

客服热线 (021) 5863-9595

绵密网络 专业服务

中达电通已建立了 41 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)6255-010 传真:(0791)6255-102	合肥 电话:(0551)2816-777 传真:(0551)2816-555	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)8836-0640 传真:(029)88360640-8000	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			

AH500 程序手册



AH500 程序手册



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-0109710-03
2017-12-29

中达电通公司版权所有
如有改动,恕不另行通知

www.delta.com.tw/ia



AH500 程序手册

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	2012/06/30
第二版	<ol style="list-style-type: none"> 第 1 章：更新第 1.1.2 节机种简易说明内容 第 2 章：更新第 2.1.1 节装置列表、第 2.1.4 节停电保持区的装置范围、第 2.2.3 节字符串、第 2.2.7 节特殊辅助继电器 SM、第 2.2.8 节特殊辅助继电器 SM 的更新时间、第 2.2.14 节特殊数据寄存器 SR、第 2.2.15 节特殊数据寄存器 SR 的更新时间及第 2.2.16 节 SM/SR 补充说明 第 3 章：增加指令 API0117、API0118、API0708、API1812。更新指令 API2207 及删除 API1809~API1811 相关数据 第 4 章：更新第 4.2 节中功能块中不支持指令及范例说明 第 6 章：增加指令 API0117、API0118、API0708、API1812、API2500、API2501、API2502 及删除 API1809~API1811 相关数据；更新指令 API0114、指令 API0204、API0205、API0212、API0217、API0218、API0219、API0310、API0400、API0702、API0703、API0705、API0707、API0812~API0817、API1000、API1001、API1002、API1003、API1004、API1005、API1007、API1202、API1205、API1301、API1510、API1514、API1516、API1603、API1701、API1702、API1703、API1704、API1800、API1803、API1806、API1807、API2100、API2103、API2108、API2110、API2200、API2201、API2202、API2203、API2204、API2205、API2207、API2300、API2301 第 7 章：更新第 7.1.1 节、第 7.1.2 节、第 7.1.3 节、第 7.1.4 节、第 7.1.8 节、第 7.1.9 节、第 7.1.10 节内容 	2016/05/05
第三版	<ol style="list-style-type: none"> 第 1 章：更新第 1.1.1 节相关手册说明、增加 AH15SCM-5A/AHRTU-ETHN-5A/AH15EN-5A/AHCPU501-RS2/AHCPU521-RS2/AHCPU531-RS2/AHCPU501-EN 机种相关信息及将所有配线模块型号变更为新型号 第 2 章：更新第 2.2.3 节 ASCII 码转换表、第 2.2.7 节增加机种 AHCPU560-EN2/AHCPU501-RS2/AHCPU521-RS2/AHCPU531-RS2/AHCPU501-EN 相关信息及新增/更新 SM、第 2.2.8 节特殊辅助继电器 SM 的更新时间、第 2.2.14 节特殊数据寄存器 SR、 	2017/12/29

版本	变更内容	发行日期
	<p>第 2.2.15 节特殊数据寄存器 SR 的更新时间及第 2.2.16 节 SM/SR 补充说明</p> <p>3. 第 3 章：第 3.1.2 节增加 API2900/API2901 及第 3.3 节更新 INV 指令符号</p> <p>4. 第 6 章：更新指令 API0702 指令说明、API0904 程序范例二说明、API1111 补充说明、API1503/1504/1505 程序范例图、API1701/1702 指令说明及程序范例图、API1800/1812 指令说明、API2116 指令装置、API2200 指令说明、API2300/2301/2302 指令说明及增加 API2900/API2901</p> <p>5. 第 7 章：更新第 7.1.1 节内容及增加 AH15SCM-5A/AH15EN-5A 机种相关信息</p>	

AH500 程序手册

目录

第 1 章 简介

1.1 一般概述	1-2
1.1.1 相关手册	1-2
1.1.2 机种简易说明	1-3
1.2 软件简述	1-9
1.2.1 编辑软件	1-9
1.2.2 程序组织单元 (POU) 与工作 (TASK)	1-11

第 2 章 装置说明

2.1 装置简介	2-2
2.1.1 装置列表	2-2
2.1.1.1 AH500 基本型 CPU 模块 (AHCPU500/510/520/ 530)	2-2
2.1.1.2 AH500 升级型 CPU 模块 (AHCPU501/511/521/531)	2-4
2.1.2 I/O 储存区的基本结构	2-5
2.1.3 停电保持储存方式	2-6
2.1.4 停电保持区的装置范围	2-6
2.2 装置功能说明	2-7
2.2.1 数值、常数	2-7
2.2.2 浮点数	2-10
2.2.2.1 单精度浮点数 (32 位浮点数)	2-10
2.2.2.2 双精度浮点数 (64 位浮点数)	2-10
2.2.2.3 十进浮点数	2-12
2.2.3 字符串	2-12
2.2.4 输入继电器 X	2-14
2.2.5 输出继电器 Y	2-14
2.2.6 辅助继电器 M	2-15
2.2.7 特殊辅助继电器 SM	2-15
2.2.8 特殊辅助继电器 SM 的更新时间	2-47

2.2.9	步进点继电器 S	2-55
2.2.10	定时器 T.....	2-55
2.2.11	计数器 C.....	2-57
2.2.12	32 位计数器 HC.....	2-58
2.2.13	数据寄存器 D	2-60
2.2.14	特殊数据寄存器 SR.....	2-60
2.2.15	特殊数据寄存器 SR 的更新时间	2-96
2.2.16	SM/SR 补充说明.....	2-98
2.2.17	连结寄存器 L.....	2-114
2.2.18	变址寄存器 E	2-114

第 3 章 指令表

3.1	指令类型.....	3-2
3.1.1	基本指令	3-2
3.1.2	应用指令	3-2
3.2	指令表说明	3-3
3.2.1	基本指令	3-3
3.2.2	应用指令 (依 API 号码排序)	3-4
3.2.3	应用指令 (依英文字母排序)	3-5
3.2.4	指令装置表说明	3-6
3.3	基本指令一览表.....	3-7
3.4	应用指令一览表.....	3-10
3.4.1	应用指令 (依 API 号码排序)	3-10
3.4.2	应用指令 (依英文字母排序)	3-42

第 4 章 指令结构

4.1	API 应用指令组成说明	4-2
4.2	指令使用限制	4-5
4.3	变址说明.....	4-7
4.4	指针寄存器 (PR) 说明.....	4-8
4.5	定时器指针寄存器 (TR) 说明	4-10
4.6	16 位计数器指针寄存器 (CR) 说明	4-12
4.7	32 位计数器指针寄存器 (HCR) 说明	4-13

第 5 章 基本指令	
5.1 基本指令一览表.....	5-2
5.2 基本指令说明	5-3
第 6 章 指令表	
6.1 比较操作指令	6-4
6.1.1 比较操作指令一览表.....	6-4
6.1.2 比较操作指令说明	6-7
6.2 四则运算指令	6-36
6.2.1 四则运算指令一览表.....	6-36
6.2.2 四则运算指令说明	6-37
6.3 数据转换指令	6-73
6.3.1 数据转换指令一览表.....	6-73
6.3.2 数据转换指令说明	6-74
6.4 数据转移指令	6-110
6.4.1 数据转移指令一览表.....	6-110
6.4.2 数据转移指令说明	6-111
6.5 程序跳转指令	6-134
6.5.1 程序跳转指令一览表.....	6-134
6.5.2 程序跳转指令说明	6-135
6.6 程序执行控制指令.....	6-143
6.6.1 程序执行控制指令一览表	6-143
6.6.2 程序执行控制指令说明.....	6-144
6.7 I/O 更新指令.....	6-150
6.7.1 I/O 更新指令一览表.....	6-150
6.7.2 I/O 更新指令说明.....	6-151
6.8 便利指令.....	6-152
6.8.1 便利指令一览表	6-152
6.8.2 便利指令说明.....	6-153
6.9 逻辑操作指令	6-187
6.9.1 逻辑操作指令一览表.....	6-187
6.9.2 逻辑操作指令说明	6-188
6.10 旋转指令	6-207

6.10.1	旋转指令一览表	6-207
6.10.2	旋转指令说明	6-208
6.11	基本指令	6-218
6.11.1	基本指令一览表	6-218
6.11.2	基本指令说明	6-219
6.12	位移指令	6-226
6.12.1	位移指令一览表	6-226
6.12.2	位移指令说明	6-227
6.13	数据处理指令	6-252
6.13.1	数据处理指令一览表	6-252
6.13.2	数据处理指令说明	6-253
6.14	结构建立指令	6-297
6.14.1	结构建立指令一览表	6-297
6.14.2	结构建立指令说明	6-298
6.15	模块的数据读写指令	6-305
6.15.1	模块的数据读写指令一览表	6-305
6.15.2	模块的数据读写指令说明	6-306
6.16	浮点数指令	6-311
6.16.1	浮点数指令一览表	6-311
6.16.2	浮点数指令说明	6-312
6.17	万年历指令	6-346
6.17.1	万年历指令一览表	6-346
6.17.2	万年历指令说明	6-347
6.18	外围设备指令	6-360
6.18.1	外围设备指令一览表	6-360
6.18.2	外围设备指令说明	6-361
6.19	通讯指令	6-374
6.19.1	通讯指令一览表	6-374
6.19.2	通讯指令说明	6-375
6.20	其它指令	6-406
6.20.1	其它指令一览表	6-406
6.20.2	其它指令说明	6-407
6.21	字符串处理指令	6-416

6.21.1	字符串处理指令一览表	6-416
6.21.2	字符串处理指令说明	6-417
6.22	以太网控制指令	6-476
6.22.1	以太网控制指令一览表	6-476
6.22.2	以太网控制指令说明	6-477
6.23	存储卡读写指令	6-502
6.23.1	存储卡读写指令一览表	6-502
6.23.2	存储卡读写指令说明	6-503
6.24	任务控制指令	6-515
6.24.1	任务控制指令一览表	6-515
6.24.2	任务控制指令说明	6-516
6.25	SFS 控制指令	6-518
6.25.1	SFS 控制指令一览表	6-518
6.25.2	SFS 控制指令说明	6-519
6.26	冗余系统控制指令	6-525
6.26.1	冗余系统控制指令一览表	6-525
6.26.2	冗余系统控制指令说明	6-526

第 7 章 错误代码

7.1	模块错误代码对应灯号及状态说明	7-2
7.1.1	CPU 模块错误代码对应灯号及状态	7-2
7.1.2	模拟模块与温度模块错误代码对应灯号	7-21
7.1.3	AH02HC-5A/AH04HC-5A 错误代码对应灯号	7-22
7.1.4	AH05PM-5A/AH10PM-5A/AH15PM-5A 错误代码对应灯号	7-23
7.1.5	AH20MC-5A 错误代码对应灯号	7-24
7.1.6	AH10EN-5A 错误代码对应灯号	7-25
7.1.7	AH10SCM-5A 错误代码对应灯号	7-25
7.1.8	AH10DNET-5A 错误代码对应灯号	7-25
7.1.9	AH10PFBM-5A 错误代码对应灯号	7-26
7.1.10	AH10PFBS-5A 错误代码对应灯号	7-27
7.1.11	AH10COPM-5A 错误代码对应灯号	7-28



第1章 简介

目录

1.1 一般概述	1-2
1.1.1 相关手册	1-2
1.1.2 机种简易说明	1-3
1.2 软件简述	1-9
1.2.1 编辑软件	1-9
1.2.2 程序组织单元 (POU) 与工作 (TASK)	1-11

1

1.1 一般概述

本手册描述 AH500 系列 PLC 主机程序设计相关的内容介绍，基本指令及应用指令操作方法说明。

1.1.1 相关手册

AH500 系列 PLC 相关手册的组成如下表所示。请搭配使用。

- **AH500 快速入门手册：**
主要介绍如何让用户在详细阅读完所有手册之前，能先快速的建立并使用本系统。
- **AH500 程序手册：**
描述AH500系列PLC主机程序设计相关的内容介绍，基本指令及应用指令操作方法说明。
- **ISPSoft 使用手册：**
有关ISPSoft操作、程序语言介绍 (Ladder、IL、SFC、FBD、ST)、POU概念及工作 (Task) 概念等。
- **AH500 硬件手册：**
电气规格、外观及尺寸等。
- **AH500 操作手册：**
CPU功能介绍、各项装置、I/O配置及故障排除等。
- **AH500 模块手册：**
特殊模块使用介绍，例如网络模块、模拟模块及温度模块等。
- **AH500 运动控制模块手册：**
主要介绍运动控制模块的规格、配线、指令及功能范例。
- **PMSoft 使用手册：**
主要介绍运动控制模块编辑软件的详细使用，包含编辑模式、联机及加密设定等内容。
- **AH500 冗余系统操作手册：**
描述 AH500 冗余系统架构、建立、程序设计及操作说明。

1.1.2 机种简易说明

分类	机种名称	说明
电源模块	AHPS05-5A	100-240VAC 50/60Hz 电源模块
	AHPS15-5A	24VDC 电源模块
CPU 模块	AHCPU500-RS2	基本型 CPU 模块，内建 RS-485*2、USB 通讯端口以及 SD 卡接口，支持 768 点 I/O，程序容量 32k steps。
	AHCPU500-EN	基本型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 768 点 I/O，程序容量 32k steps。
	AHCPU501-RS2	升级型 CPU 模块，内建 RS-485 * 2、USB 通讯端口以及 SD 卡接口，支持 768 点 I/O，程序容量 48k steps。
	AHCPU501-EN	升级型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 768 点 I/O，程序容量 48k steps。
	AHCPU510-RS2	基本型 CPU 模块，内建 RS-485*2、USB 通讯端口以及 SD 卡接口，支持 1280 点 I/O，程序容量 64k steps。
	AHCPU510-EN	基本型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 1280 点 I/O，程序容量 64k steps。
	AHCPU511-RS2	升级型 CPU 模块，内建 RS-485 * 2、USB 通讯端口以及 SD 卡接口，支持 1280 点 I/O，程序容量 96k steps。
	AHCPU511-EN	升级型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 1280 点 I/O，程序容量 96k steps。
	AHCPU520-RS2	基本型 CPU 模块，内建 RS-485*2、USB 通讯端口以及 SD 卡接口，支持 2304 点 I/O，程序容量 128k steps。
	AHCPU520-EN	基本型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 2304 点 I/O，程序容量 128k steps。
	AHCPU521-RS2	升级型 CPU 模块，内建 RS-485 * 2、USB 通讯端口以及 SD 卡接口，支持 2304 点 I/O，程序容量 192k steps。
	AHCPU521-EN	升级型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 2304 点 I/O，程序容量 192k steps。
	AHCPU530-RS2	基本型 CPU 模块，内建 RS-485*2、USB 通讯端口以及 SD 卡接口，支持 4352 点 I/O，程序容量 256k steps。
	AHCPU530-EN	基本型 CPU 模块，内建以太网、RS-485、USB 通讯端

1

分类	机种名称	说明
		口以及 SD 卡接口，支持 4352 点 I/O，程序容量 256k steps。
	AHCPU531-RS2	升级型 CPU 模块，内建 RS-485 * 2、USB 通讯端口以及 SD 卡接口，支持 4352 点 I/O，程序容量 384k steps。
	AHCPU531-EN	升级型 CPU 模块，内建以太网、RS-485、USB 通讯端口以及 SD 卡接口，支持 4352 点 I/O，程序容量 384k steps。
主背板	AHBP04M1-5A	CPU/RTU 专用 4 槽主背板
	AHBP06M1-5A	CPU/RTU 专用 6 槽主背板
	AHBP08M1-5A	CPU/RTU 专用 8 槽主背板
	AHBP12M1-5A	CPU/RTU 专用 12 槽主背板
扩展背板	AHBP06E1-5A	CPU/RTU 专用 6 槽扩展背板
	AHBP08E1-5A	CPU/RTU 专用 8 槽扩展背板
数位 I/O 模块	AH16AM10N-5A	24VDC，5mA，16 点输入，端子台
	AH32AM10N-5A	24VDC，5mA，32 点输入，端子台
	AH32AM10N-5B	24VDC，5mA，32 点输入，DB37 连接器
	AH32AM10N-5C	24VDC，5mA，32 点输入，牛角座连接器
	AH64AM10N-5C	24VDC，3.2mA，64 点输入，牛角座连接器
	AH16AM30N-5A	100 ~ 240VAC，4.5mA ~ 9mA (100V，50Hz)，16 点输入，端子台
数位 I/O 模块	AH16AR10N-5A	24VDC，5mA，16 点输入，端子台 (具备快速中断功能)
	AH16AN01R-5A	240VAC/24VDC，2A，16 输出点，继电器，端子台
	AH16AN01T-5A	12 ~ 24VDC，0.5A，16 输出点，漏型输出，端子台
	AH16AN01P-5A	12 ~ 24VDC，0.5A，16 输出点，源型输出，端子台
	AH32AN02T-5A	12 ~ 24VDC，0.1A，32 输出点，漏型输出，端子台
	AH32AN02T-5B	12 ~ 24VDC，0.1A，32 输出点，漏型输出，DB37 连接器
	AH32AN02T-5C	12 ~ 24VDC，0.1A，32 输出点，漏型输出，牛角座连接器
	AH32AN02P-5A	12 ~ 24VDC，0.1A，32 输出点，源型输出，端子台
	AH32AN02P-5B	12 ~ 24VDC，0.1A，32 输出点，源型输出，DB37 连接器
	AH32AN02P-5C	12 ~ 24VDC，0.1A，32 输出点，源型输出，牛角座连

分类	机种名称	说明
		接器
	AH64AN02T-5C	12 ~ 24VDC · 0.1A · 64 点输出 · 漏型输出 · 牛角座连接器
	AH64AN02P-5C	12 ~ 24VDC · 0.1A · 64 点输出 · 源型输出 · 牛角座连接器
	AH16AN01S-5A	100 ~ 240VAC · 0.5A · 16 输出点 · TRIAC · 端子台
	AH16AP11R-5A	24VDC · 5mA · 8 输入点 · 240VAC/24VDC · 2A · 8 输出点 · 继电器 · 端子台
	AH16AP11T-5A	24VDC · 5mA · 8 输入点 · 12 ~ 24VDC · 0.5A · 8 输出点 · 漏型输出 · 端子台
	AH16AP11P-5A	24VDC · 5mA · 8 输入点 · 12 ~ 24VDC · 0.5A · 8 输出点 · 源型输出 · 端子台
模拟 I/O 模块	AH04AD-5A	4 通道模拟信号输入 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA · -20mA~20mA 转换时间：150us/通道
	AH08AD-5A	8 通道模拟信号输入 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA · -20mA~20mA 转换时间：150us/通道
	AH08AD-5B	8 通道模拟信号输入 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 转换时间：150us/通道
	AH08AD-5C	8 通道模拟信号输入 硬件分辨率：16 位 0/4mA~20mA · -20mA~20mA 转换时间：150us/通道
模拟 I/O 模块	AH04DA-5A	4 通道模拟信号输出 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA

1

分类	机种名称	说明
		转换时间：150us/通道
	AH08DA-5A	8 通道模拟信号输出 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA 转换时间：150us/通道
	AH08DA-5B	8 通道模拟信号输出 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 转换时间：150us/通道
	AH08DA-5C	8 通道模拟信号输出 硬件分辨率：16 位 0/4mA~20mA 转换时间：150us/通道
	AH06XA-5A	4 通道模拟信号输入 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA · -20mA~20mA 转换时间：150us/通道 2 通道模拟信号输出 硬件分辨率：16 位 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA 转换时间：150us/通道
温度模块	AH04PT-5A	4 通道 4 线式或 3 线式 RTD 温度感测 传感器型式 :Pt100 ·Pt1000 ·Ni100 ·Ni1000 ·或 0~300Ω 分辨率：0.1°C/0.1°F (16 位) 转换时间：4 线式转换时间:150ms/通道，3 线式转换时间：300ms/通道。
	AH08PTG-5A	8 通道 4 线式/3 线式/2 线式 RTD 温度感测 传感器形式：Pt100、Pt1000、Ni100、Ni1000，或 0~300Ω，分辨率：0.1°C /0.1°F (16 位) 转换时间：20ms/4 通道，200ms/8 通道。

分类	机种名称	说明
	AH04TC-5A	4 通道热电耦温度感测 传感器型：J · K · R · S · T · E · N 或 -150~+150mV 分辨率：0.1°C/0.1°F 转换时间：200ms/通道
温度模块	AH08TC-5A	8 通道热电耦温度感测 传感器型：J · K · R · S · T · E · N 或 -150~+150mV 分辨率：0.1°C/0.1°F 转换时间：200ms/通道
运动控制模块	AH02HC-5A	2 通道高速计数器模块 (200kHz)
	AH04HC-5A	4 通道高速计数器模块 (200kHz)
	AH05PM-5A	2 轴脉冲型运动控制模块 (1MHz)
	AH10PM-5A	6 轴脉冲型运动控制模块 (4 轴 1MHz / 2 轴 200kHz)
	AH15PM-5A	4 轴脉冲型运动控制模块 (1MHz)
	AH20MC-5A	12 轴 DMCNET (Delta Motion Control Network) 运动控制模块 (10Mbps)
网络模块	AH10EN-5A	以太网通讯模块，可以做主站或者从站。 内建两个以太网接口，支持 MODBUS TCP 主站 支持 EtherNet/IP (V2.0)
	AH15EN-5A	以太网通讯模块，可以做主站或者从站。 内建两个以太网接口，支持 MODBUS TCP 主站。 支援 IEC60870-5-104。
	AH10SCM-5A	串行通讯模块，内建两组 RS-485/422 接口，电源及通讯全隔离，支持 MODBUS 与 UD Link 协议。
	AH15SCM-5A	串行通讯模块，内建两组 RS-232 接口，电源及通讯全隔离，支持 MODBUS 与 UD Link 协议
	AH10DNET-5A	DeviceNet 通讯模块，可以做主站或者从站，通讯速率最大可达 1Mbps。
	AH10PFBM-5A	PROFIBUS 主站通讯模块
	AH10PFBS-5A	PROFIBUS 从站通讯模块
	AH10COPM-5A	CANopen 通讯模块，可以做主站或者从站。
远程 I/O 通讯模块	AHRTU-DNET-5A	DeviceNet 远程 I/O 通讯模块
	AHRTU-PFBS-5A	PROFIBUS 远程 I/O 通讯模块
	AHRTU-ETHN-5A	Ethernet 远程 I/O 通讯模块
扩展通讯线	AHACAB06-5A	扩展背板专用 0.6m 扩展通讯线

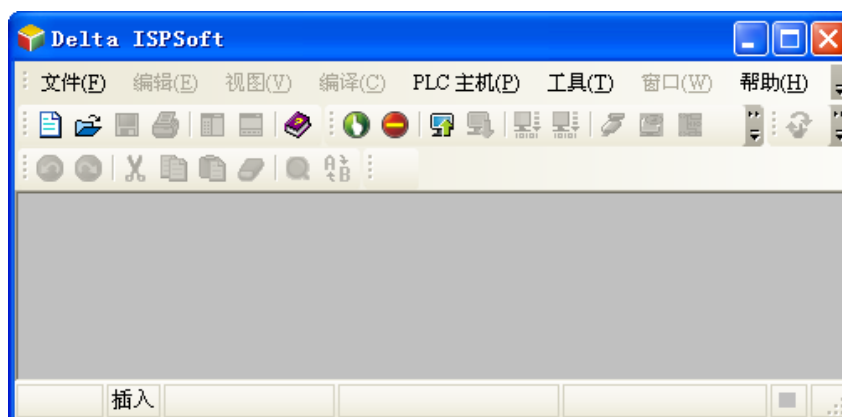
1

分类	机种名称	说明
	AHACAB10-5A	扩展背板专用 1.0m 扩展通讯线
	AHACAB15-5A	扩展背板专用 1.5m 扩展通讯线
	AHACAB30-5A	扩展背板专用 3.0m 扩展通讯线
	AHAADP01/02EF-5A	扩展背板专用光纤模块
I/O 连接线	UC-ET010-24A	1.0m I/O 连接线 (牛角座) · 适用 AH32AM10N-5C 及 AH64AM10N-5C。
	UC-ET010-24C	1.0m I/O 连接线 (牛角座) · 适用 AH32AN02T-5C、AH32AN02P-5C、AH64AN02T-5C 及 AH64AN02P-5C。
	UC-ET010-33B	1.0m I/O 连接线 (DB37) · 适用 AH32AM10N-5B、AH32AN02T-5B 及 AH32AN02P-5B。
	UC-ET010-13B	1.0m I/O 连接线 · 适用 AH04HC-5A 及 AH20MC-5A。
	UC-ET010-15B	1.0m I/O 连接线 · 适用 AH10PM-5A 及 AH15PM-5A。
配线模块	UB-10-ID32A	I/O 配线模块 (32 点输入) · 适用 AH32AM10N-5C 及 AH64AM10N-5C。
	UB-10-ID32B	I/O 配线模块 (16 点继电器输出) · 适用 AH32AN02T-5C 及 AH64AN02T-5C。
	UB-10-OR16A	I/O 配线模块 (16 点继电器输出) · 适用 AH32AN02P-5C 及 AH64AN02P-5C。
	UB-10-OR16B	I/O 配线模块 (32 点输入) · 适用 AH32AM10N-5B。
	UB-10-OR32A	I/O 配线模块 (32 点继电器输出) · 适用 AH32AN02T-5B。
	UB-10-OR32B	I/O 配线模块 (32 点继电器输出) · 适用 AH32AN02P-5B。
	UB-10-OT32A	I/O 配线模块 (32 点晶体管输出) · 适用 AH32AN02T-5C、AH32AN02P-5C、AH64AN02T-5C 及 AH64AN02P-5C。
	UB-10-OT32B	I/O 配线模块 (32 点晶体管输出) · 适用 AH32AN02T-5B 及 AH32AN02P-5B。
	UB-10-IO16C	I/O 配线模块 · 适用 AH04HC-5A 及 AH20MC-5A。
	UB-10-IO24C	I/O 配线模块 · 适用 AH10PM-5A。
	UB-10-IO34C	I/O 配线模块 · 适用 AH15PM-5A。
防护模块	AHASP01-5A	未使用插槽的专用防护模块

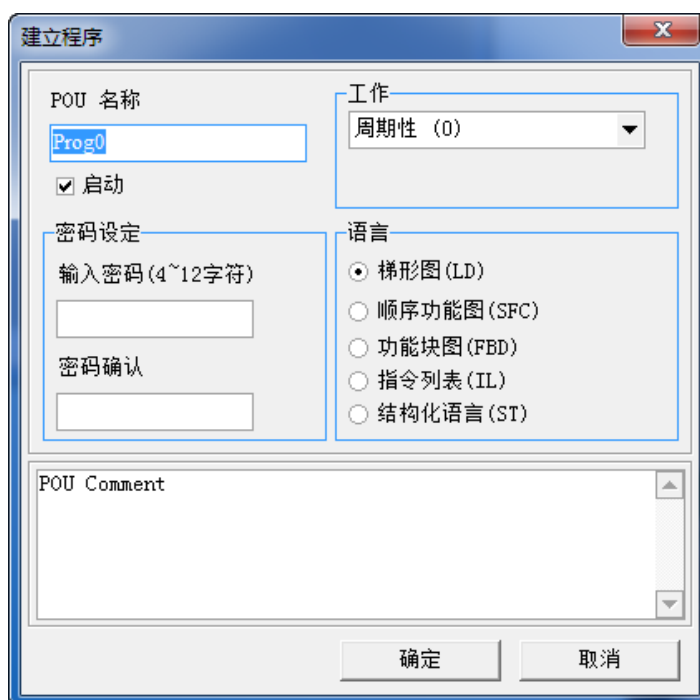
1.2 软件简述

1.2.1 编辑软件

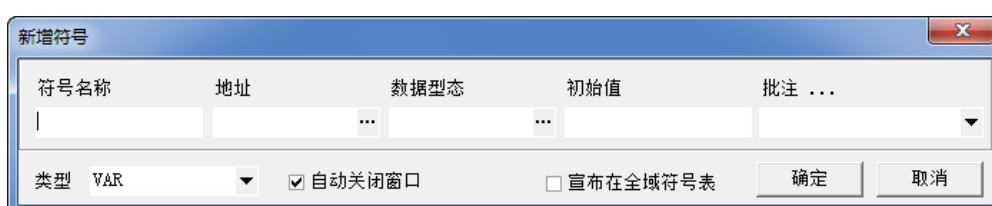
编辑软件 ISPSOft 的概廓：



- 5 种编程语言：包括指令列表 (IL)、结构化语言 (ST)、梯形图 (LD)、顺序功能图 (SFC) 及功能区块图 (FBD)。



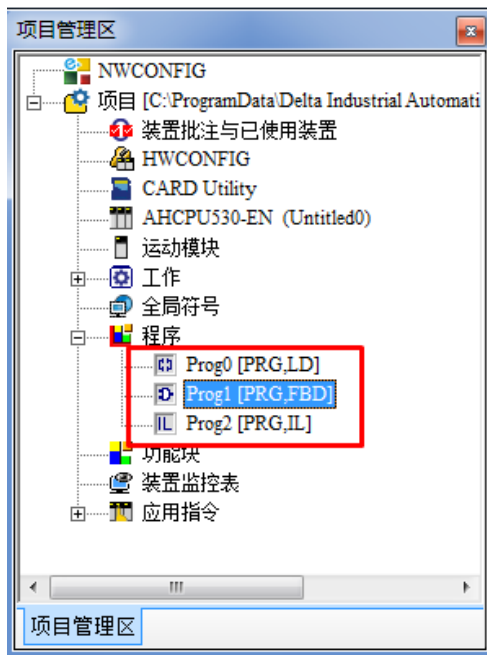
- 使用变量的观念，让用户可自行定义变量符号来取代 PLC 的装置名称，除了大大地提高程序的可读性之外，更省去了用户费心配置装置地址的时间。



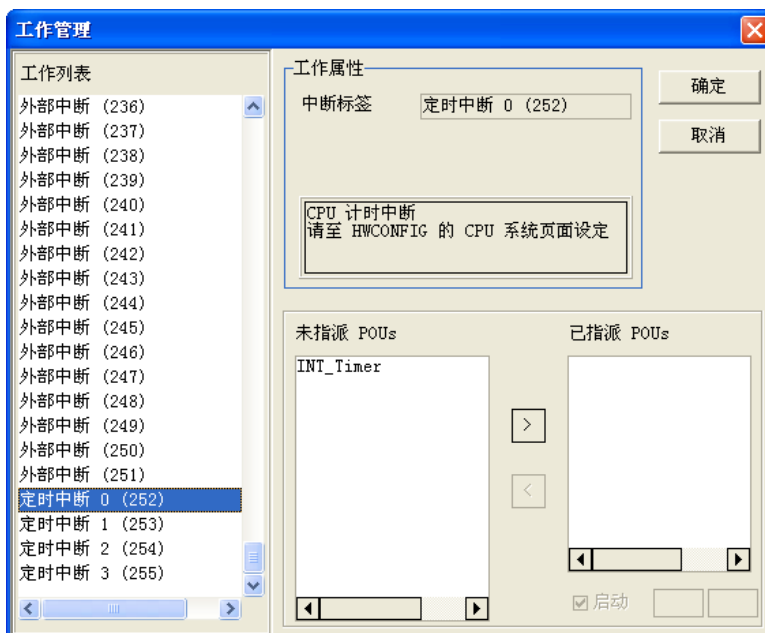
1

全局符号						
类型	符号名称	地址	数据类型	初始值	符号批注	
VAR	停止	M1	BOOL	N/A		
VAR	启动	M0	BOOL	N/A		
VAR	急停	M2	BOOL	N/A		

- 导入程序组织单元 (POU) 的架构，除了可将原本的主程序切割成若干个程序单元之外，并以函数 (Function) 与功能块 (Function Block) 来取代传统的子程序，让程序的架构可以更模块化也更便于管理。



- 使用工作 (Task) 的观念来管理程序的执行顺序，将程序开发的工作提升至项目管理的层次，让大规模的程序开发变的更便于管理。



1.2.2 程序组织单元 (POU) 与工作 (TASK)

程序组织单元 (Program Organization Unit, 简称 POU) 是建构 PLC 程序的基本元素, 有别于传统的 PLC 程序, 由 IEC61131-3 所导入的程序架构, 其特色在于将原本一大段的程序分割成若干个小单元, 而这些小单元便称之为 POU。POU 大致可分为以下三种。

1. 程序 (PROG): 程序 POU 于 PLC 程序中扮演着主要程序的角色, 其执行方式可由设计者自行定义为周期扫描或中断模式, 并可在工作 (TASK) 清单中安排各个程序 POU 的扫描顺序。
2. 功能块 (FB): 功能块 (Function Block, 简称 FB) 本身所代表的意义类似于子程序, 而在功能块内部所定义的程序, 则必须要在该功能块被程序 POU 所呼叫, 并输入相关的参数之后才会执行。
3. 函数 (FC): 函数 (Function, 简称 FC) 本身所代表的意义较接近于宏指令, 也就是我们可将多笔的运算指令或功能写在函数 POU 当中, 之后再于程序 POU 或功能块 POU 当中呼叫使用即可。

工作 (TASK), 是指规定使各个程序按照何种顺序或中断条件执行的功能。工作 (TASK) 的意义在于赋予各个程序 POU 一个明确的执行任务, 并指定每个程序 POU 之间的执行顺序或是启动的方式。基本上一个项目中的程序 POU 并非都会参与实际的执行, 而是必须在指派 TASK 之后才可确定该 POU 是否执行以及如何执行; 当 POU 未被指派 TASK 时, 该 POU 仅会被当做一般的原始码而与项目一起储存, 本身并不会被编译为 PLC 的执行码。此外, 仅有程序 POU 需要被指派 TASK, 至于功能块 (FB) 或函数 (FC) 的执行方式, 则是依据呼叫它们的上位 POU 程序而定。TASK 的运行方式大致可分为周期性、定时中断, 以及条件中断 3 种

1. 周期性: 被指派至周期性 TASK 的程序 POU, 其执行方式便是单纯的来回扫描, 按照顺序执行。
2. 定时中断工作 (TASK): 被指派至定时中断工作的程序 POU, 当定时中断工作的时间到达后, 被分配至该定时中断工作的所有程序 POU 便会依照排列顺序执行。
3. 条件中断工作 (TASK): 条件中断亦可区分为多种类型, 如外部中断、I/O 中断...等, 规划前必须先确认所选用的项目机种支持哪些中断模式。当程序 POU 被分配至条件中断 TASK 时, 其功能便类似中断子程序, 当中断条件成立时, 如外部触发中断的某个接点被触发, 被分配至该 TASK 的所有 POU 便会依照排列顺序执行一遍。

MEMO



2

第2章 装置说明

目录

2.1	装置简介	2-2
2.1.1	装置列表	2-2
2.1.1.1	AH500 基本型 CPU 模块 (AHCPU500/510/520/530)	2-2
2.1.1.2	AH500 升级型 CPU 模块 (AHCPU501/511/521/531)	2-4
2.1.2	I/O 储存区的基本结构	2-5
2.1.3	停电保持储存方式	2-6
2.1.4	停电保持区的装置范围	2-6
2.2	装置功能说明	2-7
2.2.1	数值、常数	2-7
2.2.2	浮点数	2-10
2.2.2.1	单精度浮点数 (32 位浮点数)	2-10
2.2.2.2	双精度浮点数 (64 位浮点数)	2-10
2.2.2.3	十进浮点数	2-12
2.2.3	字符串	2-12
2.2.4	输入继电器 X	2-14
2.2.5	输出继电器 Y	2-14
2.2.6	辅助继电器 M	2-15
2.2.7	特殊辅助继电器 SM	2-15
2.2.8	特殊辅助继电器 SM 的更新时间	2-47
2.2.9	步进点继电器 S	2-55
2.2.10	定时器 T	2-55
2.2.11	计数器 C	2-57
2.2.12	32 位计数器 HC	2-58
2.2.13	数据寄存器 D	2-60
2.2.14	特殊数据寄存器 SR	2-60
2.2.15	特殊数据寄存器 SR 的更新时间	2-96
2.2.16	SM/SR 补充说明	2-98
2.2.17	连结寄存器 L	2-114
2.2.18	变址寄存器 E	2-114

2.1 装置简介

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等各种装置的配置和功能做说明。

2

2.1.1 装置列表

2.1.1.1 AH500 基本型 CPU 模块 (AHCPU500/510/520/530)

形式	装置名称		装置数	范围
位装置	输入继电器	X	1024 (AHCPU500)	X0.0~X63.15
			2048 (AHCPU510)	X0.0~X127.15
			4096 (AHCPU520)	X0.0~X255.15
			8192 (AHCPU530)	X0.0~X511.15
	输出继电器	Y	1024 (AHCPU500)	Y0.0~X63.15
			2048 (AHCPU510)	Y0.0~X127.15
			4096 (AHCPU520)	Y0.0~X255.15
			8192 (AHCPU530)	Y0.0~Y511.15
	数据寄存器	D	262144 (AHCPU500)	D0.0~D16383.15
			524288 (AHCPU510)	D0.0~D32767.15
			1048576 (AHCPU520/530)	D0.0~D65535.15
	连结寄存器	L	262144 (AHCPU500)	L0.0~L16383.15
			524288 (AHCPU510)	L0.0~L32767.15
			1048576 (AHCPU520/530)	L0.0~L65535.15
	辅助继电器	M	8192	M0~M8191
	特殊辅助标志	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	
32 位计数器	HC	64	HC0~HC63	
字符装置	输入继电器	X	64 (AHCPU500)	X0~X63
			128 (AHCPU510)	X0~X127
			256 (AHCPU520)	X0~X255
			512 (AHCPU530)	X0~X511
	输出继电器	Y	64 (AHCPU500)	Y0~Y63

形式	装置名称	装置数	范围	
		128 (AHCPU510)	Y0~Y127	
		256 (AHCPU520)	Y0~Y255	
		512 (AHCPU530)	Y0~Y511	
	数据寄存器	D	16384 (AHCPU500)	D0~D16383
			32768 (AHCPU510)	D0~D32767
			65536 (AHCPU520/530)	D0~D65535
	特殊数据寄存器	SR	2048	SR0~SR2047
	连结寄存器	L	16384 (AHCPU500)	L0~L16383
			32768 (AHCPU510)	L0~L32767
			65536 (AHCPU520/530)	L0~L65535
定时器	T	2048	T0~T2047	
计数器	C	2048	C0~C2047	
32 位计数器	HC	64 (128 words)	HC0~HC63	
变址寄存器	E	32	E0~E31	
常数*	十进制	K	16 位运算：-32768~32767 32 位运算：-2147483648~2147483647	
常数*	十六进制	16#	16 位运算：16#0~16#FFFF 32 位运算：16#0~16#FFFFFFFF	
	单精度浮点数	F	32 位运算： $\pm 1.17549435^{-38} \sim \pm 3.40282347^{+38}$	
	双精度浮点数	DF	64 位运算： $\pm 2.2250738585072014^{-308} \sim \pm 1.7976931348623157^{+308}$	
字符串*	字符串	“\$”	1~31 个字	

*1：十进制的表示方式，在第5、6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5、6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5、6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”。

2.1.1.2 AH500 升级型 CPU 模块 (AHCPU501/511/521/531)

2

形式	装置名称		装置数	范围
位装置	输入继电器	X	2048 (AHCPU501)	X0.0~X127.15
			4096 (AHCPU511)	X0.0~X255.15
			8192 (AHCPU521)	X0.0~X511.15
			16384 (AHCPU531)	X0.0~X1023.15
	输出继电器	Y	2048 (AHCPU501)	Y0.0~Y127.15
			4096 (AHCPU511)	Y0.0~Y255.15
			8192 (AHCPU521)	Y0.0~Y511.15
			16384 (AHCPU531)	Y0.0~Y1023.15
	数据寄存器	D	393216 (AHCPU501)	D0.0~D24575.15
			786432 (AHCPU511)	D0.0~D49151.15
			1572864 (AHCPU521)	D0.0~D98303.15
			2097152 (AHCPU531)	D0.0~D131071.15
	链接寄存器	L	393216 (AHCPU501)	L0.0~L24575.15
			786432 (AHCPU511)	L0.0~L49151.15
			1572864 (AHCPU521)	L0.0~L98303.15
			2097152 (AHCPU531)	L0.0~L131071.15
	辅助继电器	M	8192	M0~M8191
	特殊辅助标志	SM	4096	SM0~SM4095
	步进点继电器	S	2048	S0~S2047
	定时器	T	2048	T0~T2047
计数器	C	2048	C0~C2047	
32 位计数器	HC	64	HC0~HC63	
字符装置	输入继电器	X	128 (AHCPU501)	X0~X127
			256 (AHCPU511)	X0~X255
			512 (AHCPU521)	X0~X511
			1024 (AHCPU531)	X0~X1023
	输出继电器	Y	128 (AHCPU501)	Y0~Y127
			256 (AHCPU511)	Y0~Y255
			512 (AHCPU521)	Y0~Y511
			1024 (AHCPU531)	Y0~Y1023
数据寄存器	D	24576 (AHCPU501)	D0~D24575	

形式	装置名称		装置数	范围
			49152 (AHCPU511)	D0~D49151
			98304 (AHCPU521)	D0~D98303
			131072 (AHCPU531)	D0~D131071
	特殊数据寄存器	SR	4096 (AHCPU511-RS2)	SR0~SR4095
	链接寄存器	L	24576 (AHCPU501)	L0~L24575
			49152 (AHCPU511)	L0~L49151
			98304 (AHCPU521)	L0~L98303
			131072 (AHCPU531)	L0~L131071
	定时器	T	2048	T0~T2047
	计数器	C	2048	C0~C2047
32 位计数器	HC	64 (128 words)	HC0~HC63	
变址寄存器	E	32	E0~E31	
常数*	十进制	K	16 位运算：-32768~32767	
			32 位运算：-2147483648~2147483647	
常数*	十六进制	16#	16 位运算：16#0~16#FFFF	
			32 位运算：16#0~16#FFFFFFFF	
	单精度浮点数	F	32 位运算：±1.17549435 ⁻³⁸ ~±3.40282347 ⁺³⁸	
双精度浮点数	DF	64 位运算：±2.2250738585072014 ⁻³⁰⁸ ~±1.7976931348623157 ⁺³⁰⁸		
字符串*	字符串	“\$”	1~31 个字	

*1：十进制的表示方式，在第5、6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5、6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5、6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”。

2.1.2 I/O 储存区的基本结构

装置	功能	位存取	字符存取	ISPSoft 修改	强制位状态
X	输入继电器	OK	OK	OK	OK
Y	输出继电器	OK	OK	OK	OK
M	辅助继电器	OK	-	OK	-
SM	特殊辅助标志	OK	-	OK	-
S	步进点继电器	OK	-	OK	-
T	定时器	OK	OK	OK	-

2

C	计数器	OK	OK	OK	-
HC	32 位计数器	OK	OK	OK	-
D	数据寄存器	OK	OK	OK	-
SR	特殊数据寄存器	-	OK	OK	-
L	连结寄存器	OK	OK	OK	-
E	变址寄存器	-	OK	OK	-

2.1.3 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 装置恢复 STOP 前状态	保持	保持	恢复 STOP 前状态
STOP=>RUN	设定非停电保持区清除	清除	保持	参照 Y 装置之设定
	设定非停电保持区保持	保持	保持	参照 Y 装置之设定
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 停电保持区的装置范围

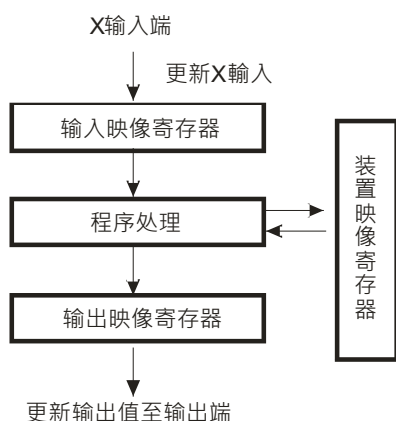
装置	功能	停电保持区范围
X	输入继电器	固定非停电保持
Y	输出继电器	固定非停电保持
M*	辅助继电器	默认 M0~M8191
SM	特殊辅助继电器	部分停电保持并且不能被改变 详细内容请参考 SM 菜单
S	步进点继电器	固定非停电保持
T*	定时器	默认 T0~T2047
C*	计数器	默认 C0~C2047
HC*	32 位计数器	默认 HC0~HC63

D*	数据寄存器	AH500-EN/RS2：默认 D0~D16383 AH501-EN/RS2：默认 D0~D24575
		其它机种默认 D0~D32767 最多可以设定 32768 个
SR	特殊数据寄存器	部分停电保持并且不能被改变 详细内容请参考 SR 菜单
L	连结寄存器	固定非停电保持
E	变址寄存器	固定非停电保持

*：表可设定停电保持区的范围，可以设定此装置不停电保持，而设定之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。

2.2 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映像寄存器内。
 2. 在程序执行中若输入信号作 ON/OFF 变化，但是输入信号映像寄存器内的状态不会改变，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映像寄存器。
- 更新输出状态：

当执行到 END 指令后将装置映像寄存器内的状态送到用户所分配的输出端。

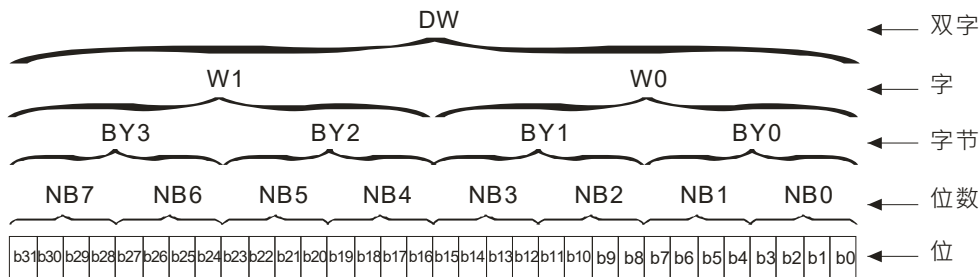
2.2.1 数值、常数

名称	说明
位 (Bit)	位为二进制数值之最基本单位，其状态非 1 即 0
位数 (Nibble)	由连续的 4 个位所组成 (如 b3 ~ b0) 可用以表示一个位数之十进制数字 0 ~ 9 或十六进制之 0 ~ F
字节 (Byte)	是由连续之两个位数所组成 (亦即 8 位，b7 ~ b0)。可表示十六进制之 00 ~ FF
字 (Word)	是由连续之两个字节所组成 (亦即 16 位，b15 ~ b0) 可表示十六进制之 4 个位数值 0000 ~ FFFF

2

名称	说明
双字 (Double Word)	是由连续之两个字所组成 (亦即 32 位 · b31 ~ b0) · 可表示十六进制之 8 个位数值 00000000~FFFFFFFF
四字	是由连续之四个字所组成 (亦即 64 位 · b63 ~ b0) · 可表示十六进制之 16 个位数值 0000000000000000 - FFFFFFFFFFFFFFFF

二进制系统中位、位数、字节、字、及双字的关系如下图所示：



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 (Binary Number · BIN)

PLC 内部之数值运算或储存均采用二进制

2. 十进制 (Decimal Number · 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)
- Constant 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：

- Constant 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)

数值对照表：

二进制 (BIN)	十进制 (DEC)	BCD (Binary Code Decimal)	十六进制 (HEX)
PLC 内部运算用	常数 K · 装置编号	BCD 相关指令	常数 16# · 装置编号
0000	0	0000	0
0001	1	0001	1
0010	2	0010	2
0011	3	0011	3
0100	4	0100	4
0101	5	0101	5
0110	6	0110	6

二进制 (BIN)	十进制 (DEC)	BCD (Binary Code Decimal)	十六进制 (HEX)
0111	7	0111	7
1000	8	1000	8
1001	9	1001	9
1010	10	-	A
1011	11	-	B
1100	12	-	C
1101	13	-	D
1110	14	-	E
1111	15	-	F
10000	16	0001 0000	10
10001	17	0001 0001	11

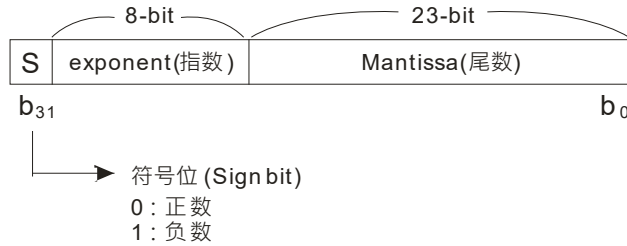
2

2.2.2 浮点数

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。

2.2.2.1 单精度浮点数 (32 位浮点数)

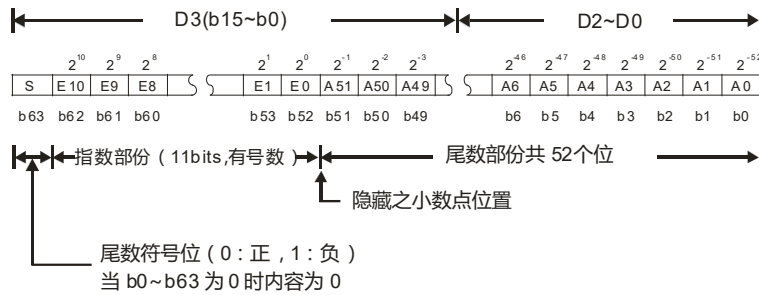
以 32 位的寄存器长度表示浮点数，而表示法系采用 IEEE754 的标准，格式如下：



表达式： $(-1)^S \times 2^{E-B} \times 1.M; B=127$

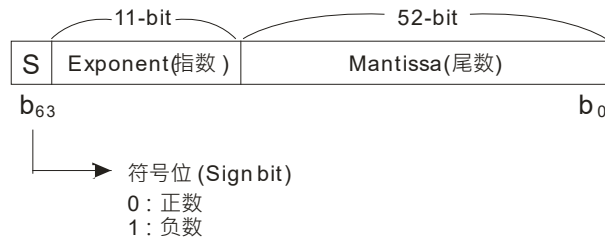
因此单精度浮点数的数目范围为 $\pm 2^{-126}$ 到 $\pm 2^{+128}$ 相当于 $\pm 1.1755 \times 10^{-38}$ 到 $\pm 3.4028 \times 10^{+38}$ 。

AH500 使用 2 个连续号码的寄存器组成 32 位的浮点数，我们以寄存器 (D1, D0) 来说明，如下所示：



2.2.2.2 双精度浮点数 (64 位浮点数)

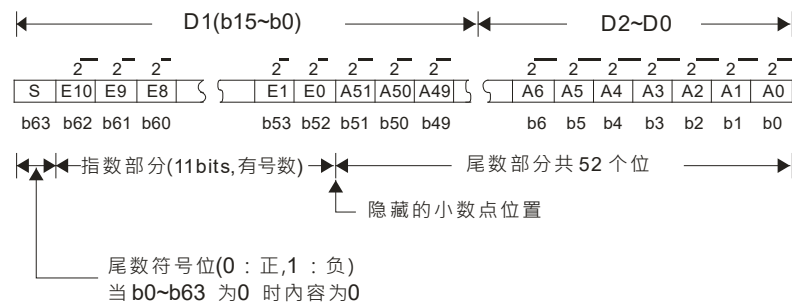
以 64 位的寄存器长度表示浮点数，而表示法系采用 IEEE754 的标准，格式如下：



表达式： $(-1)^S \times 2^{E-B} \times 1.M; B = 1023$

因此双精度浮点数的数目范围为 $\pm 2^{-1022}$ 到 $\pm 2^{+1024}$ 相当于 $\pm 2.2250 \times 10^{-308}$ 到 $\pm 1.7976 \times 10^{+308}$ 。

AH500 使用 4 个连续号码的寄存器组成 64 位浮点数，我们以寄存器 (D3 · D2 · D1 · D0) 来说明，如下所示：



范例一：

以单精度浮点数表示 23

步骤一：将 23 转换成二进制数字：23.0=10111

步骤二：将二进制数字正规化：10111=1.0111 × 2⁴，其中 0111 为尾数，4 为指数。

步骤三：求出指数部份的储存值

$$\because E-B=4 \rightarrow E-127=4 \therefore E=131=10000011_2$$

步骤四：组合符号位·指数·尾数成为浮点数。

$$0\ 10000011\ 011100000000000000000000_2=41B80000_{16}$$

以双精度浮点数表示 23

步骤一：将 23 转换成二进制数字：23.0=10111

步骤二：将二进制数字正规化：10111=1.0111 × 2⁴，其中 0111 为尾数，4 为指数。

步骤三：求出指数部份的储存值

$$\because E-B=4 \rightarrow E-1023=4 \therefore E=1027=10000000011_2$$

步骤四：组合符号位·指数·尾数成为浮点数。

$$0\ 10000000011\ 011100_2 \\ =4037000000000000_{16}$$

范例二：

以单精度浮点数表示-23.0

-23.0 浮点格式与 23.0 的转换步骤完全相同，只需将符号位改为 1 即可。

$$1\ 10000011\ 011100000000000000000000_2=C1B80000_{16}$$

以双精度浮点数表示-23.0

-23.0 浮点格式与 23.0 的转换步骤完全相同，只需将符号位改为 1 即可。

$$1\ 10000000011\ 011100_2 \\ =C037000000000000_{16}$$

2.2.2.3 十进浮点数

- ◆ 单精度浮点数跟双精度浮点数的内容比较无法被人所接受，因此，单精度浮点数跟双精度浮点数可转换成十进浮点数来供人作判断。但是 PLC 对小数点的运算仍旧是使用单精度浮点数跟双精度浮点数。
- ◆ 32 位十进浮点数是使用 2 个连续号码的寄存器来表现，较小编号的寄存器号码存放常数部份、较大编号的寄存器号码存放指数部份。

就以寄存器 (D1、D0) 来存放一个十进浮点数为例，如下所示。

$$\text{十进浮点数} = [\text{常数D0}] * 10^{[\text{指数D1}]}$$

底数 D0 = ±1.000~±9.999

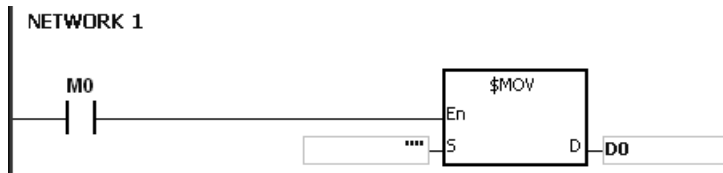
指数 D1 = -41~+35

此外，底数 100 不存在于 D0 的内容，因为，100 是以 1.000×10^{-1} 来表示。32 位十进浮点数的范围为 $\pm 1175 \times 10^{-41}$ 到 $\pm 402 \times 10^{+35}$ 。

2.2.3 字符串

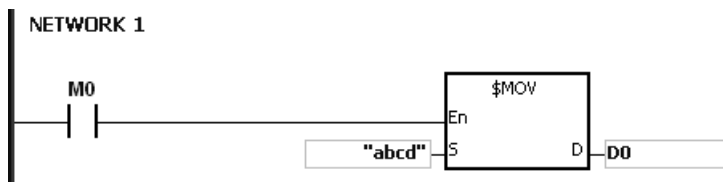
字符串可以处理的是 ASCII 编码的字 (*1)，一个完整的字符串定义为字符串的起始字符到结束字符 NULL 码 (16#00) 为止。若用户直接输入的是字符串，则最多可以输入 31 个字且 ISPSOft 会自动补上结束字符 16#00。若用户输入的是寄存器，则需在结束的后面补上一个结束字符 16#00。

1. 字符串搬移 NULL



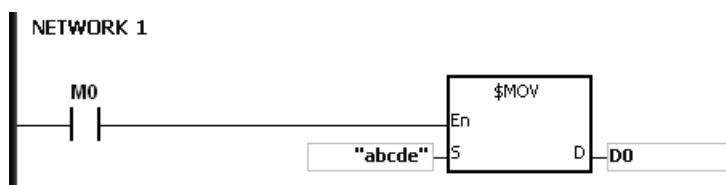
D0=0 (NULL)。

2. 字符串为偶数的时候：



D0	16#62 (b)	16#61 (a)
D1	16#64 (d)	16#63 (c)
D2	0 (NULL)	

3. 字符串为奇数的时候：



D0	16#62 (b)	16#61 (a)
D1	16#64 (d)	16#63 (c)
D2	0 (NULL)	16#65 (e)

*1：ASCII 码转换表

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Hex	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
ASCII	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Hex	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
ASCII	SP	!	"	#	\$	%	&	'	()	*	+	.	-	.	/
Hex	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
ASCII	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
Hex	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
ASCII	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Hex	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
ASCII	P	Q	R	S	T	U	V	W	X	Y	Z	☒	☒	☒	☒	☒
Hex	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
ASCII	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
Hex	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
ASCII	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒

注：标示为☒，均为不可视字符，请勿设定。

2.2.4 输入继电器 X

- 输入接点 X 的功能：
输入接点 X 与输入装置（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 接点于程序中使用次数没有限制。输入接点 X 的 ON/OFF 只会跟随输入装置的 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，编号的大小跟随 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）

2.2.5 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 接点于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，其输出状态的决心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，编号的大小跟随 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）

2.2.6 辅助继电器 M

辅助继电器M有A、B接点，而且于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：一般用辅助继电器于PLC运转时若遇到停电，其状态将全部被复归为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于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。

2.2.7 特殊辅助继电器 SM

每一个特殊辅助继电器均有其特定的用途，未定义的特殊标志请勿使用。

特殊辅助继电器 (SM)，它的种类及功能如下所示。在编号前有“*”记号可参考SM/SR补充说明，像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之说明，可进一步了解系统信息。

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
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
*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
*SM20	电源供应异常纪录	○	○	○	○	○	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	○	X	○	X	X	OFF	OFF	-	R/W	OFF
*SM98	COM1 接收等待	○	○	○	○	○	OFF	OFF	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM99	COM2 接收等待	0	X	0	X	X	OFF	OFF	-	R	OFF
*SM100	COM1 接收完毕	0	0	0	0	0	OFF	OFF	-	R/W	OFF
*SM101	COM2 接收完毕	0	X	0	X	X	OFF	OFF	-	R/W	OFF
*SM102	COM1 MODRW 或 RS 数据接收错误	0	0	0	0	0	OFF	OFF	-	R	OFF
*SM103	COM2 MODRW 或 RS 数据接收错误	0	X	0	X	X	OFF	OFF	-	R	OFF
*SM104	COM1 接收超时	0	0	0	0	0	OFF	OFF	-	R/W	OFF
*SM105	COM2 接收超时	0	X	0	X	X	OFF	OFF	-	R/W	OFF
*SM106	COM1 8/16 位处理模式选择 · ON : 8 位 · OFF : 16 位	0	0	0	0	0	OFF	-	-	R/W	OFF
*SM107	COM2 8/16 位处理模式选择 · ON : 8 位 · OFF : 16 位	0	X	0	X	X	OFF	-	-	R/W	OFF
SM108	COM1 接收结束标志	V1.0 1	V1.01	0	0	0	-	-	-	R/W	OFF
SM109	COM2 接收结束标志	V1.0 1	X	0	X	X	-	-	-	R/W	OFF
*SM204	非停电保持区域全部清除	0	0	0	0	0	OFF	-	-	R/W	OFF
*SM205	停电保持区域全部清除	0	0	0	0	0	OFF	-	-	R/W	OFF
SM206	输出全部禁止	0	0	0	0	0	OFF	-	-	R/W	OFF
SM209	COM1 通讯协议变更用	0	0	0	0	0	OFF	-	-	R/W	OFF
SM210	COM1 ASCII/RTU 模式选择 · ON 时为 RTU 模式	0	0	0	0	0	OFF	-	-	R/W	OFF
SM211	COM2 通讯协议变更用	0	X	0	X	X	OFF	-	-	R/W	OFF
SM212	COM2 ASCII/RTU 模式选择 · ON 时为 RTU 模式	0	X	0	X	X	OFF	-	-	R/W	OFF
SM215	PLC 运行状态	0	0	0	0	0	OFF	ON	OFF	R/W	OFF
SM220	万年历±30 秒校正	0	0	0	0	0	OFF	OFF	-	R/W	OFF
*SM400	常开接点	0	0	0	0	0	ON	ON	ON	R	ON
*SM401	常闭接点	0	0	0	0	0	OFF	OFF	OFF	R	OFF
*SM402	启动正向 (RUN 的瞬间“ON”) 脉冲	0	0	0	0	0	OFF	ON	OFF	R	OFF
*SM403	启动反向 (RUN 的瞬间“OFF”) 脉冲	0	0	0	0	0	ON	OFF	ON	R	ON
*SM404	10ms 时钟脉冲 · 5ms ON/5ms OFF	0	0	0	0	0	OFF	-	-	R	OFF

SM	功能说明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SM405	100ms 时钟脉冲 · 50ms ON/50ms OFF	0	0	0	0	0	OFF	-	-	R	OFF
*SM406	200ms 时钟脉冲 · 100ms ON/100ms OFF	0	0	0	0	0	OFF	-	-	R	OFF
*SM407	1s 时钟脉冲 · 0.5s ON/0.5s OFF	0	0	0	0	0	OFF	-	-	R	OFF
*SM408	2s 时钟脉冲 · 1s ON/1s OFF	0	0	0	0	0	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	0	0	0	0	0	OFF	-	-	R	OFF
*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	0	0	0	0	0	OFF	-	-	R	OFF
*SM418	背板 1 (主背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM420	背板 2 (第一组冗余扩展背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM421	背板 2 (第一组冗余扩展背板) Port4 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM422	背板 3 (第二组冗余扩展背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM423	背板 3 (第二组冗余扩展背板) Port4 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM424	背板 4 (第三组冗余扩展背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM425	背板 4 (第三组冗余扩展背板) Port4 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM426	背板 5 (第四组冗余扩展背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM427	背板 5 (第四组冗余扩展背板) Port4 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM428	背板 6 (第五组冗余扩展背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM429	背板 6 (第五组冗余扩展背板) Port4 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM430	背板 7 (第六组冗余扩展背板) Port2 联机状态	X	X	0	0	0	OFF	-	-	R	OFF
*SM431	背板 7 (第六组冗余扩展背板) Port4 联机状态	X	X	0	0	0	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	联机状态										
*SM450	储存卡是否存在标志 ·ON :存在/OFF : 不存在	0	0	0	0	0	OFF	-	-	R	OFF
*SM451	储存卡写保护开关(Protect Switch) · ON : 写保护/OFF : 无写保护	0	0	0	0	0	OFF	-	-	R	OFF
*SM452	储存卡正被存取(Accessed)中 ·ON : 存取中/OFF : 无存取	0	0	0	0	0	OFF	-	-	R	OFF
*SM453	储存卡运行中有错误发生 ·ON-表错 误发生	0	0	0	0	0	OFF	-	-	R	OFF
SM600	零标号 (Zero flag)	0	0	0	0	0	OFF	-	-	R	OFF
SM601	借位标号 (Borrow flag)	0	0	0	0	0	OFF	-	-	R	OFF
SM602	进位标号 (Carry flag)	0	0	0	0	0	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)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM605	SMOV 工作模式指定	0	0	0	0	0	OFF	-	-	R/W	OFF
SM606	8/16 位工作模式	0	0	0	0	0	OFF	-	-	R/W	OFF
SM607	矩阵比较标志 · 比较相同值 (SM607 为 ON) 或不同值 (SM607 为 OFF)	0	0	0	0	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 为 ON	0	0	0	0	0	OFF	-	-	R	OFF
SM609	矩阵搜寻起始标志 · 当 SM609 为 ON 时由第 0 个 bit 开始比较	0	0	0	0	0	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 为 ON	0	0	0	0	0	OFF	-	-	R	OFF
SM611	矩阵指针错误标志 · 指针 Pr 值超出范 围则 SM611 为 ON	0	0	0	0	0	OFF	-	-	R	OFF
SM612	矩阵指针递增标志 · 将指针目前值+1	0	0	0	0	0	OFF	-	-	R/W	OFF
SM613	矩阵指针清除标志 · 将指针目前值清 除为 0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM614	矩阵旋转位移输出进位标志	0	0	0	0	0	OFF	-	-	R	OFF
SM615	矩阵位移输入补位标志	0	0	0	0	0	OFF	-	-	R/W	OFF
SM616	矩阵旋转位移方向标志 SM616 为 OFF : 左移 · SM616 为 ON : 右移	0	0	0	0	0	OFF	-	-	R/W	OFF

SM	功能说明	CPUsX0-RS2	CPUsX0-EN	CPUsX1-RS2	CPUsX1-EN	CPUs60-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM617	矩阵计数字元为 0 或位为 1 标志	0	0	0	0	0	OFF	-	-	R/W	OFF
SM618	矩阵计数结果为 0 时 ON	0	0	0	0	0	OFF	-	-	R/W	OFF
SM619	当 EI 指令被执行时 ON	0	0	0	0	0	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	0	0	0	0	0	OFF	-	-	R	OFF
SM621	HC0 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM622	HC1 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM623	HC2 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM624	HC3 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM625	HC4 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM626	HC5 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM627	HC6 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM628	HC7 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM629	HC8 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM630	HC9 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM631	HC10 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM632	HC11 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM633	HC12 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM634	HC13 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM635	HC14 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM636	HC15 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM637	HC16 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM638	HC17 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM639	HC18 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM640	HC19 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM641	HC20 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM642	HC21 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM643	HC22 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM644	HC23 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM645	HC24 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM646	HC25 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM647	HC26 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM648	HC27 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM649	HC28 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM650	HC29 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM651	HC30 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM652	HC31 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM653	HC32 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM654	HC33 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM655	HC34 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM656	HC35 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM657	HC36 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM658	HC37 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM659	HC38 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM660	HC39 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM661	HC40 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM662	HC41 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM663	HC42 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM664	HC43 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM665	HC44 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM666	HC45 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM667	HC46 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM668	HC47 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM669	HC48 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM670	HC49 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM671	HC50 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM672	HC51 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM673	HC52 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF

SM	功能说明	CP U5X0-RS2	CP U5X0-EN	CP U5X1-RS2	CP U5X1-EN	CP U560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM674	HC53 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM675	HC54 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM676	HC55 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM677	HC56 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM678	HC57 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM679	HC58 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM680	HC59 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM681	HC60 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM682	HC61 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM683	HC62 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM684	HC63 计数模式设定 (ON 时为下数)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM685	DSCLP 使用浮点数运算	0	0	0	0	0	OFF	-	-	R/W	OFF
SM686	RAMP 指令连续执行设定	0	0	0	0	0	OFF	-	-	R/W	OFF
SM687	RAMP 指令执行完毕	0	0	0	0	0	OFF	-	-	R/W	OFF
SM688	INCD 指令执行完毕	0	0	0	0	0	OFF	-	-	R/W	OFF
SM690	字符串控制模式	0	0	0	0	0	OFF	-	-	R/W	OFF
SM691	HKY 输入为 16 位模式 ON : 十六进制输入 · OFF : A~F 为功能键	0	0	0	0	0	OFF	-	-	R/W	OFF
SM692	HKY 指令执行完毕标志 (ON 一个扫描周期)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM693	SEGL 指令执行完毕标志 (ON 一个扫描周期)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM694	DSW 指令执行完毕标志 (ON 一个扫描周期)	0	0	0	0	0	OFF	-	-	R/W	OFF
SM695	弧度/角度使用标志 · ON 的时候表示角度	0	0	0	0	0	OFF	-	-	R/W	OFF
SM699	MODBUS TCP 初始化错误标志	X	V1.01	X	0	0	OFF	-	-	R	OFF
SM700	MODBUS TCP 联机 1 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM701	MODBUS TCP 联机 2 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM702	MODBUS TCP 联机 3 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM703	MODBUS TCP 联机 4 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM704	MODBUS TCP 联机 5 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM705	MODBUS TCP 联机 6 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM706	MODBUS TCP 联机 7 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM707	MODBUS TCP 联机 8 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM708	MODBUS TCP 联机 9 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM709	MODBUS TCP 联机 10 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM710	MODBUS TCP 联机 11 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM711	MODBUS TCP 联机 12 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM712	MODBUS TCP 联机 13 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM713	MODBUS TCP 联机 14 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM714	MODBUS TCP 联机 15 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM715	MODBUS TCP 联机 16 启动标志	X	V1.01	X	0	0	OFF	-	-	R/W	OFF
SM716	MODBUS TCP 联机 17 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM717	MODBUS TCP 联机 18 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM718	MODBUS TCP 联机 19 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM719	MODBUS TCP 联机 20 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM720	MODBUS TCP 联机 21 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM721	MODBUS TCP 联机 22 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM722	MODBUS TCP 联机 23 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM723	MODBUS TCP 联机 24 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM724	MODBUS TCP 联机 25 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM725	MODBUS TCP 联机 26 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM726	MODBUS TCP 联机 27 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM727	MODBUS TCP 联机 28 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM728	MODBUS TCP 联机 29 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM729	MODBUS TCP 联机 30 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM730	MODBUS TCP 联机 31 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF
SM731	MODBUS TCP 联机 32 启动标志	X	0*1	X	0*6	0	OFF	-	-	R/W	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM732	MODBUS TCP 联机 33 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM733	MODBUS TCP 联机 34 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM734	MODBUS TCP 联机 35 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM735	MODBUS TCP 联机 36 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM736	MODBUS TCP 联机 37 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM737	MODBUS TCP 联机 38 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM738	MODBUS TCP 联机 39 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM739	MODBUS TCP 联机 40 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM740	MODBUS TCP 联机 41 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM741	MODBUS TCP 联机 42 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM742	MODBUS TCP 联机 43 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM743	MODBUS TCP 联机 44 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM744	MODBUS TCP 联机 45 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM745	MODBUS TCP 联机 46 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM746	MODBUS TCP 联机 47 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM747	MODBUS TCP 联机 48 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM748	MODBUS TCP 联机 49 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM749	MODBUS TCP 联机 50 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM750	MODBUS TCP 联机 51 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM751	MODBUS TCP 联机 52 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM752	MODBUS TCP 联机 53 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM753	MODBUS TCP 联机 54 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM754	MODBUS TCP 联机 55 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM755	MODBUS TCP 联机 56 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM756	MODBUS TCP 联机 57 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM757	MODBUS TCP 联机 58 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM758	MODBUS TCP 联机 59 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM759	MODBUS TCP 联机 60 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM760	MODBUS TCP 联机 61 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM761	MODBUS TCP 联机 62 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM762	MODBUS TCP 联机 63 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM763	MODBUS TCP 联机 64 启动标志	X	0*2	X	0*4	0	OFF	-	-	R/W	OFF
SM764	MODBUS TCP 联机 65 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM765	MODBUS TCP 联机 66 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM766	MODBUS TCP 联机 67 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM767	MODBUS TCP 联机 68 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM768	MODBUS TCP 联机 69 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM769	MODBUS TCP 联机 70 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM770	MODBUS TCP 联机 71 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM771	MODBUS TCP 联机 72 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM772	MODBUS TCP 联机 73 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM773	MODBUS TCP 联机 74 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM774	MODBUS TCP 联机 75 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM775	MODBUS TCP 联机 76 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM776	MODBUS TCP 联机 77 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM777	MODBUS TCP 联机 78 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM778	MODBUS TCP 联机 79 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM779	MODBUS TCP 联机 80 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM780	MODBUS TCP 联机 81 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM781	MODBUS TCP 联机 82 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM782	MODBUS TCP 联机 83 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM783	MODBUS TCP 联机 84 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM784	MODBUS TCP 联机 85 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM785	MODBUS TCP 联机 86 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM786	MODBUS TCP 联机 87 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM787	MODBUS TCP 联机 88 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM788	MODBUS TCP 联机 89 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM789	MODBUS TCP 联机 90 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM790	MODBUS TCP 联机 91 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM791	MODBUS TCP 联机 92 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM792	MODBUS TCP 联机 93 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM793	MODBUS TCP 联机 94 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM794	MODBUS TCP 联机 95 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM795	MODBUS TCP 联机 96 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM796	MODBUS TCP 联机 97 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM797	MODBUS TCP 联机 98 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM798	MODBUS TCP 联机 99 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM799	MODBUS TCP 联机 100 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM800	MODBUS TCP 联机 101 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM801	MODBUS TCP 联机 102 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM802	MODBUS TCP 联机 103 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM803	MODBUS TCP 联机 104 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM804	MODBUS TCP 联机 105 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM805	MODBUS TCP 联机 106 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM806	MODBUS TCP 联机 107 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM807	MODBUS TCP 联机 108 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM808	MODBUS TCP 联机 109 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM809	MODBUS TCP 联机 110 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM810	MODBUS TCP 联机 111 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM811	MODBUS TCP 联机 112 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM812	MODBUS TCP 联机 113 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM813	MODBUS TCP 联机 114 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM814	MODBUS TCP 联机 115 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM815	MODBUS TCP 联机 116 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM816	MODBUS TCP 联机 117 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM817	MODBUS TCP 联机 118 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF
SM818	MODBUS TCP 联机 119 启动标志	X	0*3	X	0*5	0	OFF	-	-	R/W	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM819	MODBUS TCP 联机 120 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM820	MODBUS TCP 联机 121 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM821	MODBUS TCP 联机 122 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM822	MODBUS TCP 联机 123 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM823	MODBUS TCP 联机 124 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM824	MODBUS TCP 联机 125 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM825	MODBUS TCP 联机 126 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM826	MODBUS TCP 联机 127 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM827	MODBUS TCP 联机 128 启动标志	X	O*3	X	O*5	O	OFF	-	-	R/W	OFF
SM828	MODBUS TCP 联机 1 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM829	MODBUS TCP 联机 2 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM830	MODBUS TCP 联机 3 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM831	MODBUS TCP 联机 4 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM832	MODBUS TCP 联机 5 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM833	MODBUS TCP 联机 6 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM834	MODBUS TCP 联机 7 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM835	MODBUS TCP 联机 8 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM836	MODBUS TCP 联机 9 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM837	MODBUS TCP 联机 10 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM838	MODBUS TCP 联机 11 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM839	MODBUS TCP 联机 12 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM840	MODBUS TCP 联机 13 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM841	MODBUS TCP 联机 14 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM842	MODBUS TCP 联机 15 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM843	MODBUS TCP 联机 16 错误标志	X	V1.01	X	O	O	OFF	-	-	R	OFF
SM844	MODBUS TCP 联机 17 错误标志	X	O*1	X	O*6	O	OFF	-	-	R	OFF
SM845	MODBUS TCP 联机 18 错误标志	X	O*1	X	O*6	O	OFF	-	-	R	OFF
SM846	MODBUS TCP 联机 19 错误标志	X	O*1	X	O*6	O	OFF	-	-	R	OFF
SM847	MODBUS TCP 联机 20 错误标志	X	O*1	X	O*6	O	OFF	-	-	R	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM848	MODBUS TCP 联机 21 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM849	MODBUS TCP 联机 22 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM850	MODBUS TCP 联机 23 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM851	MODBUS TCP 联机 24 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM852	MODBUS TCP 联机 25 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM853	MODBUS TCP 联机 26 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM854	MODBUS TCP 联机 27 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM855	MODBUS TCP 联机 28 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM856	MODBUS TCP 联机 29 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM857	MODBUS TCP 联机 30 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM858	MODBUS TCP 联机 31 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM859	MODBUS TCP 联机 32 错误标志	X	0*1	X	0*6	0	OFF	-	-	R	OFF
SM860	MODBUS TCP 联机 33 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM861	MODBUS TCP 联机 34 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM862	MODBUS TCP 联机 35 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM863	MODBUS TCP 联机 36 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM864	MODBUS TCP 联机 37 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM865	MODBUS TCP 联机 38 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM866	MODBUS TCP 联机 39 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM867	MODBUS TCP 联机 40 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM868	MODBUS TCP 联机 41 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM869	MODBUS TCP 联机 42 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM870	MODBUS TCP 联机 43 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM871	MODBUS TCP 联机 44 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM872	MODBUS TCP 联机 45 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM873	MODBUS TCP 联机 46 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM874	MODBUS TCP 联机 47 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM875	MODBUS TCP 联机 48 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM876	MODBUS TCP 联机 49 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM877	MODBUS TCP 联机 50 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM878	MODBUS TCP 联机 51 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM879	MODBUS TCP 联机 52 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM880	MODBUS TCP 联机 53 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM881	MODBUS TCP 联机 54 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM882	MODBUS TCP 联机 55 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM883	MODBUS TCP 联机 56 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM884	MODBUS TCP 联机 57 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM885	MODBUS TCP 联机 58 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM886	MODBUS TCP 联机 59 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM887	MODBUS TCP 联机 60 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM888	MODBUS TCP 联机 61 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM889	MODBUS TCP 联机 62 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM890	MODBUS TCP 联机 63 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM891	MODBUS TCP 联机 64 错误标志	X	0*2	X	0*4	0	OFF	-	-	R	OFF
SM892	MODBUS TCP 联机 65 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM893	MODBUS TCP 联机 66 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM894	MODBUS TCP 联机 67 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM895	MODBUS TCP 联机 68 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM896	MODBUS TCP 联机 69 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM897	MODBUS TCP 联机 70 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM898	MODBUS TCP 联机 71 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM899	MODBUS TCP 联机 72 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM900	MODBUS TCP 联机 73 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM901	MODBUS TCP 联机 74 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM902	MODBUS TCP 联机 75 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM903	MODBUS TCP 联机 76 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM904	MODBUS TCP 联机 77 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM905	MODBUS TCP 联机 78 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM906	MODBUS TCP 联机 79 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM907	MODBUS TCP 联机 80 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM908	MODBUS TCP 联机 81 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM909	MODBUS TCP 联机 82 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM910	MODBUS TCP 联机 83 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM911	MODBUS TCP 联机 84 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM912	MODBUS TCP 联机 85 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM913	MODBUS TCP 联机 86 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM914	MODBUS TCP 联机 87 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM915	MODBUS TCP 联机 88 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM916	MODBUS TCP 联机 89 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM917	MODBUS TCP 联机 90 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM918	MODBUS TCP 联机 91 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM919	MODBUS TCP 联机 92 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM920	MODBUS TCP 联机 93 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM921	MODBUS TCP 联机 94 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM922	MODBUS TCP 联机 95 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM923	MODBUS TCP 联机 96 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM924	MODBUS TCP 联机 97 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM925	MODBUS TCP 联机 98 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM926	MODBUS TCP 联机 99 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM927	MODBUS TCP 联机 100 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM928	MODBUS TCP 联机 101 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM929	MODBUS TCP 联机 102 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM930	MODBUS TCP 联机 103 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM931	MODBUS TCP 联机 104 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM932	MODBUS TCP 联机 105 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM933	MODBUS TCP 联机 106 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM934	MODBUS TCP 联机 107 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM935	MODBUS TCP 联机 108 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM936	MODBUS TCP 联机 109 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM937	MODBUS TCP 联机 110 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM938	MODBUS TCP 联机 111 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM939	MODBUS TCP 联机 112 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM940	MODBUS TCP 联机 113 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM941	MODBUS TCP 联机 114 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM942	MODBUS TCP 联机 115 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM943	MODBUS TCP 联机 116 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM944	MODBUS TCP 联机 117 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM945	MODBUS TCP 联机 118 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM946	MODBUS TCP 联机 119 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM947	MODBUS TCP 联机 120 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM948	MODBUS TCP 联机 121 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM949	MODBUS TCP 联机 122 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM950	MODBUS TCP 联机 123 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM951	MODBUS TCP 联机 124 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM952	MODBUS TCP 联机 125 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM953	MODBUS TCP 联机 126 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM954	MODBUS TCP 联机 127 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
SM955	MODBUS TCP 联机 128 错误标志	X	0*3	X	0*5	0	OFF	-	-	R	OFF
*SM1000	Ethernet 设定标志。 ON 时，将 SR1000~SR1006 的数据 写入闪存	X	0	X	0	0	-	-	-	R/W	OFF
*SM1001	Ethernet Port1 网络线联机状态	X	X	X	0	0	-	-	-	R	OFF
*SM1002	Ethernet Port2 网络线联机状态	X	X	X	X	0	-	-	-	R	OFF
*SM1003	光纤同步电缆连接状态	X	X	X	X	0	-	-	-	R	OFF
*SM1089	MODBUS TCP Server 联机已满	X	V1.06	X	V2.0 0	0	OFF	-	-	R	OFF
*SM1090	TCP 联机忙碌	X	0	X	0	0	OFF	-	-	R	OFF

SM	功能说明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SM1091	UDP 联机忙碌	X	O	X	O	O	OFF	-	-	R	OFF
SM1100	网络线未连接	X	O	X	O	O	OFF	-	-	R	OFF
*SM1106	以太网网络联机错误标志	X	O	X	O	O	OFF	-	-	R	OFF
*SM1107	HWCONFIG 以太网网络-基本设定 参数设定错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1108	HWCONFIG 以太网网络-进阶设定 过滤器设定错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1109	HWCONFIG 以太网网络-进阶设定 TCP/UDP 所设定的本机通讯端口已被使用	X	O	X	O	O	OFF	-	-	R	OFF
*SM1112	Email 相关的参数设定错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1113	Email 发送服务失败	X	O	X	O	O	OFF	-	-	R	OFF
*SM1116	Email 发送条件 1.开关	X	O	X	O	O	OFF	-	-	R	OFF
*SM1117	Email 发送条件 1	X	O	X	O	O	OFF	-	-	R	OFF
*SM1118	Email 发送条件 1 被触发，因以太网网络联机失败，邮件无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1119	Email 发送条件 1 被触发，邮件发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1120	Email 发送条件 1 被触发，因邮件内容有错误，无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1121	Email 发送条件 1 被触发，邮件发送中。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1122	Email 发送条件 1 被触发，SMTP 回复超时	X	O	X	O	O	OFF	-	-	R	OFF
*SM1123	Email 发送条件 1 被触发，SMTP 回复错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1124	Email 发送条件 1 被触发，附件大小错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1125	Email 发送条件 1 被触发，找不到附件	X	O	X	O	O	OFF	-	-	R	OFF
*SM1126	Email 发送条件 2.开关	X	O	X	O	O	OFF	-	-	R	OFF
*SM1127	Email 发送条件 2	X	O	X	O	O	OFF	-	-	R	OFF
*SM1128	Email 发送条件 2 被触发，因以太网网络联机失败，邮件无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM1129	Email 发送条件 2 被触发 · 邮件发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1130	Email 发送条件 2 被触发 · 因邮件内容有错误 · 无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1131	Email 发送条件 2 被触发 · 邮件发送中。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1132	Email 发送条件 2 被触发 · SMTP 回复超时	X	O	X	O	O	OFF	-	-	R	OFF
*SM1133	Email 发送条件 2 被触发 · SMTP 回复错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1134	Email 发送条件 2 被触发 · 附件大小错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1135	Email 发送条件 2 被触发 · 找不到附件	X	O	X	O	O	OFF	-	-	R	OFF
*SM1136	Email 发送条件 3.开关	X	O	X	O	O	OFF	-	-	R	OFF
*SM1137	Email 发送条件 3	X	O	X	O	O	OFF	-	-	R	OFF
*SM1138	Email 发送条件 3 被触发 · 因以太网网络联机失败 · 邮件无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1139	Email 发送条件 3 被触发 · 邮件发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1140	Email 发送条件 3 被触发 · 因邮件内容有错误 · 无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1141	Email 发送条件 3 被触发 · 邮件发送中。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1142	Email 发送条件 3 被触发 · SMTP 回复超时	X	O	X	O	O	OFF	-	-	R	OFF
*SM1143	Email 发送条件 3 被触发 · SMTP 回复错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1144	Email 发送条件 3 被触发 · 附件大小错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1145	Email 发送条件 3 被触发 · 找不到附件	X	O	X	O	O	OFF	-	-	R	OFF
*SM1146	Email 发送条件 4.开关	X	O	X	O	O	OFF	-	-	R	OFF
*SM1147	Email 发送条件 4	X	O	X	O	O	OFF	-	-	R	OFF
*SM1148	Email 发送条件 4 被触发 · 因以太网网络联机失败 · 邮件无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1149	Email 发送条件 4 被触发 · 邮件发送成	X	O	X	O	O	OFF	-	-	R	OFF

SM	功能说明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
	功。										
*SM1150	Email 发送条件 4 被触发·因邮件内容有错误·无法发送成功。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1151	Email 发送条件 4 被触发·邮件发送中。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1152	Email 发送条件 4 被触发·SMTP 回复超时	X	0	X	0	0	OFF	-	-	R	OFF
*SM1153	Email 发送条件 4 被触发·SMTP 回复错误	X	0	X	0	0	OFF	-	-	R	OFF
*SM1154	Email 发送条件 4 被触发·附件大小错误	X	0	X	0	0	OFF	-	-	R	OFF
*SM1155	Email 发送条件 4 被触发·找不到附件	X	0	X	0	0	OFF	-	-	R	OFF
*SM1156	Email 发送条件 5.开关	X	0	X	0	0	OFF	-	-	R	OFF
*SM1157	Email 发送条件 5	X	0	X	0	0	OFF	-	-	R	OFF
*SM1158	Email 发送条件 5 被触发·因以太网网络联机失败·邮件无法发送成功。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1159	Email 发送条件 5 被触发·邮件发送成功。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1160	Email 发送条件 5 被触发·因邮件内容有错误·无法发送成功。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1161	Email 发送条件 5 被触发·邮件发送中。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1162	Email 发送条件 5 被触发·SMTP 回复超时	X	0	X	0	0	OFF	-	-	R	OFF
*SM1163	Email 发送条件 5 被触发·SMTP 回复错误	X	0	X	0	0	OFF	-	-	R	OFF
*SM1164	Email 发送条件 5 被触发·附件大小错误	X	0	X	0	0	OFF	-	-	R	OFF
*SM1165	Email 发送条件 5 被触发·找不到附件	X	0	X	0	0	OFF	-	-	R	OFF
*SM1166	Email 发送条件 6.开关	X	0	X	0	0	OFF	-	-	R	OFF
*SM1167	Email 发送条件 6	X	0	X	0	0	OFF	-	-	R	OFF
*SM1168	Email 发送条件 6 被触发·因以太网网络联机失败·邮件无法发送成功。	X	0	X	0	0	OFF	-	-	R	OFF
*SM1169	Email 发送条件 6 被触发·邮件发送成功。	X	0	X	0	0	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM1170	Email 发送条件 6 被触发，因邮件内容有错误，无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1171	Email 发送条件 6 被触发，邮件发送中。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1172	Email 发送条件 6 被触发，SMTP 回复超时	X	O	X	O	O	OFF	-	-	R	OFF
*SM1173	Email 发送条件 6 被触发，SMTP 回复错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1174	Email 发送条件 6 被触发，附件大小错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1175	Email 发送条件 6 被触发，找不到附件	X	O	X	O	O	OFF	-	-	R	OFF
*SM1176	Email 发送条件 7.开关	X	O	X	O	O	OFF	-	-	R	OFF
*SM1177	Email 发送条件 7	X	O	X	O	O	OFF	-	-	R	OFF
*SM1178	Email 发送条件 7 被触发，因以太网网络联机失败，邮件无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1179	Email 发送条件 7 被触发，邮件发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1180	Email 发送条件 7 被触发，因邮件内容有错误，无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1181	Email 发送条件 7 被触发，邮件发送中。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1182	Email 发送条件 7 被触发，SMTP 回复超时	X	O	X	O	O	OFF	-	-	R	OFF
*SM1183	Email 发送条件 7 被触发，SMTP 回复错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1184	Email 发送条件 7 被触发，附件大小错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1185	Email 发送条件 7 被触发，找不到附件	X	O	X	O	O	OFF	-	-	R	OFF
*SM1186	Email 发送条件 8.开关	X	O	X	O	O	OFF	-	-	R	OFF
*SM1187	Email 发送条件 8	X	O	X	O	O	OFF	-	-	R	OFF
*SM1188	Email 发送条件 8 被触发，因以太网网络联机失败，邮件无法发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1189	Email 发送条件 8 被触发，邮件发送成功。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1190	Email 发送条件 8 被触发，因邮件内容	X	O	X	O	O	OFF	-	-	R	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
	有错误，无法发送成功。										
*SM1191	Email 发送条件 8 被触发，邮件发送中。	X	O	X	O	O	OFF	-	-	R	OFF
*SM1192	Email 发送条件 8 被触发，SMTP 回复超时	X	O	X	O	O	OFF	-	-	R	OFF
*SM1193	Email 发送条件 8 被触发，SMTP 回复错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1194	Email 发送条件 8 被触发，附件大小错误	X	O	X	O	O	OFF	-	-	R	OFF
*SM1195	Email 发送条件 8 被触发，找不到附件	X	O	X	O	O	OFF	-	-	R	OFF
SM1196	Socket 配置错误	X	O	X	O	O	OFF	-	-	R/W	OFF
SM1270	TCP Socket 1--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1271	TCP Socket 1--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1272	TCP Socket 1--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1273	TCP Socket 1--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1274	TCP Socket 1--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1275	TCP Socket 1--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1276	TCP Socket 1--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1277	TCP Socket 1--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1278	TCP Socket 2--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1279	TCP Socket 2--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1280	TCP Socket 2--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1281	TCP Socket 2--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1282	TCP Socket 2--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1283	TCP Socket 2--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1284	TCP Socket 2--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1285	TCP Socket 2--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1286	TCP Socket 3--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1287	TCP Socket 3--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1288	TCP Socket 3--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1289	TCP Socket 3--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM1290	TCP Socket 3--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1291	TCP Socket 3--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1292	TCP Socket 3--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1293	TCP Socket 3--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1294	TCP Socket 4--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1295	TCP Socket 4--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1296	TCP Socket 4--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1297	TCP Socket 4--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1298	TCP Socket 4--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1299	TCP Socket 4--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1300	TCP Socket 4--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1301	TCP Socket 4--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1302	TCP Socket 5--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1303	TCP Socket 5--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1304	TCP Socket 5--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1305	TCP Socket 5--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1306	TCP Socket 5--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1307	TCP Socket 5--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1308	TCP Socket 5--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1309	TCP Socket 5--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1310	TCP Socket 6--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1311	TCP Socket 6--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1312	TCP Socket 6--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1313	TCP Socket 6--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1314	TCP Socket 6--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1315	TCP Socket 6--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1316	TCP Socket 6--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1317	TCP Socket 6--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1318	TCP Socket 7--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM1319	TCP Socket 7--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1320	TCP Socket 7--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1321	TCP Socket 7--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1322	TCP Socket 7--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1323	TCP Socket 7--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1324	TCP Socket 7--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1325	TCP Socket 7--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1326	TCP Socket 8--联机标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1327	TCP Socket 8--数据接收完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1328	TCP Socket 8--数据传送完成标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1329	TCP Socket 8--开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1330	TCP Socket 8--关闭标志	X	O	X	O	O	ON	ON	-	R	ON
SM1331	TCP Socket 8--传送中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1332	TCP Socket 8--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1333	TCP Socket 8--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1334	UDP Socket 1--联机已开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1335	UDP Socket 1--数据已接收标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1336	UDP Socket 1--数据已传送标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1337	UDP Socket 1--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1338	UDP Socket 1--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1339	UDP Socket 2--联机已开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1340	UDP Socket 2--数据已接收标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1341	UDP Socket 2--数据已传送标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1342	UDP Socket 2--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1343	UDP Socket 2--错误标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1344	UDP Socket 3--联机已开启标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1345	UDP Socket 3--数据已接收标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1346	UDP Socket 3--数据已传送标志	X	O	X	O	O	OFF	OFF	-	R	OFF
SM1347	UDP Socket 3--接收中标志	X	O	X	O	O	OFF	OFF	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM1348	UDP Socket 3--错误标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1349	UDP Socket 4--联机已开启标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1350	UDP Socket 4--数据已接收标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1351	UDP Socket 4--数据已传送标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1352	UDP Socket 4--接收中标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1353	UDP Socket 4--错误标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1354	UDP Socket 5--联机已开启标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1355	UDP Socket 5--数据已接收标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1356	UDP Socket 5--数据已传送标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1357	UDP Socket 5--接收中标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1358	UDP Socket 5--错误标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1359	UDP Socket 6--联机已开启标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1360	UDP Socket 6--数据已接收标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1361	UDP Socket 6--数据已传送标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1362	UDP Socket 6--接收中标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1363	UDP Socket 6--错误标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1364	UDP Socket 7--联机已开启标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1365	UDP Socket 7--数据已接收标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1366	UDP Socket 7--数据已传送标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1367	UDP Socket 7--接收中标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1368	UDP Socket 7--错误标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1369	UDP Socket 8--联机已开启标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1370	UDP Socket 8--数据已接收标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1371	UDP Socket 8--数据已传送标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1372	UDP Socket 8--接收中标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1373	UDP Socket 8--错误标志	X	0	X	0	0	OFF	OFF	-	R	OFF
SM1374	Web 设定错误	X	0	X	0	0	OFF	-	-	R	OFF
SM1375	TCP Socket 1--奇数字节启动标志	X	V1.05	X	0	0	OFF	-	-	R/W	OFF
SM1376	TCP Socket 2--奇数字节启动标志	X	V1.05	X	0	0	OFF	-	-	R/W	OFF
SM1377	TCP Socket 3--奇数字节启动标志	X	V1.05	X	0	0	OFF	-	-	R/W	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
SM1378	TCP Socket 4--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1379	TCP Socket 5--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1380	TCP Socket 6--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1381	TCP Socket 7--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1382	TCP Socket 8--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1383	UDP Socket 1--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1384	UDP Socket 2--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1385	UDP Socket 3--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1386	UDP Socket 4--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1387	UDP Socket 5--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1388	UDP Socket 6--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1389	UDP Socket 7--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
SM1390	UDP Socket 8--奇数字节启动标志	X	V1.05	X	O	O	OFF	-	-	R/W	OFF
*SM1392 ↓ SM1423	PLC LINK 联机标志 (从站 ID 1) ↓ PLC LINK 联机标志 (从站 ID 32)	O	O	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 联机标志 (从站 ID 1) ↓ COM1-MODBUS 联机标志 (从站 ID 32)	X	X	O	O	O	-	-	-	R/W	OFF
*SM1424 ↓ SM1455	PLC LINK 数据交换标志 (从站 ID 1) ↓ PLC LINK 数据交换标志(从站 ID 32)	O	O	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 数据交换中标志(从站 ID 1) ↓ COM1-MODBUS 数据交换中标志(从站 ID 32)	X	X	O	O	O	OFF	-	-	R	OFF
*SM1456 ↓ SM1487	PLC LINK 读取错误 (从站 ID 1) ↓ PLC LINK 读取错误 (从站 ID 32)	O	O	X	X	X	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	COM1-MODBUS 读取错误 (从站 ID 1) ↓ COM1-MODBUS 读取错误 (从站 ID 32)	X	X	O	O	O	OFF	-	-	R	OFF
	PLC LINK 写入错误 (从站 ID 1) ↓ PLC LINK 写入错误 (从站 ID 32)	O	O	X	X	X	OFF	-	-	R	OFF
*SM1488 ↓ SM1519	COM1-MODBUS 写入错误 (从站 ID 1) ↓ COM1-MODBUS 写入错误 (从站 ID 32)	X	X	O	O	O	OFF	-	-	R	OFF
	PLC LINK 读取数据完毕 (从站 ID 1) ↓ PLC LINK 读取数据完毕 (从站 ID 32)	O	O	X	X	X	OFF	-	-	R	OFF
*SM1520 ↓ SM1551	COM1-MODBUS 读取数据完毕 (从站 ID 1) ↓ COM1-MODBUS 读取数据完毕 (从站 ID 32)	X	X	O	O	O	OFF	-	-	R	OFF
	PLC LINK 写入数据完毕 (从站 ID 1) ↓ PLC LINK 写入数据完毕 (从站 ID 32)	O	O	X	X	X	OFF	-	-	R	OFF
*SM1552 ↓ SM1583	COM1-MODBUS 写入数据完毕 (从站 ID 1) ↓ COM1-MODBUS 写入数据完毕 (从站 ID 32)	X	X	O	O	O	OFF	-	-	R	OFF
*SM1584	PLC LINK 联机启动标志	O	O	X	X	X	OFF	-	-	R/W	OFF
	COM1-MODBUS 联机启动标志	X	X	O	O	O	OFF	-	-	R/W	OFF
*SM1585	PLC LINK 用户分配标志	O	O	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 用户分配标志	X	X	O	O	O	-	-	-	R/W	OFF
*SM1586	PLC LINK 自动模式标志	O	O	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 自动模式标志	X	X	O	O	O	-	-	-	R/W	OFF
*SM1587	PLC LINK 手动模式标志	O	O	X	X	X	-	-	-	R/W	OFF

SM	功能说明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SM1588	PLC LINK 检测标志	0	0	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 检测标志	X	X	0	0	0	OFF	-	-	R	OFF
*SM1589	PLC LINK 错误标志	0	0	X	X	X	OFF	-	-	R	OFF
*SM1590	PLC LINK 装置地址错误	0	0	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 装置地址错误	X	X	0	0	0	OFF	-	-	R	OFF
*SM1591	PLC LINK 逾时	0	0	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 逾时	X	X	0	0	0	OFF	-	-	R	OFF
*SM1592	PLC LINK 限制时间错误	0	0	X	X	X	OFF	-	-	R	OFF
*SM1593	PLC LINK 标准 MODBUS 通讯协议 (0) /AH 通讯协议 (1)	0	0	X	X	X	-	-	-	R/W	OFF
*SM1594	PLC LINK 在目前环境自动检测从站。 只有在 PLC Link 于停止模式可被使用。 OFF：检测结束或等待检测（预设）。 ON 正在检测从站。	0	0	X	X	X	OFF	-	-	R/W	OFF
	COM1-MODBUS 在目前环境自动检测从站。 OFF:检测结束或等待检测(预设)。 ON:正在检测从站。只有在 COM1-MODBUS 于停止模式可被使用。	X	X	0	0	0	OFF	-	-	R/W	OFF
*SM1595	用户分配从站 ID (1) /自动分配从站 ID (0)	0	0	X	X	X	-	-	-	R/W	OFF
*SM1596	PLC LINK 联机操作错误标志	0	0	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 联机操作错误标志	X	X	0	0	0	OFF	-	-	R	OFF
*SM1597	PLC LINK 扩展端口(SM1597=ON 表示启动扩展端口)	0	0	X	X	X	-	-	-	R/W	OFF
*SM1598	PLC LINK 同步读/写功能 (SM1598=ON 表示启动同步读写)	0	0	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 同步读/写功能 (SM1598=ON 表示启动同步读写)	X	X	0	0	0	-	-	-	R/W	OFF
SM1599 ↓ SM1630	COM1-MODBUS 从站 1 启动标志 ↓ COM1-MODBUS 从站 32 启动标志	X	X	0	0	0	OFF	-	-	R/W	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
SM1720 ↓ SM1751	PLC LINK 变更读取功能码为 0x04 标志 (从站 ID 1) ↓ PLC LINK 变更读取功能码为 0x04 标志 (从站 ID 32)	0	0	X	X	X	OFF	OFF	-	R/W	OFF
SM1752 ↓ SM1768	COM2-MODBUS 数据交换中标志(从站 ID 1) ↓ COM2-MODBUS 数据交换中标志(从站 ID 17)	X	X	0	X	X	OFF	-	-	R	OFF
SM1769	Ether Link 状态标志	0	0	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 数据交换中标志(从站 ID 18)	X	X	0	X	X	OFF	-	-	R	OFF
*SM1770	Ether Link 起始标志 (CPU)	0	0	X	X	X	OFF	-	-	R/W	OFF
	COM2-MODBUS 数据交换中标志(从站 ID 19)	X	X	0	X	X	OFF	-	-	R	OFF
*SM1771	COM2-MODBUS 数据交换中标志(从站 ID 20)	X	X	0	X	X	OFF	-	-	R	OFF
*SM1772 ↓ SM1783	Ether Link 起始标志 (端口 0) ↓ Ether Link 起始标志 (端口 11)	0	0	X	X	X	OFF	-	-	R/W	OFF
	COM2-MODBUS 数据交换中标志(从站 ID 21) ↓ COM2-MODBUS 数据交换中标志(从站 ID 32)	X	X	0	X	X	OFF	-	-	R	OFF
*SM1784 ↓ SM1787	Ether Link 起始标志 (端口 12) ↓ Ether Link 起始标志 (端口 15)	0	0	X	X	X	OFF	-	-	R/W	OFF
	COM2-MODBUS 从站 1 读取错误标志 ↓ COM2-MODBUS 从站 4 读取错误标志	X	X	0	X	X	OFF	-	-	R	OFF
*SM1788	Ether Link 错误标志 (CPU)	0	0	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 从站 5 读取错误标志	X	X	0	X	X	OFF	-	-	R	OFF

SM	功能说明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	志										
*SM1789	COM2-MODBUS 从站 6 读取错误标志	X	X	O	X	X	OFF	-	-	R	OFF
*SM1790 ↓ SM1805	Ether Link 错误标志 (端口 0) ↓ Ether Link 错误标志 (端口 15)	O	O	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 从站 7 读取错误标志 ↓ COM2-MODBUS 从站 22 读取错误标志	X	X	O	X	X	OFF	-	-	R	OFF
SM1806	Ether Link 错误标志 (CPU)	O	O	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 从站 23 读取错误标志	X	X	O	X	X	OFF	-	-	R	OFF
*SM1807	COM2-MODBUS 从站 24 读取错误标志	X	X	O	X	X	OFF	-	-	R	OFF
*SM1808 ↓ SM1815	Ether Link 状态标志 (端口 0) ↓ Ether Link 状态标志 (端口 7)	O	O	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 从站 25 读取错误标志 ↓ COM2-MODBUS 从站 32 读取错误标志	X	X	O	X	X	OFF	-	-	R	OFF
	Ether Link 状态标志 (端口 8) ↓ Ether Link 状态标志 (端口 15)	O	O	X	X	X	OFF	-	-	R	OFF
*SM1816 ↓ SM1823	COM2-MODBUS 从站 1 写入错误标志 ↓ COM2-MODBUS 从站 8 写入错误标志	X	X	O	X	X	OFF	-	-	R	OFF
SM1824 ↓ SM1847	Ether Link 区块 1 主动标志 ↓ Ether Link 区块 24 主动标志	O	O	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 从站 9 写入错误标志 ↓	X	X	O	X	X	OFF	-	-	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	COM2-MODBUS 从站 32 写入错误标志										
	Ether Link 区块 25 主动标志 ↓ Ether Link 区块 56 主动标志	0	0	X	X	X	OFF	-	-	R	OFF
SM1848 ↓ SM1879	COM2-MODBUS 从站 1 读取数据完毕标志 ↓ COM2-MODBUS 从站 32 读取数据完毕标志	X	X	0	X	X	OFF	-	-	R	OFF
	Ether Link 区块 57 主动标志 ↓ Ether Link 区块 88 主动标志	0	0	X	X	X	OFF	-	-	R	OFF
SM1880 ↓ SM1911	COM2-MODBUS 从站 1 写入数据完毕标志 ↓ COM2-MODBUS 从站 32 写入数据完毕标志	X	X	0	X	X	OFF	-	-	R	OFF
SM1912	Ether Link 区块 89 主动标志	0	0	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 同步读/写功能	X	X	0	X	X	OFF	-	-	R/W	OFF
SM1913 ↓ SM1944	Ether Link 区块 90 主动标志 ↓ Ether Link 区块 121 主动标志 ↓ COM2-MODBUS 从站 1 启动标志 ↓ COM2-MODBUS 从站 32 启动标志	0	0	X	X	X	OFF	-	-	R	OFF
		X	X	0	X	X	OFF	-	-	R/W	OFF
SM1945 ↓ SM1951	Ether Link 区块 122 主动标志 ↓ Ether Link 区块 128 主动标志	0	0	X	X	X	OFF	-	-	R	OFF
SM2000	EMDRW 1 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2001	EMDRW 1 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2002	EMDRW 1 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2003	EMDRW 1 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2004	EMDRW 1 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF	STOP	RUN	属性	出厂设定
							↓ ON	↓ RUN	↓ STOP		
SM2005	EMDRW 1 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2006	EMDRW 2 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2007	EMDRW 2 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2008	EMDRW 2 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2009	EMDRW 2 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2010	EMDRW 2 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2011	EMDRW 2 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2012	EMDRW 3 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2013	EMDRW 3 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2014	EMDRW 3 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2015	EMDRW 3 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2016	EMDRW 3 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2017	EMDRW 3 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2018	EMDRW 4 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2019	EMDRW 4 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2020	EMDRW 4 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2021	EMDRW 4 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2022	EMDRW 4 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2023	EMDRW 4 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2024	EMDRW 5 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2025	EMDRW 5 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2026	EMDRW 5 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2027	EMDRW 5 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2028	EMDRW 5 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2029	EMDRW 5 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2030	EMDRW 6 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2031	EMDRW 6 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2032	EMDRW 6 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2033	EMDRW 6 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF

2

SM	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SM2034	EMDRW 6 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2035	EMDRW 6 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2036	EMDRW 7 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2037	EMDRW 7 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2038	EMDRW 7 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2039	EMDRW 7 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2040	EMDRW 7 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2041	EMDRW 7 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2042	EMDRW 8 送信标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2043	EMDRW 8 等待标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2044	EMDRW 8 接收标志	X	0	X	0	0	OFF	OFF	OFF	R/W	OFF
SM2045	EMDRW 8 错误标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2046	EMDRW 8 逾时标志	X	0	X	0	0	OFF	OFF	OFF	R	OFF
SM2047	EMDRW 8 关闭标志	X	0	X	0	0	ON	ON	ON	R	ON
SM2048 ↓ SM2079	EtherNet/IP I/O connection 1 错误标志 ↓ EtherNet/IP I/O connection 32 错误标志	X	X	X	V2.0 0	0	OFF	-	-	R	OFF
SM2080 ↓ SM2111	EtherNet/IP I/O connection 33 错误标志 ↓ EtherNet/IP I/O connection 64 错误标志	X	X	X	V2.0 0*6	0	OFF	-	-	R	OFF
SM2112 ↓ SM2175	EtherNet/IP I/O connection 65 错误标志 ↓ EtherNet/IP I/O connection 128 错误标志	X	X	X	V2.0 0*4	0	OFF	-	-	R	OFF
SM2176 ↓ SM2303	EtherNet/IP I/O connection 129 错误标志 ↓ EtherNet/IP I/O connection 256 错误标志	X	X	X	V2.0 0*5	0	OFF	-	-	R	OFF

注：有*之 SM 请参考 SM/SR 补充说明

*1：仅支持机种 AHCPU530-EN、AHCPU520-EN 及 AHCPU510-EN。

*2：仅支持机种 AHCPU530-EN 及 AHCPU520-EN。

*3：仅支持机种 AHCPU530-EN。

*4：仅支持机种 AHCPU531-EN 及 AHCPU521-EN。

*5：仅支持机种 AHCPU531-EN。

*6：仅支持机种 AHCPU531-EN、AHCPU521-EN 及 AHCPU511-EN。

2.2.8 特殊辅助继电器 SM 的更新时间

适用机种	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
	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
	SM20	曾经发生内部电源供应不足后，电源又恢复到正常状态时记录。
	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

2

适用机种	特殊辅助继电器	更新时间
	SM404	系统自动设定与清除—每 5ms 更新一次
	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
	SM418	系统自动设定与清除
	SM420~SM431	系统自动设定与清除
	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、SM601、 SM602	系统自动设定与清除—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、SM616、 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
	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	用户自行设定

适用机种	特殊辅助继电器	更新时间
AH5X0	SM699	1.每次下载 Ether Link 或数据交换参数后 2.使用系统还原功能后
AH5X1		1.每次下载 MODBUS TCP 数据交换表后 2.使用系统还原功能后
AH5X0	SM700-SM827	1.下载数据交换参数后 2.每个扫描时间更新
AH5X1		用户自行设定及清除
	SM828-SM955	1.下载数据交换参数后 2.每个扫描时间更新
	SM1000	用户自行设定
	SM1001~1003	用户自行设定
	SM1089	每个扫描时间更新
	SM1090	TCP 连接满时为 ON
	SM1091	UDP 连接满时为 ON
	SM1100	执行以太网控制指令 (API2200-API2205) 时或网络线重新连接时
	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
	SM1112	设定错误时为 ON
	SM1113	服务器错误时为 ON
	SM1116	PLC 参数的触发条件启用时为 ON
	SM1117	PLC 参数的触发条件被触发时为 ON
	SM1118	自从启用本触发条件后，无任何邮件被发送时为 ON
	SM1119	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1120	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1121	触发条件启用时且邮件已送出时为 ON
	SM1122	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1123	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1124	触发条件被触发时且附件大小超过限制时为 ON
	SM1125	触发条件被触发时且附件不存在时为 ON
	SM1126	PLC 参数的触发条件启用时为 ON
	SM1127	PLC 参数的触发条件被触发时为 ON
	SM1128	自从启用本触发条件后，无任何邮件被发送时为 ON
	SM1129	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1130	启用本触发条件且最近一封邮件发送错误时为 ON

2

适用機種	特殊辅助继电器	更新时间
	SM1131	触发条件启用时且邮件已送出时为 ON
	SM1132	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1133	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1134	触发条件被触发时且附件大小超过限制时为 ON
	SM1135	触发条件被触发时且附件不存在时为 ON
	SM1136	PLC 参数的触发条件启用时为 ON
	SM1137	PLC 参数的触发条件被触发时为 ON
	SM1138	自从启用本触发条件后，无任何邮件被发送时为 ON
	SM1139	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1140	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1141	触发条件启用时且邮件已送出时为 ON
	SM1142	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1143	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1144	触发条件被触发时且附件大小超过限制时为 ON
	SM1145	触发条件被触发时且附件不存在时为 ON
	SM1146	PLC 参数的触发条件启用时为 ON
	SM1147	PLC 参数的触发条件被触发时为 ON
	SM1148	自从启用本触发条件后，无任何邮件被发送时为 ON
	SM1149	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1150	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1151	触发条件启用时且邮件已送出时为 ON
	SM1152	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1153	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1154	触发条件被触发时且附件大小超过限制时为 ON
	SM1155	触发条件被触发时且附件不存在时为 ON
	SM1156	PLC 参数的触发条件启用时为 ON
	SM1157	PLC 参数的触发条件被触发时为 ON
	SM1158	自从启用本触发条件后，无任何邮件被发送时为 ON
	SM1159	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1160	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1161	触发条件启用时且邮件已送出时为 ON
	SM1162	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1163	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1164	触发条件被触发时且附件大小超过限制时为 ON
	SM1165	触发条件被触发时且附件不存在时为 ON
	SM1166	PLC 参数的触发条件启用时为 ON

适用机种	特殊辅助继电器	更新时间
	SM1167	PLC 参数的触发条件被触发时为 ON
	SM1168	自从启用本触发条件后·无任何邮件被发送时为 ON
	SM1169	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1170	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1171	触发条件启用时且邮件已送出时为 ON
	SM1172	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1173	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1174	触发条件被触发时且附件大小超过限制时为 ON
	SM1175	触发条件被触发时且附件不存在时为 ON
	SM1176	PLC 参数的触发条件启用时为 ON
	SM1177	PLC 参数的触发条件被触发时为 ON
	SM1178	自从启用本触发条件后·无任何邮件被发送时为 ON
	SM1179	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1180	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1181	触发条件启用时且邮件已送出时为 ON
	SM1182	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1183	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1184	触发条件被触发时且附件大小超过限制时为 ON
	SM1185	触发条件被触发时且附件不存在时为 ON
	SM1186	PLC 参数的触发条件启用时为 ON
	SM1187	PLC 参数的触发条件被触发时为 ON
	SM1188	自从启用本触发条件后·无任何邮件被发送时为 ON
	SM1189	触发条件启用时且最后一封邮件已成功送出时为 ON
	SM1190	启用本触发条件且最近一封邮件发送错误时为 ON
	SM1191	触发条件启用时且邮件已送出时为 ON
	SM1192	触发条件被触发时且 SMTP 服务器响应逾时时为 ON
	SM1193	触发条件被触发时且 SMTP 服务器响应错误时为 ON
	SM1194	触发条件被触发时且附件大小超过限制时为 ON
	SM1195	触发条件被触发时且附件不存在时为 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
AH5X0	SM1392 ↓ SM1423	PLC Link 启动且主站连接从站站号 1 时为 ON ↓ PLC Link 启动且主站连接从站站号 32 时为 ON
AH5X1		COM1-MODBUS 启动且主站连接从站站号 1 时为 ON ↓

2

适用機種	特殊辅助继电器	更新时间
		COM1-MODBUS 启动且主站连接从站站号 32 时为 ON
AH5X0	SM1424	PLC Link 启动且主站存取从站站号 1 的数据时为 ON
	↓	↓
	SM1455	PLC Link 启动且主站存取从站站号 32 的数据时为 ON
AH5X1		COM1-MODBUS 启动且主站存取从站站号 1 的数据时为 ON
		↓
		COM1-MODBUS 启动且主站存取从站站号 32 的数据时为 ON
AH5X0	SM1456	PLC Link 启动且从站站号 1 读取错误时为 ON
	↓	↓
	SM1487	PLC Link 启动且从站站号 32 读取错误时为 ON
AH5X1		COM1-MODBUS 启动且从站站号 1 读取错误时为 ON
		↓
		COM1-MODBUS 启动且从站站号 32 读取错误时为 ON
AH5X0	SM1488	PLC Link 启动且从站站号 1 写入错误时为 ON
	↓	↓
	SM1519	PLC Link 启动且从站站号 32 写入错误时为 ON
AH5X1		COM1-MODBUS 启动且从站站号 1 写入错误时为 ON
		↓
		COM1-MODBUS 启动且从站站号 32 写入错误时为 ON
AH5X0	SM1520	PLC Link 启动且主站结束读取从站站号 1 时为 ON
	↓	↓
	SM1551	PLC Link 启动且主站结束读取从站站号 32 时为 ON
AH5X1		COM1-MODBUS 启动且主站结束读取从站站号 1 时为 ON
		↓
		COM1-MODBUS 启动且主站结束读取从站站号 32 时为 ON
AH5X0	SM1552	PLC Link 启动且主站结束写入从站站号 1 时为 ON
	↓	↓
	SM1583	PLC Link 启动且主站结束写入从站站号 32 时为 ON
AH5X1	SM1552	COM1-MODBUS 启动且主站结束写入从站站号 1 时为 ON
	↓	↓
	SM1583	COM1-MODBUS 启动且主站结束写入从站站号 32 时为 ON
	SM1584~SM1587	用户自行设定及清除
	SM1588	主站检测从站时为 ON
	SM1589	错误发生时为 ON
	SM1590	装置编号错误时为 ON
	SM1591	逾时时为 ON
	SM1592	轮询次数不正确时为 ON
	SM1593~SM1595	用户自行设定及清除
AH5X0	SM1596	PLC Link 启动且有错误发生时 · SM1596 为 ON
AH5X1		COM1-MODBUS 启动且有错误发生时 · SM1596 为 ON
	SM1597~SM1630	用户自行设定及清除

适用機種	特殊辅助继电器	更新时间
	SM1720~SM1751	用户自行设定及清除
	SM1752~SM1768	每一扫描周期更新一次
AH5X0	SM1769	Ether Link 有错误发生时为 ON
AH5X1		每一扫描周期更新一次
AH5X0	SM1770	用户自行设定及清除
AH5X1		每一扫描周期更新一次
	SM1771	每一扫描周期更新一次
AH5X0	SM1772~SM1788	用户自行设定及清除
AH5X1		每一扫描周期更新一次
	SM1789	每一扫描周期更新一次
AH5X0	SM1790~SM1805	相对应之通讯端口有误时为 ON
AH5X1		每一扫描周期更新一次
AH5X0	SM1806	相对应之通讯口的 Ether Link 功能启动时为 ON
AH5X1		每一扫描周期更新一次
	SM1807	每一扫描周期更新一次
AH5X0	SM1808~SM1823	相对应之通讯口的 Ether Link 功能启动时为 ON
AH5X1		每一扫描周期更新一次
	SM1824~SM1911	每一扫描周期更新一次
AH5X0	SM1912	每一扫描周期更新一次
AH5X1		用户自行设定及清除
	SM1913~SM1951	每一扫描周期更新一次
	SM2000	用户设定·传送完毕后系统自动清除
	SM2001	EMDRW 指令执行时更新
	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	EMDRW 指令执行时更新
	SM2006	用户设定·传送完毕后系统自动清除
	SM2007	EMDRW 指令执行时更新
	SM2008	EMDRW 指令执行时更新
	SM2009	EMDRW 指令执行时且有错误发生时更新
	SM2010	EMDRW 指令执行时且响应逾时时更新
	SM2011	EMDRW 指令执行时更新
	SM2012	用户设定·传送完毕后系统自动清除
	SM2013	EMDRW 指令执行时更新
	SM2014	EMDRW 指令执行时更新

2

适用机种	特殊辅助继电器	更新时间
	SM2015	EMDRW 指令执行时且有错误发生时更新
	SM2016	EMDRW 指令执行时且响应逾时时更新
	SM2017	EMDRW 指令执行时更新
	SM2018	用户设定・传送完毕后系统自动清除
	SM2019	EMDRW 指令执行时更新
	SM2020	EMDRW 指令执行时更新
	SM2021	EMDRW 指令执行时且有错误发生时更新
	SM2022	EMDRW 指令执行时且响应逾时时更新
	SM2023	EMDRW 指令执行时更新
	SM2024	用户设定・传送完毕后系统自动清除
	SM2025	EMDRW 指令执行时更新
	SM2026	EMDRW 指令执行时更新
	SM2027	EMDRW 指令执行时且有错误发生时更新
	SM2028	EMDRW 指令执行时且响应逾时时更新
	SM2029	EMDRW 指令执行时更新
	SM2030	用户设定・传送完毕后系统自动清除
	SM2031	EMDRW 指令执行时更新
	SM2032	EMDRW 指令执行时更新
	SM2033	EMDRW 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	EMDRW 指令执行时更新
	SM2036	用户设定・传送完毕后系统自动清除
	SM2037	EMDRW 指令执行时更新
	SM2038	EMDRW 指令执行时更新
	SM2039	EMDRW 指令执行时且有错误发生时更新
	SM2040	EMDRW 指令执行时且响应逾时时更新
	SM2041	EMDRW 指令执行时更新
	SM2042	用户设定・传送完毕后系统自动清除
	SM2043	EMDRW 指令执行时更新
	SM2044	EMDRW 指令执行时更新
	SM2045	EMDRW 指令执行时且有错误发生时更新
	SM2046	EMDRW 指令执行时且响应逾时时更新
	SM2047	EMDRW 指令执行时更新
	SM2048 ↓ SM2303	每一扫描周期更新一次

注：上表字段「适用机种」中，标示「AH5X0」者，表示适用机种 AHCPU500/510/520/530；标示「AH5X1」者，表示适用机种 AHCPU501/511/521/531；表格空白无任何标示，表示适用所有机种。

2.2.9 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，而且于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。

2.2.10 定时器 T

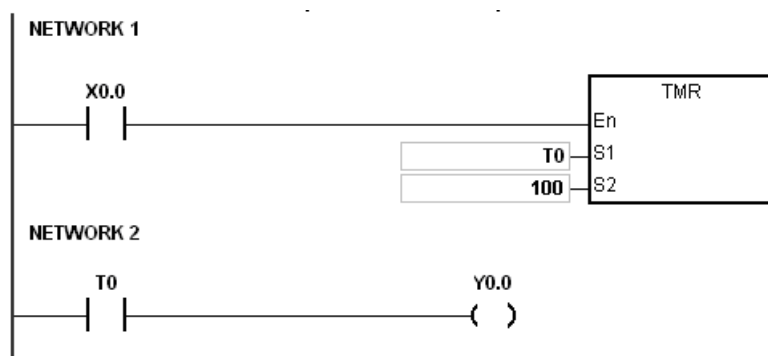
- 100ms 定时器：TMR 指令所指定之 T 定时器以 100ms 为单位计时
- 1ms 定时器：TMRH 指令所指定之 T 定时器以 1ms 为单位计时。
- 副程序专用定时器为 T1920~T2047。
- 运算型 T 定时器为 ST0~ST2047，但若要使用装置监控，就是监控 T0~T2047。
- 在程序中同一个 T 定时器如果重复使用（包含使用在不同指令 TMR、TMRH 中），则设定值以最快到达的为主。
- 在程序中同一个 T 定时器如果重复使用，其中一个条件接点 OFF 时则 T 会 OFF。
- 在程序中同一个 T 定时器如果重复使用为 T 与 ST，其中一个条件接点 OFF 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

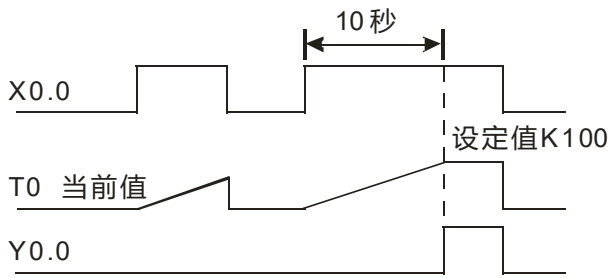
A. 一般用定时器 T

一般用定时器在 TMR 指令执行时计时一次，在 TMR 指令执行时，若计时到达，则输出线圈导通。

- 当 X0.0=ON 时，定时器 T0 的当前值以 100ms 采上数计时，当定时器当前值=设定值 100 时，输出线圈 T0=ON。
- 当 X0.0=OFF 或停电时，定时器 T0 之当前值清为 0，输出线圈 T0 变为 OFF。



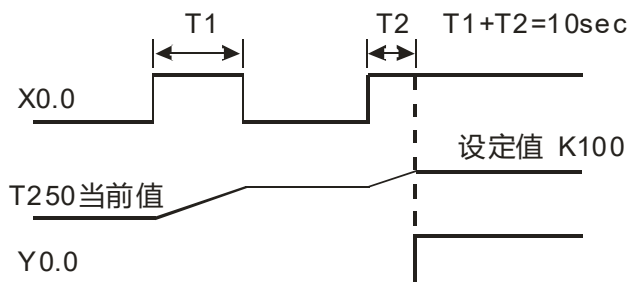
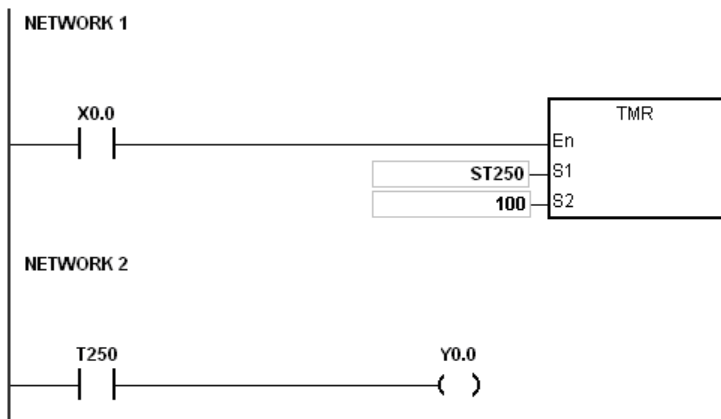
2



B. 运算型定时器 ST

运算型定时器在 TMR 指令执行时计时一次，在 TMR 指令执行时，若计时到达，则输出线圈导通。只要在装置 T 之前加上一个 S，就会变成运算型定时器 ST 装置，表示目前的 T 变成运算型定时器，则条件接点 OFF 时运算型 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。

- 当 X0.0=ON 时，定时器 T250 之当前值以 100ms 采上数计时，当定时器当前值=设定值 100 时，输出线圈 T250=ON。
- 当计时中若 X0.0=OFF 时，定时器 T250 暂停计时，当前值不变，待 X0.0 再 ON 时，继续计时，其当前值往上累加直到定时器当前值=设定值 100 时，输出线圈 T250=ON。



C. 功能块 (Function Block) 用定时器 (T/ST)

功能块或中断插入中若使用到定时器时，请使用定时器 T1920~T2047。

功能块用定时器于 TMR 指令或 END 指令执行时计时一次，在 TMR 指令或 END 指令执行时，若定时器当前值等于设定值，则输出线圈导通。

一般用定时器，若是使用在功能块或中断插入中而该功能块不被执行时，定时器就无法正确的被计时。

2.2.11 计数器 C

16 位计数器特点：

项目	16 位计数器
类型	一般型
编号	C0~C2047
计数方向	上数
设定值	0~32,767
设定值的指定	常数或数据寄存器 D
当前值的变化	计数到达设定值就不再计数
输出接点	计数到达设定值，接点导通并保持
复归动作	RST 指令被执行时当前值归零，接点被复归成 OFF
接点动作	在扫描结束时，统一动作

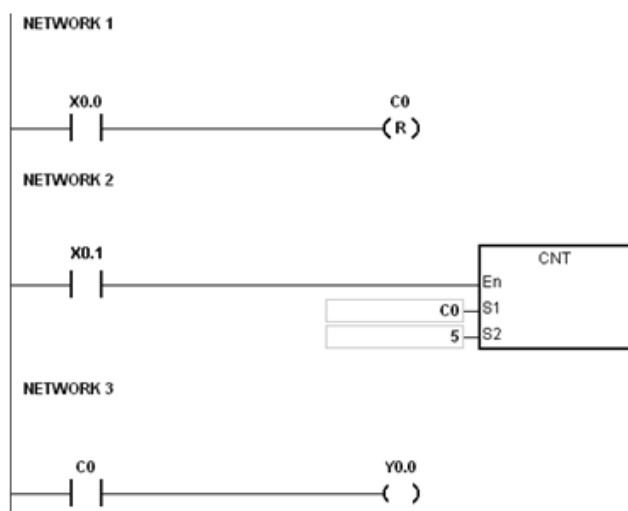
计数器的功能：

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值，亦可使用数据寄存器 D 当成设定值。

16 位计数器：

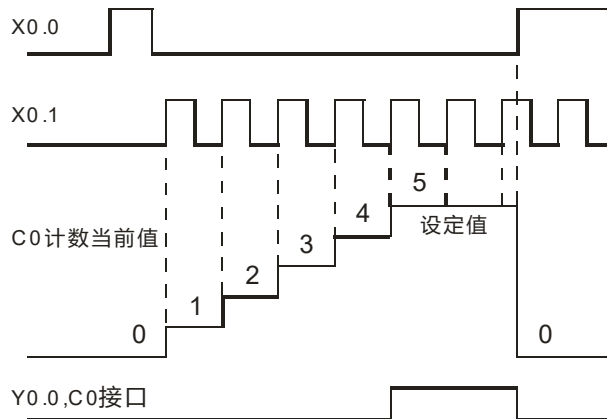
1. 16 位计数器的设定范围：0~32,767。(0 与 1 相同，在第一次计数时输出接点马上导通。)
2. 一般用计数器在 PLC 停电的时候，计数器当前值即被清除，若为停电保持型计数器会将停电前的当前值及计数器接点状态储存着，复电后会继续累计。
3. 若使用 MOV 指令、ISPSOft 将一个大于设定值的数值传送到 C0 当前值寄存器时，在下次 X0.1 由 OFF→ON 时，C0 计数器接点即变成 ON，同时当前值内容变成与设定值相同。
4. 计数器之设定值可使用常数直接设定或使用寄存器 D 中之数值作间接设定。
5. 设定值可使用常数或使用数据寄存器 D 作为设定值可以是正负数。计数器当前值由 32,767 再往上累计时则变为-32,768。

范例：



2

1. 当 X0.0=ON 时 RST 指令被执行，C0 的当前值归零，输出接点被复归为 OFF。
2. 当 X0.1 由 OFF→ON 时，计数器之当前值将执行上数（加一）的动作。
3. 当计数器 C0 计数到达设定值 5 时，C0 接点导通，C0 当前值 = 设定值=5。之后的 X0.1 触发信号 C0 完全不接受，C0 当前值保持在 5 处。



2.2.12 32 位计数器 HC

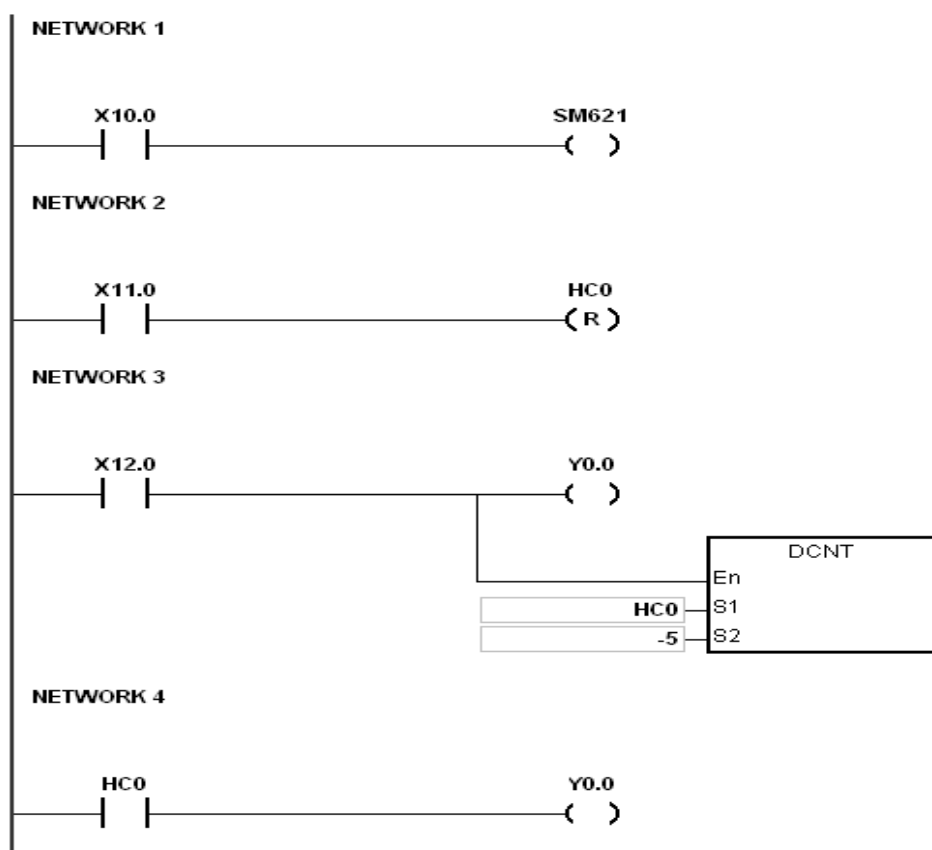
32 位计数器特点：

项目	32 位计数器
类型	一般型
编号	HC0~HC63
计数方向	上、下数
设定值	-2,147,483,648~+2,147,483,647
设定值的指定	常数或数据寄存器 D (指定 2 个)
当前值的变化	计数到达设定值后，仍继续计数
输出接点	上数到达设定值接点导通并保持 ON 下数到达设定值接点复归成 OFF
复归动作	RST 指令被执行时当前值归零，接点被复归成 OFF
接点动作	在扫描结束时，统一动作

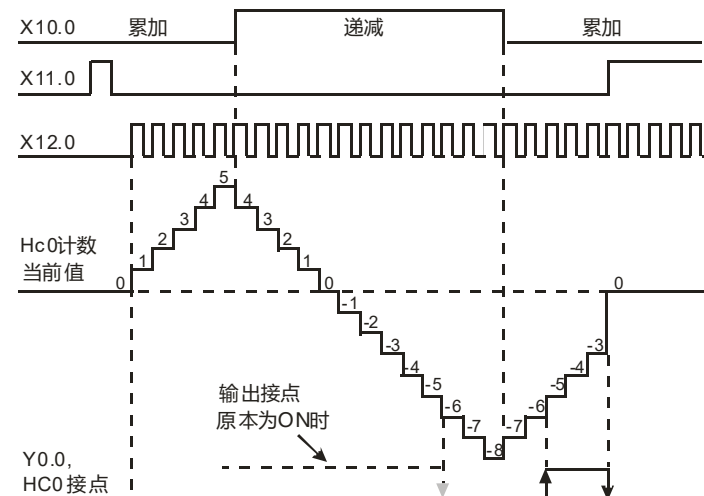
32 位一般用加减算计数器：

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 为加算，SM621=ON 时决定 HC0 为减算其余类推。
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。

范例：



1. X10.0 驱动 SM621 来决定 HC0 为加算或减算。
2. 当 X11.0 由 OFF→ON 时，RST 指令执行，HC0 之当前值被清为 0，且接点变为 OFF。
3. 当 X12.0 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。
4. 当计数器 HC0 的当前值从-6→-5 变化时，HC0 接点由 OFF→ON。当计数器 HC0 之当前值从-5→-6 变化时，HC0 接点由 ON→OFF。
5. 若使用 MOV 指令，将一个大於设定值的数值传送到 HC0 当前值寄存器时，在下次 X12.0 由 OFF→ON 时，HC0 计数器接点即变成 ON，同时当前值内容变成与设定值相同。



2.2.13 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之数值数据，亦可将两个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。亦可将四个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。也可用于与DIO之外的模块更新CR值之用，与模块更新CR值的D装置配置设定请参考ISPSOft手册的硬件组态说明。

寄存器依其性质可区分为下列二种：

1. 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 手册的硬件组态说明，但断电时仍会被清除为 0。
2. 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。

2.2.14 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），它的种类及功能如下所示。在编号前有“*”记号可参考SM/SR补充说明，像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之说明，可进一步了解系统信息。

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
SR0	运算错误之侦错号码	0	0	0	0	0	0	0	-	R	0
SR1 ↓ SR2	32位运算错误地址锁定	0	0	0	0	0	0	0	-	R	0
SR4	文法检查错误之侦错号码	0	0	0	0	0	0	0	-	R	0
SR5	32位文法检查错误的地址锁定	0	0	0	0	0	0	0	-	R	0
SR6		0	0	0	0	0	0	0	-	R	0
SR8	WDT定时器ON的Step地址	0	0	0	0	0	0	-	-	R	0
SR24 ↓ SR25	前一次扫描周期同步装置的数据量 (kbyte)	X	X	X	X	0	0	-	-	R	0
SR26 ↓ SR27	同步装置的历史最大数据量 (kbyte)	X	X	X	X	0	0	-	-	R	0
SR28 ↓ SR29	非周期性同步的运行时间 (ms)	X	X	X	X	0	0	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0-R S2	CP U5X0-E N	CP U5X1-R S2	CP U5X1-E N	CP U560-E N2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出 厂 设 定
SR30 ↓ SR31	周期性同步的运行时间 (ms)	X	X	X	X	O	0	-	-	R	0
*SR32	主电源模块状态	V1. 08	V1. 08	O	O	O	-	-	-	R	0
*SR33	冗余电源模块状态	V1. 08	V1. 08	O	O	O	-	-	-	R	0
*SR40	错误纪录 (Error Log) 的有效组数	O	O	O	O	O	-	-	-	R	0
*SR41	错误纪录的指针	O	O	O	O	O	-	-	-	R	0
*SR42	第一组的错误纪录之背版ID及插槽ID	O	O	O	O	O	-	-	-	R	0
*SR43	第一组的错误纪录之模块ID	O	O	O	O	O	-	-	-	R	0
*SR44	第一组的错误纪录之错误代码	O	O	O	O	O	-	-	-	R	0
*SR45	第一组的错误纪录之时间年及月	O	O	O	O	O	-	-	-	R	0
*SR46	第一组的错误纪录之时间日及时	O	O	O	O	O	-	-	-	R	0
*SR47	第一组的错误纪录之时间分及秒	O	O	O	O	O	-	-	-	R	0
*SR48	第二组的错误纪录之背版 ID 及插槽 ID	O	O	O	O	O	-	-	-	R	0
*SR49	第二组的错误纪录之模块 ID	O	O	O	O	O	-	-	-	R	0
*SR50	第二组的错误纪录之错误代码	O	O	O	O	O	-	-	-	R	0
*SR51	第二组的错误纪录之时间年及月	O	O	O	O	O	-	-	-	R	0
*SR52	第二组的错误纪录之时间日及时	O	O	O	O	O	-	-	-	R	0
*SR53	第二组的错误纪录之时间分及秒	O	O	O	O	O	-	-	-	R	0
*SR54	第三组的错误纪录之背版 ID 及插槽 ID	O	O	O	O	O	-	-	-	R	0
*SR55	第三组的错误纪录之模块 ID	O	O	O	O	O	-	-	-	R	0
*SR56	第三组的错误纪录之错误代码	O	O	O	O	O	-	-	-	R	0
*SR57	第三组的错误纪录之时间年及月	O	O	O	O	O	-	-	-	R	0
*SR58	第三组的错误纪录之时间日及时	O	O	O	O	O	-	-	-	R	0
*SR59	第三组的错误纪录之时间分及秒	O	O	O	O	O	-	-	-	R	0
*SR60	第四组的错误纪录之背版 ID 及插槽 ID	O	O	O	O	O	-	-	-	R	0
*SR61	第四组的错误纪录之模块 ID	O	O	O	O	O	-	-	-	R	0
*SR62	第四组的错误纪录之错误代码	O	O	O	O	O	-	-	-	R	0
*SR63	第四组的错误纪录之时间年及月	O	O	O	O	O	-	-	-	R	0
*SR64	第四组的错误纪录之时间日及时	O	O	O	O	O	-	-	-	R	0
*SR65	第四组的错误纪录之时间分及秒	O	O	O	O	O	-	-	-	R	0
*SR66	第四组的错误纪录之背版 ID 及插槽 ID	O	O	O	O	O	-	-	-	R	0
SR67	第五组的错误纪录之模块 ID	O	O	O	O	O	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SR68	第五组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
SR69	第五组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
SR70	第五组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
SR71	第五组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
SR72	第六组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR73	第六组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR74	第六组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR75	第六组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR76	第六组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR77	第六组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR78	第七组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR79	第七组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR80	第七组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR81	第七组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR82	第七组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR83	第七组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR84	第八组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR85	第八组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR86	第八组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR87	第八组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR88	第八组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR89	第八组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR90	第九组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR91	第九组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR92	第九组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR93	第九组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR94	第九组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR95	第九组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR96	第十组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR97	第十组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR98	第十组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR99	第十组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0-R S2	CP U5X0-E N	CP U5X1-R S2	CP U5X1-E N	CP U560-E N2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
*SR100	第十组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR101	第十组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR102	第十一组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR103	第十一组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR104	第十一组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR105	第十一组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR106	第十一组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR107	第十一组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR108	第十二组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR109	第十二组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR110	第十二组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR111	第十二组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR112	第十二组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR113	第十二组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR114	第十三组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR115	第十三组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR116	第十三组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR117	第十三组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR118	第十三组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR119	第十三组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR120	第十三组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR121	第十四组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR122	第十四组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR123	第十四组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR124	第十四组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR125	第十四组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SR126	第十五组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR127	第十五组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR128	第十五组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR129	第十五组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR130	第十五组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR131	第十五组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR132	第十六组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR133	第十六组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR134	第十六组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR135	第十六组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR136	第十六组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR137	第十六组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR138	第十七组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR139	第十七组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR140	第十七组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR141	第十七组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
SR142	第十七组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR143	第十七组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR144	第十八组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR145	第十八组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR146	第十八组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR147	第十八组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR148	第十八组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR149	第十八组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR150	第十九组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR151	第十九组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR152	第十九组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR153	第十九组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR154	第十九组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR155	第十九组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR156	第二十组的错误纪录之背版 ID 及插槽 ID	0	0	0	0	0	-	-	-	R	0
*SR157	第二十组的错误纪录之模块 ID	0	0	0	0	0	-	-	-	R	0
*SR158	第二十组的错误纪录之错误代码	0	0	0	0	0	-	-	-	R	0
*SR159	第二十组的错误纪录之时间年及月	0	0	0	0	0	-	-	-	R	0
*SR160	第二十组的错误纪录之时间日及时	0	0	0	0	0	-	-	-	R	0
*SR161	第二十组的错误纪录之时间分及秒	0	0	0	0	0	-	-	-	R	0
*SR201	PLC COM1通讯地址	0	0	0	0	0	-	-	-	R/W	1
*SR202	PLC COM2通讯地址	0	X	0	X	X	-	-	-	R/W	3
*SR209	COM1通讯协议	0	0	0	0	0	-	-	-	R/W	16# 0024
*SR210	COM1通讯超时时间 (单位: ms) , 设定为0则为不Time Out	0	0	0	0	0	3000 ms	-	-	R/W	3000 ms
*SR211	COM1命令重送次数	0	0	0	0	0	-	-	-	R/W	3
*SR212	COM2通讯协议	0	X	0	X	X	-	-	-	R/W	16# 0024
*SR213	COM2通讯超时时间 (单位: ms) , 设定为0表不Time Out	0	X	0	X	X	3000	-	-	R/W	3000 ms
*SR214	COM2命令重送次数	0	0	0	0	X	-	-	-	R/W	3
*SR215	COM1界面代号	0	0	0	0	0	-	-	-	R/W	0
*SR216	COM2界面代号	0	X	0	X	X	-	-	-	R/W	0
*SR220	万年历 (RTC) 年 (公元) 00~99	0	0	0	0	0	-	-	-	R	0
*SR221	万年历 (RTC) 月01~12	0	0	0	0	0	-	-	-	R	1
*SR222	万年历 (RTC) 日01~31	0	0	0	0	0	-	-	-	R	1
*SR223	万年历 (RTC) 时00~23	0	0	0	0	0	-	-	-	R	0
*SR224	万年历 (RTC) 分00~59	0	0	0	0	0	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR225	万年历 (RTC) 秒00~59	0	0	0	0	0	-	-	-	R	0
*SR226	万年历 (RTC) 星期1~7	0	0	0	0	0	-	-	-	R	1
*SR227	储存下载动作的有效组数 (最多纪录 20次)	0	0	0	0	0	-	-	-	R	0
*SR228	最新一次下载动作的纪录指针。	0	0	0	0	0	-	-	-	R	0
*SR229	下载纪录第1组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR230	下载纪录第1组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR231	下载纪录第1组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR232	下载纪录第1组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR233	下载纪录第2组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR234	下载纪录第2组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR235	下载纪录第2组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR236	下载纪录第2组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR237	下载纪录第3组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR238	下载纪录第3组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR239	下载纪录第3组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR240	下载纪录第3组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR241	下载纪录第4组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR242	下载纪录第4组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR243	下载纪录第4组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR244	下载纪录第4组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR245	下载纪录第5组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR246	下载纪录第5组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR247	下载纪录第5组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR248	下载纪录第5组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR249	下载纪录第6组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR250	下载纪录第6组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR251	下载纪录第6组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR252	下载纪录第6组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR253	下载纪录第7组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR254	下载纪录第7组·纪录时间年月	0	0	0	0	0	-	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
*SR255	下载纪录第7组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR256	下载纪录第7组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR257	下载纪录第8组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR258	下载纪录第8组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR259	下载纪录第8组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR260	下载纪录第8组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR261	下载纪录第9组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR262	下载纪录第9组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR263	下载纪录第9组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR264	下载纪录第9组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR265	下载纪录第10组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR266	下载纪录第10组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR267	下载纪录第10组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR268	下载纪录第10组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR269	下载纪录第11组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR270	下载纪录第11组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR271	下载纪录第11组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR272	下载纪录第11组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR273	下载纪录第12组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR274	下载纪录第12组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR275	下载纪录第12组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR276	下载纪录第12组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR277	下载纪录第13组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR278	下载纪录第13组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR279	下载纪录第13组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR280	下载纪录第13组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR281	下载纪录第14组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR282	下载纪录第14组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR283	下载纪录第14组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR284	下载纪录第14组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR285	下载纪录第15组·纪录动作编号	0	0	0	0	0	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
*SR286	下载纪录第15组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR287	下载纪录第15组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR288	下载纪录第15组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR289	下载纪录第16组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR290	下载纪录第16组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR291	下载纪录第16组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR292	下载纪录第16组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR293	下载纪录第17组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR294	下载纪录第17组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR295	下载纪录第17组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR296	下载纪录第17组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR297	下载纪录第18组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR298	下载纪录第18组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR299	下载纪录第18组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR300	下载纪录第18组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR301	下载纪录第19组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR302	下载纪录第19组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR303	下载纪录第19组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR304	下载纪录第19组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR305	下载纪录第20组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR306	下载纪录第20组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR307	下载纪录第20组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR308	下载纪录第20组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR309	纪录PLC状态变更的总数(最大20组)	0	0	0	0	0	-	-	-	R	0
*SR310	纪录PLC状态变更最新一次的指针	0	0	0	0	0	-	-	-	R	0
*SR311	PLC状态变更纪录第1组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR312	PLC状态变更纪录第1组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR313	PLC状态变更纪录第1组·纪录时间日時	0	0	0	0	0	-	-	-	R	0
*SR314	PLC状态变更纪录第1组·纪录时间分	0	0	0	0	0	-	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0-RS2	CP U5X0-EN	CP U5X1-RS2	CP U5X1-EN	CP U560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	秒										
*SR315	PLC状态变更纪录第2组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR316	PLC状态变更纪录第2组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR317	PLC状态变更纪录第2组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR318	PLC状态变更纪录第2组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR319	PLC状态变更纪录第3组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR320	PLC状态变更纪录第3组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR321	PLC状态变更纪录第3组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR322	PLC状态变更纪录第3组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR323	PLC状态变更纪录第4组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR324	PLC状态变更纪录第4组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR325	PLC状态变更纪录第4组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR326	PLC状态变更纪录第4组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR327	PLC状态变更纪录第5组·纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR328	PLC状态变更纪录第5组·纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR329	PLC状态变更纪录第5组·纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR330	PLC状态变更纪录第5组·纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR331	PLC状态变更纪录第6组·纪录动作编号	0	0	0	0	0	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	号										
*SR332	PLC状态变更纪录第6组，纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR333	PLC状态变更纪录第6组，纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR334	PLC状态变更纪录第6组，纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR335	PLC状态变更纪录第7组，纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR336	PLC状态变更纪录第7组，纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR337	PLC状态变更纪录第7组，纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR338	PLC状态变更纪录第7组，纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR339	PLC状态变更纪录第8组，纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR340	PLC状态变更纪录第8组，纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR341	PLC状态变更纪录第8组，纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR342	PLC状态变更纪录第8组，纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR343	PLC状态变更纪录第9组，纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR344	PLC状态变更纪录第9组，纪录时间年月	0	0	0	0	0	-	-	-	R	0
*SR345	PLC状态变更纪录第9组，纪录时间日时	0	0	0	0	0	-	-	-	R	0
*SR346	PLC状态变更纪录第9组，纪录时间分秒	0	0	0	0	0	-	-	-	R	0
*SR347	PLC状态变更纪录第10组，纪录动作编号	0	0	0	0	0	-	-	-	R	0
*SR348	PLC状态变更纪录第10组，纪录时间	0	0	0	0	0	-	-	-	R	0

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	年月										
*SR349	PLC状态变更纪录第10组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR350	PLC状态变更纪录第10组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR351	PLC状态变更纪录第11组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR352	PLC状态变更纪录第11组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR353	PLC状态变更纪录第11组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR354	PLC状态变更纪录第11组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR355	PLC状态变更纪录第12组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR356	PLC状态变更纪录第12组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR357	PLC状态变更纪录第12组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR358	PLC状态变更纪录第12组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR359	PLC状态变更纪录第13组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR360	PLC状态变更纪录第13组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR361	PLC状态变更纪录第13组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR362	PLC状态变更纪录第13组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR363	PLC状态变更纪录第14组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR364	PLC状态变更纪录第14组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR365	PLC状态变更纪录第14组·纪录时间	0	0	0	0	0	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	日時										
*SR366	PLC状态变更纪录第14组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR367	PLC状态变更纪录第15组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR368	PLC状态变更纪录第15组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR369	PLC状态变更纪录第15组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR370	PLC状态变更纪录第15组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR371	PLC状态变更纪录第16组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR372	PLC状态变更纪录第16组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR373	PLC状态变更纪录第16组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR374	PLC状态变更纪录第16组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR375	PLC状态变更纪录第17组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR376	PLC状态变更纪录第17组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR377	PLC状态变更纪录第17组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR378	PLC状态变更纪录第17组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR379	PLC状态变更纪录第18组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR380	PLC状态变更纪录第18组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR381	PLC状态变更纪录第18组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR382	PLC状态变更纪录第18组·纪录时间	0	0	0	0	0	-	-	-	R	0

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	分秒										
*SR383	PLC状态变更纪录第19组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR384	PLC状态变更纪录第19组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR385	PLC状态变更纪录第19组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR386	PLC状态变更纪录第19组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR387	PLC状态变更纪录第20组·纪录动作 编号	0	0	0	0	0	-	-	-	R	0
*SR388	PLC状态变更纪录第20组·纪录时间 年月	0	0	0	0	0	-	-	-	R	0
*SR389	PLC状态变更纪录第20组·纪录时间 日時	0	0	0	0	0	-	-	-	R	0
*SR390	PLC状态变更纪录第20组·纪录时间 分秒	0	0	0	0	0	-	-	-	R	0
*SR391	万年历 (RTC) 年 (公元) 00~99	0	0	0	0	0	-	-	-	R	0
*SR392	万年历 (RTC) 月01~12	0	0	0	0	0	-	-	-	R	1
*SR393	万年历 (RTC) 日01~31	0	0	0	0	0	-	-	-	R	1
*SR394	万年历 (RTC) 时00~23	0	0	0	0	0	-	-	-	R	0
*SR395	万年历 (RTC) 分00~59	0	0	0	0	0	-	-	-	R	0
*SR396	万年历 (RTC) 秒00~59	0	0	0	0	0	-	-	-	R	0
*SR397	万年历 (RTC) 星期1~7	0	0	0	0	0	-	-	-	R	1
SR402	RUN后·每100微秒加1·重复计数由 0~32767再由-32768~0	X	X	V 1. 01	V 1. 01	0	0	-	-	R	0
SR404	RUN后·每1毫秒加1·重复计数由 0~32767再由-32768~0	X	X	V 1. 01	V 1. 01	0	0	-	-	R	0
SR407	RUN后·每秒加1·重复计数由 0~32767再由-32768~0	0	0	0	0	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	0	0	0	0	0	-	R/W	0
*SR409	储存数值n于2n秒时钟·设定可为1到 32767	0	0	0	0	0	-	-	-	R/W	30

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SR410	储存数值n于2n毫秒时钟	0	0	0	0	0	-	-	-	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)	0	0	0	0	0	0	-	-	R	0
SR412	SR412:储存微秒部份。(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	0	0	0	0	0	-	-	R	0
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	0	0	0	0	0	-	-	R	0
SR414		0	0	0	0	0	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	0	0	0	0	0	-	-	R	0
SR416		0	0	0	0	0	0	-	-	R	0
*SR440 ↓ *SR442	主机MAC值	0	0	0	0	0	MAC值	-	-	R	MAC值
*SR443 ↓ *SR451	主机产品序号	0	0	0	0	0	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	0	0	0	0	0	-	-	-	R	0
SR621	RS指令中断字符(COM1)	0	0	0	0	0	-	-	-	R/W	0
SR622	RS指令中断字符(COM2)	0	X	0	X	X	-	-	-	R/W	0
SR623	位0~15:中断程序I0~I15条件·由IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR624	位0~15:中断程序I16~I31条件·由IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR625	位0~15:中断程序I32~I47条件·由IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR626	位0~15:中断程序I48~I63条件·由IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR627	位0~15:中断程序I64~I79条件·由IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF

