和电子设备中具有广泛的应用。
- 参数设定内容如下：

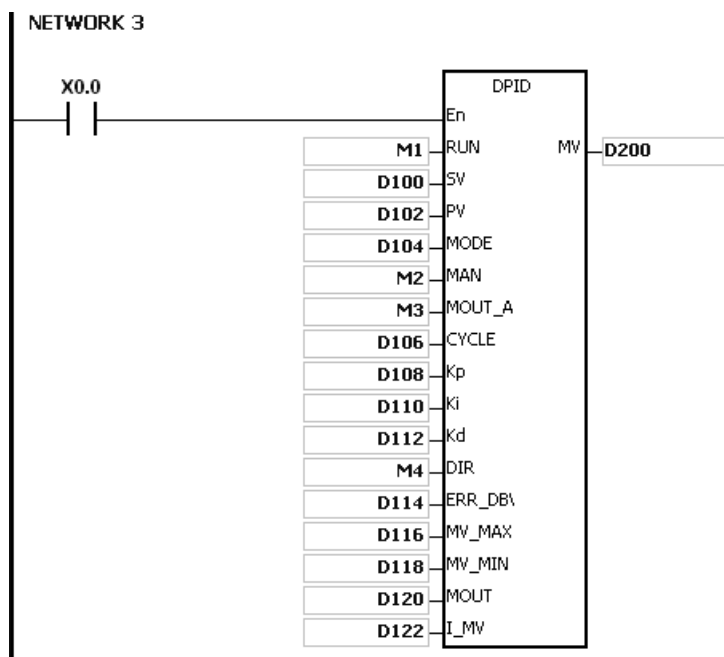
装置编号	功能	设定范围	说明
PID_RUN	启动 PID 运算	TRUE : PID 指令开始运算。 FALSE : MV 值清除为 0，PID 不运算。	
SV	SV 值	单精度浮点 数范围	目标值
PV	PV 值	单精度浮点 数范围	当前值

装置编号	功能	设定范围	说明
PID_MODE	PID 控制模式	0 : 自动控制, 当 PID_MAN 由 TRUE 转为 FALSE 时, MV 值会由当时输出的 MV 值开始进行自动运算。 1 : 温度控制专用的自动调整参数功能, 调整完毕时将自动改为 0, 并且填入最适用的 K_P 、 K_I 及 K_D 等参数。 2 : 与 0 相同, 但当 PID_MAN 由 TRUE 转为 FALSE 时, MV 值会由当时内部运算的 MV 值开始进行自动运算。 设定值超出范围, 将视为 0。	
PID_MAN	PID A/M 模式	TRUE : Manual, MV 值会依 MOUT 值输出, 但仍在 MV_MAX 与 MV_MIN 之间, 当 PID_MODE 为 1 时此设定无效。 FALSE : Auto, MV 值会依 PID 公式计算后输出, 输出值在 MV_MAX 与 MV_MIN 之间。	
MOUT_AUTO	MOUT Auto change 模式	TRUE : Auto, MOUT 的值会随着 MV 值改变。 FALSE : Normal, MOUT 的值不会随着 MV 值改变。	
Cycle	取样时间 (T_s)	1~2,000 (单位:10ms)	每次扫描到本指令时, 就会以 Cycle 设定的取样时间来计算 PID (不会计算扫描时间是否有到达取样时间), 并更新输出值 (MV)。 T_s 小于 1 则内定为 1, 大于 2000 则为 2000。 当 PID 指令用于 Interval Interrupt Task 时, Cycle 时间以 Interval Interrupt Task 的时间为主。
Kp	比例增益 (K_P)	正数单精度浮点数范围	为 SV-PV 间的误差放大比例值, 如果小于 0, K_P 将为 0
Ki	积分增益 (K_I)	正数单精度浮点数范围	I 计算值系数 (K_I), 如果小于 0, K_I 将为 0
Kd	微分增益 (K_D)	正数单精度浮点数范围	D 计算值系数 (K_D), 如果小于 0, K_D 将为 0
PID_DIR	PID 正反向	TRUE : 反向动作 ($E=SV-PV$) FALSE : 正向动作 ($E=PV-SV$)	
ERR_DBW	偏差量 (E) 不作用范围	单精度浮点数范围	偏差量 (E) 等于 SV-PV 的误差值, 当设定值为 0 即表示不启动此功能。例: 设定值为 5 或 -5, 则 E 在 -5~5 的区间偏差量将为 0

装置编号	功能		设定范围	说明
MV_MAX	输出值 (MV) 饱和上限		单精度浮点数范围	例：设定 1,000，则输出值 (MV) 大于 1,000 时将以 1,000 输出 需大于 S3+7，否则上限值与下限值将互换
MV_MIN	输出值 (MV) 饱和下限		单精度浮点数范围	例：设定-1,000，则输出值 (MV) 小于 -1,000 时将以-1,000 输出
MOUT	MV 手动值		配合 PID_MAN 模式使用，用户直接设定 MV 输出值。	
I_MV (连续占用 6 个 DWord 装置)	I_MV	暂存累积的积分值	单精度浮点数范围	为累积的积分值，通常只供参考用，但是用户还是可以依需求清除或修改，不过须以单精度浮点数修改之，当 MV 超出 MV_MAX 或 MV_MIN 时，I_MV 值不会再改变
	I_MV+1	暂存前次 PV 值	为前次测定值，通常只供参考用，但是用户还是可以依需求修改。	
	I_MV+2 ~ I_MV+5	系统用参数，用户请勿使用。		
MV	MV 输出值		MV 值会介于 MV_MAX 与 MV_MIN 之间。	

程序范例：

1. 执行 PID 指令前先将参数设定完成。
2. X0.0=ON 的时候指令被执行，M1=ON 的时候 PID 指令才开始进行运算，M1=OFF 时 MV 值为 0，MV 值存于 D200 中。X0.0 变成 OFF 时，指令不被执行，之前的数据没有变化。



补充说明：

1. 指令无使用次数的限制，但是 $I_MV \sim I_MV_{+5}$ 所指定的寄存器号码不可重复。
2. I_MV 占用 12 个寄存器，于上述程序例当中 I_MV 指定 PID 指令的参数设定区域为 D122~D133。
3. PID 指令只能使用在 Cyclic Task 与 Interval Interrupt Task 当中，当指令用于 Interval Interrupt Task 时，Cycle 时间以 Interval Interrupt Task 的时间为主。
4. PID 被启动后，每次只要被扫描到都会以 Cycle 所设定的时间来做 PID 运算并直接输出 MV 值，并不会自动计算扫描时间是否有到达取样时间才输出，但如果本指令使用在时间中断程序内，取样时间将与时间中断程序的间隔时间相同，PID 指令会以时间中断程序的中断间隔时间来做 PID 的运算。
5. PID 的测定值 (PV) 在 PID 执行运算动作前必须是一个稳定值。如果要抓取特殊模块的输入值作 PID 运算时，请注意这些模块的 A/D 转换时间。

PID 计算公式：

1. 当 PID_MODE 控制模式选择为 0、2，自动控制模式。
2. 当 PID_MODE 控制模式选择为 1，自动调整模式，当自动调整完成， PID_MODE 会变成 0 自动控制模式。
 - a) PID 的运算分成正向动作，逆向动作。而正逆向动作由 DIR 来指定。

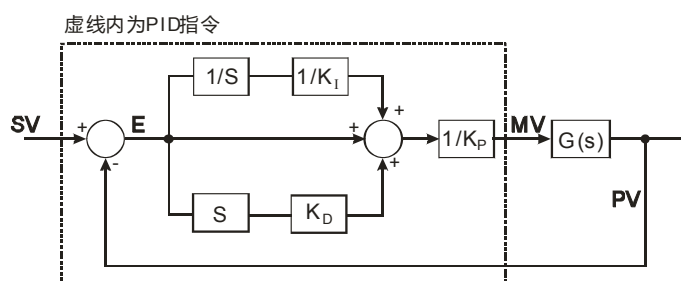
PID 的表达式：

$$MV = K_p E(t) + K_I \int_0^t E(t) dt + K_D * \frac{dE(t)}{dt}$$

其中 $E(t)S$ 表示 $E(t)$ 的微分值，以及 $E(t) \frac{1}{S}$ 表示 $E(t)$ 的积分值。

动作方向	PID 运算方式
正向动作、自动	$E(t) = PV(t) - SV(t)$
逆向动作	$E(t) = SV(t) - PV(t)$

- 控制方块图：下图之 S 表示微分的动作，其动作定义为现在 $E(t)$ 值减去前次 $E(t)$ 值后，再除以取样时间的动作，另外 $1/S$ 表示积分的动作，其动作定义为前次积分值加上这次偏差量乘以取样时间的值，最后图中的 $G(S)$ 表示受控装置。



- 符号说明：

MV : 输出值

$E(t)$: 偏差量。正向动作 $E(t) = PV - SV$ ，反向动作 $E(t) = SV - PV$

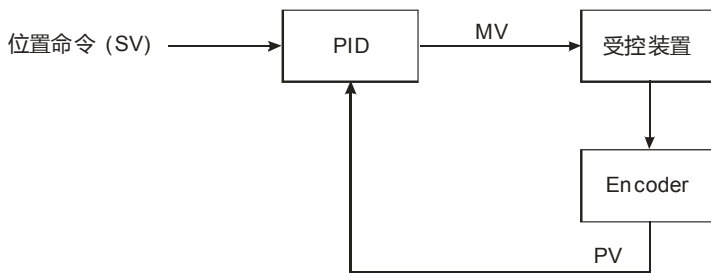
K_p : 比例增益

- PV : 测定值
- SV : 目标值
- K_D : 微分增益
- K_I : 积分增益

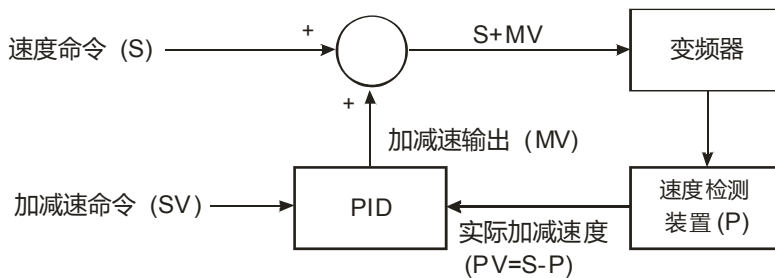
注意事项和建议：

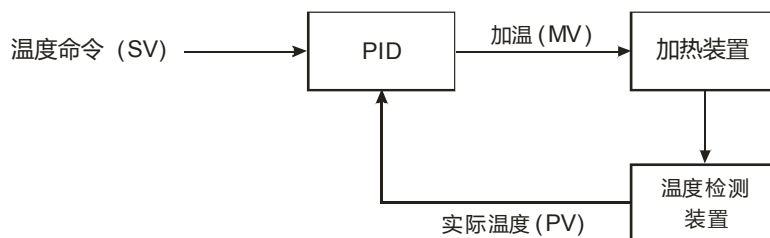
1. 由于可使用 PID 指令的控制环境很多，因此请适当的选取控制功能，例如：当选择自动调整参数（PID_MODE=1）功能时，就请勿使用于马达控制环境中，以免造成控制不当的现象发生。
2. 用户于调整 K_P 、 K_I 及 K_D 三个主要参数时（PID_MODE=0、2），请先调整 K_P 值（依经验值设定），而 K_I 及 K_D 值先设定为 0，等到调整到大致上可控制时，再依序调整 K_I 值（由小到大）以及 K_D 值（由小到大），调整范例如范例四所示。其中 K_P 值为 100 则表示 100%，即对偏差值的增益为 1，小于 100%将对偏差值衰减，大于 100%将对偏差值放大。
3. 当用户选用（PID_MODE=1）时，建议请使用在停电保持区之 D 寄存器来储存参数，以免自动调整过的参数因停电后而消失。经过自动调整过的参数，并不能保证一定适用于每个控制的环境，因此用户当然可自行修改调整过的参数，不过建议最好只修改 K_I 或 K_D 数值就好。
4. 本指令动作须配合许多参数值控制，因此请勿随意设定参数值，以免造成无法控制的现象。

范例 1：使用 PID 指令于位置控制时的方块图（PID_MODE=0、2）



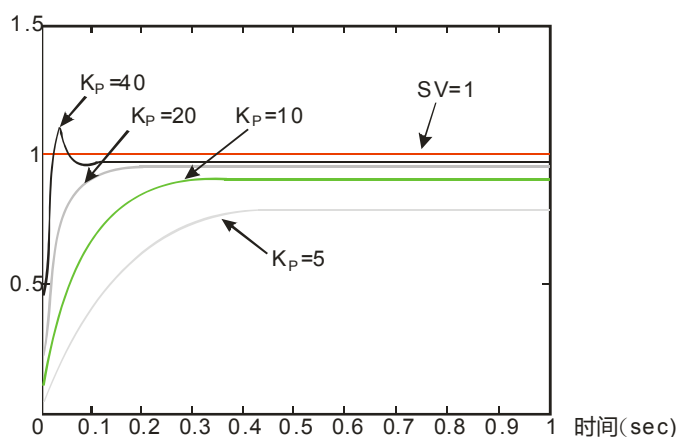
范例 2：与变频器搭配控制时的方块图（PID_MODE=0、2）



范例 3 : 使用 PID 指令于温度控制时的方块图 (PID_MODE=0、2)**范例 4 :** PID 指令参数调整建议步骤说明

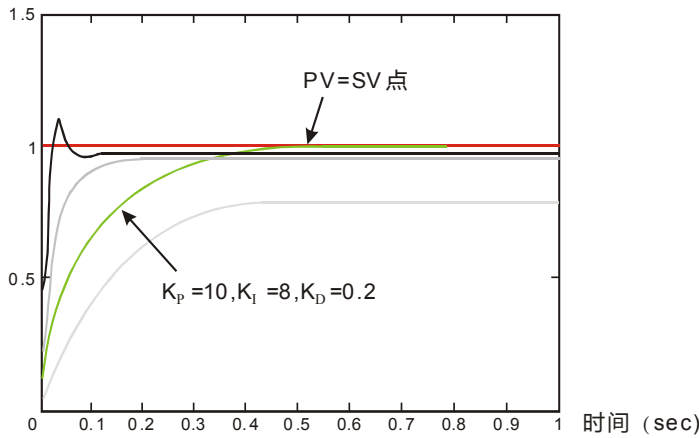
假设控制系统之受控体 $G(s)$ 的转移函数为一阶的函数 $G(s) = \frac{b}{s+a}$ (一般马达的模型均为此函数), 命令值 SV 为 1, 取样时间 T_s 为 10ms。建议调整步骤如下:

步骤 1 : 首先将 K_I 及 K_D 值设为 0, 接着先后分别设定 K_P 为 5、10、20 及 40, 并分别记录其 SV 及 PV 状态, 其结果如下图所示。



步骤 2 : 观察上图后得知 K_P 为 40 时, 其反应会有过冲现象, 因此不选用, 而 K_P 为 20 时, 其 PV 反应曲线接近 SV 值且不会有过冲现象, 但是由于启动过快, 因此输出值 MV 瞬间值会很大, 所以考虑暂不选用, 接着 K_P 为 10 时, 其 PV 反应曲线接近 SV 值并且是比较平滑接近, 因此考虑使用此值, 最后 K_P 为 5 时, 其反应过慢, 因此也暂不考虑使用。

步骤 3 : 选定 K_P 为 10 后, 先调整 K_I 值由小到大 (如 1、2、4 至 8), 以不超过 K_P 值为原则, 然后再调整 K_D 由小到大 (如 0.01、0.05、0.1 及 0.2), 以不超过 K_P 的 10% 为原则, 最后可得如下图之 PV 与 SV 的关系图。



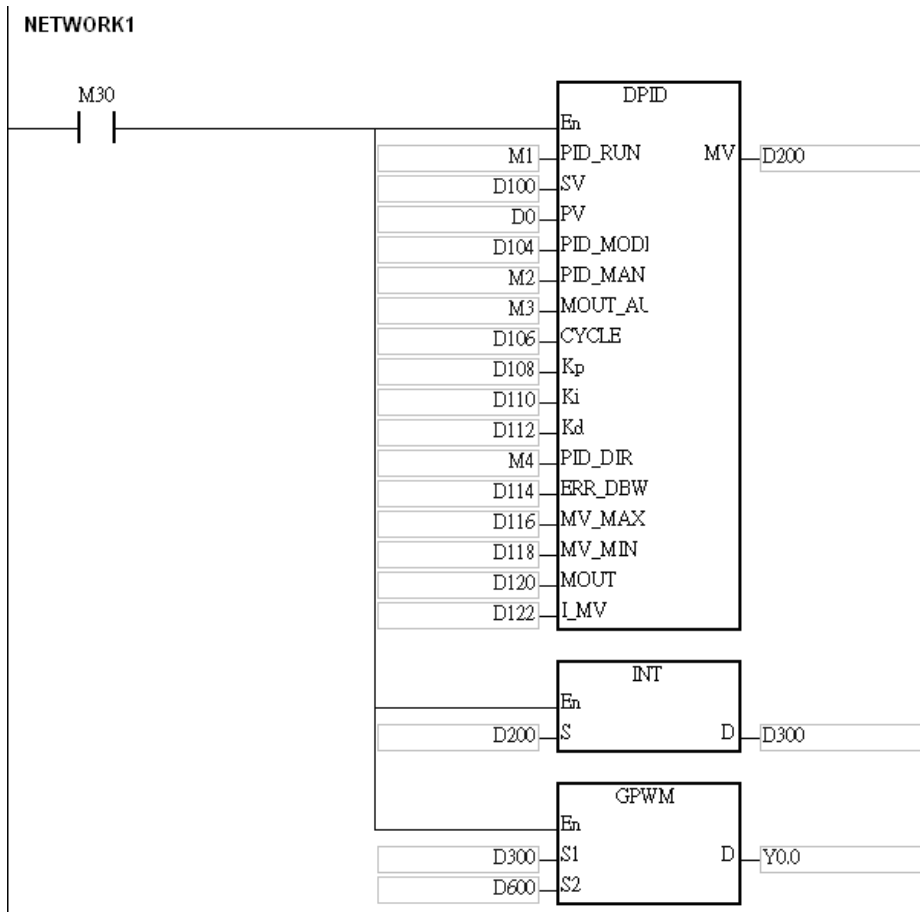
附注：本范例仅供参考，因此用户还需依实际控制系统的状况，自行调整其适合的控制参数。

实例 1：使用自动调整功能控制温度

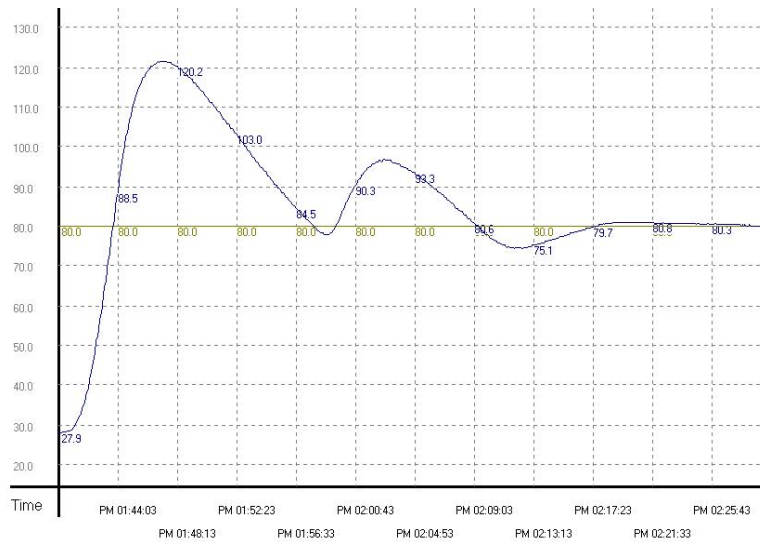
控制目的：利用自动调整功能计算出最佳的 PID 温度控制的参数。

控制说明：

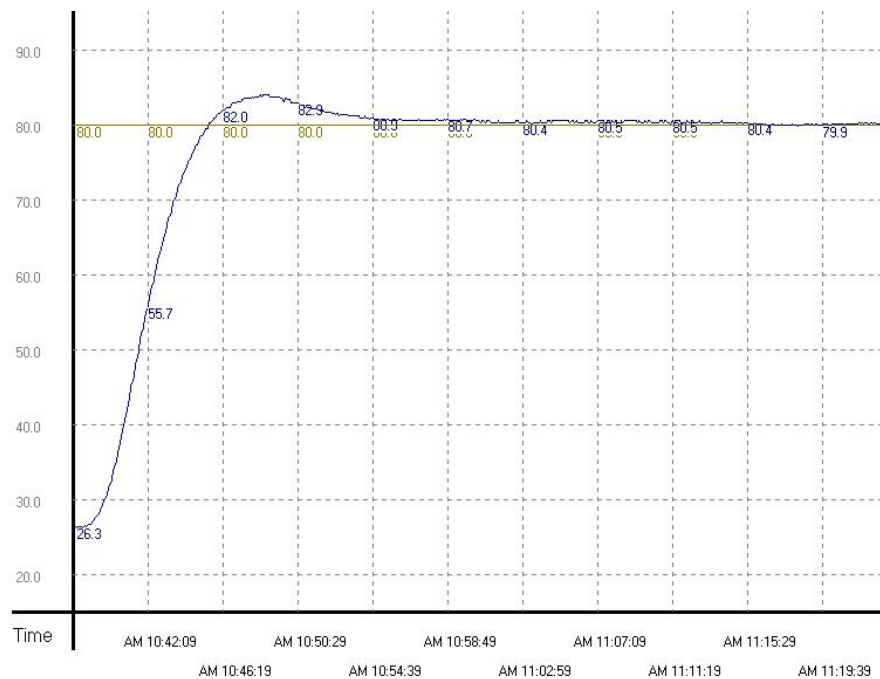
由于一般用户对于第一次控制的温度环境特性通常不太了解，因此可先使用自动调整功能（PID_MODE=1）做一初步调整，待调整完毕后，本指令将自动修改控制模式为（PID_MODE=0）。本实例的控制环境为烤箱。范例程序如下图所示：



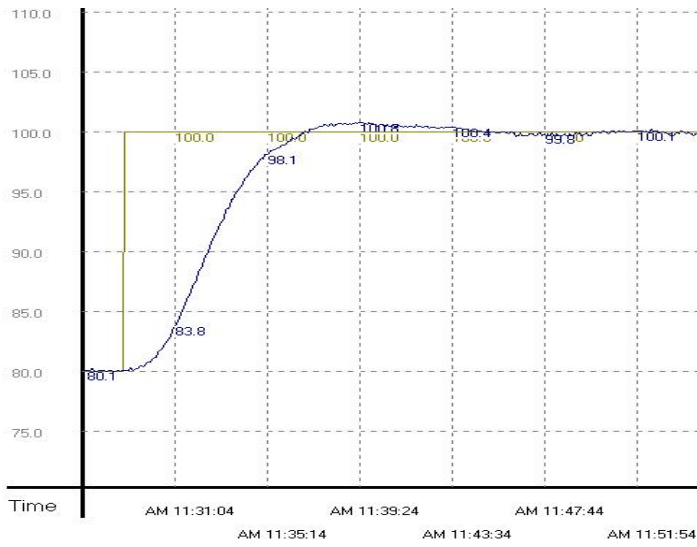
自适应功能的实验结果如下所示：



使用调整后参数做温度控制专用功能的实验结果如下所示：



由上图可看出经过自适应后的温度控制结果还不错，而且控制时间大约只使用了 20 分钟。接着验证目标温度由 80 度修改成 100 度，则得到的结果如下图所示：



3

由上图中可看出由 80 度所调整出来的参数使用到 100 度时，还是可以达到控制温度的目的，而且控制时间也不会太长。

FB/FC	指令码			操作数						功能				
FC	D*	PIDE		参照下表						PID 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
PID_RUN	●*													
SV										●*				
PV										●*				
PID_MODE			●*					●*						
PID_MAN	●*													
MOUT_AUTO	●*													
CYCLE			●*					●*						
KC_Kp										●*				
Ti_Ki										●*				
Td_Kd										●*				
Tf										●*				
PID_EQ	●*													
PID_DE	●*													
PID_DIR	●*													
ERR_DBW										●*				
MV_MAX										●*				
MV_MIN										●*				
MOUT										●*				
BIAS										●*				
I_MV										●*				
MV										●*				

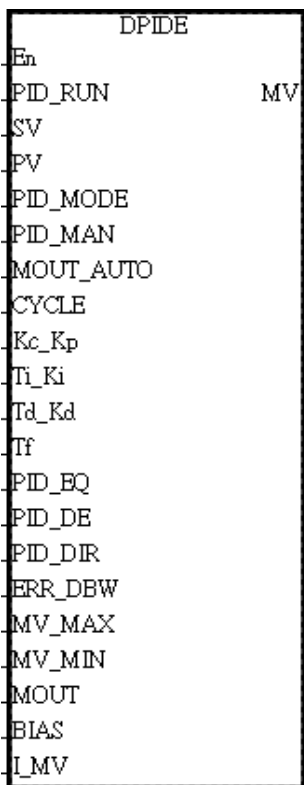
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
PID_RUN	●	●	●					●	●	●			●				
SV								●	●				●				●
PV								●	●				●				●
PID_MODE								●	●				●				
PID_MAN	●	●	●					●	●	●			●				
MOUT_AUTO	●	●	●					●	●	●			●				
CYCLE								●	●				●				
KC_Kp								●	●				●				
Ti_Ki								●	●				●				
Td_Kd								●	●				●				
Tf								●	●				●				
PID_EQ	●	●	●					●	●	●			●				
PID_DE	●	●	●					●	●	●			●				
PID_DIR	●	●	●					●	●	●			●				
ERR_DBW								●	●				●				●

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
MV_MAX								●	●				●				●
MV_MIN								●	●				●				●
MOUT								●	●				●				
BIAS								●	●				●				●
I_MV								●	●				●				
MV								●	●				●				

脉冲执行型	16 位指令	32 位指令
-	-	AH 运动控制主机

图形：

3



- PID_RUN** : 启动 PID 运算
- SV** : 目标值
- PV** : 当前值
- PID_MODE** : PID 控制模式
- PID_MAN** : PID Auto/Manual 模式
- MOUT_AUTO** : 手动值 (MOUT) 自动更新模式
- CYCLE** : 取样时间 (ms)
- Kc_Kp** : 比例项系数
- Ti_Ki** : 积分项系数 (sec 或 1/sec)
- Td_Kd** : 微分项系数 (sec)
- Tf** : 微分作用时间常数 (sec)
- PID_EQ** : PID 计算式选择
- PID_DE** : PID 微分项误差计算选择
- PID_DIR** : PID 正反向
- ERR_DBW** : ERR 的不作用范围
- MV_MAX** : MV 输出上限值
- MV_MIN** : MV 输出下限值
- MOUT** : MV 手动值
- BIAS** : 前馈控制输出值
- I_MV** : 积分项累计值
- MV** : 输出值

指令说明：

1. PID 进阶运算控制的专用指令，当指令被主机执行时才会进行 PID 运算与处理。PID 表示“比例、积分和微分”。PID 控制在机械设备、气动设备和电子设备中具有广泛的应用。
2. 参数设定内容如下：

装置编号	数据类型	功能	设定范围	说明
PID_RUN	BOOL	启动 PID 运算	TRUE：PID 指令开始运算。 FALSE：MV 值清除为 0，PID 不运算。	
SV	REAL	SV 值	单精度浮点数范围	目标值
PV	REAL	PV 值	单精度浮点数范围	当前值
PID_MODE	DWORD/DINT	PID 控制模式	0：自动控制，当 PID_MAN 由 TRUE 转为 FALSE 时，MV 值会由当时输出的 MV 值开始进行自动运算。 1：温度控制专用的自动调整参数功能，调整完毕时将自动进入自动控制模式（PID_MODE 改为 0），并且填入最适用的 Kc_Kp、Ti_Ki、Td_Kd 及 Tf 等参数。	
PID_MAN	BOOL	PID A/M 模式	TRUE：Manual，MV 值会依 MOUT 值输出，但仍在 MV_MAX 与 MV_MIN 之间，当 PID_MODE 为 1 时此设定无效。 FALSE：Auto，MV 值会依 PID 公式计算后输出，输出值在 MV_MAX 与 MV_MIN 之间。	
MOUT_AUTO	BOOL	MOUT 自动更新模式	TRUE：Auto，MOUT 的值会随着 MV 值改变。 FALSE：Normal，MOUT 的值不会随着 MV 值改变。	
CYCLE	DWORD/DINT	取样时间 (T _s)	1~40,000 (单位: ms)	每次扫描到本指令时，就会以 CYCLE 设定的取样时间来计算 PID（主机并不会依照此数值来自动判断时间并自动执行），并更新输出值 (MV)。CYCLE 小于 1 则内定为 1，大于 40,000 时则为 40,000。 当 PID 指令用于时间中断时，主机会自动以时间中断的中断时间来计算 PID，此时 CYCLE 的设定无效。

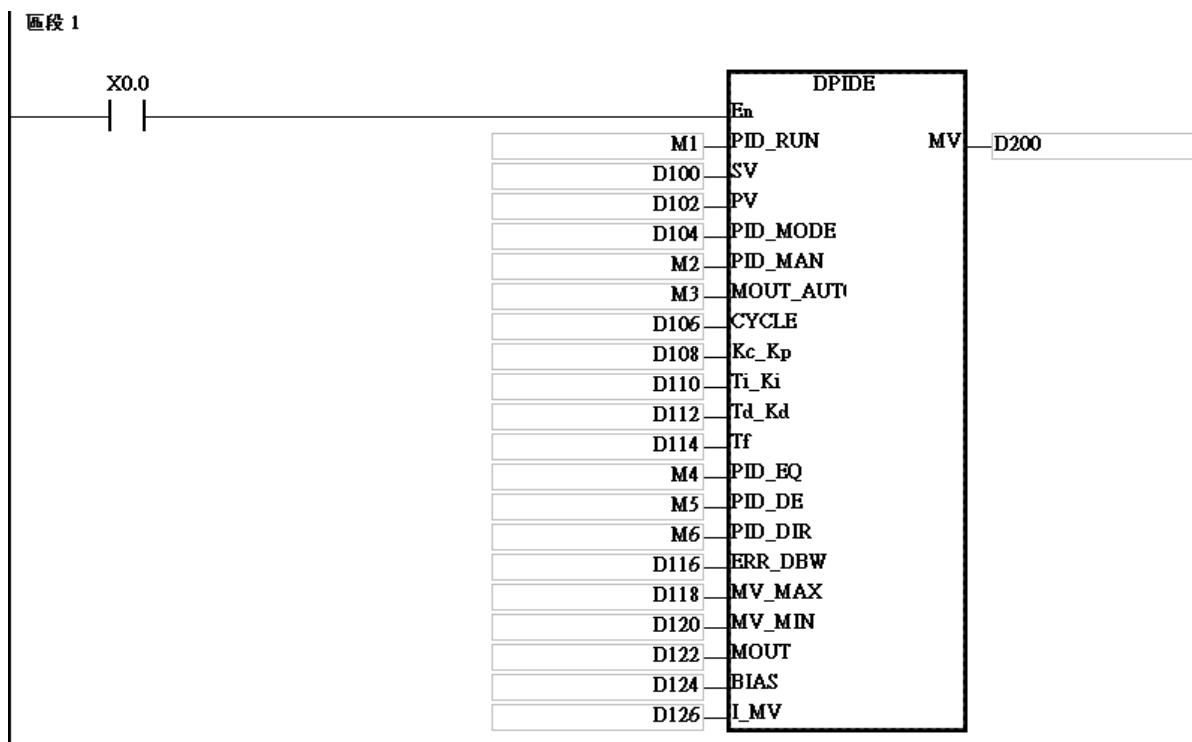
装置编号	数据类型	功能	设定范围	说明
Kc_Kp	REAL	比例项系数 (Kc or Kp, 依 PID_EQ 参数决定使用何种系数)	正数单精度浮点数范围	P 计算值系数, 如果小于 0 则 Kc_Kp 将为 0, 在 Independent 下若 Kc_Kp 等于 0 则表示不使用 P 控制。
Ti_Ki	REAL	积分项系数 (Ti or Ki, 依 PID_EQ 参数决定使用何种系数)	正数单精度浮点数范围 (单位: Ti = sec; Ki = 1/sec)	I 计算值系数, 如果小于 0 则 Ti_Ki 将为 0, 当 Ti_Ki 等于 0 时则表示不使用 I 控制。
Td_Kd	REAL	微分项系数 (Td or Kd, 依 PID_EQ 参数决定使用何种系数)	正数单精度浮点数范围 (单位: sec)	D 计算值系数, 如果小于 0 则 Td_Kd 将为 0, 当 Td_Kd 等于 0 时则表示不使用 D 控制。
Tf	REAL	微分作用时间常数 (Tf)	正数单精度浮点数范围 (单位: sec)	微分作用时间常数, 如果小于 0 则 Tf 将为 0, 当 Tf 等于 0 时则表示不使用微分作用时间的控制 (Derivative Smoothing)。
PID_EQ	BOOL	PID 计算式选择	TRUE : Dependent Formula FALSE : Independent Formula	
PID_DE	BOOL	PID 微分项误差计算选择	TRUE : 使用当前值 (PV) 的变化量来计算微分项的控制值 (Derivative of PV) FALSE : 使用偏差量 (E) 的变化量来计算微分项的控制值 (Derivative of E)	
PID_DIR	BOOL	PID 正反向	TRUE : 反向动作 (E=SV-PV) FALSE : 正向动作 (E=PV-SV)	
ERR_DBW	REAL	偏差量 (E) 不作用范围	单精度浮点数范围	偏差量 (E) 等于 SV-PV 或 PV-SV, 当 ERR_DBW 设定为 0 时即表示不启动此功能, 否则主机会去检查这次的 E 值是否小于 ERR_DBW 的绝对值且是否符合 Cross 状态转换条件, 若都有则将 E 值视为 0 之后进行 PID 计算, 否则依照正常处理将 E 值代入 PID 计算。

装置编号	数据类型	功能		设定范围	说明	
MV_MAX	REAL	输出值 (MV) 饱和上限		单精度浮点数范围	例：设定 1,000，则输出值 (MV) 大于 1,000 时将以 1,000 输出。 此设定值需大于 MV_MIN，否则主机会自动将 MV_MAX 与 MV_MIN 进行互换。	
MV_MIN	REAL	输出值 (MV) 饱和下限		单精度浮点数范围	例：设定-1,000，则输出值 (MV) 小于-1,000 时将以 -1,000 输出。 此设定值需小于 MV_MAX，否则主机会自动将 MV_MAX 与 MV_MIN 进行互换。	
MOUT	REAL	MV 手动值		单精度浮点数范围	配合 PID_MAN 模式使用，当 PID 设定在手动模式 (PID_MAN=True) 时，则输出值 (MV) 会依照手动设定值 (MOUT) 输出，但是仍然在 MV_MAX 与 MV_MIN 之间。	
BIAS	REAL	前馈控制输出值		单精度浮点数范围	使用于 PID 前馈控制。	
I_MV(连续占用 10 个 DWORD 装置)	REAL	I_MV	暂存累积的积分值	单精度浮点数范围	为累积的积分值，通常只供参考用，但是用户还是可以依需求清除或修改，不过须以单精度浮点数修改之，当 MV 超出 MV_MAX 或 MV_MIN 时，I_MV 值不会再改变	
		I_MV+1	暂存前次 E 值	系统纪录前次偏差量。		
		I_MV+2 ~ I_MV+5	系统用参数，用户请勿使用。			
		I_MV+6	系统纪录前次 PV 值。			
		I_MV+7 ~ I_MV+9	系统用参数，用户请勿使用。			

装置编号	数据类型	功能	设定范围	说明
MV	REAL	MV 输出值	MV 值会介于 MV_MAX 与 MV_MIN 之间。	

程序范例：

1. 执行 DPIDE 指令前先将参数设定完成。
2. X0.0=ON 的时候指令被执行，M1=ON 的时候 DPIDE 指令才开始进行运算，M1=OFF 时 MV 值为 0，MV 值存于 D200 中。X0.0 变成 OFF 时，指令不被执行，之前的数据没有变化。

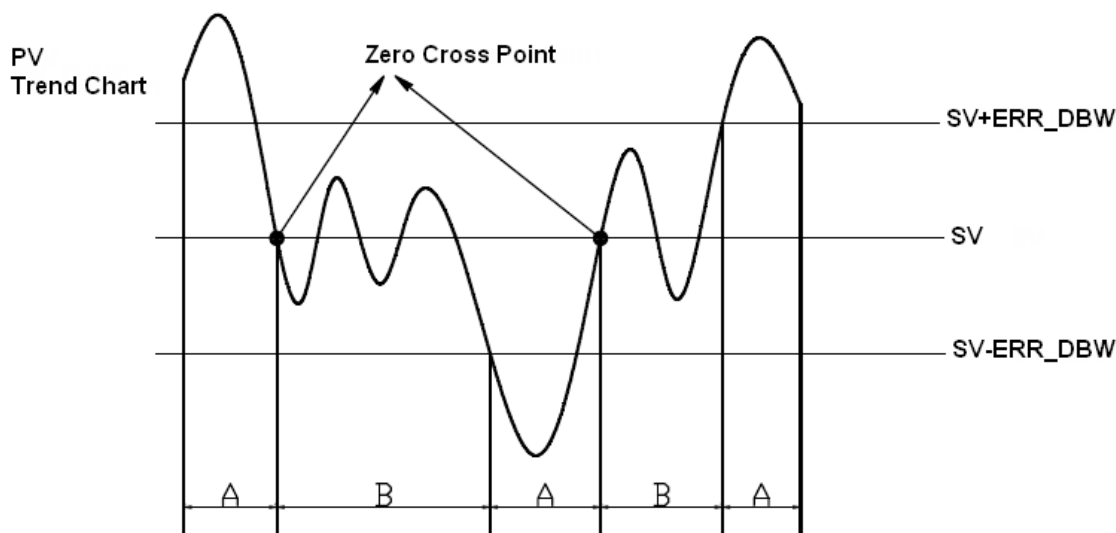


补充说明：

1. 指令无使用次数的限制，但是 I_MV~I_MV+9 所指定的寄存器号码不可被其他程序重复使用。
2. I_MV 占用 20 个 Word 寄存器，上述程序例当中 I_MV 所占用的设定区域为 D126~D145。
3. DPIDE 指令只能使用在周期性工作 (Cyclic Task) 与时间中断当中，当指令用于时间中断时，取样时间 (CYCLE) 会自动以时间中断的时间为主。
4. DPIDE 只要被扫描到都会以取样时间 (CYCLE) 所设定的时间来做 PID 运算并直接输出 MV 值，并不会自动计算扫描时间是否有到达取样时间 (CYCLE) 才输出，但如果本指令使用在时间中断程序内，取样时间 (CYCLE) 将与时间中断程序的间隔时间相同，DPIDE 指令会以时间中断程序的中断间隔时间来做 PID 的运算。
5. DPIDE 的当前值 (PV) 在 PID 执行运算动作前必须是一个稳定值。如果要抓取特殊模块的输入值作 PID 运算时，请注意这些模块的 A/D 转换时间。

6. 当 PV 值进入 ERR_DBW 的范围时,一开始主机仍会依照 E 值进行 PID 计算,直到 PV 穿过 SV 值时(Zero Cross Point)代表 Cross Status 成立,此时会将 E 值视为 0 代入 PID 计算,一直到 PV 值超出 ERR_DBW 的范围时才会恢复将 E 值代入 PID 计算,若 PID_DE=True 则表示使用 PV 值来进行微分项的计算,则在 Cross Status 条件成立后,主机会将 Delta PV 视为 0 进行 PID 微分项的计算。(Delta PV=当前 PV-前次 PV)

例如以下的 PV 趋势图中, A 的区段主机会依照正常的 PID 进行计算,而 B 的区段主机会将 E 或 Delta PV 视为 0 进行 PID 计算。



PID 计算公式：

1. 当 PID_MODE 控制模式设定为 0 时,为自动控制模式。

- Independent Formula & Derivative of E (PID_EQ=False & PID_DE=False)

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

- Independent Formula & Derivative of PV (PID_EQ=False & PID_DE=True)

$$CV = K_p E + K_i \int_0^t E dt - K_d \frac{dPV}{dt} + BIAS$$

$$E = SV - PV$$

or

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dPV}{dt} + BIAS$$

$$E = PV - SV$$

- Dependent Formula & Derivative of E (PID_EQ=True & PID_DE=False)

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dE}{dt} \right] + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

- Dependent Formula & Derivative of PV (PID_EQ=True & PID_DE=True)

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt - T_d \frac{dPV}{dt} \right] + BIAS$$

$$E = SV - PV$$

or

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dPV}{dt} \right] + BIAS$$

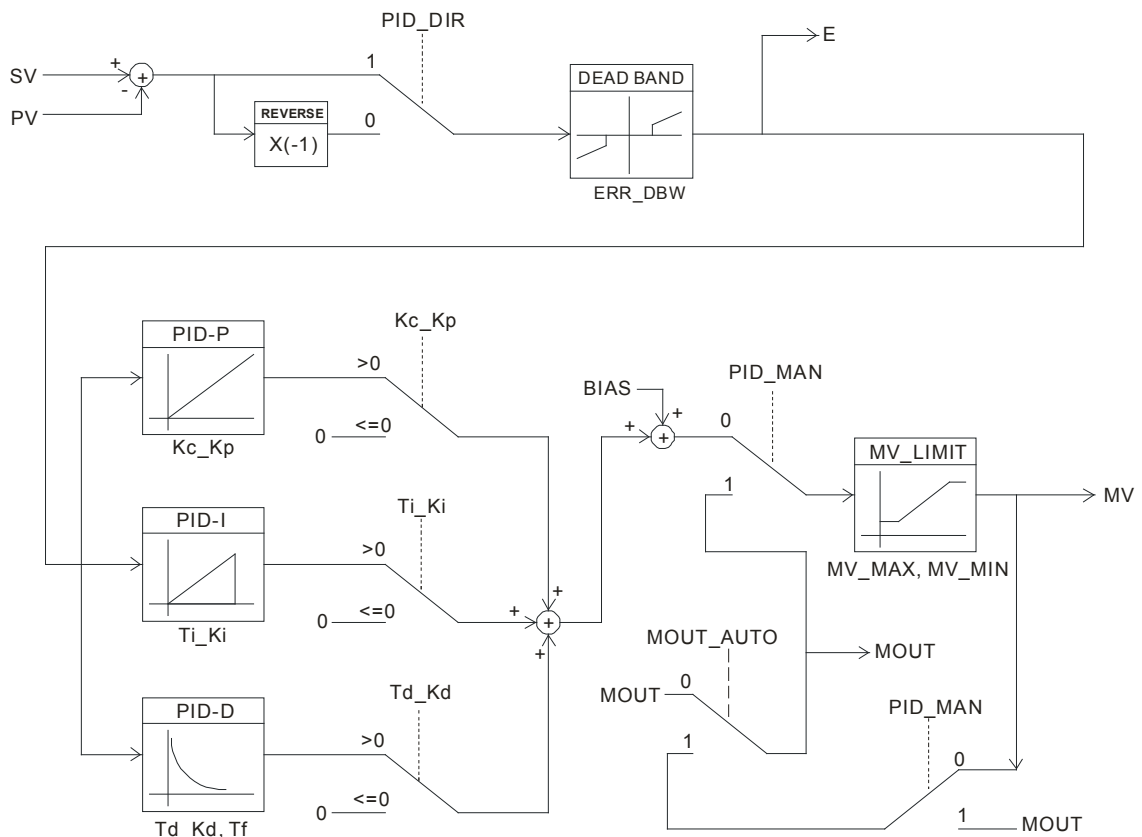
$$E = PV - SV$$

※上述所有公式中的 CV 值为 DPIDE 的 MV 值

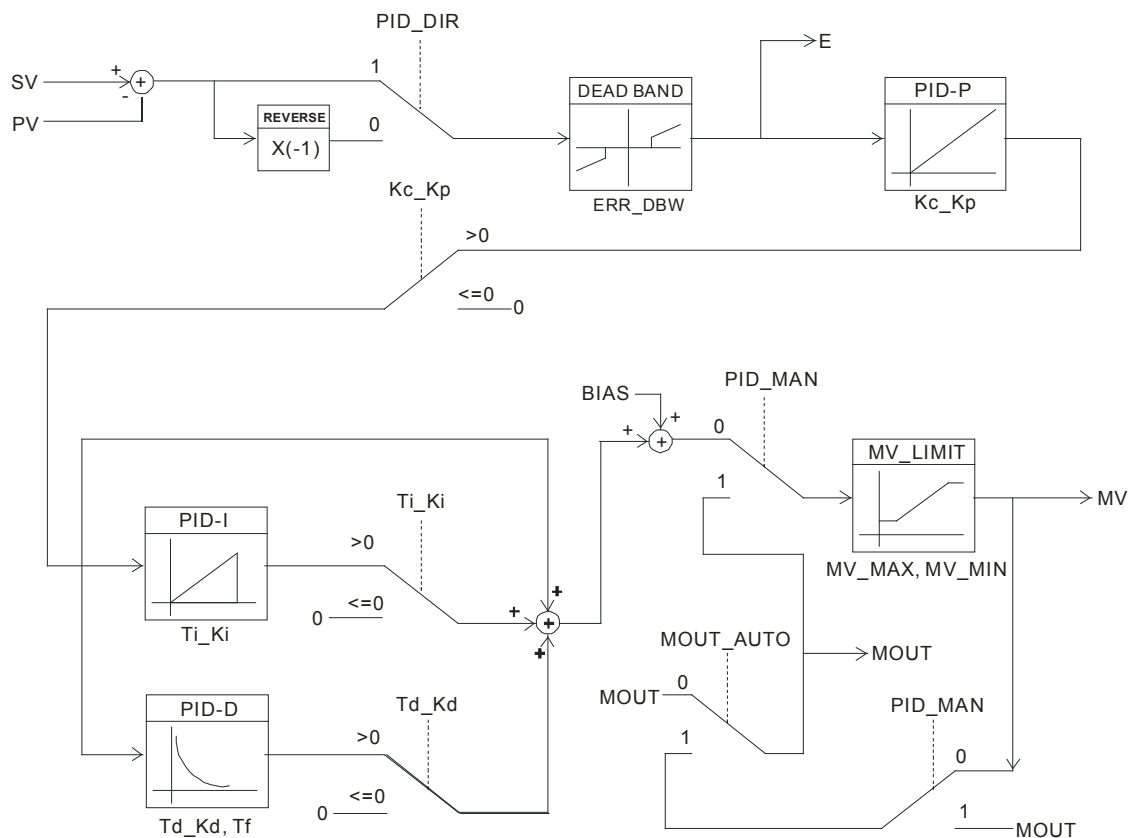
2. 当 PID_MODE 控制模式选择为 1，为自动调整模式，当自动调整完成后，PID_MODE 会自动变成 0 转换为自动控制模式。

PID 控制方块图：

PID Block Diagram (Independent)



PID Block Diagram (Dependent)



3

注意事项和建议：

1. 由于可使用 DPIDE 指令的控制环境很多，因此请适当的选取控制功能，例如：当选择自动调整参数（**PID_MODE=1**）功能时，MV 值会在 MAX/MIN 两个值之间做切换，所以请尽量勿使用于快速反应的马达控制环境中，以免造成在自动调适时系统因快速的剧烈变化，而造成对人员危险或损坏系统的现象发生。
2. 用户于调整 Kc_Kp、Ti_Ki 及 Td_Kd 三个主要参数时（**PID_MODE=0**），请先调整 Kc_Kp 值（依经验值设定），而 Ti_Ki 及 Td_Kd 值先设定为 0，等到调整到大致上可控制时，再依序调整 Ti_Ki 值（由小到大）以及 Td_Kd 值（由小到大）。其中 Kc_Kp 值为 1 则表示 100%，即对偏差值的增益为 1，小于 100% 将对偏差值衰减，大于 100% 将对偏差值放大。
3. 当用户选用（**PID_MODE=1**）时，建议请使用在停电保持区之 D 寄存器来储存参数，以免自动调整过的参数因停电后而消失。经过自动调整过的参数，并不能保证一定适用于每个控制的环境，因此用户当然可自行修改调整过的参数，不过建议最好只修改 Ti_Ki 或 Td_Kd 数值就好。
4. 本指令动作须配合许多参数值控制，因此请勿随意设定参数值，以免造成无法控制之现象。

3.11 逻辑操作指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>WAND</u>	<u>DAND</u>	✓	逻辑与 (AND) 运算	7
FC	<u>MAND</u>	—	✓	矩阵与 (AND) 运算	9
FC	<u>WOR</u>	<u>DOR</u>	✓	逻辑或 (OR) 运算	7
FC	<u>MOR</u>	—	✓	矩阵或 (OR) 运算	9
FC	<u>WXOR</u>	<u>DXOR</u>	✓	逻辑异或 (XOR) 运算	7
FC	<u>MXOR</u>	—	✓	矩阵异或 (XOR) 运算	9
FC	<u>WXNR</u>	<u>DXNR</u>	✓	逻辑同或 (XNR) 运算	7
FC	<u>MXNR</u>	—	✓	矩阵同或 (XNR) 运算	9
FC	<u>LD&</u>	<u>DLD&</u>	—	$S_1 \& S_2$	5
FC	<u>LD </u>	<u>DLD </u>	—	$S_1 S_2$	5
FC	<u>LD^</u>	<u>DLD^</u>	—	$S_1 \wedge S_2$	5
FC	<u>AND&</u>	<u>DAND&</u>	—	$S_1 \& S_2$	5
FC	<u>AND </u>	<u>DAND </u>	—	$S_1 S_2$	5
FC	<u>AND^</u>	<u>DAND^</u>	—	$S_1 \wedge S_2$	5
FC	<u>OR&</u>	<u>DOR&</u>	—	$S_1 \& S_2$	5
FC	<u>OR </u>	<u>DOR </u>	—	$S_1 S_2$	5
FC	<u>OR^</u>	<u>DOR^</u>	—	$S_1 \wedge S_2$	5

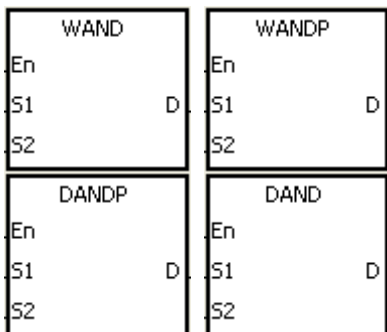
FB/FC	指令码			操作数	功能
FC	W D*	AND	P	S ₁ , S ₂ , D	逻辑与 (AND) 运算

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●	●*				●	●*						
S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●	●	●	●		●	○	●	○	○		
S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



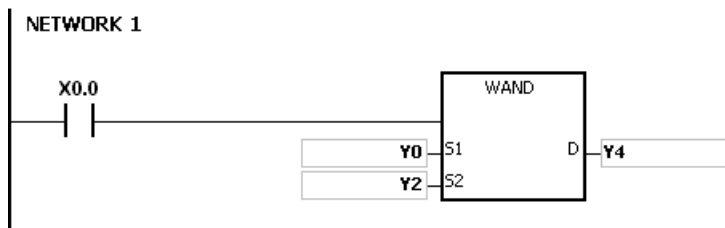
S₁ : 数据源装置 1
 S₂ : 数据源装置 2
 D : 运算结果

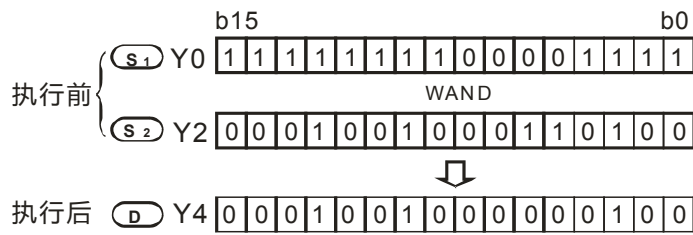
指令说明：

1. 两个数据源：S₁及 S₂作逻辑的“与” (AND) 运算并将结果存于 D。
2. DAND 才可使用 HC 装置。
3. 逻辑的“与” (AND) 运算之规则为两位均为 1 结果始为 1，否则为 0。

程序范例一：

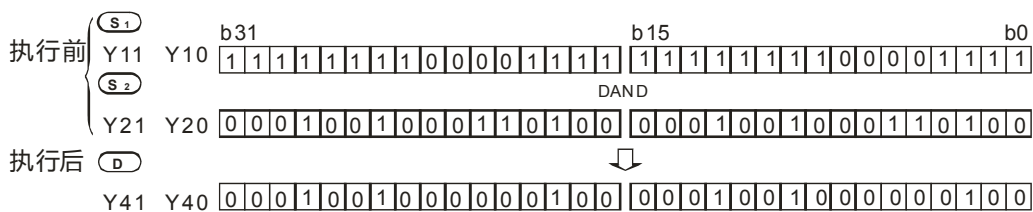
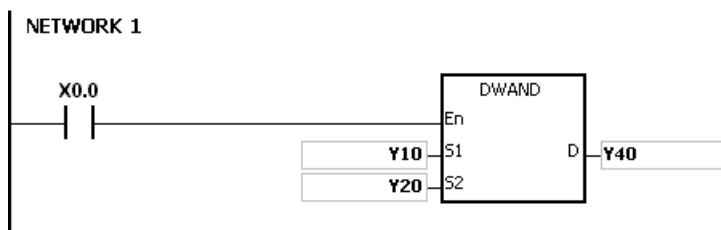
当 X0.0=ON 时，16 位 Y0 与 Y2 作 WAND，逻辑与 (AND) 运算，将结果存于 Y4 中。





程序范例二：

当 X0.0=ON 时，32 位 (Y11、Y10) 与 (Y21、Y20) 作 DAND，逻辑与 (AND) 运算，将结果存于 (Y41、Y40) 中。



FB/FC	指令码		操作数				功能									
FC		MAND	P	S_1, S_2, D, N				矩阵与 (AND) 运算								

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
D		●					●							
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●				●				
N	●	●			●	●	●	●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



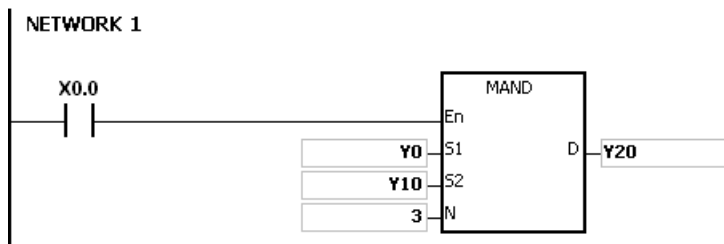
- S_1 : 矩阵来源装置 1
- S_2 : 矩阵来源装置 2
- D : 运算结果
- n : 数组长度

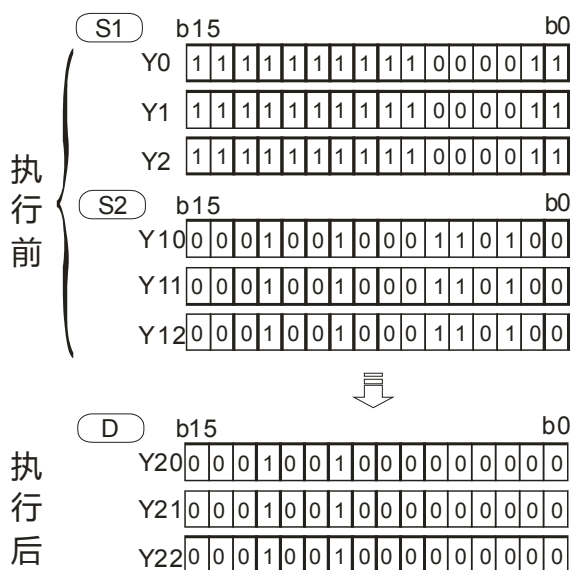
指令说明：

1. 两个矩阵来源依数组长度 n 将 S_1 及 S_2 作矩阵的“与” (AND) 运算并将结果存于 D。
2. 矩阵的“与” (AND) 运算之规则为两位均为 1 结果始为 1，否则为 0。
3. n 操作数的范围值为 1~256。

程序范例：

当 X0.0=ON 时，16 位寄存器 Y0~Y2 共 3 列与 16 位寄存器 Y10~Y12 共 3 列作 MAND，矩阵与 (AND) 运算，将结果存于 16 位寄存器 Y20~Y22 共 3 列中。

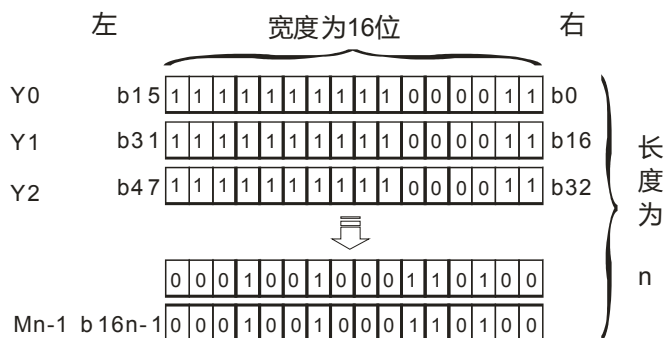




补充说明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n < 1$ 或 $n > 256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 矩阵指令说明：

- 矩阵是 1 个以上连续的 16 位寄存器所组成，组成矩阵之寄存器个数称为矩阵之长度 n ，一个矩阵共有 $16 \times n$ 个位（点），其运算单位一次只有一个位（点）。
- 矩阵指令是将 $16 \times n$ 个矩阵位（序号由 $b_0 \sim b_{16n-1}$ ）当作一连串单点之集合，而自此集合中指定某一单点作运作，而不将之当作数值看待。
- 矩阵指令主要在处理单点对多点（矩阵）或多点对多点之状态处理，如搬移、拷贝、比较、搜寻等，为极为方便和重要的应用指令。
- 在矩阵指令运作中，通常需要一个 16 位寄存器来指定矩阵中 $16n$ 个单点的某个单点当作运算对象，此寄存器称为矩阵的指针 Pr（Pointer），由用户于指令中指定，其有效范围为 $0 \sim 16n-1$ ，分别对应至矩阵中之位 $b_0 \sim b_{16n-1}$ 。
- 矩阵运作中有左、右位移或循环，我们定义高编号者为左，低编号者为右，如下图示。



- 矩阵宽度（C）固定为 16 位（bits）。

- Pr : 为矩阵之指针, 例如 Pr 值为 15 指到 b15 的位。

范例 : 以 Y0, n=3 构成的矩阵, Y0=16#AAAA, Y1=16#5555, Y2=16#AAFF

C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	Y0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	Y1
1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	Y2

范例 : 以 X0, n=3 构成的矩阵, X0=16#37, X1=16#68, X2=16#45

C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X0
0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X2

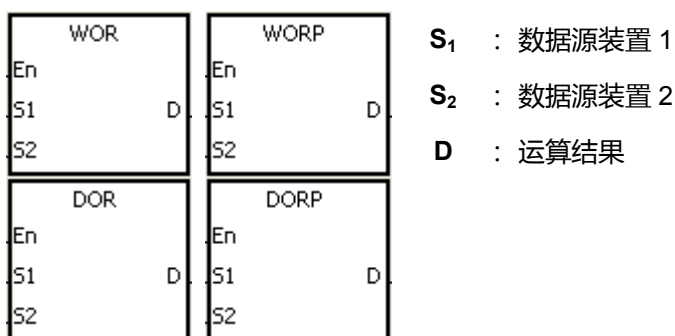
FB/FC	指令码			操作数	功能
FC	W D*	OR	P	S ₁ , S ₂ , D	逻辑或 (OR) 运算

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：

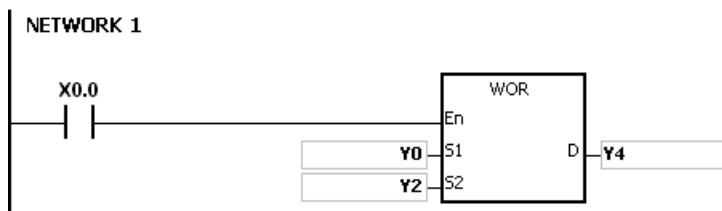


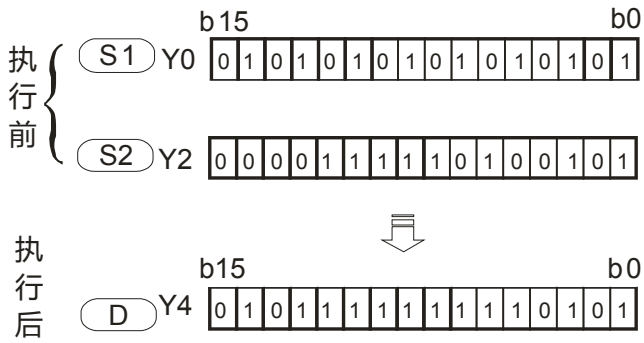
指令说明：

1. 两个数据源：**S₁**及**S₂**作逻辑的“或”（OR）运算结果存于**D**。
2. DOR 才可使用 HC 装置。
3. 逻辑的“或”（OR）运算的规则两位有任一为 1 则结果为 1，两者均为 0 结果才为 0。

程序范例一：

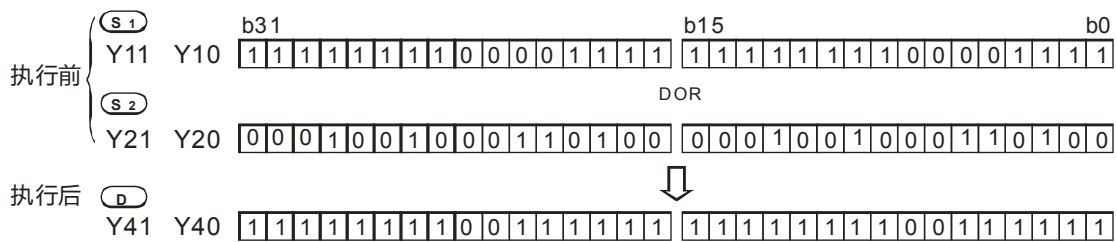
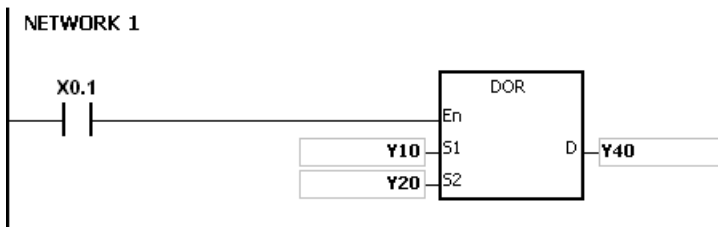
当 X0.0=ON 时，16 位 Y0 与 Y2 作 WOR，逻辑或（OR）运算，将结果存于 Y4 中。





程序范例二：

当 X0.1=ON 时，32 位 (Y11、 Y10) 与 (Y21、 Y20) 作 DOR，逻辑或 (OR) 运算，将结果存于 (Y41、 Y40) 中。



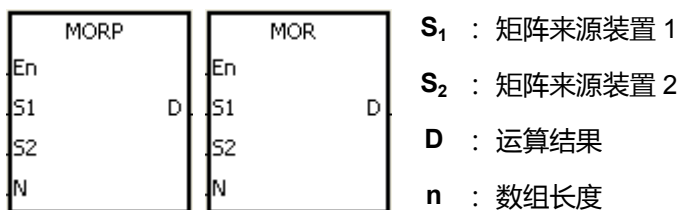
FB/FC	指令码		操作数				功能				
FC	MOR	P	S_1, S_2, D, N				矩阵或 (OR) 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
D		●					●							
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●				●				
N	●	●			●	●	●	●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

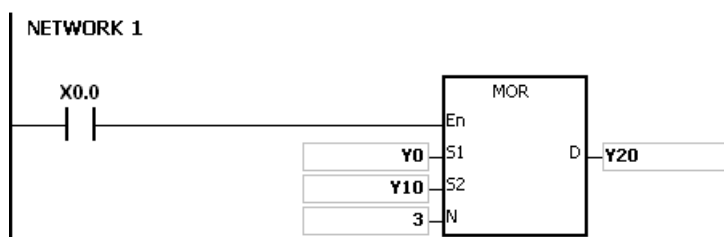


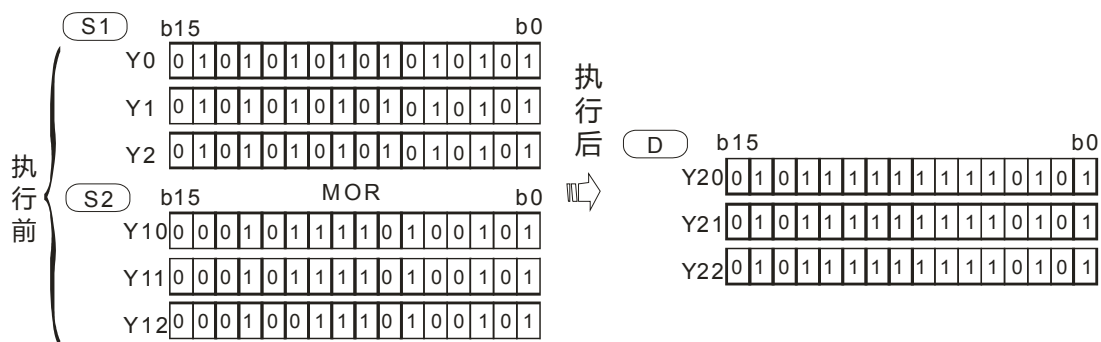
指令说明：

1. 两个矩阵来源依数组长度 n 将 S_1 及 S_2 作矩阵的“或” (OR) 运算并将结果存于 D。
2. 矩阵的“或” (OR) 运算的规则为两位有任一为 1 则结果为 1，两者均为 0 结果才为 0。
3. n 操作数的范围值为 1~256。

程序范例：

当 X0.0=ON 时，16 位寄存器 Y0~Y2 共 3 列与 16 位寄存器 Y10~Y12 共 3 列作 MOR，矩阵或 (OR) 运算，将结果存于 16 位寄存器 Y20~Y22 共 3 列中。





补充说明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n < 1$ 或 $n > 256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	W D*	XOR	P	S ₁ , S ₂ , D	逻辑异或 (XOR) 运算

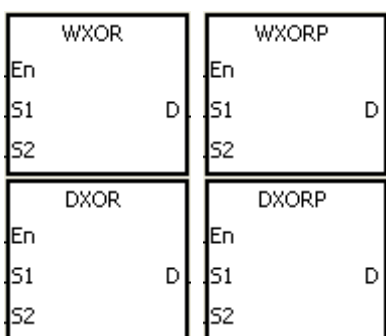
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



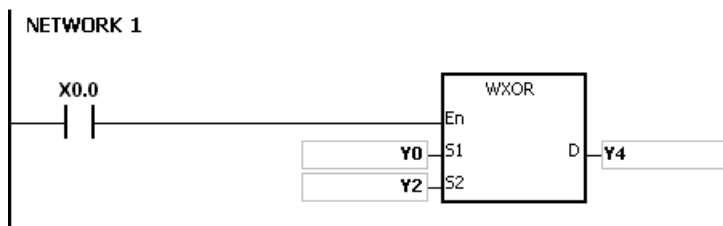
S₁ : 数据源装置 1
 S₂ : 数据源装置 2
 D : 运算结果

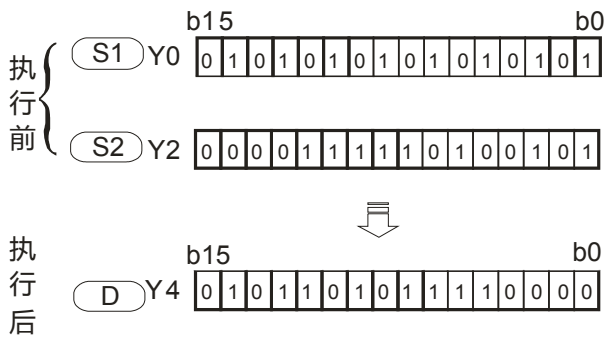
指令说明：

1. 两个数据源：S₁ 及 S₂ 作逻辑的“异或” (XOR) 运算结果存于 D。
2. DXOR 才可使用 HC 装置。
3. 逻辑的“异或” (XOR) 运算之规则为两者相同结果为 0，两者不同结果为 1。

程序范例一：

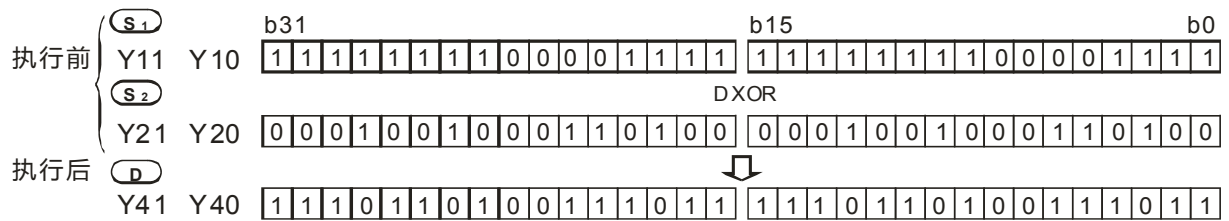
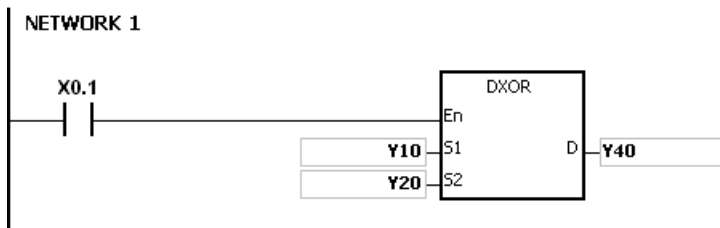
当 X0.0=ON 时，16 位 Y0 与 Y2 作 WXOR，逻辑异或 (XOR) 运算，将结果存于 Y4 中。





程序范例二：

当 X0.1=ON 时，32 位 (Y11、Y10) 与 (Y21、Y20) 作 DXOR，逻辑异或 (XOR) 运算，将结果存于 (Y41、Y40) 中。



3

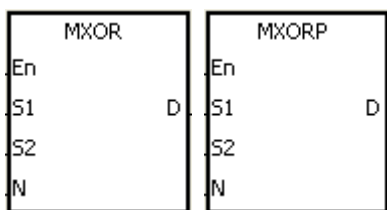
FB/FC	指令码		操作数				功能					
FC		MXOR	P	S_1, S_2, D, N				矩阵异或 (XOR) 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
D		●					●							
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●				●				
N	●	●			●	●	●	●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



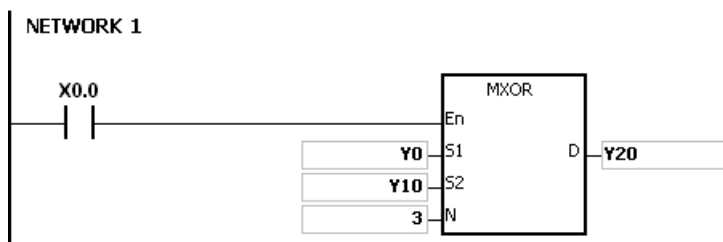
S_1 : 矩阵来源装置 1
 S_2 : 矩阵来源装置 2
D : 运算结果
n : 数组长度

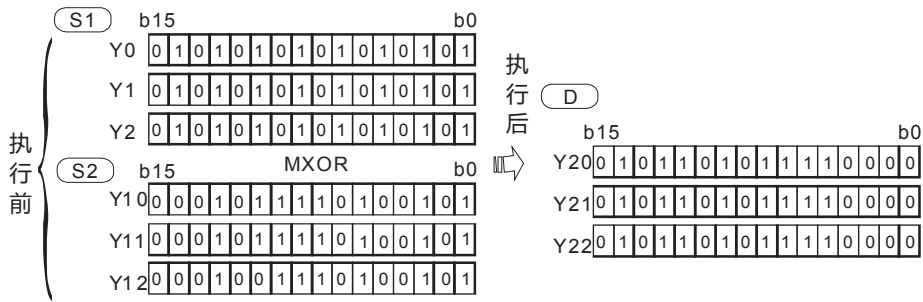
指令说明：

1. 两个矩阵来源依数组长度 n 将 S_1 及 S_2 作矩阵的“异或” (XOR) 运算并将结果存于 D。
2. 矩阵的“异或” (XOR) 运算的规则为两者相同结果为 0，两者不同结果为 1。
3. n 操作数的范围值为 1~256。

程序范例：

当 X0.0=ON 时，16 位寄存器 Y0~Y2 共 3 列与 16 位寄存器 Y10~Y12 共 3 列作 MXOR，矩阵异或 (XOR) 运算，将结果存于 16 位寄存器 Y20~Y22 共 3 列中。





补充说明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n < 1$ 或 $n > 256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	W D*	XNR	P	S ₁ , S ₂ , D	同或 (XNR) 运算

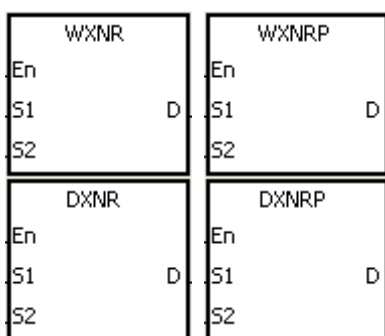
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



S₁ : 数据源装置 1

S₂ : 数据源装置 2

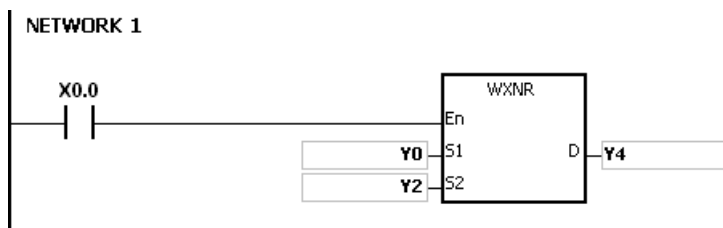
D : 运算结果

指令说明：

1. 两个数据源：S₁及 S₂作逻辑的“同或”（XNR）运算结果存于 D。
2. DXNR 才可使用 HC 装置。
3. 逻辑的“同或”（XNR）运算的规则为两者相同结果为 1，两者不同结果为 0。

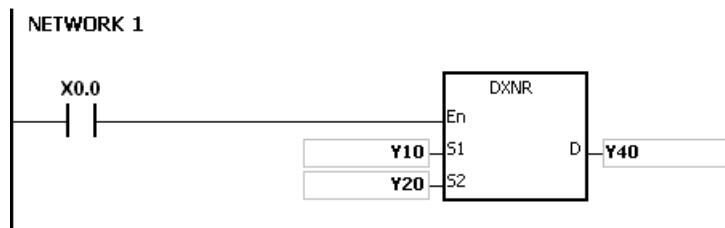
程序范例一：

当 X0.0=ON 时，16 位 Y0 与 Y2 作 WXNR，逻辑同或（XNR）运算，将结果存于 Y4 中。



程序范例二：

当 X0.0=ON 时, 32 位(Y11、 Y10)与(Y21、 Y20)作 DXOR ,逻辑同或(XNR)运算, 将结果存于(Y41、 Y40) 中。



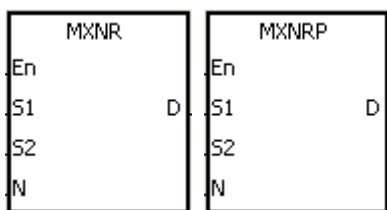
FB/FC	指令码		操作数				功能					
FC		MXNR	P	S₁, S₂, D, N				矩阵同或 (XNR) 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●					●							
D		●					●							
N		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●				●				
N	●	●			●	●	●	●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



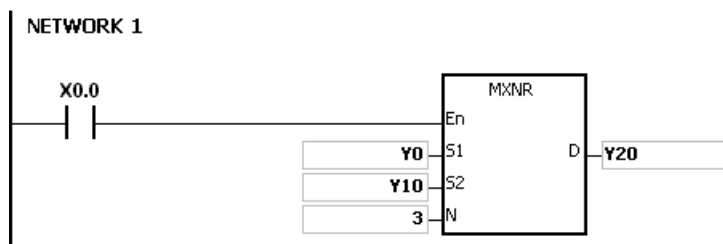
S₁ : 矩阵来源装置 1
 S₂ : 矩阵来源装置 2
 D : 运算结果
 n : 数组长度

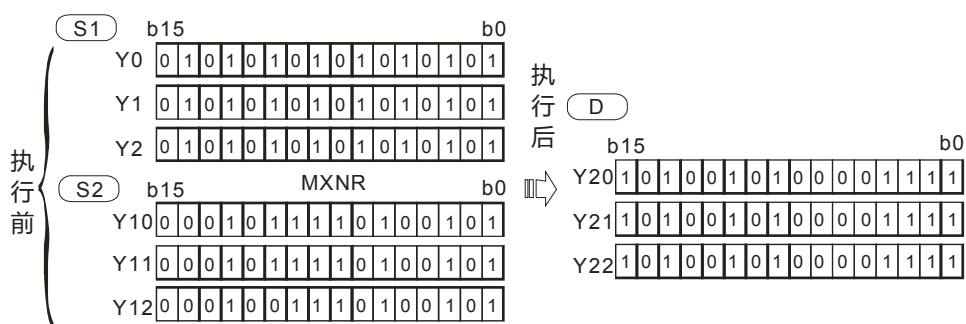
指令说明：

1. 两个矩阵来源依数组长度 n 将 S₁ 及 S₂ 作矩阵的同或 (XNR) 运算并将结果存于 D。
2. 矩阵的同或 (XNR) 运算的规则为两者相同结果为 1，两者不同结果为 0。
3. n 操作数的范围值为 1~256。

程序范例：

当 X0.0=ON 时，16 位寄存器 Y0~Y2 共 3 列与 16 位寄存器 Y10~Y12 共 3 列作 MXNR，矩阵 XNR 运算，将结果存于 16 位寄存器 Y20~Y22 共 3 列中。



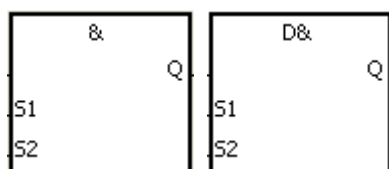


补充说明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数							功能										
FC	D*	LD #		S ₁ , S ₂							接点类型逻辑运算 LD #										
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING							
S ₁ , S ₂		●	●*				●	●*													
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF				
S ₁ , S ₂	●	●				●	●	●	●		●	○		○	○						
脉冲执行型							16 位指令					32 位指令									
-							AH 运动控制主机					AH 运动控制主机									

图形：



S₁ : 数据源装置 1

S₂ : 数据源装置 2

以 LD&跟 DLD&为例

指令说明：

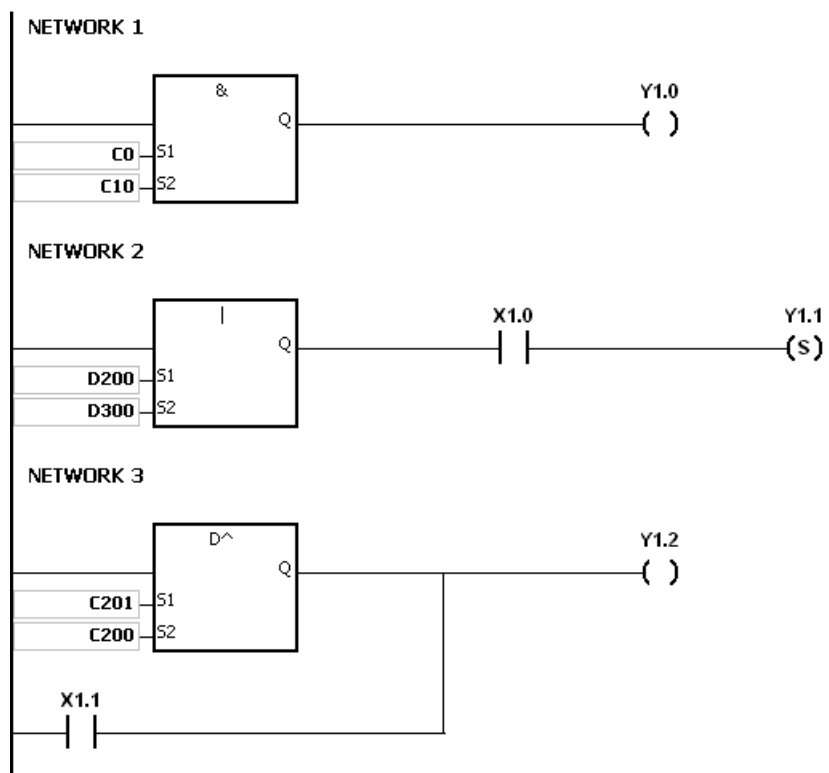
- S₁ 与 S₂ 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- DLD # 才可使用 HC 装置。
- LD # 的指令可直接与母线连接使用。

16-bit 指令	32-bit 指令	导通条件	非导通条件
LD&	DLD&	S ₁ &S ₂ ≠ 0	S ₁ &S ₂ = 0
LD	DLD	S ₁ S ₂ ≠ 0	S ₁ S ₂ = 0
LD^	DLD^	S ₁ ^S ₂ ≠ 0	S ₁ ^S ₂ = 0

- & : 逻辑的“与” (AND) 运算。
- | : 逻辑的“或” (OR) 运算。
- ^ : 逻辑的“异或” (XOR) 运算。

程序范例：

- C0 与 C10 的内容做逻辑的“与” (AND) 运算不等于 0 时，Y1.0=ON。
- D200 与 D300 的内容做逻辑的“或” (OR) 运算不等于 0 时，而且 X1.0=ON 的时候，Y1.1=ON 并保持住。
- C201 与 C200 的内容做逻辑的“异或” (XOR) 运算不等于 0 时或是 X1.1=ON 的时候，Y1.2=ON。

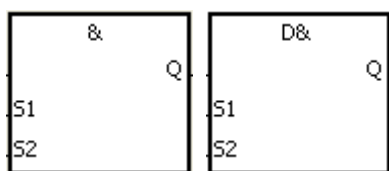


补充说明：

S₁、S₂ 装置不合法，接点不导通，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码				操作数				功能									
FC	D*	AND #			S ₁ , S ₂				接点类型逻辑运算 AND #									
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S ₁ , S ₂		●	●*				●	●*										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S ₁ , S ₂	●	●				●	●	●	●		●	○		○	○			
				脉冲执行型				16 位指令				32 位指令						
				-				AH 运动控制主机				AH 运动控制主机						

图形：



以 AND&跟 DAND&为例

S₁ : 数据源装置 1

S₂ : 数据源装置 2

指令说明：

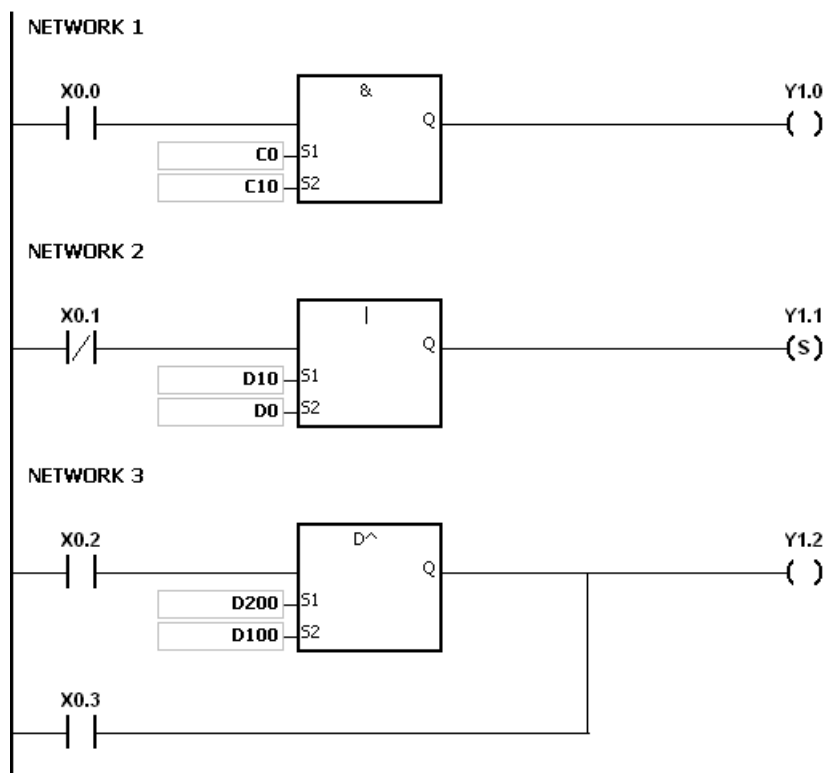
1. S₁与 S₂之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
2. DAND # 才可使用 HC 装置。
3. AND # 的指令是与接点串接的运算指令。

16-bit 指令	32-bit 指令	导通条件	非导通条件
AND&	DAND&	S ₁ &S ₂ ≠ 0	S ₁ &S ₂ = 0
AND	DAND	S ₁ S ₂ ≠ 0	S ₁ S ₂ = 0
AND^	DAND^	S ₁ ^S ₂ ≠ 0	S ₁ ^S ₂ = 0

4. & : 逻辑的“与” (AND) 运算。
5. | : 逻辑的“或” (OR) 运算。
6. ^ : 逻辑的“异或” (XOR) 运算。

程序范例：

1. 当 X0.0=ON 时且 C0 与 C10 的内容做逻辑的“与” (AND) 运算不等于 0 时，Y1.0=ON。
2. 当 X0.1=OFF 时且 D10 与 D0 的内容做逻辑的“或” (OR) 运算不等于 0 时，Y1.1=ON 并保持住。
3. 当 X0.2=ON 时且 32 位寄存器 (D200, D201) 与 32 位寄存器 (D100, D101) 的内容做逻辑的“异或” (XOR) 运算不等于 0 时或是 X0.3=ON 的时候，Y1.2=ON。

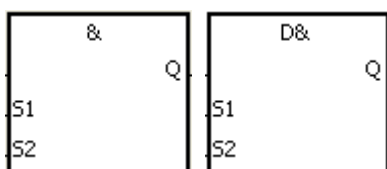


补充说明：

S₁、S₂ 装置不合法，接点不导通，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数								功能								
FC	D*	OR #		S ₁ , S ₂								接点类型逻辑运算 OR #								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING						
S ₁ , S ₂		●	●*				●	●*												
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF			
S ₁ , S ₂	●	●				●	●	●	●		●	○		○	○					
				脉冲执行型				16 位指令				32 位指令								
				-				AH 运动控制主机				AH 运动控制主机								

图形：



S₁ : 数据源装置 1

S₂ : 数据源装置 2

以 OR&跟 DOR&为例

指令说明：

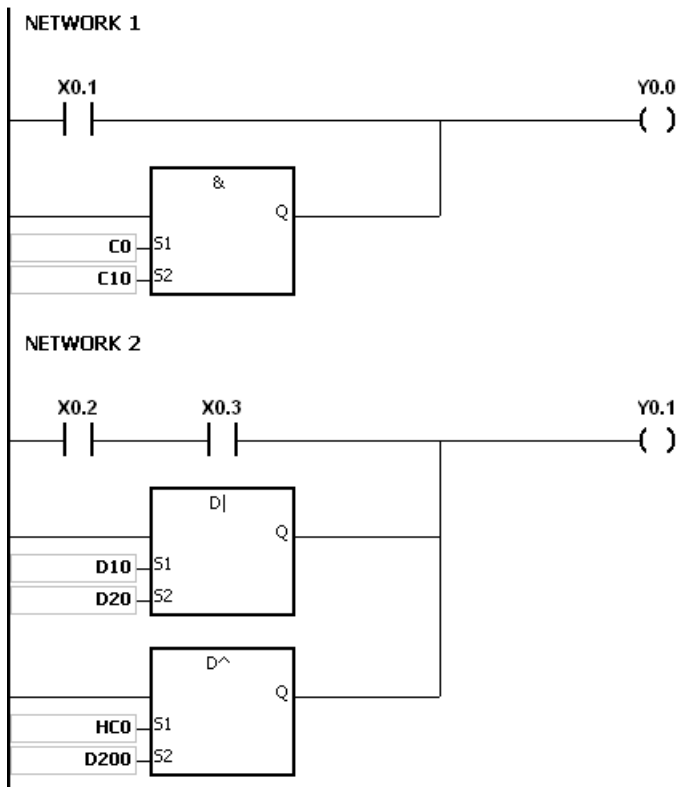
- S₁ 与 S₂ 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- DOR # 才可使用 HC 装置。
- OR # 的指令是与接点并接的运算指令。

16-bit 指令	32-bit 指令	导通条件	非导通条件
OR&	DOR&	S ₁ &S ₂ ≠ 0	S ₁ &S ₂ = 0
OR	DOR	S ₁ S ₂ ≠ 0	S ₁ S ₂ = 0
OR^	DOR^	S ₁ ^S ₂ ≠ 0	S ₁ ^S ₂ = 0

- & : 逻辑的“与” (AND) 运算。
- | : 逻辑的“或” (OR) 运算。
- ^ : 逻辑的“异或” (XOR) 运算。

程序范例：

- 当 X0.1=ON 时或 C0 与 C10 的内容做逻辑的“与” (AND) 运算不等于 0 时，Y0.0=ON。
- 当 X0.2 及 X0.3 都等于 ON 的时候，或者是 32-bit 寄存器 (D10, D11) 与 32 位寄存器 (D20, D21) 的内容做逻辑的“或” (OR) 运算不等于 0 时，或者是 32 位计数器 HC0 与 32 位寄存器 (D200, D201) 的内容做逻辑的“异或” (XOR) 运算不等于 0 时，Y0.1=ON。



补充说明：

S₁、S₂ 装置不合法，接点不导通，SM0=ON，错误码 SR0=16#2003。

3.12 循环指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>ROR</u>	<u>DROR</u>	✓	右循环	5
FC	<u>RCR</u>	<u>DRCR</u>	✓	附进位标志右循环	5
FC	<u>ROL</u>	<u>DROL</u>	✓	左循环	5
FC	<u>RCL</u>	<u>DRCL</u>	✓	附进位标志左循环	5
FC	<u>MBR</u>	—	✓	矩阵位循环	7

FB/FC	指令码			操作数	功能
FC	D*	ROR	P	D, n	右循环

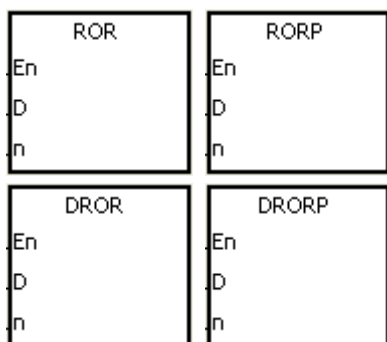
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●	●	●	●		●	○	●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



D : 欲循环的装置
n : 一次循环的位数

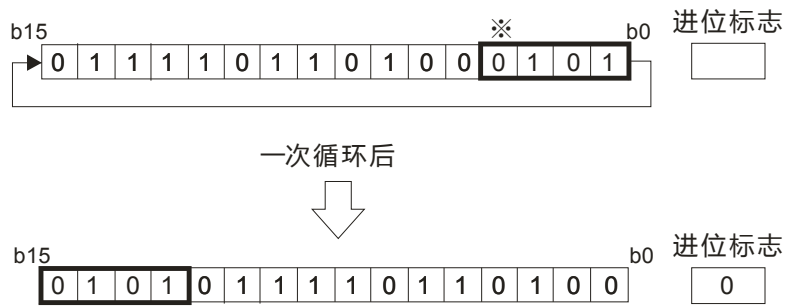
指令说明：

1. 将 D 所指定的装置内容一次向右循环 n 个位。
2. DROR 才可以使用 HC 装置。
3. 16 位指令 n=1~16, 32 位指令 n=1~32。
4. 本指令一般都是使用脉冲执行型指令 (RORP、DRORP)。

程序范例：

当 X0.0 从 OFF→ON 变化时, D10 的 16 个位以 4 个位为一组往右循环, 如下图所示标明※的位内容被传送至进位标志信号 SM602 内。



**补充说明：**

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	D*	RCR	P	D, n	附进位标志右循环

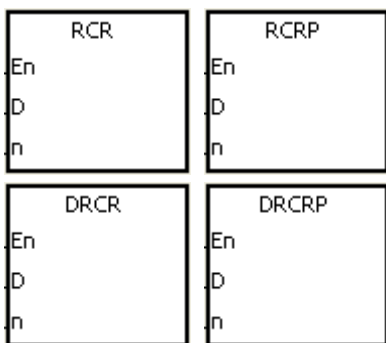
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●	●	●	●		●	○	●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



D : 欲循环的装置
n : 一次循环的位数

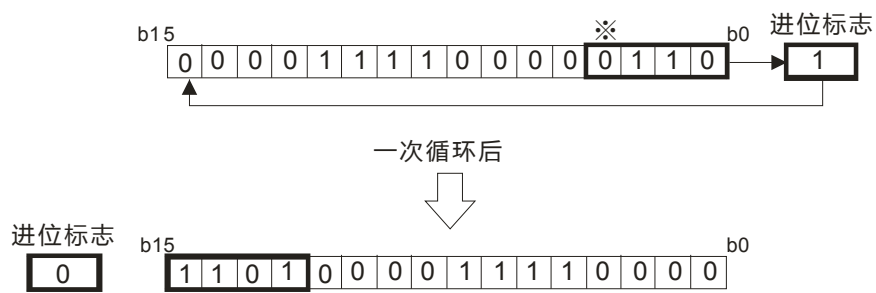
指令说明：

1. 将 D 所指定的装置内容连同进位标志 SM602，一次向右循环 n 个位。
2. 32 位指令才可以使用 HC 装置。
3. 16 位指令 n=1~16，32 位指令 n=1~32。
4. 本指令一般都是使用脉冲执行型指令（RCRP、DRCRP）。

程序范例：

当 X0.0 从 OFF→ON 变化时，D10 的 16 个位连同进位标志 SM602 共 17 个位以 4 个位为一组往右循环，如下图所示标明※的位内容被传送至进位标志信号 SM602 内。



**补充说明：**

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	D*	ROL	P	D, n	左循环

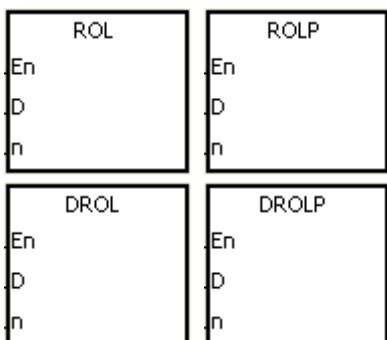
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●	●	●	●		●	○	●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



D : 欲循环的装置
n : 一次循环的位数

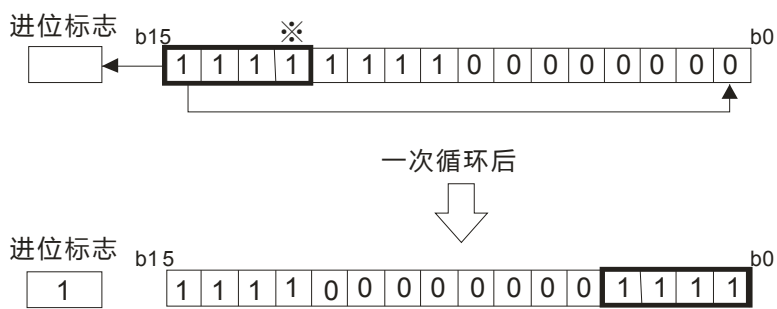
指令说明：

1. 将 D 所指定的装置内容一次向左循环 n 个位。
2. 32 位指令才可以使用 HC 装置。
3. 16 位指令 n=1~16，32 位指令 n=1~32。
4. 本指令一般都是使用脉冲执行型指令（ROLP、DROLP）。

程序范例：

当 X0.0 从 OFF→ON 变化时，D10 的 16 个位以 4 个位一组往左循环，如下图所示标明※的位内容被传送至进位标志信号 SM602 内。



**补充说明：**

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	D*	RCL	P	D, n	附进位标志左循环

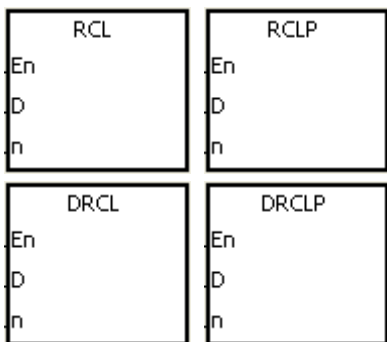
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●	●	●	●		●	○	●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	32 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：



D : 欲循环的装置
n : 一次循环的位数

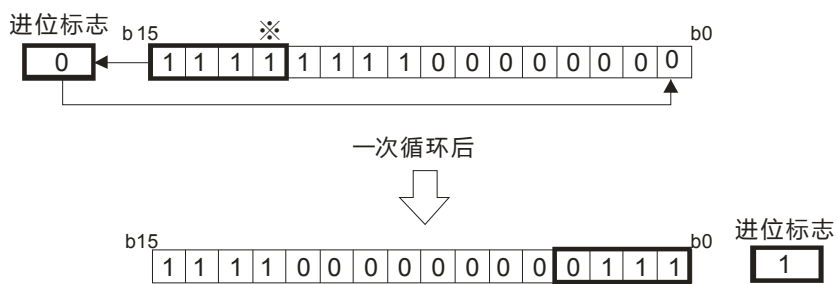
指令说明：

1. 将 D 所指定的装置内容连同进位标志 SM602，一次向左循环 n 个位。
2. 32 位指令才可以使用 HC 装置。
3. 16 位指令 n=1~16，32 位指令 n=1~32。
4. 本指令一般都是使用脉冲执行型指令（RCLP、DRCLP）。

程序范例：

当 X0.0 从 OFF→ON 变化时，D10 的 16 个位连同进位标志 SM602 共 17 个位以 4 个位一组往左循环，如下图所示标明※的位内容被传送至进位标志信号 SM602 内。





补充说明：

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

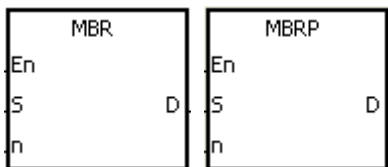
FB/FC	指令码			操作数	功能
FC		MBR	P	S, D, n	矩阵位循环

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
N		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
N	●	●			●	●	●	●	●		●		●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



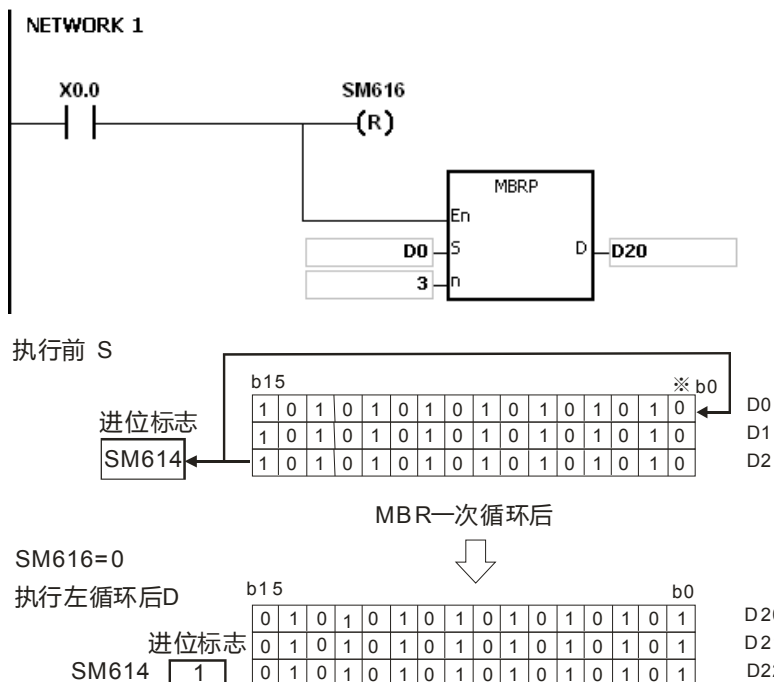
S : 矩阵来源装置
D : 运算结果
n : 数组长度

指令说明：

1. 矩阵来源依数组长度 **n** 将 **S** 矩阵位做左右循环控制，SM616=0 决定矩阵位左循环，SM616=1 决定矩阵位右循环。每次循环一位，因循环造成之空位（左旋时为 b0，右旋时为 b16n-1）由旋出位（左旋时为 b16n-1 右旋时为 b0）状态填补。将结果存入 **D**。旋出位不但用以填补前述之空位，同时并将状态送到进位标志 SM614。
2. **n** 操作数值的范围为 1~256。
3. 本指令一般都是使用脉冲执行型指令（MBRP）。

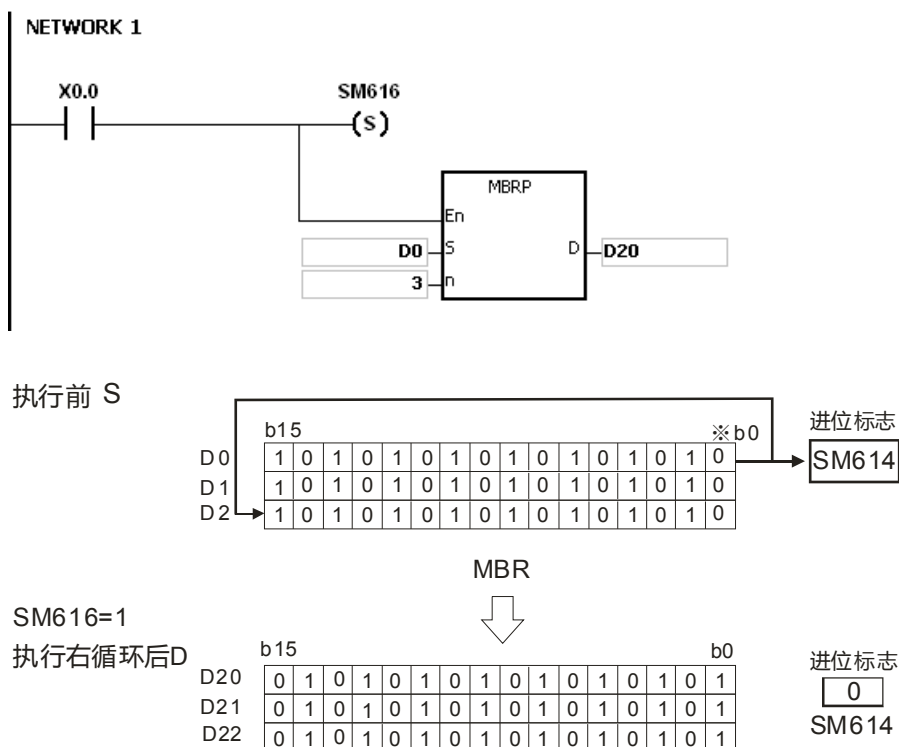
程序范例一：

当 X0.0=ON 时 SM616=OFF 决定矩阵为左循环，16 位寄存器 D0~D2 矩阵循环，将结果存于 16 位寄存器 D20~D22 矩阵中，下图标明※位的内容将传送至进位标志 SM614。



程序范例二：

当 X0.1=ON 时，SM616=ON 作矩阵右循环，16 位寄存器 D0~D2 矩阵作右循环，将结果存于 16 位寄存器 D20~D22 矩阵中，下图标明※位的内容将传送至进位标志 SM614。



补充说明：

1. S+n-1、D+n-1 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n<1 或 n>256 时，指令不执行，SM0=ON，SR0 错误码=16#200B。

3. 标志信号说明：

SM614： 矩阵循环位移输出进位标志

SM616： 矩阵循环位移方向标志

3.13 基本指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>RST</u>	—	—	接点或寄存器清除	3
FC	<u>TMR</u>	—	—	16 位定时器	5
FC	<u>TMRH</u>	—	—	16 位定时器	5
FC	<u>CNT</u>	—	—	16 位计数器	5
FC	—	<u>DCNT</u>	—	32 位计数器	5

FB/FC	指令码			操作数			功能							
FC		RST		D			接点或寄存器清除							

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●	●	●	●	●	●	●	●	●	●	○	●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：

3

元件 **D** : 清除组件编号
 —(R)

指令说明：

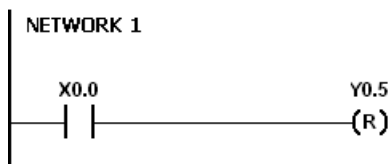
1. 当 RST 指令被驱动，其指定的组件的动作如下：

组件	状态
Bit	线圈及接点都会被设定为 OFF。
T, C, HC/AC	目前计时或计数值会被设为 0，且线圈及接点都会被设定为 OFF。
Word	内容值会被设为 0。

2. 若 RST 指令没有被执行，其指定组件的状态保持不变。
3. 本指令有支持 DY 装置。

程序范例：

当 X0.0=ON 时，Y0.5 会被设为 OFF。



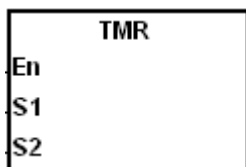
FB/FC	指令码			操作数				功能					
FC		TMR		S_1, S_2				16 位定时器					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●					●							
S_2		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1					●												
S_2	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



S_1 : 定时器编号
 S_2 : 定时器设定值

指令说明：

详细内容说明请参考 TMRH 指令的指令说明。

FB/FC	指令码		操作数				功能				
FC		TMRH	S₁, S₂				16 位定时器				

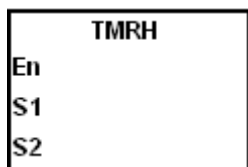
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁					○												
S ₂	○	○						○	○		○		○	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

3

图形：



S₁ : 定时器编号
S₂ : 定时器设定值

指令说明：

1. TMR 指令所指定的 T 定时器以 100ms 为单位计时，TMRH 指令所指定的 T 定时器以 1ms 为单位计时。
2. 副程序专用 T 定时器为 T1920~T2047。
3. TMR 跟 TMRH 的定时器内容值范围为 0~32767。
4. 在程序中同一个 T 定时器如果重复使用（包含使用在不同指令 TMR、TMRH 中），则以最快到达的为主。
5. 使用 TMR 指令时，只要在 T 之前加上一个 S，就会变成累积定时器 ST 装置，表示目前的 T 变成累积定时器，则条件接点 OFF 时 T 的值不会被清除，条件接点=ON 的时候，T 由目前的值开始累积计时。
6. 在程序中同一个 T 定时器如果重复使用，其中一个条件接点 OFF 时，则 T 会 OFF。
7. 在程序中同一个 T 定时器如果重复使用为 T 与 ST，其中一个条件接点 OFF 时则 T 会 OFF。
8. 当 T 定时器 ON->OFF 且条件式为 ON 时，T 计时值归零并重新计数。
9. 当 TMR 指令执行时，其所指定的定时器线圈受电，定时器开始计时，当到达所指定的定时值（计时值>=设定值），其接点动作如下：

NO (Normally Open) 接点	开路
NC (Normally Closed) 接点	闭合

程序范例一：

当 X0.0 为 ON 时，定时器 T0 会加载设定值 50，当计时到达时 T0=50，T0 接点会 ON。

**程序范例二：**

当 X0.0 为 ON 时，定时器 T0 会加载设定值 50，当定时器数到 T0=25 时，将 X0.0 OFF=>ON，定时器会从 T0=25 继续数到 T0=50，且 T0 接点会 ON。

**程序范例三：**

当 X0.0=ON 时，定时器 T5 会加载设定值 1000，当计时到达时 T5=1000，T5=ON。

**程序范例四：**

当 X0.0 为 ON 时，定时器 T5 会加载设定值 1000，当定时器数到 T5=500 时，将 X0.0 OFF=>ON，定时器会从 T5=500 继续数到 T5=1000，且 T5 接点会 ON。

**补充说明：**

S₁操作数，若使用 ISPSOft 宣告，则数据类型为 TIMER。

FB/FC	指令码		操作数				功能				
FC		CNT	S₁, S₂				16 位计数器				

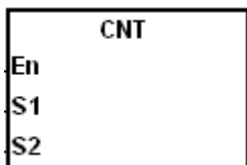
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁						○											
S ₂	○	○						○	○		○	○	○	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

3

图形：



S₁ : 计数器编号
S₂ : 计数器设定值

指令说明：

- 当 CNT 指令由 OFF->ON 执行，表示所指定的计数器线圈由失电→受电，则该计数器计数值加 1，当计数到达所指定的数值（计数值=设定值），其接点动作如下：

NO (Normally Open) 接点	开路
NC (Normally Closed) 接点	闭合

- 当计数到达之后，若再有计数脉冲输入，其接点及计数值均保持不变，若要重新计数或作清除的动作，请利用 RST 指令。

程序范例：

当 SM408 的状态第一次 ON 时，计数器 C0 会加载设定值 10 并开始计数，当 SM408 的状态经过 10 次 OFF=>ON 的变化后，计数器 C0 的值计数到达 C0=10，且 C0 的接点为 ON。

当 C0 为 ON 之后，虽然 SM408 继续 OFF=>ON，但 C0 的值已经计数到达就不会再累加。



补充说明：

S₁ 操作数，若使用 ISPSOft 宣告，则数据类型为 COUNTER。

FB/FC	指令码			操作数				功能						
FC	DCNT			S ₁ , S ₂				32 位计数器						

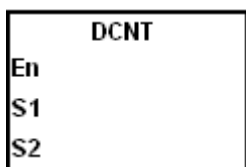
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁			●					●						
S ₂			●					●						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁							○										
S ₂	○	○						○	○		○		○	○	○		

脉冲执行型	16-bit 指令码	32 位指令
-	-	AH 运动控制主机

3

图形：



S₁ : 计数器编号
S₂ : 计数器设定值

指令说明：

- DCNT 为 32 位计数器 HC0 至 HC63 的启动指令。
- 一般用加减算计数器 HC0~HC63，当 DCNT 指令由 OFF→ON 时，计数器之当前值将执行上数（加一）的动作或下数（减一）的动作，依 SM621~SM684 的设定模式。
- 当 DCNT 指令 OFF 时，该计数器停止计数，但原有计数值不会被清除，可使用指令 RST 清除计数值及其接点。

程序范例：

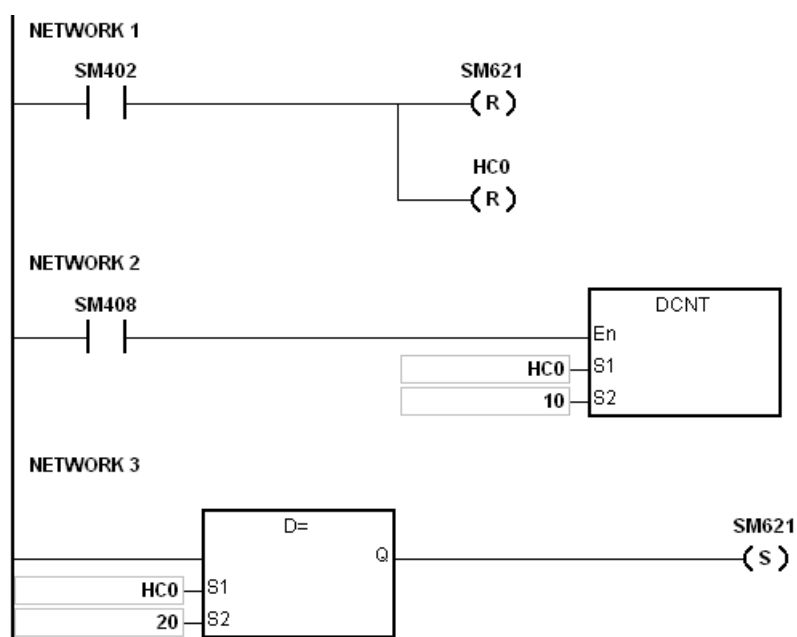
当 PLC RUN 时，定时器 HC0 被清除，且 SM621=OFF 将执行上数，此时 SM408 的状态为第一次 ON，所以计数器 HC0 会加载设定值 10 并开始计数。

当 SM408 的状态经过 10 次 OFF=>ON 的变化后，计数器 HC0 的值计数到达 HC0=10，且 HC0 的接点为 ON。

当 HC0 为 ON 之后，因为 SM408 继续 OFF=>ON，所以虽然 HC0 的值已经计数到达，但还是会继续累加。

当计数器继续上数到达 HC0=20 时，程序会设定 SM621=ON 将执行下数，当 SM408 的状态再经过 10 次 OFF=>ON 的变化后，计数器 HC0 的值计数到达由 HC0=10 递减到 HC0=9 时，HC0 的接点会被 OFF。

当 HC0 的接点被 OFF 后，因为 SM408 继续 OFF=>ON，所以 HC0 的值还是会继续递减。

**补充说明：**

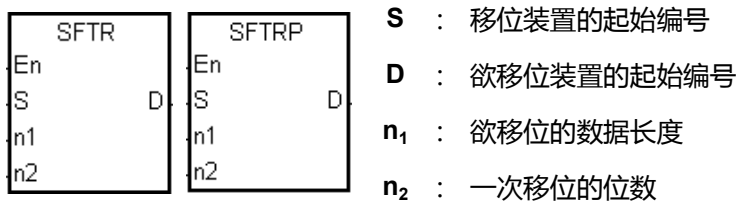
1. SM621~SM684 的设定模式，详细使用说明，请参考第 2 章的 32 位计数器 HC 的使用说明。
2. S₁ 操作数，若使用 ISPSOft 宣告，则数据类型为 COUNTER。

3.14 位移指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>SFTR</u>	—	✓	位右移	9
FC	<u>SFTL</u>	—	✓	位左移	9
FC	<u>WSFR</u>	—	✓	寄存器右移	9
FC	<u>WSFL</u>	—	✓	寄存器左移	9
FC	<u>SFWR</u>	—	✓	位移写入	7
FC	<u>SFRD</u>	—	✓	位移读出	7
FC	<u>SFPO</u>	—	✓	读出数据串行最新数据	5
FC	<u>SFDEL</u>	—	✓	删除数据串行中的数据	7
FC	<u>SFINS</u>	—	✓	插入数据到数据串行中	7
FC	<u>MBS</u>	—	✓	矩阵位位移	7
FC	<u>SFR</u>	—	✓	16 位寄存器位右移	5
FC	<u>SFL</u>	—	✓	16 位寄存器位左移	5
FC	<u>BSFR</u>	—	✓	n 个位右移 1 个位	5
FC	<u>BSFL</u>	—	✓	n 个位左移 1 个位	5
FC	<u>NSFR</u>	—	✓	n 个寄存器右移	5
FC	<u>NSFL</u>	—	✓	n 个寄存器左移	5

FB/FC	指令码			操作数							功能						
FC		SFTR	P	S, D, n₁, n₂							位右移						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S	●																
D	●																
n₁, n₂		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●	●	●				●	●	●			●				
D	●	●	●	●				●	●	●			●				
n₁, n₂	●	●						●	●		●		●	○	○		
							脉冲执行型				16 位指令				32 位指令		
							AH 运动控制主机				AH 运动控制主机				-		

图形：

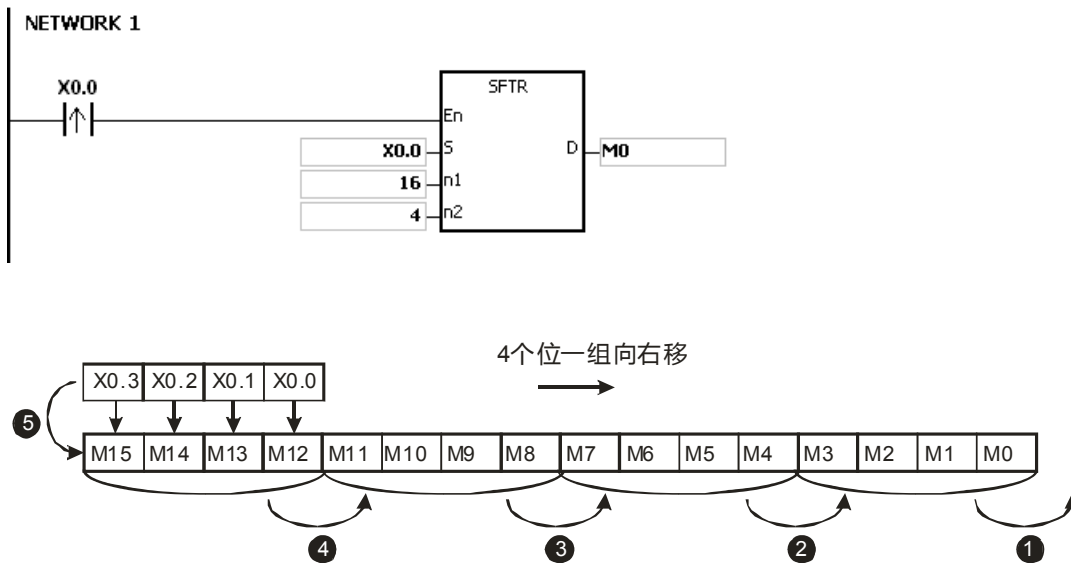


指令说明：

1. 将 **D** 开始的起始编号，具有 **n₁** 个数字元（位移寄存器长度）的位装置，以 **n₂** 位个数来右移。而 **S** 开始起始编号以 **n₂** 位个数移入 **D** 中来填补位空位。
2. 本指令一般都是使用脉冲执行型指令（SFTRP）。
3. **n₁** 值的范围为 1~1024，**n₂** 值的范围为 1~**n₁**。

程序范例：

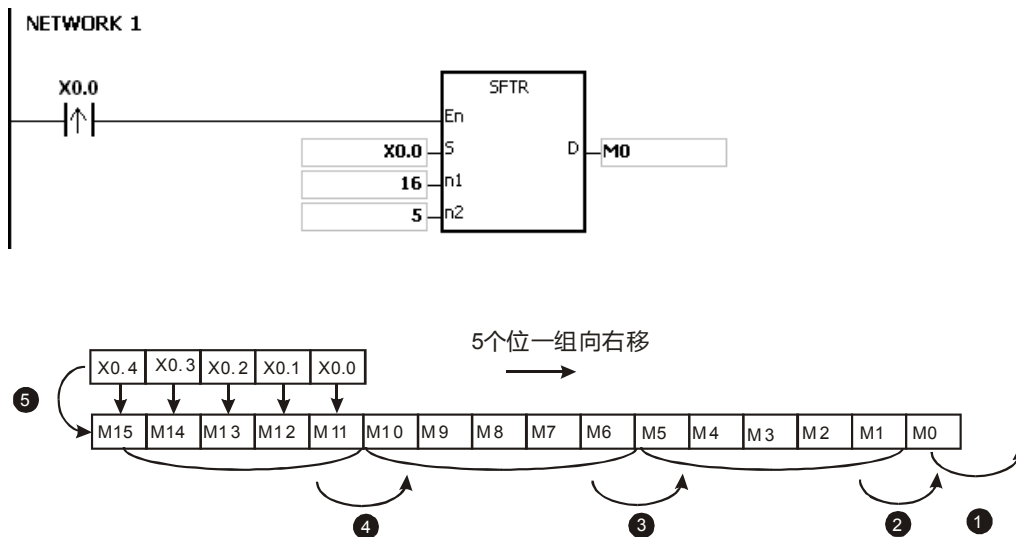
1. 在 X0.0 上升缘时，由 M0~M15 组成 16 位，以 4 位作右移。
2. 扫描一次的位右移动作依照下列编号 1~5 动作。
 - ① M3~M0 → 进位
 - ② M7~M4 → M3~M0
 - ③ M11~M8 → M7~M4
 - ④ M15~M12 → M11~M8
 - ⑤ X0.3~X0.0 → M15~M12 完成



3. 在 X0.0 上升缘时，由 M0~M15 组成 16 位，以 5 位作右移。

4. 扫描一次的位右移动作依照下列编号 1~5 动作。

- ① M0 → 进位
- ② M5 → M0
- ③ M10~M6 → M5~M1
- ④ M15~M11 → M10~M6
- ⑤ X0.4~X0.0 → M15~M11 完成

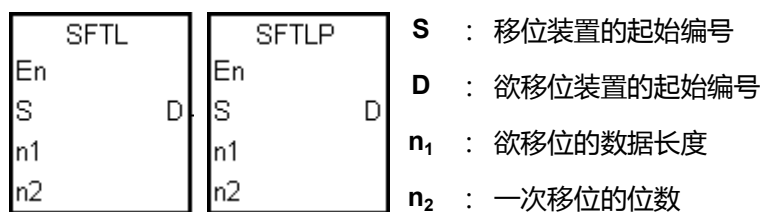


补充说明：

- 1. $S+n_2-1$ 、 $D+n_1-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
- 2. $n_1 < 1$ 或 $n_1 > 1024$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
- 3. $n_2 < 1$ 或 $n_2 > n_1$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数					功能								
FC		SFTL	P	S, D, n₁, n₂					位左移								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S	●																
D	●																
n₁, n₂		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●	●	●				●	●	●			●				
D	●	●	●	●				●	●	●			●				
n₁, n₂	●	●						●	●		●		●	○	○		
							脉冲执行型			16 位指令				32 位指令			
							AH 运动控制主机			AH 运动控制主机				-			

图形：

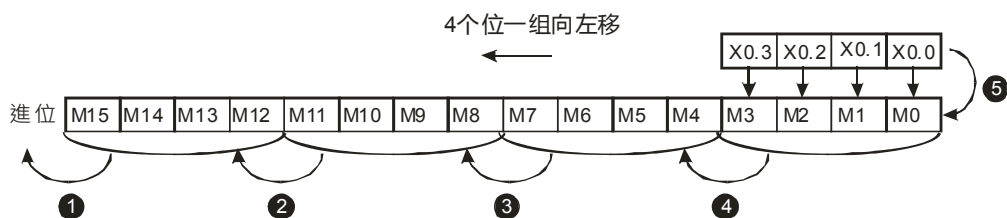
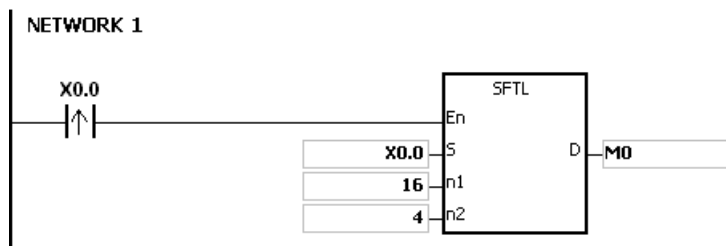


指令说明：

1. 将 **D** 开始的起始编号，具有 **n₁** 个数字元（位移寄存器长度）的位装置，以 **n₂** 位个数来左移。而 **S** 开始起始编号以 **n₂** 位个数移入 **D** 中来填补位空位。
2. 本指令一般都是使用脉冲执行型指令（SFTLP）。
3. **n₁** 值的范围为 1~1024，**n₂** 值的范围为 1~**n₁**。

程序范例：

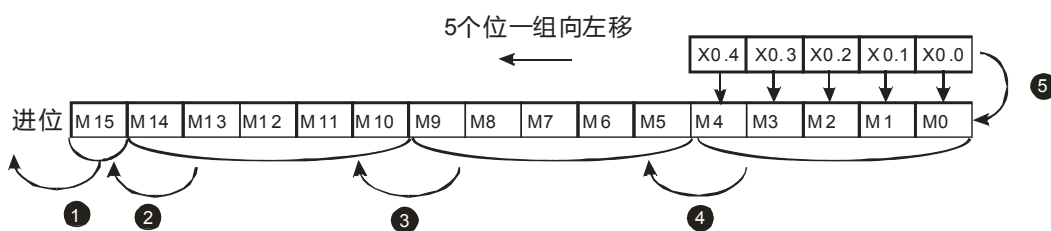
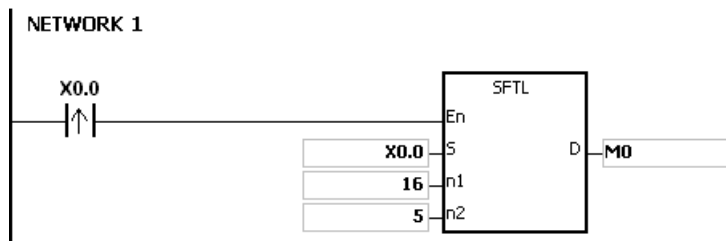
1. 在 X0.0 上升缘时，由 M0~M15 组成 16 位，以 4 位作左移。
2. 扫描一次的位左移动作依照下列编号 1~5 动作。
 - ① M15~M12 → 进位
 - ② M11~M8 → M15~M12
 - ③ M7~M4 → M11~M8
 - ④ M3~M0 → M7~M4
 - ⑤ X0.3~X0.0 → M3~M0 完成



3. 在 X0.0 上升缘时，由 M0~M15 组成 16 位，以 5 位作左移。

4. 扫描一次的位左移动作依照下列编号 1~5 动作。

- ❶ M15 → 进位
- ❷ M10 → M15
- ❸ M9~M5 → M14~M10
- ❹ M4~M0 → M9~M5
- ❺ X0.4~X0.0 → M4~M0 完成

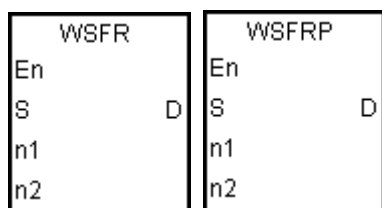


补充说明：

1. $S+n_2-1$ 、 $D+n_1-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 1024$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数					功能								
FC		WSFR	P	S, D, n ₁ , n ₂					寄存器右移								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●					●										
D		●					●										
n ₁ , n ₂		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n ₁ , n ₂	●	●						●	●		●		●	○	○		
							脉冲执行型			16 位指令				32 位指令			
							AH 运动控制主机			AH 运动控制主机				-			

图形：



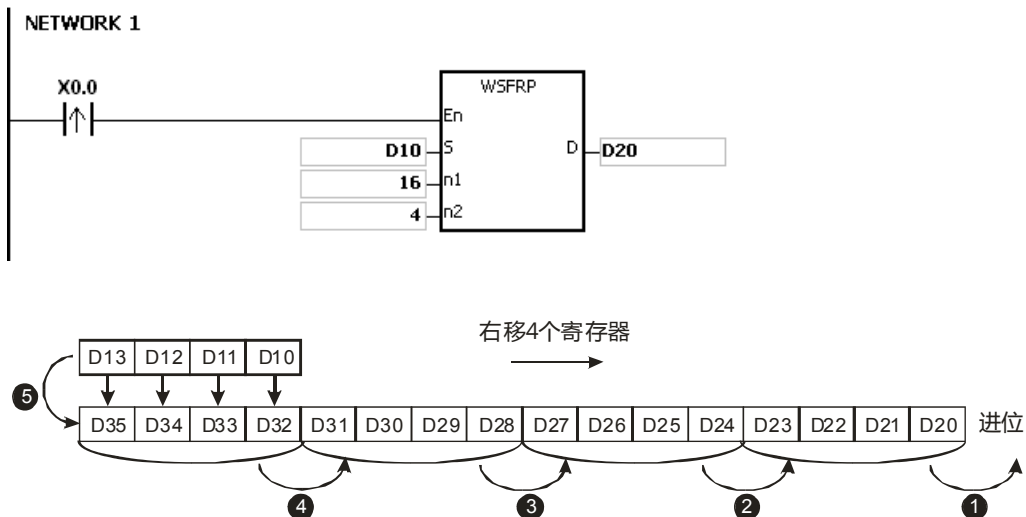
- S : 移位装置的起始编号
D : 欲移位装置的起始编号
n₁ : 欲移位的数据长度
n₂ : 一次移位的位数

指令说明：

- 将 D 开始之起始编号，具有 n₁ 个字符长度的数据串行，以 n₂ 个字符来右移。而 S 开始起始编号以 n₂ 字符个数移入 D 中来填充字符空位。
- 本指令一般都是使用脉冲执行型指令（WSFRP）。
- n₁ 值的范围为 1~512，n₂ 值的范围为 1~n₁。

程序范例一：

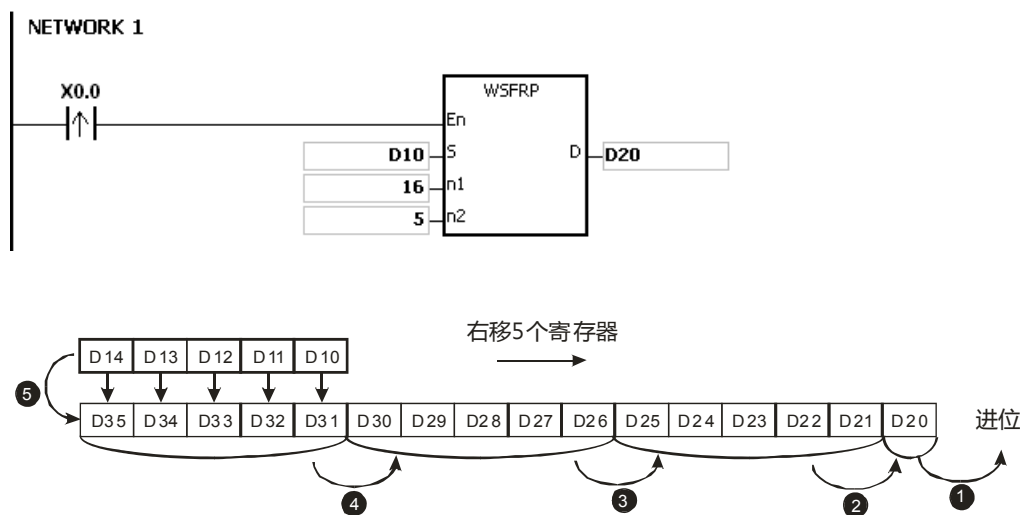
- X0.0=OFF→ON 时，由 D20~D35 所组成的 16 个寄存器数据串行为位移区域，以 4 个寄存器来右移。
- 扫描一次的字符右移动作依照下列编号 1~5 动作。
 - ① D23~D20 → 进位
 - ② D27~D24 → D23~D20
 - ③ D31~D28 → D27~D24
 - ④ D35~D32 → D31~D28
 - ⑤ D13~D10 → D35~D32 完成



