



台達電子工業股份有限公司

33068 桃園縣桃園市興隆路18號
TEL:886-3-3626301
FAX:886-3-3716301

*本使用手冊內容若有變更，恕不另行通知

AH-0109700-04

AH500 程式手冊



AH500 程式手冊



AH500 程式手冊

版本修訂一覽表

版本	變更內容	發行日期
第一版	第一版發行	2012/06/30
第二版	<ol style="list-style-type: none">第 1 章：更新第 1.1.2 節機種簡易說明內容第 2 章：更新第 2.1.1 節裝置列表、第 2.1.4 節停電保持區的裝置範圍、第 2.2.3 節字串、第 2.2.7 節特殊輔助繼電器 SM、第 2.2.8 節特殊輔助繼電器 SM 的更新時間、第 2.2.14 節特殊資料暫存器 SR、第 2.2.15 節特殊資料暫存器 SR 的更新時間及第 2.2.16 節 SM/SR 補充說明第 3 章：增加指令 API0117、API0118、API0708、API1812。更新指令 API2207 及刪除 API1809~API1811 相關資料第 4 章：更新第 4.2 節中功能塊中不支援指令及範例說明第 6 章：增加指令 API0117、API0118、API0708、API1812、API2500、API2501、API2502 及刪除 API1809~API1811 相關資料；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 節、第 7.1.2 節、第 7.1.3 節、第 7.1.4 節、第 7.1.8 節、第 7.1.9 節、第 7.1.10 節內容	2015/11/20

版本	變更內容	發行日期
第三版	<ol style="list-style-type: none"> 1. 第 1 章：更新第 1.1.1 節相關手冊說明、增加 AH15SCM-5A/AHRTU-ETHN-5A/AH15EN-5A/AHCPU501-RS2/AHCPU521-RS2/AHCPU531-RS2/AHCPU501-EN 機種相關資訊及將所有配線模組型號變更為新型號 2. 第 2 章：更新第 2.2.3 節 ASCII 碼轉換表、第 2.2.7 節增加機種 AHCPU560-EN2/AHCPU501-RS2/AHCPU521-RS2/AHCPU531-RS2/AHCPU501-EN 相關資訊及新增/更新 SM、第 2.2.8 節特殊輔助繼電器 SM 的更新時間、第 2.2.14 節特殊資料暫存器 SR、第 2.2.15 節特殊資料暫存器 SR 的更新時間及第 2.2.16 節 SM/SR 補充說明 3. 第 3 章：第 3.1.2 節增加 API2900/API2901 及第 3.3 節更新 INV 指令符號 4. 第 6 章：更新指令 API0702 指令說明、API0904 程式範例二說明、API1111 補充說明、API1503/1504/1505 程式範例圖、API1701/1702 指令說明及程式範例圖、API1800/1812 指令說明、API2116 指令裝置、API2200 指令說明、API2300/2301/2302 指令說明及增加 API2900/API2901 5. 第 7 章：更新第 7.1.1 節內容及增加 AH15SCM-5A/AH15EN-5A 機種相關資訊 	2017/10/31
第四版	<ol style="list-style-type: none"> 1. 第 2 章第 2.2.7 節增加機種 AHCPU521-DNP3 相關資訊、新增 SM454-SM457、新增 SM2304-SM2319、更新 SM2048-SM2303；第 2.2.14 節增加機種 AHCPU521-DNP3 相關資訊、新增 SR418-SR430、新增 SR900-SR902、新增 SR2046-SR2047、更新 SR2048-SR2559；第 2.2.16 節更新第 15 項 TCP/UDP Socket 設定說明、更新第 19 項電源模組相關旗標說明、更新第 20 項備援延伸背板連線狀態說明、增加第 22 項 EtherNet/IP 相關旗標說明 2. 第 3 章第 3.1.2 節增加 API1607 及 API2208；第 3.4 節增加浮點數指令 64 位元內容、增加萬年曆指令 API1607 內容、增加乙太網控制指令 API2208 內容 3. 第 6 章 API0104 增加指令說明第 2 項、API1103 更新程式範例說明、API1401 指令說明增加 AH 備援系統使用說明、API1500-API1516 增加 64 位元指令說明、增加指令 API1607、API1808 補充說明更新當通訊命令為 0x05 與 0x06 時說明、增加指令 API2208、API2901 指令說明更新運算元 D、S 說明及說明列表內容 4. 第 7 章增加錯誤碼 16#0068/16#212A/16#6013/16#E206-16#E28B、16#E2A0-16#E2AB、更新錯誤碼 16#2026/16#2029/16#6010/16#6011/16#6212-16#6214/16#6218-16#621A/16#6400-16#6405/ 	2020/07/06

AH500 程式手冊

目錄

第 1 章 簡介

1.1 一般概述	1-2
1.1.1 相關手冊	1-2
1.1.2 機種簡易說明	1-2
1.2 軟體簡述	1-8
1.2.1 編輯軟體	1-8
1.2.2 程式組織單元 (POU) 與工作 (TASK)	1-10

第 2 章 程式規劃撰寫

2.1 裝置簡介	2-2
2.1.1 裝置列表	2-2
2.1.1.1 AH500 基本型 CPU 模組 (AHCPU500/510/520/530)	2-2
2.1.1.2 AH500 進階型 CPU 模組 (AHCPU501/511/521/531)	2-3
2.1.2 I/O 儲存區的基本結構	2-5
2.1.3 停電保持記憶方式	2-6
2.1.4 停電保持區的裝置範圍	2-6
2.2 裝置功能說明	2-7
2.2.1 數值、常數	2-7
2.2.2 浮點數	2-9
2.2.2.1 單精度浮點數 (32 位元浮點數)	2-9
2.2.2.2 雙精度浮點數 (64 位元浮點數)	2-10
2.2.2.3 十進浮點數	2-11
2.2.3 字串	2-11
2.2.4 輸入繼電器 X	2-13
2.2.5 輸出繼電器 Y	2-13
2.2.6 輔助繼電器 M	2-14
2.2.7 特殊輔助繼電器 SM	2-14
2.2.8 特殊輔助繼電器 SM 的更新時間	2-45
2.2.9 步進點繼電器 S	2-53

2.2.10 計時器 T	2-53
2.2.11 計數器 C	2-55
2.2.12 32 位計數器 HC	2-56
2.2.13 資料暫存器 D	2-58
2.2.14 特殊資料暫存器 SR	2-58
2.2.15 特殊資料暫存器 SR 的更新時間	2-93
2.2.16 SM/SR 補充說明	2-95
2.2.17 連結暫存器 L	2-110
2.2.18 間接指定暫存器 E	2-110

第 3 章 指令表

3.1 指令類型	3-2
3.1.1 基本指令	3-2
3.1.2 應用指令	3-2
3.2 指令表說明	3-3
3.2.1 基本指令	3-3
3.2.2 應用指令 (依 API 號碼排序)	3-4
3.2.3 應用指令 (依英文字母排序)	3-5
3.2.4 指令裝置表說明	3-6
3.3 基本指令一覽表	3-7
3.4 應用指令一覽表	3-9
3.4.1 應用指令 (依 API 號碼排序)	3-9
3.4.2 應用指令 (依英文字母排序)	3-42

第 4 章 指令結構

4.1 API 應用指令組成說明	4-2
4.2 指令使用限制	4-5
4.3 間接指定說明	4-7
4.4 指標暫存器 (PR) 說明	4-8
4.5 計時器指標暫存器 (TR) 說明	4-10
4.6 16 位元計數器指標暫存器 (CR) 說明	4-12
4.7 32 位元計數器指標暫存器 (HCR) 說明	4-13

第 5 章 基本指令	
5.1 基本指令一覽表	5-2
5.2 基本指令說明	5-3
第 6 章 指令表	
6.1 比較操作指令	6-4
6.1.1 比較操作指令一覽表	6-4
6.1.2 比較操作指令說明	6-7
6.2 四則運算指令說明	6-36
6.2.1 四則運算指令一覽表	6-36
6.2.2 四則運算指令說明	6-37
6.3 資料轉換指令說明	6-73
6.3.1 資料轉換指令一覽表	6-73
6.3.2 資料轉換指令說明	6-74
6.4 資料轉移指令說明	6-110
6.4.1 資料轉移指令一覽表	6-110
6.4.2 資料轉移指令說明	6-111
6.5 程式跳躍指令	6-134
6.5.1 程式跳躍指令一覽表	6-134
6.5.2 程式跳躍指令說明	6-135
6.6 程式執行控制指令	6-143
6.6.1 程式執行控制指令一覽表	6-143
6.6.2 程式執行控制指令說明	6-144
6.7 I/O 更新指令	6-151
6.7.1 I/O 更新指令一覽表	6-151
6.7.2 I/O 更新指令說明	6-152
6.8 便利指令	6-153
6.8.1 便利指令一覽表	6-153
6.8.2 便利指令說明	6-154
6.9 邏輯操作指令	6-189
6.9.1 邏輯操作指令一覽表	6-189
6.9.2 邏輯操作指令說明	6-190
6.10 旋轉指令	6-212

6.10.1 旋轉指令一覽表	6-212
6.10.2 旋轉指令說明	6-213
6.11 基本指令	6-223
6.11.1 基本指令一覽表	6-223
6.11.2 基本指令說明	6-224
6.12 移位指令	6-231
6.12.1 移位指令一覽表	6-231
6.12.2 移位指令說明	6-232
6.13 資料處理指令	6-257
6.13.1 資料處理指令一覽表	6-257
6.13.2 資料處理指令說明	6-258
6.14 結構建立指令	6-302
6.14.1 結構建立指令一覽表	6-302
6.14.2 結構建立指令說明	6-303
6.15 模組的資料讀寫指令	6-310
6.15.1 模組的資料讀寫指令一覽表	6-310
6.15.2 模組的資料讀寫指令說明	6-311
6.16 浮點數指令	6-316
6.16.1 浮點數指令一覽表	6-316
6.16.2 浮點數指令說明	6-317
6.17 萬年曆指令	6-355
6.17.1 萬年曆指令一覽表	6-355
6.17.2 萬年曆指令說明	6-356
6.18 週邊設備指令	6-379
6.18.1 週邊設備指令一覽表	6-379
6.18.2 週邊設備指令說明	6-380
6.19 通訊指令	6-394
6.19.1 通訊指令一覽表	6-394
6.19.2 通訊指令說明	6-395
6.20 其他指令	6-429
6.20.1 其他指令一覽表	6-429
6.20.2 其他指令說明	6-430
6.21 字串處理指令	6-439

6.21.1 字串處理指令一覽表	6-439
6.21.2 字串處理指令說明	6-440
6.22 乙太網控制指令	6-499
6.22.1 乙太網控制指令一覽表	6-499
6.22.2 乙太網控制指令說明	6-500
6.23 記憶卡讀寫指令	6-532
6.23.1 記憶卡讀寫指令一覽表	6-532
6.23.2 記憶卡讀寫指令說明	6-533
6.24 任務控制指令	6-545
6.24.1 任務控制指令一覽表	6-545
6.24.2 任務控制指令說明	6-546
6.25 SFC 控制指令	6-548
6.25.1 SFC 控制指令一覽表	6-548
6.25.2 SFC 控制指令說明	6-549
6.26 備援控制指令	6-555
6.26.1 備援控制指令一覽表	6-555
6.26.2 備援控制指令說明	6-556

第 7 章 錯誤代碼

7.1 模組錯誤代碼對應燈號及狀態說明	7-2
7.1.1 CPU 模組錯誤代碼對應燈號及狀態	7-2
7.1.2 類比模組與溫度模組錯誤代碼對應燈號	7-20
7.1.3 AH02HC-5A/AH04HC-5A 錯誤代碼對應燈號	7-21
7.1.4 AH05PM-5A/AH10PM-5A/AH15PM-5A 錯誤代碼對應燈號	7-22
7.1.5 AH20MC-5A 錯誤代碼對應燈號	7-23
7.1.6 AH10EN-5A/AH15EN-5A 錯誤代碼對應燈號	7-24
7.1.7 AH10SCM-5A/AH15SCM-5A 錯誤代碼對應燈號	7-24
7.1.8 AH10DNET-5A 錯誤代碼對應燈號	7-25
7.1.9 AH10PFBM-5A 錯誤代碼對應燈號	7-26
7.1.10 AH10PFBS-5A 錯誤代碼對應燈號	7-26
7.1.11 AH10COPM-5A 錯誤代碼對應燈號	7-27

1

第1章 簡介

目錄

1.1	一般概述	1-2
1.1.1	相關手冊.....	1-2
1.1.2	機種簡易說明	1-2
1.2	軟體簡述	1-9
1.2.1	編輯軟體.....	1-9
1.2.2	程式組織單元 (POU) 與工作 (TASK)	1-11

1

1.1 一般概述

本手冊描述 AH500 系列 PLC 主機程式設計相關的內容介紹，基本指令及應用指令操作方法說明。

1.1.1 相關手冊

AH500 系列 PLC 相關手冊的組成如下表所示。請搭配使用。

- AH500 快速入門手冊：
主要介紹如何讓使用者在詳細閱讀完所有手冊之前，能先快速的建立並使用本系統。
- AH500 程式手冊：
描述AH500系列PLC主機程式設計相關的內容介紹，基本指令及應用指令操作方法說明。
- ISPSOft 使用手冊：
有關ISPSOft操作、程式語言介紹 (Ladder、IL、SFC、FBD、ST)、POU概念及工作 (Task) 概念等。
- AH500 硬體手冊：
電氣規格、外觀及尺寸等。
- AH500 操作手冊：
CPU功能介紹、各項裝置、I/O配置及故障排除等。
- AH500 模組手冊：
特殊模組使用介紹，例如網路模組、類比模組及溫度模組等。
- AH500 運動控制模組手冊：
主要介紹運動控制模組的規格、配線、指令及功能範例。
- PMSOft 使用手冊：
主要介紹運動控制模組編輯軟體的詳細使用，包含編輯模式、連線及加密設定等內容。
- AH500 備援系統操作手冊：
描述AH500備援系統架構、建立、程式設計及操作說明。

1.1.2 機種簡易說明

分類	機種名稱	說明
電源模組	AHPS05-5A	100-240VAC 50/60Hz 電源模組
	AHPS15-5A	24VDC 電源模組
CPU 模組	AHCPU500-RS2	基本型 CPU 模組，內建 RS-485*2、USB 通訊埠以及 SD 卡介面，支援 768 點 I/O，程式容量 32k steps。

分類	機種名稱	說明
CPU 模組	AHCPU500-EN	基本型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 768 點 I/O，程式容量 32k steps。
	AHCPU501-RS2	進階型 CPU 模組，內建 RS-485 * 2、USB 通訊埠以及 SD 卡介面，支援 768 點 I/O，程式容量 48k steps。
	AHCPU501-EN	進階型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 768 點 I/O，程式容量 48k steps。
	AHCPU510-RS2	基本型 CPU 模組，內建 RS-485*2、USB 通訊埠以及 SD 卡介面，支援 1280 點 I/O，程式容量 64k steps。
	AHCPU510-EN	基本型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 1280 點 I/O，程式容量 64k steps。
	AHCPU511-RS2	進階型 CPU 模組，內建 RS-485 * 2、USB 通訊埠以及 SD 卡介面，支援 1280 點 I/O，程式容量 96k steps。
	AHCPU511-EN	進階型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 1280 點 I/O，程式容量 96k steps。
	AHCPU520-RS2	基本型 CPU 模組，內建 RS-485*2、USB 通訊埠以及 SD 卡介面，支援 2304 點 I/O，程式容量 128k steps。
	AHCPU520-EN	基本型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 2304 點 I/O，程式容量 128k steps。
	AHCPU521-RS2	進階型 CPU 模組，內建 RS-485 * 2、USB 通訊埠以及 SD 卡介面，支援 2304 點 I/O，程式容量 192k steps。
	AHCPU521-EN	進階型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 2304 點 I/O，程式容量 192k steps。
	AHCPU530-RS2	基本型 CPU 模組，內建 RS-485*2、USB 通訊埠以及 SD 卡介面，支援 4352 點 I/O，程式容量 256k steps。
	AHCPU530-EN	基本型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 4352 點 I/O，程式容量 256k steps。
	AHCPU531-RS2	進階型 CPU 模組，內建 RS-485 * 2、USB 通訊埠以及 SD 卡介面，支援 4352 點 I/O，程式容量 384k steps。
	AHCPU531-EN	進階型 CPU 模組，內建乙太網、RS-485、USB 通訊埠以及 SD 卡介面，支援 4352 點 I/O，程式容量 384k steps。
主背板	AHBP04M1-5A	CPU/RTU 專用 4 槽主背板
	AHBP06M1-5A	CPU/RTU 專用 6 槽主背板
	AHBP08M1-5A	CPU/RTU 專用 8 槽主背板
	AHBP12M1-5A	CPU/RTU 專用 12 槽主背板

1

分類	機種名稱	說明
延伸背板	AHBP06E1-5A	CPU/RTU 專用 6 槽延伸背板
	AHBP08E1-5A	CPU/RTU 專用 8 槽延伸背板
數位 I/O 模組	AH16AM10N-5A	24VDC · 5mA · 16 點輸入 · 端子台
	AH32AM10N-5A	24VDC · 5mA · 32 點輸入 · 端子台
	AH32AM10N-5B	24VDC · 5mA · 32 點輸入 · DB37 連接器
	AH32AM10N-5C	24VDC · 5mA · 32 點輸入 · 牛角座連接器
	AH64AM10N-5C	24VDC · 3.2mA · 64 點輸入 · 牛角座連接器
	AH16AM30N-5A	100 ~ 240VAC · 4.5mA ~ 9mA (100V · 50Hz) · 16 點輸入 · 端子台
	AH16AR10N-5A	24VDC · 5mA · 16 點輸入 · 端子台 (具備快速中斷功能)
	AH16AN01R-5A	240VAC/24VDC · 2A · 16 輸出點 · 繼電器 · 端子台
	AH16AN01T-5A	12 ~ 24VDC · 0.5A · 16 輸出點 · 漏型輸出 · 端子台
	AH16AN01P-5A	12 ~ 24VDC · 0.5A · 16 輸出點 · 源型輸出 · 端子台
	AH32AN02T-5A	12 ~ 24VDC · 0.1A · 32 輸出點 · 漏型輸出 · 端子台
	AH32AN02T-5B	12 ~ 24VDC · 0.1A · 32 輸出點 · 漏型輸出 · DB37 連接器
	AH32AN02T-5C	12 ~ 24VDC · 0.1A · 32 輸出點 · 漏型輸出 · 牛角座連接器
	AH32AN02P-5A	12 ~ 24VDC · 0.1A · 32 輸出點 · 源型輸出 · 端子台
	AH32AN02P-5B	12 ~ 24VDC · 0.1A · 32 輸出點 · 源型輸出 · DB37 連接器
	AH32AN02P-5C	12 ~ 24VDC · 0.1A · 32 輸出點 · 源型輸出 · 牛角座連接器
	AH64AN02T-5C	12 ~ 24VDC · 0.1A · 64 點輸出 · 漏型輸出 · 牛角座連接器
	AH64AN02P-5C	12 ~ 24VDC · 0.1A · 64 點輸出 · 源型輸出 · 牛角座連接器
	AH16AN01S-5A	100 ~ 240VAC · 0.5A · 16 輸出點 · TRIAC · 端子台
	AH16AP11R-5A	24VDC · 5mA · 8 輸入點 · 240VAC/24VDC · 2A · 8 輸出點 · 繼電器 · 端子台
AH16AP11T-5A	24VDC · 5mA · 8 輸入點 · 12 ~ 24VDC · 0.5A · 8 輸出點 · 漏型輸出 · 端子台	
AH16AP11P-5A	24VDC · 5mA · 8 輸入點 · 12 ~ 24VDC · 0.5A · 8 輸出點 · 源型輸出 · 端子台	

分類	機種名稱	說明
類比 I/O 模組	AH04AD-5A	4 通道類比信號輸入 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA · -20mA~20mA 轉換時間：150us/通道
	AH08AD-5A	8 通道類比信號輸入 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA · -20mA~20mA 轉換時間：150us/通道
	AH08AD-5B	8 通道類比信號輸入 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 轉換時間：150us/通道
	AH08AD-5C	8 通道類比信號輸入 硬體解析度：16 位元 0/4mA~20mA · -20mA~20mA 轉換時間：150us/通道
	AH04DA-5A	4 通道類比信號輸出 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA 轉換時間：150us/通道
	AH08DA-5A	8 通道類比信號輸出 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA 轉換時間：150us/通道
	AH08DA-5B	8 通道類比信號輸出 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 轉換時間：150us/通道

1

分類	機種名稱	說明
類比 I/O 模組	AH08DA-5C	8 通道類比信號輸出 硬體解析度：16 位元 0/4mA~20mA 轉換時間：150us/通道
	AH06XA-5A	4 通道類比信號輸入 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA · -20mA~20mA 轉換時間：150us/通道 2 通道類比信號輸出 硬體解析度：16 位元 0/1V~5V · -5V~5V · 0V~10V · -10V~10V 0/4mA~20mA 轉換時間：150us/通道
溫度模組	AH04PT-5A	4 通道 4 線式或 3 線式 RTD 溫度感測 感測器型式：Pt100 · Pt1000 · Ni100 · Ni1000 · 或 0~300Ω 解析度：0.1°C/0.1°F (16 位元) 轉換時間：4 線式轉換時間:150ms/通道 · 3 線式轉換時間： 300ms/通道。
	AH08PTG-5A	8 通道 4 線式/3 線式/2 線式 RTD 溫度感測 感測器形式：Pt100、Pt1000、Ni100、Ni1000、或 0~300Ω · 解析度：0.1°C /0.1°F (16 位元) 轉換時間：20ms/4 通道 · 200ms/8 通道。
	AH04TC-5A	4 通道熱電耦溫度感測 感測器型：J · K · R · S · T · E · N 或 -150~+150mV 解析度：0.1°C/0.1°F 轉換時間：200ms/通道
	AH08TC-5A	8 通道熱電耦溫度感測 感測器型：J · K · R · S · T · E · N 或 -150~+150mV 解析度：0.1°C/0.1°F 轉換時間：200ms/通道
運動控制模組	AH02HC-5A	2 通道高速計數器模組 (200kHz)
	AH04HC-5A	4 通道高速計數器模組 (200kHz)
	AH05PM-5A	2 軸脈波型運動控制模組 (1MHz)

分類	機種名稱	說明
運動控制模組	AH10PM-5A	6 軸脈波型運動控制模組 (4 軸 1MHz / 2 軸 200kHz)
	AH15PM-5A	4 軸脈波型運動控制模組 (1MHz)
	AH20MC-5A	12 軸 DMCNET (Delta Motion Control Network) 運動控制模組 (10Mbps)
網路模組	AH10EN-5A	乙太網路通訊模組，可以做主站或者從站。 內建兩個乙太網接口，支援 MODBUS TCP 主站。 支援 EtherNet/IP (V2.0)
	AH15EN-5A	乙太網路通訊模組，可以做主站或者從站。 內建兩個乙太網接口，支援 MODBUS TCP 主站。 支援 IEC60870-5-104。
	AH10SCM-5A	串列通訊模組，內建兩組 RS-485/422 接口，電源及通訊全隔離，支援 MODBUS 與 UD Link 協定。
	AH15SCM-5A	串列通訊模組，內建兩組 RS-232 接口，電源及通訊全隔離，支援 MODBUS 與 UD Link 協定
	AH10DNET-5A	DeviceNet 通訊模組，可以做主站或者從站，通訊速率最大可達 1Mbps。
	AH10PFBM-5A	PROFIBUS 主站通訊模組
	AH10PFBS-5A	PROFIBUS 從站通訊模組
	AH10COPM-5A	CANopen 通訊模組，可以做主站或者從站。
遠程 I/O 通訊 模組	AHRTU-DNET-5A	DeviceNet 遠程 I/O 通訊模組
	AHRTU-PFBS-5A	PROFIBUS 遠程 I/O 通訊模組
	AHRTU-ETHN-5A	Ethernet 遠程 I/O 通訊模組
延伸通訊線	AHACAB06-5A	延伸背板專用 0.6m 延伸通訊線
	AHACAB10-5A	延伸背板專用 1.0m 延伸通訊線
	AHACAB15-5A	延伸背板專用 1.5m 延伸通訊線
	AHACAB30-5A	延伸背板專用 3.0m 延伸通訊線
	AHAADP01/02EF-5A	延伸背板專用光纖模組
I/O 連接線	UC-ET010-24A	1.0m I/O 連接線 (牛角座)，適用 AH32AM10N-5C 及 AH64AM10N-5C。
	UC-ET010-24C	1.0m I/O 連接線 (牛角座)，適用 AH32AN02T-5C、AH32AN02P-5C、AH64AN02T-5C 及 AH64AN02P-5C。

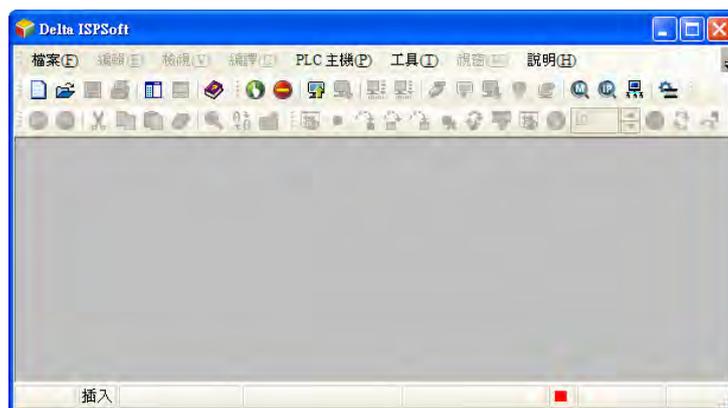
1

分類	機種名稱	說明
I/O 連接線	UC-ET010-33B	1.0m I/O 連接線 (DB37) · 適用 AH32AM10N-5B 、 AH32AN02T-5B 及 AH32AN02P-5B 。
	UC-ET010-13B	1.0m I/O 連接線 · 適用 AH04HC-5A 及 AH20MC-5A 。
	UC-ET010-15B	1.0m I/O 連接線 · 適用 AH10PM-5A 及 AH15PM-5A 。
配線模組	UB-10-ID32A	I/O 配線模組 (32 點輸入) · 適用 AH32AM10N-5C 及 AH64AM10N-5C 。
	UB-10-ID32B	I/O 配線模組 (16 點繼電器輸出) · 適用 AH32AN02T-5C 及 AH64AN02T-5C 。
	UB-10-OR16A	I/O 配線模組 (16 點繼電器輸出) · 適用 AH32AN02P-5C 及 AH64AN02P-5C 。
	UB-10-OR16B	I/O 配線模組 (32 點輸入) · 適用 AH32AM10N-5B 。
	UB-10-OR32A	I/O 配線模組 (32 點繼電器輸出) · 適用 AH32AN02T-5B 。
	UB-10-OR32B	I/O 配線模組 (32 點繼電器輸出) · 適用 AH32AN02P-5B 。
	UB-10-OT32A	I/O 配線模組 (32 點電晶體輸出) · 適用 AH32AN02T-5C 、 AH32AN02P-5C 、 AH64AN02T-5C 及 AH64AN02P-5C 。
	UB-10-OT32B	I/O 配線模組 (32 點電晶體輸出) · 適用 AH32AN02T-5B 及 AH32AN02P-5B 。
	UB-10-IO16C	I/O 配線模組 · 適用 AH04HC-5A 及 AH20MC-5A 。
	UB-10-IO24C	I/O 配線模組 · 適用 AH10PM-5A 。
	UB-10-IO34C	I/O 配線模組 · 適用 AH15PM-5A 。
防護模組	AHASP01-5A	未使用插槽之專用防護模組

1.2 軟體簡述

1.2.1 編輯軟體

編輯軟體 ISPSOft 的概廓：



- 5 種編程語言：包括指令列表 (IL)、結構化語言 (ST)、階梯圖 (LD)、循序功能圖 (SFC) 及功能區塊圖 (FBD)。



1

- 使用變數的觀念，讓使用者可自行定義變數符號來取代 PLC 的裝置名稱，除了大大地提高程式的可讀性之外，更省去了使用者費心配置裝置位址的時間。

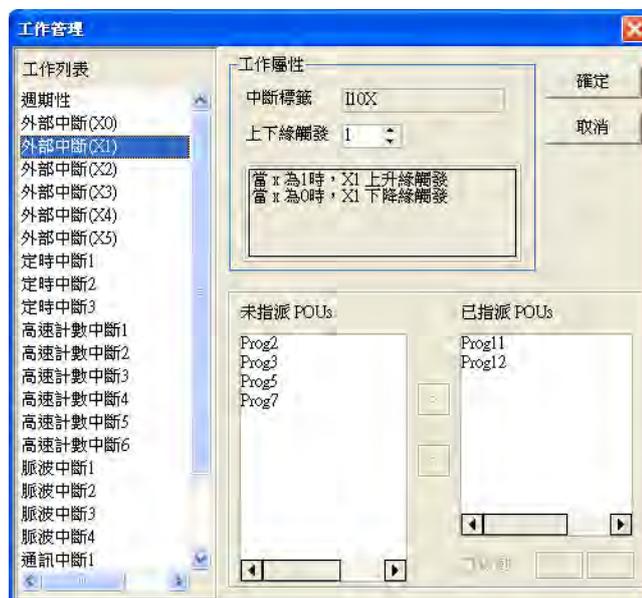


區域符號					
類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	IN_0	N/A [Auto]	BOOL	FALSE	
VAR	IN_1	N/A [Auto]	BOOL	FALSE	
VAR	IN_2	N/A [Auto]	BOOL	FALSE	
VAR	OUT_0	N/A [Auto]	BOOL	FALSE	
VAR	OUT_1	N/A [Auto]	BOOL	FALSE	

- 導入程式組織單元 (POU) 的架構，除了可將原本的主程式切割成若干個程序單元之外，並以函式 (Function) 與功能塊 (Function Block) 來取代傳統的副程式，讓程式的架構可以更模組化也更便於管理。



- 使用工作 (Task) 的觀念來管理程式的執行順序，將程式開發的工作提升至專案管理的層次，讓大規模的程式開發變的更便於管理。



1.2.2 程式組織單元 (POU) 與工作 (TASK)

程式組織單元 (Program Organization Unit · 簡稱 POU) 是建構 PLC 程式的基本元素，有別於傳統的 PLC 程式，由 IEC61131-3 所導入的程式架構，其特色在於將原本一大段的程式分割成若干個小單元，而這些小單元便稱之為 POU。POU 大致可分為以下三種。

1. 程式 (PROG): 程式 POU 於 PLC 程式中扮演著主要程序的角色，其執行方式可由設計者自行定義為週期掃描或中斷模式，並可在工作 (TASK) 清單中安排各個程式 POU 的掃描順序。
2. 功能塊 (FB): 功能塊 (Function Block · 簡稱 FB) 本身所代表的意義類似於副程式，而在功能塊內部所定義的程序，則必須要在該功能塊被程式 POU 所呼叫，並輸入相關的參數之後才會執行。
3. 函式 (FC): 函式 (Function · 簡稱 FC) 本身所代表的意義較接近於巨集指令，亦即我們可將多筆的運算指令或功能寫在函式 POU 當中，之後再於程式 POU 或功能塊 POU 當中呼叫使用即可。

工作 (TASK)，是指規定使各個程式按照何種順序或中斷條件執行的功能。工作 (TASK) 的意義在於賦予各個程式 POU 一個明確的執行任務，並指定每個程式 POU 之間的執行順序或是啟動的方式。基本上一個專案中的程式 POU 並非都會參與實際的執行，而是必須在指派 TASK 之後才可確定該 POU 是否執行以及如何執行；當 POU 未被指派 TASK 時，該 POU 僅會被當做一般的原始碼而與專案一起儲存，本身並不會被編譯為 PLC 的執行碼。此外，僅有程式 POU 需要被指派 TASK，至於功能塊 (FB) 或函式 (FC) 的執行方式，則是依據呼叫它們的上位 POU 程式而定。TASK 的運行方式大致可分為週期性、定時中斷，以及條件中斷 3 種

1

1. 週期性：被指派至週期性 TASK 的程式 POU，其執行方式便是單純的來回掃描，按照順序執行。
2. 定時中斷工作 (TASK)：被指派至定時中斷工作的程式 POU，當定時中斷工作的時間到達後，被分配至該定時中斷工作的所有程式 POU 便會依照排列順序執行。
3. 條件中斷工作 (TASK)：條件中斷亦可區分為多種類型，如外部中斷、I/O 中斷...等，規劃前必須先確認所選用的專案機種支援哪些中斷模式。當程式 POU 被分配至條件中斷 TASK 時，其功能便類似中斷副程式，當中斷條件成立時，如外部觸發中斷的某個接點被觸發，被分配至該 TASK 的所有 POU 便會依照排列順序執行一遍。

2

第2章 裝置說明

目錄

2.1	裝置簡介	2-2
2.1.1	裝置列表.....	2-2
2.1.1.1	AH500 基本型 CPU 模組 (AHCPU500/510/520/530)	2-2
2.1.1.2	AH500 進階型 CPU 模組 (AHCPU501/511/521/531)	2-3
2.1.2	I/O 儲存區的基本結構	2-5
2.1.3	停電保持記憶方式	2-6
2.1.4	停電保持區的裝置範圍	2-6
2.2	裝置功能說明.....	2-7
2.2.1	數值、常數.....	2-7
2.2.2	浮點數	2-9
2.2.2.1	單精度浮點數 (32 位元浮點數)	2-9
2.2.2.2	雙精度浮點數 (64 位元浮點數)	2-10
2.2.2.3	十進浮點數	2-11
2.2.3	字串	2-11
2.2.4	輸入繼電器 X	2-13
2.2.5	輸出繼電器 Y	2-13
2.2.6	輔助繼電器 M.....	2-14
2.2.7	特殊輔助繼電器 SM	2-14
2.2.8	特殊輔助繼電器 SM 的更新時間.....	2-45
2.2.9	步進點繼電器 S	2-53
2.2.10	計時器 T	2-53
2.2.11	計數器 C.....	2-55
2.2.12	32 位計數器 HC.....	2-56
2.2.13	資料暫存器 D	2-58
2.2.14	特殊資料暫存器 SR	2-58
2.2.15	特殊資料暫存器 SR 的更新時間	2-93
2.2.16	SM/SR 補充說明.....	2-95
2.2.17	連結暫存器 L.....	2-110
2.2.18	間接指定暫存器 E	2-110

2.1 裝置簡介

本章節針對 PLC 所處理的數值、字串和輸入、輸出、輔助繼電器、計時器、計數器及資料暫存器等各種裝置的配置和功能做說明。

2

2.1.1 裝置列表

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

形式	裝置名稱		裝置數	範圍
位元 裝置	輸入繼電器	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
	輸出繼電器	Y	1024 (AHCPU500)	Y0.0~Y63.15
			2048 (AHCPU510)	Y0.0~Y127.15
			4096 (AHCPU520)	Y0.0~Y255.15
			8192 (AHCPU530)	Y0.0~Y511.15
	資料暫存器	D	262144 (AHCPU500)	D0.0~D16383.15
			524288 (AHCPU510)	D0.0~D32767.15
			1048576 (AHCPU520/530)	D0.0~D65535.15
	連結暫存器	L	262144 (AHCPU500)	L0.0~L16383.15
			524288 (AHCPU510)	L0.0~L32767.15
			1048576 (AHCPU520/530)	L0.0~L65535.15
	輔助繼電器	M	8192	M0~M8191
	特殊輔助旗標	SM	2048	SM0~SM2047
步進點繼電器	S	2048	S0~S2047	
計時器	T	2048	T0~T2047	
計數器	C	2048	C0~C2047	
32 位元計數器	HC	64	HC0~HC63	
字元 裝置	輸入繼電器	X	64 (AHCPU500)	X0~X63
			128 (AHCPU510)	X0~X127
			256 (AHCPU520)	X0~X255
			512 (AHCPU530)	X0~X511
	輸出繼電器	Y	64 (AHCPU500)	Y0~Y63
			128 (AHCPU510)	Y0~Y127
			256 (AHCPU520)	Y0~Y255
			512 (AHCPU530)	Y0~Y511

形式	裝置名稱	裝置數	範圍		
字元裝置	資料暫存器	D	16384 (AHCPU500) 32768 (AHCPU510) 65536 (AHCPU520/530)	D0~D16383 D0~D32767 D0~D65535	
		特殊資料暫存器	SR	2048	SR0~SR2047
		連結暫存器	L	16384 (AHCPU500) 32768 (AHCPU510) 65536 (AHCPU520/530)	L0~L16383 L0~L32767 L0~L65535
	計時器		T	2048	T0~T2047
	計數器		C	2048	C0~C2047
	32 位元計數器	HC	64 (128 words)	HC0~HC63	
	間接指定暫存器	E	32	E0~E31	
	常數*	十進制	K	16 位元運算：-32768~32767 32 位元運算：-2147483648~2147483647	
十六進制		16#	16 位元運算：16#0~16#FFFF 32 位元運算：16#0~16#FFFFFFFF		
單精度浮點數		F	32位元運算：±1.17549435 ⁻³⁸ ~±3.40282347 ⁺³⁸		
雙精度浮點數		DF	64 位元運算：±2.2250738585072014 ⁻³⁰⁸ ~±1.7976931348623157 ⁺³⁰⁸		
字串*	字串	“\$”	1~31 個字		

*1：十進制的表示方式，在第5、6章的指令裝置表中以K來表示，但在ISPSoft中直接輸入值，例如K50，請直接輸入50。

*2：浮點數的表示方式，在第5、6章的指令裝置表中以F/DF來表示，但在ISPSoft中是直接以小數點的方式來表示，例如要輸入F500的浮點數，請直接輸入500.0。

*3：字串的表示方式，在第5、6章的指令裝置表中以“\$”來表示，但在ISPSoft中是以“”方式來表示，例如要輸入字串1234，請直接輸入“1234”。

2.1.1.2 AH500 進階型 CPU 模組 (AHCPU501/511/521/531)

形式	裝置名稱	裝置數	範圍		
位元裝置	輸入繼電器	X	2048 (AHCPU501) 4096 (AHCPU511) 8192 (AHCPU521) 16384 (AHCPU531)	X0.0~X127.15 X0.0~X255.15 X0.0~X511.15 X0.0~X1023.15	
		輸出繼電器	Y	2048 (AHCPU501) 4096 (AHCPU511) 8192 (AHCPU521) 16384 (AHCPU531)	Y0.0~Y127.15 Y0.0~Y255.15 Y0.0~Y511.15 Y0.0~Y1023.15

2

形式	裝置名稱		裝置數	範圍
位元 裝置	資料暫存器	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
	連結暫存器	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
	輔助繼電器	M	8192	M0~M8191
	特殊輔助旗標	SM	4096	SM0~SM4095
	步進點繼電器	S	2048	S0~S2047
	計時器	T	2048	T0~T2047
	計數器	C	2048	C0~C2047
32 位元計數器	HC	64	HC0~HC63	
字元 裝置	輸入繼電器	X	128 (AHCPU501)	X0~X127
			256 (AHCPU511)	X0~X255
			512 (AHCPU521)	X0~X511
			1024 (AHCPU531)	X0~X1023
	輸出繼電器	Y	128 (AHCPU501)	Y0~Y127
			256 (AHCPU511)	Y0~Y255
			512 (AHCPU521)	Y0~Y511
			1024 (AHCPU531)	Y0~Y1023
	資料暫存器	D	24576 (AHCPU501)	D0~D24575
			49152 (AHCPU511)	D0~D49151
			98304 (AHCPU521)	D0~D98303
			131072 (AHCPU531)	D0~D131071
	特殊資料暫存器	SR	4096	SR0~SR4095
	連結暫存器	L	24576 (AHCPU501)	L0~L24575
			49152 (AHCPU511)	L0~L49151
			98304 (AHCPU521)	L0~L98303
			131072 (AHCPU531)	L0~L131071
計時器	T	2048	T0~T2047	
計數器	C	2048	C0~C2047	
32 位元計數器	HC	64 (128 words)	HC0~HC63	
間接指定暫存器	E	32	E0~E31	

形式	裝置名稱		裝置數	範圍
常數*	十進制	K	16 位元運算：-32768~32767 32 位元運算：-2147483648~2147483647	
	十六進制	16#	16 位元運算：16#0~16#FFFF 32 位元運算：16#0~16#FFFFFFFF	
常數*	單精度浮點數	F	32位元運算：±1.17549435 ⁻³⁸ ~±3.40282347 ⁺³⁸	
	雙精度浮點數	DF	64 位元運算：±2.2250738585072014 ⁻³⁰⁸ ~ ±1.7976931348623157 ⁺³⁰⁸	
字串*	字串	"\$"	1~31 個字	

*1：十進制的表示方式，在第5、6章的指令裝置表中以K來表示，但在ISPSoft中直接輸入值，例如K50，請直接輸入50。

*2：浮點數的表示方式，在第5、6章的指令裝置表中以F/DF來表示，但在ISPSoft中是直接以小數點的方式來表示，例如要輸入F500的浮點數，請直接輸入500.0。

*3：字串的表示方式，在第5、6章的指令裝置表中以"\$"來表示，但在ISPSoft中是以" "方式來表示，例如要輸入字串1234，請直接輸入"1234"。

2.1.2 I/O 儲存區的基本結構

裝置	功能	位元存取	字元存取	ISPSoft 修改	強制位元狀態
X	輸入繼電器	OK	OK	OK	OK
Y	輸出繼電器	OK	OK	OK	OK
M	輔助繼電器	OK	-	OK	-
SM	特殊輔助旗標	OK	-	OK	-
S	步進點繼電器	OK	-	OK	-
T	計時器	OK	OK	OK	-
C	計數器	OK	OK	OK	-
HC	32 位計數器	OK	OK	OK	-
D	資料暫存器	OK	OK	OK	-
SR	特殊資料暫存器	-	OK	OK	-
L	連結暫存器	OK	OK	OK	-
E	間接指定暫存器	-	OK	OK	-

2.1.3 停電保持記憶方式

PLC 動作		記憶體類型	非停電保持區	停電保持區	Y 裝置
電源 OFF=>ON			清除	保持	清除
STOP=> RUN	設定 Y 裝置清除		保持	保持	清除
	設定 Y 裝置保持		保持	保持	保持
	設定 Y 裝置回復 STOP 前狀態		保持	保持	回復 STOP 前狀態
STOP=> RUN	設定非停電保持區 清除		清除	保持	參照 Y 裝置之設定
	設定非停電保持區 保持		保持	保持	參照 Y 裝置之設定
RUN=>STOP			保持	保持	保持
SM204ON (清除所有的非停電保持區域)			清除	保持	清除
SM205ON (清除所有停電保持區域)			保持	清除	保持
出廠設定值			0	0	0

2.1.4 停電保持區的裝置範圍

裝置	功能	停電保持區範圍
X	輸入繼電器	固定非停電保持
Y	輸出繼電器	固定非停電保持
M*	輔助繼電器	預設 M0~M8191
SM	特殊輔助繼電器	部分停電保持並且不能被改變 詳細內容請參考 SM 功能表
S	步進點繼電器	固定非停電保持
T*	計時器	預設 T0~T2047
C*	計數器	預設 C0~C2047
HC*	32 位計數器	預設 HC0~HC63
D*	資料暫存器	AH500-EN/RS2 : 預設 D0~D16383 AH501-EN/RS2 : 預設 D0~D24575
		其它機種預設 D0~D32767 , 最多可以設定 32768 個

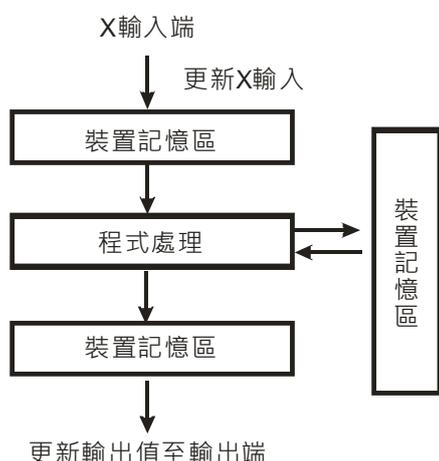
裝置	功能	停電保持區範圍
SR	特殊資料暫存器	部分停電保持並且不能被改變 詳細內容請參考 SR 功能表
L	連結暫存器	固定非停電保持
E	間接指定暫存器	固定非停電保持

*：表可設定停電保持區的範圍，可以設定此裝置不停電保持，而設定之範圍最大不能超過裝置範圍，其中 D 裝置最多只能設定 32768 個 D 裝置，例如：可以設定 D50~D32817 為停電保持區或設定 D32768~D65535 為停電保持區，而其預設為 D0~D32767 為停電保持區。

2

2.2 裝置功能說明

PLC對於程式的處理流程（結束再生方式）



- 更新輸入信號：
 1. PLC 在執行程式之前會將外部輸入信號狀態讀入至輸入信號記憶區內。
 2. 在程式執行中若輸入信號作 ON/OFF 變化，但是輸入信號記憶區內的狀態不會改變，一直到下一次掃描開始才會再更新輸入信號。
- 程式處理：

PLC 更新輸入信號後，開始從程式的起始位址依序執行程式中的每一指令，其處理結果存入各裝置記憶區。
- 更新輸出狀態：

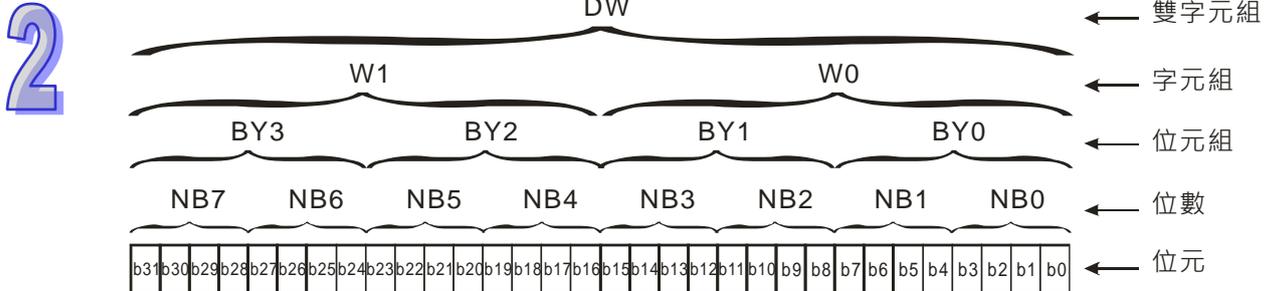
當執行到 END 指令後將裝置記憶區內的狀態送到使用者所分配的輸出端。

2.2.1 數值、常數

名稱	說明
位元 (Bit)	位元為二進制數值之最基本單位，其狀態非 1 即 0
位數 (Nibble)	由連續的 4 個位元所組成 (如 b3 ~ b0) 可用以表示一個位數之十進制數字 0 ~ 9 或十六進制之 0 ~ F
位元組 (Byte)	是由連續之兩個位數所組成 (亦即 8 位元，b7 ~ b0)。可表示十六進制之 00 ~ FF
字元組 (Word)	是由連續之兩個位元組所組成 (亦即 16 位元，b15 ~ b0) 可表示十六進制之 4 個位數值 0000 ~ FFFF
雙字元組 (Double Word)	是由連續之兩個字元組所組成 (亦即 32 位元，b31 ~ b0)，可表示十六進制之 8 個位數值 00000000~FFFFFFFF

名稱	說明
四字元組	是由連續之四個字元組所組成 (亦即 64 位元 · b63~b0) · 可表示十六進制之 16 個位數值 0000000000000000 - FFFFFFFF

二進制系統中位元、位數、位元組、字元組、及雙字元組的關係如下圖所示：



PLC 內部依據各種不同控制目的，共使用 4 種數值類型執行運算的工作，各種數值的任務及功能如下說明。

1. 二進制 (Binary Number · BIN)

PLC 內部之數值運算或儲存均採用二進制

2. 十進制 (Decimal Number · DEC)

十進制在 PLC 應用的時機如下：

- 作為計時器 T、計數器 C/HC 等的設定值，例：TMR C0 50。(K 常數)
- S、M、T、C、E...等裝置的編號，例：M10、T30。(裝置編號)
- X、Y、D...等裝置小數點前後的編號，例：X0.0、Y0.11、D10.0。(裝置編號)
- Constant K：在應用指令中作為運算元使用，例：MOV 123 D0。(K 常數)

3. BCD (Binary Code Decimal · BCD)

以一個位數或 4 個位元來表示一個十進制的資料，故連續的 16 個位元可以表示 4 位數的十進制數值資料。

4. 十六進制 (Hexadecimal Number · HEX)

十六進制在 PLC 應用的時機如下：

- Constant 16#：在應用指令中作為運算元使用，例：MOV 16#1A2B D0。(十六進制常數)

數值對照表：

二進制 (BIN)	十進制 (DEC)	BCD (Binary Code Decimal)	十六進制 (HEX)
PLC 內部運算用	常數 K · 裝置編號	BCD 相關指令	常數 16# · 裝置編號
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
0111	7	0111	7
1000	8	1000	8

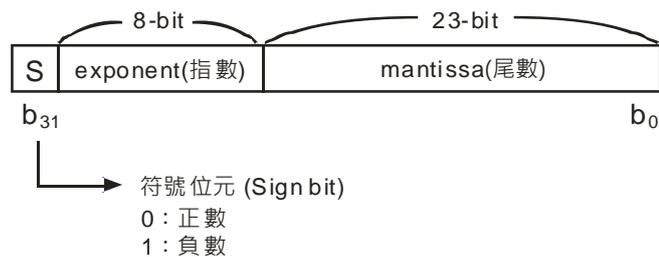
二進制 (BIN)	十進制 (DEC)	BCD (Binary Code Decimal)	十六進制 (HEX)
1001	9	1001	9
1010	10	-	A
1011	11	-	B
1100	12	-	C
1101	13	-	D
1110	14	-	E
1111	15	-	F
10000	16	0001 0000	10
10001	17	0001 0001	11

2.2.2 浮點數

浮點數的表示方式在 ISPSOft 中是以小數點的方式來表示，例如要輸入 500 的浮點數，必須輸入 500.0。

2.2.2.1 單精度浮點數 (32 位元浮點數)

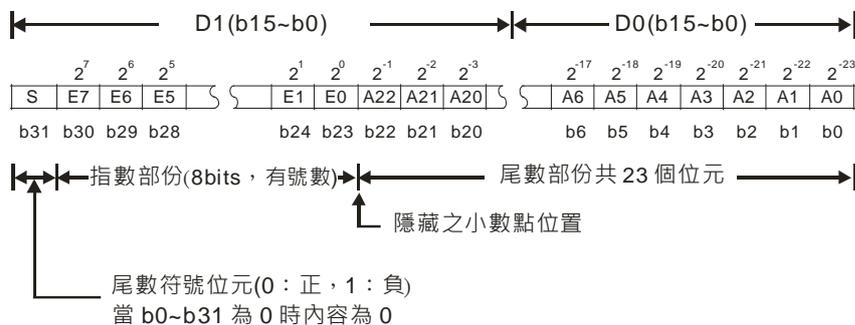
以 32 位元的暫存器長度表示浮點數，而表示法系採用 IEEE754 的標準，格式如下：



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

因此單精度浮點數的數目範圍為 $\pm 2^{-126}$ 到 $\pm 2^{+128}$ 相當於 $\pm 1.1755 \times 10^{-38}$ 到 $\pm 3.4028 \times 10^{+38}$ 。

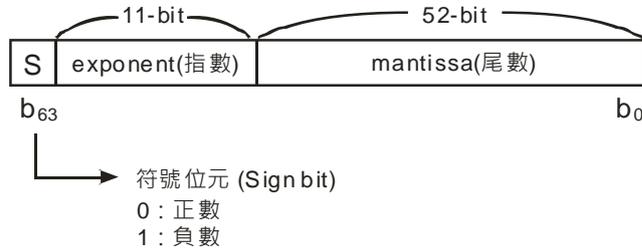
AH500 使用 2 個連續號碼的暫存器組成 32 位元的浮點數，我們以暫存器 (D1 , D0) 來說明，如下所示：



2.2.2.2 雙精度浮點數 (64 位元浮點數)

以 64 位元的暫存器長度表示浮點數，而表示法系採用 IEEE754 的標準，格式如下：

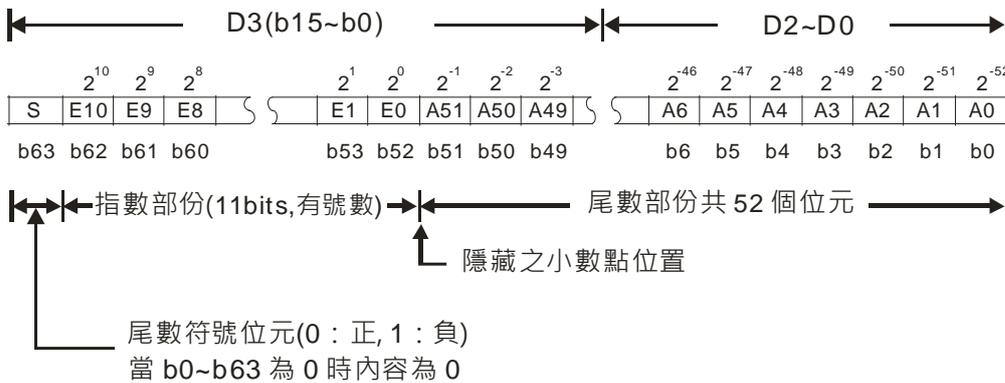
2



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

因此雙精度浮點數的數目範圍為 $\pm 2^{-1022}$ 到 $\pm 2^{+1024}$ 相當於 $\pm 2.2250 \times 10^{-308}$ 到 $\pm 1.7976 \times 10^{+308}$ 。

AH500 使用 4 個連續號碼的暫存器組成 64 位元浮點數，我們以暫存器 (D3 · D2 · D1 · D0) 來說明，如下所示：



範例一：

以單精度浮點數表示 23

步驟一：將 23 轉換成二進制數字：23.0=10111

步驟二：將二進制數字正規化：10111=1.0111 X2⁴，其中 0111 為尾數，4 為指數。

步驟三：求出指數部份的儲存值

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

步驟四：組合符號位元，指數，尾數成為浮點數。

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

以雙精度浮點數表示 23

步驟一：將 23 轉換成二進制數字：23.0=10111

步驟二：將二進制數字正規化：10111=1.0111 X2⁴，其中 0111 為尾數，4 為指數。

步驟三：求出指數部份的儲存值

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

步驟四：組合符號位元，指數，尾數成為浮點數。

$0\ 1000000011\ 011100_2$
 $=4037000000000000_{16}$

範例二：

以單精度浮點數表示-23.0

-23.0 浮點格式與 23.0 的轉換步驟完全相同，只需將符號位元改為 1 即可。

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

以雙精度浮點數表示-23.0

-23.0 浮點格式與 23.0 的轉換步驟完全相同，只需將符號位元改為 1 即可。

$1\ 1000000011\ 011100_2$

$= C037000000000000_{16}$

2.2.2.3 十進浮點數

- ◆ 單精度浮點數跟雙精度浮點數的內容比較無法被人所接受，因此，單精度浮點數跟雙精度浮點數可轉換成十進浮點數來供人作判斷。但是 PLC 對小數點的運算仍舊是使用單精度浮點數跟雙精度浮點數。
- ◆ 32 位元十進浮點數是使用 2 個連續號碼的暫存器來表現，較小編號的暫存器號碼存放常數部份、較大編號的暫存器號碼存放指數部份。

就以暫存器 (D1、D0) 來存放一個十進浮點數為例，如下所示。

$$\text{十進浮點數} = [\text{常數 D0}] \times 10^{[\text{指數 D1}]}$$

底數 D0 = $\pm 1.000 \sim \pm 9.999$

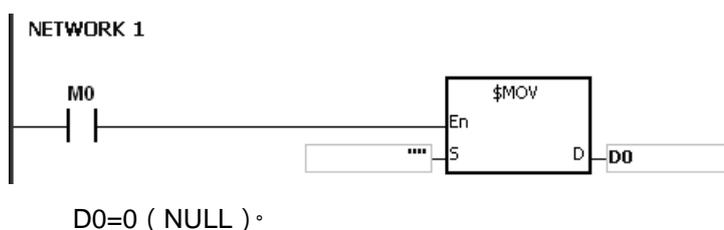
指數 D1 = $-41 \sim +35$

此外，底數 100 不存在於 D0 的內容，因為，100 是以 1.000×10^{-1} 來表現。32 位元十進浮點數的範圍為 $\pm 1175 \times 10^{-41}$ 到 $\pm 402 \times 10^{+35}$ 。

2.2.3 字串

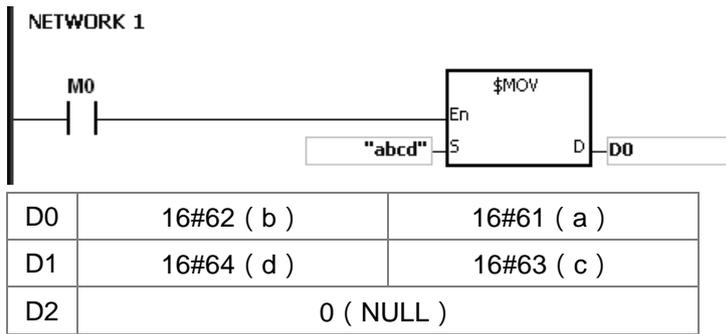
字串可以處理的是 ASCII 編碼的字 (*1)，一個完整的字串定義為字串的起始字元到結束字元 NULL 碼 (16#00) 為止。若使用者直接輸入的是字串，則最多可以輸入 31 個字且 ISPSOft 會自動補上結束字元 16#00。若使用者輸入的是暫存器，則需在結束的後面補上一個結束字元 16#00。

1. 字串搬移 NULL

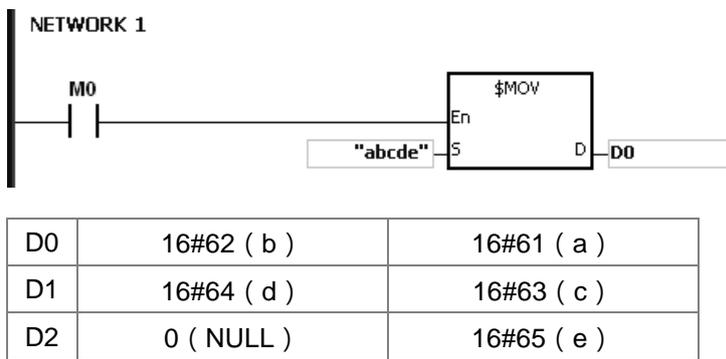


2

2. 字串為偶數的時候：



3. 字串為奇數的時候：



*1 : ASCII 碼轉換表

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	{		}	~	☒

註：標示為☒，均為不可視字元，請勿設定。

2.2.4 輸入繼電器 X

- 輸入接點 X 的功能：
輸入接點 X 與輸入裝置（按鈕開關、旋鈕開關、數字開關等的外部設備）連接，讀取輸入訊號進入 PLC。每一個輸入接點 X 的 A 或 B 接點於程式中使用次數沒有限制。輸入接點 X 之 ON/OFF 只會跟隨輸入裝置的 ON/OFF 做變化。
- 輸入接點的編號（以十進制編號）：
對 PLC 系列而言，輸入端的編號固定從 X0.0 開始算，編號的多寡跟隨 DIO 模組的輸入點數大小而變化，隨著與主機的連接順序來推算出。PLC 機種最大輸入點數可達 8192 點，範圍如下：X0.0 ~ X511.15。
- 輸入的種類：
輸入有刷新輸入和直接輸入 2 種
 1. 刷新輸入：採用程式執行前的外部輸入刷新時接收的 ON/OFF 資料來進行運算的輸入方式（如：LD X0.0）
 2. 直接輸入：採用指令執行時從外部輸入接收的 ON/OFF 資料進行運算的輸入方式（如：LD DX0.0）

2.2.5 輸出繼電器 Y

- 輸出接點 Y 的功能：
輸出接點 Y 的任務就是送出 ON/OFF 信號來驅動連接輸出接點 Y 的負載（外部信號燈、數字顯示器、電磁閥等）。輸出接點分成三種，一為繼電器（Relay），二為電晶體（Transistor），三為交流矽控器（TRIAC (Thyristors)）。每一個輸出接點 Y 的 A 或 B 接點於程式中使用次數沒有限制，但輸出 Y 的編號，在程式建議僅能使用一次，否則依 PLC 的程式掃描原理，其輸出狀態的決定權會落在程式中最後的輸出 Y 的電路。
- 輸出接點的編號（以十進制編號）：
對 PLC 系列而言，輸出端的編號固定從 Y0.0 開始算，編號的多寡跟隨 DIO 模組的輸出點數大小而變化，隨著與主機的連接順序來推算出。PLC 機種最大輸出點數可達 8192 點，範圍如下：Y0.0 ~ Y511.15。
未實際配置使用的 Y 編號可當作一般的裝置用。
- 輸出的種類：
輸出有刷新輸出和直接輸出 2 種
 1. 刷新輸出：採用程式執行到 END 指令，依據 ON/OFF 資料來進行實際輸出方式（如：OUT Y0.0）
 2. 直接輸出：採用指令執行時，直接依據 ON/OFF 資料進行實際輸出方式（如：OUT DY0.0）

2.2.6 輔助繼電器 M

輔助繼電器M有A、B接點，而且於程式當中使用次數無限制，使用者可利用輔助繼電器M來組合控制迴路，但無法直接驅動外部負載。依其性質可區分為下列二種：

1. 一般用：一般用輔助繼電器於PLC運轉時若遇到停電，其狀態將全部被復歸為OFF，再送電時其狀態仍為OFF。
2. 停電保持用：停電保持用輔助繼電器於PLC運轉時若遇到停電，其狀態將全部被保持，再送電時其狀態為停電前狀態。

2.2.7 特殊輔助繼電器 SM

每一個特殊輔助繼電器都有其特定之功用，未定義的特殊旗標請勿使用。

特殊輔助繼電器 (SM)，它的種類及功能如下所示。在編號前有“*”記號可參考SM/SR補充說明，像其中屬性欄中標示為“R”者，表示僅可作讀取的動作，若標示為“R/W”，表示可作讀寫的動作。另若標示為“-”，表示無變化。標示為“#”，則表示系統會依照PLC狀態作設定，使用者可讀取該設定值對照手冊之說明，可進一步了解系統資訊。

SM	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠設定
								↓ ON	↓ RUN	↓ STOP		
SM0	演算錯誤	○	○	○	○	○	○	OFF	OFF	-	R	OFF
SM1	演算錯誤鎖定	○	○	○	○	○	○	OFF	OFF	-	R	OFF
SM5	指令/運算元檢查錯誤	○	○	○	○	○	○	OFF	OFF	-	R	OFF
*SM8	逾時監視錯誤	○	○	○	○	○	○	OFF	-	-	R	OFF
SM9	系統錯誤	○	○	○	○	○	○	OFF	-	-	R	OFF
SM10	I/O 匯流排錯誤	○	○	○	○	○	○	OFF	-	-	R	OFF
*SM20	電源供應異常紀錄	○	○	○	○	○	○	OFF	-	-	R	OFF
*SM22	清除錯誤紀錄	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM23	清除下載紀錄	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM24	清除 PLC 狀態變更紀錄	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM25	當線上編輯模式啟動時，線上編輯處理旗標開啟	○	○	○	○	○	○	OFF	-	-	R	OFF
SM26	當除錯模式啟動時，除錯模式處理旗標開啟	○	○	○	○	○	○	OFF	-	-	R	OFF
*SM96	COM1 送信旗標	○	○	○	○	○	○	OFF	OFF	-	R/W	OFF
*SM97	COM2 送信旗標	○	X	○	X	X	X	OFF	OFF	-	R/W	OFF
*SM98	COM1 接收等待	○	○	○	○	○	○	OFF	OFF	-	R	OFF

SM	功能說明	CPUsX0-RS2	CPUsX0-EN	CPUsX1-RS2	CPUsX1-EN	CPUs60-EN2	CPUs21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SM99	COM2 接收等待	O	X	O	X	X	X	OFF	OFF	-	R	OFF
*SM100	COM1 接收完畢	O	O	O	O	O	O	OFF	OFF	-	R/W	OFF
*SM101	COM2 接收完畢	O	X	O	X	X	X	OFF	OFF	-	R/W	OFF
*SM102	COM1 MODRW 或 RS 資料接收錯誤	O	O	O	O	O	O	OFF	OFF	-	R	OFF
*SM103	COM2 MODRW 或 RS 資料接收錯誤	O	X	O	X	X	X	OFF	OFF	-	R	OFF
*SM104	COM1 接收逾時	O	O	O	O	O	O	OFF	OFF	-	R/W	OFF
*SM105	COM2 接收逾時	O	X	O	X	X	X	OFF	OFF	-	R/W	OFF
*SM106	COM1 8/16 位元處理模式選擇・ON : 8 位元・OFF : 16 位元	O	O	O	O	O	O	OFF	-	-	R/W	OFF
*SM107	COM2 8/16 位元處理模式選擇・ON : 8 位元・OFF : 16 位元	O	X	O	X	X	X	OFF	-	-	R/W	OFF
SM108	COM1 接收結束旗標	V1.01	V1.01	O	O	O	O	-	-	-	R/W	OFF
SM109	COM2 接收結束旗標	V1.01	X	O	X	X	X	-	-	-	R/W	OFF
*SM204	非停電保持區域全部清除	O	O	O	O	O	O	OFF	-	-	R/W	OFF
*SM205	停電保持區域全部清除	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM206	輸出全部禁止	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM209	COM1 通訊協定變更改用	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM210	COM1 ASCII/RTU 模式選擇・ON 時為 RTU 模式	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM211	COM2 通訊協定變更改用	O	X	O	X	X	X	OFF	-	-	R/W	OFF
SM212	COM2 ASCII/RTU 模式選擇・ON 時為 RTU 模式	O	X	O	X	X	X	OFF	-	-	R/W	OFF
SM215	PLC 運行狀態	O	O	O	O	O	O	OFF	ON	OFF	R/W	OFF
SM220	萬年曆±30 秒校正	O	O	O	O	O	O	OFF	OFF	-	R/W	OFF
*SM400	常開接點	O	O	O	O	O	O	ON	ON	ON	R	ON
*SM401	常閉接點	O	O	O	O	O	O	OFF	OFF	OFF	R	OFF
*SM402	啟始正向 (RUN 的瞬間“ON”) 脈波	O	O	O	O	O	O	OFF	ON	OFF	R	OFF
*SM403	啟始負向 (RUN 的瞬間“OFF”) 脈波	O	O	O	O	O	O	ON	OFF	ON	R	ON
*SM404	10ms 時鐘脈衝・5ms ON/5ms OFF	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM405	100ms 時鐘脈衝・50ms ON/50ms OFF	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM406	200ms 時鐘脈衝・100ms ON/100ms OFF	O	O	O	O	O	O	OFF	-	-	R	OFF

2

SM	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
*SM407	1s 時鐘脈衝 · 0.5s ON/0.5s OFF	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM408	2s 時鐘脈衝 · 1s ON/1s OFF	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM409	2n 秒時鐘脈衝 · n (秒) 開/n (秒) 關 · n 的時間間隔指定於 SR409	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM410	2n 毫秒時鐘脈衝 · n (毫秒) 開/n (毫秒) 關 · n 的時間間隔指定於 SR410	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM418	背板 1 (主背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM420	背板 2 (第一組備援延伸背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM421	背板 2 (第一組備援延伸背板) Port4 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM422	背板 3 (第二組備援延伸背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM423	背板 3 (第二組備援延伸背板) Port4 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM424	背板 4 (第三組備援延伸背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM425	背板 4 (第三組備援延伸背板) Port4 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM426	背板 5 (第四組備援延伸背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM427	背板 5 (第四組備援延伸背板) Port4 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM428	背板 6 (第五組備援延伸背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM429	背板 6 (第五組備援延伸背板) Port4 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM430	背板 7 (第六組備援延伸背板) Port2 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM431	背板 7 (第六組備援延伸背板) Port4 連線狀態	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM450	記憶卡是否存在旗標 · ON : 存在/OFF : 不存在	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM451	記憶卡防寫開關 (Protect Switch) · ON : 防寫/OFF : 無防寫	O	O	O	O	O	O	OFF	-	-	R	OFF
*SM452	記憶卡正被存取 (Accessed) 中 · ON : 存取中/OFF : 無存取	O	O	O	O	O	O	OFF	-	-	R	OFF

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
*SM453	記憶卡運行中有錯誤發生，ON-表錯誤發生	O	O	O	O	O	O	OFF	-	-	R	OFF
SM454	DATA LOGGER 啟動旗標 (ON: 啟動，OFF:關閉)	X	X	V2.01	V2.01	V1.01	O	OFF	-	-	R/W	OFF
SM455	DATA LOGGER 取樣狀態旗標 (ON: 緩衝區已滿或循環中)	X	X	V2.01	V2.01	V1.01	O	OFF	-	-	R	OFF
SM456	DATA LOGGER SD 卡動作旗標，須搭配 SR902 (ON:執行 SD 動作)	X	X	V2.01	V2.01	V1.01	O	OFF	-	-	R/W	OFF
SM457	DATA LOGGER 參數狀態旗標 (ON: 參數已設定)	X	X	V2.01	V2.01	V1.01	O	OFF	-	-	R	OFF
SM600	零旗號 (Zero flag)	O	O	O	O	O	O	OFF	-	-	R	OFF
SM601	借位旗號 (Borrow flag)	O	O	O	O	O	O	OFF	-	-	R	OFF
SM602	進位旗號 (Carry flag)	O	O	O	O	O	O	OFF	-	-	R	OFF
SM604	SORT 排序指令工作模式設定 (ON-降冪排序，OFF-升冪排序)	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM605	SMOV 工作模式指定	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM606	8/16 位元工作模式	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM607	矩陣比較旗標，比較相同值 (SM607 為 ON) 或不同值 (SM607 為 OFF)	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM608	矩陣搜尋結束旗標，當比較到最後一個 bit 時，SM608 為 ON	O	O	O	O	O	O	OFF	-	-	R	OFF
SM609	矩陣搜尋起始旗標，當 SM609 為 ON 時由第 0 個 bit 開始比較	O	O	O	O	O	O	OFF	-	-	R	OFF
SM610	矩陣位元尋找旗標，比較到達時立即停止比較動作，SM610 為 ON	O	O	O	O	O	O	OFF	-	-	R	OFF
SM611	矩陣指標錯誤旗標，指標 Pr 值超出範圍則 SM611 為 ON	O	O	O	O	O	O	OFF	-	-	R	OFF
SM612	矩陣指標遞增旗標，將指標目前值+1	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM613	矩陣指標清除旗標，將指標目前值清除為 0	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM614	矩陣旋轉位移輸出進位旗標	O	O	O	O	O	O	OFF	-	-	R	OFF
SM615	矩陣位移輸入補位旗標	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM616	矩陣旋轉位移方向旗標 SM616 為 OFF: 左移，SM616 為 ON: 右移	O	O	O	O	O	O	OFF	-	-	R/W	OFF
SM617	矩陣計數位元為 0 或位元為 1 旗標	O	O	O	O	O	O	OFF	-	-	R/W	OFF

2

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM618	矩陣計數結果為 0 時 ON	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM619	當 EI 指令被執行時 ON	0	0	0	0	0	0	OFF	OFF	-	R	OFF
SM620	表格比較指令 CMPT#全部輸出旗標 · 若比較結果為全都輸出則 SM620=ON	0	0	0	0	0	0	OFF	-	-	R	OFF
SM621	HC0 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM622	HC1 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM623	HC2 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM624	HC3 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM625	HC4 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM626	HC5 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM627	HC6 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM628	HC7 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM629	HC8 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM630	HC9 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM631	HC10 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM632	HC11 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM633	HC12 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM634	HC13 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM635	HC14 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM636	HC15 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM637	HC16 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM638	HC17 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM639	HC18 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM640	HC19 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM641	HC20 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM642	HC21 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM643	HC22 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM644	HC23 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM645	HC24 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF
SM646	HC25 計數模式設定 (ON 時為下數)	0	0	0	0	0	0	OFF	-	-	R/W	OFF

SM	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM647	HC26 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM648	HC27 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM649	HC28 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM650	HC29 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM651	HC30 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM652	HC31 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM653	HC32 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM654	HC33 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM655	HC34 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM656	HC35 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM657	HC36 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM658	HC37 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM659	HC38 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM660	HC39 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM661	HC40 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM662	HC41 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM663	HC42 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM664	HC43 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM665	HC44 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM666	HC45 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM667	HC46 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM668	HC47 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM669	HC48 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM670	HC49 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM671	HC50 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM672	HC51 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM673	HC52 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM674	HC53 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM675	HC54 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF

2

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM676	HC55 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM677	HC56 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM678	HC57 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM679	HC58 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM680	HC59 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM681	HC60 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM682	HC61 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM683	HC62 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM684	HC63 計數模式設定 (ON 時為下數)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM685	DSCLP 使用浮點數運算	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM686	RAMP 指令連續執行設定	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM687	RAMP 指令執行完畢	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM688	INCD 指令執行完畢	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM690	字串控制模式	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM691	HKY 輸入為 16 位元模式 ON：十六進制輸入·OFF：A~F 為功能鍵	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM692	HKY 指令執行完畢旗標 (ON 一個掃描週期)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM693	SEGL 指令執行完畢旗標 (ON 一個掃描週期)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM694	DSW 指令執行完畢旗標 (ON 一個掃描週期)	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM695	徑度/角度使用旗標·ON 的時候表示角度	○	○	○	○	○	○	OFF	-	-	R/W	OFF
SM699	MODBUS TCP 初始化錯誤旗標	X	V1.01	X	○	○	○	OFF	-	-	R	OFF
SM700	MODBUS TCP 連線 1 啟動旗標	X	V1.01	X	○	○	○	OFF	-	-	R/W	OFF
SM701	MODBUS TCP 連線 2 啟動旗標	X	V1.01	X	○	○	○	OFF	-	-	R/W	OFF
SM702	MODBUS TCP 連線 3 啟動旗標	X	V1.01	X	○	○	○	OFF	-	-	R/W	OFF
SM703	MODBUS TCP 連線 4 啟動旗標	X	V1.01	X	○	○	○	OFF	-	-	R/W	OFF
SM704	MODBUS TCP 連線 5 啟動旗標	X	V1.01	X	○	○	○	OFF	-	-	R/W	OFF
SM705	MODBUS TCP 連線 6 啟動旗標	X	V1.01	X	○	○	○	OFF	-	-	R/W	OFF

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM706	MODBUS TCP 連線 7 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM707	MODBUS TCP 連線 8 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM708	MODBUS TCP 連線 9 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM709	MODBUS TCP 連線 10 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM710	MODBUS TCP 連線 11 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM711	MODBUS TCP 連線 12 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM712	MODBUS TCP 連線 13 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM713	MODBUS TCP 連線 14 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM714	MODBUS TCP 連線 15 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM715	MODBUS TCP 連線 16 啟動旗標	X	V1.01	X	O	O	O	OFF	-	-	R/W	OFF
SM716	MODBUS TCP 連線 17 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM717	MODBUS TCP 連線 18 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM718	MODBUS TCP 連線 19 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM719	MODBUS TCP 連線 20 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM720	MODBUS TCP 連線 21 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM721	MODBUS TCP 連線 22 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM722	MODBUS TCP 連線 23 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM723	MODBUS TCP 連線 24 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM724	MODBUS TCP 連線 25 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM725	MODBUS TCP 連線 26 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM726	MODBUS TCP 連線 27 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM727	MODBUS TCP 連線 28 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM728	MODBUS TCP 連線 29 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM729	MODBUS TCP 連線 30 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM730	MODBUS TCP 連線 31 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM731	MODBUS TCP 連線 32 啟動旗標	X	O*1	X	O*6	O	O	OFF	-	-	R/W	OFF
SM732	MODBUS TCP 連線 33 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM733	MODBUS TCP 連線 34 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM734	MODBUS TCP 連線 35 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF

2

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM735	MODBUS TCP 連線 36 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM736	MODBUS TCP 連線 37 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM737	MODBUS TCP 連線 38 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM738	MODBUS TCP 連線 39 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM739	MODBUS TCP 連線 40 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM740	MODBUS TCP 連線 41 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM741	MODBUS TCP 連線 42 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM742	MODBUS TCP 連線 43 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM743	MODBUS TCP 連線 44 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM744	MODBUS TCP 連線 45 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM745	MODBUS TCP 連線 46 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM746	MODBUS TCP 連線 47 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM747	MODBUS TCP 連線 48 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM748	MODBUS TCP 連線 49 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM749	MODBUS TCP 連線 50 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM750	MODBUS TCP 連線 51 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM751	MODBUS TCP 連線 52 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM752	MODBUS TCP 連線 53 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM753	MODBUS TCP 連線 54 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM754	MODBUS TCP 連線 55 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM755	MODBUS TCP 連線 56 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM756	MODBUS TCP 連線 57 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM757	MODBUS TCP 連線 58 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM758	MODBUS TCP 連線 59 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM759	MODBUS TCP 連線 60 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM760	MODBUS TCP 連線 61 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM761	MODBUS TCP 連線 62 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM762	MODBUS TCP 連線 63 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF
SM763	MODBUS TCP 連線 64 啟動旗標	X	O*2	X	O*4	O	O	OFF	-	-	R/W	OFF

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM764	MODBUS TCP 連線 65 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM765	MODBUS TCP 連線 66 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM766	MODBUS TCP 連線 67 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM767	MODBUS TCP 連線 68 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM768	MODBUS TCP 連線 69 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM769	MODBUS TCP 連線 70 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM770	MODBUS TCP 連線 71 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM771	MODBUS TCP 連線 72 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM772	MODBUS TCP 連線 73 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM773	MODBUS TCP 連線 74 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM774	MODBUS TCP 連線 75 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM775	MODBUS TCP 連線 76 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM776	MODBUS TCP 連線 77 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM777	MODBUS TCP 連線 78 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM778	MODBUS TCP 連線 79 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM779	MODBUS TCP 連線 80 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM780	MODBUS TCP 連線 81 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM781	MODBUS TCP 連線 82 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM782	MODBUS TCP 連線 83 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM783	MODBUS TCP 連線 84 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM784	MODBUS TCP 連線 85 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM785	MODBUS TCP 連線 86 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM786	MODBUS TCP 連線 87 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM787	MODBUS TCP 連線 88 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM788	MODBUS TCP 連線 89 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM789	MODBUS TCP 連線 90 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM790	MODBUS TCP 連線 91 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM791	MODBUS TCP 連線 92 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM792	MODBUS TCP 連線 93 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF

2

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM793	MODBUS TCP 連線 94 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM794	MODBUS TCP 連線 95 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM795	MODBUS TCP 連線 96 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM796	MODBUS TCP 連線 97 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM797	MODBUS TCP 連線 98 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM798	MODBUS TCP 連線 99 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM799	MODBUS TCP 連線 100 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM800	MODBUS TCP 連線 101 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM801	MODBUS TCP 連線 102 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM802	MODBUS TCP 連線 103 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM803	MODBUS TCP 連線 104 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM804	MODBUS TCP 連線 105 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM805	MODBUS TCP 連線 106 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM806	MODBUS TCP 連線 107 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM807	MODBUS TCP 連線 108 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM808	MODBUS TCP 連線 109 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM809	MODBUS TCP 連線 110 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM810	MODBUS TCP 連線 111 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM811	MODBUS TCP 連線 112 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM812	MODBUS TCP 連線 113 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM813	MODBUS TCP 連線 114 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM814	MODBUS TCP 連線 115 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM815	MODBUS TCP 連線 116 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM816	MODBUS TCP 連線 117 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM817	MODBUS TCP 連線 118 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM818	MODBUS TCP 連線 119 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM819	MODBUS TCP 連線 120 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM820	MODBUS TCP 連線 121 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM821	MODBUS TCP 連線 122 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM822	MODBUS TCP 連線 123 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM823	MODBUS TCP 連線 124 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM824	MODBUS TCP 連線 125 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM825	MODBUS TCP 連線 126 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM826	MODBUS TCP 連線 127 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM827	MODBUS TCP 連線 128 啟動旗標	X	O*3	X	O*5	O	X	OFF	-	-	R/W	OFF
SM828	MODBUS TCP 連線 1 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM829	MODBUS TCP 連線 2 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM830	MODBUS TCP 連線 3 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM831	MODBUS TCP 連線 4 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM832	MODBUS TCP 連線 5 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM833	MODBUS TCP 連線 6 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM834	MODBUS TCP 連線 7 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM835	MODBUS TCP 連線 8 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM836	MODBUS TCP 連線 9 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM837	MODBUS TCP 連線 10 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM838	MODBUS TCP 連線 11 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM839	MODBUS TCP 連線 12 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM840	MODBUS TCP 連線 13 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM841	MODBUS TCP 連線 14 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM842	MODBUS TCP 連線 15 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM843	MODBUS TCP 連線 16 錯誤旗標	X	V1.01	X	O	O	O	OFF	-	-	R	OFF
SM844	MODBUS TCP 連線 17 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM845	MODBUS TCP 連線 18 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM846	MODBUS TCP 連線 19 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM847	MODBUS TCP 連線 20 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM848	MODBUS TCP 連線 21 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM849	MODBUS TCP 連線 22 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM850	MODBUS TCP 連線 23 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF

2

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM851	MODBUS TCP 連線 24 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM852	MODBUS TCP 連線 25 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM853	MODBUS TCP 連線 26 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM854	MODBUS TCP 連線 27 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM855	MODBUS TCP 連線 28 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM856	MODBUS TCP 連線 29 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM857	MODBUS TCP 連線 30 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM858	MODBUS TCP 連線 31 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM859	MODBUS TCP 連線 32 錯誤旗標	X	O*1	X	O*6	O	O	OFF	-	-	R	OFF
SM860	MODBUS TCP 連線 33 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM861	MODBUS TCP 連線 34 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM862	MODBUS TCP 連線 35 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM863	MODBUS TCP 連線 36 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM864	MODBUS TCP 連線 37 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM865	MODBUS TCP 連線 38 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM866	MODBUS TCP 連線 39 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM867	MODBUS TCP 連線 40 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM868	MODBUS TCP 連線 41 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM869	MODBUS TCP 連線 42 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM870	MODBUS TCP 連線 43 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM871	MODBUS TCP 連線 44 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM872	MODBUS TCP 連線 45 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM873	MODBUS TCP 連線 46 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM874	MODBUS TCP 連線 47 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM875	MODBUS TCP 連線 48 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM876	MODBUS TCP 連線 49 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM877	MODBUS TCP 連線 50 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM878	MODBUS TCP 連線 51 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM879	MODBUS TCP 連線 52 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM880	MODBUS TCP 連線 53 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM881	MODBUS TCP 連線 54 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM882	MODBUS TCP 連線 55 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM883	MODBUS TCP 連線 56 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM884	MODBUS TCP 連線 57 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM885	MODBUS TCP 連線 58 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM886	MODBUS TCP 連線 59 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM887	MODBUS TCP 連線 60 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM888	MODBUS TCP 連線 61 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM889	MODBUS TCP 連線 62 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM890	MODBUS TCP 連線 63 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM891	MODBUS TCP 連線 64 錯誤旗標	X	O*2	X	O*4	O	O	OFF	-	-	R	OFF
SM892	MODBUS TCP 連線 65 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM893	MODBUS TCP 連線 66 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM894	MODBUS TCP 連線 67 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM895	MODBUS TCP 連線 68 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM896	MODBUS TCP 連線 69 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM897	MODBUS TCP 連線 70 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM898	MODBUS TCP 連線 71 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM899	MODBUS TCP 連線 72 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM900	MODBUS TCP 連線 73 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM901	MODBUS TCP 連線 74 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM902	MODBUS TCP 連線 75 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM903	MODBUS TCP 連線 76 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM904	MODBUS TCP 連線 77 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM905	MODBUS TCP 連線 78 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM906	MODBUS TCP 連線 79 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM907	MODBUS TCP 連線 80 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM908	MODBUS TCP 連線 81 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF

2

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM909	MODBUS TCP 連線 82 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM910	MODBUS TCP 連線 83 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM911	MODBUS TCP 連線 84 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM912	MODBUS TCP 連線 85 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM913	MODBUS TCP 連線 86 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM914	MODBUS TCP 連線 87 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM915	MODBUS TCP 連線 88 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM916	MODBUS TCP 連線 89 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM917	MODBUS TCP 連線 90 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM918	MODBUS TCP 連線 91 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM919	MODBUS TCP 連線 92 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM920	MODBUS TCP 連線 93 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM921	MODBUS TCP 連線 94 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM922	MODBUS TCP 連線 95 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM923	MODBUS TCP 連線 96 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM924	MODBUS TCP 連線 97 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM925	MODBUS TCP 連線 98 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM926	MODBUS TCP 連線 99 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM927	MODBUS TCP 連線 100 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM928	MODBUS TCP 連線 101 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM929	MODBUS TCP 連線 102 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM930	MODBUS TCP 連線 103 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM931	MODBUS TCP 連線 104 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM932	MODBUS TCP 連線 105 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM933	MODBUS TCP 連線 106 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM934	MODBUS TCP 連線 107 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM935	MODBUS TCP 連線 108 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM936	MODBUS TCP 連線 109 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM937	MODBUS TCP 連線 110 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM938	MODBUS TCP 連線 111 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM939	MODBUS TCP 連線 112 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM940	MODBUS TCP 連線 113 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM941	MODBUS TCP 連線 114 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM942	MODBUS TCP 連線 115 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM943	MODBUS TCP 連線 116 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM944	MODBUS TCP 連線 117 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM945	MODBUS TCP 連線 118 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM946	MODBUS TCP 連線 119 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM947	MODBUS TCP 連線 120 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM948	MODBUS TCP 連線 121 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM949	MODBUS TCP 連線 122 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM950	MODBUS TCP 連線 123 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM951	MODBUS TCP 連線 124 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM952	MODBUS TCP 連線 125 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM953	MODBUS TCP 連線 126 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM954	MODBUS TCP 連線 127 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
SM955	MODBUS TCP 連線 128 錯誤旗標	X	O*3	X	O*5	O	X	OFF	-	-	R	OFF
*SM1000	Ethernet 設定旗標。 ON 時·將 SR1000~SR1006 的資料寫入快閃記憶體	X	O	X	O	O	O	-	-	-	R/W	OFF
*SM1001	Ethernet Port1 網路線連線狀態	X	X	X	O	O	O	-	-	-	R	OFF
*SM1002	Ethernet Port2 網路線連線狀態	X	X	X	X	O	X	-	-	-	R	OFF
*SM1003	光纖同步電纜連線狀態	X	X	X	X	O	X	-	-	-	R	OFF
*SM1089	MODBUS TCP Server 連線已滿	X	V1.06	X	V2.00	O	O	OFF	-	-	R	OFF
*SM1090	TCP 連線忙碌	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1091	UDP 連線忙碌	X	O	X	O	O	O	OFF	-	-	R	OFF
SM1100	網路線未連接	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1106	乙太網路連線錯誤旗標	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1107	HWCONFIG 乙太網路-基本設定 參數設定錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF

2

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
*SM1108	HWCONFIG 乙太網路-進階設定 過濾器設定錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1109	HWCONFIG 乙太網路-進階設定 TCP/UDP 所設定的本機通訊埠已被使用	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1112	Email 相關的參數設定錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1113	Email 發送服務失敗	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1116	Email 發送條件 1.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1117	Email 發送條件 1	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1118	Email 發送條件 1 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1119	Email 發送條件 1 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1120	Email 發送條件 1 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1121	Email 發送條件 1 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1122	Email 發送條件 1 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1123	Email 發送條件 1 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1124	Email 發送條件 1 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1125	Email 發送條件 1 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1126	Email 發送條件 2.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1127	Email 發送條件 2	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1128	Email 發送條件 2 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1129	Email 發送條件 2 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1130	Email 發送條件 2 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1131	Email 發送條件 2 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1132	Email 發送條件 2 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1133	Email 發送條件 2 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF

SM	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SM1134	Email 發送條件 2 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1135	Email 發送條件 2 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1136	Email 發送條件 3.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1137	Email 發送條件 3	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1138	Email 發送條件 3 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1139	Email 發送條件 3 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1140	Email 發送條件 3 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1141	Email 發送條件 3 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1142	Email 發送條件 3 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1143	Email 發送條件 3 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1144	Email 發送條件 3 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1145	Email 發送條件 3 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1146	Email 發送條件 4.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1147	Email 發送條件 4	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1148	Email 發送條件 4 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1149	Email 發送條件 4 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1150	Email 發送條件 4 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1151	Email 發送條件 4 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1152	Email 發送條件 4 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1153	Email 發送條件 4 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1154	Email 發送條件 4 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF

2

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
*SM1155	Email 發送條件 4 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1156	Email 發送條件 5.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1157	Email 發送條件 5	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1158	Email 發送條件 5 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1159	Email 發送條件 5 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1160	Email 發送條件 5 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1161	Email 發送條件 5 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1162	Email 發送條件 5 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1163	Email 發送條件 5 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1164	Email 發送條件 5 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1165	Email 發送條件 5 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1166	Email 發送條件 6.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1167	Email 發送條件 6	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1168	Email 發送條件 6 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1169	Email 發送條件 6 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1170	Email 發送條件 6 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1171	Email 發送條件 6 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1172	Email 發送條件 6 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1173	Email 發送條件 6 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1174	Email 發送條件 6 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1175	Email 發送條件 6 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1176	Email 發送條件 7.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1177	Email 發送條件 7	X	O	X	O	O	O	OFF	-	-	R	OFF

SM	功能說明	CPUsX0-RS2	CPUsX0-EN	CPUsX1-RS2	CPUsX1-EN	CPUs60-EN2	CPUs21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SM1178	Email 發送條件 7 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1179	Email 發送條件 7 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1180	Email 發送條件 7 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1181	Email 發送條件 7 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1182	Email 發送條件 7 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1183	Email 發送條件 7 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1184	Email 發送條件 7 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1185	Email 發送條件 7 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1186	Email 發送條件 8.開關	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1187	Email 發送條件 8	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1188	Email 發送條件 8 被觸發，因乙太網路連線失敗，郵件無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1189	Email 發送條件 8 被觸發，郵件發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1190	Email 發送條件 8 被觸發，因郵件內容有錯誤，無法發送成功。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1191	Email 發送條件 8 被觸發，郵件發送中。	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1192	Email 發送條件 8 被觸發，SMTP 回覆逾時	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1193	Email 發送條件 8 被觸發，SMTP 回覆錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1194	Email 發送條件 8 被觸發，附件大小錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
*SM1195	Email 發送條件 8 被觸發，找不到附件	X	O	X	O	O	O	OFF	-	-	R	OFF
SM1196	Socket 配置錯誤	X	O	X	O	O	O	OFF	-	-	R/W	OFF
SM1270	TCP Socket 1--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1271	TCP Socket 1--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1272	TCP Socket 1--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1273	TCP Socket 1--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF

2

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM1274	TCP Socket 1--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1275	TCP Socket 1--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1276	TCP Socket 1--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1277	TCP Socket 1--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1278	TCP Socket 2--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1279	TCP Socket 2--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1280	TCP Socket 2--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1281	TCP Socket 2--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1282	TCP Socket 2--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1283	TCP Socket 2--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1284	TCP Socket 2--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1285	TCP Socket 2--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1286	TCP Socket 3--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1287	TCP Socket 3--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1288	TCP Socket 3--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1289	TCP Socket 3--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1290	TCP Socket 3--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1291	TCP Socket 3--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1292	TCP Socket 3--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1293	TCP Socket 3--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1294	TCP Socket 4--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1295	TCP Socket 4--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1296	TCP Socket 4--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1297	TCP Socket 4--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1298	TCP Socket 4--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1299	TCP Socket 4--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1300	TCP Socket 4--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1301	TCP Socket 4--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1302	TCP Socket 5--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM1303	TCP Socket 5--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1304	TCP Socket 5--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1305	TCP Socket 5--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1306	TCP Socket 5--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1307	TCP Socket 5--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1308	TCP Socket 5--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1309	TCP Socket 5--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1310	TCP Socket 6--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1311	TCP Socket 6--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1312	TCP Socket 6--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1313	TCP Socket 6--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1314	TCP Socket 6--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1315	TCP Socket 6--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1316	TCP Socket 6--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1317	TCP Socket 6--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1318	TCP Socket 7--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1319	TCP Socket 7--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1320	TCP Socket 7--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1321	TCP Socket 7--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1322	TCP Socket 7--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1323	TCP Socket 7--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1324	TCP Socket 7--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1325	TCP Socket 7--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1326	TCP Socket 8--連線旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1327	TCP Socket 8--資料接收完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1328	TCP Socket 8--資料傳送完成旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1329	TCP Socket 8--開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1330	TCP Socket 8--關閉旗標	X	O	X	O	O	O	ON	ON	-	R	ON
SM1331	TCP Socket 8--傳送中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF

2

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM1332	TCP Socket 8--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1333	TCP Socket 8--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1334	UDP Socket 1--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1335	UDP Socket 1--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1336	UDP Socket 1--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1337	UDP Socket 1--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1338	UDP Socket 1--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1339	UDP Socket 2--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1340	UDP Socket 2--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1341	UDP Socket 2--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1342	UDP Socket 2--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1343	UDP Socket 2--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1344	UDP Socket 3--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1345	UDP Socket 3--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1346	UDP Socket 3--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1347	UDP Socket 3--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1348	UDP Socket 3--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1349	UDP Socket 4--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1350	UDP Socket 4--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1351	UDP Socket 4--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1352	UDP Socket 4--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1353	UDP Socket 4--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1354	UDP Socket 5--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1355	UDP Socket 5--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1356	UDP Socket 5--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1357	UDP Socket 5--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1358	UDP Socket 5--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1359	UDP Socket 6--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1360	UDP Socket 6--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM1361	UDP Socket 6--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1362	UDP Socket 6--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1363	UDP Socket 6--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1364	UDP Socket 7--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1365	UDP Socket 7--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1366	UDP Socket 7--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1367	UDP Socket 7--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1368	UDP Socket 7--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1369	UDP Socket 8--連線已開啟旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1370	UDP Socket 8--資料已接收旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1371	UDP Socket 8--資料已傳送旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1372	UDP Socket 8--接收中旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1373	UDP Socket 8--錯誤旗標	X	O	X	O	O	O	OFF	OFF	-	R	OFF
SM1374	Web 設定錯誤	X	O	X	O	O	O	OFF	-	-	R	OFF
SM1375	TCP Socket 1--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1376	TCP Socket 2--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1377	TCP Socket 3--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1378	TCP Socket 4--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1379	TCP Socket 5--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1380	TCP Socket 6--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1381	TCP Socket 7--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1382	TCP Socket 8--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1383	UDP Socket 1--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1384	UDP Socket 2--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1385	UDP Socket 3--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1386	UDP Socket 4--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1387	UDP Socket 5--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1388	UDP Socket 6--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1389	UDP Socket 7--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF
SM1390	UDP Socket 8--奇數位元組啟動旗標	X	V1.05	X	O	O	O	OFF	-	-	R/W	OFF

2

SM	功能說明	CP5X0-RS2	CP5X0-EN	CP5X1-RS2	CP5X1-EN	CP560-EN2	CP521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
*SM1392 ↓ SM1423	PLC LINK 連線旗標 (從站 ID 1) ↓ PLC LINK 連線旗標 (從站 ID 32)	O	O	X	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 連線旗標(從站 ID 1) ↓ COM1-MODBUS 連線旗標 (從站 ID 32)	X	X	O	O	O	O	-	-	-	R/W	OFF
*SM1424 ↓ SM1455	PLC LINK 資料交換旗標 (從站 ID 1) ↓ PLC LINK 資料交換旗標 (從站 ID 32)	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 資料交換中旗標 (從站 ID 1) ↓ COM1-MODBUS 資料交換中旗標 (從站 ID 32)	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1456 ↓ SM1487	PLC LINK 讀取錯誤 (從站 ID 1) ↓ PLC LINK 讀取錯誤 (從站 ID 32)	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 讀取錯誤(從站 ID 1) ↓ COM1-MODBUS 讀取錯誤 (從站 ID 32)	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1488 ↓ SM1519	PLC LINK 寫入錯誤 (從站 ID 1) ↓ PLC LINK 寫入錯誤 (從站 ID 32)	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 寫入錯誤(從站 ID 1) ↓ COM1-MODBUS 寫入錯誤 (從站 ID 32)	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1520 ↓ SM1551	PLC LINK 讀取資料完畢 (從站 ID 1) ↓ PLC LINK 讀取資料完畢 (從站 ID 32)	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 讀取資料完畢 (從站 ID 1) ↓ COM1-MODBUS 讀取資料完畢 (從站 ID 32)	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1552 ↓ SM1583	PLC LINK 寫入資料完畢 (從站 ID 1) ↓ PLC LINK 寫入資料完畢 (從站 ID 32)	O	O	X	X	X	X	OFF	-	-	R	OFF

SM	功能說明	CPUsX0-RS2	CPUsX0-EN	CPUsX1-RS2	CPUsX1-EN	CPUs60-EN2	CPUs21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SM1552 ↓ SM1583	COM1-MODBUS 寫入資料完畢 (從站 ID 1) ↓ COM1-MODBUS 寫入資料完畢 (從站 ID 32)	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1584	PLC LINK 連線啟動旗標	O	O	X	X	X	X	OFF	-	-	R/W	OFF
	COM1-MODBUS 連線啟動旗標	X	X	O	O	O	O	OFF	-	-	R/W	OFF
*SM1585	PLC LINK 使用者分配旗標	O	O	X	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 使用者分配旗標	X	X	O	O	O	O	-	-	-	R/W	OFF
*SM1586	PLC LINK 自動模式旗標	O	O	X	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 自動模式旗標	X	X	O	O	O	O	-	-	-	R/W	OFF
*SM1587	PLC LINK 手動模式旗標	O	O	X	X	X	X	-	-	-	R/W	OFF
*SM1588	PLC LINK 偵測旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 偵測旗標	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1589	PLC LINK 錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
*SM1590	PLC LINK 裝置位址錯誤	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 裝置位址錯誤	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1591	PLC LINK 逾時	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 逾時	X	X	O	O	O	O	OFF	-	-	R	OFF
*SM1592	PLC LINK 限制時間錯誤	O	O	X	X	X	X	OFF	-	-	R	OFF
*SM1593	PLC LINK 標準 MODBUS 通訊協定(0) /AH 通訊協定 (1)	O	O	X	X	X	X	-	-	-	R/W	OFF
*SM1594	PLC LINK 在目前環境自動偵測從站。只有在 PLC Link 於停止模式可被使用。OFF：偵測結束或等待偵測 (預設)。ON 正在偵測從站。	O	O	X	X	X	X	OFF	-	-	R/W	OFF
	COM1-MODBUS 在目前環境自動偵測從站。OFF:偵測結束或等待偵測 (預設)。ON:正在偵測從站。只有在 COM1-MODBUS 於停止模式可被使用。	X	X	O	O	O	O	OFF	-	-	R/W	OFF
*SM1595	使用者分配從站 ID (1) /自動分配從站 ID (0)	O	O	X	X	X	X	-	-	-	R/W	OFF
*SM1596	PLC LINK 連線操作錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM1-MODBUS 連線操作錯誤旗標	X	X	O	O	O	O	OFF	-	-	R	OFF

2

SM	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
*SM1597	PLC LINK 延伸埠 (SM1597=ON 表示啟動延伸埠)	O	O	X	X	X	X	-	-	-	R/W	OFF
*SM1598	PLC LINK 同步讀/寫功能 (SM1598=ON 表示啟動同步讀寫)	O	O	X	X	X	X	-	-	-	R/W	OFF
	COM1-MODBUS 同步讀/寫功能 (SM1598=ON 表示啟動同步讀寫)	X	X	O	O	O	O	-	-	-	R/W	OFF
SM1599 ↓ SM1630	COM1-MODBUS 從站 1 啟動旗標 ↓ COM1-MODBUS 從站 32 啟動旗標	X	X	O	O	O	O	OFF	-	-	R/W	OFF
SM1720 ↓ SM1751	PLC LINK 變更讀取功能碼為 0x04 旗標 (從站 ID 1) ↓ PLC LINK 變更讀取功能碼為 0x04 旗標 (從站 ID 32)	O	O	X	X	X	X	OFF	OFF	-	R/W	OFF
SM1752 ↓ SM1768	COM2-MODBUS 資料交換中旗標 (從站 ID 1) ↓ COM2-MODBUS 資料交換中旗標 (從站 ID 17)	X	X	O	X	X	X	OFF	-	-	R	OFF
SM1769	Ether Link 狀態旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 資料交換中旗標 (從站 ID 18)	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1770	Ether Link 起始旗標 (CPU)	O	O	X	X	X	X	OFF	-	-	R/W	OFF
	COM2-MODBUS 資料交換中旗標 (從站 ID 19)	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1771	COM2-MODBUS 資料交換中旗標 (從站 ID 20)	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1772 ↓ SM1783	Ether Link 起始旗標 (埠 0) ↓ Ether Link 起始旗標 (埠 11)	O	O	X	X	X	X	OFF	-	-	R/W	OFF
	COM2-MODBUS 資料交換中旗標 (從站 ID 21) ↓ COM2-MODBUS 資料交換中旗標 (從站 ID 32)	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1784 ↓ SM1787	Ether Link 起始旗標 (埠 12) ↓ Ether Link 起始旗標 (埠 15)	O	O	X	X	X	X	OFF	-	-	R/W	OFF

SM	功能說明	CPUsX0-RS2	CPUsX0-EN	CPUsX1-RS2	CPUsX1-EN	CPUs60-EN2	CPUs21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SM1784 ↓ SM1787	COM2-MODBUS 從站 1 讀取錯誤旗標 ↓ COM2-MODBUS 從站 4 讀取錯誤旗標	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1788	Ether Link 錯誤旗標 (CPU) COM2-MODBUS 從站 5 讀取錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
*SM1789	COM2-MODBUS 從站 6 讀取錯誤旗標	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1790 ↓ SM1805	Ether Link 錯誤旗標 (埠 0) ↓ Ether Link 錯誤旗標 (埠 15) COM2-MODBUS 從站 7 讀取錯誤旗標 ↓ COM2-MODBUS 從站 22 讀取錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
SM1806	Ether Link 錯誤旗標 (CPU) COM2-MODBUS 從站 23 讀取錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
*SM1807	COM2-MODBUS 從站 24 讀取錯誤旗標	X	X	O	X	X	X	OFF	-	-	R	OFF
*SM1808 ↓ SM1815	Ether Link 狀態旗標 (埠 0) ↓ Ether Link 狀態旗標 (埠 7) COM2-MODBUS 從站 25 讀取錯誤旗標 ↓ COM2-MODBUS 從站 32 讀取錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
*SM1816 ↓ SM1823	Ether Link 狀態旗標 (埠 8) ↓ Ether Link 狀態旗標 (埠 15)	O	O	X	X	X	X	OFF	-	-	R	OFF
*SM1816 ↓ SM1823	COM2-MODBUS 從站 1 寫入錯誤旗標 ↓ COM2-MODBUS 從站 8 寫入錯誤旗標	X	X	O	X	X	X	OFF	-	-	R	OFF
SM1824 ↓ SM1847	Ether Link 區塊 1 主動旗標 ↓ Ether Link 區塊 24 主動旗標 COM2-MODBUS 從站 9 寫入錯誤旗標 ↓ COM2-MODBUS 從站 32 寫入錯誤旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
SM1848 ↓ SM1879	Ether Link 區塊 25 主動旗標 ↓ Ether Link 區塊 56 主動旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
SM1848 ↓ SM1879	COM2-MODBUS 從站 1 讀取資料完畢 旗標 ↓ COM2-MODBUS 從站 32 讀取資料完畢 旗標	X	X	O	X	X	X	OFF	-	-	R	OFF

2

SM	功能說明	CPUsX0-RS2	CPUsX0-EN	CPUsX1-RS2	CPUsX1-EN	CPUs60-EN2	CPUs21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM1880 ↓ SM1911	Ether Link 區塊 57 主動旗標 ↓ Ether Link 區塊 88 主動旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 從站 1 寫入資料完畢 旗標 ↓ COM2-MODBUS 從站 32 寫入資料完畢 旗標	X	X	O	X	X	X	OFF	-	-	R	OFF
SM1912	Ether Link 區塊 89 主動旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 同步讀/寫功能	X	X	O	X	X	X	OFF	-	-	R/W	OFF
SM1913 ↓ SM1944	Ether Link 區塊 90 主動旗標 ↓ Ether Link 區塊 121 主動旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
	COM2-MODBUS 從站 1 啟動旗標 ↓ COM2-MODBUS 從站 32 啟動旗標	X	X	O	X	X	X	OFF	-	-	R/W	OFF
SM1945 ↓ SM1951	Ether Link 區塊 122 主動旗標 ↓ Ether Link 區塊 128 主動旗標	O	O	X	X	X	X	OFF	-	-	R	OFF
SM2000	EMDRW 1 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2001	EMDRW 1 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2002	EMDRW 1 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2003	EMDRW 1 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2004	EMDRW 1 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2005	EMDRW 1 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2006	EMDRW 2 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2007	EMDRW 2 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2008	EMDRW 2 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2009	EMDRW 2 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2010	EMDRW 2 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2011	EMDRW 2 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2012	EMDRW 3 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2013	EMDRW 3 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2014	EMDRW 3 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2015	EMDRW 3 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF

SM	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SM2016	EMDRW 3 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2017	EMDRW 3 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2018	EMDRW 4 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2019	EMDRW 4 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2020	EMDRW 4 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2021	EMDRW 4 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2022	EMDRW 4 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2023	EMDRW 4 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2024	EMDRW 5 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2025	EMDRW 5 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2026	EMDRW 5 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2027	EMDRW 5 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2028	EMDRW 5 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2029	EMDRW 5 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2030	EMDRW 6 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2031	EMDRW 6 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2032	EMDRW 6 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2033	EMDRW 6 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2034	EMDRW 6 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2035	EMDRW 6 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2036	EMDRW 7 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2037	EMDRW 7 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2038	EMDRW 7 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2039	EMDRW 7 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2040	EMDRW 7 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2041	EMDRW 7 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2042	EMDRW 8 送信旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2043	EMDRW 8 等待旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2044	EMDRW 8 接收旗標	X	O	X	O	O	O	OFF	OFF	OFF	R/W	OFF
SM2045	EMDRW 8 錯誤旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF

2

SM	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠設定
SM2046	EMDRW 8 逾時旗標	X	O	X	O	O	O	OFF	OFF	OFF	R	OFF
SM2047	EMDRW 8 關閉旗標	X	O	X	O	O	O	ON	ON	ON	R	ON
SM2048 ↓ SM2079	EtherNet/IP Scanner (主站) 資料交換 連線 1 啟動旗標 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 32 啟動旗標	X	X	X	V2.00	O	X	OFF	-	-	R	OFF
SM2080 ↓ SM2111	EtherNet/IP Scanner (主站) 資料交換 連線 33 啟動旗標 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 64 啟動旗標	X	X	X	V2.00 *6	O	X	OFF	-	-	R	OFF
SM2112 ↓ SM2175	EtherNet/IP Scanner (主站) 資料交換 連線 65 啟動旗標 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 128 啟動旗標	X	X	X	V2.00 *4	O	X	OFF	-	-	R	OFF
SM2176 ↓ SM2303	EtherNet/IP Scanner (主站) 資料交換 連線 129 啟動旗標 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 256 啟動旗標	X	X	X	V2.00 *5	O	X	OFF	-	-	R	OFF
SM2304 ↓ SM2311	Ethernet/IP Adapter (從站) I/O Connection 1 錯誤旗標 ↓ Ethernet/IP Adapter (從站) I/O Connection 8 錯誤旗標	X	X	X	V2.03	V1.02	X	OFF	-	-	R	OFF
SM2312 ↓ SM2319	EtherNet/IP Adapter (從站) I/O Connection 1 連線旗標 ↓ EtherNet/IP Adapter (從站) I/O Connection 8 連線旗標	X	X	X	V2.03	V1.02	X	OFF	-	-	R	OFF

註：有*之 SM 請參考 SM/SR 補充說明

*1：僅支援機種 AHCPU530-EN、AHCPU520-EN 及 AHCPU510-EN。

*2：僅支援機種 AHCPU530-EN 及 AHCPU520-EN。

*3：僅支援機種 AHCPU530-EN。

*4：僅支援機種 AHCPU531-EN 及 AHCPU521-EN。

*5：僅支援機種 AHCPU531-EN。

*6：僅支援機種 AHCPU531-EN、AHCPU521-EN 及 AHCPU511-EN。

2.2.8 特殊輔助繼電器 SM 的更新時間

適用機種	特殊輔助繼電器	更新時間
	SM0~SM1	系統自動設定與清除-程式執行有誤自動設 ON
	SM5	系統自動設定與清除- (1) 重新將程式寫入 PLC (2) 重新上電 PLC 第一次 Stop=>Run
	SM8	系統自動設定與清除-WDT 發生自動設 ON
	SM9	系統自動設定與清除-System 有誤自動設 ON
	SM10	系統自動設定與清除-I/O BUS 有誤自動設 ON
	SM20	曾經發生內部電源供應不足後，電源又回復到正常狀態時記錄。
	SM22、SM23、SM24	使用者設定，系統自動清除-ON 表執行清除
	SM25、SM26	系統自動設定與清除-每一掃描週期更新一次
	SM96、SM97	使用者設定，傳送完畢後系統自動清除
	SM98、SM99	系統自動設定與清除-傳送通訊命令自動設 ON
	SM100、SM101	系統自動設定，使用者清除-接收到回覆的通訊命令自動設 ON
	SM102、SM103	系統自動設定，使用者清除-接收到回覆的通訊命令有誤自動設 ON
	SM104、SM105	系統自動設定，使用者清除-當接收逾時時自動設 ON
	SM106、SM107	使用者設定與清除-ON 表 8 位元模式，OFF 表 16 位元模式
	SM108、SM109	使用者設定與清除
	SM204、SM205	使用者設定，系統自動清除-ON 表執行清除
	SM206	使用者設定與清除-ON 表輸出全部禁止
	SM209	使用者設定，系統自動清除-ON 表執行 COM1 通訊協定變更
	SM210	使用者設定與清除-ON 表 COM1 為 RTU 模式
	SM211	使用者設定，系統自動清除-ON 表執行 COM2 通訊協定變更
	SM212	使用者設定與清除-ON 表 COM2 為 RTU 模式
	SM215	使用者設定與清除-ON 表 PLC RUN
	SM220	使用者設定與清除-ON 表萬年曆±30 秒校正
	SM400、SM401	系統自動設定與清除-每一掃描週期更新一次
	SM402、SM403	系統自動設定與清除-每執行至 END 更新
	SM404	系統自動設定與清除-每 5ms 更新一次
	SM405	系統自動設定與清除-每 50ms 更新一次
	SM406	系統自動設定與清除-每 100ms 更新一次
	SM407	系統自動設定與清除-每 0.5s 更新一次
	SM408	系統自動設定與清除-每 1s 更新一次
	SM409	系統自動設定與清除-每 n (s) 更新一次，n (s) 由 SR409 設定
	SM410	系統自動設定與清除-每 n (s) 更新一次，n (s) 由 SR410 設定
	SM418	系統自動設定與清除