特殊 寄存器 SR	功能说明	CP U5X0-R S2	CP U5X0-E N	CP U5X1-R S2	CP U5X1-E N	CP U560-E N2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
SR628	位0~15：中断程序I80~I95条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR629	位0~15：中断程序I96~I111条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR630	位0~15：中断程序I112~I127条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR631	位0~15：中断程序I128~I143条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR632	位0~15：中断程序I144~I159条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR633	位0~15：中断程序I160~I175条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR634	位0~15：中断程序I176~I191条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR635	位0~15：中断程序I192~I207条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR636	位0~15：中断程序I208~I213条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR637	位0~15：中断程序I214~I229条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
SR638	位0~15：中断程序I230~I255条件，由 IMASK指令设定	0	0	0	0	0	FFFF	-	-	R	FFFF
*SR655 ↓ SR662	纪录背板1的I/O表映像错误或IO模块 发生错误 ↓ 纪录背板8的I/O表映像错误或IO模块 发生错误	0	0	0	0	0	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	0	0	0	0	0	0	-	-	R	0
*SR675 ↓ SR682	纪录背板2-插槽0的I/O表发生的映射 错误码 ↓	0	0	0	0	0	0	-	-	R	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
	纪录背板2-插槽7的I/O表发生的映射错误码										
*SR683 ↓ SR690	纪录背板3-插槽0的I/O表发生的映射错误码 ↓ 纪录背板3-插槽7的I/O表发生的映射错误码	0	0	0	0	0	0	-	-	R	0
*SR691 ↓ SR698	纪录背板4-插槽0的I/O表发生的映射错误码 ↓ 纪录背板4-插槽7的I/O表发生的映射错误码	0	0	0	0	0	0	-	-	R	0
*SR699 ↓ SR706	纪录背板5-插槽0的I/O表发生的映射错误码 ↓ 纪录背板5-插槽7的I/O表发生的映射错误码	0	0	0	0	0	0	-	-	R	0
*SR707 ↓ SR714	纪录背板6-插槽0的I/O表发生的映射错误码 ↓ 纪录背板6-插槽7的I/O表发生的映射错误码	0	0	0	0	0	0	-	-	R	0
*SR715 ↓ SR722	纪录背板7-插槽0的I/O表发生的映射错误码 ↓ 纪录背板7-插槽7的I/O表发生的映射错误码	0	0	0	0	0	0	-	-	R	0
*SR723 ↓ SR730	纪录背板8-插槽0的I/O表发生的映射错误码 ↓ 纪录背板8-插槽7的I/O表发生的映射错误码	0	0	0	0	0	0	-	-	R	0
*SR731	主电源模块外部24V低电压检测	V 1. 08	V 1. 08	0	0	0	0	-	-	R	0
*SR732	冗余电源模块外部24V低电压检测	V 1. 08	V 1. 08	0	0	0	0	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
*SR1000	以太网IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	C0A8
*SR1001	以太网IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0101
*SR1002	以太网网络屏蔽地址上位字符组	X	O	X	O	O	-	-	-	R/W	FFFF
*SR1003	以太网网络屏蔽地址下位字符组	X	O	X	O	O	-	-	-	R/W	0000
*SR1004	以太网网关地址上位字符组	X	O	X	O	O	-	-	-	R/W	C0A8
*SR1005	以太网网关地址下位字符组	X	O	X	O	O	-	-	-	R/W	0101
*SR1006	TCP保持连接时间	X	O	X	O	O	-	-	-	R/W	60
SR1007	以太网传输速度	X	O	X	O	O	0	-	-	R	0
SR1008	以太网传输模式	X	O	X	O	O	0	-	-	R	0
SR1100	输入报文计数器上位字符组	X	O	X	O	O	0	-	-	R	0
SR1101	输入报文计数器下位字符组	X	O	X	O	O	0	-	-	R	0
SR1102	输入计数器上位字符组	X	O	X	O	O	0	-	-	R	0
SR1103	输入计数器下位字符组	X	O	X	O	O	0	-	-	R	0
SR1104	输出报文计数上位字符组	X	O	X	O	O	0	-	-	R	0
SR1105	输出报文计数下位字符组	X	O	X	O	O	0	-	-	R	0
SR1106	输出计数器上位字符组	X	O	X	O	O	0	-	-	R	0
SR1107	输出计数器下位字符组	X	O	X	O	O	0	-	-	R	0
*SR1116	Email计数器	X	O	X	O	O	0	-	-	R	0
*SR1117	Email错误计数器	X	O	X	O	O	0	-	-	R	0
*SR1118	TCP Socket 1--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1119	TCP Socket 1--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1120	TCP Socket 1--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1121	TCP Socket 1--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1122	TCP Socket 1--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1123	TCP Socket 1--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1124	TCP Socket 1--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1125	TCP Socket 1--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1126	TCP Socket 1--接收数据地址上位字	X	O	X	O	O	-	-	-	R/W	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
	符组										
*SR1127	TCP Socket 1--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1128	TCP Socket 1--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1129	TCP Socket 1--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1130	TCP Socket 1--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1131	TCP Socket 2--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1132	TCP Socket 2--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1133	TCP Socket 2--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1134	TCP Socket 2--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1135	TCP Socket 2--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1136	TCP Socket 2--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1137	TCP Socket 2--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1138	TCP Socket 2--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1139	TCP Socket 2--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1140	TCP Socket 2--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1141	TCP Socket 2--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1142	TCP Socket 2--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1143	TCP Socket 2--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1144	TCP Socket 3--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1145	TCP Socket 3--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1146	TCP Socket 3--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1147	TCP Socket 3--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1148	TCP Socket 3--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1149	TCP Socket 3--传送数据地址上位字	X	O	X	O	O	-	-	-	R/W	0

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
	符组										
*SR1150	TCP Socket 3--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1151	TCP Socket 3--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1152	TCP Socket 3--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1153	TCP Socket 3--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1154	TCP Socket 3--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1155	TCP Socket 3--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1156	TCP Socket 3--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1157	TCP Socket 4--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1158	TCP Socket 4--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1159	TCP Socket 4--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1160	TCP Socket 4--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1161	TCP Socket 4--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1162	TCP Socket 4--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1163	TCP Socket 4--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1164	TCP Socket 4--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1165	TCP Socket 4--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1166	TCP Socket 4--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1167	TCP Socket 4--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1168	TCP Socket 4--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1169	TCP Socket 4--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1170	TCP Socket 5--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1171	TCP Socket 5--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SR1172	TCP Socket 5--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1173	TCP Socket 5--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1174	TCP Socket 5--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1175	TCP Socket 5--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1176	TCP Socket 5--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1177	TCP Socket 5--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1178	TCP Socket 5--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1179	TCP Socket 5--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1180	TCP Socket 5--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1181	TCP Socket 5--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1182	TCP Socket 5--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1183	TCP Socket 6--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1184	TCP Socket 6--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1185	TCP Socket 6--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1186	TCP Socket 6--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1187	TCP Socket 6--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1188	TCP Socket 6--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1189	TCP Socket 6--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1190	TCP Socket 6--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1191	TCP Socket 6--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1192	TCP Socket 6--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1193	TCP Socket 6--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
*SR1194	TCP Socket 6--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1195	TCP Socket 6--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1196	TCP Socket 7--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1197	TCP Socket 7--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1198	TCP Socket 7--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1199	TCP Socket 7--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1200	TCP Socket 7--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1201	TCP Socket 7--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1202	TCP Socket 7--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1203	TCP Socket 7--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1204	TCP Socket 7--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1205	TCP Socket 7--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1206	TCP Socket 7--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1207	TCP Socket 7--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1208	TCP Socket 7--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1209	TCP Socket 8--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1210	TCP Socket 8--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1211	TCP Socket 8--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1212	TCP Socket 8--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1213	TCP Socket 8--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1214	TCP Socket 8--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1215	TCP Socket 8--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1216	TCP Socket 8--接收数据长度	X	O	X	O	O	-	-	-	R/W	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	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符组	X	O	X	O	O	-	-	-	R/W	0
*SR1218	TCP Socket 8--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1219	TCP Socket 8--保持连接时间	X	O	X	O	O	-	-	-	R/W	1000
*SR1220	TCP Socket 8--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1221	TCP Socket 8--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1222	UDP Socket 1--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1223	UDP Socket 1--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1224	UDP Socket 1--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1225	UDP Socket 1--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1226	UDP Socket 1--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1227	UDP Socket 1--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1228	UDP Socket 1--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1229	UDP Socket 1--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1230	UDP Socket 1--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1231	UDP Socket 1--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1232	UDP Socket 1--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1233	UDP Socket 1--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1234	UDP Socket 2--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1235	UDP Socket 2--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1236	UDP Socket 2--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1237	UDP Socket 2--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1238	UDP Socket 2--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1239	UDP Socket 2--传送数据地址上位字	X	O	X	O	O	-	-	-	R/W	0

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
	符组										
*SR1240	UDP Socket 2--传送数据地址下位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1241	UDP Socket 2--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1242	UDP Socket 2--接收数据地址上位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1243	UDP Socket 2--接收数据地址下位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1244	UDP Socket 2--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1245	UDP Socket 2--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1246	UDP Socket 3--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1247	UDP Socket 3--远程IP地址上位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1248	UDP Socket 3--远程IP地址下位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1249	UDP Socket 3--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1250	UDP Socket 3--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1251	UDP Socket 3--传送数据地址上位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1252	UDP Socket 3--传送数据地址下位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1253	UDP Socket 3--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1254	UDP Socket 3--接收数据地址上位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1255	UDP Socket 3--接收数据地址下位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1256	UDP Socket 3--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1257	UDP Socket 3--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1258	UDP Socket 4--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1259	UDP Socket 4--远程IP地址上位字 符组	X	O	X	O	O	-	-	-	R/W	0
*SR1260	UDP Socket 4--远程IP地址下位字 符组	X	O	X	O	O	-	-	-	R/W	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SR1261	UDP Socket 4--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1262	UDP Socket 4--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1263	UDP Socket 4--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1264	UDP Socket 4--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1265	UDP Socket 4--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1266	UDP Socket 4--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1267	UDP Socket 4--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1268	UDP Socket 4--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1269	UDP Socket 4--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1270	UDP Socket 5--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1271	UDP Socket 5--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1272	UDP Socket 5--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1273	UDP Socket 5--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1274	UDP Socket 5--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1275	UDP Socket 5--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1276	UDP Socket 5--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1277	UDP Socket 5--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1278	UDP Socket 5--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1279	UDP Socket 5--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1280	UDP Socket 5--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1281	UDP Socket 5--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1282	UDP Socket 6--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1283	UDP Socket 6--远程IP地址上位字符	X	O	X	O	O	-	-	-	R/W	0

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
	组										
*SR1284	UDP Socket 6--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1285	UDP Socket 6--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1286	UDP Socket 6--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1287	UDP Socket 6--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1288	UDP Socket 6--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1289	UDP Socket 6--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1290	UDP Socket 6--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1291	UDP Socket 6--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1292	UDP Socket 6--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1293	UDP Socket 6--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1294	UDP Socket 7--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1295	UDP Socket 7--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1296	UDP Socket 7--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1297	UDP Socket 7--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1298	UDP Socket 7--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1299	UDP Socket 7--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1300	UDP Socket 7--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1301	UDP Socket 7--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1302	UDP Socket 7--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1303	UDP Socket 7--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1304	UDP Socket 7--接收数据计数器	X	O	X	O	O	0	-	-	R	0

2

特殊寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂设定
*SR1305	UDP Socket 7--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1306	UDP Socket 8--本地通讯端口	X	O	X	O	O	-	-	-	R/W	0
*SR1307	UDP Socket 8--远程IP地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1308	UDP Socket 8--远程IP地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1309	UDP Socket 8--远程通信端口	X	O	X	O	O	-	-	-	R/W	0
*SR1310	UDP Socket 8--传送数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1311	UDP Socket 8--传送数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1312	UDP Socket 8--传送数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1313	UDP Socket 8--接收数据长度	X	O	X	O	O	-	-	-	R/W	0
*SR1314	UDP Socket 8--接收数据地址上位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1315	UDP Socket 8--接收数据地址下位字符组	X	O	X	O	O	-	-	-	R/W	0
*SR1316	UDP Socket 8--接收数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1317	UDP Socket 8--传送数据计数器	X	O	X	O	O	0	-	-	R	0
*SR1318	Socket输入计数器	X	O	X	O	O	0	-	-	R	0
*SR1319	Socket输出计数器	X	O	X	O	O	0	-	-	R	0
*SR1320	Socket错误计数器	X	O	X	O	O	0	-	-	R	0
*SR1329	主机背板ID	O	O	O	O	O	-	-	-	R/W	0
*SR1330	主机插槽ID	O	O	O	O	O	-	-	-	R/W	0
*SR1331	RTU号码	O	O	O	O	O	-	-	-	R/W	0
*SR1332	扩展背板ID	O	O	O	O	O	-	-	-	R/W	0
*SR1333	扩展插槽ID	O	O	O	O	O	-	-	-	R/W	0
*SR1334	端口号	O	O	O	O	O	-	-	-	R/W	0
*SR1335	PLC Link周期时间	O	O	X	X	O	-	-	-	R	0
	COM1-MODBUS周期时间	X	X	O	O	O	-	-	-	R	0
*SR1336	PLC Link联机从站总数	O	O	X	X	O	-	-	-	R	0
	COM1-MODBUS联机从站总数	X	X	O	O	O	-	-	-	R	0
*SR1337	PLC Link数据交换时间	O	O	X	X	O	-	-	-	R/W	0

特殊 寄存器 SR	功能说明	CP U5X0-RS2	CP U5X0-EN	CP U5X1-RS2	CP U5X1-EN	CP U560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属 性	出 厂 设 定
	COM1-MODBUS数据交换次数	X	X	O	O	O	-	-	-	R/W	0
*SR1338	PLC Link使用者定义限制时间	O	O	X	X	X	-	-	-	R/W	0
*SR1339	PLC Link传送cmd间隔时间	O	O	X	X	X	-	-	-	R/W	1
	COM1-MODBUS传送cmd间隔时间	X	X	O	O	O	-	-	-	R/W	1
*SR1340 ↓ SR1371	PLC Link从站1读取装置类别 ↓ (0 : 寄存器 · 1 : 输出线圈 · 其他 : 不支持)	O	O	X	X	X	-	-	-	R/W	0
	PLC Link从站32读取装置类别										
	COM1-MODBUS从站1读取装置类别 ↓ (0 : 寄存器 · 1 : 输出线圈 · 其他 : 不支持) COM1-MODBUS读取装置类别32	X	X	O	O	O	-	-	-	R/W	0
*SR1372 ↓ SR1403	PLC Link从站1写入装置类别 ↓ (0 : 寄存器 · 1 : 输出线圈 · 其他 : 不支持)	O	O	X	X	X	-	-	-	R/W	0
	PLC Link从站32写入装置类别										
	COM1-MODBUS从站1写入装置类别 ↓ (0 : 寄存器 · 1 : 输出线圈 · 其他 : 不支持) COM1-MODBUS从站32写入装置类 别	X	X	O	O	O	-	-	-	R/W	0
*SR1404 ↓ SR1467	PLC Link从站1读取数据主站存放起 始地址 (SR1404 · SR1405) ↓	O	O	X	X	X	-	-	-	R/W	0
	PLC Link从站32读取数据主站存放起 始地址 (SR1466 · SR1467)										
	COM1-MODBUS从站1读取数据主站 存放起始地址 (SR1404 · SR1405) ↓ COM1-MODBUS从站32读取数据主 站存放起始地址 (SR1466 · SR1467)	X	X	O	O	O	-	-	-	R/W	0
*SR1468 ↓ SR1531	PLC Link从站1写入数据主站存放起 始地址 (SR1468 · SR1469) ↓ PLC Link从站32写入数据主站存放起	O	O	X	X	X	-	-	-	R/W	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	始地址 (SR1530 · SR1531)										
	COM1-MODBUS从站1写入数据主站 存放起始地址 (SR1468 · SR11469) ↓ COM1-MODBUS从站32写入数据主 站存放起始地址 (SR1530 · SR1531)	X	X	O	O	O	-	-	-	R/W	0
*SR1532 ↓ SR1595	PLC Link读取从站1通讯起始地址设 定 (SR1532 · SR1533) ↓ PLC Link读取从站32通讯起始地址设 定 (SR1594 · SR1595)	O	O	X	X	X	-	-	-	R/W	0
	COM1-MODBUS读取从站1通讯起始 地址设定 (SR1532 · SR1533) ↓ COM1-MODBUS读取从站32通讯起 始地址设定数据 (SR1594 · SR1595)	X	X	O	O	O	-	-	-	R/W	0
*SR1596 ↓ SR1659	PLC Link写入从站1通讯起始地址设 定 (SR1596 · SR1597) ↓ PLC Link写入从站32通讯起始地址设 定 (SR1658 · SR1659)	O	O	X	X	X	-	-	-	R/W	0
	COM1-MODBUS写入从站1通讯起始 地址设定 (SR1596 · SR1597) ↓ COM1-MODBUS写入从站32通讯起 始地址设定 (SR1658 · SR1659)	X	X	O	O	O	-	-	-	R/W	0
*SR1660 ↓ SR1691	PLC Link从站1读取数据长度 ↓ PLC Link从站32读取数据长度	O	O	X	X	X	-	-	-	R/W	0
	COM1-MODBUS从站1读取数据长度 ↓ COM1-MODBUS从站32读取数据长 度	X	X	O	O	O	-	-	-	R/W	0
*SR1692 ↓ SR1723	PLC Link从站1写入数据长度 ↓ PLC Link从站32写入数据长度	O	O	X	X	X	-	-	-	R/W	0

特殊 寄存器 SR	功能说明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	COM1-MODBUS从站1写入数据长度 ↓ COM1-MODBUS从站32写入数据长度	X	X	O	O	O	-	-	-	R/W	0
*SR1724 ↓ SR1755	PLC Link从站1类别 ↓ PLC Link从站32类别	O	O	X	X	X	-	-	-	R/W	0
*SR1756 ↓ SR1787	PLC Link从站1 ID ↓ PLC Link从站32 ID	O	O	X	X	X	-	-	-	R/W	1 ↓ 32
	COM1-MODBUS从站1 ID ↓ COM1-MODBUS从站32 ID	X	X	O	O	O	-	-	-	R/W	1 ↓ 32
*SR1792 ↓ SR1823	Ether Link区块1 IP地址 (SR1792 · SR1793) ↓ Ether Link区块16 IP地址 (SR1822 · SR1823)	O	O	X	X	X	-	-	-	R	0
SR1824 ↓ SR1855	Ether Link区块17 IP地址 (SR1824 · SR1825) ↓ Ether Link区块32 IP地址 (SR1854 · SR1855)	O ₁ []	O ₁ [*]	X	X	X	-	-	-	R	0
SR1856 ↓ SR1919	Ether Link区块33 IP地址 (SR1856 · SR1857) ↓ Ether Link区块64 IP地址 (SR1918 · SR1919)	O ₂ []	O ₂ [*]	X	X	X	-	-	-	R	0
SR1920 ↓ SR2047	Ether Link区块65 IP地址 (SR1920 · SR1921) ↓ Ether Link区块128IP地址 (SR2046 · SR2047)	O ₃ []	O ₃ [*]	X	X	X	-	-	-	R	0
SR2048 ↓ SR2079	EtherNet/IP I/O Connection 1 错误代 码 ↓	X	X	O	O	O	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
	EtherNet/IP I/O Connection 32 错误 代码										
SR2080 ↓ SR2111	EtherNet/IP I/O Connection 33 错误 代码 ↓ EtherNet/IP I/O Connection 64 错误 代码	X	X	O	O *6	O	-	-	-	R	0
SR2112 ↓ SR2175	EtherNet/IP I/O Connection 65 错误 代码 ↓ EtherNet/IP I/O Connection 128 错误 代码	X	X	O	O *4	O	-	-	-	R	0
SR2176 ↓ SR2303	EtherNet/IP I/O Connection 129 错误 代码 ↓ EtherNet/IP I/O Connection 256 错误 代码	X	X	O	O *5	O	-	-	-	R	0
SR2304 ↓ SR2335	EtherNet/IP I/O connection 1 运行状 态 ↓ EtherNet/IP I/O connection 32 运行 状态	X	X	O	O	O	-	-	-	R	0
SR2336 ↓ SR2367	EtherNet/IP I/O connection 33 运行 状态 ↓ EtherNet/IP I/O connection 64 运行 状态	X	X	O	O *6	O	-	-	-	R	0
SR2368 ↓ SR2431	EtherNet/IP I/O connection 65 运行 状态 ↓ EtherNet/IP I/O connection 128 运行 状态	X	X	O	O *4	O	-	-	-	R	0
SR2432 ↓ SR2559	EtherNet/IP I/O connection 129 运行 状态 ↓ EtherNet/IP I/O connection 256 运行 状态	X	X	O	O *5	O	-	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0-R S2	CP U5X0-E N	CP U5X1-R S2	CP U5X1-E N	CP U560-E N2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出 厂 设 定
SR2560	第1组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2561	第1组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2562	第1组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2563 ↓ SR2564	第1组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2565	第2组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2566	第2组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2567	第2组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2568 ↓ SR2569	第2组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2570	第3组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2571	第3组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2572	第3组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2573 ↓ SR2574	第3组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2575	第4组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2576	第4组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2577	第4组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2578 ↓ SR2579	第4组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2580	第5组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SR2581	第5组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2582	第5组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2583 ↓ SR2584	第5组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2585	第6组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2586	第6组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2587	第6组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2588 ↓ SR2589	第6组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2590	第7组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2591	第7组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2592	第7组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2593 ↓ SR2594	第7组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2595	第8组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2596	第8组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2597	第8组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2598 ↓ SR2599	第8组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2600	第9组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2601	第9组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0-R S2	CP U5X0-E N	CP U5X1-R S2	CP U5X1-E N	CP U560-E N2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出 厂 设 定
SR2602	第9组的错误纪录之RTU背版ID及插槽ID	X	X	O	O	O	-	-	-	R	0
SR2603 ↓ SR2604	第9组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2605	第10组的错误纪录之RTU IP或ID上位字符组	X	X	O	O	O	-	-	-	R	0
SR2606	第10组的错误纪录之RTU IP或ID下位字符组	X	X	O	O	O	-	-	-	R	0
SR2607	第10组的错误纪录之RTU背版ID及插槽ID	X	X	O	O	O	-	-	-	R	0
SR2608 ↓ SR2609	第10组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2610	第11组的错误纪录之RTU IP或ID上位字符组	X	X	O	O	O	-	-	-	R	0
SR2611	第11组的错误纪录之RTU IP或ID下位字符组	X	X	O	O	O	-	-	-	R	0
SR2612	第11组的错误纪录之RTU背版ID及插槽ID	X	X	O	O	O	-	-	-	R	0
SR2613 ↓ SR2614	第11组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2615	第12组的错误纪录之RTU IP或ID上位字符组	X	X	O	O	O	-	-	-	R	0
SR2616	第12组的错误纪录之RTU IP或ID下位字符组	X	X	O	O	O	-	-	-	R	0
SR2617	第12组的错误纪录之RTU背版ID及插槽ID	X	X	O	O	O	-	-	-	R	0
SR2618 ↓ SR2619	第12组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2620	第13组的错误纪录之RTU IP或ID上位字符组	X	X	O	O	O	-	-	-	R	0
SR2621	第13组的错误纪录之RTU IP或ID下位字符组	X	X	O	O	O	-	-	-	R	0
SR2622	第13组的错误纪录之RTU背版ID及插槽ID	X	X	O	O	O	-	-	-	R	0

2

特殊 寄存器 SR	功能说明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出厂 设定
SR2623 ↓ SR2624	第13组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2625	第14组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2626	第14组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2627	第14组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2628 ↓ SR2629	第14组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2630	第15组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2631	第15组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2632	第15组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2633 ↓ SR2634	第15组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2635	第16组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2636	第16组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2637	第16组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2638 ↓ SR2639	第16组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2640	第17组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2641	第17组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2642	第17组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2643 ↓ SR2644	第17组的运算错误地址	X	X	O	O	O	-	-	-	R	0

特殊 寄存器 SR	功能说明	CP U5X0-RS2	CP U5X0-EN	CP U5X1-RS2	CP U5X1-EN	CP U560-EN2	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	属性	出 厂 设 定
SR2645	第18组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2646	第18组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2647	第18组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2648 ↓ SR2649	第18组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2650	第19组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2651	第19组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2652	第19组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2653 ↓ SR2654	第19组的运算错误地址	X	X	O	O	O	-	-	-	R	0
SR2655	第20组的错误纪录之RTU IP或ID上位 字符组	X	X	O	O	O	-	-	-	R	0
SR2656	第20组的错误纪录之RTU IP或ID下位 字符组	X	X	O	O	O	-	-	-	R	0
SR2657	第20组的错误纪录之RTU背版ID及插 槽ID	X	X	O	O	O	-	-	-	R	0
SR2658 ↓ SR2659	第20组的运算错误地址	X	X	O	O	O	-	-	-	R	0

注：*之 SR 请参考 SM/SR 补充说明

*1：仅支持机种 AHCPU530-EN、AHCPU520-EN、AHCPU510-EN、AHCPU530-RS2、
AHCPU520-RS2、AHCPU510-RS2。

*2：仅支持机种 AHCPU530-EN、AHCPU520-EN、AHCPU530-RS2 及 AHCPU520-RS2。

*3：仅支持机种 AHCPU530-EN 及 AHCPU530-RS2。

*4：仅支持机种 AHCPU531-EN 及 AHCPU521-EN。

*5：仅支持机种 AHCPU531-EN。

*6：仅支持机种 AHCPU531-EN、AHCPU521-EN 及 AHCPU511-EN。

2.2.15 特殊数据寄存器 SR 的更新时间

适用机种	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
	SR24~SR31	同步动作完成时更新
	SR32	主电源模块发生异常时更新
	SR33	冗余电源模块发生异常时更新
	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
	SR402	每100微秒更新一次
	SR404	每1毫秒更新一次
	SR407	每1秒更新一次
	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	当主电源模块的外部24V低电压检测·状态改变时会更新此区域
	SR732	当冗余电源模块的外部24V低电压检测·状态改变时会更新此区域
	SR1000~SR1006	用户自行设定及清除
	SR1007	上电与网络线链接时
	SR1008	上电与网络线链接时
	SR1100~SR1107	每一扫描周期更新一次
	SR1116~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

适用机种	特殊数据寄存器	更新时间
	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
	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~SR1334	用户自行设定及清除
AH5X0	SR1335~SR1336	PLC Link启动时·每一扫描周期更新一次。
AH5X1		COM1-MODBUS启动时·每一扫描周期更新一次。
	SR1337~SR1787	用户自行设定及清除
	SR1792~SR2559	每一扫描周期更新一次
	SR2560~SR2659	有错误发生时更新一次

注：上表字段「适用机种」中·标示「AH5X0」者·表示适用机种 AHCPU500/510/520/530；标示「AH5X1」者·表示适用机种 AHCPU501/511/521/531；表格空白无任何标示·表示适用所有机种。

2.2.16 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，会将错误记录清除，以及清除警告的灯号。

3. 万年历

- SM220 · SR220~SR226 · 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 6.17 节万年历指令。

4. 通讯相关功能

- SM96~SM107 · SM209~SM212 · SR201~SR202 · SR209~SR216

SR215 · SR216 用来记录 PLC COM 通讯端口的代号，代号代表的端口功能如下表：

代号	0	1	2
端口功能	RS-232	RS-485	RS-422

若 PLC COM 端口为 RS-485 · RS-232 · RS-422 时，SR209 记录 PLC COM1 的通讯格式，SR212 记录 PLC COM2 的通讯格式。通讯协议设定值设定方式如下表说明，其它有关 SM、SR 设定通讯的详细说明请参考第 6.19 节通讯指令。

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	(H 1)	:	4800	
	0010	(H 2)	:	9600	
	0011	(H 3)	:	19200	
	0100	(H 4)	:	38400	
	0101	(H 5)	:	57600	
	0110	(H 6)	:	115200	
	0111	(H 7)	:	230400	RS-232 不支持
	1000	(H 8)	:	460800	RS-232 不支持
	1001	(H 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。

关于各装置的停电保持区请参考停电保持区的装置范围。

2

6. PLC 错误记录 (Error Log)

● SR40~SR161

SR40：储存错误记录 (Error Log) 的有效组数，最多 20 组，且每一组占用 6 个寄存器。

SR41：错误记录的指针，会指向最新一组的错误记录。当错误发生时，错误记录指针增加 1。

指针的范围值为 0~19，例如：SR41=3 表示第 4 组。

SR42~SR161：记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 上	SR42 下	SR43	SR44	SR45 上	SR45 下	SR46 上	SR46 下	SR47 上	SR47 下
2	SR48 上	SR48 下	SR49	SR50	SR51 上	SR51 下	SR52 上	SR52 下	SR53 上	SR53 下
3	SR54 上	SR54 下	SR55	SR56	SR57 上	SR57 下	SR58 上	SR58 下	SR59 上	SR59 下
4	SR60 上	SR60 下	SR61	SR62	SR63 上	SR63 下	SR64 上	SR64 下	SR65 上	SR65 下
5	SR66 上	SR66 下	SR67	SR68	SR69 上	SR69 下	SR70 上	SR70 下	SR71 上	SR71 下
6	SR72 上	SR72 下	SR73	SR74	SR75 上	SR75 下	SR76 上	SR76 下	SR77 上	SR77 下
7	SR78 上	SR78 下	SR79	SR80	SR81 上	SR81 下	SR82 上	SR82 下	SR83 上	SR83 下
8	SR84 上	SR84 下	SR85	SR86	SR87 上	SR87 下	SR88 上	SR88 下	SR89 上	SR89 下
9	SR90 上	SR90 下	SR91	SR92	SR93 上	SR93 下	SR94 上	SR94 下	SR95 上	SR95 下
10	SR96 上	SR96 下	SR97	SR98	SR99 上	SR99 下	SR100 上	SR100 下	SR101 上	SR101 下
11	SR102 上	SR102 下	SR103	SR104	SR105 上	SR105 下	SR106 上	SR106 下	SR107 上	SR107 下
12	SR108 上	SR108 下	SR109	SR110	SR111 上	SR111 下	SR112 上	SR112 下	SR113 上	SR113 下
13	SR114 上	SR114 下	SR115	SR116	SR117 上	SR117 下	SR118 上	SR118 下	SR119 上	SR119 下
14	SR120 上	SR120 下	SR121	SR122	SR123 上	SR123 下	SR124 上	SR124 下	SR125 上	SR125 下
15	SR126 上	SR126 下	SR127	SR128	SR129 上	SR129 下	SR130 上	SR130 下	SR131 上	SR131 下
16	SR132 上	SR132 下	SR133	SR134	SR135 上	SR135 下	SR136 上	SR136 下	SR137 上	SR137 下
17	SR138 上	SR138 下	SR139	SR140	SR141 上	SR141 下	SR142 上	SR142 下	SR143 上	SR143 下
18	SR144 上	SR144 下	SR145	SR146	SR147 上	SR147 下	SR148 上	SR148 下	SR149 上	SR149 下
19	SR150 上	SR150 下	SR151	SR152	SR153 上	SR153 下	SR154 上	SR154 下	SR155 上	SR155 下
20	SR156 上	SR156 下	SR157	SR158	SR159 上	SR159 下	SR160 上	SR160 下	SR161 上	SR161 下

7. PLC 下载记录 (Down Load 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。目前会记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：下载动作的记录指针 (Download Log pointer)，会指向最新一组的下载动作记录。每执行一次下载动作时，下载动作记录指標增加 1。指针的范围值为 0~19，例如 SR228=3 表示第 4 组。

SR229~SR308：记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 上	SR230 下	SR231 上	SR231 下	SR232 上	SR232 下
2	SR233	SR234 上	SR234 下	SR235 上	SR235 下	SR236 上	SR236 下
3	SR237	SR238 上	SR238 下	SR239 上	SR239 下	SR240 上	SR240 下
4	SR241	SR242 上	SR242 下	SR243 上	SR243 下	SR244 上	SR244 下
5	SR245	SR246 上	SR246 下	SR247 上	SR247 下	SR248 上	SR248 下
6	SR249	SR250 上	SR250 下	SR251 上	SR251 下	SR252 上	SR252 下
7	SR253	SR254 上	SR254 下	SR255 上	SR255 下	SR256 上	SR256 下
8	SR257	SR258 上	SR258 下	SR259 上	SR259 下	SR260 上	SR260 下
9	SR261	SR262 上	SR262 下	SR263 上	SR263 下	SR264 上	SR264 下
10	SR265	SR266 上	SR266 下	SR267 上	SR267 下	SR268 上	SR268 下
11	SR269	SR270 上	SR270 下	SR271 上	SR271 下	SR272 上	SR272 下
12	SR273	SR274 上	SR274 下	SR275 上	SR275 下	SR276 上	SR276 下
13	SR277	SR278 上	SR278 下	SR279 上	SR279 下	SR280 上	SR280 下
14	SR281	SR282 上	SR282 下	SR283 上	SR283 下	SR284 上	SR284 下
15	SR285	SR286 上	SR286 下	SR287 上	SR287 下	SR288 上	SR288 下
16	SR289	SR290 上	SR290 下	SR291 上	SR291 下	SR292 上	SR292 下
17	SR293	SR294 上	SR294 下	SR295 上	SR295 下	SR296 上	SR296 下
18	SR297	SR298 上	SR298 下	SR299 上	SR299 下	SR300 上	SR300 下
19	SR301	SR302 上	SR302 下	SR303 上	SR303 下	SR304 上	SR304 下
20	SR305	SR306 上	SR306 下	SR307 上	SR307 下	SR308 上	SR308 下

*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 状态变更记录 (PLC Status Log)

- SR309~SR390

SR309：记录 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。目前会记录的 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 回归出厂设定 (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 状态变更的记录。当 PLC 的状态被变更一次，记录指针增加 1。指针的范围值为 0~19。例如 SR310=3 表示第 4 组。

SR311~SR390：记录各组 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 上	SR312 下	SR313 上	SR313 下	SR314 上	SR314 下
2	SR315	SR316 上	SR316 下	SR317 上	SR317 下	SR318 上	SR318 下
3	SR319	SR320 上	SR320 下	SR321 上	SR321 下	SR322 上	SR322 下
4	SR323	SR324 上	SR324 下	SR325 上	SR325 下	SR326 上	SR326 下
5	SR327	SR328 上	SR328 下	SR329 上	SR329 下	SR330 上	SR330 下
6	SR331	SR332 上	SR332 下	SR333 上	SR333 下	SR334 上	SR334 下
7	SR335	SR336 上	SR336 下	SR337 上	SR337 下	SR338 上	SR338 下
8	SR339	SR340 上	SR340 下	SR341 上	SR341 下	SR342 上	SR342 下
9	SR343	SR344 上	SR344 下	SR345 上	SR345 下	SR346 上	SR346 下
10	SR347	SR348 上	SR348 下	SR349 上	SR349 下	SR350 上	SR350 下
11	SR351	SR352 上	SR352 下	SR353 上	SR353 下	SR354 上	SR354 下
12	SR355	SR356 上	SR356 下	SR357 上	SR357 下	SR358 上	SR358 下
13	SR359	SR360 上	SR360 下	SR361 上	SR361 下	SR362 上	SR362 下
14	SR363	SR364 上	SR364 下	SR365 上	SR365 下	SR366 上	SR366 下
15	SR367	SR368 上	SR368 下	SR369 上	SR369 下	SR370 上	SR370 下
16	SR371	SR372 上	SR372 下	SR373 上	SR373 下	SR374 上	SR374 下
17	SR375	SR376 上	SR376 下	SR377 上	SR377 下	SR378 上	SR378 下
18	SR379	SR380 上	SR380 下	SR381 上	SR381 下	SR382 上	SR382 下
19	SR383	SR384 上	SR384 下	SR385 上	SR385 下	SR386 上	SR386 下

组数	动作 编号	*PLC 变更时间					
		年	月	日	时	分	秒
20	SR387	SR388 上	SR388 下	SR389 上	SR389 下	SR390 上	SR390 下

*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

2

9. PLC 运行标志

● SM400~SM403

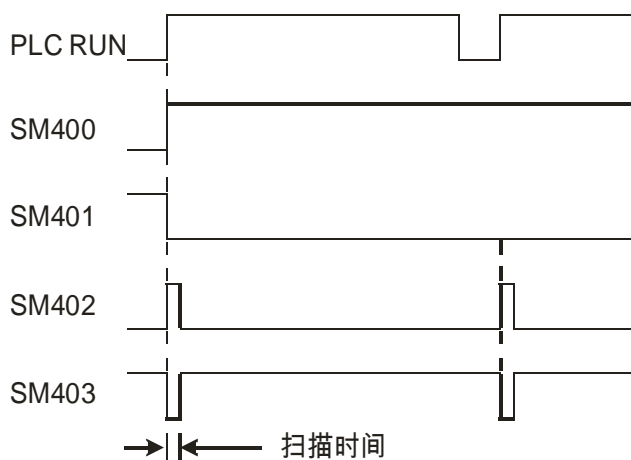
SM400：常开接点。



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。即启动反向（RUN 的瞬间“OFF”）脉冲。



10. 内部时间脉冲

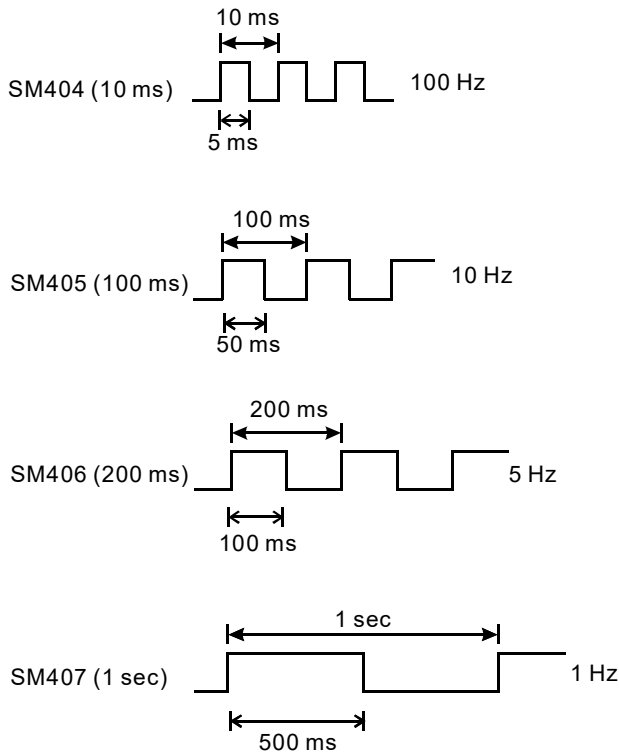
- SM404~SM410 · SR409~SR410

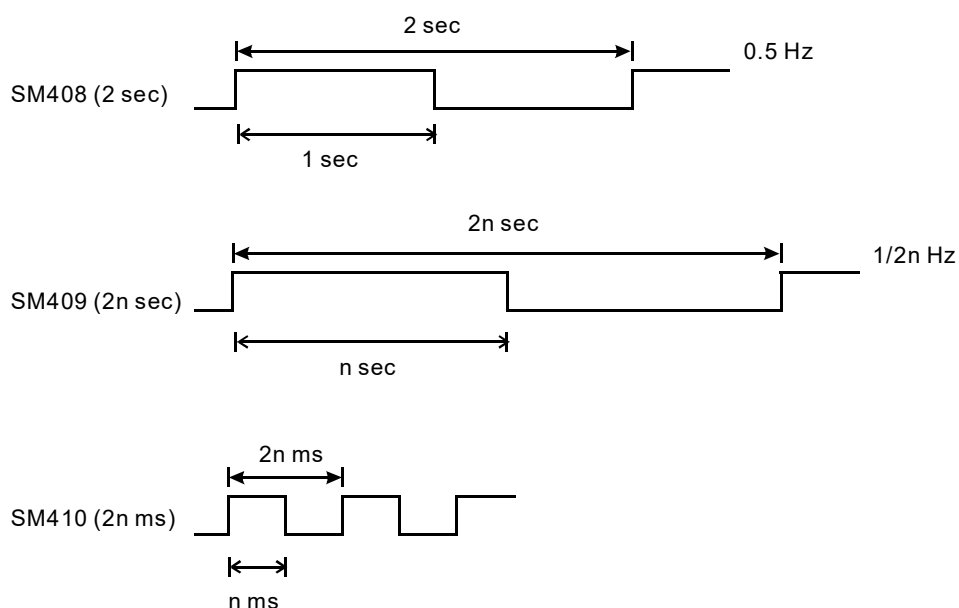
PLC 内部提供七种时钟脉冲。只要 PLC 通上电源，这七种时钟脉冲会自动动作，其中有两个提供了用户可以自行设定间隔时间的时钟脉冲。

2

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：





2

11. 存储卡相关标志

- SM450~SM453 · SR453

存储卡主要提供用户备份 PLC 内容。其对应的 SM · SR 功能如下说明，详细使用方式请参考 6.24 节存储卡读写指令。

装置	功能说明
SM450	存储卡是否存在标志，ON：存在/OFF：不存在。
SM451	存储卡写保护开关（Protect Switch），ON：写保护/OFF：无写保护。
SM452	存储卡正被存取（Accessed）中，ON：存取中/OFF：无存取。
SM453	存储卡运行中有错误发生，ON-表错误发生
SR453	当存储卡有错误发生，错误代码将被记录。

12. I/O 模块相关标志

- SR655~SR730 记录 I/O 表映像错误或 I/O 模块发生错误。

SR655~SR730：记录 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	扩展背板							
	背板 1	背板 2	背板 3	背板 4	背板 5	背板 6	背板 7	背板 8	
装置	SR655	SR656	SR657	SR658	SR659	SR660	SR661	SR662	
插槽 0	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0	Bit0	
插槽 1	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	Bit1	
插槽 2	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2	

2

说明	主背板	扩展背板						
	背板 1	背板 2	背板 3	背板 4	背板 5	背板 6	背板 7	背板 8
插槽 3	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3	Bit3
插槽 4	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4	Bit4
插槽 5	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5	Bit5
插槽 6	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6	Bit6
插槽 7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7	Bit7
插槽 8	Bit8	-	-	-	-	-	-	-
插槽 9	Bit9	-	-	-	-	-	-	-
插槽 10	Bit10	-	-	-	-	-	-	-
插槽 11	Bit11	-	-	-	-	-	-	-

SR663~SR730：记录 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记录错误码，得知发生何种错误。各背板插槽对应的 SR 如下：

说明 背板 插槽	主背板	扩展背板						
	背板 1	背板 2	背板 3	背板 4	背板 5	背板 6	背板 7	背板 8
插槽 0	SR663	SR675	SR683	SR691	SR699	SR707	SR715	SR723
插槽 1	SR664	SR676	SR684	SR692	SR700	SR708	SR716	SR724
插槽 2	SR665	SR677	SR685	SR693	SR701	SR709	SR717	SR725
插槽 3	SR666	SR678	SR686	SR694	SR702	SR710	SR718	SR726
插槽 4	SR667	SR679	SR687	SR695	SR703	SR711	SR719	SR727
插槽 5	SR668	SR680	SR688	SR696	SR704	SR712	SR720	SR728
插槽 6	SR669	SR681	SR689	SR697	SR705	SR713	SR721	SR729
插槽 7	SR670	SR682	SR690	SR698	SR706	SR714	SR722	SR730
插槽 8	SR671	-	-	-	-	-	-	-
插槽 9	SR672	-	-	-	-	-	-	-
插槽 10	SR673	-	-	-	-	-	-	-
插槽 11	SR674	-	-	-	-	-	-	-

13. 以太网相关标志

- SM1001~ SM1003、SM1089、SM1090、SM1091、SM1106~SM1109

SM 编号	标志说明	标志作用
SM1001	Ethernet Port1 网络线联机状态	OFF：Ethernet Port1 未连接 ON：Ethernet Port1 已连接

SM 编号	标志说明	标志作用
SM1002	Ethernet Port2 网络线联机状态	OFF : Ethernet Port2 未连接 ON : Ethernet Port2 已连接
SM1003	光纤同步电缆连接状态	OFF : 光纤同步电缆未连接 ON : 光纤同步电缆已连接
SM1089	MODBUS TCP Server 联机已满	ON : MODBUS TCP Server 联机已满
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 本地通讯口已被使用	ON 时表示使用到同到的连接通讯口

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 AH500 操作手册第 12.2 节。

14. Email 发送的设定

- SM1112~SM1113 · SM1116~SM1195

要发送 Email 前须先设定邮件的相关参数，若设定失败 SM1112 会被设为 ON。若 Email 发送后失败，则 Email 服务失败标志 SM1113 会被设定为 ON。

邮件发送条件 1 (Trigger1) ~ 邮件发送条件 8 (Trigger8) 的参数设定与触发后的结果标志 (SM1116~SM1195) 说明如下表：

功能说明	项目							
	Trigger 1	Trigger 2	Trigger 3	Trigger 4	Trigger 5	Trigger 6	Trigger 7	Trigger 8
Email 触发开关	SM1116	SM1126	SM1136	SM1146	SM1156	SM1166	SM1176	SM1186
	基本设定错误，则设为 ON							
Email 触发标志	SM1117	SM1127	SM1137	SM1147	SM1157	SM1167	SM1177	SM1187
	滤波器设定错误，则设为 ON							
Email 触发状态 0	SM1118	SM1128	SM1138	SM1148	SM1158	SM1168	SM1178	SM1188
	自从启用本触发条件后，无任何邮件被发送							
Email 触发状态 1	SM1119	SM1129	SM1139	SM1149	SM1159	SM1169	SM1179	SM1189
	触发条件启用且最后一封邮件已成功送出							
Email 触发状态 2	SM1120	SM1130	SM1140	SM1150	SM1160	SM1170	SM1180	SM1190
	启用本触发条件且最近一封邮件发送错误							

2

项目 功能 说明	Trigger 1	Trigger 2	Trigger 3	Trigger 4	Trigger 5	Trigger 6	Trigger 7	Trigger 8
Email 触 发状态 3	SM1121	SM1131	SM1141	SM1151	SM1161	SM1171	SM1181	SM1191
	触发条件启用且邮件已送出							
SMTP 回 复逾时标 志	SM1122	SM1132	SM1142	SM1152	SM1162	SM1172	SM1182	SM1192
	触发条件被触发且 SMTP 服务器响应逾时							
SMTP 回 复错误标 志	SM1123	SM1133	SM1143	SM1153	SM1163	SM1173	SM1183	SM1193
	触发条件被触发且 SMTP 服务器响应错误							
Email 附 件大小错 误标志	SM1124	SM1134	SM1144	SM1154	SM1164	SM1174	SM1184	SM1194
	触发条件被触发且附件大小超过限制							
找不到附 件错误标 志	SM1125	SM1135	SM1145	SM1155	SM1165	SM1175	SM1185	SM1195
	触发条件被触发且附件不存在							

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 AH500 操作手册第 12.2 节。

15. 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 为设定透过 TCP 协议进行数据交换之 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 项目	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
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

Socket 编号 项目	1	2	3	4	5	6	7	8
接收地址 L	SR1127	SR1140	SR1153	SR1166	SR1179	SR1192	SR1205	SR1218
保持联机时间	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

SR1222~SR1317 为设定透过 UDP 协议进行数据交换之 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
远程通讯端口	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 6.22 节以太网控制指令。

16. PLC LINK 相关功能

● SM1392~SM1598 · SR1335~SR1787

PLC LINK 支持 PLC COM1，最多可连接 32 台从站，与 AH500 系列机种最多可以读写 450 笔 Word 或 7200 笔 Coil；与其它支持标准 MODBUS 机种最多可以读写 100 笔 Word 或 1600 笔 Coil。

2

		主站							
		从站 1		从站 2		...	从站 32		
		读	写	读	写	...	读	写	
停电保持区	主站上的地址 :将读取到的数据写到此地址里 (SR1404 、 SR1405)	主站上的地址 :将此地址里的数据传给从站 (SR1468 、 SR1469)	主站上的地址 :将读取到的数据写到此地址里 (SR1406 、 SR1407)	主站上的地址 :将此地址里的数据传给从站 (SR1470 、 SR1471)	...	主站上的地址 :将读取到的数据写到此地址里 (SR1466 、 SR1467)	主站上的地址 :将此地址里的数据传给从站 (SR1530 、 SR1531)		
	从站上的地址 :读取装置起始地址 (SR1532 、 SR1533)	从站上的地址 :写入装置起始地址 (SR1596 、 SR1597)	从站上的地址 :读取装置起始地址 (SR1534 、 SR1535)	从站上的地址 :写入装置起始地址 (SR1598 、 SR1599)	...	从站上的地址 :读取装置起始地址 (SR1594 、 SR1595)	从站上的地址 :写入装置起始地址 (SR1658 、 SR1659)		
	读取装置长度 (SR1660)	写入装置长度 (SR1692)	读取装置长度 (SR1661)	写入装置长度 (SR1693)	...	读取装置长度 (SR1691)	写入装置长度 (SR1723)		
	装置类别 (SR1340)	装置类别 (SR1372)	装置类别 (SR1341)	装置类别 (SR1373)	...	装置类别 (SR1371)	装置类别 (SR1403)		
	从站的机种类型 SR1724			从站的机种类型 SR1725			...	从站的机种类型 SR1755	
	从站站号 ID (SR1756)			从站站号 ID (SR1757)			...	从站站号 (SR1787)	
	从站的机种类型 SR1724			从站的机种类型 SR1725			...	从站的机种类型 SR1755	
非停电保持区	链接标志。观看从站是否有链接 (SM1392)		链接标志。观看从站是否有链接 (SM1393)		...	链接标志。观看从站是否有链接 (SM1423)			
	数据交换标志。主从站是否有在做数据交换的动作 (SM1424)		数据交换标志。主从站是否有在做数据交换的动作 (SM1425)		...	数据交换标志。主从站是否有在做数据交换的动作 (SM1455)			
	读取错误标志 (SM1456)	写入错误标志 (SM1488)	读取错误标志 (SM1457)	写入错误标志 (SM1489)	...	读取错误标志 (SM1487)	写入错误标志 (SM1519)		
	读取完成标志 (ON->OFF) (SM1520)		读取完成标志 (ON->OFF) (SM1521)		...	读取完成标志 (ON->OFF) (SM1551)			
	写入完成标志 (ON->OFF) (SM1552)		写入完成标志 (ON->OFF) (SM1553)		...	写入完成标志 (ON->OFF) (SM1583)			

详细说明请参考 AH500 操作手册第 11.1 节 PLC Link。

17. Ether LINK 相关功能

端口	Ether Link 启动标志 OFF : 停止 ON : 启动	Ether Link 运作错误标志 OFF : 运作错误 ON : 运作正确	Ether Link 运作状态标志 OFF : 停止中 ON : 执行中
CPU	SM1770	SM1788	SM1806
Port 0	SM1772	SM1790	SM1808

端口	Ether Link 启动标志	Ether Link 运作错误标志	Ether Link 运作状态标志
	OFF : 停止 ON : 启动	OFF : 运作错误 ON : 运作正确	OFF : 停止中 ON : 执行中
Port 1	SM1773	SM1791	SM1809
Port 2	SM1774	SM1792	SM1810
Port 3	SM1775	SM1793	SM1811
Port 4	SM1776	SM1794	SM1812
Port 5	SM1777	SM1795	SM1813
Port 6	SM1778	SM1796	SM1814
Port 7	SM1779	SM1797	SM1815
Port 8	SM1780	SM1798	SM1816
Port 9	SM1781	SM1799	SM1817
Port 10	SM1782	SM1800	SM1818
Port 11	SM1783	SM1801	SM1819
Port 12	SM1784	SM1802	SM1820
Port 13	SM1785	SM1803	SM1821
Port 14	SM1786	SM1804	SM1822
Port 15	SM1787	SM1805	SM1823

详细说明请参考 AH500 操作手册第 11.2 节 Ether Link。

18. IP 地址设定

- SR1792~SR2047

装置	功能	说明
SR1792	区块 1 IP 地址	区块 1 连接的 IP 地址上 8 位 Ex : 远程位置为 192.168.1.100 , 则此寄存器的值填 16#C0A8
SR1793	区块 1 IP 地址	区块 1 连接的 IP 地址下 8 位 Ex 远程位置为 192.168.1.100 则此寄存器的值填 16#0164
		⋮
SR2046	区块 128 IP 地址	区块 128 连接的 IP 地址上 8 位 Ex : 远程位置为 192.168.1.100 , 则此寄存器的值填 16#C0A8
SR2047	区块 128 IP 地址	区块 128 连接的 IP 地址下 8 位 Ex 远程位置为 192.168.1.100 则此寄存器的值填 16#0164

Ether Link 部份之 SM、SR 的详细说明请参考 AH500 操作手册 11.2 节 Ether Link。

19. 电源模块相关标志

- SM20、SR32~SR33、SR731、SR732

装置	功能	说明
SM20	电源供应异常纪录	曾经发生内部电源供应不足后, 电源又回复到正常状态时记录。
SR32	主电源模块状态	ON : 主电源模块异常 OFF : 主电源模块正常运行



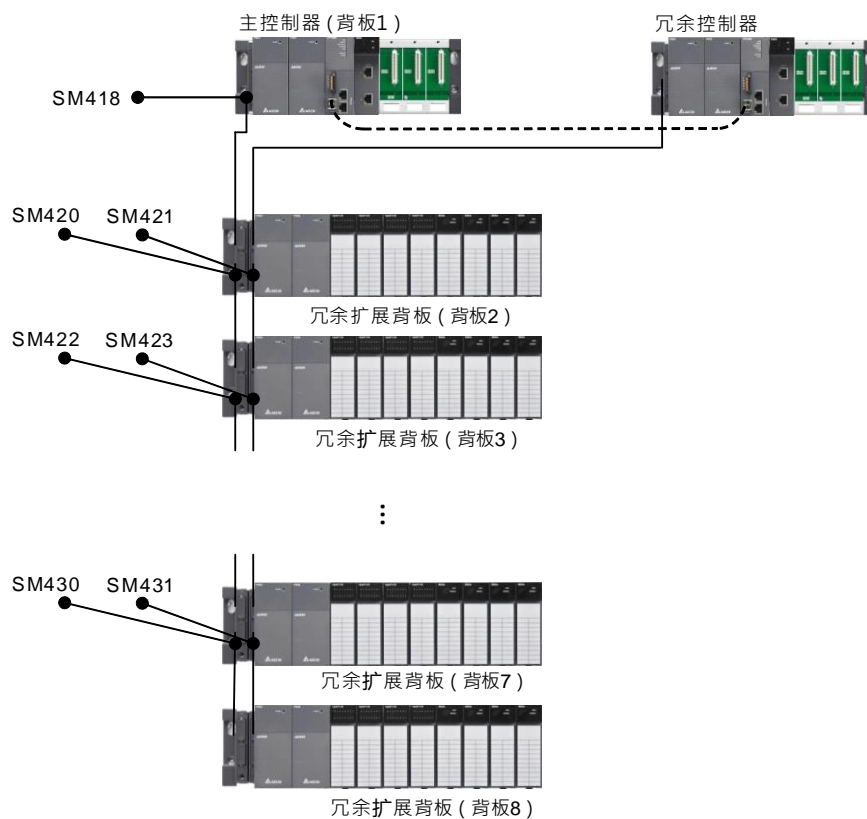
装置	功能	说明
SR33	冗余电源模块状态	ON：冗余电源模块异常 OFF：冗余电源模块正常运行
SR731	主电源模块外部 24V 低电压检测	ON：主电源模块外部 24V 电源检测点输入异常 OFF：主电源模块外部 24V 电源检测点输入正常
SR732	冗余电源模块外部 24V 低电压检测	ON：冗余电源模块外部 24V 电源检测点输入异常 OFF：冗余电源模块外部 24V 电源检测点输入正常

说明		电源模块状态		外部 24V 低电压检测	
		主电源模块	冗余电源模块	主电源模块	冗余电源模块
装置		SR32	SR33	SR731	SR732
主背板	背板 1	Bit0	Bit0	Bit0	Bit0
	背板 2	Bit1	Bit1	Bit1	Bit1
扩展背板	背板 3	Bit2	Bit2	Bit2	Bit2
	背板 4	Bit3	Bit3	Bit3	Bit3
	背板 5	Bit4	Bit4	Bit4	Bit4
	背板 6	Bit5	Bit5	Bit5	Bit5
	背板 7	Bit6	Bit6	Bit6	Bit6
	背板 8	Bit7	Bit7	Bit7	Bit7

20. 冗余扩展背板联机状态

- SM418~SM431：冗余背板扩展界面通讯端口状态（ON：联机正常，OFF：联机异常）

说明		电源模块状态	
		Port2	Port4
(冗余)主背板	背板 1	SM418	-
冗余扩展背板	背板 2	SM420	SM421
	背板 3	SM422	SM423
	背板 4	SM424	SM425
	背板 5	SM426	SM427
	背板 6	SM428	SM429
	背板 7	SM430	SM431



21. 主机信息相关标志

- SR440~SR442、SR443~SR451
 SR440~SR442：主机 MAC Address。
 SR443~SR451：主机产品序号。

主机 MAC Address 00:18:23:10:F5:1A		主机产品序号 CPU52120W15480004	
装置	内容	装置	内容
SR440	0018	SR443	PC
SR441	2310	SR444	5U
SR442	F51A	SR445	12
		SR446	02
		SR447	1W
		SR448	45
		SR449	08
		SR450	00
		SR451	40

AHCPU 主机会自动将 MAC Address 与产品序号以 ASCII Code 格式存入相对应的 SR 装置。

2.2.17 连结寄存器 L

L装置主要用于PLC Link或Ether Link的数据交换功能，当AH500对AH500进行数据交换时，可以使用L装置作为数据交换的缓冲区，详细说明请参考AH500操作手册第12章便利功能使用说明。

连接寄存器L的装置编号为L0~L65535共65536个Words（因机种不同，其装置范围有所差异），也可当作一般的辅助寄存器使用。

2

2.2.18 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~E31。变址的使用方式，请参考第4.2节变址说明。

3

第3章 指令表

目录

3.1 指令类型	3-2
3.1.1 基本指令	3-2
3.1.2 应用指令	3-2
3.2 指令表说明	3-3
3.2.1 基本指令	3-3
3.2.2 应用指令 (依 API 号码排序)	3-4
3.2.3 应用指令 (依英文字母排序)	3-5
3.2.4 指令装置表说明	3-6
3.3 基本指令一览表	3-7
3.4 应用指令一览表	3-10
3.4.1 应用指令 (依 API 号码排序)	3-10
3.4.2 应用指令 (依英文字母排序)	3-42

3.1 指令类型

AH500 指令的主要类型包括顺序基本指令、应用指令。

3.1.1 基本指令

分类	说明
接点指令	载入接点·串联接点·并联接点...等
连接指令	储存或读取接点间操作结果的连接
输出指令	位设备输出·脉冲输出
主控制指令	控制共通串联接点
高低电平检出接点指令	触发加载接点·串联接点·并联接点的相关指令
上下微分输出指令	位设备微分输出
其它指令	其它指令

3.1.2 应用指令

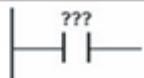


API	分类	说明
0000~0065	比较操作指令	比较操作如=、<>、>、>=、<、<=...等
0100~0118	四则运算指令	BIN 或 BCD 的加法、减法、乘法或除法
0200~0219	数据转换指令	将 BCD 转换成 BIN 和将 BIN 转换成 BCD
0300~0310	数据转移指令	传送指定的数据
0400~0402	程序跳转指令	程序跳转
0500~0502	程序执行控制指令	允许或禁止中断程序
0600	I/O 更新指令	运行局部更新
0700~0708	便利指令	用于以下目的的指令:计数器增加/减小·教学定时器·特殊功能定时器·循环台最短距离控制...等
0800~0817	逻辑操作指令	逻辑操作·如逻辑加法·逻辑乘法等
0900~0904	循环指令	指定数据的循环移位
1000~1004	基本指令	定时器·计数器 指令
1100~1115	移位指令	指定数据的移位
1200~1223	数据处理指令	16 位数据查询·数据处理如译码和编码
1300~1302	结构建立指令	循环回路·调用功能块...等
1400~1401	模块的数据读/写指令	特殊功能模块的数据读/写
1500~1524	浮点数指令	浮点数运算指令
1600~1606	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
1700~1704	外围设备指令	连接到外围设备的 I/O
1800~1812	通讯指令	通讯控制接口设备

API	分类	说明
1900~1905	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
2100~2121	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
2200~2207	以太网控制指令	控制以太网数据交换相关指令
2300~2302	存储卡读写指令	存储卡读写相关指令
2400~2401	任务控制指令	程序中任务控制相关指令
2500~2502	SFC 控制指令	控制 SFC 相关指令
2900~2901	冗余系统控制指令	冗余系统相关指令

3

3.2 指令表说明

3.2.1 基本指令

指令码	符号	功能	操作数
LD		载入/串联/并联/接点	DX · X · Y · M · S · T · C · HC · D · L · SM · PR
AND			
OR			

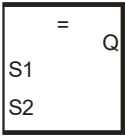
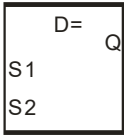
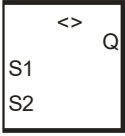
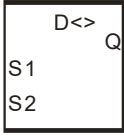
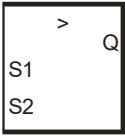
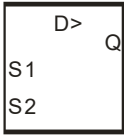
①
②
③
④

说明：

- ①：指令的名称
- ②：在 ISPSOft 梯形图所显示的图形
- ③：指令的功能用途说明
- ④：指令所支持的操作数

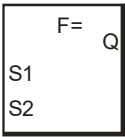
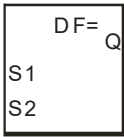
3.2.2 应用指令 (依 API 号码排序)

3

API	指令码		P 指令	符号		功能
	16位	32位				
0000	LD=	DLD=	-			接点类型比较 S1=S2导通 S1≠S2不导通
0001	LD<>	DLD<>	-			接点类型比较 S1≠S2导通 S1=S2不导通
0002	LD>	DLD>	-			接点类型比较 S1>S2导通 S1≤S2不导通

↓
↓
↓
↓
↓
↓

①
②
③
④
⑤
⑥

API	指令码		P 指令	符号		功能
	32位	64位				
0018	FLD=	DFLD=	-			接点类型比较 S1=S2 导通 S1≠S2 不导通

↓

⑦

说明：

①：指令 API 编号。

②：指令的名称。

- ③：16 位的指令，如有支持 32 位，则在指令前加 D，即为 32 位指令。
- ④：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：在 ISPSOft 梯形图所显示的图形。
- ⑥：指令的功能用途说明。
- ⑦：32 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。

3.2.3 应用指令 (依英文字母排序)

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
G	0209	GBIN	DGBIN	-	√	GRY码→BIN变换
	0402	GOEND	-	-	-	跳转到 END
	1902	GPWM	-	-	-	一般用脉冲波宽调频
	0208	GRY	DGRY	-	√	BIN→GRY码变换
H	2104	HABIN	DHABIN	-	√	十六进制ASCLL→十六进制 BIN变换
	1701	HKY	DHKY	-	-	16键键盘输入
	1604	HOUR	DHOUR	-	-	运转定时器

说明：

- ①：指令名称开头英文字母的分类。
- ②：指令 API 编号。
- ③ ~ ⑤：指令的名称。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，如有支持 64 位，则在指令前加 D，即为 64 位浮点数指令。
- ⑥：是否有脉冲型指令 (P 指令)，有：√。无：—。
- 要使用脉冲型指令，需在指令后加 P。
- ⑦：指令的功能用途说明。

3.3 基本指令一览表

● 接点指令

指令码	符号	功能	操作数
LD		载入/串联/并联 A 接点	DX、X、Y、M、S、T、C、 HC、D、L、SM、PR
AND			
OR			
LDI		载入/串联/并联 B 接点	DX、X、Y、M、S、T、C、 HC、D、L、SM、PR
ANI			
ORI			

● 连接指令



指令码	符号	功能	操作数
ANB		串联回路方块	—
ORB		并联回路方块	—
MPS	—	存入堆栈指令	—
MRD	—	堆栈读取指针不动	—
MPP	—	读出堆栈	—

● 输出指令

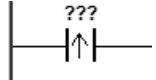

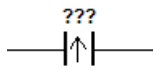
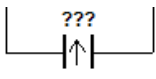
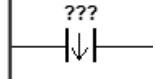

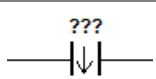
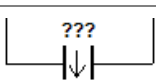
指令码	符号	功能	执行条件	操作数
OUT		驱动线圈		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR
SET		动作保持 (ON)		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR

3





● 主控制指令

指令码	符号	功能	操作数
MC		共通串联接点之连结	N
MCR		共通串联接点之解除	N

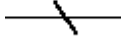

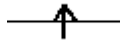
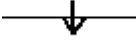
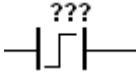
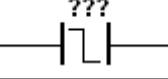
● 上下沿检出接点指令

指令码	符号	功能	执行条件	操作数
LDP		上升沿检出动作 开始/串联/并联连接		DX、X、Y、M、S、T、C、 HC、D、L、SM、PR
PED				
ANDP				
APED				
ORP				
OPED				
LDF		下降沿检出动作 开始/串联/并联连接		DX、X、Y、M、S、T、C、 HC、D、W、L、SM、PR
NED				
ANDF				
ANED				
ORF				
ONED				

● 上下微分输出指令

指令码	符号	功能	执行条件	操作数
PLS		上微分输出		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR
PLF		下微分输出		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR

● 其它指令

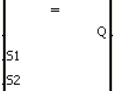
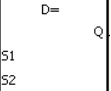
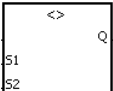
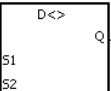
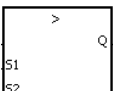
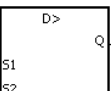
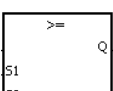
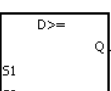
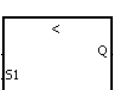
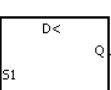
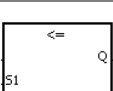
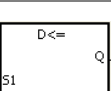
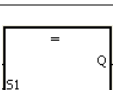
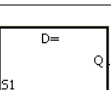
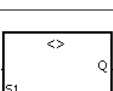
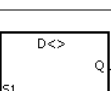
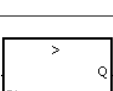
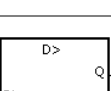
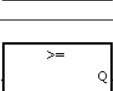
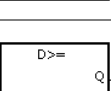
指令码	符号	功能	操作数
INV		运算结果反相	-
NOP	-	无动作	-
PSTOP		PLC 程序停止执行	-
NP		上升沿触发指令	-
PN		下降沿触发指令	-
FB_NP		上升沿触发指令	S
FB_PN		下降沿触发指令	S

3.4 应用指令一览表

3.4.1 应用指令 (依 API 号码排序)

● 比较操作指令

3

API	指令码		P 指令	符号		功能
	16 位	32 位				
0000	LD=	DLD=	-			接点类型比较 S1 = S2 导通 S1 ≠ S2 不导通
0001	LD<>	DLD<>	-			接点类型比较 S1 ≠ S2 导通 S1 = S2 不导通
0002	LD>	DLD>	-			接点类型比较 S1 > S2 导通 S1 ≤ S2 不导通
0003	LD>=	DLD>=	-			接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通
0004	LD<	DLD<	-			接点类型比较 S1 < S2 导通 S1 ≥ S2 不导通
0005	LD<=	DLD<=	-			接点类型比较 S1 ≤ S2 导通 S1 > S2 不导通
0006	AND=	DAND=	-			串连接点类型比较 S1 = S2 导通 S1 ≠ S2 不导通
0007	AND<>	DAND<>	-			串连接点类型比较 S1 ≠ S2 导通 S1 = S2 不导通
0008	AND>	DAND>	-			串连接点类型比较 S1 > S2 导通 S1 ≤ S2 不导通
0009	AND>=	DAND>=	-			串连接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通

API	指令码		P 指令	符号	功能
	16 位	32 位			
0010	AND<	DAND<	-		串连接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
0011	AND<=	DAND<=	-		串连接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
0012	OR=	DOR=	-		并联接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
0013	OR<>	DOR<>	-		并联接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
0014	OR>	DOR>	-		并联接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
0015	OR>=	DOR>=	-		并联接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
0016	OR<	DOR<	-		并联接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
0017	OR<=	DOR<=	-		并联接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通

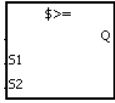
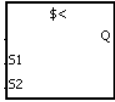
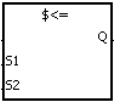
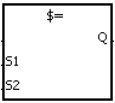
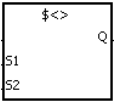
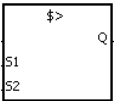
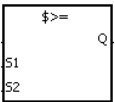
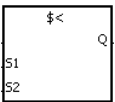
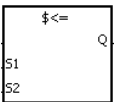
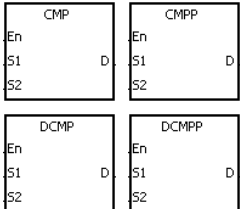
API	指令码		P 指令	符号	功能
	32 位	64 位			
0018	FLD=	DFLD=	-		浮点数接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
0019	FLD<>	DFLD<>	-		浮点数接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
0020	FLD>	DFLD>	-		浮点数接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通

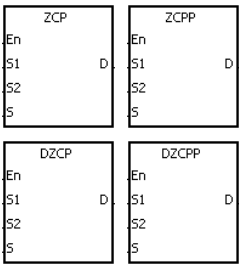
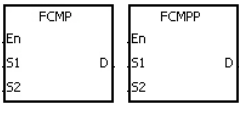
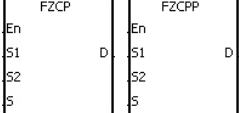
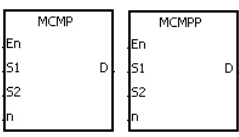
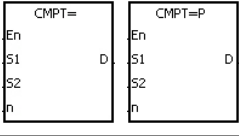
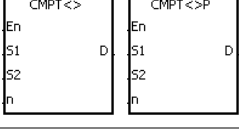
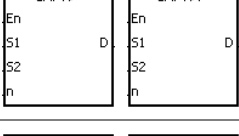
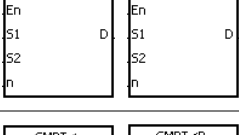
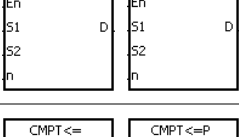
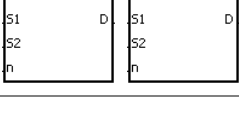
3

API	指令码		P 指令	符号	功能
	32 位	64 位			
0021	FLD>=	DFLD>=	-		浮点数接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
0022	FLD<	DFLD<	-		浮点数接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
0023	FLD<=	DFLD<=	-		浮点数接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
0024	FAND=	DFAND=	-		浮点数串连接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
0025	FAND<>	DFAND<>	-		浮点数串连接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
0026	FAND>	DFAND>	-		浮点数串连接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
0027	FAND>=	DFAND>=	-		浮点数串连接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
0028	FAND<	DFAND<	-		浮点数串连接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
0029	FAND<=	DFAND<=	-		浮点数串连接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
0030	FOR=	DFOR=	-		浮点数并连接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
0031	FOR<>	DFOR<>	-		浮点数并连接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
0032	FOR>	DFOR>	-		浮点数并连接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通

API	指令码		P 指令	符号	功能
	32 位	64 位			
0033	FOR>=	DFOR>=	-		浮点数并连接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
0034	FOR<	DFOR<	-		浮点数并连接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
0035	FOR<=	DFOR<=	-		浮点数并连接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
0036	LD\$=	-	-		字符串接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
0037	LD\$<>	-	-		字符串接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
0038	LD\$>	-	-		字符串接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通
0039	LD\$>=	-	-		字符串接点类型比较 $S1 \geq S2$ 导通 $S1 < S2$ 不导通
0040	LD\$<	-	-		字符串接点类型比较 $S1 < S2$ 导通 $S1 \geq S2$ 不导通
0041	LD\$<=	-	-		字符串接点类型比较 $S1 \leq S2$ 导通 $S1 > S2$ 不导通
0042	AND\$=	-	-		字符串串连接点类型比较 $S1 = S2$ 导通 $S1 \neq S2$ 不导通
0043	AND\$<>	-	-		字符串串连接点类型比较 $S1 \neq S2$ 导通 $S1 = S2$ 不导通
0044	AND\$>	-	-		字符串串连接点类型比较 $S1 > S2$ 导通 $S1 \leq S2$ 不导通

3

API	指令码		P 指令	符号	功能
	32 位	64 位			
0045	AND\$>=	—	—		字符串串连接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通
0046	AND\$<	—	—		字符串串连接点类型比较 S1 < S2 导通 S1 ≥ S2 不导通
0047	AND\$<=	—	—		字符串串连接点类型比较 S1 ≤ S2 导通 S1 > S2 不导通
0048	OR\$=	—	—		字符串串连接点类型比较 S1 = S2 导通 S1 ≠ S2 不导通
0049	OR\$<>	—	—		字符串串连接点类型比较 S1 ≠ S2 导通 S1 = S2 不导通
0050	OR\$>	—	—		字符串串连接点类型比较 S1 > S2 导通 S1 ≤ S2 不导通
0051	OR\$>=	—	—		字符串串连接点类型比较 S1 ≥ S2 导通 S1 < S2 不导通
0052	OR\$<	—	—		字符串串连接点类型比较 S1 < S2 导通 S1 ≥ S2 不导通
0053	OR\$<=	—	—		字符串串连接点类型比较 S1 ≤ S2 导通 S1 > S2 不导通
0054	CMP	DCMP	✓		比较设定输出

API	指令码		P 指令	符号	功能
	32 位	64 位			
0055	ZCP	DZCP	✓		区域比较
0056	—	FCMP	✓		浮点数比较
0057	—	FZCP	✓		浮点数区域比较
0058	MCMP	—	✓		矩阵比较
0059	CMPT=	—	✓		表格比较，=
0060	CMPT<>	—	✓		表格比较，≠
0061	CMPT>	—	✓		表格比较，>
0062	CMPT>=	—	✓		表格比较，≥
0063	CMPT<	—	✓		表格比较，<
0064	CMPT<=	—	✓		表格比较，≤

API	指令码		P 指令	符号	功能
	32 位	64 位			
0065	CHKADR	-	-		接点类型指针寄存器地址检查

● 四则运算指令

3

API	指令码		P 指令	符号	功能
	16 位	32 位			
0100	+	D+	✓		BIN 加法 $S1+S2=D$
0101	-	D-	✓		BIN 减法 $S1-S2=D$
0102	*	D*	✓		BIN 乘法 $S1*S2=D$
0103	/	D/	✓		BIN 除法 $S1/S2=D$

API	指令码		P 指令	符号	功能
	32 位	64 位			
0104	F+	DF+	✓		浮点数加法 $S1+S2=D$

API	指令码		P 指令	符号		功能																															
	32 位	64 位																																			
0105	F-	DF-	✓	<table border="1"> <tr><td colspan="2">F-</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">F-P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DF-</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DF-P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	F-		En	D	S1		S2		F-P		En	D	S1		S2		DF-		En	D	S1		S2		DF-P		En	D	S1		S2		浮点数减法 $S1-S2=D$
F-																																					
En	D																																				
S1																																					
S2																																					
F-P																																					
En	D																																				
S1																																					
S2																																					
DF-																																					
En	D																																				
S1																																					
S2																																					
DF-P																																					
En	D																																				
S1																																					
S2																																					
0106	F*	DF*	✓	<table border="1"> <tr><td colspan="2">F*</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">F*P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DF*</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DF*P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	F*		En	D	S1		S2		F*P		En	D	S1		S2		DF*		En	D	S1		S2		DF*P		En	D	S1		S2		浮点数乘法 $S1*S2=D$
F*																																					
En	D																																				
S1																																					
S2																																					
F*P																																					
En	D																																				
S1																																					
S2																																					
DF*																																					
En	D																																				
S1																																					
S2																																					
DF*P																																					
En	D																																				
S1																																					
S2																																					
0107	F/	DF/	✓	<table border="1"> <tr><td colspan="2">F/</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">F/P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DF/</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DF/P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	F/		En	D	S1		S2		F/P		En	D	S1		S2		DF/		En	D	S1		S2		DF/P		En	D	S1		S2		浮点数除法 $S1/S2=D$
F/																																					
En	D																																				
S1																																					
S2																																					
F/P																																					
En	D																																				
S1																																					
S2																																					
DF/																																					
En	D																																				
S1																																					
S2																																					
DF/P																																					
En	D																																				
S1																																					
S2																																					
0108	B+	DB+	✓	<table border="1"> <tr><td colspan="2">B+</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">B+P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB+</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB+P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	B+		En	D	S1		S2		B+P		En	D	S1		S2		DB+		En	D	S1		S2		DB+P		En	D	S1		S2		BCD 加法 $S1+S2=D$
B+																																					
En	D																																				
S1																																					
S2																																					
B+P																																					
En	D																																				
S1																																					
S2																																					
DB+																																					
En	D																																				
S1																																					
S2																																					
DB+P																																					
En	D																																				
S1																																					
S2																																					
0109	B-	DB-	✓	<table border="1"> <tr><td colspan="2">B-</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">B-P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB-</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB-P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	B-		En	D	S1		S2		B-P		En	D	S1		S2		DB-		En	D	S1		S2		DB-P		En	D	S1		S2		BCD 减法 $S1-S2=D$
B-																																					
En	D																																				
S1																																					
S2																																					
B-P																																					
En	D																																				
S1																																					
S2																																					
DB-																																					
En	D																																				
S1																																					
S2																																					
DB-P																																					
En	D																																				
S1																																					
S2																																					
0110	B*	DB*	✓	<table border="1"> <tr><td colspan="2">B*</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">B*P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB*</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB*P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	B*		En	D	S1		S2		B*P		En	D	S1		S2		DB*		En	D	S1		S2		DB*P		En	D	S1		S2		BCD 乘法 $S1*S2=D$
B*																																					
En	D																																				
S1																																					
S2																																					
B*P																																					
En	D																																				
S1																																					
S2																																					
DB*																																					
En	D																																				
S1																																					
S2																																					
DB*P																																					
En	D																																				
S1																																					
S2																																					
0111	B/	DB/	✓	<table border="1"> <tr><td colspan="2">B/</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">B/P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB/</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td colspan="2">DB/P</td></tr> <tr><td>En</td><td>D</td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	B/		En	D	S1		S2		B/P		En	D	S1		S2		DB/		En	D	S1		S2		DB/P		En	D	S1		S2		BCD 除法 $S1/S2=D$
B/																																					
En	D																																				
S1																																					
S2																																					
B/P																																					
En	D																																				
S1																																					
S2																																					
DB/																																					
En	D																																				
S1																																					
S2																																					
DB/P																																					
En	D																																				
S1																																					
S2																																					

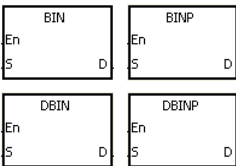
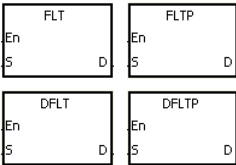
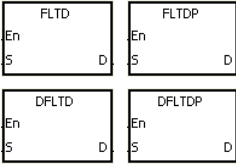
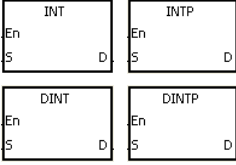
3

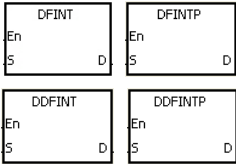
API	指令码		P 指令	符号	功能																
	32 位	64 位																			
0112	BK+	-	✓	<table border="1"> <tr> <td>En</td> <td>BK+</td> <td>En</td> <td>BK+P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> <tr> <td>n</td> <td></td> <td>n</td> <td></td> </tr> </table>	En	BK+	En	BK+P	S1	D	S1	D	S2		S2		n		n		连续区块 BIN 加法
En	BK+	En	BK+P																		
S1	D	S1	D																		
S2		S2																			
n		n																			
0113	BK-	-	✓	<table border="1"> <tr> <td>En</td> <td>BK-</td> <td>En</td> <td>BK-P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> <tr> <td>n</td> <td></td> <td>n</td> <td></td> </tr> </table>	En	BK-	En	BK-P	S1	D	S1	D	S2		S2		n		n		连续区块 BIN 减法
En	BK-	En	BK-P																		
S1	D	S1	D																		
S2		S2																			
n		n																			
0114	\$+	-	✓	<table border="1"> <tr> <td>En</td> <td>\$+</td> <td>En</td> <td>\$+P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> </table>	En	\$+	En	\$+P	S1	D	S1	D	S2		S2		字符串链接				
En	\$+	En	\$+P																		
S1	D	S1	D																		
S2		S2																			
0115	INC	DINC	✓	<table border="1"> <tr> <td>En</td> <td>INC</td> <td>En</td> <td>INCP</td> </tr> <tr> <td>D</td> <td></td> <td>D</td> <td></td> </tr> <tr> <td>En</td> <td>DINC</td> <td>En</td> <td>DINCP</td> </tr> <tr> <td>D</td> <td></td> <td>D</td> <td></td> </tr> </table>	En	INC	En	INCP	D		D		En	DINC	En	DINCP	D		D		BIN 加一
En	INC	En	INCP																		
D		D																			
En	DINC	En	DINCP																		
D		D																			
0116	DEC	DDEC	✓	<table border="1"> <tr> <td>En</td> <td>DEC</td> <td>En</td> <td>DECP</td> </tr> <tr> <td>D</td> <td></td> <td>D</td> <td></td> </tr> <tr> <td>En</td> <td>DDEC</td> <td>En</td> <td>DDECP</td> </tr> <tr> <td>D</td> <td></td> <td>D</td> <td></td> </tr> </table>	En	DEC	En	DECP	D		D		En	DDEC	En	DDECP	D		D		BIN 减一
En	DEC	En	DECP																		
D		D																			
En	DDEC	En	DDECP																		
D		D																			

API	指令码		P 指令	符号	功能																								
	16 位	32 位																											
0117	MUL16	MUL32	✓	<table border="1"> <tr> <td>En</td> <td>MUL16</td> <td>En</td> <td>MUL16P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> <tr> <td>En</td> <td>MUL32</td> <td>En</td> <td>MUL32P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> </table>	En	MUL16	En	MUL16P	S1	D	S1	D	S2		S2		En	MUL32	En	MUL32P	S1	D	S1	D	S2		S2		16 位专用 BIN 乘法/ 32 位专用 BIN 乘法
En	MUL16	En	MUL16P																										
S1	D	S1	D																										
S2		S2																											
En	MUL32	En	MUL32P																										
S1	D	S1	D																										
S2		S2																											
0118	DIV16	DIV32	✓	<table border="1"> <tr> <td>En</td> <td>DIV16</td> <td>En</td> <td>DIV16P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> <tr> <td>En</td> <td>DIV32</td> <td>En</td> <td>DIV32P</td> </tr> <tr> <td>S1</td> <td>D</td> <td>S1</td> <td>D</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> <td></td> </tr> </table>	En	DIV16	En	DIV16P	S1	D	S1	D	S2		S2		En	DIV32	En	DIV32P	S1	D	S1	D	S2		S2		16 位专用 BIN 除法/ 32 位专用 BIN 除法
En	DIV16	En	DIV16P																										
S1	D	S1	D																										
S2		S2																											
En	DIV32	En	DIV32P																										
S1	D	S1	D																										
S2		S2																											

● 数据转换指令

API	指令码		P 指令	符号	功能																
	16 位	32 位																			
0200	BCD	DBCD	✓	<table border="1"> <tr> <td>En</td> <td>BCD</td> <td>En</td> <td>BCDP</td> </tr> <tr> <td>S</td> <td>D</td> <td>S</td> <td>D</td> </tr> <tr> <td>En</td> <td>DBCD</td> <td>En</td> <td>DBCDP</td> </tr> <tr> <td>S</td> <td>D</td> <td>S</td> <td>D</td> </tr> </table>	En	BCD	En	BCDP	S	D	S	D	En	DBCD	En	DBCDP	S	D	S	D	BIN→BCD 变换
En	BCD	En	BCDP																		
S	D	S	D																		
En	DBCD	En	DBCDP																		
S	D	S	D																		

API	指令码		P 指令	符号	功能
	16 位	32 位			
0201	BIN	DBIN	✓		BCD→BIN 变换
0202	FLT	DFLT	✓		BIN 整数→二进浮点数变换
0203	FLTD	DFLTD	✓		BIN 整数→64 位浮点数变换
0204	INT	DINT	✓		二进浮点数→BIN 整数变换

API	指令码		P 指令	符号	功能
	32 位	64 位			
0205	FINT	DFINT	✓		64 位浮点数→BIN 整数变换

API	指令码		P 指令	符号	功能
	16 位	32 位			
0206	MMOV	—	✓		16→32 位数值转换
0207	RMOV	—	✓		32→16 位数值转换
0208	GRY	DGRY	✓		BIN→GRY 码变换

3

API	指令码		P 指令	符号	功能								
	16 位	32 位											
0209	GBIN	DGBIN	✓	<table border="1"> <tr> <td>GBIN</td> <td>GBINP</td> </tr> <tr> <td>En S D</td> <td>En S D</td> </tr> <tr> <td>DGBIN</td> <td>DGBINP</td> </tr> <tr> <td>En S D</td> <td>En S D</td> </tr> </table>	GBIN	GBINP	En S D	En S D	DGBIN	DGBINP	En S D	En S D	GRY 码→BIN 变换
GBIN	GBINP												
En S D	En S D												
DGBIN	DGBINP												
En S D	En S D												
0210	NEG	DNEG	✓	<table border="1"> <tr> <td>NEG</td> <td>NEGP</td> </tr> <tr> <td>En D</td> <td>En D</td> </tr> <tr> <td>DNEG</td> <td>DNEGP</td> </tr> <tr> <td>En D</td> <td>En D</td> </tr> </table>	NEG	NEGP	En D	En D	DNEG	DNEGP	En D	En D	取负数(取 2 的补码)
NEG	NEGP												
En D	En D												
DNEG	DNEGP												
En D	En D												
0211	-	FNEG	✓	<table border="1"> <tr> <td>FNEG</td> <td>FNEGP</td> </tr> <tr> <td>En D</td> <td>En D</td> </tr> </table>	FNEG	FNEGP	En D	En D	32 位浮点正负符号反相				
FNEG	FNEGP												
En D	En D												
0212	-	FBCD	✓	<table border="1"> <tr> <td>FBCD</td> <td>FBCDP</td> </tr> <tr> <td>En S D</td> <td>En S D</td> </tr> </table>	FBCD	FBCDP	En S D	En S D	二进浮点数→十进浮点数				
FBCD	FBCDP												
En S D	En S D												
0213	-	FBIN	✓	<table border="1"> <tr> <td>FBIN</td> <td>FBINP</td> </tr> <tr> <td>En S D</td> <td>En S D</td> </tr> </table>	FBIN	FBINP	En S D	En S D	十进浮点数→二进浮点数				
FBIN	FBINP												
En S D	En S D												
0214	BKBCD	-	✓	<table border="1"> <tr> <td>BKBCD</td> <td>BKBCDP</td> </tr> <tr> <td>En S D n</td> <td>En S D n</td> </tr> </table>	BKBCD	BKBCDP	En S D n	En S D n	连续区块 BIN→BCD 变换				
BKBCD	BKBCDP												
En S D n	En S D n												
0215	BKBIN	-	✓	<table border="1"> <tr> <td>BKBIN</td> <td>BKBINP</td> </tr> <tr> <td>En S D n</td> <td>En S D n</td> </tr> </table>	BKBIN	BKBINP	En S D n	En S D n	连续区块 BCD→BIN 变换				
BKBIN	BKBINP												
En S D n	En S D n												
0216	SCAL	-	✓	<table border="1"> <tr> <td>SCAL</td> <td>SCALP</td> </tr> <tr> <td>En S1 D S2 S3</td> <td>En S1 D S2 S3</td> </tr> </table>	SCAL	SCALP	En S1 D S2 S3	En S1 D S2 S3	比例运算				
SCAL	SCALP												
En S1 D S2 S3	En S1 D S2 S3												
0217	SCLP	DSCLP	✓	<table border="1"> <tr> <td>SCLP</td> <td>SCLPP</td> </tr> <tr> <td>En S1 D S2</td> <td>En S1 D S2</td> </tr> <tr> <td>DSCLP</td> <td>DSCLPP</td> </tr> <tr> <td>En S1 D S2</td> <td>En S1 D S2</td> </tr> </table>	SCLP	SCLPP	En S1 D S2	En S1 D S2	DSCLP	DSCLPP	En S1 D S2	En S1 D S2	参数型比例运算
SCLP	SCLPP												
En S1 D S2	En S1 D S2												
DSCLP	DSCLPP												
En S1 D S2	En S1 D S2												
0218	LINE	DLINE	✓	<table border="1"> <tr> <td>LINE</td> <td>LINEP</td> </tr> <tr> <td>En S D n</td> <td>En S D n</td> </tr> <tr> <td>DLINE</td> <td>DLINEP</td> </tr> <tr> <td>En S D n</td> <td>En S D n</td> </tr> </table>	LINE	LINEP	En S D n	En S D n	DLINE	DLINEP	En S D n	En S D n	COLUMN to LINE
LINE	LINEP												
En S D n	En S D n												
DLINE	DLINEP												
En S D n	En S D n												

API	指令码		P 指令	符号	功能
	16 位	32 位			
0219	COLM	DCOLM	✓		LINE to COLUMN

● 数据转移指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
0300	MOV	DMOV	✓		数据移动 S：数据来源 D：数据目的地

API	指令码		P 指令	符号	功能
	32 位	64 位			
0301	—	DFMOV	✓		64 位浮点数数据移动 S：数据来源 D：数据目的地

API	指令码		P 指令	符号	功能
	16 位	32 位			
0302	\$MOV	—	✓		字符串移动
0303	CML	DCML	✓		反转传送
0304	BMOV	—	✓		全部传送
0305	NMOV	DNMOV	✓		多点移动

3

API	指令码		P 指令	符号	功能
	16 位	32 位			
0306	XCH	DXCH	✓		数据的交换
0307	BXCH	—	✓		全部交换
0308	SWAP	DSWAP	✓		上/下 BYTE 变换
0309	SMOV	—	✓		位数移动
0310	MOVB	—	✓		多位移动

● 程序跳转指令


API	指令码		P 指令	符号	功能
	16 位	32 位			
0400	CJ	—	✓		条件跳转
0401	JMP	—	—		无条件跳转
0402	GOEND	—	—		跳转到 END

● 程序执行控制指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
0500	DI	—	—		中断插入禁止

API	指令码		P 指令	符号	功能
	16 位	32 位			
0501	EI	—	—		中断插入允许
0502	IMASK	—	—		中断控制

● I/O 更新指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
0600	REF	—	✓		I/O 更新处理

● 便利指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
0700	ALT	—	✓		ON/OFF 交替
0701	TTMR	—	—		教导式定时器
0702	STMR	—	—		特殊定时器
0703	RAMP	—	—		倾斜信号
0704	MTR	—	—		矩阵输入
0705	ABSD	DABSD	—		绝对方式凸轮控制
0706	INCD	—	—		相对方式凸轮控制
0707	PID	DPID	—		PID 运算

3

3

API	指令码		P 指令	符号	功能
	16 位	32 位			
0708	—	DPIDE	—	<div style="border: 1px solid black; padding: 5px;"> DPIDE En PID_RUN MV SV PV PID_MODE PID_MAN MOUT_AUTO CYCLE Kc_Kp Ti_Ki Td_Kd Tf PID_BQ PID_DE PID_DIR ERR_DBW MV_MAX MV_MIN MOUT BIAS LMV </div>	PID 运算

● 逻辑操作指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
0800	WAND	DAND	✓	<div style="display: flex; flex-wrap: wrap;"> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> WAND En S1 D S2 </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> WANDP En S1 D S2 </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> DANDP En S1 D S2 </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> DAND En S1 D S2 </div> </div>	逻辑及(AND)运算
0801	MAND	—	✓	<div style="display: flex; flex-wrap: wrap;"> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> MAND En S1 D S2 N </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> MANDP En S1 D S2 N </div> </div>	矩阵及(AND)运算
0802	WOR	DOR	✓	<div style="display: flex; flex-wrap: wrap;"> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> WOR En S1 D S2 </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> WORP En S1 D S2 </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> DOR En S1 D S2 </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> DORP En S1 D S2 </div> </div>	逻辑或(OR)运算
0803	MOR	—	✓	<div style="display: flex; flex-wrap: wrap;"> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> MOR En S1 D S2 N </div> <div style="border: 1px solid black; padding: 2px; margin: 2px;"> MORP En S1 D S2 N </div> </div>	矩阵或(OR)运算

API	指令码		P 指令	符号	功能
	16 位	32 位			
0804	WXOR	DXOR	✓		逻辑异或(XOR)运算
0805	MXOR	—	✓		矩阵异或(XOR)运算
0806	WXNR	DXNR	✓		逻辑同或(XNR)运算
0807	MXNR	—	✓		矩阵同或(XNR)运算
0809	LD&	DLD&	—		S1&S2
0810	LD	DLD	—		S1 S2
0811	LD^	DLD^	—		S1^S2
0812	AND&	DAND&	—		S1&S2
0813	AND	DAND	—		S1 S2
0814	AND^	DAND^	—		S1^S2
0815	OR&	DOR&	—		S1&S2

API	指令码		P 指令	符号	功能
	16 位	32 位			
0816	OR	DOR	—		S1 S2
0817	OR^	DOR^	—		S1^S2

3

● 循环指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
0900	ROR	DROR	✓		右循环
0901	RCR	DRCR	✓		附进位标志右循环
0902	ROL	DROL	✓		欲循环之装置
0903	RCL	DRCL	✓		附进位标志左循环
0904	MBR	—	✓		矩阵位循环

● 基本指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1000	RST	—	—	元件 —(R)	接点或寄存器清除
1001	TMR	—	—		16 位定时器
1002	TMRH	—	—		16 位定时器
1003	CNT	—	—		16 位计数器
1004	—	DCNT	—		32 位计数器

3

● 移位指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1100	SFTR	—	✓		位右移
1101	SFTL	—	✓		位左移
1102	WSFR	—	✓		寄存器右移
1103	WSFL	—	✓		寄存器左移
1104	SFWR	—	✓		位移写入

3

API	指令码		P 指令	符号	功能
	16 位	32 位			
1105	SFRD	—	✓		位移读出
1106	SFPO	—	✓		读出数据串行最新数据
1107	SFDEL	—	✓		删除数据串行中的数据
1108	SFINS	—	✓		插入数据到数据串行中
1109	MBS	—	✓		矩阵位位移
1110	SFR	—	✓		16 位寄存器位右移
1111	SFL	—	✓		16 位寄存器位左移
1112	BSFR	—	✓		n 个位右移 1 个位
1113	BSFL	—	✓		n 个位左移 1 个位
1114	NSFR	—	✓		n 个寄存器右移
1115	NSFL	—	✓		n 个寄存器左移

● 位处理指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1200	SER	DSER	✓		多点比较
1201	SUM	DSUM	✓		ON 位数量
1202	DECO	—	✓		译码器
1203	ENCO	—	✓		编码器
1204	SEGD	—	✓		7 段显示器解码
1205	SORT	DSORT	-		数据排序
1206	ZRST	—	✓		区域清除
1207	BON	DBON	✓		ON 位判定


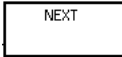

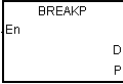
3

API	指令码		P 指令	符号	功能
	16 位	32 位			
1208	MEAN	DMEAN	✓		平均值
1209	CCD	—	✓		总和检查
1210	ABS	DABS	✓		绝对值
1211	MINV	—	✓		矩阵反相
1212	MBRD	—	✓		矩阵位读出
1213	MBWR	—	✓		矩阵位写入
1214	MBC	—	✓		矩阵位状态计数
1215	DIS	—	✓		16 位数据的 4 位分组
1216	UNI	—	✓		16 位数据的 4 位链接

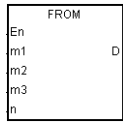
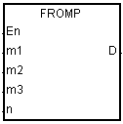

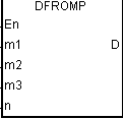
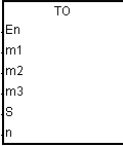
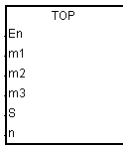
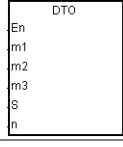
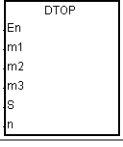
API	指令码		P 指令	符号	功能
	16 位	32 位			
1217	WSUM	DWSUM	✓		16 位数据的总和计算
1218	BSET	—	✓		字符件的位设定
1219	BRST	—	✓		字符件的位复位
1220	BKRST	—	✓		指定区域清除
1221	LIMIT	DLIMIT	✓		BIN 16 位和 BIN 32 位数据的高低限控制
1222	BAND	DBAND	✓		BIN 16 位和 32 位死区控制
1223	ZONE	DZONE	✓		BIN 16 位和 32 位区域控制

3

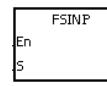



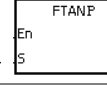
● 结构建立指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1300	FOR	—	—		循环回路起始
1301	NEXT	—	—		循环回路结束
1302	BREAK	—	✓	 	强制结束 FOR-NEXT 循环

● 模块的数据读/写指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1400	FROM	DFROM	✓	   	特殊模块 CR 数据读出
1401	TO	DTO	✓	   	特殊模块 CR 数据写入

● 浮点数指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1500	—	FSIN	✓	 	二进浮点数 SIN 运算
1501	—	FCOS	✓	 	二进浮点数 COS 运算
1502	—	FTAN	✓	 	二进浮点数 TAN 运算

API	指令码		P 指令	符号		功能
	16 位	32 位				
1503	—	FASIN	✓			二进浮点数 ASIN 运算
1504	—	FACOS	✓			二进浮点数 ACOS 运算
1505	—	FATAN	✓			二进浮点数 ATAN 运算
1506	—	FSINH	✓			二进浮点数 SINH 运算
1507	—	FCOSH	✓			二进浮点数 COSH 运算
1508	—	FTANH	✓			二进浮点数 TANH 运算
1509	—	FRAD	✓			角度→弧度
1510	—	FDEG	✓			弧度→角度
1511	SQR	DSQR	✓			BIN 开平方根
1512	—	FSQR	✓			浮点数开平方根
1513	—	FEXP	✓			浮点数取指数
1514	—	FLOG	✓			浮点数取对数
1515	—	FLN	✓			二进浮点数取自然对数
1516	—	FPOW	✓			浮点数权值指令
1517	RAND	—	✓			随机数值

3

API	指令码		P 指令	符号		功能
	16 位	32 位				
1518	BSQR	DBSQR	✓			BCD 开平方根
1519	—	BSIN	✓			BCD SIN 运算
1520	—	BCOS	✓			BCD COS 运算
1521	—	BTAN	✓			BCD TAN 运算
1522	—	BASIN	✓			BCD ASIN 运算
1523	—	BACOS	✓			BCD ACOS 运算
1524	—	BATAN	✓			BCD ATAN 运算

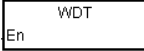
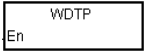


● 万年历指令

API	指令码		P 指令	符号		功能
	16 位	32 位				
1600	TRD	—	✓			万年历数据读出
1601	TWR	—	✓			万年历数据写入
1602	T+	—	✓			万年历数据加算
1603	T-	—	✓			万年历数据减算
1604	HOUR	DHOUR	—			运转定时器
1605	TCMP	—	✓			万年历数据比较

3

API	指令码		P 指令	符号	功能
	16 位	32 位			
1803	STOP	—	—		变频器停止指令
1804	RDST	—	—		变频器状态读取
1805	RSTEF	—	—		变频器异常重置
1806	LRC	—	✓		和检查 LRC 模式
1807	CRC	—	✓		和检查 CRC 模式
1808	MODRW	—	—		MODBUS 数据读写
1810	WRITE	—	—		远程装置数据写入
1811	RPASS	—	—		转送报文至远程装置
1812	COMRS	—	—		通讯传送与接收指令

● 其它指令

API	指令码		P 指令	符号		功能
	16 位	32 位				
1900	WDT	—	✓			逾时监视定时器
1901	DELAY	—	✓			延迟指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
1902	GPWM	—	—	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> GPWM En S1 S2 S2 </div>	一般用脉宽调变
1903	TIMCHK	—	—	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> TIMCHK En S1 D S2 </div>	时间检查
1904	EPUSH	—	✓	<div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> EPUSH En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px;"> EPUSHP En S D </div>	指针寄存器存入
1905	EPOP	—	✓	<div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> EPOP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px;"> EPOPP En S D </div>	指针寄存器读出

● 字符串处理指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
2100	BINDA	DBINDA	✓	<div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> BINDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> BINDAP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> DBINDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px;"> DBINDAP En S D </div>	有号数十进制→ASCII 变换
2101	BINHA	DBINHA	✓	<div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> BINHA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> BINHAP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> DBINHA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px;"> DBINHAP En S D </div>	BIN 十六进制→十六进制 ASCII 变换
2102	BCDDA	DBCDDA	✓	<div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> BCDDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> BCDDAP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> DBCDDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px;"> DBCDDAP En S D </div>	BCD→ASCII 变换
2103	DABIN	DDABIN	✓	<div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> DABIN En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> DABINP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px; margin-right: 10px;"> DDABIN En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 5px;"> DDABINP En S D </div>	有号数十进制 ASCII→有 号数十进制 BIN 变换

3

API	指令码		P 指令	符号		功能																																				
	16 位	32 位																																								
2104	HABIN	DHABIN	✓	<table border="1"> <tr><td>En</td><td>HABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DHABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	HABIN	D	S			En	DHABIN	D	S			<table border="1"> <tr><td>En</td><td>HABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DHABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	HABINP	D	S			En	DHABINP	D	S			十六进制 ASCII→十六进制 BIN 变换												
En	HABIN	D																																								
S																																										
En	DHABIN	D																																								
S																																										
En	HABINP	D																																								
S																																										
En	DHABINP	D																																								
S																																										
2105	DABCD	DDABCD	✓	<table border="1"> <tr><td>En</td><td>DABCD</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABCD</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	DABCD	D	S			En	DDABCD	D	S			<table border="1"> <tr><td>En</td><td>DABCDP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABCDP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	DABCDP	D	S			En	DDABCDP	D	S			ASCII→BCD 变换												
En	DABCD	D																																								
S																																										
En	DDABCD	D																																								
S																																										
En	DABCDP	D																																								
S																																										
En	DDABCDP	D																																								
S																																										
2106	\$LEN	—	✓	<table border="1"> <tr><td>En</td><td>\$LEN</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$LEN	D	S			<table border="1"> <tr><td>En</td><td>\$LENP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$LENP	D	S			计算字符串长度																								
En	\$LEN	D																																								
S																																										
En	\$LENP	D																																								
S																																										
2107	\$STR	\$DSTR	✓	<table border="1"> <tr><td>En</td><td>\$STR</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>D\$STR</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$STR	D	S1			S2			En	D\$STR	D	S1			S2			<table border="1"> <tr><td>En</td><td>\$STRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>D\$STRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$STRP	D	S1			S2			En	D\$STRP	D	S1			S2			BIN→String
En	\$STR	D																																								
S1																																										
S2																																										
En	D\$STR	D																																								
S1																																										
S2																																										
En	\$STRP	D																																								
S1																																										
S2																																										
En	D\$STRP	D																																								
S1																																										
S2																																										
2108	\$VAL	\$DVAL	✓	<table border="1"> <tr><td>En</td><td>\$VAL</td><td>D1</td></tr> <tr><td>S</td><td></td><td>D2</td></tr> </table> <table border="1"> <tr><td>En</td><td>D\$VAL</td><td>D1</td></tr> <tr><td>S</td><td></td><td>D2</td></tr> </table>	En	\$VAL	D1	S		D2	En	D\$VAL	D1	S		D2	<table border="1"> <tr><td>En</td><td>\$VALP</td><td>D1</td></tr> <tr><td>S</td><td></td><td>D2</td></tr> </table> <table border="1"> <tr><td>En</td><td>D\$VALP</td><td>D1</td></tr> <tr><td>S</td><td></td><td>D2</td></tr> </table>	En	\$VALP	D1	S		D2	En	D\$VALP	D1	S		D2	String→BIN												
En	\$VAL	D1																																								
S		D2																																								
En	D\$VAL	D1																																								
S		D2																																								
En	\$VALP	D1																																								
S		D2																																								
En	D\$VALP	D1																																								
S		D2																																								
2109	\$FSTR	—	✓	<table border="1"> <tr><td>En</td><td>\$FSTR</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$FSTR	D	S1			S2			<table border="1"> <tr><td>En</td><td>\$FSTRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$FSTRP	D	S1			S2			Float→String																		
En	\$FSTR	D																																								
S1																																										
S2																																										
En	\$FSTRP	D																																								
S1																																										
S2																																										
2110	\$FVAL	—	✓	<table border="1"> <tr><td>En</td><td>\$FVAL</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$FVAL	D	S			<table border="1"> <tr><td>En</td><td>\$FVALP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$FVALP	D	S			String→Float																								
En	\$FVAL	D																																								
S																																										
En	\$FVALP	D																																								
S																																										
2111	\$RIGHT	—	✓	<table border="1"> <tr><td>En</td><td>\$RIGHT</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table>	En	\$RIGHT	D	S			n			<table border="1"> <tr><td>En</td><td>\$RIGHTP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table>	En	\$RIGHTP	D	S			n			从右边截取字符串																		
En	\$RIGHT	D																																								
S																																										
n																																										
En	\$RIGHTP	D																																								
S																																										
n																																										
2112	\$LEFT	—	✓	<table border="1"> <tr><td>En</td><td>\$LEFT</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table>	En	\$LEFT	D	S			n			<table border="1"> <tr><td>En</td><td>\$LEFTP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table>	En	\$LEFTP	D	S			n			从左边截取字符串																		
En	\$LEFT	D																																								
S																																										
n																																										
En	\$LEFTP	D																																								
S																																										
n																																										
2113	\$MIDR	—	✓	<table border="1"> <tr><td>En</td><td>\$MIDR</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$MIDR	D	S1			S2			<table border="1"> <tr><td>En</td><td>\$MIDRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$MIDRP	D	S1			S2			区段截取字符串																		
En	\$MIDR	D																																								
S1																																										
S2																																										
En	\$MIDRP	D																																								
S1																																										
S2																																										
2114	\$MIDW	—	✓	<table border="1"> <tr><td>En</td><td>\$MIDW</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$MIDW	D	S1			S2			<table border="1"> <tr><td>En</td><td>\$MIDWP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$MIDWP	D	S1			S2			区段字符串取代																		
En	\$MIDW	D																																								
S1																																										
S2																																										
En	\$MIDWP	D																																								
S1																																										
S2																																										
2115	\$SER	—	✓	<table border="1"> <tr><td>En</td><td>\$SER</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>N</td><td></td><td></td></tr> </table>	En	\$SER	D	S1			S2			N			<table border="1"> <tr><td>En</td><td>\$SERP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>N</td><td></td><td></td></tr> </table>	En	\$SERP	D	S1			S2			N			字符串搜寻												
En	\$SER	D																																								
S1																																										
S2																																										
N																																										
En	\$SERP	D																																								
S1																																										
S2																																										
N																																										

API	指令码		P 指令	符号		功能
	16 位	32 位				
2116	\$RPLC	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$RPLC En S1 D S2 S3 S4 </div>	<div style="border: 1px solid black; padding: 2px;"> \$RPLCP En S1 D S2 S3 S4 </div>	字符串取代
2117	\$DEL	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$DEL En S1 D S2 S3 </div>	<div style="border: 1px solid black; padding: 2px;"> \$DELP En S1 D S2 S3 </div>	指定字符串删除
2118	\$CLR	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$CLR En S </div>	<div style="border: 1px solid black; padding: 2px;"> \$CLRCP En S </div>	字符串清除
2119	\$INS	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$INS En S1 D S2 S3 </div>	<div style="border: 1px solid black; padding: 2px;"> \$INSP En S1 D S2 S3 </div>	字符串插入
2120	\$FMOD	—	✓	<div style="border: 1px solid black; padding: 2px;"> FMOD En S1 D S2 </div>	<div style="border: 1px solid black; padding: 2px;"> FMODP En S1 D S2 </div>	浮点数转 BCD 浮点数
2121	\$FREXP	—	✓	<div style="border: 1px solid black; padding: 2px;"> FREXP En S1 D S2 </div>	<div style="border: 1px solid black; padding: 2px;"> FREXPP En S1 D S2 </div>	BCD 浮点数转浮点数

3

● 以太网控制指令

API	指令码		P 指令	符号		功能
	16 位	32 位				
2200	SOPEN	—	✓	<div style="border: 1px solid black; padding: 2px;"> SOPEN En S1 S2 S3 </div>	<div style="border: 1px solid black; padding: 2px;"> SOPENP En S1 S2 S3 </div>	开启 Socket
2201	SSEND	—	✓	<div style="border: 1px solid black; padding: 2px;"> SSEND En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px;"> SSENDP En S1 S2 </div>	透过已开启的 Socket 传送数据
2202	SRCVD	—	✓	<div style="border: 1px solid black; padding: 2px;"> SRCVD En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px;"> SRCVDP En S1 S2 </div>	透过已开启的 Socket 接收数据
2203	SCLOSE	—	✓	<div style="border: 1px solid black; padding: 2px;"> SCLOSE En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px;"> SCLOSEP En S1 S2 </div>	关闭 Socket
2204	MSEND	—	✓	<div style="border: 1px solid black; padding: 2px;"> MSEND En S1 D S2 S3 </div>	<div style="border: 1px solid black; padding: 2px;"> MSENDP En S1 D S2 S3 </div>	寄送电子邮件

3

API	指令码		P 指令	符号		功能
	16 位	32 位				
2205	EMDRW	—	✓			MODBUS TCP 数据读写
2206	—	DINTOA	✓			IP→字符串变换
2207	—	DIATON	✓			字符串→IP 变换

● 存储卡读写指令

API	指令码		P 指令	符号		功能
	16 位	32 位				
2300	MWRIT	—	✓			写入存储卡数据
2301	MREAD	—	✓			读取存储卡数据
2302	MTWRIT	—	✓			写入字符串至存储卡

● 任务控制指令

API	指令码		P 指令	符号		功能
	16 位	32 位				
2400	TKON	—	✓			Cyclic 任务 Task 启动
2401	TKOFF	—	✓			Cyclic 任务 Task 关闭

● SFC 控制指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
2500	SFCRUN	—	—	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SFCRUN En S1 S2 S3 </div>	SFC 启动
2501	SFCPSE	—	—	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SFCPSE En S1 S2 </div>	SFC 暂停
2502	SFCSTP	—	—	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SFCSTP En S </div>	SFC 停止

● 冗余系统控制指令

API	指令码		P 指令	符号	功能
	16 位	32 位			
2900	SSO	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SSOP En D </div>	冗余系统切换
2901	RCS	—	—	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> RCS En S D </div>	读取/设定冗余系统信息

3

3.4.2 应用指令 (依英文字母排序)

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
符号	0101	—	D-	—	✓	BIN 减法
	0114	\$+	—	—	✓	字符串链接
	2118	\$CLR	—	—	✓	字符串清除
	2117	\$DEL	—	—	✓	指定字符串删除
	2109	\$FSTR	—	—	✓	Float→String
	2110	\$FVAL	—	—	✓	String→Float
	2119	\$INS	—	—	✓	字符串插入
	2112	\$LEFT	—	—	✓	从左边截取字符串
	2106	\$LEN	—	—	✓	计算字符串长度
符号	2113	\$MIDR	—	—	✓	区段截取字符串
	2114	\$MIDW	—	—	✓	区段字符串取代
	0302	\$MOV	—	—	✓	字符串移动
	2111	\$RIGHT	—	—	✓	从右边截取字符串
	2116	\$RPLC	—	—	✓	字符串取代
	2115	\$SER	—	—	✓	字符串搜寻
	2107	\$STR	D\$STR	—	✓	BIN→String
	2108	\$VAL	D\$VAL	—	✓	String→BIN
	0102	*	D*	—	✓	BIN 乘法
	0103	/	D/	—	✓	BIN 除法
	0100	+	D+	—	✓	BIN 加法
A	1210	ABS	DABS	—	✓	绝对值
	0705	ABSD	DABSD	—	—	绝对方式凸轮控制
	0700	ALT	—	—	✓	ON/OFF 交替
	0046	AND\$<	—	—	—	字符串类型比较 S1 < S2
	0047	AND\$<=	—	—	—	字符串类型比较 S1 ≤ S2
	0043	AND\$<>	—	—	—	字符串类型比较 S1 ≠ S2
	0042	AND\$=	—	—	—	字符串类型比较 S1 = S2
	0044	AND\$>	—	—	—	字符串类型比较 S1 > S2
	0045	AND\$>=	—	—	—	字符串类型比较 S1 ≥ S2
	0812	AND&	DAND&	—	—	接点类型逻辑运算 S1&S2
	0814	AND^	DAND^	—	—	接点类型逻辑运算 S1^S2
	0813	AND	DAND	—	—	接点类型逻辑运算 S1 S2
	0010	AND<	DAND<	—	—	接点类型比较 S1 < S2

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
	0011	AND<=	DAND<=	—	—	接点类型比较 $S1 \leq S2$
	0007	AND<>	DAND<>	—	—	接点类型比较 $S1 \neq S2$
	0006	AND=	DAND=	—	—	接点类型比较 $S1 = S2$
	0008	AND>	DAND>	—	—	接点类型比较 $S1 > S$
	0009	AND>=	DAND>=	—	—	接点类型比较 $S1 \geq S2$
	-	ANED	—	—	—	下降沿检出动作串联连接
	-	APED	—	—	—	上升沿检出动作串联连接
	1703	ARWS	—	—	—	箭头键盘输入
B	0109	B-	DB-	—	✓	BCD 减法
	0110	B*	DB*	—	✓	BCD 乘法
	0111	B/	DB/	—	✓	BCD 除法
B	0108	B+	DB+	—	✓	BCD 加法
	1523	BACOS	—	—	✓	BCD ACOS 运算
	1222	BAND	DBAND	—	✓	BIN 16 位和 32 位死区控制
	1522	BASIN	—	—	✓	BCD ASIN 运算
	1524	BATAN	—	—	✓	BCD ATAN 运算
	0200	BCD	DBCD	—	✓	BIN→BCD 变换
	2102	BCDDA	DBCDDA	—	✓	BCD→ASCII 变换
	1520	BCOS	—	—	✓	BCD COS 运算
	0201	BIN	DBIN	—	✓	BCD→BIN 变换
	2100	BINDA	DBINDA	—	✓	有号数十进制→ASCII 变换
	2101	BINHA	DBINHA	—	✓	BIN 十六进制→十六进制 ASCII 变换
	0113	BK-	—	—	✓	连续区块 BIN 减法
	0112	BK+	—	—	✓	连续区块 BIN 加法
	0214	BKBCD	—	—	✓	连续区块 BIN→BCD 变换
	0215	BKBIN	—	—	✓	连续区块 BCD→BIN 变换
	1220	BKRST	—	—	✓	指定区域清除
	0304	BMOV	—	—	✓	全部传送
	1207	BON	DBON	—	✓	ON 位判定
	1302	BREAK	—	—	✓	强制结束 FOR-NEXT 循环
	1219	BRST	—	—	✓	字软组件的位复位
	1218	BSET	—	—	✓	字软组件的位设定
	1113	BSFL	—	—	✓	n 个位左移 1 个位
	1112	BSFR	—	—	✓	n 个位右移 1 个位
1519	BSIN	—	—	✓	BCD SIN 运算	

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
	1518	BSQR	DBSQR	—	✓	BCD 开平方根
	1521	BTAN	—	—	✓	BCD TAN 运算
	0307	BXCH	—	—	✓	全部交换
C	1209	CCD	—	—	✓	总和检查
	0065	CHKADR	—	—	—	接点类型指针寄存器地址检查
	0400	CJ	—	—	✓	条件跳转
	1812	COMRS	—	—	—	通讯传送与接收指令
	0303	CML	DCML	—	✓	反转传送
	0054	CMP	DCMP	—	✓	比较设定输出
	0063	CMPT<	—	—	✓	表格比较<
C	0064	CMPT<=	—	—	✓	表格比较<=
	0060	CMPT<>	—	—	✓	表格比较<>
	0059	CMPT=	—	—	✓	表格比较=
	0061	CMPT>	—	—	✓	表格比较>
	0062	CMPT>=	—	—	✓	表格比较>=
	1003	CNT	—	—	—	16 位计数器
	0219	COLM	DCOLM	—	✓	LINE to COLUMN
1807	CRC	—	—	—	和检查 CRC 模式	
D	2105	DABCD	DDABCD	—	✓	ASCII→BCD 变换
	2103	DABIN	DDABIN	—	✓	有号数十进制 ASCII→有号数十进制 BIN 变换
	1004	DCNT	—	—	—	32 位计数器
	0116	DEC	DDEC	—	✓	BIN 减一
	1202	DECO	—	—	✓	译码器
	1901	DELAY	—	—	✓	延迟指令
	0301	—	—	DFMOV	✓	64 位浮点数数据移动
	0500	DI	—	—	—	中断插入禁止
	2207	DIATON	—	—	✓	字符串→IP 变换
	2206	DINTOA	—	—	✓	IP→字符串变换
	1215	DIS	—	—	✓	16 位数据的 4 位分组
	0118	DIV16	DIV32	—	✓	16 位专用 BIN 除法/ 32 位专用 BIN 除法
	0708	—	DPIDE	—	—	PID 运算
1702	DSW	—	—	—	指拨开关输入	

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
E	0501	EI	—	—	—	中断插入允许
	2205	EMDRW	—	—	✓	MODBUS TCP 数据读写
	1203	ENCO	—	—	✓	编码器
	1905	EPOP	—	—	✓	指针寄存器读出
	1904	EPUSH	—	—	✓	指针寄存器存入
F	0105	—	F-	DF-	✓	浮点数减法
	0106	—	F*	DF*	✓	浮点数乘法
	0107	—	F/	DF/	✓	浮点数除法
	0104	—	F+	DF+	✓	浮点数加法
	1504	—	FACOS	—	✓	浮点数 ACOS 运算
	0028	—	FAND<	DFAND<	—	浮点数接点类型比较 $S1 < S2$
	0029	—	FAND<=	DFAND<=	—	浮点数接点类型比较 $S1 \leq S2$
	0025	—	FAND<>	DFAND<>	—	浮点数接点类型比较 $S1 \neq S2$
	0024	—	FAND=	DFAND=	—	浮点数接点类型比较 $S1 = S2$
F	0026	—	FAND>	DFAND>	—	浮点数接点类型比较 $S1 > S$
	0027	—	FAND>=	DFAND>=	—	浮点数接点类型比较 $S1 \geq S2$
	1503	—	FASIN	—	✓	浮点数 ASIN 运算
	1505	—	FATAN	—	✓	浮点数 ATAN 运算
	0212	—	FBCD	—	✓	浮点数→十进浮点数
	0213	—	FBIN	—	✓	十进浮点数→浮点数
	0056	—	FCMP	—	✓	浮点数比较
	1501	—	FCOS	—	✓	浮点数 COS 运算
	1507	—	FCOSH	—	✓	浮点数 COSH 运算
	1510	—	FDEG	—	✓	弧度→角度
	1513	—	FEXP	—	✓	浮点数取指数
	0205	—	FINT	DFINT	✓	64 位浮点数→BIN 整数变换
	0022	—	FLD<	DFLD<	—	浮点数接点类型比较 $S1 < S2$
	0023	—	FLD<=	DFLD<=	—	浮点数接点类型比较 $S1 \leq S2$
	0019	—	FLD<>	DFLD<>	—	浮点数接点类型比较 $S1 \neq S2$
	0018	—	FLD=	DFLD=	—	浮点数接点类型比较 $S1 = S2$
	0020	—	FLD>	DFLD>	—	浮点数接点类型比较 $S1 > S$
	0021	—	FLD>=	DFLD>=	—	浮点数接点类型比较 $S1 \geq S2$
	1515	—	FLN	—	✓	浮点数取自然对数
	1514	—	FLOG	—	✓	浮点数取对数
0202	FLT	DFLT	—	✓	BIN 整数→浮点数变换	

3

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
	0203	FLTD	DFLTD	—	✓	BIN 整数→64 位浮点数变换
	2120	FMOD	—	—	✓	浮点数转 BCD 浮点数
	0211	FNEG	—	—	✓	32 位浮点正负符号反相
	1300	FOR	—	—	—	循环回路起始
	0034	—	FOR<	DFOR<	—	浮点数接点类型比较 $S1 < S2$
	0035	—	FOR<=	DFOR<=	—	浮点数接点类型比较 $S1 \leq S2$
	0031	—	FOR<>	DFOR<>	—	浮点数接点类型比较 $S1 \neq S2$
	0030	—	FOR=	DFOR=	—	浮点数接点类型比较 $S1 = S2$
	0032	—	FOR>	DFOR>	—	浮点数接点类型比较 $S1 > S$
	0033	—	FOR>=	DFOR>=	—	浮点数接点类型比较 $S1 \geq S2$
	1516	—	FPOW	—	✓	浮点数权值指令
	1509	—	FRAD	—	✓	角度→弧度
	2121	FREXP	—	—	✓	BCD 浮点数转浮点数
	1400	FROM	DFROM	—	✓	特殊模块 CR 数据读出
F	1500	—	FSIN	—	✓	浮点数 SIN 运算
	1506	—	FSINH	—	✓	浮点数 SINH 运算
	1512	—	FSQR	—	✓	浮点数开平方根
	1502	—	FTAN	—	✓	浮点数 TAN 运算
	1508	—	FTANH	—	✓	浮点数 TANH 运算
	1801	FWD	—	—	—	VFD-A 变频器正转指令
	0057	—	FZCP	—	✓	浮点数区域比较
G	0209	GBIN	DGBIN	—	✓	GRY 码→BIN 变换
	0402	GOEND	—	—	—	跳转到 END
	1902	GPWM	—	—	—	一般用脉宽调变
	0208	GRY	DGRY	—	✓	BIN→GRY 码变换
H	2104	HABIN	DHABIN	—	✓	十六进制 ASCII→十六进制 BIN 变换
	1701	HKY	DHKY	—	—	16 键键盘输入
	1604	HOUR	DHOUR	—	—	运转定时器
I	0502	IMASK	—	—	—	中断控制
	0115	INC	DINC	—	✓	BIN 加一
	0706	INCD	—	—	—	相对方式凸轮控制
	0204	INT	DINT	—	✓	浮点数→BIN 整数变换
J	0401	JMP	—	—	—	无条件跳转
L	0040	LD\$<	—	—	—	字符串类型比较 $S1 < S2$
	0041	LD\$<=	—	—	—	字符串类型比较 $S1 \leq S2$

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
	0037	LD\$<>	—	—	—	字符串类型比较 S1≠S2
	0036	LD\$=	—	—	—	字符串类型比较 S1 = S2
	0038	LD\$>	—	—	—	字符串类型比较 S1 > S2
	0039	LD\$>=	—	—	—	字符串类型比较 S1 ≥ S2
	0809	LD&	DLD&	—	—	接点类型逻辑运算 S1&S2
	0811	LD^	DLD^	—	—	接点类型逻辑运算 S1^S2
	0810	LD	DLD	—	—	接点类型逻辑运算 S1 S2
	0004	LD<	DLD<	—	—	接点类型比较 S1 < S2
	0005	LD<=	DLD<=	—	—	接点类型比较 S1 ≤ S2
	0001	LD<>	DLD<>	—	—	接点类型比较 S1≠S2
	0000	LD=	DLD=	—	—	接点类型比较 S1 = S2
	0002	LD>	DLD>	—	—	接点类型比较 S1 > S
	0003	LD>=	DLD>=	—	—	接点类型比较 S1 ≥ S2
	L	1221	LIMIT	DLIMIT	—	✓
0218		LINE	DLINE	—	✓	COLUMN to LINE
1806		LRC	—	—	—	和检查 LRC 模式
M	0801	MAND	—	—	✓	矩阵及(AND)运算
	1214	MBC	—	—	✓	矩阵位状态计数
	0904	MBR	—	—	✓	矩阵位循环
	1212	MBRD	—	—	✓	矩阵位读出
	1109	MBS	—	—	✓	矩阵位位移
	1213	MBWR	—	—	✓	矩阵位写入
	0058	MCMP	—	—	✓	矩阵比较
	1208	MEAN	DMEAN	—	✓	平均值
	1211	MINV	—	—	✓	矩阵反相
	0206	MMOV	—	—	✓	16→32 位数值转换
	1808	MODRW	—	—	—	MODBUS 数据读写
	0803	MOR	—	—	✓	矩阵或(OR)运算
	0300	MOV	DMOV	—	✓	数据移动
	0310	MOVB	—	—	✓	多位移动
	2301	MREAD	—	—	—	读取存储卡数据
	2204	MSEND	—	—	✓	寄送电子邮件
	0704	MTR	—	—	—	矩阵输入
	2302	MTWRIT	—	—	—	写入字符串至存储卡
0117	MUL16	MUL32	—	✓	16 位专用 BIN 乘法/	

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
						32 位专用 BIN 乘法
	2300	MWRIT	—	—	—	写入存储卡数据
	0807	MXNR	—	—	✓	矩阵同或(XNR)运算
	0805	MXOR	—	—	✓	矩阵异或(XOR)运算
N	-	NED	—	—	—	下降沿检出动作开始连接
	0210	NEG	DNEG	—	✓	取负数(取 2 的补码)
	1301	NEXT	—	—	—	循环回路结束
	0305	NMOV	DNMOV	—	✓	多点移动
	1115	NSFL	—	—	✓	n 个寄存器左移
	1114	NSFR	—	—	✓	n 个寄存器右移
O	-	ONED	—	—	—	下降沿检出动作并联连接
	-	OPED	—	—	—	上升沿检出动作并联连接
	0052	OR\$<	—	—	—	字符串类型比较 S1 < S2
	0053	OR\$<=	—	—	—	字符串类型比较 S1 ≤ S2
	0049	OR\$<>	—	—	—	字符串类型比较 S1 ≠ S2
O	0048	OR\$=	—	—	—	字符串类型比较 S1 = S2
	0050	OR\$>	—	—	—	字符串类型比较 S1 > S2
	0051	OR\$>=	—	—	—	字符串类型比较 S1 ≥ S2
	0815	OR&	DOR&	—	—	接点类型逻辑运算 S1&S2
	0817	OR^	DOR^	—	—	接点类型逻辑运算 S1^S2
	0816	OR	DOR	—	—	接点类型逻辑运算 S1 S2
	0016	OR<	DOR<	—	—	接点类型比较 S1 < S2
	0017	OR<=	DOR<=	—	—	接点类型比较 S1 ≤ S2
	0013	OR<>	DOR<>	—	—	接点类型比较 S1 ≠ S2
	0012	OR=	DOR=	—	—	接点类型比较 S1 = S2
	0014	OR>	DOR>	—	—	接点类型比较 S1 > S
	0015	OR>=	DOR>=	—	—	接点类型比较 S1 ≥ S2
P	-	PED	—	—	—	上升沿检出动作开始连接
	0707	PID	—	—	—	PID 运算
R	0703	RAMP	—	—	—	倾斜信号
	1517	RAND	—	—	✓	随机数值
	0903	RCL	DRCL	—	✓	附进位标志左循环
	0901	RCR	DRCR	—	✓	附进位标志右循环
	2901	RCS	—	—	—	读取/设定冗余系统信息
	1804	RDST	—	—	—	VFD-A 变频器状态读取
	0600	REF	—	—	✓	I/O 更新处理

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
	1802	REV	—	—	—	VFD-A 变频器反转指令
	0207	RMOV	—	—	✓	32→16 位数值转换
	0902	ROL	DROL	—	✓	左循环
	0900	ROR	DROR	—	✓	右循环
	1811	RPASS	—	—	—	转送报文至远程装置
	1800	RS	—	—	—	串行数据传输
	1000	RST	—	—	—	接点或寄存器清除
	1805	RSTEF	—	—	—	VFD-A 变频器异常重置
S	0216	SCAL	—	—	✓	比例运算
	2203	SCLOSE	—	—	✓	关闭 Socket
	0217	SCLP	DSCLP	—	✓	比例运算
	1204	SEGD	—	—	✓	7 段显示器解码
S	1704	SEGL	—	—	—	七段显示器扫描输出
	1200	SER	DSER	—	✓	多点比较
	2501	SFCPSE	—	—	—	SFC 暂停
	2500	SFCRUN	—	—	—	SFC 启动
	2502	SFCSTP	—	—	—	SFC 停止
	1107	SFDEL	—	—	✓	删除数据串行中的数据
	1108	SFINS	—	—	✓	插入数据到数据串行中
	1111	SFL	—	—	✓	16 位寄存器位左移
	1106	SFPO	—	—	✓	读出数据串行最新数据
	1110	SFR	—	—	✓	16 位寄存器位右移
	1105	SFRD	—	—	✓	位移读出
	1101	SFTL	—	—	✓	位左移
	1100	SFTR	—	—	✓	位右移
	1104	SFWR	—	—	✓	位移写入
	0309	SMOV	—	—	✓	位数移动
	2200	SOPEN	—	—	✓	开启 Socket
	1205	SORT	DSORT	—	—	数据排序
	1511	SQR	DSQR	—	✓	BIN 开平方根
	2202	SRCVD	—	—	✓	透过已开启的 Socket 接收数据
	2201	SSEND	—	—	✓	透过已开启的 Socket 传送数据
	2900	SSO	—	—	✓	冗余系统切换
0702	STMR	—	—	—	特殊定时器	
1803	STOP	—	—	—	VFD-A 变频器停止指令	

分类	API	指令码 (位)			P 指令	功能
		16	32	64		
	1201	SUM	DSUM	—	✓	ON 位数量
	0308	SWAP	DSWAP	—	✓	上/下 BYTE 变换
T	1603	T-	—	—	✓	万年历数据减算
	1602	T+	—	—	✓	万年历数据加算
	1605	TCMP	—	—	✓	万年历数据比较
	1903	TIMCHK	—	—	—	时间检查
	2401	TKOFF	—	—	✓	任务 TKS 关闭
	2400	TKON	—	—	✓	任务 TKS 启动
	1700	TKY	DTKY	—	—	10 键键盘输入
	1001	TMR	—	—	—	16 位定时器
	1002	TMRH	—	—	—	16 位定时器
	1401	TO	DTO	—	✓	特殊模块 CR 数据写入
	1600	TRD	—	—	✓	万年历数据读出
	0701	TTMR	—	—	—	教导式定时器
	1601	TWR	—	—	✓	万年历数据写入
	1606	TZCP	—	—	✓	万年历数据区域比较
	U	1216	UNI	—	—	✓
W	0800	WAND	DAND	—	✓	逻辑及(AND)运算
	1900	WDT	—	—	✓	逾时监视定时器
	0802	WOR	DOR	—	✓	逻辑或(OR)运算
	1810	WRITE	—	—	—	远程装置数据写入
	1103	WSFL	—	—	✓	寄存器左移
	1102	WSFR	—	—	✓	寄存器右移
	1217	WSUM	DWSUM	—	✓	16 位数据的总和计算
	0806	WXNR	DXNR	—	✓	逻辑同或(XNR)运算
	0804	WXOR	DXOR	—	✓	逻辑异或(XOR)运算
X	0306	XCH	DXCH	—	✓	数据的交换
Z	0055	ZCP	DZCP	—	✓	区域比较
	1223	ZONE	DZONE	—	✓	BIN 16 位和 32 位区域控制
	1206	ZRST	—	—	✓	区域清除

4

第4章 指令结构

目录

4.1 API应用指令组成说明	4-2
4.2 指令使用限制.....	4-5
4.3 变址说明	4-7
4.4 指针寄存器 (PR) 说明.....	4-8
4.5 定时器指针寄存器 (TR) 说明.....	4-10
4.6 16 位计数器指针寄存器 (CR) 说明	4-12
4.7 32 位计数器指针寄存器 (HCR) 说明	4-13

4.1 API 应用指令组成说明

PLC 指令提供一个特定的指令码及 API 编号，以便记忆。下面的表格中指令的 API 编号为 0300。指令码为 MOV，MOV 的功能描述是“数据移动”。

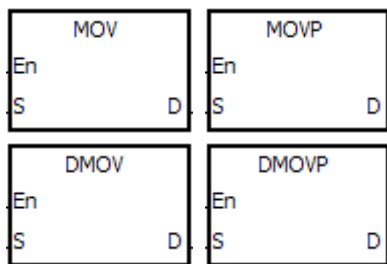
API	指令码			操作数						功能					
0300	D	MOV	P	S · D						数据移动					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

4

符号：



S : 数据来源 Word/Double Word
D : 数据目的地 Word/Double Word

- 操作数区域会列出各种指令所用到的装置。S · D · n · m 会根据它的功能使用于操作数，如果不止一个操作数并且功能相同时，会附加编号使用，如 S₁ · S₂...
- 脉冲执行型指令要求在指令码后面加上“P”，而 32 位指令要求在指令码前面加上“D”，如“D***P”，“***”是指令码。
- 操作数区域中，装置 PR 为变址寄存器，PR 指针使用方式请参考 ISPSOft 使用手册及 4.4 节指针寄存器说明。
- 如果要在功能块中使用指令时，操作数区域中支持装置 T · C · HC，则必须使用 TR · CR · HCR 指针寄存器来运算，详细说明请参考 4.5~4.7 节。
- 操作数区域中，32 位单精度浮点数表示为 F，64 位双精度浮点数表示为 DF。
- 上表中含●的装置支持E修饰，含○的装置不支持E修饰。例如，操作数S₁支持D装置的E修饰。
- 适用机种注明在表格的右下角。详细的指令变化情况，可以对照着表格下方确认指令是否支持脉冲执行型，16 位指令，32 位指令，64 位指令。
- MOV 指令在 ISPSOft 中梯形图显示之方块符号说明如下：

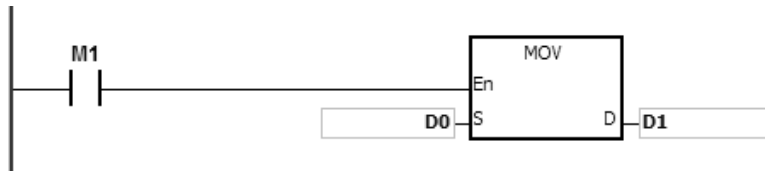
MOV · MOV P · DMOV · DMOV P : 表示指令码
En : 表示允许
S : 数据来源 (适用操作数格式为 Word/Double Word)
D : 数据目的地 (适用操作数格式为 Word/Double Word)

指令组成：

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好几个操作数所组合而成。

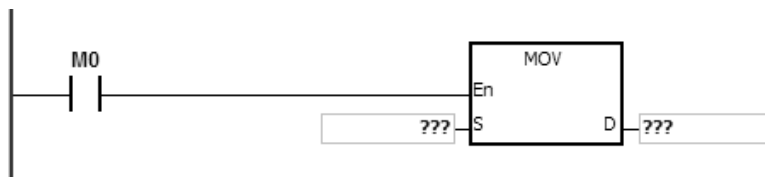
每个应用指令均有其指令编号（API）及专用的名称符号，例如：API0300 的指令码符号为 MOV（数据搬移）。

直接输入法：利用梯形图编辑软件 ISPSOft 作该指令的输入，以 MOV 指令而言只需要直接输入指令的名称与操作数指定“MOV D0 D1”即可：



拖曳输入法：选择 ISPSOft 左侧项目管理区内的函数库，将数据处理分类中“MOV”指令拖曳到梯形图编辑区即可，操作数再另行指定。

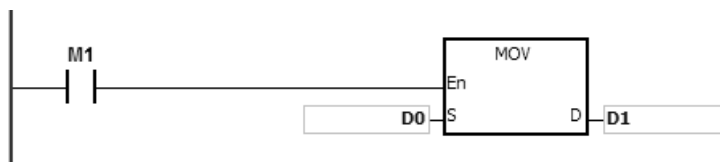
工具列输入法：选择 ISPSOft 工具列的操作数/应用指令/功能块选取按钮，选取函数库类别，再选择数据处理分类中“MOV”指令插入即可，操作数再另行指定。



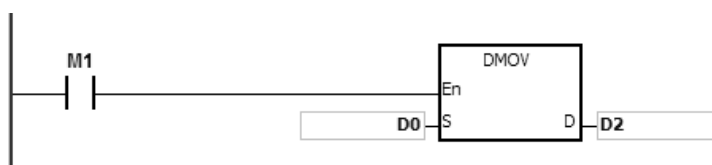
S	来源操作数；若来源操作数有一个以上，那么则以 S_1 、 S_2 ... 分别表示。
D	目的操作数；若目的操作数有一个以上，那么则以 D_1 、 D_2 ... 分别表示。
若操作数只可指定常数K/H或寄存器时，那么则以 m 、 m_1 、 m_2 、 n 、 n_1 、 n_2 表示。	

操作数长度（16 位指令或 32 位指令或浮点数指令）：**16 位指令或 32 位指令**

操作数的数值内容，其长度可分为 16 位及 32 位，因此部份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。

16 位 MOV 指令

当 M1=ON，D0 被传送至 D1

32 位 DMOV 指令

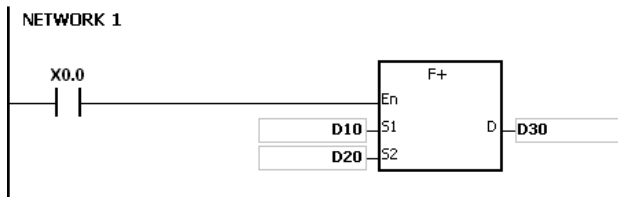
当 M1=ON 时，(D1·D0) 的内容被传送至 (D3·D2)

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。

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 位指令前加上“D”来表示即可。

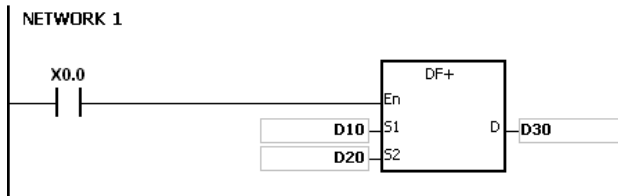
32 位单精度浮点数指令 F+



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)

4

64 位单精度浮点数指令 DF+

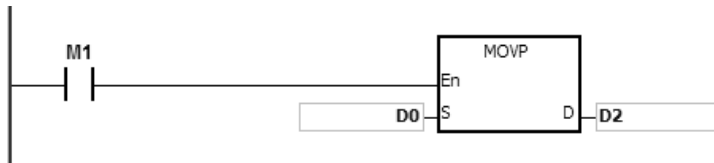


当 X0.0=ON 时，双精度浮点数 (D13 · D12 · D11 · D10) + 双精度浮点数 (D23 · D22 · D21 · D20) 的内容被传送至 (D33 · D32 · D31 · D30)

连续执行型/脉冲执行型：

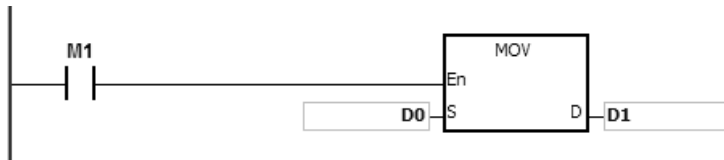
1. 以指令的执行方式来说亦可分成「连续执行型」及「脉冲执行型」2 种。由于指令不被执行时，所需的执行时间比较短，因此程序中尽可能的使用脉冲执行型指令可减少扫描周期。
2. “脉冲”功能可以让相关的指令启动上升沿触发的控制输入。指令在一个扫描周期内被执行 ON。
3. 之后，若控制输入保持为 ON，且关联的指令没有执行，为了重新执行指令，控制输入必须再次从 OFF 到 ON。
4. 脉冲执行型指令。

脉冲执行型



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

连续执行型



于 M1=ON 的每次扫描周期，MOV 指令均被执行一次，因此称之为连续执行型指令

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

操作数的指定对象：

1. 输入继电器：X0.0 ~ X511.15 或 X0 ~ X511
 2. 输出继电器：Y0.0 ~ Y511.15 或 Y0 ~ Y511
 3. 内部继电器：M0 ~ M8191
 4. 步进标志：S0 ~ S2047
 5. 定时器：T0 ~ T2047
 6. 16 位计数器：C0 ~ C2047
 7. 32 位计数器：HC0 ~ HC63
 8. 数据寄存器：D0 ~ D65535 或 D0.0 ~ D65535.15
 9. Link 寄存器：L0 ~ L65535 或 L0.0 ~ D65535.15
 10. 特殊标志：SM0 ~ SM2047
 11. 特殊寄存器：SR0 ~ SR2047
 12. 变址寄存器：E0 ~ E31
 13. 指针寄存器：PR0 ~ PR15
 14. 定时器指针寄存器：TR0 ~ TR7
 15. 16 位计数器指针寄存器：CR0 ~ CR7
 16. 32 位计数器指针寄存器：HCR0 ~ HCR7
 17. 常数：十进制常数 K，十六进制常数 16#
 18. 字符串：“\$”
 19. 浮点数：单精度浮点数 F，双精度位浮点数 DF
 20. 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。
 21. 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 被使用的规则亦相同。
 22. 32 位计数器 HC 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。
- PS. 有关装置的介绍请参考第二章装置说明。

4.2 指令使用限制

- 只能在功能块 (Function Block) 中使用的指令
API0065 CHKADR, FB_NP, FB_PN, NED, ANED, ONED, PED, APED, OPED。
- 中断工作 (TASK) 中不能使用的指令
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，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	NED/ANED/ONED

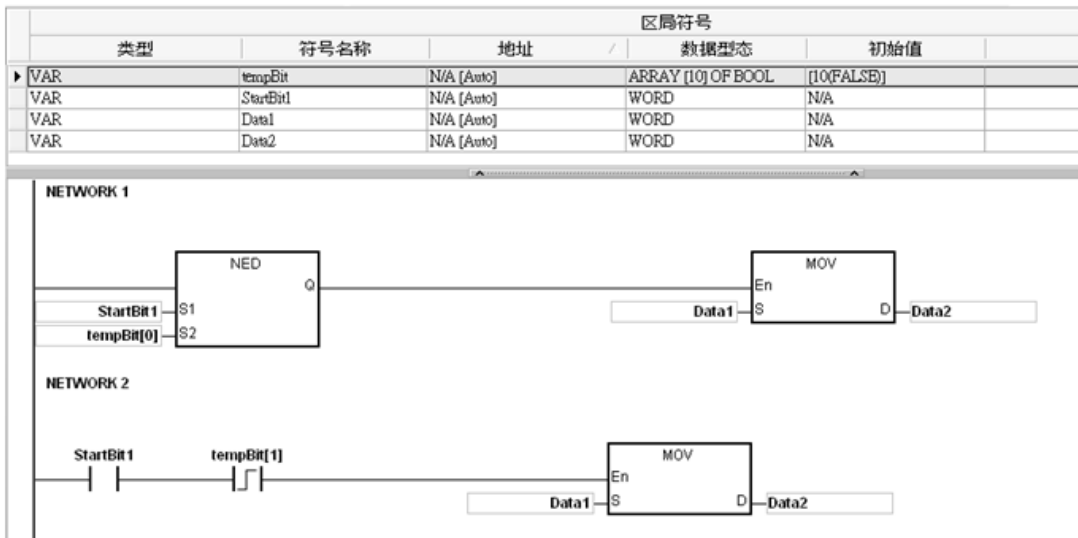
功能块中不可使用指令	功能块中替代指令
LDF/ANDF/ORF	PED/APED/OPED
PLS	-
PLF	-
NP	FB_NP
PN	FB_PN
MC	-
MCR	-
应用命令中所有脉冲执行型指令	注一

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。

4

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）会执行一次 MOV 指令。
3. 当 StartBit1 由 OFF=>ON 时，方法二（NEXTWORK2）会执行一次 MOV 指令。
4. 系统用的 tempBit 变量，不可重复使用。



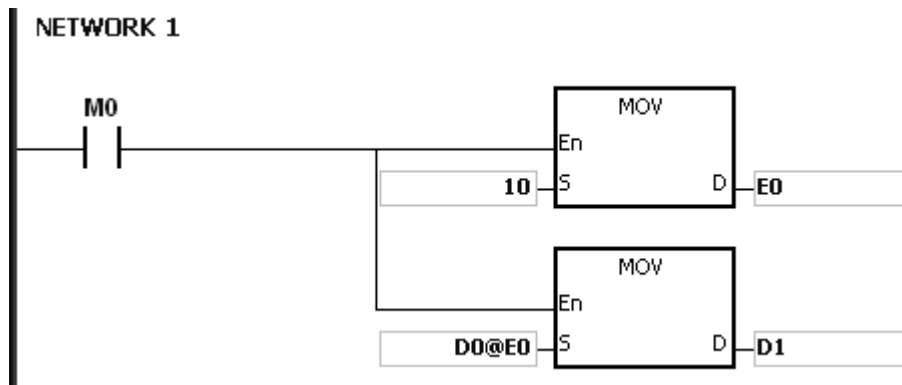
4.3 变址说明

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~E31。

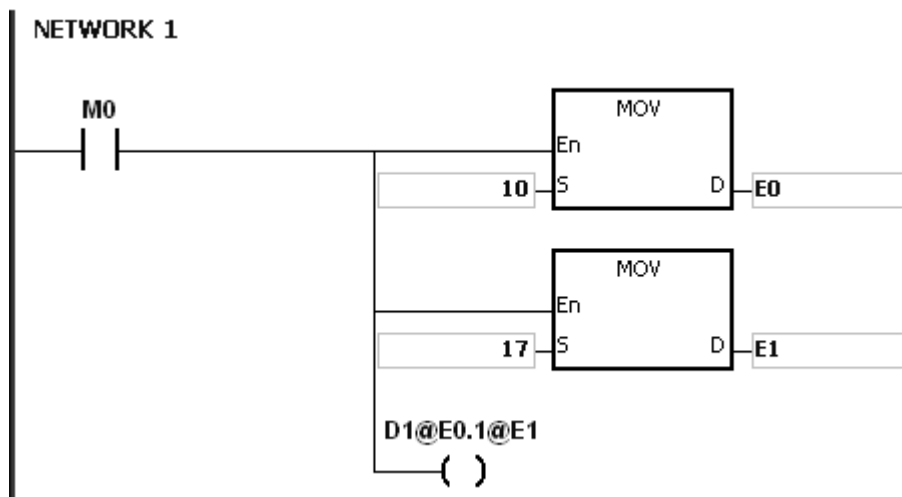
变址的使用方式范列如下：

1. 直接使用寄存器名称来修饰。

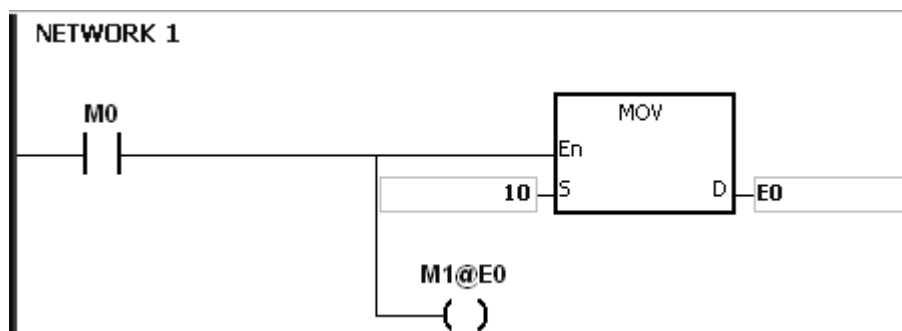
当 M0=ON 时，E0=10，则 $D0@E0=D(0+10)=D10$ ， $D1=D10$ 。



当 M0=ON 时，E0=10，E1=17，则计变址的计算方式为 $D1@E0=D(1+10)=D11$ ，而后面 Bit 的部份 $1@E1=(1+17)=18$ ，但 Bit 的部份最大数为 15，则 $m=18/16=1$ ， $n=18\%16=2$ （取余数），所以最后修饰的结果为 $D(11+m).n=D12.2$ ，则结果 D12.2 会 ON。



当 M0=ON 时，E0=10， $M1@E0=M(1+10)=M11$ ，则 M11=ON。

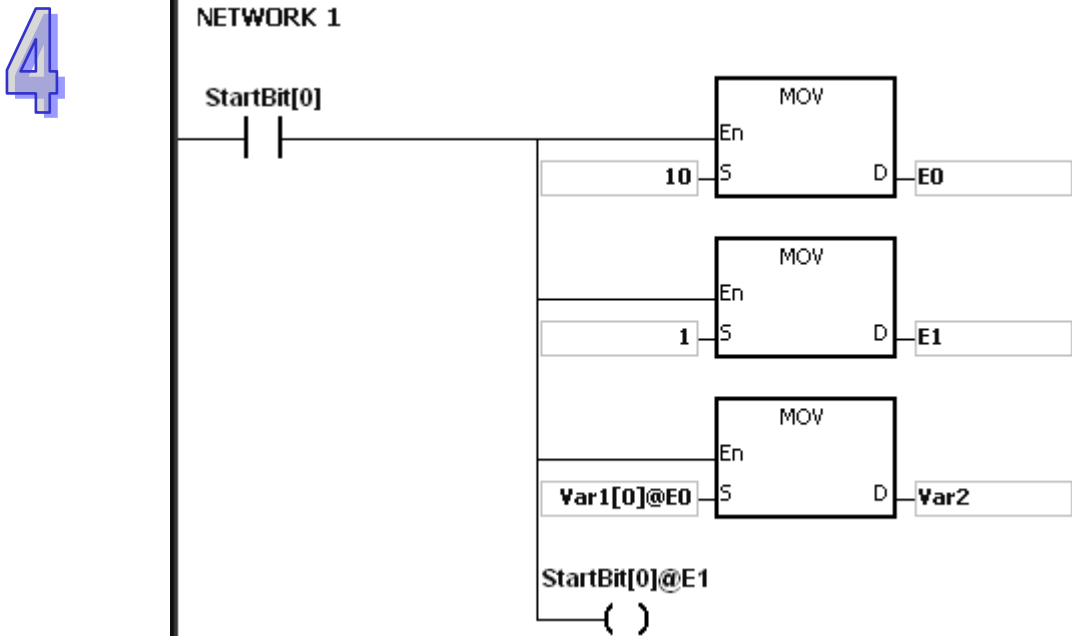


2. 先宣告变量再修饰。

- 先在 ISPSOft 的变量表宣告三个变量 StartBit · Var1 · Var2
 StartBit 的类型为 BOOL 数组，大小为 2 个 Bit，范围为 StartBit[0]~StartBit[1]
 Var1 的类型为 WORD 数组，大小为 11 个 WORD，范围为 Var1[0]~Var1[10]
 Var2 的类型为 WORD，大小为 1 个 WORD

区局符号				
类型	符号名称	地址	数据类型	初始值
VAR	StartBit	N/A [Auto]	BOOL	FALSE
VAR	Var1	N/A [Auto]	ARRAY [11] OF WORD	N/A
VAR	Var2	N/A [Auto]	WORD	0

- 当 StartBit[0]=ON 时，E0=10，E1=1，Var1[0]@E0=Var1[10]，所以 Var2=Var1[10]，StartBit[0]@E1=StartBit[1]，所以 StartBit=ON。



补充说明：当使用 ISPSOft 宣告变量时，又要将变量变址偏移，必需注意变量偏移到哪个地址，避免程序造成误动作。

4.4 指针寄存器 (PR) 说明

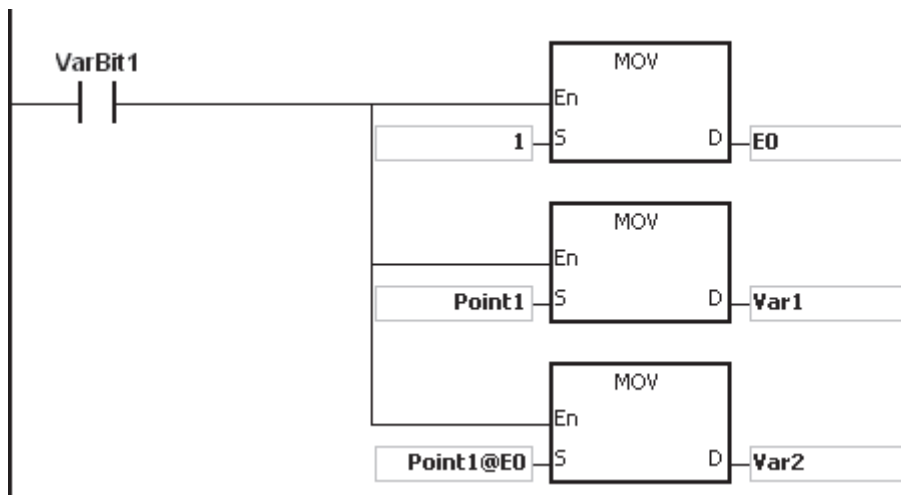
- ISPSOft 支持功能块的功能，当功能块的变量类别宣告为 VAR_IN_OUT，数据类型为 POINTER (指针) 时，此变量将配置 PR 装置，PR 可以传入的装置为 (X · Y · D · L) 以及 Address 为 ISPSOft 自动配置的变量。
- 每个功能块中可以宣告的 PR 个数为 16 个：PR0 ~ PR15，或是使用位 PR0.0 ~ PR15.15。