程序范例二：

1. X0.0=OFF→ON 时，由 D20~D35 所组成的 16 个寄存器数据串行为位移区域，以 5 个寄存器来右移。
2. 扫描一次的字符右移动作依照下列编号 1~5 动作。

- ① D20 → 进位
- ② D25 → D20
- ③ D30~D26 → D25~D21
- ④ D35~D31 → D30~D26
- ⑤ D14~D10 → D35~D31 完成

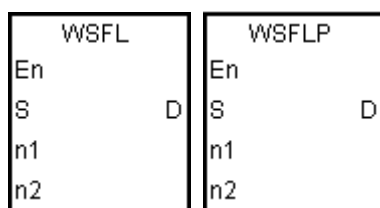


补充说明：

1. S+n₂-1、D+n₁-1 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n₁<1 或 n₁>512 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. n₂<1 或 n₂>n₁ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数						功能							
FC		WSFL	P	S, D, n ₁ , n ₂						寄存器左移							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●					●										
D		●					●										
n ₁ , n ₂		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n ₁ , n ₂	●	●						●	●		●		●	○	○		
							脉冲执行型			16 位指令				32 位指令			
							AH 运动控制主机			AH 运动控制主机				-			

图形：



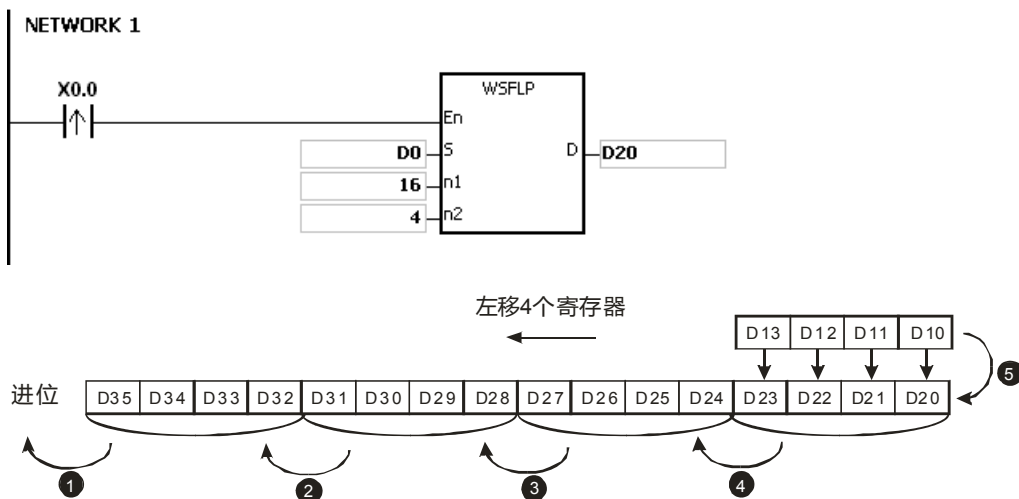
- S : 移位装置的起始编号
D : 欲移位装置的起始编号
n₁ : 欲移位的数据长度
n₂ : 一次移位的位数

指令说明：

1. 将 D 开始之起始编号，具有 n₁ 个字符长度的数据串行，以 n₂ 个字符来左移。而 S 开始起始编号以 n₂ 字符个数移入 D 中来填充字符空位。
2. 本指令一般都是使用脉冲执行型指令（WSFLP）
3. n₁ 值的范围为 1~512，n₂ 值的范围为 1~n₁。

程序范例：

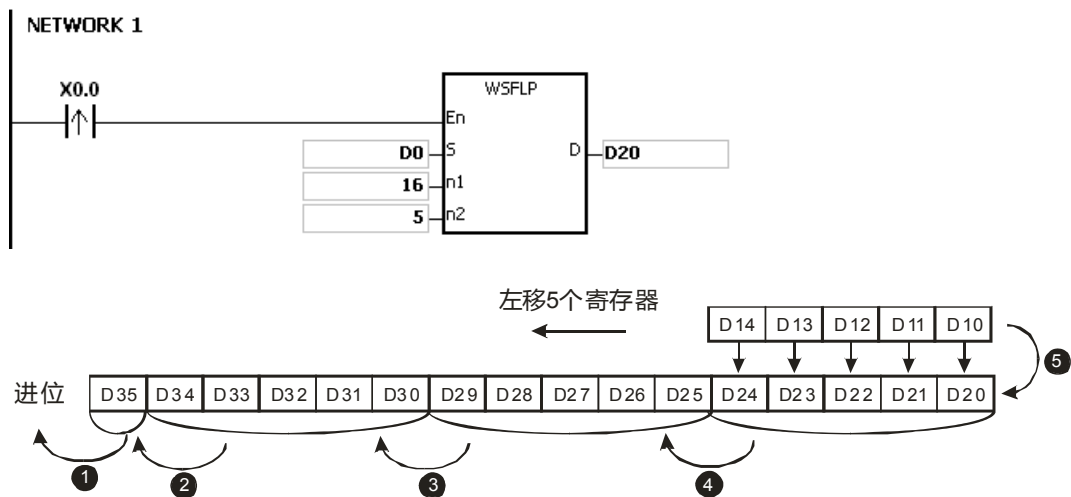
1. 在 X0.0 上升缘时，由 D20~D35 组成 16 位，以 4 位作左移。
2. 扫描一次的位左移动作依照下列编号 1~5 动作。
 - ① D35~D32 → 进位
 - ② D31~D28 → D35~D32
 - ③ D27~D24 → D31~D28
 - ④ D23~D20 → D27~D24
 - ⑤ D13~D10 → D23~D20 完成



3. 在 X0.0 上升缘时，由 D20~D35 组成 16 位，以 5 位作左移。

4. 扫描一次的位左移动作依照下列编号 1~5 动作。

- ❶ D35 → 进位
- ❷ D30 → D35
- ❸ D29~D25 → D34~D30
- ❹ D24~D20 → D29~D25
- ❺ D14~D10 → D24~D20 完成



补充说明：

1. $S+n_2-1$ 、 $D+n_1-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 512$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

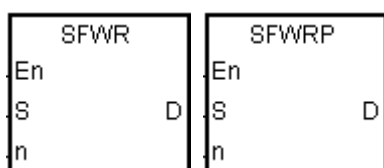
FB/FC	指令码		操作数				功能					
FC		SFWR	P	S, D, n				位移写入				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●		●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S : 位移写入数据串行的装置

D : 数据串行的起始编号

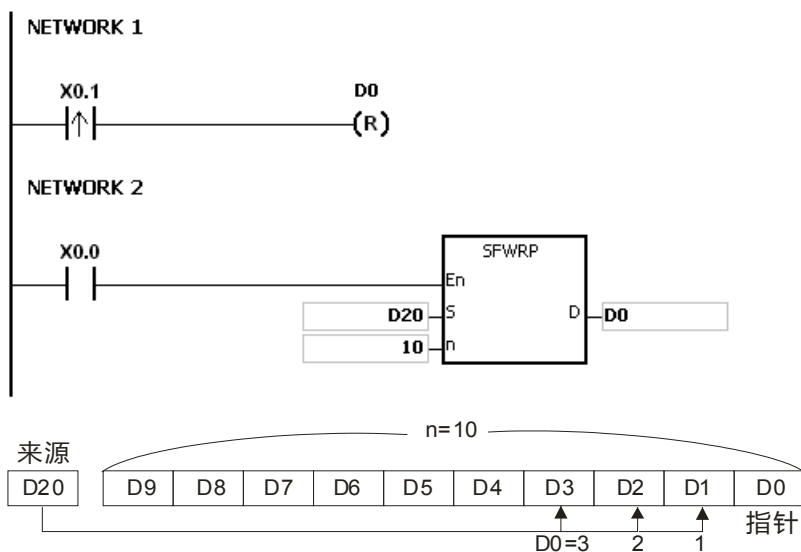
n : 数据串行的长度

指令说明：

1. 将 D 起始编号开始 n 个字符装置的数据串行定义为先入先出数据串行，以第一个编号装置作为指针，当指令执行时，指针内容值先加 1，之后 S 所指定的装置其内容值会写入先入先出数据串行中由指针所指定的位置。当指针的内容大于等于 n-1 时，本指令不再处理写入的新值，进位标志信号 SM602=ON。
2. 本指令一般都是使用脉冲执行型指令（SFWRP）。
3. n 值的范围为 2~512。

程序范例：

1. 开始先将指针 D0 清除为 0，当 X0.0 = OFF→ON 变化时，D20 的内容被传送至 D1 当中，指针 D0 内容变成 1。变更 D20 的内容后，将 X0.0 再 OFF→ON 一次，则 D20 的内容被传送至 D2 当中，D0 内容变成 2。
2. 指令执行一次位移写入动作依照下列动作。
 - D20 的内容被传送至 D1 当中。
 - 指针 D0 内容变成 1。



补充说明：

3

1. D 的内容值 <0 ，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. $n<2$ 或 $n>512$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 本指令 SFWR 与 SFRD 可搭配使用，执行先入先出数据串行的写入读出控制。

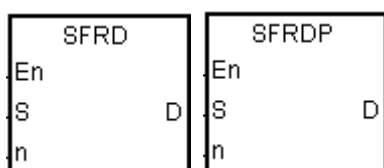
FB/FC	指令码		操作数			功能	
FC		SFRD	P	S, D, n			位移读出

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●	○	○		
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S : 数据串行之起始编号

D : 数据串行位移读出之装置

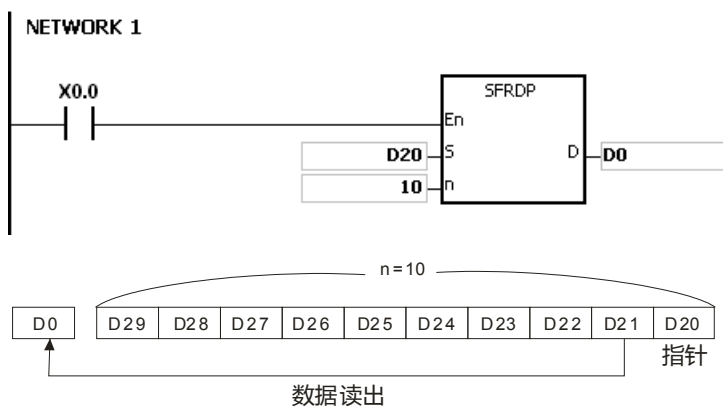
n : 数据串行之长度

指令说明：

1. 将 **S** 起始编号开始 **n** 个字符装置的数据串行定义为先入先出数据串行，以第一个编号装置作为指针，当指令执行时，**S** 内容值先减 1，之后 (**S+1**) 所指定的装置其内容值会写入 **D** 所指定的位置，(**S+n-1**) ~ (**S+2**) 全部右移一个寄存器，(**S+n-1**) 的内容不变，当 **S** 的内容等于 0 时，本指令不再处理数据读出的动作，零标志信号 SM600=ON。
2. 本指令一般都是使用脉冲执行型指令 (SFRDP)。
3. **n** 值的范围为 2~512。

程序范例：

1. 当 X0.0 从 OFF→ON 变化时，D21 的内容被传送至 D0 当中，D29~D22 全部往右位移一个寄存器 (D29 内容保持不变)，D20 内容减 1。
2. 指令执行一次位移读出动作依照下列动作。
 - D21 的内容被读出传送至 D0 当中。
 - D29~D22 全部往右位移一个寄存器。
 - D20 内容减 1。



补充说明：

1. $S < 0$ ，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $S+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. $n < 2$ 或 $n > 512$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 本指令 SFWR 与 SFRD 可搭配使用，执行先入先出数据串行的写入读出控制。

3

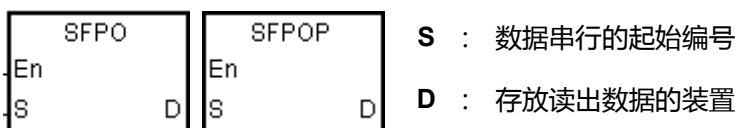
FB/FC	指令码		操作数				功能						
FC		SFPO	P	S, D				读出数据串行最新数据					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

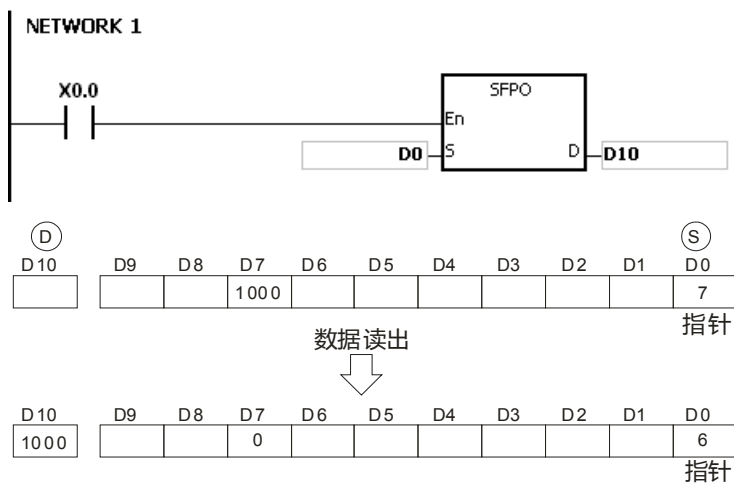


指令说明：

1. 将 **S** 起始编号开始的数据串行，以第一个编号装置作为指针，当指令执行时，**S** 内容值所指定的指针内容值会写入 **D** 所指定的位置，**S** 内容值所指定的指针内容值被清除为 0，**S** 内容值减一，当 **S** 的内容等于 0 时，本指令不再处理数据读出的动作，零标志信号 SM600=ON。
2. 本指令一般都是使用脉冲执行型指令（SFPOP）。

程序范例：

当 X0.0=ON 时，D0 内容所指定的位置内容搬移至 D10 当中，搬移后指定位置清除为 0，D0 内容减一。



补充说明：

1. **S** 内容值 <0 ，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **S** 编号+**S** 内容值，装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

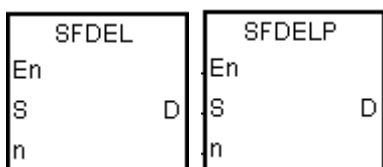
FB/FC	指令码		操作数				功能				
FC	SFDEL	P	S, D, n				删除数据串行中的数据				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●	○	○		
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



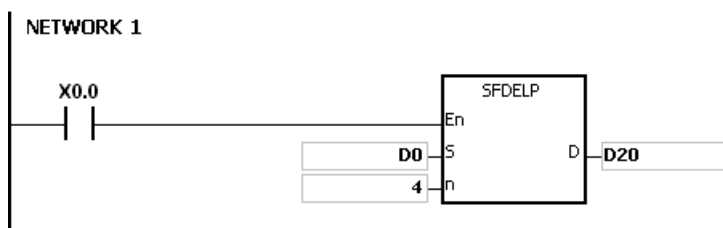
- S : 数据串行的起始编号
- D : 储存被删除的数据的装置
- n : 要删除的位置

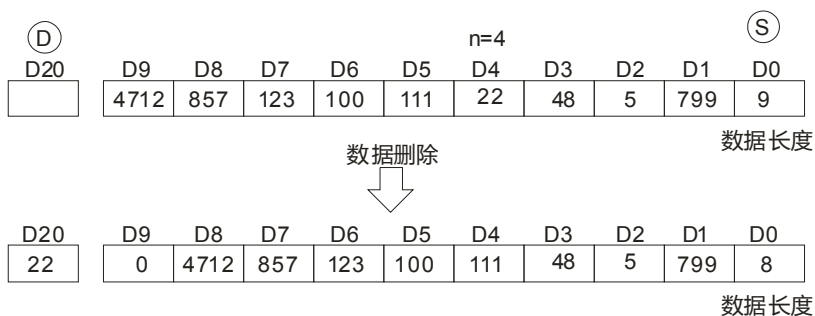
指令说明：

1. S 为数据串行的长度，(S+1) ~ (S+S 内容值) 为数据串行的内容，当指令执行时，S+n 的内容值会被储存在 D 中，并删除 S+n 的内容值，然后 (S+n+1) ~ (S+S 内容值) 的装置内容全部右移一个寄存器，(S+S 内容值) 的内容归零，最后将 S 内容值减一，当 S 的内容等于 0 时，本指令不再处理数据删除的动作，零标志信号 SM600=ON。
2. 本指令一般都是使用脉冲执行型指令 (SFDELP)。
3. n 操作数值的范围为 1~32767。

程序范例：

给定 D0=9, 当 X0.0=ON 时, 将 n=4 指定的 D4 内容搬移至 D20 当中, 并删除 D4 的内容值, 然后将 D5~D9 的内容向前递补, 最后将 D0 内容减一。





补充说明：

1. **S** 内容值<0，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **S+n** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. **S+S** 内容值，装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. **n>S** 内容值，指令不执行，SM0=ON，错误码 SR0=16#200B。
5. **n<=0** 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数				功能				
FC		SFINS	P	S, D, n				插入数据到数据串行中			

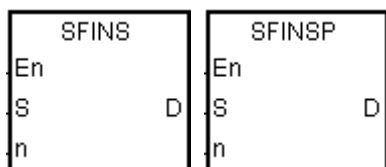
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●	○	○		
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



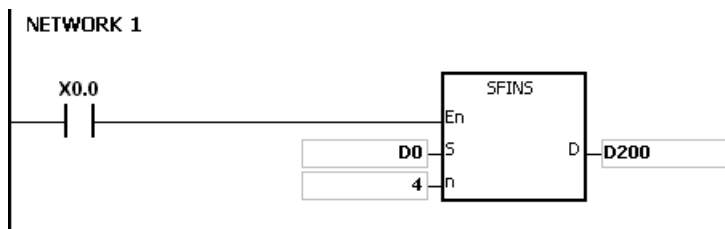
- S** : 数据串行的起始编号
- D** : 要插入数据串行的数据
- n** : 要插入的位置

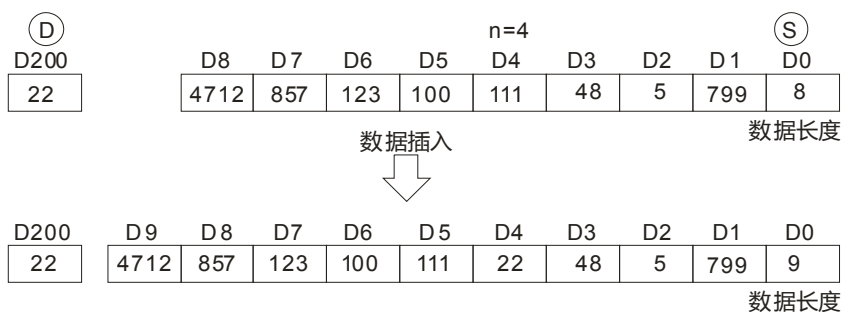
指令说明：

1. **S** 为数据串行的长度，(**S+1**) ~ (**S+S** 内容值) 为数据串行的内容，当指令执行时，**D** 的内容值会被插入 **S+n** 中，原本 (**S+n**) ~ (**S+S** 内容值) 的数据全部往下移动一个寄存器，最后将内容值加一，**S** 内容值=32767 时，本指令不再处理写入的新值，**S** 内容值不再往上增加，进位标志信号 SM602=ON。
2. 本指令一般都是使用脉冲执行型指令 (SFINSP)。
3. **n** 操作数值的范围为 1~32767。

程序范例：

给定 D0=8，当 X0.0=ON 时，先将 D200 的内容值插入 n=4 指定的 D4 中，再将原先 D4~D8 的内容值向下位移填入 D5~D9 中，最后将 D0 内容加一。





补充说明：

1. $S \leq 0$ ，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $S+n$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. $S+S$ 内容值+1，装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. $n>S$ 内容值，指令不执行，SM0=ON，错误码 SR0=16#200B。
5. $n < 0$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

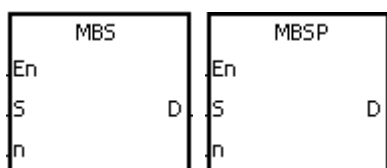
FB/FC	指令码		操作数				功能					
FC		MBS	P	S, D, n				矩阵位移				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●				●				
n	●	●			●	●	●	●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S : 矩阵来源装置

D : 运算结果

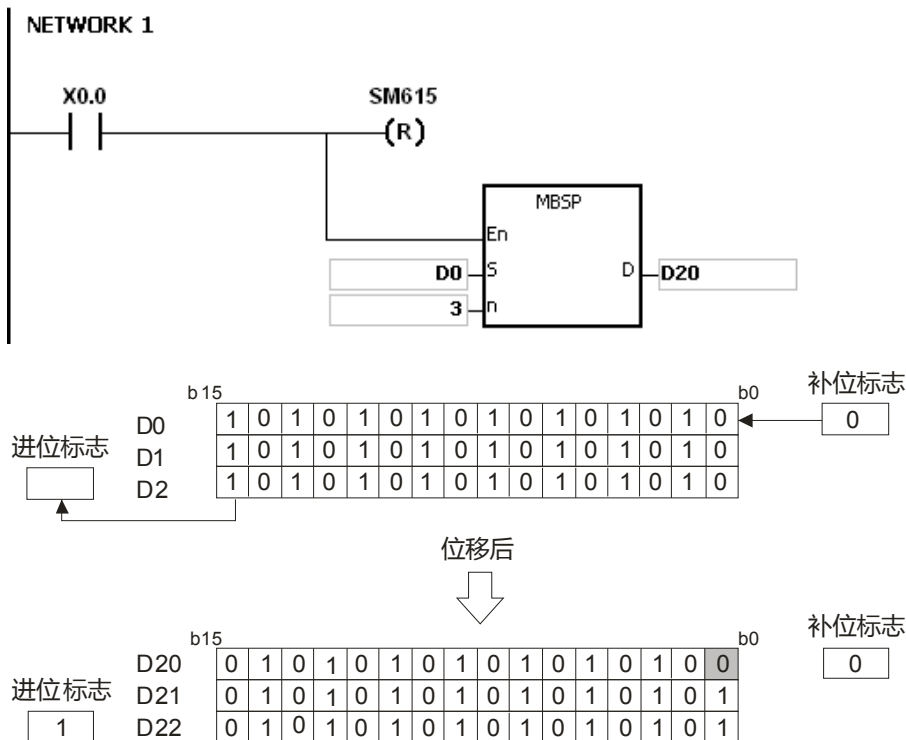
n : 数组长度

指令说明：

1. 矩阵来源依数组长度 **n** 将 **S** 矩阵位做左右位移控制，SM616=0 决定矩阵位左移，SM616=1 决定矩阵位右移。每次移动一位，因位移而腾出的空位（左移时为 b0，右移时为 b16n-1）则以补位标志 SM615 的状态填补。而因位移而挤出之位（左移时为 b16n-1，右移时为 b0）状态则送到进位标志 SM614 去，然后将结果存入 **D**。
2. **n** 操作数值的范围为 1~256。
3. 本指令一般都是使用脉冲执行型指令（MBSP）。

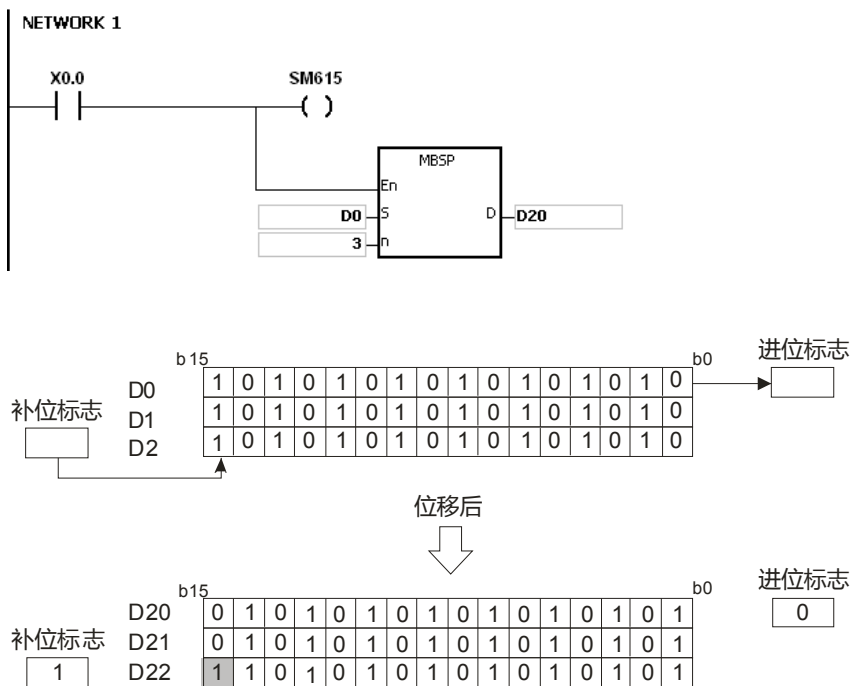
程序范例一：

当 X0.0=ON 时，矩阵循环位移方向标志 SM616=OFF 作矩阵左移。设矩阵位移输入补位标志 SM615=0，16 位寄存器 D0~D2 矩阵作左移，将结果存于 16 位寄存器 D20~D22 矩阵中，矩阵循环位移输出进位标志 SM614=1。



程序范例二：

当 X0.0=ON 时，矩阵循环位移方向标志 SM616=ON 作矩阵右移，设矩阵位移输入补位标志 SM615=1，16 位寄存器 D0~D2 矩阵作右移，将结果存于 16 位寄存器 D20~D22 矩阵中，矩阵循环位移输出进位标志 SM614=0。



补充说明：

1. S+n-1、D+n-1 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

2. $n < 1$ 或 $n > 256$ 时, 指令不执行, $SM0=ON$, 错误码 $SR0=16\#200B$ 。

3. 标志信号说明

SM614 : 矩阵循环位移输出进位标志

SM615 : 矩阵位移输入补位标志

SM616 : 矩阵循环位移方向标志

FB/FC	指令码		操作数		功能
FC		SFR	P	D, n	16 位寄存器位右移

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



指令说明：

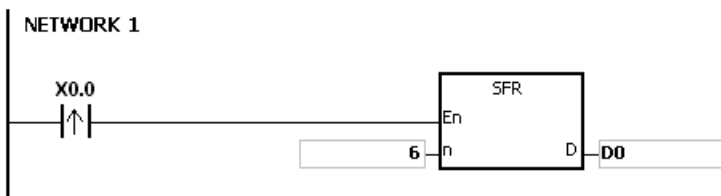
1. 将 **D** 往右移 **n** 个数，而因位移而腾出之空位 ($b_{15} \sim b_{15-n+1}$) 则补零。因位移而挤出的第 **n-1** 位 (b_{n-1}) 状态则送到进位标志 SM602 去。
2. **n** 内容值的范围为 1~16。
3. 本指令一般都是使用脉冲执行型指令 (SFRP)。

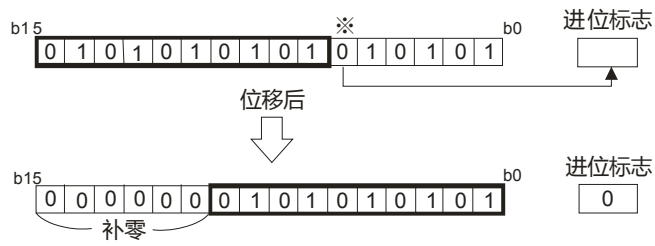
程序范例：

当 X0.0=ON 时，D0 的 b0~b15，往右移动 6 个数，而 b5 的状态传送到 SM602，位移后，b10~b15 内容清除为 0。

扫描一次的位右移动作依照下列编号 1~3 动作。

- ❶ b5~b0 → 进位 (SM602=b5)
- ❷ b15~b6 → b9~b0
- ❸ 0 → b15~b10 完成



**补充说明：**

n<0 或 n>16，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数		功能
FC		SFL	P	D, n	16 位寄存器位左移

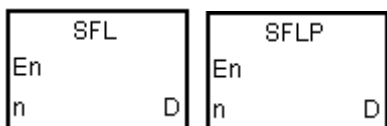
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



D : 欲位移的装置
n : 一次位移的位数

指令说明：

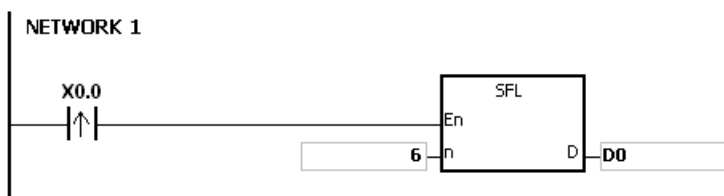
1. 将 **D** 往左移 **n** 个数，而因位移而腾出的空位 ($b_0 \sim b_{n-1}$) 则补零。因位移而挤出的第 $16-n$ 位 (b_{16-n}) 状态则送到进位标志 SM602 去。
2. **n** 内容值的范围为 1~16。
3. 本指令一般都是使用脉冲执行型指令 (SFLP)。

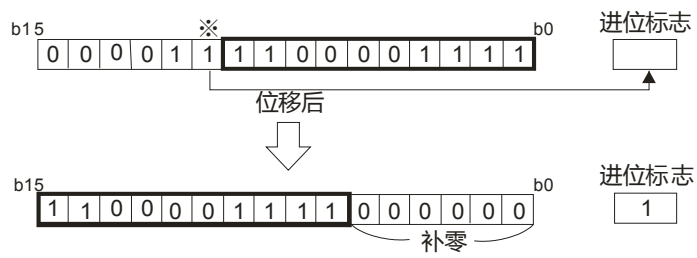
程序范例：

当 X0.0=ON 时，D0 的 b0~b15，往左移动 6 个数，而 b10 的状态传送到 SM602，位移后，b0~b5 内容清除为 0。

扫描一次的位右移动作依照下列编号 1~3 动作。

- ❶ b15~b10 → 进位 (SM602=b10)
- ❷ b9~b0 → b15~b6
- ❸ 0 → b5~b0 完成



**补充说明：**

$n < 0$ 或 $n > 16$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数		功能
FC	BSFR	P	D, n		n 个位右移 1 个位 (位装置)

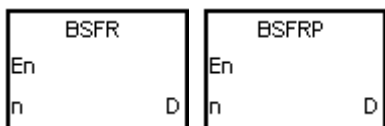
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●													
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●	●	●				●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



D : 移位装置的起始编号

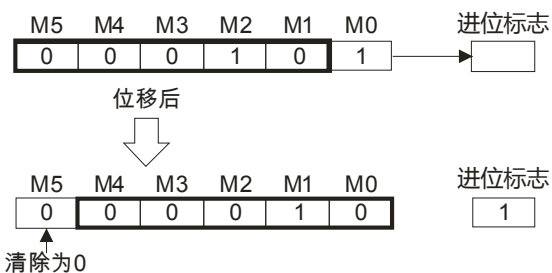
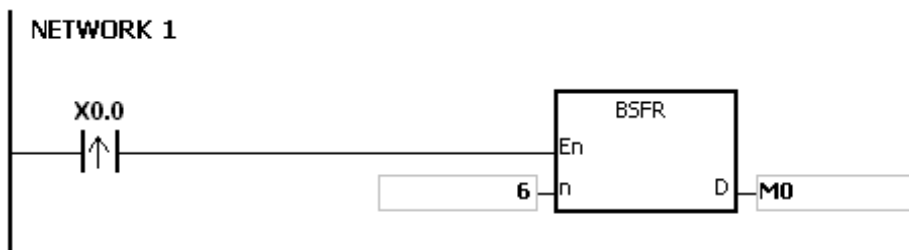
n : 数据长度

指令说明：

1. 将 D 开始的 n 个位，右移 1 个位，D+n-1 位被清除为 0，D 位状态则送到进位标志 SM602。
2. 本指令一般都是使用脉冲执行型指令 (BSFRP)。
3. n 操作数的范围为 1~1024。

程序范例：

当 X0.0=ON 时，M0~M5 依序向右位移 1 个位，M5 的状态清除为 0，M0 的状态传送至进位标志 SM602。



补充说明：

1. $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n<1$ 或 $n>1024$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数	功能
FC	BSFL	P	D, n	n 个位左移 1 个位

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●													
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●	●	●				●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



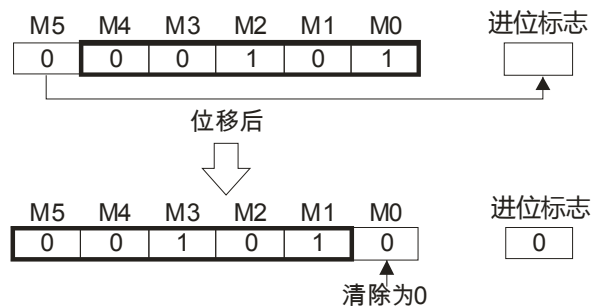
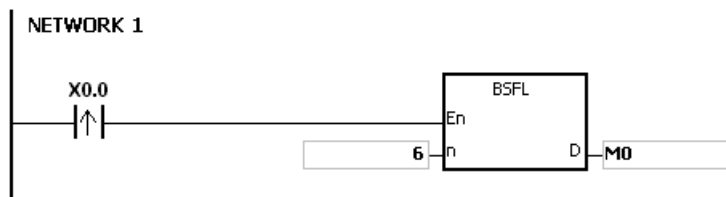
D : 移位装置的起始编号
n : 数据长度

指令说明：

1. 将 **D** 开始的 **n** 个位，左移 1 个位，**D** 位被清除为 0，**D+n-1** 位状态则送到进位标志 SM602。
2. 本指令一般都是使用脉冲执行型指令（BSFLP）。
3. **n** 操作数的范围为 1~1024。

程序范例：

当 X0.0=ON 时，M0~M5 内容依序向左移位 1 个位，M0 的状态清除为 0，M5 的状态传送至进位标志 SM602。



补充说明：

1. $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n<1$ 或 $n>1024$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数		功能
FC		NSFR	P	D, n	n 个寄存器右移

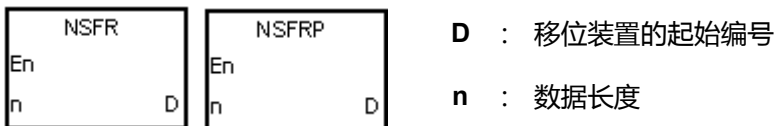
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

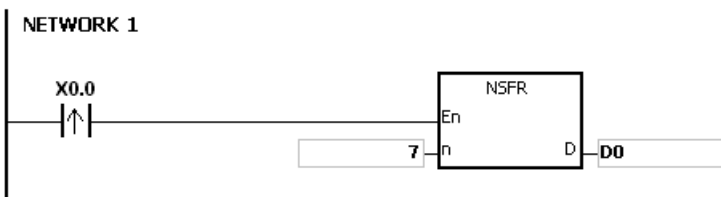


指令说明：

1. 将 **D** 开始的 **n** 个寄存器右移，**D+n-1** 装置被清除为 0。
2. 本指令一般都是使用脉冲执行型指令（NSFRP）。
3. **n** 操作数的范围为 1~512。

程序范例：

当 X0.0=ON 时，D1~D6 内容依序向右位移 1 个寄存器位置，最后 D6 的内容清除为 0。



补充说明：

1. $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n<1$ 或 $n>512$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数		功能
FC		NSFL	P	D, n	n 个寄存器左移

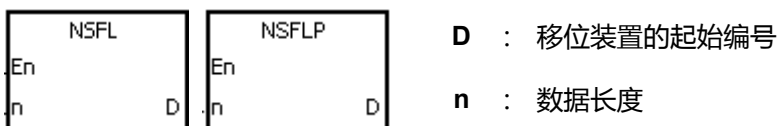
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

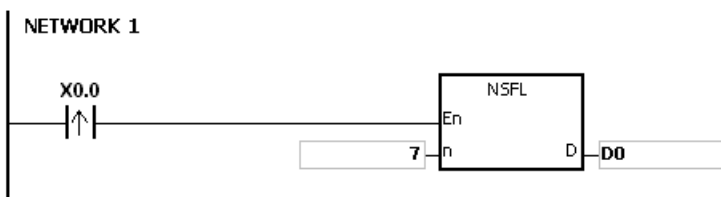


指令说明：

1. 将 D 开始的 n 个寄存器左移，D 装置被清除为 0。
2. 本指令一般都是使用脉冲执行型指令（NSFLP）。
3. n 操作数的范围为 1~512。

程序范例：

当 X0.0=ON 时，D0~D5 内容依序向左位移 1 个寄存器位置，最后 D0 的内容清除为 0。



D6	D5	D4	D3	D2	D1	D0
30	2235	9578	754	28	423	11

位移后



D6	D5	D4	D3	D2	D1	D0
2235	9578	754	28	423	11	0

清除为0

补充说明：

1. $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n<1$ 或 $n>512$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

3.15 数据处理指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>SER</u>	<u>DSER</u>	✓	多点比较	9
FC	<u>SUM</u>	<u>DSUM</u>	✓	ON 位数量	5
FC	<u>DECO</u>	—	✓	译码器	7
FC	<u>ENCO</u>	—	✓	编码器	7
FC	<u>SEGD</u>	—	✓	7 段显示器解码	5
FC	<u>SORT</u>	<u>DSORT</u>	—	数据排序	11
FC	<u>ZRST</u>	—	✓	区域清除	5
FC	<u>BON</u>	<u>DBON</u>	✓	ON 位判定	7
FC	<u>MEAN</u>	<u>DMEAN</u>	✓	平均值	7
FC	<u>CCD</u>	—	✓	总和检查	7
FC	<u>ABS</u>	<u>DABS</u>	✓	绝对值	3
FC	<u>MINV</u>	—	✓	矩阵反相	7
FC	<u>MBRD</u>	—	✓	矩阵位读出	7
FC	<u>MBWR</u>	—	✓	矩阵位写入	7
FC	<u>MBC</u>	—	✓	矩阵位状态计数	7
FC	<u>DIS</u>	—	✓	16位数据的4位分组	7
FC	<u>UNI</u>	—	✓	16位数据的4位链结	7
FC	<u>WSUM</u>	<u>DWSUM</u>	✓	求和	7
FC	<u>BSET</u>	—	✓	字符装置位动作保持 ON	5
FC	<u>BRST</u>	—	✓	字符装置位清除	5
FC	<u>BKRST</u>	—	✓	指定区域清除	5
FC	<u>LIMIT</u>	<u>DLIMIT</u>	✓	高低限控制	9
FC	<u>BAND</u>	<u>DBAND</u>	✓	死区控制	9
FC	<u>ZONE</u>	<u>DZONE</u>	✓	区域控制	9

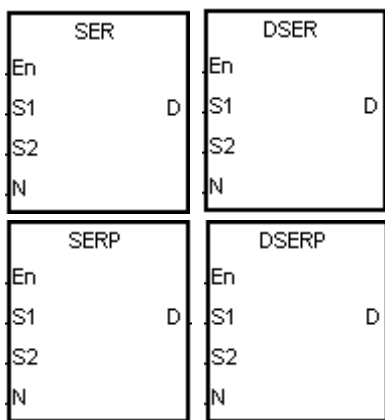
FB/FC	指令码			操作数	功能
FC	D*	SER	P	S ₁ , S ₂ , D, n	多点比较

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●	●*				●	●*						
S ₂		●	●*				●	●*						
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●	●	●	●		●		●				
S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●				●				
n	●	●			●	●	●	●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



S₁：多点比较的数据区块的起始装置

S₂：欲比较的数值数据

D：存放比较结果的起始装置

n：被比较的数据区块长度

指令说明：

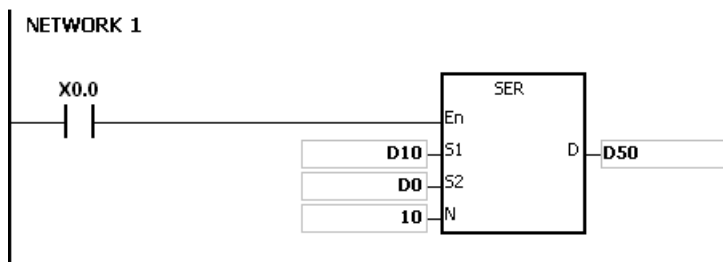
- S₁ 指定被比较寄存器区域的号码，n 指定被比较的笔数，该多笔被比较寄存器的内容与 S₂ 所指定的数据，以有号数十进制作比较，比较结果被存放于 D~D+4 寄存器当中，并说明如下：

装置	说明
D	相等值的数据数
D+1	第一个相等值的编号
D+2	最后一个相等值的编号
D+3	最小值的编号
D+4	最大值的编号

- 16 位指令中，n 操作数的内容值为 1~256；32 位指令中，n 操作数的内容值为 1~128。
- 32 位指令才可以使用 HC 装置。

程序范例：

- 当 X0.0=ON 时，由 D10~D19 组成之数据区块与 D0 作比较，结果存放在 D50~D54 中，当相等值不存在时，D50~D52 的内容全部为 0。
- 所有比较数据之最小值编号记录在 D53，最大值编号记录在 D54。当最小值、最大值不只一个时，会记录编号大者。



S ₁	内容值	比较数据	数据编号	结果	D	内容值	说明
D10	88	S ₂ D0=100	0		D50	4	相等值的数据数
D11	100		1	相等	D51	1	第一个相等值的编号
D12	110		2		D52	8	最后一个相等值的编号
D13	150		3		D53	7	最小值的编号
D14	100		4	相等	D54	9	最大值的编号
D15	300		5				
D16	100		6	相等			
D17	5		7	最小			
D18	100		8	相等			
D19	500		9	最大			

补充说明：

- S₁+n-1 或 D+4，超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
- 16 位指令 n<1 或 n>256 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
- 32 位指令 n<1 或 n>128 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
- 16 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [5] of WORD/INT。
- 32 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [5] of DWORD/DINT。

FB/FC	指令码			操作数	功能
FC	D*	SUM	P	S, D	ON 位数量

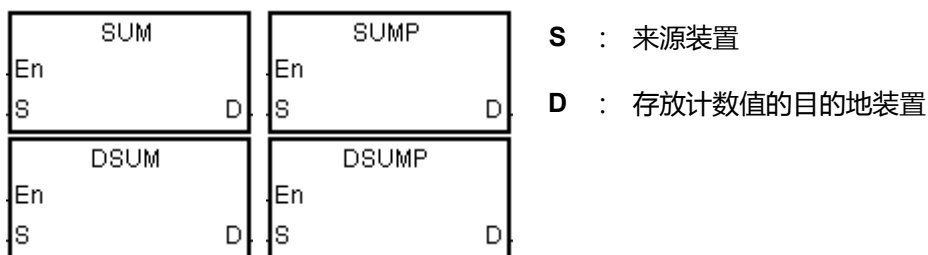
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：

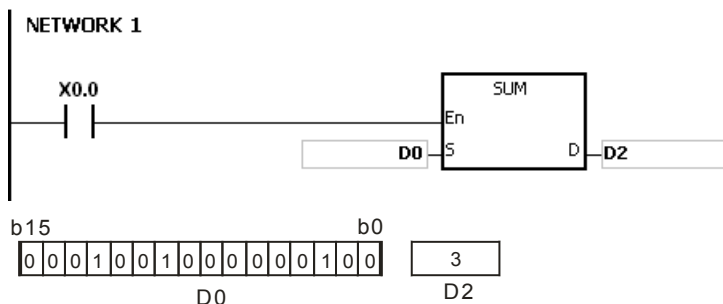


指令说明：

1. 在 S 中，所有位内容为“1”的总数将被储存于 D。
2. 如果来源装置 S 的位全部为“0”时，零标志信号 SM600=ON。
3. 32 位指令才可以使用 HC 装置。

程序范例：

当 X0.0 为 ON 时，D0 的 16 个位中，内容为“1”的位总数被存于 D2 当中。



补充说明：

装置超出范围时，指令不执行，SM0=ON，SR0 错误码=16# 2003。

FB/FC	指令码		操作数				功能				
FC		DECO	P	S, D, n				译码器			

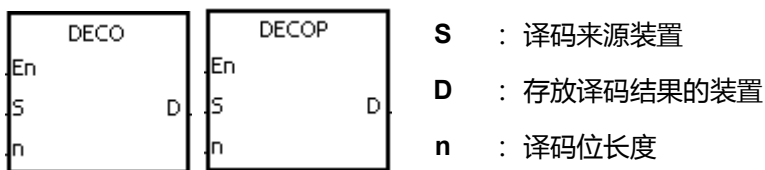
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●	●					●							
D	●	●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●	●	●	●	●		●	●	●	●	○	●	○	○		
D	●	●	●	●	●	●		●	●	●	●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

3

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

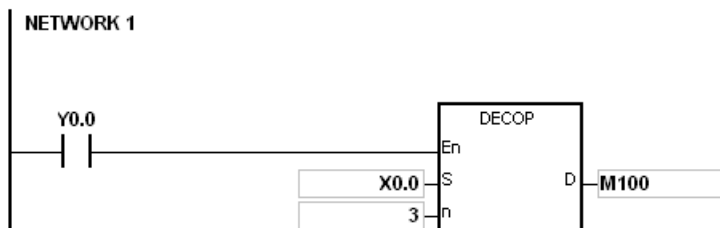


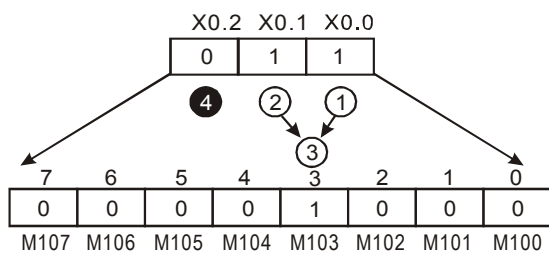
指令说明：

1. 来源装置 S 的下位“n”位作译码，并将其“2ⁿ”位长度的结果存于 D。
2. 当 D 为位装置时，n = 1~8，当 n=8 时，可做最大解码 2⁸=256 点。（须注意译码后的装置储存范围，勿重复使用）。
3. D 为字符装置时，n = 1~4，当 n=4 时，可做最大解码 2⁴=16 点。
4. 本指令一般都是使用脉冲执行型指令（DECOP）。

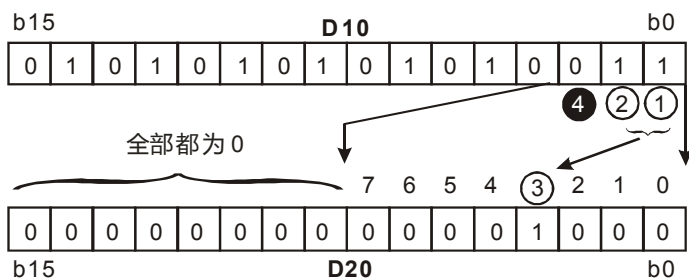
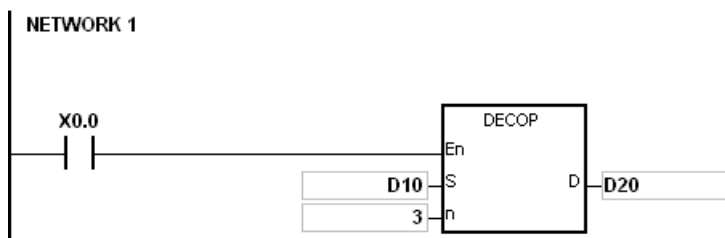
程序范例一：

1. Y0.0 = OFF→ON 时，DECO 指令将 X0.0~X0.2 的内容值译码到 M100~M107。
2. 当数据源 X0.0~X0.2 的内容值为 1+2 = 3 时，从 M100 开始算第 3 个位 M103 设定为 1。
3. 当 DECO 指令执行过后，而 Y0.0 变为 OFF，已经做解码输出者照常动作。



**程序范例二：**

1. X0.0 = OFF→ON 时，DECO 指令将 D10 中 (b2~b0) 的内容值解码到 D20 的 (b7~b0)。D20 中未被使用之位 (b15~b8) 全部变为 0。
2. D10 的下位 3 位作译码存放于 D20 的下位 8 位，上 8 位皆为 0。
3. 当 DECO 指令执行过后，而 X0.0 变为 OFF 后，已经做解码输出者照常动作。

**补充说明：**

1. D 为位装置时，若 $n < 1$ 或 $n > 8$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. D 为字符装置时，若 $n < 1$ 或 $n > 4$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 若 S 为位装置时， $S+n-1$ 位超过范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. 若 D 为位装置时， $D+(2^n)-1$ 位超过范围，指令不执行，SM0=ON，错误码 SR0=16#2003。

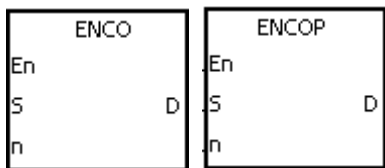
FB/FC	指令码		操作数			功能					
FC		ENCO	P	S, D, n			编码器				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●	●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●	●	●	●	●		●	●	●	●	○	●				
D	●	●			●	●		●	●			○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



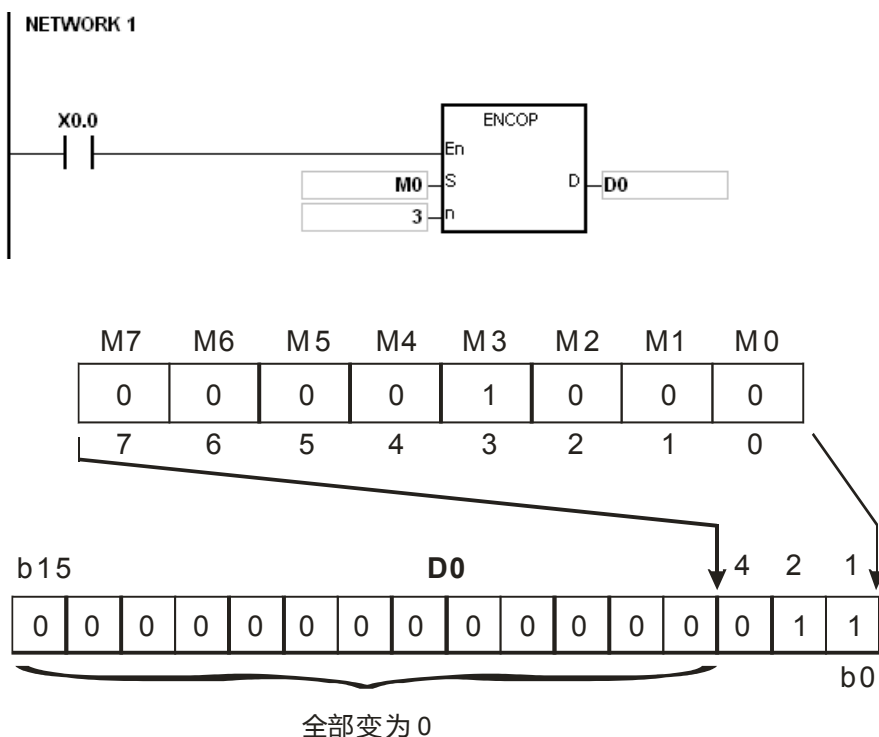
S : 编码来源装置
D : 存放编码结果的装置
n : 编码位长度

指令说明：

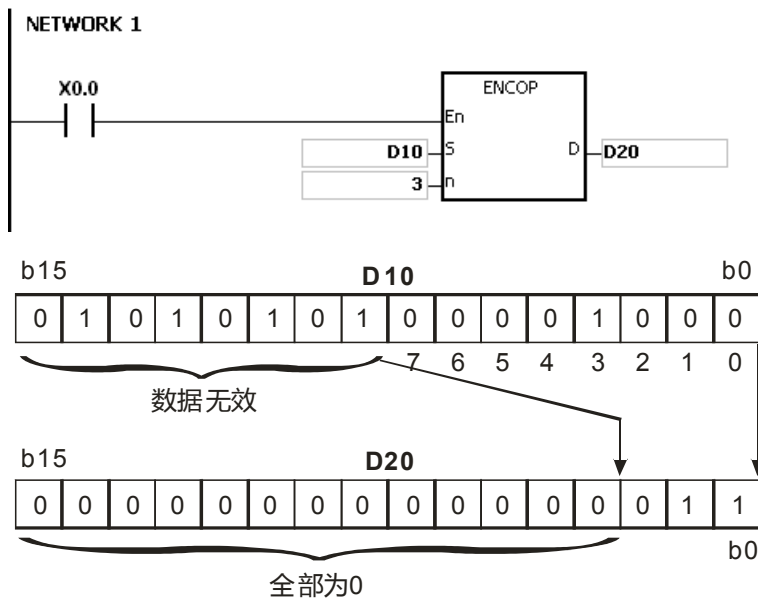
1. 来源装置 **S** 的下位“ 2^n ”位长度的数据作编码，并将结果存于 **D**。
2. 若数据源 **S** 有多数位为 1 时，则处理由高位往低位的第 1 个为 1 之位。
3. **S** 为位装置时， $n = 1\sim 8$ ，当 $n=8$ 时，可做 $2^8=256$ 点编码。
4. **S** 为字符装置时， $n = 1\sim 4$ ，当 $n=4$ 时，可做 $2^4=16$ 点编码。
5. 本指令一般都是使用脉冲执行型指令（ENCOP）。

程序范例一：

1. 当 $X0.0 = OFF \rightarrow ON$ 时，ENCO 指令将 2^3 位数据（M0~M7）编码存放于 D0 的下位 3 位（b2~b0）内，D0 中未被使用的位（b15~b3）全部变为 0。
2. 当 ENCO 指令执行过后，而 $X0.0$ 变为 OFF 后，**D** 内数据不变。

**程序范例二：**

1. 当 X0.0 = OFF→ON 时，D10 内 2^3 bits 数据 (b0~b7) 编码存放于 D20 的下位 3 位 (b2~b0) 内，D20 中未被使用的位 (b15~b3) 全部变为 0。(D10 内 b8~b15 为无效数据)
2. 当 ENCO 指令执行过后，而 X0.0 变为 OFF，D 内数据不变。

**补充说明：**

1. 若数据源 S 都没有位为 1 时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. S 为位装置时，若 $n < 1$ 或 $n > 8$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. S 为字符装置时，若 $n < 1$ 或 $n > 4$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

4. 若 **S** 为位装置时, $S + (2^n) - 1$ 位超过范围, 指令不执行, SM0=ON, 错误码 SR0=16#2003。
5. 若 **D** 为位装置时, $D+n-1$ 位超过范围, 指令不执行, SM0=ON, 错误码 SR0=16#2003。

FB/FC	指令码		操作数				功能					
FC		SEGD	P	S, D				7 段显示器解码				

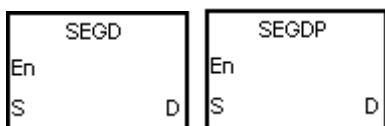
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



S : 欲译码的来源装置

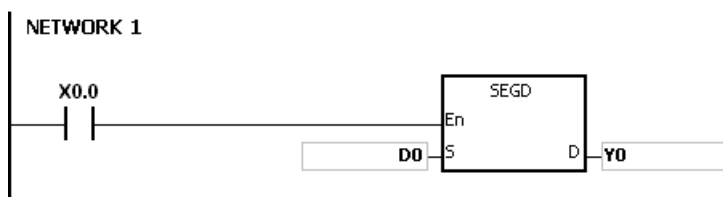
D : 译码后的输出装置

指令说明：


来源装置 **S** 的下位 4 个位 (b0~b3) ，被解码成 7 段显示器的格式，并将结果存于 **D**。

程序范例：

当 X0.0=ON 时，D0 的下位 4 个位 (b0~b3) 的内容 (0~F : 16 进制) 被解码成 7 段显示器输出，解码的结果暂存于 Y0.0~Y0.15 当中。若指定数据超出 4 个位，仍取下位 4 个位的内容译码。



七段显示器解码表：

16进数	位组合	七段显示器的构成	各节段状态						显示数据	
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)		B6(g)
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	ON	OFF	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	b
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	c
D	1101		OFF	ON	ON	ON	ON	OFF	ON	d
E	1110		ON	OFF	OFF	ON	ON	ON	ON	E
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	F

3

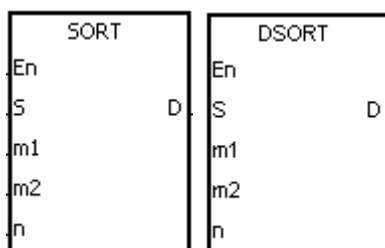
FB/FC	指令码		操作数				功能			
FC	D*	SORT	S, m ₁ , m ₂ , D, n				数据排序			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
m ₁ , m ₂		●	●*				●	●*						
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				
m ₁ , m ₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●				●				
n	●	●			●	●	●	●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	AH 运动控制主机

图形：



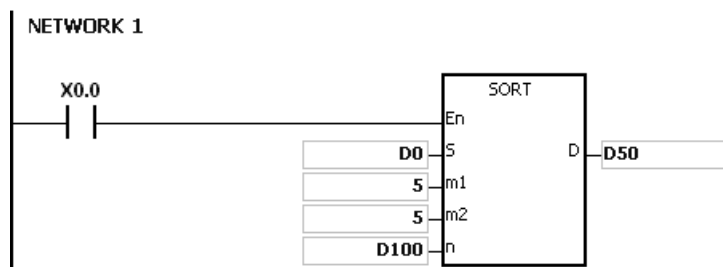
- S : 原始数据区块的起始装置
m₁ : 被排序之数据组数
m₂ : 每笔数据之字段数
D : 存放排序结果数据区块的起始装置
n : 数据排序的参考值

指令说明：

1. 排序结果显示于 D 所指定的起始号码开始算的 m₁×m₂ 个寄存器当中，因此，S 与 D 指定同一个寄存器的话，排序结果将与原来被排序的数据 S 内容相同。
2. m₁=1~32, m₂=1~6, n=1~m₂。
3. S 寄存器的起始号码的最右边编号指定 0 比较理想。
4. n 的内容值在指令执行中被改变时，则指令会以被更改后的值来做排序。
5. SM604=OFF 时为升幂排序，SM604=ON 时为降序排序。
6. 本指令的执行条件，建议使用脉冲执行指令，避免一直重复排序。
7. 32 位指令才可以使用 HC 装置。

程序范例：

将 SM604=OFF，当 X0.0=OFF→ON 时，指定执行数据进行升幂排序作业，排序完成时，SM603=ON。



1. 排序数据构成例

		数据数：m ₂ 个				
		数据字段				
	行	1	2	3	4	5
	列	学生编号	国文	英文	数学	理化
↑ 数据 个 数 : m ₁ 个 ↓	1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
	2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
	3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
	4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
	5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

2. D100=3 时的排序后数据

		数据数： m_2 个				
		数据字段				
↑ 数据 个 数 ： m_1 个 ↓	行 列	1	2	3	4	5
		学生编号	国文	英文	数学	理化
	1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
	2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
	3	(D52) 1	(D57) 90	(D62) 75	(D67) 66	(D72) 79
	4	(D53) 5	(D58) 95	(D63) 79	(D68) 75	(D73) 69
	5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

3. D100=5 时的排序后数据。

		数据数： m_2 个				
		数据字段				
↑ 数据 个 数 ： m_1 个 ↓	行 列	1	2	3	4	5
		学生编号	国文	英文	数学	理化
	1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
	2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
	3	(D52) 5	(D57) 95	(D62) 79	(D67) 75	(D72) 69
	4	(D53) 1	(D58) 90	(D63) 75	(D68) 66	(D73) 79
	5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

补充说明：

1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. m_1 、 m_2 或 n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	ZRST	P		D ₁ , D ₂	区域清除

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D ₁	●	●					●							
D ₂	●	●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D ₁	●	●	●	●	●	●	●	●	●	●		○	●				
D ₂	●	●	●	●	●	●	●	●	●	●		○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



D₁ : 区域清除起始装置

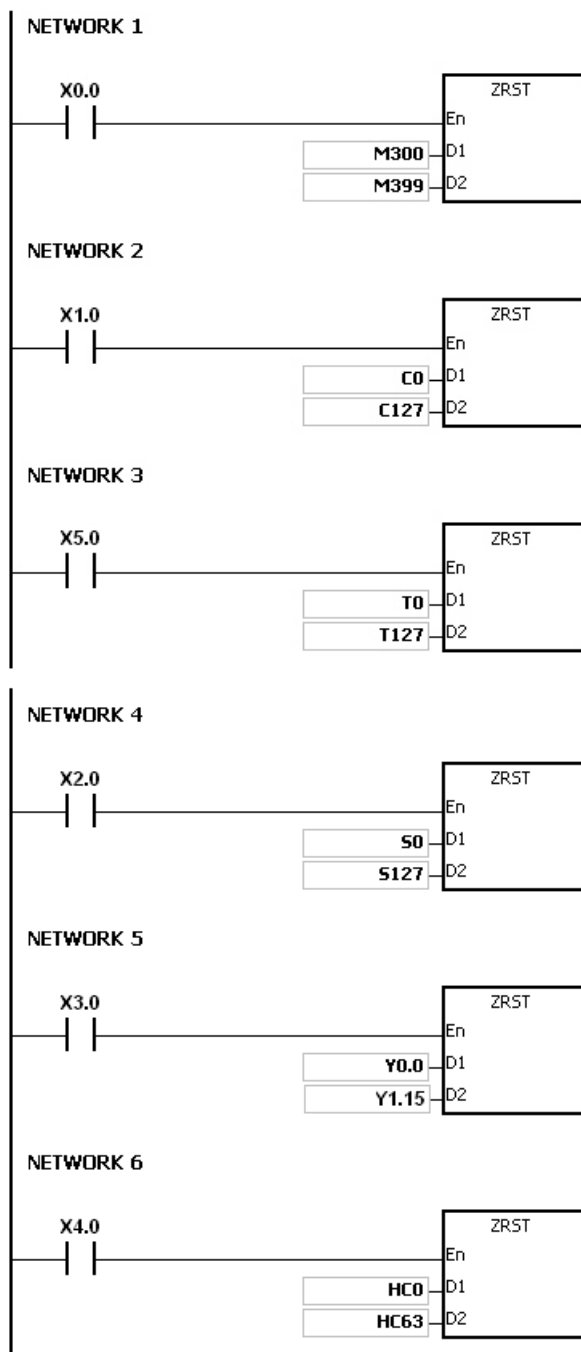
D₂ : 区域清除结束装置

指令说明：

1. 清除 D₁~D₂的操作数内容值。
2. 当 D₁操作数编号 > D₂操作数编号时，只有 D₂指定之操作数被清除。

程序范例：

1. 当 X0.0 为 ON 时，辅助继电器 M300~M399 被清除成 OFF。
2. 当 X1.0 为 ON 时，16 位计数器 C0~C127 全部清除。（写入 0，并将接点及线圈清除成 OFF）。
3. 当 X5.0 为 ON 时，定时器 T0~T127 全部清除。（写入 0，并将接点及线圈清除成 OFF）。
4. 当 X2.0 为 ON 时，步进点 S0~S127 被清除成 OFF。
5. 当 X3.0 为 ON 时，数据寄存器 Y0.0~Y1.15 被清除成 OFF。
6. 当 X4.0 为 ON 时，32 位计数器 HC0~HC63 全部清除。（写入 0，并将接点及线圈清除成 OFF）。



补充说明：

1. D_1 、 D_2 操作数装置类别不相同，指令不执行，SM0=ON，错误码 SR0=16#2007。
2. D_1 、 D_2 操作数数据格式不相同，指令不执行，SM0=ON，错误码 SR0=16#2007。

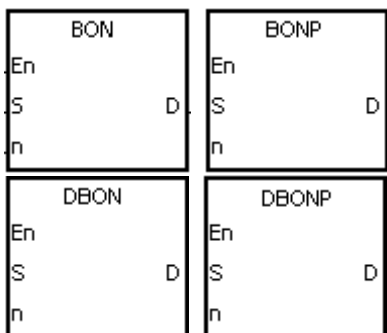
FB/FC	指令码			操作数	功能
FC	D*	BON	P	S, D, n	ON 位判定

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D	●/●*													
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●	●	●	●	●		●	●	●			●				
n	●	●			●	●	●	●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



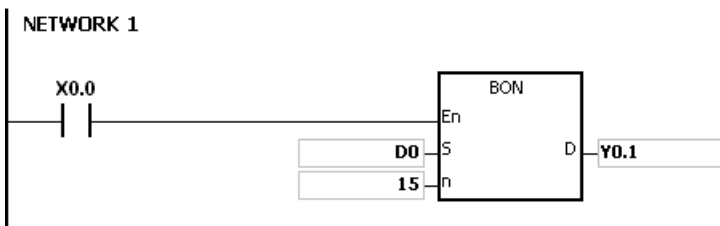
S : 来源装置
 D : 存放判定结果的装置
 n : 指定判定的位 (自 0 开始编号)

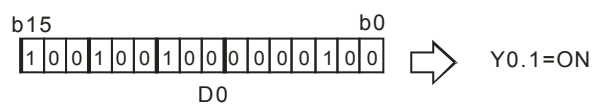
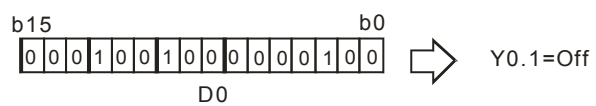
指令说明：

1. 判断 S 内容值的第 n 个位的状态，并将结果存放到 D。
2. 16 位指令 n=0~15，32 位指令 n=0~31。
3. 32 位指令才可以使用 HC 装置。

程序范例：

1. 当 X0.0=ON 时，若是 D0 的第 15 个位为“1”时，Y0.1=ON，为“0”时，Y0.1=OFF。
2. X0.0 变成 OFF 时，Y0.1 仍保持之前的状态。





补充说明：

n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

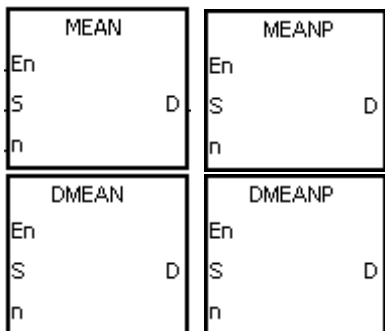
FB/FC	指令码			操作数	功能
FC	D*	MEAN	P	S, D, n	平均值

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						
n		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●		●	○	●				
n	●	●			●	●	●	●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



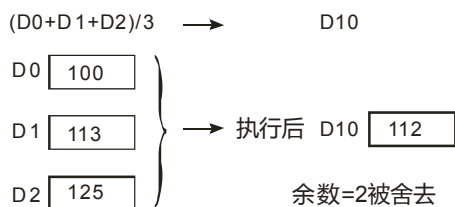
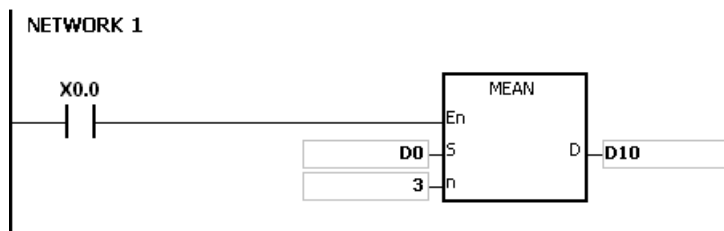
S : 欲取平均值的起始装置
 D : 存放平均值的装置
 n : 取平均值的装置个数

指令说明：

1. 将 S 起始的 n 个装置内容值相加后取平均值存入 D 中。
2. 如果计算中出现余数时，余数会被舍去。
3. 16 位指令中，n 操作数的内容值为 1~256；32 位指令中，n 操作数的内容值为 1~128。
4. 32 位指令才可以使用 HC 装置。

程序范例：

当 X0.0=ON 时，D0 开始算的 3 个 (n=3) 寄存器的内容全部相加，相加之后再除以 3 以求得平均值并存于指定的 D10 当中，余数被舍去。



补充说明：

1. 16 位指令中， $n < 1$ 或 $n > 256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. 32 位指令中， $n < 1$ 或 $n > 128$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 若 S+n-1 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。

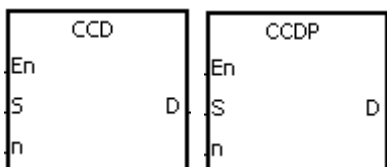
FB/FC	指令码			操作数			功能					
FC		CCD	P	S, D, n			总和检查					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



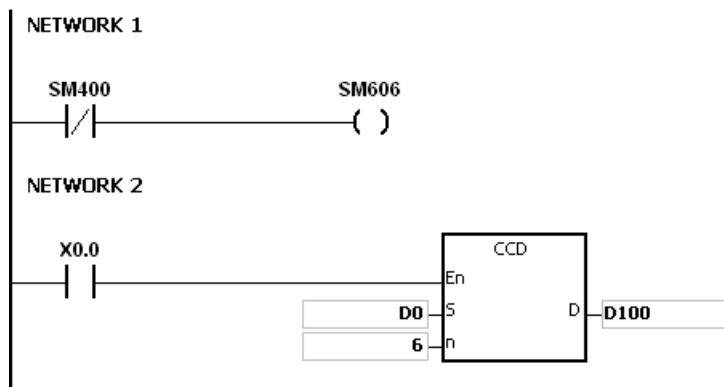
S : 数据源起始装置
D : 存放总和检查的结果
n : 数据个数

指令说明：

1. 本指令用来作通信时，为了确保数据传输时的正确性所做的字符串总和检查（Sum Check）。
2. 16 位转换模式：当 SM606=OFF 时，指定为 16 位转换模式。将 S 所指定寄存器起始号码开始算的 n 个数据（以 8 位为单位）内容作加总，加总结果存放于 D 所指定的寄存器当中，而极性位存放于 D+1 当中。
3. 8 位转换模式：当 SM606=ON 时，指定为 8 位转换模式。将 S 所指定寄存器起始号码开始算的 n 个数据（以 8 位为单位，只有下 8 位有效）内容作加总，加总结果存放于 D 所指定的寄存器当中，而极性位存放于 D+1 当中。
4. n=1~256。

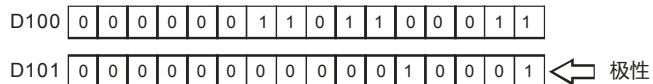
程序范例一：

1. 当 SM606=OFF 时，指定为 16 位转换模式。
2. 当 X0.0=ON 时，将 D0 所指定寄存器起始号码开始算的 6 个数据（以 8 位为单位 n=6 代表指定 D0~D2）内容作加总，加总结果存放于 D100 所指定的寄存器当中，而极性位存放于 D101 当中。



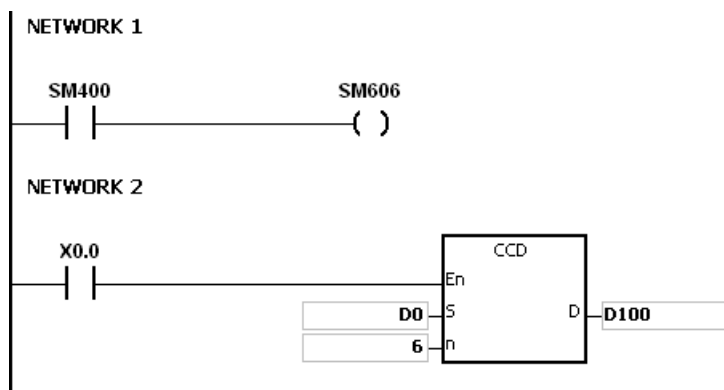
S	数据的内容
D0 下	100 = 0 1 1 0 0 1 0 0
D0 上	111 = 0 1 1 0 1 1 1 1 ←
D1 下	120 = 0 1 1 1 1 0 0 0
D1 上	202 = 1 1 0 0 1 0 1 0
D2 下	123 = 0 1 1 1 1 0 1 1 ←
D2 上	211 = 1 1 0 1 0 0 1 1 ←
D100	867 合计
D101	0 0 0 1 0 0 1 1 ←

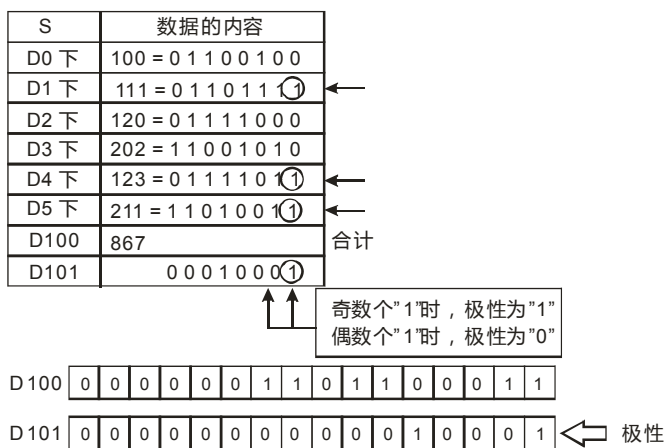
奇数个“1”时，极性为“1”
偶数个“1”时，极性为“0”



程序范例二：

1. 当 SM606=ON 时，指定为 8 位转换模式。
2. 当 X0.0=ON 时，将 D0 所指定寄存器起始号码开始算的 6 个数据（以 8 位为单位 n=6 代表指定 D0~D5）内容作加总，加总结果存放于 D100 所指定的寄存器当中，而极性位存放于 D101 当中。





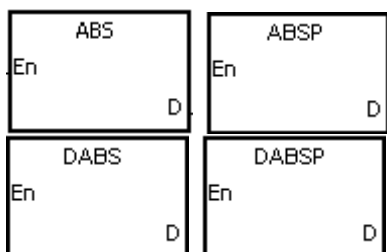
补充说明：

1. SM606=ON (8 位模式) 时 , S+n-1 装置超出范围时 , 指令不执行 , SM0=ON , 错误码 SR0=16#2003。
2. SM606=OFF(16 位模式) 时 , S+n/2-1 装置超出范围时 , 指令不执行 , SM0=ON , 错误码 SR0=16#2003。
3. n<1 或 n>256 时 , 指令不执行 , SM0= ON , 错误码 SR0=16#200B。
4. D 操作数 , 若使用 ISPSofT 宣告 , 则数据类型为 ARRAY [2] of WORD/INT。

3

FB/FC	指令码			操作数								功能					
FC	D*	ABS	P	D								绝对值					
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
D		●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●	●	●	●		●	○					
				脉冲执行型				16 位指令				32 位指令					
				AH 运动控制主机				AH 运动控制主机				AH 运动控制主机					

图形：



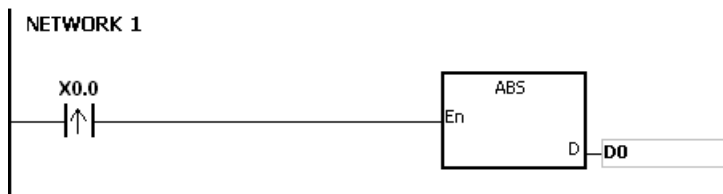
D：欲取绝对值的装置

指令说明：

1. 当 ABS 指令执行时，被指定的组件 D 取绝对值。
2. 本指令一般都是使用脉冲执行型指令（ABSP）。
3. 32 位指令才可以使用 HC 装置。

程序范例：

假设执行前 D0=-1234，当 X0.0=OFF→ON 时，D0 内容取绝对值，执行后 D0=1234。



FB/FC	指令码			操作数				功能						
FC		MINV	P	S, D, n				矩阵反相						

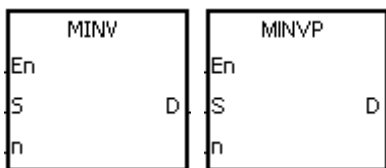
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●				●				
n	●	●			●	●		●	●		●		●	○	○		

3

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



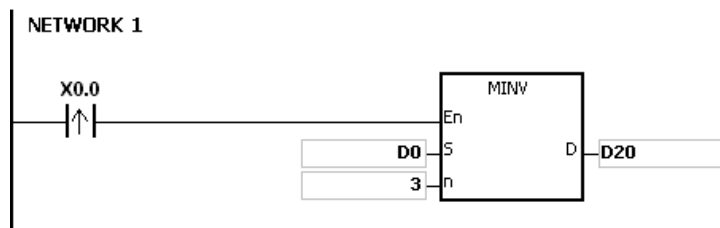
- S : 矩阵来源装置
- D : 运算结果
- n : 数组长度

指令说明：

1. 矩阵来源 S 依数组长度 n 作矩阵的反相运算并将结果存于 D。
2. n 操作数的范围值为 1~256。

程序范例：

当 X0.0=ON 时，16 位寄存器 D0~D2 共 3 列作 MINV，矩阵反相运算，将结果存于 16 位寄存器 D20~D22 共 3 列中。



	b15		b0															
D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

执行后 ↓

	b15		b0															
D20	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D21	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D22	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

补充说明：

1. $S+n-1$ 、 $D+n-1$ 装置超出范围时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
2. $n<1$ 或 $n>256$ 时，指令不执行， $SM0=ON$ ， $SR0$ 错误码= $16\#200B$ 。

FB/FC	指令码		操作数				功能				
FC		MBRD	P	S, D, n				矩阵位读出			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S : 矩阵来源装置
n : 数组长度
D : 指针 Pr, 用以存放目标的位置值

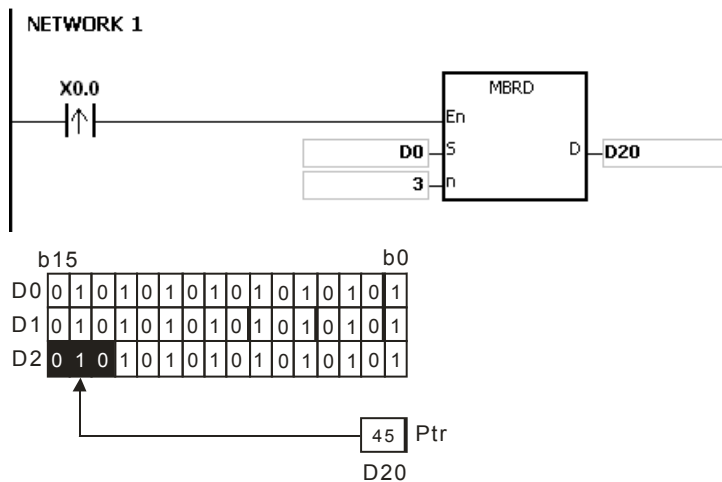
指令说明：

- 当指令执行时，一开始判断矩阵指针清除标志 SM613 是否为 ON，若为 ON，指针 **D** 清除为 0，再将指针 **D** 所指定的 bit 状态读取到矩阵循环位移输出进位标志 SM614，当读取完一个 bit 时，判断矩阵指针递增标志 SM612 是否 ON，若 ON 把指针 **D** 的值加 1，然后结束本指令之执行。
- 当读到最后一个 bit 时矩阵搜寻结束标志 SM608=ON，指针 **D** 记录着读取的 bit 的编号，然后结束本指令之执行。
- n** 操作数的范围值为 1~256。
- 矩阵之指针 **Pr** (Pointer)，由用户于指令中指定，其有效范围为 0 ~ 16n-1，分别对应至矩阵中的位 b0 ~ b16n-1。若 **Pr** 值超出此范围则矩阵指针错误标志 SM611 设为 1，且本指令不执行。

程序范例：

- 当 X0.0 由 OFF→ON 时，设指针清除标志 SM613=0、矩阵指针递增标志 SM612=1，所以每读取一次指针 **Pr** 增加 1。
- 设指针当前值 D20=45，当 X0.0 由 OFF→ON 时动作 3 次，可得到如 ❶，❷，❸ 三个执行结果。
 - ❶ D20=46，矩阵循环位移输出进位标志 SM614=0，矩阵搜寻结束标志 SM608=0。
 - ❷ D20=47，矩阵循环位移输出进位标志 SM614=1，矩阵搜寻结束标志 SM608=0。

③ D20=47，矩阵循环位移输出进位标志 SM614=0，矩阵搜寻结束标志 SM608=1。



补充说明：

1. $S+n-1$ 装置超出范围时，指令不执行，SM0=ON，SR0 错误码=16#2003。
2. $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，SR0 错误码=16#200B。
3. 标志信号说明：

SM608： 矩阵搜寻结束标志，当比较到最后一个 bit 时，SM608=1
 SM611： 矩阵指针错误标志，指针 Pr 值超出此范围则 SM611=1
 SM612： 矩阵指针递增标志，将指针目前值+1
 SM613： 矩阵指针清除标志，将指针目前值清除为 0
 SM614： 矩阵循环位移输出进位标志

FB/FC	指令码		操作数			功能				
FC		MBWR	P	S, D, n			矩阵位写入			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●				●				
D	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



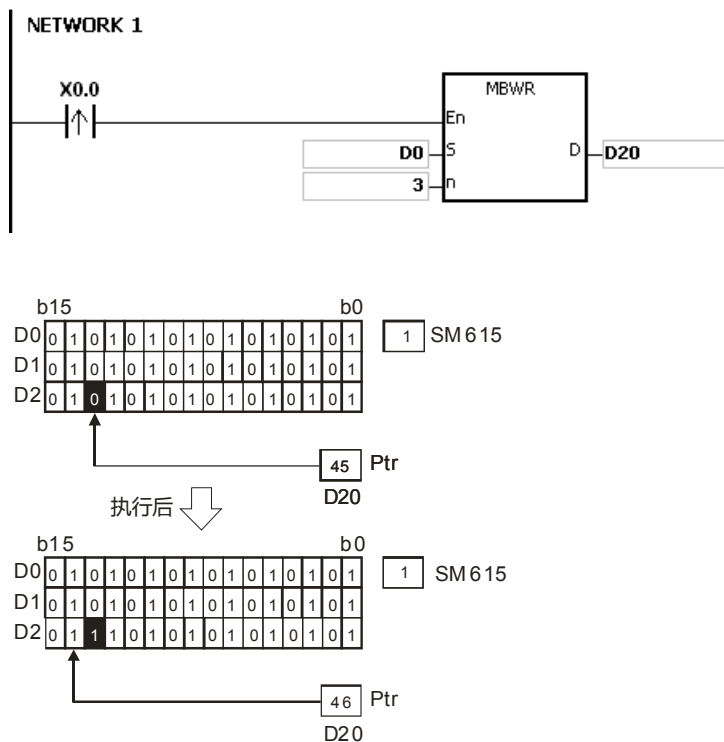
S : 矩阵来源装置
n : 数组长度
D : 指针 Pr, 用以存放目标之位置值

指令说明：

- 当指令执行时，一开始判断矩阵指针清除标志 SM613 是否为 ON，若为 ON，指针 **D** 清除为 0，把矩阵位移输入补位标志 SM615 的值，由指针 **D** 所指定的位写入 **S** 矩阵，当写完一个 bit 时，判断矩阵指针递增标志 SM612 是否 ON，若 ON 把指针 **D** 的值加 1，然后结束本指令之执行。
- 当写到最后一个 bit 时矩阵搜寻结束标志 SM608=ON，指针 **D** 记录着读取的 bit 的编号，然后结束本指令的执行。若 **D** 的值超过范围则 SM611=ON。
- n 操作数的范围值为 1~256。
- 矩阵之指针 Pr(Pointer)，由用户于指令中指定，其有效范围为 0 ~ 16n-1，分别对应至矩阵中之位 b0 ~ b16n-1。若 Pr 值超出此范围则矩阵指针错误标志 SM611 设为 1，且本指令不执行。

程序范例：

- 当 X0.0 由 OFF→ON 时，设指针清除标志 SM613=0、矩阵指针递增标志 SM612=1，所以每写入一次指针 Pr 增加 1。
- 设指针当前值 D20=45，矩阵位移输入补位标志 SM615 状态为 1 当 X0.0 由 OFF→ON 时动作 1 次，可得到如下执行结果。D20=45，矩阵位移输入补位标志 SM615=1，矩阵搜寻结束标志 SM608=0。

**补充说明：**

1. $S+n-1$ 装置超出范围时，指令不执行，SM0=ON，SR0 错误码=16#2003。
2. $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，SR0 错误码=16#200B。
3. 标志信号说明：

SM608： 矩阵搜寻结束标志，当比较到最后一个 bit 时，SM608=1

SM611： 矩阵指针错误标志，指针 Pr 值超出此范围则 SM611=1

SM612： 矩阵指针递增标志

SM613： 矩阵指针清除标志

SM615： 矩阵位移输入补位标志

FB/FC	指令码		操作数				功能				
FC		MBC	P	S, D, n				矩阵位状态计数			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



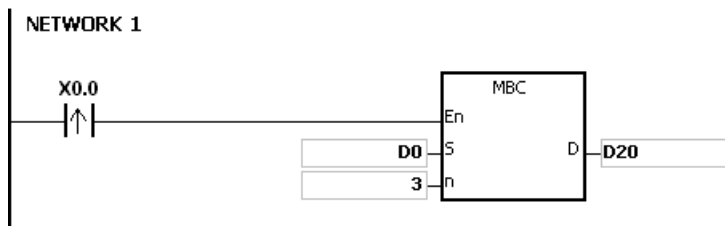
S : 矩阵来源装置
n : 数组长度
D : 运算结果

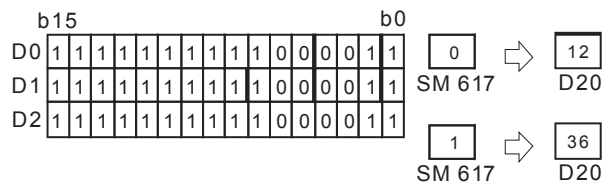
指令说明：

1. 依数组长度 **n** 计算 **S** 矩阵中所有位为 1 或为 0 的个数，并将数目存于 **D** 中。
2. 当 SM617=1 时计算矩阵位为 1 的个数，SM617=0 时，计算矩阵位为 0 的个数。当计算出来的结果为 0 时，SM618=1。
3. **n** 操作数的范围值为 1~256。

程序范例：

当 X0.0 为 ON 时，D0~D2 的矩阵中，当 SM617=1 时计算矩阵位为 1 的位总数被存于 D20 当中。当 SM617=0 时计算矩阵位为 0 的位总数被存于 D20 当中。





补充说明：

1. **S+n-1** 装置超出范围时，指令不执行，SM0=ON，SR0 错误码=16#2003。
2. **n<1** 或 **n>256** 时，指令不执行，SM0=ON，SR0 错误码=16#200B。
3. 标志信号说明：

- SM617： 矩阵计数字元为 0 或位为 1 标志
- SM618： 矩阵计数结果为 0 时 ON

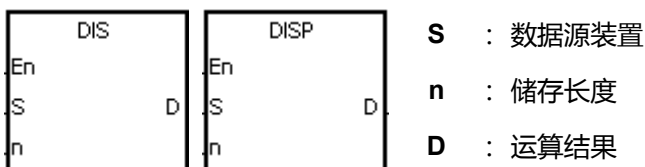
FB/FC	指令码		操作数			功能				
FC		DIS	P	S, D, n			16 位数据的 4 位分组			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●				●				
n	●	●			●	●		●	●		●	○	●	○	○		

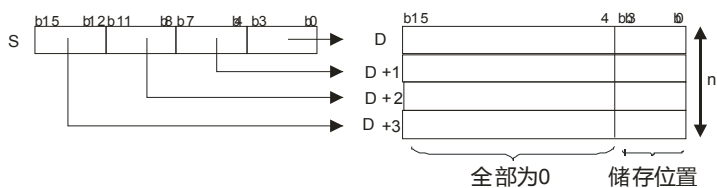
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

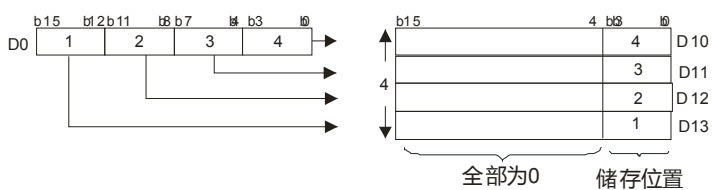
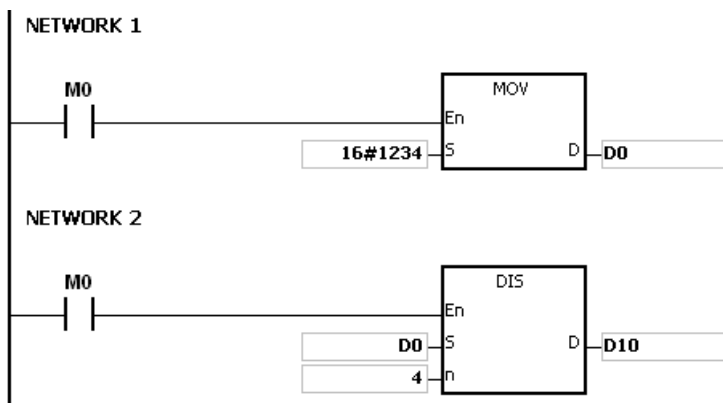
1. 将 **S** 所指定寄存器内容值，从 b_0 开始以 4 个位为 1 组，分别存放于 $D \sim D+(n-1)$ 所指定的寄存器的低 4 位中 ($b_0 \sim b_3$)。



2. $n=1 \sim 4$ 。

程序范例：

D0 的内容为 16#1234。启动 M0，执行 DIS 指令时，将 D0 的 $b_0 \sim b_{15}$ 内容分为 4 个 Bit 一组，分别移动至 D10~D13 的 $b_0 \sim b_3$ 当中。



补充说明：

1. $D \sim D + (n - 1)$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n < 1$ 或 $n > 4$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

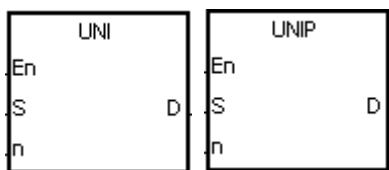
FB/FC	指令码		操作数				功能				
FC		UNI	P	S, D, n				16 位数据的 4 位链结			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

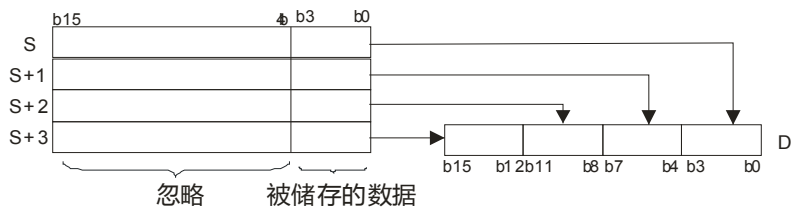
图形：



S : 数据源装置
n : 数据源长度
D : 运算结果

指令说明：

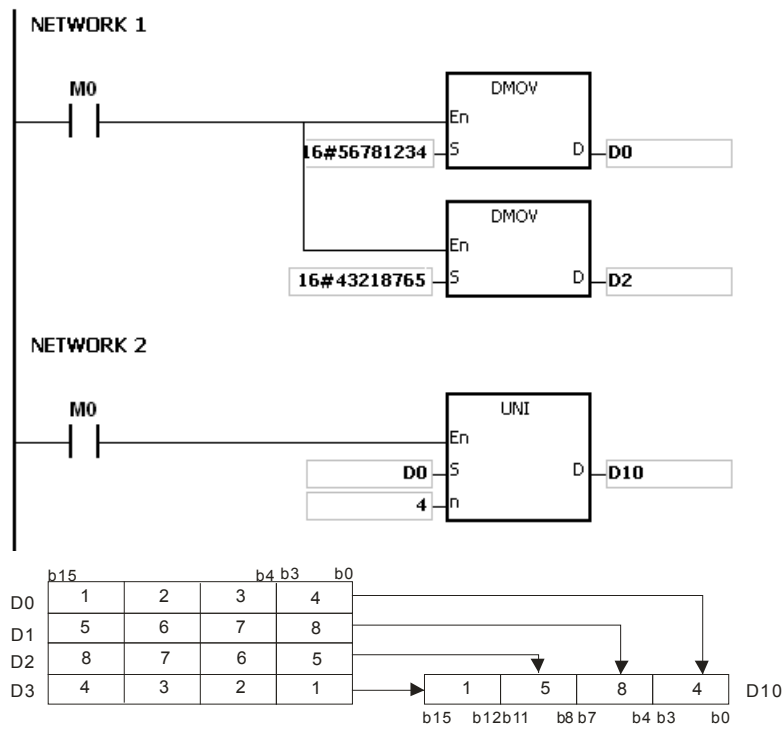
1. 将 **S~S+ (n-1)** 所指定寄存器内容值 ($b_0 \sim b_3$)，依序存放于 **D** ($b_0 \sim b_{15}$) 所指定的寄存器中。



2. $n=1 \sim 4$ 。

程序范例：

D0~D3 的内容为分别为 16#1234、16#5678、16#8765、16#4321。启动 M0，执行 UNI 指令时，将 D0~D3 的 $b_0 \sim b_3$ 内容，分别储存至 D10 的 $b_0 \sim b_{15}$ 当中。



补充说明：

1. $S \sim S + (n - 1)$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n < 1$ 或 $n > 4$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

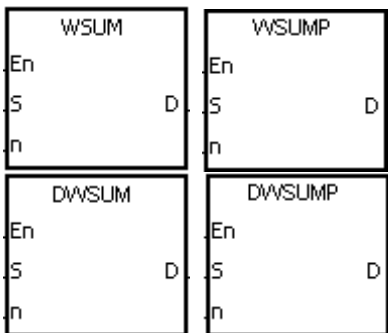
FB/FC	指令码			操作数	功能
FC	D*	WSUM	P	S, D, n	求和

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						
n			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				
D	●	●			●	●	●	●	●		●		●				
n	●	●			●	●	●	●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

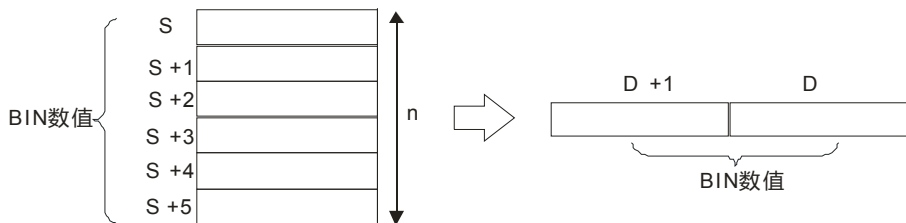
图形：

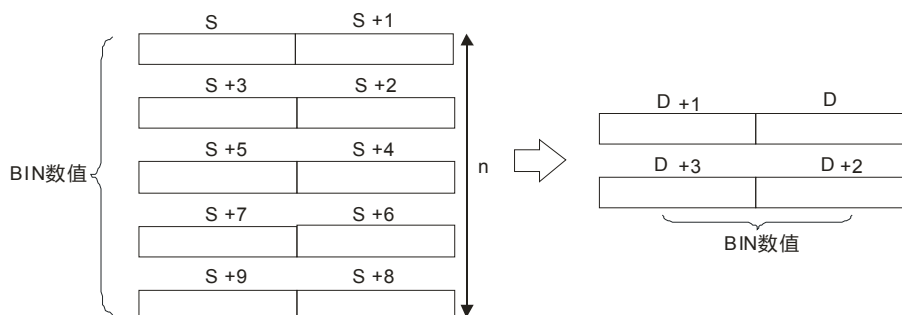


S : 数据源装置
 n : 数据源长度
 D : 运算结果

指令说明：

1. 将 S~S+n-1 所指定寄存器内容值，以有号数十进制作相加，存放于 D 所指定的寄存器中。

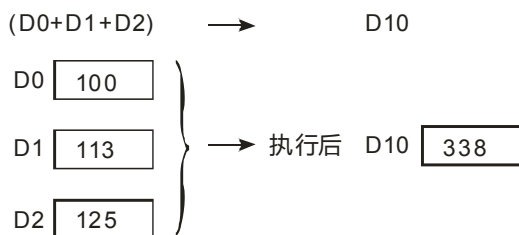
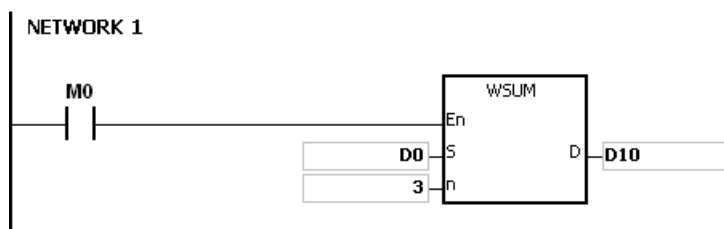




- 16 位指令中， n 操作数的内容值为 1~256；32 位指令中， n 操作数的内容值为 1~128。
- 32 位指令才可以使用 HC 装置。

程序范例：

当执行 WSUM 指令时，会将 D0~D2 的内容相加后，计算结果储存至 D10 当中。



补充说明：

- 16 位指令中， $n < 1$ 或 $n > 256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
- 32 位指令中， $n < 1$ 或 $n > 128$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
- 若 S+n-1 或 D 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码		操作数		功能
FC		BSET	P	S, D, n	字符装置位动作保持 ON

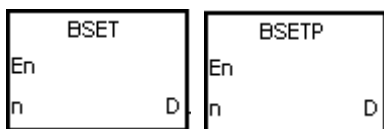
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



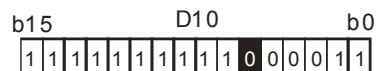
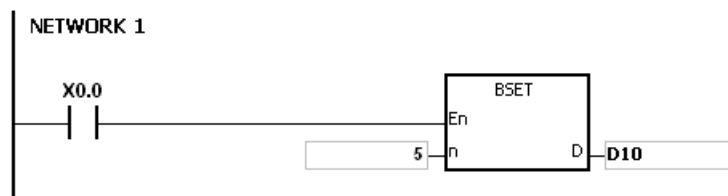
D : 装置编号
n : 要设定的 bit 号码

指令说明：

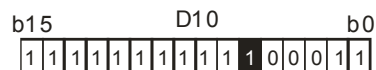
1. 将 D 所指定寄存器的第 n 个位设为 1。
2. 当 BSET 指令被驱动，其指定的位被设定为 ON，且被设定的位会维持 ON，不管 BSET 指令是否仍被驱动，可利用 BRST 指令将该位设为 OFF。
3. n=0~15。

程序范例：

当 X0.0=ON 时，D10 的第 5 个 Bit 会设定为 1。



执行后 ↓



补充说明：

n<0 或 n>15 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码		操作数		功能
FC		BRST	P	D, n	字符装置位清除

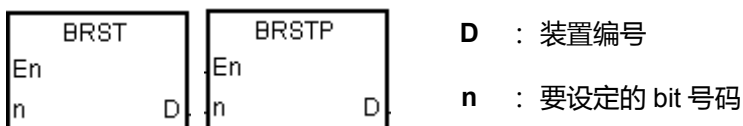
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

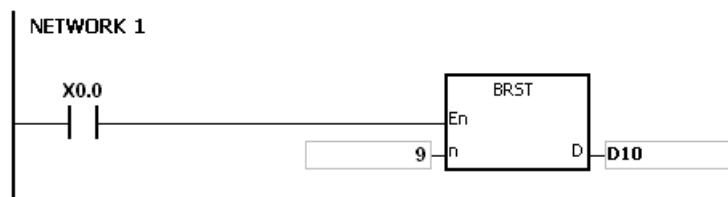


指令说明：

1. 将 D 所指定寄存器，的第 n 个位设为 0。
2. 当 BRST 指令被驱动，其指定的位会被设定为 OFF。
3. n=0~15。

程序范例：

当 X0.0=ON 时，D10 的第 9 个 bit 会清除为 0。



执行后 ↓



补充说明：

n<0 或 n>15 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

FB/FC	指令码			操作数	功能
FC	BKRST	P		D, n	指定区域清除

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●	●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●			○	●				
n	●	●	●	●	●	●	●	●	●	●	●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



D : 装置编号

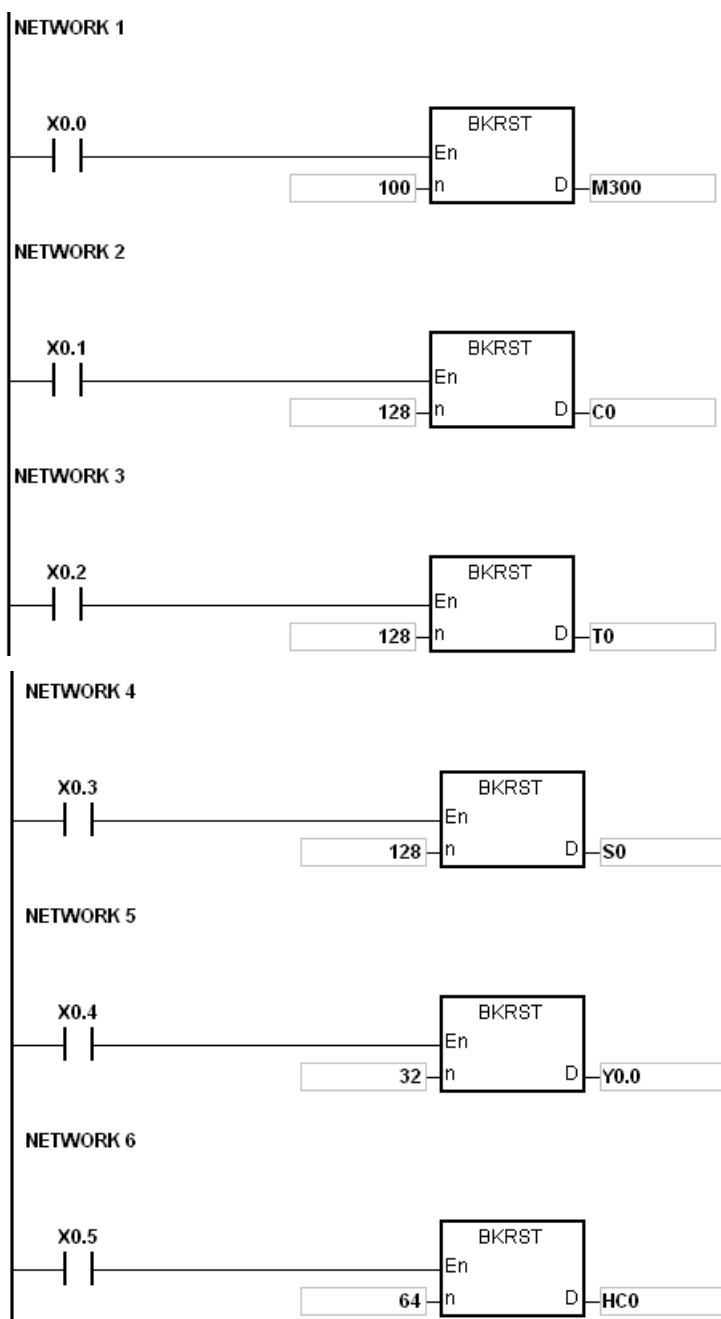
n : 长度

指令说明：

1. 清除 **D~D+ (n-1)** 的操作数内容值。
2. **n=1~1024**

程序范例：

1. 当 X0.0 为 ON 时，辅助继电器 M300~M399 被清除成 OFF。
2. 当 X0.1 为 ON 时，计数器 C0~C127 全部清除。（写入 0，并将接点及线圈清除成 OFF）。
3. 当 X0.2 为 ON 时，定时器 T0~T127 全部清除。（写入 0，并将接点及线圈清除成 OFF）。
4. 当 X0.3 为 ON 时，步进点 S0~S127 被清除成 OFF。
5. 当 X0.4 为 ON 时，数据寄存器 Y0.0~Y1.15 被清除成 OFF。
6. 当 X0.5 为 ON 时，计数器 HC0~HC63 全部清除。（写入 0，并将接点及线圈清除成 OFF）。

**补充说明：**

1. $D \sim D + (n - 1)$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n < 0$ 或 $n > 1024$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。

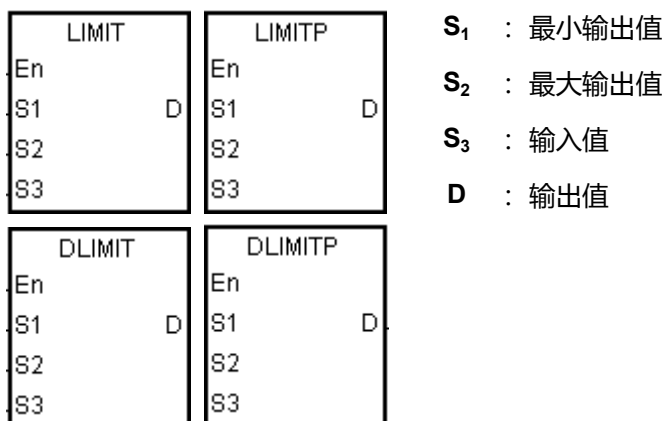
FB/FC	指令码			操作数	功能
FC	D*	LIMIT	P	S ₁ , S ₂ , S ₃ , D	高低限控制

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂ , S ₃		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂ , S ₃	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：

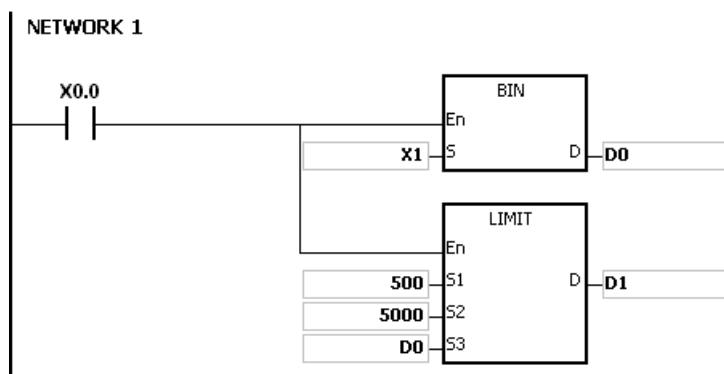


指令说明：

- 输入值 S₃ 与最小输出值 S₁ 及最大输出值 S₂ 作比较，比较后将结果存入输出值 D，说明如下：
 - 若最小输出值 S₁ > 输入值 S₃，输出值 D = 最小输出值 S₁。
 - 若最大输出值 S₂ < 输入值 S₃，输出值 D = 最大输出值 S₂。
 - 若最小输出值 S₁ ≤ 输入值 S₃ ≤ 最大输出值 S₂，输出值 D = 输入值 S₃。
 - 若最小输出值 S₁ > 最大输出值 S₂，输出值 D 不执行。
- 32 位指令才可以使用 HC 装置。

程序范例：

- 当 X0.0=ON 时，X1 的装置状态转为 BIN 值储存在 D0，D0 内的数值经 500~5000 范围限制计算后，将结果储存到 D1。



最小输出值	最大输出值	输入值 D0	功能	输出值 D1
500	5000	499	$D0 < 500$	500
		5001	$D0 > 5000$	5000
		600	$500 \leq D0 \leq 5000$	600

补充说明：

$S_1 > S_2$ 时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数	功能
FC	D*	BAND	P	S ₁ , S ₂ , S ₃ , D	死区控制

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂ , S ₃		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂ , S ₃	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

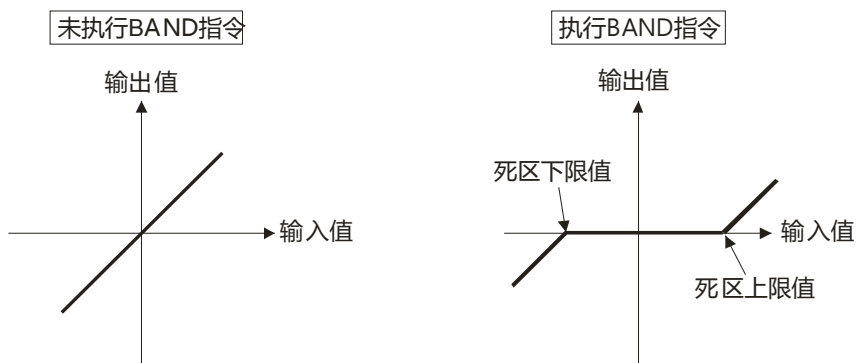
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



指令说明：

- 输入值 S₃ 与死区下限值 S₁ 及死区上限值 S₂ 作相减，相减后将结果存入 D。说明如下：
 - 若最小输出值 S₁ > 输入值 S₃，输出值 D = S₃ - S₁。
 - 若最大输出值 S₂ < 输入值 S₃，输出值 D = S₃ - S₂。
 - 若最小输出值 S₁ ≤ 输入值 S₃ ≤ 最大输出值 S₂，输出值 D = 0。
 - 若最小输出值 S₁ > 最大输出值 S₂，输出值 D 不执行。
- 32 位指令才可以使用 HC 装置。
- 图形说明如下：

4. S_1 、 S_2 和 S_3 的范围：

- BAND 指令：-32768~32767

当输出值 D 运算结果超过-32768 或 32767，范例说明如下：

死区下限值 $S_1=10$ ，输入值 $S_3=-32768$

输出值 $D=-32768-10=16\#8000-16\#000A=16\#7FF6=32758$

- DBAND 指令：-2147483648~2147483647

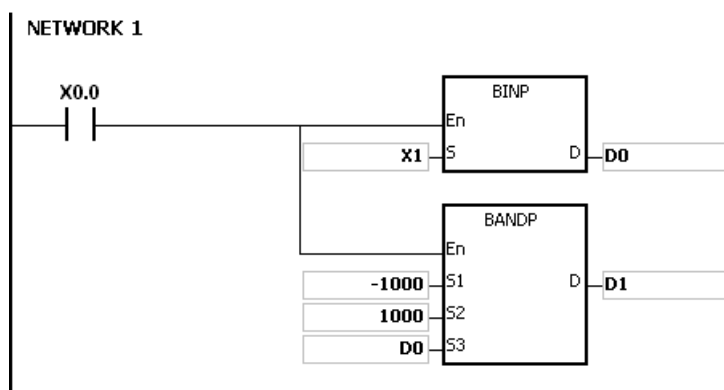
当输出值 D 运算结果超过-2147483648 或 2147483647，范例说明如下：

死区下限值 (S_1+1, S_1) =1000，输入值 (S_3+1, S_3) =-2147483648

输出值($D+1, D$)=-2147483648-1000=16#80000000-16#000003E8=16#7FFFFC18=2147482648

程序范例一：

当 X0.0 为 ON 时，X1 的 BCD 数值执行从-1000 至 1000 的范围控制，将控制结果储存至 D1 当中。

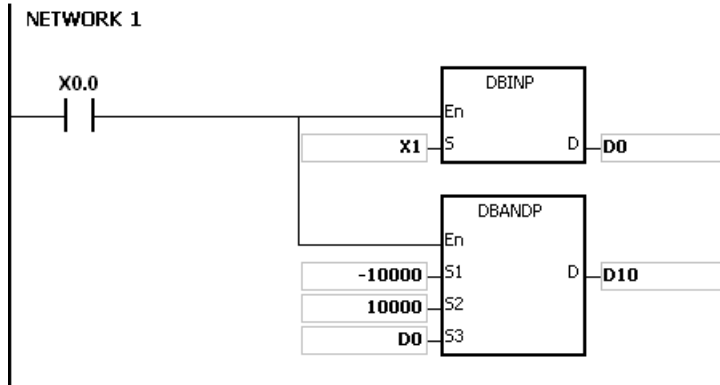


执行结果：

死区下限值	死区上限值	输入值 D0	功能	输出值 D1
-1000	1000	-1200	$D0 < -1000 \Rightarrow D1 = D0 - (-1000)$	-200
		1200	$D0 > 1000 \Rightarrow D1 = D0 - 1000$	200
		500	$-1000 \leq D0 \leq 1000 \Rightarrow D0 = 0$	0

程序范例二：

当 X0.0 为 ON 时，(X2 , X1) 的 BCD 数值执行从-10000 至 10000 的范围控制，将控制结果储存至 (D11 , D10) 当中。



执行结果：

3

死区下限值	死区上限值	输入值 (D1 , D0)	功能	输出值 (D11 , D10)
-10000	10000	-12000	(D1 , D0) < -10000 => (D11 , D10) = (D1 , D0) - (-10000)	-2000
		12000	(D1 , D0) > 10000 => (D11 , D10) = (D1 , D0) - 10000	2000
		5000	-10000 ≤ (D1 , D0) ≤ 10000 => (D1 , D0) = 0	0

补充说明：

S₁>S₂时，指令不执行，SM0=ON，错误码 SR0=16#2003。

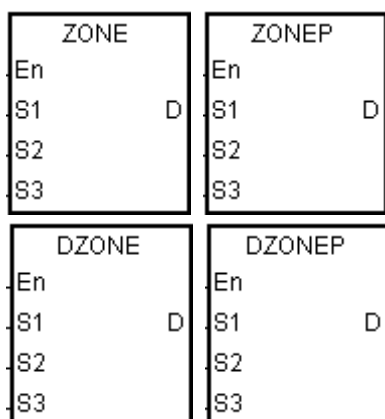
FB/FC	指令码			操作数	功能
FC	D*	ZONE	P	S ₁ , S ₂ , S ₃ , D	区域控制

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂ , S ₃		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁ , S ₂ , S ₃	●	●			●	●	●	●	●			○	●	○	○		
D	●	●			●	●	●	●	●			○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



S₁ : 负偏差值

S₂ : 正偏差值

S₃ : 输入值

D : 输出值

指令说明：

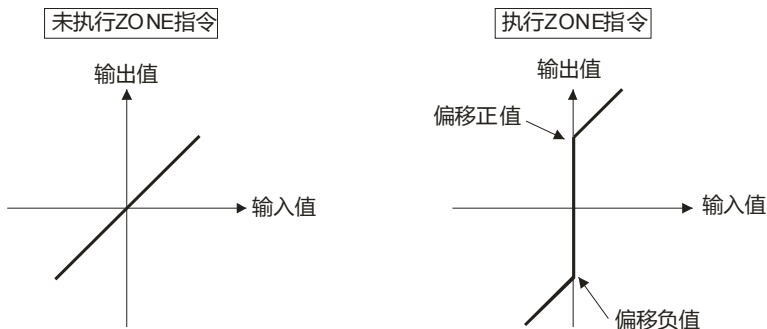
1. 输入值 S₃ 与负偏差值 S₁ 或正偏差值 S₂ 以作相加，并将结果存入 D。说明如下：

若输入值 S₃ < 0，输出值 D = S₃ + S₁。

若输入值 S₃ > 0，输出值 D = S₃ + S₂。

若输入值 S₃ = 0，输出值 D = 0。

2. 图形说明如下：



3. 32 位指令才可以使用 HC 装置。

4. S_1 、 S_2 和 S_3 的范围：

- ZONE 指令：-32768~32767

当输出值 D 运算结果超过-32768 或 32767，范例说明如下：

偏差负值 $S_1=-100$ ，输入值 $S_3=-32768$

输出值 $D = (-32768) + (-100) = 16\#8000 + 16\#FF9C = 16\#7F9C = 32668$

- DZONE 指令：-2147483648~2147483647

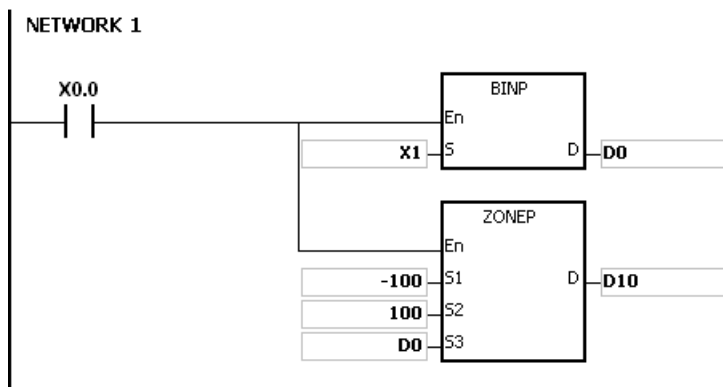
当输出值 D 运算结果超过-2147483648 或 2147483647，范例说明如下：

偏差负值 (S_1+1, S_1) = -1000，输入值 (S_3+1, S_3) = -2147483648

输出值 ($D+1, D$) = -2147483648 + (-1000)
 = $16\#80000000 + 16\#FFFFFFC18 = 16\#7FFFFFFC18 = 2147482648$

程序范例一：

当 X0.0 为 ON 时，X1 的 BCD 数值执行从-100 至 100 的偏差值相加，将偏移结果储存至 D10 当中。

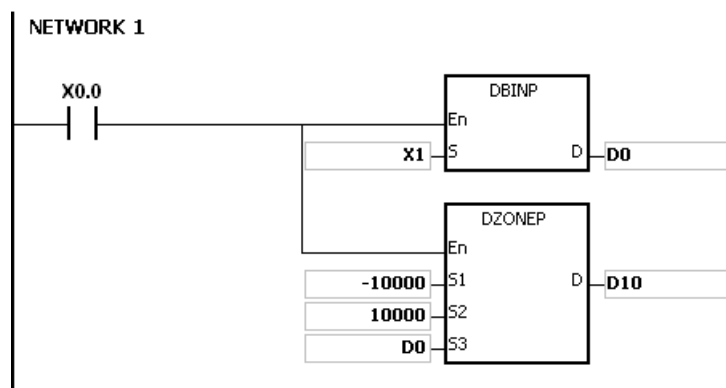


执行结果：

负偏差值	正偏差值	输入值 D0	功能	输出值 D10
-100	100	-10	$D0 < 0 \Rightarrow D10 = (-10) + (-100)$	-110
		0	$D0 = 0 \Rightarrow D10 = 0$	0
		50	$D0 > 0 \Rightarrow D10 = 50 + 100$	150

程序范例二：

当 X0.0 为 ON 时，(X2 , X1) 的 BCD 数值执行从 -10000 至 10000 的偏差值相加，将偏移结果储存至 (D11 , D10) 当中。



负偏差值	正偏差值	输入值 (D1 , D0)	功能	输出值 (D11 , D10)
-10000	10000	-10	$(D1 , D0) < 0$ $\Rightarrow (D11 , D10)$ $= (-10) + (-10000)$	-10010
		0	$(D1 , D0) = 0$ $\Rightarrow (D11 , D10) = 0$	0
		50	$(D1 , D0) > 0$ $\Rightarrow (D11 , D10) = 50 + 10000$	10050

3.16 结构建立指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>FOR</u>	-	-	循环回路起始	3
FC	<u>NEXT</u>	-	-	循环回路结束	1
FC	<u>BREAK</u>	-	✓	强制结束 FOR-NEXT 跳转	5

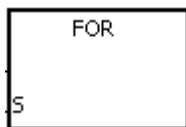
FB/FC	指令码			操作数				功能					
FC		FOR		S				循环回路起始					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

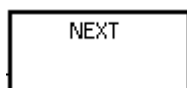
图形：



S : 回路重复执行的次数

FB/FC	指令码		操作数	功能
FC		NEXT	-	循环回路结束
			脉冲执行型	16 位指令
			-	AH 运动控制主机
				32 位指令
				-

图形：

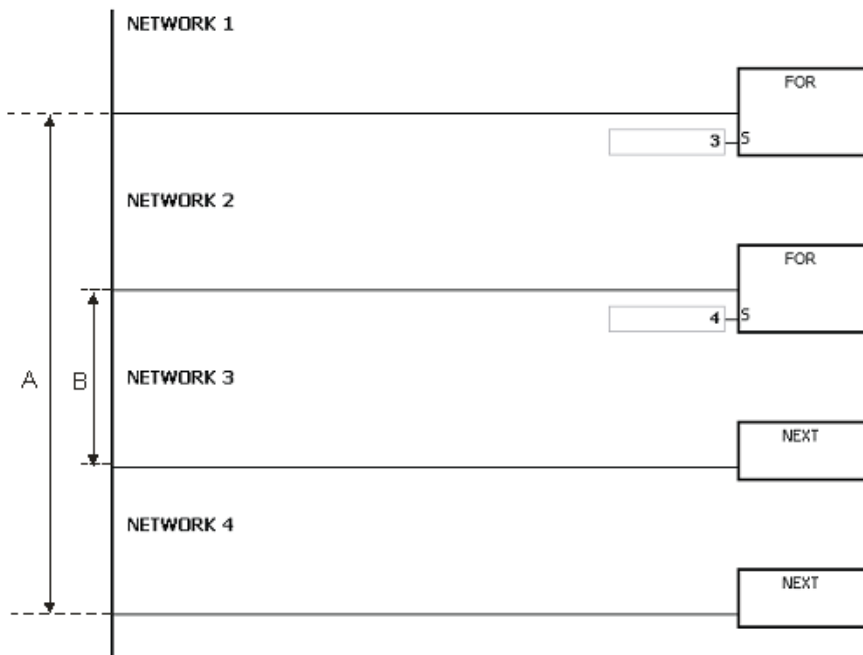


指令说明：

1. 由 FOR 指令指定 FOR~NEXT 跳转来回执行 N 次后跳出 FOR~NEXT 跳转往下继续执行。
2. 指定次数范围 $N=1\sim 32,767$ ，当指定次数范围 $N\leq 1$ 时，都视为是 1。
3. 当不执行 FOR~NEXT 回路时，可使用 CJ 指令来跳出回路。
4. 下列情形会产生错误：
 - NEXT 指令在 FOR 指令之前。
 - 有 FOR 指令没有 NEXT 指令。
 - FEND 或 END 指令之后有 NEXT 指令时。
 - FOR~NEXT 指令个数不同时。
5. 循环式 FOR~NEXT 回路最多可使用 32 层，但要注意回路次数过多时，会使 PLC 扫描时间增加有可能造成逾时监视定时器动作，而导致错误产生。可使用 WDT 指令来改善。

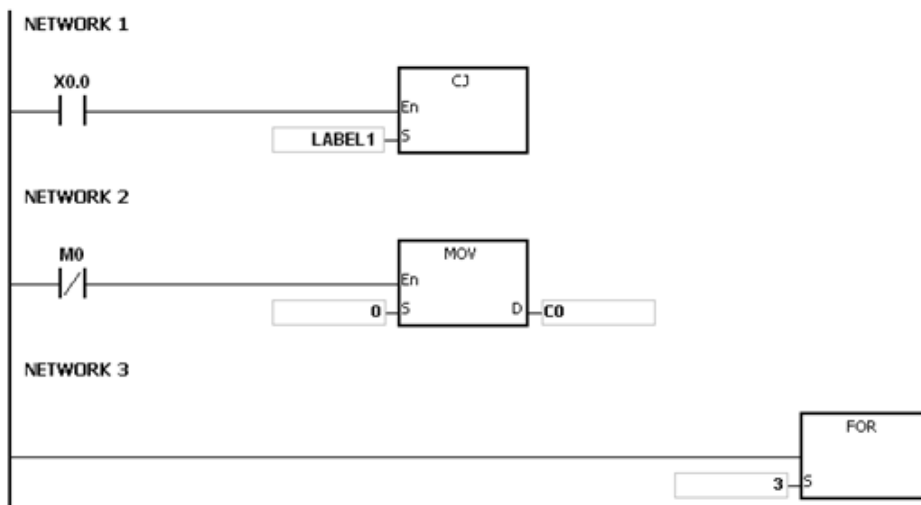
程序范例一：

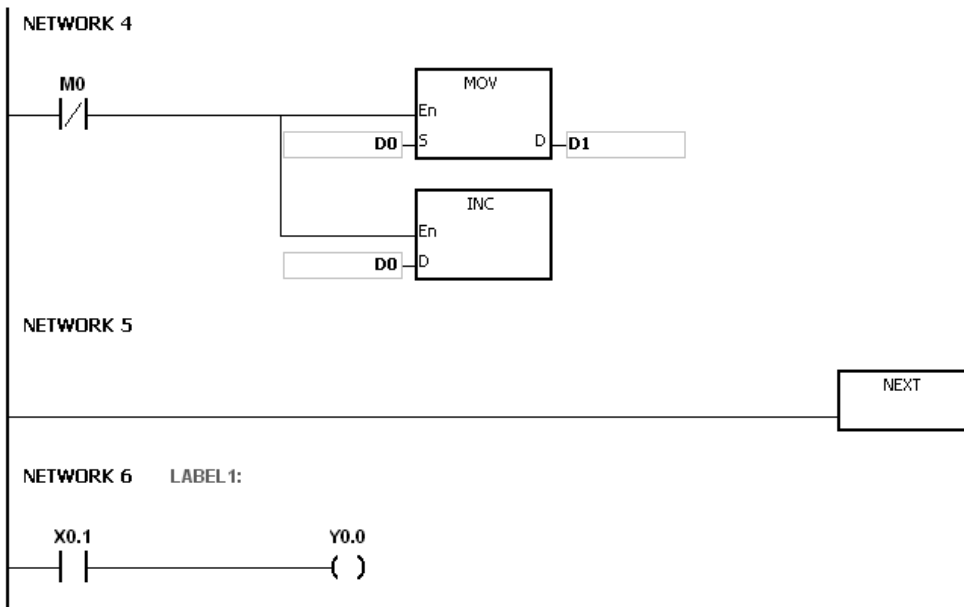
A 程序执行 3 次后在到 NEXT 指令以后的程序继续执行。而 A 程序每执行一次 B 程序会执行四次，所以 B 程序合计共执行 $3\times 4 = 12$ 次。



程序范例二：

当 X0.0=OFF 时，PLC 会执行 FOR~NEXT 之间的程序，当 X0.0=ON 时，CJ 指令执行跳转至 LABEL 1：即 NETWORK 6 处，FOR~NEXT 之间的程序 NETWORK 4~NETWORK 5 跳过不执行。

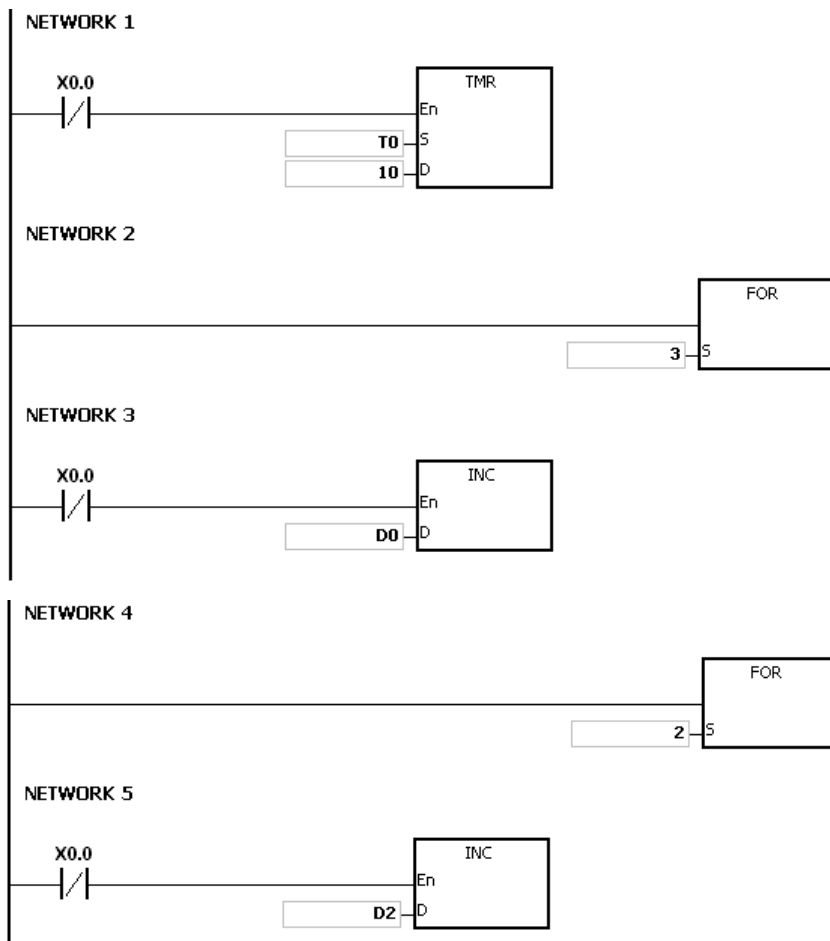


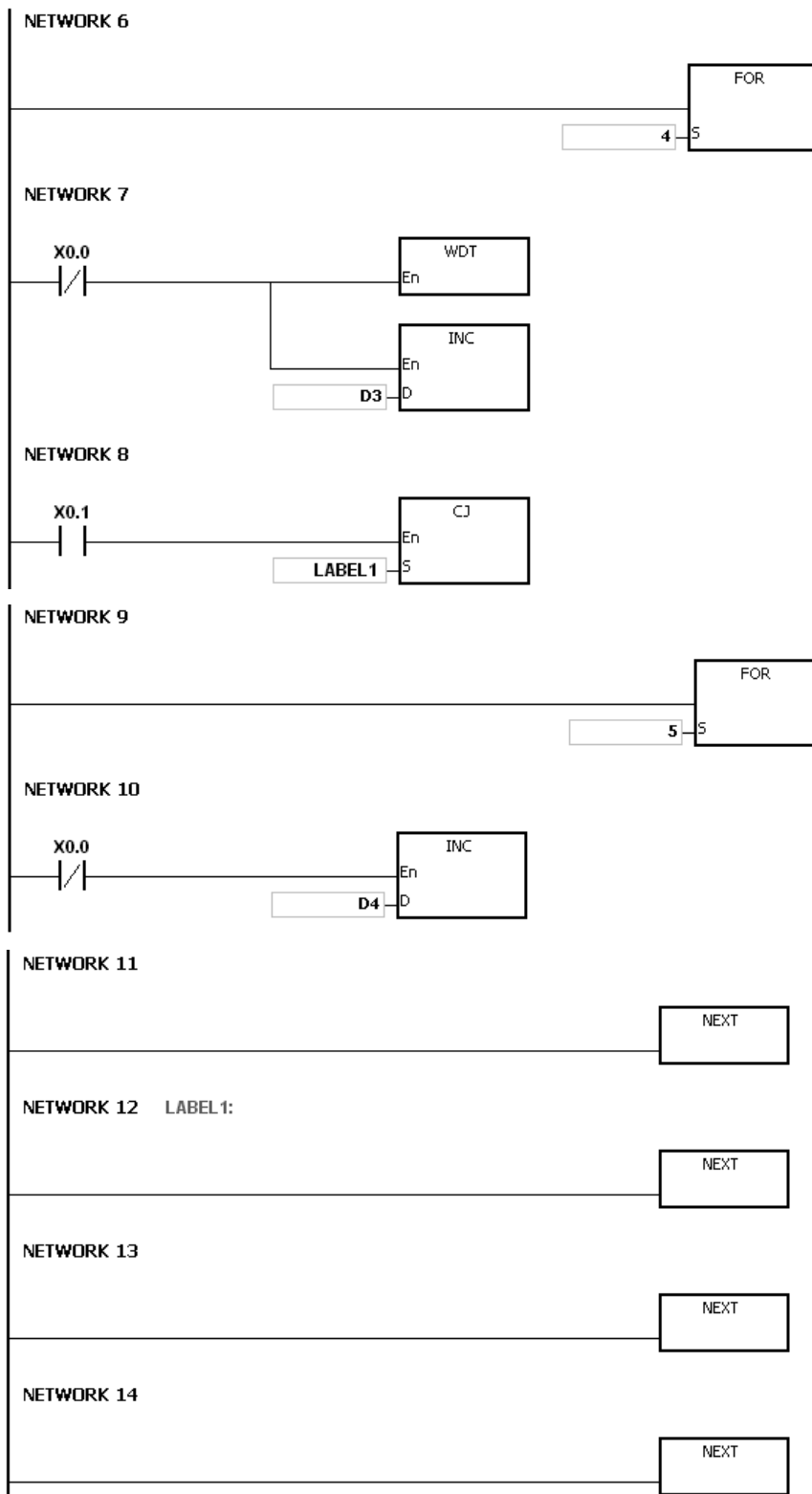


3

程序范例三：

当不执行 FOR~NEXT 时,可使用 CJ 指令来跳转。NETWORK 8 的 X0.1=ON 时,CJ 指令执行跳转至 Label 1: 即 NETWORK 12 处, NETWORK 9~NETWORK 11 跳过不执行。