2

適用機種	特殊輔助繼電器	更新時間
	SM420~SM431	系統自動設定與清除
	SM450	系統自動設定與清除-ON 表記憶卡被插入主機
	SM451	使用者設定與清除-ON 表記憶卡防寫
	SM452	系統自動設定與清除-ON 表記憶卡正在執行存取動作
	SM453	系統自動設定與清除-ON 表記憶卡運行中有錯誤發生
	SM454	使用者設定與清除
	SM455	系統自動設定與清除
	SM456	使用者設定 ON 執行儲存・完成後系統自動清除
	SM457	系統自動設定與清除
	SM600、SM601、 SM602	系統自動設定與清除-指令執行時更新
	SM604	使用者設定與清除-ON 降冪排序 SORT 指令執行時更新
	SM605	使用者設定與清除
	SM606	使用者設定與清除-ON 表 8 位元模式
	SM607	使用者自行設定
	SM608	指令執行時更新
	SM609	使用者自行設定
	SM610、SM611	指令執行時更新
	SM612、SM613	使用者自行設定
	SM614	指令執行時更新
	SM615、SM616、 SM617	使用者自行設定
	SM618	指令執行時更新
	SM619	EI 或 DI 指令執行時更新
	SM620	CMPT 指令執行時更新
	SM621~SM686	使用者自行設定
	SM687	RAMP 指令執行時更新
	SM688	INCD 指令執行時更新
	SM690、SM691	使用者自行設定
	SM692	HKY 指令執行時更新
	SM693	SEGL 指令執行時更新
	SM694	DSW 指令執行時更新
	SM695	使用者自行設定
AH5X0	SM699	1.每次下載 Ether Link 或資料交換參數後 2.使用系統還原功能後
AH5X1		1.每次下載 MODBUS TCP 資料交換表後 2.使用系統還原功能後

適用機種	特殊輔助繼電器	更新時間
AH5X0	SM700-SM827	1.下載資料交換參數後 2.每個掃描時間更新
AH5X1		使用者自行設定及清除
	SM828-SM955	1.下載資料交換參數後 2.每個掃描時間更新
	SM1000	使用者自行設定
	SM1001~1003	使用者自行設定
	SM1089	每個掃描時間更新
	SM1090	TCP 連接滿時為 ON
	SM1091	UDP 連接滿時為 ON
	SM1100	執行乙太網控制指令 (API2200-API2205) 時或網路線重新連接時
	SM1106	PHY 初始化失敗時為 ON
	SM1107	IP、子網遮罩、閘道設定有誤時為 ON
	SM1108	濾波器設定有誤時為 ON
	SM1109	Socket 功能啟動且使用同一個通訊埠時為 ON
	SM1112	設定錯誤時為 ON
	SM1113	伺服器錯誤時為 ON
	SM1116	PLC 參數的觸發條件啟用時為 ON
	SM1117	PLC 參數的觸發條件被觸發時為 ON
	SM1118	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1119	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1120	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1121	觸發條件啟用時且郵件已送出時為 ON
	SM1122	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1123	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1124	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1125	觸發條件被觸發時且附件不存在時為 ON
	SM1126	PLC 參數的觸發條件啟用時為 ON
	SM1127	PLC 參數的觸發條件被觸發時為 ON
	SM1128	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1129	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1130	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1131	觸發條件啟用時且郵件已送出時為 ON
	SM1132	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1133	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1134	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1135	觸發條件被觸發時且附件不存在時為 ON

2

適用機種	特殊輔助繼電器	更新時間
	SM1136	PLC 參數的觸發條件啟用時為 ON
	SM1137	PLC 參數的觸發條件被觸發時為 ON
	SM1138	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1139	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1140	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1141	觸發條件啟用時且郵件已送出時為 ON
	SM1142	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1143	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1144	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1145	觸發條件被觸發時且附件不存在時為 ON
	SM1146	PLC 參數的觸發條件啟用時為 ON
	SM1147	PLC 參數的觸發條件被觸發時為 ON
	SM1148	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1149	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1150	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1151	觸發條件啟用時且郵件已送出時為 ON
	SM1152	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1153	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1154	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1155	觸發條件被觸發時且附件不存在時為 ON
	SM1156	PLC 參數的觸發條件啟用時為 ON
	SM1157	PLC 參數的觸發條件被觸發時為 ON
	SM1158	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1159	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1160	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1161	觸發條件啟用時且郵件已送出時為 ON
	SM1162	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1163	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1164	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1165	觸發條件被觸發時且附件不存在時為 ON
	SM1166	PLC 參數的觸發條件啟用時為 ON
	SM1167	PLC 參數的觸發條件被觸發時為 ON
	SM1168	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1169	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1170	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1171	觸發條件啟用時且郵件已送出時為 ON

適用機種	特殊輔助繼電器	更新時間
	SM1172	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1173	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1174	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1175	觸發條件被觸發時且附件不存在時為 ON
	SM1176	PLC 參數的觸發條件啟用時為 ON
	SM1177	PLC 參數的觸發條件被觸發時為 ON
	SM1178	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1179	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1180	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1181	觸發條件啟用時且郵件已送出時為 ON
	SM1182	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1183	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1184	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1185	觸發條件被觸發時且附件不存在時為 ON
	SM1186	PLC 參數的觸發條件啟用時為 ON
	SM1187	PLC 參數的觸發條件被觸發時為 ON
	SM1188	自從啟用本觸發條件後，無任何郵件被發送時為 ON
	SM1189	觸發條件啟用時且最後一封郵件已成功送出時為 ON
	SM1190	啟用本觸發條件且最近一封郵件發送錯誤時為 ON
	SM1191	觸發條件啟用時且郵件已送出時為 ON
	SM1192	觸發條件被觸發時且 SMTP 伺服器回應逾時時為 ON
	SM1193	觸發條件被觸發時且 SMTP 伺服器回應錯誤時為 ON
	SM1194	觸發條件被觸發時且附件大小超過限制時為 ON
	SM1195	觸發條件被觸發時且附件不存在時為 ON
	SM1196	Socket 配置有誤時為 ON
	SM1270~SM1373	執行 socket 功能時更新
	SM1374~SM1390	使用者自行設定及清除
AH5X0	SM1392	PLC Link 啟動且主站連接從站站號 1 時為 ON
	↓	↓
	SM1423	PLC Link 啟動且主站連接從站站號 32 時為 ON
AH5X1		COM1-MODBUS 啟動且主站連接從站站號 1 時為 ON
		↓
		COM1-MODBUS 啟動且主站連接從站站號 32 時為 ON
AH5X0	SM1424	PLC Link 啟動且主站存取從站站號 1 的資料時為 ON
	↓	↓
	SM1455	PLC Link 啟動且主站存取從站站號 32 的資料時為 ON

2

適用機種	特殊輔助繼電器	更新時間
AH5X1	SM1424	COM1-MODBUS 啟動且主站存取從站站號 1 的資料時為 ON
	↓ SM1455	COM1-MODBUS 啟動且主站存取從站站號 32 的資料時為 ON
AH5X0	SM1456 ↓ SM1487	PLC Link 啟動且從站站號 1 讀取錯誤時為 ON
AH5X1		COM1-MODBUS 啟動且從站站號 1 讀取錯誤時為 ON
AH5X0	SM1488 ↓ SM1519	PLC Link 啟動且從站站號 1 寫入錯誤時為 ON
AH5X1		COM1-MODBUS 啟動且從站站號 1 寫入錯誤時為 ON
AH5X0	SM1520 ↓ SM1551	PLC Link 啟動且主站結束讀取從站站號 1 時為 ON
AH5X1		COM1-MODBUS 啟動且主站結束讀取從站站號 1 時為 ON
AH5X0	SM1552 ↓ SM1583	PLC Link 啟動且主站結束寫入從站站號 1 時為 ON
AH5X1		COM1-MODBUS 啟動且主站結束寫入從站站號 1 時為 ON
	SM1584~SM1587	使用者自行設定及清除
	SM1588	主站偵測從站時為 ON
	SM1589	錯誤發生時為 ON
	SM1590	裝置編號錯誤時為 ON
	SM1591	逾時時為 ON
	SM1592	輪詢次數不正確時為 ON
	SM1593~SM1595	使用者自行設定及清除
AH5X0	SM1596	PLC Link 啟動且有錯誤發生時 · SM1596 為 ON
AH5X1		COM1-MODBUS 啟動且有錯誤發生時 · SM1596 為 ON
	SM1597~SM1630	使用者自行設定及清除
	SM1720~SM1751	使用者自行設定及清除
	SM1752~SM1768	每一掃描週期更新一次
AH5X0	SM1769	Ether Link 有錯誤發生時為 ON

適用機種	特殊輔助繼電器	更新時間
AH5X1	SM1769	每一掃描週期更新一次
AH5X0	SM1770	使用者自行設定及清除
AH5X1		每一掃描週期更新一次
	SM1771	每一掃描週期更新一次
AH5X0	SM1772~SM1788	使用者自行設定及清除
AH5X1		每一掃描週期更新一次
	SM1789	每一掃描週期更新一次
AH5X0	SM1790~SM1805	相對應之通訊埠有誤時為 ON
AH5X1		每一掃描週期更新一次
AH5X0	SM1806	相對應之通訊口的 Ether Link 功能啟動時為 ON
AH5X1		每一掃描週期更新一次
	SM1807	每一掃描週期更新一次
AH5X0	SM1808~SM1823	相對應之通訊口的 Ether Link 功能啟動時為 ON
AH5X1		每一掃描週期更新一次
	SM1824~SM1911	每一掃描週期更新一次
AH5X0	SM1912	每一掃描週期更新一次
AH5X1	SM1912	使用者自行設定及清除
	SM1913~SM1951	每一掃描週期更新一次
	SM2000	使用者設定・傳送完畢後系統自動清除
	SM2001	EMDRW 指令執行時更新
	SM2002	EMDRW 指令執行時更新
	SM2003	EMDRW 指令執行時且有錯誤發生時更新
	SM2004	EMDRW 指令執行時且回應逾時時更新
	SM2005	EMDRW 指令執行時更新
	SM2006	使用者設定・傳送完畢後系統自動清除
	SM2007	EMDRW 指令執行時更新
	SM2008	EMDRW 指令執行時更新
	SM2009	EMDRW 指令執行時且有錯誤發生時更新
	SM2010	EMDRW 指令執行時且回應逾時時更新
	SM2011	EMDRW 指令執行時更新
	SM2012	使用者設定・傳送完畢後系統自動清除
	SM2013	EMDRW 指令執行時更新
	SM2014	EMDRW 指令執行時更新
	SM2015	EMDRW 指令執行時且有錯誤發生時更新
	SM2016	EMDRW 指令執行時且回應逾時時更新
	SM2017	EMDRW 指令執行時更新

2

適用機種	特殊輔助繼電器	更新時間
	SM2018	使用者設定・傳送完畢後系統自動清除
	SM2019	EMDRW 指令執行時更新
	SM2020	EMDRW 指令執行時更新
	SM2021	EMDRW 指令執行時且有錯誤發生時更新
	SM2022	EMDRW 指令執行時且回應逾時時更新
	SM2023	EMDRW 指令執行時更新
	SM2024	使用者設定・傳送完畢後系統自動清除
	SM2025	EMDRW 指令執行時更新
	SM2026	EMDRW 指令執行時更新
	SM2027	EMDRW 指令執行時且有錯誤發生時更新
	SM2028	EMDRW 指令執行時且回應逾時時更新
	SM2029	EMDRW 指令執行時更新
	SM2030	使用者設定・傳送完畢後系統自動清除
	SM2031	EMDRW 指令執行時更新
	SM2032	EMDRW 指令執行時更新
	SM2033	EMDRW 指令執行時且有錯誤發生時更新
	SM2034	EMDRW 指令執行時且回應逾時時更新
	SM2035	EMDRW 指令執行時更新
	SM2036	使用者設定・傳送完畢後系統自動清除
	SM2037	EMDRW 指令執行時更新
	SM2038	EMDRW 指令執行時更新
	SM2039	EMDRW 指令執行時且有錯誤發生時更新
	SM2040	EMDRW 指令執行時且回應逾時時更新
	SM2041	EMDRW 指令執行時更新
	SM2042	使用者設定・傳送完畢後系統自動清除
	SM2043	EMDRW 指令執行時更新
	SM2044	EMDRW 指令執行時更新
	SM2045	EMDRW 指令執行時且有錯誤發生時更新
	SM2046	EMDRW 指令執行時且回應逾時時更新
	SM2047	EMDRW 指令執行時更新
	SM2048 ↓ SM2319	每一掃描週期更新一次

註：上表欄位「適用機種」中，標示「AH5X0」者，表適用機種 AHCPU500/510/520/530；標示「AH5X1」者，表適用機種 AHCPU501/511/521/531；表格空白無任何標示者，表適用所有機種。

2.2.9 步進點繼電器 S

步進點繼電器的功能：

步進點繼電器 S 在工程自動化控制中可輕易的設定程序，其為步進階梯圖最基本的裝置，使用在步進階梯圖（或稱順序功能圖，Sequential Function Chart，SFC）中，SFC 使用說明請參考 ISPSOft 使用手冊。

步進點繼電器 S 的裝置編號為 S0~S2047 共 2048 點，各步進點繼電器 S 與輸出繼電器 Y 一樣有輸出線圈及 A、B 接點，而且於程式當中使用次數無限制，但無法直接驅動外部負載。步進繼電器（S）不用於步進階梯圖時，可當作一般的輔助繼電器使用。

2.2.10 計時器 T

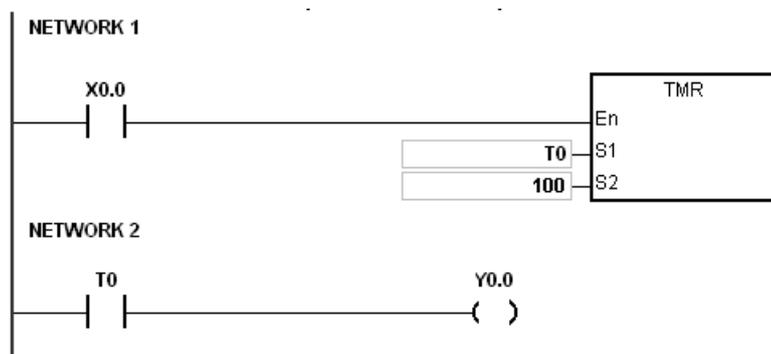
- 100ms 計時器：TMR 指令所指定之 T 計時器以 100ms 為單位計時
- 1ms 計時器：TMRH 指令所指定之 T 計時器以 1ms 為單位計時。
- 副程序專用計時器為 T1920~T2047。
- 積算型 T 計時器為 ST0~ST2047，但若要使用裝置監控，就是監控 T0~T2047。
- 在程式中同一個 T 計時器如果重覆使用（包含使用在不同指令 TMR、TMRH 中），則設定值以最快到達的為主。
- 在程式中同一個 T 計時器如果重覆使用，其中一個條件接點 OFF 時則 T 會 OFF。
- 在程式中同一個 T 計時器如果重覆使用為 T 與 ST，其中一個條件接點 OFF 時則 T 會 OFF。
- 當 T 計時器 ON->OFF 且條件式為 ON 時，T 計時值歸零並重新計時。
- 當 TMR 指令執行時，其所指定的計時器線圈受電，計時器開始計時，當到達所指定的定時值（計時值>=設定值），其接點動作如下：

NO (Normally Open) 接點	開路
NC (Normally Closed) 接點	閉合

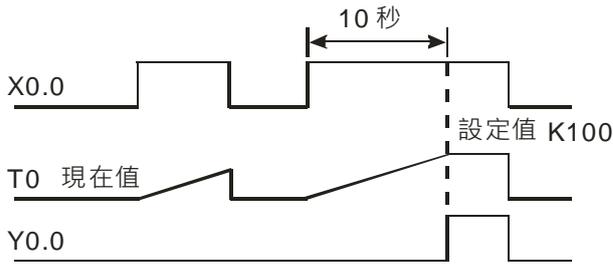
A. 一般用計時器 T

一般用計時器在 TMR 指令執行時計時一次，在 TMR 指令執行時，若計時到達，則輸出線圈導通。

- 當 X0.0=ON 時，計時器 T0 之現在值以 100ms 採上數計時，當計時器現在值=設定值 100 時，輸出線圈 T0=ON。
- 當 X0.0=OFF 或停電時，計時器 T0 之現在值清為 0，輸出線圈 T0 變為 OFF。



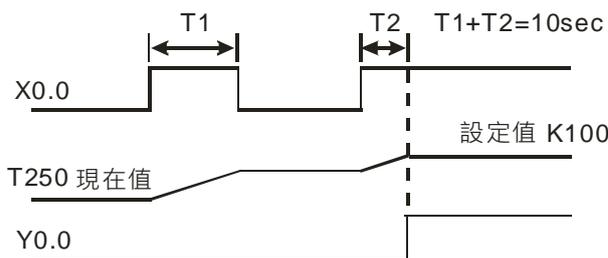
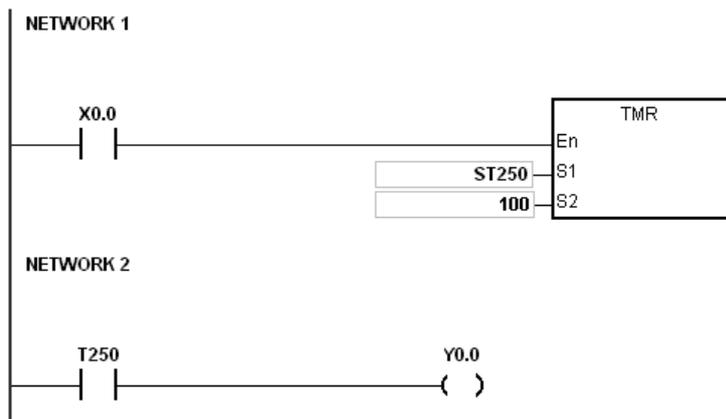
2



B. 積算型計時器 ST

積算型計時器在 TMR 指令執行時計時一次，在 TMR 指令執行時，若計時到達，則輸出線圈導通。只要在裝置 T 之前加上一個 S，就會變成積算型計時器 ST 裝置，表示目前的 T 變成積算型計時器，則條件接點 OFF 時積算型 T 的值不會被清除，條件接點=ON 的時候，T 由目前的值開始累積計時。

- 當 X0.0=ON 時，計時器 T250 之現在值以 100ms 採上數計時，當計時器現在值=設定值 100 時，輸出線圈 T250=ON。
- 當計時中若 X0.0=OFF 時，計時器 T250 暫停計時，現在值不變，待 X0.0 再 ON 時，繼續計時，其現在值往上累加直到計時器現在值=設定值 100 時，輸出線圈 T250=ON。



C. 功能塊 (Function Block) 用計時器 (T/ST)

功能塊或中斷插入中若使用到計時器時，請使用計時器 T1920~T2047。

功能塊用計時器於 TMR 指令或 END 指令執行時計時一次，在 TMR 指令或 END 指令執行時，若計時器現在值等於設定值，則輸出線圈導通。

一般用之計時器，若是使用在功能塊或中斷插入中而該功能塊不被執行時，計時器就無法正確的被計時。

2.2.11 計數器 C

16 位元計數器特點：

項目	16 位元計數器
類型	一般型
編號	C0~C2047
計數方向	上數
設定值	0~32,767
設定值的指定	常數或資料暫存器 D
現在值的變化	計數到達設定值就不再計數
輸出接點	計數到達設定值，接點導通並保持
復歸動作	RST 指令被執行時現在值歸零，接點被復歸成 OFF
接點動作	在掃描結束時，統一動作

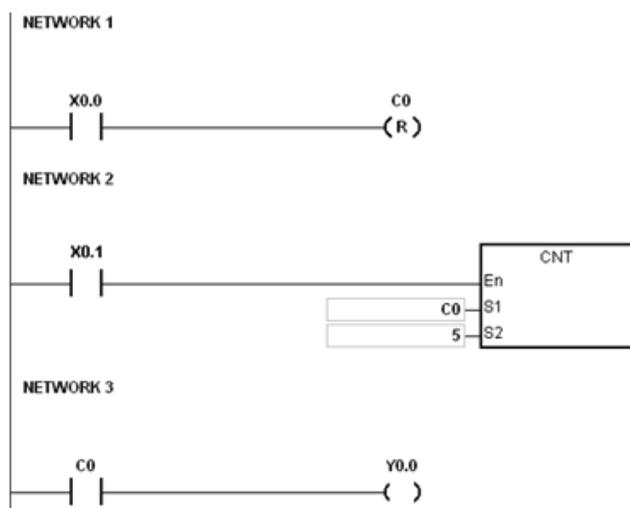
計數器的功能：

計數器之計數脈波輸入信號由 OFF→ON 時，計數器現在值等於設定值時輸出線圈導通，設定值為十進制常數值，亦可使用資料暫存器 D 當成設定值。

16 位元計數器：

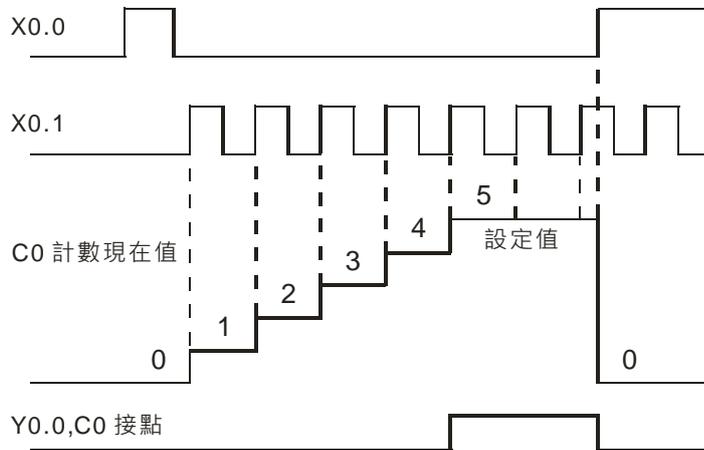
- 16 位元計數器的設定範圍：0~32,767。(0 與 1 相同，在第一次計數時輸出接點馬上導通。)
- 一般用計數器在 PLC 停電的時候，計數器現在值即被清除，若為停電保持型計數器會將停電前的現在值及計數器接點狀態記憶著，復電後會繼續累計。
- 若使用 MOV 指令、ISPSOft 將一個大於設定值的數值傳送到 C0 現在值暫存器時，在下次 X0.1 由 OFF→ON 時，C0 計數器接點即變成 ON，同時現在值內容變成與設定值相同。
- 計數器之設定值可使用常數直接設定或使用暫存器 D 中之數值作間接設定。
- 設定值可使用常數或使用資料暫存器 D 作為設定值可以是正負數。計數器現在值由 32,767 再往上累計時則變為-32,768。

範例：



2

1. 當 X0.0=ON 時 RST 指令被執行，C0 的現在值歸零，輸出接點被復歸為 OFF。
2. 當 X0.1 由 OFF→ON 時，計數器之現在值將執行上數（加一）的動作。
3. 當計數器 C0 計數到達設定值 5 時，C0 接點導通，C0 現在值 = 設定值=5。之後的 X0.1 觸發信號 C0 完全不接受，C0 現在值保持在 5 處。



2.2.12 32 位計數器 HC

32 位元計數器特點：

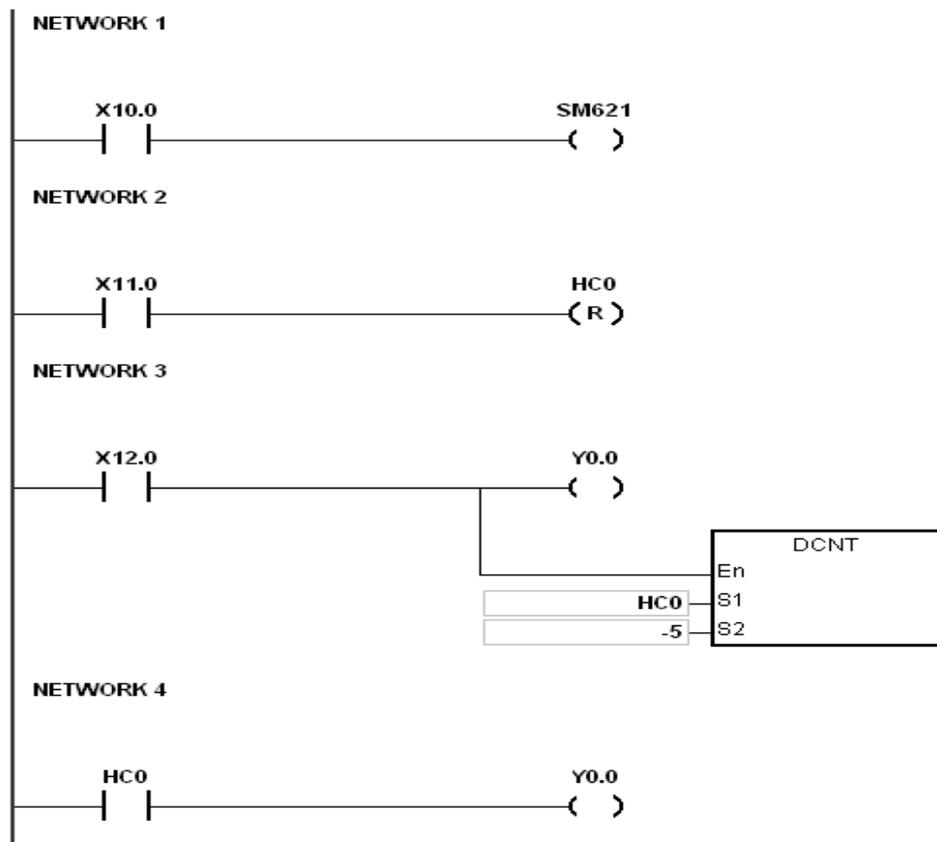
項目	32 位元計數器
類型	一般型
編號	HC0~HC63
計數方向	上、下數
設定值	-2,147,483,648~+2,147,483,647
設定值的指定	常數或資料暫存器 D (指定 2 個)
現在值的變化	計數到達設定值後，仍繼續計數
輸出接點	上數到達設定值接點導通並保持 ON 下數到達設定值接點復歸成 OFF
復歸動作	RST 指令被執行時現在值歸零，接點被復歸成 OFF
接點動作	在掃描結束時，統一動作

32 位元一般用加減算計數器：

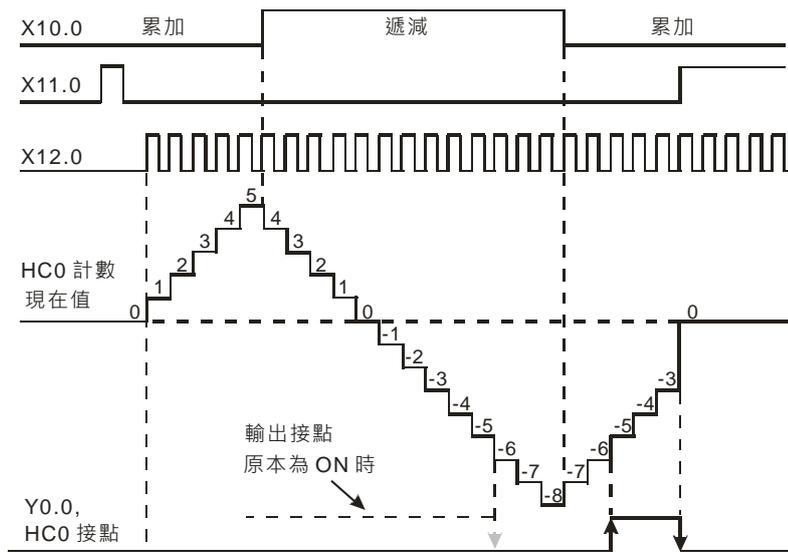
1. 32 位元一般用計數器的設定範圍：-2,147,483,648~2,147,483,647。
2. 32 位元一般用加減算計數器切換上下數用特殊輔助繼電器：由 SM621~SM684 來決定。例：SM621=OFF 時決定 HC0 為加算，SM621=ON 時決定 HC0 為減算其餘類推。
3. 設定值可使用常數或使用資料暫存器 D 作為設定值可以是正負數，若使用資料暫存器 D 則一個設定值佔用兩個連續的資料暫存器。
4. 一般用計數器在 PLC 停電的時候，計數器現在值即被清除，若為停電保持型計數器，則會將停電前的現在值及計數器接點狀態記憶著，復電後會繼續累計。

5. 計數器現在值由 2,147,483,647 再往上累計時則變為-2,147,483,648。同理計數器現在值由 -2,147,483,648 再往下遞減時，則變為 2,147,483,647。

範例：



1. X10.0 驅動 S621 來決定 HC0 為加算或減算。
2. 當 X11.0 由 OFF→ON 時，RST 指令執行，HC0 之現在值被清為 0，且接點變為 OFF。
3. 當 X12.0 由 OFF→ON 時，計數器之現在值將執行上數（加一）的動作或下數（減一）的動作。
4. 當計數器 HC0 之現在值從-6→-5 變化時，HC0 接點由 OFF→ON。當計數器 HC0 之現在值從 -5→-6 變化時，HC0 接點由 ON→OFF。
5. 若使用 MOV 指令、ISPSOft 將一個大於設定值的數值傳送到 HC0 現在值暫存器時，在下次 X12.0 由 OFF→ON 時，HC0 計數器接點即變成 ON，同時現在值內容變成與設定值相同。



2.2.13 資料暫存器 D

用於儲存數值資料，其資料長度為16位元（-32,768 ~ +32,767），最高位元為正負號，可儲存-32,768~+32,767之數值資料，亦可將兩個16位元暫存器合併成一個32位元暫存器（D+1，D編號小的為下16位元）使用，而其最高位元為正負號，可儲存-2,147,483,648~+2,147,483,647之數值資料。亦可將四個16位元暫存器合併成一個64位元暫存器（D+3，D+2，D+1，D編號小的為下16位元）使用，而其最高位元為正負號，可儲存-9,223,372,036,854,776~+9,223,372,036,854,775,807。也可用於與DIO之外的模組更新CR值之用，與模組更新CR值的D裝置配置設定請參考ISPSOft手冊的硬體組態說明。

暫存器依其性質可區分為下列二種：

1. 一般用暫存器：當 PLC 由 STOP→RUN 或斷電時，暫存器內的數值資料會被清除為 0，如果想要 PLC 由 STOP→RUN 時，資料會保持不被清除，請參考 ISPSOft 手冊的硬體組態說明，但斷電時仍會被清除為 0。
2. 停電保持用暫存器：當 PLC 斷電時此區域的暫存器資料不會被清除，仍保持其斷電前之數值。清除停電保持用暫存器的內容值，可使用 RST 或 ZRST 指令。

2.2.14 特殊資料暫存器 SR

每個特殊資料暫存器均有其特殊定義與用途，主要作為存放系統狀態、錯誤訊息、監視狀態之用。特殊資料暫存器（SR），它的種類及功能如下所示。在編號前有“”記號可參考SM/SR補充說明，像其中屬性欄中標示為“R”者，表示僅可作讀取的動作，若標示為“R/W”，表示可作讀寫的動作。另若標示為“-”，表示無變化。標示為“#”，則表示系統會依照PLC狀態作設定，使用者可讀取該設定值對照手冊之說明，可進一步了解系統資訊。

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬 性	出 廠 設 定
SR0	演算錯誤之偵錯號碼	0	0	0	0	0	0	0	0	-	R	0
SR1 ↓ SR2	32位元演算錯誤位址鎖定	0	0	0	0	0	0	0	0	-	R	0
SR4	文法檢查錯誤之偵錯號碼	0	0	0	0	0	0	0	0	-	R	0
SR5	32位元文法檢查錯誤的位址鎖定	0	0	0	0	0	0	0	0	-	R	0
SR6		0	0	0	0	0	0	0	0	-	R	0
SR8	WDT計時器ON的Step位址	0	0	0	0	0	0	0	-	-	R	0
SR24 ↓ SR25	前一次掃描週期同步裝置的資料量 (kbyte)	X	X	X	X	0	X	0	-	-	R	0
SR26 ↓ SR27	同步裝置的歷史最大資料量 (kbyte)	X	X	X	X	0	X	0	-	-	R	0
SR28 ↓ SR29	非週期性同步的執行時間 (ms)	X	X	X	X	0	X	0	-	-	R	0
SR30 ↓ SR31	週期性同步的執行時間 (ms)	X	X	X	X	0	X	0	-	-	R	0
*SR32	主電源模組狀態	V1. 08	V1. 08	0	0	0	0	-	-	-	R	0
*SR33	備援電源模組狀態	V1. 08	V1. 08	0	0	0	0	-	-	-	R	0
*SR40	錯誤紀錄 (Error Log) 的有效組數	0	0	0	0	0	0	-	-	-	R	0
*SR41	錯誤紀錄的指標	0	0	0	0	0	0	-	-	-	R	0
*SR42	第一組的錯誤紀錄之背版ID及插槽ID	0	0	0	0	0	0	-	-	-	R	0
*SR43	第一組的錯誤紀錄之模組ID	0	0	0	0	0	0	-	-	-	R	0
*SR44	第一組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR45	第一組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR46	第一組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR47	第一組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR48	第二組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR49	第二組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR50	第二組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR51	第二組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR52	第二組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR53	第二組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR54	第三組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR55	第三組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR56	第三組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR57	第三組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR58	第三組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR59	第三組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR60	第四組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR61	第四組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR62	第四組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR63	第四組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR64	第四組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR65	第四組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR66	第四組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
SR67	第五組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
SR68	第五組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
SR69	第五組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
SR70	第五組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
SR71	第五組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
SR72	第六組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR73	第六組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR74	第六組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR75	第六組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR76	第六組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR77	第六組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR78	第七組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR79	第七組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR80	第七組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR81	第七組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR82	第七組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR83	第七組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR84	第八組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR85	第八組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR86	第八組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR87	第八組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR88	第八組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR89	第八組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR90	第九組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR91	第九組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR92	第九組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR93	第九組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR94	第九組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR95	第九組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR96	第十組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR97	第十組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR98	第十組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR99	第十組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR100	第十組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR101	第十組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR102	第十一組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR103	第十一組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR104	第十一組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR105	第十一組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR106	第十一組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR107	第十一組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR108	第十二組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR109	第十二組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR110	第十二組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR111	第十二組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR112	第十二組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR113	第十二組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR114	第十三組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR115	第十三組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR116	第十三組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR117	第十三組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR118	第十三組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR119	第十三組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR120	第十三組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR121	第十四組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR122	第十四組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR123	第十四組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR124	第十四組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR125	第十四組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR126	第十五組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR127	第十五組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR128	第十五組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR129	第十五組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR130	第十五組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR131	第十五組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR132	第十六組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR133	第十六組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR134	第十六組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR135	第十六組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR136	第十六組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR137	第十六組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR138	第十七組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR139	第十七組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR140	第十七組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR141	第十七組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
SR142	第十七組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR143	第十七組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR144	第十八組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR145	第十八組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR146	第十八組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR147	第十八組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR148	第十八組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR149	第十八組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR150	第十九組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR151	第十九組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR152	第十九組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR153	第十九組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR154	第十九組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR155	第十九組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR156	第二十組的錯誤紀錄之背版 ID 及插槽 ID	0	0	0	0	0	0	-	-	-	R	0
*SR157	第二十組的錯誤紀錄之模組 ID	0	0	0	0	0	0	-	-	-	R	0
*SR158	第二十組的錯誤紀錄之錯誤代碼	0	0	0	0	0	0	-	-	-	R	0
*SR159	第二十組的錯誤紀錄之時間年及月	0	0	0	0	0	0	-	-	-	R	0
*SR160	第二十組的錯誤紀錄之時間日及時	0	0	0	0	0	0	-	-	-	R	0
*SR161	第二十組的錯誤紀錄之時間分及秒	0	0	0	0	0	0	-	-	-	R	0
*SR201	PLC COM1通訊位址	0	0	0	0	0	0	-	-	-	R/W	1
*SR202	PLC COM2通訊位址	0	X	0	X	X	X	-	-	-	R/W	3
*SR209	COM1通訊協定	0	0	0	0	0	0	-	-	-	R/W	16# 0024
*SR210	COM1通訊逾時時間 (單位: ms) , 設定為0則為不Time Out	0	0	0	0	0	0	3000 ms	-	-	R/W	3000 ms
*SR211	COM1命令重送次數	0	0	0	0	0	0	-	-	-	R/W	3
*SR212	COM2通訊協定	0	X	0	X	X	X	-	-	-	R/W	16# 0024
*SR213	COM2通訊逾時時間 (單位: ms) , 設定為0表不Time Out	0	X	0	X	X	X	3000	-	-	R/W	3000 ms
*SR214	COM2命令重送次數	0	0	0	0	X	0	-	-	-	R/W	3
*SR215	COM1介面代號	0	0	0	0	0	0	-	-	-	R/W	0
*SR216	COM2介面代號	0	X	0	X	X	X	-	-	-	R/W	0
*SR220	萬年曆 (RTC) 年 (西元) 00~99	0	0	0	0	0	0	-	-	-	R	0
*SR221	萬年曆 (RTC) 月01~12	0	0	0	0	0	0	-	-	-	R	1

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR222	萬年曆 (RTC) 日01~31	0	0	0	0	0	0	-	-	-	R	1
*SR223	萬年曆 (RTC) 時00~23	0	0	0	0	0	0	-	-	-	R	0
*SR224	萬年曆 (RTC) 分00~59	0	0	0	0	0	0	-	-	-	R	0
*SR225	萬年曆 (RTC) 秒00~59	0	0	0	0	0	0	-	-	-	R	0
*SR226	萬年曆 (RTC) 星期1~7	0	0	0	0	0	0	-	-	-	R	1
*SR227	儲存下載動作的有效組數 (最多紀錄20次)	0	0	0	0	0	0	-	-	-	R	0
*SR228	最新一次下載動作的紀錄指標。	0	0	0	0	0	0	-	-	-	R	0
*SR229	下載紀錄第1組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR230	下載紀錄第1組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR231	下載紀錄第1組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR232	下載紀錄第1組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR233	下載紀錄第2組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR234	下載紀錄第2組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR235	下載紀錄第2組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR236	下載紀錄第2組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR237	下載紀錄第3組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR238	下載紀錄第3組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR239	下載紀錄第3組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR240	下載紀錄第3組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR241	下載紀錄第4組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR242	下載紀錄第4組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR243	下載紀錄第4組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR244	下載紀錄第4組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR245	下載紀錄第5組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR246	下載紀錄第5組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR247	下載紀錄第5組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR248	下載紀錄第5組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR249	下載紀錄第6組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR250	下載紀錄第6組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR251	下載紀錄第6組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR252	下載紀錄第6組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR253	下載紀錄第7組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR254	下載紀錄第7組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR255	下載紀錄第7組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR256	下載紀錄第7組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR257	下載紀錄第8組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR258	下載紀錄第8組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR259	下載紀錄第8組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR260	下載紀錄第8組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR261	下載紀錄第9組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR262	下載紀錄第9組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR263	下載紀錄第9組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR264	下載紀錄第9組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR265	下載紀錄第10組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR266	下載紀錄第10組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR267	下載紀錄第10組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR268	下載紀錄第10組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR269	下載紀錄第11組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR270	下載紀錄第11組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR271	下載紀錄第11組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR272	下載紀錄第11組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR273	下載紀錄第12組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR274	下載紀錄第12組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR275	下載紀錄第12組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR276	下載紀錄第12組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR277	下載紀錄第13組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR278	下載紀錄第13組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR279	下載紀錄第13組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR280	下載紀錄第13組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR281	下載紀錄第14組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR282	下載紀錄第14組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR283	下載紀錄第14組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR284	下載紀錄第14組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR285	下載紀錄第15組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR286	下載紀錄第15組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR287	下載紀錄第15組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR288	下載紀錄第15組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR289	下載紀錄第16組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR290	下載紀錄第16組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR291	下載紀錄第16組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR292	下載紀錄第16組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR293	下載紀錄第17組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR294	下載紀錄第17組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR295	下載紀錄第17組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR296	下載紀錄第17組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR297	下載紀錄第18組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR298	下載紀錄第18組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR299	下載紀錄第18組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR300	下載紀錄第18組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR301	下載紀錄第19組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR302	下載紀錄第19組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR303	下載紀錄第19組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR304	下載紀錄第19組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR305	下載紀錄第20組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR306	下載紀錄第20組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR307	下載紀錄第20組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR308	下載紀錄第20組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR309	紀錄PLC狀態變更的總數（最大20組）	0	0	0	0	0	0	-	-	-	R	0
*SR310	紀錄PLC狀態變更最新一次的指標	0	0	0	0	0	0	-	-	-	R	0
*SR311	PLC狀態變更紀錄第1組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR312	PLC狀態變更紀錄第1組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR313	PLC狀態變更紀錄第1組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR314	PLC狀態變更紀錄第1組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR315	PLC狀態變更紀錄第2組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR316	PLC狀態變更紀錄第2組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR317	PLC狀態變更紀錄第2組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR318	PLC狀態變更紀錄第2組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR319	PLC狀態變更紀錄第3組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR320	PLC狀態變更紀錄第3組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR321	PLC狀態變更紀錄第3組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR322	PLC狀態變更紀錄第3組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR323	PLC狀態變更紀錄第4組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR324	PLC狀態變更紀錄第4組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR325	PLC狀態變更紀錄第4組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR326	PLC狀態變更紀錄第4組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR327	PLC狀態變更紀錄第5組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR328	PLC狀態變更紀錄第5組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR329	PLC狀態變更紀錄第5組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR330	PLC狀態變更紀錄第5組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR331	PLC狀態變更紀錄第6組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR332	PLC狀態變更紀錄第6組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR333	PLC狀態變更紀錄第6組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR334	PLC狀態變更紀錄第6組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR335	PLC狀態變更紀錄第7組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR336	PLC狀態變更紀錄第7組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR337	PLC狀態變更紀錄第7組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR338	PLC狀態變更紀錄第7組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR339	PLC狀態變更紀錄第8組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR340	PLC狀態變更紀錄第8組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR341	PLC狀態變更紀錄第8組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR342	PLC狀態變更紀錄第8組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR343	PLC狀態變更紀錄第9組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR344	PLC狀態變更紀錄第9組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR345	PLC狀態變更紀錄第9組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR346	PLC狀態變更紀錄第9組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR347	PLC狀態變更紀錄第10組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR348	PLC狀態變更紀錄第10組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR349	PLC狀態變更紀錄第10組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR350	PLC狀態變更紀錄第10組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR351	PLC狀態變更紀錄第11組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR352	PLC狀態變更紀錄第11組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR353	PLC狀態變更紀錄第11組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR354	PLC狀態變更紀錄第11組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR355	PLC狀態變更紀錄第12組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR356	PLC狀態變更紀錄第12組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR357	PLC狀態變更紀錄第12組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR358	PLC狀態變更紀錄第12組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR359	PLC狀態變更紀錄第13組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR360	PLC狀態變更紀錄第13組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR361	PLC狀態變更紀錄第13組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR362	PLC狀態變更紀錄第13組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR363	PLC狀態變更紀錄第14組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR364	PLC狀態變更紀錄第14組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR365	PLC狀態變更紀錄第14組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR366	PLC狀態變更紀錄第14組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR367	PLC狀態變更紀錄第15組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR368	PLC狀態變更紀錄第15組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR369	PLC狀態變更紀錄第15組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR370	PLC狀態變更紀錄第15組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR371	PLC狀態變更紀錄第16組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR372	PLC狀態變更紀錄第16組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR373	PLC狀態變更紀錄第16組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR374	PLC狀態變更紀錄第16組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR375	PLC狀態變更紀錄第17組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR376	PLC狀態變更紀錄第17組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0
*SR377	PLC狀態變更紀錄第17組，紀錄時間日時	○	○	○	○	○	○	-	-	-	R	0
*SR378	PLC狀態變更紀錄第17組，紀錄時間分秒	○	○	○	○	○	○	-	-	-	R	0
*SR379	PLC狀態變更紀錄第18組，紀錄動作編號	○	○	○	○	○	○	-	-	-	R	0
*SR380	PLC狀態變更紀錄第18組，紀錄時間年月	○	○	○	○	○	○	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR381	PLC狀態變更紀錄第18組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR382	PLC狀態變更紀錄第18組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR383	PLC狀態變更紀錄第19組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR384	PLC狀態變更紀錄第19組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR385	PLC狀態變更紀錄第19組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR386	PLC狀態變更紀錄第19組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR387	PLC狀態變更紀錄第20組，紀錄動作編號	0	0	0	0	0	0	-	-	-	R	0
*SR388	PLC狀態變更紀錄第20組，紀錄時間年月	0	0	0	0	0	0	-	-	-	R	0
*SR389	PLC狀態變更紀錄第20組，紀錄時間日時	0	0	0	0	0	0	-	-	-	R	0
*SR390	PLC狀態變更紀錄第20組，紀錄時間分秒	0	0	0	0	0	0	-	-	-	R	0
*SR391	萬年曆 (RTC) 年 (西元) 00~99	0	0	0	0	0	0	-	-	-	R	0
*SR392	萬年曆 (RTC) 月01~12	0	0	0	0	0	0	-	-	-	R	1
*SR393	萬年曆 (RTC) 日01~31	0	0	0	0	0	0	-	-	-	R	1
*SR394	萬年曆 (RTC) 時00~23	0	0	0	0	0	0	-	-	-	R	0
*SR395	萬年曆 (RTC) 分00~59	0	0	0	0	0	0	-	-	-	R	0
*SR396	萬年曆 (RTC) 秒00~59	0	0	0	0	0	0	-	-	-	R	0
*SR397	萬年曆 (RTC) 星期1~7	0	0	0	0	0	0	-	-	-	R	1
SR402	RUN後，每100微秒加1，重複計數由0~32767再由-32768~0	X	X	V1.01	V1.01	0	V1.01	0	0	-	R	0
SR404	RUN後，每1毫秒加1，重複計數由0~32767再由-32768~0	X	X	V1.01	V1.01	0	V1.01	0	0	-	R	0
SR407	RUN後，每秒加1，重複計數由0~32767再由-32768~0	0	0	0	0	0	0	0	0	-	R/W	0

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬 性	出 廠 設 定
SR408	RUN後，每完成一次掃描後加1，重複計數由0~32767再由-32768~0	○	○	○	○	○	○	0	0	-	R/W	0
*SR409	儲存數值n於2n秒時鐘，設定可為1到32767	○	○	○	○	○	○	-	-	-	R/W	30
*SR410	儲存數值n於2n毫秒時鐘	○	○	○	○	○	○	-	-	-	R/W	30
SR411	目前掃描時間儲存至SR411及SR412，測量單位為100微秒。SR411：儲存毫秒部分（儲存範圍：0~65535）SR412：	○	○	○	○	○	○	0	-	-	R	0
SR412	儲存毫秒部份。（儲存範圍：0~900）例：目前掃描時間為12.3毫秒，儲存以下值，SR411=12；SR412=300	○	○	○	○	○	○	0	-	-	R	0
SR413	儲存掃描時間最大值到SR413及SR414，測量單位為100微秒SR413：儲	○	○	○	○	○	○	0	-	-	R	0
SR414	存毫秒	○	○	○	○	○	○	0	-	-	R	0
SR415	儲存掃描時間最小值到SR415及SR416，測量單位為100微秒。SR415：	○	○	○	○	○	○	0	-	-	R	0
SR416	儲存毫秒	○	○	○	○	○	○	0	-	-	R	0
SR418	記錄主背板與第一塊Redundant延伸背板之間目前使用哪個port通訊	X	X	○	○	○	○	0	-	-	R	0
SR420	記錄目前第一塊Redundant延伸背板與下一塊背板之間使用那個port通訊	X	X	○	○	○	○	0	-	-	R	0
SR422	記錄目前第二塊Redundant延伸背板與下一塊背板之間使用那個port通訊	X	X	○	○	○	○	0	-	-	R	0
SR424	記錄目前第三塊Redundant延伸背板與下一塊背板之間使用那個port通訊	X	X	○	○	○	○	0	-	-	R	0
SR426	記錄目前第四塊Redundant延伸背板與下一塊背板之間使用那個port通訊	X	X	○	○	○	○	0	-	-	R	0
SR428	記錄目前第五塊Redundant延伸背板與下一塊背板之間使用那個port通訊	X	X	○	○	○	○	0	-	-	R	0
SR430	記錄目前第六塊Redundant延伸背板與下一塊背板之間使用那個port通訊	X	X	○	○	○	○	0	-	-	R	0
*SR440 ↓ *SR442	主機MAC值	○	○	○	○	○	○	MA C值	-	-	R	MAC 值

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SR443 ↓ *SR451	主機產品序號	O	O	O	O	O	O	序 號 值	-	-	R	序號 值
*SR453	當記憶卡有錯誤發生，錯誤代碼將被記錄	O	O	O	O	O	O	-	-	-	R	0
SR621	RS指令中斷字元 (COM1)	O	O	O	O	O	O	-	-	-	R/W	0
SR622	RS指令中斷字元 (COM2)	O	X	O	X	X	X	-	-	-	R/W	0
SR623	位元0~15：中斷程式I0~I15條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR624	位元0~15：中斷程式I16~I31條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR625	位元0~15：中斷程式I32~I47條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR626	位元0~15：中斷程式I48~I63條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR627	位元0~15：中斷程式I64~I79條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR628	位元0~15：中斷程式I80~I95條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR629	位元0~15：中斷程式I96~I111條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR630	位元0~15：中斷程式I112~I127條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR631	位元0~15：中斷程式I128~I143條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR632	位元0~15：中斷程式I144~I159條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR633	位元0~15：中斷程式I160~I175條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR634	位元0~15：中斷程式I176~I191條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF
SR635	位元0~15：中斷程式I192~I207條件，由IMASK指令設定	O	O	O	O	O	O	FFFF	-	-	R	FFFF

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SR636	位元0~15：中斷程式I208~I213條件，由IMASK指令設定	0	0	0	0	0	0	FFFF	-	-	R	FFFF
SR637	位元0~15：中斷程式I214~I229條件，由IMASK指令設定	0	0	0	0	0	0	FFFF	-	-	R	FFFF
SR638	位元0~15：中斷程式I230~I255條件，由IMASK指令設定	0	0	0	0	0	0	FFFF	-	-	R	FFFF
*SR655 ↓ SR662	紀錄背板1的I/O表映射錯誤或IO模組發生錯誤 ↓ 紀錄背板8的I/O表映射錯誤或IO模組發生錯誤	0	0	0	0	0	0	0	-	-	R	0
*SR663 ↓ SR674	紀錄背板1-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板1-插槽11的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR675 ↓ SR682	紀錄背板2-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板2-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR683 ↓ SR690	紀錄背板3-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板3-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR691 ↓ SR698	紀錄背板4-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板4-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR699 ↓ SR706	紀錄背板5-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板5-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SR707 ↓ SR714	紀錄背板6-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板6-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR715 ↓ SR722	紀錄背板7-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板7-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR723 ↓ SR730	紀錄背板8-插槽0的I/O表發生的映射錯誤碼 ↓ 紀錄背板8-插槽7的I/O表發生的映射錯誤碼	0	0	0	0	0	0	0	-	-	R	0
*SR731	主電源模組外部24V低電壓偵測	V1.08	V1.08	0	0	0	0	0	-	-	R	0
*SR732	備援電源模組外部24V低電壓偵測	V1.08	V1.08	0	0	0	0	0	-	-	R	0
SR900	DATA LOGGER已取樣筆數上位字元組	X	X	V2.01	V2.01	X	0	0	-	-	R	0
SR901	DATA LOGGER已取樣筆數下位字元組	X	X	V2.01	V2.01	X	0	0	-	-	R	0
SR902	DATA LOGGER SD卡動作代碼 · 須搭配SM456 (16#5AA5:將取樣資料寫入SD卡)	X	X	V2.01	V2.01	X	0	0	-	-	R/W	0
*SR1000	乙太網路IP位址上位字元組	X	0	X	0	0	0	-	-	-	R/W	C0A8
*SR1001	乙太網路IP位址下位字元組	X	0	X	0	0	0	-	-	-	R/W	0101
*SR1002	乙太網路網路遮罩位址上位字元組	X	0	X	0	0	0	-	-	-	R/W	FFFF
*SR1003	乙太網路網路遮罩位址下位字元組	X	0	X	0	0	0	-	-	-	R/W	0000
*SR1004	乙太網路閘道位址上位字元組	X	0	X	0	0	0	-	-	-	R/W	C0A8
*SR1005	乙太網路閘道位址下位字元組	X	0	X	0	0	0	-	-	-	R/W	0101
*SR1006	TCP保持連線時間	X	0	X	0	0	0	-	-	-	R/W	60
SR1007	乙太網路傳輸速度	X	0	X	0	0	0	0	-	-	R	0
SR1008	乙太網路傳輸模式	X	0	X	0	0	0	0	-	-	R	0
SR1100	輸入封包計數器上位字元組	X	0	X	0	0	0	0	-	-	R	0
SR1101	輸入封包計數器下位字元組	X	0	X	0	0	0	0	-	-	R	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
SR1102	輸入計數器上位字元組	X	O	X	O	O	O	0	-	-	R	0
SR1103	輸入計數器下位字元組	X	O	X	O	O	O	0	-	-	R	0
SR1104	輸出封包計數上位字元組	X	O	X	O	O	O	0	-	-	R	0
SR1105	輸出封包計數下位字元組	X	O	X	O	O	O	0	-	-	R	0
SR1106	輸出計數器上位字元組	X	O	X	O	O	O	0	-	-	R	0
SR1107	輸出計數器下位字元組	X	O	X	O	O	O	0	-	-	R	0
*SR1116	Email計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1117	Email錯誤計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1118	TCP Socket 1--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1119	TCP Socket 1--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1120	TCP Socket 1--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1121	TCP Socket 1--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1122	TCP Socket 1--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1123	TCP Socket 1--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1124	TCP Socket 1--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1125	TCP Socket 1--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1126	TCP Socket 1--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1127	TCP Socket 1--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1128	TCP Socket 1--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1129	TCP Socket 1--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1130	TCP Socket 1--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1131	TCP Socket 2--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1132	TCP Socket 2--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1133	TCP Socket 2--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1134	TCP Socket 2--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1135	TCP Socket 2--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1136	TCP Socket 2--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1137	TCP Socket 2--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1138	TCP Socket 2--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1139	TCP Socket 2--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1140	TCP Socket 2--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1141	TCP Socket 2--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1142	TCP Socket 2--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1143	TCP Socket 2--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1144	TCP Socket 3--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1145	TCP Socket 3--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1146	TCP Socket 3--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1147	TCP Socket 3--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1148	TCP Socket 3--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1149	TCP Socket 3--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1150	TCP Socket 3--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1151	TCP Socket 3--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1152	TCP Socket 3--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1153	TCP Socket 3--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1154	TCP Socket 3--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1155	TCP Socket 3--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1156	TCP Socket 3--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1157	TCP Socket 4--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1158	TCP Socket 4--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1159	TCP Socket 4--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1160	TCP Socket 4--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1161	TCP Socket 4--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1162	TCP Socket 4--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1163	TCP Socket 4--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1164	TCP Socket 4--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1165	TCP Socket 4--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1166	TCP Socket 4--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1167	TCP Socket 4--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1168	TCP Socket 4--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1169	TCP Socket 4--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1170	TCP Socket 5--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1171	TCP Socket 5--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1172	TCP Socket 5--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1173	TCP Socket 5--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1174	TCP Socket 5--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1175	TCP Socket 5--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1176	TCP Socket 5--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1177	TCP Socket 5--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1178	TCP Socket 5--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1179	TCP Socket 5--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1180	TCP Socket 5--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1181	TCP Socket 5--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1182	TCP Socket 5--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1183	TCP Socket 6--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1184	TCP Socket 6--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1185	TCP Socket 6--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1186	TCP Socket 6--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1187	TCP Socket 6--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1188	TCP Socket 6--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1189	TCP Socket 6--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1190	TCP Socket 6--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1191	TCP Socket 6--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1192	TCP Socket 6--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1193	TCP Socket 6--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1194	TCP Socket 6--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1195	TCP Socket 6--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1196	TCP Socket 7--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1197	TCP Socket 7--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1198	TCP Socket 7--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1199	TCP Socket 7--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1200	TCP Socket 7--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1201	TCP Socket 7--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1202	TCP Socket 7--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1203	TCP Socket 7--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1204	TCP Socket 7--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1205	TCP Socket 7--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1206	TCP Socket 7--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1207	TCP Socket 7--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1208	TCP Socket 7--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1209	TCP Socket 8--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1210	TCP Socket 8--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1211	TCP Socket 8--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1212	TCP Socket 8--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1213	TCP Socket 8--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1214	TCP Socket 8--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1215	TCP Socket 8--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1216	TCP Socket 8--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1217	TCP Socket 8--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1218	TCP Socket 8--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1219	TCP Socket 8--保持連線時間	X	O	X	O	O	O	-	-	-	R/W	1000
*SR1220	TCP Socket 8--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1221	TCP Socket 8--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1222	UDP Socket 1--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1223	UDP Socket 1--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1224	UDP Socket 1--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1225	UDP Socket 1--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1226	UDP Socket 1--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1227	UDP Socket 1--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1228	UDP Socket 1--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1229	UDP Socket 1--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1230	UDP Socket 1--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1231	UDP Socket 1--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1232	UDP Socket 1--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1233	UDP Socket 1--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1234	UDP Socket 2--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1235	UDP Socket 2--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1236	UDP Socket 2--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1237	UDP Socket 2--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1238	UDP Socket 2--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1239	UDP Socket 2--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1240	UDP Socket 2--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1241	UDP Socket 2--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1242	UDP Socket 2--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1243	UDP Socket 2--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1244	UDP Socket 2--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1245	UDP Socket 2--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1246	UDP Socket 3--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1247	UDP Socket 3--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1248	UDP Socket 3--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1249	UDP Socket 3--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1250	UDP Socket 3--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1251	UDP Socket 3--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1252	UDP Socket 3--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1253	UDP Socket 3--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1254	UDP Socket 3--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1255	UDP Socket 3--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1256	UDP Socket 3--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1257	UDP Socket 3--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1258	UDP Socket 4--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1259	UDP Socket 4--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1260	UDP Socket 4--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1261	UDP Socket 4--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1262	UDP Socket 4--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1263	UDP Socket 4--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1264	UDP Socket 4--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1265	UDP Socket 4--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1266	UDP Socket 4--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1267	UDP Socket 4--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1268	UDP Socket 4--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1269	UDP Socket 4--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1270	UDP Socket 5--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1271	UDP Socket 5--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1272	UDP Socket 5--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1273	UDP Socket 5--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1274	UDP Socket 5--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1275	UDP Socket 5--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1276	UDP Socket 5--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1277	UDP Socket 5--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1278	UDP Socket 5--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1279	UDP Socket 5--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1280	UDP Socket 5--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1281	UDP Socket 5--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1282	UDP Socket 6--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1283	UDP Socket 6--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1284	UDP Socket 6--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1285	UDP Socket 6--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1286	UDP Socket 6--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1287	UDP Socket 6--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1288	UDP Socket 6--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1289	UDP Socket 6--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1290	UDP Socket 6--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1291	UDP Socket 6--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1292	UDP Socket 6--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1293	UDP Socket 6--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1294	UDP Socket 7--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1295	UDP Socket 7--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1296	UDP Socket 7--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1297	UDP Socket 7--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1298	UDP Socket 7--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1299	UDP Socket 7--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1300	UDP Socket 7--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1301	UDP Socket 7--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1302	UDP Socket 7--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1303	UDP Socket 7--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1304	UDP Socket 7--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1305	UDP Socket 7--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1306	UDP Socket 8--本地通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1307	UDP Socket 8--遠端IP位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1308	UDP Socket 8--遠端IP位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1309	UDP Socket 8--遠端通訊埠	X	O	X	O	O	O	-	-	-	R/W	0
*SR1310	UDP Socket 8--傳送資料長度	X	O	X	O	O	O	-	-	-	R/W	0

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF	STOP	RUN	屬性	出廠 設定
								↓ ON	↓ RUN	↓ STOP		
*SR1311	UDP Socket 8--傳送資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1312	UDP Socket 8--傳送資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1313	UDP Socket 8--接收資料長度	X	O	X	O	O	O	-	-	-	R/W	0
*SR1314	UDP Socket 8--接收資料位址上位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1315	UDP Socket 8--接收資料位址下位字元組	X	O	X	O	O	O	-	-	-	R/W	0
*SR1316	UDP Socket 8--接收資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1317	UDP Socket 8--傳送資料計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1318	Socket輸入計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1319	Socket輸出計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1320	Socket錯誤計數器	X	O	X	O	O	O	0	-	-	R	0
*SR1329	主機背板ID	O	O	X	X	X	X	-	-	-	R/W	0
*SR1330	主機插槽ID	O	O	X	X	X	X	-	-	-	R/W	0
*SR1331	RTU號碼	O	O	X	X	X	X	-	-	-	R/W	0
*SR1332	延伸背板ID	O	O	X	X	X	X	-	-	-	R/W	0
*SR1333	延伸插槽ID	O	O	X	X	X	X	-	-	-	R/W	0
*SR1334	埠號	O	O	X	X	X	X	-	-	-	R/W	0
*SR1335	PLC Link週期時間	O	O	X	X	O	X	-	-	-	R	0
	COM1-MODBUS週期時間	X	X	O	O	O	O	-	-	-	R	0
*SR1336	PLC Link連線從站總數	O	O	X	X	O	X	-	-	-	R	0
	COM1-MODBUS連線從站總數	X	X	O	O	O	O	-	-	-	R	0
*SR1337	PLC Link資料交換時間	O	O	X	X	O	X	-	-	-	R/W	0
	COM1-MODBUS資料交換次數	X	X	O	O	O	O	-	-	-	R/W	0
*SR1338	PLC Link使用者定義限制時間	O	O	X	X	X	X	-	-	-	R/W	0
*SR1339	PLC Link傳送cmd間隔時間	O	O	X	X	X	X	-	-	-	R/W	1
	COM1-MODBUS傳送cmd間隔時間	X	X	O	O	O	O	-	-	-	R/W	1

2

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬 性	出 廠 設 定
*SR1340 ↓ SR1371	PLC Link從站1讀取裝置類別 ↓(0:暫存器,1:輸出線圈,其他:不 支援) PLC Link從站32讀取裝置類別 COM1-MODBUS從站1讀取裝置類別 ↓(0:暫存器,1:輸出線圈,其他:不 支援) COM1-MODBUS讀取裝置類別32	O	O	X	X	X	X	-	-	-	R/W	0
*SR1372 ↓ SR1403	PLC Link從站1寫入裝置類別 ↓(0:暫存器,1:輸出線圈,其他:不 支援) PLC Link從站32寫入裝置類別 COM1-MODBUS 從站 1 寫入裝置類別 ↓(0:暫存器,1:輸出線圈,其他:不 支援) COM1-MODBUS從站32寫入裝置類別	O	O	X	X	X	X	-	-	-	R/W	0
*SR1404 ↓ SR1467	PLC Link從站1讀取資料主站存放起始 位址 (SR1404·SR1405) ↓ PLC Link從站32讀取資料主站存放起始 位址 (SR1466·SR1467) COM1-MODBUS從站1讀取資料主站存 放起始位址 (SR1404·SR1405) ↓ COM1-MODBUS從站32讀取資料主站 存放起始位址 (SR1466·SR1467)	O	O	X	X	X	X	-	-	-	R/W	0
*SR1468 ↓ SR1531	PLC Link從站1寫入資料主站存放起始 位址 (SR1468·SR1469) ↓ PLC Link從站32寫入資料主站存放起始 位址 (SR1530·SR1531)	O	O	X	X	X	X	-	-	-	R/W	0

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出 廠 設 定
*SR1468 ↓ SR1531	COM1-MODBUS從站1寫入資料主站存放起始位址 (SR1468 · SR11469) ↓ COM1-MODBUS從站32寫入資料主站存放起始位址 (SR1530 · SR1531)	X	X	O	O	O	O	-	-	-	R/W	0
*SR1532 ↓ SR1595	PLC Link讀取從站1通訊起始位址設定 (SR1532 · SR1533) ↓ PLC Link讀取從站32通訊起始位址設定 (SR1594 · SR1595)	O	O	X	X	X	X	-	-	-	R/W	0
	COM1-MODBUS讀取從站1通訊起始位址設定 (SR1532 · SR1533) ↓ COM1-MODBUS讀取從站32通訊起始位址設定資料 (SR1594 · SR1595)	X	X	O	O	O	O	-	-	-	R/W	0
*SR1596 ↓ SR1659	PLC Link寫入從站1通訊起始位址設定 (SR1596 · SR1597) ↓ PLC Link寫入從站32通訊起始位址設定 (SR1658 · SR1659)	O	O	X	X	X	X	-	-	-	R/W	0
	COM1-MODBUS寫入從站1通訊起始位址設定 (SR1596 · SR1597) ↓ COM1-MODBUS寫入從站32通訊起始位址設定 (SR1658 · SR1659)	X	X	O	O	O	O	-	-	-	R/W	0
*SR1660 ↓ SR1691	PLC Link從站1讀取資料長度 ↓ PLC Link從站32讀取資料長度	O	O	X	X	X	X	-	-	-	R/W	0
	COM1-MODBUS從站1讀取資料長度 ↓ COM1-MODBUS從站32讀取資料長度	X	X	O	O	O	O	-	-	-	R/W	0
*SR1692 ↓ SR1723	PLC Link從站1寫入資料長度 ↓ PLC Link從站32寫入資料長度	O	O	X	X	X	X	-	-	-	R/W	0

2

特殊 暫存器 SR	功能說明	CPUSX0-RS2	CPUSX0-EN	CPUSX1-RS2	CPUSX1-EN	CPUS60-EN2	CPUS21-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
*SR1692 ↓ SR1723	COM1-MODBUS從站1寫入資料長度 ↓ COM1-MODBUS從站32寫入資料長度	X	X	O	O	O	O	-	-	-	R/W	0
*SR1724 ↓ SR1755	PLC Link從站1類別 ↓ PLC Link從站32類別	O	O	X	X	X	X	-	-	-	R/W	0
*SR1756 ↓ SR1787	PLC Link從站1 ID ↓ PLC Link從站32 ID ↓ COM1-MODBUS從站1 ID ↓ COM1-MODBUS從站32 ID	O	O	X	X	X	X	-	-	-	R/W	1 ↓ 32 1 ↓ 32
*SR1792 ↓ SR1823	Ether Link 區塊1 IP位址 (SR1792 · SR1793) ↓ Ether Link 區塊16 IP位址 (SR1822 · SR1823)	O	O	X	X	X	X	-	-	-	R	0
*SR1824 ↓ SR1855	Ether Link 區塊17 IP位址 (SR1824 · SR1825) ↓ Ether Link 區塊32 IP位址 (SR1854 · SR1855)	O *1	O *1	X	X	X	X	-	-	-	R	0
*SR1856 ↓ SR1919	Ether Link 區塊33 IP位址 (SR1856 · SR1857) ↓ Ether Link 區塊64 IP位址 (SR1918 · SR1919)	O *2	O *2	X	X	X	X	-	-	-	R	0
*SR1920 ↓ SR2047	Ether Link 區塊65 IP位址 (SR1920 · SR1921) ↓ Ether Link 區塊128IP位址 (SR2046 · SR2047)	O *3	O *3	X	X	X	X	-	-	-	R	0
SR2046	EtherNet/IP Adapter (從站) 連線數	X	X	X	V2. 03	V1. 02	X	-	-	-	R	0
SR2047	EtherNet/IP Scanner (主站) 連線數	X	X	X	V2. 03	V1. 02	X	-	-	-	R	0

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出 廠 設 定
SR2048 ↓ SR2079	EtherNet/IP Scanner (主站) 資料交換 連線 1 錯誤代碼 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 32 錯誤代碼	X	X	O	O	O	X	-	-	-	R	0
SR2080 ↓ SR2111	EtherNet/IP Scanner (主站) 資料交換 連線 33 錯誤代碼 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 64 錯誤代碼	X	X	O	O *6	O	X	-	-	-	R	0
SR2112 ↓ SR2175	EtherNet/IP Scanner (主站) 資料交換 連線 65 錯誤代碼 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 128 錯誤代碼	X	X	O	O *4	O	X	-	-	-	R	0
SR2176 ↓ SR2303	EtherNet/IP Scanner (主站) 資料交換 連線 129 錯誤代碼 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 256 錯誤代碼	X	X	O	O *5	O	X	-	-	-	R	0
SR2304 ↓ SR2335	EtherNet/IP Scanner (主站) 資料交換 連線 1 運行狀態 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 32 運行狀態	X	X	O	O	O	X	-	-	-	R	0
SR2336 ↓ SR2367	EtherNet/IP Scanner (主站) 資料交換 連線 33 運行狀態 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 64 運行狀態	X	X	O	O *6	O	X	-	-	-	R	0
SR2368 ↓ SR2431	EtherNet/IP Scanner (主站) 資料交換 連線 65 運行狀態 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 128 運行狀態	X	X	O	O *4	O	X	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SR2432 ↓ SR2559	EtherNet/IP Scanner (主站) 資料交換 連線 129 運行狀態 ↓ EtherNet/IP Scanner (主站) 資料交換 連線 256 運行狀態	X	X	O	O *5	O	X	-	-	-	R	0
SR2560	第1組的錯誤紀錄之RTU IP或ID上位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2561	第1組的錯誤紀錄之RTU IP或ID下位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2562	第1組的錯誤紀錄之RTU背版ID及插槽 ID	X	X	O	O	O	O	-	-	-	R	0
SR2563 ↓ SR2564	第1組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2565	第2組的錯誤紀錄之RTU IP或ID上位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2566	第2組的錯誤紀錄之RTU IP或ID下位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2567	第2組的錯誤紀錄之RTU背版ID及插槽 ID	X	X	O	O	O	O	-	-	-	R	0
SR2568 ↓ SR2569	第2組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2570	第3組的錯誤紀錄之RTU IP或ID上位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2571	第3組的錯誤紀錄之RTU IP或ID下位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2572	第3組的錯誤紀錄之RTU背版ID及插槽 ID	X	X	O	O	O	O	-	-	-	R	0
SR2573 ↓ SR2574	第3組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2575	第4組的錯誤紀錄之RTU IP或ID上位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2576	第4組的錯誤紀錄之RTU IP或ID下位字 元組	X	X	O	O	O	O	-	-	-	R	0
SR2577	第4組的錯誤紀錄之RTU背版ID及插槽 ID	X	X	O	O	O	O	-	-	-	R	0

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出 廠 設 定
SR2578 ↓ SR2579	第4組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2580	第5組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2581	第5組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2582	第5組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2583 ↓ SR2584	第5組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2585	第6組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2586	第6組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2587	第6組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2588 ↓ SR2589	第6組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2590	第7組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2591	第7組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2592	第7組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2593 ↓ SR2594	第7組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2595	第8組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2596	第8組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2597	第8組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SR2598 ↓ SR2599	第8組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2600	第9組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2601	第9組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2602	第9組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2603 ↓ SR2604	第9組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2605	第10組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2606	第10組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2607	第10組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2608 ↓ SR2609	第10組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2610	第11組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2611	第11組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2612	第11組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2613 ↓ SR2614	第11組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2615	第12組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2616	第12組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2617	第12組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2618 ↓ SR2619	第12組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0

特殊 暫存器 SR	功能說明	CP U5X0- RS2	CP U5X0- EN	CP U5X1- RS2	CP U5X1- EN	CP U560- EN2	CP U521- DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出 廠 設 定
SR2620	第13組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2621	第13組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2622	第13組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2623 ↓ SR2624	第13組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2625	第14組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2626	第14組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2627	第14組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2628 ↓ SR2629	第14組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2630	第15組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2631	第15組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2632	第15組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2633 ↓ SR2634	第15組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2635	第16組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2636	第16組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2637	第16組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2638 ↓ SR2639	第16組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0

2

特殊 暫存器 SR	功能說明	CPU5X0-RS2	CPU5X0-EN	CPU5X1-RS2	CPU5X1-EN	CPU560-EN2	CPU521-DNP3	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	屬性	出廠 設定
SR2640	第17組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2641	第17組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2642	第17組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2643 ↓ SR2644	第17組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2645	第18組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2646	第18組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2647	第18組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2648 ↓ SR2649	第18組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2650	第19組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2651	第19組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2652	第19組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2653 ↓ SR2654	第19組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0
SR2655	第20組的錯誤紀錄之RTU IP或ID上位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2656	第20組的錯誤紀錄之RTU IP或ID下位字元組	X	X	O	O	O	O	-	-	-	R	0
SR2657	第20組的錯誤紀錄之RTU背版ID及插槽ID	X	X	O	O	O	O	-	-	-	R	0
SR2658 ↓ SR2659	第20組的演算錯誤位址	X	X	O	O	O	O	-	-	-	R	0

註：*之 SR 請參考 SM/SR 補充說明

*1：僅支援機種 AHCPU530-EN、AHCPU520-EN、AHCPU510-EN、AHCPU530-RS2、AHCPU520-RS2、AHCPU510-RS2。

*2：僅支援機種 AHCPU530-EN、AHCPU520-EN、AHCPU530-RS2 及 AHCPU520-RS2。

*3：僅支援機種 AHCPU530-EN 及 AHCPU530-RS2。

*4：僅支援機種 AHCPU531-EN 及 AHCPU521-EN。

*5：僅支援機種 AHCPU531-EN。

*6：僅支援機種 AHCPU531-EN、AHCPU521-EN 及 AHCPU511-EN。

2.2.15 特殊資料暫存器 SR 的更新時間

適用機種	特殊資料暫存器	更新時間
	SR0~SR2	程式執行錯誤時更新
	SR4	語法檢查錯誤時更新
	SR5~SR6	程式下載至PLC時/電源開啟後第一次從Stop到Run時更新
	SR8	WDT發生時更新
	SR24~SR31	同步動作完成時更新
	SR32	主電源模組發生異常時更新
	SR33	備援電源模組發生異常時更新
	SR40~SR161	有錯誤發生時更新一次
	SR201~SR216	使用者自行設定及清除
	SR220~SR226	每一掃描週期更新一次
	SR227~SR308	程式下載至PLC時更新
	SR309~SR390	PLC狀態改變時更新
	SR391~SR397	每掃描週期更新
	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	每執行至End時更新
	SR409~SR410	使用者自行設定及清除
	SR411~SR416	每執行至End時更新
	SR440~SR451	上電時更新
	SR453	錯誤發生時更新
	SR621~SR622	使用者自行設定及清除
	SR623~SR638	IMASK指令執行時更新
	SR655~SR730	當I/O模組有錯誤時，會更新此區域
	SR731	當主電源模組的外部24V低電壓偵測，狀態改變時會更新此區域
	SR732	當備援電源模組的外部24V低電壓偵測，狀態改變時會更新此區域
	SR900	當DATA LOGGER紀錄裝置內容時更新此區域
	SR901	當DATA LOGGER紀錄裝置內容時更新此區域
	SR902	使用者自行設定及清除

2

適用機種	特殊資料暫存器	更新時間
	SR1000~SR1006	使用者自行設定及清除
	SR1007	上電與網路線連結時
	SR1008	上電與網路線連結時
	SR1100~SR1107	每一掃描週期更新一次
	SR1116~SR1128	參數下載至PLC時更新
	SR1129~SR1130	參數下載至PLC時或電源開啟時更新
	SR1131~SR1141	參數下載至PLC時更新
	SR1142~SR1143	參數下載至PLC時或電源開啟時更新
	SR1144~SR1154	參數下載至PLC時更新
	SR1155~SR1156	參數下載至PLC時或電源開啟時更新
	SR1157~SR1167	參數下載至PLC時更新
	SR1168~SR1169	參數下載至PLC時或電源開啟時更新
	SR1170~SR1180	參數下載至PLC時更新
	SR1181~SR1182	參數下載至PLC時或電源開啟時更新
	SR1183~SR1193	參數下載至PLC時更新
	SR1194~SR1195	參數下載至PLC時或電源開啟時更新
	SR1196~SR1206	參數下載至PLC時更新
	SR1207~SR1208	參數下載至PLC時或電源開啟時更新
	SR1209~SR1219	參數下載至PLC時更新
	SR1220~SR1221	參數下載至PLC時或電源開啟時更新
	SR1222~SR1231	參數下載至PLC時更新
	SR1232~SR1233	參數下載至PLC時或電源開啟時更新
	SR1234~SR1243	參數下載至PLC時更新
	SR1244~SR1245	參數下載至PLC時或電源開啟時更新
	SR1246~SR1255	參數下載至PLC時更新
	SR1256~SR1257	參數下載至PLC時或電源開啟時更新
	SR1258~SR1267	參數下載至PLC時更新
	SR1268~SR1269	參數下載至PLC時或電源開啟時更新
	SR1270~SR1279	參數下載至PLC時更新
	SR1280~SR1281	參數下載至PLC時或電源開啟時更新
	SR1282~SR1291	參數下載至PLC時更新
	SR1292~SR1293	參數下載至PLC時或電源開啟時更新
	SR1294~SR1303	參數下載至PLC時更新
	SR1304~SR1305	參數下載至PLC時或電源開啟時更新
	SR1306~SR1315	參數下載至PLC時更新
	SR1316~SR1320	參數下載至PLC時或電源開啟時更新

適用機種	特殊資料暫存器	更新時間
	SR1329~SR1334	使用者自行設定及清除
AH5X0	SR1335~SR1336	PLC Link啟動時，每一掃描週期更新一次。
AH5X1		COM1-MODBUS啟動時，每一掃描週期更新一次。
	SR1337~SR1787	使用者自行設定及清除
	SR1792~SR2559	每一掃描週期更新一次
	SR2560~SR2659	有錯誤發生時更新一次

註：上表欄位「適用機種」中，標示「AH5X0」者，表適用機種 AHCPU500/510/520/530；標示「AH5X1」者，表適用機種 AHCPU501/511/521/531；表格空白無任何標示者，表適用所有機種。

2.2.16 SM/SR 補充說明

1. 掃描逾時定時器

- SM8/SR8

當程式執行時發生掃描逾時 PLC ERROR 錯誤指示燈恆亮，此時 SM8=ON。

SR8 之內容值為 WDT 定時器 ON 之 STEP 位址。

2. 清除警告燈號

- SM22

當 SM22 為 ON 的時候，會將錯誤紀錄清除，以及清除警告的燈號。

3. 萬年曆

- SM220，SR220~SR226，SR391~SR397

SM220：萬年曆時鐘的±30 秒校正。當 SM220 OFF→ON 觸發時作校正。

若萬年曆時鐘秒為 0~29 秒時，分不動，秒歸 0。

若萬年曆時鐘秒為 30~59 秒時，分加 1，秒歸 0。

萬年曆時鐘 SR220~SR226，SR391~SR397 對應的功能以及內容值如下：

裝置		功能名稱	內容值範圍
BCD 碼	十進制		
SR220	SR391	年	0~99 (西元右兩位)
SR221	SR392	月	1~12
SR222	SR393	日	1~31
SR223	SR394	時	0~23
SR224	SR395	分	0~59
SR225	SR396	秒	0~59
SR226	SR397	星期	1~7

SR391~SR397 分別對應到 SR220~SR226，不同之處為 SR220~SR226 為 SR391~SR397 的 BCD 值，例如：12 月在 SR392 中為 12，在 SR221 中為 BCD 碼 12。
萬年曆時鐘的讀取及校正請參考 6.17 節萬年曆指令。

4. 通訊相關功能

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

SR215，SR216 用來紀錄 PLC COM 通訊介面的代號，代號代表的介面功能如下表：

代號	0	1	2
介面功能	RS-232	RS-485	RS-422

若 PLC COM 介面為 RS-485，RS-232，RS-422 時，SR209 紀錄 PLC COM1 的通訊格式，SR212 紀錄 PLC COM2 的通訊格式。通訊協定設定值設定方式如下表說明，其他有關 SM、SR 設定通訊的詳細說明請參考第 6.19 節通訊指令。

b0	資料長度			7 (內容值=0)		8 (內容值=1)	
b1 b2	同位元			00	:	無 (None)	
				01	:	奇同位 (Odd)	
				10	:	偶同位 (Even)	
b3	stop bits			1 bit (內容值=0)		2 bits (內容值=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 不支援	
	1000	(H 8)	:	460800		RS-232 不支援	
	1001	(H 9)	:	921600		RS-232 不支援	
b8~b15	無定義 (保留)						

5. 清除裝置內容旗標

- SM204/SM205

裝置編號	被清除的裝置
SM204 清除非停電保持區域	非停電保持的 X·Y·S·M·L 接點、 非停電保持的 T·C·HC 內容值及接點、 非停電保持的 D·E 內容值 清除裝置約需要 530ms 的時間，此時 WDT 不會動作。
SM205 清除停電保持區域	停電保持的 T·C·HC 內容值及接點、 停電保持的 M 接點 停電保持的 D 內容值 清除裝置約需要 30ms 的時間，此時 WDT 不會動作。

關於各裝置的停電保持區請參考停電保持區的裝置範圍。

6. PLC 錯誤紀錄 (Error Log)

- SR40~SR161

SR40：儲存錯誤紀錄 (Error Log) 的有效組數，最多 20 組，且每一組佔用 6 個暫存器。

SR41：錯誤紀錄的指標，會指向最新一組的錯誤紀錄。當錯誤發生時，錯誤紀錄指標增加 1。

指標的範圍值為 0~19，例如：SR41=3 表示第 4 組。

SR42~SR161：紀錄各組錯誤發生的時間及位置，SR 對應的功能說明如下：

組別	背板	插槽	模組 ID	錯誤代碼	錯誤發生時間					
					年	月	日	時	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 下載紀錄 (Download Log)

- SR227~SR308

SR227：儲存下載動作的有效組數，最多 20 組，且每一組佔用 4 個暫存器。目前會紀錄的下載動作編號為 1~3，如下表說明。

下載動作	動作編號
下載程式	1
下載 PLC 設定	2

下載動作	動作編號
下載 I/O 表	3

SR228：下載動作的紀錄指標 (Download Log pointer)，會指向最新一組的下載動作紀錄。每執行一次下載動作時，下載動作紀錄指標增加 1。指標的範圍值為 0~19，例如 SR228=3 表示第 4 組。

SR229~SR308：紀錄各組程式下載的時間及動作編號，SR 對應的功能說明如下。

2

組數	動作編號	*下載動作時間					
		年	月	日	時	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

*下載時間：時間資料的存為 BCD 格式。時間的內容值範圍如下：

功能名稱	內容值範圍
年	00~99 (西元右兩位)
月	01~12
日	01~31
時	00~23
分	00~59
秒	00~59

8. PLC 狀態變更紀錄 (PLC Status Log)

● SR309~SR390

SR309：紀錄 PLC 狀態變更 (PLC Status Log) 的有效組數，最多 20 組，且每一組佔用 4 個暫存器。目前會紀錄的 PLC 狀態變更編號為 1~6，如下表說明。

PLC 變更動作	動作編號
上電 (POWER ON)	1
斷電 (POWER OFF)	2
PLC 開始運轉 (PLC RUN)	3
PLC 停止運轉 (PLC STOP)	4
PLC 回歸出廠設定 (1.RST 鈕 ; 2.通訊命令)	5
PLC 主機按下 CLR 鈕 (清除停電保持裝置資料)	6

SR310：PLC 狀態變更的指標 (PLC Status Log Pointer)，會指向最新一組的 PLC 狀態變更的紀錄。當 PLC 的狀態被變更一次，紀錄指標增加 1。指標的範圍值為 0~19，例如 SR310=3 表示第 4 組。

SR311~SR390：紀錄各組 PLC 狀態變更發生的時間，SR 對應的功能說明如下：

組數	動作 編號	*PLC 變更時間					
		年	月	日	時	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

2

組數	動作 編號	*PLC 變更時間					
		年	月	日	時	分	秒
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
20	SR387	SR388 上	SR388 下	SR389 上	SR389 下	SR390 上	SR390 下

*PLC 變更時間：時間資料的存為 BCD 格式。時間的內容值範圍如下：

功能名稱	內容值範圍
年	00~99 (西元右兩位)
月	01~12
日	01~31
時	00~23
分	00~59
秒	00~59

9. PLC 運行旗標

- SM400~SM403

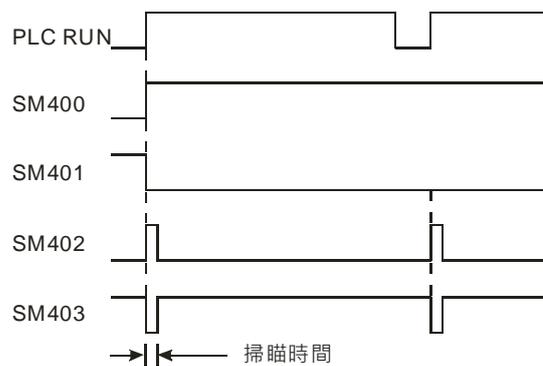
SM400：常開接點。



SM401：常閉接點。

SM402：PLC 開始 RUN 的第一次掃描 ON，之後保持為 OFF。該脈波的寬度為一次掃描時間，當要作各種初始設定工作時使用本接點。

SM403：PLC 開始 RUN 的第一次掃描 OFF，之後一直 ON。即啟始負向 (RUN 的瞬間“OFF”) 脈波。



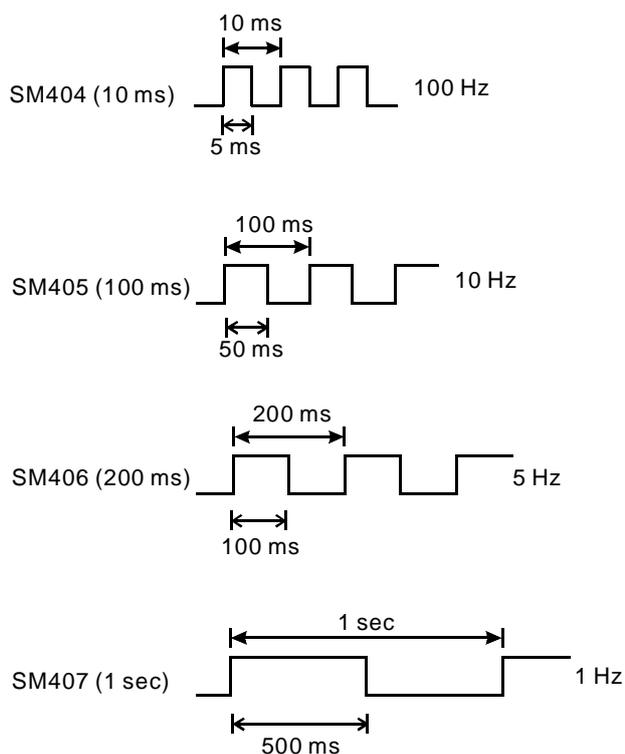
10. 內部時間脈衝

● SM404~SM410 · SR409~SR410

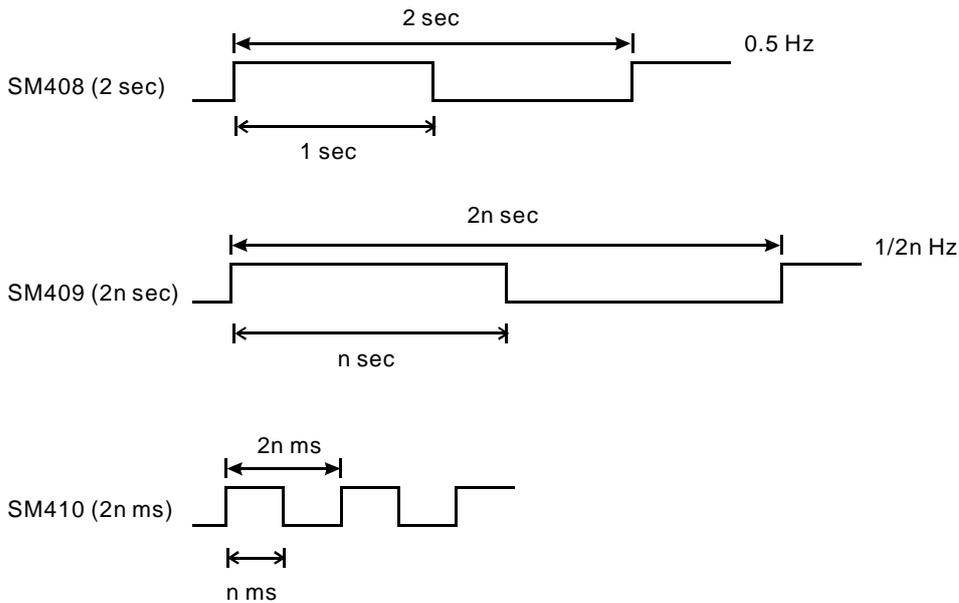
PLC 內部提供七種時鐘脈波。只要 PLC 通上電源，這七種時鐘脈波會自動動作，其中有兩個提供了使用者可以自行設定間隔時間的時鐘脈波。

裝置	功能說明
SM404	10ms 時鐘脈衝 · 5ms ON/5ms OFF
SM405	100ms 時鐘脈衝 · 50ms ON/50ms OFF
SM406	200ms 時鐘脈衝 · 100ms ON/100ms OFF
SM407	1s 時鐘脈衝 · 0.5s ON/0.5s OFF
SM408	2s 時鐘脈衝 · 1s ON/1s OFF
SM409	2n 秒時鐘脈衝 · n (秒) 開/n (秒) 關 · n 的時間間隔指定於 SR409
SM410	2n 毫秒時鐘脈衝 · n (毫秒) 開/n (毫秒) 關 · n 的時間間隔指定於 SR410

時鐘脈衝示意圖如下：



2



11. 記憶卡相關旗標

- SM450~SM453 · SR453

記憶卡主要提供使用者備份 PLC 內容。其對應的 SM · SR 功能如下說明，詳細使用方式請參考 6.24 節記憶卡讀寫指令。

裝置	功能說明
SM450	記憶卡是否存在旗標，ON：存在/OFF：不存在。
SM451	記憶卡防寫開關 (Protect Switch)，ON：防寫/OFF：無防寫。
SM452	記憶卡正被存取 (Accessed) 中，ON：存取中/OFF：無存取。
SM453	記憶卡運行中有錯誤發生，ON-表錯誤發生
SR453	當記憶卡有錯誤發生，錯誤代碼將被記錄。

12. I/O 模組相關旗標

- SR655~SR730 紀錄 I/O 表映射錯誤或 I/O 模組發生錯誤。

SR655~SR730：紀錄 I/O 表發生映射錯誤。

當 I/O 表發生映射錯誤時，屬於此模組的 SR 對應的位元會 ON，用以表示此模組發生錯誤。使用者可以讀取下列對應的 SR 值，知道哪些背板的插槽發生錯誤。例如：SR655 的 Bit5 為 ON 表示背板 1 的插槽 5 發生錯誤。

說明	主背板	延伸背板							
	背板 1	背板 2	背板 3	背板 4	背板 5	背板 6	背板 7	背板 8	
裝置	SR655	SR656	SR657	SR658	SR659	SR660	SR661	SR662	
插槽 0	Bit0								
插槽 1	Bit1								

說明	主背板	延伸背板						
	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 2	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2	Bit2
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：紀錄 I/O 表發生映射錯誤的錯誤代碼。

當 I/O 表發生映射錯誤，屬於此模組的 SR 會記錄錯誤碼，使用者可以讀取下列對應的裝置所紀錄錯誤碼，得知發生何種錯誤。各背板插槽對應的 SR 如下：

說明 背板 插槽	主背板	延伸背板						
	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. 乙太網路相關旗標

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

SM 編號	旗標說明	旗標作用
SM1001	Ethernet Port1 網路線連線狀態	OFF : Ethernet Port1 未連接 ON : Ethernet Port1 已連接
SM1002	Ethernet Port2 網路線連線狀態	OFF : Ethernet Port2 未連接 ON : Ethernet Port2 已連接
SM1003	光纖同步電纜連線狀態	OFF : 光纖同步電纜未連接 ON : 光纖同步電纜已連接
SM1089	MODBUS TCP Server 連線已滿	ON : MODBUS TCP Server 連線已滿
SM1090	TCP 連線忙碌	ON 時表示 TCP 連線逾時
SM1091	UDP 連線忙碌	ON 時表示 UDP 連線逾時
SM1106	乙太網路連線錯誤	OFF : 乙太網路自動協商成功 ON : 乙太網路自動協商失敗
SM1107	乙太網路基本設定錯誤	OFF : 基本設定正確 ON : 基本設定錯誤
SM1108	乙太網路限定通訊對象設置錯誤	OFF : 限定通訊對象設置正確 ON : 限定通訊對象設置錯誤
SM1109	TCP/UDP 本地埠已被使用	ON 時表示使用到同到的連接埠

關於旗標對應的 LED 燈狀態、錯誤代碼及其他詳細說明請參考 AH500 操作手冊第 12.2 節。

14. Email 發送的設定

- SM1112~SM1113、SM1116~SM1195

要發送 Email 前須先設定郵件的相關參數，若設定失敗 SM1112 會被設為 ON。若 Email 發送後失敗，則 Email 服務失敗旗標 SM1113 會被設定為 ON。

郵件發送條件 1 (Trigger1) ~ 郵件發送條件 8 (Trigger8) 的參數設定與觸發後的結果旗標 (SM1116~SM1195) 說明如下表：

項目	Trigger 1	Trigger 2	Trigger 3	Trigger 4	Trigger 5	Trigger 6	Trigger 7	Trigger 8
功能說明								
Email 觸發開關	SM1116	SM1126	SM1136	SM1146	SM1156	SM1166	SM1176	SM1186
	基本設定錯誤，則設為 ON							
Email 觸發旗標	SM1117	SM1127	SM1137	SM1147	SM1157	SM1167	SM1177	SM1187
	濾波器設定錯誤，則設為 ON							
Email 觸發狀態 0	SM1118	SM1128	SM1138	SM1148	SM1158	SM1168	SM1178	SM1188
	自從啟用本觸發條件後，無任何郵件被發送							

項目 功能 說明	Trigger 1	Trigger 2	Trigger 3	Trigger 4	Trigger 5	Trigger 6	Trigger 7	Trigger 8
Email 觸 發狀態 1	SM1119	SM1129	SM1139	SM1149	SM1159	SM1169	SM1179	SM1189
	觸發條件啟用且最後一封郵件已成功送出							
Email 觸 發狀態 2	SM1120	SM1130	SM1140	SM1150	SM1160	SM1170	SM1180	SM1190
	啟用本觸發條件且最近一封郵件發送錯誤							
Email 觸 發狀態 3	SM1121	SM1131	SM1141	SM1151	SM1161	SM1171	SM1181	SM1191
	觸發條件啟用且郵件已送出							
SMTP 回 覆逾時旗 標	SM1122	SM1132	SM1142	SM1152	SM1162	SM1172	SM1182	SM1192
	觸發條件被觸發且 SMTP 伺服器回應逾時							
SMTP 回 覆錯誤旗 標	SM1123	SM1133	SM1143	SM1153	SM1163	SM1173	SM1183	SM1193
	觸發條件被觸發且 SMTP 伺服器回應錯誤							
Email 附 件大小錯 誤旗標	SM1124	SM1134	SM1144	SM1154	SM1164	SM1174	SM1184	SM1194
	觸發條件被觸發且附件大小超過限制							
找不到附 件錯誤旗 標	SM1125	SM1135	SM1145	SM1155	SM1165	SM1175	SM1185	SM1195
	觸發條件被觸發且附件不存在							

關於旗標對應的 LED 燈狀態、錯誤代碼及其他詳細說明請參考 AH500 操作手冊第 12.2 節。

15. TCP/UDP Socket 設定

SR1118~SR1320, TCP/UDP Socket 設定，最多可分別設定 8 組 TCP 與 UDP Socket 通道。

SR1118~SR1221 為設定透過 TCP 協定進行資料交換之 Socket 通訊埠及資料範圍。

SR1222~SR1317 為設定透過 UDP 協定進行資料交換之 Socket 通訊埠及資料範圍。詳細 Socket 設定說明請參考 6.22 節乙太網控制指令。

16. PLC LINK 相關功能

- SM1392~SM1598 · SR1335~SR1787

PLC LINK 支援 PLC COM1，最多可連接 32 台從站，與 AH500 系列機種最多可以讀寫 450 筆 Word 或 7200 筆 Coil；與其它支援標準 MODBUS 機種最多可以讀寫 100 筆 Word 或 1600 筆 Coil。

2

	主站						
	從站 1		從站 2		...	從站 32	
	讀	寫	讀	寫	...	讀	寫
停電保持區	主站上的位址：將讀取到的資料寫到此位址裡 (SR1404 、 SR1405)	主站上的位址：將此位址裡的資料傳給從站 (SR1468 、 SR1469)	主站上的位址：將讀取到的資料寫到此位址裡 (SR1406 、 SR1407)	主站上的位址：將此位址裡的資料傳給從站 (SR1470 、 SR1471)	...	主站上的位址：將讀取到的資料寫到此位址裡 (SR1466 、 SR1467)	主站上的位址：將此位址裡的資料傳給從站 (SR1530 、 SR1531)
	從站上的位址：讀取裝置起始位址 (SR1532 、 SR1533)	從站上的位址：寫入裝置起始位址 (SR1596 、 SR1597)	從站上的位址：讀取裝置起始位址 (SR1534 、 SR1535)	從站上的位址：寫入裝置起始位址 (SR1598 、 SR1599)	...	從站上的位址：讀取裝置起始位址 (SR1594 、 SR1595)	從站上的位址：寫入裝置起始位址 (SR1658 、 SR1659)
	讀取裝置長度 (SR1660)	寫入裝置長度 (SR1692)	讀取裝置長度 (SR1661)	寫入裝置長度 (SR1693)	...	讀取裝置長度 (SR1691)	寫入裝置長度 (SR1723)
	裝置類別 (SR1340)	裝置類別 (SR1372)	裝置類別 (SR1341)	裝置類別 (SR1373)	...	裝置類別 (SR1371)	裝置類別 (SR1403)
	從站的機種型態 SR1724		從站的機種型態 SR1725		...	從站的機種型態 SR1755	
	從站站號 ID (SR1756)		從站站號 ID (SR1757)		...	從站站號 (SR1787)	
	從站站號 ID (SR1756)		從站站號 ID (SR1757)		...	從站站號 (SR1787)	
非停電保持區	鏈結旗標。觀看從站是否有鏈結 (SM1392)		鏈結旗標。觀看從站是否有鏈結 (SM1393)		...	鏈結旗標。觀看從站是否有鏈結 (SM1423)	
	資料交換旗標。主從站是否有在做資料交換的動作 (SM1424)		資料交換旗標。主從站是否有在做資料交換的動作 (SM1425)		...	資料交換旗標。主從站是否有在做資料交換的動作 (SM1455)	
非停電保持區	讀取錯誤旗標 (SM1456)	寫入錯誤旗標 (SM1488)	讀取錯誤旗標 (SM1457)	寫入錯誤旗標 (SM1489)	...	讀取錯誤旗標 (SM1487)	寫入錯誤旗標 (SM1519)
	讀取完成旗標 (ON->OFF) (SM1520)		讀取完成旗標 (ON->OFF) (SM1521)		...	讀取完成旗標 (ON->OFF) (SM1551)	
	寫入完成旗標 (ON->OFF) (SM1552)		寫入完成旗標 (ON->OFF) (SM1553)		...	寫入完成旗標 (ON->OFF) (SM1583)	

詳細說明請參考 AH500 操作手冊第 11.1 節 PLC Link。

17. Ether LINK 相關功能

端口	Ether Link 啟動旗標	Ether Link 運作錯誤旗標	Ether Link 運作狀態旗標
	OFF : 停止 ON : 啟動	OFF : 運作錯誤 ON : 運作正確	OFF : 停止中 ON : 執行中
CPU	SM1770	SM1788	SM1806
Port 0	SM1772	SM1790	SM1808
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

詳細說明請參考 AH500 操作手冊第 11.2 節 Ether Link。

18. IP 位址設定

- SR1792~SR2047

裝置	功能	說明
SR1792	區塊 1 IP 位址	區塊 1 連接的 IP 位址上 8 位元 Ex: 遠端位置為 192.168.1.100, 則此暫存器的值填 16#C0A8
SR1793	區塊 1 IP 位址	區塊 1 連接的 IP 位址下 8 位元 Ex: 遠端位置為 192.168.1.100, 則此暫存器的值填 16#0164
⋮		
SR2046	區塊 128 IP 位址	區塊 128 連接的 IP 位址上 8 位元 Ex: 遠端位置為 192.168.1.100, 則此暫存器的值填 16#C0A8
SR2047	區塊 128 IP 位址	區塊 128 連接的 IP 位址下 8 位元 Ex: 遠端位置為 192.168.1.100, 則此暫存器的值填 16#0164

Ether Link 部份之 SM、SR 的詳細說明請參考 AH500 操作手冊 11.2 節 Ether Link。

19. 電源模組相關旗標

- SM20、SR32~SR33、SR731、SR732

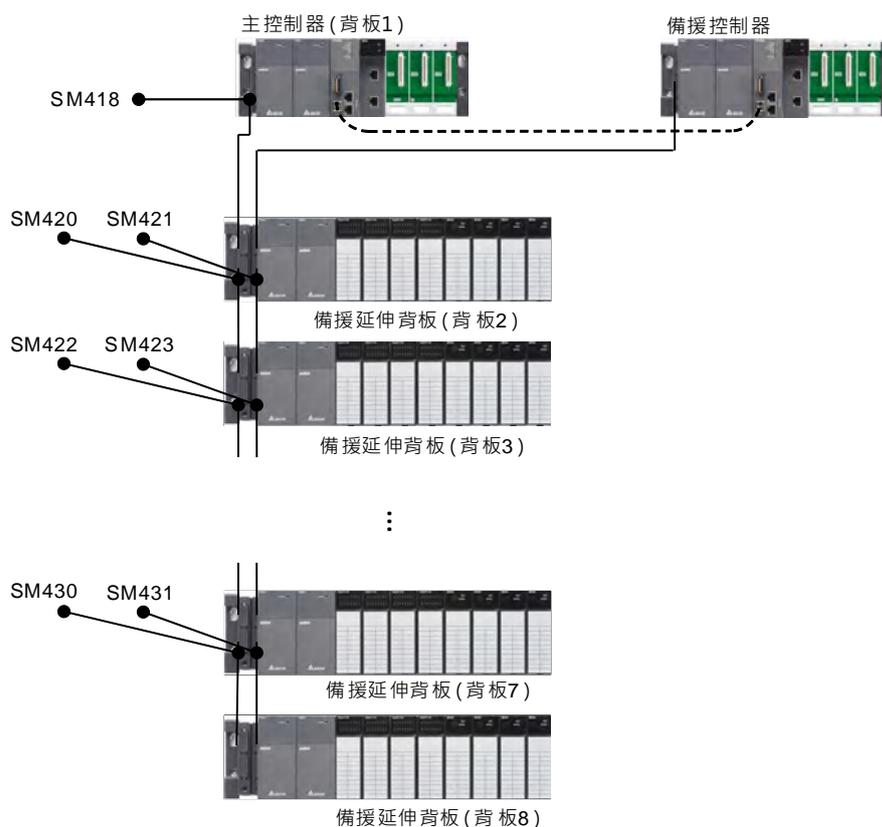
裝置	功能	說明
SM20	電源供應異常紀錄	曾經發生內部電源供應不足後，電源又回復到正常狀態時記錄。
SR32	左側電源模組狀態	ON：左側電源模組異常 OFF：左側電源模組正常運行
SR33	右側電源模組狀態	ON：右側電源模組異常 OFF：右側電源模組正常運行
SR731	左側電源模組外部 24V 低電壓偵測	ON：左側電源模組外部 24V 電源偵測點輸入異常 OFF：左側電源模組外部 24V 電源偵測點輸入正常
SR732	右側電源模組外部 24V 低電壓偵測	ON：右側電源模組外部 24V 電源偵測點輸入異常 OFF：右側電源模組外部 24V 電源偵測點輸入正常

說明		電源模組狀態		外部 24V 低電壓偵測	
		左側電源模組	右側電源模組	左側電源模組	右側電源模組
裝置		SR32	SR33	SR731	SR732
主背板	背板 1	Bit0	Bit0	Bit0	Bit0
	背板 2	Bit1	Bit1	Bit1	Bit1
	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. 備援延伸背板連線狀態

- SM418~SM431：備援背板延伸介面通訊埠狀態（ON：連線正常，OFF：連線異常）

說明		介面通訊埠狀態	
		Port2	Port4
(備援) 主背板	背板 1	SM418	-
	背板 2	SM420	SM421
備援延伸背板	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. 主機資訊相關旗標

- SR440~SR442、SR443~SR451
 SR440~SR442：主機 MAC Address。
 SR443~SR451：主機產品序號。

主機 MAC Address 00:18:23:10:F5:1A		主機產品序號 CPU52120W15480004	
裝置	內容	裝置	內容
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 主機會自動將 MAC Address 與產品序號以 ASCII Code 格式存入相對應的 SR 裝置。

2

22. EtherNet/IP 相關旗標

- SM2048 ~ SM2303 EtherNet/IP Scanner (主站) 資料交換連線 1~256 啟動旗標。
使用者可透過此旗標，啟動或關閉 AH 主機與 Adapter (從站) 的連線。
ON：啟動此連線資料交換。
OFF：關閉此連線資料交換。
- SM2304 ~ SM2311 EtherNet/IP Adapter (從站) I/O Connection 1~8 錯誤旗標。
使用者可透過此旗標，判斷 AH 主機與 Scanner (主站) 之間的連線是否異常。
ON：連線異常。
OFF：連線正常。
- SM2312 ~ SM2319 EtherNet/IP Adapter (從站) I/O Connection 1~8 連線旗標。
使用者可透過此旗標，判斷 AH 主機與 Scanner (主站) 之間的連線狀態。
ON：連線已建立。
OFF：無連線。
- SR2047 EtherNet/IP Scanner (主站) 連線數。
使用者可透過此旗標，得知 AH 主機與 Adapter (從站) 的連線數。
- SR2048 ~ SR2303：EtherNet/IP Scanner (主站) 資料交換連線 1~256 錯誤代碼。
其錯誤代碼代表意義，請參照「EtherNet/IP 操作手冊」章節「CH6 故障排除」。
- SR2304 ~ SR2559：EtherNet/IP Scanner (主站) 資料交換連線 1~256 運行狀態。
此 SR 暫存器內容值代表意義如下
0：代表此連線發生異常或未啟動，可透過相對應的暫存器 SR2048 ~ SR2303 來讀取此連線的錯誤狀態。
1：代表此連線正常運行。

2.2.17 連結暫存器 L

L裝置主要用於PLC Link或Ether Link的資料交換功能，當AH500對AH500進行資料交換時，可以使用L裝置作為資料交換的緩衝區，詳細說明請參考AH500操作手冊第11章便利功能使用說明。

連接暫存器L的裝置編號為L0~L65535共65536個Words (因機種不同，其裝置範圍有所差異)，也可當作一般的輔助暫存器使用。

2.2.18 間接指定暫存器 E

間接指定暫存器E是16位元的資料暫存器，跟一般的暫存器一樣可以被讀、寫，但主要功能是做間接指定暫存器使用，使用範圍為E0~E31。間接指定的使用方式，請參考第4.2節間接指定說明。

3

第3章 指令表

目錄

3.1	指令類型	3-2
3.1.1	基本指令	3-2
3.1.2	應用指令	3-2
3.2	指令表說明	3-3
3.2.1	基本指令	3-3
3.2.2	應用指令 (依 API 號碼排序)	3-4
3.2.3	應用指令 (依英文字母排序)	3-5
3.2.4	指令裝置表說明	3-6
3.3	基本指令一覽表	3-7
3.4	應用指令一覽表	3-9
3.4.1	應用指令 (依 API 號碼排序)	3-9
3.4.2	應用指令 (依英文字母排序)	3-42

3.1 指令類型

AH500 指令的主要類型包括順序基本指令、應用指令。

3.1.1 基本指令

分類	說明
接點指令	載入接點，串聯接點，並聯接點...等
連接指令	儲存或讀取接點間操作結果的連接
輸出指令	位元設備輸出，脈波輸出
主控制指令	控制共通串聯接點
上下緣檢出接點指令	觸發載入接點，串聯接點，並聯接點的相關指令
上下微分輸出指令	位元設備微分輸出
其他指令	其他指令

3.1.2 應用指令

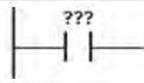
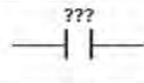
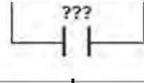
API	分類	說明
0000~0065	比較操作指令	比較操作如=、<>、>、>=、<、<=...等
0100~0118	四則運算指令	BIN 或 BCD 的加法、減法、乘法或除法
0200~0219	資料轉換指令	將 BCD 轉換成 BIN 和將 BIN 轉換成 BCD
0300~0310	資料轉移指令	傳送指定的資料
0400~0402	程式跳躍指令	程式跳躍
0500~0502	程式執行控制指令	允許或禁止中斷程式
0600	I/O 更新指令	運行局部更新
0700~0708	便利指令	用於以下目的的指令：計數器增加/減小，教學計時器，特殊功能計時器，旋轉臺最短距離控制...等
0800~0817	邏輯操作指令	邏輯操作，如邏輯加法，邏輯乘法等
0900~0904	旋轉指令	指定資料的旋轉移位
1000~1004	基本指令	計時器，計數器 指令
1100~1115	移位指令	指定資料的移位
1200~1223	資料處理指令	16 位元資料查詢，資料處理如解碼和編碼
1300~1302	結構建立指令	巢狀迴路，呼叫功能塊...等
1400~1401	模組的資料讀/寫指令	特殊功能模組的資料讀/寫
1500~1524	浮點數指令	浮點數運算指令
1600~1607	萬年曆指令	萬年曆時間（年，月，日，小時，分，秒和星期）的讀取、更新與比較
1700~1704	週邊設備指令	連接到週邊設備的 I/O
1800~1812	通訊指令	通訊控制周邊設備

API	分類	說明
1900~1905	其他指令	其他不符合以上範疇的指令，如通訊逾時重置指令和定時時鐘指令
2100~2121	字串處理指令	CBIN/BCD 和 ASCII 之間的轉換；BIN 和字串之間的轉換；浮點十進位數字和字串之間的轉換、字串處理等等
2200~2208	乙太網控制指令	控制乙太網資料交換相關指令
2300~2302	記憶卡讀寫指令	記憶卡讀寫相關指令
2400~2401	任務控制指令	程式中任務控制相關指令
2500~2502	SFC 控制指令	控制 SFC 相關指令
2900~2901	備援系統控制指令	備援系統相關指令

3

3.2 指令表說明

3.2.1 基本指令

指令碼	符號	功能	運算元
LD		載入/串聯/並聯 A 接點	DX, X, Y, M, S, T, C, HC, D, L, SM, PR
AND			
OR			

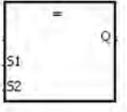
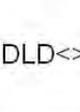
①
②
③
④

說明：

- ①：指令的名稱
- ②：在 ISPSOft 階梯圖所顯示的圖形
- ③：指令的功能用途說明
- ④：指令所支援的運算元

3.2.2 應用指令 (依 API 號碼排序)

3

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0000	LD=	DLD=	-		接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通
0001	LD<>	DLD<>	-		接點型態比較 S1 ≠ S2 導通 S1 = S2 不導通
0002	LD>	DLD>	-		接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通

①
②
③
④
⑤
⑥

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0018	FLD=	DFLD=	-		浮點數接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通