范例：

1. 先使用 ISPSOft 建立一个程式 POU (程序组织单元) 。
2. 建立一个功能块，名称为 FB0 。



3. FB0 功能块中的程序。



4. 在 FB0 功能块中的变量宣告。

类别选择 VAR_IN_OUT，符号名称名称为 Point1，数据类型选择 POINTER，此变量将配置为指针寄存器。

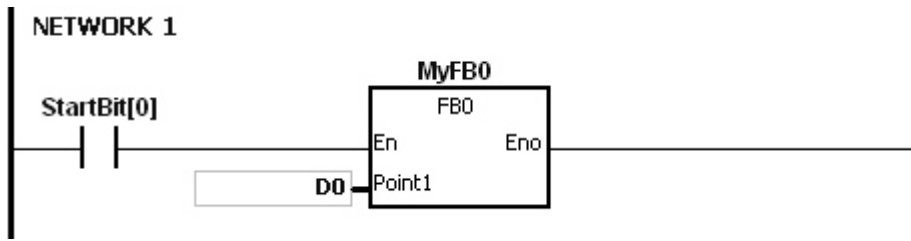
区局符号				
类型	符号名称	地址	数据类型	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR	Var1	N/A [Auto]	WORD	0
VAR	Var2	N/A [Auto]	WORD	0
VAR_IN_OUT	Point1	N/A [Auto]	POINTER	N/A

5. POU 中的变量宣告。

区局符号				
类型	符号名称	地址	数据类型	初始值
VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	[2(FALSE)]
VAR	CVar1	N/A [Auto]	ARRAY [2] OF WORD	[2(0)]
VAR	MyFB0	N/A [Auto]	FB0	N/A

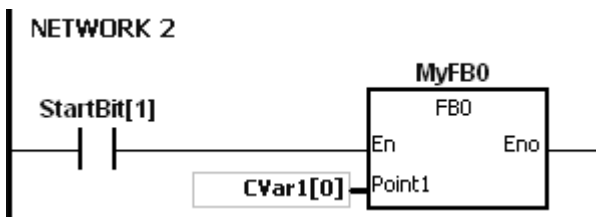
6. 然后在 POU 中调用 FB0 。
7. POU 中的程序。

方法一：当 StartBit[0]=ON 时，D0 的地址将传入给 FB0 中的 Point1。



当 FB0 功能块中的 VarBit1=ON 时， $E0=1$ ， $Var1=D0$ ， $Point1@E0=D(0+1)=D1$ ，所以 $Var2=D1$ 。

方法二：当 StartBit[1]=ON 时，CVar1[0]的地址将传入给 FB0 中的 Point1。



当 FB0 功能块中的 VarBit1=ON 时， $E0=1$ ， $Var1=CVar1[0]$ ， $Point1@E0=CVar1(0+1)=Cvar1[1]$ ，所以 $Var2=CVar1[1]$ 。

4

4.5 定时器指针寄存器 (TR) 说明

- ISPSOft 支持功能块的功能，如果要在功能块中使用定时器时，必需在功能块中宣告一个 TR 装置来并在调用功能块时传入 T 装置的指针。
- 当功能块的变量类别宣告为 VAR_IN_OUT，数据类型为 T_POINTER 时，此变量将配置 TR 装置，TR 可以传入的装置为 (T) 以及 ISPSOft 配置为定时器的变量。
- 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，操作数区域中支持装置 T，则必须使用 TR 指针寄存器来运算。

范例：

1. 先使用 ISPSOft 建立一个程序 POU (程序组织单元)。
2. 建立一个功能块，名称为 FB0。

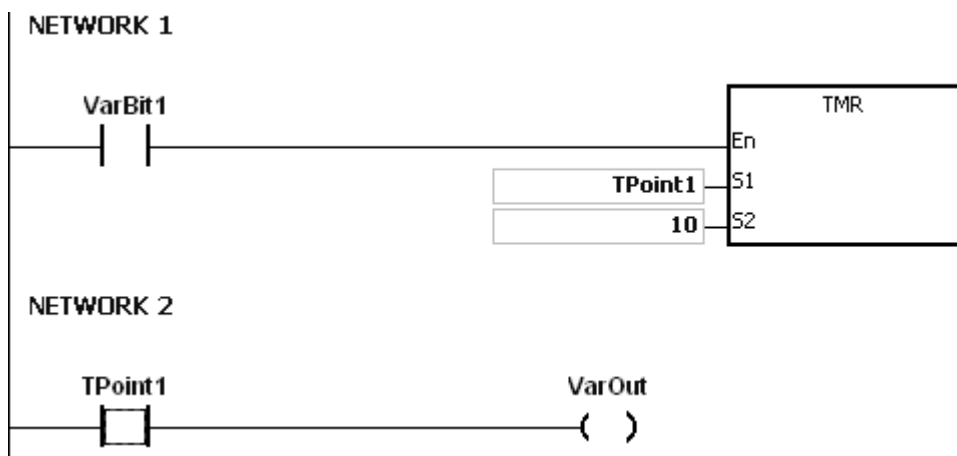


3. 在 FB0 功能块中的变量宣告。

类别选择 VAR_IN_OUT，符号名称名称为 TPoint1，数据类型选择 T_POINTER，此变量将配置为定时器指针寄存器。

全局符号				
类型	符号名称	地址	数据类型	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR_IN_OUT	TPoint1	T0	T_POINTER	N/A
VAR	VarOut	N/A [Auto]	BOOL	FALSE

4. FB0 功能块中的程序。



4

5. POU 中的变量宣告。

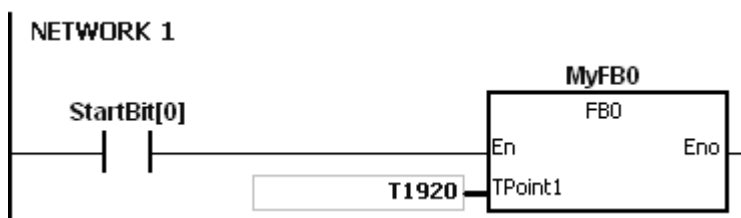
其中 CVar1 必需宣告为 TIMER 的数据型态

全局符号				
类型	符号名称	地址	数据类型	初始值
VAR	StartBit	N/A [Auto]	ARRAY [2] OF E	[2(FALSE)]
VAR	CVar1	T0	TIMER	0
VAR	MyFB0	N/A [Auto]	FB0	N/A

6. 然后在 POU 中调用 FB0。

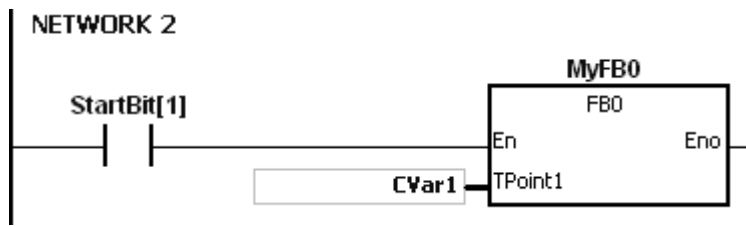
7. POU 中的程序。

方法一：当 StartBit[0]=ON 时，T1920 的地址将传入给 FB0 中的 TPoint1。



当 FB0 功能块中的 VarBit1=ON 时，TMR 指令执行，TPoint1 (T1920) 开始计数，当 TPoint1 计数到达时 VarOut=ON。

方法二：当 StartBit[1]=ON 时，CVar1[0]的地址将传入给 FB0 中的 TPoint1。



当 FB0 功能块中的 VarBit1=ON 时，TMR 指令执行，TPoint1 (CVar1) 开始计数，当 TPoint1 计数到达时 VarOut=ON。

4.6 16 位计数器指针寄存器 (CR) 说明

4

- ISPSOft 支持功能块的功能，如果要在功能块中使用 16 位计数器时，必需在功能块中宣告一个 CR 装置，并在调用功能块时传入 C 装置的指针。
- 当功能块的变量类别宣告为 VAR_IN_OUT，数据类型为 C_POINTER 时，此变量将配置 CR 装置，CR 可以传入的装置为 (C) 以及 ISPSOft 配置为计数器的变量。
- 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，操作数区域中支持装置 C，则必须使用 CR 指针寄存器来运算。

范例：

1. 先使用 ISPSOft 建立一个程序 POU (程序组织单元)。
2. 建立一个功能块，名称为 FB0。



3. 在 FB0 功能块中的变量宣告。
类别选择 VAR_IN_OUT，符号名称为 CPoint1，数据类型选择 C_POINTER，此变量将配置为计数器指针寄存器。

区属符号				
类型	符号名称	地址	数据类型	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR_IN_OUT	CPoint1	CR0	C_POINTER	N/A

4. FB0 功能块中的程序。



5. POU 中的变量宣告。

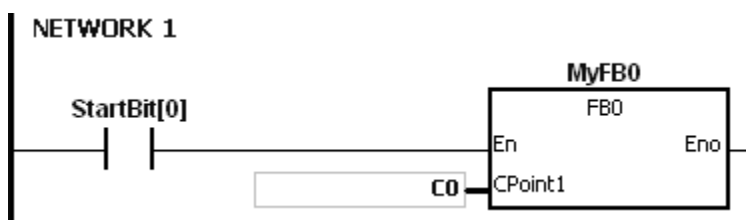
其中 CVar1 必需宣告为 COUNTER 的数据型态。

区局符号				
类型	符号名称	地址	数据型态	初始值
VAR	StartBit	N/A [Auto]	ARRAY [2] OF E [2(FALSE)]	
VAR	CVar1	C1	COUNTER	N/A
VAR	MyFB0	N/A [Auto]	FB0	N/A

6. 然后在 POU 中调用 FB0。

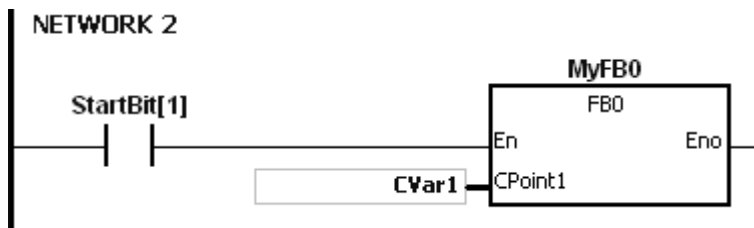
7. POU 中的程序。

方法一：当 StartBit[0]=ON 时，C0 的地址将传入给 FB0 中的 CPoint1。



当 FB0 功能块中的 VarBit1=ON 时，CPoint1 (C0) =ON。

方法二：当 StartBit[1]=ON 时，CVar1 的地址将传入给 FB0 中的 CPoint1。



当 FB0 功能块中的 VarBit1=ON 时，CPoint1 (CVar1) =ON。

4.7 32 位计数器指针寄存器 (HCR) 说明

- ISPSOft 支持功能块的功能，如果要在功能块中使用 32 位计数器时，必需在功能块中宣告一个 HCR 装置，并在调用功能块时传入 HC 装置的指针。
- 当功能块的变量类别宣告为 VAR_IN_OUT，数据型态为 HC_POINTER 时，此变量将配置 HCR 装置，HCR 可以传入的装置为 (HC) 以及 ISPSOft 配置为计数器的变量。
- 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，操作数区域中支持装置 HC，则必须使用 HCR 指针寄存器来运算。

范例：

1. 先使用 ISPSOft 建立一个程序 POU (程序组织单元)。
2. 建立一个功能块，名称为 FB0。



3. 在 FB0 功能块中的变量宣告。

类别选择 VAR_IN_OUT，符号名称为 HCPoint1，数据类型选择 HC_POINTER，此变量将配置为计数器指针寄存器。

全局符号				
类型	符号名称	地址	数据类型	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR_IN_OUT	HCPoint1	N/A [Auto]	HC_POINTER	N/A

4. FB0 功能块中的程序。



5. POU 中的变量宣告。

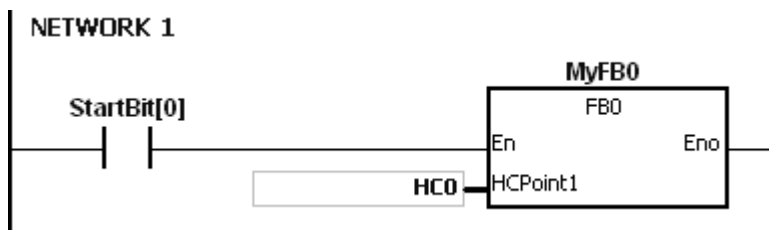
其中 CVar1 必需宣告为 COUNTER 的数据型态，并自行在地址字段中填入 HC 装置的实际地址。

全局符号				
类型	符号名称	地址	数据类型	初始值
VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	[2](FALSE)
VAR	CVar1	HC1	COUNTER	N/A
VAR	MyFB0	N/A [Auto]	FB0	N/A

6. 然后在 POU 中调用 FB0。

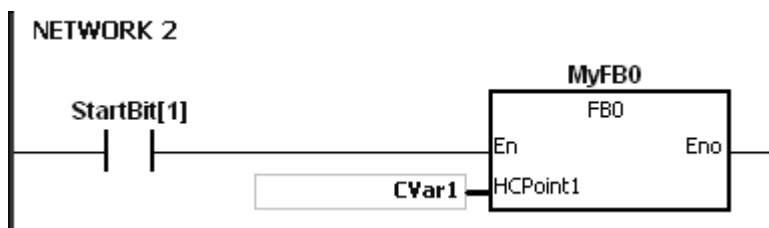
7. 在 POU 中调用 FB0。

方法一：当 StartBit[0]=ON 时，HC0 的地址将传入给 FB0 中的 HCPoint1。



当 FB0 功能块中的 VarBit1=ON 时，HCPoint1 (HC0) =ON。

方法二：当 StartBit[1]=ON 时，CVar1 的地址将传入给 FB0 中的 HCPoint1。



当 FB0 功能块中的 VarBit1=ON 时，HCPoint1 (CVar1) =ON。

5

第5章 基本指令

目录

5.1 基本指令一览表.....	5-2
5.2 基本指令说明.....	5-3

5.1 基本指令一览表

指令码	功能	操作数	STEPS	页码
<u>LD/AND/OR</u>	载入/串联/并联 A 接点	DX、X、Y、M、SM、S、 T、C、HC、D、L、PR	1-2	5-3
<u>LDI/ANI/ORI</u>	载入/串联/并联 B 接点	DX、X、Y、M、SM、S、 T、C、HC、D、L、PR	1-2	5-4
<u>ANB/ORB</u>	串联/并联回路方块	-	1	5-5
<u>MPS/MRD/MPP</u>	存入/读出/读出堆栈	-	1	5-6
<u>OUT</u>	驱动线圈	DY、X、Y、M、SM、S、 T、C、HC、D、L、PR	1-2	5-7
<u>SET</u>	动作保持 (ON)	DY、X、Y、M、SM、S、 T、C、HC、D、L、PR	1-2	5-8
<u>MC/MCR</u>	共通串联接点之连结/解除	N	1	5-9
<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、D、L、PR	1-2	5-11
<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、D、L、PR	1-2	5-12
<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、D、L、PR	5	5-13
<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、D、L、PR	5	5-14
<u>PLS</u>	上微分输出	DY、X、Y、M、SM、S、 T、C、HC、D、L、PR	1-2	5-15
<u>PLF</u>	下微分输出	DY、X、Y、M、SM、S、 T、C、HC、D、L、PR	1-2	5-16
<u>INV</u>	运算结果反相	-	1	5-17
<u>NOP</u>	无动作	-	1	5-18
<u>NP</u>	上升沿触发指令	-	1	5-19
<u>PN</u>	下降沿触发指令	-	1	5-20
<u>FB_NP</u>	上升沿触发指令	X、Y、M、SM、S、T、 C、HC、D、L、PR	1-2	5-21
<u>FB_PN</u>	下降沿触发指令	X、Y、M、SM、S、T、 C、HC、D、L、PR	1-2	5-22
<u>PSTOP</u>	PLC 程序停止执行	-	1	5-23

5.2 基本指令说明

指令码		操作数										功能		
LD/AND/OR		S										载入/串联/并联 A 接点		
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR	
S	●		●	●	●	●	●	●	●	●	●	●	●	

符号：

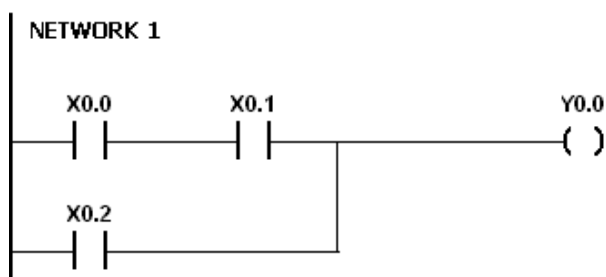


指令说明：

- 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。



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

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

符号：

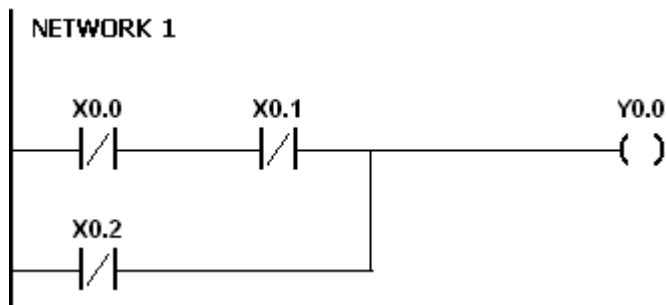


指令说明：

- LDI 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。
- ANI 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。
- ORI 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。

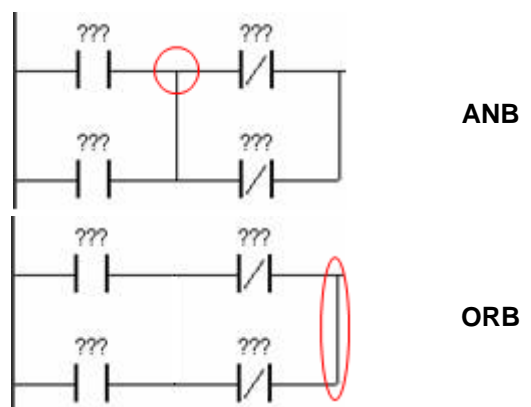
程序范例：

- 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。



指令码	操作数	功能
ANB/ORB	-	串联/并联回路方块

符号：

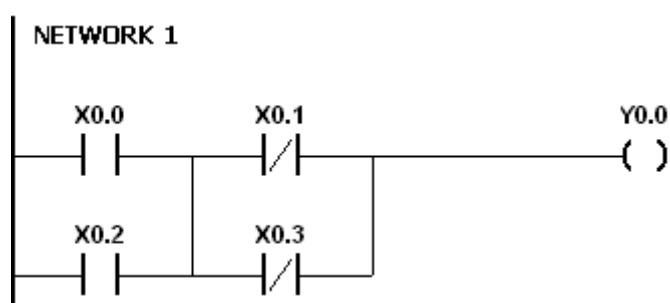


指令说明：

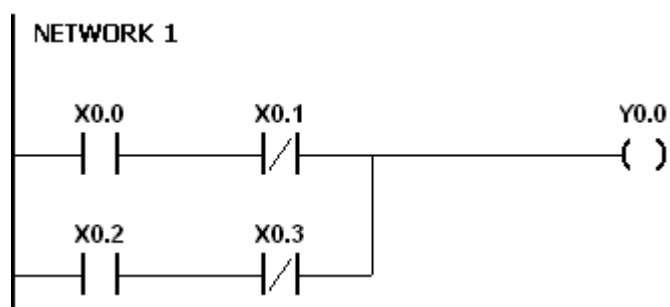
1. ANB 是将前一保存的逻辑结果与目前累积寄存器的内容作“与”（AND）的运算。
2. ORB 是将前一保存的逻辑结果与目前累积寄存器的内容作“或”（OR）的运算。

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。



指令码	操作数	功能
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 X0.0 的 A 接点，存入堆栈。
2. 串联 X0.1 的 A 接点，驱动 Y0.1 线圈，读出堆栈（指针不动）。
3. 串联 X0.2 的 A 接点，驱动 M0 线圈，读出堆栈。

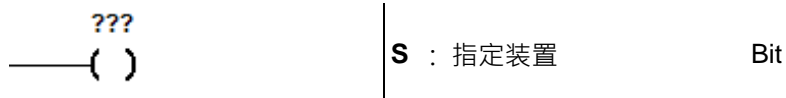
指令：	操作说明：
LD X0.0	载入X0.0的A接点
MPS	存入堆栈
AND X0.1	串联X0.1的A接点
OUT Y0.1	驱动Y0.1线圈
MRD	读出堆栈（指针不动）
AND X0.2	串联X0.2的A接点
OUT M0	驱动M0线圈
MPP	读出堆栈
OUT Y0.2	驱动Y0.2线圈
END	程序结束

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。

指令码		操作数							功能				
OUT		S							驱动线圈				
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S		●	●	●	●	●	●	●	●	●	●	●	●

符号：



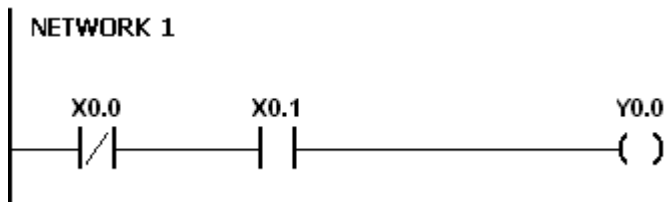
指令说明：

1. 将 OUT 指令之前的逻辑运算结果输出至指定的组件。
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：

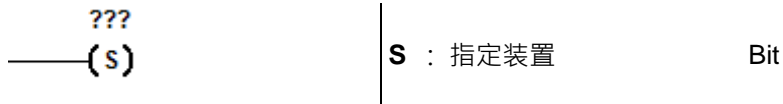
1. 加载 X0.0 的 B 接点，串联 X0.1 的 A 接点，驱动 Y0.0 线圈。
2. 当 X0.0=OFF 且 X0.1=ON 时，Y0.0=ON。



指令码	操作数	功能
SET	S	动作保持 (ON)

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

符号：



指令说明：

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

程序范例：

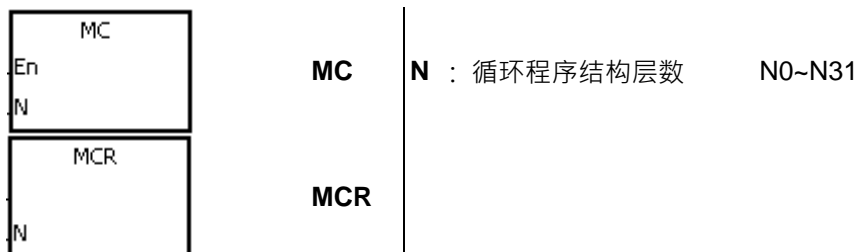
1. 载入 X0.0 的 B 接点，串联 Y0.0 的 A 接点，Y0.1 动作保持 (ON)。
2. 当 X0.0=OFF 且 Y0.0=ON 时，Y0.1=ON 且即使运算结果改变，Y0.1 亦保持 ON 的状态。



5

指令码	操作数	功能
MC/MCR	N	共通串联接点之连结/解除

符号：



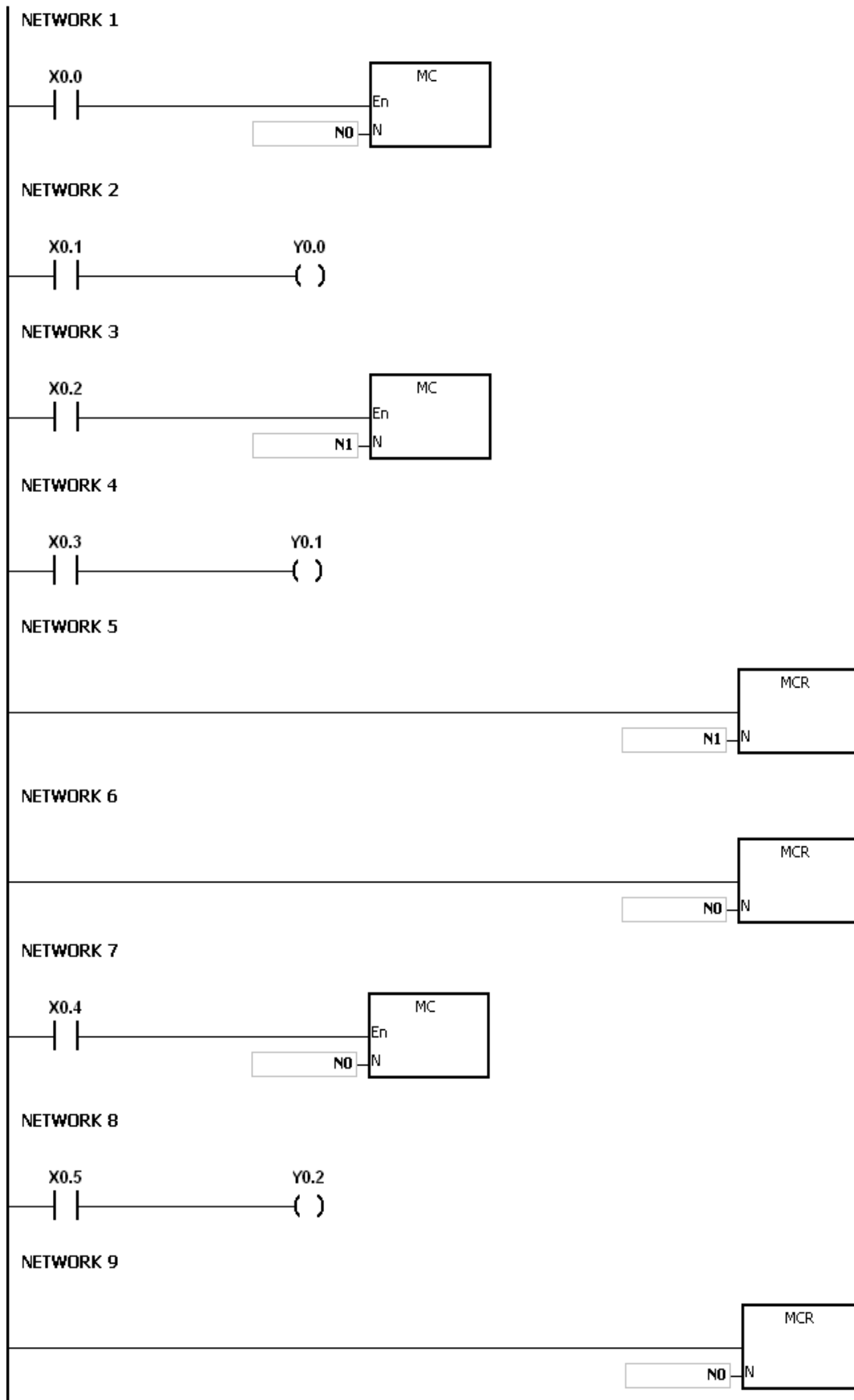
指令说明：

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 指令驱动的组件	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：



5

指令码		操作数						功能					
LDP/ANDP/ORP		S						上升沿检出动作开始/串联/并联连接					
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S	●		●	●	●	●	●	●	●	●	●	●	●

符号：

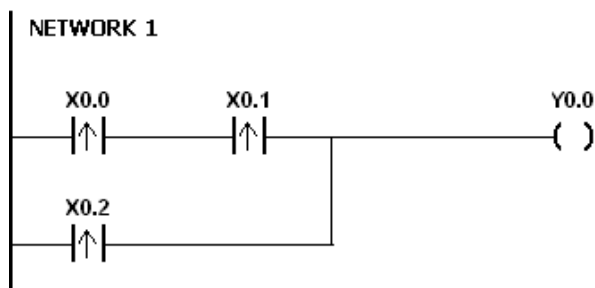


指令说明：

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。



指令码		操作数										功能		
LDF/ANDF/ORF		S										下降沿检出动作开始/串联/并联连接		
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR	
S	●		●	●	●	●	●	●	●	●	●	●	●	

符号：

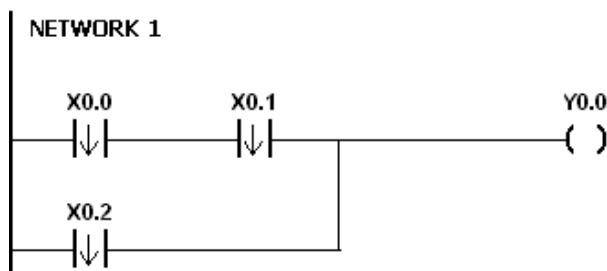


指令说明：

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。

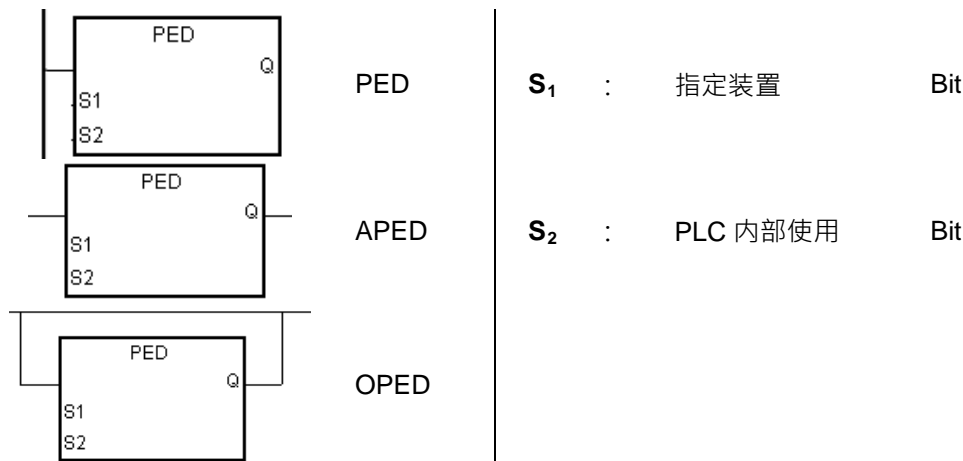
程序范例：

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。



指令码		操作数		功能									
PED/APED/OPED		$S_1 \cdot S_2$		上升沿检出动作开始/串联/并联连接									
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S_1			●	●	●	●	●	●	●	●	●	●	●
S_2			●	●	●	●	●	●	●	●	●	●	●

符号：

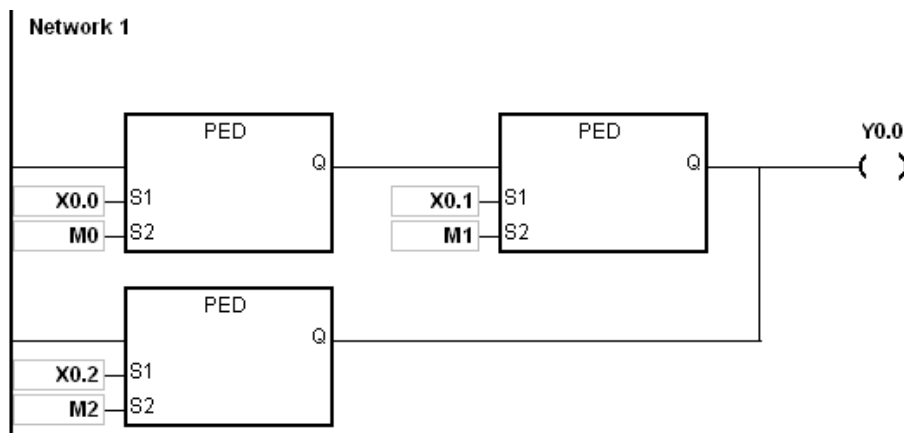


指令说明：

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 S_2 的Bit装置，让PLC内部记住该接点的上一次状态，此 S_2 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. 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。

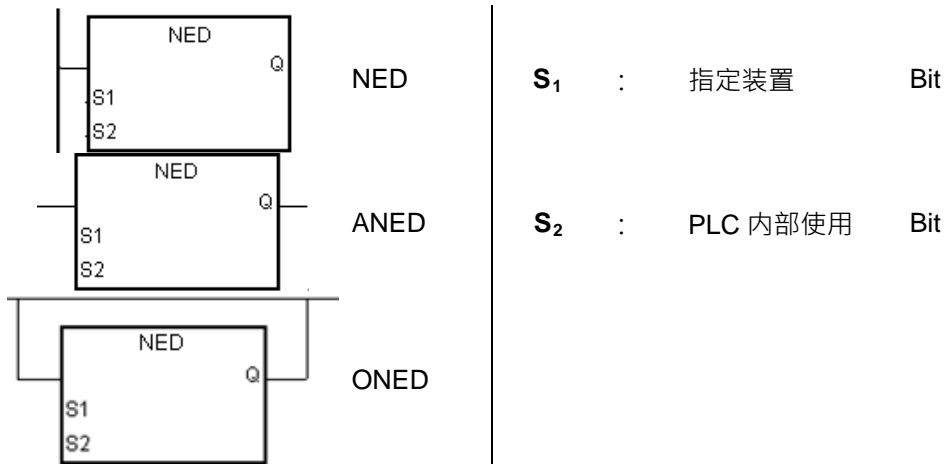
程序范例：

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。



指令码		操作数		功能									
NED/ANED/ONED		$S_1 \cdot S_2$		下降沿检出动作开始/串联/并联连接									
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S_1			●	●	●	●	●	●	●	●	●	●	●
S_2			●	●	●	●	●	●	●	●	●	●	●

符号：

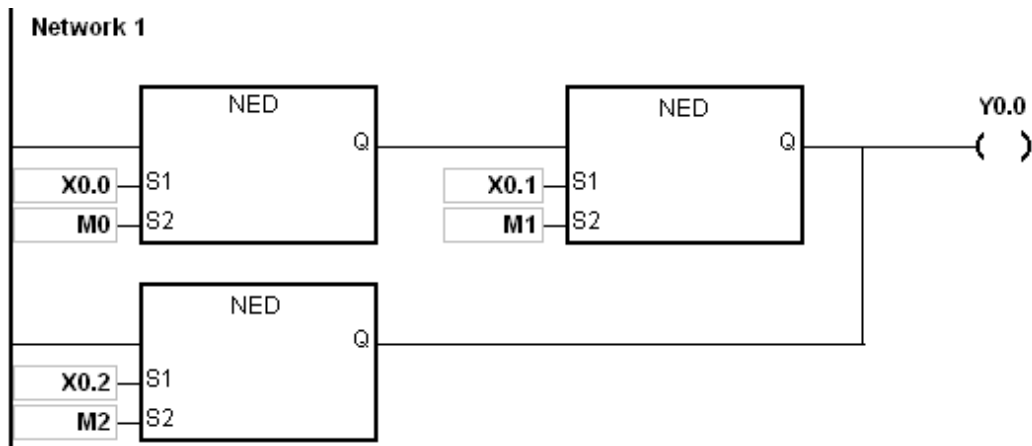


指令说明：

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 S_2 的Bit装置，让PLC内部记住该接点的上一次状态，此 S_2 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。

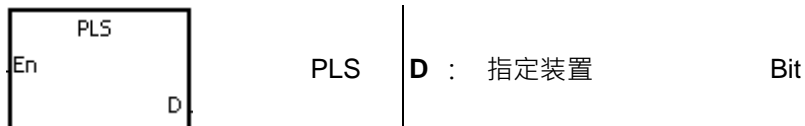
程序范例：

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。



指令码		操作数										功能		
PLS		D										上微分输出		
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR	
S		●	●	●	●	●	●	●	●	●	●	●	●	

符号：

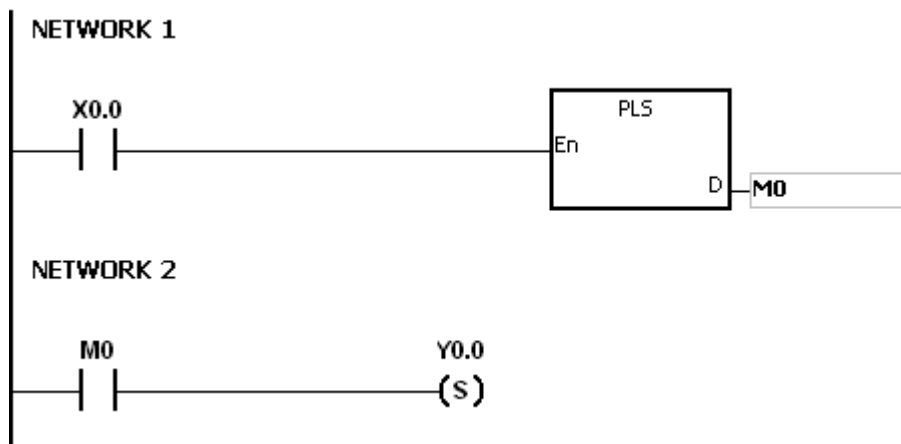


指令说明：

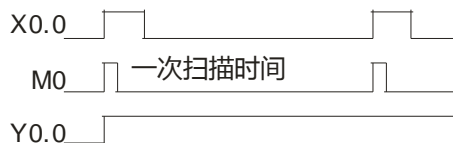
1. 上微分输出指令。当条件接点由 OFF 到 ON (上升沿触发) 时，PLS 指令被执行，D 送出一脉冲，脉冲长度为一次扫描时间。
2. 功能块中请勿使用。

程序范例：

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



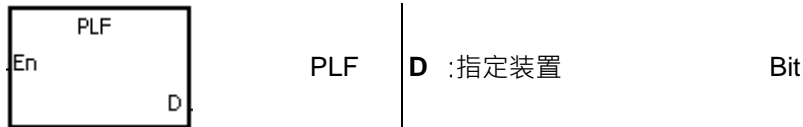
时序图：



指令码	操作数	功能
PLF	D	下微分输出

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

符号：



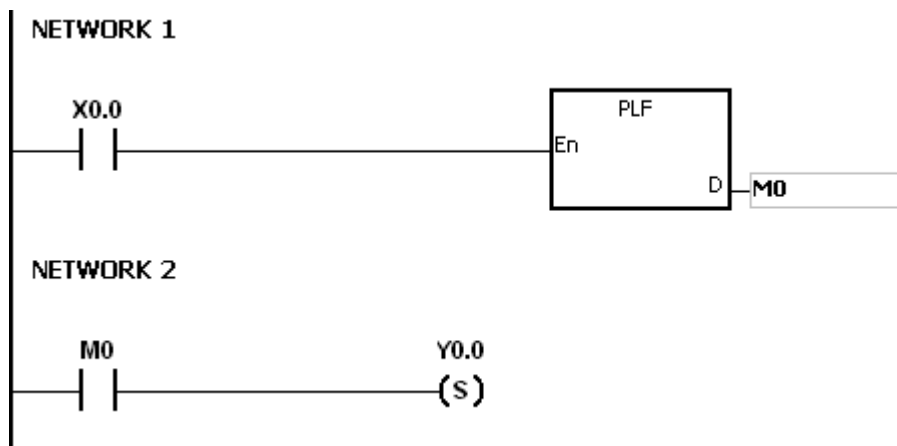
指令说明：

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 送出一次脉冲, 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 Pulse 的时间; M0=ON 时, Y0.0 Set 为 ON。

5



时序图：



指令码	操作数	功能
INV	-	运算结果反相

符号：

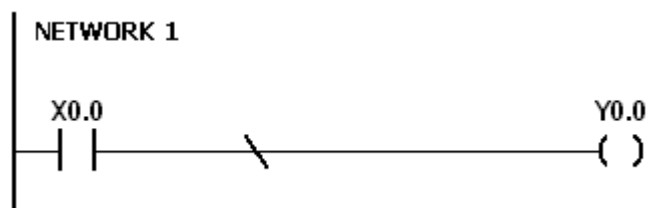


指令说明：

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。



指令码	操作数	功能
NOP	无	无动作

符号：无

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

5

指令码	操作数	功能
NP	-	上升沿触发指令

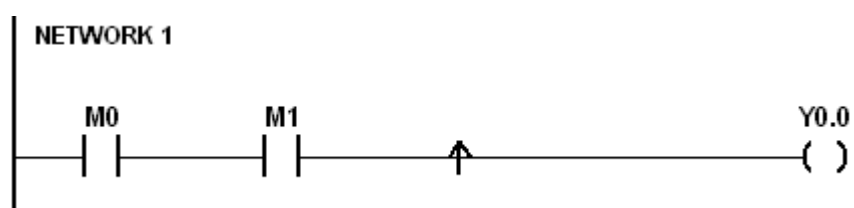
符号：



指令说明：

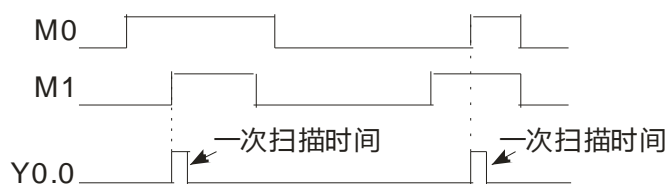
1. 当累积寄存器由 0 变为 1 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. 在 Function Block 中请使用 FB_NP 指令。

程序范例：



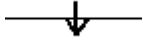
指令：	操作说明：
LD M0	载入 M0 的 A 接点
AND M1	串入 M1 之 A 接点
NP	上升沿触发指令
OUT Y0.0	驱动线圈 Y0.0

时序图：



指令码	操作数	功能
PN	-	下降沿触发指令

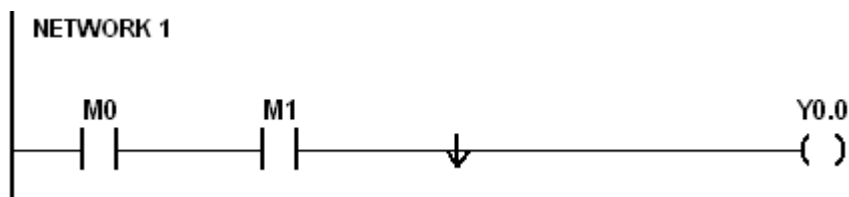
符号：



指令说明：

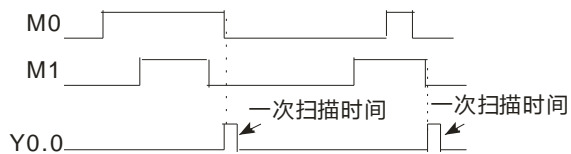
1. 当累积寄存器由 1 变为 0 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. 在 Function Block 中请使用 FB_P N 指令。

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：



指令码		操作数										功能	
FB_NP		S										上升沿触发指令	
装置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S			●	●	●	●	●	●	●	●	●	●	●

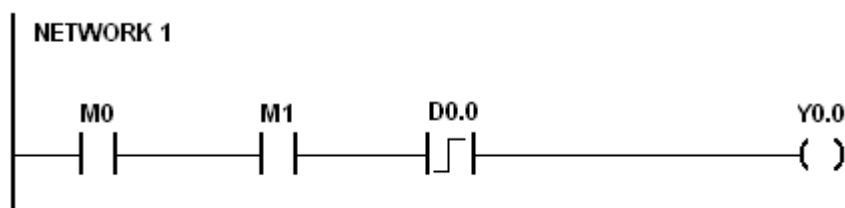
符号：



指令说明：

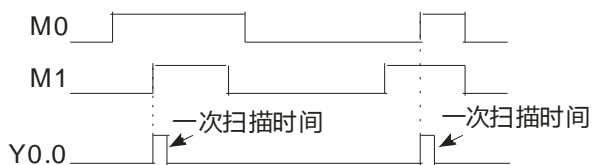
1. 当累积寄存器由 0 变为 1 时，此指令将使累积寄存器维持一次扫描周期的 1，然后第二次扫描周期之后，自动将累积寄存器改为 0。
2. S 装置，让 PLC 内部记住该接点的上一次状态，此 S 装置在程序中的其它地方请勿重复使用，否则可能会造成执行结果错误。
3. 本指令只能在 Function Block 中使用。

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：



指令码	操作数	功能
FB_PN	S	下降沿触发指令

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

符号：

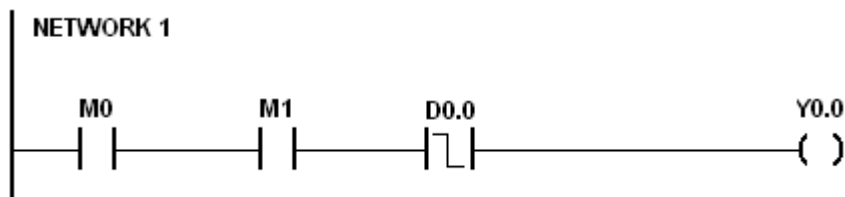


指令说明：

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。

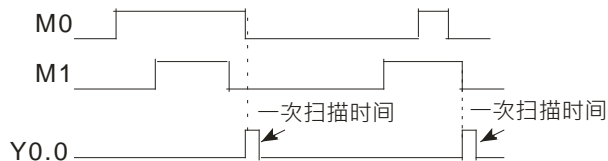
程序范例：

5



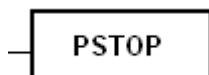
指令：	操作说明：
LD M0	载入 M0 的 A 接点
AND M1	串入 M1 之 A 接点
FB_PN D0.0	下降沿触发指令
OUT Y0.0	驱动线圈 Y0.0

时序图：



指令码	操作数	功能
PSTOP	-	PLC 程序停止执行

符号：

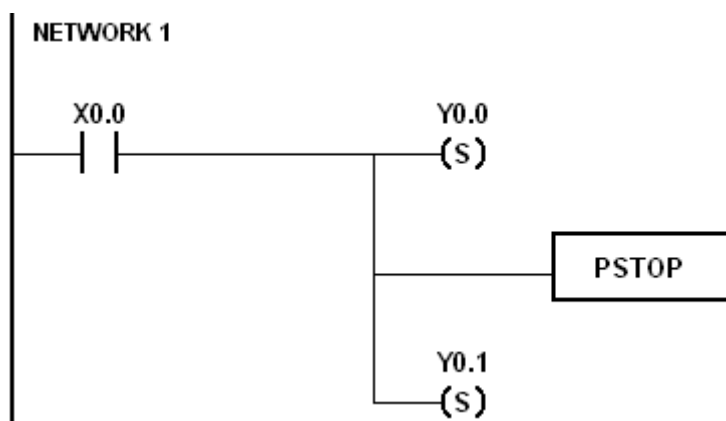


指令说明：

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。



MEMO

5

6

第6章 指令表

目录

6.1 比较操作指令.....	6-4
6.1.1 比较操作指令一览表	6-4
6.1.2 比较操作指令说明	6-7
6.2 四则运算指令.....	6-36
6.2.1 四则运算指令一览表	6-36
6.2.2 四则运算指令说明	6-37
6.3 数据转换指令.....	6-73
6.3.1 数据转换指令一览表	6-73
6.3.2 数据转换指令说明	6-74
6.4 数据转移指令.....	6-110
6.4.1 数据转移指令一览表	6-110
6.4.2 数据转移指令说明	6-111
6.5 程序跳转指令.....	6-134
6.5.1 程序跳转指令一览表	6-134
6.5.2 程序跳转指令说明	6-135
6.6 程序执行控制指令	6-143
6.6.1 程序执行控制指令一览表	6-143
6.6.2 程序执行控制指令说明.....	6-144
6.7 I/O 更新指令	6-150
6.7.1 I/O 更新指令一览表.....	6-150
6.7.2 I/O 更新指令说明	6-151
6.8 便利指令	6-152
6.8.1 便利指令一览表	6-152
6.8.2 便利指令说明.....	6-153
6.9 逻辑操作指令.....	6-187
6.9.1 逻辑操作指令一览表	6-187
6.9.2 逻辑操作指令说明	6-188

6.10 旋转指令	6-207
6.10.1 旋转指令一览表	6-207
6.10.2 旋转指令说明	6-208
6.11 基本指令	6-218
6.11.1 基本指令一览表	6-218
6.11.2 基本指令说明	6-219
6.12 位移指令	6-226
6.12.1 位移指令一览表	6-226
6.12.2 位移指令说明	6-227
6.13 数据处理指令	6-252
6.13.1 数据处理指令一览表	6-252
6.13.2 数据处理指令说明	6-253
6.14 结构建立指令	6-297
6.14.1 结构建立指令一览表	6-297
6.14.2 结构建立指令说明	6-298
6.15 模块的数据读写指令	6-305
6.15.1 模块的数据读写指令一览表	6-305
6.15.2 模块的数据读写指令说明	6-306
6.16 浮点数指令	6-311
6.16.1 浮点数指令一览表	6-311
6.16.2 浮点数指令说明	6-312
6.17 万年历指令	6-346
6.17.1 万年历指令一览表	6-346
6.17.2 万年历指令说明	6-347
6.18 外围设备指令	6-360
6.18.1 外围设备指令一览表	6-360
6.18.2 外围设备指令说明	6-361
6.19 通讯指令	6-374
6.19.1 通讯指令一览表	6-374
6.19.2 通讯指令说明	6-375
6.20 其它指令	6-406

6.20.1 其它指令一览表	6-406
6.20.2 其它指令说明	6-407
6.21 字符串处理指令	6-416
6.21.1 字符串处理指令一览表	6-416
6.21.2 字符串处理指令说明	6-417
6.22 以太网控制指令	6-476
6.22.1 以太网控制指令一览表	6-476
6.22.2 以太网控制指令说明	6-477
6.23 存储卡读写指令	6-502
6.23.1 存储卡读写指令一览表	6-502
6.23.2 存储卡读写指令说明	6-503
6.24 任务控制指令	6-515
6.24.1 任务控制指令一览表	6-515
6.24.2 任务控制指令说明	6-516
6.25 SFC 控制指令	6-518
6.25.1 SFC 控制指令一览表	6-518
6.25.2 SFC 控制指令说明	6-519
6.26 冗余控制指令	6-525
6.26.1 冗余控制指令一览表	6-525
6.26.2 冗余控制指令说明	6-526

6.1 比较操作指令

6.1.1 比较操作指令一览表

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
0000	LD=	DLD=	—	—	$S1 = S2$	5
0001	LD<>	DLD<>	—	—	$S1 \neq S2$	5
0002	LD>	DLD>	—	—	$S1 > S2$	5
0003	LD>=	DLD>=	—	—	$S1 \geq S2$	5
0004	LD<	DLD<	—	—	$S1 < S2$	5
0005	LD<=	DLD<=	—	—	$S1 \leq S2$	5
0006	AND=	DAND=	—	—	$S1 = S2$	5
0007	AND<>	DAND<>	—	—	$S1 \neq S2$	5
0008	AND>	DAND>	—	—	$S1 > S2$	5
0009	AND>=	DAND>=	—	—	$S1 \geq S2$	5
0010	AND<	DAND<	—	—	$S1 < S2$	5
0011	AND<=	DAND<=	—	—	$S1 \leq S2$	5
0012	OR=	DOR=	—	—	$S1 = S2$	5
0013	OR<>	DOR<>	—	—	$S1 \neq S2$	5
0014	OR>	DOR>	—	—	$S1 > S2$	5
0015	OR>=	DOR>=	—	—	$S1 \geq S2$	5
0016	OR<	DOR<	—	—	$S1 < S2$	5
0017	OR<=	DOR<=	—	—	$S1 \leq S2$	5
0018	—	FLD=	DFLD=	—	$S1 = S2$	5-7
0019	—	FLD<>	DFLD<>	—	$S1 \neq S2$	5-7
0020	—	FLD>	DFLD>	—	$S1 > S2$	5-7
0021	—	FLD>=	DFLD>=	—	$S1 \geq S2$	5-7
0022	—	FLD<	DFLD<	—	$S1 < S2$	5-7
0023	—	FLD<=	DFLD<=	—	$S1 \leq S2$	5-7
0024	—	FAND=	DFAND=	—	$S1 = S2$	5-7
0025	—	FAND<>	DFAND<>	—	$S1 \neq S2$	5-7
0026	—	FAND>	DFAND>	—	$S1 > S2$	5-7
0027	—	FAND>=	DFAND>=	—	$S1 \geq S2$	5-7
0028	—	FAND<	DFAND<	—	$S1 < S2$	5-7

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
0029	–	FAND<=	DFAND<=	–	$S1 \leq S2$	5-7
0030	–	FOR=	DFOR=	–	$S1 = S2$	5-7
0031	–	FOR<>	DFOR<>	–	$S1 \neq S2$	5-7
0032	–	FOR>	DFOR>	–	$S1 > S2$	5-7
0033	–	FOR>=	DFOR>=	–	$S1 \geq S2$	5-7
0034	–	FOR<	DFOR<	–	$S1 < S2$	5-7
0035	–	FOR<=	DFOR<=	–	$S1 \leq S2$	5-7
0036	LD\$=	–	–	–	$S1 = S2$	5-17
0037	LD\$<>	–	–	–	$S1 \neq S2$	5-17
0038	LD\$>	–	–	–	$S1 > S2$	5-17
0039	LD\$>=	–	–	–	$S1 \geq S2$	5-17
0040	LD\$<	–	–	–	$S1 < S2$	5-17
0041	LD\$<=	–	–	–	$S1 \leq S2$	5-17
0042	AND\$=	–	–	–	$S1 = S2$	5-17
0043	AND\$<>	–	–	–	$S1 \neq S2$	5-17
0044	AND\$>	–	–	–	$S1 > S2$	5-17
0045	AND\$>=	–	–	–	$S1 \geq S2$	5-17
0046	AND\$<	–	–	–	$S1 < S2$	5-17
0047	AND\$<=	–	–	–	$S1 \leq S2$	5-17
0048	OR\$=	–	–	–	$S1 = S2$	5-17
0049	OR\$<>	–	–	–	$S1 \neq S2$	5-17
0050	OR\$>	–	–	–	$S1 > S2$	5-17
0051	OR\$>=	–	–	–	$S1 \geq S2$	5-17
0052	OR\$<	–	–	–	$S1 < S2$	5-17
0053	OR\$<=	–	–	–	$S1 \leq S2$	5-17
0054	CMP	DCMP	–	✓	比较设定输出	7
0055	ZCP	DZCP	–	✓	区域比较	9
0056	FCMP	–	–	✓	浮点数比较	7-9
0057	FZCP	–	–	✓	浮点数区域比较	9-12
0058	MCMP	–	–	✓	矩阵比较	9
0059	CMPT=	–	–	✓	表格比较 · =	9
0060	CMPT<>	–	–	✓	表格比较 · <>	9
0061	CMPT>	–	–	✓	表格比较 · >	9
0062	CMPT>=	–	–	✓	表格比较 · >=	9

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
0063	CMPT<	-	-	✓	表格比较 · <	9
0064	CMPT<=	-	-	✓	表格比较 · <=	9
0065	CHKADR	-	-	-	接点类型指针寄存器地址检查	7

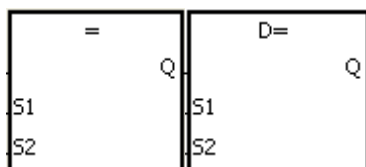
6

6.1.2 比较操作指令说明

API	指令码		操作数										功能				
0000~ 0005	D	LD※	S1 · S2										接点类型比较 LD※				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●	●	●	●		●	○	●	○	○		
S ₂	●	●			●	●	●	●	●		●	○	●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
-	AH500	AH500

符号：

S₁ : 数据来源 1

Word/Double Word

S₂ : 数据来源 2

Word/Double Word

以 LD=和 DLD=为例

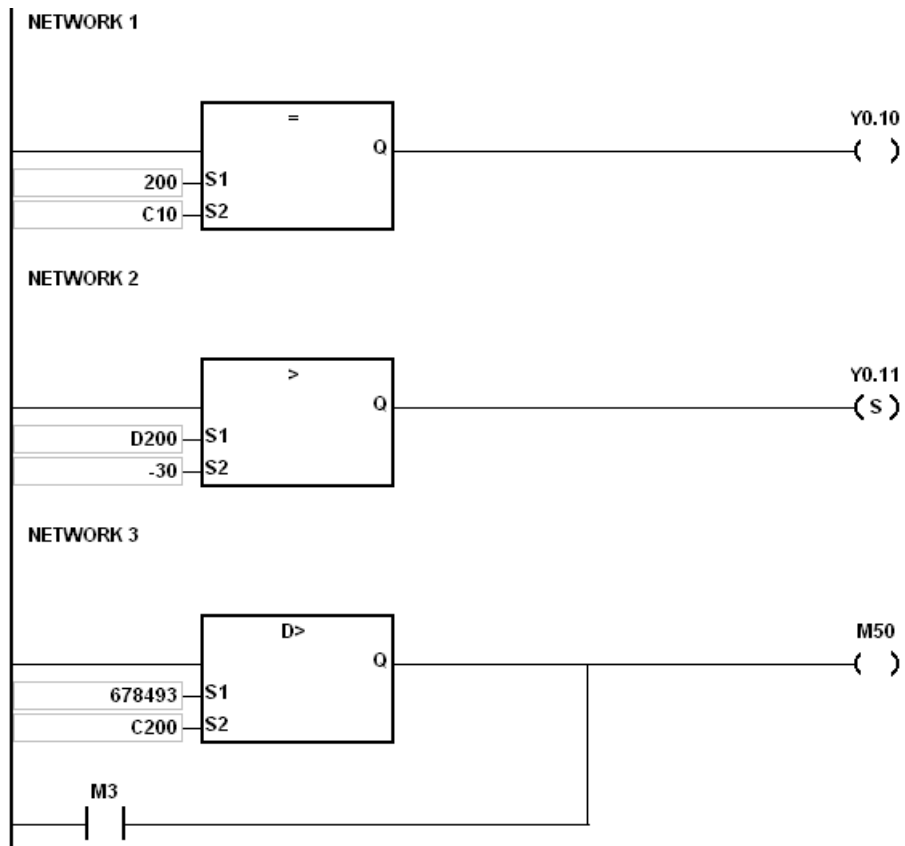
指令说明：

- 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。

API No.	16 位指令	32 位指令	导通条件	非导通条件
0000	LD =	DLD =	S ₁ = S ₂	S ₁ ≠ S ₂
0001	LD < >	DLD < >	S ₁ ≠ S ₂	S ₁ = S ₂
0002	LD >	DLD >	S ₁ > S ₂	S ₁ ≤ S ₂
0003	LD > =	DLD > =	S ₁ ≥ S ₂	S ₁ < S ₂
0004	LD <	DLD <	S ₁ < S ₂	S ₁ ≥ S ₂
0005	LD < =	DLD < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

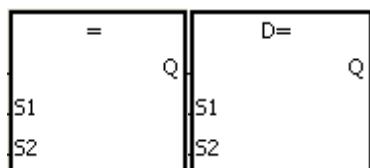
- C10 的内容等于 200 时，Y0.10=ON。
- 当 D200 的内容大于-30，Y0.11=ON 并保持住。
- (C201 · C200) 的内容小于 678,493 或者是 M3=ON 时，M50=ON。



6

API	指令码			操作数								功能					
0006~ 0011	D	AND※		S1 · S2								接点类型比较 AND※					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●	●	●	●		●	○	●	○	○		
S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
										脉冲执行型	16 位指令 (5 steps)			32 位指令 (5 steps)			
										-	AH500			AH500			

符号：



S₁ : 数据来源 1

Word/Double Word

S₂ : 数据来源 2

Word/Double Word

以 AND=和 DAND=为例

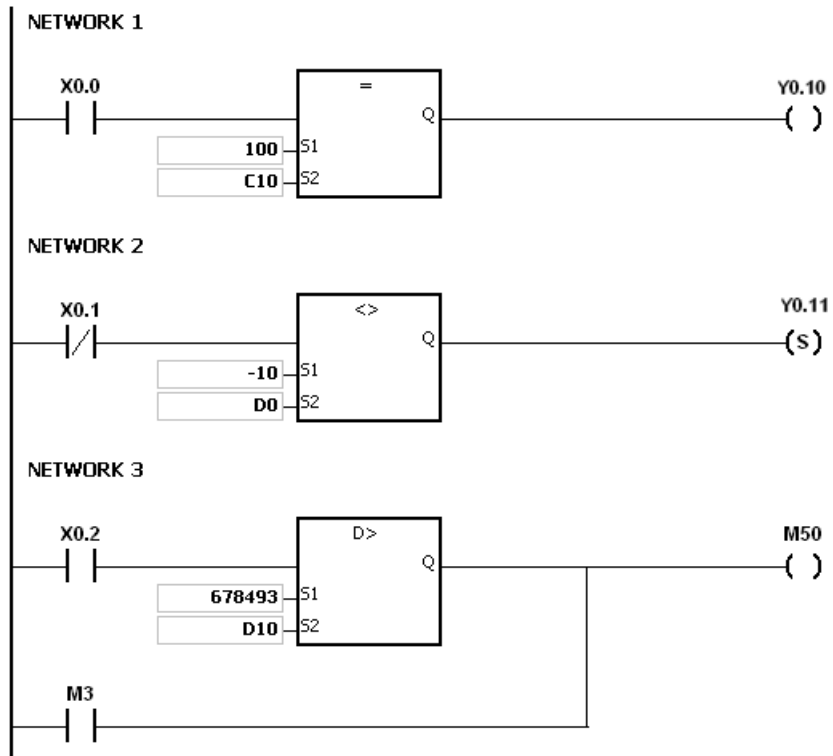
指令说明：

- 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。
- 32 位指令才可以使用 HC 装置。

API No.	16 位指令	32 位指令	导通条件	非导通条件
0006	AND =	DAND =	S ₁ = S ₂	S ₁ ≠ S ₂
0007	AND < >	DAND < >	S ₁ ≠ S ₂	S ₁ = S ₂
0008	AND >	DAND >	S ₁ > S ₂	S ₁ ≤ S ₂
0009	AND > =	DAND > =	S ₁ ≥ S ₂	S ₁ < S ₂
0010	AND <	DAND <	S ₁ < S ₂	S ₁ ≥ S ₂
0011	AND < =	DAND < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

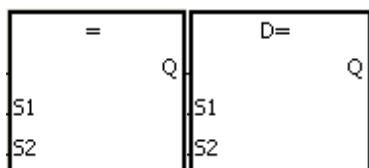
- 当 X0.0=ON 时且 C10 的现在值又等于 100 时，Y0.10=ON。
- 当 X0.1=OFF 而寄存器 D0 的内容又不等于-10 时，Y0.11=ON 并保持住。
- 当 X0.2=ON 而且 32 位寄存器(D11 · D0)的内容又小于 678,493 时或 M3=ON 时，M50=ON。



6

API	指令码		操作数										功能				
0012~ 0017	D	OR※	S1 · S2										接点类型比较 OR※				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●	●	●	●		●	○	●	○	○		
S ₂	●	●			●	●	●	●	●		●	○	●	○	○		
脉冲执行型		16 位指令 (5 steps)					32 位指令 (5 steps)										
-		AH500					AH500										

符号：



S₁ : 数据来源 1

Word/Double Word

S₂ : 数据来源 2

Word/Double Word

以 OR=和 DOR=为例

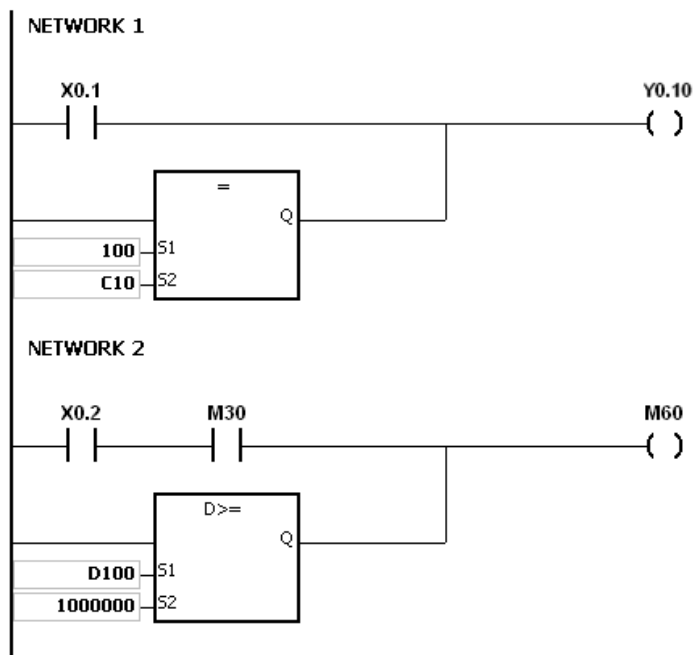
指令说明：

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。

API No.	16 位指令	32 位指令	导通条件	非导通条件
0012	OR =	DOR =	S ₁ = S ₂	S ₁ ≠ S ₂
0013	OR < >	DOR < >	S ₁ ≠ S ₂	S ₁ = S ₂
0014	OR >	DOR >	S ₁ > S ₂	S ₁ ≤ S ₂
0015	OR > =	DOR > =	S ₁ ≥ S ₂	S ₁ < S ₂
0016	OR <	DOR <	S ₁ < S ₂	S ₁ ≥ S ₂
0017	OR < =	DOR < =	S ₁ ≤ S ₂	S ₁ > S ₂

程序范例：

1. 当 X0.1=ON 时，或者是 C10 的现在值等于 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。



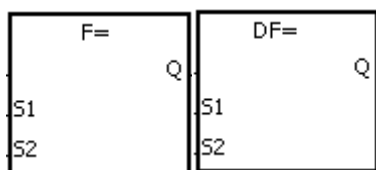
6

API	指令码		操作数				功能			
0018~0023	D	FLD※	S1 · S2				浮点数接点类型比较 LD※			

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

脉冲执行型	32 位指令 (5-7 steps)	64 位指令 (5-7 steps)
-	AH500	AH500

符号：



S₁ : 数据来源 1 Double Word/Long Word
 S₂ : 数据来源 2 Double Word/Long Word

以 FLD=和 DFLD=为例

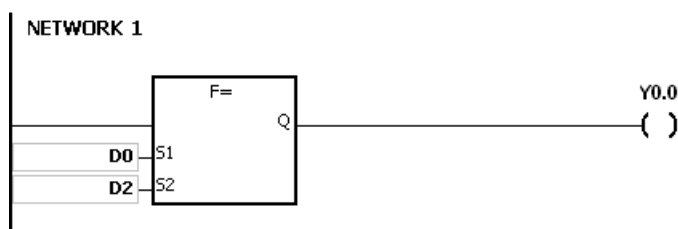
指令说明：

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

API No.	32 位指令	64 位指令	导通条件	非导通条件
0018	FLD =	DFLD =	S ₁ = S ₂	S ₁ ≠ S ₂
0019	FLD < >	DFLD < >	S ₁ ≠ S ₂	S ₁ = S ₂
0020	FLD >	DFLD >	S ₁ > S ₂	S ₁ ≤ S ₂
0021	FLD > =	DFLD > =	S ₁ ≥ S ₂	S ₁ < S ₂
0022	FLD <	DFLD <	S ₁ < S ₂	S ₁ ≥ S ₂
0023	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：

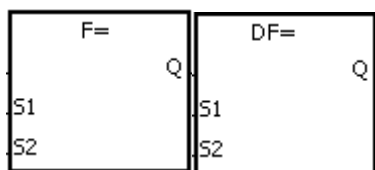
1. 当 S₁ 或 S₂ 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。

API	指令码		操作数										功能				
0024~0029	D	FAND※	S1 · S2										浮点数接点类型比较 AND※				

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

脉冲执行型	32 位指令 (5-7 steps)	64 位指令 (5-7 steps)
-	AH500	AH500

符号：



以 FAND=和 DFAND=为例

S₁ : 数据来源 1 Double Word/Long Word
 S₂ : 数据来源 2 Double Word/Long Word

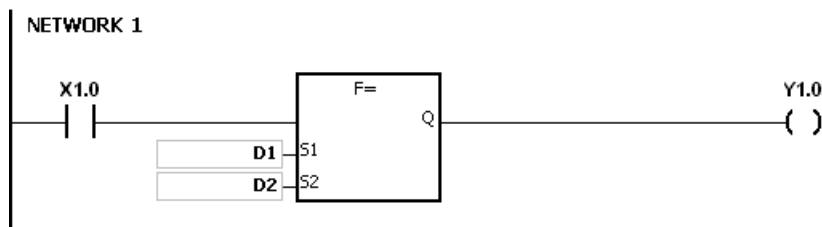
指令说明：

- 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。

API No.	32 位指令	64 位指令	导通条件	非导通条件
0024	FAND =	DFAND =	S ₁ = S ₂	S ₁ ≠ S ₂
0025	FAND < >	DFAND < >	S ₁ ≠ S ₂	S ₁ = S ₂
0026	FAND >	DFAND >	S ₁ > S ₂	S ₁ ≤ S ₂
0027	FAND > =	DFAND > =	S ₁ ≥ S ₂	S ₁ < S ₂
0028	FAND <	DFAND <	S ₁ < S ₂	S ₁ ≥ S ₂
0029	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。

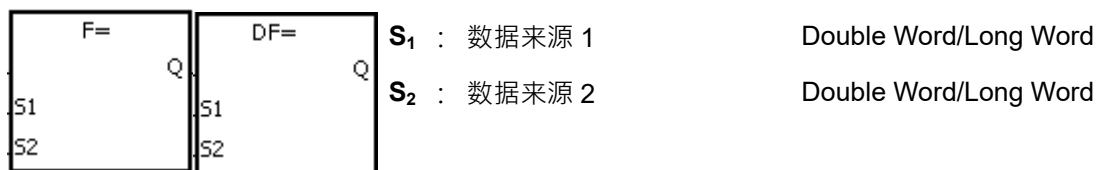


补充说明：

- 当 S₁ 或 S₂ 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。

API	指令码			操作数								功能					
0030~ 0035	D	FOR※		S1 · S2								浮点数接点类型比较 OR※					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁	●	●			●	●	●	●	●		●	○	●				○
S ₂	●	●			●	●	●	●	●		●	○	●				○
脉冲执行型		32 位指令 (5-7 steps)					64 位指令 (5-7 steps)										
-		AH500					AH500										

符号：



以 FOR=和 DFOR=为例

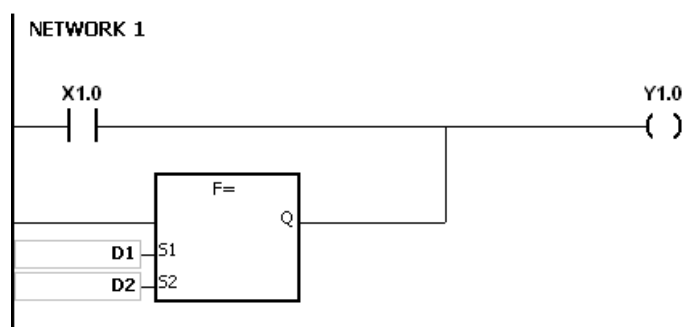
指令说明：

1. 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。

API No.	32 位指令	64 位指令	导通条件	非导通条件
0030	FOR =	DFOR =	S ₁ = S ₂	S ₁ ≠ S ₂
0031	FOR < >	DFOR < >	S ₁ ≠ S ₂	S ₁ = S ₂
0032	FOR >	DFOR >	S ₁ > S ₂	S ₁ ≤ S ₂
0033	FOR > =	DFOR > =	S ₁ ≥ S ₂	S ₁ < S ₂
0034	FOR <	DFOR <	S ₁ < S ₂	S ₁ ≥ S ₂
0035	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：

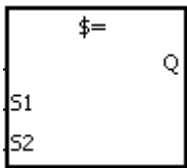
1. 当 S₁ 或 S₂ 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。

API	指令码	操作数	功能
0036~0041	LD\$※	S1 · S2	字符串类型比较 LD\$※

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

脉冲执行型	16 位指令 (5-17 steps)	32 位指令
-	AH500	-

符号 :



S₁ : 数据来源 1 String
 S₂ : 数据来源 2 String

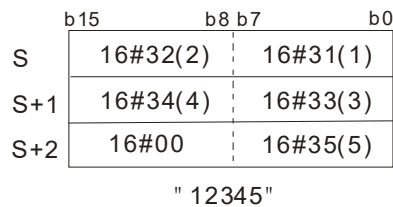
以 LD\$=为例

指令说明 :

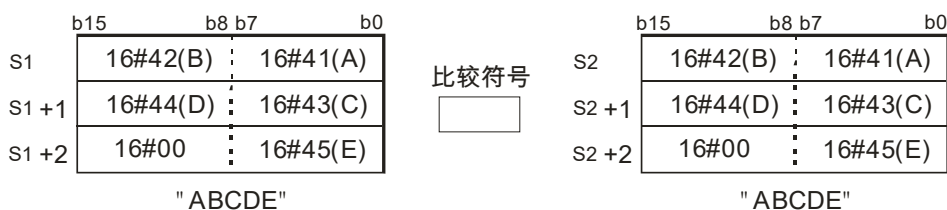
- S₁ 与 S₂ 的内容以字符串类型做比较的指令。以“LD\$=”做为例子，比较结果为“等于”时，该接点导通；“不等于”时，该接点不导通。

API No.	指令	导通条件	非导通条件
0036	LD\$ =	S ₁ = S ₂	S ₁ ≠ S ₂
0037	LD\$ < >	S ₁ ≠ S ₂	S ₁ = S ₂
0038	LD\$ >	S ₁ > S ₂	S ₁ ≤ S ₂
0039	LD\$ > =	S ₁ ≥ S ₂	S ₁ < S ₂
0040	LD\$ <	S ₁ < S ₂	S ₁ ≥ S ₂
0041	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：



- 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“ABCD F ”而 S_2 的字符串是“ABC D E ”，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：

	b15	b8	b7	b0		b15	b8	b7	b0	
S_1	16#42(B)		16#41(A)		比较符号 <input type="checkbox"/>	S_2	16#42(B)		16#41(A)	
S_1+1	16#44(D)		16#43(C)			S_2+1	16#44(D)		16#43(C)	
S_1+2	16#00		16#46(F)			S_2+2	16#00		16#45(E)	
"ABCD F "						"ABC D E "				

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

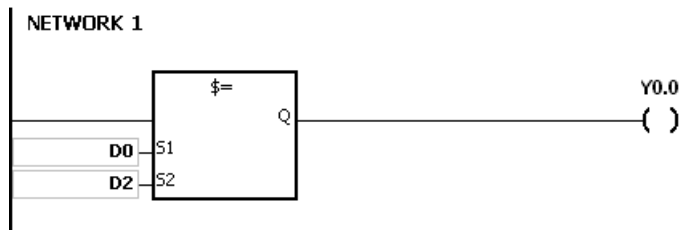
5. 字符串长度不相等时，以字符串长度较长的为大；如下范例 S_1 的字符串是“1234567”而 S_2 的字符串是“99999”，因为 S_1 的字符串长度是 7 而 S_2 的字符串是 5，所以认定 S_1 的字符串> S_2 的字符串，各指令执行结果如下表所列：

	b15	b8	b7	b0		b15	b8	b7	b0	
S_1	16#32(2)		16#31(1)		比较符号 <input type="checkbox"/>	S_2	16#39(9)		16#39(9)	
S_1+1	16#34(4)		16#33(3)			S_2+1	16#39(9)		16#39(9)	
S_1+2	16#36(6)		16#35(5)			S_2+2	16#00		16#39(9)	
S_1+3	16#00		16#37(7)			"99999"				
"1234567"										

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。

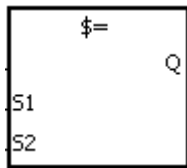


补充说明：

1. 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。

API	指令码		操作数										功能				
0042~0047	AND\$※		S1 · S2										字符串类型比较 AND\$※				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●	○	●			○	
S ₂	●	●			●	●		●	●		●	○	●			○	
脉冲执行型		16 位指令 (5-17 steps)					32 位指令										
-		AH500					-										

符号：



以 AND\$= 为例

S₁ : 数据来源 1

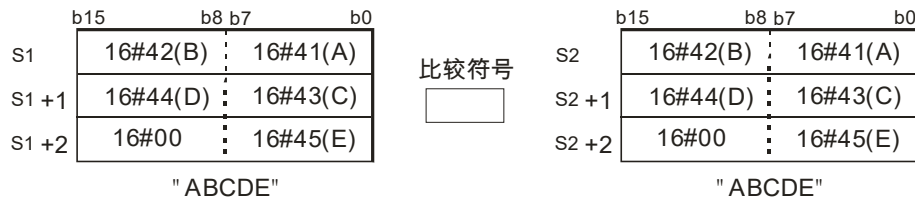
String

S₂ : 数据来源 2

String

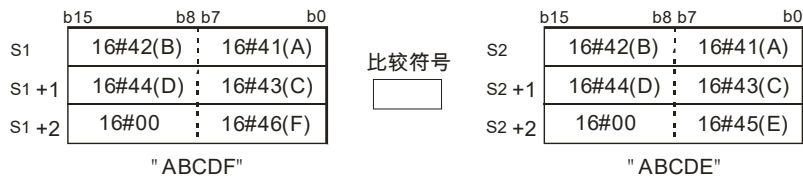
指令说明：

- S₁ 与 S₂ 的内容以字符串类型做比较的指令。以“AND\$=”做例子，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：



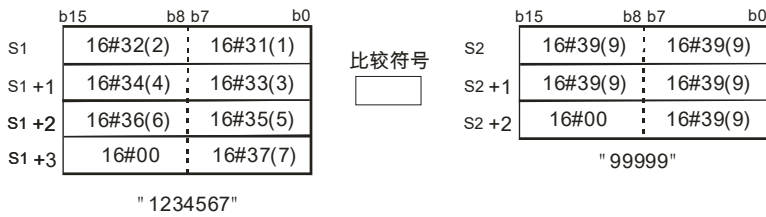
比较符号	比较结果
\$ =	导通
\$ < >	不导通
\$ >	不导通
\$ > =	导通
\$ <	不导通
\$ < =	导通

- 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 S₁ 的字符串是“ABCDF”而 S₂ 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 S₁ 的字符串>S₂ 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

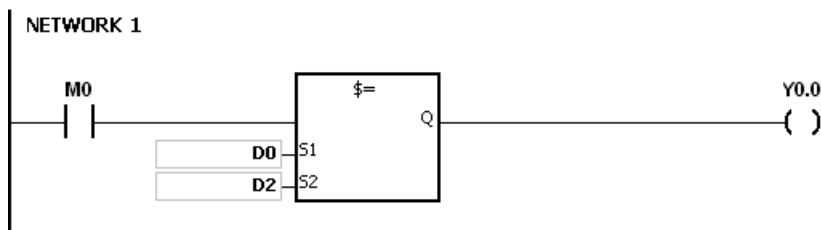
5. 字符串长度不相等时，以字符串长度较长的大；如下范例 S₁ 的字符串是“1234567”而 S₂ 的字符串是“99999”，因为 S₁ 的字符串长度是 7 而 S₂ 的字符串是 5，所以认定 S₁ 的字符串 > S₂ 的字符串，各指令执行结果如下表所列：



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：

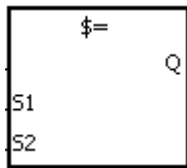
当 M0=ON 且 D0~16#00 的字符串等于 D2~16#00 的字符串，则 Y0.0 ON。



API	指令码		操作数										功能				
0048~0053		OR\$※	S1 · S2										字符串类型比较 OR\$※				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●	○	●			○	
S ₂	●	●			●	●		●	●		●	○	●			○	

脉冲执行型	16 位指令 (5-17 steps)	32 位指令
-	AH500	-

符号：



以 OR\$= 为例

S₁ : 数据来源 1

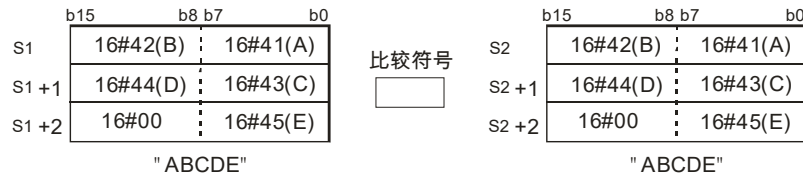
String

S₂ : 数据来源 2

String

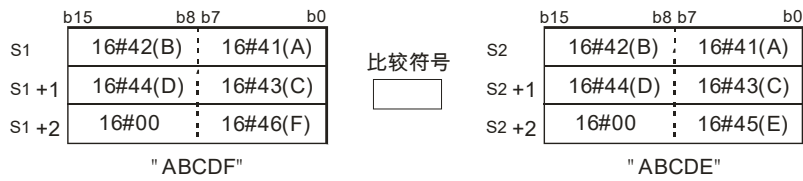
指令说明：

- 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：



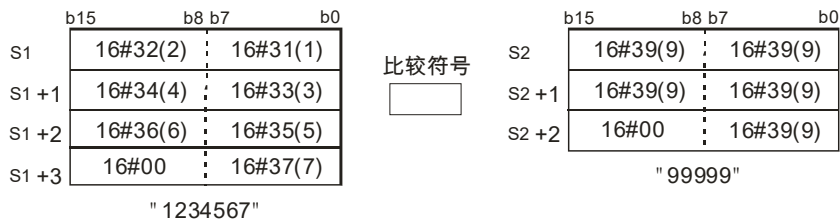
比较符号	比较结果
\$ =	导通
\$ < >	不导通
\$ >	不导通
\$ > =	导通
\$ <	不导通
\$ < =	导通

- 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 S₁ 的字符串是“ABCDF”而 S₂ 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 S₁ 的字符串>S₂ 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

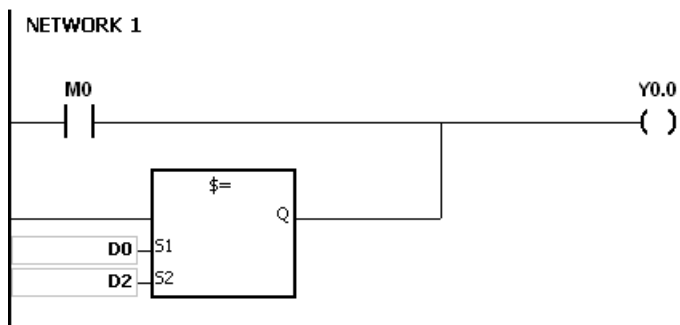
5. 字符串长度不相等时，以字符串长度较长的大；如下范例 **S₁** 的字符串是“1234567”而 **S₂** 的字符串是“99999”，因为 **S₁** 的字符串长度是 7 而 **S₂** 的字符串是 5，所以认定 **S₁** 的字符串 > **S₂** 的字符串，各指令执行结果如下表所列：



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

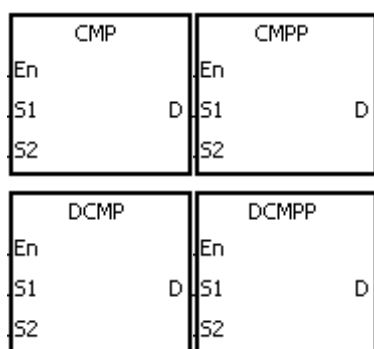


API	指令码			操作数							功能						
0054	D	CMP	P	S1 · S2 · D							比较设定输出						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



S₁ : 比较值 1
Word/double word

S₂ : 比较值 2
Word/double word

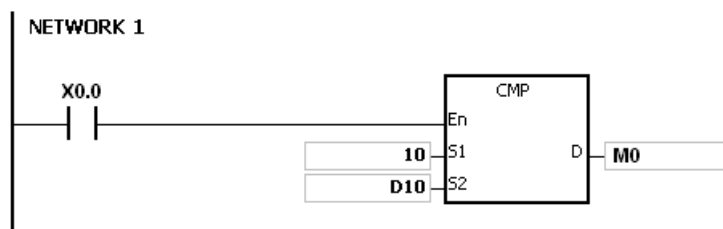
D : 比较结果
Bit

指令说明：

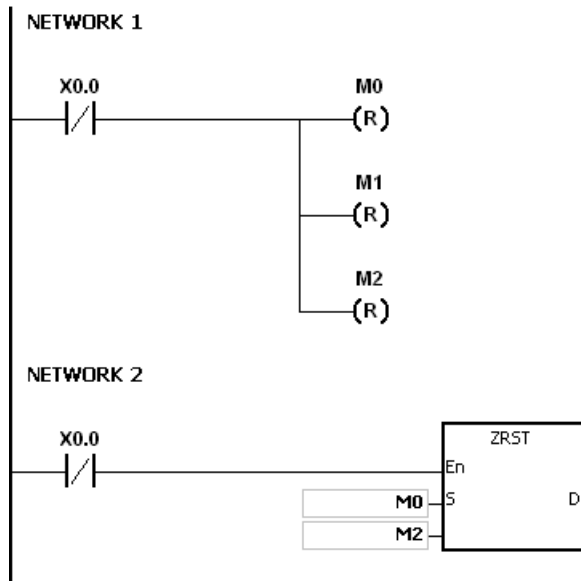
1. 将操作数 **S₁** 和 **S₂** 的内容以有号数十进制大小比较，其比较结果在 **D** 中表示。
2. 操作数 **D** 占用 3 个连续装置。**D**、**D+1**、**D+2** 用于储存比较结果，如果 **S₁>S₂**，**D=ON**；如果 **S₁=S₂**，**D+1=ON**；如果 **S₁<S₂**，**D+2=ON**。
3. **DCMP**、**DCMPP** 指令才可使用 **HC** 装置。

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=OFF**，**CMP** 指令停止执行并且 **M0**、**M1**、**M2** 保持它们的现在值。



3. 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。

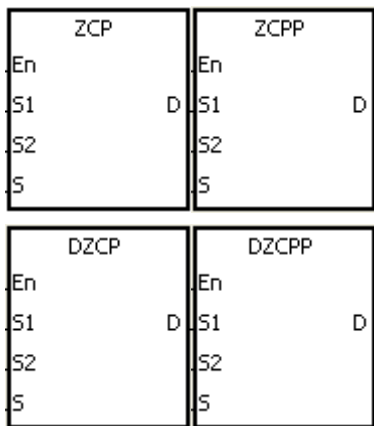
6

API	指令码			操作数							功能						
0055	D	ZCP	P	S1 · S2 · S · D							区域比较						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令 (9 steps)
AH500	AH500	AH500

符号：



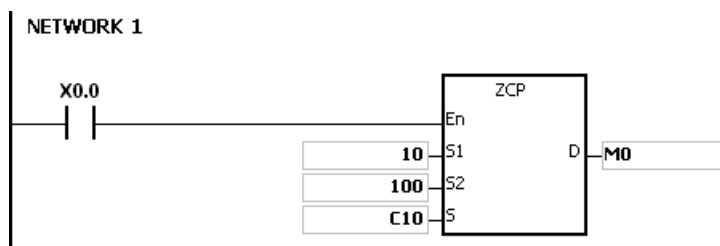
- S1** : 区间比较的下限值 Word/Double Word
- S2** : 区间比较的上限值 Word/Double Word
- S** : 比较值 Word/Double Word
- D** : 比较结果 Bit

指令说明：

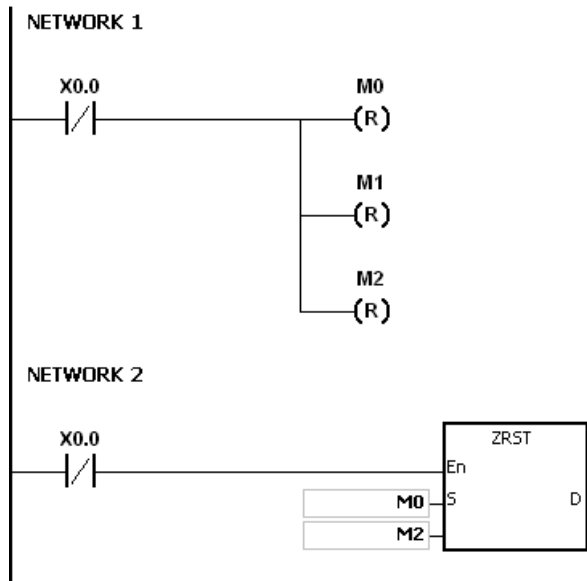
1. 比较值 **S** 与下限 **S₁** 及上限 **S₂** 以有号数十进制数值做比较，其比较结果在 **D** 作表示。
2. 操作数 **S₁** 必须比 **S₂** 小。当 **S₁>S₂**，指令执行时把 **S₁** 作为上/下限值进行比较。
3. 操作数 **D** 占用 3 个连续的装置 **D** **D+1** **D+2** 储存比较结果。如果 **S₁>S** **D=ON**；如果 **S₁≤S≤S₂**，**D+1=ON**；如果 **S₂<S**，**D+2=ON**。
4. **DZCP**、**DZCPP** 指令才可使用 **HC** 装置。

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。



3. 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。

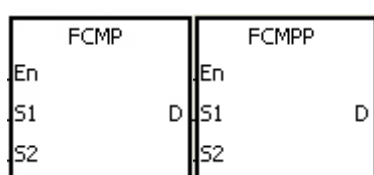
6

API	指令码		操作数				功能			
0056	FCMP	P	S ₁ · S ₂ · D				浮点数比较			

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

脉冲执行型	32 位指令 (7-9 steps)	64 位指令
AH500	AH500	-

符号：



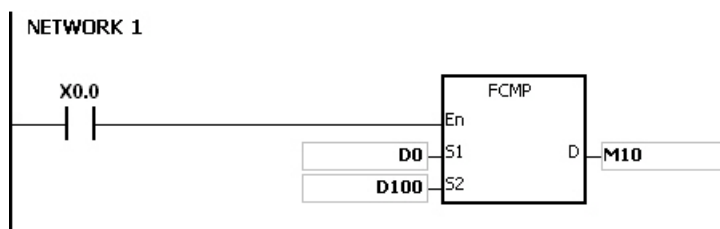
S₁ : 浮点数比较值 1 Double Word
S₂ : 浮点数比较值 2 Double Word
D : 比较结果 Bit

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：

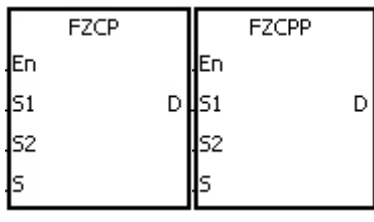
1. 当 S₁ 或 S₂ 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。
2. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of BOOL。
3. D+2 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码		操作数				功能			
0057	FZCP	P	S1 · S2 · S · D				浮点数区域比较			

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

脉冲执行型	32 位指令 (9-12 steps)	64 位指令
AH500	AH500	-

符号：



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

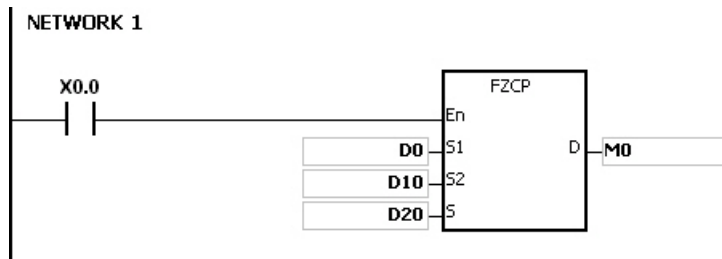
指令说明：

- 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。
- 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。
- 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。

6

程序范例：

- 若比较结果指定装置为 M0，则自动占有 M0~M2。
- 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。
- 若要清除其结果请使用 RST 或 ZRST 指令。



补充说明：

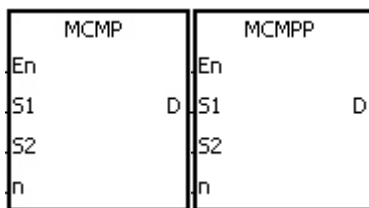
- 当 S₁ 或 S₂ 或 S 的内容值超出浮点数可以表示的范围时，接点不导通，SM0=ON，错误码 SR0=16#2013。
- D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of BOOL。
- D+2 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码		操作数								功能							
0058		MCMP	P	S1 · S2 · n · D								矩阵比较						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



- S₁ : 矩阵来源装置 1 Word
- S₂ : 矩阵来源装置 2 Word
- n : 数组长度 Word
- D : 指针 Pr 用以存放目标的地址值 Word

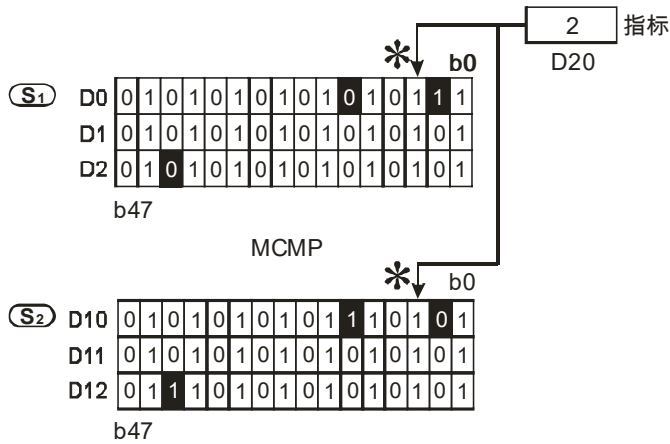
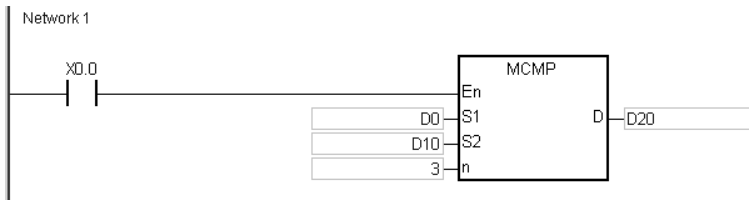
指令说明：

- 每次比较依两个矩阵来源指针 Pr 地址，将 S₁ 或 S₂ 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。
- n 操作数的范围为 1~256。
- 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 的值超过范围时指标错误标志 SM611=1。
- 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)，由使用者于指令中指定，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标错误标志 SM611 设为 1，且本指令不执行。
- 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，故由指标当时值加 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。
- 设指标当时值 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：

1. 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。

2. 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： 矩阵指标错误标志，指标 Pr 值超出此范围则 SM611=1



API	指令码			操作数								功能			
0059~0064		CMPT※	P	S1 · S2 · n · D								表格比较指令			

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

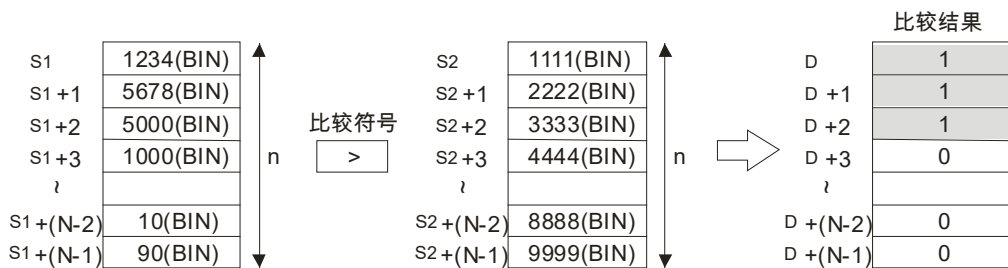
脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：

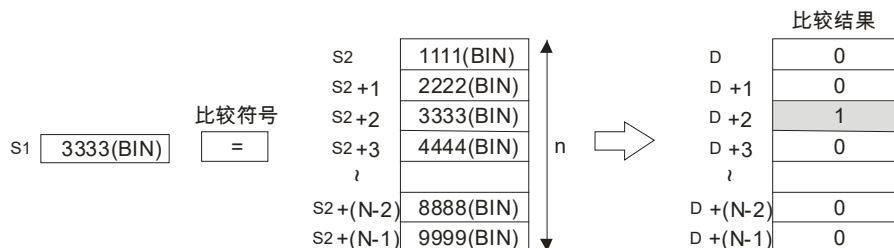
CMPT=		CMPT=P		S ₁ : 来源起始装置 1 Word	
En		En			S ₂ : 来源起始装置 2 Word
S1	D	S1	D		
S2		S2			
n		n		D : 目标装置。 Bit	

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：



- 如果 S₁ 的来源直接指定为 -32768~32767，则比较方式如下：

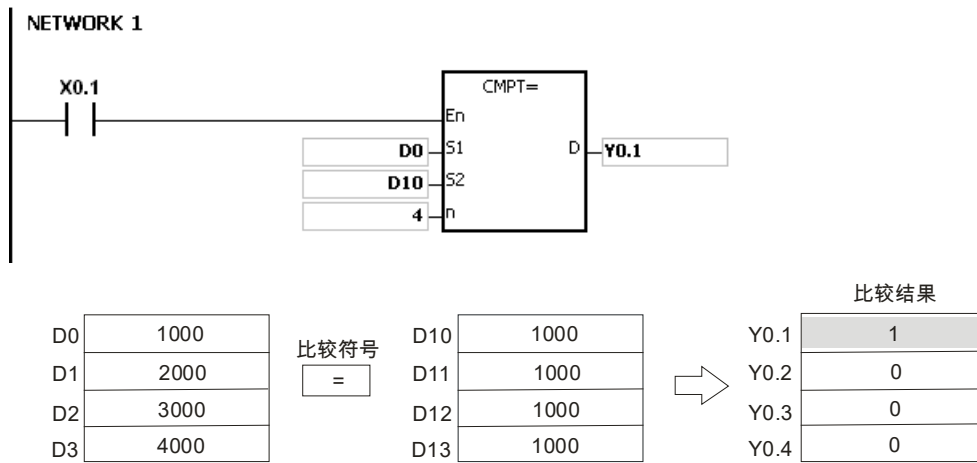


7. 各指令比较结果如下：

API No.	16 位指令	比较结果	
		1	0
0059	CMPT =	$S_1 = S_2$	$S_1 \neq S_2$
0060	CMPT < >	$S_1 \neq S_2$	$S_1 = S_2$
0061	CMPT >	$S_1 > S_2$	$S_1 \leq S_2$
0062	CMPT > =	$S_1 \geq S_2$	$S_1 < S_2$
0063	CMPT <	$S_1 < S_2$	$S_1 \geq S_2$
0064	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 Error Code=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。

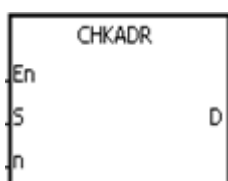
6

API	指令码	操作数	功能
0065	CHKADR	S · n · D	接点类型指针寄存器地址检查

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：



S : 指针寄存器

POINTER/T_POINTER/
C_POINTER/HC_POINTER

n : 装置个数

Word

D : 检查结果

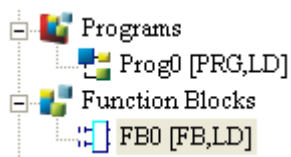
Bit

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。



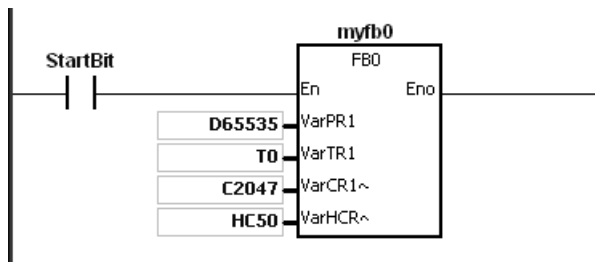
并且在 Program 中宣告两个变量如下：

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

- 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。

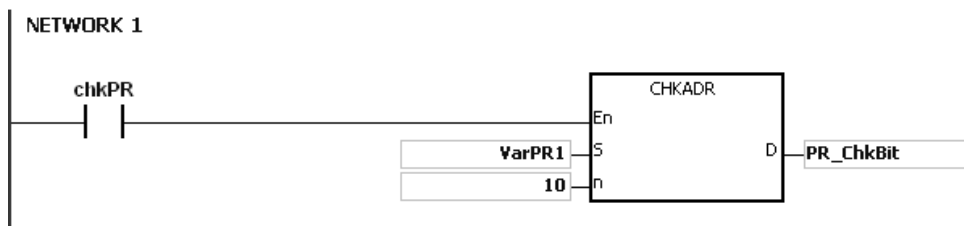
Local Symbols				
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR_IN_OUT	VarPR1	N/A [Auto]	POINTER	N/A
VAR_IN_OUT	VarTR1	N/A [Auto]	T_POINTER	N/A
VAR_IN_OUT	VarCR1	N/A [Auto]	C_POINTER	N/A
VAR_IN_OUT	VarHCR1	N/A [Auto]	HC_POINTER	N/A
VAR	PR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	TR_ChkBit	N/A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

- 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。

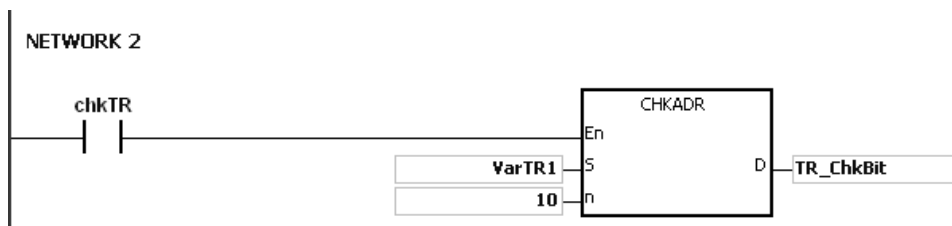


6

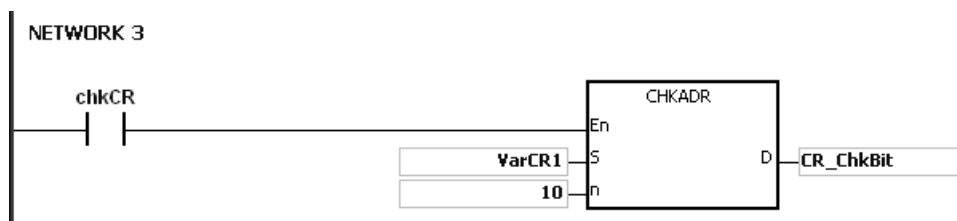
- 在 FB0 中使用 CHKADR 来检查 VarPR1、VarTR1、VarCR1、VarHCR1 是否超出范围。
- 当 chkPR=ON，VarPR1 所代表的实际装置为 D65535，而 D 装置的合法范围为 D0~D65535，所以 $D65535+10-1=D65544$ 已超出范围，PR_ChkBit=OFF。



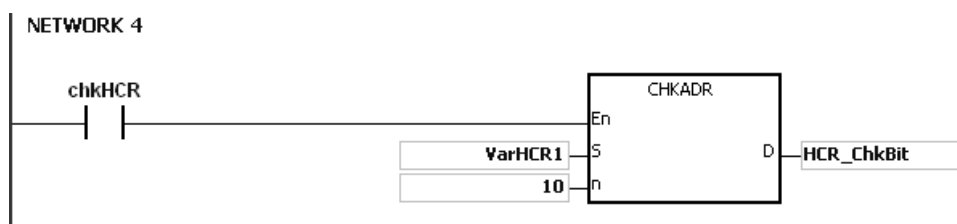
- 当 chkTR=ON，VarTR1 所代表的实际装置为 T0，而 T 装置的合法范围为 T0~T2047，所以 $T0+10-1=T9$ 没有超出范围，TR_ChkBit=ON。



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 $SR0=16\#200B$ 。

6.2 四则运算指令

6.2.1 四则运算指令一览表

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
0100	+	D+	-	✓	BIN 加法	7
0101	-	D-	-	✓	BIN 减法	7
0102	*	D*	-	✓	BIN 乘法	7
0103	/	D/	-	✓	BIN 除法	7
0104	-	F+	DF+	✓	浮点数加法	7-9
0105	-	F-	DF-	✓	浮点数减法	7-9
0106	-	F*	DF*	✓	浮点数乘法	7-9
0107	-	F/	DF/	✓	浮点数除法	7-9
0108	B+	DB+	-	✓	BCD 加法	7
0109	B-	DB-	-	✓	BCD 减法	7
0110	B*	DB*	-	✓	BCD 乘法	7
0111	B/	DB/	-	✓	BCD 除法	7
0112	BK+	-	-	✓	连续区块 BIN 加法	9
0113	BK-	-	-	✓	连续区块 BIN 减法	9
0114	\$+	-	-	✓	字符串链接	7-19
0115	INC	DINC	-	✓	BIN 加一	3
0116	DEC	DDEC	-	✓	BIN 减一	3
0117	MUL16	MUL32	-	✓	16 位专用 BIN 乘法/ 32 位专用 BIN 乘法	7
0118	DIV16	DIV32	-	✓	16 位专用 BIN 除法/ 32 位专用 BIN 除法	7

6

6.2.2 四则运算指令说明

API	指令码			操作数							功能						
0100	D	+	P	$S_1 \cdot S_2 \cdot D$							BIN 加法						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

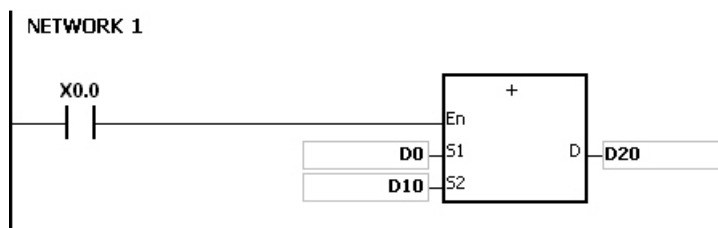


指令说明：

- S_1 及 S_2 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 零标志 (Zero flag)，SM601 借位标志 (Borrow flag)，SM602 进位标志 (Carry flag)。
- 运算结果为 0 时，零标志 (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，进位标志 (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，进位标志 (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。

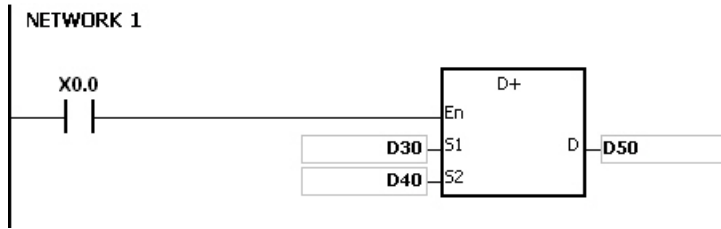


- 当 D0=100，D10=10，D0+D10=110，D20=110。

- 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 · 则进位标志 $SM602=ON$ · $D20=16\#0$ · 因为运算结果为 $16\#0$ · 所以零标志 $SM600=ON$ 。

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 ($D31$ 、 $D30$) = $16\#80000000$ · ($D41$ 、 $D40$) = $16\#FFFFFF$ · ($D31$ 、 $D30$) + ($D41$ 、 $D40$) = $16\#17FFFFFF$ · 此时运算结果超出 32 位 BIN 可表示范围 · 则进位标志 $SM602=ON$ · ($D51$ 、 $D50$) = $16\#7FFFFFFF$ 。

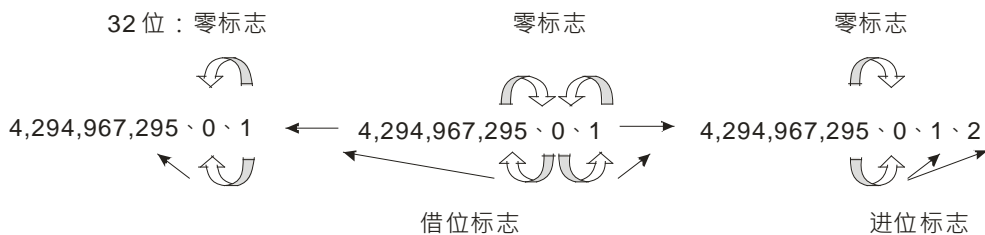
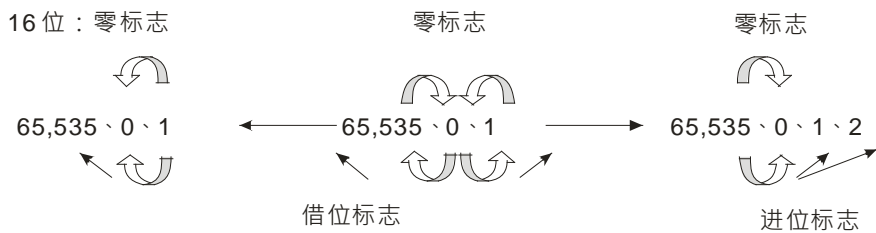
标志操作：

16 位指令：

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

32 位指令：

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



6

API	指令码			操作数						功能					
0101	D	-	P	$S_1 \cdot S_2 \cdot D$						BIN 减法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

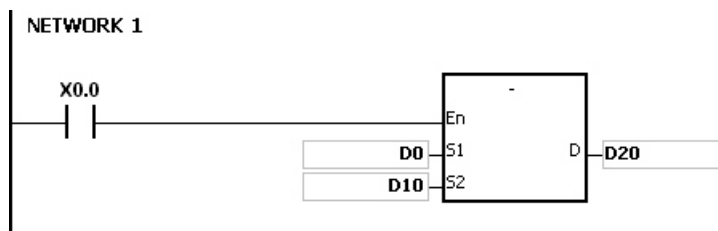


指令说明：

- S_1 及 S_2 以 BIN 方式相减的结果存于 D。
- 32 位指令才可以使用 HC 装置。
- 标志：SM600 零标志 (Zero flag) · SM601 借位标志 (Borrow flag) · SM602 进位标志 (Carry flag)。
- 运算结果为 0 时，零标志 (Zero flag) SM600 为 ON，否则为 OFF。
- 运算发生借位时，借位标志 (Borrow flag) SM601 为 ON，否则为 OFF。

程序范例一：

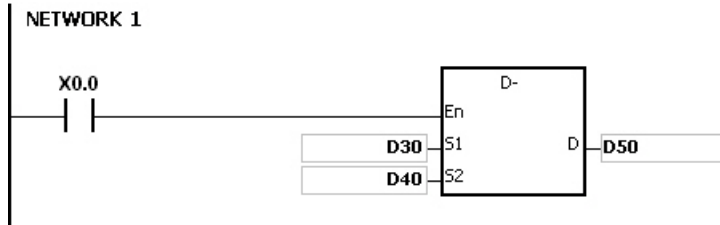
16 位 BIN 减法：当 X0.0=ON 时，将 D0 内容减掉 D10 内容将差存在 D20 之内容中，如下说明：



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

程序范例二：

32 位 BIN 减法：当 X0.0=ON 时，(D31、D30) 内容减掉 (D41、D40) 之内容将差存在 (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#FFFFFF，(D31、D30) - (D41、D40) = 16#F8000001，运算发生借位，借位标志 SM601=ON，(D51、D50) = 16#80000001。

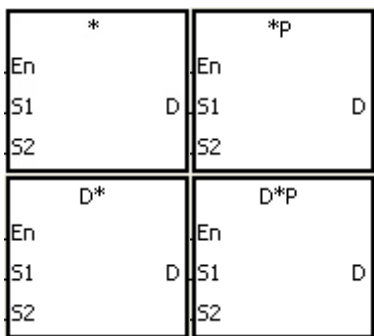
6

API	指令码			操作数						功能					
0102	D	*	P	$S_1 \cdot S_2 \cdot D$						BIN 乘法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

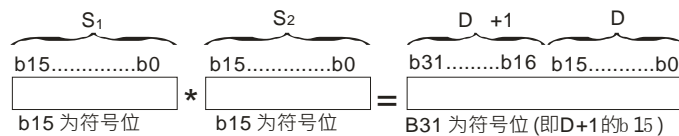
符号：



S_1 : 被乘数 Word/Double Word
 S_2 : 乘数 Word/Double Word
D : 积 Double Word/Long Word

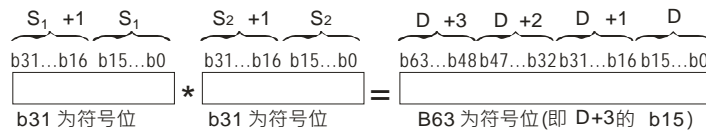
指令说明：

- S_1 及 S_2 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。

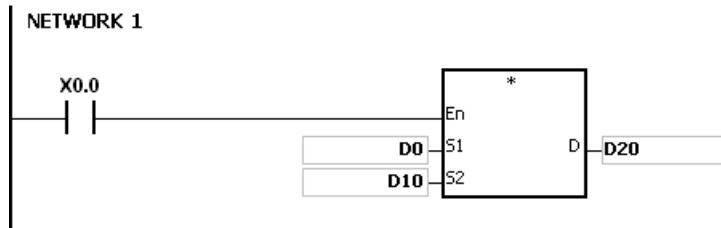
- 32 位 BIN 乘法运算：



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 位之积，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) 之 OFF/ON 来指示。OFF 表示正的 (0)，同时 ON 表示负的 (1)。



$$D0 \times D10 = (D21 \cdot D20)$$

16 位 × 16 位 = 32 位

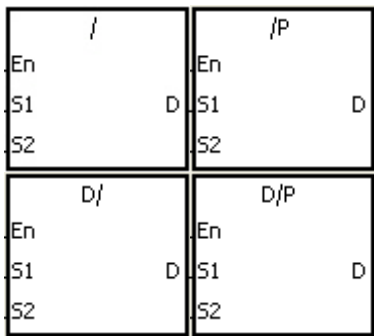
6

API	指令码			操作数						功能					
0103	D	/	P	$S_1 \cdot S_2 \cdot D$						BIN 除法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

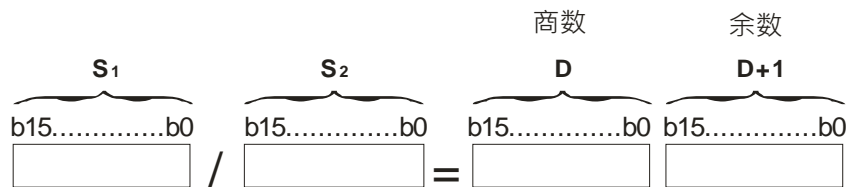
符号：



- S_1 : 被除数 Word/Double Word
- S_2 : 除数 Word/Double Word
- D : 商及余数 Word/Double Word

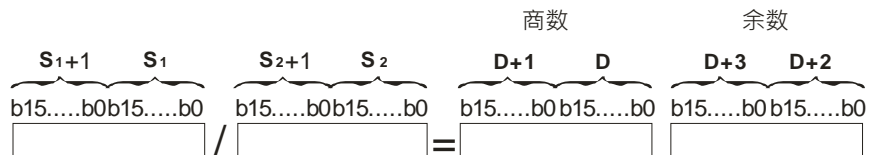
指令说明：

1. S_1 及 S_2 以有符号二进制方式相除后的商及余数存于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 操作数连续占用两个，D 储存商，D+1 储存余数。

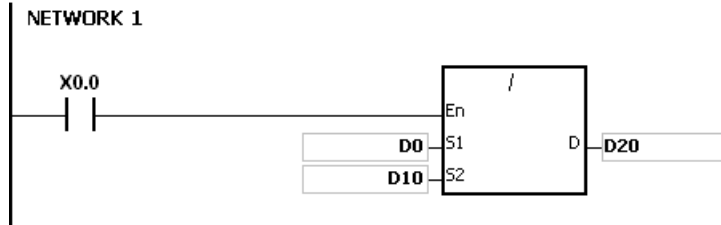
5. 32 位 BIN 除法运算：



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。所得结果之正负由最高位位之 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。

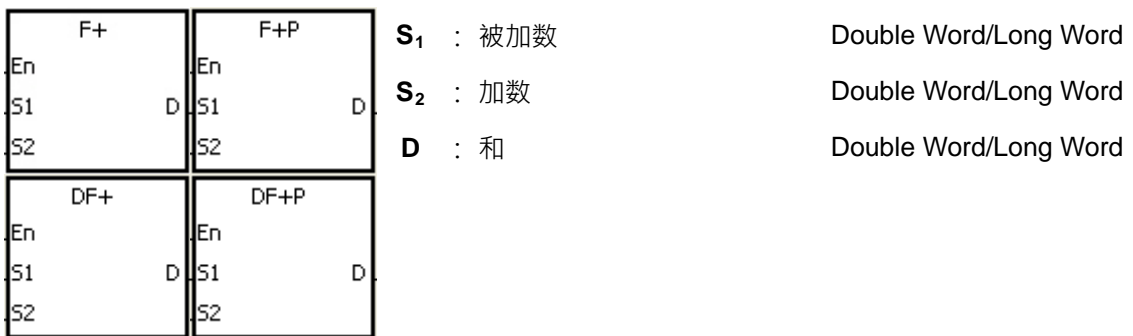
6

API	指令码			操作数							功能				
0104	D	F+	P	$S_1 \cdot S_2 \cdot D$							浮点数加法				

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

脉冲执行型	32 位指令 (7-9 steps)	64 位指令 (7-9 steps)
AH500	AH500	AH500

符号：

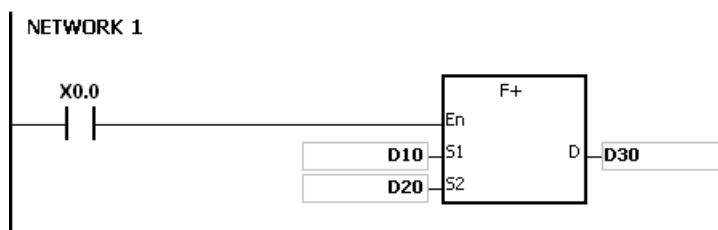


指令说明：

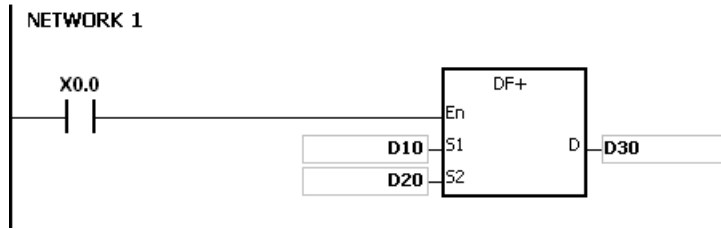
- S_1 及 S_2 以浮点数方式相加的结果存于D。
- 32 位单精度浮点数加法：
 - 运算结果为 0 时，零标志 (Zero flag) SM600 为 ON。
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FF7FFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，则 D=16#7F7FFFFFFF。
- 64 位双精度浮点数加法：
 - 运算结果为 0 时，零标志 (Zero flag) SM600 为 ON。
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，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) 之内容当中。



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 之内容将结果存在 (D33 · D32 · D31 · D30) =16# 40123851EB851EB8 (表浮点值 4.555) 之内容当中。



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。

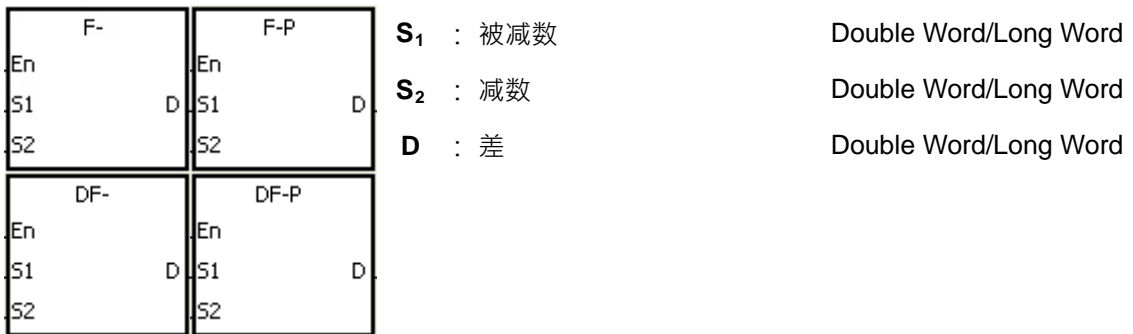
6

API	指令码			操作数						功能					
0105	D	F-	P	$S_1 \cdot S_2 \cdot D$						浮点数减法					

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

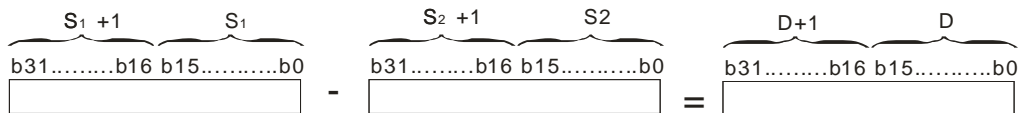
脉冲执行型	32 位指令 (7-9 steps)	64 位指令 (7-9 steps)
AH500	AH500	AH500

符号：

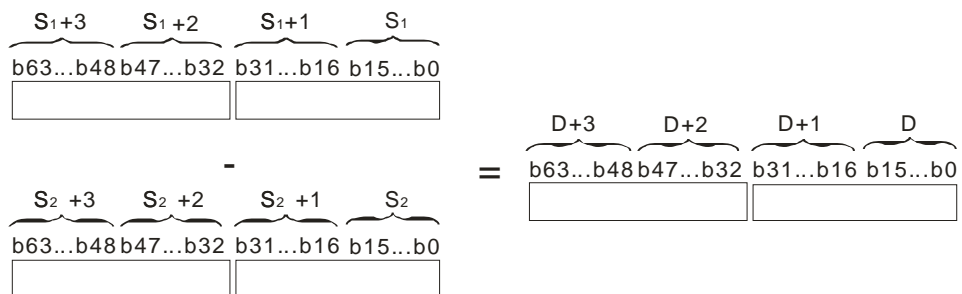


指令说明：

- S_1 及 S_2 以浮点数类型相减的结果存于D。
- 若运算结果为0，则零标志 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。

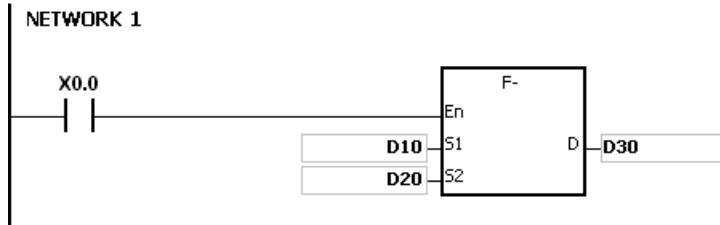


- 64 位双精度浮点数减法：
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

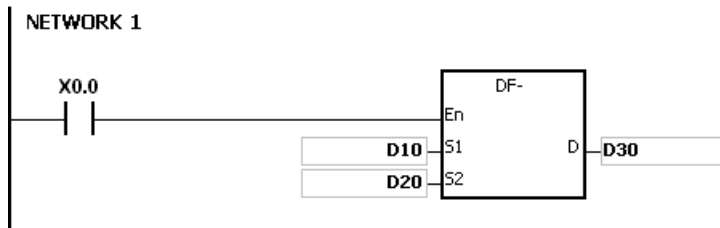


程序范例：

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) 之内容，将结果存在 (D31 · D30) 之内容当中。



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) 之内容，将结果存在 (D33 · D32 · D31 · D30) 之内容当中。



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。

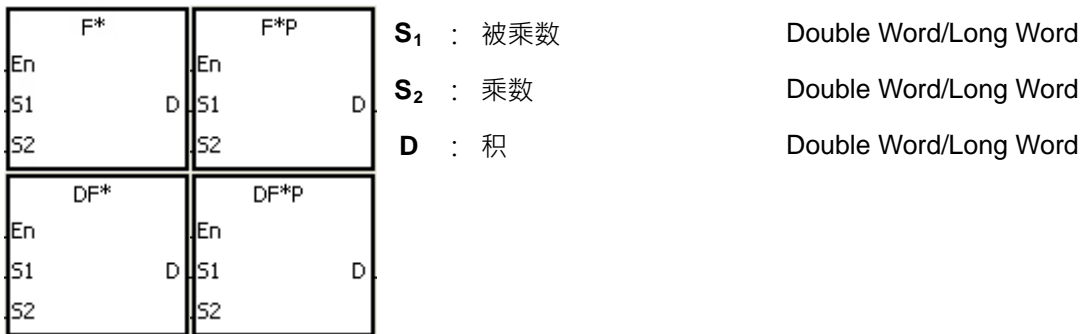
6

API	指令码			操作数						功能					
0106	D	F*	P	$S_1 \cdot S_2 \cdot D$						浮点数乘法					

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

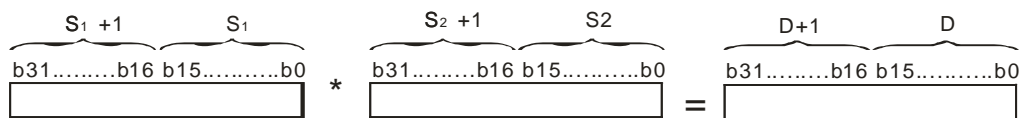
脉冲执行型	32 位指令 (7-9 steps)	64 位指令 (7-9 steps)
AH500	AH500	AH500

符号：

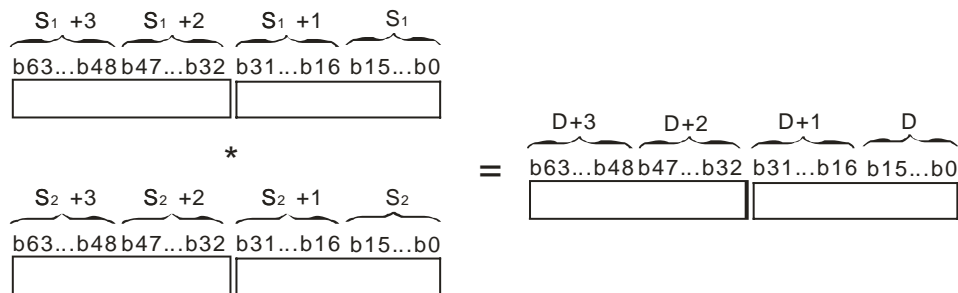


指令说明：

- S_1 及 S_2 以浮点数类型相乘的结果存于D。
- 若运算结果为0，则零标志 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。

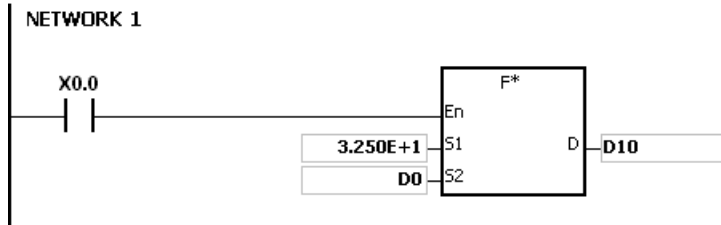


- 64 位双精度浮点数乘法：
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

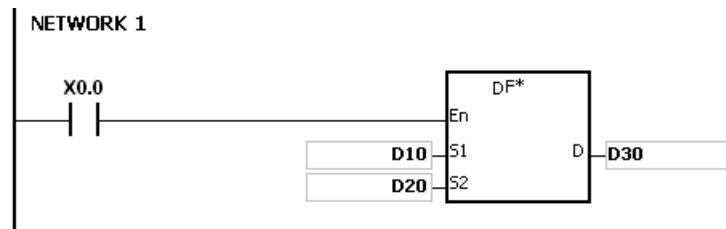


程序范例：

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) 之内容，并将结果存在 (D11 · D10) 之内容当中。



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



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。

6

API	指令码			操作数						功能					
0107	D	F/	P	$S_1 \cdot S_2 \cdot D$						浮点数除法					

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

脉冲执行型	32 位指令 (7-9 steps)	64 位指令 (7-9 steps)
AH500	AH500	AH500

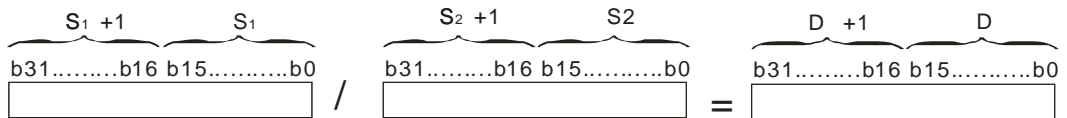
符号：

F/	F/P	S_1 : 被除数 Double Word/Long Word	
En	En		S_2 : 除数 Double Word/Long Word
S1	S1		
S2	S2		
		D : 商 Double Word/Long Word	

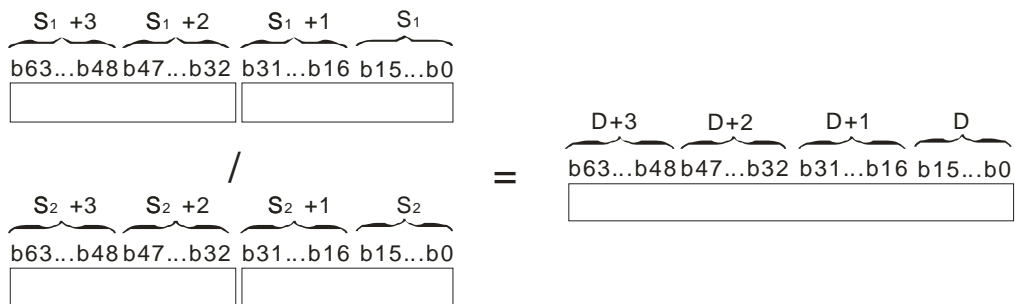
DF/	DF/P
En	En
S1	S1
S2	S2

指令说明：

- S_1 及 S_2 以单精度浮点数类型相除的结果存于D。
- 若运算结果为0，则零标志 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。

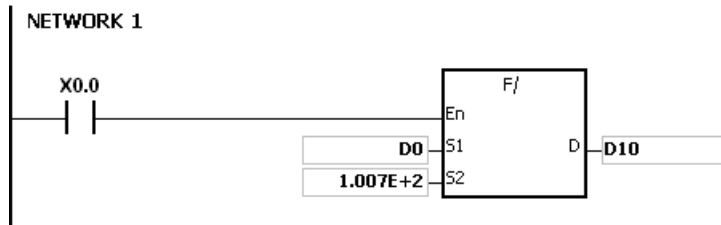


- 64 位双精度浮点数除法：
 - 若运算结果的绝对值小于可表示之最小浮点值，则 D=16#FFEFFFFFFFFFFFFFFF。
 - 若运算结果的绝对值大于可表示之最大浮点值，则 D=16#7FEFFFFFFFFFFFFFFF。

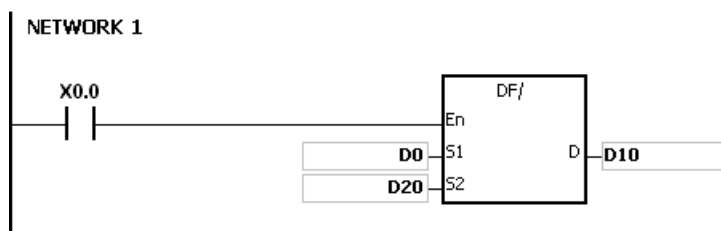


程序范例：

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



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) 之内容，并将结果存在 (D13 · D12 · D11 · D10) 的内容当中。



补充说明：

1. 除数为 0 时，指令不执行，SM0=ON，错误码 SR0=16#2012。
2. 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。

6

API	指令码			操作数						功能					
0108	D	B+	P	$S_1 \cdot S_2 \cdot D$						BCD 加法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

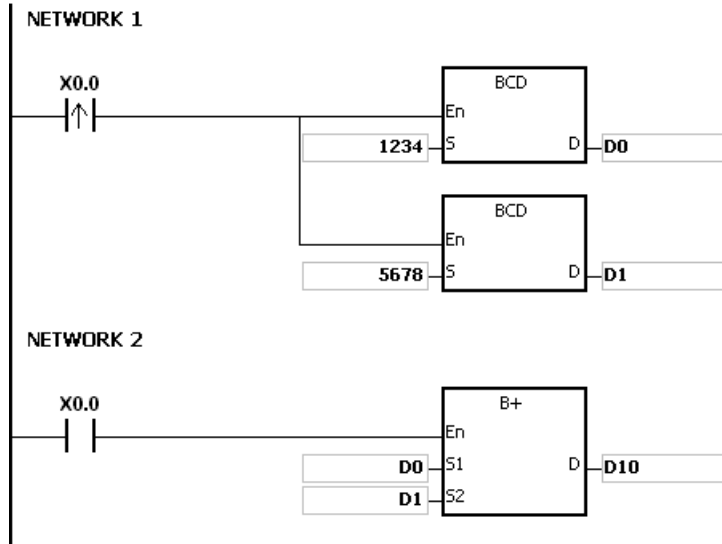
B+	B+P	S_1 : 被加数 Word/Double Word
En S1 S2	En S1 S2	
DB+	DB+P	S_2 : 加数 Word/Double Word
En S1 S2	En S1 S2	
		D : 和 Word/Double Word

指令说明：

- S_1 及 S_2 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 $S_1=9999$ (BCD) · $S_2=0002$ (BCD) · 则 $S_1+S_2=10001$ (BCD) · 则进位被忽略 · $D=0001$ (BCD)。
- 32 位 BCD 加法：
当 $S_1=99999999$ (BCD) · $S_2=00000002$ (BCD) · 则 $S_1+S_2=100000001$ (BCD) · 则进位被忽略 · $D=00000001$ (BCD)。

程序范例：

当 X0.0=ON 时，各别将常数 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 S1 或 S2 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 S1 或 S2 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。



API	指令码			操作数						功能					
0109	D	B-	P	$S_1 \cdot S_2 \cdot D$						BCD 减法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

En	B-	En	B-P	S_1 : 被减数 Word/Double Word S_2 : 减数 Word/Double Word D : 差 Word/Double Word
S1	D	S1	D	
S2		S2		
En	DB-	En	DB-P	
S1	D	S1	D	
S2		S2		

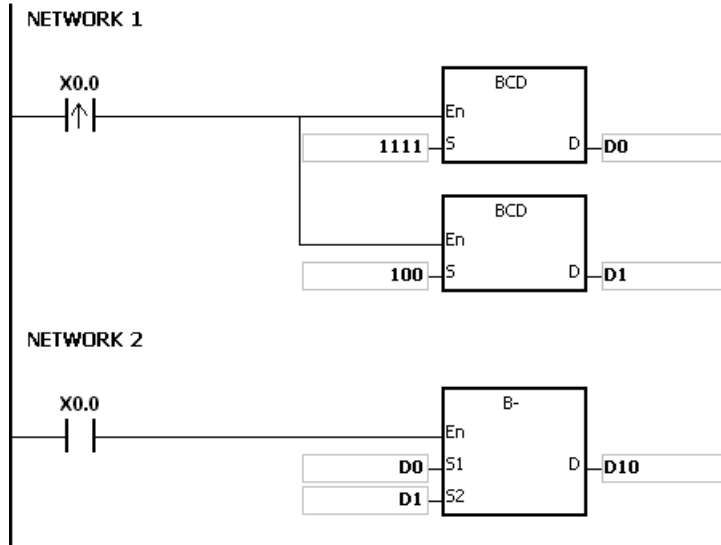
指令说明：

1. S_1 及 S_2 以BCD方式相减的结果存于D。
2. DB-指令才可以使用 HC 装置。
3. BCD 值以 HEX 表示且每一个位是都要在 0~9 之间。
4. 16 位 BCD 减法：
 - 当 $S_1=9999$ (BCD) · $S_2=9998$ (BCD) · 则 $S_1-S_2=0001$ (BCD) · 则D=0001 (BCD) 。
 - 当 $S_1=0001$ (BCD) · $S_2=9999$ (BCD) · 则 $S_1-S_2=-9998$ (BCD) · 则D=0002 (BCD) 。
 - 当 $S_1=0001$ (BCD) · $S_2=0004$ (BCD) · 则 $S_1-S_2=-0003$ (BCD) · 则D=9997 (BCD) 。
5. 32 位 BCD 减法：
 - 当 $S_1=99999999$ (BCD) · $S_2=99999998$ (BCD) · 则 $S_1-S_2=00000001$ (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) 。

6

程序范例：

当 X0.0=ON 时，各别将常数 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。

6

API	指令码			操作数						功能					
0110	D	B*	P	$S_1 \cdot S_2 \cdot D$						BCD 乘法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

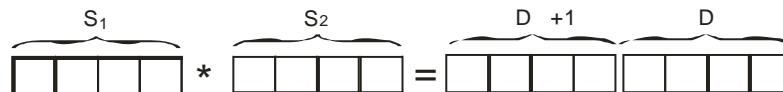
符号：

<table border="1" style="width: 100%; height: 100%;"> <tr><th colspan="2">B*</th></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> </table>	B*		En		S1	D	S2		<table border="1" style="width: 100%; height: 100%;"> <tr><th colspan="2">B*P</th></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> </table>	B*P		En		S1	D	S2		S_1 : 被乘数	Word/Double Word
B*																			
En																			
S1	D																		
S2																			
B*P																			
En																			
S1	D																		
S2																			
<table border="1" style="width: 100%; height: 100%;"> <tr><th colspan="2">DB*</th></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> </table>	DB*		En		S1	D	S2		<table border="1" style="width: 100%; height: 100%;"> <tr><th colspan="2">DB*P</th></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> </table>	DB*P		En		S1	D	S2		S_2 : 乘数	Word/Double Word
DB*																			
En																			
S1	D																		
S2																			
DB*P																			
En																			
S1	D																		
S2																			
		D : 积	Double Word/Long Word																

指令说明：

- S_1 及 S_2 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：

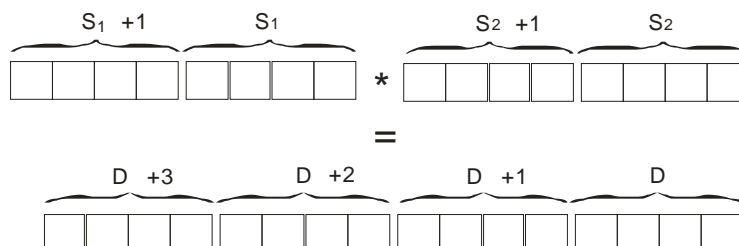
- 当 $S_1=1234$ (BCD) · $S_2=5678$ (BCD) · 则 $D=07006652$ (BCD)。



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：

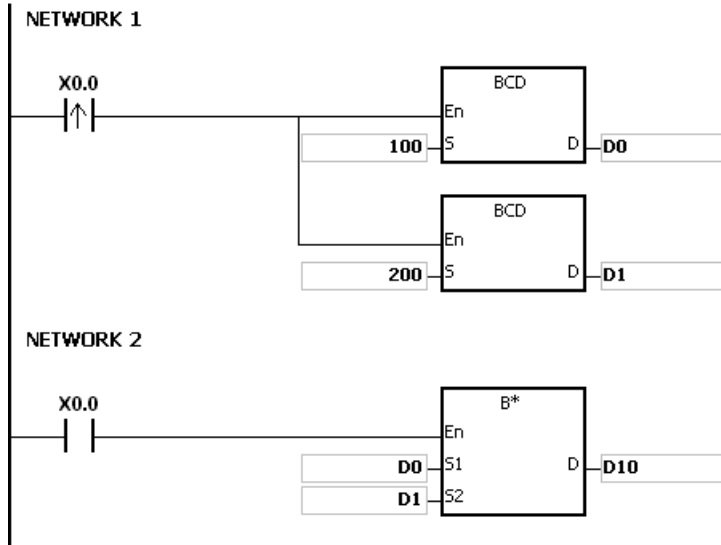
- 当 $S_1=99999999$ (BCD) · $S_2=99999998$ (BCD) · 则 $D=9999999700000002$ (BCD)。



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：

当 X0.0=ON 时，各别将常数 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 S_1 或 S_2 超出 BCD 可表示范围时 0~9999，指令不执行，SM0=ON，Error Code=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，Error Code=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。

6

API	指令码			操作数						功能					
0111	D	B/	P	$S_1 \cdot S_2 \cdot D$						BCD 除法					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

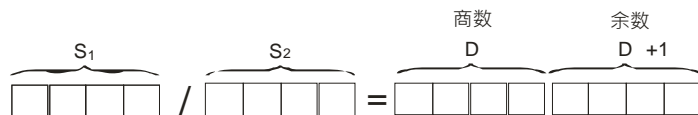
B/		B/P	
En		En	
S1	D	S1	D
S2		S2	

DB/		DB/P	
En		En	
S1	D	S1	D
S2		S2	

S_1 : 被除数 Word/Double Word
 S_2 : 除数 Word/Double Word
D : 商及余数 Word/Double Word

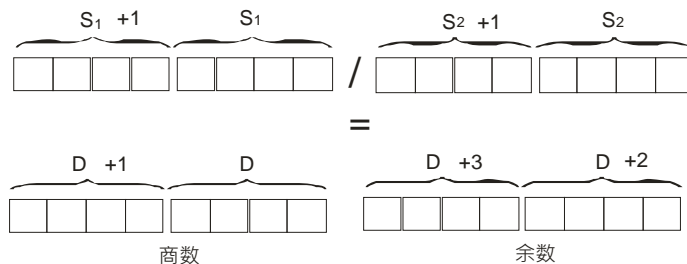
指令说明：

- S_1 及 S_2 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 $S_1=5678$ (BCD) · $S_2=1234$ (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。

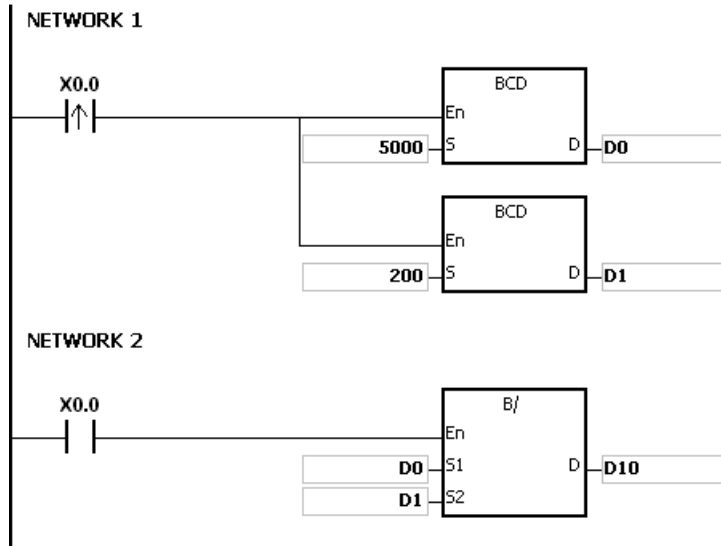
- 32 位 BCD 除法：
 - 当 $S_1=87654321$ (BCD) · $S_2=12345678$ (BCD) · 则 $(D+1 \cdot D)=00000007$ (BCD) · $(D+3 \cdot D+2)=01234575$ 。



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，各别将常数 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，Error Code=16#2012。
2. 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，Error Code=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 S₁ 或 S₂ 超出 BCD 可表示范围时 0~99999999，指令不执行，SM0=ON，Error Code=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。

6

API	指令码			操作数						功能					
0112	BK+	P		$S_1 \cdot S_2 \cdot n \cdot D$						连续区块 BIN 加法					

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">BK+</td></tr> <tr><td style="text-align: center;">En</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">n</td></tr> </table>	BK+	En	S1	S2	n	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="text-align: center;">BK+P</td></tr> <tr><td style="text-align: center;">En</td></tr> <tr><td style="text-align: center;">S1</td></tr> <tr><td style="text-align: center;">S2</td></tr> <tr><td style="text-align: center;">n</td></tr> </table>	BK+P	En	S1	S2	n	<p>S_1 : 被加数 Word</p> <p>S_2 : 加数 Word</p> <p>n : 要相加的笔数 Word</p> <p>D : 和 Word</p>
BK+												
En												
S1												
S2												
n												
BK+P												
En												
S1												
S2												
n												

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，零标志 (Zero flag) SM600 为 ON。
4. 运算结果小于 -32,768 时，借位标志 (Borrow flag) SM601 为 ON。
5. 运算结果大于 32,767 时，进位标志 (Carry flag) SM602 为 ON。
6. 当 S_2 是装置 (非 K/16#) 的时候：

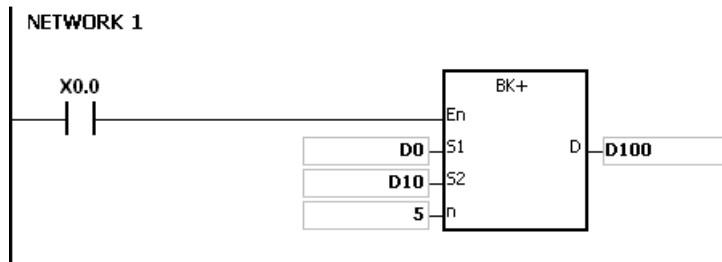
$$\begin{array}{c} S_1 \\ S_1+1 \\ \vdots \\ S_1+n-1 \end{array} \left. \begin{array}{c} 1 \\ 2 \\ \vdots \\ 5 \end{array} \right\} n + \begin{array}{c} S_2 \\ S_2+1 \\ \vdots \\ S_2+n-1 \end{array} \left. \begin{array}{c} 1 \\ 2 \\ \vdots \\ 5 \end{array} \right\} n = \begin{array}{c} D \\ D+1 \\ \vdots \\ D+n-1 \end{array} \left. \begin{array}{c} 1+1=2 \\ 2+2=4 \\ \vdots \\ 5+5=10 \end{array} \right\} n$$

7. 当 S_2 是常数 K/16# 的时候：

$$\begin{array}{c} S_1 \\ S_1+1 \\ \vdots \\ S_1+n-1 \end{array} \left. \begin{array}{c} 1 \\ 2 \\ \vdots \\ 5 \end{array} \right\} n + \begin{array}{c} S_2 \\ S_2 \\ \vdots \\ S_2 \end{array} \left. \begin{array}{c} 10 \\ 10 \\ \vdots \\ 10 \end{array} \right\} n = \begin{array}{c} D \\ D+1 \\ \vdots \\ D+n-1 \end{array} \left. \begin{array}{c} 1+10=11 \\ 2+10=12 \\ \vdots \\ 5+10=15 \end{array} \right\} n$$

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。

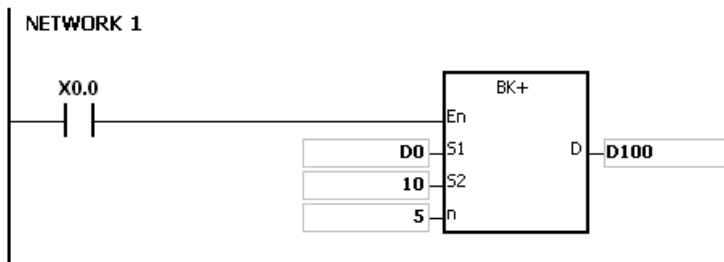


执行结果

D0	1	D10	10	D100	11
D1	2	D11	11	D101	13
D2	3	D12	12	D102	15
D3	4	D13	13	D103	17
D4	5	D14	14	D104	19

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。



执行结果

D0	1	10	D100	11
D1	2	10	D101	12
D2	3	10	D102	13
D3	4	10	D103	14
D4	5	10	D104	15

6

补充说明：

1. S₁、S₂或D装置的起始地址~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 S₁~S₁+n-1 的装置与 D~D+n-1 的装置有重复时，指令不执行，SM0=ON，错误码 SR0=16#200C。
4. 若 S₂~S₂+n-1 的装置与 D~D+n-1 的装置有重复时，指令不执行，SM0=ON，错误码 SR0=16#200C。
5. 若 S₁~S₁+n-1 的装置与 S₂~S₂+n-1 的装置有重复时，指令不执行，SM0=ON，错误码 SR0=16#200C。

API	指令码			操作数						功能							
0113		BK-	P	$S_1 \cdot S_2 \cdot n \cdot D$						连续区块 BIN 减法							

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

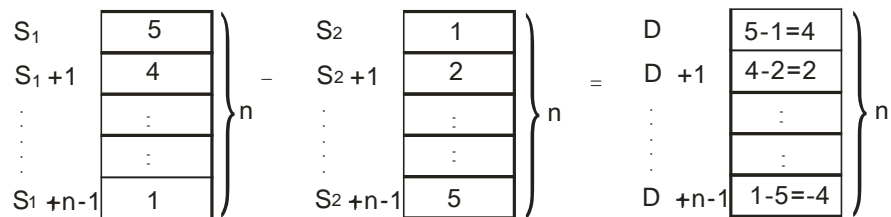
脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：

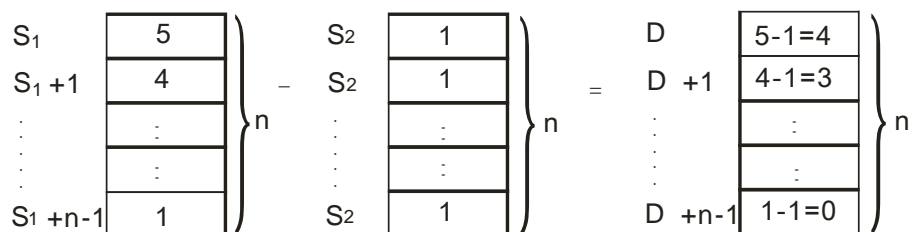
BK-		BK-P		S_1 : 被减数	Word
En		En		S_2 : 减数	Word
S1	D	S1	D	n : 要相减的笔数	Word
S2		S2		D : 差	Word
n		n			

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，零标志 (Zero flag) SM600为ON。
4. 运算结果小于-32,768时，借位标志 (Borrow flag) SM601为ON。
5. 运算结果大于32,767时，进位标志 (Carry flag) SM602为ON。
6. 当 S_2 是装置 (非K/16#) 的时候：

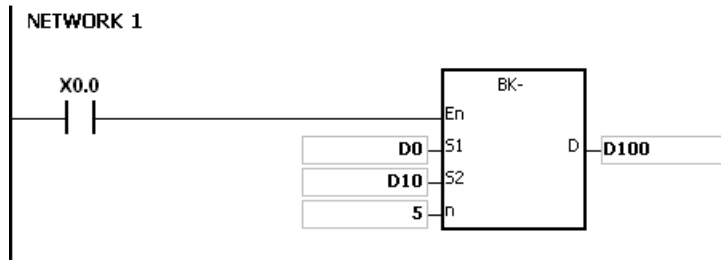


7. 当 S_2 是常数K/16#的时候：



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。

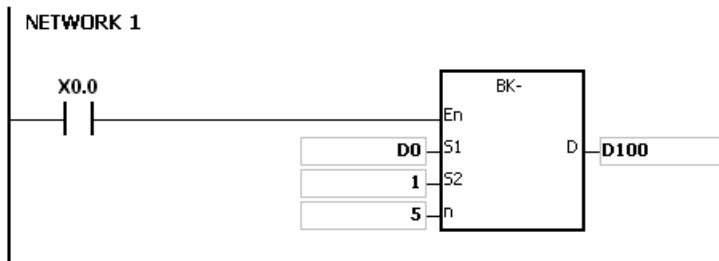


执行结果

D0	5	D10	1	D100	4	1 SM600
D1	4	D11	2	D101	2	
D2	3	D12	3	D102	0	
D3	2	D13	4	D103	-2	
D4	1	D14	5	D104	-4	

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。



执行结果

D0	10	1	D100	9
D1	9	1	D101	8
D2	8	1	D102	7
D3	7	1	D103	6
D4	6	1	D104	5

补充说明：

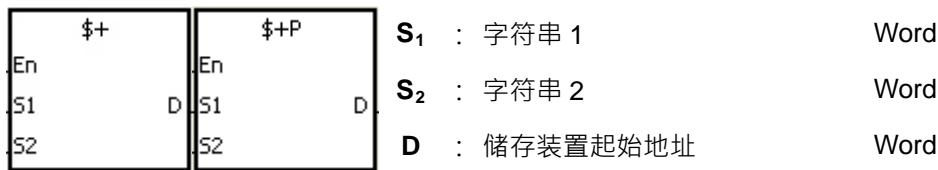
1. S₁、S₂或D装置的起始地址~n-1 的地址，超出装置范围时，指令不执行SM0=ON，Error Code=16#2003。
2. 若 n<1 或 n>256 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 若 S₁~S₁+n-1 的装置与 D~D+n-1 的装置有重复时，指令不执行，SM0=ON，错误码 SR0=16#200C。
4. 若 S₂~S₂+n-1 的装置与 D~D+n-1 的装置有重复时，指令不执行，SM0=ON，错误码 SR0=16#200C。
5. 若 S₁~S₁+n-1 的装置与 S₂~S₂+n-1 的装置有重复时，指令不执行，SM0=ON，错误码 SR0=16#200C。