**补充说明：**

指针 P 在 ISPSOft 中的使用方式，请参考 **ISPSOft 使用手册** 说明。

FB/FC	指令码		操作数	功能
FC	BREAK	P	D, P	强制结束 FOR-NEXT 跳转

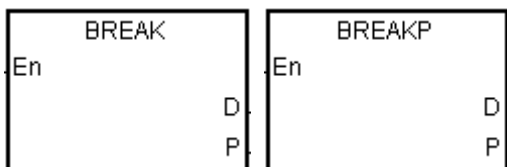
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							
P														

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
P																	

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：



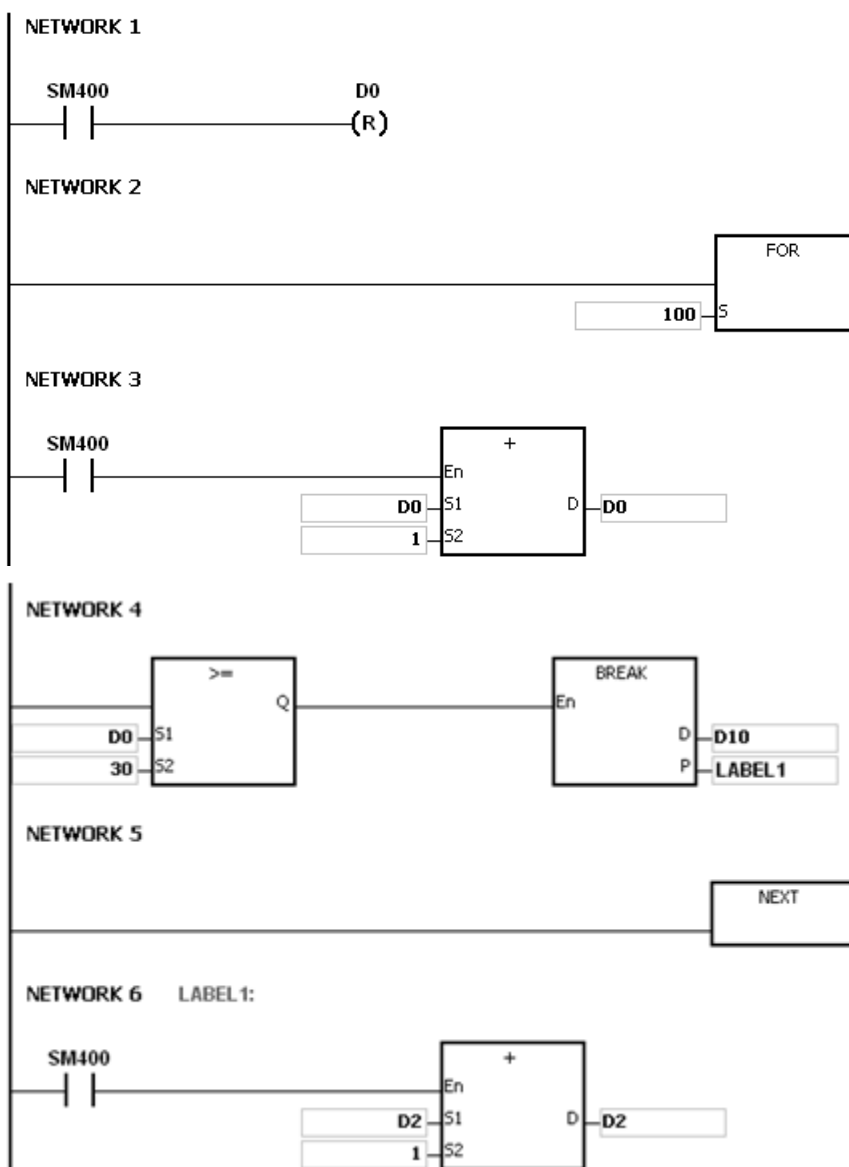
D : 储存剩余未执行完的跳转数
P : 强制结束跳转后要执行的指针

指令说明：

1. 强制结束 FOR-NEXT 的跳转，将尚未执行完的跳转数存放到 **D**，并跳至 **P** 所指定的指针开始执行。
2. BREAK 指令执行时，将 FOR-NEXT 剩余尚未执行的跳转数存放到 **D** 中，包含 BREAK 指令正在执行时的这一次。

程序范例：

程序开始执行 100 次 FOR-NEXT 跳转的程序 D0 内容加 1，当 D0 内容等于 30 时，将强制结束 FOR-NEXT 跳转，将剩余的跳转数 71 储存至 D10，并跳至 LABEL1（即 NETWORK 6 处）执行 D2 内容加 1。



补充说明：

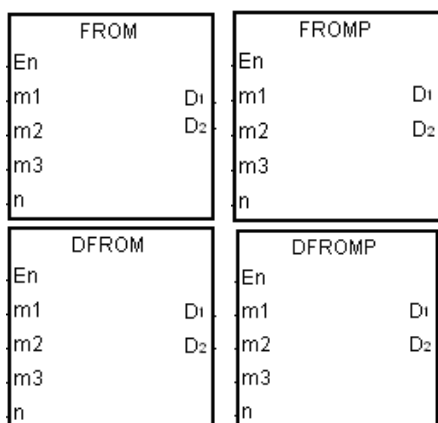
1. BREAK 指令所指定的 P 指针不存在，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2004。
2. BREAK 指令写在 FOR-NEXT 之外，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2017。
3. 指针 P 在 ISPSOft 中的使用方式，请参考 **ISPSOft 使用手册**说明。

3.17 模块的数据读写指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>FROM</u>	<u>DFROM</u>	✓	特殊模块 CR 数据读出	13
FC	<u>TO</u>	<u>DTO</u>	✓	特殊模块 CR 数据写入	13

FB/FC	指令码			操作数						功能							
FC	D*	FROM	P	$m_1, m_2, m_3, D_1, D_2, n$						特殊模块 CR 数据读出							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
m_1, m_2, m_3		●	●*				●	●*									
D_1, D_2		●	●*				●	●*									
n		●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
m_1, m_2, m_3	●	●			●	●	●	●	●				●	○	○		
D_1, D_2	●	●			●	●	●	●	●				●				
n	●	●			●	●	●	●	●				●	○	○		
						脉冲执行型			16 位指令				32 位指令				
						AH 运动控制主机			AH 运动控制主机				AH 运动控制主机				

图形：

 m_1 : Rack 代码 m_2 : Slot 代码 m_3 : 欲读取特殊模块的 CR (Controlled Register) 编号 D_1 : 存放读取数据的位置 D_2 : 存放错误代码的位置

n : 一次读取之数据笔数

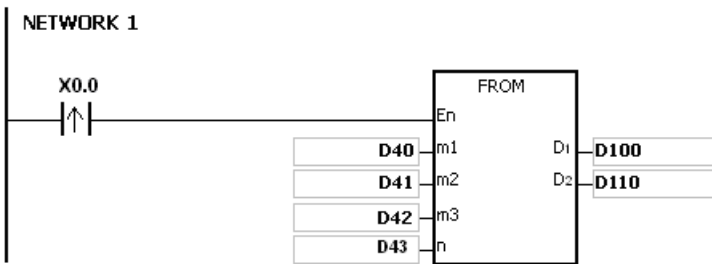
指令说明：

1. AH500 系列 PLC 可利用此指令读取特殊模块之 CR 数据。
2. m_1 : 背板 Rack 代码, $m_1=1\sim 8$; 其中 $m_1=1$ 代表主背板, $m_1=2\sim 8$ 代表扩展背板。
3. m_2 : 插槽 Slot 代码, $m_2=0\sim 11$; 若 $m_1=1$ (主背板), 则 m_2 (插槽) = $0\sim 11$; 若 $m_1=2\sim 8$ (扩展背板), 则 $m_2=0\sim 7$ 。
4. m_3 : 欲读取特殊模块的 CR (Controlled Register) 编号
5. D_2 : 开始执行 FROM 时, D_2 会被设定为 0 (表示无错误), 当有错误时, D_2 为非 0。有关错误代码说明请参考补充说明。
6. n : 读取之数据笔数; 16 位指令中, $n=1\sim 256$; 32 位指令中, $n=1\sim 128$ 。

- 7. 32 位指令才可以使用 HC 装置。
- 8. 特殊模块所在之编号算法请参考 TO 指令中，指令操作数的规则说明。
- 9. 特殊模块包含 AIO、NIO 与 PIO。

程序范例：

当启动 X0.0=OFF→ON 时，会执行 FROM 的应用指令，读取置放于 CPU 模块右侧第一个特殊模块，AH-10SCM-A5 的 COM1 数据交换读取触发的运作模式（CR#7），并且将回复的 CR#7 储存到 D100，因为执行无误所以 D110=16#0000。



各参数使用说明如下：

- 模块位于主背板，因此背板（Rack）编号 D40 设定为 16#0001。
- 模块放置在第一个插槽，因此插槽（Slot）编号 D41 设定为 16#0000。
- 模块之 COM1 数据交换读取触发设定值为 CR#7，因此 CR 编号 D42 设定为 16#0007。
- 模块之 COM1 数据交换读取触发设定值只占用一个寄存器，因此读取笔数 D43 设定为 1。
- 模块回复 CR#7 的数据会储存在 D100 里。

补充说明：

1. m₁ 与 m₂ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 D1~D1+n-1 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. n 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 由于使用 FROM 会降低 CPU 模块与 I/O 模块的执行效能，所以不建议使用。
5. 错误代码说明

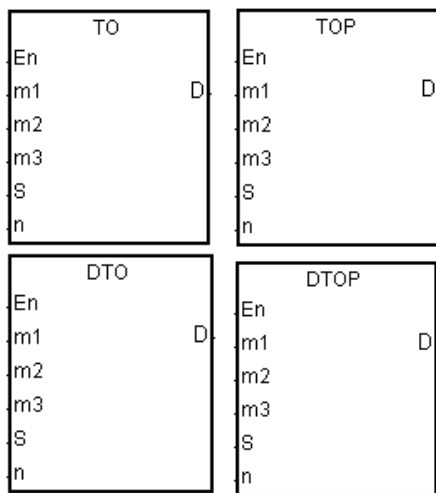
错误代码	说明
16#2003	请参考补充说明 1 跟 2
16#200B	请参考补充说明 3
16#1400	协处理器存取错误
16#1401	I/O 模块存取错误
16#1402	I/O 模块不符合 I/O 配置设定
16#1407	协处理器通讯错误

FB/FC	指令码			操作数					功能							
FC	D*	TO	P	m ₁ , m ₂ , m ₃ , S, D, n					特殊模块 CR 数据写入							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING		
m ₁ , m ₂ , m ₃		●	●*				●	●*								
S		●	●*				●	●*								
D		●	●*				●	●*								
n		●	●*				●	●*								

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
m ₁ , m ₂ , m ₃	●	●			●	●	●	●	●				●	○	○		
S	●	●			●	●	●	●	●				●	○	○		
D	●	●			●	●	●	●	●				●				
n	●	●			●	●	●	●	●				●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



- m₁ : Rack 代码
- m₂ : Slot 代码
- m₃ : 欲写入特殊模块之 CR (Controlled Register) 编号
- S : 写入 CR 的数据
- D : 存放错误代码的位置
- n : 一次写入的数据笔数

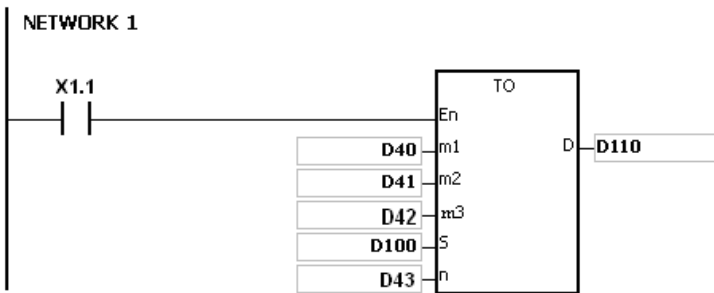
指令说明：

1. AH500 系列 PLC 可利用此指令读取特殊模块的 CR 数据。
2. m₁ : 背板 Rack 代码, m₁=1~8 ; 其中 m₁=1 代表主背板, m₁=2~8 代表扩展背板。
3. m₂ : 插槽 Slot 代码, m₂=0~11 ; 若 m₁=1 (主背板) , 则 m₂ (插槽) =0~11 ; 若 m₁=2~8 (扩展背板) , 则 m₂=0~7。
4. m₃ : 欲写入特殊模块之 CR (Controlled Register) 编号

5. D : 开始执行 TO 时, D 会被设定为 0 (表示无错误), 当有错误时, D 为非 0。有关错误代码说明请参考补充说明。
6. n : 读取之数据笔数
7. 16 位指令中, n=1~256 ; 32 位指令中, n=1~128。
8. 32 位指令才可以使用 HC 装置。
9. 特殊模块包含 AIO、NIO 与 PIO。
10. 当 S 为 KH 时, 会传送 n 个 KH 给指定的 I/O 模块。例如 : S 为 16#0001, n 为 3, 则传送三个 16#0001 给 I/O 模块。

程序范例 :

当启动 X1.1=OFF→ON 时, 会执行 TO 的应用指令 将置放于 CPU 模块右侧第一个特殊模块 AH-10SCM-A5 的 COM1 数据交换读取触发的运作模式 (CR#0007) 从不触发转换成触发一次, 因为执行无误所以 D110=16#0000。



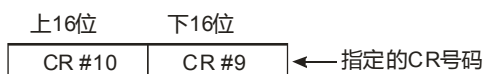
各参数使用说明如下 :

- 模块位于主背板, 因此背板 (Rack) 编号 D40 设定为 16#0001。
- 模块放置在第一个插槽, 因此插槽 (Slot) 编号 D41 设定为 16#0000。
- 模块之 COM1 数据交换读取触发设定值为 CR#7, 因此 CR 编号 D42 设定为 16#0007。
- 模块之 COM1 数据交换读取触发设定值只占用一个寄存器, 因此写入笔数 D43 设定为 1。
- 要写入模块 CR#0007 的数据会储存在 D100 里, 因此 D100 设定为 16#0002。

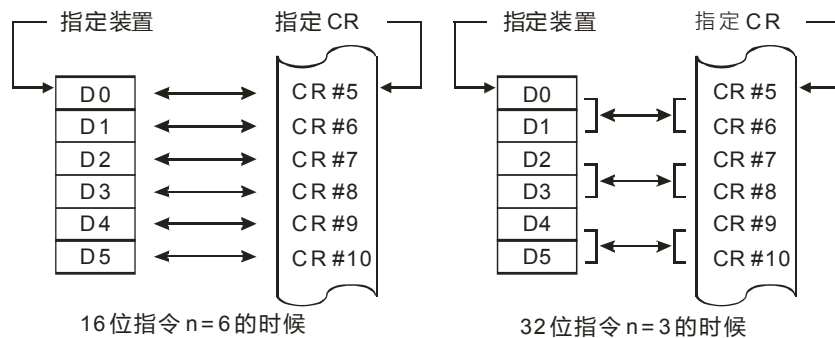
指令操作数的规则说明 :

- m_1 : 背板 Rack 代码, $m_1=1\sim 8$; 其中 $m_1=1$ 代表主背板, $m_1=2\sim 8$ 代表扩展背板。
- m_2 : 插槽 Slot 代码, $m_2=0\sim 11$ 。
若 $m_1=1$ (主背板), 则 m_2 (插槽) =0~11 ; 若 $m_1=2\sim 8$ (扩展背板), 则 $m_2=0\sim 7$ 。
- m_3 : CR 的号码, 特殊模块的内部内建 16 位长度的内存, 称之为 CR (Controlled Register)。CR 的编号以 10 进制编码#0~#N, 特殊模块的各种运转情况及设定值均被包含在里面。N 的个数依据不同模块而有所不同。
- 最多可挂 68 台特殊模块, 且不占用 I/O 点数。

- 如果使用 FROM/TO 指令时，一次以一个编号的 CR 为读出/写入单位，若是使用 DFROM/DTO 指令时，一次以 2 个编号的 CR 为读出/写入单位。



- 传送组数 n ，16 位指令的 $n=2$ 与 32 位指令的 $n=1$ 意义相同。



补充说明：

1. m_1 与 m_2 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 $S \sim S+n-1$ 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. n 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 由于使用 TO 会降低 CPU 模块与 I/O 模块的执行效能，所以不建议使用。
5. 错误说明

错误代码	说明
16#2003	请参考补充说明 1 跟 2
16#200B	请参考补充说明 3
16#1400	协处理器存取错误
16#1401	I/O 模块存取错误
16#1402	I/O 模块不符合 I/O 配置设定
16#1407	协处理器通讯错误

3.18 浮点数指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	–	<u>FSIN</u>	✓	浮点数 SIN 运算	5-6
FC	–	<u>FCOS</u>	✓	浮点数 COS 运算	5-6
FC	–	<u>FTAN</u>	✓	浮点数 TAN 运算	5-6
FC	–	<u>FASIN</u>	✓	浮点数 ASIN 运算	5-6
FC	–	<u>FACOS</u>	✓	浮点数 ACOS 运算	5-6
FC	–	<u>FATAN</u>	✓	浮点数 ATAN 运算	5-6
FC	–	<u>FSINH</u>	✓	浮点数 SINH 运算	5-6
FC	–	<u>FCOSH</u>	✓	浮点数 COSH 运算	5-6
FC	–	<u>FTANH</u>	✓	浮点数 TANH 运算	5-6
FC	–	<u>FRAD</u>	✓	角度→弧度	5-6
FC	–	<u>FDEG</u>	✓	弧度→角度	5-6
FC	<u>SQR</u>	<u>DSQR</u>	✓	BIN 开平方根	5
FC	–	<u>FSQR</u>	✓	浮点数开平方根	5-6
FC	–	<u>FEXP</u>	✓	浮点数取指数	5-6
FC	–	<u>FLOG</u>	✓	浮点数取对数	7-9
FC	–	<u>FLN</u>	✓	浮点数取自然对数	5-6
FC	–	<u>FPOW</u>	✓	浮点数权值指令	7-9
FC	<u>RAND</u>	–	✓	随机数值	7
FC	<u>BSQR</u>	<u>DBSQR</u>	✓	BCD 开平方根	5
FC	<u>BSIN</u>	–	✓	BCD SIN 运算	5
FC	<u>BCOS</u>	–	✓	BCD COS 运算	5
FC	<u>BTAN</u>	–	✓	BCD TAN 运算	5
FC	<u>BASIN</u>	–	✓	BCD ASIN 运算	5
FC	<u>BACOS</u>	–	✓	BCD ACOS 运算	5
FC	<u>BATAN</u>	–	✓	BCD ATAN 运算	5

FB/FC	指令码			操作数				功能				
FC	D*	FSIN	P	S, D				浮点数 SIN 运算				

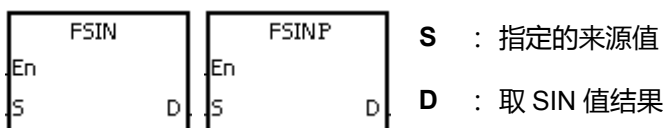
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

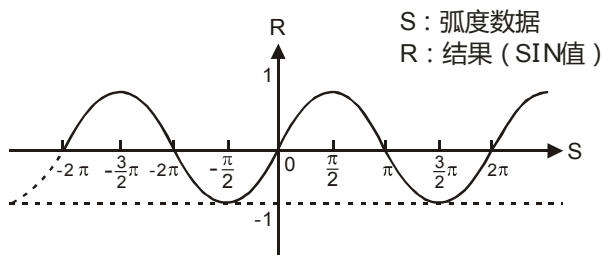
3

图形：



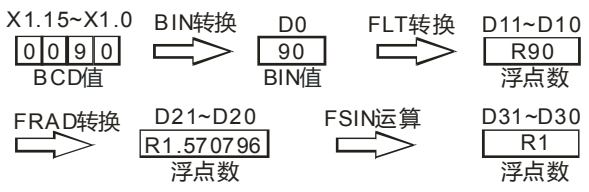
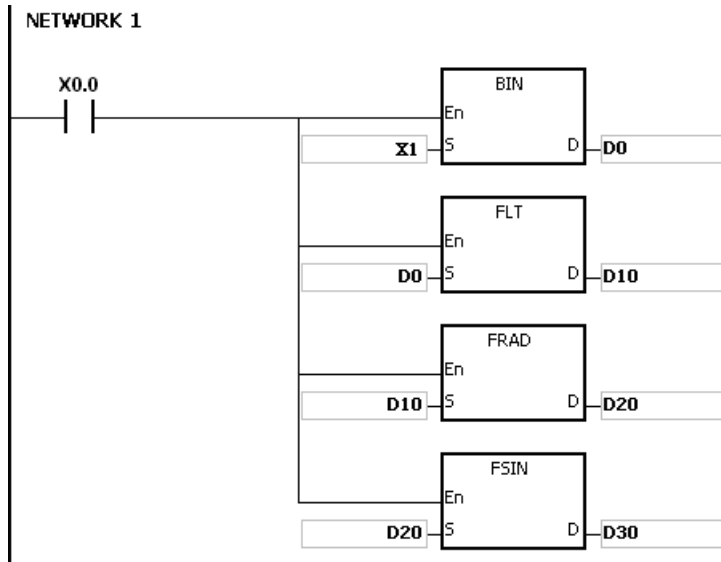
指令说明：

1. S 所指定的来源可为弧度或角度，由标志 SM695 决定。
2. 当 SM695=OFF 时，指定为弧度模式，弧度值=角度×π/180。
3. 当 SM695=ON 时，指定为角度模式，角度值=弧度×180/π，角度范围：0°≤角度值≤360°。
4. 若转换结果为 0，则零标号 SM600=ON。
5. 将 S 所指定的来源值，求取 SIN 值后存于 D 所指定的寄存器当中。
6. 下图显示弧度与结果的关系：



程序范例：

当 X0.0=ON 时 将 (X1.15~X1.0) 的 BCD 值转成 BIN 值存于 D0 将 D0 做 FLT 运算转成浮点数后存于 (D11 , D10) ，将 (D11 , D10) 做 FRAD 运算转成弧度存于 (D21 , D20) ，再将 (D21 , D20) 的弧度做 SIN 运算后存于 (D31 , D30) 当中，内容为浮点数。



补充说明：

1. 当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。
2. 当 SM695=ON，且 S 的内容值小于 0 或大于 360，指令不执行，SM0=ON，错误码 SR0=16#2003。

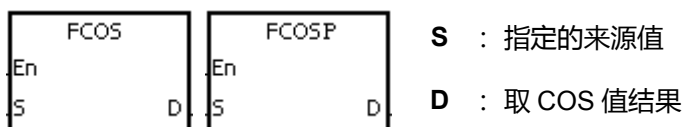
FB/FC	指令码			操作数				功能				
FC	D*	FCOS	P	S, D				浮点数 COS 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

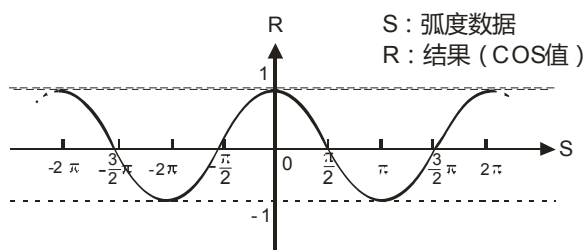
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



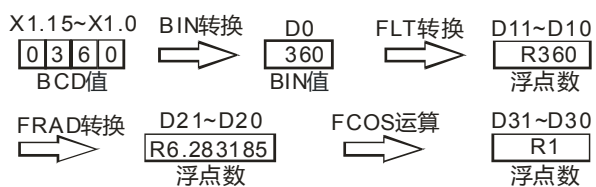
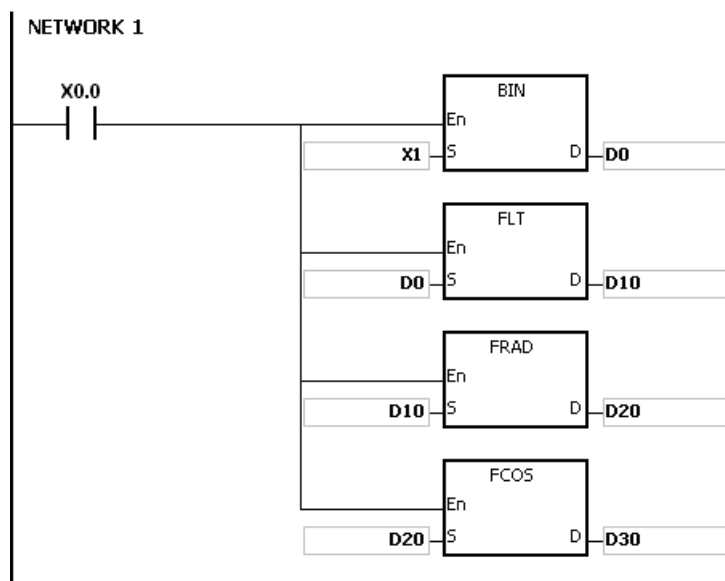
指令说明：

1. S 所指定的来源可为弧度或角度，由标志 SM695 决定。
2. 当 SM695=OFF 时，指定为弧度模式，弧度值=角度 $\times\pi/180$ 。
3. 当 SM695=ON 时，指定为角度模式，角度值=弧度 $\times 180/\pi$ ，角度范围： $0^{\circ}\leq\text{角度值}\leq 360^{\circ}$ 。
4. 若转换结果为 0，则零标号 SM600=ON。
5. 将 S 所指定的来源值，求取 COS 值后存于 D 所指定的寄存器当中。
6. 下图显示弧度与结果的关系：



程序范例：

当 X0.0=ON 时 将 (X1.15~X1.0) 的 BCD 值转成 BIN 值存于 D0 将 D0 做 FLT 运算转成浮点数后存于 (D11, D10)，将 (D11, D10) 做 DRAD 运算转成弧度存于 (D21, D20)，再将 (D21, D20) 的弧度做 COS 运算后存于 (D31, D30) 当中，内容为浮点数。



补充说明：

1. 当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。
2. 当 SM695=ON，且 S 的内容值小于 0 或大于 360，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数				功能				
FC	D*	FTAN	P	S, D				浮点数 TAN 运算				

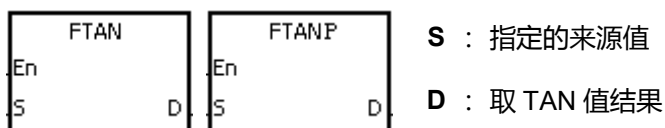
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

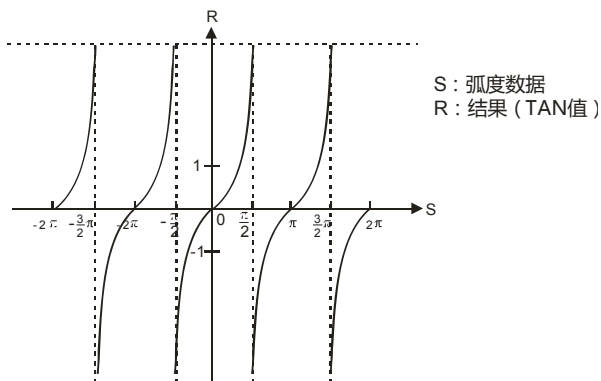
3

图形：



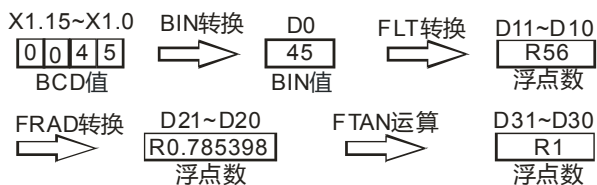
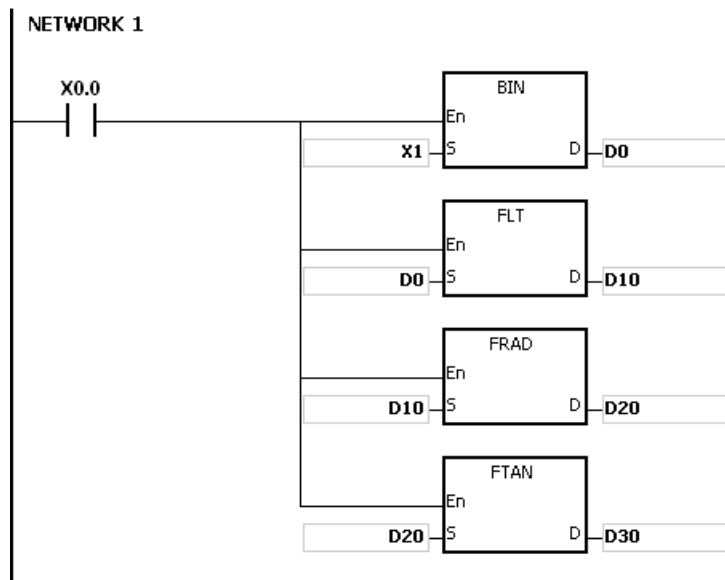
指令说明：

1. S 所指定的来源可为弧度或角度，由标志 SM695 决定。
2. 当 SM695=OFF 时，指定为弧度模式，弧度值=角度×π/180。
3. 当 SM695=ON 时，指定为角度模式，角度值=弧度×180/π，角度范围：0°≤角度值≤360°。
4. 若转换结果为 0，则零标号 SM600=ON。
5. 将 S 所指定的来源值，求取 TAN 值后存于 D 所指定的寄存器当中。
6. 下图显示弧度与结果的关系：



程序范例：

当 X0.0=ON 时 将 (X1.15~X1.0) 的 BCD 值转成 BIN 值存于 D0 将 D0 做 FLT 运算转成浮点数后存于 (D11 , D10) ，将 (D11 , D10) 做 DRAD 运算转成弧度存于 (D21 , D20) ，再将 (D21 , D20) 的弧度做 TAN 运算后存于 (D31 , D30) 当中，内容为浮点数。

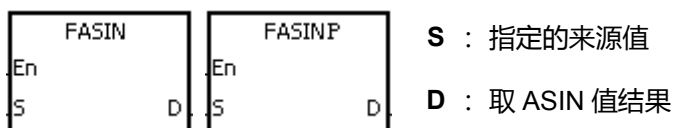


补充说明：

1. 当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。
2. 当 SM695=ON，且 S 的内容值小于 0 或大于等于 360 或等于 90 或等于 270 时，指令不执行，SM0=ON，错误码 SR0=16#2003。

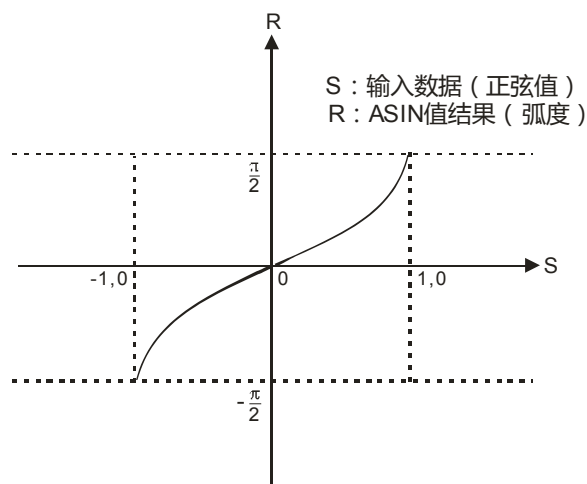
FB/FC	指令码			操作数							功能						
FC	D*	FASIN	P	S, D							浮点数 ASIN 运算						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S										●	●*						
D			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				
							脉冲执行型			16 位指令			32 位指令				
							AH 运动控制主机			AH 运动控制主机			-				

图形：



指令说明：

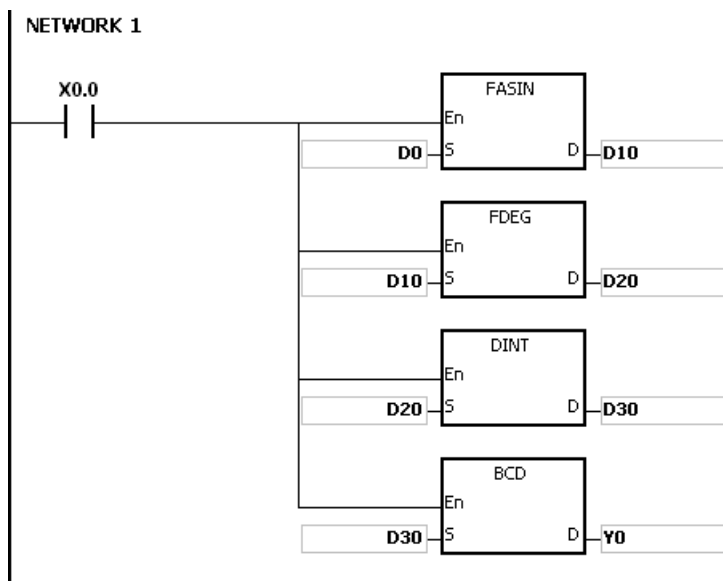
1. ASIN 值= \sin^{-1} ，下图显示输入数据与结果的关系：



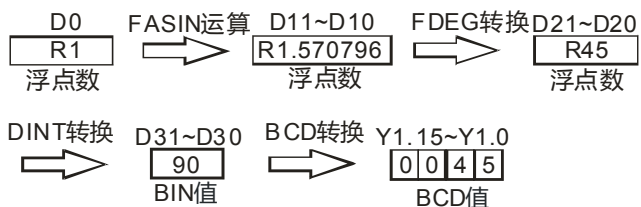
2. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

当 X0.0=ON 时，指定浮点数 (D1, D0) 求取 ASIN 值后存于 (D11, D10) 当中，将 (D11, D10) 的结果做 FDEG 转为角度存于 (D21, D20)，再将 (D21, D20) 做 DINT 转换成整数存于 (D31, D30)，再透过 BCD 指令将求得角度输出于 Y0.15~Y0.0。



3



补充说明：

1. **S** 操作数指定的正弦值数值的十进制浮点值只能介于-1.0~+1.0 之间，若不在此范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 **S** 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

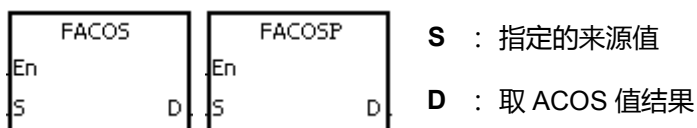
FB/FC	指令码			操作数					功能				
FC	D*	FACOS	P	S, D					浮点数 ACOS 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

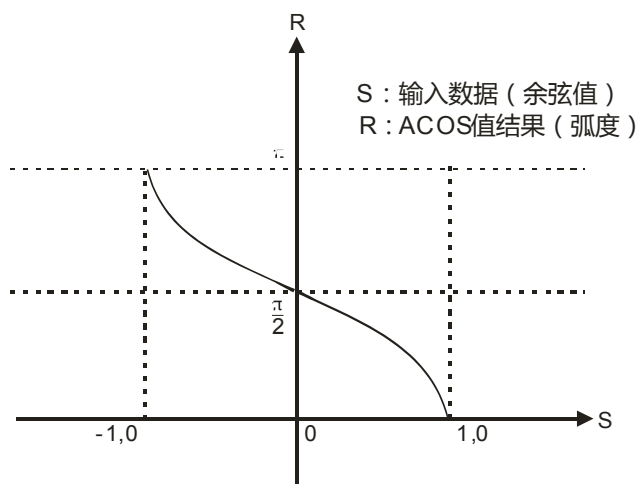
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

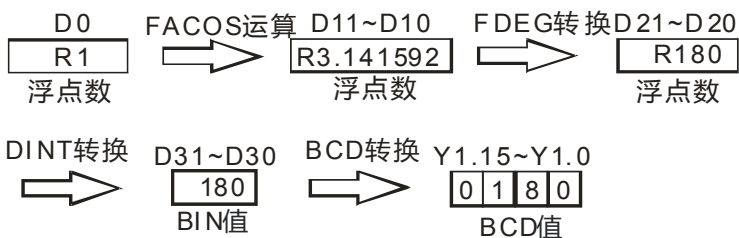
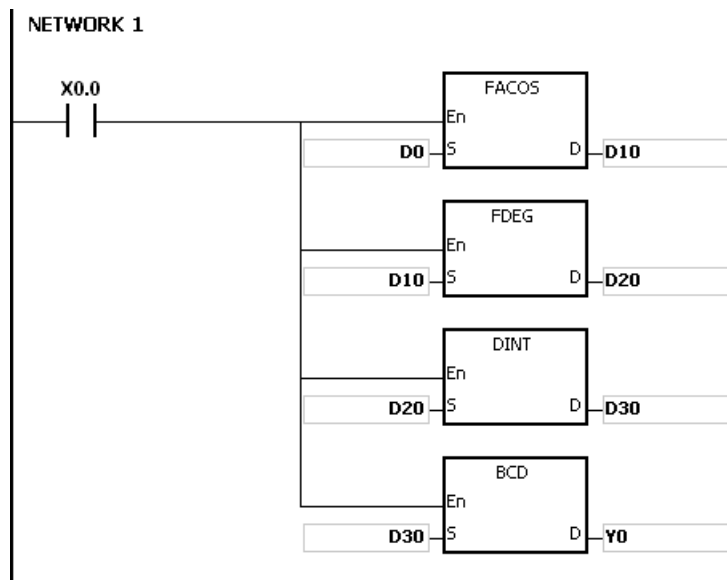
1. $ACOS 值 = \cos^{-1}$ ，下图显示输入数据与结果的关系：



2. 若转换结果的绝对值大于可表示的最大浮点值，则进位标号 SM602=ON。
3. 若转换结果的绝对值小于可表示的最小浮点值，则借位标号 SM601=ON。
4. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

当 X0.0=ON 时，指定浮点数 (D1, D0) 求取 ACOS 值后存于 (D11, D10) 当中，将 (D11, D10) 的结果做 FDEG 转为角度存于 (D21, D20) ，再将 (D21, D20) 做 DINT 转换成整数存于 (D31, D30) ，再透过 BCD 指令将求得角度输出于 Y0.15~Y0.0。



补充说明：

1. S 操作数指定的正弦值数值的十进制浮点值只能介于-1.0~+1.0 之间，若不在此范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数				功能				
FC	D*	FATAN	P	S, D				浮点数 ATAN 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

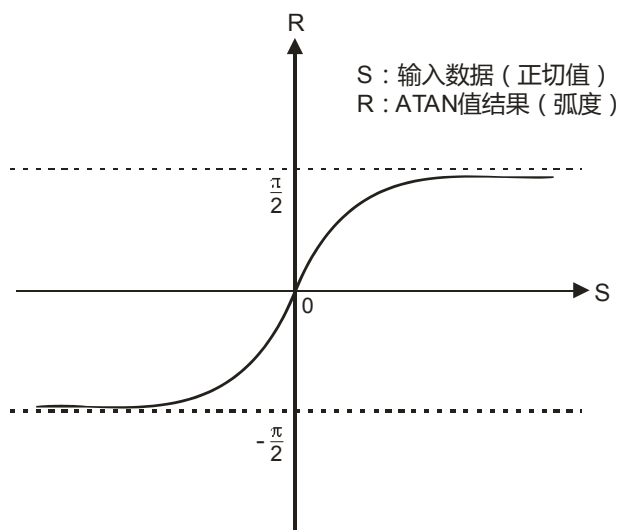
3

图形：



指令说明：

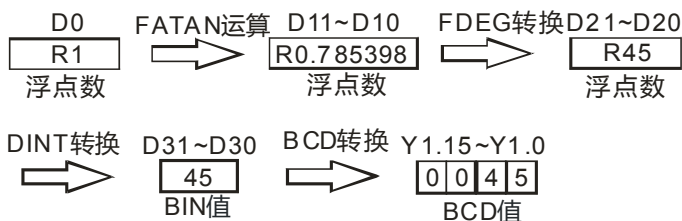
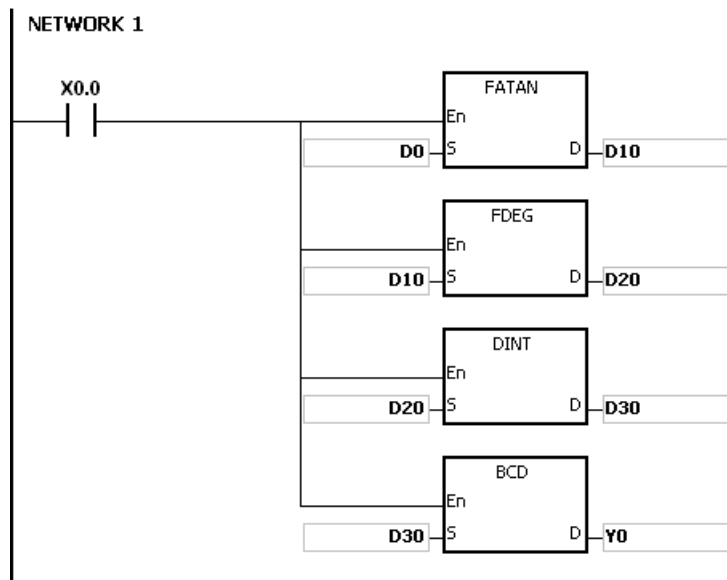
1. ATAN 值 = \tan^{-1}
2. 下图显示输入数据与结果的关系：



3. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

当 X0.0=ON 时，指定浮点数 (D1, D0) 求取 ATAN 值后存于 (D11, D10) 当中，将 (D11, D10) 的结果做 FDEG 转为角度存于 (D21, D20)，再将 (D21, D20) 做 DINT 转换成整数存于 (D31, D30)，再透过 BCD 指令将求得角度输出于 Y0.15~Y0.0。



补充说明：

当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数				功能				
FC	D*	FSINH	P	S, D				浮点数 SINH 运算				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

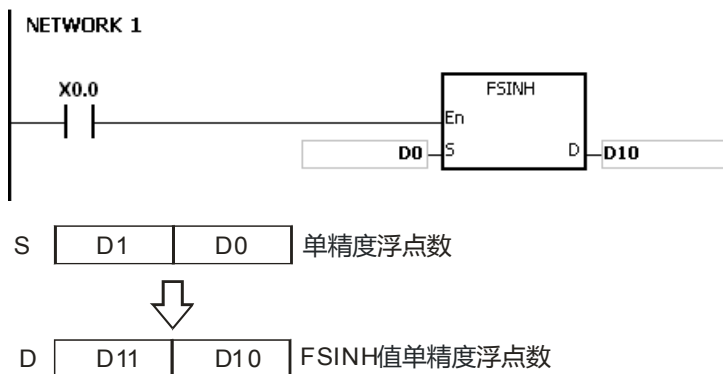


指令说明：

1. SINH 值 = $(e^S - e^{-S}) / 2$ 。
2. 若转换结果的绝对值大于可表示的最大浮点值，则 D=16#7F800000，进位标号 SM602=ON。
3. 若转换结果的绝对值小于可表示的最小浮点值，则 D=16#FF800000，借位标号 SM601=ON。
4. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

1. 当 X0.0=ON 时，指定浮点数 (D1, D0) 求取 SINH 值后存于 (D11, D10) 当中，内容为浮点数。



2. 若转换结果的绝对值大于可表示的最大浮点值，则进位标号 SM602=ON。
3. 若转换结果的绝对值小于可表示的最小浮点值，则借位标号 SM601=ON。

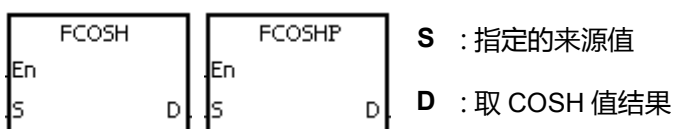
4. 若转换结果为 0，则零标号 SM600=ON。

补充说明：

当 **S** 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数							功能							
FC	D*	FCOSH	P	S, D							浮点数 COSH 运算							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S										●	●*							
D			●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●		●				○	
D	●	●			●	●	●	●	●		●		●					
							脉冲执行型				16 位指令				32 位指令			
							AH 运动控制主机				AH 运动控制主机				-			

图形：

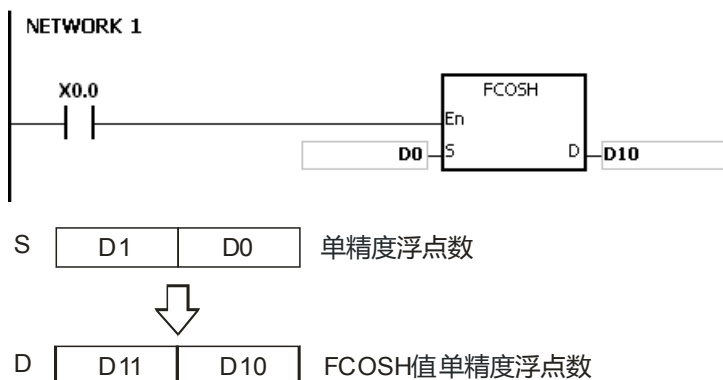


指令说明：

1. COSH 值 = $(e^s + e^{-s}) / 2$ 。
2. 若转换结果的绝对值大于可表示之最大浮点值，则 D=16#7F800000，进位标号 SM602=ON。
3. 若转换结果的绝对值小于可表示之最小浮点值，则 D=16#FF800000，借位标号 SM601=ON。
4. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

1. 当 X0.0=ON 时，指定浮点数 (D1, D0) 求取 COSH 值后存于 (D11, D10) 当中，内容为浮点数。



2. 若转换结果的绝对值大于可表示的最大浮点值，则进位标号 SM602=ON。
3. 若转换结果的绝对值小于可表示的最小浮点值，则借位标号 SM601=ON。

4. 若转换结果为 0，则零标号 SM600=ON。

补充说明：

当 **S** 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码			操作数				功能				
FC	D*	FTANH	P	S, D				浮点数 TANH 运算				

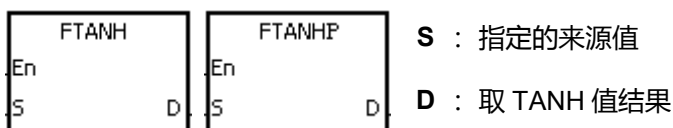
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

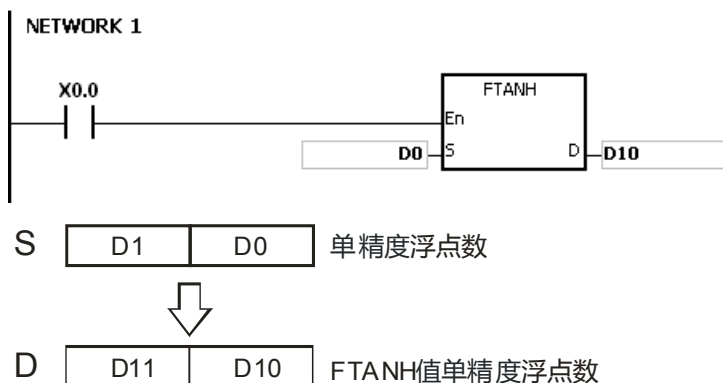


指令说明：

1. $TANH 值 = (e^s - e^{-s}) / (e^s + e^{-s})$ 。
2. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

1. 当 X0.0=ON 时，指定浮点数 (D1, D0) 求取 TANH 值后存于 (D11, D10) 当中，内容为浮点数。



2. 若转换结果为 0，则零标号 SM600=ON。

补充说明：

当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码		操作数			功能				
FC		FRAD	P	S, D			角度→弧度			

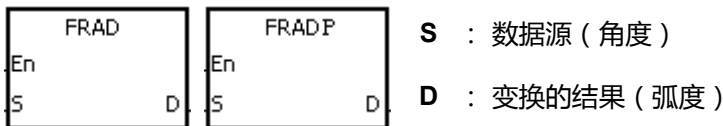
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●				
D				●					●					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

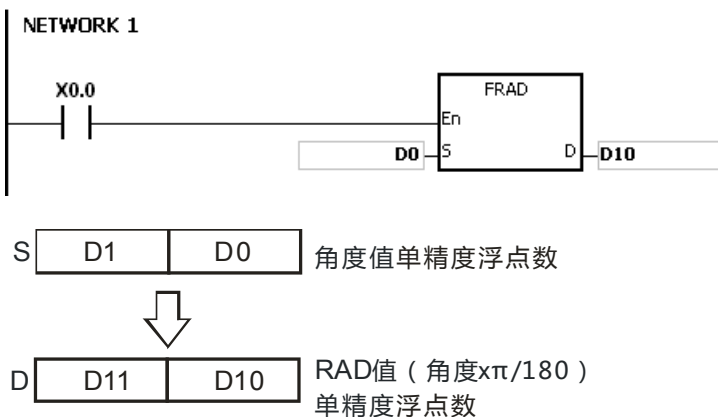


指令说明：

1. 使用下列公式将角度转换成弧度。
2. 弧度 = 角度 × (π/180)
3. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

当 X0.0=ON 时，指定浮点数 (D1, D0) 之角度值，将角度转换成弧度值后存于 (D11, D10) 当中，内容为浮点数。



补充说明：

当 **S** 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码		操作数				功能					
FC		FDEG	P	S, D				弧度→角度				

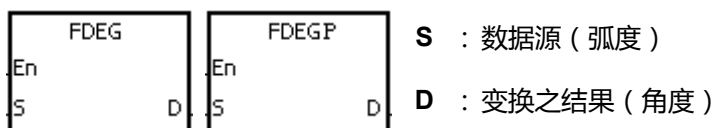
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●				
D				●					●					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

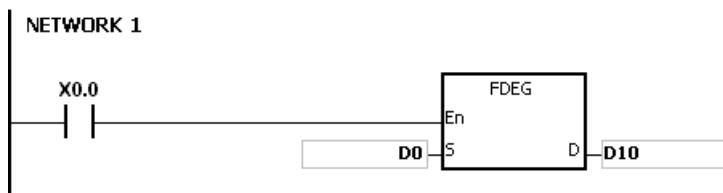


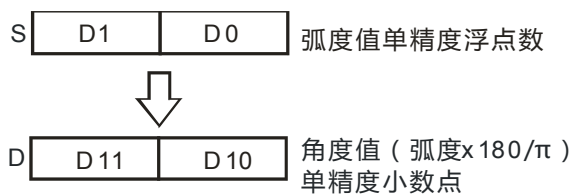
指令说明：

1. 使用下列公式将弧度转换成角度。
2. 角度 = 弧度 × (180/π)
3. 若转换结果的绝对值大于可表示的最大浮点值，则 D=16#7F7F F FFF。
4. 若转换结果的绝对值小于可表示的最小浮点值，则 D=16#FF7 F F FFF。
5. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

当 X0.0=ON 时，指定浮点数 (D1, D0) 的角度值，将弧度值转换成角度后存于 (D11, D10) 当中，内容为浮点数。





补充说明：

当 **S** 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

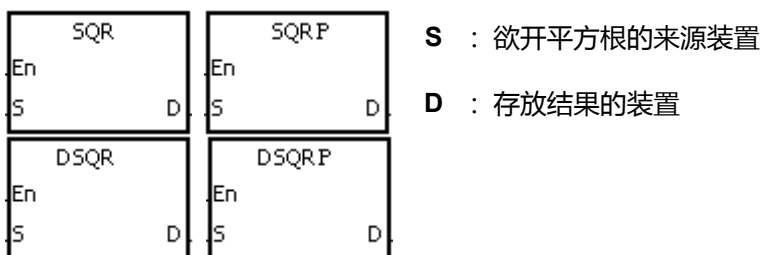
FB/FC	指令码			操作数	功能
FC	D*	SQR	P	S, D	开平方根

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●	○	○		
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：

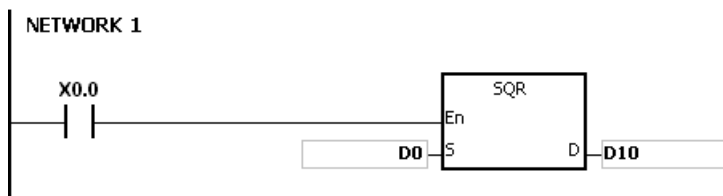


指令说明：

1. 将 S 所指定之装置内容值开平方根后，存放于 D 所指定的装置。
2. 运算结果 D 只求整数，小数点被舍弃。有小数点被舍弃时，借位标志信号 SM601=ON。
3. 运算结果 D 为 0 时，零标志信号 SM600=ON。

程序范例：

当 X0.0=ON 时，将 D0 内容值开平方根后，存放于 D10 内。



补充说明：

S 只可以指定正数，若指定负数时，PLC 视为“指令运算错误”，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数	功能
FC	D*	FSQR	P	S, D	浮点数开平方根

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S										●	●*			
D			●	●*				●	●*					

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

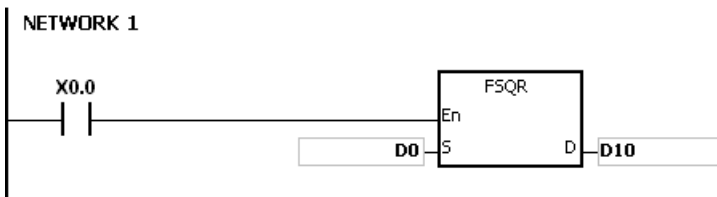


指令说明：

- S 所指定的寄存器内容被开平方，所得的结果暂存于 D 所指定的寄存器内容，开平方的动作全部以浮点数类型进行。
- 若运算结果为 0，则零标号 SM600=ON。

程序范例一：

当 X0.0=ON 时，将浮点数 (D11, D10) 取开平方根，将结果存放至 (D11, D10) 所指定的寄存器当中。

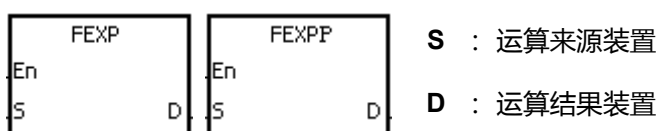


补充说明：

S 只可以指定正数，若指定负数时，PLC 视为“指令运算错误”，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数							功能							
FC	D*	FEXP	P	S, D							浮点数取指数							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S										●	●*							
D			●	●*				●	●*									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●		●				○	
D	●	●			●	●	●	●	●		●		●					
							脉冲执行型				16 位指令				32 位指令			
							AH 运动控制主机				AH 运动控制主机				-			

图形：

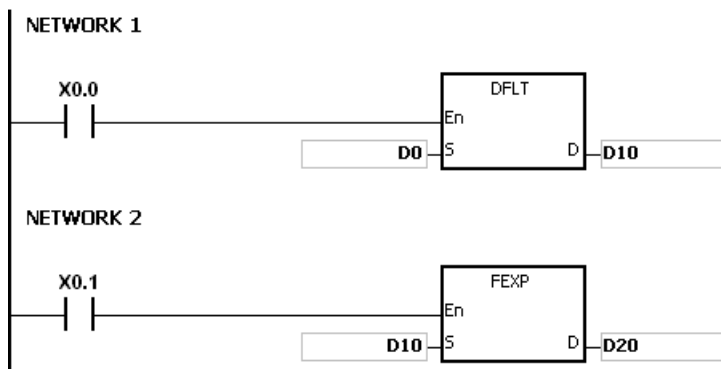


指令说明：

1. 以 $e=2.71828$ 为底数，S 为指数做 EXP 运算。
2. $EXP[D+1, D]=[S+1, S]$
3. S 内容正负数都有效，指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S 需转换为浮点数。
4. D 操作数内容值 $=e^S$ ； $e=2.71828$ ，S 为指定的源数据。
5. 若转换结果的绝对值大于可表示之最大浮点值，则 $D=16\#7F800000$ ，进位标号 $SM602=ON$ 。
6. 若运算结果为 0，则零标号 $SM600=ON$ 。

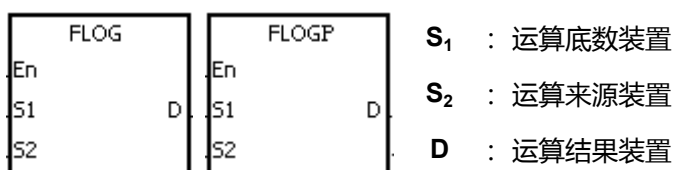
程序范例：

1. 当 X0.0 为 ON 时，将 (D1, D0) 值转成浮点数存于 (D11, D10) 寄存器中。
2. 当 X0.1 为 ON 时，(D11, D10) 为指数做 EXP 运算，其值为浮点数并存放于 (D21, D20) 寄存器中。



FB/FC	指令码			操作数							功能						
FC	D*	FLOG	P	S ₁ , S ₂ , D							浮点数取对数						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁ , S ₂										●	●*						
D			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				
							脉冲执行型	16 位指令					32 位指令				
							AH 运动控制主机	AH 运动控制主机					-				

图形：

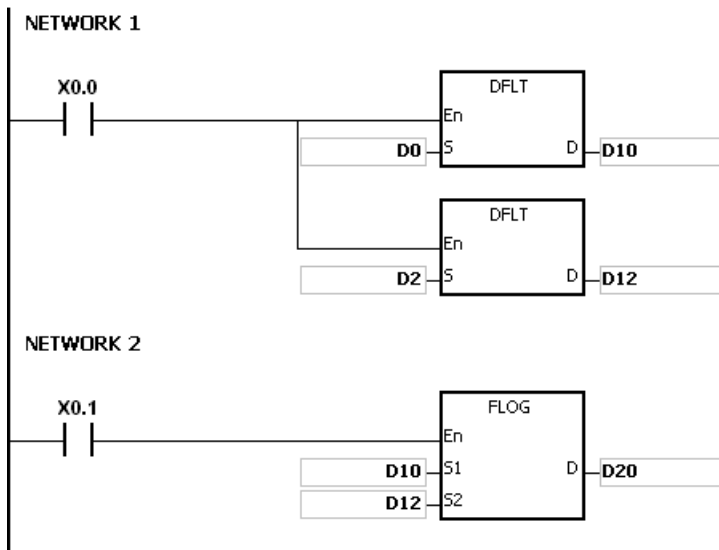


指令说明：

1. 将 S₁ 内容及 S₂ 内容为操作数做 log 运算，结果存放于 D。
2. S₁、S₂ 内容值只有正数有效，指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S₁、S₂ 需转换为浮点数。
3. S₁^D = S₂，求 D 值 → Log_{S₁}S₂ = D
4. 例：已知 S₁ = 5，S₂ = 125，求 D = log₅125 = ?
5. S₁^D = S₂ → 5^D = 125 → D = log₅125 = 3
6. 若运算结果为 0，则零标号 SM600 = ON。

程序范例：

1. 当 X0.0 为 ON 时，将 (D1, D0) 内容及 (D3, D2) 内容转成浮点数分别存于 (D11, D10) 及 (D13, D12) 32 位寄存器中。
2. 当 X0.1 为 ON 时，将 (D11, D10) 及 (D13, D12) 32 位寄存器浮点数做 Log 运算并将结果存于 (D21, D20) 32 位寄存器中。



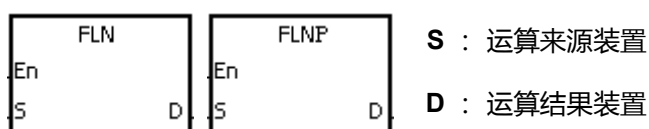
补充说明：

3

S_1 内容值 ≤ 1 或 $S_2 \leq 0$ 时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数					功能								
FC	D*	FLN	P	S, D					浮点数取自然对数								
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S										●	●*						
D			●	●*				●	●*								
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				
					脉冲执行型					16 位指令				32 位指令			
					AH 运动控制主机					AH 运动控制主机				-			

图形：

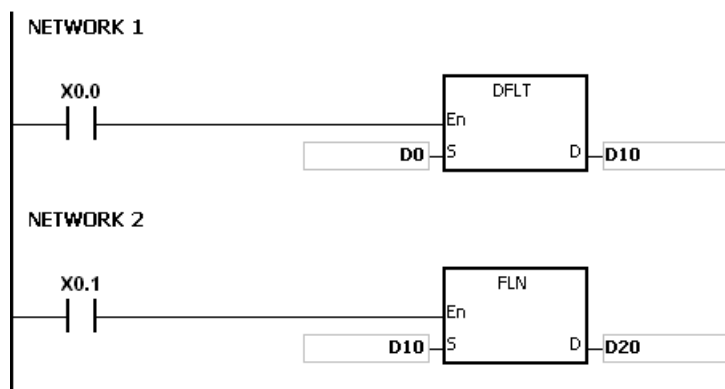


指令说明：

1. 以 S 为操作数做自然对数 \ln 运算。
2. $LN[S+1, S]=[D+1, D]$
3. S 内容只有正数有效，指定 D 寄存器时必须使用 32 位数据格式，运算时均以浮点数方式执行，故 S 需转换为浮点数。
4. $e D=S \rightarrow D$ 操作数内容值 $=\ln S$ ；S 为指定的源数据。
5. 若运算结果为 0，则零标号 SM600=ON。

程序范例：

1. 当 X0.0 为 ON 时，将 (D1, D0) 值转成浮点数存于 (D11, D10) 寄存器中。
2. 当 X0.1 为 ON 时，将 (D11, D10) 寄存器为真数做 \ln 运算，其值为浮点数并存放于 (D21, D20) 寄存器中。

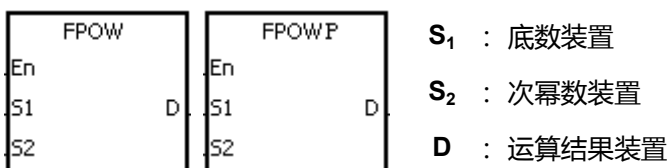


补充说明：

S≤0 时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数						功能							
FC		FPOW	P	S_1, S_2, D						浮点数权值指令							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S_1, S_2										●							
D			●					●									
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				
						脉冲执行型			16 位指令				32 位指令				
						AH 运动控制主机			AH 运动控制主机				-				

图形：



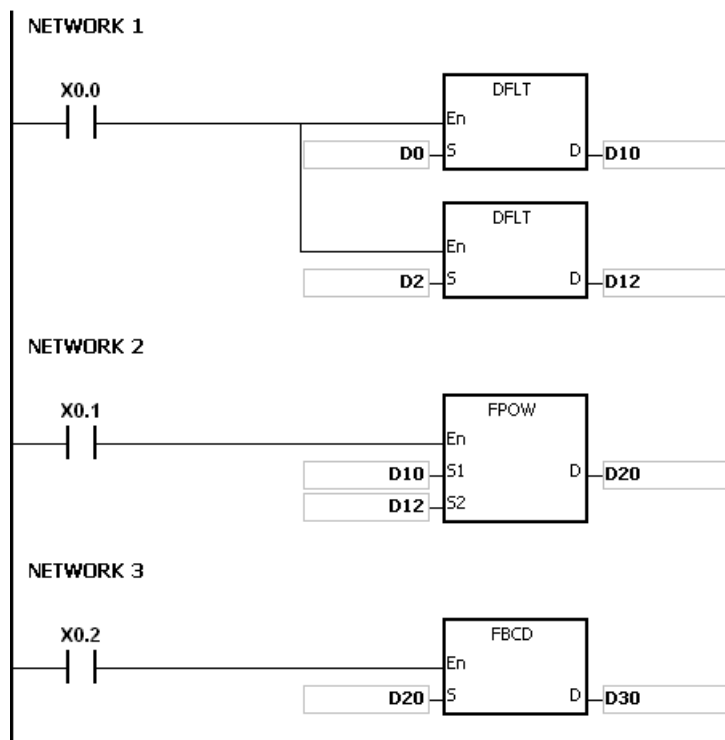
指令说明：

1. 将单精度浮点数数据 S_1 以 S_2 的次幂数相乘后存放于 D。
2. $D=POW[S_1+1, S_1]^{[S_2+1, S_2]}$
3. S_1 内容值只有正数有效, S_2 内容值正负值都有效。指定 D 寄存器时必须使用 32 位数据格式, 运算时均以浮点数方式执行, 故 S_1, S_2 需转换为浮点数。
4. 例: $S_1^{S_2}=D$, 求 D 值?
已知 $S_1=5, S_2=3$, 则 $D=5^3=125$
5. 若运算结果的绝对值大于可表示之最大浮点值, 则进位标号 SM602=ON。
6. 若运算结果的绝对值小于可表示之最小浮点值, 则借位标号 SM601=ON。
7. 若转换结果的绝对值大于可表示之最大浮点值, 则 $D=16\#7F7FFFFF$ 。
8. 若转换结果的绝对值小于可表示之最小浮点值, 则 $D=16\#FF7FFFFF$ 。
9. 若运算结果为 0, 则零标号 SM600=ON。

程序范例：

1. 当 X0.0 为 ON 时, 将 (D1, D0) 内容及 (D3, D2) 内容转成浮点数分别存于 (D11, D10) 及 (D13, D12) 32 位寄存器中。

2. 当 X0.1 为 ON 时,将(D11 ,D10)及(D13 ,D12)32 位寄存器浮点数做 POW 运算并将结果存于(D21 , D20) 32 位寄存器中。
3. 当 X0.2 为 ON 时,将 (D21 , D20) 32 位寄存器浮点数转成十进浮点数并存于 (D31 , D30) 寄存器中。
(此时 D31 为表示 D30 的 10 次幂方)



补充说明：

S₁<0 时,指令不执行, SM0=ON, 错误码 SR0=16#2003。

FB/FC	指令码		操作数				功能						
FC		RAND	P	S_1, S_2, D				随机数值					

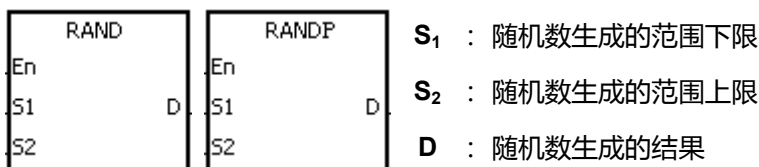
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1, S_2	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

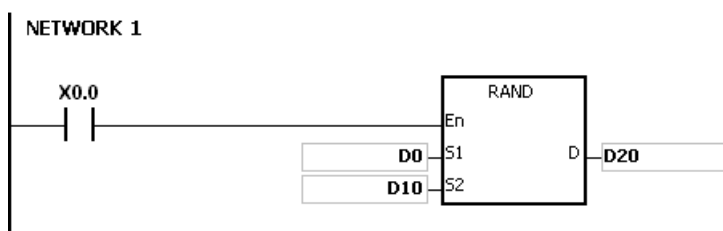


指令说明：

1. 产生介于范围下限 S_1 与范围上限 S_2 之随机数，将结果存放到 D 内。
2. 若用户输入 $S_1 > S_2$ ，指令执行时把 S_1 作为上限值， S_2 下限值进行比较。

程序范例：

当 X0.0=ON 时，RAND 指令产生介于范围下限 D0 与范围上限 D10 之随机数，将结果存放到 D20 内。



补充说明：

S_1 或 S_2 操作数内容值范围为 0~32767，内容值超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

FB/FC	指令码			操作数	功能
FC	D*	BSQR	P	S, D	BCD 开平方根

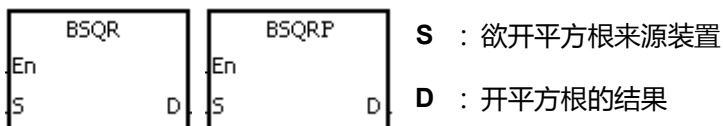
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

3

图形：

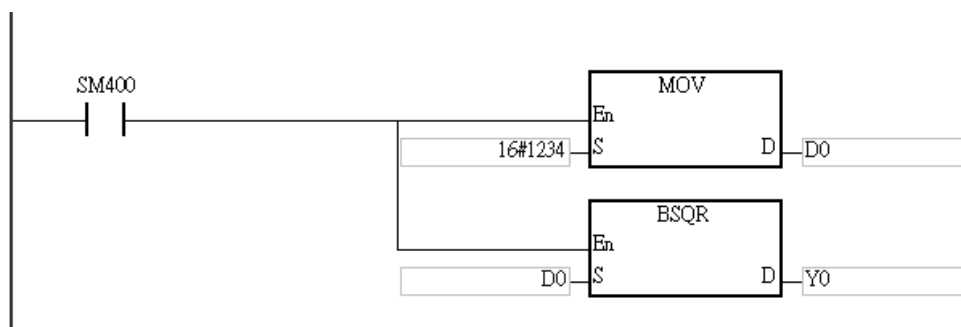


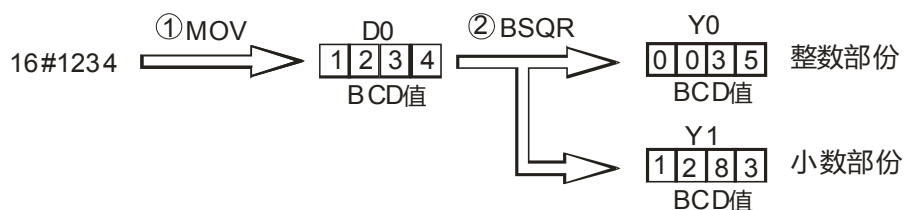
指令说明：

1. 将 S 所指定的装置内容值开平方根后，将结果的整数部份存放于 D，小数部份存于 D+1 所指定的装置。
2. 数据源 S 的内容有效数值范围：BCD (0~9,999)，DBCD (0~99,999,999)。
3. BSQR 小数部份计算到小数点后第 4 位。
4. DBSQR 小数部份计算到小数点后第 8 位。
5. 运算结果 D 为 0 时，零标志信号 SM600=ON。

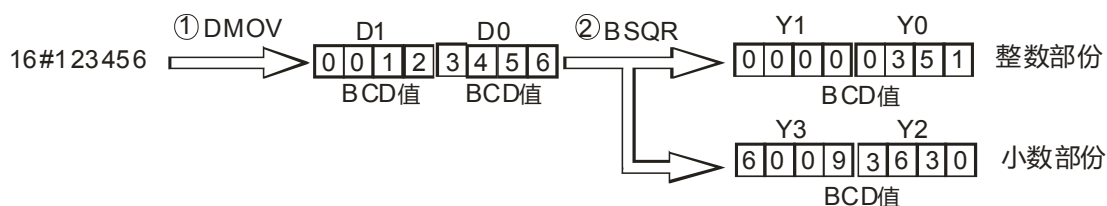
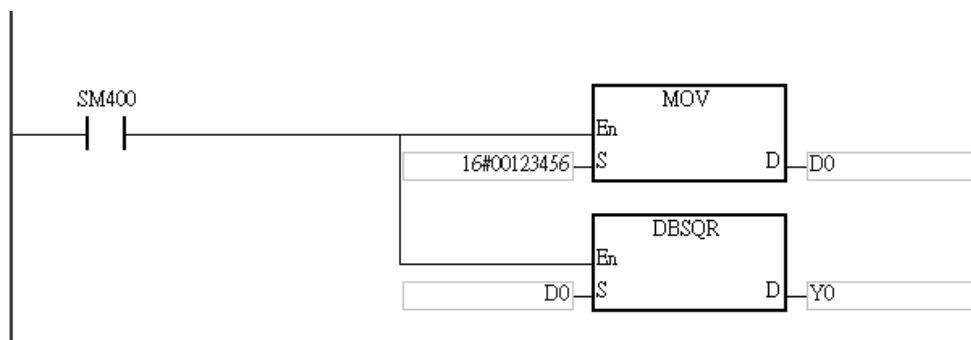
程序范例一：

D0 的内容经开根号后，将结果整数部分存放至 Y0，小数点部分存放至 Y1。



**程序范例二：**

D0 的内容经开根号后，将结果整数部分存放至 Y0，小数点部分存放至 Y1。

**补充说明：**

1. 当 S 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误，SM0=ON，错误码 SR0=16#200D。
2. 16 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
3. 32 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of DWORD/DINT。

FB/FC	指令码		操作数				功能					
FC		BSIN	P	S, D				BCD SIN 运算				

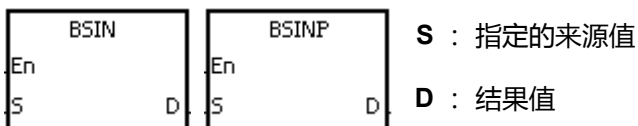
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

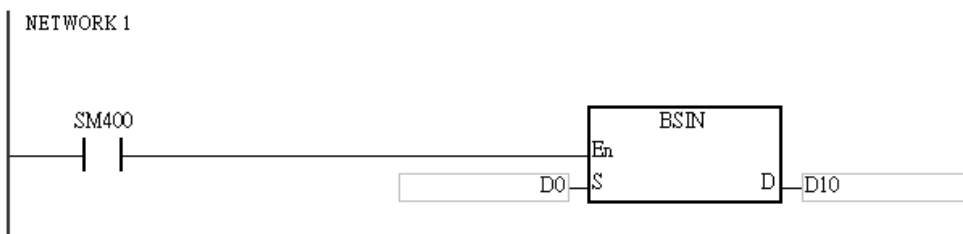


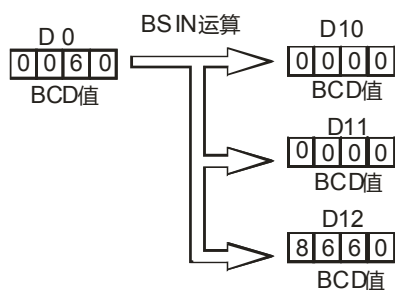
指令说明：

1. **S** 所指定的来源为角度,求取 SIN 值,运算结果的符号存放于 **D**,整数部份存于 **D+1**,小数部份存于 **D+2**。
2. 角度范围： $0^{\circ} \leq \text{角度值} < 360^{\circ}$ 。
3. 运算结果计算到小数点后第 5 位四舍五入。
4. 若转换结果为 0, 则零标号 SM600=ON。

程序范例：

D0 的内容经过 SIN 运算后将结果储存至 D 操作数中,符号部份储存于 D10;整数部份储存于 D11;小数部份储存于 D12。



**补充说明：**

1. 当 **S** 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误，SM0=ON，错误码 SR0=16#200D。
2. 当 **S** 的数据内容不在 0°到 360°，则将会产生运算错误，SM0=ON，错误码 SR0=16#2003。
3. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

FB/FC	指令码		操作数				功能					
FC		BCOS	P	S, D				BCD COS 运算				

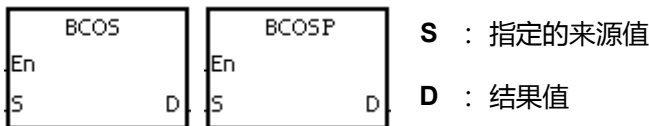
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

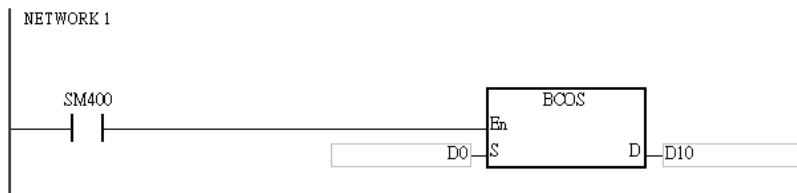


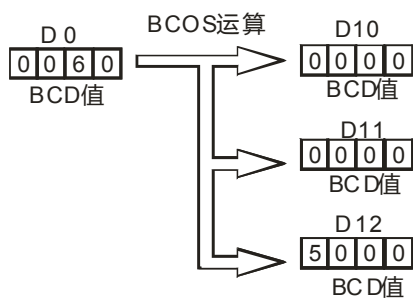
指令说明：

1. **S** 所指定的来源为角度，求取 COS 值，运算结果的符号存放于 **D**，整数部份存于 **D+1**，小数部份存于 **D+2**。
2. 角度范围： $0^\circ \leq \text{角度值} < 360^\circ$ 。
3. 运算结果计算到小数点后第 5 位四舍五入。
4. 若转换结果为 0，则零标号 SM600=ON。

程序范例：

D0 的内容经过 COS 运算后将结果储存至 D 操作数中，符号部份储存于 D10；整数部份储存于 D11；小数部份储存于 D12。



**补充说明：**

1. 当 **S** 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误，SM0=ON，错误码 SR0=16#200D。
2. 当 **S** 的数据内容不在 0°到 360°，则将会产生运算错误，SM0=ON，错误码 SR0=16#2003。
3. **D** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

4.

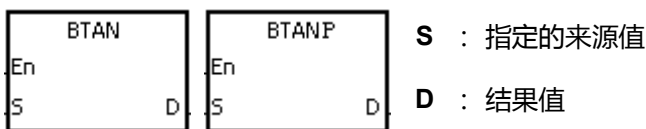
FB/FC	指令码			操作数						功能					
FC		BTAN	P	S, D						BCD TAN 运算					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

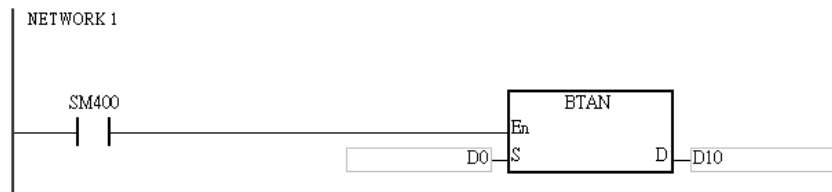


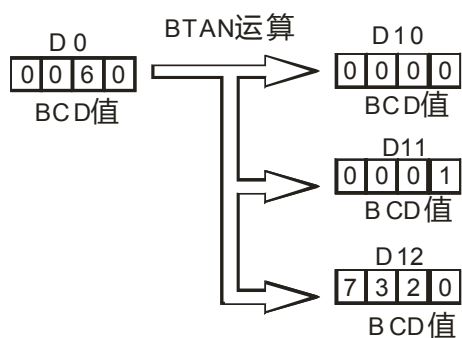
指令说明：

- S** 所指定的来源为角度，求取 TAN 值，运算结果的符号存放于 **D**，整数部份存于 **D+1**，小数部份存于 **D+2**。
- 角度范围： $0^{\circ} \leq \text{角度值} < 360^{\circ}$ 。
- 运算结果计算到小数点后第 5 位四舍五入。
- 若转换结果为 0，则零标号 SM600=ON。

程序范例：

D0 的内容经过 TAN 运算后将结果储存至 D 操作数中，符号部份储存于 D10；整数部份储存于 D11；小数部份储存于 D12。



**补充说明：**

1. 当 **S** 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误，SM0=ON，错误码 SR0=16#200D。
2. 当 **S** 的数据内容不在 0°到 360°，则将会产生运算错误，SM0=ON，错误码 SR0=16#2003。
3. 当 **S** 的数据内容=90°或 270°，则将会产生运算错误，SM0=ON，错误码 SR0=16#2003。
4. **D** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

5.

FB/FC	指令码		操作数				功能					
FC		BASIN	P	S, D				BCD ASIN 运算				

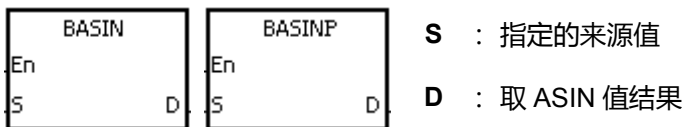
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

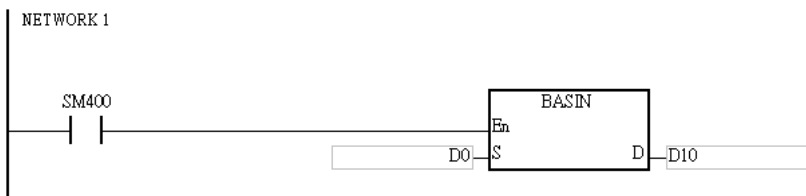


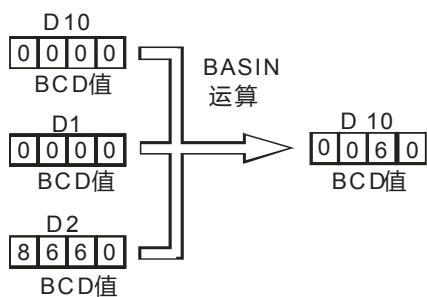
指令说明：

1. **S** 所指定的来源（BCD），求取 ASIN 值= \sin^{-1} 值，运算的结果存放于 **D**（角度）。
2. **S** 的内容值代表符号，0=正值，1=负值，S+1 代表整数部份，S+2 代表小数部份。
3. 运算结果将小数四舍五入。
4. 运算结果为 0° 到 90° 和 270° 到 360° 之间的 BCD 值（角度）。

程序范例：

D0 内容为 SIN^{-1} 之正负号，D1 内容为 SIN^{-1} 的整数部分，D2 值为 SIN^{-1} 小数部份。经过 BASIN 指令运算后，其结果四舍五入至整数储存于 D10 内。



**补充说明：**

1. 小数部份输入注意事项：以 0.5 为例，输入时，需输入 $S=0$ ， $S+1=0$ ， $S+2=16\#5000$ 。
2. 当 S 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误， $SM0=ON$ ，错误码 $SR0=16\#200D$ 。
3. S 操作数指定的正弦值数值只能介-1.0~+1.0 之间，若不在此范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
4. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

5.

FB/FC	指令码		操作数				功能						
FC		BACOS	P	S, D				BCD ACOS 运算					

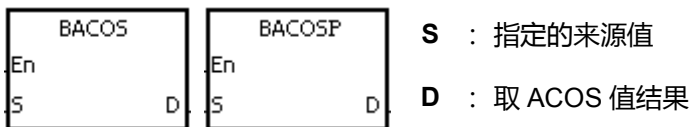
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

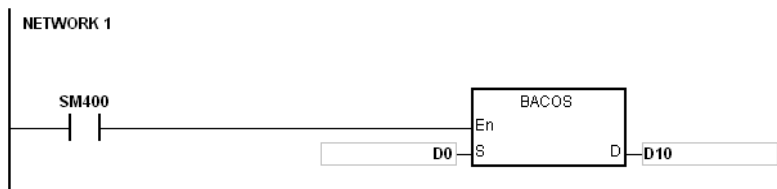


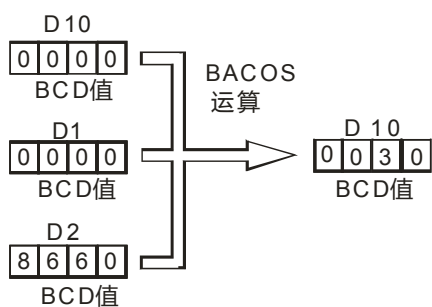
指令说明：

1. **S** 所指定的来源 (BCD)，求取 ACOS 值= \cos^{-1} 值，运算的结果存放于 **D** (角度)。
2. **S** 的内容值代表符号，0=正值，1=负值，**S**+1 代表整数部份，**S**+2 代表小数部份。
3. 运算结果将小数四舍五入。
4. 运算结果为 0°到 180°之间的 BCD 值 (角度)。

程序范例：

D0 内容为 \cos^{-1} 之正负号，D1 内容为 \cos^{-1} 之整数部分，D2 值为 \cos^{-1} 小数部份。经过 BACOS 指令运算后，其结果四舍五入至整数储存于 D10 内。



**补充说明：**

1. 小数部份输入注意事项：以 0.5 为例，输入时，需输入 $S=0$ ， $S+1=0$ ， $S+2=16\#5000$ 。
2. 当 S 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误， $SM0=ON$ ，错误码 $SR0=16\#200D$ 。
3. S 操作数指定的正弦值数值只能介于 -1.0~+1.0 之间，若不在此范围，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
4. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

FB/FC	指令码		操作数				功能					
FC		BATAN	P	S, D				BCD ATAN 运算				

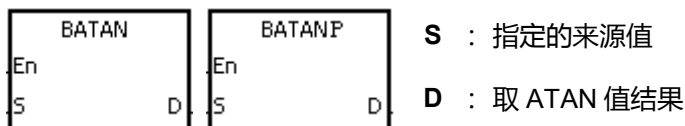
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

图形：

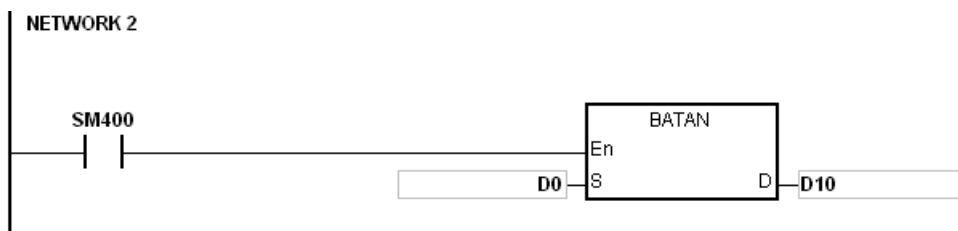


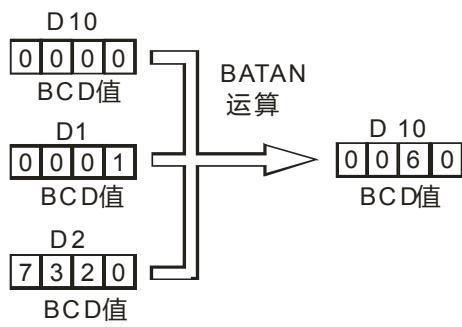
指令说明：

1. **S** 所指定的来源 (BCD)，求取 ATAN 值= \tan^{-1} 值，运算的结果存放于 **D** (角度)。
2. **S** 的内容值代表符号，0=正值，1=负值，**S**+1 代表整数部份，**S**+2 代表小数部份。
3. 运算结果将小数四舍五入。
4. 运算结果为 0°到 90°和 270°到 360°之间的 BCD 值 (角度)。

程序范例：

D0 内容为 \tan^{-1} 的正负号，D1 内容为 \tan^{-1} 的整数部分，D2 值为 \tan^{-1} 小数部份。经过 BATAN 指令运算后，其结果四舍五入至整数储存于 D10 内。



**补充说明：**

1. 小数部份输入注意事项：以 0.5 为例，输入时，需输入 $S=0$ ， $S+1=0$ ， $S+2=16\#5000$ 。
2. 当 S 的数据内容并非为 BCD 值（以 Hex 表示有任一数位不在 0~9 的范围内），则将会产生运算错误， $SM0=ON$ ，错误码 $SR0=16\#200D$ 。
3. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

3.19 万年历指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>TRD</u>	-	✓	万年历数据读出	3
FC	<u>IWR</u>	-	✓	万年历数据写入	3
FC	<u>I+</u>	-	✓	万年历数据加法运算	7
FC	<u>I-</u>	-	✓	万年历数据减法运算	7
FC	<u>HOUR</u>	<u>DHOUR</u>	-	运转定时器	7
FC	<u>ICMP</u>	-	✓	万年历数据比较	11
FC	<u>IZCP</u>	-	✓	万年历数据区域比较	9

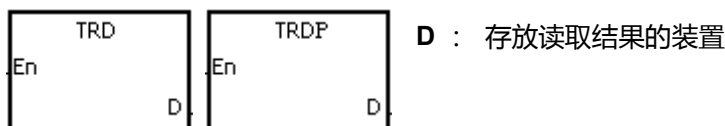
FB/FC	指令码			操作数			功能									
FC		TRD	P	D			万年历数据读出									

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

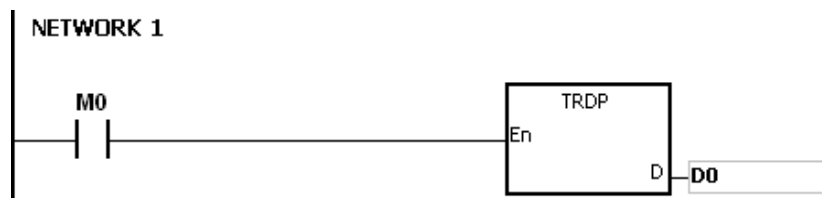


指令说明：

1. D : 万年历当前时间读出后存放的装置。
2. D 操作数会占用连续 7 个装置。
3. 主机内建万年历时钟可提供年、星期、月、日、时、分、秒及共 7 组数据存放于 SR391~SR397 当中，TRD 指令的功能就是让程序设计者直接将万年历当前时间读出至指定的 7 个寄存器当中。
4. SR391 只读取公元年份的右 2 位。

程序范例：

当 M0=ON 时，将万年历当前时间读出至指定的 D0~D6 寄存器当中，SR397 的内容 1 表星期一、2 表星期二，类推，7 表星期日。



SR	项目	内容	→	一般 D	项目
SR391	年 (公元)	00~99	→	D0	年 (公元)
SR392	月	1~12	→	D1	月
SR393	日	1~31	→	D2	日

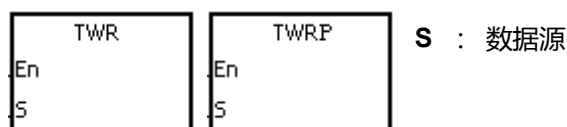
SR	项目	内容		一般 D	项目
SR394	时	0~23	→	D3	时
SR395	分	0~59	→	D4	分
SR396	秒	0~59	→	D5	秒
SR397	星期	1~7	→	D6	星期

补充说明：

1. D+6 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. SM220：±30 秒补正，ON 时作补正。（0~29 秒时归 0，30~59 秒时，分加 1、秒归 0）。
3. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [7] of WORD/INT。

FB/FC	指令码			操作数							功能						
FC		TWR	P	S							万年历数据写入						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
							脉冲执行型			16 位指令				32 位指令			
							AH 运动控制主机			AH 运动控制主机				-			

图形：



指令说明：

1. **S**：存放欲写入万年历新设定值的装置。
2. **S** 操作数会占用连续 7 个装置。
3. 要调整主机内建万年历时钟的时候，可使用本指令将正确的当前时间写入至内藏万年历时钟当中。
4. 本指令被执行时，新的设定时间立刻被写入至 PLC 内部的万年历时钟当中，因此，执行本指令时，请注意所写入的新设定时间与写入当时的当前时间是否吻合。

程序范例：

当 M0=ON 时，将正确的当前时间写入至内藏万年历时钟当中。



一般 D	项目	内容		SR	项目
D20	年 (公元)	00~99	→	SR391	年 (公元)
D21	月	1~12	→	SR392	月
D22	日	1~31	→	SR393	日
D23	时	0~23	→	SR394	时
D24	分	0~59	→	SR395	分
D25	秒	0~59	→	SR396	秒
D26	星期	1~7	→	SR397	星期

新的设定时间

万年历时钟

补充说明：

1. 若 S 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若 S+6 超出装置范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [7] of WORD/INT。

3

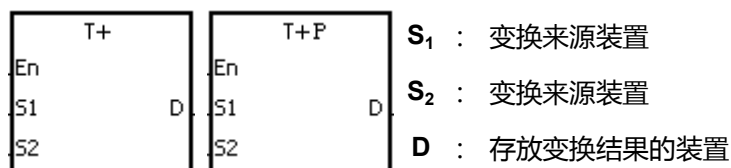
FB/FC	指令码			操作数	功能
FC		T+	P	S ₁ , S ₂ , D	万年历数据加法运算

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

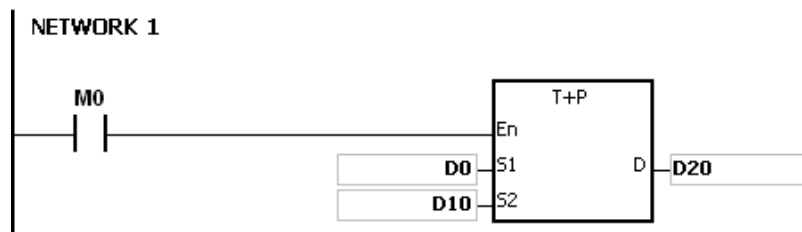


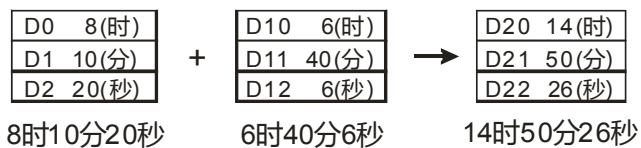
指令说明：

1. 将 S₁ 所指定的万年历数据时、分、秒与 S₂ 所指定的万年历数据时、分、秒相加，所得到的结果存于指定 D 所指定的寄存器时、分、秒当中。
2. S₁、S₂、D 操作数，皆连续占用 3 个装置。
3. 加法运算结果若大于等于 24 小时的话，进位标志 SM602=ON、D 显示加法运算总值减掉 24 小时所得的结果。
4. 加法运算结果若是等于 0 (0 时 0 分 0 秒)，零标志 SM600=ON。

程序范例：

当 M0=ON 时，T+指令执行，将 D0~D2 所指定的万年历数据时、分、秒与 D10~D12 所指定的万年历数据时、分、秒相加，所得到的结果存于 D20~D22 所指定的寄存器中得到加总后的时、分、秒。





补充说明：

1. 若 S_1 、 S_2 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若 S_1+2 、 S_2+2 、 $D+2$ 超出装置范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
4. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
5. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

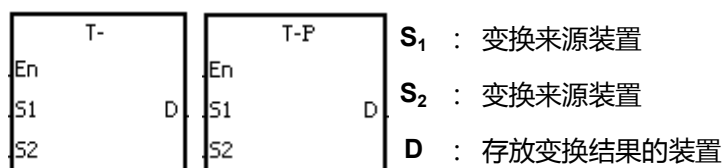
FB/FC	指令码			操作数	功能
FC		T-	P	S ₁ , S ₂ , D	万年历数据减法运算

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

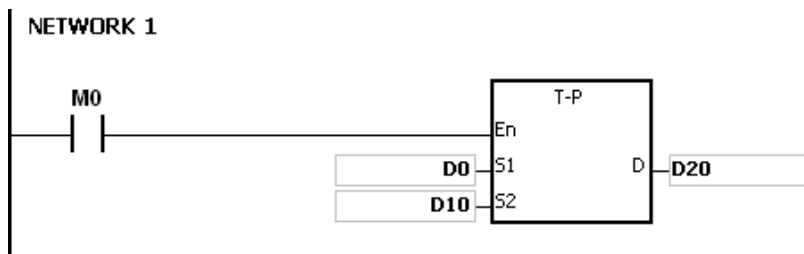
1. 将 S₁ 所指定的万年历数据时、分、秒减掉 S₂ 所指定的万年历数据时、分、秒，所得到的结果存于指定 D 所指定的寄存器时、分、秒当中。
2. S₁、S₂、D 操作数，皆连续占用 3 个装置。
3. 减法运算结果若为负数时，借位标号 SM601=ON、该负数再加上 24 小时所得的结果显示 D 所指定的寄存器当中。
4. 减法运算结果若是等于 0 (0 时 0 分 0 秒)，零标志 SM600=ON。

程序范例：

1. 当 M0=ON 时，TADD 指令执行，将 D0~D2 所指定的万年历数据时、分、秒与 D10~D12 所指定的万年历数据时、分、秒相减，所得到的结果存于指定 D20~D22 所指定的寄存器时、分、秒当中。

D0 20(时)	-	D10 14(时)	→	D20 5(时)
D1 20(分)		D11 30(分)		D21 49(分)
D2 5(秒)		D12 8(秒)		D22 57(秒)
20时20分5秒		14时30分8秒		5时49分57秒

2. 减法运算结果若为负数时，借位标号 SM601=ON。



补充说明：

1. 若 S_1 、 S_2 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若 S_1+2 、 S_2+2 、 $D+2$ 超出装置范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
4. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
5. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

3

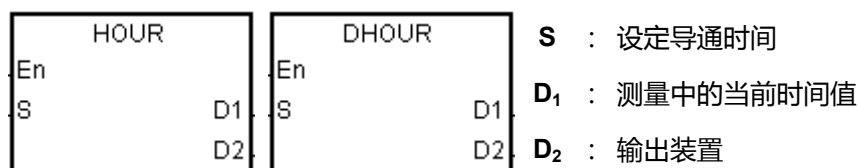
FB/FC	指令码			操作数			功能					
FC	D*	HOUR		S, D ₁ , D ₂			运转定时器					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D ₁		●	●*				●	●*						
D ₂	●/●*	●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D ₁	●	●						●	●		●	○	●				
D ₂	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

1. S：设定导通时间，单位：小时。D₁：测量中的当前时间值，单位：小时。D₂：输出装置。
2. S 为设定导通时间，单位：小时。

16 位指令：1~32,767

32 位指令：1~2,147,483,647

3. HOUR：

D₁ 为测量中之当前时间值（0~32,767），单位：小时。

D₁+1 为未满 1 个小时的当前时间值，设定范围 0~3,599，单位：秒。

D₁+2 系统内部使用，指令执行时不可更改，否则会造成时间计算错误。

测量中的当前时间值到达最大数值 32,767 小时、3,599 秒时会停止计时测量，要重新计时须将当前时间值 D₁、D₁+1 清除为 0。

4. D HOUR：

(D₁+1、D₁) 为测量中的当前时间值（0~2,147,483,647），单位：小时，连续占用两个装置。

D₁+2 为未满 1 个小时的当前时间值，设定范围 0~3,599，单位：秒，占用一个装置。

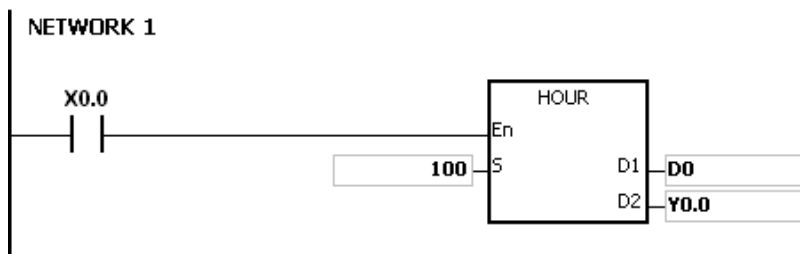
D₁+3 系统内部使用，指令执行时不可更改，否则会造成时间计算错误，占用一个装置。

测量中的当前时间值到达最大数值 2,147,483,647 小时、3,599 秒时会停止计时测量，要重新计时须将当前时间值 D_1 、 D_1+1 、 D_1+2 清除为 0。

5. 将输入接点导通时间做计时，当到达设定时间时（以小时为单位），会将输出装置导通，计数未到达时，输出装置不导通。可提供用户管理机械的运作计时或维修。
6. 当输出装置导通后，定时器会继续计时。
7. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。

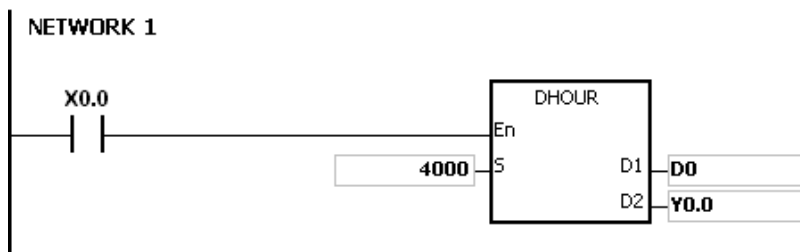
Example 1 :

16 位指令当 X0.0=ON 时，开始计时，当到达 100 小时 Y0.0 导通，而 D0 会记录测量中的当前时间值（单位：小时），D1 会记录测量中不足 1 小时的当前时间值 0~3599（单位：秒），D2 程序保留使用，执行中不得更改内容值，否则会造成错误。



Example 2 :

32 位指令当 X0.0=ON 时，开始计时，当到达 4000 小时 Y0.0 导通，而 D1、D0 会记录测量中的当前时间值（单位：小时），D2 会记录测量中不足 1 小时的当前时间值 0~3599（单位：秒），D3 程序保留使用，执行中不得更改内容值，否则会造成错误。



补充说明 :

1. $S \leq 0$ 时指令不执行，输出装置状态不变。
2. HOUR 中 $D_1 < 0$ 时指令不执行，输出装置状态不变。
3. HOUR 中 D_1+2 超出装置范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. DHOUR 中 $(D_1+1, D_1) < 0$ 时指令不执行，输出装置状态不变。
5. DHOUR 中 D_1+3 超出装置范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
6. 16 位指令 D_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
7. 32 位指令 D_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of DWORD/DINT。

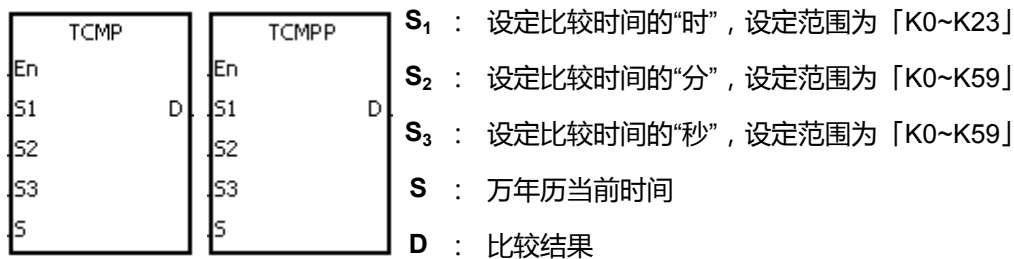
FB/FC	指令码			操作数							功能						
FC		TCMP	P	S_1, S_2, S_3, S, D							万年历数据比较						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1, S_2, S_3		●					●							
S		●					●							
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1, S_2, S_3	●	●			●	●		●	●		●	○	●	○	○		
S	●	●			●	●		●	●		●	○	●				
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

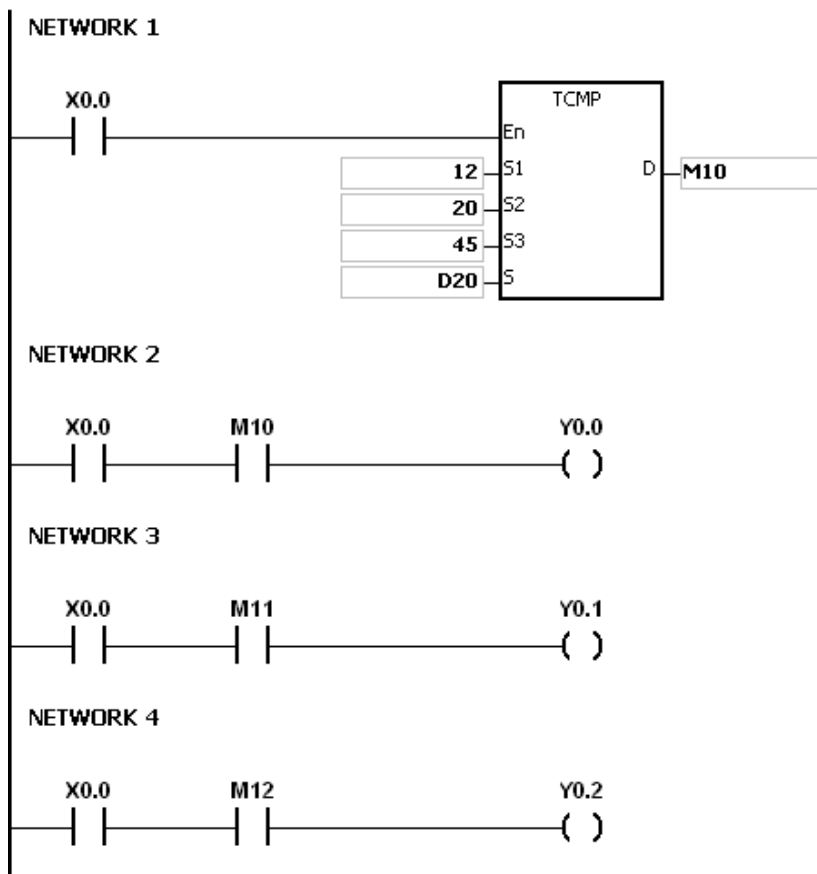


指令说明：

1. 将由 $S_1 \sim S_3$ 所指定的时、分、秒设定值与 S 起始的万年历时、分、秒当前值做比较，其比较结果在 D 作表示。
2. S 为万年历当前时间之“时”，内容为 0~23。 $S+1$ 为万年历当前时间之“分”，内容为 0~59。 $S+2$ 为万年历当前时间之“秒”，内容为 0~59。
3. 操作数 D 占用 3 个连续装置。 $D, D+1, D+2$ 用于储存比较结果。
4. 通常 S 所指定的万年历当前时间通常是预先使用 TRD 指令将万年历当前时间读入后再使用 TCMP 指令进行比较。
5. 设定值 ($S_1 \sim S_3$) > S 当前值 $D=ON, D+1=OFF, D+2=OFF$ 。
6. 设定值 ($S_1 \sim S_3$) = S 当前值 $D=OFF, D+1=ON, D+2=OFF$ 。
7. 设定值 ($S_1 \sim S_3$) < S 当前值 $D=OFF, D+1=OFF, D+2=ON$ 。

程序范例：

1. 当 X0.0=ON 时，指令执行，将 D20~D22 万年历当前时间与设定值 12 时 20 分 45 秒做比较，将结果显示到 M10~M12。当 X0.0 由 ON→OFF 变化时，指令不执行，但是 M10~M12 之前的 ON/OFF 状态仍被保持住。
2. 若需要得到 \geq 、 \leq 、 $\#$ 的结果时，可将 M10~M12 串并联即可取得。

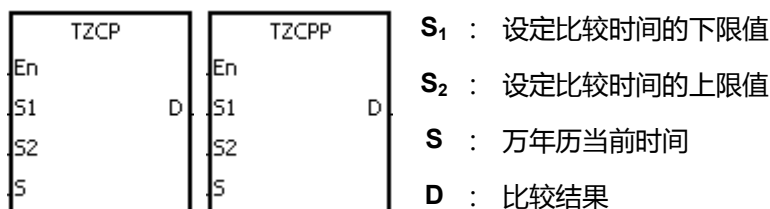


补充说明：

1. S+2 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. D+2 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 若 S 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. 若 S₁~S₃ 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
5. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of BOOL。

FB/FC	指令码			操作数							功能												
FC		TZCP	P	S₁, S₂, S, D							万年历数据区域比较												
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING									
S ₁ , S ₂ , S		●					●																
D	●																						
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF						
S ₁ , S ₂ , S	●	●			●	●		●	●		●	○	●										
D	●	●	●	●				●	●	●			●										
							脉冲执行型	16 位指令					32 位指令										
							AH 运动控制主机							AH 运动控制主机					-				

图形：

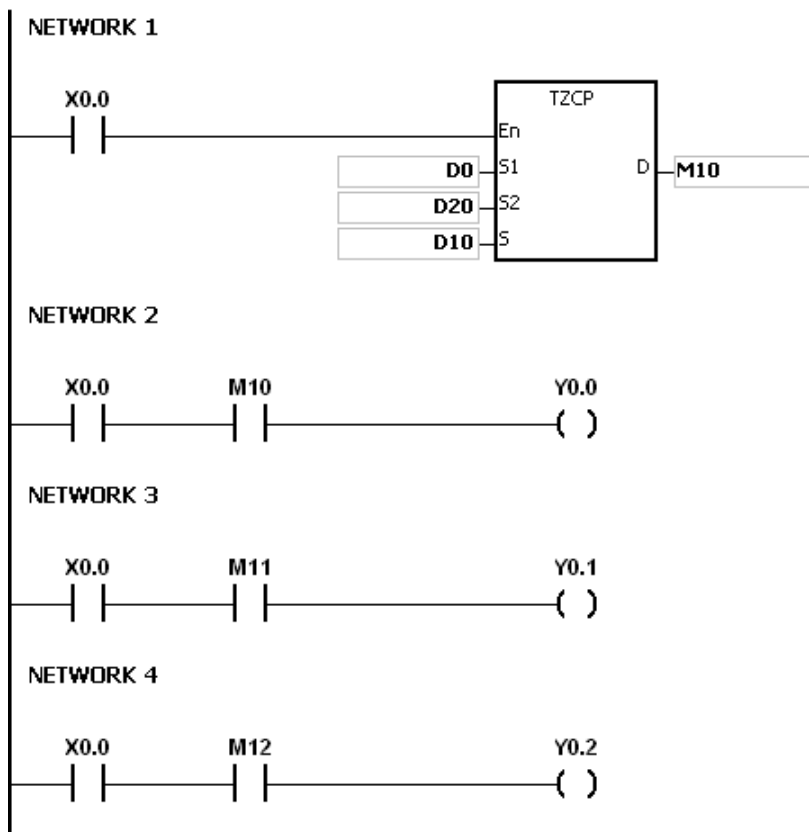


指令说明：

1. 将由 **S** 所指定的万年历当前时间时、分、秒值与 **S₁** 所指定设定比较时间的下限值及 **S₂** 所指定设定比较时间的上限值做区域比较，其比较结果在 **D** 作表示。
2. **S₁**、**S₁+1**、**S₁+2**：设定比较时间下限值的“时”、“分”、“秒”。
3. **S₂**、**S₂+1**、**S₂+2**：设定比较时间上限值的“时”、“分”、“秒”。
4. **S**、**S+1**、**S+2**：为万年历当前时间的“时”、“分”、“秒”。
5. 操作数 **S₁** 时间须比 **S₂** 时间小。当 **S₁>S₂**，指令执行时把 **S₁** 时间作为上/下限值进行比较。
6. 通常 **S** 所指定的万年历当前时间通常是预先使用 TRD 指令将万年历当前时间读入后再使用 TZCP 指令进行比较。
7. 若当前时间 **S** 小于下限值 **S₁** 且 **S** 小于上限值 **S₂** 时，则 **D** 为 ON，若当前时间 **S** 大于下限值 **S₁** 且 **S** 大于上限值 **S₂** 时，则 **D+2** 为 ON，其余状态则 **D+1** 为 ON。

程序范例：

当 X0.0=ON 时，TZCP 指令执行，M10~M12 其中的一会 ON，当 X0.0=OFF 时，TZCP 指令不执行，M10~M12 状态保持在 X0.0=OFF 之前的状态。



补充说明：

1. S_1+2 、 S_2+2 、 $S+2$ 、 $D+2$ 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. S_1 、 S_2 、 S 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
4. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
5. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
6. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

3.20 外围设备指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>TKY</u>	<u>DTKY</u>	-	10 键键盘输入	7
FC	<u>HKY</u>	<u>DHKY</u>	-	16 键键盘输入	9
FC	<u>DSW</u>	-	-	指拨开关输入	9
FC	<u>ARWS</u>	-	-	箭头键盘输入	9
FC	<u>SEGL</u>	-	-	七段显示器扫描输出	7

FB/FC	指令码		操作数			功能				
FC	D*	TKY	S, D ₁ , D ₂			10 键键盘输入				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●/●*													
D ₁		●	●*				●	●*						
D ₂	●/●*													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●	●	●				●	●				●				
D ₁	●	●			●	●		●	●		●	○	●				
D ₂		●	●	●				●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	AH 运动控制主机

图形：



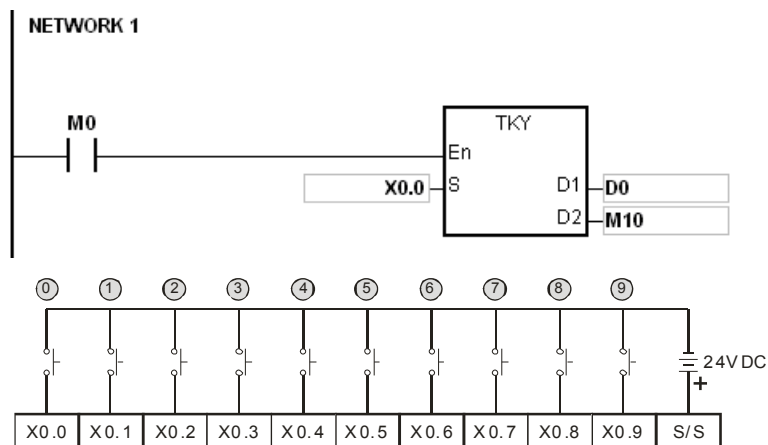
S : 击键起始装置
 D₁ : 击键值存放处
 D₂ : 按键输出信号

指令说明：

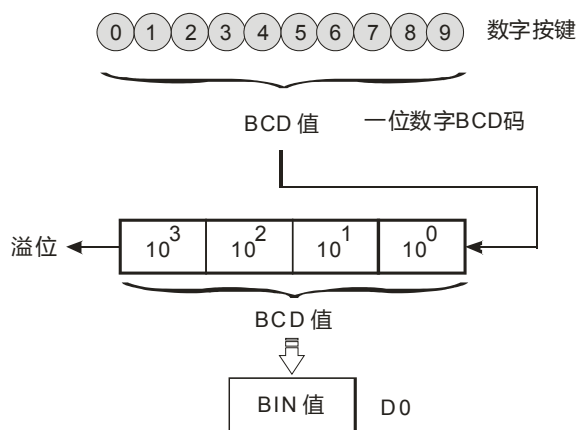
1. 本指令指定由 S 开始的 10 个外部输入点，依序代表 10 进位数字的 0~9。这 10 个外部输入点分别接上 10 个按键，依据这 10 个按键被压下的先后顺序可输入 4 位 10 进位数字 0~9,999 (16 位指令)，或 8 位 10 进位数字 0~99,999,999 (32 位指令)，并将输入的数值存放在 D₁，而 D₂ 则存放键盘的按键情形。
2. S 连续占用 10 个 bit。
3. D₂ 连续占用 11 个 bit，指令执行中，请勿改变 bit 的状态。
4. 当条件接点未启动时，D₂ 开始的连续 11 个 bit 输出点会保持 OFF。
5. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。

程序范例：

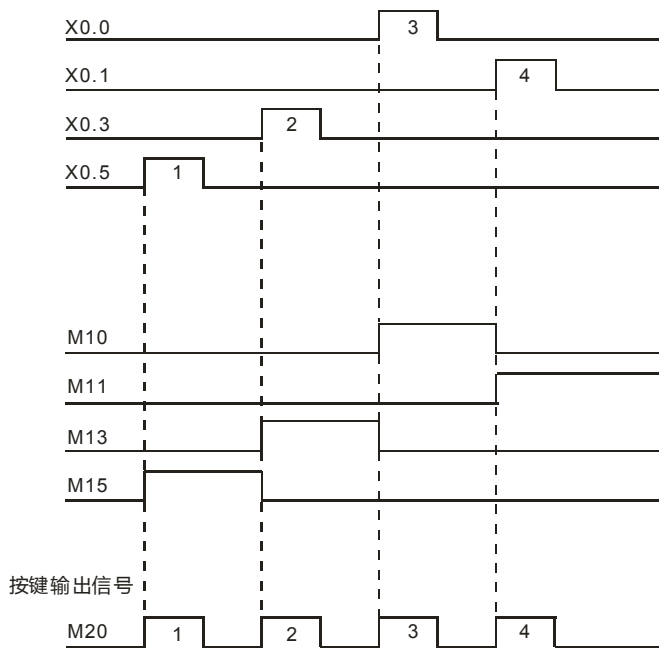
1. 指令指定 X0.0 开始的 10 个输入端与 0~9 的 10 个按键连接，当 M0=ON 时，指令执行，将键盘输入的数值以 BIN 值的形态存入 D0 中，而按键的情况则放在 M10~M19。



注：此范例使用AH16AM10N数位输入模块



2. 如下列时序图所示，连接在数字键盘 X0.5、X0.3、X0.0、X0.1 的 4 个按键以①、②、③、④ 的顺序作打入的动作，结果为 5,301 被暂存在 D0 当中，D0 最大可容纳 9,999，超过 4 位数时，最前面的位数溢位。
3. X0.2 被按下后，至别的按键被按之前，M12=ON，其它的数按键也相同。
4. 当 X0.0~X0.9 当中任何一个按键被按下时，M10~M19 当中一点对应 ON。
5. 任何一个按键被按下时，M20=ON。
6. 当条件接点 M0 变成 OFF 时，D0 之前的值无变化，但是，M10~M20 全部变成 OFF。



补充说明：

1. **S** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [10] of BOOL。
2. **D₂** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [11] of BOOL。

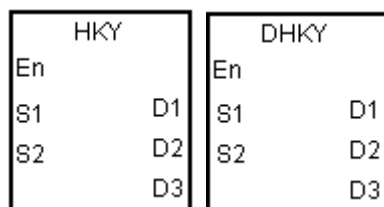
FB/FC	指令码			操作数				功能						
FC	D*	HKY		S, D ₁ , D ₂ , D ₃				16 键键盘输入						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁	●/●*													
S ₂		●/●*					●/●*							
D ₁	●/●*													
D ₂			●*				●	●*						
D ₃	●/●*													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●																
S ₂	●	●			●	●		●	●				●				
D ₁		●															
D ₂	●	●			●	●		●	●		●	○	●				
D ₃		●	●	●				●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	AH 运动控制主机

图形：



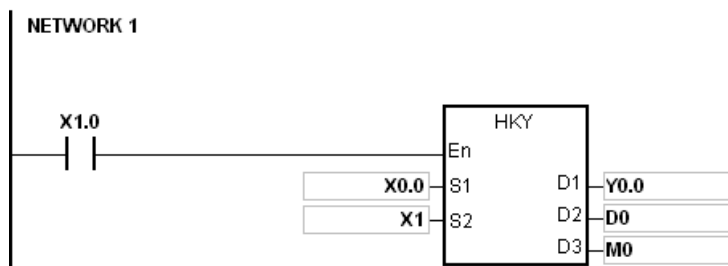
- S₁ : 按键扫描输入起始装置
- S₂ : 系统内部使用
- D₁ : 按键扫描输出起始装置
- D₂ : 击键值存放处
- D₃ : 按键输出信号

指令说明：

- 本指令指定由 S 开始的连续 4 个外部输入点及由 D₁ 开始的连续 4 个外部输出点以矩阵扫描的方式构成 16 键的键盘。键盘输入的数值存放在 D₂，而 D₃ 则存放键盘的按键情形，如果有数个按键同时被按下时，以 S 的编号小的为优先。
- 由数字键盘所打入的值被暂存于 D₂ 当中，使用 16 位指令 HKY 时，D₂ 最大可容纳 9,999，超过 4 位数时，最前面的位数溢位。使用 32 位指令 DHKY 时，D₂ 最大可容纳 99,999,999，超过 8 位数时，最前面的位数溢位。
- 指令执行完毕 SM692 会 ON；每执行完一次矩阵扫描，SM692 会 ON 一个扫描周期。

程序范例：

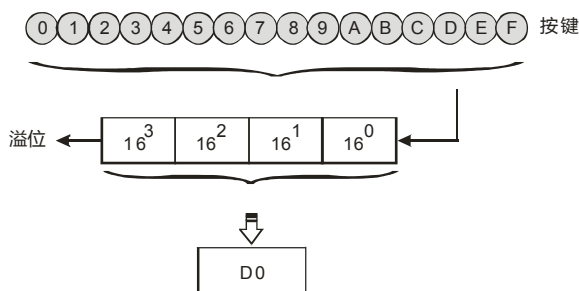
1. 指令指定 X0.0~X0.3 等 4 个输入端与 Y0.0~Y0.3 等 4 个输入端构成扫描 16 键的键盘。当 X1.0=ON 时，指令执行，由键盘输入的数值以 BIN 值的形态存入 D0 中，而按键的情况则放在 M0~M7。



2. 标志 SM691 的功能：

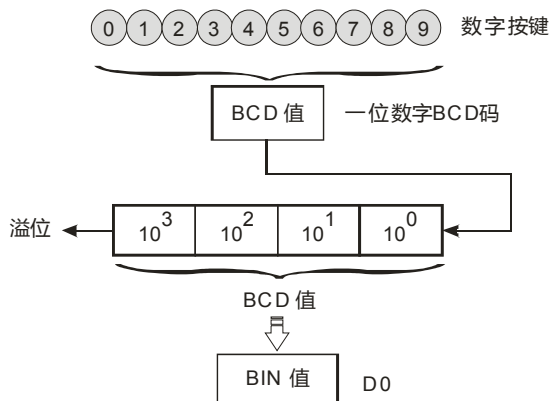
- SM691=ON 时，则 HKY 指令可以输入 0~F 的 16 进位数值。

数字输入：



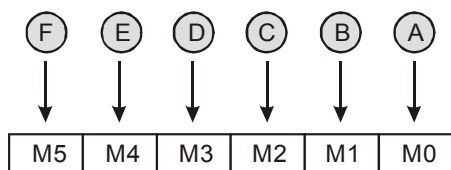
- SM691=OFF 时，则 HKY 指令 A~F 当成功能键使用。

数字输入：



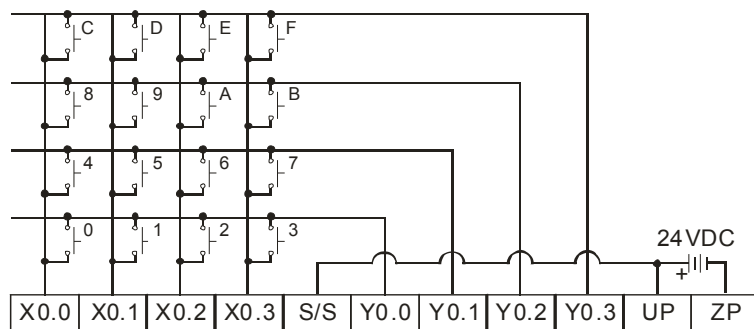
功能键输入：

- 按 A 键时，M0=ON 并保持，接着再按 D 键时，M0 变成 OFF、M3=ON 并保持。
- 复数个按键同时按，以先按者优先。



3. 按键输出信号：

- A~F 当中任何一个按键被按时，M6=ON 一次。
 - 0~9 当中任何一个按键被按时，M7=ON 一次。
4. 当条件接点 X1.0 变成 OFF 时，D0 之前的输入值无变化，但是 M0~M7 全部变成 OFF。
 5. 外部配线：



注：本范例使用AH16AP11T晶体管输出模块

补充说明：

1. 本指令被执行时，当扫描周期太长或太短都可能造成击键不良，因此可运用下列技巧来克服。
 - 当扫描周期太短时，可能造成 I/O 来不及反应而无法读取正确的击键，此时，可将扫描时间加以固定。
 - 当扫描周期太长时，可能会使按键反应变得迟钝，可将此指令写在时间中断工作（TASK）内，固定时间执行此指令。
2. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。
3. D₁ 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。
4. D₃ 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of BOOL。

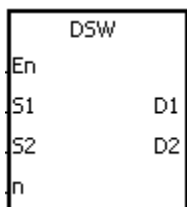
FB/FC	指令码		操作数				功能			
FC	DSW		S, D ₁ , D ₂ , n				指拨开关输入			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁	●													
S ₂		●					●							
D ₁	●													
D ₂		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●																
S ₂	●	●			●	●		●	●				●				
D ₁		●															
D ₂	●	●			●	●		●	●				●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



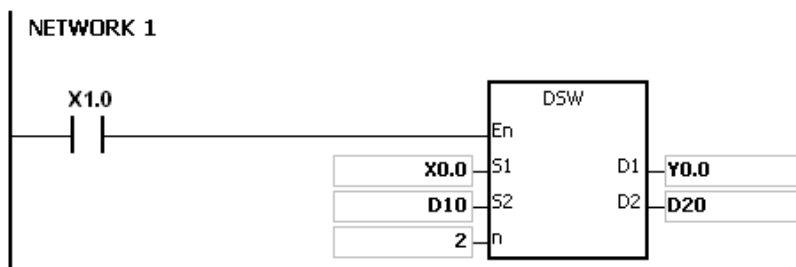
- S₁ : 指拨开关扫描输入起始装置
- S₂ : 系统内部使用
- D₁ : 指拨开关扫描输出起始装置
- D₂ : 指拨开关设定值存放处
- n : 指拨开关所连接的组数

指令说明：

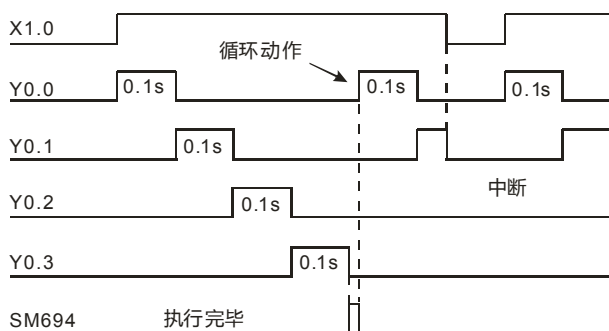
1. 本指令由 S₁ 开始的连续 4 个或 8 个外部输入点及由 D₁ 开始的连续 4 个外部输出点扫描读取 1 组或 2 组 4 位数指拨开关，指拨开关设定值存放在 D₂，由 n 决定读取 4 位数指拨开关有 1 组或 2 组。
2. 当 n=1 时，D2 操作数占用一个寄存器。n=2 时，D2 操作数会连续占用 2 个寄存器。
3. S₂、S₂₊₁ 占用两个装置给系统内部使用，用户请勿变更内容值。
4. 每循环扫描一次完毕标志信号 SM694=ON 一个扫描周期。
5. 当条件接点未启动时，D₁ 开始的连续 4 个外部输出点会保持 OFF。
6. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。

程序范例：

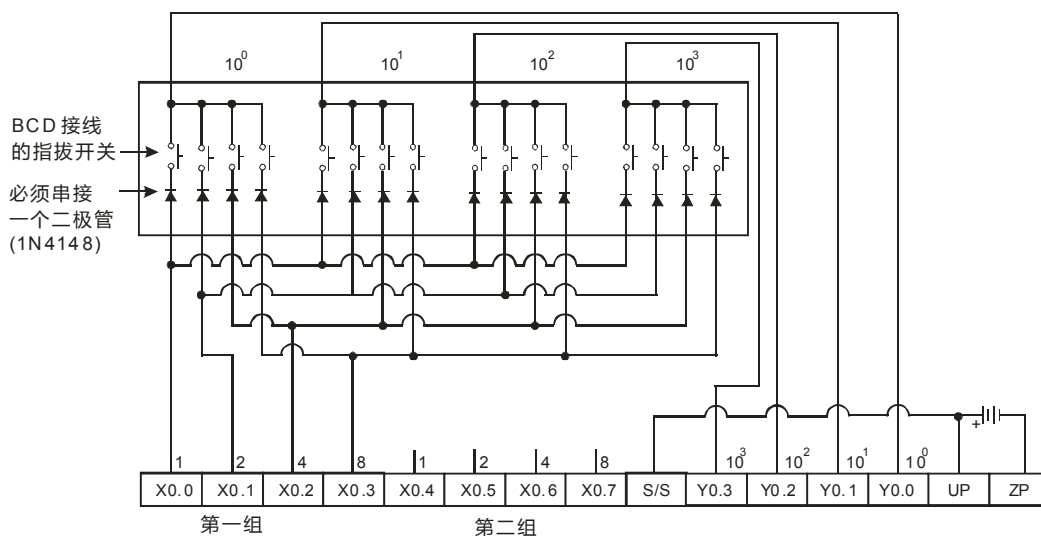
1. 由 X0.0~X0.3 及 Y0.0~Y0.3 组成第一组指拨开关回路，由 X0.4~X0.7 及 Y0.0~Y0.3 组成第二组指拨开关回路。当 X1.0=ON 时，指令开始执行，第一组指拨开关的设定值被读入并转换成 BIN 值后存放至 D20 中，第二组指拨开关的设定值被读入并转换成 BIN 值后存放至 D21 中。



2. 当 X1.0=ON 时，Y0.0~Y0.3 自动循环扫描 ON，每循环一次，执行完毕标志信号 SM694=ON 一个扫描周期。
3. 扫描用输出端 Y0.0~Y0.3 请使用晶体管输出。



4. 指拨开关输入



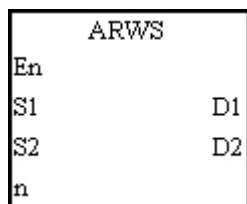
注：此范例使用AH16AP11T晶体管模块

补充说明：

1. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. D_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。

FB/FC	指令码			操作数							功能							
FC	ARWS			S_1, S_2, D_1, D_2, n							箭头键盘输入							
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING				
S_1	●																	
S_2		●					●											
D_1		●					●											
D_2	●																	
n		●					●											
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S_1	●	●	●	●				●	●				●					
S_2	●	●			●	●		●	●				●					
D_1	●	●			●	●		●	●			○	●					
D_2		●																
n	●	●						●	●		●		●	○	○			
							脉冲执行型				16 位指令				32 位指令			
							-				AH 运动控制主机				-			