⑦

說明：

- ①：指令 API 編號。
- ②：指令的名稱。
- ③：16 位元的指令，如有支援 32 位元，則在指令前加 D，即為 32 位元指令。
- ④：是否有 PULSE 指令，有：✓。無：-。
有 PULSE 時，要使用 PULSE 指令，只需在指令後加 P。
- ⑤：在 ISPSOft 階梯圖所顯示的圖形。
- ⑥：指令的功能用途說明。
- ⑦：32 位元的浮點數指令，如有支援 64 位元，則在指令前加 D，即為 64 位元浮點數指令。

3.2.3 應用指令 (依英文字母排序)

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
G	0209	GBIN	DGBIN	—	✓	GRY 碼→BIN 變換
	0402	GOEND	—	—	—	跳躍到 END
	1902	GPWM	—	—	—	一般用脈波寬調變
	0208	GRY	DGRY	—	✓	BIN→GRY 碼變換
H	2104	HABIN	DHABIN	—	✓	十六進制 ASCII→十六進制 BIN 變換
	1701	HKY	DHKY	—	—	16 鍵鍵盤輸入
	1604	HOUR	DHOUR	—	—	運轉計時器

↓ ↓ ↓ ↓ ↓ ↓ ↓
 ① ② ③ ④ ⑤ ⑥ ⑦

說明：

①：指令名稱開頭英文字母的分類。

②：指令 API 編號。

③ ~ ⑤：指令的名稱。16 位元的指令，如有支援 32 位元，則在指令前加 D，即為 32 位元指令。

32 位元的浮點數指令，如有支援 64 位元，則在指令前加 D，即為 64 位元浮點數指令。

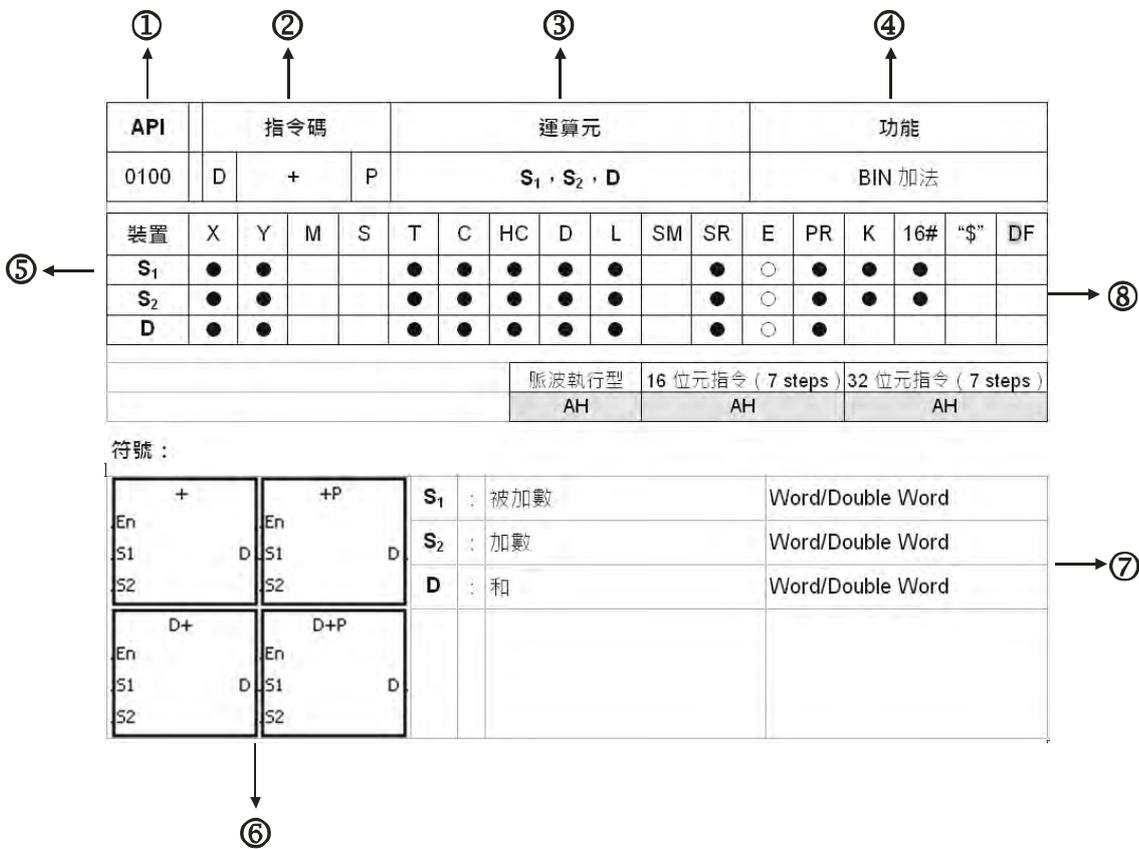
⑥：是否有脈波型指令 (P 指令)。有：✓。無：—。

要使用脈波型指令，需在指令後加 P。

⑦：指令的功能用途說明。

3.2.4 指令裝置表說明

3



說明：

- ①：指令 API 編號。
- ②：指令的名稱。16 位元的指令，如有支援 32 位元，則在指令前加 D，即為 32 位元指令。32 位元的浮點數指令，如有支援 64 位元，則在指令前加 D，即為 64 位浮點數元指令。有脈波執行型的指令則會在指令名稱後有 P。
- ③：指令的運算元。
- ④：指令的功能用途說明。
- ⑤：表示指令每個運算元所支援的裝置
 - 其中 K 表示為十進制，在 ISPSOft 中直接輸入數字，如：要輸入 30，在 ISPSOft 中需輸入 30。
 - 16#表示為十六進制，如：要輸入 30，在 ISPSOft 中需輸入 16#1E。
 - F/DF：浮點數的表示方式，但在 ISPSOft 中是直接以小數點的方式來表示，例如要輸入 F500 的浮點數，請直接輸入 500.0。
 - "\$":字串的表示方式，在 ISPSOft 中是以 " 方式來表示，例如要輸入字串 1234，請直接輸入 "1234"。
 - ：空心圓，表示此裝置不支援 E 修飾。
 - ：實心圓，表示此裝置支援 E 修飾。
- ⑥：接腳示意圖。

⑦：運算元的單位。

⑧：此指令之格式，是否有脈波執行型、16 位元、32 位元、64 位元等，及此指令的 Step 數。

3.3 基本指令一覽表

● 接點指令

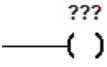
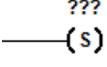
指令碼	符號	功能	運算元
LD		載入/串聯/並聯 A 接點	DX、X、Y、M、S、T、C、 HC、D、L、SM、PR
AND			
OR			
LDI		載入/串聯/並聯 B 接點	DX、X、Y、M、S、T、C、 HC、D、L、SM、PR
ANI			
ORI			

● 連接指令

指令碼	符號	功能	運算元
ANB		串聯迴路方塊	—
ORB		並聯迴路方塊	—
MPS	—	存入堆疊指令	—
MRD	—	堆疊讀取指標不動	—
MPP	—	讀出堆疊	—

3

● 輸出指令

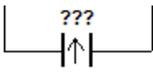
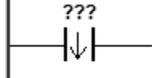
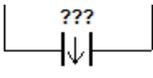
指令碼	符號	功能	執行條件	運算元
OUT		驅動線圈		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR
SET		動作保持 (ON)		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR

● 主控制指令

3

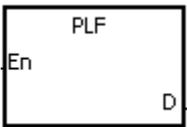
指令碼	符號	功能	運算元
MC		共通串聯接點之連結	N
MCR		共通串聯接點之解除	N

● 上下緣檢出接點指令

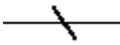
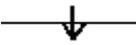
指令碼	符號	功能	執行條件	運算元
LDP		正緣檢出動作開始/串聯/並聯連接		DX、X、Y、M、S、T、C、 HC、D、L、SM、PR
PED				
ANDP				
APED		負緣檢出動作開始/串聯/並聯連接		DX、X、Y、M、S、T、C、 HC、D、W、L、SM、PR
ORP				
OPED				
LDF		負緣檢出動作開始/串聯/並聯連接		DX、X、Y、M、S、T、C、 HC、D、W、L、SM、PR
NED				
ANED				
ORF		負緣檢出動作開始/串聯/並聯連接		DX、X、Y、M、S、T、C、 HC、D、W、L、SM、PR
ONED				

● 上下微分輸出指令

指令碼	符號	功能	執行條件	運算元
PLS		上微分輸出		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR

指令碼	符號	功能	執行條件	運算元
PLF		下微分輸出		DY、X、Y、M、S、T、C、 HC、D、L、SM、PR

- 其他指令

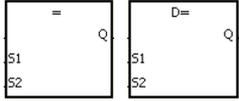
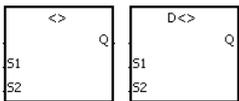
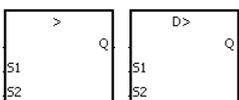
指令碼	符號	功能	運算元
INV		運算結果反相	-
NOP	-	無動作	-
PSTOP		PLC 程式停止執行	-
NP		上升緣觸發指令	-
PN		下降緣觸發指令	-
FB_NP		上升緣觸發指令	S
FB_PN		下降緣觸發指令	S

3

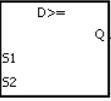
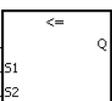
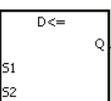
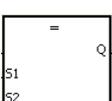
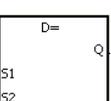
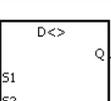
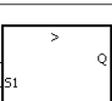
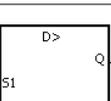
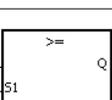
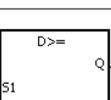
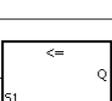
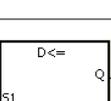
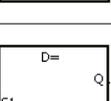
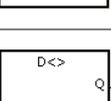
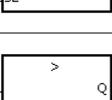
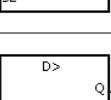
3.4 應用指令一覽表

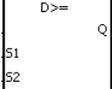
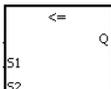
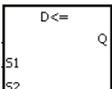
3.4.1 應用指令 (依 API 號碼排序)

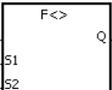
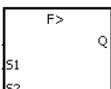
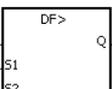
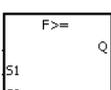
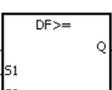
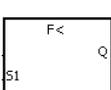
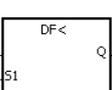
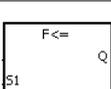
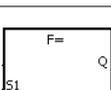
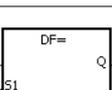
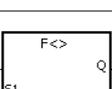
- 比較操作指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0000	LD=	DLD=	-		接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通
0001	LD<>	DLD<>	-		接點型態比較 S1 ≠ S2 導通 S1 = S2 不導通
0002	LD>	DLD>	-		接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通

3

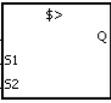
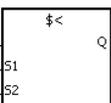
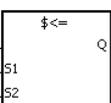
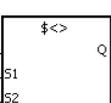
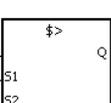
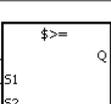
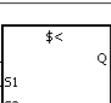
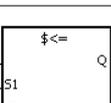
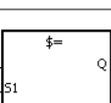
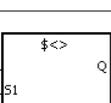
API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
0003	LD>=	DLD>=	-			接點型態比較 S1 ≥ S2 導通 S1 < S2 不導通
0004	LD<	DLD<	-			接點型態比較 S1 < S2 導通 S1 ≥ S2 不導通
0005	LD<=	DLD<=	-			接點型態比較 S1 ≤ S2 導通 S1 > S2 不導通
0006	AND=	DAND=	-			串連接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通
0007	AND<>	DAND<>	-			串連接點型態比較 S1 ≠ S2 導通 S1 = S2 不導通
0008	AND>	DAND>	-			串連接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通
0009	AND>=	DAND>=	-			串連接點型態比較 S1 ≥ S2 導通 S1 < S2 不導通
0010	AND<	DAND<	-			串連接點型態比較 S1 < S2 導通 S1 ≥ S2 不導通
0011	AND<=	DAND<=	-			串連接點型態比較 S1 ≤ S2 導通 S1 > S2 不導通
0012	OR=	DOR=	-			並聯接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通
0013	OR<>	DOR<>	-			並聯接點型態比較 S1 ≠ S2 導通 S1 = S2 不導通
0014	OR>	DOR>	-			並聯接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0015	OR>=	DOR>=	—	 	並聯接點型態比較 $S1 \geq S2$ 導通 $S1 < S2$ 不導通
0016	OR<	DOR<	—	 	並聯接點型態比較 $S1 < S2$ 導通 $S1 \geq S2$ 不導通
0017	OR<=	DOR<=	—	 	並聯接點型態比較 $S1 \leq S2$ 導通 $S1 > S2$ 不導通

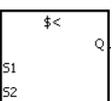
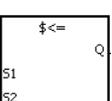
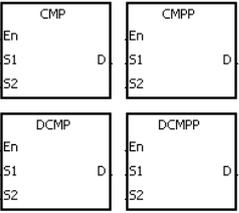
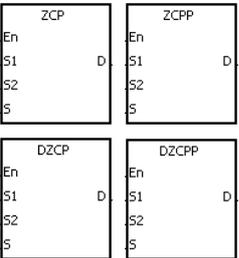
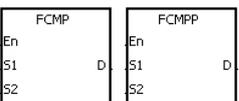
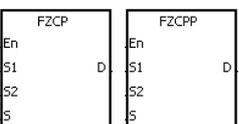
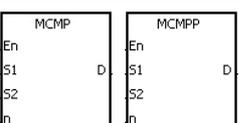
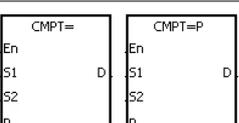
API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0018	FLD=	DFLD=	—	 	浮點數接點型態比較 $S1 = S2$ 導通 $S1 \neq S2$ 不導通
0019	FLD<>	DFLD<>	—	 	浮點數接點型態比較 $S1 \neq S2$ 導通 $S1 = S2$ 不導通
0020	FLD>	DFLD>	—	 	浮點數接點型態比較 $S1 > S2$ 導通 $S1 \leq S2$ 不導通
0021	FLD>=	DFLD>=	—	 	浮點數接點型態比較 $S1 \geq S2$ 導通 $S1 < S2$ 不導通
0022	FLD<	DFLD<	—	 	浮點數接點型態比較 $S1 < S2$ 導通 $S1 \geq S2$ 不導通
0023	FLD<=	DFLD<=	—	 	浮點數接點型態比較 $S1 \leq S2$ 導通 $S1 > S2$ 不導通
0024	FAND=	DFAND=	—	 	浮點數串連接點型態比較 $S1 = S2$ 導通 $S1 \neq S2$ 不導通
0025	FAND<>	DFAND<>	—	 	浮點數串連接點型態比較 $S1 \neq S2$ 導通 $S1 = S2$ 不導通

3

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0026	FAND>	DFAND>	-		浮點數串連接點型態比較 $S1 > S2$ 導通 $S1 \leq S2$ 不導通
0027	FAND>=	DFAND>=	-		浮點數串連接點型態比較 $S1 \geq S2$ 導通 $S1 < S2$ 不導通
0028	FAND<	DFAND<	-		浮點數串連接點型態比較 $S1 < S2$ 導通 $S1 \geq S2$ 不導通
0029	FAND<=	DFAND<=	-		浮點數串連接點型態比較 $S1 \leq S2$ 導通 $S1 > S2$ 不導通
0030	FOR=	DFOR=	-		浮點數並連接點型態比較 $S1 = S2$ 導通 $S1 \neq S2$ 不導通
0031	FOR<>	DFOR<>	-		浮點數並連接點型態比較 $S1 \neq S2$ 導通 $S1 = S2$ 不導通
0032	FOR>	DFOR>	-		浮點數並連接點型態比較 $S1 > S2$ 導通 $S1 \leq S2$ 不導通
0033	FOR>=	DFOR>=	-		浮點數並連接點型態比較 $S1 \geq S2$ 導通 $S1 < S2$ 不導通
0034	FOR<	DFOR<	-		浮點數並連接點型態比較 $S1 < S2$ 導通 $S1 \geq S2$ 不導通
0035	FOR<=	DFOR<=	-		浮點數並連接點型態比較 $S1 \leq S2$ 導通 $S1 > S2$ 不導通
0036	LD\$=	-	-		字串接點型態比較 $S1 = S2$ 導通 $S1 \neq S2$ 不導通
0037	LD\$<>	-	-		字串接點型態比較 $S1 \neq S2$ 導通 $S1 = S2$ 不導通

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0038	LD\$>	—	—		字串接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通
0039	LD\$>=	—	—		字串接點型態比較 S1 ≥ S2 導通 S1 < S2 不導通
0040	LD\$<	—	—		字串接點型態比較 S1 < S2 導通 S1 ≥ S2 不導通
0041	LD\$<=	—	—		字串接點型態比較 S1 ≤ S2 導通 S1 > S2 不導通
0042	AND\$=	—	—		字串串連接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通
0043	AND\$<>	—	—		字串串連接點型態比較 S1 ≠ S2 導通 S1 = S2 不導通
0044	AND\$>	—	—		字串串連接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通
0045	AND\$>=	—	—		字串串連接點型態比較 S1 ≥ S2 導通 S1 < S2 不導通
0046	AND\$<	—	—		字串串連接點型態比較 S1 < S2 導通 S1 ≥ S2 不導通
0047	AND\$<=	—	—		字串串連接點型態比較 S1 ≤ S2 導通 S1 > S2 不導通
0048	OR\$=	—	—		字串並連接點型態比較 S1 = S2 導通 S1 ≠ S2 不導通
0049	OR\$<>	—	—		字串並連接點型態比較 S1 ≠ S2 導通 S1 = S2 不導通

3

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0050	OR\$>	—	—		字串並連接點型態比較 S1 > S2 導通 S1 ≤ S2 不導通
0051	OR\$>=	—	—		字串並連接點型態比較 S1 ≥ S2 導通 S1 < S2 不導通
0052	OR\$<	—	—		字串並連接點型態比較 S1 < S2 導通 S1 ≥ S2 不導通
0053	OR\$<=	—	—		字串並連接點型態比較 S1 ≤ S2 導通 S1 > S2 不導通
0054	CMP	DCMP	✓		比較設定輸出
0055	ZCP	DZCP	✓		區域比較
0056	—	FCMP	✓		浮點數比較
0057	—	FZCP	✓		浮點數區域比較
0058	MCMP	—	✓		矩陣比較
0059	CMPT=	—	✓		表格比較，=

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0060	CMPT<>	—	✓		表格比較，≠
0061	CMPT>	—	✓		表格比較，>
0062	CMPT>=	—	✓		表格比較，≥
0063	CMPT<	—	✓		表格比較，<
0064	CMPT<=	—	✓		表格比較，≤
0065	CHKADR	—	—		接點型態指標暫存器位址檢查

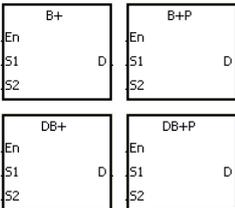
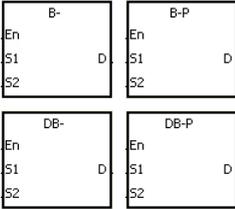
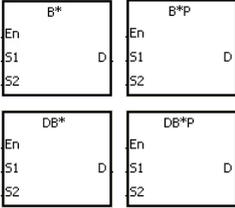
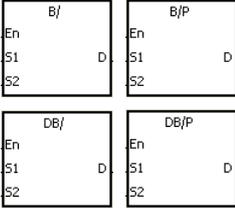
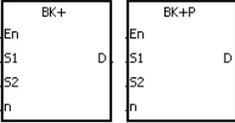
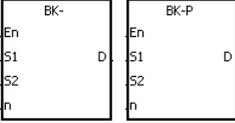
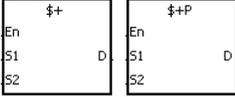
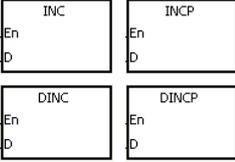
● 四則運算指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0100	+	D+	✓		BIN 加法 $S1+S2=D$
0101	-	D-	✓		BIN 減法 $S1-S2=D$

3

API	指令碼		P 指令	符號	功能																								
	16 位元	32 位元																											
0102	*	D*	✓	<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> <table border="1"> <tr> <td>En</td> <td>D*</td> <td>En</td> <td>D*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		En	D*	En	D*P	S1	D	S1	D	S2		S2		BIN 乘法 $S1 * S2 = D$
En	*	En	*P																										
S1	D	S1	D																										
S2		S2																											
En	D*	En	D*P																										
S1	D	S1	D																										
S2		S2																											
0103	/	D/	✓	<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> <table border="1"> <tr> <td>En</td> <td>D/</td> <td>En</td> <td>D/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		En	D/	En	D/P	S1	D	S1	D	S2		S2		BIN 除法 $S1 / S2 = D$
En	/	En	/P																										
S1	D	S1	D																										
S2		S2																											
En	D/	En	D/P																										
S1	D	S1	D																										
S2		S2																											

API	指令碼		P 指令	符號	功能																								
	32 位元	64 位元																											
0104	F+	DF+	✓	<table border="1"> <tr> <td>En</td> <td>F+</td> <td>En</td> <td>F+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> <table border="1"> <tr> <td>En</td> <td>DF+</td> <td>En</td> <td>DF+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	F+	En	F+P	S1	D	S1	D	S2		S2		En	DF+	En	DF+P	S1	D	S1	D	S2		S2		浮點數加法 $S1 + S2 = D$
En	F+	En	F+P																										
S1	D	S1	D																										
S2		S2																											
En	DF+	En	DF+P																										
S1	D	S1	D																										
S2		S2																											
0105	F-	DF-	✓	<table border="1"> <tr> <td>En</td> <td>F-</td> <td>En</td> <td>F-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> <table border="1"> <tr> <td>En</td> <td>DF-</td> <td>En</td> <td>DF-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	F-	En	F-P	S1	D	S1	D	S2		S2		En	DF-	En	DF-P	S1	D	S1	D	S2		S2		浮點數減法 $S1 - S2 = D$
En	F-	En	F-P																										
S1	D	S1	D																										
S2		S2																											
En	DF-	En	DF-P																										
S1	D	S1	D																										
S2		S2																											
0106	F*	DF*	✓	<table border="1"> <tr> <td>En</td> <td>F*</td> <td>En</td> <td>F*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> <table border="1"> <tr> <td>En</td> <td>DF*</td> <td>En</td> <td>DF*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	F*	En	F*P	S1	D	S1	D	S2		S2		En	DF*	En	DF*P	S1	D	S1	D	S2		S2		浮點數乘法 $S1 * S2 = D$
En	F*	En	F*P																										
S1	D	S1	D																										
S2		S2																											
En	DF*	En	DF*P																										
S1	D	S1	D																										
S2		S2																											
0107	F/	DF/	✓	<table border="1"> <tr> <td>En</td> <td>F/</td> <td>En</td> <td>F/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> <table border="1"> <tr> <td>En</td> <td>DF/</td> <td>En</td> <td>DF/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	F/	En	F/P	S1	D	S1	D	S2		S2		En	DF/	En	DF/P	S1	D	S1	D	S2		S2		浮點數除法 $S1 / S2 = D$
En	F/	En	F/P																										
S1	D	S1	D																										
S2		S2																											
En	DF/	En	DF/P																										
S1	D	S1	D																										
S2		S2																											

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0108	B+	DB+	✓		BCD 加法 $S1+S2=D$
0109	B-	DB-	✓		BCD 減法 $S1-S2=D$
0110	B*	DB*	✓		BCD 乘法 $S1*S2=D$
0111	B/	DB/	✓		BCD 除法 $S1/S2=D$
0112	BK+	—	✓		連續區塊 BIN 加法
0113	BK-	—	✓		連續區塊 BIN 減法
0114	\$+	—	✓		字串鏈結
0115	INC	DINC	✓		BIN 加一

3

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0116	DEC	DDEC	✓		BIN 減一

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0117	MUL16	MUL32	✓		16 位元專用 BIN 乘法/ 32 位元專用 BIN 乘法
0118	DIV16	DIV32	✓		16 位元專用 BIN 除法/ 32 位元專用 BIN 除法

● 資料轉換指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0200	BCD	DBCD	✓		BIN→BCD 變換
0201	BIN	DBIN	✓		BCD→BIN 變換
0202	FLT	DFLT	✓		BIN 整數→二進浮點數變換
0203	FLTD	DFLTD	✓		BIN 整數→64 位元浮點數變換

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0204	INT	DINT	✓		二進浮點數→BIN 整數變換

API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0205	FINT	DFINT	✓		64 位元浮點數→BIN 整數變換

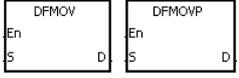
API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0206	MMOV	—	✓		16→32 位元數值轉換
0207	RMOV	—	✓		32→16 位元數值轉換
0208	GRY	DGRY	✓		BIN→GRY 碼變換
0209	GBIN	DGBIN	✓		GRY 碼→BIN 變換
0210	NEG	DNEG	✓		取負數 (取 2 的補數)
0211	—	FNEG	✓		32 位元浮點正負符號反相
0212	—	FBCD	✓		二進浮點數→十進浮點數

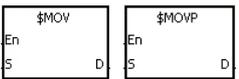
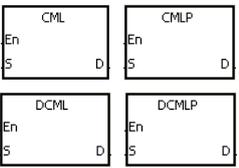
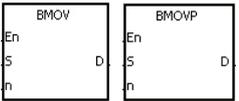
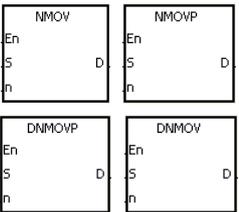
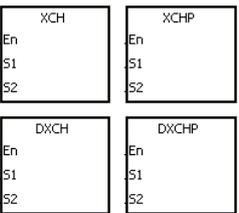
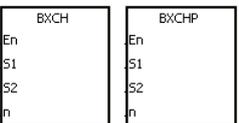
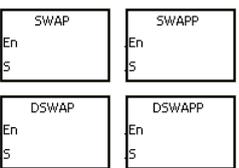
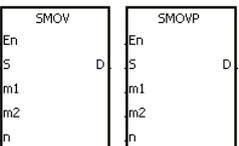
3

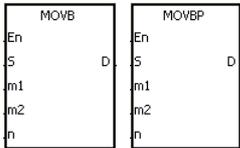
API	指令碼		P 指令	符號	功能																				
	16 位元	32 位元																							
0213	-	FBIN	✓	<table border="1"> <tr> <td>FBIN</td> <td>FBINP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	FBIN	FBINP	En	En	S	S	D	D	十進浮點數→二進浮點數												
FBIN	FBINP																								
En	En																								
S	S																								
D	D																								
0214	BKBCD	-	✓	<table border="1"> <tr> <td>BKBCD</td> <td>BKBCDP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	BKBCD	BKBCDP	En	En	S	S	D	D	連續區塊 BIN→BCD 變換												
BKBCD	BKBCDP																								
En	En																								
S	S																								
D	D																								
0215	BKBIN	-	✓	<table border="1"> <tr> <td>BKBIN</td> <td>BKBINP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	BKBIN	BKBINP	En	En	S	S	D	D	連續區塊 BCD→BIN 變換												
BKBIN	BKBINP																								
En	En																								
S	S																								
D	D																								
0216	SCAL	-	✓	<table border="1"> <tr> <td>SCAL</td> <td>SCALP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S1</td> <td>S1</td> </tr> <tr> <td>S2</td> <td>S2</td> </tr> <tr> <td>S3</td> <td>S3</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	SCAL	SCALP	En	En	S1	S1	S2	S2	S3	S3	D	D	比例運算								
SCAL	SCALP																								
En	En																								
S1	S1																								
S2	S2																								
S3	S3																								
D	D																								
0217	SCLP	DSCLP	✓	<table border="1"> <tr> <td>SCLP</td> <td>SCLPP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S1</td> <td>S1</td> </tr> <tr> <td>S2</td> <td>S2</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table> <table border="1"> <tr> <td>DSCLP</td> <td>DSCLPP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S1</td> <td>S1</td> </tr> <tr> <td>S2</td> <td>S2</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	SCLP	SCLPP	En	En	S1	S1	S2	S2	D	D	DSCLP	DSCLPP	En	En	S1	S1	S2	S2	D	D	參數型比例運算
SCLP	SCLPP																								
En	En																								
S1	S1																								
S2	S2																								
D	D																								
DSCLP	DSCLPP																								
En	En																								
S1	S1																								
S2	S2																								
D	D																								
0218	LINE	DLINE	✓	<table border="1"> <tr> <td>LINE</td> <td>LINEP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table> <table border="1"> <tr> <td>DLINE</td> <td>DLINEP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	LINE	LINEP	En	En	S	S	D	D	DLINE	DLINEP	En	En	S	S	D	D	COLUMN to LINE				
LINE	LINEP																								
En	En																								
S	S																								
D	D																								
DLINE	DLINEP																								
En	En																								
S	S																								
D	D																								
0219	COLM	DCOLM	✓	<table border="1"> <tr> <td>COLM</td> <td>COLMP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table> <table border="1"> <tr> <td>DCOLM</td> <td>DCOLMP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	COLM	COLMP	En	En	S	S	D	D	DCOLM	DCOLMP	En	En	S	S	D	D	LINE to COLUMN				
COLM	COLMP																								
En	En																								
S	S																								
D	D																								
DCOLM	DCOLMP																								
En	En																								
S	S																								
D	D																								

● 資料轉移指令

API	指令碼		P 指令	符號	功能																
	16 位元	32 位元																			
0300	MOV	DMOV	✓	<table border="1"> <tr> <td>MOV</td> <td>MOVP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table> <table border="1"> <tr> <td>DMOV</td> <td>DMOVP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>S</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	MOV	MOVP	En	En	S	S	D	D	DMOV	DMOVP	En	En	S	S	D	D	資料移動 S：資料來源 D：資料目的地
MOV	MOVP																				
En	En																				
S	S																				
D	D																				
DMOV	DMOVP																				
En	En																				
S	S																				
D	D																				

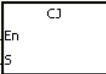
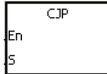
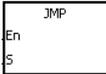
API	指令碼		P 指令	符號	功能
	32 位元	64 位元			
0301	–	DFMOV	✓		64 位元浮點數資料移動 S：資料來源 D：資料目的地

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0302	\$MOV	–	✓		字串移動
0303	CML	DCML	✓		反轉傳送
0304	BMOV	–	✓		全部傳送
0305	NMOV	DNMOV	✓		多點移動
0306	XCH	DXCH	✓		資料的交換
0307	BXCH	–	✓		全部交換
0308	SWAP	DSWAP	✓		上/下 BYTE 變換
0309	SMOV	–	✓		位數移動

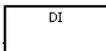
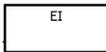
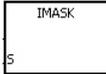
API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0310	MOVB	—	✓		多位元移動

● 程式跳躍指令

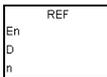
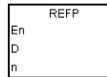
3

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0400	CJ	—	✓	 	條件跳躍
0401	JMP	—	—		無條件跳躍
0402	GOEND	—	—		跳躍到 END

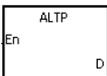
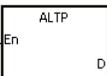
● 程式執行控制指令

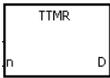
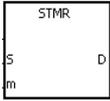
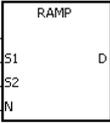
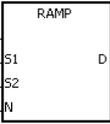
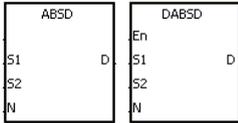
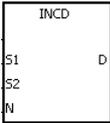
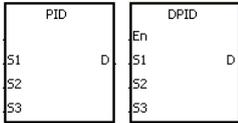
API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0500	DI	—	—		中斷插入禁能
0501	EI	—	—		中斷插入致能
0502	IMASK	—	—		中斷控制

● I/O 更新指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0600	REF	—	✓	 	I/O 更新處理

● 便利指令

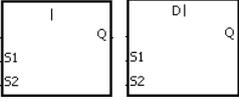
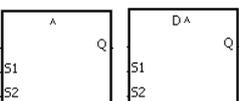
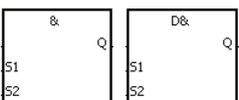
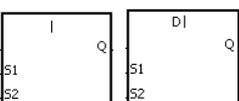
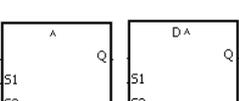
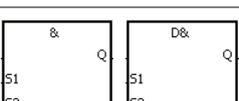
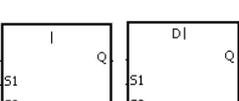
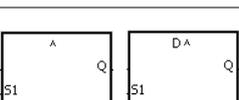
API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0700	ALT	—	✓	 	ON/OFF 交替

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0701	TTMR	—	—		教導式計時器
0702	STMR	—	—		特殊計時器
0703	RAMP	—	—		傾斜信號
0704	MTR	—	—		矩陣輸入
0705	ABSD	DABSD	—		絕對方式凸輪控制
0706	INCD	—	—		相對方式凸輪控制
0707	PID	DPID	—		PID 運算
0708	—	DPIDE	—		PID 運算

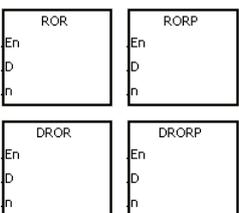
● 邏輯操作指令

3

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0800	WAND	DAND	✓		邏輯及 (AND) 運算
0801	MAND	—	✓		矩陣及 (AND) 運算
0802	WOR	DOR	✓		邏輯或 (OR) 運算
0803	MOR	—	✓		矩陣或 (OR) 運算
0804	WXOR	DXOR	✓		邏輯互斥或 (XOR) 運算
0805	MXOR	—	✓		矩陣互斥或 (XOR) 運算
0806	WXNR	DXNR	✓		邏輯互容或 (XNR) 運算
0807	MXNR	—	✓		矩陣互容或 (XNR) 運算
0809	LD&	DLD&	—		S1&S2

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
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
0816	OR	DOR	—		S1 S2
0817	OR^	DOR^	—		S1^S2

● 旋轉指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0900	ROR	DROR	✓		右旋轉

3

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
0901	RCR	DRCR	✓		附進位旗標右旋轉
0902	ROL	DROL	✓		欲旋轉之裝置
0903	RCL	DRCL	✓		附進位旗標左旋轉
0904	MBR	—	✓		矩陣位元旋轉

● 基本指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1000	RST	—	—	<p>元件 —(R)</p>	接點或暫存器清除
1001	TMR	—	—		16 位元計時器
1002	TMRH	—	—		16 位元計時器
1003	CNT	—	—		16 位元計數器
1004	—	DCNT	—		32 位元計數器

● 移位指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1100	SFTR	—	✓		位元右移
1101	SFTL	—	✓		位元左移
1102	WSFR	—	✓		暫存器右移
1103	WSFL	—	✓		暫存器左移
1104	SFWR	—	✓		位移寫入
1105	SFRD	—	✓		位移讀出
1106	SFPO	—	✓		讀出資料串列最新資料
1107	SFDEL	—	✓		刪除資料串列中的資料
1108	SFINS	—	✓		插入資料到資料串列中
1109	MBS	—	✓		矩陣位元位移
1110	SFR	—	✓		16 位元暫存器位元右移
1111	SFL	—	✓		16 位元暫存器位元左移

3

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
1112	BSFR	—	✓			n 個位元右移 1 個位元
1113	BSFL	—	✓			n 個位元左移 1 個位元
1114	NSFR	—	✓			n 個暫存器右移
1115	NSFL	—	✓			n 個暫存器左移

● 位元處理指令

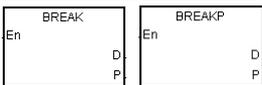
API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
1200	SER	DSER	✓			多點比較
1201	SUM	DSUM	✓			ON 位元數量
1202	DECO	—	✓			解碼器
1203	ENCO	—	✓			編碼器
1204	SEGD	—	✓			7 段顯示器解碼
1205	SORT	DSORT	-			資料排序

API	指令碼		P 指令	符號	功能																
	16 位元	32 位元																			
1206	ZRST	—	✓	<table border="1"> <tr> <td>ZRST</td> <td>ZRSTP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>D1</td> <td>D1</td> </tr> <tr> <td>D2</td> <td>D2</td> </tr> </table>	ZRST	ZRSTP	En	En	D1	D1	D2	D2	區域清除								
ZRST	ZRSTP																				
En	En																				
D1	D1																				
D2	D2																				
1207	BON	DBON	✓	<table border="1"> <tr> <td>BON</td> <td>BONP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table> <table border="1"> <tr> <td>DBON</td> <td>DBONP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	BON	BONP	En	En	S	D	n	n	DBON	DBONP	En	En	S	D	n	n	ON 位元判定
BON	BONP																				
En	En																				
S	D																				
n	n																				
DBON	DBONP																				
En	En																				
S	D																				
n	n																				
1208	MEAN	DMEAN	✓	<table border="1"> <tr> <td>MEAN</td> <td>MEANP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table> <table border="1"> <tr> <td>DMEAN</td> <td>DMEANP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	MEAN	MEANP	En	En	S	D	n	n	DMEAN	DMEANP	En	En	S	D	n	n	平均值
MEAN	MEANP																				
En	En																				
S	D																				
n	n																				
DMEAN	DMEANP																				
En	En																				
S	D																				
n	n																				
1209	CCD	—	✓	<table border="1"> <tr> <td>CCD</td> <td>CCDP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	CCD	CCDP	En	En	S	D	n	n	總和檢查								
CCD	CCDP																				
En	En																				
S	D																				
n	n																				
1210	ABS	DABS	✓	<table border="1"> <tr> <td>ABS</td> <td>ABSP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table> <table border="1"> <tr> <td>DABS</td> <td>DABSP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>D</td> <td>D</td> </tr> </table>	ABS	ABSP	En	En	D	D	DABS	DABSP	En	En	D	D	絕對值				
ABS	ABSP																				
En	En																				
D	D																				
DABS	DABSP																				
En	En																				
D	D																				
1211	MINV	—	✓	<table border="1"> <tr> <td>MINV</td> <td>MINVP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	MINV	MINVP	En	En	S	D	n	n	矩陣反相								
MINV	MINVP																				
En	En																				
S	D																				
n	n																				
1212	MBRD	—	✓	<table border="1"> <tr> <td>MBRD</td> <td>MBRDP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	MBRD	MBRDP	En	En	S	D	n	n	矩陣位元讀出								
MBRD	MBRDP																				
En	En																				
S	D																				
n	n																				
1213	MBWR	—	✓	<table border="1"> <tr> <td>MBWR</td> <td>MBWRP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	MBWR	MBWRP	En	En	S	D	n	n	矩陣位元寫入								
MBWR	MBWRP																				
En	En																				
S	D																				
n	n																				
1214	MBC	—	✓	<table border="1"> <tr> <td>MBC</td> <td>MBCP</td> </tr> <tr> <td>En</td> <td>En</td> </tr> <tr> <td>S</td> <td>D</td> </tr> <tr> <td>n</td> <td>n</td> </tr> </table>	MBC	MBCP	En	En	S	D	n	n	矩陣位元狀態計數								
MBC	MBCP																				
En	En																				
S	D																				
n	n																				

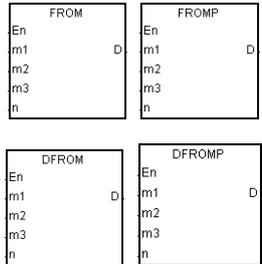
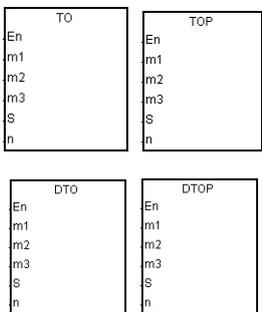
3

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1215	DIS	—	✓		16 位元資料的 4 位元分組
1216	UNI	—	✓		16 位元資料的 4 位元鏈結
1217	WSUM	DWSUM	✓		16 位元資料的總和計算
1218	BSET	—	✓		字元件的位設定
1219	BRST	—	✓		字元件的位復位
1220	BKRST	—	✓		指定區域清除
1221	LIMIT	DLIMIT	✓		BIN 16 位元和 BIN 32 位元資料的高低限控制
1222	BAND	DBAND	✓		BIN 16 位和 32 位死區控制
1223	ZONE	DZONE	✓		BIN 16 位元和 32 位元區域控制

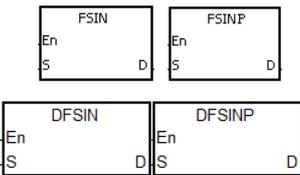
● 結構建立指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1300	FOR	—	—		巢串迴路起始
1301	NEXT	—	—		巢串迴路結束
1302	BREAK	—	✓		強制結束 FOR-NEXT 迴圈

● 模組的資料讀/寫指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1400	FROM	DFROM	✓		特殊模組 CR 資料讀出
1401	TO	DTO	✓		特殊模組 CR 資料寫入

● 浮點數指令

API	指令碼			P 指令	符號	功能
	16 位元	32 位元	64 位元			
1500	—	FSIN	DFSIN	✓		二進浮點數 SIN 運算

3

API	指令碼			P 指令	符號	功能
	16 位元	32 位元	64 位元			
1501	—	FCOS	DFCOS	✓		二進浮點數 COS 運算
1502	—	FTAN	DFTAN	✓		二進浮點數 TAN 運算
1503	—	FASIN	DFASIN	✓		二進浮點數 ASIN 運算
1504	—	FACOS	DFACOS	✓		二進浮點數 ACOS 運算
1505	—	FATAN	DFATAN	✓		二進浮點數 ATAN 運算
1506	—	FSINH	DFSINH	✓		二進浮點數 SINH 運算
1507	—	FCOSH	DFCOSH	✓		二進浮點數 COSH 運算
1508	—	FTANH	DFTANH	✓		二進浮點數 TANH 運算

API	指令碼			P 指令	符號	功能
	16 位元	32 位元	64 位元			
1509	—	FRAD	DFRAD	✓		角度→徑度
1510	—	FDEG	DFDEG	✓		徑度→角度
1511	SQR	DSQR	—	✓		BIN 開平方根
1512	—	FSQR	DFSQR	✓		浮點數開平方根
1513	—	FEXP	DFEXP	✓		浮點數取指數
1514	—	FLOG	DFLOG	✓		浮點數取對數
1515	—	FLN	DFLN	✓		二進浮點數取自然對數
1516	—	FPOW	DFPOW	✓		浮點數權值指令

3

3

API	指令碼			P 指令	符號	功能
	16 位元	32 位元	64 位元			
1517	RAND	—	—	✓		亂數值
1518	BSQR	DBSQR	—	✓		BCD 開平方根
1519	—	BSIN	—	✓		BCD SIN 運算
1520	—	BCOS	—	✓		BCD COS 運算
1521	—	BTAN	—	✓		BCD TAN 運算
1522	—	BASIN	—	✓		BCD ASIN 運算
1523	—	BACOS	—	✓		BCD ACOS 運算
1524	—	BATAN	—	✓		BCD ATAN 運算

● 萬年曆指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1600	TRD	—	✓		萬年曆資料讀出
1601	TWR	—	✓		萬年曆資料寫入
1602	T+	—	✓		萬年曆資料加算
1603	T-	—	✓		萬年曆資料減算

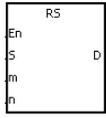
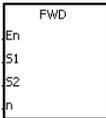
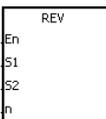
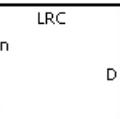
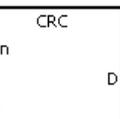
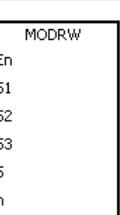
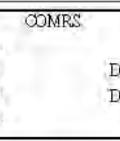
API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1604	HOUR	DHOUR	—		運轉計時器
1605	TCMP	—	✓		萬年曆資料比較
1606	TZCP	—	✓		萬年曆資料區域比較
1607	DST	—	✓		日光節約時間

● 週邊設備指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1700	TKY	DTKY	—		10 鍵鍵盤輸入
1701	HKY	DHKY	—		16 鍵鍵盤輸入
1702	DSW	—	—		指撥開關輸入
1703	ARWS	—	—		箭頭鍵盤輸入
1704	SEGL	—	—		七段顯示器掃描輸出

● 通訊指令

3

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1800	RS	—	—		串列資料傳輸
1801	FWD	—	—		變頻器正轉指令
1802	REV	—	—		變頻器反轉指令
1803	STOP	—	—		變頻器停止指令
1804	RDST	—	—		變頻器狀態讀取
1805	RSTEF	—	—		變頻器異常重置
1806	LRC	—	✓		和檢查 LRC 模式
1807	CRC	—	✓		和檢查 CRC 模式
1808	MODRW	—	—		MODBUS 資料讀寫
1812	COMRS	—	—		通訊傳送與接收指令

● 其他指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
1900	WDT	—	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> WDT En </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> WDT En </div>	逾時監視計時器
1901	DELAY	—	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> DELAY En S </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> DELAY En S </div>	延遲指令
1902	GPWM	—	—	<div style="display: inline-block; border: 1px solid black; padding: 2px;"> GPWM En S1 S2 S2 </div>	一般用脈波波寬調變
1903	TIMCHK	—	—	<div style="display: inline-block; border: 1px solid black; padding: 2px;"> TIMCHK En S1 D S2 </div>	時間檢查
1904	EPUSH	—	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> EPUSH En D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> EPUSH En D </div>	指標暫存器存入
1905	EPOP	—	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> EPOP En D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> EPOP En D </div>	指標暫存器讀出

● 字串處理指令

API	指令碼		P 指令	符號	功能
	16 位元	32 位元			
2100	BINDA	DBINDA	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> BINDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> BINDAP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> DBINDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> DBINDAP En S D </div>	有號數十進制→ASCII 變換
2101	BINHA	DBINHA	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> BINHA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> BINHAP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> DBINHA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> DBINHAP En S D </div>	BIN 十六進制→十六進制 ASCII 變換
2102	BCDDA	DBCDDA	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> BCDDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> BCDDAP En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px; margin-right: 10px;"> DBCDDA En S D </div> <div style="display: inline-block; border: 1px solid black; padding: 2px;"> DBCDDAP En S D </div>	BCD→ASCII 變換

3

API	指令碼		P 指令	符號		功能																																				
	16 位元	32 位元																																								
2103	DABIN	DDABIN	✓	<table border="1"> <tr><td>En</td><td>DABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	DABIN	D	S		D	En	DDABIN	D	S		D	<table border="1"> <tr><td>En</td><td>DABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	DABINP	D	S		D	En	DDABINP	D	S		D	有號數十進制 ASCII→有號數十進制 BIN 變換												
En	DABIN	D																																								
S		D																																								
En	DDABIN	D																																								
S		D																																								
En	DABINP	D																																								
S		D																																								
En	DDABINP	D																																								
S		D																																								
2104	HABIN	DHABIN	✓	<table border="1"> <tr><td>En</td><td>HABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table> <table border="1"> <tr><td>En</td><td>DHABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	HABIN	D	S		D	En	DHABIN	D	S		D	<table border="1"> <tr><td>En</td><td>HABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table> <table border="1"> <tr><td>En</td><td>DHABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	HABINP	D	S		D	En	DHABINP	D	S		D	十六進制 ASCII→十六進制 BIN 變換												
En	HABIN	D																																								
S		D																																								
En	DHABIN	D																																								
S		D																																								
En	HABINP	D																																								
S		D																																								
En	DHABINP	D																																								
S		D																																								
2105	DABCD	DDABCD	✓	<table border="1"> <tr><td>En</td><td>DABCD</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABCD</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	DABCD	D	S		D	En	DDABCD	D	S		D	<table border="1"> <tr><td>En</td><td>DABCDP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABCDP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	DABCDP	D	S		D	En	DDABCDP	D	S		D	ASCII→BCD 變換												
En	DABCD	D																																								
S		D																																								
En	DDABCD	D																																								
S		D																																								
En	DABCDP	D																																								
S		D																																								
En	DDABCDP	D																																								
S		D																																								
2106	\$LEN	—	✓	<table border="1"> <tr><td>En</td><td>\$LEN</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	\$LEN	D	S		D	<table border="1"> <tr><td>En</td><td>\$LENP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	\$LENP	D	S		D	計算字串長度																								
En	\$LEN	D																																								
S		D																																								
En	\$LENP	D																																								
S		D																																								
2107	\$STR	\$DSTR	✓	<table border="1"> <tr><td>En</td><td>\$STR</td><td>D</td></tr> <tr><td>S1</td><td></td><td>D</td></tr> <tr><td>S2</td><td></td><td>D</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>D</td></tr> <tr><td>S2</td><td></td><td>D</td></tr> </table>	En	\$STR	D	S1		D	S2		D	En	D\$STR	D	S1		D	S2		D	<table border="1"> <tr><td>En</td><td>\$STRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td>D</td></tr> <tr><td>S2</td><td></td><td>D</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>D</td></tr> <tr><td>S2</td><td></td><td>D</td></tr> </table>	En	\$STRP	D	S1		D	S2		D	En	D\$STRP	D	S1		D	S2		D	BIN→String
En	\$STR	D																																								
S1		D																																								
S2		D																																								
En	D\$STR	D																																								
S1		D																																								
S2		D																																								
En	\$STRP	D																																								
S1		D																																								
S2		D																																								
En	D\$STRP	D																																								
S1		D																																								
S2		D																																								
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>D</td></tr> <tr><td>S2</td><td></td><td>D</td></tr> </table>	En	\$FSTR	D	S1		D	S2		D	<table border="1"> <tr><td>En</td><td>\$FSTRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td>D</td></tr> <tr><td>S2</td><td></td><td>D</td></tr> </table>	En	\$FSTRP	D	S1		D	S2		D	Float→String																		
En	\$FSTR	D																																								
S1		D																																								
S2		D																																								
En	\$FSTRP	D																																								
S1		D																																								
S2		D																																								
2110	\$FVAL	—	✓	<table border="1"> <tr><td>En</td><td>\$FVAL</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	\$FVAL	D	S		D	<table border="1"> <tr><td>En</td><td>\$FVALP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> </table>	En	\$FVALP	D	S		D	String→Float																								
En	\$FVAL	D																																								
S		D																																								
En	\$FVALP	D																																								
S		D																																								
2111	\$RIGHT	—	✓	<table border="1"> <tr><td>En</td><td>\$RIGHT</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> <tr><td>n</td><td></td><td>D</td></tr> </table>	En	\$RIGHT	D	S		D	n		D	<table border="1"> <tr><td>En</td><td>\$RIGHTP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> <tr><td>n</td><td></td><td>D</td></tr> </table>	En	\$RIGHTP	D	S		D	n		D	從右邊擷取字串																		
En	\$RIGHT	D																																								
S		D																																								
n		D																																								
En	\$RIGHTP	D																																								
S		D																																								
n		D																																								
2112	\$LEFT	—	✓	<table border="1"> <tr><td>En</td><td>\$LEFT</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> <tr><td>n</td><td></td><td>D</td></tr> </table>	En	\$LEFT	D	S		D	n		D	<table border="1"> <tr><td>En</td><td>\$LEFTP</td><td>D</td></tr> <tr><td>S</td><td></td><td>D</td></tr> <tr><td>n</td><td></td><td>D</td></tr> </table>	En	\$LEFTP	D	S		D	n		D	從左邊擷取字串																		
En	\$LEFT	D																																								
S		D																																								
n		D																																								
En	\$LEFTP	D																																								
S		D																																								
n		D																																								
2113	\$MIDR	—	✓	<table border="1"> <tr><td>En</td><td>\$MIDR</td><td>D</td></tr> <tr><td>S1</td><td></td><td>D</td></tr> <tr><td>S2</td><td></td><td>D</td></tr> </table>	En	\$MIDR	D	S1		D	S2		D	<table border="1"> <tr><td>En</td><td>\$MIDRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td>D</td></tr> <tr><td>S2</td><td></td><td>D</td></tr> </table>	En	\$MIDRP	D	S1		D	S2		D	區段擷取字串																		
En	\$MIDR	D																																								
S1		D																																								
S2		D																																								
En	\$MIDRP	D																																								
S1		D																																								
S2		D																																								

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
2114	\$MIDW	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$MIDW En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px;"> \$MIDWP En S1 S2 </div>	區段字串取代
2115	\$SER	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$SER En S1 S2 N </div>	<div style="border: 1px solid black; padding: 2px;"> \$SERP En S1 S2 N </div>	字串搜尋
2116	\$RPLC	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$RPLC En S1 S2 S3 S4 </div>	<div style="border: 1px solid black; padding: 2px;"> \$RPLCP En S1 S2 S3 S4 </div>	字串取代
2117	\$DEL	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$DEL En S1 S2 S3 </div>	<div style="border: 1px solid black; padding: 2px;"> \$DELP En S1 S2 S3 </div>	指定字串刪除
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>	字串清除
2119	\$INS	—	✓	<div style="border: 1px solid black; padding: 2px;"> \$INS En S1 S2 S3 </div>	<div style="border: 1px solid black; padding: 2px;"> \$INSP En S1 S2 S3 </div>	字串插入
2120	\$FMOD	—	✓	<div style="border: 1px solid black; padding: 2px;"> FMOD En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px;"> FMODP En S1 S2 </div>	浮點數轉 BCD 浮點數
2121	\$FREXP	—	✓	<div style="border: 1px solid black; padding: 2px;"> FREXP En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px;"> FREXPP En S1 S2 </div>	BCD 浮點數轉浮點數

3

● 乙太網控制指令

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
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>	開啟 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>	透過已開啟的 Socket 傳送資料
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>	透過已開啟的 Socket 接收資料

3

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
2203	SCLOSE	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SCLOSE En S1 S2 </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SCLOSEP En S1 S2 </div>	關閉 Socket
2204	MSEND	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MSEND En S1 S2 S3 D </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MSENDP En S1 S2 S3 D </div>	寄送電子郵件
2205	EMDRW	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> EMDRW En S1 S2 S3 S n </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> EMDRWP En S1 S2 S3 S n </div>	MODBUS TCP 資料讀寫
2206	—	DINTOA	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> DINTOA En S D </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> DINTOAP En S D </div>	IP→字串變換
2207	—	DIATON	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> DIATON En S D </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> DIATONP En S D </div>	字串→IP 變換
2208	EIPRW	—	—	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> EIPRW En S1 S2 S3 S4 S5 S6 S7 n S </div>		EtherNet/IP 讀寫通訊指令

● 記憶卡讀寫指令

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
2300	MWRIT	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MWRIT En C S S1 S2 S3 S4 </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MWRITP En C S S1 S2 S3 S4 </div>	寫入記憶卡資料
2301	MREAD	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MREAD En C S S1 S2 S3 D </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MREADP En C S S1 S2 S3 D </div>	讀取記憶卡資料

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
2302	MTWRIT	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MTWRIT En C S S1 S2 S3 </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> MTWRITP En C S S1 S2 S3 </div>	寫入字串至記憶卡

● 任務控制指令

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
2400	TKON	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> TKON En S </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> TKONP En S </div>	Cyclic 任務 Task 啟動
2401	TKOFF	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> TKOFF En S </div>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> TKOFFP En S </div>	Cyclic 任務 Task 關閉

● SFC 控制指令

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

● 備援系統控制指令

API	指令碼		P 指令	符號		功能
	16 位元	32 位元				
2900	SSO	—	✓	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> SSOP En D </div>		備援系統切換
2901	RCS	—	—	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> RCS En S D </div>		讀取/設定備援系統資訊

3.4.2 應用指令 (依英文字母排序)

3

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
符號	0101	—	D-	—	✓	BIN 減法
	0114	\$+	—	—	✓	字串鏈結
	2118	\$CLR	—	—	✓	字串清除
	2117	\$DEL	—	—	✓	指定字串刪除
	2109	\$FSTR	—	—	✓	Float→String
	2110	\$FVAL	—	—	✓	String→Float
	2119	\$INS	—	—	✓	字串插入
	2112	\$LEFT	—	—	✓	從左邊擷取字串
	2106	\$LEN	—	—	✓	計算字串長度
	2113	\$MIDR	—	—	✓	區段擷取字串
	2114	\$MIDW	—	—	✓	區段字串取代
	0302	\$MOV	—	—	✓	字串移動
	2111	\$RIGHT	—	—	✓	從右邊擷取字串
	2116	\$RPLC	—	—	✓	字串取代
	2115	\$SER	—	—	✓	字串搜尋
	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	—	✓	絕對值
	0705	ABSD	DABSD	—	—	絕對方式凸輪控制
	0700	ALT	—	—	✓	ON/OFF 交替
	0046	AND\$<	—	—	—	字串型態比較 S1 < S2
	0047	AND\$<=	—	—	—	字串型態比較 S1 ≤ S2
	0043	AND\$<>	—	—	—	字串型態比較 S1 ≠ S2
	0042	AND\$=	—	—	—	字串型態比較 S1 = S2
	0044	AND\$>	—	—	—	字串型態比較 S1 > S2
	0045	AND\$>=	—	—	—	字串型態比較 S1 ≥ S2
	0812	AND&	DAND&	—	—	接點型態邏輯運算 S1&S2
	0814	AND^	DAND^	—	—	接點型態邏輯運算 S1^S2
	0813	AND	DAND	—	—	接點型態邏輯運算 S1 S2
	0010	AND<	DAND<	—	—	接點型態比較 S1 < S2

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
A	0011	AND<=	DAND<=	—	—	接點型態比較 $S1 \leq S2$
	0007	AND<>	DAND<>	—	—	接點型態比較 $S1 \neq S2$
	0006	AND=	DAND=	—	—	接點型態比較 $S1 = S2$
	0008	AND>	DAND>	—	—	接點型態比較 $S1 > S$
	0009	AND>=	DAND>=	—	—	接點型態比較 $S1 \geq S2$
	-	ANED	—	—	—	負緣檢出動作串聯連接
	-	APED	—	—	—	正緣檢出動作串聯連接
	1703	ARWS	—	—	—	箭頭鍵盤輸入
B	0109	B-	DB-	—	✓	BCD 減法
	0110	B*	DB*	—	✓	BCD 乘法
	0111	B/	DB/	—	✓	BCD 除法
	0108	B+	DB+	—	✓	BCD 加法
	1523	BACOS	—	—	✓	BCD ACOS 運算
	1222	BAND	DBAND	—	✓	BIN 16 位和 32 位死區控制
	1522	BASIN	—	—	✓	BCD ASIN 運算
	1524	BATAN	—	—	✓	BCD ATAN 運算
	0200	BCD	DBCD	—	✓	BIN→BCD 變換
	2102	BCDDA	DBCDDA	—	✓	BCD→ASCII 變換
	1520	BCOS	—	—	✓	BCD COS 運算
	0201	BIN	DBIN	—	✓	BCD→BIN 變換
	2100	BINDA	DBINDA	—	✓	有號數十進制→ASCII 變換
	2101	BINHA	DBINHA	—	✓	BIN 十六進制→十六進制 ASCII 變換
	0113	BK-	—	—	✓	連續區塊 BIN 減法
	0112	BK+	—	—	✓	連續區塊 BIN 加法
	0214	BKBCD	—	—	✓	連續區塊 BIN→BCD 變換
	0215	BKBIN	—	—	✓	連續區塊 BCD→BIN 變換
	1220	BKRST	—	—	✓	指定區域清除
	0304	BMOV	—	—	✓	全部傳送
	1207	BON	DBON	—	✓	ON 位元判定
	1302	BREAK	—	—	✓	強制結束 FOR-NEXT 迴圈
	1219	BRST	—	—	✓	字軟元件的位復位
	1218	BSET	—	—	✓	字軟元件的位設定
	1113	BSFL	—	—	✓	n 個位元左移 1 個位元
	1112	BSFR	—	—	✓	n 個位元右移 1 個位元
	1519	BSIN	—	—	✓	BCD SIN 運算

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
B	1518	BSQR	DBSQR	—	✓	BCD 開平方根
	1521	BTAN	—	—	✓	BCD TAN 運算
	0307	BXCH	—	—	✓	全部交換
C	1209	CCD	—	—	✓	總和檢查
	0065	CHKADR	—	—	—	接點型態指標暫存器位址檢查
	0400	CJ	—	—	✓	條件跳躍
	1812	COMRS	—	—	—	通訊傳送與接收指令
	0303	CML	DCML	—	✓	反轉傳送
	0054	CMP	DCMP	—	✓	比較設定輸出
	0063	CMPT<	—	—	✓	表格比較<
	0064	CMPT<=	—	—	✓	表格比較<=
	0060	CMPT<>	—	—	✓	表格比較<>
	0059	CMPT=	—	—	✓	表格比較=
	0061	CMPT>	—	—	✓	表格比較>
	0062	CMPT>=	—	—	✓	表格比較>=
	1003	CNT	—	—	—	16 位元計數器
	0219	COLM	DCOLM	—	✓	LINE to COLUMN
	1807	CRC	—	—	—	和檢查 CRC 模式
	D	2105	DABCD	DDABCD	—	✓
2103		DABIN	DDABIN	—	✓	有號數十進制 ASCII→有號數十進制 BIN 變換
1004		DCNT	—	—	—	32 位元計數器
0116		DEC	DDEC	—	✓	BIN 減一
1202		DECO	—	—	✓	解碼器
1901		DELAY	—	—	✓	延遲指令
0301		—	—	DFMOV	✓	64 位元浮點數資料移動
0500		DI	—	—	—	中斷插入禁能
2207		DIATON	—	—	✓	字串→IP 變換
2206		DINTOA	—	—	✓	IP→字串變換
1215		DIS	—	—	✓	16 位元資料的 4 位元分組
0118		DIV16	DIV32	—	✓	16 位元專用 BIN 除法/ 32 位元專用 BIN 除法
0708		—	DPIDE	—	—	PID 運算
1607		DST	—	—	✓	日光節約時間
1702	DSW	—	—	—	指撥開關輸入	

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
E	0501	EI	—	—	—	中斷插入致能
	2205	EMDRW	—	—	✓	MODBUS TCP 資料讀寫
	1203	ENCO	—	—	✓	編碼器
	1905	EPOP	—	—	✓	指標暫存器讀出
	1904	EPUSH	—	—	✓	指標暫存器存入
	2208	EIPRW	—	—	—	EtherNet/IP 讀寫通訊指令
F	0105	—	F-	DF-	✓	浮點數減法
	0106	—	F*	DF*	✓	浮點數乘法
	0107	—	F/	DF/	✓	浮點數除法
	0104	—	F+	DF+	✓	浮點數加法
	1504	—	FACOS	DFACOS	✓	浮點數 ACOS 運算
	0028	—	FAND<	DFAND<	—	浮點數接點型態比較 $S1 < S2$
	0029	—	FAND<=	DFAND<=	—	浮點數接點型態比較 $S1 \leq S2$
	0025	—	FAND<>	DFAND<>	—	浮點數接點型態比較 $S1 \neq S2$
	0024	—	FAND=	DFAND=	—	浮點數接點型態比較 $S1 = S2$
	0026	—	FAND>	DFAND>	—	浮點數接點型態比較 $S1 > S$
	0027	—	FAND>=	DFAND>=	—	浮點數接點型態比較 $S1 \geq S2$
	1503	—	FASIN	DFASIN	✓	浮點數 ASIN 運算
	1505	—	FATAN	DFATAN	✓	浮點數 ATAN 運算
	0212	—	FBCD	—	✓	浮點數→十進浮點數
	0213	—	FBIN	—	✓	十進浮點數→浮點數
	0056	—	FCMP	—	✓	浮點數比較
	1501	—	FCOS	DFCOS	✓	浮點數 COS 運算
	1507	—	FCOSH	DFCOSH	✓	浮點數 COSH 運算
	1510	—	FDEG	DFDEG	✓	徑度→角度
	1513	—	FEXP	DFEXP	✓	浮點數取指數
	0205	—	FINT	DFINT	✓	64 位元浮點數→BIN 整數變換
	0022	—	FLD<	DFLD<	—	浮點數接點型態比較 $S1 < S2$
	0023	—	FLD<=	DFLD<=	—	浮點數接點型態比較 $S1 \leq S2$
	0019	—	FLD<>	DFLD<>	—	浮點數接點型態比較 $S1 \neq S2$
	0018	—	FLD=	DFLD=	—	浮點數接點型態比較 $S1 = S2$
	0020	—	FLD>	DFLD>	—	浮點數接點型態比較 $S1 > S$
	0021	—	FLD>=	DFLD>=	—	浮點數接點型態比較 $S1 \geq S2$
	1515	—	FLN	DFLN	✓	浮點數取自然對數
1514	—	FLOG	DFLOG	✓	浮點數取對數	

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
F	0202	FLT	DFLT	—	✓	BIN 整數→浮點數變換
	0203	FLTD	DFLTD	—	✓	BIN 整數→64 位元浮點數變換
	2120	FMOD	—	—	✓	浮點數轉 BCD 浮點數
	0211	FNEG	—	—	✓	32 位元浮點正負符號反相
	1300	FOR	—	—	—	巢串迴路起始
	0034	—	FOR<	DFOR<	—	浮點數接點型態比較 $S1 < S2$
	0035	—	FOR<=	DFOR<=	—	浮點數接點型態比較 $S1 \leq S2$
	0031	—	FOR<>	DFOR<>	—	浮點數接點型態比較 $S1 \neq S2$
	0030	—	FOR=	DFOR=	—	浮點數接點型態比較 $S1 = S2$
	0032	—	FOR>	DFOR>	—	浮點數接點型態比較 $S1 > S$
	0033	—	FOR>=	DFOR>=	—	浮點數接點型態比較 $S1 \geq S2$
	1516	—	FPOW	DFPOW	✓	浮點數權值指令
	1509	—	FRAD	DFRAD	✓	角度→徑度
	2121	FREXP	—	—	✓	BCD 浮點數轉浮點數
	1400	FROM	DFROM	—	✓	特殊模組 CR 資料讀出
	1500	—	FSIN	DFSIN	✓	浮點數 SIN 運算
	1506	—	FSINH	DFSINH	✓	浮點數 SINH 運算
	1512	—	FSQR	DFSQR	✓	浮點數開平方根
	1502	—	FTAN	DFTAN	✓	浮點數 TAN 運算
	1508	—	FTANH	DFTANH	✓	浮點數 TANH 運算
1801	FWD	—	—	—	VFD-A 變頻器正轉指令	
0057	—	FZCP	—	✓	浮點數區域比較	
G	0209	GBIN	DGBIN	—	✓	GRY 碼→BIN 變換
	0402	GOEND	—	—	—	跳躍到 END
	1902	GPWM	—	—	—	一般用脈波波寬調變
	0208	GRY	DGRY	—	✓	BIN→GRY 碼變換
H	2104	HABIN	DHABIN	—	✓	十六進制 ASCII→十六進制 BIN 變換
	1701	HKY	DHKY	—	—	16 鍵鍵盤輸入
	1604	HOUR	DHOUR	—	—	運轉計時器
I	0502	IMASK	—	—	—	中斷控制
	0115	INC	DINC	—	✓	BIN 加一
	0706	INCD	—	—	—	相對方式凸輪控制
	0204	INT	DINT	—	✓	浮點數→BIN 整數變換
J	0401	JMP	—	—	—	無條件跳躍
L	0040	LD\$<	—	—	—	字串型態比較 $S1 < S2$

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
L	0041	LD\$<=	—	—	—	字串型態比較 $S1 \leq S2$
	0037	LD\$<>	—	—	—	字串型態比較 $S1 \neq S2$
	0036	LD\$=	—	—	—	字串型態比較 $S1 = S2$
	0038	LD\$>	—	—	—	字串型態比較 $S1 > S2$
	0039	LD\$>=	—	—	—	字串型態比較 $S1 \geq S2$
	0809	LD&	DLD&	—	—	接點型態邏輯運算 $S1 \& S2$
	0811	LD^	DLD^	—	—	接點型態邏輯運算 $S1 \wedge S2$
	0810	LD	DLD	—	—	接點型態邏輯運算 $S1 S2$
	0004	LD<	DLD<	—	—	接點型態比較 $S1 < S2$
	0005	LD<=	DLD<=	—	—	接點型態比較 $S1 \leq S2$
	0001	LD<>	DLD<>	—	—	接點型態比較 $S1 \neq S2$
	0000	LD=	DLD=	—	—	接點型態比較 $S1 = S2$
	0002	LD>	DLD>	—	—	接點型態比較 $S1 > S$
	0003	LD>=	DLD>=	—	—	接點型態比較 $S1 \geq S2$
	1221	LIMIT	DLIMIT	—	✓	限制 BIN 值的範圍
	0218	LINE	DLINE	—	✓	COLUMN to LINE
	1806	LRC	—	—	—	和檢查 LRC 模式
M	0801	MAND	—	—	✓	矩陣及 (AND) 運算
	1214	MBC	—	—	✓	矩陣位元狀態計數
	0904	MBR	—	—	✓	矩陣位元旋轉
	1212	MBRD	—	—	✓	矩陣位元讀出
	1109	MBS	—	—	✓	矩陣位元位移
	1213	MBWR	—	—	✓	矩陣位元寫入
	0058	MCMP	—	—	✓	矩陣比較
	1208	MEAN	DMEAN	—	✓	平均值
	1211	MINV	—	—	✓	矩陣反相
	0206	MMOV	—	—	✓	16→32 位元數值轉換
	1808	MODRW	—	—	—	MODBUS 資料讀寫
	0803	MOR	—	—	✓	矩陣或 (OR) 運算
	0300	MOV	DMOV	—	✓	資料移動
	0310	MOVB	—	—	✓	多位元移動
	2301	MREAD	—	—	—	讀取記憶卡資料
	2204	MSEND	—	—	✓	寄送電子郵件
	0704	MTR	—	—	—	矩陣輸入
	2302	MTWRIT	—	—	—	寫入字串至記憶卡

3

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
M	0117	MUL16	MUL32	—	✓	16 位元專用 BIN 乘法/ 32 位元專用 BIN 乘法
	2300	MWRIT	—	—	—	寫入記憶卡資料
	0807	MXNR	—	—	✓	矩陣互容或 (XNR) 運算
	0805	MXOR	—	—	✓	矩陣互斥或 (XOR) 運算
N	-	NED	—	—	—	負緣檢出動作開始連接
	0210	NEG	DNEG	—	✓	取負數 (取 2 的補數)
	1301	NEXT	—	—	—	巢串迴路結束
	0305	NMOV	DNMOV	—	✓	多點移動
	1115	NSFL	—	—	✓	n 個暫存器左移
	1114	NSFR	—	—	✓	n 個暫存器右移
O	-	ONED	—	—	—	負緣檢出動作並聯連接
	-	OPED	—	—	—	正緣檢出動作並聯連接
	0052	OR\$<	—	—	—	字串型態比較 S1 < S2
	0053	OR\$<=	—	—	—	字串型態比較 S1 ≤ S2
	0049	OR\$<>	—	—	—	字串型態比較 S1 ≠ S2
	0048	OR\$=	—	—	—	字串型態比較 S1 = S2
	0050	OR\$>	—	—	—	字串型態比較 S1 > S2
	0051	OR\$>=	—	—	—	字串型態比較 S1 ≥ S2
	0815	OR&	DOR&	—	—	接點型態邏輯運算 S1&S2
	0817	OR^	DOR^	—	—	接點型態邏輯運算 S1^S2
	0816	OR	DOR	—	—	接點型態邏輯運算 S1 S2
	0016	OR<	DOR<	—	—	接點型態比較 S1 < S2
	0017	OR<=	DOR<=	—	—	接點型態比較 S1 ≤ S2
	0013	OR<>	DOR<>	—	—	接點型態比較 S1 ≠ S2
0012	OR=	DOR=	—	—	接點型態比較 S1 = S2	
0014	OR>	DOR>	—	—	接點型態比較 S1 > S	
0015	OR>=	DOR>=	—	—	接點型態比較 S1 ≥ S2	
P	-	PED	—	—	—	正緣檢出動作開始連接
	0707	PID	—	—	—	PID 運算
R	0703	RAMP	—	—	—	傾斜信號
	1517	RAND	—	—	✓	亂數值
	0903	RCL	DRCL	—	✓	附進位旗標左旋轉
	0901	RCR	DRCR	—	✓	附進位旗標右旋轉
	2901	RCS	—	—	—	讀取/設定備援系統資訊
	1804	RDST	—	—	—	VFD-A 變頻器狀態讀取

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
R	0600	REF	—	—	✓	I/O 更新處理
	1802	REV	—	—	—	VFD-A 變頻器反轉指令
	0207	RMOV	—	—	✓	32→16 位元數值轉換
	0902	ROL	DROL	—	✓	左旋轉
	0900	ROR	DROR	—	✓	右旋轉
	1800	RS	—	—	—	串列資料傳輸
	1000	RST	—	—	—	接點或暫存器清除
	1805	RSTEF	—	—	—	VFD-A 變頻器異常重置
S	0216	SCAL	—	—	✓	比例運算
	2203	SCLOSE	—	—	✓	關閉 Socket
	0217	SCLP	DSCLP	—	✓	比例運算
	1204	SEGD	—	—	✓	7 段顯示器解碼
	1704	SEGL	—	—	—	七段顯示器掃描輸出
	1200	SER	DSER	—	✓	多點比較
	2501	SFCPSE	—	—	—	SFC 暫停
	2500	SFCRUN	—	—	—	SFC 啟動
	2502	SFCSTP	—	—	—	SFC 停止
	1107	SFDEL	—	—	✓	刪除資料串列中的資料
	1108	SFINS	—	—	✓	插入資料到資料串列中
	1111	SFL	—	—	✓	16 位元暫存器位元左移
	1106	SFPO	—	—	✓	讀出資料串列最新資料
	1110	SFR	—	—	✓	16 位元暫存器位元右移
	1105	SFRD	—	—	✓	位移讀出
	1101	SFTL	—	—	✓	位元左移
	1100	SFTR	—	—	✓	位元右移
	1104	SFWR	—	—	✓	位移寫入
	0309	SMOV	—	—	✓	位數移動
	2200	SOPEN	—	—	✓	開啟 Socket
	1205	SORT	DSORT	—	—	資料排序
	1511	SQR	DSQR	—	✓	BIN 開平方根
	2202	SRCVD	—	—	✓	透過已開啟的 Socket 接收資料
	2201	SSEND	—	—	✓	透過已開啟的 Socket 傳送資料
	2900	SSO	—	—	✓	備援系統切換
	0702	STMR	—	—	—	特殊計時器
1803	STOP	—	—	—	VFD-A 變頻器停止指令	

分類	API	指令碼 (位元)			P 指令	功能
		16	32	64		
S	1201	SUM	DSUM	—	✓	ON 位元數量
	0308	SWAP	DSWAP	—	✓	上/下 BYTE 變換
T	1603	T-	—	—	✓	萬年曆資料減算
	1602	T+	—	—	✓	萬年曆資料加算
	1605	TCMP	—	—	✓	萬年曆資料比較
	1903	TIMCHK	—	—	—	時間檢查
	2401	TKOFF	—	—	✓	任務 TKS 關閉
	2400	TKON	—	—	✓	任務 TKS 啟動
	1700	TKY	DTKY	—	—	10 鍵鍵盤輸入
	1001	TMR	—	—	—	16 位元計時器
	1002	TMRH	—	—	—	16 位元計時器
	1401	TO	DTO	—	✓	特殊模組 CR 資料寫入
	1600	TRD	—	—	✓	萬年曆資料讀出
	0701	TTMR	—	—	—	教導式計時器
	1601	TWR	—	—	✓	萬年曆資料寫入
	1606	TZCP	—	—	✓	萬年曆資料區域比較
	U	1216	UNI	—	—	✓
W	0800	WAND	DAND	—	✓	邏輯及 (AND) 運算
	1900	WDT	—	—	✓	逾時監視計時器
	0802	WOR	DOR	—	✓	邏輯或 (OR) 運算
	1103	WSFL	—	—	✓	暫存器左移
	1102	WSFR	—	—	✓	暫存器右移
	1217	WSUM	DWSUM	—	✓	16 位元資料的總和計算
	0806	WXNR	DXNR	—	✓	邏輯互容或 (XNR) 運算
	0804	WXOR	DXOR	—	✓	輯互斥或 (XOR) 運算
X	0306	XCH	DXCH	—	✓	資料的交換
Z	0055	ZCP	DZCP	—	✓	區域比較
	1223	ZONE	DZONE	—	✓	BIN 16 位元和 32 位元區域控制
	1206	ZRST	—	—	✓	區域清除

4

第4章 指令結構

目錄

4.1	API 應用指令組成說明.....	4-2
4.2	指令使用限制.....	4-5
4.3	間接指定說明.....	4-7
4.4	指標暫存器 (PR) 說明.....	4-8
4.5	計時器指標暫存器 (TR) 說明.....	4-10
4.6	16 位元計數器指標暫存器 (CR) 說明.....	4-12
4.7	32 位元計數器指標暫存器 (HCR) 說明.....	4-13

4.1 API 應用指令組成說明

PLC 指令提供一個特定的指令碼及 API 編號，以便記憶。下面的表格中指令的 API 編號為 0300。指令碼為 MOV，MOV 的功能描述是“資料移動”。

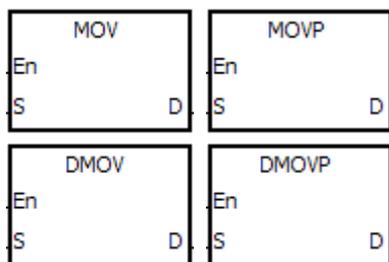
API	指令碼			運算元						功能					
0300	D	MOV	P	S · D						資料移動					

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

4

符號：



S：資料來源 Word/Double Word
D：資料目的地 Word/Double Word

1. 運算元區域會列出各種指令所用到的裝置。S · D · n · m 會根據它的功能使用於運算元，如果不止一個運算元並且功能相同時，會附加編號使用，如 S₁ · S₂...
2. 脈波執行型指令要求在指令碼後面加上“P”，而 32 位元指令要求在指令碼前面加上“D”，如“D***P”，“***”是指令碼。
3. 運算元區域中，裝置 PR 為間接尋址功能暫存器，PR 指標使用方式請參考 ISPSOft 使用手冊及 4.4 節指標暫存器說明。
4. 如果要在功能塊中使用指令時，運算元區域中有支援裝置 T · C · HC，則必須使用 TR · CR · HCR 指標暫存器來運算，詳細說明請參考 4.5~4.7 節。
5. 運算元區域中，32 位元單精度浮點數表示為 F，64 位元雙精度浮點數表示為 DF。
6. 上表中含●的裝置支援 E 修飾，含○的裝置不支援 E 修飾。例如，運算元 S₁ 支援 D 裝置的 E 修飾。
7. 適用機種註明在表格的右下角。詳細的指令變化情況，可以對照著表格下方確認指令是否支援脈波執行型，16 位元指令，32 位元指令，64 位元指令。
8. MOV 指令在 ISPSOft 中階梯圖顯示之方塊符號說明如下：

MOV · MOV P · DMOV · DMOV P：表示指令碼

En：表示致能

S：資料來源 (適用運算元格式為 Word/Double Word)

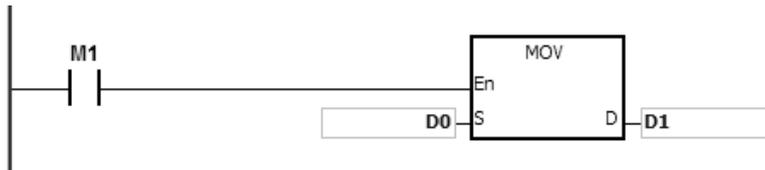
D：資料目的地 (適用運算元格式為 Word/Double Word)

指令組成：

應用指令中有些指令僅有指令部份（指令碼）構成，例如：EI·DI...或 WDT 等等，但是大部份都是指令部份再加上好幾個運算元所組合而成。

每個應用指令均有其指令編號（API）及專用的名稱符號，例如：API0300 的指令碼符號為 MOV（資料搬移）。

直接輸入法：利用階梯圖編輯軟體 ISPSOft 作該指令的輸入，以 MOV 指令而言只需要直接輸入指令的名稱與運算元指定“MOV D0 D1”即可：



拖曳輸入法：選擇 ISPSOft 左側專案管理區內的函式庫，將資料處理分類中“MOV”指令拖曳到階梯圖編輯區即可，運算元再另行指定。

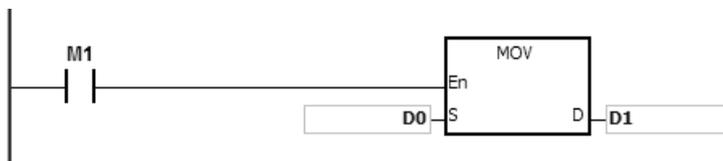
工具列輸入法：選擇 ISPSOft 工具列的運算元/應用指令/功能塊選取按鈕，選取函式庫類別，再選擇資料處理分類中“MOV”指令插入即可，運算元再另行指定。



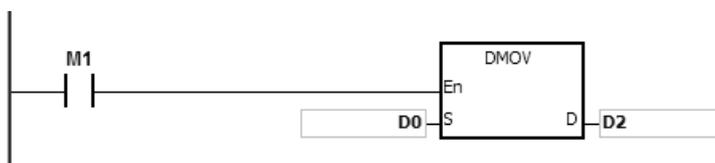
S	來源運算元；若來源運算元有一個以上，那麼則以 S_1 、 S_2 ...分別表示。
D	目的運算元；若目的運算元有一個以上，那麼則以 D_1 、 D_2 ...分別表示。
若運算元只可指定常數 K/H 或暫存器時，那麼則以 m 、 m_1 、 m_2 、 n 、 n_1 、 n_2 表示。	

運算元長度（16 位元指令或 32 位元指令或浮點數指令）：**16 位元指令或 32 位元指令**

運算元的數值內容，其長度可分為 16 位元及 32 位元，因此部份指令處理不同長度的資料則分為 16 及 32 位元的指令，用以區分 32 位元的指令只需要在 16 位元指令前加上“D”來表示即可。

16 位元 MOV 指令

當 M1=ON，D0 被傳送至 D1

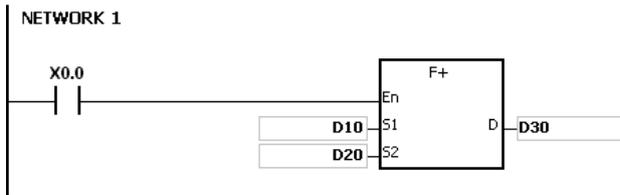
32 位元 DMOV 指令

當 M1=ON 時，(D1·D0) 的內容被傳送至 (D3·D2)

浮點數指令

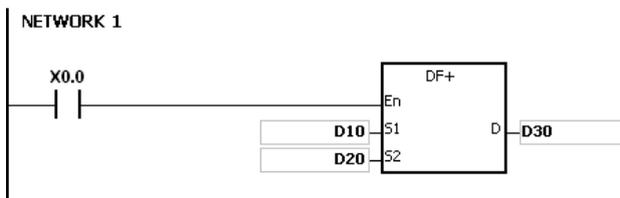
浮點數指令分為 32 位元浮點數指令跟 64 位元浮點數指令，指令中 32 位元浮點數指令為單精度浮點數指令，64 位元浮點數指令為雙精度浮點數指令。詳細的浮點數說明可以參考第二章的浮點數內容。浮點數指令運算元的數值內容表示方式，其長度可分為 32 位元及 64 位元，因此部份指令處理不同長度的資料則分為 32 及 64 位元的指令，浮點數指令的開頭都會以“F”來表示，而用以區分 64 位元的指令只需要在 32 位元指令前加上“D”來表示即可。

32 位元單精度浮點數指令 F+



當 X0.0=ON 時，單精度浮點數 (D11 · D10) + 單精度浮點數 (D21 · D20) 的內容被傳送至 (D31 · D30)

64 位元單精度浮點數指令 DF+

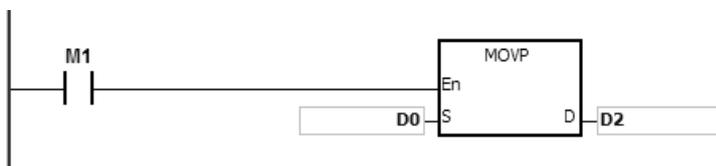


當 X0.0=ON 時，雙精度浮點數 (D13 · D12 · D11 · D10) + 雙精度浮點數 (D23 · D22 · D21 · D20) 的內容被傳送至 (D33 · D32 · D31 · D30)

連續執行型 脈波執行型：

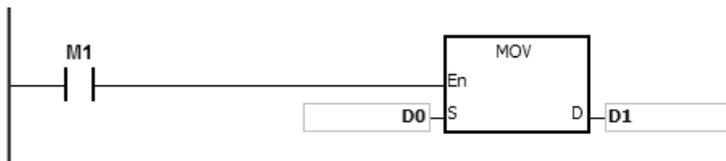
1. 以指令的執行方式來說亦可分成「連續執行型」及「脈波執行型」2 種。由於指令不被執行時，所需的執行時間比較短，因此程式中盡可能的使用脈波執行型指令可減少掃描週期。
2. “脈波”功能可以讓相關的指令啟動上升緣觸發的控制輸入。指令在一個掃描週期內被執行 ON。
3. 之後，若控制輸入保持為 ON，且關聯的指令沒有執行，為了重新執行指令，控制輸入必須再次從 OFF 到 ON。
4. 脈波執行型指令。

脈波執行型



當 M1 由 OFF→ON 變化時，MOVP 指令被執行一次，該次掃描指令不再被執行，因此稱之為脈波執行型指令

連續執行型



於 M1=ON 的每次掃描周期，MOV 指令均被執行一次，因此稱之為連續執行型指令

上圖的條件接點 M1=OFF 時，指令不被執行，目的地運算元 D 的內容沒有變化

運算元的指定對象：

1. 輸入繼電器：X0.0 ~ X511.15 或 X0 ~ X511
 2. 輸出繼電器：Y0.0 ~ Y511.15 或 Y0 ~ Y511
 3. 內部繼電器：M0 ~ M8191
 4. 步進旗標：S0 ~ S2047
 5. 計時器：T0 ~ T2047
 6. 16 位元計數器：C0 ~ C2047
 7. 32 位元計數器：HC0 ~ HC63
 8. 資料暫存器：D0 ~ D65535 或 D0.0 ~ D65535.15
 9. Link 暫存器：L0 ~ L65535 或 L0.0 ~ D65535.15
 10. 特殊旗標：SM0 ~ SM2047
 11. 特殊暫存器：SR0 ~ SR2047
 12. 間接指定暫存器：E0 ~ E31
 13. 指標暫存器：PR0 ~ PR15
 14. 計時器指標暫存器：TR0 ~ TR7
 15. 16 位元計數器指標暫存器：CR0 ~ CR7
 16. 32 位元計數器指標暫存器：HCR0 ~ HCR7
 17. 常數：十進制常數 K，十六進制常數 16#
 18. 字串：“\$”
 19. 浮點數：單精度浮點數 F，雙精度位浮點數 DF
 20. 暫存器一般為 16 位元長度，也就是 1 個暫存器，若指定 32 位元長度的資料暫存器時，是指定連續號碼的 2 個暫存器。
 21. 若 32 位元指令的運算元指定 D0，則 (D1，D0) 所組成的 32 位元資料暫存器被佔用，D1 為上位 16 位元，而 D0 為下位 16 位元。計時器 T，及 16 位元計數器 C 被使用的規則亦相同。
 22. 32 位元計數器 HC 若是當資料暫存器來使用時，只有 32 位元指令的運算元可指定。
- PS.有關裝置的介紹請參考第二章裝置說明。

4.2 指令使用限制

- 只能在功能塊 (Function Block) 中使用的指令
API0065 CHKADR，FB_NP，FB_PN，NED，ANED，ONED，PED，APED，OPED。
- 中斷工作 (TASK) 中不能使用的指令
GOEND。
- 功能塊 (Function Block) 中不支援的指令
LDP，ANDP，ORP，LDF，ANDF，ORF，PLS，PLF，NP，PN，MC/MCR，GOEND，應用命令中所有脈波執行型指令。上述部分指令若還是要使用在功能塊中，可以替代的指令如下：

功能塊中不可使用指令	功能塊中替代指令
LDP/ANDP/ORP	PED/APED/OPED

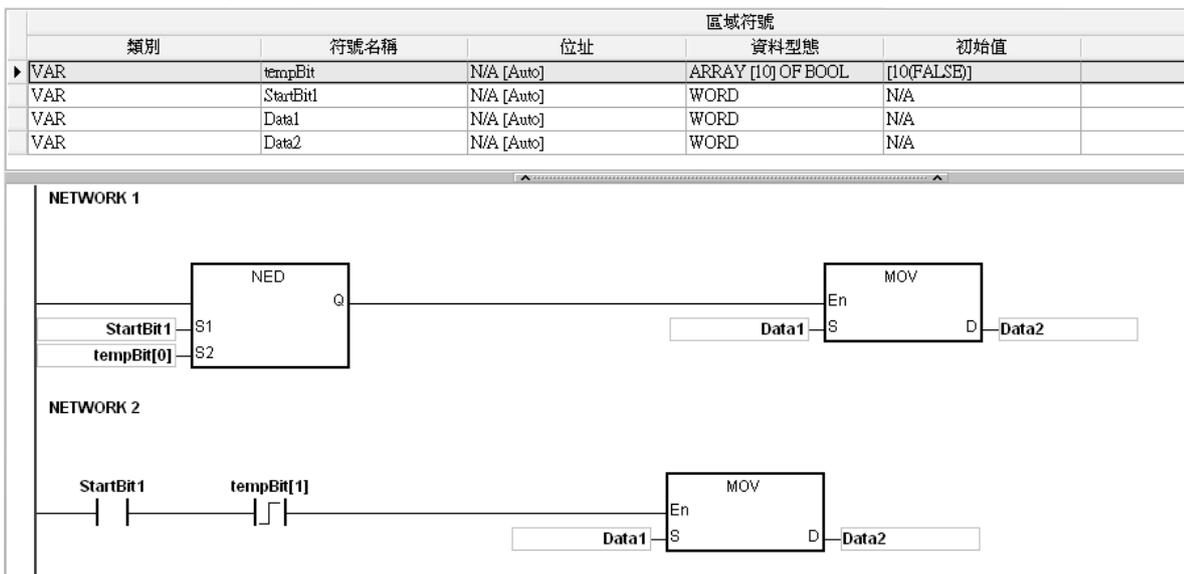
功能塊中不可使用指令	功能塊中替代指令
LDF/ANDF/ORF	NED/ANED/ONED
PLS	-
PLF	-
NP	FB_NP
PN	FB_PN
MC	-
MCR	-
應用命令中所有脈波執行型指令	註一

註一：脈波執行型指令，不可使用在功能塊中，如果要在 FB 功能塊中達到脈波執行型指令功能，可參考下列範例。

4

範例：

1. 先宣告系統用的 10 個 Bit 變數 tempBit[10]。
2. 當 StartBit1 由 ON=>OFF 時，方法一 (NETWORK1) 會執行一次 MOV 指令。
3. 當 StartBit1 由 OFF=>ON 時，方法二 (NETWORK2) 會執行一次 MOV 指令。
4. 系統用的 tempBit 變數，不可重複使用。



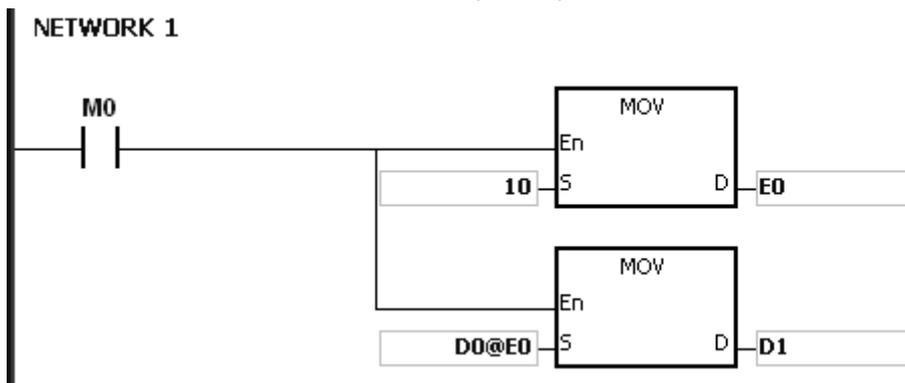
4.3 間接指定說明

間接指定暫存器 E 是 16 位元的資料暫存器，跟一般的暫存器一樣可以被讀、寫，但主要功能是做間接指定暫存器使用，使用範圍為 E0~E31。

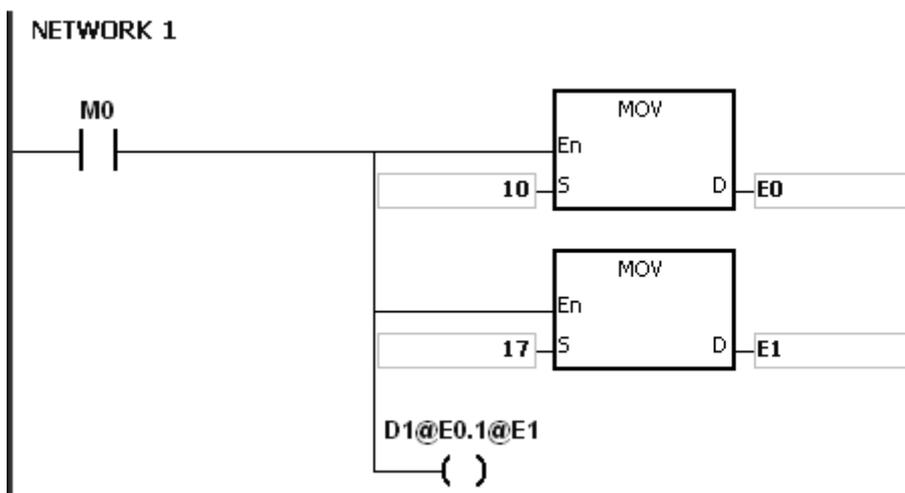
間接指定的使用方式範列如下：

1. 直接使用暫存器名稱來修飾。

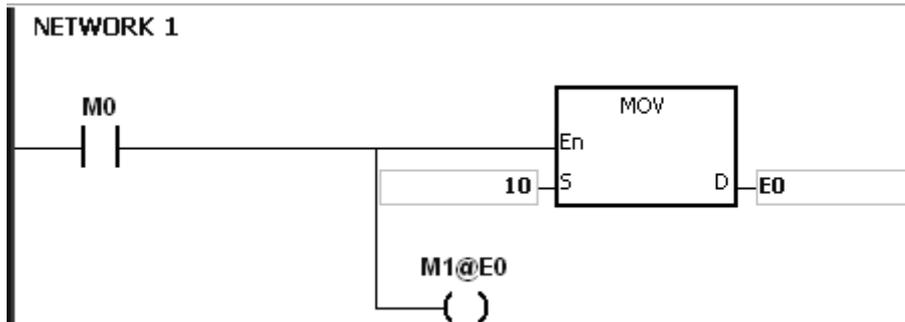
當 M0=ON 時，E0=10，則 $D0@E0=D(0+10)=D10$ ， $D1=D10$ 。



當 M0=ON 時，E0=10，E1=17，則計間接指定的計算方式為 $D1@E0=D(1+10)=D11$ ，而後面 Bit 的部份 $1@E1=(1+17)=18$ ，但 Bit 的部份最大數為 15，則 $m=18/16=1$ ， $n=18\%16=2$ （取餘數），所以最後修飾的結果為 $D(11+m).n=D12.2$ ，則結果 D12.2 會 ON。



當 M0=ON 時，E0=10， $M1@E0=M(1+10)=M11$ ，則 M11=ON。



4

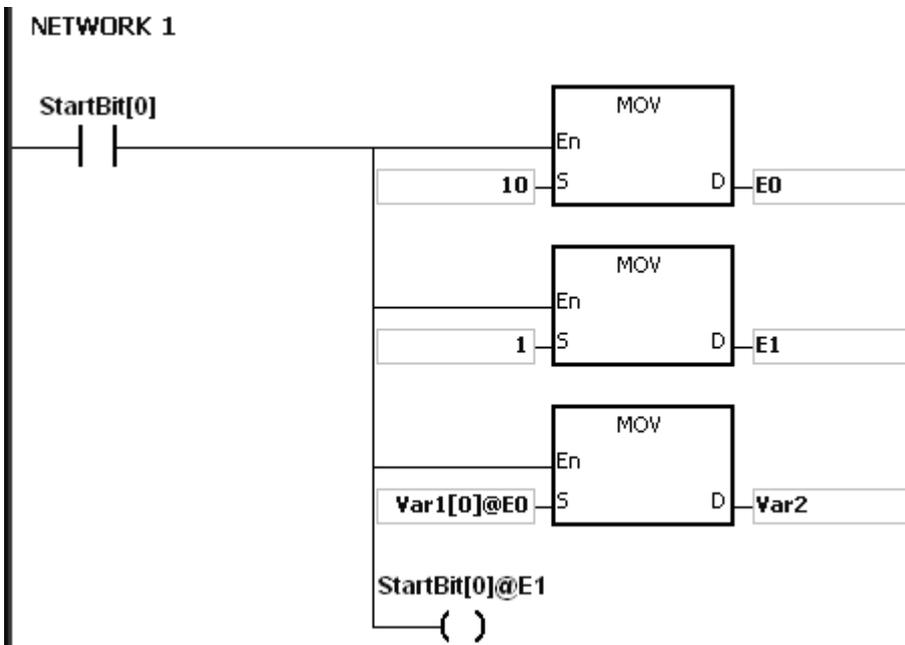
2. 先宣告變數再修飾。

- 先在 ISPSoft 的變數表宣告三個變數 StartBit · Var1 · Var2
 StartBit 的類型為 BOOL 陣列，大小為 2 個 Bit，範圍為 StartBit[0]~StartBit[1]
 Var1 的類型為 WORD 陣列，大小為 11 個 WORD，範圍為 Var1[0]~Var1[10]
 Var2 的類型為 WORD，大小為 1 個 WORD

區域符號				
類別	符號名稱	位址	資料型態	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

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

4



補充說明：當使用 ISPSoft 宣告變數時，又要將變數間接指定偏移，必需注意變數偏移到哪個位址，避免程式造成誤動作。

4.4 指標暫存器 (PR) 說明

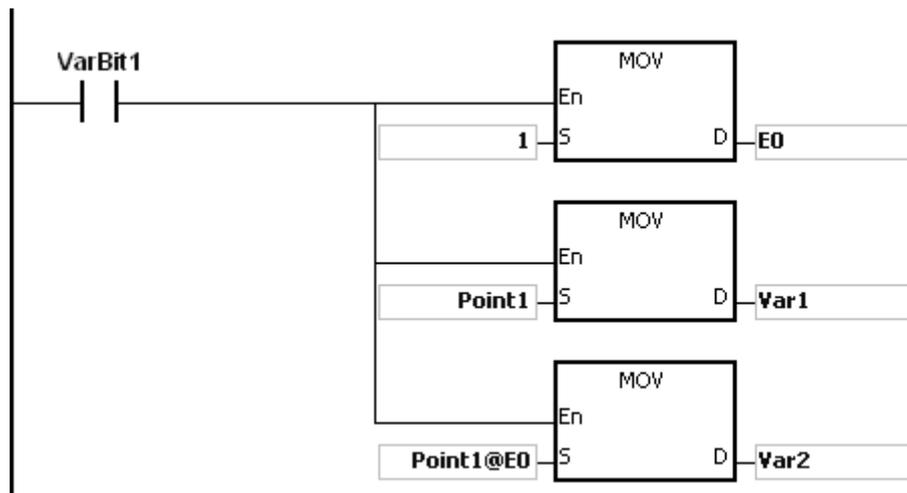
- ISPSoft 有支援功能塊的功能，當功能塊的變數類別宣告為 VAR_IN_OUT，資料型態為 POINTER (指標) 時，此變數將配置 PR 裝置，PR 可以傳入的裝置為 (X · Y · D · L) 以及 Address 為 ISPSoft 自動配置的變數。
- 每個功能塊中可以宣告的 PR 個數為 16 個：PR0 ~ PR15，或是使用位元 PR0.0 ~ PR15.15。

範例：

1. 先使用 ISPSOft 建立一個程式 POU (程式組織單元) 。
2. 建立一個功能塊，名稱為 FB0 。



3. FB0 功能塊中的程式。



4. 在 FB0 功能塊中的變數宣告。

類別選擇 VAR_IN_OUT，符號名稱為 Point1，資料型態選擇 POINTER，此變數將配置為指標暫存器。

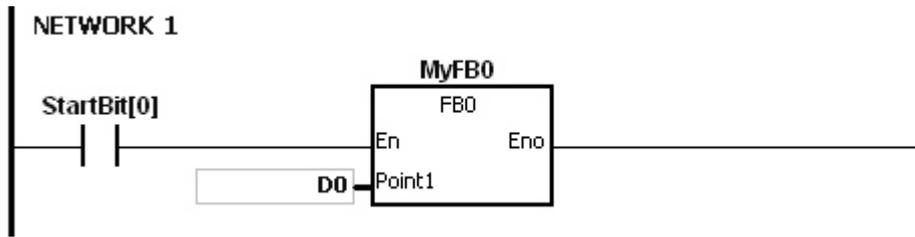
區域符號				
類別	符號名稱	位址	資料型態	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR	Var1	N/A [Auto]	WORD	0
VAR	Var2	N/A [Auto]	WORD	0
VAR_IN_OUT	Point1	N/A [Auto]	POINTER	N/A

5. POU 中的變數宣告。

區域符號				
類別	符號名稱	位址	資料型態	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

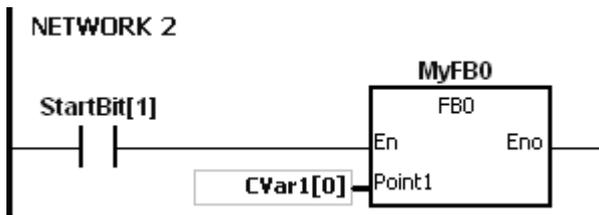
6. 然後在 POU 中呼叫 FB0。
7. POU 中的程式。

方法一：當 StartBit[0]=ON 時，D0 的位址將傳入給 FB0 中的 Point1。



當 FB0 功能塊中的 VarBit1=ON 時， $E0=1$ ， $Var1=D0$ ， $Point1@E0=D(0+1)=D1$ ，所以 $Var2=D1$ 。

方法二：當 StartBit[1]=ON 時，CVar1[0]的位址將傳入給 FB0 中的 Point1。



當 FB0 功能塊中的 VarBit1=ON 時， $E0=1$ ， $Var1=CVar1[0]$ ， $Point1@E0=CVar1(0+1)=Cvar1[1]$ ，所以 $Var2=CVar1[1]$ 。

4

4.5 計時器指標暫存器 (TR) 說明

- ISPSOft 有支援功能塊的功能，如果要在功能塊中使用計時器時，必需在功能塊中宣告一個 TR 裝置來並在呼叫功能塊時傳入 T 裝置的指標。
- 當功能塊的變數類別宣告為 VAR_IN_OUT，資料型態為 T_POINTER 時，此變數將配置 TR 裝置，TR 可以傳入的裝置為 (T) 以及 ISPSOft 配置為計時器的變數。
- 每個功能塊中可以宣告 TR 的個數為 8 個：TR0~TR7。
- 如果要在功能塊中使用指令時，運算元區域中有支援裝置 T，則必須使用 TR 指標暫存器來運算。

範例：

1. 先使用 ISPSOft 建立一個程式 POU (程式組織單元)。
2. 建立一個功能塊，名稱為 FB0。

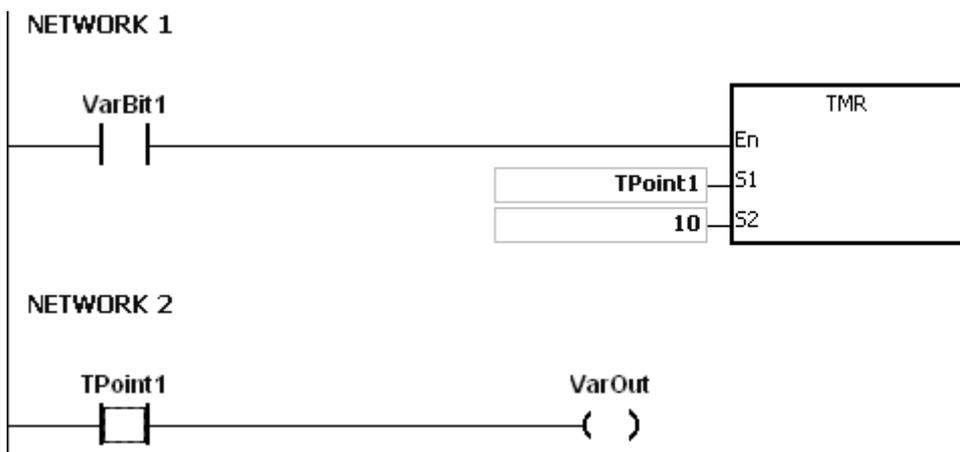


3. 在 FB0 功能塊中的變數宣告。

類別選擇 VAR_IN_OUT，符號名稱為 TPoint1，資料型態選擇 T_POINTER，此變數將配置為計時器指標暫存器。

區域符號				
類別	符號名稱	位址	資料型態	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR_IN_OUT	TPoint1	TR0	T_POINTER	N/A
VAR	VarOut	N/A [Auto]	BOOL	FALSE

4. FB0 功能塊中的程式。



4

5. POU 中的變數宣告。

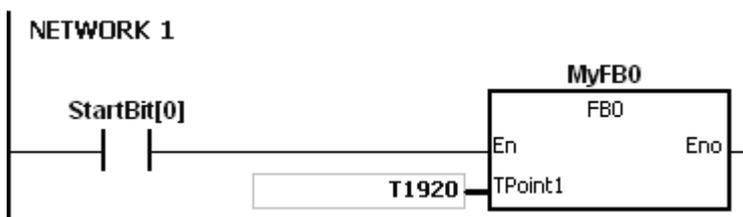
其中 CVar1 必需宣告為 TIMER 的資料型態

區域符號				
類別	符號名稱	位址	資料型態	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. 然後在 POU 中呼叫 FB0。

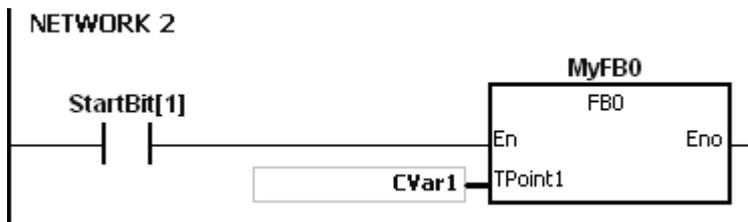
7. POU 中的程式。

方法一：當 StartBit[0]=ON 時，T1920 的位址將傳入給 FB0 中的 TPoint1。



當 FB0 功能塊中的 VarBit1=ON 時，TMR 指令執行，TPoint1 (T1920) 開始計數，當 TPoint1 計數到達時 VarOut=ON。

方法二：當 StartBit[1]=ON 時，CVar1[0]的位址將傳入給 FB0 中的 TPoint1。



當 FB0 功能塊中的 VarBit1=ON 時，TMR 指令執行，TPoint1 (CVar1) 開始計數，當 TPoint1 計數到達時 VarOut=ON。

4.6 16 位元計數器指標暫存器 (CR) 說明

- ISPSOft 有支援功能塊的功能，如果要在功能塊中使用 16 位元計數器時，必需在功能塊中宣告一個 CR 裝置，並在呼叫功能塊時傳入 C 裝置的指標。
- 當功能塊的變數類別宣告為 VAR_IN_OUT，資料型態為 C_POINTER 時，此變數將配置 CR 裝置，CR 可以傳入的裝置為 (C) 以及 ISPSOft 配置為計數器的變數。
- 每個功能塊中可以宣告 CR 的個數為 8 個：CR0~CR7。
- 如果要在功能塊中使用指令時，運算元區域中有支援裝置 C，則必須使用 CR 指標暫存器來運算。

範例：

1. 先使用 ISPSOft 建立一個程式 POU (程式組織單元)。
2. 建立一個功能塊，名稱為 FB0。



3. 在 FB0 功能塊中的變數宣告。
類別選擇 VAR_IN_OUT，符號名稱為 CPoint1，資料型態選擇 C_POINTER，此變數將配置為計數器指標暫存器。

區域符號				
類別	符號名稱	位址	資料型態	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR_IN_OUT	CPoint1	CR0	C_POINTER	N/A

4. FB0 功能塊中的程式。



5. POU 中的變數宣告。

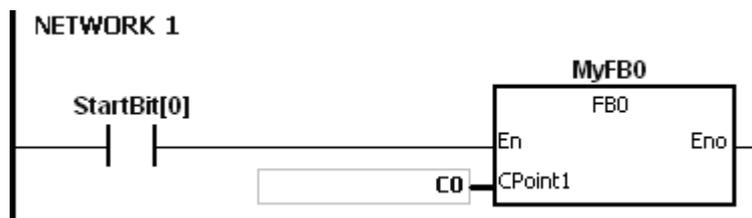
其中 CVar1 必需宣告為 COUNTER 的資料型態。

區域符號				
類別	符號名稱	位址	資料型態	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. 然後在 POU 中呼叫 FB0。

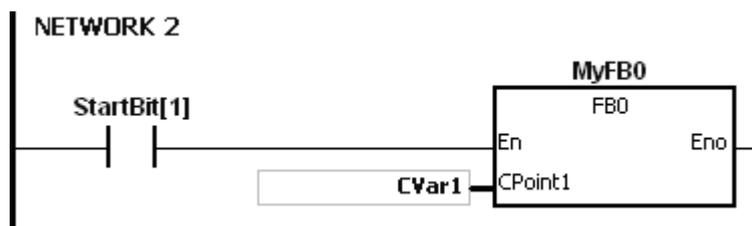
7. POU 中的程式。

方法一：當 StartBit[0]=ON 時，C0 的位址將傳入給 FB0 中的 CPoint1。



當 FB0 功能塊中的 VarBit1=ON 時，CPoint1 (C0) =ON。

方法二：當 StartBit[1]=ON 時，CVar1 的位址將傳入給 FB0 中的 CPoint1。



當 FB0 功能塊中的 VarBit1=ON 時，CPoint1 (CVar1) =ON。

4

4.7 32 位元計數器指標暫存器 (HCR) 說明

- ISPSOft 有支援功能塊的功能，如果要在功能塊中使用 32 位元計數器時，必需在功能塊中宣告一個 HCR 裝置，並在呼叫功能塊時傳入 HC 裝置的指標。
- 當功能塊的變數類別宣告為 VAR_IN_OUT，資料型態為 HC_POINTER 時，此變數將配置 HCR 裝置，HCR 可以傳入的裝置為 (HC) 以及 ISPSOft 配置為計數器的變數。
- 每個功能塊中可以宣告 HCR 的個數為 8 個：HCR0~HCR7。
- 如果要在功能塊中使用指令時，運算元區域中有支援裝置 HC，則必須使用 HCR 指標暫存器來運算。

範例：

1. 先使用 ISPSOft 建立一個程式 POU (程式組織單元)。
2. 建立一個功能塊，名稱為 FB0。



- 在 FB0 功能塊中的變數宣告。

類別選擇 VAR_IN_OUT，符號名稱為 HCPoint1，資料型態選擇 HC_POINTER，此變數將配置為計數器指標暫存器。

區域符號				
類別	符號名稱	位址	資料型態	初始值
VAR	VarBit1	N/A [Auto]	BOOL	FALSE
VAR_IN_OUT	HCPoint1	N/A [Auto]	HC_POINTER	N/A