API	指令码			操作数								功能			
0114		\$+	P	S ₁ · S ₂ · D								字符串链接			

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

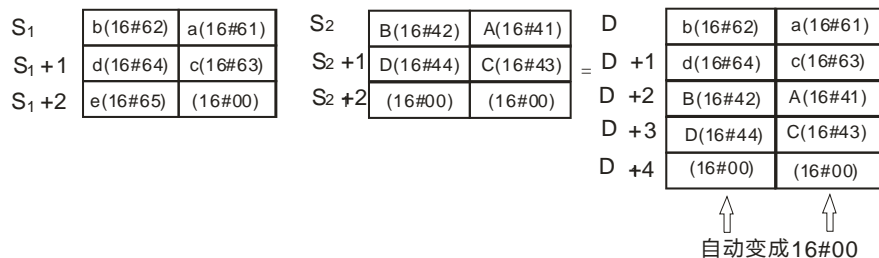
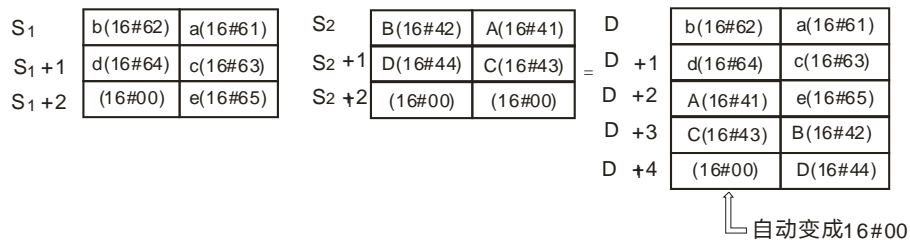
脉冲执行型	16 位指令 (7-19 steps)	32 位指令
AH500	AH500	-

符号 :



指令说明 :

1. 该指令执行时, 将S₁操作数S₁~16#00 (不搬 16#00) 与S₂操作数S₂~ 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。
2. S₁操作数及S₂操作数两个字符串链接的结果存于D操作数的说明如下 :



3. 若S₁或S₂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S₁或S₂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。

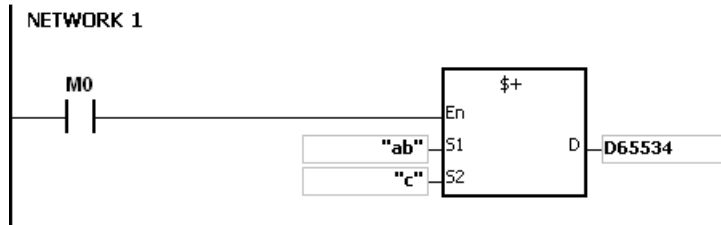
6

S ₁	b(16#62)	a(16#61)
S ₁ +1	d(16#64)	c(16#63)
S ₁ +2	(16#00)	e(16#65)

5. 字符串“abcde”在S₁运算元表示为：

程序范例：

给定 S1=“ab”，S2=“C”，启动条件接点 M0 后，D65534=16#6261，D65535=16#0063



补充说明：

1. 若S₁或S₂操作数为字符串时，S₁或S₂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

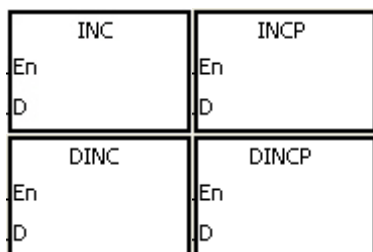
范例：字符串+“ABCDE” D0 D100 则 STEP 数=1 (指令) +3 (字符串) +2 (D0) +2 (D100) =8

2. 当D装置的空间不足以存放S₁+S₂字符串时，指令不执行，SM0=ON，错误码SR0=16#2003。
3. S₁或S₂装置与D装置有重复时，指令不执行，SM0=ON，错误码SR0=16#200C。
4. S₁或S₂没有 16#00 当结尾时，指令不执行，SM0=ON，错误码SR0=16#200E。

6

API	指令码			操作数								功能					
0115	D	INC	P	D								BIN 加 1					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型											16 位指令 (3 steps)			32 位指令 (3 steps)			
AH500											AH500			AH500			

符号：



D : 目的地装置

Word/Double Word

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。

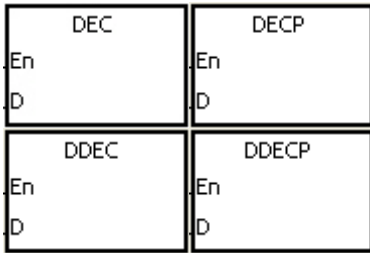
程序范例：

当 X0.0=OFF→ON 时，D0 内容自动加 1。



API	指令码			操作数								功能					
0116	D	DEC	P	D								BIN 减 1					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型											16 位指令 (3 steps)			32 位指令 (3 steps)			
AH500											AH500			AH500			

符号：



D : 目的地装置

Word/Double Word

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。



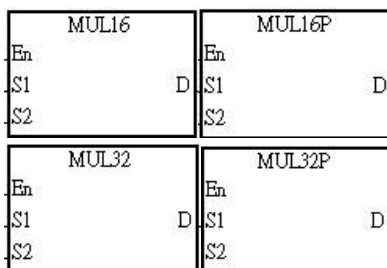
6

API	指令码			操作数							功能						
0117	MUL16	MUL32	P	$S_1 \cdot S_2 \cdot D$							16 位专用 BIN 乘法 32 位专用 BIN 乘法						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

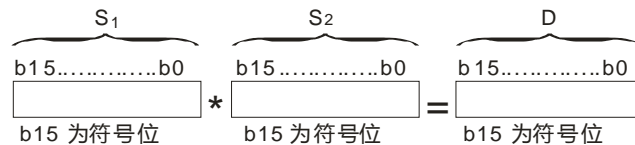
符号：



S_1 : 被乘数 Word/Double Word
 S_2 : 乘数 Word/Double Word
D : 积 Word/Double Word

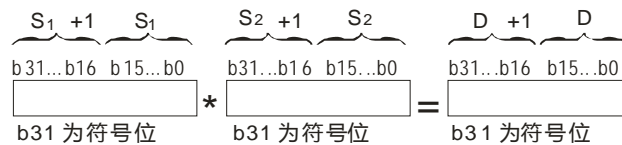
指令说明：

- S_1 及 S_2 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 位缓存器中，且符号位 b15=0 为正数，符号位 b15=1 为负数。

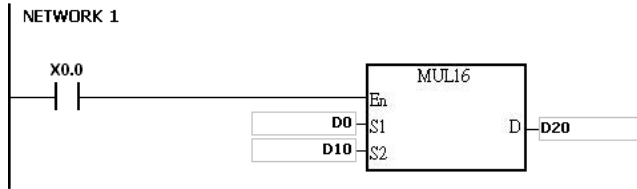
- 32 位 BIN 乘法运算：



积为 32 位数据，储存在 (D+1 · D) 组成的 32 位缓存器中，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 位之积，结果存在 D20。结果之正负由最高位(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

补充说明：

1. 当 16 位乘法之积超出 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 进位旗标为 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 位乘法之积超出 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 进位旗标为 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。

6

API	指令码			操作数							功能						
0118		DIV16 DIV32	P	$S_1 \cdot S_2 \cdot D$							16 位专用 BIN 除法 32 位专用 BIN 除法						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">DIV16</td> <td style="width: 20px;"></td> <td style="text-align: center;">DIV16P</td> </tr> <tr> <td>En</td> <td></td> <td>En</td> </tr> <tr> <td>S1</td> <td style="text-align: center;">D</td> <td>S1</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> </tr> </table>	DIV16		DIV16P	En		En	S1	D	S1	S2		S2	S_1 : 被除数	Word/Double Word
DIV16		DIV16P												
En		En												
S1	D	S1												
S2		S2												
	S_2 : 除数	Word/Double Word												
	D : 商	Word/Double Word												
<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">DIV32</td> <td style="width: 20px;"></td> <td style="text-align: center;">DIV32P</td> </tr> <tr> <td>En</td> <td></td> <td>En</td> </tr> <tr> <td>S1</td> <td style="text-align: center;">D</td> <td>S1</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> </tr> </table>	DIV32		DIV32P	En		En	S1	D	S1	S2		S2		
DIV32		DIV32P												
En		En												
S1	D	S1												
S2		S2												

指令说明：

1. S_1 及 S_2 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：

$$\begin{array}{c}
 \text{商数} \\
 \text{S}_1 \quad \text{S}_2 \quad \text{D} \\
 \text{b15.....b0} \quad \text{b15.....b0} \quad \text{b15.....b0} \\
 \boxed{} / \boxed{} = \boxed{}
 \end{array}$$

D 操作数储存商。

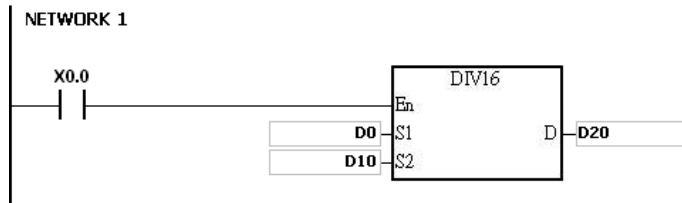
5. 32 位 BIN 除法运算：

$$\begin{array}{c}
 \text{商数} \\
 \text{S}_{1+1} \quad \text{S}_1 \quad \text{S}_{2+1} \quad \text{S}_2 \quad \text{D}_{+1} \quad \text{D} \\
 \text{b15.....b015.....b0} \quad \text{b15.....b015.....b0} \quad \text{b15.....b015.....b0} \\
 \boxed{} / \boxed{} = \boxed{}
 \end{array}$$

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。所得结果之正负由最高位位之 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. 若需要纪录余数，请改用 API0103，详细说明请参考该指令。

6

6.3 数据转换指令

6.3.1 数据转换指令一览表

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
<u>0200</u>	BCD	DBCD	—	✓	BIN→BCD 变换	5
<u>0201</u>	BIN	DBIN	—	✓	BCD→BIN 变换	5
<u>0202</u>	FLT	DFLT	—	✓	BIN 整数→单精度浮点数变换	5
<u>0203</u>	FLTD	DFLTD	—	✓	BIN 整数→双精度浮点数变换	5
<u>0204</u>	INT	DINT	—	✓	单精度浮点数→BIN 整数变换	5
<u>0205</u>	—	FINT	DFINT	✓	双精度浮点数→BIN 整数变换	5
<u>0206</u>	MMOV	—	—	✓	16→32 位数值转换	5
<u>0207</u>	RMOV	—	—	✓	32→16 位数值转换	5
<u>0208</u>	GRY	DGRY	—	✓	BIN→GRY 码变换	5
<u>0209</u>	GBIN	DGBIN	—	✓	GRY 码→BIN 变换	5
<u>0210</u>	NEG	DNEG	—	✓	取负数 (取 2 的补码)	3
<u>0211</u>	—	FNEG	—	✓	单精度浮点数正负符号反相	3
<u>0212</u>	—	FBCD	—	✓	单精度浮点数→十进浮点数	5
<u>0213</u>	—	FBIN	—	✓	十进浮点数→单精度浮点数	5
<u>0214</u>	BKBCD	—	—	✓	连续区块 BIN→BCD 变换	7
<u>0215</u>	BKBIN	—	—	✓	连续区块 BCD→BIN 变换	7
<u>0216</u>	SCAL	—	—	✓	比例运算	9
<u>0217</u>	SCLP	DSCLP	—	✓	参数型比例运算	9
<u>0218</u>	LINE	DLINE	—	✓	COLUMN to LINE	7
<u>0219</u>	COLM	DCOLM	—	✓	LINE to COLUMN	7

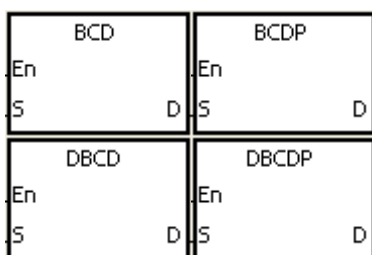
6.3.2 数据转换指令说明

API	指令码			操作数						功能					
0200	D	BCD	P	S · D						BIN→BCD 变换					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



S : 变换来源装置 Word/Double Word
D : 变换之结果 Word/Double Word

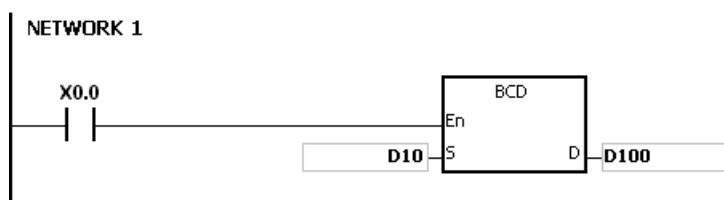
指令说明：

1. 数据来源 **S** 的内容 (BIN 值) 作 BCD 的转换 , 存于 **D** 。
2. DBCD 才可以使使用 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 。

6

程序范例：

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



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

补充说明：

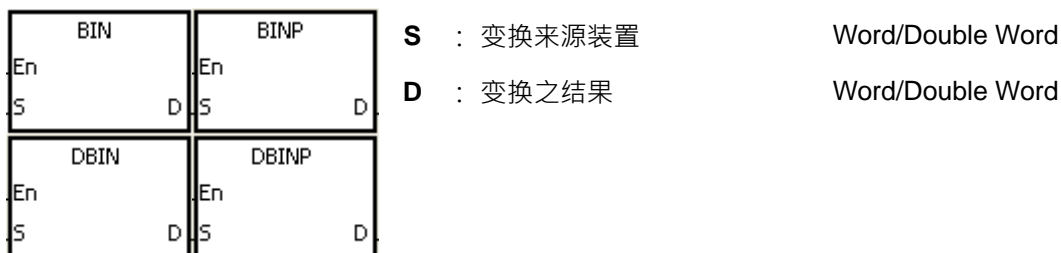
1. 在 BCD 转换结果若超过 0~9,999 , 指令不执行 , SM0=ON , 错误码 SR0=16#200D (BCD 值以 HEX 表示有任一位数不在 0~9 的范围内) 。
2. 在 DBCD 转换结果若超过 0~99,999,999 , 指令不执行 , SM0=ON , 错误码 SR0=16#200D (BCD 值以 HEX 表示有任一位数不在 0~9 的范围内) 。

API	指令码			操作数							功能						
0201	D	BIN	P	S · D							BCD→BIN 变换						

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：

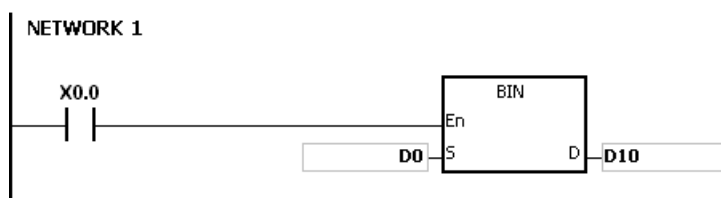


指令说明：

1. 数据来源 **S** 的内容 (BCD : 0~9,999) · DBCD (0~99,999,999) 作 BIN 的转换 · 存于 **D** 。
2. 数据来源 **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：

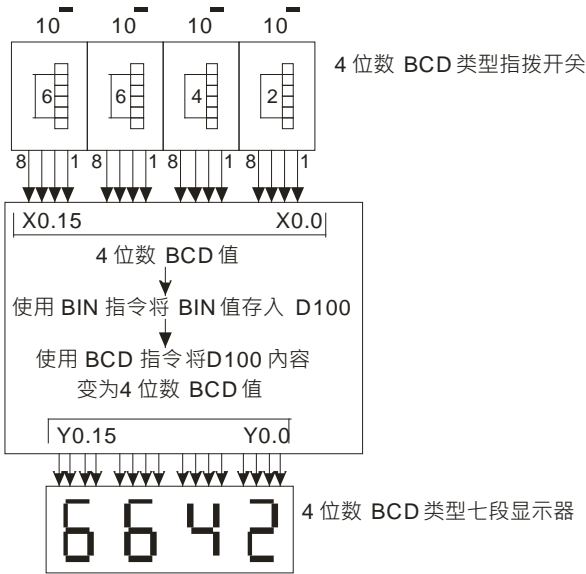
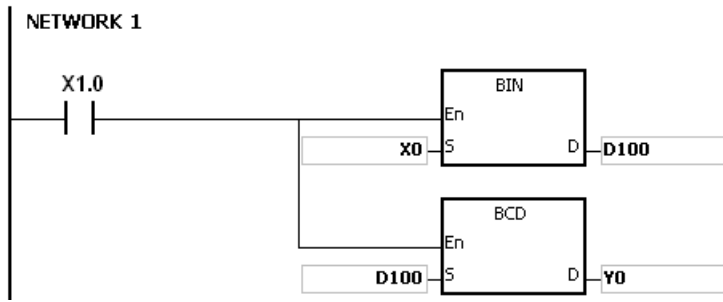
当 X0.0=ON 时，D0 之 BCD 值被转换成 BIN 值后，将结果存于 D10 中。



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. 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 之 BIN 值转换成 BCD 值传送到 Y0.0~Y0.15 。

6



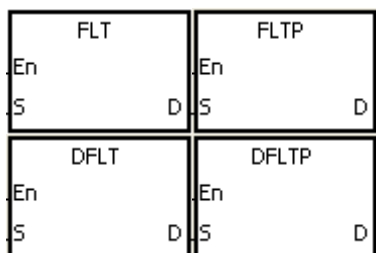
6

API	指令码			操作数									功能				
0202	D	FLT	P	S · D									BIN 整数→单精度浮点数变换				

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：

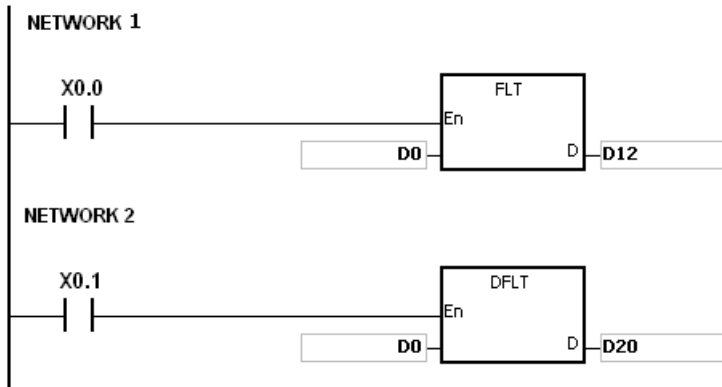
**S**：变换来源装置 Word/Double Word**D**：存放变换结果的装置 Double Word

指令说明：

1. 将 BIN 整数变换成单精度浮点数值。
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** 存放变换结果之装置占用 2 个寄存器。
4. DFLT 中 **S** 变换来源装置占用 2 个寄存器，**D** 存放变换结果之装置占用 2 个寄存器。
 - 若转换结果的绝对值大于可表示之最大浮点值，则进位标志 SM602=ON，**D** 存放最大浮点值。
 - 若转换结果的绝对值小于可表示之最小浮点值，则借位标志 SM601=ON，**D** 存放最小浮点值。
 - 若转换结果为 0，则零标志 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值存放于 (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值存放于 (D21、D20)。
3. 若 D0=10，则 X0.0=ON，转换后单精度浮点数值为 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值为 16#47C35000，存于 32 位寄存器 (D21、D20) 内。

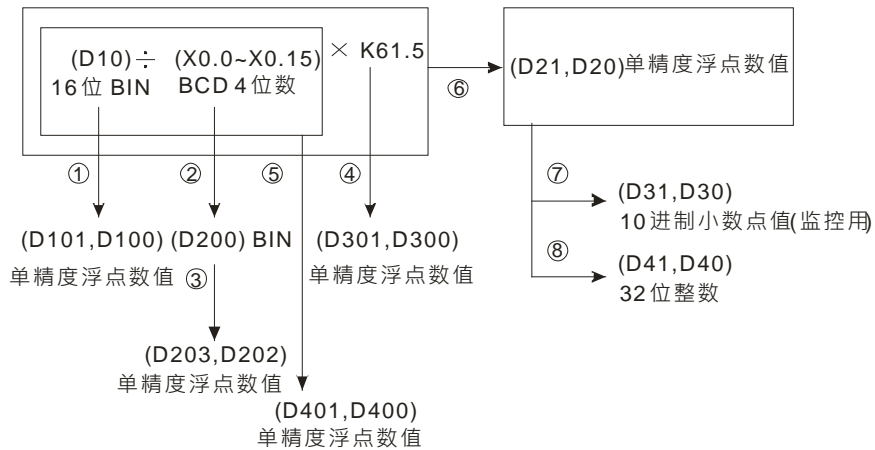


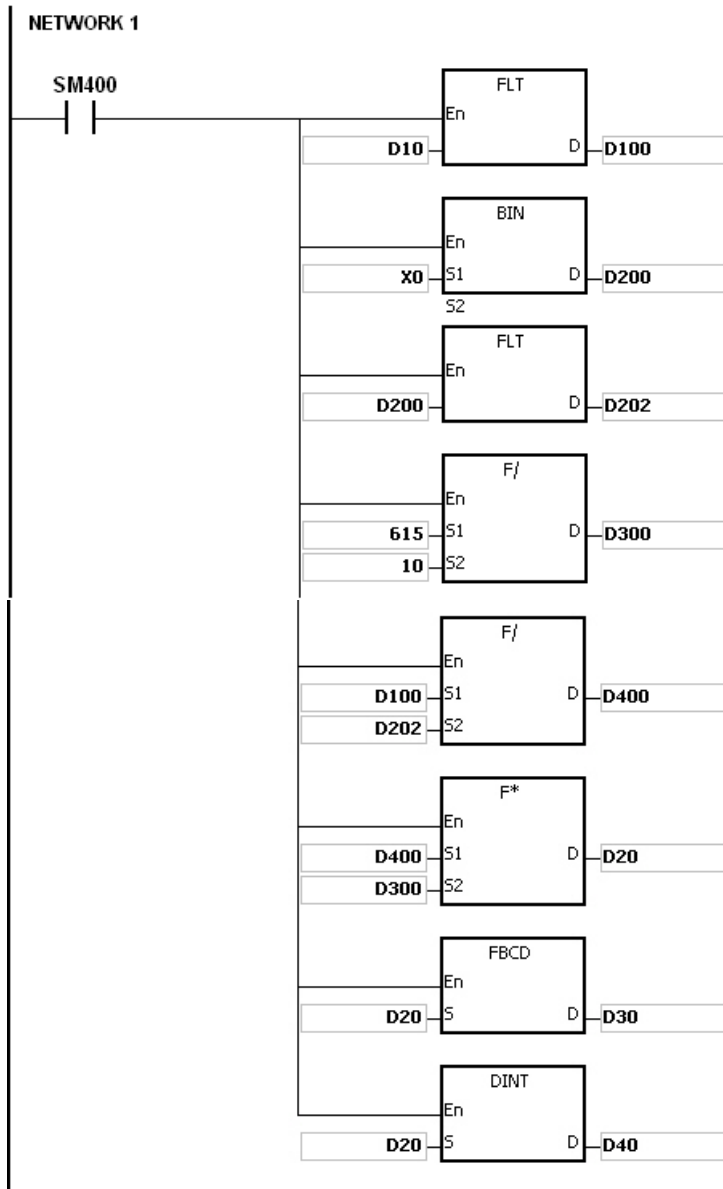
程序范例二：

使用应用指令来完成下列的算式。

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

6





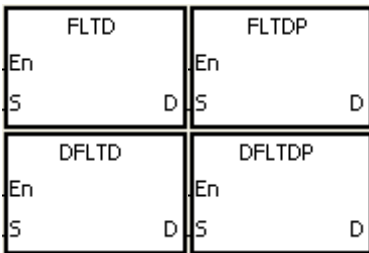
6

API	指令码			操作数								功能					
0203	D	FLTD	P	S · D								BIN 整数转换双精度浮点数					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



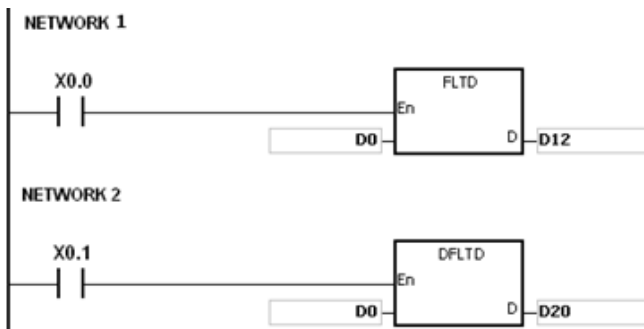
S : 变换来源装置 Word/Double Word
D : 存放变换结果之装置 Long Word

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** 存放变换结果之装置占用 4 个寄存器。
4. DFLTDP 中 **S** 变换来源装置占用 2 个寄存器，**D** 存放变换结果之装置占用 4 个寄存器。
5. 若转换结果的绝对值大于可表示之最大浮点值，则进位标志 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之最小浮点值，则借位标志 SM601=ON，**D** 存放最小浮点值。
7. 若转换结果为 0，则零标志 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值存放于 (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值存放于 (D23、D22、D21、D20)。
3. 若 D0=10，则 X0.0=ON，转换后双精度浮点数值为 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值为 16#40F86A0000000000，存于 64 位寄存器 (D23、D22、D21、D20)。

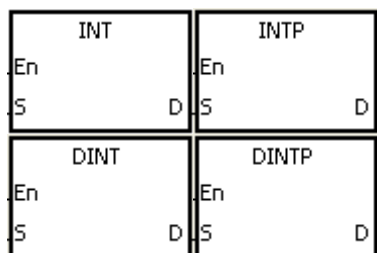


API	指令码			操作数								功能					
0204	D	INT	P	S · D								单精度浮点数→BIN 整数变换					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



S : 变换来源装置

Double Word

D : 变换之结果

Word/Double Word

指令说明：

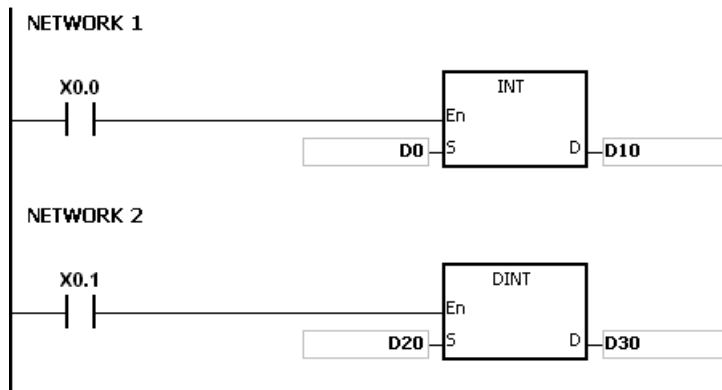
1. S 所指定的寄存器内容以单精度浮点数类型被变换成 BIN 整数暂存于 D 所指定的寄存器当中，BIN 整数浮点数被舍弃。
2. INT 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果之装置占用 1 个寄存器。
3. DINT 中 S 变换来源装置占用 2 个寄存器，D 存放变换结果之装置占用 2 个寄存器。
4. INT 中 D 不可使用 HC 装置。
5. 本指令的动作与 API0202 (FLT) 指令刚好相反。
6. 变换结果若为 0 时，零标志 SM600=ON。
7. 变换结果有浮点数被舍弃时，借位标志 SM601=ON。
8. 变换结果若超出下列范围时 (溢位)，进位标志 SM602=ON。

INT/IINTP : -32,768~32,767

DINT/DINTP : -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) 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，将单精度浮点数 (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。



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。

6

API	指令码			操作数								功能					
0205	D	FINT	P	S · D								双精度浮点数转换 BIN 整数					

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

脉冲执行型	32 位指令 (5 steps)	64 位指令 (5 steps)
AH500	AH500	AH500

符号：

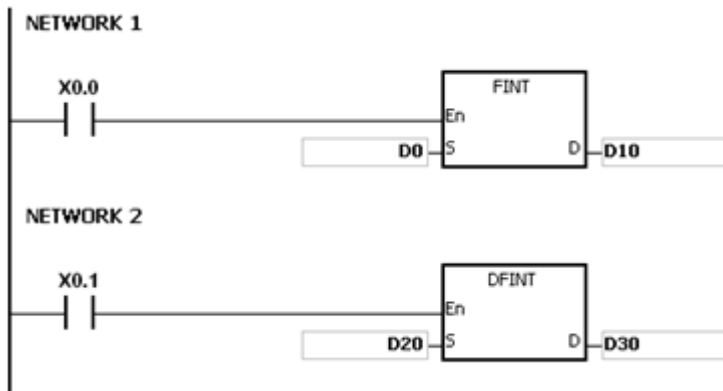
FINT En S D	FINTP En S D	S : 变换来源装置 D : 变换之结果	Long Word Word/Double Word
DFINT En S D	DFINTP En S D		

指令说明：

- S** 所指定的寄存器内容以双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。
- FINT 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 1 个寄存器。
- DFINT 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。
- FINT 跟 FLTP 中 **D** 不可使用 HC 装置。
- 本指令的动作与 API0203 (FLTD) 指令刚好相反。
- 变换结果若为 0 时，零标志 SM600=ON。
- 变换结果有浮点数被舍弃时，借位标志 SM601=ON。
- 变换结果若超出下列范围时 (溢位)，进位标志 SM602=ON。
FINT/FINTP : -32,768~32,767
DFINT/DFINTP : -2,147,483,648~2,147,483,647

程序范例：

- 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。
- 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。



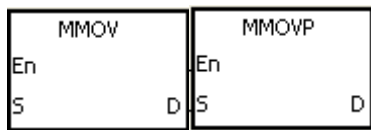
6

API	指令码		操作数								功能				
0206	MMOV	P	S · D								16→32 位数值转换				

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



S : 变换来源装置

Word

D : 变换之结果

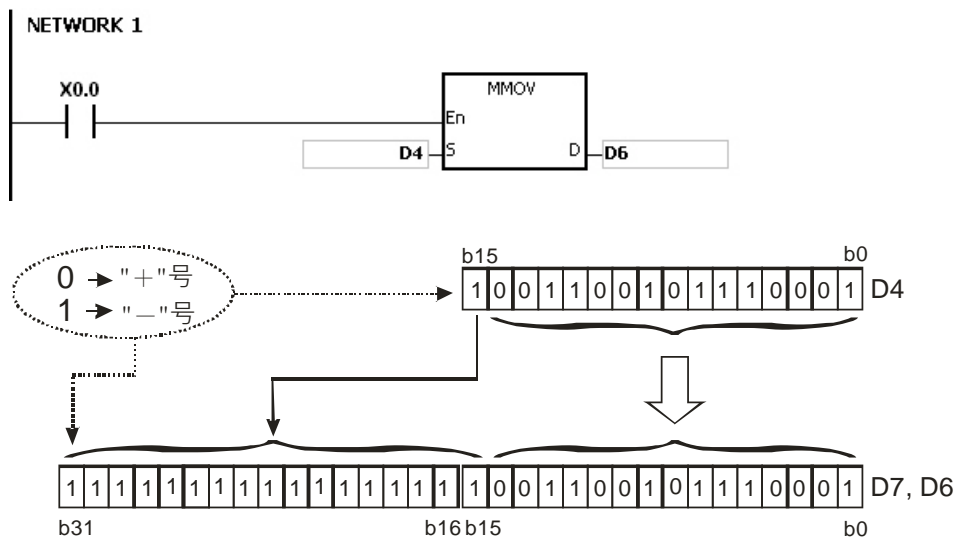
Double Word

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：

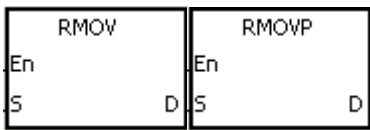
当 X0.0 为 0N 时，D4 的 b15 位数据传送到 (D7 · D6) 的 b15 到 b31 位，变成负数 (和 D4 的一样)。



6

API	指令码			操作数							功能						
0207	RMOV	P		S · D							32→16 位数值转换						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				
脉冲执行型										16 位指令 (5 steps)				32 位指令			
AH500										AH500				-			

符号：



S : 变换来源装置 Double Word
 D : 变换之结果 Word

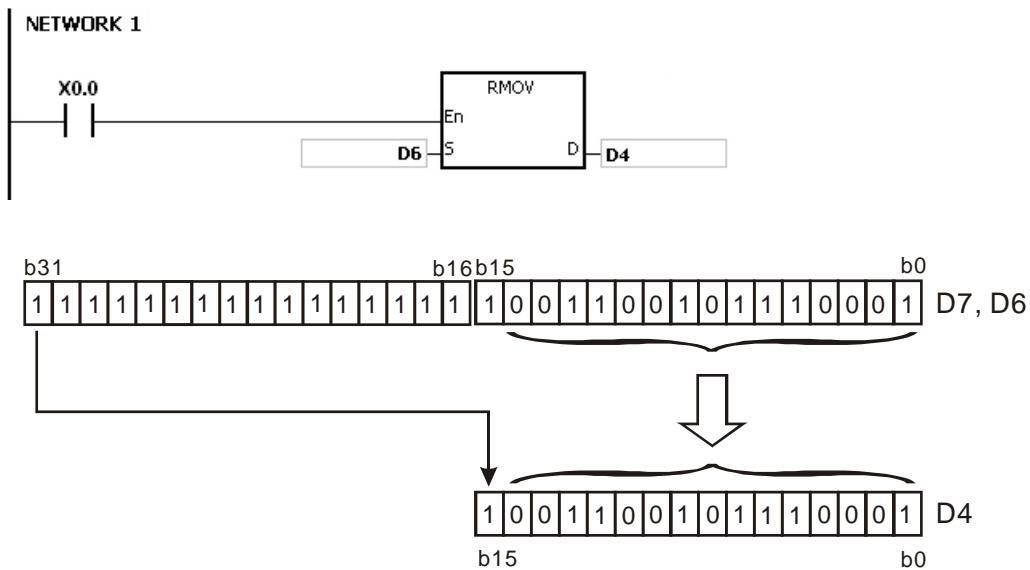
指令说明：

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 0N 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。

6

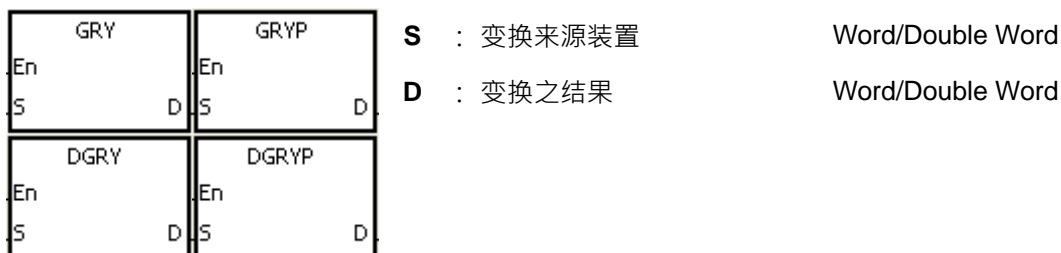


API	指令码			操作数								功能					
0208	D	GRY	P	S · D								BIN→GRAY 码变换					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：

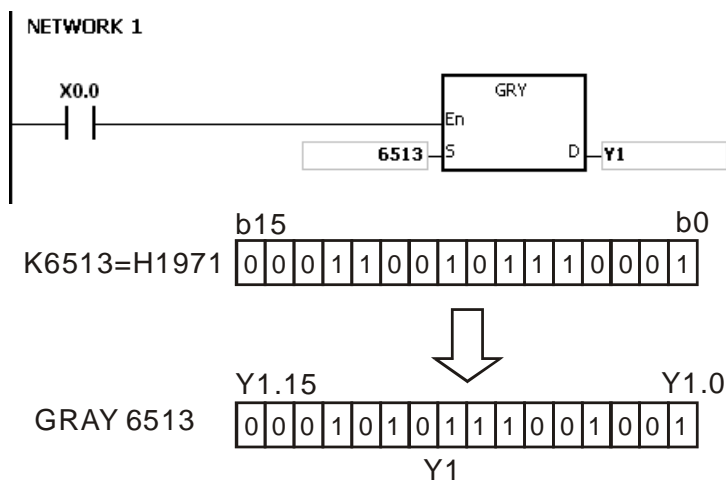


指令说明：

1. 将 S 所指定装置之内容值 (BIN 值) 变换格雷码 (GRAY CODE) 后存放到 D 所指定之装置中。
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) 后存放到 Y1.0~Y1.15。



补充说明：

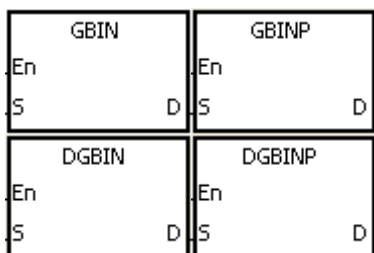
S 的内容值小于 0，视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码			操作数							功能						
0209	D	GBIN	P	S · D							GRAY 码→BIN 变换						

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



S : 变换来源装置 Word/Double Word

D : 变换之结果 Word/Double Word

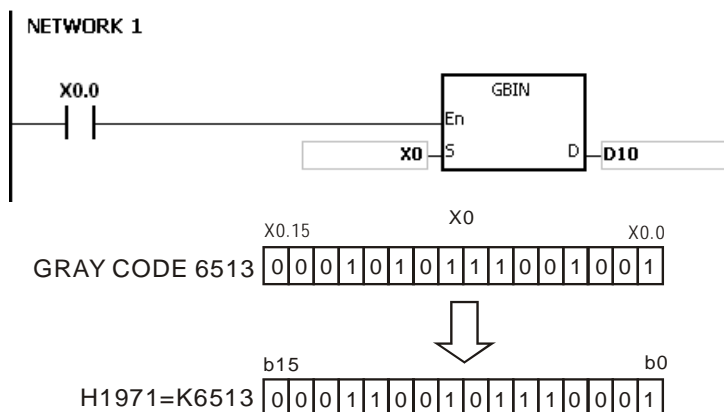
指令说明：

1. 将 **S** 所指定装置之内容值 (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

6

程序范例：

当 X0.0 时，将 X0.0~X0.15 输入点所连接之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。

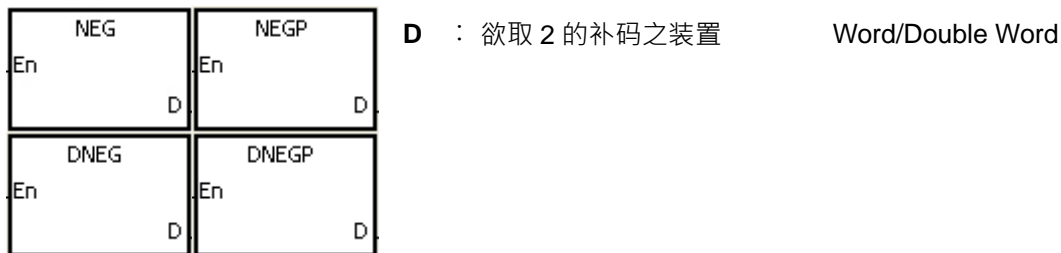


补充说明：

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。

API	指令码			操作数								功能					
0210	D	NEG	P	D								2 的补码					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型										16 位指令 (3 steps)			32 位指令 (3 steps)				
AH500										AH500			AH500				

符号：

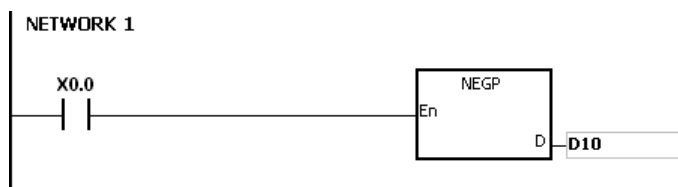


指令说明：

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：

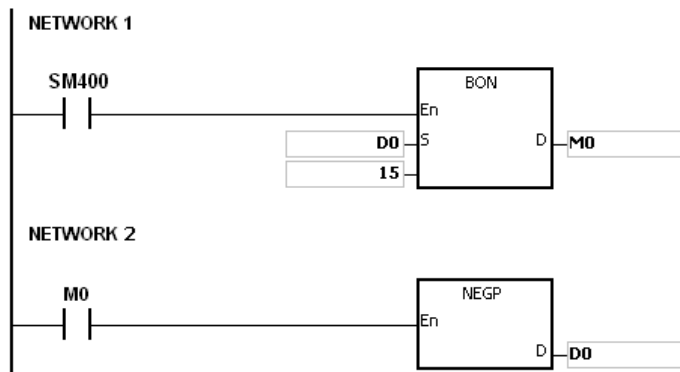
当 X0.0=OFF→ON 时，D10 内容的各位全部反相(0→1、1→0)后再加 1 存放于原寄存器 D10 当中。



程序范例二：

求负数的绝对值：

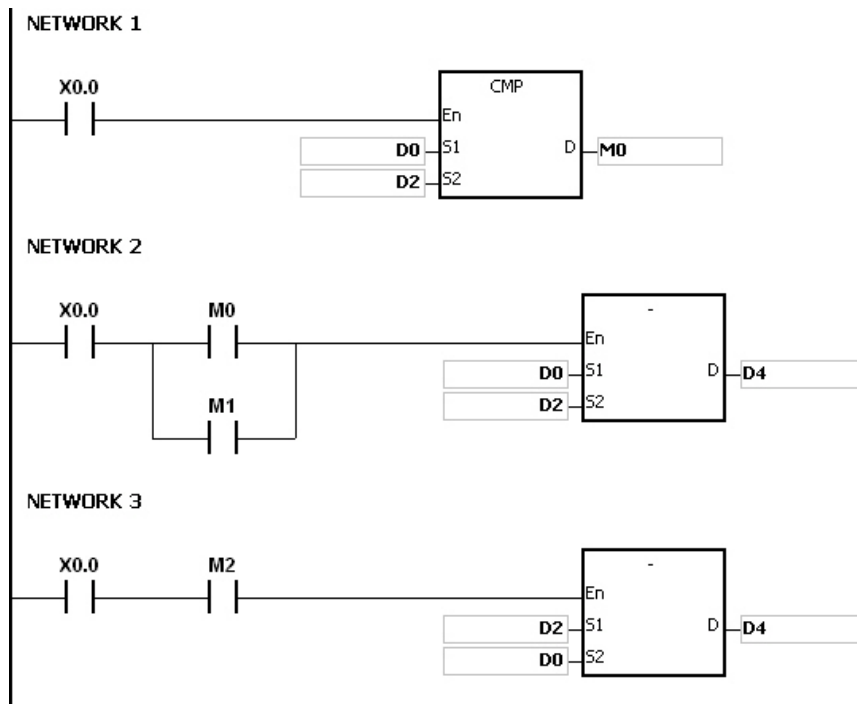
1. 当 D0 的第 15 个位为“1”时，M0=ON。(D0 表示为负数)
2. M0=ON 时，用 NEG 指令将 D0 取 2 的补码可得到其绝对值。



程序范例三：

减法运算之差取绝对值，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。



6

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

(D0)=1

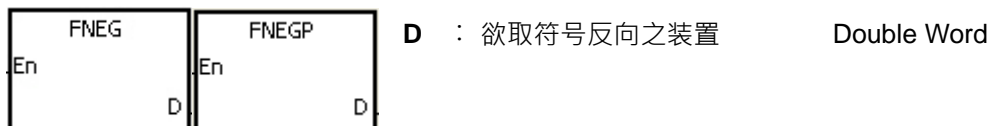
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0)=0

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

API	指令码			操作数								功能							
0211		FNEG	P	D								单精度浮点数正负符号反相							
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF		
D	●	●			●	●	●	●	●		●	○	●						
										脉冲执行型			32 位指令 (3 steps)			64 位指令			
										AH500			AH500			-			

符号：



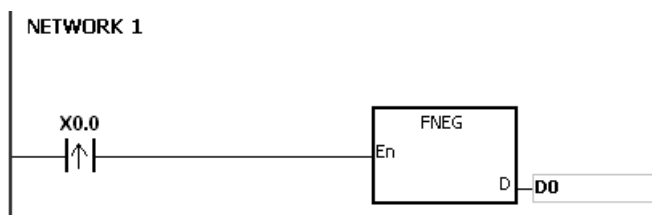
指令说明：

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)。

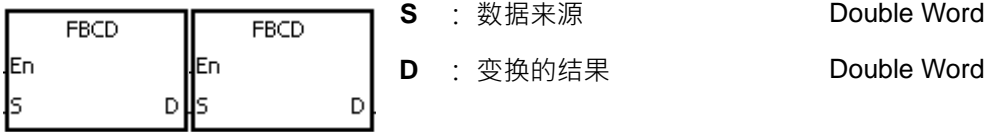
执行前 (D1、D0) = 16#2E0F9000 (正值)，当 X0.0 从 OFF→ON 时，(D1、D0) 的数据以单精度浮点数做正负符号的反相运作，因此执行后 (D1、D0) = 16#AE0F9000 (负值)。



API	指令码			操作数							功能						
0212		FBCD	P	S · D							单精度浮点数 → 十进浮点数						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	32 位指令 (5 steps)	64 位指令
AH500	AH500	-

符号：

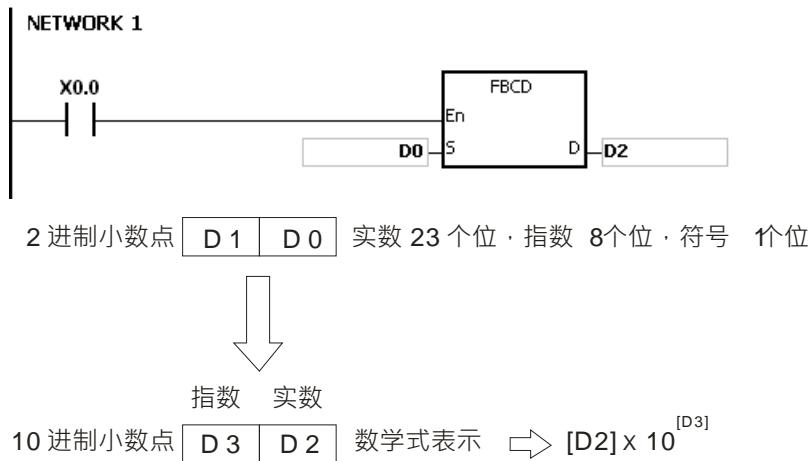


指令说明：

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)。



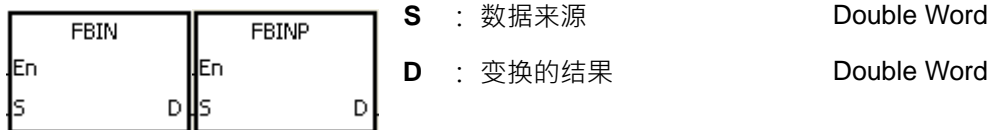
补充说明：

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。

API	指令码			操作数								功能					
0213		FBIN	P	S · D								十进浮点数→单精度浮点数					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
D	●	●			●	●	●	●	●		●	○	●				

脉冲执行型	32 位指令 (5 steps)	64 位指令
AH500	AH500	-

符号：

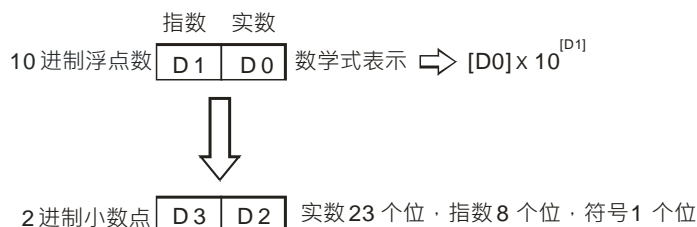
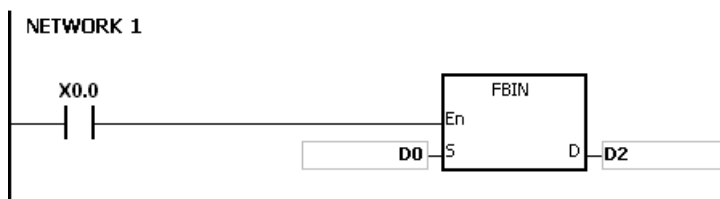


指令说明：

1. 将 S 所指定的寄存器中的十进制浮点数变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=1.234x10⁶。
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变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 ±1175x10⁻⁴¹ 到 ±3402x10⁺³⁵。若运算结果为 0，则零标志 SM600=ON。

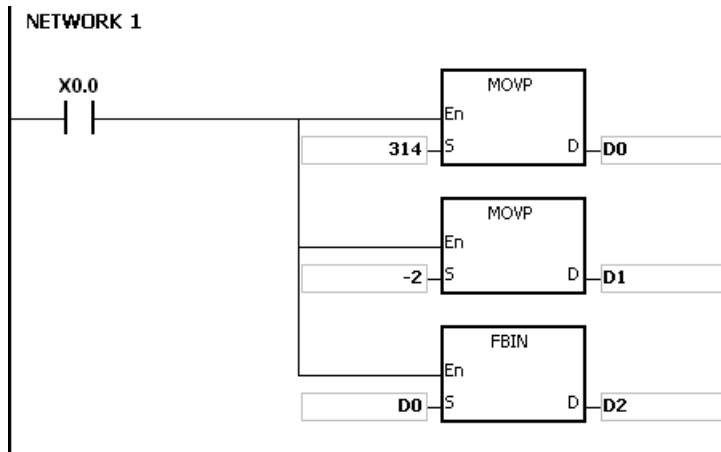
程序范例一：

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 指令可将浮点数值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：

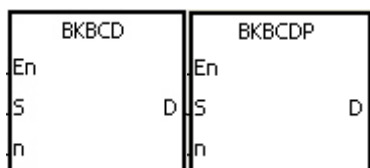
若 S 操作数的十进浮点数实数范围不在-9,999~+9,999 之间或指数范围不在-41~+35 之间，指令不执行，SM0=ON，错误码 SR0=16#2013。

API	指令码		操作数							功能						
0214	BKBCD	P	S · n · D							连续区块 BIN→BCD 变换						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



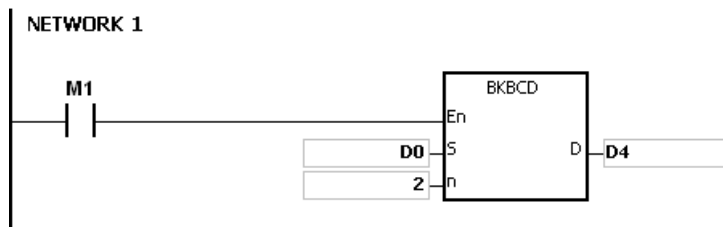
- S** : 数据来源 Word
- n** : 欲转换的数据数 Word
- D** : 变换的结果 Word

指令说明：

1. 将 n 笔的数据来源 S 的内容 (BIN 值) 作 BCD 的转换 · 存于 D。
2. 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：

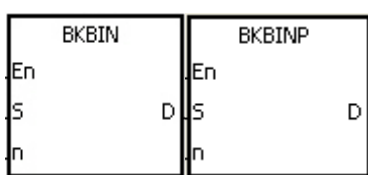
1. 若 n<1 或 n>256 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. S+n-1，D+n-1，装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 在 BCD 变换结果若超过 0~9,999，指令不执行，SM0=ON，SR0 记录错误码 16#200D。(BCD 值以 HEX 表示有任一位数不在 0~9 的范围内)。
4. 若 S~S+n-1 的装置与 D~D+n-1 的装置有重复时 指令不执行 · SM0=ON 错误码 SR0=16#200C。

API	指令码			操作数							功能				
0215		BKBIN	P	S · n · D							连续区块 BCD→BIN 变换				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



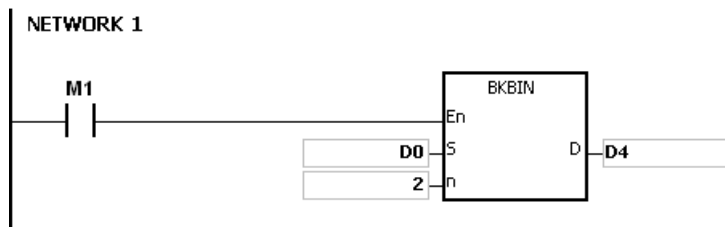
- S : 数据来源 Word
- n : 欲转换的数据数 Word
- D : 变换的结果 Word

指令说明：

1. 将 n 笔的数据来源 S 的内容 (BCD : 0~9,999) 作 BIN 的转换，存于 D。
2. 数据来源 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 表示有任一数位不在 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 的装置有重复时 指令不执行 SM0=ON 错误码 SR0=16#200C。

6

API	指令码			操作数							功能						
0216		SCAL	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							比例值运算						

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令
AH500	AH500	-

符号：

SCAL		SCALP		S_1	Word
En		En		S_2	Word
S1	D	S1	D	S_3	Word
S2		S2		D	Word
S3		S3			

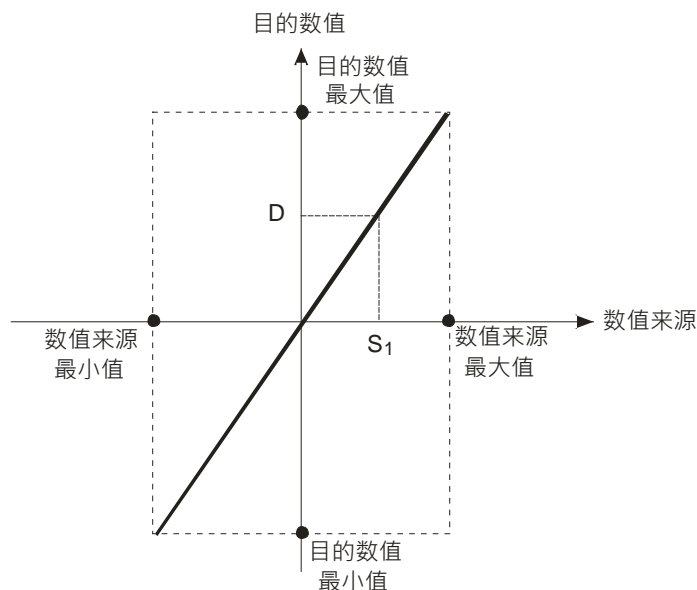
指令说明：

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

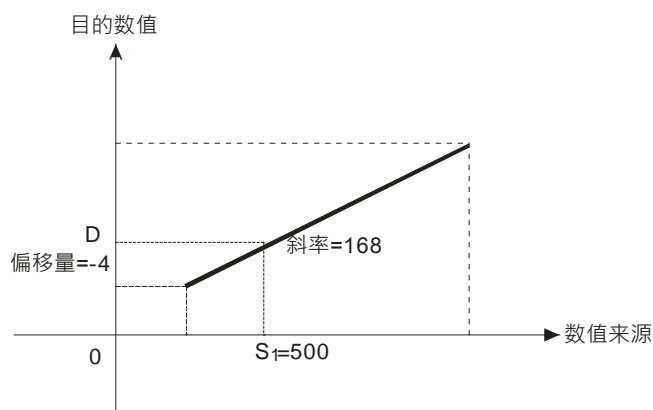
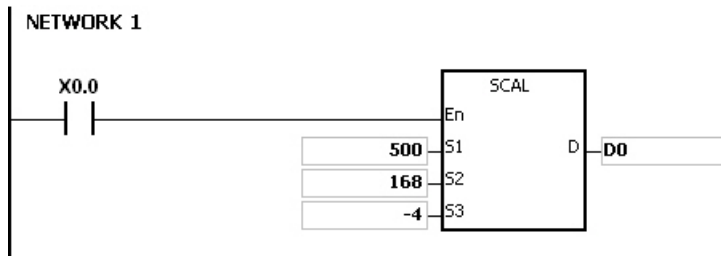
偏移量公式为 $S_3 = \text{目的数值最小值} - \text{数值来源最小值} \times S_2 \div 1,000$

其输出曲线将如下图所示：



程序范例一：

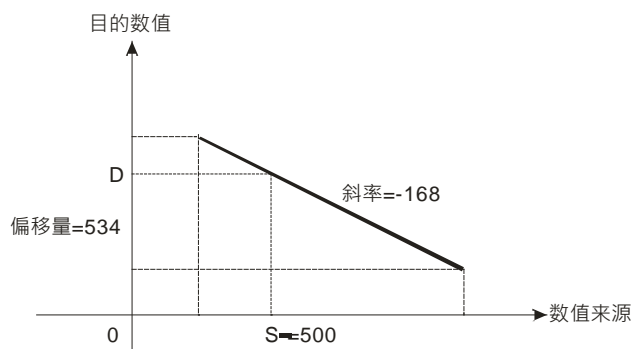
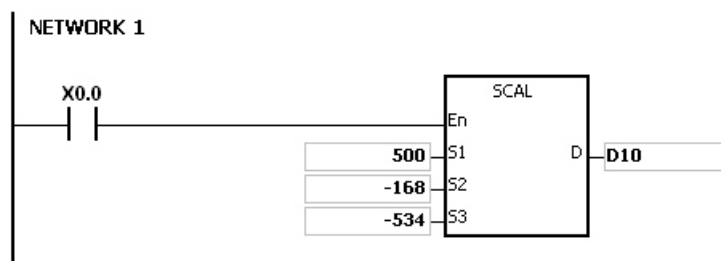
1. 已知 S_1 数值数据来源为 500 · S_2 斜率为 168 · S_3 偏移量为 -4 · 当 X0.0=ON 时 · SCAL 指令执行 · 可在 D0 得到所要求的比例值。
2. 运算方式： $D0 = (500 \times 168) \div 1,000 + (-4) = 80$



6

程序范例二：

1. 已知 S_1 数值数据来源为 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：

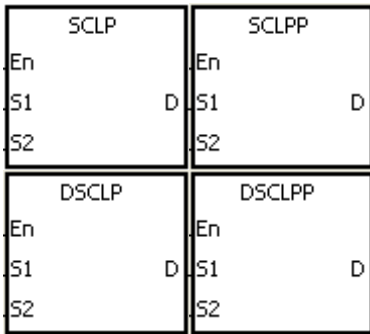
1. 此 SCAL 指令适用于已知斜率与偏移量，若不知斜率与偏移量建议使用 SCLP 指令来做运算。
2. 输入参数 S_2 时，其输入数值必须为 $-32,768 \sim 32,767$ 之间的数值（实际数值为 $-32,768 \sim 32,767$ ），若是 S_2 实际数值超过范围时，请改用 SCLP 指令运算。
3. 用户运用斜率换算公式时，须注意数值来源最大值，必须大于数值来源最小值，而目的数值最大值，并不限制大于目的数值最小值。
4. 若 D 的值 $> 32,767$ 则 $D = 32,767$ ，若 D 的值 $< -32,768$ 则 $D = -32,768$ 。

API	指令码			操作数						功能					
0217	D	SCLP	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$						参数型比例值运算					

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

脉冲执行型	16 位指令 (9 steps)	32 位指令 (9 steps)
AH500	AH500	AH500

符号：



- S_1 : 数值数据来源 Word/Double word
- S_2 : 参数 Word/Double word
- D : 目的地装置 Word/Double word

指令说明：

1. 32 位指令才可以使用 HC 装置。
2. 16 位指令 S_2 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

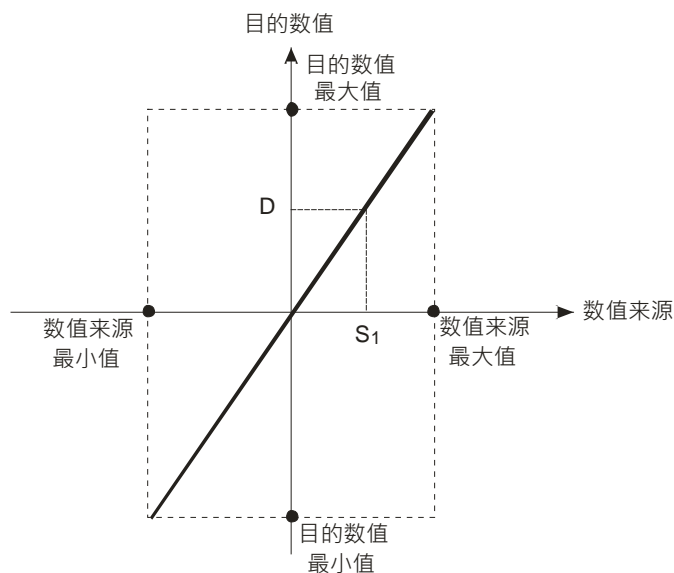
3. 16 位指令 S_2 操作数将连续占用 4 个装置。
4. 32 位指令 S_2 参数设定内容如下。

装置编号	参数名称与说明	设定范围	
		整数	浮点数
S_2 、 S_2+1	数值来源最大值	-2,147,483,648~ 2,147,483,647	32bit 浮点数范围
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。

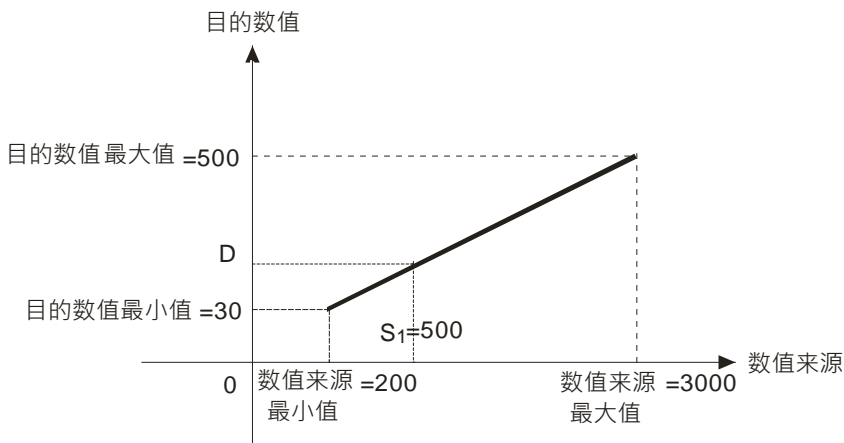
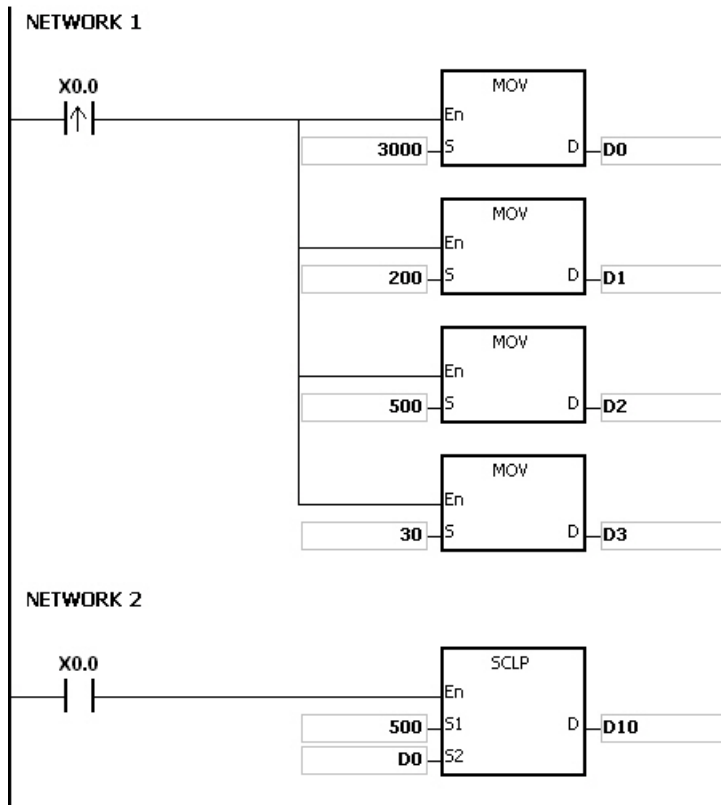
6

7. 指令内部运算公式为 $D=[(S_1-\text{数值来源最小值}) \times (\text{目的数值最大值}-\text{目的数值最小值})] \div (\text{数值来源最大值}-\text{数值来源最小值}) + \text{目的数值最小值}$ 。
8. 来源数值和目的数值运算关系：
 $y=kx+b$
 $y=\text{目的数值}(D)$
 $k=\text{斜率}=(\text{目的数值最大值}-\text{最小值}) \div (\text{来源数值最大值}-\text{最小值})$
 $x=\text{来源数值}(S_1)$
 $b=\text{偏移量}=\text{目的数值最小值}-\text{来源数值最小值} \times \text{斜率}$
 将上面的各参数带入公式 $y=kx+b$ ，即可推导得出指令内部运算公式：
 $y=kx+b=D=kS_1+b=\text{斜率} \times S_1 + \text{偏移量} = \text{斜率} \times S_1 + \text{目的数值最小值}-\text{来源数值最小值} \times \text{斜率} = \text{斜率} \times (S_1-\text{来源数值最小值}) + \text{目的数值最小值} = (S_1-\text{来源数值最小值}) \times (\text{目的数值最大值}-\text{目的数值最小值}) \div (\text{来源数值最大值}-\text{来源数值最小值}) + \text{目的数值最小值}$ 。
9. 假如 $S_1 > \text{数值来源最大值}$ ，则 $S_1 = \text{数值来源最大值}$ ；假如 $S_1 < \text{数值来源最小值}$ ，则 $S_1 = \text{数值来源最小值}$ ；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。



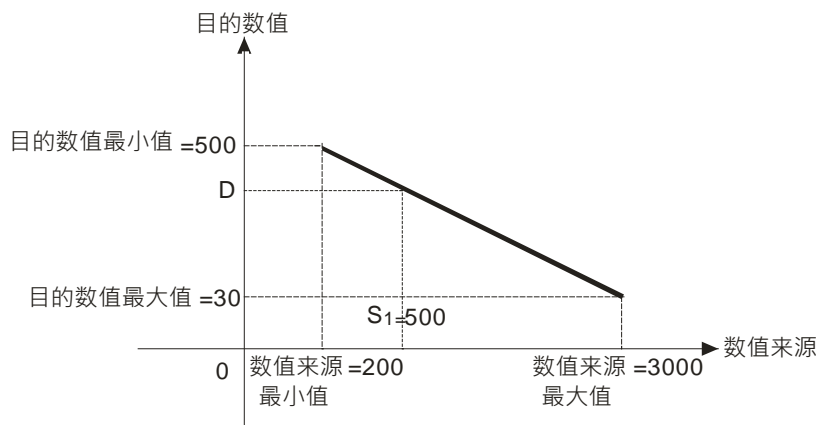
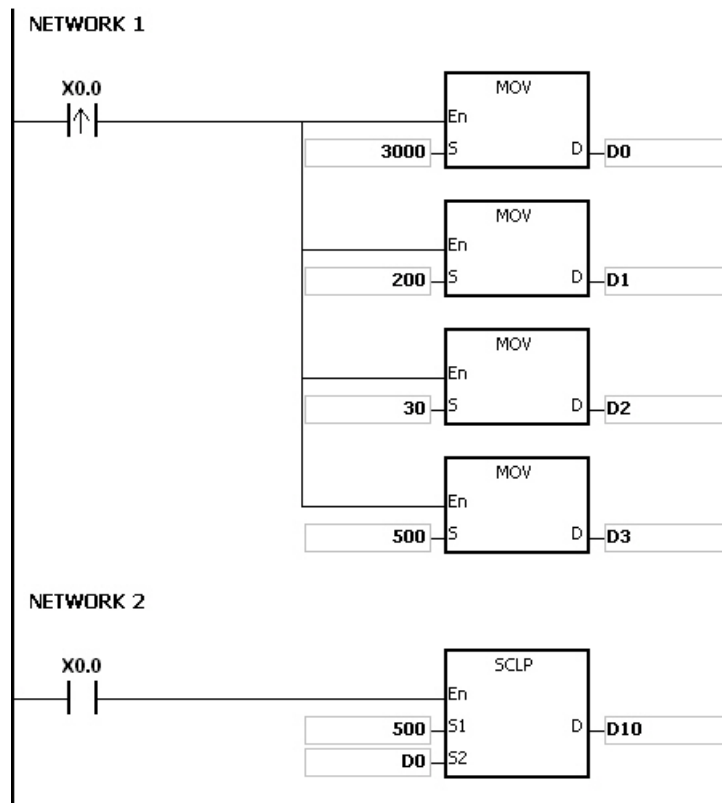
程序范例一：

1. 已知 S_1 数值数据来源为 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， $D10=80$ 。



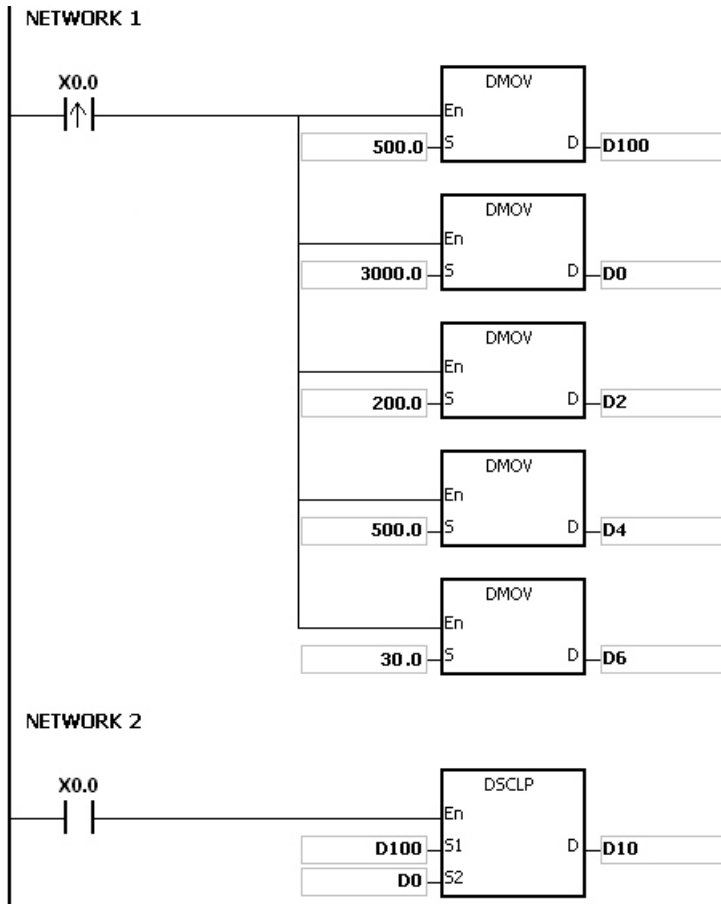
程序范例二：

1. 已知 S_1 数值数据来源为 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， $D10 = 450$ 。

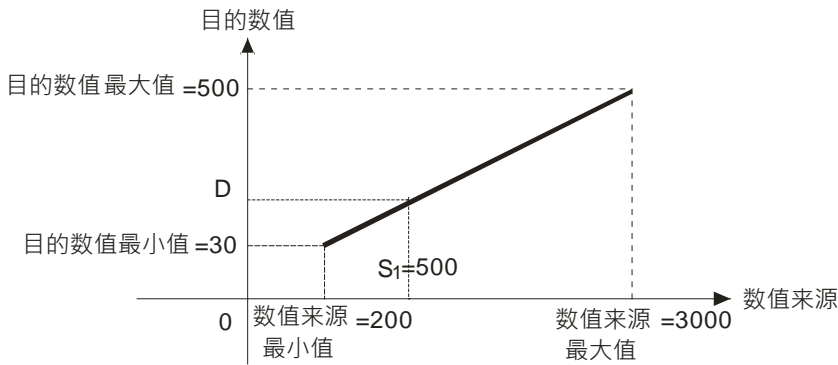


程序范例三：

1. 已知 S_1 数值数据来源 $D100$ 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， $D10 = 80.0$ 。



6

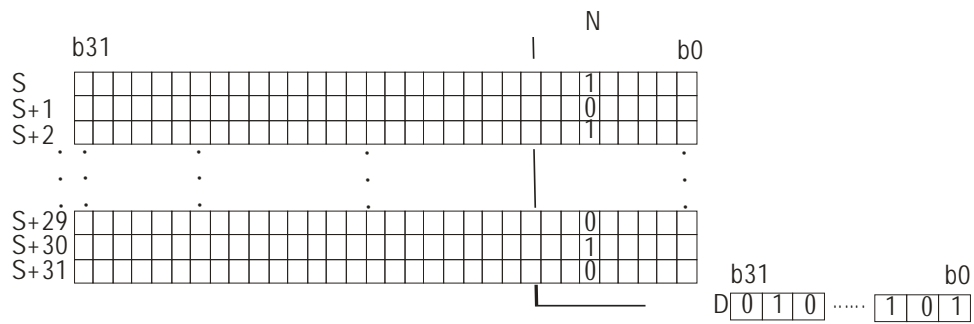


补充说明：

1. 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。
2. 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。

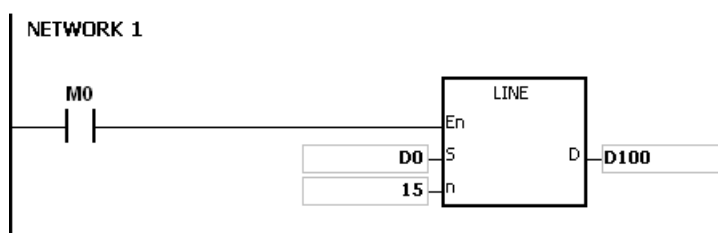
-
5. 数值来源最大值等于数值来源最小值，视为运算错误，指令不执行，SM0=ON，错误码SR0=16#2012。
 6. 16 位指令S₂操作数，若使用ISPSoft宣告，则数据类型为ARRAY [4] of WORD。
 7. 32 位指令S₂操作数，若使用ISPSoft宣告，则数据类型为ARRAY [4] of DWORD。

8. 以 32 位指令说明：



程序范例：

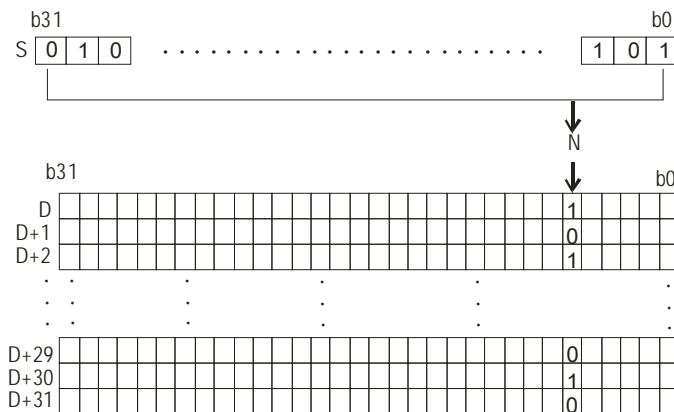
M0=ON 时，D0~D14 的第 15 个 Bit 储存于 D100 的 b0~b15 当中。



补充说明：

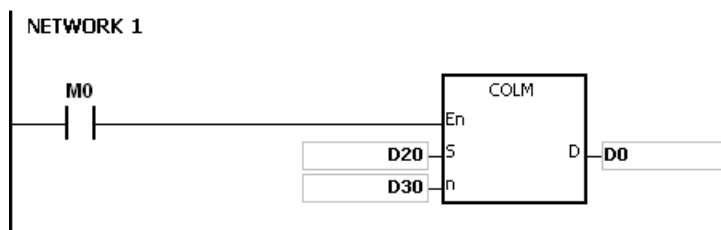
1. 16 位指令 **S+15** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 32 位指令 **S+31** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. **n** 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

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。

6.4 数据转移指令

6.4.1 数据转移指令一览表

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
0300	MOV	DMOV	—	✓	数据移动	5
0301	—	—	DFMOV	✓	双精度浮点数数据移动	5-6
0302	\$MOV	—	—	✓	字符串移动	5-11
0303	CML	DCML	—	✓	反转传送	5
0304	BMOV	—	—	✓	全部传送	7
0305	NMOV	DNMOV	—	✓	多点移动	7
0306	XCH	DXCH	—	✓	数据的交换	5
0307	BXCH	—	—	✓	全部交换	7
0308	SWAP	DSWAP	—	✓	上/下 BYTE 变换	3
0309	SMOV	—	—	✓	位数移动	11
0310	MOVB	—	—	✓	多位移动	7

6

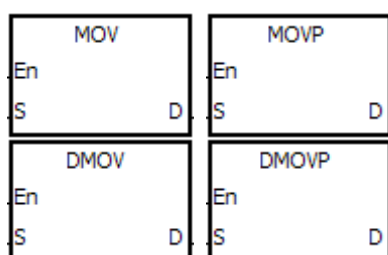
6.4.2 数据转移指令说明

API	指令码			操作数								功能					
0300	D	MOV	P	S · D								数据移动					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



S : 数据来源 Word/Double Word

D : 数据目的地 Word/Double Word

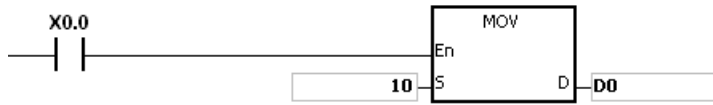
指令说明：

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。

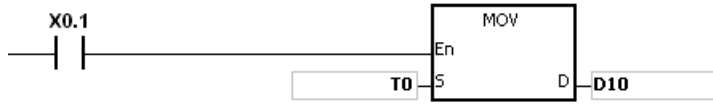
程序范例：

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。

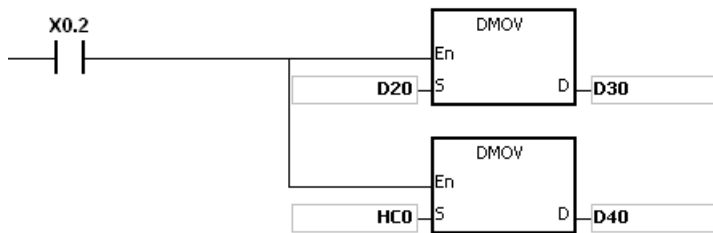
NETWORK 1



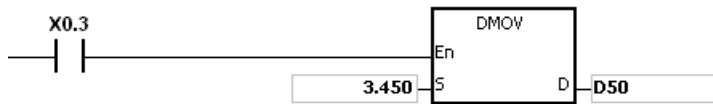
NETWORK 2



NETWORK 3



NETWORK 4



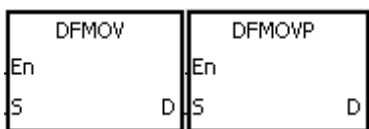
6

API	指令码			操作数							功能						
0301	D	FMOV	P	S · D							双精度浮点数数据移动						

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

脉冲执行型	32 位指令	64 位指令 (5-6 steps)
AH500	-	AH500

符号：



S : 数据来源

Long Word

D : 数据目的地

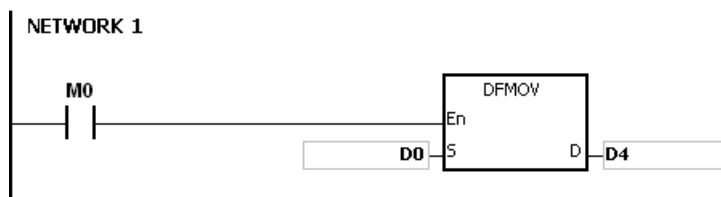
Double Word/Long Word

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、DFMOV P 指令为双精度搬移指令。

程序范例：

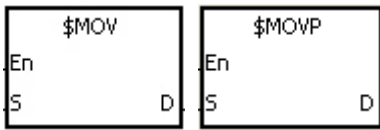
当 M0=ON 时，将 D0~D3 的数值传送到 D4~D7 当中。



API	指令码			操作数							功能						
0302		\$MOV	P	S · D							字符串移动						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●			○	
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5-11 steps)	32 位指令
AH500	AH500	-

符号：



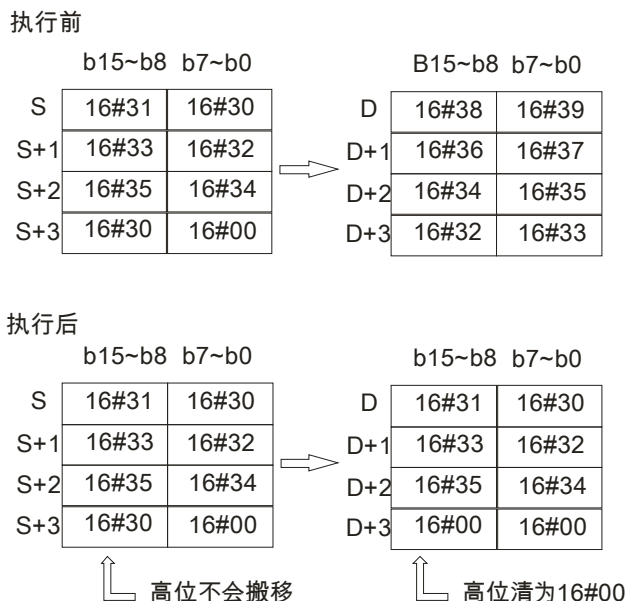
S : 数据来源 String
D : 数据搬移目的地 String

指令说明：

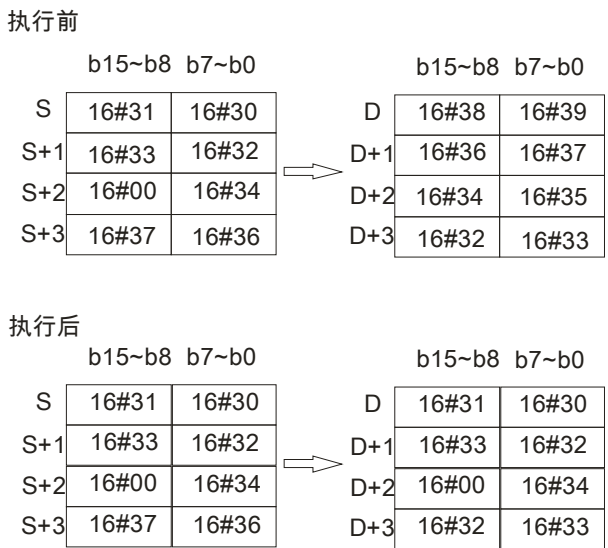
- 若 **S** 操作数为字符串时，可以搬移 1~31 个字符；且字符串的 STEP 数计算方式为 STEP 数=1+(字符串数+1)/4 (若不能整除则无条件进位)。STEP 数计算表格如下：

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

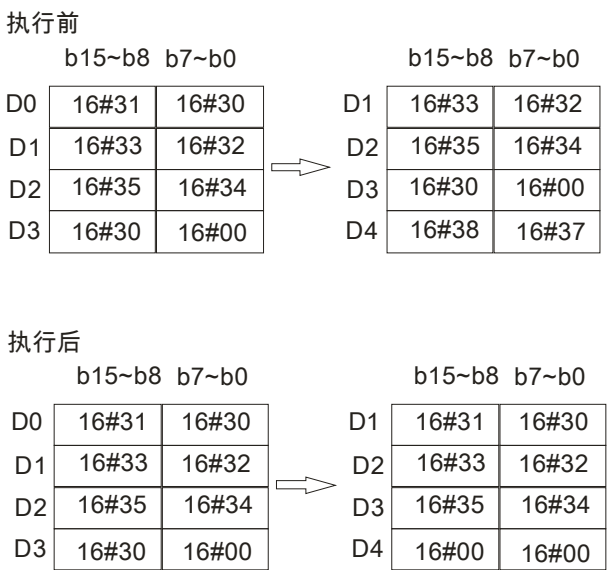
- 若 **S** 操作数为字符串且该指令执行时，将 **S** 操作数的字符串直接搬移至 **D** 操作数内且结尾会上结后缀 (16#00)。
- 若 **S** 操作数要做搬移的不是字符串时，就要在结尾后面有一个结后缀 (16#00)。
- 若 **S** 操作数不是字符串且该指令执行时，将 **S**~16#00 之间的字符串直接搬移至 **D** 内 (含结后缀 (16#00))。当指令不执行时，**D** 内容不会变化。
- 若 **S** 操作数不是字符串且该指令执行时，第一个字符就是结后缀 (16#00)，则还是会搬移结后缀。
- 当 16#00 字符串结尾在低位时，执行结果如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



7. 当 16#00 字符串结尾在高位时，执行结果如下：

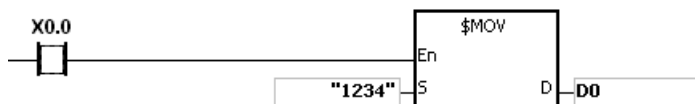


8. 当 S 跟 D 装置重叠且 S<D 时，搬移的顺序从结尾 16#00 开始搬移，一直搬到 D，不会发生 S 被覆盖的情形。



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~DD3 的值如下：



S 操作数

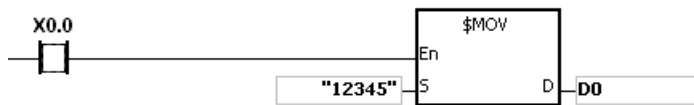
字符串	'1'	'2'	'3'	'4'
HEX	16#31	16#32	16#33	16#34

执行后 · D 操作数

装置	上 8 位	下 8 位	批注
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 为结束字符 · 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	批注
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 为结束字符
D3	不变	不变	

6

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	批注
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 为结束字符

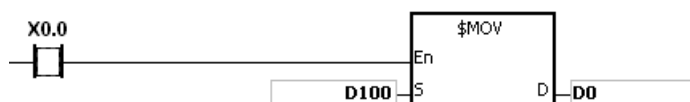
执行后 · D 操作数

装置	上 8 位	下 8 位	批注
D0	16#31	16#30	'1'=16#31 · '0'=16#30
D1	16#33	16#32	'3'=16#33 · '2'=16#32

装置	上 8 位	下 8 位	批注
D2	16#35	16#34	'5'=16#35 · '4'=16#34
D3	16#00	16#00	下 8 位 16#00 为结束字符 · 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

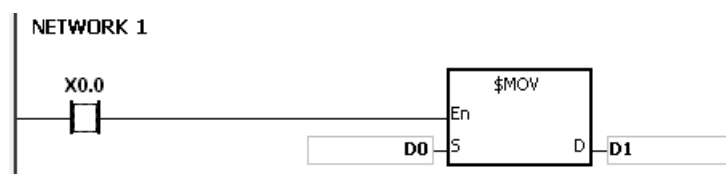
装置	上 8 位	下 8 位	批注
D100	16#31	16#30	'1'=16#31 · '0'=16#30
D101	16#33	16#32	'3'=16#33 · '2'=16#32
D102	16#00	16#34	16#00 为结束字符 · '4'=16#34
D103	16#37	16#36	'7'=16#37 · '6'=16#36

执行后 · **D** 操作数

装置	上 8 位	下 8 位	批注
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 为结束字符 · '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	批注
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 为结束字符
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	批注
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 为结束字符， 上 8 位 16#00 为自动补零
D5	不变	不变	

补充说明：

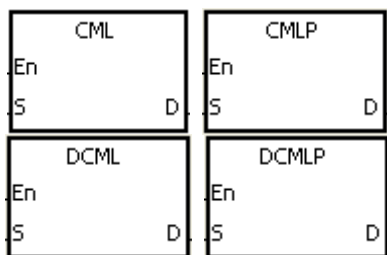
1. 当 **S** 中没有 16#00 当结尾时，指令不执行。SM0=ON，错误码 SR0=16#200E。
2. 当 **D** 装置空间不足以存放 **S** 的字符串时，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码			操作数								功能				
0303	D	CML	P	S · D								反转传送				

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



S : 数据来源 Word/Double Word

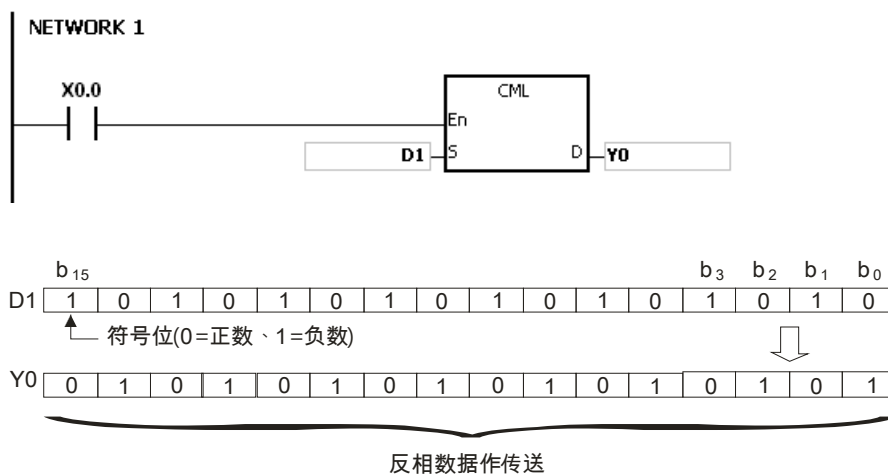
D : 数据搬移目的地 Word/Double Word

指令说明：

- 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。
- 32 位指令才可使用 HC 装置。

程序范例一：

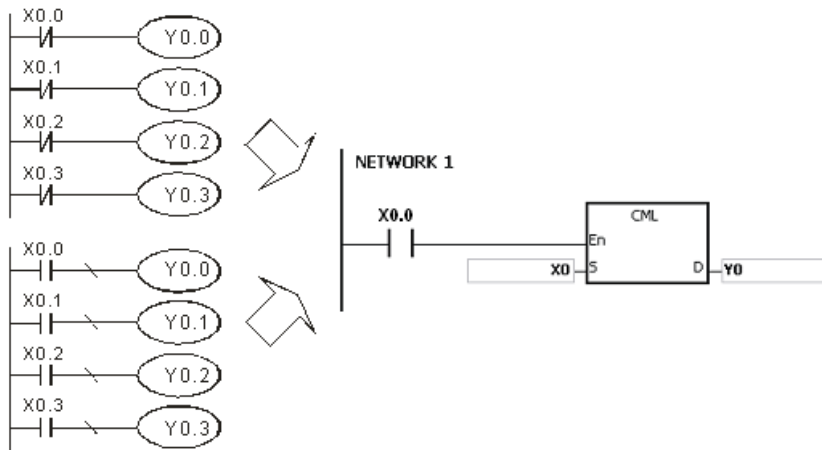
当 X0.0=ON 时，将 D1 之 b0~b15 内容反相后传送到 Y0.0~Y0.15。



6

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

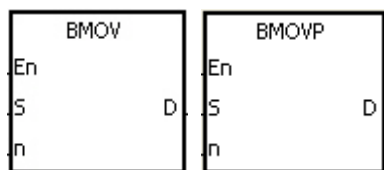


API	指令码		操作数										功能			
0304	BMOV	P	S · D · n										全部传送			

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：

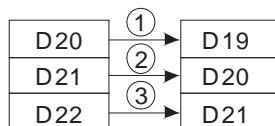


S : 数据来源 Word
D : 数据搬移目的地 Word
n : 传送长度 Word

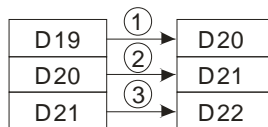
指令说明：

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，

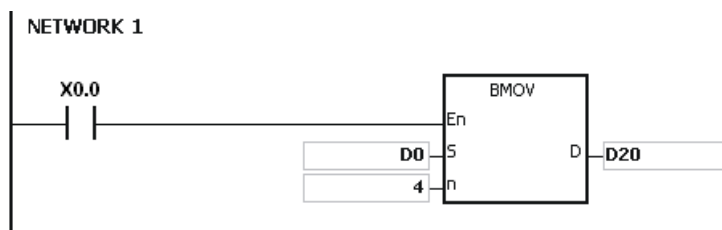


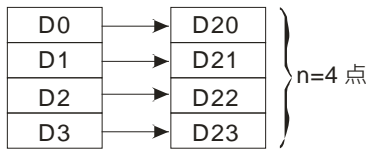
当 **S<D** 时，是以③→②→①的顺序传送。



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。

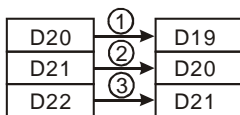
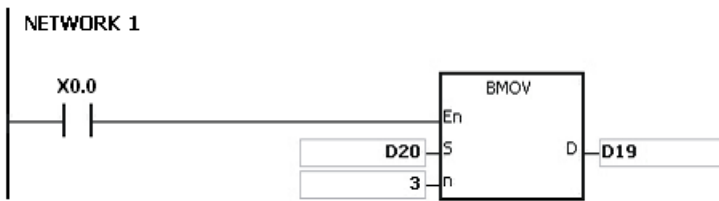




程序范例二：

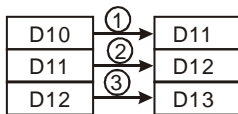
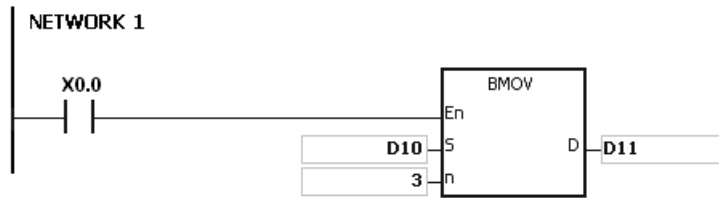
为了防止两个操作数所指定传送的号码重叠时，所造成的混乱，请注意两个操作数所指定号码大小的安排，如下所示：

1. 当 $S > D$ 时，以①→②→③的顺序传送



2. 当 $S < D$ 时，其执行结果以③→②→①的顺序传送。

6



补充说明：

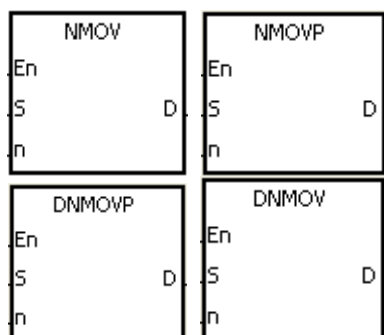
1. 当 $D+n-1$ 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 当 $S+n-1$ 超过装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 当 $n > 256$ 或 $n < 1$ 时，指令不会执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能				
0305	D	NMOV	P	S · D · n								多点移动				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

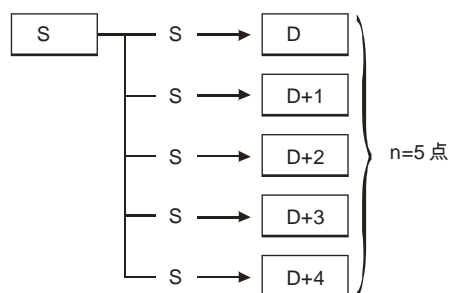
符号：



S : 数据来源 Word/Double Word
D : 数据搬移目的地 Word/Double Word
n : 传送长度 Word/Double Word

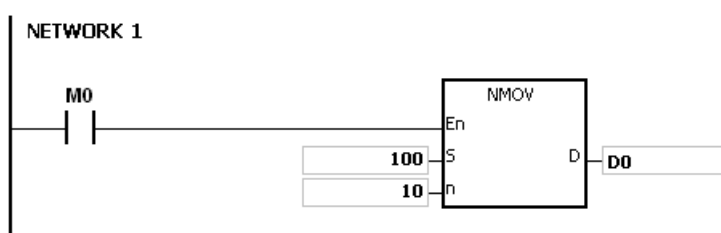
指令说明：

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 $n=1\sim 256$ ，DNMOV 时 $n=1\sim 128$ 。



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。

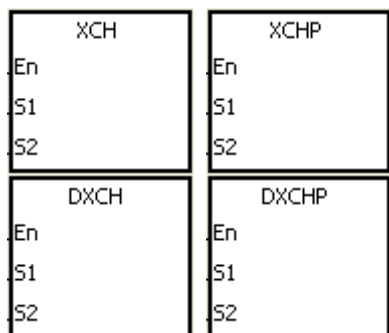
6

API	指令码			操作数									功能				
	D	XCH	P	S ₁ · S ₂									数据的交换				
0306																	

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



S₁ : 要交换的数据 1 Word/Double Word

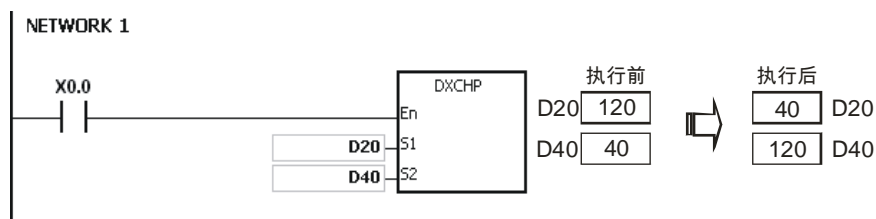
S₂ : 要交换的数据 2 Word/Double Word

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 HC 装置。

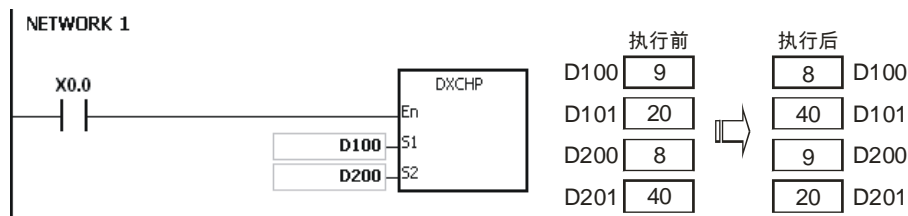
程序范例一：

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。

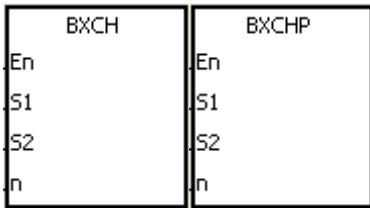


API	指令码		操作数				功能						
0307		BXCH	P	$S_1 \cdot S_2 \cdot n$				全部交换					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

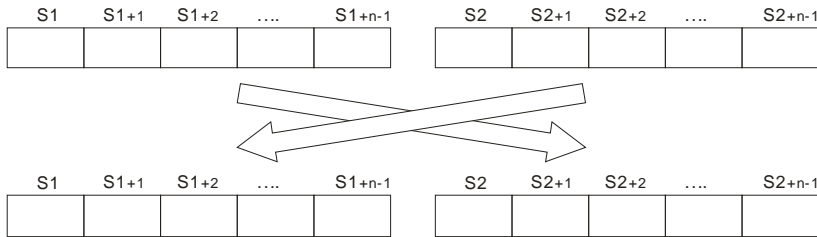
符号：



- S_1 : 要交换的数据 1 Word/Double Word
- S_2 : 要交换的数据 2 Word/Double Word
- n : 长度 Word/Double Word

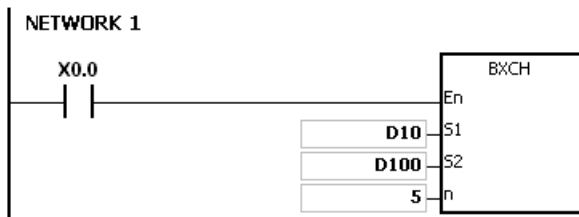
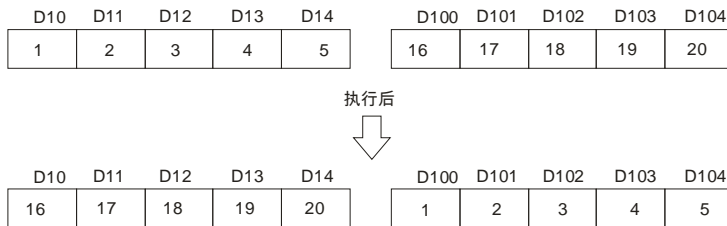
指令说明：

- $S_1 \sim S_1+n-1$ 及 $S_2 \sim S_2+n-1$ 所指定之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。

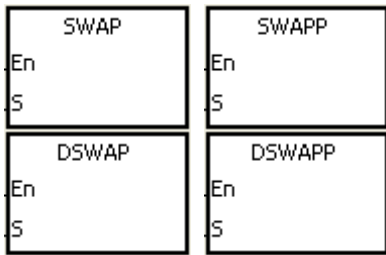


补充说明：

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。

API	指令码			操作数							功能						
0308	D	SWAP	P	S							上/下 BYTE 变换						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
脉冲执行型										16 位指令 (3 steps)			32 位指令 (3 steps)				
AH500										AH500			AH500				

符号：



S : 数据来源

Word/Double Word

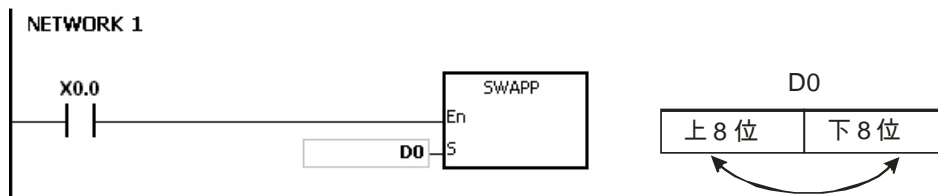
指令说明：

- 16 位指令，将 S 的上下 8 位的内容互相交换。
- 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。
- 32 位指令才可以使用 HC 装置。

程序范例一：

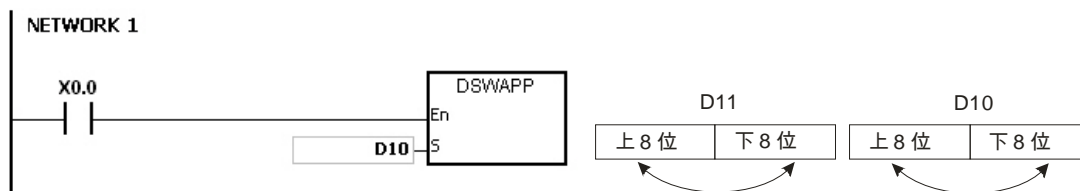
当 X0.0=ON 时，将 D0 之上位 8 位与下位 8 位的内容互相交换。

6



程序范例二：

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

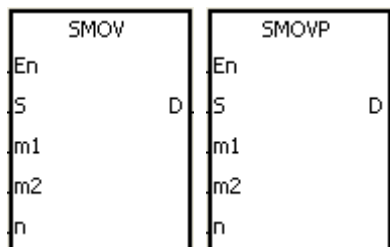


API	指令码		操作数								功能						
0309		SMOV	P	S · m ₁ · m ₂ · D · n								位数移动					

装置	X	Y	M	S	T	C	HC	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 (11 steps)	32 位指令
AH500	AH500	-

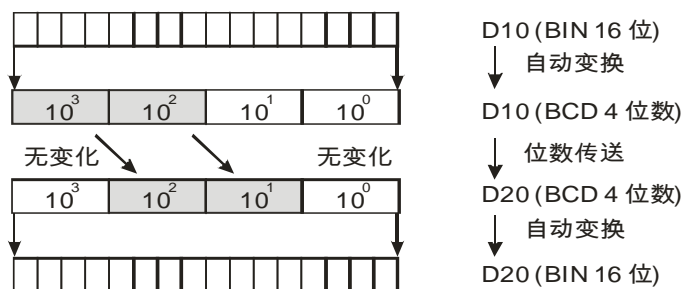
符号：



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

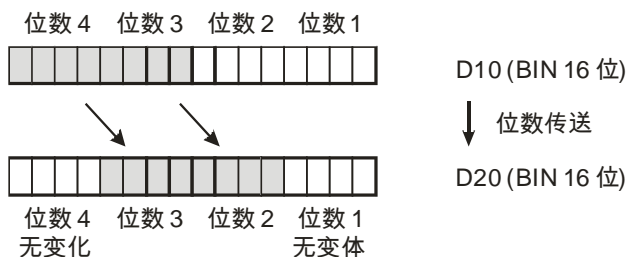
指令说明：

- 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

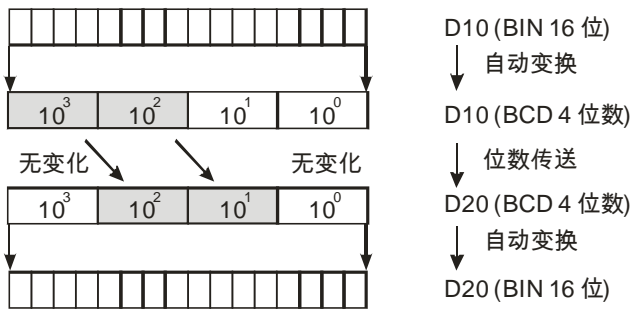
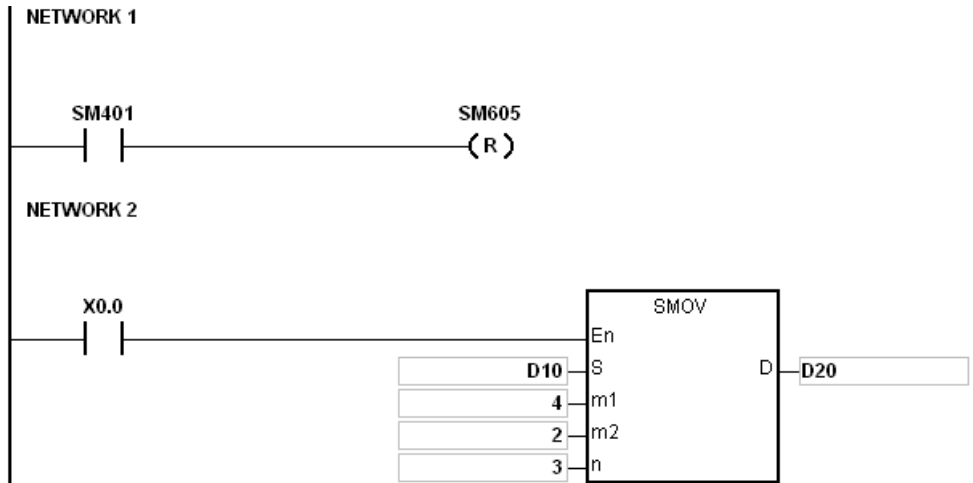
- SM605=ON 时，为 BIN 模式。



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位数内容传送到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。

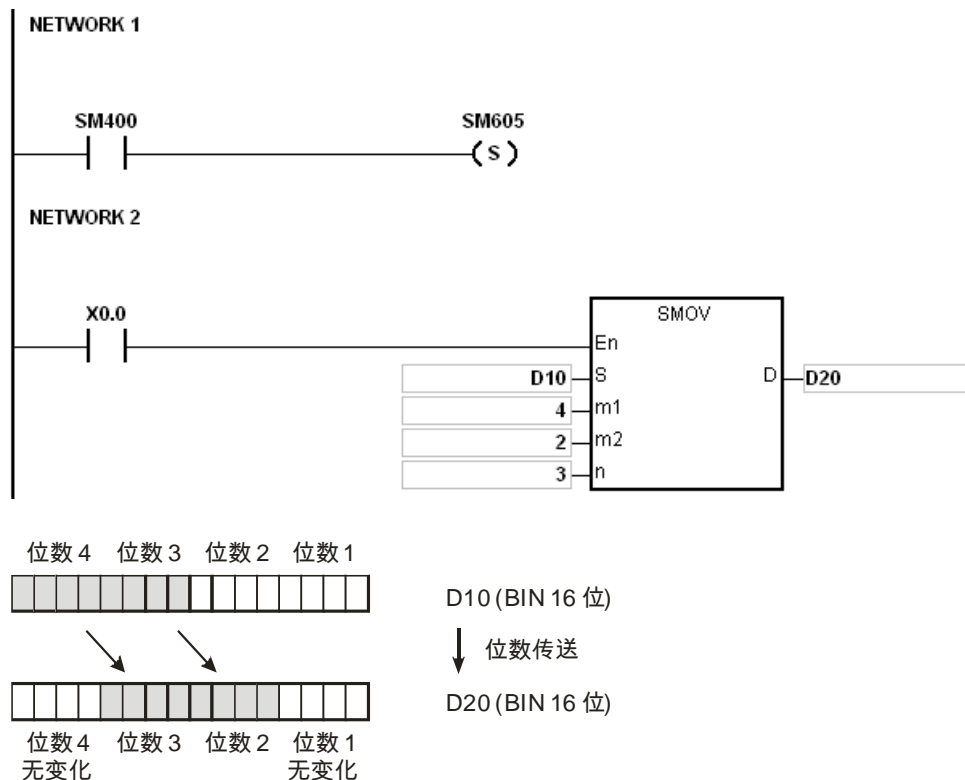


若执行前 D10=1234 · D20=5678 · 执行完毕后 · D10 不变 · D20=5128。

6

程序范例二：

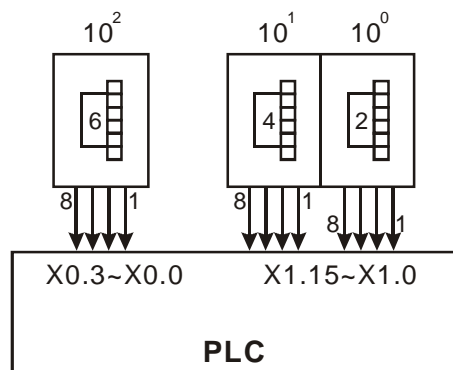
当 SM605=ON (BIN 模式) 时，使用 SMOV 指令的话，D10、D20 并不会作 BCD 变换，而是以 BIN 型态 4 个位为一个单位作传送。

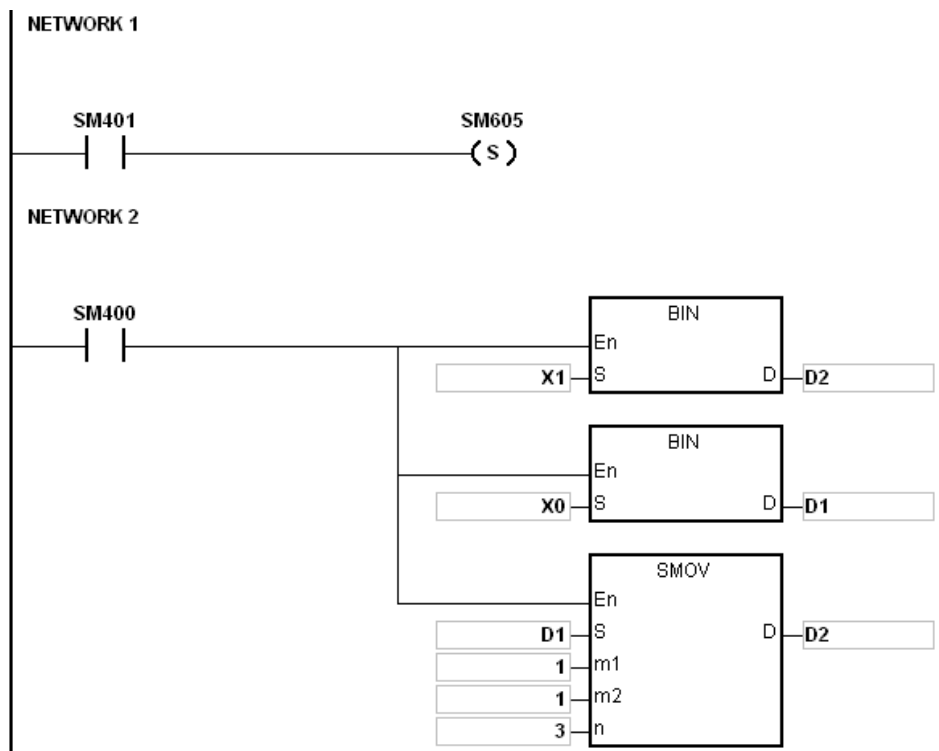


若执行前 D10=16#1234，D20=16#5678，执行完毕后，D10 不变，D20=16#5128。

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 $SR0=16\#200B$ 。

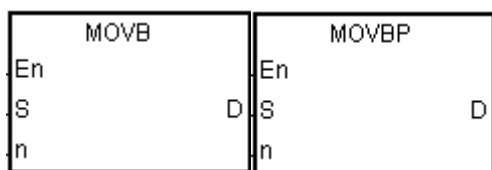
6

API	指令码			操作数								功能					
0310		MOVB	P	S · n · D								多位移动					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



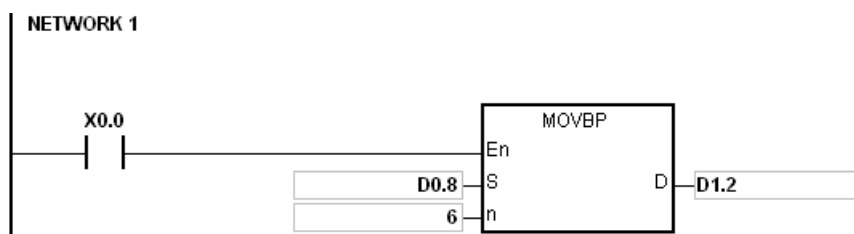
- S** : 要移动的数据来源 Bit
- n** : 传送笔数 Word
- D** : 目的地装置要存放数据来源的起始位数 Bit

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 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，Error Code=16#200B。

程序范例：

X0.0 为 ON 时，PLC RUN 后，D0.8~D0.13 之内容传送至 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。

6.5 程序跳转指令

6.5.1 程序跳转指令一览表

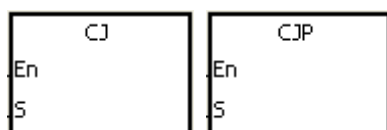
API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>0400</u>	CJ	—	✓	条件跳转	3
<u>0401</u>	JMP	—	—	无条件跳转	3
<u>0402</u>	GOEND	—	—	跳转到 END	1

6.5.2 程序跳转指令说明

API	指令码			操作数								功能					
0400		CJ	P	S								条件跳转					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S																	

脉冲执行型	16 位指令 (3 steps)	32 位指令
AH500	AH500	-

符号：



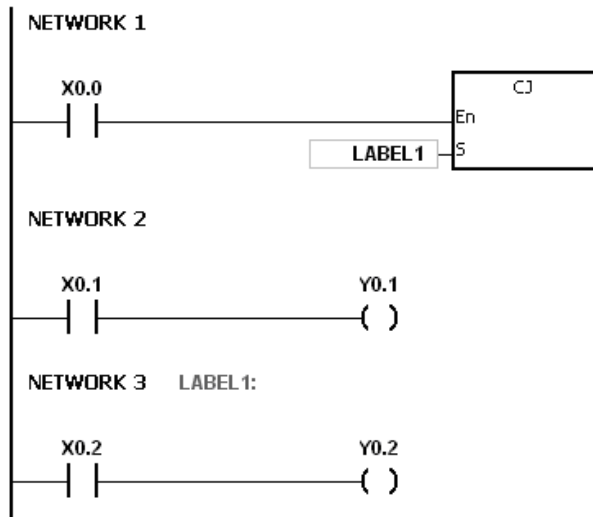
S：条件跳转的目的指标

指令说明：

1. 当使用者希望 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 所指的程序若在 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标 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。
 - 一般应用指令不会被执行。

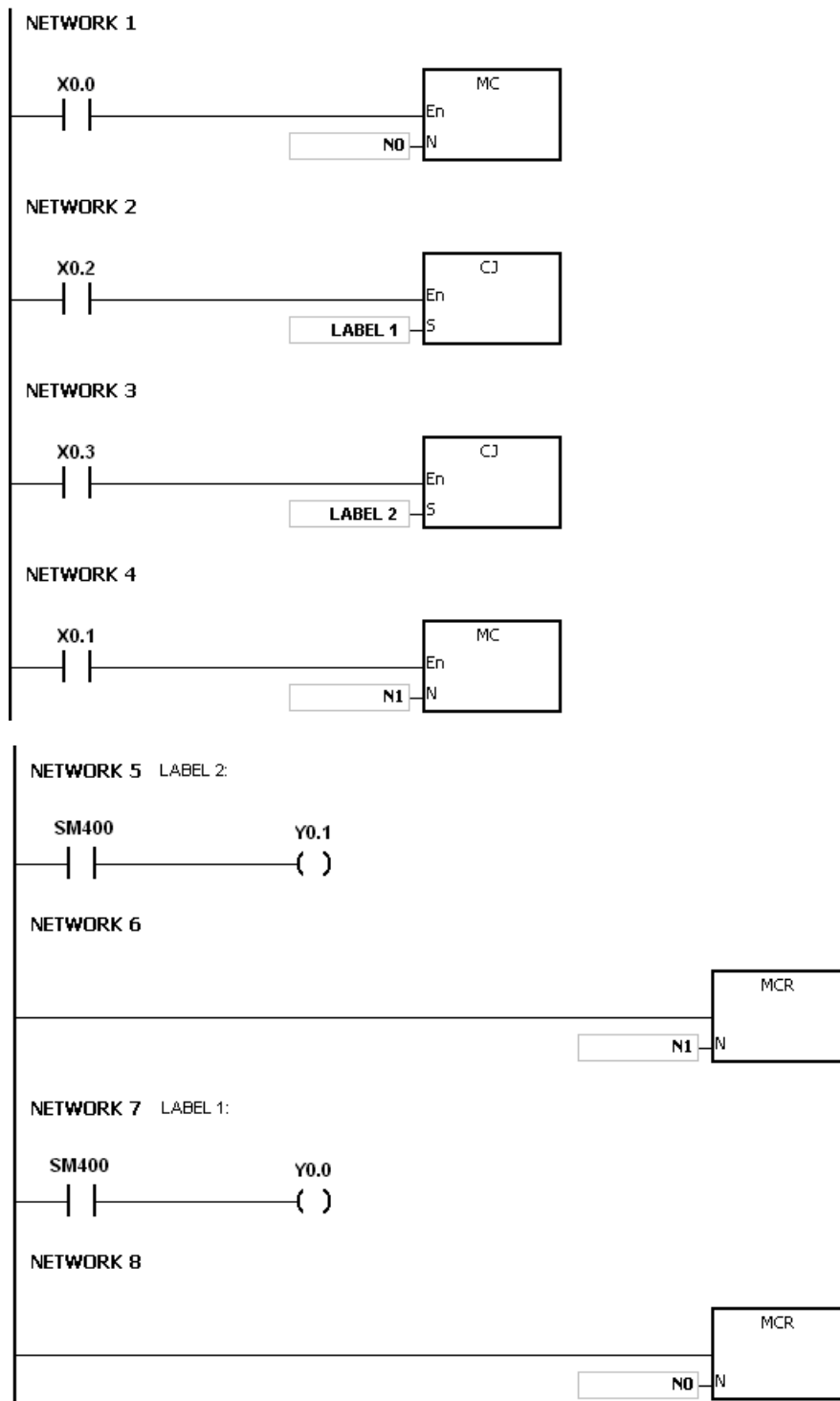
程序范例一：

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，中间地址跳过不执行。
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 内部自订的堆栈中，而此堆栈由 PLC 自行控制，使用者无法改变；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。



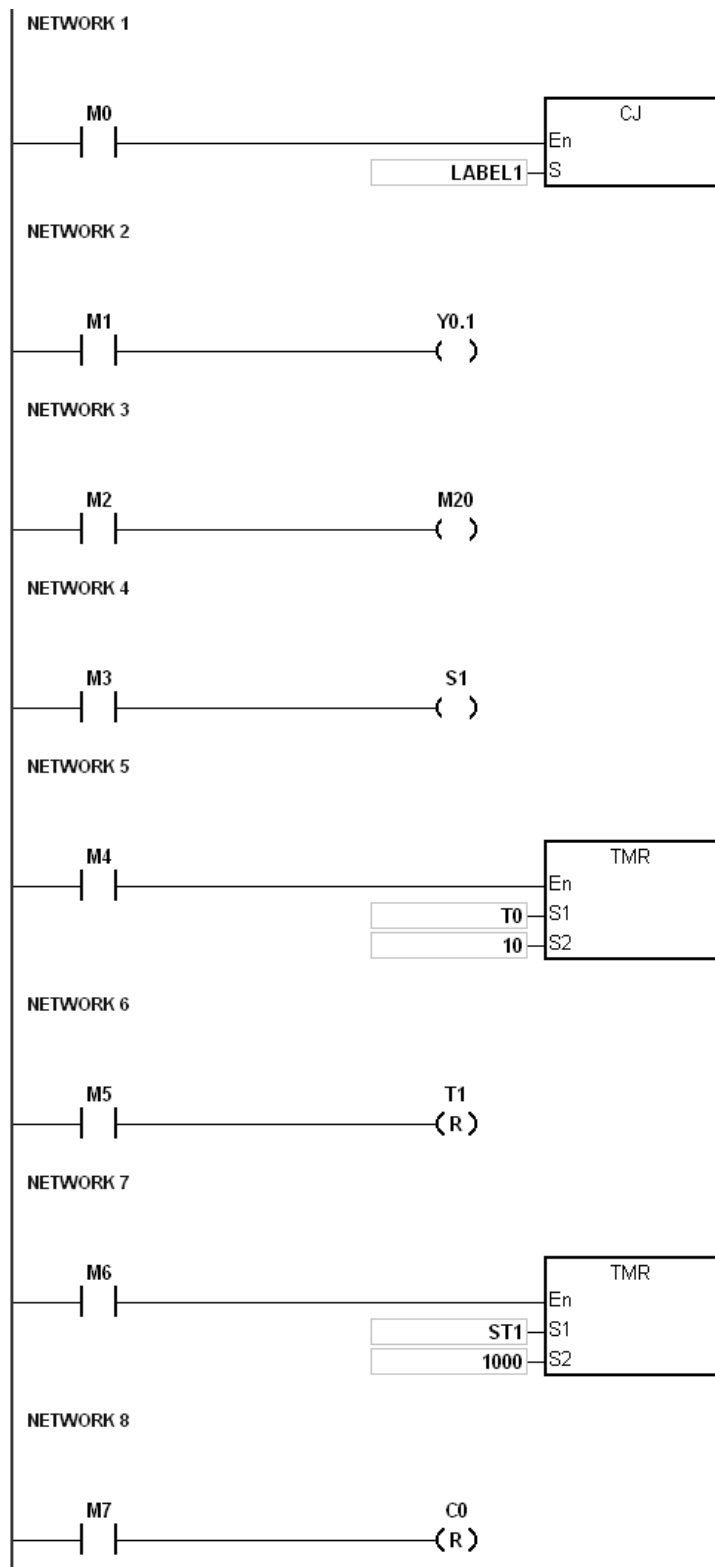
6

程序范例三：

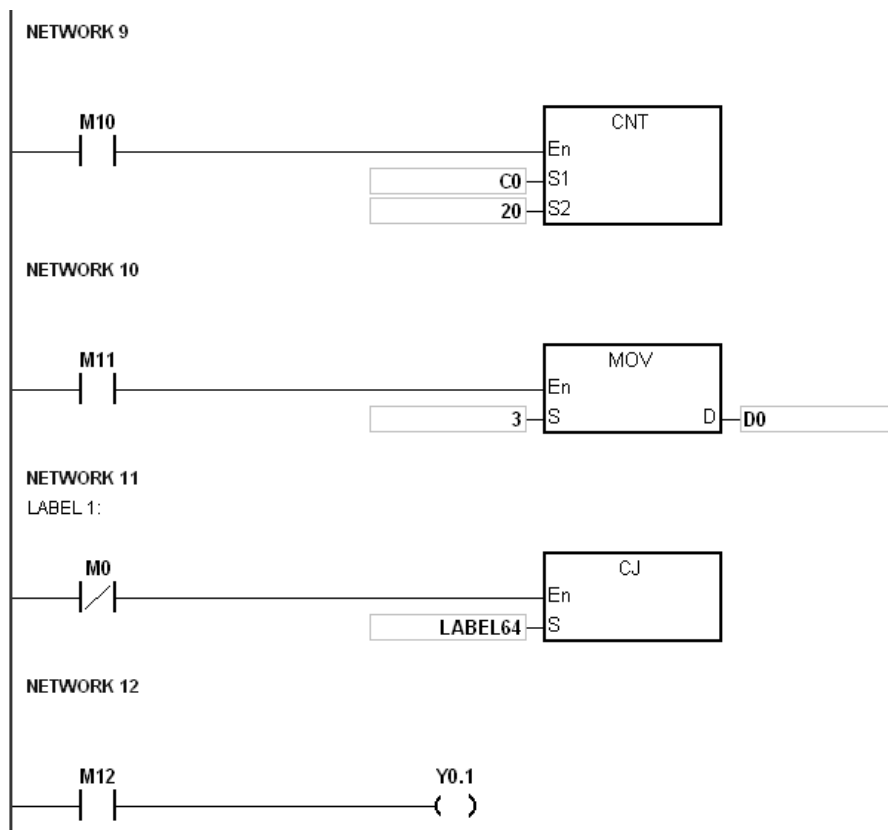
底下表格为下列程序中，各个装置状态变化。

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。



6



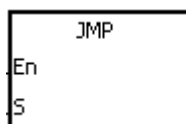
补充说明：

指标 P 在 ISPSOft 中的使用方式，请参考 ISPSOft 的使用手册说明。

6

API	指令码			操作数								功能					
0401		JMP		S								无条件跳转					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S																	
脉冲执行型											16 位指令 (3 steps)			32 位指令			
-											AH500			-			

符号：



S : 跳转之目的指标

指令说明：

1. 无条件跳转到程序中的某个 P 指标。
2. 指针 P 所指之程序若在 JMP 指令之前，需注意会发生 WDT 逾时之错误，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 保持跳转发生前之状态。
 - 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。

API	指令码	操作数	功能
0402	GOEND	—	跳转到 END

脉冲执行型	16 位指令 (1 step)	32 位指令
-	AH500	-

符号：

GOEND

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，所有装置组件内容值与状态保持不变。

6.6 程序执行控制指令

6.6.1 程序执行控制指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>0500</u>	DI	–	–	中断插入禁止	1
<u>0501</u>	EI	–	–	中断插入允许	1
<u>0502</u>	IMASK	–	–	中断控制	3

6.6.2 程序执行控制指令说明

API	指令码	操作数	功能
0500	DI	-	中断插入禁止

脉冲执行型	16 位指令 (1 step)	32 位指令
-	AH500	-

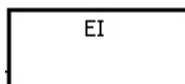
符号：

DI

API	指令码		操作数	功能
0501		EI	-	中断插入允许

脉冲执行型	16 位指令 (1 step)	32 位指令
-	AH500	-

符号：



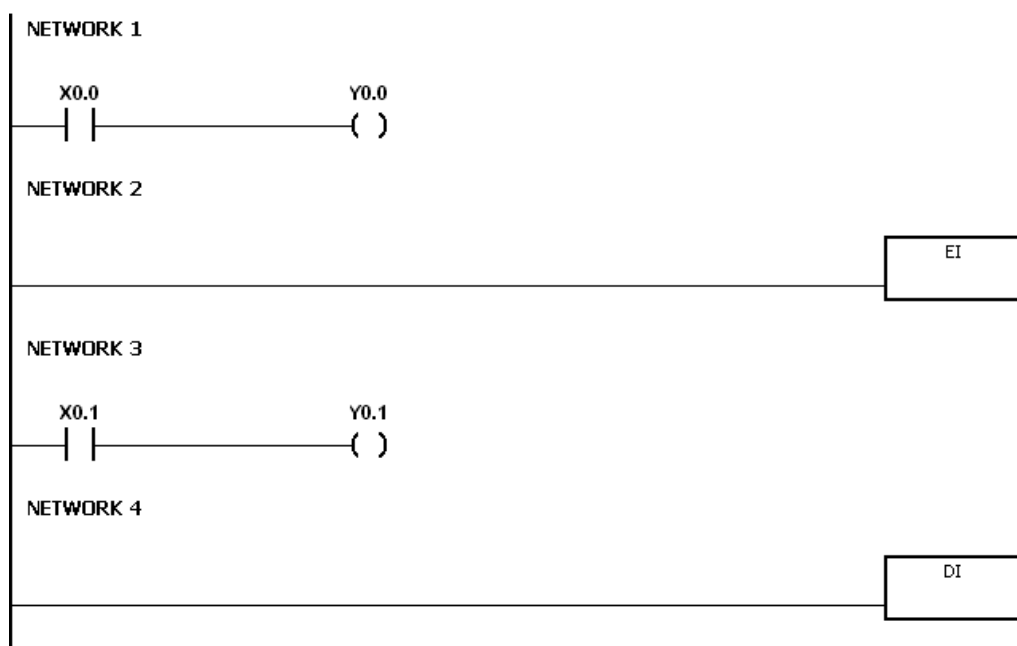
指令说明：

1. EI 表示程序中允许使用中断工作 (TASK) (TASK I0~TASK I255，请参考 AH500 操作手册第 6.6 节)。
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，同时发生以指标编号较小者优先。
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：



6

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 API1800 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

Bit 15 ~8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
保留位	背板使用							

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

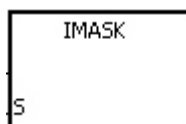
中断编号	说明	优先级
I0	I/O 中断 0	1
I1	I/O 中断 1	2
I2	I/O 中断 2	3
I3	I/O 中断 3	4
I4	I/O 中断 4	5
I5	I/O 中断 5	6
I6	I/O 中断 6	7
I7	I/O 中断 7	8
I8	I/O 中断 8	9
I9	I/O 中断 9	10
I10	I/O 中断 10	11
I11	I/O 中断 11	12
I12	I/O 中断 12	13
I13	I/O 中断 13	14
I14	I/O 中断 14	15
I15	I/O 中断 15	16
I16	I/O 中断 16	17
I17	I/O 中断 17	18
I18	I/O 中断 18	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	RS 通讯请求中断 COM1	33
I33	RS 通讯请求中断 COM2	34

中断编号	说明	优先级
I34	24V 低电压检测中断(LV2)：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	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

6

API	指令码			操作数								功能					
0502	IMASK			S								中断控制					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				
脉冲执行型											16 位指令 (3 steps)			32 位指令			
-											AH500			-			

符号：



S : 数据来源

Word

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 Error Code=16#2003。

6.7 I/O更新指令

6.7.1 I/O更新指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>0600</u>	REF	—	✓	I/O 更新处理	5

6

6.7.2 I/O更新指令说明

API	指令码			操作数								功能				
0600		REF	P	D · n								I/O 更新处理				

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号:



D : 欲 I/O 更新处理之起始装置 Bit

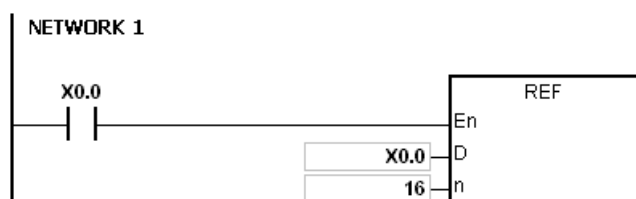
n : I/O 更新处理之数目 1~256

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。因此在演算过程中需要最新的 I/O 数据, 则可利用本指令。

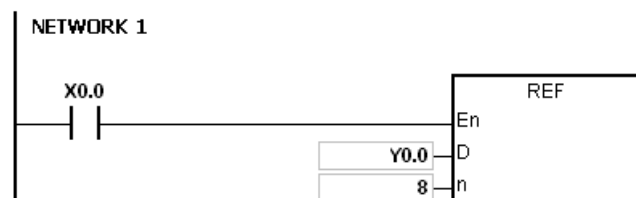
程序范例一:

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 之 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。

6.8 便利指令

6.8.1 便利指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>0700</u>	ALT	—	✓	ON/OFF 交替	3
<u>0701</u>	TTMR	—	—	示教式定时器	5
<u>0702</u>	STMR	—	—	特殊定时器	7
<u>0703</u>	RAMP	—	—	斜坡信号	9
<u>0704</u>	MTR	—	—	矩阵输入	9
<u>0705</u>	ABSD	DABSD	—	绝对方式凸轮控制	9
<u>0706</u>	INCD	—	—	相对方式凸轮控制	9
<u>0707</u>	—	DPID	—	PID 运算	35
<u>0708</u>	—	DPIDE	—	PID 运算	43

6

6.8.2 便利指令说明

API	指令码			操作数										功能			
0700		ALT	P	D										ON/OFF 交替			
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●				●	●	●			●				

脉冲执行型	16 位指令 (3 steps)	32 位指令
AH500	AH500	-

符号：



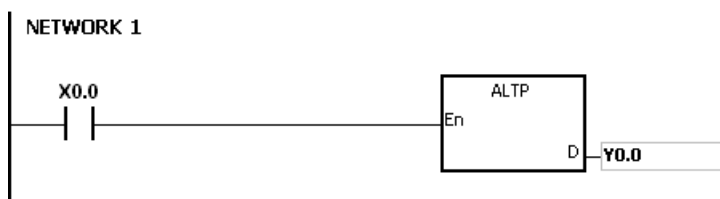
D : 目的地装置 Bit

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）。

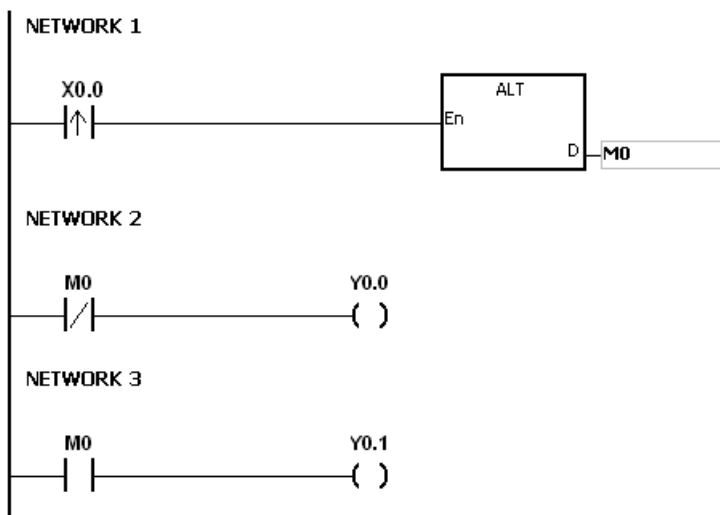
程序范例一：

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。



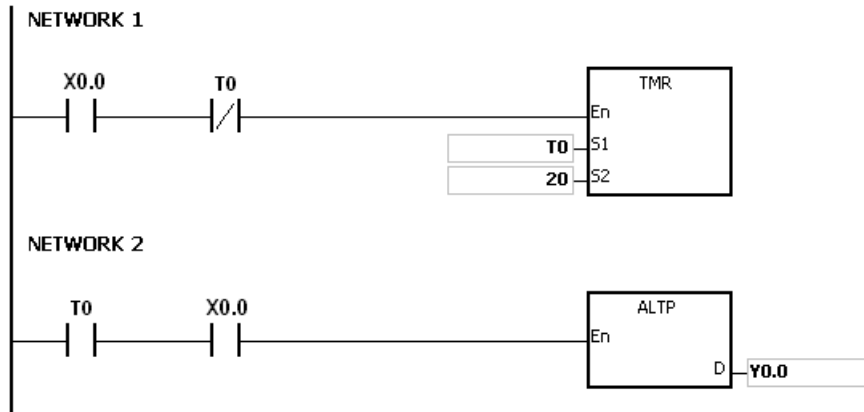
程序范例二：

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。



API	指令码		操作数											功能			
0701	TTMR		D · n											示教式定时器			
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●						●	●		●		●				
n	●	●						●	●		●		●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
-	AH500	-

符号：



D : 储存按钮开关 ON 时间之装置编号 Word

n : 倍数设定 Word

指令说明：

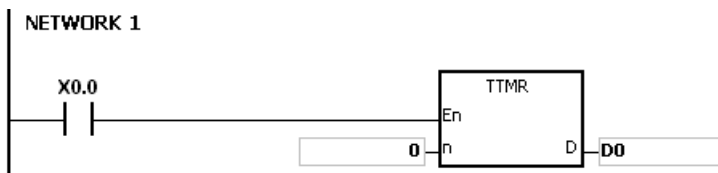
1. 将开关 ON 时间以秒为单位乘以 **n** 倍数后存放于 **D** 内，**D+1** 为系统内部使用，指令执行时不可更改，否则会造成时间计算错误。
2. **D** 的内容值在条件接点为 ON 的时候会先清为 0。
3. 倍数设定：**n=0** 时，**D** 以秒为单位，**n=1** 时，**D** 乘以 10 倍以 100ms 为单位，**n=2** 时，**D** 乘以 100 倍以 10ms 为单位。

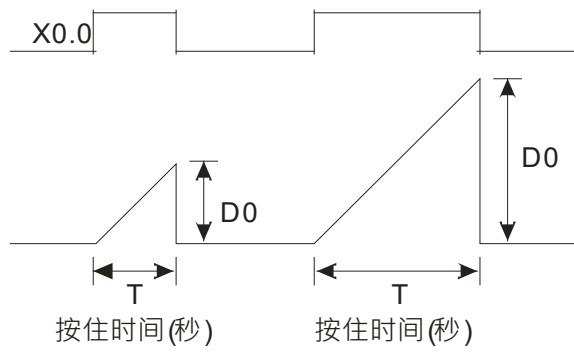
n	D
K0 (单位：s)	1×T
K1 (单位：100 ms)	10×T
K2 (单位：10 ms)	100×T

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。





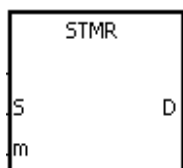
补充说明：

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] of WORD/INT。

6

API	指令码				操作数								功能				
0702	STMR				S · m · D								特殊定时器				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S					○												
m	●	●						●	●		●		●	○	○		
D	●	●	●	●				●	●	●			●				
										脉冲执行型		16 位指令 (7 steps)			32 位指令		
										-		AH500			-		

符号：



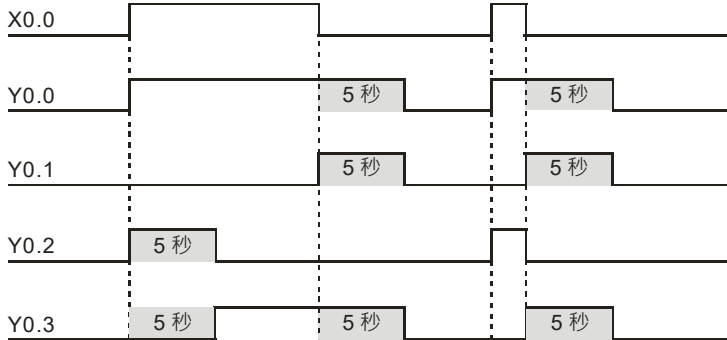
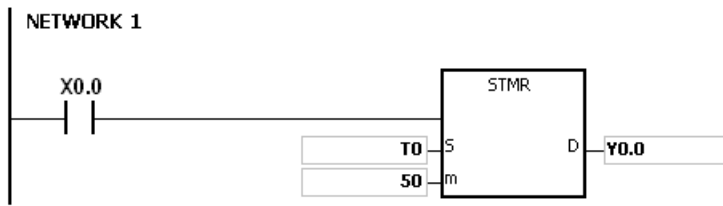
S : 定时器编号 T0~T2047
m : 定时器设定值 Word
D : 输出装置之起始编号 Bit

指令说明：

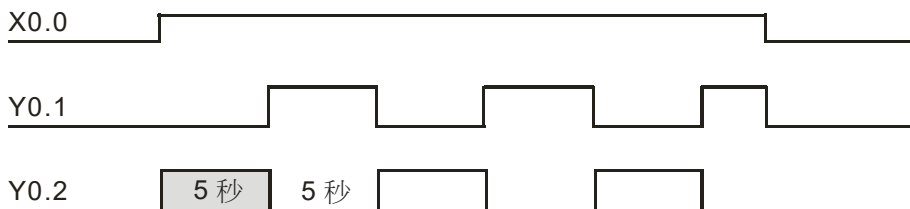
1. STMR 指令，用来产生 OFF 延迟，一次触发及闪烁回路的专用指令。
2. STMR 指令所指定的 T 定时器以 100ms 为单位。
3. STMR 指令所指定的定时器号码不可重复使用。
4. **D** 连续占用 4 个 bit。
5. 执行此指令前，请先 RST D~D+3 的状态。
6. 当条件接点未启动时，如果装置内容为以下两种，则 D、D+1、D+3 的状态会 ON m 秒之后再 OFF，除了以下两种情形之外，D~D+3 的状态会保持 OFF。
 - T 定时器内容值如果小于或等于 m，且 D 的状态为 ON，D+1 的状态为 OFF。
 - T 定时器内容值小于 m，且 D、D+1、D+3 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 的内容被复归为 0。



补充说明：

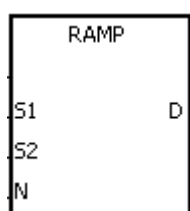
1. **D+3** 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **m<=0**，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. **D** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。

API	指令码		操作数				功能						
0703		RAMP	$S_1 \cdot S_2 \cdot D \cdot n$				斜坡信号						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
-	AH500	-

符号：



S_1 : 斜坡信号之起点设定 Word
 S_2 : 斜坡信号之终点设定 Word
D : 斜坡信号之经过时间值 Word
n : 扫描次数 Word

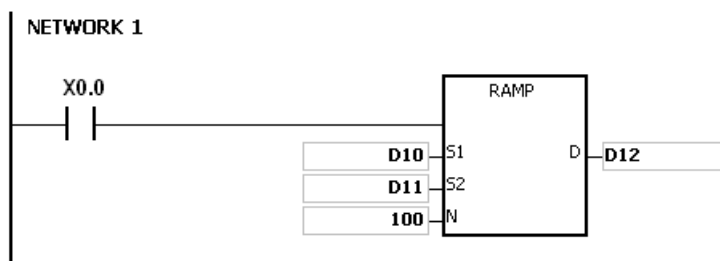
指令说明：

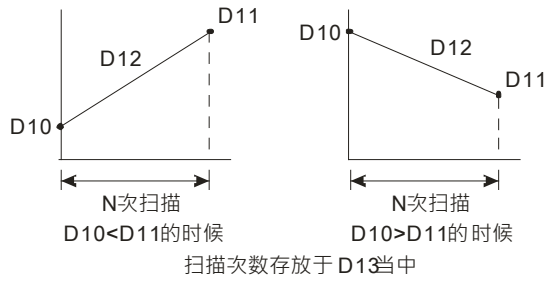
1. 本指令是一个求斜率的指令，斜率是线性与扫描时间有绝对的关系，因此使用本指令时，通常必须预先将扫描时间加以固定。
2. 预先将倾斜信号之起点设定值写入 S_1 及倾斜信号之终点设定值写入 S_2 内，当条件接点=ON 时，D 从 S_1 所设定的值朝 S_2 迈进（增加），扫描次数存放于 $D+1$ 当中，当 $D = S_2$ 或 $D+1=n$ 已到达扫描次数时，SM687 完成标志=ON。
3. 当条件接点未启动时， $D1$ 、 $D+1$ 的内容值为 0，SM687 完成标志=OFF。
4. 使用在线编辑时，请重新启动条件接点，以达到指令的初始化。
5. 若需要固定扫描时间，请参考 ISPSOFT 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。

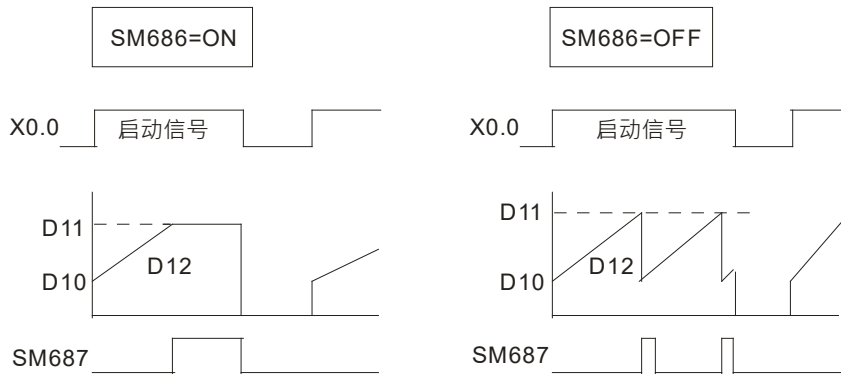
1. 指令执行中，启动信号 X0.0 变成 OFF 时，指令停止执行，当 X0.0 再度 ON 的时候，SM687=OFF，D12 的内容被复归为 D10 的设定值，D13 的内容被清除为 0，再重新计算。
2. SM686=OFF 时，SM687=ON，D12 的内容被复归成 D10 的设定值，D13 的内容被清除为 0。





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. 启动模式标志信号 SM686 的 ON/OFF，D12 的内容变化如下：



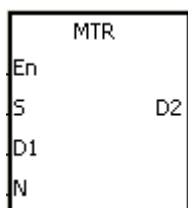
6

API	指令码		操作数				功能						
0704	MTR		S · D ₁ · D ₂ · n				矩阵输入						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
-	AH500	-

符号：



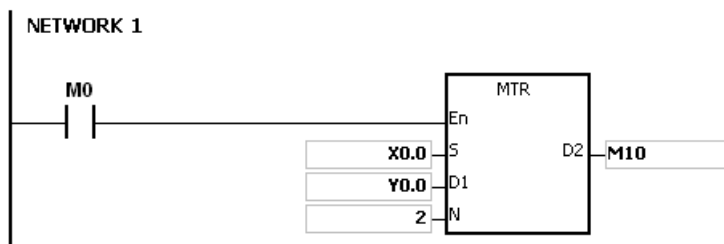
- S : 矩阵扫描输入起始装置 Bit
- D₁ : 矩阵扫描输出起始装置 Bit
- D₂ : 矩阵扫描所对应起始装置 Bit
- n : 矩阵扫描之列数 Word

指令说明：

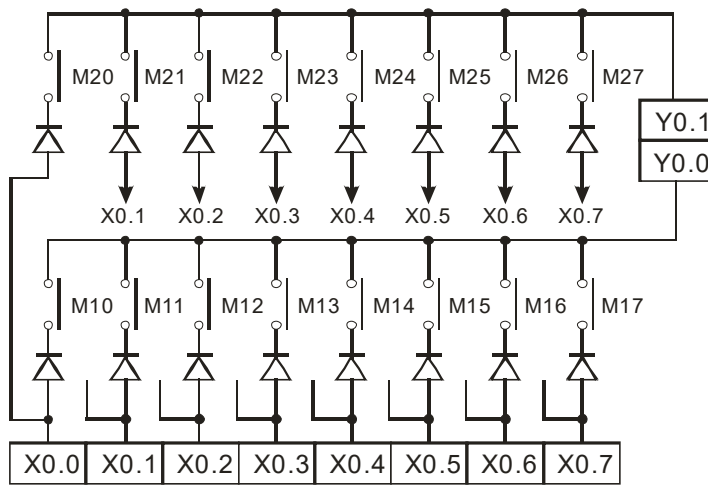
- S 指定矩形所有连接输入端的起始号码，一旦指定后，该号码开始算连续 8 点为矩阵输入端。
- D₁ 指定那一个晶体管输出 Y 为矩阵扫描的起始号码，条件接点 OFF 时，D₁ 起始的 n 个装置状态会保持 OFF。
- 每一次扫描周期可以更新一列的输入点，每一列为 16 点，从 1~n 列反复的扫描。
- 本指令由 S 起始的连续 8 个输入端，以 D₁ 起始的 n 个外部输出点用矩阵扫描的方式读取 n 列的 8 个开关，变成 8×n 的多点矩阵输入点。并将扫描读取的开关状态反应在 D₂ 起始之装置。
- 使用本指令时，最大可将 8 个输入开关并接 8 列可得 64 个输入点 (8×8=64)。
- 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。
- 本指令的条件接点一般都使用常 ON 接点 SM400。
- n 操作数的范围 2~8。

程序范例一：

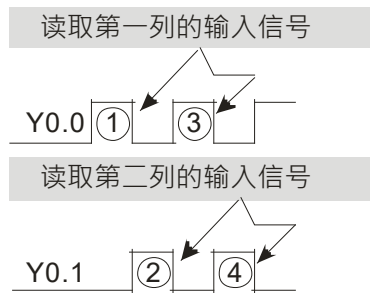
- 当 M0=ON，MTR 指令开始执行，外部 2 列共 16 个开关之状态被顺序读取并存放在内部继电器 M10~M17，M20~M27。



2. 下图由 X0.0~X0.7 及 Y0.0~Y0.7 构成 2 列矩阵输入回路之外部接线图，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。



6

补充说明：

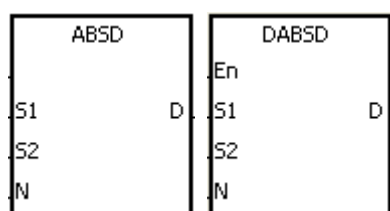
1. 本指令被执行时，当扫描周期太长或太短都可能造成按键输入不良因此可运用下列技巧来克服。
2. 当扫描周期太短时，可能造成 I/O 来不及反应而无法读取正确之按键输入，此时，可将扫描时间加以固定。
3. 当扫描周期太长时，可能会使按键反应变得迟钝，可将此指令写在时间中断工作 (TASK) 内，固定时间执行此指令。
4. **S+7**、**D₁+n-1**、**D₂+ (n*8) -1** 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
5. **n<2** 或 **n>8** 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
6. **S** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of BOOL。

API	指令码			操作数								功能				
0705	D	ABSD		$S_1 \cdot S_2 \cdot D \cdot n$								绝对方式凸轮控制				

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

脉冲执行型	16 位指令 (9 steps)	32 位指令 (9 steps)
-	AH500	AH500

符号：



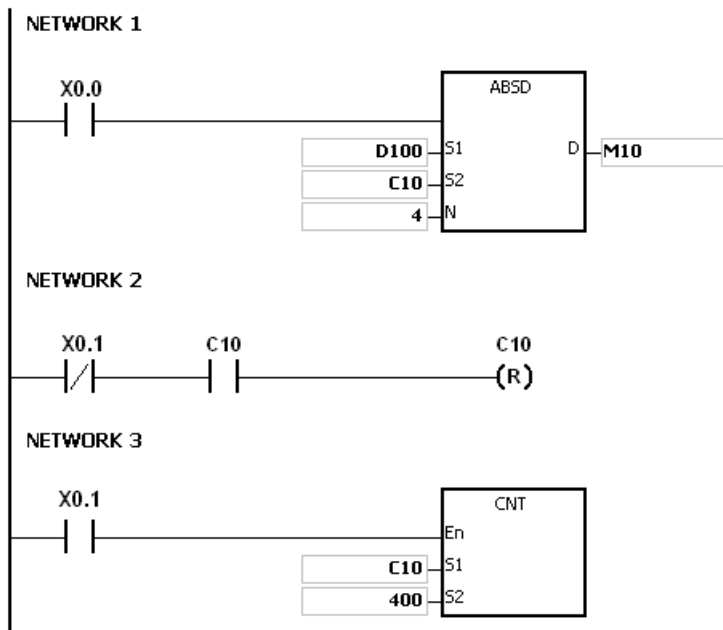
- S_1 : 指比较表起始装置 Word/Double
- S_2 : 比较值 Word/Double
- D : 比较结果起始编号 Bit
- n : 多段比较的组数 Word

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 $n=1\sim 256$ ，DABSD 时 $n=1\sim 128$ 。

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。

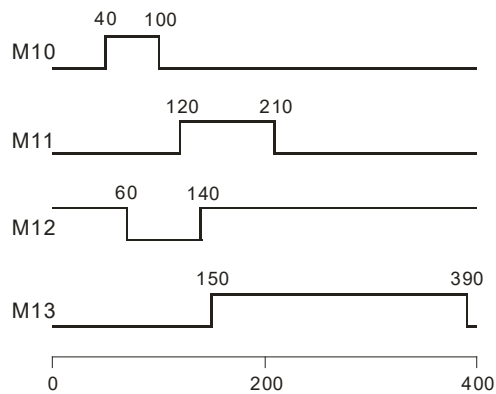


4. 在大于等于下限值且小于等于上限值范围内对应之 M10~M13 会 ON，否则为 OFF。

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



6

补充说明：

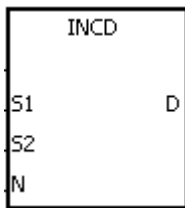
1. ABSD 指令中， $S+2*n-1$ 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. DABSD 指令中， $S+4*n-1$ 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. ABSD 指令中， $D+n-1$ 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. DABSD 指令中， $D+2*n-1$ 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
5. ABSD 指令中，若 $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
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。

API	指令码			操作数								功能					
0706		INCD		$S_1 \cdot S_2 \cdot n \cdot D$								相对方式凸轮控制					

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
-	AH500	-

符号：



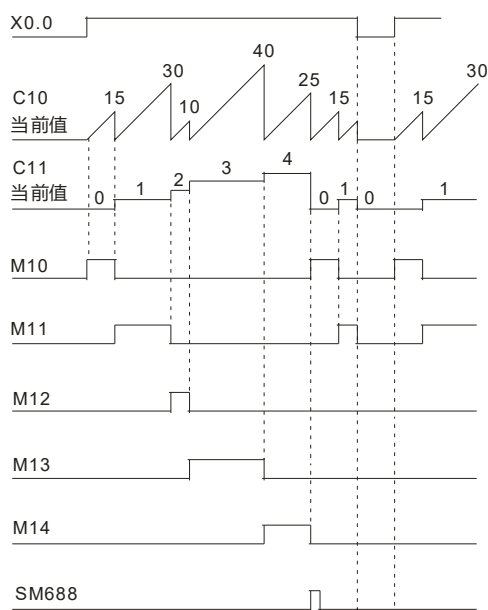
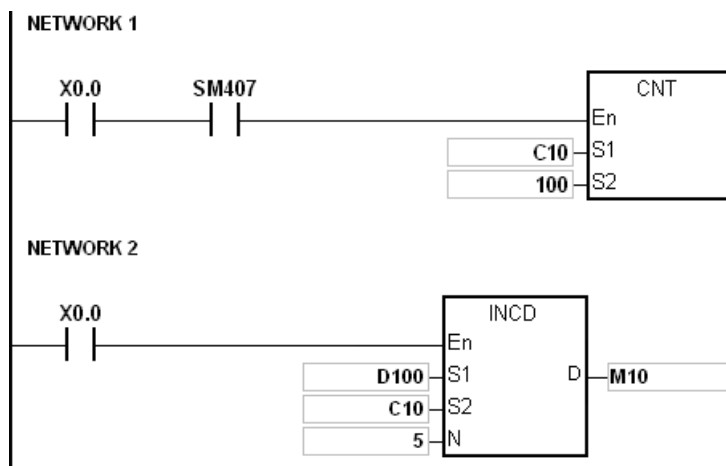
- S_1 : 指比较表起始装置 Word
- S_2 : 计数器编号 Word
- D : 比较结果起始编号 Bit
- n : 多段比较的组数 Word