图形：

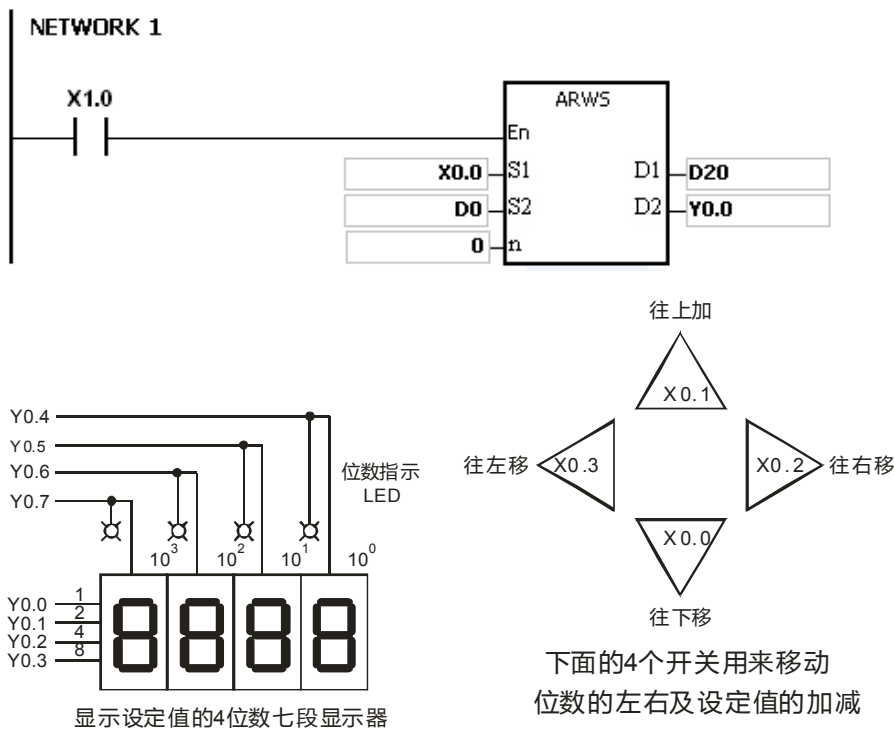
 S_1 ：击键起始装置 S_2 ：系统内部使用 D_1 ：欲显示在七段显示器的装置 D_2 ：七段显示器扫描输出起始装置 n ：输出信号及扫描信号的极性指示

指令说明：

1. 本指令执行， S_1 定义为下键， S_1+1 定义为上键， S_1+2 定义为右键， S_1+3 定义为左键，利用上下左右键来执行外部设定值的操作及显示。将设定值存放于 D_1 当中，设定值范围：0~9,999。
2. S_1 连续占用 4 个 bit 装置。
3. S_2 系统内部使用，用户请勿变更内容值。
4. D_2 连续占用 8 个 bit 装置。
5. 条件接点未启点时， D_2 连续的 8 个 bit 装置保持 OFF。
6. n 的范围：0~3，功能请参考 SEGL 的补充说明。
7. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。

程序范例：

1. 本指令执行，X0.0 定义为下键，X0.1 定义为上键，X0.2 定义为右键，X0.3 定义为左键，利用上下左右键来执行外部设定值的操作及显示。将设定值存放于 D20 当中，设定值范围：0~9,999。
2. 当 X1.0=ON 时，位数 10^3 为有效设定位数，如果按左按键时，则有效设定位数呈现 $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$ 的方向循环跳动。
3. 如果按右移按键，则有效设定位呈现 $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3 \rightarrow 10^2$ 的方向循环跳动。在循环的同时，由 Y0.4~Y0.7 所连接的位数指示灯也循环 ON 作有效设定位数的指示。
4. 如果按往上加按键时，则有效设定位数的内容由 $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 8 \rightarrow 9 \rightarrow 0 \rightarrow 1$ 作变化。如果按往下减按键时，则有效设定位数的内容由 $0 \rightarrow 9 \rightarrow 8 \rightarrow \dots \rightarrow 1 \rightarrow 0 \rightarrow 9$ 作变化，同时，变化值也被显示在七段显示器上。



补充说明：

1. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. S₁ 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。
3. D₂ 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of BOOL。

FB/FC	指令码		操作数				功能						
FC		SEGL	S, D, n				7 段显示器扫描输出						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							
D	●													
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●	○	●				
S ₂	●	●			●	●		●	●				●				
D		●															
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



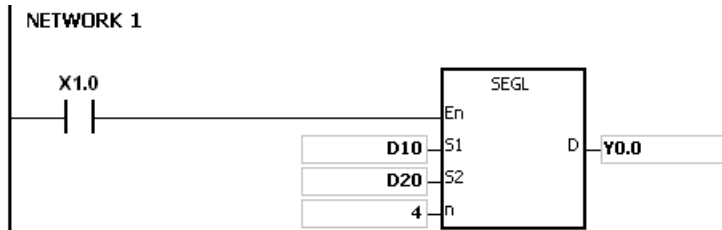
- S₁ : 欲显示七段显示器的来源装置
 S₂ : 系统内部使用
 D : 七段显示器扫描输出起始装置
 n : 输出信号及扫描信号的正负逻辑设定

指令说明：

1. 本指令占用 D 开始的连续 8 个或 12 个外部输出点,作为 1 组或 2 组 4 位数七段显示器的显示数据及扫描信号输出。每个位数均带有 7 段显示器驱动器,该驱动器是将输入的 BCD 码转换 7 段显示器的驱动信号;驱动器并带有栓锁控制信号,可将 7 段显示器显示保持。
2. S₁ 中之数值为欲显示到第一组七段显示器的数值, S₁₊₁ 中的数值为欲显示到第二组七段显示器的数值。
3. S₂ 系统内部使用,用户请勿变更内容值。
4. n=0~7。请参考补充说明。
5. 由 n 决定扫描输出 4 位数七段显示器有 1 组或 2 组,且 n 也用来指定 PLC 输出端的正负逻辑输出。
6. 4 位数 1 组时,占用输出点 8 个,4 位数 2 组时,占用输出点 12 个。
7. 本指令执行时,扫描输出端顺序循环动作,指令执行中条件接点变成 OFF 再 ON 时,扫描输出端重新执行。
8. 每循环扫描一次完毕标志信号 SM693=ON 一个扫描周期。

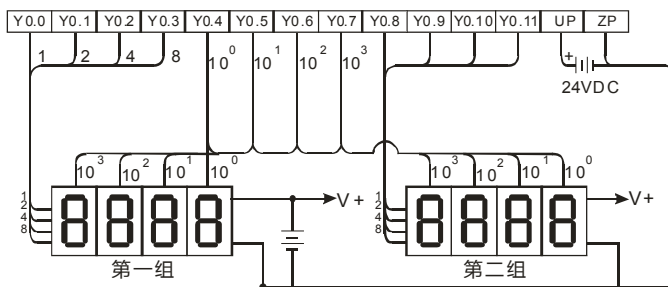
程序范例：

1. 当 X1.0=ON 时，指令开始执行，由 Y0.0~Y0.7 构成七段显示器扫描回路，D10 中的数值被转换成 BCD 码后送到第一组七段显示器显示出来，D11 中的数值被转换成 BCD 码后送到第二组七段显示器显示出来，若 D10 或 D11 中的数值超过 9,999 将发生运算错误。



2. 当 X1.0=ON 时，Y0.4~Y0.7 会自动循环扫描，每循环扫描一次需 12 个扫描时间，每循环扫描一次完毕标志信号 SM693=ON 一个扫描周期。
3. 一组 4 位数的时候 n=0~3。
 - 将已经译码的 7 段显示模块 1、2、4、8 各端各自并接后连接至 PLC 的 Y0.0~Y0.3，而各位数的 Latch 端单独连接至 PLC 的 Y0.4~Y0.7。
 - 当 X1.0=ON 时，指令被执行，D10 的内容随着 Y0.4~Y0.7 的循环动作被顺序传送到七段显示器作显示。
4. 二组 4 位数的时候 n=4~7。
 - 将已经解码的七段显示器 1、2、4、8 各端各自并接后连接至 PLC 的 Y0.8~Y0.11，而各位数的 Latch 端与第一组共享 Y0.4~Y0.7。
 - D10 的内容被传送到第一组七段显示器上、D11 的内容被传送到第二组七段显示器作显示。若 D10=1234，D11=4321，则第一组将会显示 1 2 3 4，第二组显示 4 3 2 1。

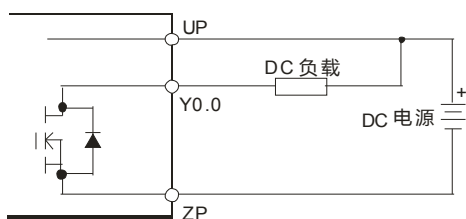
5. 七段显示器扫描输出接线图：



注：此范例使用AH16AN01T晶体管输出模块

补充说明：

1. n 的设定值：是用来设定晶体管输出为正极性或负极性回路，连接的七段显示器是一组 4 位数或者是二组 4 位数。
2. PLC 输出点必须选用为晶体管模块，输出为 NPN 型式，采用集极式输出，在电路的连接上，输出必须连接一提升电阻至 DC 电源（小于 30VDC），因此当输出点 Y 导通时，信号输出为低电位。



- BCD 码正逻辑（负极性）输出

BCD 数值				Y 输出 (BCD 码)				信号输出			
b ₃	b ₂	b ₁	b ₀	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

- BCD 码负逻辑（正极性）输出

BCD 数值				Y 输出 (BCD 码)				信号输出			
b ₃	b ₂	b ₁	b ₀	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	1	0	1	1	0	1	0	0	1

- 显示扫描栓锁 (Latch) 信号

正逻辑（负极性）		负逻辑（正极性）	
Y 输出 (Latch)	输出控制信号	Y 输出 (Latch)	输出控制信号
1	0	0	1

- 参数 n 的设定值

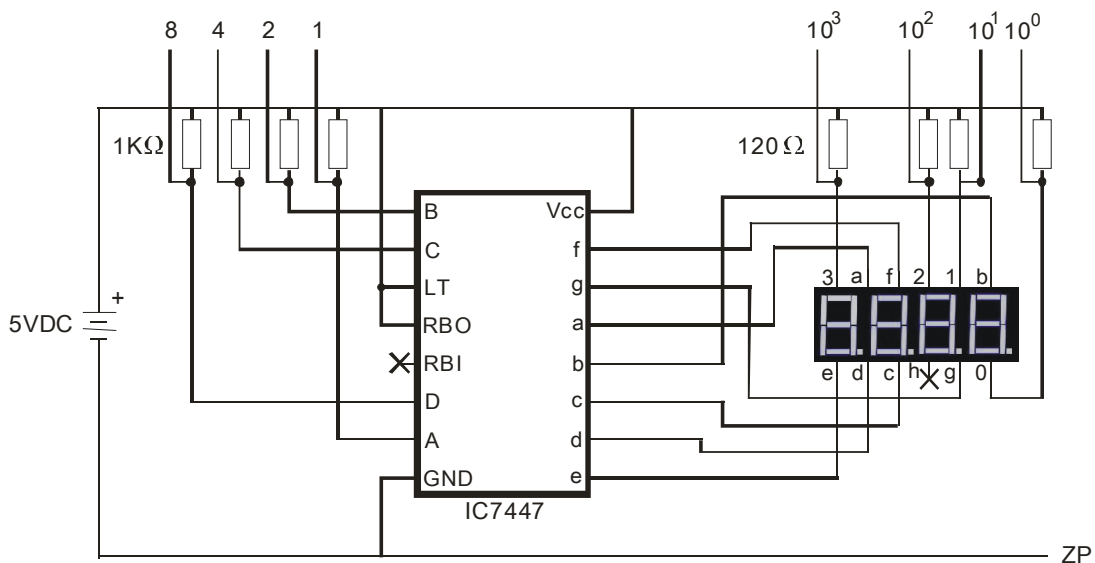
7 段显示器组数	一组				二组			
BCD 码数据 Y 输出	+		-		+		-	
显示扫描栓锁信号	+	-	+	-	+	-	+	-
n	0	1	2	3	4	5	6	7

‘+’：正逻辑（负极性）输出

‘-’：反逻辑（正极性）输出

- ◆ PLC 的晶体管输出极性与 7 段显示器的输入极性是否相同或者是不同时，可通过可参数 n 的设定值来相互匹配。

◆ 采用共阳极四位数字 7 段显示器搭配 IC7447 配置如下图



3

3.21 通讯指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>RS</u>	—	—	串行数据传输	9
FC	<u>LRC</u>	—	✓	和检查 LRC 模式	7
FC	<u>CRC</u>	—	✓	和检查 CRC 模式	7
FC	<u>MODRW</u>	—	—	MODBUS 数据读写	11
FC	<u>COMRS</u>	—	—	通讯传送与接收指令	11

FB/FC	指令码		操作数				功能			
FC	RS		S, m, D, n				传送用户自行定义的通讯命令			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
m		●					●							
D		●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●				●				
m	●	●			●	●		●	●				●	○	○		
D	●	●			●	●		●	●				●				
n	●	●			●	●		●	●				●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



- S : 传送数据的起始装置
- m : 传送数据的笔数
- D : 接收数据的起始装置
- n : 接收数据的笔数

指令说明：

1. 此指令专为主机使用串联式通讯接口 (RS-232/422/485) 所提供的便利指令, 用户只要将其相关参数 (S、m、D、n) 设定完成, 即可发送与接收命令, 若使用 E 修饰装置起始位置时, 请勿在指令执行期间变更 E 的设定值, 否则容易造成数据读取或写入错误。
2. 若仅发送或接收数据, 可将装置长度 (m、n) 设定为 0 即可。
3. 本指令在程序中使用次数并无限制, 但是不同指令 (如 MODRW、FWD...) 在同一时间使用相同的通讯端口时, 仅会只有一个指令被执行。
4. RS 指令传送或接收未完成时, 无法改变传送或接收装置数据内容。
5. 传送或接收装置最大长度 (m、n) 为 500 words。
6. 本指令可以藉由设定特殊寄存器来选择 8 位或 16 位模式 (SM106 或 SM107)。

通讯能力设定

在执行串行通讯命令之前，用户必须先行设定通讯能力（如 RS232/485、传输速率...），设定通讯能力的方式有两种，用户可以在 HWCONFIG 内针对 PLC 通讯端口直接进行设定，或在程序内针对其相对应的特殊辅助继电器来进行设定。

1. 利用 HWCONFIG 来针对通讯端口进行通讯能力设定（请参照 ISPSOft 使用手册）
2. 程序内针对各个特殊寄存器，来进行通讯端口通讯能力设定，其设定方式如范例，寄存器内容格式请参照本章节补充说明。

传输数据格式说明

数据的传送格式分成几个部分 8 位与 16 位模式，8 位模式是将 16 位数据分成上 8 位，下 8 位，上 8 位被省略，仅下 8 位为有效数据可做数据的传送和接收。如模式为 16 位时要特别注意高低位的部分，请参照下表。

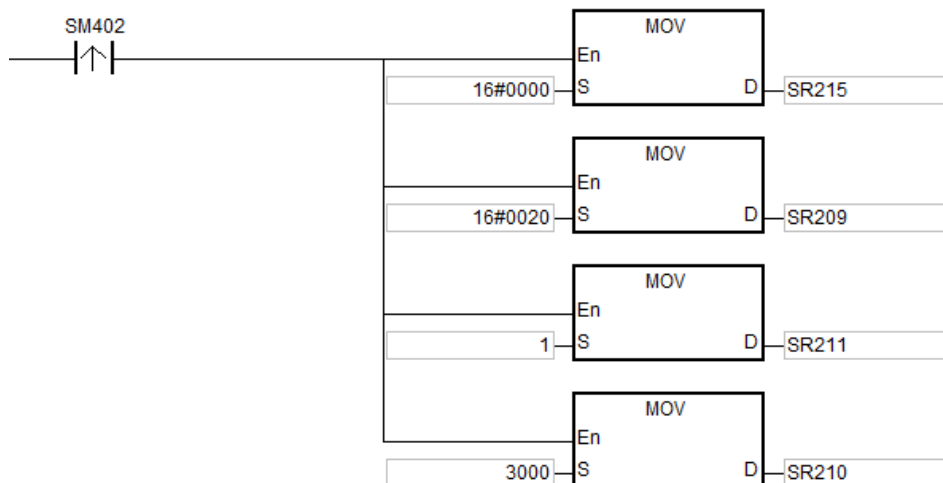
8 位模式，传送数据 0x01234567							
D10 (上)	D10 (下)	D11 (上)	D11 (下)	D12 (上)	D12 (下)	D13 (上)	D13 (下)
*	16#01	*	16#23	*	16#45	*	16#67
16 位模式，传送数据 0x1234567							
D10		D11		D12		D13	
16#2301		16#6745					

通讯协议设定范例：

1. 本范例为将通讯端口的通讯协议设定为 RS232 9600，7，N，1。
2. 设定通讯端口传输方式为 RS232 (SR215=0)。
3. 设定通讯端口其通讯速率与格式为 9600，7，N，1 (SR209=16#0020)。
4. 设定其自动询问次数为 1 (SR211=1)。
5. 设定其逾时时间为 3000ms (SR210=3000)。
6. 设定为 16 位模式 (SM106=OFF)。
7. 通讯端口通讯能力设定生效 (SM209=ON)。

用户也可于 ISPSOft - > HWCONFIG - > COM Port 内直接对通讯端口进行设定，则可以省略此步骤。

Network 1



Network 2



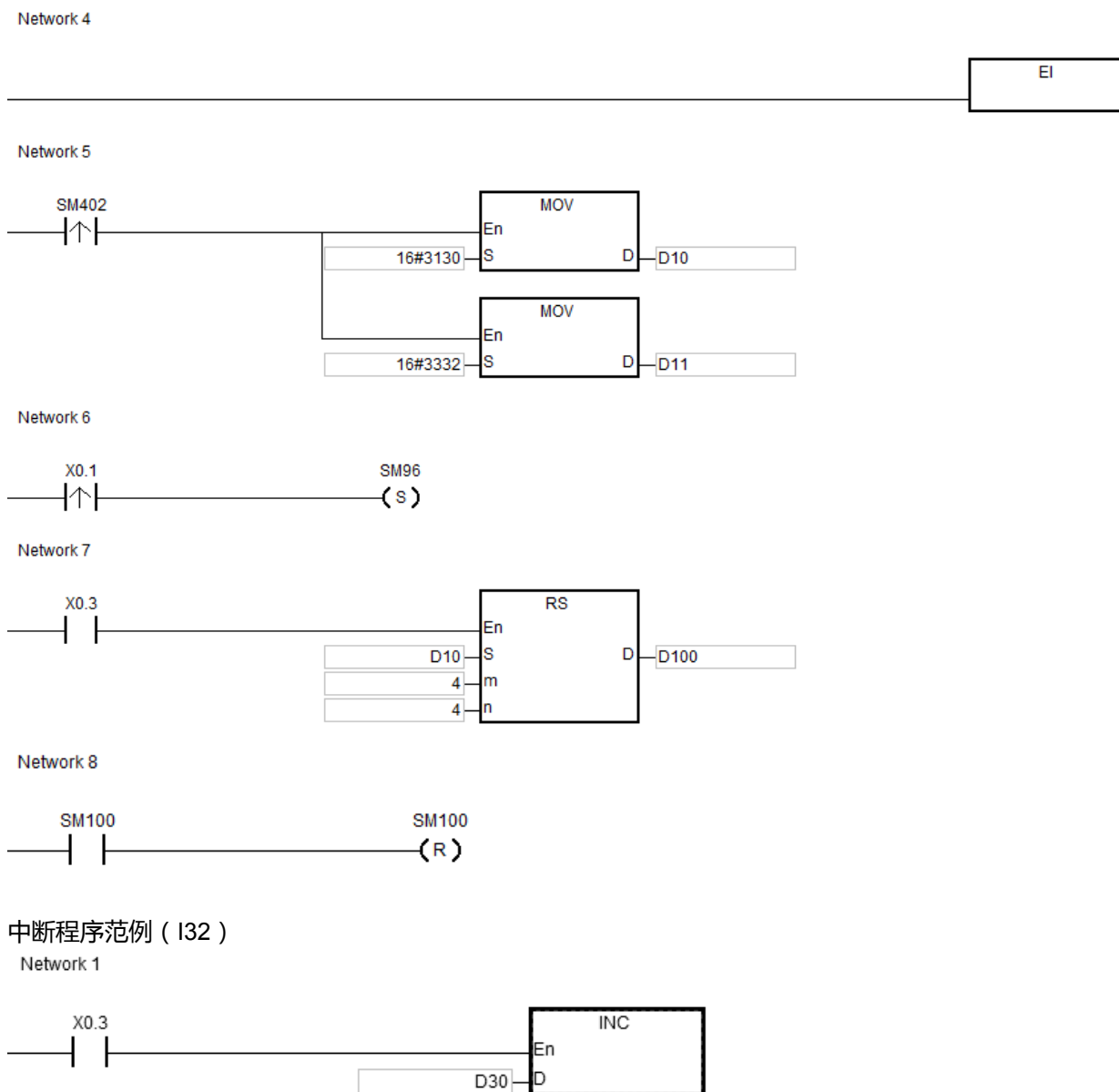
Network 3



程序范例：

1. 利用 HWCONFIG 或是特殊辅助寄存器，来设定通讯端口通讯能力（如设定范例所述）。
2. 先将传送数据内容预先写入 D10 与 D11 寄存器内，再将 SM96（送信要求标志）设为 ON。
3. 当 X0.1 与 X0.3=ON 时，RS 指令执行 PLC 即进入等待传送、接收数据的状态，开始执行后，D10 开始连续传送 n 个 bytes 到外部机器，在传送结束时，SM96 会自动 RESET 成 OFF（请勿利用程序执行 RST SM96），当有数据回复回来的时候，将其存入由 D100 开始的连续寄存器内。
4. 当数据接收完毕标志（SM100）自动 ON，程序中处理完接收数据后，须将 SM100 RESET 为 OFF，再度进入等待传送接收的状态。但请勿利用 PLC 程序连续执行 RST SM100。
5. 当 PLC 接收到的回复命令中，含有特殊字符（SR621）时，将会触发 I32，使得 D30 加一。

范例程序：



补充说明：

1. **M 或 n** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#200B。
2. COM1 与 COM2 通讯 RS / MODRW 指令相关标志信号：

标志信号		功能说明	动作
COM1	COM2		
SM96	SM97	送信要求，当用户要利用 RS/MODRW 指令将数据传送与接收，必须用脉冲指令将 SM96/SM97 设为 ON，若上述指令开始执行，则 PLC 执行数据传送接收的动作。当上述指令执行数据传送完毕后会自动将 SM96/SM97 清除。	用户设定，系统自动清除

标志信号		功能说明	动作
COM1	COM2		
SM98	SM99	接收等待，当 SM98、SM99 为 ON 时，表示 PLC 目前正等待接收数据中。	系统产生以及系统自动清除
SM100	SM101	接收完毕，当 RS/MODRW 指令接收完毕后将 SM100/SM101 设为 ON，用户在程序中可利用 SM100 为 ON 时，处理所接收到的数据。当接收到的数据处理完毕后，必须将 SM100、SM101 清除为 OFF。	系统自动设定，用户清除
SM102	SM103	MODRW、RS...等数据接收错误，错误代码纪录在 Error Log 中。	系统产生以及系统自动清除
SM104	SM105	接收超时，用户若有设定通讯超时（SR210*SR211 或 SR213*SR214 的时间为超时时间），若超出设定值数据尚未接收完毕则会启动此标志。若状态解除后必须将 SM104/SM105 清除为 OFF。	系统自动设定，用户清除
SM106	SM107	8/16 位处理模式选择，ON 为 8 位模式，OFF 为 16 位模式。	用户设定及清除
SM108	SM109	接收结束标志。 ON 当 PLC 在接收的字符时，会去判断是否有通讯中断字符（SR621/SR622 low byte）存在，若有检测到中断字符时，执行通讯中断子程序，执行完后结束接收数据。 OFF：当 PLC 在接收的字符时，会去判断是否有通讯中断字符（SR621/SR622 low byte）存在，若有检测到中断字符时，执行通讯中断子程序，执行完后继续接收数据，直到完成接收（接收数据长度为 n）。（默认值）。	用户设定及清除
SM209	SM211	通讯协议变更用，根据特殊数据寄存器 SR201、SR202、SR209、SR210、SR211、SR212、SR213、SR214、SR215、SR216、SM210、SM212 的设定，作为通讯协议设定的重置。可将 SM209/SM211 设为 ON，此时，COM1 或 COM2 的通讯协议即依 SR201、SR202、SR209、SR210、SR211、SR212、SR213、SR214、SR215、SR216、SM210、SM212 的设定改变，且 SM209/SM211 会自动设定成 OFF。	用户设定，系统自动清除

3. COM1 或 COM2 通讯 RS/MODRW 指令相关设定的特殊数据寄存器：

特 D		功能说明
COM1	COM2	
SR201	SR202	PLC 主机通讯地址，当 PLC 主机当从站时的通讯地址。
SR210	SR213	通讯超时异常，时间定义（ms），设定 Timeout 时间。当设定值大于 0 时，则 RS/MODRW 指令执行，进入接收模式后，若在指定的时间内没有收到第一个字符或任二字符之间的时间超过此设定值，PLC 会将 SM104/SM105 设为 ON，用户可利用

特 D		功能说明
COM1	COM2	
		此标志作通讯逾时的处理。但必须记得：处理完后，必须将 SM104/SM105 清除，此寄存器在 RS 指令可被设为 0，此时逾时侦测功能被取消，但在 MODRW 指令则为限定在 100~65535 (ms) 区间内。
SR621	X	RS 指令特定字符通讯接收中断请求 (I32)，当通讯接收的字符=SR621 的 Low Byte 时，触发中断 I32。当 n=0 时，中断不反应。
X	SR622	RS 指令特定长度通讯接收中断请求 (I33)，当通讯接收的字符=SR622 的 Low Byte 时，触发中断 I33。当 n=0 时，中断不反应。

4. SR215, SR216 用来纪录 PLC COM 通讯接口的代号，代号代表的接口功能如下表：

代号	0	1	2
接口功能	RS232	RS485	RS422

5. SR209、SR212：RS-485/RS-232 通讯协议，其设定方法请参考下表：

b0	数据长度			7 (内容值=0)	8 (内容值=1)
b1 b2	同位			00	无 (None)
				01	奇同位 (Odd)
				10	偶同位 (Even)
b3	stop bits			1 bit (内容值=0)	2 bits (内容值=1)
b4 b5 b6 b7	0001	(16#1)	:	4800	
	0010	(16#2)	:	9600	
	0011	(16#3)	:	19200	
	0100	(16#4)	:	38400	
	0101	(16#5)	:	57600	
	0110	(16#6)	:	115200	
	0111	(16#7)	:	230400	RS-232 不支持
	1000	(16#8)	:	460800	RS-232 不支持
1001	(16#9)	:	921600	RS-232 不支持	
b8~b15	无定义 (保留)				

6. PLC 相关通讯传输速度如下：

速率 (bps)	RTU 逾时时间间隔 (ms)	速率 (bps)	RTU 逾时时间间隔 (ms)
4800	9	115200	1
9600	5	230400	1
19200	3	460800	1
38400	2	921600	1
57600	1		

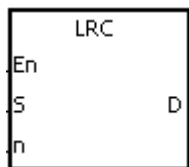
FB/FC	指令码		操作数				功能				
FC	LRC		S, n, D				检查 LRC 模式				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●				●				
n	●	●			●	●		●	●				●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



- S : ASCII 模式和检查运算起始装置
- n : 运算组数
- D : 存放运算结果之起始装置

指令说明：

1. LRC 检查码：请参考补充说明。
2. n : 运算组数须为偶数，范围 1~1000 不在此范围则视为运算错误，指令不执行，SM0、SM1=ON，SR0 记录错误码 16#200B。
3. 16 位转换模式：当 SM606=OFF 时，S 起始装置将其 16 进位数据区分为上 8 位、下 8 位，将每一个位数做 LRC 检查码运算，传送到 D 的上 8 位及下 8 位中，运算的位数以 n 来设定。
4. 8 位转换模式：当 SM606=ON 时，S 起始装置将其 16 进位数据区分为上 8 位（无效数据）、下 8 位，将每一个位数做 LRC 检查码运算，传送到 D 的下 8 位中占用 2 个寄存器，运算的位数以 n 来设定。（D 的上 8 位全部为 0）

程序范例：

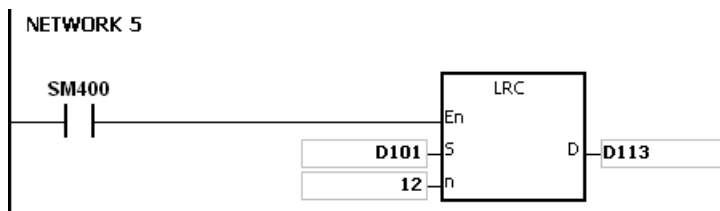
1. PLC 与 VFD-S 系列交流电机驱动器联机（ASCII 模式，SM210 =OFF）、（8 位模式，SM606=ON），传送数据预先写入读取 VFD-S 参数地址 16#2101 开始的 6 笔数据。

PLC⇒VFD-S，PLC 传送：“：01 03 2101 0006 D4 CR LF”

PLC 传送数据寄存器（PLC 传送信息）

寄存器	DATA		说明	
D100 下	':'	16#3A	STX	
D101 下	'0'	16#30	ADR 1	ADR (10) 为交流电机驱动器地址
D102 下	'1'	16#31	ADR 0	
D103 下	'0'	16#30	CMD 1	CMD (10) 为命令码
D104 下	'3'	16#33	CMD 0	
D105 下	'2'	16#32	起始数据地址 Starting Data Address	
D106 下	'1'	16#31		
D107 下	'0'	16#30		
D108 下	'1'	16#31		
D109 下	'0'	16#30	数据 (word) 个数 Number of Data (count by word)	
D110 下	'0'	16#30		
D111 下	'0'	16#30		
D112 下	'6'	16#36		
D113 下	'D'	16#44	LRC CHK 0	LRC CHK (01) 为错误检查码
D114 下	'4'	16#34	LRC CHK 1	
D115 下	CR	16#0D	END	
D116 下	LF	16#0A		

上列 LRC CHK (0 , 1) 为错误检查码可由指令 LRC 算出 (8 位 Mode , SM606=ON)



LRC 检查码 : $16\#01+16\#03+16\#21+16\#01+16\#00+16\#06=16\#2C$, 然后取 2 的补码= $16\#D4$ 。此时 , 'D' ($16\#44$) 存于 D113 下 8 位内 , '4' ($16\#34$) 存于 D114 下 8 位内。

补充说明 :

1. 有一通讯数据 ASCII 模式 , 格式如下 :

STX	':'	起始字符=':' (16#3A)
Address Hi	'0'	通讯地址 :
Address Lo	'1'	8-bit 地址由 2 个 ASCII 码组合
Function Hi	'0'	功能码 :
Function Lo	'3'	8-bit 功能码由 2 个 ASCII 码组合
DATA (n-1)	'2'	数据内容 :
.....	'1'	$n \times 8$ -bit 数据内容由 $2n$ 个 ASCII 码组合

DATA 0	'0'	
	'2'	
	'0'	
	'0'	
	'0'	
	'2'	
LRC CHK Hi	'D'	LRC 检查码：
LRC CHK Lo	'7'	8-bit 检查码由 2 个 ASCII 码组合
END Hi	CR	终止符：
END Lo	LF	END Hi=CR (16#0D) , END Lo=LF (16#0A)

2. LRC 检查码：由通讯地址到数据内容结束加起来的值取 2 的补码即为检查码 (LRC Check)。例如：
 $16\#01+16\#03+16\#21+16\#02+16\#00+16\#02=16\#29$ ，然后取 2 的补码= $16\#D7$ 。

3

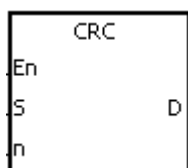
FB/FC	指令码	操作数	功能
FC	CRC	S, n, D	检查 CRC 模式

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●				●				
n	●	●			●	●		●	●				●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



S : RTU 模式和检查运算起始装置

n : 运算组数

D : 存放运算结果的起始装置。

指令说明：

1. CRC 检查码：请参考补充说明
2. **n** :范围 1~1000 不在此范围则视为运算错误,指令不执行,SM0、SM1=ON,SR0 记录错误码 16#200B。
3. 16 位转换模式：当 SM606=OFF 时，**S** 起始装置其数据区分为上 8 位、下 8 位，将每一个位数做 CRC 检查码运算，传送到 **D** 的上 8 位及下 8 位中，运算的位数以 **n** 来设定。
4. 8 位转换模式：当 SM606=ON 时，**S** 起始装置其数据区分为上 8 位（无效数据）、下 8 位，将每一个位数做 CRC 检查码运算，传送到 **D** 的下 8 位中占用 2 个寄存器，运算的位数以 **n** 来设定。（**D** 的上 8 位全部为 0）

程序范例：

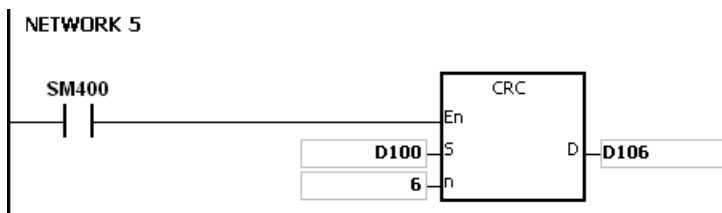
1. PLC 与 VFD-S 系列交流电机驱动器联机（RTU 模式，SM210=ON）、（16 位模式，SM606=ON），传送数据预先写入欲写入 VFD-S 参数地址 16#2000 写入内容为 16#12。

PLC VFD-S，PLC 传送：01 06 2000 0012 02 07

PLC 传送数据寄存器（PLC 传送信息）

寄存器	DATA	说明
D100 下	16#01	Address
D101 下	16#06	Function
D102 下	16#20	数据地址
D103 下	16#00	Data Address
D104 下	16#00	数据内容
D105 下	16#12	Data content
D106 下	16#02	CRC CHK 0
D107 下	16#07	CRC CHK 1

上列 CRC CHK (0 , 1) 为错误检查码可由指令 CRC 算出 (8 位 Mode , SM606=ON)



CRC 检查码：此时，16#02 存于 D106 下 8 位内，16#07 存于 D107 下 8 位内。

补充说明：

1. 有一通讯数据的 RTU 模式，格式如下：

START	时间间隔
Address	通讯地址：8-bit 二进制地址
Function	功能码：8-bit 二进制
DATA (n-1)	数据内容：
.....	n×8-bit 数据
DATA 0	
CRC CHK Low	CRC 检查码：
CRC CHK High	16-bit CRC 检查码由 2 个 8-bit 二进制组合
END	时间间隔

2. CRC 检查码：检查码由 Address 到 Data content 结束。其运算规则如下：

- 步骤一： 令 16-bit 寄存器 (CRC 寄存器) =16#FFFF
- 步骤二： Exclusive OR 第一个 8-bit Byte 的信息指令与低位 16-bit CRC 寄存器，Exclusive OR，将结果存入 CRC 寄存器内。
- 步骤三： 右移一位 CRC 寄存器，将 0 填入高位处。
- 步骤四： 检查右移的值，如果是 0 将步骤 3 的新值存入 CRC 寄存器内否则 Exclusive OR 16#A001 与 CRC 寄存器，将结果存入 CRC 寄存器内。
- 步骤五： 重复步骤 3~步骤 4，将 8-bit 全部运算完成。

步骤六：重复步骤 2 ~ 步骤 5，取下一个 8-bit 的信息指令，直到所有信息指令运算完成。最后，得到的 CRC 寄存器的值，即是 CRC 的检查码。值得注意的是 CRC 的检查码必须交换放置于信息指令的检查码中。

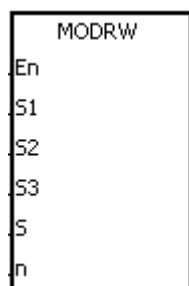
FB/FC	指令码		操作数				功能						
FC	MODRW		S ₁ , S ₂ , S ₃ , S, n				MODBUS 数据读写						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁ , S ₂ , S ₃		●					●							
S	●	●					●							
n		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁ , S ₂ , S ₃	●	●			●	●		●	●				●	○	○		
S	●	●			●	●		●	●				●				
n	●	●			●	●		●	●				●	○	○		

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



- S₁ : 联机装置地址
- S₂ : 通讯功能码
- S₃ : 欲读写数据的地址
- S : 欲读写的数据存放寄存器
- n : 读写数据长度

指令说明：

1. S₁ : 联机装置地址 (Unit Address) 。指定范围 0~255。
2. S₂ : 通讯功能码 (Function Code) 。

例如：

PLC 的读取多笔位装置 (Bit) (非 Discretes Input 装置) 命令为 1 (16#01) ，
 PLC 的读取多笔位装置 (Bit) (仅 Discretes Input 装置) 命令为 2 (16#02) ，
 PLC 的读取多笔字符装置 (Word) (非 Input Register 装置) 命令为 3 (16#03) ，
 PLC 的读取多笔字符装置 (Word) (仅 Input Register 装置) 命令为 4 (16#04) ，
 PLC 的单笔位装置 (Bit) 状态写入命令为 5 (16#05) ，
 PLC 的单笔字符装置 (Word) 数据写入命令为 6 (16#06) ，
 PLC 的多笔位装置 (Bit) 状态写入命令为 15 (16#0F) ，

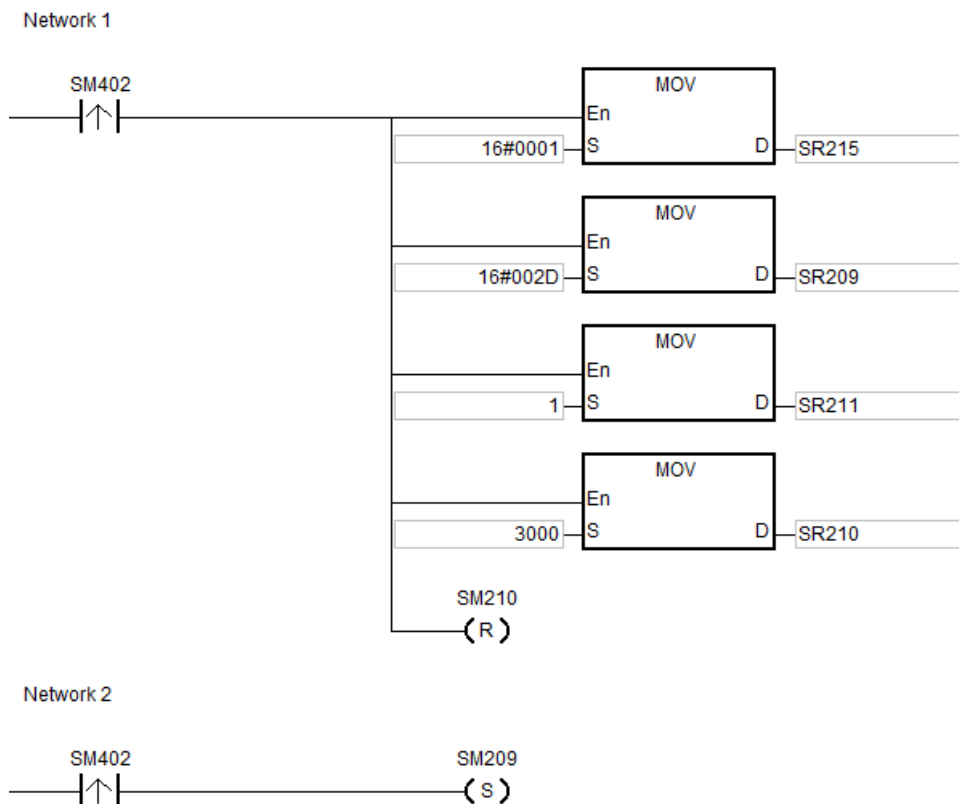
PLC 的多笔字符装置 (Word) 数据写入命令为 16 (16#10) 。

目前仅支持上述功能码，其余功能码将无法执行。请参考下列程序范例。

3. **S₃** : 欲读写数据的地址 (Device Address)。联机装置的内部装置地址，若地址对于被指定的装置不合法，则联机装置会响应错误信息，PLC 将错误码储存于 Error Log 里。
4. **S** : 欲读写的数据 (Source or Destination)。由用户设定寄存器，将欲写入数据长度的数据事先存入寄存器内。或数据读取后存放的寄存器。
5. **n** : 读写数据长度 (Data Length)，设定的数据量不可以大于 240 bytes。Coil (输出点) 相关的通讯命令，其数据单位为 Bit，n 的范围为 1~1920。Register 相关的通讯命令，其数据单位为 Word，则 n 的范围为 1~120。
6. 本指令在程序中使用次数并无限制，但是不同指令 (如 MODRW、FWD...) 在同一时间使用相同的通讯端口时，仅会只有一个指令被执行。
7. 当发生通讯逾时时，逾时标志会 ON 起来 (SM104，SM105)。若此问题已解除，则可以将此逾时标志 (SM104，SM105) 设定为 OFF，在使用 MODRW 指令时 逾时时间不可设为 0 必须在 100 ~ 65535ms 区间内。
8. MODBUS ASCII 模式时，用户只需设定好，所欲传送的数据，本指令会自动加入起始字符 (:)、检查码 (LRC) 和终止符 (CR LF)，其接收数据以 ASCII 字符形式储存于内部寄存器内，AH 运动控制主机自动将其内容转换为 HEX 数值存放在 **S** 中。
9. MODBUS RTU 模式时，用户只需设定好，所欲传送的数据，本指令会自动加入检查码 (CRC)，其接收数据以 HEX 数值形式储存于 **S** 中。

通讯协议设定范例：

1. 通讯协议设定成 RS485 ASCII，9600，8，E，1，用户可利用 HWCONFIG 或是特殊寄存器来设定通讯端口通讯能力。(HWCONFIG 相关设定请参照 ISPSOFT 使用手册，通讯相关 SR，SM 寄存器设定可参考 RS 通讯应用指令)。
2. 设定通讯端口传输方式为 RS485 (SR215=1)。
3. 设定通讯端口其通讯速率与格式为 9600，8，E，1 (SR209=16#002D)。
4. 设定其自动询问次数为 1 (SR211=1)。
5. 设定其逾时时间为 3000ms (SR210=3000)。
6. 设定为 ASCII 模式 (SM210=OFF)。
7. 通讯端口通讯能力设定生效 (SM209=ON)。

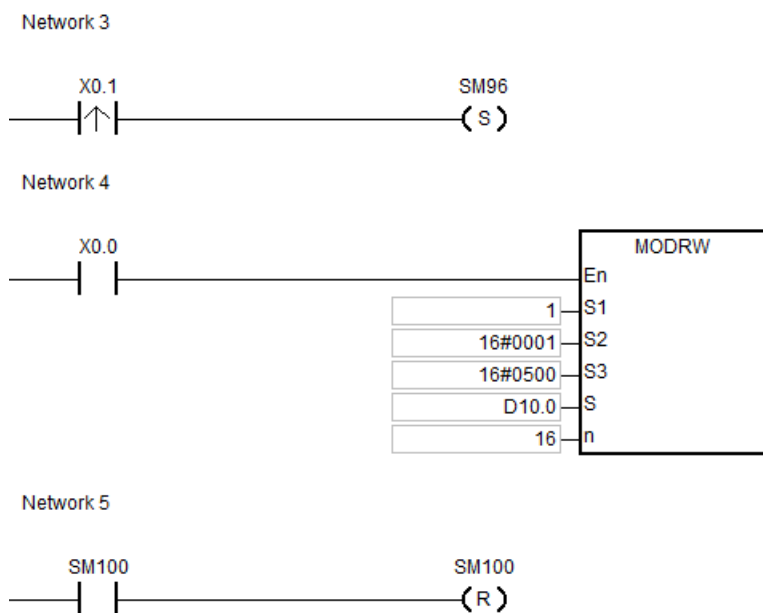


用户也可于 ISPSOft - > HWCONFIG - > COM Port 内直接对通讯端口进行设定，则可以省略此步骤。

程序范例一：

1. 功能码 01 (16#01)：读取多笔位数据。（此范例读取 16 笔）
2. AH 运动控制主机与 DVP-ES2 联机：

当 SM96 为 ON 且 X0.0 也为 ON 的状态时，将会对 DVP-ES2 进行数据的读取，从 Y0 开始至 Y15(Y0 的地址为 16#0500)。
3. DVP-ES2 所回复的内容会储存在 D10.0 到 D10.15。
4. DVP-ES2 所回传的数据接收完毕后，会对 DVP-ES2 所回传的数据格式进行确认，是否符合数据格式，如果正确无误则 SM100=ON，如果回传数据格式错误则 SM102=ON。

**程序范例二：**

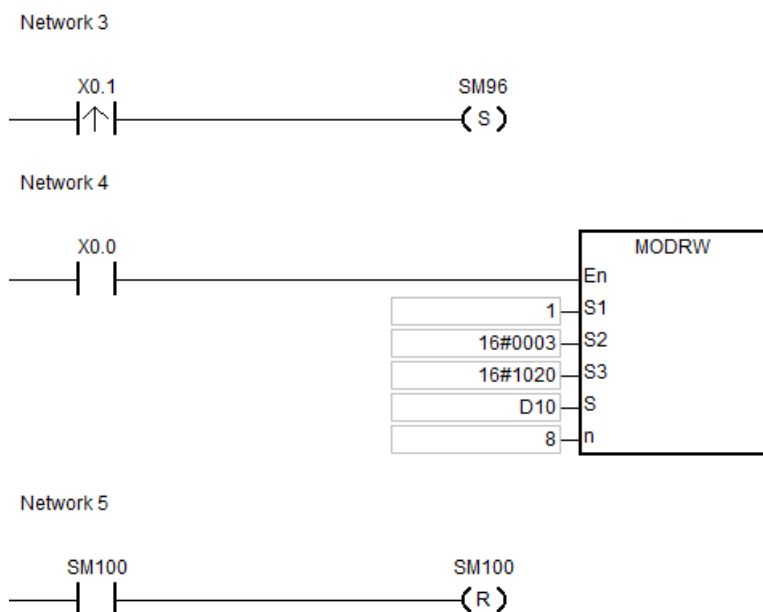
1. 功能码 03 (16#03) : 读取多笔寄存器数据。(此范例读取 8 笔)

2. AH 运动控制主机与 DVP-ES2 联机

当 SM96 为 ON 且 X0.0 也为 ON 的状态时 将会对 DVP-ES2 进行数据的读取 从 D20 开始至 D27(D20 的地址为 16#1020)。

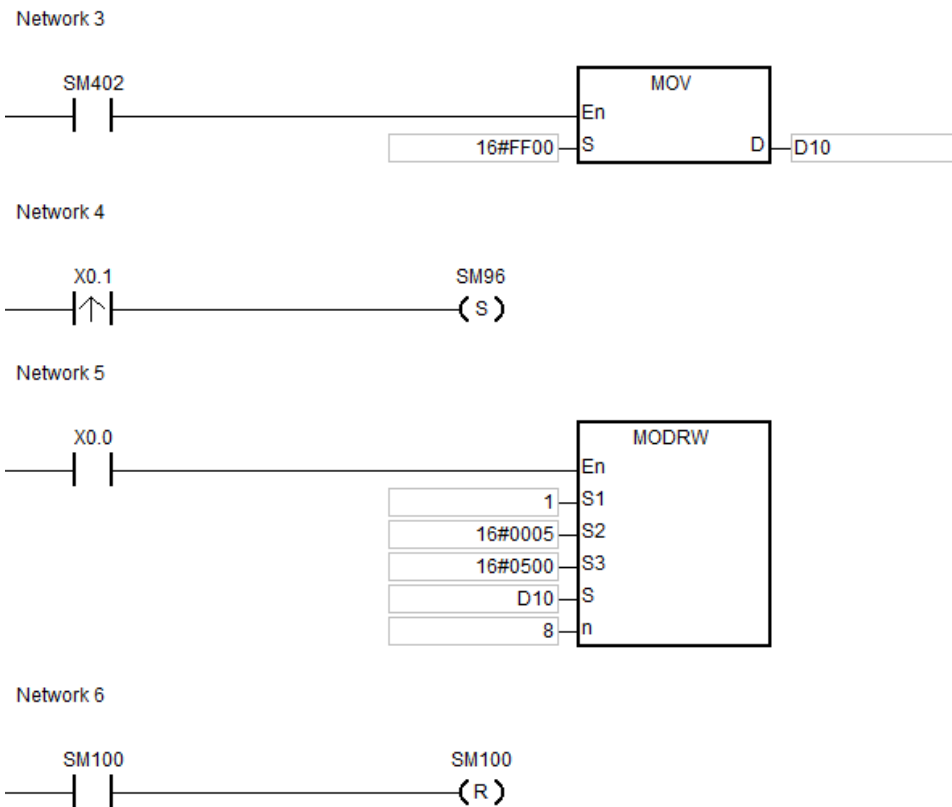
3. DVP-ES2 所回传的数据会储存在 D10 至 D17 装置内。

4. DVP-ES2 所回传的数据接收完毕后, 会对 DVP-ES2 所回传的数据格式进行确认, 是否符合数据格式, 如果正确无误则 SM100=ON, 如果回传数据格式错误则 SM102=ON。



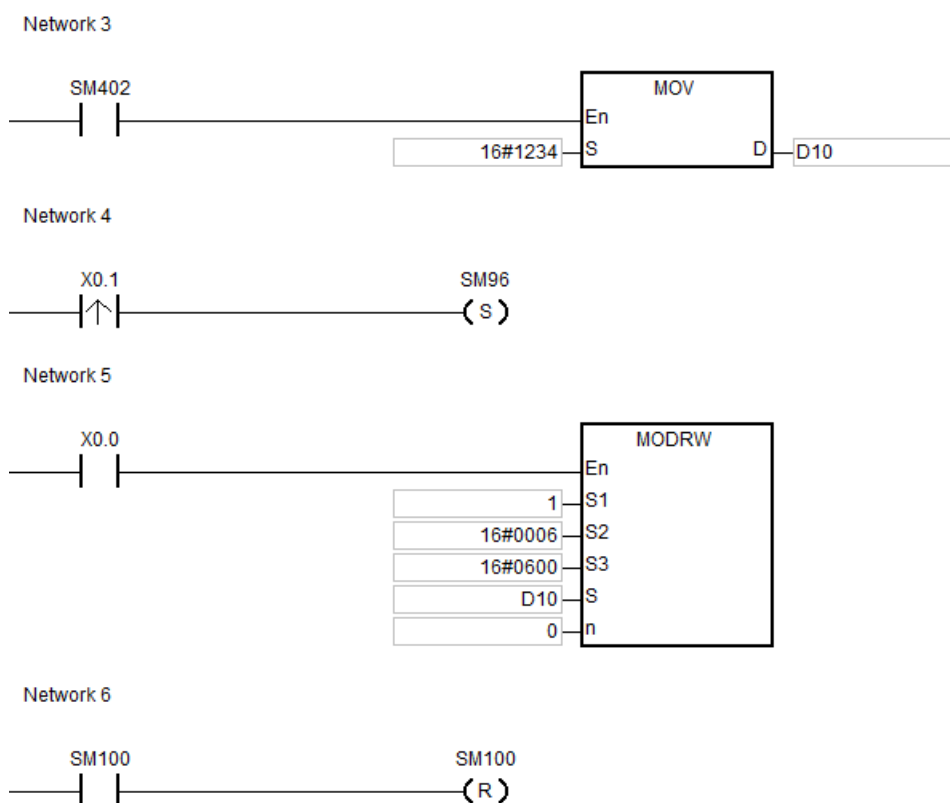
程序范例三：

1. 功能码 05 (16#05) : 写入单个位数据至 DVP-ES2。
2. ASCII Mode : AH 运动控制主机与 DVP-ES2 联机
 当 SM96 为 ON 且 X0.0 也为 ON 的状态时 将会对 DVP-ES2 的 Y0 写入数据(Y0 的地址为 16#0500)。
3. DVP-ES2 所回传的数据接收完毕后, 会对 DVP-ES2 所回传的数据格式进行确认, 是否符合数据格式, 如果正确无误则 SM100=ON, 如果回传数据格式错误则 SM102=ON。
4. 当 DVP-ES2 接收到此通讯命令的时候, 会强制 Y0 为 ON 的状态。
5. 此功能码为写入功能, 所以在本指令中的 n 值, 并不会被使用到。



程序范例四：

1. 功能码 06 (16#06) : 写入单个寄存器数据至 DVP-ES2。
2. ASCII Mode : AH 运动控制主机与 DVP-ES2 联机
 当 SM96 为 ON 且 X0.0 也为 ON 的状态时 将会对 DVP-ES2 的 T0 写入数据(T0 的地址为 16#0600)。
3. DVP-ES2 所回传的数据接收完毕后, 会对 DVP-ES2 所回传的数据格式进行确认, 是否符合数据格式, 如果正确无误则 SM100=ON, 如果回传数据格式错误则 SM102=ON。
4. 当 DVP-ES2 接收到此通讯命令的时候, 会对 T0 写入 D10 装置内的数据。
5. 此功能码为写入功能, 所以在本指令中的 n 值, 并不会被使用到。

**补充说明：**

1. S_1 或 S_2 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. S 装置不足以写入或读取 n 所指定的长度时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
3. n 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#200B。
4. 当 S2 通讯功能码指定为 Bit 读写时 S 必须指定为 Bit 装置，否则视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
5. 当 S2 通讯功能码指定为字符装置读写时 S 必须指定为字符装置，否则视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
6. 当通讯命令为 0x05 与 0x06 时，n 无作用，指令只会写入 1 个 Bit 或 Word。
7. 若执行 MODRW 的时候，送信标志（SM96、SM97）都没有 ON 起来的话，则不执行。
8. 当发生通讯超时（Timeout）的时候，通讯超时标志（SM104、SM105）会被设定为 ON，等待（Waiting）标志（SM98、SM99）会被设定为 OFF。
9. 当接收指令有错误（Error）时，错误标志（SM102、SM103）会被设定为 ON，等待（Waiting）标志（SM98、SM99）会被设定为 OFF。
10. 当 S2 为读写 Word 装置，则读写对方设备装置也必须是 Word 装置，当 S2 为读写位装置，则读写对方设备装置也必须为位装置。

11.RS-485/RS-232 通讯 MODRW 指令相关标志信号与特殊寄存器：

标志信号		功能说明
COM1	COM2	
SM96	SM97	送信要求
SM98	SM99	接收等待
SM100	SM101	接收完成
SM102	SM103	接收发生错误
SM104	SM105	通讯超时
SM209	SM211	通讯协议变更用

标志信号的功能详细说明请参考 RS 通讯应用指令。

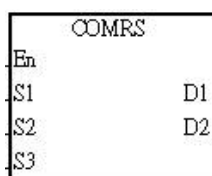
FB/FC	指令码		操作数				功能						
FC	COMRS		S ₁ , S ₂ , S ₃ , D ₁ , D ₂				通讯传送与接收指令						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							
S ₃		●					●							
D ₁		●					●							
D ₂		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●				●	○	○		
S ₂	●	●			●	●		●	●								
S ₃	●	●			●	●		●	●				●	○	○		
D ₁	●	●			●	●		●	●								
D ₂	●	●			●	●		●	●								

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



S₁ : 通讯端口编号 (1~2)

S₂ : 传送数据源

S₃ : 传送长度设定值

D₁ : 接收通讯数据的起始装置组件

D₂ : 接收数据结束条件参数设定

指令说明：

- 当接收结束条件有指定特定字符时，建议使用于 ASCII 编码的通讯数据，若非 ASCII 编码的通讯数据，则建议改用通讯超时时间做为结束条件。
- COMRS 指令仅支持 AH500 主机内建的 COM，AHCPU5xx-EN 系列为 COM1，AHCPU5xx-RS2 系列为 COM1 与 COM 2。
- S₁ : 通讯端口编号范围。
S₁ = 1 (COM1)，S₁ = 2 (COM2)，若超出 PLC 通讯端口范围时，则指令将不会执行任何通讯数据接收。
- S₂ 通讯数据传送来源与 S₃ 传送长度设定值，其范例说明如下：假设 S₂ 选择 D100 及 S₃ 为 10，则指令将会传送 D100~D109 的下 8 位数值共 10 个字，由 S₁ 指定的通讯端口传送出去。

5. 当 S_3 传送长度设定值为 0 时，即表示不需要传送字符串；传送长度最大值为 1000 个字符。
6. D_1 为已接收数据长度值， $D_{1+1} \sim D_{1+n}$ 为存放接收数据地址，说明如下：假设 D_1 设为 D200， $D_2=3$ 及 D_{2+1} 设定终止符 16#0D0A，则指令将会依照接收数据顺序存放于 D201 之后的下 8 位中（上 8 位不变化），直到两个连续结束位为 16#0D 与 16#0A 时，即停止接收数据，并将接收长度填入 D200 内，接着设定完成标志。
7. D_{2+0} 为接收模式参数设定， D_{2+1} 为接收条件参数设定，其详细说明如下表所示。

说明列表如下：

D_2	D_2 接收模式说明	D_{2+1} 参数内容说明	备注
0	不接收通讯数据	不使用	即传送完成后，设定完成标志。
1	通讯数据不连续时间超出 D_{2+1} 设定值时，即为接收完成。	时间长度设定值，时间单位为 1ms，设定值范围为 2~3000。	设定值超出 3000 时，其值为 3000；小于 2 时，其值为 2。
2	接收数据有一个特定字符为结尾字符	特定结尾字符设定值	如特定字符为 16#0A 则 D_{2+1} 设定为 16#000A
3	接收数据有两个连续特定字符为结尾字符	两个特定结尾字符设定值	如两个特定字符顺序为 16#0D 与 16#0A，则 D_{2+1} 设定为 16#0D0A
4	接收数据有一个特定字符为开始字符，以及不连续时间超出 D_{2+1} 设定值时，即为接收完成。	上 8 位为特定开始字符，下 8 位为不连续时间设定值（范围 2~255ms）。	如开始字符为 16#3A 以及不连续时间 15ms，则 D_{2+1} 设定为 16#3A0F
5	接收数据有一个特定字符为开始字符与一个结尾字符	特定开始与结尾字符设定值	如开始字符为 16#3A，结尾字符为 16#0A，则 D_{2+1} 设定为 16#3A0A
6	接收数据有一个特定长度为接收完成	D_{2+1} 为接收长度设定值	如接收 10 个字符，则 D_{2+1} 设定为 10
其它	超出现有支持模式范围，则指令将不会被启动。		

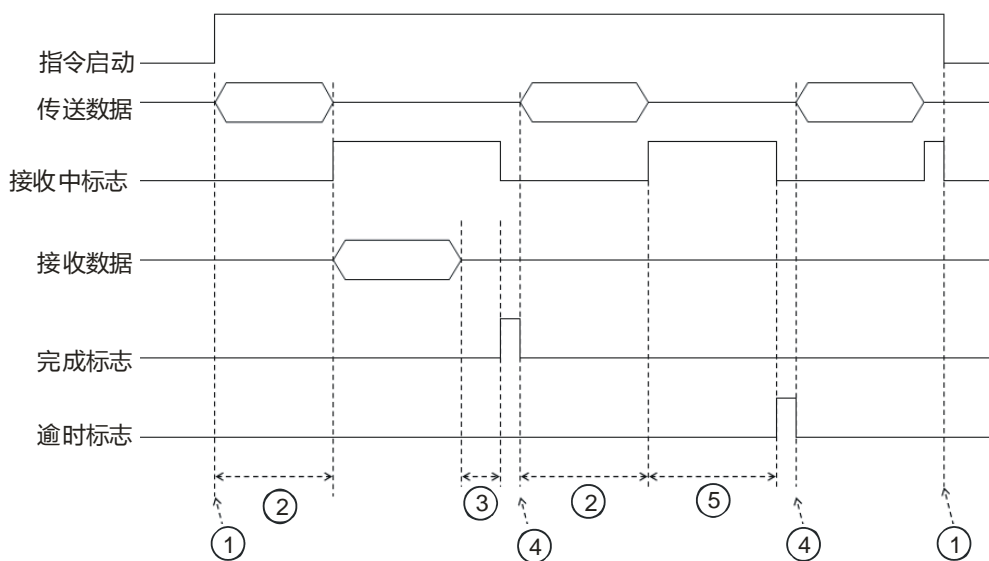
8. D_{2+0} 为接收模式参数设定， D_{2+1} 为接收条件参数设定，其详细说明如下表所示。
9. 通讯端口对应的特殊辅助继电器 SM 与特殊数据寄存器 SR 如下：

通讯端口编号	COM1	COM2	备注
接收中标志	SM98	SM99	由 PLC 设定 ON 为接收中，接收结束也由 PLC 清除为 OFF。

通讯端口编号	COM1	COM2	备注
完成标志	SM100	SM101	由 PLC 设定 ON 为完成，但须由用户程序清除 OFF，然后等待下一笔通讯数据。
通讯超时标志	SM104	SM105	由 PLC 设定 ON 为超时，但须由用户程序清除 OFF，然后等待下一笔通讯数据。
超时时间设定	SR210	SR213	数值为 0 表示不启动通讯超时功能，时间单位为 1ms。
数据传送/接收模式	SM106	SM107	传送及接收时的数据格式，ON 时为 8 位，OFF 时为 16 位。

10. 通讯传送与接收时序图如下：

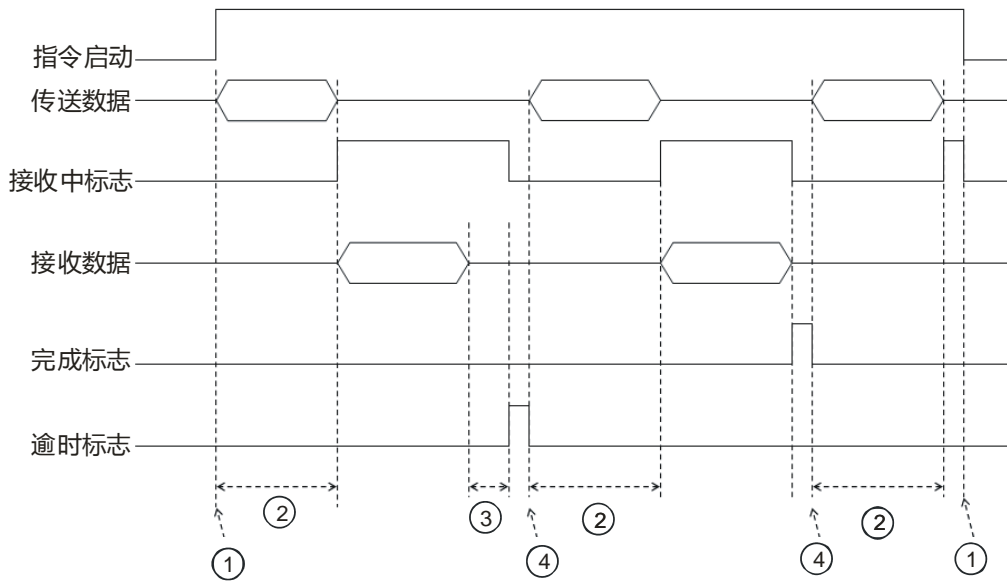
- 接收模式为 0 时
需注意数据传送时无法中途取消，此时即使指令启动条件不成立，仍会继续传送数据，惟传送完成后不会将完成标志设为 ON。
- 接收模式为 1 或 4 时



图标符号说明：

- ① → 由用户设定启动/关闭
- ② → 传送数据时间，此期间通讯超时不会被计时。
- ③ → 接收到第 1 个字符时，即开始累积计时结束时间，每接收一个字符即清除累积计时值一次，直到累积计时超出 D2+1 设定值之后，则产生通讯完成标志。
- ④ → 由用户清除完成或超时标志，若指令还处于开启状态，则下个周期扫描到指令时，将自动重新传送下一笔通讯数据。
- ⑤ → 当接收开始时即开始累积计时，直到累积计时超出通讯超时设定值之后，则产生通讯超时标志。建议通讯超时时间最好设定比 D2+1 结束时间长。

- 接收模式为 2、3、5 或 6 时



图标符号说明：

- ① → 由用户设定启动/关闭
- ② → 传送数据时间，此期间通讯逾时不会被计时。
- ③ → 接收开始时即开始累积计时，每接收一个字符即清除累积计时值一次，直到累积计时超出通讯逾时设定值之后，则产生通讯逾时标志。
- ④ → 由用户清除完成或逾时标志，若指令还处于开启状态，则下个周期扫描到指令时，将自动重新传送下一笔通讯数据。

11. 通讯传送与接收时序图如下：

《8 位模式》：用户将编辑好的命令储存在传送装置起始地址上，传送出去的命令包含头后缀。将 16 位数据分成上 8 位，下 8 位，上 8 位被省略，仅下 8 位为有效数据可做数据的传送和接收。(以标准 MODBUS 为例)

传送数据：(PLC→外部机器)

D10下	D11下	D12下	D13下	D14下	D15下	D16下
头码		来源数据寄存器由D10下 8位开始			尾码 1	尾码 2
长度=7						

接收数据：(外部机器→PLC)

D100下	D101下	D102下	D103下	D104下	D105下	D106下
头码	来源数据寄存器由D100下 8位开始 长度=7				尾码 1	尾码 2

《16 位模式》：用户将编辑好的命令储存在传送装置起始地址上，传送出去的命令包含头后缀。当 SM106/SM107=OFF 时，指定为 16 位转换模式，将 16 位数据分成上 8 位与下 8 位做数据的传送和接收。

传送数据：(PLC→外部机器)

D10下	D10上	D11下	D11上	D12下	D12上	D13下
头码	来源数据寄存器由D10下 8位开始 长度=7				尾码 1	尾码 2

接收数据：(外部机器→PLC)

D100下	D100上	D101下	D101上	D102下	D102上	D103下
头码	来源数据寄存器由D100下 8位开始 长度=7				尾码 1	尾码 2

PLC 接收数据会将外部机器传入数据报含头码、后缀一起接收，所以长度的设定要注意。

补充说明：

1. 此 COMRS 通讯指令无使用次数限制，但每个通讯端口都仅能被一条通讯指令启动，因此后面启动的通讯指令将不会被执行。
2. 通讯指令包含有 RS、MODRW、FWD 及 REV 等等都会占用通讯端口的通讯指令。
3. 此 COMRS 指令无检查码机制之设计，须此功能者可搭配现有提供便利指令自行组合运用。
4. D2 内容值为 2、3、5、6 时，建议设定通讯超时。若有设定通讯超时，当指令没有收到终止符时，才能重新进行重传机制。

5. 每次刚启动指令或重新开始接收新的下一笔通讯数据时，此指令并不会自动清除 D1+0 ~ D1+n 装置内的数值，相关的接收数据信息（如接收数据是否有接收？与接收长度为何？）皆以完成标志由 OFF 变为 ON 时为基准。若需清除这些接收数据区域时，建议可搭配 ZRST 指令清除。
6. 当 S1 的内容值超出范围时（1~2），指令不执行，SM0=ON，错误码 SR0=16#2003。
7. S2 装置不足以读取 S3 所指定的长度时，指令不执行，SM0=ON，错误码为 SR0=16#2003。
8. 当 D2 的内容值超出范围时（0~6），指令不执行，SM0=ON，错误码 SR0=16#2003。
9. 当 D2 的内容值为 6，D1 装置不足以写入 D2+1 所指定的长度时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
10. 当 D2 的内容为 1~5 时，接收数据超出 D1 最大装置范围，则超出装置范围的接收数据将会被忽略。
11. 当接收完成标志变为 ON 时，其通讯数据将暂停接收，若此期间通讯端口还有数据传送，将不再接收。
12. 当 S3 传送长度设定值为小于 0 或大于 1000 时，指令不执行，SM0=ON，错误码为 SR0=16#2003。

3.22 其它指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>WDT</u>	—	✓	逾时监视定时器	1
FC	<u>DELAY</u>	—	✓	延时指令	3
FC	<u>GPWM</u>	—	—	一般用脉冲宽度调变	7
FC	<u>TIMCHK</u>	—	—	时间检查	7
FC	<u>EPUSH</u>	—	✓	变址寄存器存入	3
FC	<u>EPOP</u>	—	✓	变址寄存器读出	3

FB/FC	指令码		操作数	功能
FC	WDT	P	—	逾时监视定时器
			脉冲执行型	16 位指令
			-	AH 运动控制主机
			32 位指令	
			-	

图形：



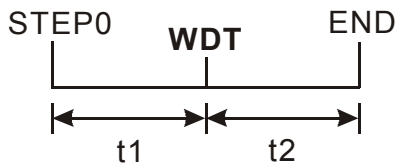
指令说明：

1. AH 运动控制主机系统中有一逾时监视定时器 (Watchdog Timer) ，用来监视 PLC 系统的正常运转。
2. WDT 指令可用来清除 PLC 中监控定时器的计时时间。当 PLC 的扫描程序时间过长，超过用户设定的 WDT 时间 (默认值 200ms) 时，PLC ERROR 的指示灯会亮，PLC 会自动变 STOP，用户只要重新做 STOP=>RUN 就可以恢复正常。

3. 令逾时监视定时器动作的时机：

- PLC 系统发生异常。
- 程序运行时间太长，造成扫描时间大于 (Watchdog Timeout) 的内容值。可以下列 2 种方法来改善。

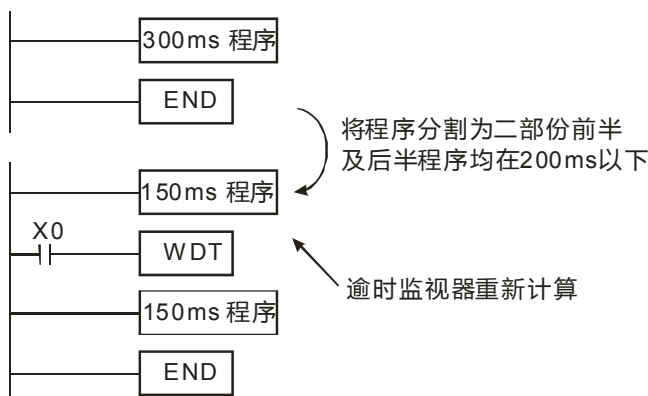
◆ 使用 WDT 指令。



◆ 要改变逾时监视时间请参考 **ISPSOft 使用手册** 中的硬件组态工具说明。

程序范例：

若程序扫描时间为 300ms，此时，可将程序分割为 2 部份，并在中间放入 WDT 指令，使得前半及后半程序都在 200ms 以下。



补充说明：

设定 WDT 时间的方式，请参考 **ISPSoft 使用手册** 说明。

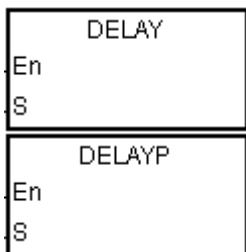
FB/FC	指令码			操作数				功能						
FC		DELAY	P	S				延时指令						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S : 延时时间

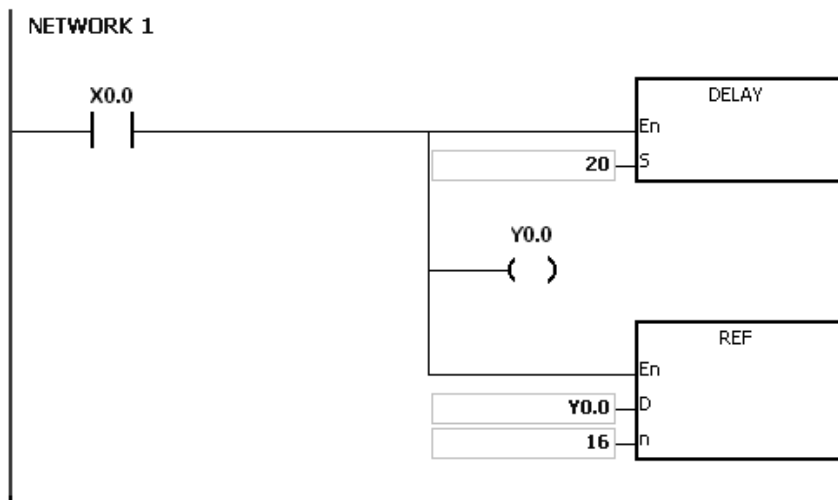
指令说明：

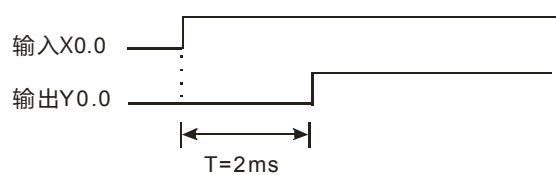
执行 DELAY 指令后，在每次扫描周期 DELAY 指令后面的程序执行会依用户指定的时间作延时。

S 延时时间的单位：0.1ms

程序范例：

当 X0.0=ON 时，执行 DELAY 指令延时 2ms 后才执行后面程序，Y0.0 导通且更新 Y0.0~Y0.15 的状态。



**补充说明：**

1. $S \leq 0$ 时，不会有延时时间。
2. $S > 1000$ ，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 用户可依实际状况来调整延时时间。
4. DELAY 指令会受到通讯或其它影响而增加延时时间。

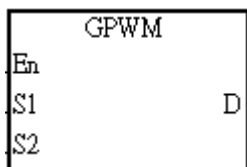
FB/FC	指令码			操作数					功能						
FC	GPWM			S ₁ , S ₂ , D					基本脉冲宽度调变						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●		●				
S ₂	●	●			●	●		●	●				●				
D		●	●	●				●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



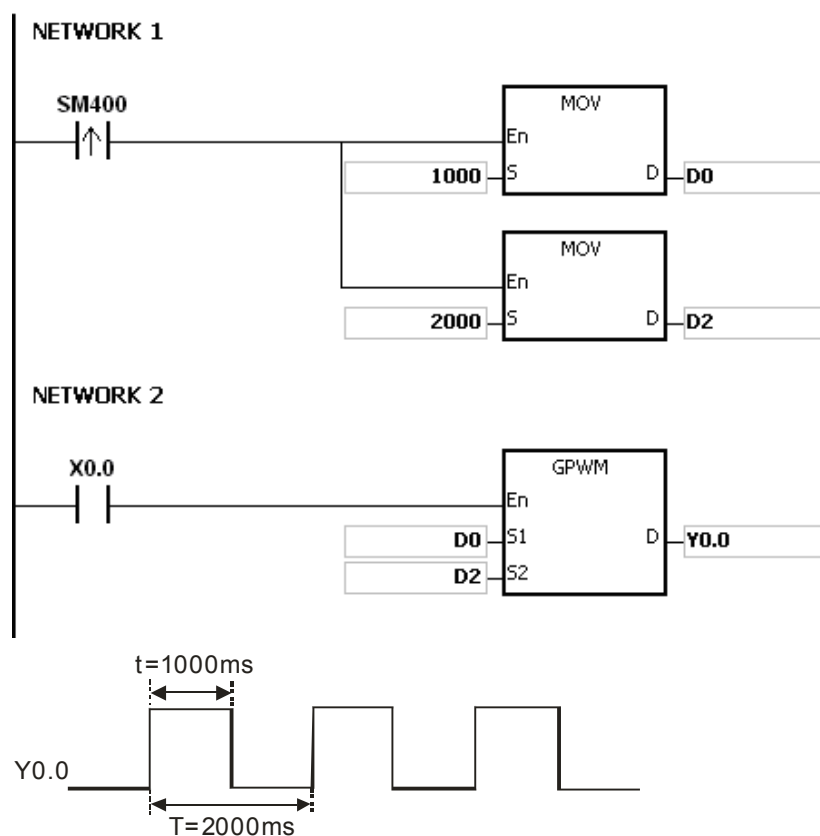
- S₁ : 脉冲输出宽度
- S₂ : 脉冲输出周期
- D : 脉冲输出装置

指令说明：

1. GPWM 指令执行时，指定 S₁ 脉冲输出宽度与 S₂ 脉冲输出周期，并由 D 脉冲输出装置输出。
2. S₁ 脉冲输出宽度指定 t：0~32,767ms。
3. S₂ 脉冲输出周期指定为 T：1~32,767ms，但 S₁ ≤ S₂。
4. S₂+1、S₂+2 为系统用参数，请勿占用。
5. 当 S₁ ≤ 0 时，脉冲输出装置无输出，当 S₁ ≥ S₂ 时，脉冲输出装置一直为 ON。
6. S₁、S₂ 可在 GPWM 指令执行时更改。
7. 当条件接点未启动时，D 装置输出 0。
8. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。

程序范例：

程序启动时，D0=1000，D2=2000，当 X0.0=ON 时，Y0.0 输出以下脉冲，当 X0.0=OFF 时，Y0.0 输出也变成 OFF。

**补充说明：**

1. 此指令是以扫描周期去计数，因此最大误差为 1 个 PLC 扫描周期。 S_1 、 S_2 与 $(S_2 - S_1)$ 的值必须 $>$ PLC 扫描周期，否则 GPWM 输出会有误动作。
2. 若将此指令置于功能块或中断中使用，则会产生 GPWM 输出不准确的情况发生，请特别注意。
3. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

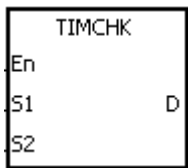
FB/FC	指令码		操作数				功能				
FC	TIMCHK		S ₁ , S ₂ , D				时间检查				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂		●					●							
D	●													

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●				●				
S ₂	●	●			●	●		●	●		●	○	●	○	○		
D	●	●	●	●				●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



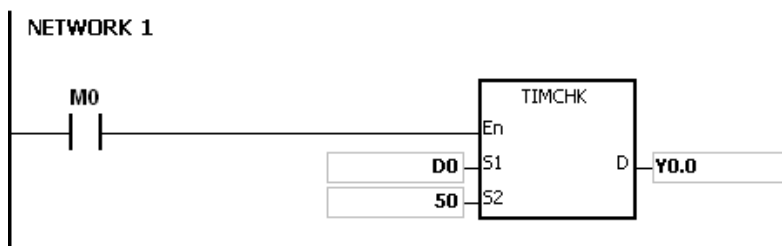
- S₁ : 存放测量值所经过的时间
- S₂ : 测量设定值
- D : 测量时间到达时的输出装置

指令说明：

1. 条件接点 ON 时，S₁ 开始计数，直到 S₁ ≥ S₂，D 输出装置=ON，之后就算条件接点变成 OFF，S₁ 内容值不变且 D 装置一样为 ON。
2. 条件接点由 OFF → ON 时，S₁ 被清除为 0，D 输出装置=OFF。
3. 计数的单位为 100ms。
4. S₁₊₁、S₁₊₂ 为系统用参数，请勿占用。
5. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。

程序范例：

M0=ON 时，D0 开始计数，直到 D0 ≥ 50 (5 秒)，Y0.0=ON，之后就算条件接点变成 OFF，D0 内容值不变且 Y0.0 一样为 ON。



补充说明：

1. 若 S_1+2 ，装置超过范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. S_1 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

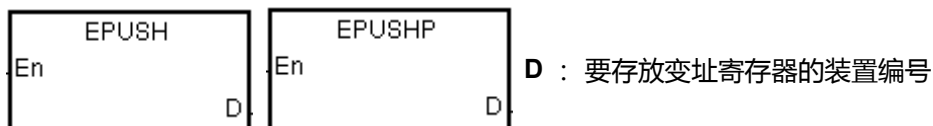
FB/FC	指令码			操作数				功能						
FC		EPUSH	P	D				变址寄存器存入						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●				●				

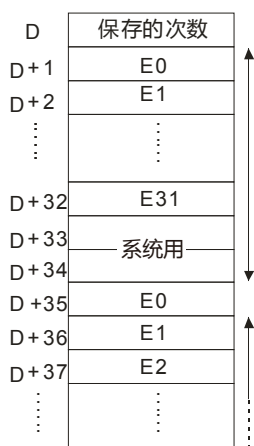
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

1. 将 E0~E31 的内容值存放到 **D** 的内容（保存的次数）所指定的装置。
2. 每次执行存放需 34 个装置（最后两个为系统保留用），因此若保存次数为 n（D 的内容值），则执行时 E0~E31 会存放在 D+34*n+1~D+34*n+32 中，且保存次数 D=n+1。
3. 如果 EPUSH 执行多次，则 E0~E31 会被依序往下存放，所以 D 的容量要够大。
4. 配合 EPOP 指令使用，可以达到 E 装置的堆栈使用。

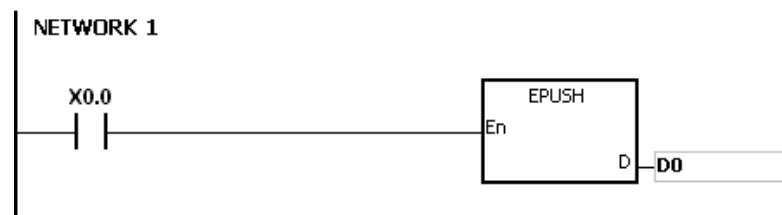


程序范例：

给定 D0=0，当第一次 X0.0=ON 时，E0~E31 的内容会分别传送至 D1~D32，D0=1，
第二次 X0.0 由 OFF→ON 时，E0~E31 的内容会分别传送至 D35~D66，D0=2。

第 N 次 X0.0 由 OFF→ON，E0~E31 的内容会分别传送至：

装置编号 (D0 的内容值*34) +1~ (D0 的内容值*34) +32



补充说明：

1. D 内容值<0 时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $D + (D \text{ 内容值} + 1) * 34 - 1$ 装置容量不足时，指令不执行，SM0=ON，错误码 SR0=16#2003。

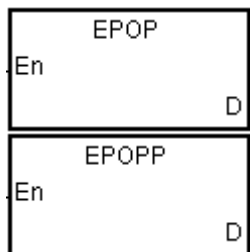
FB/FC	指令码		操作数				功能				
FC	EPOP	P	D				变址寄存器读出				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

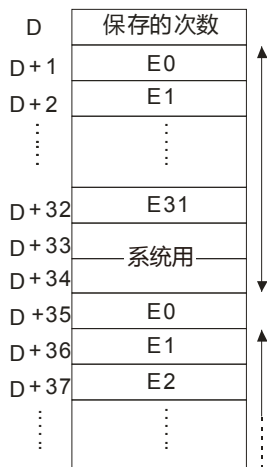
图形：



D：要读出内容值到变址寄存器的装置编号

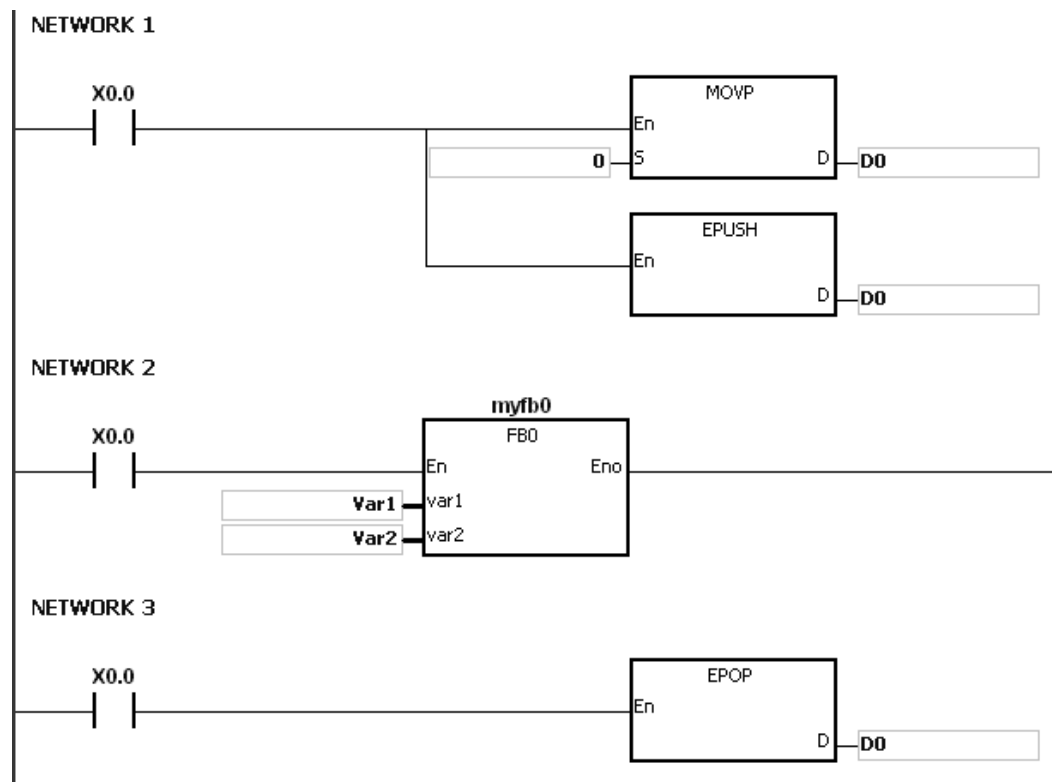
指令说明：

1. 将 **D** 的内容值读出到装置 E0~E31，并且将 **D** 的内容值（保存的次数）减 1。
2. 每次执行存放需 34 个装置（最后的两个装置为系统保留用），因此若保存次数为 n（**D** 的内容值），则该次会读取 $D+34 \times (n-1) + 1 \sim D+34 \times (n-1) + 32$ 中的内容值写入 E0~E31，且保存次数 $D=n-1$ 。
3. 取出的顺序为最后放入 **D** 中的值。



程序范例：

当 X0.0=ON 时，先将 D0 内容值设为 0，然后使用 EPUSH 将 E0~E31 的内容分别传送至 D1~D32，再开始执行功能块 FB0，当功能块 FB0 执行完毕之后，再用 EPOP 将 D1~D32 的值读出至 E0~E31。



补充说明：

1. D 内容值 ≤ 0 时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. D + D 内容值 $\times 34 - 1$ 装置容量不足时，指令不执行，SM0=ON，错误码 SR0=16#2003。

3.23 字符串处理指令

FB/FC	指令码 (位)			P 指 令	功能	Step
	16	32	64			
FC	<u>BINDA</u>	<u>DBINDA</u>	—	✓	有号数十进制→ASCII 变换	5
FC	<u>BINHA</u>	<u>DBINHA</u>	—	✓	BIN 十六进制→十六进制 ASCII 变换	5
FC	<u>BCDDA</u>	<u>DBCDDA</u>	—	✓	BCD→ASCII 变换	5
FC	<u>DABIN</u>	<u>DDABIN</u>	—	✓	有号数十进制 ASCII→有号数十进制 BIN 变换	5-11
FC	<u>HABIN</u>	<u>DHABIN</u>	—	✓	十六进制 ASCII→十六进制 BIN 变换	5-11
FC	<u>DABCD</u>	<u>DDABCD</u>	—	✓	BCD ASCII→BCD 变换	5-11
FC	<u>\$LEN</u>	—	—	✓	计算字符串长度	5-11
FC	<u>\$STR</u>	<u>\$DSTR</u>	—	✓	BIN→String	7
FC	<u>\$VAL</u>	<u>\$DVAL</u>	—	✓	String→BIN	7-13
FC	<u>\$FSTR</u>	—	—	✓	Float→String	7-8
FC	<u>\$FVAL</u>	—	—	✓	String→Float	5-11
FC	<u>\$RIGHT</u>	—	—	✓	从右边获取字符串	7-13
FC	<u>\$LEFT</u>	—	—	✓	从左边获取字符串	7-13
FC	<u>\$MIDR</u>	—	—	✓	区段获取字符串	7-13
FC	<u>\$MIDW</u>	—	—	✓	区段字符串取代	7-13
FC	<u>\$SER</u>	—	—	✓	字符串搜索	9-21
FC	<u>\$RPLC</u>	—	—	✓	字符串取代	11-17
FC	<u>\$DEL</u>	—	—	✓	指定字符串删除	9
FC	<u>\$CLR</u>	—	—	✓	字符串清除	3
FC	<u>\$INS</u>	—	—	✓	字符串插入	9-15
FC	—	<u>FMOD</u>	—	✓	浮点数转 BCD 浮点数	7-8
FC	<u>FREXP</u>	—	—	✓	BCD 浮点数转浮点数	7

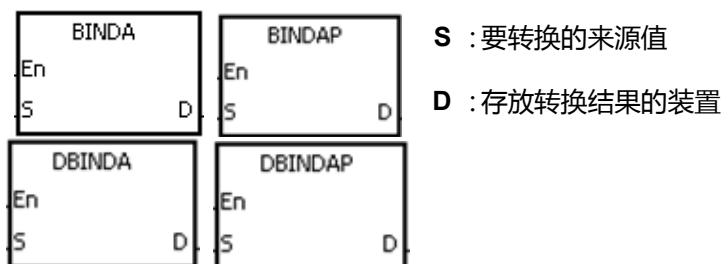
FB/FC	指令码			操作数	功能
FC	D*	BINDA	P	S, D	有号数十进制→ASCII 变换

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●/●*					●/●*							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

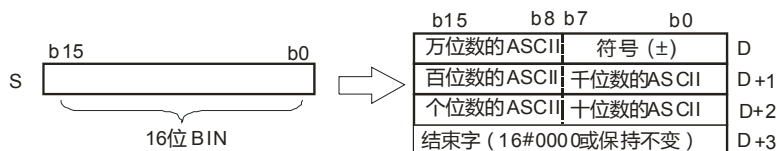
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



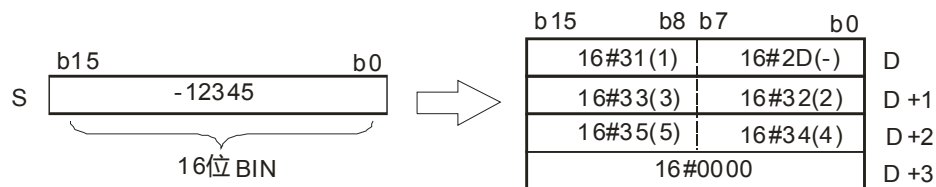
指令说明：

1. 将有号数十进制 BIN 值的数据源 S 做 ASCII 的转换，并储存于 D。
2. 本指令支持字符串结束模式标志 SM690，来控制字符串的终止符。
3. 16 位指令中，S 的范围-32768~32767，且固定为 6 位数的 BIN 值；D 连续占用四个 Word 装置，数据转换方式如图说明：

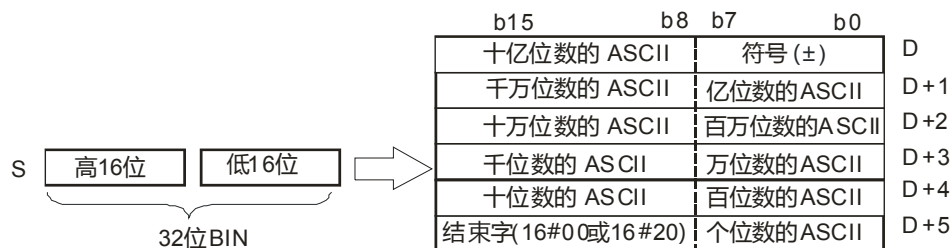


其中 SM690=OFF 时，D+3 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+3 的值保持不变。

如果 S 的内容值为正数，则 D 的符号会被填入 16#20；如果 S 的内容值为负数，则 D 的符号会被填入 16#2D。如下范例：S=-12345 且 SM690=OFF 则转换后的值如图说明：

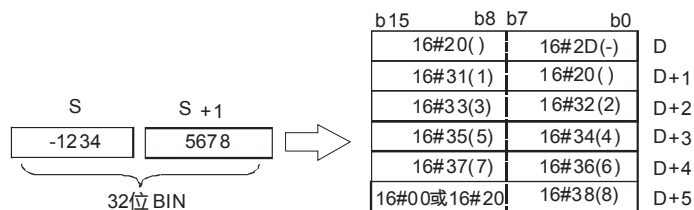


4. 32 位指令中，S 的范围-2147483648~2147483647，且固定为 11 位数的 BIN 值；D 连续占用六个 Word 装置，数据转换方式如图说明：

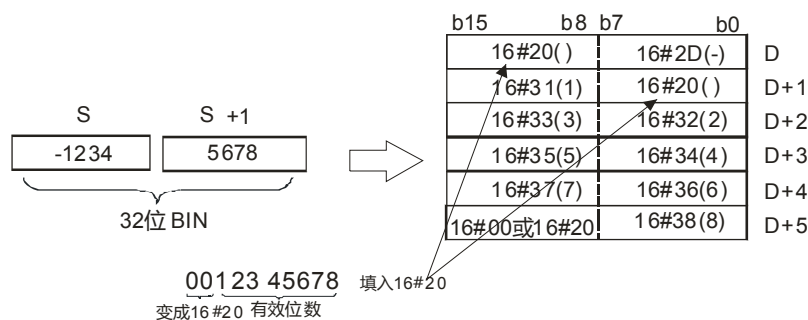


其中 SM690=OFF 时，D+5 的上 8 位填入 16#00；SM690=ON 时，D+5 的上 8 位填入 16#20。

如果 S 的内容值为正数，则 D 的符号会被填入 16#20；如果 S 的内容值为负数，则 D 的符号会被填入 16#2D。如下范例：S=-12345678 则转换后的值如图说明：

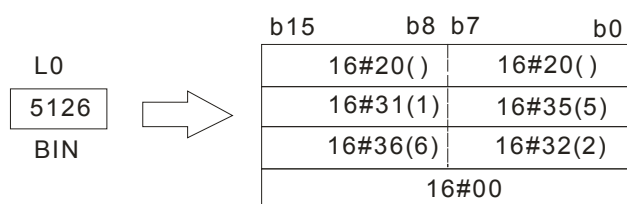
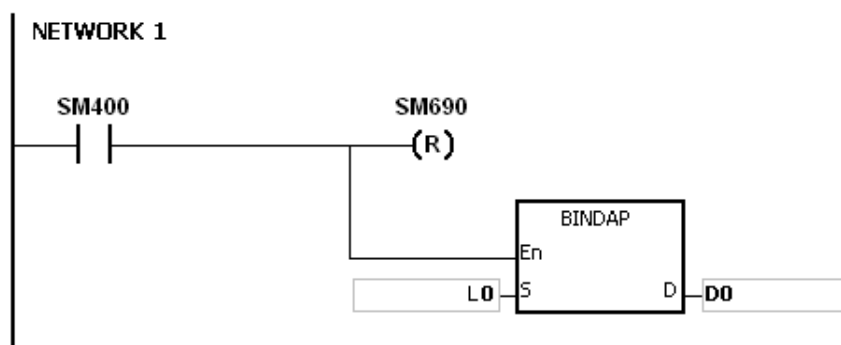


5. 若 S 内容值 (BIN) 如果有位数不足时，则 D 相对应的存放地址，会被补 16#20，例如：



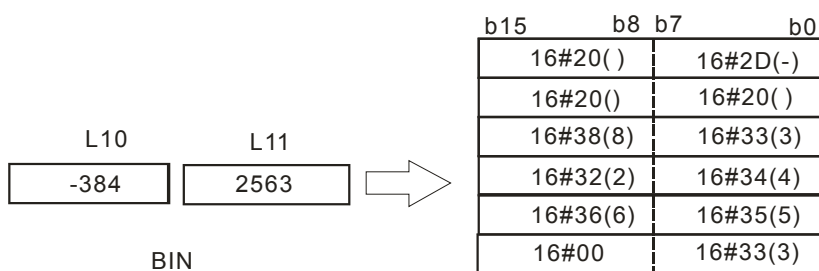
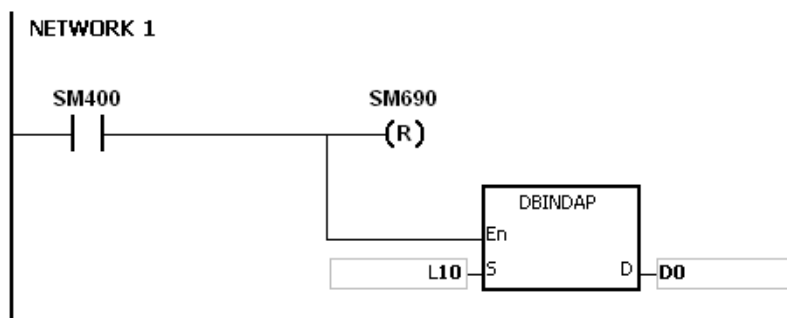
程序范例一：

给定 L0=5126 且 SM690=OFF，PLC RUN 时 D0=16#2020、D1=16#3135、D2=16#3632、D3=16#0000。



程序范例二：

给定 L10=-3842563 且 SM690=OFF，PLC RUN 时 D0=16#202D、D1=16#2020、D2=16#3833、D3=16#3234、D4=16#3635、D5=16#0033。



补充说明：

1. 16 位指令中 D+3 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
2. 32 位指令中 D+5 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
3. 16 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of WORD/INT。
4. 32 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [6] of WORD/INT。

FB/FC	指令码			操作数			功能								
FC	D*	BINHA	P	S, D			BIN 十六进制→十六进制 ASCII 变换								

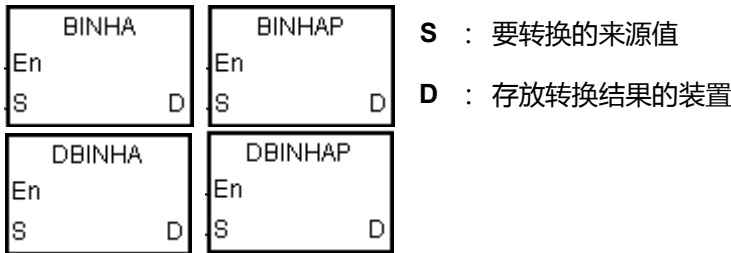
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●/●*					●/●*							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

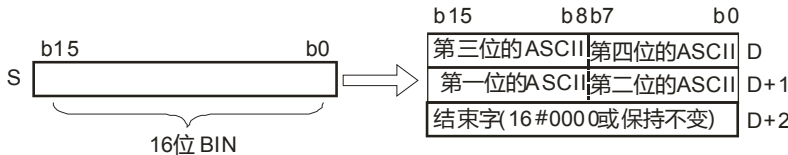
3

图形：

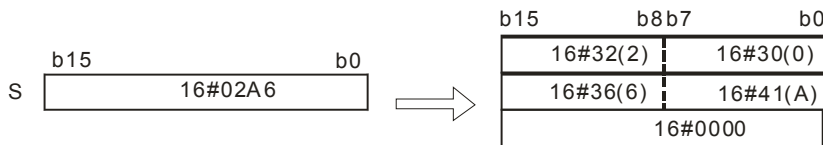


指令说明：

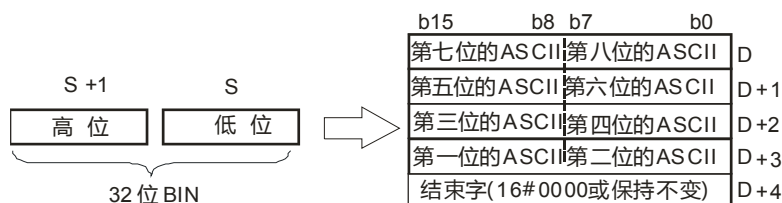
1. 将十六进制 BIN 值的数据源 S 做 ASCII 的转换，并储存于 D。
2. 本指令支持字符串结束模式标志 SM690，来控制字符串的终止符。
3. 16 位指令中，S 的范围 16#0000~16#FFFF，且固定为 4 位数的 BIN 值；D 连续占用三个 Word 装置，数据转换方式如图说明：



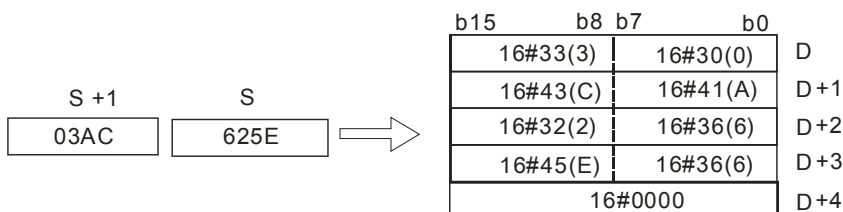
其中 SM690=OFF 时，D+2 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+2 的值保持不变。如下范例：S=16#02A6 且 SM690=OFF，则转换后的结果如下图所示。



4. 32 位指令中，S 的范围 16#00000000~16#FFFFFFFF，且固定为 8 位数的 BIN 值；D 连续占用五个 Word 装置，数据转换方式如图说明：

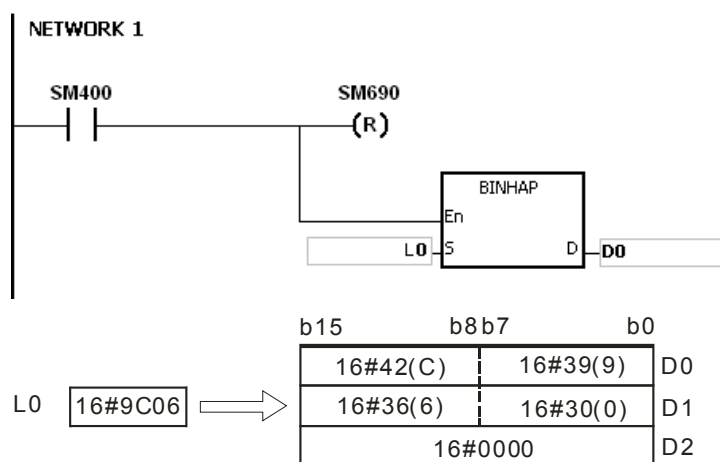


其中 SM690=OFF 时，D+4 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+4 的值保持不变。例如 S=16#03AC625E 且 SM690=OFF，则转换后的结果如下图所示。



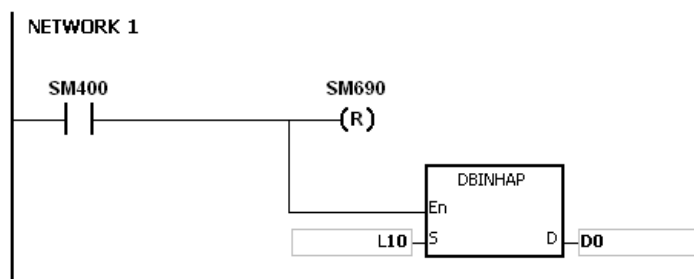
程序范例一：

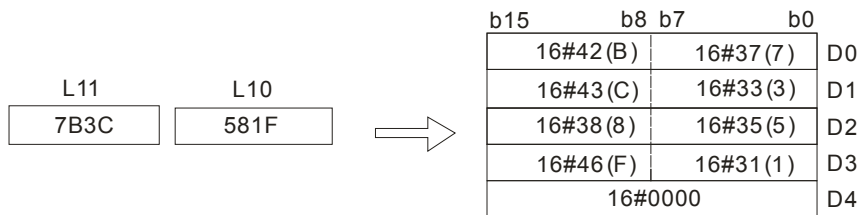
给定 L0=16#9C06 且 SM690=OFF，PLC RUN 时 D0=16#4239、D1=16#3630、D2=16#0000。



程序范例二：

给定 L10=16#7B3C581F 且 SM690=OFF，PLC RUN 时 D0=16#4237、D1=16#4333、D2=16#3835、D3=16#4631、D4=16#0000。





补充说明：

1. 16 位指令中 **D**+2 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
2. 32 位指令中 **D**+4 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
3. 16 位指令 **D** 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
4. 32 位指令 **D** 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [5] of WORD/INT。

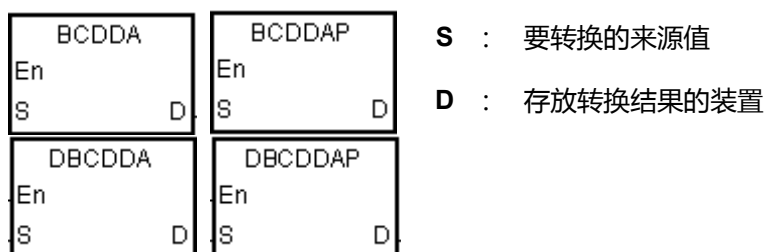
FB/FC	指令码			操作数	功能
FC	D*	BCDDA	P	S, D	BCD→ASCII 变换

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●/●*					●/●*							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

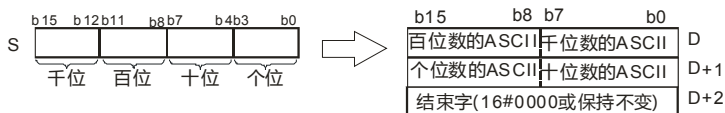
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



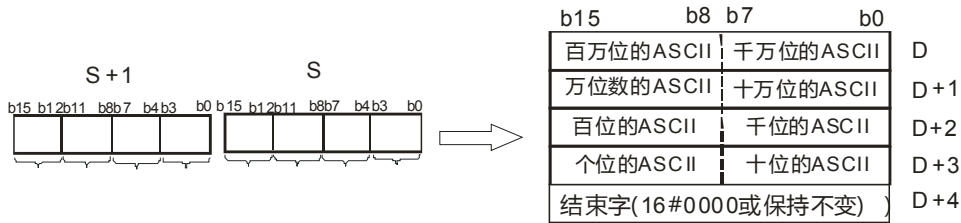
指令说明：

1. 将 BCD 值的数据源 S 做 ASCII 的转换，并储存于 D。
2. 本指令支持字符串结束模式标志 SM690，来控制字符串的终止符。
3. 16 位指令中，S 的范围 BCD 0~9999，且固定为 4 位数的 BCD 值；D 连续占用三个 Word 装置，数据转换方式如图说明：



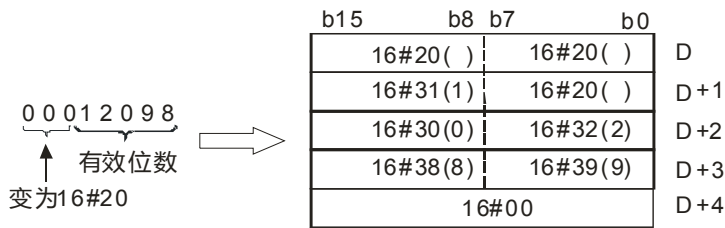
其中 SM690=OFF 时，D+2 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+2 的值保持不变。

4. 32 位指令中，S 的范围 BCD 0~99999999，且固定为 8 位数的 BCD 值；D 连续占用五个 Word 装置，数据转换方式如图说明：



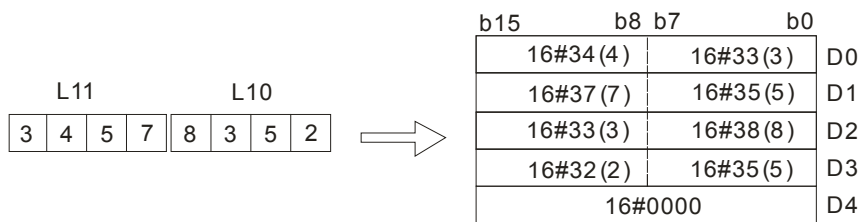
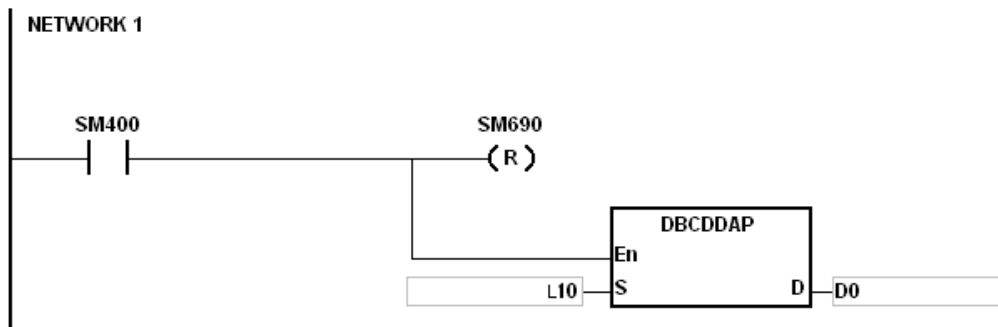
其中 SM690=OFF 时，D+5 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+5 的值保持不变。

5. 若 S 内容值如果有位数不足，则 D 相对应的存放地址，会被补 16#20，例如：



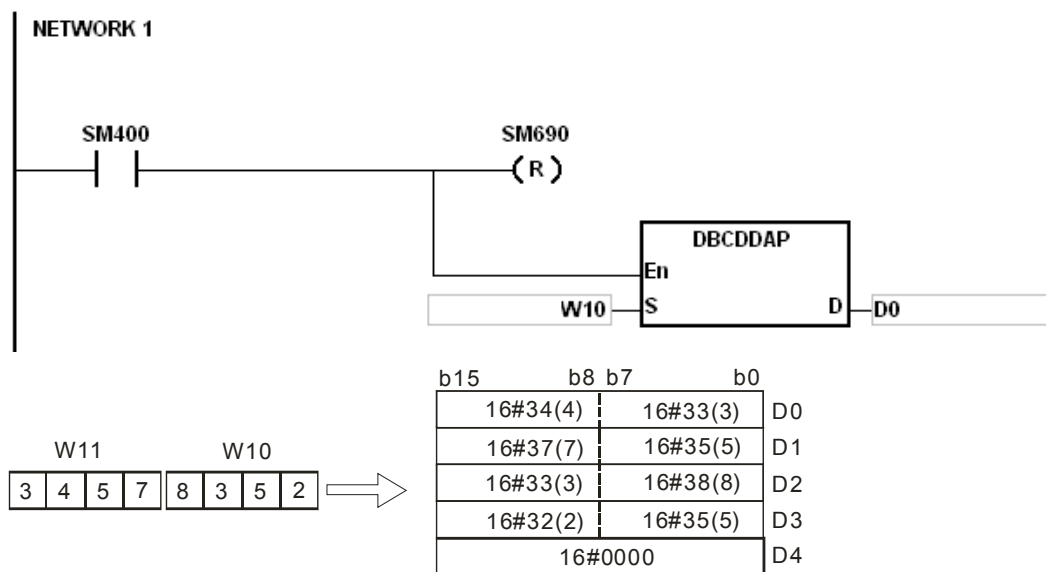
程序范例一：

给定 L10=BCD 1295 且 SM690=OFF，PLC RUN 时 D0=16#3231、D1=16#3539、D2=16#0000。



程序范例二：

给定 L10=BCD 34578352 且 SM690=OFF，PLC RUN 时 D0=16#3433、D1=16#3735、D2=16#3338、D3=16#3235、D4=16#0000。

**补充说明：**

1. 16 位指令中，若 **S** 内容值不在 0~9999，指令不执行，SM0=ON，错误码 SR0=16#200D。（BCD 值以 Hex 表示有任一位数不在 0~9 的范围内）。
2. 32 位指令中，若 **S** 内容值不在 0~99999999，指令不执行，SM0=ON，错误码 SR0=16#200D。（BCD 值以 Hex 表示有任一位数不在 0~9 的范围内）。
3. 16 位指令中 **D+2** 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
4. 32 位指令中 **D+4** 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
5. 16 位指令 **D** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
6. 32 位指令 **D** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [5] of WORD/INT。

FB/FC	指令码			操作数	功能
FC	D*	DABIN	P	S, D	有号数十进制 ASCII→有号数十进制 BIN 变换

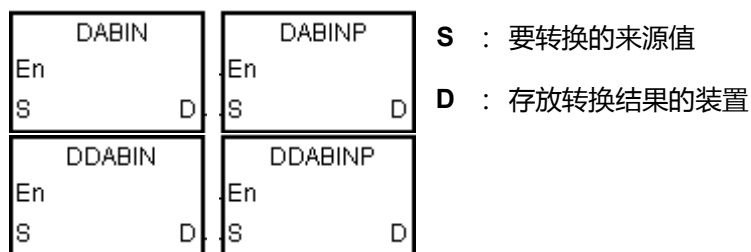
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●/●*					●/●*							
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

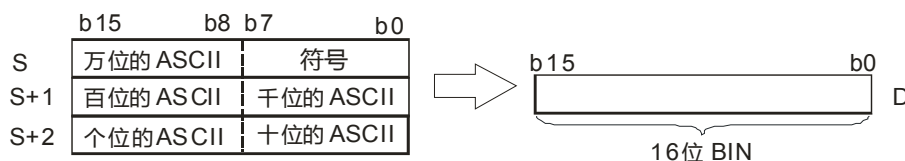
3

图形：

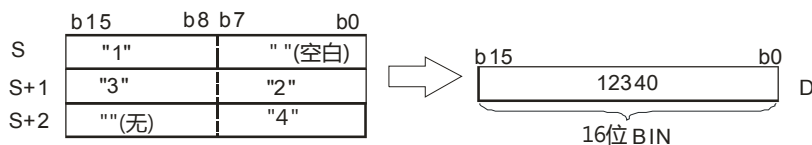


指令说明：

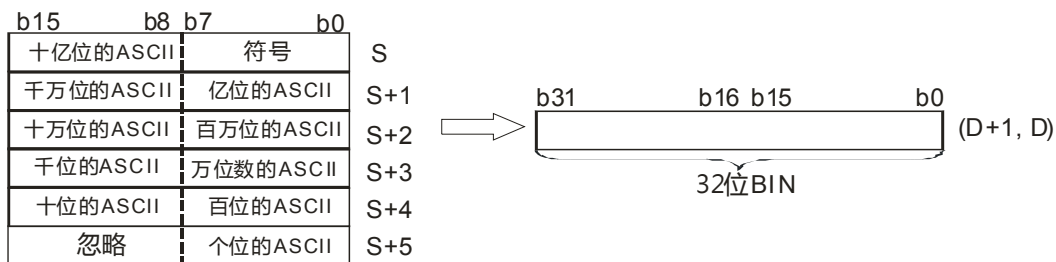
1. 将有号数十进制 ASCII 值的数据源 S 转换成有号数十进制 BIN，并储存于 D。
2. 16 位指令中，S 连续占用三个 Word 装置，且组成的 S 内容值范围 ASCII-32768~32767；若 S 来源如输入字符串，范围为“-32768”~“32767”。



3. 16 位指令中，若 S 来源为字符串，且不足 6 个字符串，则不足处视为“0”；第一个字符用来表示正负号，“ ”(空白)表示正数，“-”表示负数。如下范例“1234”：

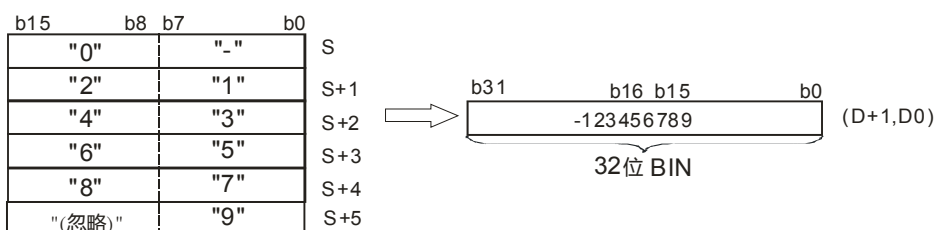


4. 32 位指令中，S 连续占用六个 Word 装置，且组成的 S 内容值范围 ASCII -2147483648~2147483647；若 S 来源如输入字符串，范围为“-2147483648”~“2147483647”。



5. 32 位指令中，若 **S** 来源为字符串，且不足 11 个字符串，则不足处视为“0”；第一个字符用来表示正负号，“ ”（空白）表示正数，“-”表示负数。

如下范例“-0123456789”：



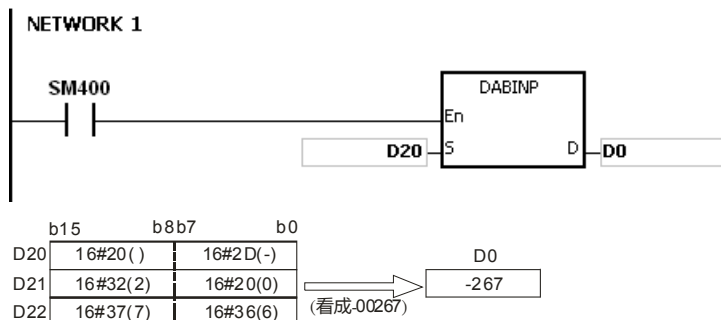
6. **S** 内容为 16#20 或 16#00 时，将当做 16#30 来做处理。

7. 符号为 16#20、16#30 或 16#2B 时，转换结果为正数，符号为 16#2D 时，转换结果为负数。

8. **S** 来源为字符串时，16 位指令长度为 1~6，32 位指令长度为 1~11。

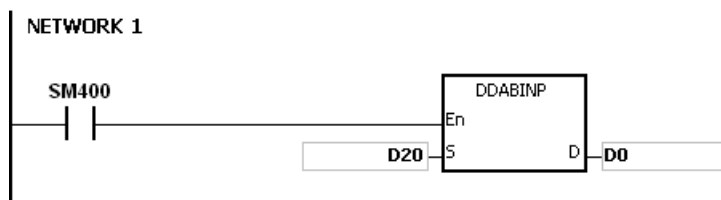
程序范例一：

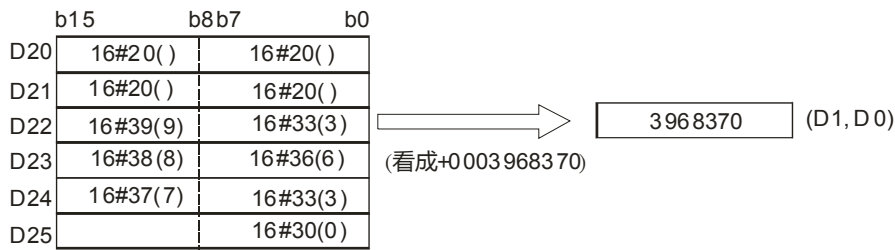
给定 D20=16#202D、D21=16#3220、D22=16#3736（即 ASCII-00267），PLC RUN 时，D0=-267。



程序范例二：

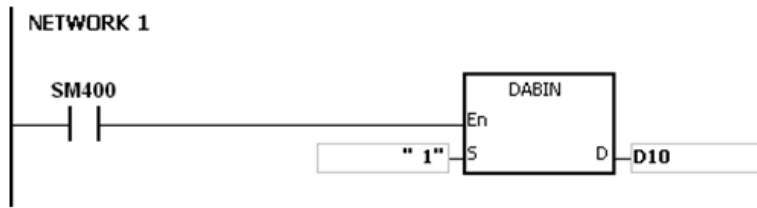
给定 D20=16#2020、D21=16#2020、D22=16#3933、D23=16# 3836、D24=16#3733、D25=16#3330（即 ASCII 0003968370），PLC RUN 时，D0=3968370。





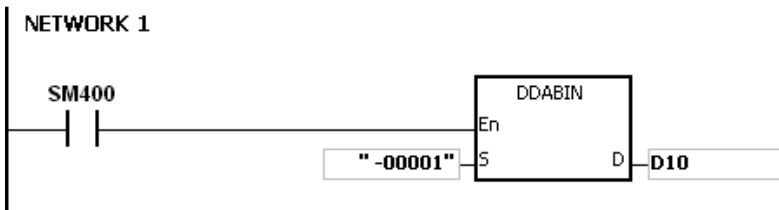
程序范例三：

S 数据源为“1”（String 类型），第一个字符需为“ ”（空白）表示正数，且“1”因为不足 6 个字符串，因此视为“10000”，PLC RUN 后 D10=10000。



程序范例四：

S 数据源为“-00001”（String 类型），第一个字符为“ ”（空白）表示正数，且“-00001”因为不足 11 个字符串，因此视为“-0000100000”，PLC RUN 后 (D11 , D10) =-100000。



补充说明：

1. 若 S 所指定的符号不是 16#20、16#30、16#2B 或 16#2D，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若 S 所指定的内容值不是 ASCII 码 16#30~16#39，16#20 或 16#0，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. S 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. 16 位指令中 S+2 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
5. 32 位指令中 S+5 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
6. 16 位指令 S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
7. 32 位指令 S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [6] of WORD/INT。

FB/FC	指令码			操作数	功能
FC	D*	HABIN	P	S, D	十六进制 ASCII→十六进制 BIN 变换

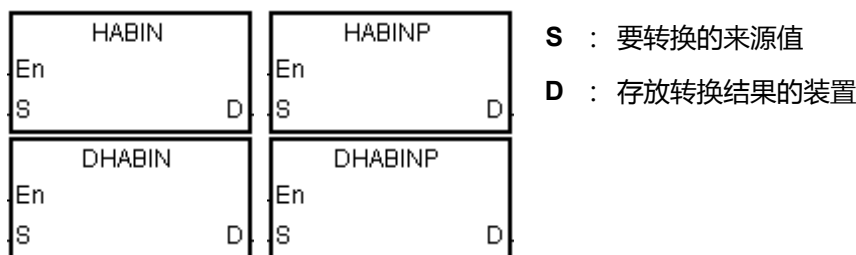
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●/●*					●/●*							
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

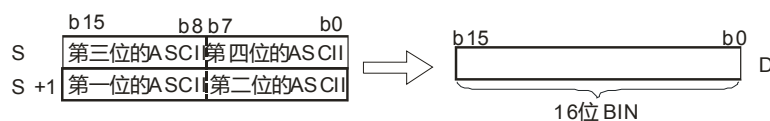
3

图形：

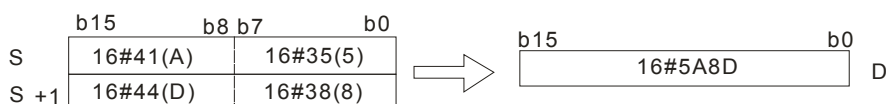


指令说明：

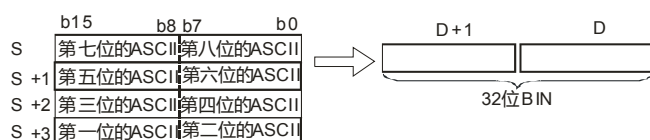
1. 数据源 S 的内容十六进制 (ASCII 值) 转换成十六进制 BIN, 存于 D。
2. 16 位指令 S 连续占用二个 Word 装置, S 内容值范围 ASCII 0000~FFFF; 若 S 来源如输入字符串, 范围为“0”~“FFFF”。



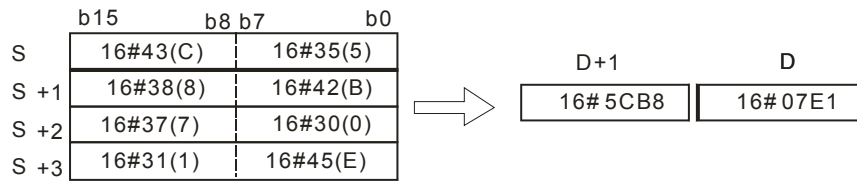
例如 S~S+1 的内容 ASCII 码分别为 5A8D, 则转换的结果为下图所示。



3. 32 位指令 S 连续占用四个 Word 装置, S 内容值范围 ASCII 00000000~FFFFFFFF; 若 S 来源如输入字符串, 范围为“0”~“FFFFFFFF”。



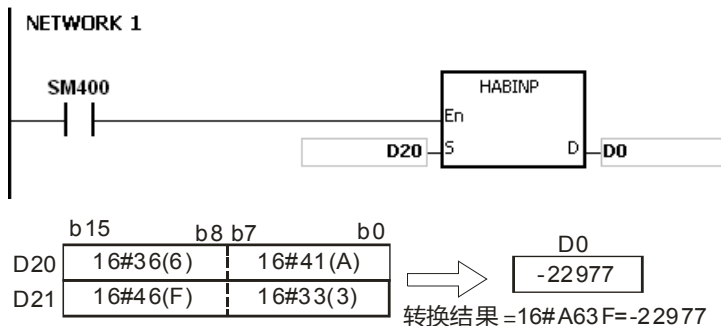
例如 S~S+3 内容 ASCII 码分别为 5CB807E1，则转换结果如下图所示。



4. S 来源为字符串时，16 位指令长度为 1~4，32 位指令长度为 1~8。

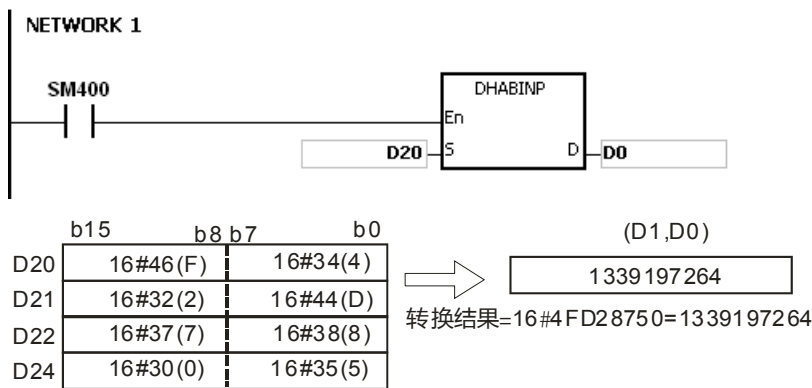
程序范例一：

给定 D20=16#3641、D21=16#4633 (即 ASCII 16#A63F=ASCII -22977)，PLC RUN 时，D0=-22977



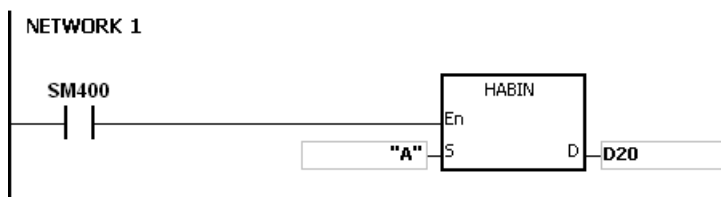
程序范例二：

给定 D20=16#4634、D21=16#3244、D22=16#3738、D23=16#3035 (即 ASCII 16#4FD28750=ASCII 1339197264)，PLC RUN 时，(D1, D0) =1339197264。



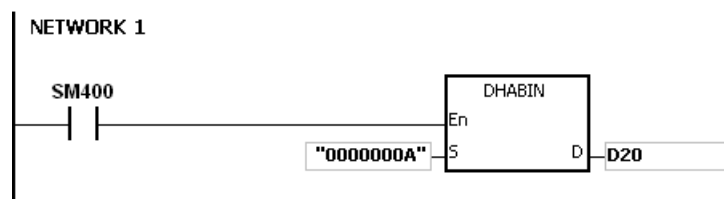
程序范例三：

S 数据源为“A”(String 类型)，因为不足 4 个字符串，因此视为“A000”，PLC RUN 后 D20=16#A000=-24576。



程序范例四：

S 数据源为“0000000A”(String 类型)，PLC RUN 后 (D21, D20) =16#0000000A=10。

**补充说明：**

1. S 内容值不是 ASCII 码 16#30~ 16#39 或 16#41~ 16#46 指令不执行, SM0=ON, 错误码 SR0=16#2003。
2. 16 位指令 S 操作数, 若使用 ISPSOft 宣告, 则数据类型为 ARRAY [2] of WORD/INT。
3. 32 位指令 S 操作数, 若使用 ISPSOft 宣告, 则数据类型为 ARRAY [4] of WORD/INT。

FB/FC	指令码			操作数				功能						
FC	D*	DABCD	P	S, D				BCD ASCII→BCD 变换						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●*				●	●*						
D		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

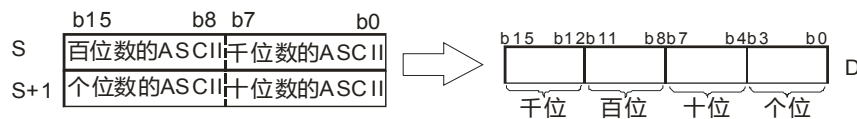
3

图形：

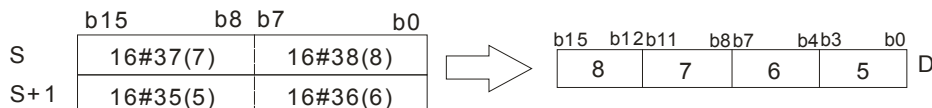


指令说明：

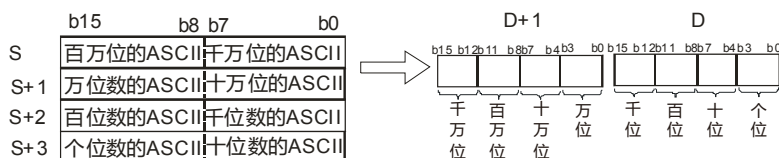
1. 数据源 S 的内容 (ASCII 值) 转换成 BCD , 存于 D.
2. 16 位指令 S 连续占用二个 Word 装置, S 内容值范围 ASCII 0000~9999 ; 若 S 来源如输入字符串, 范围为“0”~“9999”。



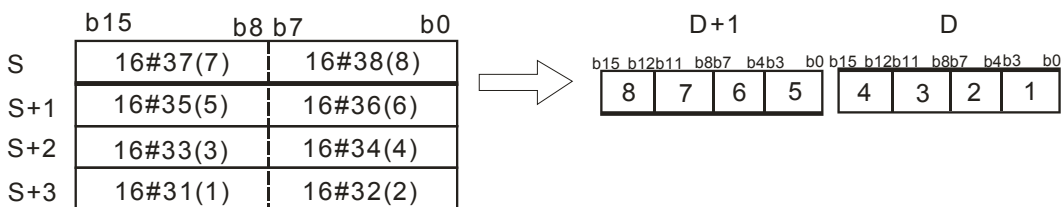
3. 例如, S~S+N 的内容分别为 ASCII 码 8765 , 则转换结果如下图所示。



4. 32 位指令 S 连续占用四个 Word 装置, S 内容值范围 ASCII 0000000~99999999 ; 若 S 来源如输入字符串, 范围为“0”~“99999999”。



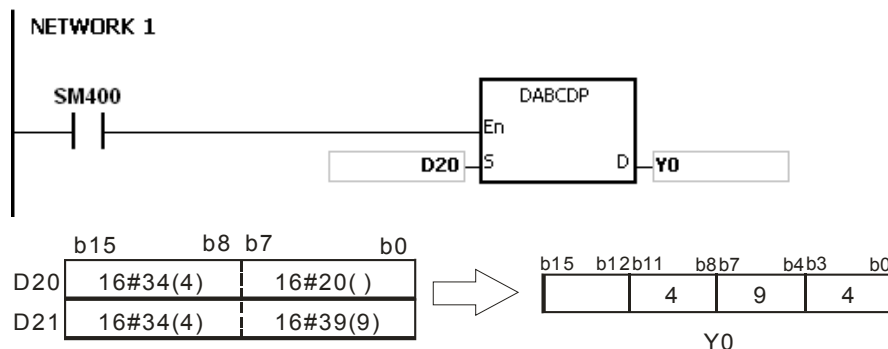
例如，S~S+N 的内容分别为 ASCII 码 87654321，则转换结果如下图所示。