- FB0 功能塊中的程式。



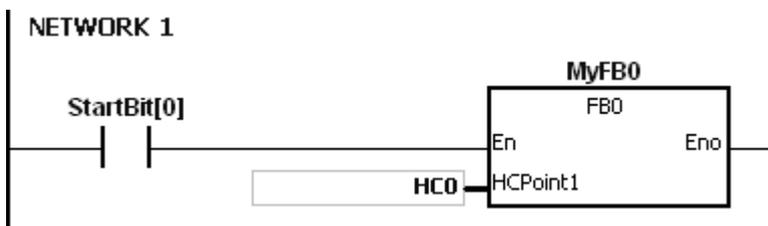
- POU 中的變數宣告。

其中 CVar1 必需宣告為 COUNTER 的資料型態，並自行在位址欄位中填入 HC 裝置的實際位址。

區域符號				
類別	符號名稱	位址	資料型態	初始值
VAR	StartBit	N/A [Auto]	ARRAY [2] OF E	[2](FALSE)]
VAR	CVar1	HC1	COUNTER	N/A
VAR	MyFB0	N/A [Auto]	FB0	N/A

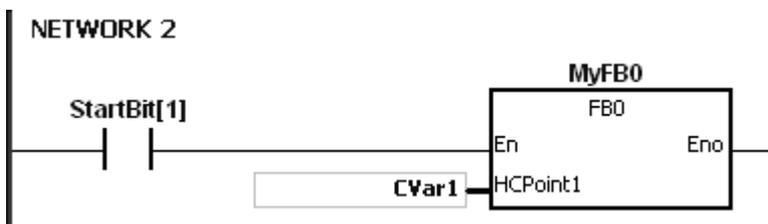
- 然後在 POU 中呼叫 FB0。
- 在 POU 中呼叫 FB0。

方法一：當 StartBit[0]=ON 時，HC0 的位址將傳入給 FB0 中的 HCPoint1。



當 FB0 功能塊中的 VarBit1=ON 時，HCPoint1 (HC0) =ON。

方法二：當 StartBit[1]=ON 時，CVar1 的位址將傳入給 FB0 中的 HCPoint1。



當 FB0 功能塊中的 VarBit1=ON 時，HCPoint1 (CVar1) =ON。

5

第5章 基本指令

目錄

5.1 基本指令一覽表.....	5-2
5.2 基本指令說明.....	5-3

5.1 基本指令一覽表

指令碼	功能	運算元	STEPS
<u>LD/AND/OR</u>	載入/串聯/並聯 A 接點	DX、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>LDI/ANI/ORI</u>	載入/串聯/並聯 B 接點	DX、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>ANB/ORB</u>	串聯/並聯迴路方塊	—	1
<u>MPS/MRD/MPP</u>	存入/讀出/讀出堆疊	—	1
<u>OUT</u>	驅動線圈	DY、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>SET</u>	動作保持 (ON)	DY、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>MC/MCR</u>	共通串聯接點之連結/解除	N	1
<u>LDP/ANDP/ORP</u>	正緣檢出動作開始/串聯/並聯連接	DX、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>LDF/ANDF/ORF</u>	負緣檢出動作開始/串聯/並聯連接	DX、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>PED/APED/OPED</u>	正緣檢出動作開始/串聯/並聯連接	X、Y、M、SM、S、T、C、HC、 D、L、PR	5
<u>NED/ANED/ONED</u>	負緣檢出動作開始/串聯/並聯連接	X、Y、M、SM、S、T、C、HC、 D、L、PR	5
<u>PLS</u>	上微分輸出	DY、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>PLF</u>	下微分輸出	DY、X、Y、M、SM、S、T、C、 HC、D、L、PR	1-2
<u>INV</u>	運算結果反相	—	1
<u>NOP</u>	無動作	—	1
<u>NP</u>	上升緣觸發指令	—	1
<u>PN</u>	下降緣觸發指令	—	1
<u>FB_NP</u>	上升緣觸發指令	X、Y、M、SM、S、T、C、HC、 D、L、PR	1-2
<u>FB_PN</u>	下降緣觸發指令	X、Y、M、SM、S、T、C、HC、 D、L、PR	1-2
<u>PSTOP</u>	PLC 程式停止執行	—	1

5.2 基本指令說明

指令碼		運算元										功能		
LD/AND/OR		S										載入/串聯/並聯 A 接點		
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR	
S	●		●	●	●	●	●	●	●	●	●	●	●	

符號：

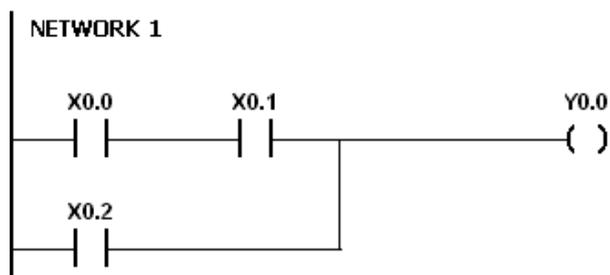


指令說明：

- LD 指令用於左母線開始的 A 接點或一個接點回路塊開始的 A 接點，它的作用是把當前內容保存，同時把取來的接點狀態存入累積暫存器內。
- AND 指令用於 A 接點的串聯連接，先讀取目前所指定串聯接點的狀態再與接點之前邏輯運算結果作“及”（AND）的運算，並將結果存入累積暫存器內。
- OR 指令用於 A 接點的並聯連接，它的作用是先讀取目前所指定串聯接點的狀態再與接點之前邏輯運算結果作“或”（OR）的運算，並將結果存入累積暫存器內。

程式範例：

- 載入 X0.0 的 A 接點，串聯 X0.1 的 A 接點，並聯 X0.2 的 A 接點，驅動 Y0.0 線圈。
- 當 X0.0 和 X0.1=ON 或 X0.2=ON 時，Y0.0=ON。



指令碼	運算元	功能
LDI/ANI/ORI	S	載入/串聯/並聯 B 接點

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

符號：

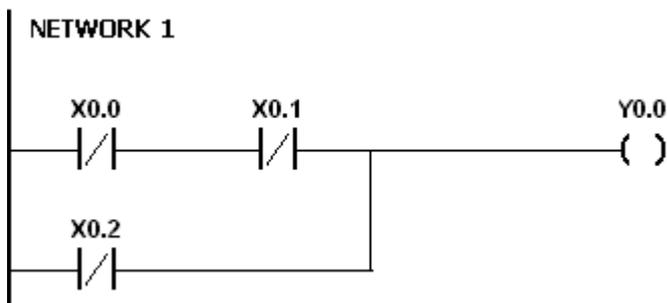


指令說明：

- LDI 指令用於左母線開始的 B 接點或一個接點回路塊開始的 B 接點，它的作用是把當前內容保存，同時把取來的接點狀態存入累積暫存器內。
- ANI 指令用於 B 接點的串聯連接，先讀取目前所指定串聯接點的狀態再與接點之前邏輯運算結果作“及” (AND) 的運算，並將結果存入累積暫存器內。
- ORI 指令用於 B 接點的並聯連接，它的作用是先讀取目前所指定串聯接點的狀態再與接點之前邏輯運算結果作“或” (OR) 的運算，並將結果存入累積暫存器內。

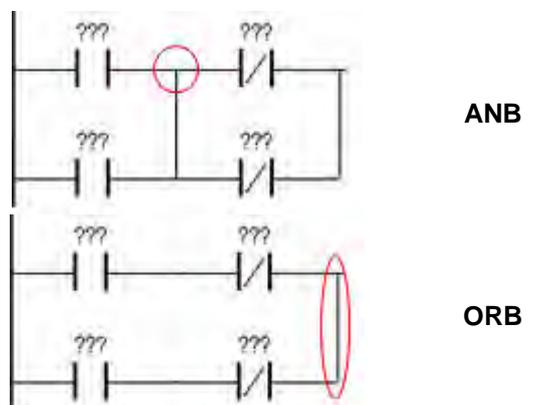
程式範例：

- 載入 X0.0 的 B 接點，串聯 X0.1 的 B 接點，並聯 X0.2 的 B 接點，驅動 Y0.0 線圈。
- 當 X0.0 和 X0.1=OFF 或 X0.2=OFF 時，Y0.0=ON。



指令碼	運算元	功能
ANB/ORB	-	串聯/並聯迴路方塊

符號：

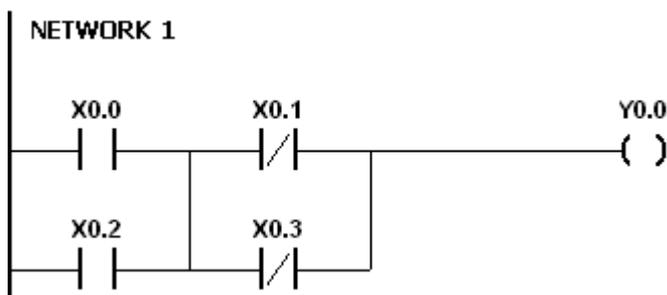


指令說明：

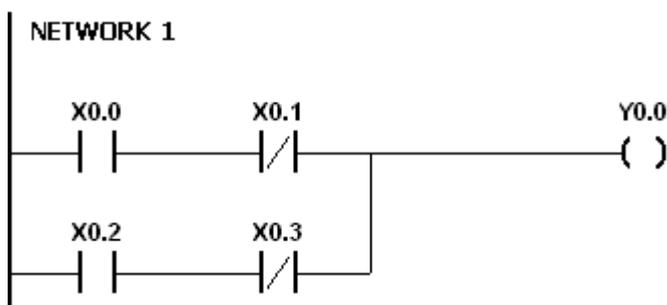
1. ANB 是將前一保存的邏輯結果與目前累積暫存器的內容作“及” (AND) 的運算。
2. ORB 是將前一保存的邏輯結果與目前累積暫存器的內容作“或” (OR) 的運算。

程式範例：

1. 載入 A 接點 X0.0，並聯 A 接點 X0.2，載入 B 接點 X0.1，並聯 B 接點 X0.3，串聯回路方塊，驅動 Y0.0 線圈。



2. 載入 A 接點 X0.0，串聯 B 接點 X0.1，載入 A 接點 X0.2，串聯 B 接點 X0.3，並聯回路方塊，驅動 Y0.0 線圈。



指令碼	運算元	功能
MPS/MRD/MPP	-	存入/讀出/讀出堆疊

指令說明：

1. MPS 存入堆疊指令，將目前累積暫存器的內容存入堆疊（堆疊指標加一）。
2. MRD 讀出指令，讀取堆疊內容存入累積暫存器（堆疊指標不動）。
3. MPP 讀出指令，自堆疊取回前一保存的邏輯運算結果，存入累積暫存器（堆疊指標減一）。

程式範例：

1. 載入 X0.0 的 A 接點，存入堆疊。
2. 串聯 X0.1 的 A 接點，驅動 Y0.1 線圈，讀出堆疊（指標不動）。
3. 串聯 X0.2 的 A 接點，驅動 M0 線圈，讀出堆疊。

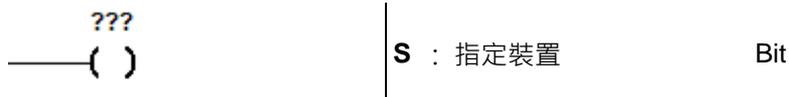
指令：	操作說明：
LD X0.0	載入X0.0的A接點
MPS	存入堆疊
AND X0.1	串聯X0.1的A接點
OUT Y0.1	驅動Y0.1線圈
MRD	讀出堆疊（指標不動）
AND X0.2	串聯X0.2的A接點
OUT M0	驅動M0線圈
MPP	讀出堆疊
OUT Y0.2	驅動Y0.2線圈
END	程式結束

備註：

1. MPS 與 MPP 要一一對應。
2. MPS 指令最多可以連續使用 31 次。

指令碼		運算元						功能					
OUT		S						驅動線圈					
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S		●	●	●	●	●	●	●	●	●	●	●	●

符號：



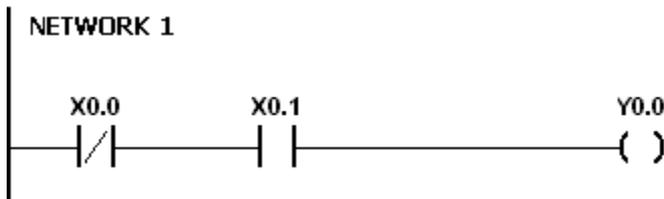
指令說明：

1. 將 OUT 指令之前的邏輯運算結果輸出至指定的元件。
2. 線圈接點動作：

運算結果	OUT 指令		
	線圈	接點	
		A 接點 (常開)	B 接點 (常閉)
FALSE	OFF	不導通	導通
TRUE	ON	導通	不導通

程式範例：

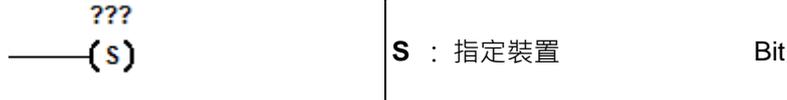
1. 載入 X0.0 的 B 接點，串聯 X0.1 的 A 接點，驅動 Y0.0 線圈。
2. 當 X0.0=OFF 且 X0.1=ON 時，Y0.0=ON。



指令碼	運算元	功能
SET	S	動作保持 (ON)

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

符號：

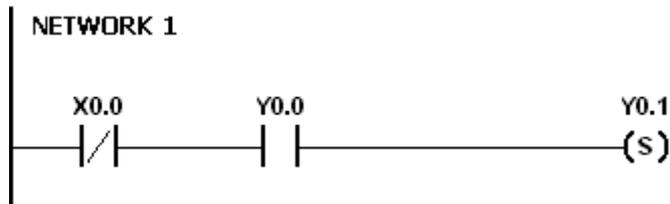


指令說明：

當 SET 指令被驅動，其指定的元件被設定為 ON，且被設定的元件會維持 ON，不管 SET 指令是否仍被驅動。可利用 RST 指令將該元件設為 OFF。

程式範例：

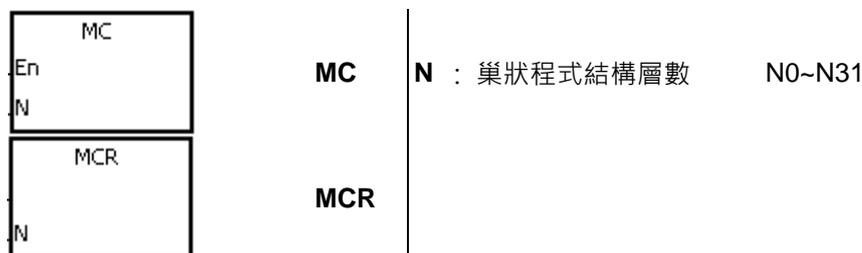
1. 載入 X0.0 之 B 接點，串聯 Y0.0 之 A 接點，Y0.1 動作保持 (ON)。
2. 當 X0.0=OFF 且 Y0.0=ON 時，Y0.1=ON 且即使運算結果改變，Y0.1 亦保持 ON 的狀態。



5

指令碼	運算元	功能
MC/MCR	N	共通串聯接點之連結/解除

符號：



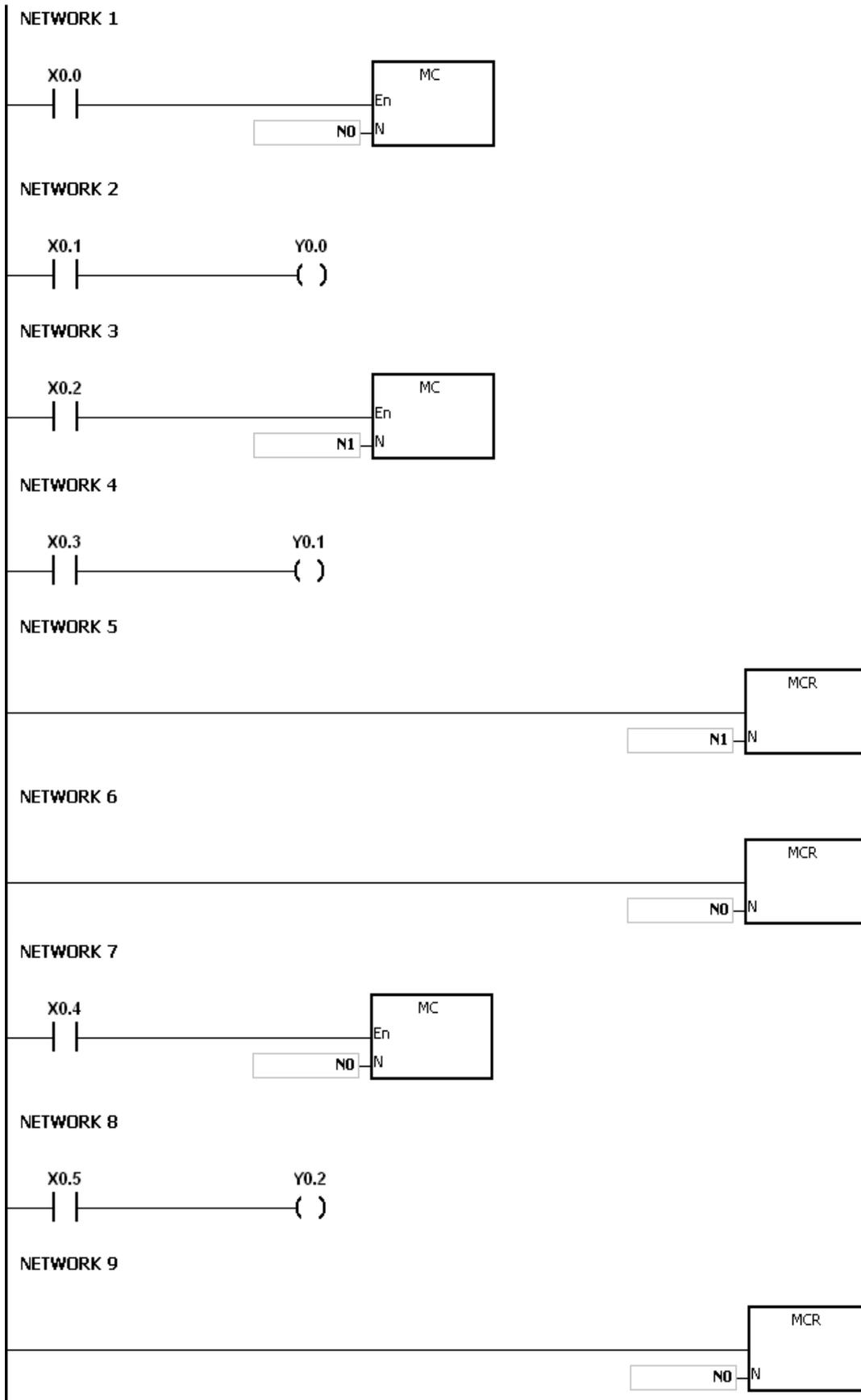
指令說明：

1. MC 為主控起始指令，當 MC 指令執行時，位於 MC 與 MCR 指令之間的指令照常執行。當 MC 指令 OFF 時，位於 MC 與 MCR 指令之間的指令動作如下所示：

指令區分	說明
一般計時器	計時值歸零，線圈失電，接點不動作
功能塊用計時器	計時值歸零，線圈失電，接點不動作
積算型計時器	線圈失電，計時值及接點保持目前狀態
計數器	線圈失電，計數值及接點保持目前狀態
OUT 指令驅動的線圈	全部不受電
SET、RST 指令驅動的元件	保持目前狀態
應用指令	全部不動作，但 FOR-NEXT 巢串迴路仍會來回執行 N 次，但 FOR-NEXT 間的任何指令依 MC-MCR 之間其它指令相同動作

2. MCR 為主控結束指令，置於主控程式最後，在 MCR 指令之前不可有接點指令。
3. MC-MCR 主控程式指令支援巢狀程式結構，最多可 32 層，使用時依 N0~N31 的順序，請參考程式範例所示。

程式範例：



5

指令碼		運算元						功能					
LDP/ANDP/ORP		S						正緣檢出動作開始/串聯/並聯連接					
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S	●		●	●	●	●	●	●	●	●	●	●	●

符號：

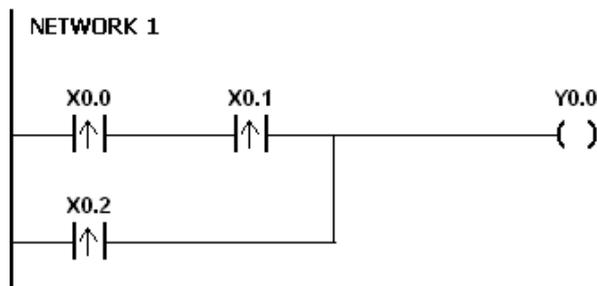


指令說明：

1. LDP 指令用法上與 LD 相同，但動作不同，它的作用是指當前內容保存，同時把取來的接點上升緣檢出狀態存入累積暫存器內。
2. ANDP 指令用於接點上升緣檢出的串聯連接。
3. ORP 指令用於接點上升緣檢出的並聯連接。
4. 正緣檢出動作，必須在指令掃描到的時候才會得知裝置目前的狀態，下一次掃描到指令才會判斷裝置狀態是否有變化。
5. 副程式中請使用 PED、APED、OPED 指令。

程式範例：

1. X0.0 上升緣檢出動作開始，串聯 X0.1 的上升緣檢出，並聯 X0.2 的上升緣檢出，驅動 Y0.0 線圈。
2. 當 X0.0 和 X0.1 同時由 OFF 到 ON 或 X0.2 由 OFF 到 ON 時，Y0.0 會 ON 一個掃描週期。



指令碼	運算元	功能
LDF/ANDF/ORF	S	負緣檢出動作開始/串聯/並聯連接

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

符號：

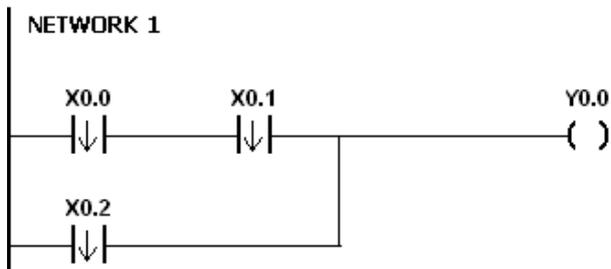


指令說明：

1. LDF 指令用法上與 LD 相同，但動作不同，它的作用是指當前內容保存，同時把取來的接點下降緣檢出狀態存入累積暫存器內。
2. ANDF 指令用於接點下降緣檢出的串聯連接。
3. ORF 指令用於接點下降緣檢出的並聯連接。
4. 負緣檢出動作，必須在指令掃描到的時候才會得知裝置目前的狀態，下一次掃描到指令才會判斷裝置狀態是否有變化
5. 副程式中請使用 NED、ANED、ONED 指令。

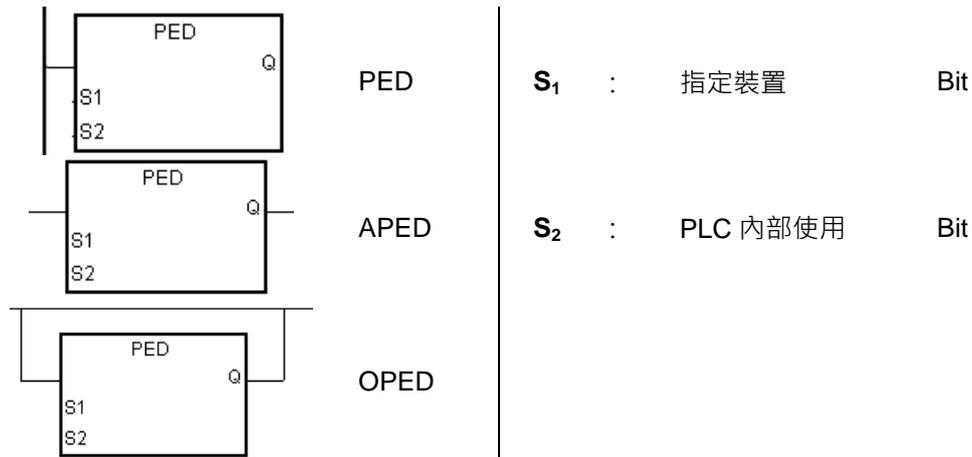
程式範例：

1. X0.0 下降緣檢出動作開始，串聯 X0.1 的下降緣檢出，並聯 X0.2 的下降緣檢出，驅動 Y0.0 線圈。
2. 當 X0.0 和 X0.1 同時由 ON 到 OFF 或 X0.2 由 ON 到 OFF 時，Y0.0 會 ON 一個掃描週期。



指令碼		運算元		功能									
PED/APED/OPED		$S_1 \cdot S_2$		正緣檢出動作開始/串聯/並聯連接									
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S_1			●	●	●	●	●	●	●	●	●	●	●
S_2			●	●	●	●	●	●	●	●	●	●	●

符號：

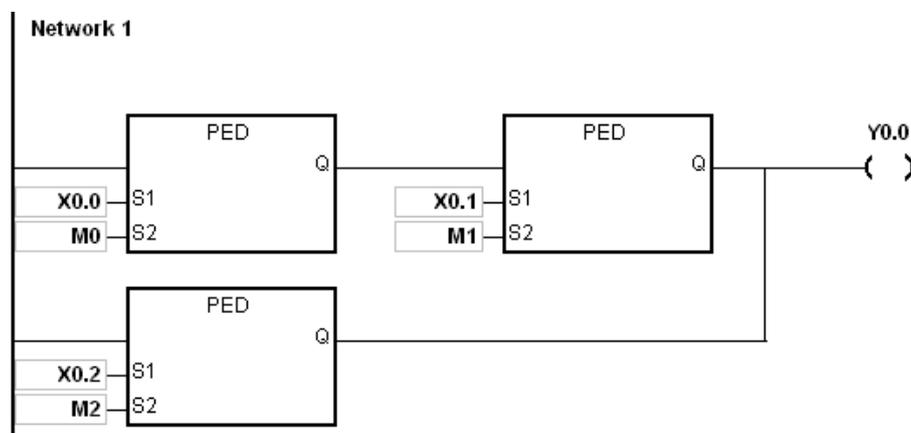


指令說明：

1. PED 指令用法上與 LDP 相同、APED 指令用法上與 ANDP 相同、OPED 指令用法上與 ORP 相同，不同之處在於需多指定一個 S_2 的 Bit 裝置，讓 PLC 內部記住該接點的上一次狀態，此 S_2 裝置在程式中的其它地方請勿重覆使用，否則可能會造成執行結果錯誤。
2. APED 指令用於接點上升緣檢出的串聯連接。
3. OPED 指令用於接點上升緣檢出的並聯連接。
4. 正緣檢出動作，必須在指令掃描到的時候才會得知裝置目前的狀態，下一次掃描到指令才會判斷裝置狀態是否有變化。
5. PED/APED/OPED 指令只能在功能塊中使用。

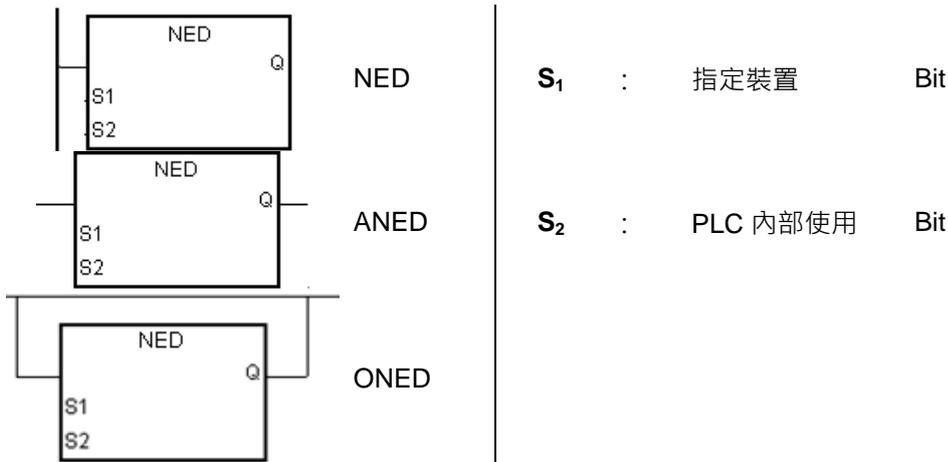
程式範例：

1. X0.0 上升緣檢出動作開始，串聯 X0.1 的上升緣檢出，並聯 X0.2 的上升緣檢出，驅動 Y0.0 線圈。
2. 當 X0.0 和 X0.1 同時由 OFF 到 ON 或 X0.2 由 OFF 到 ON 時，Y0.0 會 ON 一個掃描週期。



指令碼		運算元		功能									
NED/ANED/ONED		$S_1 \cdot S_2$		負緣檢出動作開始/串聯/並聯連接									
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S_1			●	●	●	●	●	●	●	●	●	●	●
S_2			●	●	●	●	●	●	●	●	●	●	●

符號：

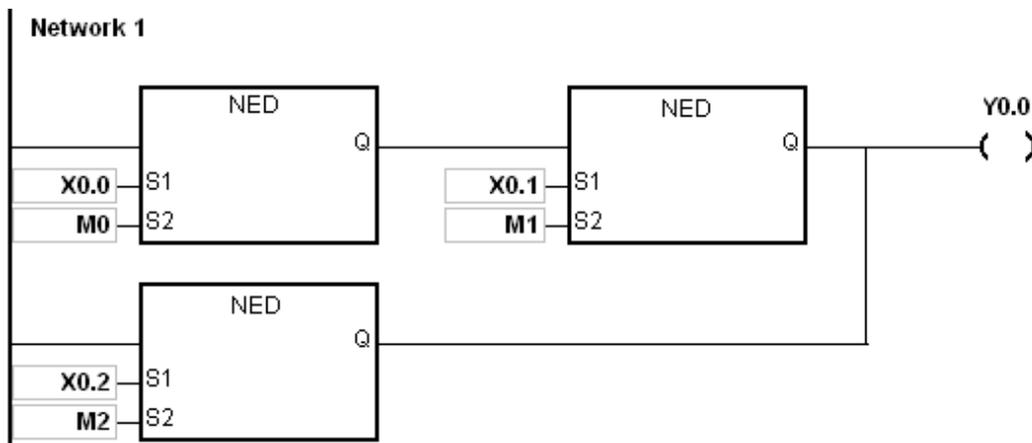


指令說明：

1. NED 指令用法上與 LDF 相同、ANED 指令用法上與 ANDF 相同、ONED 指令用法上與 ORF 相同，不同之處在於需多指定一個 S_2 的 Bit 裝置，讓 PLC 內部記住該接點的上一次狀態，此 S_2 裝置在程式中的其它地方請勿重覆使用，否則可能會造成執行結果錯誤。
2. ANED 指令用於接點下降緣檢出的串聯連接。
3. ONED 指令用於接點下降緣檢出的並聯連接。
4. 負緣檢出動作，必須在指令掃描到的時候才會得知裝置目前的狀態，下一次掃描到指令才會判斷裝置狀態是否有變化。
5. NED/ANED/ONED 指令只能在功能塊中使用。

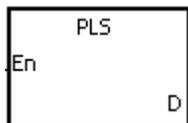
程式範例：

1. X0.0 下降緣檢出動作開始，串聯 X0.1 的下降緣檢出，並聯 X0.2 的下降緣檢出，驅動 Y0.0 線圈。
2. 當 X0.0 和 X0.1 同時由 ON 到 OFF 或 X0.2 由 ON 到 OFF 時，Y0.0 會 ON 一個掃描週期。



指令碼		運算元										功能		
PLS		D										上微分輸出		
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR	
S		●	●	●	●	●	●	●	●	●	●	●	●	

符號：



PLS



D : 指定裝置

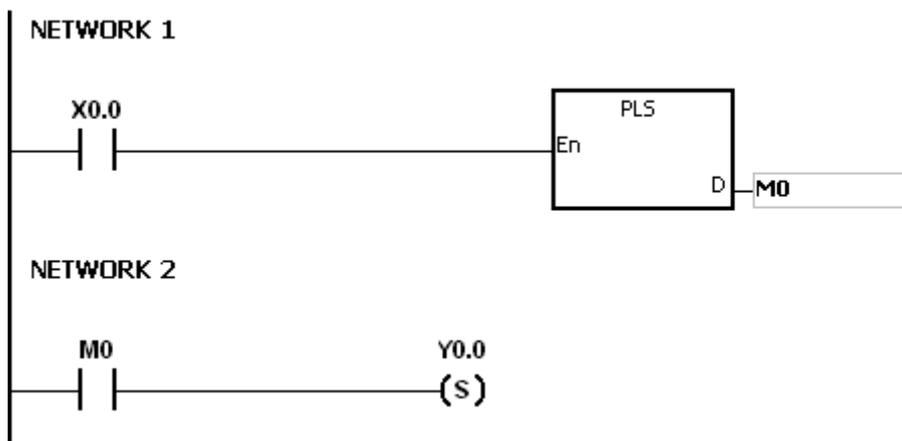
Bit

指令說明：

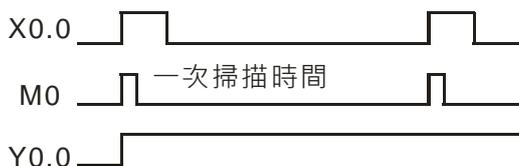
1. 上微分輸出指令。當條件接點由 OFF 到 ON (正緣觸發) 時, PLS 指令被執行, D 送出一個脈波, 脈波長度為一次掃描時間。
2. 功能塊中請勿使用。

程式範例：

當 X0.0=ON 時, M0 ON 一個 Pulse 的時間; M0=ON 時, Y0.0 Set 為 ON。



時序圖：



指令碼	運算元	功能
PLF	D	下微分輸出

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

符號：



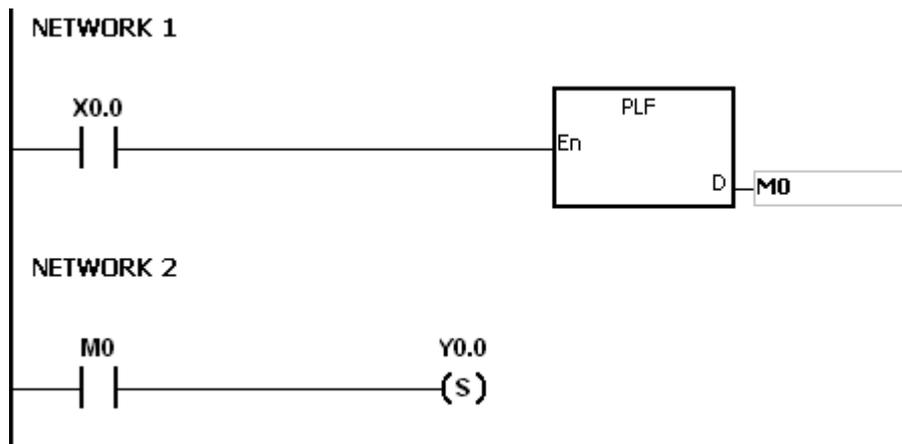
指令說明：

1. 下微分輸出指令。當條件接點由 ON 到 OFF (負緣觸發) 時，PLF 指令被執行，D 送出一次脈波，脈波長度為一次掃描時間。
2. 功能塊中請勿使用。

程式範例：

當 X0.0=ON 到 OFF 時，M0 ON 一個 Pulse 的時間；M0=ON 時，Y0.0 Set 為 ON。

5



時序圖：



指令碼	運算元	功能
INV	-	運算結果反相

符號：

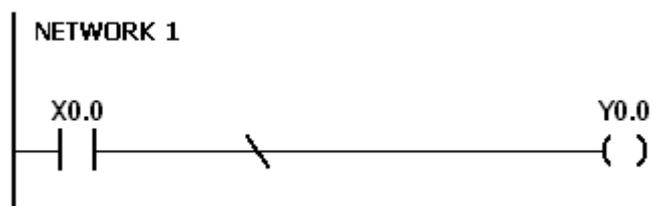


指令說明：

將 INV 指令之前的邏輯運算結果反相存入累積暫存器內。

程式範例：

當 X0.0=ON 時，Y0.0=OFF；X0.0=OFF 時，Y0.0=ON。



指令碼	運算元	功能
NOP	無	無動作

符號：無

指令說明：

指令 **NOP** 在程式不做任何運算，因此執行後仍會保持原邏輯運算結果，使用時機如下：想要刪除某一指令，而又不想改變程式長度，則可以 **NOP** 指令取代。

本指令僅支援 ISPSOft 的指令列表 (Instruction List) 功能編輯，不支援階梯圖編輯。

程式範例：

ISPSOft 的指令列表中

指令：	操作說明：
LD X0.0	載入 X0.0 的 A 接點
NOP	無動作
OUT Y1.0	驅動線圈 Y1.0

5

指令碼	運算元	功能
NP	-	上升緣觸發指令

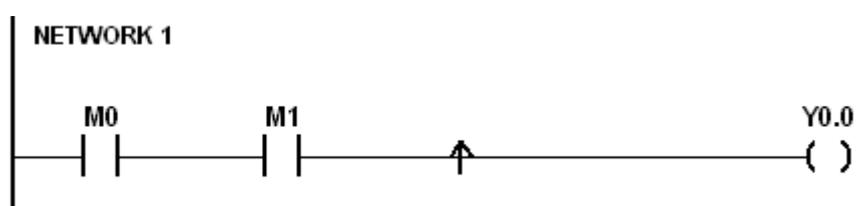
符號：



指令說明：

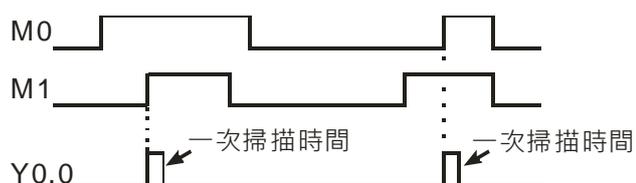
- 當累積暫存器由 0 變為 1 時，此指令將使累積暫存器維持一次掃描週期的 1，然後第二次掃描週期之後，自動將累積暫存器改為 0。
- 在 Function Block 中請使用 FB_NP 指令。

程式範例：



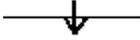
指令：		操作說明：
LD	M0	載入 M0 的 A 接點
AND	M1	串入 M1 之 A 接點
NP		上升緣觸發指令
OUT	Y0.0	驅動線圈 Y0.0

時序圖：



指令碼	運算元	功能
PN	-	下降緣觸發指令

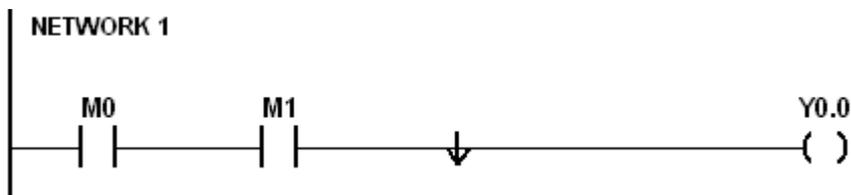
符號：



指令說明：

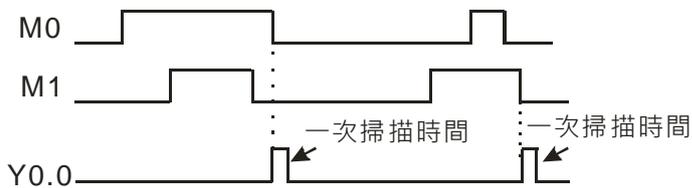
1. 當累積暫存器由 1 變為 0 時，此指令將使累積暫存器維持一次掃描週期的 1，然後第二次掃描週期之後，自動將累積暫存器改為 0。
2. 在 Function Block 中請使用 FB_P N 指令。

程式範例：



指令：	操作說明：
LD M0	載入 M0 的 A 接點
AND M1	串入 M1 之 A 接點
PN	下降緣觸發指令
OUT Y0.0	驅動線圈 Y0.0

時序圖：



指令碼		運算元										功能	
FB_NP		S										上升緣觸發指令	
裝置	DX	DY	X	Y	M	SM	S	T	C	HC	D	L	PR
S			●	●	●	●	●	●	●	●	●	●	●

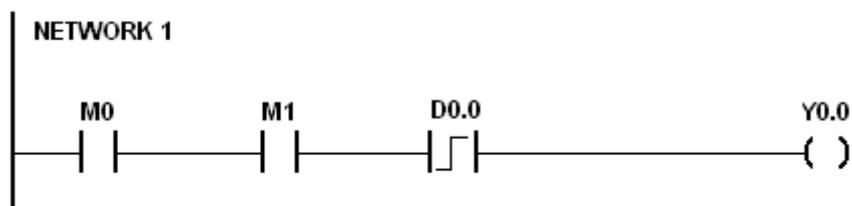
符號：



指令說明：

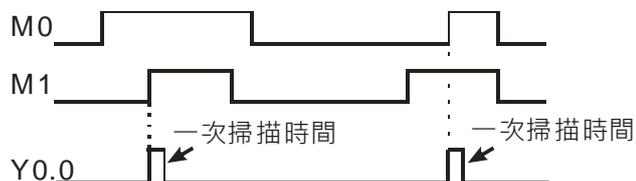
1. 當累積暫存器由 0 變為 1 時，此指令將使累積暫存器維持一次掃描週期的 1，然後第二次掃描週期之後，自動將累積暫存器改為 0。
2. **S** 裝置，讓 PLC 內部記住該接點的上一次狀態，此 **S** 裝置在程式中的其它地方請勿重覆使用，否則可能會造成執行結果錯誤。
3. 本指令只能在 Function Block 中使用。

程式範例：



指令：	操作說明：
LD M0	載入 M0 的 A 接點
AND M1	串入 M1 之 A 接點
FB_NP D0.0	上升緣觸發指令
OUT Y0.0	驅動線圈 Y0.0

時序圖：



指令碼	運算元	功能
FB_PN	S	下降緣觸發指令

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

符號：

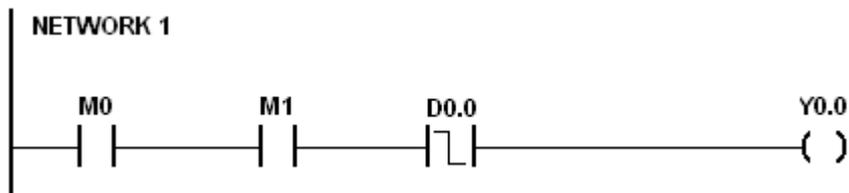


指令說明：

- 當累積暫存器由 1 變為 0 時，此指令將使累積暫存器維持一次掃描週期的 1，然後第二次掃描週期之後，自動將累積暫存器改為 0。
- S** 裝置，讓 PLC 內部記住該接點的上一次狀態，此 **S** 裝置在程式中的其它地方請勿重覆使用，否則可能會造成執行結果錯誤。
- 本指令只能在 Function Block 中使用。

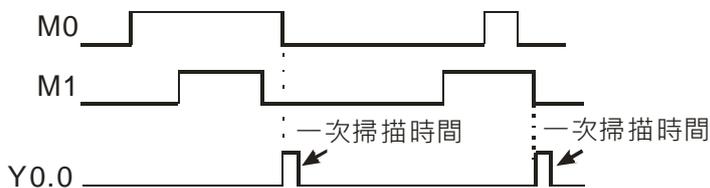
程式範例：

5



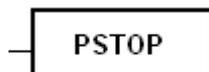
指令：		操作說明：
LD	M0	載入 M0 的 A 接點
AND	M1	串入 M1 之 A 接點
FB_PN	D0.0	下降緣觸發指令
OUT	Y0.0	驅動線圈 Y0.0

時序圖：



指令碼	運算元	功能
PSTOP	-	PLC 程式停止執行

符號：

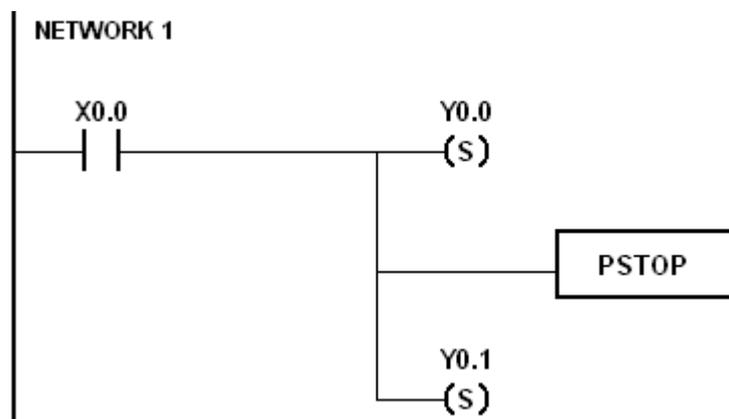


指令說明：

當條件接點啟動時，指令執行 END 動作，並且將 PLC 的狀態由 RUN 變成 STOP。

程式範例：

當 X0.0=ON 時，Y0.0 被設定為 ON，且 Y0.1 保持 OFF 的狀態，且 PLC 的狀態變成 STOP。



MEMO

5

6

第6章 指令表

目錄

6.1 比較操作指令.....	6-4
6.1.1 比較操作指令一覽表.....	6-4
6.1.2 比較操作指令說明.....	6-7
6.2 四則運算指令說明.....	6-36
6.2.1 四則運算指令一覽表.....	6-36
6.2.2 四則運算指令說明.....	6-37
6.3 資料轉換指令說明.....	6-73
6.3.1 資料轉換指令一覽表.....	6-73
6.3.2 資料轉換指令說明.....	6-74
6.4 資料轉移指令說明.....	6-110
6.4.1 資料轉移指令一覽表.....	6-110
6.4.2 資料轉移指令說明.....	6-111
6.5 程式跳躍指令.....	6-134
6.5.1 程式跳躍指令一覽表.....	6-134
6.5.2 程式跳躍指令說明.....	6-135
6.6 程式執行控制指令.....	6-143
6.6.1 程式執行控制指令一覽表.....	6-143
6.6.2 程式執行控制指令說明.....	6-144
6.7 I/O 更新指令.....	6-151
6.7.1 I/O 更新指令一覽表.....	6-151
6.7.2 I/O 更新指令說明.....	6-152
6.8 便利指令.....	6-153
6.8.1 便利指令一覽表.....	6-153
6.8.2 便利指令說明.....	6-154
6.9 邏輯操作指令.....	6-189
6.9.1 邏輯操作指令一覽表.....	6-189
6.9.2 邏輯操作指令說明.....	6-190
6.10 旋轉指令.....	6-212

6.10.1 旋轉指令一覽表	6-212
6.10.2 旋轉指令說明	6-213
6.11 基本指令	6-223
6.11.1 基本指令一覽表	6-223
6.11.2 基本指令說明	6-224
6.12 移位指令	6-231
6.12.1 移位指令一覽表	6-231
6.12.2 移位指令說明	6-232
6.13 資料處理指令	6-257
6.13.1 資料處理指令一覽表	6-257
6.13.2 資料處理指令說明	6-258
6.14 結構建立指令	6-302
6.14.1 結構建立指令一覽表	6-302
6.14.2 結構建立指令說明	6-303
6.15 模組的資料讀寫指令	6-310
6.15.1 模組的資料讀寫指令一覽表	6-310
6.15.2 模組的資料讀寫指令說明	6-311
6.16 浮點數指令	6-316
6.16.1 浮點數指令一覽表	6-316
6.16.2 浮點數指令說明	6-317
6.17 萬年曆指令	6-355
6.17.1 萬年曆指令一覽表	6-355
6.17.2 萬年曆指令說明	6-356
6.18 週邊設備指令	6-379
6.18.1 週邊設備指令一覽表	6-379
6.18.2 週邊設備指令說明	6-380
6.19 通訊指令	6-394
6.19.1 通訊指令一覽表	6-394
6.19.2 通訊指令說明	6-395
6.20 其他指令	6-429
6.20.1 其他指令一覽表	6-429
6.20.2 其他指令說明	6-430
6.21 字串處理指令	6-439

6.21.1 字串處理指令一覽表.....	6-439
6.21.2 字串處理指令說明.....	6-440
6.22 乙太網控制指令	6-499
6.22.1 乙太網控制指令一覽表	6-499
6.22.2 乙太網控制指令說明.....	6-500
6.23 記憶卡讀寫指令	6-532
6.23.1 記憶卡讀寫指令一覽表	6-532
6.23.2 記憶卡讀寫指令說明.....	6-533
6.24 任務控制指令	6-545
6.24.1 任務控制指令一覽表.....	6-545
6.24.2 任務控制指令說明.....	6-546
6.25 SFC 控制指令.....	6-548
6.25.1 SFC 控制指令一覽表	6-548
6.25.2 SFC 控制指令說明.....	6-549
6.26 備援控制指令	6-555
6.26.1 備援控制指令一覽表	6-555
6.26.2 備援控制指令說明	6-556

6.1 比較操作指令

6.1.1 比較操作指令一覽表

API	指令碼 (位元)			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	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
<u>0029</u>	–	FAND<=	DFAND<=	–	$S1 \leq S2$	5-7
<u>0030</u>	–	FOR=	DFOR=	–	$S1 = S2$	5-7
<u>0031</u>	–	FOR<>	DFOR<>	–	$S1 \neq S2$	5-7
<u>0032</u>	–	FOR>	DFOR>	–	$S1 > S2$	5-7
<u>0033</u>	–	FOR>=	DFOR>=	–	$S1 \geq S2$	5-7
<u>0034</u>	–	FOR<	DFOR<	–	$S1 < S2$	5-7
<u>0035</u>	–	FOR<=	DFOR<=	–	$S1 \leq S2$	5-7
<u>0036</u>	LD\$=	–	–	–	$S1 = S2$	5-17
<u>0037</u>	LD\$<>	–	–	–	$S1 \neq S2$	5-17
<u>0038</u>	LD\$>	–	–	–	$S1 > S2$	5-17
<u>0039</u>	LD\$>=	–	–	–	$S1 \geq S2$	5-17
<u>0040</u>	LD\$<	–	–	–	$S1 < S2$	5-17
<u>0041</u>	LD\$<=	–	–	–	$S1 \leq S2$	5-17
<u>0042</u>	AND\$=	–	–	–	$S1 = S2$	5-17
<u>0043</u>	AND\$<>	–	–	–	$S1 \neq S2$	5-17
<u>0044</u>	AND\$>	–	–	–	$S1 > S2$	5-17
<u>0045</u>	AND\$>=	–	–	–	$S1 \geq S2$	5-17
<u>0046</u>	AND\$<	–	–	–	$S1 < S2$	5-17
<u>0047</u>	AND\$<=	–	–	–	$S1 \leq S2$	5-17
<u>0048</u>	OR\$=	–	–	–	$S1 = S2$	5-17
<u>0049</u>	OR\$<>	–	–	–	$S1 \neq S2$	5-17
<u>0050</u>	OR\$>	–	–	–	$S1 > S2$	5-17
<u>0051</u>	OR\$>=	–	–	–	$S1 \geq S2$	5-17
<u>0052</u>	OR\$<	–	–	–	$S1 < S2$	5-17
<u>0053</u>	OR\$<=	–	–	–	$S1 \leq S2$	5-17
<u>0054</u>	CMP	DCMP	–	✓	比較設定輸出	7
<u>0055</u>	ZCP	DZCP	–	✓	區域比較	9
<u>0056</u>	FCMP	–	–	✓	浮點數比較	7-9
<u>0057</u>	FZCP	–	–	✓	浮點數區域比較	9-12
<u>0058</u>	MCMP	–	–	✓	矩陣比較	9
<u>0059</u>	CMPT=	–	–	✓	表格比較 · =	9
<u>0060</u>	CMPT<>	–	–	✓	表格比較 · <>	9
<u>0061</u>	CMPT>	–	–	✓	表格比較 · >	9
<u>0062</u>	CMPT>=	–	–	✓	表格比較 · >=	9

API	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
0063	CMPT<	-	-	✓	表格比較 · <	9
0064	CMPT<=	-	-	✓	表格比較 · <=	9
0065	CHKADR	-	-	-	接點型態指標暫存器位址檢查	7

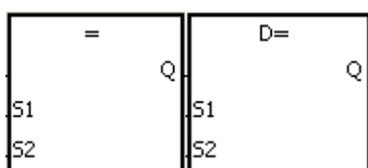
6.1.2 比較操作指令說明

API	指令碼			運算元							功能						
0000~ 0005	D	LD※		$S_1 \cdot S_2$							接點型態比較 LD※						

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
-	AH500	AH500

符號：

 S_1 : 資料來源 1

Word/Double Word

 S_2 : 資料來源 2

Word/Double Word

以 LD=跟 DLD=為例

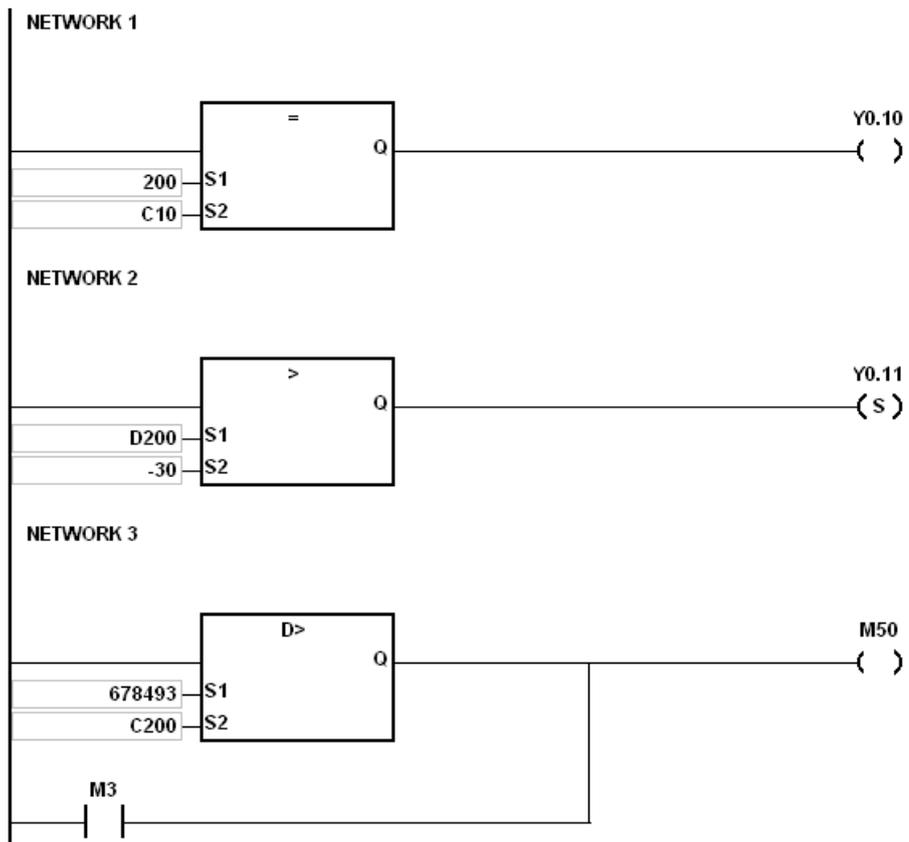
指令說明：

- S_1 與 S_2 的內容做比較的指令。以“LD=”做為例子，比較結果為“等於”時，該指令導通；“不等於”時，該指令不導通。
- 32 位元指令才可以使用 HC 裝置。

API No.	16 位元指令	32 位元指令	導通條件	非導通條件
0000	LD =	DLD =	$S_1 = S_2$	$S_1 \neq S_2$
0001	LD < >	DLD < >	$S_1 \neq S_2$	$S_1 = S_2$
0002	LD >	DLD >	$S_1 > S_2$	$S_1 \leq S_2$
0003	LD > =	DLD > =	$S_1 \geq S_2$	$S_1 < S_2$
0004	LD <	DLD <	$S_1 < S_2$	$S_1 \geq S_2$
0005	LD < =	DLD < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

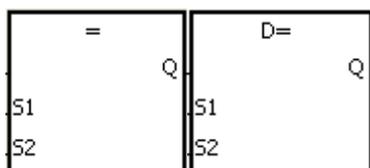
- C10 的內容等於 200 時，Y0.10=ON。
- 當 D200 的內容大於-30，Y0.11=ON 並保持住。
- (C201 · C200) 的內容小於 678,493 或者是 M3=ON 時，M50=ON。



6

API	指令碼			運算元								功能					
0006~ 0011	D	AND※		$S_1 \cdot S_2$								接點型態比較 AND※					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●	●	●	●		●	○	●	○	○		
S_2	●	●			●	●	●	●	●		●	○	●	○	○		
										脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)					
										-	AH500	AH500					

符號：

 S_1 : 資料來源 1

Word/Double Word

 S_2 : 資料來源 2

Word/Double Word

以 AND=跟 DAND=為例

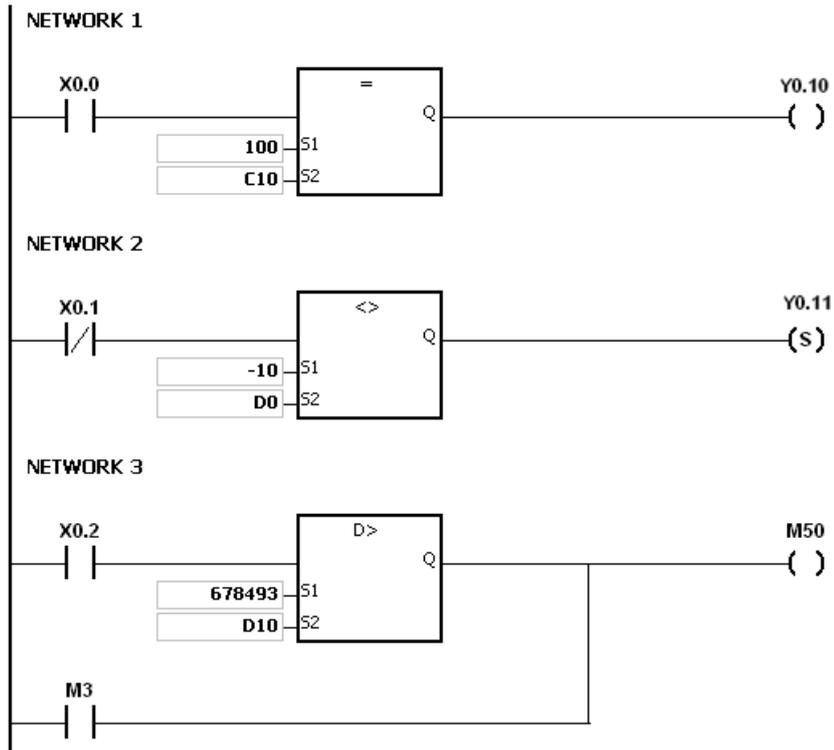
指令說明：

- S_1 與 S_2 的內容做比較的指令。以“AND=”做為例子，比較結果為“等於”時，該指令導通；“不等於”時，該指令不導通。
- 32 位元指令才可以使用 HC 裝置。

API No.	16 位元指令	32 位元指令	導通條件	非導通條件
0006	AND =	DAND =	$S_1 = S_2$	$S_1 \neq S_2$
0007	AND < >	DAND < >	$S_1 \neq S_2$	$S_1 = S_2$
0008	AND >	DAND >	$S_1 > S_2$	$S_1 \leq S_2$
0009	AND > =	DAND > =	$S_1 \geq S_2$	$S_1 < S_2$
0010	AND <	DAND <	$S_1 < S_2$	$S_1 \geq S_2$
0011	AND < =	DAND < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

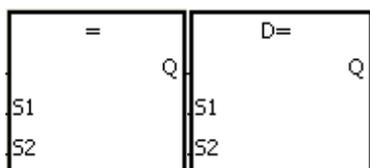
- 當 X0.0=ON 時且 C10 的現在值又等於 100 時，Y0.10=ON。
- 當 X0.1=OFF 而暫存器 D0 的內容又不等於-10 時，Y0.11=ON 並保持住。
- 當 X0.2=ON 而且 32 位元暫存器(D11·D0)的內容又小於 678,493 時或 M3=ON 時，M50=ON。



6

API	指令碼			運算元								功能						
0012~ 0017	D	OR※		$S_1 \cdot S_2$								接點型態比較 OR※						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S_1	●	●			●	●	●	●	●		●	○	●	○	○			
S_2	●	●			●	●	●	●	●		●	○	●	○	○			
										脈波執行型	16 位元指令 (5 steps)					32 位元指令 (5 steps)		
										-	AH500					AH500		

符號：

 S_1 : 資料來源 1

Word/Double Word

 S_2 : 資料來源 2

Word/Double Word

以 OR=跟 DOR=為例

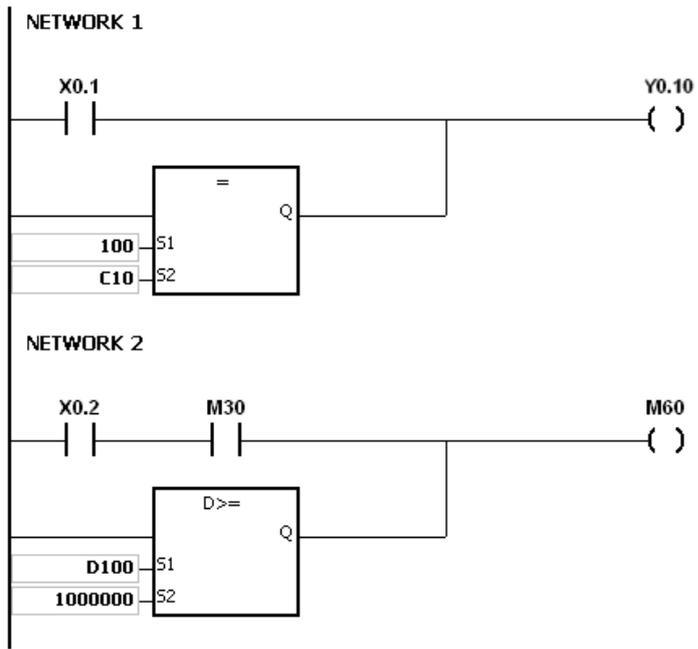
指令說明：

- S_1 與 S_2 的內容做比較的指令。以“OR=”做為例子，比較結果為“等於”時，該指令導通；“不等於”時，該指令不導通。
- 32 位元指令才可以使用 HC 裝置。

API No.	16 位元指令	32 位元指令	導通條件	非導通條件
0012	OR =	DOR =	$S_1 = S_2$	$S_1 \neq S_2$
0013	OR < >	DOR < >	$S_1 \neq S_2$	$S_1 = S_2$
0014	OR >	DOR >	$S_1 > S_2$	$S_1 \leq S_2$
0015	OR > =	DOR > =	$S_1 \geq S_2$	$S_1 < S_2$
0016	OR <	DOR <	$S_1 < S_2$	$S_1 \geq S_2$
0017	OR < =	DOR < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

- 當 X0.1=ON 時，或者是 C10 的現在值等於 100 時，Y0.10=ON。
- 當 X0.2 及 M30 都等於 ON 時，或者是 32 位元暫存器 (D101 · D100) 的內容大於或等於 1,000,000 時，M60=ON。



6

API	指令碼		運算元										功能			
0018~0023	D	FLD※	$S_1 \cdot S_2$										浮點數接點型態比較 LD※			

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

脈波執行型	32 位元指令 (5-7 steps)	64 位元指令 (5-7 steps)
-	AH500	AH500

符號：



S_1 : 資料來源 1 Double Word/Long Word
 S_2 : 資料來源 2 Double Word/Long Word

以 FLD= 跟 DFLD= 為例

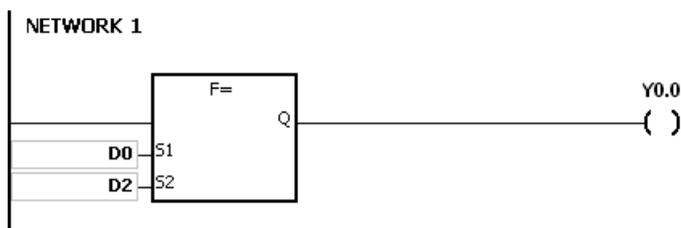
指令說明：

- S_1 與 S_2 的內容以浮點數型態做比較的指令。以“FLD=”做為例子，比較結果為“等於”時，該指令導通；“不等於”時，該指令不導通。

API No.	32 位元指令	64 位元指令	導通條件	非導通條件
0018	FLD =	DFLD =	$S_1 = S_2$	$S_1 \neq S_2$
0019	FLD < >	DFLD < >	$S_1 \neq S_2$	$S_1 = S_2$
0020	FLD >	DFLD >	$S_1 > S_2$	$S_1 \leq S_2$
0021	FLD > =	DFLD > =	$S_1 \geq S_2$	$S_1 < S_2$
0022	FLD <	DFLD <	$S_1 < S_2$	$S_1 \geq S_2$
0023	FLD < =	DFLD < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

以單精度浮點數指令為例，D0 的值等於 D2 的值時，Y0.0 ON。



補充說明：

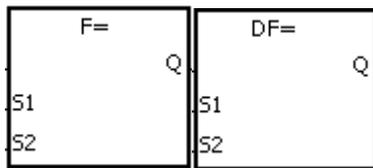
- 當 S_1 或 S_2 的內容值超出浮點數可以表示的範圍時，接點不導通，SM0=ON，錯誤碼 SR0=16#2013。

API	指令碼			運算元								功能					
0024~0029	D	FAND※		$S_1 \cdot S_2$								浮點數接點型態比較 AND※					

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

脈波執行型	32 位元指令 (5-7 steps)	64 位元指令 (5-7 steps)
-	AH500	AH500

符號：



S₁ : 資料來源 1 Double Word/Long Word
 S₂ : 資料來源 2 Double Word/Long Word

以 FAND=跟 DFAND=為例

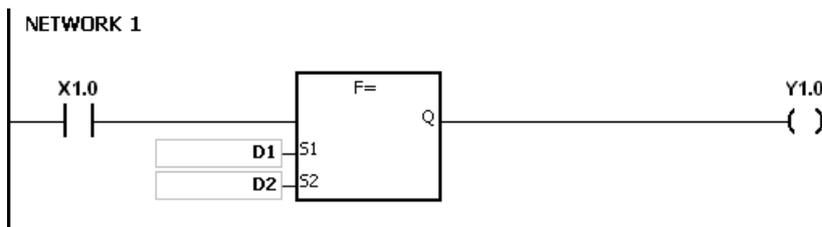
指令說明：

- S₁ 與 S₂ 的內容以浮點數型態做比較的指令。以“FAND=”做為例子，比較結果為“等於”時，該指令導通；“不等於”時，該指令不導通。

API No.	32 位元指令	64 位元指令	導通條件	非導通條件
0024	FAND =	DFAND =	$S_1 = S_2$	$S_1 \neq S_2$
0025	FAND < >	DFAND < >	$S_1 \neq S_2$	$S_1 = S_2$
0026	FAND >	DFAND >	$S_1 > S_2$	$S_1 \leq S_2$
0027	FAND > =	DFAND > =	$S_1 \geq S_2$	$S_1 < S_2$
0028	FAND <	DFAND <	$S_1 < S_2$	$S_1 \geq S_2$
0029	FAND < =	DFAND < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

以單精度浮點數指令為例，當 X1.0=ON 且 D1 的值等於 D2 的值時，Y1.0 ON。



補充說明：

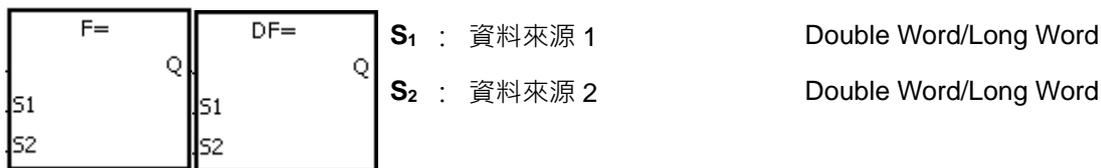
- 當 S₁ 或 S₂ 的內容值超出浮點數可以表示的範圍時，接點不導通，SM0=ON，錯誤碼 SR0=16#2013。

API	指令碼			運算元					功能				
0030~0035	D	FOR※		$S_1 \cdot S_2$					浮點數接點型態比較 OR※				

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

脈波執行型	32 位元指令 (5-7 steps)	64 位元指令 (5-7 steps)
-	AH500	AH500

符號：



以 FOR=跟 DFOR=為例

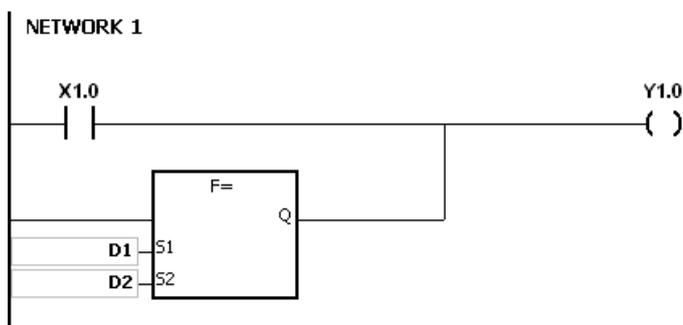
指令說明：

1. S₁ 與 S₂ 的內容以浮點數型態做比較的指令。以“FOR=”做為例子，比較結果為“等於”時，該指令導通；“不等於”時，該指令不導通。

API No.	32 位元指令	64 位元指令	導通條件	非導通條件
0030	FOR =	DFOR =	$S_1 = S_2$	$S_1 \neq S_2$
0031	FOR < >	DFOR < >	$S_1 \neq S_2$	$S_1 = S_2$
0032	FOR >	DFOR >	$S_1 > S_2$	$S_1 \leq S_2$
0033	FOR > =	DFOR > =	$S_1 \geq S_2$	$S_1 < S_2$
0034	FOR <	DFOR <	$S_1 < S_2$	$S_1 \geq S_2$
0035	FOR < =	DFOR < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

X1.0=ON 或者 D1 的值等於 D2 時，Y1.0=ON。



補充說明：

1. 當 S₁ 或 S₂ 的內容值超出浮點數可以表示的範圍時，接點不導通，SM0=ON，錯誤碼 SR0=16#2013。

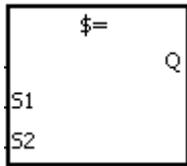
6

API	指令碼			運算元							功能						
0036~0041	LD\$※			S ₁ · S ₂							字串型態比較 LD\$※						

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

脈波執行型	16 位元指令 (5-17 steps)	32 位元指令
-	AH500	-

符號：



S₁ : 資料來源 1

String

S₂ : 資料來源 2

String

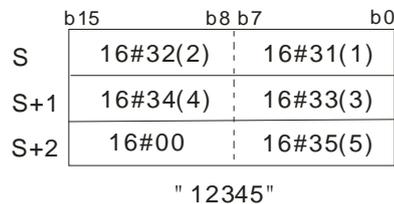
以 LD\$= 為例

指令說明：

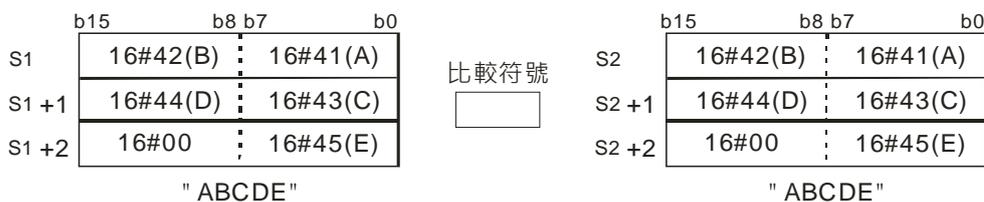
- S₁ 與 S₂ 的內容以字串型態做比較的指令。以“LD\$=”做為例子，比較結果為“等於”時，該接點導通；“不等於”時，該接點不導通。

API No.	指令	導通條件	非導通條件
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 (這裡的 n 表示第 n 個裝置，沒有限制) 中有包含 16#00 時，才能判斷為一個完整字串，例如：



- 字串完全相同時，如下範例 S₁ 與 S₂ 的字串都是“ABCDE”，各指令執行結果如下表所列：



比較符號	比較結果
\$ =	導通
\$ < >	不導通
\$ >	不導通
\$ > =	導通
\$ <	不導通
\$ < =	導通

4. 字串長度相同，但內容不相同時，以遇到的第一個不相同字的字元值 (ASCII 值) 來比較哪個字串較大；如下範例 **S₁** 的字串是“ABCDF”而 **S₂** 的字串是“ABCDE”，彼此遇到的第一個不同字為“F” (16#46) 跟“E” (16#45)，因為 16#46>16#45，所以認定 **S₁** 的字串>**S₂** 的字串，各指令執行結果如下表所列：

	b15	b8 b7	b0		b15	b8 b7	b0
S1	16#42(B)	16#41(A)		比較符號 □	S2	16#42(B)	16#41(A)
S1 +1	16#44(D)	16#43(C)			S2 +1	16#44(D)	16#43(C)
S1 +2	16#00	16#46(F)			S2 +2	16#00	16#45(E)
" ABCDF "					" ABCDE "		

如上範例

比較符號	比較結果
\$ =	不導通
\$ < >	導通
\$ >	導通
\$ > =	導通
\$ <	不導通
\$ < =	不導通

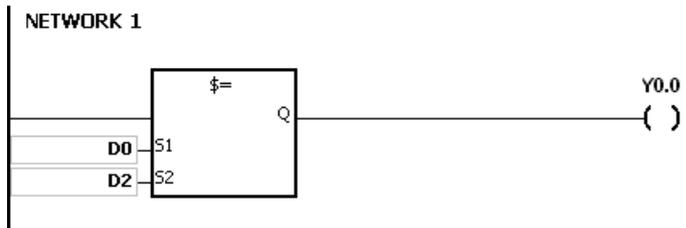
5. 字串長度不相等時，以字串長度較長的為大；如下範例 **S₁** 的字串是“1234567”而 **S₂** 的字串是“99999”，因為 **S₁** 的字串長度是 7 而 **S₂** 的字串是 5，所以認定 **S₁** 的字串>**S₂** 的字串，各指令執行結果如下表所列：

	b15	b8 b7	b0		b15	b8 b7	b0
S1	16#32(2)	16#31(1)		比較符號 □	S2	16#39(9)	16#39(9)
S1 +1	16#34(4)	16#33(3)			S2 +1	16#39(9)	16#39(9)
S1 +2	16#36(6)	16#35(5)			S2 +2	16#00	16#39(9)
S1 +3	16#00	16#37(7)			" 99999 "		
" 1234567 "							

比較符號	比較結果
\$ =	不導通
\$ < >	導通
\$ >	導通
\$ > =	導通
\$ <	不導通
\$ < =	不導通

程式範例：

D0~16#00 的字串等於 D2~16#00 的字串時，Y0.0 ON。



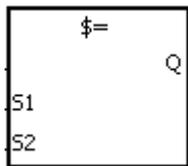
補充說明：

1. 字串的字尾無 16#00 當做結束符號，指令不執行，SM0=ON，錯誤碼 SR0=16#200E。

API	指令碼		運算元										功能				
0042~0047		AND\$※	$S_1 \cdot S_2$										字串型態比較 AND\$※				
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●		●	●		●	○	●			○	
S_2	●	●			●	●		●	●		●	○	●			○	

脈波執行型	16 位元指令 (5-17 steps)	32 位元指令
-	AH500	-

符號：



以 AND\$= 為例

S_1 : 資料來源 1
 S_2 : 資料來源 2

String
 String

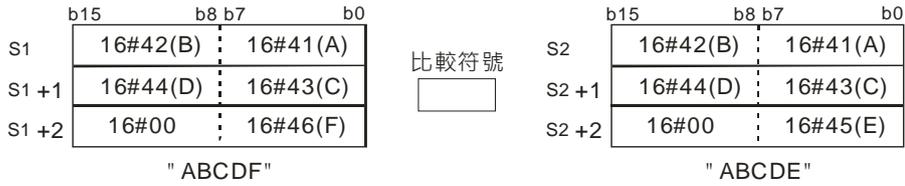
指令說明：

- S_1 與 S_2 的內容以字串型態做比較的指令。以“AND\$=”做為例子，比較結果為“等於”時，該接點導通，“不等於”時，該接點不導通。
- S 一直到 $S+n$ (這裡的 n 表示第 n 個裝置，沒有限制) 中有包含 16#00 時，才能當做一個完整的字串。
- 字串完全相同時，如下範例 S_1 與 S_2 的字串都是“ABCDE”，各指令執行結果如下表所列：

	b15	b8	b7	b0		b15	b8	b7	b0	
S_1	16#42(B)		16#41(A)		比較符號 <input type="checkbox"/>	16#42(B)		16#41(A)		
$S_1 + 1$	16#44(D)		16#43(C)			16#44(D)		16#43(C)		
$S_1 + 2$	16#00		16#45(E)			16#00		16#45(E)		
"ABCDE"						"ABCDE"				

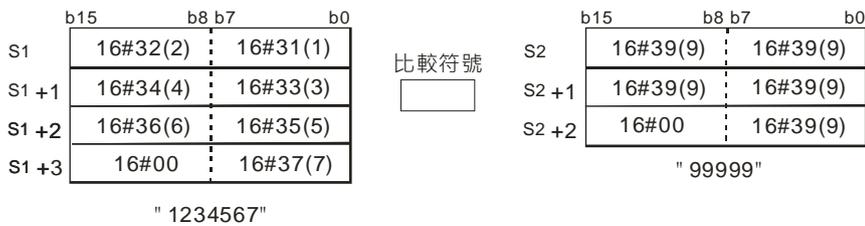
比較符號	比較結果
\$ =	導通
\$ < >	不導通
\$ >	不導通
\$ > =	導通
\$ <	不導通
\$ < =	導通

- 字串長度相同，但內容不相同時，以遇到的第一個不相同字的字元值 (ASCII 值) 來比較哪個字串較大；如下範例 S_1 的字串是“ABCDF”而 S_2 的字串是“ABCDE”，彼此遇到的第一個不同字為“F” (16#46) 跟“E” (16#45)，因為 16#46 > 16#45，所以認定 S_1 的字串 > S_2 的字串，各指令執行結果如下表所列：



比較符號	比較結果
\$ =	不導通
\$ < >	導通
\$ >	導通
\$ > =	導通
\$ <	不導通
\$ < =	不導通

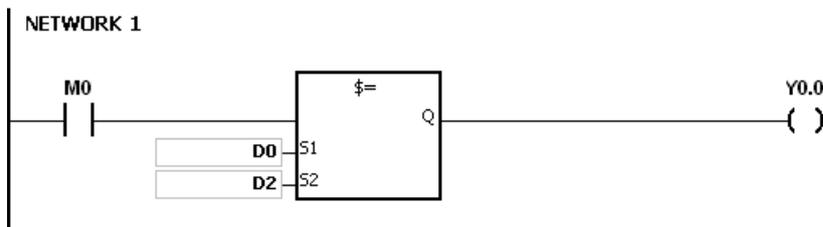
5. 字串長度不相等時，以字串長度較長的為大；如下範例 **S₁** 的字串是“1234567”而 **S₂** 的字串是“99999”，因為 **S₁** 的字串長度是 7 而 **S₂** 的字串是 5，所以認定 **S₁** 的字串 > **S₂** 的字串，各指令執行結果如下表所列：



比較符號	比較結果
\$ =	不導通
\$ < >	導通
\$ >	導通
\$ > =	導通
\$ <	不導通
\$ < =	不導通

程式範例：

當 M0=ON 且 D0~16#00 的字串等於 D2~16#00 的字串，則 Y0.0 ON。

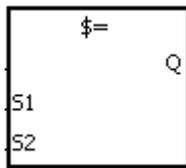


API	指令碼	運算元	功能
0048~0053	OR\$※	S ₁ · S ₂	字串型態比較 OR\$※

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

脈波執行型	16 位元指令 (5-17 steps)	32 位元指令
-	AH500	-

符號：



S₁ : 資料來源 1
S₂ : 資料來源 2

String
String

以 OR\$= 為例

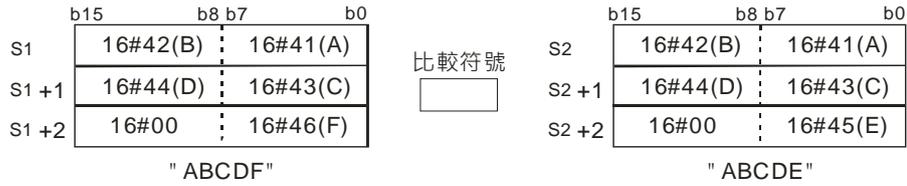
指令說明：

- S₁ 與 S₂ 的內容以字串型態做比較的指令。以“OR\$=”做為例子，比較結果為“等於”時，該接點導通；“不等於”時，該接點不導通。
- S 一直到 S+n (這裡的 n 表示第 n 個裝置，沒有限制) 中有包含 16#00 時，才能當做一個完整的字串。
- 字串完全相同時，如下範例 S₁ 與 S₂ 的字串都是“ABCDE”，各指令執行結果如下表所列：

	b15	b8	b7	b0	比較符號	b15	b8	b7	b0	
S ₁	16#42(B)		16#41(A)		□	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#45(E)			S ₂ +2	16#00		16#45(E)	
"ABCDE"						"ABCDE"				

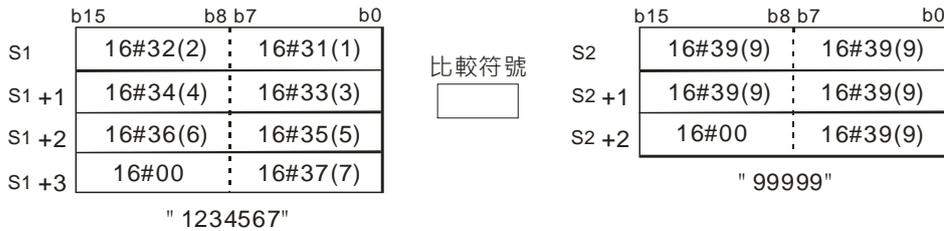
比較符號	比較結果
\$ =	導通
\$ < >	不導通
\$ >	不導通
\$ > =	導通
\$ <	不導通
\$ < =	導通

- 字串長度相同，但內容不相同時，以遇到的第一個不相同字的字元值 (ASCII 值) 來比較哪個字串較大；如下範例 S₁ 的字串是“ABCDF”而 S₂ 的字串是“ABCDE”，彼此遇到的第一個不同字為“F” (16#46) 跟“E” (16#45)，因為 16#46>16#45，所以認定 S₁ 的字串>S₂ 的字串，各指令執行結果如下表所列：



比較符號	比較結果
\$ =	不導通
\$ < >	導通
\$ >	導通
\$ > =	導通
\$ <	不導通
\$ < =	不導通

5. 字串長度不相等時，以字串長度較長的為大；如下範例 **S₁** 的字串是“1234567”而 **S₂** 的字串是“99999”，因為 **S₁** 的字串長度是 7 而 **S₂** 的字串是 5，所以認定 **S₁** 的字串 > **S₂** 的字串，各指令執行結果如下表所列：

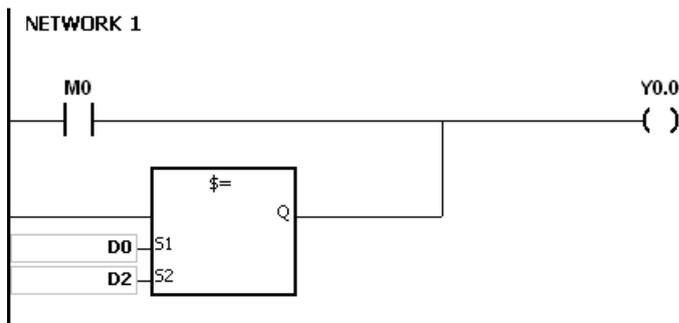


比較符號	比較結果
\$ =	不導通
\$ < >	導通
\$ >	導通
\$ > =	導通
\$ <	不導通
\$ < =	不導通

6

程式範例：

當 M0=ON 或者是 D0~16#00 的字串等於 D2~16#00 的字串時，Y0.0 ON。



API	指令碼			運算元							功能						
0054	D	CMP	P	$S_1 \cdot S_2 \cdot D$							比較設定輸出						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

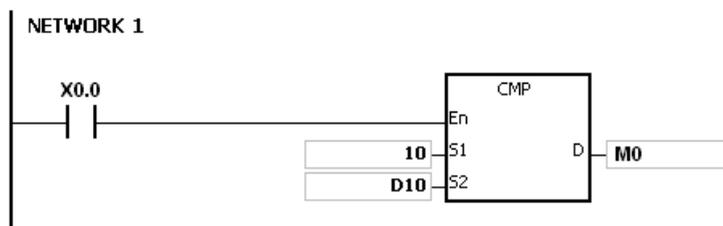


指令說明：

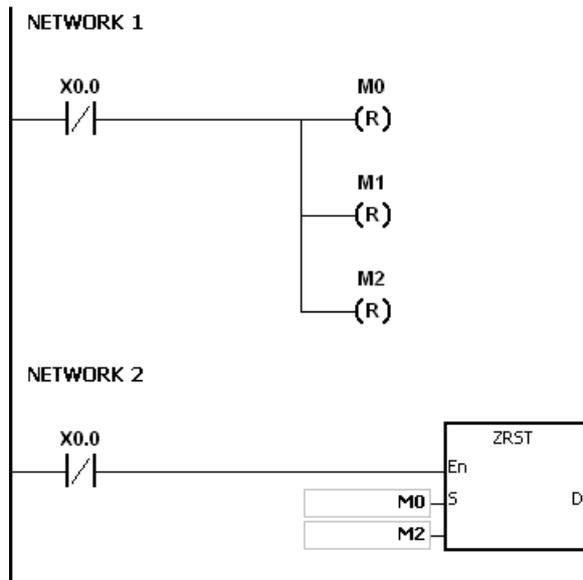
- 將運算元 S₁ 和 S₂ 的內容以有號數十進制作大小比較，其比較結果在 D 作表示。
- 運算元 D 佔用 3 個連續裝置。D · D+1 · D+2 用於儲存比較結果，如果 S₁>S₂ · D=ON；如果 S₁=S₂ · D+1=ON；如果 S₁<S₂ · D+2=ON。
- DCMP、DCMPP 指令才可使用 HC 裝置。

程式範例：

- 如果運算元 D 設為 M0，那麼比較結果會顯示在 M0 · M1 · M2 中，如下所示。
- 當 X0.0=ON · CMP 指令執行，M0 · M1 · M2 其中的某個裝置會 ON。當 X0.0=OFF · CMP 指令停止執行並且 M0 · M1 · M2 保持它們的現在值。



3. 若要清除其比較結果請使用 RST 或 ZRST 指令。



補充說明：

1. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of BOOL。
2. **D+2** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

6

API	指令碼			運算元							功能						
0055	D	ZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$							區域比較						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S₁	●	●			●	●	●	●	●		●	○	●	○	○		
S₂	●	●			●	●	●	●	●		●	○	●	○	○		
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●	●	●				●	●	●			●				

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
AH500	AH500	AH500

符號：

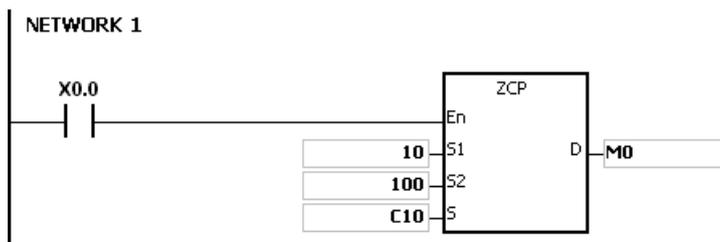


指令說明：

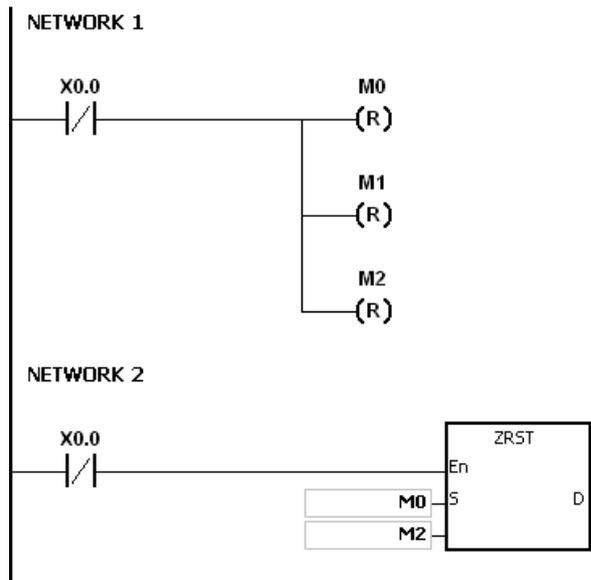
- 比較值 **S** 與下限 **S₁** 及上限 **S₂** 以有號數十進制數值做比較，其比較結果在 **D** 作表示。
- 運算元 **S₁** 必須比 **S₂** 小。當 **S₁>S₂**，指令執行時把 **S₁** 作為上/下限值進行比較。
- 運算元 **D** 佔用 3 個連續的裝置，**D**·**D+1**·**D+2** 儲存比較結果。如果 **S₁>S**·**D=ON**；如果 **S₁≤S≤S₂**，**D+1=ON**；如果 **S₂<S**·**D+2=ON**。
- DZCP、DZCPP 指令才可使用 HC 裝置。

程式範例：

- 比較結果指定裝置為 **M0**，則自動佔有 **M0**、**M1** 及 **M2**。
- 當 **X0.0=ON** 時，ZCP 指令執行，**M0**、**M1** 及 **M2** 其中之一會 ON，當 **X0.0=OFF** 時，ZCP 指令不執行，**M0**、**M1** 及 **M2** 狀態保持在 **X0.0=OFF** 之前的狀態。



3. 若要清除其結果請使用 RST 或 ZRST 指令。



補充說明：

1. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of BOOL。
2. **D+2** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

6

API	指令碼		運算元				功能			
0056	FCMP	P	$S_1 \cdot S_2 \cdot D$				浮點數比較			

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

脈波執行型	32 位元指令 (7-9 steps)	64 位元指令
AH500	AH500	-

符號：

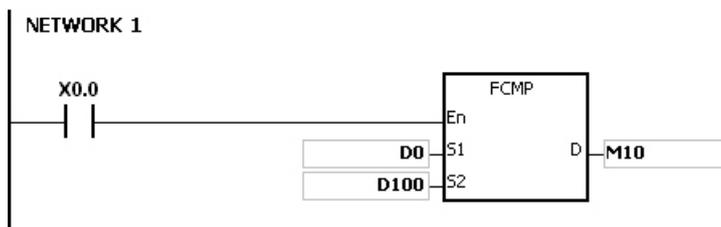


指令說明：

1. 浮點數比較值 S_1 與浮點數比較值 S_2 進行比較，比較的結果 (> , = , <) 在 D 作表示。
2. 運算元 D 佔用 3 個連續裝置。D、D+1、D+2 用於儲存比較結果，如果 $S_1 > S_2$ ，D=ON；如果 $S_1 = S_2$ ，D+1=ON；如果 $S_1 < S_2$ ，D+2=ON。

程式範例：

1. 若比較結果指定裝置為 M10 則自動佔有 M10~M12。
2. 當 X0.0=ON 時，FCMP 指令執行，M10~M12 其中之一會 ON。當 X0.0=OFF，FCMP 指令不執行，M10~M12 狀態保持在 X0.0=OFF 之前的狀態。
3. 若需得到 \geq ， \leq ， \neq 的結果時，可將 M10~M12 串並聯即可取得。
4. 若要清除其結果請使用 RST 或 ZRST 指令。



補充說明：

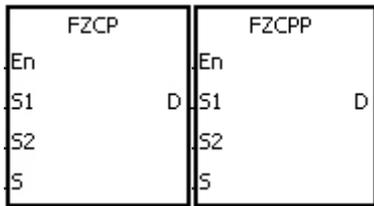
1. 當 S_1 或 S_2 的內容值超出浮點數可以表示的範圍時，接點不導通，SM0=ON，錯誤碼 SR0=16#2013。
2. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of BOOL。
3. D+2 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能					
0057	FZCP	P		$S_1 \cdot S_2 \cdot S \cdot D$								浮點數區域比較					

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

脈波執行型	32 位元指令 (9-12 steps)	64 位元指令
AH500	AH500	-

符號：



- S_1 : 浮點數區域比較 Double Word
- S_2 : 區域比較之浮點數上限值 Double Word
- S : 浮點數比較值 Double Word
- D : 比較結果 Bit

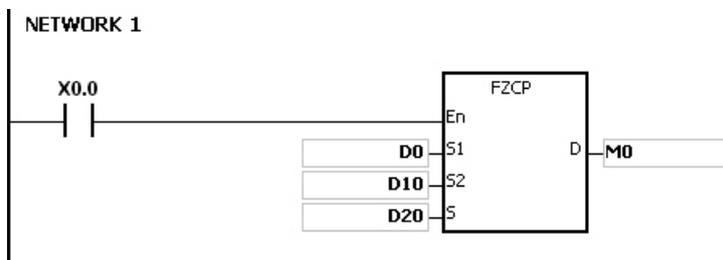
指令說明：

- 浮點數比較值 S 與浮點數下限值 S_1 及浮點數上限值 S_2 做比較，其比較結果在 D 做表示。
- 運算元 S_1 必須比 S_2 小。當 $S_1 > S_2$ ，指令執行時把 S_1 作為上/下限值進行比較。
- 運算元 D 佔用 3 個連續的裝置： $D \cdot D+1 \cdot D+2$ 儲存比較結果。如果 $S_1 > S$ ， $D=ON$ ；如果 $S_1 \leq S \leq S_2$ ， $D+1=ON$ ；如果 $S_2 < S$ ， $D+2=ON$ 。

6

程式範例：

- 若比較結果指定裝置為 $M0$ ，則自動佔有 $M0 \sim M2$ 。
- 當 $X0.0=ON$ 時，FZCP 指令執行， $M0 \sim M2$ 其中之一會 ON。當 $X0.0=OFF$ ，FZCP 指令不執行， $M0 \sim M2$ 狀態保持在 $X0.0=OFF$ 之前的狀態。
- 若要清除其結果請使用 RST 或 ZRST 指令。



補充說明：

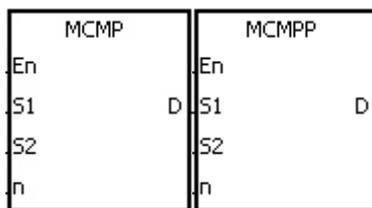
- 當 S_1 或 S_2 或 S 的內容值超出浮點數可以表示的範圍時，接點不導通， $SM0=ON$ ，錯誤碼 $SR0=16\#2013$ 。
- D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of BOOL。
- $D+2$ 超過裝置範圍時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。

API	指令碼			運算元							功能				
0058		MCMP	P	$S_1 \cdot S_2 \cdot n \cdot D$							矩陣比較				

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



- S_1 : 矩陣來源裝置 1 Word
- S_2 : 矩陣來源裝置 2 Word
- n : 陣列長度 Word
- D : 指標 Pr 用以存放目標的位址值 Word

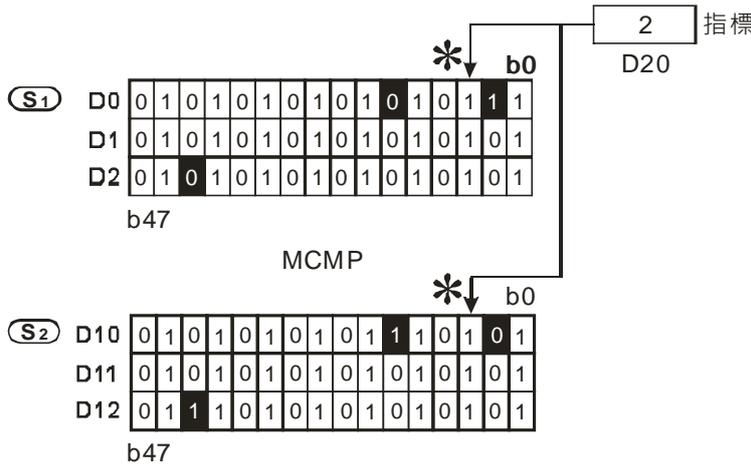
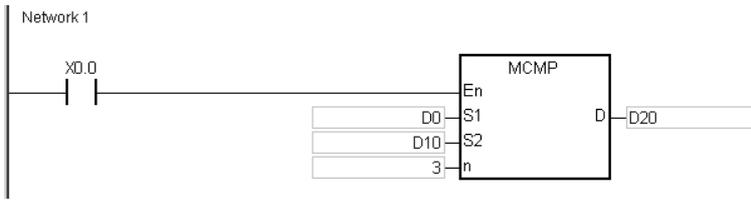
指令說明：

- 每次比較依兩個矩陣來源指標 Pr 位址。將 S_1 或 S_2 兩個矩陣中的每一個 bit 從位址 D+1 開始做比較，找出值不同的位址，再將此位址值存到 D 中，完成此次比較。
- n 運算元的範圍為 1~256。
- 由矩陣比較旗標 SM607 決定比較相同值 (SM607=1) 或不同值 (SM607=0)，當比較到達時立即停止比較動作，矩陣位元尋找旗標 SM610=1。當比較到最後一個 bit 時，矩陣搜尋結束旗標 SM608=ON 比較到達的編號存於 D 中，下一次掃描週期時，再由第 0 個 bit 開始比較，同時矩陣搜尋起始旗標 SM609=1。當 D 的值超過範圍時指標錯誤旗標 SM611=1。
- 在矩陣指令運作中，通常需要有一個 16 位元暫存器來指定矩陣中 16n 個單點的某個單點當作運算物件。此暫存器稱為矩陣的指標 Pr (Pointer)，由使用者於指令中指定，其有效範圍為 0~16n-1，分別對應至矩陣中的位元 b0~b16n-1。在運作中應避免更動到 Pr 值，以免影響其正確的比較找尋，若 Pr 值超出此範圍則矩陣指標錯誤旗標 SM611 設為 1，且本指令不執行。
- 若矩陣搜尋結束旗標 SM608 與矩陣位元尋找旗標 SM610 同時發生則會同時=1。

程式範例：

- 當 X0.0 由 OFF→ON，矩陣搜尋起始旗標 SM609=0，故由指標當時值加 1 的 bit 位址(標注 * 處) 開始往下比較找尋位元狀態不同 (SM607=0 為找不同) 者。
- 設指標當時值 D20=2，當 X0.0 由 OFF→ON 時動作 4 次，可得到如 (① · ② · ③ · ④) 四個執行結果。
 - D20=5，矩陣位元尋找旗標 SM610=1，矩陣搜尋結束旗標 SM608=0。
 - D20=45，矩陣位元尋找旗標 SM610=1，矩陣搜尋結束旗標 SM608=0。
 - D20=47，矩陣位元尋找旗標 SM610=0，矩陣搜尋結束旗標 SM608=1。

- D20=1，矩陣位元尋找旗標 SM610=1，矩陣搜尋結束旗標 SM608=0。



補充說明：

1. 演算錯誤碼說明：

當 S_1+n-1 或 S_2+n-1 的裝置超過範圍時，該指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

若 n 運算元的值不在 1~256 之間，該指令不執行，且 SM0=ON，錯誤碼 SR0=16#200B。

2. 旗標信號說明：

- SM607： 矩陣比較旗標，比較相同值 SM607=1，比較不同值 SM607=0
- SM608： 矩陣搜尋結束旗標，當比較到最後一個 bit 時，SM608=1
- SM609： 矩陣搜尋起始旗標，由第一個 bit 開始比較，SM609=1
- SM610： 矩陣位元尋找旗標，比較到達時立即停止比較動作 SM610=1
- SM611： 矩陣指標錯誤旗標，指標 Pr 值超出此範圍則 SM611=1

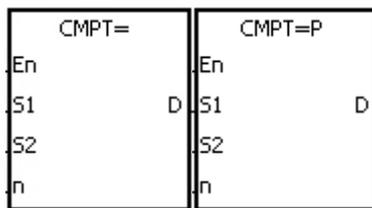


API	指令碼				運算元								功能			
0059~0064		CMPT※	P		$S_1 \cdot S_2 \cdot n \cdot D$								表格比較指令			

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

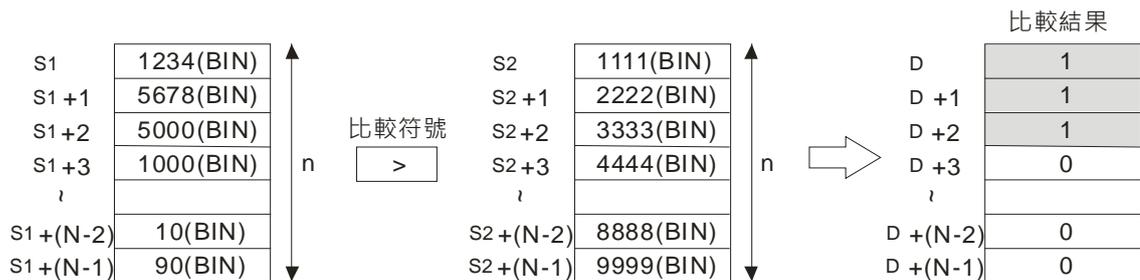
符號：



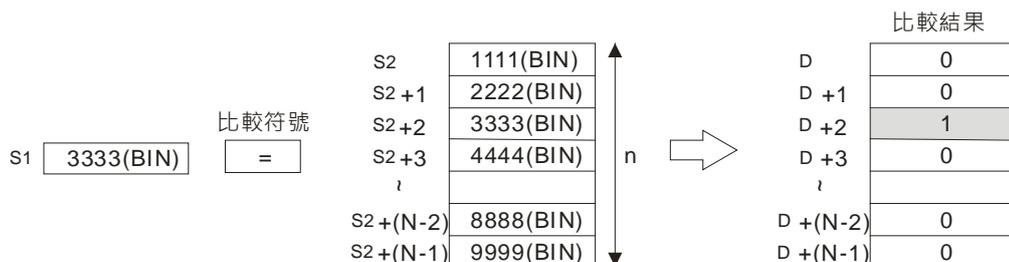
- S_1 : 來源起始裝置 1 Word
- S_2 : 來源起始裝置 2 Word
- n : 比較資料長度 Word
- D : 目標裝置 Bit

指令說明：

1. S_1 與 S_2 的內容以有號數十進制做 n 筆比較，將結果存放於 D 。
2. n 運算元的範圍為 1~256。
3. D 運算元寫入值都以 1 位元寫入，未被對應之 bit 值不變。
4. 表格比較指令 CMPT#全部輸出旗標，若比較結果為全都輸出（全都 ON），則 SM620=ON，反之 SM620=OFF。
5. 如果 S_1 的來源指定為裝置，則比較方式如下：



6. 如果 S_1 的來源直接指定為 -32768~-32767，則比較方式如下：

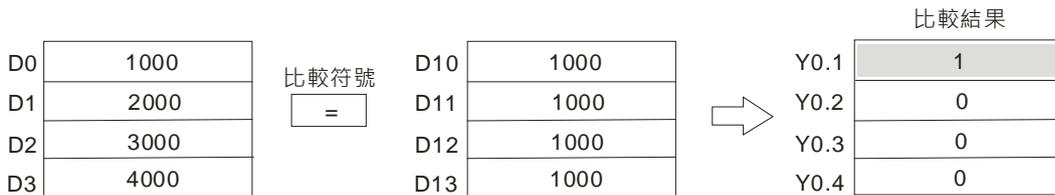
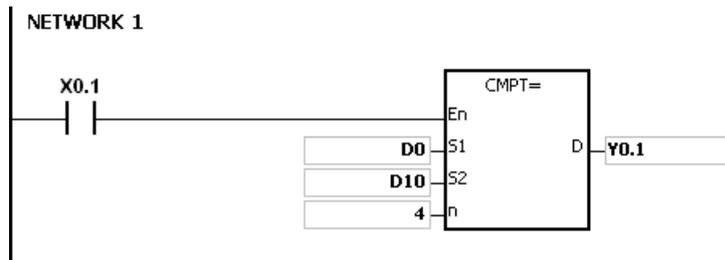


7. 各指令比較結果如下：

API No.	16 位元指令	比較結果	
		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 < =	$S_1 \leq S_2$	$S_1 > S_2$

程式範例：

D0~D3 與 D10~D13 資料比對，若比較結果相同，對應的 Y0.1~Y0.4=ON。



6

補充說明：

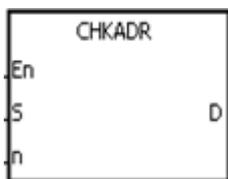
1. 若 n 不在 1~256 之間，指令不執行，SM0=ON，SR0 Error Code=16#200B。
2. $S_1 \sim S_{1+n}$ 與 $S_2 \sim S_{2+n}$ 與 D 運算元的裝置如果不足，則指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元							功能				
0065		CHKADR		S · n · D							接點型態指標暫存器位址檢查				

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：



- S** : 指標暫存器
POINTER/T_POINTER/
C_POINTER/HC_POINTER
- n** : 裝置個數
Word
- D** : 檢查結果
Bit

指令說明：

1. 檢查 **S** 指標暫存器的內容值~ (**S** 的內容值+n-1) 是否超過 **S** 指標暫存器所指定的裝置類別範圍。檢查結果沒有超出裝置範圍時，**D** 裝置為 ON，否則 **D** 裝置為 OFF。
2. **S** 支援 PR、TR、CR、HCR 指標暫存器。
3. **n** 的範圍 1~1024。
4. 本指令只可以在功能塊中使用。

程式範例：

1. 使用 ISPSOft 建立一個 Program，及一個 Function Block。



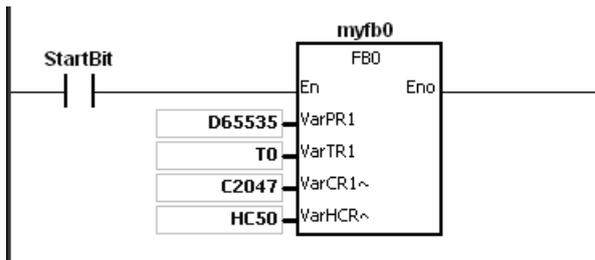
並且在 Program 中宣告兩個變數如下：

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 分別指定為 POINTER、T_POINTER、C_POINTER、HC_POINTER。

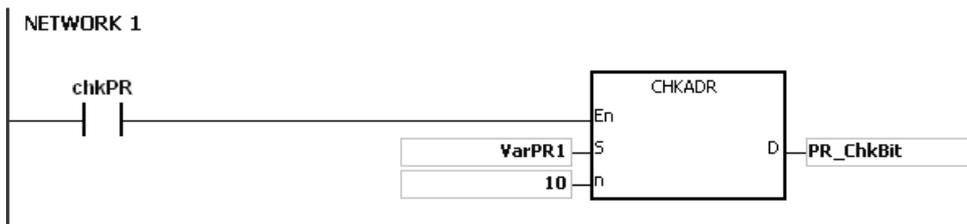
Local Symbols				
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR_IN_OUT	VarPR1	N/A [Auto]	POINTER	N/A
VAR_IN_OUT	VarTR1	N/A [Auto]	T_POINTER	N/A
VAR_IN_OUT	VarCR1	N/A [Auto]	C_POINTER	N/A
VAR_IN_OUT	VarHCR1	N/A [Auto]	HC_POINTER	N/A
VAR	PR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	TR_ChkBit	N/A [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 中呼叫已建立好的 Function Block FB0，並且將 D65535、T0、C2047、HC50 指定給 FB0 中的 VarPR1、VarTR1、VarCR1、VarHCR1。

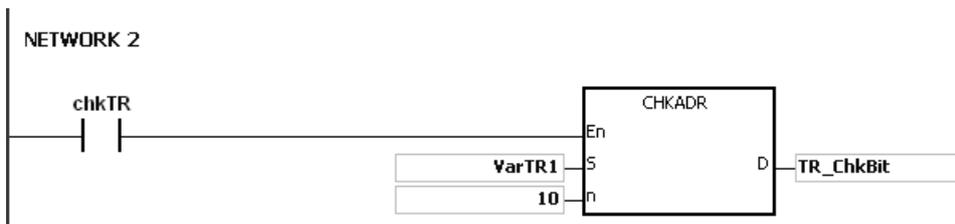


6

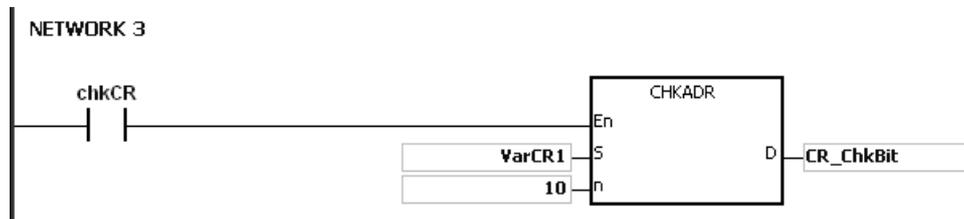
- 在 FB0 中使用 CHKADR 來檢查 VarPR1、VarTR1、VarCR1、VarHCR1 是否超出範圍。
- 當 chkPR=ON，VarPR1 所代表的實際裝置為 D65535，而 D 裝置的合法範圍為 D0~D65535，所以 $D65535+10-1=D65544$ 已超出範圍，PR_ChkBit=OFF。



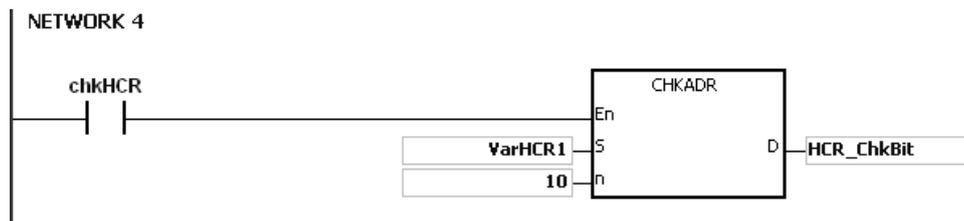
- 當 chkTR=ON，VarTR1 所代表的實際裝置為 T0，而 T 裝置的合法範圍為 T0~T2047，所以 $T0+10-1=T9$ 沒有超出範圍，TR_ChkBit=ON。



7. 當 $chkCR=ON$ ， $VarCR1$ 所代表的實際裝置為 C2047，而 C 裝置的合法範圍為 C0~C2047，所以 $C2047+10-1=C2056$ 已超出範圍， $CR_ChkBit=OFF$ 。



8. 當 $chkHCR=ON$ ， $VarHCR1$ 所代表的實際裝置為 HC50，而 HC 裝置的合法範圍為 HC0~HC63，所以 $HC50+10-1=HC59$ 沒有超出範圍， $HCR_ChkBit=ON$ 。



補充說明：

1. **S** 的內容值(實際裝置位址)如果一開始就不合法，超出裝置範圍時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. 若 **n** 不在 1~1024 之間，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。

6.2 四則運算指令

6.2.1 四則運算指令一覽表

API	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
0100	+	D+	-	✓	BIN 加法	7
0101	-	D-	-	✓	BIN 減法	7
0102	*	D*	-	✓	BIN 乘法	7
0103	/	D/	-	✓	BIN 除法	7
0104	-	F+	DF+	✓	浮點數加法	7-9
0105	-	F-	DF-	✓	浮點數減法	7-9
0106	-	F*	DF*	✓	浮點數乘法	7-9
0107	-	F/	DF/	✓	浮點數除法	7-9
0108	B+	DB+	-	✓	BCD 加法	7
0109	B-	DB-	-	✓	BCD 減法	7
0110	B*	DB*	-	✓	BCD 乘法	7
0111	B/	DB/	-	✓	BCD 除法	7
0112	BK+	-	-	✓	連續區塊 BIN 加法	9
0113	BK-	-	-	✓	連續區塊 BIN 減法	9
0114	\$+	-	-	✓	字串鏈結	7-19
0115	INC	DINC	-	✓	BIN 加一	3
0116	DEC	DDEC	-	✓	BIN 減一	3
0117	MUL16	MUL32	-	✓	16 位元專用 BIN 乘法/ 32 位元專用 BIN 乘法	7
0118	DIV16	DIV32	-	✓	16 位元專用 BIN 除法/ 32 位元專用 BIN 除法	7