指令说明：

1. 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。
2. 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 的当前值被复归为 0 重新计数。而目前执行之段数被暂存于 S_2+1 当中。
3. 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。
4. 当条件接点未启动时， S_2 、 S_2+1 的内容值为 0， $D \sim D+n-1$ 状态为 OFF，SM688 为 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。
6. n 操作数的范围 1~256。

程序范例：

1. 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。
2. 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 的当前值被复归为 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 全部被复归为 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：

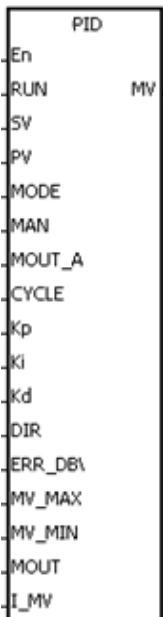
1. S_2+1 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. S_1+n-1 超出装置范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. $D+n-1$ 超出装置范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
5. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。

API	指令码			操作数								功能					
0707	D	PID		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								PID 运算					

装置	X	Y	M	S	T	C	HC	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 (35 steps)
-	-	AH500

符号：



- PID_RUN** : 启动 PID 运算 BIT
- SV** : 目标值 (SV) DWORD
- PV** : 当前值 (PV) DWORD
- PID_MODE** : PID 控制模式 DWORD
- PID_MAN** : PID A/M 模式 (PID_MAN) BIT
- MOUT_AUTO** : MOUT_AUTO BIT
- CYCLE** : 取样时间 (CYCLE) DWORD
- Kp** : P 计算值系数 (Kp) DWORD
- Ki** : I 计算值系数 (Ki) DWORD
- Kd** : D 计算值系数 (Kd) DWORD
- PID_DIR** : PID 正反向 (PID_DIR) BIT

ERR_DBW	: ERR 的不作用范围 (ERR_DBW)	DWORD
MV_MAX	: MV 上限值 (MV_MAX)	DWORD
MV_MIN	: MV 下限值 (MV_MIN)	DWORD
MOUT	: MV 手动值 (MOUT)	DWORD
I_MV	: 累计积分项的数值 (I_MV)	DWORD
MV	: MV 输出值 (MV)	DWORD

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	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	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 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。	

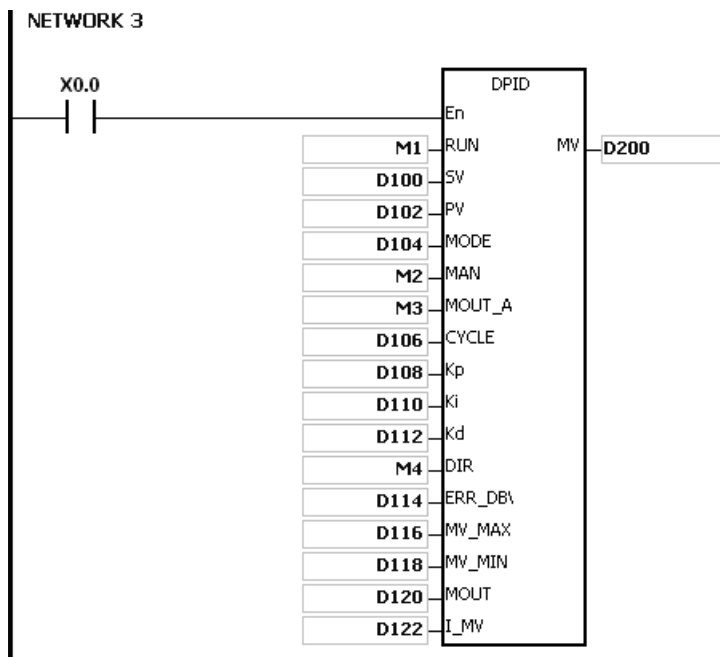
6

装置编号	数据类型	功能		设定范围	说明
Cycle	DWORD/ DINT	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。
K_P	REAL	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
K_I	REAL	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
K_D	REAL	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	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 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
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 , 需大于 S3+7 , 否则上限值与下限值将互换
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
MOUT	REAL	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)	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

装置编号	数据类型	功能	设定范围	说明
				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	REAL	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. 指令无使用次数之限制，但是 I_MV~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：

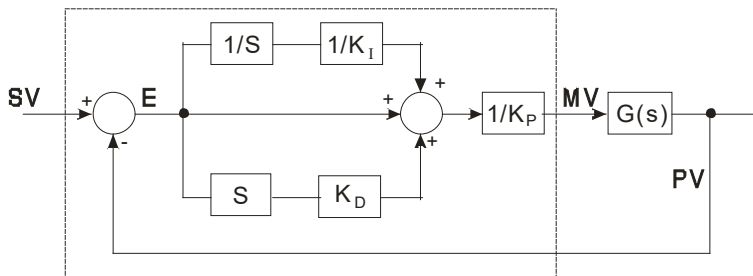
$$MV = K_p E(t) + K_I \int_0^t E(t) dt + K_D * \frac{dE(t)}{dt}$$

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 演算方式
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，再除以取样时间之动作，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。

虚线内为PID指令



- 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

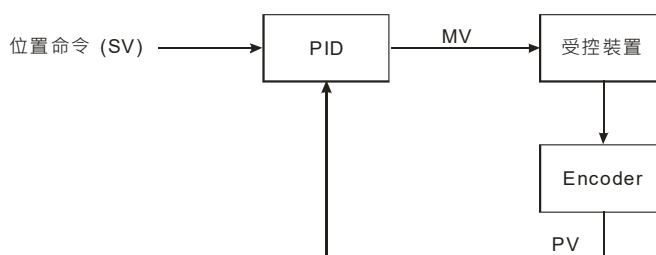
注意事项和建议：

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，就请勿使用于伺服电机控制环境中，以免造成控制不当的现象发生。

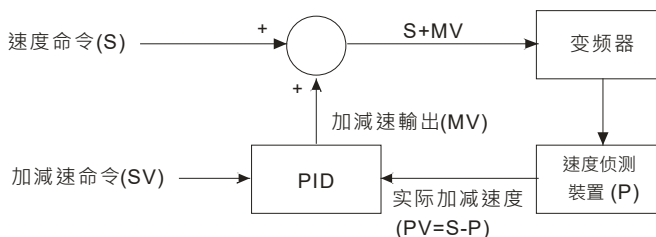
6

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 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 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，以免造成无法控制之现象。

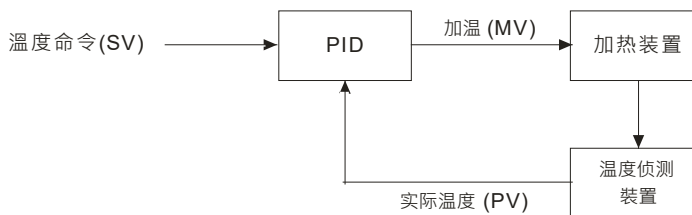
范例 1：使用 PID 指令于位置控制时的方块图 ($PID_MODE=0、2$)



范例 2：与变频器搭配控制时之方块图 ($PID_MODE=0、2$)



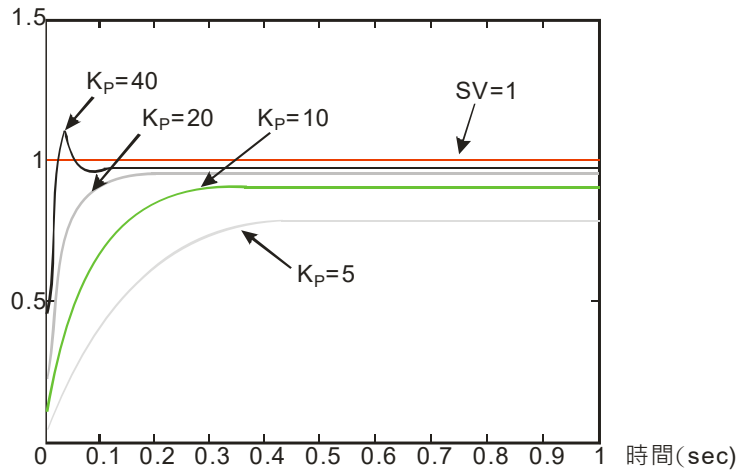
范例 3：使用 PID 指令于温度控制时的方块图 ($PID_MODE=0、2$)



范例 4：PID 指令参数调整建议步骤说明

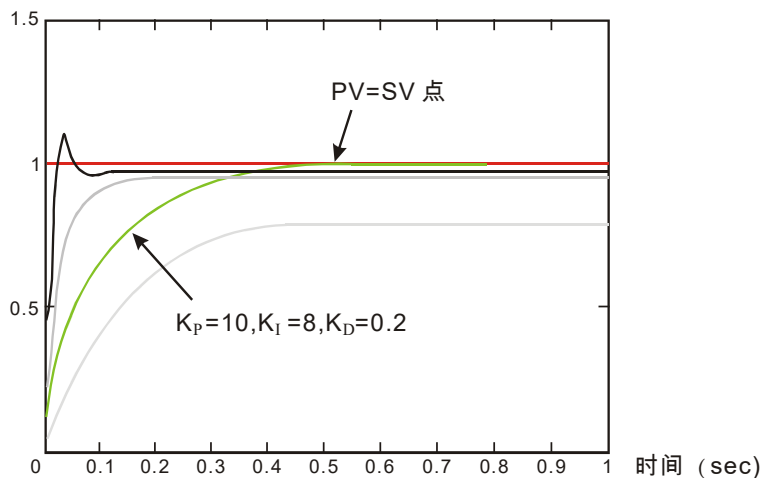
假设控制系统之受控体 $G(s)$ 的转移函数为一阶的函数 $G(s) = \frac{b}{s+a}$ (一般伺服电机的模型均为此函数)，命令值 SV 为 1，取样时间 T_s 为 10ms。建议调整步骤如下：

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之状况，自行调整其适合之控制参数。

实例 1：使用自动调整功能控制温度

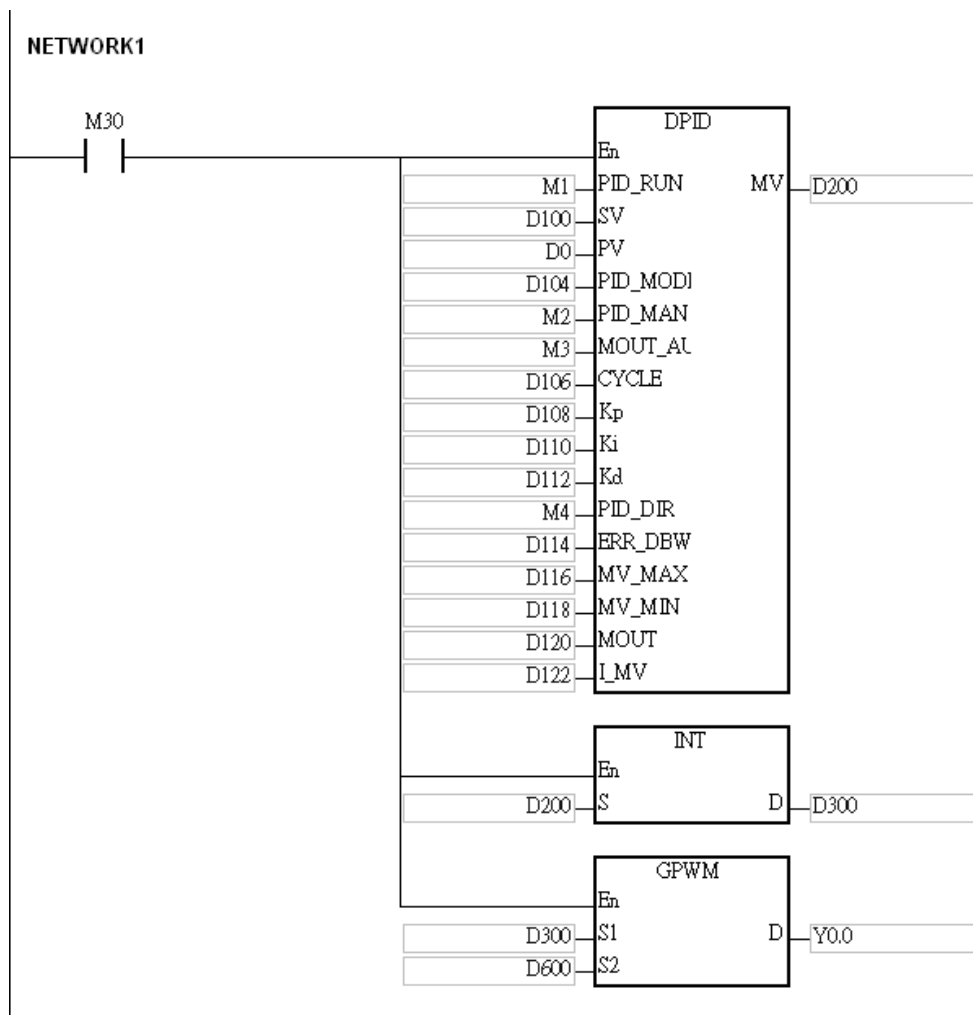
控制目的：利用自动调整功能计算出最佳的 PID 温度控制的参数。

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

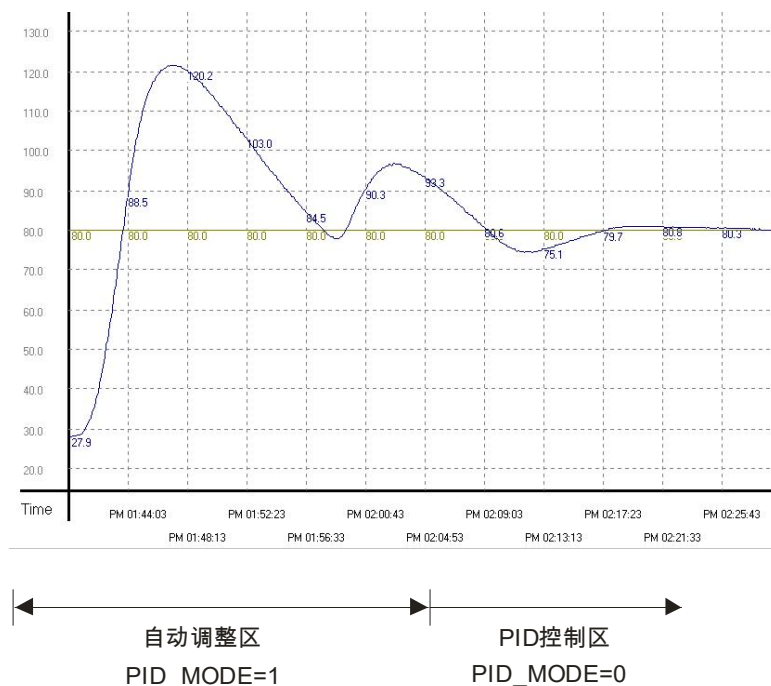
($PID_MODE=1$) 做一初步调整，待调整完毕后，本指令将自动修改控制模式为 ($PID_MODE=0$) 。

本实例的控制环境为烤箱。范例程序如下图所示：

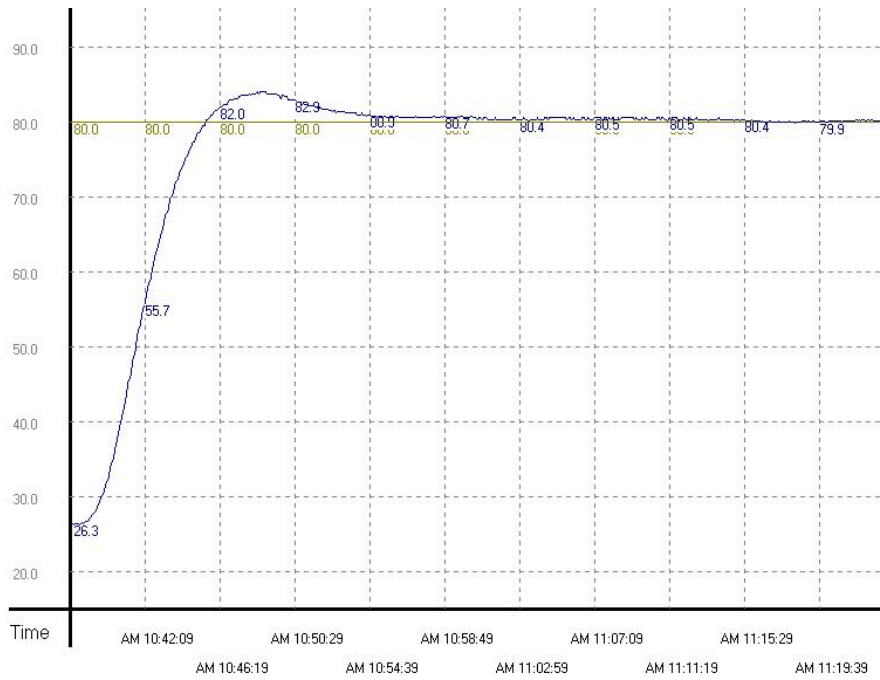


6

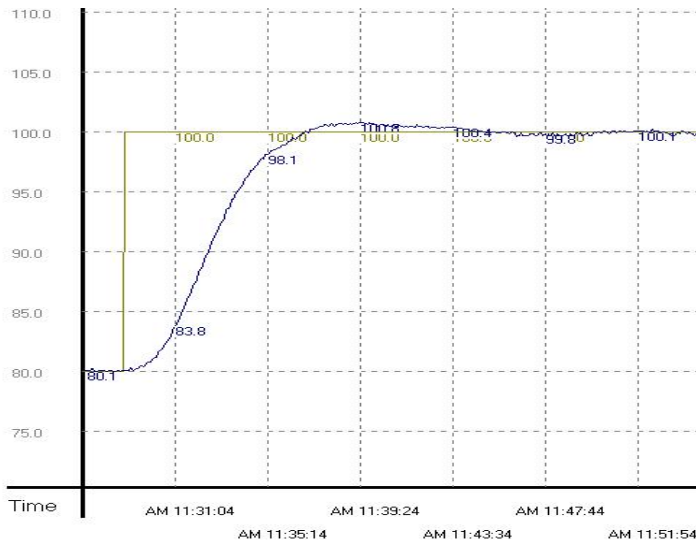
自我调整功能的实验结果如下所示：



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我调整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：



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。

6

API	脚本		操作数								功能						
0708	D	PIDE	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								PID 运算						

装置	X	Y	M	S	T	C	HC	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								●	●				●				●
MV_MAX								●	●				●				●
MV_MIN								●	●				●				●
MOUT								●	●				●				
BIAS								●	●				●				●
I_MV								●	●				●				
MV								●	●				●				

脉冲执行型	16 位指令	32 位指令 (43 steps)
-	-	AH500

6

符號：

DPIDE	
En	
PID_RUN	MV
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	

PID_RUN	： 启动 PID 运算	BIT
SV	： 目标值	DWORD
PV	： 当前值	DWORD
PID_MODE	： PID 控制模式	DWORD
PID_MAN	： PID Auto/Manual 模式	BIT
MOUT_AUTO	： 手动值（MOUT）自动更新模式	BIT
CYCLE	： 取样时间（ms）	DWORD
Kc_Kp	： 比例项系数	DWORD
Ti_Ki	： 积分项系数（sec 或 1/sec）	DWORD
Td_Kd	： 微分项系数（sec）	DWORD
Tf	： 微分作用时间常数（sec）	DWORD
PID_EQ	： PID 计算式选择	BIT
PID_DE	： PID 微分项误差计算选择	BIT
PID_DIR	： PID 正反向	BIT
ERR_DBW	： ERR 的不作用范围	DWORD
MV_MAX	： MV 输出上限值	DWORD
MV_MIN	： MV 输出下限值	DWORD
MOUT	： MV 手动值	DWORD
BIAS	： 前馈控制输出值	DWORD
I_MV	： 积分项累计值	DWORD
MV	： 输出值	DWORD

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

装置编号	数据类型	功能	设定范围	说明
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。

装置编号	数据类型	功能	设定范围	说明
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 · 主机自动以时间中断的中断时间

装置编号	数据类型	功能	设定范围	说明
				来计算 PID，此时 CYCLE 的设定无效。
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

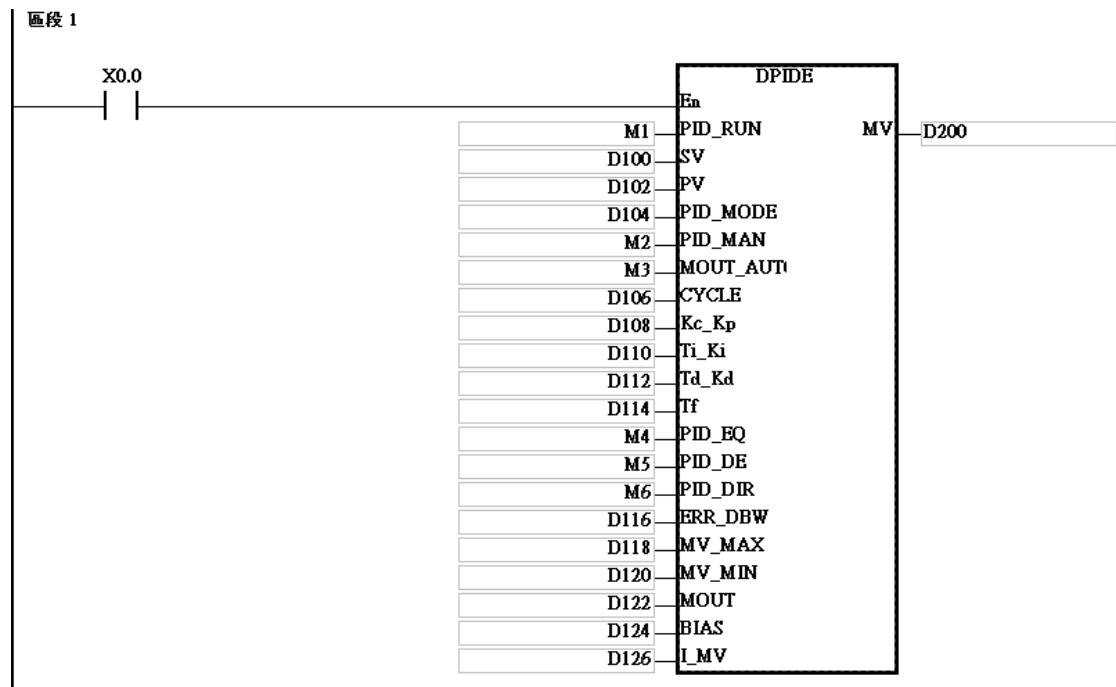
装置编号	数据类型	功能		设定范围	说明
					能·否则主机会去检查这次的 E 值是否小于 ERR_DBW 的绝对值且是否符合 Cross 状态转换条件·若都有则将 E 值视为 0 之后进行 PID 计算·否则依照正常处理将 E 值代入 PID 计算。
MV_MAX	REAL	输出值 (MV) 饱和上限		单精度浮点数范围	例：设定 1,000·则输出值 (MV) 大于 1,000 时将以 1,000 输出。此设定值需大于 MV_MIN·否则主机会自动将 MV_MAX 与 MV_MIN 进行互换。
MV_MIN	REAL	输出值 (MV) 饱和下限		单精度浮点数范围	例：设定-1,000·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

装置编号	数据类型	功能	设定范围	说明
				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	系统记录前次偏差量。
		I_MV+2 ~ I_MV+5		系统用参数，用户请勿使用。
		I_MV+6		系统记录前次 PV 值。
		I_MV+7 ~ I_MV+9		系统用参数，用户请勿使用。
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。

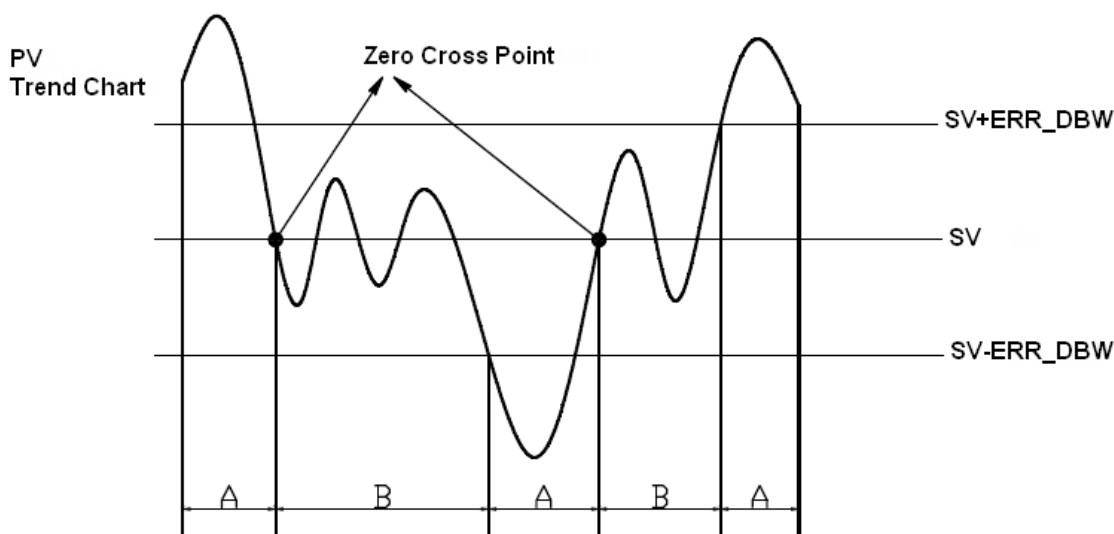
6



补充说明：

1. 指令无使用次数之限制，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 缓存器，于上述程序例当中 **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，主机将会将 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 的区段主机将依照正常的 PID 进行计算，而 B 的区段主机将 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。

甲、Independent Formula & Derivative of E (PID_EQ=False & PID_DE=False)

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

乙、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

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dPV}{dt} + BIAS$$

$$E = PV - SV$$

丙、Dependent Formula & Derivative of E (PID_EQ=True & PID_DE=False)

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dE}{dt} \right] + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

丁、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

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dPV}{dt} \right] + BIAS$$

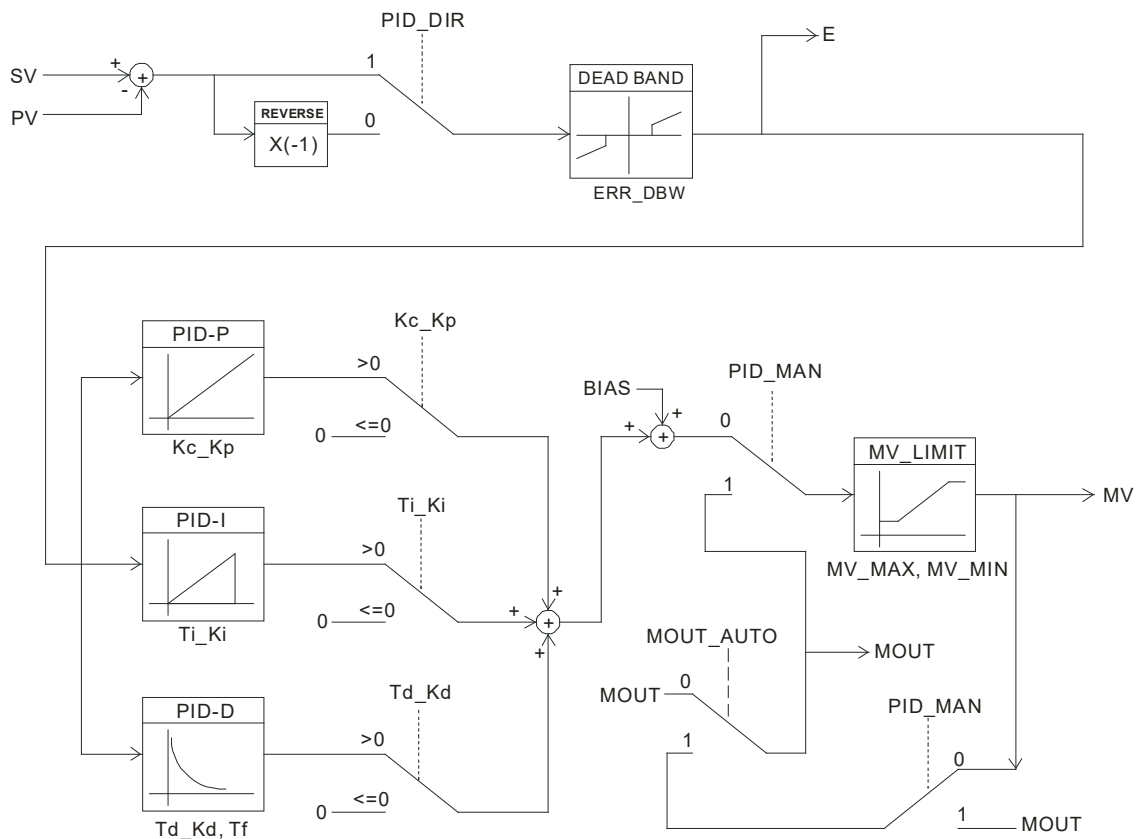
$$E = PV - SV$$

※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。

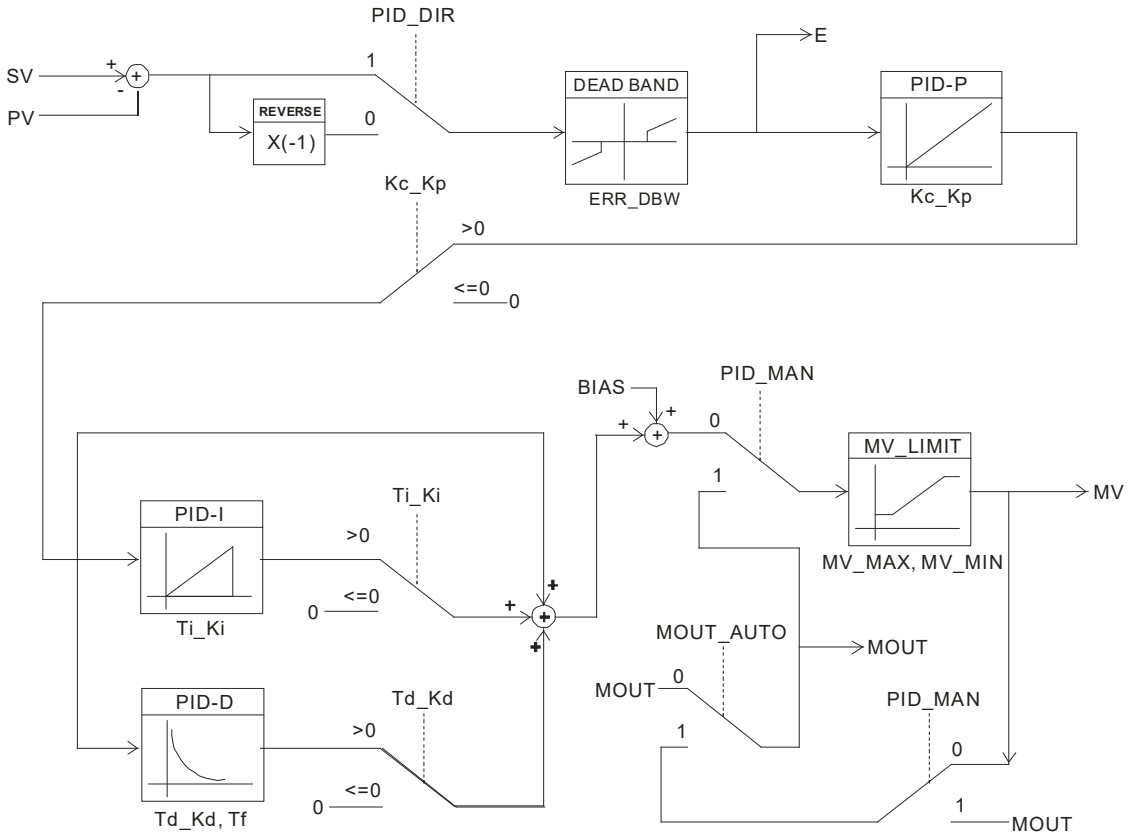
PID 控制方块图：

PID Block Diagram (Independent)



6

PID Block Diagram (Dependent)



6

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伺服电机控制环境中，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，建议使用在停电保持区之 D 缓存器来储存参数，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。

6.9 逻辑操作指令

6.9.1 逻辑操作指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>0800</u>	WAND	DAND	✓	逻辑及 (AND) 运算	7
<u>0801</u>	MAND	—	✓	矩阵及 (AND) 运算	9
<u>0802</u>	WOR	DOR	✓	逻辑或 (OR) 运算	7
<u>0803</u>	MOR	—	✓	矩阵或 (OR) 运算	9
<u>0804</u>	WXOR	DXOR	✓	逻辑异或 (XOR) 运算	7
<u>0805</u>	MXOR	—	✓	矩阵异或 (XOR) 运算	9
<u>0806</u>	WXNR	DXNR	✓	逻辑同或 (XNR) 运算	7
<u>0807</u>	MXNR	—	✓	矩阵同或 (XNR) 运算	9
<u>0809</u>	LD&	DLD&	—	$S_1 \& S_2$	5
<u>0810</u>	LD	DLD	—	$S_1 S_2$	5
<u>0811</u>	LD^	DLD^	—	$S_1 \wedge S_2$	5
<u>0812</u>	AND&	DAND&	—	$S_1 \& S_2$	5
<u>0813</u>	AND	DAND	—	$S_1 S_2$	5
<u>0814</u>	AND^	DAND^	—	$S_1 \wedge S_2$	5
<u>0815</u>	OR&	DOR&	—	$S_1 \& S_2$	5
<u>0816</u>	OR	DOR	—	$S_1 S_2$	5
<u>0817</u>	OR^	DOR^	—	$S_1 \wedge S_2$	5

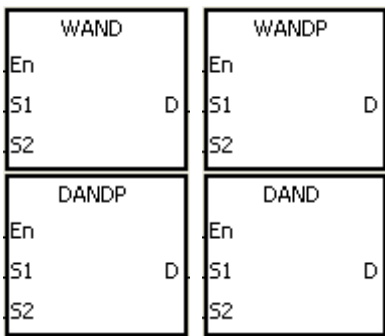
6.9.2 逻辑操作指令说明

API	指令码			操作数								功能					
0800	W D	AND	P	$S_1 \cdot S_2 \cdot D$								逻辑及 (AND) 运算					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



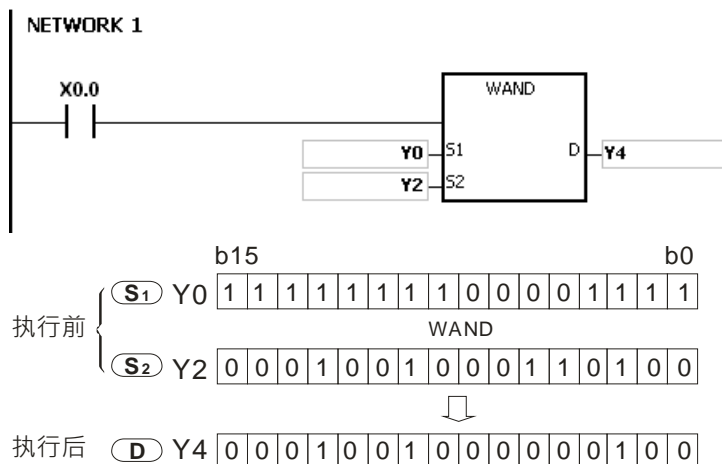
- S_1 : 数据来源装置 1 Word/Double Word
- S_2 : 数据来源装置 2 Word/Double Word
- D : 运算结果 Word/Double Word

指令说明：

- 两个数据源： S_1 及 S_2 作逻辑的“与” (AND) 运算并将结果存于D。
- DAND 才可使用 HC 装置。
- 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。

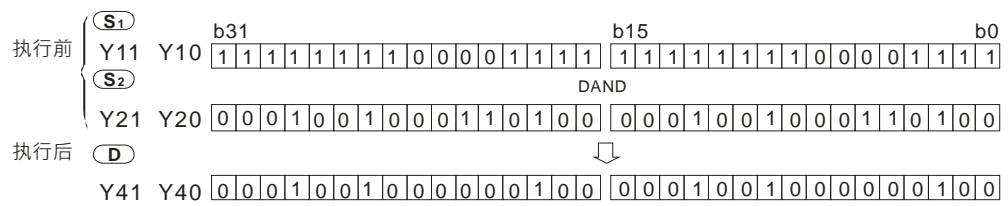
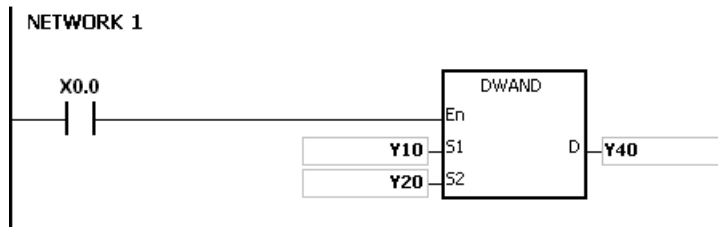
程序范例一：

当 X0.0=ON 时，16 位 Y0 与 Y2 作 WAND，逻辑及 (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 · 逻辑及 (AND) 运算，将结果存于 (Y41、Y40) 中。

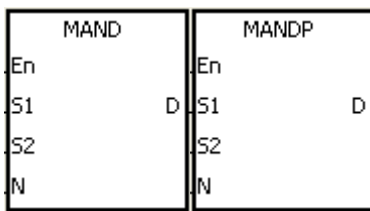


API	指令码		操作数							功能							
0801		MAND	P	$S_1 \cdot S_2 \cdot D \cdot n$							矩阵及 (AND) 运算						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



- S_1 : 矩阵来源装置 1 Word
- S_2 : 矩阵来源装置 2 Word
- D : 运算结果 Word
- n : 数组长度 Word

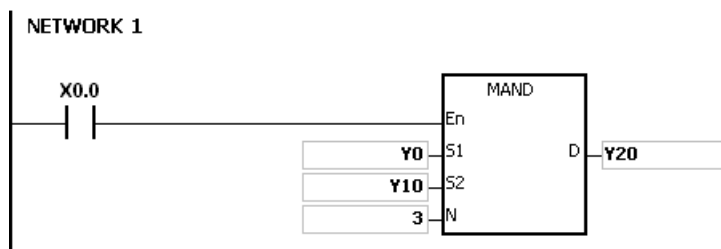
指令说明：

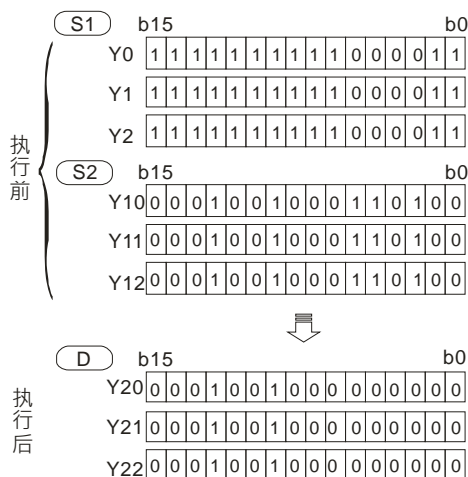
- 两个矩阵来源依数组长度n将 S_1 及 S_2 作矩阵的“与” (AND) 运算并将结果存于D。
- 矩阵的“与” (AND) 运算之规则为两位均为 1 结果始为 1，否则为 0。
- n 操作数的范围值为 1~256。

6

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 矩阵及(AND) 运算，将结果存于 16 位寄存器 Y20~Y22 共 3 列中。

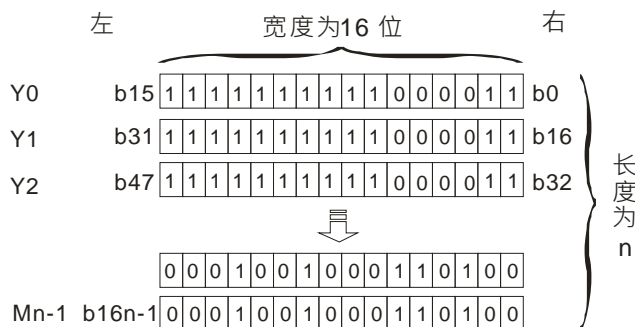




补充说明：

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 个以上连续之 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 $b0 \sim b16n-1$ ）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，为极为方便和重要之应用指令。
- 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 $16n$ 个单点之某个单点当作运算对象，此寄存器称为矩阵之指针 Pr （Pointer），由使用者于指令中指定，其有效范围为 $0 \sim 16n-1$ ，分别对应至矩阵中之位 $b0 \sim b16n-1$ 。
- 矩阵运作中有左、右位移或旋转，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 : 为矩阵之指标，例如 Pr 值为 15 指到 b15 之位。

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

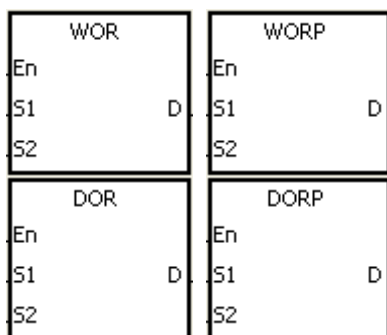
C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X0
0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X2

API	指令码			操作数							功能						
0802	W D	OR	P	S₁ · S₂ · D							逻辑或 (OR) 运算						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



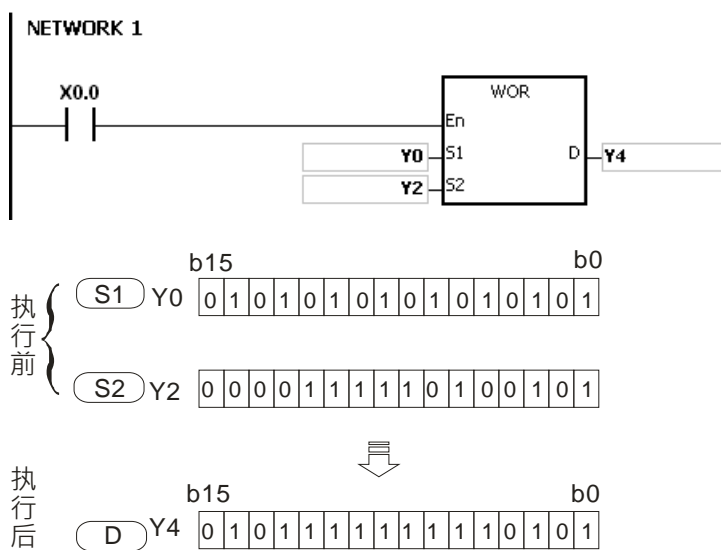
- S₁** : 数据来源装置 1 Word/Double Word
- S₂** : 数据来源装置 2 Word/Double Word
- D** : 运算结果 Word/Double Word

指令说明：

- 两个数据源：**S₁**及**S₂**作逻辑的“或” (OR) 运算结果存于**D**。
- DOR 才可使用 HC 装置。
- 逻辑的“或” (OR) 运算之规则两位有任一为 1 则结果为 1，两者均为 0 结果才为 0。

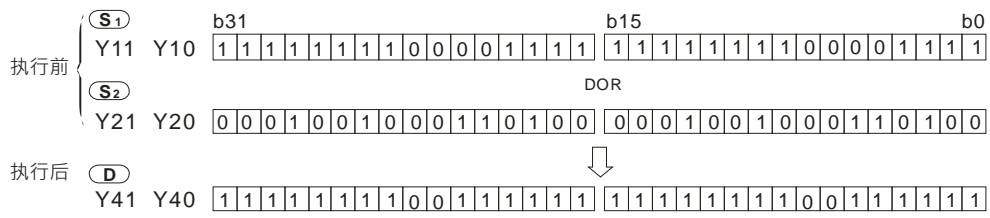
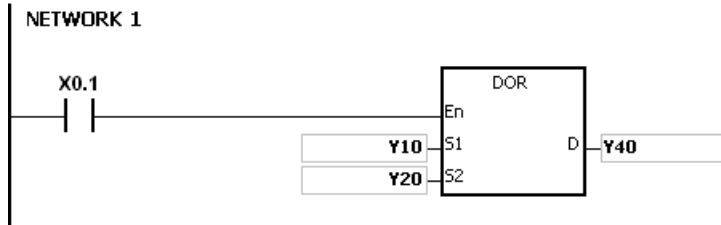
程序范例一：

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。



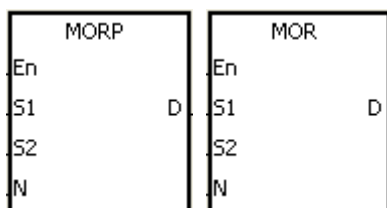
6

API	指令码			操作数								功能				
0803	MOR	P		$S_1 \cdot S_2 \cdot D \cdot n$								矩阵或 (OR) 运算				

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



- S_1 : 矩阵来源装置 1 Word
- S_2 : 矩阵来源装置 2 Word
- D : 运算结果 Word
- n : 数组长度 Word

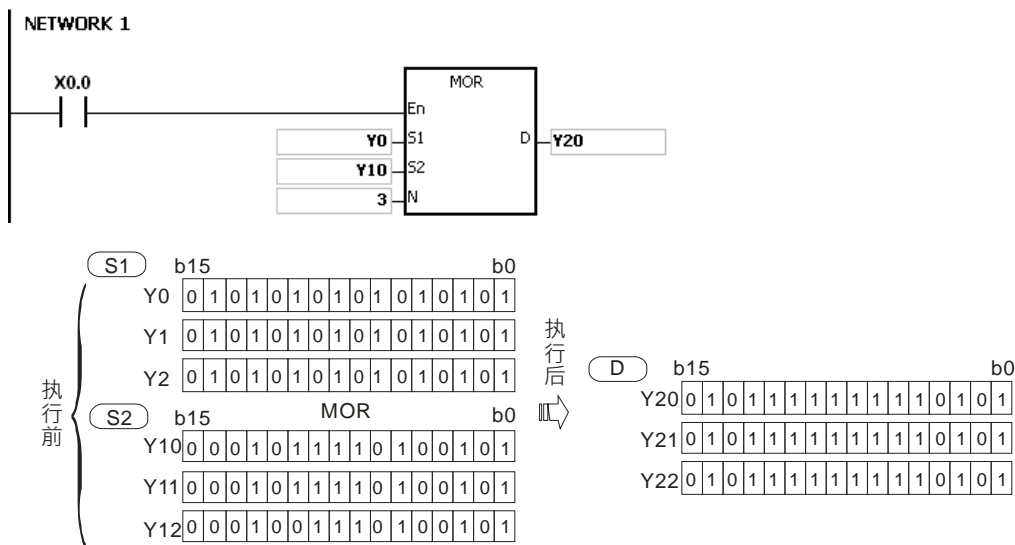
指令说明：

- 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。
- 矩阵的“或”(OR)运算之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。
- 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。

6



补充说明：

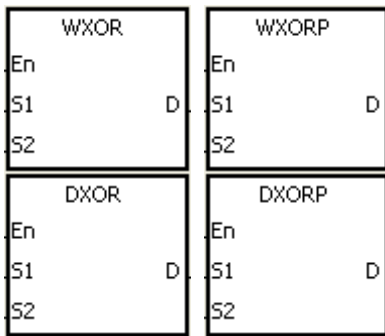
- S_1+n-1 、 S_2+n-1 、 $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码SR0=16#2003。
- $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数						功能					
0804	W D	XOR	P	$S_1 \cdot S_2 \cdot D$						逻辑异或 (XOR) 运算					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



S₁ : 数据来源装置 1 Word/Double Word
 S₂ : 数据来源装置 2 Word/Double Word
 D : 运算结果 Word/Double Word

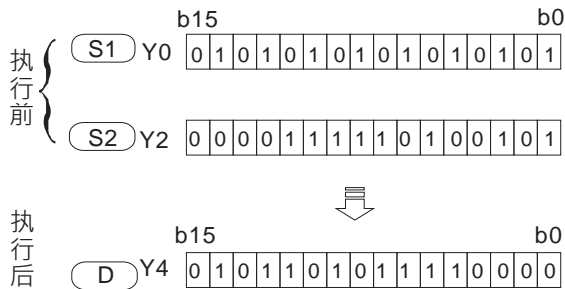
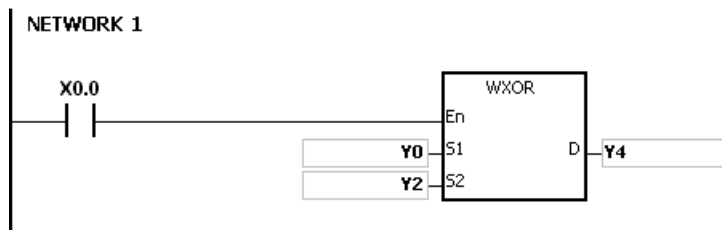
指令说明：

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。

6

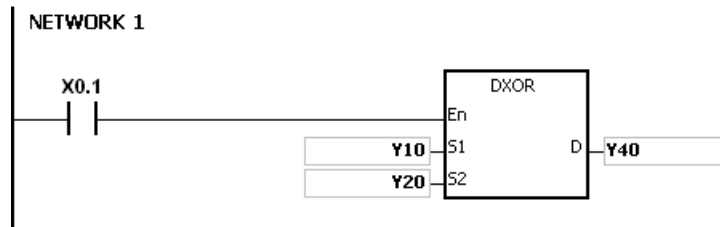
程序范例一：

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。



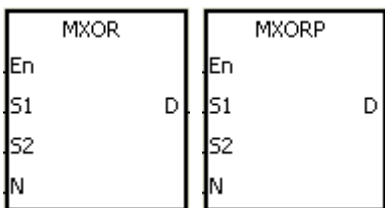
执行前	(S ₁)	Y11	Y10	b31	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	b15	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	b0
	(S ₂)	Y21	Y20		0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	DXOR	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	
执行后	(D)	Y41	Y40		1	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1	↓	1	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1	

API	指令码		操作数							功能						
0805	MXOR	P	$S_1 \cdot S_2 \cdot D \cdot n$							矩阵异或 (XOR) 运算						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



- S_1 : 矩阵来源装置 1 Word
- S_2 : 矩阵来源装置 2 Word
- D : 运算结果 Word
- n : 数组长度 Word

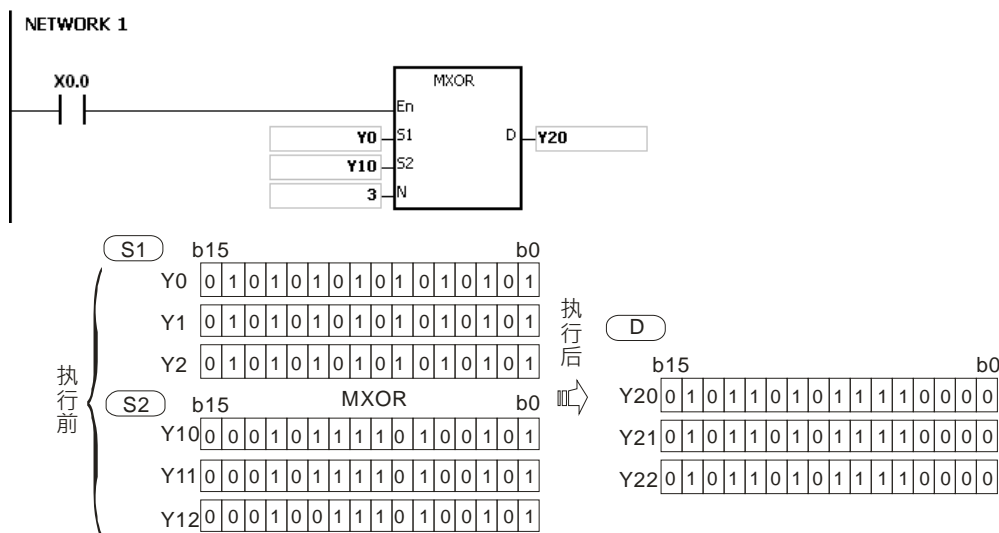
指令说明：

- 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。
- 矩阵的“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。
- n 操作数的范围值为 1~256。

6

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：

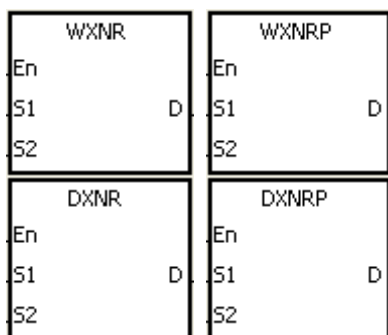
- S_1+n-1 、 S_2+n-1 、 $D+n-1$ 装置超出范围时，指令不执行，SM0=ON，错误码SR0=16#2003。
- $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能					
0806	W D	XNR	P	S₁ · S₂ · D								同或 (XNR) 运算					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



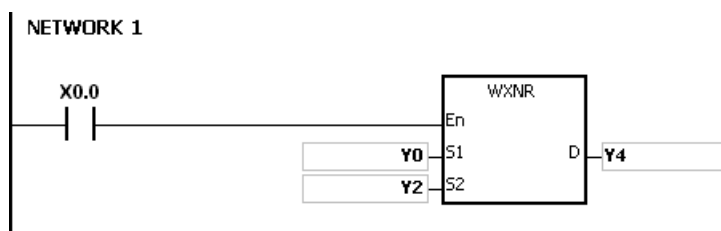
- S₁** : 数据来源装置 1 Word/Double Word
- S₂** : 数据来源装置 2 Word/Double Word
- D** : 运算结果 Word/Double Word

指令说明：

- 两个数据源：**S₁**及**S₂**作逻辑的“同或” (XNR) 运算结果存于**D**。
- DXNR 才可使用 HC 装置。
- 逻辑的“同或” (XNR) 运算之规则为两者相同结果为 1，两者不同结果为 0。

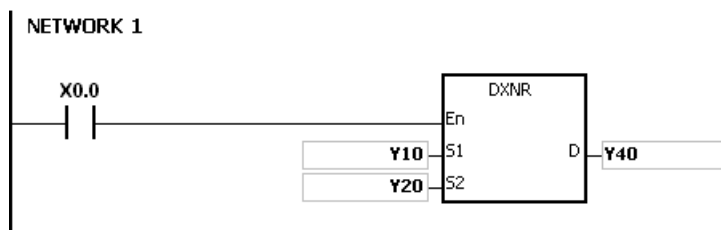
程序范例一：

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。



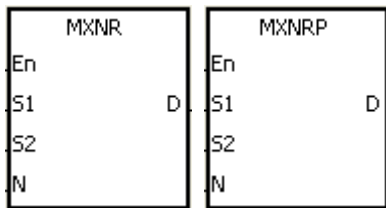
6

API	指令码		操作数							功能							
0807		MXNR	P	$S_1 \cdot S_2 \cdot D \cdot n$							矩阵同或 (XNR) 运算						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



- S_1 : 矩阵来源装置 1 Word
- S_2 : 矩阵来源装置 2 Word
- D : 运算结果 Word
- n : 数组长度 Word

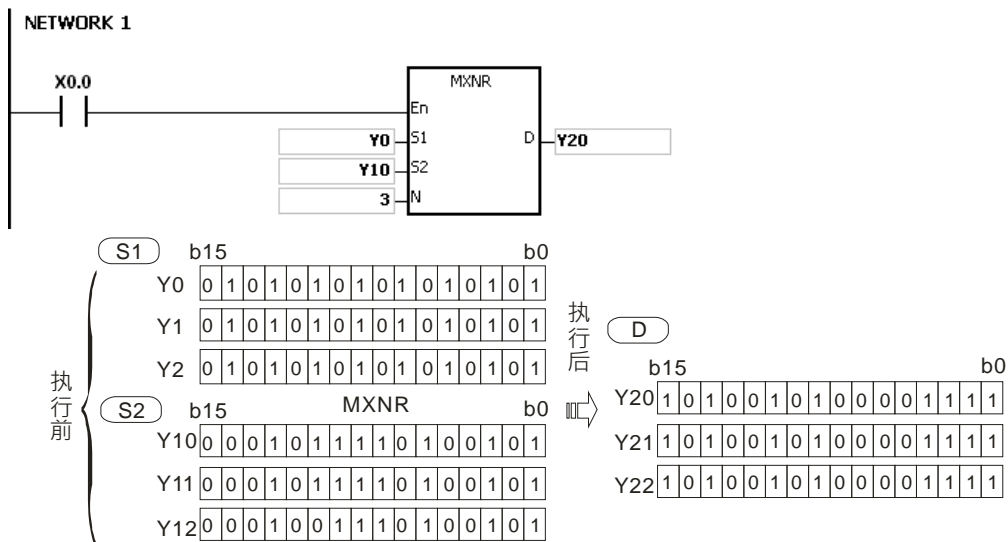
指令说明：

- 两个矩阵来源依数组长度n将 S_1 及 S_2 作矩阵的同或 (XNR) 运算并将结果存于D。
- 矩阵的同或 (XNR) 运算之规则为两者相同结果为 1，两者不同结果为 0。
- n 操作数的范围值为 1~256。

6

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。

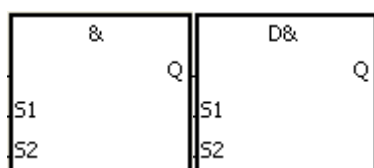


补充说明：

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

API	指令码			操作数								功能					
0809~0811	D	LD #		$S_1 \cdot S_2$								接点类型逻辑运算 LD #					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●				●	●	●	●		●	○	●	○	○		
S_2	●	●				●	●	●	●		●	○	●	○	○		
脉冲执行型		16 位指令 (5 steps)				32 位指令 (5 steps)											
-		AH500				AH500											

符号：



以 LD&跟 DLD&为例

 S_1 : 数据来源装置 1 Word/Double Word S_2 : 数据来源装置 2 Word/Double Word

指令说明：

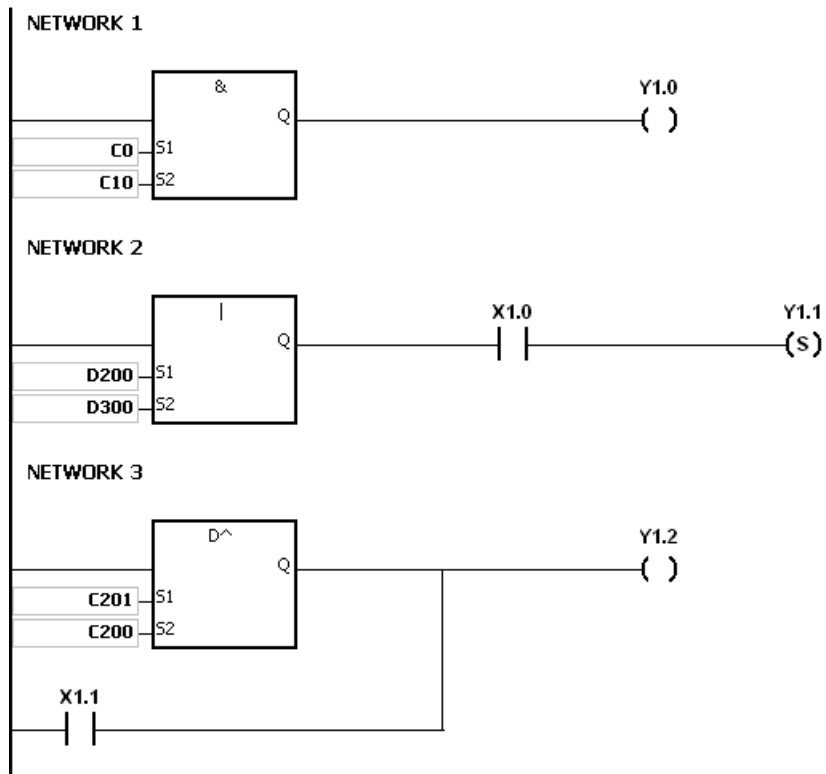
- S_1 与 S_2 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- DLD # 才可使用 HC 装置。
- LD # 的指令可直接与母线连接使用。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
0809	LD&	DLD&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0810	LD	DLD	$S_1 S_2 \neq 0$	$S_1 S_2 = 0$
0811	LD^	DLD^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- & : 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。



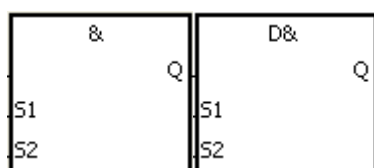
补充说明：

S₁、S₂装置不合法，接点不导通，SM0=ON，错误码SR0=16#2003。

6

API	指令码			操作数							功能						
0812~0814	D	AND #		$S_1 \cdot S_2$							接点类型逻辑运算 AND #						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●				●	●	●	●		●	○	●	○	○		
S_2	●	●				●	●	●	●		●	○	●	○	○		
脉冲执行型										16 位指令 (5 steps)				32 位指令 (5 steps)			
-										AH500				AH500			

符号：



以 AND&跟 DAND&为例

 S_1 : 数据来源装置 1 Word/Double Word S_2 : 数据来源装置 2 Word/Double Word

指令说明：

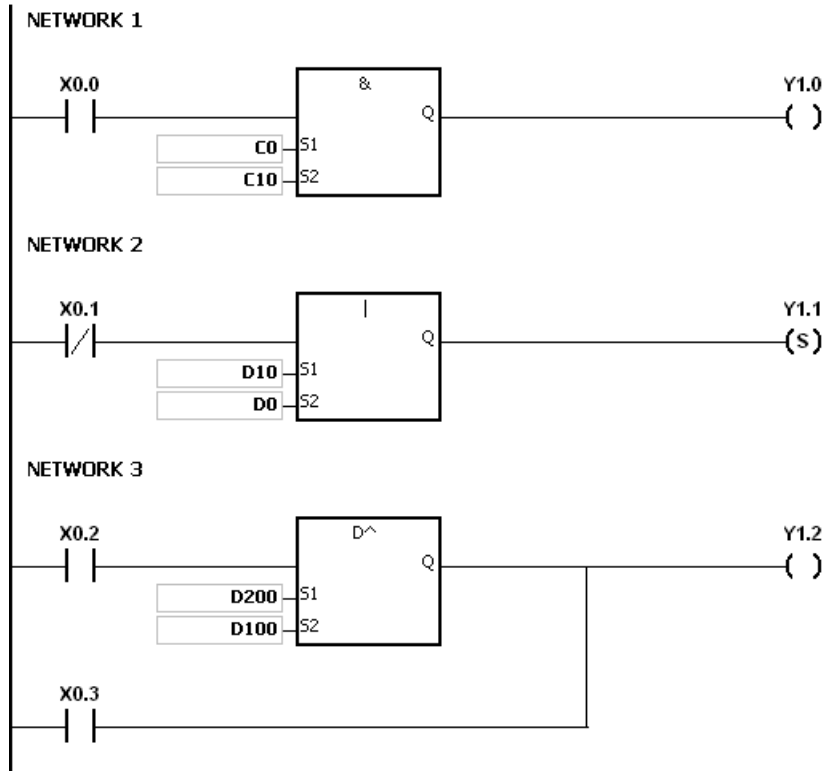
- S_1 与 S_2 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- DAND # 才可使用 HC 装置。
- AND # 的指令是与接点串接的运算指令。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
0812	AND&	DAND&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0813	AND	DAND	$S_1 S \neq 0$	$S_1 S_2 = 0$
0814	AND^	DAND^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S = 0$

- & : 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 X0.0=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。
- 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。
- 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。



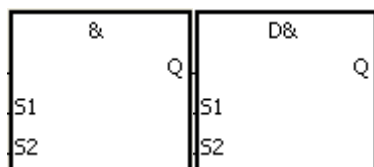
补充说明：

S₁、S₂装置不合法，接点不导通，SM0=ON，错误码SR0=16#2003。

6

API	指令码			操作数								功能					
0815~0817	D	OR #		$S_1 \cdot S_2$								接点类型逻辑运算 OR #					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●				●	●	●	●		●	○	●	○	○		
S_2	●	●				●	●	●	●		●	○	●	○	○		
脉冲执行型		16 位指令 (5 steps)				32 位指令 (5 steps)											
-		AH500				AH500											

符号：



以 OR&跟 DOR&为例

 S_1 : 数据来源装置 1 Word/Double Word S_2 : 数据来源装置 2 Word/Double Word

指令说明：

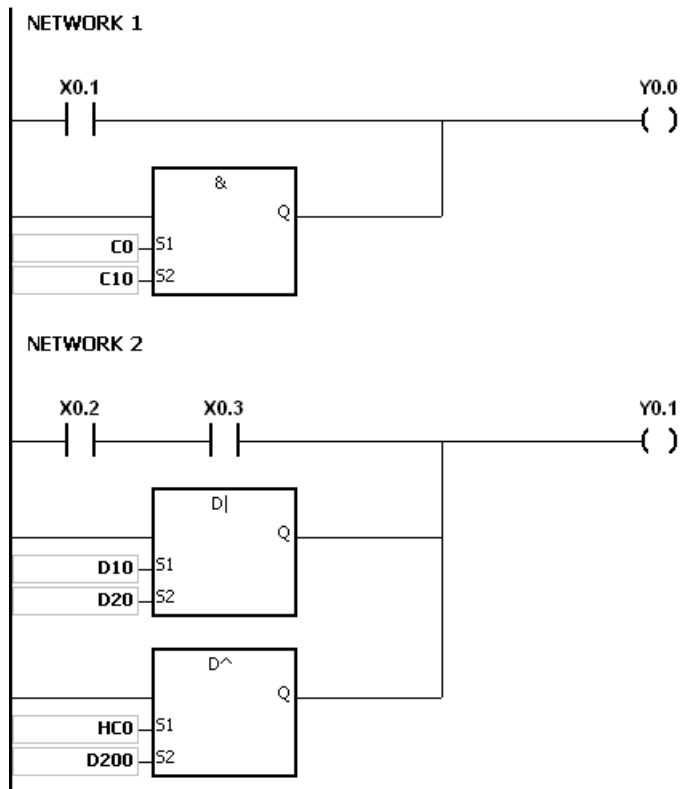
- S_1 与 S_2 之内容作比较的指令，比较结果不为 0 时，该指令导通，比较结果为 0 时，该指令不导通。
- DOR # 才可使用 HC 装置。
- OR # 的指令是与接点并接的运算指令。

API No.	16-bit 指令	32-bit 指令	导通条件	非导通条件
0815	OR&	DOR&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0816	OR	DOR	$S_1 S_2 \neq 0$	$S_1 S_2 = 0$
0817	OR^	DOR^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- & : 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。

6

6.10 旋转指令

6.10.1 旋转指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>0900</u>	ROR	DROR	✓	右旋转	5
<u>0901</u>	RCR	DRCR	✓	附进位标志右旋转	5
<u>0902</u>	ROL	DROL	✓	左旋转	5
<u>0903</u>	RCL	DRCL	✓	附进位标志左旋转	5
<u>0904</u>	MBR	—	✓	矩阵位旋转	7

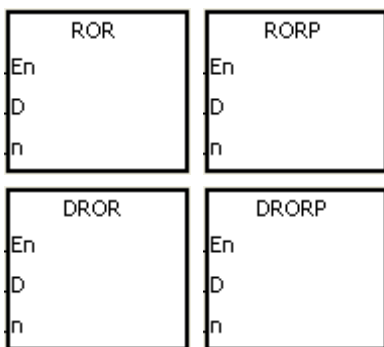
6.10.2 旋转指令说明

API	指令码			操作数							功能						
0900	D	ROR	P	D · n							右旋转						

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

脉冲执行型	16 位指令(5 steps)	32 位指令(5 steps)
AH500	AH500	AH500

符号：



D : 欲旋转之装置 Word/Double Word
n : 一次旋转之位数 Word/Double Word

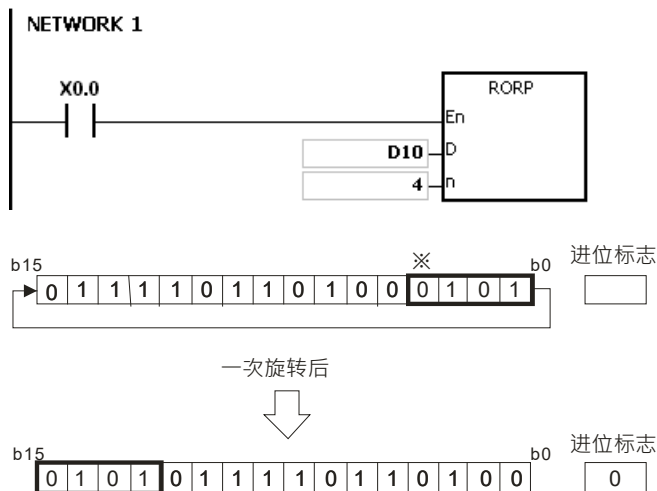
指令说明：

6

1. 将 **D** 所指定的装置内容一次向右旋转 **n** 个位。
2. **DROR** 才可以使使用 **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 个位为一组往右旋转 · 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：

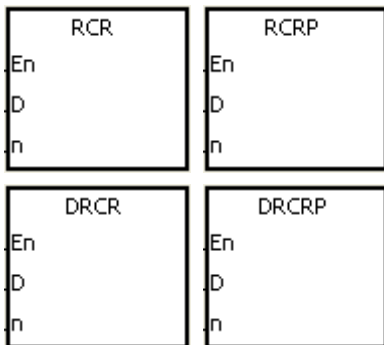
1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能				
0901	D	RCR	P	D · n								附进位标志右旋转				

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

脉冲执行型	16 位指令(5 steps)	32 位指令(5 steps)
AH500	AH500	AH500

符号：



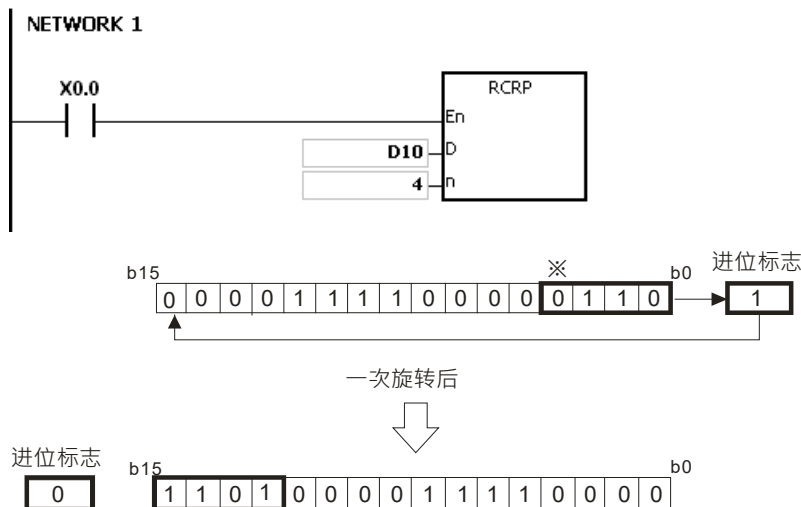
D : 欲旋转的装置 Word/Double Word
n : 一次旋转的位数 Word/Double Word

指令说明：

1. 将 **D** 所指定的装置内容连同进位标志 SM602，一次向右旋转 **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 个位为一组往右旋转，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：

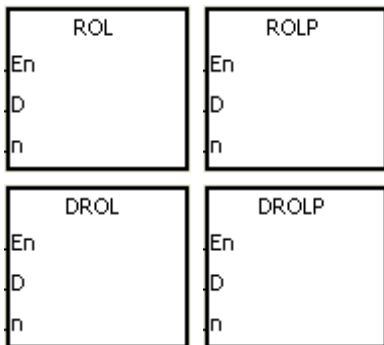
1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能				
0902	D	ROL	P	D · n								左旋转				

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

脉冲执行型	16 位指令(5 steps)	32 位指令(5 steps)
AH500	AH500	AH500

符号：



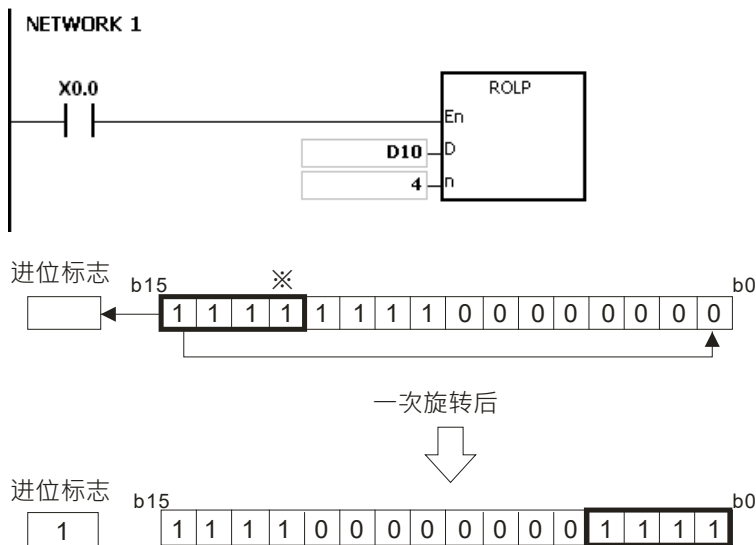
D : 欲旋转的装置 Word/Double Word
n : 一次旋转的位数 Word/Double Word

指令说明：

1. 将 **D** 所指定的装置内容一次向左旋转 **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 个位一组往左旋转 · 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：

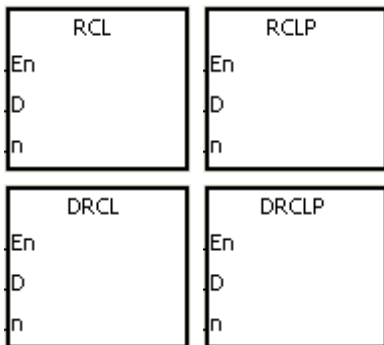
1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能				
0903	D	RCL	P	D · n								附进位标志左旋转				

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

脉冲执行型	16 位指令(5 steps)	32 位指令(5 steps)
AH500	AH500	AH500

符号：



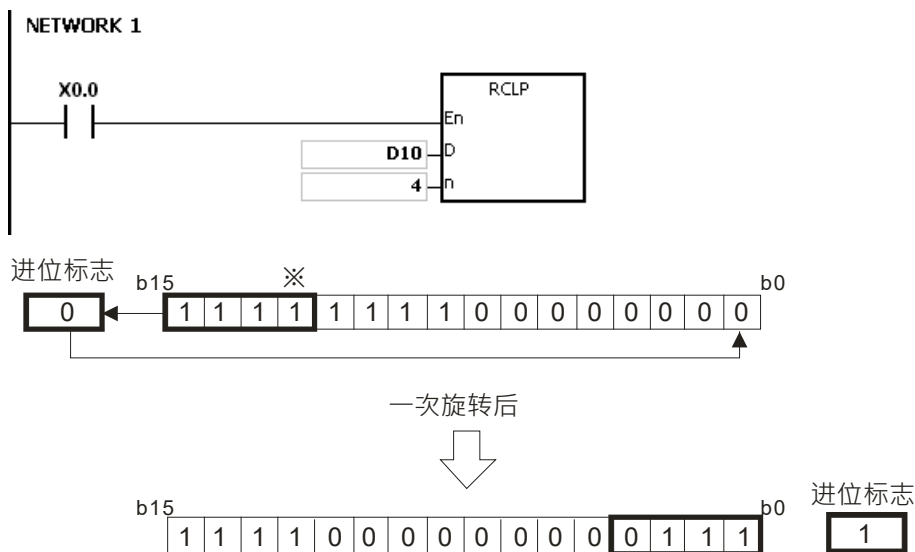
D : 欲旋转的装置 Word/Double Word
n : 一次旋转的位数 Word/Double Word

指令说明：

1. 将 **D** 所指定的装置内容连同进位标志 **SM602**，一次向左旋转 **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 个位一组往左旋转，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：

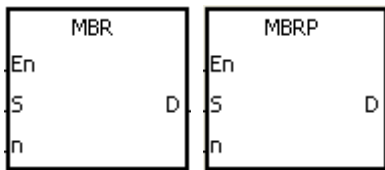
1. 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数							功能				
0904		MBR	P	S · D · n							矩阵位旋转				

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

脉冲执行型	16 位指令(7 steps)	32 位指令
AH500	AH500	-

符号：



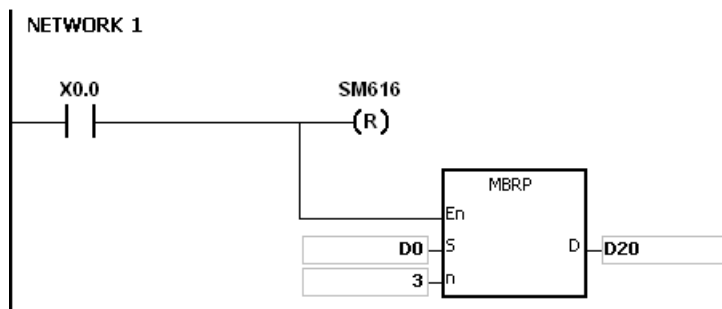
- S** : 矩阵来源装置 Word
- D** : 运算结果 Word
- n** : 数组长度 Word

指令说明：

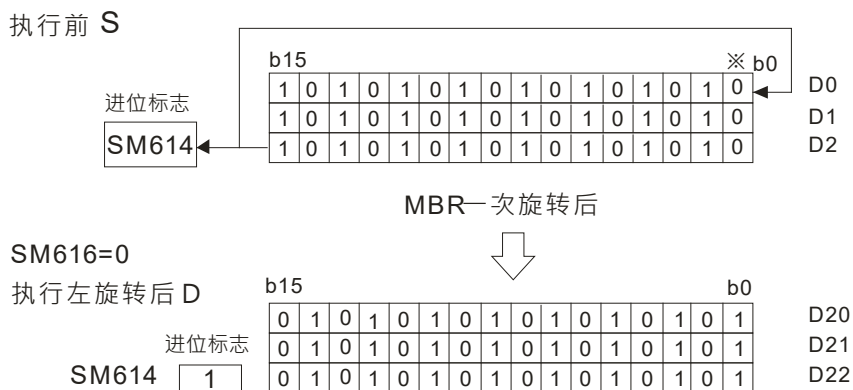
1. 矩阵来源依数组长度 **n** 将 **S** 矩阵位做左右旋转控制，SM616=0 决定矩阵位左旋转，SM616=1 决定矩阵位右旋转。每次旋转一位，因旋转造成之空位（左旋时为 b0，右旋时为 b16n-1）由旋出位（左旋时为 b16n-1 右旋时为 b0）状态填补。将结果存入 **D**。旋出位不但用以填补前述之空位，同时并将之状态送到进位标志 SM614。
2. **n** 操作数值的范围为 1~256。
3. 本指令一般都是使用脉冲执行型指令（MBRP）。

程序范例一：

当 X0.0=ON 时 SM616=OFF 决定矩阵为左旋转，16 位寄存器 D0~D2 矩阵旋转，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。

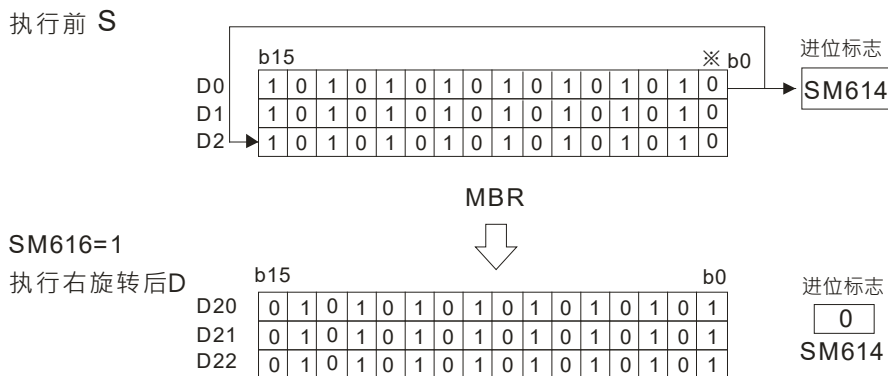
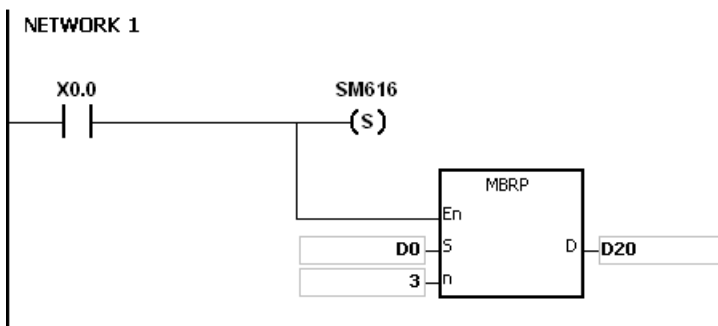


6



程序范例二：

当 X0.0=ON 时，SM616=ON 作矩阵右旋转，16 位寄存器 D0~D2 矩阵作右旋转，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。



6

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： 矩阵旋转位移输出进位标志

SM616： 矩阵旋转位移方向标志

6.11 基本指令

6.11.1 基本指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1000</u>	RST	—	—	接点或寄存器清除	3
<u>1001</u>	TMR	—	—	16 位定时器	5
<u>1002</u>	TMRH	—	—	16 位定时器	5
<u>1003</u>	CNT	—	—	16 位计数器	5
<u>1004</u>	—	DCNT	—	32 位计数器	5

6.11.2 基本指令说明

API	指令码			操作数									功能				
1000		RST		D									接点或寄存器清除				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●	●	●	●	●	●	●	●	○	●				
脉冲执行型		16 位指令 (3 steps)			32 位指令												
-		AH500			-												

符号：

元件 **D** : 清除元件编号 Bit/Word
 —(R)

指令说明：

1. 当 RST 指令被驱动，其指定的元件的动作如下：

元件	状态
Bit	线圈及接点都会被设定为 OFF。
T · C · HC	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，其指定元件的状态保持不变。
3. 本指令有支持 DY 装置。

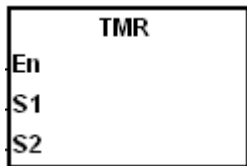
程序范例：

当 X0.0=ON 时，Y0.5 会被设为 OFF。



API	指令码			操作数								功能							
1001		TMR		S1 · S2								16 位定时器							
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF		
S1					●														
S2	●	●						●	●		●		●	○	○				
											脉冲执行型			16 位指令 (5 steps)			32 位指令		
											-			AH500			-		

符号：



S₁ : 定时器编号

Word

S₂ : 定时器设定值

Word

指令说明：

详细内容说明请参考 API1002 TMRH 指令的指令说明。

API	指令码			操作数								功能					
1002	TMRH			S ₁ · S ₂								16 位定时器					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁					○												
S ₂	○	○						○	○		○		○	○	○		
							脉冲执行型				16 位指令 (5 steps)			32 位指令			
							-				AH500			-			

符号：



S₁ : 定时器编号 Word

S₂ : 定时器设定值 Word

指令说明：

1. TMR 指令所指定之 T 定时器以 100ms 为单位计时，TMRH 指令所指定之 T 定时器以 1ms 为单位计时。
2. 子程序专用 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 变成累进定时器，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。

6



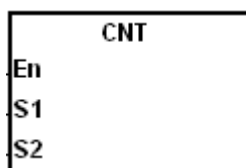
补充说明：

S1 操作数，若使用 ISPSOft 宣告，则数据类型为 TIMER。

API	指令码			操作数								功能					
1003	CNT			S ₁ · S ₂								16 位计数器					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁						○											
S ₂	○	○						○	○		○		○	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
-	AH500	-

符号：



S₁ : 计数器编号 Word

S₂ : 计数器设定值 Word

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。

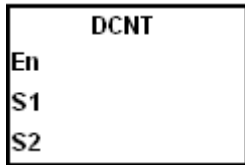


补充说明：

S1 操作数，若使用 ISPSOft 宣告，则数据类型为 COUNTER。

API	指令码				操作数								功能				
1004		DCNT			$S_1 \cdot S_2$								32 位计数器				
装置	X	Y	M	S	T	C	HC	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 (5 steps)		
											-		-		AH500		

符号：



S_1 : 计数器编号

Double Word

S_2 : 计数器设定值

Double Word

指令说明：

- DCNT 为 32 位计数器 HC0 至 HC63 之启动指令。
- 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：

6

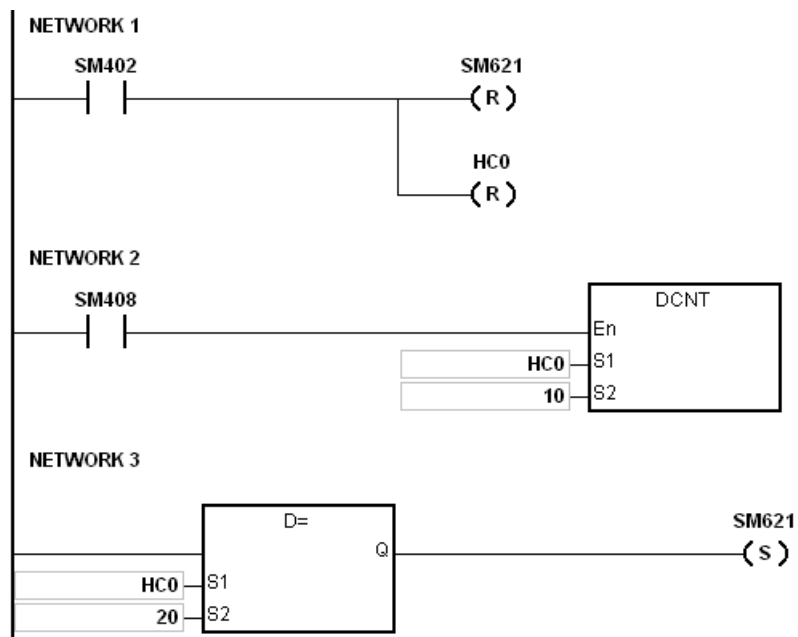
当 PLC RUN 时，定时器 HC0 被清除，且 SM621=OFF 将执行上数，此时 SM408 的状态为第一次 ON，所以计数器 HC0 会加载设定值 10 并开始计数。

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。

6.12 位移指令

6.12.1 位移指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
1100	SFTR	–	✓	位右移	9
1101	SFTL	–	✓	位左移	9
1102	WSFR	–	✓	寄存器右移	9
1103	WSFL	–	✓	寄存器左移	9
1104	SFWR	–	✓	位移写入	7
1105	SFRD	–	✓	位移读出	7
1106	SFPO	–	✓	读出数据串行最新数据	5
1107	SFDEL	–	✓	删除数据串行中的数据	7
1108	SFINS	–	✓	插入数据到数据串行中	7
1109	MBS	–	✓	矩阵位位移	7
1110	SFR	–	✓	16 位寄存器位右移	5
1111	SFL	–	✓	16 位寄存器位左移	5
1112	BSFR	–	✓	n 个位右移 1 个位	5
1113	BSFL	–	✓	n 个位左移 1 个位	5
1114	NSFR	–	✓	n 个寄存器右移	5
1115	NSFL	–	✓	n 个寄存器左移	5

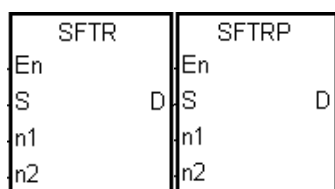
6.12.2 位移指令说明

API	指令码			操作数								功能					
1100		SFTR	P	S · D · n ₁ · n ₂								位右移					

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



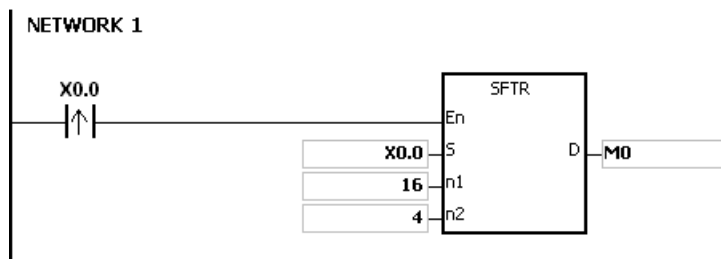
- S** : 位移装置的起始编号 Bit
- D** : 欲位移装置的起始编号 Bit
- n₁** : 欲位移的数据长度 Word
- n₂** : 一次位移的位数 Word

指令说明：

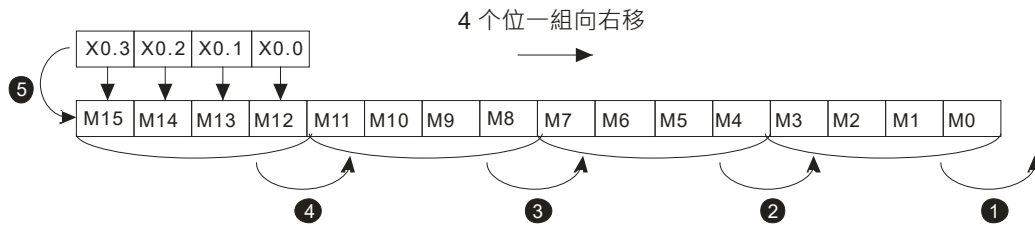
- 将 **D** 开始之起始编号，具有 **n₁** 个数字符 (位移寄存器长度) 的位装置，以 **n₂** 位个数来右移。而 **S** 开始起始编号以 **n₂** 位个数移入 **D** 中来填借位空位。
- 本指令一般都是使用脉冲执行型指令 (SFTRP)。
- n₁** 值的范围为 1~1024，**n₂** 值的范围为 1~**n₁**。

程序范例：

- 在 X0.0 上升沿时，由 M0~M15 组成 16 位，以 4 位作右移。
- 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

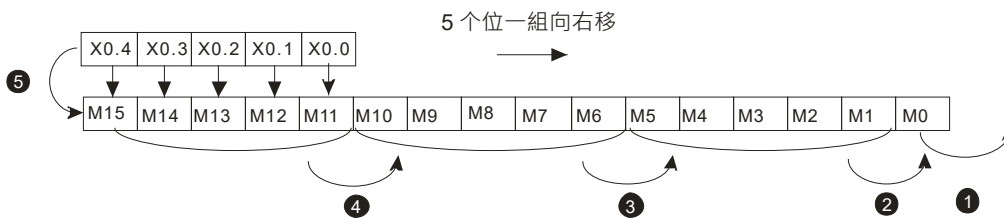
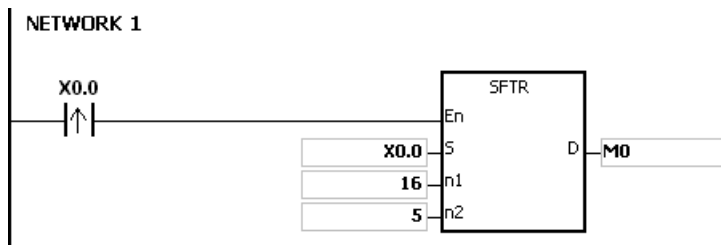


6



1. 在 X0.0 上升沿时，由 M0~M15 组成 16 位，以 5 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 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



6

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。

API	指令码		操作数										功能			
1101	SFTL	P	S · D · n ₁ · n ₂										位左移			

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：

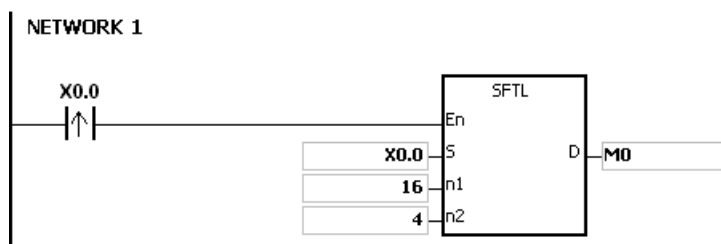
SFTL	SFTLP	S	Bit
En	En	D	Bit
S	S	n ₁	Word
n ₁	n ₁	n ₂	Word
n ₂	n ₂		

指令说明：

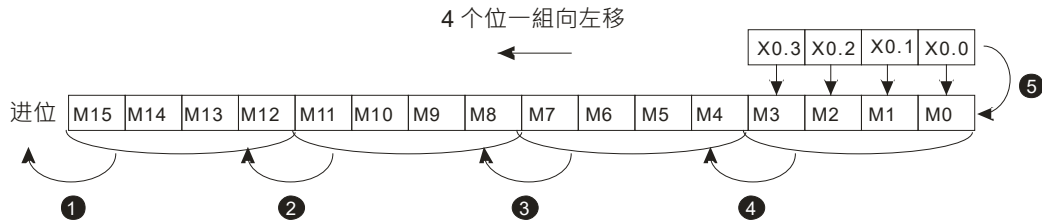
1. 将 D 开始之起始编号，具有 n₁ 个数字符 (位移寄存器长度) 的位装置，以 n₂ 位个数来左移。而 S 开始起始编号以 n₂ 位个数移入 D 中来填借位空位。
2. 本指令一般都是使用脉冲执行型指令 (SFTLP)。
3. n₁ 值的范围为 1~1024，n₂ 值的范围为 1~n₁。

程序范例：

1. 在 X0.0 上升沿时，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

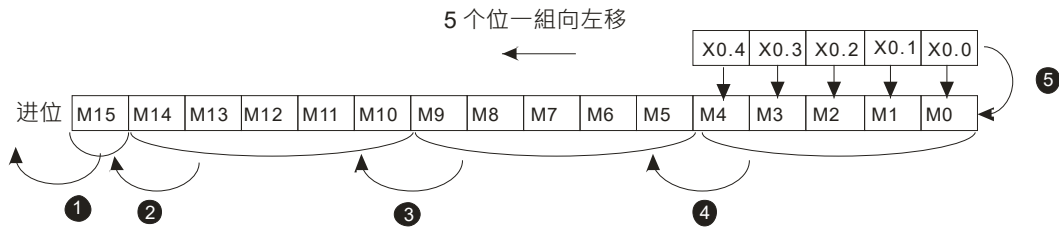
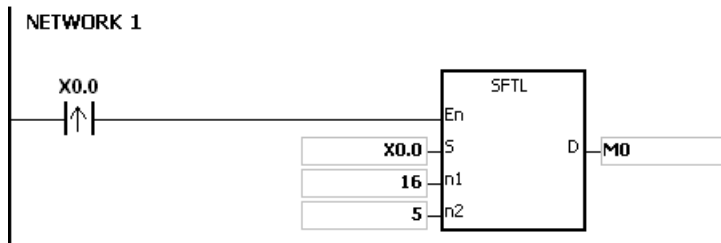


6



3. 在 X0.0 上升沿时，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。

- ① M15 → 进位
- ② M10 → M15
- ③ M9~M5 → M14~M10
- ④ M4~M0 → M9~M5
- ⑤ X0.4~X0.0 → M4~M0 完成



6

补充说明：

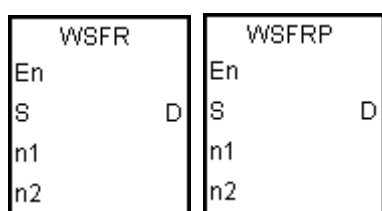
1. $S+n_2-1$ 、 $D+n_1-1$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 1024$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码		操作数				功能						
1102	WSFR	P	S · D · n ₁ · n ₂				寄存器右移						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



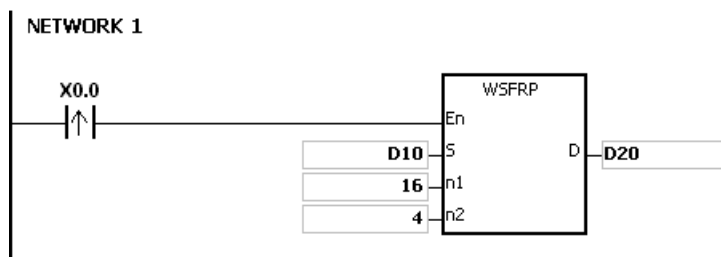
- S** : 位移装置的起始编号 Word
- D** : 欲位移装置的起始编号 Word
- n₁** : 欲位移的数据长度 Word
- n₂** : 一次位移的位数 Word

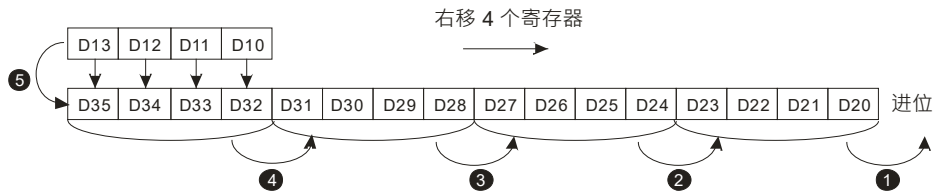
指令说明：

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** 个字符来右移。而 **S** 开始起始编号以 **n₂** 字符个数移入 **D** 中来填补字符空位。
2. 本指令一般都是使用脉冲执行型指令 (WSFRP)。
3. **n₁** 值的范围为 1~512，**n₂** 值的范围为 1~**n₁**。

程序范例一：

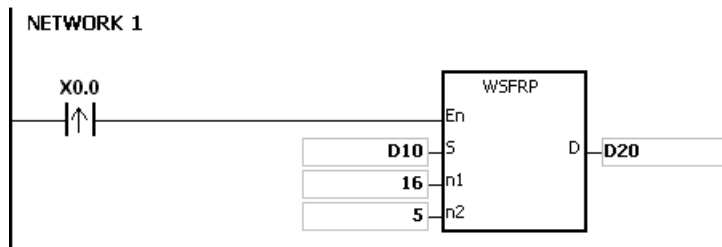
1. X0.0=OFF→ON 时，由 D20~D35 所组成之 16 个寄存器数据串行为位移区域，以 4 个寄存器来右移。
2. 扫描一次的字符右移动作依照下列编号 1~5 动作。
 - ① D23~D20 → 进位
 - ② D27~D24 → D23~D20
 - ③ D31~D28 → D27~D24
 - ④ D35~D32 → D31~D28
 - ⑤ D13~D10 → D35~D32 完成



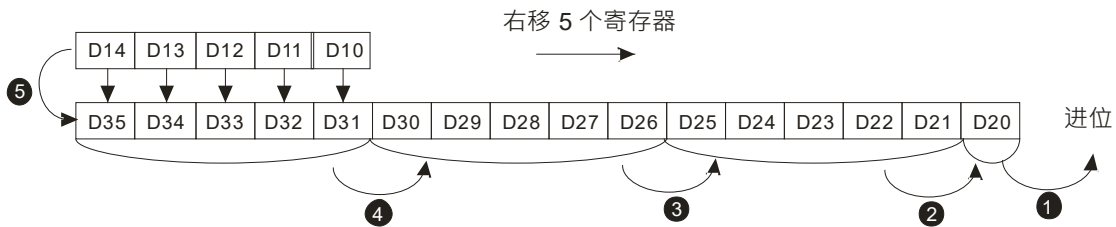


程序范例二：

1. X0.0=OFF→ON 时，由 D20~D35 所组成之 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



6



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。

API	指令码			操作数								功能				
1103	WSFL	P		S · D · n ₁ · n ₂								寄存器左移				

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



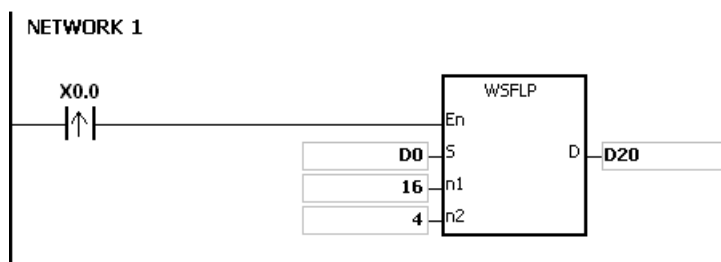
- S** : 位移装置的起始编号 Word
- D** : 欲位移装置的起始编号 Word
- n₁** : 欲位移的数据长度 Word
- n₂** : 一次位移的位数 Word

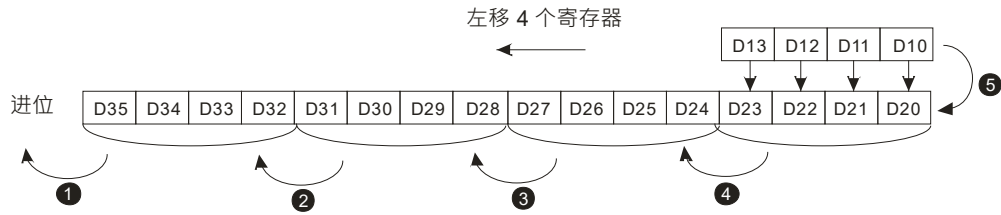
指令说明：

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** 中来填补字符空位。
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 上升沿时，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

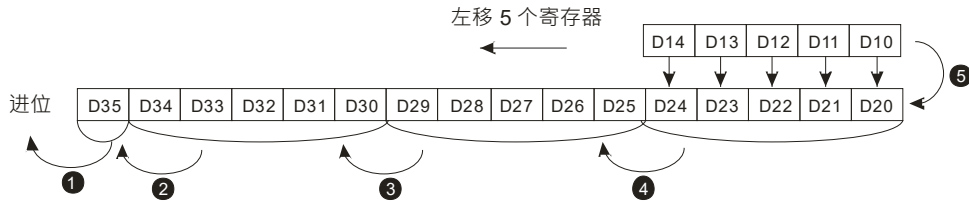
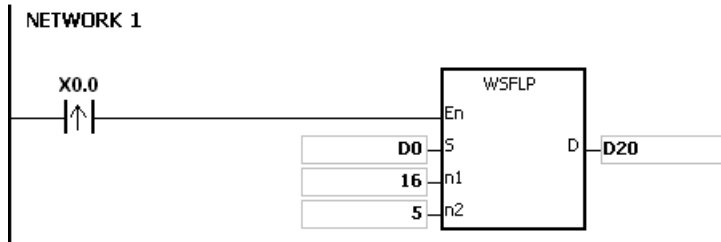




3. 在 X0.0 上升沿时，由 D20~D35 组成 16 位，以 5 位作左移。

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



6

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。

API	指令码			操作数								功能				
1104		SFWR	P	S · D · n								位移写入				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：

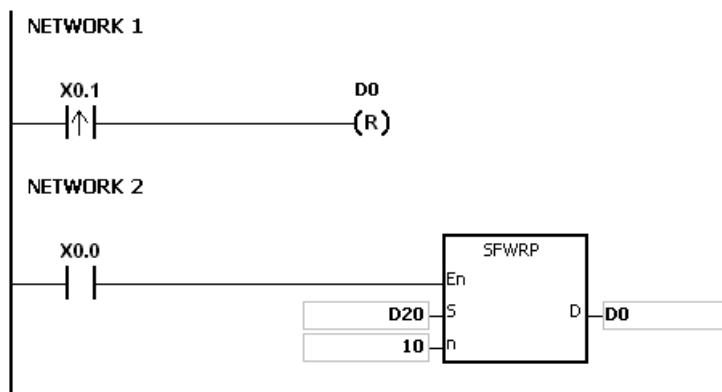


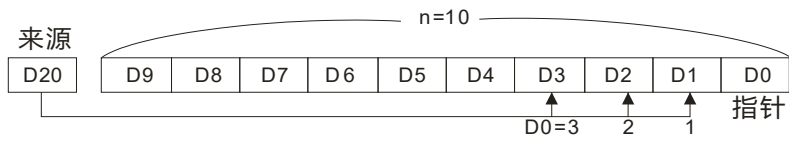
指令说明：

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：

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 API1104 SFWR 与 API1105 SFRD 可搭配使用，执行先入先出数据串行的写入读出控制。

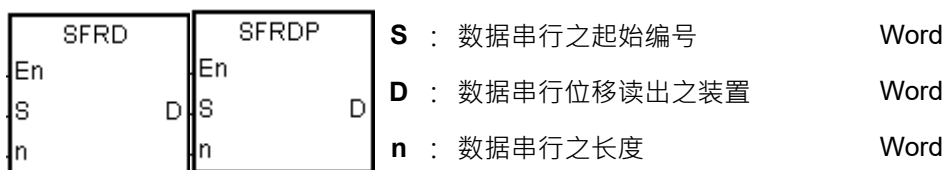
6

API	指令码			操作数								功能				
1105	SFRD	P		S · D · n								位移读出				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：

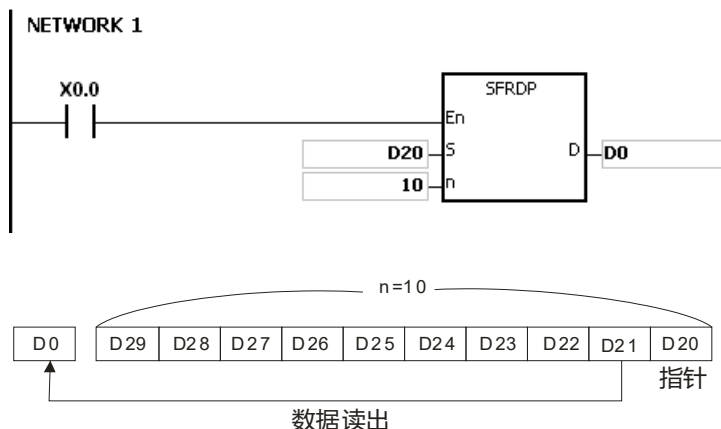


指令说明：

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：

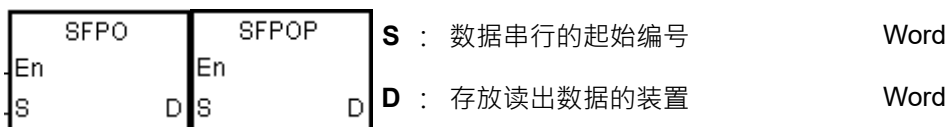
1. **S**<0，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **S+n-1** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. **n**<2 或 **n**>512，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 本指令 API1104 SFWR 与 API1105 SFRD 可搭配使用，执行先入先出数据串行的写入读出控制。

6

API	指令码			操作数								功能					
1106		SFPO	P	S · D								读出数据串行最新数据					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

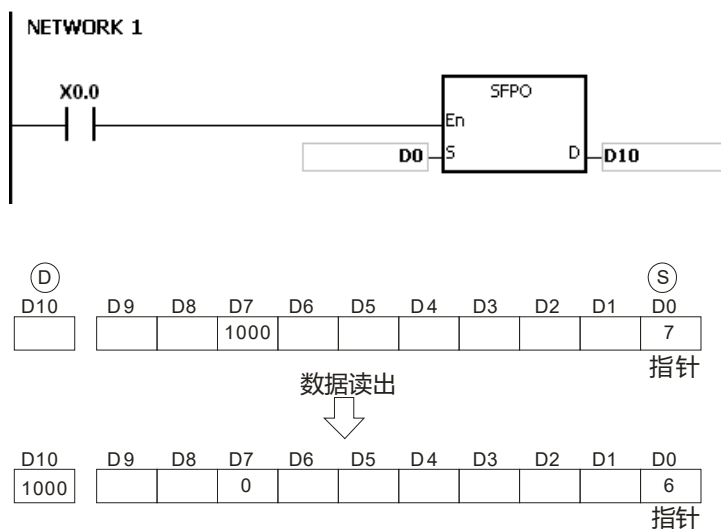


指令说明：

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。



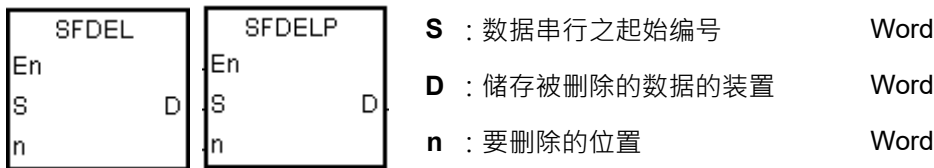
补充说明：

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。

API	指令码			操作数								功能					
1107		SFDEL	P	S · D · n								删除数据串行中的数据					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：

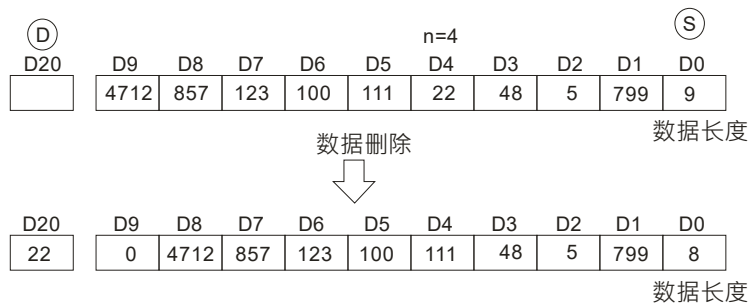
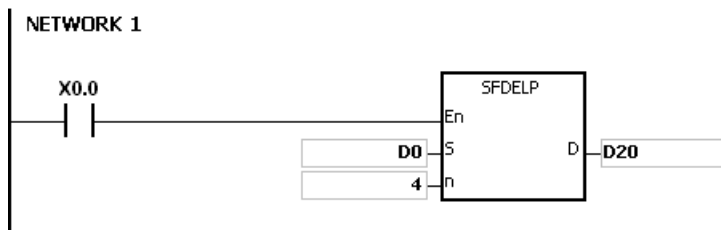


指令说明：

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：

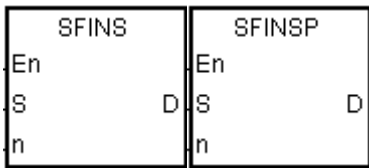
1. **S** 内容值 <0 ，指令不执行，**SM0=ON**，错误码 **SR0=16#2003**。
2. **S+n** 装置超出范围时，指令不执行，**SM0=ON**，错误码 **SR0=16#2003**。
3. **S+S** 内容值，装置超出范围时，指令不执行，**SM0=ON**，错误码 **SR0=16#2003**。
4. **n>S** 内容值，指令不执行，**SM0=ON**，错误码 **SR0=16#200B**。
5. **n \leq 0** 时，指令不执行，**SM0=ON**，错误码 **SR0=16#200B**。

API	指令码			操作数							功能						
1108		SFINS	P	S · D · n							插入数据到数据串行中						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



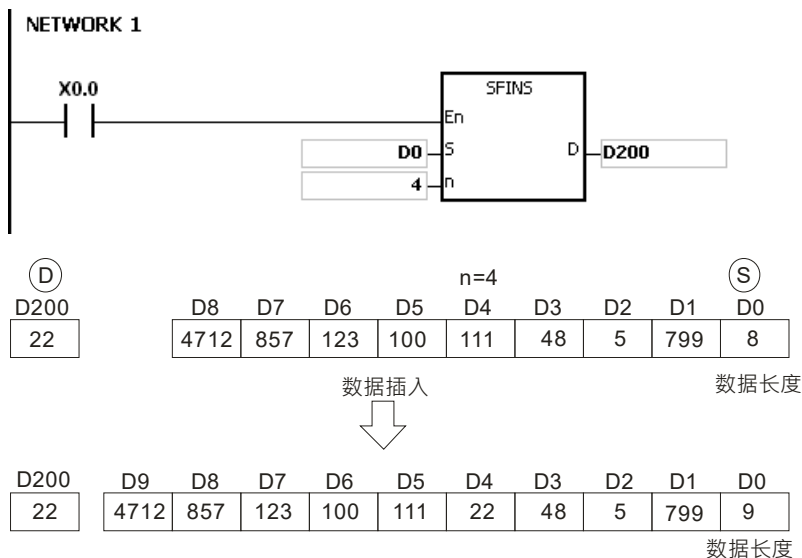
- S** : 数据串行的起始编号 Word
- D** : 要插入数据串行的数据 Word
- n** : 要插入的位置 Word

指令说明：

- 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 。
- 本指令一般都是使用脉冲执行型指令 (SFINSP) 。
- 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：

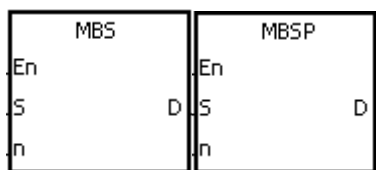
1. $S \leq 0$ ，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. $S+n$ 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. $S+S$ 内容值+1，装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. $n > S$ 内容值，指令不执行，SM0=ON，错误码 SR0=16#200B。
5. $n \leq 0$ 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码		操作数				功能					
1109	MBS	P	S · D · n				矩阵位移					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



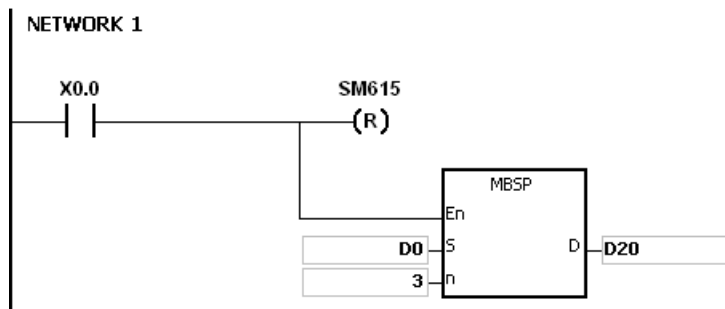
- S**：矩阵来源装置 Word
- D**：运算结果 Word
- n**：数组长度 Word

指令说明：

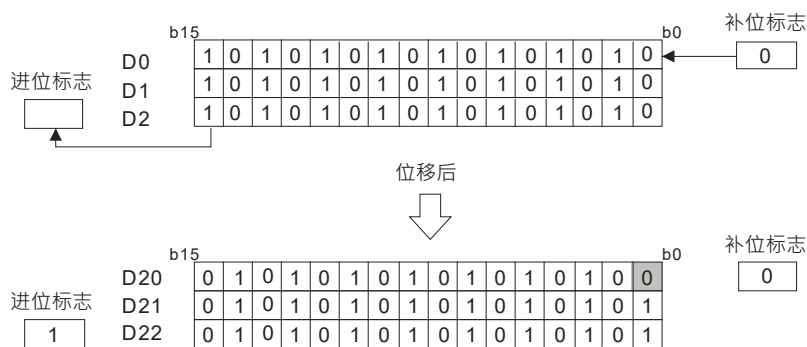
1. 矩阵来源依数组长度 **n** 将 **S** 矩阵位做左右位移控制，SM616=0 决定矩阵位左移，SM616=1 决定矩阵位右移。每次移动一位，因位移而腾出之空位（左移时为 b0，右移时为 b16n-1）则以借位标志 SM615 之状态填补。而因位移而挤出之位（左移时为 b16n-1，右移时为 b0）状态则送到进位标志 SM614 去，然后将结果存入 **D**。
2. n 操作数值的范围为 1~256。
3. 本指令一般都是使用脉冲执行型指令（MBSP）。

程序范例一：

当 X0.0=ON 时，矩阵旋转位移方向标志 SM616=OFF 作矩阵左移。设矩阵位移输入借位标志 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，矩阵旋转位移输出进位标志 SM614=1。

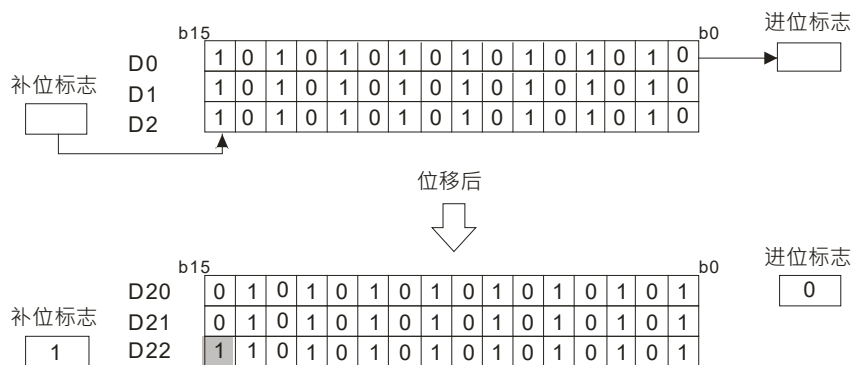
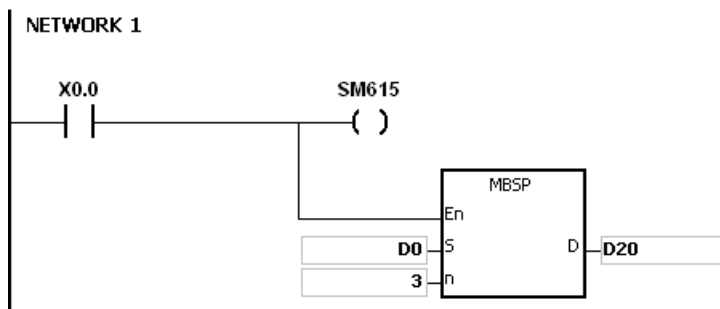


6



程序范例二：

当 X0.0=ON 时 矩阵旋转位移方向标志 SM616=ON 作矩阵右移 设矩阵位移输入借位标志 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 矩阵旋转位移输出进位标志 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

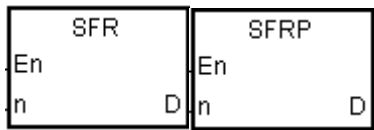
- SM614： 矩阵旋转位移输出进位标志
- SM615： 矩阵位移输入借位标志
- SM616： 矩阵旋转位移方向标志

API	指令码			操作数								功能					
1110		SFR	P	D · n								16 位寄存器位右移					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



D：欲位移之装置 Word
n：一次位移之位數 Word

指令说明：

1. 将 **D** 往右移 **n** 个位数，而因位移而腾出之空位 ($b_{15} \sim b_{15-n+1}$) 则补零。因位移而挤出之第 **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**)。

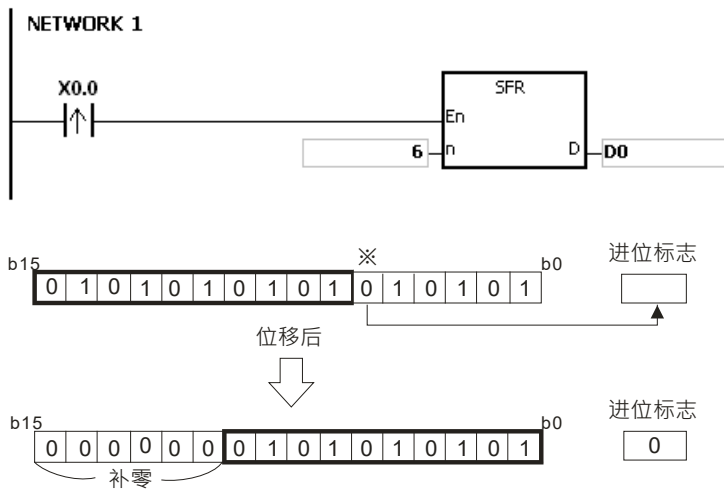
程序范例：

当 X0.0=ON 时，D0 的 b0~b15，往右移动 6 个位数，而 b5 的状态传送到 **SM602**，位移后，b10~b15 内容清除为 0。

6

扫描一次的位右移动作依照下列编号 1~3 动作。

- ① b5~b0 → 进位 (**SM602**=b5)
- ② b15~b6 → b9~b0
- ③ 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。

API	指令码			操作数								功能					
1111		SFL	P	D · n								16 位寄存器位左移					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



D : 欲位移之装置 Word
n : 一次位移之位数 Word

指令说明：

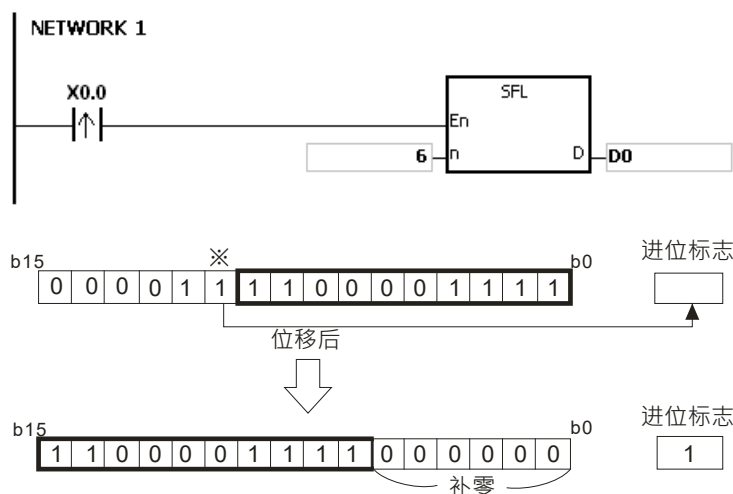
1. 将 D 往左移 n 个数，而因位移而腾出之空位 (b₀~b_{n-1}) 则补零，因位移而挤出之第 16-n 位 (b_{16-n}) 状态则送到进位标志 SM602 去。
2. n 内容值的范围为 1~16。
3. 本指令一般都是使用脉冲执行型指令 (SFLP)。

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。

- ① b₁₅~b₁₀ → 进位 (SM602=b₁₀)
- ② b₉~b₀ → b₁₅~b₆
- ③ 0 → b₅~b₀ 完成



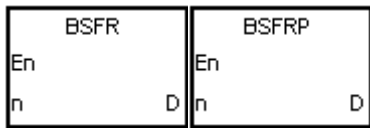
补充说明：

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

API	指令码			操作数							功能						
1112		BSFR	P	D · n							n 个位右移 1 个位 (位装置)						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●				●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



D：位移装置之起始编号 Bit
n：数据长度 Word

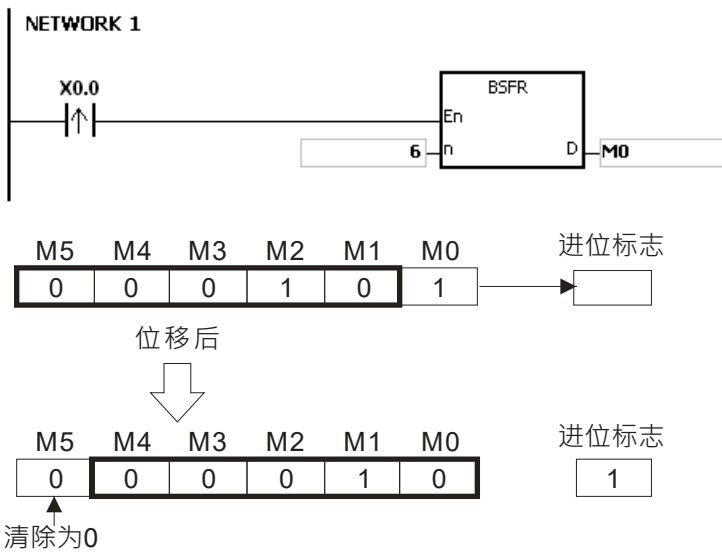
指令说明：

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。

6



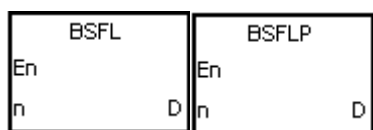
补充说明：

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。

API	指令码			操作数								功能					
1113	BSFL	P		D · n								n 个位左移 1 个位					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●				●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



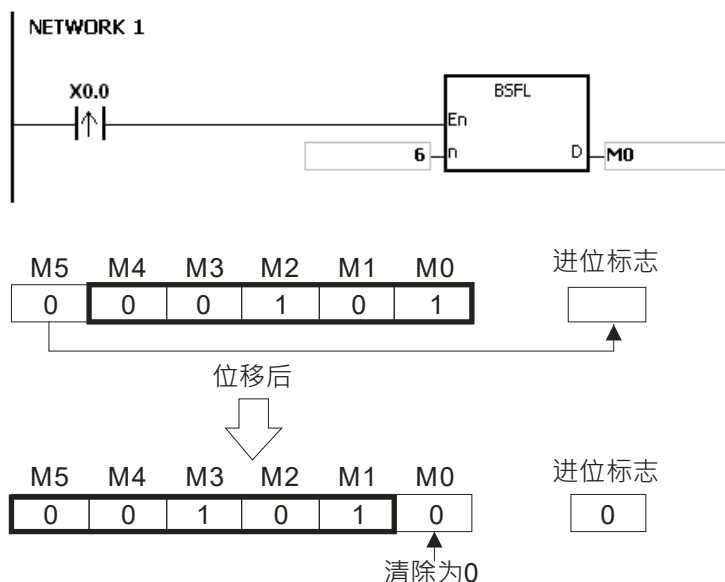
D : 位移装置的起始编号 Bit
n : 数据长度 Word

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 内容依序向左位移 1 个位，M0 的状态清除为 0，M5 的状态传送至进位标志 SM602。



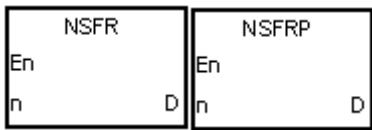
补充说明：

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。

API	指令码			操作数								功能					
1114		NSFR	P	D · n								n 个寄存器右移					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



D : 位移装置的起始编号 Word
n : 数据长度 Word

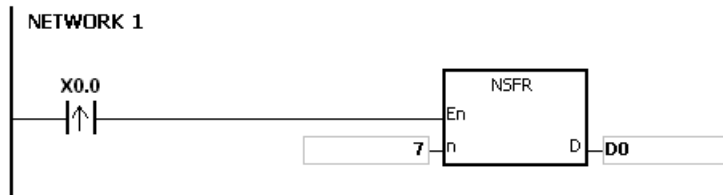
指令说明：

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。

6



D6	D5	D4	D3	D2	D1	D0
30	2235	9578	754	28	423	11

位移后



D6	D5	D4	D3	D2	D1	D0
0	30	2235	9578	754	28	423

清除为0

补充说明：

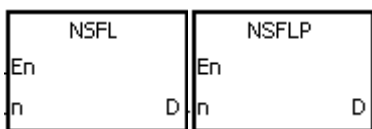
1. **D+n-1** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **n<1** 或 **n>512**，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数							功能						
1115	NSFL	P		D · n							n 个寄存器左移						

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



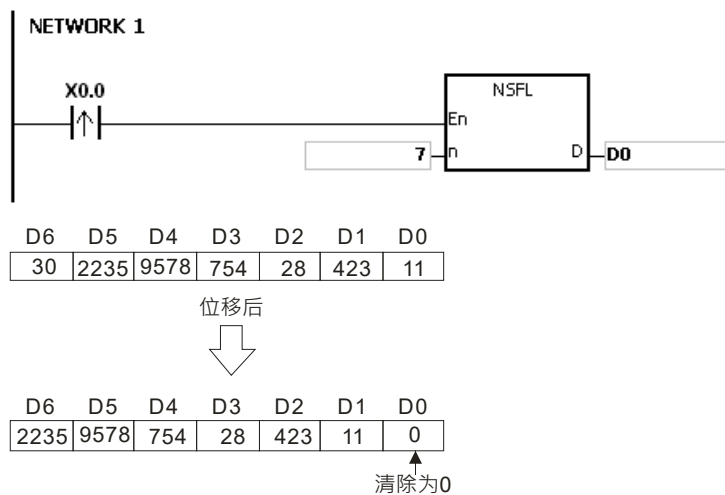
D : 位移装置的起始编号 Word
n : 数据长度 Word

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。



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。

6.13 数据处理指令

6.13.1 数据处理指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
1200	SER	DSER	✓	多点比较	9
1201	SUM	DSUM	✓	ON 位数量	5
1202	DECO	—	✓	译码器	7
1203	ENCO	—	✓	编码器	7
1204	SEGD	—	✓	7 段显示器解码	5
1205	SORT	DSORT	—	数据排序	11
1206	ZRST	—	✓	区域清除	5
1207	BON	DBON	✓	ON 位判定	7
1208	MEAN	DMEAN	✓	平均值	7
1209	CCD	—	✓	总和检查	7
1210	ABS	DABS	✓	绝对值	3
1211	MINV	—	✓	矩阵反相	7
1212	MBRD	—	✓	矩阵位读出	7
1213	MBWR	—	✓	矩阵位写入	7
1214	MBC	—	✓	矩阵位状态计数	7
1215	DIS	—	✓	16位数据的4位分组	7
1216	UNI	—	✓	16位数据的4位链接	7
1217	WSUM	DWSUM	✓	求和	7
1218	BSET	—	✓	字符装置位动作保持 ON	5
1219	BRST	—	✓	字符装置位清除	5
1220	BKRST	—	✓	指定区域清除	5
1221	LIMIT	DLIMIT	✓	高低限控制	9
1222	BAND	DBAND	✓	死区控制	9
1223	ZONE	DZONE	✓	区域控制	9

6.13.2 数据处理指令说明

API	指令码			操作数							功能			
1200	D	SER	P	$S_1 \cdot S_2 \cdot D \cdot n$							多点比较			

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

脉冲执行型	16 位指令 (9 steps)	32 位指令 (9 steps)
AH500	AH500	AH500

符号：

SER		DSER	
En		En	
S1	D	S1	D
S2		S2	
N		N	
SERP		DSERP	
En		En	
S1	D	S1	D
S2		S2	
N		N	

 S_1 ：多点比较之数据区块的起始装置 Word/Double Word S_2 ：欲比较之数值数据 Word/Double Word

D：存放比较结果之起始装置 Word/Double Word

n：被比较之数据区块长度 Word/Double Word

指令说明：

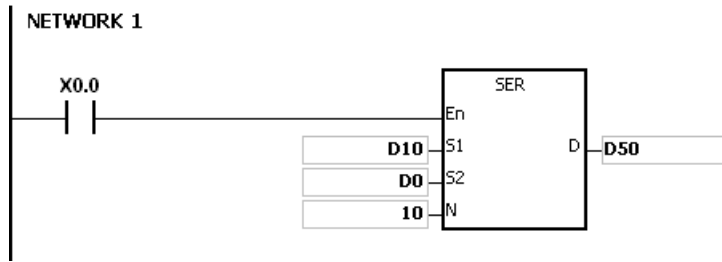
- S_1 指定被比较寄存器区域的号码，n 指定被比较的笔数，该多笔被比较寄存器的内容与 S_2 所指定的数据，以有号数十进制作比较，比较结果被存放于 $D \sim D+4$ 寄存器当中，并说明如下：

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 $D10 \sim D19$ 组成之数据区块与 $D0$ 作比较，结果存放在 $D50 \sim D54$ 中，当相等值不存在时， $D50 \sim D52$ 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：

1. 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。
2. 16 位指令 n<1 或 n>256 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 32 位指令 n<1 或 n>128 时，指令不执行，SM0=ON，错误码 SR0=16#200B。
4. 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。
5. 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。

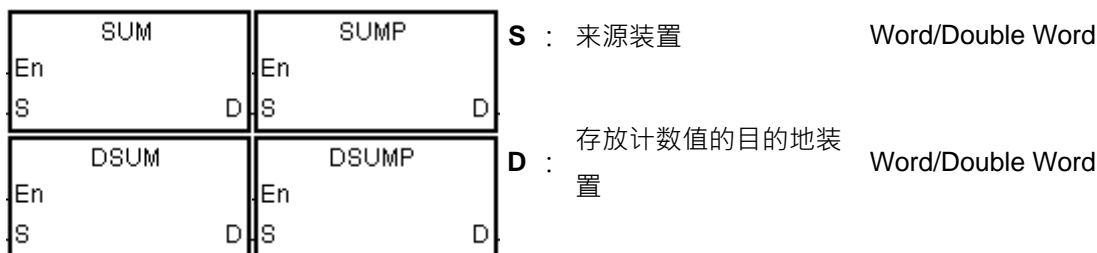
6

API	指令码			操作数							功能				
1201	D	SUM	P	S · D							ON 位数量				

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：

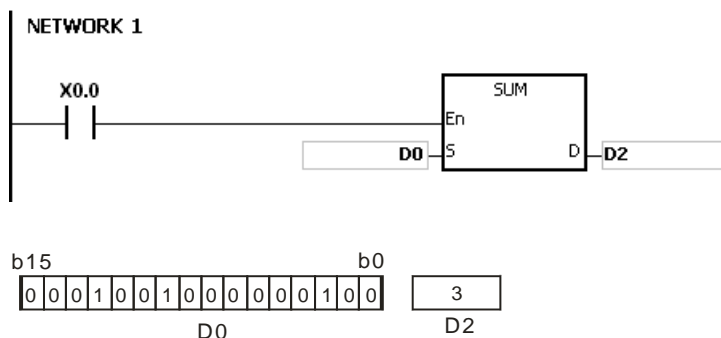


指令说明：

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：

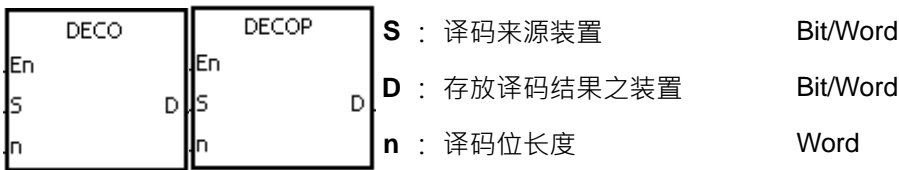
装置超出范围时，指令不执行，SM0=ON，SR0 Error Code=16# 2003。

API	指令码			操作数								功能					
1202		DECO	P	S · D · n								译码器					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



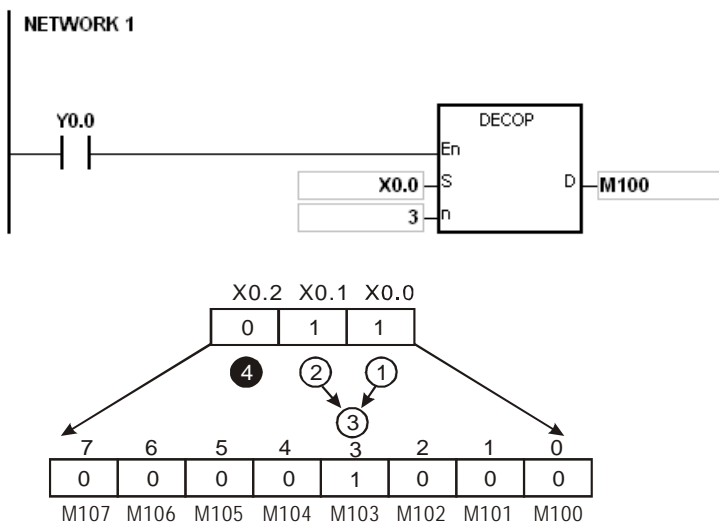
指令说明：

1. 来源装置**S**的下位“**n**”位作译码，并将其“ 2^n ”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 $2^8=256$ 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 $2^4=16$ 点。
4. 本指令一般都是使用脉冲执行型指令 (DECOP)。

6

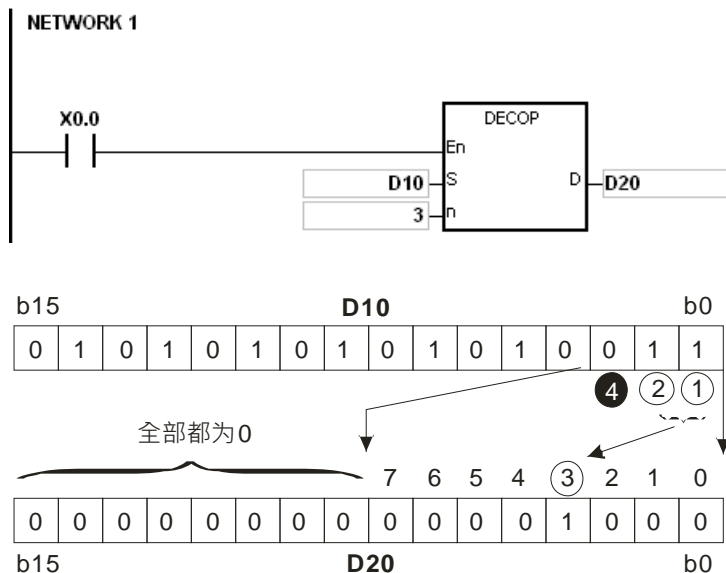
程序范例一：

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 之下位 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：

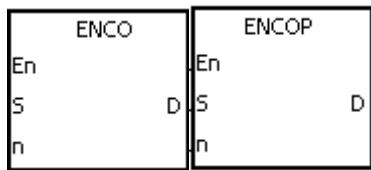
1. **D** 为位装置时，若 $n < 1$ 或 $n > 8$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. **D** 为字符装置时，若 $n < 1$ 或 $n > 4$ ，指令不执行，SM0=ON，错误码 SR0=16#200B。
3. 若 **S** 为位装置时， $S+n-1$ 位超过范围，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. 若 **D** 为位装置时， $D+(2^n)-1$ 位超过范围，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码		操作数				功能						
1203	ENCO	P	S · D · n				编码器						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



- S** : 编码来源装置 Bit/Word
- D** : 存放编码结果之装置 Word
- n** : 编码位长度 Word

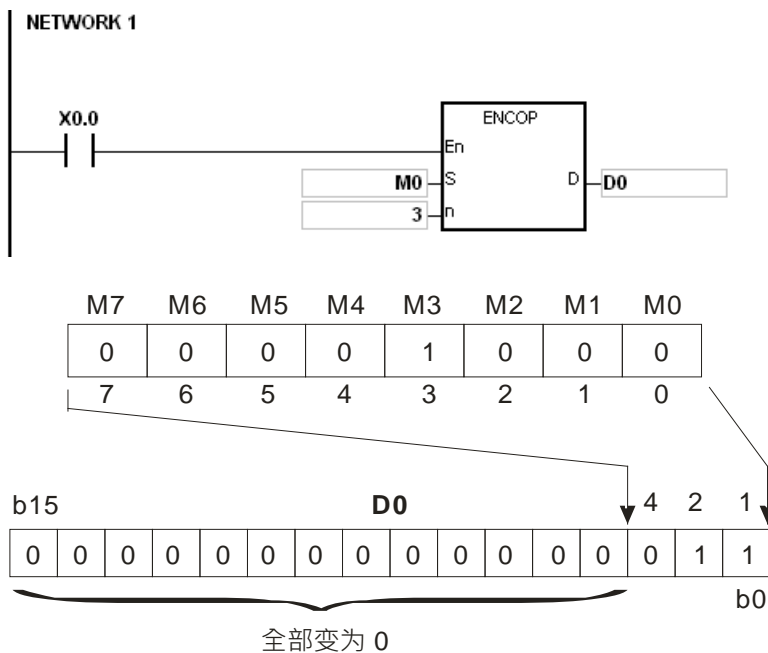
指令说明：

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. 若数据来源**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** = 1~8，当**n**=8 时，可做 $2^8=256$ 点编码。
4. **S**为字符装置时，**n** = 1~4，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)。

6

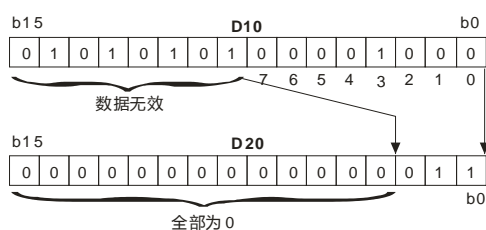
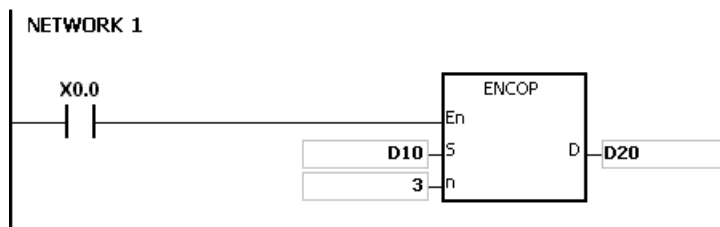
程序范例一：

1. 当X0.0 = OFF→ON时，ENCO指令将 2^3 位数据 (M0~M7) 编码存放于D0 之下位 3 位 (b2~b0) 内，D0 中未被使用之位 (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 之下位 3 位 (b2~b0) 内，D20 中未被使用之位 (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. 若数据来源 **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。

API	指令码			操作数							功能						
1204		SEGD	P	S · D							7 段显示器解码						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



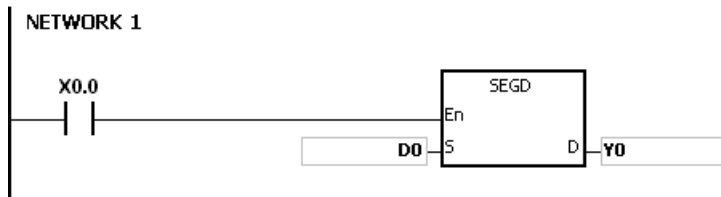
S : 欲译码之来源装置 Word
D : 译码后之输出装置 Word

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：

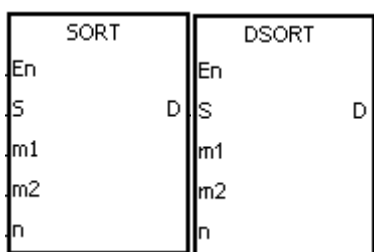
16进制	位组合	七段显示器的构成	各节点状态							显示数据	
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)		
0	0000		ON	ON	ON	ON	ON	ON	OFF	0	
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	ON	OFF	ON	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	ON	b
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	OFF	c
D	1101		OFF	ON	ON	ON	ON	OFF	ON	ON	d
E	1110		ON	OFF	OFF	ON	ON	ON	ON	ON	e
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	ON	f

API	指令码			操作数							功能				
1205	D	SORT		S · m₁ · m₂ · D · n							数据排序				

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

脉冲执行型	16 位指令 (11 steps)	32 位指令 (11 steps)
-	AH500	AH500

符号：



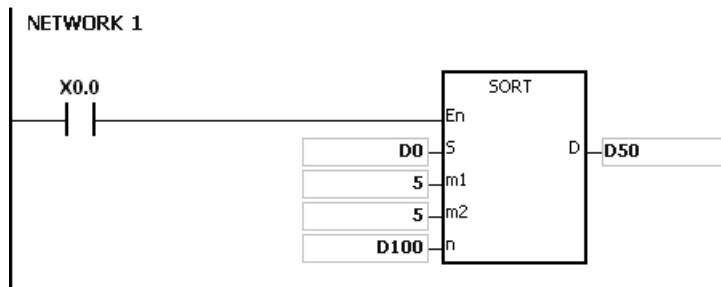
- S** : 原始数据区块之起始装置 Word/Double Word
- m₁** : 被排序之数据组数 Word/Double Word
- m₂** : 每笔数据之字段数 Word/Double Word
- D** : 存放排序结果数据区块之起始装置 Word/Double Word
- n** : 数据排序的参考值 Word/Double Word

指令说明：

- 排序结果显示于 **D** 所指定的起始号码开始算的 $m_1 \times m_2$ 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。
- $m_1=1 \sim 32$ · $m_2=1 \sim 6$ · $n=1 \sim m_2$ 。
- SM604=OFF 时为升幂排序，SM604=ON 时为降序排序。
- 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。



2. 排序数据构成例

		← 数据数 : m_2 个 →				
		数据域位				
	行	1	2	3	4	5
	列	学生编号	国文	英文	数学	理化
↑ 资 料 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

3. D100=3 时的排序后数据

		← 数据数 : m_2 个 →				
		数据域位				
	行	1	2	3	4	5
	列	学生编号	国文	英文	数学	理化
↑ 资 料 个 数 : m_1 个 ↓	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

6

4. D100=5 时的排序后数据。

		数据域位				
		数据数：m ₂ 个				
	行	1	2	3	4	5
	列	学生编号	国文	英文	数学	理化
↑ 资 料 个 数 : m ₁ 个 ↓	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. m₁、m₂或n超出范围时，指令不执行，SM0=ON，错误码SR0=16#200B。

API	指令码			操作数								功能					
1206		ZRST	P	$D_1 \cdot D_2$								区域清除					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D_1	●	●	●	●	●	●	●	●	●	●		○					
D_2	●	●	●	●	●	●	●	●	●	●		○					

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



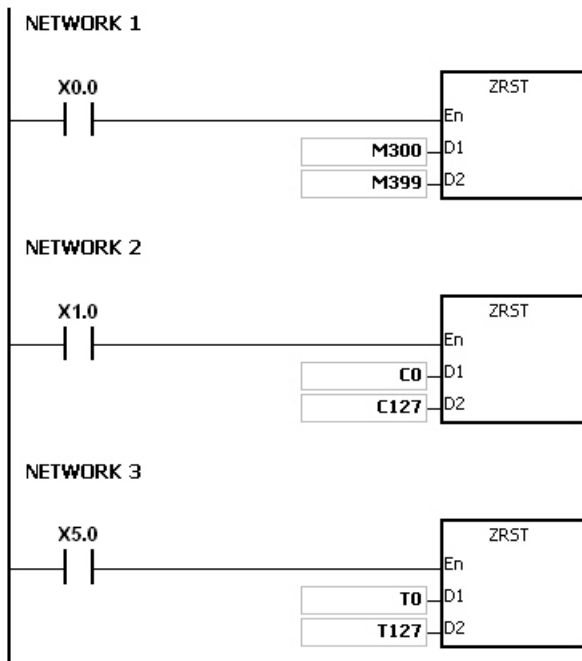
D_1 : 区域清除起始装置 Bit/Word
 D_2 : 区域清除结束装置 Bit/Word

指令说明：

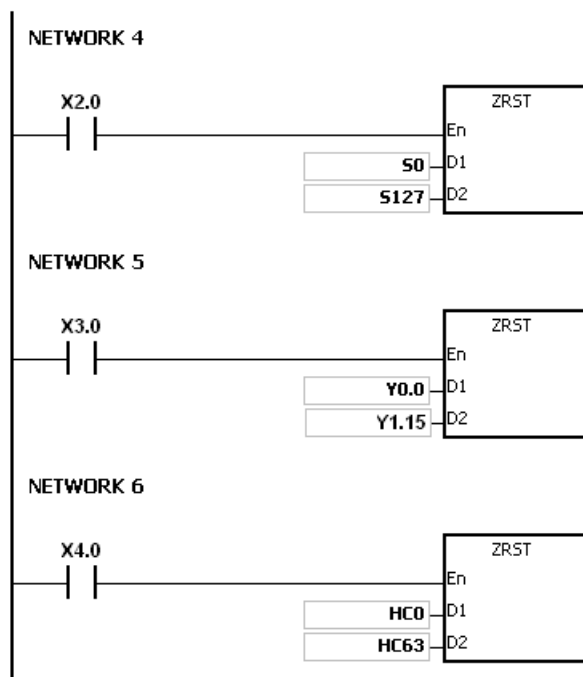
- 清除 $D_1 \sim D_2$ 的操作数内容值。
- 当 D_1 操作数编号 > D_2 操作数编号时，只有 D_2 指定之操作数被清除。

程序范例：

- 当 X0.0 为 ON 时，辅助继电器 M300~M399 被清除成 OFF。
- 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)。
- 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)。
- 当 X2.0 为 ON 时，步进点 S0~S127 被清除成 OFF。
- 当 X3.0 为 ON 时，数据寄存器 Y0.0~Y1.15 被清除成 OFF。
- 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)。



6



补充说明：

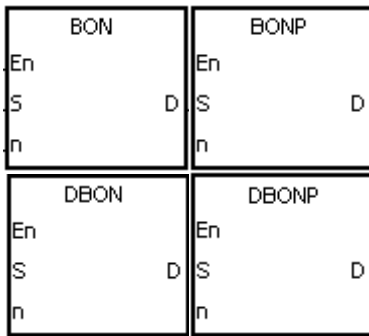
1. D_1 、 D_2 操作数装置类别不相同，指令不执行，SM0=ON，错误码SR0=16#2007。
2. D_1 、 D_2 操作数数据格式不相同，指令不执行，SM0=ON，错误码SR0=16#2007。

API	指令码			操作数							功能				
1207	D	BON	P	S · D · n							ON 位判定				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



- S** : 来源装置 Word/Double Word
- D** : 存放判定结果之装置 Bit
- n** : 指定判定之位 (自 0 开始编号) Word/Double Word

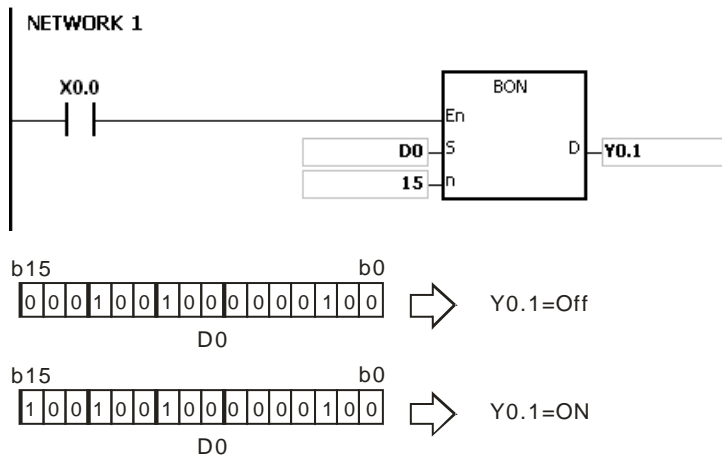
指令说明：

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。

6

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：

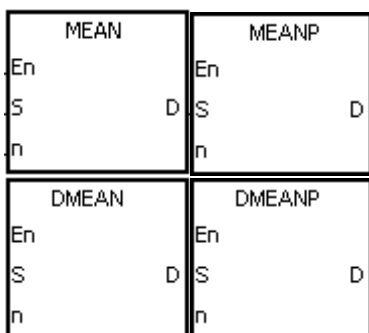
n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数							功能				
1208	D	MEAN	P	S · D · n							平均值				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：



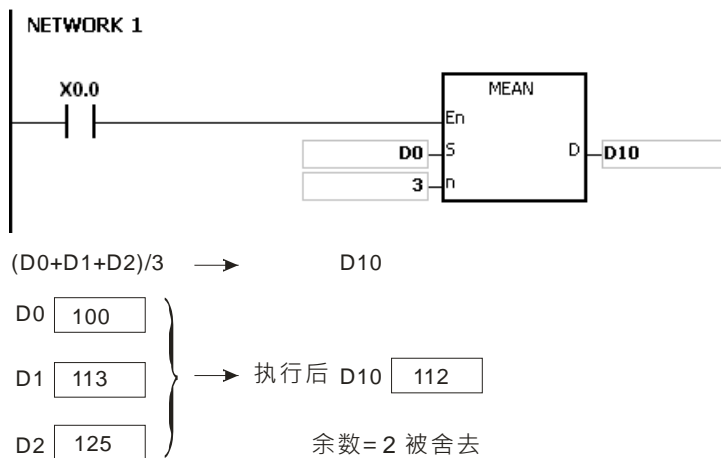
- S** : 欲取平均值之起始装置 Word/Double Word
- D** : 存放平均值之装置 Word/Double Word
- n** : 取平均值之装置个数 Word/Double Word

指令说明：

1. 将 **S** 起始之 **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 以求得平均值并存储于指定的 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 $SR0=16\#2003$ 。

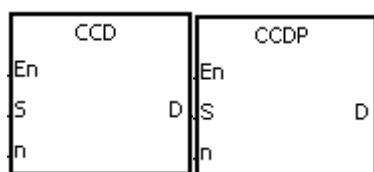
6

API	指令码			操作数							功能				
1209		CCD	P	S · D · n							总和检查				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



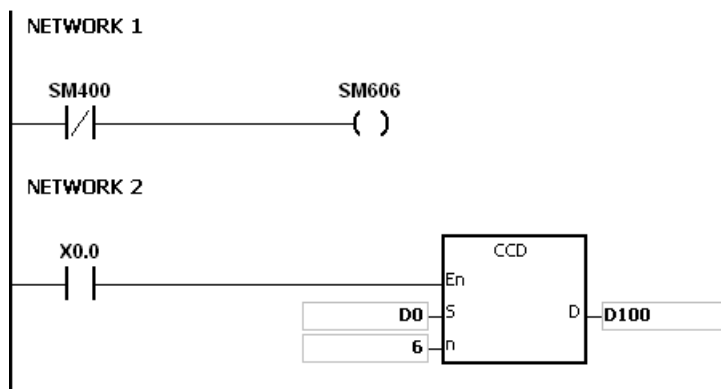
- S** : 数据来源起始装置 Word
- D** : 存放总和检查之结果 Word
- n** : 数据个数 Word