5. S 内容为 16#20 或 16#00 时，将当做 16#30 来做处理。
6. S 来源为字符串时，16 位指令长度为 1~4，32 位指令长度为 1~8。

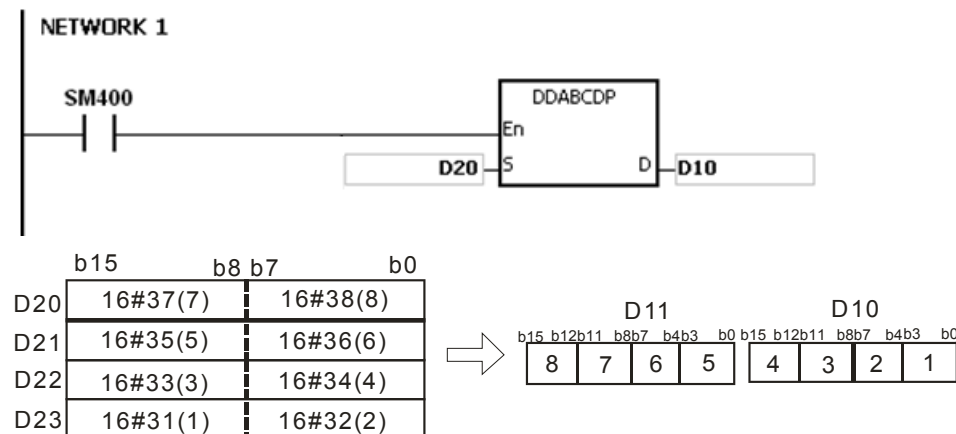
程序范例一：

给定 D20=16#3420、D21=16#3439（即 ASCII 494），PLC RUN 时，转换为 Y0=16#494。



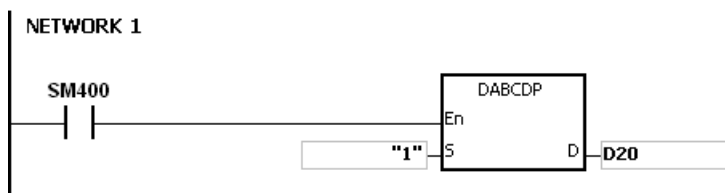
程序范例二：

给定 D20=16#3738、D21=16#3536、D22=16#3334、D23=16#3132（即 ASCII 87654321），PLC RUN 时，转换为 (D11, D10) = 16#87654321。



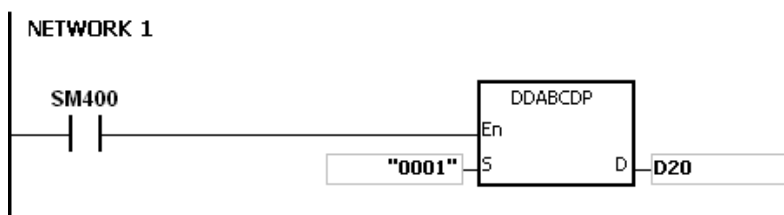
程序范例三：

S 数据源为“1”(String 类型) ,因为不足 4 个字符串 ,因此视为“1000” ,PLC RUN 时 ,转换为 D20=16#1000。



程序范例四：

S 数据源为“0001”(String 类型) ,因为不足 8 个字符串 ,因此视为“00010000” ,PLC RUN 时 ,转换为(D21 , D20=16#10000。



补充说明：

1. S 内容值不是 ASCII 码 16#30~16#39 指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. S 字符串长度超出范围时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. 16 位指令 S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
4. 32 位指令 S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of WORD/INT。

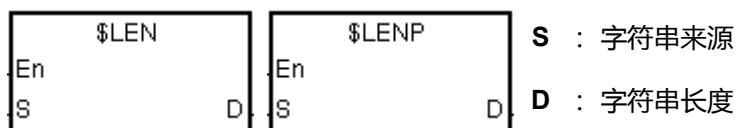
FB/FC	指令码		操作数				功能				
FC		\$LEN	P	S, D				计算字符串长度			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														●
D		●	●				●	●						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●			○	
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



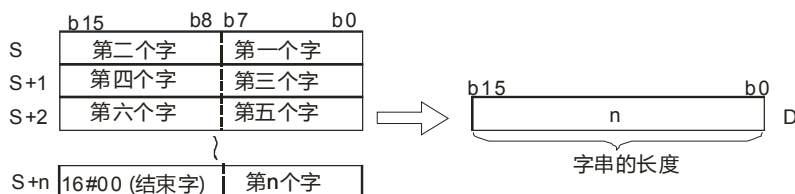
指令说明：

1. 计算数据源 **S** 的内容（字符串）长度，一直计算到字符串结尾 16#00 之前的长度，并将字符串长度存于 **D**。

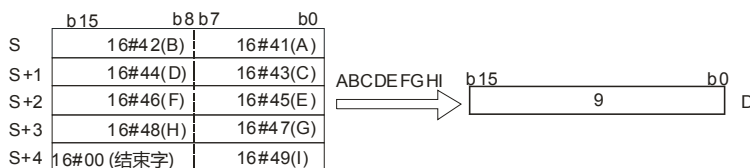
2. 字符串长度=0~65535，

如果字符串长度=65536=16#10000，则 D=0。

如果字符串长度=65537=16#10001，则 D=1。

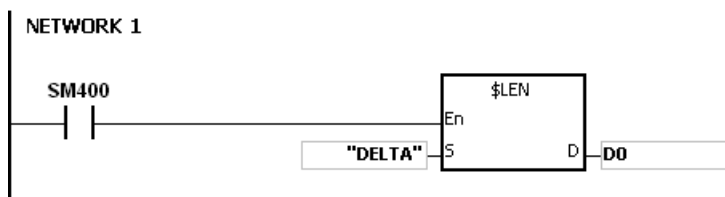


例如 S~S+n 的内容为 ABCDEFGHI，则计算结果如下图所示。



程序范例一：

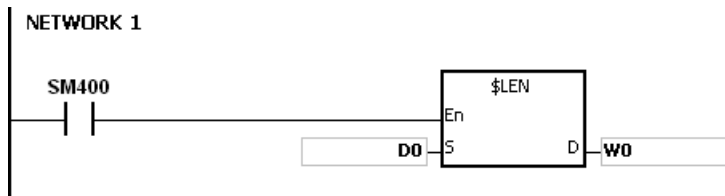
S 数据源给定'DELTA'，则 PLC 执行后 D0=5。



程序范例二：

S 数据源给定 D0~D2 如下表，则 PLC 执行后 L0=5。

D0	16#45 (E)	16#44 (D)
D1	16#54 (T)	16#4C (L)
D2	16#00 (终止符)	16#44 (A)



补充说明：

字符串的字尾无 16#00 当做结束符号，指令不执行，SM0=ON，错误码 SR0=16#200E。

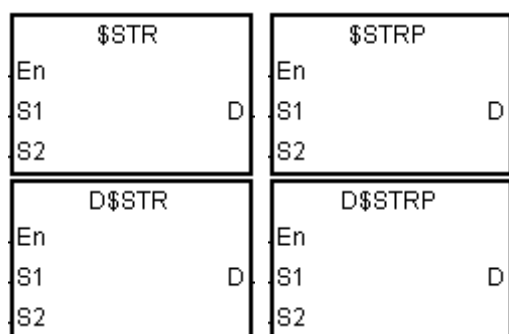
FB/FC	指令码			操作数			功能									
FC	D*	\$STR	P	S ₁ , S ₂ , D			BIN→String									

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●	●*				●	●*						
S ₂		●	●*				●	●*						
D		●/●*					●/●*							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●		●	○	●				
S ₂	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

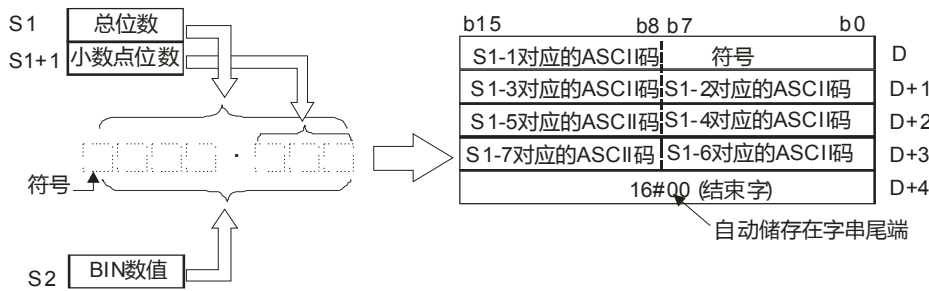
图形：



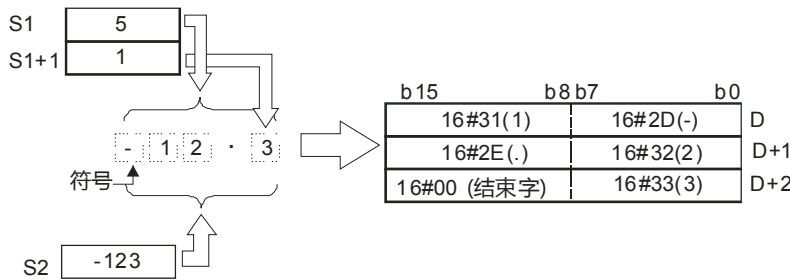
S₁ : 要转换的位数装置起始编号
 S₂ : 要被转换的来源
 D : 存放转换结果的装置起始编号

指令说明：

- S₂ 的内容加上一个小数点，小数点后的位数由 S₁+1 所指定，并将转换后的结果依照 S₁ 所指定的总长度存于 D。
- \$STR :
 - S₁ 总长度范围=2~8。
 - S₁+1 所指定的小数点后的位数，范围=0~5，而且 S₁+1 必须≤S₁-3。
 - S₂ 的范围=-32768~32767。



例如总位数为 5，小数点位数为 1，内容为-123，转换后的结果如下图所示。

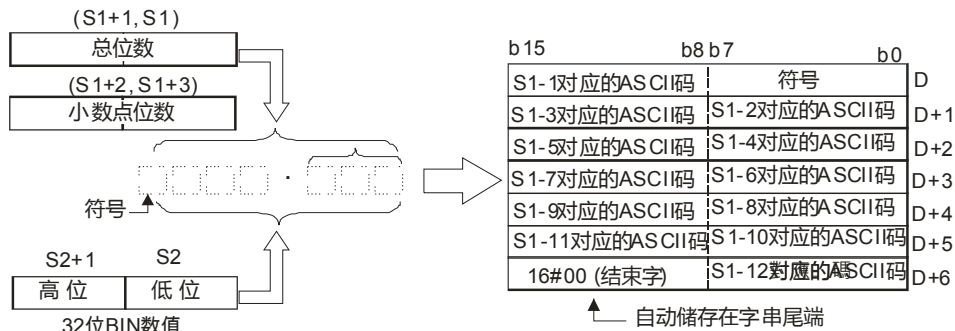


3. D\$STR :

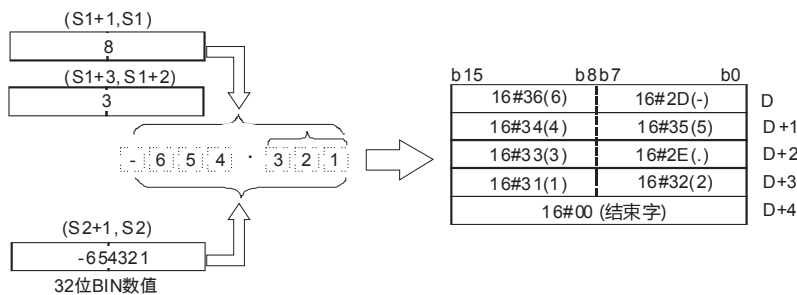
S₁ 总长度范围=2~13。

S₁+1 所指定的小数点后的位数，范围=0~10，而且 S₁+1 必须<= S₁-3。

S₂ 的范围=-2147483648~2147483647。



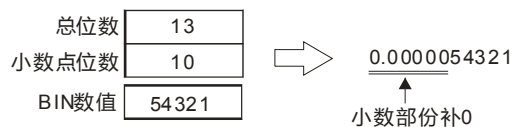
例如总位数为 8，小数点位数为 3，内容为-654321，转换后的结果如下图所示。



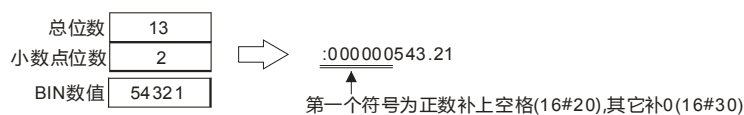
4. 如果 S₂ 为正数，则 D 的符号位置被填入 16#20，为负数，填入 16#2D。

5. D 的小数点位置被填入 16#2E。

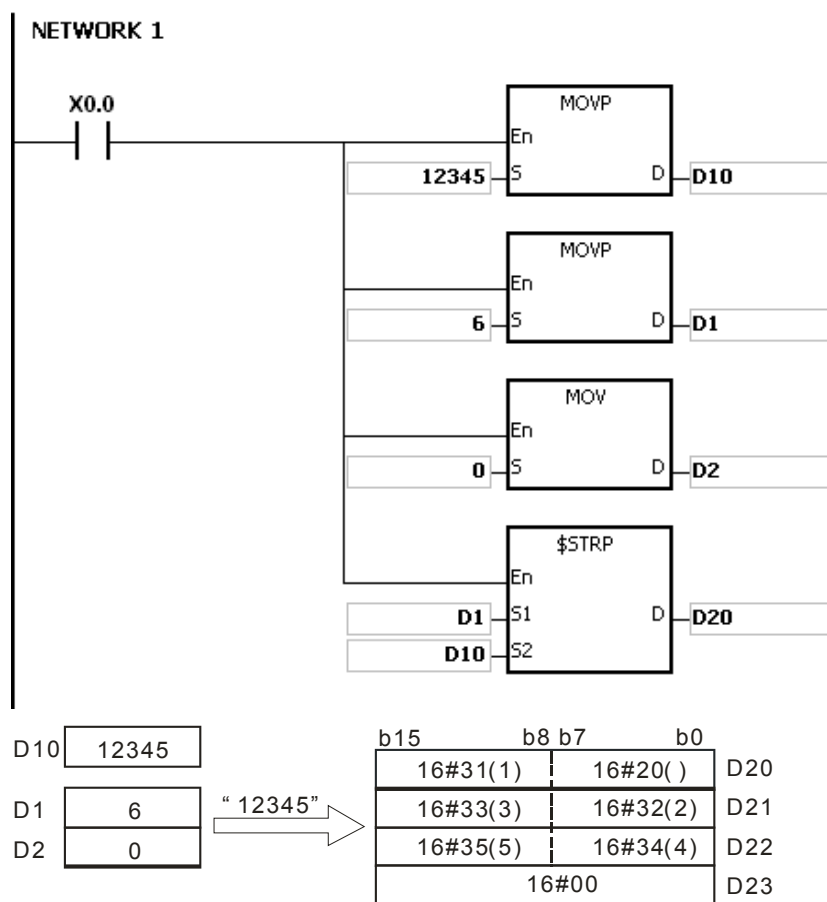
6. 如果 S_1+1 的位数 $> S_2$ 的位数时，空出的位数自动补 0。



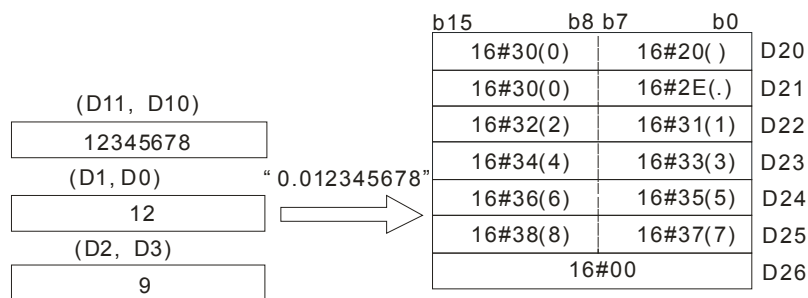
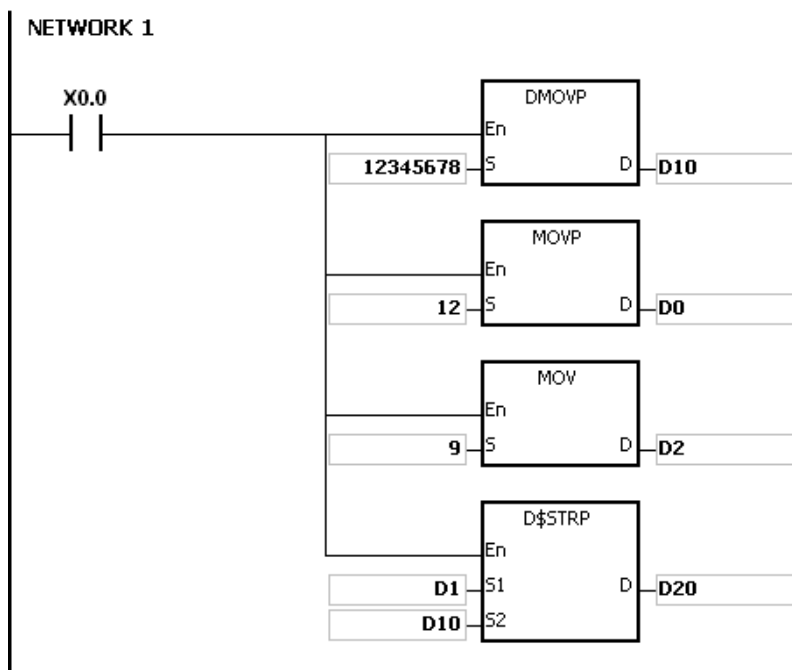
7. 如果 S_1 总位数 $> S_2$ 加上小数点跟符号的位数时，空出的位数自动补上‘0’。



程序范例一：



程序范例二：



补充说明：

1. S_1 总长度超出范围时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. S_{1+i} 位数超出范围时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. S_{1+i} 必须 $\leq S_1-3$ ，不符合此范围时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
4. S_1 总位数 $< S_2$ 加上小数点跟符号的位数时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
5. 16 位指令 S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
6. 32 位指令 S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of DWORD/DINT。

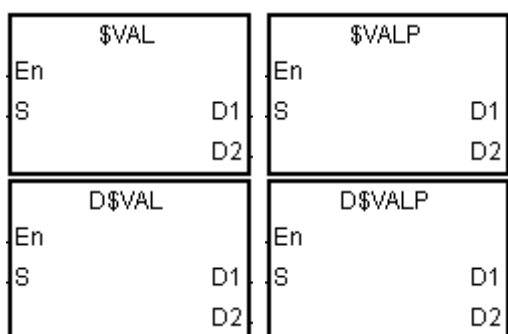
FB/FC	指令码			操作数	功能
FC	D*	\$VAL	P	S, D ₁ , D ₂	String→BIN

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														●/●*
D ₁		●	●*				●	●*						
D ₂		●	●*				●	●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
D ₁	●	●			●	●		●	●				●				
D ₂	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	AH 运动控制主机

图形：



S : 要转换的来源值

D₁ : 存放转换后的位数

D₂ : 存放转换后的 BIN 值

指令说明：

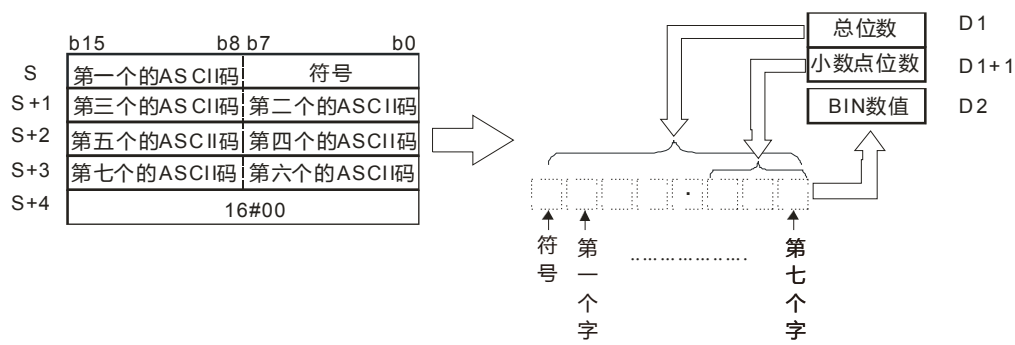
1. S 的内容字符串，转换成 BIN 值，并将总位数存放在 D₁，小数点后有几位数存放在 D₁+1，BIN 值存放在 D₂。

2. \$VAL：

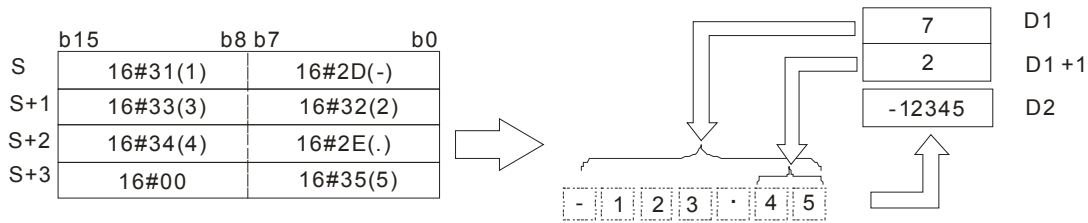
S 最多占用五个 Word 装置。

S 字符串总长度范围=2~8。

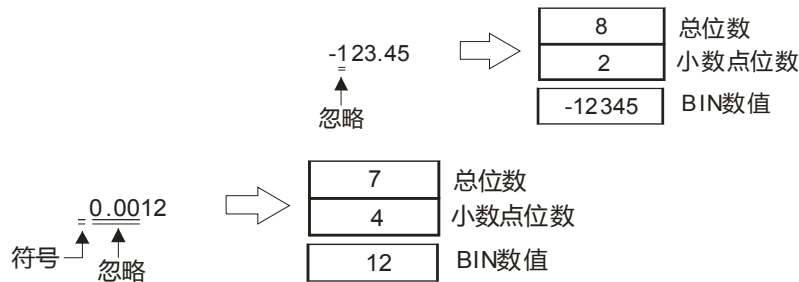
如果 S 字符串中有小数点，则小数点 16#2E 的位置，需在 3~(字符串总长度-1)之中。



例如 S~S+3 的内容分别为-123.45，则运算结果如下图所示。



S 字符串在符号与第一个非零数值之间 如果有 16#20 或 16#30 则在转换成 BIN 数值时 会忽略 16#20 或 16#30。



忽略小数点的点数 (16#2E) 不看时，则 S 字符串的范围=-32768~32767。例如：“1235.3”就是要检查“12353”是否在 S 的范围内。

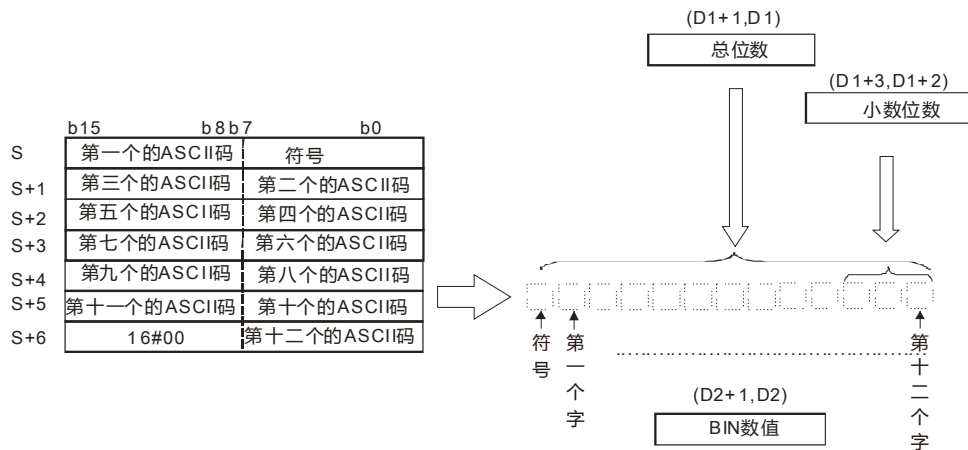
3. D\$VAL :

S 最多占用七个 Word 装置。

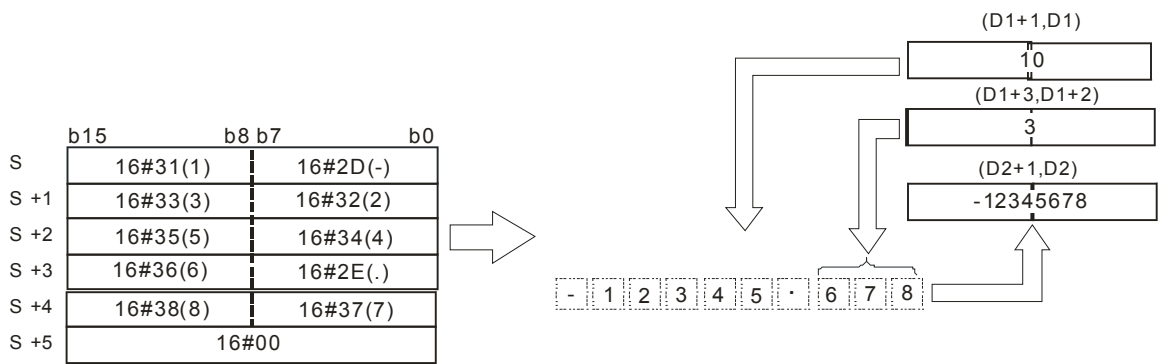
字符串总长度范围=2~13。

如果 S 字符串中有小数点，则小数点 16#2E 的位置，需在 3~ (字符串总长度-1) 之中。

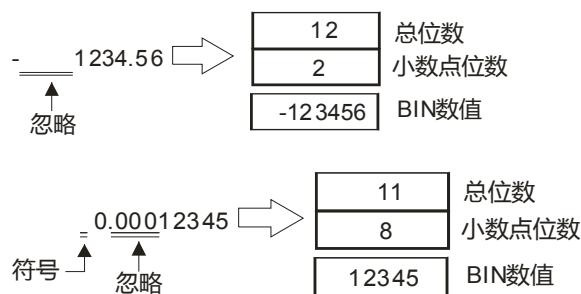
忽略小数点的点 (16#2E) 不看时，则 S 字符串的范围=-2147483648~2147483647。



例如 S~S+n 的内容分别为-12345.678，则运算结果如下图所示。



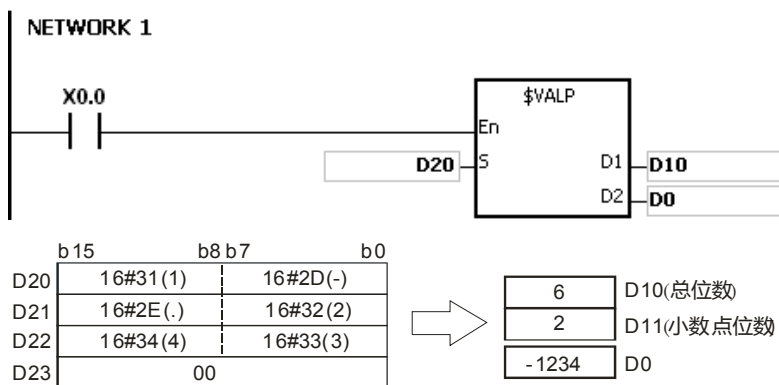
S 字符串在符号与第一个非零数值之间 如果有 16#20 或 16#30 则在转换成 BIN 数值时 会忽略 16#20 或 16#30。



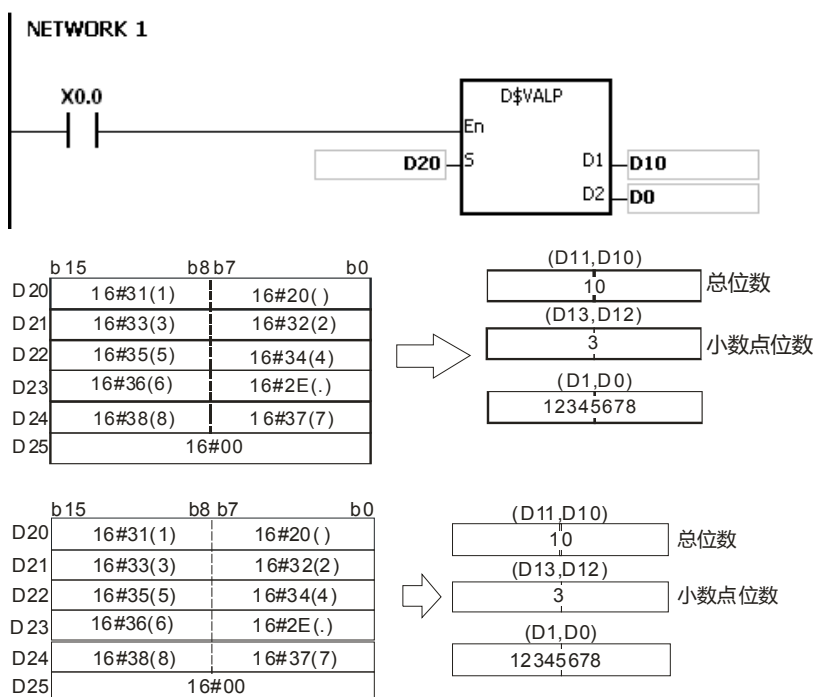
忽略小数点的点(16#2E)不看时,则 S 字符串的范围=-2147483648~2147483647。例如 :“1234567.8”就是要检查“12345678”是否在 S 的范围内。

4. 如果 S 的符号位置被填入 16#20、16#2B、16#30，则转换结果为正数，填入 16#2D，则转换结果为负数。
5. S 字符串除了符号、小数点，还有可能会被忽略的部份可以填入 16#20 或 16#30，其余只能填入 16#30~16#39。

程序范例一：



程序范例二：



补充说明：

1. S 字符串长度超出范围时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. S 所指定的符号不是 16#20、16#30、16#2B 或 16#2D，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. S 字符串中的小数点符号位置不正确，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
4. S 字符串转换后的 BIN 值超出范围，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
5. S 字符串除了符号、小数点，还有可能会被忽略的部份可以填入 16#20 或 16#30，其余要转成 BIN 值的部份不在 16#30~16#39 范围内，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
6. 16 位指令 D1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
7. 32 位指令 D1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of DWORD/DINT。

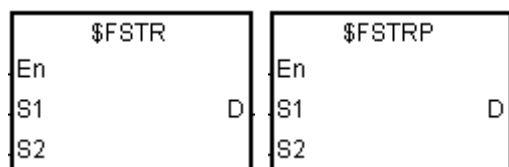
FB/FC	指令码			操作数				功能					
FC	\$FSTR	P		S ₁ , S ₂ , D				Float→String					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁														●
S ₂		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●		●				○
S ₂	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
-	AH 运动控制主机	-

图形：



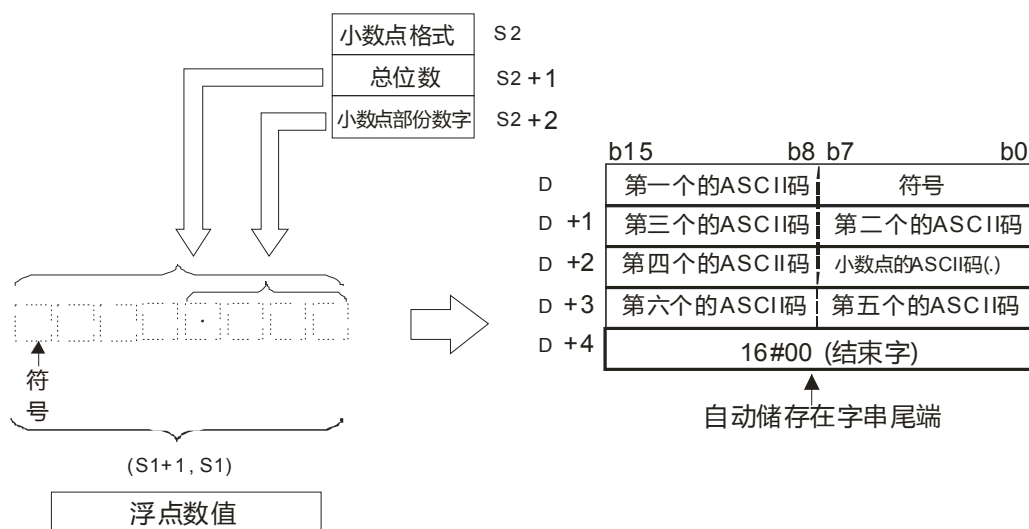
- S₁ : 要被转换的来源浮点数
- S₂ : 要转换的显示设置装置起始编号
- D : 存放转换结果的装置起始编号

指令说明：

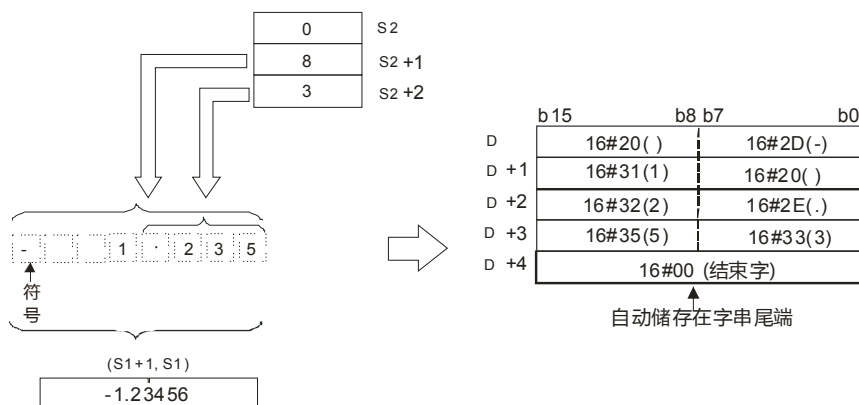
1. 依据 S₂ 的显示设置，将 S₁ 的浮点值转换成字符串，并将结果存于 D。
2. 转换的结果依据 S₂ 的显示设置而有所不同。
3. 总位数 S₂₊₁ 设定范围为 2~24。

S ₂	0: 小数格式 1: 指数格式
S ₂₊₁	总位数
S ₂₊₂	小数部份位数

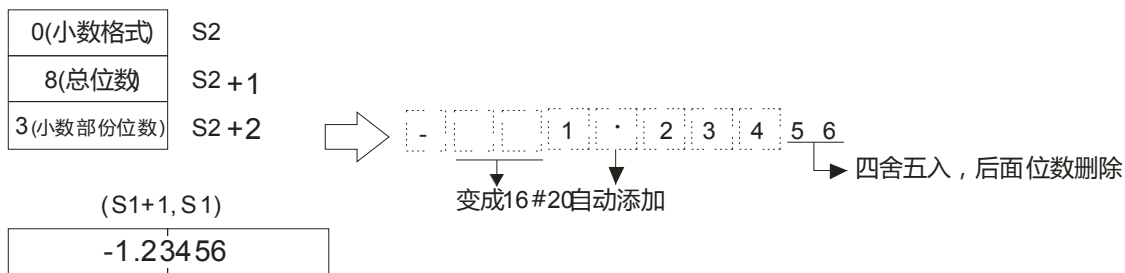
4. 当使用小数格式 (S₂=0) 时：



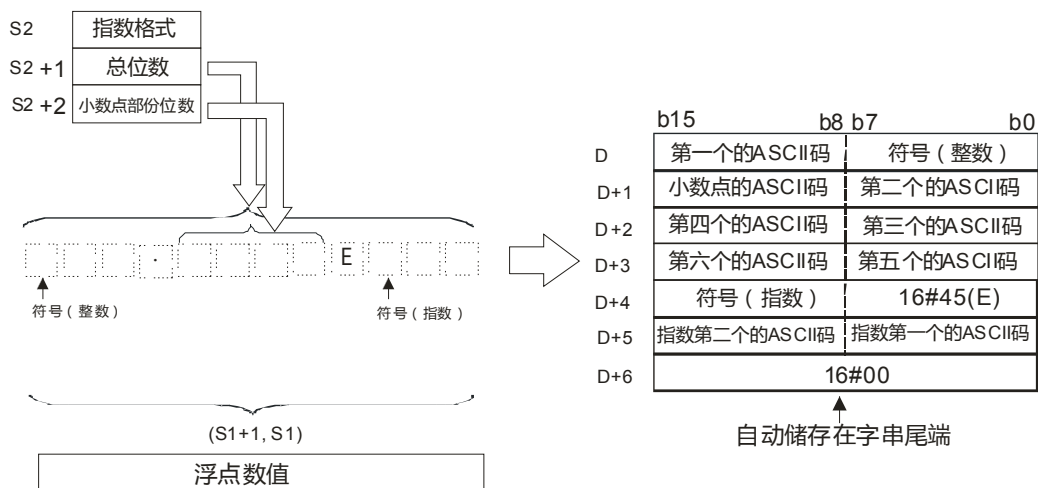
举例说明，总位数 8 位，小数点位数 3 位，数值为 -1.23456，则运算结果如下所示：



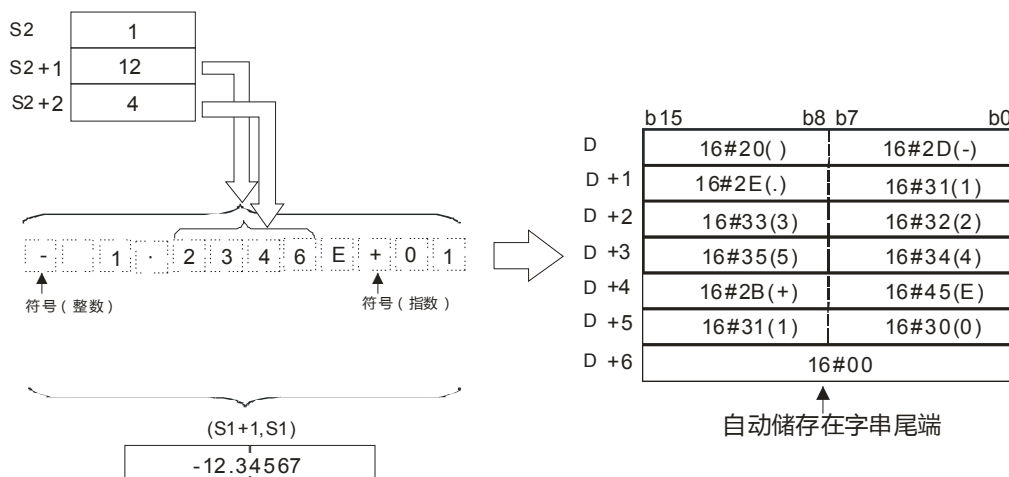
- 总位数 S_{2+1} 可以指定的范围如下：
 当小数点后的位数 $S_{2+2}=0$ 时， $2 \leq S_{2+1} \leq 24$ ，整数部分的位数 ≤ 23 。
 当小数点后的位数 S_{2+2} 不等于 0 时， $S_{2+2}+3 \leq S_{2+1} \leq 24$ ，整数部分的位数 $\leq 22-S_{2+2}$ 。
- 小数点后的位数 S_{2+2} 可以指定的范围为：0~7，但是 S_{2+2} 不等于 0 时， S_{2+2} 必需要 $\leq (S_{2+1}$ 总位数) -3。
- 如果来源浮点数 S_1 为正数，则 D 的符号位置被填入 $16\#20$ ，为负数，填入 $16\#2D$ 。
- 如果浮点数的长度超出总位数 S_2+1 ，则低位的数字会被删除，并做四舍五入。
- 如果小数点后的位数 $S_2+2>0$ ，则 $16\#2E$ (“.”)，会自动存放在指定的位数之前。
- 转换后的结果，如果长度小于总位数，则从符号之后到浮点数实数之前的位数会被填入 $16\#20$ 。
- 在转换结果的字符串最后面，将自动加上终止符 $16\#00$ 。



5. 当使用指数格式 (S₂=1) 时 :

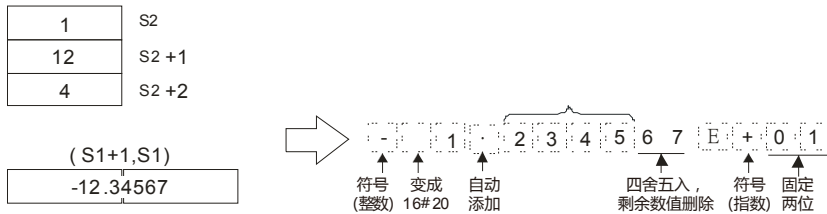


举例说明, 总位数 12 位, 小数点位数 4 位, 数值为-12.34567, 则运算结果如下所示 :



- 总位数 S₂₊₁ 可以指定的范围如下 :
 当小数点后的位数 S₂₊₂=0 时, 6 ≤ S₂₊₁ ≤ 24。
 当小数点后的位数 S₂₊₂ 不等于 0 时, S₂₊₂+7 ≤ S₂₊₁ ≤ 24。
- 小数点后的位数 S₂₊₂ 可以指定的范围为 : 0~7, 但是 S₂₊₂ 不等于 0 时, S₂₊₂ 必需要 ≤ (S₂₊₁ 总位数) - 7。
- 如果来源浮点数 S₁ 为正数, 则 D 的符号位置被填入 16#20, 为负数, 填入 16#2D。

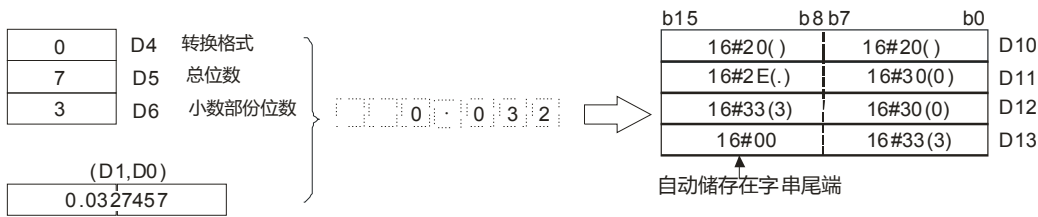
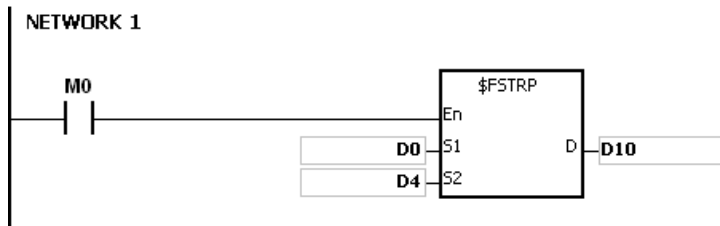
- 整数部份固定为 1 个位数，并且在符号与整数之间自动填入 16#20，以补足总位数。
- 如果小数点后的位数 $S_2+2>0$ ，则 16#2E (“.”)，会自动存放在指定的位数之前。
- 如果指数值为正数，则 D 的指数符号位置被填入 16#2B，为负数，填入 16#2D。
- 指数部份固定为 2 位数，如果只有 1 位数，则被填入 16#30 (“0”)。
- 在转换结果的字符串最后面，将自动加上 16#00 当做字符串的结尾。



3

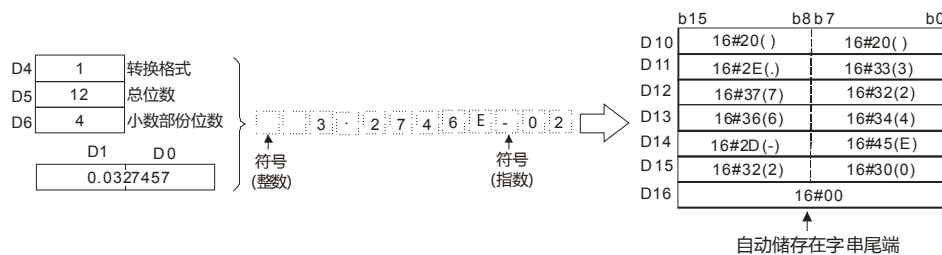
程序范例一：

将浮点值内容 (D1 , D0) 转为小数形式 (D4=0) 的字符串。



程序范例二：

将浮点值内容 (D1 , D0) 转为指数形式 (D4=1) 的字符串。



补充说明：

1. 当 S_1 超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。
2. 当 S_2 符号所指定的值不是 0 也不是 1，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 当总位数 S_{2+1} 超出以下范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
 - 当使用小数格式时
 - 当小数点后的位数 $S_{2+2}=0$ 时， $2 \leq S_{2+1} \leq 24$ ，**整数部分的位数 ≤ 23** 。
 - 当小数点后的位数 S_{2+2} 不等于 0 时， $S_{2+2}+3 \leq S_{2+1} \leq 24$ ，**整数部分的位数 $\leq 22-S_{2+2}$** 。
 - 当使用指数格式时
 - 当小数点后的位数 $S_{2+2}=0$ 时， $6 \leq S_{2+1} \leq 24$ 。
 - 当小数点后的位数 S_{2+2} 不等于 0 时， $S_{2+2}+7 \leq S_{2+1} \leq 24$ 。
4. 当小数点后的位数 S_{2+2} 超出以下范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
 - 当使用小数格式时：
 - 0~7，但是 S_{2+2} 必需要 $\leq (S_{2+1} \text{ 总位数}) - 3$ 。
 - 当使用指数格式时：
 - 0~7，但是 S_{2+2} 必需要 $\leq (S_{2+1} \text{ 总位数}) - 7$ 。
5. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

FB/FC	指令码		操作数				功能			
FC	\$FVAL	P	S, D				String→Float			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														●
D			●					●						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
D	●	●			●	●		●	●		●		●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

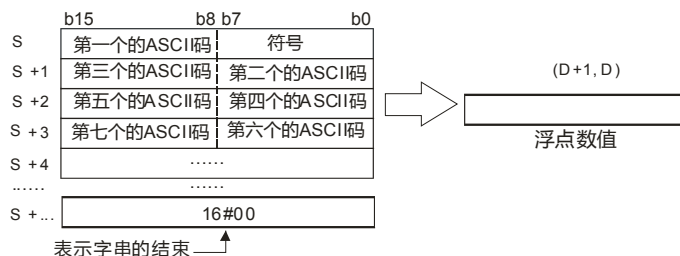
3

图形：



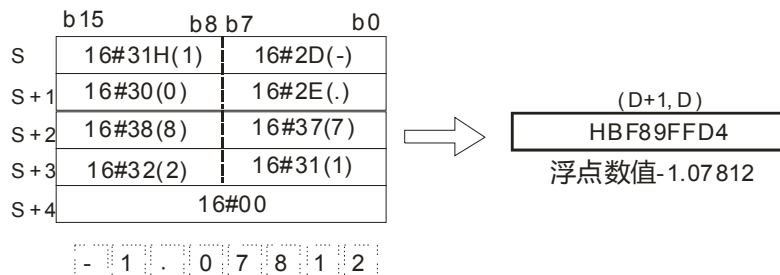
指令说明：

1. 将 S 的浮点数字串转换成浮点值，并将结果存于 D。

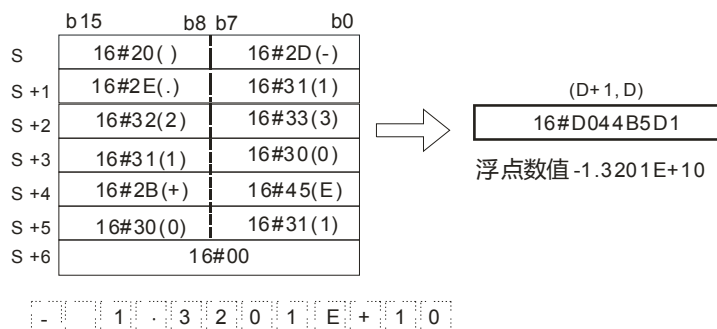


2. S 的浮点数字串分为小数格式跟指数格式两种，格式会由主机自动根据 S 的输入判别是哪一种。

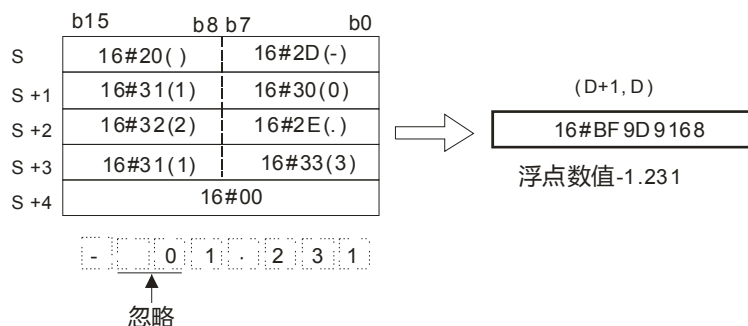
- 当使用小数格式时，范例如下：



- 当使用指数格式时，范例如下：

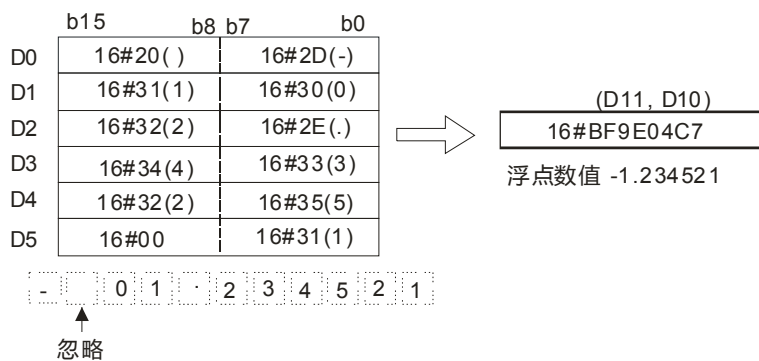
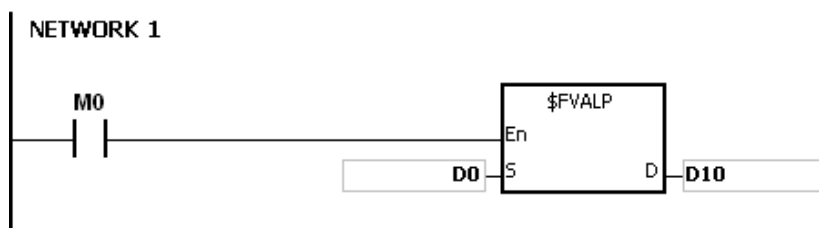


3. 如果 **S** 的符号位置被填入 16#20、16#30 或 16#2B，则转换结果为正数，填入 16#2D，则转换结果为负数。
4. **S** 的字符串中除了起始的零之外，其余的 16#20 和 16#30 在转换时全部忽略。

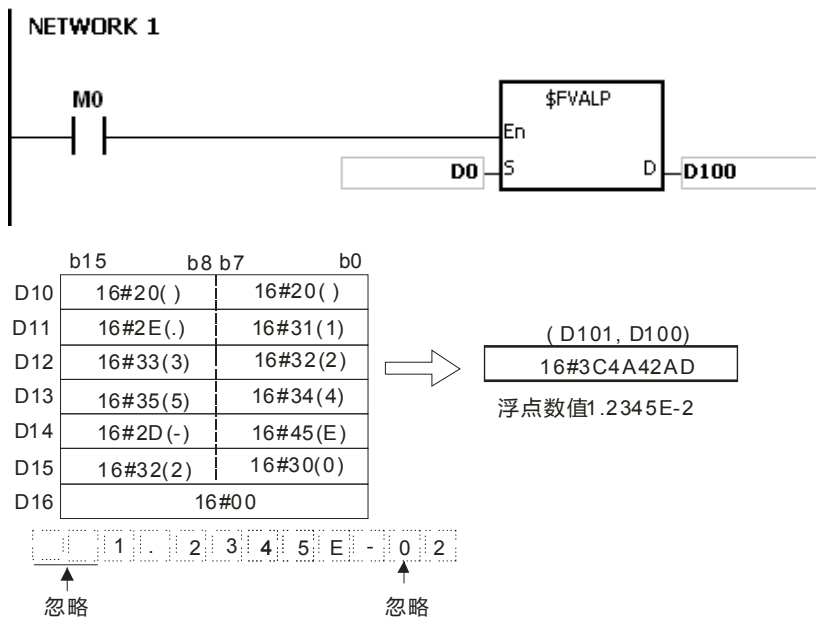


5. **S** 的浮点数字符串长度，最长为 24 个字。

程序范例一：



程序范例二：



补充说明：

1. S 字符串中没有 16#00 当结尾，SM0=ON，SR0 记录错误码 16#200E。
2. S 字符串长度超出范围时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. S 所指定的符号不是 16#20、16#30、16#2B 或 16#2D，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
4. S 字符串中出现“(16#2E)或“(16#2B)或“(16#2D)符号两次或两次以上，指令不执行，SM0=ON，SR0 记录错误码 16#2003，其中 S₁ 字符串中开头的“(16#2D)符号不列入出现次数的计算中。
5. S 字符串中的整数和小数部份不是符号“0”(16#30)~“9”(16#39)，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
6. S 字符串中的指数部份不是符号“E”(16#45)或“(16#2B)或“(16#2D)或“(16#30)~“9”(16#39)，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
7. 当转换结果超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

FB/FC	指令码		操作数			功能		
FC	\$RIGHT	P	S, n, D			从右边获取字符串		

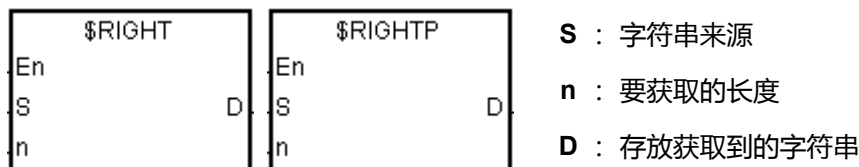
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														●
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

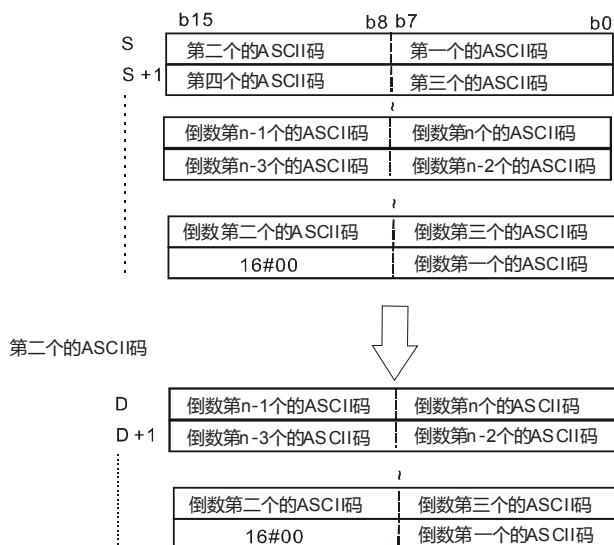
3

图形：

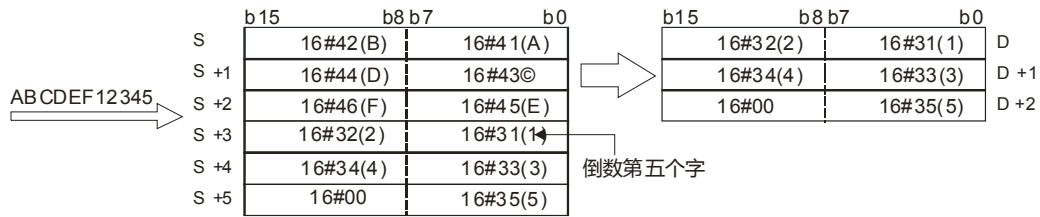


指令说明：

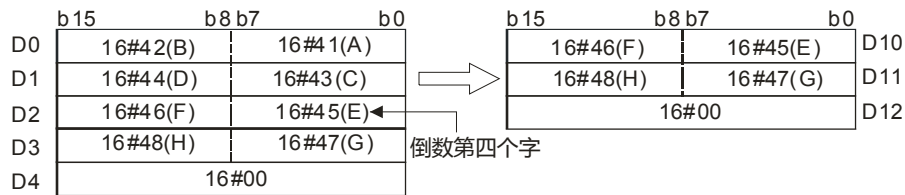
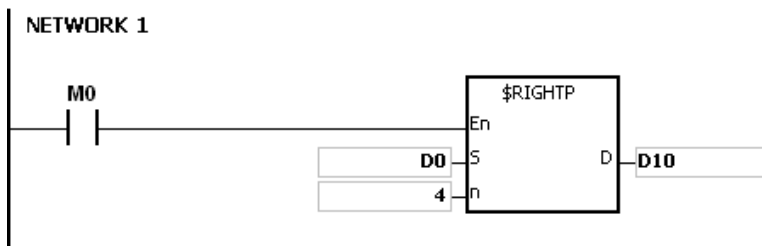
1. 从 S 字符串的最右边开始获取 n 个字，并将获取出来的字符串存放到 D。
2. 当 n=0 时，D 装置会存放 0。



例如，S 的内容为 ABCDEF12345，当 n=5 时从最右边开始获取 5 个字符，转换结果如下图所示。



程序范例：



补充说明：

1. S 字符串中没有 16#00 当结尾，SM0=ON，SR0 记录错误码 16#200E。
2. n<0 或 n>S 的字符串长度时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. D 不足 n 填入字符串长度时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

FB/FC	指令码		操作数			功能	
FC	\$LEFT	P	S, n, D			从左边获取字符串	

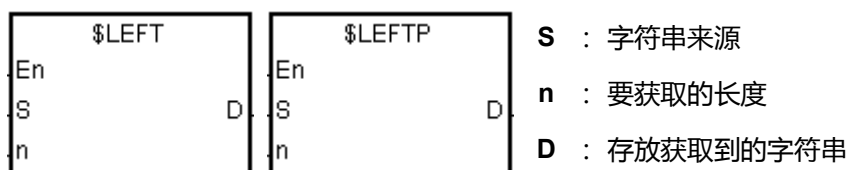
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														●
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

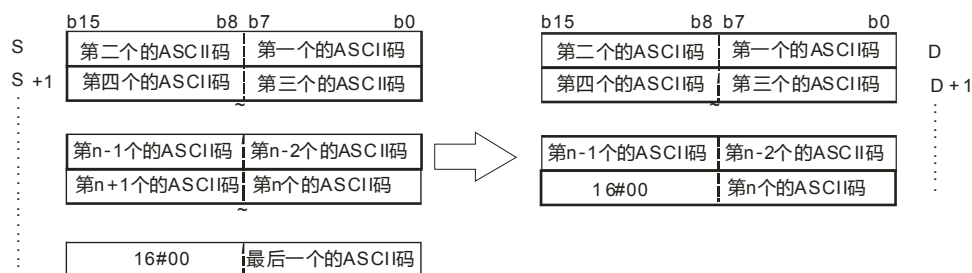
3

图形：

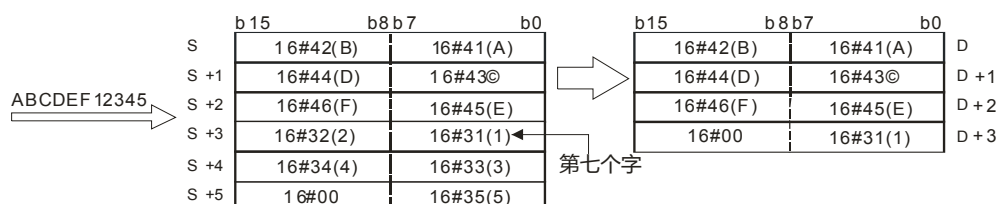


指令说明：

1. 从 S 字符串的最左边开始获取 n 个字，并将获取出来的字符串存放到 D。
2. 当 n=0 时，D 装置会存放 0。

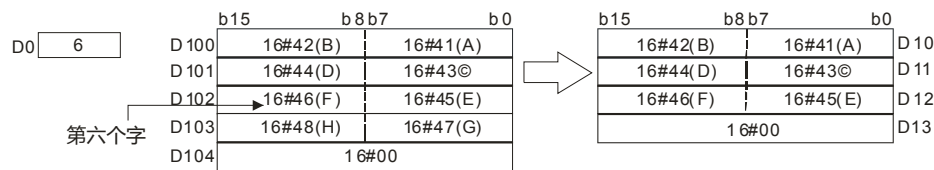
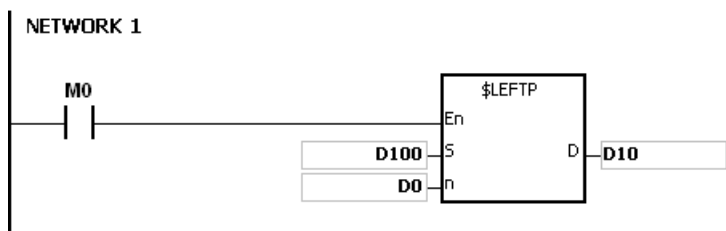


例如，S=ABCDEF12345，当 n=7 时从最左边开始获取 7 个字符，转换结果如下图所示。



程序范例：

M0=ON 时，执行\$LEFT 指令，从 D100 开始的数据开始获取，取 6 个字符后将结果储存至 D10。



补充说明：

3

1. S 字符串中没有 16#00 当结尾，SM0=ON，SR0 记录错误码 16#200E。
2. n<0 或 n>S 的字符串长度时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. D 不足 n 填入字符串长度时，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

FB/FC	指令码		操作数				功能						
FC	\$MIDR	P	S ₁ , S ₂ , D				区段获取字符串						

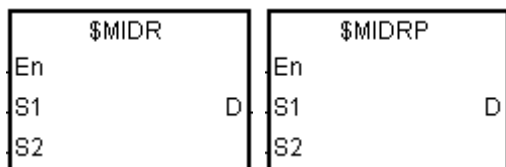
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁														●
S ₂		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●		●		●			○	
S ₂	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

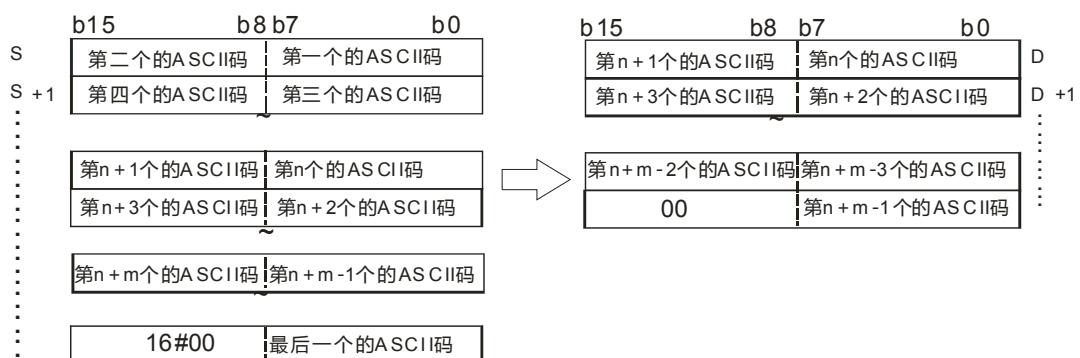
图形：



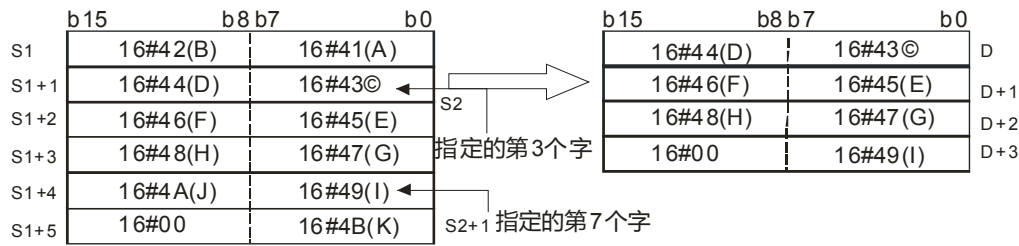
S₁ : 字符串来源
 S₂ : 要获取的区段设定值
 D : 存放获取到的字符串

指令说明：

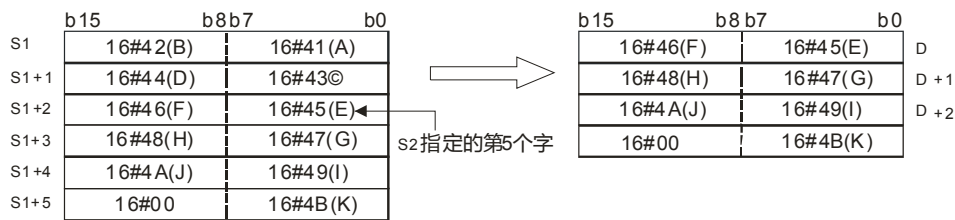
1. 将 S₁ 字符串从 S₂ 所指定的字开始获取 S₂+1 个字，并将获取出来的字符串存放 D，当 S₁=n，S₂+1=m 如图说明。



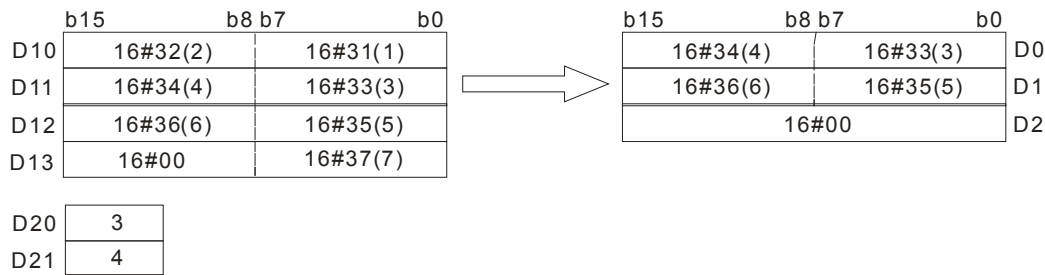
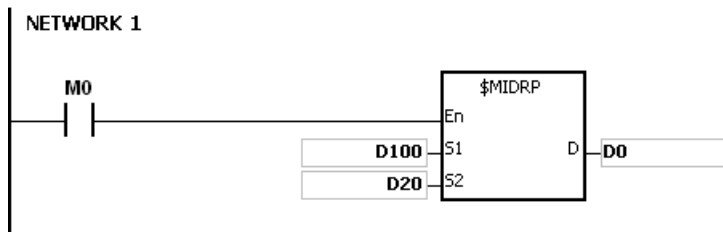
2. 例如 S=ABCDEFGHIJK，当 S₂=3 且 S₂+1=7 时，从最左边第 3 个字符开始获取 7 个字，转换结果如下图所示。



3. 当 $S_{2+1}=0$ 时，指令不动作。
4. 当 $S_{2+1}=-1$ 时，获取 S_2 开始到 S_1 结尾的所有字符串。
5. 例如 $S_1=ABCDEFGHIJK$ ，当 $S_2=5$ 且 $S_{2+1}=-1$ 时，如下图说明。



程序范例：



补充说明：

1. S_1 字符串中没有 16#00 当结尾，SM0=ON，SR0 记录错误码 16#200E。
2. $S_2 \leq 0$ 或 $S_{2+1} < -1$ 时，SM0=ON，SR0 记录错误码 16#2003。
3. S_2 超出 S_1 的字符串长度，SM0=ON，SR0 记录错误码 16#2003。
4. S_{2+1} 所指定的长度超出 S_1 字符串的范围，SM0=ON，SR0 记录错误码 16#2003。
5. 16 位指令 S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。

FB/FC	指令码			操作数				功能						
FC	\$MIDW	P		S ₁ , S ₂ , D				区段字符串取代						

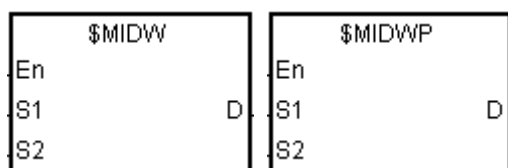
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁														●
S ₂		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●		●		●			○	
S ₂	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

3

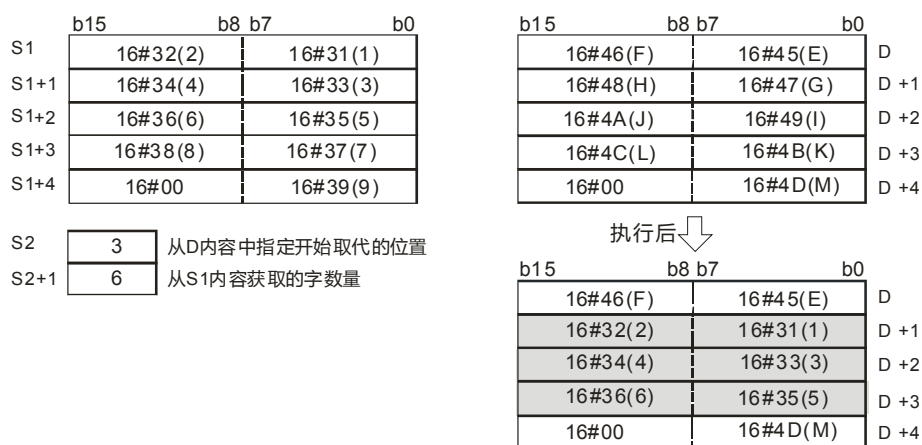
图形：



S₁ : 字符串来源
 S₂ : 要取代的区段设定值
 D : 要被取代的字符串

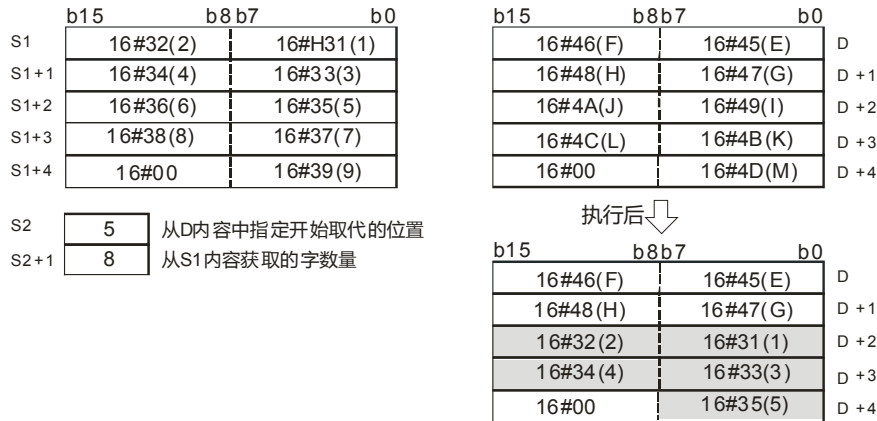
指令说明：

1. S₂ : D 内容中开始被取代的起始位置。S₂+1 : S₁ 内容被获取出的字符数量。
2. 从 S₁ 字符串的第 1 个字开始获取 S₂+1 个字，并将获取出来的字符串，从 D 字符串中 S₂ 所指定的字开始取代原本 D 中的字。

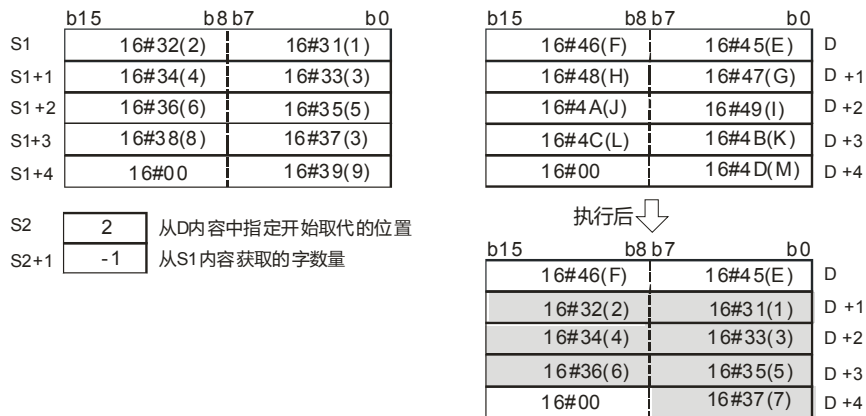


3. 当 S₂+1=0 时，指令不动作。

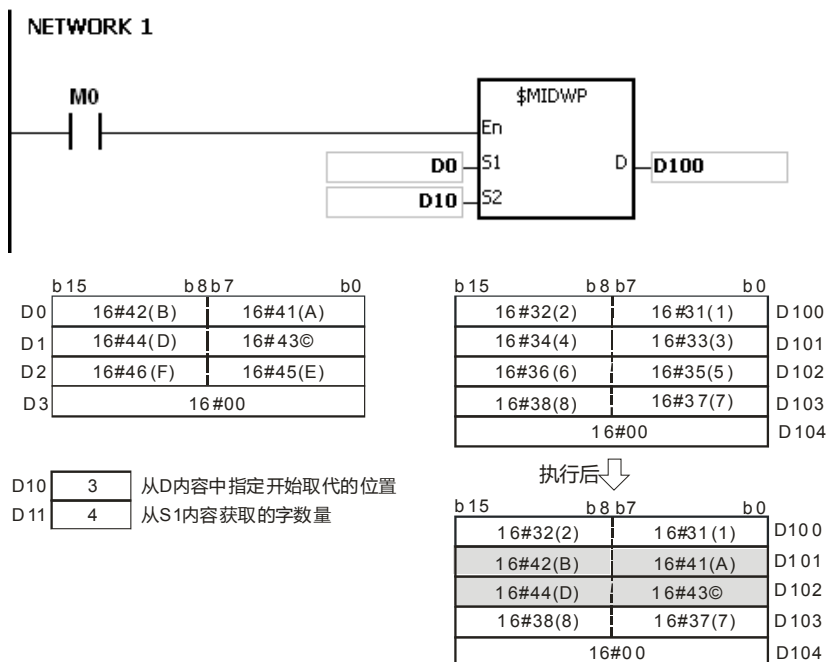
4. 如果 S_2+1 所指定的长度大于 D 字符串的长度，则只取代到 D 字符串的最后一个字。



5. 当 $S_2+1=-1$ 时，获取 S_1 的所有字符串。



程序范例：



补充说明：

1. S_1 字符串中没有 16#00 当结尾, SM0=ON, 记录错误码 SR0=16#200E。
2. D 字符串中没有 16#00 当结尾, SM0=ON, 记录错误码 SR0=16#200E。
3. $S_2 \leq 0$ 或 $S_2 > D$ 的字符串长度, SM0=ON, 记录错误码 SR0=16#2003。
4. $S_2+1 < -1$ 或 S_2+1 所指定的长度超出 S_1 字符串的范围, SM0=ON, 记录错误码 SR0=16#2003。
5. 16 位指令 S_2 操作数, 若使用 ISPSOft 宣告, 则数据类型为 ARRAY [2] of WORD/INT。

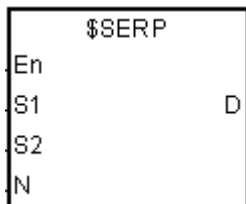
FB/FC	指令码		操作数	功能
FC	\$SER	P	S ₁ , S ₂ , n, D	字符串搜索

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁														●
S ₂														●
n		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●		●		●			○	
S ₂	●	●			●	●		●	●		●		●			○	
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S₁ : 要搜索的字符串

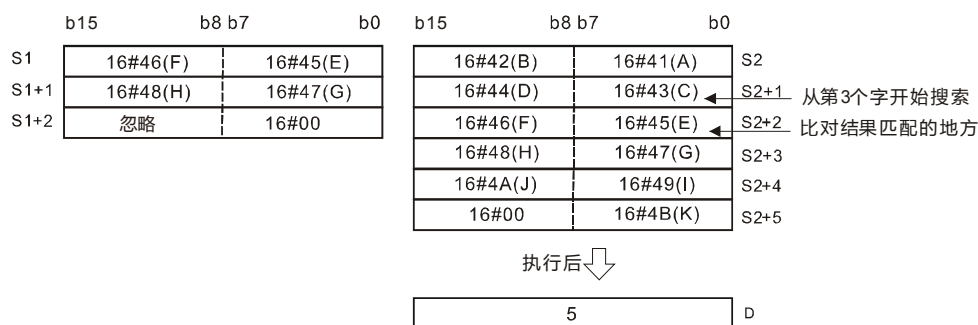
S₂ : 要被搜索的字符串

n : 从 S₂ 的第 n 个字开始搜索

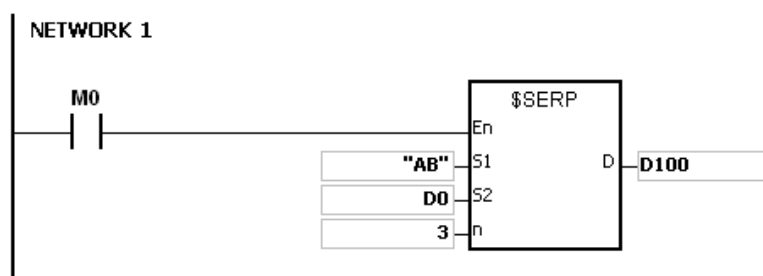
D : 搜索结果

指令说明：

1. 从 S₂ 字符串的第 n 个字开始搜索与 S₁ 相同的字符串，如果搜索到相同的字符串，则将此字符串的起始位置位于 S₂ 字符串的第几个字，存于 D 中，如果没搜索到相同的字符串，则 D=0。
2. 例如，在 S₂="ABCDEFGHJK" 中搜索 S₁="EFGH"，从 S₂ 的第 3 个字符 (n=3) 开始搜索，搜索结果为 S₁ 存在 S₂ 的起始位置为 5 并存在 D=5。



程序范例：



补充说明：

1. S₁字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
2. S₂字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
3. n<=0 或 n> S₂的字符串长度，SM0=ON，记录错误码 SR0=16#2003。

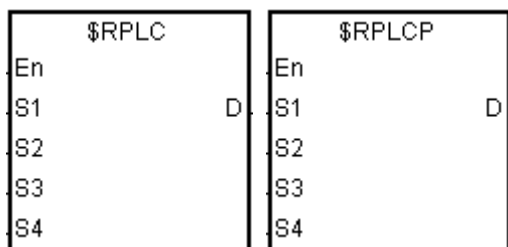
FB/FC	指令码		操作数				功能			
FC	\$RPLC	P	S ₁ , S ₂ , S ₃ , S ₄ , D				字符串取代			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂														●
S ₃		●					●							
S ₄		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●	●	●		●				
S ₂	●	●			●	●		●	●	●	●		●			○	
S ₃	●	●			●	●		●	●	●	●	○	●	○	○		
S ₄	●	●			●	●		●	●	●	●	○	●	○	○		
D	●	●			●	●		●	●	●			●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

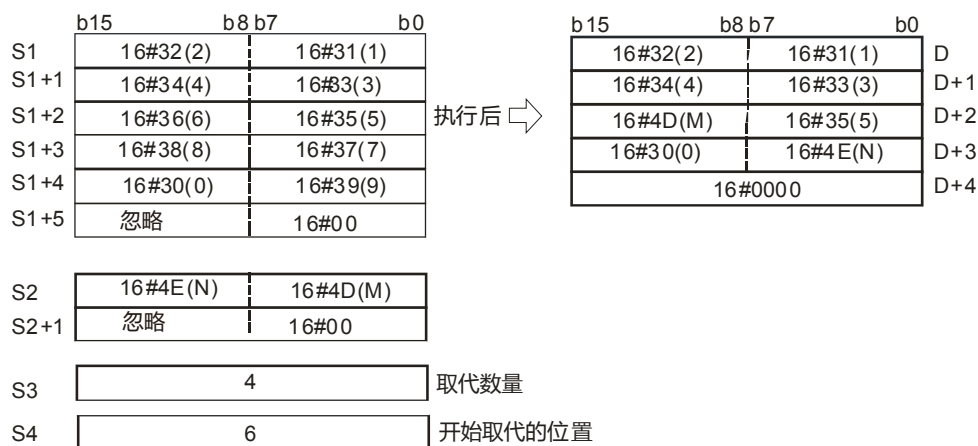
图形：



- S₁ : 要被取代的字符串
- S₂ : 要取代的新字符串
- S₃ : S₁ 要被取代掉的长度
- S₄ : 从 S₁ 的第 S₄ 个字开始取代
- D : 取代后的字符串存放位置

指令说明：

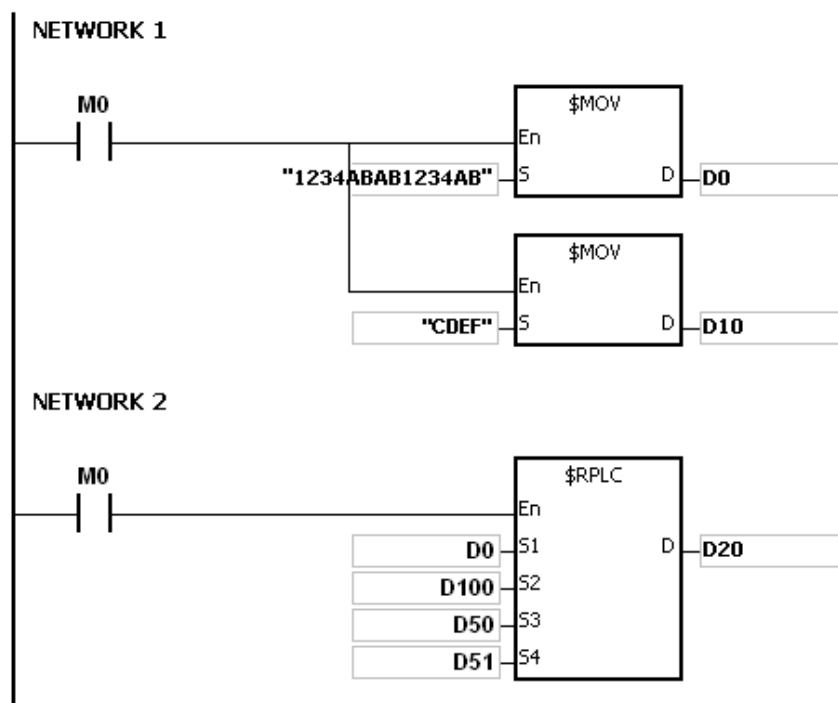
- 从 S₁ 字符串的第 S₄ 个字开始，用 S₂ 的字符串取代 S₁ 中的字符串，取代的长度为 S₃，并将结果存于 D 中，如下图范例说明。
- 将字符串“1234567890”中的第 6 个字开始的 4 个字取代为“MN”，取代后的结果为“12345MN0”。



3. S₂ 字符串如果是 16#00，则此指令功能相当于将字符串删除。
4. 若欲取代的字符串长度 S₃>S₁ 可被取代的长度，则只取代到 S₁ 的最后一个字。
5. S₃=0，指令不动作。

程序范例：

当 M0=ON 时，D0~D7 的内容为“1234ABAB1234AB”，D10~D11 的内容为“CDEF”；当 \$RPLC 指令执行，将用 D10~D11 的内容取代 D0~D7 中由 D51 指定的起始位置取代 D50 所指定的个数 结果储存于 D20~D27 当中。



当 D50 指定值为 3，D51 指定值为 4 时，执行的结果如下图所示。

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)		D21
D2	16#42(B)	16#41(A)		D22
D3	16#42(B)	16#41(A)		D23
D4	16#32(2)	16#31(1)		D24
D5	16#34(4)	16#33(3)		D25
D6	16#42(B)	16#41(A)		D26
D7	忽略	16#00		D27

执行后 →

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#43Ⓞ	16#33(3)		D21
D2	16#45(E)	16#44(D)		D22
D3	16#41(A)	16#46(F)		D23
D4	16#31(1)	16#42(B)		D24
D5	16#33(3)	16#32(2)		D25
D6	16#41(A)	16#34(4)		D26
D7	16#00	16#42(B)		D27

D10	16#44(D)	16#43Ⓞ
D11	16#45(F)	16#45(E)
D12	忽略	16#00

D50 取代数量
 D51 开始取代的位置

当 D50 指定值为 4，D51 指定值为 4 时，执行的结果如下图所示。

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)		D21
D2	16#42(B)	16#41(A)		D22
D3	16#42(B)	16#41(A)		D23
D4	16#32(2)	16#31(1)		D24
D5	16#34(4)	16#33(3)		D25
D6	16#42(B)	16#41(A)		D26
D7	忽略	16#00		D27

执行后 →

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#43Ⓞ	16#33(3)		D21
D2	16#45(E)	16#44(D)		D22
D3	16#42(B)	16#46(F)		D23
D4	16#32(2)	16#31(1)		D24
D5	16#34(4)	16#33(3)		D25
D6	16#42(B)	16#41(A)		D26
D7	16#0000			D27

D10	16#44(D)	16#43Ⓞ
D11	16#45(F)	16#45(E)
D12	忽略	16#00

D50 取代数量
 D51 开始取代的位置

当 D50 指定值为 20，D51 指定值为 4 时，执行的结果如下图所示。

	b 15	b 8 b 7	b 0	
D0	16#32(2)		16#31(1)	D20
D1	16#34(4)		16#33(3)	D21
D2	16#42(B)		16#41(A)	D22
D3	16#42(B)		16#41(A)	D23
D4	16#32(2)		16#31(1)	
D5	16#34(4)		16#33(3)	
D6	16#42(B)		16#41(A)	
D7	忽略		16#00	

执行后 →

	b 15	b 8 b 7	b 0	
D0	16#32(2)		16#31(1)	D20
D1	16#43Ⓞ		16#33(3)	D21
D2	16#45(E)		16#44(D)	D22
D3	16#00		16#46(F)	D23

D10	16#44(D)	16#43Ⓞ
D11	16#45(F)	16#45(E)
D12	忽略	16#00

D50 取代数量

D51 开始取代的位置

当 D50 指定值为 3，D51 指定值为 4，且 D10=16#00 表示未指定取代字符串时，执行的结果如下图所示。没有指定取代字符串，执行后原本的字符串被删除。

	b 15	b 8 b 7	b 0	
D0	16#32(2)		16#31(1)	D20
D1	16#34(4)		16#33(3)	D21
D2	16#42(B)		16#41(A)	D22
D3	16#42(B)		16#41(A)	D23
D4	16#32(2)		16#31(1)	D24
D5	16#34(4)		16#33(3)	D25
D6	16#42(B)		16#41(A)	
D7	16#00			

执行后 →

	b 15	b 8 b 7	b 0	
D0	16#32(2)		16#31(1)	D20
D1	16#41(A)		16#33(3)	D21
D2	16#31(1)		16#42(B)	D22
D3	16#33(3)		16#32(2)	D23
D4	16#41(A)		16#34(4)	D24
D5	16#00		16#42(B)	D25

D10

D50 取代数量

D51 开始取代的位置

补充说明：

1. S_1 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
2. S_2 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
3. $S_3 < 0$ 或 $S_4 \leq 0$ 或 S_4 指定的位置超出 S_1 的长度，SM0=ON，记录错误码 SR0=16#2003。

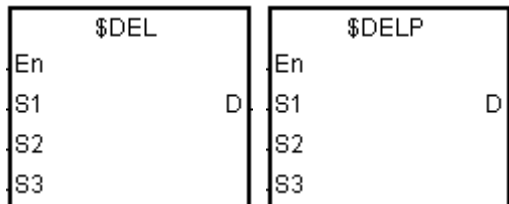
FB/FC	指令码		操作数	功能
FC	\$DEL	P	S ₁ , S ₂ , S ₃ , D	指定字符串删除

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁							●							●
S ₂		●					●							
S ₃		●					●							
D														●

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●		●		●				
S ₂	●	●			●	●		●	●		●	○	●	○	○		
S ₃	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

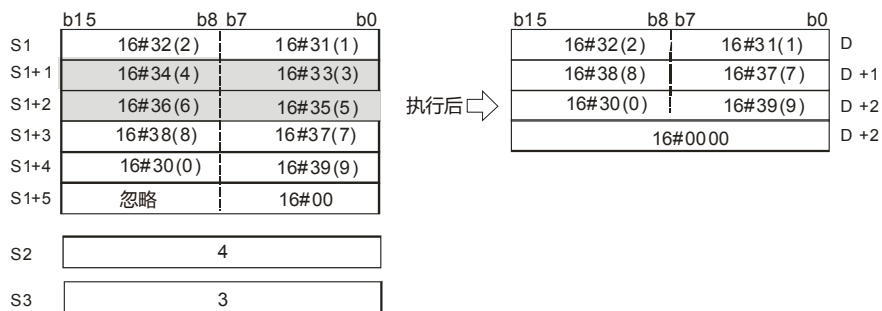
图形：



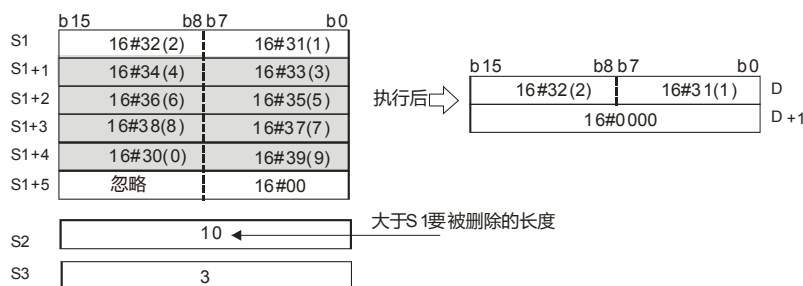
S₁ : 字符串来源
 S₂ : 要删除的个数
 S₃ : 从 S₁ 的第 S₃ 个字开始删除
 D : 删除后的字符串

指令说明：

1. 从 S₁ 字符串的第 S₃ 个字开始，删除 S₂ 个字符串，并将结果存于 D 中。
2. 如下图说明，将 S₁ 的字符串“1234567890”，删除掉第 3 个开始共 4 个字符串，删除结果为“127890”并存在 D 中



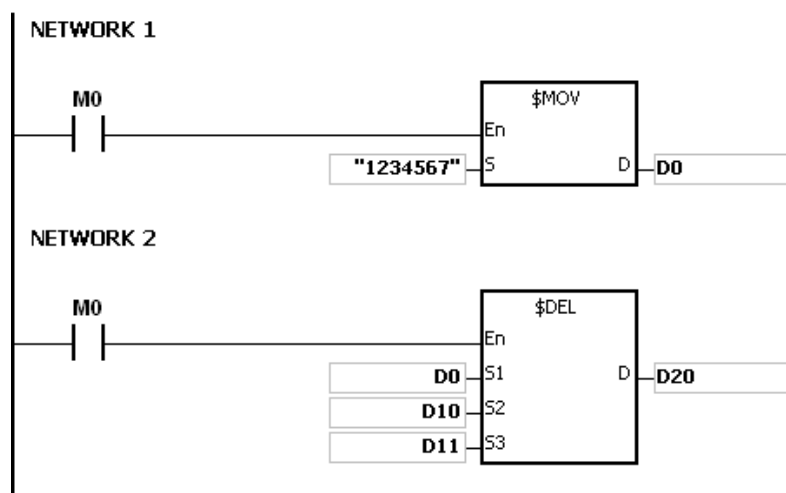
3. 如果欲删除的字符串长度 S₂ > S₁ 要被删除的字符串长度，则只删除到 S₁ 的最后一个字，并将终止符 16#00 存入 D 中。



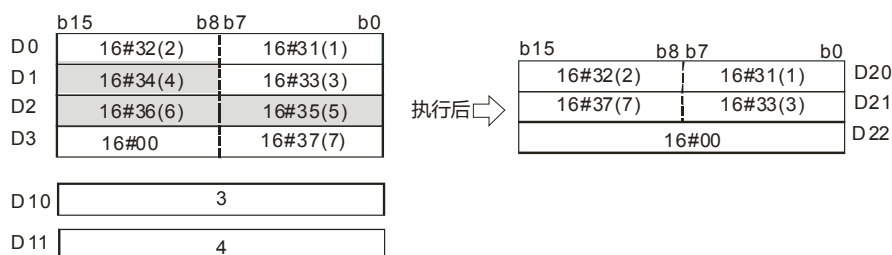
4. $S_2=0$ ，指令不动作。

程序范例：

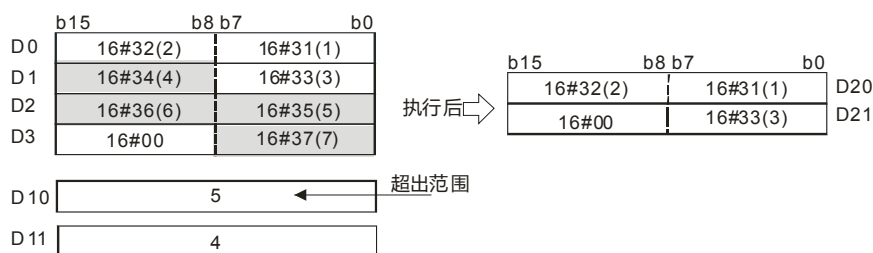
当 M0=ON 时 ,D0~D3 的内容为“1234567” ,当 \$DEL 指令执行 将从 D11 内容的指定位置开始 删除 D0~D3 内的字符串，删除的字符数为 D10 内容所指定的数值，结果储存于 D20~D22 当中。



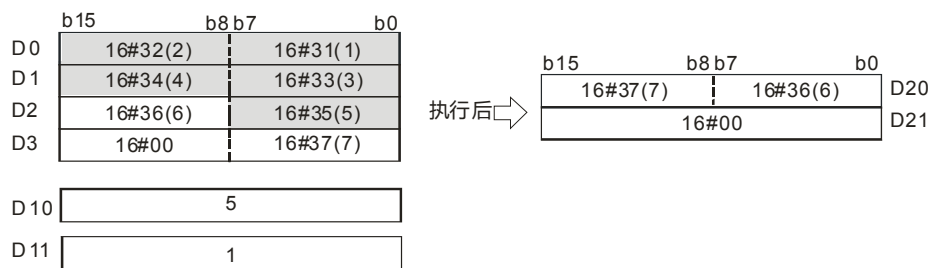
当 D10 指定删除数量为 3，D11 指定从第 4 个字符串开始删除时，执行的结果如下图所示。



当 D10 指定删除数量为 5，D11 指定从第 4 个字符串开始删除时，执行的结果如下图所示。由于指定的删除数量超出范围，故删除到最后一个字后结束。



当 D10 指定删除数量为 5，D11 指定从第 1 个字符串开始删除时，执行的结果如下图所示。



补充说明：

1. S_1 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
2. $S_2 < 0$ 或 $S_3 < 0$ 或 S_3 指定的位置超出 S_1 的长度，SM0=ON，记录错误码 SR0=16#2003。

FB/FC	指令码			操作数							功能						
FC		\$CLR	P	S							字符串清除						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S														●			
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●				●				
脉冲执行型				16 位指令				32 位指令									
AH 运动控制主机				AH 运动控制主机				-									

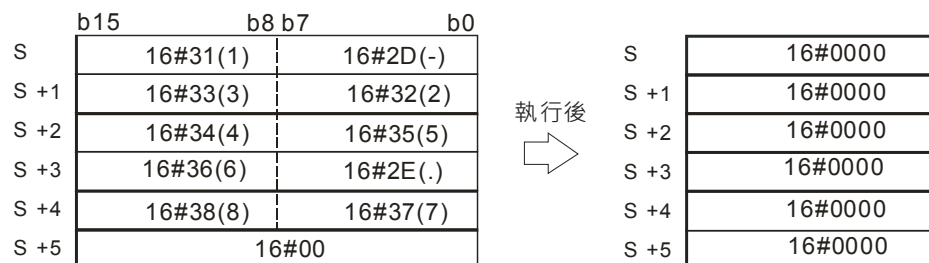
图形：



S：要清除的字符串来源

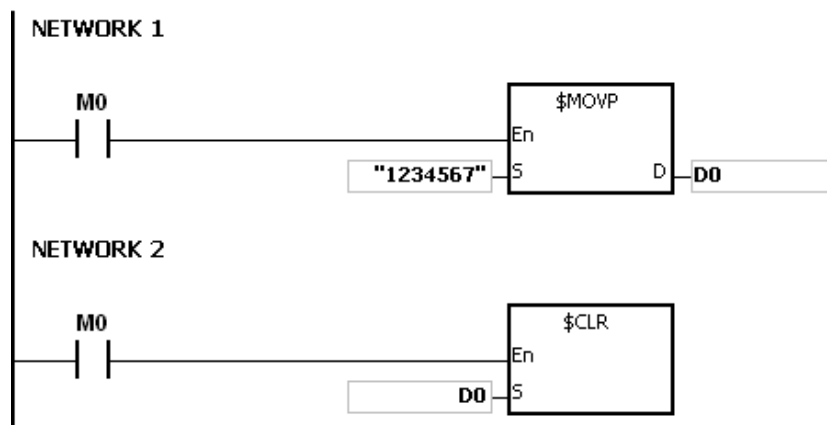
指令说明：

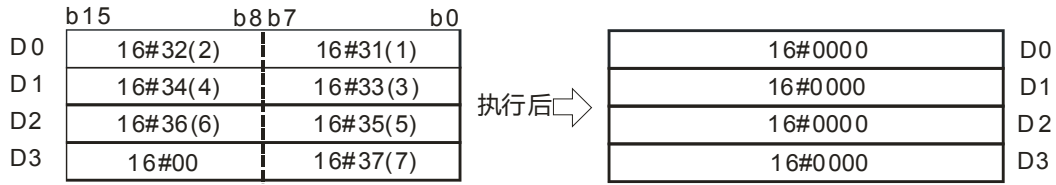
将 S 中的字符串全部清除为 0。



程序范例：

将 D0 的字符串内容清除，如下图说明。





补充说明：

S 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。

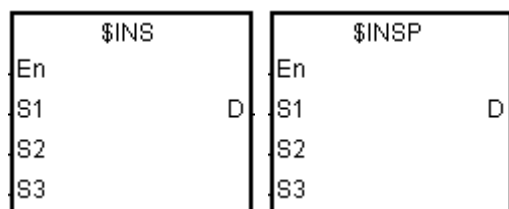
FB/FC	指令码		操作数				功能				
FC		\$INS	P	S₁, S₂, S₃, D				字符串插入			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁		●					●							
S ₂														●
S ₃		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●		●				
S ₂	●	●			●	●		●	●		●		●			○	
S ₃	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S₁ : 来源字符串

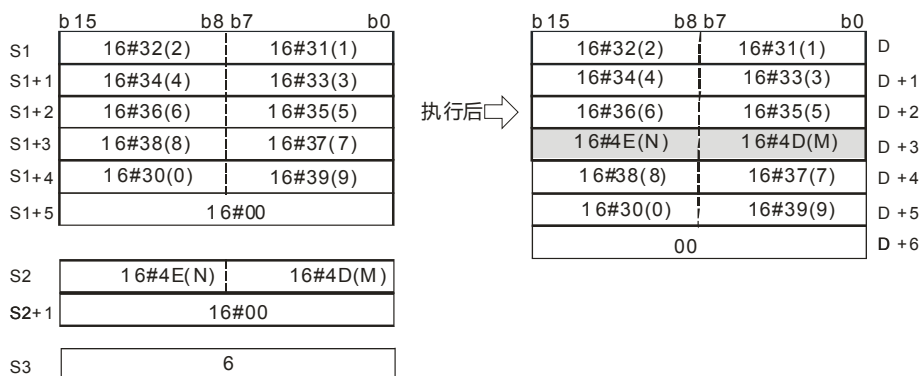
S₂ : 要插入的字符串

S₃ : 从 S₁ 的第 S₃ 个字之后开始插入

D : 存放插入之后的字符串

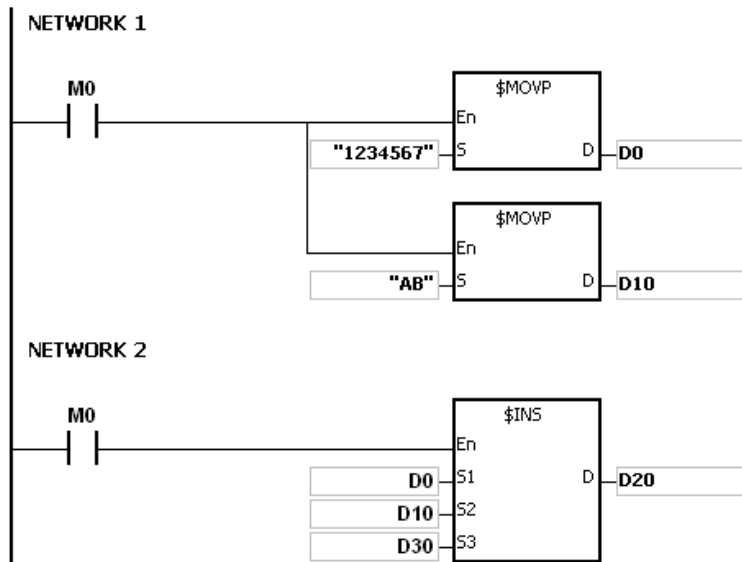
指令说明：

1. 从 S₁ 字符串的第 S₃ 个字之后开始，插入 S₂ 的字符串，并将结果存于 D 中。
2. 如果 S₁ 或 S₂ 有任何一个字符串为空字符串，则另一个不是空字符串的值会被存于 D 中。
3. 如果 S₁ 和 S₂ 都为空字符串，则 D 为 16#0000。



程序范例：

当 M0=ON 时，D0~D3 的内容为“1234567”，D10 的内容为“AB”；当 \$INS 指令执行，将从 D30 内容所指定的位置之后开始，插入 D10 的内容，结果储存于 D20~D24 当中。



D30 指定从第 1 个字符后开始插入，执行的结果如下图所示。

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		
D1	16#34(4)	16#33(3)		
D2	16#36(6)	16#35(5)		
D3	16#00	16#37(7)		
D10	16#42(B)	16#41(A)		
D11	忽略	16#00		
D30	1			

执行后 →

	b15	b8 b7	b0	
D20	16#41(A)	16#31(1)		
D21	16#32(2)	16#42(B)		
D22	16#34(4)	16#33(3)		
D23	16#36(6)	16#35(5)		
D24	16#00	16#37(7)		

D30 指定从第 0 个字符后开始插入，执行的结果如下图所示。

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		
D1	16#34(4)	16#33(3)		
D2	16#36(6)	16#35(5)		
D3	16#00	16#37(7)		
D10	16#42(B)	16#41(A)		
D11	忽略	16#00		
W0	0			

执行后 →

	b15	b8 b7	b0	
D20	16#42(B)	16#41(A)		
D21	16#32(2)	16#31(1)		
D22	16#34(4)	16#33(3)		
D23	16#36(6)	16#35(5)		
D24	16#00	16#37(7)		

补充说明：

1. S₁ 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
2. S₂ 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
3. S₃<0 或 S₃ 指定的位置超出 S₁ 的长度，SM0=ON，记录错误码 SR0=16#2003。

FB/FC	指令码		操作数				功能					
FC		FMOD	P	S_1, S_2, D				浮点数转 BCD 浮点数				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1			●					●						
S_2		●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●		●		●				○
S_2	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	32 位指令	64 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

1. 将浮点数 S_1 中的小数点右移 S_2 的位数后，再转成 BCD 浮点数格式，并将结果存于 D 中。



BCD 浮点数格式说明如下：

S_2 ：小数点指定位移位数， S_2 的范围为 0~7。

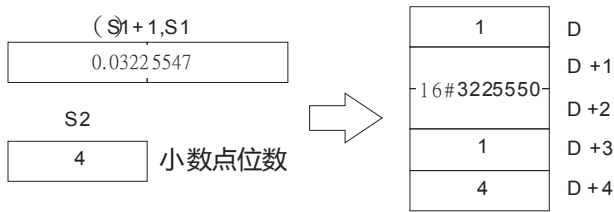
D：纪录符号，若 S_1 是正数，则 $D=0$ ；若 S_1 是负数，则 $D=1$ 。

($D+2, D+1$)：纪录浮点数转换后的 7 位数 BCD 值。

$D+3$ ：纪录浮点数转成 BCD 浮点数格式之后的指数符号。若指数符号是正数，则 $D+3=0$ ；若指数符号是负数，则 $D+3=1$ 。

$D+4$ ：纪录 BCD 浮点数格式的指数值。

例如 $S_1 = -0.03225547$ ，指定小数点右移位数为 4，经转换后的结果如下图所示。



S_2 指定小数点右移位数为 4，所以 S_1 的小数点往右移 4 位后为 -322.5547。

将 -322.5547 视为 -3225547E-4，转换为 BCD 浮点数格式后为

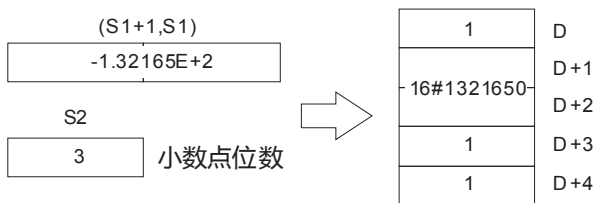
D=1 (因为 S_1 是负数)。

(D+2, D+1) = 16#3225550 (转成 7 位数的 BCD，第 7 位数被四舍五入)。

D+3=1 (因为指数的符号是负数)。

D+4=4 (因为指数的数值为 4)。

例如 $S_1 = -1.32165E+2$ ，指定小数点右移位数为 3，经转换后的结果如下图所示。



原本的 S_1 浮点数为 $-1.32165E+2 = -132.165$ ，

S_2 指定小数点右移 3 位，所以 S_1 的小数点往右移 3 位后为 -132165。

将 -132165 视为 -1321650E-1 (因为需固定 7 位数 BCD)，转换为 BCD 浮点数格式后为

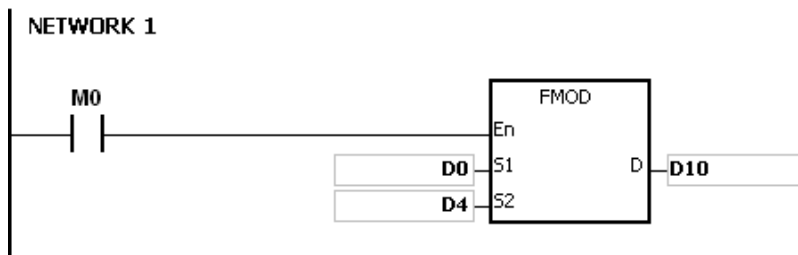
D = 1 (因为 S_1 是负数)。

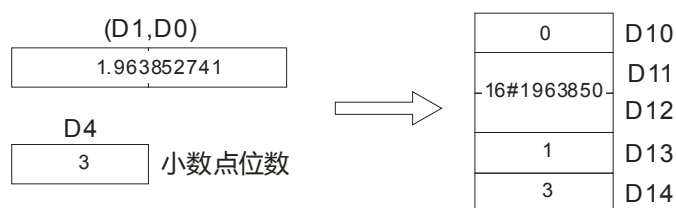
(D+2, D+1) = 16#1321650 (转成 7 位数的 BCD，第 7 位数被四舍五入)。

D+3=1 (因为指数的符号是负数)。

D+4=1 (因为指数的数值为 1)。

程序范例：





(D1、 D0) = 1.963852741 , D4 = 3 , 所以右移 3 个位数之后 = 1963.852741。

D10 = 0 , 因为来源 (D1、 D0) 值为正数。

将 1963.852741 视为 1963852E-3 (因为需固定 7 位数 BCD) , 转换为 BCD 浮点数格式后为

(D12、 D11) = 16#1963850 (取七位数 , 第七位四舍五入)。

D13 = 1 (指数符号为负数)。

D14 = 3 (指数值为 3)。

补充说明：

1. 当 S_1 超出浮点数可以表示的范围时 , 指令不执行 , SM0=ON , 错误码 SR0=16#2013。
2. S_2 的内容值超出范围时 , 指令不执行 , SM0=ON , 错误码 SR0=16#2003。
3. 32 位指令 D 操作数 , 若使用 ISPSOft 宣告 , 则数据类型为 ARRAY [5] of WORD/INT。

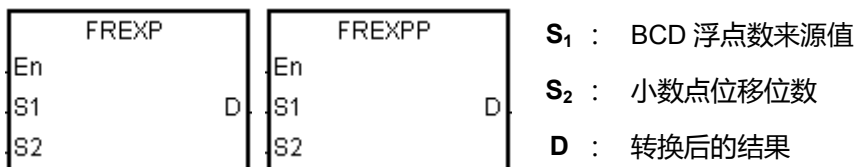
FB/FC	指令码		操作数				功能						
FC	FREXP	P	S_1, S_2, D				BCD 浮点数转浮点数						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●					●							
S_2		●					●							
D			●					●						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●		●	●		●		●				
S_2	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●		●				

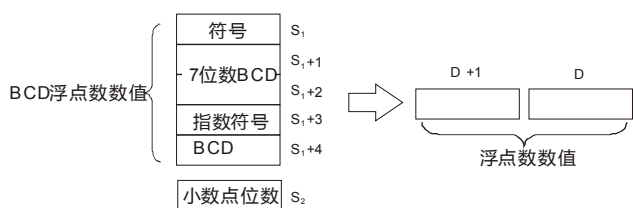
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

将 BCD 浮点数格式数值 S_1 ，依照 S_2 设定的小数点左移的位数，转成浮点数，并将结果存于 D 中。

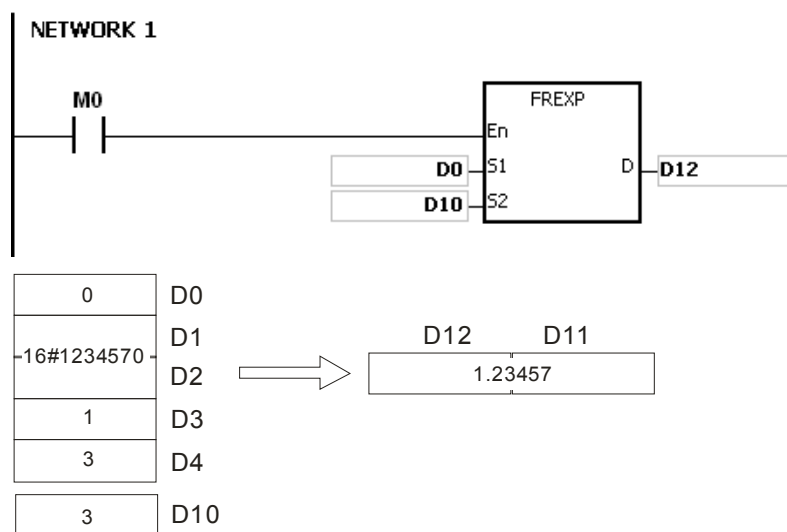


BCD 浮点数格式说明如下：

- S_1 纪录浮点数的符号，若符号为 0 表浮点数为正数，若符号为 1 表浮点数为负数。
- (S_1+2, S_1+1) 纪录 BCD 浮点数表示值的 7 位数整数值。
- S_1+3 纪录指数部分的符号，若指数符号为 0 表示指数值为正数，若指数符号为 1 表示指数值为负数。
- S_1+4 ，纪录 BCD 的指数值，BCD 指数的范围为 0~38。
- S_2 ：纪录小数点指定左移位数， S_2 的范围为 0~7。（来源转成浮点数之后，小数点再左移 S_2 个数）

程序范例：

启动条件接点 M0，将 BCD 浮点数转成浮点数，说明如下：



D0=0 表示浮点数为正数.

(D2 , D1) =16#1234570 表示浮点数的整数部分

D3=1 表示指数符号为负数

D4=3 表示指数值为 3

将 D0~D4 转换成浮点数后为 1234570E-3=1234.57

D10=3 表示小数点需左移 3 位数，转移后的浮点数为 1.23457 并纪录在 (D12 , D11)

补充说明：

1. 当 S_1 符号所指定的值不是 0 也不是 1，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. S_1+1 ， S_1+2 所指定的 BCD 位数>7，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. S_1+1 ， S_1+2 的数据内容并非为 BCD 值（以 Hex 表示有任一位数不在 0~9 的范围内），则将会产生运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#200D。
4. 当 S_1+3 符号所指定的值不是 0 也不是 1，指令不执行，SM0=ON，错误码 SR0=16#2003。
5. S_1+4 ，BCD 指数内容值<0 或>38，指令不执行，SM0=ON，错误码 SR0=16#2003。
6. S_2 的内容值<0 或>7，指令不执行，SM0=ON，错误码 SR0=16#2003。
7. 32 位指令 S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [5] of WORD/INT。

3.24 以太网控制指令

FB/F C	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>EMDRW</u>	–	✓	MODBUS TCP 数据读写	11
FC	–	<u>DINTOA</u>	✓	IP→字符串变换	5
FC	–	<u>DIATON</u>	✓	字符串→IP 变换	5-11
FC	<u>SOPEN</u>		✓	开启 SOCKET	7
FC	<u>SSEND</u>	–	✓	通过已开启的 SOCKET 传送数据	5
FC	<u>SRCVD</u>	–	✓	通过已开启的 SOCKET 接收数据	5
FC	<u>SCLOSE</u>	–	✓	关闭 SOCKET	5

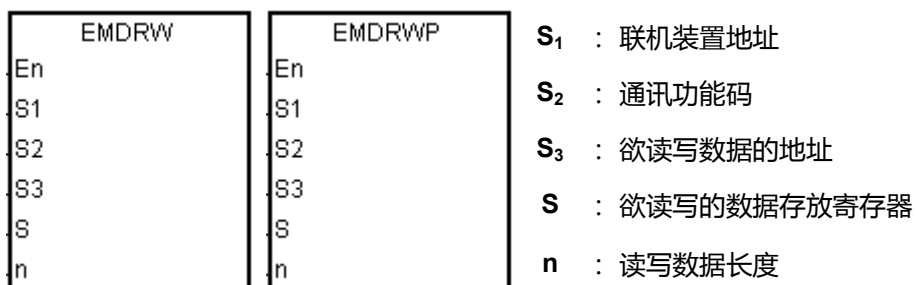
FB/FC	指令码		操作数				功能				
FC	EMDRW	P	S_1, S_2, S_3, S, n				MODBUS TCP 数据读写				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●					●							
S_2, S_3		●					●							
S	●	●					●							
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●		●	●			○	●				
S_2, S_3	●	●			●	●		●	●			○	●	○	○		
S	●	●	●		●	●		●	●			○	●				
D	●	●			●	●		●	●			○	●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



指令说明：

- 使用本指令前，需用 ISPSOft 完成以下设定。
 - 在 PLC 参数设定 -> 基本设定中设定本机的 IP 地址、网络屏蔽。
- S_1 设定：

地址	描述	设定范围
S_1	通讯站号	0-255
S_1+1	远程 IP High Word	范例：远程 IP 地址=172.16.144.230
S_1+2	远程 IP Low Word	$S_1+1=16\#AC10$ $S_1+2=16\#90E6$
S_1+3	指令完成后是否关闭联机	0：关闭联机，1：保持联机（依 PLC 参数设定中 TCP 联机保持时间决定联机何时关闭）

- S_2 ：通讯功能码（Function Code）。

例如：

- AH 运动控制主机的读取多笔位装置 (Bit) (非 Discrete Input 装置) 命令为 1 (16#01) ，
- AH 运动控制主机的读取多笔位装置 (Bit) (仅 Discrete Input 装置) 命令为 2 (16#02) ，
- AH 运动控制主机的读取多笔字符装置 (Word) (非 Input Register 装置) 命令为 3 (16#03) ，
- AH 运动控制主机的读取多笔字符装置 (Word) (仅 Input Register 装置) 命令为 4 (16#04) ，
- AH 运动控制主机的单笔位装置 (Bit) 状态写入命令为 5 (16#05) ，
- AH 运动控制主机的单笔字符装置 (Word) 数据写入命令为 6 (16#06) ，
- AH 运动控制主机的多笔位装置 (Bit) 状态写入命令为 15 (16#0F) ，
- AH 运动控制主机的多笔字符装置 (Word) 数据写入命令为 16 (16#10) 。

目前仅支持上述功能码，其余功能码将无法执行。请参考下列程序范例。

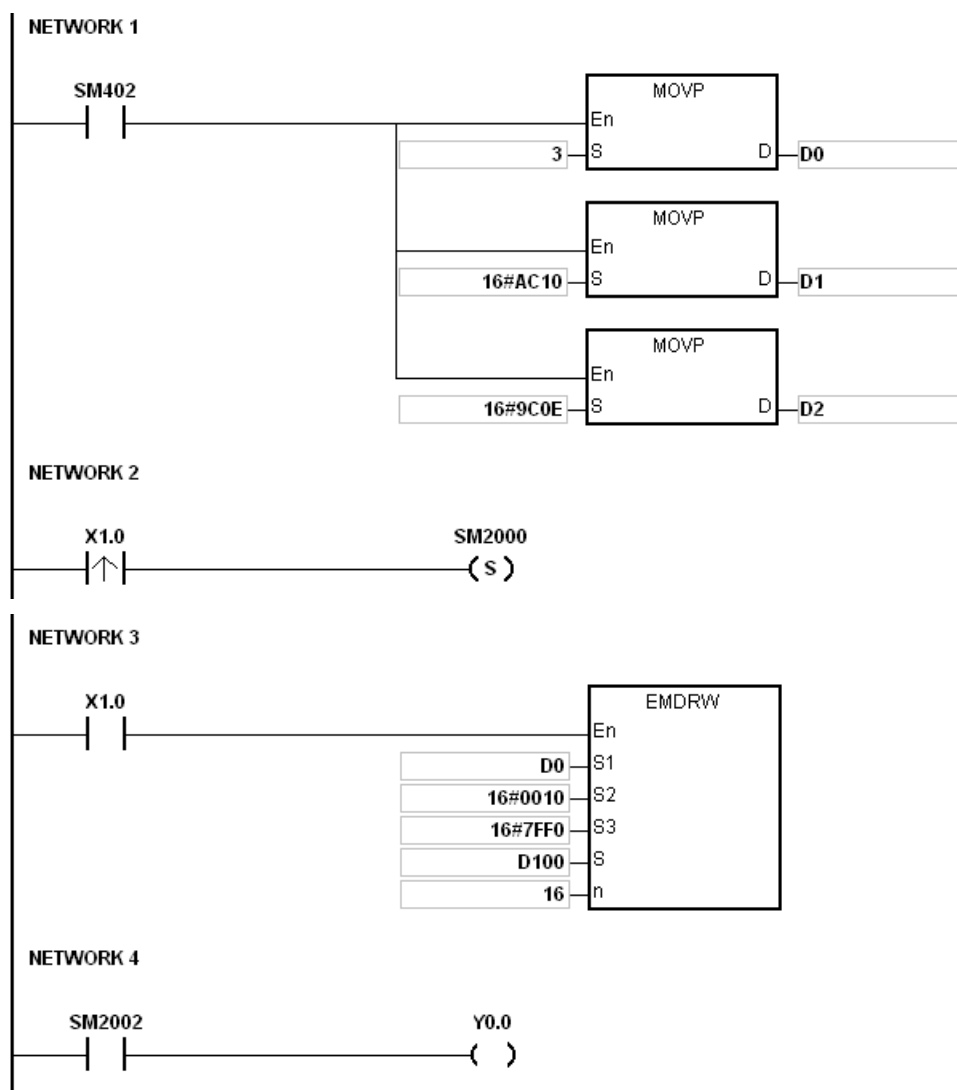
- 3**
4. **S₃**：欲读写数据的地址（装置地址）。联机装置的内部装置地址。
 5. **S**：欲读写之数据（来源或目的）。由用户设定寄存器，将欲写入数据长度的数据事先存入寄存器内。或数据读取后存放之寄存器。
 6. **n**：读写数据长度（数据长度），指定范围 1~240（Byte）。设定的数据量不可以大于 240 字符。Coil 相关的通讯命令，其数据单位为 Bit，n 的范围为 1~1920。Register 相关的通讯命令，其数据单位为 Word，则 n 的范围为 1~120。

标志 EMDRW	要求标志	等待中标志	已接收标志	错误标志	逾时标志	联机已关闭标志
1	SM2000	SM2001	SM2002	SM2003	SM2004	SM2005
2	SM2006	SM2007	SM2008	SM2009	SM2010	SM2011
3	SM2012	SM2013	SM2014	SM2015	SM2016	SM2017
4	SM2018	SM2019	SM2020	SM2021	SM2022	SM2023
5	SM2024	SM2025	SM2026	SM2027	SM2028	SM2029
6	SM2030	SM2031	SM2032	SM2033	SM2034	SM2035
7	SM2036	SM2037	SM2038	SM2039	SM2040	SM2041
8	SM2042	SM2043	SM2044	SM2045	SM2046	SM2047

7. 本指令于程序中使用次数并无限制，但是同时间仅有 8 个指令被执行。
8. 若指令执行时，同时有不只一个送信标志为 ON，则以标志数字最小者优先执行。
9. 本指令一般都是使用脉冲执行型指令（EMDRWP）。

程序范例：

设定远程站号为 3。



补充说明：

1. S_1 、 S_2 、 S 内容值超出范围时，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
2. 若 n 操作数的内容值不在范围内，指令不执行， $SM0=ON$ ， $SR0$ 记录错误码 16#200B。
3. S_1+3 装置超出范围时，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
4. S 装置不足以写入或读取 n 所指定的长度时，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
5. 当 S_2 通讯功能码指定为 Bit 读取时 S 必需指定为 Bit 装置，否则视为运算错误，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
6. 当 S_2 通讯功能码指定为 Word 读写时 S 必需指定为 Word 装置，否则视为运算错误，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
7. 当通讯命令为 0x05 与 0x06 时， n 无作用，指令只会写入 1 个 Bit 或 Word。

8. 当指令执行时，若某要求标志为 ON，但其对应的联机已关闭标志不为 ON，则系统会自动寻找是否有其它要求标志为 ON 且其对应的联机已关闭标志为 ON 来执行指令，若无则此次指令不执行。
9. 联机时，当远程机器的 IP，最高位为 0、127 或是大于 225 的值，则会发生错误，对应的错误标志为 ON，错误码 16#6403。
10. 联机时，当与远程机器中断联机时，错误码为 16#6401；或者联机逾时时，错误码为 16#6402。
11. 传送指令时，当远程机器中断联机时，错误码为 16#6401；或者 Ack 逾时时，错误码为 16#6402。
12. 接收命令时，当通讯功能码错误时，错误码为 16#6404；当长度错误时，错误码为 16#6405 或者回复逾时时，错误码为 16#6402。
13. 关闭联机时，当与远程机器中断联机时，错误码为 16#6401；或者联机逾时时，错误码为 16#6402。
14. S_1 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of WORD/INT。
15. 当执行此指令时，若网络线未连接，则 SM1100 为 ON，错误码 16#600D。

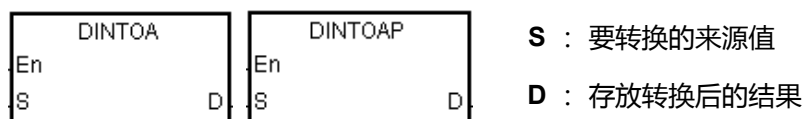
FB/FC	指令码			操作数	功能
FC	D*	INTOA	P	S, D	IP→字符串变换

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S			●*					●*						
D		●*					●*							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●	○	○		
D	●	●			●	●		●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	-	AH 运动控制主机

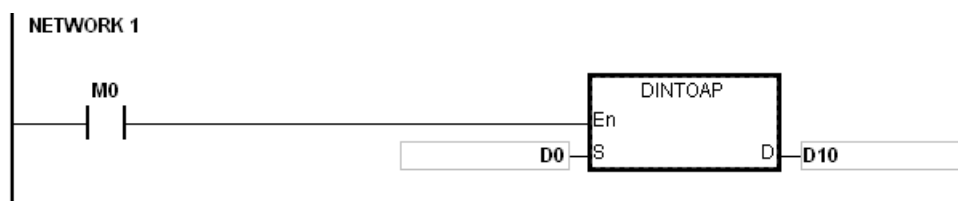
图形：



指令说明：

1. 将 S 的内容值转换成字符串类型的 IP 地址，并且将结果存到 D。
2. D 连续占用 8 个装置。

程序范例一：



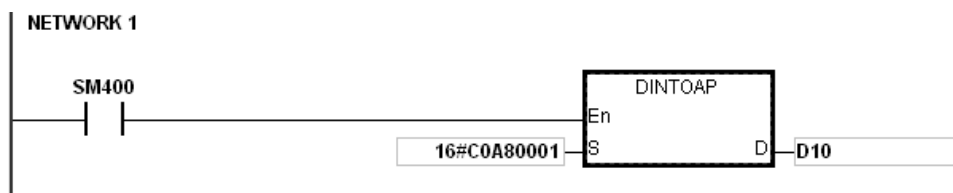
D0	16#C0(192)	16#A8(168)
D1	16#00(0)	16#01(1)

执行后 →

D10	"g"	"1"
D11	"."	"2"
D12	"6"	"1"
D13	"."	"8"
D14	"0"	"0"
D15	"."	"0"
D16	"0"	"0"
D17	16#00	"1"

"192.168.0.1"

程序范例二：



C0A8001(Hex) 执行后 →

D10	"9"	"1"	"192.168.0.1"
D11	."	"2"	
D12	"6"	"1"	
D13	."	"8"	
D14	"0"	"0"	
D15	."	"0"	
D16	"0"	"0"	
D17	16#00	"1"	

3

补充说明：

D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of WORD/INT。

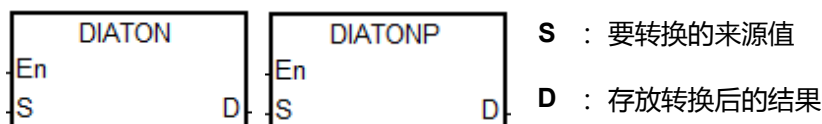
FB/FC	指令码			操作数	功能
FC	D*	IATON	P	S, D	字符串→IP 变换

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														●
D			●*					●*						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●			○	
D	●	●			●	●		●	●				●				

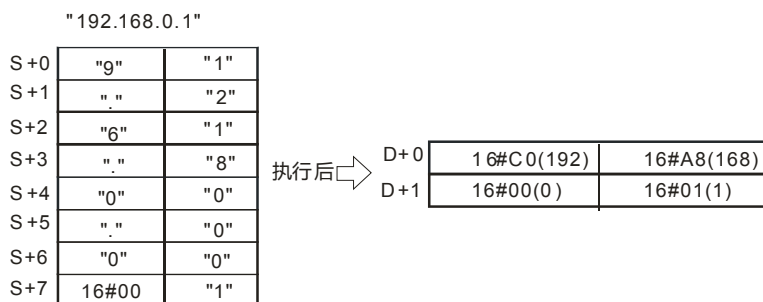
脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	-	AH 运动控制主机

图形：



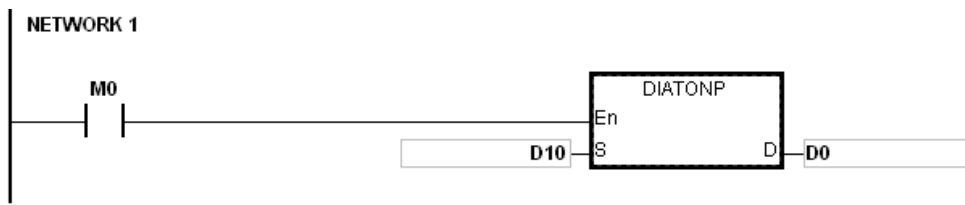
指令说明：

1. 将 S 的字符串 IP 地址转换成整数类型，并且将结果存到 D。
2. S 连续占用 8 个装置。



3. S 的字符串 IP 地址来源，每个区段数值固定为 3 个字符，中间以一个“.”16#2E 隔开。
4. S 的字符串 IP 地址来源，每个区段数值转换出来的结果不能>255。
5. 如果 S 是由用户直接输入字符串，则每个区段不必固定输入 3 个字符，例如：“192.168.0.1”，只要直接输入即可，不必输入“192.168.000.001”。

程序范例一：



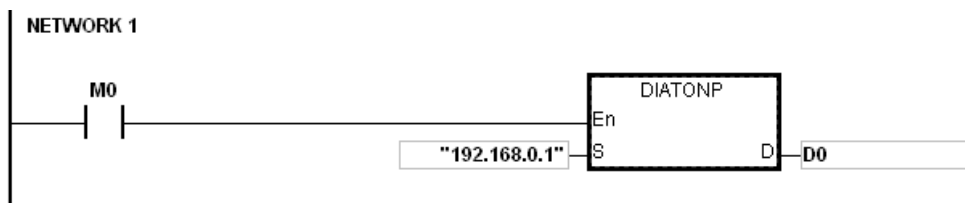
"192.168.0.1"

D10	"9"	"1"
D11	"."	"2"
D12	"6"	"1"
D13	"."	"8"
D14	"0"	"0"
D15	"."	"0"
D16	"0"	"0"
D17	16#00	"1"

执行后 →

D0	16#C0(192)	16#A8(168)
D1	16#00(0)	16#01(1)

程序范例二：



"192.168.0.1" 执行后 →

D0	16#C0(192)	16#A8(168)
D1	16#00(0)	16#01(1)

补充说明：

1. S 字符串中没有 16#00 当结尾，SM0=ON，SR0 错误码为 16#200E。
2. S 字符串除了小数点，其余要转成 BIN 值的部份不在 16#30~16#39 范围内，指令不执行，SM0=ON，SR0 错误码为 16#2003。
3. S 字符串中小数点的地址“.”符号不为 16#2E 时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
4. S 字符串中小数点“.”符号的个数不等于 3 个时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
5. S 的字符串来源，若有任一个区段的数值转换出来的结果>255，指令不执行，SM0=ON，SR0 错误码为 16#2003。
6. S 的字符串 IP 地址来源，每个区段数值固定 1~3 个字符，若超过则指令不执行，SM0=ON，SR0 错误码为 16#2003。
7. S 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of WORD/INT。

FB/FC	脚本			操作数			功能					
FC		SOPEN	P	$S_1 \cdot S_2 \cdot S_3$			开启 Socket					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●												
S_2		●												
S_3		●												

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●	○	○		
S_3	●	●			●	●		●	●			○	●	○	○		

脉波执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

符号：

SOPEN		SOPENP	
En		En	
S1		S1	
S2		S2	
S3		S3	

S_1 : Socket 模式. Word

S_2 : Socket 编号 Word

S_3 : 开启模式 Word

指令说明：

- S_1 为 Socket 模式，欲开启 TCP Socket 时， S_1 为 1 时，开启 UDP Socket 时， S_1 为 0， S_2 为 Socket 编号，对应以太网网络中 Socket 设定；欲开启 TCP 联机时， S_3 为 1 时，表示 AH500 为从端，主动送出联机要求给主端，为 S_3 为 0 时，表示 AH500 为主端，等待来自从端的联机要求，欲开启 UDP 联机时， S_3 可设为 0 或 1，不影响使用结果。
- $S_1=0\sim 1$ ， $S_2=1\sim 8$ ， $S_3=0\sim 1$ 。
- 使用本指令前，需用 ISPSOft - > HWCONFIG 完成以下设定，或者用 MOV 指令将 Socket 数据搬入对应的 SR 缓存器中。
 - 在 PLC 参数设定 - > 基本设定 - > 设定本机的 IP 地址、网络屏蔽。
 - 在 PLC 参数设定 - > 进阶设定 - > 通讯接口 - > 启动通讯接口功能。
 - 在 PLC 参数设定 - > 进阶设定 - > 通讯接口 - > TCP/UDP 通讯接口联机设定
欲使用的 Socket
- TCP Socket 设定对应寄存器，其中除了传送数据计数器及接收数据计数器是只读，其他寄存器的值皆可更改。

Socket 编号 项目	1	2	3	4	5	6	7	8
本机通讯端 □	SR1118	SR1131	SR1144	SR1157	SR1170	SR1183	SR1196	SR1209
远程 IP H	SR1119	SR1132	SR1145	SR1158	SR1171	SR1184	SR1197	SR1210
远程 IP L	SR1120	SR1133	SR1146	SR1159	SR1172	SR1185	SR1198	SR1211
远程通信端 □	SR1121	SR1134	SR1147	SR1160	SR1173	SR1186	SR1199	SR1212
传送长度	SR1122	SR1135	SR1148	SR1161	SR1174	SR1187	SR1200	SR1213
传送地址 H	SR1123	SR1136	SR1149	SR1162	SR1175	SR1188	SR1201	SR1214
传送地址 L	SR1124	SR1137	SR1150	SR1163	SR1176	SR1189	SR1202	SR1215
接收长度	SR1125	SR1138	SR1151	SR1164	SR1177	SR1190	SR1203	SR1216
接收地址 H	SR1126	SR1139	SR1152	SR1165	SR1178	SR1191	SR1204	SR1217
接收地址 L	SR1127	SR1140	SR1153	SR1166	SR1179	SR1192	SR1205	SR1218
保持连接时 间(Sec)	SR1128	SR1141	SR1154	SR1167	SR1180	SR1193	SR1206	SR1219
传送数据计 数器	SR1129	SR1142	SR1155	SR1168	SR1181	SR1194	SR1207	SR1220
接收数据计 数器	SR1130	SR1143	SR1156	SR1169	SR1182	SR1195	SR1208	SR1221

5. 联机保持时间若设为 0，则表示没有该联机会保持等待状态，不会逾时。

6. UDP Socket 设定对应寄存器，其中除了传送数据计数器及接收数据计数器是只读，其他寄存器的值皆可更改。

Socket 编号 项目	1	2	3	4	5	6	7	8
本机通讯端 □	SR1222	SR1234	SR1246	SR1258	SR1270	SR1282	SR1294	SR1306
远程 IP H	SR1223	SR1235	SR1247	SR1259	SR1271	SR1283	SR1295	SR1317
远程 IP L	SR1224	SR1236	SR1248	SR1260	SR1272	SR1284	SR1296	SR1318
远程通信埠	SR1225	SR1237	SR1249	SR1261	SR1273	SR1285	SR1297	SR1309
传送长度	SR1226	SR1238	SR1250	SR1262	SR1274	SR1286	SR1298	SR1310
传送地址 H	SR1227	SR1239	SR1251	SR1263	SR1275	SR1287	SR1299	SR1311
传送地址 L	SR1228	SR1240	SR1252	SR1264	SR1276	SR1288	SR1300	SR1312
接收长度	SR1229	SR1241	SR1253	SR1265	SR1277	SR1289	SR1301	SR1313
接收地址 H	SR1230	SR1242	SR1254	SR1266	SR1278	SR1290	SR1302	SR1314
接收地址 L	SR1231	SR1243	SR1255	SR1267	SR1279	SR1291	SR1303	SR1315
传送数据计 数器	SR1232	SR1244	SR1256	SR1268	SR1280	SR1292	SR1304	SR1316
接收数据计 数器	SR1233	SR1245	SR1257	SR1269	SR1281	SR1293	SR1305	SR1317

7. 开启 TCP Socket 时 · Socket IP 和通讯端口编号设定。

远程 IP	本机通讯端口	远程通信端口	说明
0.0.0.0	0	0	不合法
0.0.0.0	不等于 0	0	仅用于主端应用 1. 接受同一个本机通讯端口编号的联机要求 2. 通过本机通讯端口接收任何装置的封包 3. 无法传送数据
0.0.0.0	0	不等于 0	不合法
特定 IP 地址	0	0	不合法
特定 IP 地址	不等于 0	0	仅用于主端应用 1. 接受设定之本机通讯端口编号和设定的特定 IP 地址的联机要求 2. 通过本机通讯端口接收特定 IP 地址的封包 3. 无法传送数据
特定 IP 地址	0	不等于 0	仅用于从端应用 1. 当 socket 联机要求传送数据时，系统会指定未使用的通讯端口为本机通讯端口。 2. 通过远程通信端口传送数据至特定 IP 地址启动
特定 IP 地址	不等于 0	不等于 0	1.接受设定的本机通讯端口编号、远程通信端口和特定 IP 地址的联机要求 2.通过远程通信端口传送数据至特定 IP 地址启动 3.通过本机通讯端口接收特定 IP 地址所传送的封包

8. 开启 TCP Socket 时，当此指令执行后，若无错误，则开始与远程装置进行联机，开启中标志为 ON，若联机成功则已联机标志为 ON，开启中标志为 OFF，反之若有错误则对应错误标志 ON。

TCP Socket 编号	开启中标志	已联机标志	错误标志
1	SM1273	SM1270	SM1277
2	SM1281	SM1278	SM1285
3	SM1289	SM1286	SM1293
4	SM1297	SM1294	SM1301
5	SM1305	SM1302	SM1309
6	SM1313	SM1310	SM1317
7	SM1321	SM1318	SM1325
8	SM1329	SM1326	SM1333

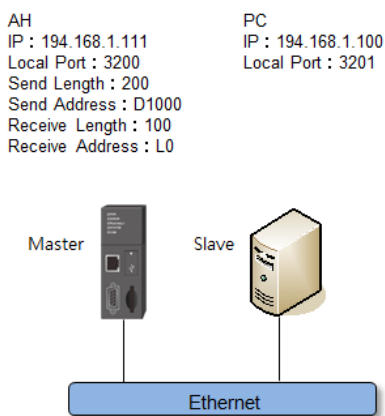
9. 开启 UDP Socket 时，当此指令执行后，若无错误，则联机已开启的标志为 ON，反之若有错误则对应错误标志 ON。

UDP Socket 编号	联机已开启标志	错误标志
1	SM1334	SM1338
2	SM1339	SM1343
3	SM1344	SM1348
4	SM1349	SM1353
5	SM1354	SM1358
6	SM1359	SM1363
7	SM1364	SM1368
8	SM1369	SM1373

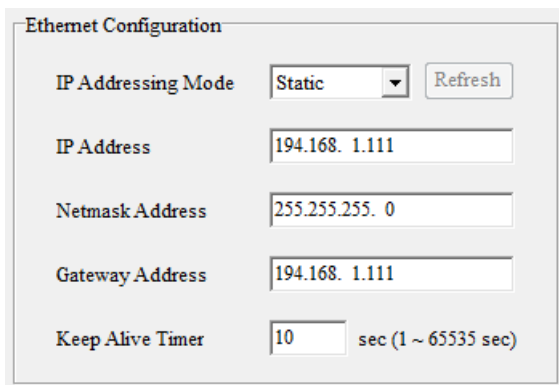
10. 本指令一般都是使用脉波执行型指令 (SOPENP) 。

程序范例：

1. 此程序范例的系统架构如下图，此范例说明如何以 PC 为从端，AH500 为主端建立一个 TCP 联机，此范例为使用第一组 TCP 联机。



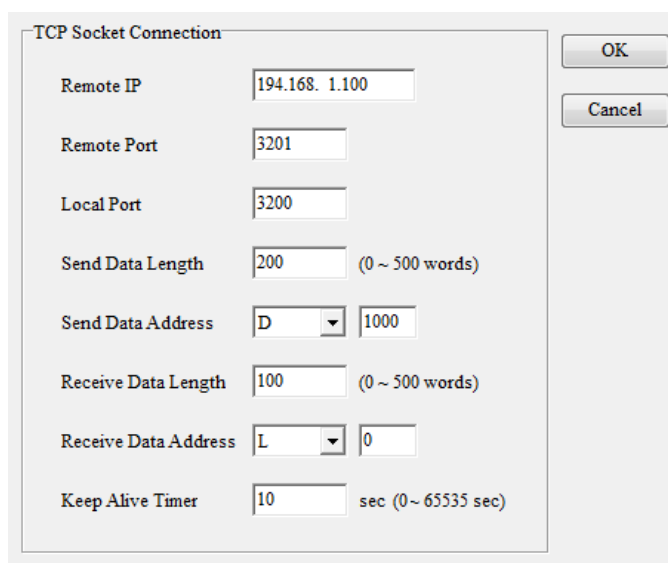
2. ISPSOft -> HWCONFIG (以太网络-基本设定)



The image shows a dialog box titled "Ethernet Configuration". It contains the following fields and controls:

- IP Addressing Mode:** A dropdown menu set to "Static" with a "Refresh" button to its right.
- IP Address:** A text input field containing "194.168. 1.111".
- Netmask Address:** A text input field containing "255.255.255. 0".
- Gateway Address:** A text input field containing "194.168. 1.111".
- Keep Alive Timer:** A text input field containing "10" followed by the text "sec (1 ~ 65535 sec)".