6

6.2.2 四則運算指令說明

API	指令碼			運算元							功能						
0100	D	+	P	$S_1 \cdot S_2 \cdot D$							BIN 加法						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

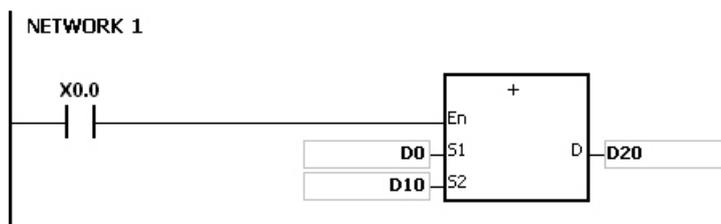


指令說明：

- S₁ 及 S₂ 以 BIN 方式相加的結果存於 D。
- 32 位元指令才可以使用 HC 裝置。
- 旗標：SM600 零旗號 (Zero flag)，SM601 借位旗號 (Borrow flag)，SM602 進位旗號 (Carry flag)。
- 演算結果為 0 時，零旗號 (Zero flag) SM600 為 ON，否則為 OFF。
- 16 位元 BIN 加法：
演算結果大於 16 位元 BIN 可表示範圍時，進位旗號 (Carry flag) SM602 為 ON，否則為 OFF。
- 32 位元 BIN 加法：
演算結果大於 32 位元 BIN 可表示範圍時，進位旗號 (Carry flag) SM602 為 ON，否則為 OFF。

程式範例一：

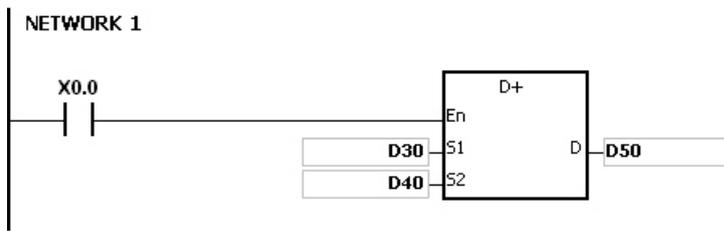
16 位元 BIN 加法：當 X0.0=ON 時，被加數 D0 內容加上加數 D10 之內容將結果存在 D20 之內容當中。



- 當 $D0=100 \cdot D10=10 \cdot D0+D10=110 \cdot D20=110$ 。
- 當 $D0=16\#7FFF \cdot D10=16\#1 \cdot D0+D10=16\#8000 \cdot D20=16\#8000$ 。
- 當 $D0=16\#FFFF \cdot D10=16\#1 \cdot D0+D10=16\#10000$ ，此時演算結果超出 16 位元 BIN 可表示範圍，則進位旗號 $SM602=ON$ ， $D20=16\#0$ ，因為演算結果為 $16\#0$ ，所以零旗號 $SM600=ON$ 。

程式範例二：

32 位元 BIN 加法：當 $X0.0=ON$ 時，被加數 ($D31 \cdot D30$) 內容加上加數 ($D41 \cdot D40$) 之內容將結果存在 ($D51 \cdot D50$) 之中。(其中 $D30 \cdot D40 \cdot D50$ 為低 16 位元資料， $D31 \cdot D41 \cdot D51$ 為高 16 位元資料)。



- 當 $(D31 \cdot D30) = 11111111 \cdot (D41 \cdot D40) = 44444444 \cdot (D31 \cdot D30) + (D41 \cdot D40) = 55555555$ ，此時演算結果， $(D51 \cdot D50) = 55555555$ 。
- 當 $(D31 \cdot D30) = 16\#80000000 \cdot (D41 \cdot D40) = 16\#FFFFFF \cdot (D31 \cdot D30) + (D41 \cdot D40) = 16\#17FFFFFF$ ，此時演算結果超出 32 位元 BIN 可表示範圍，則進位旗號 $SM602=ON$ ， $(D51 \cdot D50) = 16\#7FFFFFFF$ 。

旗標操作：

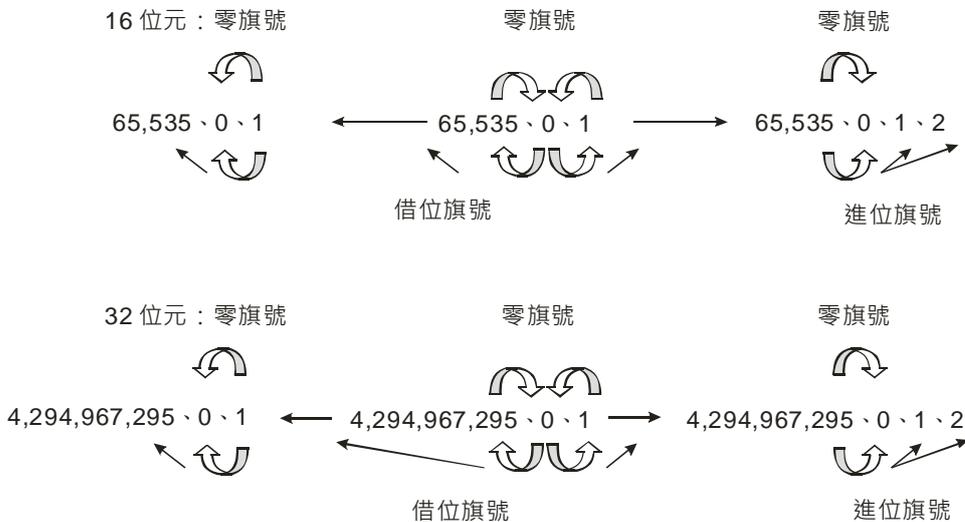
6

16 位元指令：

1. 如果操作結果是“0”，零旗標 $SM600$ 被設定成 ON。
2. 如果操作結果超出 65,535，進位旗標 $SM602$ 被設定成 ON。

32 位元指令：

1. 如果操作結果是“0”，零旗標 $SM600$ 被設定成 ON。
2. 如果操作結果超出 4,294,967,295，進位旗標 $SM602$ 被設定成 ON。

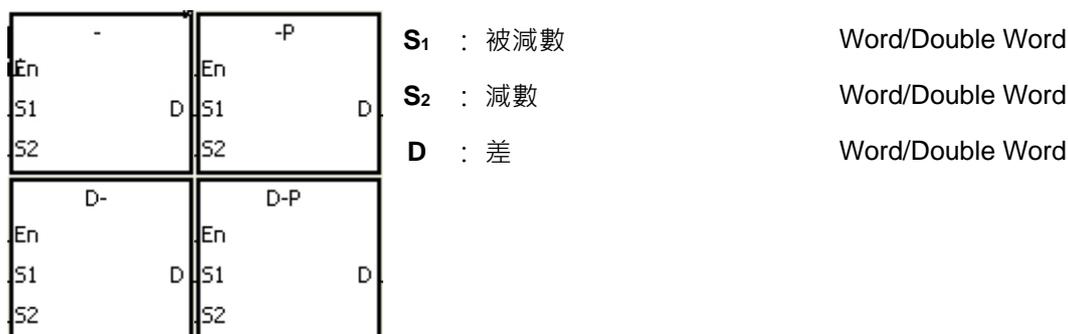


API	指令碼			運算元						功能							
0101	D	-	P	S₁ · S₂ · D						BIN 減法							

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

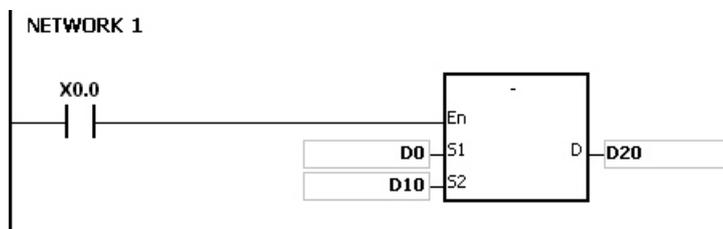


指令說明：

1. S₁ 及 S₂ 以 BIN 方式相減的結果存於 D。
2. 32 位元指令才可以使用 HC 裝置。
3. 旗標：SM600 零旗號 (Zero flag) · SM601 借位旗號 (Borrow flag) · SM602 進位旗號 (Carry flag)。
4. 演算結果為 0 時，零旗號 (Zero flag) SM600 為 ON，否則為 OFF。
5. 演算發生借位時，借位旗號 (Borrow flag) SM601 為 ON，否則為 OFF。

程式範例一：

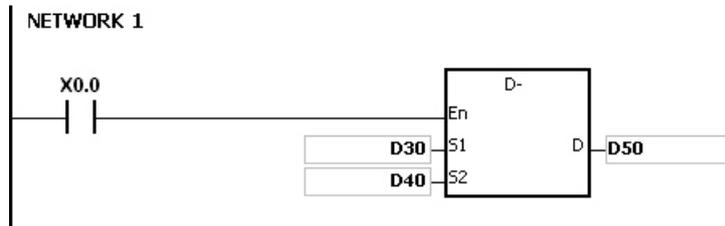
16 位元 BIN 減法：當 X0.0=ON 時，將 D0 內容減掉 D10 內容將差存在 D20 之內容中，如下說明：



- 當 D0=100 · D10=10 · D0-D10=90 · D20=90。
- 當 D0=16#8000 · D10=16#1 · D0-D10=16#7FFF · D20=16#7FFF。
- 當 D0=16#1 · D10=16#2 · D0-D10=16#FFFF，演算發生借位，借位旗號 SM601=ON，D20=16#FFFF。
- 當 D0=16#0 · D10=16#FFFF · D0-D10=16#F0001，演算發生借位，借位旗號 SM601=ON，D20=16#1。

程式範例二：

32 位元 BIN 減法：當 X0.0=ON 時，(D31、D30) 內容減掉 (D41、D40) 之內容將差存在 (D51、D50) 之中。(其中 D30、D40、D50 為低 16 位元資料，D31、D41、D51 為高 16 位元資料)，如下說明：



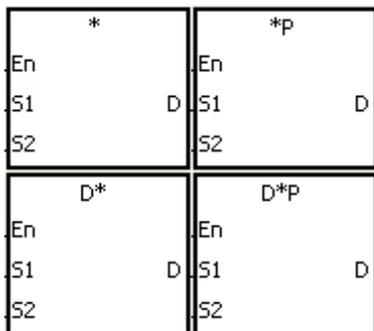
- 當 (D31、D30) = 55555555，(D41、D40) = 11111111，(D31、D30) - (D41、D40) = 44444444，(D51、D50) = 44444444。
- 當 (D31、D30) = 16#80000000，(D41、D40) = 16#FFFFFF，(D31、D30) - (D41、D40) = 16#F8000001，演算發生借位，借位旗號 SM601=ON，(D51、D50) = 16#80000001。

API	指令碼			運算元						功能					
0102	D	*	P	S₁ · S₂ · D						BIN 乘法					

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S₁	●	●			●	●	●	●	●		●	○	●	○	○		
S₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

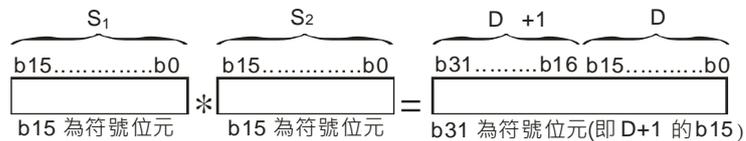
符號：



- S₁** : 被乘數 Word/Double Word
- S₂** : 乘數 Word/Double Word
- D** : 積 Double Word/Long Word

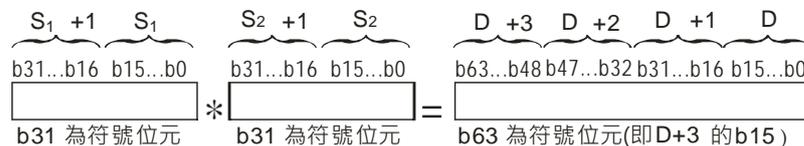
指令說明：

- S₁** 及 **S₂** 以有號數二進制方式相乘後的積存於 **D**。
- D*** 指令才可以使用 **HC** 裝置。
- 16 位元 BIN 乘法運算：



積為 32 位元資料，儲存在 (**D+1** · **D**) 組成的 32 位元暫存器中，且符號位元 **b31=0** 為正數，符號位元 **b31=1** 為負數。

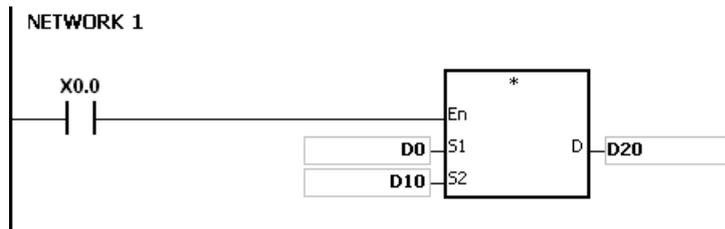
- 32 位元 BIN 乘法運算：



積為 64 位元資料，儲存在 (**D+3** · **D+2** · **D+1** · **D**) 組成的 64 位元暫存器中，且符號位元 **b63=0** 為正數，符號位元 **b63=1** 為負數。

程式範例：

16 位元 D0 乘上 16 位元 D10 得到一個 32 位元之積，結果存在 (D21 · D20)。高 16 位元存於 D21，低 16 位元存於 D20，結果之正負由最高位元 (b31) 之 OFF/ON 來指示。OFF 表示正的 (0)，同時 ON 表示負的 (1)。



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

16 位元 × 16 位元 = 32 位元

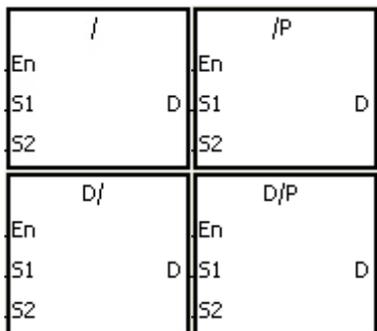
6

API	指令碼			運算元						功能					
0103	D	/	P	S₁ · S₂ · D						BIN 除法					

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S₁	●	●			●	●	●	●	●		●	○	●	○	○		
S₂	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

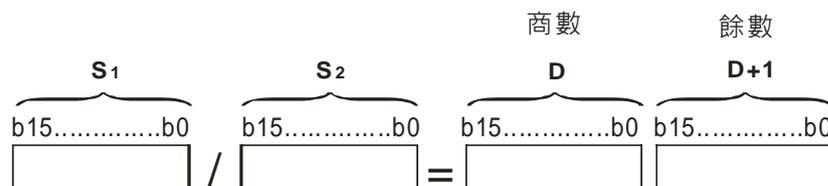
符號：



- S₁** : 被除數 Word/Double Word
- S₂** : 除數 Word/Double Word
- D** : 商及餘數 Word/Double Word

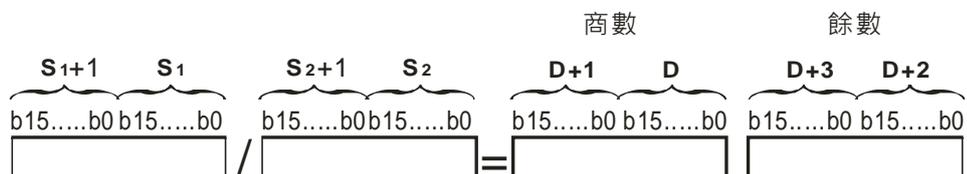
指令說明：

1. **S₁** 及 **S₂** 以有號數二進制方式相除後的商及餘數存於 **D**。
2. 32 位元指令才可以使用 **HC** 裝置。
3. 符號位元=0 為正數，符號位元=1 為負數。
4. 16 位元 BIN 除法運算：



D 運算元連續佔用兩個，**D** 儲存商，**D+1** 儲存餘數。

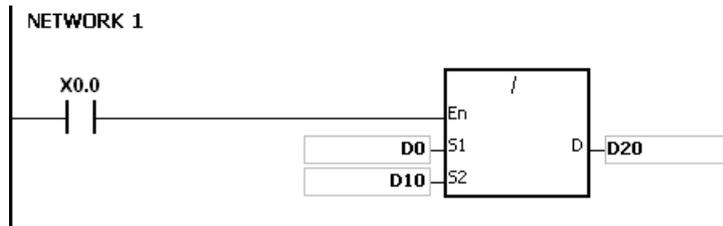
5. 32 位元 BIN 除法運算：



D 運算元連續佔用兩個，(**D+1** · **D**) 儲存商，(**D+3** · **D+2**) 儲存餘數。

程式範例：

當 X0.0=ON 時，被除數 D0 除以除數 D10 而結果商被指定放於 D20，餘數指定放於 D21 內。所得結果之正負由最高位位元之 OFF/ON 來代表正或負值。

**補充說明：**

1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 若除數為零，指令不執行，SM0=ON，錯誤碼 SR0=16#2012。
3. 16 位元指令的 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
4. 32 位元指令的 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of DWORD/DINT。

API	指令碼			運算元							功能				
0104	D	F+	P	$S_1 \cdot S_2 \cdot D$							浮點數加法				

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

脈波執行型	32 位元指令 (7-9 steps)	64 位元指令 (7-9 steps)
AH500	AH500	AH500

符號：

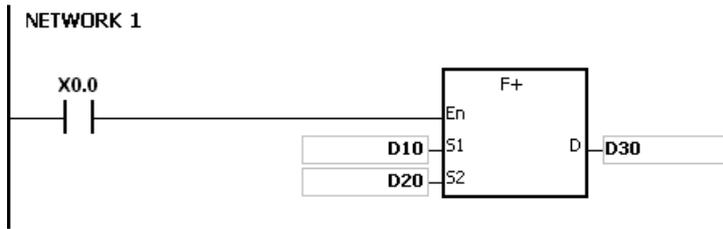
F+	F+P	S_1 : 被加數 Double Word/Long Word
En S1 S2	En S1 S2	
DF+	DF+P	S_2 : 加數 Double Word/Long Word
En S1 S2	En S1 S2	
		D : 和 Double Word/Long Word

指令說明：

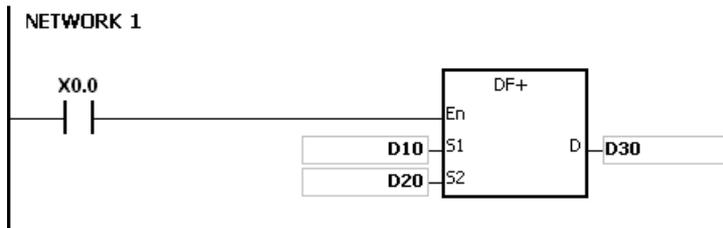
- S_1 及 S_2 以浮點數方式相加的結果存於 D 。
- 若運算結果為 0，則零旗號 $SM600=ON$ 。
- 32 位元單精度浮點數加法：
 - 演算結果為 0 時，零旗號 (Zero flag) $SM600$ 為 ON。
 - 若運算結果的絕對值小於可表示之最小浮點值，則 $D=16\#FF7FFFFFFF$ 。
 - 若運算結果的絕對值大於可表示之最大浮點值，則 $D=16\#7F7FFFFFFF$ 。
- 64 位元雙精度浮點數加法：
 - 演算結果為 0 時，零旗號 (Zero flag) $SM600$ 為 ON。
 - 若運算結果的絕對值小於可表示之最小浮點值，則 $D=16\#FFEFFFFFFFFFFFFFFF$ 。
 - 若運算結果的絕對值大於可表示之最大浮點值，則 $D=16\#7FEFFFFFFFFFFFFFFF$ 。

程式範例：

單精度浮點數加法：當 $X0.0=ON$ 時，被加數 ($D11 \cdot D10$) = $16\#3FB9999A$ (表浮點值 1.450) 加上加數 ($D21 \cdot D20$) = $16\#4046B852$ (表浮點值 3.105) 將結果存在 ($D31 \cdot D30$) = $16\#4091C28F$ (表浮點值 4.555) 之內容當中。



雙精度浮點數加法：當 X0.0=ON 時，被加數 (D13 · D12 · D11 · D10) = 16#3FF7333333333333 (表浮點值 1.450) 內容加上加數 (D23 · D22 · D21 · D20) = 16#4008D70A3D70A3D7 之內容將結果存在 (D33 · D32 · D31 · D30) = 16# 40123851EB851EB8 (表浮點值 4.555) 之內容當中。



補充說明：

當 S₁ 或 S₂ 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

API	指令碼			運算元							功能				
0105	D	F-	P	S₁ · S₂ · D							浮點數減法				

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S₁	●	●			●	●	●	●	●		●	○	●				○
S₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	32 位元指令 (7-9 steps)	64 位元指令 (7-9 steps)
AH500	AH500	AH500

符號：

F-	F-P	S₁ : 被減數 Double Word/Long Word
En S1 S2	En S1 S2	
DF-	DF-P	S₂ : 減數 Double Word/Long Word
En S1 S2	En S1 S2	
		D : 差 Double Word/Long Word

指令說明：

- S₁** 及 **S₂** 以浮點數型態相減的結果存於 **D**。
- 若運算結果為 0，則零旗號 **SM600=ON**。
- 32 位元單精度浮點數減法：

- 若運算結果的絕對值小於可表示之最小浮點值，則 **D=16#FF7FFFFFFF**。
- 若運算結果的絕對值大於可表示之最大浮點值，則 **D=16#7F7FFFFFFF**。

$$\begin{array}{r}
 \overbrace{b31 \dots b16}^{S_1 + 1} \quad \overbrace{b15 \dots b0}^{S_1} \\
 \hline
 - =
 \end{array}$$

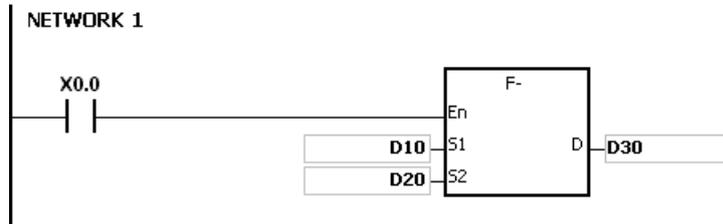
- 64 位元雙精度浮點數減法：

- 若運算結果的絕對值小於可表示之最小浮點值，則 **D=16#FFEFFFFFFFFFFFFFFF**。
- 若運算結果的絕對值大於可表示之最大浮點值，則 **D=16#7FEFFFFFFFFFFFFFFF**。

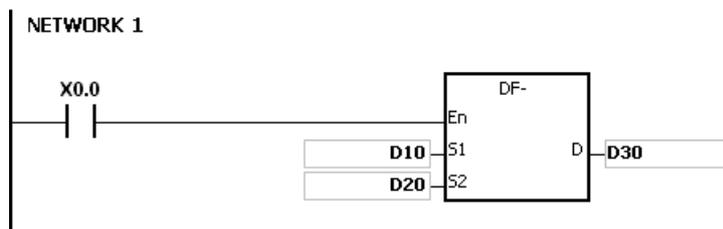
$$\begin{array}{r}
 \overbrace{b63 \dots b48}^{S_1 + 3} \quad \overbrace{b47 \dots b32}^{S_1 + 2} \quad \overbrace{b31 \dots b16}^{S_1 + 1} \quad \overbrace{b15 \dots b0}^{S_1} \\
 \hline
 - \\
 \hline
 \overbrace{b63 \dots b48}^{S_2 + 3} \quad \overbrace{b47 \dots b32}^{S_2 + 2} \quad \overbrace{b31 \dots b16}^{S_2 + 1} \quad \overbrace{b15 \dots b0}^{S_2} \\
 \hline
 - =
 \end{array}$$

程式範例：

32 位元單精度浮點數減法：當 X0.0=ON 時，被減數 (D11 · D10) 內容減去減數 (D21 · D20) 之內容將結果存在 (D31 · D30) 之內容當中。



64 位元雙精度浮點數減法：當 X0.0=ON 時，被減數 (D13 · D12 · D11 · D10) 內容減去減數 (D23 · D22 · D21 · D20) 之內容將結果存在 (D33 · D32 · D31 · D30) 之內容當中。



補充說明：

當 S₁ 或 S₂ 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

6

API	指令碼			運算元						功能					
0106	D	F*	P	$S_1 \cdot S_2 \cdot D$						浮點數乘法					

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

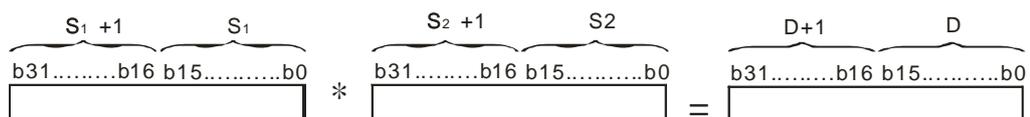
脈波執行型	32 位元指令 (7-9 steps)	64 位元指令 (7-9 steps)
AH500	AH500	AH500

符號：

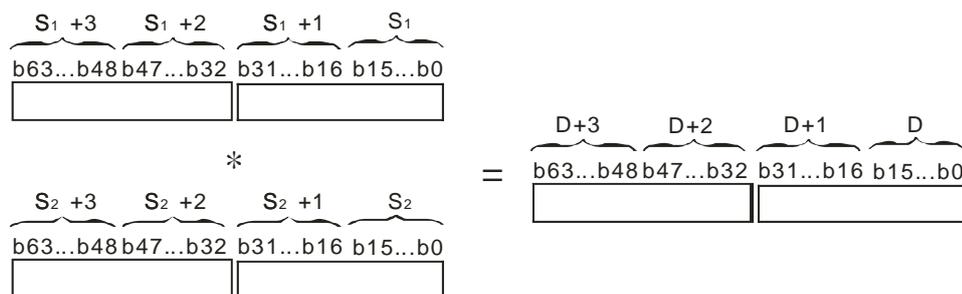
<table style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2" style="text-align: center;">F*</th></tr> <tr><td style="width: 50%;">En</td><td style="width: 50%;">En</td></tr> <tr><td>S1</td><td>D S1</td></tr> <tr><td>S2</td><td>D S2</td></tr> </table>	F*		En	En	S1	D S1	S2	D S2	<table style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2" style="text-align: center;">F*P</th></tr> <tr><td style="width: 50%;">En</td><td style="width: 50%;">En</td></tr> <tr><td>S1</td><td>D S1</td></tr> <tr><td>S2</td><td>D S2</td></tr> </table>	F*P		En	En	S1	D S1	S2	D S2	S_1 : 被乘數	Double Word/Long Word
F*																			
En	En																		
S1	D S1																		
S2	D S2																		
F*P																			
En	En																		
S1	D S1																		
S2	D S2																		
<table style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2" style="text-align: center;">DF*</th></tr> <tr><td style="width: 50%;">En</td><td style="width: 50%;">En</td></tr> <tr><td>S1</td><td>D S1</td></tr> <tr><td>S2</td><td>D S2</td></tr> </table>	DF*		En	En	S1	D S1	S2	D S2	<table style="width: 100%; border-collapse: collapse;"> <tr><th colspan="2" style="text-align: center;">DF*P</th></tr> <tr><td style="width: 50%;">En</td><td style="width: 50%;">En</td></tr> <tr><td>S1</td><td>D S1</td></tr> <tr><td>S2</td><td>D S2</td></tr> </table>	DF*P		En	En	S1	D S1	S2	D S2	S_2 : 乘數	Double Word/Long Word
DF*																			
En	En																		
S1	D S1																		
S2	D S2																		
DF*P																			
En	En																		
S1	D S1																		
S2	D S2																		
		D : 積	Double Word/Long Word																

指令說明：

1. S_1 及 S_2 以浮點數型態相乘的結果存於 D。
2. 若運算結果為 0，則零旗號 SM600=ON。
3. 32 位元單精度浮點數乘法：
 - 若運算結果的絕對值小於可表示之最小浮點值，則 D=16#FF7FFFFFFF。
 - 若運算結果的絕對值大於可表示之最大浮點值，則 D=16#7F7FFFFFFF。

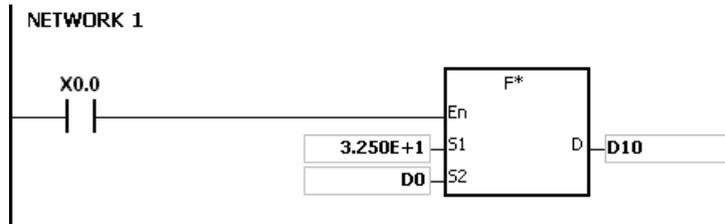


4. 64 位元雙精度浮點數乘法：
 - 若運算結果的絕對值小於可表示之最小浮點值，則 D=16#FFEFFFFFFFFFFFFFFF。
 - 若運算結果的絕對值大於可表示之最大浮點值，則 D=16#7FEFFFFFFFFFFFFFFF。

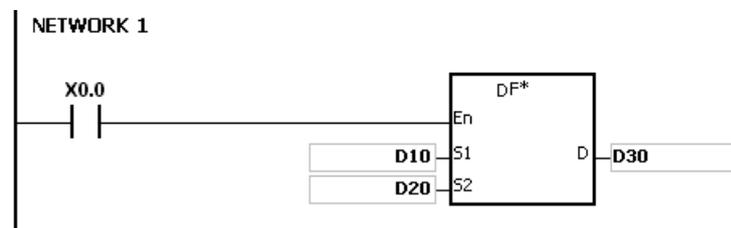


程式範例：

32 位元單精度浮點數乘法：當 X0.0=ON 時，被乘數常數 32.5 乘上乘數 (D1 · D0) 之內容，並將結果存在 (D11 · D10) 之內容當中。



64 位元雙精度浮點數乘法：當 X0.0=ON 時，被乘數 (D13 · D12 · D11 · D10) 之內容乘上乘數 (D23 · D22 · D21 · D20) 之內容，並將結果存在 (D33 · D32 · D31 · D30) 之內容當中。



補充說明：

當 S₁ 或 S₂ 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

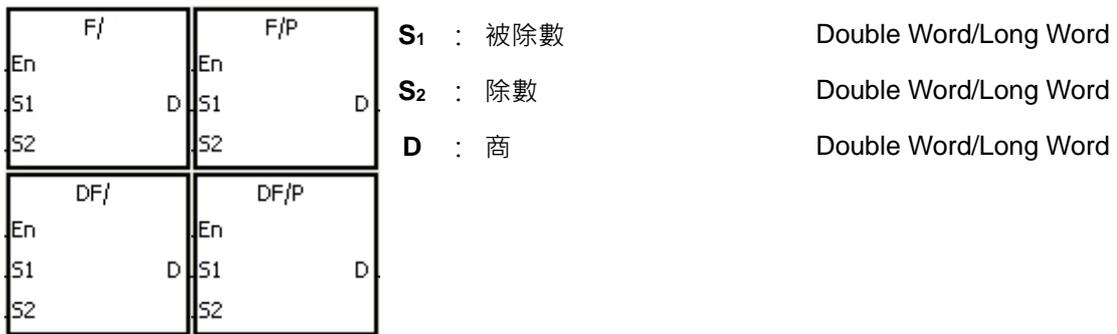
6

API	指令碼			運算元							功能				
0107	D	F/	P	S₁ · S₂ · D							浮點數除法				

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S₁	●	●			●	●	●	●	●		●	○	●				○
S₂	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	32 位元指令(7-9 steps)	64 位元指令(7-9 steps)
AH500	AH500	AH500

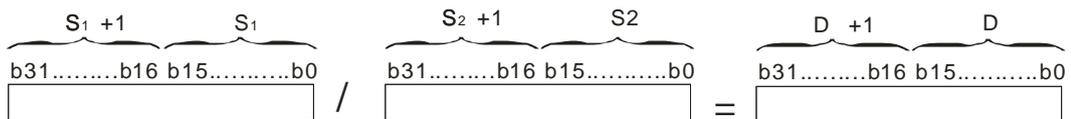
符號：



指令說明：

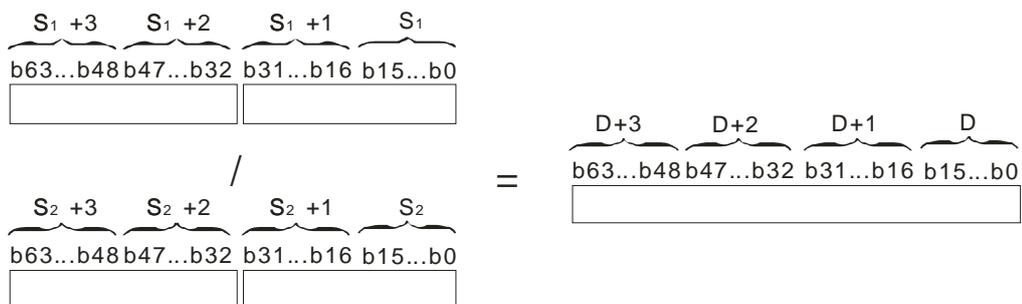
- S₁** 及 **S₂** 以單精度浮點數型態相除的結果存於 **D**。
- 若運算結果為 0，則零旗號 **SM600=ON**。
- 32 位元單精度浮點數除法：

- 若運算結果的絕對值小於可表示之最小浮點值，則 **D=16#FF7FFFFFFF**。
- 若運算結果的絕對值大於可表示之最大浮點值，則 **D=16#7F7FFFFFFF**。



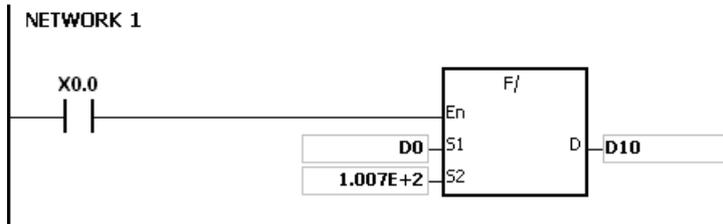
- 64 位元雙精度浮點數除法：

- 若運算結果的絕對值小於可表示之最小浮點值，則 **D=16#FFEFFFFFFFFFFFFFFF**。
- 若運算結果的絕對值大於可表示之最大浮點值，則 **D=16#7FEFFFFFFFFFFFFFFF**。

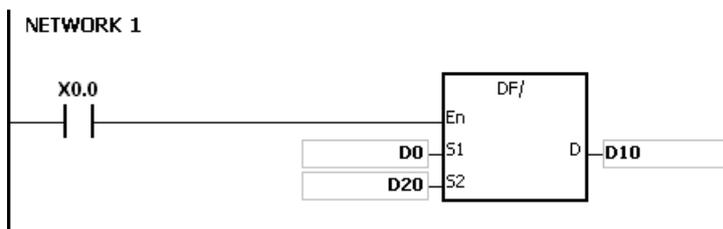


程式範例：

32 位元單精度浮點數除法：當 X0.0=ON 時，被除數 (D1 · D0) 之內容除以除數 100.7，並將結果存在 (D11 · D10) 之內容當中。



64 位元雙精度浮點數除法：當 X0.0=ON 時，被除數 (D3 · D2 · D1 · D0) 之內容除以除數 (D23 · D22 · D21 · D20) 之內容，並將結果存在 (D13 · D12 · D11 · D10) 之內容當中。



補充說明：

1. 除數為 0 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2012。
2. 當 S1 或 S2 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

6

API	指令碼			運算元							功能						
0108	D	B+	P	$S_1 \cdot S_2 \cdot D$							BCD 加法						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

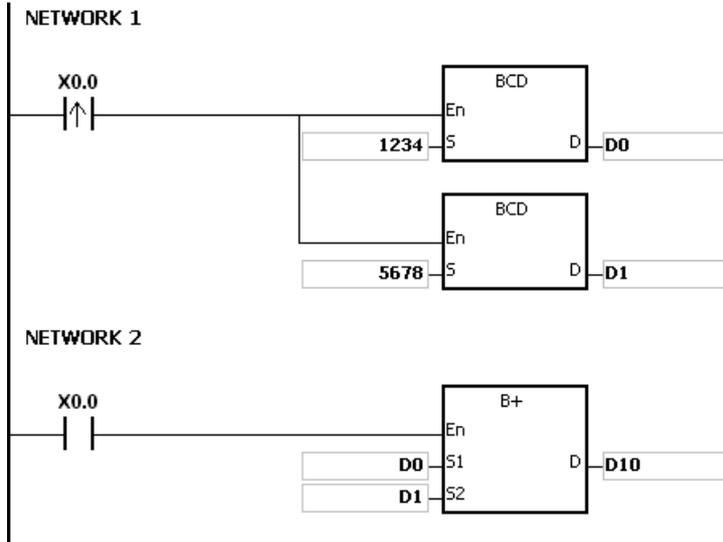
B+	B+P	S_1 : 被加數 Word/Double Word
En S1 S2	En S1 S2	
DB+	DB+P	S_2 : 加數 Word/Double Word
En S1 S2	En S1 S2	
		D : 和 Word/Double Word

指令說明：

- S_1 及 S_2 以 BCD 方式相加的結果存於 D 。
- DB+ 指令才可以使用 HC 裝置。
- BCD 值以 HEX 表示且每一個位是都要在 0~9 之間。
- 16 位元 BCD 加法：
當 $S_1=9999$ (BCD) · $S_2=0002$ (BCD) · 則 $S_1+S_2=10001$ (BCD) · 則進位被忽略 · $D=0001$ (BCD)。
- 32 位元 BCD 加法：
當 $S_1=99999999$ (BCD) · $S_2=00000002$ (BCD) · 則 $S_1+S_2=100000001$ (BCD) · 則進位被忽略 · $D=00000001$ (BCD)。

程式範例：

當 X0.0=ON 時，各別將常數 1234 和 5678 作 BCD 轉換後儲存至 D0 與 D1，接著 D0 與 D1 作 BCD 加法後將結果儲存至 D10 當中。



補充說明：

1. B+指令時當 S1 或 S2 超出 BCD 可表示範圍時 0~9999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D (BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
2. DB+指令時當 S1 或 S2 超出 BCD 可表示範圍時 0~99999999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D (BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
3. 此指令不支援零旗標 SM600、借位旗標 SM601 以及進位旗標 SM602。

6

API	指令碼			運算元							功能						
0109	D	B-	P	$S_1 \cdot S_2 \cdot D$							BCD 減法						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

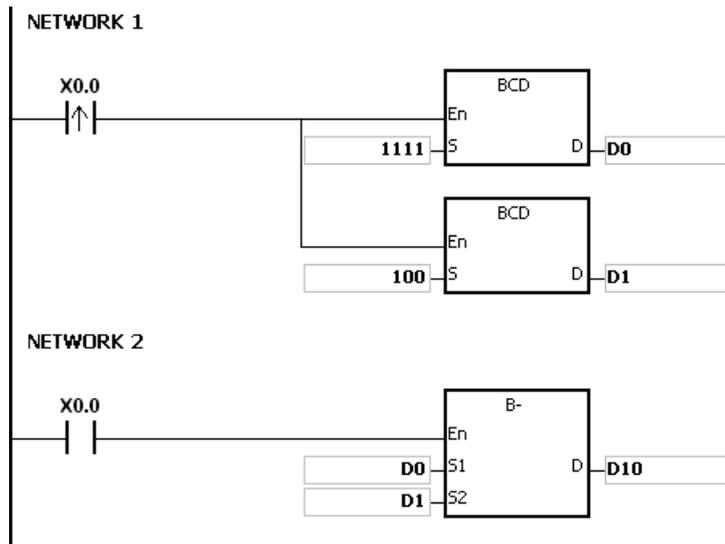
B-	B-P	S_1 : 被減數 Word/Double Word
En S1 S2	En S1 S2	
DB-	DB-P	S_2 : 減數 Word/Double Word
En S1 S2	En S1 S2	
		D : 差 Word/Double Word

指令說明：

- S_1 及 S_2 以 BCD 方式相減的結果存於 D 。
- DB-指令才可以使用 HC 裝置。
- BCD 值以 HEX 表示且每一個位是都要在 0~9 之間。
- 16 位元 BCD 減法：
 - 當 $S_1=9999$ (BCD) · $S_2=9998$ (BCD) · 則 $S_1-S_2=0001$ (BCD) · 則 $D=0001$ (BCD) 。
 - 當 $S_1=0001$ (BCD) · $S_2=9999$ (BCD) · 則 $S_1-S_2=-9998$ (BCD) · 則 $D=0002$ (BCD) 。
 - 當 $S_1=0001$ (BCD) · $S_2=0004$ (BCD) · 則 $S_1-S_2=-0003$ (BCD) · 則 $D=9997$ (BCD) 。
- 32 位元 BCD 減法：
 - 當 $S_1=99999999$ (BCD) · $S_2=99999998$ (BCD) · 則 $S_1-S_2=00000001$ (BCD) · $D=00000001$ (BCD) 。
 - 當 $S_1=00000001$ (BCD) · $S_2=99999999$ (BCD) · 則 $S_1-S_2=-99999998$ (BCD) · 則 $D=00000002$ (BCD) 。
 - 當 $S_1=00000001$ (BCD) · $S_2=00000004$ (BCD) · 則 $S_1-S_2=-00000003$ (BCD) · 則 $D=99999997$ (BCD) 。

程式範例：

當 X0.0=ON 時，各別將常數 1111 和 100 作 BCD 轉換後儲存至 D0 與 D1，接著 D0 與 D1 作 BCD 減法後將結果儲存至 D10 當中。



補充說明：

1. B-指令時當 S₁ 或 S₂ 超出 BCD 可表示範圍時 0~9999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D (BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
2. DB-指令時當 S₁ 或 S₂ 超出 BCD 可表示範圍時 0~99999999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D (BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
3. 此指令不支援零旗標 SM600、借位旗標 SM601 以及進位旗標 SM602。

6

API	指令碼			運算元						功能					
0110	D	B*	P	$S_1 \cdot S_2 \cdot D$						BCD 乘法					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

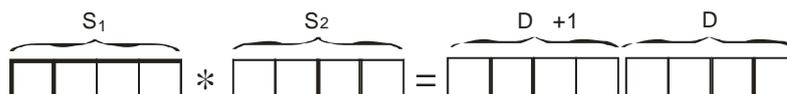
符號：

B*	B*P	S_1 : 被乘數 Word/Double Word
En	En	
S1 D	S1 D	
S2	S2	S_2 : 乘數 Word/Double Word
		D : 積 Double Word/Long Word
DB*	DB*P	
En	En	
S1 D	S1 D	
S2	S2	

指令說明：

- S_1 及 S_2 以 BCD 方式相乘的結果存於 D。
- DB*指令才可以使用 HC 裝置。
- BCD 值以 HEX 表示且每一個位是都要在 0~9 之間。
- 16 位元 BCD 乘法：

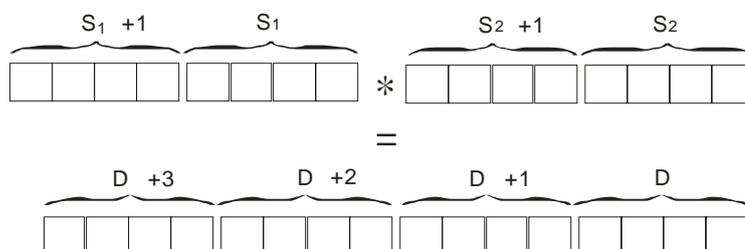
- 當 $S_1=1234$ (BCD) · $S_2=5678$ (BCD) · 則 $D=07006652$ (BCD) 。



積為 32 位元資料，儲存在 (D+1 · D) 組成的 32 位元暫存器中。

- 32 位元 BCD 乘法：

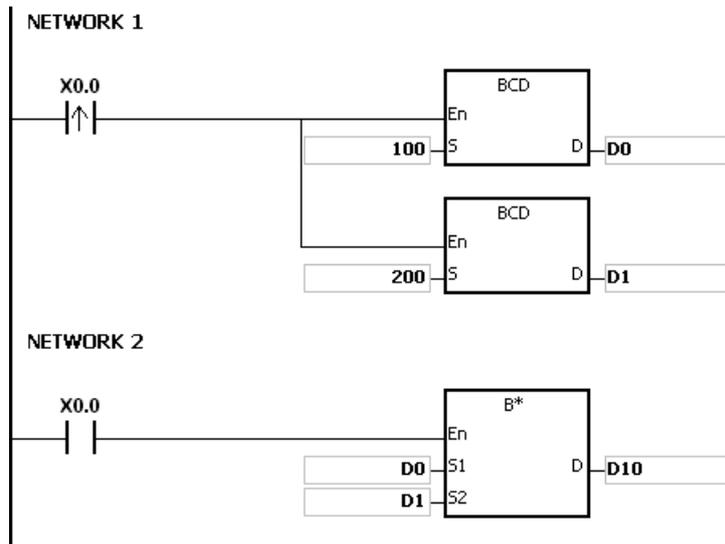
- 當 $S_1=99999999$ (BCD) · $S_2=99999998$ (BCD) · 則 $D=9999999700000002$ (BCD) 。



積為 64 位元資料，儲存在 (D+3 · D+2 · D+1 · D) 組成的 64 位元暫存器中。

程式範例：

當 X0.0=ON 時，各別將常數 100 和 200 作 BCD 轉換後儲存至 D0 與 D1，接著 D0 與 D1 作 BCD 乘法後將結果儲存至 D10~D11 當中。



補充說明：

1. B* 指令時當 S₁ 或 S₂ 超出 BCD 可表示範圍時 0~9999，指令不執行，SM0=ON，Error Code=16#200D (BCD 值以 Hex 表示有任一位置不在 0~9 的範圍內)。
2. DB* 指令時當 S₁ 或 S₂ 超出 BCD 可表示範圍時 0~99999999，指令不執行，SM0=ON，Error Code=16#200D (BCD 值以 Hex 表示有任一位置不在 0~9 的範圍內)。
3. 此指令不支援零旗標 SM600、借位旗標 SM601 以及進位旗標 SM602。

6

API	指令碼			運算元						功能					
0111	D	B/	P	$S_1 \cdot S_2 \cdot D$						BCD 除法					

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

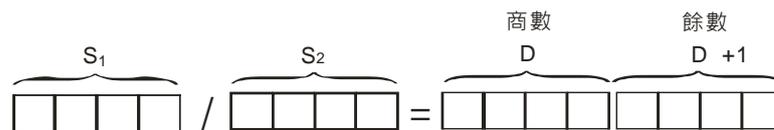
脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

B/	B/P	S_1 : 被除數 Word/Double Word S_2 : 除數 Word/Double Word D : 商及餘數 Word/Double Word
En	En	
S1	S1	
S2	S2	
DB/	DB/P	
En	En	
S1	S1	
S2	S2	

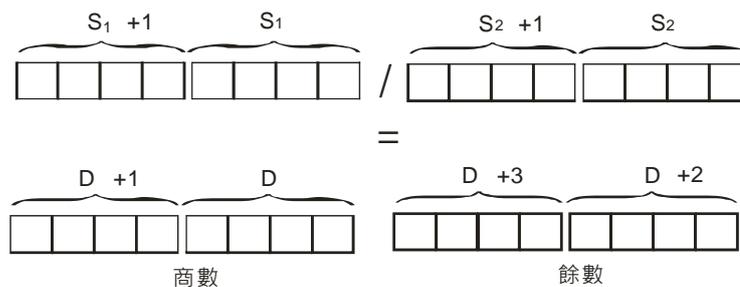
指令說明：

- S_1 及 S_2 以 BCD 方式相除的結果存於 D 。
- DB/指令才可以使用 HC 裝置。
- BCD 值以 HEX 表示且每一個位是都要在 0~9 之間。
- 16 位元 BCD 除法：
 - 當 $S_1=5678$ (BCD) , $S_2=1234$ (BCD) , 則 $D=0004$, $D+1=0742$ (BCD) 。



D 運算元連續佔用兩個 , D 儲存商 , $D+1$ 儲存餘數。

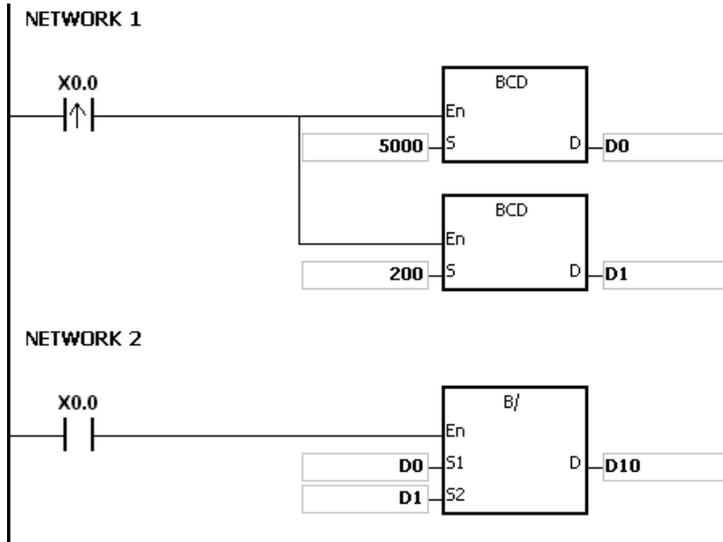
- 32 位元 BCD 除法：
 - 當 $S_1=87654321$ (BCD) , $S_2=12345678$ (BCD) , 則 $(D+1 \cdot D) = 00000007$ (BCD) , $(D+3 \cdot D+2) = 01234575$ 。



D 運算元連續佔用兩個 , $(D+1 \cdot D)$ 儲存商 , $(D+3 \cdot D+2)$ 儲存餘數。

程式範例：

當 X0.0=ON 時，各別將常數 5000 和 200 作 BCD 轉換後儲存至 D0 與 D1，接著 D0 與 D1 作 BCD 除法後將商與餘數儲存至 D10~D11 當中。



補充說明：

1. 除數為 0 時，指令不執行，SM0=ON，Error Code=16#2012。
2. B/指令時當 S₁ 或 S₂ 超出 BCD 可表示範圍時 0~9999，指令不執行，SM0=ON，Error Code=16#200D (BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
3. DB/指令時當 S₁ 或 S₂ 超出 BCD 可表示範圍時 0~99999999，指令不執行，SM0=ON，Error Code=16#200D (BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
4. 此指令不支援零旗標 SM600、借位旗標 SM601 以及進位旗標 SM602。
5. 16 位元指令的 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
6. 32 位元指令的 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of DWORD/DINT。

6

API	指令碼			運算元						功能					
0112		BK+	P	$S_1 \cdot S_2 \cdot n \cdot D$						連續區塊 BIN 加法					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">BK+</td> <td style="width: 20px;"></td> <td style="text-align: center;">BK+P</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>	BK+		BK+P	En		En	S1	D	S1	S2		S2	n		n	<p>S_1 : 被加數 Word</p> <p>S_2 : 加數 Word</p> <p>n : 要相加的筆數 Word</p> <p>D : 和 Word</p>
BK+		BK+P														
En		En														
S1	D	S1														
S2		S2														
n		n														

指令說明：

1. 連續 n 個 S_1 及 S_2 以 BIN 方式相加的結果存於 D 。
2. n 運算元的範圍為 1~256。
3. 演算結果為 0 時，零旗號 (Zero flag) SM600 為 ON。
4. 演算結果小於 -32,768 時，借位旗號 (Borrow flag) SM601 為 ON。
5. 演算結果大於 32,767 時，進位旗號 (Carry flag) SM602 為 ON。
6. 當 S_2 是裝置 (非 K/16#) 的時候：

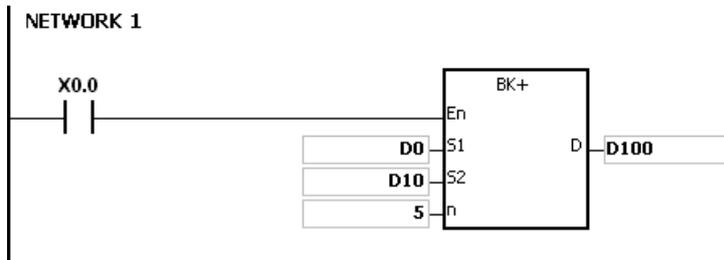
$$\begin{array}{c} S_1 \\ S_1+1 \\ \vdots \\ S_1+n-1 \end{array} \left. \begin{array}{c} 1 \\ 2 \\ \vdots \\ 5 \end{array} \right\} n + \begin{array}{c} S_2 \\ S_2+1 \\ \vdots \\ S_2+n-1 \end{array} \left. \begin{array}{c} 1 \\ 2 \\ \vdots \\ 5 \end{array} \right\} n = \begin{array}{c} D \\ D+1 \\ \vdots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

7. 當 S_2 是常數 K/16# 的時候：

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

程式範例一：

X0.0=ON 時，將被加數 D0~D4 內容與加數 D10~D14 內容作 BIN 相加後儲存至 D100~D104 當中。

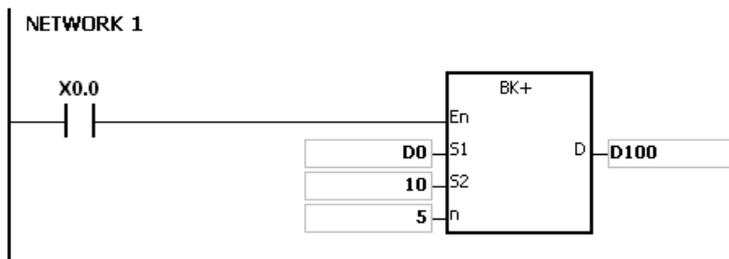


執行結果

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

程式範例二：

X0.0=ON 時，將被加數 D0~D4 的內容與加數 10 的內容作 BIN 相加後儲存至 D100~D104 當中。



執行結果

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

補充說明：

1. S₁、S₂ 或 D 裝置的起始位址~n-1 的位址，超出裝置範圍時，指令不執行 SM0=ON，錯誤碼 SR0=16#2003。
2. 若 n<1 或 n>256 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. 若 S₁~S_{1+n-1} 的裝置與 D~D+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。
4. 若 S₂~S_{2+n-1} 的裝置與 D~D+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。
5. 若 S₁~S_{1+n-1} 的裝置與 S₂~S_{2+n-1} 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。

API	指令碼			運算元						功能							
0113		BK-	P	$S_1 \cdot S_2 \cdot n \cdot D$						連續區塊 BIN 減法							

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

BK-		BK-P			
En		En		S_1 :	被減數
S1	D	S1	D	S_2 :	減數
S2		S2		n :	要相減的筆數
n		n		D :	差

指令說明：

- 連續 n 個 S_1 及 S_2 以 BIN 方式相減的結果存於 D 。
- n 運算元的範圍為 1~256。
- 演算結果為 0 時，零旗號 (Zero flag) SM600 為 ON。
- 演算結果小於 -32,768 時，借位旗號 (Borrow flag) SM601 為 ON。
- 演算結果大於 32,767 時，進位旗號 (Carry flag) SM602 為 ON。
- 當 S_2 是裝置 (非 K/16#) 的時候：

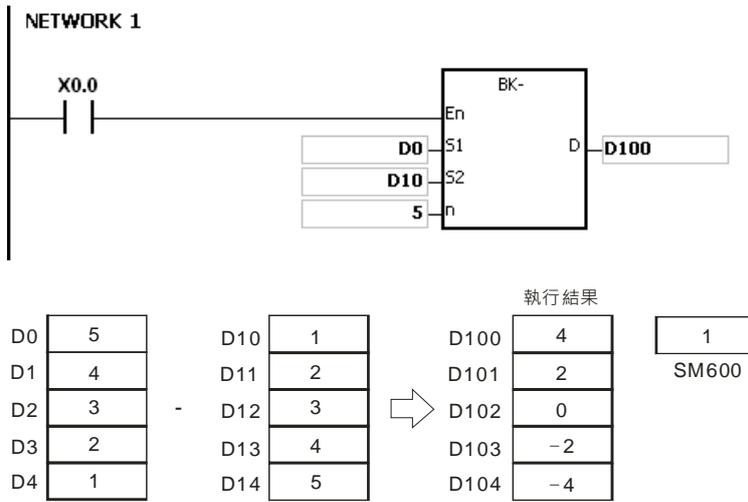
$$\begin{array}{c}
 S_1 \quad \begin{array}{|c|} \hline 5 \\ \hline \end{array} \\
 S_1+1 \quad \begin{array}{|c|} \hline 4 \\ \hline \end{array} \\
 \vdots \\
 \vdots \\
 S_1+n-1 \quad \begin{array}{|c|} \hline 1 \\ \hline \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} S_1 \\ S_1+1 \\ \vdots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

- 當 S_2 是常數 K/16# 的時候：

$$\begin{array}{c}
 S_1 \quad \begin{array}{|c|} \hline 5 \\ \hline \end{array} \\
 S_1+1 \quad \begin{array}{|c|} \hline 4 \\ \hline \end{array} \\
 \vdots \\
 \vdots \\
 S_1+n-1 \quad \begin{array}{|c|} \hline 1 \\ \hline \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} S_1 \\ S_1+1 \\ \vdots \\ \vdots \\ S_1+n-1 \end{array}} \right\} n
 \quad - \quad
 \begin{array}{c}
 S_2 \quad \begin{array}{|c|} \hline 1 \\ \hline \end{array} \\
 S_2 \quad \begin{array}{|c|} \hline 1 \\ \hline \end{array} \\
 \vdots \\
 \vdots \\
 S_2 \quad \begin{array}{|c|} \hline 1 \\ \hline \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} S_2 \\ S_2 \\ \vdots \\ \vdots \\ S_2 \end{array}} \right\} n
 \quad = \quad
 \begin{array}{c}
 D \quad \begin{array}{|c|} \hline 5-1=4 \\ \hline \end{array} \\
 D+1 \quad \begin{array}{|c|} \hline 4-1=3 \\ \hline \end{array} \\
 \vdots \\
 \vdots \\
 D+n-1 \quad \begin{array}{|c|} \hline 1-1=0 \\ \hline \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} D \\ D+1 \\ \vdots \\ \vdots \\ D+n-1 \end{array}} \right\} n$$

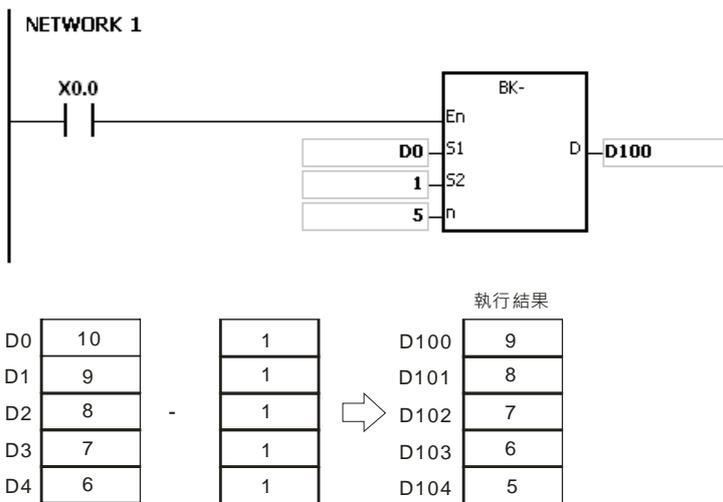
程式範例一：

X0.0=ON 時，將被減數 D0~D4 內容與減數 D10~D14 內容作 BIN 相減後儲存至 D100~D104 當中。



程式範例二：

X0.0=ON 時，將被加數 D0~D4 的內容與減數 1 的內容作 BIN 相減後儲存至 D100~D104 當中。



補充說明：

1. S₁、S₂ 或 D 裝置的起始位址~n-1 的位址，超出裝置範圍時，指令不執行 SM0=ON，Error Code=16#2003。
2. 若 n<1 或 n>256 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. 若 S₁~S₁+n-1 的裝置與 D~D+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。
4. 若 S₂~S₂+n-1 的裝置與 D~D+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。
5. 若 S₁~S₁+n-1 的裝置與 S₂~S₂+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。

API	指令碼			運算元							功能						
0114		\$+	P	S₁ · S₂ · D							字串鏈結						

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

脈波執行型	16 位元指令 (7-19 steps)	32 位元指令
AH500	AH500	-

符號：

\$+	\$+P	S ₁ : 字串 1	Word
En	En	S ₂ : 字串 2	Word
S1	D S1	D : 儲存裝置起始位址	Word
S2	S2		

指令說明：

- 該指令執行時，將 **S₁** 運算元 **S₁~16#00** (不搬 16#00) 與 **S₂** 運算元 **S₂~ 16#00** (不搬 16#00) 之間的字串鏈結後搬移至 **D** 運算元內且結尾會補上結尾碼 (16#00) 。當指令不執行時，**D** 內容不會變化。

- S₁** 運算元及 **S₂** 運算元兩個字串鏈結的結果存於 **D** 運算元的說明如下：

S ₁	b(16#62)	a(16#61)	S ₂	B(16#42)	A(16#41)	D	b(16#62)	a(16#61)
S ₁ +1	d(16#64)	c(16#63)	S ₂ +1	D(16#44)	C(16#43)	D + 1	d(16#64)	c(16#63)
S ₁ +2	(16#00)	e(16#65)	S ₂ +2	(16#00)	(16#00)	D + 2	A(16#41)	e(16#65)
						D + 3	C(16#43)	B(16#42)
						D + 4	(16#00)	D(16#44)

↑ 自動變成16#00

S ₁	b(16#62)	a(16#61)	S ₂	B(16#42)	A(16#41)	D	b(16#62)	a(16#61)
S ₁ +1	d(16#64)	c(16#63)	S ₂ +1	D(16#44)	C(16#43)	D + 1	d(16#64)	c(16#63)
S ₁ +2	e(16#65)	(16#00)	S ₂ +2	(16#00)	(16#00)	D + 2	B(16#42)	A(16#41)
						D + 3	D(16#44)	C(16#43)
						D + 4	(16#00)	(16#00)

↑ ↑ 自動變成16#00

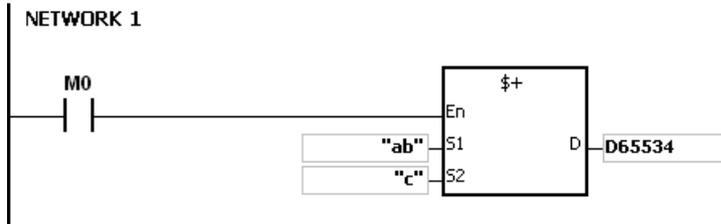
- 若 **S₁** 或 **S₂** 運算元的來源不是字串時，就要在資料結尾後面有一個結尾碼 (16#00) 。
- 若 **S₁** 或 **S₂** 運算元不是字串且該指令執行時，第一個字元就是結尾碼 (16#00) ，則還是會搬移結尾碼 (16#00) 做鏈結。

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. 字串"abcde"在 S_1 運算元表示為：

程式範例：

給定 S_1 ="ab"， S_2 ="c"，啟動條件接點 M_0 後， $D65534=16\#6261$ ， $D65535=16\#0063$



補充說明：

1. 若 S_1 或 S_2 運算元為字串時， S_1 或 S_2 運算元各可以搬移 1~31 個字元；且字串的 STEP 數計算方式為 $STEP 數 = 1 + (字元數 + 1) / 4$ (若不能整除則無條件進位)。STEP 數計算表格如下：

字元數	1~3	4~7	8~11	12~15	16~19	20~23	24~27	28~31
STEP 數	2	3	4	5	6	7	8	9

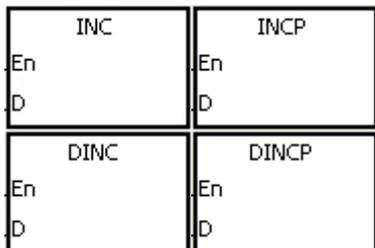
範例：字串+"ABCDE" D_0 D_{100} 則 $STEP 數 = 1 (指令) + 3 (字串) + 2 (D_0) + 2 (D_{100}) = 8$

- 當 D 裝置的空間不足以存放 S_1+S_2 字串時，指令不執行， $SM_0=ON$ ，錯誤碼 $SR_0=16\#2003$ 。
- S_1 或 S_2 裝置與 D 裝置有重疊時，指令不執行， $SM_0=ON$ ，錯誤碼 $SR_0=16\#200C$ 。
- S_1 或 S_2 沒有 16#00 當結尾時，指令不執行， $SM_0=ON$ ，錯誤碼 $SR_0=16\#200E$ 。

6

API	指令碼			運算元							功能						
0115	D	INC	P	D							BIN 加 1						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
										脈波執行型		16 位元指令 (3 steps)			32 位元指令 (3 steps)		
										AH500		AH500			AH500		

符號：



D : 目的地裝置

Word/Double Word

指令說明：

1. D 內容加 1。
2. DINC 指令才可以使用 HC 裝置。
3. 16 位元運算時，32,767 再加 1 則變為-32,768。32 位元運算時，2,147,483,647 再加 1 則變為 -2,147,483,648。

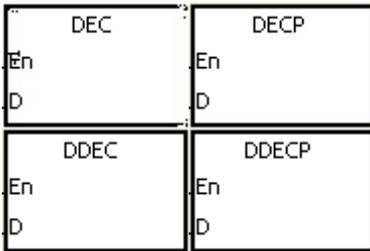
程式範例：

當 X0.0=OFF→ON 時，D0 內容自動加 1。



API	指令碼			運算元								功能					
0116	D	DEC	P	D								BIN 減 1					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脈波執行型											16 位元指令(3 steps)			32 位元指令(3 steps)			
AH500											AH500			AH500			

符號：



D : 目的地裝置

Word/Double Word

指令說明：

1. D 內容減 1。
2. DDEC 指令才可以使用 HC 裝置。
3. 16 位元運算時，-32,768 再減 1 則變為 32,767。32 位元運算時，-2,147,483,648 再減 1 則變為 2,147,483,647。

程式範例：

當 X0.0=OFF→ON 時，D0 內容自動減 1。



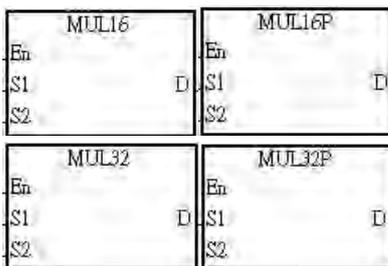
6

API	指令碼			運算元							功能						
0117	MUL16	MUL32	P	$S_1 \cdot S_2 \cdot D$							16 位元專用 BIN 乘法 32 位元專用 BIN 乘法						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



S₁ : 被乘數 Word/Double Word
 S₂ : 乘數 Word/Double Word
 D : 積 Word/Double Word

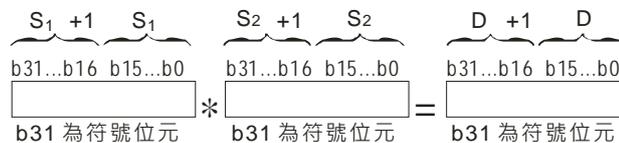
指令說明：

1. S₁ 及 S₂ 以有號數二進制方式相乘後的積存於 D。
2. MUL32 指令才可以使用 HC 裝置。
3. 16 位元 BIN 乘法運算：



積為 16 位元資料，儲存在 D 的 16 位元暫存器中，且符號位元 b15=0 為正數，符號位元 b15=1 為負數。

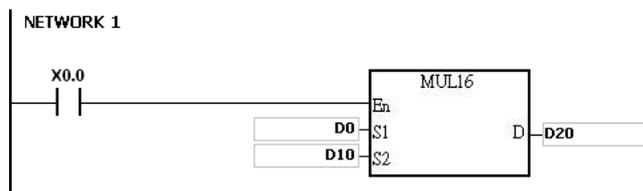
4. 32 位元 BIN 乘法運算：



積為 32 位元資料，儲存在 (D+1 · D) 組成的 32 位元暫存器中，且符號位元 b31=0 為正數，符號位元 b31=1 為負數。

程式範例：

16 位元 D0 乘上 16 位元 D10 得到一個 16 位元之積，結果存在 D20。結果之正負由最高位元 (b15) 之 OFF/ON 來指示。OFF 表示正的 (0)，同時 ON 表示負的 (1)。



$D0 \times D10 = D20$

16 位元 \times 16 位元 = 16 位元

補充說明：

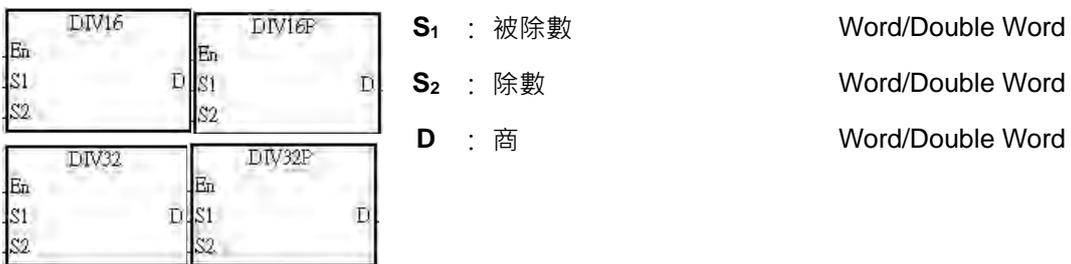
1. 當 16 位元乘法之積超出 16 位元有號數可表示範圍時，則數值比 16 位元最大正數 (K32767) 還大或者數值比最小負數 (K-32768) 還小時，設定 SM602 進位旗標為 ON；並只寫入低 16 位元的數值
2. 若 16 位元指令相乘結果需要得到完整的數值(紀錄為 32 位元)，請改用 API0102 * / *P 指令，詳細說明請參考該指令。
3. 當 32 位元乘法之積超出 32 位元有號數可表示範圍時，則數值比 32 位元最大正數 (K2147483647) 還大或者數值比最小負數 (K-2147483648) 還小時，設定 SM602 進位旗標為 ON；並只寫入低 32 位元的數值。
4. 若 32 位元指令相乘結果需要得到完整的數值 (紀錄為 64 位元)，請改用 API0102 D* / D*P 指令，詳細說明請參考該指令。

API	指令碼			運算元							功能						
0118		DIV16 DIV32	P	$S_1 \cdot S_2 \cdot D$							16 位元專用 BIN 除法 32 位元專用 BIN 除法						

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

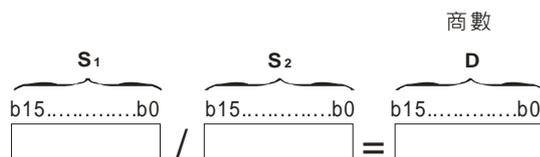
脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



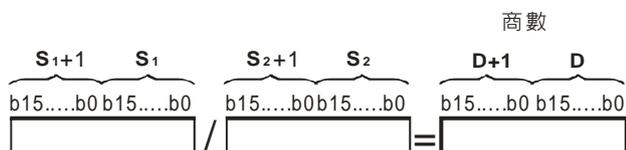
指令說明：

- S_1 及 S_2 以有號數二進制方式相除後的商存於 D。
- 32 位元指令才可以使用 HC 裝置。
- 符號位元=0 為正數，符號位元=1 為負數。
- 16 位元 BIN 除法運算：



D 運算元儲存商。

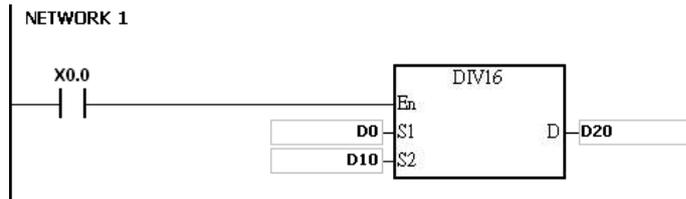
- 32 位元 BIN 除法運算：



D 運算元續佔用兩個 · (D+1 · D) 儲存商。

程式範例：

當 X0.0=ON 時，被除數 D0 除以除數 D10 而結果商被指定放於 D20。所得結果之正負由最高位元之 OFF/ON 來代表正或負值。



補充說明：

1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 若除數為零，指令不執行，SM0=ON，錯誤碼 SR0=16#2012。
3. 若需要紀錄餘數，請改用 API0103，詳細說明請參考該指令。

6.3 資料轉換指令

6.3.1 資料轉換指令一覽表

API	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
<u>0200</u>	BCD	DBCD	–	✓	BIN→BCD 變換	5
<u>0201</u>	BIN	DBIN	–	✓	BCD→BIN 變換	5
<u>0202</u>	FLT	DFLT	–	✓	BIN 整數→單精度浮點數變換	5
<u>0203</u>	FLTD	DFLTD	–	✓	BIN 整數→雙精度浮點數變換	5
<u>0204</u>	INT	DINT	–	✓	單精度浮點數→BIN 整數變換	5
<u>0205</u>	–	FINT	DFINT	✓	雙精度浮點數→BIN 整數變換	5
<u>0206</u>	MMOV	–	–	✓	16→32 位元數值轉換	5
<u>0207</u>	RMOV	–	–	✓	32→16 位元數值轉換	5
<u>0208</u>	GRY	DGRY	–	✓	BIN→GRY 碼變換	5
<u>0209</u>	GBIN	DGBIN	–	✓	GRY 碼→BIN 變換	5
<u>0210</u>	NEG	DNEG	–	✓	取負數 (取 2 的補數)	3
<u>0211</u>	–	FNEG	–	✓	單精度浮點數正負符號反相	3
<u>0212</u>	–	FBCD	–	✓	單精度浮點數→十進浮點數	5
<u>0213</u>	–	FBIN	–	✓	十進浮點數→單精度浮點數	5
<u>0214</u>	BKBCD	–	–	✓	連續區塊 BIN→BCD 變換	7
<u>0215</u>	BKBIN	–	–	✓	連續區塊 BCD→BIN 變換	7
<u>0216</u>	SCAL	–	–	✓	比例運算	9
<u>0217</u>	SCLP	DSCLP	–	✓	參數型比例運算	9
<u>0218</u>	LINE	DLINE	–	✓	COLUMN to LINE	7
<u>0219</u>	COLM	DCOLM	–	✓	LINE to COLUMN	7

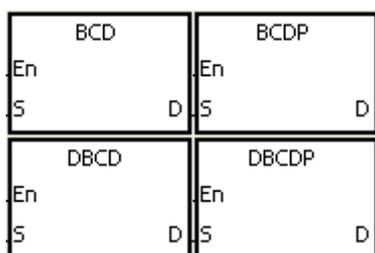
6.3.2 資料轉換指令說明

API	指令碼			運算元							功能						
0200	D	BCD	P	S · D							BIN→BCD 變換						

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：



S : 變換來源裝置 Word/Double Word
D : 變換之結果 Word/Double Word

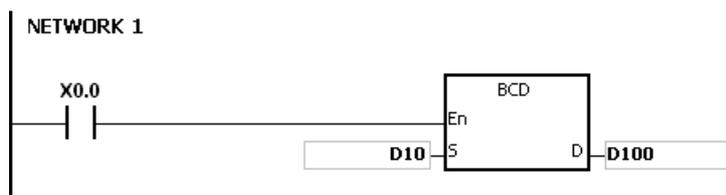
指令說明：

- 資料來源 **S** 的內容 (BIN 值) 作 BCD 的轉換，存於 **D**。
- DBCD 才可以使用 HC 裝置。
- PLC 內的四則運算及 INC、DEC 指令都是以 BIN 方式來執行。所以在應用方面，當要看到 10 進制數值的顯示器時，用 BCD 轉換即可將 BIN 值變為 BCD 值輸出。

6

程式範例：

- 當 X0.0=ON 時，D10 之 BIN 值被轉換成 BCD 值後，將結果存於 D100 當中。



- 若 D10=16#04D2=1234，則執行結果 D100=16#1234

補充說明：

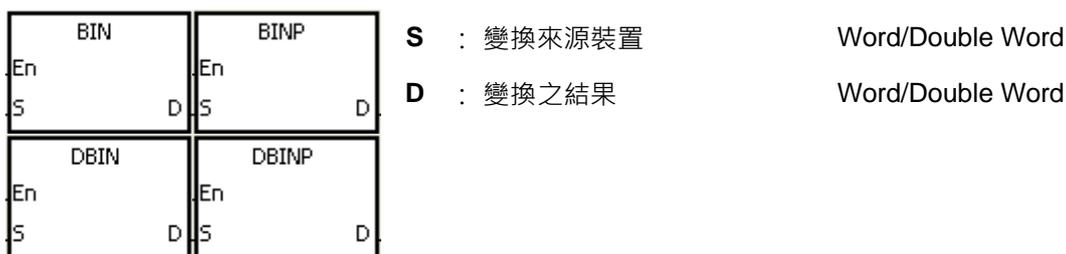
- 在 BCD 轉換結果若超過 0~9,999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D。(BCD 值以 HEX 表示有任一位數不在 0~9 的範圍內)。
- 在 DBCD 轉換結果若超過 0~99,999,999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D。(BCD 值以 HEX 表示有任一位數不在 0~9 的範圍內)。

API	指令碼			運算元								功能				
0201	D	BIN	P	S · D								BCD→BIN 變換				

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

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：

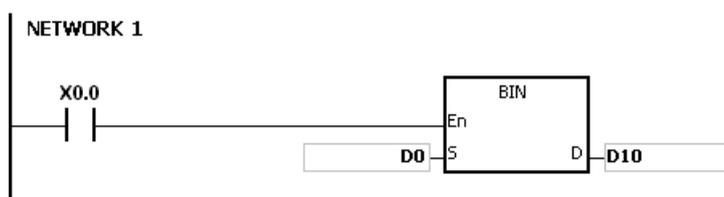


指令說明：

- 資料來源 **S** 的內容 (BCD : 0~9,999) · DBCD (0~99,999,999) 作 BIN 的轉換 · 存於 D 。
- 資料來源 **S** 的內容有效數值範圍：BCD (0~9,999) · DBCD (0~99,999,999) 。
- 32 位元指令才可以使用 HC 裝置。
- 常數 K、16#會自動轉換成 BIN，故不需運用此指令。

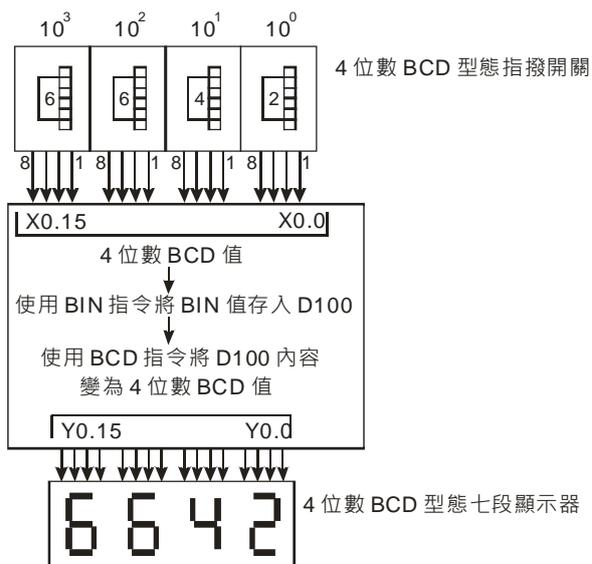
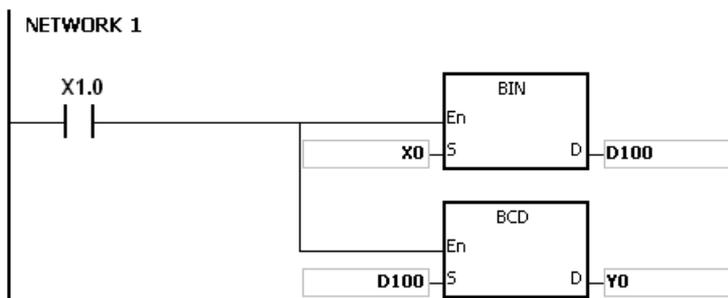
程式範例：

當 X0.0=ON 時，D0 之 BCD 值被轉換成 BIN 值後，將結果存於 D10 中。



補充說明：

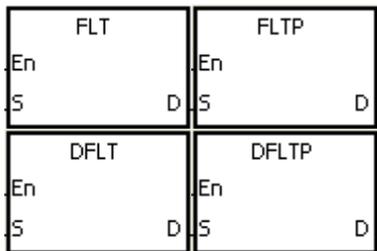
- 當 **S** 的資料內容並非為 BCD 值(以 HEX 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
- BCD 與 BIN 指令應用說明：
 - 當 PLC 要從外界讀取一個 BCD 型態指撥開關時，就必須使用 BIN 指令先將讀取到的資料轉換成 BIN 值再儲存在 PLC 內。
 - 當 PLC 要將內部儲存的資料經由外界一個 BCD 型態的 7 段顯示器顯示出來時，就必須使用 BCD 指令先將要顯示的內部資料轉換成 BCD 值再送到 7 段顯示器。
 - 當 X1.0=ON 時，將 X0.0~X0.15 BCD 值轉換成 BIN 值傳送到 D100，再將 D100 之 BIN 值轉換成 BCD 值傳送到 Y0.0~Y0.15。



6

API	指令碼			運算元								功能					
0202	D	FLT	P	S · D								BIN 整數→單精度浮點數變換					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
D	●	●			●	●	●	●	●		●	○	●				
											脈波執行型		16 位元指令(5 steps)		32 位元指令(5 steps)		
											AH500		AH500		AH500		

符號：



S：變換來源裝置

Word/Double Word

D：存放變換結果之裝置

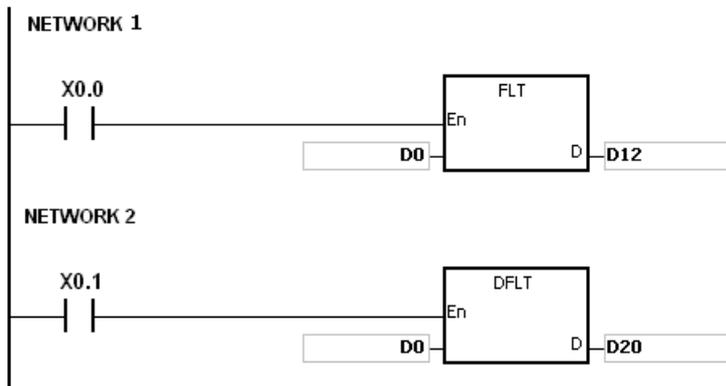
Double Word

指令說明：

- 將 BIN 整數變換成單精度浮點數值。
- FLT 中 S 不可使用 HC 裝置。
- FLT 中 S 變換來源裝置佔用 1 個暫存器，D 存放變換結果之裝置佔用 2 個暫存器。
- DFLT 中 S 變換來源裝置佔用 2 個暫存器，D 存放變換結果之裝置佔用 2 個暫存器。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則進位旗號 SM602=ON，D 存放最大浮點值。
 - 若轉換結果的絕對值小於可表示之最小浮點值，則借位旗號 SM601=ON，D 存放最小浮點值。
 - 若轉換結果為 0，則零旗號 SM600=ON。

程式範例一：

- 當 X0.0=ON 時，將 D0 (內為 BIN 整數) 變換成單精度浮點數值存放於 (D13、D12)。
- 當 X0.1=ON 時，將 D1、D0 (內為 BIN 整數) 變換成單精度浮點數值存放於 (D21、D20)。
- 若 D0=10，則 X0.0=ON，轉換後單精度浮點數值為 16#41200000，存於 32 位元暫存器 (D13、D12) 內。
- 若 32 位元暫存器 (D1、D0)=100,000，則 X0.1=ON，轉換後單精度浮點數值為 16#47C35000，存於 32 位元暫存器 (D21、D20) 內。

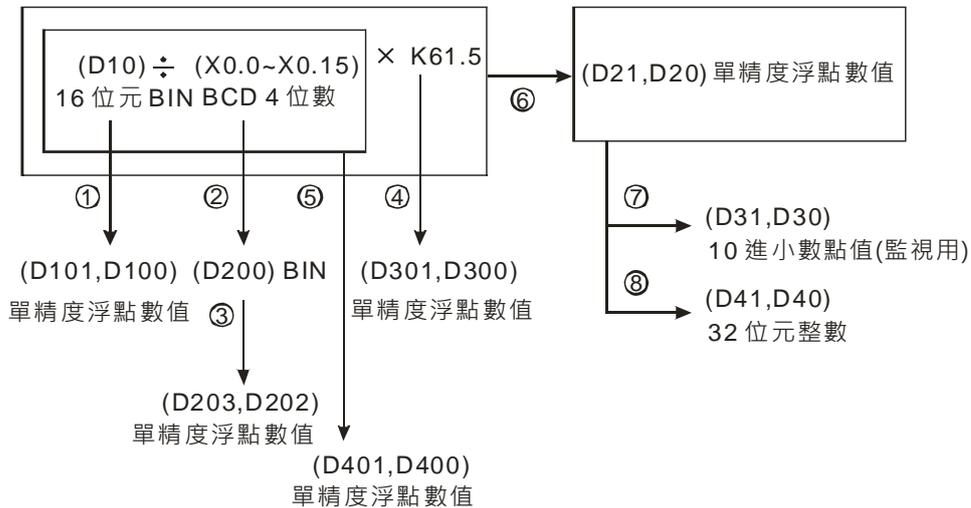


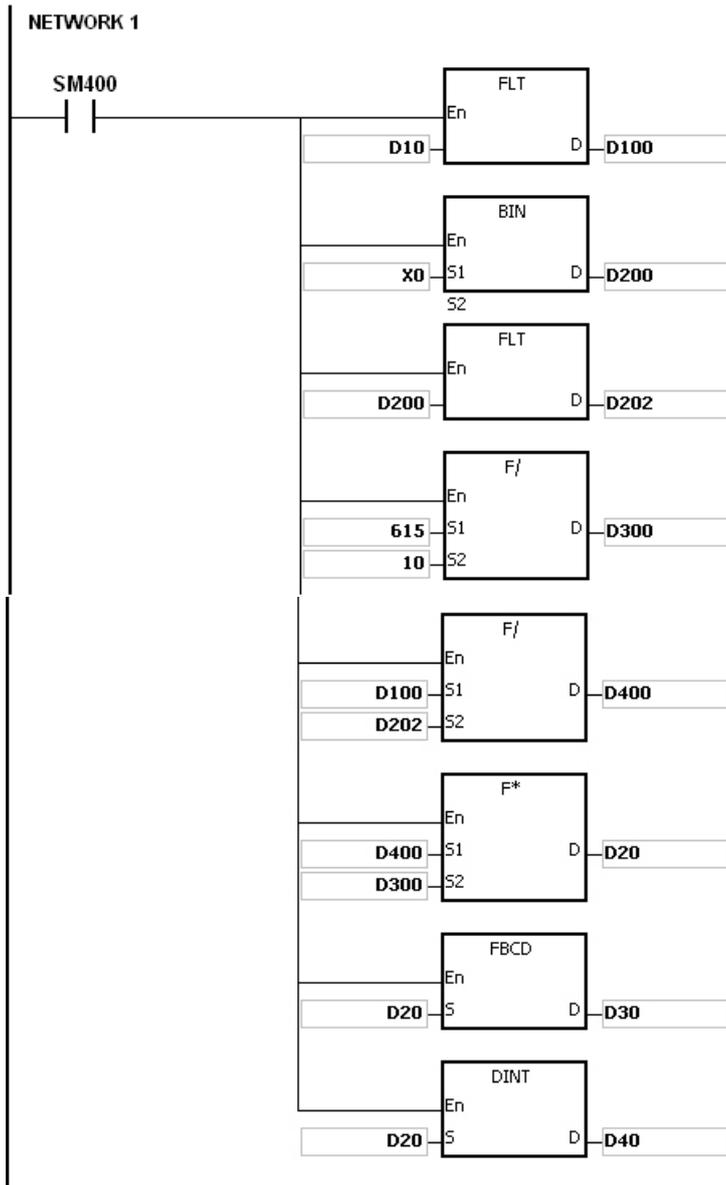
程式範例二：

使用應用指令來完成下列的算式。

- 將 D10 (內為 BIN 整數) 變換成單精度浮點數值存放於 (D101、D100)。
- 將 X0.0~X0.15 (BCD 值) 變換成 D200 (BIN 值)。
- 將 D200 (內為 BIN 整數) 變換成單精度浮點數值存放於 (D203、D202)。
- 將常數 615÷10 結果存於 D301、D300 (單精度浮點數值)。
- 單精度浮點數除法 (D101、D100) ÷ (D203、D202) 結果存於 D401、D400 (單精度浮點數值)。
- 單精度浮點數乘法 (D401、D400) × (D301、D300) 結果存於 D21、D20 (單精度浮點數值)。
- 單精度浮點數值 D21、D20 變換成 10 進小數點值 D31、D30。
- 單精度浮點數值 D21、D20 變換成 BIN 整數 D41、D40。

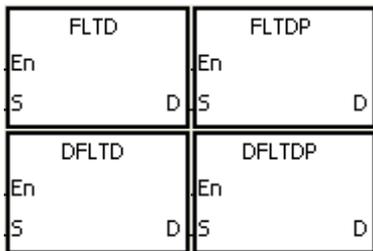
6





API	指令碼			運算元							功能						
0203	D	FLTD	P	S · D							BIN 整數轉換雙精度浮點數						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
D	●	●			●	●	●	●	●		●	○	●				
脈波執行型											16 位元指令(5 steps)			32 位元指令(5 steps)			
AH500											AH500			AH500			

符號：



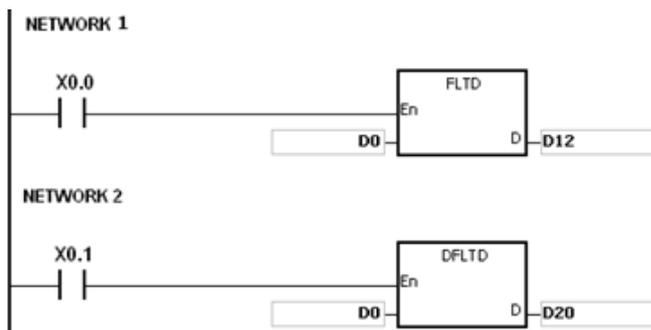
S : 變換來源裝置 Word/Double Word
D : 存放變換結果之裝置 Long Word

指令說明：

1. 當該指令執行時，將 BIN 整數變換成雙精度浮點數。
2. FLTD 中 S 不可使用 HC 裝置。
3. FLTD 中 S 變換來源裝置佔用 1 個暫存器，D 存放變換結果之裝置佔用 4 個暫存器。
4. DFLTDP 中 S 變換來源裝置佔用 2 個暫存器，D 存放變換結果之裝置佔用 4 個暫存器。
5. 若轉換結果的絕對值大於可表示之最大浮點值，則進位旗號 SM602=ON，D 存放最大浮點值。
6. 若轉換結果的絕對值小於可表示之最小浮點值，則借位旗號 SM601=ON，D 存放最小浮點值。
7. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

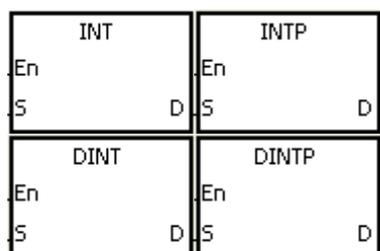
1. 當 X0.0=ON 時，將 16 位元 BIN 整數 D0 變換成雙精度浮點數值存放於(D15、D14、D13、D12)。
2. 當 X0.1=ON 時，將 32 位元 BIN 整數 (D1、D0) 變換成雙精度浮點數值存放於 (D23、D22、D21、D20)。
3. 若 D0=10，則 X0.0=ON，轉換後雙精度浮點數值為 16#4024000000000000，存於 64 位元暫存器 (D15、D14、D13、D12)。
4. 若 32 位元暫存器 (D1、D0) =100,000，則 X0.1=ON，轉換後雙精度浮點數值為 16#40F86A0000000000，存於 64 位元暫存器 (D23、D22、D21、D20)。



API	指令碼			運算元								功能					
0204	D	INT	P	S · D								單精度浮點數→BIN 整數變換					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：



S：變換來源裝置

Double Word

D：變換之結果

Word/Double Word

指令說明：

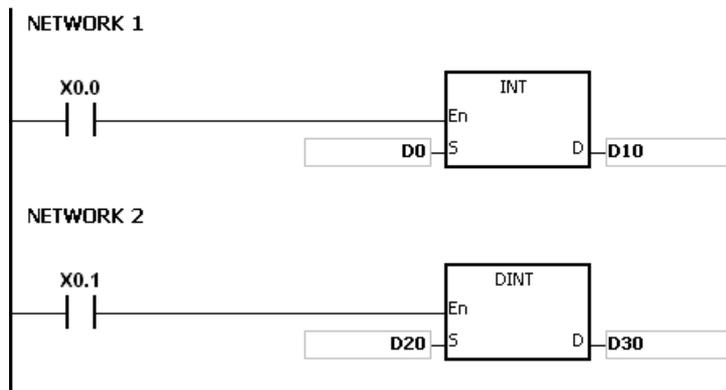
- S 所指定的暫存器內容以單精度浮點數型態被變換成 BIN 整數暫存於 D 所指定的暫存器當中，BIN 整數浮點數被捨棄。
- INT 中 S 變換來源裝置佔用 2 個暫存器，D 存放變換結果之裝置佔用 1 個暫存器。
- DINT 中 S 變換來源裝置佔用 2 個暫存器，D 存放變換結果之裝置佔用 2 個暫存器。
- INT 中 D 不可使用 HC 裝置。
- 本指令的動作與 API0202 (FLT) 指令剛好相反。
- 變換結果若為 0 時，零旗標 SM600=ON。
- 變換結果有浮點數被捨棄時，借位旗標 SM601=ON。
- 變換結果若超出下列範圍時 (溢位)，進位旗標 SM602=ON。

INT/IINTP：-32,768~32,767

DINT/DINTP：-2,147,483,648~2,147,483,647

程式範例：

- 當 X0.0=ON 時，將單精度浮點數 (D1，D0) 變換成 BIN 整數將結果存放至 D10 當中，BIN 整數浮點數被捨棄。
- 當 X0.1=ON 時，將單精度浮點數 (D21，D20) 變換成 BIN 整數將結果存放至 (D31，D30) 當中，BIN 整數浮點數被捨棄。



補充說明：

當 **S** 內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

API	指令碼			運算元							功能						
0205	D	FINT	P	S · D							雙精度浮點數轉換 BIN 整數						

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

脈波執行型	32 位元指令(5 steps)	64 位元指令(5 steps)
AH500	AH500	AH500

符號：

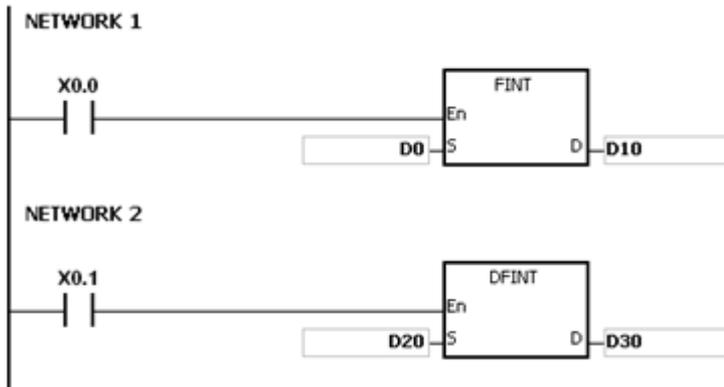
FINT En S	FINTP En S	S : 變換來源裝置 D : 變換之結果	Long Word Word/Double Word
DFINT En S	DFINTP En S		

指令說明：

- S 所指定的暫存器內容以雙精度浮點數型態被變換成 BIN 整數暫存於 D 所指定的暫存器當中，BIN 整數浮點數被捨棄。
- FINT 中 S 變換來源裝置佔用 4 個暫存器，D 存放變換結果之裝置佔用 1 個暫存器。
- DFINT 中 S 變換來源裝置佔用 4 個暫存器，D 存放變換結果之裝置佔用 2 個暫存器。
- FINT 跟 FLTP 中 D 不可使用 HC 裝置。
- 本指令的動作與 API0203 (FLTD) 指令剛好相反。
- 變換結果若為 0 時，零旗標 SM600=ON。
- 變換結果有浮點數被捨棄時，借位旗標 SM601=ON。
- 變換結果若超出下列範圍時 (溢位)，進位旗標 SM602=ON。
FINT/FINTP : -32,768~32,767
DFINT/DFINTP : -2,147,483,648~2,147,483,647

程式範例：

- 當 X0.0=ON 時，將雙精度浮點數 (D3 · D2 · D1 · D0) 變換成 BIN 整數將結果存放至 D10 當中，BIN 整數浮點數被捨棄。
- 當 X0.1=ON 時，將雙精度浮點數 (D23 · D22 · D21 · D20) 變換成 BIN 整數將結果存放至 (D31 · D30) 當中，BIN 整數浮點數被捨棄。



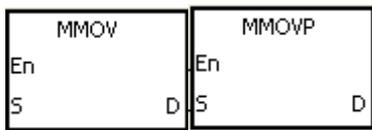
6

API	指令碼			運算元								功能					
0206		MMOV	P	S · D								16→32 位元數值轉換					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令
AH500	AH500	-

符號：



S : 變換來源裝置
D : 變換之結果

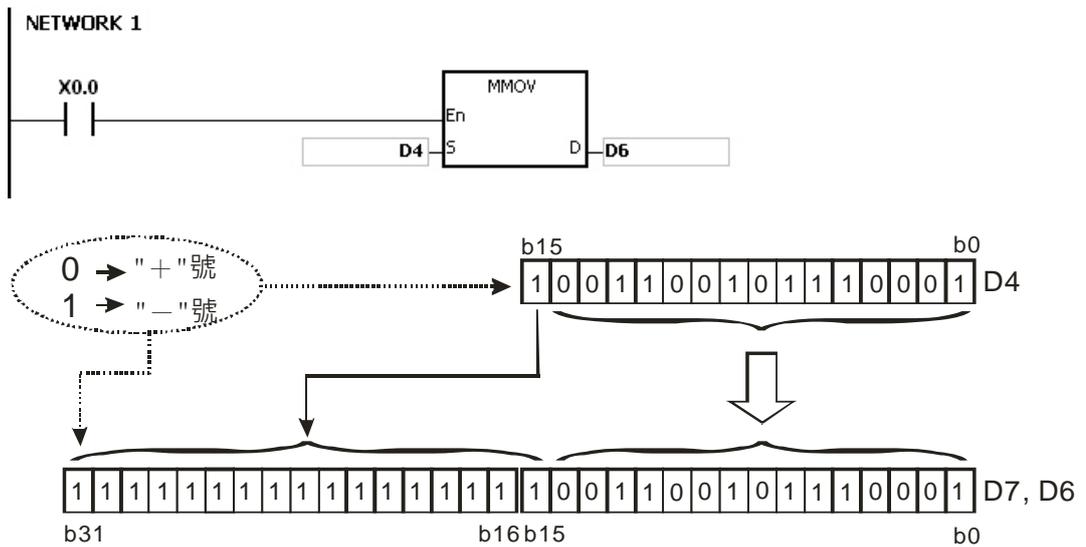
Word
Double Word

指令說明：

將 16 位元裝置 **S** 中的資料傳送到 32 位元的裝置 **D** 中，其中指定的符號位元被重複的複製存放在目的地。

程式範例：

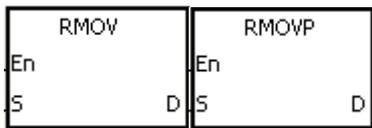
當 X0.0 為 ON 時，D4 的 b15 位元數據傳送到 (D7 · D6) 的 b15 到 b31 位元，變成負數 (和 D4 的一樣)。



6

API	指令碼			運算元							功能						
0207		RMOV	P	S · D							32→16 位元數值轉換						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				
脈波執行型											16 位元指令(5 steps)					32 位元指令	
AH500											AH500					-	

符號：



S : 變換來源裝置 Double Word
D : 變換之結果 Word

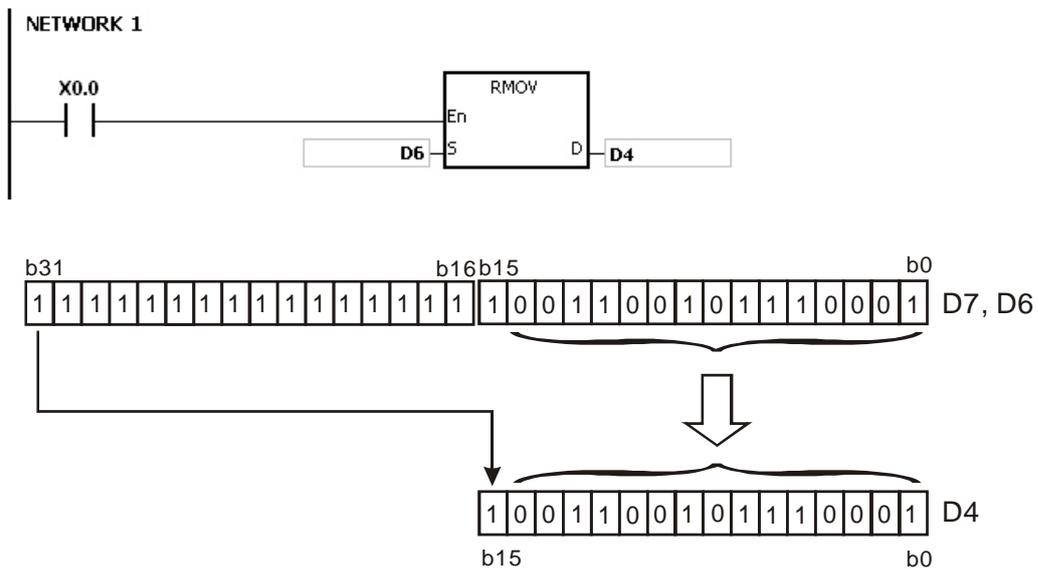
指令說明：

將 32 位元裝置 **S** 中的資料傳送到 16 位元的裝置 **D** 中，其中指定的符號位元被保留。

程式範例：

當 X0.0 為 0N 時，S 中最高位元(D7 : b31)被傳送到 D 中最高位元(D4 : b15)中，其它位元(b0~b14)則對應傳送，而 b15~b30 被忽略未被傳送。

6

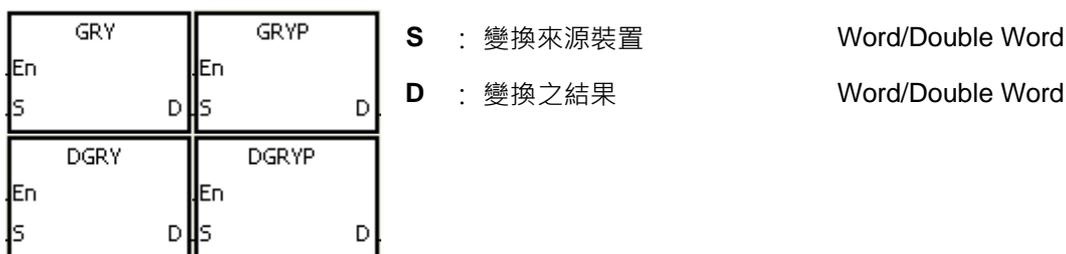


API	指令碼			運算元								功能					
0208	D	GRY	P	S · D								BIN→GRAY 碼變換					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：

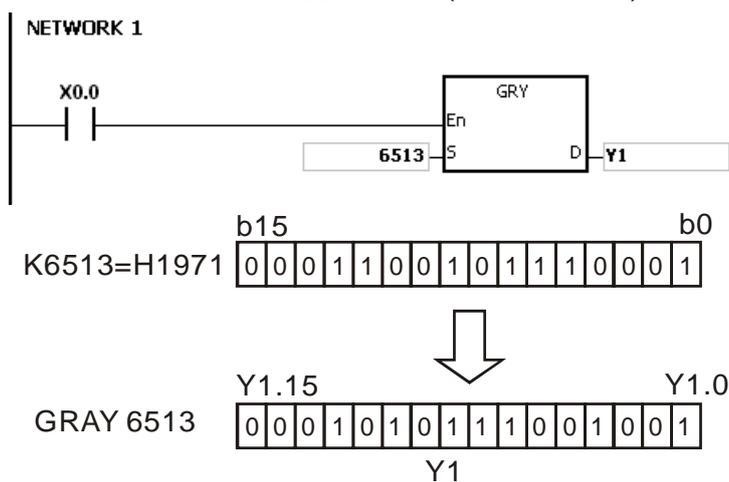


指令說明：

- 將 S 所指定裝置之內容值 (BIN 值) 變換格雷碼 (GRAY CODE) 後存放到 D 所指定之裝置中。
- DGRY 才可使用 HC 裝置。
- S 的有效範圍如下所示：
 16 位元指令：0~32,767
 32 位元指令：0~2,147,483,647

程式範例：

當 X0.0=ON 時，將常數 6513 變換格雷碼 (GRAY CODE) 後存放到 Y1.0~Y1.15。



補充說明：

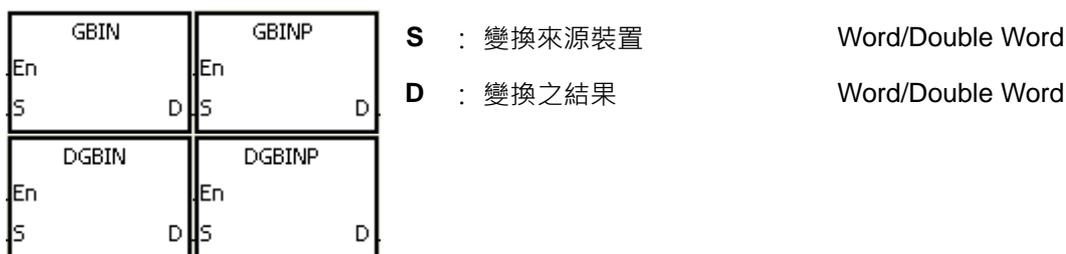
S 的內容值小於 0，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元							功能					
0209	D	GBIN	P	S · D							GRAY 碼→BIN 變換					

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：



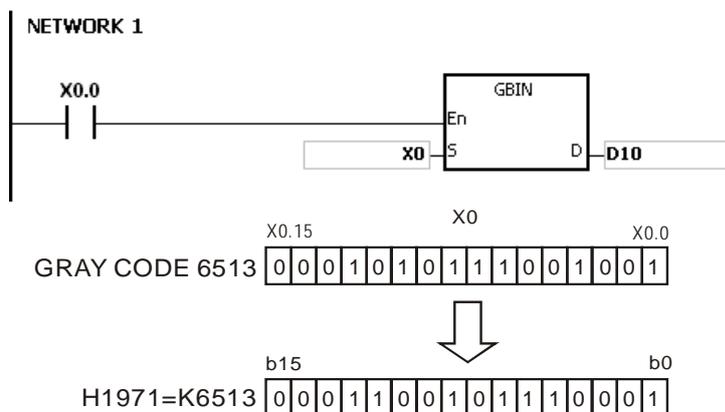
指令說明：

- 將 **S** 所指定裝置之內容值 (格雷碼 (GRAY CODE)) 變換成 BIN 值後存放到 **D** 所指定之裝置中。
- 本指令將連接於 PLC 輸入端的絕對位置型編碼器 (此編碼器的輸出值通常是格雷碼) 的內容變換成 BIN 值存放到指定的暫存器當中。
- S** 的有效範圍如下所示：
 16 位元指令：0~32,767
 32 位元指令：0~2,147,483,647

6

程式範例：

當 X0.0 時，將 X0.0~X0.15 輸入點所連接之絕對位置型編碼器其格雷碼 (GRAY CODE) 變換成 BIN 值後存放到 D10 中。



補充說明：

S 的內容值小於 0，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元							功能						
	D	NEG	P	D							2 的補數						
0210																	
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
脈波執行型											16 位元指令 (3 steps)			32 位元指令 (3 steps)			
AH500											AH500			AH500			

符號：



D : 欲取 2 的補數之裝置

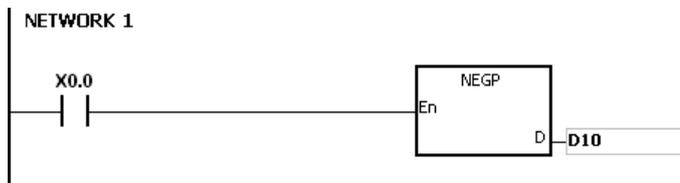
Word/Double Word

指令說明：

1. 本指令可將負數的 BIN 值轉換成絕對值。
2. DNEG 才可使用 HC 裝置。
3. 本指令一般都是使用脈波執行型指令 (NEGP、DNEGP)。

程式範例一：

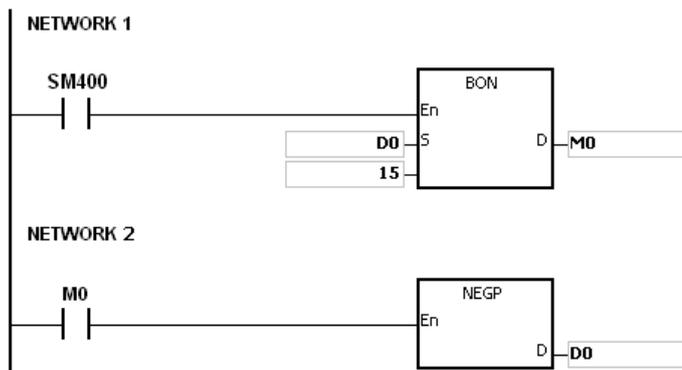
當 X0.0=OFF→ON 時，D10 內容的各位元全部反相 (0→1、1→0) 後再加 1 存放於原暫存器 D10 當中。



程式範例二：

求負數的絕對值：

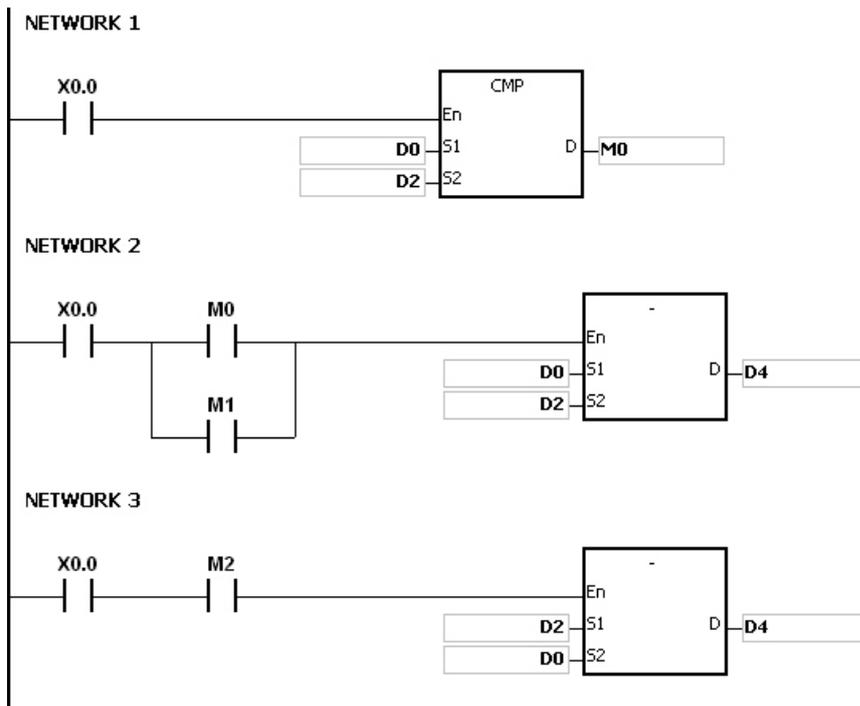
1. 當 D0 的第 15 個位元為“1”時，M0=ON。(D0 表示為負數)
2. M0=ON 時，用 NEG 指令將 D0 取 2 的補數可得到其絕對值。