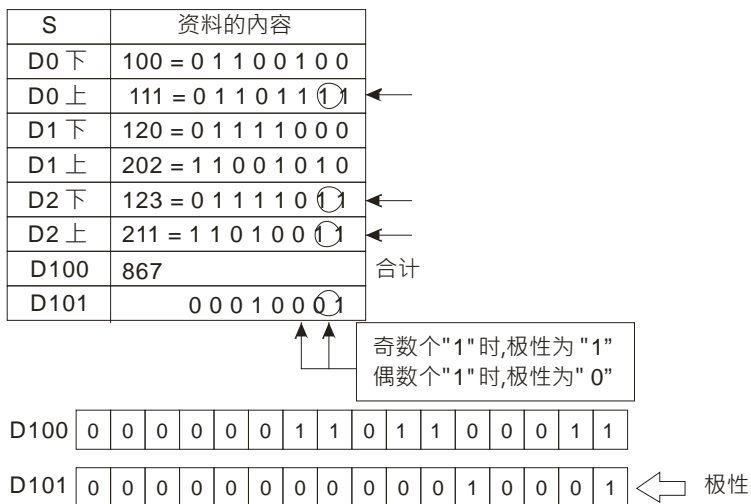
指令说明：

1. 本指令用来作通信时，为了确保数据传输时之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：

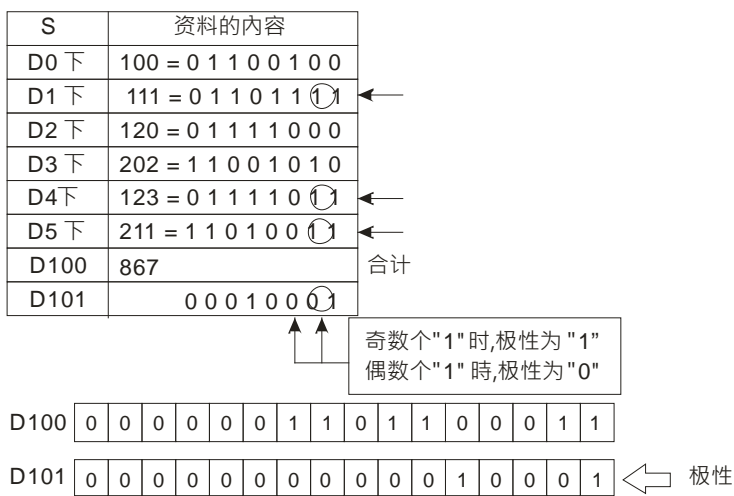
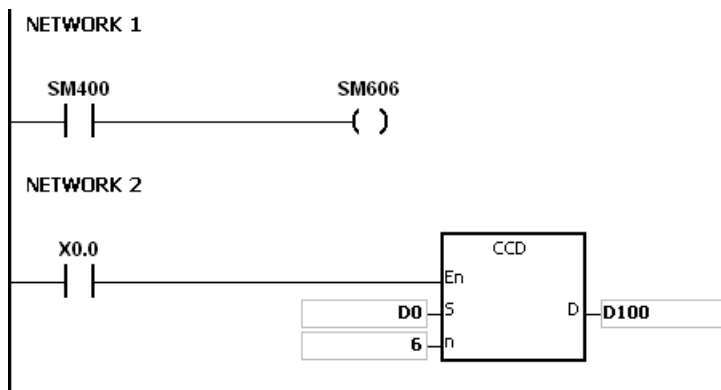
1. 当 SM606=OFF 时，指定为 16 位转换模式。
2. 当 X0.0=ON 时，将 D0 所指定寄存器起始号码开始算的 6 个数据 (以 8 位为单位 n=6 代表指定 D0~D2) 内容作加总，加总结果存放于 D100 所指定的寄存器当中，而极性位存放于 D101 当中。





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。

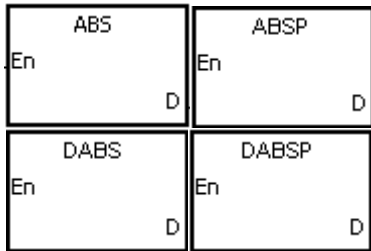


补充说明：

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。

API	指令码			操作数							功能						
1210	D	ABS	P	D							绝对值						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脉冲执行型							16 位指令 (3 steps)					32 位指令 (3 steps)					
AH500							AH500					AH500					

符号：



D：欲取绝对值之装置 Word/Double Word

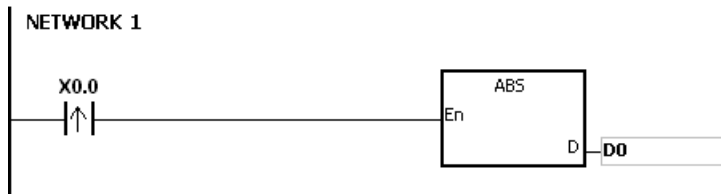
指令说明：

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。

6

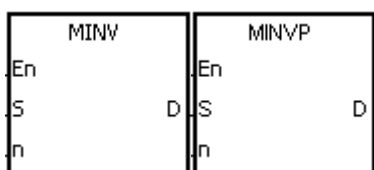


API	指令码		操作数						功能						
1211		MINV	P	S · D · n						矩阵反相					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



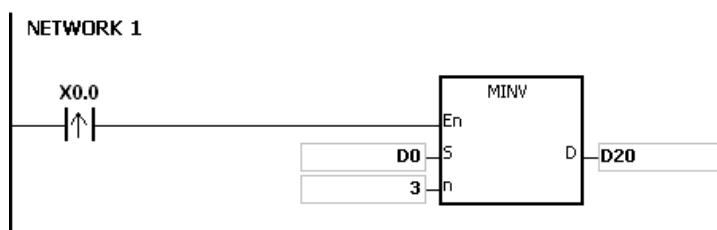
- S** : 矩阵来源装置 Word
- D** : 运算结果 Word
- n** : 数组长度 Word

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
D1	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

执行后 ↓

	b15													b0		
D20	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
D22	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

补充说明：

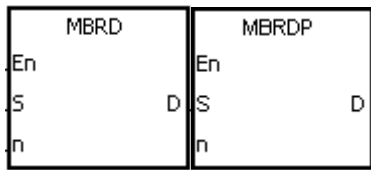
1. **S+n-1**、**D+n-1** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **n<1** 或 **n>256** 时，指令不执行，SM0=ON，SR0 Error Code=16#200B。

API	指令码			操作数							功能						
1212		MBRD	P	S · n · D							矩阵位读出						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



- S**：矩阵来源装置 Word
- n**：数组长度 Word
- D**：指针 Pr，用以存放目标之位置值 Word

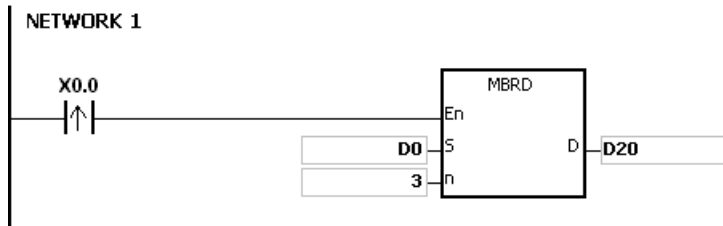
指令说明：

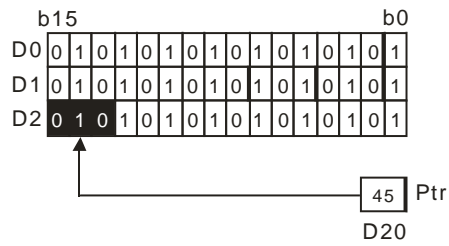
1. 当指令执行时，一开始判断矩阵指标清除标志 SM613 是否为 ON，若为 ON，指标 D 清除为 0，再将指标 D 所指定的 bit 状态读取到矩阵旋转位移输出进位标志 SM614，当读取完一个 bit 时，判断矩阵指标递增标志 SM612 是否 ON，若 ON 把指标 D 的值加 1，然后结束本指令之执行。
2. 当读到最后一个 bit 时矩阵搜寻结束标志 SM608=ON，指针 D 记录着读取的 bit 之编号，然后结束本指令之执行。
3. n 操作数的范围值为 1~256。
4. 矩阵之指针 Pr (Pointer)，由使用者于指令中指定，其有效范围为 0~16n-1，分别对应至矩阵中之位 b₀ ~ b_{16n-1}。若 Pr 值超出此范围则矩阵指标错误标志 SM611 设为 1，且本指令不执行。

6

程序范例：

1. 当 X0.0 由 OFF→ON 时，设指标清除标志 SM613=0、矩阵指标递增标志 SM612=1，所以每读取一次指标 Pr 增加 1。
2. 设指标当时值 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，矩阵旋转位移输出进位标志 SM614=0，矩阵搜寻结束标志 SM608=0。
 - ❷ D20=47，矩阵旋转位移输出进位标志 SM614=1，矩阵搜寻结束标志 SM608=0。
 - ❸ D20=47，矩阵旋转位移输出进位标志 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 Error Code=16#2003。
2. **n<1** 或 **n>256** 时，指令不执行，SM0=ON，SR0 Error Code=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： 矩阵指标错误标志，指标 Pr 值超出此范围则 SM611=1
 - SM612： 矩阵指标递增标志，将指标目前值+1
 - SM613： 矩阵指标清除标志，将指标目前值清除为 0
 - SM614： 矩阵旋转位移输出进位标志

API	指令码			操作数							功能				
1213		MBWR	P	S · n · D							矩阵位写入				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



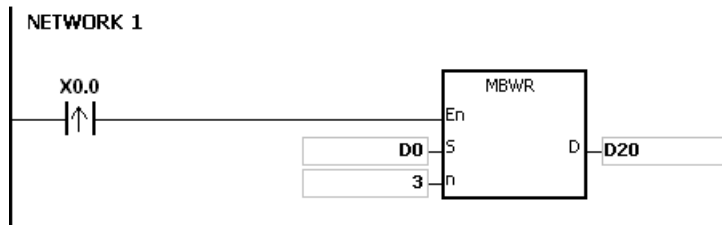
- S** : 矩阵来源装置 Word
- n** : 数组长度 Word
- D** : 指针 Pr，用以存放目标之位置值 Word

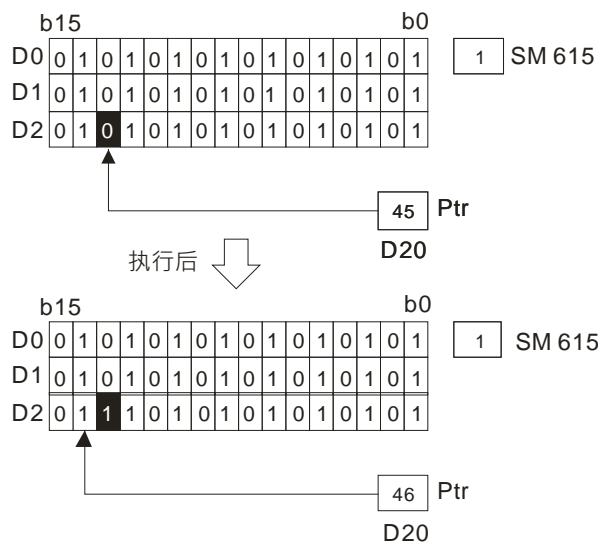
指令说明：

1. 当指令执行时，一开始判断矩阵指标清除标志 SM613 是否为 ON，若为 ON，指标 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，判断矩阵指标递增标志 SM612 是否 ON，若 ON 把指标 D 的值加 1，然后结束本指令之执行。
2. 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 之编号，然后结束本指令之执行。若 D 的值超过范围则 SM611=ON。
3. n 操作数的范围值为 1~256。
4. 矩阵之指针 Pr (Pointer)，由使用者于指令中指定，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标错误标志 SM611 设为 1，且本指令不执行。

程序范例：

1. 当 X0.0 由 OFF→ON 时，设指标清除标志 SM613=0、矩阵指标递增标志 SM612=1，所以每写入一次指标 Pr 增加 1。
2. 设指标当时值 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 Error Code=16#2003。
2. $n<1$ 或 $n>256$ 时，指令不执行，SM0=ON，SR0 Error Code=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： 矩阵指标错误标志，指标 Pr 值超出此范围则 SM611=1
 SM612： 矩阵指标递增标志
 SM613： 矩阵指标清除标志
 SM615： 矩阵位移输入补位标志

API	指令码			操作数							功能						
1214		MBC	P	S · n · D							矩阵位状态计数						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●		●	○	○		
D	●	●			●	●		●	●		●		●				

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



- S** : 矩阵来源装置 Word
- n** : 数组长度 Word
- D** : 运算结果 Word

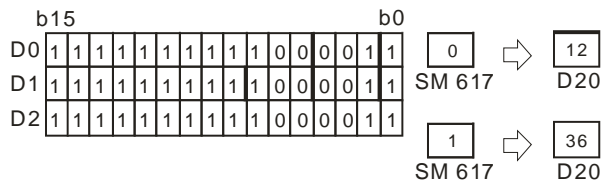
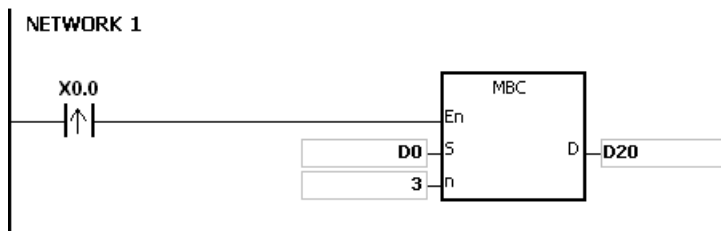
指令说明：

- 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。
- 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**。
- n** 操作数的范围值为 1~256。

程序范例：

6

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：

- S+n-1** 装置超出范围时，指令不执行，**SM0=ON**，**SR0 Error Code=16#2003**。
- n<1** 或 **n>256** 时，指令不执行，**SM0=ON**，**SR0 Error Code=16#200B**。
- 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

API	指令码			操作数								功能					
1215		DIS	P	S · n · D								16 位数据的 4 位分组					

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

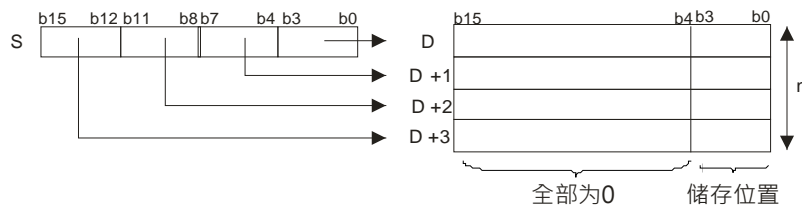
脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



指令说明：

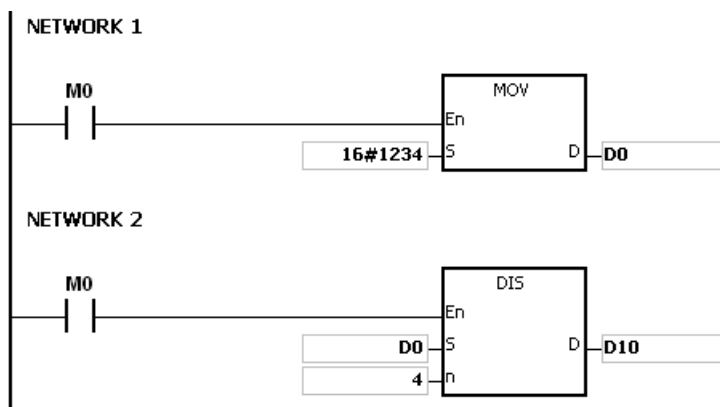
1. 将**S**所指定寄存器内容值，从**b₀**开始以 4 个位为 1 组，分别存放于**D~D+(n-1)**所指定的寄存器的低 4 位中 (**b₀~b₃**)。

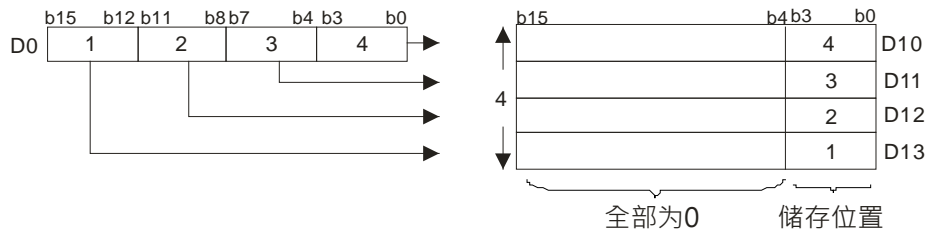


2. **n=1~4**。

程序范例：

D0 的内容为 16#1234。启动 M0，执行 DIS 指令时，将 D0 的 b₀~b₁₅ 内容分为 4 个 Bit 一组，分别移动至 D10~D13 的 b₀~b₃ 当中。





补充说明：

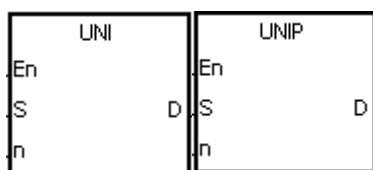
1. **D~D+ (n-1)** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **n<1 或 n>4** 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能				
1216		UNI	P	S · n · D								16 位数据的 4 位链接				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

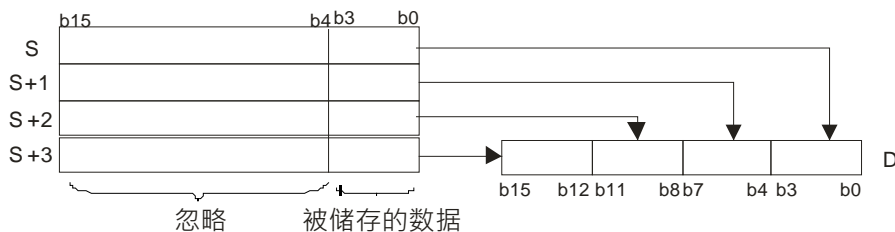
符号：



- S** : 数据来源装置 Word
- n** : 数据来源长度 Word
- D** : 运算结果 Word

指令说明：

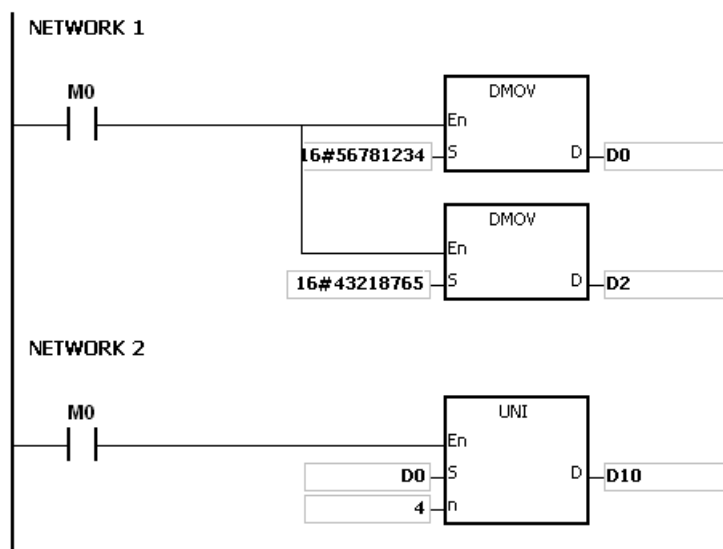
1. 将 **S~S+ (n-1)** 所指定寄存器内容值 (b₀~b₃) · 依序存放于 **D** (b₀~b₁₅) 所指定的寄存器中。



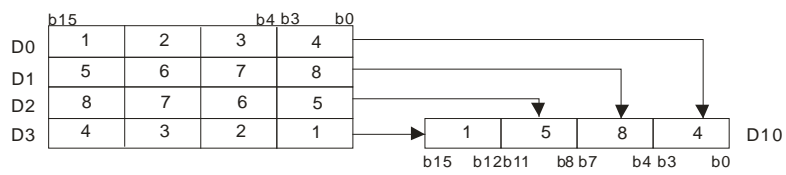
2. **n=1~4**。

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 b0~b3 内容，分别储存至 D10 的 b0~b15 当中。



6



补充说明：

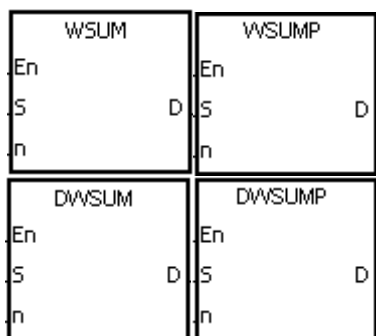
1. **S~S+ (n-1)** 装置超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. **n<1** 或 **n>4** 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数							功能				
1217	D	WSUM	P	S · n · D							求和				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

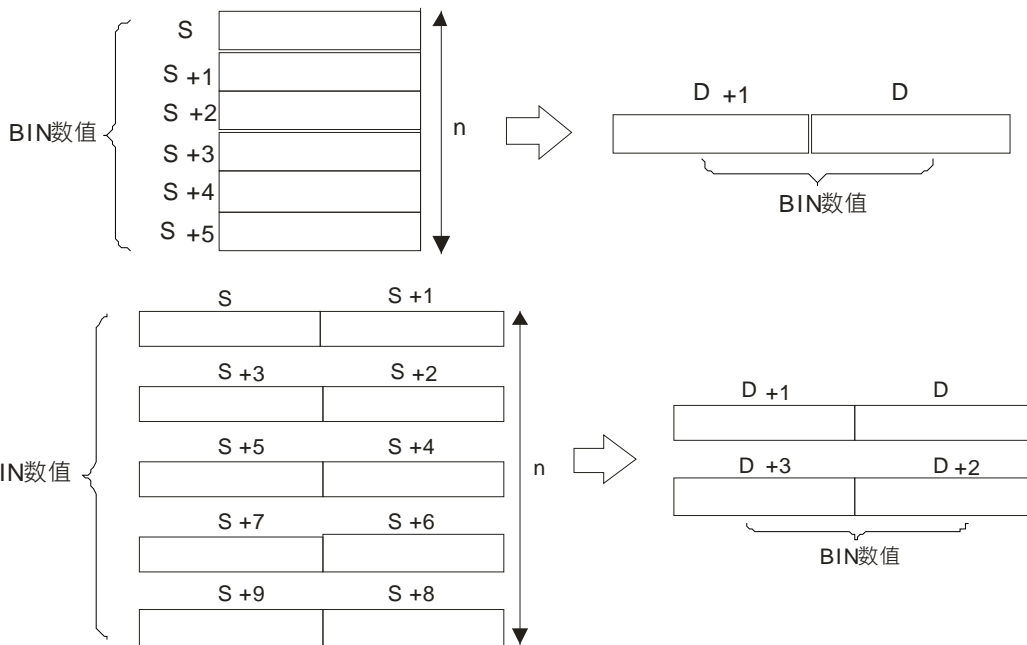
符号：



- S** : 数据来源装置 Word/Double Word
- n** : 数据来源长度 Word/Double Word
- D** : 运算结果 Double Word/Long Word

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。

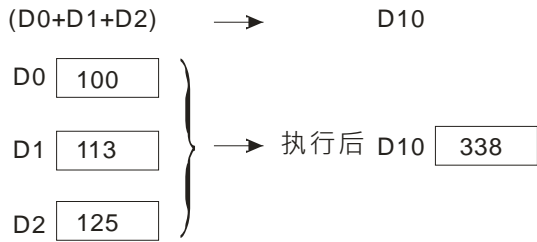
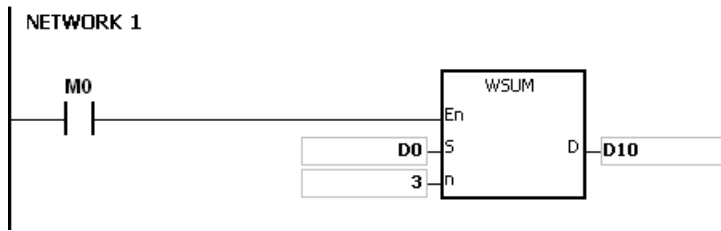


2. 16 位指令中，**n** 操作数的内容值为 1~256；32 位指令中，**n** 操作数的内容值为 1~128。
3. 32 位指令才可以使用 HC 装置。

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。

6



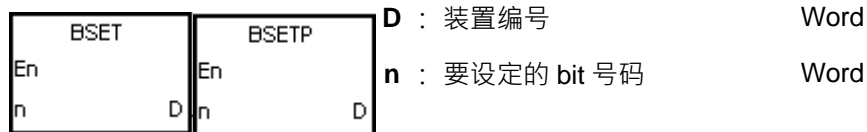
补充说明：

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 $S+n-1$ 或 D 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。

API	指令码			操作数								功能					
1218		BSET	P	D · n								字符装置位动作保持 ON					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

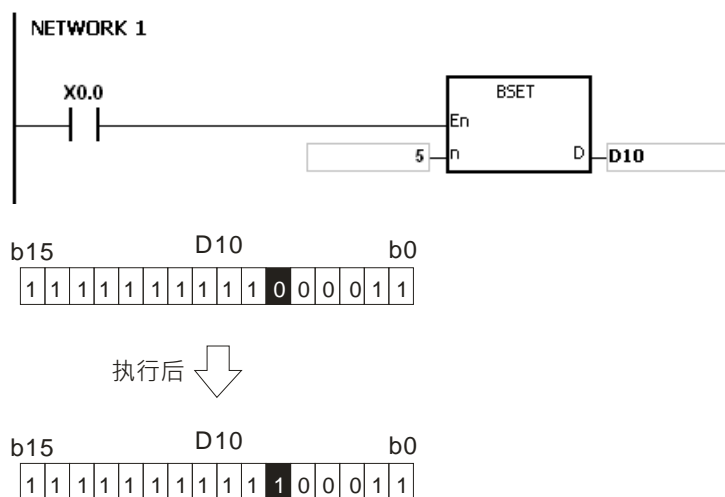


指令说明：

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。



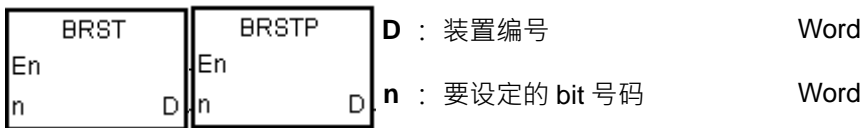
补充说明：

n<0 或 n>15 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能					
1219		BRST	P	D · n								字符装置位清除					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●		●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



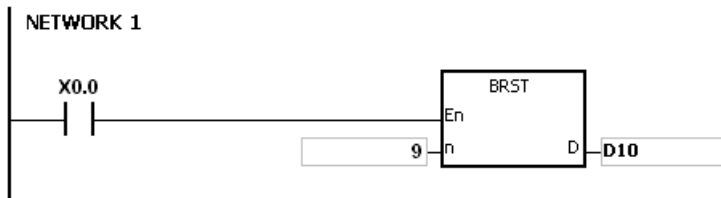
指令说明：

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。

6



执行后 ↓



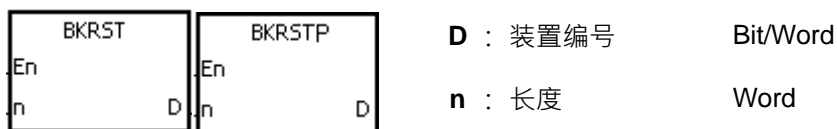
补充说明：

n<0 或 n>15 时，指令不执行，SM0=ON，错误码 SR0=16#200B。

API	指令码			操作数								功能					
1220		BKRST	P	D · n								指定区域清除					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●			○	●				
n	●	●	●	●	●	●	●	●	●	●	●	○	●	○	○		

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

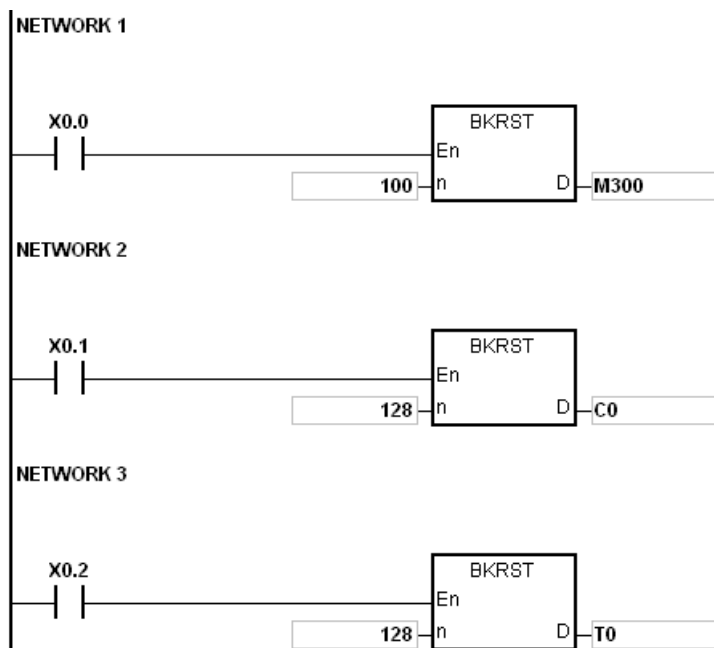


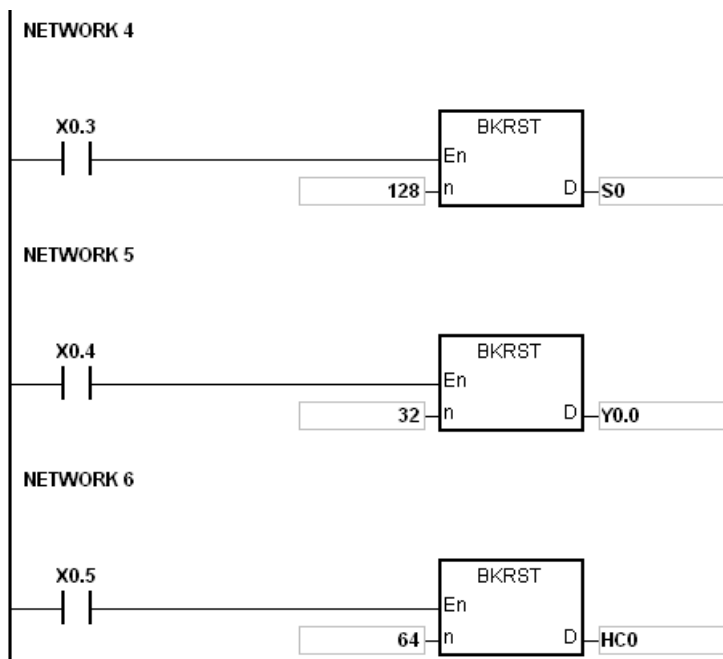
指令说明：

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。

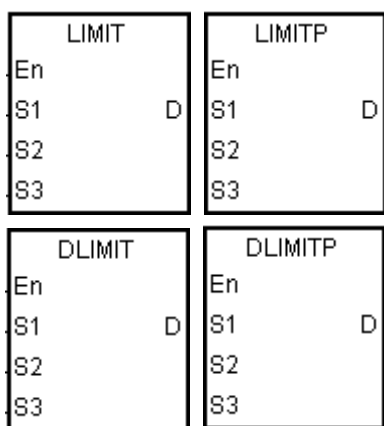
6

API	指令码			操作数							功能				
1221	D	LIMIT	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							高低限控制				

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令 (9 steps)
AH500	AH500	AH500

符号：



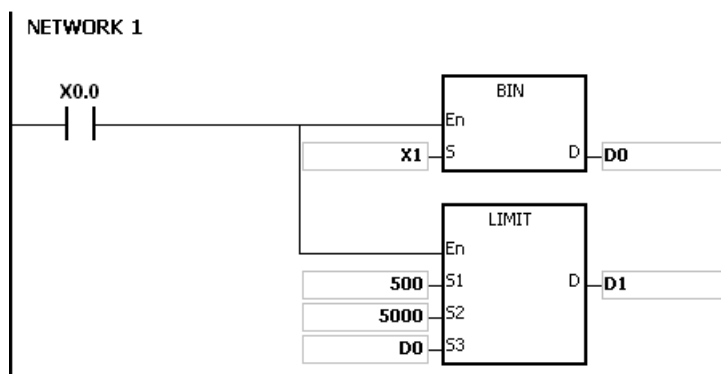
- S_1 : 最小输出值 Word/Double Word
- S_2 : 最大输出值 Word/Double Word
- S_3 : 输入值 Word/Double Word
- D : 输出值 Word/Double Word

指令说明：

1. 输入值 S_3 与最小输出值 S_1 及最大输出值 S_2 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 $S_1 >$ 输入值 S_3 ，输出值D=最小输出值 S_1 。
 - 若最大输出值 $S_2 <$ 输入值 S_3 ，输出值D=最大输出值 S_2 。
 - 若最小输出值 $S_1 \leq$ 输入值 $S_3 \leq$ 最大输出值 S_2 ，输出值D=输入值 S_3 。
 - 若最小输出值 $S_1 >$ 最大输出值 S_2 ，输出值D不执行。
2. 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，将结果储存在 D1。



6

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。

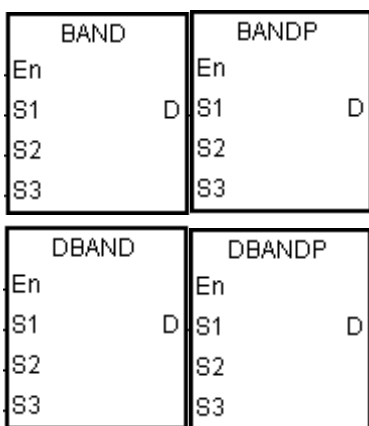
6

API	指令码			操作数							功能				
1222	D	BAND	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							死区控制				

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令 (9 steps)
AH500	AH500	AH500

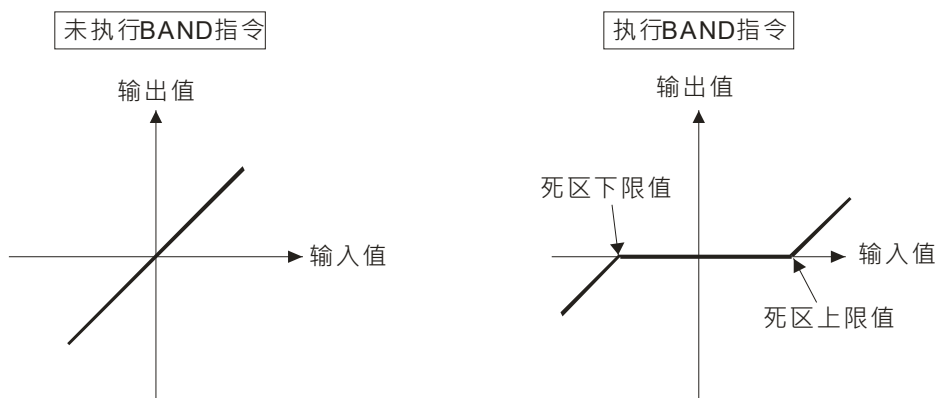
符号：



- S_1 : 死区下限值 Word/Double Word
- S_2 : 死区上限值 Word/Double Word
- S_3 : 输入值 Word/Double Word
- D : 输出值 Word/Double Word

指令说明：

1. 输入值 S_3 与死区下限值 S_1 及死区上限值 S_2 作相减，相减后将结果存入D。说明如下：
 - 若最小输出值 $S_1 >$ 输入值 S_3 ，输出值 $D = S_3 - S_1$ 。
 - 若最大输出值 $S_2 <$ 输入值 S_3 ，输出值 $D = S_3 - S_2$ 。
 - 若最小输出值 $S_1 \leq$ 输入值 $S_3 \leq$ 最大输出值 S_2 ，输出值 $D = 0$ 。
 - 若最小输出值 $S_1 >$ 最大输出值 S_2 ，输出值D不执行。
2. 32 位指令才可以使用 HC 装置。
3. 图形说明如下：

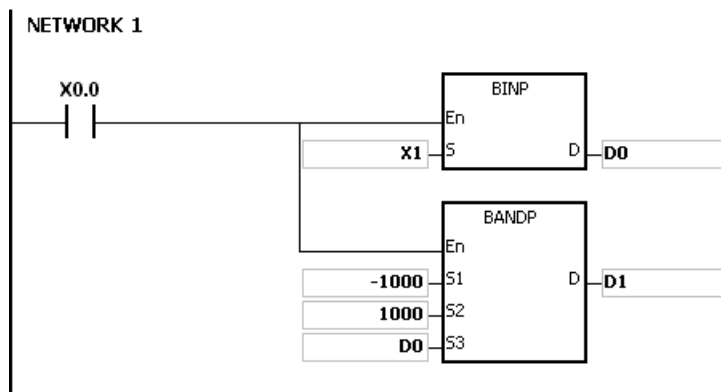


4. S_1 、 S_2 和 S_3 的范围：

- 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 ($S_1+1 \cdot S_1$) =1000，输入值 ($S_3+1 \cdot S_3$) =-2147483648
 输出值 ($D+1 \cdot D$) =-2147483648-1000=16#80000000-16#000003E8=16#7FFFC18
 =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。

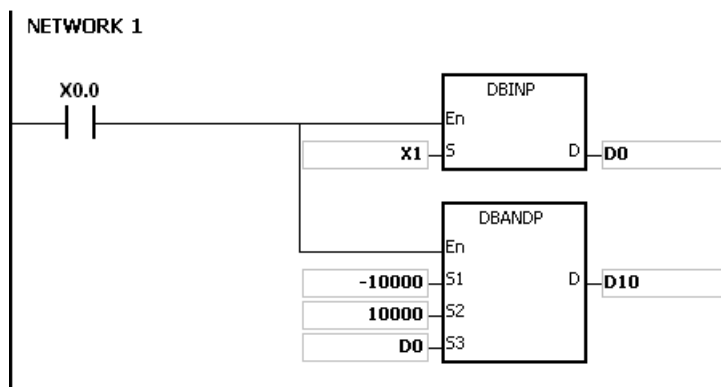


执行结果：

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：

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	$(D1 \cdot D0) < -10000$ $\Rightarrow (D11 \cdot D10)$ $= (D1 \cdot D0) - (-10000)$	-2000
		12000	$(D1 \cdot D0) > 10000$ $\Rightarrow (D11 \cdot D10)$ $= (D1 \cdot D0) - 10000$	2000
		5000	$-10000 \leq (D1 \cdot D0) \leq 10000$ $\Rightarrow (D1 \cdot D0) = 0$	0

补充说明：

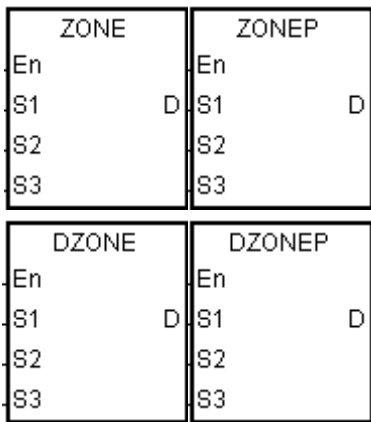
$S_1 > S_2$ 时，指令不执行，SM0=ON，错误码SR0=16#2003。

API	指令码			操作数						功能					
1223	D	ZONE	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$						区域控制					

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令 (9 steps)
AH500	AH500	AH500

符号：

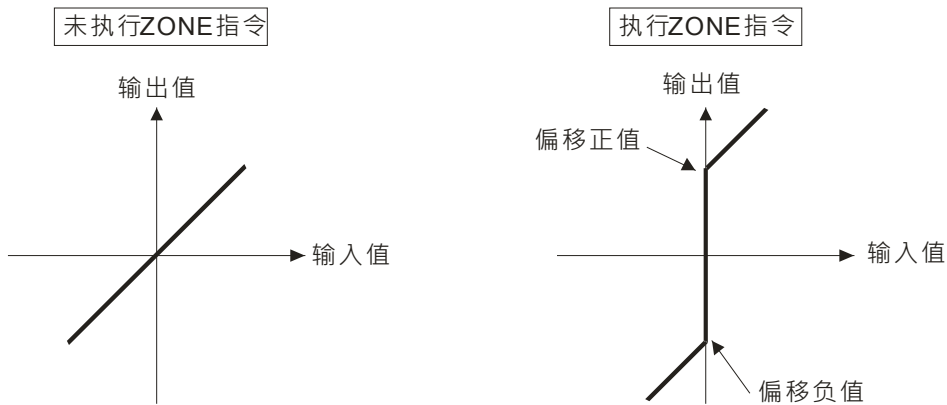


- S_1 : 负偏差值 Word/Double Word
- S_2 : 正偏差值 Word/Double Word
- S_3 : 输入值 Word/Double Word
- D : 输出值 Word/Double Word

6

指令说明：

1. 输入值 S_3 与负偏差值 S_1 或正偏差值 S_2 以作相加，并将结果存入D。说明如下：
 - 若输入值 $S_3 < 0$ ，输出值 $D = S_3 + S_1$ 。
 - 若输入值 $S_3 > 0$ ，输出值 $D = S_3 + S_2$ 。
 - 若输入值 $S_3 = 0$ ，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：

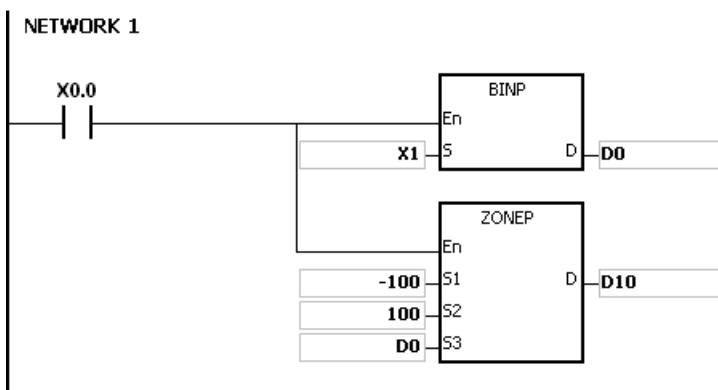
偏差负值 ($S_1+1 \cdot S_1$) = -1000，输入值 ($S_3+1 \cdot S_3$) = -2147483648

输出值 ($D+1 \cdot D$) = -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。

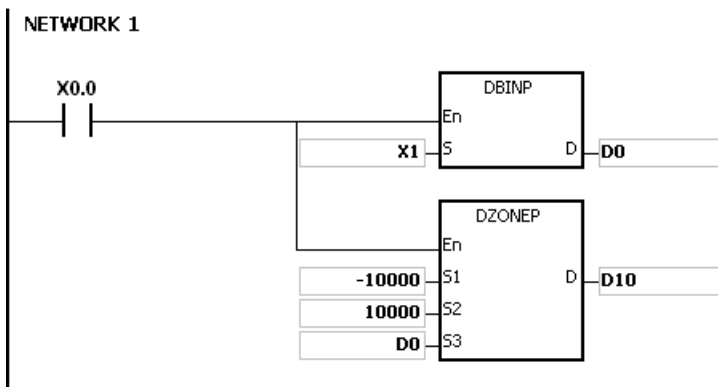


执行结果：

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	$(D1 \cdot D0) < 0$ $\Rightarrow (D11 \cdot D10)$ $= (-10) + (-10000)$	-10010
		0	$(D1 \cdot D0) = 0$ $\Rightarrow (D11 \cdot D10) = 0$	0
		50	$(D1 \cdot D0) > 0$ $\Rightarrow (D11 \cdot D10) = 50 + 10000$	10050

6

6.14 结构建立指令

6.14.1 结构建立指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1300</u>	FOR	—	—	循环回路起始	3
<u>1301</u>	NEXT	—	—	循环回路结束	1
<u>1302</u>	BREAK	—	✓	强制结束 FOR-NEXT 循环	5

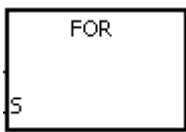
6.14.2 结构建立指令说明

API	指令码		操作数				功能			
1300	FOR		S				循环回路起始			

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

脉冲执行型	16 位指令 (3 steps)	32 位指令
-	AH500	-

符号：



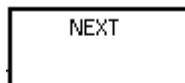
S : 回路重复执行的次数

WORD

API	指令码	操作数	功能
1301	NEXT	-	循环回路结束

脉冲执行型	16 位指令 (1 step)	32 位指令
-	AH500	-

符号：

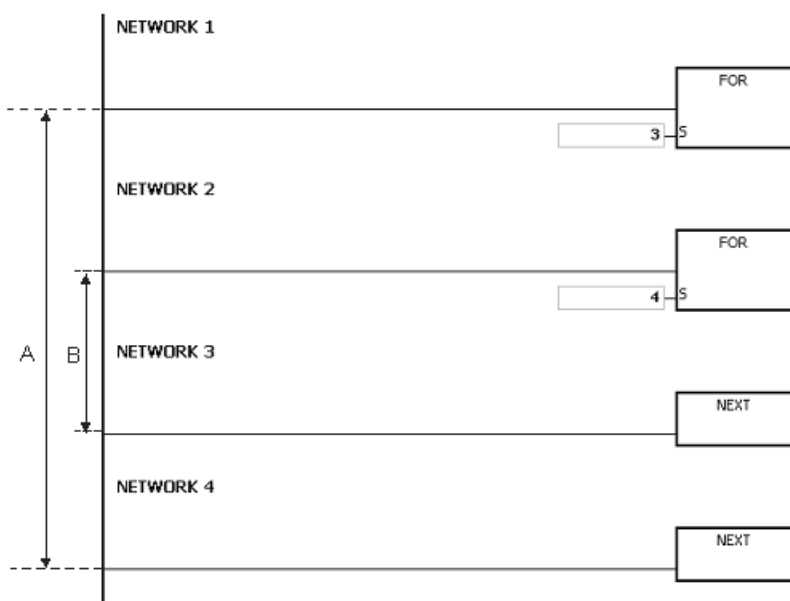


指令说明：

1. 由 FOR 指令指定 FOR~NEXT 循环来回执行 N 次后跳出 FOR~NEXT 循环往下继续执行。
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。

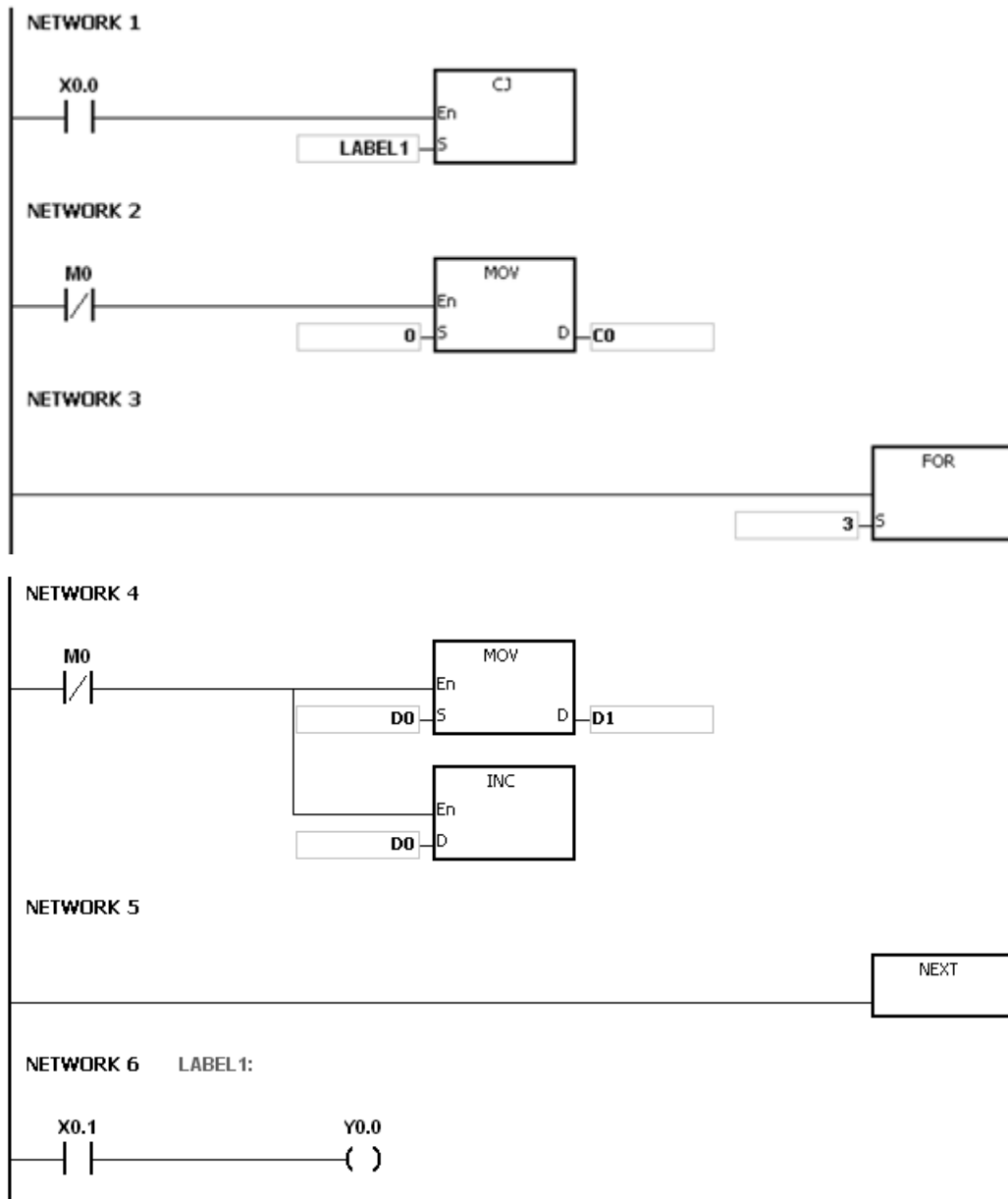
程序范例一：

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：

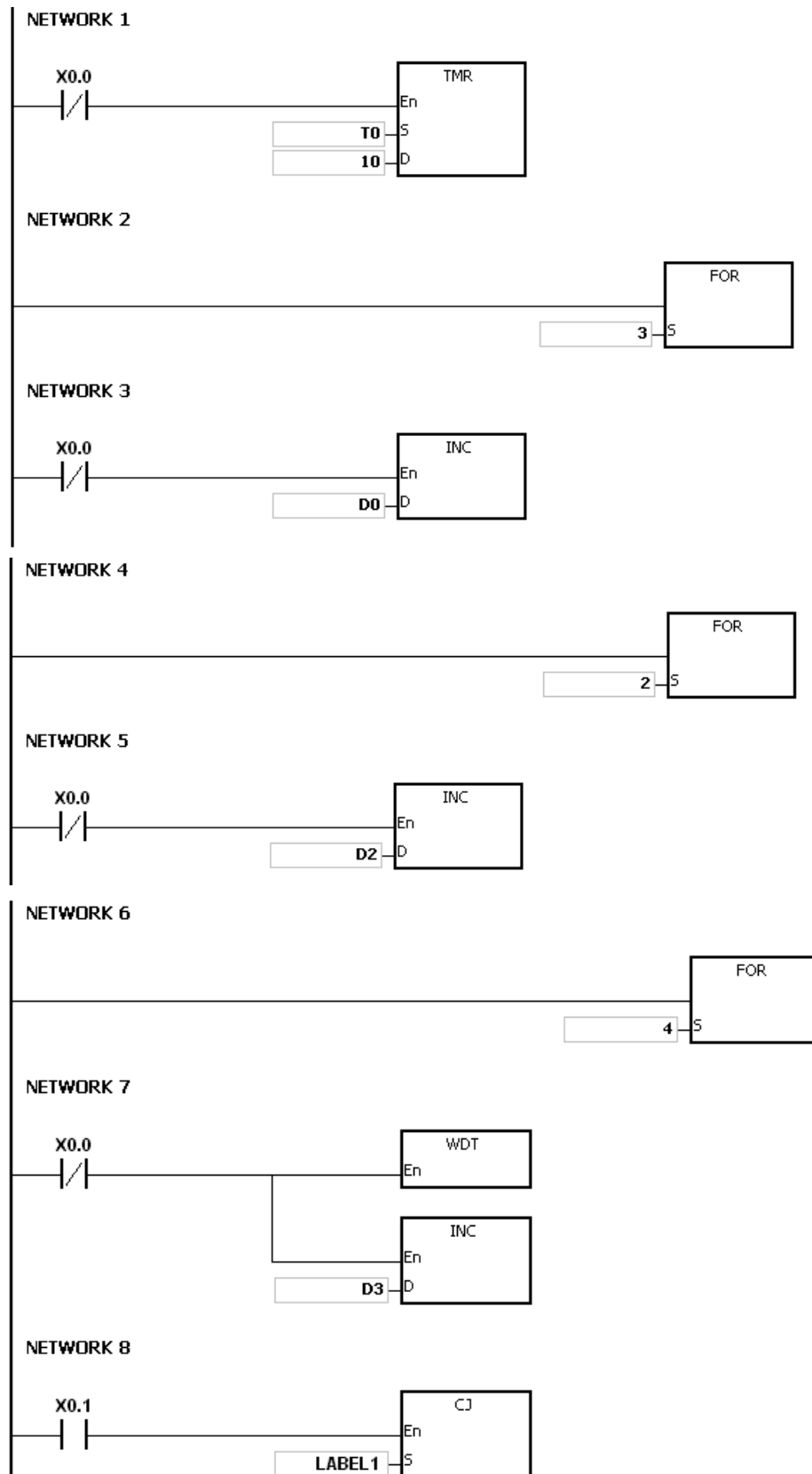
当 X0.0=OFF 时，PLC 会执行 FOR~NEXT 之间的程序，当 X0.0=ON 时，CJ 指令执行跳转到 LABEL 1：即 NETWORK 6 处，FOR~NEXT 之间的程序 NETWORK 4~NETWORK 5 跳过不执行。

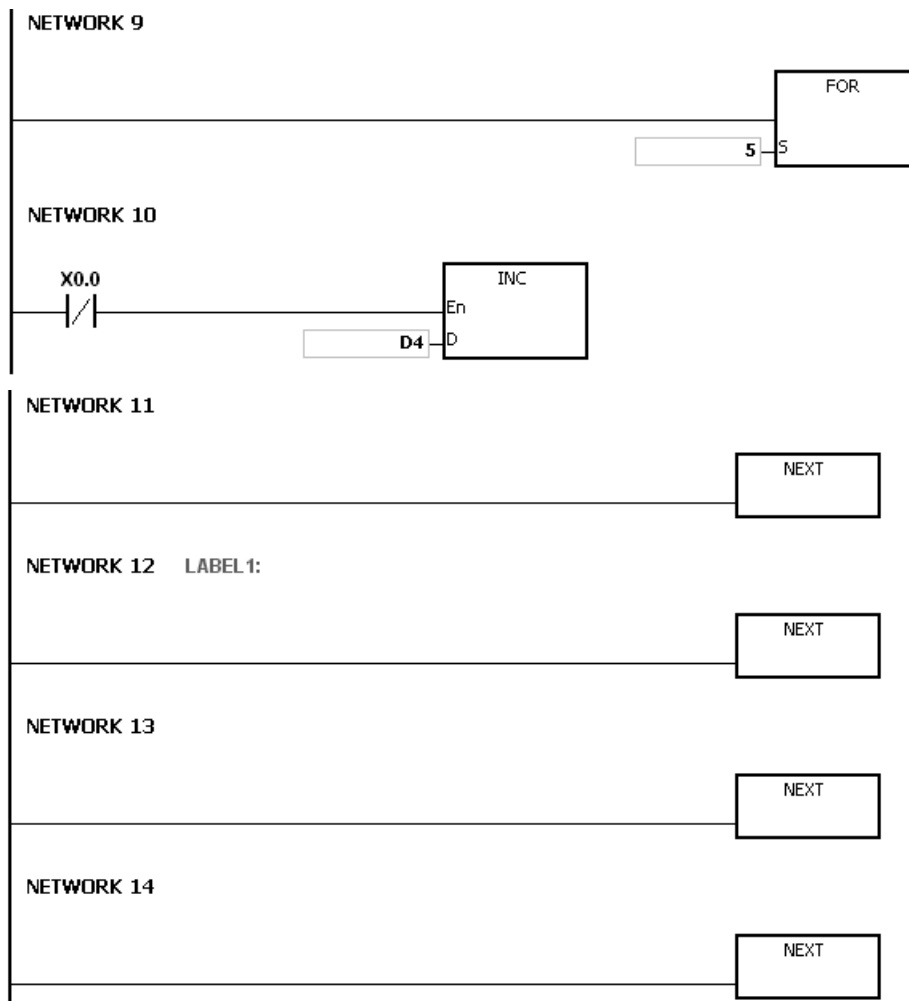


6

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 指令执行跳转到 Label 1：即 NETWORK 12 处，NETWORK 9~NETWORK 11 跳过不执行。





6

补充说明：

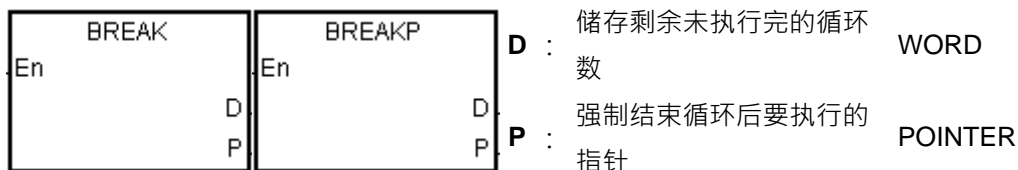
指针 P 在 ISPSOft 中的使用方式，请参考 ISPSOft 的使用手册说明。

API	指令码			操作数							功能				
1302	BREAK	P		D · P							强制结束 FOR-NEXT 循环				

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

脉冲执行型	16 位指令 (3 steps)	32 位指令
AH500	AH500	-

符号：

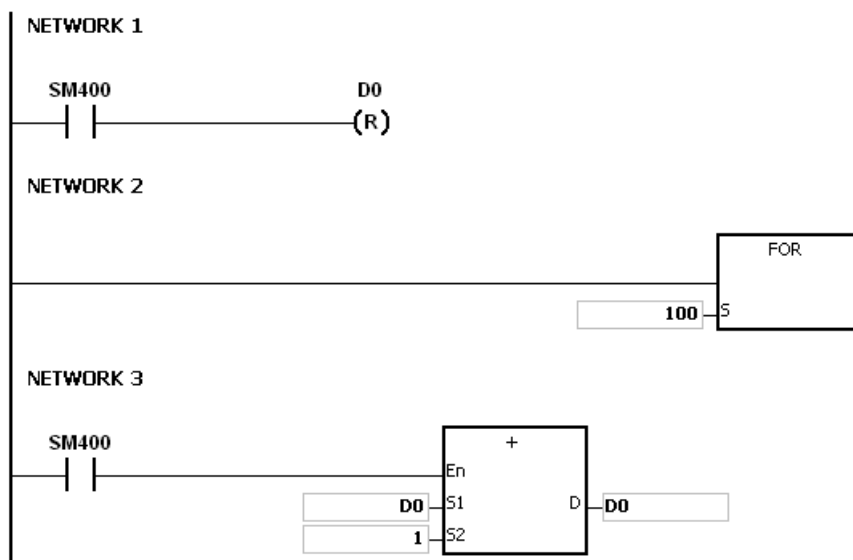


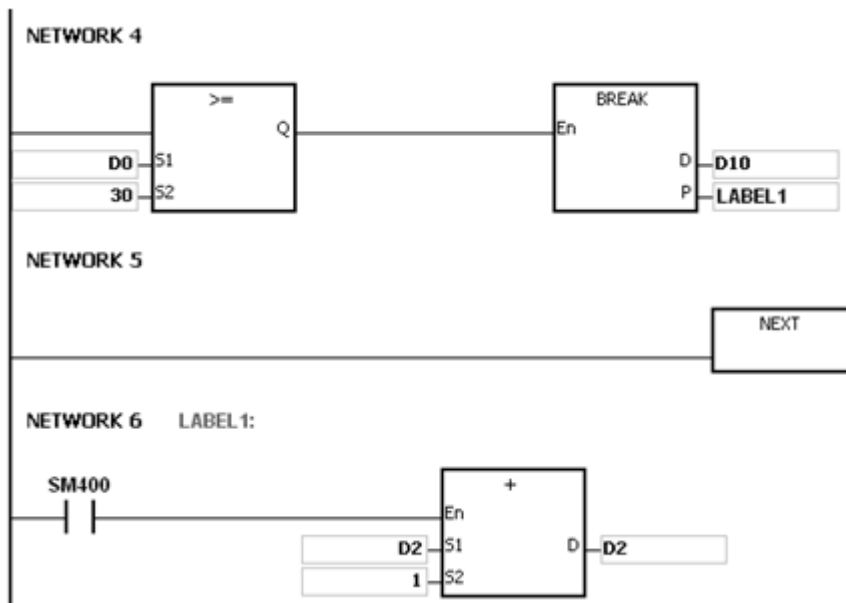
指令说明：

1. 强制结束 FOR-NEXT 的循环，将尚未执行完的循环数存放到 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 剩余尚未执行的循环数存放到 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 循环的程序 D0 内容加 1，当 D0 内容等于 30 时，将强制结束 FOR-NEXT 循环，将剩余的循环数 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 的使用手册说明。

6

6.15 模块的数据读写指令

6.15.1 模块的数据读写指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1400</u>	FROM	DFROM	✓	特殊模块 CR 数据读出	13
<u>1401</u>	TO	DTO	✓	特殊模块 CR 数据写入	13

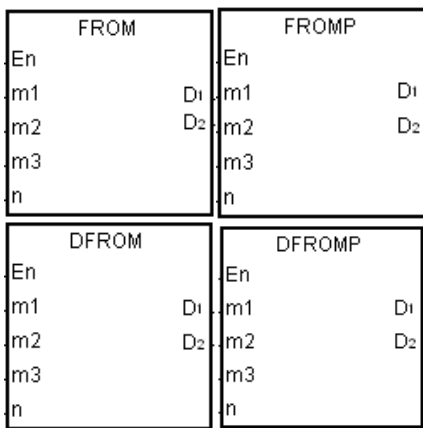
6.15.2 模块的数据读写指令说明

API	指令码			操作数							功能						
1400	D	FROM	P	$m_1 \cdot m_2 \cdot m_3 \cdot D_1 \cdot D_2 \cdot n$							特殊模块 CR 数据读出						

装置	X	Y	M	S	T	C	HC	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 (13 steps)	32 位指令 (13 steps)
AH500	AH500	AH500

符号：



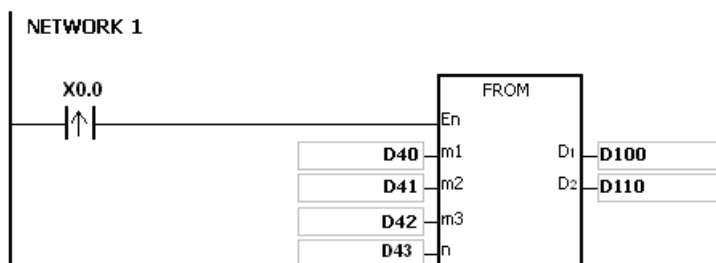
- m_1 : Rack 代码 Word/DWord
- m_2 : Slot 代码 Word/DWord
- m_3 : 欲读取特殊模块之 CR (Controlled Register) 编号 Word/DWord
- D_1 : 存放读取数据的位置 Word/DWord
- D_2 : 存放错误代码的位置 Word/DWord
- n : 一次读取之数据笔数 Word/DWord

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, $m_1=2\sim 8$ 代表延伸背板。
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) =0~11; 若 $m_1=2\sim 8$ (延伸背板), 则 $m_2=0\sim 7$ 。
4. m_3 : 欲读取特殊模块之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，读取放置于 CPU module 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. 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 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 module 与 I/O Module 的执行效能，所以不建议使用。
5. 错误代码说明

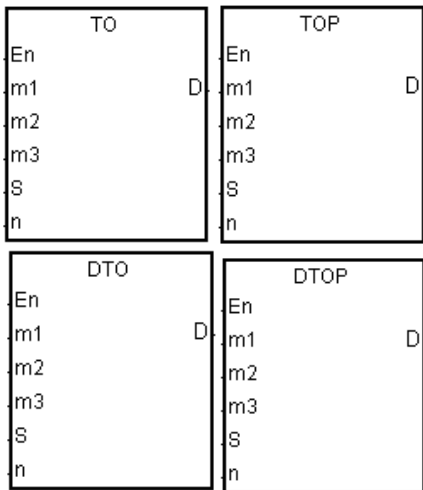
错误代码	说明
16#2003	请参考补充说明 1 跟 2
16#200B	请参考补充说明 3
16#1400	辅助处理器存取错误
16#1401	I/O 模块存取错误
16#1402	I/O 模块不符合 I/O 配置设定
16#1407	辅助处理器通讯错误

API	指令码			操作数								功能				
1401	D	TO	P	$m_1 \cdot m_2 \cdot m_3 \cdot S \cdot D \cdot n$								特殊模块 CR 数据写入				

装置	X	Y	M	S	T	C	HC	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 (13 steps)	32 位指令 (13 steps)
AH500	AH500	AH500

符号：



- m_1 : Rack 代码 Word/DWord
- m_2 : Slot 代码 Word/DWord
- m_3 : 欲写入特殊模块之 CR
 (Controlled Register) 编号 Word/DWord
- S : 写入 CR 的数据 Word/DWord
- D : 存放错误代码的位置 Word/DWord
- n : 一次写入之数据笔数 Word/DWord

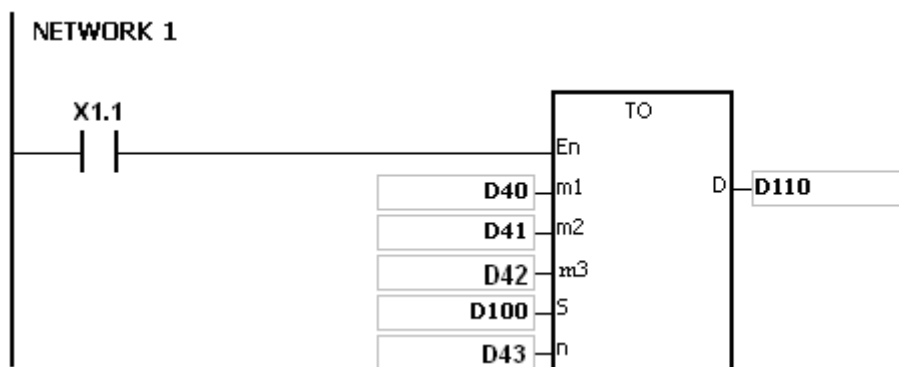
指令说明：

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, $m_1=2\sim 8$ 代表延伸背板。
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 $m_1=2\sim 8$ (延伸背板) , 则 $m_2=0\sim 7$ 。
4. m_3 : 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, $n=1\sim 256$; 32 位指令中, $n=1\sim 128$ 。
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 module。例如：S 为 16#0001，n 为 3，则传送三个 16#0001 给 I/O Module。

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 module 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。

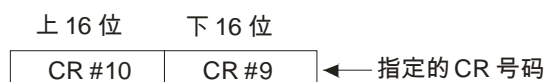


各参数使用说明如下：

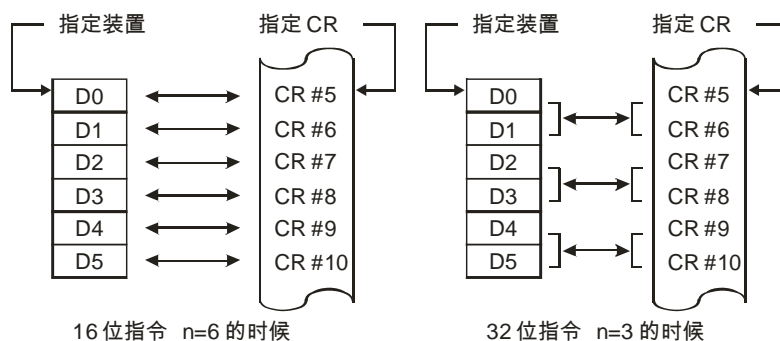
- 模块位于主背板，因此背板（Rack）编号 D40 设定为 16#0001。
- 模块放置在第一个插槽，因此插槽（Slot）编号 D41 设定为 16#0000。
- 模块之 COM1 数据交换读取触发设定值为 CR#7，因此 CR 编号 D42 设定为 16#0007。
- 模块之 COM1 数据交换读取触发设定值只占用一个寄存器，因此写入笔数 D43 设定为 1。
- 要写入模块 CR#0007 的数据会储存在 D100 里，因此 D100 设定为 16#0002。

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， $m_1=2\sim 8$ 代表延伸背板。
- 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 $m_1=2\sim 8$ （延伸背板），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. $m1$ 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 $S\sim S+n-1$ 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 module 与 I/O Module 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	辅助处理器存取错误
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	辅助处理器通讯错误

6.16 浮点数指令

6.16.1 浮点数指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
1500	–	FSIN	✓	浮点数 SIN 运算	5-6
1501	–	FCOS	✓	浮点数 COS 运算	5-6
1502	–	FTAN	✓	浮点数 TAN 运算	5-6
1503	–	FASIN	✓	浮点数 ASIN 运算	5-6
1504	–	FACOS	✓	浮点数 ACOS 运算	5-6
1505	–	FATAN	✓	浮点数 ATAN 运算	5-6
1506	–	FSINH	✓	浮点数 SINH 运算	5-6
1507	–	FCOSH	✓	浮点数 COSH 运算	5-6
1508	–	FTANH	✓	浮点数 TANH 运算	5-6
1509	–	FRAD	✓	角度→弧度	5-6
1510	–	FDEG	✓	弧度→角度	5-6
1511	SQR	DSQR	✓	BIN 开平方根	5
1512	–	FSQR	✓	浮点数开平方根	5-6
1513	–	FEXP	✓	浮点数取指数	5-6
1514	–	FLOG	✓	浮点数取对数	7-9
1515	–	FLN	✓	浮点数取自然对数	5-6
1516	–	FPOW	✓	浮点数权值指令	7-9
1517	RAND	–	✓	随机数值	7
1518	BSQR	DBSQR	✓	BCD 开平方根	5
1519	–	BSIN	✓	BCD SIN 运算	5
1520	–	BCOS	✓	BCD COS 运算	5
1521	–	BTAN	✓	BCD TAN 运算	5
1522	–	BASIN	✓	BCD ASIN 运算	5
1523	–	BACOS	✓	BCD ACOS 运算	5
1524	–	BATAN	✓	BCD ATAN 运算	5

6.16.2 浮点数指令说明

API	指令码		操作数				功能			
1500	FSIN	P	S · D				浮点数 SIN 运算			

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

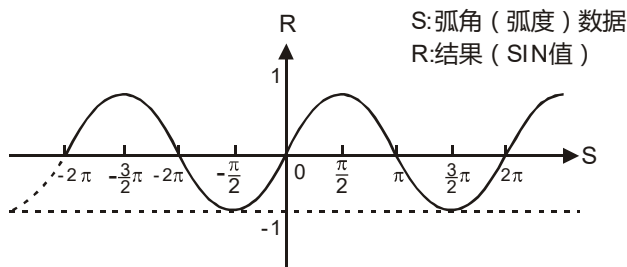
脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

FSIN	FSINP	S : 指定的来源值 Double Word
En	En	
S	D	D : 取 SIN 值结果 Double Word

指令说明：

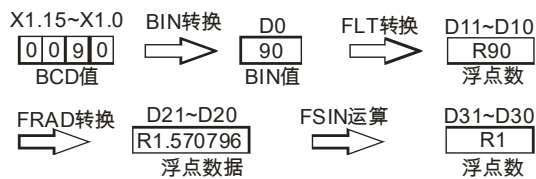
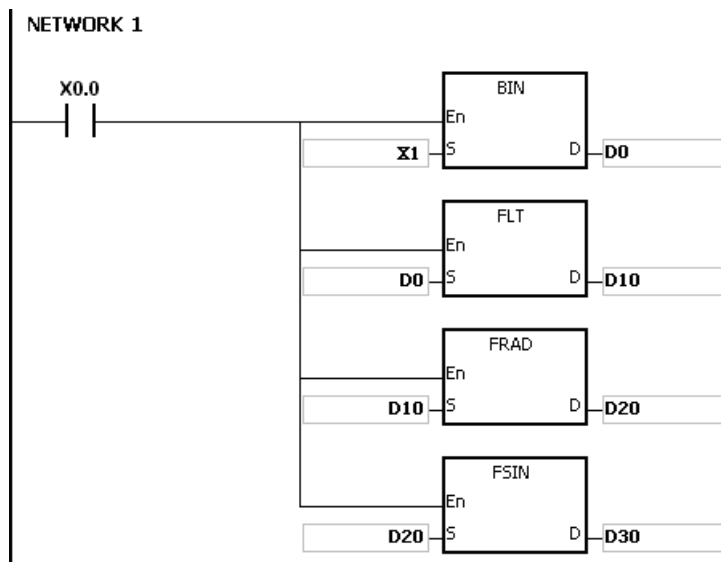
1. S 所指定的来源可为弧度或角度，由标志 SM695 决定。
2. 当 SM695=OFF 时，指定为弧度模式，弧度值=角度×π/180。
3. 当 SM695=ON 时，指定为角度模式，角度值=弧度×180/π，角度范围：0°≤角度值≤360°。
4. 若转换结果为 0，则零标志 SM600=ON。
5. 将 S 所指定的来源值，求取 SIN 值后存于 D 所指定的寄存器当中。
6. 下图显示弧角（弧度、弧度）与结果的关系：



6

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 运算转成浮点数值后存于 (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：

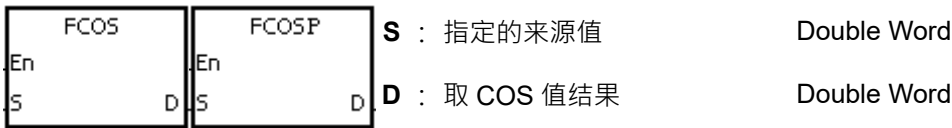
1. 当 **S** 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。
2. 当 SM695=ON 且 **S** 的内容值小于 0 或大于 360 指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码			操作数							功能				
1501		FCOS	P	S · D							浮点数 COS 运算				

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

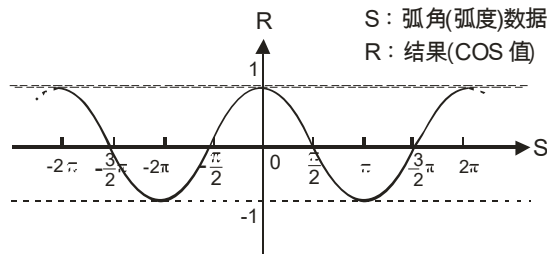
脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



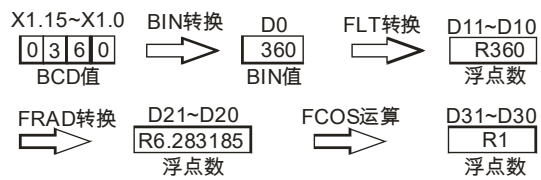
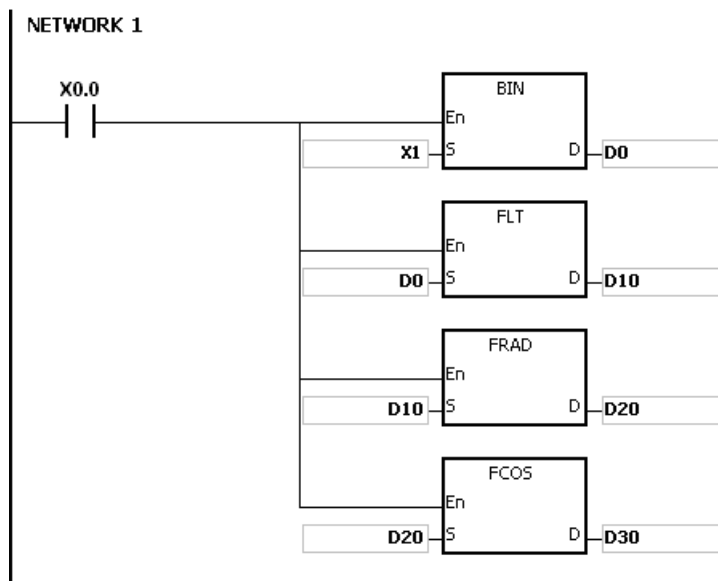
指令说明：

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，则零标志 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. 下图显示弧角与结果的关系：



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 运算转成浮点数值后存于 (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。

API	指令码		操作数				功能						
1502		FTAN	P	S · D				浮点数 TAN 运算					

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

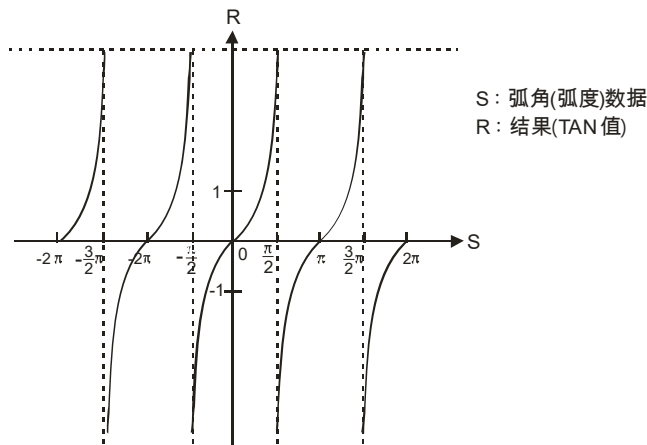


S：指定的来源值 Double Word

D：取 TAN 值结果 Double Word

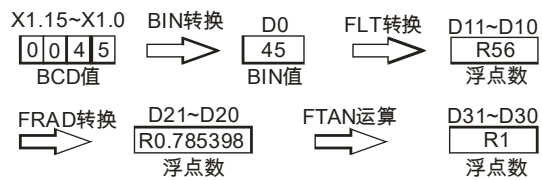
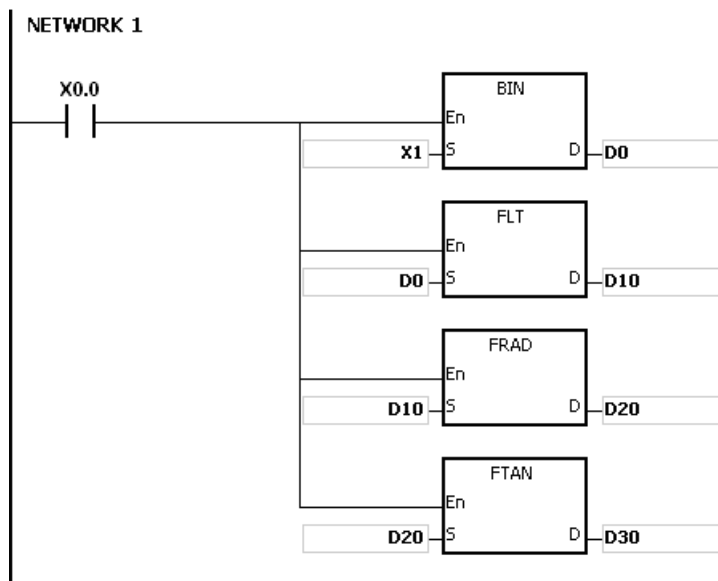
指令说明：

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，则零标志 **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. 下图显示弧角与结果的关系：



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 运算转成浮点数值后存于 (**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。

API	指令码		操作数				功能						
1503		FASIN	P	S · D				浮点数 ASIN 运算					

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

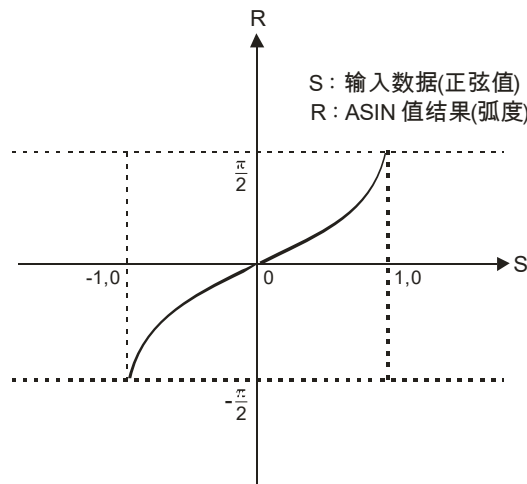


S：指定的来源值 Double Word

D：取 ASIN 值结果 Double Word

指令说明：

1. ASIN 值 = \sin^{-1} · 下图显示输入数据与结果的关系：

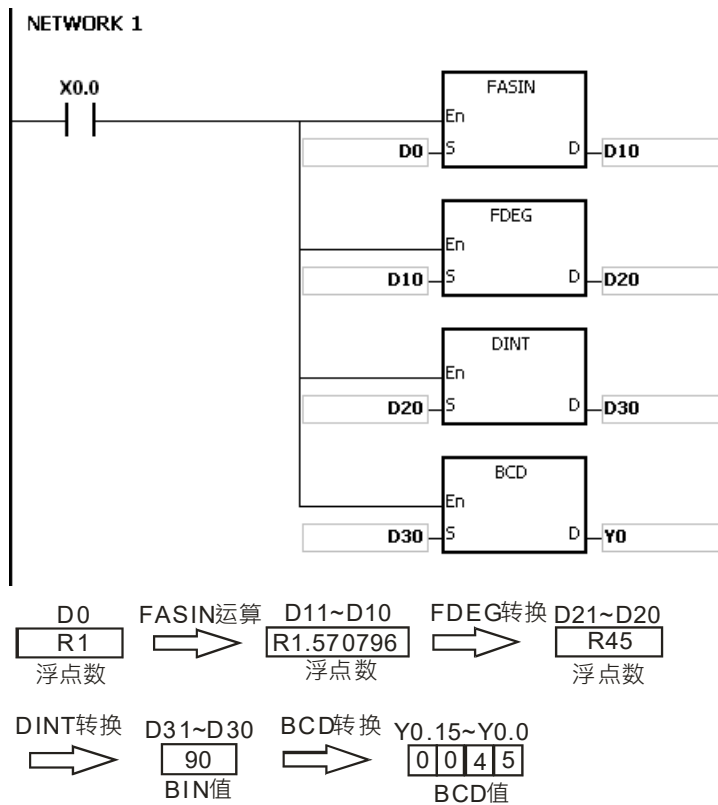


2. 若转换结果为 0，则零标志 SM600=ON。

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。

6



补充说明：

1. **S** 操作数指定的正弦值数值之十进浮点值只能介于-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。

API	指令码		操作数				功能						
1504		FACOS	P	S · D				浮点数 ACOS 运算					

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

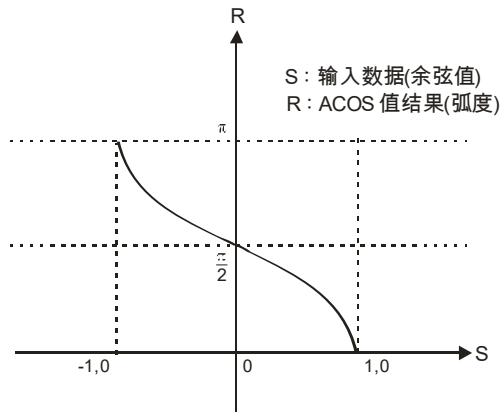


S：指定的来源值 Double Word

D：取 ACOS 值结果 Double Word

指令说明：

1. ACOS 值 = \cos^{-1} ，下图显示输入数据与结果的关系：

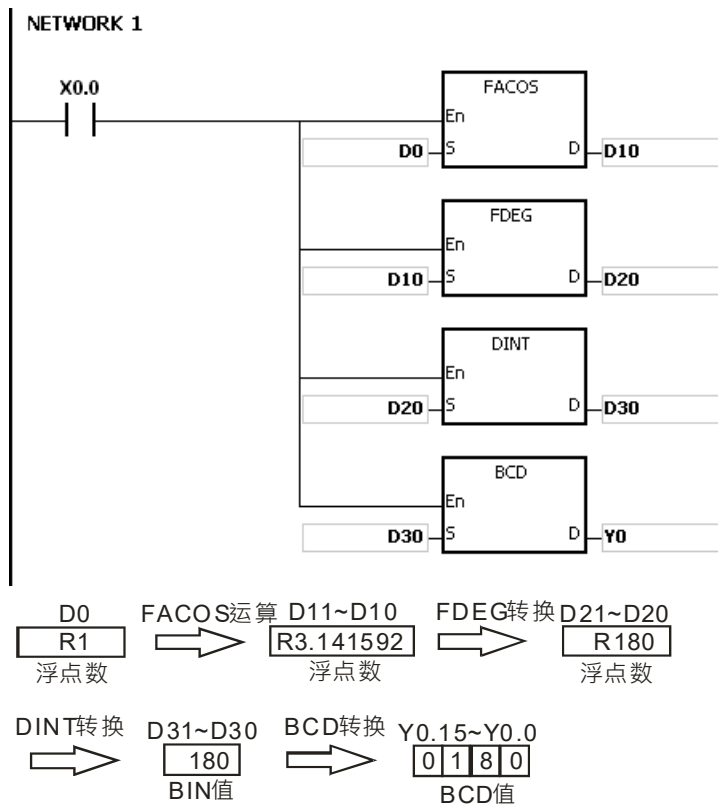


- 若转换结果的绝对值大于可表示之最大浮点值，则进位标志 SM602=ON。
- 若转换结果的绝对值小于可表示之最小浮点值，则借位标志 SM601=ON。
- 若转换结果为 0，则零标志 SM600=ON。

程序范例：

当 X0.0=ON 时，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。

6



补充说明：

1. **S** 操作数指定的正弦值数值之十进浮点值只能介于-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。

API	指令码			操作数							功能						
1505		FATAN	P	S · D							浮点数 ATAN 运算						

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



S：指定的来源值

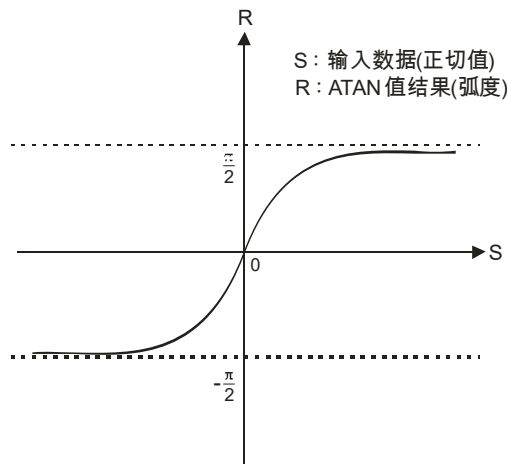
Double Word

D：取 ATAN 值结果

Double Word

指令说明：

1. ATAN 值= \tan^{-1}
2. 下图显示输入数据与结果的关系：

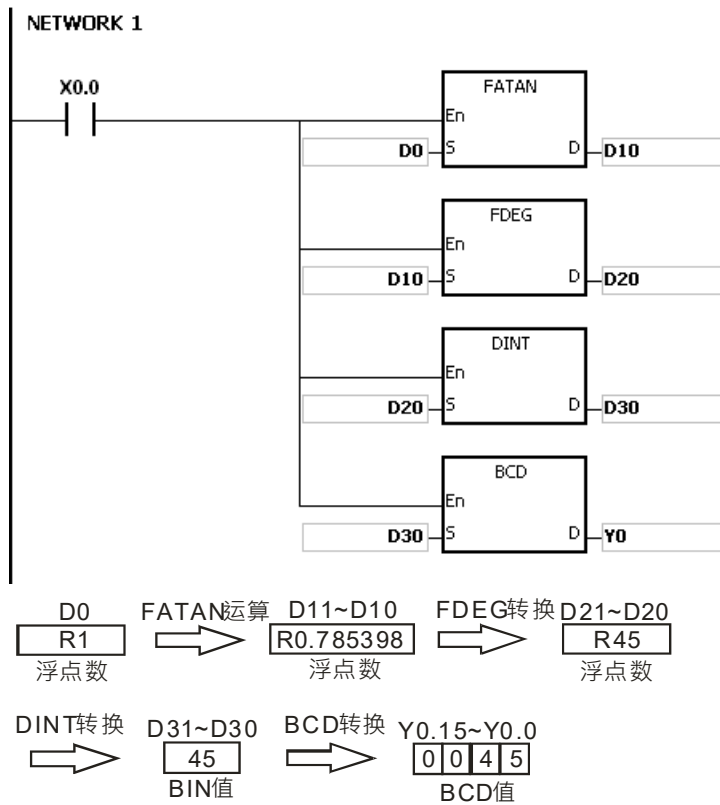


3. 若转换结果为 0，则零标志 SM600=ON。

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。

6



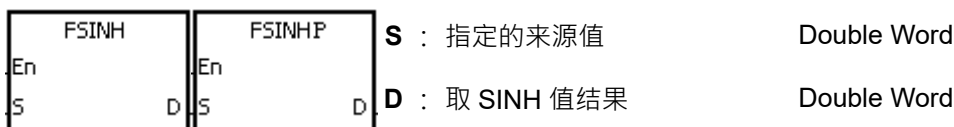
补充说明：

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。

API	指令码			操作数							功能						
1506		FSINH	P	S · D							浮点数 SINH 运算						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



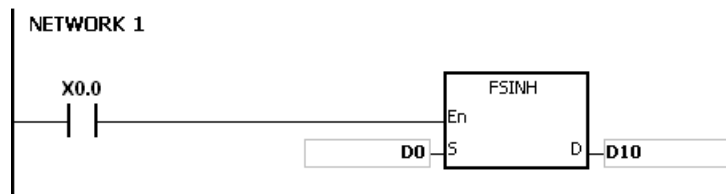
指令说明：

1. SINH 值 = $(e^s - e^{-s}) / 2$ 。
2. 若转换结果的绝对值大于可表示之最大浮点值，则 D=16#7F800000，进位标志 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，借位标志 SM601=ON。
4. 若转换结果为 0，则零标志 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。

6



S

D1	D0
----	----

 单精度浮点数



D

D11	D10
-----	-----

 FSINH值单精度浮点数

2. 若转换结果的绝对值大于可表示之最大浮点值，则进位标志 SM602=ON。
3. 若转换结果的绝对值小于可表示之最小浮点值，则借位标志 SM601=ON。
4. 若转换结果为 0，则零标志 SM600=ON。

补充说明：

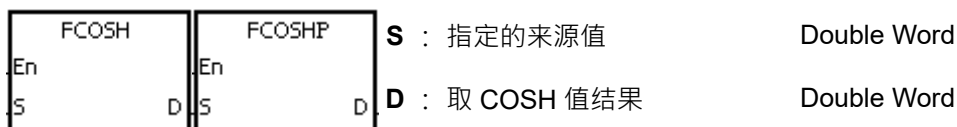
当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

API	指令码			操作数								功能					
1507	FCOSH	P		S · D								浮点数 COSH 运算					

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

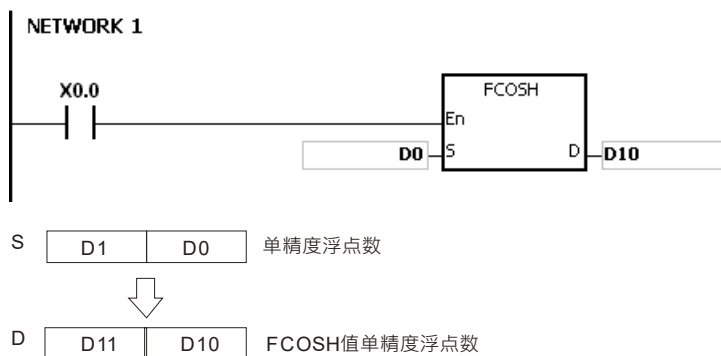


指令说明：

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，进位标志 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，借位标志 SM601=ON。
4. 若转换结果为 0，则零标志 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之最大浮点值，则进位标志 SM602=ON。
3. 若转换结果的绝对值小于可表示之最小浮点值，则借位标志 SM601=ON。
4. 若转换结果为 0，则零标志 SM600=ON。

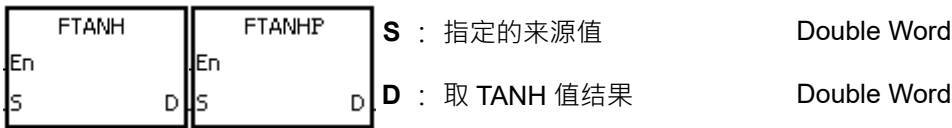
补充说明：

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。

API	指令码			操作数							功能						
1508		FTANH	P	S · D							浮点数 TANH 运算						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



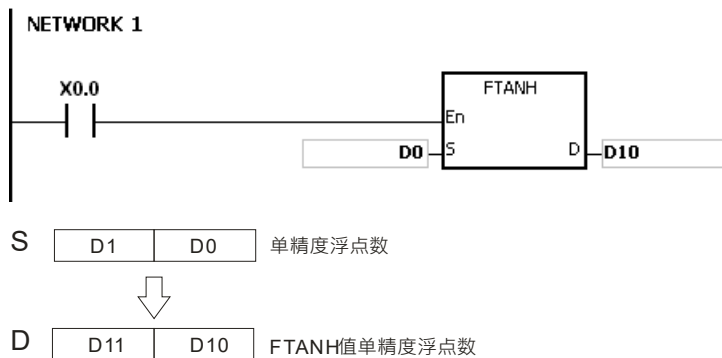
指令说明：

1. TANH 值 = $(e^s - e^{-s}) / (e^s + e^{-s})$ 。
2. 若转换结果为 0，则零标志 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。

6



2. 若转换结果为 0，则零标志 SM600=ON。

补充说明：

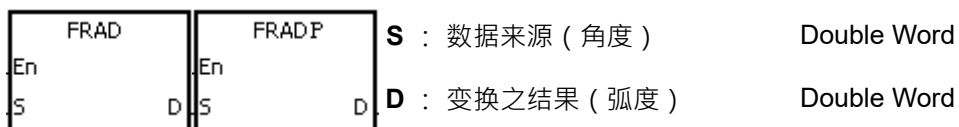
当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

API	指令码			操作数								功能					
1509		FRAD	P	S · D								角度→弧度					

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

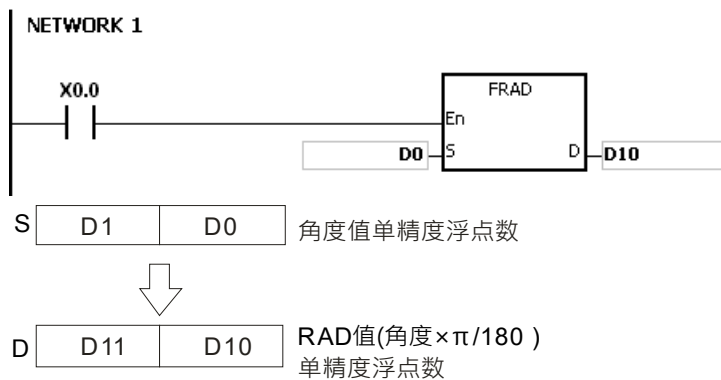


指令说明：

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，则零标志 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。



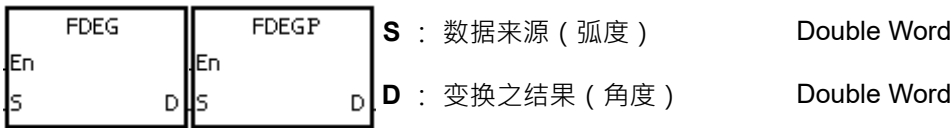
补充说明：

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。

API	指令码			操作数							功能						
1510		FDEG	P	S · D							弧度→角度						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



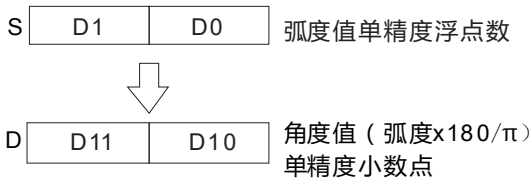
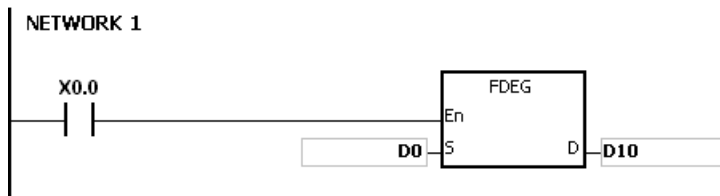
指令说明：

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之最大浮点值，则 D=16#7F7FFFFF。
4. 若转换结果的绝对值小于可表示之最小浮点值，则 D=16#FF7FFFFF。
5. 若转换结果为 0，则零标志 SM600=ON。

程序范例：

当 X0.0=ON 时，指定浮点数 (D1 · D0) 之角度值，将弧度值转换成角度后存于 (D11 · D10) 当中，内容为浮点数。

6



补充说明：

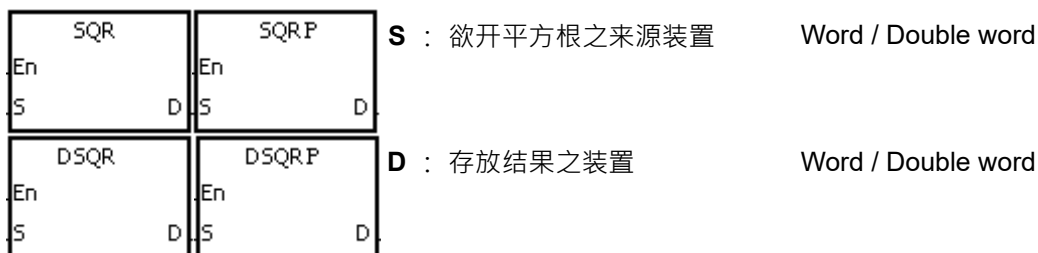
当 S 的内容值超出浮点数可以表示的范围时，指令不执行，SM0=ON，错误码 SR0=16#2013。

API	指令码			操作数								功能				
	1511	D	SQR	P	S · D								开平方根			

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：

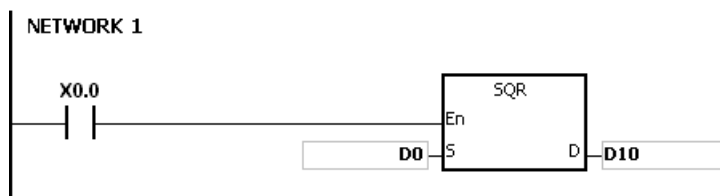


指令说明：

1. 将 **S** 所指定之装置内容值开平方根后，存放于 **D** 所指定之装置。
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。



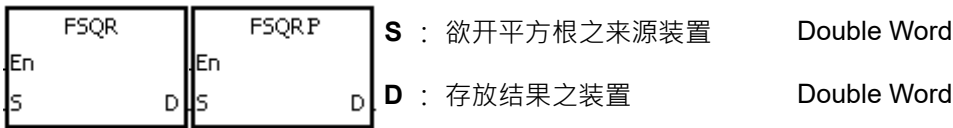
补充说明：

1. **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。

API	指令码			操作数							功能						
1512		FSQR	P	S · D							浮点数开平方根						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



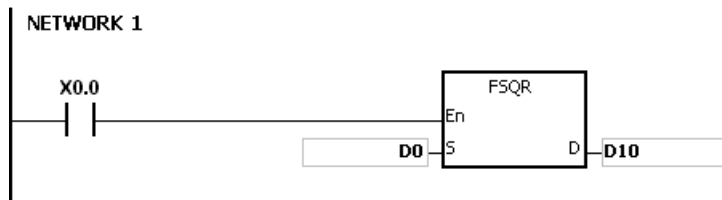
指令说明：

1. 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。
2. 若运算结果为 0，则零标志 SM600=ON。

程序范例一：

当 X0.0=ON 时，将浮点数 (D1 · D0) 取开平方根，将结果存放至 (D11 · D10) 所指定的寄存器当中。

6



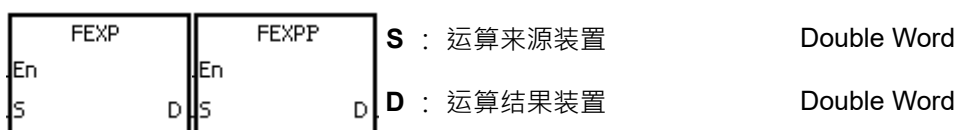
补充说明：

1. 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。

API	指令码			操作数								功能					
1513		FEXP	P	S · D								浮点数取指数					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：

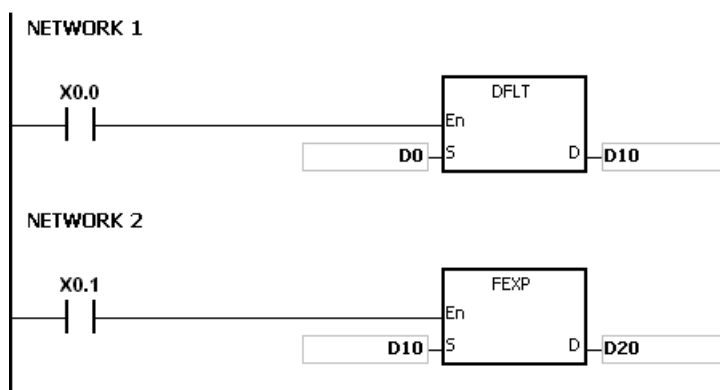


指令说明：

1. 以 $e=2.71828$ 为底数，S 为指数做 EXP 运算。
2. $EXP[D+1 \cdot D]=[S+1 \cdot S]$
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 需转换为浮点数值。
4. D 操作数内容值= e^S ； $e=2.71828$ ，S 为指定的来源数据。
5. 若转换结果的绝对值大于可表示之最大浮点值，则 $D=16\#7F800000$ ，进位标志 $SM602=ON$ 。
6. 若运算结果为 0，则零标志 $SM600=ON$ 。

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，其值为浮点数值并存放于 (D21 · D20) 寄存器中。



API	指令码		操作数				功能						
1514		FLOG	P	$S_1 \cdot S_2 \cdot D$				浮点数取对数					

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

脉冲执行型	16 位指令 (7-9 steps)	32 位指令
AH500	AH500	-

符号：

FLOG	FLOGP	S_1 : 运算底数装置 Double Word S_2 : 运算来源装置 Double Word D : 运算结果装置 Double Word
En	En	
S1 D	S1 D	
S2	S2	

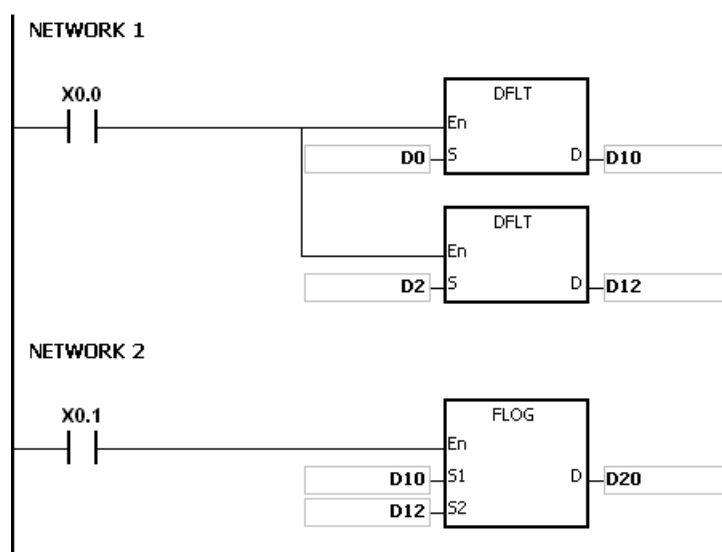
指令说明：

1. 将 S_1 内容及 S_2 内容为操作数做 log 运算，结果存放于 D。
2. S_1 、 S_2 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 S_1 、 S_2 需转换为浮点数值。
3. $S_1^D = S_2$ ，求 D 值 $\rightarrow \text{Log}_{S_1} S_2 = D$
4. 例：已知 $S_1=5$ ， $S_2=125$ ，求 $D = \log_5 125 = ?$
5. $S_1^D = S_2 \rightarrow 5^D = 125 \rightarrow D = \log_5 125 = 3$
6. 若运算结果为 0，则零标志 $SM600 = ON$ 。

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 位寄存器浮点数值做 Log 运算并将结果存于 (D21 · D20) 32 位寄存器中。

6



补充说明：

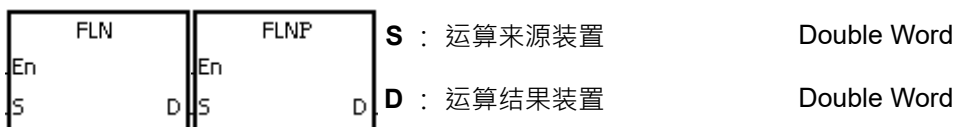
1. S_1 内容值 ≤ 1 或 $S_2 \leq 0$ 时，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码			操作数							功能						
1515		FLN	P	S · D							浮点数取自然对数						

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

脉冲执行型	16 位指令 (5-6 steps)	32 位指令
AH500	AH500	-

符号：



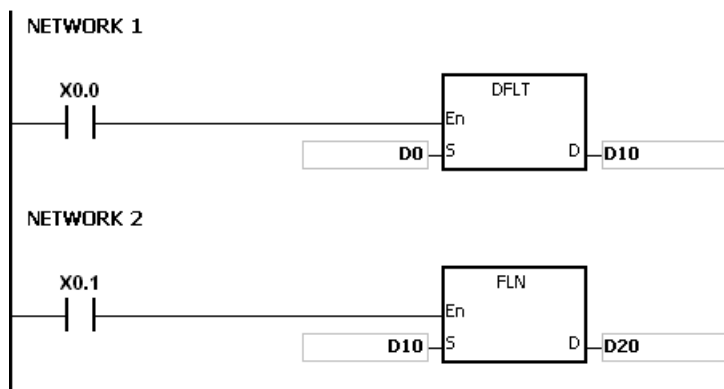
指令说明：

- 以 **S** 为操作数做自然对数 ln 运算。
- $LN[S+1 \cdot S]=[D+1 \cdot D]$
- 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** 需转换为浮点数值。
- $e D=S \rightarrow D$ 操作数内容值= $\ln S$ ；**S** 为指定的来源数据。
- 若运算结果为 0，则零标志 $SM600=ON$ 。

程序范例：

6

- 当 X0.0 为 ON 时，将 (D1 · D0) 值转成浮点数存于 (D11 · D10) 寄存器中。
- 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：

- $S \leq 0$ 时，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。

API	指令码		操作数							功能						
1516	FPOW	P	$S_1 \cdot S_2 \cdot D$							浮点数权值指令						

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

脉冲执行型	16 位指令 (7-9 steps)	32 位指令
AH500	AH500	-

符号：

FPOW		FPOWP		
En		En		S_1 : 底数装置
S1	D	S1	D	S_2 : 次幂数装置
S2		S2		D : 运算结果装置

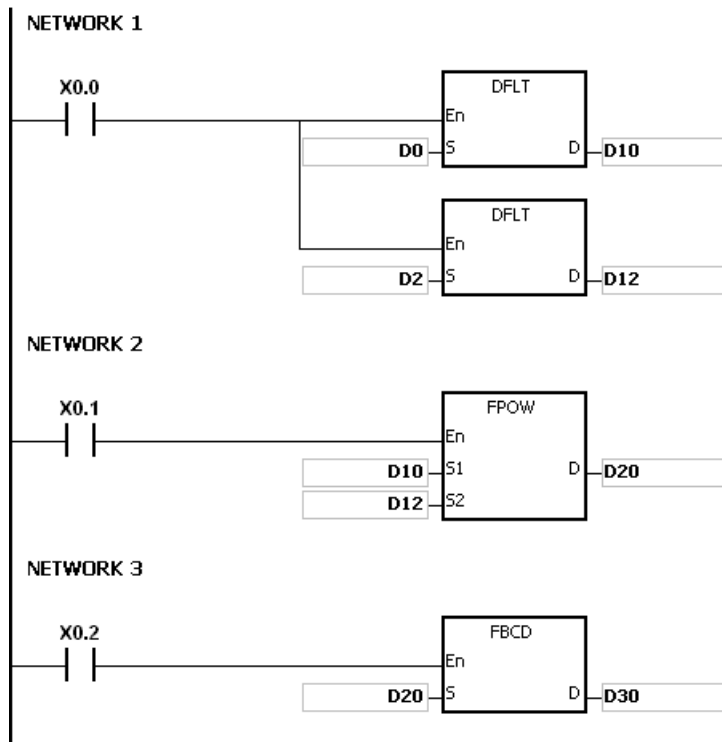
Double Word
Double Word
Double Word

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. $D=POW[S_1+1 \cdot S_1]^{[S_2+1 \cdot S_2]}$
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 需转换为浮点数值。
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，则进位标志 SM602=ON。
6. 若运算结果的绝对值小于可表示之最小浮点值，则借位标志 SM601=ON。
7. 若转换结果的绝对值大于可表示之最大浮点值，则 $D=16\#7F7FFFFFFF$ 。
8. 若转换结果的绝对值小于可表示之最小浮点值，则 $D=16\#FF7FFFFFFF$ 。
9. 若运算结果为 0，则零标志 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 位寄存器浮点数值转成十进浮点数值并存于 (D31 · D30) 寄存器中。(此时 D31 为表示 D30 的 10 次幂方)



补充说明：

1. $S_1 < 0$ 时，指令不执行，SM0=ON，错误码 SR0=16#2003。

6

API	指令码		操作数				功能						
1517	RAND	P	$S_1 \cdot S_2 \cdot D$				随机数值						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：

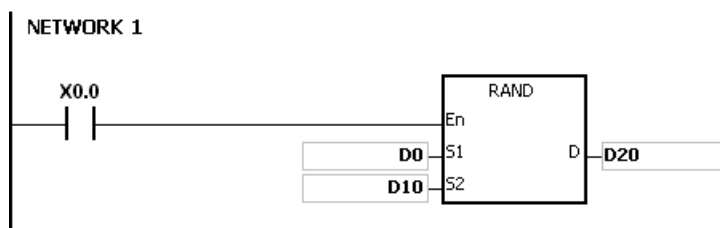
RAND		RANDP		
En		En		S_1 : 随机数产生的范围下限 Word
S1	D	S1	D	S_2 : 随机数产生的范围上限 Word
S2		S2		D : 随机数产生的结果 Word

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. 若使用者输入 $S_1 > S_2$ ，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：

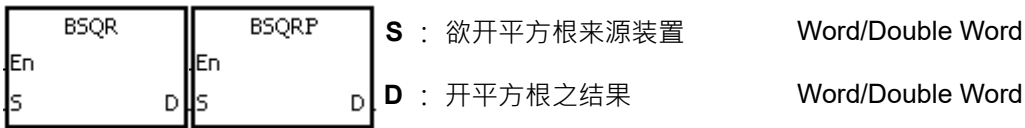
S_1 或 S_2 操作数内容值范围为 0~32767，内容值超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码			操作数							功能						
1518	D	BSQR	P	S · D							BCD 开平方根						

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：

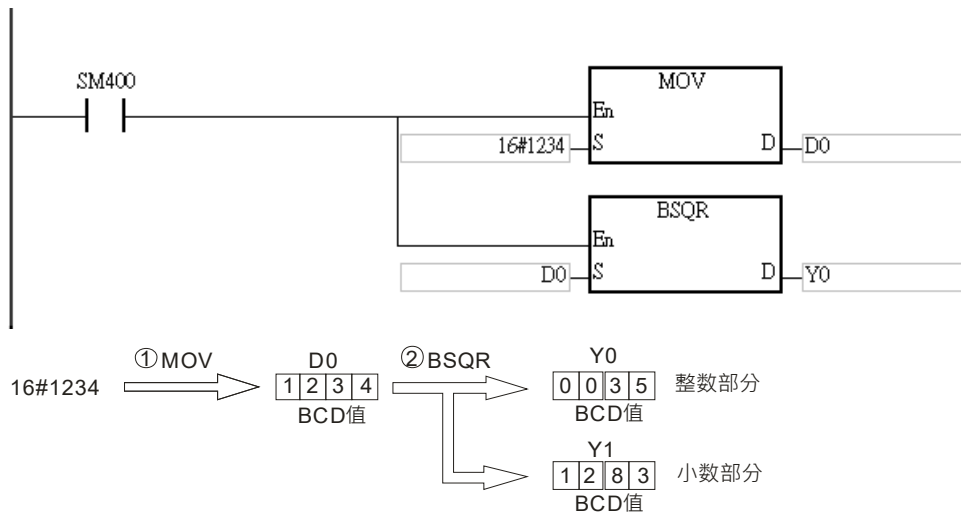


指令说明：

1. 将 **S** 所指定之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** 所指定之装置。
2. 数据来源 **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。

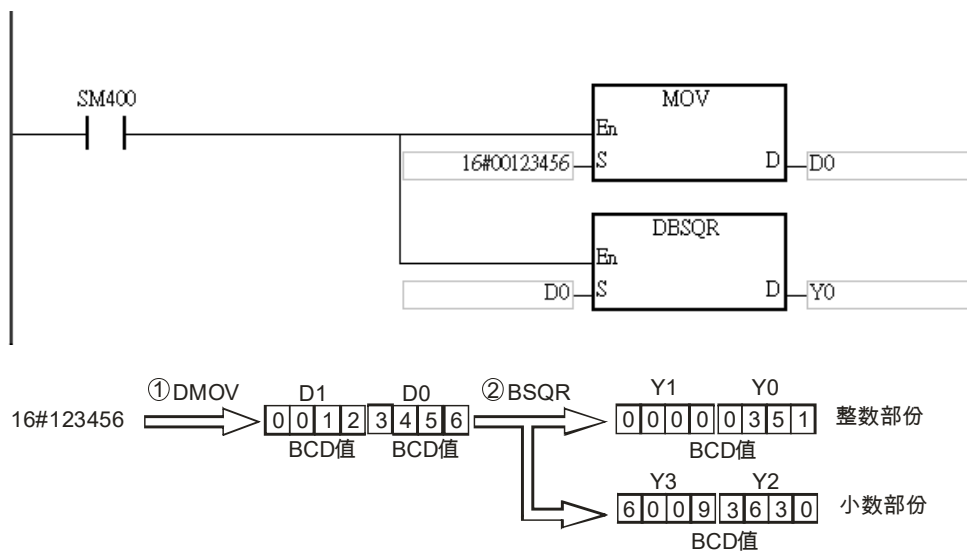
程序范例一：

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。



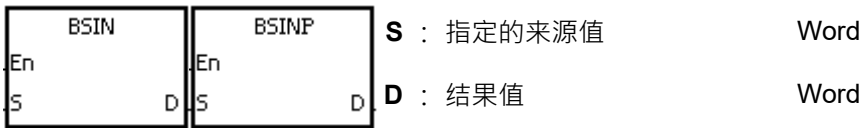
补充说明：

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。

API	指令码			操作数							功能						
1519		BSIN	P	S · D							BCD SIN 运算						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



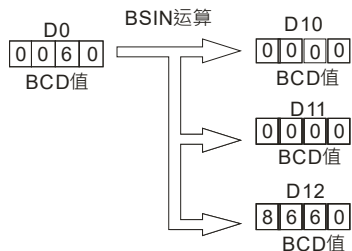
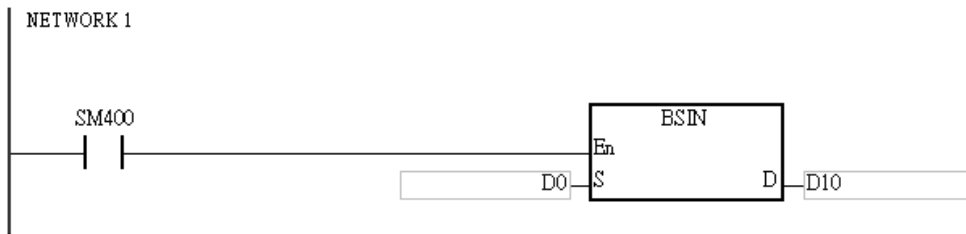
指令说明：

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，则零标志 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。

6



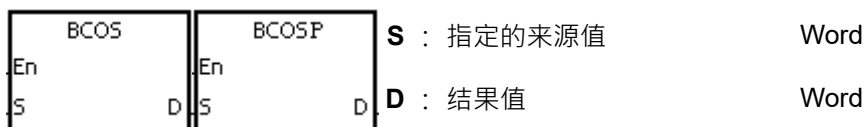
补充说明：

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。

API	指令码			操作数								功能					
1520		BCOS	P	S · D								BCD COS 运算					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

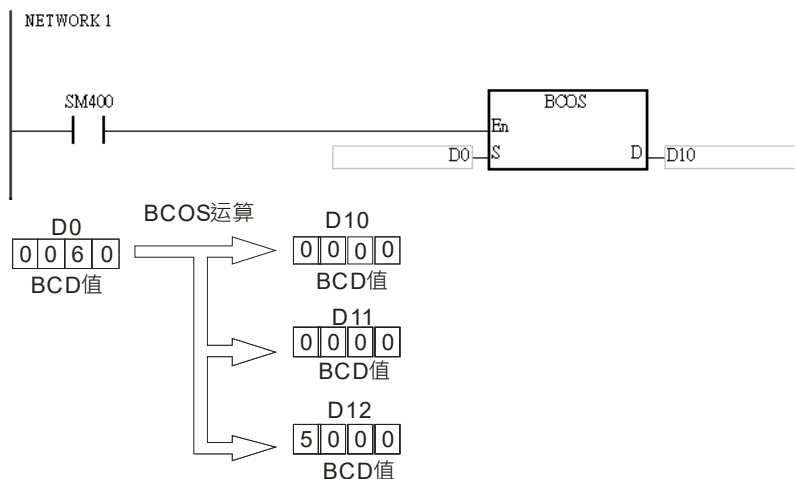


指令说明：

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° ≤ 角度值 < 360°。
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，则零标志 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。



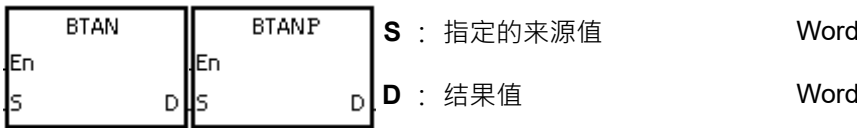
补充说明：

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. 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。

API	指令码			操作数							功能						
1521		BTAN	P	S · D							BCD TAN 运算						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

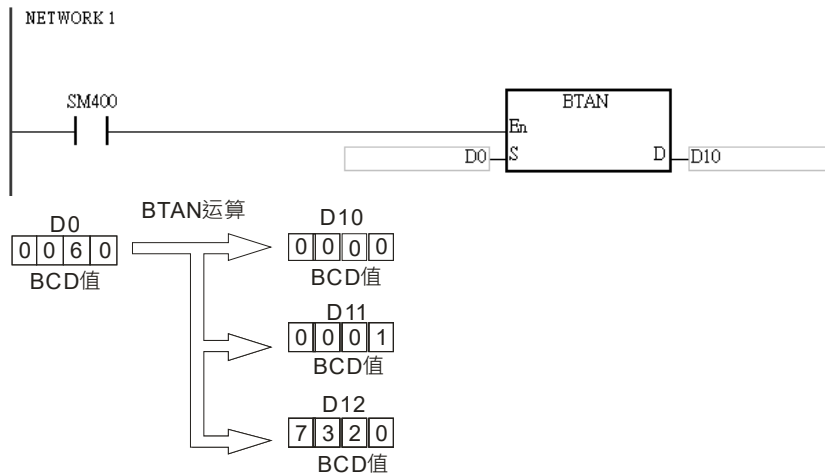


指令说明：

1. 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。
2. 角度范围：0° ≤ 角度值 < 360°。
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，则零标志 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。



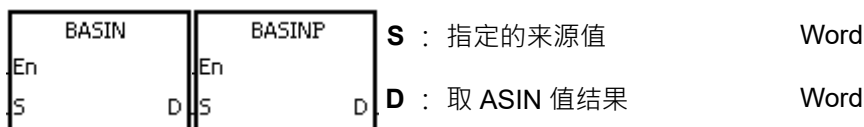
补充说明：

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。

API	指令码			操作数							功能						
1522	BASIN		P	S · D							BCD ASIN 运算						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



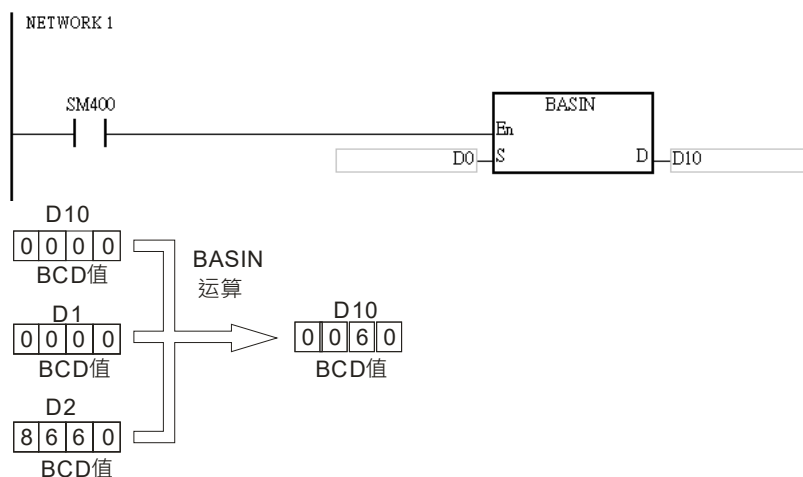
指令说明：

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} 之整数部分 · 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：

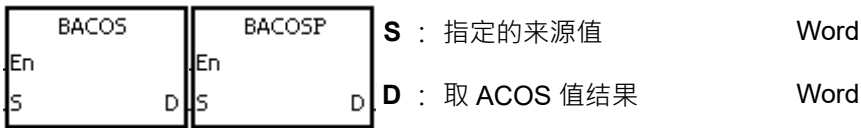
1. 小数部份输入注意事项：以 0.5 为例 · 输入时 · 需输入 S=0 · S+1=0 · S+2=16#5000 。
2. 当 S 的数据内容并非为 BCD 值 (以 Hex 表示有任一位数不在 0~9 的范围内) · 则将会产生运算错误 · SM0=ON · 错误码 SR0=16#200D 。
3. S 操作数指定的正弦值数值只能介 -1.0~+1.0 之间 · 若不在此范围 · 指令不执行 · SM0=ON · 错误码 SR0=16#2003 。
4. S 操作数 · 若使用 ISPSOft 宣告 · 则数据类型为 ARRAY [3] of WORD/INT 。

API	指令码				操作数								功能			
1523		BACOS	P		S · D								BCD ACOS 运算			

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

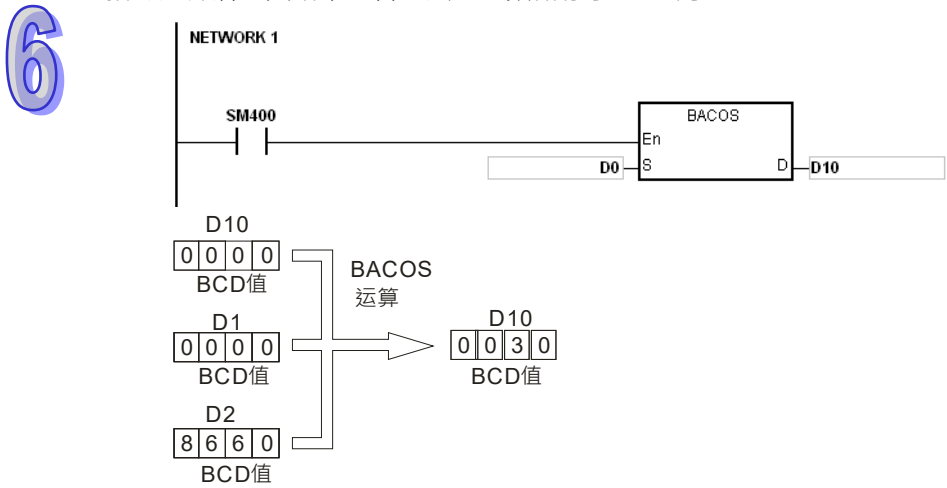


指令说明：

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。



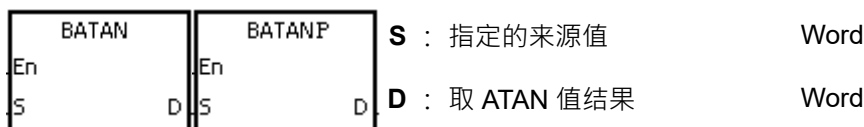
补充说明：

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。

API	指令码			操作数								功能					
1524	BATAN		P	S · D								BCD ATAN 运算					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：



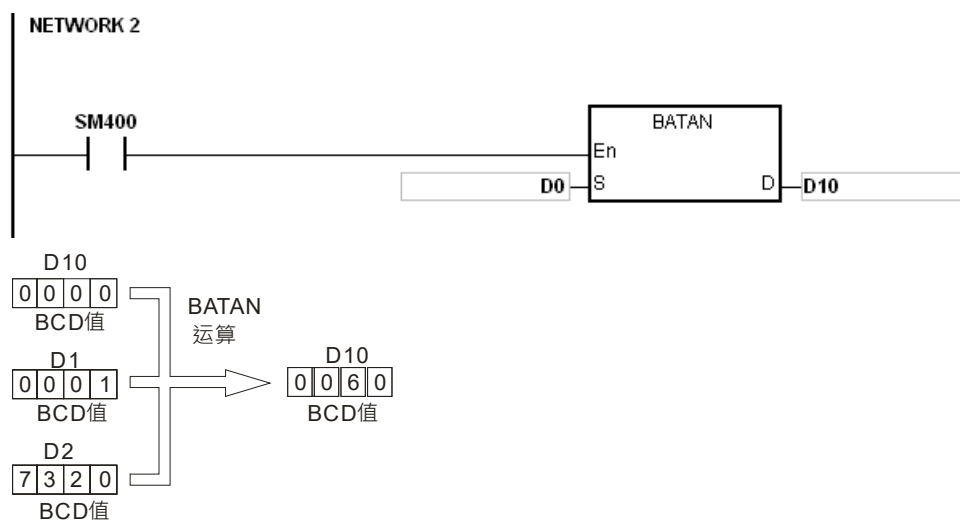
指令说明：

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} 之正负号 · D1 内容为 \tan^{-1} 之整数部分 · 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 · 若使用 ISPSOft 宣告 · 则数据类型为 ARRAY [3] of WORD/INT 。

6

6.17 万年历指令

6.17.1 万年历指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1600</u>	TRD	—	✓	万年历数据读出	3
<u>1601</u>	TWR	—	✓	万年历数据写入	3
<u>1602</u>	T+	—	✓	万年历数据加算	7
<u>1603</u>	T-	—	✓	万年历数据减算	7
<u>1604</u>	HOURL	DHOURL	—	运转定时器	7
<u>1605</u>	TCMP	—	✓	万年历数据比较	11
<u>1606</u>	TZCP	—	✓	万年历数据区域比较	9

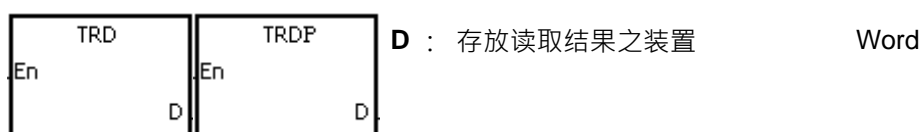
6.17.2 万年历指令说明

API	指令码			操作数							功能			
1600		TRD	P	D							万年历数据读出			

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

脉冲执行型	16 位指令 (3 steps)	32 位指令
AH500	AH500	-

符号：

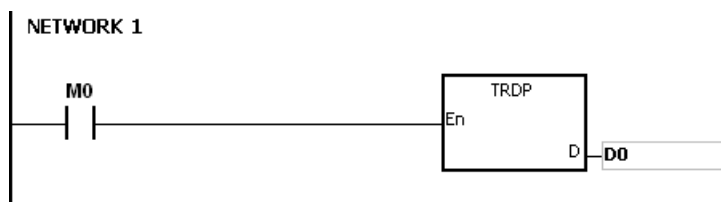


指令说明：

1. D : 万年历当前时间读出后存放之装置。
2. D 操作数会占用连续 7 个装置。
3. 主机内建万年历时钟共提供年、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 之内容 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
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

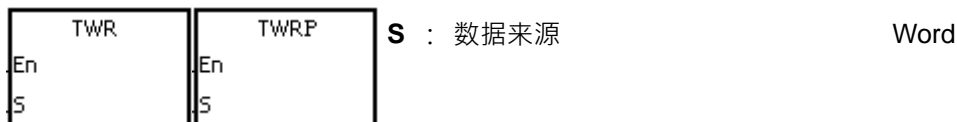
补充说明：

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**。

6

API	指令码			操作数							功能						
1601		TWR	P	S							万年历数据写入						
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
脉冲执行型											16 位指令 (3 steps)			32 位指令			
AH500											AH500			-			

符号：



指令说明：

1. **S**：存放欲写入万年历新设定值之装置。
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：

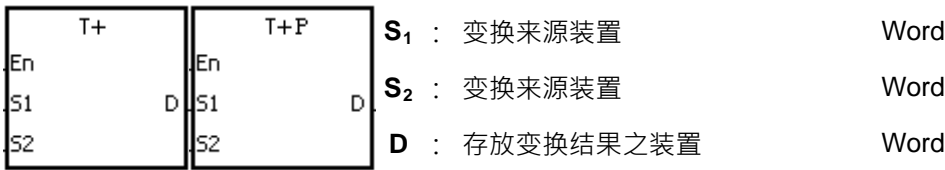
1. 若 **S** 内容值超出范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 若 **S+6** 超出装置范围，则视为运算错误，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. **S** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [7] of WORD/INT。

API	指令码			操作数							功能						
1602		T+	P	S₁ · S₂ · D							万年历数据加算						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



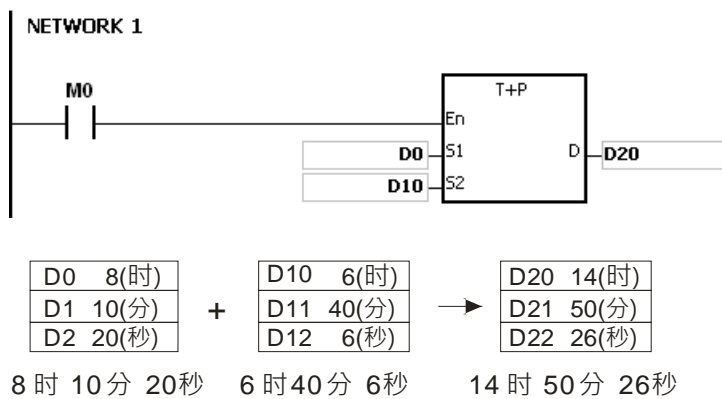
指令说明：

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. 加算结果若大于等于24小时的话，进位标志SM602=ON、D显示加算总值减掉24小时所得的结果。
4. 加算结果若是等于0(0时0分0秒)，零标志SM600=ON。

6

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之时、分、秒。



补充说明：

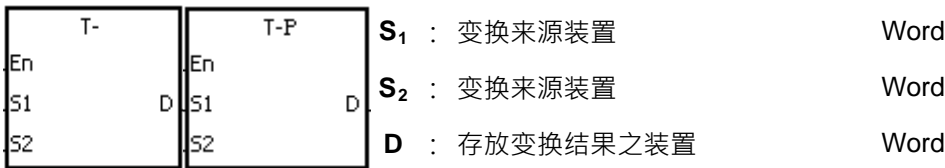
1. 若 S_1 、 S_2 内容值超出范围，则视为运算错误，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
2. 若 S_1+2 、 S_2+2 、 $D+2$ 超出装置范围，则视为运算错误，指令不执行， $SM0=ON$ ，错误码 $SR0=16\#2003$ 。
3. S_1 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
4. S_2 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [3] of WORD/INT。
5. D 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

API	指令码			操作数							功能				
1603		T-	P	$S_1 \cdot S_2 \cdot D$							万年历数据减算				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：



指令说明：

1. 将 S_1 所指定的万年历数据时、分、秒减掉 S_2 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. S_1 、 S_2 、D操作数，皆连续占用3个装置。
3. 减算结果若为负数时，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，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。

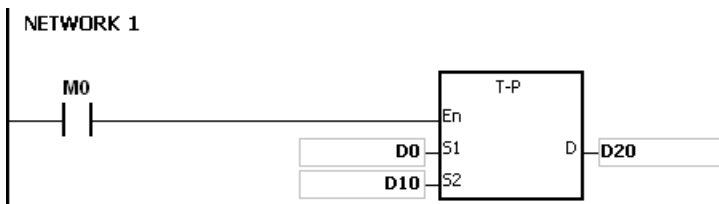


20时 20分 5秒 14时 30分 8秒 5时 49分 57秒

2. 减算结果若为负数时，借位标号 SM601=ON。



5时 20分 30秒 19时 11分 15秒 10时 9分 15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。

API	指令码		操作数				功能					
1604	D	HOUR	S · D ₁ · D ₂				运转定时器					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
-	AH500	AH500

符号：

HOUR		D HOUR		S	: 设定导通时间	Word/Double Word
En		En		D ₁	: 测量中之当前时间值	Word/Double Word
S	D1	S	D1	D ₂	: 输出装置	bit
	D2		D2			

指令说明：

1. S：设定导通时间，单位：小时。D₁：测量中之当前时间值，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。

测量中之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) 为测量中之当前时间值（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。

测量中之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 D₁、D₁+1、D₁+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。

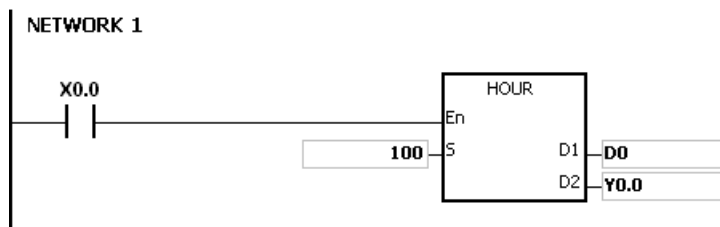
6. 当输出装置导通后，定时器会继续计时。

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。

6

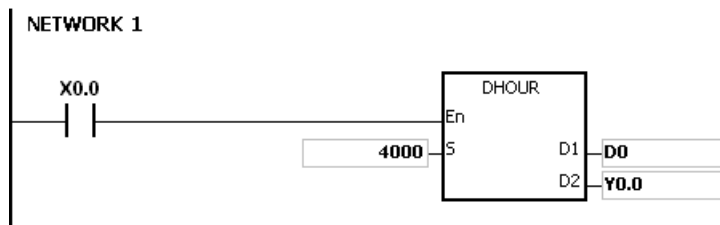
程序范例一：

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 会记录测量中之当前时间值（单位：小时），D1 会记录测量中不足 1 小时之当前时间值 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。



程序范例二：

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 会记录测量中之当前时间值（单位：小时），D2 会记录测量中不足 1 小时之当前时间值 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. D HOUR 中 $(D_1 + 1, D_1) < 0$ 时指令不执行，输出装置状态不变。
5. D HOUR 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。

API	指令码		操作数							功能							
1605		TCMP	P	$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot D$							万年历数据比较						

装置	X	Y	M	S	T	C	HC	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 (11 steps)	32 位指令
AH500	AH500	-

符号：

<table border="1"> <tr><th colspan="2">TCMP</th></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	TCMP		En		S1	D	S2		S3		S		<table border="1"> <tr><th colspan="2">TCMPP</th></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	TCMPP		En		S1	D	S2		S3		S		S_1 : 设定比较时间之“时”，设定范围为「K0~K23」 Word
	TCMP																									
	En																									
	S1	D																								
	S2																									
S3																										
S																										
TCMPP																										
En																										
S1	D																									
S2																										
S3																										
S																										
		S_2 : 设定比较时间之“分”，设定范围为「K0~K59」 Word																								
		S_3 : 设定比较时间之“秒”，设定范围为「K0~K59」 Word																								
		S : 万年历当前时间 Word																								
		D : 比较结果 bit																								

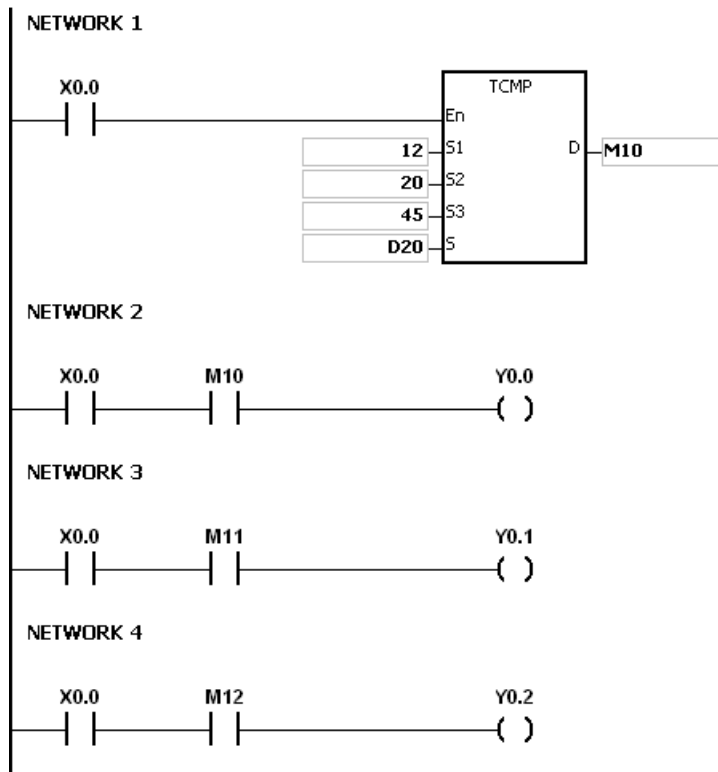
指令说明：

1. 将由 $S_1 \sim S_3$ 所指定的时、分、秒设定值与 S 起始之万年历时、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，指令不被执行，但是 M10~M12 之前的 ON/OFF 状态仍被保持住。
2. 若需要得到 \geq 、 \leq 、 \neq 之结果时，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。

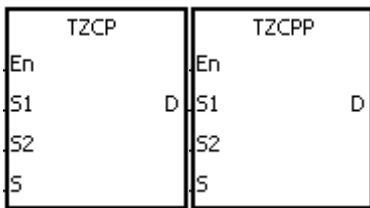
6

API	指令码		操作数				功能						
1606		TZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$				万年历数据区域比较					

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：



- S_1 : 设定比较时间之下限值 Word
- S_2 : 设定比较时间之上限值 Word
- S : 万年历当前时间 Word
- D : 比较结果 bit

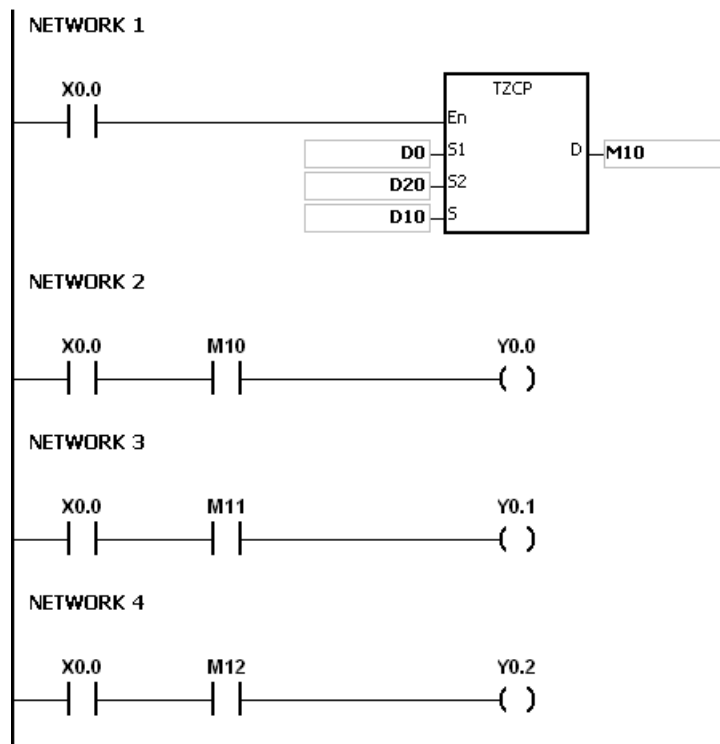
指令说明：

1. 将由 S 所指定的万年历当前时间时、分、秒值与 S_1 所指定设定比较时间之下限值及 S_2 所指定设定比较时间之上限值做区域比较，其比较结果在 D 作表示。
2. S_1 、 S_1+1 、 S_1+2 ：设定比较时间下限值的“时”、“分”、“秒”。
3. S_2 、 S_2+1 、 S_2+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 S_1 时间须比 S_2 时间小。当 $S_1 > S_2$ ，指令执行时把 S_1 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. 当当前时间 S 小于下限值 S_1 且 S 小于上限值 S_2 时，则 D 为ON，当当前时间 S 大于下限值 S_1 且 S 大于上限值 S_2 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 其中之一会 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。

6.18 外围设备指令

6.18.1 外围设备指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1700</u>	TKY	DTKY	—	10 键键盘输入	7
<u>1701</u>	HKY	DHKY	—	16 键键盘输入	9
<u>1702</u>	DSW	—	—	指拨开关输入	9
<u>1703</u>	ARWS	—	—	箭头键盘输入	9
<u>1704</u>	SEGL	—	—	七段显示器扫描输出	7

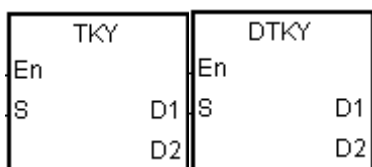
6.18.2 外围设备指令说明

API	指令码			操作数									功能				
1700	D	TKY		S · D ₁ · D ₂									10 键键盘输入				

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

脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
-	AH500	AH500

符号：



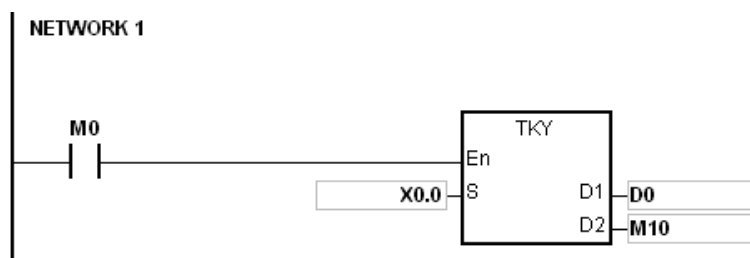
S : 按键输入起始装置 Bit
D₁ : 按键输入值存放处 Word/Double Word
D₂ : 按键输出信号 Bit

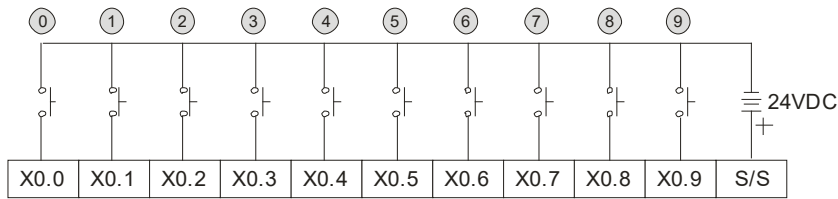
指令说明：

1. 本指令指定由 **S** 开始之 10 个外部输入点，依序代表 10 进位数字之 0~9。这 10 个外部输入点分别接上 10 个按键，依据这 10 个按键被压下之先后顺序可输入 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)，并将输入之数值存放在 **D₁**，而 **D₂** 则存放键盘之按键情形。
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。

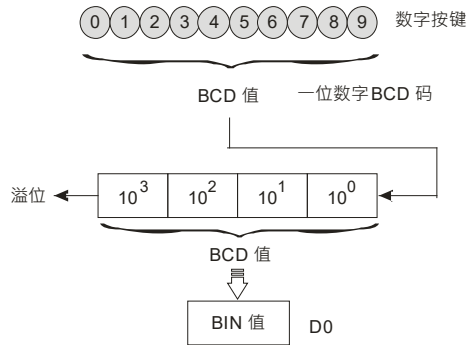
程序范例：

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，而按键之情况则放在 M10~M19。



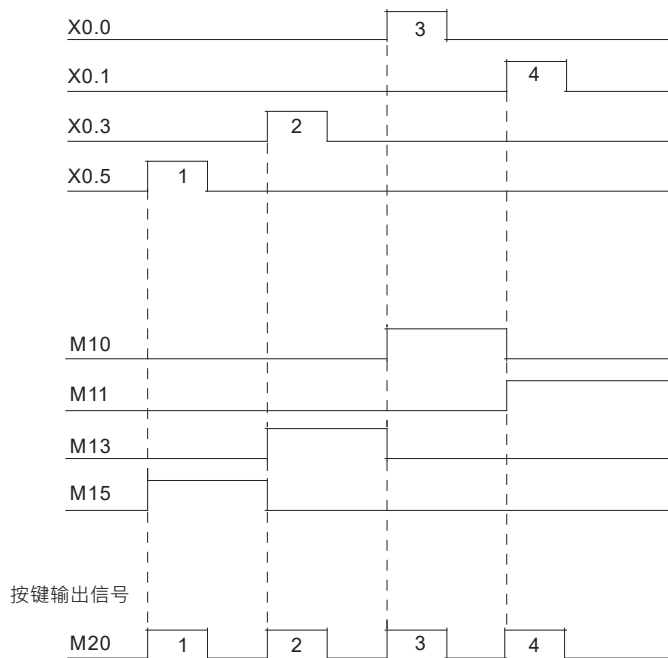


注: 此范例使用AH16AM10N 数字输入模块



2. 如下列时序图所示，连接于数字键盘 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 被暂存于 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，其它的数按键亦相同。
4. 当 X0.0~X0.9 当中任何一个按键被按下时，M10~M19 当中一一对应 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。

6



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。

API	指令码			操作数						功能					
1701	D	HKY		$S \cdot D_1 \cdot D_2 \cdot D_3$						16 键键盘输入					

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令 (9 steps)
-	AH500	AH500

符号：

HKY		DHKY	
En		En	
S1	D1	S1	D1
S2	D2	S2	D2
	D3		D3

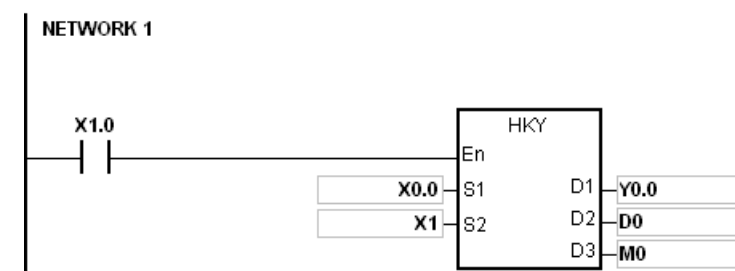
- S₁ : 按键扫描输入起始装置 Bit
S₂ : 系统内部使用 Word
D₁ : 按键扫描输出起始装置 Bit
D₂ : 按键输入值存放处 Word/Double Word
D₃ : 按键输出信号 Bit

指令说明：

1. 本指令指定由 **S** 开始之连续 4 个外部输入点及由 **D₁** 开始之连续 4 个外部输出点以矩阵扫描之方式构成 16 键之键盘。键盘输入之数值存放在 **D₂**，而 **D₃** 则存放键盘之按键情形，如果有数个按键同时被按下时，以 **S** 的编号小的为优先。
2. 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。
3. 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：

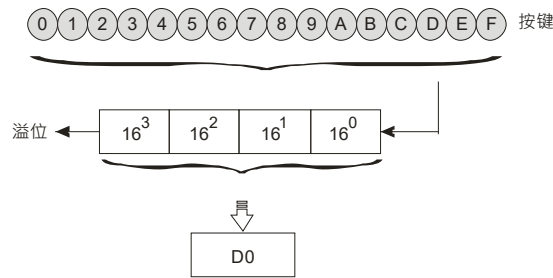
1. 指令指定 X0.0~X0.3 等 4 个输入端与 Y0.0~Y0.3 等 4 个输出端构成扫描 16 键之键盘。当 X1.0=ON 时，指令执行，由键盘输入的数值以 BIN 值的形态存入 D0 中，而按键之情况则放在 M0~M7。



标志 SM691 之功能：

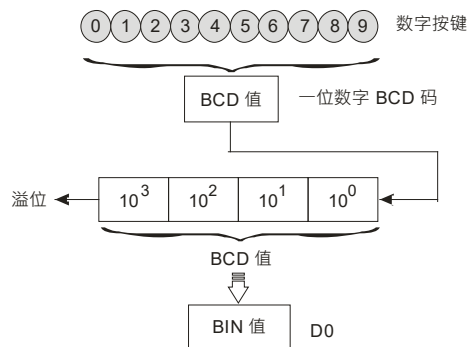
- SM691=ON 时，则 HKY 指令可以输入 0~F 的 16 进位数值。

■ 数字输入：



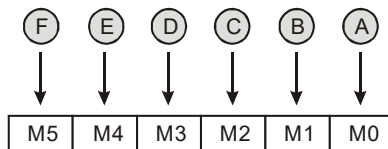
- SM691=OFF 时，则 HKY 指令 A~F 当功能键使用。

■ 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。



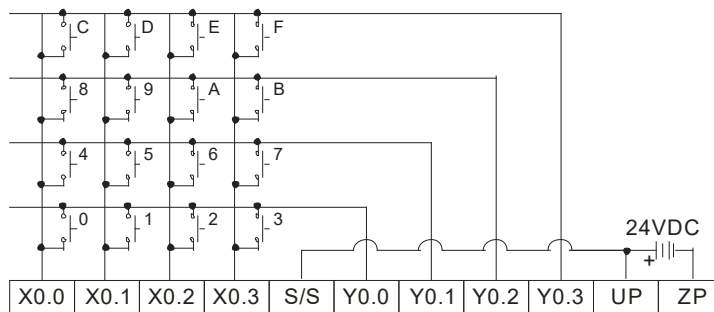
2. 按键输出信号：

- 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。

3. 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。

6

4. 外部配线：



注: 本范例使用AH16AP11晶体管输出模块

补充说明：

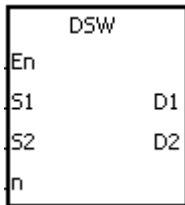
1. 本指令被执行时，当扫描周期太长或太短都可能造成按键输入不良，因此可运用下列技巧来克服。
2. 当扫描周期太短时，可能造成 I/O 来不及反应而无法读取正确之按键输入，此时，可将扫描时间加以固定。
3. 当扫描周期太长时，可能会使按键反应变得迟钝，可将此指令写在时间中断工作（TASK）内，固定时间执行此指令。
4. **S** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。
5. **D₁** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。
6. **D₃** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of BOOL。

API	指令码		操作数				功能					
1702		DSW	S · D ₁ · D ₂ · n				指拨开关输入					

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令
-	AH500	-

符号：



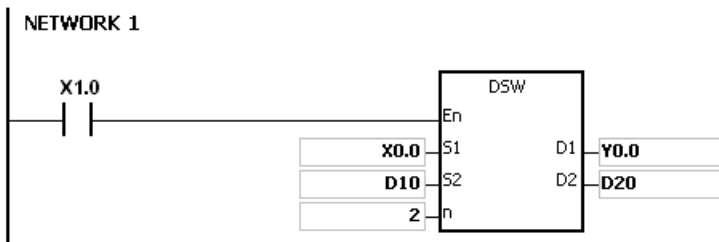
- S₁ : 指拨开关扫描输入起始装置 Bit
- S₂ : 系统内部使用 Word
- D₁ : 指拨开关扫描输出起始装置 Bit
- D₂ : 指拨开关设定值存放处 Word
- n : 指拨开关所连接之组数 Word

指令说明：

- 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。
- 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。
- S₂、S₂₊₁ 占用两个装置给系统内部使用，使用者请勿变更内容值。
- 每循环扫描一次完毕标志信号 SM694=ON 一个扫描周期。
- 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。
- 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。

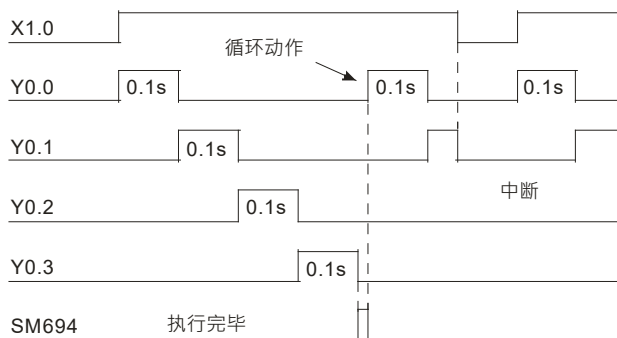
程序范例：

- 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 Y0.4~Y0.7 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。