3. ISPSOft -> HWCONFIG (以太网络-进阶设定 -> 通讯接口 -> TCP)



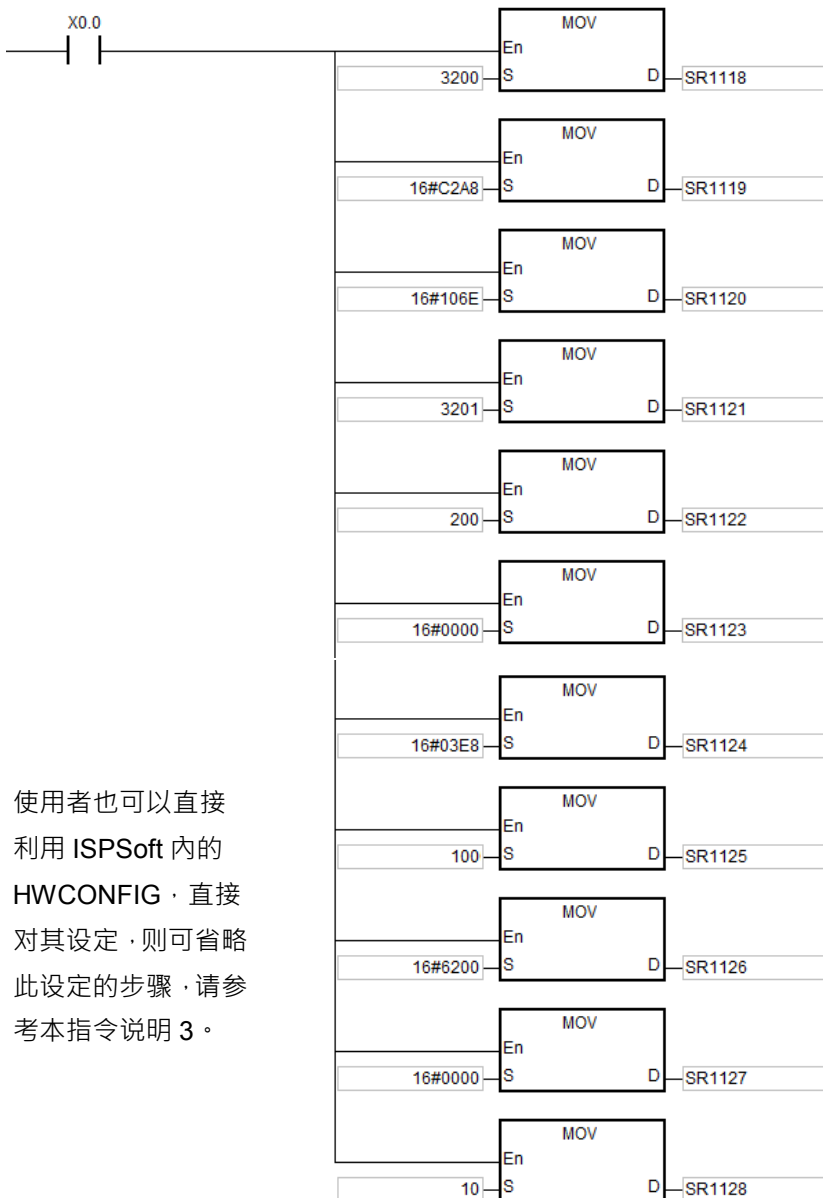
The image shows a dialog box titled "TCP Socket Connection". It contains the following fields and controls:

- Remote IP:** A text input field containing "194.168. 1.100".
- Remote Port:** A text input field containing "3201".
- Local Port:** A text input field containing "3200".
- Send Data Length:** A text input field containing "200" with the text "(0 ~ 500 words)" to its right.
- Send Data Address:** A dropdown menu set to "D" and a text input field containing "1000".
- Receive Data Length:** A text input field containing "100" with the text "(0 ~ 500 words)" to its right.
- Receive Data Address:** A dropdown menu set to "L" and a text input field containing "0".
- Keep Alive Timer:** A text input field containing "10" followed by the text "sec (0 ~ 65535 sec)".

Buttons for "OK" and "Cancel" are located on the right side of the dialog box.

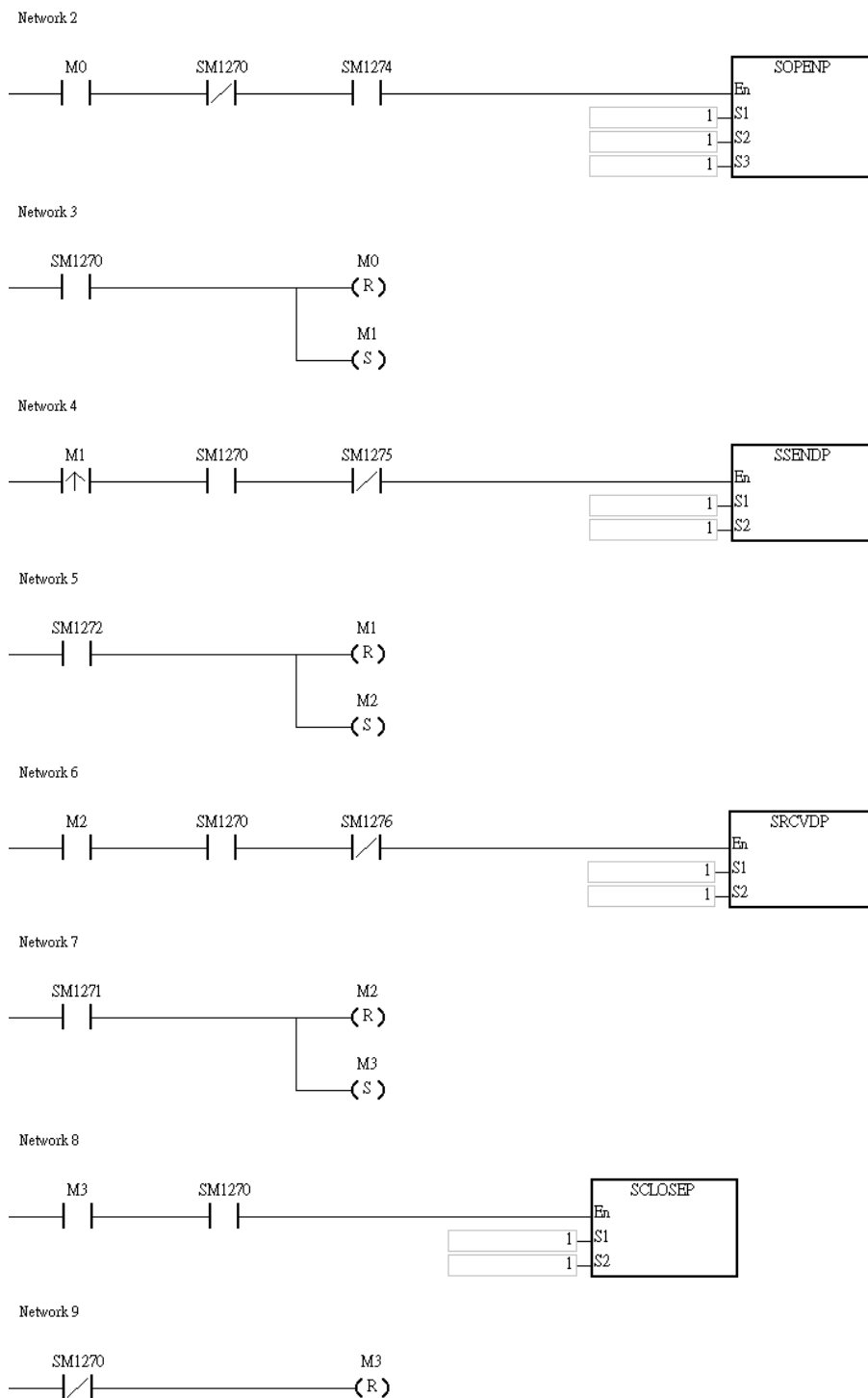
4. 当 X0.0=ON 时，设定 TCP Socket1 的信息，也可先行在 ISPSOft PLC 参数设定中设定，即可省略 Network 1。
5. 当 M0=ON 时，检查欲联机的 Socket 是否已联机或是联机中的状况，如果不是则正式进入联机程序，联机完成后清除 M0，并使 M2=ON。
6. 当 M1=ON 时，准备传送数据，检查欲传送的 Socket 是否为已联机，并且无数据在传送中的状态，若是则开始传送数据，若否则此指令无法运行，传送完成后清除 M1，并使 M1=ON。
7. 当 M2=ON 时，准备接收资料，检查欲接收的 Socket 是否为已联机，并且不是在接收中的状态；若已是，则此 Socket 转为接收中状态；若否，则此指令无法运行。当接收完成后清除 M2，并使 M3=ON。
8. 当 M3=ON 时，准备关闭联机，检查欲关闭的 Socket 是否为已联机状态，若已联机则开始关闭联机，若尚未联机则此指令无法运行，当联机关闭后，清除 M3。

Network 1



使用者也可以直接
利用 ISPSOft 內的
HWCONFIG · 直接
对其设定，则可省略
此设定的步骤，请参
考本指令说明 3。

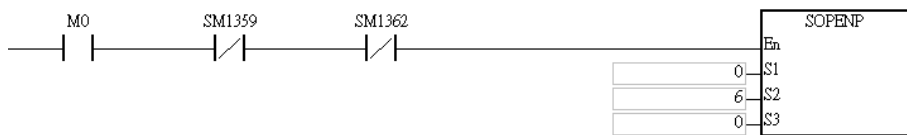
3



9. 此程序另一范例如下，此范例说明如何使用 AH 建立一个 UDP 联机，此范例为使用 UDP 第六组联机。
10. 当 M0=ON 时，检查欲联机的 Socket 是否处于联机已开启或是正在接收中状态，如果不是则正式进入联机程序，联机完成后清除 M0，并使 M1=ON。
11. 当 M1=ON 时，开始传送数据，传送完成后清除 M1，并使 M2=ON。
12. 当 M2=ON 时，准备接收资料，检查欲接收的 Socket 是否为已联机，并且检查是否在接收中状态，若于接收中状态则此 Socket 转为接收中状态，若不是接收中状态，则此指令无法运行，当数据接收完成后清除 M3，并使 M3=ON。

13.当 M3=ON 时，开始关闭联机，检查欲关闭的 Socket 是否为已联机状态，若已联机则开始关闭联机，若尚未联机则此指令无法运行，联机关闭后清除 M3。

Network 1



Network 2



Network 3



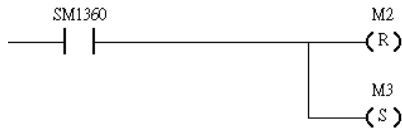
Network 4



Network 5



Network 6



Network 7



Network 8



3

补充说明：

1. S₁、S₂、S₃ 内容值超出范围时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
2. 当 AH 为主端，主动开启一个 TCP 联机时，若主端联机数已达上限，则会发生错误，对应的错误标志为 ON，错误码 16#600A，以上状况则指令不执行。

3. 当 AH 为主端，主动开启一个 TCP 联机时，若远程通信端口为 0，则会发生错误，对应的错误标志为 ON，错误码为 16#6202，以上状况则指令不执行。
4. 当开启一个 UDP 联机时，若主端联机数已达上限或是已有相同的联机，则会发生错误，对应的错误标志为 ON，错误码 16#600B，以上状况则指令不执行。
5. 当 AH 为从端，被动开启一个 TCP 联机时，若本机通讯端口已被使用，则会发生错误，对应的错误标志为 ON，错误码 16#600C；若本机通讯端口为 0，则会发生错误，对应的错误标志为 ON，错误码 16#6201，以上状况则指令不执行。
6. 当开启一个 UDP 联机时，若本机通讯端口已被使用，则会发生错误，对应的错误标志为 ON，错误码 16#600C，以上状况则指令不执行。
7. 当开启一个 TCP 联机时，欲联机的 IP，最高位为 0、127 或是大于 223 的值，则会发生错误，对应的错误标志为 ON，错误码 16#6200，以上状况则指令不执行。
8. 当开启一个 UDP 联机时，欲联机的 IP，最高位为 0、127 或是大于 223 的值，则会发生错误，对应的错误标志为 ON，错误码 16#6209，以上状况则指令不执行。
9. 当开启一个 UDP 联机时，若本机通讯端口和远程通信埠皆为 0，则会发生错误，对应的错误标志为 ON，错误码 16#620A，以上状况则指令不执行。
10. 当开启一个 TCP 联机时，若此 Socket 为已联机或在联机中，则会发生错误，对应的错误标志为 ON，错误码 16#6217，若此 Socket 正在关闭中，则会发生错误，对应的错误标志为 ON，错误码为 16#621A，以上状况则指令不执行。
11. 当联机过程中，对方主机放弃此联机，则会发生错误，对应的错误标志为 ON，错误码 16#6214；若对方主机回复逾时，则会发生错误，对应的错误标志为 ON，错误码 16#6212。
12. 如果主从端皆为使用 AH 系列 PLC，且逾时时间设定相同，如从端先发生逾时，则从端会自行切断联机，所以主端不会有错误标志发生，反之主端先发生逾时，则主端会发生错误标志，并切断联机。

FB/FC	脚本		操作数				功能					
FC		SSEND	P	$S_1 \cdot S_2$				通过已开启的 Socket 传送数据				

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●												
S_2		●												

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●	○	○		

脉波执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

符号：

SSEND	SSENDP
En	En
S1	S1
S2	S2

S_1 : Socket 模式. Word

S_2 : Socket 编号 Word

指令说明：

- S_1 为 Socket 模式，欲开启 TCP Socket 时， S_1 为 1，开启 UDP Socket 时， S_1 为 0， S_2 为 Socket 编号对应以太网中 Socket 设定。
- $S_1=0\sim 1$ ， $S_2=1\sim 8$ 。
- 使用本指令前，需先以 SOPEN 开启 Socket 联机，当已联机标志（TCP）或是已开启标志（UDP）为 ON 时，方可使用本指令进行数据传输。
- 通过 TCP Socket 传送数据时，当此指令执行后，若无错误，则开始传送数据，传送中标志为 ON，若传送成功则已传送标志为 ON，传送中标志为 OFF，反之若有错误则对应错误标志 ON。

TCP Socket 编号	传送中标志	已传送标志	错误标志
1	SM1275	SM1272	SM1277
2	SM1283	SM1280	SM1285
3	SM1291	SM1288	SM1293
4	SM1299	SM1296	SM1301
5	SM1307	SM1304	SM1309
6	SM1315	SM1312	SM1317
7	SM1323	SM1320	SM1325
8	SM1331	SM1328	SM1333

5. 通过 UDP Socket 传送数据时，当此指令执行后，若无错误，则数据已传送标志为 ON，反之若有错误则对应错误标志 ON。

UDP Socket 编号	已传送标志	错误标志
1	SM1336	SM1338
2	SM1341	SM1343
3	SM1346	SM1348
4	SM1351	SM1353
5	SM1356	SM1358
6	SM1361	SM1363
7	SM1366	SM1368
8	SM1371	SM1373

6. 本指令一般都是使用脉波执行型指令 (SSENDP)。
7. 使用此指令时，如欲传送奇数长度字节数据，可使用下列标志进行设定。

Socket 编号	奇数字节启动标志 (TCP)	奇数字节启动标志 (UDP)
1	SM1375	SM1383
2	SM1376	SM1384
3	SM1377	SM1385
4	SM1378	SM1386
5	SM1379	SM1387
6	SM1380	SM1388
7	SM1381	SM1389
8	SM1382	SM1390

奇数字节启动标志功能说明：

- 当传送的数据长度设定为 4 个 Word 时。(其相关设定与程序范例，请参考 SOPEN 指令章节)

TCP Socket Connection

Remote IP

Remote Port

Local Port

Send Data Length (0 ~ 500 words)

Send Data Address

Receive Data Length (0 ~ 500 words)

Receive Data Address

Keep Alive Timer sec (1 ~ 65535 sec)

- 当未使用奇数字节启动标志时，则实际传送的数据内容为 8 个 Bytes，若使用奇数字节启动标志时，则实际传送的数据内容为 7 个 Bytes，其实际输出的数据内容如下表。

主机传送数据地址内容							
D0		D1		D2		D3	
High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Bytes	High Byte	Low Byte
16#01	16#02	16#03	16#04	16#05	16#06	16#07	16#08
实际传送数据内容 (奇数字节启动标志 = OFF)							
0102 0304 0506 0708							
实际传送数据内容 (奇数字节启动标志 = ON)							
0102 0304 0506 07							

补充说明：

1. **S₁**、**S₂** 内容值超出范围时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
2. 在 TCP Socket 中，若传送欲读取的位置，不属于 Socket 允许的寄存器装置，则会发生错误，对应的错误标志为 ON，错误码 16#6203；若欲传送的长度超过 1000，则会发生错误，对应的错误标志为 ON，错误码 16#6204；若欲传送起始位置加上欲传送的长度，超过该装置的范围，则会发生错误，对应的错误标志为 ON，错误码 16#6205，以上状况则指令不执行。
3. 在 UDP Socket 中，若传送欲读取的位置，不属于 Socket 允许的寄存器装置，则会发生错误，对应的错误标志为 ON，错误码 16#620C；若欲传送的长度超过 1000，则会发生错误，对应的错误标志为 ON，错误码 16#620D；若传送取起始位置加上欲传送的长度，超过该装置的范围，则会发生错误，对应的错误标志为 ON，错误码 16#620E，以上状况则指令不执行。
4. 在 UDP Socket 中，传送数据时若远程通信端口为 0 时，则会发生错误，对应的错误标志为 ON，错误码 16#620B，以上状况则指令不执行。
5. 当传送过程中，对方主机放弃此联机，则会发生错误，对应的错误标志为 ON，错误码 16#6214；若对方主机回复逾时，则会发生错误，对应的错误标志为 ON，错误码 16#6212。
6. 当执行此指令时，若网络线未连接，则 SM1100 为 ON，错误码 16#600D。

FB/FC	脚本			操作数			功能			
FC		SRCVD	P	$S_1 \cdot S_2$			通过已开启的 Socket 接收数据			

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●												
S_2		●												

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●	○	○		

脉波执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

符号：

SRCVD		SRCVDP		S_1 : Socket 模式	Word
En		En		S_2 : Socket 编号	Word
S1		S1			
S2		S2			

指令说明：

- S_1 为 Socket 模式，欲开启 TCP Socket 时， S_1 为 1，开启 UDP Socket 时， S_1 为 0， S_2 为 Socket 编号对应以太网网络中 Socket 设定。
- $S_1=0\sim 1$ ， $S_2=1\sim 8$ 。
- 使用本指令前，需先以 SOPEN 开启 Socket 联机，当已联机标志（TCP）或是已开启标志（UDP）为 ON 时，方可使用本指令进行数据传输。
- 通过 TCP Socket 接收数据时，当此指令执行后，若无错误，则开始接收数据，接收中标志为 ON，若接收成功则已接收标志为 ON，接收中标志为 OFF，反之若有错误则对应错误标志 ON。

TCP Socket 编号	接收中标志	已接收标志	错误标志
1	SM1276	SM1271	SM1277
2	SM1284	SM1279	SM1285
3	SM1292	SM1287	SM1293
4	SM1300	SM1295	SM1301
5	SM1308	SM1303	SM1309
6	SM1316	SM1311	SM1317
7	SM1324	SM1319	SM1325
8	SM1332	SM1327	SM1333

5. 通过 UDP Socket 接收数据时，当此指令执行后，若无错误，则数据接收中标志为 ON，等到收到数据后，已接收标志为 ON，反之若有错误则对应错误标志 ON。

UDP Socket 编号	接收中标志	已接收标志	错误标志
1	SM1337	SM1335	SM1338
2	SM1342	SM1340	SM1343
3	SM1347	SM1345	SM1348
4	SM1352	SM1350	SM1353
5	SM1357	SM1355	SM1358
6	SM1362	SM1360	SM1363
7	SM1367	SM1365	SM1368
8	SM1372	SM1370	SM1373

6. 本指令一般都是使用脉波执行型指令 (SRCVDP)。

程序范例：

请参考 SOPEN 程序范例。

补充说明：

1. S_1 、 S_2 内容值超出范围时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
2. 在 TCP Socket 中，若接收后欲存取的位置，不属于 Socket 允许的寄存器装置，则会发生错误，对应的错误标志为 ON，错误码 16#6206；若欲存取的长度超过 1000，则会发生错误，对应的错误标志为 ON，错误码 16#6207；若欲存取起始位置加上欲存取的长度，超过该装置的范围，则会发生错误，对应的错误标志为 ON，错误码 16#6208，以上状况则指令不执行。
3. 在 UDP Socket 中，若接收后欲存取的位置，不属于 Socket 允许的寄存器装置，则会发生错误，对应的错误标志为 ON，错误码 16#620F；若欲存取的长度超过 1000，则会发生错误，对应的错误标志为 ON，错误码 16#6210；若欲存取起始位置加上欲存取的长度，超过该装置的范围，则会发生错误，对应的错误标志为 ON，错误码 16#6211，以上状况则指令不执行。
4. 若实际接收封包的长度，大于设定的接收长度，则只会存取设定接收长度的数据，多余的数据将会被忽略，并产生错误，对应的错误标志为 ON，错误码 16#6213。
5. 当接收过程中，对方主机放弃此联机，则会发生错误，对应的错误标志为 ON，错误码 16#6214；若对方主机回复逾时，则会发生错误，对应的错误标志为 ON，错误码 16#6212。
6. 当执行此指令时，若网络线未连接，则 SM1100 为 ON，错误码 16#600D。

FB/FC	脚本			操作数	功能
FC		SCLOSE	P	$S_1 \cdot S_2$	关闭 Socket

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S_1		●												
S_2		●												

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●	○	○		

脉波执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

符号：

SCLOSE	SCLOSEP
En	En
S_1	S_1
S_2	S_2

 S_1 : Socket 模式

Word

 S_2 : Socket 编号

Word

指令说明：

- S_1 为 Socket 模式，欲关闭 TCP Socket 时， S_1 为 1，关闭 UDP Socket 时， S_1 为 0； S_2 为 Socket 编号对应以太网网络中 Socket 设定。
- $S_1=0\sim 1$ ， $S_2=1\sim 8$ 。
- 关闭 Socket 前必须确认此 Socket 为联机状态，否则此指令不会执行。
- TCP Socket 若由主端执行关闭指令，则从端会继续对本机通讯端口进行监听（联机中标志为 ON）；若由从端执行关闭指令，则从端在关闭完成后也取消对本机通讯端口的监听，除上述之外，关闭后对应的标志皆为 OFF。
- UDP Socket 执行关闭指令后，对应的标志皆为 OFF。
- 关闭 TCP Socket 时，当此指令执行后，若无错误，则开始关闭与远程通信端口的联机，关闭中标志为 ON，若关闭成功则关闭中标志为 OFF，反之若有错误则对应错误标志 ON。

Socket 编号	关闭中标志	错误标志
1	SM1274	SM1277
2	SM1282	SM1285
3	SM1290	SM1293
4	SM1298	SM1301
5	SM1306	SM1309
6	SM1314	SM1317
7	SM1322	SM1325
8	SM1330	SM1333

- 关闭 UDP Socket 时，当此指令执行后，若无错误，则已联机标志为 OFF，反之若有错误则对应错误标志 ON。

Socket 编号	错误标志
1	SM1338
2	SM1343
3	SM1348
4	SM1353
5	SM1358
6	SM1363
7	SM1368
8	SM1373

8. 本指令一般都是使用脉波执行型指令 (**SCLOSEP**) 。

程序范例：

请参考 SOPEN 程序范例。

补充说明：

1. **S₁**、**S₂** 内容值超出范围时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
2. 当关闭过程中，对方主机放弃此联机，则会发生错误，对应的错误标志为 **ON**，错误码 **16#6214**；若对方主机回复逾时，则会发生错误，对应的错误标志为 **ON**，错误码 **16#6212**。
3. 当执行此指令时，若网络线未连接，则 SM1100 为 ON，错误码 16#600D。

3

3.25 储存卡读写指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>MWRIT</u>	—	—	将 PLC 数据数据写入到储存卡里	13
FC	<u>MREAD</u>	—	—	将数据数据从储存卡里读取到 PLC	13
FC	<u>MTWRIT</u>	—	—	将文字写入到储存卡里	11

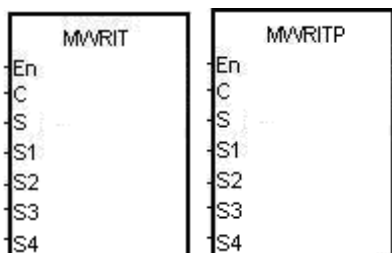
FB/FC	指令码		操作数				功能						
FC	MWRIT	P	C, S, S₁, S₂, S₃, S₄				将 PLC 数据数据写入到储存卡里						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
C		●					●							
S		●					●							
S₁			●					●						
S₂		●					●							
S₃		●					●							
S₄			●					●						

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●				
S₄								●	●				●	○	○		

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

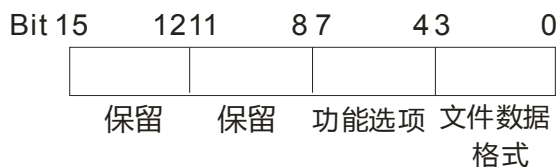


- C** : 控制参数
- S** : 数据源
- S₁** : 欲写入的数据长度
- S₂** : 换行
- S₃** : 文件名
- S₄** : 文件内的数据地址

指令说明：

1. 操作数说明：

- 控制参数 **C**：



说明项目	代码	说明
文件数据格式	0	二进制。默认值。
		副文件名为.dmd。
		每个数据的单位为 Word。
	1	每个数据之间用逗号做为区隔。
		每个数据的单位为 Word。
		副文件名为.csv。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	2	每个数据之间用逗号做为区隔。
		每个数据的单位为 DWord。
		副文件名为.csv。
		采用 ASCII 的字符。
	3	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 Word。
		副文件名为.txt。
		采用 ASCII 的字符。
	4	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 DWord。
		副文件名为.txt。
		采用 ASCII 的字符。
	5	数据与数据之间没有区隔符号。
		每个数据的单位为 Word。
		副文件名为.txt。
		采用 ASCII 的字符。
6	数据与数据之间没有区隔符号。	
	每个数据的单位为 DWord。	
	副文件名为.txt。	
	采用 ASCII 的字符。	
		储存的数据采用 16 进制来表示。

说明项目	代码	说明
功能选项	0	附加 (Append)。将用户欲写入的数据，从文件的最后一个数据开始写入。 默认值。 若文件不存在，则会自动产生。
	1	覆写 (overwrite)。将用户欲写入的数据，从指定的文件的数据地址开始写入。 若文件不存在，则会自动产生。
保留	-	固定为 0。

- 数据源 **S**：数据源的起始地址。
- 欲写入的数据长度 **S₁**：用来设定欲写入文件的数据数据长度，0 表示不写入。

项目	说明
计算单位	计算的单位需依照操作数 C 的文件数据格式来做为辨别，若文件数据格式为 0、1、3 或 5 (请参考操作数 C)，则计算单位为 Word；若文件数据格式为 2、4 或 6 (请参考操作数 C)，则计算单位为 DWord。
参数单位	DWord。
数据数据长度	设定长度不可以超出该装置范围 (有关装置范围请参考 Ch2 装置说明)，文件写入的数据量不可以超出 4GB。

- 换行 **S₂**：可用来设定写入多少数据 (Word/DWord) 后，换到下一行，可设定范围为 0~256。(0 表示不换行)
- 文件名 **S₃~S₃₊₄**：**S₃** 占用连续 5 个装置来指定写入数据的文件名，且文件名不可以超过九个字符 (包含终止位 16#00)，如果没有填入字符串终止符会有错误。当有读取到终止位时，将会停止读取字符，且检查文件名是否合法。合法命名的字符为 A~Z、a~z 与 0~9。副文件名会依据 **C** 的文件数据格式，而自动给予副文件名，生成的文件会在预设文件夹里。若文件名为“Test1”，则写入装置样式如下：

D	'e'	'T'	ASCII代码 ➔	D	16#65	16#54
D+1	't'	's'		D+1	16#74	16#73
D+2	NUL	'1'		D+2	16#00	16#31

- 默认文件夹路径：

机种名称	文件夹路径
AH10EMC-5A	PLC CARD\AH10EMC\UserProg

- 文件内的数据地址 **S₄**：用来指定写入文件的起始地址。

说明项目	说明
计算单位	计算的单位需依照操作数 C 的文件数据格式来做为辨别，若文件数据格式为 0、1、3 或 5，则计算单位为 Word；若文件数据格式为 2、4 或 6，则计算单位为 DWord。(请参考操作数 C)

说明项目	说明
S ₄ 的参数单位	此参数单位为 Dword。
S ₄ 使用方式	若 C 的功能项目为 0 (附加)，则此操作数不会被使用到。 若 C 的功能项目为 1 (覆写)，则会将数据写入到 S ₄ 指定在文件里的数据地址。 此 S ₄ 指所定的数据地址不可以超出文件所拥有的数据总个数。当 S ₄ 为 0 时，表示为第一个位置。

2. 相关标志

标志	说明
SM450	若储存卡有存在于 CPU 模块上，则此标志会为 ON 的状态。
SM451	储存卡在写保护的模式里
SM452	储存卡正在读/写的动作
SM453	当储存卡有发生错误的时候，此 SM 会设定为 ON。当此 SM 为 ON 时，需自行设定为 OFF。错误代码请参考 SR453。

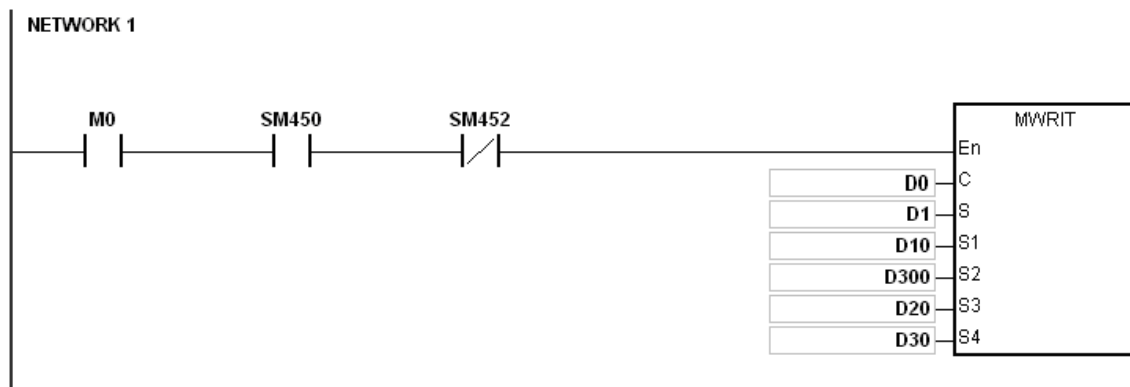
3. 相关错误代码 (SR453)

错误代码	说明
16#005E	当 CPU 模块在进行储存卡初始化流程时，发生错误而无法进行初始化流程。
16#005F	路径错误或文件不存在。
16#0060	无法产生预设文件夹。
16#0061	储存卡内存空间不足。
16#0062	储存卡为写保护。
16#0063	当数据写入到文件时，发生错误。
16#0064	当要读取文件数据时，无法读取文件的数据。
16#0065	文件属性为只读状态。

4. 当写入文件数据格式为 0 时，则读取文件数据格式也一定要为 0，不然无法读取，且 SM453 会 ON 起来。其它文件数据格式依此类推。

程序范例：

当储存卡有插入 CPU 模块的时候，SM450 为 ON 的状态；当 MWRIT 开始运作的时候，SM452 为 ON；当 MWRIT 完成写入动作的时候，SM452 为 OFF。可使用脉冲指令 MWRITP，避免指令持续执行，持续写入储存卡，导致写入次数超过储存卡限制而储存卡故障。



操作数	设定值	说明
D0	16#0011	写入文件。 文件数据格式：每个数据之间有用逗号来作为区隔。每个数据的单位为 Word。副文件名为.csv。采用 ASCII 的字符。
D1	-	写入数据内容。
D10、D11	16#00000030	总共写入长度为 48 个 Word。
D300	16#000A	每写入 10 个 Word 换下一行。
D20	D20=16#6554 D21=16#7473 D22=16#0031	写入文件名为“Test1”。
D30、D31	16#00000000	从文件最初的地址开始写入。

补充说明：

1. C 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. S₁ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. S₂ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
4. S₃ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

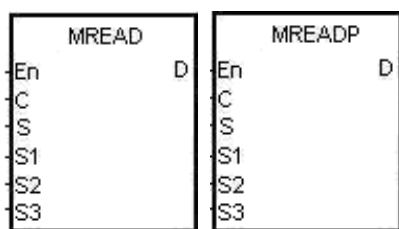
FB/FC	指令码		操作数				功能							
FC	MREAD	P	C, S, S₁, S₂, S₃, D				将数据数据从储存卡里读取到 PLC							

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
C		●					●							
S		●					●							
S₁			●					●						
S₂		●					●							
S₃			●					●						
D		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●	○	○		
D								●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



C : 控制参数

S : 文件名

S₁ : 文件的数据地址

S₂ : 系统内容使用

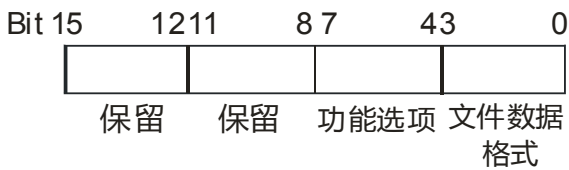
S₃ : 欲读取的数据长度

D : 数据目的地

指令说明：

1. 操作数说明：

- 控制参数 **C**：



说明项目	代码	说明
文件数据格式	0	二进制 (0 为默认值)。
		副文件名为 .dmd。
		每个数据的单位为 Word。
	1	每个数据之间用逗点做为区隔。
		每个数据的单位为 Word。
		副文件名为 .csv。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	2	每个数据之间用逗点做为区隔。
		每个数据的单位为 DWord。
		副文件名为 .csv。
		采用 ASCII 的字符。
	3	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 Word。
		副文件名为 .txt。
		采用 ASCII 的字符。
	4	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 DWord。
		副文件名为 .txt。
		采用 ASCII 的字符。
	5	数据与数据之间没有区隔符号。
		每个数据的单位为 Word。
		副文件名为 .txt。
		采用 ASCII 的字符。
6	数据与数据之间没有区隔符号。	
	每个数据的单位为 DWord。	
	副文件名为 .txt。	
	采用 ASCII 的字符。	
		储存的数据采用 16 进制来表示。

3

说明项目	代码	说明
功能项目	0	读取指定文件内的数据数据 (0 为默认值)。
	1	计算文件内有多少个 Words/DWords。此计算结果会写入 D 与 D+1 里。 若在操作数 C 中的文件数据格式为 0、1、3、5, 则计算单位为 Word ; 若文件数据格式为 2、4、6, 则计算单位为 DWord。
保留	-	固定为 0

- 文件名 **S~S+4** : **S** 占用连续 5 个装置来指定写入数据的文件名, 且文件名不可以超过九个字符 (包含终止位 16#00), 如果没有填入字符串终止符会有错误。当有读取到终止位时, 将会停止读取字符, 且检查文件名是否合法。合法命名的字符为 A~Z、a~z 与 0~9。副文件名会依据 **C** 的文件数据格式, 而自动给予副文件名, 生成的文件会在预设文件夹里。若文件名为“Test1”, 则写入装置样式如下:

D	'e'	'T'	ASCII代码	D	16#65	16#54
D+1	't'	's'	→	D+1	16#74	16#73
D+2	NUL	'1'		D+2	16#00	16#31

- 默认文件夹路径:

机种名称	文件夹路径
AH10EMC-5A	PLC CARD\AH10EMC\UserProg

- 文件的数据地址 **S₁**: 用来指定读取文件的起始地址。

项目	说明
计算单位	计算的单位需依照操作数 C 的文件数据格式来做为辨别, 若文件数据格式为 0、1、3 或 5, 则计算单位为 Word ; 若文件数据格式为 2、4 或 6, 则计算单位为 DWord。(请参考操作数 C)
参数单位	此参数单位为 DWord。
使用方式	所设定的起始数据地址不可以超过文件内所拥有的数据总个数。当 S₁ 为 0 时, 表示为第一个位置。

- 欲读取的数据长度 **S₃**: 设定的数据数据长度不可以超出装置最大范围。若超出文件所记载的数据 (Word/DWord) 个数, 读取的数据各数会以文件记载的为主, 0 表示不读取, 单位为 DWord。
- 数据目的地 **D**: 数据 (Word/DWord) 储存的起始装置地址。

2. 相关标志

标志	说明
SM450	若储存卡有存在于 CPU 模块上, 则此标志会为 ON 的状态。
SM451	储存卡在写保护的模式里。
SM452	储存卡正在读/写的动作。
SM453	当储存卡有发生错误的时候, 此 SM 会设定为 ON。当此 SM 为 ON 时, 需自行设定为 OFF。错误代码请参考 SR453。

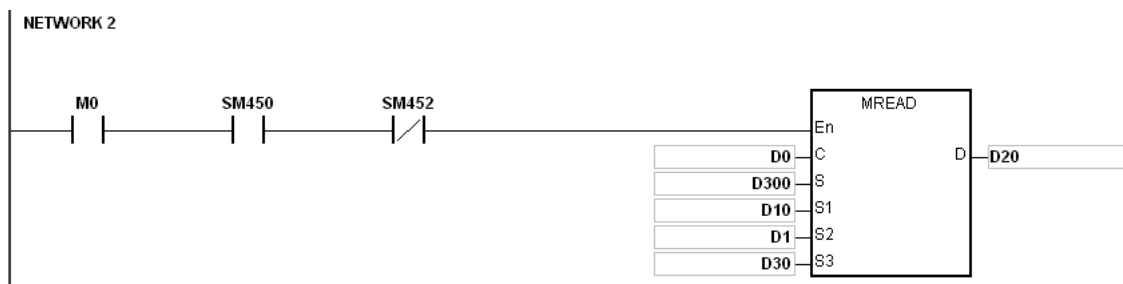
相关错误代码 (SR453)

错误代码	说明
16#005E	当 CPU 模块再进行储存卡初始化流程时，发生错误而无法进行初始化流程。
16#005F	路径错误或文件不存在。
16#0060	无法产生预设文件夹。
16#0061	储存卡内存空间不足。
16#0062	储存卡为写保护。
16#0063	当数据写入到文件时，发生错误。
16#0064	当要读取文件数据时，无法读取文件的数据。

- 当写入文件数据格式为 0 时，则读取文件数据格式也一定要为 0，不然无法读取，且 SM453 会 ON 起来。其它文件数据格式依此类推。

程序范例：

当储存卡有插入 CPU 模块的时候，SM450 为 ON。当 MREAD 开始运作的时候，SM452 为 ON 的状态，当 MREAD 完成读取动作的时候，SM452 为 OFF。



操作数	设定值	说明
D0	16#0011	读取文件数据 文件数据格式：每个数据之间有用逗号来作为区隔。每个数据的单位为 Word。副文件名为.csv。采用 ASCII 的字符。
D300	D300=16#6554 D301=16#7473 D302=16#0031	读取文件名为“Test1”
D10、D11	16#00000000	从文件最初的地址开始读取
D1	16#000A	每读取 10 个 Word 换下一行
D30、D31	16#00000020	总共读取长度为 32 个 Word
D20	-	将读取的数据储存至 D20

补充说明：

- C 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
- S₂ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
- S₃ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
- D 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

FB/FC	指令码		操作数				功能						
FC	MTWRIT	P	C, S, S₁, S₂, S₃				将文字写入到储存卡里						

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
C		●					●							
S		●					●							
S₁, S₂		●					●							
S₃		●					●							

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
C								●	●				●	○	○		
S								●	●				●				
S₁, S₂								●	●				●	○	○		
S₃								●	●				●				

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：

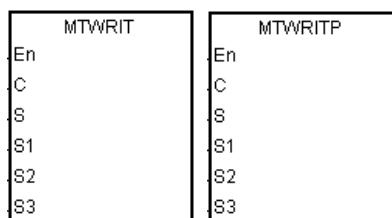
C : 控制参数

S : 数据源

S₁ : 数据长度

S₂ : 分隔符

S₃ : 文件名



指令说明：

1. 操作数说明：

- 控制参数 **C**：

参数数值	说明
0 (append)	若文件存在，将装置的数据写在文件的最后一个 byte。 若文件不存在，会自动产生出来。
1 (overwrite)	若文件存在，则新的数据会覆盖到旧的数据，文件的大小以新的数据为主。 若文件不存在，会自动产生出来。

- **数据源 S**：指定要写入储存卡的装置数据源。若欲写入“12345”的字符串到文本文件里的话，则储存在装置里的数据如下图所示。由于以 byte 做为基本单位，所以储存在 Word 的时候，第一个 byte 会在 Word 的 low byte 里，第二个 byte 会在 Word 的 High byte 里，其于以此类推。最后一个 byte 为终止符号“16#00”，表示写入结束。

D300	D300+1	D300+2
byte 2	byte 1	byte 4
byte 4	byte 3	byte 6
byte 5		
16#32	16#31	16#34
16#33	16#00	16#35

- **数据长度 S₁**：用来决定写入到储存卡的数据长度，单位为 byte。不能超过指定装置的最大范围。每次写入的长度不可以超过 255 bytes，0 表示不写入。
- **分隔符 S₂**：每次写完数据到储存卡的时候，会将此符号加入到数据里的最后一个 byte 的下两个 bytes。例：**S₁=N**，**S₂**分隔符之情况如下方表格，则 **S₂**分隔符写入到储存卡的情况会如下方表格之说明。

S ₂ 操作数		说明
High byte	Low byte	
16#00	16#00 或非 16#00	写入 N 个 Bytes 到文件里。
非 16#00	16#00	写入 N+1 个 Bytes 到文件里，其中 N+1 为 S ₂ 操作数的 High Byte。
非 16#00	非 16#00	写入 N+2 个 Bytes 到文件里，其中 N+1 为 S ₂ 操作数的 High Byte，N+2 为 S ₂ 操作数的 Low Byte。

- **文件名 S₃~S₃+4**：S₃ 占用连续 5 个装置来指定写入数据的文件名，且文件名不可以超过九个字符（包含终止位 16#00），如果没有填入字符串终止符会有错误。当有读取到终止位时，将会停止读取字符，且检查文件名是否合法。合法命名的字符为 A~Z、a~z 与 0~9。副文件名会依据 C 的文件数据格式，而自动给于副文件名，生成的文件会在预设文件夹里。若文件名为“Test1”，则写入装置样式如下：

D	'e'	'T'	ASCII代码	D	16#65	16#54
D+1	't'	's'		D+1	16#74	16#73
D+2	NUL	'1'		D+2	16#00	16#31

- **默认文件夹路径：**

机种名称	文件夹路径
AH10EMC-5A	PLC CARD\AH10EMC\UserProg

2. 相关标志

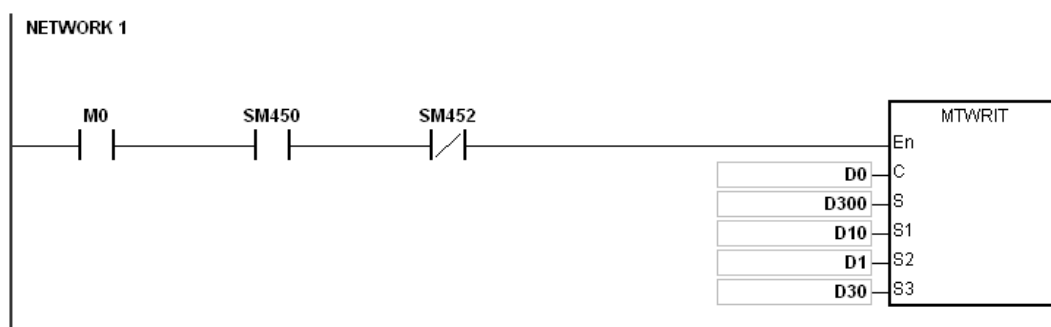
标志	说明
SM450	若储存卡有存在于 CPU 模块上，则此标志会为 ON 的状态。
SM451	储存卡在写保护的模式里
SM452	储存卡正在读/写的动作
SM453	当储存卡有发生错误的时候，此 SM 会设定为 ON。当此 SM 为 ON 时，需自行设定为 OFF。错误代码请参考 SR453。

相关错误代码 (SR453)

错误代码	说明
16#005E	当 CPU 模块再进行储存卡初始化流程时，发生错误而无法进行初始化流程。
16#005F	路径错误或文件不存在
16#0060	无法产生预设文件夹
16#0061	储存卡内存空间不足
16#0062	储存卡为写保护
16#0063	当数据写入到文件时，发生错误
16#0064	当要读取文件数据时，无法读取文件的数据
16#0065	文件属性为只读状态

程序范例：

当储存卡有插入 CPU 模块的时候，SM450 会为 ON 的状态，且接点会闭合。当 MTWRIT 开始运作的时候，SM452 会为 ON 的状态，且接点会打开，当 MTWRIT 完成写入动作的时候，SM452 会为 OFF 的状态，且接点会闭合。



操作数	设定值	说明
D0	16#0001	覆写文件 文件数据格式：每个字符的单位为 byte。副文件名为.txt。采用 ASCII 的字符。将 D300 的内容写入到文件里。
D300	-	写入数据内容
D10	16#000A	写入字符串长度为 32 个 bytes。
D1	16#0A00	每次写完数据后，将此符号加入到最后一个 byte
D30	D30=16#6554 D31=16#7473 D32=16#0031	写入的文件名为 Test1

补充说明：

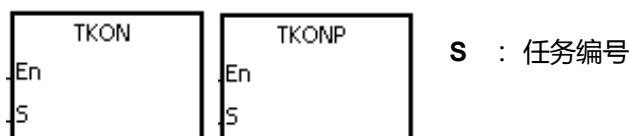
1. C 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. S₁ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. S₃ 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

3.26 任务控制指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>TKON</u>	-	✓	Cyclic 任务 Task 启动	3
FC	<u>TKOFF</u>	-	✓	Cyclic 任务 Task 关闭	3

FB/FC	指令码			操作数				功能									
FC		TKON	P	S				Cyclic 任务 Task 启动									
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●			●		●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●						●	●		●						S
脉冲执行型				16 位指令				32 位指令									
AH 运动控制主机				AH 运动控制主机				-									

图形：



指令说明：

1. 启动 S 所指定的 Cyclic Task。
2. 当 PLC 由 STOP→RUN 时，会依据 ISPSOft 程序下载时 Cyclic Task 是否启动来执行。
3. 操作数说明：
 - S 的范围为 0~31。
 - Tasks 的建立与启动请参考 ISPSOft 使用手册。

程序范例：

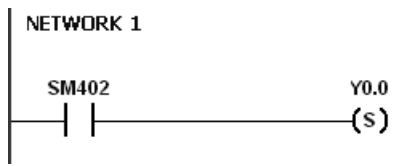
程序写入主机时，只指定启动 Task Cyclic_0 任务，PLC RUN 后，因为程序中指示 TKON 1 表示执行 Task Cyclic_1，所以 Y0.0=ON。

在 ISPSOft 中先建立两个 Cyclic Tasks，只先指定 Cyclic_0 在 PLC RUN 时为 On，Cyclic_1 先预设为不执行。

Cyclic_0 的 Prog0 中 TKON 1 启动 Cyclic_1



Cyclic_1 的 Prog1 的程序就会被执行



补充说明：

有关任务的说明，请参考 **ISPSOft 使用手册**。

FB/FC	指令码			操作数				功能									
FC		TKOFF	P	S				Cyclic 任务 Task 关闭									
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●						●	●		●						S
脉冲执行型				16 位指令				32 位指令									
AH 运动控制主机				AH 运动控制主机				-									

图形：



指令说明：

1. 关闭 S 所指定的 Cyclic Task。
2. 当 PLC 由 STOP→RUN 时，会依据 ISPSOft 程序下载时 Cyclic Task 是否启动来执行。
3. 操作数说明：
 - S 的范围为 0~31。
 - Tasks 的建立与启动请参考 ISPSOft 使用手册。

程序范例：

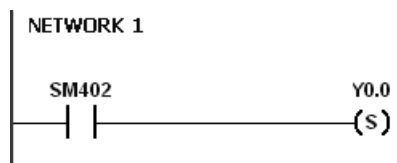
程序写入主机时，就指定启动 Task Cyclic_0 及 Tasks Cyclic_1 任务，PLC RUN 后，因为 TKOFF1 表示关闭 Task Cyclic_1 因此 Y0.0=OFF。

在 ISPSOft 中先建立两个 Cyclic Tasks，先指定 Cyclic_0 和 Cyclic_1 在 PLC RUN 时为执行。

Cyclic_0 的 Prog0 中 TKOFF 1 关闭 Cyclic_1



Cyclic_1 的 Prog1 的程序不会执行



补充说明：

有关任务的说明，请参考 **ISPSoft 使用手册**。

3.27 SFC控制指令

FB/FC	指令码 (位)		P 指令	功能	Step
	16	32			
FC	<u>SFCRUN</u>	-	-	SFC 启动	7
FC	<u>SFCPSE</u>	-	-	SFC 暂停	5
FC	<u>SFCSTP</u>	-	-	SFC 停止	3

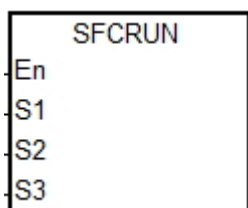
FB/FC	指令码		操作数				功能					
FC	SFCRUN		S ₁ , S ₂ , S ₃				SFC 启动					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S ₁														
S ₂		●					●							
S ₃														

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁																	
S ₂	●	●						●	●		●			○	○		
S ₃																	

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S₁ : SFC 名称

S₂ : 控制参数

S₃ : 要启动的 STEP 名称

指令说明：

1. 依照 S₂ 的设定，启动 S₁ 所指定的 SFC 程序。
2. 指令执行时，S₁ 所指定的 SFC 程序并不会立即启动，要等到扫描到 SFC 程序时才会启动。
3. 操作数说明：
 - S₁ 为 SFC 程序的名称。
 - S₂=0，启动 S₁ 指定的 SFC 程序，SFC 中的 SFC/STEP/ACTION/TRANSITION 状态跟参数会被清除，并从初始 STEP 开始执行。
 - S₂=1，启动 S₁ 指定的 SFC 程序，SFC 中的 SFC/STEP/ACTION/TRANSITION 状态跟参数会被清除，并从 S₃ 指定的 STEP 开始执行。
 - S₂=2，启动 S₁ 指定的 SFC 程序，不会清除 SFC 中的 SFC/STEP/ACTION/TRANSITION 状态跟参数，并从原本暂停的位置开始执行。
 - S₃ 为 S₁ 指定的 SFC 程序中的 STEP 名称。
4. S₂ 的范围 0~2，超出范围时，视为 0。

5. 当 SFC 程序状态为 RUN 时，执行此指令无效。

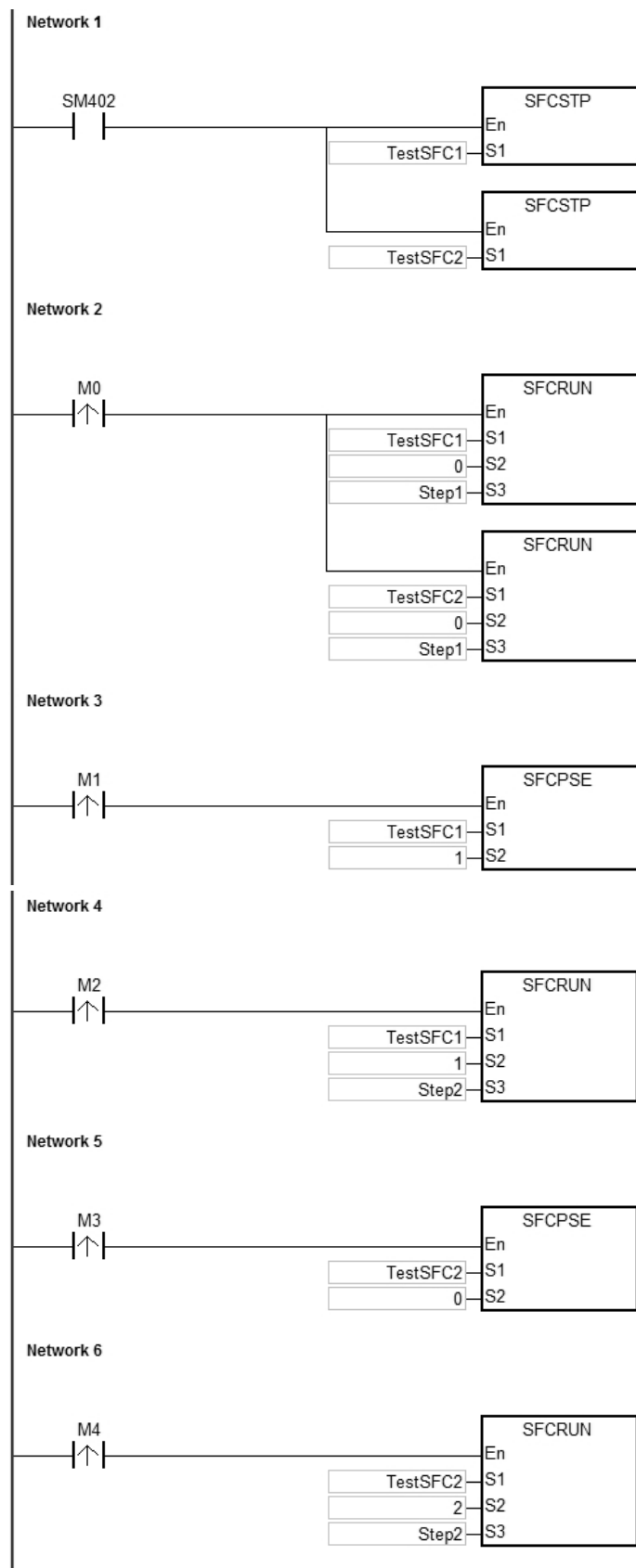
程序范例：

在程序中建立一个 LD 语言 POU 名称为 Main 及两个 SFC 语言 POU 名称为 TestSFC1 与 TestSFC2。

1. 系统执行 (RUN) 时，TestSFC1 与 TestSFC2 立刻执行 SFCSTP 动作，两个 SFC POU 会停止执行。
2. 当 M0 由 OFF→ON 时，TestSFC1/ TestSFC2 POU 执行动作 (SFCRUN)^{注一}，执行动作请参考 TestSFC1/ TestSFC2 POU 程序内容。设定 S_2 控制参数 0，会清除 SFC 中之状态及参数，并且从 Step1 开始执行。设定 S_2 控制参数 1，会清除 SFC 中之状态及参数，并且从 S_3 指定的 Step 开始执行。
3. 当 M1 由 OFF→ON 时，TestSFC1 POU 会暂停，控制参数设为 1，SFC 中正在执行中的 ACTION，程序中的输出会被清除 (做 Final scan)。
4. 当 M2 由 OFF→ON 时，TestSFC1 POU 执行动作，执行动作请参考 TestSFC1 POU 程序内容。设定控制参数 1，会清除 SFC 中的状态及参数，并且从 Step2 开始执行。
5. 当 M3 由 OFF→ON 时，TestSFC2 POU 会暂停，控制参数设为 0，SFC 中的输出保持不变，不做 Final scan。
6. 当 M4 由 OFF→ON 时，TestSFC2 POU 执行动作，执行动作请参考 TestSFC2 POU 程序内容。设定控制参数 2，会保持 SFC 中之状态及参数，并且从暂停时执行到的 Step 开始执行。
 - S 的范围为 0~31。
 - Task 的建立与启动请参考 ISPSOft 使用手册。

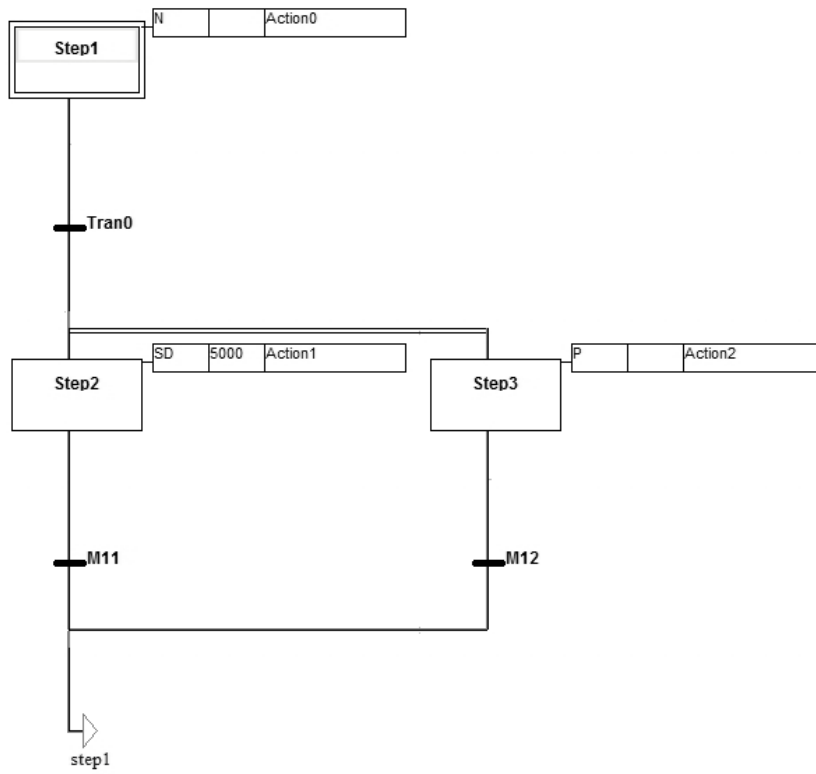
注一：SFCRUN 要等到下一个扫描时间才会启动指定的 SFC POU。

Main POU

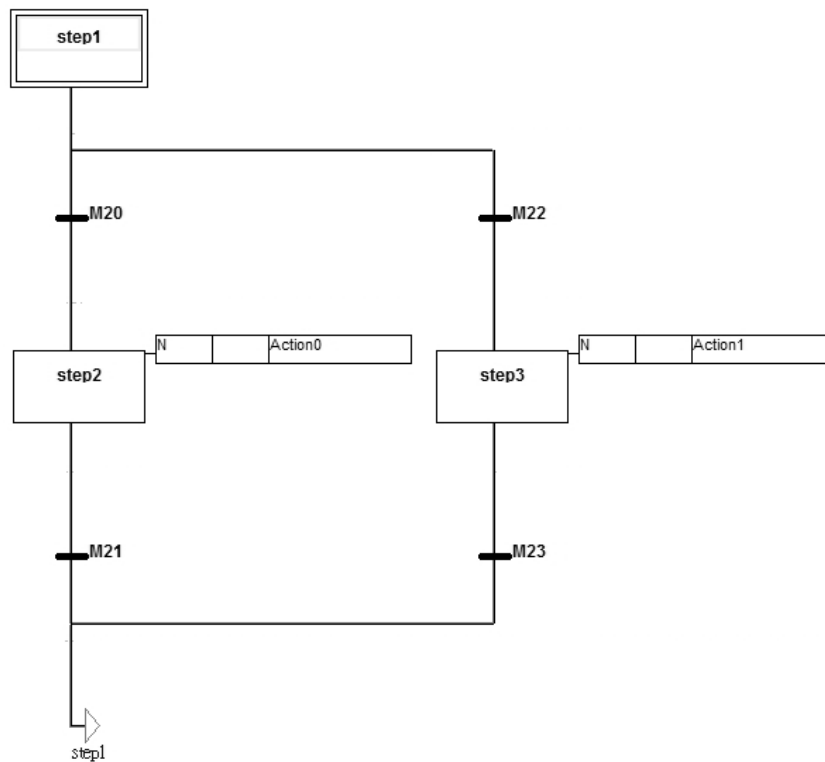


3

TestSFC1 POU



TestSFC2 POU

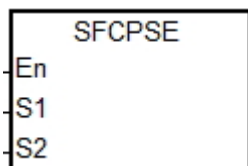


补充说明：

有关 SFC 的说明，请参考 ISPSOft 使用手册。

FB/FC	指令码			操作数							功能						
FC	SFCPSE			S ₁ , S ₂							SFC 暂停						
数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S ₁																	
S ₂		●					●										
装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁																	
S ₂	●	●						●	●		●			○	○		
							脉冲执行型			16 位指令				32 位指令			
							AH 运动控制主机			AH 运动控制主机				-			

图形：



S₁ : SFC 名称

S₂ : 控制参数

指令说明：

1. 依照 S₂ 的设定，暂停 S₁ 所指定的 SFC 程序。
2. 指令执行时，S₁ 所指定的 SFC 程序并不会立即暂停，要等到扫描到 SFC 程序时才会暂停。
3. 进入暂停时，SFC 中的 STEP/ACTION/TRANSITION 状态跟参数不变。
4. 操作数说明：
 - S₁ 为 SFC 程序的名称。
 - S₂=0，暂停 S₁ 指定的 SFC 程序，SFC 程序中的输出保持不变，正在执行中的 ACTION，不执行 Final Scan。
 - S₂=1，暂停 S₁ 指定的 SFC 程序，正在执行中的 ACTION 程序中的输出会被清除（执行 Final Scan）。
5. S₂ 的范围 0~1，超出范围时，视为 0。
6. 当 SFC 程序状态为 PAUSE/STOP，执行此指令无效。

程序范例：

请参考 SFCRUN 程序范例。

补充说明：

有关 SFC 的说明，请参考 ISPSOFT 使用手册。

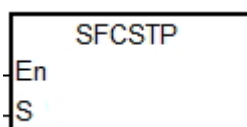
FB/FC	指令码			操作数				功能					
FC	SFCSTP			S				SFC 停止					

数据类型	BOOL	WORD	DWORD	LWORD	UINT	UDINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S														

装置	X	Y	M	S	T	C	HC/AC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S																	

脉冲执行型	16 位指令	32 位指令
AH 运动控制主机	AH 运动控制主机	-

图形：



S : SFC 名称

指令说明：

1. 停止 S 所指定的 SFC 程序。
2. 指令执行时，S 所指定的 SFC 程序并不会立即停止，要等到扫描到 SFC 程序时才会停止。
3. S 指定的 SFC 程序，正在执行中的 ACTION，程序中的输出会被清除（执行 Final Scan）。
4. 当 SFC 程序状态为 STOP，执行此指令无效。

程序范例：

请参考 SFCRUN 程序范例。

补充说明：

有关 SFC 的说明，请参考 ISPSOFT 使用手册。

MEMO



附录A

目录

A.1	特殊辅助继电器 (SM)	A-3
A.1.1	特殊辅助继电器 SM 的更新时间	A-34
A.2	特殊数据寄存器 (SR)	A-38
A.2.1	特殊数据寄存器 SR 的更新时间	A-61
A.3	SM/SR 补充说明	A-64
A.4	标准指令 (依英文字母排序)	A-77
A.5	错误码与故障排除	A-86
A.5.1	错误码与对应灯号状态说明	A-86
AHxxEMC-5A	A-87	
对应的错误信息表	A-87	
错误码和灯号指示说明	A-88	
模拟 I/O 模块及温度模块	A-100	
AH02HC-5A/AH04HC-5A	A-101	
AH05PM-5A/AH10PM-5A/AH15PM-5A	A-102	
AH20MC-5A	A-103	
AH10EN-5A/AH15EN-5A	A-104	
AH10SCM-5A/AH15SCM-5A	A-104	
AH10DNET-5A	A-104	
AH10PFBM-5A	A-105	
AH10PFBS-5A	A-106	
AH10COPM-5A	A-107	
A.5.2	错误码与故障排除	A-108
AHxxEMC-5A	A-108	
ERROR 灯号常亮	A-108	
ERROR 灯号闪烁	A-109	
BUS FAULT 灯号常亮	A-126	

BUS FAULT 灯号闪烁.....	A-127
其它	A-127
模拟 I/O 模块及温度模块故障排除.....	A-138
AH02HC-5A/AH04HC-5A	A-140
AH05PM-5A/AH10PM-5A/AH15PM-5A	A-141
AH20MC-5A.....	A-142
AH10EN-5A/AH15EN-5A.....	A-144
AH10SCM-5A/AH15SCM-5A	A-144
AH10DNET-5A	A-144
AH10PFBM-5A	A-145
AH10PFBS-5A	A-146
AH10COPM-5A.....	A-146
A.5.3 极限错误排除.....	A-147
软极限错误排除	A-147
硬极限错误排除	A-148
A.6 数据类型：列举 (Enum)	A-149

A.1 特殊辅助继电器 (SM)

每一个特殊辅助继电器都有其特定的功用，未定义的特殊标志请勿使用。

特殊辅助继电器 (SM) 的列表及功能如下所示。编号前有 “*” 记号的 SM 装置，可参考 **A.3 SM/SR 补充说明**。其中属性栏中标示为 “R” 者，表示仅可作读取的动作，若标示为 “R/W”，表示可作读写的动作。另若标示为 “-”，表示无变化。标示为 “#”，则表示系统会依照 PLC 状态作设定，用户可读取该设定值对照手册的说明，进一步了解系统信息。

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM0	运算错误	○	○	OFF	OFF	-	R	OFF
SM1	运算错误锁定	○	○	OFF	OFF	-	R	OFF
SM5	指令/操作数检查错误	○	○	OFF	OFF	-	R	OFF
SM7	24V 电源供应不足	○	○	OFF	OFF	OFF	R	OFF
*SM8	逾时监视错误	○	○	OFF	-	-	R	OFF
SM9	系统错误	○	○	OFF	-	-	R	OFF
SM10	I/O 总线错误	○	○	OFF	-	-	R	OFF
*SM22	清除错误纪录	○	○	OFF	-	-	R/W	OFF
SM23	清除下载记录	○	○	OFF	-	-	R/W	OFF
SM24	清除 PLC 状态变更纪录	○	○	OFF	-	-	R/W	OFF
SM25	当在线编辑模式启动时，在线编辑处理标志开启	○	○	OFF	-	-	R	OFF
SM26	当除错模式启动时，除错模式处理标志开启	○	○	OFF	-	-	R	OFF
*SM96	COM1 送信标志	×	×	OFF	OFF	-	R/W	OFF
*SM97	COM2 送信标志	○	×	OFF	OFF	-	R/W	OFF
*SM98	COM1 接收等待	×	×	OFF	OFF	-	R	OFF
*SM99	COM2 接收等待	○	×	OFF	OFF	-	R	OFF
*SM100	COM1 接收完毕	×	×	OFF	OFF	-	R/W	OFF
*SM101	COM2 接收完毕	○	×	OFF	OFF	-	R/W	OFF
*SM102	COM1 MODRW 或 RS 数据接收错误	×	×	OFF	OFF	-	R	OFF
*SM103	COM2 MODRW 或 RS 数据接收错误	○	×	OFF	OFF	-	R	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM104	COM1 接收超时	×	×	OFF	OFF	—	R/W	OFF
*SM105	COM2 接收超时	○	×	OFF	OFF	—	R/W	OFF
*SM106	COM1 8/16 位处理模式选择 · ON : 8 位 · OFF : 16 位	×	×	OFF	—	—	R/W	OFF
*SM107	COM2 8/16 位处理模式选择 · ON : 8 位 · OFF : 16 位	○	×	OFF	—	—	R/W	OFF
*SM109	COM2 接收结束标志	○	×	—	—	—	R/W	OFF
*SM204	非停电保持区域全部清除	○	○	OFF	—	—	R/W	OFF
*SM205	停电保持区域全部清除	○	○	OFF	—	—	R/W	OFF
SM206	输出全部禁止	○	○	OFF	—	—	R/W	OFF
*SM209	COM1 通讯协议变更用	○	○	OFF	—	—	R/W	OFF
*SM210	COM1 ASCII/RTU 模式选择 · ON 时为 RTU 模式	○	○	OFF	—	—	R/W	OFF
*SM211	COM2 通讯协议变更用	○	×	OFF	—	—	R/W	OFF
*SM212	COM2 ASCII/RTU 模式选择 · ON 时为 RTU 模式	○	×	OFF	—	—	R/W	OFF
SM215	运行状态	○	○	OFF	ON	OFF	R/W	OFF
SM220	万年历±30 秒校正	○	○	OFF	OFF	—	R/W	OFF
*SM400	常开接点	○	○	ON	ON	ON	R	ON
*SM401	常闭接点	○	○	OFF	OFF	OFF	R	OFF
*SM402	开始正向 (RUN 的瞬间 “ON”) 脉冲	○	○	OFF	ON	OFF	R	OFF
*SM403	开始负向 (RUN 的瞬间 “OFF”) 脉冲	○	○	ON	OFF	ON	R	ON
*SM404	10ms 时钟脉冲 · 5ms ON/5ms OFF	○	○	OFF	—	—	R	OFF
*SM405	100ms 时钟脉冲 · 50ms ON/50ms OFF	○	○	OFF	—	—	R	OFF
*SM406	200ms 时钟脉冲 · 100ms ON/100ms OFF	○	○	OFF	—	—	R	OFF
*SM407	1s 时钟脉冲 · 0.5s ON/0.5s OFF	○	○	OFF	—	—	R	OFF
*SM408	2s 时钟脉冲 · 1s ON/1s OFF	○	○	OFF	—	—	R	OFF
*SM409	2n 秒时钟脉冲 · n (秒) 开/n (秒) 关 · n 的时间间隔指定于 SR409	○	○	OFF	—	—	R	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM410	2n 毫秒时钟脉冲 · n (毫秒) 开/n (毫秒) 关 · n 的时间间隔指定于 SR410	○	○	OFF	-	-	R	OFF
*SM450	储存卡是否存在标志 · ON : 存在/OFF : 不存在	○	○	OFF	-	-	R	OFF
*SM451	储存卡写保护开关 (Protect Switch) · ON : 写保护/OFF : 无写保护	○	○	OFF	-	-	R	OFF
*SM452	储存卡正被存取 (Accessed) 中 · ON : 存取中/OFF : 无存取	○	○	OFF	-	-	R	OFF
*SM453	储存卡运行中有错误发生 · ON-表错误发生	○	○	OFF	-	-	R	OFF
SM600	零标号 (Zero flag)	○	○	OFF	-	-	R	OFF
SM601	借位标号 (Borrow flag)	○	○	OFF	-	-	R	OFF
SM602	进位标号 (Carry flag)	○	○	OFF	-	-	R	OFF
SM603	SORT 排序指令执行完毕	○	○	OFF	-	-	R	OFF
SM604	SORT 排序指令工作模式设定 (ON-降序排序 · OFF-升序排序)	○	○	OFF	-	-	R/W	OFF
SM605	SMOV 工作模式指定	○	○	OFF	-	-	R/W	OFF
SM606	8/16 位工作模式	○	○	OFF	-	-	R/W	OFF
SM607	矩阵比较标志 · 比较相同值 (SM607=1) 或不同值 (SM607=0)	○	○	OFF	-	-	R/W	OFF
SM608	矩阵搜寻结束标志 · 当比较到最后一个 bit 时 · SM608=1	○	○	OFF	-	-	R	OFF
SM609	矩阵搜寻起始标志 · 当 SM609=1 时由第 0 个 bit 开始比较	○	○	OFF	-	-	R	OFF
SM610	矩阵位寻找标志 · 比较到达时立即停止比较动作 · SM610=1	○	○	OFF	-	-	R	OFF
SM611	矩阵指针错误标志 · 指针 Pr 值超出范围则 SM611=1	○	○	OFF	-	-	R	OFF
SM612	矩阵指针递增标志 · 将指针目前值+1	○	○	OFF	-	-	R/W	OFF
SM613	矩阵指针清除标志 · 将指针目前值清除为 0	○	○	OFF	-	-	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM614	矩阵循环位移输出进位标志	○	○	OFF	-	-	R	OFF
SM615	矩阵位移输入补位标志	○	○	OFF	-	-	R/W	OFF
SM616	矩阵循环位移方向标志 SM616=0 : 左移 · SM616=1 : 右移	○	○	OFF	-	-	R/W	OFF
SM617	矩阵计数字元为 0 或位为 1 标志	○	○	OFF	-	-	R/W	OFF
SM618	矩阵计数结果为 0 时 ON	○	○	OFF	-	-	R/W	OFF
SM619	当 EI 指令被执行时 ON	○	○	OFF	OFF	-	R	OFF
SM620	表格比较指令 CMPT#全部输出标志 · 若比较结果 为全都输出则 SM620=ON	○	○	OFF	-	-	R	OFF
SM621	HC0 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM622	HC1 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM623	HC2 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM624	HC3 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM625	HC4 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM626	HC5 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM627	HC6 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM628	HC7 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM629	HC8 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM630	HC9 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM631	HC10 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM632	HC11 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM633	HC12 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM634	HC13 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM635	HC14 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM636	HC15 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM637	HC16 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF
SM638	HC17 计数模式设定 (ON 时为下数)	○	○	OFF	-	-	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM639	HC18 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM640	HC19 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM641	HC20 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM642	HC21 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM643	HC22 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM644	HC23 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM645	HC24 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM646	HC25 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM647	HC26 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM648	HC27 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM649	HC28 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM650	HC29 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM651	HC30 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM652	HC31 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM653	HC32 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM654	HC33 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM655	HC34 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM656	HC35 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM657	HC36 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM658	HC37 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM659	HC38 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM660	HC39 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM661	HC40 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM662	HC41 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM663	HC42 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM664	HC43 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM665	HC44 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM666	HC45 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM667	HC46 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM668	HC47 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM669	HC48 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM670	HC49 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM671	HC50 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM672	HC51 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM673	HC52 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM674	HC53 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM675	HC54 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM676	HC55 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM677	HC56 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM678	HC57 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM679	HC58 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM680	HC59 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM681	HC60 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM682	HC61 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM683	HC62 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM684	HC63 计数模式设定 (ON 时为下数)	○	○	OFF	—	—	R/W	OFF
SM685	DSCLP 使用浮点数运算	○	○	OFF	—	—	R/W	OFF
SM686	RAMP 指令连续执行设定	○	○	OFF	—	—	R/W	OFF
SM687	RAMP 指令执行完毕	○	○	OFF	—	—	R/W	OFF
SM688	INCD 指令执行完毕	○	○	OFF	—	—	R/W	OFF
SM690	字符串控制模式	○	○	OFF	—	—	R/W	OFF
SM691	HKY 输入为 16 位模式	○	○	OFF	—	—	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	ON : 十六进制输入 · OFF : A~F 为功能键							
SM692	HKY 指令执行完毕标志 (ON 一个扫描周期)	○	○	OFF	-	-	R/W	OFF
SM693	SEGL 指令执行完毕标志 (ON 一个扫描周期)	○	○	OFF	-	-	R/W	OFF
SM694	DSW 指令执行完毕标志 (ON 一个扫描周期)	○	○	OFF	-	-	R/W	OFF
SM695	弧度/角度使用标志 · ON 的时候表示角度	○	○	OFF	-	-	R/W	OFF
SM700	MODBUS TCP 联机 1 启动标志	○	○	OFF	-	-	R/W	OFF
SM701	MODBUS TCP 联机 2 启动标志	○	○	OFF	-	-	R/W	OFF
SM702	MODBUS TCP 联机 3 启动标志	○	○	OFF	-	-	R/W	OFF
SM703	MODBUS TCP 联机 4 启动标志	○	○	OFF	-	-	R/W	OFF
SM704	MODBUS TCP 联机 5 启动标志	○	○	OFF	-	-	R/W	OFF
SM705	MODBUS TCP 联机 6 启动标志	○	○	OFF	-	-	R/W	OFF
SM706	MODBUS TCP 联机 7 启动标志	○	○	OFF	-	-	R/W	OFF
SM707	MODBUS TCP 联机 8 启动标志	○	○	OFF	-	-	R/W	OFF
SM708	MODBUS TCP 联机 9 启动标志	○	○	OFF	-	-	R/W	OFF
SM709	MODBUS TCP 联机 10 启动标志	○	○	OFF	-	-	R/W	OFF
SM710	MODBUS TCP 联机 11 启动标志	○	○	OFF	-	-	R/W	OFF
SM711	MODBUS TCP 联机 12 启动标志	○	○	OFF	-	-	R/W	OFF
SM712	MODBUS TCP 联机 13 启动标志	○	○	OFF	-	-	R/W	OFF
SM713	MODBUS TCP 联机 14 启动标志	○	○	OFF	-	-	R/W	OFF
SM714	MODBUS TCP 联机 15 启动标志	○	○	OFF	-	-	R/W	OFF
SM715	MODBUS TCP 联机 16 启动标志	○	○	OFF	-	-	R/W	OFF
SM716	MODBUS TCP 联机 17 启动标志	○	○	OFF	-	-	R/W	OFF
SM717	MODBUS TCP 联机 18 启动标志	○	○	OFF	-	-	R/W	OFF
SM718	MODBUS TCP 联机 19 启动标志	○	○	OFF	-	-	R/W	OFF
SM719	MODBUS TCP 联机 20 启动标志	○	○	OFF	-	-	R/W	OFF
SM720	MODBUS TCP 联机 21 启动标志	○	○	OFF	-	-	R/W	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM721	MODBUS TCP 联机 22 启动标志	○	○	OFF	—	—	R/W	OFF
SM722	MODBUS TCP 联机 23 启动标志	○	○	OFF	—	—	R/W	OFF
SM723	MODBUS TCP 联机 24 启动标志	○	○	OFF	—	—	R/W	OFF
SM724	MODBUS TCP 联机 25 启动标志	○	○	OFF	—	—	R/W	OFF
SM725	MODBUS TCP 联机 26 启动标志	○	○	OFF	—	—	R/W	OFF
SM726	MODBUS TCP 联机 27 启动标志	○	○	OFF	—	—	R/W	OFF
SM727	MODBUS TCP 联机 28 启动标志	○	○	OFF	—	—	R/W	OFF
SM728	MODBUS TCP 联机 29 启动标志	○	○	OFF	—	—	R/W	OFF
SM729	MODBUS TCP 联机 30 启动标志	○	○	OFF	—	—	R/W	OFF
SM730	MODBUS TCP 联机 31 启动标志	○	○	OFF	—	—	R/W	OFF
SM731	MODBUS TCP 联机 32 启动标志	○	○	OFF	—	—	R/W	OFF
SM732	MODBUS TCP 联机 33 启动标志	○	○	OFF	—	—	R/W	OFF
SM733	MODBUS TCP 联机 34 启动标志	○	○	OFF	—	—	R/W	OFF
SM734	MODBUS TCP 联机 35 启动标志	○	○	OFF	—	—	R/W	OFF
SM735	MODBUS TCP 联机 36 启动标志	○	○	OFF	—	—	R/W	OFF
SM736	MODBUS TCP 联机 37 启动标志	○	○	OFF	—	—	R/W	OFF
SM737	MODBUS TCP 联机 38 启动标志	○	○	OFF	—	—	R/W	OFF
SM738	MODBUS TCP 联机 39 启动标志	○	○	OFF	—	—	R/W	OFF
SM739	MODBUS TCP 联机 40 启动标志	○	○	OFF	—	—	R/W	OFF
SM740	MODBUS TCP 联机 41 启动标志	○	○	OFF	—	—	R/W	OFF
SM741	MODBUS TCP 联机 42 启动标志	○	○	OFF	—	—	R/W	OFF
SM742	MODBUS TCP 联机 43 启动标志	○	○	OFF	—	—	R/W	OFF
SM743	MODBUS TCP 联机 44 启动标志	○	○	OFF	—	—	R/W	OFF
SM744	MODBUS TCP 联机 45 启动标志	○	○	OFF	—	—	R/W	OFF
SM745	MODBUS TCP 联机 46 启动标志	○	○	OFF	—	—	R/W	OFF
SM746	MODBUS TCP 联机 47 启动标志	○	○	OFF	—	—	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM747	MODBUS TCP 联机 48 启动标志	○	○	OFF	-	-	R/W	OFF
SM748	MODBUS TCP 联机 49 启动标志	○	○	OFF	-	-	R/W	OFF
SM749	MODBUS TCP 联机 50 启动标志	○	○	OFF	-	-	R/W	OFF
SM750	MODBUS TCP 联机 51 启动标志	○	○	OFF	-	-	R/W	OFF
SM751	MODBUS TCP 联机 52 启动标志	○	○	OFF	-	-	R/W	OFF
SM752	MODBUS TCP 联机 53 启动标志	○	○	OFF	-	-	R/W	OFF
SM753	MODBUS TCP 联机 54 启动标志	○	○	OFF	-	-	R/W	OFF
SM754	MODBUS TCP 联机 55 启动标志	○	○	OFF	-	-	R/W	OFF
SM755	MODBUS TCP 联机 56 启动标志	○	○	OFF	-	-	R/W	OFF
SM756	MODBUS TCP 联机 57 启动标志	○	○	OFF	-	-	R/W	OFF
SM757	MODBUS TCP 联机 58 启动标志	○	○	OFF	-	-	R/W	OFF
SM758	MODBUS TCP 联机 59 启动标志	○	○	OFF	-	-	R/W	OFF
SM759	MODBUS TCP 联机 60 启动标志	○	○	OFF	-	-	R/W	OFF
SM760	MODBUS TCP 联机 61 启动标志	○	○	OFF	-	-	R/W	OFF
SM761	MODBUS TCP 联机 62 启动标志	○	○	OFF	-	-	R/W	OFF
SM762	MODBUS TCP 联机 63 启动标志	○	○	OFF	-	-	R/W	OFF
SM763	MODBUS TCP 联机 64 启动标志	○	○	OFF	-	-	R/W	OFF
SM764	MODBUS TCP 联机 65 启动标志	○	○	OFF	-	-	R/W	OFF
SM765	MODBUS TCP 联机 66 启动标志	○	○	OFF	-	-	R/W	OFF
SM766	MODBUS TCP 联机 67 启动标志	○	○	OFF	-	-	R/W	OFF
SM767	MODBUS TCP 联机 68 启动标志	○	○	OFF	-	-	R/W	OFF
SM768	MODBUS TCP 联机 69 启动标志	○	○	OFF	-	-	R/W	OFF
SM769	MODBUS TCP 联机 70 启动标志	○	○	OFF	-	-	R/W	OFF
SM770	MODBUS TCP 联机 71 启动标志	○	○	OFF	-	-	R/W	OFF
SM771	MODBUS TCP 联机 72 启动标志	○	○	OFF	-	-	R/W	OFF
SM772	MODBUS TCP 联机 73 启动标志	○	○	OFF	-	-	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM773	MODBUS TCP 联机 74 启动标志	○	○	OFF	-	-	R/W	OFF
SM774	MODBUS TCP 联机 75 启动标志	○	○	OFF	-	-	R/W	OFF
SM775	MODBUS TCP 联机 76 启动标志	○	○	OFF	-	-	R/W	OFF
SM776	MODBUS TCP 联机 77 启动标志	○	○	OFF	-	-	R/W	OFF
SM777	MODBUS TCP 联机 78 启动标志	○	○	OFF	-	-	R/W	OFF
SM778	MODBUS TCP 联机 79 启动标志	○	○	OFF	-	-	R/W	OFF
SM779	MODBUS TCP 联机 80 启动标志	○	○	OFF	-	-	R/W	OFF
SM780	MODBUS TCP 联机 81 启动标志	○	○	OFF	-	-	R/W	OFF
SM781	MODBUS TCP 联机 82 启动标志	○	○	OFF	-	-	R/W	OFF
SM782	MODBUS TCP 联机 83 启动标志	○	○	OFF	-	-	R/W	OFF
SM783	MODBUS TCP 联机 84 启动标志	○	○	OFF	-	-	R/W	OFF
SM784	MODBUS TCP 联机 85 启动标志	○	○	OFF	-	-	R/W	OFF
SM785	MODBUS TCP 联机 86 启动标志	○	○	OFF	-	-	R/W	OFF
SM786	MODBUS TCP 联机 87 启动标志	○	○	OFF	-	-	R/W	OFF
SM787	MODBUS TCP 联机 88 启动标志	○	○	OFF	-	-	R/W	OFF
SM788	MODBUS TCP 联机 89 启动标志	○	○	OFF	-	-	R/W	OFF
SM789	MODBUS TCP 联机 90 启动标志	○	○	OFF	-	-	R/W	OFF
SM790	MODBUS TCP 联机 91 启动标志	○	○	OFF	-	-	R/W	OFF
SM791	MODBUS TCP 联机 92 启动标志	○	○	OFF	-	-	R/W	OFF
SM792	MODBUS TCP 联机 93 启动标志	○	○	OFF	-	-	R/W	OFF
SM793	MODBUS TCP 联机 94 启动标志	○	○	OFF	-	-	R/W	OFF
SM794	MODBUS TCP 联机 95 启动标志	○	○	OFF	-	-	R/W	OFF
SM795	MODBUS TCP 联机 96 启动标志	○	○	OFF	-	-	R/W	OFF
SM796	MODBUS TCP 联机 97 启动标志	○	○	OFF	-	-	R/W	OFF
SM797	MODBUS TCP 联机 98 启动标志	○	○	OFF	-	-	R/W	OFF
SM798	MODBUS TCP 联机 99 启动标志	○	○	OFF	-	-	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM799	MODBUS TCP 联机 100 启动标志	○	○	OFF	—	—	R/W	OFF
SM800	MODBUS TCP 联机 101 启动标志	○	○	OFF	—	—	R/W	OFF
SM801	MODBUS TCP 联机 102 启动标志	○	○	OFF	—	—	R/W	OFF
SM802	MODBUS TCP 联机 103 启动标志	○	○	OFF	—	—	R/W	OFF
SM803	MODBUS TCP 联机 104 启动标志	○	○	OFF	—	—	R/W	OFF
SM804	MODBUS TCP 联机 105 启动标志	○	○	OFF	—	—	R/W	OFF
SM805	MODBUS TCP 联机 106 启动标志	○	○	OFF	—	—	R/W	OFF
SM806	MODBUS TCP 联机 107 启动标志	○	○	OFF	—	—	R/W	OFF
SM807	MODBUS TCP 联机 108 启动标志	○	○	OFF	—	—	R/W	OFF
SM808	MODBUS TCP 联机 109 启动标志	○	○	OFF	—	—	R/W	OFF
SM809	MODBUS TCP 联机 110 启动标志	○	○	OFF	—	—	R/W	OFF
SM810	MODBUS TCP 联机 111 启动标志	○	○	OFF	—	—	R/W	OFF
SM811	MODBUS TCP 联机 112 启动标志	○	○	OFF	—	—	R/W	OFF
SM812	MODBUS TCP 联机 113 启动标志	○	○	OFF	—	—	R/W	OFF
SM813	MODBUS TCP 联机 114 启动标志	○	○	OFF	—	—	R/W	OFF
SM814	MODBUS TCP 联机 115 启动标志	○	○	OFF	—	—	R/W	OFF
SM815	MODBUS TCP 联机 116 启动标志	○	○	OFF	—	—	R/W	OFF
SM816	MODBUS TCP 联机 117 启动标志	○	○	OFF	—	—	R/W	OFF
SM817	MODBUS TCP 联机 118 启动标志	○	○	OFF	—	—	R/W	OFF
SM818	MODBUS TCP 联机 119 启动标志	○	○	OFF	—	—	R/W	OFF
SM819	MODBUS TCP 联机 120 启动标志	○	○	OFF	—	—	R/W	OFF
SM820	MODBUS TCP 联机 121 启动标志	○	○	OFF	—	—	R/W	OFF
SM821	MODBUS TCP 联机 122 启动标志	○	○	OFF	—	—	R/W	OFF
SM822	MODBUS TCP 联机 123 启动标志	○	○	OFF	—	—	R/W	OFF
SM823	MODBUS TCP 联机 124 启动标志	○	○	OFF	—	—	R/W	OFF
SM824	MODBUS TCP 联机 125 启动标志	○	○	OFF	—	—	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM825	MODBUS TCP 联机 126 启动标志	○	○	OFF	—	—	R/W	OFF
SM826	MODBUS TCP 联机 127 启动标志	○	○	OFF	—	—	R/W	OFF
SM827	MODBUS TCP 联机 128 启动标志	○	○	OFF	—	—	R/W	OFF
SM828	MODBUS TCP 联机 1 错误标志	○	○	OFF	—	—	R/W	OFF
SM829	MODBUS TCP 联机 2 错误标志	○	○	OFF	—	—	R/W	OFF
SM830	MODBUS TCP 联机 3 错误标志	○	○	OFF	—	—	R/W	OFF
SM831	MODBUS TCP 联机 4 错误标志	○	○	OFF	—	—	R/W	OFF
SM832	MODBUS TCP 联机 5 错误标志	○	○	OFF	—	—	R/W	OFF
SM833	MODBUS TCP 联机 6 错误标志	○	○	OFF	—	—	R/W	OFF
SM834	MODBUS TCP 联机 7 错误标志	○	○	OFF	—	—	R/W	OFF
SM835	MODBUS TCP 联机 8 错误标志	○	○	OFF	—	—	R/W	OFF
SM836	MODBUS TCP 联机 9 错误标志	○	○	OFF	—	—	R/W	OFF
SM837	MODBUS TCP 联机 10 错误标志	○	○	OFF	—	—	R/W	OFF
SM838	MODBUS TCP 联机 11 错误标志	○	○	OFF	—	—	R/W	OFF
SM839	MODBUS TCP 联机 12 错误标志	○	○	OFF	—	—	R/W	OFF
SM840	MODBUS TCP 联机 13 错误标志	○	○	OFF	—	—	R/W	OFF
SM841	MODBUS TCP 联机 14 错误标志	○	○	OFF	—	—	R/W	OFF
SM842	MODBUS TCP 联机 15 错误标志	○	○	OFF	—	—	R/W	OFF
SM843	MODBUS TCP 联机 16 错误标志	○	○	OFF	—	—	R/W	OFF
SM844	MODBUS TCP 联机 17 错误标志	○	○	OFF	—	—	R/W	OFF
SM845	MODBUS TCP 联机 18 错误标志	○	○	OFF	—	—	R/W	OFF
SM846	MODBUS TCP 联机 19 错误标志	○	○	OFF	—	—	R/W	OFF
SM847	MODBUS TCP 联机 20 错误标志	○	○	OFF	—	—	R/W	OFF
SM848	MODBUS TCP 联机 21 错误标志	○	○	OFF	—	—	R/W	OFF
SM849	MODBUS TCP 联机 22 错误标志	○	○	OFF	—	—	R/W	OFF
SM850	MODBUS TCP 联机 23 错误标志	○	○	OFF	—	—	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM851	MODBUS TCP 联机 24 错误标志	○	○	OFF	—	—	R/W	OFF
SM852	MODBUS TCP 联机 25 错误标志	○	○	OFF	—	—	R/W	OFF
SM853	MODBUS TCP 联机 26 错误标志	○	○	OFF	—	—	R/W	OFF
SM854	MODBUS TCP 联机 27 错误标志	○	○	OFF	—	—	R/W	OFF
SM855	MODBUS TCP 联机 28 错误标志	○	○	OFF	—	—	R/W	OFF
SM856	MODBUS TCP 联机 29 错误标志	○	○	OFF	—	—	R/W	OFF
SM857	MODBUS TCP 联机 30 错误标志	○	○	OFF	—	—	R/W	OFF
SM858	MODBUS TCP 联机 31 错误标志	○	○	OFF	—	—	R/W	OFF
SM859	MODBUS TCP 联机 32 错误标志	○	○	OFF	—	—	R/W	OFF
SM860	MODBUS TCP 联机 33 错误标志	○	○	OFF	—	—	R/W	OFF
SM861	MODBUS TCP 联机 34 错误标志	○	○	OFF	—	—	R/W	OFF
SM862	MODBUS TCP 联机 35 错误标志	○	○	OFF	—	—	R/W	OFF
SM863	MODBUS TCP 联机 36 错误标志	○	○	OFF	—	—	R/W	OFF
SM864	MODBUS TCP 联机 37 错误标志	○	○	OFF	—	—	R/W	OFF
SM865	MODBUS TCP 联机 38 错误标志	○	○	OFF	—	—	R/W	OFF
SM866	MODBUS TCP 联机 39 错误标志	○	○	OFF	—	—	R/W	OFF
SM867	MODBUS TCP 联机 40 错误标志	○	○	OFF	—	—	R/W	OFF
SM868	MODBUS TCP 联机 41 错误标志	○	○	OFF	—	—	R/W	OFF
SM869	MODBUS TCP 联机 42 错误标志	○	○	OFF	—	—	R/W	OFF
SM870	MODBUS TCP 联机 43 错误标志	○	○	OFF	—	—	R/W	OFF
SM871	MODBUS TCP 联机 44 错误标志	○	○	OFF	—	—	R/W	OFF
SM872	MODBUS TCP 联机 45 错误标志	○	○	OFF	—	—	R/W	OFF
SM873	MODBUS TCP 联机 46 错误标志	○	○	OFF	—	—	R/W	OFF
SM874	MODBUS TCP 联机 47 错误标志	○	○	OFF	—	—	R/W	OFF
SM875	MODBUS TCP 联机 48 错误标志	○	○	OFF	—	—	R/W	OFF
SM876	MODBUS TCP 联机 49 错误标志	○	○	OFF	—	—	R/W	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM877	MODBUS TCP 联机 50 错误标志	○	○	OFF	—	—	R/W	OFF
SM878	MODBUS TCP 联机 51 错误标志	○	○	OFF	—	—	R/W	OFF
SM879	MODBUS TCP 联机 52 错误标志	○	○	OFF	—	—	R/W	OFF
SM880	MODBUS TCP 联机 53 错误标志	○	○	OFF	—	—	R/W	OFF
SM881	MODBUS TCP 联机 54 错误标志	○	○	OFF	—	—	R/W	OFF
SM882	MODBUS TCP 联机 55 错误标志	○	○	OFF	—	—	R/W	OFF
SM883	MODBUS TCP 联机 56 错误标志	○	○	OFF	—	—	R/W	OFF
SM884	MODBUS TCP 联机 57 错误标志	○	○	OFF	—	—	R/W	OFF
SM885	MODBUS TCP 联机 58 错误标志	○	○	OFF	—	—	R/W	OFF
SM886	MODBUS TCP 联机 59 错误标志	○	○	OFF	—	—	R/W	OFF
SM887	MODBUS TCP 联机 60 错误标志	○	○	OFF	—	—	R/W	OFF
SM888	MODBUS TCP 联机 61 错误标志	○	○	OFF	—	—	R/W	OFF
SM889	MODBUS TCP 联机 62 错误标志	○	○	OFF	—	—	R/W	OFF
SM890	MODBUS TCP 联机 63 错误标志	○	○	OFF	—	—	R/W	OFF
SM891	MODBUS TCP 联机 64 错误标志	○	○	OFF	—	—	R/W	OFF
SM892	MODBUS TCP 联机 65 错误标志	○	○	OFF	—	—	R/W	OFF
SM893	MODBUS TCP 联机 66 错误标志	○	○	OFF	—	—	R/W	OFF
SM894	MODBUS TCP 联机 67 错误标志	○	○	OFF	—	—	R/W	OFF
SM895	MODBUS TCP 联机 68 错误标志	○	○	OFF	—	—	R/W	OFF
SM896	MODBUS TCP 联机 69 错误标志	○	○	OFF	—	—	R/W	OFF
SM897	MODBUS TCP 联机 70 错误标志	○	○	OFF	—	—	R/W	OFF
SM898	MODBUS TCP 联机 71 错误标志	○	○	OFF	—	—	R/W	OFF
SM899	MODBUS TCP 联机 72 错误标志	○	○	OFF	—	—	R/W	OFF
SM900	MODBUS TCP 联机 73 错误标志	○	○	OFF	—	—	R/W	OFF
SM901	MODBUS TCP 联机 74 错误标志	○	○	OFF	—	—	R/W	OFF
SM902	MODBUS TCP 联机 75 错误标志	○	○	OFF	—	—	R/W	OFF

A1

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM903	MODBUS TCP 联机 76 错误标志	○	○	OFF	—	—	R/W	OFF
SM904	MODBUS TCP 联机 77 错误标志	○	○	OFF	—	—	R/W	OFF
SM905	MODBUS TCP 联机 78 错误标志	○	○	OFF	—	—	R/W	OFF
SM906	MODBUS TCP 联机 79 错误标志	○	○	OFF	—	—	R/W	OFF
SM907	MODBUS TCP 联机 80 错误标志	○	○	OFF	—	—	R/W	OFF
SM908	MODBUS TCP 联机 81 错误标志	○	○	OFF	—	—	R/W	OFF
SM909	MODBUS TCP 联机 82 错误标志	○	○	OFF	—	—	R/W	OFF
SM910	MODBUS TCP 联机 83 错误标志	○	○	OFF	—	—	R/W	OFF
SM911	MODBUS TCP 联机 84 错误标志	○	○	OFF	—	—	R/W	OFF
SM912	MODBUS TCP 联机 85 错误标志	○	○	OFF	—	—	R/W	OFF
SM913	MODBUS TCP 联机 86 错误标志	○	○	OFF	—	—	R/W	OFF
SM914	MODBUS TCP 联机 87 错误标志	○	○	OFF	—	—	R/W	OFF
SM915	MODBUS TCP 联机 88 错误标志	○	○	OFF	—	—	R/W	OFF
SM916	MODBUS TCP 联机 89 错误标志	○	○	OFF	—	—	R/W	OFF
SM917	MODBUS TCP 联机 90 错误标志	○	○	OFF	—	—	R/W	OFF
SM918	MODBUS TCP 联机 91 错误标志	○	○	OFF	—	—	R/W	OFF
SM919	MODBUS TCP 联机 92 错误标志	○	○	OFF	—	—	R/W	OFF
SM920	MODBUS TCP 联机 93 错误标志	○	○	OFF	—	—	R/W	OFF
SM921	MODBUS TCP 联机 94 错误标志	○	○	OFF	—	—	R/W	OFF
SM922	MODBUS TCP 联机 95 错误标志	○	○	OFF	—	—	R/W	OFF
SM923	MODBUS TCP 联机 96 错误标志	○	○	OFF	—	—	R/W	OFF
SM924	MODBUS TCP 联机 97 错误标志	○	○	OFF	—	—	R/W	OFF
SM925	MODBUS TCP 联机 98 错误标志	○	○	OFF	—	—	R/W	OFF
SM926	MODBUS TCP 联机 99 错误标志	○	○	OFF	—	—	R/W	OFF
SM927	MODBUS TCP 联机 100 错误标志	○	○	OFF	—	—	R/W	OFF
SM928	MODBUS TCP 联机 101 错误标志	○	○	OFF	—	—	R/W	OFF

A1

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM929	MODBUS TCP 联机 102 错误标志	○	○	OFF	-	-	R/W	OFF
SM930	MODBUS TCP 联机 103 错误标志	○	○	OFF	-	-	R/W	OFF
SM931	MODBUS TCP 联机 104 错误标志	○	○	OFF	-	-	R/W	OFF
SM932	MODBUS TCP 联机 105 错误标志	○	○	OFF	-	-	R/W	OFF
SM933	MODBUS TCP 联机 106 错误标志	○	○	OFF	-	-	R/W	OFF
SM934	MODBUS TCP 联机 107 错误标志	○	○	OFF	-	-	R/W	OFF
SM935	MODBUS TCP 联机 108 错误标志	○	○	OFF	-	-	R/W	OFF
SM936	MODBUS TCP 联机 109 错误标志	○	○	OFF	-	-	R/W	OFF
SM937	MODBUS TCP 联机 110 错误标志	○	○	OFF	-	-	R/W	OFF
SM938	MODBUS TCP 联机 111 错误标志	○	○	OFF	-	-	R/W	OFF
SM939	MODBUS TCP 联机 112 错误标志	○	○	OFF	-	-	R/W	OFF
SM940	MODBUS TCP 联机 113 错误标志	○	○	OFF	-	-	R/W	OFF
SM941	MODBUS TCP 联机 114 错误标志	○	○	OFF	-	-	R/W	OFF
SM942	MODBUS TCP 联机 115 错误标志	○	○	OFF	-	-	R/W	OFF
SM943	MODBUS TCP 联机 116 错误标志	○	○	OFF	-	-	R/W	OFF
SM944	MODBUS TCP 联机 117 错误标志	○	○	OFF	-	-	R/W	OFF
SM945	MODBUS TCP 联机 118 错误标志	○	○	OFF	-	-	R/W	OFF
SM946	MODBUS TCP 联机 119 错误标志	○	○	OFF	-	-	R/W	OFF
SM947	MODBUS TCP 联机 120 错误标志	○	○	OFF	-	-	R/W	OFF
SM948	MODBUS TCP 联机 121 错误标志	○	○	OFF	-	-	R/W	OFF
SM949	MODBUS TCP 联机 122 错误标志	○	○	OFF	-	-	R/W	OFF
SM950	MODBUS TCP 联机 123 错误标志	○	○	OFF	-	-	R/W	OFF
SM951	MODBUS TCP 联机 124 错误标志	○	○	OFF	-	-	R/W	OFF
SM952	MODBUS TCP 联机 125 错误标志	○	○	OFF	-	-	R/W	OFF
SM953	MODBUS TCP 联机 126 错误标志	○	○	OFF	-	-	R/W	OFF
SM954	MODBUS TCP 联机 127 错误标志	○	○	OFF	-	-	R/W	OFF

A1

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM955	MODBUS TCP 联机 128 错误标志	○	○	OFF	—	—	R/W	OFF
SM1000	Ethernet 设定标志。 ON 时，将 SR1000~SR1006 的数据写入闪存	○	○	-	—	—	R/W	OFF
SM1001	Ethernet 联机标志	○	○	OFF	—	—	R	OFF
SM1090	TCP 联机忙碌	○	○	OFF	—	—	R	OFF
SM1091	UDP 联机忙碌	x	x	OFF	—	—	R	OFF
SM1106	基本管理-以太网网络联机错误	○	○	OFF	—	—	R	OFF
SM1107	以太网网络基本管理-基本设定错误	○	○	OFF	—	—	R	OFF
SM1108	以太网网络基本管理-限定通讯对象设置错误	○	○	OFF	—	—	R	OFF
SM1109	TCP/UDP Socket 基本管理-本地端口已被使用	x	x	OFF	—	—	R	OFF
SM1196	Socket 配置错误	x	x	OFF	—	—	R/W	OFF
SM1270	TCP Socket 1--联机标志	○	○	OFF	OFF	—	R	OFF
SM1271	TCP Socket 1--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1272	TCP Socket 1--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1273	TCP Socket 1--开启标志	○	○	OFF	OFF	—	R	OFF
SM1274	TCP Socket 1—关闭标志	○	○	ON	ON	—	R	ON
SM1275	TCP Socket 1--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1276	TCP Socket 1--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1277	TCP Socket 1--错误标志	○	○	OFF	OFF	—	R	OFF
SM1278	TCP Socket 2--联机标志	○	○	OFF	OFF	—	R	OFF
SM1279	TCP Socket 2--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1280	TCP Socket 2--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1281	TCP Socket 2--开启标志	○	○	OFF	OFF	—	R	OFF
SM1282	TCP Socket 2—关闭标志	○	○	ON	ON	—	R	ON
SM1283	TCP Socket 2--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1284	TCP Socket 2--接收中标志	○	○	OFF	OFF	—	R	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1285	TCP Socket 2--错误标志	○	○	OFF	OFF	—	R	OFF
SM1286	TCP Socket 3--联机标志	○	○	OFF	OFF	—	R	OFF
SM1287	TCP Socket 3--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1288	TCP Socket 3--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1289	TCP Socket 3--开启标志	○	○	OFF	OFF	—	R	OFF
SM1290	TCP Socket 3--关闭标志	○	○	ON	ON	—	R	ON
SM1291	TCP Socket 3--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1292	TCP Socket 3--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1293	TCP Socket 3--错误标志	○	○	OFF	OFF	—	R	OFF
SM1294	TCP Socket 4--联机标志	○	○	OFF	OFF	—	R	OFF
SM1295	TCP Socket 4--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1296	TCP Socket 4--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1297	TCP Socket 4--开启标志	○	○	OFF	OFF	—	R	OFF
SM1298	TCP Socket 4--关闭标志	○	○	ON	ON	—	R	ON
SM1299	TCP Socket 4--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1300	TCP Socket 4--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1301	TCP Socket 4--错误标志	○	○	OFF	OFF	—	R	OFF
SM1302	TCP Socket 5--联机标志	○	○	OFF	OFF	—	R	OFF
SM1303	TCP Socket 5--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1304	TCP Socket 5--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1305	TCP Socket 5--开启标志	○	○	OFF	OFF	—	R	OFF
SM1306	TCP Socket 5--关闭标志	○	○	ON	ON	—	R	ON
SM1307	TCP Socket 5--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1308	TCP Socket 5--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1309	TCP Socket 5--错误标志	○	○	OFF	OFF	—	R	OFF
SM1310	TCP Socket 6--联机标志	○	○	OFF	OFF	—	R	OFF

A1

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1311	TCP Socket 6--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1312	TCP Socket 6--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1313	TCP Socket 6--开启标志	○	○	OFF	OFF	—	R	OFF
SM1314	TCP Socket 6—关闭标志	○	○	ON	ON	—	R	ON
SM1315	TCP Socket 6--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1316	TCP Socket 6--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1317	TCP Socket 6--错误标志	○	○	OFF	OFF	—	R	OFF
SM1318	TCP Socket 7--联机标志	○	○	OFF	OFF	—	R	OFF
SM1319	TCP Socket 7--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1320	TCP Socket 7--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1321	TCP Socket 7--开启标志	○	○	OFF	OFF	—	R	OFF
SM1322	TCP Socket 7--关闭标志	○	○	ON	ON	—	R	ON
SM1323	TCP Socket 7--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1324	TCP Socket 7--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1325	TCP Socket 7--错误标志	○	○	OFF	OFF	—	R	OFF
SM1326	TCP Socket 8--联机标志	○	○	OFF	OFF	—	R	OFF
SM1327	TCP Socket 8--数据接收完成标志	○	○	OFF	OFF	—	R	OFF
SM1328	TCP Socket 8--数据传送完成标志	○	○	OFF	OFF	—	R	OFF
SM1329	TCP Socket 8--开启标志	○	○	OFF	OFF	—	R	OFF
SM1330	TCP Socket 8--关闭标志	○	○	ON	ON	—	R	ON
SM1331	TCP Socket 8--传送中标志	○	○	OFF	OFF	—	R	OFF
SM1332	TCP Socket 8--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1333	TCP Socket 8--错误标志	○	○	OFF	OFF	—	R	OFF
SM1334	UDP Socket 1--联机已开启标志	○	○	OFF	OFF	—	R	OFF
SM1335	UDP Socket 1--数据已接收标志	○	○	OFF	OFF	—	R	OFF
SM1336	UDP Socket 1--数据已传送标志	○	○	OFF	OFF	—	R	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1337	UDP Socket 1--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1338	UDP Socket 1--错误标志	○	○	OFF	OFF	—	R	OFF
SM1339	UDP Socket 2--联机已开启标志	○	○	OFF	OFF	—	R	OFF
SM1340	UDP Socket 2--数据已接收标志	○	○	OFF	OFF	—	R	OFF
SM1341	UDP Socket 2--数据已传送标志	○	○	OFF	OFF	—	R	OFF
SM1342	UDP Socket 2--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1343	UDP Socket 2--错误标志	○	○	OFF	OFF	—	R	OFF
SM1344	UDP Socket 3--联机已开启标志	○	○	OFF	OFF	—	R	OFF
SM1345	UDP Socket 3--数据已接收标志	○	○	OFF	OFF	—	R	OFF
SM1346	UDP Socket 3--数据已传送标志	○	○	OFF	OFF	—	R	OFF
SM1347	UDP Socket 3--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1348	UDP Socket 3--错误标志	○	○	OFF	OFF	—	R	OFF
SM1349	UDP Socket 4--联机已开启标志	○	○	OFF	OFF	—	R	OFF
SM1350	UDP Socket 4--数据已接收标志	○	○	OFF	OFF	—	R	OFF
SM1351	UDP Socket 4--数据已传送标志	○	○	OFF	OFF	—	R	OFF
SM1352	UDP Socket 4--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1353	UDP Socket 4--错误标志	○	○	OFF	OFF	—	R	OFF
SM1354	UDP Socket 5--联机已开启标志	○	○	OFF	OFF	—	R	OFF
SM1355	UDP Socket 5--数据已接收标志	○	○	OFF	OFF	—	R	OFF
SM1356	UDP Socket 5--数据已传送标志	○	○	OFF	OFF	—	R	OFF
SM1357	UDP Socket 5--接收中标志	○	○	OFF	OFF	—	R	OFF
SM1358	UDP Socket 5--错误标志	○	○	OFF	OFF	—	R	OFF
SM1359	UDP Socket 6--联机已开启标志	○	○	OFF	OFF	—	R	OFF
SM1360	UDP Socket 6--数据已接收标志	○	○	OFF	OFF	—	R	OFF
SM1361	UDP Socket 6--数据已传送标志	○	○	OFF	OFF	—	R	OFF
SM1362	UDP Socket 6--接收中标志	○	○	OFF	OFF	—	R	OFF

A1

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1363	UDP Socket 6--错误标志	○	○	OFF	OFF	-	R	OFF
SM1364	UDP Socket 7--联机已开启标志	○	○	OFF	OFF	-	R	OFF
SM1365	UDP Socket 7--数据已接收标志	○	○	OFF	OFF	-	R	OFF
SM1366	UDP Socket 7--数据已传送标志	○	○	OFF	OFF	-	R	OFF
SM1367	UDP Socket 7--接收中标志	○	○	OFF	OFF	-	R	OFF
SM1368	UDP Socket 7--错误标志	○	○	OFF	OFF	-	R	OFF
SM1369	UDP Socket 8--联机已开启标志	○	○	OFF	OFF	-	R	OFF
SM1370	UDP Socket 8--数据已接收标志	○	○	OFF	OFF	-	R	OFF
SM1371	UDP Socket 8--数据已传送标志	○	○	OFF	OFF	-	R	OFF
SM1372	UDP Socket 8--接收中标志	○	○	OFF	OFF	-	R	OFF
SM1373	UDP Socket 8--错误标志	○	○	OFF	OFF	-	R	OFF
SM1374	Web 设定错误	○	○	OFF	-	-	R	OFF
SM1375	TCP Socket 1--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1376	TCP Socket 2--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1377	TCP Socket 3--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1378	TCP Socket 4--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1379	TCP Socket 5--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1380	TCP Socket 6--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1381	TCP Socket 7--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1382	TCP Socket 8--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1383	UDP Socket 1--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1384	UDP Socket 2--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1385	UDP Socket 3--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1386	UDP Socket 4--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1387	UDP Socket 5--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1388	UDP Socket 6--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1389	UDP Socket 7--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1390	UDP Socket 8--奇数字节启动标志	○	○	OFF	-	-	R/W	OFF
SM1752	COM2-MODBUS 数据交换中标志 (从站 ID 1)	○	×	OFF	-	-	R	OFF
SM1753	COM2-MODBUS 数据交换中标志 (从站 ID 2)	○	×	OFF	-	-	R	OFF
SM1754	COM2-MODBUS 数据交换中标志 (从站 ID 3)	○	×	OFF	-	-	R	OFF
SM1755	COM2-MODBUS 数据交换中标志 (从站 ID 4)	○	×	OFF	-	-	R	OFF
SM1756	COM2-MODBUS 数据交换中标志 (从站 ID 5)	○	×	OFF	-	-	R	OFF
SM1757	COM2-MODBUS 数据交换中标志 (从站 ID 6)	○	×	OFF	-	-	R	OFF
SM1758	COM2-MODBUS 数据交换中标志 (从站 ID 7)	○	×	OFF	-	-	R	OFF
A1 SM1759	COM2-MODBUS 数据交换中标志 (从站 ID 8)	○	×	OFF	-	-	R	OFF
SM1760	COM2-MODBUS 数据交换中标志 (从站 ID 9)	○	×	OFF	-	-	R	OFF
SM1761	COM2-MODBUS 数据交换中标志 (从站 ID 10)	○	×	OFF	-	-	R	OFF
SM1762	COM2-MODBUS 数据交换中标志 (从站 ID 11)	○	×	OFF	-	-	R	OFF
SM1763	COM2-MODBUS 数据交换中标志 (从站 ID 12)	○	×	OFF	-	-	R	OFF
SM1764	COM2-MODBUS 数据交换中标志 (从站 ID 13)	○	×	OFF	-	-	R	OFF
SM1765	COM2-MODBUS 数据交换中标志 (从站 ID 14)	○	×	OFF	-	-	R	OFF
SM1766	COM2-MODBUS 数据交换中标志 (从站 ID 15)	○	×	OFF	-	-	R	OFF
SM1767	COM2-MODBUS 数据交换中标志 (从站 ID 16)	○	×	OFF	-	-	R	OFF
SM1768	COM2-MODBUS 数据交换中标志 (从站 ID 17)	○	×	OFF	-	-	R	OFF
SM1769	COM2-MODBUS 数据交换中标志 (从站 ID 18)	○	×	OFF	-	-	R	OFF
SM1770	COM2-MODBUS 数据交换中标志 (从站 ID 19)	○	×	OFF	-	-	R	OFF
SM1771	COM2-MODBUS 数据交换中标志 (从站 ID 20)	○	×	OFF	-	-	R	OFF
SM1772	COM2-MODBUS 数据交换中标志 (从站 ID 21)	○	×	OFF	-	-	R	OFF
SM1773	COM2-MODBUS 数据交换中标志 (从站 ID 22)	○	×	OFF	-	-	R	OFF
SM1774	COM2-MODBUS 数据交换中标志 (从站 ID 23)	○	×	OFF	-	-	R	OFF
SM1775	COM2-MODBUS 数据交换中标志 (从站 ID 24)	○	×	OFF	-	-	R	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1776	COM2-MODBUS 数据交换中标志 (从站 ID 25)	○	×	OFF	-	-	R	OFF
SM1777	COM2-MODBUS 数据交换中标志 (从站 ID 26)	○	×	OFF	-	-	R	OFF
SM1778	COM2-MODBUS 数据交换中标志 (从站 ID 27)	○	×	OFF	-	-	R	OFF
SM1779	COM2-MODBUS 数据交换中标志 (从站 ID 28)	○	×	OFF	-	-	R	OFF
SM1780	COM2-MODBUS 数据交换中标志 (从站 ID 29)	○	×	OFF	-	-	R	OFF
SM1781	COM2-MODBUS 数据交换中标志 (从站 ID 30)	○	×	OFF	-	-	R	OFF
SM1782	COM2-MODBUS 数据交换中标志 (从站 ID 31)	○	×	OFF	-	-	R	OFF
SM1783	COM2-MODBUS 数据交换中标志 (从站 ID 32)	○	×	OFF	-	-	R	OFF
SM1784	COM2-MODBUS 读取错误标志 (从站 ID 1)	○	×	OFF	-	-	R	OFF
SM1785	COM2-MODBUS 读取错误标志 (从站 ID 2)	○	×	OFF	-	-	R	OFF
SM1786	COM2-MODBUS 读取错误标志 (从站 ID 3)	○	×	OFF	-	-	R	OFF
SM1787	COM2-MODBUS 读取错误标志 (从站 ID 4)	○	×	OFF	-	-	R	OFF
SM1788	COM2-MODBUS 读取错误标志 (从站 ID 5)	○	×	OFF	-	-	R	OFF
SM1789	COM2-MODBUS 读取错误标志 (从站 ID 6)	○	×	OFF	-	-	R	OFF
SM1790	COM2-MODBUS 读取错误标志 (从站 ID 7)	○	×	OFF	-	-	R	OFF
SM1791	COM2-MODBUS 读取错误标志 (从站 ID 8)	○	×	OFF	-	-	R	OFF
SM1792	COM2-MODBUS 读取错误标志 (从站 ID 9)	○	×	OFF	-	-	R	OFF
SM1793	COM2-MODBUS 读取错误标志 (从站 ID 10)	○	×	OFF	-	-	R	OFF
SM1794	COM2-MODBUS 读取错误标志 (从站 ID 11)	○	×	OFF	-	-	R	OFF
SM1795	COM2-MODBUS 读取错误标志 (从站 ID 12)	○	×	OFF	-	-	R	OFF
SM1796	COM2-MODBUS 读取错误标志 (从站 ID 13)	○	×	OFF	-	-	R	OFF
SM1797	COM2-MODBUS 读取错误标志 (从站 ID 14)	○	×	OFF	-	-	R	OFF
SM1798	COM2-MODBUS 读取错误标志 (从站 ID 15)	○	×	OFF	-	-	R	OFF
SM1799	COM2-MODBUS 读取错误标志 (从站 ID 16)	○	×	OFF	-	-	R	OFF
SM1800	COM2-MODBUS 读取错误标志 (从站 ID 17)	○	×	OFF	-	-	R	OFF
SM1801	COM2-MODBUS 读取错误标志 (从站 ID 18)	○	×	OFF	-	-	R	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1802	COM2-MODBUS 读取错误标志 (从站 ID 19)	○	×	OFF	-	-	R	OFF
SM1803	COM2-MODBUS 读取错误标志 (从站 ID 20)	○	×	OFF	-	-	R	OFF
SM1804	COM2-MODBUS 读取错误标志 (从站 ID 21)	○	×	OFF	-	-	R	OFF
SM1805	COM2-MODBUS 读取错误标志 (从站 ID 22)	○	×	OFF	-	-	R	OFF
SM1806	COM2-MODBUS 读取错误标志 (从站 ID 23)	○	×	OFF	-	-	R	OFF
SM1807	COM2-MODBUS 读取错误标志 (从站 ID 24)	○	×	OFF	-	-	R	OFF
SM1808	COM2-MODBUS 读取错误标志 (从站 ID 25)	○	×	OFF	-	-	R	OFF
SM1809	COM2-MODBUS 读取错误标志 (从站 ID 26)	○	×	OFF	-	-	R	OFF
SM1810	COM2-MODBUS 读取错误标志 (从站 ID 27)	○	×	OFF	-	-	R	OFF
SM1811	COM2-MODBUS 读取错误标志 (从站 ID 28)	○	×	OFF	-	-	R	OFF
SM1812	COM2-MODBUS 读取错误标志 (从站 ID 29)	○	×	OFF	-	-	R	OFF
SM1813	COM2-MODBUS 读取错误标志 (从站 ID 30)	○	×	OFF	-	-	R	OFF
SM1814	COM2-MODBUS 读取错误标志 (从站 ID 31)	○	×	OFF	-	-	R	OFF
SM1815	COM2-MODBUS 读取错误标志 (从站 ID 32)	○	×	OFF	-	-	R	OFF
SM1816	COM2-MODBUS 从站 1 写入错误标志	○	×	OFF	-	-	R	OFF
SM1817	COM2-MODBUS 从站 2 写入错误标志	○	×	OFF	-	-	R	OFF
SM1818	COM2-MODBUS 从站 3 写入错误标志	○	×	OFF	-	-	R	OFF
SM1819	COM2-MODBUS 从站 4 写入错误标志	○	×	OFF	-	-	R	OFF
SM1820	COM2-MODBUS 从站 5 写入错误标志	○	×	OFF	-	-	R	OFF
SM1821	COM2-MODBUS 从站 6 写入错误标志	○	×	OFF	-	-	R	OFF
SM1822	COM2-MODBUS 从站 7 写入错误标志	○	×	OFF	-	-	R	OFF
SM1823	COM2-MODBUS 从站 8 写入错误标志	○	×	OFF	-	-	R	OFF
SM1824	COM2-MODBUS 从站 9 写入错误标志	○	×	OFF	-	-	R	OFF
SM1825	COM2-MODBUS 从站 10 写入错误标志	○	×	OFF	-	-	R	OFF
SM1826	COM2-MODBUS 从站 11 写入错误标志	○	×	OFF	-	-	R	OFF
SM1827	COM2-MODBUS 从站 12 写入错误标志	○	×	OFF	-	-	R	OFF