程式範例三：

減法運算之差取絕對值，當 X0.0=ON 時：

1. 若 $D0 > D2$ 時，M0=ON。
2. 若 $D0 = D2$ 時，M1=ON。
3. 若 $D0 < D2$ 時，M2=ON。
4. 此可得 D4 保持為正值。



6

補充說明：

數的表現及絕對值：

1. 正負數是以暫存器最上位 (最左邊) 的位元內容來表現，為“0”時，為正數、為“1”時，為負數。
2. 遇到負數時，可使用 NEG 指令將它轉成絕對值。

(D0)=2

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

(D0)=1

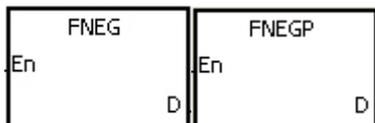
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0)=0

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

API	指令碼			運算元								功能						
0211		FNEG	P	D								單精度浮點數正負符號反相						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
D	●	●			●	●	●	●	●		●	○	●					
											脈波執行型		32 位元指令(3 steps)		64 位元指令			
											AH500		AH500		-			

符號：



D：欲取符號反向之裝置 Double Word

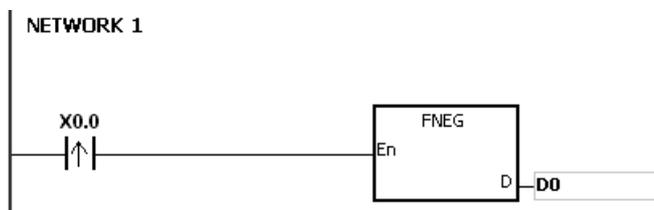
指令說明：

將裝置 D 中的內容，以單精度浮點數的型態做正負符號反相。

程式範例：

執行前 (D1、D0) = 16#AE0F9000 (負值)，當 X0.0 從 OFF→ON 時，(D1、D0) 的資料以單精度浮點數做正負符號的反相運作，因此執行後 (D1、D0) = 16#2E0F9000 (正值)。

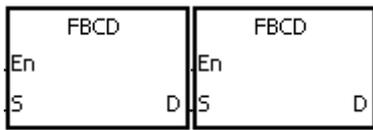
執行前 (D1、D0) = 16#2E0F9000 (正值)，當 X0.0 從 OFF→ON 時，(D1、D0) 的資料以單精度浮點數做正負符號的反相運作，因此執行後 (D1、D0) = 16#AE0F9000 (負值)。



API	指令碼			運算元							功能						
0212		FBCD	P	S · D							單精度浮點數→十進浮點數						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	32 位元指令(5 steps)	64 位元指令
AH500	AH500	-

符號：



S : 資料來源 Double Word
D : 變換的結果 Double Word

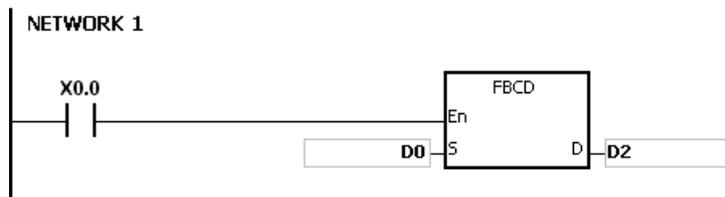
指令說明：

- 將 **S** 所指定的暫存器中的單精度浮點數變換成十進制浮點數寄存於 **D** 所指定的暫存器當中。
- PLC 是以單精度浮點數型態作浮點數運算的依據。FBCD 指令就是用來將單精度浮點數變換成十進制浮點數型態的專用指令。
- 旗標：SM600 為零旗標，SM601 為借位旗標，SM602 為進位旗標。
 若轉換結果的絕對值大於可表示的最大浮點值，則進位旗標 SM602=ON。
 若轉換結果的絕對值小於可表示的最小浮點值，則借位旗標 SM601=ON。
 若轉換結果為 0，則零旗標 SM600=ON。

6

程式範例：

當 X0.0= ON 時 (D1、D0) 內的單精度浮點數被轉換成十進制浮點數存於 (D3、D2)。



2 進小數點 D1 D0 實數 23 個位元，指數 8 個位元，符號 1 個位元



10 進小數點 D3 D2 指數 實數 數學式表示 ⇨ $[D2] \times 10^{[D3]}$

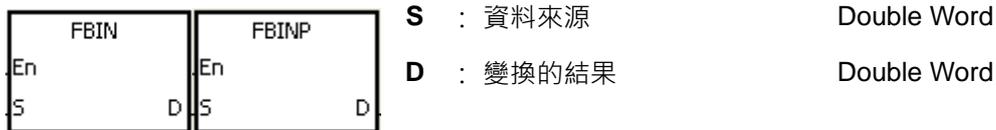
補充說明：

當 **S** 內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

API	指令碼			運算元								功能					
0213		FBIN	P	S · D								十進浮點數→單精度浮點數					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	32 位元指令(5 steps)	64 位元指令
AH500	AH500	-

符號：

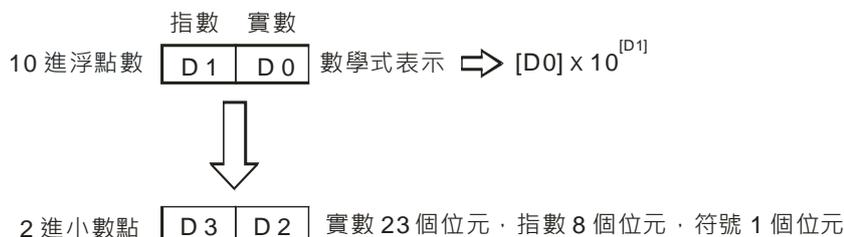
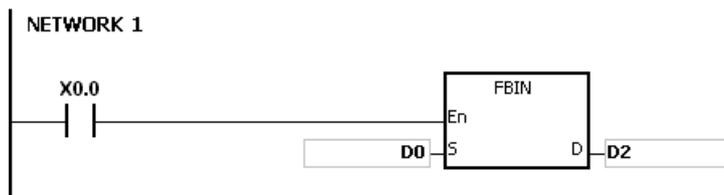


指令說明：

1. 將 S 所指定的暫存器中的十進制浮點數變換成單精度浮點數並寄存於 D 所指定的暫存器當中。
2. 例如：S=1234 · S+1=3 將變換成 S=1.234x10⁶。
3. D 必須是單精度浮點數形式，S 和 S+1 中分別用十進制表示實數和指數。
4. FBIN 指令就是用來將十進浮點數變換成單精度浮點數型態的專用指令。
5. 十進浮點數實數範圍為-9,999~+9,999，指數範圍為-41~+35，實際 PLC 十進浮點數的範圍為 ±1175x10⁻⁴¹ 到 ±3402x10⁺³⁵。若運算結果為 0，則零旗標 SM600=ON。

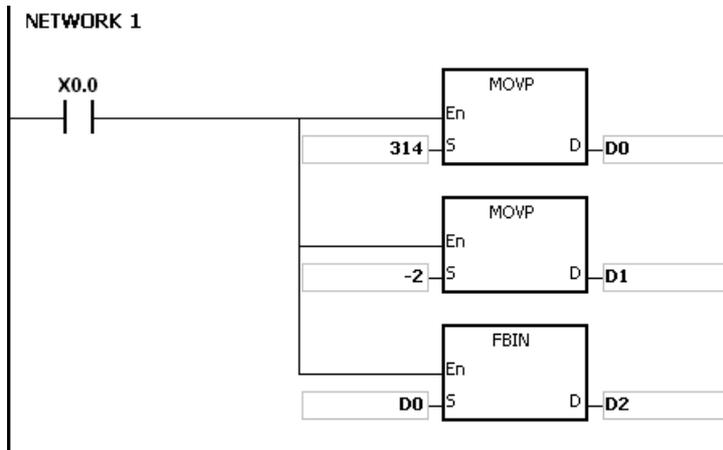
程式範例一：

當 X0.0=ON 時，指定暫存器 (D1、D0) 中的十進制浮點數轉換成單精度浮點數並寄存於 (D3、D2) 中。



程式範例二：

1. 在進行浮點數運算前必須適用 FLT 指令 BIN 整數變換成單精度浮點數，變換的前提是被變換值必須是 BIN 整數，然而，FBIN 指令可將浮點數值變換成單精度浮點數。
2. 當 X0.0=ON 時，將 K314 搬移到 D0，將 K-2 搬移到 D1，組成十進制浮點數型態 ($3.14=314 \times 10^{-2}$)。



補充說明：

若 S 運算元的十進浮點數實數範圍不在 -9,999~+9,999 之間或指數範圍不在 -41~+35 之間，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。

API	指令碼		運算元								功能					
0214	BKBCD	P	S · n · D								連續區塊 BIN→BCD 變換					

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：

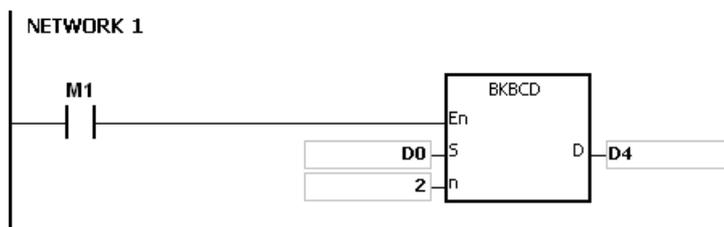


指令說明：

1. 將 n 筆的資料來源 S 的內容 (BIN 值) 作 BCD 的轉換，存於 D。
2. n=1~256。

程式範例：

當 M1=ON 時，將 D0、D1 連續兩筆資料中的 BIN 轉換成 BCD 值並存於 D4、D5 中。



補充說明：

1. 若 n<1 或 n>256 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
2. S+n-1，D+n-1，裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 在 BCD 變換結果若超過 0~9,999，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#200D。(BCD 值以 HEX 表示有任一位置不在 0~9 的範圍內)。
4. 若 S~S+n-1 的裝置與 D~D+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。

API	指令碼			運算元							功能						
0215		BKBIN	P	S · n · D							連續區塊 BCD→BIN 變換						

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：



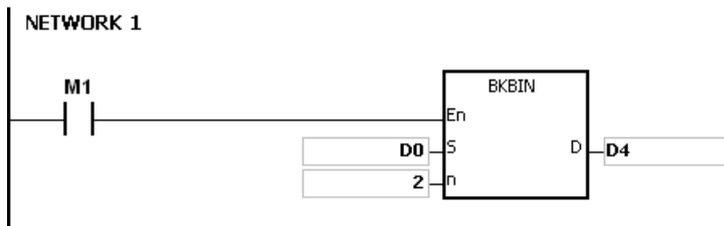
- S : 資料來源 Word
- n : 欲轉換的資料數 Word
- D : 變換的結果 Word

指令說明：

- 將 n 筆的資料來源 S 的內容 (BCD : 0~9,999) 作 BIN 的轉換，存於 D。
- 資料來源 S 的內容有效數值範圍：BCD (0~9,999)。
- n=1~256。

程式範例：

當 M1=ON 時，將 D0、D1 連續兩筆資料中的 BCD 轉換成 BIN 並存於 D4、D5 中。



補充說明：

- 若 n<1 或 n>256 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
- S+n-1，D+n-1，裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
- 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一數不在 0~9 的範圍內)，則將會產生運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#200D。
- 若 S~S+n-1 的裝置與 D~D+n-1 的裝置有重疊時，指令不執行，SM0=ON，錯誤碼 SR0=16#200C。

API	指令碼		運算元							功能							
0216		SCAL	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							比例值運算						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●	○	●	○	○		
S ₂	●	●			●	●		●	●		●	○	●	○	○		
S ₃	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

SCAL		SCALP		S ₁	Word
En		En		S ₂	Word
S1	D	S1	D	S ₃	Word
S2		S2		D	Word
S3		S3			

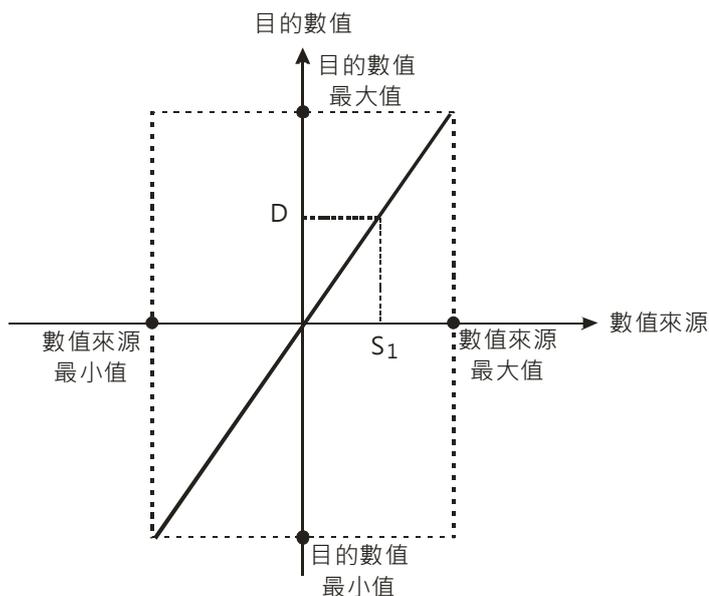
指令說明：

- 指令內部運算公式為 $D = (S_1 \times S_2) \div 1,000 + S_3$ 。
- 其中 S₂ 與 S₃ 的數值須由使用者依下列斜率與偏移量公式先行運算，然後將小數點 4 捨 5 入後，再取 16 位元的整數值輸入。

斜率公式為 $S_2 = [(\text{目的數值最大值} - \text{目的數值最小值}) \div (\text{數值來源最大值} - \text{數值來源最小值})] \times 1,000$

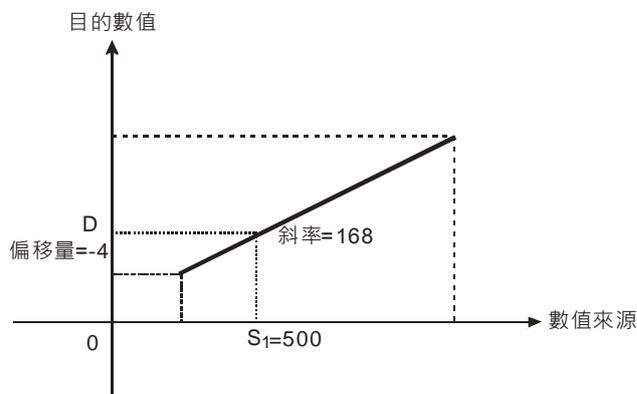
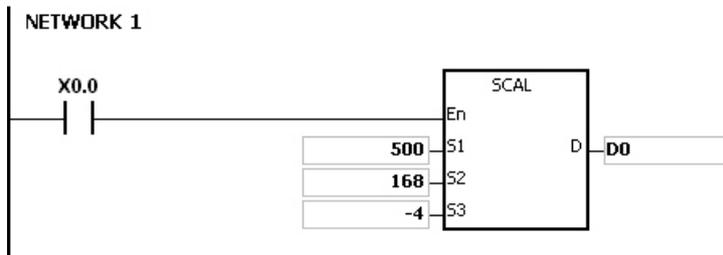
偏移量公式為 $S_3 = \text{目的數值最小值} - \text{數值來源最小值} \times S_2 \div 1,000$

其輸出曲線將如下圖所示：



程式範例一：

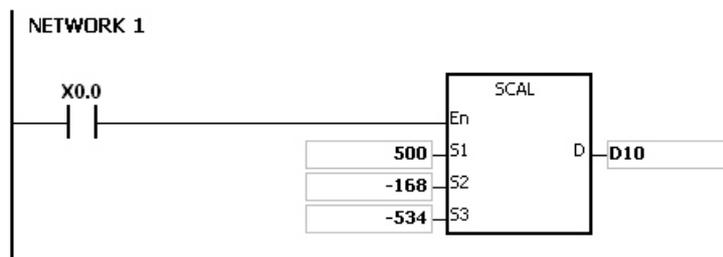
1. 已知 S_1 數值資料來源為 500 · S_2 斜率為 168 · S_3 偏移量為 -4 · 當 X0.0=ON 時 · SCAL 指令執行 · 可在 D0 得到所要求的比例值。
2. 運算方式： $D0 = (500 \times 168) \div 1,000 + (-4) = 80$

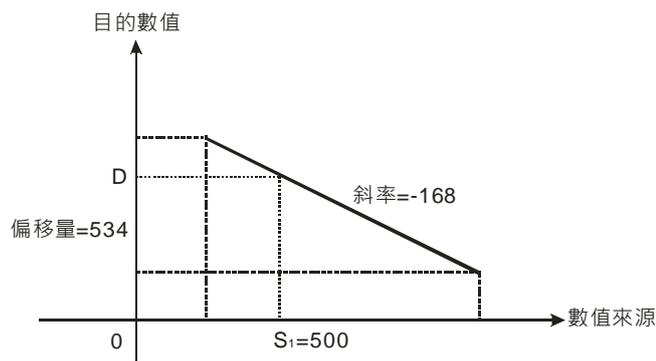


6

程式範例二：

1. 已知 S_1 數值資料來源為 500 · S_2 斜率為 -168 · S_3 偏移量為 534 · 當 X0.0=ON 時 · SCAL 指令執行 · 可在 D10 得到所要求的比例值。
2. 運算方式： $D10 = (500 \times -168) \div 1,000 + 534 = 450$





補充說明：

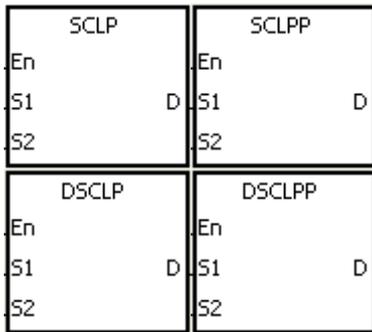
1. 此 SCAL 指令適用於已知斜率與偏移量，若不知斜率與偏移量建議使用 SCLP 指令來做運算。
2. 輸入參數 S_2 時，其輸入數值必須為 $-32,768 \sim 32,767$ 之間的數值(實際數值為 $-32,768 \sim 32,767$)，若是 S_2 實際數值超過範圍時，請改用 SCLP 指令運算。
3. 使用者運用斜率換算公式時，須注意數值來源最大值，必須大於數值來源最小值，而目的數值最大值，並不限制大於目的數值最小值。
4. 若 D 的值 $> 32,767$ 則 $D = 32,767$ ，若 D 的值 $< -32,768$ 則 $D = -32,768$ 。

API	指令碼			運算元							功能						
0217	D	SCLP	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							參數型比例值運算						

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

脈波執行型	16 位元指令(9 steps)	32 位元指令(9 steps)
AH500	AH500	AH500

符號：



- S_1 : 數值資料來源 Word/Double word
- S_2 : 參數 Word/Double word
- D : 目的地裝置 Word/Double word

指令說明：

1. 32 位元指令才可以使用 HC 裝置。
2. 16 位元指令 S_2 參數設定內容如下。

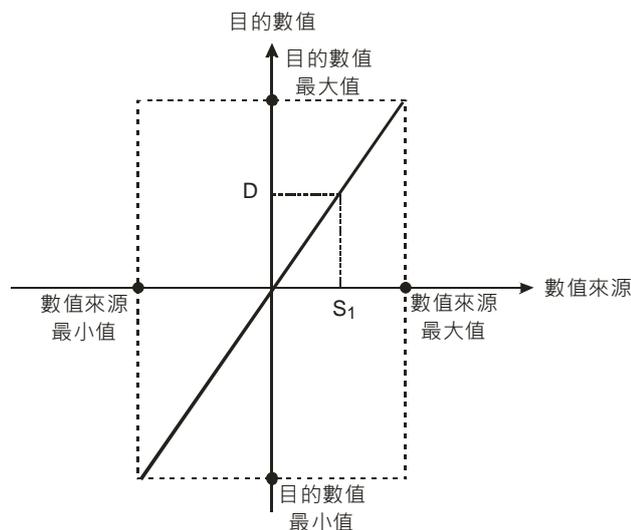
裝置編號	參數名稱與說明	設定範圍
S_2	數值來源最大值	-32,768~32,767
S_2+1	數值來源最小值	-32,768~32,767
S_2+2	目的數值最大值	-32,768~32,767
S_2+3	目的數值最小值	-32,768~32,767

3. 16 位元指令 S_2 運算元將連續佔用 4 個裝置。
4. 32 位元指令 S_2 參數設定內容如下。

裝置編號	參數名稱與說明	設定範圍	
		整數	浮點數
S_2 、 S_2+1	數值來源最大值	-2,147,483,648~ 2,147,483,647	32bit 浮點數範圍
S_2+2 、3	數值來源最小值		
S_2+4 、5	目的數值最大值		
S_2+6 、7	目的數值最小值		

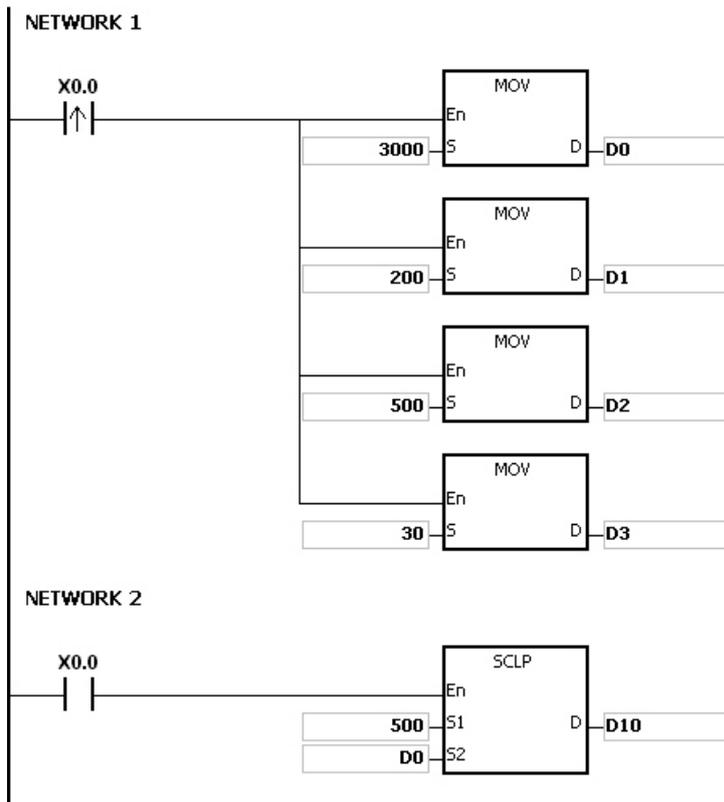
5. 32 位元指令 S_2 運算元將連續佔用 8 個裝置。
6. 旗標 SM685 為十進制整數與單精度浮點數使用旗標。在 32 位元指令中若要使用浮點數做運算可以設定 SM685=ON，若要使用十進制整數做運算可以設定 SM685=OFF。

7. 指令內部運算公式為 $D=[(S_1-\text{數值來源最小值}) \times (\text{目的數值最大值}-\text{目的數值最小值})] \div (\text{數值來源最大值}-\text{數值來源最小值}) + \text{目的數值最小值}$ 。
8. 來源數值和目的數值運算關係：
 $y=kx+b$
 $y=\text{目的數值}(D)$
 $k=\text{斜率}=(\text{目的數值最大值}-\text{最小值}) \div (\text{來源數值最大值}-\text{最小值})$
 $x=\text{來源數值}(S_1)$
 $b=\text{偏移量}=\text{目的數值最小值}-\text{來源數值最小值} \times \text{斜率}$
 將上面的各參數帶入公式 $y=kx+b$ ，即可推導得出指令內部運算公式：
 $y=kx+b=D=kS_1+b=\text{斜率} \times S_1 + \text{偏移量} = \text{斜率} \times S_1 + \text{目的數值最小值}-\text{來源數值最小值} \times \text{斜率} = \text{斜率} \times (S_1-\text{來源數值最小值}) + \text{目的數值最小值} = (S_1-\text{來源數值最小值}) \times (\text{目的數值最大值}-\text{目的數值最小值}) \div (\text{來源數值最大值}-\text{來源數值最小值}) + \text{目的數值最小值}$ 。
9. 假如 $S_1 > \text{數值來源最大值}$ ，則 $S_1 = \text{數值來源最大值}$ ；假如 $S_1 < \text{數值來源最小值}$ ，則 $S_1 = \text{數值來源最小值}$ ；當輸入數值與參數設定完成後，則其輸出曲線將如下圖所示。

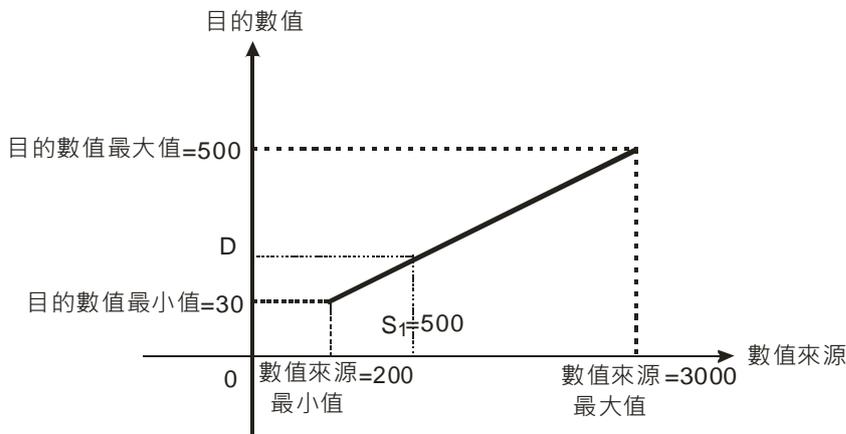


程式範例一：

1. 已知 S_1 數值資料來源為 500，數值來源最大值 $D0=3,000$ ，數值來源最小值 $D1=200$ ，目的數值最大值 $D2=500$ ，目的數值最小值 $D3=30$ 當 $X0.0=ON$ 時，SCLP 指令執行，可在 $D10$ 得到所要求的比例值。
2. 運算方式： $D10=[(500-200) \times (500-30)] \div (3,000-200) + 30=80.35$ ，取整數， $D10=80$ 。

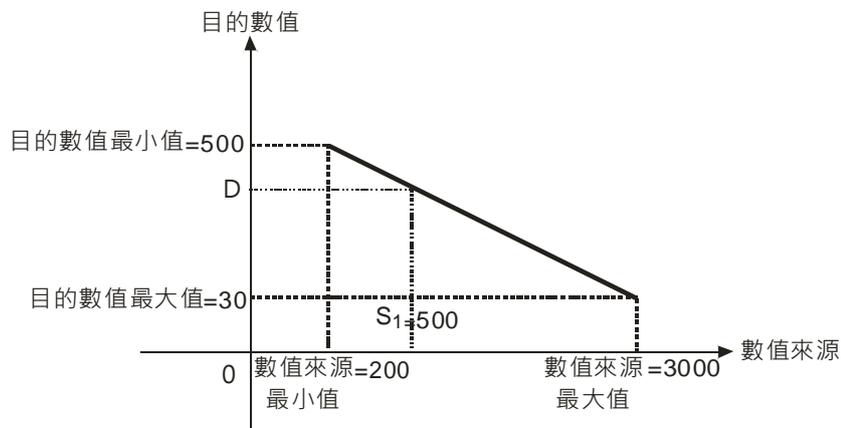
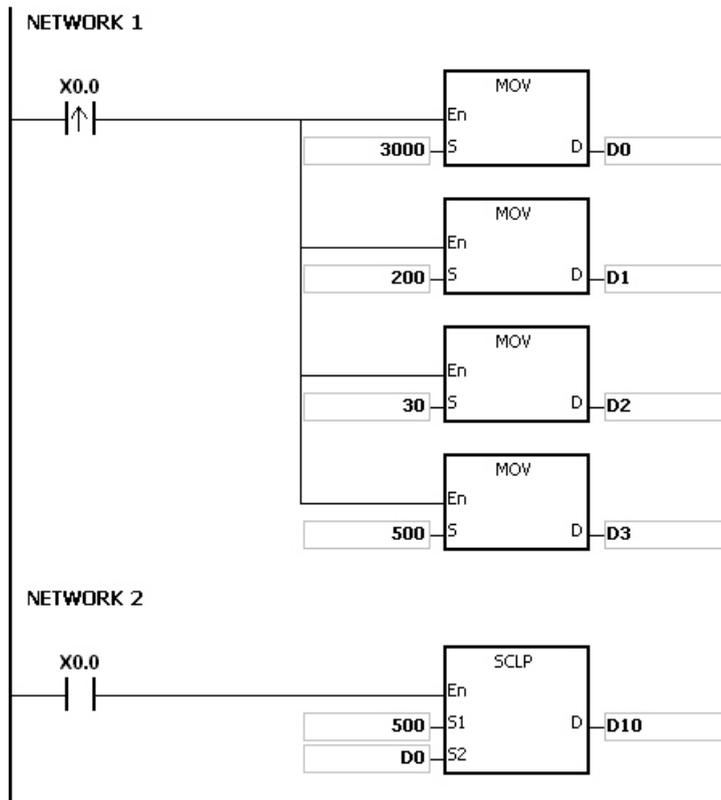


6



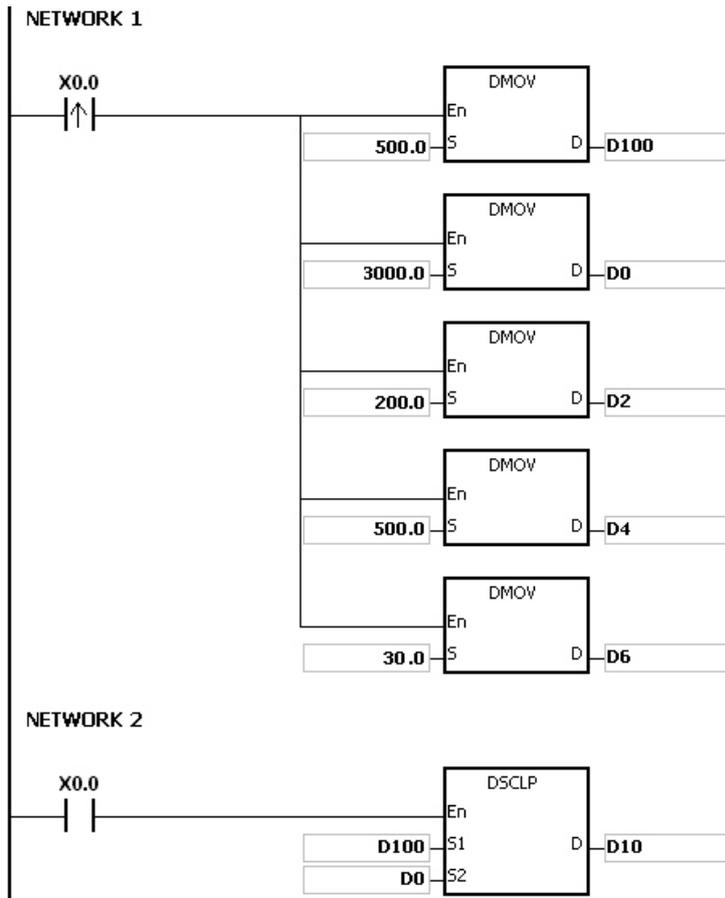
程式範例二：

1. 已知 S_1 數值資料來源為 500，數值來源最大值=D0=3,000，數值來源最小值 D1=200，目的數值最大值 D2=30，目的數值最小值 D3=500 當 X0.0=ON 時，SCLP 指令執行，可在 D10 得到所要求的比例值。
2. 運算方式： $D10 = [(500 - 200) \times (30 - 500)] \div (3,000 - 200) + 500 = 449.64$ 。四捨五入取整數， $D10 = 450$ 。

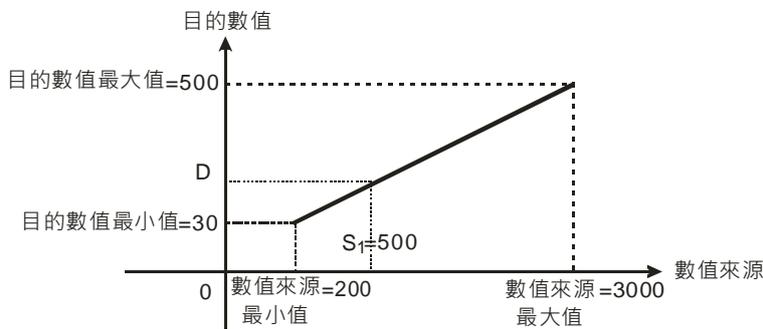


程式範例三：

1. 已知 S_1 數值資料來源 D100 為 500.0，數值來源最大值 D0 為 3000.0，數值來源最小值 D2 為 200.0，目的數值最大值 D4 為 500.0，目的數值最小值 D6 為 30.0 當 X0.0=ON 時，SET SM685，使用浮點數運算且 DSCLP 指令執行，可在 D10 得到所要求的比例值。
2. 運算方式： $D10 = [(500.0 - 200.0) \times (500.0 - 30.0)] \div (3000.0 - 200.0) + 30.0 = 80.35$ ，取整數， $D10 = 80.0$ 。



6



補充說明：

1. 16 位元 S_1 運算元數值設定範圍：數值來源最大值 $\geq S_1 \geq$ 數值來源最小值，-32,768~32,767 超出邊界值以邊界值運算。
2. 32 位元 S_1 整數運算元數值設定範圍：數值來源最大值 $\geq S_1 \geq$ 數值來源最小值，-2,147,483,648~2,147,483,647 超出邊界值以邊界值運算。
3. 32 位元 S_1 浮點數運算元數值設定範圍：數值來源最大值 $\geq S_1 \geq$ 數值來源最小值，依單精度浮點數範圍，超出邊界值以邊界值運算。
4. 使用者運用時，須注意數值來源最大值，必須大於數值來源最小值，而目的數值最大值，並不限制大於目的數值最小值。
5. 數值來源最大值等於數值來源最小值，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2012。

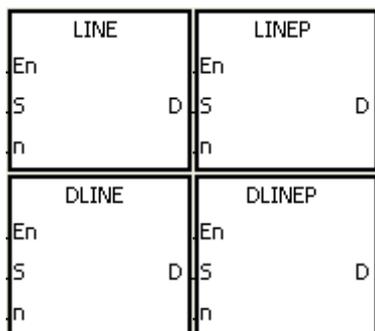
6. 16 位元指令 **S₂** 運算元，若使用 ISPSoft 宣告，則資料型態為 ARRAY [4] of WORD。
7. 32 位元指令 **S₂** 運算元，若使用 ISPSoft 宣告，則資料型態為 ARRAY [4] of DWORD。

API	指令碼			運算元						功能					
0218	D	LINE	P	S · n · D						COLUMN to LINE					

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

脈波執行型	16 位元指令(7 steps)	32 位元指令(7 steps)
AH500	AH500	AH500

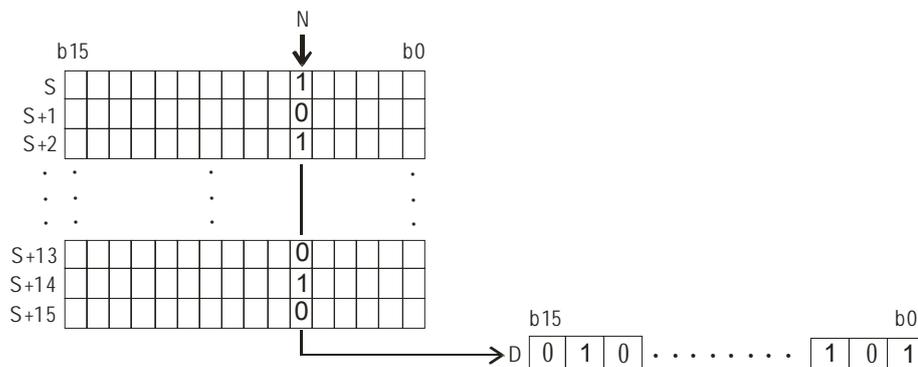
符號：



- S** : Data source Word/Double Word
- n** : Number of Bit Word/Double Word
- D** : Data destination Word/Double Word

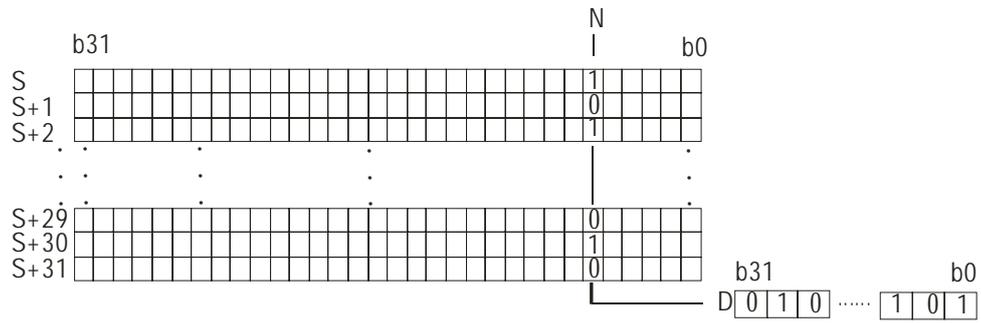
指令說明：

- 16 位元指令 **S** 運算元固定佔用連續 16 筆資料來源 **S~S+15**，其中每一筆為 16 位元暫存器。
- 32 位元指令 **S** 運算元固定佔用連續 32 筆資料來源 **S~S+31**，其中每一筆為 32 位元暫存器。
- n** 運算元表示要擷取 **S** 運算元中每一筆的第 **n** 個 Bit。並在 16 位元指令中限制 **n=0~15**，在 32 位元指令中限制 **n=0~31**。
- 16 位元指令將 16 筆的資料來源 **S~S+15** 的第 **n** 個 Bit，依 **S~S+15** 的順序存於 **D** 運算元的 **b0~b15**。
- 32 位元指令將 32 筆的資料來源 **S~S+31** 的第 **n** 個 Bit，依 **S~S+31** 的順序存於 **D** 運算元的 **b0~b31**。
- 32 位元指令才可使用 **HC** 裝置。
- 以 16 位元指令說明：



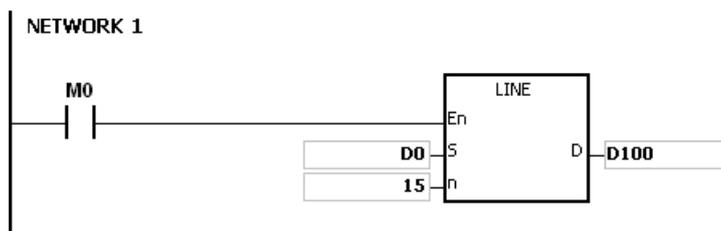
6

8. 以 32 位元指令說明：



程式範例：

M0=ON 時，D0~D14 的第 15 個 Bit 儲存於 D100 的 b0~b15 當中。



補充說明：

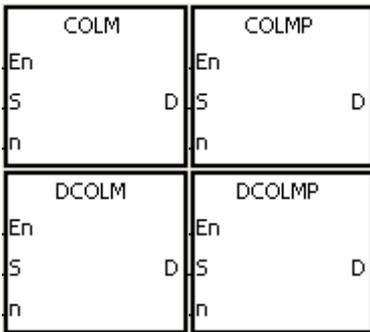
1. 16 位元指令 **S+15** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 32 位元指令 **S+31** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元						功能					
0219	D	COLM	P	S · n · D						LINE to COLUMN					

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

脈波執行型	16 位元指令(7 steps)	32 位元指令(7 steps)
AH500	AH500	AH500

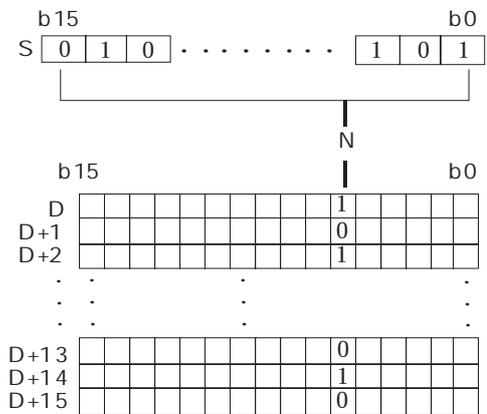
符號：



- S : Data source Word/Double Word
- n : Number of Bit Word/Double Word
- D : Data destination Word/Double Word

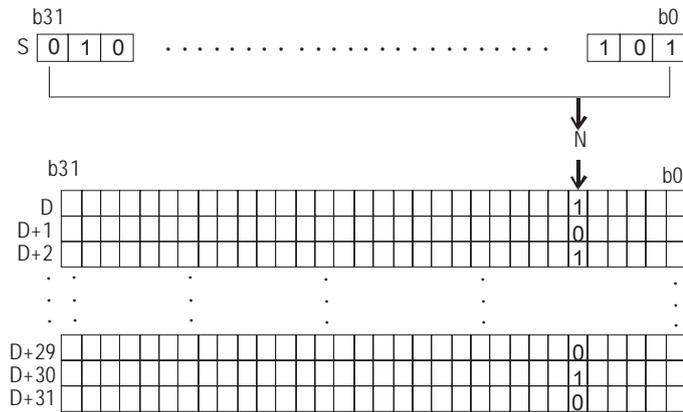
指令說明：

- 16 位元指令 D 運算元固定佔用連續 16 筆資料來源 D~D+15，其中每一筆為 16 位元暫存器。
- 32 位元指令 D 運算元固定佔用連續 32 筆資料來源 D~D+31，其中每一筆為 32 位元暫存器。
- n 運算元表示要填入 D 運算元中每一筆的第 n 個 Bit。並在 16 位元指令中限制 n=0~15，在 32 位元指令中限制 n=0~31。
- 16 位元指令將資料來源 S 依 b0~b15 的順序，依序存於 D~D+15 中每一筆的第 n 個 Bit。
- 32 位元指令將資料來源 S 依 b0~b31 的順序，依序存於 D~D+31 中每一筆的第 n 個 Bit。
- 32 位元指令才可使用 HC 裝置。
- 以 16 位元指令說明：



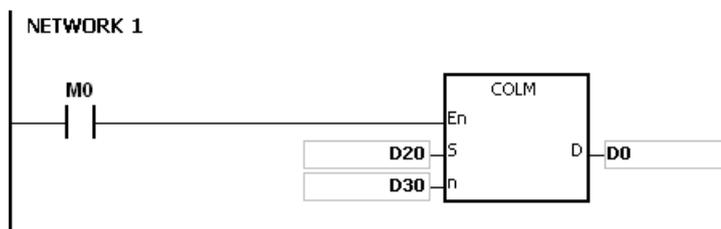
6

8. 以 32 位元指令說明：



程式範例：

當 M0=ON 時，給定 D30=3，依序將 D20 的 b0~b15 存放到 D0~D15 的第 3 個位元。



補充說明：

1. 16 位元指令 **D+15** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 32 位元指令 **D+31** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

6.4 資料轉移指令

6.4.1 資料轉移指令一覽表

API	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
0300	MOV	DMOV	–	✓	資料移動	5
0301	–	–	DFMOV	✓	雙精度浮點數資料移動	5-6
0302	\$MOV	–	–	✓	字串移動	5-11
0303	CML	DCML	–	✓	反轉傳送	5
0304	BMOV	–	–	✓	全部傳送	7
0305	NMOV	DNMOV	–	✓	多點移動	7
0306	XCH	DXCH	–	✓	資料的交換	5
0307	BXCH	–	–	✓	全部交換	7
0308	SWAP	DSWAP	–	✓	上/下 BYTE 變換	3
0309	SMOV	–	–	✓	位數移動	11
0310	MOVB	–	–	✓	多位元移動	7

6

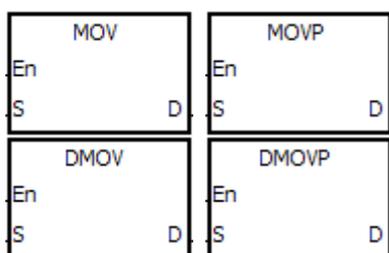
6.4.2 資料轉移指令說明

API	指令碼			運算元								功能					
0300	D	MOV	P	S · D								資料移動					

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：



S : 資料來源 Word/Double Word

D : 資料目的地 Word/Double Word

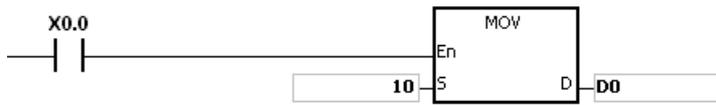
指令說明：

- 當該指令執行時，將 **S** 的內容直接搬移至 **D** 內。當指令不執行時，**D** 內容不會變化。
- 32 位元指令 **S** 才可使用 **F** 浮點數。
- 32 位元指令才可使用 **HC** 裝置。

程式範例：

- 16 位元資料搬移，須使用 **MOV** 指令。
 - 當 X0.0=OFF 時，D0 內容沒有變化，若 X0.0=ON 時，將數值 10 傳送至 D0 資料暫存器內。
 - 當 X0.1=OFF 時，D10 內容沒有變化，若 X0.1=ON 時，將 T0 現在值傳送至 D10 資料暫存器內。
- 32 位元資料搬移，須使用 **DMOV** 指令。
 - 當 X0.2=OFF 時，(D31、D30)、(D41、D40) 內容沒有變化，若 X0.2=ON 時，將 (D21、D20) 現在值傳送至 (D31、D30) 資料暫存器內。同時，將 HC0 現在值傳送至 (D41、D40) 資料暫存器內。
- F 浮點數搬移，須使用 **DMOV** 指令。
 - 當 X0.3=OFF 時，(D51、D50) 內容沒有變化，若 X0.3=ON 時，將浮點數 3.450 轉換為二進浮點值傳送至 (D51、D50) 資料暫存器內。

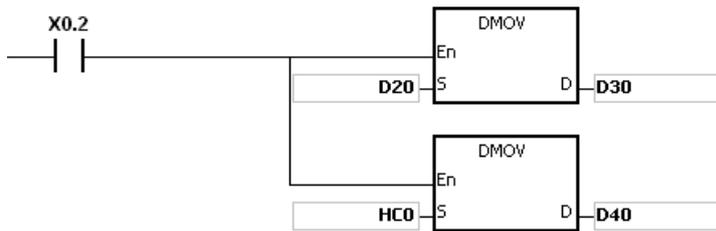
NETWORK 1



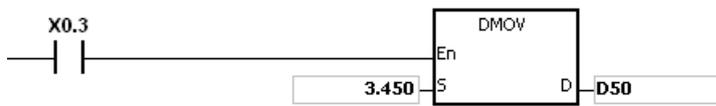
NETWORK 2



NETWORK 3



NETWORK 4

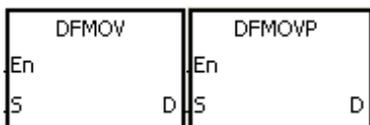


6

API	指令碼			運算元							功能						
0301	D	FMOV	P	S · D							雙精度浮點數資料移動						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				○
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	32 位元指令	64 位元指令 (5-6 steps)
AH500	-	AH500

符號：



S：資料來源

Long Word

D：資料目的地

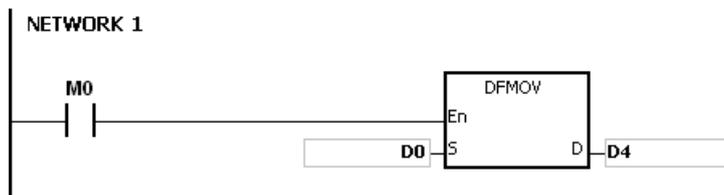
Double Word/Long Word

指令說明：

1. 當該指令執行時，將 **S** 的內容直接搬移至 **D** 內。當指令不執行時，**D** 內容不會變化。
2. 僅支援 64 位元指令。
3. DFMOV、DFMOV P 指令為雙精度搬移指令。

程式範例：

當 M0=ON 時，將 D0~D3 的數值傳送到 D4~D7 當中。

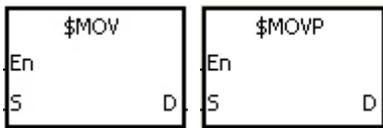


API	指令碼			運算元						功能					
0302		\$MOV	P	S · D						字串移動					

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

脈波執行型	16 位元指令(5-11 steps)	32 位元指令
AH500	AH500	-

符號：



S : 資料來源 String
D : 資料搬移目的地 String

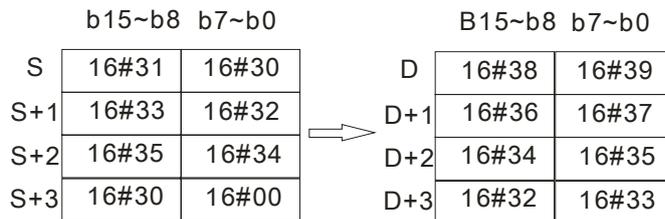
指令說明：

- 若 **S** 運算元為字串時，可以搬移 1~31 個字元；且字串的 STEP 數計算方式為 STEP 數=1+(字串數+1) / 4 (若不能整除則無條件進位)。STEP 數計算表格如下：

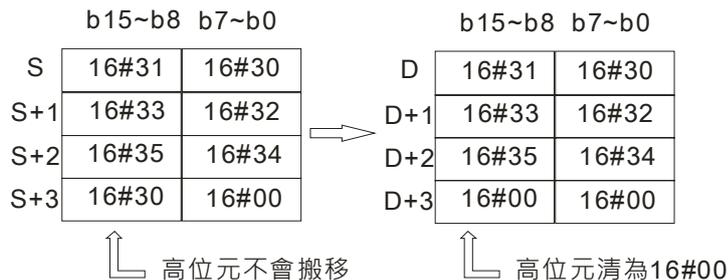
字串數	1~3	4~7	8~11	12~15	16~19	20~23	24~27	28~31
STEP 數	2	3	4	5	6	7	8	9

- 若 **S** 運算元為字串且該指令執行時，將 **S** 運算元的字串直接搬移至 **D** 運算元內且結尾會補上結尾碼 (16#00)。
- 若 **S** 運算元要做搬移的不是字串時，就要在結尾後面有一個結尾碼 (16#00)。
- 若 **S** 運算元不是字串且該指令執行時，將 **S**~16#00 之間的字串直接搬移至 **D** 內 (含結尾碼 (16#00))。當指令不執行時，**D** 內容不會變化。
- 若 **S** 運算元不是字串且該指令執行時，第一個字元就是結尾碼 (16#00)，則還是會搬移結尾碼。
- 當 16#00 字串結尾在低位元時，執行結果如下：

執行前



執行後



7. 當 16#00 字串結尾在高位元時，執行結果如下：

執行前

b15~b8 b7~b0		b15~b8 b7~b0			
S	16#31	16#30	D	16#38	16#39
S+1	16#33	16#32	D+1	16#36	16#37
S+2	16#00	16#34	D+2	16#34	16#35
S+3	16#37	16#36	D+3	16#32	16#33

執行後

b15~b8 b7~b0		b15~b8 b7~b0			
S	16#31	16#30	D	16#31	16#30
S+1	16#33	16#32	D+1	16#33	16#32
S+2	16#00	16#34	D+2	16#00	16#34
S+3	16#37	16#36	D+3	16#32	16#33

8. 當 S 跟 D 裝置重疊且 S<D 時，搬移的順序從結尾 16#00 開始搬移，一直搬到 D，不會發生 S 被覆蓋的情形。

執行前

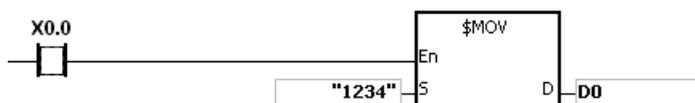
b15~b8 b7~b0		b15~b8 b7~b0			
D0	16#31	16#30	D1	16#33	16#32
D1	16#33	16#32	D2	16#35	16#34
D2	16#35	16#34	D3	16#30	16#00
D3	16#30	16#00	D4	16#38	16#37

執行後

b15~b8 b7~b0		b15~b8 b7~b0			
D0	16#31	16#30	D1	16#31	16#30
D1	16#33	16#32	D2	16#33	16#32
D2	16#35	16#34	D3	16#35	16#34
D3	16#30	16#00	D4	16#00	16#00

程式範例一：

若 S 運算元為偶數字串“1234”時，當條件接點 X0.0 啟動後，D0~DD3 的值如下：



S 運算元

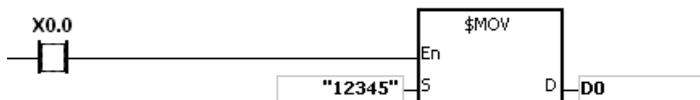
字串	'1'	'2'	'3'	'4'
HEX	16#31	16#32	16#33	16#34

執行後 · D 運算元

裝置	上 8 位元	下 8 位元	註解
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 位元 16#00 為結束字元 · 上 8 位元 16#00 為自動補零
D3	不變	不變	

程式範例二：

若 S 運算元為奇數字串"12345"時 · 當條件接點 X0.0 啟動後 · D0~D3 的值如下：



S 運算元

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

執行後 · D 運算元

裝置	上 8 位元	下 8 位元	註解
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 位元 16#00 為結束字元
D3	不變	不變	

6

程式範例三：

若 S 運算元非字串且結尾字元 16#00 出現在低位元時 · 執行結果如下：



S 運算元

裝置	上 8 位元	下 8 位元	註解
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 為結束字元

執行後 · D 運算元

裝置	上 8 位元	下 8 位元	註解
D0	16#31	16#30	'1'=16#31 · '0'=16#30
D1	16#33	16#32	'3'=16#33 · '2'=16#32

裝置	上 8 位元	下 8 位元	註解
D2	16#35	16#34	'5'=16#35 · '4'=16#34
D3	16#00	16#00	下 8 位元 16#00 為結束字元 · 上 8 位元 16#00 為自動補零
D4	不變	不變	

程式範例四：

若 **S** 運算元非字串且結尾字元 16#00 出現在高位元時，執行結果如下：



S 運算元

裝置	上 8 位元	下 8 位元	註解
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 為結束字元 · '4'=16#34
D103	16#37	16#36	'7'=16#37 · '6'=16#36

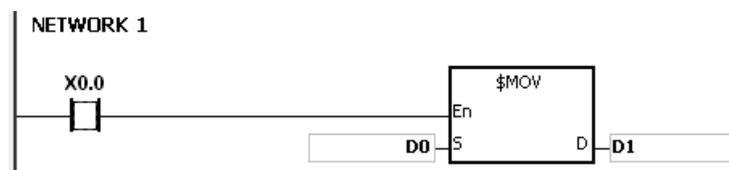
執行後 · **D** 運算元

裝置	上 8 位元	下 8 位元	註解
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 為結束字元 · '4'=16#34
D3	不變	不變	

6

程式範例五：

當 **S** 跟 **D** 裝置重疊且 **S**<**D** 時，搬移的順序從結尾 16#00 開始搬移，一直搬到 **D**，不會發生 **S** 被覆蓋的情形。



S 運算元

裝置	上 8 位元	下 8 位元	註解
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 為結束字元
D4	16#38	16#37	'8'=16#38 · '7'=16#37

執行後，**D** 運算元

裝置	上 8 位元	下 8 位元	註解
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 位元 16#00 為結束字元， 上 8 位元 16#00 為自動補零
D5	不變	不變	

補充說明：

1. 當 **S** 中沒有 16#00 當結尾時，指令不執行。SM0=ON，錯誤碼 SR0=16#200E。
2. 當 **D** 裝置空間不足以存放 **S** 的字串時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

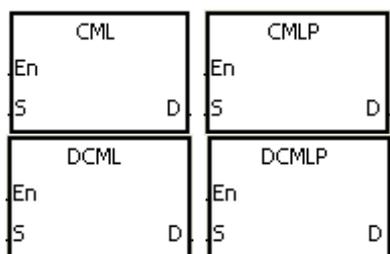
6

API	指令碼			運算元								功能					
0303	D	CML	P	S · D								反轉傳送					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：



S : 資料來源 Word/Double Word

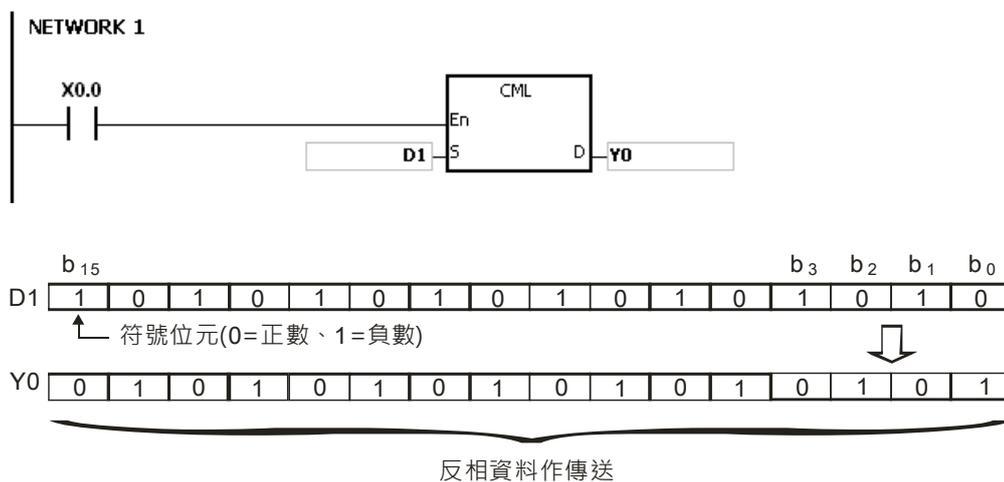
D : 資料搬移目的地 Word/Double Word

指令說明：

- 希望作反相輸出時，使用本指令。將 **S** 的內容全部反相 (0→1、1→0) 傳送至 **D** 當中。如果內容為常數時，此常數自動被轉換成 BIN 值。
- 32 位元指令才可使用 **HC** 裝置。

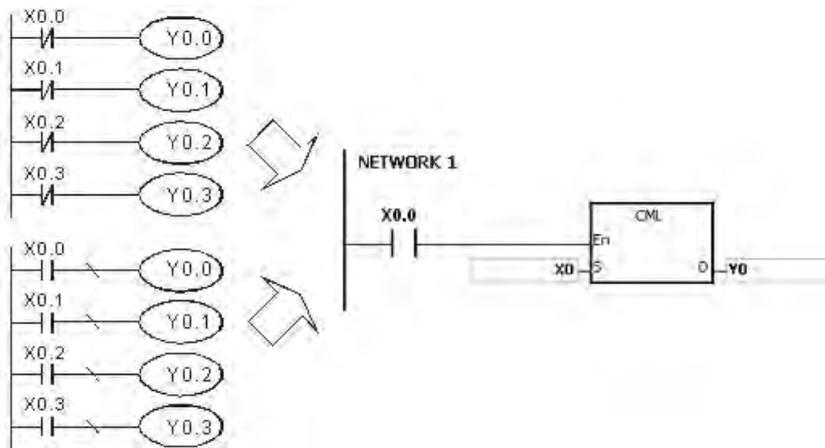
程式範例一：

當 X0.0=ON 時，將 D1 之 b0~b15 內容反相後傳送到 Y0.0~Y0.15。



程式範例二：

下圖左邊的回路也可以使用 CML 指令來表現，如下圖所示

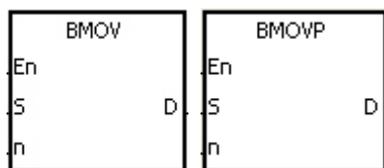


API	指令碼		運算元								功能					
0304	BMOV	P	S · D · n								全部傳送					

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：

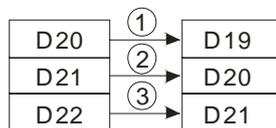


S : 資料來源 Word
D : 資料搬移目的地 Word
n : 傳送長度 Word

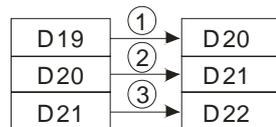
指令說明：

1. 所指定的裝置起始號碼開始算 **n** 個暫存器的內容被傳送至 **D** 所指定的裝置起始號碼開始算 **n** 個暫存器當中。
2. **n=1~256**。
3. 為了防止兩個運算元所指定傳送的號碼重疊時，所造成的混亂，請注意兩個運算元所指定號碼大小的安排，如下所示：

當 **S>D** 時，以 ①→②→③ 的順序傳送。

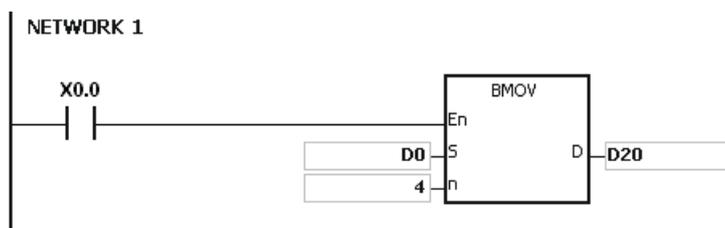


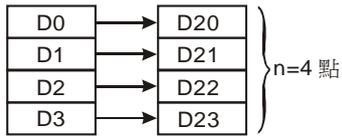
當 **S<D** 時，是以 ③→②→① 的順序傳送。



程式範例一：

當 X0.0=ON 時，D0~D3 的 4 個暫存器的內容被傳送至 D20~D23 的 4 個暫存器內。

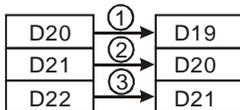
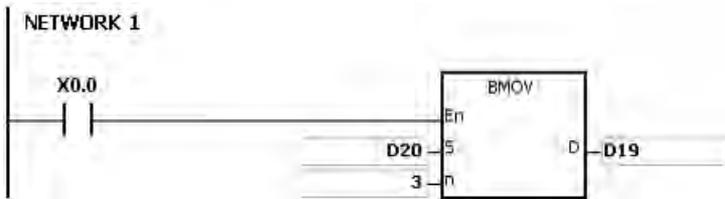




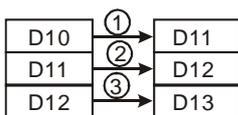
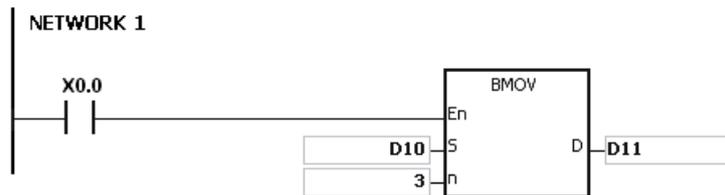
程式範例二：

為了防止兩個運算元所指定傳送的號碼重疊時，所造成的混亂，請注意兩個運算元所指定號碼大小的安排，如下所示：

1. 當 **S>D** 時，以①→②→③的順序傳送



2. 當 **S<D** 時，其執行結果以③→②→①的順序傳送。



補充說明：

1. 當 **D+n-1** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 當 **S+n-1** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 當 **n>256** 或 **n<1** 時，指令不會執行，SM0=ON，錯誤碼 SR0=16#200B。

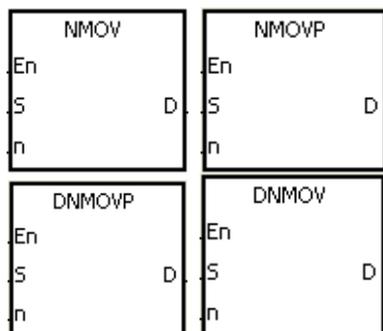
6

API	指令碼			運算元							功能						
0305	D	NMOV	P	S · D · n							多點移動						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

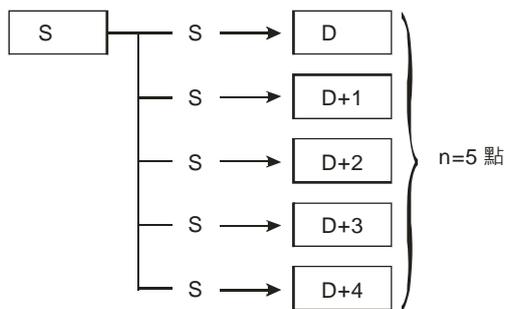
符號：



S : 資料來源 Word/Double Word
D : 資料搬移目的地 Word/Double Word
n : 傳送長度 Word/Double Word

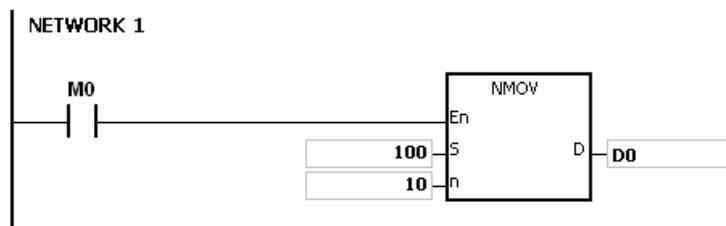
指令說明：

- 當該指令執行時，將 **S** 的內容直接搬移至 **D** 開始算起的 **n** 個裝置中，當指令不執行時，**D** 內容不會變化。
- HC 裝置只可在 32 位元指令時使用。
- NMOV 時 n=1~256，DNMOV 時 n=1~128。



程式範例：

M0 為 ON 時，D0~D9 的值為 100。



補充說明：

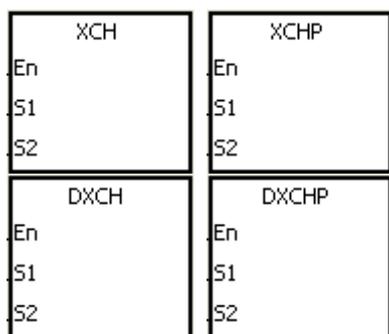
1. 當 $D \sim D+n-1$ 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 16 位元指令中，當 $n > 256$ 或 $n < 1$ 時，指令不會執行，SM0=ON，錯誤碼 SR0=16#200B。
3. 32 位元指令中，當 $n > 128$ 或 $n < 1$ 時，指令不會執行，SM0=ON，錯誤碼 SR0=16#200B。

6

API	指令碼			運算元								功能					
0306	D	XCH	P	$S_1 \cdot S_2$								資料的交換					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●	●	●	●		●	○	●				
S_2	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：



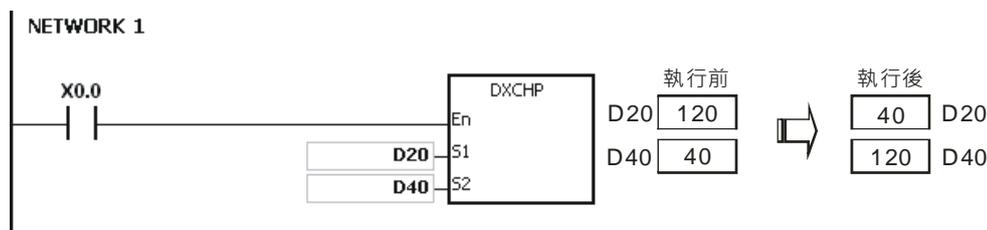
S_1 : 要交換的資料 1 Word/Double Word
 S_2 : 要交換的資料 2 Word/Double Word

指令說明：

- S_1 及 S_2 所指定之裝置內容值互相交換。
- 32 位元指令才可以使用 HC 裝置。

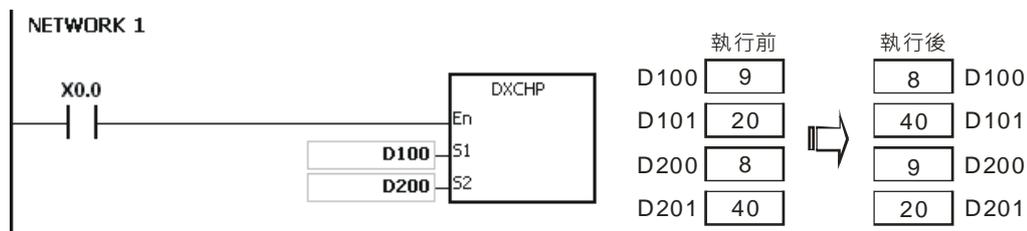
程式範例一：

X0.0=OFF→ON 時，D20 與 D40 的內容互相交換。



程式範例二：

X0.0=OFF→ON 時，D100 與 D200 的內容互相交換。

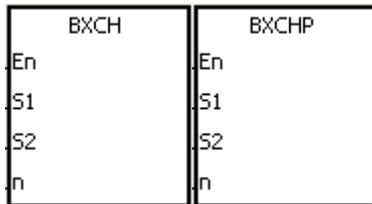


API	指令碼		運算元					功能					
0307		BXCH	P	$S_1 \cdot S_2 \cdot n$					全部交換				

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

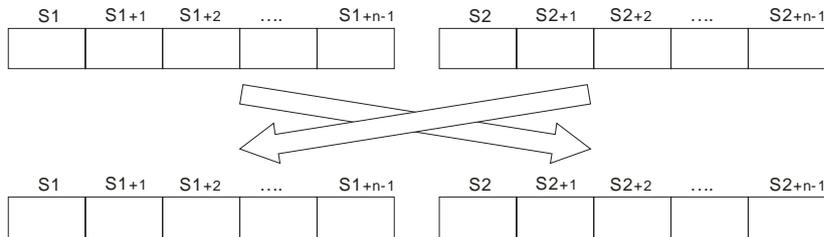
符號：



S_1 : 要交換的資料 1 Word/Double Word
 S_2 : 要交換的資料 2 Word/Double Word
n : 長度 Word/Double Word

指令說明：

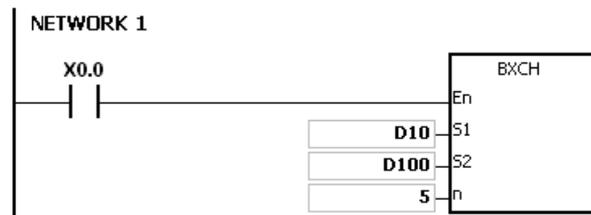
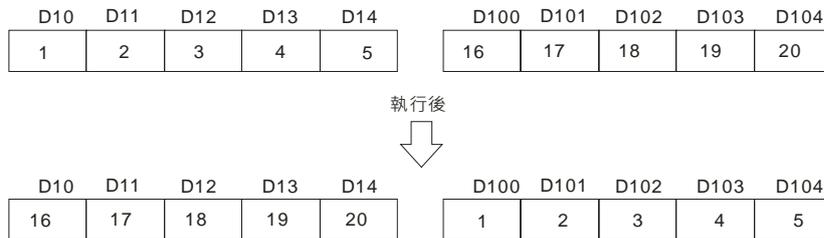
- $S_1 \sim S_1+n-1$ 及 $S_2 \sim S_2+n-1$ 所指定之裝置內容值互相交換。
- $n=1 \sim 256$ 。



6

程式範例：

X0.0=ON 時 · D10~D14 的內容與 D100~D104 的內容互相交換。

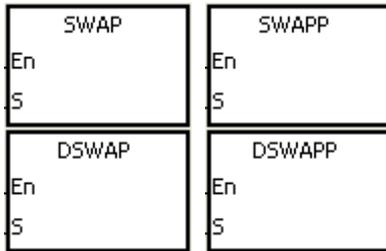


補充說明：

1. 當 S_1+n-1 的範圍超出 S_1 裝置時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. 當 S_2+n-1 的範圍超出 S_2 裝置時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
3. 當 $n>256$ 或 $n<1$ 時，指令不會執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。

API	指令碼			運算元								功能					
0308	D	SWAP	P	S								上/下 BYTE 變換					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●				
脈波執行型											16 位元指令(3 steps)			32 位元指令(3 steps)			
AH500											AH500			AH500			

符號：



S : 資料來源

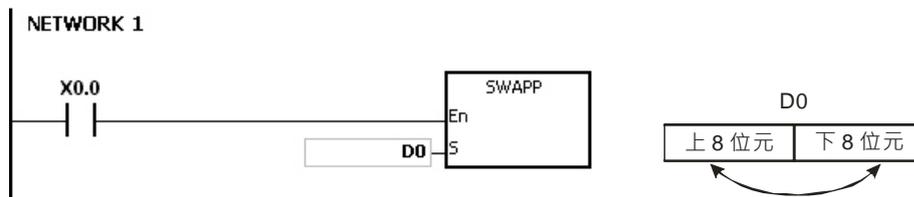
Word/Double Word

指令說明：

- 16 位元指令，將 S 的上下 8 位元的內容互相交換。
- 32 位元指令將 S 的上 16 位元的上下 8 位元內容互相交換，及下 16 位元的上下 8 位元內容互相交換。
- 32 位元指令才可以使用 HC 裝置。

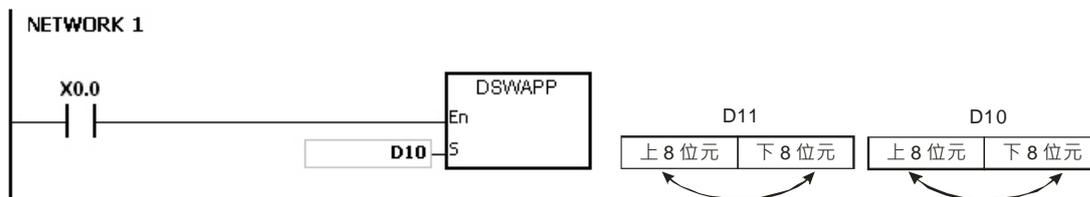
程式範例一：

當 X0.0=ON 時，將 D0 之上位 8 位元與下位 8 位元的內容互相交換。



程式範例二：

當 X0.0=ON 時，將 D11 之上位 8 位元與下位 8 位元的內容互相交換，D10 之上位 8 位元與下位 8 位元的內容互相交換。



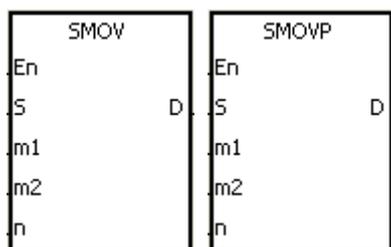
6

API	指令碼			運算元								功能					
0309		SMOV	P	S · m₁ · m₂ · D · n								位數移動					

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

脈波執行型	16 位元指令 (11 steps)	32 位元指令
AH500	AH500	-

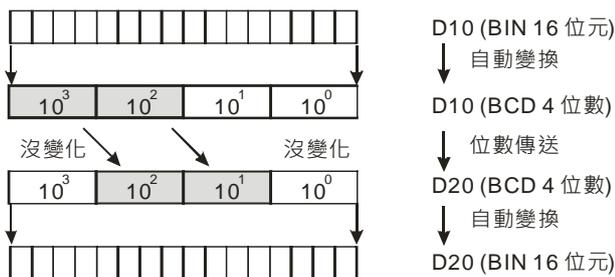
符號：



- S** : 要移動的資料來源 Word
- m₁** : 資料來源要傳送的起始位數 Word
- m₂** : 資料來源要傳送的位數之個數 Word
- D** : 傳送之目的地裝置 Word
- n** : 目的地裝置要存放資料來源的起始位數 Word

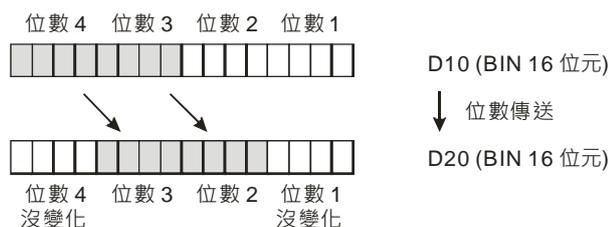
指令說明：

- 此指令可將資料重新分配或合成。當該指令執行時，指定 **S** 的第 **m₁** 位數開始往低位計算的 **m₂** 位數內容傳送至 **D** 的第 **n** 位數開始往低位計算的 **m₂** 位數中。
- m₁**=1~4 · **m₂**=1~**m₁** · **n**=**m₂**~4。(4 個 bit 為一個單位)
- SM605=OFF 時，為 BCD 模式，指令會自動將 **S** 的內容當成 BCD 的模式來執行。



若 S=K1234 · D=K5678 · 則執行後 S=1234 · D=5128

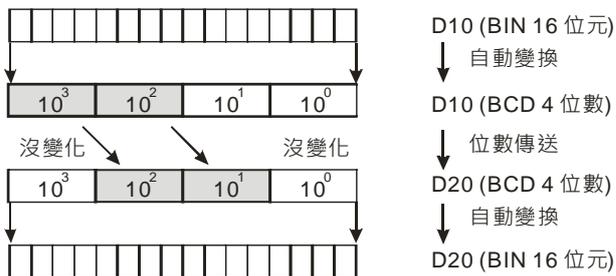
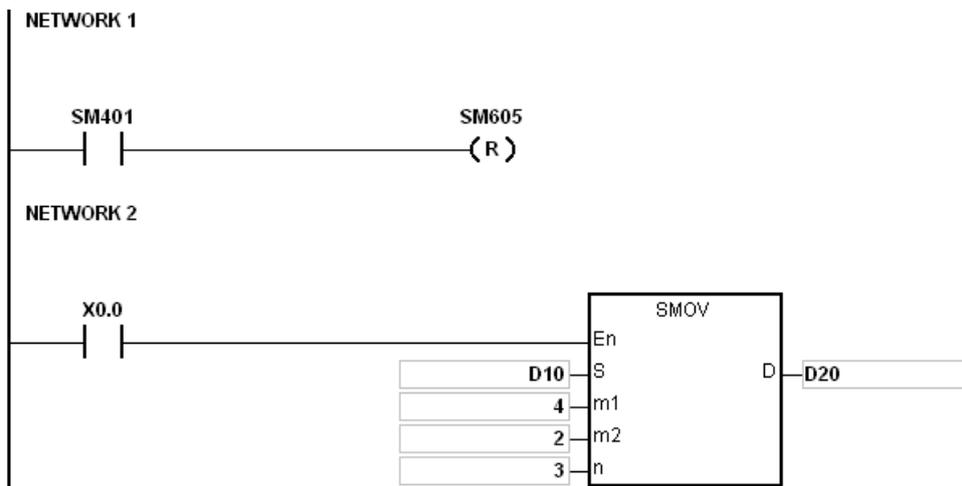
- SM605=ON 時，為 BIN 模式。



若 S=16#1234 · D=16#5678 · 則執行後 S=16#1234 · D=16#5128。

程式範例一：

1. 當 SM605=OFF 時 (BCD 模式) · X0.0=ON · 指定 D10 的 10 進制數值的第 4 位數 (亦即千位數) 開始往低位計算的 2 位數內容傳送至 D20 的 10 進制數值的第 3 位數 (亦即百位數) 開始往低位計算的 2 位數中。而 D20 的 10^3 及 10^0 於本指令被執行後內容沒有變化。
2. 當 BCD 值超過 0~9,999 的範圍時 · PLC 判定為運算錯誤 · 指令不執行 · SM0=ON · 錯誤碼 SR0=16#200D。

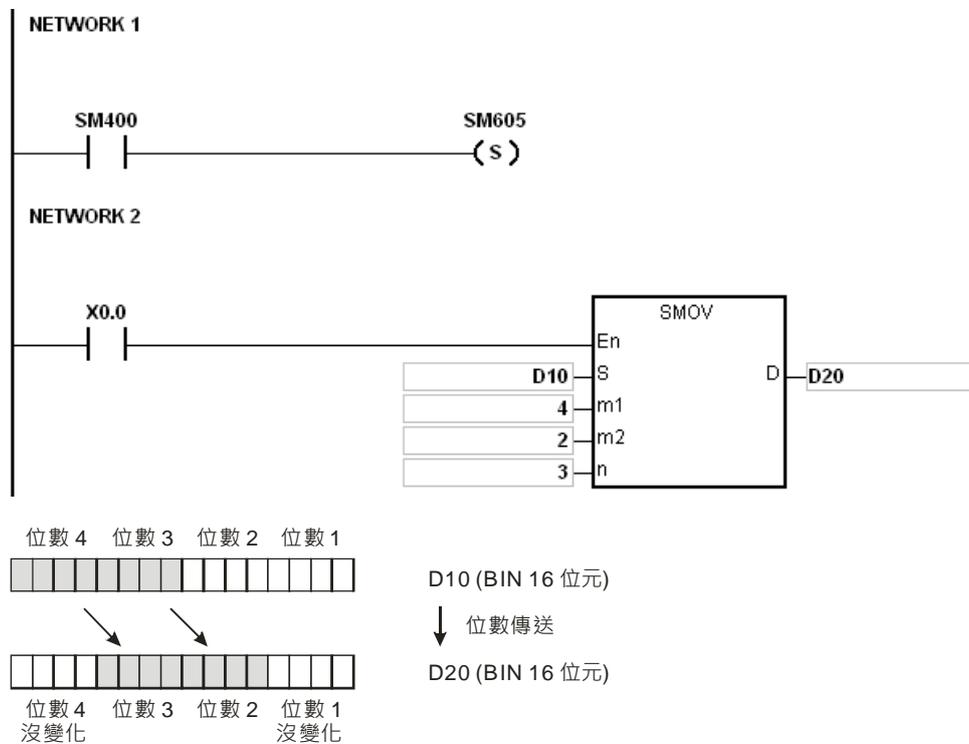


若執行前 D10=1234 · D20=5678 · 執行完畢後 · D10 不變 · D20=5128。

6

程式範例二：

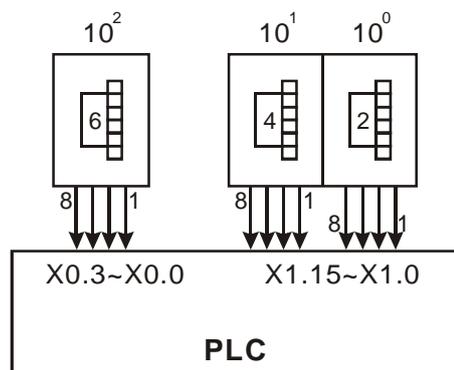
當 SM605=ON (BIN 模式) 時，使用 SMOV 指令的話，D10、D20 並不會作 BCD 變換，而是以 BIN 型態 4 個位元為一個單位作傳送。

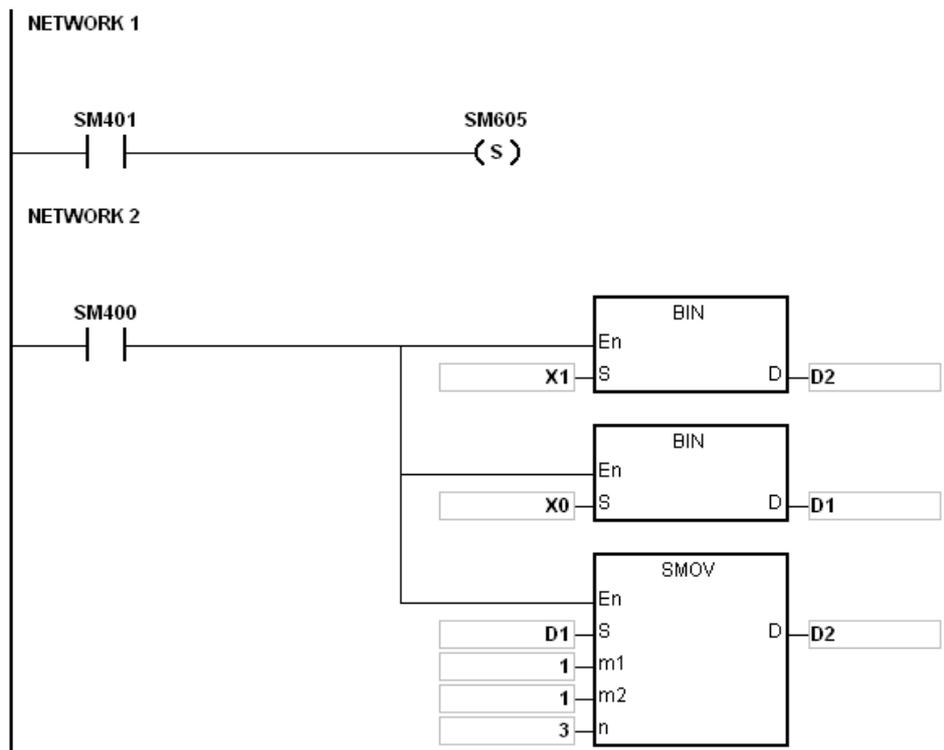


若執行前 D10=16#1234，D20=16#5678，執行完畢後，D10 不變，D20=16#5128。

程式範例三：

1. 連接於非連續編號輸入端的指撥開關可使用本指令來合成。
2. 將右 2 位指撥開關傳送至 D2 的右 2 位，左 1 位指撥開關傳送至 D1 的右 1 位數當中。
3. 使用 SMOV 指令將 D1 的第 1 位傳送至 D2 的第 3 位數將兩組指撥開關合成 1 組。





補充說明：

1. 當指令為 BCD 模式時，如果 $S < 0$ 或 $S > 9999$ 或 $D < 0$ 或 $D > 9999$ ，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200D$ 。
2. $m_1 < 1$ 或 $m_1 > 4$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
3. 當 $m_2 < 1$ 或 $m_2 > m_1$ ，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
4. 當 $n < m_2$ 或 $n > 4$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。

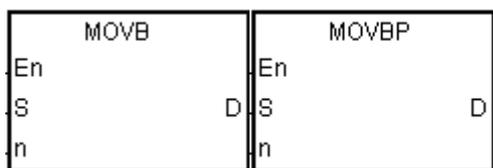
6

API	指令碼			運算元								功能					
0310		MOVB	P	S · n · D								多位元移動					

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：



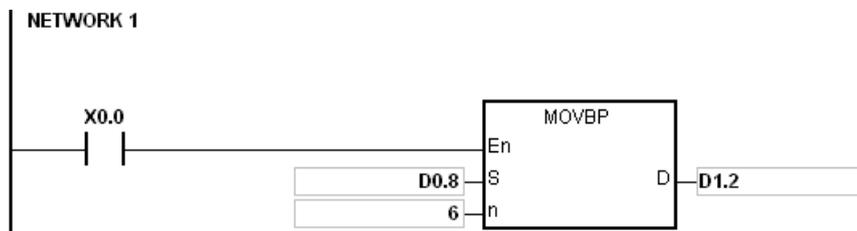
S : 要移動的資料來源 Bit
n : 傳送筆數 Word
D : 目的地裝置要存放資料來源的起始位數 Bit

指令說明：

1. 當該指令執行時，指定 **S** 起始位數內容傳送至 **D** 開始的 **n** 個位數中。
2. **S**、**D** 為 T、C、HC 裝置時，只傳送裝置的狀態，不會傳送裝置的現在值。
3. **n**=1~256，**n**<1 或 **n**>256 時指令不執行，SM0=ON，Error Code=16#200B。

程式範例：

X0.0 為 ON 時，PLC RUN 後，D0.8~D0.13 之內容傳送至 D1.2~D1.7。



補充說明：

1. 當 **D+n-1** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 當 **S+n-1** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

6.5 程式跳躍指令

6.5.1 程式跳躍指令一覽表

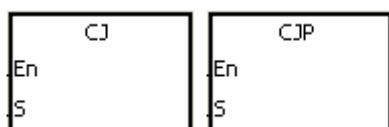
API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>0400</u>	CJ	–	✓	條件跳躍	3
<u>0401</u>	JMP	–	–	無條件跳躍	3
<u>0402</u>	GOEND	–	–	跳躍到 END	1

6.5.2 程式跳躍指令說明

API	指令碼			運算元								功能					
0400		CJ	P	S								條件跳躍					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S																	

脈波執行型	16 位元指令 (3 steps)	32 位元指令
AH500	AH500	-

符號：



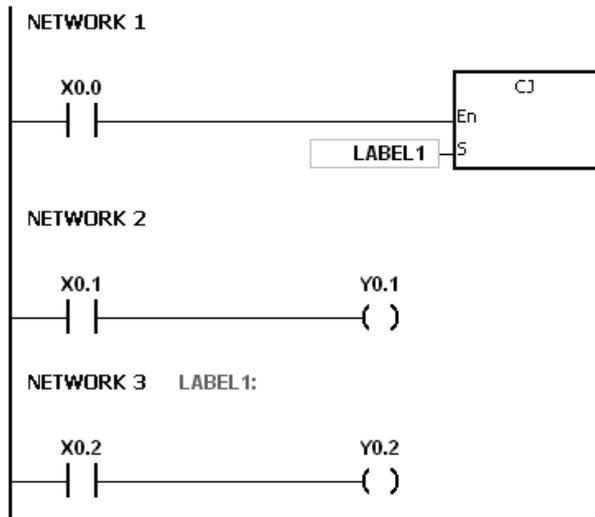
S：條件跳躍之目的指標

指令說明：

- 當使用者希望 PLC 程式中的某一部份不需要執行時，以縮短掃描時間，以及使用於雙重輸出時，可使用 CJ 或 CJP 指令。
- 指標 P 所指之程式若在 CJ 指令之前，需注意會發生 WDT 逾時之錯誤，PLC 停止運轉，請注意使用。
- CJ 指令可重複指定同一指標 P。
- 跳躍執行中各種裝置動作情形說明：
 - Y、M、S 保持跳躍發生前之狀態。
 - 執行計時中計時器會暫停計時。
 - 一般應用指令不會被執行。

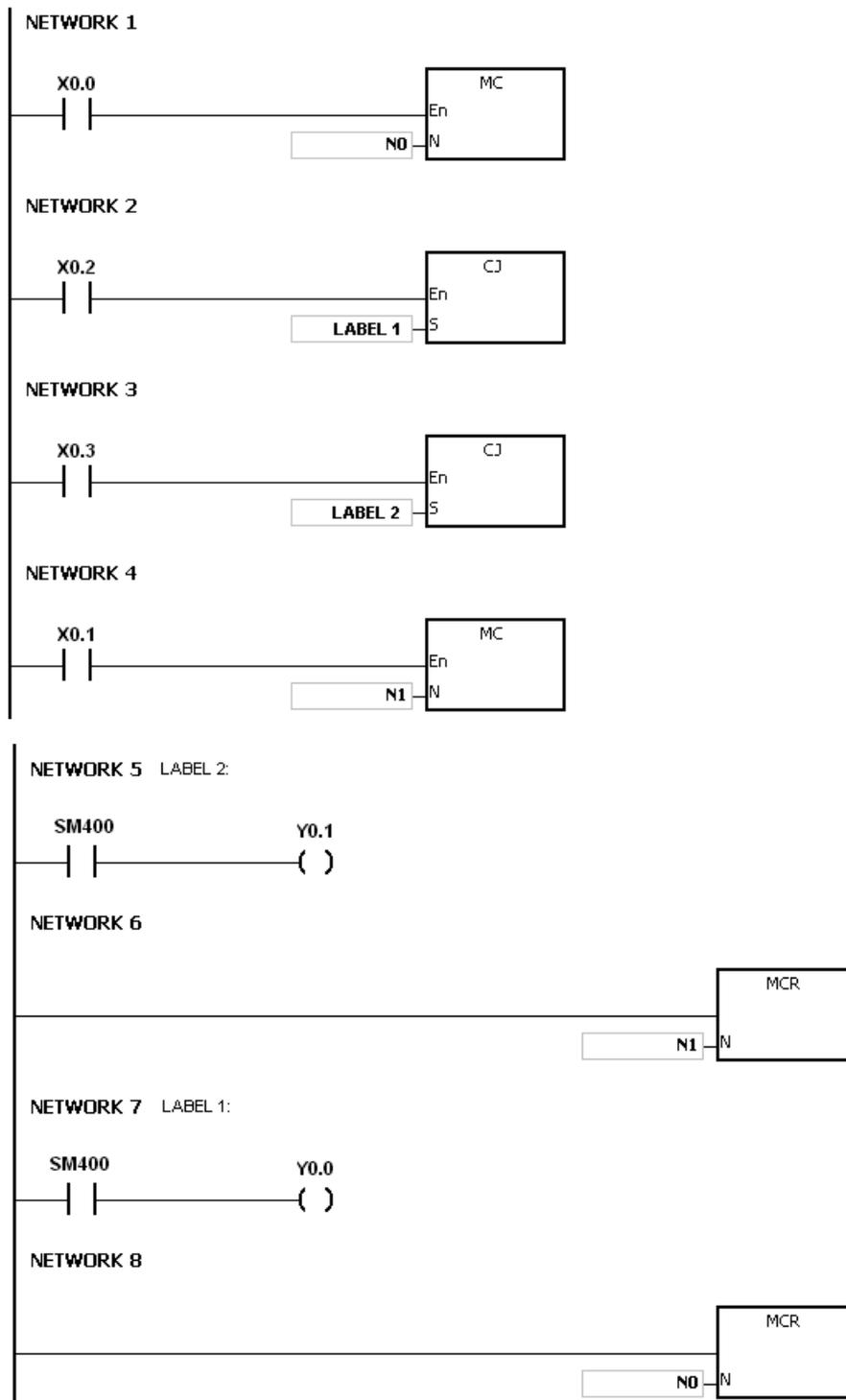
程式範例一：

- 當 X0.0=ON 時，程式自動從位址 0 跳躍至位址 N (即指定之標籤 LABEL1:) 繼續執行，中間位址跳過不執行。
- 當 X0.0=OFF 時，程式如同一般程式由位址 0 繼續往下執行，此時 CJ 指令不被執行。



程式範例二：

1. CJ 指令在 MC、MCR 指令間可使用在下列五種狀況：
 - (a) 在 MC~MCR 外。
 - (b) 在 MC 外至 MC 內，如下圖 LABEL1:以下迴路有效。
 - (c) 同一 N 層 MC 內至 MC 內。
 - (d) 在 MC 內至 MCR 外。
 - (e) 自 MC~MCR 內跳至另一 MC~MCR 內。
2. 執行 MC 指令時，PLC 會將之前開關接點的狀態推入 PLC 內部自訂的堆疊中，而此堆疊由 PLC 自行控制，使用者無法改變；而後當執行到 MCR 指令時，會由堆疊的最上層取出之前的開關接點狀態，當上面(b)、(d)、(e)的狀況下時，則有可能會發生推入 PLC 內部堆疊與取出堆疊的次數不相同的情況，遇到這種狀況時，堆疊最多能堆入 32 層，而另外取出堆疊的值最多取到堆疊為空時則不再取出，所以在搭配 CJ 等跳躍指令時須注意堆疊的堆入和取出。



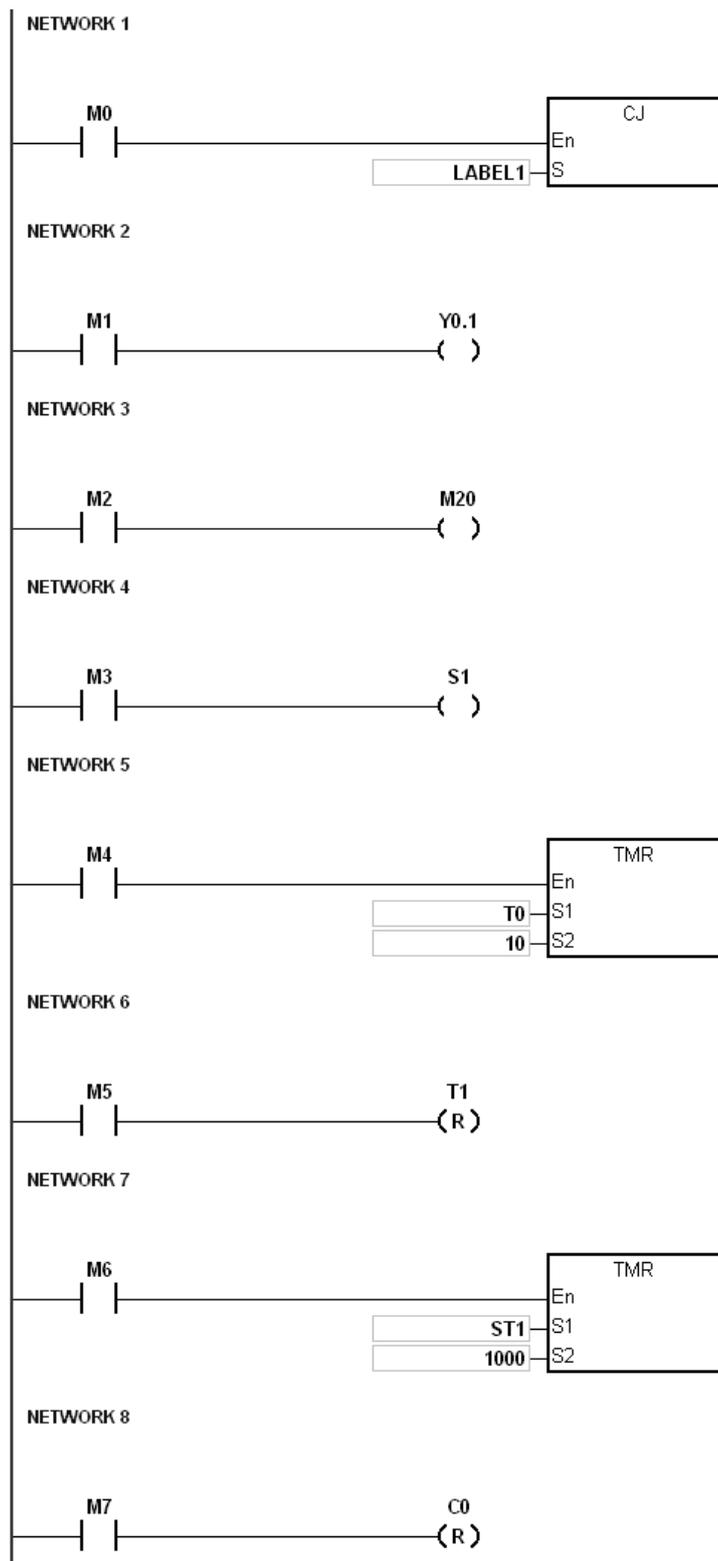
6

程式範例三：

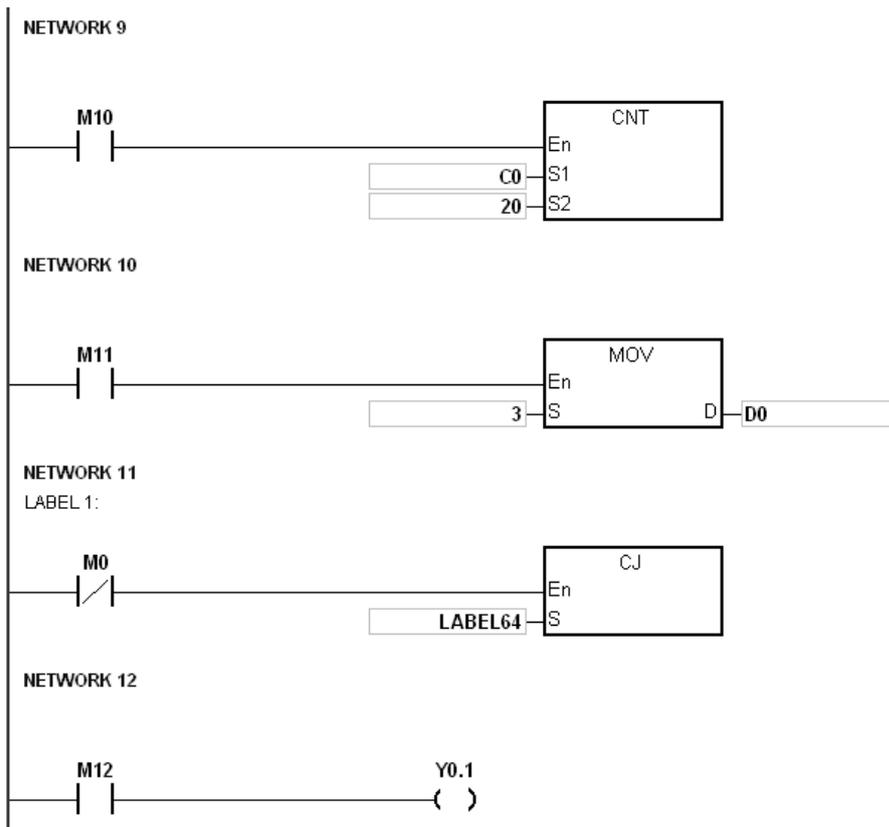
底下表格為下列程式中，各個裝置狀態變化。

裝置	CJ 執行前 接點狀態	CJ 執行中 接點狀態	CJ 執行中輸出線圈狀態
Y、M、S	M1、M2、M3 OFF	M1、M2、M3 由 OFF→ON	Y0.1 ^{註1} 、M20、S1 OFF
	M1、M2、M3 ON	M1、M2、M3 由 ON→OFF	Y0.1 ^{註1} 、M20、S1 ON
計時器	M4 OFF	M4 由 OFF→ON	計時器不作計時動作
	M4 ON	M4 由 ON→OFF	計時器立即停止計時並保持，M0 由 ON 到 OFF，計時器清零。
積算型計時器	M6 OFF	M6 OFF→ON	計時器 ST1 不作計時動作
	M6 ON	M6 ON→OFF	積算型計時器一旦計時動作被啟動，若遇到 CJ 指令時，則計時動作停止但保持。 M0 由 ON→OFF，ST1 仍保持
計數器	M7, M10 OFF	M10 ON/OFF 觸發	計數器不計數
	M7 OFF, M10 ON/OFF 觸發	M10 ON/OFF 觸發	計數器 C0 停止計數並保持，M0 OFF 後，C0 繼續計數
應用指令	M11 OFF	M11 OFF→ON	應用指令不執行
	M11 ON	M11 ON→OFF	被跳過的應用指令不執行

註 1：Y0.1 為雙重輸出，M0 為 OFF 時，由 M1 控制，M0 為 ON 時，由 M12 控制。



6



補充說明：

指標 P 在 ISPSOft 中的使用方式，請參考 ISPSOft 的使用手冊說明。

6

API	指令碼			運算元								功能							
0401		JMP		S								無條件跳躍							
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF		
S																			
											脈波執行型			16 位元指令(3 steps)			32 位元指令		
											-			AH500			-		

符號：



S：跳躍之目的指標

指令說明：

1. 無條件跳躍到程式中的某個 P 指標。
2. 指標 P 所指之程式若在 JMP 指令之前，需注意會發生 WDT 逾時之錯誤，PLC 停止運轉，請注意使用。
3. 跳躍執行中各種裝置動作情形說明：
 - Y、M、S 保持跳躍發生前之狀態。
 - 執行計時中計時器會暫停計時。
 - 計時器之清除指令若在跳躍前被驅動，則在跳躍執行中該裝置仍處於清除狀態。
 - 一般應用指令不會被執行。

API	指令碼	運算元	功能
0402	GOEND	—	跳躍到 END
		脈波執行型	16 位元指令 (1 step)
		-	AH500
			32 位元指令
			-

符號：



指令說明：

1. 條件成立時跳躍到程式中的 END。
2. 功能塊跟中斷工作 (TASK) 與 FOR NEXT 中不支援此指令。
3. GOEND 執行時所略過的指令皆不動作，所有裝置元件內容值與狀態保持不變。

6.6 程式執行控制指令

6.6.1 程式執行控制指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>0500</u>	DI	–	–	中斷插入禁能	1
<u>0501</u>	EI	–	–	中斷插入致能	1
<u>0502</u>	IMASK	–	–	中斷控制	3

6.6.2 程式執行控制指令說明

API	指令碼		運算元	功能
0500		DI	-	中斷插入禁能

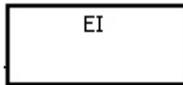
脈波執行型	16 位元指令 (1 step)	32 位元指令
-	AH500	-

符號：

DI

API	指令碼	運算元	功能
0501	EI	-	中斷插入致能
		脈波執行型	16 位元指令 (1 step)
		-	AH500
			32 位元指令
			-

符號：



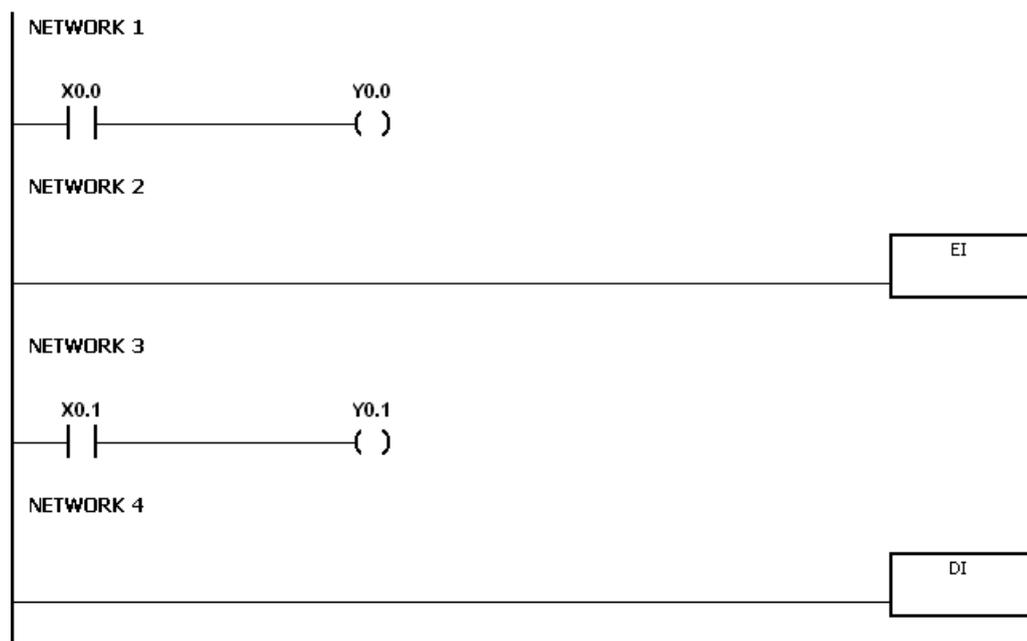
指令說明：

1. EI 表示程式中允許使用中斷工作 (TASK) (TASK I0~TASK I255，請參考 AH500 操作手冊第 6.6 節)。
2. 程式中在 EI 指令到 DI 指令間允許使用中斷工作 (TASK)，在程式中若無中斷插入禁能之區間時，則可以不使用 DI 指令。
3. 中斷工作 (TASK) 執行中，不會再執行其它產生的中斷 (但會記住有中斷產生)，會先執行完目前的中斷工作 (TASK)，才去執行下一個中斷工作 (TASK)。
4. 當多數中斷發生時，以執行者優先，同時發生以指標編號較小者優先。
5. 在 DI~EI 指令之間發生的中斷要求無法立即執行，此要求會被記憶一次，並在中斷許可範圍內時，才去執行中斷程式。
6. 當中斷處理中要即時 I/O 動作時，可在程式中寫入 REF 指令更新 I/O 狀態。

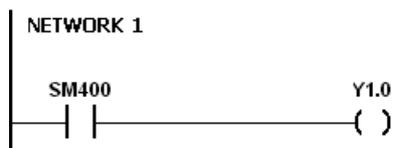
程式範例：

PLC 執行時，當 Cyclic_0 程式掃描到 EI 指令到 DI 指令間，如果有中斷工作 (TASK) 被啟動則會執行中斷工作 (TASK)，而當中斷工作 (TASK) 執行完畢後，則返回主程式並繼續往下執行。

Cyclic_0 程式：



中斷工作 (TASK) :



補充說明 :

中斷工作 (TASK) 0~255 共 256 個。

1. I/O 中斷 (I0~I31)

特殊高速模組使用，模組經由 HWCONFIG 設定好中斷條件以及中斷編號，並經由 ISPSOft 下載中斷程式到 PLC，PLC 運轉時，當模組設定的中斷條件成立，就會執行所對應的中斷程式。

2. 通訊中斷 (I32 · I33)

當 RS 指令，特定字元通訊接收中斷請求，也可當一般中斷使用。詳細說明請參考 API1800 RS 指令。

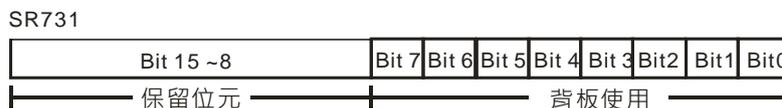
COM1 : I32。

COM2 : I33。

3. 24V 低電壓偵測中斷

能藉由 AHPS05 上的 LV2 (VS+/VS-) 偵測點，偵測外部 24V 的電壓是否正常，當外部 24V 不正常時，讓使用者可藉由中斷副程式 (I34) 執行相對應的流程。

說明：每一塊背板發生低電壓時，將 SR731 裡相對應的 bit 設定為 ON，低電壓恢復後該 bit 會設定為 OFF。SR731 的 Bit8~15 為保留位元。



例：

(a) 主背板發生 24V 低電壓時，SR731 的第 0 個 bit 會被設為 ON。

(b) 第一塊延伸背板發生 24V 低電壓時，SR731 的第 1 個 bit 會被設為 ON，以此類推。

4. 外部中斷 (I40~I251)

當有周邊裝置 (例：特殊 I/O 模組) 發出請求中斷時，PLC 會去執行特定的中斷工作 (Task)。

5. 定時中斷 (I252~I255)

定時中斷 0 (I252) : 預設值 100ms (1~1000ms)

定時中斷 1 (I253) : 預設值 40ms (1~1000ms)

定時中斷 2 (I254) : 預設值 20ms (1~1000ms)

定時中斷 3 (I255) : 預設值 10ms (1~1000ms)

在一定的間隔時間內，執行此定時中斷工作(Task)。例如：每隔 10ms 執行此定時中斷工作(Task)。

6

- 優先順序

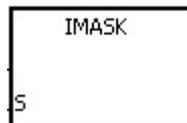
中斷編號	說明	優先順序
I0	I/O 中斷 0	1
I1	I/O 中斷 1	2
I2	I/O 中斷 2	3
I3	I/O 中斷 3	4
I4	I/O 中斷 4	5
I5	I/O 中斷 5	6
I6	I/O 中斷 6	7
I7	I/O 中斷 7	8
I8	I/O 中斷 8	9
I9	I/O 中斷 9	10
I10	I/O 中斷 10	11
I11	I/O 中斷 11	12
I12	I/O 中斷 12	13
I13	I/O 中斷 13	14
I14	I/O 中斷 14	15
I15	I/O 中斷 15	16
I16	I/O 中斷 16	17
I17	I/O 中斷 17	18
I18	I/O 中斷 18	19
I19	I/O 中斷 19	20
I20	I/O 中斷 20	21
I21	I/O 中斷 21	22
I22	I/O 中斷 22	23
I23	I/O 中斷 23	24
I24	I/O 中斷 24	25
I25	I/O 中斷 25	26
I26	I/O 中斷 26	27
I27	I/O 中斷 27	28
I28	I/O 中斷 28	29
I29	I/O 中斷 29	30
I30	I/O 中斷 30	31
I31	I/O 中斷 31	32
I32	RS 通訊請求中斷 COM1	33
I33	RS 通訊請求中斷 COM2	34

中斷編號	說明	優先順序
I34	24V 低電壓偵測中斷(LV2)：偵測外部 24V 的電壓是否正常·當外部 24V 不正常時·讓使用者可藉由中斷副程式 (I34) 執行相對應的流程	35
I35~I39	保留	36~40
I40~I251	外部中斷	41~252
I252	定時中斷 0·預設值：100ms (1ms~1000ms)	253
I253	定時中斷 1·預設值：40ms (1ms~1000ms)	254
I254	定時中斷 2·預設值：20ms (1ms~1000ms)	255
I255	定時中斷 3·預設值：10ms (1ms~1000ms)	256

6

API	指令碼			運算元								功能					
0502	IMASK			S								中斷控制					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				
脈波執行型											16 位元指令(3 steps)			32 位元指令			
-											AH500			-			

符號：



S : 資料來源

Word

指令說明：

1. **S~S+15** 的內容作中斷允許或禁止的控制，當 bit=1，且 EI 指令有執行時，相對應的中斷程式才可以執行，bit=0 時，相對應的中斷程式無法執行。
2. 執行此指令時會將 **S~S+15** 的內容值搬移至 SR623~SR638。
3. 如果沒有執行此指令，則中斷允許或禁止的控制 bit 會以 SR623~SR638 的內容來決定。