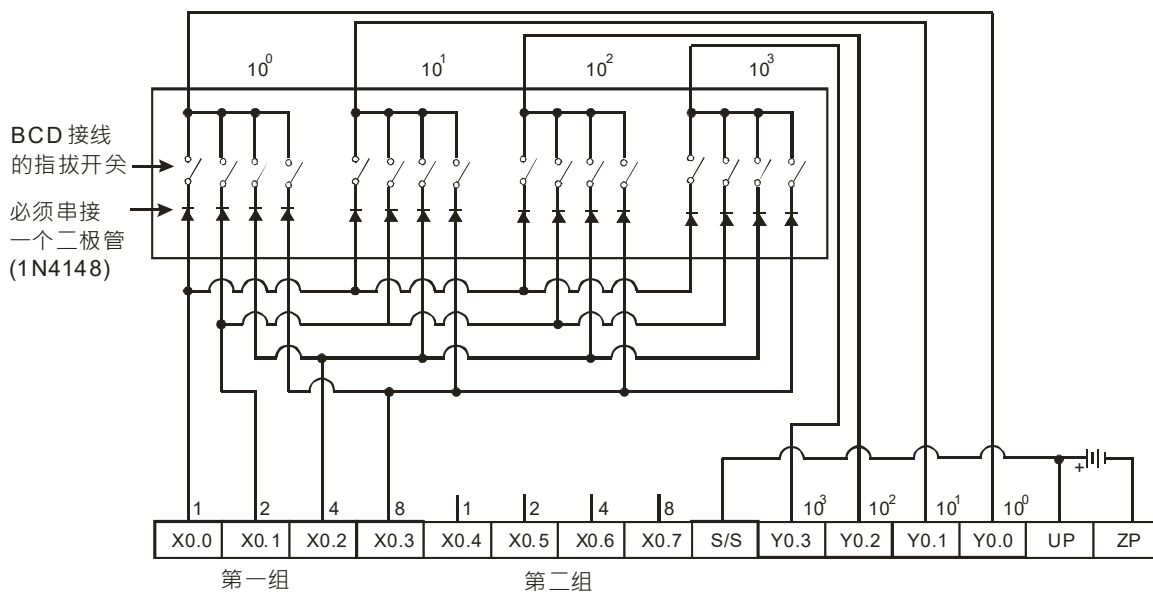


- 当 X1.0=ON 时，Y0.0~Y0.3 自动循环扫描 ON，每循环一次，执行完毕标志信号 SM694=ON 一个扫描周期。

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晶体管输出模块

补充说明：

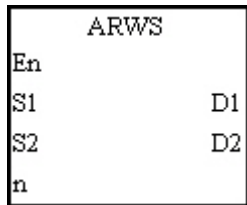
1. **n** 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B。
2. **D₁** 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of BOOL。

API	指令码		操作数							功能						
1703		ARWS	S · D₁ · D₂ · n							箭头键盘输入						

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令
-	AH500	-

符号：



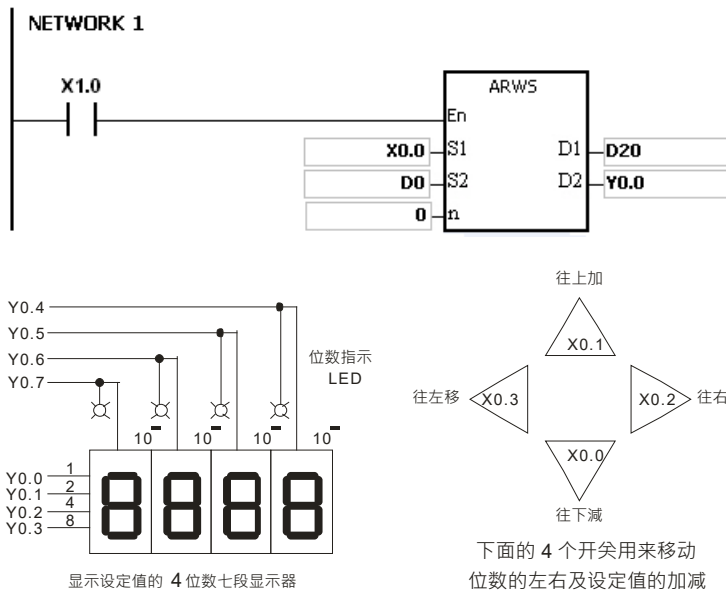
- S₁ : 按键输入起始装置 Bit
- S₂ : 系统内部使用 Word
- D₁ : 欲显示于七段显示器之装置 Word
- D₂ : 七段显示器扫描输出起始装置 Bit
- n : 输出信号及扫描信号之极性指示 Word

指令说明：

- 本指令执行，S₁ 定义为下键，S₁+1 定义为上键，S₁+2 定义为右键，S₁+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 D₁ 当中，设定值范围：0~9,999。
- S₁ 连续占用 4 个 bit 装置。
- S₂ 系统内部使用，使用者请勿变更内容值。
- D₂ 连续占用 8 个 bit 装置。
- 条件接点未启点时，D₂ 连续的 8 个 bit 装置保持 OFF。
- n 的范围：0~3，功能请参考 API1704 SEGL 的补充说明。
- 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：

- 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。
- 当 X1.0=ON 时，位数 10³ 为有效设定位数，如果按左按键时，则有效设定位数呈现 10³→10⁰→10¹→10²→10³→10⁰ 的方向循环跳动。
- 如果按右移按键，则有效设定位呈现 10³→10²→10¹→10⁰→10³→10² 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亦循环 ON 作有效设定位数的指示。
- 如果按往上加按键时，则有效设定位数的内容由 0→1→2→...8→9→0→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→9→8→...1→0→9 作变化，同时，变化值亦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。

API	指令码		操作数						功能					
1704		SEGL	S · D · n						7 段显示器扫描输出					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：



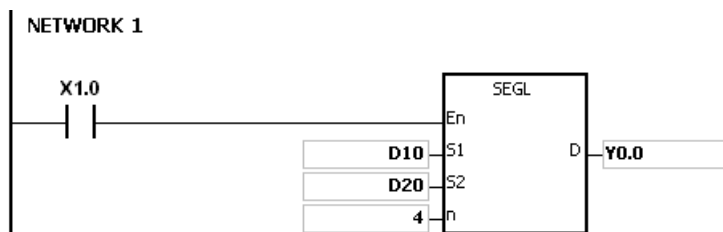
- S₁** : 欲显示于七段显示器之来源装置 Word
- S₂** : 系统内部使用 Word
- D** : 七段显示器扫描输出起始装置 Bit
- n** : 输出信号及扫描信号之正负逻辑设定 Word

指令说明：

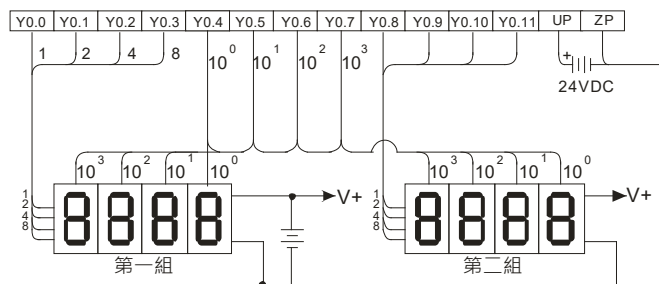
1. 本指令占用 **D** 开始的连续 8 个或 12 个外部输出点，作为 1 组或 2 组 4 位数七段显示器的显示数据及扫描信号输出。每个位数均带有 7 段显示器驱动器，该驱动器是将输入的 BCD 码转换 7 段显示器的驱动信号；驱动器并带有栓锁控制信号，可将 7 段显示器显示保持。
2. **S₁** 中之数值为欲显示到第一组七段显示器之数值，**S₁₊₁** 中之数值为欲显示到第二组七段显示器之数值。
3. **S₂** 系统内部使用，使用者请勿变更内容值。
4. **n=0~7**。请参考补充说明。
5. 由 **n** 决定扫描输出 4 位数七段显示器有 1 组或 2 组，且 **n** 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** 中之数值被转换成 BCD 码后送到第一组七段显示器显示出来，**D11** 中之数值被转换成 BCD 码后送到第二组七段显示器显示出来，若 **D10** 或 **D11** 中之数值超过 9,999 将发生运算错误。



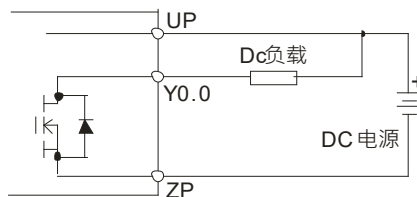
2. 当 X1.0=ON 时，Y0.4~Y0.7 会自动循环扫描，每循环扫描一次需 12 个扫描时间，每循环扫描一次完毕标志信号 SM693=ON 一个扫描周期。
3. 一组 4 位数的时候 n=0~3。
 - 将已经译码的 7 段显示模块 1、2、4、8 各端各自并接后连接至 PLC 的 Y0.0~Y0.3，而各位数的 Latch 端单独连接至 PLC 的 Y0.4~Y0.7。
 - 当 X1.0=ON 时，指令被执行，D10 的内容随着 Y0.4~Y0.7 的循环动作被顺序传送到七段显示器作显示。
4. 二组 4 位数的时候 n=4~7。
 - 将已经解碼的七段显示器 1、2、4、8 各端各自并接后连接至 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晶体管输出模组

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，采开集极式输出，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

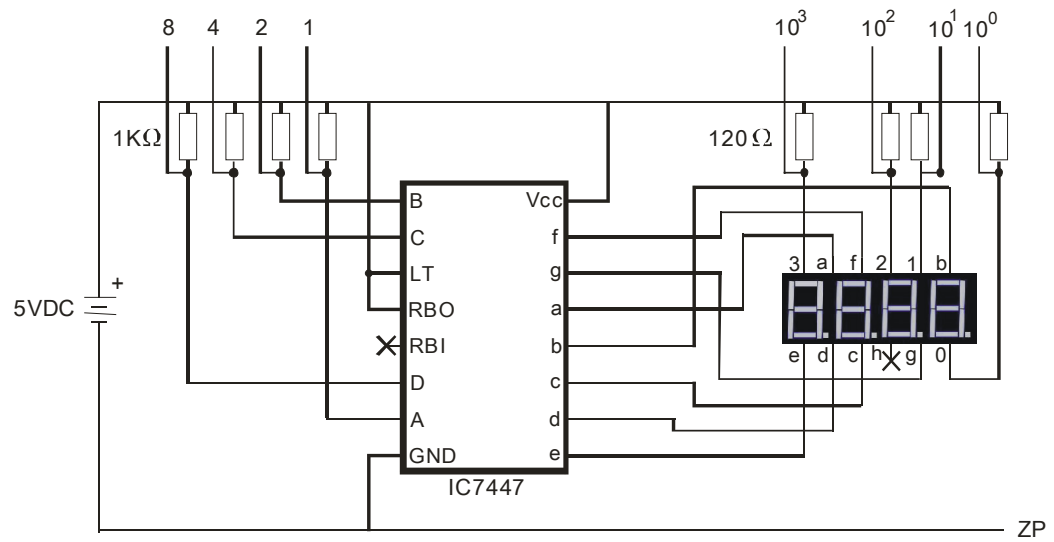
‘+’：正逻辑 (负极性) 输出

‘-’：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，可透过可参数 n 的设定值来相互匹配。

6

- 采用共阳极四位 7 段显示器搭配 IC7447 配置如下图



6.19 通讯指令

6.19.1 通讯指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1800</u>	RS	—	—	串行数据传输	9
<u>1801</u>	FWD	—	—	交流电机正转指令	7
<u>1802</u>	REV	—	—	交流电机反转指令	7
<u>1803</u>	STOP	—	—	交流电机停止指令	3
<u>1804</u>	RDST	—	—	交流电机状态读取	5
<u>1805</u>	RSTEF	—	—	交流电机异常复位	3
<u>1806</u>	LRC	—	✓	和检查 LRC 模式	7
<u>1807</u>	CRC	—	✓	和检查 CRC 模式	7
<u>1808</u>	MODRW	—	—	MODBUS 数据读写	11
<u>1812</u>	COMRS	—	—	通讯传送与接收指令	11

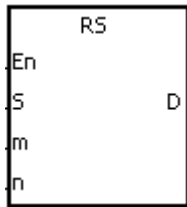
6.19.2 通讯指令说明

API	指令码		操作数							功能						
1800		RS	S · m · D · n							传送用户自行定义的通讯命令						

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

脉冲执行型	16 位指令 (9 steps)	32 位指令
-	AH500	-

符号：



- | | | |
|------------|-----------|------|
| S ： | 传送数据的起始装置 | Word |
| m ： | 传送数据的笔数 | Word |
| D ： | 接收数据的起始装置 | Word |
| n ： | 接收数据的笔数 | Word |

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. 本指令于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)。
7. 若接口设备之通讯格式符合 MODBUS 之通讯格式，PLC 提供通讯便利指令 MODRW 供用户使用。详细使用说明请参考个别指令之说明。

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，或于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. 于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

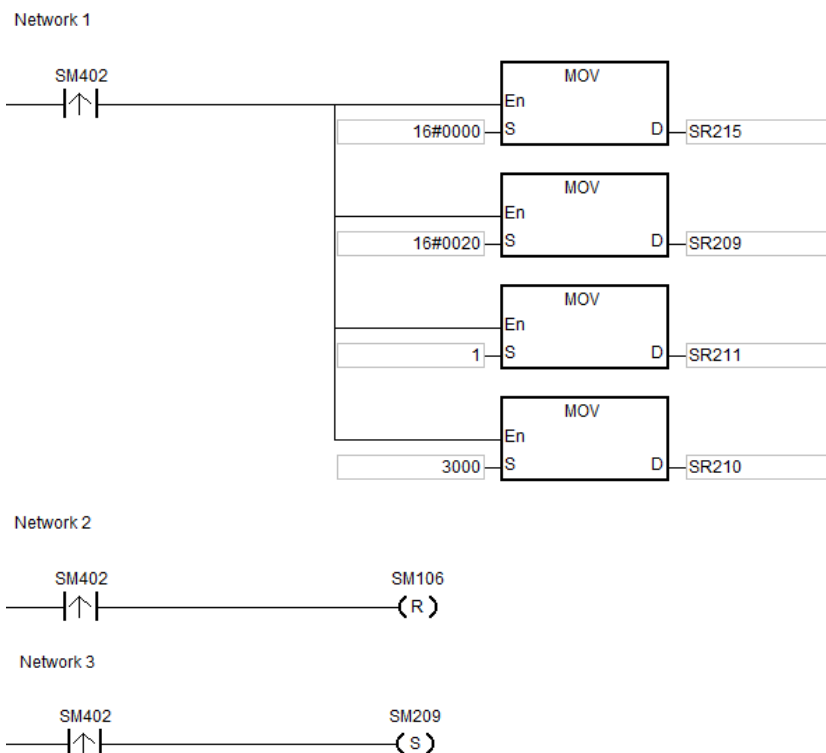
数据的传送格式分成几个部分 8 位与 16 位模式。8 位模式是将 16 位数据分成上 8 位、下 8 位，上 8 位被省略，仅下 8 位为有效数据可做数据的传送和接收。如模式为 16 位时要特别注意高低位的部分，请参照下表。

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)。

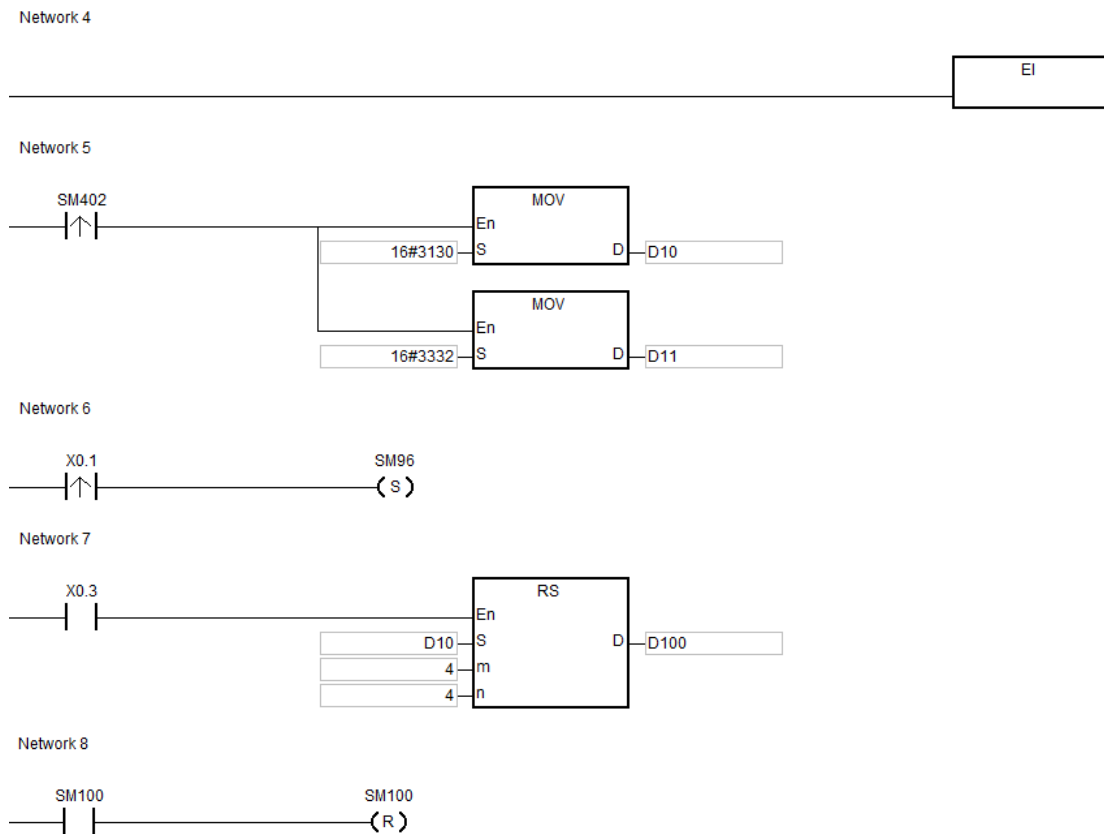
用户若在 ISPSOft - > HWCONFIG - > COM Port 内直接对通讯端口进行设定，则可以省略此步骤。



程序范例：

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 开始之连续寄存器内。
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



范例中断程序（I32）



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：

6

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，必须用脉波指令将 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
SM98	SM99	接收等待，当 SM98、SM99 为 ON 时，表示 PLC 目前正等待接收资料中。	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，若有检查到中断字符时，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，若有检查到中断字符时，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 指令相关设定之特殊数据寄存器：

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，用户可利用此标志作通讯超时的处理。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，此时超时侦测功能被取消，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 通信协议，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 不支援
	1000	(16#8)	:	460800		RS-232 不支援
	1001	(16#9)	:	921600		RS-232 不支援
b8~b15	无定义 (保留)					

6. PLC 相关通讯传输速度如下：

6

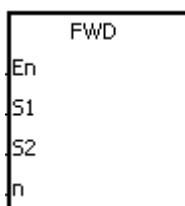
速率 (bps)	RTU 超时时间间隔 (ms)	速率 (bps)	RTU 超时时间间隔 (ms)
4800	9	115200	1
9600	5	230400	1
19200	3	460800	1
38400	2	921600	1
57600	1		

API	指令码			操作数							功能						
1801		FWD		$S_1 \cdot S_2 \cdot n$							交流电机正转						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：



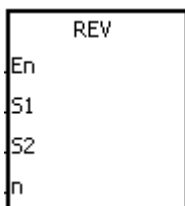
S_1 : 连线装置地址 Word
 S_2 : 交流电机运转频率 Word
 n : 模式选择 Word

API	指令码			操作数							功能						
1802		REV		$S_1 \cdot S_2 \cdot n$							交流电机反转						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：



S_1 : 连线装置地址 Word
 S_2 : 交流电机运转频率 Word
 n : 模式选择 Word

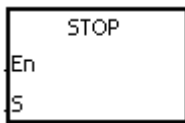
6

API	指令码	操作数	功能
1803	STOP	S ₁	交流电机停止

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

脉冲执行型	16 位指令 (3 steps)	32 位指令
-	AH500	-

符号：



S₁ : 连线装置地址

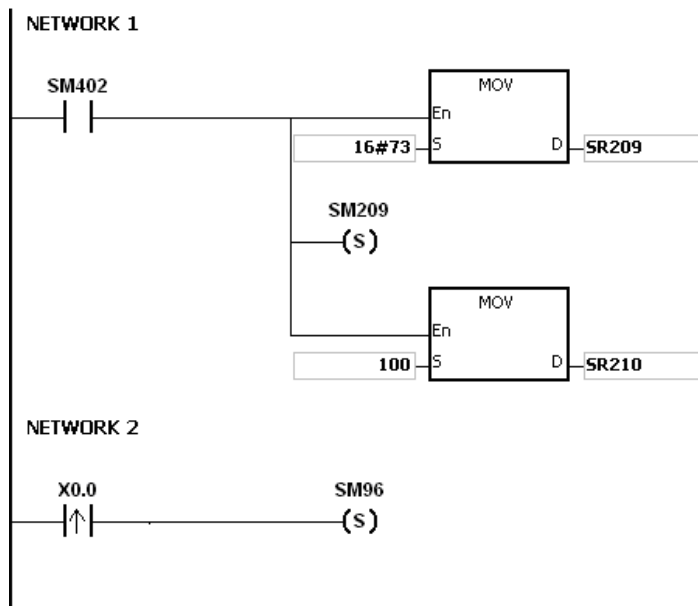
Word

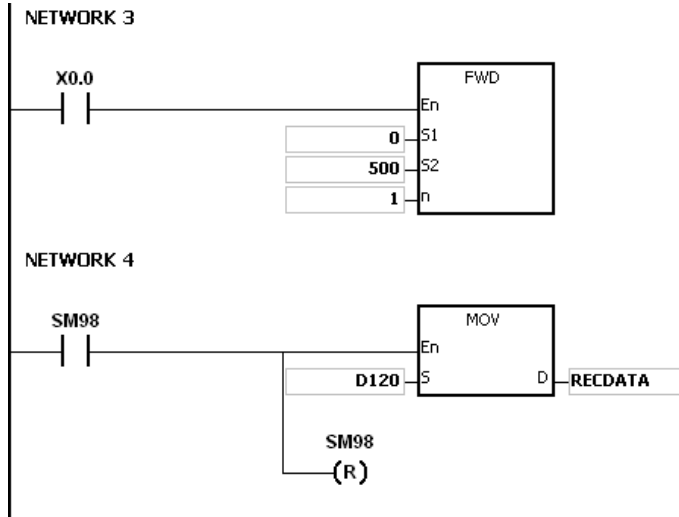
指令说明：

1. FWD/REV/STOP 为台达交流电机 VFD 系列专用的通讯便利指令，对交流电机下达正转/反转/停止的指令。此指令在应用时，必须配合通讯逾时设定 (SR210、SR212)。
2. S₁ : 范围 0~255，当站号为 0 时表示对所有交流电机进行广播通讯。
3. S₂ 频率值之设定值与数值单位，请参考交流电机使用手册，但在 STOP 指令为无此引脚。
4. n 模式选择依指令说明如下：
 FWD 指令：正转模式 n=0：一般正转模式，n=1：JOG 正转模式，其余数值皆无法支持。
 REV 指令：反转模式 n=0：一般反转模式，n=1：JOG 反转模式，其余数值皆无法支持。
 STOP 指令：无。
5. 当正转模式为 JOG 正转模式时，其 S₂ 的频率值将会无效，如须修改 JOG 频率请参考交流电机使用手册。

程序范例：

1. PLC 与 VFD 系列交流电机连线，通讯逾时及接收数据错误之重传数据。





PLC⇒VFD · PLC 传送：“ : 01 10 2000 0002 04 0012 01F4 C2 CR LF”

VFD⇒PLC · PLC 接收：“ : 01 10 2000 0002 CD CR LF”

PLC 传送数据 (传送信息)

Data		说明	
'0'	16#30	ADR 1	ADR (10) 为交流电机地址
'1'	16#31	ADR 0	
'1'	16#31	CMD 1	CMD (10) 为功能码
'0'	16#30	CMD 0	
'2'	16#32	数据地址 Data Address	
'0'	16#30		
'0'	16#30		
'0'	16#30	数据内容 Data contents	
'0'	16#30		
'0'	16#30		
'2'	16#32	Byte 数目	
'0'	16#30		
'4'	16#34	数据内容 1 16#12 为正转启动	
'0'	16#30		
'1'	16#31		
'2'	16#32	数据内容 2 运转频率=500Hz 16#01F4	
'0'	16#30		
'1'	16#31		
'F'	16#46		
'4'	16#34	LRC CHK (01) 为错误校验码	
'C'	16#43		
'2'	16#32	LRC CHK 0	

6

PLC 接收数据 (响应信息)

Data		说明	
'0'	16#30	ADR 1	ADR (10) 为交流电机地址
'1'	16#31	ADR 0	
'1'	16#31	CMD 1	CMD (10) 为功能码
'0'	16#30	CMD 0	
'2'	16#32	数据地址 Data Address	
'0'	16#30		
'0'	16#30		
'0'	16#30		
'0'	16#30	寄存器数目 Number of Register	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'C'	16#43	LRC CHK 1	
'D'	16#44	LRC CHK 0	

补充说明：

1. 相关标志信号与相关设定之特殊寄存器请参考 API1800 RS 指令补充说明。
2. FWD、REV、STOP、RDST、RSTEF 指令于程序中使用次数并无限制，但是同时间仅有一个指令被执行。
3. S_1 内容值超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003
4. n 超出范围时，指令不执行，SM0=ON，错误码 SR0=16#200B
5. FWD、REV、STOP、RDST、RSTEF 指令，符合 MODBUS 通讯格式。

API	指令码		操作数				功能						
1804		RDST	S · D				交流电机状态读取						

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
-	AH500	-

符号：



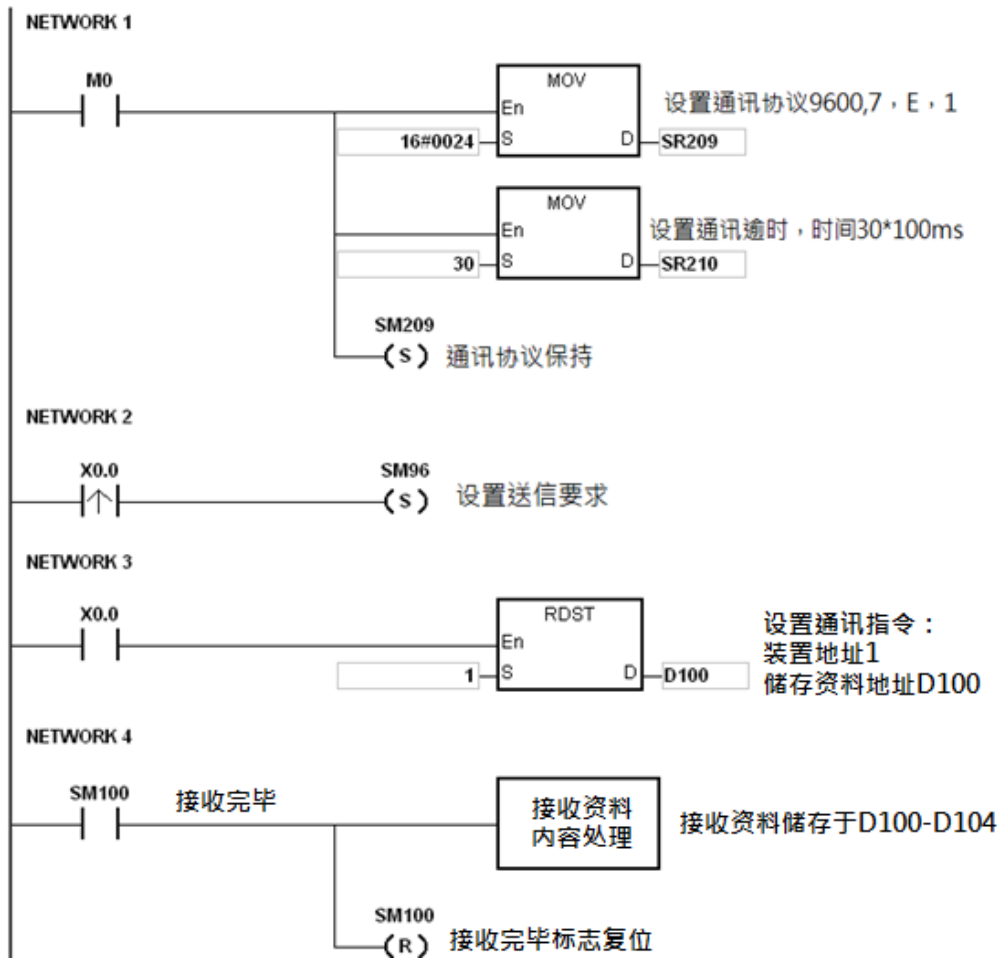
S : 连线装置地址 Word
D : 储存装置起始地址 Word

指令说明：

- S** : 范围 1~255。不能使用 0。
- D** : 将交流电机回复的数据，储存在用户指定的装置地址里。
- 此指令将会读取交流电机参数地址 16#2100~16#2104 之状态(详细状态说明请参考交流电机使用手册)，并于接收完成后储存于 **D** 参数所指定的装置地址里；不过当读取通讯接收信息错误或接收逾时发生时，此 **D** 内容将不会被更改，因此建议要判断交流电机状态信息前，请先确认接收完成标志是否已有被设定。
- 装置起始地址会占用到 5 个装置，例如：**Dn**、**Dn+1**、**Dn+2**、**Dn+3**、**Dn+4**。所以装置起始地址须要保留 5 个装置的地址。

程序范例：

- PLC 与 VFD 系列交流电机连线 (ASCII Mode · SM210 · SM212=OFF)，当通讯逾时之重传数据。
- 读取交流电机参数地址 16#2100~16#2104 之状态，储存于 D100~D104 之中。



PLC⇒VFD PLC 传送：“ : 01 03 2100 0005 D6 CR LF” (ASCII)

VFD⇒PLC · PLC 接收：“ : 01 03 0A 0000 0500 01F4 0000 0000 F8 CR LF” (ASCII)

PLC 传送数据 (传送信息)

Data		说明	
'0'	16#30	ADR 1	ADR (10) 为交流电机地址
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	CMD (10) 为命令码
'3'	16#33	CMD 0	
2'	16#32	起始数据地址 Starting Data Address	
'1'	16#31		
'0'	16#30		
'0'	16#30		
'0'	16#30	数据 (word) 个数 Number of Data (count by word)	
'0'	16#30		
'0'	16#30		
'5'	16#35		
'D'	16#44	LRC CHK 1	LRC CHK (01)
'6'	16#36	LRC CHK 0	为错误校验码 code

PLC 接收数据寄存器 (响应信息)

Data		说明	
'0'	16#30	ADR 1	
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	
'3'	16#33	CMD 0	
'0'	16#30	数据 (BYTE) 个数	
'A'	16#41	Number of Data (count by Byte)	
'0'	16#30	地址 16#2100 的内容	PLC 自动将 ASCII 字符转换为数值储存于 D100=16#0000
'0'	16#30		
'0'	16#30		
'0'	16#30		
'0'	16#30	地址 16#2101 的内容	PLC 自动将 ASCII 字符转换为数值储存于 D101=16#0500
'5'	16#35		
'0'	16#30		
'0'	16#30		
'0'	16#33	地址 16#2102 的内容	PLC 自动将 ASCII 字符转换为数值储存于 D1072=16#01F4
'1'	16#45		
'F'	16#30		
'4'	16#30		
'0'	16#30	地址 16#2103 的内容	PLC 自动将 ASCII 字符转换为数值储存于 D1073=16#0000
'0'	16#30		
'0'	16#30		
'0'	16#30		
'0'	16#30	地址 16#2104 的内容	PLC 自动将 ASCII 字符转换为数值储存于 D1074=16#0000
'0'	16#30		
'0'	16#30		
'0'	16#30		
'2'	16#32	LRC CHK 1	
'A'	16#41	LRC CHK 0	

补充说明：

1. 相关标志信号与相关设定之特殊寄存器请参考 API1800 RS 指令补充说明。
2. FWD、REV、STOP、RDST、RSTEF 指令于程序中使用次数并无限制，但是同时间仅有一个指令被执行。
3. S 内容值超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
4. D+4 超过装置范围时，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 [5] of WORD/INT。

API	指令码	操作数	功能
1805	RSTEF	S	交流电机异常复位

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

脉冲执行型	16 位指令 (3 steps)	32 位指令
-	AH500	-

符号：



S：连线装置地址

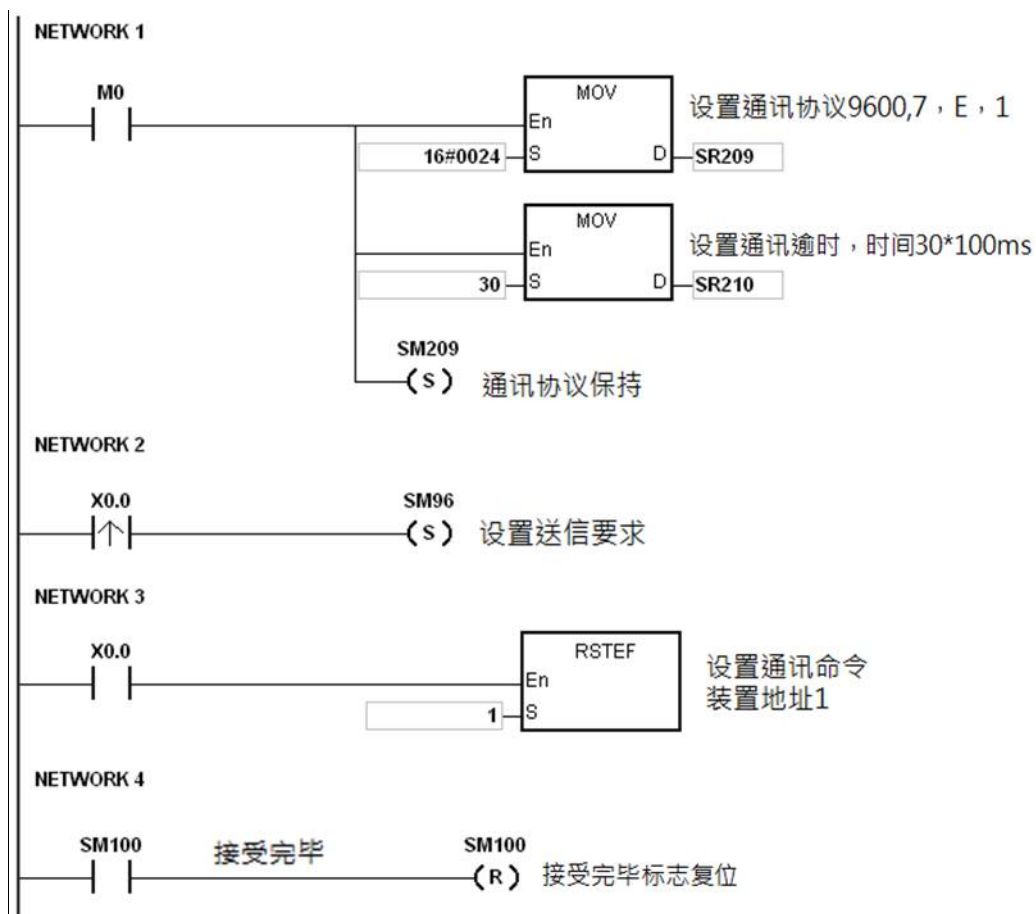
Word

指令说明：

- S：范围 0~255，当站号为 0 时表示对所有交流电机进行广播通讯。

程序范例：COM2 (RS-485)

PLC 与 VFD 系列交流电机连线 (ASCII 模式，SM210、SM212=OFF)，当通讯逾时之重传数据。



PLC⇒VFD · PLC 传送“ : 01 06 2002 0002 D5 CR LF” (ASCII)

VFD⇒PLC · PLC 接收：“ : 01 06 2002 0002 D5 CR LF” (ASCII)

PLC 传送数据 (传送信息)

Data		说明	
'0'	16#30	ADR 1	ADR (10) 为交流电机地址
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	CMD (10) 为命令码
'6'	16#36	CMD 0	
'2'	16#32	数据地址 Data Address	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'0'	16#30	数据内容 Data contents	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'D'	16#44	LRC CHK 1	LRC CHK (01) 为错误校验码
'5'	16#35	LRC CHK 0	

PLC 接收数据 (响应信息)

Data		说明	
'0'	16#30	ADR 1	
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	
'6'	16#36	CMD 0	
'2'	16#32	数据地址 Data Address	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'0'	16#30	寄存器数目 Number of Register	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'D'	16#44	LRC CHK 1	
'5'	16#35	LRC CHK 0	

补充说明：

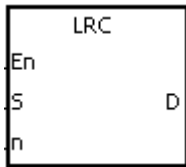
1. 相关标志信号与相关设定之特殊寄存器请参考 API1800 RS 指令补充说明。
2. FWD、REV、STOP、RDST、RSTEF 指令于程序中使用次数并无限制，但是同时间仅有一个指令被执行。
3. **S** 内容值超出范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。

API	指令码		操作数							功能						
1806		LRC	S · n · D							和检查 LRC 模式						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：



- S** : ASCII 模式和检查运算起始装置 Word
n : 运算组数 Word
D : 存放运算结果之起始装置 Word

指令说明：

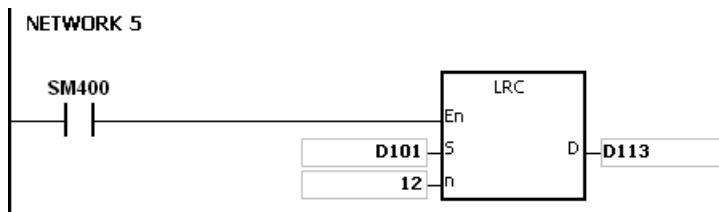
1. LRC 校验码：请参考补充说明。
2. **n** : 运算组数须为偶数，范围 1~1000 不在此范围则视为运算错误，指令不执行，SM0、SM1=ON，SR0 记录错误码 16#200B。
3. 16 位转换模式：当 SM606=OFF 时，S 起始装置将其 16 进位数据区分为上 8 位、下 8 位，将每一个位数做 LRC 校验码运算，传送到 D 的上 8 位及下 8 位中，运算的位数以 n 来设定。
4. 8 位转换模式：当 SM606=ON 时，S 起始装置将其 16 进位数据区分为上 8 位（无效数据）、下 8 位，将每一个位数做 LRC 校验码运算，传送到 D 的下 8 位中占用 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 系列交流电机连线（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 开始之 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) 为错误校验码
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) 为错误校验码可由指令 LRC 算出 (8 位 Mode · SM606=ON)



LRC 校验码：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. 有一通讯数据之 ASCII 模式，格式如下：

STX	‘:’	起始字符=‘:’ (16#3A)
Address Hi	‘0’	通讯地址： 8-bit 地址由 2 个 ASCII 码组合
Address Lo	‘1’	
Function Hi	‘0’	功能码： 8-bit 功能码由 2 个 ASCII 码组合
Function Lo	‘3’	
DATA (n-1)	‘2’	数据内容： n×8-bit 数据内容由 2n 个 ASCII 码组合
.....	‘1’	
DATA 0	‘0’	
	‘2’	

	'0'	
	'0'	
	'0'	
	'2'	
LRC CHK Hi	'D'	LRC 校验码：
LRC CHK Lo	'7'	8-bit 校验码由 2 个 ASCII 码组合
END Hi	CR	结束字符：
END Lo	LF	END Hi=CR (16#0D) · END Lo=LF (16#0A)

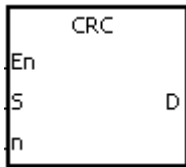
2. LRC 校验码：由通讯地址到数据内容结束加起来的值取 2 的补码即为校验码 (LRC Check)。例如：16#01+16#03+16#21+16#02+16#00+16#02=16#29，然后取 2 的补码 =16#D7。

API	指令码		操作数				功能								
1807		CRC			S · n · D				检查 CRC 模式						

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：



- S : RTU 模式和检查运算起始装置
- n : 运算组数
- D : 存放运算结果之起始装置。

指令说明：

1. CRC 校验码：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 校验码运算，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 校验码运算，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：

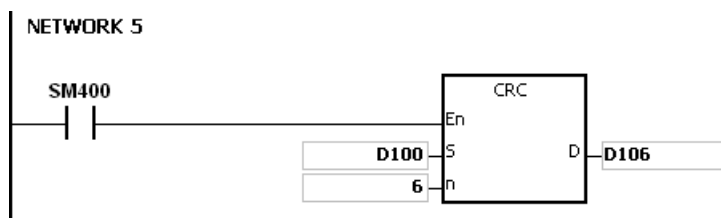
1. PLC 与 VFD-S 系列交流电机连线（RTU 模式，SM210=ON）、（16 位模式，SM606=ON），传送数据预先写入欲写入 VFD-S 参数地址 16#2000 写入内容为 16#12。

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) 为错误校验码可由指令 CRC 算出 (8 位 Mode · SM606=ON)



CRC 校验码：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. 有一通讯数据之 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 校验码：
CRC CHK High	16-bit CRC 校验码由 2 个 8-bit 二进制组合
END	时间间隔

2. CRC 校验码：校验码由 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。

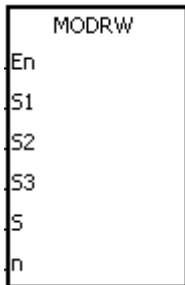
步骤六：重复步骤 2~步骤 5，取下一个 8-bit 的信息指令，直到所有信息指令运算完成。最后，得到的 CRC 寄存器的值，即是 CRC 的校验码。值得注意的是 CRC 的校验码必须交换放置于信息指令的校验码中。

API	指令码		操作数							功能						
1808		MODRW	$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot n$							MODBUS 数据读写						

装置	X	Y	M	S	T	C	HC	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 (11 steps)	32 位指令
-	AH500	-

符号：



- S_1 : 连线装置地址 Word
- S_2 : 通讯功能码 Word
- S_3 : 欲读写数据的地址 Word
- S : 欲读写之数据存放寄存器 Bit/ Word
- n : 读写数据长度 Word

指令说明：

1. S_1 : 连线装置地址 (Unit Address)。指定范围 0~255。
2. S_2 : 通讯功能码 (Function Code)。

例如：

AH500 的读取多笔位装置 (Bit) (非 Discreted Input 装置) 命令为 1 (16#01)。

AH500 的读取多笔位装置 (Bit) (仅 Discreted Input 装置) 命令为 2 (16#02)。

AH500 的读取多笔字符装置 (Word) (非 Input Register 装置) 命令为 3 (16#03)。

AH500 的读取多笔字符装置 (Word) (仅 Input Register 装置) 命令为 4 (16#04)。

AH500 的单笔位装置 (Bit) 状态写入命令为 5 (16#05)。

AH500 的单笔字符装置 (Word) 数据写入命令为 6 (16#06)。

AH500 的多笔位装置 (Bit) 状态写入命令为 15 (16#0F)。

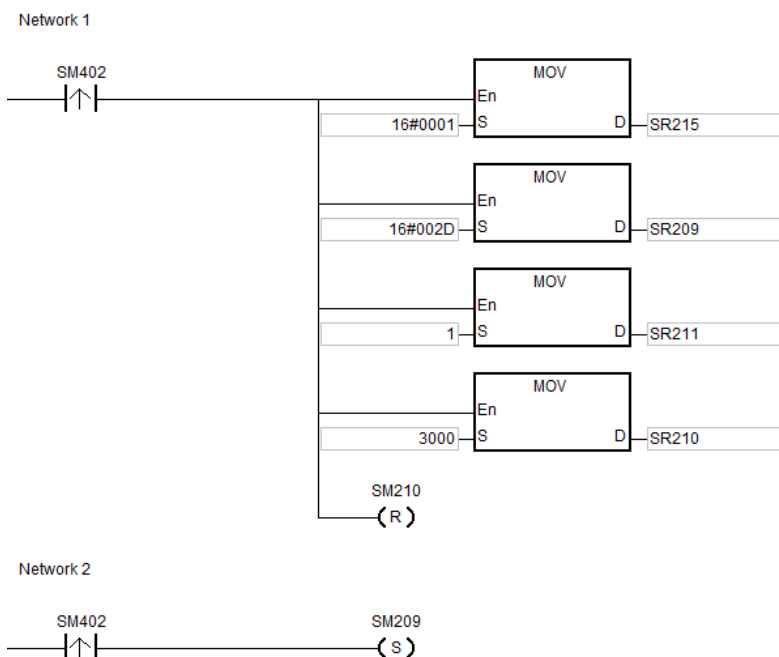
AH500 的多笔字符装置 (Word) 数据写入命令为 16 (16#10)。

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. S_3 : 欲读写数据的地址 (Device Address)。连线装置的内部装置地址，若地址对于被指定的装置不合法，则连线装置会响应错误信息，AH500 将错误码储存于 Error Log 里。(例如 16#8000 对 DVP-ES2 不合法，会将 error code 储存到 Error Log 里，错误码请参考 DVP-PLC 使用手册)。
4. S : 欲读写之数据 (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。或数据读取后存放之寄存器。

5. **n** : 读写数据长度 (Data Length) , 设定的数据量不可以大于 240 bytes 。 Coil 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. 本指令于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. 于 MODBUS ASCII 模式时 , 用户只需设定好 所欲传送之数据 , 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 , AH500 自动将其内容转换为 HEX 数值存放在 S 中。
9. 于 MODBUS RTU 模式时 , 用户只需设定好 所欲传送之数据 , 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 :

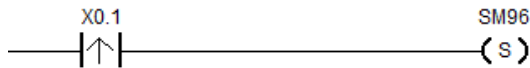
1. 通讯协议设定成 RS485 ASCII , 9600 , 8 , E , 1 , 用户可利用 HWCONFIG 或是特殊寄存器来设定通讯端口通讯能力。(HWCONFIG 相关设定请参照 ISPSOFT 使用手册 , 通讯相关 SR , SM 寄存器设定可参考 API1800 RS 通讯应用指令)。
2. 设定通讯端口传输方式为 RS485 (SR215=1)。
3. 设定通讯端口其通讯速率与格式为 9600 , 8 , E , 1 (SR209=16#002D)。
4. 设定其自动询问次数为 1 (SR211=1)。
5. 设定其逾时时间为 3000ms (SR210=3000)。
6. 设定为 ASCII 模式 (SM210=OFF)
7. 通讯端口通讯能力设定生效 (SM209=ON)。



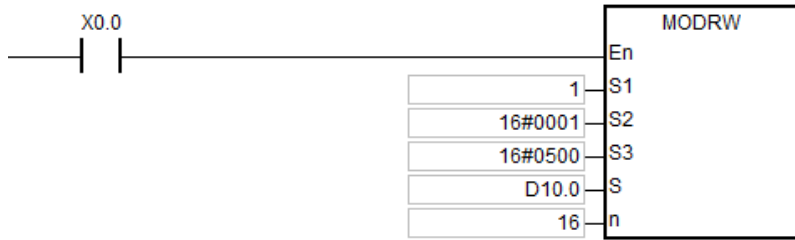
用户若在 ISPSOft - > HWCONFIG - > COM Port 内直接对通讯端口进行设定，则可以省略此步骤。
程序范例一：

1. 功能码 01 (16#01)：读取多笔位数据。(此范例读取 16 笔)
2. AH500 与 DVP-ES2 连线：
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。

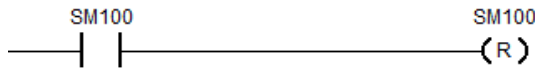
Network 3



Network 4



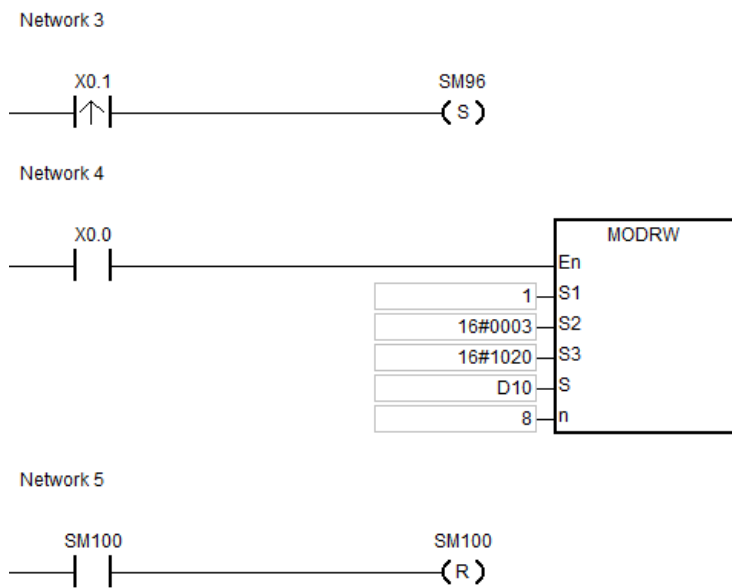
Network 5



程序范例二：

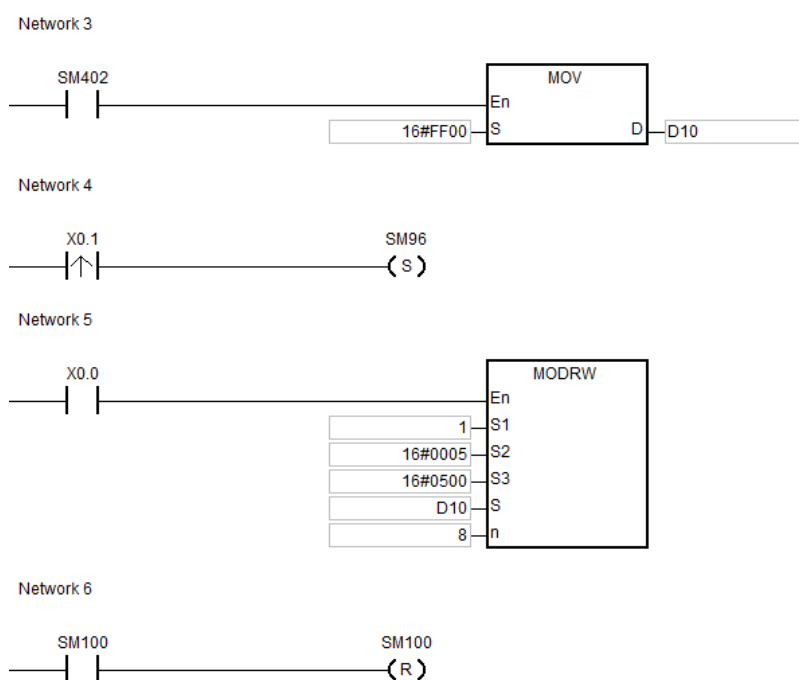
1. 功能码 03 (16#03)：读取多笔寄存器数据。(此范例读取 8 笔)
2. AH500 与 DVP-ES2 联机
当 SM96 为 ON 且 X0.0 也为 ON 的状态时，将会对 DVP-ES2 进行数据的读取，从 D20 开始至 D27 (D20 的地址为 16#1020)。
3. DVP-ES2 所回传的数据会储存在 D10 至 D17 装置内。
4. DVP-ES2 所回传的数据接收完毕后，会对 DVP-ES2 所回传的数据格式进行确认，是否符合数据格式，如果正确无误则 SM100=ON，如果回传数据格式错误则 SM102=ON。

6



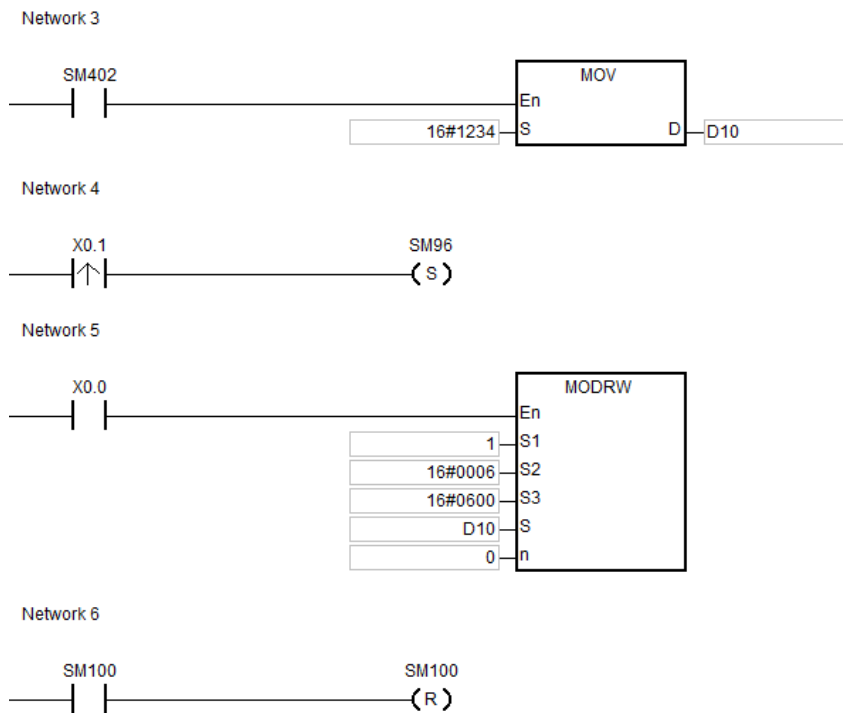
程序范例三：

1. 功能码 05 (16#05) : 写入单个位数据至 DVP-ES2 。
2. ASCII Mode : AH500 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，所以于本指令中的 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 : AH500 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，所以于本指令中的 n 值，并不会被使用到。



补充说明：

1. S₁ 或 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。
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 S_2 为读写 Word 装置 , 则读写对方设备装置也必须是 Word 装置 , 当 S_2 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

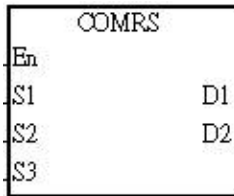
标志信号的功能详细说明请参考 API1800 RS 通讯应用指令 。

API	指令码		操作数						功能					
1812	COMRS		S ₁ , S ₂ , S ₃ , D ₁ , D ₂						通讯传送与接收指令					

装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	“\$”	DF
S ₁	●	●			●	●		●	●				○	○		
S ₂	●	●			●	●		●	●							
S ₃	●	●			●	●		●	●				○	○		
D ₁	●	●			●	●		●	●							
D ₂	●	●			●	●		●	●							

脉冲执行型	16 位指令 (11 steps)	32 位指令
-	AH500	-

符号：



- S₁** : 通讯端口编号 (1~2) Word
- S₂** : 传送数据来源 Word
- S₃** : 传送长度设定值 Word
- D₁** : 接收通讯数据的起始装置组件 Word
- D₂** : 接收数据结束条件参数设定 Word

6

指令说明：

1. 仅适用在版本 V1.03 以上。
2. 当接收结束条件有指定特定字符时，建议使用在 ASCII 编码之通讯数据，若非 ASCII 编码之通讯数据，则建议改用通讯逾时时间做为结束条件。
3. COMRS 指令仅支持 AH500 主机内建的 COM · AH5x0-EN 系列为 COM1 · AH5x0-RS2 系列为 COM1 与 COM 2。
4. **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。
5. **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，并设定为 8 位模式，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。
6. 当 **S₃** 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。
7. **D₁** 为已接收数据长度值，**D₁+1~ D₁+n** 为存放接收数据地址，说明如下：假设 **D₁** 设为 D200，**D₂=3** 及 **D₂+1** 设定结束字符 16#0D0A，则指令将会依照接收数据顺序存放在 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。

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。

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。		

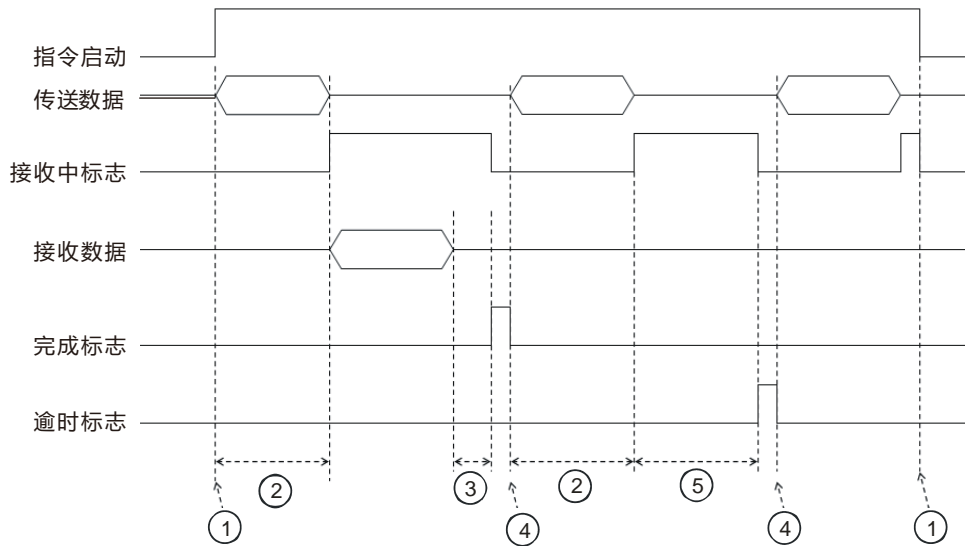
9. 当 D_2 内容值为 1~5 时, 最大接收长度为 1000 字符, 到达 1000 字符时, 指令自动结束。

10. 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。
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。

11. 通讯传送与接收时序图如下：

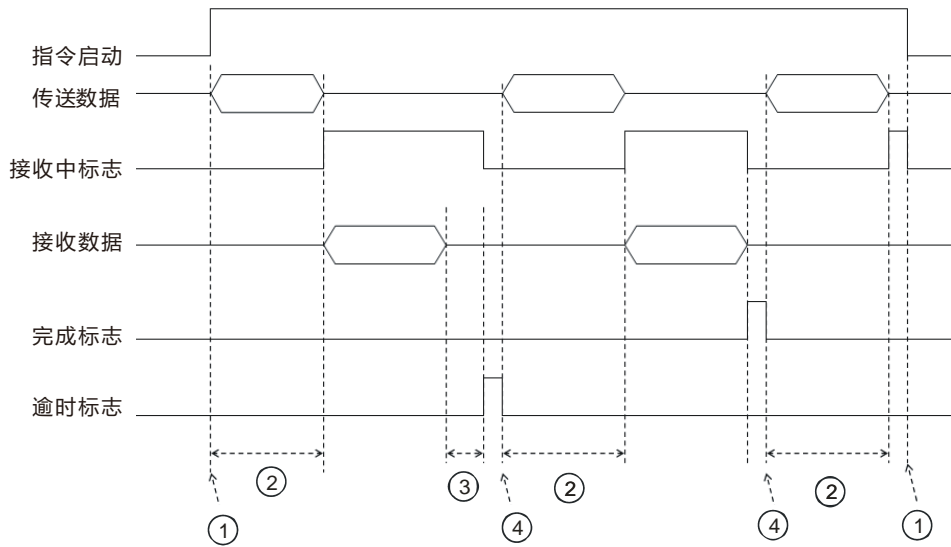
- 接收模式为 0 时
需注意数据传送时无法中途取消，此时即使指令启动条件不成立，仍会继续传送数据，惟传送完成后不会将完成标志设为 ON。
- 接收模式为 1 或 4 时



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 D_2+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 D_2+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，若指令还处在开启状态，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。

12. 数据传送/接收模式：

《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
起始码		来源数据寄存器由D10下			结束码1 结束码2	
8位开始 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
起始码		来源数据寄存器由D100下			结束码1 结束码2	
8位开始 长度=7						

《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，传送出去的命令包起始码及结束码。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
起始码		来源数据寄存器由D10下 8位开始 长度= 7			结束码1 结束码2	

接收数据：(外部机器→PLC)

D100下	D101上	D102下	D103上	D104下	D105上	D106下
起始码		来源数据寄存器由D100下 8位开始 长度= 7			结束码1 结束码2	

PLC 接收数据会将外部机器传入数据包含头码、后缀一起接收，所以长度之设定要注意。

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，因此后面启动之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之通讯指令。
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之便利指令自行组合运用。
4. D2 内容值为 2、3、5、6 时，建议设定通讯逾时。若有设定通讯逾时，当指令没有收到结束字符时，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 $D_1+0 \sim D_1+n$ 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 S_1 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. S_2 装置不足以读取 S_3 所指定的长度时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
8. 当 D_2 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 D_2 的内容值为 6， D_1 装置不足以写入 D_2+1 所指定的长度时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
10. 当 D_2 的内容为 1~5 时，接收数据超出 D_1 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 S_3 传送长度设定值为小于 0 或大于 1000 时 指令不执行 SM0=ON SR0 错误码为 16#2003。

6.20 其它指令

6.20.1 其它指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>1900</u>	WDT	—	✓	逾时监视定时器	1
<u>1901</u>	DELAY	—	✓	延时指令	3
<u>1902</u>	GPWM	—	—	一般用脉冲宽度调变	7
<u>1903</u>	TIMCHK	—	—	时间检查	7
<u>1904</u>	EPUSH	—	✓	变址寄存器存入	3
<u>1905</u>	EPOP	—	✓	变址寄存器读出	3

6.20.2 其它指令说明

API	指令码		操作数	功能
1900	WDT	P	—	逾时监视定时器

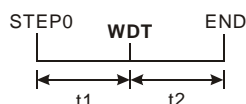
脉冲执行型	16 位指令 (1 step)	32 位指令
—	AH500	—

符号：



指令说明：

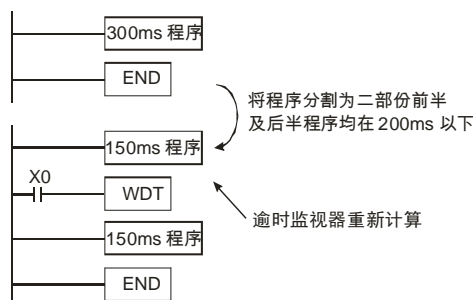
1. AH500 系列 PLC 系统中有一逾时监视定时器 (Watchdog Timer)，用来监视 PLC 系统的正常运转。
2. WDT 指令可用来清除 PLC 中之监控定时器之计时时间。当 PLC 的扫描程序时间过长，超过使用者设定的 WDT 时间(默认值 200ms)时，PLC ERROR 的指示灯会亮，PLC 会自动变 STOP，使用者只要重新做 STOP=>RUN 就可以恢复正常。
3. 令逾时监视定时器动作的时机：
 - PLC 系统发生异常。
 - 程序执行时间太长，造成扫描时间大于 (Watchdog Timeout) 的内容值。可以下列 2 种方法来改善。
 - (a) 使用 WDT 指令。



- (b) 要改变逾时监视时间请参考 ISPSOft 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：

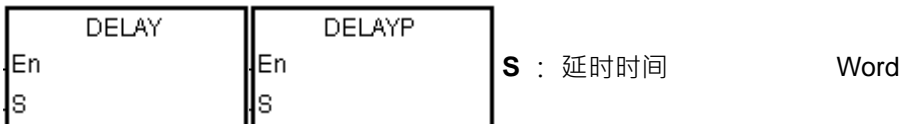
设定 WDT 时间的方式，请参考 ISPSOft 的使用手册说明。

API	指令码			操作数						功能					
1901		DELAY	P	S						延时指令					

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

脉冲执行型	16 位指令 (3 steps)	32 位指令
AH500	AH500	—

符号：



指令说明：

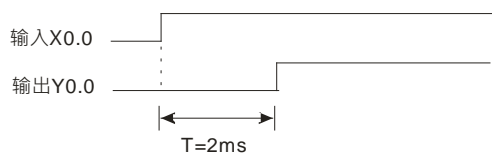
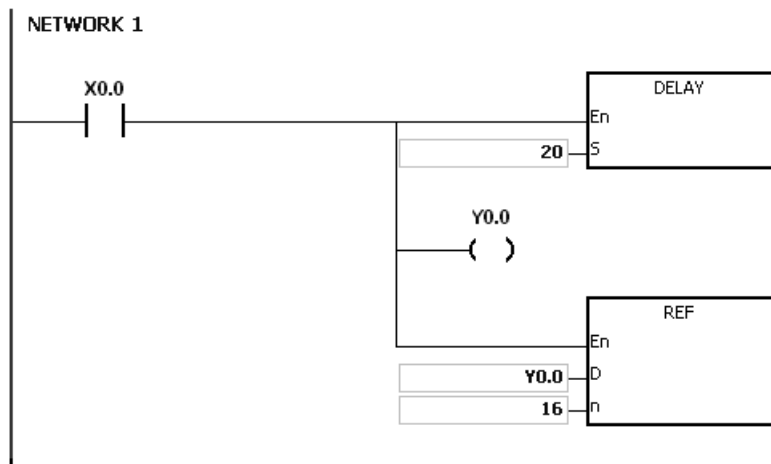
执行 DELAY 指令后，在每次扫描周期 DELAY 指令后面之程序执行会依使用者指定之时间作延时。

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。

6



补充说明：

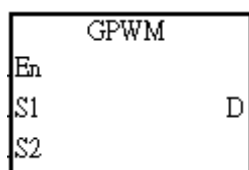
1. **S**≤0 时，不会有延时时间。
2. **S**>1000，指令不执行，SM0=ON，错误码 SR0=16#2003。
3. 使用者可依实际状况来调整延时时间。
4. DELAY 指令会受到通讯或其它影响而增加延时时间。

API	指令码		操作数						功能					
1902	GPWM		$S_1 \cdot S_2 \cdot D$						基本脉冲宽度调变					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
—	AH500	—

符号：



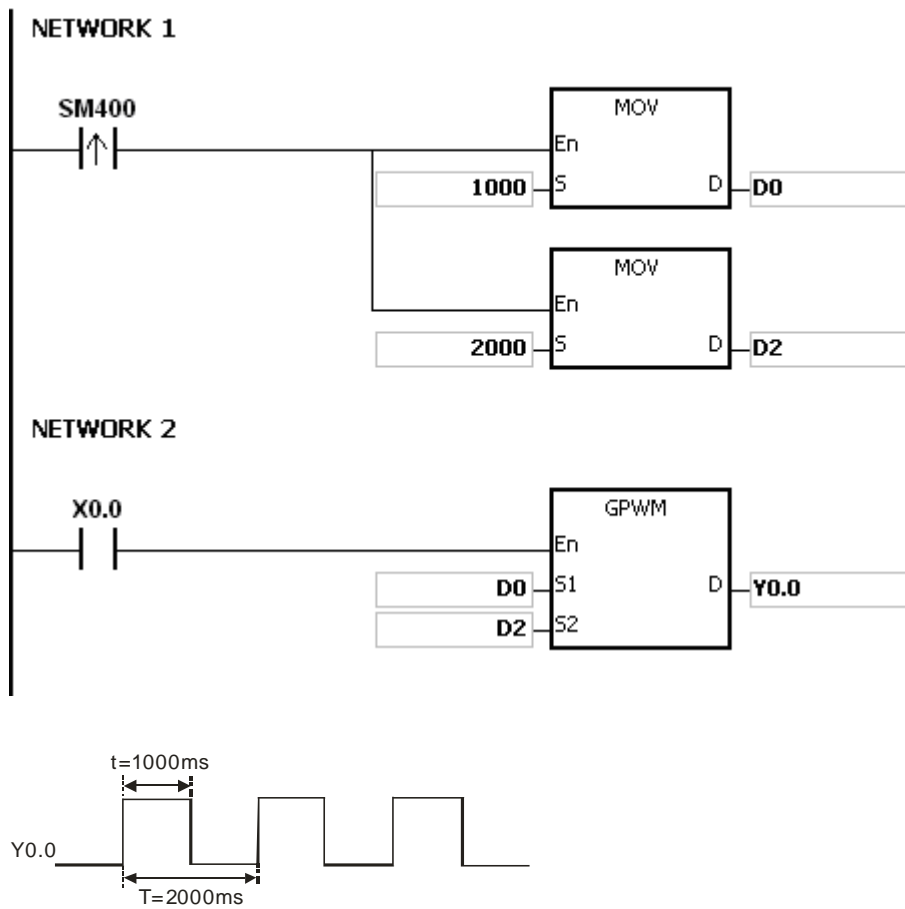
S_1 : 脉冲输出宽度 Word
 S_2 : 脉冲输出周期 Word
D : 脉冲输出装置 Bit

指令说明：

1. GPWM指令执行时，指定 S_1 脉冲输出宽度与 S_2 脉冲输出周期，并由D脉冲输出装置输出。
2. S_1 脉冲输出宽度指定t：0~32·767ms。
3. S_2 脉冲输出周期指定为T：1~32·767ms，但 $S_1 \leq S_2$ 。
4. S_2+1 、 S_2+2 为系统用参数，请勿占用。
5. 当 $S_1 \leq 0$ 时，脉冲输出装置无输出，当 $S_1 \geq S_2$ 时，脉冲输出装置一直为ON。
6. S_1 、 S_2 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。



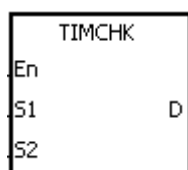
6

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。S₁、S₂ 与 (S₂-S₁) 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. S₂ 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

API	指令码			操作数							功能								
1903	TIMCHK			$S_1 \cdot S_2 \cdot D$							时间检查								
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF		
S_1	●	●			●	●		●	●				●						
S_2	●	●			●	●		●	●		●	○	●	○	○				
D	●	●	●	●				●	●				●						
											脉冲执行型			16 位指令 (7 steps)			32 位指令		
											—			AH500			—		

符号：



S_1 : 存放测量值所经过的时间 Word

S_2 : 测量设定值 Word

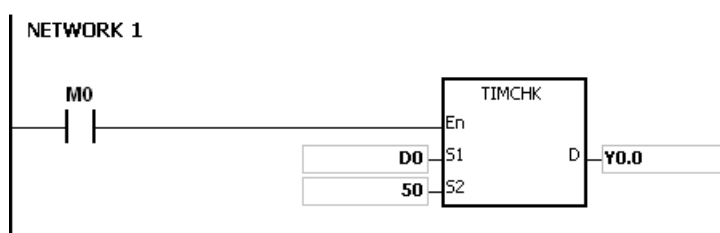
D : 测量时间到达时的输出装置 Bit

指令说明：

1. 条件接点ON时， S_1 开始计数，直到 $S_1 \geq S_2$ ，D输出装置=ON，之后就算条件接点变成OFF， S_1 内容值不变且D装置一样为ON。
2. 条件接点由OFF→ON时， S_1 被清除为0，D输出装置=OFF。
3. 计数的单位为100ms。
4. S_{1+1} 、 S_{1+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 $D0 \geq 50$ (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。

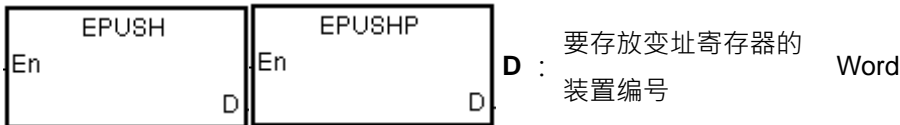


补充说明：

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。

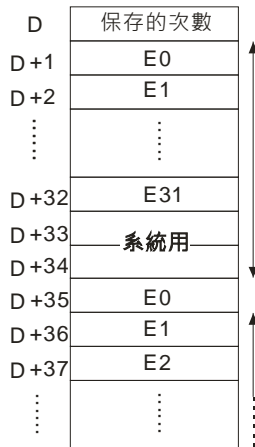
API	指令码			操作数								功能					
1904		EPUSH	P	D								变址寄存器存入					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●				●				
脉冲执行型										16 位指令 (3 steps)			32 位指令				
AH500										AH500			—				

符号：



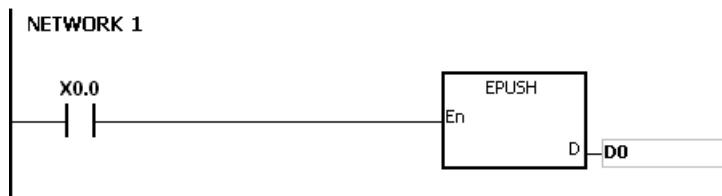
指令说明：

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

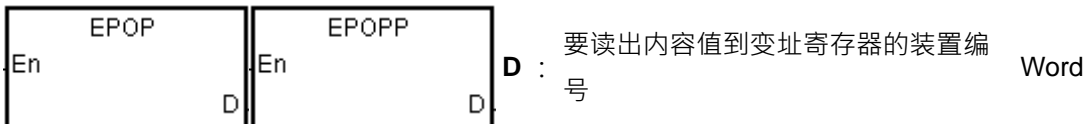


补充说明：

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 $SR0=16\#2003$ 。

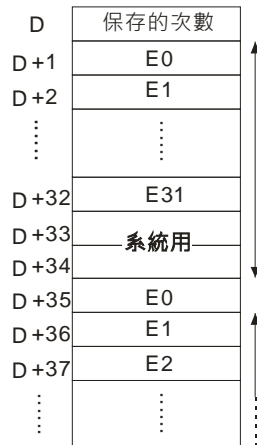
API	指令码			操作数								功能					
1905		EPOP	P	D								变址寄存器读出					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●				●				
脉冲执行型										16 位指令 (3 steps)			32 位指令				
AH500										AH500			—				

符号：



指令说明：

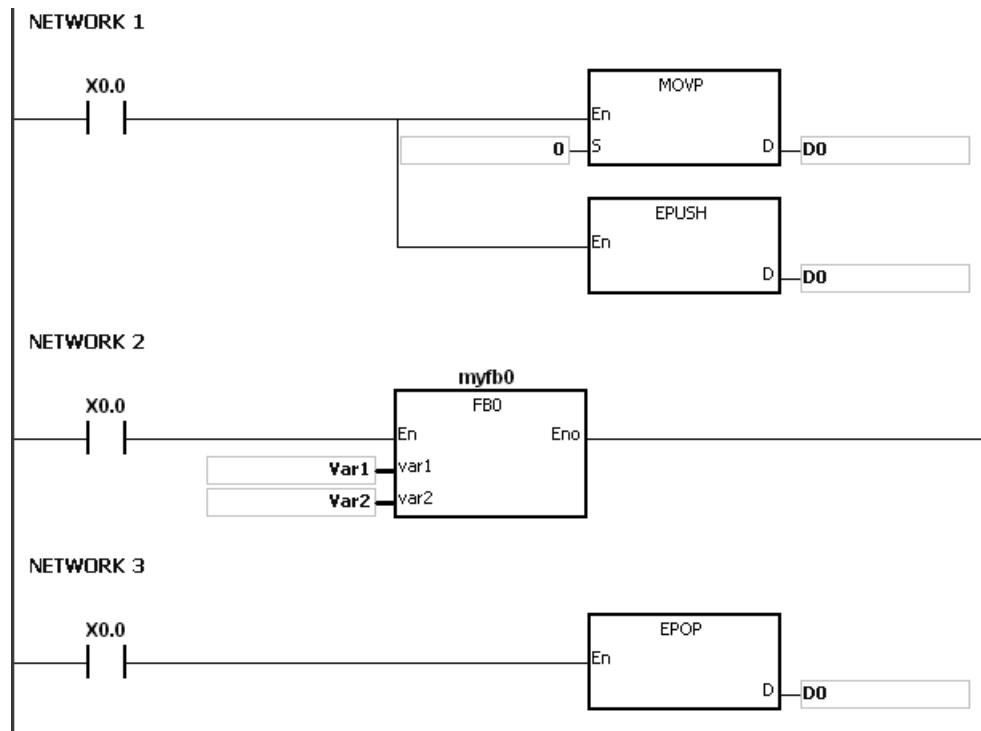
1. 将 **D** 的内容值读出到装置 E0~E31，并且将 **D** 的内容值（保存的次数）减 1。
2. 每次执行存放需 34 个装置（最后的两个装置为系统保留用），因此若保存次数为 n（**D** 的内容值），则该次会读取 $D+34*(n-1)+1 \sim D+34*(n-1)+32$ 中的内容值写入 E0~E31，且保存次数 $D=n-1$ 。
3. 取出的顺序为最后放入 **D** 中的值。



6

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。

6.21 字符串处理指令

6.21.1 字符串处理指令一览表

API	指令码 (位)			P 指令	功能	STEPS
	16	32	64			
2100	BINDA	DBINDA	–	✓	有号数十进制→ASCII 变换	5
2101	BINHA	DBINHA	–	✓	BIN 十六进制→十六进制 ASCII 变换	5
2102	BCDDA	DBCDDA	–	✓	BCD→ASCII 变换	5
2103	DABIN	DDABIN	–	✓	有号数十进制 ASCII→有号 数十进制 BIN 变换	5-11
2104	HABIN	DHABIN	–	✓	十六进制 ASCII→十六进制 BIN 变换	5-11
2105	DABCD	DDABCD	–	✓	BCD ASCII→BCD 变换	5-11
2106	\$LEN	–	–	✓	计算字符串长度	5-11
2107	\$STR	\$DSTR	–	✓	BIN→String	7
2108	\$VAL	\$DVAL	–	✓	String→BIN	7-13
2109	\$FSTR	–	–	✓	Float→String	7-8
2110	\$FVAL	–	–	✓	String→Float	5-11
2111	\$RIGHT	–	–	✓	从右边读取字符串	7-13
2112	\$LEFT	–	–	✓	从左边读取字符串	7-13
2113	\$MIDR	–	–	✓	区段读取字符串	7-13
2114	\$MIDW	–	–	✓	区段字符串取代	7-13
2115	\$SER	–	–	✓	字符串搜寻	9-21
2116	\$RPLC	–	–	✓	字符串取代	11-17
2117	\$DEL	–	–	✓	指定字符串删除	9
2118	\$CLR	–	–	✓	字符串清除	3
2119	\$INS	–	–	✓	字符串插入	9-15
2120	–	FMOD	–	✓	浮点数转 BCD 浮点数	7-8
2121	FREXP	–	–	✓	BCD 浮点数转浮点数	7

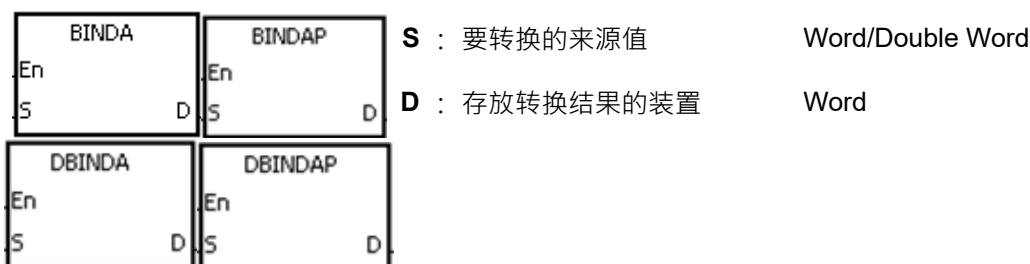
6.2.1.2 字符串处理指令说明

API	指令码			操作数								功能				
2100	D	BINDA	P	S · D								有号数十进制→ASCII 变换				

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

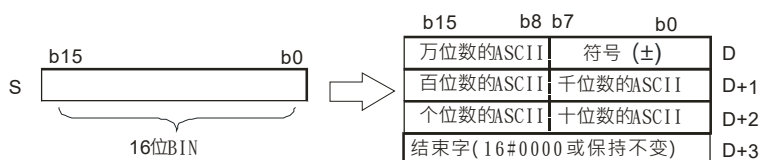
脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



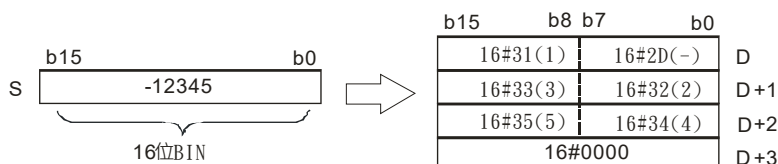
指令说明：

1. 将有号数十进制 BIN 值的数据来源 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，来控制字符串的结束字符。
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：



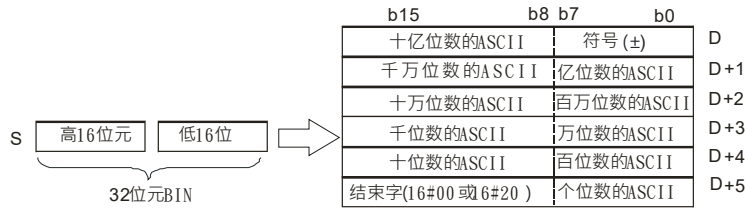
其中 SM690=OFF 时，D+3 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+3 的值保持不变。

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：

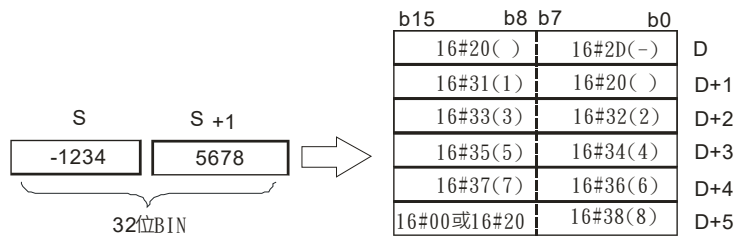


6

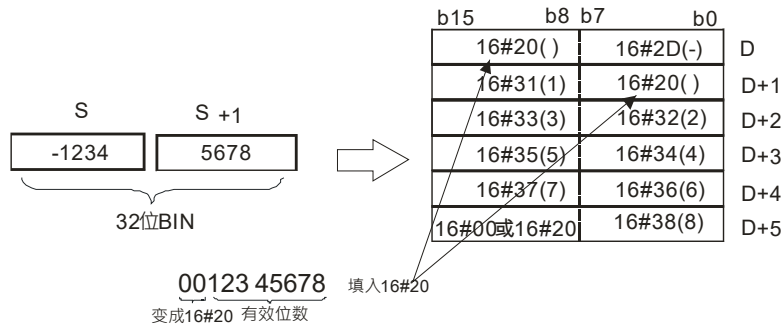
4. 32 位指令中·S 的范围-2147483648~2147483647·且固定为 11 位数的 BIN 值；D 连续占用六个 Word 装置·数据转换方式如图说明：



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：



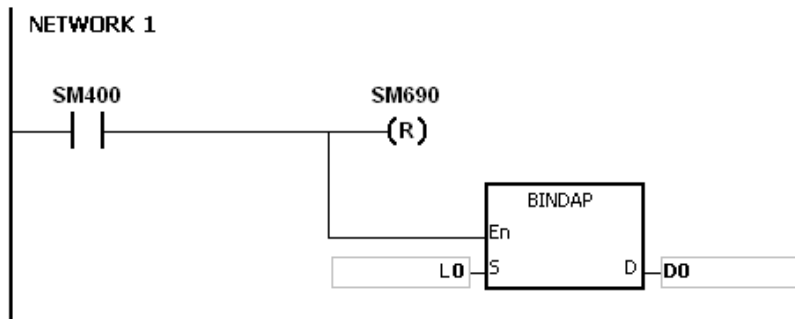
5. 若 S 内容值 (BIN) 如果有位数不足时·则 D 相对应的存放地址·会被补 16#20·例如：

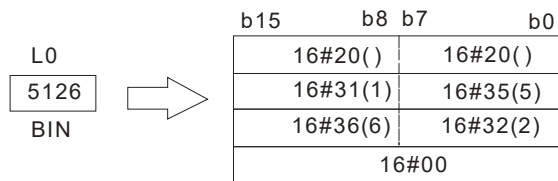


6

程序范例一：

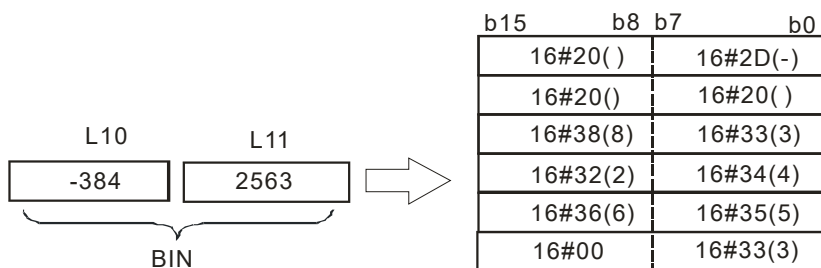
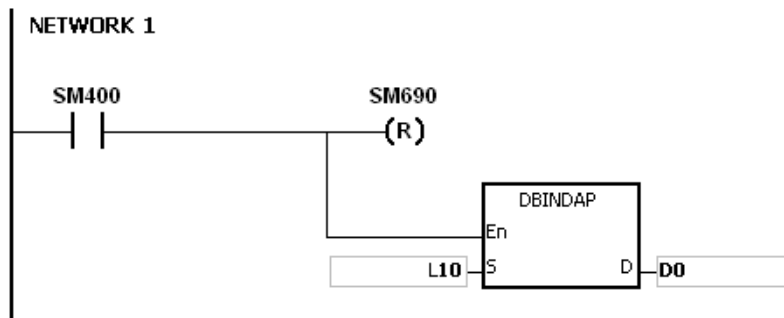
给定 L0=5126 且 SM690=OFF·PLC RUN 时 D0=16#2020、D1=16#3135、D2=16#3632、D3=16#0000。





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：

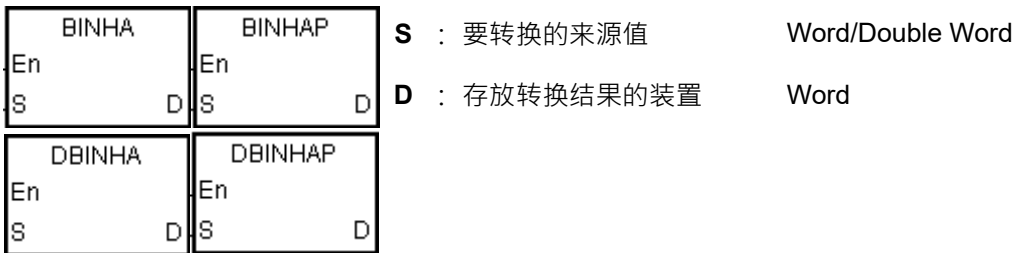
1. 16 位指令中 D+3 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
2. 32 位指令中 D+5 装置，超过装置范围时，SM0=ON，错误码 SR0=16#2003。
3. 16 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [4] of WORD/INT。
4. 32 位指令 D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [6] of WORD/INT。

API	指令码			操作数							功能				
2101	D	BINHA	P	S · D							BIN 十六进制→十六进制 ASCII 变换				

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

脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

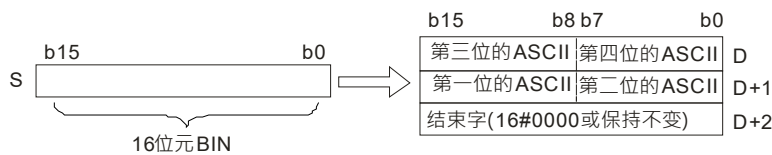
符号：



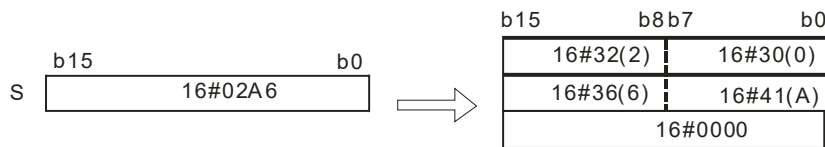
指令说明：

1. 将十六进制 BIN 值的数据来源 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，来控制字符串的结束字符。
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：

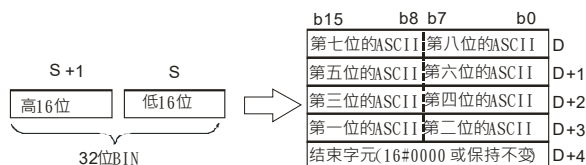
6



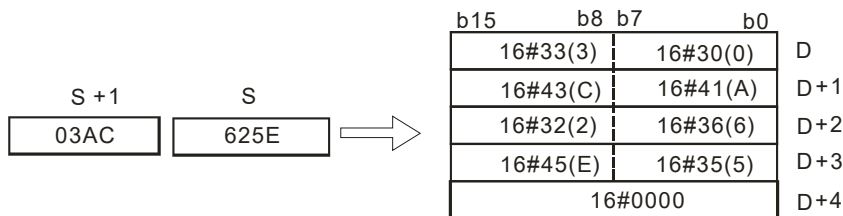
其中 SM690=OFF 时，D+2 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+2 的值保持不变。如下范例：S=16#02A6 且 SM690=OFF，则转换后的结果如下图所示。



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：

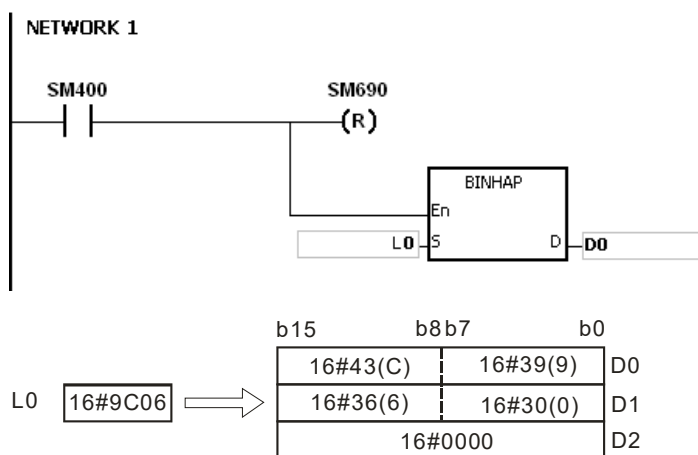


其中 SM690=OFF 时，D+4 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+4 的值保持不变。例如 S=16#03AC625E 且 SM690=OFF，则转换后的结果如下图所示。



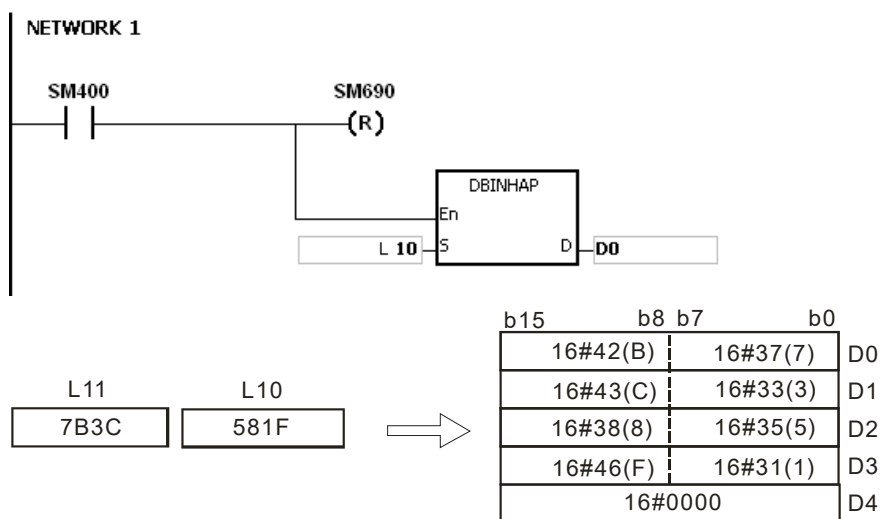
程序范例一：

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。

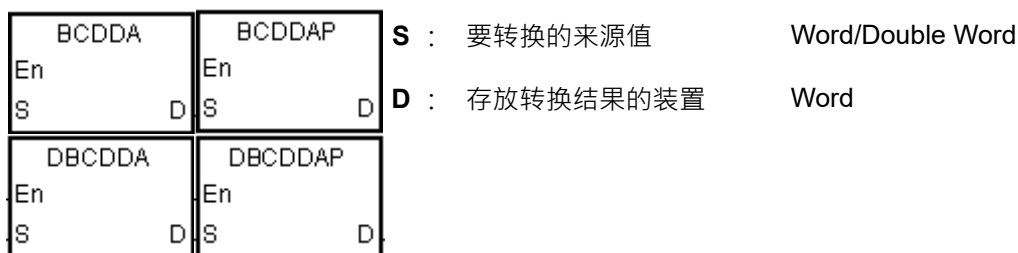
6

API	指令码			操作数							功能						
2102	D	BCDDA	P	S · D							BCD→ASCII 变换						

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

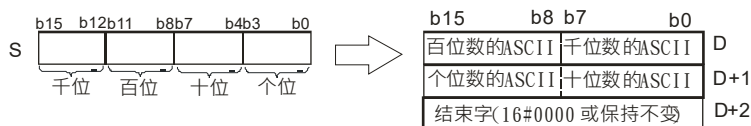
脉冲执行型	16 位指令 (5 steps)	32 位指令 (5 steps)
AH500	AH500	AH500

符号：



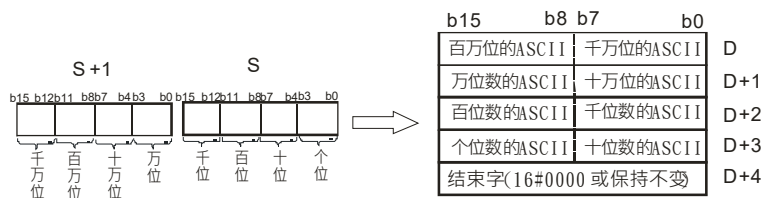
指令说明：

1. 将 BCD 值的数据来源 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，来控制字符串的结束字符。
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：



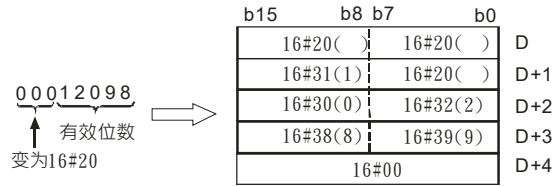
其中 SM690=OFF 时，D+2 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+2 的值保持不变。

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：



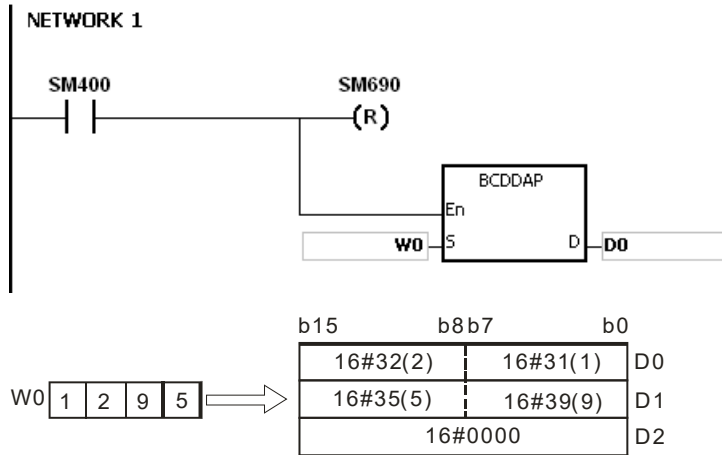
其中 SM690=OFF 时，D+5 填入字符串结束模式 16#0000；SM690=ON 时，不填入字符串结束模式，因此 D+5 的值保持不变。

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：

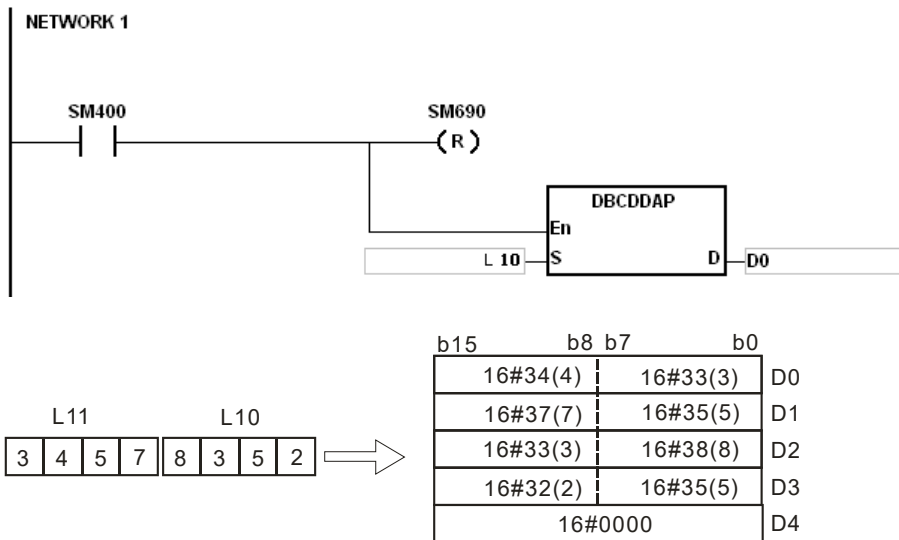
给定 L0=BCD 1295 且 SM690=OFF，PLC RUN 时 D0=16#3231、D1=16#3539、D2=16#0000。



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。

6



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。

API	指令码			操作数								功能				
2103	D	DABIN	P	S · D								有号数十进制 ASCII → 有号数十进制 BIN 变换				

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

脉冲执行型	16 位指令 (5-11 steps)	32 位指令 (5-11 steps)
AH500	AH500	AH500

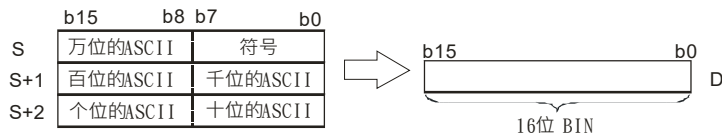
符号：



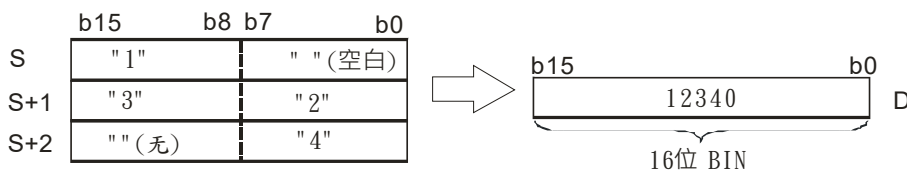
S：要转换的来源值 Word
D：存放转换结果的装置 Word/Double Word

指令说明：

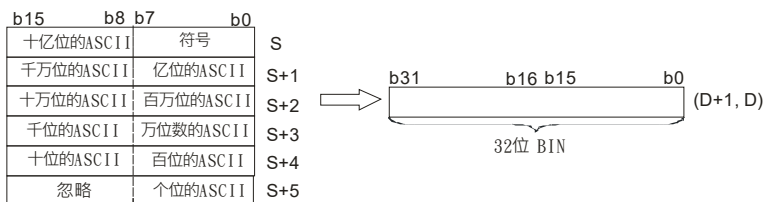
1. 将有号十进制 ASCII 值的数据来源 S 转换成有号数十进制 BIN，并储存于 D。
2. 16 位指令中，S 连续占用三个 Word 装置，且组成的 S 内容值范围 ASCII-32768~32767；若 S 来源如输入字符串，范围为“-32768”~“32767”。



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”：

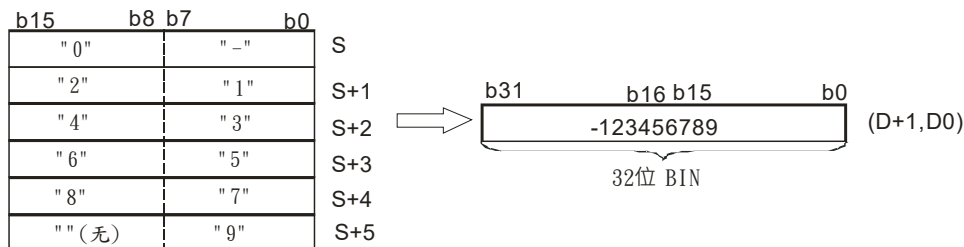


4. 32 位指令中，S 连续占用六个 Word 装置，且组成的 S 内容值范围 ASCII -2147483648~2147483647；若 S 来源如输入字符串，范围为“-2147483648”~“2147483647”。



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。

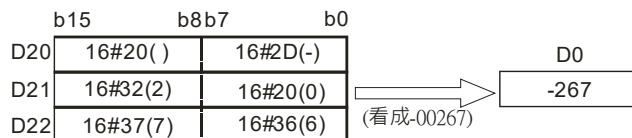
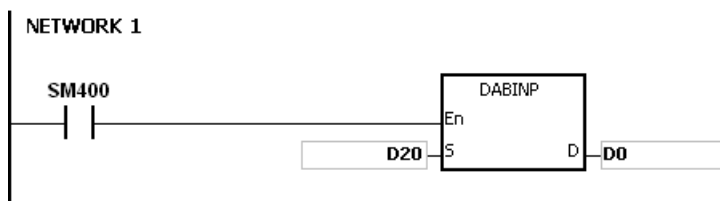
如下范例“-0123456789”：



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。

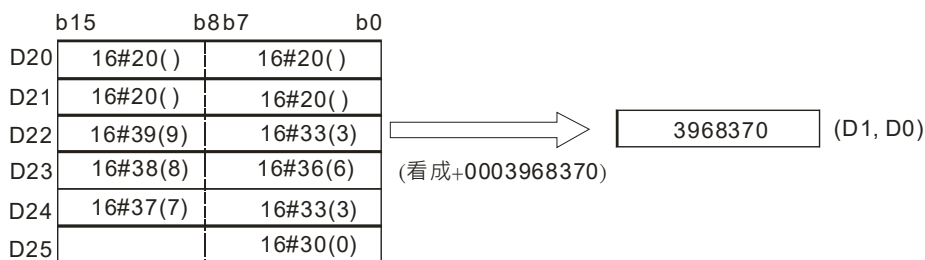
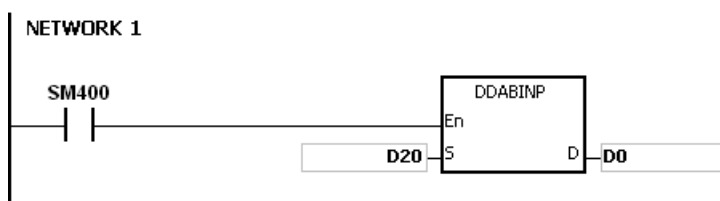
程序范例一：

给定 D20=16#202D、D21=16#3220、D22=16#3736（即 ASCII-00267），PLC RUN 时，D0=-267。



程序范例二：

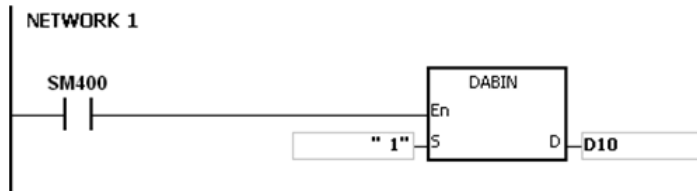
给定 D20=16#2020、D21=16#2020、D22=16#3933、D23=16#3836、D24=16#3733、D25=16#3330（即 ASCII 0003968370），PLC RUN 时，D0=3968370。



6

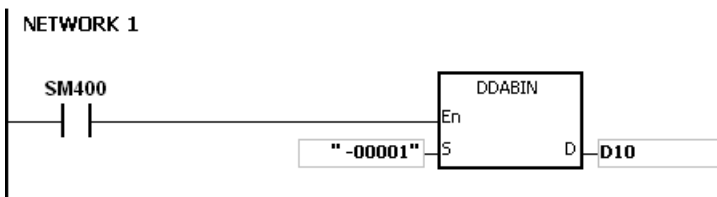
程序范例三：

S 数据来源为“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 数据来源为“-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 。

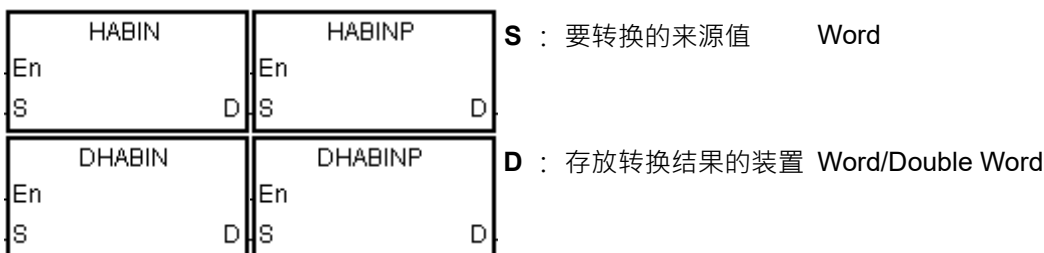
6

API	指令码			操作数								功能				
2104	D	HABIN	P	S · D								十六进制 ASCII→十六进制 BIN 变换				

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

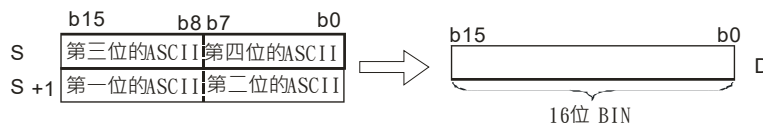
脉冲执行型	16 位指令 (5-11 steps)	32 位指令 (5-11 steps)
AH500	AH500	AH500

符号：

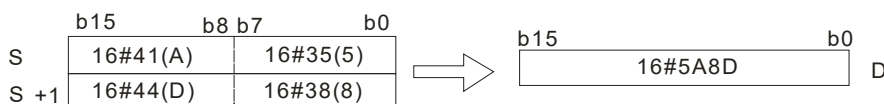


指令说明：

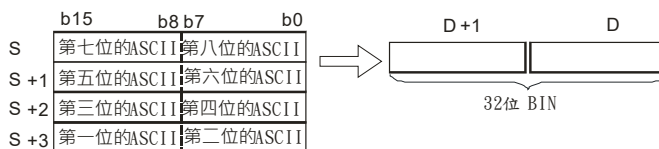
1. 数据来源 **S** 的内容十六进制 (ASCII 值) 转换成十六进制 BIN · 存于 **D**。
2. 16 位指令 **S** 连续占用二个 Word 装置 · **S** 内容值范围 ASCII 0000~FFFF；若 **S** 来源如输入字符串 · 范围为“0”~“FFFF”。



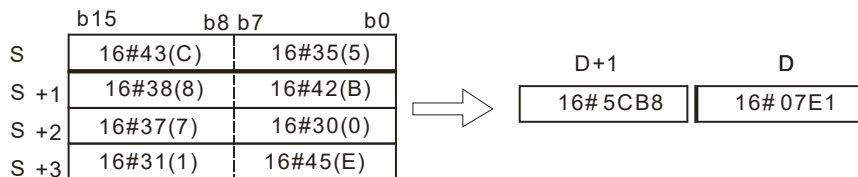
例如 S~S+1 的内容 ASCII 码分别为 5A8D · 则转换的结果为下图所示。



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”。



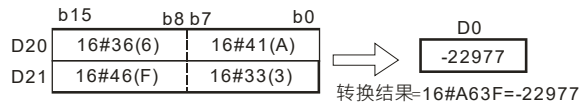
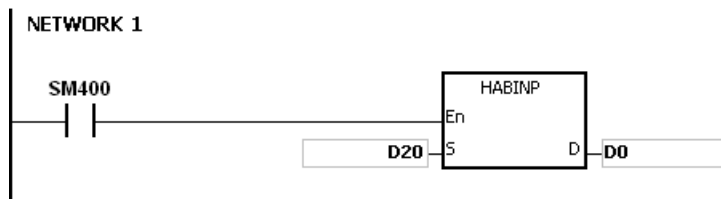
例如 S~S+3 内容 ASCII 码分别为 5CB807E1 · 则转换结果如下图所示。



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。

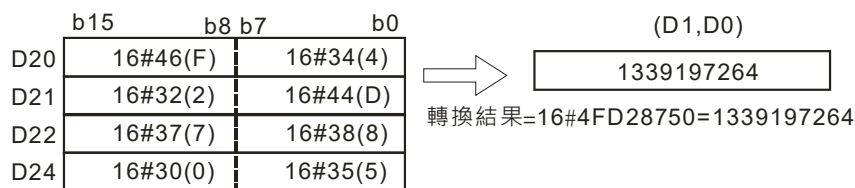
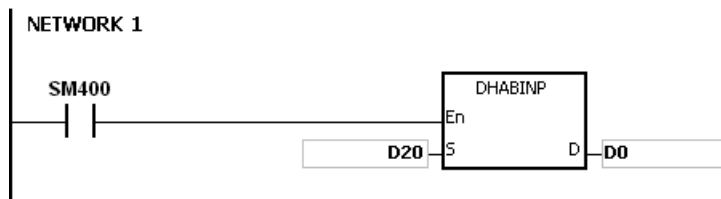
程序范例一：

给定 D20=16#3641、D21=16#4633 (即 ASCII 16#A63F=ASCII -22977) · PLC RUN 时 · D0=-22977



程序范例二：

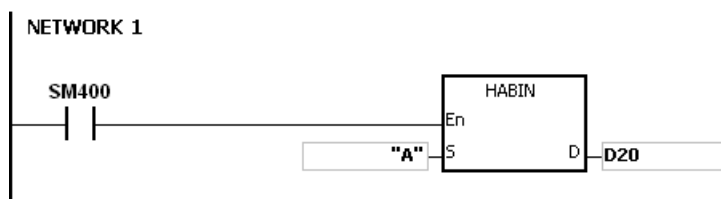
给定 D20=16#4634、D21=16#3244、D22=16#3738、D23=16#3035 (即 ASCII 16#4FD28750=ASCII 1339197264) · PLC RUN 时 · (D1 · D0) =1339197264。



6

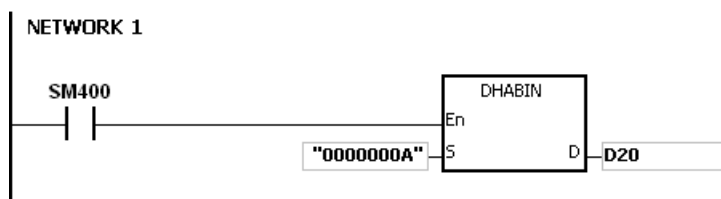
程序范例三：

S 数据来源为“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 数据来源为“0000000A” (String 类型) · PLC RUN 后 (D21 · D20) =16#0000000A=10。



补充说明：

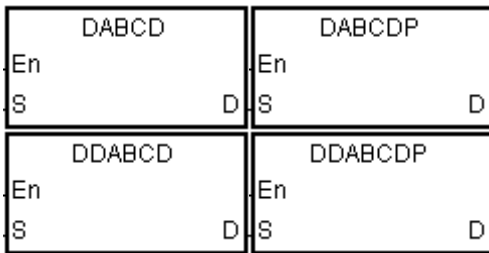
1. **S** 内容值不是 ASCII 码 16#30~ 16#39 或 16#41~ 16#46，指令不执行，SM0=ON，错误码 SR0=16#2003。
2. 16 位指令 **S** 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
3. 32 位指令 **S** 操作数，若使用 ISPSoft 宣告，则数据类型为 ARRAY [4] of WORD/INT。

API	指令码			操作数							功能						
2105	D	DABCD	P	S · D							BCD ASCII→BCD 变换						

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

脉冲执行型	16 位指令 (5-11 steps)	32 位指令 (5-11 steps)
AH500	AH500	AH500

符号：

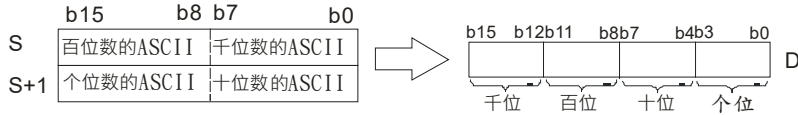


S : 要转换的来源值 Word/Double Word

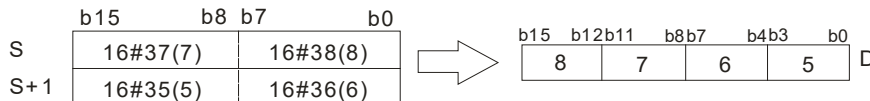
D : 存放转换结果的装置 Word/Double Word

指令说明：

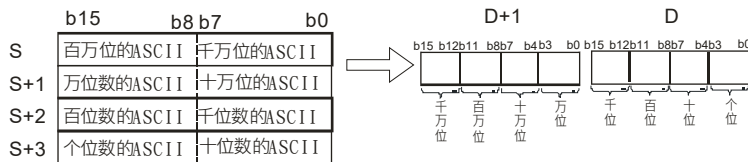
1. 数据来源 S 的内容 (ASCII 值) 转换成 BCD · 存于 D 。
2. 16 位指令 S 连续占用二个 Word 装置 · S 内容值范围 ASCII 0000~9999 ; 若 S 来源如输入字符串 · 范围为“0”~“9999” 。



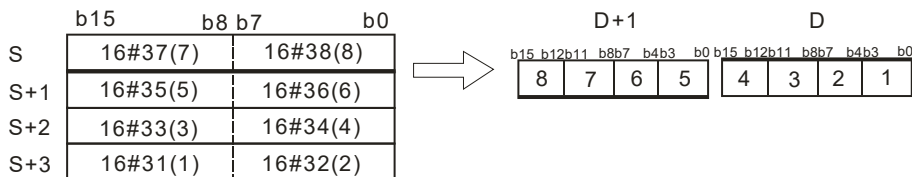
3. 例如 · S~S+N 的内容分别为 ASCII 码 8765 · 则转换结果如下图所示 。



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 。



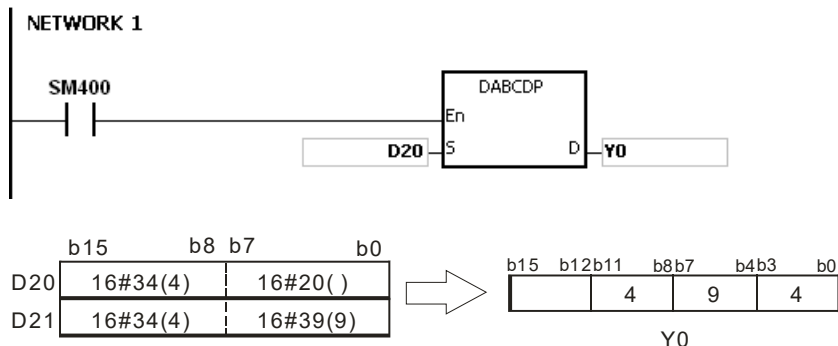
5. S 内容为 16#20 或 16#00 时 · 将当做 16#30 来做处理 。

6

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。

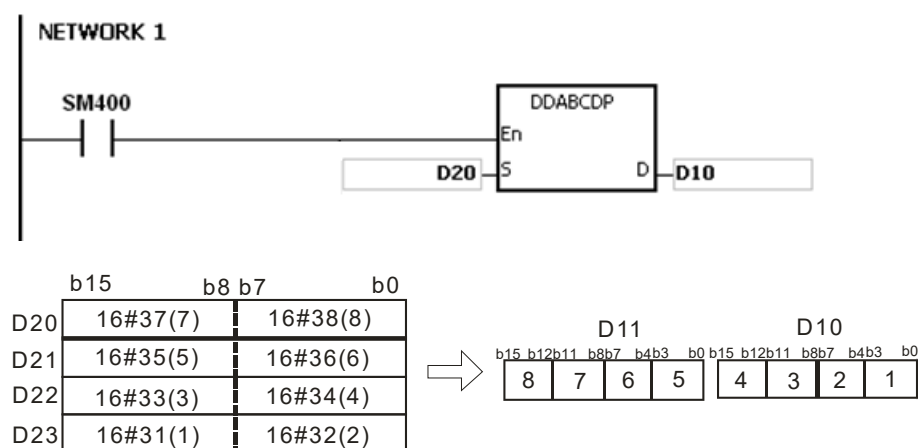
程序范例一：

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：

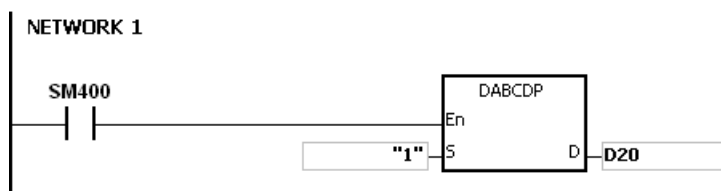
给定 D20=16#3738、D21=16#3536、D22=16#3334、D23=16#3132(即 ASCII 87654321)，PLC RUN 时，转换为 (D11、D10)=16#87654321。



6

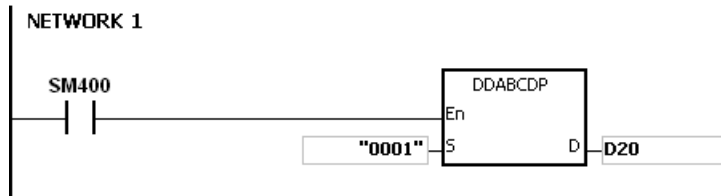
程序范例三：

S 数据来源为“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 数据来源为“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。

API	指令码			操作数								功能					
2106	\$LEN	P		S · D								计算字符串长度					

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

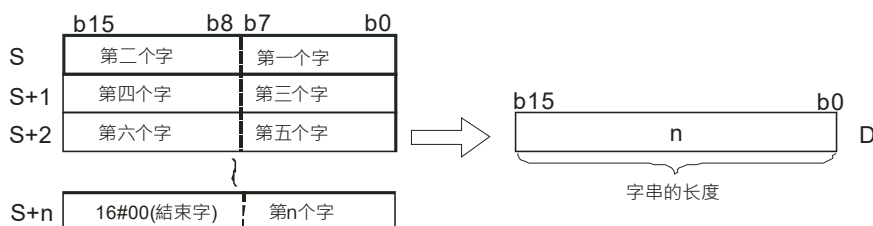
脉冲执行型	16 位指令 (5-11 steps)	32 位指令
AH500	AH500	-

符号：

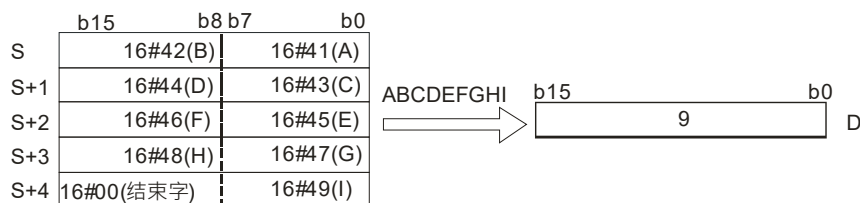


指令说明：

1. 计算数据来源 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。

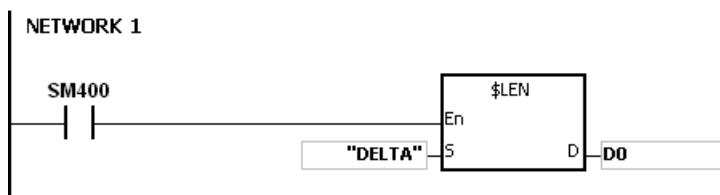


例如 S~S+n 的内容为 ABCDEFGHI，则计算结果如下图所示。



程序范例一：

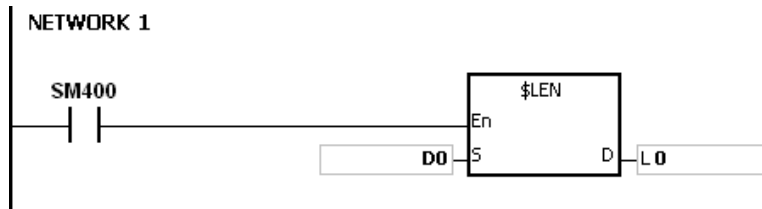
S 数据来源给定'DELTA'，则 PLC 执行后 D0=5。



程序范例二：

S 数据来源给定 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 (结束字符)	16#41 (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。

API	指令码			操作数							功能				
2107	D	\$STR	P	$S_1 \cdot S_2 \cdot D$							BIN→String				

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

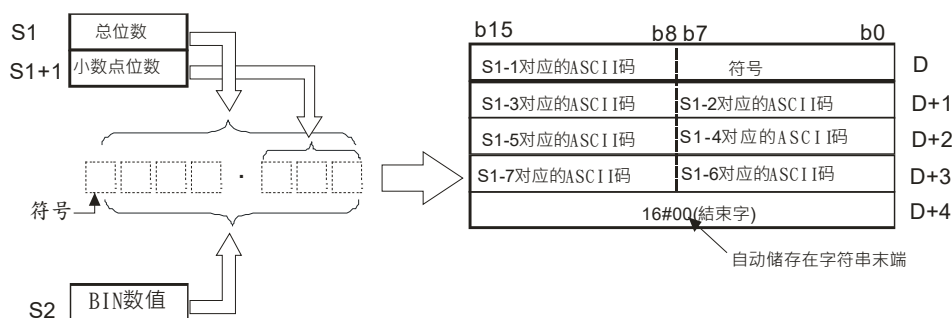
脉冲执行型	16 位指令 (7 steps)	32 位指令 (7 steps)
AH500	AH500	AH500

符号：

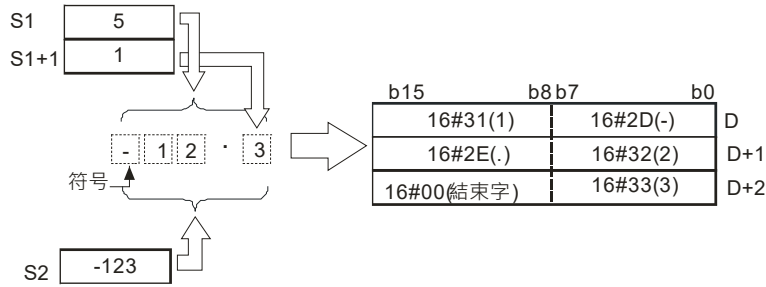
En	\$STR	En	\$STRP	D	S_1 : 要转换的位数装置起 始编号	Word/Double Word
S1		S1				
S2		S2				
En	D\$STR	En	D\$STRP	D	S_2 : 要被转换的来源	Word/Double Word
S1		S1				
S2		S2				
				D	存放转换结果的装置 起始编号	Word

指令说明：

- S_2 的内容加上一个小数点，小数点后的位数由 S_{1+1} 所指定，并将转换后的结果依照 S_1 所指定的总长度存于 D。
- \$STR :**
 S_1 总长度范围=2~8。
 S_{1+1} 所指定的小数点后的位数，范围=0~5，而且 S_{1+1} 必须 $\leq S_1-3$ 。
 S_2 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。

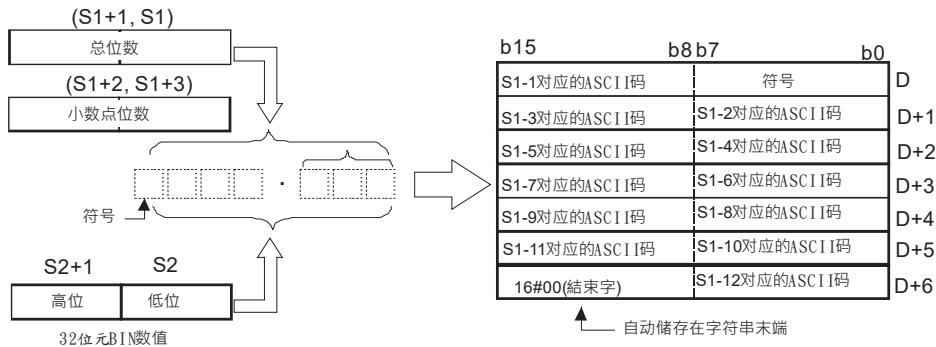


3. D\$STR :

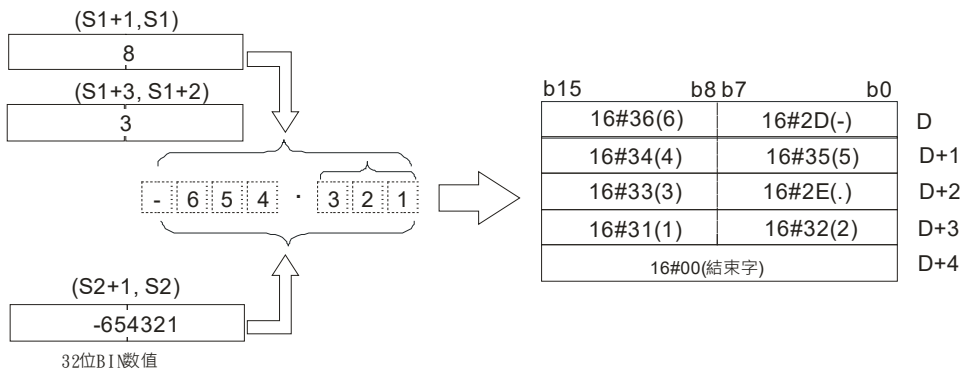
S₁ 总长度范围=2~13。

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。

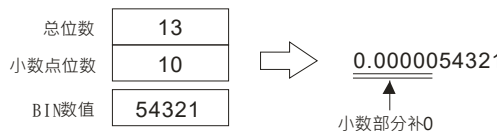
S₂ 的范围=-2147483648~2147483647。



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。

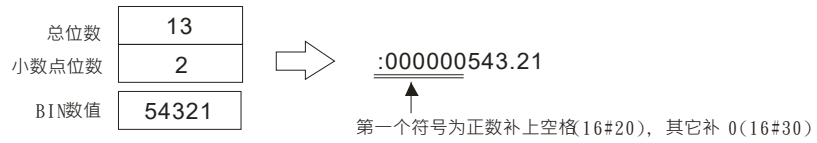


4. 如果 S₂ 为正数，则 D 的符号位置被填入 16#20，为负数，填入 16#2D。
5. D 的小数点位置被填入 16#2E。
6. 如果 S₁+1 的位数>S₂的位数时，空出的位数自动补 0。

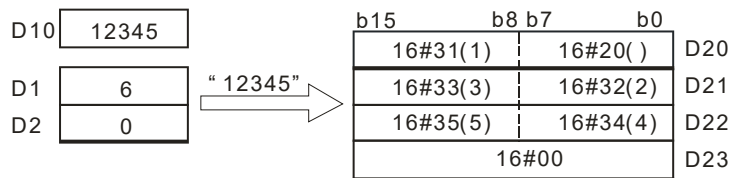
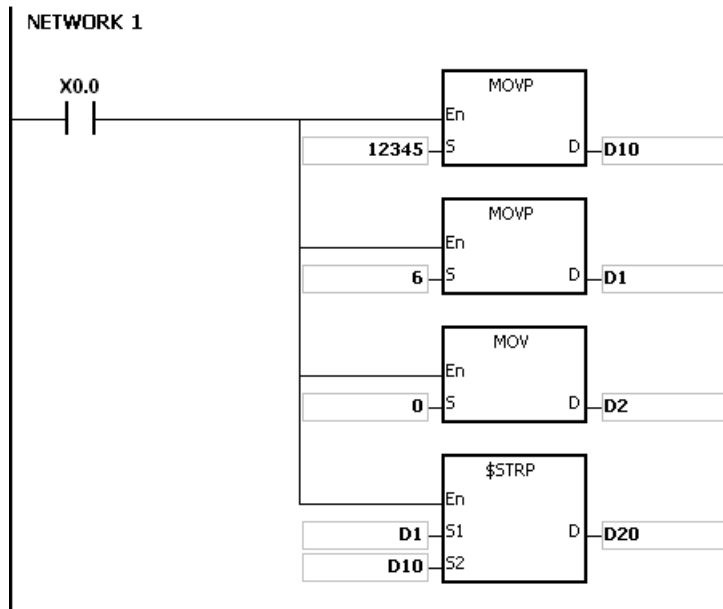


6

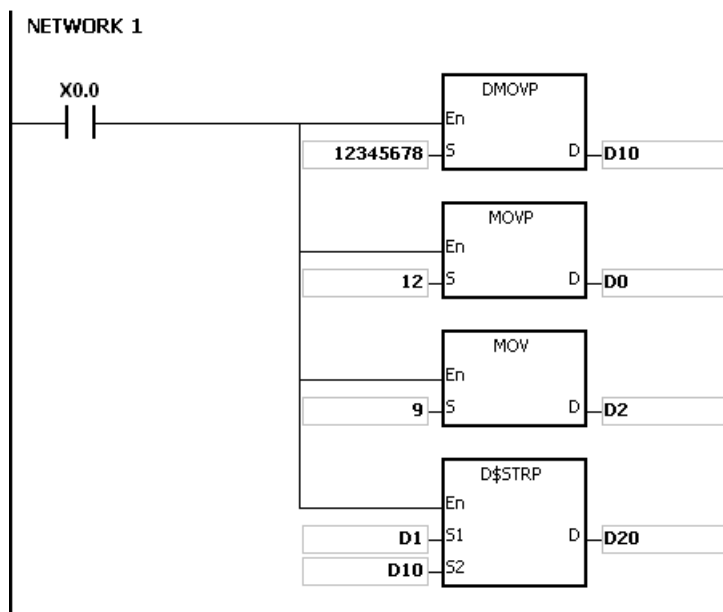
7. 如果 S_1 总位数 $> S_2$ 加上小数点跟符号的位数时，空出的位数自动补上‘0’。

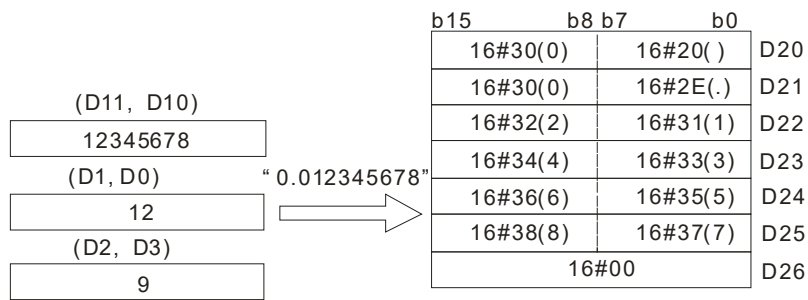


程序范例一：



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+1} 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+1} 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。

6

API	指令码			操作数								功能				
2108	D	\$VAL	P	S · D ₁ · D ₂								String→BIN				

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

脉冲执行型	16 位指令 (7-13 steps)	32 位指令 (7-13 steps)
AH500	AH500	AH500

符号：

\$VAL		\$VALP	
En		En	
S		S	
	D1		D1
	D2		D2

D\$VAL		D\$VALP	
En		En	
S		S	
	D1		D1
	D2		D2

S : 要转换的来源值 Word
 D₁ : 存放转换后的位数 Word/Double Word
 D₂ : 存放转换后的 BIN 值 Word/Double Word

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。

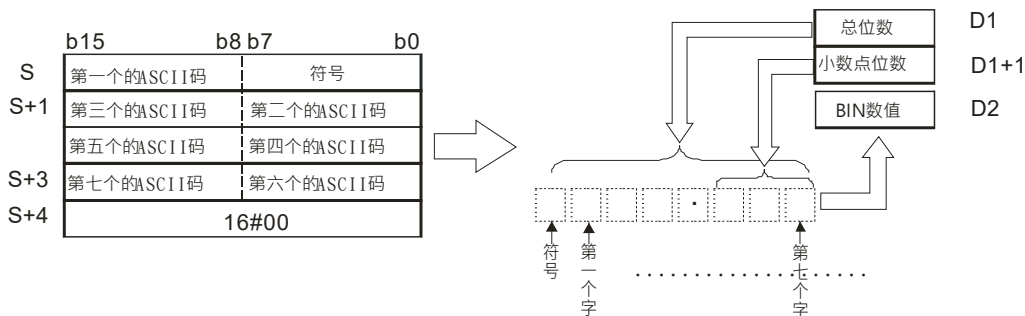
2. \$VAL：

S 最多占用五个 Word 装置。

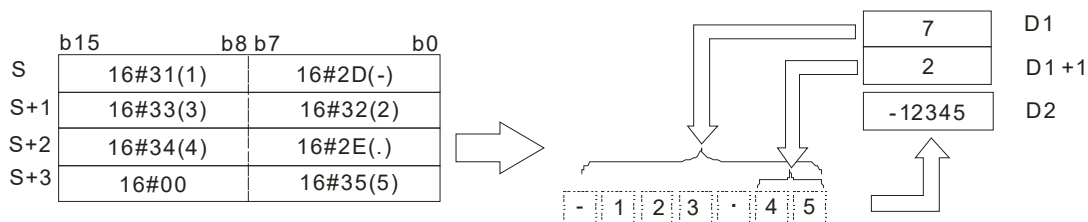
S 字符串总长度范围=2~8。

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。

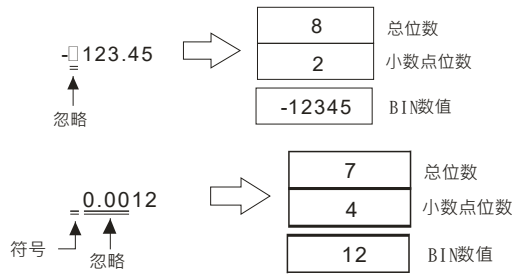
6



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** :

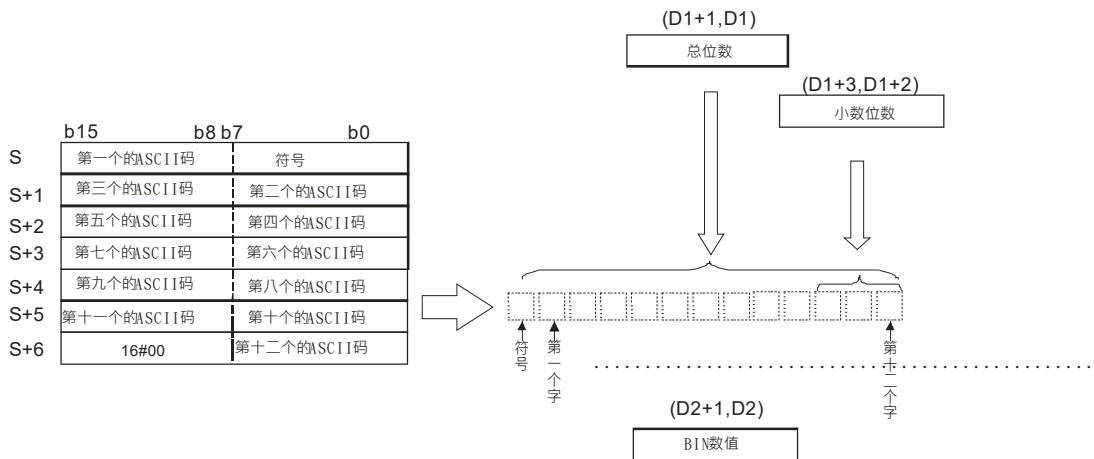
S 最多占用七个 **Word** 装置。

字符串总长度范围=2~13。

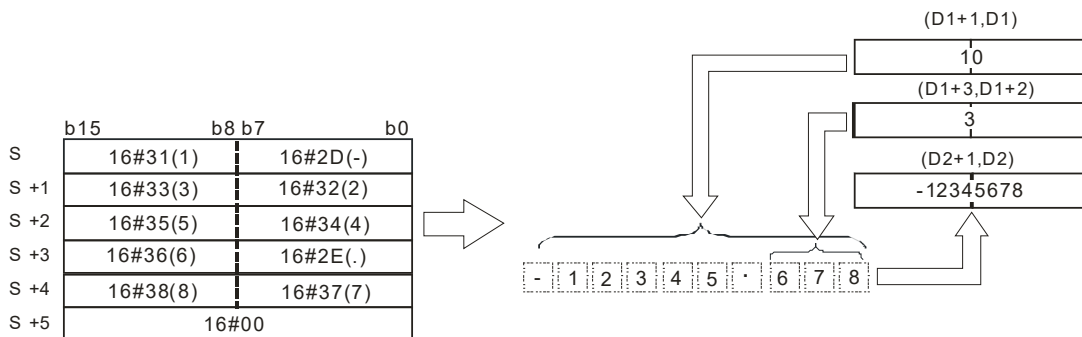
如果 **S** 字符串中有小数点，则小数点 **16#2E** 的位置，需在 3~ (字符串总长度-1) 之中。

忽略小数点的点 (**16#2E**) 不看时，则 **S** 字符串的范围=-2147483648~2147483647。

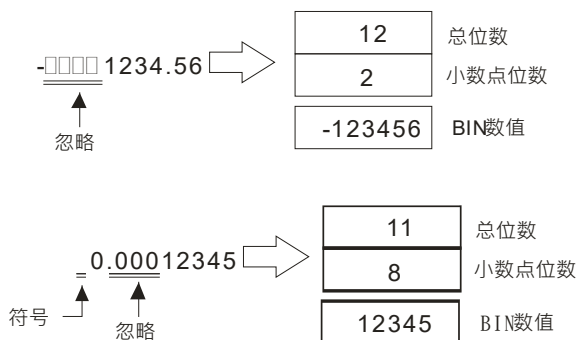
6



例如 **S~S+n** 的内容分别为-12345.678，则运算结果如下图所示。



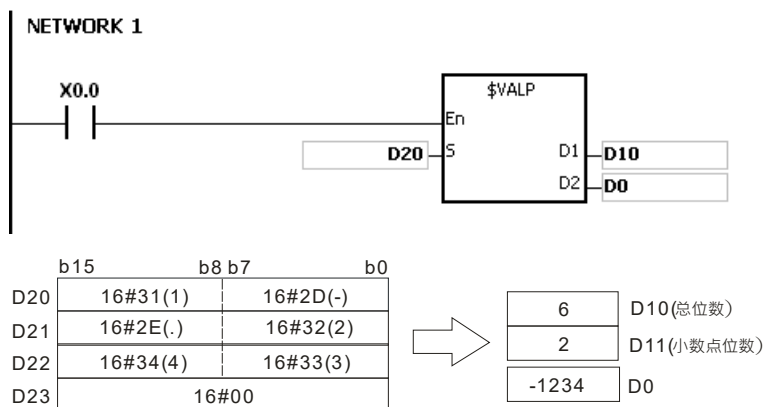
S 字符串在符号与第一个非零数值之间，如果有 16#20 或 16#30，则在转换成 BIN 数值时，会忽略 16#20 或 16#30。



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。

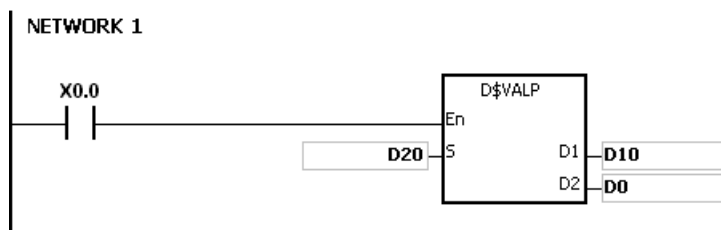
4. 如果 S 的符号位置被填入 16#20、16#2B、16#30，则转换结果为正数，填入 16#2D，则转换结果为负数。
5. S 字符串除了符号、小数点，还有可能会被忽略的部份可以填入 16#20 或 16#30，其余只能填入 16#30~16#39。

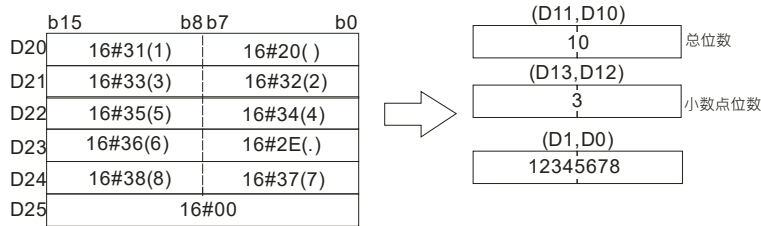
程序范例一：



6

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。

API	指令码		操作数										功能			
2109	\$FSTR	P	$S_1 \cdot S_2 \cdot D$										Float→String			

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

脉冲执行型	16 位指令 (7-8 steps)	32 位指令
AH500	AH500	-

符号：

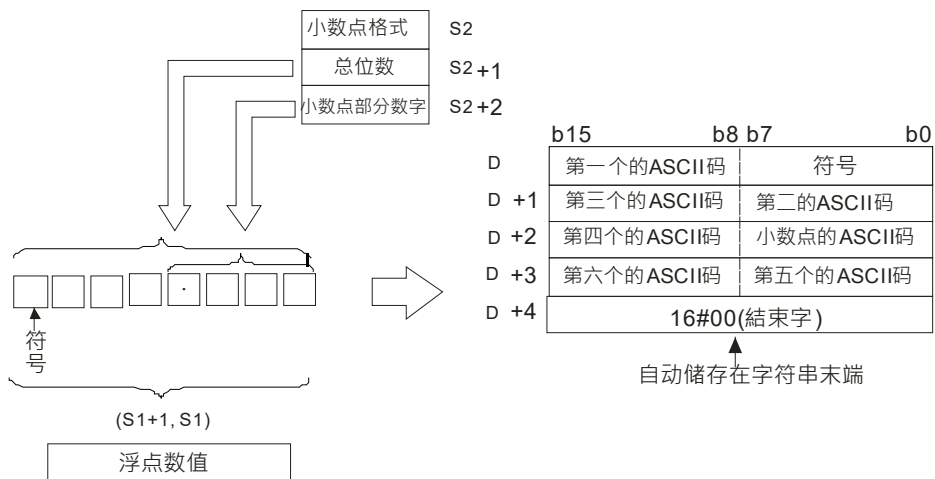
\$FSTR		\$FSTRP		S_1 : 要被转换的来源浮点数值	Double Word
En		En		S_2 : 要转换的显示设定装置起始编号	Word
S1	D	S1	D	D : 存放转换结果的装置起始编号	Word
S2		S2			

指令说明：

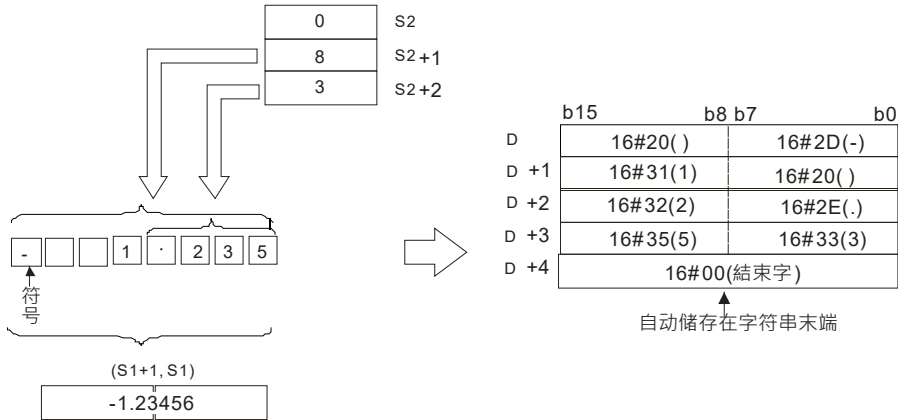
1. 依据 S_2 的显示设定，将 S_1 的浮点值转换成字符串，并将结果存于 D。
2. 转换的结果依据 S_2 的显示设定而有所不同。
3. 总位数 S_{2+1} 设定范围为 2~24。

S2	0: 小数格式 1: 指数格式
S_{2+1}	总位数
S_{2+2}	小数部分位数

4. 当使用小数格式 ($S_2=0$) 时：

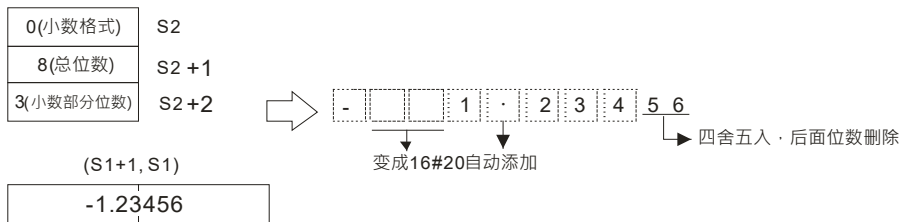


举例说明，总位数 8 位，小数点位数 3 位，数值为 -1.23456，则运算结果如下所示：

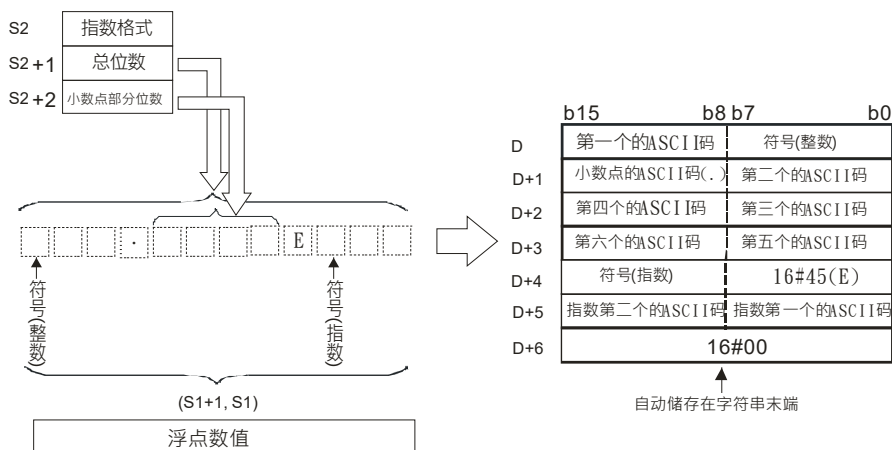


- 总位数 S_{2+1} 可以指定的范围如下：
 当小数点后的位数 $S_{2+2}=0$ 时， $2 \leq S_{2+1} \leq 24$ ，整数部分的位数 ≤ 23 。
 当小数点后的位数 S_{2+2} 不等于 0 时， $S_{2+2}+3 \leq S_{2+1} \leq 24$ ，整数部分的位数 $\leq 22-S_{2+2}$ 。
- 小数点后的位数 S_{2+2} 可以指定的范围为：0~7，但是 S_{2+2} 不等于 0 时， S_{2+2} 必需要 $\leq (S_{2+1} \text{ 总位数}) - 3$ 。
- 如果来源浮点数 S_1 为正数，则 D 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，将自动加上结束字符 16#00。

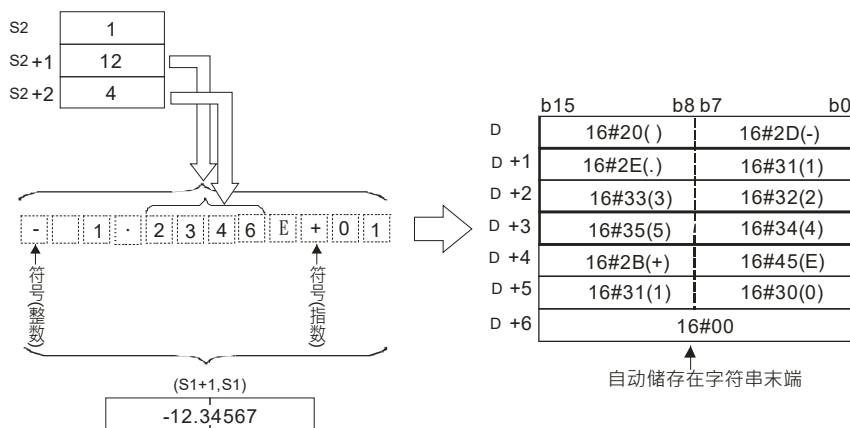
6



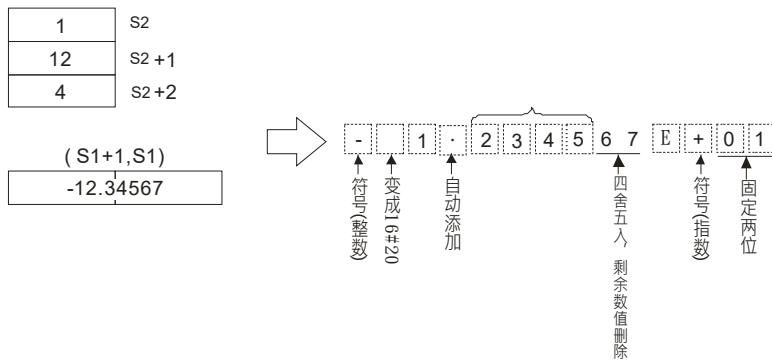
5. 当使用指数格式 ($S_2=1$) 时:



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:

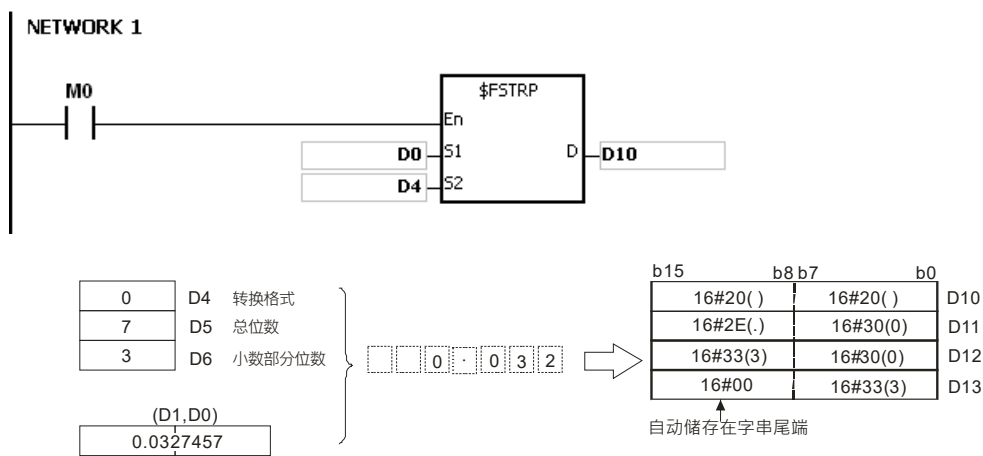


- 总位数 S_{2+1} 可以指定的范围如下:
 当小数点后的位数 $S_{2+2}=0$ 时, $6 \leq S_{2+1} \leq 24$ 。
 当小数点后的位数 S_{2+2} 不等于 0 时, $S_{2+2}+7 \leq S_{2+1} \leq 24$ 。
- 小数点后的位数 S_{2+2} 可以指定的范围为: 0~7, 但是 S_{2+2} 不等于 0 时, S_{2+2} 必需要 $\leq (S_{2+1} - 7)$ 。
- 如果来源浮点数 S_1 为正数, 则 D 的符号位置被填入 16#20, 为负数, 填入 16#2D。
- 整数部份固定为 1 个位数, 并且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。



程序范例一：

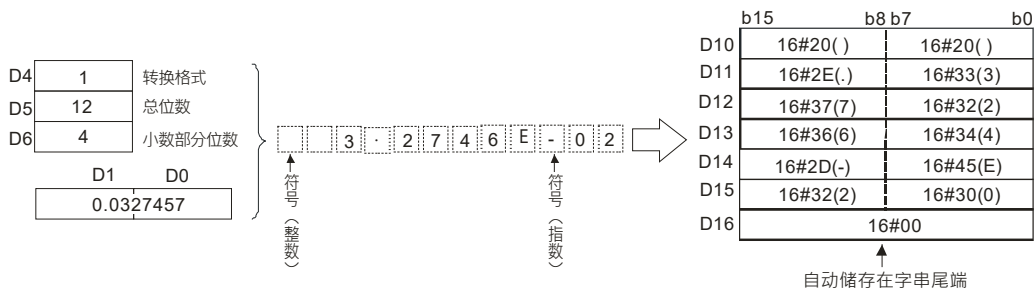
将浮点值内容 (D1 · D0) 转为小数形式 (D4=0) 的字符串。



6

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 S₁ 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 S₂ 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 S₂₊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 S₂₊₂=0 时，2 ≤ S₂₊₁ ≤ 24，整数部分的位数 ≤ 23。
当小数点后的位数 S₂₊₂ 不等于 0 时，S₂₊₂+3 ≤ S₂₊₁ ≤ 24，整数部分的位数 ≤ 22-S₂₊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， $S_{2+2}+7 \leq S_{2+1} \leq 24$ 。

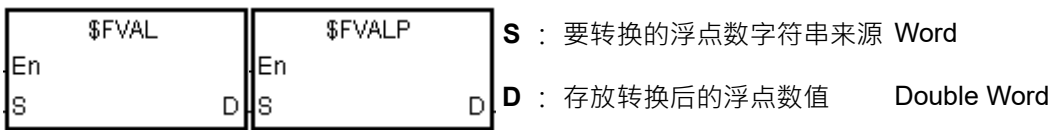
4. 当小数点后的位数 S_{2+2} 超出以下范围时，指令不执行，SM0=ON，错误码 SR0=16#2003。
 - 当使用小数格式时：
0~7，但是 S_{2+2} 必需要 $\leq (S_{2+1} \text{ 总位数}) - 3$ 。
 - 当使用指数格式时：
0~7，但是 S_{2+2} 必需要 $\leq (S_{2+1} \text{ 总位数}) - 7$ 。
5. S_2 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [3] of WORD/INT。