A1

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1828	COM2-MODBUS 从站 13 写入错误标志	○	×	OFF	-	-	R	OFF
SM1829	COM2-MODBUS 从站 14 写入错误标志	○	×	OFF	-	-	R	OFF
SM1830	COM2-MODBUS 从站 15 写入错误标志	○	×	OFF	-	-	R	OFF
SM1831	COM2-MODBUS 从站 16 写入错误标志	○	×	OFF	-	-	R	OFF
SM1832	COM2-MODBUS 从站 17 写入错误标志	○	×	OFF	-	-	R	OFF
SM1833	COM2-MODBUS 从站 18 写入错误标志	○	×	OFF	-	-	R	OFF
SM1834	COM2-MODBUS 从站 19 写入错误标志	○	×	OFF	-	-	R	OFF
SM1835	COM2-MODBUS 从站 20 写入错误标志	○	×	OFF	-	-	R	OFF
SM1836	COM2-MODBUS 从站 21 写入错误标志	○	×	OFF	-	-	R	OFF
SM1837	COM2-MODBUS 从站 22 写入错误标志	○	×	OFF	-	-	R	OFF
SM1838	COM2-MODBUS 从站 23 写入错误标志	○	×	OFF	-	-	R	OFF
SM1839	COM2-MODBUS 从站 24 写入错误标志	○	×	OFF	-	-	R	OFF
SM1840	COM2-MODBUS 从站 25 写入错误标志	○	×	OFF	-	-	R	OFF
SM1841	COM2-MODBUS 从站 26 写入错误标志	○	×	OFF	-	-	R	OFF
SM1842	COM2-MODBUS 从站 27 写入错误标志	○	×	OFF	-	-	R	OFF
SM1843	COM2-MODBUS 从站 28 写入错误标志	○	×	OFF	-	-	R	OFF
SM1844	COM2-MODBUS 从站 29 写入错误标志	○	×	OFF	-	-	R	OFF
SM1845	COM2-MODBUS 从站 30 写入错误标志	○	×	OFF	-	-	R	OFF
SM1846	COM2-MODBUS 从站 31 写入错误标志	○	×	OFF	-	-	R	OFF
SM1847	COM2-MODBUS 从站 32 写入错误标志	○	×	OFF	-	-	R	OFF
SM1848	COM2-MODBUS 从站 1 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1849	COM2-MODBUS 从站 2 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1850	COM2-MODBUS 从站 3 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1851	COM2-MODBUS 从站 4 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1852	COM2-MODBUS 从站 5 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1853	COM2-MODBUS 从站 6 读取数据完毕标志	○	×	OFF	-	-	R	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1854	COM2-MODBUS 从站 7 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1855	COM2-MODBUS 从站 8 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1856	COM2-MODBUS 从站 9 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1857	COM2-MODBUS 从站 10 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1858	COM2-MODBUS 从站 11 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1859	COM2-MODBUS 从站 12 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1860	COM2-MODBUS 从站 13 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1861	COM2-MODBUS 从站 14 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1862	COM2-MODBUS 从站 15 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1863	COM2-MODBUS 从站 16 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1864	COM2-MODBUS 从站 17 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1865	COM2-MODBUS 从站 18 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1866	COM2-MODBUS 从站 19 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1867	COM2-MODBUS 从站 20 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1868	COM2-MODBUS 从站 21 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1869	COM2-MODBUS 从站 22 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1870	COM2-MODBUS 从站 23 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1871	COM2-MODBUS 从站 24 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1872	COM2-MODBUS 从站 25 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1873	COM2-MODBUS 从站 26 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1874	COM2-MODBUS 从站 27 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1875	COM2-MODBUS 从站 28 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1876	COM2-MODBUS 从站 29 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1877	COM2-MODBUS 从站 30 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1878	COM2-MODBUS 从站 31 读取数据完毕标志	○	×	OFF	-	-	R	OFF
SM1879	COM2-MODBUS 从站 32 读取数据完毕标志	○	×	OFF	-	-	R	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1880	COM2-MODBUS 从站 1 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1881	COM2-MODBUS 从站 2 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1882	COM2-MODBUS 从站 3 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1883	COM2-MODBUS 从站 4 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1884	COM2-MODBUS 从站 5 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1885	COM2-MODBUS 从站 6 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1886	COM2-MODBUS 从站 7 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1887	COM2-MODBUS 从站 8 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1888	COM2-MODBUS 从站 9 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1889	COM2-MODBUS 从站 10 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1890	COM2-MODBUS 从站 11 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1891	COM2-MODBUS 从站 12 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1892	COM2-MODBUS 从站 13 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1893	COM2-MODBUS 从站 14 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1894	COM2-MODBUS 从站 15 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1895	COM2-MODBUS 从站 16 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1896	COM2-MODBUS 从站 17 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1897	COM2-MODBUS 从站 18 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1898	COM2-MODBUS 从站 19 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1899	COM2-MODBUS 从站 20 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1900	COM2-MODBUS 从站 21 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1901	COM2-MODBUS 从站 22 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1902	COM2-MODBUS 从站 23 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1903	COM2-MODBUS 从站 24 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1904	COM2-MODBUS 从站 25 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1905	COM2-MODBUS 从站 26 写入数据完毕标志	○	×	OFF	-	-	R	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1906	COM2-MODBUS 从站 27 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1907	COM2-MODBUS 从站 28 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1908	COM2-MODBUS 从站 29 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1909	COM2-MODBUS 从站 30 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1910	COM2-MODBUS 从站 31 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1911	COM2-MODBUS 从站 32 写入数据完毕标志	○	×	OFF	-	-	R	OFF
SM1912	COM2-MODBUS 同步读/写功能	○	×	OFF	-	-	R/W	OFF
SM1913	COM2-MODBUS 从站 1 启动标志	○	×	OFF	-	-	R/W	OFF
SM1914	COM2-MODBUS 从站 2 启动标志	○	×	OFF	-	-	R/W	OFF
SM1915	COM2-MODBUS 从站 3 启动标志	○	×	OFF	-	-	R/W	OFF
SM1916	COM2-MODBUS 从站 4 启动标志	○	×	OFF	-	-	R/W	OFF
SM1917	COM2-MODBUS 从站 5 启动标志	○	×	OFF	-	-	R/W	OFF
SM1918	COM2-MODBUS 从站 6 启动标志	○	×	OFF	-	-	R/W	OFF
SM1919	COM2-MODBUS 从站 7 启动标志	○	×	OFF	-	-	R/W	OFF
SM1920	COM2-MODBUS 从站 8 启动标志	○	×	OFF	-	-	R/W	OFF
SM1921	COM2-MODBUS 从站 9 启动标志	○	×	OFF	-	-	R/W	OFF
SM1922	COM2-MODBUS 从站 10 启动标志	○	×	OFF	-	-	R/W	OFF
SM1923	COM2-MODBUS 从站 11 启动标志	○	×	OFF	-	-	R/W	OFF
SM1924	COM2-MODBUS 从站 12 启动标志	○	×	OFF	-	-	R/W	OFF
SM1925	COM2-MODBUS 从站 13 启动标志	○	×	OFF	-	-	R/W	OFF
SM1926	COM2-MODBUS 从站 14 启动标志	○	×	OFF	-	-	R/W	OFF
SM1927	COM2-MODBUS 从站 15 启动标志	○	×	OFF	-	-	R/W	OFF
SM1928	COM2-MODBUS 从站 16 启动标志	○	×	OFF	-	-	R/W	OFF
SM1929	COM2-MODBUS 从站 17 启动标志	○	×	OFF	-	-	R/W	OFF
SM1930	COM2-MODBUS 从站 18 启动标志	○	×	OFF	-	-	R/W	OFF
SM1931	COM2-MODBUS 从站 19 启动标志	○	×	OFF	-	-	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1932	COM2-MODBUS 从站 20 启动标志	○	×	OFF	-	-	R/W	OFF
SM1933	COM2-MODBUS 从站 21 启动标志	○	×	OFF	-	-	R/W	OFF
SM1934	COM2-MODBUS 从站 22 启动标志	○	×	OFF	-	-	R/W	OFF
SM1935	COM2-MODBUS 从站 23 启动标志	○	×	OFF	-	-	R/W	OFF
SM1936	COM2-MODBUS 从站 24 启动标志	○	×	OFF	-	-	R/W	OFF
SM1937	COM2-MODBUS 从站 25 启动标志	○	×	OFF	-	-	R/W	OFF
SM1938	COM2-MODBUS 从站 26 启动标志	○	×	OFF	-	-	R/W	OFF
SM1939	COM2-MODBUS 从站 27 启动标志	○	×	OFF	-	-	R/W	OFF
SM1940	COM2-MODBUS 从站 28 启动标志	○	×	OFF	-	-	R/W	OFF
SM1941	COM2-MODBUS 从站 29 启动标志	○	×	OFF	-	-	R/W	OFF
SM1942	COM2-MODBUS 从站 30 启动标志	○	×	OFF	-	-	R/W	OFF
SM1943	COM2-MODBUS 从站 31 启动标志	○	×	OFF	-	-	R/W	OFF
SM1944	COM2-MODBUS 从站 32 启动标志	○	×	OFF	-	-	R/W	OFF
*SM2000	EMDRW 1 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2001	EMDRW 1 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2002	EMDRW 1 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2003	EMDRW 1 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2004	EMDRW 1 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2005	EMDRW 1 关闭标志	○	○	ON	ON	ON	R	ON
*SM2006	EMDRW 2 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2007	EMDRW 2 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2008	EMDRW 2 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2009	EMDRW 2 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2010	EMDRW 2 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2011	EMDRW 2 关闭标志	○	○	ON	ON	ON	R	ON
*SM2012	EMDRW 3 送信标志	○	○	OFF	OFF	OFF	R/W	OFF

SM	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM2013	EMDRW 3 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2014	EMDRW 3 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2015	EMDRW 3 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2016	EMDRW 3 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2017	EMDRW 3 关闭标志	○	○	ON	ON	ON	R	ON
*SM2018	EMDRW 4 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2019	EMDRW 4 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2020	EMDRW 4 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2021	EMDRW 4 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2022	EMDRW 4 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2023	EMDRW 4 关闭标志	○	○	ON	ON	ON	R	ON
*SM2024	EMDRW 5 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2025	EMDRW 5 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2026	EMDRW 5 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2027	EMDRW 5 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2028	EMDRW 5 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2029	EMDRW 5 关闭标志	○	○	ON	ON	ON	R	ON
*SM2030	EMDRW 6 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2031	EMDRW 6 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2032	EMDRW 6 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2033	EMDRW 6 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2034	EMDRW 6 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2035	EMDRW 6 关闭标志	○	○	ON	ON	ON	R	ON
*SM2036	EMDRW 7 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2037	EMDRW 7 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2038	EMDRW 7 接收标志	○	○	OFF	OFF	OFF	R/W	OFF

SM	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM2039	EMDRW 7 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2040	EMDRW 7 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2041	EMDRW 7 关闭标志	○	○	ON	ON	ON	R	ON
*SM2042	EMDRW 8 送信标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2043	EMDRW 8 等待标志	○	○	OFF	OFF	OFF	R	OFF
*SM2044	EMDRW 8 接收标志	○	○	OFF	OFF	OFF	R/W	OFF
*SM2045	EMDRW 8 错误标志	○	○	OFF	OFF	OFF	R	OFF
*SM2046	EMDRW 8 逾时标志	○	○	OFF	OFF	OFF	R	OFF
*SM2047	EMDRW 8 关闭标志	○	○	ON	ON	ON	R	ON

注：有标注*的 SM 请参考 附录 A.3 SM/SR 补充说明。

A.1.1 特殊辅助继电器 SM 的更新时间

特殊辅助继电器	更新时间
SM0~SM1	系统自动设定与清除-程序执行有误自动设 ON
SM5	系统自动设定与清除- (1) 重新将程序写入 PLC (2) 重新上电 PLC 第一次 Stop=>Run
SM7	24V 电源供应不足时自动设 ON, 需用户自行清除
SM8	系统自动设定与清除-WDT 发生自动设 ON
SM9	系统自动设定与清除-System 有误自动设 ON
SM10	系统自动设定与清除-I/O BUS 有误自动设 ON
SM22 · SM23 · SM24	用户设定, 系统自动清除-ON 表执行清除
SM25~SM26	系统自动设定与清除-每一扫描周期更新一次
SM96~SM97	用户设定, 传送完毕后系统自动清除
SM98~SM99	系统自动设定与清除-传送通讯命令自动设 ON
SM100~SM101	系统自动设定, 用户清除-接收到回复的通讯命令自动设 ON
SM102~SM103	系统自动设定, 用户清除-接收到回复的通讯命令有误自动设 ON
SM104~SM105	系统自动设定, 用户清除-当接收逾时时自动设 ON
SM106~SM107	用户设定与清除-ON 表 8 位模式, OFF 表 16 位模式
SM108~SM109	用户设定与清除
SM204~SM205	用户设定, 系统自动清除-ON 表执行清除
SM206	用户设定与清除-ON 表输出全部禁止
SM209	用户设定, 系统自动清除-ON 表执行 COM1 通讯协议变更
SM210	用户设定与清除-ON 表 COM1 为 RTU 模式
SM211	用户设定, 系统自动清除-ON 表执行 COM2 通讯协议变更
SM212	用户设定与清除-ON 表 COM2 为 RTU 模式
SM215	用户设定与清除-ON 表 PLC RUN
SM220	用户设定与清除-ON 表万年历±30 秒校正
SM400~SM401	系统自动设定与清除-每一扫描周期更新一次
SM402~SM403	系统自动设定与清除-每执行至 END 更新
SM404	系统自动设定与清除-每 5ms 更新一次

特殊辅助继电器	更新时间
SM405	系统自动设定与清除—每 50ms 更新一次
SM406	系统自动设定与清除—每 100ms 更新一次
SM407	系统自动设定与清除—每 0.5s 更新一次
SM408	系统自动设定与清除—每 1s 更新一次
SM409	系统自动设定与清除—每 n (s) 更新一次 · n (s) 由 SR409 设定
SM410	系统自动设定与清除—每 n (s) 更新一次 · n (s) 由 SR410 设定
SM450	系统自动设定与清除—ON 表储存卡被插入主机
SM451	用户设定与清除—ON 表储存卡写保护
SM452	系统自动设定与清除—ON 表储存卡正在执行存取动作
SM453	系统自动设定与清除—ON 表储存卡运行中有错误发生
SM600~SM602	系统自动设定与清除—指令执行时更新
SM603	系统自动设定与清除—SORT 指令执行时更新
SM604	用户设定与清除—ON 降序排序 SORT 指令执行时更新
SM605	用户设定与清除
SM606	用户设定与清除—ON 表 8 位模式
SM607	用户自行设定
SM608	指令执行时更新
SM609	用户自行设定
SM610~SM611	指令执行时更新
SM612~SM613	用户自行设定
SM614	指令执行时更新
SM615~SM617	用户自行设定
SM618	指令执行时更新
SM619	EI 或 DI 指令执行时更新
SM620	CMPT 指令执行时更新
SM621~SM686	用户自行设定
SM687	RAMP 指令执行时更新
SM688	INCD 指令执行时更新
SM690~SM691	用户自行设定

特殊辅助继电器	更新时间
SM692	HKY 指令执行时更新
SM693	SEGL 指令执行时更新
SM694	DSW 指令执行时更新
SM695 · SM1000 · SM1001	用户自行设定
SM1090	TCP 连接满时为 ON
SM1091	UDP 连接满时为 ON
SM1106	PHY 初始化失败时为 ON
SM1107	IP、子网屏蔽、网关设定有误时为 ON
SM1108	滤波器设定有误时为 ON
SM1109	Socket 功能启动且使用同一个通讯端口时为 ON
SM1196	Socket 配置有误时为 ON
SM1270~SM1373	执行 socket 功能时更新
SM1374~SM1390	用户自行设定及清除
SM1752~SM1911	每一扫描周期更新一次
SM1912	用户自行设定及清除
SM1913~SM1944	每一扫描周期更新一次
SM2000	用户自行设定及清除
SM2001~SM2002	EMDRW 指令执行时更新
SM2003	EMDRW 指令执行时且有错误发生时更新
SM2004	EMDRW 指令执行时且响应逾时时更新
SM2005	用户自行设定及清除
SM2006~SM2007	EMDRW 指令执行时更新
SM2008	指令执行时且有错误发生时更新
SM2009	EMDRW 指令执行时且响应逾时时更新
SM2010	用户自行设定及清除
SM2011~SM2012	EMDRW 指令执行时更新
SM2013	指令执行时且有错误发生时更新
SM2014	EMDRW 指令执行时且响应逾时时更新

特殊辅助继电器	更新时间
SM2015	用户自行设定及清除
SM2016~SM2017	EMDRW 指令执行时更新
SM2018	指令执行时且有错误发生时更新
SM2019	EMDRW 指令执行时且响应逾时时更新
SM2020	用户自行设定及清除
SM2021~SM2022	EMDRW 指令执行时更新
SM2023	指令执行时且有错误发生时更新
SM2024	EMDRW 指令执行时且响应逾时时更新
SM2025	用户自行设定及清除
SM2026~SM2027	EMDRW 指令执行时更新
SM2028	指令执行时且有错误发生时更新
SM2029	EMDRW 指令执行时且响应逾时时更新
SM2030	用户自行设定及清除
SM2031~SM2032	EMDRW 指令执行时更新
SM2033	指令执行时且有错误发生时更新
SM2034	EMDRW 指令执行时且响应逾时时更新
SM2035	用户自行设定及清除
SM2036~SM2037	EMDRW 指令执行时更新
SM2038	指令执行时且有错误发生时更新
SM2039	EMDRW 指令执行时且响应逾时时更新
SM2040	用户自行设定及清除
SM2041~SM2042	EMDRW 指令执行时更新
SM2043	指令执行时且有错误发生时更新
SM2044	EMDRW 指令执行时且响应逾时时更新
SM2045	用户自行设定及清除
SM2046~SM2047	EMDRW 指令执行时更新

A.2 特殊数据寄存器 (SR)

每个特殊数据寄存器都有其特殊定义与用途，主要作为存放系统状态、错误信息、监视状态之用。特殊数据寄存器 (SR) 的种类及功能如下所示。编号前有 “*” 记号的 SR 装置可参考 **A.3 SM/SR 补充说明**，其中属性栏中标示为 “R” 者，表示仅可作读取的动作，若标示为 “R/W”，表示可作读取的动作。另若标示为 “-”，表示无变化。标示为 “#”，则表示系统会依照 PLC 状态作设定，用户可读取该设定值对照手册的说明，进一步了解系统信息。

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SR0	运算错误的检测错误号码	○	○	0	0	-	R	0
SR1	32位运算错误地址锁定	○	○	0	0	-	R	0
SR2		○	○	0	0	-	R	0
SR4	文法检查错误的检测错误号码	○	○	0	0	-	R	0
SR5	32位文法检查错误的地址锁定	○	○	0	0	-	R	0
SR6		○	○	0	0	-	R	0
SR8	WDT定时器ON的Step地址	○	○	0	-	-	R	0
*SR40	错误纪录 (Error Log) 的有效组数	○	○	-	-	-	R	0
*SR41	错误纪录的指针	○	○	-	-	-	R	0
*SR42	第一组的错误纪录的背版ID及插槽ID	○	○	-	-	-	R	0
*SR43	第一组的错误纪录的模块ID	○	○	-	-	-	R	0
*SR44	第一组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR45	第一组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR46	第一组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR47	第一组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR48	第二组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR49	第二组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR50	第二组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR51	第二组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR52	第二组的错误纪录的时间日及时	○	○	-	-	-	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR53	第二组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR54	第三组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR55	第三组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR56	第三组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR57	第三组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR58	第三组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR59	第三组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR60	第四组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR61	第四组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR62	第四组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR63	第四组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR64	第四组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR65	第四组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR66	第四组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
SR67	第五组的错误纪录的模块 ID	○	○	-	-	-	R	0
SR68	第五组的错误纪录的错误代码	○	○	-	-	-	R	0
SR69	第五组的错误纪录的时间年及月	○	○	-	-	-	R	0
SR70	第五组的错误纪录的时间日及时	○	○	-	-	-	R	0
SR71	第五组的错误纪录的时间分及秒	○	○	-	-	-	R	0
SR72	第六组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR73	第六组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR74	第六组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR75	第六组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR76	第六组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR77	第六组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR78	第七组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR79	第七组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR80	第七组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR81	第七组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR82	第七组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR83	第七组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR84	第八组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR85	第八组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR86	第八组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR87	第八组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR88	第八组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR89	第八组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR90	第九组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR91	第九组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR92	第九组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR93	第九组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR94	第九组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR95	第九组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR96	第十组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR97	第十组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR98	第十组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR99	第十组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR100	第十组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR101	第十组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR102	第十一组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR103	第十一组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR104	第十一组的错误纪录的错误代码	○	○	—	—	—	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR105	第十一组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR106	第十一组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR107	第十一组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR108	第十二组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR109	第十二组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR110	第十二组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR111	第十二组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR112	第十二组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR113	第十二组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR114	第十三组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR115	第十三组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR116	第十三组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR117	第十三组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR118	第十三组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR119	第十三组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR120	第十三组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR121	第十四组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR122	第十四组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR123	第十四组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR124	第十四组的错误纪录的时间日及时	○	○	-	-	-	R	0
*SR125	第十四组的错误纪录的时间分及秒	○	○	-	-	-	R	0
*SR126	第十五组的错误纪录的背板 ID 及插槽 ID	○	○	-	-	-	R	0
*SR127	第十五组的错误纪录的模块 ID	○	○	-	-	-	R	0
*SR128	第十五组的错误纪录的错误代码	○	○	-	-	-	R	0
*SR129	第十五组的错误纪录的时间年及月	○	○	-	-	-	R	0
*SR130	第十五组的错误纪录的时间日及时	○	○	-	-	-	R	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR131	第十五组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR132	第十六组的错误纪录的背版 ID 及插槽 ID	○	○	—	—	—	R	0
*SR133	第十六组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR134	第十六组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR135	第十六组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR136	第十六组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR137	第十六组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR138	第十七组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR139	第十七组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR140	第十七组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR141	第十七组的错误纪录的时间年及月	○	○	—	—	—	R	0
SR142	第十七组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR143	第十七组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR144	第十八组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR145	第十八组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR146	第十八组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR147	第十八组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR148	第十八组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR149	第十八组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR150	第十九组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0
*SR151	第十九组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR152	第十九组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR153	第十九组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR154	第十九组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR155	第十九组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR156	第二十组的错误纪录的背板 ID 及插槽 ID	○	○	—	—	—	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR157	第二十组的错误纪录的模块 ID	○	○	—	—	—	R	0
*SR158	第二十组的错误纪录的错误代码	○	○	—	—	—	R	0
*SR159	第二十组的错误纪录的时间年及月	○	○	—	—	—	R	0
*SR160	第二十组的错误纪录的时间日及时	○	○	—	—	—	R	0
*SR161	第二十组的错误纪录的时间分及秒	○	○	—	—	—	R	0
*SR201	PLC COM1通讯地址	○	○	—	—	—	R/W	1
*SR202	PLC COM2通讯地址	○	×	—	—	—	R/W	3
*SR209	COM1通讯协议	○	○	—	—	—	R/W	16# 0024
*SR210	COM1通讯逾时时间 (单位 : ms) · 设定为0则为不Time Out	○	○	3000 ms	—	—	R/W	3000 ms
*SR211	COM1命令重送次数	○	○	—	—	—	R/W	3
*SR212	COM2通讯协议	○	×	—	—	—	R/W	16# 0024
*SR213	COM2通讯逾时时间 (单位 : ms) · 设定为0表不Time Out	○	×	3000 ms	—	—	R/W	3000 ms
*SR214	COM2命令重送次数	○	○	—	—	—	R/W	3
*SR215	COM1界面代号	×	×	—	—	—	R/W	0
*SR216	COM2界面代号	×	×	—	—	—	R/W	0
*SR220	万年历 (RTC) 年 (公元) 00~99	○	×	—	—	—	R	0
*SR221	万年历 (RTC) 月01~12	○	×	—	—	—	R	1
*SR222	万年历 (RTC) 日01~31	○	×	—	—	—	R	1
*SR223	万年历 (RTC) 时00~23	○	×	—	—	—	R	0
*SR224	万年历 (RTC) 分00~59	○	×	—	—	—	R	0
*SR225	万年历 (RTC) 秒00~59	○	×	—	—	—	R	0
*SR226	万年历 (RTC) 星期1~7	○	×	—	—	—	R	1

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR227	储存下载动作的有效组数 (最多纪录20次)	○	○	-	-	-	R	0
*SR228	最新一次下载动作的纪录指针。	○	○	-	-	-	R	0
*SR229	下载纪录第1组, 纪录动作编号	○	○	-	-	-	R	0
*SR230	下载纪录第1组, 纪录时间年月	○	○	-	-	-	R	0
*SR231	下载纪录第1组, 纪录时间日时	○	○	-	-	-	R	0
*SR232	下载纪录第1组, 纪录时间分秒	○	○	-	-	-	R	0
*SR233	下载纪录第2组, 纪录动作编号	○	○	-	-	-	R	0
*SR234	下载纪录第2组, 纪录时间年月	○	○	-	-	-	R	0
*SR235	下载纪录第2组, 纪录时间日时	○	○	-	-	-	R	0
*SR236	下载纪录第2组, 纪录时间分秒	○	○	-	-	-	R	0
*SR237	下载纪录第3组, 纪录动作编号	○	○	-	-	-	R	0
*SR238	下载纪录第3组, 纪录时间年月	○	○	-	-	-	R	0
*SR239	下载纪录第3组, 纪录时间日时	○	○	-	-	-	R	0
*SR240	下载纪录第3组, 纪录时间分秒	○	○	-	-	-	R	0
*SR241	下载纪录第4组, 纪录动作编号	○	○	-	-	-	R	0
*SR242	下载纪录第4组, 纪录时间年月	○	○	-	-	-	R	0
*SR243	下载纪录第4组, 纪录时间日时	○	○	-	-	-	R	0
*SR244	下载纪录第4组, 纪录时间分秒	○	○	-	-	-	R	0
*SR245	下载纪录第5组, 纪录动作编号	○	○	-	-	-	R	0
*SR246	下载纪录第5组, 纪录时间年月	○	○	-	-	-	R	0
*SR247	下载纪录第5组, 纪录时间日时	○	○	-	-	-	R	0
*SR248	下载纪录第5组, 纪录时间分秒	○	○	-	-	-	R	0
*SR249	下载纪录第6组, 纪录动作编号	○	○	-	-	-	R	0
*SR250	下载纪录第6组, 纪录时间年月	○	○	-	-	-	R	0
*SR251	下载纪录第6组, 纪录时间日时	○	○	-	-	-	R	0
*SR252	下载纪录第6组, 纪录时间分秒	○	○	-	-	-	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR253	下载纪录第7组·纪录动作编号	○	○	—	—	—	R	0
*SR254	下载纪录第7组·纪录时间年月	○	○	—	—	—	R	0
*SR255	下载纪录第7组·纪录时间日时	○	○	—	—	—	R	0
*SR256	下载纪录第7组·纪录时间分秒	○	○	—	—	—	R	0
*SR257	下载纪录第8组·纪录动作编号	○	○	—	—	—	R	0
*SR258	下载纪录第8组·纪录时间年月	○	○	—	—	—	R	0
*SR259	下载纪录第8组·纪录时间日时	○	○	—	—	—	R	0
*SR260	下载纪录第8组·纪录时间分秒	○	○	—	—	—	R	0
*SR261	下载纪录第9组·纪录动作编号	○	○	—	—	—	R	0
*SR262	下载纪录第9组·纪录时间年月	○	○	—	—	—	R	0
*SR263	下载纪录第9组·纪录时间日时	○	○	—	—	—	R	0
*SR264	下载纪录第9组·纪录时间分秒	○	○	—	—	—	R	0
*SR265	下载纪录第10组·纪录动作编号	○	○	—	—	—	R	0
*SR266	下载纪录第10组·纪录时间年月	○	○	—	—	—	R	0
*SR267	下载纪录第10组·纪录时间日时	○	○	—	—	—	R	0
*SR268	下载纪录第10组·纪录时间分秒	○	○	—	—	—	R	0
*SR269	下载纪录第11组·纪录动作编号	○	○	—	—	—	R	0
*SR270	下载纪录第11组·纪录时间年月	○	○	—	—	—	R	0
*SR271	下载纪录第11组·纪录时间日时	○	○	—	—	—	R	0
*SR272	下载纪录第11组·纪录时间分秒	○	○	—	—	—	R	0
*SR273	下载纪录第12组·纪录动作编号	○	○	—	—	—	R	0
*SR274	下载纪录第12组·纪录时间年月	○	○	—	—	—	R	0
*SR275	下载纪录第12组·纪录时间日时	○	○	—	—	—	R	0
*SR276	下载纪录第12组·纪录时间分秒	○	○	—	—	—	R	0
*SR277	下载纪录第13组·纪录动作编号	○	○	—	—	—	R	0
*SR278	下载纪录第13组·纪录时间年月	○	○	—	—	—	R	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR279	下载纪录第13组·纪录时间日时	○	○	—	—	—	R	0
*SR280	下载纪录第13组·纪录时间分秒	○	○	—	—	—	R	0
*SR281	下载纪录第14组·纪录动作编号	○	○	—	—	—	R	0
*SR282	下载纪录第14组·纪录时间年月	○	○	—	—	—	R	0
*SR283	下载纪录第14组·纪录时间日时	○	○	—	—	—	R	0
*SR284	下载纪录第14组·纪录时间分秒	○	○	—	—	—	R	0
*SR285	下载纪录第15组·纪录动作编号	○	○	—	—	—	R	0
*SR286	下载纪录第15组·纪录时间年月	○	○	—	—	—	R	0
*SR287	下载纪录第15组·纪录时间日时	○	○	—	—	—	R	0
*SR288	下载纪录第15组·纪录时间分秒	○	○	—	—	—	R	0
*SR289	下载纪录第16组·纪录动作编号	○	○	—	—	—	R	0
*SR290	下载纪录第16组·纪录时间年月	○	○	—	—	—	R	0
*SR291	下载纪录第16组·纪录时间日时	○	○	—	—	—	R	0
*SR292	下载纪录第16组·纪录时间分秒	○	○	—	—	—	R	0
*SR293	下载纪录第17组·纪录动作编号	○	○	—	—	—	R	0
*SR294	下载纪录第17组·纪录时间年月	○	○	—	—	—	R	0
*SR295	下载纪录第17组·纪录时间日时	○	○	—	—	—	R	0
*SR296	下载纪录第17组·纪录时间分秒	○	○	—	—	—	R	0
*SR297	下载纪录第18组·纪录动作编号	○	○	—	—	—	R	0
*SR298	下载纪录第18组·纪录时间年月	○	○	—	—	—	R	0
*SR299	下载纪录第18组·纪录时间日时	○	○	—	—	—	R	0
*SR300	下载纪录第18组·纪录时间分秒	○	○	—	—	—	R	0
*SR301	下载纪录第19组·纪录动作编号	○	○	—	—	—	R	0
*SR302	下载纪录第19组·纪录时间年月	○	○	—	—	—	R	0
*SR303	下载纪录第19组·纪录时间日时	○	○	—	—	—	R	0
*SR304	下载纪录第19组·纪录时间分秒	○	○	—	—	—	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR305	下载纪录第20组·纪录动作编号	○	○	-	-	-	R	0
*SR306	下载纪录第20组·纪录时间年月	○	○	-	-	-	R	0
*SR307	下载纪录第20组·纪录时间日时	○	○	-	-	-	R	0
*SR308	下载纪录第20组·纪录时间分秒	○	○	-	-	-	R	0
*SR309	纪录PLC状态变更的总数(最大20组)	○	○	-	-	-	R	0
*SR310	纪录PLC状态变更最新一次的指针	○	○	-	-	-	R	0
*SR311	PLC状态变更纪录第1组·纪录动作编号	○	○	-	-	-	R	0
*SR312	PLC状态变更纪录第1组·纪录时间年月	○	○	-	-	-	R	0
*SR313	PLC状态变更纪录第1组·纪录时间日时	○	○	-	-	-	R	0
*SR314	PLC状态变更纪录第1组·纪录时间分秒	○	○	-	-	-	R	0
*SR315	PLC状态变更纪录第2组·纪录动作编号	○	○	-	-	-	R	0
*SR316	PLC状态变更纪录第2组·纪录时间年月	○	○	-	-	-	R	0
*SR317	PLC状态变更纪录第2组·纪录时间日时	○	○	-	-	-	R	0
*SR318	PLC状态变更纪录第2组·纪录时间分秒	○	○	-	-	-	R	0
*SR319	PLC状态变更纪录第3组·纪录动作编号	○	○	-	-	-	R	0
*SR320	PLC状态变更纪录第3组·纪录时间年月	○	○	-	-	-	R	0
*SR321	PLC状态变更纪录第3组·纪录时间日时	○	○	-	-	-	R	0
*SR322	PLC状态变更纪录第3组·纪录时间分秒	○	○	-	-	-	R	0
*SR323	PLC状态变更纪录第4组·纪录动作编号	○	○	-	-	-	R	0
*SR324	PLC状态变更纪录第4组·纪录时间年月	○	○	-	-	-	R	0
*SR325	PLC状态变更纪录第4组·纪录时间日时	○	○	-	-	-	R	0
*SR326	PLC状态变更纪录第4组·纪录时间分秒	○	○	-	-	-	R	0
*SR327	PLC状态变更纪录第5组·纪录动作编号	○	○	-	-	-	R	0
*SR328	PLC状态变更纪录第5组·纪录时间年月	○	○	-	-	-	R	0
*SR329	PLC状态变更纪录第5组·纪录时间日时	○	○	-	-	-	R	0
*SR330	PLC状态变更纪录第5组·纪录时间分秒	○	○	-	-	-	R	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR331	PLC状态变更纪录第6组·纪录动作编号	○	○	—	—	—	R	0
*SR332	PLC状态变更纪录第6组·纪录时间年月	○	○	—	—	—	R	0
*SR333	PLC状态变更纪录第6组·纪录时间日時	○	○	—	—	—	R	0
*SR334	PLC状态变更纪录第6组·纪录时间分秒	○	○	—	—	—	R	0
*SR335	PLC状态变更纪录第7组·纪录动作编号	○	○	—	—	—	R	0
*SR336	PLC状态变更纪录第7组·纪录时间年月	○	○	—	—	—	R	0
*SR337	PLC状态变更纪录第7组·纪录时间日時	○	○	—	—	—	R	0
*SR338	PLC状态变更纪录第7组·纪录时间分秒	○	○	—	—	—	R	0
*SR339	PLC状态变更纪录第8组·纪录动作编号	○	○	—	—	—	R	0
*SR340	PLC状态变更纪录第8组·纪录时间年月	○	○	—	—	—	R	0
*SR341	PLC状态变更纪录第8组·纪录时间日時	○	○	—	—	—	R	0
*SR342	PLC状态变更纪录第8组·纪录时间分秒	○	○	—	—	—	R	0
*SR343	PLC状态变更纪录第9组·纪录动作编号	○	○	—	—	—	R	0
*SR344	PLC状态变更纪录第9组·纪录时间年月	○	○	—	—	—	R	0
*SR345	PLC状态变更纪录第9组·纪录时间日時	○	○	—	—	—	R	0
*SR346	PLC状态变更纪录第9组·纪录时间分秒	○	○	—	—	—	R	0
*SR347	PLC状态变更纪录第10组·纪录动作编号	○	○	—	—	—	R	0
*SR348	PLC状态变更纪录第10组·纪录时间年月	○	○	—	—	—	R	0
*SR349	PLC状态变更纪录第10组·纪录时间日時	○	○	—	—	—	R	0
*SR350	PLC状态变更纪录第10组·纪录时间分秒	○	○	—	—	—	R	0
*SR351	PLC状态变更纪录第11组·纪录动作编号	○	○	—	—	—	R	0
*SR352	PLC状态变更纪录第11组·纪录时间年月	○	○	—	—	—	R	0
*SR353	PLC状态变更纪录第11组·纪录时间日時	○	○	—	—	—	R	0
*SR354	PLC状态变更纪录第11组·纪录时间分秒	○	○	—	—	—	R	0
*SR355	PLC状态变更纪录第12组·纪录动作编号	○	○	—	—	—	R	0
*SR356	PLC状态变更纪录第12组·纪录时间年月	○	○	—	—	—	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR357	PLC状态变更纪录第12组, 纪录时间日时	○	○	-	-	-	R	0
*SR358	PLC状态变更纪录第12组, 纪录时间分秒	○	○	-	-	-	R	0
*SR359	PLC状态变更纪录第13组, 纪录动作编号	○	○	-	-	-	R	0
*SR360	PLC状态变更纪录第13组, 纪录时间年月	○	○	-	-	-	R	0
*SR361	PLC状态变更纪录第13组, 纪录时间日时	○	○	-	-	-	R	0
*SR362	PLC状态变更纪录第13组, 纪录时间分秒	○	○	-	-	-	R	0
*SR363	PLC状态变更纪录第14组, 纪录动作编号	○	○	-	-	-	R	0
*SR364	PLC状态变更纪录第14组, 纪录时间年月	○	○	-	-	-	R	0
*SR365	PLC状态变更纪录第14组, 纪录时间日时	○	○	-	-	-	R	0
*SR366	PLC状态变更纪录第14组, 纪录时间分秒	○	○	-	-	-	R	0
*SR367	PLC状态变更纪录第15组, 纪录动作编号	○	○	-	-	-	R	0
*SR368	PLC状态变更纪录第15组, 纪录时间年月	○	○	-	-	-	R	0
*SR369	PLC状态变更纪录第15组, 纪录时间日时	○	○	-	-	-	R	0
*SR370	PLC状态变更纪录第15组, 纪录时间分秒	○	○	-	-	-	R	0
*SR371	PLC状态变更纪录第16组, 纪录动作编号	○	○	-	-	-	R	0
*SR372	PLC状态变更纪录第16组, 纪录时间年月	○	○	-	-	-	R	0
*SR373	PLC状态变更纪录第16组, 纪录时间日时	○	○	-	-	-	R	0
*SR374	PLC状态变更纪录第16组, 纪录时间分秒	○	○	-	-	-	R	0
*SR375	PLC状态变更纪录第17组, 纪录动作编号	○	○	-	-	-	R	0
*SR376	PLC状态变更纪录第17组, 纪录时间年月	○	○	-	-	-	R	0
*SR377	PLC状态变更纪录第17组, 纪录时间日时	○	○	-	-	-	R	0
*SR378	PLC状态变更纪录第17组, 纪录时间分秒	○	○	-	-	-	R	0
*SR379	PLC状态变更纪录第18组, 纪录动作编号	○	○	-	-	-	R	0
*SR380	PLC状态变更纪录第18组, 纪录时间年月	○	○	-	-	-	R	0
*SR381	PLC状态变更纪录第18组, 纪录时间日时	○	○	-	-	-	R	0
*SR382	PLC状态变更纪录第18组, 纪录时间分秒	○	○	-	-	-	R	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR383	PLC状态变更纪录第19组·纪录动作编号	○	○	—	—	—	R	0
*SR384	PLC状态变更纪录第19组·纪录时间年月	○	○	—	—	—	R	0
*SR385	PLC状态变更纪录第19组·纪录时间日时	○	○	—	—	—	R	0
*SR386	PLC状态变更纪录第19组·纪录时间分秒	○	○	—	—	—	R	0
*SR387	PLC状态变更纪录第20组·纪录动作编号	○	○	—	—	—	R	0
*SR388	PLC状态变更纪录第20组·纪录时间年月	○	○	—	—	—	R	0
*SR389	PLC状态变更纪录第20组·纪录时间日时	○	○	—	—	—	R	0
*SR390	PLC状态变更纪录第20组·纪录时间分秒	○	○	—	—	—	R	0
SR391	万年历 (RTC) 年 (公元) 00~99	○	×	—	—	—	R	0
SR392	万年历 (RTC) 月01~12	○	×	—	—	—	R	1
SR393	万年历 (RTC) 日01~31	○	×	—	—	—	R	1
SR394	万年历 (RTC) 时00~23	○	×	—	—	—	R	0
SR395	万年历 (RTC) 分00~59	○	×	—	—	—	R	0
SR396	万年历 (RTC) 秒00~59	○	×	—	—	—	R	0
SR397	万年历 (RTC) 星期1~7	○	×	—	—	—	R	1
SR407	RUN后·每秒加1·重复计数由0~32767再由-32768~0	○	○	0	0	—	R/W	0
SR408	RUN后·每完成一次扫描后加1·重复计数由0~32767再由-32768~0	○	○	0	0	—	R/W	0
*SR409	储存数值n于2n秒时钟·设定可为1到32767	○	○	—	—	—	R/W	30
*SR410	储存数值n于2n毫秒时钟	○	○	—	—	—	R/W	30
SR411	目前扫描时间储存至SR411及SR412·测量单位为100微秒·SR411:储存毫秒部分(储存范围:0~65535)SR412:储存微秒部份·		○	0	—	—	R	0
SR412	(储存范围:0~900) 例:目前扫描时间为12.3毫秒·储存以下值·SR411=12; SR412=300	○	○	0	—	—	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SR413	储存扫描时间最大值到SR413及SR414 测量单位为100微秒SR413：储存毫秒	○	○	0	-	-	R	0
SR414			○	0	-	-	R	0
SR415	储存扫描时间最小值到SR415及SR416 测量单位为100微秒。SR415：储存毫秒	○	○	0	-	-	R	0
SR416			○	0	-	-	R	0
SR440 ↓ SR442	主机MAC值	○	○	MAC 值	-	-	R	MAC 值
SR443 ↓ SR451	主机产品序号	○	○	序号 值	-	-	R	序号 值
*SR453	当储存卡有错误发生，错误代码将被记录	○	○	-	-	-	R	0
SR621	RS指令中断字符 (COM1)	○	○	-	-	-	R/W	0
SR622	RS指令中断字符 (COM2)	x	x	-	-	-	R/W	0
SR623	位0~15：中断程序I0~I15条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR624	位0~15：中断程序I16~I31条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR625	位0~15：中断程序I32~I47条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR626	位0~15：中断程序I48~I63条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR627	位0~15：中断程序I64~I79条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR628	位0~15：中断程序I80~I95条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR629	位0~15：中断程序I96~I111条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF
SR630	位0~15：中断程序I112~I127条件，由IMASK指令设定	○	○	FFFF	-	-	R	FFFF

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SR631	位0~15：中断程序I128~I143条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR632	位0~15：中断程序I144~I159条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR633	位0~15：中断程序I160~I175条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR634	位0~15：中断程序I176~I191条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR635	位0~15：中断程序I192~I207条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR636	位0~15：中断程序I208~I213条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR637	位0~15：中断程序I214~I229条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
SR638	位0~15：中断程序I230~I255条件，由IMASK指令设定	○	○	FFFF	—	—	R	FFFF
*SR655	纪录背板1的I/O表映像错误或IO模块发生错误	○	○	0	—	—	R	0
*SR663 ↓ SR674	纪录背板1-插槽0的I/O表发生的映射错误码 ↓ 纪录背板1-插槽11的I/O表发生的映射错误码	○	x	0	—	—	R	0
SR731	外部24V低电压检测	○	○	0	—	—	R	0
*SR1000	以太网网络I地址上位字	○	○	—	—	—	R/W	C0A8
*SR1001	以太网网络IP地址下位字	○	○	—	—	—	R/W	0101
*SR1002	以太网网络屏蔽地址上位字	○	○	—	—	—	R/W	FFFF
*SR1003	以太网网络屏蔽地址下位字	○	○	—	—	—	R/W	FF00

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1004	以太网网关地址上位字	○	○	-	-	-	R/W	C0A8
*SR1005	以太网网关地址下位字	○	○	-	-	-	R/W	0101
*SR1006	TCP保持连接时间	○	○	-	-	-	R/W	0060
SR1007	以太网传输速度	○	○	0	-	-	R	0
SR1008	以太网传输模式	○	○	0	-	-	R	0
*SR1118	TCP Socket 1--本地通讯端口	○	○	-	-	-	R/W	0
*SR1119	TCP Socket 1--远程IP地址上位字	○	○	-	-	-	R/W	0
*SR1120	TCP Socket 1--远程IP地址下位字	○	○	-	-	-	R/W	0
*SR1121	TCP Socket 1--远程通讯端口	○	○	-	-	-	R/W	0
*SR1122	TCP Socket 1--传送数据长度	○	○	-	-	-	R/W	0
*SR1123	TCP Socket 1--传送数据地址上位字	○	○	-	-	-	R/W	0
*SR1124	TCP Socket 1--传送数据地址下位字	○	○	-	-	-	R/W	0
*SR1125	TCP Socket 1--接收数据长度	○	○	-	-	-	R/W	0
*SR1126	TCP Socket 1--接收数据地址上位字	○	○	-	-	-	R/W	0
*SR1127	TCP Socket 1--接收数据地址下位字	○	○	-	-	-	R/W	0
*SR1128	TCP Socket 1--保持连接时间	○	○	-	-	-	R/W	1000
*SR1129	TCP Socket 1--接收数据计数器	○	○	0	-	-	R	0
*SR1130	TCP Socket 1--传送数据计数器	○	○	0	-	-	R	0
*SR1131	TCP Socket 2--本地通讯端口	○	○	-	-	-	R/W	0
*SR1132	TCP Socket 2--远程IP地址上位字	○	○	-	-	-	R/W	0
*SR1133	TCP Socket 2--远程IP地址下位字	○	○	-	-	-	R/W	0
*SR1134	TCP Socket 2--远程通讯端口	○	○	-	-	-	R/W	0
*SR1135	TCP Socket 2--传送数据长度	○	○	-	-	-	R/W	0
*SR1136	TCP Socket 2--传送数据地址上位字	○	○	-	-	-	R/W	0
*SR1137	TCP Socket 2--传送数据地址下位字	○	○	-	-	-	R/W	0
*SR1138	TCP Socket 2--接收数据长度	○	○	-	-	-	R/W	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1139	TCP Socket 2--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1140	TCP Socket 2--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1141	TCP Socket 2--保持连接时间	○	○	—	—	—	R/W	1000
*SR1142	TCP Socket 2--接收数据计数器	○	○	0	—	—	R	0
*SR1143	TCP Socket 2--传送数据计数器	○	○	0	—	—	R	0
*SR1144	TCP Socket 3--本地通讯端口	○	○	—	—	—	R/W	0
*SR1145	TCP Socket 3--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1146	TCP Socket 3--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1147	TCP Socket 3--远程通讯端口	○	○	—	—	—	R/W	0
*SR1148	TCP Socket 3--传送数据长度	○	○	—	—	—	R/W	0
*SR1149	TCP Socket 3--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1150	TCP Socket 3--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1151	TCP Socket 3--接收数据长度	○	○	—	—	—	R/W	0
*SR1152	TCP Socket 3--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1153	TCP Socket 3--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1154	TCP Socket 3--保持连接时间	○	○	—	—	—	R/W	1000
*SR1155	TCP Socket 3--接收数据计数器	○	○	0	—	—	R	0
*SR1156	TCP Socket 3--传送数据计数器	○	○	0	—	—	R	0
*SR1157	TCP Socket 4--本地通讯端口	○	○	—	—	—	R/W	0
*SR1158	TCP Socket 4--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1159	TCP Socket 4--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1160	TCP Socket 4--远程通讯端口	○	○	—	—	—	R/W	0
*SR1161	TCP Socket 4--传送数据长度	○	○	—	—	—	R/W	0
*SR1162	TCP Socket 4--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1163	TCP Socket 4--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1164	TCP Socket 4--接收数据长度	○	○	—	—	—	R/W	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1165	TCP Socket 4--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1166	TCP Socket 4--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1167	TCP Socket 4--保持连接时间	○	○	—	—	—	R/W	1000
*SR1168	TCP Socket 4--接收数据计数器	○	○	0	—	—	R	0
*SR1169	TCP Socket 4--传送数据计数器	○	○	0	—	—	R	0
*SR1170	TCP Socket 5--本地通讯端口	○	○	—	—	—	R/W	0
*SR1171	TCP Socket 5--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1172	TCP Socket 5--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1173	TCP Socket 5--远程通讯端口	○	○	—	—	—	R/W	0
*SR1174	TCP Socket 5--传送数据长度	○	○	—	—	—	R/W	0
*SR1175	TCP Socket 5--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1176	TCP Socket 5--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1177	TCP Socket 5--接收数据长度	○	○	—	—	—	R/W	0
*SR1178	TCP Socket 5--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1179	TCP Socket 5--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1180	TCP Socket 5--保持连接时间	○	○	—	—	—	R/W	1000
*SR1181	TCP Socket 5--接收数据计数器	○	○	0	—	—	R	0
*SR1182	TCP Socket 5--传送数据计数器	○	○	0	—	—	R	0
*SR1183	TCP Socket 6--本地通讯端口	○	○	—	—	—	R/W	0
*SR1184	TCP Socket 6--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1185	TCP Socket 6--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1186	TCP Socket 6--远程通讯埠	○	○	—	—	—	R/W	0
*SR1187	TCP Socket 6--传送数据长度	○	○	—	—	—	R/W	0
*SR1188	TCP Socket 6--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1189	TCP Socket 6--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1190	TCP Socket 6--接收数据长度	○	○	—	—	—	R/W	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1191	TCP Socket 6--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1192	TCP Socket 6--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1193	TCP Socket 6--保持连接时间	○	○	—	—	—	R/W	1000
*SR1194	TCP Socket 6--接收数据计数器	○	○	0	—	—	R	0
*SR1195	TCP Socket 6--传送数据计数器	○	○	0	—	—	R	0
*SR1196	TCP Socket 7--本地通讯端口	○	○	—	—	—	R/W	0
*SR1197	TCP Socket 7--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1198	TCP Socket 7--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1199	TCP Socket 7--远程通讯端口	○	○	—	—	—	R/W	0
*SR1200	TCP Socket 7--传送数据长度	○	○	—	—	—	R/W	0
*SR1201	TCP Socket 7--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1202	TCP Socket 7--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1203	TCP Socket 7--接收数据长度	○	○	—	—	—	R/W	0
*SR1204	TCP Socket 7--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1205	TCP Socket 7--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1206	TCP Socket 7--保持连接时间	○	○	—	—	—	R/W	1000
*SR1207	TCP Socket 7--接收数据计数器	○	○	0	—	—	R	0
*SR1208	TCP Socket 7--传送数据计数器	○	○	0	—	—	R	0
*SR1209	TCP Socket 8--本地通讯端口	○	○	—	—	—	R/W	0
*SR1210	TCP Socket 8--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1211	TCP Socket 8--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1212	TCP Socket 8--远程通讯端口	○	○	—	—	—	R/W	0
*SR1213	TCP Socket 8--传送数据长度	○	○	—	—	—	R/W	0
*SR1214	TCP Socket 8--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1215	TCP Socket 8--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1216	TCP Socket 8--接收数据长度	○	○	—	—	—	R/W	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1217	TCP Socket 8--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1218	TCP Socket 8--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1219	TCP Socket 8--保持连接时间	○	○	—	—	—	R/W	1000
*SR1220	TCP Socket 8--接收数据计数器	○	○	0	—	—	R	0
*SR1221	TCP Socket 8--传送数据计数器	○	○	0	—	—	R	0
*SR1222	UDP Socket 1--本地通讯端口	○	○	—	—	—	R/W	0
*SR1223	UDP Socket 1--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1224	UDP Socket 1--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1225	UDP Socket 1--远程通讯端口	○	○	—	—	—	R/W	0
*SR1226	UDP Socket 1--传送数据长度	○	○	—	—	—	R/W	0
*SR1227	UDP Socket 1--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1228	UDP Socket 1--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1229	UDP Socket 1--接收数据长度	○	○	—	—	—	R/W	0
*SR1230	UDP Socket 1--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1231	UDP Socket 1--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1232	UDP Socket 1--接收数据计数器	○	○	0	—	—	R	0
*SR1233	UDP Socket 1--传送数据计数器	○	○	0	—	—	R	0
*SR1234	UDP Socket 2--本地通讯端口	○	○	—	—	—	R/W	0
*SR1235	UDP Socket 2--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1236	UDP Socket 2--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1237	UDP Socket 2--远程通讯端口	○	○	—	—	—	R/W	0
*SR1238	UDP Socket 2--传送数据长度	○	○	—	—	—	R/W	0
*SR1239	UDP Socket 2--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1240	UDP Socket 2--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1241	UDP Socket 2--接收数据长度	○	○	—	—	—	R/W	0
*SR1242	UDP Socket 2--接收数据地址上位字	○	○	—	—	—	R/W	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1243	UDP Socket 2--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1244	UDP Socket 2--接收数据计数器	○	○	0	—	—	R	0
*SR1245	UDP Socket 2--传送数据计数器	○	○	0	—	—	R	0
*SR1246	UDP Socket 3--本地通讯端口	○	○	—	—	—	R/W	0
*SR1247	UDP Socket 3--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1248	UDP Socket 3--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1249	UDP Socket 3--远程通讯端口	○	○	—	—	—	R/W	0
*SR1250	UDP Socket 3--传送数据长度	○	○	—	—	—	R/W	0
*SR1251	UDP Socket 3--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1252	UDP Socket 3--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1253	UDP Socket 3--接收数据长度	○	○	—	—	—	R/W	0
*SR1254	UDP Socket 3--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1255	UDP Socket 3--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1256	UDP Socket 3--接收数据计数器	○	○	0	—	—	R	0
*SR1257	UDP Socket 3--传送数据计数器	○	○	0	—	—	R	0
*SR1258	UDP Socket 4--本地通讯端口	○	○	—	—	—	R/W	0
*SR1259	UDP Socket 4--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1260	UDP Socket 4--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1261	UDP Socket 4--远程通讯端口	○	○	—	—	—	R/W	0
*SR1262	UDP Socket 4--传送数据长度	○	○	—	—	—	R/W	0
*SR1263	UDP Socket 4--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1264	UDP Socket 4--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1265	UDP Socket 4--接收数据长度	○	○	—	—	—	R/W	0
*SR1266	UDP Socket 4--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1267	UDP Socket 4--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1268	UDP Socket 4--接收数据计数器	○	○	0	—	—	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1269	UDP Socket 4--传送数据计数器	○	○	0	—	—	R	0
*SR1270	UDP Socket 5--本地通讯端口	○	○	—	—	—	R/W	0
*SR1271	UDP Socket 5--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1272	UDP Socket 5--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1273	UDP Socket 5--远程通讯端口	○	○	—	—	—	R/W	0
*SR1274	UDP Socket 5--传送数据长度	○	○	—	—	—	R/W	0
*SR1275	UDP Socket 5--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1276	UDP Socket 5--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1277	UDP Socket 5--接收数据长度	○	○	—	—	—	R/W	0
*SR1278	UDP Socket 5--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1279	UDP Socket 5--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1280	UDP Socket 5--接收数据计数器	○	○	0	—	—	R	0
*SR1281	UDP Socket 5--传送数据计数器	○	○	0	—	—	R	0
*SR1282	UDP Socket 6--本地通讯端口	○	○	—	—	—	R/W	0
*SR1283	UDP Socket 6--远程IP地址上位字	○	○	—	—	—	R/W	0
*SR1284	UDP Socket 6--远程IP地址下位字	○	○	—	—	—	R/W	0
*SR1285	UDP Socket 6--远程通讯端口	○	○	—	—	—	R/W	0
*SR1286	UDP Socket 6--传送数据长度	○	○	—	—	—	R/W	0
*SR1287	UDP Socket 6--传送数据地址上位字	○	○	—	—	—	R/W	0
*SR1288	UDP Socket 6--传送数据地址下位字	○	○	—	—	—	R/W	0
*SR1289	UDP Socket 6--接收数据长度	○	○	—	—	—	R/W	0
*SR1290	UDP Socket 6--接收数据地址上位字	○	○	—	—	—	R/W	0
*SR1291	UDP Socket 6--接收数据地址下位字	○	○	—	—	—	R/W	0
*SR1292	UDP Socket 6--接收数据计数器	○	○	0	—	—	R	0
*SR1293	UDP Socket 6--传送数据计数器	○	○	0	—	—	R	0
*SR1294	UDP Socket 7--本地通讯端口	○	○	—	—	—	R/W	0

SR	功能说明	AH-xxEMC (主机)	AH-xxEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1295	UDP Socket 7--远程IP地址上位字	○	○	-	-	-	R/W	0
*SR1296	UDP Socket 7--远程IP地址下位字	○	○	-	-	-	R/W	0
*SR1297	UDP Socket 7--远程通讯端口	○	○	-	-	-	R/W	0
*SR1298	UDP Socket 7--传送数据长度	○	○	-	-	-	R/W	0
*SR1299	UDP Socket 7--传送数据地址上位字	○	○	-	-	-	R/W	0
*SR1300	UDP Socket 7--传送数据地址下位字	○	○	-	-	-	R/W	0
*SR1301	UDP Socket 7--接收数据长度	○	○	-	-	-	R/W	0
*SR1302	UDP Socket 7--接收数据地址上位字	○	○	-	-	-	R/W	0
*SR1303	UDP Socket 7--接收数据地址下位字	○	○	-	-	-	R/W	0
*SR1304	UDP Socket 7--接收数据计数器	○	○	0	-	-	R	0
*SR1305	UDP Socket 7--传送数据计数器	○	○	0	-	-	R	0
*SR1306	UDP Socket 8--本地通讯端口	○	○	-	-	-	R/W	0
*SR1307	UDP Socket 8--远程IP地址上位字	○	○	-	-	-	R/W	0
*SR1308	UDP Socket 8--远程IP地址下位字	○	○	-	-	-	R/W	0
*SR1309	UDP Socket 8--远程通讯端口	○	○	-	-	-	R/W	0
*SR1310	UDP Socket 8--传送数据长度	○	○	-	-	-	R/W	0
*SR1311	UDP Socket 8--传送数据地址上位字	○	○	-	-	-	R/W	0
*SR1312	UDP Socket 8--传送数据地址下位字	○	○	-	-	-	R/W	0
*SR1313	UDP Socket 8--接收数据长度	○	○	-	-	-	R/W	0
*SR1314	UDP Socket 8--接收数据地址上位字	○	○	-	-	-	R/W	0
*SR1315	UDP Socket 8--接收数据地址下位字	○	○	-	-	-	R/W	0
*SR1316	UDP Socket 8--接收数据计数器	○	○	0	-	-	R	0
*SR1317	UDP Socket 8--传送数据计数器	○	○	0	-	-	R	0
*SR1318	Socket输入计数器	○	○	0	-	-	R	0
*SR1319	Socket输出计数器	○	○	0	-	-	R	0
*SR1320	Socket错误计数器	○	○	0	-	-	R	0

SR	功能说明	AH-XXEMC (主机)	AH-XXEMC (模块)	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR1329	主机背板ID	○	×	—	—	—	R/W	0
*SR1330	主机插槽ID	○	×	—	—	—	R/W	0

注：有标注*之 SR 请参考附录 A.3 SM/SR 补充说明。

A.2.1 特殊数据寄存器 SR 的更新时间

特殊数据寄存器	更新时间
SR0~SR2	程序执行错误时更新
SR4	语法检查错误时更新
SR5~SR6	程序下载至PLC时/电源开启后第一次从Stop到Run时更新
SR8	WDT发生时更新
SR40~SR161	有错误发生时更新一次
SR201~SR216	用户自行设定及清除
SR220~SR226	每一扫描周期更新一次
SR227~SR308	程序下载至PLC时更新
SR309~SR390	PLC状态改变时更新
SR391~SR397	每扫描周期更新
SR407	每1s更新一次
SR408	每执行至End时更新
SR409~SR410	用户自行设定及清除
SR411~SR416	每执行至End时更新
SR440~SR451	上电时更新
SR453	错误发生时更新
SR621~SR622	用户自行设定及清除
SR623~SR638	IMASK指令执行时更新
SR655~SR730	当I/O模块有错误时，会更新此区域
SR731	当外部24V低电压检测，状态改变时会更新此区域

特殊数据寄存器	更新时间
SR1000~SR1006	用户自行设定及清除
SR1007	上电与网络线连接时
SR1008	上电与网络线连接时
SR1118~SR1128	参数下载至PLC时更新
SR1129~SR1130	参数下载至PLC时或电源开启时更新
SR1131~SR1141	参数下载至PLC时更新
SR1142~SR1143	参数下载至PLC时或电源开启时更新
SR1144~SR1154	参数下载至PLC时更新
SR1155~SR1156	参数下载至PLC时或电源开启时更新
SR1157~SR1167	参数下载至PLC时更新
SR1168~SR1169	参数下载至PLC时或电源开启时更新
SR1170~SR1180	参数下载至PLC时更新
SR1181~SR1182	参数下载至PLC时或电源开启时更新
SR1183~SR1193	参数下载至PLC时更新
SR1194~SR1195	参数下载至PLC时或电源开启时更新
SR1196~SR1206	参数下载至PLC时更新
SR1207~SR1208	参数下载至PLC时或电源开启时更新
SR1209~SR1219	参数下载至PLC时更新
SR1220~SR1221	参数下载至PLC时或电源开启时更新
SR1222~SR1231	参数下载至PLC时更新
SR1232~SR1233	参数下载至PLC时或电源开启时更新
SR1234~SR1243	参数下载至PLC时更新
SR1244~SR1245	参数下载至PLC时或电源开启时更新
SR1246~SR1255	参数下载至PLC时更新
SR1256~SR1257	参数下载至PLC时或电源开启时更新
SR1258~SR1267	参数下载至PLC时更新
SR1268~SR1269	参数下载至PLC时或电源开启时更新
SR1270~SR1279	参数下载至PLC时更新
SR1280~SR1281	参数下载至PLC时或电源开启时更新

特殊数据寄存器	更新时间
SR1282~SR1291	参数下载至PLC时更新
SR1292~SR1293	参数下载至PLC时或电源开启时更新
SR1294~SR1303	参数下载至PLC时更新
SR1304~SR1305	参数下载至PLC时或电源开启时更新
SR1306~SR1315	参数下载至PLC时更新
SR1316~SR1320	参数下载至PLC时或电源开启时更新
SR1329~SR1330	用户自行设定及清除

A.3 SM/SR 补充说明

1. 扫描逾时定时器

- SM8/SR8

当程序执行时发生扫描逾时 PLC ERROR 错误指示灯常亮，此时 SM8=ON。

SR8 之内容值为 WDT 定时器 ON 之 STEP 地址。

2. 清除警告灯号

- SM22

当 SM22 为 ON 的时候，会将错误纪录清除，以及清除警告的灯号。

3. 万年历

- SM220，SR220~SR226 及 SR391~SR397

SM220: 万年历时钟的±30 秒校正。当 SM220 OFF→ON 触发时作校正。

若万年历时钟秒为 0~29 秒时，分不动，秒归 0。

若万年历时钟秒为 30~59 秒时，分加 1，秒归 0。

万年历时钟 SR220~SR226，SR391~SR397 对应的功能以及内容值如下：

装置		功能名称	内容值范围
BCD 码	十进制		
SR220	SR391	年	0~99 (公元右两位)
SR221	SR392	月	1~12
SR222	SR393	日	1~31
SR223	SR394	时	0~23
SR224	SR395	分	0~59
SR225	SR396	秒	0~59
SR226	SR397	星期	1~7

SR391~SR397 分别对应到 SR220~SR226，不同之处为 SR220~SR226 为 SR391~SR397 的 BCD 值，例如：12 月在 SR392 中为 12，在 SR221 中为 BCD 码 12。

万年历时钟的读取及校正请参考 3.19 节万年历指令。

4. 通讯相关功能

- SM96~SM109，SM209~SM212，SR201~SR202 及 SR209~SR216

SR215，SR216 用来纪录 PLC COM 通讯接口的代号，代号代表的接口功能如下表：

代号	0	1	2
接口功能	RS-232	RS-485	RS-422

若 PLC COM 接口为 RS-485，RS-232，RS-422 时，SR209 纪录 PLC COM1 的通讯格式，SR212

纪录 PLC COM2 的通讯格式。通讯协议设定值设定方式如下表说明。

b0	数据长度		7 (内容值=0)	8 (内容值=1)	
b1 b2	同位		00	: 无 (None)	
			01	: 奇同位 (Odd)	
			10	: 偶同位 (Even)	
b3	Stop bit		1 bit (内容值=0)	2 bits (内容值=1)	
b4	0001	(16#1)	:	4800	
b5	0010	(16#2)	:	9600	
b6	0011	(16#3)	:	19200	
b7	0100	(16#4)	:	38400	
	0101	(16#5)	:	57600	
	0110	(16#6)	:	115200	
	0111	(16#7)	:	230400	RS-232 不支持
	1000	(16#8)	:	460800	RS-232 不支持
	1001	(16#9)	:	921600	RS-232 不支持
b8~b15	无定义 (保留)				

5. 清除装置内容标志

● SM204/SM205

装置编号	被清除的装置
SM204 清除非停电保持区域	非停电保持的 X · Y · S · M · L 接点、 非停电保持的 T · C · HC 内容值及接点、 非停电保持的 D · E 内容值 清除装置约需要 530ms 的时间，此时 WDT 不会动作。
SM205 清除停电保持区域。	停电保持的 T · C · HC 内容值及接点、 停电保持的 M 接点 停电保持的 D 内容值 清除装置约需要 30ms 的时间，此时 WDT 不会动作。

关于详细操作设定请参考 **AH Motion Controller 操作手册** (第四章 ISPSOft 操作介绍)。

6. PLC 错误纪录 (Error Log)

● SR40~SR161

SR40: 储存错误纪录 (Error Log) 的有效组数，最多 20 组，且每一组占用 6 个寄存器。

SR41: 错误纪录的指针，会指向最新一组的错误纪录。当错误发生时，错误纪录指针增加 1。指针的范围值为 0~19，例如：SR41=3 表示第 4 组。

SR42~SR161：纪录各组错误发生的时间及位置，SR 对应的功能说明如下：

组别	背板	插槽	模块ID	错误代码	错误发生时间					
					年	月	日	时	分	秒
1	SR42 高字节	SR42 低字节	SR43	SR44	SR45 高字节	SR45 低字节	SR46 高字节	SR46 低字节	SR47 高字节	SR47 低字节
2	SR48 高字节	SR48 低字节	SR49	SR50	SR51 高字节	SR51 低字节	SR52 高字节	SR52 低字节	SR53 高字节	SR53 低字节
3	SR54 高字节	SR54 低字节	SR55	SR56	SR57 高字节	SR57 低字节	SR58 高字节	SR58 低字节	SR59 高字节	SR59 低字节
4	SR60 高字节	SR60 低字节	SR61	SR62	SR63 高字节	SR63 低字节	SR64 高字节	SR64 低字节	SR65 高字节	SR65 低字节
5	SR66 高字节	SR66 低字节	SR67	SR68	SR69 高字节	SR69 低字节	SR70 高字节	SR70 低字节	SR71 高字节	SR71 低字节
6	SR72 高字节	SR72 低字节	SR73	SR74	SR75 高字节	SR75 低字节	SR76 高字节	SR76 低字节	SR77 高字节	SR77 低字节
7	SR78 高字节	SR78 低字节	SR79	SR80	SR81 高字节	SR81 低字节	SR82 高字节	SR82 低字节	SR83 高字节	SR83 低字节
8	SR84 高字节	SR84 低字节	SR85	SR86	SR87 高字节	SR87 低字节	SR88 高字节	SR88 低字节	SR89 高字节	SR89 低字节
9	SR90 高字节	SR90 低字节	SR91	SR92	SR93 高字节	SR93 低字节	SR94 高字节	SR94 低字节	SR95 高字节	SR95 低字节
10	SR96 高字节	SR96 低字节	SR97	SR98	SR99 高字节	SR99 低字节	SR100 高字节	SR100 低字节	SR101 高字节	SR101 低字节
11	SR102 高字节	SR102 低字节	SR103	SR104	SR105 高字节	SR105 低字节	SR106 高字节	SR106 低字节	SR107 高字节	SR107 低字节
12	SR108 高字节	SR108 低字节	SR109	SR110	SR111 高字节	SR111 低字节	SR112 高字节	SR112 低字节	SR113 高字节	SR113 低字节
13	SR114 高字节	SR114 低字节	SR115	SR116	SR117 高字节	SR117 低字节	SR118 高字节	SR118 低字节	SR119 高字节	SR119 低字节
14	SR120 高字节	SR120 低字节	SR121	SR122	SR123 高字节	SR123 低字节	SR124 高字节	SR124 低字节	SR125 高字节	SR125 低字节
15	SR126 高字节	SR126 低字节	SR127	SR128	SR129 高字节	SR129 低字节	SR130 高字节	SR130 低字节	SR131 高字节	SR131 低字节
16	SR132 高字节	SR132 低字节	SR133	SR134	SR135 高字节	SR135 低字节	SR136 高字节	SR136 低字节	SR137 高字节	SR137 低字节
17	SR138 高字节	SR138 低字节	SR139	SR140	SR141 高字节	SR141 低字节	SR142 高字节	SR142 低字节	SR143 高字节	SR143 低字节
18	SR144 高字节	SR144 低字节	SR145	SR146	SR147 高字节	SR147 低字节	SR148 高字节	SR148 低字节	SR149 高字节	SR149 低字节
19	SR150 高字节	SR150 低字节	SR151	SR152	SR153 高字节	SR153 低字节	SR154 高字节	SR154 低字节	SR155 高字节	SR155 低字节

组别	背板	插槽	模块 ID	错误代码	错误发生时间					
					年	月	日	时	分	秒
20	SR156 高字节	SR156 低字节	SR157	SR158	SR159 高字节	SR159 低字节	SR160 高字节	SR160 低字节	SR161 高字节	SR161 低字节

7. PLC 下载纪录 (Download Log)

● SR227~SR308

SR227：储存下载动作的有效组数，最多 20 组，且每一组占用 4 个寄存器。目前会纪录的下载动作编号为 1~3，如下表说明。

下载动作	动作编号
下载程序	1
下载 PLC 设定	2
下载 I/O 表	3

SR228：下载动作的纪录指针 (Download Log pointer)，会指向最新一组的下载动作纪录。每执行一次下载动作时，下载动作纪录指摺增加 1。指针的范围值为 0~19，例如 SR228=3 表示第 4 组。

SR229~SR308：纪录各组程序下载的时间及动作编号，SR 对应的功能说明如下。

组数	动作编号	*下载动作时间					
		年	月	日	时	分	秒
1	SR229	SR230 高字节	SR230 低字节	SR231 高字节	SR231 低字节	SR232 高字节	SR232 低字节
2	SR233	SR234 高字节	SR234 低字节	SR235 高字节	SR235 低字节	SR236 高字节	SR236 低字节
3	SR237	SR238 高字节	SR238 低字节	SR239 高字节	SR239 低字节	SR240 高字节	SR240 低字节
4	SR241	SR242 高字节	SR242 低字节	SR243 高字节	SR243 低字节	SR244 高字节	SR244 低字节
5	SR245	SR246 高字节	SR246 低字节	SR247 高字节	SR247 低字节	SR248 高字节	SR248 低字节
6	SR249	SR250 高字节	SR250 低字节	SR251 高字节	SR251 低字节	SR252 高字节	SR252 低字节
7	SR253	SR254 高字节	SR254 低字节	SR255 高字节	SR255 低字节	SR256 高字节	SR256 低字节
8	SR257	SR258 高字节	SR258 低字节	SR259 高字节	SR259 低字节	SR260 高字节	SR260 低字节
9	SR261	SR262 高字节	SR262 低字节	SR263 高字节	SR263 低字节	SR264 高字节	SR264 低字节
10	SR265	SR266	SR266	SR267	SR267	SR268	SR268

A3

组 数	动作 编号	*下载动作时间					
		年	月	日	时	分	秒
		高字节	低字节	高字节	低字节	高字节	低字节
11	SR269	SR270 高字节	SR270 低字节	SR271 高字节	SR271 低字节	SR272 高字节	SR272 低字节
12	SR273	SR274 高字节	SR274 低字节	SR275 高字节	SR275 低字节	SR276 高字节	SR276 低字节
13	SR277	SR278 高字节	SR278 低字节	SR279 高字节	SR279 低字节	SR280 高字节	SR280 低字节
14	SR281	SR282 高字节	SR282 低字节	SR283 高字节	SR283 低字节	SR284 高字节	SR284 低字节
15	SR285	SR286 高字节	SR286 低字节	SR287 高字节	SR287 低字节	SR288 高字节	SR288 低字节
16	SR289	SR290 高字节	SR290 低字节	SR291 高字节	SR291 低字节	SR292 高字节	SR292 低字节
17	SR293	SR294 高字节	SR294 低字节	SR295 高字节	SR295 低字节	SR296 高字节	SR296 低字节
18	SR297	SR298 高字节	SR298 低字节	SR299 高字节	SR299 低字节	SR300 高字节	SR300 低字节
19	SR301	SR302 高字节	SR302 低字节	SR303 高字节	SR303 低字节	SR304 高字节	SR304 低字节
20	SR305	SR306 高字节	SR306 低字节	SR307 高字节	SR307 低字节	SR308 高字节	SR308 低字节

*下载时间：时间数据的存为 BCD 格式。时间的内容值范围如下：

功能名称	内容值范围
年	00~99 (公元右两位)
月	01~12
日	01~31
时	00~23
分	00~59
秒	00~59

8. PLC 状态变更纪录 (PLC Status Log)

● SR309~SR390

SR309 : 纪录 PLC 状态变更 (PLC Status Log) 的有效组数，最多 20 组，且每一组占用 4 个寄存器。

目前会纪录的 PLC 状态变更编号为 1~6，如下表说明。

PLC 变更动作	动作编号
上电 (POWER ON)	1
断电 (POWER OFF)	2
PLC 开始运转 (PLC RUN)	3
PLC 停止运转 (PLC STOP)	4
PLC 恢复出厂设定 (1.RST 钮 ; 2.通讯命令)	5
PLC 主机按下 CLR 钮 (清除停电保持装置数据)	6

SR310 : PLC 状态变更的指针 (PLC Status Log Pointer) , 会指向最新一组的 PLC 状态变更的纪录。
当 PLC 的状态被变更一次 , 纪录指针增加 1。指针的范围值为 0~19 , 例如 SR310=3 表示第 4 组。

SR311~SR390 : 纪录各组 PLC 状态变更发生的时间 , SR 对应的功能说明如下 :

组数	动作编号	*PLC 变更时间					
		年	月	日	时	分	秒
1	SR311	SR312 高字节	SR312 低字节	SR313 高字节	SR313 低字节	SR314 高字节	SR314 低字节
2	SR315	SR316 高字节	SR316 低字节	SR317 高字节	SR317 低字节	SR318 高字节	SR318 低字节
3	SR319	SR320 高字节	SR320 低字节	SR321 高字节	SR321 低字节	SR322 高字节	SR322 低字节
4	SR323	SR324 高字节	SR324 低字节	SR325 高字节	SR325 低字节	SR326 高字节	SR326 低字节
5	SR327	SR328 高字节	SR328 低字节	SR329 高字节	SR329 低字节	SR330 高字节	SR330 低字节
6	SR331	SR332 高字节	SR332 低字节	SR333 高字节	SR333 低字节	SR334 高字节	SR334 低字节
7	SR335	SR336 高字节	SR336 低字节	SR337 高字节	SR337 低字节	SR338 高字节	SR338 低字节
8	SR339	SR340 高字节	SR340 低字节	SR341 高字节	SR341 低字节	SR342 高字节	SR342 低字节
9	SR343	SR344 高字节	SR344 低字节	SR345 高字节	SR345 低字节	SR346 高字节	SR346 低字节
10	SR347	SR348 高字节	SR348 低字节	SR349 高字节	SR349 低字节	SR350 高字节	SR350 低字节
11	SR351	SR352 高字节	SR352 低字节	SR353 高字节	SR353 低字节	SR354 高字节	SR354 低字节
12	SR355	SR356	SR356	SR357	SR357	SR358	SR358

组 数	动作 编号	*PLC 变更时间					
		年	月	日	时	分	秒
		高字节	低字节	高字节	低字节	高字节	低字节
13	SR359	SR360 高字节	SR360 低字节	SR361 高字节	SR361 低字节	SR362 高字节	SR362 低字节
14	SR363	SR364 高字节	SR364 低字节	SR365 高字节	SR365 低字节	SR366 高字节	SR366 低字节
15	SR367	SR368 高字节	SR368 低字节	SR369 高字节	SR369 低字节	SR370 高字节	SR370 低字节
16	SR371	SR372 高字节	SR372 低字节	SR373 高字节	SR373 低字节	SR374 高字节	SR374 低字节
17	SR375	SR376 高字节	SR376 低字节	SR377 高字节	SR377 低字节	SR378 高字节	SR378 低字节
18	SR379	SR380 高字节	SR380 低字节	SR381 高字节	SR381 低字节	SR382 高字节	SR382 低字节
19	SR383	SR384 高字节	SR384 低字节	SR385 高字节	SR385 低字节	SR386 高字节	SR386 低字节
20	SR387	SR388 高字节	SR388 低字节	SR389 高字节	SR389 低字节	SR390 高字节	SR390 低字节

*PLC 变更时间：时间数据的存为 BCD 格式。时间的内容值范围如下：

功能名称	内容值范围
年	00~99 (公元右两位)
月	01~12
日	01~31
时	00~23
分	00~59
秒	00~59

9. PLC 运行标志

- SM400~SM403

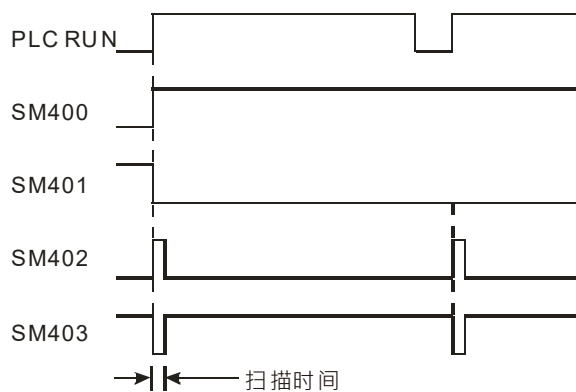
SM400：常开接点。



SM401：常闭接点。

SM402：PLC 开始 RUN 的第一次扫描 ON，之后保持为 OFF。该脉冲的宽度为一次扫描时间，当要作各种初始设定工作时使用本接点。

SM403：PLC 开始 RUN 的第一次扫描 OFF，之后一直 ON。即开始负向（RUN 的瞬间“OFF”）脉冲。



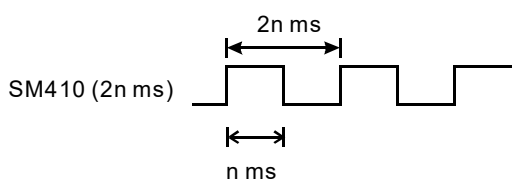
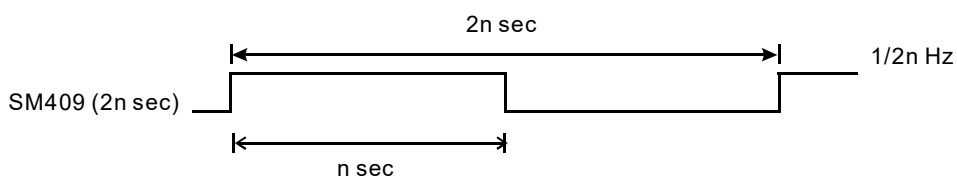
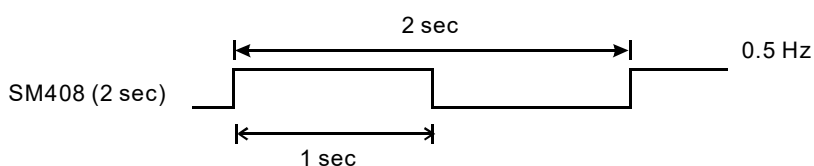
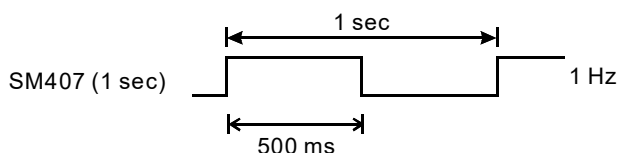
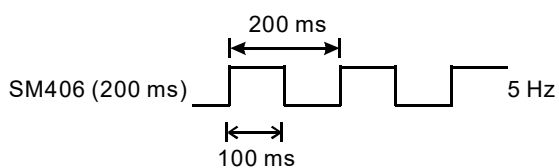
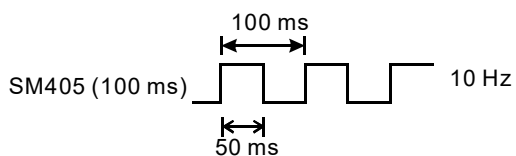
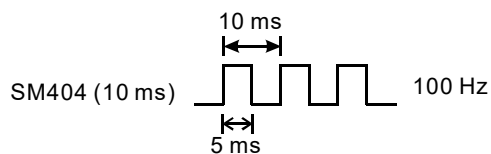
10. 内部时间脉冲

● SM404~SM410 及 SR409~SR410

PLC 内部提供七种时钟脉冲。只要 PLC 通上电源，这七种时钟脉冲会自动动作，其中有两个提供了用户可以自行设定间隔时间的时钟脉冲。

装置	功能说明
SM404	10ms 时钟脉冲，5ms ON/5ms OFF
SM405	100ms 时钟脉冲，50ms ON/50ms OFF
SM406	200ms 时钟脉冲，100ms ON/100ms OFF
SM407	1s 时钟脉冲，0.5s ON/0.5s OFF
SM408	2s 时钟脉冲，1s ON/1s OFF
SM409	2n 秒时钟脉冲，n (秒) 开/n (秒) 关，n 的时间间隔指定于 SR409
SM410	2n 毫秒时钟脉冲，n (毫秒) 开/n (毫秒) 关，n 的时间间隔指定于 SR410

时钟脉冲示意图如下：



A3

11. 储存卡相关标志

- SM450~SM453 · SR453

储存卡主要提供用户备份 PLC 内容。其对应的 SM · SR 功能如下说明

装置	功能说明
SM450	储存卡是否存在标志 · ON : 存在/OFF : 不存在。
SM451	储存卡写保护开关 (Protect Switch) · ON : 写保护/OFF : 无写保护。
SM452	储存卡正被存取 (Accessed) 中 · ON : 存取中/OFF : 无存取。

装置	功能说明
SM453	储存卡运行中有错误发生，ON-表错误发生
SR453	当储存卡有错误发生，错误代码将被记录。

12. I/O 模块相关标志

- SR655~SR730 纪录 I/O 表映像错误及 I/O 模块发生错误。

- SR655~SR730：纪录 I/O 表发生映像错误。

当 I/O 表发生映像错误时，属于此模块的 SR 对应的位会 ON，用以表示此模块发生错误。用户可以读取下列对应的 SR 值，知道哪些背板的插槽发生错误。例如：SR655 的 Bit5 为 ON 表示背板 1 的插槽 5 发生错误。

说明	主背板
	背板 1
装置	SR655
插槽 0	Bit0
插槽 1	Bit1
插槽 2	Bit2
插槽 3	Bit3
插槽 4	Bit4
插槽 5	Bit5
插槽 6	Bit6
插槽 7	Bit7
插槽 8	Bit8
插槽 9	Bit9
插槽 10	Bit10
插槽 11	Bit11

- SR663~SR674：纪录 I/O 表发生映像错误的错误代码。

当 I/O 表发生映像错误，属于此模块的 SR 会记录错误码，用户可以读取下列对应的装置所纪录错误码，得知发生何种错误。各背板插槽对应的 SR 如下：

说明	主背板
	背板 1
插槽 0	SR663
插槽 1	SR664
插槽 2	SR665
插槽 3	SR666
插槽 4	SR667

说明	主背板
	背板 1
插槽 5	SR668
插槽 6	SR669
插槽 7	SR670
插槽 8	SR671
插槽 9	SR672
插槽 10	SR673
插槽 11	SR674

13. 以太网相关标志

SM1090、SM1091、SM1106~SM1109

SM 编号	标志说明	标志作用
SM1090	TCP 联机忙碌	ON 时表示 TCP 联机超时
SM1091	UDP 联机忙碌	ON 时表示 UDP 联机超时
SM1106	以太网网络联机错误	OFF：以太网网络自动协商成功 ON：以太网网络自动协商失败
SM1107	以太网网络基本设定错误	OFF：基本设定正确 ON：基本设定错误
SM1108	以太网网络限定通讯对象设置错误	OFF：限定通讯对象设置正确 ON：限定通讯对象设置错误
SM1109	TCP/UDP 本地端口已被使用	ON 时表示使用到同到的端口

关于标志对应的 LED 灯状态、错误代码及其它详细说明请参考 **AH Motion Controller 操作手册**

14. TCP/UDP Socket 设定

SR1118~SR1320, TCP/UDP Socket 设定，最多可分别设定 8 组 TCP 与 UDP Socket 通道。

SR1118~SR1221 为设定通过 TCP 协议进行数据交换的 Socket 通讯端口及数据范围。TCP Socket 设定对应寄存器 SR，请参考下表：

Socket 编号	1	2	3	4	5	6	7	8
本机通讯端口	SR1118	SR1131	SR1144	SR1157	SR1170	SR1183	SR1196	SR1209
远程 IP H	SR1119	SR1132	SR1145	SR1158	SR1171	SR1184	SR1197	SR1210
远程 IP L	SR1120	SR1133	SR1146	SR1159	SR1172	SR1185	SR1198	SR1211
远程通讯端口	SR1121	SR1134	SR1147	SR1160	SR1173	SR1186	SR1199	SR1212

Socket 编号 项目	1	2	3	4	5	6	7	8
传送长度	SR1122	SR1135	SR1148	SR1161	SR1174	SR1187	SR1200	SR1213
传送地址 H	SR1123	SR1136	SR1149	SR1162	SR1175	SR1188	SR1201	SR1214
传送地址 L	SR1124	SR1137	SR1150	SR1163	SR1176	SR1189	SR1202	SR1215
接收长度	SR1125	SR1138	SR1151	SR1164	SR1177	SR1190	SR1203	SR1216
接收地址 H	SR1126	SR1139	SR1152	SR1165	SR1178	SR1191	SR1204	SR1217
接收地址 L	SR1127	SR1140	SR1153	SR1166	SR1179	SR1192	SR1205	SR1218
保持连接 时间	SR1128	SR1141	SR1154	SR1167	SR1180	SR1193	SR1206	SR1219
传送数据 计数器	SR1129	SR1142	SR1155	SR1168	SR1181	SR1194	SR1207	SR1220
接收数据 计数器	SR1130	SR1143	SR1156	SR1169	SR1182	SR1195	SR1208	SR1221

A3

SR1222~SR1317 为设定通过 UDP 协议进行数据交换的 Socket 通讯端口及数据范围。UDP Socket 设定对应寄存器 SR，请参考下表：

Socket 编号 项目	1	2	3	4	5	6	7	8
本机通讯端 口	SR1222	SR1234	SR1246	SR1258	SR1270	SR1282	SR1294	SR1306
远程 IP H	SR1223	SR1235	SR1247	SR1259	SR1271	SR1283	SR1295	SR1317
远程 IP L	SR1224	SR1236	SR1248	SR1260	SR1272	SR1284	SR1296	SR1318
远程通讯埠	SR1225	SR1237	SR1249	SR1261	SR1273	SR1285	SR1297	SR1309
传送长度	SR1226	SR1238	SR1250	SR1262	SR1274	SR1286	SR1298	SR1310
传送地址 H	SR1227	SR1239	SR1251	SR1263	SR1275	SR1287	SR1299	SR1311
传送地址 L	SR1228	SR1240	SR1252	SR1264	SR1276	SR1288	SR1300	SR1312
接收长度	SR1229	SR1241	SR1253	SR1265	SR1277	SR1289	SR1301	SR1313
接收地址 H	SR1230	SR1242	SR1254	SR1266	SR1278	SR1290	SR1302	SR1314
接收地址 L	SR1231	SR1243	SR1255	SR1267	SR1279	SR1291	SR1303	SR1315
传送数据计 数器	SR1232	SR1244	SR1256	SR1268	SR1280	SR1292	SR1304	SR1316
接收数据计 数器	SR1233	SR1245	SR1257	SR1269	SR1281	SR1293	SR1305	SR1317

详细 Socket 设定说明请参考 3.24 节以太网控制指令。

注：数据长度和计数器数据类型皆为 Byte 。

A.4 标准指令 (依英文字母排序)

- 表格说明

分类	FB/FC	指令码 (位)			P指令	功能
		16	32	64		
A	FC	AND&	DAND&	-	-	接点类型逻辑运算S1&S2
	FC	AND^	DAND^	-	-	接点类型逻辑运算S1^S2
	FC	AND	DAND	-	-	接点类型逻辑运算S1 S2
	FC	AND<	DAND<	-	-	接点类型比较S1<S2

指令表字段说明		
1	分类	指令名称的第一个英文单字
2	FB/FC	FB ：功能块； FC ：函数
3	指令码 (位)	指令名称 (16 /32 /64 位) 32 ：16 位的指令，如有支持 32 位，则在指令前加 D，即为 32 位指令。 64 ：32 位的浮点数指令，如有支持 64 位，则在指令前加 D，即为 64 位浮点数指令。
4	P 指令	是否有 PULSE 指令，有：✓。无：-。 有 PULSE 时，要使用 PULSE 指令，只需在指令后加 P。
5	功能	指令的功能用途说明。

标准指令列表

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
符号	FC	+	D+	-	✓	BIN 加法
	FC	-	D-	-	✓	BIN 减法
	FC	*	D*	-	✓	BIN 乘法
	FC	/	D/	-	✓	BIN 除法
	FC	\$+	-	-	✓	字符串链结
	FC	\$CLR	-	-	✓	字符串清除
	FC	\$DEL	-	-	✓	指定字符串删除
	FC	\$FSTR	-	-	✓	Float→String
	FC	\$FVAL	-	-	✓	String→Float
	FC	\$INS	-	-	✓	字符串插入

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
符号	FC	\$LEFT	—	—	✓	从左边获取字符串
	FC	\$LEN	—	—	✓	计算字符串长度
	FC	\$MIDR	—	—	✓	区段获取字符串
	FC	\$MIDW	—	—	✓	区段字符串取代
	FC	\$MOV	—	—	✓	字符串移动
	FC	\$RIGHT	—	—	✓	从右边获取字符串
	FC	\$RPLC	—	—	✓	字符串取代
	FC	\$SER	—	—	✓	字符串搜寻
	FC	\$STR	D\$STR	—	✓	BIN→String
	FC	\$VAL	D\$VAL	—	✓	String→BIN
A	FC	ABS	DABS	—	✓	绝对值
	FC	ABSD	DABSD	—	✓	绝对方式凸轮控制
	FC	ALT	—	—	—	ON/OFF 交替
	FC	AND\$<	—	—	—	字符串类型比较 $S1 < S2$
	FC	AND\$<=	—	—	—	字符串类型比较 $S1 \leq S2$
	FC	AND\$<>	—	—	—	字符串类型比较 $S1 \neq S2$
	FC	AND\$=	—	—	—	字符串类型比较 $S1 = S2$
	FC	AND\$>	—	—	—	字符串类型比较 $S1 > S2$
	FC	AND\$>=	—	—	—	字符串类型比较 $S1 \geq S2$
	FC	AND&	DAND&	—	—	接点类型逻辑运算 $S1 \& S2$
	FC	AND^	DAND^	—	—	接点类型逻辑运算 $S1 \wedge S2$
	FC	AND	DAND	—	—	接点类型逻辑运算 $S1 S2$
	FC	AND<	DAND<	—	—	接点类型比较 $S1 < S2$
	FC	AND<=	DAND<=	—	—	接点类型比较 $S1 \leq S2$
	FC	AND<>	DAND<>	—	—	接点类型比较 $S1 \neq S2$
	FC	AND=	DAND=	—	—	接点类型比较 $S1 = S2$
	FC	AND>	DAND>	—	—	接点类型比较 $S1 > S$
	FC	AND>=	DAND>=	—	—	接点类型比较 $S1 \geq S2$
	FC	ARWS	—	—	—	箭头键盘输入
	B	FC	B+	DB+	—	✓
FC		B-	DB-	—	✓	BCD 乘法
FC		B*	DB*	—	✓	BCD 除法
FC		B/	DB/	—	✓	BCD 加法
FC		BACOS	—	—	✓	BCD ACOS 运算
FC		BAND	DBAND	—	✓	BIN 16 位和 32 位死区控制

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
B	FC	BASIN	—	—	✓	BCD ASIN 运算
	FC	BATAN	—	—	✓	BCD ATAN 运算
	FC	BCD	DBCD	—	✓	BIN→BCD 变换
	FC	BCDDA	DBCDDA	—	✓	BCD→ASCII 变换
	FC	BCOS	—	—	✓	BCD COS 运算
	FC	BIN	DBIN	—	✓	BCD→BIN 变换
	FC	BINDA	DBINDA	—	✓	有号数十进制→ASCII 变换
	FC	BINHA	DBINHA	—	✓	BIN 十六进制→十六进制 ASCII 变换
	FC	BMOV	—	—	✓	全部传送
	FC	BON	DBON	—	✓	ON 位判定
	FC	BREAK	—	—	✓	强制结束 FOR-NEXT 循环
	FC	BRST	—	—	✓	字软组件的位复位
	FC	BSET	—	—	✓	字软组件的位设定
	FC	BSFL	—	—	✓	n 个位左移 1 个位
	FC	BSFR	—	—	✓	n 个位右移 1 个位
	FC	BSIN	—	—	✓	BCD SIN 运算
	FC	BSQR	DBSQR	—	✓	BCD 开平方根
	FC	BTAN	—	—	✓	BCD TAN 运算
	FC	BXCH	—	—	✓	全部交换
C	FC	CCD	—	—	✓	总和检查
	FC	CHKADR	—	—	—	接点类型指针寄存器地址检查
	FC	CJ	—	—	✓	条件跳转
	FC	COMRS	—	—	—	通讯传送与接收指令
	FC	CML	DCML	—	✓	通讯传送与接收指令
	FC	CMP	DCMP	—	✓	反转传送
	FC	CMPT<	—	—	✓	表格比较<
	FC	CMPT<=	—	—	✓	表格比较<=
	FC	CMPT<>	—	—	✓	表格比较<>
	FC	CMPT=	—	—	✓	表格比较=
	FC	CMPT>	—	—	✓	表格比较>
	FC	CMPT>=	—	—	✓	表格比较>=
	FC	CNT	—	—	—	16 位计数器
	FC	COLM	DCOLM	—	✓	LINE → COLUMN
	FC	CRC	—	—	—	和检查 CRC 模式

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
D	FC	DABCD	DDABCD	—	✓	ASCII→BCD 变换
	FC	DABIN	DDABIN	—	✓	有号数十进制 ASCII→有号数十进制 BIN 变换
	FC	DCNT	—	—	—	32 位计数器
	FC	DEC	DDEC	—	✓	BIN 减一
	FC	DECO	—	—	✓	译码器
	FC	DELAY	—	—	✓	延迟指令
	FC	—	—	DFMOV	✓	64 位浮点数数据移动
	FC	—	—	DFTOF	✓	双精度浮点数→单精度浮点数转换
	FC	DI	—	—	—	中断插入禁止
	FC	—	DIATON	—	✓	字符串→IP 变换
	FC	—	DINTOA	—	✓	IP→字符串变换
	FC	DIS	—	—	✓	16 位数据的 4 位分组
	FC	DIV16	DIV32	—	✓	16 位专用 BIN 除法/ 32 位专用 BIN 除法
	FC	—	DPIDE	—	—	PID 运算
FC	DSW	—	—	—	指拨开关输入	
E	FC	EI	—	—	—	中断插入允许
	FC	EMDRW	—	—	✓	MODBUS TCP 数据读写
	FC	ENCO	—	—	✓	编码器
	FC	EPOP	—	—	✓	指针寄存器读出
	FC	EPUSH	—	—	✓	指针寄存器存入
F	FC	—	F+	DF+	✓	浮点数加法
	FC	—	F-	DF-	✓	浮点数减法
	FC	—	F*	DF*	✓	浮点数乘法
	FC	—	F/	DF/	✓	浮点数除法
	FC	—	FACOS	—	✓	浮点数 ACOS 运算
	FC	—	FAND<	DFAND<	—	浮点数接点类型比较 $S1 < S2$
	FC	—	FAND<=	DFAND<=	—	浮点数接点类型比较 $S1 \leq S2$
	FC	—	FAND<>	DFAND<>	—	浮点数接点类型比较 $S1 \neq S2$
	FC	—	FAND=	DFAND=	—	浮点数接点类型比较 $S1 = S2$
	FC	—	FAND>	DFAND>	—	浮点数接点类型比较 $S1 > S$
	FC	—	FAND>=	DFAND>=	—	浮点数接点类型比较 $S1 \geq S2$
FC	—	FASIN	—	✓	浮点数 ASIN 运算	

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
F	FC	—	FATAN	—	✓	浮点数 ATAN 运算
	FC	—	FCMP	—	✓	浮点数比较
	FC	—	FCOS	—	✓	浮点数 COS 运算
	FC	—	FCOSH	—	✓	浮点数 COSH 运算
	FC	—	FDEG	—	✓	弧度→角度
	FC	—	FEXP	—	✓	浮点数取指数
	FC	—	FINT	DFINT	✓	64 位浮点数→BIN 整数变换
	FC	—	FLD<	DFLD<	—	浮点数接点类型比较 $S1 < S2$
	FC	—	FLD<=	DFLD<=	—	浮点数接点类型比较 $S1 \leq S2$
	FC	—	FLD<>	DFLD<>	—	浮点数接点类型比较 $S1 \neq S2$
	FC	—	FLD=	DFLD=	—	浮点数接点类型比较 $S1 = S2$
	FC	—	FLD>	DFLD>	—	浮点数接点类型比较 $S1 > S$
	FC	—	FLD>=	DFLD>=	—	浮点数接点类型比较 $S1 \geq S2$
	FC	—	FLN	—	✓	浮点数取自然对数
	FC	—	FLOG	—	✓	浮点数取对数
	FC	FLT	DFLT	—	✓	BIN 整数→浮点数变换
	FC	FLTD	DFLTD	—	✓	BIN 整数→64 位浮点数变换
	FC	—	FMOD	—	✓	浮点数转 BCD 浮点数
	FC	—	FNEG	—	✓	32 位浮点正负符号反相
	FC	FOR	—	—	—	循环回路起始
	FC	—	FOR<	DFOR<	—	浮点数接点类型比较 $S1 < S2$
	FC	—	FOR<=	DFOR<=	—	浮点数接点类型比较 $S1 \leq S2$
	FC	—	FOR<>	DFOR<>	—	浮点数接点类型比较 $S1 \neq S2$
	FC	—	FOR=	DFOR=	—	浮点数接点类型比较 $S1 = S2$
	FC	—	FOR>	DFOR>	—	浮点数接点类型比较 $S1 > S$
	FC	—	FOR>=	DFOR>=	—	浮点数接点类型比较 $S1 \geq S2$
	FC	—	FPOW	—	✓	浮点数权值指令
	FC	—	FRAD	—	✓	角度→弧度
	FC	FREXP	—	—	✓	BCD 浮点数转浮点数
	FC	FROM	DFROM	—	✓	特殊模块 CR 数据读出
	FC	—	FSIN	—	✓	浮点数 SIN 运算
	FC	—	FSINH	—	✓	浮点数 SINH 运算
	FC	—	FSQR	—	✓	浮点数开平方根
	FC	—	FTAN	—	✓	浮点数 TAN 运算
FC	—	FTANH	—	✓	浮点数 TANH 运算	

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
	FC	—	FTODF	—	✓	单精度浮点数→双精度浮点数转换
	FC	—	FZCP	—	✓	浮点数区域比较
G	FC	GBIN	DGBIN	—	✓	GRY 码→BIN 变换
	FC	GOEND	—	—	—	跳转到 END
	FC	GPWM	—	—	—	一般用脉冲宽度调变
	FC	GRY	DGRY	—	✓	BIN→GRY 码变换
H	FC	HABIN	DHABIN	—	✓	十六进制 ASCII→十六进制 BIN 变换
	FC	HKY	DHKY	—	—	16 键键盘输入
	FC	HOUR	DHOUR	—	—	运转定时器
II	FC	IMASK	—	—	—	中断控制
	FC	INC	DINC	—	✓	BIN 加一
	FC	INCD	—	—	—	相对方式凸轮控制
	FC	INT	DINT	—	✓	浮点数→BIN 整数变换
J	FC	JMP	—	—	—	无条件跳转
L	FC	LD\$<	—	—	—	字符串类型比较 S1 < S2
	FC	LD\$<=	—	—	—	字符串类型比较 S1 ≤ S2
	FC	LD\$<>	—	—	—	字符串类型比较 S1 ≠ S2
	FC	LD\$=	—	—	—	字符串类型比较 S1 = S2
	FC	LD\$>	—	—	—	字符串类型比较 S1 > S2
	FC	LD\$>=	—	—	—	字符串类型比较 S1 ≥ S2
	FC	LD&	DLD&	—	—	接点类型逻辑运算 S1&S2
	FC	LD^	DLD^	—	—	接点类型逻辑运算 S1^S2
	FC	LD	DLD	—	—	接点类型逻辑运算 S1 S2
	FC	LD<	DLD<	—	—	接点类型比较 S1 < S2
	FC	LD<=	DLD<=	—	—	接点类型比较 S1 ≤ S2
	FC	LD<>	DLD<>	—	—	接点类型比较 S1 ≠ S2
	FC	LD=	DLD=	—	—	接点类型比较 S1 = S2
	FC	LD>	DLD>	—	—	接点类型比较 S1 > S
	FC	LD>=	DLD>=	—	—	接点类型比较 S1 ≥ S2
	FC	LIMIT	DLIMIT	—	✓	限制 BIN 值的范围
	FC	LINE	DLINE	—	✓	COLUMN to LINE
	FC	LRC	—	—	—	和检查 LRC 模式

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
M	FC	MAND	—	—	✓	矩阵及 (AND) 运算
	FC	MBC	—	—	✓	矩阵位状态计数
	FC	MBR	—	—	✓	矩阵位循环
	FC	MBRD	—	—	✓	矩阵位读出
	FC	MBS	—	—	✓	矩阵位位移
	FC	MBWR	—	—	✓	矩阵位写入
	FC	MCMP	—	—	✓	矩阵比较
	FC	MEAN	DMEAN	—	✓	平均值
	FC	MINV	—	—	✓	矩阵反相
	FC	MMOV	—	—	✓	16→32 位数值转换
	FC	MODRW	—	—	—	MODBUS 数据读写
	FC	MOR	—	—	✓	矩阵或 (OR) 运算
	FC	MOV	DMOV	—	✓	数据移动
	FC	MOVB	—	—	✓	多位移动
	FC	MREAD	—	—	—	读取储存卡数据
	FC	MTR	—	—	—	矩阵输入
	FC	MTWRIT	—	—	—	写入字符串至储存卡
	FC	MUL16	MUL32	—	✓	16 位专用 BIN 乘法/ 32 位专用 BIN 乘法
	FC	MWRIT	—	—	—	写入储存卡数据
	N	FC	NEG	DNEG	—	✓
FC		NEXT	—	—	—	循环回路结束
FC		NMOV	DNMOV	—	✓	多点移动
FC		NSFL	—	—	✓	n 个寄存器左移
FC		NSFR	—	—	✓	n 个寄存器右移
O	FC	OR\$<	—	—	—	字符串类型比较 S1 < S2
	FC	OR\$<=	—	—	—	字符串类型比较 S1 ≤ S2
	FC	OR\$<>	—	—	—	字符串类型比较 S1 ≠ S2
	FC	OR\$=	—	—	—	字符串类型比较 S1 = S2
	FC	OR\$>	—	—	—	字符串类型比较 S1 > S2
	FC	OR\$>=	—	—	—	字符串类型比较 S1 ≥ S2
	FC	OR&	DOR&	—	—	接点类型逻辑运算 S1&S2
	FC	OR^	DOR^	—	—	接点类型逻辑运算 S1^S2

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
O	FC	OR	DOR	—	—	接点类型逻辑运算 S1 S2
	FC	OR<	DOR<	—	—	接点类型比较 S1 < S2
	FC	OR<=	DOR<=	—	—	接点类型比较 S1 ≤ S2
	FC	OR<>	DOR<>	—	—	接点类型比较 S1 ≠ S2
	FC	OR=	DOR=	—	—	接点类型比较 S1 = S2
	FC	OR>	DOR>	—	—	接点类型比较 S1 > S
	FC	OR>=	DOR>=	—	—	接点类型比较 S1 ≥ S2
P	FC	—	DPID	—	—	PID 运算
R	FC	RAMP	—	—	—	倾斜信号
	FC	RAND	—	—	✓	随机数值
	FC	RCL	DRCL	—	✓	附进位标志左循环
	FC	RCR	DRCR	—	✓	附进位标志右循环
	FC	REF	—	—	✓	I/O 更新处理
	FC	RMOV	—	—	✓	32→16 位数值转换
	FC	ROL	DROL	—	✓	左循环
	FC	ROR	DROR	—	✓	右循环
	FC	RS	—	—	—	串行数据传输
	FC	RST	—	—	—	接点或寄存器清除
S	FC	SCAL	—	—	✓	比例运算
	FC	SCLP	DSCLP	—	✓	比例运算
	FC	SCLOSE	—	—	✓	关闭 Socket
	FC	SEGD	—	—	✓	7 段显示器解码
	FC	SER	DSER	—	✓	多点比较
	FC	SFCPSE	—	—	—	SFC 暂停
	FC	SFCRUN	—	—	—	SFC 启动
	FC	SFCSTP	—	—	—	SFC 停止
	FC	SFDEL	—	—	✓	删除数据串行中的数据
	FC	SFINS	—	—	✓	插入数据到数据串行中
	FC	SFL	—	—	✓	16 位寄存器位左移
	FC	SFPO	—	—	✓	读出数据串行最新数据
	FC	SFR	—	—	✓	16 位寄存器位右移
	FC	SFRD	—	—	✓	位移读出
	FC	SFTL	—	—	✓	位左移
	FC	SFTR	—	—	✓	位右移
FC	SFWR	—	—	✓	位移写入	

分类	FB/FC	指令码 (位)			P 指令	功能
		16	32	64		
	FC	SORT	DSORT	—	—	数据排序
	FC	SOPEN	—	—	✓	开启 Socket
	FC	SQR	DSQR	—	✓	BIN 开平方根
	FC	SSEND	—	—	✓	通过已开启的 Socket 传送数据
	FC	SRCVD	—	—	✓	通过已开启的 Socket 接收数据
	FC	STMR	—	—	—	特殊定时器
	FC	SUM	DSUM	—	✓	ON 位数量
	FC	SWAP	DSWAP	—	✓	上/下 BYTE 变换
T	FC	T-	—	—	✓	万年历数据减算
	FC	T+	—	—	✓	万年历数据加算
	FC	TCMP	—	—	✓	万年历数据比较
	FC	TIMCHK	—	—	—	时间检查
	FC	TKOFF	—	—	✓	任务 TKS 关闭
	FC	TKON	—	—	✓	任务 TKS 启动
	FC	TKY	DTKY	—	—	10 键键盘输入
	FC	TMR	—	—	—	16 位定时器
	FC	TMRH	—	—	—	16 位定时器
	FC	TO	DTO	—	✓	特殊模块 CR 数据写入
	FC	TRD	—	—	✓	万年历数据读出
	FC	TTMR	—	—	—	教导式定时器
	FC	TWR	—	—	✓	万年历数据写入
	FC	TZCP	—	—	✓	万年历数据区域比较
U	FC	UNI	—	—	✓	16 位数据的 4 位链结
W	FC	WAND	DAND	—	✓	逻辑与 (AND) 运算
	FC	WDT	—	—	✓	逾时监视定时器
	FC	WOR	DOR	—	✓	逻辑或 (OR) 运算
	FC	WSFL	—	—	✓	寄存器左移
	FC	WSFR	—	—	✓	寄存器右移
	FC	WSUM	DWSUM	—	✓	16 位数据的总和计算
	FC	WXNR	DXNR	—	✓	逻辑同或 (XNR) 运算
	FC	WXOR	DXOR	—	✓	逻辑异或 (XOR) 运算
X	FC	XCH	DXCH	—	✓	数据的交换
Z	FC	ZCP	DZCP	—	✓	区域比较
	FC	ZONE	DZONE	—	✓	BIN 16 位和 32 位区域控制
	FC	ZRST	—	—	✓	区域清除

A.5 错误码与故障排除

当发生错误时，可以通过错误码和对应的灯号状态找出解决错误的方式。有关详细的故障排除过程，请参考 **AH Motion Controller 操作手册**。

A.5.1 错误码与对应灯号状态说明

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#000A	扫描逾时 (错误标志SM8)	停止	闪烁	保持
16#000B	PLC程序损毁	停止	常亮	保持
16#000C	下载PLC程序校验错误	停止	闪烁	保持



1



2



3



4

字段简介		
1	错误码	该错误发生时系统所产生的错误代码
2	说明	该错误的说明
3	CPU 状态	该错误发生时，CPU 主机的状态变化。 停止 ：发生该错误时 CPU 停止运行。 持续 ：发生该错误时 CPU 持续运行。 用户自行定义 ：CPU 状态的变化可让用户自行定义。
4	LED 灯号指示	该错误发生时的主机灯号变化。 RUN ：系统执行中 ERROR ：系统错误灯号 BUS FAULT ：I/O 总线错误灯号。 SYSTEM ：系统状态

● 灯号说明

AH Motion Controller CPU 可配合专用背板做为独立 CPU 使用，或是装载在 AH500 系列 PLC 的主机右侧作为模块使用。在两种使用情境下，LED 对应灯号的显示也有所不同，请见下表说明。

模式	灯号	说明
CPU	RUN	指示 CPU 的运行状态 常亮：用户程序执行中 灯灭：用户程序停止中 闪烁：用户程序处于检测错误模式中

模式	灯号	说明
CPU	ERROR	指示 CPU 的错误状态 常亮：系统严重错误发生 灯灭：系统正常 闪烁：系统非严重错误发生
	BUS FAULT	指示 I/O Bus 的错误状态 常亮：I/O Bus 严重错误发生 灯灭：I/O Bus 正常 闪烁：I/O Bus 非严重错误发生
	SYSTEM	指示 CPU 的系统状态 常亮：外部 I/O 被强制锁定 灯灭：系统处于默认状态 闪烁：Reset/Clear 动作执行中
模块	RUN	指示 CPU 的运行状态 常亮：用户程序执行中 灯灭：用户程序停止中 闪烁：用户程序处于检测错误模式中
	ERROR	指示 MODULE 的错误状态。 常亮：MODULE 严重错误发生。 灯灭：MODULE 正常。 闪烁：MODULE 非严重错误发生。

A5

AHxxEMC-5A

将程序写入 AH Motion Controller 系列运动控制器内部后，在不同的程序区块若发生 ERROR，错误指示灯闪烁，错误标志 On，原因可能是指令操作数（装置）使用不合法，程序语法回路错误，或运动参数设定不当，可根据错误寄存器的错误码（16 进制编码）并对照下表，以得知错误原因。

对应的错误信息表

SM：特殊继电器 SR：特殊寄存器	程序错误
	POU
错误标志	-
运算错误	SM0
运算错误锁定	SM1
指令/操作数检查错误	SM5
运算错误的检测错误号码	SR0

SM：特殊继电器 SR：特殊寄存器	程序错误
	POU
32 位运算错误地址锁定	SR1/SR2
文法检查错误的检测错误号码	SR4
32 位文法检查错误的地址锁定	SR5/SR6

错误码和灯号指示说明

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#000A	扫描超时 (错误标志 SM8)	停止	闪烁	保持
16#000B	PLC 程序毁损	停止	常亮	保持
16#000C	下载 PLC 程序校验错误	停止	闪烁	保持
16#000D	CPU 参数毁损	停止	常亮	保持
16#000E	程序或参数下载中 · PLC 无法切换至 RUN	停止	闪烁	保持
16#000F	PLC 原始程序毁损	持续	保持	保持
16#0010	CPU 内存存取被拒	停止	常亮	保持
16#0011	PLC ID 错误 (错误标志 SM9)	持续	常亮	保持
16#0012	PLC 密码错误 (错误标志 SM9)	持续	常亮	保持
16#0013	I/O 模块无法设置运行/停止 (错误标志 SM10)	停止	保持	常亮
16#0014	无法执行系统复制程序 (错误标志 SM9 ; 此问题发生时 · ERROR&Bus Fault 灯会维持常亮)	停止	常亮	常亮
16#0015	模块配置数据错误 (错误标志 SM10)	停止	常亮	保持
16#0016	模块设定数据错误 (错误标志 SM10)	停止	常亮	保持
16#0017	D 对应装置设定错误 (错误标志 SM10)	停止	常亮	保持
16#0018	串行端口异常 (错误标志 SM9)	持续	闪烁	保持
16#0019	USB 异常 (错误标志 SM9)	持续	闪烁	保持
16#001A	系统备份文件 (DUP) 内容错误	持续	闪烁	保持
16#001B	定时中断 (编号 0) 时间设置错误	停止	常亮	保持
16#001C	定时中断 (编号 1) 时间设置错误	停止	常亮	保持
16#001D	定时中断 (编号 2) 时间设置错误	停止	常亮	保持
16#001E	定时中断 (编号 3) 时间设置错误	停止	常亮	保持
16#001F	程序扫描超时定时器设置错误	停止	常亮	保持
16#0020	固定扫描时间设置错误	停止	常亮	保持
16#0021	固定扫描时间设置错误	停止	常亮	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#0022	下载 CPU 模块参数校验错误	停止	常亮	保持
16#0023	PLC 参数设定 · Y 装置状态 (STOP -> RUN) 设定错误	停止	常亮	保持
16#0024	背板无 IO 模块	持续	保持	保持
16#0026	通讯能力占用扫描时间比率设定错误	停止	常亮	保持
16#0027	M 装置停电保持区范围设定错误	停止	常亮	保持
16#0028	D 装置停电保持区范围设定错误	停止	常亮	保持
16#0029	T 装置停电保持区范围设定错误	停止	常亮	保持
16#002A	C 装置停电保持区范围设定错误	停止	常亮	保持
16#002B	HC 装置停电保持区范围设定错误	停止	常亮	保持
16#0033	COM 1 通讯设置设置错误 (错误标志 SM9)	持续	闪烁	保持
16#0034	COM 1 站号设置错误 (错误标志 SM9)	持续	闪烁	保持
16#0035	COM 1 传输方式设置错误 (错误标志 SM9)	持续	闪烁	保持
16#0038	COM 2 通讯设置设置错误 (错误标志 SM9)	持续	闪烁	保持
16#0039	COM 2 站号设置错误 (错误标志 SM9)	持续	闪烁	保持
16#003A	COM 2 传输方式设置错误 (错误标志 SM9)	持续	闪烁	保持
16#0050	停电保持区 SM 储存区块异常	持续	常亮	保持
16#0051	停电保持区 SR 寄存器异常	持续	常亮	保持
16#0052	停电保持区 M 储存区块异常	持续	常亮	保持
16#0053	停电保持区 T 寄存器异常	持续	常亮	保持
16#0054	停电保持区 C 寄存器异常	持续	常亮	保持
16#0055	停电保持区 HC 寄存器异常	持续	常亮	保持
16#0056	停电保持区 T 储存区块异常	持续	常亮	保持
16#0057	停电保持区 C 储存区块异常	持续	常亮	保持
16#0058	停电保持区 HC 储存区块异常	持续	常亮	保持
16#0059	停电保持区 D 寄存器异常	持续	常亮	保持
16#005A	停电保持区 W 寄存器异常	持续	常亮	保持
16#005E	储存卡的初始程序错误 (错误标志 SM453)	持续	保持	保持
16#005F	在储存卡中 · 欲读取不存在的文件 · 或写入不存在路径的文件。 (错误标志 SM453)	持续	保持	保持
16#0061	储存卡容量不足 (错误标志 SM453)	持续	保持	保持
16#0062	储存卡为写保护模式 (错误标志 SM453)	持续	保持	保持
16#0063	数据写入储存卡的文件时有错误 (错误标志 SM453)	持续	保持	保持
16#0064	储存卡的文件无法被读取 (错误标志 SM453)	持续	保持	保持
16#0065	储存卡中的文件为只读状态 (错误标志 SM453)	持续	保持	保持
16#0066	系统备份时错误	持续	闪烁	保持
16#0067	系统还原的系统参数长度超出 CPU 模块的系统参数长度	持续	闪烁	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#1401	模块存取错误 (错误标志 SM9)	停止	保持	常亮
16#1402	实际模块不符合配置设定 (错误标志 SM9)	停止	保持	常亮
16#1403	从模块读取数据错误 (错误标志 SM9)	停止	保持	常亮
16#1405	搜寻不到 I/O 模块的设定参数 (错误标志 SM9)	停止	保持	常亮
16#140B	通讯模块数量超过上限 (错误标志 SM9)	停止	保持	常亮
16#140C	高速数据交换检查码错误	停止	保持	常亮
16#140D	实际 Power ID 不符合配置设定	停止	保持	常亮
16#140E	模块高速数据交换数量超出支持最大范围	停止	保持	常亮
16#140F	高速数据交换错误	停止	保持	常亮
16#1801	CPU 模块未设定中断工作	停止	保持	常亮
16#2000	PLC 程序无 END 指令 (错误标志 SM5)	停止	闪烁	保持
16#2001	项目程序内容有误：程序语法错误	停止	闪烁	保持
16#2002	GOEND 使用的地方错误 (错误标志 SM5)	停止	闪烁	保持
16#2003	程序中使用的装置超过可用范围 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#2004	CJ/JMP 指令跳转的 P 地址错误，或是 P 装置重复使用。(错误标志 SM0/SM5)	停止	闪烁	保持
16#2005	MC/MCR 相对应的 N 值不同，或数量不一样多(错误标志 SM5)	停止	闪烁	保持
16#2006	n 不是从 0 开始或是 n 的值不连续 (错误标志 SM5)	停止	闪烁	保持
16#2007	ZRST 指令操作数使用不当 (错误标志 SM5)	停止	闪烁	保持
16#200A	无效的指令 (错误标志 SM5)	停止	闪烁	保持
16#200B	n 操作数或其它 K/H 操作数超出范围 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#200C	部份指令不允许操作数发生重迭 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#200D	BIN 转成 BCD 时发生错误 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#200E	字符串没有 0x00 当做结尾 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#200F	指令不支持 E 装置修饰 (错误标志 SM5)	停止	闪烁	保持
16#2010	指令不支持该装置类别/编码错误/16 位指令但 K·H 却是 32 位的编码 (错误标志 SM5)	停止	闪烁	保持
16#2011	操作数的数目错误 (错误标志 SM5)	停止	闪烁	保持
16#2012	除法运算错误 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#2013	浮点数格式错误，超出可转换范围 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#2014	TKON/TKOFF 指令所指定的 TASK 编号错误或超出范围 (错误标志 SM5)	停止	闪烁	保持
16#2015	CALL 指令超过 32 层 (错误标志 SM0)	自定义	闪烁	保持
16#2016	FOR-NEXT 指令超过 32 层 (错误标志 SM0/SM5)	自定义	闪烁	保持
16#2017	FOR 跟 NEXT 的指令数目不同 (错误标志 SM5)	停止	闪烁	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#2018	在 FEND 之后的 P 指针没有相对应的 SRET · 或是有 SRET 但没有 P 指针 (错误标志 SM5)	停止	闪烁	保持
16#2019	Interrupt I 的地址不是在 FEND 之后 (错误标志 SM5)	停止	闪烁	保持
16#201A	IRET/SRET 的地址不是在 FEND 之后 (错误标志 SM5)	停止	闪烁	保持
16#201B	I 没有相对应的 IRET · 或是有 IRET 但没有 I (错误标志 SM5)	停止	闪烁	保持
16#201C	END 指令不是在程序的最后一个地址 (错误标志 SM5)	停止	闪烁	保持
16#201D	有 call 指令但没有 MAR 指令 (错误标志 SM5)	停止	闪烁	保持
16#201E	MODRW 指令中的功能代码错误 (错误标志 SM102/103)	自定义	闪烁	保持
16#201F	MODRW 指令中的数据长度错误 (错误标志 SM102/103)	自定义	闪烁	保持
16#2020	MODRW 的回复命令错误 (错误标志 SM102/103)	自定义	闪烁	保持
16#2021	MODRW 回复命令的校验和 (Checksum) 错误 (错误标志 SM102/103)	自定义	闪烁	保持
16#2022	MODRW 指令的命令不符合 ASCII 格式 (错误标志 SM102/103)	自定义	闪烁	保持
16#2023	MODRW 指令的通讯超时 (错误标志 SM102/103)	自定义	闪烁	保持
16#2024	RS 指令的通讯超时数值无效 (错误标志 SM102/103)	自定义	闪烁	保持
16#2025	RS 指令的通讯超时 (错误标志 SM102/103)	自定义	闪烁	保持
16#2026	RS 指令的中断指针错误	自定义	闪烁	保持
16#2027	FWD 应用指令异常	自定义	闪烁	保持
16#2028	REV 应用指令异常	自定义	闪烁	保持
16#2029	STOP 应用指令异常	自定义	闪烁	保持
16#202A	RS DT 应用指令异常	自定义	闪烁	保持
16#202B	RSTEF 应用指令异常	自定义	闪烁	保持
16#202C	IO 中断服务程序 0 不存在	停止	闪烁	保持
16#202D	IO 中断服务程序 1 不存在	停止	闪烁	保持
16#202E	IO 中断服务程序 2 不存在	停止	闪烁	保持
16#202F	IO 中断服务程序 3 不存在	停止	闪烁	保持
16#2030	IO 中断服务程序 4 不存在	停止	闪烁	保持
16#2031	IO 中断服务程序 5 不存在	停止	闪烁	保持
16#2032	IO 中断服务程序 6 不存在	停止	闪烁	保持
16#2033	IO 中断服务程序 7 不存在	停止	闪烁	保持
16#2034	IO 中断服务程序 8 不存在	停止	闪烁	保持
16#2035	IO 中断服务程序 9 不存在	停止	闪烁	保持
16#2036	IO 中断服务程序 10 不存在	停止	闪烁	保持
16#2037	IO 中断服务程序 11 不存在	停止	闪烁	保持
16#2038	IO 中断服务程序 12 不存在	停止	闪烁	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#2039	IO 中断服务程序 13 不存在	停止	闪烁	保持
16#203A	IO 中断服务程序 14 不存在	停止	闪烁	保持
16#203B	IO 中断服务程序 15 不存在	停止	闪烁	保持
16#203C	IO 中断服务程序 16 不存在	停止	闪烁	保持
16#203D	IO 中断服务程序 17 不存在	停止	闪烁	保持
16#203E	IO 中断服务程序 18 不存在	停止	闪烁	保持
16#203F	IO 中断服务程序 19 不存在	停止	闪烁	保持
16#2040	IO 中断服务程序 20 不存在	停止	闪烁	保持
16#2041	IO 中断服务程序 21 不存在	停止	闪烁	保持
16#2042	IO 中断服务程序 22 不存在	停止	闪烁	保持
16#2043	IO 中断服务程序 23 不存在	停止	闪烁	保持
16#2044	IO 中断服务程序 24 不存在	停止	闪烁	保持
16#2045	IO 中断服务程序 25 不存在	停止	闪烁	保持
16#2046	IO 中断服务程序 26 不存在	停止	闪烁	保持
16#2047	IO 中断服务程序 27 不存在	停止	闪烁	保持
16#2048	IO 中断服务程序 28 不存在	停止	闪烁	保持
16#2049	IO 中断服务程序 29 不存在	停止	闪烁	保持
16#204A	IO 中断服务程序 30 不存在	停止	闪烁	保持
16#204B	IO 中断服务程序 31 不存在	停止	闪烁	保持
16#2054 16#2127	外部中断服务程序 40 不存在 外部中断服务程序 251 不存在	停止	闪烁	保持
16#2128	SFC Action 时间属性设定错误 (错误标志 SM0/SM1)	自定义	闪烁	保持
16#2129	SFC Action 重置属性设置错误 (错误标志 SM0/SM1)	自定义	闪烁	保持
16#3040	凸轮点数据超过范围或不存在	持续	闪烁	保持
16#3100	功能块输入引脚超出规定范围	持续	闪烁	保持
16#3102	功能块内部的子功能块发生错误	持续	闪烁	保持
16#3103	空包功能块两检测 seneor 间距为负值	持续	闪烁	保持
16#3104	相位移功能尚未准备完成	持续	闪烁	保持
16#3105	送加功能尚未准备完成	持续	闪烁	保持
16#3106	在前次链钩修正尚未完成时, 重复触发修正	持续	闪烁	保持
16#3107	在前次膜轴修正尚未完成时, 重复触发修正	持续	闪烁	保持
16#3108	在前次切刀修正尚未完成时, 重复触发修正	持续	闪烁	保持
16#3400	轴设定错误	持续	保持	保持
16#3401	SDODataType 设定错误 (0~199)	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#3404	通道编号超过该功能通道数	持续	保持	保持
16#3405	运转速度小于或等于零	持续	闪烁	保持
16#340A	原点模式设定错误	持续	闪烁	保持
16#340B	轴设定位置为 0	持续	闪烁	保持
16#3410	坐标系统或输出脉冲形式设定错误	持续	闪烁	保持
16#3411	速度%错误	持续	闪烁	保持
16#3414	计数脉冲型式设定错误	持续	保持	保持
16#3415	比较条件设定错误	持续	保持	保持
16#3419	主轴位置设定小于零	持续	闪烁	保持
16#341B	最大速度设定错误	持续	闪烁	保持
16#3430	GroupNum 重复	持续	闪烁	保持
16#3431	Group 轴号重复	持续	闪烁	保持
16#3432	GroupNum 不存在	持续	闪烁	保持
16#3433	Group 轴数不足	持续	闪烁	保持
16#3434	轴组运行时启动 DFB_GroupDisable 功能块	持续	闪烁	保持
16#3435	Group 中互设同一轴	持续	闪烁	保持
16#3436	GroupNum 起始轴编号是零	持续	闪烁	保持
16#3437	GroupNum 设定超过范围	持续	闪烁	保持
16#3438	GroupNum 在 ErrorStop 状态中	持续	闪烁	保持
16#343A	Group 被执行 ImmediateStop 功能块	持续	闪烁	保持
16#343B	Group 中其它轴发生问题	持续	闪烁	保持
16#3461	PDO 设定缺少必要的传输参数	持续	闪烁	保持
16#3463	轴所指定的 ECAT Slave 不存在	持续	闪烁	保持
16#3500	轴状态不在 Disable or Standstill 或轴状态在 Coordinated 且轴组状态不再 Standby	持续	闪烁	保持
16#3501	功能块 Channel 重复使用	持续	闪烁	保持
16#3502	不允许位置设定	持续	闪烁	保持
16#3505	Cam 点数据写入失败	持续	闪烁	保持
16#3506	轴在 Coordinated	持续	闪烁	保持
16#3507	轴在 ErrorStop	持续	闪烁	保持
16#3508	轴不在 StandStill 状态	持续	闪烁	保持
16#3509	轴在 Stopping	持续	闪烁	保持
16#350B	轴加速时间太短	持续	闪烁	保持
16#350C	轴减速时间太短	持续	闪烁	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#350D	读取凸轮数据长度超过范围	持续	闪烁	保持
16#350E	写入凸轮数据长度超过范围	持续	闪烁	保持
16#350F	轴在 Synchronized	持续	闪烁	保持
16#3510	凸轮曲线直线的速度设定错误	持续	闪烁	保持
16#3511	凸轮曲线直线的加速度设定错误	持续	闪烁	保持
16#3512	Cam 点数据不存在	持续	保持	保持
16#3513	设定写入主轴位置小于前一个点	持续	闪烁	保持
16#3514	设定凸轮百分比超过范围 (0~100)	持续	闪烁	保持
16#3526	轴前一个运动过程发生错误	持续	闪烁	保持
16#3600	错误的状态机转换	持续	闪烁	保持
16#3601	Buffer mode 的 buffer 数量已满	持续	闪烁	保持
16#3602	执行多个无法同时启动功能	持续	闪烁	保持
16#3603	Buffermode 设置参数错误	持续	闪烁	保持
16#3604	功能块运动方向错误	持续	闪烁	保持
16#3605	轴组或是单轴相关运动功能块参数设定错误	持续	闪烁	保持
16#3606	运动目标距离为 0	持续	闪烁	保持
16#3607	目标速度超出范围	持续	闪烁	保持
16#3608	目标速度超出范围	持续	闪烁	保持
16#3612	到达正向极限	持续	闪烁	保持
16#3613	到达反向极限	持续	闪烁	保持
16#3614	超出伺服极限	持续	闪烁	保持
16#3617	加速度超出范围	持续	闪烁	保持
16#3618	减速度超出范围	持续	闪烁	保持
16#3619	站号不存在	持续	闪烁	保持
16#3620	SDO 排程缓冲区已满	持续	闪烁	保持
16#3622	SDO OD 数据类型不合	持续	闪烁	保持
16#3623	SDO 超时	持续	闪烁	保持
16#3624	SDO 数据写入错误	持续	闪烁	保持
16#3625	SDO 数据读出错误	持续	闪烁	保持
16#3626	SDO Retry 次数设定超过范围	持续	闪烁	保持
16#3630	AxisPara 设定错误	持续	闪烁	保持
16#3631	AxisPara.GearRatioNume / GearRatioDeno 无法设定到伺服	持续	闪烁	保持
16#3635	此轴参数不可修改	持续	闪烁	保持
16#3636	此轴参数设定错误	持续	闪烁	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#3637	AxisPara.SWPosLimit 与 AxisPara.SWNegLimit 包含的范围有误 (两者皆为 0 导致范围为 0 ; 范围未包含目前指令位置)	持续	闪烁	保持
16#3638	AxisPara.RotaryMaxPos 与 AxisPara.RotaryMinPos 包含的范围有误 (两者皆为 0 导致范围为 0)	持续	闪烁	保持
16#3639	在轴非 Disabled 时不可修改 GearRatio	持续	闪烁	保持
16#3800	运行过程中轴断线	持续	闪烁	保持
16#3801	EtherCAT Axis 发生错误	持续	闪烁	保持
16#3901	网络重新启动失败	持续	闪烁	保持
16#3904	SDO 无法读取该参数	持续	闪烁	保持
16#3905	SDO 无法写入该数值	持续	闪烁	保持
16#3906	MaxTorque 写入失败	持续	闪烁	保持
16#3907	虚拟轴不支持此功能	持续	闪烁	保持
16#3909	目前网络正在执行其它动作	持续	闪烁	保持
16#390C	运动过程中轴发生错误	持续	闪烁	保持
16#3910	在未啮合状态下取消啮合	持续	闪烁	保持
16#3911	轴软极限错误	持续	闪烁	保持
16#3912	功能块位置输入引脚超出旋转轴范围	持续	闪烁	保持
16#3913	同步运动啮合失败	持续	闪烁	保持
16#3914	GearInPos 速度设定太小	持续	闪烁	保持
16#3915	GearInPos 急跳度设定太小	持续	闪烁	保持
16#3916	GearInPos 啮合时间太短	持续	闪烁	保持
16#3917	GearInPos 开始啮合时主轴速度为 0	持续	闪烁	保持
16#3918	啮合速度大于轴最高速度	持续	闪烁	保持
16#3919	GearInPos 主轴运转方向相反	持续	闪烁	保持
16#3920	GearInPos 加速度设定太小	持续	闪烁	保持
16#3921	GearInPos MasterStartDistancs 设定超出范围	持续	闪烁	保持
16#3922	GearInPos 啮合位移量过小	持续	闪烁	保持
16#3923	GearInPos 啮合位移量过大	持续	闪烁	保持
16#3924	GearInPos 开始啮合时主轴速度有变化	持续	闪烁	保持
16#3950	Capture 无法工作在脉冲速度大于 1MHz	持续	闪烁	保持
16#3951	DFB_CamCurve2 的参数设定有误	持续	闪烁	保持
16#3953	Capture2 功能重复使用同个 Channel 编号	持续	闪烁	保持
16#3954	Torque Slope 写入失败	持续	闪烁	保持
16#3955	Torque Velocity 写入失败	持续	闪烁	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#3A00	CamIn 的 CAM Table 设定错误	持续	闪烁	保持
16#3A01	CamIn 的 Master 设定错误	持续	闪烁	保持
16#3A02	CamIn 太早更换 Cam 表	持续	闪烁	保持
16#3A03	CamIn Activation Mode 设定超出范围	持续	闪烁	保持
16#3A04	CamIn Start Mode 设定超出范围	持续	闪烁	保持
16#3A05	CamIn 的 Master Scaling 数值设定为 0.0	持续	闪烁	保持
16#3A06	CamIn 的 Slave Scaling 数值设定为 0.0	持续	闪烁	保持
16#3A07	CamIn 的 MasterSyncPosition 设定错误	持续	闪烁	保持
16#3A08	CamIn 的 ActivationPosition 设定错误	持续	闪烁	保持
16#3A10	CamIn 的 Master Start Position 设定太小	持续	闪烁	保持
16#3A11	CamIn 开始咬合时主轴速度有变化	持续	闪烁	保持
16#3A12	CamIn 的 Start Mode 中所需的速度太小	持续	闪烁	保持
16#3A13	CamIn 的 Start Mode 中所需的速度太小	持续	闪烁	保持
16#3A14	CamIn 的 Start Mode 中所需的啮合时间太短	持续	闪烁	保持
16#3A15	CamIn 的 Start Mode 中所需的急跳度太小	持续	闪烁	保持
16#3A16	CamIn 的 Start Mode 中所需的最大加速度太小	持续	闪烁	保持
16#3A17	CamIn 的 Start Mode 中所需的开始距离太小	持续	闪烁	保持
16#3A18	CamIn 的 Start Mode 中所需的开始距离太大	持续	闪烁	保持
16#3A19	同一轴同时执行太多 CamIn	持续	闪烁	保持
16#3A20	Master 运行方向为负向运动	持续	闪烁	保持
16#3A21	在 MC_CamIn 未启动状态下触发 MC_CamOut	持续	闪烁	保持
16#3D00	EtherCAT ENI 文件与目前硬件配置不合	持续	闪烁	保持
16#3D01	EtherCAT slave 遗失	持续	闪烁	保持
16#3D03	EtherCAT DC 时间设定过小(小于支持轴数的最小扫描时间)	持续	闪烁	保持
16#6001	IP 地址不合法	持续	闪烁	保持
16#6002	网络屏蔽地址不合法	持续	闪烁	保持
16#6003	网关地址不合法	持续	闪烁	保持
16#6004	以太网络的 IP 地址过滤设置错误	持续	闪烁	保持
16#6006	以太网络的静态 ARP 表设置错误	持续	闪烁	保持
16#600D	RJ45 端口未连接 (错误标志 SM1100)	持续	保持	保持
16#620D	UDP 通讯接口 (Socket) 功能的传送数据长度不合法	持续	保持	保持
16#6212	远程装置响应超时	持续	保持	保持
16#6213	接收数据超过限制	持续	保持	保持
16#6214	远程装置拒绝联机	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#6400	EMDRW 指令操作的联机数超出限制或未设定送信标志	持续	保持	保持
16#6401	远程装置中止联机	持续	保持	保持
16#6402	远程装置响应超时	持续	保持	保持
16#6403	API 指令的远程 IP 地址不合法	持续	保持	保持
16#6404	收到不支持的 MODBUS 功能代码	持续	保持	保持
16#6405	MODBUS 回复信息的 Byte Count 与实际的数据长度不符	持续	保持	保持
16#6501	远程装置响应超时 (错误标志 SM828-SM955)	持续	OFF	OFF
16#6502	远程装置回复报文错误 (错误标志 SM828-SM955)	持续	OFF	OFF
16#6700	MODBUS TCP 数据交换初始化错误	持续	保持	保持
16#6701	MODBUS TCP 数据交换超时	持续	保持	保持
16#6702	MODBUS TCP 数据交换接收错误	持续	保持	保持
16#7002	CPU 模块不支持此功能	持续	保持	保持
16#7203	无效的访问代码 (Access Code)	持续	保持	保持
16#7401	功能码 (Function Code) 错误	持续	保持	保持
16#7402	报文超出最大数据长度	持续	保持	保持
16#7407	命令中包含非 ASCII 字符	持续	保持	保持
16#7408	PLC 处于运行 (RUN) 模式	持续	保持	保持
16#740A	主机内存正在写入或写入失败	持续	保持	保持
16#740B	清除或重置动作正在进行中	持续	保持	保持
16#740C	通讯命令中的背板编号不正确	持续	保持	保持
16#740D	通讯命令中的插槽编号不正确	持续	保持	保持
16#740E	清除内存的过程发生错误	持续	保持	保持
16#740F	通讯超时	持续	保持	保持
16#7410	回复命令的功能码 (Function Code) 不一致	持续	保持	保持
16#7412	因 SW1 ON 所以数据无法下载至 CPU 模块	持续	保持	保持
16#757D	输入 PLC 密码的剩余次数为 0	持续	保持	保持
16#757E	输入的 PLC 密码错误	持续	保持	保持
16#8105	下载的项目程序内容有误：下载的程序语法错误	持续	保持	保持
16#8106	下载的项目程序内容有误：执行码超过限制长度	持续	保持	保持
16#8107	下载的项目程序内容有误：原始码超过限制长度	持续	保持	保持
16#8230	下载的主机参数有误：IP 地址不合法	持续	保持	保持
16#8231	下载的主机参数有误：网络屏蔽地址不合法	持续	保持	保持
16#8232	下载的主机参数有误：网关地址不合法	持续	保持	保持
16#8233	下载的主机参数有误：IP 地址过滤设定错误	持续	保持	保持
16#8235	下载的主机参数有误：静态 ARP 表错误	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#8236	下载的主机参数有误：NTP 设定错误	持续	保持	保持
16#8240	下载的主机参数有误：Ether iLink	持续	保持	保持
16#8242	Gcode Data ERROR	持续	闪烁	保持
16#8243	ECAM Data ERROR	持续	闪烁	保持
16#8244	ENI Data ERROR	持续	闪烁	保持
16#8245	EtherCat Data ERROR	持续	闪烁	保持
16#8246	Axes Parameters Data ERROR	持续	闪烁	保持
16#8247	External Gcode Data ERROR	持续	闪烁	保持
16#8522	自动扫描检测执行中	持续	保持	保持
16#853B	IO 模块未配置	持续	保持	保持
16#853C	IO 模块不存在	持续	保持	保持
16#854B	IO 模块未配置	持续	保持	保持
16#854C	IO 模块不存在	持续	保持	保持
16#8572	模块配置表检查码错误	持续	保持	保持
16#8576	模块参数设定检查码错误	持续	保持	保持
16#857A	模块参数映像表检查码错误	持续	保持	保持
16#85E1	IO 中断编号不正确	持续	保持	保持
16#85E2	IO 中断服务程序不存在	持续	保持	保持
16#860F	系统还原错误	持续	保持	保持
16#8611	储存卡不存在·或储存卡格式错误	持续	保持	保持
16#9A33	PLC Link / COM1 MODBUS 从站 19 通讯错误	持续	保持	保持
16#9A34	PLC Link / COM1 MODBUS 从站 20 通讯错误	持续	保持	保持
16#9A35	PLC Link / COM1 MODBUS 从站 21 通讯错误	持续	保持	保持
16#9A47	PLC Link / COM1 MODBUS 从站 7 无回应 (错误标志 SM1591)	持续	保持	保持
16#9B01	COM2 Modbus 初始化错误	持续	保持	保持
16#9B21	COM2 MODBUS 从站 1 通讯错误	持续	保持	保持
16#9B22	COM2 MODBUS 从站 2 通讯错误	持续	保持	保持
16#9B23	COM2 MODBUS 从站 3 通讯错误	持续	保持	保持
16#9B24	COM2 MODBUS 从站 4 通讯错误	持续	保持	保持
16#9B25	COM2 MODBUS 从站 5 通讯错误	持续	保持	保持
16#9B26	COM2 MODBUS 从站 6 通讯错误	持续	保持	保持
16#9B27	COM2 MODBUS 从站 7 通讯错误	持续	保持	保持
16#9B28	COM2 MODBUS 从站 8 通讯错误	持续	保持	保持
16#9B29	COM2 MODBUS 从站 9 通讯错误	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9B2A	COM2 MODBUS 从站 10 通讯错误	持续	保持	保持
16#9B2B	COM2 MODBUS 从站 11 通讯错误	持续	保持	保持
16#9B2C	COM2 MODBUS 从站 12 通讯错误	持续	保持	保持
16#9B2D	COM2 MODBUS 从站 13 通讯错误	持续	保持	保持
16#9B2E	COM2 MODBUS 从站 14 通讯错误	持续	保持	保持
16#9B2F	COM2 MODBUS 从站 15 通讯错误	持续	保持	保持
16#9B30	COM2 MODBUS 从站 16 通讯错误	持续	保持	保持
16#9B31	COM2 MODBUS 从站 17 通讯错误	持续	保持	保持
16#9B32	COM2 MODBUS 从站 18 通讯错误	持续	保持	保持
16#9B33	COM2 MODBUS 从站 19 通讯错误	持续	保持	保持
16#9B34	COM2 MODBUS 从站 20 通讯错误	持续	保持	保持
16#9B35	COM2 MODBUS 从站 21 通讯错误	持续	保持	保持
16#9B36	COM2 MODBUS 从站 22 通讯错误	持续	保持	保持
16#9B37	COM2 MODBUS 从站 23 通讯错误	持续	保持	保持
16#9B38	COM2 MODBUS 从站 24 通讯错误	持续	保持	保持
16#9B39	COM2 MODBUS 从站 25 通讯错误	持续	保持	保持
16#9B3A	COM2 MODBUS 从站 26 通讯错误	持续	保持	保持
16#9B3B	COM2 MODBUS 从站 27 通讯错误	持续	保持	保持
16#9B3C	COM2 MODBUS 从站 28 通讯错误	持续	保持	保持
16#9B3D	COM2 MODBUS 从站 29 通讯错误	持续	保持	保持
16#9B3E	COM2 MODBUS 从站 30 通讯错误	持续	保持	保持
16#9B3F	COM2 MODBUS 从站 31 通讯错误	持续	保持	保持
16#9B40	COM2 MODBUS 从站 32 通讯错误	持续	保持	保持
16#9B41	COM2 MODBUS 从站 1 无回应	持续	保持	保持
16#9B42	COM2 MODBUS 从站 2 无回应	持续	保持	保持
16#9B43	COM2 MODBUS 从站 3 无回应	持续	保持	保持
16#9B44	COM2 MODBUS 从站 4 无回应	持续	保持	保持
16#9B45	COM2 MODBUS 从站 5 无回应	持续	保持	保持
16#9B46	COM2 MODBUS 从站 6 无回应	持续	保持	保持
16#9B47	COM2 MODBUS 从站 7 无回应	持续	保持	保持
16#9B48	COM2 MODBUS 从站 8 无回应	持续	保持	保持
16#9B49	COM2 MODBUS 从站 9 无回应	持续	保持	保持
16#9B4A	COM2 MODBUS 从站 10 无回应	持续	保持	保持
16#9B4B	COM2 MODBUS 从站 11 无回应	持续	保持	保持
16#9B4C	COM2 MODBUS 从站 12 无回应	持续	保持	保持
16#9B4D	COM2 MODBUS 从站 13 无回应	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9B4E	COM2 MODBUS 从站 14 无回应	持续	保持	保持
16#9B4F	COM2 MODBUS 从站 15 无回应	持续	保持	保持
16#9B50	COM2 MODBUS 从站 16 无回应	持续	保持	保持
16#9B51	COM2 MODBUS 从站 17 无回应	持续	保持	保持
16#9B52	COM2 MODBUS 从站 18 无回应	持续	保持	保持
16#9B53	COM2 MODBUS 从站 19 无回应	持续	保持	保持
16#9B54	COM2 MODBUS 从站 20 无回应	持续	保持	保持
16#9B55	COM2 MODBUS 从站 21 无回应	持续	保持	保持
16#9B56	COM2 MODBUS 从站 22 无回应	持续	保持	保持
16#9B57	COM2 MODBUS 从站 23 无回应	持续	保持	保持
16#9B58	COM2 MODBUS 从站 24 无回应	持续	保持	保持
16#9B59	COM2 MODBUS 从站 25 无回应	持续	保持	保持
16#9B5A	COM2 MODBUS 从站 26 无回应	持续	保持	保持
16#9B5B	COM2 MODBUS 从站 27 无回应	持续	保持	保持
16#9B5C	COM2 MODBUS 从站 28 无回应	持续	保持	保持
16#9B5D	COM2 MODBUS 从站 29 无回应	持续	保持	保持
16#9B5E	COM2 MODBUS 从站 30 无回应	持续	保持	保持
16#9B5F	COM2 MODBUS 从站 31 无回应	持续	保持	保持
16#9B60	COM2 MODBUS 从站 32 无回应	持续	保持	保持