S	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100
S+1	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116
S+2	147	146	145	144	143	142	141	140	139	138	137	136	135	134	133	132
S+3	163	162	161	160	159	158	157	156	155	154	153	152	151	150	149	148
S+4	179	178	177	176	175	174	173	172	171	170	169	168	167	166	165	164
S+5	195	194	193	192	191	190	189	188	187	186	185	184	183	182	181	180
S+6	1111	1110	1109	1108	1107	1106	1105	1104	1103	1102	1101	1100	1099	1098	1097	1096
S+7	1127	1126	1125	1124	1123	1122	1121	1120	1119	1118	1117	1116	1115	1114	1113	1112
S+8	1143	1142	1141	1140	1139	1138	1137	1136	1135	1134	1133	1132	1131	1130	1129	1128
S+9	1159	1158	1157	1156	1155	1154	1153	1152	1151	1150	1149	1148	1147	1146	1145	1144
S+10	1175	1174	1173	1172	1171	1170	1169	1168	1167	1166	1165	1164	1163	1162	1161	1160
S+11	1191	1190	1189	1188	1187	1186	1185	1184	1183	1182	1181	1180	1179	1178	1177	1176
S+12	1207	1206	1205	1204	1203	1202	1201	1200	1199	1198	1197	1196	1195	1194	1193	1192
S+13	1223	1222	1221	1220	1219	1218	1217	1216	1215	1214	1213	1212	1211	1210	1209	1208
S+14	1239	1238	1237	1236	1235	1234	1233	1232	1231	1230	1229	1228	1227	1226	1225	1224
S+15	1255	1254	1253	1252	1251	1250	1249	1248	1247	1246	1245	1244	1243	1242	1241	1240

補充說明：

S~S+15 裝置超出範圍時，指令不執行，SM0=ON，SR0 Error Code=16#2003。

6

6.7 I/O 更新指令

6.7.1 I/O 更新指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>0600</u>	REF	-	✓	I/O 更新處理	5

6.7.2 I/O 更新指令說明

API	指令碼			運算元								功能				
0600		REF	P	D · n								I/O 更新處理				

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號:



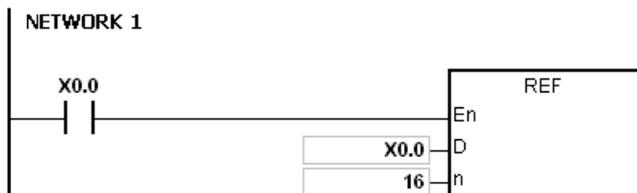
D : 欲 I/O 更新處理之起始裝置 Bit
n : I/O 更新處理之數目 1~256

指令說明 :

PLC 的 I/O 更新狀態全部為程式掃描至 END 後，才作狀態的更新，其中輸入點的狀態是在程式開始掃描時，自外部輸入點的狀態讀入存在輸入點記憶體中，而輸出端子在 END 指令後，才將輸出點記憶體內容送至輸出裝置。因此在演算過程中需要最新的 I/O 資料，則可利用本指令。

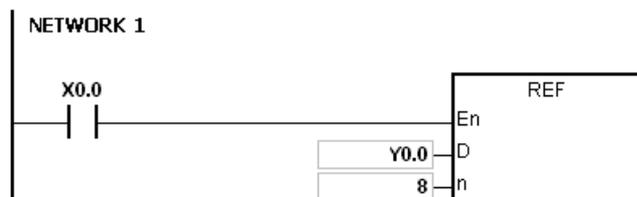
程式範例一：

當 X0.0=ON 時，PLC 會立即讀取 X0.0~X0.15 之輸入點狀態，輸入信號更新，並沒有輸入延遲。



程式範例二：

當 X0.0=ON 時，Y0.0~Y0.7 之 8 點輸出信號即時被送至輸出端，輸出信號立即更新，不必到 END 指令才輸出。



補充說明：

1. **D+n-1** 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 當 **n>256** 或 **n<1** 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

6.8 便利指令

6.8.1 便利指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>0700</u>	ALT	–	✓	ON/OFF 交替	3
<u>0701</u>	TTMR	–	–	教導式計時器	5
<u>0702</u>	STMR	–	–	特殊計時器	7
<u>0703</u>	RAMP	–	–	傾斜信號	9
<u>0704</u>	MTR	–	–	矩陣輸入	9
<u>0705</u>	ABSD	DABSD	–	絕對方式凸輪控制	9
<u>0706</u>	INCD	–	–	相對方式凸輪控制	9
<u>0707</u>	–	DPID	–	PID 運算	35
<u>0708</u>	–	DPIDE	–	PID 運算	43

6.8.2 便利指令說明

API	指令碼			運算元								功能				
0700		ALT	P	D								ON/OFF 交替				

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

脈波執行型	16 位元指令 (3 steps)	32 位元指令
AH500	AH500	-

符號：



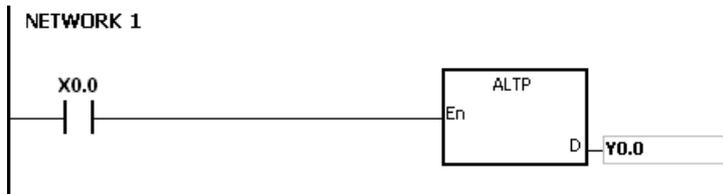
D : 目的地裝置 Bit

指令說明：

1. ALT 指令執行時，D ON/OFF 交換。
2. 本指令一般都是使用脈波執行型指令 (ALTP)。

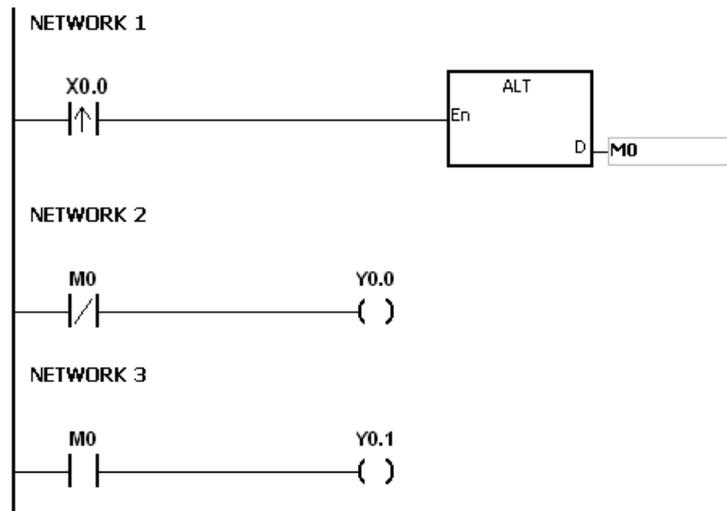
程式範例一：

當第一次 X0.0 從 OFF→ON 時，Y0.0=ON。第二次 X0.0 從 OFF→ON 時，Y0.0=OFF。



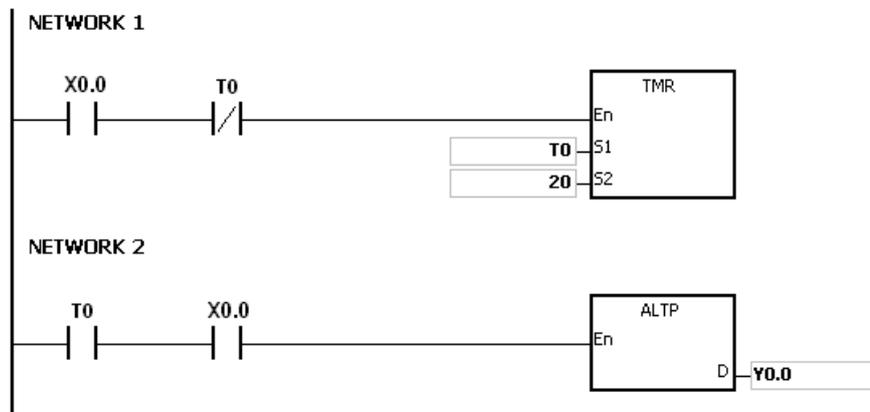
程式範例二：

使用單一開關控制啟動與停止。一開始時，M0=OFF 故 Y0.0=ON、Y0.1=OFF，當 X0.0 作第一次 ON/OFF 時，M0=ON 故 Y0.1=ON、Y0.0=OFF，第二次 ON/OFF 時，M0=OFF 故 Y0.0=ON 而 Y0.1=OFF。



程式範例三：

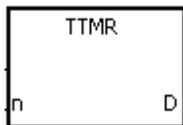
產生閃爍之動作。當 X0.0=ON 時，T0 每隔 2 秒產生一個脈波，Y0.0 輸出會依 T0 脈波產生 ON/OFF 交替。



API	指令碼			運算元								功能					
0701		TTMR		D · n								教導式計時器					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●						●	●		●		●				
n	●	●						●	●		●		●	○	○		

脈波執行型	16 位元指令(5 steps)	32 位元指令
-	AH500	-

符號：



D：儲存按鈕開關 ON 時間之裝置編號 Word
n：倍數設定 Word

指令說明：

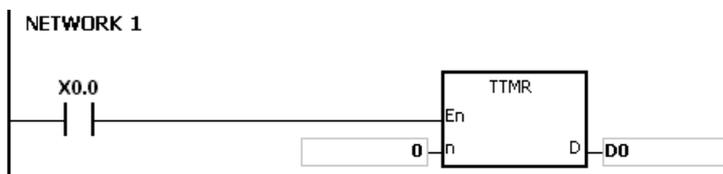
- 將開關 ON 時間以秒為單位乘以 **n** 倍數後存放於 **D** 內，**D+1** 為系統內部使用，指令執行時不可更改，否則會造成時間計算錯誤。
- D** 的內容值在條件接點為 ON 的時候會先清為 0。
- 倍數設定：**n=0** 時，**D** 以秒為單位，**n=1** 時，**D** 乘以 10 倍以 100ms 為單位，**n=2** 時，**D** 乘以 100 倍以 10ms 為單位。

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

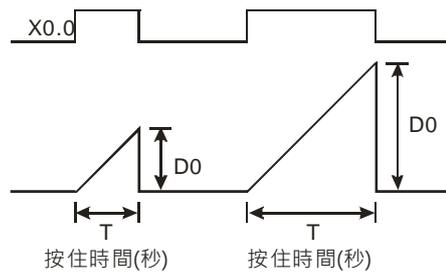
- 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。
- n** 運算元的範圍為 0~2。

程式範例一：

- 按鈕開關 X0.0 被按住時間 (X0.0 的 ON 時間)，由 **n** 來指定該時間的倍數，並將位數時間存入於 **D0** 當中。由此，可使用按鈕開關來調整計時器的設定值。
- 當 X0.0 變成 OFF 時，**D0** 內容沒有變化。



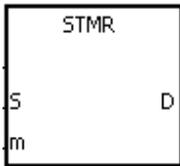
6

**補充說明：**

1. **D+1** 超出裝置範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n < 0$ 或 $n > 2$ 時，指令不執行，SM0=ON，SR0 記錄錯誤碼=16#200B。
3. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。

API	指令碼			運算元								功能					
0702		STMR		S · m · D								特殊計時器					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S					○												
m	●	●						●	●		●		●	○	○		
D	●	●	●	●				●	●	●			●				
脈波執行型											16 位元指令(7 steps)			32 位元指令			
-											AH500			-			

符號：



- S** : 計時器編號 T0~T2047
- m** : 計時器設定值 Word
- D** : 輸出裝置之起始編號 Bit

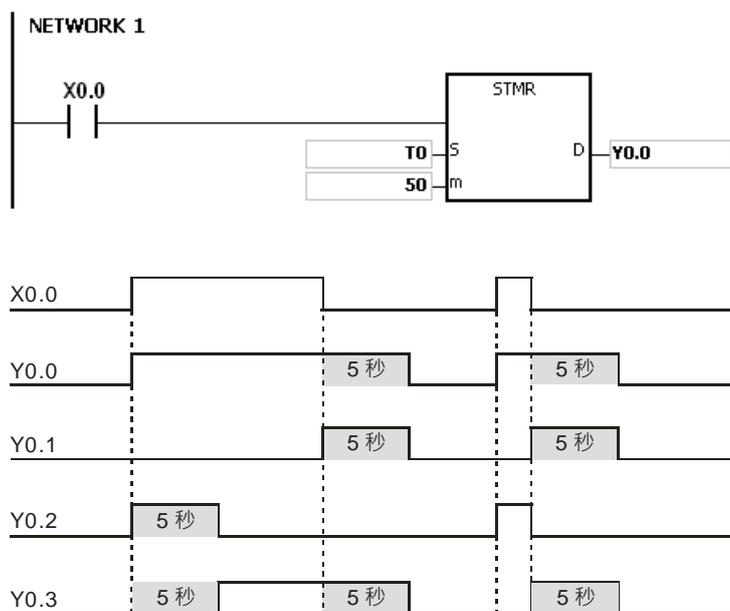
指令說明：

1. STMR 指令，用來產生 OFF 延遲，一次觸發及閃爍回路的專用指令。
2. STMR 指令所指定之 T 計時器以 100ms 為單位。
3. STMR 指令所指定的計時器號碼不可重複使用。
4. **D** 連續佔用 4 個 bit。
5. 執行此指令前，請先 RST D~D+3 的狀態。
6. 當條件接點未啟動時，如果裝置內容為以下兩種，則 D、D+1、D+3 的狀態會 ON m 秒之後再 OFF，除了以下兩種情形之外，D~D+3 的狀態會保持 OFF。
 - T 計時器內容值如果小於或等於 m，且 D 的狀態為 ON，D+1 的狀態為 OFF。
 - T 計時器內容值小於 m，且 D、D+1、D+3 的狀態為 ON，D+2 的狀態為 OFF。
7. 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。
8. **m** 運算元的範圍為 1~32767

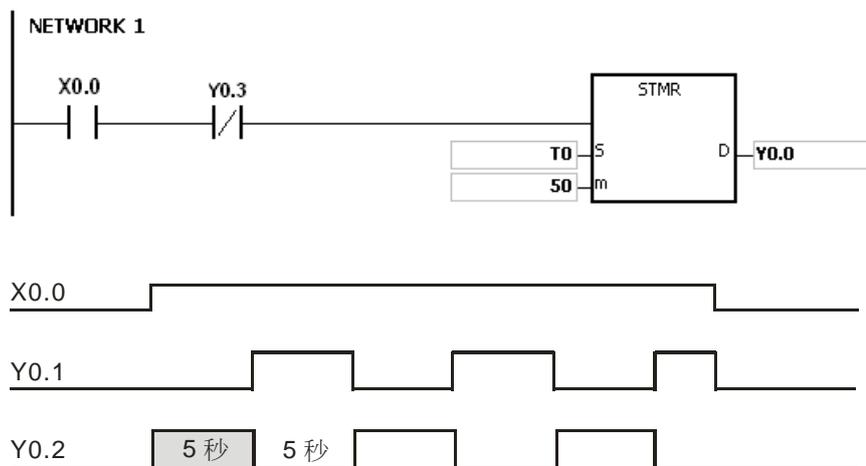
程式範例：

1. 當 X0.0=ON 時，STMR 指令指定計時器 T0，T0 的設定值為 5 秒。
2. Y0.0 為 OFF 延遲接點：當 X0.0 由 OFF 變 ON 時，Y0.0=ON，到 X0.0 由 ON 變 OFF 時，延遲 5 秒後 Y0.0=OFF。
3. Y0.1 於 X0.0 由 ON 變 OFF 時，作一次 5 秒 Y0.1=ON 輸出。
4. Y0.2 於 X0.0 由 OFF 變 ON 時，作一次 5 秒 Y0.2=ON 輸出。
5. Y0.3 於 X0.0 由 OFF 變 ON 時，延遲 5 秒後 Y0.3=ON，到 X0.0 由 ON 變 OFF 時，延遲 5 秒後 Y0.3=OFF。

6



6. 在條件接點 X0.0 後面加一個 Y0.3 的 b 接點，則 Y0.1、Y0.2 可作閃爍回路輸出。當 X0.0 變成 OFF 時，Y0.0、Y0.1 及 Y0.3 變成 OFF，T0 的內容被復歸為 0。



補充說明：

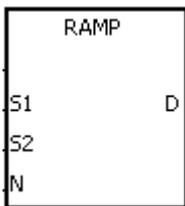
1. **D+3** 超出裝置範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **m<=0**，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of BOOL。

API	指令碼		運算元				功能					
0703	RAMP		$S_1 \cdot S_2 \cdot D \cdot n$				傾斜信號					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
-	AH500	-

符號：



- S_1 : 傾斜信號之起點設定 Word
- S_2 : 傾斜信號之終點設定 Word
- D : 傾斜信號之經過時間值 Word
- n : 掃描次數 Word

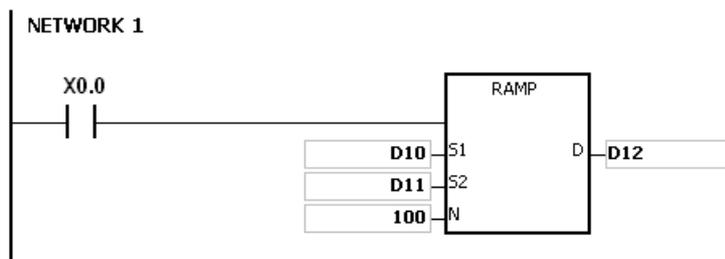
指令說明：

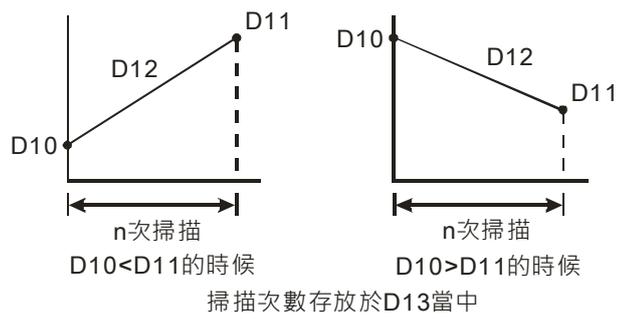
1. 本指令是一個求斜率的指令，斜率是線性與掃描時間有絕對的關係，因此使用本指令時，通常必須預先將掃描時間加以固定。
2. 預先將傾斜信號之起點設定值寫入 S_1 及傾斜信號之終點設定值寫入 S_2 內，當條件接點=ON 時，D 從 S_1 所設定的值朝 S_2 邁進 (增加)，掃描次數存放於 D+1 當中，當 $D = S_2$ 或 $D+1=n$ 已到達掃描次數時，SM687 完成旗標=ON。
3. 當條件接點未啟動時，D、D+1 的內容值為 0，SM687 完成旗標=OFF。
4. 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。
5. 若需要固定掃描時間，請參考 ISPSOFT 手冊中的硬體組態工具 說明。
6. n 運算元的範圍為 1~32767。

程式範例：

本指令若是與類比信號輸出搭配使用時，可執行緩衝啟動/停止的動作。

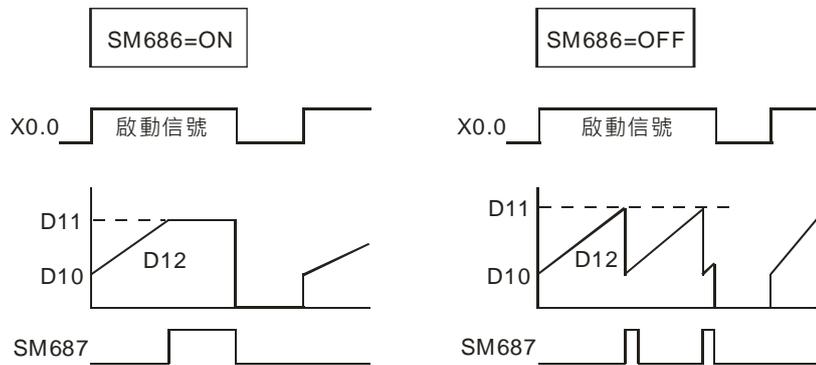
1. 指令執行中，啟動信號 X0.0 變成 OFF 時，指令停止執行，當 X0.0 再度 ON 的時候，SM687=OFF，D12 的內容被復歸為 D10 的設定值，D13 的內容被清除為 0，再重新計算。
2. SM686=OFF 時，SM687=ON，D12 的內容被復歸成 D10 的設定值，D13 的內容被清除為 0。





補充說明：

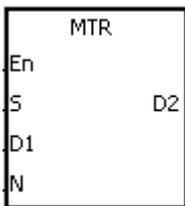
1. D+1 超出裝置範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n \leq 0$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
4. 啟動模式旗標信號 SM686 的 ON/OFF，D12 的內容變化如下：



API	指令碼				運算元							功能					
0704		MTR			S · D ₁ · D ₂ · n							矩陣輸入					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	○																
D ₁		○															
D ₂	○	○	○	○				○	○				○				
n	●	●	●	●				●	●		●		●	○	○		

脈波執行型	16 位元指令 (9 steps)	32 位元指令
-	AH500	-

符號：



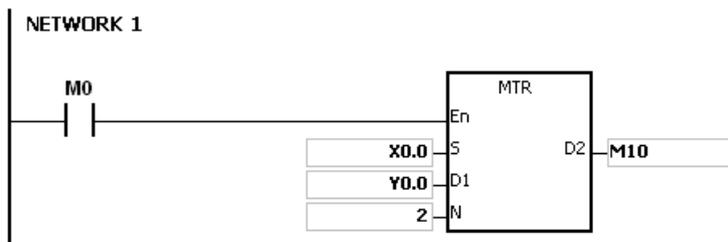
- S** : 矩陣掃描輸入起始裝置 Bit
- D₁** : 矩陣掃描輸出起始裝置 Bit
- D₂** : 矩陣掃描所對應起始裝置 Bit
- n** : 矩陣掃描之列數 Word

指令說明：

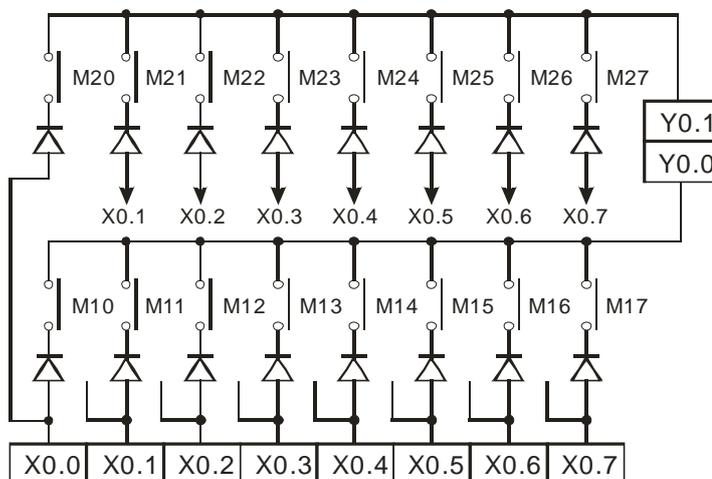
- S** 指定矩形所有連接輸入端的起始號碼，一旦指定後，該號碼開始算連續 8 點為矩陣輸入端。
- D₁** 指定那一個電晶體輸出 Y 為矩陣掃描的起始號碼，條件接點 OFF 時，**D₁** 起始的 **n** 個裝置狀態會保持 OFF。
- 每一次掃描周期可以更新一系列的輸入點，每一列為 16 點，從 1~**n** 列反覆的掃描。
- 本指令由 **S** 起始的連續 8 個輸入端，以 **D₁** 起始的 **n** 個外部輸出點用矩陣掃描之方式讀取 **n** 列的 8 個開關，變成 **8×n** 的多點矩陣輸入點。並將掃描讀取的開關狀態反應在 **D₂** 起始之裝置。
- 使用本指令時，最大可將 8 個輸入開關並接 8 列可得 64 個輸入點 (**8×8=64**)。
- 本指令的執行間隔必須比模組 I/O 更新的時間長，否則無法讀到正確的輸入點。
- 本指令的條件接點一般都使用常 ON 接點 SM400。
- n** 運算元的範圍 2~8。

程式範例一：

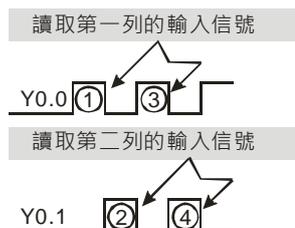
- 當 M0=ON，MTR 指令開始執行，外部 2 列共 16 個開關之狀態被順序讀取並存放在內部繼電器 M10~M17，M20~M27。



2. 下圖由 X0.0~X0.7 及 Y0.0~Y0.7 構成 2 列矩陣輸入迴路之外部接線圖，16 個開關對應到內部繼電器 M10~M17，M20~M27。搭配 MTR 指令使用。



3. 以上圖為例，X0.0 開始算的 8 點由 Y0.0~Y0.1 (n=2) 作矩陣掃描，D₂ 指定讀入結果的起始號碼為 M10，代表第一列的開始被讀入至 M10~M17，第二列被讀入至 M20~M27。



補充說明：

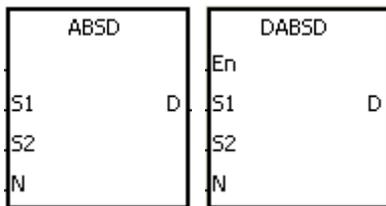
1. 本指令被執行時，當掃描週期太長或太短都可能造成按鍵輸入不良因此可運用下列技巧來克服。
2. 當掃描週期太短時，可能造成 I/O 來不及反應而無法讀取正確之按鍵輸入，此時，可將掃描時間加以固定。
3. 當掃描週期太長時，可能會使按鍵反應變得遲鈍，可將此指令寫在時間中斷工作 (TASK) 內，固定時間執行此指令。
4. $S+7$ 、 D_1+n-1 、 $D_2+(n*8)-1$ 超出裝置範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
5. $n<2$ 或 $n>8$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
6. S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [8] of BOOL。

API	指令碼			運算元							功能						
0705	D	ABSD		$S_1 \cdot S_2 \cdot D \cdot n$							絕對方式凸輪控制						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
-	AH500	AH500

符號：



- S_1 : 指比較表起始裝置 Word/Double
- S_2 : 比較值 Word/Double
- D : 比較結果起始編號 Bit
- n : 多段比較的組數 Word

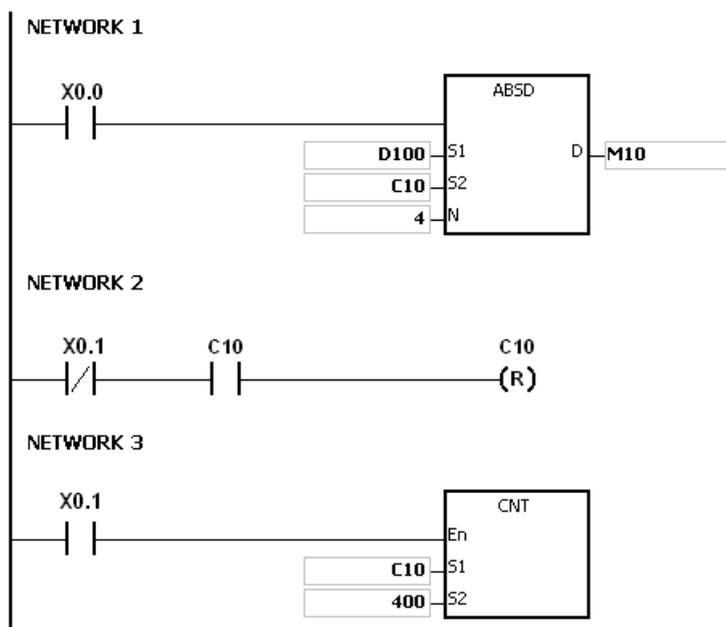
指令說明：

1. ABSD 指令為為對應計數器的現在值產生多個輸出波形的指令，通常用來做絕對方式凸輪控制。
2. DABSD 指令才可以使用 HC 裝置。
3. ABSD 時 $n=1\sim 256$ ，DABSD 時 $n=1\sim 128$ 。

6

程式範例一：

1. 於 ABSD 指令被執行前使用 MOV 指令預先將各設定值寫入至 D100~D107。偶數 D 號碼的內容為下限值，奇數 D 號碼的內容為上限值。
2. 當 X0.0=ON 時，計數器 C10 的現在值與 D100~D107 等 4 組上下限值作區域比較，結果分別反應在 M10~M13。
3. X0.0=OFF 時，原 M10~M13 的 ON/OFF 狀態不會變化。

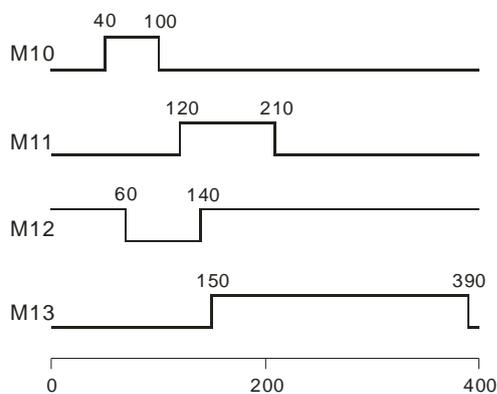


4. 在大於等於下限值且小於等於上限值範圍內對應之 M10~M13 會 ON，否則為 OFF。

下限值	上限值	C10 現在值	輸出
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. 若下限值大於上限值時，則小於上限值 ($C10 < 60$) 或大於下限值 ($C10 > 140$) 時，M12=ON，否則為 OFF。

下限值	上限值	C10 現在值	輸出
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

補充說明：

1. ABSD 指令中， $S+2*n-1$ 超出裝置範圍，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. DABSD 指令中， $S+4*n-1$ 超出裝置範圍，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
3. ABSD 指令中， $D+n-1$ 超出裝置範圍，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
4. DABSD 指令中， $D+2*n-1$ 超出裝置範圍，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
5. ABSD 指令中，若 $n<1$ 或 $n>256$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
6. DABSD 指令中，若 $n<1$ 或 $n>128$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。

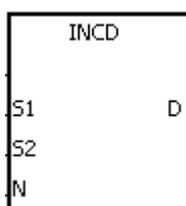
6

API	指令碼			運算元								功能					
0706	INCD			$S_1 \cdot S_2 \cdot n \cdot D$								相對方式凸輪控制					

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

脈波執行型	16 位元指令(9 steps)	32 位元指令
-	AH500	-

符號：



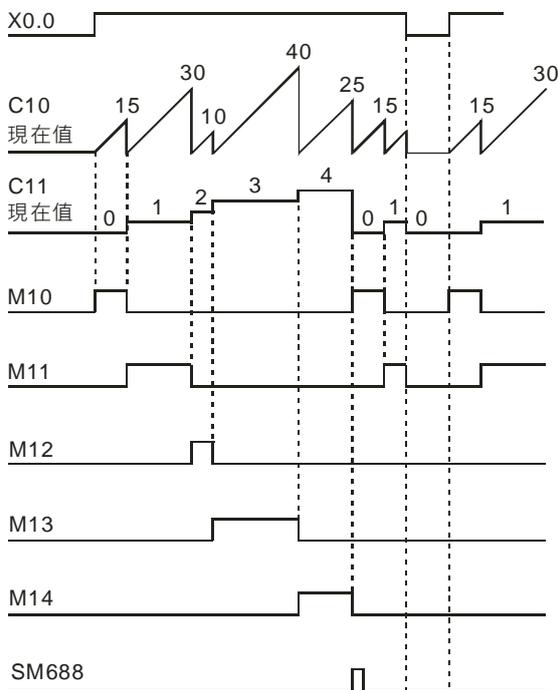
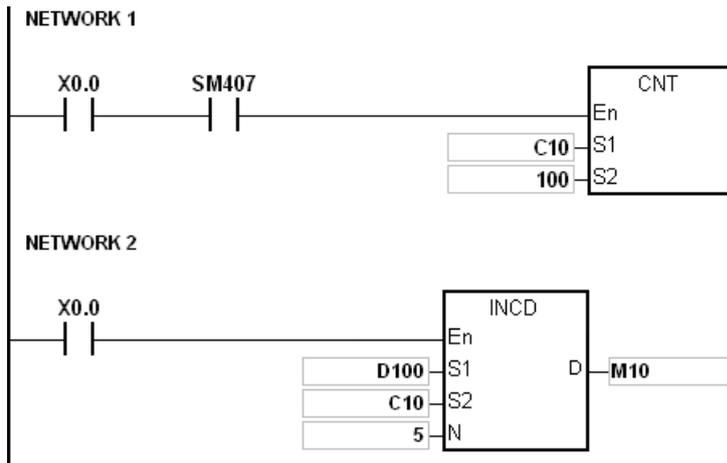
S_1 : 指比較表起始裝置 Word
 S_2 : 計數器編號 Word
D : 比較結果起始編號 Bit
n : 多段比較的組數 Word

指令說明：

1. INCD 指令為用一對計數器產生多個輸出波形的指令，通常用來作相對方式凸輪控制。
2. S_2 的現在值與 S_1 的設定值作比較，每到達一個設定值， S_2 的現在值被復歸為 0 重新計數。而目前執行之段數被暫存於 S_2+1 當中。
3. n 的組數比較完成時，指令執行完畢旗標 SM688 會 ON 一次掃描週期。
4. 當條件接點未啟動時， S_2 、 S_2+1 的內容值為 0， $D-D+n-1$ 狀態為 OFF，SM688 為 OFF。
5. 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。
6. n 運算元的範圍 1~256。

程式範例：

1. 於 INCD 指令被執行前，使用 MOV 指令預先將各設定值寫入至 D100~D104 當中，D100=15、D101=30、D102=10、D103=40、D104=25。
2. 計數器 C10 的現在值與 D100~D104 的設定值作比較，每到達一個設定值，C10 的現在值被復歸為 0 重新計數。
3. 而目前執行之段數被暫存於 C11 當中。
4. C11 的內容每變動 1 時，M10~M14 相對應動作，請參考下列時序圖。
5. 5 組數比較完成時，指令執行完畢旗標 SM688 會 ON 一次掃描週期。
6. 當 X0.0 由 ON 變成 OFF 時，C10 及 C11 全部被復歸為 0，M10~M14 亦全部變成 OFF，當 X0.0 再度 ON 時，本指令被從頭執行起。



6

補充說明：

1. S_2+1 超出裝置範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. S_1+n-1 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. $D+n-1$ 超出裝置範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. $n<1$ 或 $n>256$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
5. S_2 運算元，若使用 ISPSofT 宣告，則資料型態為 ARRAY [2] of WORD/INT。

API	指令碼		運算元										功能			
0707	D	PID	PID_RUN · SV · PV · PID_MODE · PID_MAN · MOUT_AUTO · CYCLE · K _p · K _i · K _d · PID_DIR · ERR_DBW · MV_MAX · MV_MIN · MOUT · I_MV · MV										PID 運算			

裝置	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								●	●				●				
K _p								●	●				●				
K _i								●	●				●				
K _d								●	●				●				
PID_DIR	●	●	●					●	●	●			●				
ERR_DBW								●	●				●				
MV_MAX								●	●				●				
MV_MIN								●	●				●				
MOUT								●	●				●				
I_MV								●	●				●				
MV								●	●				●				

脈波執行型	16 位元指令	32 位元指令 (35 steps)
-	-	AH500

6

符號：

PID	
En	
RUN	MV
SV	
PV	
MODE	
MAN	
MOUT_A	
CYCLE	
K _p	
K _i	
K _d	
DIR	
ERR_DBW	
MV_MAX	
MV_MIN	
MOUT	
I_MV	

PID_RUN	: 啟動 PID 運算	BIT
SV	: 目標值 (SV)	DWORD
PV	: 現在值 (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	: 取樣時間 (CYCLE)	DWORD
K_p	: P 計算值系數 (K _p)	DWORD
K_i	: I 計算值系數 (K _i)	DWORD
K_d	: D 計算值系數 (K _d)	DWORD
PID_DIR	: PID 正反向 (PID_DIR)	BIT
ERR_DBW	: ERR 的不作用範圍 (ERR_DBW)	DWORD

MV_MAX	: MV 上限值 (MV_MAX)	DWORD
MV_MIN	: MV 下限值 (MV_MIN)	DWORD
MOUT	: MV 手動值 (MOUT)	DWORD
I_MV	: 累計積分項的數值 (I_MV)	DWORD
MV	: MV 輸出值 (MV)	DWORD

指令說明：

- PID 運算控制的專用指令，於取樣時間到達後的該次掃描才執行 PID 運算動作。PID 表示“比例、積分和微分”。PID 控制在機械設備、氣動設備和電子設備中具有廣泛的應用。
- 參數設定內容如下：

裝置編號	資料型態	功能	設定範圍	說明
PID_RUN	BOOL	啟動 PID 運算	TRUE：PID 指令開始運算。 FALSE：MV 值清除為 0，PID 不運算。	
SV	REAL	SV 值	單精度浮點 數範圍	目標值
PV	REAL	PV 值	單精度浮點 數範圍	現在值
PID_MODE	DWORD/DINT	PID 控制模式	0：自動控制，當 PID_MAN 由 TRUE 轉為 FALSE 時，MV 值會由當時輸出的 MV 值開始進行自動運算。 1：溫度控制專用的自動調整參數功能，調整完畢時將自動改為 0，並且填入最適用的 K_p 、 K_i 及 K_d 等參數。 2：與 0 相同，但當 PID_MAN 由 TRUE 轉為 FALSE 時，MV 值會由當時內部運算的 MV 值開始進行自動運算。 設定值超出範圍，將視為 0。	
PID_MAN	BOOL	PID A/M 模式	TRUE：Manual，MV 值會依 MOUT 值輸出，但仍在 MV_MAX 與 MV_MIN 之間，當 PID_MODE 為 1 時此設定無效。 FALSE：Auto，MV 值會依 PID 公式計算後輸出，輸出值在 MV_MAX 與 MV_MIN 之間。	
MOUT_AUTO	BOOL	MOUT Auto change 模式	TRUE：Auto，MOUT 的值會隨著 MV 值改變。 FALSE：Normal，MOUT 的值不會隨著 MV 值改變。	

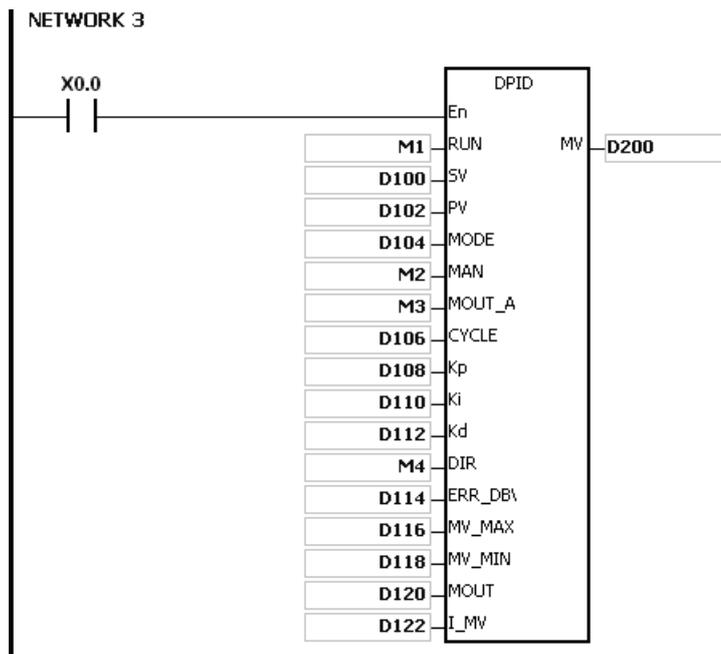
裝置編號	資料型態	功能	設定範圍	說明
Cycle	DWORD/DINT	取樣時間 (T_s)	1~2,000 (單位 : 10ms)	每次掃描到本指令時，就會以 Cycle 設定的取樣時間來計算 PID(不會計算掃描時間是否有到達取樣時間)，並更新輸出值 (MV)。 T_s 小於 1 則內定為 1，大於 2000 則為 2000。 當 PID 指令用於 Interval Interrupt Task 時，Cycle 時間以 Interval Interrupt Task 的時間為主。
K_P	REAL	比例增益 (K_P)	正數單 精度浮 點數範 圍	為 SV-PV 間的誤差放大比例值，如果小於 0， K_P 將為 0
K_I	REAL	積分增益 (K_I)	正數單 精度浮 點數範 圍	I 計算值系數 (K_I)，如果小於 0， K_I 將為 0
K_D	REAL	微分增益 (K_D)	正數單 精度浮 點數範 圍	D 計算值系數 (K_D)，如果小於 0， K_D 將為 0
PID_DIR	BOOL	PID 正反向	TRUE：反向動作 ($E=SV-PV$) FALSE：正向動作 ($E=PV-SV$)	
ERR_DBW	REAL	偏差量 (E) 不作用範圍	單精度 浮點數 範圍	偏差量 (E) 等於 SV-PV 的誤差值，當設定值為 0 即表示不啟動此功能。 例：設定值為 5 或 -5，則 E 在 -5~5 的區間偏差量將為 0
MV_MAX	REAL	輸出值 (MV) 飽和上限	單精度 浮點數 範圍	例：設定 1,000，則輸出值 (MV) 大於 1,000 時將以 1,000 輸出，需大於 S3+7，否則上限值與下限值將互換

裝置編號	資料型態	功能		設定範圍	說明
MV_MIN	REAL	輸出值 (MV) 飽和 和下限		單精度 浮點數 範圍	例：設定-1,000，則輸出值 (MV) 小於-1,000 時將以 -1,000 輸出
MOUT	REAL	MV 手動值		配合 PID_MAN 模式使用，使用者直接 設定 MV 輸出值。	
I_MV (連續 佔用 6 個 DWord 裝 置)	REAL	I_MV	暫存 累積的積 分值	單精度 浮點數 範圍	為累積的積分值，通常只 供參考用，但是使用者還 是可以依需求清除或修 改，不過須以單精度浮點 數修改之，當 MV 超出 MV_MAX 或 MV_MIN 時，I_MV 值不會再改變
		I_MV+1	暫存 前次 PV 值	為前次測定值，通常只供參考用，但 是使用者還是可以依需求修改。	
		I_MV+2 ~ I_MV+5	系統用參數，使用者請勿使用。		
MV	REAL	MV 輸出 值		MV 值會介於 MV_MAX 與 MV_MIN 之間。	

6

程式範例：

1. 執行 PID 指令前將參數設定完成。
2. X0.0=ON 的時候指令被執行，M1=ON 的時候 PID 指令才開始進行運算，M1=OFF 時 MV 值為 0，MV 值存於 D200 中。X0.0 變成 OFF 時，指令不被執行，之前的資料沒有變化。



補充說明：

1. 指令無使用次數之限制，但是 $I_MV \sim I_MV_{+5}$ 所指定的暫存器號碼不可重覆。
2. I_MV 佔用 12 個暫存器，於上述程式例當中 I_MV 指定 PID 指令的參數設定區域為 D122~D133。
3. PID 指令只能使用在 Cyclic Task 與 Interval Interrupt Task 當中，當指令用於 Interval Interrupt Task 時，Cycle 時間以 Interval Interrupt Task 的時間為主。
4. PID 被啟動後，每次只要被掃描到都會以 Cycle 所設定的時間來做 PID 運算並直接輸出 MV 值，並不會自動計算掃描時間是否有到達取樣時間才輸出，但如果本指令使用在時間中斷程式內，取樣時間將與時間中斷程式的間隔時間相同，PID 指令會以時間中斷程式的中斷間隔時間來做 PID 的運算。
5. PID 的測定值 (PV) 在 PID 執行運算動作前必須是一個穩定值。如果要抓取特殊模組的輸入值作 PID 運算時，請注意這些模組的 A/D 轉換時間。

PID 計算公式：

1. 當 **PID_MODE** 控制模式選擇為 0、2，自動控制模式。
2. 當 **PID_MODE** 控制模式選擇為 1，自動調整模式，當自動調整完成，**PID_MODE** 會變成 0 自動控制模式。
 - a) PID 的運算分成正向動作，逆向動作。而正逆向動作由 **DIR** 來指定。

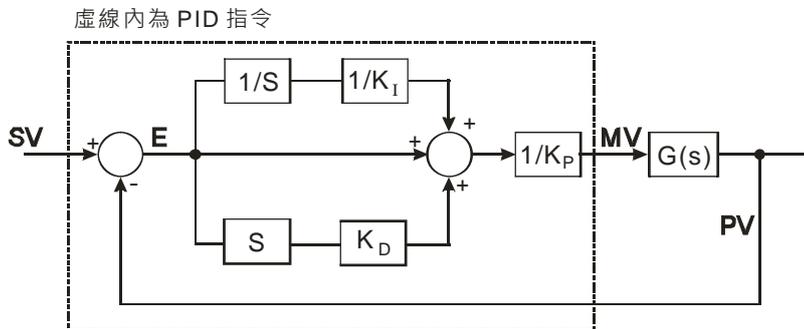
PID 的運算式：

$$MV = K_p E(t) + K_I \int_0^t E(t) dt + K_D * \frac{dE(t)}{dt}$$

其中 $E(t)S$ 表示 $E(t)$ 的微分值，以及 $E(t) \frac{1}{S}$ 表示 $E(t)$ 的積分值。

動作方向	PID 演算方式
正向動作、自動	$E(t) = PV(t) - SV(t)$
逆向動作	$E(t) = SV(t) - PV(t)$

- 控制方塊圖：下圖之 S 表示微分的動作，其動作定義為現在 $E(t)$ 值減去前次 $E(t)$ 值後，再除以取樣時間之動作，另外 $1/S$ 表示積分的動作，其動作定義為前次積分值加上這次偏差量乘以取樣時間的值，最後圖中的 $G(S)$ 表示受控裝置。



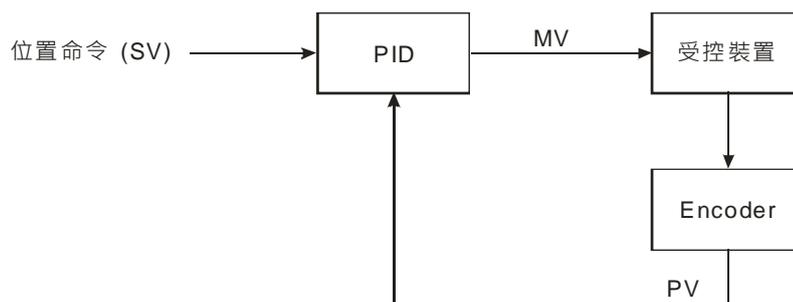
- 符號說明：
 - MV ：輸出值
 - $E(t)$ ：偏差量。正向動作 $E(t) = PV - SV$ ，反向動作 $E(t) = SV - PV$
 - K_p ：比例增益
 - PV ：測定值
 - SV ：目標值
 - K_D ：微分增益
 - K_I ：積分增益

注意事項和建議：

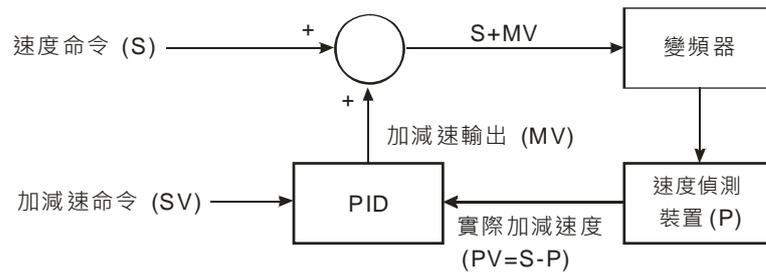
6

1. 由於可使用 PID 指令的控制環境很多，因此請適當的選取控制功能，例如：當選擇自動調整參數 ($PID_MODE=1$) 功能時，就請勿使用於馬達控制環境中，以免造成控制不當的現象發生。
2. 使用者於調整 K_p 、 K_I 及 K_D 三個主要參數時 ($PID_MODE=0、2$)，請先調整 K_p 值 (依經驗值設定)，而 K_I 及 K_D 值先設定為 0，等到調整到大致上可控制時，再依序調整 K_I 值 (由小到大) 以及 K_D 值 (由小到大)，調整範例如範例四所示。其中 K_p 值為 1 則表示 100%，即對偏差值的增益為 1，小於 100% 將對偏差值衰減，大於 100% 將對偏差值放大。
3. 當使用者選用 ($PID_MODE=1$) 時，建議請使用在停電保持區之 D 暫存器來儲存參數，以免自動調整過的參數因停電後而消失。經過自動調整過的參數，並不能保證一定適用於每個控制的環境，因此使用者當然可自行修改調整過的參數，不過建議最好只修改 K_I 或 K_D 數值就好。
4. 本指令動作須配合許多參數值控制，因此請勿隨意設定參數值，以免造成無法控制之現象。

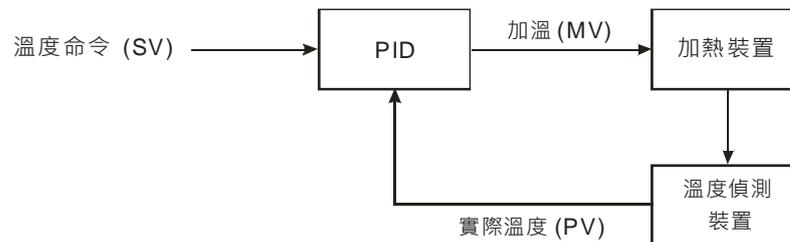
範例 1：使用 PID 指令於位置控制時的方塊圖 ($PID_MODE=0、2$)



範例 2：與變頻器搭配控制時之方塊圖 (PID_MODE=0、2)



範例 3：使用 PID 指令於溫度控制時的方塊圖 (PID_MODE=0、2)

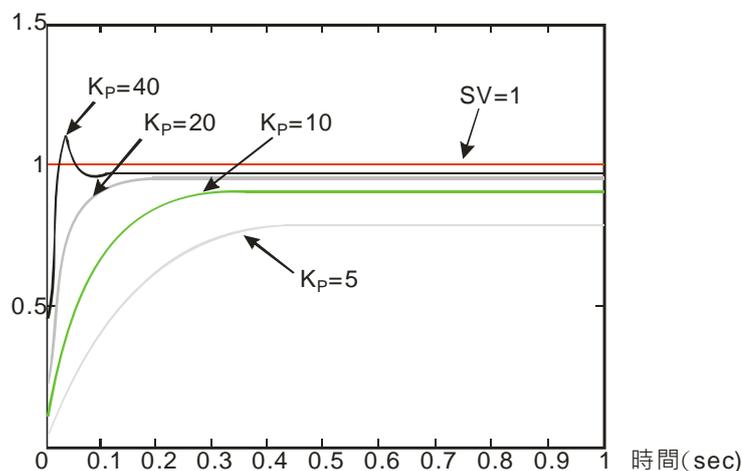


範例 4：PID 指令參數調整建議步驟說明

假設控制系統之受控體 $G(s)$ 的轉移函數為一階的函數 $G(s) = \frac{b}{s+a}$ (一般馬達的模型均為此函數)。

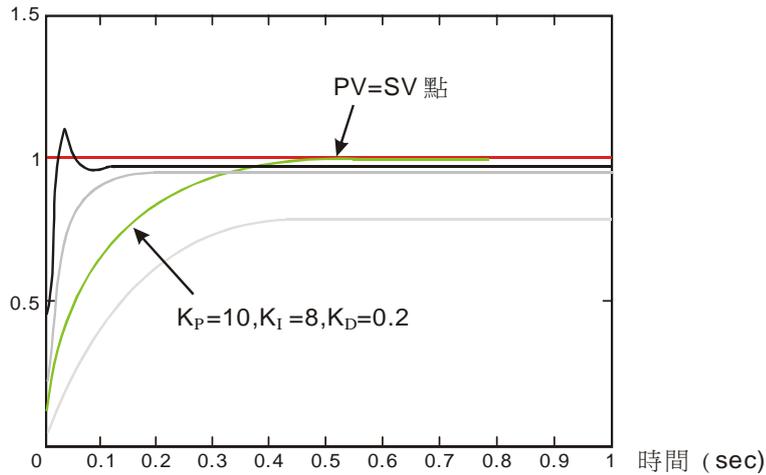
命令值 SV 為 1，取樣時間 T_s 為 10ms。建議調整步驟如下：

步驟 1：首先將 K_I 及 K_D 值設為 0，接著先後分別設定 K_P 為 5、10、20 及 40，並分別記錄其 SV 及 PV 狀態，其結果如下圖所示。



步驟 2：觀察上圖後得知 K_P 為 40 時，其反應會有過衝現象，因此不選用，而 K_P 為 20 時，其 PV 反應曲線接近 SV 值且不會有過衝現象，但是由於啟動過快，因此輸出值 MV 瞬間值會很大，所以考慮暫不選用，接著 K_P 為 10 時，其 PV 反應曲線接近 SV 值並且是比較平滑接近，因此考慮使用此值，最後 K_P 為 5 時，其反應過慢，因此也暫不考慮使用。

步驟 3：選定 K_P 為 10 後，先調整 K_I 值由小到大 (如 1、2、4 至 8)，以不超過 K_P 值為原則，然後再調整 K_D 由小到大 (如 0.01、0.05、0.1 及 0.2)，以不超過 K_P 的 10% 為原則，最後可得如下圖之 PV 與 SV 的關係圖。



附註：本範例僅供參考，因此使用者還需依實際控制系統之狀況，自行調整其適合之控制參數。

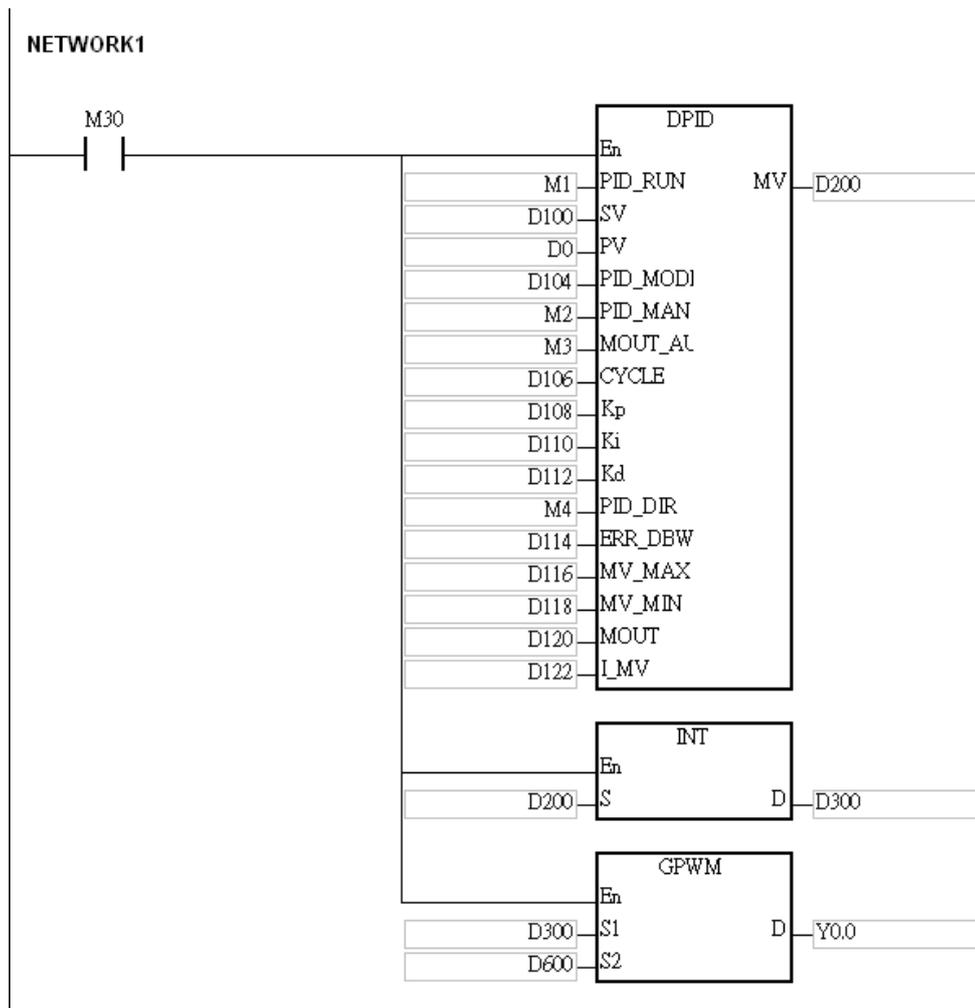
實例 1：使用自動調整功能控制溫度

控制目的：利用自動調整功能計算出最佳的 PID 溫度控制的參數。

控制說明：

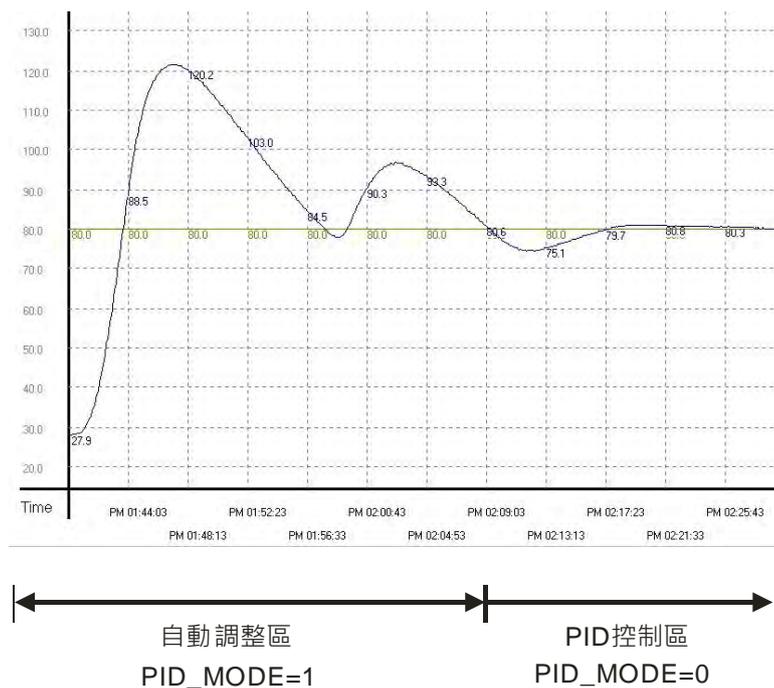
由於一般使用者對於第一次控制的溫度環境特性通常不太了解，因此可先使用自動調整功能 (**PID_MODE=1**) 做一初步調整，待調整完畢後，本指令將自動修改控制模式為 (**PID_MODE=0**) 。本實例的控制環境為烤箱。範例程式如下圖所示：

6

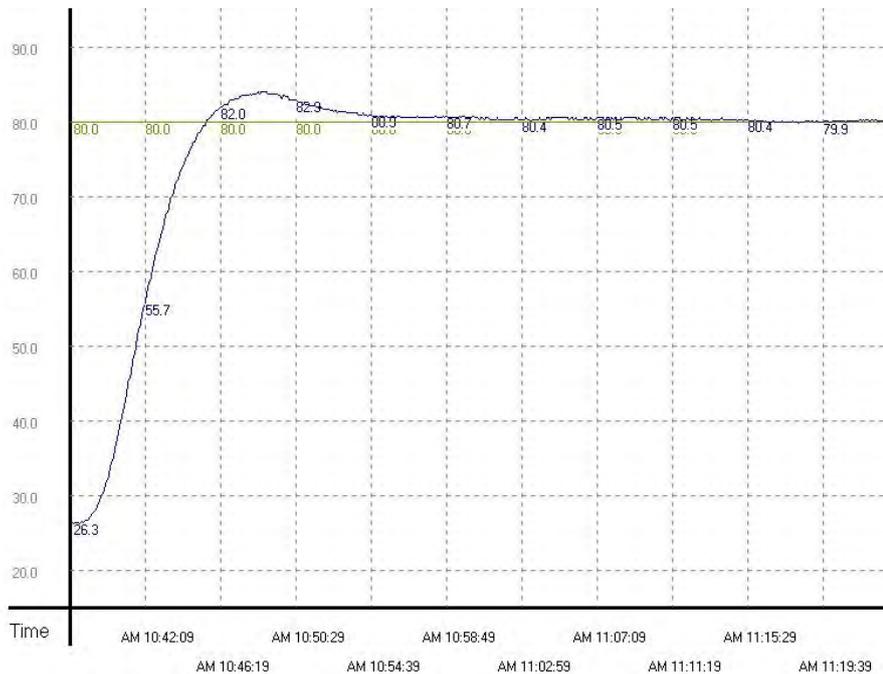


6

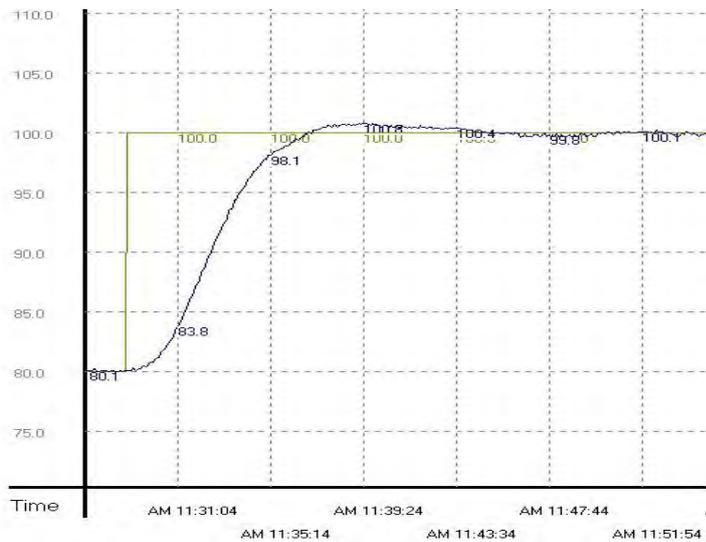
自我調整功能的實驗結果如下所示：



使用調整後參數做溫度控制專用功能的實驗結果如下所示：



由上圖可看出經過自我調整後的溫度控制結果還不錯，而且控制時間大約只使用了 20 分鐘。接著驗證目標溫度由 80 度修改成 100 度，則得到的結果如下圖所示：



由上圖中可看出由 80 度所調整出來的參數使用到 100 度時，還是可以達到控制溫度的目的，而且控制時間也不會太長。

6

API	指令碼			運算元										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 運算			

裝置	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								●	●				●				

脈波執行型	16 位元指令	32 位元指令 (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	： 啟動 PID 運算	BIT
SV	： 目標值	DWORD
PV	： 現在值	DWORD
PID_MODE	： PID 控制模式	DWORD
PID_MAN	： PID Auto/Manual 模式	BIT
MOUT_AUTO	： 手動值 (MOUT) 自動更新模式	BIT
CYCLE	： 取樣時間 (ms)	DWORD
Kc_Kp	： 比例項系數	DWORD
Ti_Ki	： 積分項系數 (sec 或 1/sec)	DWORD
Td_Kd	： 微分項系數 (sec)	DWORD
Tf	： 微分作用時間常數 (sec)	DWORD
PID_EQ	： PID 計算式選擇	BIT
PID_DE	： PID 微分項誤差計算選擇	BIT
PID_DIR	： PID 正反向	BIT
ERR_DBW	： ERR 的不作用範圍	DWORD
MV_MAX	： MV 輸出上限值	DWORD
MV_MIN	： MV 輸出下限值	DWORD
MOUT	： MV 手動值	DWORD
BIAS	： 前饋控制輸出值	DWORD
I_MV	： 積分項累計值	DWORD
MV	： 輸出值	DWORD

指令說明：

1. PID 進階運算控制的專用指令，當指令被主機執行時才會進行 PID 運算與處理。PID 表示“比例、積分和微分”。PID 控制在機械設備、氣動設備和電子設備中具有廣泛的應用。
2. 參數設定內容如下：

裝置編號	資料型態	功能	設定範圍	說明
PID_RUN	BOOL	啟動 PID 運算	TRUE：PID 指令開始運算。 FALSE：MV 值清除為 0，PID 不運算。	
SV	REAL	SV 值	單精度浮點數範圍	目標值

裝置編號	資料型態	功能	設定範圍	說明
PV	REAL	PV 值	單精度浮 點數範圍	現在值
PID_MODE	DWORD/DINT	PID 控制模式	0 : 自動控制 · 當 PID_MAN 由 TRUE 轉為 FALSE 時 · MV 值會由當時輸出的 MV 值開始進行自動運算。 1 : 溫度控制專用的自動調整參數功能 · 調整完畢時將自動進入自動控制模式(PID_MODE 改為 0) · 並且填入最適用的 Kc_Kp · Ti_Ki · Td_Kd 及 Tf 等參數。	
PID_MAN	BOOL	PID A/M 模式	TRUE : Manual · MV 值會依 MOUT 值輸出 · 但仍在 MV_MAX 與 MV_MIN 之間 · 當 PID_MODE 為 1 時此設定無效。 FALSE : Auto · MV 值會依 PID 公式計算後輸出 · 輸出值在 MV_MAX 與 MV_MIN 之間。	
MOUT_AUTO	BOOL	MOUT 自動更新模式	TRUE : Auto · MOUT 的值會隨著 MV 值改變。 FALSE : Normal · MOUT 的值不會隨著 MV 值改變。	
CYCLE	DWORD/DINT	取樣時間 (T _s)	1~40,000 (單位: ms)	每次掃描到本指令時 · 就會以 CYCLE 設定的取樣時間來計算 PID (主機並不會依照此數值來自動判斷時間並自動執行) · 並更新輸出值 (MV) 。 CYCLE 小於 1 則內定為 1 · 大於 40,000 時則為 40,000 。 當 PID 指令用於時間中斷時 · 主機會自動以時間中斷的中斷時間來計算 PID · 此時 CYCLE 的設定無效。

裝置編號	資料型態	功能	設定範圍	說明
Kc_Kp	REAL	比例項系數(Kc or Kp · 依 PID_EQ 參數決定使用何種系數)	正數單精度浮點數範圍	P 計算值系數 · 如果小於 0 則 Kc_Kp 將為 0 · 在 Independent 下若 Kc_Kp 等於 0 則表示不使用 P 控制 。
Ti_Ki	REAL	積分項系數 (Ti or Ki · 依 PID_EQ 參數決定使用何種系數)	正數單精度浮點數範圍 (單位: Ti = sec; Ki = 1/sec)	I 計算值系數 · 如果小於 0 則 Ti_Ki 將為 0 · 當 Ti_Ki 等於 0 時則表示不使用 I 控制 。
Td_Kd	REAL	微分項系數(Td or K _d · 依 PID_EQ 參數決定使用何種系數)	正數單精度浮點數範圍 (單位: sec)	D 計算值系數 · 如果小於 0 則 Td_Kd 將為 0 · 當 Td_Kd 等於 0 時則表示不使用 D 控制 。
Tf	REAL	微分作用時間常數 (Tf)	正數單精度浮點數範圍 (單位: sec)	微分作用時間常數 · 如果小於 0 則 Tf 將為 0 · 當 Tf 等於 0 時則表示不使用微分作用時間的控制 (Derivative Smoothing) 。
PID_EQ	BOOL	PID 計算式選擇	TRUE : Dependent Formula FALSE : Independent Formula	
PID_DE	BOOL	PID 微分項誤差計算選擇	TRUE : 使用現在值 (PV) 的變化量來計算微分項的控制值(Derivative of PV) FALSE : 使用偏差量 (E) 的變化量來計算微分項的控制值 (Derivative of E)	
PID_DIR	BOOL	PID 正反向	TRUE : 反向動作 (E=SV-PV) FALSE : 正向動作 (E=PV-SV)	

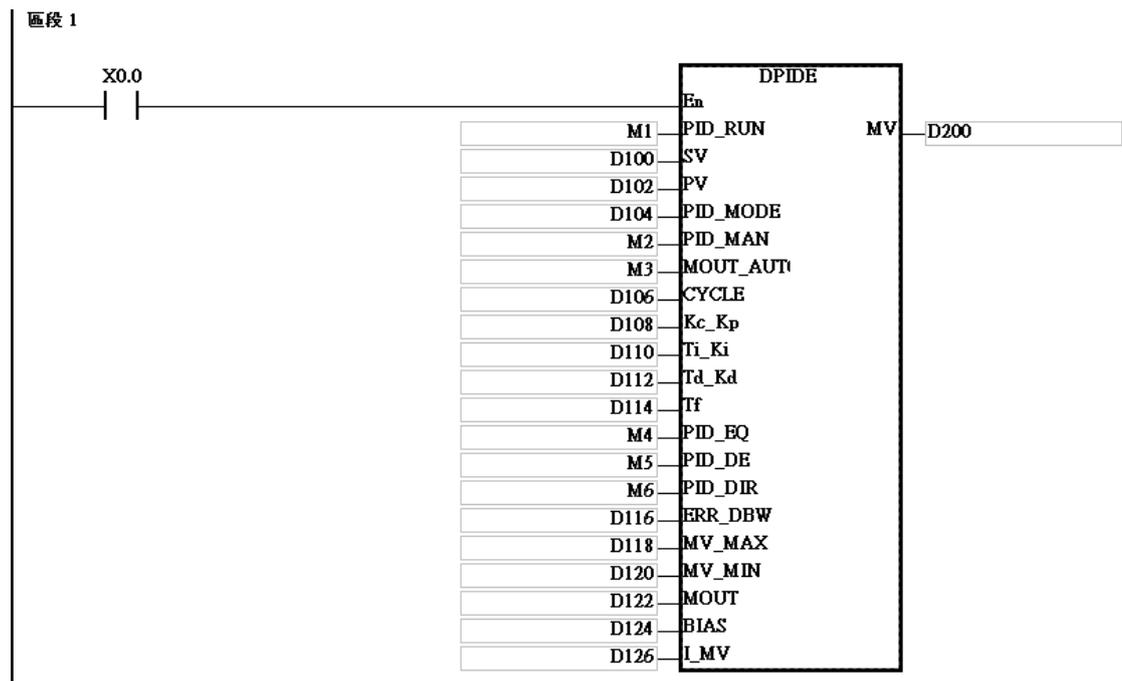
裝置編號	資料型態	功能	設定範圍	說明
ERR_DBW	REAL	偏差量 (E) 不作用範圍	單精度浮點數範圍	偏差量 (E) 等於 SV-PV 或 PV-SV · 當 ERR_DBW 設定為 0 時即表示不啟動此功能 · 否則主機會去檢查這次的 E 值是否小於 ERR_DBW 的絕對值且是否符合 Cross 狀態轉換條件 · 若都有則將 E 值視為 0 之後進行 PID 計算 · 否則依照正常處理將 E 值代入 PID 計算。
MV_MAX	REAL	輸出值 (MV) 飽和上限	單精度浮點數範圍	例：設定 1,000 · 則輸出值 (MV) 大於 1,000 時將以 1,000 輸出 · 此設定值需大於 MV_MIN · 否則主機會自動將 MV_MAX 與 MV_MIN 進行互換。
MV_MIN	REAL	輸出值 (MV) 飽和下限	單精度浮點數範圍	例：設定 -1,000 · 則輸出值 (MV) 小於 -1,000 時將以 -1,000 輸出 · 此設定值需小於 MV_MAX · 否則主機會自動將 MV_MAX 與 MV_MIN 進行互換。
MOUT	REAL	MV 手動值	單精度浮點數範圍	配合 PID_MAN 模式使用 · 當 PID 設定在手動模式 (PID_MAN=True) 時 · 則輸出值 (MV) 會依照手動設定值 (MOUT) 輸出 · 但是仍然在 MV_MAX 與 MV_MIN 之間。

裝置編號	資料型態	功能		設定範圍	說明	
BIAS	REAL	前饋控制輸出值		單精度浮點數範圍	使用於 PID 前饋控制。	
I_MV (連續佔用 10 個 DWord 裝置)	REAL	I_MV	暫存累積的積分值	單精度浮點數範圍	為累積的積分值，通常只供參考用，但是使用者還是可以依需求清除或修改，不過須以單精度浮點數修改之，當 MV 超出 MV_MAX 或 MV_MIN 時，I_MV 值不會再改變	
		I_MV+1	暫存前次 E 值	系統紀錄前次偏差量。		
		I_MV+2 ~ I_MV+5	系統用參數，使用者請勿使用。			
		I_MV+6	系統紀錄前次 PV 值。			
		I_MV+7 ~ I_MV+9	系統用參數，使用者請勿使用。			
MV	REAL	MV 輸出值	MV 值會介於 MV_MAX 與 MV_MIN 之間。			

6

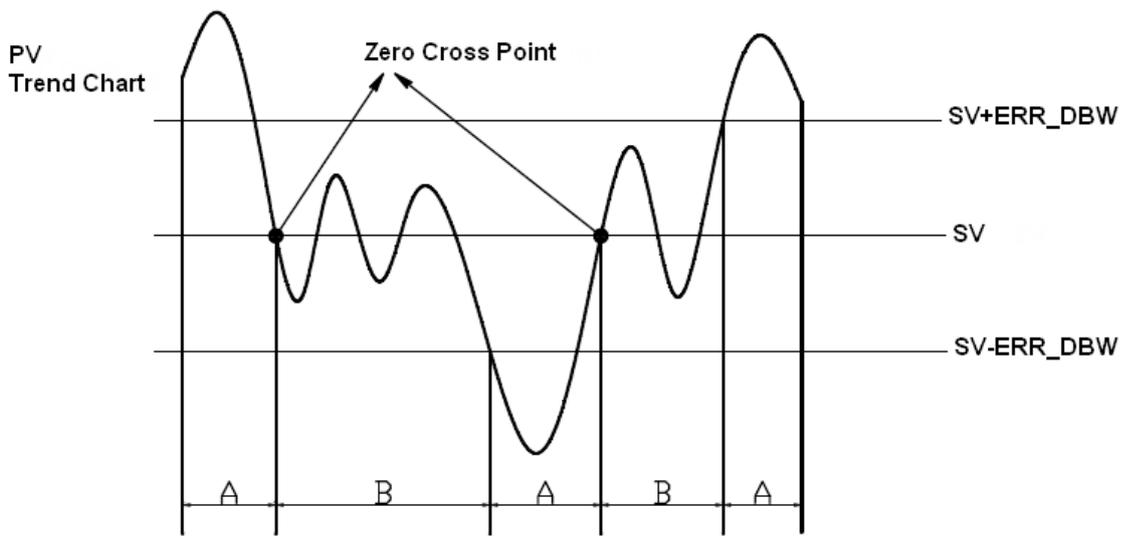
程式範例：

1. 執行 DPIDE 指令前將參數設定完成。
2. X0.0=ON 的時候指令被執行，M1=ON 的時候 DPIDE 指令才開始進行運算，M1=OFF 時 MV 值為 0，MV 值存於 D200 中。X0.0 變成 OFF 時，指令不被執行，之前的資料沒有變化。



補充說明：

1. 指令無使用次數之限制，但是 **I_MV~I_MV+9** 所指定的暫存器號碼不可被其他程序重覆使用。
 2. **I_MV** 佔用 20 個 Word 暫存器，於上述程式例當中 **I_MV** 所佔用的設定區域為 D126~D145。
 3. DPIDE 指令只能使用在週期性工作 (Cyclic Task) 與時間中斷當中，當指令用於時間中斷時，取樣時間 (CYCLE) 會自動以時間中斷的時間為主。
 4. DPIDE 只要被掃描到都會以取樣時間 (CYCLE) 所設定的時間來做 PID 運算並直接輸出 MV 值，並不會自動計算掃描時間是否有到達取樣時間 (CYCLE) 才輸出，但如果本指令使用在時間中斷程式內，取樣時間 (CYCLE) 將與時間中斷程式的間隔時間相同，DPIDE 指令會以時間中斷程式的中斷間隔時間來做 PID 的運算。
 5. DPIDE 的現在值 (PV) 在 PID 執行運算動作前必須是一個穩定值。如果要抓取特殊模組的輸入值作 PID 運算時，請注意這些模組的 A/D 轉換時間。
 6. 當 PV 值進入 ERR_DBW 的範圍時，一開始主機仍會依照 E 值進行 PID 計算，直到 PV 穿過 SV 值時 (Zero Cross Point) 代表 Cross Status 成立，此時會將 E 值視為 0 代入 PID 計算，一直到 PV 值超出 ERR_DBW 的範圍時才會恢復將 E 值代入 PID 計算，若 PID_DE=True 則表示使用 PV 值來進行微分項的計算，則在 Cross Status 條件成立後，主機會將 Delta PV 視為 0 進行 PID 微分項的計算。(Delta PV=當前 PV-前次 PV)
- 例如以下的 PV 趨勢圖中，A 的區段主機會依照正常的 PID 進行計算，而 B 的區段主機會將 E 或 Delta PV 視為 0 進行 PID 計算。



PID 計算公式：

1. 當 **PID_MODE** 控制模式設定為 0 時，為自動控制模式。

- **Independent Formula & Derivative of E (PID_EQ=False & PID_DE=False)**

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

- **Independent Formula & Derivative of PV (PID_EQ=False & PID_DE=True)**

$$CV = K_p E + K_i \int_0^t E dt - K_d \frac{dPV}{dt} + BIAS$$

$$E = SV - PV$$

or

$$CV = K_p E + K_i \int_0^t E dt + K_d \frac{dPV}{dt} + BIAS$$

$$E = PV - SV$$

- **Dependent Formula & Derivative of E (PID_EQ=True & PID_DE=False)**

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dE}{dt} \right] + BIAS$$

$$E = SV - PV \quad \text{or} \quad E = PV - SV$$

6

● **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

$$CV = K_c \left[E + \frac{1}{T_i} \int_0^t E dt + T_d \frac{dPV}{dt} \right] + BIAS$$

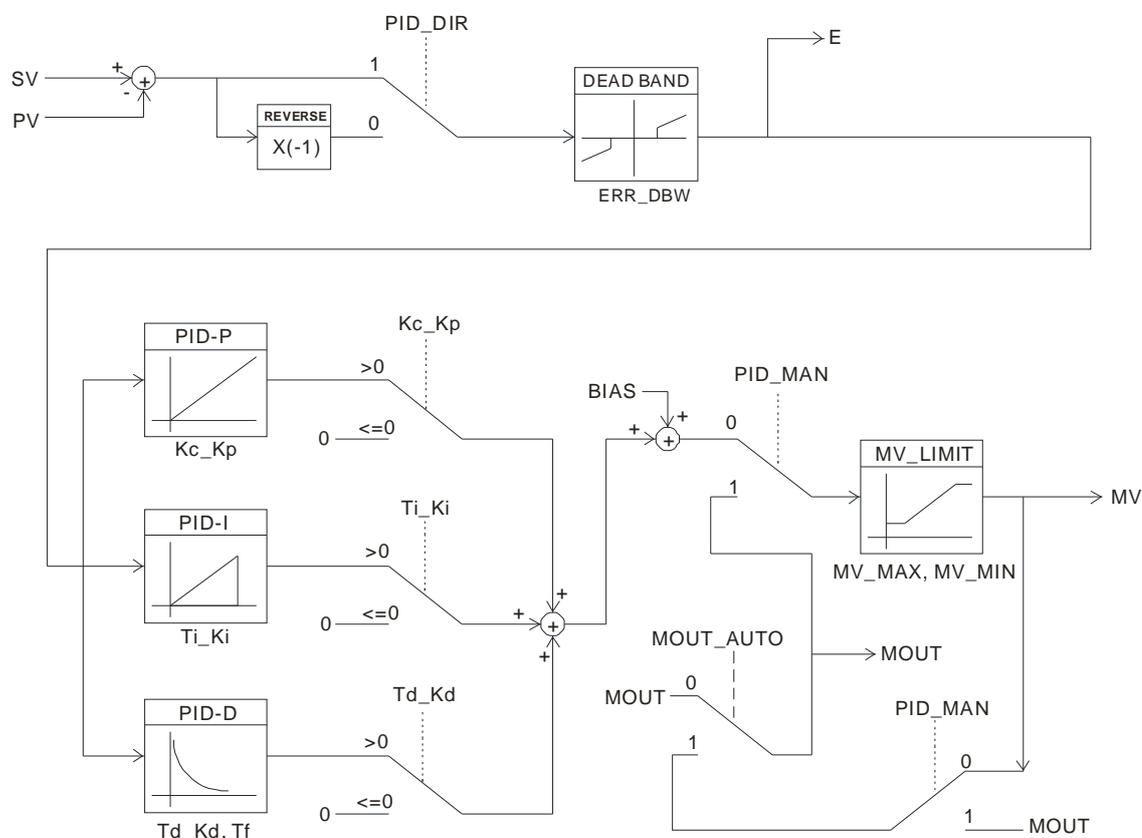
$$E = PV - SV$$

※上述所有公式中的 **CV** 值為 DPIDE 的 **MV** 值

- 當 **PID_MODE** 控制模式選擇為 1，為自動調整模式，當自動調整完成後，**PID_MODE** 會自動變成 0 轉換為自動控制模式。

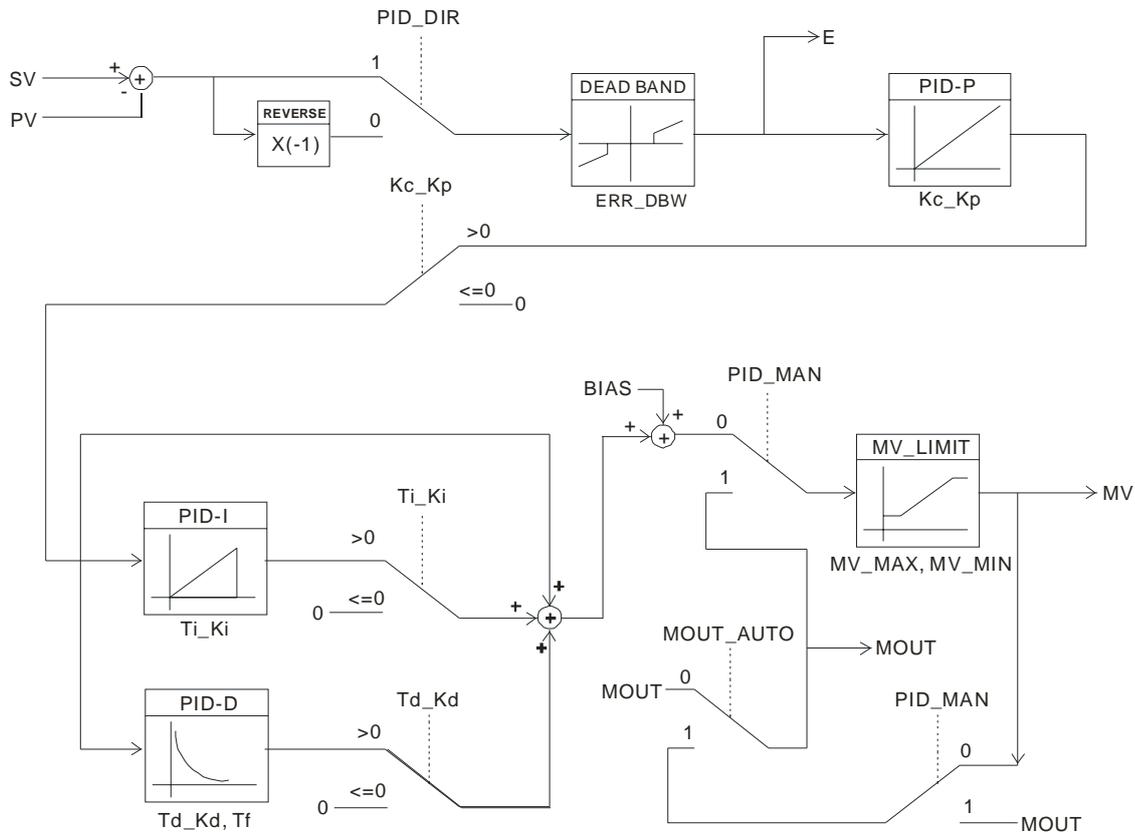
PID 控制方塊圖：

PID Block Diagram (Independent)



6

PID Block Diagram (Dependent)



6

注意事項和建議：

1. 由於可使用 DPIDE 指令的控制環境很多，因此請適當的選取控制功能，例如：當選擇自動調整參數 (PID_MODE=1) 功能時，MV 值會在 MAX/MIN 兩個值之間做切換，所以請盡量勿使用於快速反應的馬達控制環境中，以免造成在自動調整時系統因快速的劇烈變化，而造成對人員危險或損壞系統的現象發生。
2. 使用者於調整 Kc_Kp、Ti_Ki 及 Td_Kd 三個主要參數時 (PID_MODE=0)，請先調整 Kc_Kp 值 (依經驗值設定)，而 Ti_Ki 及 Td_Kd 值先設定為 0，等到調整到大致上可控制時，再依序調整 Ti_Ki 值 (由小到大) 以及 Td_Kd 值 (由小到大)。其中 Kc_Kp 值為 1 則表示 100%，即對偏差值的增益為 1，小於 100%將對偏差值衰減，大於 100%將對偏差值放大。
3. 當使用者選用 (PID_MODE=1) 時，建議請使用在停電保持區之 D 暫存器來儲存參數，以免自動調整過的參數因停電後而消失。經過自動調整過的參數，並不能保證一定適用於每個控制的環境，因此使用者當然可自行修改調整過的參數，不過建議最好只修改 Ti_Ki 或 Td_Kd 數值就好。
4. 本指令動作須配合許多參數值控制，因此請勿隨意設定參數值，以免造成無法控制之現象。

6.9 邏輯操作指令

6.9.1 邏輯操作指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>0800</u>	WAND	DAND	✓	邏輯及 (AND) 運算	7
<u>0801</u>	MAND	–	✓	矩陣及 (AND) 運算	9
<u>0802</u>	WOR	DOR	✓	邏輯或 (OR) 運算	7
<u>0803</u>	MOR	–	✓	矩陣或 (OR) 運算	9
<u>0804</u>	WXOR	DXOR	✓	邏輯互斥或 (XOR) 運算	7
<u>0805</u>	MXOR	–	✓	矩陣互斥或 (XOR) 運算	9
<u>0806</u>	WXNR	DXNR	✓	邏輯互容或 (XNR) 運算	7
<u>0807</u>	MXNR	–	✓	矩陣互容或 (XNR) 運算	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

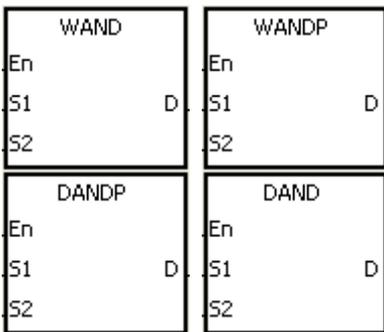
6.9.2 邏輯操作指令說明

API	指令碼			運算元								功能					
0800	W D	AND	P	$S_1 \cdot S_2 \cdot D$								邏輯及 (AND) 運算					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



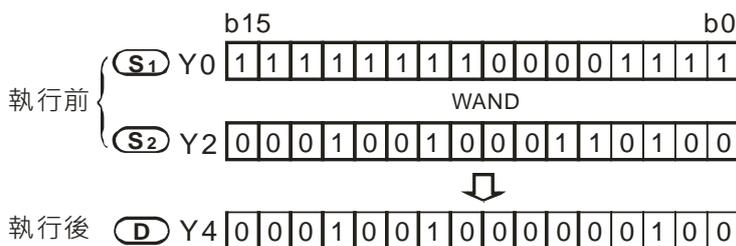
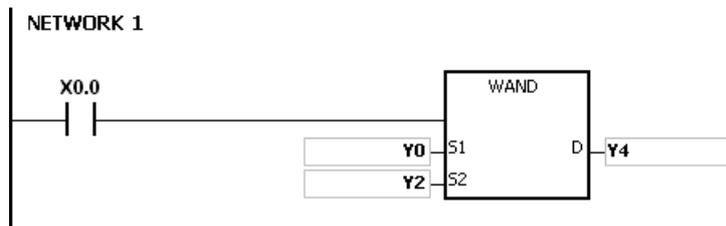
- S_1 : 資料來源裝置 1 Word/Double Word
- S_2 : 資料來源裝置 2 Word/Double Word
- D : 運算結果 Word/Double Word

指令說明：

1. 兩個資料源： S_1 及 S_2 作邏輯的“及” (AND) 運算並將結果存於 D。
2. DAND 才可使用 HC 裝置。
3. 邏輯的“及” (AND) 運算之規則為兩位元均為 1 結果始為 1，否則為 0。

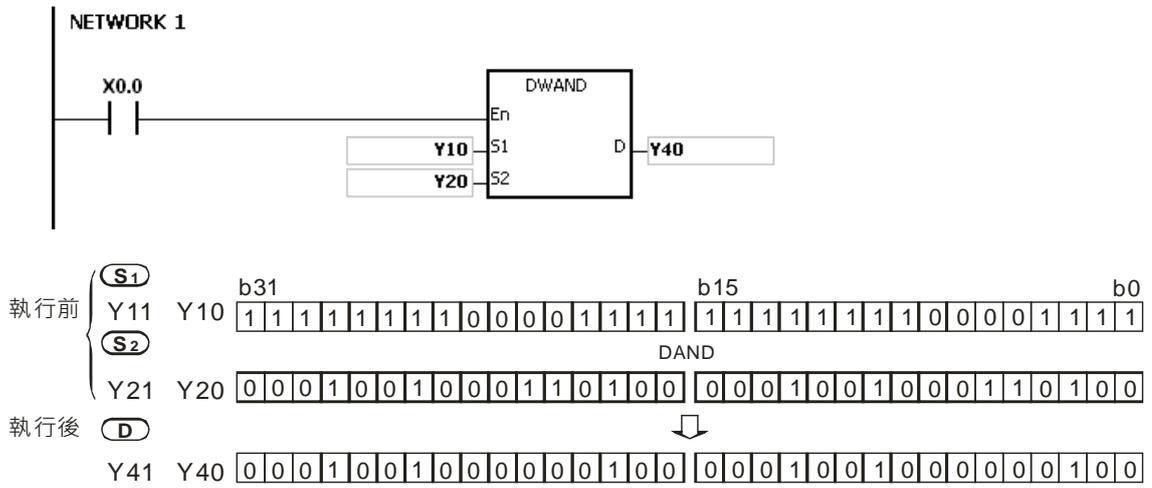
程式範例一：

當 X0.0=ON 時，16 位元 Y0 與 Y2 作 WAND，邏輯及 (AND) 運算，將結果存於 Y4 中。



程式範例二：

當 X0.0=ON 時，32 位元 (Y11、Y10) 與 (Y21、Y20) 作 DAND 邏輯及 (AND) 運算，將結果存於 (Y41、Y40) 中。

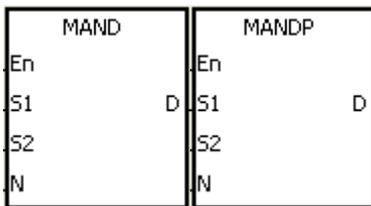


API	指令碼		運算元							功能							
0801		MAND	P	$S_1 \cdot S_2 \cdot D \cdot n$							矩陣及 (AND) 運算						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



- S_1 : 矩陣來源裝置 1 Word
- S_2 : 矩陣來源裝置 2 Word
- D : 運算結果 Word
- n : 陣列長度 Word

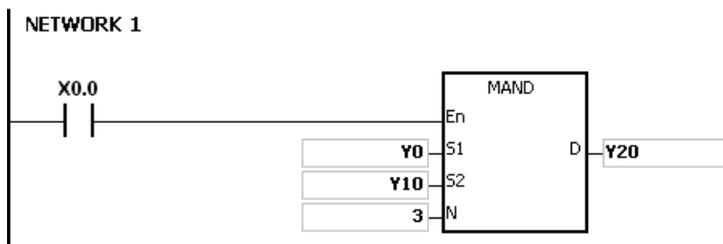
指令說明：

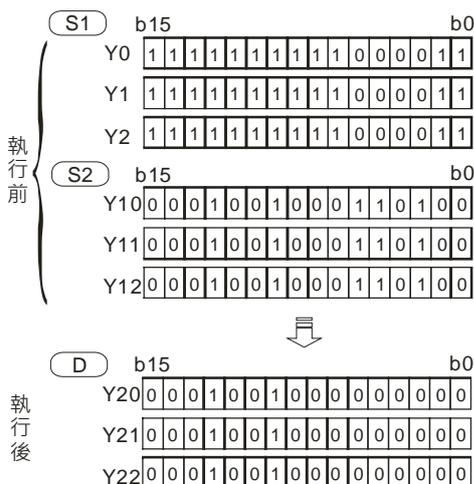
- 兩個矩陣來源依陣列長度 n 將 S_1 及 S_2 作矩陣的“及” (AND) 運算並將結果存於 D。
- 矩陣的“及” (AND) 運算之規則為兩位元均為 1 結果始為 1，否則為 0。
- n 運算元的範圍值為 1~256。

6

程式範例：

當 X0.0=ON 時，16 位元暫存器 Y0~Y2 共 3 列與 16 位元暫存器 Y10~Y12 共 3 列作 MAND · 矩陣及 (AND) 運算，將結果存於 16 位元暫存器 Y20~Y22 共 3 列中。





補充說明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 裝置超出範圍時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. $n<1$ 或 $n>256$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
3. 矩陣指令說明：
 - 矩陣是 1 個以上連續之 16 位元暫存器所組成，組成矩陣之暫存器個數稱為矩陣之長度 n ，一個矩陣共有 $16 \times n$ 個位元（點），其運算單位一次只有一個位元（點）。
 - 矩陣指令是將 $16 \times n$ 個矩陣位元（序號由 $b0 \sim b16n-1$ ）當作一連串單點之集合，而自此集中指定某一單點作運作，而不將之當作數值看待。
 - 矩陣指令主要在處理單點對多點（矩陣）或多點對多點之狀態處理，如搬移、拷貝、比較、搜尋等，為極為方便和重要之應用指令。
 - 在矩陣指令運作中，通常需要有一個 16 位元暫存器來指定矩陣中 $16n$ 個單點之某個單點當作運算對象，此暫存器稱為矩陣之指標 Pr (Pointer)，由使用者於指令中指定，其有效範圍為 $0 \sim 16n-1$ ，分別對應至矩陣中之位元 $b0 \sim b16n-1$ 。
 - 矩陣運作中有左、右位移或旋轉，我們定義高編號者為左，低編號者為右，如下圖示。



- 矩陣寬度 (C) 固定為 16 位元 (bits)。

- Pr : 為矩陣之指標，例如 Pr 值為 15 指到 b15 之位元。

範例：以 Y0 · n=3 構成的矩陣 · 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

範例：以 X0 · n=3 構成的矩陣 · X0=16#37 · X1=16#68 · X2=16#45

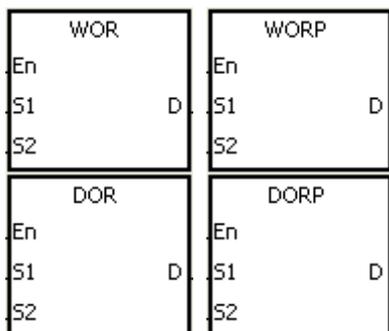
C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	C ₉	C ₈	C ₇	C ₆	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X0
0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X2

API	指令碼			運算元							功能						
0802	W D	OR	P	$S_1 \cdot S_2 \cdot D$							邏輯或 (OR) 運算						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



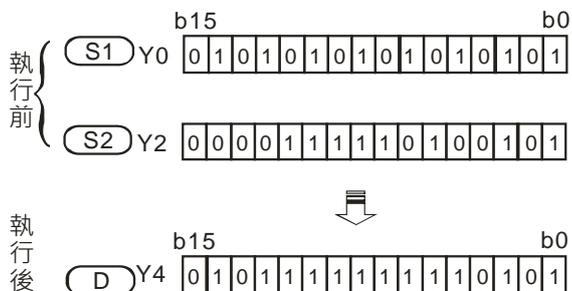
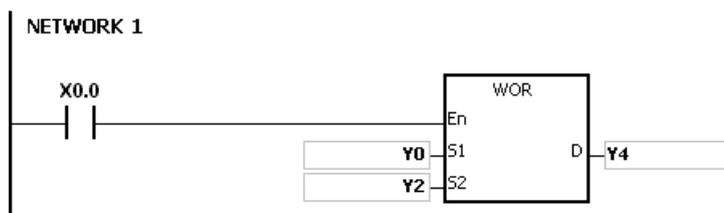
S_1 : 資料來源裝置 1 Word/Double Word
 S_2 : 資料來源裝置 2 Word/Double Word
D : 運算結果 Word/Double Word

指令說明：

- 兩個資料源： S_1 及 S_2 作邏輯的“或” (OR) 運算結果存於 D。
- DOR 才可使用 HC 裝置。
- 邏輯的“或” (OR) 運算之規則兩位元有任一為 1 則結果為 1，兩者均為 0 結果才為 0。

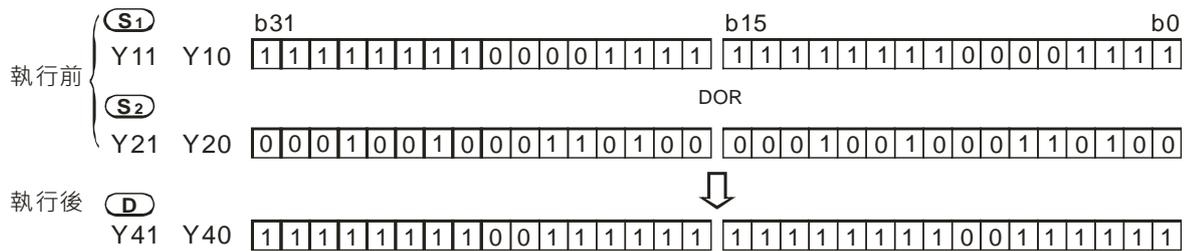
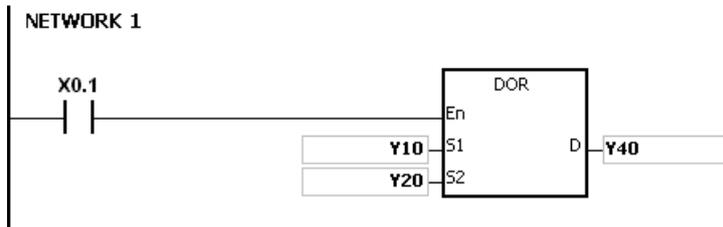
程式範例一：

當 X0.0=ON 時，16 位元 Y0 與 Y2 作 WOR，邏輯或 (OR) 運算，將結果存於 Y4 中。



程式範例二：

當 X0.1=ON 時，32 位元 (Y11、Y10) 與 (Y21、Y20) 作 DOR，邏輯或 (OR) 運算，將結果存於 (Y41、Y40) 中。



6

API	指令碼			運算元								功能					
0803		MOR	P	$S_1 \cdot S_2 \cdot D \cdot n$								矩陣或 (OR) 運算					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

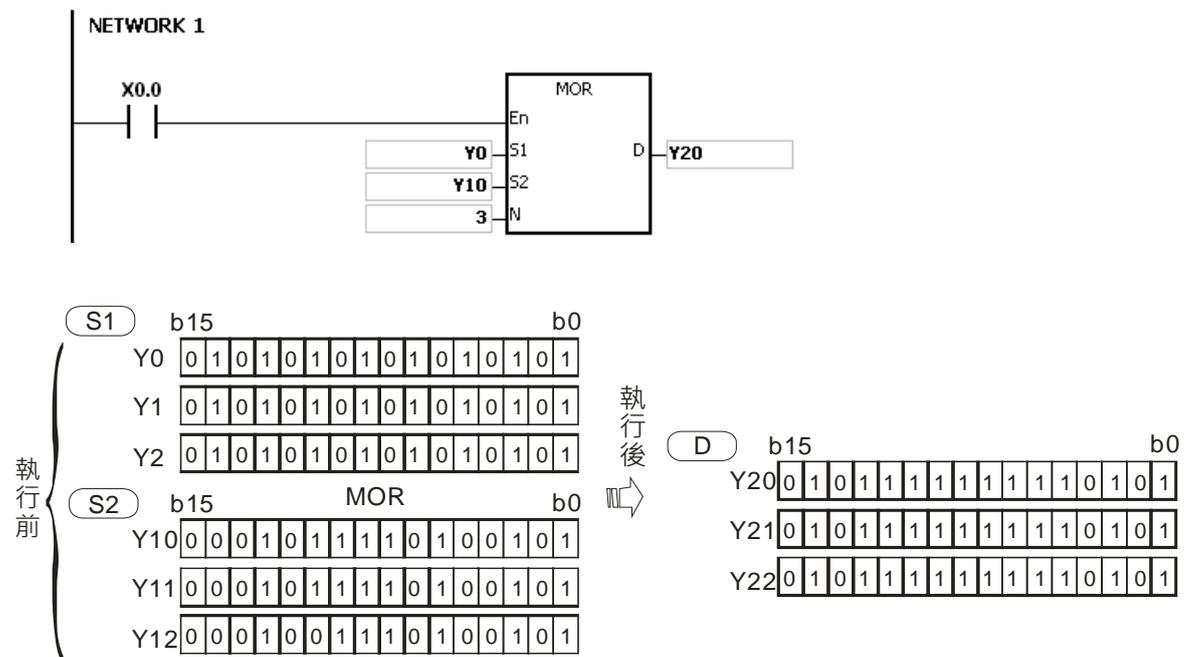


指令說明：

- 兩個矩陣來源依陣列長度 n 將 S_1 及 S_2 作矩陣的“或” (OR) 運算並將結果存於 D 。
- 矩陣的“或” (OR) 運算之規則為兩位元有任一為 1 則結果為 1，兩者均為 0 結果才為 0。
- n 運算元的範圍值為 1~256。

程式範例：

當 X0.0=ON 時，16 位元暫存器 Y0~Y2 共 3 列與 16 位元暫存器 Y10~Y12 共 3 列作 MOR，矩陣或 (OR) 運算，將結果存於 16 位元暫存器 Y20~Y22 共 3 列中。



補充說明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n<1$ 或 $n>256$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

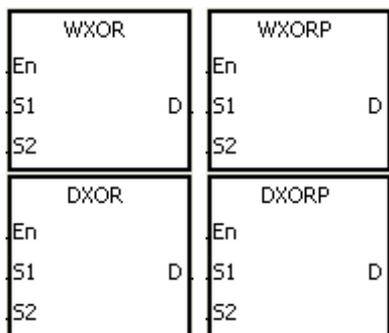
6

API	指令碼			運算元							功能						
0804	W D	XOR	P	$S_1 \cdot S_2 \cdot D$							邏輯互斥或 (XOR) 運算						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



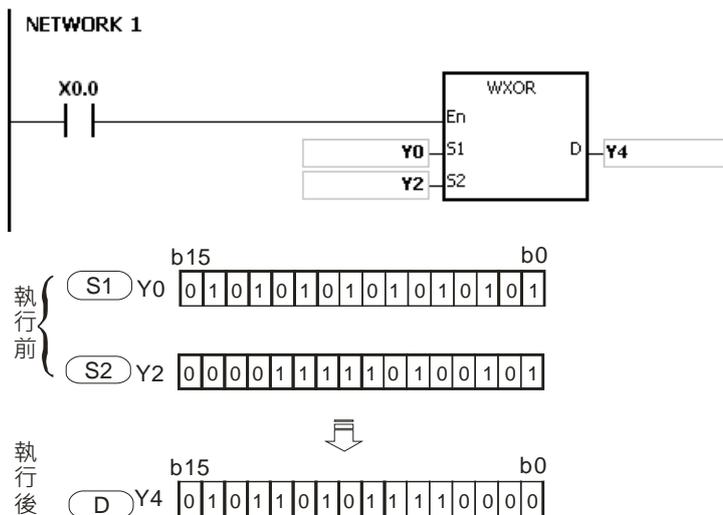
- S_1 : 資料來源裝置 1 Word/Double Word
- S_2 : 資料來源裝置 2 Word/Double Word
- D : 運算結果 Word/Double Word

指令說明：

- 兩個資料源： S_1 及 S_2 作邏輯的“互斥或” (XOR) 運算結果存於 D。
- DXOR 才可使用 HC 裝置。
- 邏輯的“互斥或” (XOR) 運算之規則為兩者相同結果為 0，兩者不同結果為 1。

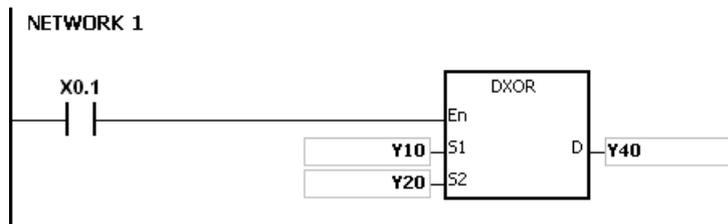
程式範例一：

當 X0.0=ON 時，16 位元 Y0 與 Y2 作 WXOR，邏輯互斥或 (XOR) 運算，將結果存於 Y4 中。



程式範例二：

當 X0.1=ON 時，32 位元 (Y11、Y10) 與 (Y21、Y20) 作 DXOR，邏輯互斥或 (XOR) 運算，將結果存於 (Y41、Y40) 中。



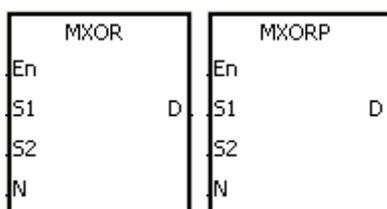
執行前	(S ₁)	Y11	Y10	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	b31	b15	b0	
	(S ₂)	Y21	Y20	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	DXOR			
執行後	(D)	Y41	Y40	1	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1	↓			
				1	1	1	0	1	1	0	1	0	0	1	1	1	0	1	1				

API	指令碼		運算元						功能					
0805	MXOR	P	$S_1 \cdot S_2 \cdot D \cdot n$						矩陣互斥或 (XOR) 運算					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



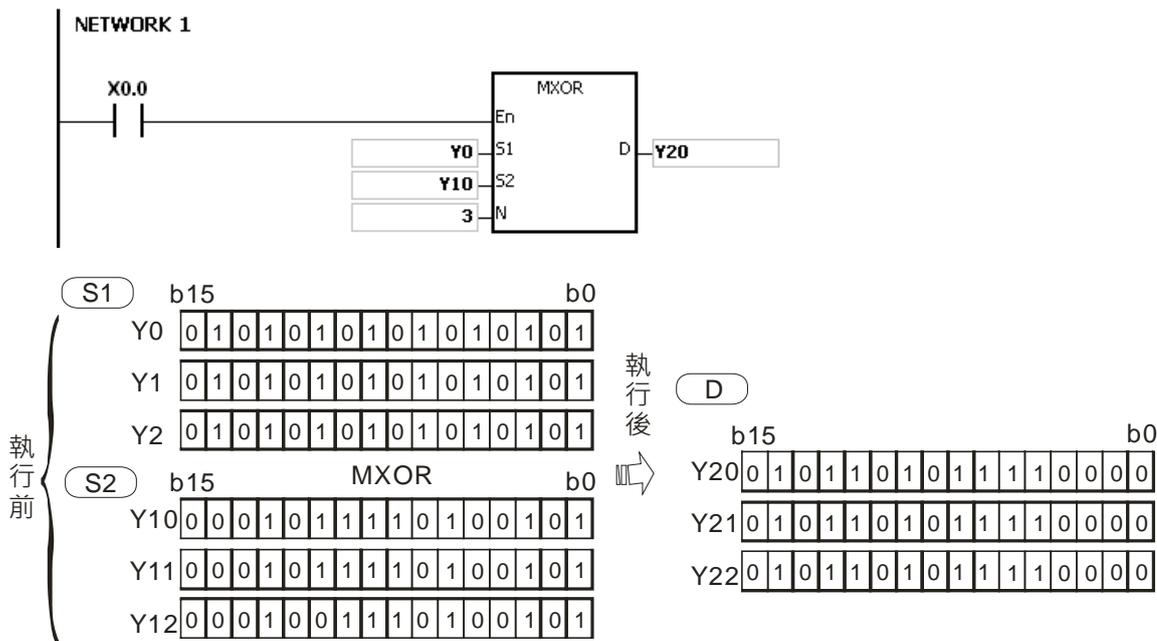
- S_1 : 矩陣來源裝置 1 Word
- S_2 : 矩陣來源裝置 2 Word
- D : 運算結果 Word
- n : 陣列長度 Word

指令說明：

- 兩個矩陣來源依陣列長度 n 將 S_1 及 S_2 作矩陣的“互斥或” (XOR) 運算並將結果存於 D。
- 矩陣的“互斥或” (XOR) 運算之規則為兩者相同結果為 0，兩者不同結果為 1。
- n 運算元的範圍值為 1~256。

程式範例：

當 X0.0=ON 時，16 位元暫存器 Y0~Y2 共 3 列與 16 位元暫存器 Y10~Y12 共 3 列作 MXOR，矩陣互斥或 (XOR) 運算，將結果存於 16 位元暫存器 Y20~Y22 共 3 列中。



補充說明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n<1$ 或 $n>256$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

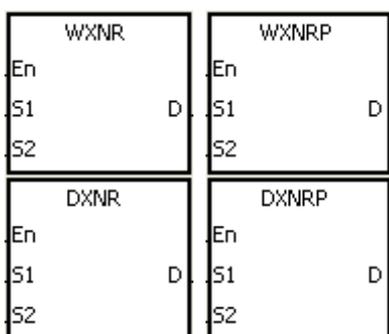
6

API	指令碼			運算元								功能					
0806	W D	XNR	P	$S_1 \cdot S_2 \cdot D$								互容或 (XNR) 運算					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



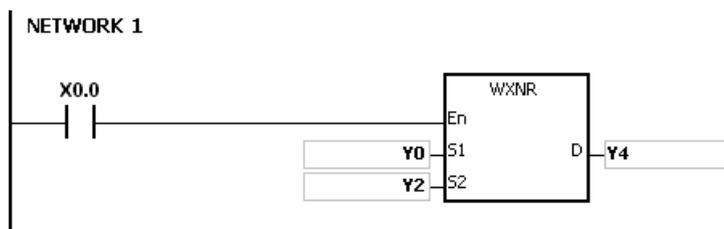
- S_1 : 資料來源裝置 1 Word/Double Word
- S_2 : 資料來源裝置 2 Word/Double Word
- D : 運算結果 Word/Double Word

指令說明：

- 兩個資料源： S_1 及 S_2 作邏輯的“互容或” (XNR) 運算結果存於 D。
- DXNR 才可使用 HC 裝置。
- 邏輯的“互容或” (XNR) 運算之規則為兩者相同結果為 1，兩者不同結果為 0。

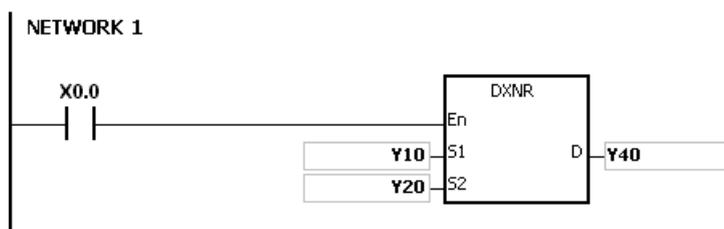
程式範例一：

當 X0.0=ON 時，16 位元 Y0 與 Y2 作 WXNR，邏輯互容或 (XNR) 運算，將結果存於 Y4 中。



程式範例二：

當 X0.0=ON 時，32 位元 (Y11、Y10) 與 (Y21、Y20) 作 DXOR，邏輯互容或 (XNR) 運算，將結果存於 (Y41、Y40) 中。

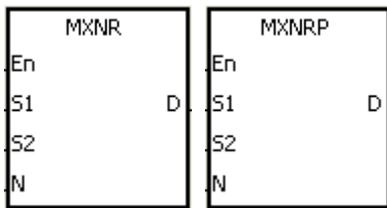


API	指令碼			運算元							功能						
0807		MXNR	P	$S_1 \cdot S_2 \cdot D \cdot n$							矩陣互容或 (XNR) 運算						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