API	指令码			操作数							功能				
2110		\$FVAL	P	S · D							String→Float				

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

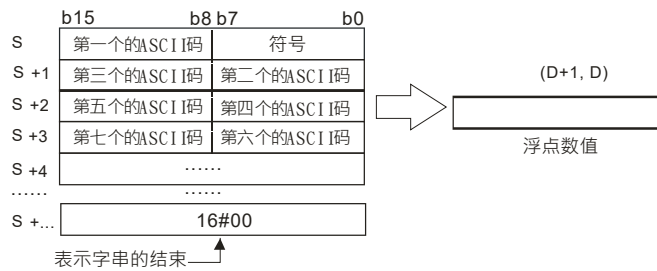
脉冲执行型	16 位指令 (5-11 steps)	32 位指令
AH500	AH500	-

符号：



指令说明：

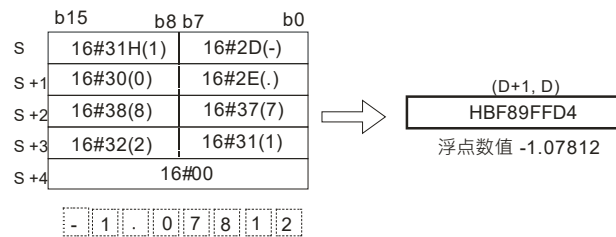
1. 将 S 的浮点数字符串转换成浮点值，并将结果存于 D。



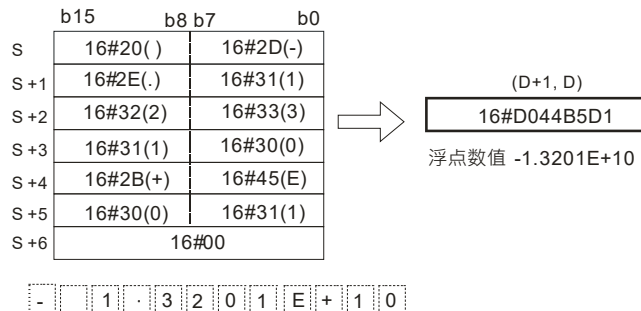
6

2. S 的浮点数字符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：

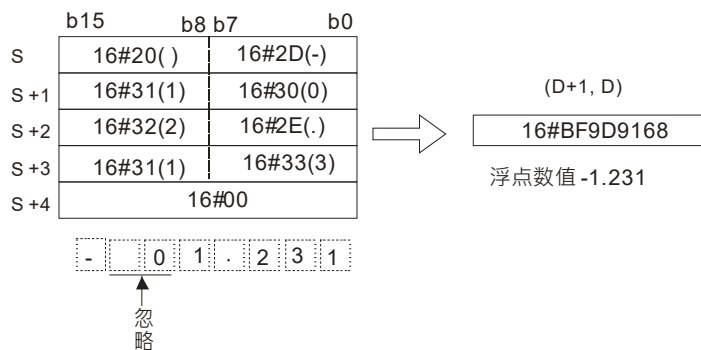


- 当使用指数格式时，范例如下：



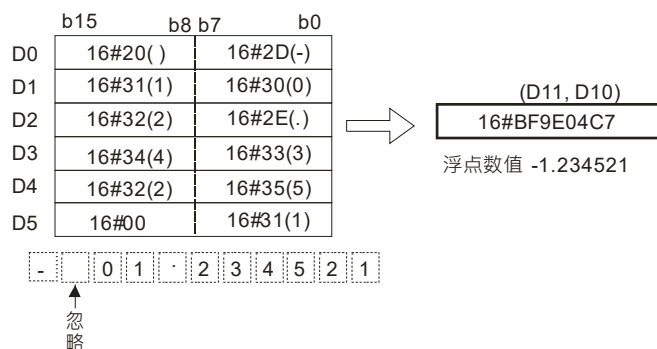
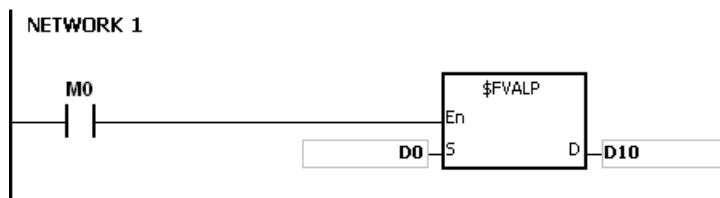
3. 如果 S 的符号位置被填入 16#20、16#30 或 16#2B，则转换结果为正数，填入 16#2D，则转换结果为负数。

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。

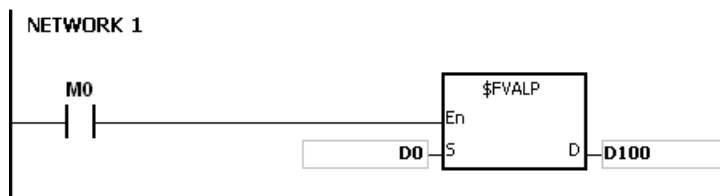


5. S 的浮点数字符串长度，最长为 24 个字。

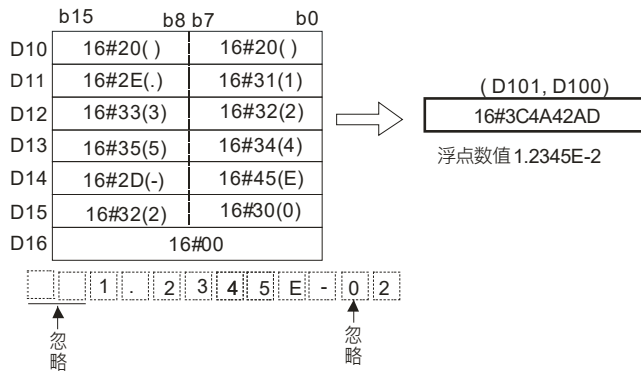
程序范例一：



程序范例二：



6



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 **S**₁ 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“0” (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。

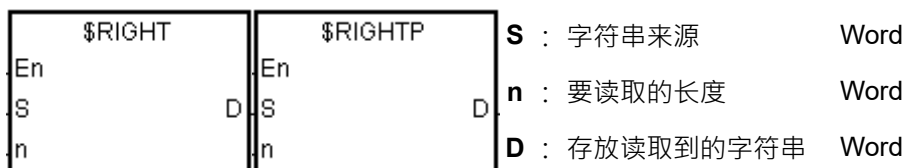


API	指令码			操作数								功能				
2111	\$RIGHT	P		S · n · D								从右边读取字符串				

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

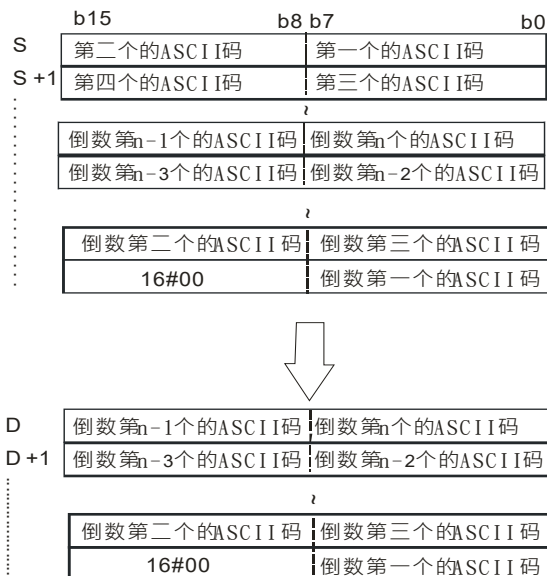
脉冲执行型	16 位指令 (7-13 steps)	32 位指令
AH500	AH500	-

符号：

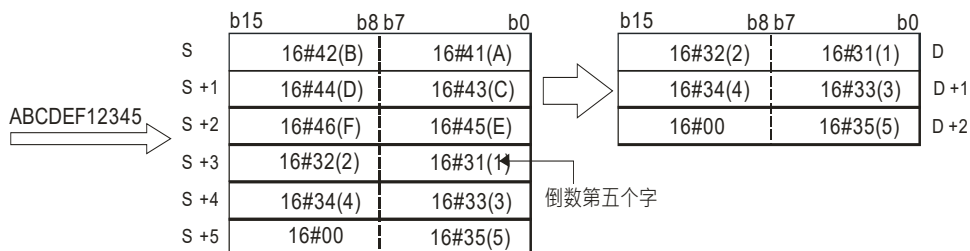


指令说明：

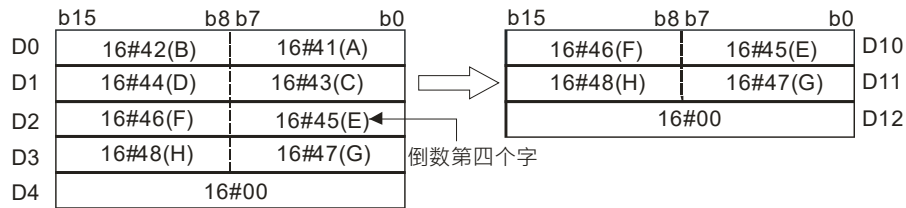
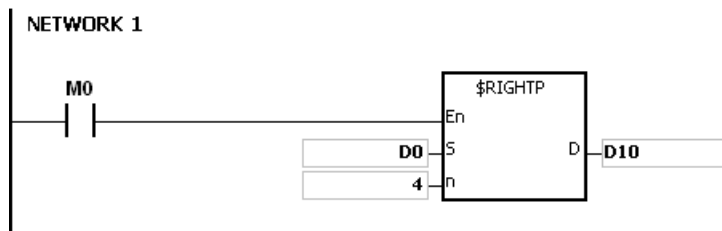
1. 从 S 字符串的最右边开始读取 n 个字，并将读取出来的字符串存放到 D。
2. 当 n=0 时，D 装置会存放 0。



例如，S 的内容为 ABCDEF12345，当 n=5 时从最右边开始读取 5 个字，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。

6

API	指令码		操作数							功能						
2112	\$LEFT	P	S · n · D							从左边读取字符串						

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

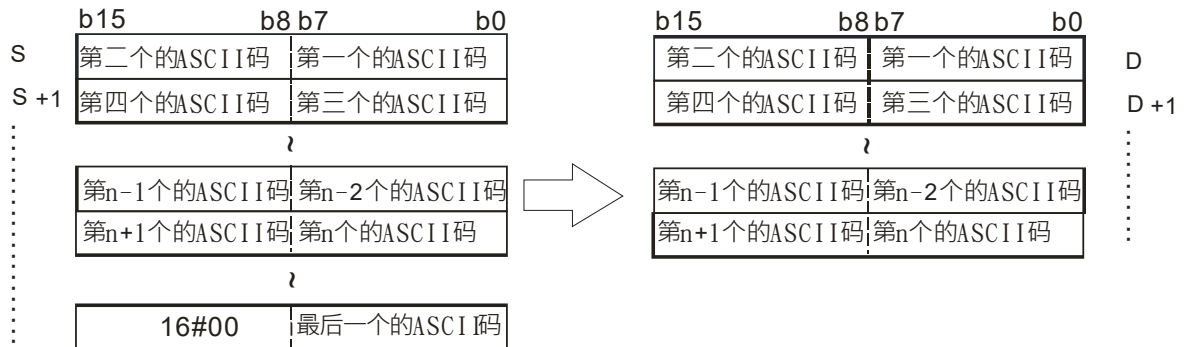
脉冲执行型	16 位指令 (7 -13 steps)	32 位指令
AH500	AH500	-

符号：

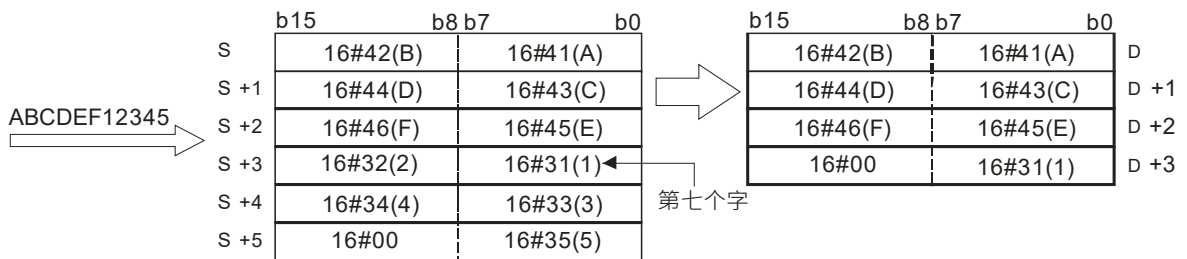
\$LEFT	\$LEFTP	S : 字符串来源 Word	
En	En		n : 要读取的长度 Word
S	S		
n	n		

指令说明：

1. 从 S 字符串的最左边开始读取 n 个字，并将读取出来的字符串存放到 D。
2. 当 n=0 时，D 装置会存放 0。

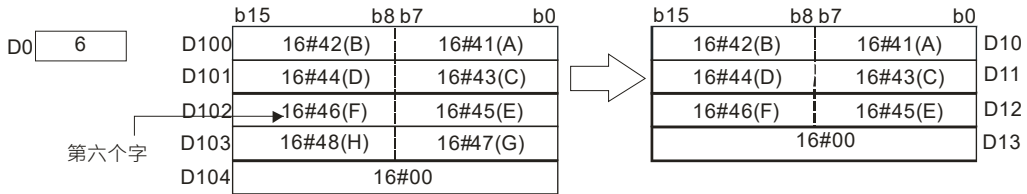
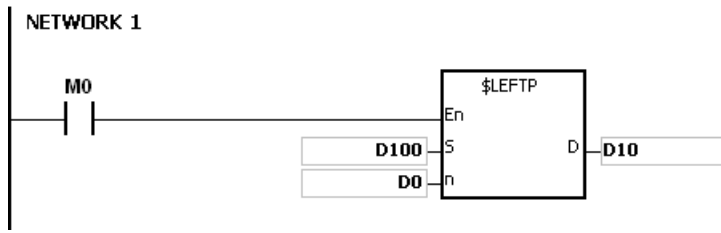


例如，S=ABCDEF12345，当 n=7 时从最左边开始读取 7 个字符，转换结果如下图所示。



程序范例：

M0=ON 时，执行\$LEFT 指令，从 D100 开始的数据开始读取，取 6 个字符后将结果储存至 D10。



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。

6

API	指令码		操作数						功能					
2113	\$MIDR	P	$S_1 \cdot S_2 \cdot D$						区段读取字符串					

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

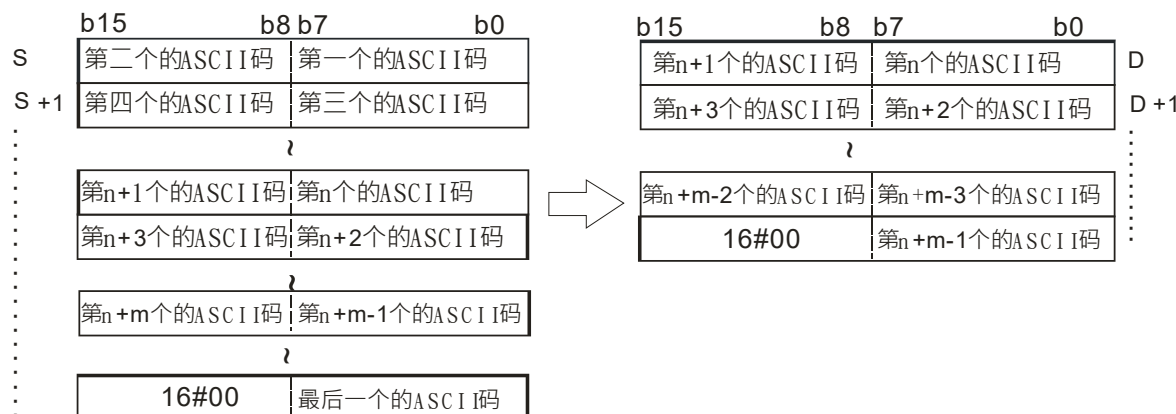
脉冲执行型	16 位指令 (7 - 13 steps)	32 位指令
AH500	AH500	-

符号：

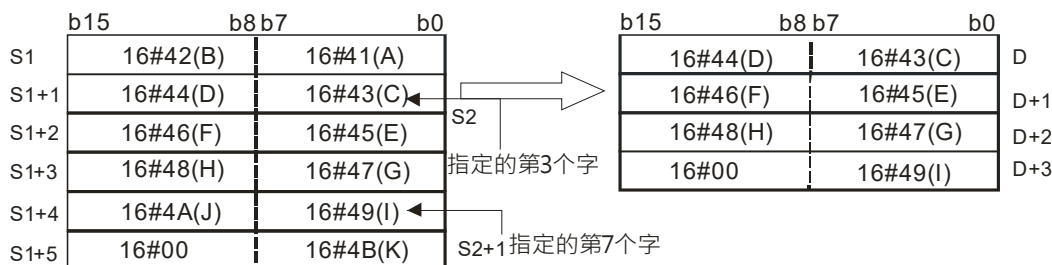
\$MIDR		\$MIDRP		S_1 : 字符串来源	Word
En		En		S_2 : 要读取的区段设定值	Word
S1	D	S1	D	D : 存放读取到的字符串	Word
S2		S2			

指令说明：

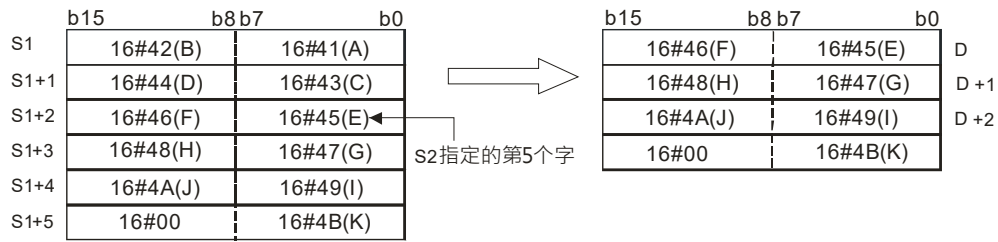
1. 将 S_1 字符串从 S_2 所指定的字开始读取 S_2+1 个字，并将读取出来的字符串存放到 D，当 $S_1=n \cdot S_2+1=m$ 如图说明。



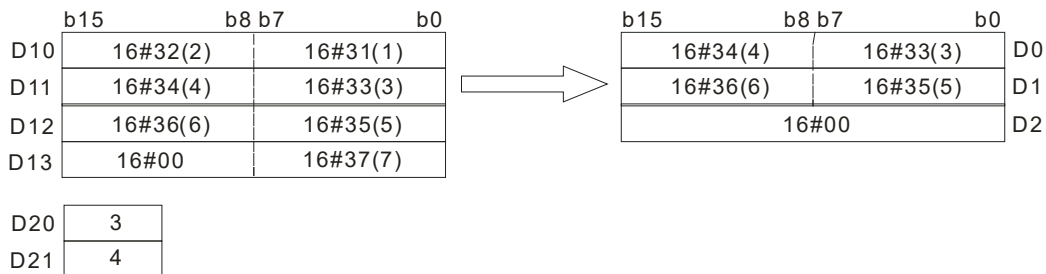
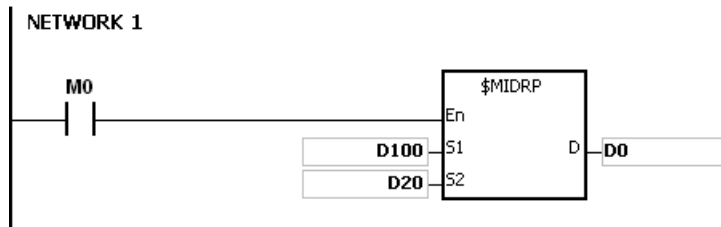
2. 例如 S=ABCDEFGHJK，当 $S_2=3$ 且 $S_2+1=7$ 时，从最左边第 3 个字符开始读取 7 个字，转换结果如下图所示。



3. 当 $S_2+1=0$ 时，指令不动作。
4. 当 $S_2+1=-1$ 时，读取 S_2 开始到 S_1 结尾的所有字符串。
5. 例如 $S_1=ABCDEFGHJK$ ，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。

6

API	指令码		操作数						功能					
2114	\$MIDW	P	$S_1 \cdot S_2 \cdot D$						区段字符串取代					

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

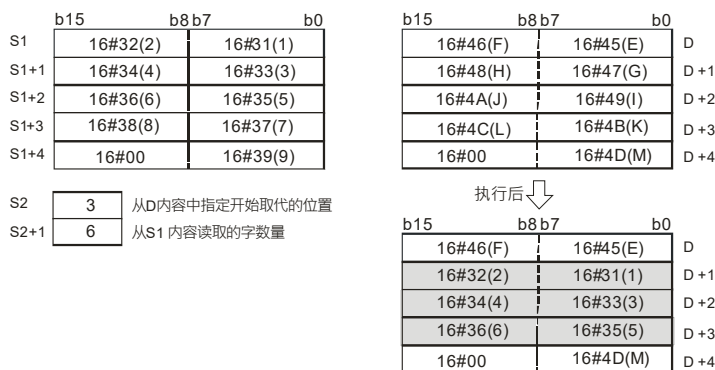
脉冲执行型	16 位指令 (7-13 steps)	32 位指令
AH500	AH500	-

符号：

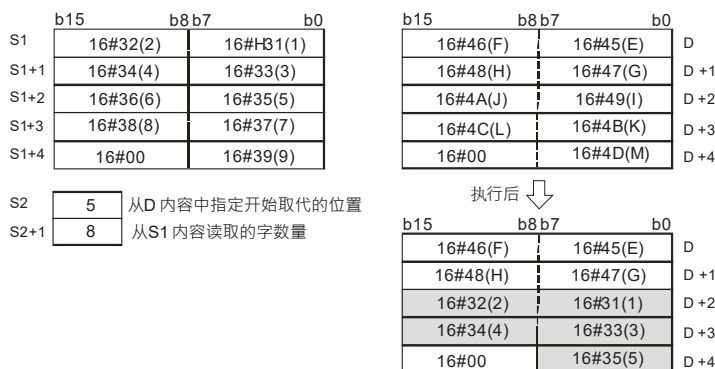
\$MIDW		\$MIDWP		S_1 : 字符串来源 Word S_2 : 要取代的区段设定值 Word D : 要被取代的字符串 Word
En		En		
S1	D	S1	D	
S2		S2		

指令说明：

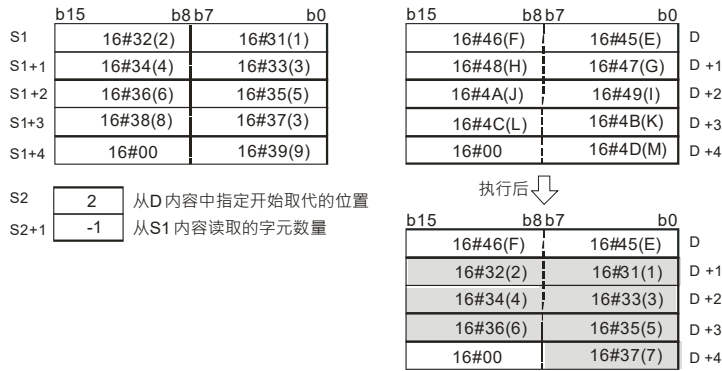
- S_2 : D 内容中开始被取代的起始位置。 S_2+1 : S_1 内容被读取出的字符数量。
- 从 S_1 字符串的第 1 个字开始读取 S_2+1 个字，并将读取出来的字符串，从 D 字符串中 S_2 所指定的字开始取代原本 D 中的字。



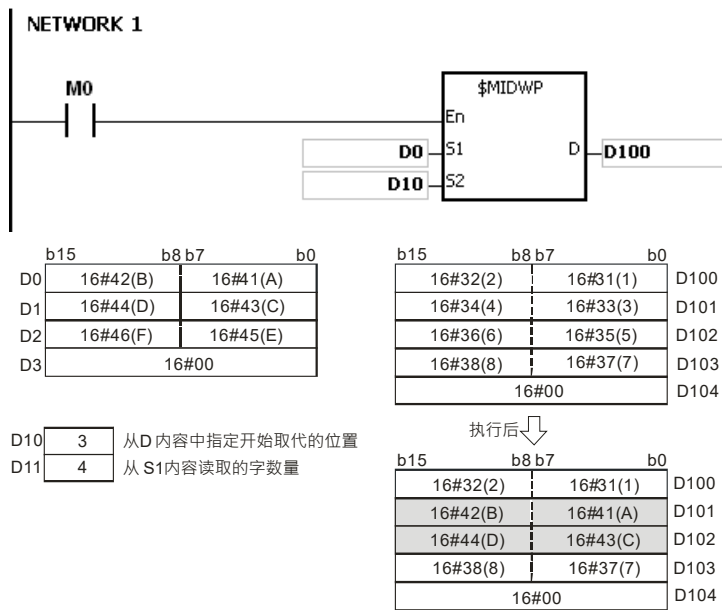
- 当 $S_2+1=0$ 时，指令不动作。
- 如果 S_2+1 所指定的长度大于 D 字符串的长度，则只取代到 D 字符串的最后一个字。



5. 当 $S_2+1=-1$ 时，读取 S_1 的所有字符串。



程序范例：



6

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。

API	指令码			操作数							功能						
2115	\$SER	P		$S_1 \cdot S_2 \cdot n \cdot D$							字符串搜寻						

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

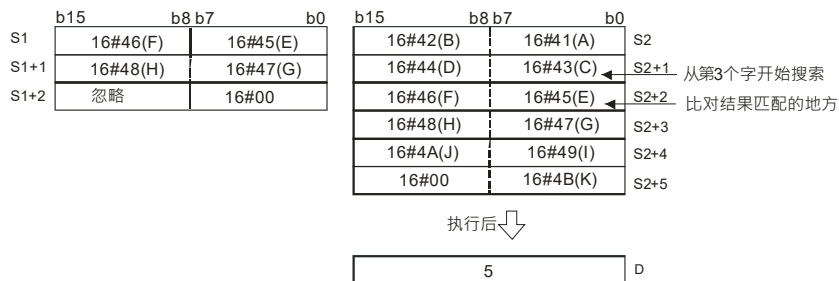
脉冲执行型	16 位指令 (9-21 steps)	32 位指令
AH500	AH500	-

符号：

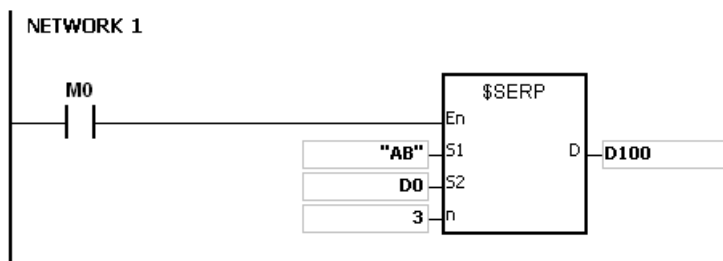
<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">\$SER</td> <td style="width: 50px;"></td> <td style="text-align: center;">\$SERP</td> </tr> <tr> <td>En</td> <td></td> <td>En</td> </tr> <tr> <td>S1</td> <td style="text-align: center;">D</td> <td>S1</td> </tr> <tr> <td>S2</td> <td></td> <td>S2</td> </tr> <tr> <td>N</td> <td></td> <td>N</td> </tr> </table>	\$SER		\$SERP	En		En	S1	D	S1	S2		S2	N		N	<p>S_1 : 要搜寻的字符串 Word</p> <p>S_2 : 要被搜寻的字符串 Word</p> <p>n : 从 S_2 的第 n 个字开始搜寻 Word</p> <p>D : 搜寻结果 Word</p>
\$SER		\$SERP														
En		En														
S1	D	S1														
S2		S2														
N		N														

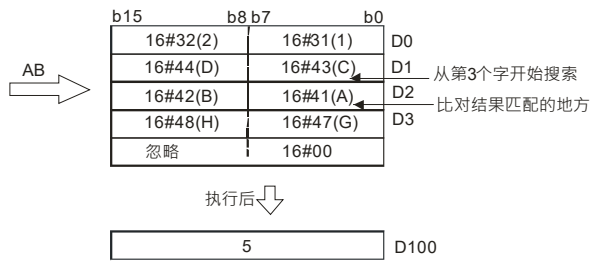
指令说明：

- 从 S_2 字符串的第 n 个字开始搜寻与 S_1 相同的字符串，如果搜寻到相同的字符串，则将此字符串的起始位置位于 S_2 字符串的第几个字，存于 D 中，如果没搜寻到相同的字符串，则 $D=0$ 。
- 例如，在 $S_2="ABCDEFGHIJK"$ 中搜寻 $S_1="EFGH"$ ，从 S_2 的第 3 个字符 ($n=3$) 开始搜寻，搜寻结果为 S_1 存在 S_2 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。

6

API	指令码		操作数							功能						
2116	\$RPLC	P	$S_1 \cdot S_2 \cdot S_3 \cdot S_4 \cdot D$							字符串取代						

装置	X	Y	M	S	T	C	HC	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 (11-17 steps)	32 位指令
AH500	AH500	-

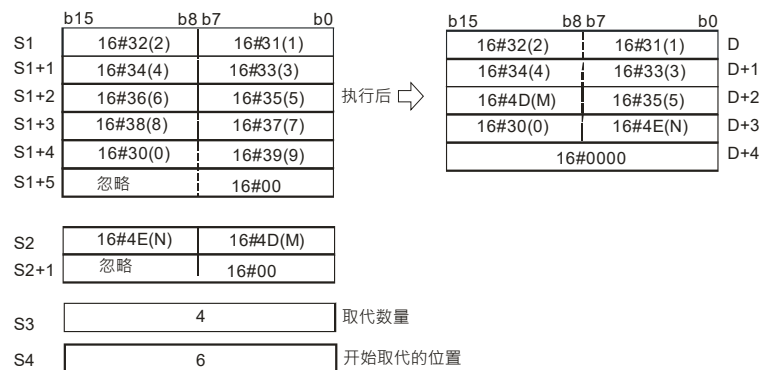
符号：

\$RPLC		\$RPLCP		S_1 : 要被取代的字符串	Word
En		En		S_2 : 要取代的新字符串	Word
S1	D	S1	D	S_3 : S_1 要被取代掉的长度	Word
S2		S2		S_4 : 从 S_1 的第 S_4 个字开始取代	Word
S3		S3		D : 取代后的字符串存放位置	Word
S4		S4			

指令说明：

- 从 S_1 字符串的第 S_4 个字开始，用 S_2 的字符串取代 S_1 中的字符串，取代的长度为 S_3 ，并将结果存于 D 中，如下图范例说明。

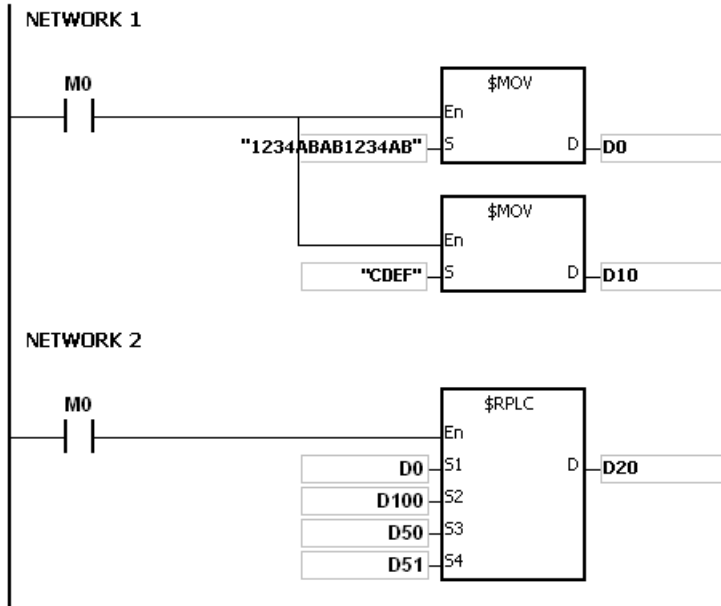
将字符串“1234567890”中的第 6 个字开始的 4 个字取代为“MN”，取代后的结果为“12345MN0”。



- S_2 字符串如果为 16#00，则此指令功能相当于将字符串删除。
- 若欲取代的字符串长度 $S_3 > S_1$ 可被取代的长度，则只取代到 S_1 的最后一个字。
- $S_3 = 0$ ，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。

6

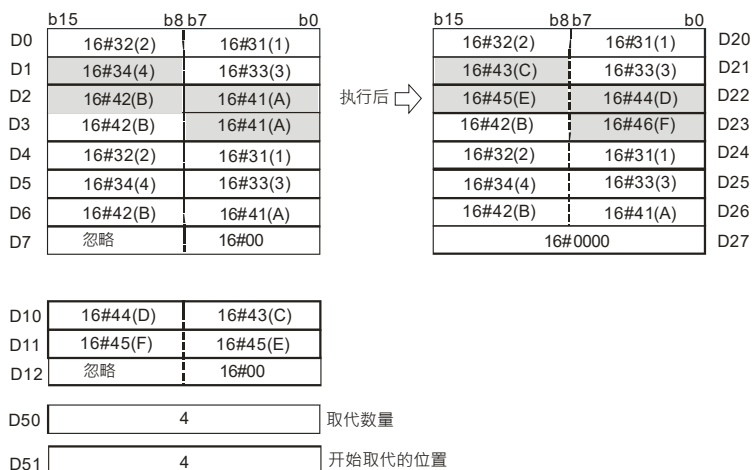
	b15	b8 b7	b0	
D0	16#32(2)		16#31(1)	D20
D1	16#34(4)		16#33(3)	D21
D2	16#42(B)		16#41(A)	D22
D3	16#42(B)		16#41(A)	D23
D4	16#32(2)		16#31(1)	D24
D5	16#34(4)		16#33(3)	D25
D6	16#42(B)		16#41(A)	D26
D7	忽略		16#00	D27

执行后 →

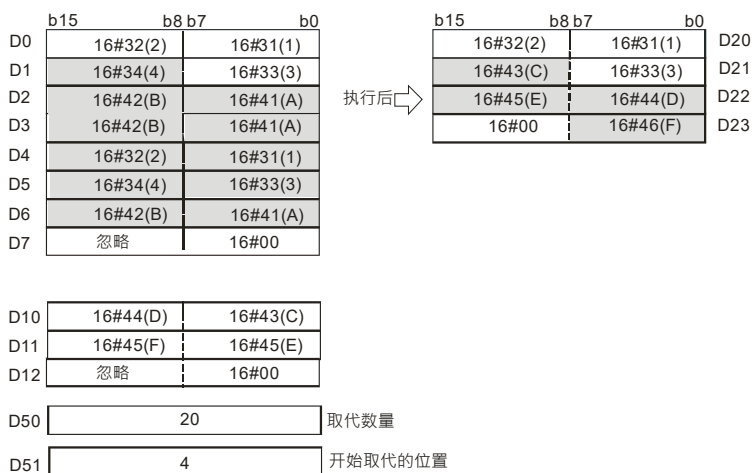
	b15	b8 b7	b0	
D10	16#44(D)		16#43(C)	
D11	16#45(F)		16#45(E)	
D12	忽略		16#00	

D50	3	取代数量
D51	4	开始取代的位置

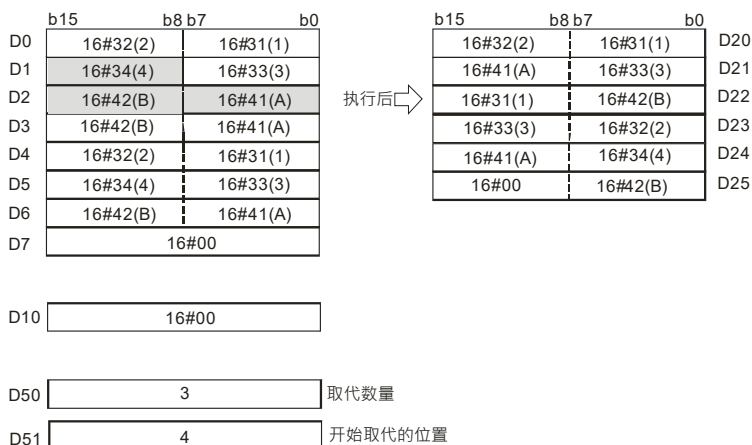
当 D50 指定值为 4，D51 指定值为 4 时，执行的结果如下图所示。



当 D50 指定值为 20，D51 指定值为 4 时，执行的结果如下图所示。



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。



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 $S_4 < 0$ 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。

6

API	指令码			操作数						功能					
2117	\$DEL	P		$S_1 \cdot S_2 \cdot S_3 \cdot D$						指定字符串删除					

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

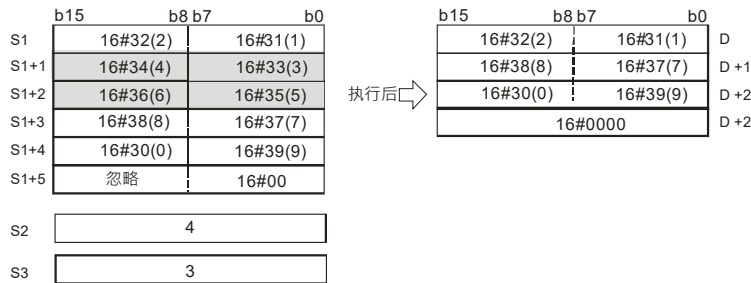
脉冲执行型	16 位指令 (9 steps)	32 位指令
AH500	AH500	-

符号：

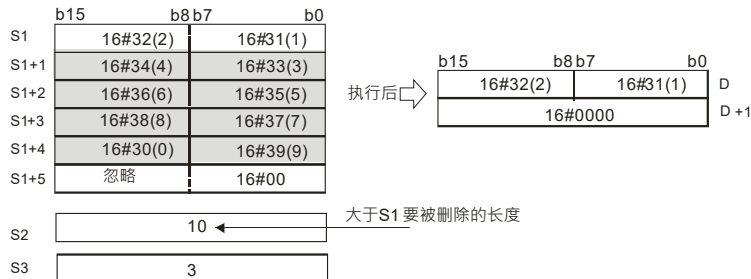
\$DEL		\$DELP		S_1 : 字符串来源	Word
En		En		S_2 : 要删除的个数	Word
S1	D	S1	D	S_3 : 从 S_1 的第 S_3 个字开始删除	Word
S2		S2		D : 删除后的字符串	Word
S3		S3			

指令说明：

- 从 S_1 字符串的第 S_3 个字开始，删除 S_2 个字符串，并将结果存于 D 中。
- 如下图说明，将 S_1 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



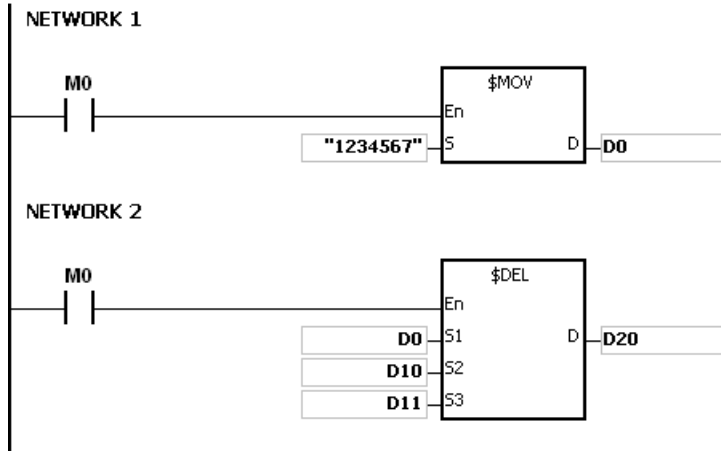
- 如果欲删除的字符串长度 $S_2 > S_1$ 要被删除的字符串长度，则只删除到 S_1 的最后一个字，并将结束字符 16#00 存入 D 中。



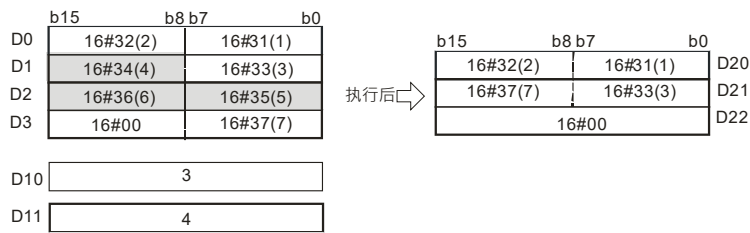
- $S_2=0$ ，指令不动作。

程序范例：

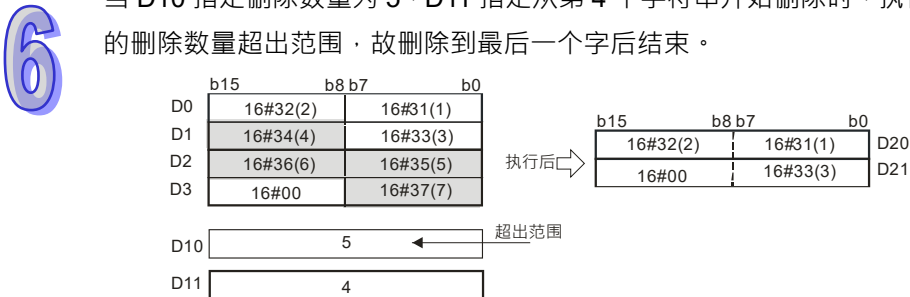
当 M0=ON 时，D0~D3 的内容为“1234567”；当 \$DEL 指令执行，将从 D11 内容的指定位置开始，删除 D0~D3 内的字符串，删除的字符数为 D10 内容所指定的数值，结果储存于 D20~D22 当中。



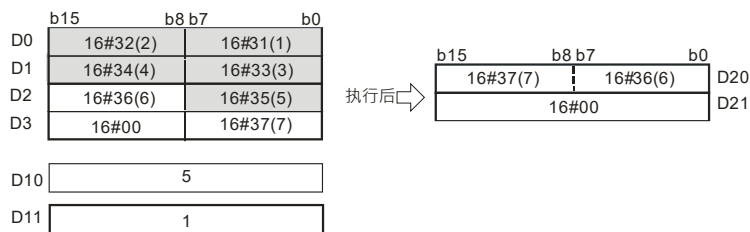
当 D10 指定删除数量为 3，D11 指定从第 4 个字符串开始删除时，执行的结果如下图所示。



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。

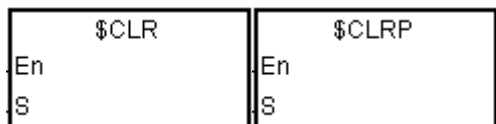


补充说明：

1. S₁ 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。
2. S₂<0 或 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。

API	指令码			操作数								功能					
2118	\$CLR	P		S								字符串清除					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●				●				
										脉冲执行型		16 位指令 (3 steps)			32 位指令		
										AH500		AH500			-		

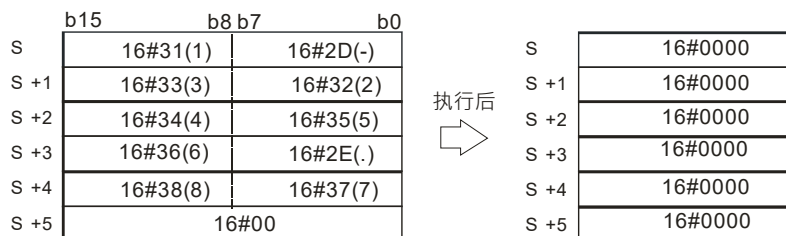
符号：



S：要清除的字符串来源 Word

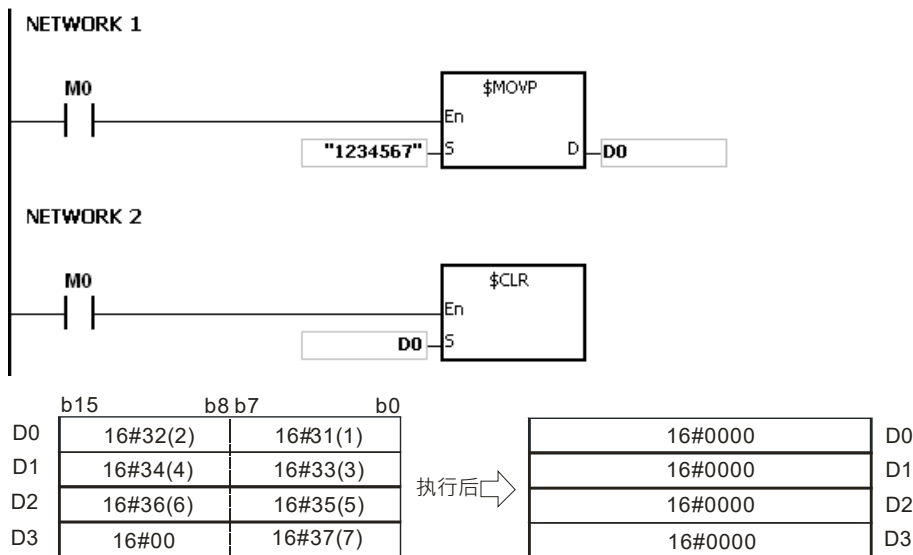
指令说明：

1. 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：

1. S 字符串中没有 16#00 当结尾，SM0=ON，记录错误码 SR0=16#200E。

API	指令码		操作数							功能							
2119		\$INS	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							字符串插入						

装置	X	Y	M	S	T	C	HC	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 (9-15 steps)	32 位指令
AH500	AH500	-

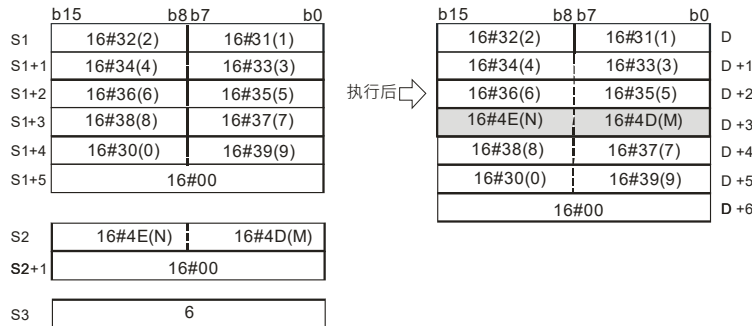
符号：

\$INS		\$INSP		S_1 : 来源字符串	Word
En		En		S_2 : 要插入的字符串	Word
S1	D	S1	D	S_3 : 从 S_1 的第 S_3 个字之后开始插入	Word
S2		S2		D : 存放插入之后的字符串	Word
S3		S3			

指令说明：

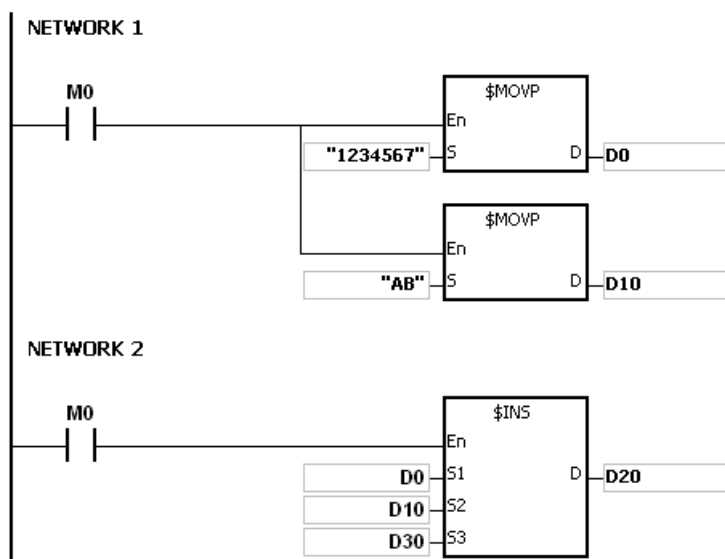
1. 从 S_1 字符串的第 S_3 个字之后开始，插入 S_2 的字符串，并将结果存于 D 中。
2. 如果 S_1 或 S_2 有任何一个字符串为空字符串，则另一个不是空字符串的值会被存于 D 中。
3. 如果 S_1 和 S_2 都为空字符串，则 D 为 16#0000。

6

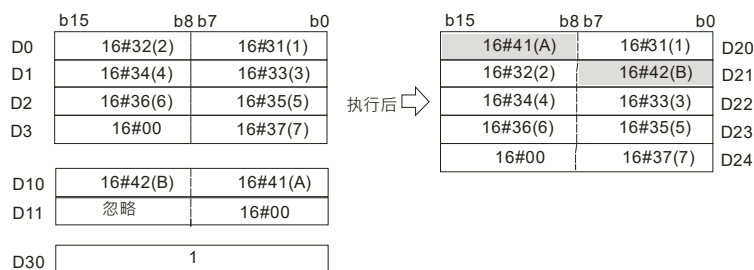


程序范例：

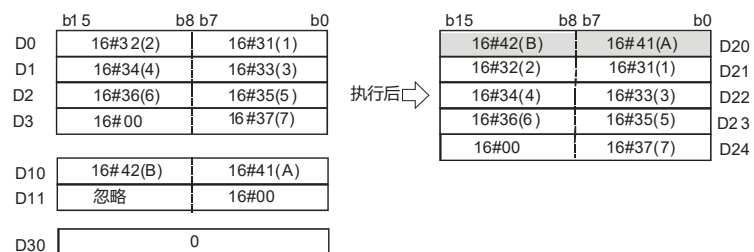
当 M0=ON 时，D0~D3 的内容为“1234567”，D10 的内容为“AB”；当 \$INS 指令执行，将从 D30 内容所指定的位置之后开始，插入 D10 的内容，结果储存于 D20~D24 当中。



D30 指定从第 1 个字符后开始插入，执行的结果如下图所示。



D30 指定从第 0 个字符后开始插入，执行的结果如下图所示。



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。

API	指令码			操作数							功能			
2120		FMOD	P	$S_1 \cdot S_2 \cdot D$							浮点数转 BCD 浮点数			

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

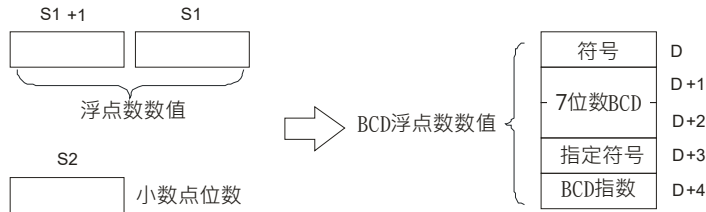
脉冲执行型	32 位指令 (7-8 steps)	64 位指令
AH500	AH500	-

符号：

FMOD		FMODP		S_1 : 来源浮点数值	Double Word
En		En		S_2 : 小数点位移位数	Word
S1	D	S1	D	D : 转换后的结果	Word
S2		S2			

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。

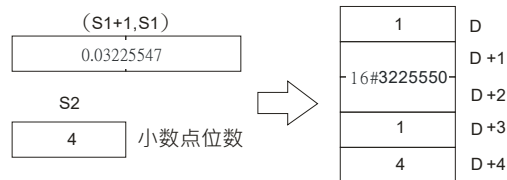
D：纪录符号，若 S_1 是正数，则 D=0；若 S_1 是负数，则 D=1。

(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

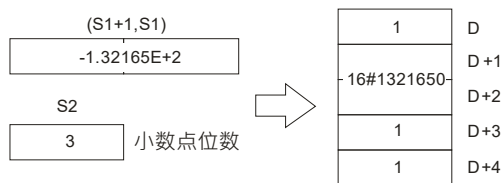
D=1 (因为 S_1 是负数)。

(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

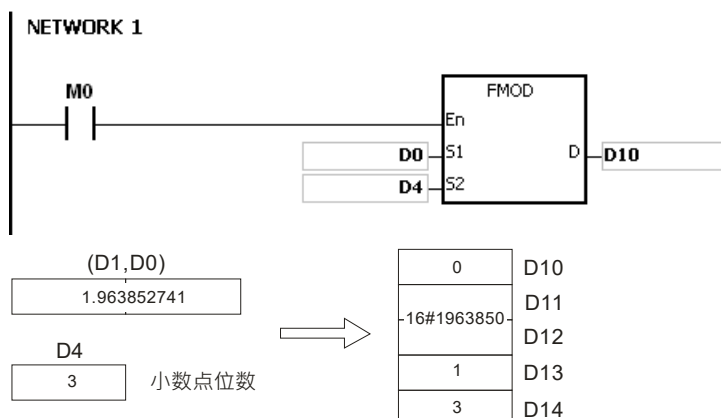
D = 1 (因为 S_1 是负数)。

(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 ·

API	指令码			操作数							功能						
2121		FREXP	P	$S_1 \cdot S_2 \cdot D$							BCD 浮点数转浮点数						

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

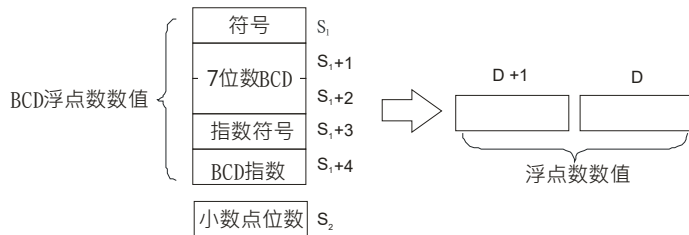
脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

符号：

FREXP		FREXPP		S_1 : BCD 浮点数来源值	Word
En		En		S_2 : 小数点位移位数	Word
S1	D	S1	D	D : 转换后的结果	Double Word
S2		S2			

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。



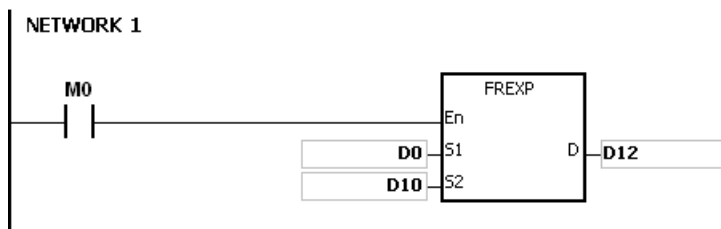
6

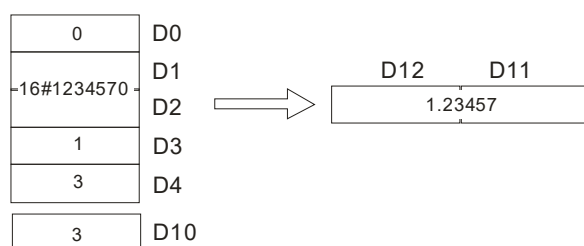
BCD 浮点数格式说明如下：

- 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 位数整数位。
- 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 个位数)

程序范例：

启动条件接点 M0，将 BCD 浮点数转成浮点数值，说明如下：





D0=0 表示浮点数值为正数.

(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。

6.22 以太网控制指令

6.22.1 以太网控制指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>2200</u>	SOPEN	–	✓	开启 Socket	7
<u>2201</u>	SSEND	–	✓	通过已开启的 Socket 传送数据	5
<u>2202</u>	SRCVD	–	✓	通过已开启的 Socket 接收数据	5
<u>2203</u>	SCLOSE	–	✓	关闭 Socket	5
<u>2204</u>	MSEND	–	✓	寄送电子邮件	9
<u>2205</u>	EMDRW	–	✓	Modbus TCP 数据读写	11
<u>2206</u>	–	DINTOA	✓	IP→字符串变换	5
<u>2207</u>	–	DIATON	✓	字符串→IP 变换	5-11

6.22.2 以太网控制指令说明

API	指令码			操作数								功能					
2200		SOPEN	P	$S_1 \cdot S_2 \cdot S_3$								开启 Socket					

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
AH500	AH500	-

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，对应以太网中 Socket 设定；欲开启 TCP 连线时， S_3 为 1 时，表示 AH500 为从端，主动送出连线要求给主端，为 S_3 为 0 时，表示 AH500 为主端，等待来自从端的连线要求，欲开启 UDP 连线时， 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 通讯接口连线设定欲使用的 Socket
- TCP Socket 设定对应缓存器，其中除了传送数据计数器及接收数据计数器是只读，其他缓存器的值皆可更改。

Socket 编号	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

Socket 编号 项目	1	2	3	4	5	6	7	8
	远程通信埠	SR1121	SR1134	SR1147	SR1160	SR1173	SR1186	SR1199
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. 连线保持时间若设为 0，则表示没有该连线会保持等待状态，不会逾时。
6. UDP Socket 设定对应缓存器，其中除了传送数据计数器及接收数据计数器是只读，其他缓存器的值皆可更改。

6

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
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	远程通信埠	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连线要求 2. 透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之特定 IP 地址的连线要求 2. 透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 连线要求传送数据时，系统会指定未使用的通讯端口为本机通讯端口。 2. 透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. 接受设定之本机通讯端口编号、远程通信端口和特定 IP 地址的连线要求 2. 透过远程通信端口传送数据至特定 IP 地址启动 3. 透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连线，开启中标志为 ON，若连线成功则已连线标志为 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	已连线标志	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，则连线已开启之标志为 ON，反之若有错误则对应错误标志 ON。

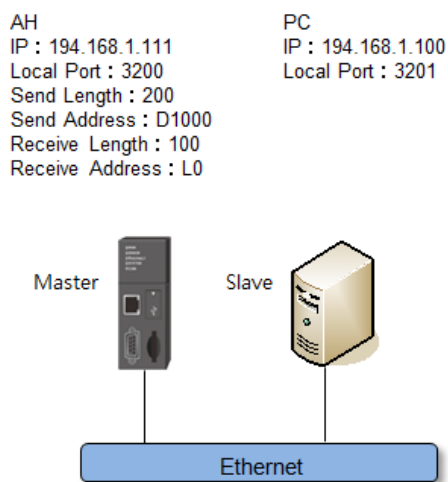
UDP Socket 编号	连线已开启标志	错误标志
1	SM1334	SM1338
2	SM1339	SM1343
3	SM1344	SM1348
4	SM1349	SM1353

UDP Socket 编号	连线已开启标志	错误标志
5	SM1354	SM1358
6	SM1359	SM1363
7	SM1364	SM1368
8	SM1369	SM1373

10. 本指令一般都是使用脉冲执行型指令 (SOPENP) 。

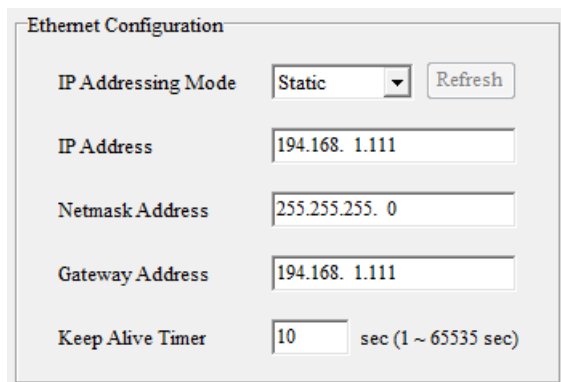
程序范例：

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 连线，此范例为使用第一组 TCP 连线。



2. ISPSOft -> HWCONFIG (以太网-基本设定)

6



3. ISPSOft -> HWCONFIG (以太网-进阶设定 -> 通讯接口 -> TCP)

TCP Socket Connection

Remote IP: 194.168. 1.100

Remote Port: 3201

Local Port: 3200

Send Data Length: 200 (0 ~ 500 words)

Send Data Address: D 1000

Receive Data Length: 100 (0 ~ 500 words)

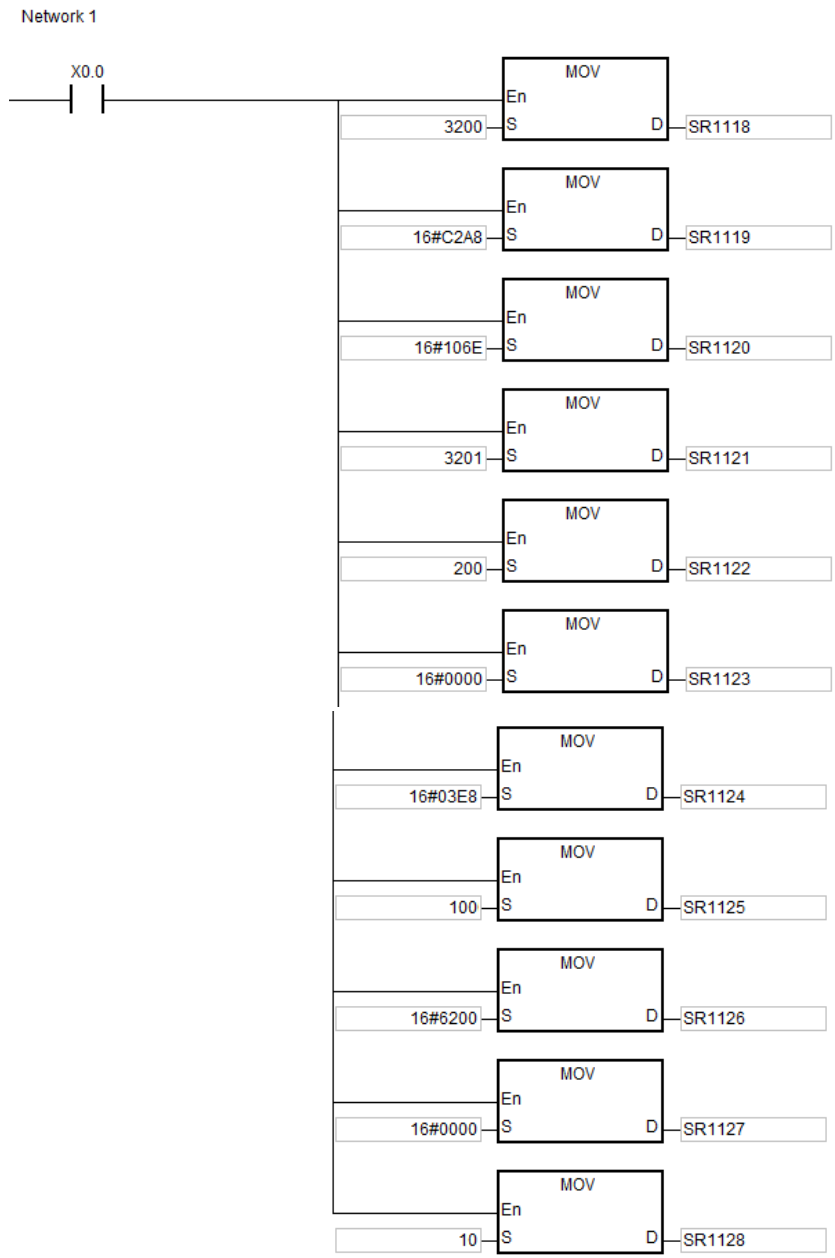
Receive Data Address: L 0

Keep Alive Timer: 10 sec (0 ~ 65535 sec)

OK

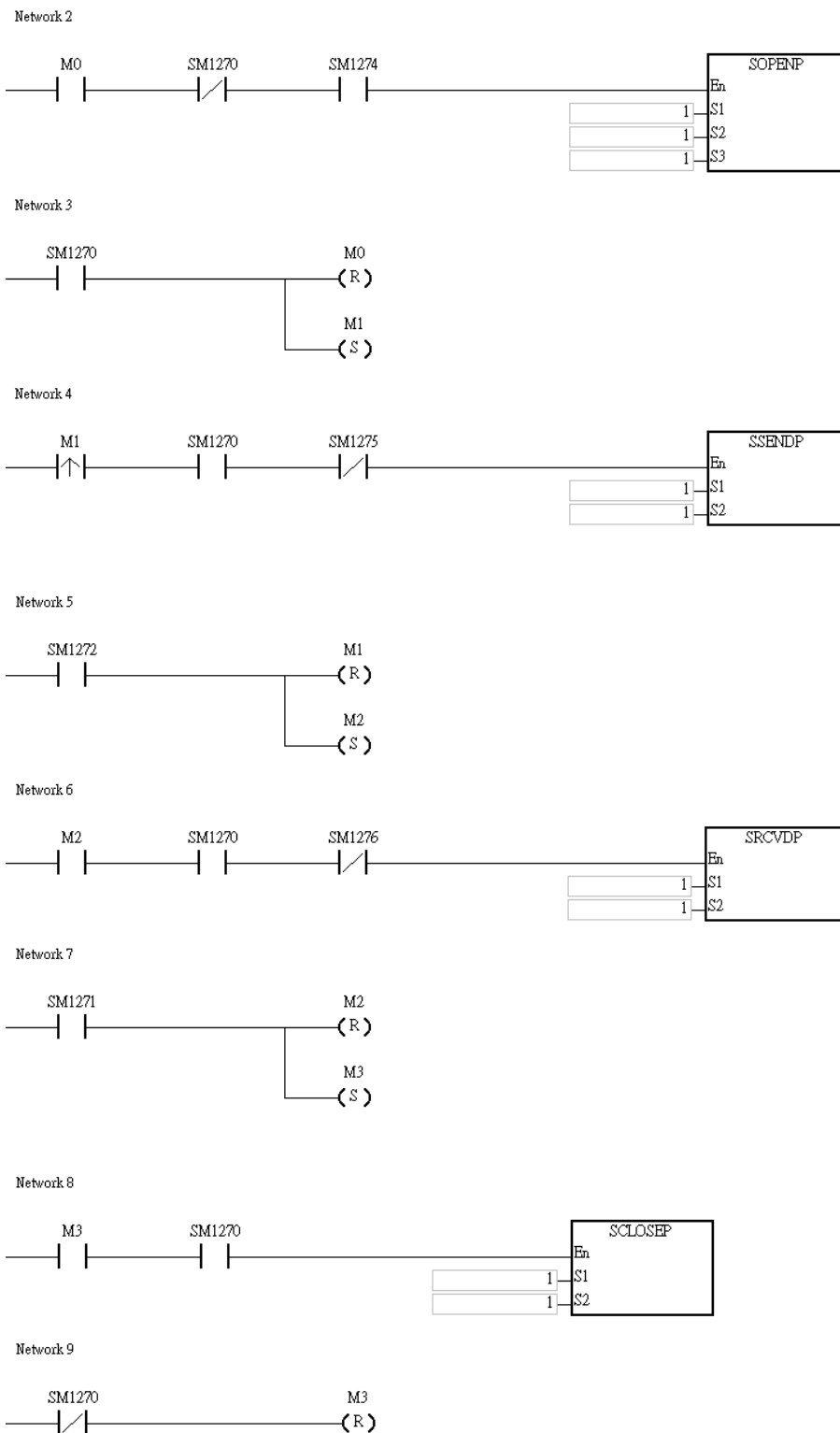
Cancel

4. 当 X0.0=ON 时，设定 TCP Socket1 的信息，亦可先行在 ISPSOFT PLC 参数设定中设定，即可省略 Network 1。
5. 当 M0=ON 时，检查欲连线的 Socket 是否已连线或是连线中的状况，如果不是则正式进入连线程序，连线完成后，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 是否为已连线，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，准备接收数据，检查欲接收的 Socket 是否为已连线，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，准备关闭连线，检查欲关闭的 Socket 是否为已连线状态，若已连线则开始关闭连线，若尚未连线则此指令无法运行，当连线关闭后，清除 M3。



用户也可以直接利用 ISPSOft 内的 HWCONFIG，直接对其设定，则可省略此设定的步骤，请参照本指令说明 3。

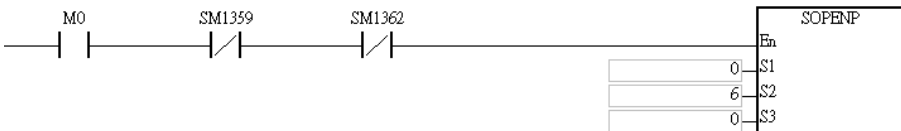
6



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 连线，此范例为使用 UDP 第六组连线。
10. 当 M0=ON 时，检查欲连线的 Socket 是否处于连线已开启或是正在接收中状态，如果不是则正式进入连线程序，连线完成后清除 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，准备接收数据，检查欲接收的 Socket 是否为已连线，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，开始关闭连线，检查欲关闭的 Socket 是否为已连线状态，若已连线则开始关闭连线，若尚未连线则此指令无法运行，连线关闭后清除 M3。

Network 1



Network 2



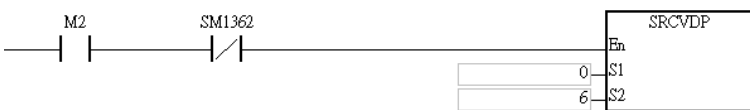
Network 3



Network 4



Network 5



Network 6



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 连线时，若主端连线数已达上限，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 连线时，若远程通信埠为 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 连线时，若主端连线数已达上限或是已有相同的连线，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 连线时，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 连线时，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 连线时，欲连线的 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 连线时，欲连线的 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 连线时，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 连线时，若此 **Socket** 为已连线或在连线中，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. 当连线过程中，对方主机放弃此连线，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，则从端会自行切断连线，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，并切断连线。

API	指令码		操作数				功能			
2201	SSEND	P	$S_1 \cdot S_2$				通过已开启的 Socket 传送数据			

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

符号：

SSEND	SSENDP	S_1 : Socket 模式. Word
En	En	
S1	S1	
S2	S2	

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 编号对应以太网络中 Socket 设定。
- $S_1=0\sim1$ ， $S_2=1\sim8$ 。
- 使用本指令前，需先以 SOPEN 开启 Socket 连线，当已连线标志 (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。

6

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

- 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

- 本指令一般都是使用脉冲执行型指令 (SSENDP)。

7. 使用此指令时，若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: 194.168.1.100

Remote Port: 3201

Local Port: 3200

Send Data Length: 4 (0 ~ 500 words)

Send Data Address: D 0

Receive Data Length: 4 (0 ~ 500 words)

Receive Data Address: L 0

Keep Alive Timer: 10 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讯端口为 **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，对方主机放弃此连线，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**。

6

API	指令码			操作数								功能					
2202		SRCVD	P	$S_1 \cdot S_2$								通过已开启的 Socket 接收数据					

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
AH500	AH500	-

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 连线，当已连线标志 (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

- 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

7. 本指令一般都是使用脉冲执行型指令 (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. 若实际接收报文的长度，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，对方主机放弃此连线，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。



API	指令码			操作数								功能					
2203	SCLOSE	P		$S_1 \cdot S_2$								关闭 Socket					

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

脉冲执行型	16 位指令 (5 Steps)	32 位指令
AH500	AH500	-

符号：

SCLOSE		SCLOSEP		S_1 : Socket 模式	Word
En		En			
S1		S1		S_2 : Socket 编号	Word
S2		S2			

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 为连线状态，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（连线中标志为 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讯端口的连线，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，则已连线标志为 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

Socket 编号	错误标志
6	SM1363
7	SM1368
8	SM1373

8. 本指令一般都是使用脉冲执行型指令 (SCLOSEP) 。

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. 当关闭过程中，对方主机放弃此连线，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。

API	指令码		操作数							功能						
2204	MSEND	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							寄送电子邮件						

装置	X	Y	M	S	T	C	HC	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 (9 steps)	32 位指令
AH500	AH500	-

符号：

MSEND		MSENDP			
En		En		S_1 ：	邮件组态 Word
S1	D	S1	D	S_2 ：	邮件主旨 Word
S2		S2		S_3 ：	邮件信息 Word
S3		S3		D：	指令完成标志 Bit

指令说明：

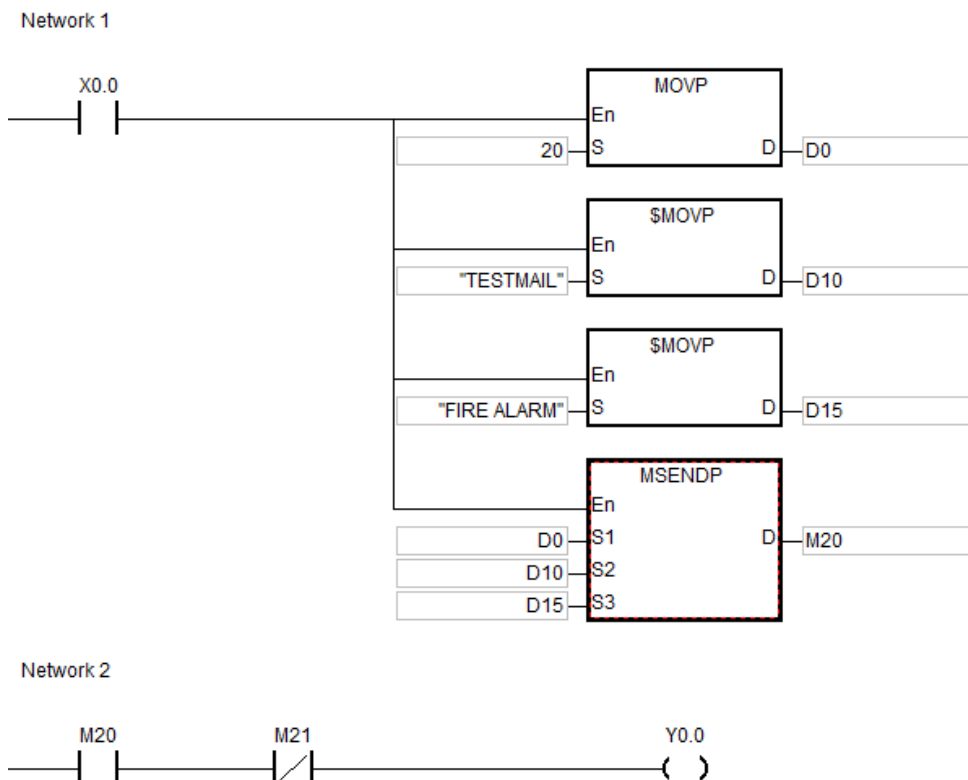
- 利用 S_1 、 S_2 和 S_3 的设定转为所需的数据，寄出 Email。
- 使用本指令前，需用 ISPSOFT 完成以下设定。
 - 在 PLC 参数设定 -> 基本设定中设定本机的 IP 地址、网络屏蔽。
 - 在 PLC 参数设定 -> 进阶设定 -> 邮件下设定邮件服务器的 IP 地址、通讯端口、本机邮件地址和主旨。
 - 在 PLC 参数设定 -> 进阶设定 -> 邮件地址与发送条件设定 -> 邮件地址设定要欲寄送的邮件位置。
 - 若外寄服务器需要身分验证，可在 PLC 参数设定 -> 进阶设定 -> 邮件下设定用户名称和密码。
- 邮件设定如下：

操作数	描述	设定范围
S1	远程邮件编号	1-256。 bit0~bit7 分别代表在 ISPSOFT 中设定的远程邮件地址 (ISPSOFT 中邮件编号为 8 组)。bit0 代表远程地址 1，以此类推。 若欲传送至此邮件地址，须建立对应的位。
S2	主旨	邮件主旨最大支持 16 字符。
S3	信息	邮件信息最大支持 64 字符。
D	指令完成标志	指令完成后，该位为 ON。若指令完成异常，下一个位为 ON。

4. 本指令一般都是使用脉冲执行型指令 (MSENDP) 。

程序范例：

当 X0.0=ON 时，寄一封邮件给远程邮件编号 3 和编号 5。(D0=00010100)，当与 SMTP 服务器通讯完毕后 M20= ON，若通讯过程中无错误 M21=OFF，输出 Y0.0=ON。



6

补充说明：

1. D+1 装置超出范围时，指令不执行，SM0=ON，SR0 错误码为 16#2003。
2. S₁<1 或 S₁>256，指令不执行，SM0=ON，SR0 错误码为 16#2003。
3. S₂ 或 S₃ 的字符串长度，系统以抓取终止符 (16#00) 为主，若达长度上限仍未读取终止符，则字符串长度等于上限值。
4. 当执行此指令时，若系统对 SMTP 服务器的连线已达上限，则会发生错误，错误码 16#6100。
5. 若 PLC 参数中设定寄送邮件需经过身分验证，又输入的账号经 SMTP 服务器验证后无效，则会发生错误，错误码 16#6108；如账号有效密码错误时，错误码 16#6109。
6. 如远程邮件地址被 SMTP 服务器判别为无效，则会发生错误，错误码 16#6111。
7. 如寄送过程中，SMTP 服务器回应逾时，则会发生错误，错误码 16#6107，寄送动作取消。
8. D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [2] of WORD/INT。
9. S₂ 与 S₃ 之间请保留一个字符，供本指令产生中断字符使用。
10. 当执行此指令时，若网络线未连接，则 SM1100 为 ON，错误码 16#600D。

API	指令码		操作数								功能				
2205	EMDRW	P	$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot n$								Modbus TCP 数据读写				

装置	X	Y	M	S	T	C	HC	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 (11 steps)	32 位指令
AH500	AH500	-

符号：

EMDRW		EMDRWP		说明	数据类型
En		En		S_1 : 连线装置地址	Word
S1		S1		S_2 : 通讯功能码	Word
S2		S2		S_3 : 欲读写数据的地址	Word
S3		S3		S : 欲读写之数据存放寄存器	Bit/ Word
S		S		n : 读写数据长度	Word
n		n			

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 $S_1+1=16\#AC10$ $S_1+2=16\#90E6$
S_1+2	远程 IP Low Word	
S_1+3	指令完成后是否关闭连线	0 : 关闭连线 · 1 : 保持连线(依 PLC 参数设定中 TCP 连线保持时间决定连线何时关闭)

S_2 : 通讯功能码 (Function Code)。

例如：

- AH500 的读取多笔位装置 (Bit) (非 Discrete Input 装置) 命令为 1 (16#01)。
- AH500 的读取多笔位装置 (Bit) (仅 Discrete Input 装置) 命令为 2 (16#02)。
- AH500 的读取多笔字符装置 (Word) (非 Input Register 装置) 命令为 3 (16#03)。
- AH500 的读取多笔字符装置 (Word) (仅 Input Register 装置) 命令为 4 (16#04)。
- AH500 的单笔位装置 (Bit) 状态写入命令为 5 (16#05)。
- AH500 的单笔字符装置 (Word) 数据写入命令为 6 (16#06)。
- AH500 的多笔位装置 (Bit) 状态写入命令为 15 (16#0F)。
- AH500 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。

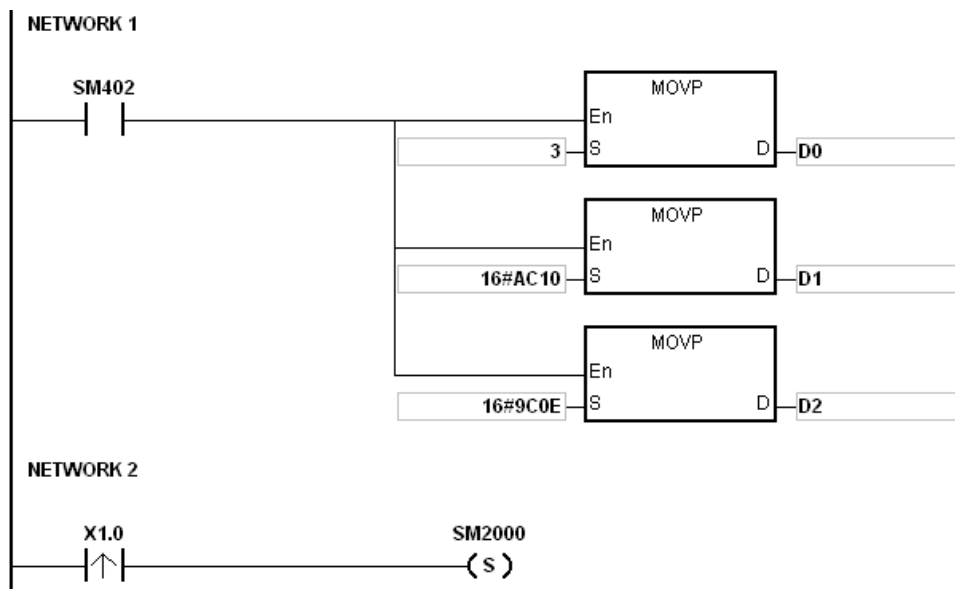
3. **S₃**：欲读写数据的地址（装置地址）。连线装置的内部装置地址。
4. **S**：欲读写之数据（来源或目的）。由用户设定寄存器，将欲写入数据长度的数据事先存入寄存器内。或数据读取后存放之寄存器。
5. **n**：读写数据长度（数据长度），指定范围 1~240（Byte）。设定的数据量不可以大于 240 字符。Coil 相关的通讯命令，其数据单位为 Bit，n 的范围为 1~1920。Register 相关的通讯命令，其数据单位为 Word，则 n 的范围为 1~120。

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	连线已关闭标志
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

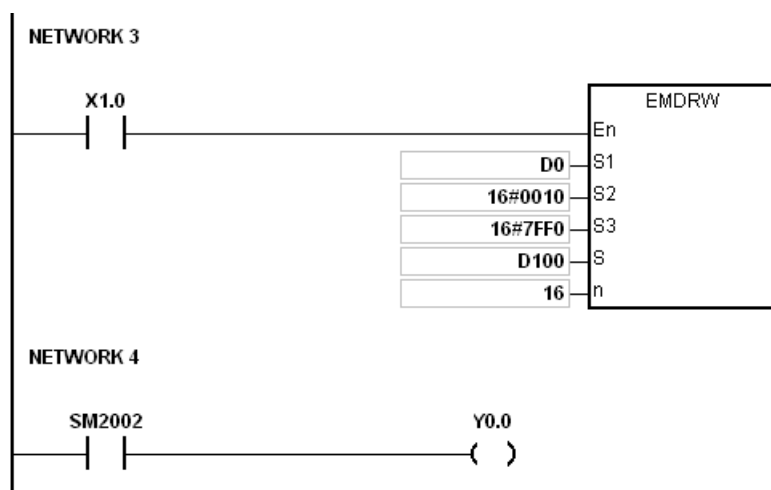
6. 本指令于程序中使用次数并无限制，但是同时间仅有 8 个指令被执行。
7. 若指令执行时，同时有不只一个送信标志为 ON，则以标志数字最小者优先执行。
8. 本指令一般都是使用脉冲执行型指令（EMDRWP）。

程序范例：

1. 设定远程站号为 3。



6



补充说明：

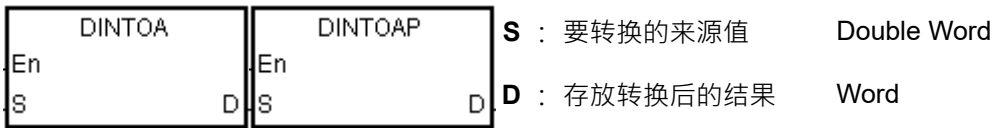
1. S_1 、 S_2 、 S 内容值超出范围时，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
2. 若 n 操作数的内容值不在范围内，指令不执行， $SM0=ON$ ， $SR0$ 记录错误码 16#200B。
3. S_1+3 装置超出范围时，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
4. S 装置不足以写入或读取 n 所指定的长度时，指令不执行， $SM0=ON$ ， $SR0$ 错误码为 16#2003。
5. 当 S_2 通讯功能码指定为 Bit 读取时 S 必需指定为 Bit 装置，否则视为运算错误，指令不执行， $SM0=ON$ ， $SR0$ 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，但其对应的连线已关闭标志不为 ON，则系统会自动寻找是否有其它要求标志为 ON 且其对应的连线已关闭标志为 ON 来执行指令，若无则此次指令不执行。
9. 连线时，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. 连线时，当与远程机器中断连线时，错误码为 16#6401；或者连线超时，错误码为 16#6402。
11. 传送指令时，当远程机器中断连线时，错误码为 16#6401；或者 Ack 超时，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 或者回复超时，错误码为 16#6402。
13. 关闭连线时，当与远程机器中断连线时，错误码为 16#6401；或者连线超时，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。

API	指令码			操作数								功能					
2206	D	INTOA	P	S · D								IP→字符串变换					

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

脉冲执行型	16 位指令	32 位指令 (5 steps)
AH500	-	AH500

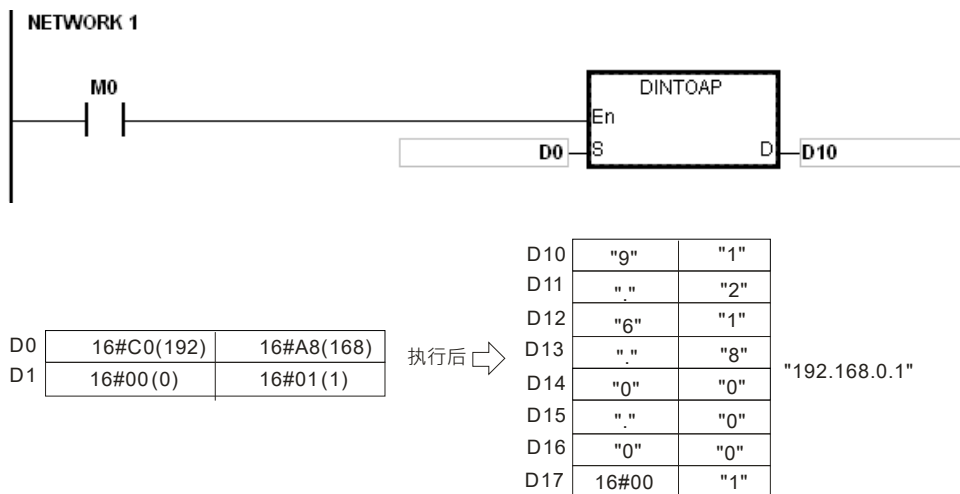
符号：



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：



程序范例二：



C0A80001(Hex) 执行后 \Rightarrow

D10	"9"	"1"
D11	"."	"2"
D12	"6"	"1"
D13	"."	"8"
D14	"0"	"0"
D15	"."	"0"
D16	"0"	"0"
D17	16#00	"1"

"192.168.0.1"

补充说明：

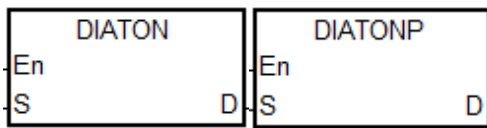
D 操作数，若使用 ISPSOft 宣告，则数据类型为 ARRAY [8] of WORD/INT。

API	指令码			操作数								功能				
2207	D	IATON	P	S · D								字符串→IP 变换				

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

脉冲执行型	16 位指令	32 位指令 (5-11 steps)
AH500	-	AH500

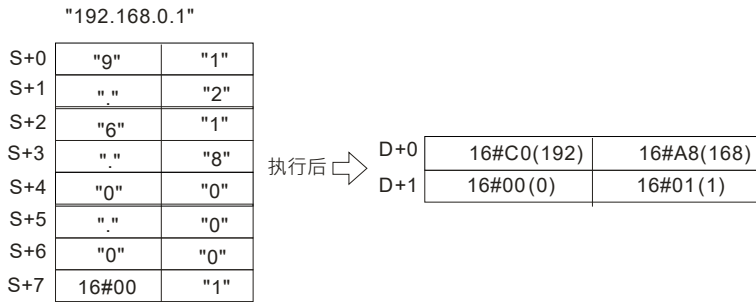
符号：



S : 要转换的来源值 String
D : 存放转换后的结果 Double Word

指令说明：

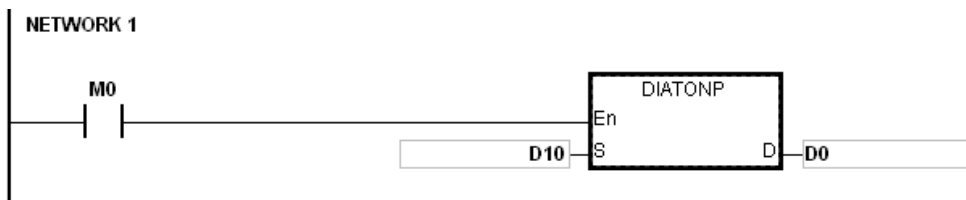
1. 将 **S** 的字符串 IP 地址转换成整数类型，并且将结果存到 **D**。
2. **S** 连续占用 8 个装置。

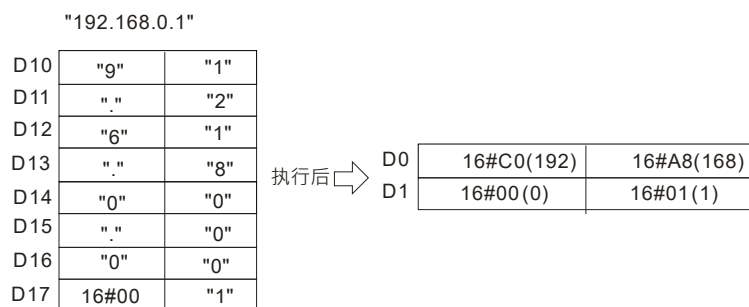


6

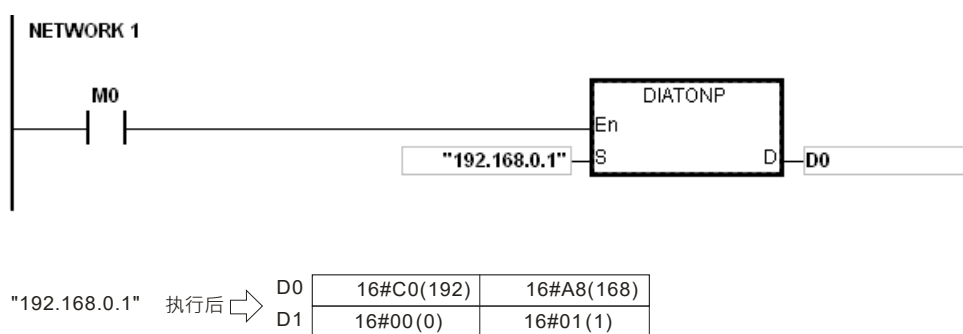
3. **S** 的字符串 IP 地址来源，每个区段数值固定为 3 个字符，中间以一个“.”16#2E 隔开。
4. **S** 的字符串 IP 地址来源，每个区段数值转换出来的结果不能>255。
5. 如果 **S** 是由用户直接输入字符串，则每个区段不必固定输入 3 个字符，例如：“192.168.0.1”，只要直接输入即可，不必输入“192.168.000.001”。

程序范例一：





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 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。

6.23 储存卡读写指令

6.23.1 储存卡读写指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>2300</u>	MWRIT	—	✓	将 PLC 数据写入到储存卡里	13
<u>2301</u>	MREAD	—	✓	将数据从储存卡里读取到 PLC	13
<u>2302</u>	MTWRIT	—	✓	将文字写入到储存卡里	11

6.23.2 储存卡读写指令说明

API	指令码			操作数								功能											
2300		MWRIT	P	C	S	S ₁	S ₂	S ₃	S ₄									将 PLC 数据写入到储存卡里					

装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
C								●	●				●	○	○		
S								●	●				●				
S ₁								●	●				●	○	○		
S ₂								●	●				●	○	○		
S ₃								●	●				●				
S ₄								●	●				●	○	○		

脉冲执行型	16 位指令 (13 steps)	32 位指令
AH500	AH500	-

符号：

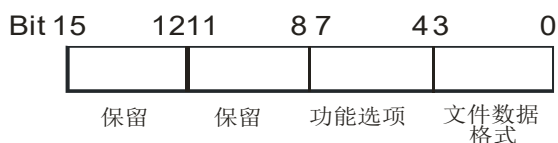
MWRIT	MWRITP
En	En
C	C
S	S
S ₁	S ₁
S ₂	S ₂
S ₃	S ₃
S ₄	S ₄

- C** : 控制参数 Word
S : 数据来源 Word
S₁ : 欲写入的数据长度 Double Word
S₂ : 换行 Word
S₃ : 文件名称 Word
S₄ : 文件内的数据地址 Double Word

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。
		副文件名称为.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。
		副文件名称为.csv。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。

6

说明项目	代码	说明
文件数据格式	2	每个数据之间用逗号做为区隔。
		每个数据的单位为 DWord。
		副文件名称为.csv。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	3	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 Word。
		副文件名称为.txt。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	4	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 DWord。
		副文件名称为.txt。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	5	数据与数据之间没有区隔符号。
		每个数据的单位为 Word。
		副文件名称为.txt。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
6	数据与数据之间没有区隔符号。	
	每个数据的单位为 DWord。	
	副文件名称为.txt。	
	采用 ASCII 的字符。	
	储存的数据采用 16 进制来表示。	
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。

- 数据来源 **S**：数据来源的起始地址。
- 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** : 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)

- 文件名称 **S₃~S₃+4** : **S₃** 占用连续 5 个装置来指定写入数据的文件名称，且文件名称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结束字符会有错误。当有读取到终止位时，将会停止读取字符，且检查文件名称是否合法。合法命名的字符为 A~Z、a~z 与 0~9。副文件名称会依据 **C** 的文件数据格式，而自动给予副文件名称，生成的文件会在默认文件夹里。若文件名称为“Test1”，则写入装置样式如下：

S3	'e'	'T'	ASCII代码 ➔	S3	16#65	16#54
S3+1	't'	's'		S3+1	16#74	16#73
S3+2	NUL	'1'		S3+2	16#00	16#31

- 默认文件夹路径：

机种名称	文件夹路径
AHCPU530-RS2	PLC CARD\AH500\UserProg
AHCPU530-EN	
AHCPU530-EN/RM	
AHCPU533-EN	

- 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)
S₄ 的参数单位	此参数单位为 Dword。
S₄ 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 S4 指定在文件里的数据地址。
	此 S4 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 S4 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 module 上，则此标志会为 ON 的状态。
SM451	储存卡在写保护的 mode 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)

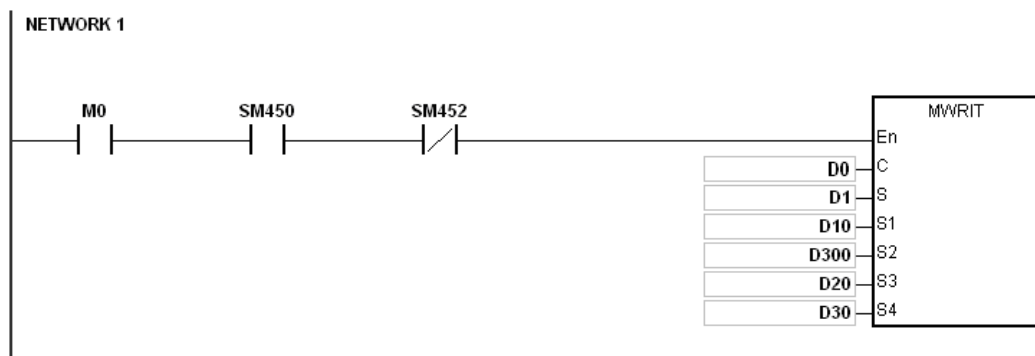
错误代码	说明
16#005E	当 CPU module 在进行储存卡初始化流程时，发生错误而无法进行初始化流程。
16#005F	路径错误或文件不存在。
16#0060	无法产生默认文件夹。
16#0061	储存卡内存空间不足。
16#0062	储存卡为写保护保护。
16#0063	当数据写入到文件时，发生错误。
16#0064	当要读取文件数据时，无法读取文件的数据。
16#0065	文件属性为只读状态。

3. 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。

6

程序范例：

当储存卡有插入 CPU module 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。副文件名称为.csv。采用 ASCII 的字符。

操作数	设定值	说明
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	写入文件名称为“Test1”。
D30、D31	16#00000000	从文件最初的地址开始写入。

补充说明：

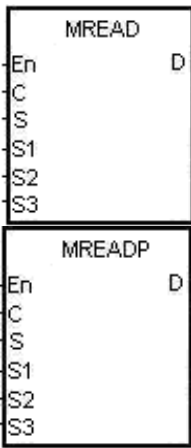
1. **C** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. **S₁** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. **S₂** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
4. **S₃** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

API	指令码		操作数							功能							
2301		MREAD	P	C · S · S₁ · S₂ · S₃ · D							将数据从储存卡里读取到 PLC						

装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●	○	○		
D								●	●				●				

脉冲执行型	16 位指令 (13 steps)	32 位指令
AH500	AH500	-

符号：

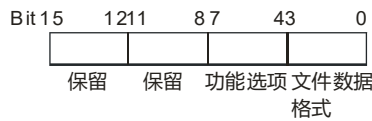


- C** : 控制参数 Word
- S** : 文件名称 Word
- S₁** : 文件的数据地址 Double Word
- S₂** : 系统内部使用 Word
- S₃** : 欲读取的数据长度 Double Word
- D** : 数据目的地 Word

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)。
		副文件名称为 .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。
		副文件名称为 .csv。
		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
文件数据格式	2	每个数据之间用逗号做为区隔。
		每个数据的单位为 DWord。
		副文件名称为.csv。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	3	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 Word。
		副文件名称为.txt。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	4	每个数据之间用 TAB 做为区隔。
		每个数据的单位为 DWord。
		副文件名称为.txt。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
	5	数据与数据之间没有区隔符号。
		每个数据的单位为 Word。
		副文件名称为.txt。
		采用 ASCII 的字符。
		储存的数据采用 16 进制来表示。
6	数据与数据之间没有区隔符号。	
	每个数据的单位为 DWord。	
	副文件名称为.txt。	
	采用 ASCII 的字符。	
	储存的数据采用 16 进制来表示。	
功能项目	0	读取指定文件内的数据 (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

- 文件名称 **S~S+4** : **S** 占用连续 5 个装置来指定写入数据的文件名称，且文件名称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结束字符会有错误。当有读取到终止位时，将会停止读取字符，且检查文件名称是否合法。合法命名的字符为 **A~Z**、**a~z** 与 **0~9**。副文件名称会依据 **C** 的文件数据格式，而自动给予副文件名称，生成的文件会在默认文件夹里。若文件名称为“Test1”，则写入装置样式如下：



- 默认文件夹路径：

机种名称	文件夹路径
AHCPU530-RS2	PLC CARD\AH500\UserProg
AHCPU530-EN	
AHCPU530-EN/RM	
AHCPU533-EN	

- 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。

6

2. 相关标志

标志	说明
SM450	若储存卡有存在于 CPU module 上，则此标志会为 ON 的状态。
SM451	储存卡在写保护的 mode 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)

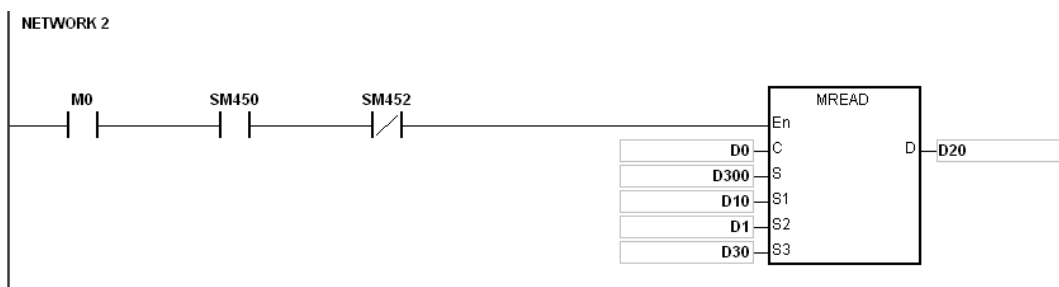
错误代码	说明
16#005E	当 CPU module 再进行储存卡初始化流程时，发生错误而无法进行初始化流程。
16#005F	路径错误或文件不存在。
16#0060	无法产生默认文件夹。
16#0061	储存卡内存空间不足。
16#0062	储存卡为写保护保护。
16#0063	当数据写入到文件时，发生错误。

错误代码	说明
16#0064	当要读取文件数据时，无法读取文件的数据。

3. 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 module 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。副文件名称为.csv。采用 ASCII 的字符。
D300	D300=16#6554 D301=16#7473 D302=16#0031	读取文件名称为“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：

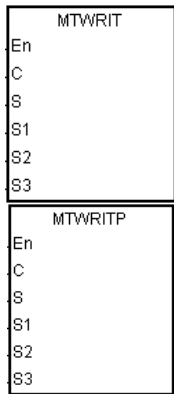
1. **C** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
2. **S2** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
3. **S3** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。
4. **D** 内容值超出范围时，视为运算错误，指令不执行，SM0=ON，SR0 记录错误码 16#2003。

API	指令码		操作数							功能							
2302		MTWRIT	P	C · S · S₁ · S₂ · S₃							将文字写入到储存卡里						

装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●				

脉冲执行型	16 位指令 (11 steps)	32 位指令
AH500	AH500	-

符号：



- C** : 控制参数 Word
- S** : 数据来源 Word
- S₁** : 数据长度 Word
- S₂** : 分隔符 Word
- S₃** : 文件名称 Word

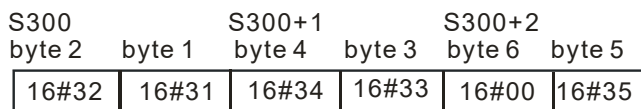
指令说明：

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。

- 数据来源 **S**：指定要写入储存卡的装置数据来源。若欲写入“12345”的字符串到文字文件里的话，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。



- 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 S_2 ：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： $S_1=N$ ， S_2 分隔符之情况如下方表格，则 S_2 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 操作数的高 Byte。
非 16#00	非 16#00	写入 N+2 个 Bytes 到文件里，其中 N+1 为 S_2 操作数的高 Byte，N+2 为 S_2 操作数的低 Byte。

- 文件名称 $S_3 \sim S_3+4$ ： S_3 占用连续 5 个装置来指定写入数据的文件名称，且文件名称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结束字符会有错误。当有读取到终止位时，将会停止读取字符，且检查文件名称是否合法。合法命名的字符为 A~Z、a~z 与 0~9。副文件名称会依据 C 的文件数据格式，而自动给予副文件名称，生成的文件会在默认文件夹里。若文件名称为“Test1”，则写入装置样式如下：

S_3	'e'	'T'	ASCII 代码	S_3	16#65	16#54
S_3+1	't'	's'		S_3+1	16#74	16#73
S_3+2	NUL	'1'		S_3+2	16#00	16#31

- 默认文件夹路径：

机种名称	文件夹路径
AHCPU530-RS2	PLC CARD\AH500\UserProg
AHCPU530-EN	
AHCPU530-EN/RM	
AHCPU533-EN	

2. 相关标志

标志	说明
SM450	若储存卡有存在于 CPU module 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。

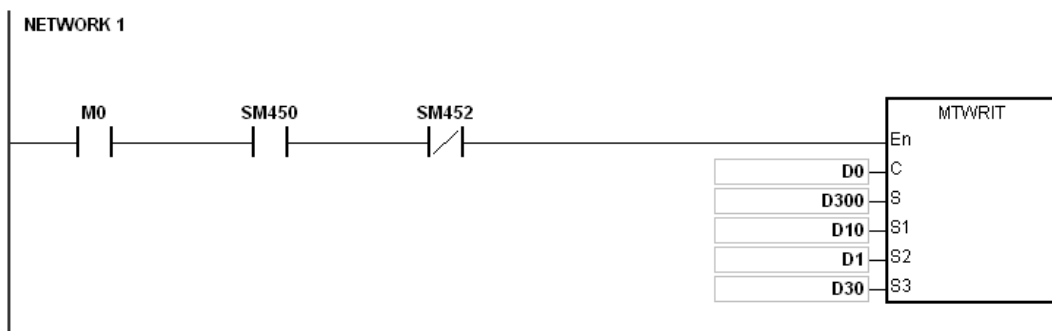
相关错误代码 (SR453)

错误代码	说明
16#005E	当 CPU module 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	无法产生默认文件夹

错误代码	说明
16#0061	储存卡内存空间不足
16#0062	储存卡为写保护保护
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 module 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。副文件名称为.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	写入之文件名称为 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。

6

6.24 任务控制指令

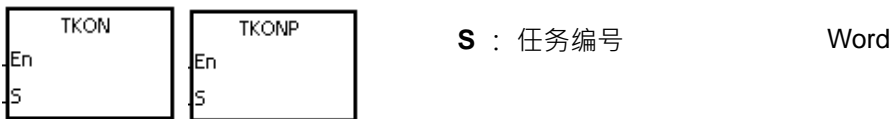
6.24.1 任务控制指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>2400</u>	TKON	–	✓	Cyclic 任务 Task 启动	3
<u>2401</u>	TKOFF	–	✓	Cyclic 任务 Task 关闭	3

6.24.2 任务控制指令说明

API	指令码			操作数									功能				
2400	TKON	P		S									Cyclic 任务 Task 启动				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	"\$"	DF	
S	●	●						●	●		●		○				
脉冲执行型			16 位指令 (3 steps)			32 位指令											
AH500			AH500			-											

符号：



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 手册。

6

程序范例：

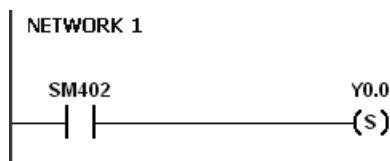
程序写入主机时，只指定启动 Task Cyclic_0 任务，PLC RUN 后，因为程序中指示 TKON 1 表示执行 Task Cyclic_1，所以 Y0.0=ON。

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 使用说明。

API	指令码			操作数								功能				
2401		TKOFF	P	S								Cyclic 任务 Task 关闭				
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	"\$"	DF
S	●	●						●	●		●		○			
							脉冲执行型			16 位指令 (3 steps)			32 位指令			
							AH500			AH500			-			

符号：



S : 任务编号

Word

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 手册。

程序范例：

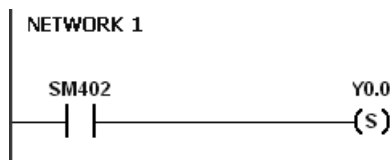
程序写入主机时，就指定启动 Task Cyclic_0 及 Tasks Cyclic_1 任务，PLC RUN 后，因为 TKOFF1 表示关闭 Task Cyclic_1 因此 Y0.0=OFF。

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 使用说明。

6.25 SFC控制指令

6.25.1 SFC控制指令一览表

API	指令碼 (位)		P 指令	功能	STEPS
	16	32			
<u>2500</u>	SFCRUN	-	-	SFC 启动	7
<u>2501</u>	SFCPSE	-	-	SFC 暂停	5
<u>2502</u>	SFCSTP	-	-	SFC 停止	3

6.25.2 SFC控制指令说明

API	指令码			操作数							功能			
2500	SFCRUN			$S_1 \cdot S_2 \cdot S_3$							SFC 启动			

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

脉冲执行型	16 位指令 (7 steps)	32 位指令
-	AH500	-

符号：

	SFCRUN
En	
S1	
S2	
S3	

S_1 : SFC 名称

S_2 : 控制参数 Word

S_3 : 要启动的 STEP 名称

指令说明：

- 依照 S_2 的设定，启动 S_1 所指定的SFC程序。
- 指令执行时， S_1 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。
- 操作数说明：
 - S_1 为SFC程序的名称。
 - $S_2=0$ ，启动 S_1 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。
 - $S_2=1$ ，启动 S_1 指定的SFC程序，SFC中的SFC/STEP/ACTION/TRANSITION状态跟参数会被清除，并从 S_3 指定的STEP开始执行。
 - $S_2=2$ ，启动 S_1 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。
 - S_3 为 S_1 指定的SFC程序中的STEP名称。
- S_2 的范围 0~2，超出范围时，视为 0。
- 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。

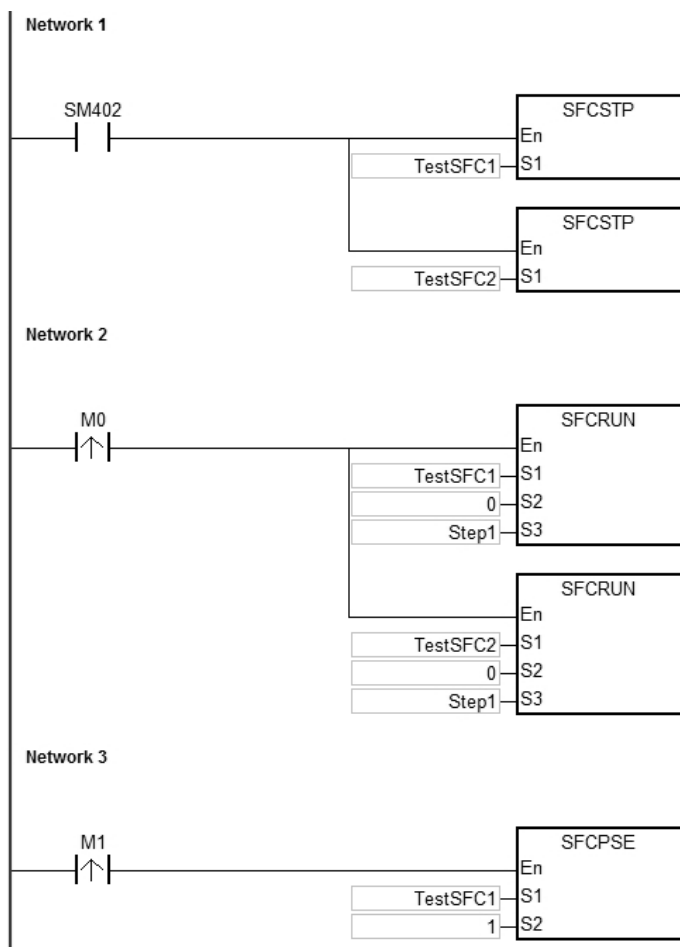
- 系统执行 (RUN) 时，TestSFC1 与 TestSFC2 立刻执行 SFCSTP 动作，两个 SFC POU 会停止执行。

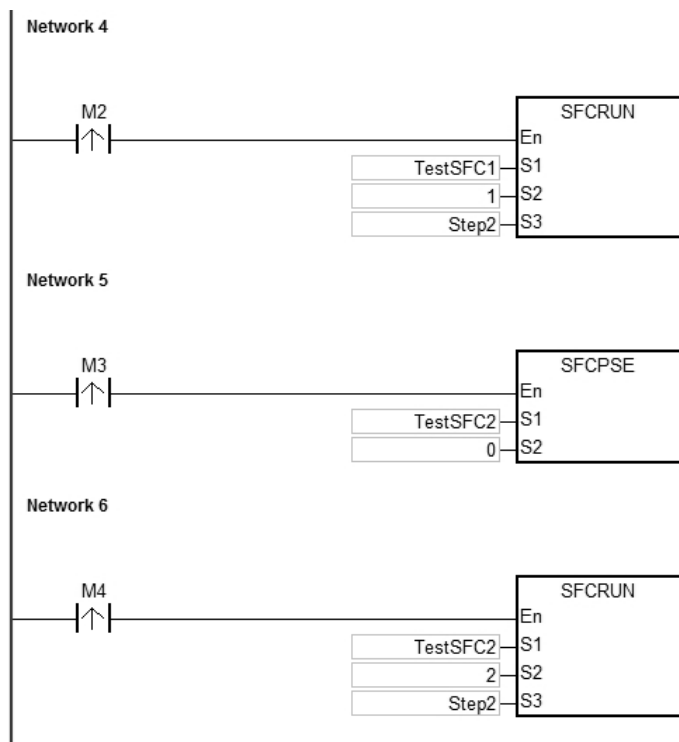
2. 当M0由OFF→ON时，TestSFC1/ TestSFC2 POU执行动作 (SFCRUN)^{注一}，执行动作请参考 TestSFC1/ TestSFC2 POU程序内容。设定S₂控制参数 0，会清除SFC中之状态及参数，并且从 Step1 开始执行。设定S₂控制参数 1，会清除SFC中之状态及参数，并且从S₃指定的Step开始执行。
3. 当 M1 由 OFF→ON 时，TestSFC1 POU 会暂停，控制参数设为 1，SFC 中正在执行中的 ACTION，程序中的输出会被清除 (做 Final scan)。
4. 当 M2 由 OFF→ON 时，TestSFC1 POU 执行动作，执行动作请参考 TestSFC1 POU 程序内容。设定控制参数 1，会清除 SFC 中之状态及参数，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 中之输出保持不变，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之 Step 开始执行。

注一：SFCRUN 要等到下一个扫描时间才会启动指定的 SFC POU。

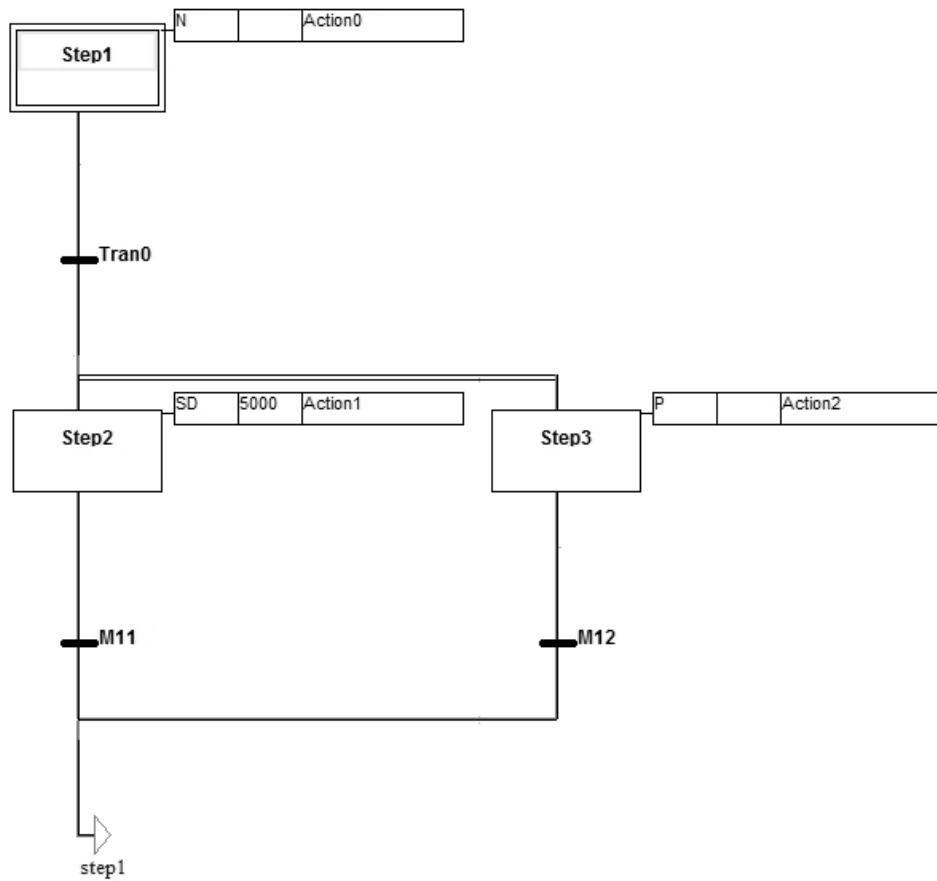
Main POU

6



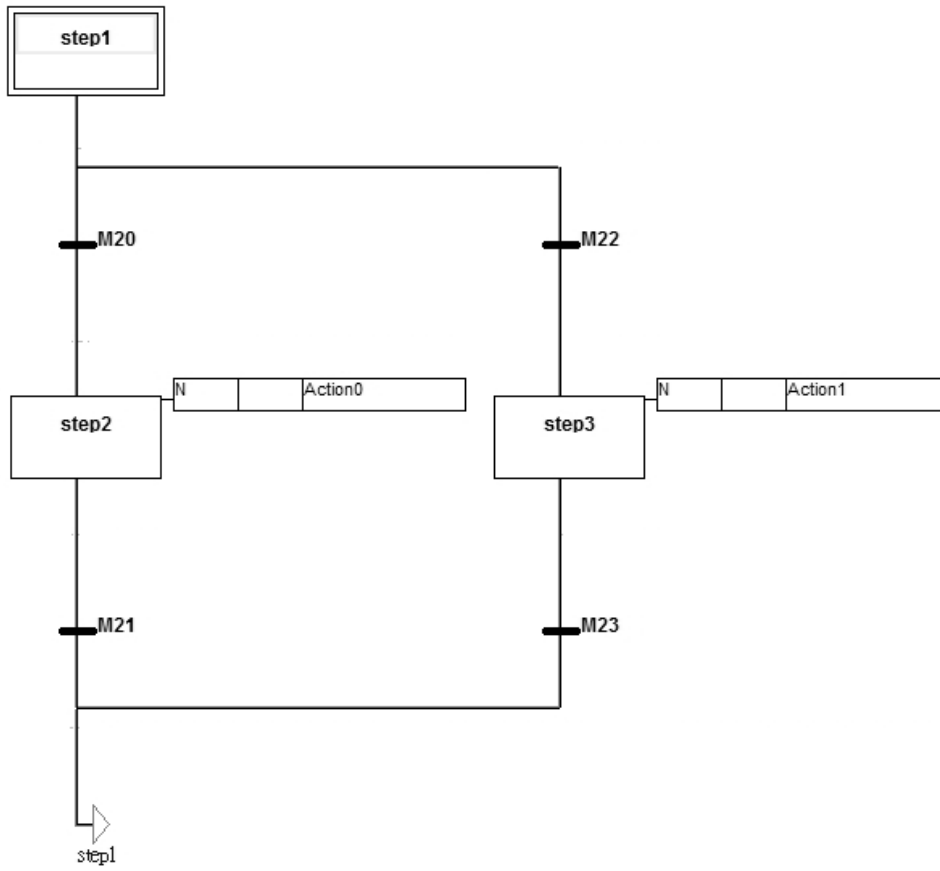


TestSFC1 POU



6

TestSFC2 POU



6

补充说明：

有关 SFC 的说明，请参考 ISPSOft 使用说明。

API	指令码			操作数								功能					
2501	SFCPSE			$S_1 \cdot S_2$								SFC 暂停					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	"\$"	DF	
S_1																	
S_2	●	●						●	●		●		○	○			
										脉冲执行型		16 位指令 (5 steps)			32 位指令		
										-		AH500			-		

符号：

	SFCPSE
En	
S1	
S2	

S_1 : SFC 名称

S_2 : 控制参数 Word

指令说明：

1. 依照 S_2 的设定，暂停 S_1 所指定的SFC程序。
2. 指令执行时， S_1 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：
 - S_1 为SFC程序的名称。
 - $S_2=0$ ，暂停 S_1 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。
 - $S_2=1$ ，暂停 S_1 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. S_2 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：

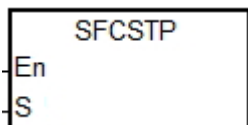
有关 SFC 的说明，请参考 ISPSOFT 使用说明。

API	指令碼			操作數						功能			
2502	SFCSTP			S						SFC 停止			

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	“\$”	DF
S																

脉冲執行型	16 位指令 (3 steps)	32 位指令
-	AH500	-

符号：



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。

6

补充说明：

有关 SFC 的说明，请参考 ISPSOFT 使用说明。

6.26 冗余系统控制指令

6.26.1 冗余系统控制指令一览表

API	指令码 (位)		P 指令	功能	STEPS
	16	32			
<u>2900</u>	SSO	—	✓	冗余系统切换	3
<u>2901</u>	RCS	—	—	读取/设定冗余系统信息	5

6.26.2 冗余系统控制指令说明

API	指令码			操作数								功能					
2900		SSO	P	D								冗余系统切换					
装置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●		●				

脉冲执行型	16 位指令 (3 steps)	32 位指令
AH500	AH500	-

符号：



D : 错误码 Word

指令说明：

1. 本指令为 AHCPU560-EN2 冗余系统主机专用指令。
2. SSOP 指令执行时，系统会从主 CPU 切换至冗余 CPU。
3. 本指令只支持脉冲执行型指令。
4. 在使用本指令之前，需用 ISPSOft -> HWCONFIG 启动冗余模式。
PLC 参数设定 -> CPU -> 系统 勾选启动冗余模式。
5. 此指令仅支持于冗余系统成立的情形下，意即冗余 CPU 必须存在且通过资格检定。

6

错误码：

D	说明
0	系统切换成功
1	系统切换失败，冗余 CPU 不符合资格
2	系统切换失败，冗余 CPU 不存在

程序范例：

当 M0=ON 时，则会执行系统切换，且 D0=0。

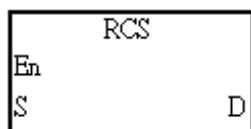


API	指令码				操作数				功能			
2901	RCS				S · D				读取/设定冗余系统信息			

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

脉冲执行型	16 位指令 (5 steps)	32 位指令
-	AH500	-

符号：



S : 冗余系统信息类别 Word

D : 冗余系统信息与执行结果 Word

指令说明：

1. 本指令为 AHCPU560-EN2 冗余系统主机专用指令。
2. 操作数 S 占用 2 个连续装置。S · S+1。
3. 操作数 D 占用 2 个连续装置。D · D+1。
4. S : 范围 0 ~ 3。
5. 在使用本指令之前，需用 ISPSOft -> HWCONFIG 启动冗余模式。
 - PLC 参数设定 -> CPU -> 系统 勾选启动冗余模式。

说明列表：

S	说明	S+1	说明	D	说明
0	系统 ID	0	读取当前主机 ID	1	ID 为 A
				2	ID 为 B
		1	设定主机 ID 为 A	不使用	
				2	设定主机 ID 为 B
1	冗余系统	不使用	1		
			2	冗余成功	
			3	冗余功能未启动	
			4	未通过资格检定	
			5	不存在冗余 CPU	
2	电源模块	不使用	1	电源模块正常	
			2	第一电源模块异常	
			3	第二电源模块异常	

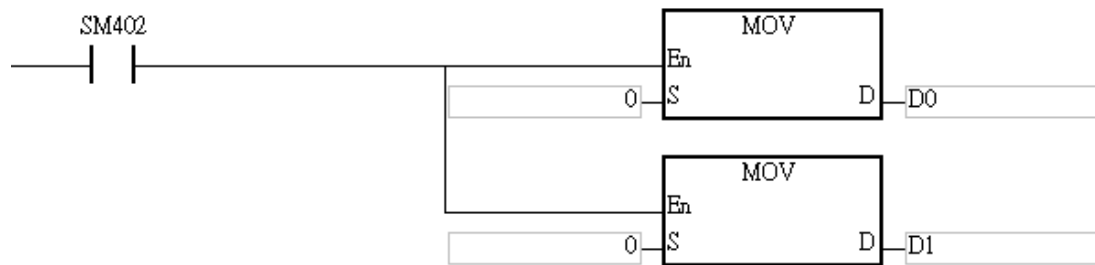
6

S	说明	S+1	说明	D	说明
3	冗余控制器		不使用	D	
				Bit0	冗余 CPU 不存在
				Bit1	RUN 灯常亮
				Bit2	RUN 灯灭
				Bit3	RUN 灯闪烁
				Bit4	ERROR 灯常亮
				Bit5	ERROR 灯灭
				Bit6	ERROR 灯闪烁
				Bit7	BUS FAULT 灯常亮
				Bit8	BUS FAULT 灯灭
				Bit9	BUS FAULT 灯闪烁
				Bit10	SYSTEM 灯常亮
				Bit11	SYSTEM 灯灭
				Bit12	SYSTEM 灯闪烁
				Bit13~ Bit15	不使用
				D+1	
				Bit0	MASTER 灯常亮
				Bit1	MASTER 灯灭
				Bit2	MASTER 灯闪烁
				Bit3	SYNC 黄灯常亮
				Bit4	SYNC 黄灯闪烁
				Bit5	SYNC 绿灯常亮
				Bit6	SYNC 绿灯闪烁
				Bit7	SYNC 红灯常亮
				Bit8	SYNC 红灯闪烁
				Bit9	SYNC 灯灭
				Bit10~ Bit15	不使用

程序范例：

此范例为读取当前主机 ID。当 M0=ON 时，如 D10=1，则此主机 ID 为 A。D10=2 时，此主机 ID 为 B。

Network 1



Network 2



MEMO

6



第7章 错误代码

目录

7.1	模块错误代码对应灯号及状态说明	7-2
7.1.1	CPU 模块错误代码对应灯号及状态	7-2
7.1.2	模拟模块与温度模块错误代码对应灯号	7-21
7.1.3	AH02HC-5A/AH04HC-5A 错误代码对应灯号	7-22
7.1.4	AH05PM-5A/AH10PM-5A/AH15PM-5A 错误代码对应灯号...	7-23
7.1.5	AH20MC-5A 错误代码对应灯号	7-24
7.1.6	AH10EN-5A/AH15EN-5A 错误代码对应灯号	7-25
7.1.7	AH10SCM-5A/AH15SCM-5A 错误代码对应灯号	7-25
7.1.8	AH10DNET-5A 错误代码对应灯号	7-25
7.1.9	AH10PFBM-5A 错误代码对应灯号	7-26
7.1.10	AH10PFBS-5A 错误代码对应灯号	7-27
7.1.11	AH10COPM-5A 错误代码对应灯号	7-28

7.1 模块错误代码对应灯号及状态说明

A. 字段简介

- a. 错误代码：该错误发生时系统所产生的错误代码。
- b. 说明：该错误的说明。
- c. 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。
 - 自定：CPU 状态的变化可让用户自行定义，请参考 AH500 操作手册第 8.2.1 节。
- d. 灯号状态：该错误发生时的主机灯号变化。
 - ERROR：系统错误灯号。
 - BUS FAULT：I/O 总线错误灯号。
 - MODULE ERROR：模块错误灯号。

● 各灯号说明

	灯号	说明
CPU	Error LED	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 LED	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。
MODULE	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。

7.1.1 CPU 模块错误代码对应灯号及状态

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

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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	无法执行系统还原程序 (错误标志 SM9)	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	IO 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
16#0022	下载 CPU 模块参数校验错误	停止	常亮	保持
16#0023	系统 PLC 参数设定内 · Y 装置状态 (STOP -> RUN) 设定选项错误	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 通信设置设置错误 (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 通信设置设置错误 (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

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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#005D	CPU 模块检测不到储存卡 (错误标志 SM453)	持续	保持	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	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#0060	CPU 模块无法建立默认文件夹 (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
16#1400	协处理器存取错误	停止	保持	常亮
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#1407	协处理器通讯错误 (错误标志 SM9)	持续	常亮	保持
16#1409	扩展背板连线中断 (错误标志 SM9)	停止	保持	常亮
16#140A	扩展背板通讯错误 (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	高速数据交换校验错误 (错误标志 SM9)	停止	保持	常亮
16#140D	实际电源 ID 不符合配置设定 (错误标志 SM9)	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 (错误标志 SM10)	停止	保持	常亮
16#140F	高速数据交换错误 (错误标志 SM11)	停止	保持	常亮
16#1410	RTU IO 模块发生错误	停止	保持	常亮

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#1411	RTU IO 模块发出警告	持续	保持	常亮
16#1420	模块的 Ethernet port 发生 Link off 状态	持续	保持	常亮
16#1421	主机读取模块之智能型模块设定信息发生错误	停止	保持	常亮
16#1422	主机写入模块之智能型模块设定信息发生错误	停止	保持	常亮
16#1801	CPU 模块未设定中断工作	持续	保持	保持
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 (错误标志 SM5)	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	部份指令不充许操作数发生重叠 (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

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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
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 回复命令校验码错误 (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 指令的中断指针异常	自定义	闪烁	保持
16#2027	FWD 应用指令异常 (错误标志 SM102/103)	自定义	闪烁	保持
16#2028	REV 应用指令异常 (错误标志 SM102/103)	自定义	闪烁	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#2029	停止应用指令异常 (错误标志 SM102/103)	自定义	闪烁	保持
16#202A	RSMT 应用指令异常 (错误标志 SM102/103)	自定义	闪烁	保持
16#202B	RSTEF 应用指令异常 (错误标志 SM102/103)	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
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

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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#6000	以太网检测速率失败 (错误标志 SM1106)	持续	闪烁	保持
16#6001	IP 地址不合法 (错误标志 SM1107)	持续	闪烁	保持
16#6002	网络屏蔽地址不合法 (错误标志 SM1107)	持续	闪烁	保持
16#6003	网关地址不合法 (错误标志 SM1107)	持续	闪烁	保持
16#6004	以太网络的 IP 地址过滤设置错误 (标志 SM1108)	持续	闪烁	保持
16#6006	以太网络的静态 ARP 表设置错误 (标志 SM1108)	持续	闪烁	保持
16#6007	NTP 设置错误 (错误标志 SM1380)	持续	闪烁	保持
16#6008	网络编号不合法 (错误标志 SM1107)	持续	闪烁	保持
16#6009	节点编号不合法 (错误标志 SM1107)	持续	闪烁	保持
16#600A	TCP 连线建立失败 (错误标志 SM1090)	持续	保持	保持
16#600B	UDP 连线建立失败 (错误标志 SM1091)	持续	保持	保持
16#600C	Socket 通讯接口已被使用 (错误标志 SM1109)	持续	保持	保持
16#600D	RJ45 接口未连接 (错误标志 SM1100)	持续	保持	保持
16#600E	AH10EN 上 RJ45 接口未连接网络线	持续	保持	保持
16#600F	Modbus TCP 服务器联机已满 (错误标志 SM1089)	持续	闪烁	保持
16#6010	IP 设定错误 (错误标志 SM1107)	持续	保持	保持
16#6011	DNS 设定错误 (错误标志 SM1107)	持续	保持	保持
16#6012	IP 地址重复错误 (错误标志 SM1107)	持续	闪烁	保持
16#6100	E-mail 连线忙碌 (错误标志 SM1113)	持续	保持	保持
16#6101	E-mail 发送条件的触发设定错误 (标志 SM1112)	持续	闪烁	保持
16#6102	E-mail 发送条件的发送时间间隔设定错误 (错误标志 SM1112)	持续	闪烁	保持
16#6103	E-mail 附件中的装置地址设定错误 (标志 SM1112)	持续	闪烁	保持
16#6104	E-mail 附件不存在 (错误标志 SM1113)	持续	保持	保持
16#6105	E-mail 附件超过容量 (错误标志 SM1113)	持续	保持	保持
16#6106	SMTP 服务器地址错误 (错误标志 SM1112)	持续	闪烁	保持
16#6107	SMTP 服务器超时 (错误标志 SM1113)	持续	保持	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#6108	寄件服务器验证错误 (错误标志 SM1112)	持续	闪烁	保持
16#6110	SMTP 服务器需要进行验证 (错误标志 SM1112)	持续	闪烁	保持
16#6111	指定的 E-mail 地址不存在 (错误标志 SM1112)	持续	闪烁	保持
16#6200	TCP 通讯接口 (Socket) 功能的远程 IP 地址不合法 (错误标志 SM1196)	持续	闪烁	保持
16#6201	TCP 通讯接口 (Socket) 功能的本地端口不合法	持续	保持	保持
16#6202	TCP 通讯接口 (Socket) 功能的远程端口不合法	持续	保持	保持
16#6203	TCP 通讯接口 (Socket) 功能的传送数据地址不合法	持续	保持	保持
16#6204	TCP 通讯接口 (Socket) 功能的传送数据长度不合法	持续	保持	保持
16#6205	TCP 通讯接口 (Socket) 功能的传送数据装置超出范围	持续	保持	保持
16#6206	TCP 通讯接口 (Socket) 功能的接收数据地址不合法	持续	保持	保持
16#6207	TCP 通讯接口 (Socket) 功能的接收数据长度不合法	持续	保持	保持
16#6208	TCP 通讯接口 (Socket) 功能的接收数据装置超出范围	持续	保持	保持
16#6209	UDP 通讯接口 (Socket) 功能的远程 IP 地址不合法 (错误标志 SM1196)	持续	闪烁	保持
16#620A	UDP 通讯接口 (Socket) 功能的本地端口不合法	持续	保持	保持
16#620B	UDP 通讯接口 (Socket) 功能的远程端口不合法	持续	保持	保持
16#620C	UDP 通讯接口 (Socket) 功能的传送数据地址不合法	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#620E	UDP 通讯接口 (Socket) 功能的传送数据装置超出范围	持续	保持	保持
16#620F	UDP 通讯接口 (Socket) 功能的接收数据地址不合法	持续	保持	保持
16#6210	UDP 通讯接口 (Socket) 功能的接收数据长度不合法	持续	保持	保持
16#6211	UDP 通讯接口 (Socket) 功能的接收数据装置超出范围	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	远程装置拒绝连线	持续	保持	保持
16#6215	目前通讯接口 (Socket) 未开启	持续	保持	保持
16#6217	目前通讯接口 (Socket) 已开启	持续	保持	保持
16#6218	目前通讯接口 (Socket) 已传送	持续	保持	保持
16#6219	目前通讯接口 (Socket) 已接收	持续	保持	保持
16#621A	目前通讯接口 (Socket) 已关闭	持续	保持	保持
16#6300	Ether Link 只可用于装置 M、D、L	持续	闪烁	保持
16#6301	Ether Link 装置地址设定超过可用的装置范围	持续	闪烁	保持
16#6302	Ether Link 的数据长度超过限制	持续	闪烁	保持
16#6303	Ether Link 的远程装置中止连线	持续	保持	保持
16#6304	Ether Link 连线忙碌	持续	保持	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#6305	Ether Link 通讯命令中的节点与本地节点不同	持续	闪烁	保持
16#6309	Ether Link 回应超时	持续	保持	保持
16#630A	模块的 ID 或设定与 Ether Link 中的设定不同	持续	闪烁	保持
16#630B	CPU 或模块的网络屏蔽设定与 Ether Link 设定不同	持续	闪烁	保持
16#6400	EMDRW 指令操作的连线数超出限制或未设定送信标志	持续	保持	保持
16#6401	远程装置中止连线	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#6500	初始化数据交换功能时错误 (错误标志 SM699)	持续	闪烁	OFF
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#7404	报文格式错误	持续	保持	保持
16#7405	位组长度 (Byte Length) 的数据错误	持续	保持	保持
16#7406	校验 (Checksum) 错误	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

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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
16#8236	下载的主机参数有误：NTP 设定错误	持续	保持	保持
16#8239	下载的主机参数有误：Email 设定错误	持续	保持	保持
16#823A	下载的主机参数有误：Email 触发设定错误	持续	保持	保持
16#823B	下载的主机参数有误：TCP 通讯接口 (Socket) 设定错误	持续	保持	保持
16#823C	下载的主机参数有误：UDP 通讯接口 (Socket) 设定错误	持续	保持	保持
16#823E	下载的主机参数有误：Web 设定错误	持续	保持	保持
16#8240	下载的主机参数有误：Ether Link	持续	保持	保持
16#8241	DNS 设定错误	持续	保持	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	闪烁	闪烁
16#8611	储存卡不存在或储存卡格式错误	持续	保持	保持
16#8612	储存卡存取错误或储存卡是只读模式	持续	保持	保持
16#9A01	PLC Link / COM1 MODBUS 从站 1 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9A02	PLC Link / COM1 MODBUS 从站 2 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A03	PLC Link / COM1 MODBUS 从站 3 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A04	PLC Link / COM1 MODBUS 从站 4 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A05	PLC Link / COM1 MODBUS 从站 5 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A06	PLC Link / COM1 MODBUS 从站 6 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A07	PLC Link / COM1 MODBUS 从站 7 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A08	PLC Link / COM1 MODBUS 从站 8 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A09	PLC Link / COM1 MODBUS 从站 9 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A0A	PLC Link / COM1 MODBUS 从站 10 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A0B	PLC Link / COM1 MODBUS 从站 11 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A0C	PLC Link / COM1 MODBUS 从站 12 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A0D	PLC Link / COM1 MODBUS 从站 13 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A0E	PLC Link / COM1 MODBUS 从站 14 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A0F	PLC Link / COM1 MODBUS 从站 15 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A10	PLC Link / COM1 MODBUS 从站 16 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A11	PLC Link / COM1 MODBUS 从站 17 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A12	PLC Link / COM1 MODBUS 从站 18 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A13	PLC Link / COM1 MODBUS 从站 19 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9A14	PLC Link / COM1 MODBUS 从站 20 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A15	PLC Link / COM1 MODBUS 从站 21 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A16	PLC Link / COM1 MODBUS 从站 22 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A17	PLC Link / COM1 MODBUS 从站 23 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A18	PLC Link / COM1 MODBUS 从站 24 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A19	PLC Link / COM1 MODBUS 从站 25 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A1A	PLC Link / COM1 MODBUS 从站 26 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A1B	PLC Link / COM1 MODBUS 从站 27 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A1C	PLC Link / COM1 MODBUS 从站 28 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A1D	PLC Link / COM1 MODBUS 从站 29 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A1E	PLC Link / COM1 MODBUS 从站 30 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A1F	PLC Link / COM1 MODBUS 从站 31 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A20	PLC Link / COM1 MODBUS 从站 32 的数据交换设定错误 (错误标志 SM1590)	持续	保持	保持
16#9A21	PLC Link / COM1 MODBUS 从站 1 通讯错误	持续	保持	保持
16#9A22	PLC Link / COM1 MODBUS 从站 2 通讯错误	持续	保持	保持
16#9A23	PLC Link / COM1 MODBUS 从站 3 通讯错误	持续	保持	保持
16#9A24	PLC Link / COM1 MODBUS 从站 4 通讯错误	持续	保持	保持
16#9A25	PLC Link / COM1 MODBUS 从站 5 通讯错误	持续	保持	保持
16#9A26	PLC Link / COM1 MODBUS 从站 6 通讯错误	持续	保持	保持
16#9A27	PLC Link / COM1 MODBUS 从站 7 通讯错误	持续	保持	保持
16#9A28	PLC Link / COM1 MODBUS 从站 8 通讯错误	持续	保持	保持
16#9A29	PLC Link / COM1 MODBUS 从站 9 通讯错误	持续	保持	保持
16#9A2A	PLC Link / COM1 MODBUS 从站 10 通讯错误	持续	保持	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9A2B	PLC Link / COM1 MODBUS 从站 11 通讯错误	持续	保持	保持
16#9A2C	PLC Link / COM1 MODBUS 从站 12 通讯错误	持续	保持	保持
16#9A2D	PLC Link / COM1 MODBUS 从站 13 通讯错误	持续	保持	保持
16#9A2E	PLC Link / COM1 MODBUS 从站 14 通讯错误	持续	保持	保持
16#9A2F	PLC Link / COM1 MODBUS 从站 15 通讯错误	持续	保持	保持
16#9A30	PLC Link / COM1 MODBUS 从站 16 通讯错误	持续	保持	保持
16#9A31	PLC Link / COM1 MODBUS 从站 17 通讯错误	持续	保持	保持
16#9A32	PLC Link / COM1 MODBUS 从站 18 通讯错误	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#9A36	PLC Link / COM1 MODBUS 从站 22 通讯错误	持续	保持	保持
16#9A37	PLC Link / COM1 MODBUS 从站 23 通讯错误	持续	保持	保持
16#9A38	PLC Link / COM1 MODBUS 从站 24 通讯错误	持续	保持	保持
16#9A39	PLC Link / COM1 MODBUS 从站 25 通讯错误	持续	保持	保持
16#9A3A	PLC Link / COM1 MODBUS 从站 26 通讯错误	持续	保持	保持
16#9A3B	PLC Link / COM1 MODBUS 从站 27 通讯错误	持续	保持	保持
16#9A3C	PLC Link / COM1 MODBUS 从站 28 通讯错误	持续	保持	保持
16#9A3D	PLC Link / COM1 MODBUS 从站 29 通讯错误	持续	保持	保持
16#9A3E	PLC Link / COM1 MODBUS 从站 30 通讯错误	持续	保持	保持
16#9A3F	PLC Link / COM1 MODBUS 从站 31 通讯错误	持续	保持	保持
16#9A40	PLC Link / COM1 MODBUS 从站 32 通讯错误	持续	保持	保持
16#9A41	PLC Link / COM1 MODBUS 从站 1 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A42	PLC Link / COM1 MODBUS 从站 2 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A43	PLC Link / COM1 MODBUS 从站 3 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A44	PLC Link / COM1 MODBUS 从站 4 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A45	PLC Link / COM1 MODBUS 从站 5 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A46	PLC Link / COM1 MODBUS 从站 6 无回应 (错误标志 SM1591)	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

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9A48	PLC Link / COM1 MODBUS 从站 8 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A49	PLC Link / COM1 MODBUS 从站 9 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A4A	PLC Link / COM1 MODBUS 从站 10 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A4B	PLC Link / COM1 MODBUS 从站 11 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A4C	PLC Link / COM1 MODBUS 从站 12 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A4D	PLC Link / COM1 MODBUS 从站 13 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A4E	PLC Link / COM1 MODBUS 从站 14 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A4F	PLC Link / COM1 MODBUS 从站 15 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A50	PLC Link / COM1 MODBUS 从站 16 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A51	PLC Link / COM1 MODBUS 从站 17 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A52	PLC Link / COM1 MODBUS 从站 18 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A53	PLC Link / COM1 MODBUS 从站 19 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A54	PLC Link / COM1 MODBUS 从站 20 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A55	PLC Link / COM1 MODBUS 从站 21 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A56	PLC Link / COM1 MODBUS 从站 22 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A57	PLC Link / COM1 MODBUS 从站 23 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A58	PLC Link / COM1 MODBUS 从站 24 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A59	PLC Link / COM1 MODBUS 从站 25 无回应 (错误标志 SM1591)	持续	保持	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#9A5A	PLC Link / COM1 MODBUS 从站 26 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A5B	PLC Link / COM1 MODBUS 从站 27 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A5C	PLC Link / COM1 MODBUS 从站 28 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A5D	PLC Link / COM1 MODBUS 从站 29 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A5E	PLC Link / COM1 MODBUS 从站 30 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A5F	PLC Link / COM1 MODBUS 从站 31 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A60	PLC Link / COM1 MODBUS 从站 32 无回应 (错误标志 SM1591)	持续	保持	保持
16#9A61	PLC Link Mode 设定错误 (错误标志 SM1589)	持续	保持	保持
16#9A62	PLC Link 轮询次数设定错误 (错误标志 SM1596)	持续	保持	保持
16#9A63	主机与通讯模块交握逾时 (错误标志 SM1596)	持续	保持	保持
16#9A64	主机内无通讯模块参数 (错误标志 SM1596)	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
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

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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
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

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
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
16#B100	I/O Connection 重复建立	持续	保持	保持
16#B106	多 Scanner 建立 I/O Connection 冲突	持续	保持	保持
16#B110	Adapter configuration 参数设定错误	持续	保持	保持
16#B111	Adapter RPI 参数设定错误	持续	保持	保持
16#B113	I/O connection 联机数不足	持续	保持	保持
16#B119	Non-Listen only 联机建立失败	持续	保持	保持
16#B127	Adapter Input size 参数错误	持续	保持	保持
16#B128	Adapter output size 设定错误	持续	保持	保持
16#B129	EDS 檔 Configuration path 参数错误	持续	保持	保持
16#B12D	Consumed tag 参数错误	持续	保持	保持
16#B12E	Produced tag 参数错误	持续	保持	保持
16#B203	I/O connection 通讯超时	持续	保持	保持
16#B204	建立 I/O Connection 时通讯超时	持续	保持	保持
16#B302	网络配置超过产品 PPS 规格	持续	保持	保持
16#B315	Adapter input/output instance 参数设定错误	持续	保持	保持
16#E206	资格检定失败(冗余控制器与主控制型号不一致)	持续	保持	保持
16#E207	资格检定失败(韧体版本不兼容)	持续	保持	保持
16#E208	冗余控制器和主控制器以太网络不在相同的实体网域	持续	保持	保持
16#E209	主系统和冗余系统实际 I/O 配置不相符(资格检定期间)	持续	保持	保持
16#E20A	冗余系统和主系统实际 I/O 配置不相符(资格检定过后)	持续	保持	保持
16#E20B	资格检定失败(系统错误, 请参考错误纪录)	持续	保持	保持
16#E20C	资格检定失败(下载中, 无法同步)	持续	保持	保持
16#E20D	资格检定失败(请参考冗余主机的错误纪录)	持续	保持	保持
16#E20E	资格检定失败(I/O 总线错误)	持续	保持	保持
16#E20F	资格检定失败(heart beat 错误, 请参考冗余主机的错误纪录)	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#E210	资格检定失败(heart beat 通讯超时)	持续	保持	保持
16#E211	资格检定失败(同步数据失败)	持续	保持	保持
16#E212	资格检定失败(冗余系统切换中)	持续	保持	保持
16#E213	资格检定失败(PLC 无程序)	持续	保持	保持
16#E214	资格检定失败(PLC 程序毁损)	持续	保持	保持
16#E215	资格检定失败(扫描超时)	持续	保持	保持
16#E216	资格检定失败(CPU 内存存取被拒)	持续	保持	保持
16#E217	资格检定失败(系统忙碌 RST)	持续	保持	保持
16#E218	资格检定失败(系统忙碌 CLR)	持续	保持	保持
16#E219	资格检定失败(系统开机未完成)	持续	保持	保持
16#E21A	资格检定失败(系统开机失败)	持续	保持	保持
16#E21B	资格检定失败(CPU 参数毁损,请参考错误纪录)	持续	保持	保持
16#E21C	资格检定失败(停电保持区块异常,请参考错误纪录)	持续	保持	保持
16#E21D	资格检定失败(CPU 参数毁损,请参考错误纪录)	持续	保持	保持
16#E21E	资格检定失败(I/O 配置表不存在)	持续	保持	保持
16#E21F	资格检定失败(I/O 配置表损毁)	持续	保持	保持
16#E221	资格检定失败(PLC 程序执行错误,请参考错误纪录)	持续	保持	保持
16#E230	冗余系统以太网联机异常	持续	保持	保持
16#E260	资格检定失败(主背板第 0 槽模块不支持冗余系统)	持续	保持	保持
16#E261	资格检定失败(主背板第 1 槽模块不支持冗余系统)	持续	保持	保持
16#E262	资格检定失败(主背板第 2 槽模块不支持冗余系统)	持续	保持	保持
16#E263	资格检定失败(主背板第 3 槽模块不支持冗余系统)	持续	保持	保持
16#E264	资格检定失败(主背板第 4 槽模块不支持冗余系统)	持续	保持	保持
16#E265	资格检定失败(主背板第 5 槽模块不支持冗余系统)	持续	保持	保持
16#E266	资格检定失败(主背板第 6 槽模块不支持冗余系统)	持续	保持	保持
16#E267	资格检定失败(主背板第 7 槽模块不支持冗余系统)	持续	保持	保持
16#E268	资格检定失败(主背板第 8 槽模块不支持冗余系统)	持续	保持	保持
16#E269	资格检定失败(主背板第 9 槽模块不支持冗余系统)	持续	保持	保持
16#E26A	资格检定失败(主背板第 10 槽模块不支持冗余系统)	持续	保持	保持
16#E26B	资格检定失败(主背板第 11 槽模块不支持冗余系统)	持续	保持	保持
16#E270	资格检定失败(主背板第 0 槽网络模块网络线未连接)	持续	保持	保持
16#E271	资格检定失败(主背板第 1 槽网络模块网络线未连接)	持续	保持	保持
16#E272	资格检定失败(主背板第 2 槽网络模块网络线未连接)	持续	保持	保持
16#E273	资格检定失败(主背板第 3 槽网络模块网络线未连接)	持续	保持	保持
16#E274	资格检定失败(主背板第 4 槽网络模块网络线未连接)	持续	保持	保持
16#E275	资格检定失败(主背板第 5 槽网络模块网络线未连接)	持续	保持	保持

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#E276	资格检定失败(主背板第 6 槽网络模块网络线未连接)	持续	保持	保持
16#E277	资格检定失败(主背板第 7 槽网络模块网络线未连接)	持续	保持	保持
16#E278	资格检定失败(主背板第 8 槽网络模块网络线未连接)	持续	保持	保持
16#E279	资格检定失败(主背板第 9 槽网络模块网络线未连接)	持续	保持	保持
16#E27A	资格检定失败(主背板第 10 槽网络模块网络线未连接)	持续	保持	保持
16#E27B	资格检定失败(主背板第 11 槽网络模块网络线未连接)	持续	保持	保持
16#E280	资格检定失败(主背板第 0 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E281	资格检定失败(主背板第 1 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E282	资格检定失败(主背板第 2 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E283	资格检定失败(主背板第 3 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E284	资格检定失败(主背板第 4 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E285	资格检定失败(主背板第 5 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E286	资格检定失败(主背板第 6 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E287	资格检定失败(主背板第 7 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E288	资格检定失败(主背板第 8 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E289	资格检定失败(主背板第 9 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E28A	资格检定失败(主背板第 10 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E28B	资格检定失败(主背板第 11 槽网络通讯模块网络模块 IP 检测失败)	持续	保持	保持
16#E290	主背板第 0 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E291	主背板第 1 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E292	主背板第 2 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E293	主背板第 3 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E294	主背板第 4 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E295	主背板第 5 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持

7

错误代码	说明	CPU 状态	灯号状态	
			ERROR	BUS FAULT
16#E296	主背板第 6 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E297	主背板第 7 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E298	主背板第 8 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E299	主背板第 9 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E29A	主背板第 10 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持
16#E29B	主背板第 11 槽网络通讯模块通讯口 heart beat 检测失败	持续	保持	保持

7.1.2 模拟模块与温度模块错误代码对应灯号

错误代码	说明	灯号状态	
		CPU BUS FAULT	MODULE 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#A201	CH0 输入信号超出硬件规格	闪烁	
16#A202	CH1 输入信号超出硬件规格	闪烁	
16#A203	CH2 输入信号超出硬件规格	闪烁	
16#A204	CH3 输入信号超出硬件规格	OFF	
16#A400	CH4 输入信号超出硬件规格	常亮	
16#A401	CH5 输入信号超出硬件规格	常亮	
16#A402	CH6 输入信号超出硬件规格	常亮	
16#A403	CH7 输入信号超出硬件规格	常亮	
16#A404	插槽电源异常	常亮	
16#A405	电源异常	常亮	
16#A406	内部错误·CJC 补偿异常	常亮	
16#A407	内部错误·出厂校正异常	常亮	
16#A600	CH0 输入信号超出硬件规格	常亮	
16#A601	CH1 输入信号超出硬件规格	常亮	
16#A602	CH2 输入信号超出硬件规格	常亮	
16#A603	CH3 输入信号超出硬件规格	常亮	
16#A800	CH4 输入信号超出硬件规格	OFF	

7

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A801	CH5 输入信号超出硬件规格	OFF	
16#A802	CH6 输入信号超出硬件规格	OFF	
16#A803	CH7 输入信号超出硬件规格	OFF	
16#A804	CH0 输入信号超出硬件规格	OFF	
16#A805	CH1 输入信号超出硬件规格	OFF	
16#A806	CH2 输入信号超出硬件规格	OFF	
16#A807	CH3 输入信号超出硬件规格	OFF	

*1. 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 的那个区段。

7.1.3 AH02HC-5A/AH04HC-5A 错误代码对应灯号

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 报警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 报警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	
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 报警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 报警输出设定极限值错误	闪烁	

7

错误代码	说明	灯号状态	
		CPU	MODULE
		BUS FAULT	ERROR
16#A036	CH3 中断编号设定超过范围	闪烁	

7.1.4 AH05PM-5A/AH10PM-5A/AH15PM-5A 错误代码对应灯号

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 指令缺少对应的指针	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#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	装置元件使用范围错误	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	
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	

7.1.5 AH20MC-5A 错误代码对应灯号

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 指令缺少对应的指针	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#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	装置元件使用范围错误	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	
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	

7

7.1.6 AH10EN-5A/AH15EN-5A 错误代码对应灯号

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	
16#A002	Host 2 IP 地址冲突	闪烁	
16#A003	Host 1 DHCP 失败	闪烁	
16#A004	Host 2 DHCP 失败	闪烁	
16#A401	硬件错误	常亮	
16#A402	系统初始化失败	常亮	

7.1.7 AH10SCM-5A/AH15SCM-5A 错误代码对应灯号

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	

7.1.8 AH10DNET-5A 错误代码对应灯号

错误代码	说明	灯号状态		
		CPU	MODULE	
		BUS FAULT	MS	NS
16#A0F0	AH10DNET-5A 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 AH10DNET-5A 扫描列表中	红灯闪烁	绿灯 闪烁	绿灯 常亮
16#A0F2	AH10DNET-5A 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	AH10DNET-5A 扫描模块进入测试模式	红灯闪烁	橙灯 常亮	橙灯 常亮
16#A0F4	AH10DNET-5A 扫描模块进入 Bus-OFF 状态	红灯闪烁	绿灯 常亮	红灯 常亮
16#A0F5	AH10DNET-5A 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

错误代码	说明	灯号状态		
		CPU	MODULE	
		BUS FAULT	MS	NS
16#A0F6	AH10DNET-5A 扫描模块的内部存储单元出错	红灯闪烁	红灯常亮	绿灯闪烁
16#A0F7	AH10DNET-5A 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	AH10DNET-5A 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	AH10DNET-5A 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	AH10DNET-5A 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

7.1.9 AH10PFBM-5A 错误代码对应灯号

7

错误代码	说明	灯号状态			
		CPU	MODULE		
		BUS FAULT	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
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

错误代码	说明	灯号状态			
		CPU	MODULE		
		BUS FAULT	RUN	SYS	DP
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

7.1.10 AH10PFBS-5A 错误代码对应灯号

错误代码	说明	灯号状态		
		CPU	MODULE	
		BUS FAULT	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 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	红灯 常亮	绿灯 常亮	红灯 常亮

7

7.1.11 AH10COPM-5A 错误代码对应灯号

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

7