模拟 I/O 模块及温度模块

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A000	CH0 输入信号超出硬件规格	闪烁	
16#A001	CH1 输入信号超出硬件规格	闪烁	
16#A002	CH2 输入信号超出硬件规格	闪烁	
16#A003	CH3 输入信号超出硬件规格	闪烁	
16#A004	CH4 输入信号超出硬件规格	闪烁	
16#A005	CH5 输入信号超出硬件规格	闪烁	
16#A006	CH6 输入信号超出硬件规格	闪烁	
16#A007	CH7 输入信号超出硬件规格	闪烁	
16#A400	CH0 输入信号超出硬件规格	常亮	
16#A401	CH1 输入信号超出硬件规格	常亮	
16#A402	CH2 输入信号超出硬件规格	常亮	
16#A403	CH3 输入信号超出硬件规格	常亮	

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A404	CH4 输入信号超出硬件规格	常亮	
16#A405	CH5 输入信号超出硬件规格	常亮	
16#A406	CH6 输入信号超出硬件规格	常亮	
16#A407	CH7 输入信号超出硬件规格	常亮	
16#A600	插槽电源异常	常亮	
16#A601	电源异常	常亮	
16#A602	内部错误·CJC 补偿异常	常亮	
16#A603	内部错误·出厂校正异常	常亮	
16#A800	CH0 输入信号超出硬件规格	OFF	
16#A801	CH1 输入信号超出硬件规格	OFF	
16#A802	CH2 输入信号超出硬件规格	OFF	
16#A803	CH3 输入信号超出硬件规格	OFF	
16#A804	CH4 输入信号超出硬件规格	OFF	
16#A805	CH5 输入信号超出硬件规格	OFF	
16#A806	CH6 输入信号超出硬件规格	OFF	
16#A807	CH7 输入信号超出硬件规格	OFF	

*注：关于输入信号超出硬件规格与工程值超出极限两种错误，模块会依据用户所自定义的亮灯方式，来决定送出的错误代码是使用#A000~16#A00F、#A400~16#A40F、#A800~16#A80F 那个区段

AH02HC-5A/AH04HC-5A

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A001	CH0 线性累加超过范围	闪烁	
16#A002	CH0 前置比例值设定超过范围	闪烁	
16#A003	CH0 移动平均值设定超过范围	闪烁	
16#A004	CH0 比较值设定超过范围	闪烁	
16#A005	CH0 警报输出设定极限值错误	闪烁	
16#A006	CH0 中断编号设定超过范围	闪烁	
16#A011	CH1 线性累加超过范围	闪烁	
16#A012	CH1 前置比例值设定超过范围	闪烁	
16#A013	CH1 移动平均值设定超过范围	闪烁	
16#A014	CH1 比较值设定超过范围	闪烁	
16#A015	CH1 警报输出设定极限值错误	闪烁	
16#A016	CH1 中断编号设定超过范围	闪烁	
16#A021	CH2 线性累加超过范围	闪烁	

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A022	CH2 前置比例值设定超过范围	闪烁	
16#A023	CH2 移动平均值设定超过范围	闪烁	
16#A024	CH2 比较值设定超过范围	闪烁	
16#A025	CH2 警报输出设定极限值错误	闪烁	
16#A026	CH2 中断编号设定超过范围	闪烁	
16#A031	CH3 线性累加超过范围	闪烁	
16#A032	CH3 前置比例值设定超过范围	闪烁	
16#A033	CH3 移动平均值设定超过范围	闪烁	
16#A034	CH3 比较值设定超过范围	闪烁	
16#A035	CH3 警报输出设定极限值错误	闪烁	
16#A036	CH3 中断编号设定超过范围	闪烁	

AH05PM-5A/AH10PM-5A/AH15PM-5A

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A002	使用子程序无内容	闪烁	
16#A003	CJ、CJN、JMP 指令缺少对应的 P 标志	闪烁	
16#A004	主程序中有子程序指针	闪烁	
16#A005	缺少子程序	闪烁	
16#A006	同一程序中的指针重复	闪烁	
16#A007	子程序指针重复	闪烁	
16#A008	不同子程序中的跳转指令指针重复	闪烁	
16#A009	跳转指令与调用子程序指令使用相同指针	闪烁	
16#A00A	跳转指令指针与子程序相同	闪烁	
16#A00B	单段速目标位置 (I) 错误	闪烁	
16#A00C	单轴运动目标位置 (II) 错误	闪烁	
16#A00D	单轴运转速度 (I) 设定错误	闪烁	
16#A00E	单轴运转速度 (II) 设定错误	闪烁	
16#A00F	原点回归速度 (V_{RT}) 设定错误	闪烁	
16#A010	原点回归减速速度 (V_{CR}) 设定错误	闪烁	
16#A011	寸动 JOG 速度设定错误	闪烁	
16#A012	单轴正转运动正向脉冲禁止输出	闪烁	
16#A013	单轴反向运动反向脉冲禁止输出	闪烁	
16#A014	到达极限	闪烁	
16#A015	装置组件使用范围错误	闪烁	
16#A017	V/Z 修饰错误	闪烁	

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A018	浮点数转换错误	闪烁	
16#A019	BCD 转换错误	闪烁	
16#A01A	除法运算错误 (除数 = 0)	闪烁	
16#A01B	一般程序错误	闪烁	
16#A01C	LD/LDI 指令连续使用 9 次以上	闪烁	
16#A01D	RPT~RPE 超过 1 层以上	闪烁	
16#A01E	SRET 使用在 RPT~RPE 之间	闪烁	
16#A01F	主程序没有 M102 结束指令或运动程序没有 M2 结束指令	闪烁	
16#A020	使用错误指令或是使用装置超过范围	闪烁	

AH20MC-5A

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A002	使用子程序无内容	闪烁	
16#A003	CJ、CJN、JMP 指令缺少对应的 P 标志	闪烁	
16#A004	主程序中有子程序指针	闪烁	
16#A005	缺少子程序	闪烁	
16#A006	同一程序中的指针重复	闪烁	
16#A007	子程序指针重复	闪烁	
16#A008	不同子程序中的跳转指令指针重复	闪烁	
16#A009	跳转指令与调用子程序指令使用相同指针	闪烁	
16#A00A	跳转指令指针与子程序相同	闪烁	
16#A00B	单段速目标位置 (I) 错误	闪烁	
16#A00C	单轴运动目标位置 (II) 错误	闪烁	
16#A00D	单轴运转速度 (I) 设定错误	闪烁	
16#A00E	单轴运转速度 (II) 设定错误	闪烁	
16#A00F	原点回归速度 (V_{RT}) 设定错误	闪烁	
16#A010	原点回归减速速度 (V_{CR}) 设定错误	闪烁	
16#A011	寸动 JOG 速度设定错误	闪烁	
16#A012	单轴正转运动正向脉冲禁止输出	闪烁	
16#A013	单轴反向运动反向脉冲禁止输出	闪烁	
16#A014	到达极限	闪烁	
16#A015	装置组件使用范围错误	闪烁	
16#A017	V/Z 修饰错误	闪烁	
16#A018	浮点数转换错误	闪烁	
16#A019	BCD 转换错误	闪烁	

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A01A	除法运算错误 (除数 = 0)	闪烁	
16#A01B	一般程序错误	闪烁	
16#A01C	LD/LDI 指令连续使用 9 次以上	闪烁	
16#A01D	RPT~RPE 超过 1 层以上	闪烁	
16#A01E	SRET 使用在 RPT~RPE 之间	闪烁	
16#A01F	主程序没有 M102 结束指令或运动程序没有 M2 结束指令	闪烁	
16#A020	使用错误指令或是使用装置超过范围	闪烁	

AH10EN-5A/AH15EN-5A

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A001	Host 1 IP 地址冲突	闪烁	闪烁
16#A002	Host 2 IP 地址冲突	闪烁	闪烁
16#A003	Host 1 DHCP 失败	闪烁	闪烁
16#A004	Host 2 DHCP 失败	闪烁	闪烁
16#A401	硬件错误	常亮	常亮
16#A402	系统初始化失败	常亮	常亮

AH10SCM-5A/AH15SCM-5A

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A002	UD Link 设定错误或是通讯失败	闪烁	
16#A401	硬件发生错误	常亮	
16#A804	COM Port 通讯错误	OFF	
16#A808	MODBUS 通讯错误	OFF	

AH10DNET-5A

错误代码	说明	灯号状态		
		CPU	MODULE	
		BUS FAULT	MS	NS
16#A0F0	10DNET 扫描模块的站号与其它节点重复·或超出范围	红灯闪烁	绿灯 闪烁	红灯 常亮
16#A0F1	没有将任何从站配置到 10DNET 扫描列表中	红灯闪烁	绿灯 闪烁	绿灯 常亮

错误代码	说明	灯号状态		
		CPU BUS FAULT	MODULE	
			MS	NS
16#A0F2	10DNET 扫描模块的工作电压过低	红灯闪烁	红灯 闪烁	红灯 闪烁
16#A0F3	10DNET 扫描模块进入测试模式	红灯闪烁	橙灯 常亮	橙灯 常亮
16#A0F4	10DNET 扫描模块进入 Bus-OFF 状态	红灯闪烁	绿灯 常亮	红灯 常亮
16#A0F5	10DNET 扫描模块检测到 DeviceNet 网络没有电源	红灯闪烁	红灯 闪烁	红灯 常亮
16#A0F6	10DNET 扫描模块的内部储存单元出错	红灯闪烁	红灯 常亮	绿灯 闪烁
16#A0F7	10DNET 扫描模块的数据交换单元出错	红灯闪烁	红灯 常亮	绿灯 闪烁
16#A0F8	10DNET 扫描模块序列号检测出错	红灯闪烁	红灯 常亮	绿灯 闪烁
16#A0F9	10DNET 扫描模块读取或写入配置数据出错	红灯闪烁	红灯 常亮	红灯 常亮
16#A0FA	10DNET 扫描模块的站号与扫描列表中配置的从站站号重复	红灯闪烁	绿灯 常亮	红灯 常亮
16#A0FB	AH10DNET 和 AH CPU 之间数据交换失败	红灯闪烁	绿灯 常亮	绿灯 常亮
16#A0FC	从站出错、AHRTU-DNET 背板插槽上的模块出错或 AHRTU-DNET 从背板连接不正常	红灯闪烁	红灯 闪烁	绿灯 常亮

A5

AH10PFBM-5A

错误代码	说明	灯号状态			
		CPU BUS FAULT	MODULE		
			RUN	SYS	DP
16#A001	主站设定为空	红灯闪烁	绿灯 常亮	绿灯 常亮	绿灯 闪烁
16#A003	主站进入检测模式	红灯闪烁	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A005	主站内部芯片通讯超时	红灯闪烁	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A00B	与 PLC 数据交换超时	红灯闪烁	绿灯 常亮	绿灯 常亮	绿灯 常亮

错误代码	说明	灯号状态			
		CPU BUS FAULT	MODULE		
			RUN	SYS	DP
16#A402	PLC 没有为主站分配 I/O 映射区	红灯 常亮	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A404	主站初始化错误	红灯 常亮	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A406	内部储存单元出错	红灯 常亮	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A407	数据交换单元出错	红灯 常亮	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A408	主站序列号检测出错	红灯 常亮	绿灯 常亮	绿灯 常亮	绿灯 常亮
16#A4E2	主站检测到有从站全部掉线	红灯 常亮	OFF	绿灯 常亮	红灯 常亮
	主站检测到有部分从站掉线	红灯 常亮	OFF	绿灯 常亮	红灯闪烁
16#A4E6	主站检测到 AHRTU-PFBS-5A 连接的模块出错	红灯 常亮	绿灯 常亮	绿灯 常亮	绿灯 常亮

AH10PFBS-5A

错误代码	说明	灯号状态		
		CPU BUS FAULT	MODULE	
			RUN	NET
16#A4F0	AH10PFBS-5A 节点地址超出范围	红灯 常亮	绿灯 常亮	绿灯 常亮
16#A4F1	内部硬件错误	红灯 常亮	绿灯 常亮	绿灯 常亮
16#A4F2	参数化错误	红灯 常亮	绿灯 常亮	绿灯 常亮
16#A4F3	组态错误	红灯 常亮	绿灯 常亮	绿灯 常亮
16#A4F4	GPIO 检测出错	红灯 常亮	绿灯 常亮	绿灯 常亮
16#A4F5	AH10PFBS-5A 进入工厂测试模式	红灯 常亮	绿灯 常亮	绿灯 常亮
16#A4F6	1. AH10PFBS-5A 未接入 PFOFIBUS-DP 网络	红灯	绿灯	红灯

错误代码	说明	灯号状态		
		CPU	MODULE	
		BUS FAULT	RUN	NET
	2. PFOFIBUS-DP 主站没有配置 AH10PFBS-5A 从站或配置 AH10PFBS-5A 节点地址与实际连接的不符	常亮	常亮	常亮

AH10COPM-5A

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A0B0	心跳信息超时	闪烁	红灯双闪
16#A0B1	从站返回的 PDO 长度与与节点列表中设定的 PDO 数据长度不符	闪烁	OFF
16#A0B2	主站 NodeGuard 信息超时	闪烁	红灯双闪
16#A0E0	AH10COPM-5A 接收到从站发送的紧急信息	闪烁	OFF
16#A0E1	从站返回的 PDO 数据长度与节点列表中设定的 PDO 数据长度不符	闪烁	OFF
16#A0E2	未接收到从站 PDO	闪烁	OFF
16#A0E3	自动 SDO 下载失败	闪烁	OFF
16#A0E4	PDO 参数设定失败	闪烁	OFF
16#A0E5	关键参数设定有误	闪烁	OFF
16#A0E6	实际网络配置与设定配置不符	闪烁	OFF
16#A0E7	从站错误控制超时	闪烁	红灯双闪
16#A0E8	主从站站号重复	闪烁	OFF
16#A0F1	CANopen Builder 软件节点列表没有增加从站	闪烁	OFF
16#A0F3	AH10COPM-5A 处于错误状态	闪烁	OFF
16#A0F4	检测到总线脱离 (Bus-off)	闪烁	红灯常亮
16#A0F5	AH10COPM-5A 节点地址设定错误	闪烁	OFF
16#A0F6	内部错误：工厂制造流程出错	闪烁	OFF
16#A0F7	内部错误：GPIO 检测出错	闪烁	OFF
16#A0F8	内部错误：内部存储器检测出错	闪烁	OFF
16#A0F9	低电压检测错误	闪烁	OFF
16#A0FA	AH10COPM-5A 韧体内部处于错误状态	闪烁	OFF
16#A0FB	AH10COPM-5A 的发送暂存区已满	闪烁	OFF
16#A0FC	AH10COPM-5A 的接收暂存区已满	闪烁	OFF

A5

A.5.2 错误码与故障排除

AHxxEMC-5A

请依据 CPU 模块上的 LED 指示灯号状态及错误代码，从以下表格中获知故障排除方式。

ERROR 灯号常亮

错误代码	说明	处置方式
16#000B	PLC 程序毁损	重新下载项目程序。
16#000D	CPU 参数毁损	重新设定并下载 HWCONFIG 中的 CPU 模块参数。
16#0010	CPU 内存存取被拒	重新下载项目程序或参数，若一再出现请联络原厂。
16#0011	PLC ID 错误 (错误标志 SM9)	确认 PLC ID。
16#0012	PLC 密码错误 (错误标志 SM9)	确认 PLC 密码。
16#0014	无法执行系统还原程序 (错误标志 SM9)	因系统备份文件内容有错，或该文件不存在于 SD 卡指定路径中。若已存在但无法完成执行，请重新产生系统备份文件。若一再出现此信息请联络原厂。(请参考 AH500 操作手册第 7.5 节及 ISPSOft 使用手册第 18.2 节)
16#0015	模块配置数据错误 (错误标志 SM10)	表示 CPU 模块内部储存的模块配置数据有误，比对 HWCONFIG 的配置与目前整体的模块配置是否相同再重新下载。
16#0016	模块设定数据错误 (错误标志 SM10)	表示 CPU 模块内部储存的模块设定有误，确认该插槽的模块版本与 HWCONFIG 的模块版本相同或更新后，再重新下载。
16#0017	D 对应装置设定错误 (错误标志 SM10)	表示 CPU 模块内部储存 D 对应装置有误，检查 HWCONFIG 的模块内部参数是否正确，并重新下载。
16#001B	定时中断 (编号 0) 时间设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#001C	定时中断 (编号 1) 时间设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#001D	定时中断 (编号 2) 时间设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#001E	定时中断 (编号 3) 时间设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#001F	程序扫描逾时定时器设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#0020	固定扫描时间设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#0021	固定扫描时间设置错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#0022	下载 CPU 模块参数校验错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#0023	PLC 参数设定，Y 装置状态 (STOP->RUN) 设定错误	重新设定 HWCONFIG 的 CPU 模块参数并重新下载。
16#0026	通讯能力占用扫描时间比率设定错误	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0027	M 装置停电保持区范围设定错误	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0028	D 装置停电保持区范围设定错误	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0029	T 装置停电保持区范围设定错误	重置主机或恢复至出厂值后，重新下载程序与参数。
16#002A	C 装置停电保持区范围设定错误	重置主机或恢复至出厂值后，重新下载程序与参数。
16#002B	HC 装置停电保持区范围设定错误	重置主机或恢复至出厂值后，重新下载程序与参数。

错误代码	说明	处置方式
16#0050	停电保持区 SM 储存区块异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0051	停电保持区 SR 寄存器异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0052	停电保持区 M 储存区块异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0053	停电保持区 T 寄存器异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0054	停电保持区 C 寄存器异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0055	停电保持区 HC 寄存器异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0056	停电保持区 T 储存区块异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0057	停电保持区 C 储存区块异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0058	停电保持区 HC 储存区块异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#0059	停电保持区 D 寄存器异常	重置主机或恢复至出厂值后，重新下载程序与参数。
16#005A	停电保持区 W 寄存器异常	重置主机或恢复至出厂值后，重新下载程序与参数。

ERROR 灯号闪烁

错误代码	说明	处置方式
16#000A	扫描逾时 (错误标志 SM8)	1. 确认 HWCONFIG 中 CPU 模块参数的 WDT 设定。 2. 确认程序是否有造成扫描时间过长的设计。
16#000C	下载 PLC 程序校验错误	重新编译后下载项目程序。
16#000E	程序或参数下载中 PLC 无法切换至 RUN。	1 待下载完毕后重试。 2. 重新上电
16#0018	串行端口异常 (错误标志 SM9)	重试联机，若一再发生请联络原厂。
16#0019	USB 异常 (错误标志 SM9)	重试联机，若一再发生请联络原厂。
16#001A	系统备份文件 (DUP) 内容错误	请重新产生系统还原文件
16#0033	COM 1 通讯设置设置错误 (错误标志 SM9)	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中 CPU 模块的 COM Port 参数。
16#0034	COM 1 站号设置错误 (错误标志 SM9)	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中 CPU 模块的 COM Port 参数。
16#0035	COM 1 传输方式设置错误 (错误标志 SM9)	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中 CPU 模块的 COM Port 参数。

错误代码	说明	处置方式
16#0038	COM 2 通讯设置设置错误 (错误标志 SM9)	1. 检查程序与相关的 SR 。 2. 重新设定 HWCONFIG 中 CPU 模块的 COM Port 参数。
16#0039	COM 2 站号设置错误 (错误标志 SM9)	1. 检查程序与相关的 SR 。 2. 重新设定 HWCONFIG 中 CPU 模块的 COM Port 参数。
16#003A	COM 2 传输方式设置错误 (错误标志 SM9)	1. 检查程序与相关的 SR 。 2. 重新设定 HWCONFIG 中 CPU 模块的 COM Port 参数。
16#0066	系统备份时错误	1. 确认储存卡是否正常，空间是否足够。 2. 重试备份程序，若仍无法备份请联络原厂。
16#0067	系统还原的系统参数长度超出 CPU 模块的系统参数长度	此错误代码为警示代码。
16#2000	PLC 程序无 END 指令 (错误标志 SM5)	1. 重新编译程序再行下载。 2. 重新安装 ISPSOFT 后，再次编译程序并下载。
16#2001	项目程序内容有误：程序语法错误	确认程序并重新编译后，再次下载项目至 PLC。
16#2002	GOEND 使用的地方错误 (错误标志 SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#2003	程序中使用的装置超过可用范围 (错误标志 SM0/SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#2004	CJ/JMP 指令跳转的 P 地址错误，或是 P 重复使用 (错误标志 SM0/SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#2005	MC/MCR 相对应的 N 值不同，或数量不一样多 (错误标志 SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#2006	n 不是从 0 开始或是 n 的值不连续 (错误标志 SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#2007	ZRST 指令操作数使用不当 (错误标志 SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#200A	无效的指令 (错误标志 SM5)	确认程序并重新编译后，再次下载项目至 PLC。
16#200B	n 操作数或其它 K/H 操作数超出范围 (错误标志 SM0/SM5)	确认程序并重新编译后，再次下载项目至 PLC。

错误代码	说明	处置方式
16#200C	部份指令不允许操作数发生重迭(错误标志 SM0/SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#200D	BIN 转成 BCD 时发生错误 (错误标志 SM0/SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#200E	字符串没有 0x00 当做结尾 (错误标志 SM0/SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#200F	指令不支持 E 装置修饰 (错误标志 SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2010	指令不支持该装置类别/编码错误/16 位指令但 K·H 却是 32 位的编码(错误标志 SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2011	操作数的数目错误 (错误标志 SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2012	除法运算错误(错误标志 SM0/SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2013	浮点数格式错误, 超出可转换范围 (错误标志 SM0/SM5)。	确认程序并重新编译后, 再次下载项目至 PLC。
16#2014	TKON/TKOFF 指令所指定的 TASK 编号错误或超出范围 (错误标志 SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2015	CALL 指令超过 32 层 (错误标志 SM0)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2016	FOR-NEXT 指令超过 32 层(错误标志 SM0/SM5)	更新 ISPSOft 至最新版本, 重新编译后再下载。
16#2017	FOR 跟 NEXT 的指令数目不同 (错误标志 SM5)	确认程序并重新编译后, 再次下载项目至 PLC。
16#2018	在 FEND 之后的 P 指针没有相对应的 SRET, 或是有 SRET 但没有 P 指针 (错误标志 SM5)	1. 重新编译程序再行下载。 2. 重新安装 ISPSOft 后, 再次编译程序并下载。
16#2019	Interrupt I 的地址不是在 FEND 之后 (错误标志 SM5)	1. 重新编译程序再行下载。 2. 重新安装 ISPSOft 后, 再次编译程序并下载。
16#201A	IRET/SRET 的地址不是在 FEND 之后 (错误标志 SM5)	1. 重新编译程序再行下载 2. 重新安装 ISPSOft 后, 再次编译程序并下载。

错误代码	说明	处置方式
16#201B	I 没有相对应的 IRET，或是有 IRET 但没有 I (错误标志 SM5)	1. 重新编译程序再行下载 2. 重新安装 ISPSOft 后，再次编译程序并下载
16#201C	END 指令不是在程序的最后一个地址 (错误标志 SM5)	1. 重新编译程序再行下载 2. 重新安装 ISPSOft 后，再次编译程序并下载
16#201D	有 call 指令但没有 MAR 指令 (错误标志 SM5)	1. 重新编译程序再行下载。 2. 重新安装 ISPSOft 后，再次编译程序并下载。
16#201E	MODRW 指令中的功能代码错误 (错误标志 SM102/103)	确认指令用法与操作数设置。请参阅手册 API 1808 说明。
16#201F	MODRW 指令中的数据长度错误 (错误标志 SM102/103)	确认指令用法与操作数设置。请参阅手册 API 1808 说明。
16#2020	MODRW 的回复命令错误 (错误标志 SM102/103)	确认从站是否支持该功能代码与指定的操作
16#2021	MODRW 回复命令的校验和 (Checksum) 错误 (错误标志 SM102/103)	1. 确认是否有干扰并重送命令。 2. 确认从站装置是否正常运作。
16#2022	MODRW 指令的命令不符合 ASCII 格式 (错误标志 SM102/103)	确认命令格式符合 ASCII
16#2023	MODRW 指令的通讯超时 (错误标志 SM102/103)	检查从站是否正常运作，联机是否正常。
16#2024	RS 指令的通讯超时数值无效 (错误标志 SM102/103)	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中，CPU 模块的 COM Port 参数。
16#2025	RS 指令的通讯超时 (错误标志 SM102/103)	检查从站是否正常运作，联机是否正常。
16#2026	RS 指令的中断指针错误	请检查 RS 通讯中断服务程序否有下载
16#2027	FWD 应用指令异常	请参考应用指令手册，检查 FWD 应用指令
16#2028	REV 应用指令异常	请参考应用指令手册，检查 REV 应用指令
16#2029	STOP 应用指令异常	请参考应用指令手册，检查 STOP 应用指令
16#202A	RSDT 应用指令异常	请参考应用指令手册，检查 RSDT 应用指令
16#202B	RSTEF 应用指令异常	请参考应用指令手册，检查 RSTEF 应用指令

错误代码	说明	处置方式
16#202C 16#204B	IO 中断服务程序 0 不存在 IO 中断服务程序 31 不存在	请下载 IO 中断服务程序 0 (I/O 中断 0) 请下载 IO 中断服务程序 31 (I/O 中断 31)
16#2054 16#2127	外部中断服务程序 40 不存在 外部中断服务程序 251 不存在	请下载外部中断服务程序 40 (外部中断 40) 请下载外部中断服务程序 251 (外部中断 251)
16#2128	SFC Action 时间属性设定错误 (错误标志 SM0/SM1)	检查 SFC Action 时间属性是否重复设定
16#2129	SFC Action 重置属性设置错误 (错误标志 SM0/SM1)	检查 SFC Action 属性设定与重置设定是否相冲突
16#3040	凸轮点数据超过范围或不不存在	错误原因：功能块有输入凸轮点编号，超过目前凸轮表范围 排除方式：修正输入凸轮点编号在目前凸轮表范围内
16#3100	功能块输入引脚超出规定范围	错误原因：功能块有输入内容，超过可接受范围 排除方式：确认输入引脚超出规定范围
16#3102	功能块内部的子功能块发生错误	错误原因：功能块内部的子功能发生错误 排除方式：重新启动功能块
16#3103	空包功能块两检测 seneor 间距为负值	错误原因：功能块计算 seneor 距离时，得到负的数值 排除方式：确认 Sensor 安装位置是否正确
16#3104	相位移功能尚未准备完成	错误原因：该功能块前次启动时，输出引脚 Done 是否还没 On ，又再次启动 排除方式：重新启动该功能块
16#3105	叠加功能尚未准备完成	错误原因：该功能块前次启动时，输出引脚 Done 是否还没 On ，又再次启动 排除方式：重新启动该功能块
16#3106	在前次链钩修正尚未完成时，重复触发修正	错误原因：主轴运转速度过快，导致链钩修正无法完成 排除方式：请根据用户需求，调整包装机所有相关参数
16#3107	在前次膜轴修正尚未完成时，重复触发修正	错误原因：主轴运转速度过快，导致膜轴修正无法完成 排除方式：请根据用户需求，调整包装机所有相关参数
16#3108	在前次切刀修正尚未完成时，重复触发修正	错误原因：主轴运转速度过快，导致切刀修正无法完成 排除方式：请根据用户需求，调整包装机所有相关参数

错误代码	说明	处置方式
16#3400	轴号设定超出范围	错误原因：该功能块输入轴号不在可使用范围内 排除方式：重新设定该功能块轴号，使其在可使用范围内，重新启动该功能块
16#3401	SDO Data Type 设定错误 (0~199)	错误原因：资料型态未对应物件字典 排除方式：确认从站之物件字典定义
16#3404	通道编号超过该功能通道数	错误原因：该功能块输入通道号不在可使用范围内 排除方式：重新设定该功能块通道号，使其在可使用范围内，重新启动该功能块
16#3405	运转速度小于或等于零	错误原因：该功能块速度输入值为负值或零 排除方式：重新设定该功能块速度值，使其大于零后，重新启动该功能块
16#340A	原点模式设定错误	错误原因：原点模式设定内容非 1~35 排除方式：重新设定该功能块的原点模式，使其数值在 1~35 之间后，重新启动该功能块
16#340B	轴设定位置为 0	错误原因：该功能块的位移量输入内容为 0 排除方式：重新设定该功能块的位移量，使其数值在大于 0 后，重新启动该功能块
16#3410	坐标系统或输出脉冲形式设定错误	错误原因：功能块的单位系输入内容非 0~2 排除方式：重新设定该功能块的单位系，使其数值在 0 到 2 之间后，重新启动该功能块
16#3411	速度%错误	错误原因：功能块的速度超载值输入内容非 0~500 排除方式：重新设定该功能块的速度超载值，使其数值在 0~500 之间后，重新启动该功能块
16#3414	计数脉冲型式设定错误	错误原因：设定接收脉冲型式范围内容并非 0~3 排除方式：重新设定该功能输入脉冲型式，使其在可使用范围内，重新启动该功能块
16#3415	比较条件设定错误	错误原因：设定比较方式内容并非 0~2 排除方式：重新设定该功能比较方式，使其在可使用范围内，重新启动该功能块
16#3419	主轴位置设定小于零	错误原因：该功能块的主轴位置输入内容为小于或等于 0 排除方式：重新设定该功能块的主轴位置，使其数值在大于 0 后，重新启动该功能块

错误代码	说明	处置方式
16#341B	最大速度设定错误	错误原因：功能块的最大速度输入内容非 1~1000000 排除方式：重新设定该功能块的速度超载值，使其数值在 1~1000000 之间后，重新启动该功能块
16#3430	GroupNum 重复	错误原因：轴组编号重复 排除方式：使用 DFB_GroupReset 排除错误
16#3431	Group 轴号重复	错误原因：DFB_GroupEnable 的 AxisNumorder1~AxisNumorder6 输入有重复数值 排除方式：重新设定该功能块的 AxisNumorder1~AxisNumorder6，使其数值不重复，重新启动该功能块
16#3432	GroupNum 不存在	错误原因：功能块的 GroupNum 输入所指定的 Group，并未致能 排除方式：重新设定该功能块的 GroupNum 为已经致能的 Group
16#3433	Group 轴数不足	错误原因：功能块的 GroupNum 输入所指定的 Group，其内轴数不足 排除方式：重新设定该功能块的 GroupNum，其 Group 内部的轴数需符合该功能的轴数量，如直线插补最少 Group 内需包含两轴，圆弧插补最少 Group 内需包含三轴
16#3434	轴组运行时启动 DFB_GroupDisable 功能块	错误原因：Group 运动中被强制 Disable 排除方式：执行 DFB_GroupReset，清除 Group 错误状态
16#3435	Group 中互设同一轴	错误原因：DFB_GroupEnable 启动时，AxisNumorder1~AxisNumorder6 所指定的轴已经有其他轴组使用 排除方式：重新设定 AxisNumorder1~AxisNumorder6，使其内容不与其他轴组互用
16#3436	GroupNum 起始轴编号是零	错误原因：DFB_GroupEnable 的 AxisNumorder1 输入内容为小于或等于 0 排除方式：重新设定 DFB_GroupEnable 的 AxisNumorder1，使其数值在大于 0 后，重新启动 DFB_GroupEnable

错误代码	说明	处置方式
16#3437	GroupNum 设定超过范围	错误原因：功能块的 GroupNum 输入内容非 1~32 排除方式：重新设定该功能块的 GroupNum，使其数值在 1 到 32 之间后，重新启动该功能块
16#3438	GroupNum 在 ErrorStop 状态中	错误原因：功能块的启动时，Group 已经发生错误，此时 Group 为错误状态 排除方式：执行 DFB_GroupReset，清除 Group 错误状态
16#343A	Group 被执行 ImmediateStop 功能块	错误原因：轴组被执行紧急停止 排除方式：使用 DFB_GroupReset 清除错误码
16#343B	Group 中其它轴发生问题	错误原因：轴组中非该轴发生问题 排除方式：使用 DFB_GroupReset 清除错误码
16#3461	PDO 设定缺少必要的传输参数	错误原因：功能块执行中，发现该功能所需的参数不在 PDO 设定参数中 排除方式：重新执行 ECAT Builder，根据功能块所叙述的必要 PDO 参数，将此参数加入 PDO 传输参数中
16#3463	轴所指定的 ECAT Slave 不存在	错误原因：ECAT 的 Slave 不存在功能块所指定的轴 排除方式：确认 ECAT 所指定的 Slave 为可做轴运动
16#3500	轴状态不在 Disable or Standstill 或轴状态在 Coordinated 且轴组状态不再 Standby	错误原因：轴执行其它功能 排除方式：停止目前所执行的动作或等待该轴运作结束
16#3501	功能块 Channel 重复使用	错误原因：此频道已在使用中 排除方式：停止目前所执行的功能块
16#3502	不允许位置设定	错误原因：功能块的目标位置超过软极限，旋转轴以及不合法的位置 排除方式：重新设定目标位置
16#3505	Cam 点数据写入失败	错误原因：凸轮写点后检查，发生读回内容非写入数值 排除方式：重新启动写入凸轮点功能块
16#3506	轴在 Coordinated	错误原因:MC_Stop 功能块启动时，目前轴的状态为轴组状态启动中 排除方式:先确认 MC_Stop 的 Execute=False 以及 Done=True，才能使用 GroupReset 将轴组状态转换为 Standby，再使用 Groupdisable 使轴脱离轴组状态启动中

错误代码	说明	处置方式
16#3507	轴在 ErrorStop	错误原因:使用轴任一功能块启动时,目前轴的状态为轴组状态启动中 排除方式:先使用 GroupReset 将轴组状态转换为 Standby,再使用 Groupdisable 使轴脱离轴组状态启动中
16#3508	轴不在 StandStill 状态	错误原因:功能块的启动时,轴已经发生错误,此时轴为错误状态 排除方式:执行 MC_Reset,清除轴错误状态
16#3509	轴在 Stopping	错误原因:功能块启动时,状态机在 Stopping 状态 排除方式:关闭 MC_Stop 功能块,让状态机回到 StandStill
16#350B	轴加速时间太短	错误原因:轴设定加速时间太短 排除方式:调整轴参数最大加速度或是增加轴加速时间
16#350C	轴减速时间太短	错误原因:轴设定减速时间太短 排除方式:调整轴参数最大减速度或是增加轴减速时间
16#350D	读取凸轮数据长度超过范围	错误原因:读取凸轮数据长度超过范围 排除方式:调整读取凸轮数据长度为(1~256)
16#350E	写入凸轮数据长度超过范围	错误原因:写入凸轮数据长度超过范围 排除方式:调整写入凸轮数据长度(1~256)
16#350F	轴在 Synchronized	错误原因:当 MC_SetOverride 启动时,轴正在同步运动中,功能块报错 排除方式:使轴状态不在 Synchronized
16#3510	凸轮曲线直线的速度设定错误	错误原因:DFB_CamKeyPointWrite 曲线类型选择直线,但两点的速度不一致 排除方式:直线两端的速度需一样
16#3511	凸轮曲线直线的加速度设定错误	错误原因:DFB_CamKeyPointWrite 曲线类型选择直线,但两点的加速度不为 0 排除方式:直线两端的加速度需为 0
16#3512	Cam 点数据不存在	错误原因:欲读取的凸轮点不存在 排除方式:确认欲读取的凸轮点是否错误,若无误则重新下载 CAM 表

错误代码	说明	处置方式
16#3513	设定写入主轴位置小于前一个点	错误原因 :欲产生的凸轮表有主轴位置小于等于前一个点主轴位置的情形 排除方式 : 确认欲写入的凸轮点是否错误
16#3514	设定凸轮百分比超过范围 (0~100)	错误原因 :DFB_CamSwitch 功能块中,百分比设定非在 0~100 范围内 排除方式 : 重新设定百分比于 0 到 100 范围内
16#3526	轴前一个运动过程发生错误	错误原因 : 轴发生错误 排除方式 : 清除错误
16#3600	错误的状态机转换	错误原因 : 此功能块的执行时 , 轴当前状态无法执行该功能 排除方式 : 因此错误会造成轴状态进入 ErrorStop , 故需执行 MC_Reset , 使轴状态恢复到 StandStill 根据状态机的叙述 , 检查程序中 有无抵触的状态切换
16#3601	Buffer mode 的 buffer 数量已满	错误原因 : 此功能块的启动时 , 发生 buffer mode 的 buffer 数量已满的状况 排除方式 : 因此错误会造成轴状态进入 ErrorStop , 故需执行 MC_Reset , 使轴状态恢复到 StandStill 检查该功能启动时 , 程序中其它功能块同时启动 buffer mode 的数量不能大于 20 个
16#3602	执行多个无法同时启动功能	错误原因:此功能块启动时 , 无法支持同时启动功能 排除方式 : 使用 MC_Reset 功能块清除错误 , 并将轴状态从 ErrorStop 切换为 StandStill
16#3603	Buffermode 设置参数错误	错误原因 : BufferMode 输入引脚不合法 排除方式 : 使用 MC_Reset 清除错误并重新设定输入参数
16#3604	功能块运动方向错误	错误原因 : 轴运转方向不合法 排除方式 : 使用 MC_Reset 清除错误并重新设定输入参数
16#3605	轴组或是单轴相关运动功能块参数设定错误	错误原因 : 轴组或是单轴功能块相关参数设定错误以致运动目标位置无法到达 排除方式 : 使用 DFB_GroupReset 或是 MC_Reset 清除错误并重新设定相关参数

错误代码	说明	处置方式
16#3606	运动目标距离为 0	错误原因：运动目标距离为 0 排除方式：使用 MC_Reset 清除错误并重新设定输入参数
16#3607	目标速度超出范围	错误原因：目标速度超出范围 排除方式：使用 MC_Reset 清除错误并重新设定输入参数
16#3608	目标速度超出范围	错误原因：目标速度超出范围 排除方式：使用 MC_Reset 清除错误并重新设定输入参数
16#3612	到达正向极限	错误原因：到达正向极限 排除方式：使用 MC_Reset 清除错误，并正转或反转回退
16#3613	到达负向极限	错误原因：到达负向极限 排除方式：使用 MC_Reset 清除错误，并正转或反转回退
16#3614	超出伺服极限	错误原因：到达伺服极限 排除方式：使用 MC_Reset 清除错误，并正转或反转回退
16#3617	加速度超出范围	错误原因：功能块执行时，加速度超过轴参数最大加速度，或是在 Buffer 接续时超过轴参数最大加速度 排除方式：重新设定功能块或轴参数的加速度
16#3618	减速度超出范围	错误原因：功能块执行时，加速度超过轴参数最大减速度，或是在 Buffer 接续时超过轴参数最大减速度 排除方式：重新设定功能块或轴参数的减速度
16#3619	站号不存在	错误原因：站址不存在 排除方式：确认站址，重启动功能块
16#3620	SDO 排程缓冲区已满	错误原因：SDO 排程缓冲区已满 排除方式：等待 SDO 排程缓冲区非满，重启动功能块
16#3622	SDO OD 数据类型不合	错误原因：OD 资料型态不合 排除方式：确认 OD 数据类型，重启动功能块
16#3623	SDO 逾时	错误原因：断线逾时 排除方式：确认连线，重启动功能块

错误代码	说明	处置方式
16#3624	SDO 数据写入错误	错误原因：从站发生错误 排除方式：排除从站错误，重启动功能块
16#3625	SDO 数据读出错误	错误原因：从站发生错误 排除方式：排除从站错误，重启动功能块
16#3626	SDO Retry 次数设定超过范围	错误原因：Retry 次数设定超过范围 排除方式：确认站址，重启动功能块
16#3630	AxisPara 设定错误	错误原因：轴参数设定错误 排除方式：调整 AxisPara 输入值，重新启动功能块
16#3631	AxisPara.GearRatioNume / GearRatioDeno 无法设定到伺服	错误原因：SDO 写入伺服参数（齿轮比）时，发生写入错误状况 排除方式：检查 SDO 写入或读出动作，是否有正在进行的状况
16#3635	此轴参数不可修改	错误原因：AxisPara 该成员不可修改 排除方式：调整 AxisPara 该成员输入值，重新启动功能块
16#3636	此轴参数设定错误	错误原因：AxisPara 该成员输入错误 排除方式：调整 AxisPara 该成员输入值，重新启动功能块
16#3637	AxisPara.SWPosLimit 与 AxisPara.SWNegLimit 包含的范围有误（两者皆为 0 导致范围为 0；范围未包含目前指令位置）	错误原因：AxisPara.SWPosLimit 与 AxisPara.SWNegLimit 包含的范围有误（两者皆为 0 导致范围为 0；范围未包含目前指令位置） 排除方式：调整 AxisPara.SWPosLimit 与 AxisPara.SWNegLimit 输入值，重新启动功能块
16#3638	AxisPara.RotaryMaxPos 与 AxisPara.RotaryMinPos 包含的范围有误（两者皆为 0 导致范围为 0）	错误原因：AxisPara.RotaryMaxPos 与 AxisPara.RotaryMinPos 包含的范围有误（两者皆为 0 导致范围为 0） 排除方式：调整 AxisPara.RotaryMaxPos 与 AxisPara.RotaryMinPos 输入值，重新启动功能块
16#3639	在轴非 Disabled 时不可修改 GearRatio	错误原因：AxisPara.AxisType 不可修改 排除方式：调整 AxisPara.GearRatioNume 与 AxisPara.GearRatioDeno 输入值或者将轴 Disable，重新启动功能块

错误代码	说明	处置方式
16#3800	运行过程中轴断线	错误原因：此功能块的执行时，发生运动网络断线的状况 排除方式：检查运动网络是否有脱落状况
16#3801	EtherCAT Axis 发生错误	错误原因：轴回报目前运动过程中发生错误或警告信息 排除方式：利用轴状态回传功能块，可得知目前轴的错误码，并利用 MC_Reset 清除此错误
16#3901	网络重新启动失败	错误原因：运动网络重置时，发生运动网络无法完成联机动作 排除方式： 检查运动网络是否有脱落状况 A2 伺服是否有上电
16#3904	SDO 无法读取该参数	错误原因：SDO 读取伺服参数时，发生读取错误状况 排除方式：检查读取参数中，Group 与 Parameter 设定是否符合伺服可接受范围
16#3905	SDO 无法写入该数值	错误原因：SDO 写入伺服参数时，发生写入错误状况 排除方式： 检查读取参数中 Group 与 Parameter 设定是否符合伺服可接受范围 检查写入的数值内容，是否符合该参数的可接受范围
16#3906	MaxTorque 写入失败	错误原因：MC_SetTorqueLimit 的 PositiveValue 或 NegativeValue 无法写入伺服 排除方式： PositiveValue 是否符合伺服可接受范围 NegativeValue 是否符合伺服可接受范围
16#3907	虚拟轴不支持此功能	错误原因：功能块执行时该轴不能为虚拟轴 排除方式：将虚拟轴设定为实轴
16#3909	目前网络正在执行其它动作	错误原因：功能块启动时，AH Motion Controller 的运动网络正在执行其它网络功能 排除方式：检查 SDO 写入或读出动作，是否有正在进行的状况
16#390C	运动过程中轴发生错误	错误原因：轴回报目前运动过程中发生错误或警告信息 排除方式：利用轴状态回传功能块，可得知目前轴的错误码，并利用 MC_Reset 清除此错误

错误代码	说明	处置方式
16#3910	在未啮合状态下取消啮合	错误原因：该轴未作 mc_gearin、mc_gearinpos、mc_combineaxes，却执行 mc_gearout 排除方式：该轴未作 mc_gearin、mc_gearinpos、mc_combineaxes，不要执行 mc_gearout
16#3911	轴软极限错误	错误原因：轴超出软极限 排除方式：使用 MC_Reset 清除错误，利用 MC_MoveAbsolute、MC_MoveRelative、MC_MoveVelocity、DFB_MPG 退回到软极限内
16#3912	功能块位置输入引脚超出旋转轴范围	错误原因：位置输入引脚超出旋转轴范围 排除方式：修正位置输入以符合旋转轴范围
16#3913	同步运动啮合失败	错误原因：啮合完成之前，主轴速度曾经改变 排除方式：啮合完成之前，主轴速度不要改变
16#3914	GearInPos 速度设定太小	错误原因：GearInPos 最大速度设定太小 排除方式：放宽最大速度
16#3915	GearInPos 急跳度设定太小	错误原因：GearInPos 最大急跳度设定太小 排除方式：放宽最大急跳度
16#3916	GearInPos 啮合时间太短	错误原因：GearInPos 啮合时间太短 排除方式：增加 MasterStartDistance
16#3917	GearInPos 开始啮合时主轴速度为 0	错误原因：GearInPos 开始啮合时主轴速度为 0 排除方式：使主轴速度不要为 0
16#3918	啮合速度大于轴最高速度	错误原因：啮合速度大于该轴允许最大速度 排除方式：调整轴参数最大速度限制
16#3919	GearInPos 主轴运转方向相反	错误原因：GearInPos 主轴运转方向相反 排除方式：使主轴与从轴运动方向相同
16#3920	GearInPos 加速度设定太小	错误原因：加速度或减速度设定过小 排除方式：增加加速度或减速度设定
16#3921	GearInPos MasterStartDistancs 设定超出范围	错误原因：MasterStartDist 设定超出范围 排除方式：确认主轴运动方向与啮合之起始与完成位置
16#3922	GearInPos 啮合位移量过小	错误原因：SlaveSyncPosition 过小 排除方式：增加 SlaveSyncPosition，或是增加加速度或减速度设定

错误代码	说明	处置方式
16#3923	GearInPos 啮合位移量过大	错误原因：SlaveSyncPosition 过大 排除方式：减少 SlaveSyncPosition，或是增加加速度或减速度设定
16#3924	GearInPos 开始啮合时主轴速度有变化	错误原因：InSync 之前，主轴速度有所改变 排除方式：InSync 之前，固定主轴速度
16#3950	capture 无法工作在脉冲速度大于 1MHz	错误原因：捕捉轴的等效脉冲速度大于 1MHz 排除方式：降低捕捉轴的等效脉冲速度，重新启动功能块
16#3951	DFB_CamCurve2 的参数设定有误	错误原因：CamCurve2 的输入有误（如 Concatenate 此时不可为 true 或其他输入参数不符正确范围） 排除方式： 1. 重新上电后，在第一次执行 DFB_CamCure2 时 concatenate 设为 false，功能块可以 Done，在 Done 之后，再开将 concatenate 设为 true，再次执行 DFB_CamCure2，就不会再出现 error 2. 检查其他输入参数是否合理
16#3953	Capture2 功能重复使用同个 Channel 编号	错误原因：同时有两组以上 DFB_Capture2 共用同一个 Channel 编号 排除方式：用其他未使用的 Channel 编号
16#3954	Torque Slope 写入失败	错误原因：MC_TorqueControl 写入的数值，该伺服不支持 排除方式：查询该伺服的手册文件，确认可写入的数值范围
16#3955	Torque Velocity 写入失败	错误原因：MC_TorqueControl 写入的数值，该伺服不支持 排除方式：查询该伺服的手册文件，确认可写入的数值范围
16#3A00	CamIn 的 CAM Table 设定错误	错误原因：指定的 CAM Table 不存在 排除方式：增加 CAM Table 或重新设定为一个存在的 CAM Table
16#3A01	CamIn 的 Master 设定错误	错误原因：主轴来源设定错误 排除方式：重新设定主轴来源

错误代码	说明	处置方式
16#3A02	CamIn 太早更换 Cam 表	错误原因：更换新的 CAM Table 动作尚未完成，又重新至换另一个 Table 排除方式：重新启动 CAM 功能
16#3A03	CamIn Activation Mode 设定超出范围	错误原因：Activation Mode 设定值非 0 或 1 排除方式：重新设定模式后启动 CAM 功能
16#3A04	CamIn Start Mode 设定超出范围	错误原因：CAM 的 Start Mode 数值设定非 0~3 排除方式：将 Start mode 设定在合理范围中重新启动 CAM
16#3A05	CamIn 的 Master Scaling 数值设定为 0.0	错误原因：CAM 的 Master Scaling 数值设定为 0 排除方式：将 Master Scaling 设定为非 0 数值重新启动 CAM
16#3A06	CamIn 的 Slave Scaling 数值设定为 0.0	错误原因：CAM 的 Slave Scaling 数值设定为 0 排除方式：将 Slave Scaling 设定为非 0 数值重新启动 CAM
16#3A07	CamIn 的 MasterSyncPosition 设定错误	错误原因：CAM 的 MasterSyncPosition 设定小于 0 排除方式：将 MasterSyncPosition 设定为非负数值后重新启动 CAM
16#3A08	CamIn 的 ActivationPosition 设定错误	错误原因：CAM 的 ActivationPosition 设定内容不合理 排除方式：将 ActivationPosition 设定在轴极限范围内后重新启动 CAM
16#3A10	CamIn 的 Master Start Position 设定太小	错误原因：CAM 的凸轮咬合点设定小于前置量 排除方式：Start Position 设定加大后重新启动 CAM 功能
16#3A11	CamIn 开始咬合时主轴速度有变化	错误原因：CAM 开始咬合时主轴加速度不为 0 排除方式：将主轴在咬合区间设定成等速运动再重新启动 CAM
16#3A12	CamIn 的 Start Mode 中所需的速度太小	错误原因：CAM 的 Start Mode 中所需的速度太小 排除方式：将速度调大后重新启动 CAM
16#3A13	CamIn 的 Start Mode 中所需的速度太小	错误原因：CAM 的 Start Mode 中所需的升速度太短 排除方式：将升速度调大后重新启动 CAM
16#3A14	CamIn 的 Start Mode 中所需的啮合时间太短	错误原因：CAM 的 Start Mode 中所需的啮合时间太短 排除方式：延后啮合时间点
16#3A15	CamIn 的 Start Mode 中所需的急跳度太小	错误原因：CAM 的 Start Mode 中所需的急跳度太小 排除方式：急跳度调大后重新启动 CAM

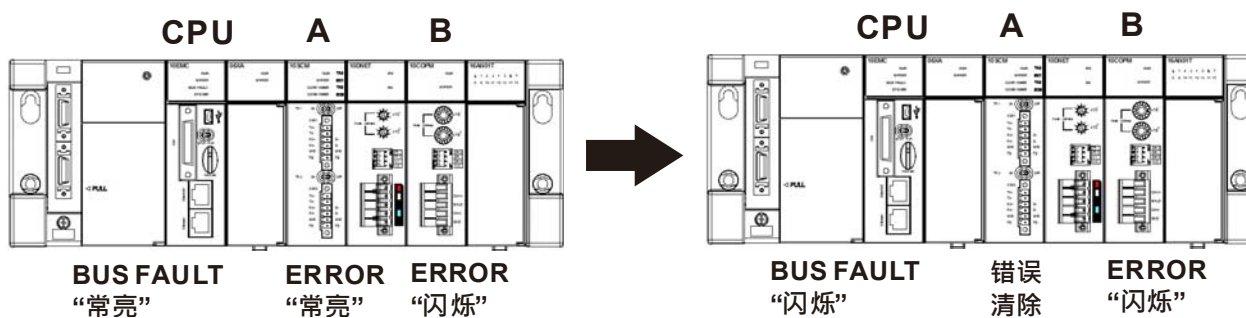
错误代码	说明	处置方式
16#3A16	CamIn 的 Start Mode 中所需的最大加速度太小	错误原因：CAM 的 Start Mode 中所需的最大加速度太小 排除方式：将加速度调大后重新启动 CAM
16#3A17	CamIn 的 Start Mode 中所需的开始距离太小	错误原因：CAM 的 Start Mode 中所需的开始距离太小 排除方式：将 Sync Position 调大后重新启动 CAM
16#3A18	CamIn 的 Start Mode 中所需的开始距离太大	错误原因：CAM 的 Start Mode 中所需的开始距离太大 排除方式：将 Sync Position 调小后重新启动 CAM
16#3A19	同一轴同时启动大于五组的 CAM	错误原因：同一轴同时启动大于五组的 CAM 排除方式：修正程序，勿同时启动
16#3A20	Master 运行方向为负向运动	错误原因：Slave 在做 CAM 运动过程中，Master 的运行方向为负向 排除方式：利用 MC_Reset 清除错误
16#3A21	在 MC_CamIn 未启动状态下触发 MC_CamOut	错误原因：指定的 Slave 轴非 CamIn 的咬合运动 排除方式：利用 MC_Reset 清除错误
16#3D00	EtherCAT ENI 文件与目前硬件配置不合	错误原因：系统内存的 ENI 文件与目前 EtherCAT 系统不一致 排除方式：重新下载符合目前 EtherCAT 系统的 ENI 文件
16#3D01	EtherCAT slave 遗失	错误原因：运动网络联机过程中，Slave 发生断线现象 排除方式：执行系统重新联机功能
16#3D03	EtherCAT DC 时间设定过小（小于支持轴数的最小扫描时间）	错误原因：EtherCAT DC 时间设定错误 8 轴最小 500us，16 轴最小 1000us，32 轴最小 2000us 排除方式：确认目前的轴数后，调整 DC 时间
16#6001	IP 地址不合法	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中，CPU 模块的以太网网络（Ethernet）参数。
16#6002	网络屏蔽地址不合法	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中，CPU 模块的以太网网络（Ethernet）参数。
16#6003	网关地址不合法	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中，CPU 模块的以太网网络（Ethernet）参数。

错误代码	说明	处置方式
16#6004	以太网络的 IP 地址过滤设置错误	重新设定 HWCONFIG 中 · CPU 模块的以太网网络 (Ethernet) 参数。
16#6006	以太网络的静态 ARP 表设置错误	重新设定 HWCONFIG 中 · CPU 模块的以太网网络 (Ethernet) 参数。
16#8242	G-code Data ERROR	1.请使用 ISPSOft 重新下载内部 G-code
16#8243	ECAM Data ERROR	1.请使用 ISPSOft 重新下载内部 ECAM
16#8244	ENI Data ERROR	1.请使用 ECATBuilder 重新下载网络配置数据
16#8245	EtherCat Data ERROR	1.请使用 ECATBuilder 重新下载网络配置数据
16#8246	Axes Parameters Data ERROR	1.请使用 ISPSOft 重新下载轴参数
16#8247	External Gcode Data ERROR	1.请确人机器上是否有 SD 卡并使用 ISPSOft 重新下载外部 Gcode

BUS FAULT 灯号常亮

A5

除了主机自行检测的错误会令 CPU 模块的 BUS FAULT 亮灯之外，该灯号亦会与模块的 ERROR 灯相互对应。当某个模块发生错误时，主机的 BUS FAULT 灯号便会与该模块的 ERROR 灯号呈现相同的亮灯方式；但当同时发生多个会让 BUS FAULT 灯号亮灯的状况时，BUS FAULT 的灯号便会选择呈现较严重错误的亮灯方式；例如，当同时有两个模块发生错误，其中模块 A 的灯号维持常亮，而模块 B 则呈现闪烁状态，此时 CPU 模块的 BUS FAULT 灯将会维持常亮；当模块 A 的故障被排除后，若模块 B 仍呈现闪烁状态，CPU 模块的 BUS FAULT 灯则会再切换为闪烁状态。关于各模块的亮灯方式请参考本章第 A.5.1 节的相关介绍。



下表为主机自行检测且会让 BUS FAULT 维持常亮的错误及处置方式。若所获取的错误代码不在下表中，请检查各模块的状况是否正常。关于各模块的错误处置方式请参考本节后续段落的相关介绍。

错误代码	说明	处置方式
16#0013	I/O 模块无法设置运行/停止 (错误标志 SM10)	因模块设定参数错误，若设定正确，请检查模块是否故障，若一再出现此信息请联络原厂。

错误代码	说明	处置方式
16#0014	无法执行系统复制程序 (错误标志 SM9 ; 此问题发生时 , ERROR&Bus Fault 灯会维持常亮)	因系统复制文件内容有错 , 或该文件不存在于指定路径中。若已存在但无法完成执行 , 请重新产生系统复制文件。若一再出现此信息请联络原厂。
16#1401	模块存取错误 (错误标志 SM9)	请联络原厂。
16#1402	实际模块不符合配置设定 (错误标志 SM9)	确认 HWCONFIG 中的模块配置设定与实际的模块配置是否吻合。
16#1403	从模块读取数据错误 (错误标志 SM9)	检查模块是否正常运作 , 若一再出现此信息请联络原厂。
16#1405	搜寻不到模块的设定参数 (错误标志 SM9)	重新设定并下载 HWCONFIG 参数。
16#140B	通讯模块数量超过上限 (错误标志 SM9)	请将通讯模块减少至符合系统支持的数量。
16#140C	高速数据交换检查码错误	请确认模块韧体版本并联络原厂
16#140D	实际 Power ID 不符合配置设定	确认 HWCONFIG 中的电源方案设定与实际的模块配置是否吻合
16#140E	模块高速数据交换数量超出支持最大范围	请确认模块韧体版本并联络原厂
16#140F	高速数据交换错误	请确认模块韧体版本并联络原厂

BUS FAULT 灯号闪烁

当 BUS FAULT 灯号呈现闪烁的状况时 , 请确认各模块的工作状态。关于各模块的亮灯方式请参考本章第 A.5.1 节的相关介绍 , 而各模块的错误处置方式则请参考本章节后续段落的相关介绍。

其它

错误代码	说明	处置方式
16#000F	PLC 原始程序损毁	重新下载项目程序。
16#0024	背板无 IO 模块	请确认 IO 模块是否存在
16#005E	储存卡的初始程序错误 (错误标志 SM453)	确认储存卡是否故障
16#005F	在储存卡中 , 欲读取不存在的文件 , 或写入不存在路径的文件 (错误标志 SM453)	确认文件路径是否正确

错误代码	说明	处置方式
16#0061	储存卡容量不足 (错误标志 SM453)	确认储存卡容量是否足够
16#0062	储存卡为写保护模式 (错误标志 SM453)	确认储存卡是否设为写保护
16#0063	数据写入储存卡的文件时有错误 (错误标志 SM453)	确认文件路径是否正确，或储存卡是否故障。
16#0064	储存卡的文件无法被读取 (错误标志 SM453)	确认文件路径是否正确，文件是否损毁。
16#0065	储存卡中的文件为只读状态 (错误标志 SM453)	将文件设为非只读
16#1801	CPU 模块未设定中断工作	确认 PLC 程序是否有建立相对应的中断工作 (24V LV Detection)
16#3400	轴设定错误 (1~32)	错误原因：该功能块输入轴号不在可使用范围内 排除方式：重新设定该功能块轴号，使其在可使用范围内，重新启动该功能块
16#3401	SDODataType 设定错误 (0~199)	错误原因：数据类型未对应对象字典 排除方式：确认从站的对象字典定义
16#3404	信道编号超过该功能信道数	错误原因：该功能块输入信道号不在可使用范围内 排除方式：重新设定该功能块信道号，使其在可使用范围内，重新启动该功能块
16#3414	计数脉冲型式设定错误	错误原因：设定接收脉冲型式范围内容并非 0~3 排除方式：重新设定该功能输入脉冲型式，使其在可使用范围内，重新启动该功能块
16#3415	比较条件设定错误	错误原因：设定比较方式内容并非 0~2 排除方式：重新设定该功能比较方式，使其在可使用范围内，重新启动该功能块
16#342A	Gcode 程序来源错误	错误原因：功能块的 GcodeID 输入所指定的 G code 文件，不存在于 AH Motion Controller 主机中或外接 SD 卡中 排除方式：重新设定该功能块的 GcodeID，此 GcodeID 所指定的文件已存在于 AH Motion Controller 主机中或外接 SD 卡中后，重新启动该功能块

错误代码	说明	处置方式
16#342B	GcodeID 设定超过范围	错误原因：功能块的 GcodeID 输入内容非 1 ~ 136 排除方式：重新设定该功能块的 GcodeID，使其数值在 1 到 136 之间后，重新启动该功能块
16#342C	Gcode 运作中	错误原因：对应的轴组在执行 DFB_GroupGcodeRun 排除方式：等待 DFB_GroupGcodeRun 完成，才可启动该功能块
16#342D	Gcode 文法检查中	错误原因：对应的轴组在执行 DFB_GroupGcodeSyntax 排除方式：需等待 DFB_GroupGcodeSyntax 执行完成，才可启动 DFB_GroupGcodeRun 功能块
16#342E	Gcode Filter 设定超过范围	错误原因：对应的轴组在执行 DFB_GroupRun Filter 设定值超过 1000 或是小于 0 排除方式：Filter 设定正确数值，才可启动该功能块
16#3430	GroupNum 重复	错误原因：轴组编号重复 排除方式：使用 DFB_GroupReset 排除错误
16#3431	Group 轴号重复	错误原因：DFB_GroupEnable 的 AxisNumorder1~AxisNumorder6 输入有重复数值 排除方式：重新设定该功能块的 AxisNumorder1~AxisNumorder6，使其数值不重复，重新启动该功能块
16#3432	GroupNum 不存在	错误原因：功能块的 GroupNum 输入所指定的 Group，并未致能 排除方式：重新设定该功能块的 GroupNum 为已经致能的 Group
16#3433	Group 轴数不足	错误原因：功能块的 GroupNum 输入所指定的 Group，其内轴数不足 排除方式：重新设定该功能块的 GroupNum，其 Group 内部的轴数需符合该功能的轴数量，如直线补间最少 Group 内需包含两轴，圆弧补间最少 Group 内需包含三轴
16#3434	轴组运行时启动 DFB_GroupDisable 功能块	错误原因：轴组运行时启动 DFB_GroupDisable 功能块 排除方式：执行 DFB_GroupReset，清除 Group 错误状态

错误代码	说明	处置方式
16#3435	Group 中互设同一轴	错误原因：DFB_GroupEnable 启动时，AxisNumorder1~ AxisNumorder6 所指定的轴已经有其它轴组使用 排除方式：重新设定 AxisNumorder1~ AxisNumorder6，使其内容不与其它轴组互用
16#3436	GroupNum 起始轴编号是零	错误原因：DFB_GroupEnable 的 AxisNumorder1 输入内容为小于或等于 0 排除方式：重新设定 DFB_GroupEnable 的 AxisNumorder1，使其数值在大于 0 后，重新启动 DFB_GroupEnable
16#3437	GroupNum 设定超过范围	错误原因：功能块的 GroupNum 输入内容非 1~16 排除方式：重新设定该功能块的 GroupNum，使其数值在 1 到 16 之间后，重新启动该功能块
16#3438	GroupNum 在 ErrorStop 状态中	错误原因：功能块的启动时，Group 已经发生错误，此时 Group 为错误状态 排除方式：执行 DFB_GroupReset，清除 Group 错误状态
16#343A	Group 被执行 ImmediateStop 功能块	错误原因：轴组被执行紧急停止 排除方式：使用 DFB_GroupReset 清除错误码
16#3463	轴所指定的 ECAT Slave 不存在	错误原因：ECAT 的 Slave 不存在功能块所指定的轴 排除方式：确认 ECAT 所指定的 Slave 为可做轴运动
16#3501	功能块 Channel 重复使用	错误原因：重复设定已被使用中的 Channel 排除方式：设定其它未被使用的 Channel 或是解除目前已被使用的 Channel
16#3505	Cam 点数据写入失败	错误原因：凸轮写点后检查，发生读回内容非写入数值 排除方式：重新启动写入凸轮点功能块
16#3512	Cam 点数据不存在	错误原因：欲读取的凸轮点不存在 排除方式：确认欲读取的凸轮点是否错误，若无误则重新下载 CAM 表
16#3619	站址不存在	错误原因：站址不存在 排除方式：确认站址，重启动功能块

错误代码	说明	处置方式
16#3620	SDO 排程缓冲区已满	错误原因：SDO 排程缓冲区已满 排除方式：等待 SDO 排程缓冲区非满，重启动功能块
16#3622	SDO OD 数据类型不合	错误原因：OD 数据类型不合 排除方式：确认 OD 数据类型，重启动功能块
16#3623	SDO 超时	错误原因：断线超时 排除方式：确认联机，重启动功能块
16#3624	SDO 数据写入错误	错误原因：从站发生错误 排除方式：排除从站错误，重启动功能块
16#3625	SDO 数据读出错误	错误原因：从站发生错误 排除方式：排除从站错误，重启动功能块
16#3626	SDO Retry 次数设定超过范围	错误原因：Retry 次数设定超过范围 排除方式：确认站址，重启动功能块
16#3950	capture 无法工作在脉冲速度大于 1MHz	错误原因：捕捉轴的等效脉冲速度大于 1MHz 排除方式：降低捕捉轴的等效脉冲速度，重新启动功能块
16#3951	CamCurve 功能块输入引脚参数错误	错误原因：CamCurve 功能块输入引脚参数错误 排除方式：(1)重新上电后，在第一次执行 DFB_CamCure2 时 concatenate 设为 false，功能块可以 Done，在 Done 之后，再开将 concatenate 设为 true，再次执行 DFB_CamCure2，就不会再出现 error (2)检查其它输入数值是否合理
16#3953	Capture 功能重复使用同个 Channel 编号	错误原因：同时有两组以上 DFB-Capture FB 共享同一个 Channel 编号 排除方式：用其它未使用的 Channel 编号
16#600D	RJ45 端口未连接 (错误标志 SM1100)	错误原因：RJ45 端口未连接 排除方式：检查通讯电缆。
16#620D	UDP 通讯接口 (Socket) 功能的传送数据长度不合法	1. 检查程序与相关的 SR。 2. 重新设定 HWCONFIG 中 CPU 模块的以太网 (Ethernet) 参数。
16#6212	远程装置响应超时	确认远程装置仍保持联机。

错误代码	说明	处置方式
16#6213	接收数据超过限制	检查程序与相关的 SR。 重新设定 HWCONFIG 中，CPU 模块的以太网网络 (Ethernet) 参数。
16#6214	远程装置拒绝联机	确认远程装置运作正常。
16#6400	EMDRW 指令操作的联机数超出限制或未设定送信标志	检查程序是否修改到送信标志。 重试设定标志及传送报文。
16#6401	远程装置中止联机	检查远程装置是否支持 MODBUS Port (502)。
16#6402	远程装置响应超时	检查远程装置运作是否正常。
16#6403	API 指令的远程 IP 地址不合法	检查程序是否正确。
16#6404	收到不支持的 MODBUS 功能代码	检查远程装置传送的命令。
16#6405	MODBUS 回复信息的 Byte Count 与实际的数据长度不符	检查远程装置传送的命令。
16#6501	远程装置响应超时 (错误标志 SM828-SM955)	用错误标志查出问题的联机号码后，确认与该装置的联机是否正常。
16#6502	远程装置回复报文错误 (错误标志 SM828-SM955)	用错误标志查出问题的联机号码后，确认该装置运作是否正常。
16#6700	MODBUS TCP 数据交换初始化错误	请检查设定值并重新下载
16#6701	MODBUS TCP 数据交换超时	请检查远程装置有支持 MODBUS 通讯协议
16#6702	MODBUS TCP 数据交换接收错误	请检查远程装置有支持 MODBUS 通讯协议
16#7002	CPU 模块不支持此功能	请检查 CPU 模块韧体版本
16#7203	无效的访问代码 (Access Code)	请检查远程装置送出的报文内容
16#7401	功能码 (Function Code) 错误	请检查远程装置送出的报文内容
16#7402	报文超出最大数据长度	请检查远程装置送出的报文内容
16#7407	命令中包含非 ASCII 字符	请检查远程装置送出的报文内容
16#7408	PLC 处于运行 (RUN) 模式	PLC 在 RUN mode 时不允许执行部份下载的动作，如程序、CPU 参数设定的下载。
16#740A	主机内存正在写入或写入失败	Flash/SD card 正在写入中，请稍后再重试。
16#740B	清除或重置动作正在进行中	PLC 正在执行 RST/CLR，请稍后再重试。
16#740C	通讯命令中的背板编号不正确	请确认主机韧体及软件版本并联络原厂
16#740D	通讯命令中的插槽编号不正确	请确认主机韧体及软件版本并联络原厂

错误代码	说明	处置方式
16#740E	清除内存的过程发生错误	请重试，若一再发生请联络原厂
16#740F	通讯逾时	请检查远程装置是否运作正常
16#7410	回复命令的功能码 (Function Code) 不一致	请检查远程装置送出的报文内容
16#7412	因 SW1 ON 所以数据无法下载至 CPU 模块	请确认 SW1 状态为 OFF
16#757D	输入 PLC 密码的剩余次数为 0	超过密码输入次数，请重上电。
16#757E	输入的 PLC 密码错误	请确认密码是否正确
16#8105	下载的项目程序内容有误：下载的程序语法错误	重新下载程序
16#8106	下载的项目程序内容有误：执行码超过限制长度	重新下载程序
16#8107	下载的项目程序内容有误：原始码超过限制长度	重新下载程序
16#8230	下载的主机参数有误：IP 地址不合法	请重新确认所下载的网络相关设定参数
16#8231	下载的主机参数有误：网络屏蔽地址不合法	请重新确认所下载的网络相关设定参数
16#8232	下载的主机参数有误：网关地址不合法	请重新确认所下载的网络相关设定参数
16#8233	下载的主机参数有误：IP 地址过滤设定错误	请重新确认所下载的网络相关设定参数
16#8235	下载的主机参数有误：静态 ARP 表错误	1.确认 HWCONFIG 中，CPU 模块的以太网网络 (Ethernet) 参数。 2.确认目前使用的 HWCONFIG 版本与主机版本是否兼容
16#8236	下载的主机参数有误：NTP 设定错误	1. 确认 HWCONFIG 中，CPU 模块的以太网网络 (Ethernet) 参数。 2. 确认目前使用的 HWCONFIG 版本与主机版本是否兼容
16#8240	下载的主机参数有误：Ether iLink	修改设定后重新下载

错误代码	说明	处置方式
16#8522	自动扫描检测执行中	模块布局检测正在进行
16#853B	IO 模块未配置	确认 HWCONFIG 中，模块配置是否正确。
16#853C	IO 模块不存在	确认 HWCONFIG 中，模块配置是否正确。
16#854B	IO 模块未配置	确认 HWCONFIG 中，模块配置是否正确。
16#854C	IO 模块不存在	确认 HWCONFIG 中，模块配置是否正确。
16#8572	模块配置表检查码错误	请确认主机固件及软件版本并联络原厂
16#8576	模块参数设定检查码错误	请确认主机固件及软件版本并联络原厂
16#867A	模块参数映像表检查码错误	请确认主机固件及软件版本并联络原厂
16#85E1	IO 中断编号不正确	请确认模块固件版本并联络原厂
16#85E2	IO 中断服务程序不存在	确认是否有下载对应的中断程序到 CPU
16#860F	系统还原错误	因系统复制文件内容有错，或该文件不存在于指定路径中。若已存在但无法完成执行，请重新产生系统复制文件。若一再出现此信息请联络原厂。
16#8611	储存卡不存在，或储存卡格式错误	无法检测到储存卡，请将储存卡格式化后重试
16#9A33	PLC Link / COM1 MODBUS 从站 19 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9A34	PLC Link / COM1 MODBUS 从站 20 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9A35	PLC Link / COM1 MODBUS 从站 21 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9A47	PLC Link / COM1 MODBUS 从站 7 无回应 (错误标志 SM1591)	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B01	COM2 Modbus 初始化错误	重设 HWCONFIG 的 COM2 Modbus 参数
16#9B21	COM2 MODBUS 从站 1 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B22	COM2 MODBUS 从站 2 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B23	COM2 MODBUS 从站 3 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。

错误代码	说明	处置方式
16#9B24	COM2 MODBUS 从站 4 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B25	COM2 MODBUS 从站 5 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B26	COM2 MODBUS 从站 6 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B27	COM2 MODBUS 从站 7 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B28	COM2 MODBUS 从站 8 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B29	COM2 MODBUS 从站 9 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B2A	COM2 MODBUS 从站 10 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B2B	COM2 MODBUS 从站 11 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B2C	COM2 MODBUS 从站 12 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B2D	COM2 MODBUS 从站 13 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B2E	COM2 MODBUS 从站 14 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B2F	COM2 MODBUS 从站 15 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B30	COM2 MODBUS 从站 16 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B31	COM2 MODBUS 从站 17 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B32	COM2 MODBUS 从站 18 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B33	COM2 MODBUS 从站 19 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。

错误代码	说明	处置方式
16#9B34	COM2 MODBUS 从站 20 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B35	COM2 MODBUS 从站 21 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B36	COM2 MODBUS 从站 22 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B37	COM2 MODBUS 从站 23 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B38	COM2 MODBUS 从站 24 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B39	COM2 MODBUS 从站 25 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B3A	COM2 MODBUS 从站 26 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B3B	COM2 MODBUS 从站 27 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B3C	COM2 MODBUS 从站 28 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B3D	COM2 MODBUS 从站 29 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B3E	COM2 MODBUS 从站 30 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B3F	COM2 MODBUS 从站 31 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B40	COM2 MODBUS 从站 32 通讯错误	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B41	COM2 MODBUS 从站 1 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B42	COM2 MODBUS 从站 2 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B43	COM2 MODBUS 从站 3 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。

错误代码	说明	处置方式
16#9B44	COM2 MODBUS 从站 4 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B45	COM2 MODBUS 从站 5 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B46	COM2 MODBUS 从站 6 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B47	COM2 MODBUS 从站 7 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B48	COM2 MODBUS 从站 8 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B49	COM2 MODBUS 从站 9 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B4A	COM2 MODBUS 从站 10 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B4B	COM2 MODBUS 从站 11 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B4C	COM2 MODBUS 从站 12 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B4D	COM2 MODBUS 从站 13 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B4E	COM2 MODBUS 从站 14 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B4F	COM2 MODBUS 从站 15 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B50	COM2 MODBUS 从站 16 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B51	COM2 MODBUS 从站 17 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B52	COM2 MODBUS 从站 18 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B53	COM2 MODBUS 从站 19 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。

错误代码	说明	处置方式
16#9B54	COM2 MODBUS 从站 20 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B55	COM2 MODBUS 从站 21 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B56	COM2 MODBUS 从站 22 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B57	COM2 MODBUS 从站 23 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B58	COM2 MODBUS 从站 24 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B59	COM2 MODBUS 从站 25 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B5A	COM2 MODBUS 从站 26 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B5B	COM2 MODBUS 从站 27 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B5C	COM2 MODBUS 从站 28 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B5D	COM2 MODBUS 从站 29 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B5E	COM2 MODBUS 从站 30 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B5F	COM2 MODBUS 从站 31 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。
16#9B60	COM2 MODBUS 从站 32 无回应	1. 检查两方的通讯设置。 2. 检查通讯电缆。

模拟 I/O 模块及温度模块故障排除

错误代码	说明	处置方式
16#A000	CH0 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH0 输入信号是否超出规格。
16#A001	CH1 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH1 输入信号是否超出规格。

错误代码	说明	处置方式
16#A002	CH2 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH2 输入信号是否超出规格。
16#A003	CH3 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH3 输入信号是否超出规格。
16#A004	CH4 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH4 输入信号是否超出规格。
16#A005	CH5 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH5 输入信号是否超出规格。
16#A006	CH6 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH6 输入信号是否超出规格。
16#A007	CH7 输入信号超出硬件规格 (模块 ERROR 灯设为闪烁)	确认 HWCONFIG 中的模块参数： 检查 CH7 输入信号是否超出规格。
16#A400	CH0 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH0 输入信号是否超出规格。
16#A401	CH1 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH1 输入信号是否超出规格。
16#A402	CH2 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH2 输入信号是否超出规格。
16#A403	CH3 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH3 输入信号是否超出规格。
16#A404	CH4 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH4 输入信号是否超出规格。
16#A405	CH5 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH5 输入信号是否超出规格。
16#A406	CH6 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH6 输入信号是否超出规格。
16#A407	CH7 输入信号超出硬件规格 (模块 ERROR 灯设为常亮)	确认 HWCONFIG 中的模块参数： 检查 CH7 输入信号是否超出规格。
16#A600	插槽电源异常	1. 检查背板是否正常。 2. 检查模块是否正常工作。
16#A601	电源异常	检查模块上，由外部提供的 24V 电源供电是否正常。
16#A602	内部错误，CJC 补偿异常	请联络原厂。
16#A603	内部错误，出厂校正异常	请联络原厂。
16#A800	CH0 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH0 输入信号是否超出规格。
16#A801	CH1 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH1 输入信号是否超出规格。

错误代码	说明	处置方式
16#A802	CH2 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH2 输入信号是否超出规格。
16#A803	CH3 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH3 输入信号是否超出规格。
16#A804	CH4 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH4 输入信号是否超出规格。
16#A805	CH5 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH5 输入信号是否超出规格。
16#A806	CH6 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH6 输入信号是否超出规格。
16#A807	CH7 输入信号超出硬件规格 (模块 ERROR 灯设为 OFF)	确认 HWCONFIG 中的模块参数： 检查 CH7 输入信号是否超出规格。

AH02HC-5A/AH04HC-5A

错误代码	说明	处置方式
16#A001	CH0 线性累加超过范围	须在程序中利用 FROM/TO 指令将 CR0 参数的 bit 1 设为 ON，以清除线性累加计数值。
16#A002	CH0 前置比例值设定超过范围	确认 HWCONFIG 中的模块参数： CH0 前置比例值符合 0~32767 范围内。
16#A003	CH0 移动平均值设定超过范围	确认 HWCONFIG 中的模块参数： CH0 移动平均值设定符合 2~60 范围内。
16#A004	CH0 比较值设定超过范围	确认 HWCONFIG 中的模块参数： CH0 比较值设定介于-999999999~999999999 之间。
16#A005	CH0 警报输出设定极限值错误	确认 HWCONFIG 中的模块参数： CH0 警报输出设定极限值介于-200000~200000。
16#A006	CH0 中断编号设定超过范围	确认 HWCONFIG 中的模块参数： CH0 中断编号设定介于 0~31。
16#A011	CH1 线性累加超过范围	须在程序中利用 FROM/TO 指令将 CR28 参数的 bit 1 设为 ON，清除线性累加计数值。
16#A012	CH1 前置比例值设定超过范围	确认 HWCONFIG 中的模块参数： CH1 前置比例值符合 0~32767 范围内。
16#A013	CH1 移动平均值设定超过范围	确认 HWCONFIG 中的模块参数： CH1 移动平均值设定符合 2~60 范围内。
16#A014	CH1 比较值设定超过范围	确认 HWCONFIG 中的模块参数： CH1 比较值设定介于-999999999~999999999 之间。
16#A015	CH1 警报输出设定极限值错误	确认 HWCONFIG 中的模块参数： CH1 警报输出设定极限值介于-200000~200000。

错误代码	说明	处置方式
16#A016	CH1 中断编号设定超过范围	确认 HWCONFIG 中的模块参数： CH1 中断编号设定介于 0~31。
16#A021	CH2 线性累加超过范围	须在程序中利用 FROM/TO 指令将 CR56 参数的 bit 1 设为 ON，以清除线性累加计数值。
16#A022	CH2 前置比例值设定超过范围	确认 HWCONFIG 中的模块参数： CH2 前置比例值符合 0~32767 范围内。
16#A023	CH2 移动平均值设定超过范围	确认 HWCONFIG 中的模块参数： CH2 移动平均值设定符合 2~60 范围内。
16#A024	CH2 比较值设定超过范围	确认 HWCONFIG 中的模块参数： CH2 比较值设定介于 -999999999~999999999 之间。
16#A025	CH2 警报输出设定极限值错误	确认 HWCONFIG 中的模块参数： CH2 警报输出设定极限值介于 -200000~200000。
16#A026	CH2 中断编号设定超过范围	确认 HWCONFIG 中的模块参数： CH2 中断编号设定介于 0~31。
16#A031	CH3 线性累加超过范围	须在程序中利用 FROM/TO 指令将 CR84 参数的 bit 1 设为 ON，清除线性累加计数值。
16#A032	CH3 前置比例值设定超过范围	确认 HWCONFIG 中的模块参数： CH3 前置比例值符合 0~32767 范围内。
16#A033	CH3 移动平均值设定超过范围	确认 HWCONFIG 中的模块参数： CH3 移动平均值设定符合 2~60 范围内。
16#A034	CH3 比较值设定超过范围	确认 HWCONFIG 中的模块参数： CH3 比较值设定介于 -999999999~999999999 之间。
16#A035	CH3 警报输出设定极限值错误	确认 HWCONFIG 中的模块参数： CH3 警报输出设定极限值介于 -200000~200000。
16#A036	CH3 中断编号设定超过范围	确认 HWCONFIG 中的模块参数： CH3 中断编号设定介于 0~31。

A5

AH05PM-5A/AH10PM-5A/AH15PM-5A

以下说明及处置方式须通过 PMSOFT V2.02 以上版本做设定

错误代码	说明	处置方式
16#A002	使用的子程序无内容	所指定子程序中必须撰写程序，而不得空白。
16#A003	CJ、CJN、JMP 指令缺少对应的指针	编写 CJ、CJN、JMP 指令要有对应的指针。
16#A004	主程序中有子程序指针	主程序中不能有子程序指针
16#A005	缺少子程序	不可调用不存在的子程序
16#A006	同一程序中的指针重复	同一程序中的指针不可重复
16#A007	子程序指针重复	子程序指针不可重复
16#A008	不同子程序中的跳转指令指针重复	不同子程序中的跳转指令指针不可重复

错误代码	说明	处置方式
16#A009	跳转指令与调用子程序指令使用相同指针	跳转指令指针与调用子程序的指针不得相同
16#A00A	跳转指令与子程序指令使用相同指针	跳转指令指针与子程序的指针不得相同
16#A00B	单段速目标位置 (I) 错误	正确设定单段速目标位置
16#A00C	单轴运动目标位置 (II) 错误	检查单轴运动两段速或是两段插入目标位置 (II) 与目标位置 (I) 位置方向是否相反
16#A00D	单轴运转速度 (I) 设定错误	设定单轴运动速度
16#A00E	单轴运转速度 (II) 设定错误	设定单轴第二段速运转速度 (II) 不为零
16#A00F	原点回归速度 (V_{RT}) 设定错误	调整回原点速度至适当值 (不可设为零)
16#A010	原点回归减速速度 (V_{CR}) 设定错误	调整回原点的速度参数，其减速速度必须小于回原点速度 (不可设为零)。
16#A011	寸动 JOG 速度设定错误	设定寸动 JOG 速度不为零
16#A012	单轴正转运动正向脉冲禁止输出	此状况系因极限 Sensor 被触发，请确认极限 Sensor 的状态以及电机运作是否在正常范围。
16#A013	单轴反向运动反向脉冲禁止输出	此状况系因极限 Sensor 被触发，请确认极限 Sensor 的状态以及电机运作是否在正常范围。
16#A014	到达极限	此状况系因极限 Sensor 被触发，请确认极限 Sensor 的状态以及电机运作是否在正常范围。
16#A015	装置组件使用范围错误	修改装置范围不超过使用范围
16#A017	V/Z 修饰错误	调整 V/Z 修饰为适当数值，避免超过范围。
16#A018	浮点数转换错误	修改程序运算避免导致异常数字出现
16#A019	BCD 转换错误	修改程序运算避免导致异常数字出现
16#A01A	除法运算错误 (除数 = 0)	修改程序运算避免导致除数为零
16#A01B	一般程序错误	修改程序符合文法
16#A01C	LD/LDI 指令连续使用 9 次以上	修改程序避免 LD/LDI 指令连续使用超过 9 次
16#A01D	RPT~RPE 超过 1 层以上	修改程序避免 RPT~RPE 指令使用超过 1 层以上
16#A01E	SRET 使用在 RPT~RPE 之间	修改程序避免 SRET 指令使用在 RPT~RPE 之间
16#A01F	主程序没有 M102 结束指令或运动程序没有 M2 结束指令	修改程序使主程序有 M102 结束指令或运动程序有 M2 结束指令
16#A020	使用错误指令或是使用装置超过范围	检查及修改程序避免使用错误指令或是确认使用装置是否超过范围

AH20MC-5A

以下说明及处置方式须通过 PMSOFT V2.02 以上版本做设定

错误代码	说明	处置方式
16#A002	使用的子程序无内容	所指定子程序中必须撰写程序，而不得空白。
16#A003	CJ、CJN、JMP 指令缺少对应的指针	编写 CJ、CJN、JMP 指令要有对应的指针。

错误代码	说明	处置方式
16#A004	主程序中有子程序指针	主程序中不能有子程序指针
16#A005	缺少子程序	不可调用不存在的子程序
16#A006	同一程序中的指针重复	同一程序中的指针不可重复
16#A007	子程序指针重复	子程序指针不可重复
16#A008	不同子程序中的跳转指令指针重复	不同子程序中的跳转指令指针不可重复
16#A009	跳转指令与调用子程序指令使用相同指针	跳转指令指针与调用子程序的指针不得相同
16#A00A	跳转指令与子程序指令使用相同指针	跳转指令指针与子程序的指针不得相同
16#A00B	单段速目标位置 (I) 错误	正确设定单段速目标位置
16#A00C	单轴运动目标位置 (II) 错误	检查单轴运动两段速或是两段插入目标位置 (II) 与目标位置 (I) 位置方向是否相反
16#A00D	单轴运转速度 (I) 设定错误	设定单轴运动速度
16#A00E	单轴运转速度 (II) 设定错误	设定单轴第二段速运转速度 (II) 不为零
16#A00F	原点回归速度 (V_{RT}) 设定错误	调整回原点速度至适当值 (不可设为零)
16#A010	原点回归减速速度 (V_{CR}) 设定错误	调整回原点的速度参数, 其减速速度必须小于回原点速度 (不可设为零)。
16#A011	寸动 JOG 速度设定错误	设定寸动 JOG 速度不为零
16#A012	单轴正转运动正向脉冲禁止输出	此状况系因极限 Sensor 被触发, 请确认极限 Sensor 的状态以及电机运作是否在正常范围。
16#A013	单轴反向运动反向脉冲禁止输出	此状况系因极限 Sensor 被触发, 请确认极限 Sensor 的状态以及电机运作是否在正常范围。
16#A014	到达极限	此状况系因极限 Sensor 被触发, 请确认极限 Sensor 的状态以及电机运作是否在正常范围。
16#A015	装置组件使用范围错误	修改装置范围不超过使用范围
16#A017	V/Z 修饰错误	调整 V/Z 修饰为适当数值, 避免超过范围。
16#A018	浮点数转换错误	修改程序运算避免导致异常数字出现
16#A019	BCD 转换错误	修改程序运算避免导致异常数字出现
16#A01A	除法运算错误 (除数 = 0)	修改程序运算避免导致除数为零
16#A01B	一般程序错误	修改程序符合文法
16#A01C	LD/LDI 指令连续使用 9 次以上	修改程序避免 LD/LDI 指令连续使用超过 9 次
16#A01D	RPT~RPE 超过 1 层以上	修改程序避免 RPT~RPE 指令使用超过 1 层以上
16#A01E	SRET 使用在 RPT~RPE 之间	修改程序避免 SRET 指令使用在 RPT~RPE 之间
16#A01F	主程序没有 M102 结束指令或运动程序没有 M2 结束指令	修改程序使主程序有 M102 结束指令或运动程序有 M2 结束指令
16#A020	使用错误指令或是使用装置超过范围	检查及修改程序避免使用错误指令或是确认使用装置是否超过范围

AH10EN-5A/AH15EN-5A

错误代码	说明	处置方式
16#A001	Host 1 IP 地址冲突	1. 联络网络管理人员并确认 IP 地址是否正确。 2. 检查 HWCONFIG 中的模块设定参数。
16#A002	Host 2 IP 地址冲突	1. 联络网络管理人员并确认 IP 地址是否正确。 2. 检查 HWCONFIG 中的模块设定参数。
16#A003	Host 1 DHCP 失败	请联络网络管理人员。
16#A004	Host 2 DHCP 失败	请联络网络管理人员。
16#A401	硬件错误	请恢复出厂默认值，若错误仍然存在，请联络原厂。
16#A402	系统初始化失败	请恢复出厂默认值，若错误仍然存在，请联络原厂。

AH10SCM-5A/AH15SCM-5A

错误代码	说明	处置方式
16#A002	UD Link 设定错误或是通讯失败	检查专属组态工具 SCMSOFT 中的设定，并试着重新下载。
16#A401	硬件发生错误	联系原厂。
16#A804	COM Port 通讯错误	1. 检查通讯电缆是否有接好。 2. 检查 HWCONFIG 与 SCMSOFT 中的设定参数，并重新下载。
16#A808	MODBUS 通讯错误	1. 检查通讯电缆是否有接好。 2. 检查 HWCONFIG 与 SCMSOFT 中的设定参数，并重新下载。

AH10DNET-5A

错误代码	说明	处置方式
16#A0F0	10DNET 扫描模块的站号与其它节点重复，或超出范围	确认 10DNET 扫描模块的节点站号在网络中是唯一的，更改节点站号后将其重新上电。
16#A0F1	没有将任何从站配置到 10DNET 扫描列表中	配置扫描列表，配置完成后下载至 10DNET。
16#A0F2	10DNET 扫描模块的工作电压过低	检查 10DNET 扫描模块以及 AH500 主机的工作电源是否正常
16#A0F3	10DNET 扫描模块进入测试模式	将模块上的功能开关 IN1 切换为 OFF 状态，并将 10DNET 重新上电。
16#A0F4	10DNET 扫描模块进入 Bus-OFF 状态	1. 检查网络通讯电缆是否正常、屏蔽线是否接地 2. 确认所有网络上的节点设备波特率是否一致 3. 检查网络的首尾两端是否都接有 121Ω 的终端电阻 4. 将扫描模块重新上电

错误代码	说明	处置方式
16#A0F5	10DNET 扫描模块检测到 DeviceNet 网络没有电源	检查网络电缆是否正常，并确认网络电源正常。
16#A0F6	10DNET 扫描模块的内部储存单元出错	将 10DNET 重新上电，若错误依然存在，请联络原厂。
16#A0F7	10DNET 扫描模块的数据交换单元出错	将 10DNET 重新上电，若错误依然存在，请联络原厂。
16#A0F8	10DNET 扫描模块序列号检测出错	将 10DNET 重新上电，若错误依然存在，请联络原厂。
16#A0F9	10DNET 扫描模块读取或写入配置数据出错	将 10DNET 重新上电，若错误依然存在，请联络原厂。
16#A0FA	10DNET 扫描模块的站号与扫描列表中配置的从站站号重复	方法一：重新设定 10DNET 扫描模块的站号，新站号不能与扫描列表中配置的从站站号重复。最后，将其重新上电。 方法二：扫描列表中不配置任何从站，再利用软件的『模拟在线』功能将空的配置数据下载到 10DNET 扫描模块。最后，将其重新上电。
16#A0FB	AH10DNET 和 AH CPU 之间数据交换失败	将 AH CPU 和 10DNET 重新上电，如果错误依然存在，请联络原厂。
16#A0FC	从站出错、AHRTU-DNET 背板插槽上的模块出错或 AHRTU-DNET 从背板连接不正常	<ol style="list-style-type: none"> 1. 检查从站的节点站号是否变化 2. 检查网络通讯电缆是否正常，如断路、松动等 3. 检查网络通讯线长度是否超过最远传输距离（请参考 AH500 模块手册第 11.3.3 节。超过最远传输距离后，将不能保证系统稳定。 4. 检查背板上的模块是否正常 5. 检查 AHRTU-DNET 从背板连接是否正常

A5

AH10PFBM-5A

错误代码	说明	处理方法
16#A001	主站设定为空	下载合适的设定
16#A003	主站进入检测模式	重新上电即可
16#A005	主站内部芯片通讯超时	重新下载合适的设定，如果错误依然存在，请联络原厂
16#A00B	与 PLC 数据交换超时	将 10PFBM 重新上电，如果错误依然存在，请联络原厂
16#A402	PLC 没有为主站分配 I/O 映射区	用 ISPSOFT 软件为主站分配合适的 I/O 映像区
16#A404	主站初始化错误	将 10PFBM 重新上电，如果错误依然存在，请联络原厂
16#A406	内部储存单元出错	将 10PFBM 重新上电，如果错误依然存在，请联络原厂

错误代码	说明	处理方法
16#A407	数据交换单元出错	将 10PFBS 重新上电，如果错误依然存在，请联络原厂
16#A408	主站序列号检测出错	将 10PFBS 重新上电，如果错误依然存在，请联络原厂
16#A4E2	主站检测到有从站掉线	1. 检视 PROFIBUS DP 总线连接是否正常 2. 检视网段两端是否加入终端电阻
16#A4E6	主站检测到 AHRTU-PFBS-5A 连接的模块出错	检查 AHRTU-PFBS-5A 连接的模块

AH10PFBS-5A

错误代码	说明	处理方法
16#A4F0	AH10PFBS-5A 节点地址超出范围	设置 AH10PFBS-5A 的节点地址在 1 ~ 125 之间
16#A4F1	内部硬件错误	重新上电，如果错误依然存在请更换一台新的 AH10PFBS-5A
16#A4F2	参数化错误	检查 AH10PFBS-5A 使用 GSD 档是否正确
16#A4F3	组态错误	检查 AH10PFBS-5A 使用 GSD 档是否正确
16#A4F4	GPIO 检测出错	重新上电，如果错误依然存在请更换一台新的 AH10PFBS-5A
16#A4F5	AH10PFBS-5A 进入工厂测试模式	设置 AH10PFBS-5A 的节点地址在 1 ~ 125 之间，重新上电
16#A4F6	1. AH10PFBS-5A 未接入 PFOFIBUS-DP 网络 2. PFOFIBUS-DP 主站没有配置 AH10PFBS-5A 从站或配置 AH10PFBS-5A 节点地址与实际连接的不符	1. 检查 AH10PFBS-5A 和 PROFIBUS DP 主站之间的通讯线连接是否正常 2. 检查 PROFIBUS DP 主站配置软件内有配置 AH10PFBS-5A 从站及配置的节点地址与实际连接的相符 3. 检查 PROFIBUS DP 主站工作是否正常

AH10COPM-5A

错误代码	说明	处理方法
16#A0B0	心跳信息超时	检查 CANopen 网络中总线线缆连接正常
16#A0B1	从站返回的 PDO 长度与与节点列表中设定的 PDO 数据长度不符	重新设定从站的 PDO 数据长度，设定完成后下载到 AH10COPM-5A。
16#A0B2	主站 NodeGuard 信息超时	检查 CANopen 网络中总线线缆连接正常
16#A0E0	AH10COPM-5A 接收到从站发送的紧急信息	通过 CANopen_EMCY 功能块读取相关信息
16#A0E1	从站返回的 PDO 数据长度与节点列表中设定的 PDO 数据长度不符	重新设定从站的 PDO 数据长度，设定完成后下载到 AH10COPM-5A。

错误代码	说明	处理方法
16#A0E2	未接收到从站 PDO	检查并确认设定正确
16#A0E3	自动 SDO 下载失败	检查并确认自动 SDO 正确
16#A0E4	PDO 参数设定失败	确认 PDO 参数设定合法
16#A0E5	关键参数设定有误	确认所连接的从站与所设定的从站一致
16#A0E6	实际网络配置与设定配置不符	确认从站工作电源及网络连接正常。
16#A0E7	从站错误控制超时	
16#A0E8	主从站站号重复	重新设定主站或从站站号并确认重新设定后的站号不重复。
16#A0F1	CANopen Builder 软件节点列表没有增加从站	将从站增加至节点列表后，重新下载配置到 AH10COPM-5A。
16#A0F3	AH10COPM-5A 处于错误状态	重新下载参数配置，如果错误依然存在，请更换一台新的 AH10COPM-5A。
16#A0F4	检测到总线脱离 (Bus-off)	请确认 CANopen 网络中总线线缆接线正确，并确认网络上所有的节点都有相同的波特率，然后将 AH10COPM-5A 重新上电。
16#A0F5	AH10COPM-5A 节点地址设定错误	设定 AH10COPM-5A 的节点地址在 1~127 之间
16#A0F6	内部错误：工厂制造流程出错	重新上电，如果错误依然存在，请更换一台新的 AH10COPM-5A。
16#A0F7	内部错误：GPIO 检测出错	
16#A0F8	内部错误：内部存储器检测出错	
16#A0F9	低电压检测错误	检查并确认 AH10COPM-5A 的工作电源正常
16#A0FA	AH10COPM-5A 韧体内部处于错误状态	重新上电 AH10COPM-5A
16#A0FB	AH10COPM-5A 的发送暂存区已满	请确认 CANopen 网络中总线线缆连接正常，再将 AH10COPM-5A 重新上电。
16#A0FC	AH10COPM-5A 的接收暂存区已满	请确认 CANopen 网络中总线线缆连接正常，再将 AH10COPM-5A 重新上电。

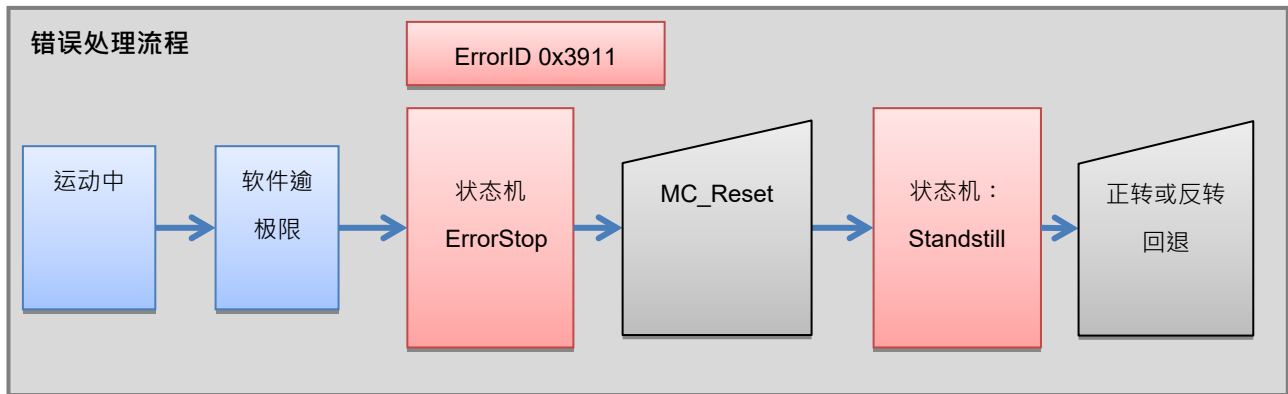
A.5.3 极限错误排除

软极限错误排除

错误码 0x3911 不论在运动中或是在启动前，都会检查对应轴的软极限设定。因为是 AHxxEMC 韧体控制，所以软件逾限时 Servo 并不会报错，只有轴会进入 ErrorStop。

注:软极限设定请参考 ISP 轴参数设定，须注意软极限范围不可设定在最大值范围上下 (含)

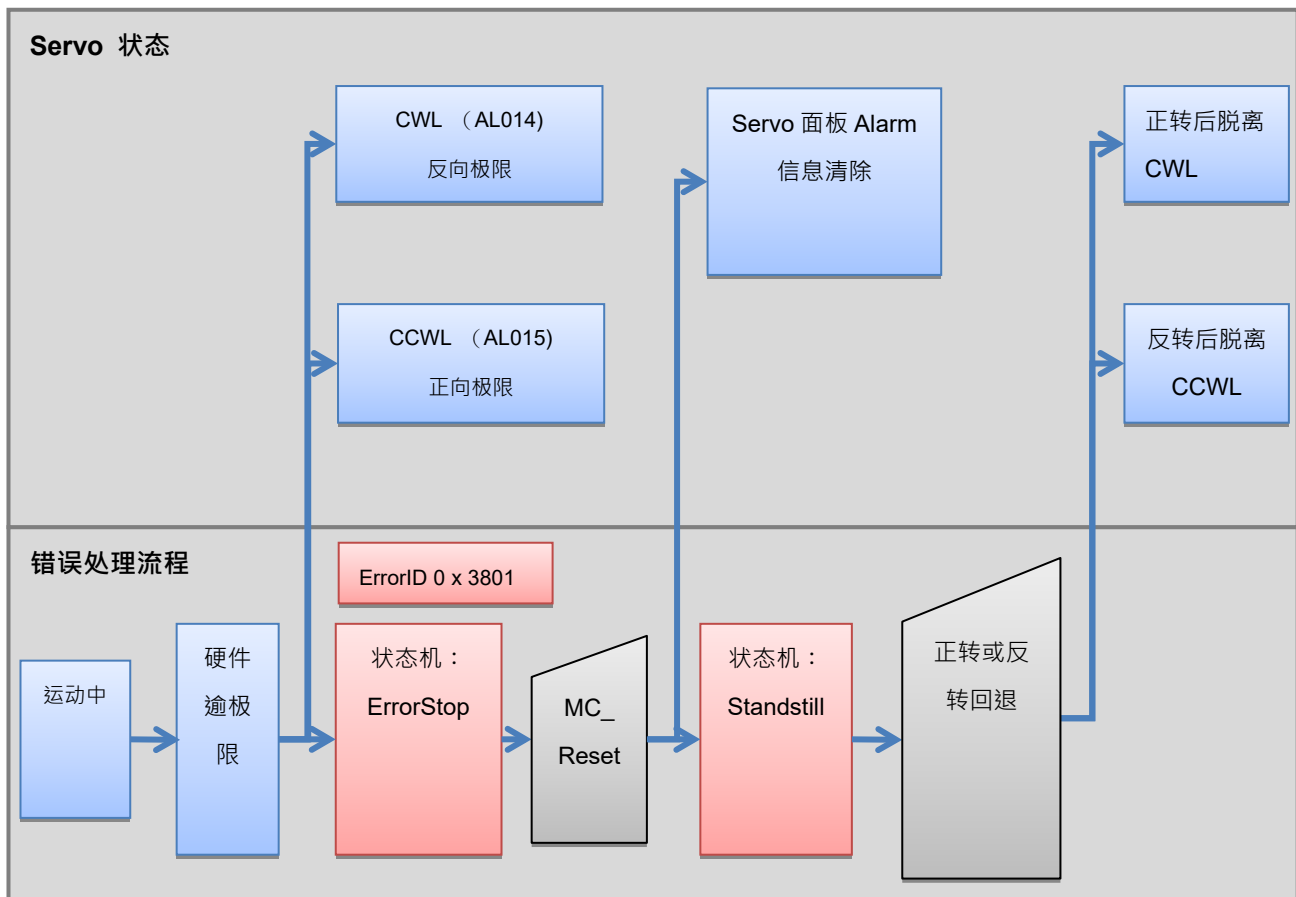
错误流程与排除方式



硬极限错误排除

当 Servo 运动时，无论运动方向，只要 CWL 或 CCWL 信号时 Servo 均会停止，并报 AL014(CWL) 或 AL015 (CCWL)。

错误流程与排除方式



A5

A.6 数据类型：枚举 (Enum)

数据类型	数值范围	说明	对应的功能块与其引脚名称
eDFB_OUTTY PE	0 : UD 1 : PD 2 : AB	0 : 输出脉冲型式为正反转脉冲 1 : 输出脉冲型式为脉冲和方向 2 : 输出脉冲型式为 AB 相脉冲	功能块 : DFB_AxisSetting2 引脚 : <i>OutputType</i>
eDFB_UNIT	0 : Motor 1 : Machine 2 : Compound	0 : 坐标系统为电机单位 1 : 坐标系统为机械单位 2 : 坐标系统为复合单位	功能块 : DFB_AxisSetting2 引脚 : <i>Unit</i>
eDFB_MODE	4096 : AxisIdle 256 : AxisStopping 4353 : AbsSeg1 4354 : RelSeg1 4355 : AbsSeg2 4356 : RelSeg2 4357 : TrSeg1 4358 : Jog 4359 : Mpg 4362 : GearIn 4363 : CamIn 4608 : GcodeStopping 4609 : GcodeRun 4864 : InterpolationStopping	0x000 : 轴闲置中 0x100 : 轴运动停止中 0x101 : 绝对单段速定位 0x102 : 相对单段速定位 0x103 : 绝对两段速定位 0x104 : 相对两段速定位 0x105 : 插入单段速定位 0x107 : 寸动运动 0x108 : 手摇轮模式运动 0x10A : 电子啮轮运动 0x10B : 电子凸轮运动 0x200 : Gcode 停止中 0x201 : Gcode 运动中 0x300 : 多轴插补停止中	功能块 : DFB_AxisStatus 引脚 : <i>Mode</i>
eDFB_ SDODataType	0 : mc16bits 1 : mc32bits	0 : 写入 16 位长度数据 1 : 写入 32 位长度数据	功能块 : DFB_ECATServoWrite 引脚 : <i>DataType</i>
eDFB_ SELECT_DEV	0 : M_DEV 5 : D_DEV 6 : W_DEV 7 : ALL	0 : 从 SD 卡读回 M 装置 5 : 从 SD 卡读回 D 装置 6 : 从 SD 卡读回 W 装置 7 : 从 SD 卡读回以上所有装置	功能块 : DFB_SDDevRead 引脚 : <i>Device</i>

数据类型	数值范围	说明	对应的功能块与其引脚名称
eDFB_ ENGAGE_TY PE	0 : ByCapture0 1 : ByCapture1 2 : ByCapture2 3 : ByCapture3 4 : ByCapture4 5 : ByCapture5 6 : ByCapture6 7 : ByCapture7 -1 : Direct	0 : 第 0 组捕捉器被触发时啮合 1 : 第 1 组捕捉器被触发时啮合 2 : 第 2 组捕捉器被触发时啮合 3 : 第 3 组捕捉器被触发时啮合 4 : 第 4 组捕捉器被触发时啮合 5 : 第 5 组捕捉器被触发时啮合 6 : 第 6 组捕捉器被触发时啮合 7 : 第 7 组捕捉器被触发时啮合 -1 : 直接啮合	功能块 : DFB_GearIn2/ DFB_CamIn2 引脚 : <i>extTrgCAPno</i>
eDFB_ ACC_CURVE	0 : Polynomial_0order 1 : Polynomial_1order 2 : SingleHypot 3 : Cycloid	0 : 常数曲线 1 : 一阶多项式曲线 2 : 单精度三角斜边曲线 3 : 摆线曲线	功能块 : DFB_CamCurve/ DFB_CamCurve2/ DFB_FlyCut2/ DFB_HorizontalFlowWrapper 引脚 : <i>AccCurve</i>
eDFB_ GEN_CURVE	0 : leftCAM 5 : rightCAM 1 : midCAMall 9 : midCAMzero 7 : midCAMbegin 8 : midCAMend	0 : 左边凸轮 5 : 右边凸轮 1 : 中间凸轮全部 9 : 中间凸轮零 7 : 中间凸轮开始 8 : 中间凸轮结束	功能块 : DFB_CamCurve / DFB_CamCurve2/ DFB_FlyCut2 引脚 : <i>eCamCurve</i>

数据类型	数值范围	说明	对应的功能块与其引脚名称
eDFB_HCNT	0 : AC0 1 : AC4 2 : AC8 3 : AC12 4 : AC16 6 : AC20	0 : 第一组高速计数器 1 : 第二组高速计数器 2 : 第三组高速计数器 3 : 第四组高速计数器 4 : 第五组高速计数器 6 : 第六组高速计数器	功能块 : DFB_HCcnt 引脚 : <i>Channel</i>
eDFB_HCNT_INTYPE	0 : UD 1 : PD 2 : AB 3 : AB4	0 : 正反转脉冲 1 : 脉冲和方向 2 : AB 相脉冲 3 : 4AB 相脉冲	功能块 : DFB_HCcnt 引脚 : <i>InputType</i>
eDFB_HTMR	0 : AC0 1 : AC4 2 : AC8 3 : AC12	0: 第一组高速定时器 1: 第二组高速定时器 2: 第三组高速定时器 3: 第四组高速定时器	功能块 : DFB_HTmr 引脚 : <i>Channel</i>
eDFB_COMP	0 : Ch0 1 : Ch1 2 : Ch2 3 : Ch3 4 : Ch4 5 : Ch5 6 : Ch6 7 : Ch7	0 : 频道 0 1 : 频道 1 2 : 频道 2 3 : 频道 3 4 : 频道 4 5 : 频道 5 6 : 频道 6 7 : 频道 7	功能块 : DFB_Compare 引脚 : <i>Channel</i>
eDFB_COMP_SOURCE	0 : Axis1 1 : Axis2 2 : Axis3 3 : Axis4 4 : AC0 5 : AC4 6 : AC8 7 : AC12 8 : AC16	0 : 第一轴 1 : 第二轴 2 : 第三轴 3 : 第四轴 4 : 第一组计数器 5 : 第二组计数器 6 : 第三组计数器 7 : 第四组计数器 8 : 第五组计数器	功能块 : DFB_Compare 引脚 : <i>Source</i>

数据类型	数值范围	说明	对应的功能块与其引脚名称
eDFB_ COMP_MODE	0 : Equal 1 : Bigger_Equal 2 : Smaller_Equal	0 : 相等 1 : 大于等于 2 : 小于等于	功能块 : DFB_Compare 引脚 : <i>Mode</i>
eDFB_ COMP_OUTD EV	0 : SetY08 1 : SetY09 2 : SetY10 3 : SetY11 4 : RstAC0 5 : RstAC4 6 : RstAC8 7 : RstAC12	0 : 设定 Y0.8 1 : 设定 Y0.9 2 : 设定 Y0.10 3 : 设定 Y0.11 4 : 清除第一组计数器计 数值 5 : 清除第二组计数器计 数值 6 : 清除第三组计数器计 数值 7 : 清除第四组计数器计 数值	功能块 : DFB_Compare 引脚 : <i>OutPutDevice</i>
eDFB_CAP	0 : Ch0 1 : Ch1 2 : Ch2 3 : Ch3 4 : Ch4 5 : Ch5 6 : Ch6 7 : Ch7	0 : 频道 0 1 : 频道 1 2 : 频道 2 3 : 频道 3 4 : 频道 4 5 : 频道 5 6 : 频道 6 7 : 频道 7	功能块 : DFB_Capture/ DFB_Capture2 引脚 : <i>Source</i>

A6

数据类型	数值范围	说明	对应的功能块与其引脚名称
eDFB_ CAP_TRIG_D EV	0 : X0p0 1 : X0p1 2 : X0p2 3 : X0p3 8 : X0p8 9 : X0p9 10 : X0p11 11 : X0p11 12 : X0p12 13 : X0p13 14 : X0p14	0 : 使用 X0.0 信号触发 1 : 使用 X0.1 信号触发 2 : 使用 X0.2 信号触发 3 : 使用 X0.3 信号触发 8 : 使用 X0.8 信号触发 9 : 使用 X0.9 信号触发 10 : 使用 X0.10 信号触发 11 : 使用 X0.11 信号触发 12 : 使用 X0.12 信号触发 13 : 使用 X0.13 信号触发 14 : 使用 X0.14 信号触发	功能块 : DFB_Capture/ DFB_Capture2 引脚 : <i>TriggerDevice</i>
eDFB_ CAP_SOURC E	0 : Axis1 1 : Axis2 2 : Axis3 3 : Axis4 4 : AC0 5 : AC4 6 : AC8 7 : AC12 8 : AC16	0 : 捕捉第一轴 1 : 捕捉第二轴 2 : 捕捉第三轴 3 : 捕捉第四轴 4 : 捕捉第一组计数器 5 : 捕捉第二组计数器 6 : 捕捉第三组计数器 7 : 捕捉第四组计数器 8 : 捕捉第五组计数器	功能块 : DFB_Capture/ DFB_Capture2 引脚 : <i>Source</i>
eDFB_ HALT_CLK_S OURCE	0 : slaveEOP 1 : masterEOP 2 : extern	0 : 从轴凸轮结束点 1 : 主轴凸轮结束点 2 : 功能块输入引脚	功能块 : DFB_FlyCut2 引脚 : <i>Halt_ClkSource</i>

数据类型	数值范围	说明	对应的功能块与其引脚名称
eMC_STATE_MACHINE	0 : Unknown 1 : ErrorStop 2 : Disabled 3 : Standstill 4 : Homing 5 : Stopping 6 : ContinuousMotion 7 : SynchronizedMotion 8 : DiscreteMotion	0 : 未知 1 : 错误 2 : 未启动 3 : 启动 4 : 原点 5 : 停止中 6 : 连续运动 7 : 同步运动 8 : 点动	--
eMC_BUFFER_MODE	0 : mcAborting 1 : mcBuffered 2 : mcBlendingLow 3 : mcBlendingPrevious 4 : mcBlendingNext 5 : mcBlendingHigh	0 : 前动作停止 1 : 等待前动作完成 2 : 接续以低的速度运行 3 : 接续以前一个速度运行 4 : 接续以下一个速度运行 5 : 接续以高的速度运行	引脚 : <i>BufferMode</i>
eMC_DIRECTION	1 : mcPositiveDirection 2 : mcShortestWay 3 : mcNegativeDirection 4 : mcCurrentDirection	1 : 正向 2 : 最短 3 : 反向 4 : 当前方向	引脚 : <i>Direction</i>
eMC_SOURCE	0 : mcCommandedValue 1 : mcSetValue 2 : mcActualValue	0 : 命令值 1 : 设定值 2 : 实际值	功能块 : MC_ReadMotionState 引脚 : <i>Source</i> 功能块 : MC_CamIn/ MC_GearIn/ MC_GearInPos 引脚 : <i>MasterValueSource</i> 功能块 : MC_CombineAxes 引脚 : <i>MasterValueSourceM1/</i> <i>MasterValueSourceM2</i> 功能块 : MC_DigitalCamSwitch 引脚 : <i>ValueSource</i>

数据类型	数值范围	说明	对应的功能块与其引脚名称
eMC_ SYNC_MODE	1 : mcRampIn_Shortest 2 : mcRampIn_Positive 3 : mcRampIn_Negative	1 : (不支持) 2 : (不支持) 3 : (不支持)	功能块 : MC_GearInPos 引脚 : <i>SyncMode</i>
eMC_ START_MODE	0 : mcJump 1 : mcRampIn_Shortest 2 : mcRampIn_Positive 3 : mcRampIn_Negative 4 : mcAbsolute 5 : mcRelative	0 : 立即进入 1 : 最短路径 2 : 正向路径 3 : 反向路径 4 : (不支持) 5 : (不支持)	功能块 : MC_CamIn 引脚 : <i>StartMode</i>
eMC_ COMBINE_MODE	1 : mcAddAxes 2 : mcSubAxes	1 : 相加 2 : 相减	功能块 : MC_CombineAxes 引脚 : <i>CombineMode</i>

MEMO

A6