- S_1 : 矩陣來源裝置 1 Word
- S_2 : 矩陣來源裝置 2 Word
- D : 運算結果 Word
- n : 陣列長度 Word

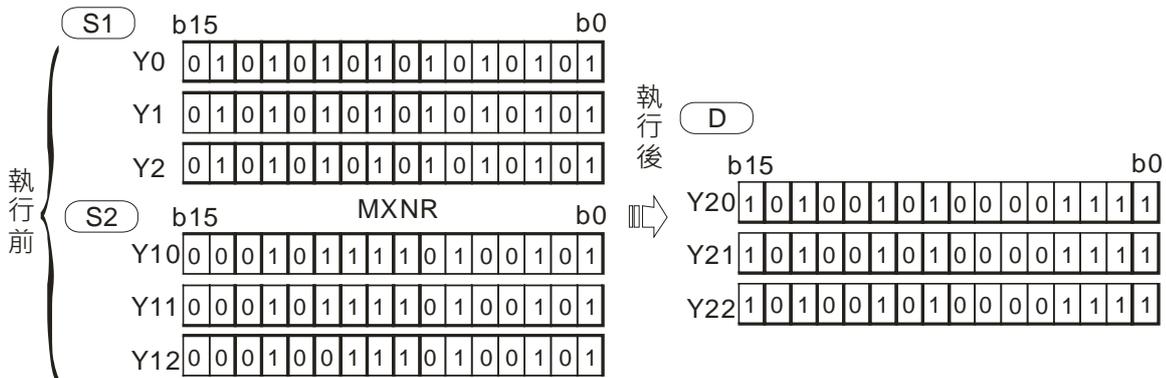
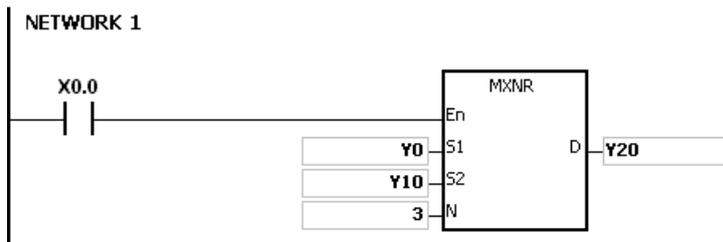
指令說明：

- 兩個矩陣來源依陣列長度 n 將 S_1 及 S_2 作矩陣的互容或 (XNR) 運算並將結果存於 D。
- 矩陣的互容或 (XNR) 運算之規則為兩者相同結果為 1，兩者不同結果為 0。
- n 運算元的範圍值為 1~256。

6

程式範例：

當 X0.0=ON 時，16 位元暫存器 Y0~Y2 共 3 列與 16 位元暫存器 Y10~Y12 共 3 列作 MXNR，矩陣 XNR 運算，將結果存於 16 位元暫存器 Y20~Y22 共 3 列中。

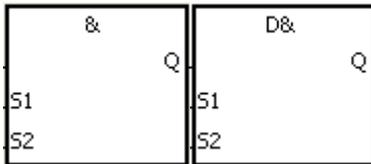


補充說明：

1. S_1+n-1 、 S_2+n-1 、 $D+n-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n<1$ 或 $n>256$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
0809~0811	D	LD #		$S_1 \cdot S_2$								接點型態邏輯運算 LD #					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●				●	●	●	●		●	○	●	○	○		
S_2	●	●				●	●	●	●		●	○	●	○	○		
										脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)					
										-	AH500	AH500					

符號：



以 LD&跟 DLD&為例

S_1 : 資料來源裝置 1 Word/Double Word
 S_2 : 資料來源裝置 2 Word/Double Word

指令說明：

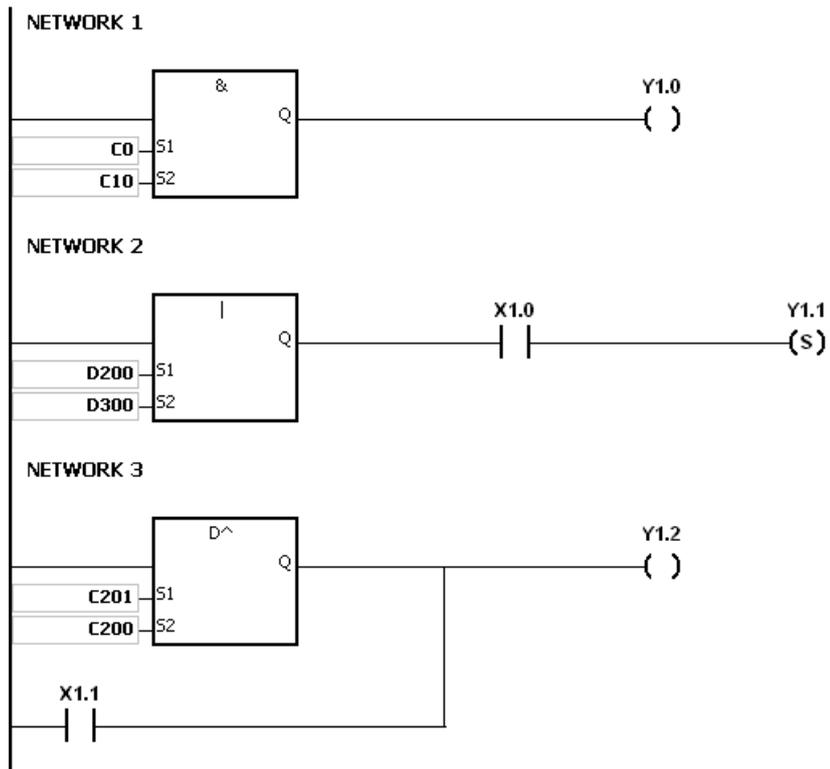
- S_1 與 S_2 之內容作比較的指令，比較結果不為 0 時，該指令導通，比較結果為 0 時，該指令不導通。
- DLD # 才可使用 HC 裝置。
- LD # 的指令可直接與母線連接使用。

API No.	16-bit 指令	32-bit 指令	導通條件	非導通條件
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^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- & : 邏輯的“及” (AND) 運算。
- | : 邏輯的“或” (OR) 運算。
- ^ : 邏輯的“互斥或” (XOR) 運算。

程式範例：

- C0 與 C10 的內容做邏輯的“及” (AND) 運算不等於 0 時，Y1.0=ON。
- D200 與 D300 的內容做邏輯的“或” (OR) 運算不等於 0 時，而且 X1.0=ON 的時候，Y1.1=ON 並保持住。
- C201 與 C200 的內容做邏輯的“互斥或” (XOR) 運算不等於 0 時或是 X1.1=ON 的時候，Y1.2=ON。

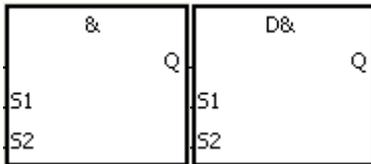


補充說明：

S₁、S₂ 裝置不合法，接點不導通，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能					
0812~0814	D	AND #		$S_1 \cdot S_2$								接點型態邏輯運算 AND #					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●				●	●	●	●		●	○	●	○	○		
S_2	●	●				●	●	●	●		●	○	●	○	○		
										脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)					
										-	AH500	AH500					

符號：



以 AND&跟 DAND&為例

S_1 : 資料來源裝置 1 Word/Double Word
 S_2 : 資料來源裝置 2 Word/Double Word

指令說明：

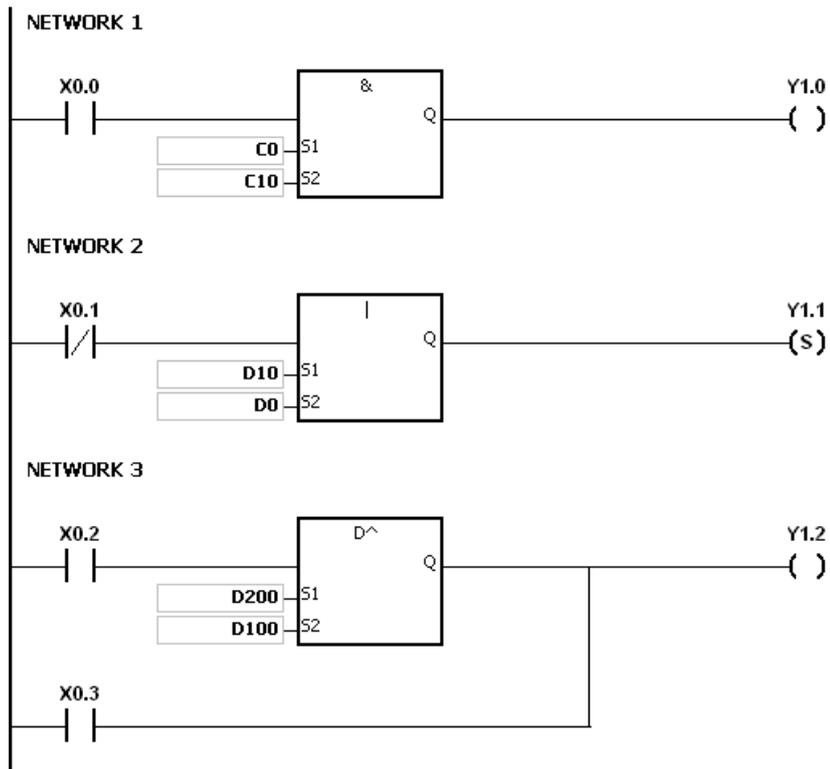
- S_1 與 S_2 之內容作比較的指令，比較結果不為 0 時，該指令導通，比較結果為 0 時，該指令不導通。
- DAND # 才可使用 HC 裝置。
- AND # 的指令是與接點串接的運算指令。

API No.	16-bit 指令	32-bit 指令	導通條件	非導通條件
0812	AND&	DAND&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0813	AND	DAND	$S_1 S_2 \neq 0$	$S_1 S_2 = 0$
0814	AND^	DAND^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- & : 邏輯的“及” (AND) 運算。
- | : 邏輯的“或” (OR) 運算。
- ^ : 邏輯的“互斥或” (XOR) 運算。

程式範例：

- 當 X0.0=ON 時且 C0 與 C10 的內容做邏輯的“及” (AND) 運算不等於 0 時，Y1.0=ON。
- 當 X0.1=OFF 時且 D10 與 D0 的內容做邏輯的“或”(OR)運算不等於 0 時，Y1.1=ON 並保持住。
- 當 X0.2=ON 時且 32 位元暫存器 (D200 · D201) 與 32 位元暫存器 (D100 · D101) 的內容做邏輯的“互斥或” (XOR) 運算不等於 0 時或是 X0.3=ON 的時候，Y1.2=ON。

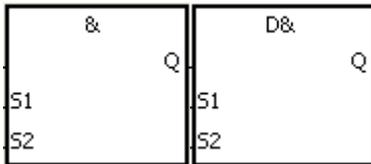


補充說明：

S₁、S₂ 裝置不合法，接點不導通，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能						
0815~0817	D	OR #		$S_1 \cdot S_2$								接點型態邏輯運算 OR #						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S_1	●	●				●	●	●	●		●	○	●	○	○			
S_2	●	●				●	●	●	●		●	○	●	○	○			
										脈波執行型			16 位元指令 (5 steps)			32 位元指令 (5 steps)		
										-			AH500			AH500		

符號：



以 OR&跟 DOR&為例

S_1 : 資料來源裝置 1 Word/Double Word
 S_2 : 資料來源裝置 2 Word/Double Word

指令說明：

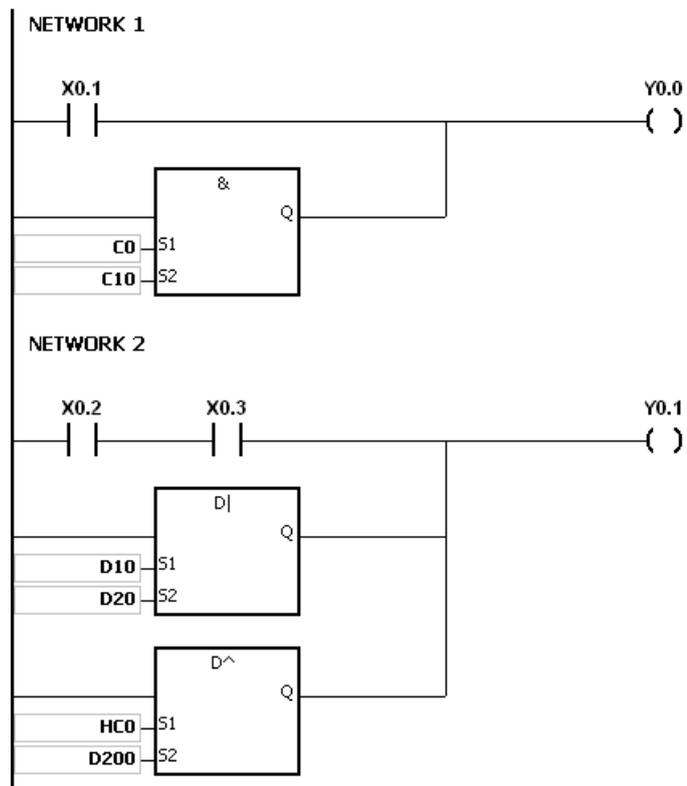
- S_1 與 S_2 之內容作比較的指令，比較結果不為 0 時，該指令導通，比較結果為 0 時，該指令不導通。
- DOR # 才可使用 HC 裝置。
- OR # 的指令是與接點並接的運算指令。

API No.	16-bit 指令	32-bit 指令	導通條件	非導通條件
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^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

- & : 邏輯的“及” (AND) 運算。
- | : 邏輯的“或” (OR) 運算。
- ^ : 邏輯的“互斥或” (XOR) 運算。

程式範例：

- 當 X0.1=ON 時或 C0 與 C10 的內容做邏輯的“及” (AND) 運算不等於 0 時，Y0.0=ON。
- 當 X0.2 及 X0.3 都等於 ON 的時候，或者是 32-bit 暫存器(D10 · D11)與 32 位元暫存器(D20 · D21) 的內容做邏輯的“或” (OR) 運算不等於 0 時，或者是 32 位元計數器 HC0 與 32 位元暫存器 (D200 · D201) 的內容做邏輯的“互斥或” (XOR) 運算不等於 0 時，Y0.1=ON。



補充說明：

S₁、S₂ 裝置不合法，接點不導通，SM0=ON，錯誤碼 SR0=16#2003。

6.10 旋轉指令

6.10.1 旋轉指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>0900</u>	ROR	DROR	✓	右旋轉	5
<u>0901</u>	RCR	DRCR	✓	附進位旗標右旋轉	5
<u>0902</u>	ROL	DROL	✓	左旋轉	5
<u>0903</u>	RCL	DRCL	✓	附進位旗標左旋轉	5
<u>0904</u>	MBR	—	✓	矩陣位元旋轉	7

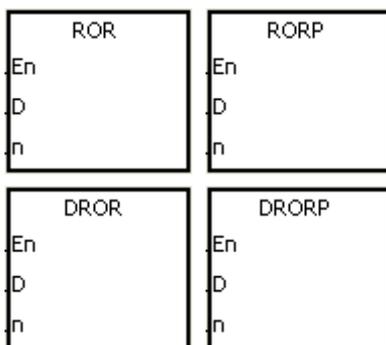
6.10.2 旋轉指令說明

API	指令碼			運算元								功能					
0900	D	ROR	P	D · n								右旋轉					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：



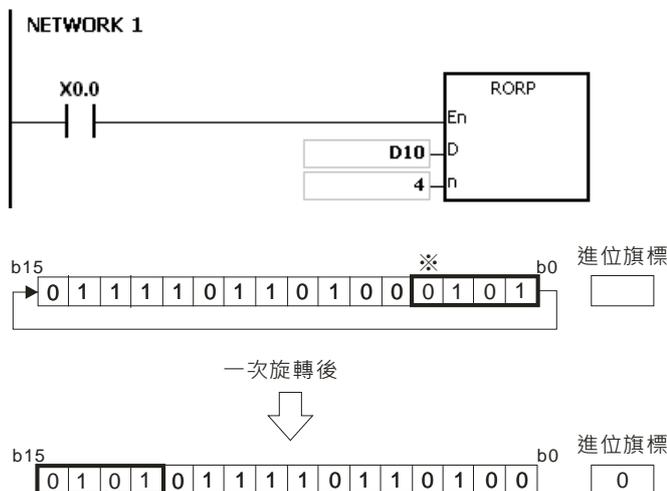
D：欲旋轉之裝置 Word/Double Word
n：一次旋轉之位元數 Word/Double Word

指令說明：

1. 將 **D** 所指定的裝置內容一次向右旋轉 **n** 個位元。
2. **DROR** 才可以使使用 **HC** 裝置。
3. 16 位元指令 n=1~16，32 位元指令 n=1~32。
4. 本指令一般都是使用脈波執行型指令 (**RORP**、**DRORP**)。

程式範例：

當 X0.0 從 OFF→ON 變化時，D10 的 16 個位元以 4 個位元為一組往右旋轉，如下圖所示標明※的位元內容被傳送至進位旗標信號 SM602 內。



補充說明：

1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

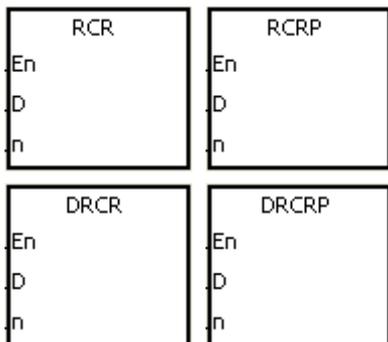
6

API	指令碼			運算元								功能					
0901	D	RCR	P	D · n								附進位旗標右旋轉					

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：



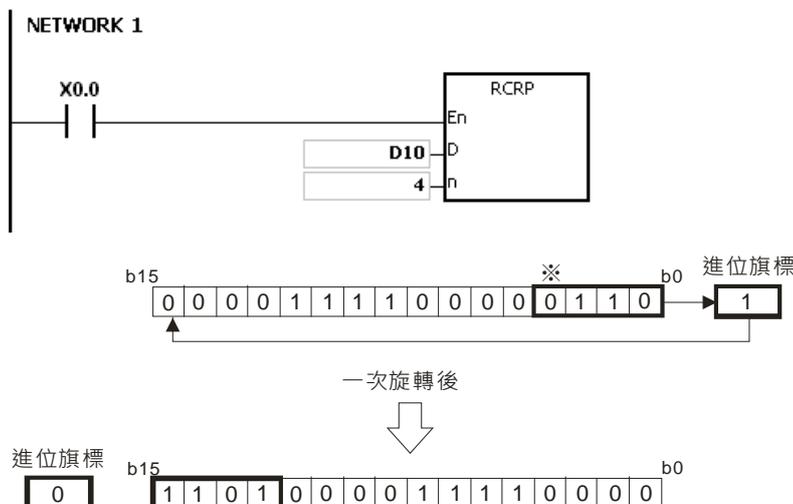
D：欲旋轉之裝置 Word/Double Word
n：一次旋轉之位元數 Word/Double Word

指令說明：

1. 將 **D** 所指定的裝置內容連同進位旗標 **SM602**，一次向右旋轉 **n** 個位元。
2. 32 位元指令才可以使用 **HC** 裝置。
3. 16 位元指令 n=1~16，32 位元指令 n=1~32。
4. 本指令一般都是使用脈波執行型指令 (RCRP、DRCRP)。

程式範例：

當 X0.0 從 OFF→ON 變化時，D10 的 16 個位元連同進位旗標 SM602 共 17 個位元以 4 個位元為一組往右旋轉，如下圖所示標明※的位元內容被傳送至進位旗標信號 SM602 內。



補充說明：

1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

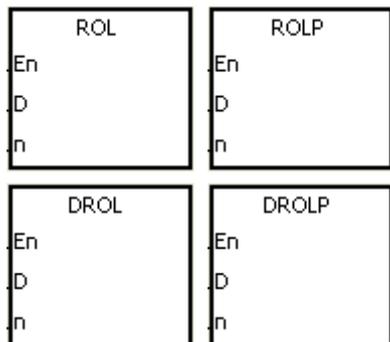
6

API	指令碼			運算元								功能					
0902	D	ROL	P	D · n								左旋轉					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：



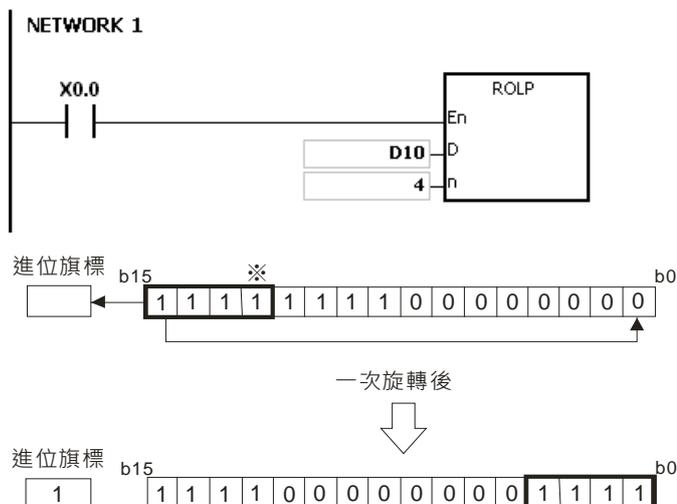
D：欲旋轉之裝置 Word/Double Word
n：一次旋轉之位元數 Word/Double Word

指令說明：

1. 將 **D** 所指定的裝置內容一次向左旋轉 **n** 個位元。
2. 32 位元指令才可以使用 HC 裝置。
3. 16 位元指令 n=1~16，32 位元指令 n=1~32。
4. 本指令一般都是使用脈波執行型指令 (ROLP、DROLP)。

程式範例：

當 X0.0 從 OFF→ON 變化時，D10 的 16 個位元以 4 個位元一組往左旋轉，如下圖所示標明※的位元內容被傳送至進位旗標信號 SM602 內。



補充說明：

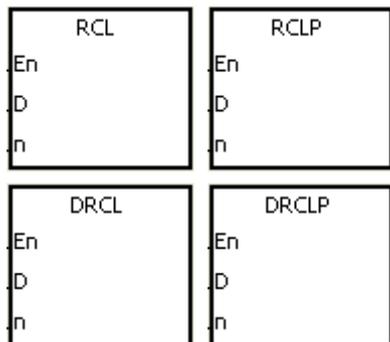
1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

6

API	指令碼			運算元								功能					
0903	D	RCL	P	D · n								附進位旗標左旋轉					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●	●	●	●		●	○	●				
n	●	●						●	●		●		●	○	○		

脈波執行型	16 位元指令(5 steps)	32 位元指令(5 steps)
AH500	AH500	AH500

符號：



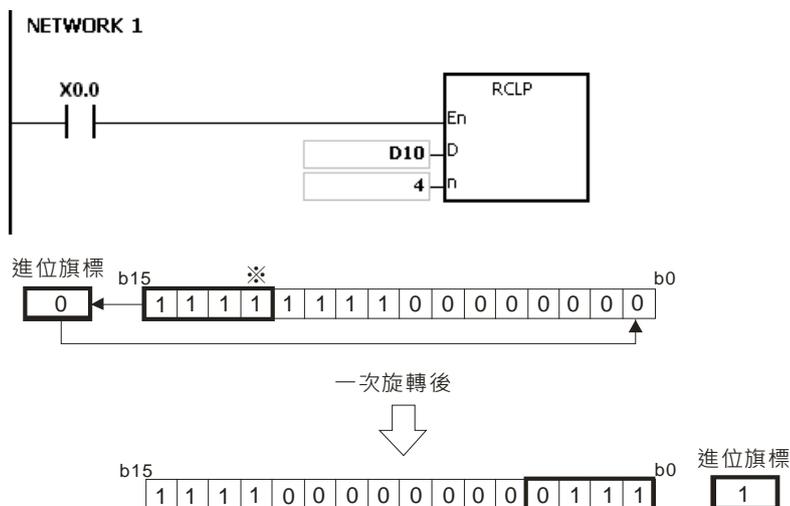
D：欲旋轉之裝置 Word/Double Word
n：一次旋轉之位元數 Word/Double Word

指令說明：

1. 將 **D** 所指定的裝置內容連同進位旗標 **SM602**，一次向左旋轉 **n** 個位元。
2. 32 位元指令才可以使用 **HC** 裝置。
3. 16 位元指令 n=1~16，32 位元指令 n=1~32。
4. 本指令一般都是使用脈波執行型指令 (RCLP、DRCLP)。

程式範例：

當 X0.0 從 OFF→ON 變化時，D10 的 16 個位元連同進位旗標 SM602 共 17 個位元以 4 個位元一組往左旋轉，如下圖所示標明※的位元內容被傳送至進位旗標信號 SM602 內。



補充說明：

1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

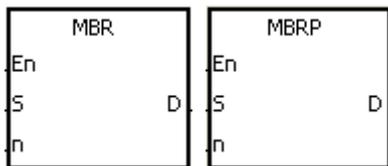
6

API	指令碼			運算元								功能				
0904		MBR	P	S · D · n								矩陣位元旋轉				

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：



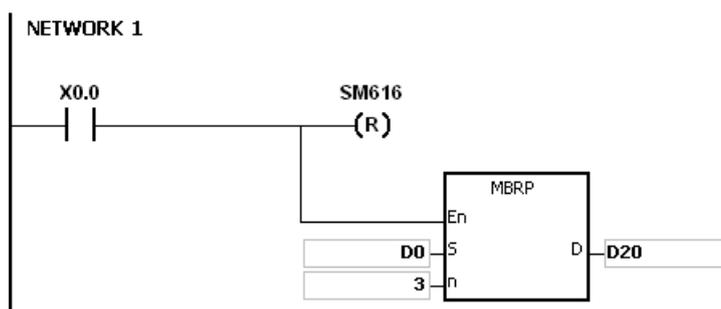
- S** : 矩陣來源裝置 Word
- D** : 運算結果 Word
- n** : 陣列長度 Word

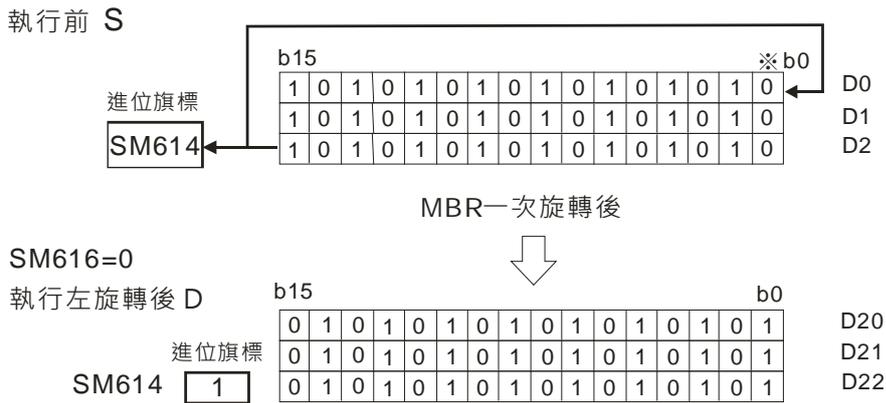
指令說明：

1. 矩陣來源依陣列長度 **n** 將 **S** 矩陣位元做左右旋轉控制。SM616=0 決定矩陣位元左旋轉，SM616=1 決定矩陣位元右旋轉。每次旋轉一位元，因旋轉造成之空位（左旋時為 b0，右旋時為 b16n-1）由旋出位元（左旋時為 b16n-1 右旋時為 b0）狀態填補。將結果存入 **D**。旋出位元不但用以填補前述之空位，同時並將之狀態送到進位旗標 SM614。
2. n 運算元值的範圍為 1~256。
3. 本指令一般都是使用脈波執行型指令（MBRP）。

程式範例一：

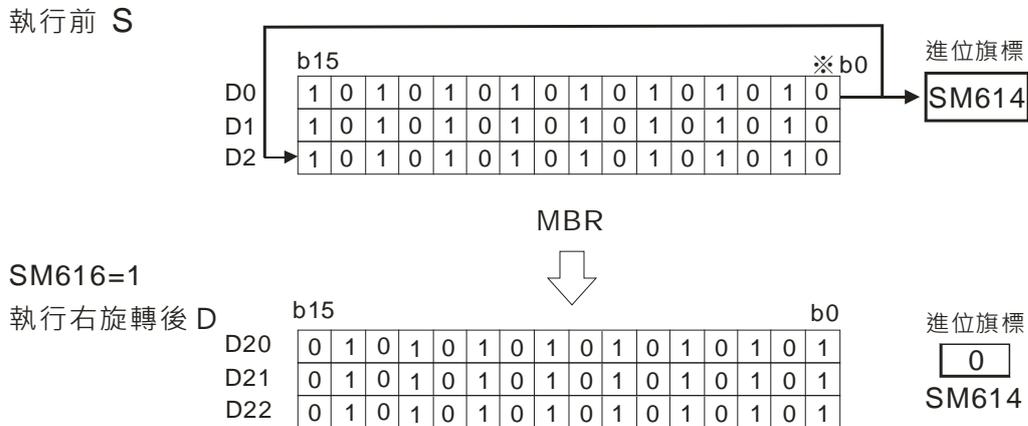
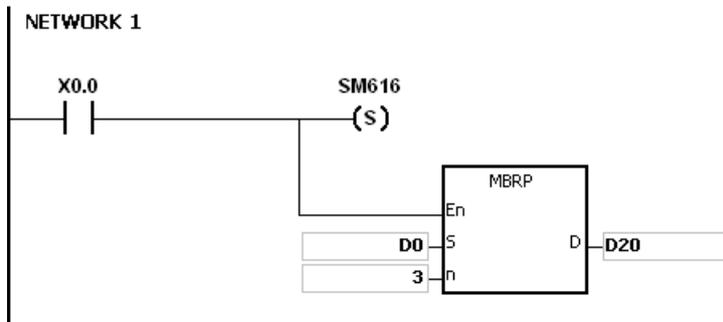
當 X0.0=ON 時 SM616=OFF 決定矩陣為左旋轉，16 位元暫存器 D0~D2 矩陣旋轉，將結果存於 16 位元暫存器 D20~D22 矩陣中，下圖標明*位元的內容將傳送至進位旗標 SM614。





程式範例二：

當 X0.0=ON 時，SM616=ON 作矩陣右旋轉，16 位元暫存器 D0~D2 矩陣作右旋轉，將結果存於 16 位元暫存器 D20~D22 矩陣中，下圖標明※位元的内容將傳送至進位旗標 SM614。



補充說明：

1. S+n-1、D+n-1 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. n<1 或 n>256 時，指令不執行，SM0=ON，SR0 錯誤碼=16#200B。
3. 旗標信號說明：
 - SM614：矩陣旋轉位移輸出進位旗標
 - SM616：矩陣旋轉位移方向旗標

6.11 基本指令

6.11.1 基本指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1000</u>	RST	–	–	接點或暫存器清除	3
<u>1001</u>	TMR	–	–	16 位元計時器	5
<u>1002</u>	TMRH	–	–	16 位元計時器	5
<u>1003</u>	CNT	–	–	16 位元計數器	5
<u>1004</u>	–	DCNT	–	32 位元計數器	5

6.11.2 基本指令說明

API	指令碼			運算元								功能					
1000		RST		D								接點或暫存器清除					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●	●	●	●	●	●	●	●	○	●				
		脈波執行型		16 位元指令 (3 steps)				32 位元指令									
		-		AH500				-									

符號：

元件
D : 清除元件編號
Bit/Word
—(R)

指令說明：

- 當 RST 指令被驅動，其指定的元件的動作如下：

元件	狀態
Bit	線圈及接點都會被設定為 OFF。
T · C · HC	目前計時或計數值會被設為 0，且線圈及接點都會被設定為 OFF。
Word	內容值會被設為 0。

- 若 RST 指令沒有被執行，其指定元件的狀態保持不變。
- 本指令有支援 DY 裝置。

程式範例：

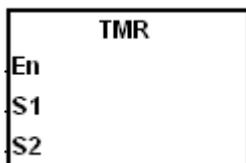
當 X0.0=ON 時，Y0.5 會被設為 OFF。



6

API	指令碼			運算元								功能						
1001	TMR			$S_1 \cdot S_2$								16 位元計時器						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S_1					●													
S_2	●	●						●	●		●		●	○	○			
											脈波執行型		16 位元指令(5 steps)		32 位元指令			
											-		AH500		-			

符號：



S_1 : 計時器編號 Word

S_2 : 計時器設定值 Word

指令說明：

詳細內容說明請參考 API1002 TMRH 指令的指令說明。

API	指令碼				運算元								功能				
1002		TMRH			S ₁ · S ₂								16 位元計時器				
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁					○												
S ₂	○	○						○	○		○		○	○	○		
							脈波執行型			16 位元指令(5 steps)			32 位元指令				
							-			AH500			-				

符號：



S₁ : 計時器編號 Word
 S₂ : 計時器設定值 Word

指令說明：

1. TMR 指令所指定之 T 計時器以 100ms 為單位計時，TMRH 指令所指定之 T 計時器以 1ms 為單位計時。
2. 副程序專用 T 計時器為 T1920~T2047。
3. TMR 跟 TMRH 的計時器內容值範圍為 0~32767。
4. 在程式中同一個 T 計時器如果重覆使用 (包含使用在不同指令 TMR、TMRH 中)，則以最快到達的為主。
5. 使用 TMR 指令時，只要在 T 之前加上一個 S，就會變成累積計時器 ST 裝置，表示目前的 T 變成累積計時器，則條件接點 OFF 時 T 的值不會被清除，條件接點=ON 的時候，T 由目前的值開始累積計時。
6. 在程式中同一個 T 計時器如果重覆使用，其中一個條件接點 OFF 時，則 T 會 OFF。
7. 在程式中同一個 T 計時器如果重覆使用為 T 與 ST，其中一個條件接點 OFF 時則 T 會 OFF。
8. 當 T 計時器 ON->OFF 且條件式為 ON 時，T 計時值歸零並重新計數。
9. 當 TMR 指令執行時，其所指定的計時器線圈受電，計時器開始計時，當到達所指定的定時值(計時值>=設定值)，其接點動作如下：

NO (Normally Open) 接點	開路
NC (Normally Closed) 接點	閉合

程式範例一：

當 X0.0 為 ON 時，計時器 T0 會載入設定值 50，當計時到達時 T0=50，T0 接點會 ON。



程式範例二：

當 X0.0 為 ON 時，計時器 T0 會載入設定值 50，當計時器數到 T0=25 時，將 X0.0 OFF=>ON，計時器會從 T0=25 繼續數到 T0=50，且 T0 接點會 ON。



程式範例三：

當 X0.0=ON 時，計時器 T5 會載入設定值 1000，當計時到達時 T5=1000，T5=ON。



程式範例四：

當 X0.0 為 ON 時，計時器 T5 會載入設定值 1000，當計時器數到 T5=500 時，將 X0.0 OFF=>ON，計時器會從 T5=500 繼續數到 T5=1000，且 T5 接點會 ON。

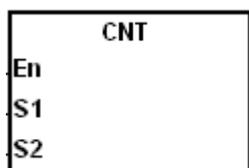


補充說明：

S₁ 運算元，若使用 ISPSOft 宣告，則資料型態為 TIMER。

API	指令碼			運算元							功能						
1003	CNT			S ₁ · S ₂							16 位元計數器						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S ₁						○											
S ₂	○	○						○	○		○		○	○	○		
脈波執行型											16 位元指令(5 steps)			32 位元指令			
-											AH500			-			

符號：



S₁ : 計數器編號 Word

S₂ : 計數器設定值 Word

指令說明：

- 當 CNT 指令由 OFF->ON 執行，表示所指定的計數器線圈由失電→受電，則該計數器計數值加 1。當計數到達所指定的數值 (計數值=設定值)，其接點動作如下：

NO (Normally Open) 接點	開路
NC (Normally Closed) 接點	閉合

- 當計數到達之後，若再有計數脈波輸入，其接點及計數值均保持不變，若要重新計數或作清除的動作，請利用 RST 指令。

6

程式範例：

當 SM408 的狀態第一次 ON 時，計數器 C0 會載入設定值 10 並開始計數，當 SM408 的狀態經過 10 次 OFF=>ON 的變化後，計數器 C0 的值計數到達 C0=10，且 C0 的接點為 ON。

當 C0 為 ON 之後，雖然 SM408 繼續 OFF=>ON，但 C0 的值已經計數到達就不會再累加。

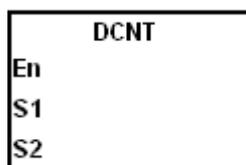


補充說明：

S₁ 運算元，若使用 ISPSOFT 宣告，則資料型態為 COUNTER。

API	指令碼			運算元								功能					
1004		DCNT		$S_1 \cdot S_2$								32 位元計數器					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1							○										
S_2	○	○						○	○		○		○	○	○		
脈波執行型											16 位元指令			32 位元指令 (5 steps)			
-											-			AH500			

符號：



S_1 ：計數器編號

Double Word

S_2 ：計數器設定值

Double Word

指令說明：

- DCNT 為 32 位元計數器 HC0 至 HC63 之啟動指令。
- 一般用加減算計數器 HC0~HC63，當 DCNT 指令由 OFF→ON 時，計數器之現在值將執行上數（加一）的動作或下數（減一）的動作，依 SM621~SM684 的設定模式。
- 當 DCNT 指令 OFF 時，該計數器停止計數，但原有計數值不會被清除，可使用指令 RST 清除計數值及其接點。

程式範例：

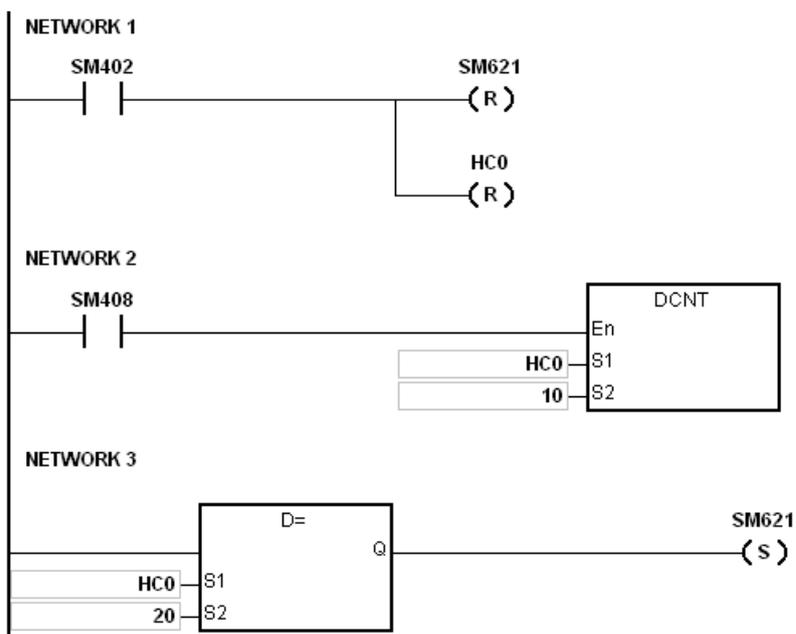
當 PLC RUN 時，計時器 HC0 被清除，且 SM621=OFF 將執行上數，此時 SM408 的狀態為第一次 ON，所以計數器 HC0 會載入設定值 10 並開始計數。

當 SM408 的狀態經過 10 次 OFF=>ON 的變化後，計數器 HC0 的值計數到達 HC0=10，且 HC0 的接點為 ON。

當 HC0 為 ON 之後，因為 SM408 繼續 OFF=>ON，所以雖然 HC0 的值已經計數到達，但還是會繼續累加。

當計數器繼續上數到達 HC0=20 時，程式會設定 SM621=ON 將執行下數，當 SM408 的狀態再經過 10 次 OFF=>ON 的變化後，計數器 HC0 的值計數到達由 HC0=10 遞減到 HC0=9 時，HC0 的接點會被 OFF。

當 HC0 的接點被 OFF 後，因為 SM408 繼續 OFF=>ON，所以 HC0 的值還是會繼續遞減。



補充說明：

1. SM621~SM684 的設定模式，詳細使用說明，請參考第 2 章的 32 位元計數器 HC 的使用說明。
2. S₁ 運算元，若使用 ISPSOft 宣告，則資料型態為 COUNTER。

6

6.12 移位指令

6.12.1 移位指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1100</u>	SFTR	–	✓	位元右移	9
<u>1101</u>	SFTL	–	✓	位元左移	9
<u>1102</u>	WSFR	–	✓	暫存器右移	9
<u>1103</u>	WSFL	–	✓	暫存器左移	9
<u>1104</u>	SFWR	–	✓	位移寫入	7
<u>1105</u>	SFRD	–	✓	位移讀出	7
<u>1106</u>	SFPO	–	✓	讀出資料串列最新資料	5
<u>1107</u>	SFDEL	–	✓	刪除資料串列中的資料	7
<u>1108</u>	SFINS	–	✓	插入資料到資料串列中	7
<u>1109</u>	MBS	–	✓	矩陣位元位移	7
<u>1110</u>	SFR	–	✓	16 位元暫存器位元右移	5
<u>1111</u>	SFL	–	✓	16 位元暫存器位元左移	5
<u>1112</u>	BSFR	–	✓	n 個位元右移 1 個位元	5
<u>1113</u>	BSFL	–	✓	n 個位元左移 1 個位元	5
<u>1114</u>	NSFR	–	✓	n 個暫存器右移	5
<u>1115</u>	NSFL	–	✓	n 個暫存器左移	5

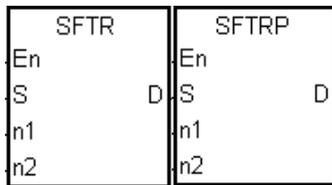
6.12.2 移位指令說明

API	指令碼		運算元								功能					
1100		SFTR	P	S · D · n₁ · n₂								位元右移				

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●	●	●				●	●	●			●				
D	●	●	●	●				●	●	●			●				
n₁	●	●						●	●		●		●	○	○		
n₂	●	●						●	●		●		●	○	○		

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



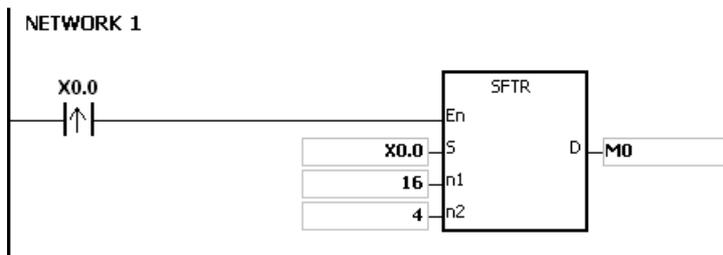
- S** : 移位裝置之起始編號 Bit
- D** : 欲移位裝置之起始編號 Bit
- n₁** : 欲移位之資料長度 Word
- n₂** : 一次移位之位元數 Word

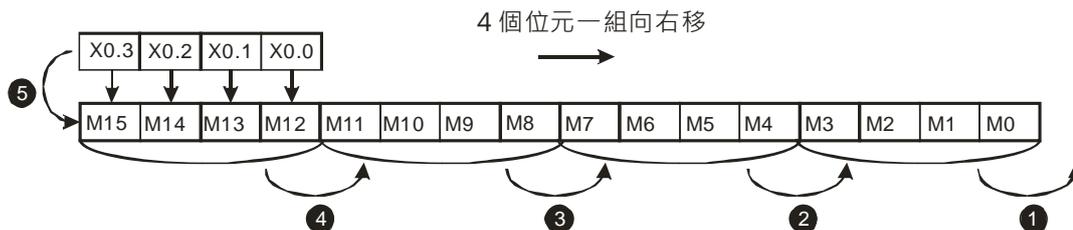
指令說明：

1. 將 **D** 開始之起始編號，具有 **n₁** 個數位元(位移暫存器長度)的位元裝置，以 **n₂** 位元個數來右移。而 **S** 開始起始編號以 **n₂** 位元個數移入 **D** 中來填補位元空位。
2. 本指令一般都是使用脈波執行型指令 (SFTRP)。
3. **n₁** 值的範圍為 1~1024，**n₂** 值的範圍為 1~**n₁**。

程式範例：

1. 在 X0.0 上升緣時，由 M0~M15 組成 16 位元，以 4 位元作右移。
2. 掃描一次的位元右移動作依照下列編號 1~5 動作。
 - ① M3~M0 → 進位
 - ② M7~M4 → M3~M0
 - ③ M11~M8 → M7~M4
 - ④ M15~M12 → M11~M8
 - ⑤ X0.3~X0.0 → M15~M12 完成

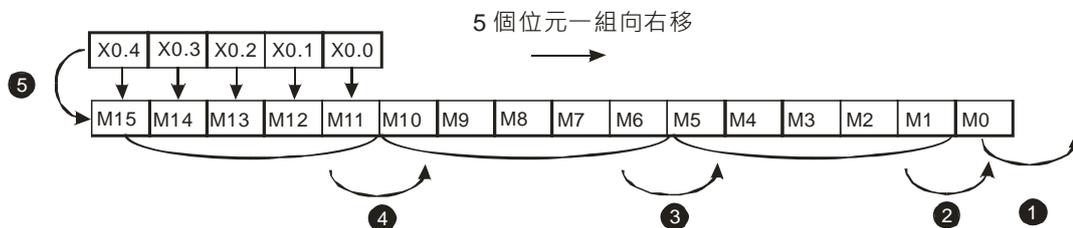
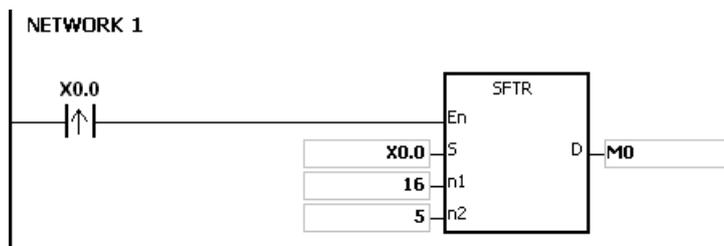




3. 在 X0.0 上升緣時，由 M0~M15 組成 16 位元，以 5 位元作右移。

4. 掃描一次的位元右移動作依照下列編號 1~5 動作。

- ❶ M0 → 進位
- ❷ M5 → M0
- ❸ M10~M6 → M5~M1
- ❹ M15~M11 → M10~M6
- ❺ X0.4~X0.0 → M15~M11 完成



6

補充說明：

1. $S+n_2-1$ 、 $D+n_1-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 1024$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1101		SFTL	P	S · D · n ₁ · n ₂								位元左移					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

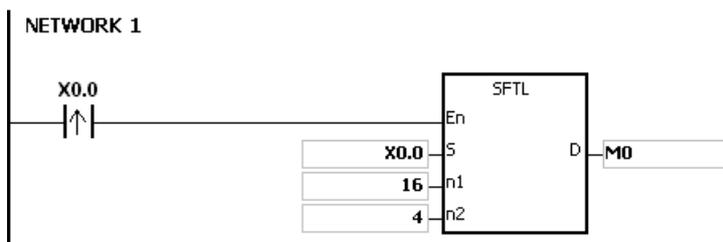
SFTL	SFTLP	S	D	n ₁	n ₂	說明	單位
En	En					移位裝置之起始編號	Bit
S	S					欲移位裝置之起始編號	Bit
n ₁	n ₁					欲移位之資料長度	Word
n ₂	n ₂					一次移位之位元數	Word

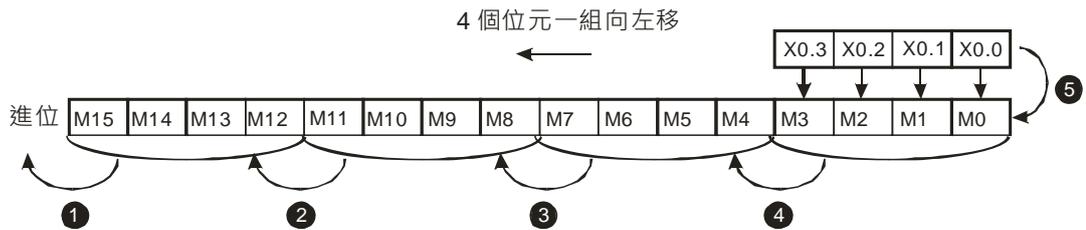
指令說明：

- 將 **D** 開始之起始編號，具有 **n₁** 個數位元(位移暫存器長度)的位元裝置，以 **n₂** 位元個數來左移。而 **S** 開始起始編號以 **n₂** 位元個數移入 **D** 中來填補位元空位。
- 本指令一般都是使用脈波執行型指令 (SFTLP)。
- n₁** 值的範圍為 1~1024，**n₂** 值的範圍為 1~**n₁**。

程式範例：

- 在 X0.0 上升緣時，由 M0~M15 組成 16 位元，以 4 位元作左移。
- 掃描一次的位元左移動作依照下列編號 1~5 動作。
 - ① M15~M12 → 進位
 - ② M11~M8 → M15~M12
 - ③ M7~M4 → M11~M8
 - ④ M3~M0 → M7~M4
 - ⑤ X0.3~X0.0 → M3~M0 完成

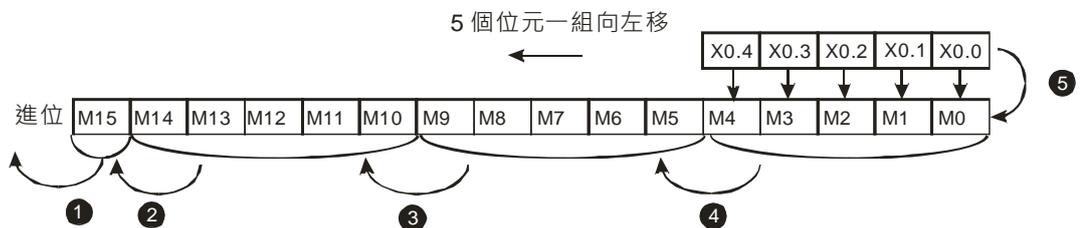
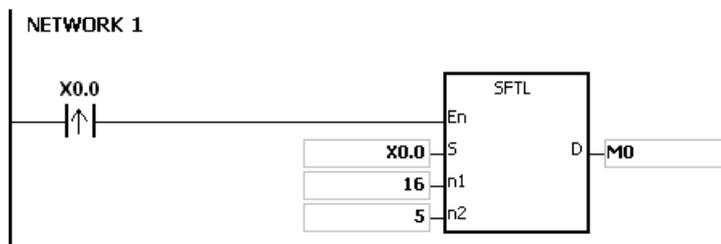




3. 在 X0.0 上升緣時，由 M0~M15 組成 16 位元，以 5 位元作左移。

4. 掃描一次的位元左移動作依照下列編號 1~5 動作。

- ❶ M15 → 進位
- ❷ M10 → M15
- ❸ M9~M5 → M14~M10
- ❹ M4~M0 → M9~M5
- ❺ X0.4~X0.0 → M4~M0 完成



補充說明：

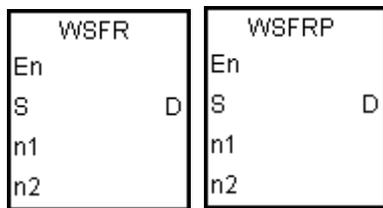
1. $S+n_2-1$ 、 $D+n_1-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 1024$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元							功能						
1102		WSFR	P	S · D · n ₁ · n ₂							暫存器右移						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



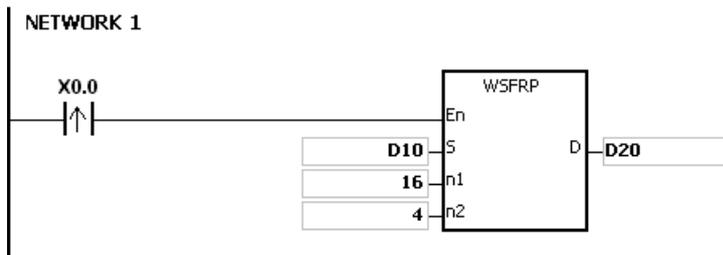
- S**：移位裝置之起始編號 Word
- D**：欲移位裝置之起始編號 Word
- n₁**：欲移位之資料長度 Word
- n₂**：一次移位之位元數 Word

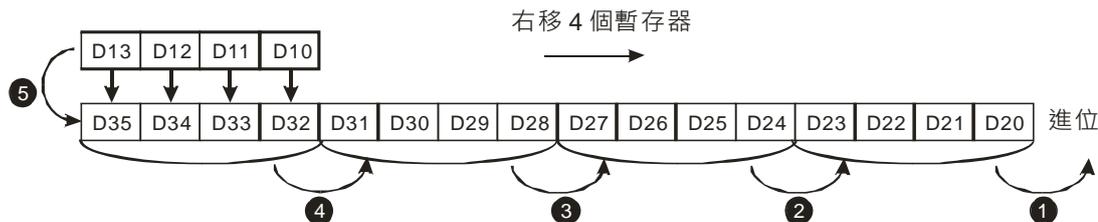
指令說明：

- 將 **D** 開始之起始編號，具有 **n₁** 個字元長度的資料串列，以 **n₂** 個字元來右移。而 **S** 開始起始編號以 **n₂** 字元個數移入 **D** 中來填補字元空位。
- 本指令一般都是使用脈波執行型指令 (WSFRP)。
- n₁** 值的範圍為 1~512 · **n₂** 值的範圍為 1~**n₁**。

程式範例一：

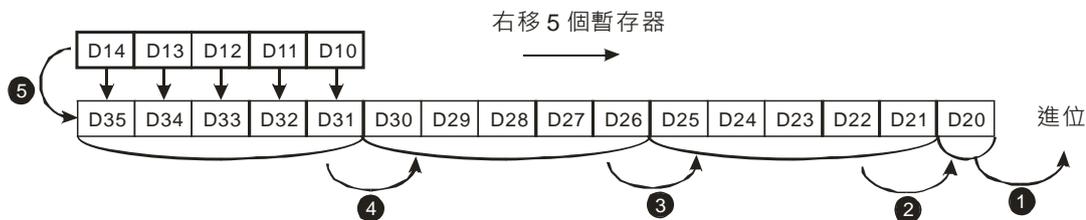
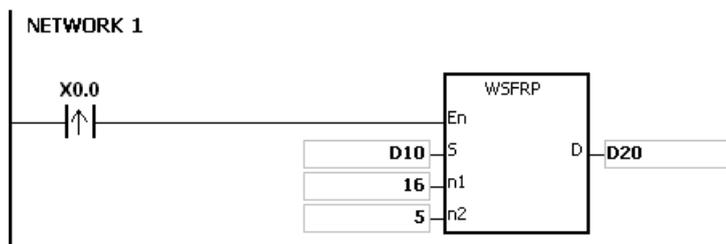
- X0.0=OFF→ON 時，由 D20~D35 所組成之 16 個暫存器資料串列為位移區域，以 4 個暫存器來右移。
- 掃描一次的字元右移動作依照下列編號 1~5 動作。
 - ❶ D23~D20 → 進位
 - ❷ D27~D24 → D23~D20
 - ❸ D31~D28 → D27~D24
 - ❹ D35~D32 → D31~D28
 - ❺ D13~D10 → D35~D32 完成





程式範例二：

1. X0.0=OFF→ON 時，由 D20~D35 所組成之 16 個暫存器資料串列為位移區域，以 5 個暫存器來右移。
2. 掃描一次的字元右移動作依照下列編號 1~5 動作。
 - ① D20 → 進位
 - ② D25 → D20
 - ③ D30~D26 → D25~D21
 - ④ D35~D31 → D30~D26
 - ⑤ D14~D10 → D35~D31 完成



補充說明：

1. $S+n_2-1$ 、 $D+n_1-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 512$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1103	WSFL	P		S · D · n ₁ · n ₂								暫存器左移					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：



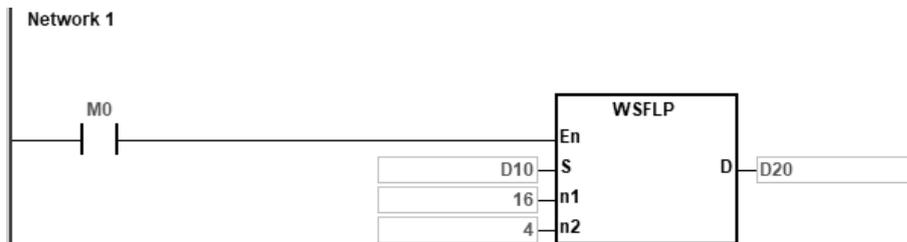
- S**：移位裝置之起始編號 Word
- D**：欲移位裝置之起始編號 Word
- n₁**：欲移位之資料長度 Word
- n₂**：一次移位之位元數 Word

指令說明：

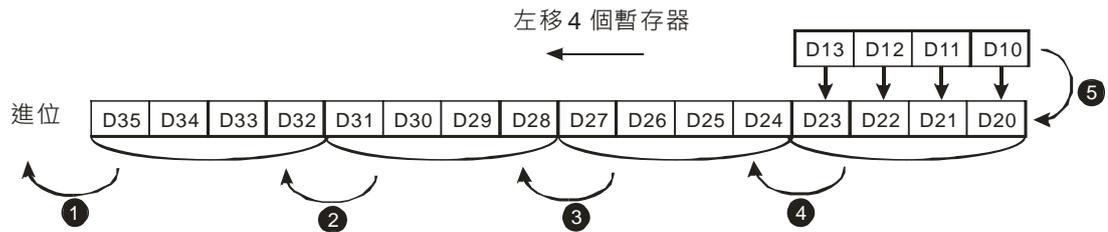
- 將 **D** 開始之起始編號，具有 **n₁** 個字元長度的資料串列，以 **n₂** 個字元來左移。而 **S** 開始起始編號以 **n₂** 字元個數移入 **D** 中來填補字元空位。
- 本指令一般都是使用脈波執行型指令 (WSFLP)
- n₁** 值的範圍為 1~512 · **n₂** 值的範圍為 1~**n₁**。

程式範例：

- 在 M0 為 ON 時，由 D20~D35 組成 16 位元，以 4 位元作左移。
- 掃描一次的位元左移動作依照下列編號 1~5 動作。
 - ❶ D35~D32 → 進位
 - ❷ D31~D28 → D35~D32
 - ❸ D27~D24 → D31~D28
 - ❹ D23~D20 → D27~D24
 - ❺ D13~D10 → D23~D20 完成



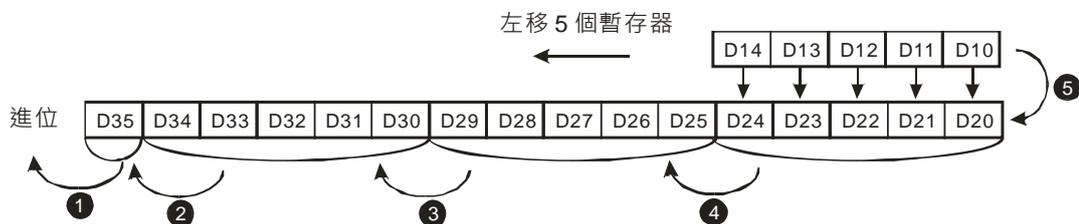
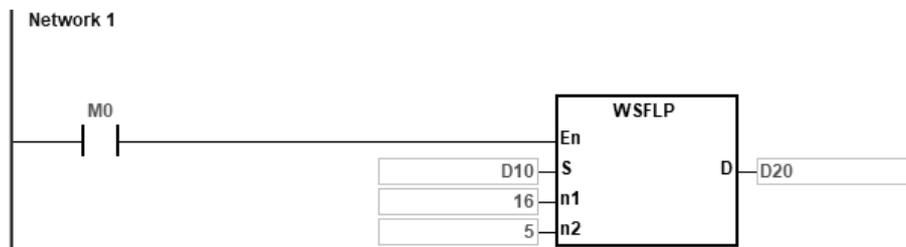
6



3. 在 M0 為 ON 時，由 D20~D35 組成 16 位元，以 5 位元作左移。

4. 掃描一次的位元左移動作依照下列編號 1~5 動作。

- ❶ D35 → 進位
- ❷ D30 → D35
- ❸ D29~D25 → D34~D30
- ❹ D24~D20 → D29~D25
- ❺ D14~D10 → D24~D20 完成



補充說明：

1. $S+n_2-1$ 、 $D+n_1-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n_1 < 1$ 或 $n_1 > 512$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. $n_2 < 1$ 或 $n_2 > n_1$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元							功能						
1104		SFWR	P	S · D · n							位移寫入						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

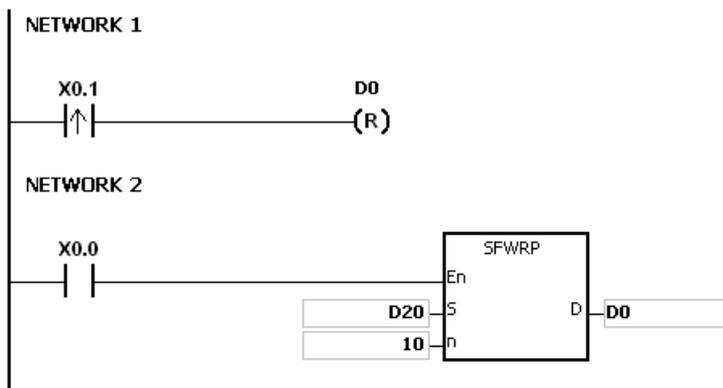
SFWR	SFWRP	S : 位移寫入資料串列之裝置 Word D : 資料串列之起始編號 Word n : 資料串列之長度 Word
En	En	
S	D	

指令說明：

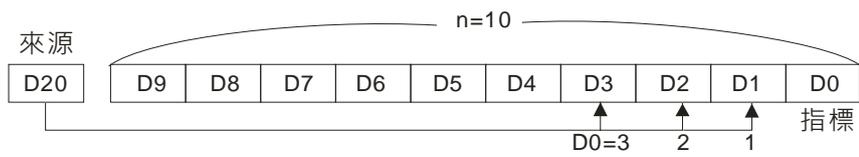
- 將 **D** 起始編號開始 **n** 個字元裝置的資料串列定義為先入先出資料串列，以第一個編號裝置作為指標，當指令執行時，指標內容值先加 1，之後 **S** 所指定的裝置其內容值會寫入先入先出資料串列中由指標所指定的位置。當指標的內容大於等於 **n-1** 時，本指令不再處理寫入的新值，進位旗標信號 SM602=ON。
- 本指令一般都是使用脈波執行型指令 (SFWRP)。
- n** 值的範圍為 2~512。

程式範例：

- 開始先將指標 D0 清除為 0，當 X0.0 = OFF→ON 變化時，D20 的內容被傳送至 D1 當中，指標 D0 內容變成 1。變更 D20 的內容後，將 X0.0 再 OFF→ON 一次，則 D20 的內容被傳送至 D2 當中，D0 內容變成 2。
- 指令執行一次位移寫入動作依照下列動作。
 - D20 的內容被傳送至 D1 當中。
 - 指標 D0 內容變成 1。



6



補充說明：

1. **D** 的內容值 <0 ，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. $D+n-1$ 裝置超出範圍時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
3. $n<2$ 或 $n>512$ ，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
4. 本指令 API1104 SFWR 與 API1105 SFRD 可搭配使用，執行先入先出資料串列的寫入讀出控制。

API	指令碼			運算元								功能				
1105		SFRD	P	S · D · n								位移讀出				

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

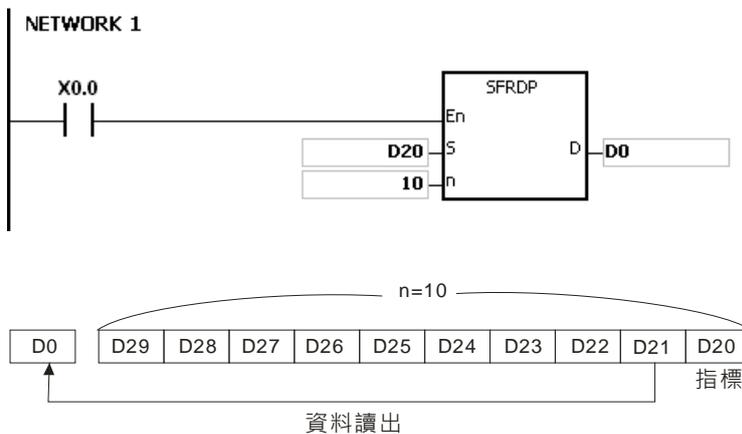
SFRD	SFRDP	S	D	n	說明
En	En	S	D	n	S : 資料串列之起始編號 Word
			D		D : 資料串列位移讀出之裝置 Word
				n	n : 資料串列之長度 Word

指令說明：

- 將 **S** 起始編號開始 **n** 個字元裝置的資料串列定義為先入先出資料串列，以第一個編號裝置作為指標，當指令執行時，**S** 內容值先減 1，之後 (**S+1**) 所指定的裝置其內容值會寫入 **D** 所指定的位置，(**S+n-1**) ~ (**S+2**) 全部右移一個暫存器，(**S+n-1**) 的內容不變，當 **S** 的內容等於 0 時，本指令不再處理資料讀出的動作，零旗標信號 SM600=ON。
- 本指令一般都是使用脈波執行型指令 (SFRDP)。
- n** 值的範圍為 2~512。

程式範例：

- 當 X0.0 從 OFF→ON 變化時，D21 的內容被傳送至 D0 當中，D29~D22 全部往右位移一個暫存器 (D29 內容保持不變)，D20 內容減 1。
- 指令執行一次位移讀出動作依照下列動作。
 - D21 的內容被讀出傳送至 D0 當中。
 - D29~D22 全部往右位移一個暫存器。
 - D20 內容減 1。



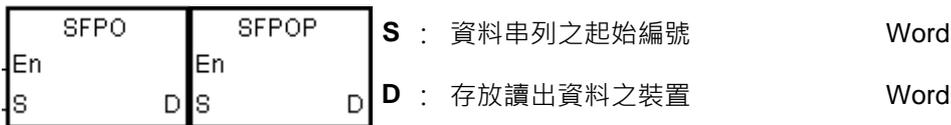
補充說明：

1. $S < 0$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $S+n-1$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. $n < 2$ 或 $n > 512$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
4. 本指令 API1104 SFWR 與 API1105 SFRD 可搭配使用，執行先入先出資料串列的寫入讀出控制。

API	指令碼			運算元							功能						
1106		SFPO	P	S · D							讀出資料串列最新資料						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●	○	●				

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：

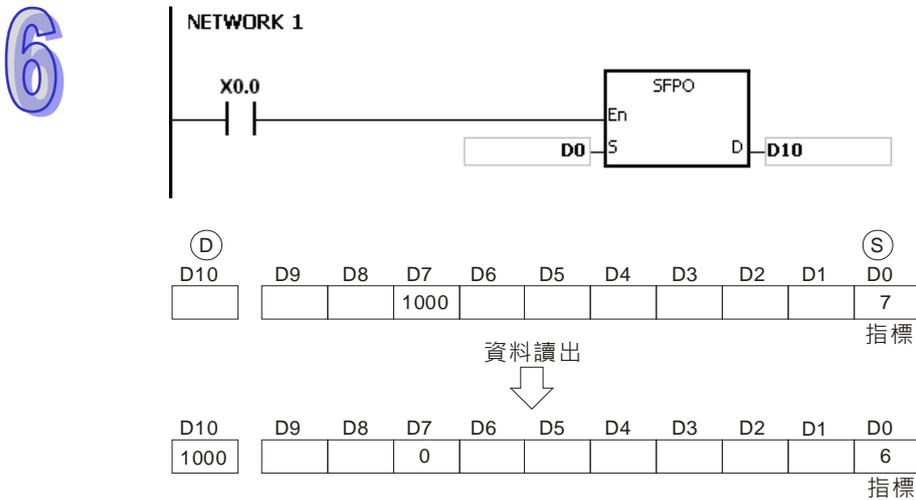


指令說明：

- 將 **S** 起始編號開始的資料串列，以第一個編號裝置作為指標，當指令執行時，**S** 內容值所指定的指標內容值會寫入 **D** 所指定的位置，**S** 內容值所指定的指標內容值被清除為 0，**S** 內容值減一，當 **S** 的內容等於 0 時，本指令不再處理資料讀出的動作，零旗標信號 SM600=ON。
- 本指令一般都是使用脈波執行型指令 (SFPOP)。

程式範例：

當 X0.0=ON 時，D0 內容所指定的位置內容搬移至 D10 當中，搬移後指定位置清除為 0，D0 內容減一。



補充說明：

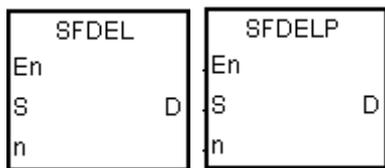
- S** 內容值 < 0，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
- S** 編號 + **S** 內容值，裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能					
1107	SFDEL	P		S · D · n								刪除資料串列中的資料					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：



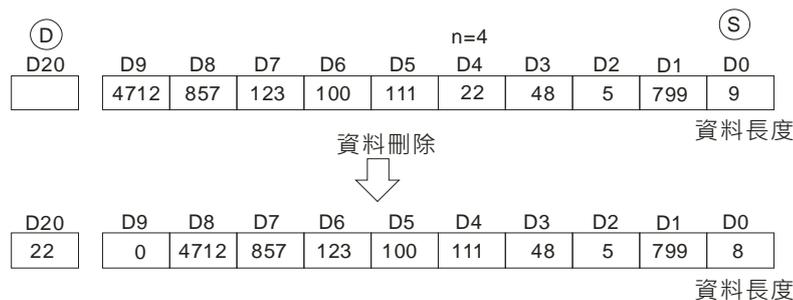
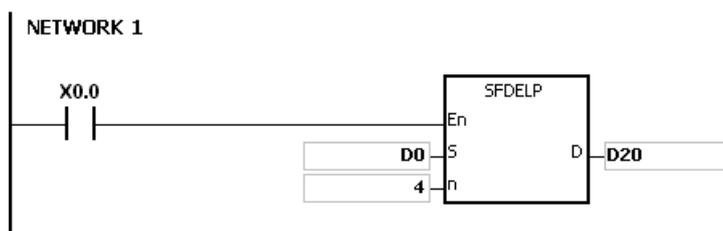
S：資料串列之起始編號 Word
D：儲存被刪除的資料的裝置 Word
n：要刪除的位置 Word

指令說明：

- S** 為資料串列的長度，(**S+1**) ~ (**S+S** 內容值) 為資料串列的內容，當指令執行時，**S+n** 的內容值會被儲存在 **D** 中，並刪除 **S+n** 的內容值，然後 (**S+n+1**) ~ (**S+S** 內容值) 的裝置內容全部右移一個暫存器，(**S+S** 內容值) 的內容歸零，最後將 **S** 內容值減一，當 **S** 的內容等於 0 時，本指令不再處理資料刪除的動作，零旗標信號 SM600=ON。
- 本指令一般都是使用脈波執行型指令 (SFDELP)。
- n** 運算元值的範圍為 1~32767。

程式範例：

給定 D0=9，當 X0.0=ON 時，將 n=4 指定的 D4 內容搬移至 D20 當中，並刪除 D4 的內容值，然後將 D5~D9 的內容向前遞補，最後將 D0 內容減一。



補充說明：

1. **S** 內容值 <0 ，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **S+n** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. **S+S** 內容值，裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. **n>S** 內容值，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
5. **n \leq 0** 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

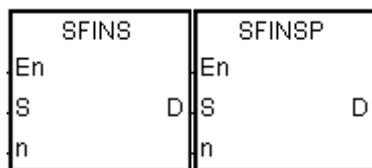
6

API	指令碼		運算元								功能					
1108	SFINS	P	S · D · n								插入資料到資料串列中					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：



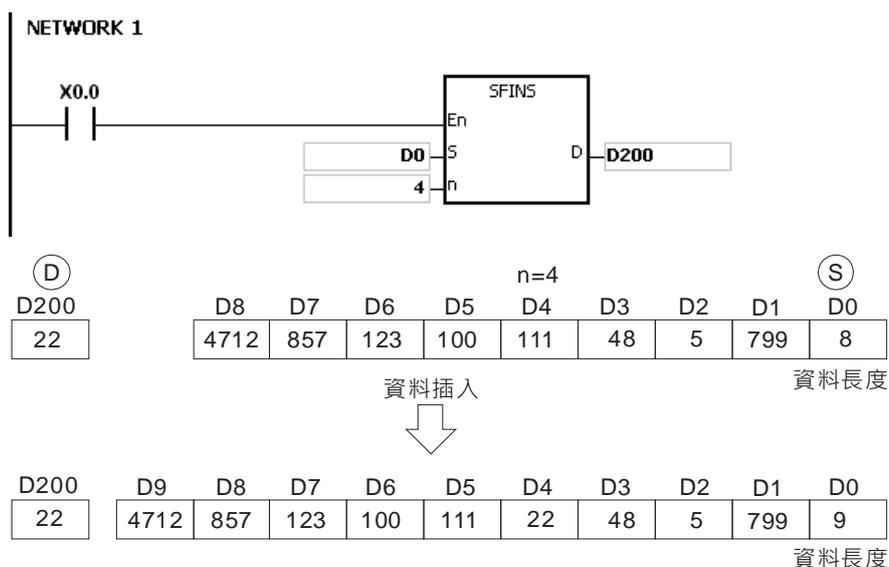
- S**：資料串列之起始編號 Word
- D**：要插入資料串列的資料 Word
- n**：要插入的位置 Word

指令說明：

- S** 為資料串列的長度，(**S+1**) ~ (**S+S** 內容值) 為資料串列的內容，當指令執行時，**D** 的內容值會被插入 **S+n** 中，原本 (**S+n**) ~ (**S+S** 內容值) 的資料全部往下移動一個暫存器，最後將內容值加一，**S** 內容值=32767 時，本指令不再處理寫入的新值，**S** 內容值不再往上增加，進位旗標信號 SM602=ON。
- 本指令一般都是使用脈波執行型指令 (SFINSP)。
- n** 運算元值的範圍為 1~32767。

程式範例：

給定 D0=8，當 X0.0=ON 時，先將 D200 的內容值插入 n=4 指定的 D4 中，再將原先 D4~D8 的內容向下位移填入 D5~D9 中，最後將 D0 內容加一。



補充說明：

1. **S**≤0，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **S+n** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. **S+S** 內容值+1，裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. **n>S** 內容值，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
5. **n**≤0 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

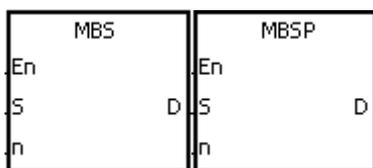
6

API	指令碼		運算元							功能						
1109	MBS	P	S · D · n							矩陣位元位移						

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：



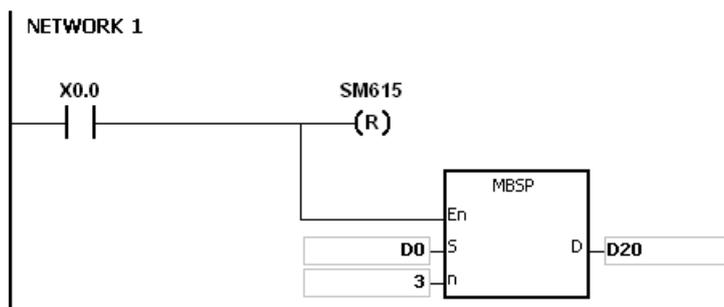
S：矩陣來源裝置 Word
D：運算結果 Word
n：陣列長度 Word

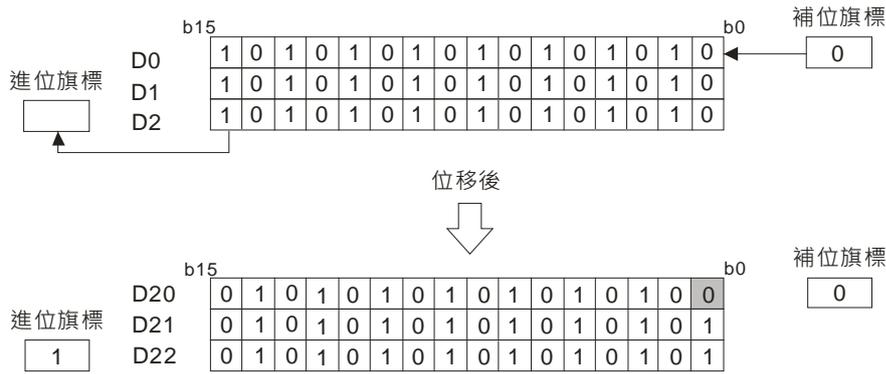
指令說明：

1. 矩陣來源依陣列長度 **n** 將 **S** 矩陣位元做左右位移控制，**SM616=0** 決定矩陣位元左移，**SM616=1** 決定矩陣位元右移。每次移動一位元，因位移而騰出之空位（左移時為 **b0**，右移時為 **b16n-1**）則以補位旗標 **SM615** 之狀態填補。而因位移而擠出之位元（左移時為 **b16n-1**，右移時為 **b0**）狀態則送到進位旗標 **SM614** 去，然後將結果存入 **D**。
2. **n** 運算元值的範圍為 1~256。
3. 本指令一般都是使用脈波執行型指令（**MBSP**）。

程式範例一：

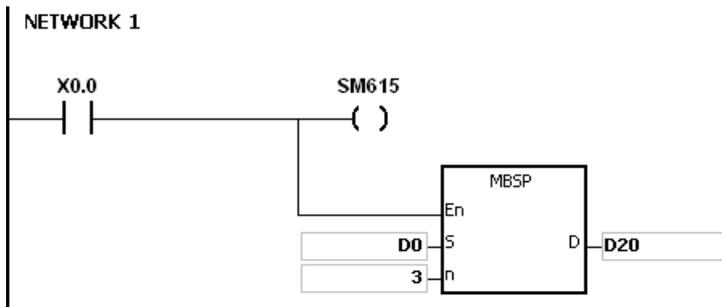
當 **X0.0=ON** 時，矩陣旋轉位移方向旗標 **SM616=OFF** 作矩陣左移。設矩陣位移輸入補位旗標 **SM615=0**，16 位元暫存器 **D0~D2** 矩陣作左移，將結果存於 16 位元暫存器 **D20~D22** 矩陣中，矩陣旋轉位移輸出進位旗標 **SM614=1**。



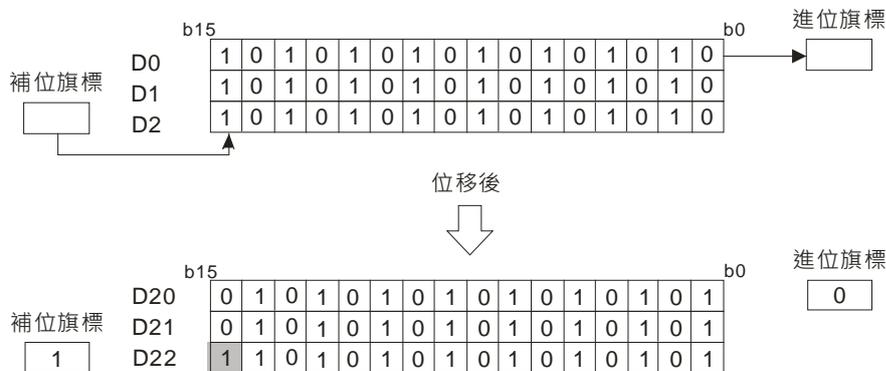


程式範例二：

當 X0.0=ON 時，矩陣旋轉位移方向旗標 SM616=ON 作矩陣右移，設矩陣位移輸入補位旗標 SM615=1，16 位元暫存器 D0~D2 矩陣作右移，將結果存於 16 位元暫存器 D20~D22 矩陣中，矩陣旋轉位移輸出進位旗標 SM614=0。



6



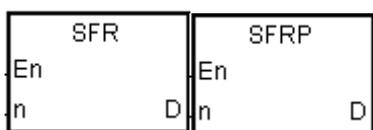
補充說明：

1. **S+n-1**、**D+n-1** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n<1** 或 **n>256** 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. 旗標信號說明
 SM614： 矩陣旋轉位移輸出進位旗標
 SM615： 矩陣位移輸入補位旗標
 SM616： 矩陣旋轉位移方向旗標

API	指令碼			運算元								功能					
1110	SFR	P		D · n								16 位元暫存器位元右移					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



D：欲位移之裝置 Word
n：一次位移之位元數 Word

指令說明：

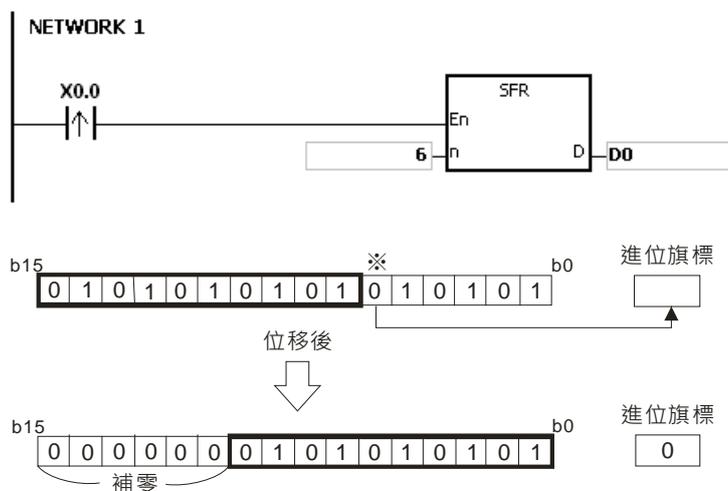
- 將 D 往右移 n 個位元數，而因位移而騰出之空位 (b₁₅~b_{15-n+1}) 則補零。因位移而擠出之第 n-1 位元 (b_{n-1}) 狀態則送到進位旗標 SM602 去。
- n 內容值的範圍為 1~16。
- 本指令一般都是使用脈波執行型指令 (SFRP)。

程式範例：

當 X0.0=ON 時，D0 的 b0~b15，往右移動 6 個位元數，而 b5 的狀態傳送到 SM602，位移後，b10~b15 內容清除為 0。

掃描一次的位元右移動作依照下列編號 1~3 動作。

- ❶ b5~b0 → 進位 (SM602=b5)
- ❷ b15~b6 → b9~b0
- ❸ 0 → b15~b10 完成



補充說明：

n<0 或 n>16，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1111		SFL	P	D · n								16 位元暫存器位元左移					

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



D : 欲位移之裝置 Word

n : 一次位移之位元數 Word

指令說明：

- 將 D 往左移 n 個位元數，而因位移而騰出之空位 (b₀~b_{n-1}) 則補零。因位移而擠出之第 16-n 位元 (b_{16-n}) 狀態則送到進位旗標 SM602 去。
- n 內容值的範圍為 1~16。
- 本指令一般都是使用脈波執行型指令 (SFLP)。

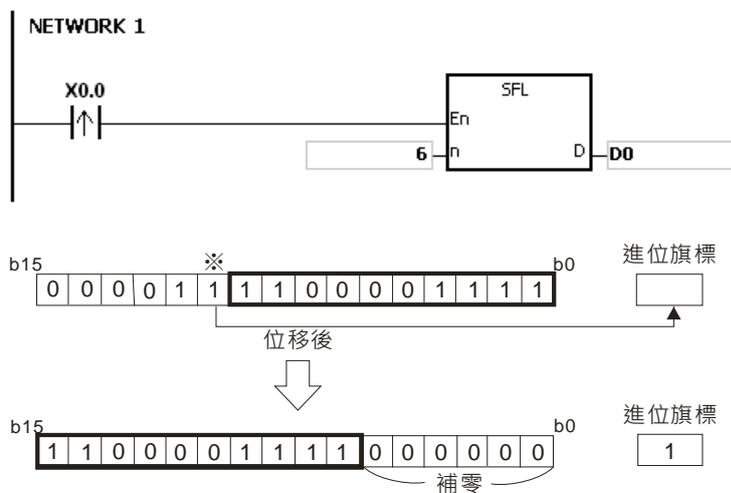
程式範例：

當 X0.0=ON 時，D0 的 b₀~b₁₅，往左移動 6 個位元數，而 b₁₀ 的狀態傳送到 SM602，位移後，b₀~b₅ 內容清除為 0。

6

掃描一次的位元右移動作依照下列編號 1~3 動作。

- 1 b₁₅~b₁₀ → 進位 (SM602=b₁₀)
- 2 b₉~b₀ → b₁₅~b₆
- 3 0 → b₅~b₀ 完成



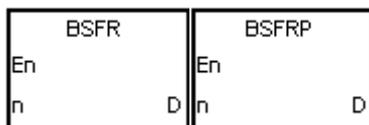
補充說明：

n<1 或 n>16，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元							功能						
1112		BSFR	P	D · n							n 個位元右移 1 個位元 (位元裝置)						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●				●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



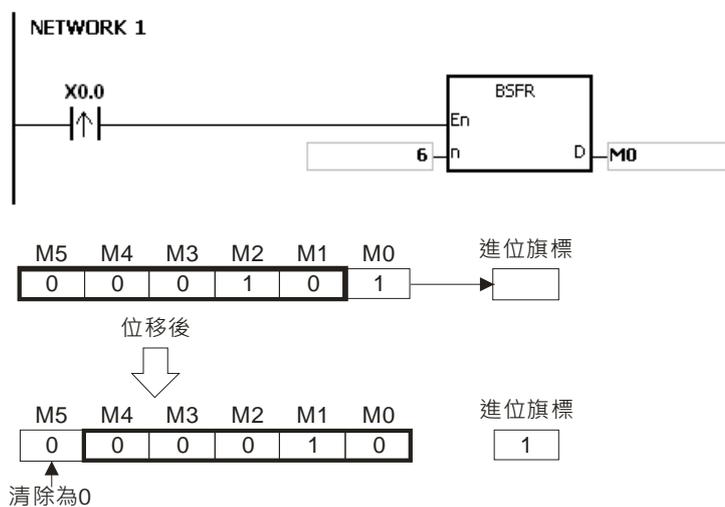
D：移位裝置之起始編號 Bit
n：資料長度 Word

指令說明：

- 將 **D** 開始的 **n** 個位元，右移 1 個位元，**D+n-1** 位元被清除為 0，**D** 位元狀態則送到進位旗標 SM602。
- 本指令一般都是使用脈波執行型指令 (BSFRP)。
- n** 運算元的範圍為 1~1024。

程式範例：

當 X0.0=ON 時，M0~M5 依序向右位移 1 個位元，M5 的狀態清除為 0，M0 的狀態傳送至進位旗標 SM602。



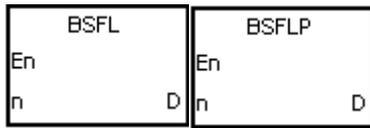
補充說明：

- D+n-1** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
- n<1** 或 **n>1024**，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1113		BSFL	P	D · n								n 個位元左移 1 個位元					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●	●	●				●	●	●			●				
n	●	●			●	●		●	●		●	○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



D：移位裝置之起始編號 Bit
n：資料長度 Word

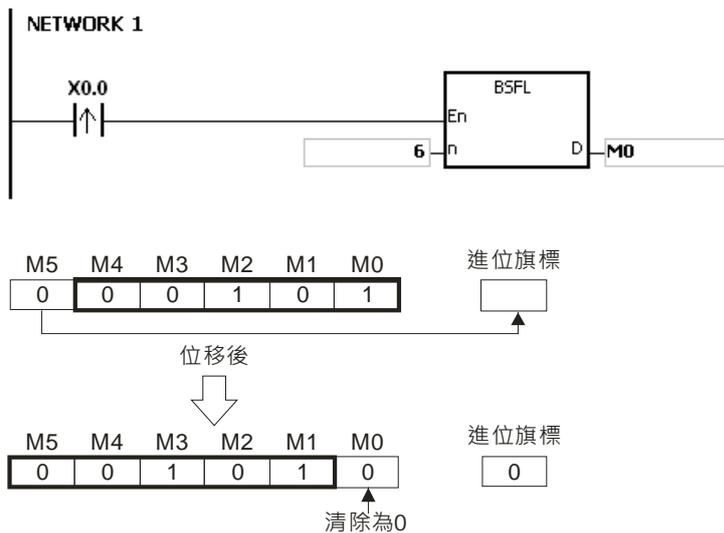
指令說明：

1. 將 **D** 開始的 **n** 個位元，左移 1 個位元，**D** 位元被清除為 0，**D+n-1** 位元狀態則送到進位旗標 **SM602**。
2. 本指令一般都是使用脈波執行型指令 (**BSFLP**)。
3. **n** 運算元的範圍為 1~1024。

程式範例：

當 X0.0=ON 時，M0~M5 內容依序向左位移 1 個位元，M0 的狀態清除為 0，M5 的狀態傳送至進位旗標 SM602。

6



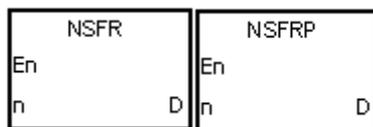
補充說明：

1. **D+n-1** 裝置超出範圍時，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2003**。
2. **n<1** 或 **n>1024**，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#200B**。

API	指令碼			運算元							功能						
1114		NSFR	P	D · n							n 個暫存器右移						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



D：移位裝置之起始編號 Word

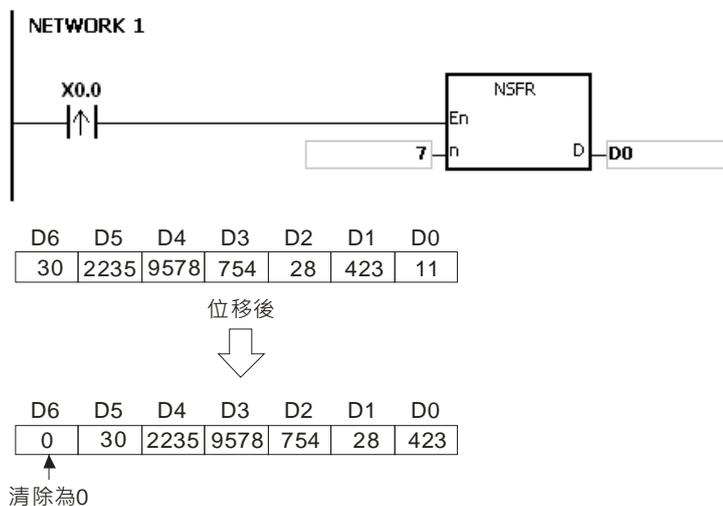
n：資料長度 Word

指令說明：

1. 將 D 開始的 n 個暫存器右移，D+n-1 裝置被清除為 0。
2. 本指令一般都是使用脈波執行型指令 (NSFRP)。
3. n 運算元的範圍為 1~512。

程式範例：

當 X0.0=ON 時，D1~D6 內容依序向右位移 1 個暫存器位置，最後 D6 的內容清除為 0。



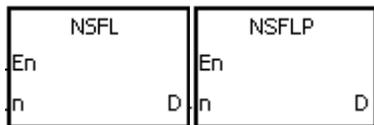
補充說明：

1. D+n-1 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. n<1 或 n>512，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元							功能						
1115		NSFL	P	D · n							n 個暫存器左移						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●	○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



D：移位裝置之起始編號 Word
n：資料長度 Word

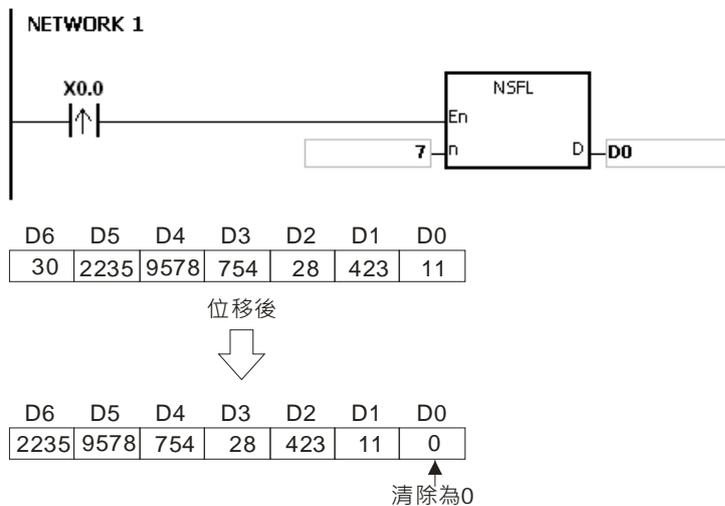
指令說明：

1. 將 **D** 開始的 **n** 個暫存器左移，**D** 裝置被清除為 0。
2. 本指令一般都是使用脈波執行型指令 (NSFLP)。
3. **n** 運算元的範圍為 1~512。

程式範例：

當 X0.0=ON 時，D0~D5 內容依序向左位移 1 個暫存器位置，最後 D0 的內容清除為 0。

6



補充說明：

1. **D+n-1** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n<1** 或 **n>512**，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

6.13 資料處理指令

6.13.1 資料處理指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1200</u>	SER	DSER	✓	多點比較	9
<u>1201</u>	SUM	DSUM	✓	ON 位元數量	5
<u>1202</u>	DECO	–	✓	解碼器	7
<u>1203</u>	ENCO	–	✓	編碼器	7
<u>1204</u>	SEGD	–	✓	7 段顯示器解碼	5
<u>1205</u>	SORT	DSORT	–	資料排序	11
<u>1206</u>	ZRST	–	✓	區域清除	5
<u>1207</u>	BON	DBON	✓	ON 位元判定	7
<u>1208</u>	MEAN	DMEAN	✓	平均值	7
<u>1209</u>	CCD	–	✓	總和檢查	7
<u>1210</u>	ABS	DABS	✓	絕對值	3
<u>1211</u>	MINV	–	✓	矩陣反相	7
<u>1212</u>	MBRD	–	✓	矩陣位元讀出	7
<u>1213</u>	MBWR	–	✓	矩陣位元寫入	7
<u>1214</u>	MBC	–	✓	矩陣位元狀態計數	7
<u>1215</u>	DIS	–	✓	16位元資料的4位元分組	7
<u>1216</u>	UNI	–	✓	16位元資料的4位元鏈結	7
<u>1217</u>	WSUM	DWSUM	✓	求和	7
<u>1218</u>	BSET	–	✓	字元裝置位元動作保持 ON	5
<u>1219</u>	BRST	–	✓	字元裝置位元清除	5
<u>1220</u>	BKRST	–	✓	指定區域清除	5
<u>1221</u>	LIMIT	DLIMIT	✓	高低限控制	9
<u>1222</u>	BAND	DBAND	✓	死區控制	9
<u>1223</u>	ZONE	DZONE	✓	區域控制	9

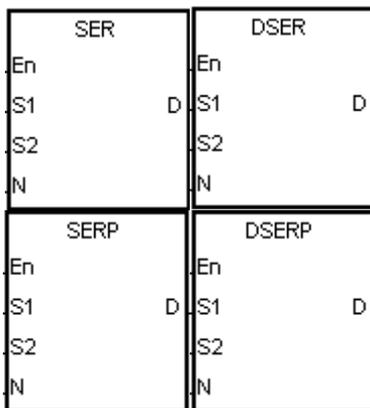
6.13.2 資料處理指令說明

API	指令碼			運算元							功能			
1200	D	SER	P	$S_1 \cdot S_2 \cdot D \cdot n$							多點比較			

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
AH500	AH500	AH500

符號：



- S_1 : 多點比較之資料區塊的起始裝置 Word/Double Word
- S_2 : 欲比較之數值資料 Word/Double Word
- D : 存放比較結果之起始裝置 Word/Double Word
- n : 被比較之資料區塊長度 Word/Double Word

指令說明：

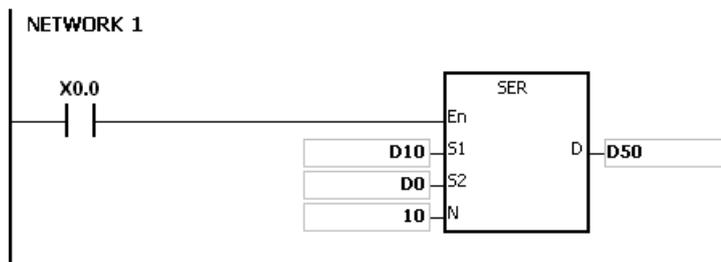
- S_1 指定被比較暫存器區域的號碼，n 指定被比較的筆數，該多筆被比較暫存器的內容與 S_2 所指定的資料，以有號數十進制作比較，比較結果被存放在 $D-D+4$ 暫存器當中，並說明如下：

裝置	說明
D	相等值的資料數
D+1	第一個相等值的編號
D+2	最後一個相等值的編號
D+3	最小值的編號
D+4	最大值的編號

- 16 位元指令中，n 運算元的內容值為 1~256；32 位元指令中，n 運算元的內容值為 1~128。
- 32 位元指令才可以使用 HC 裝置。

程式範例：

- 當 X0.0=ON 時，由 D10~D19 組成之資料區塊與 D0 作比較，結果存放在 D50~D54 中，當相等值不存在時，D50~D52 的內容全部為 0。
- 所有比較資料之最小值編號記錄在 D53，最大值編號記錄在 D54。當最小值、最大值不只一個時，會記錄編號大者。



S ₁	內容值	比較資料	資料編號	結果	D	內容值	說明
D10	88	S ₂ D0=100	0		D50	4	相等值的資料數
D11	100		1	相等	D51	1	第一個相等值的編號
D12	110		2		D52	8	最後一個相等值的編號
D13	150		3		D53	7	最小值的編號
D14	100		4	相等	D54	9	最大值的編號
D15	300		5				
D16	100		6	相等			
D17	5		7	最小			
D18	100		8	相等			
D19	500		9	最大			

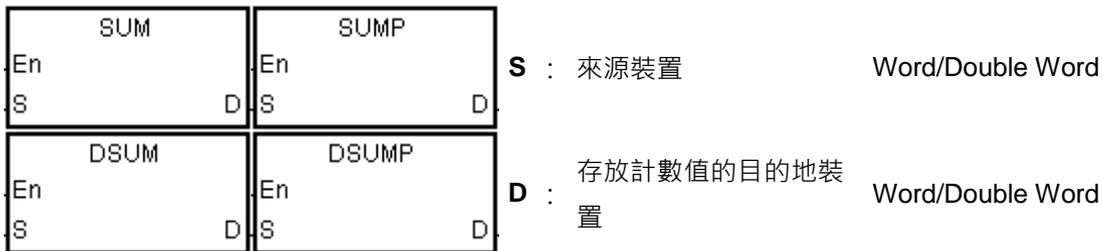
補充說明：

1. S₁+n-1 或 D+4，超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 16 位元指令 n<1 或 n>256 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. 32 位元指令 n<1 或 n>128 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
4. 16 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [5] of WORD/INT。
5. 32 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [5] of DWORD/DINT。

API	指令碼			運算元							功能						
1201	D	SUM	P	S · D							ON 位元數量						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：

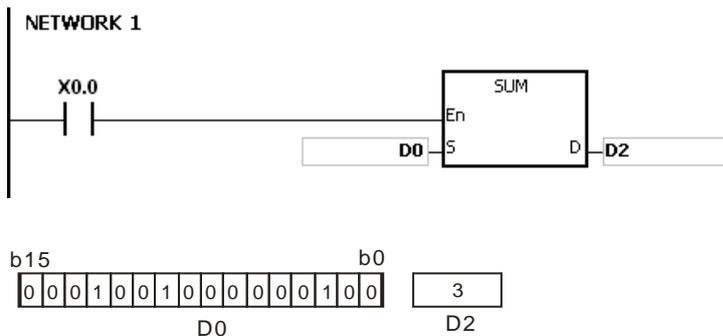


指令說明：

1. 在 **S** 中，所有位元內容為“1”的總數將被儲存於 **D**。
2. 如果來源裝置 **S** 的位元全部為“0”時，零旗標信號 **SM600=ON**。
3. 32 位元指令才可以使用 **HC** 裝置。

程式範例：

當 X0.0 為 ON 時，D0 的 16 個位元中，內容為“1”的位元總數被存於 D2 當中。



補充說明：

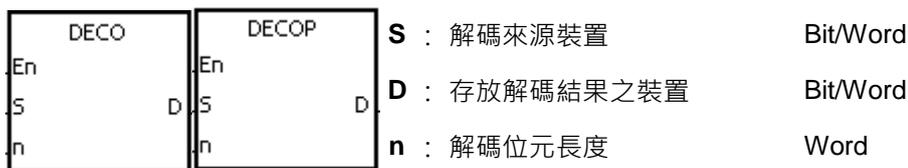
裝置超出範圍時，指令不執行，**SM0=ON**，**SR0 Error Code=16# 2003**。

API	指令碼			運算元								功能					
1202	DECO	P		S · D · n								解碼器					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

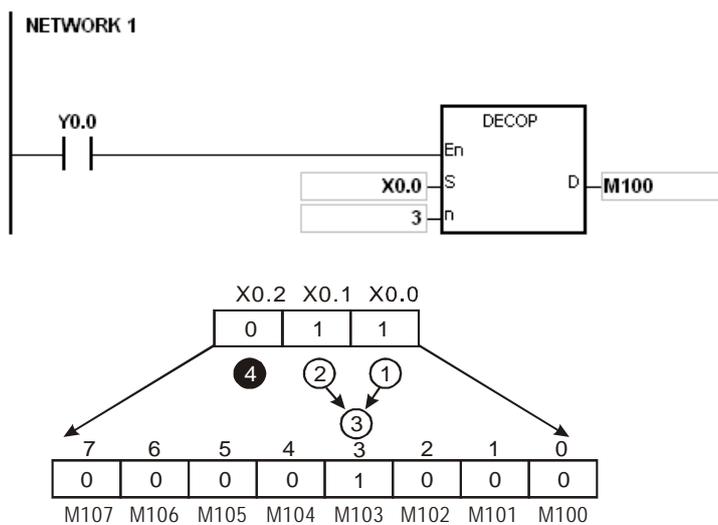


指令說明：

1. 來源裝置 **S** 的下位“n”位元作解碼，並將其“2ⁿ”位元長度的結果存於 **D**。
2. 當 **D** 為位元裝置時，**n** = 1~8，當 **n**=8 時，可做最大解碼 2⁸=256 點。(須注意解碼後的裝置儲存範圍，勿重複使用)。
3. **D** 為字元裝置時，**n** = 1~4，當 **n**=4 時，可做最大解碼 2⁴=16 點。
4. 本指令一般都是使用脈波執行型指令 (DECOP)。

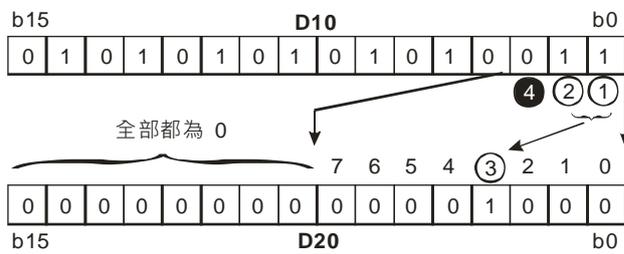
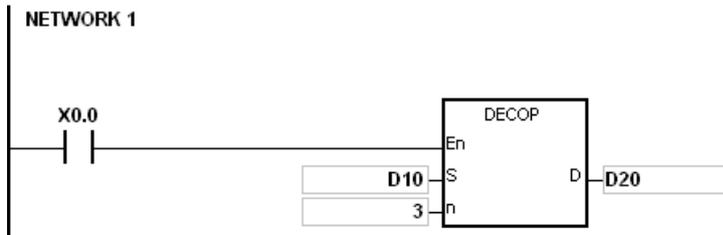
程式範例一：

1. Y0.0 = OFF→ON 時，DECO 指令將 X0.0~X0.2 的內容值解碼到 M100~M107。
2. 當資料源 X0.0~X0.2 的內容值為 1+2 = 3 時，從 M100 開始算第 3 個位元 M103 設定為 1。
3. 當 DECO 指令執行過後，而 Y0.0 變為 OFF，已經做解碼輸出者照常動作。



程式範例二：

1. X0.0 = OFF→ON 時，DECO 指令將 D10 中 (b2~b0) 的內容值解碼到 D20 的 (b7~b0)。D20 中未被使用之位元 (b15~b8) 全部變為 0。
2. D10 的下位 3 位元作解碼存放於 D20 之下位 8 位元，上 8 位元皆為 0。
3. 當 DECO 指令執行過後，而 X0.0 變為 OFF 後，已經做解碼輸出者照常動作。



補充說明：

1. D 為位元裝置時，若 $n < 1$ 或 $n > 8$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
2. D 為字元裝置時，若 $n < 1$ 或 $n > 4$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. 若 S 為位元裝置時，S+n-1 位元超過範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. 若 D 為位元裝置時，D+(2^n)-1 位元超過範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

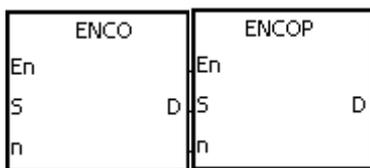
6

API	指令碼			運算元								功能					
1203	ENCO	P		S · D · n								編碼器					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：



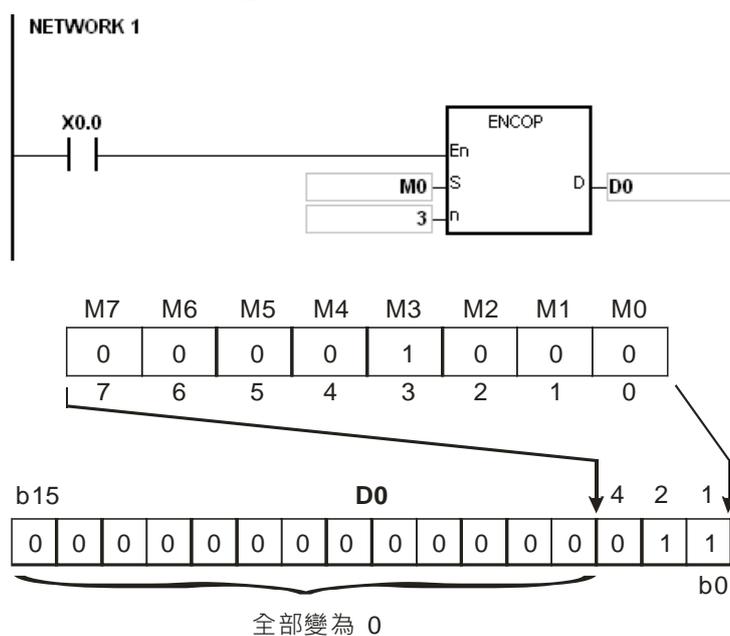
- S : 編碼來源裝置 Bit/Word
- D : 存放編碼結果之裝置 Word
- n : 編碼位元長度 Word

指令說明：

- 來源裝置 S 的下位“2ⁿ”位元長度的資料作編碼，並將結果存於 D。
- 若資料來源 S 有多數位元為 1 時，則處理由高位元往低位元的第 1 個為 1 之位元。
- S 為位元裝置時，n = 1~8，當 n=8 時，可做 2⁸=256 點編碼。
- S 為字元裝置時，n = 1~4，當 n=4 時，可做 2⁴=16 點編碼。
- 本指令一般都是使用脈波執行型指令 (ENCOP)。

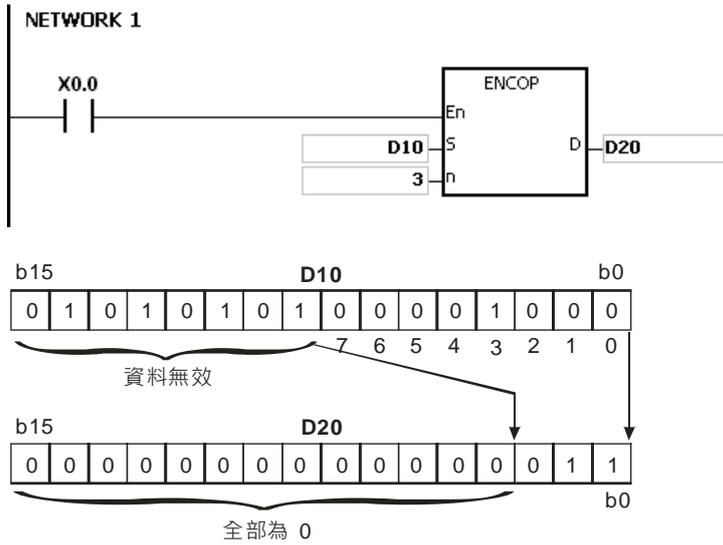
程式範例一：

- 當 X0.0 = OFF → ON 時，ENCO 指令將 2³ 位元資料 (M0~M7) 編碼存放於 D0 之下位 3 位元 (b2~b0) 內，D0 中未被使用之位元 (b15~b3) 全部變為 0。
- 當 ENCO 指令執行過後，而 X0.0 變為 OFF 後，D 內資料不變。



程式範例二：

1. 當 X0.0 = OFF → ON 時，D10 內 2³ bits 資料 (b0~b7) 編碼存放於 D20 之下位 3 位元 (b2~b0) 內，D20 中未被使用之位元 (b15~b3) 全部變為 0。(D10 內 b8~b15 為無效資料)
2. 當 ENCO 指令執行過後，而 X0.0 變為 OFF，D 內資料不變。



補充說明：

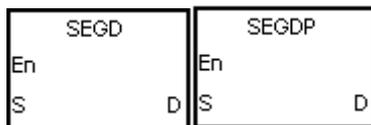
1. 若資料來源 **S** 都沒有位元為 1 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **S** 為位元裝置時，若 $n < 1$ 或 $n > 8$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
3. **S** 為字元裝置時，若 $n < 1$ 或 $n > 4$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
4. 若 **S** 為位元裝置時， $S + (2^n) - 1$ 位元超過範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
5. 若 **D** 為位元裝置時， $D + n - 1$ 位元超過範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

6

API	指令碼			運算元								功能					
1204	SEGD	P		S · D								7 段顯示器解碼					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



S：欲解碼之來源裝置 Word

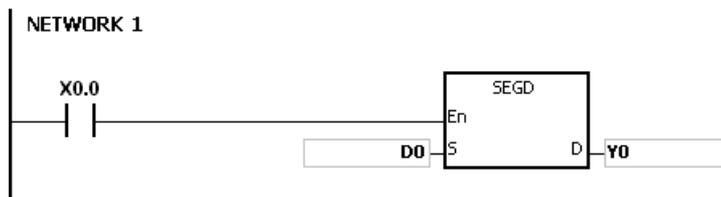
D：解碼後之輸出裝置 Word

指令說明：

來源裝置 **S** 的下位 4 個位元 (b0~b3) · 被解碼成 7 段顯示器的格式 · 並將結果存於 **D**。

程式範例：

當 X0.0=ON 時 · D0 的下位 4 個位元 (b0~b3) 的內容 (0~F : 16 進制) 被解碼成 7 段顯示器輸出 · 解碼的結果暫存於 Y0.0~Y0.15 當中 · 若指定資料超出 4 個位元 · 仍取下位 4 個位元的內容解碼。



七段顯示器解碼表：

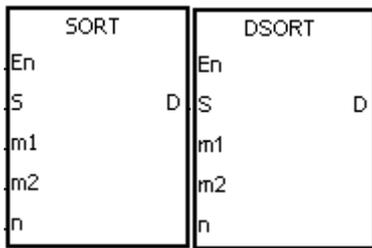
16進數	位元組合	七段顯示器的構成	各節段狀態						顯示資料	
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)		B6(g)
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	ON	OFF	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	b
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	C
D	1101		OFF	ON	ON	ON	ON	OFF	ON	d
E	1110		ON	OFF	OFF	ON	ON	ON	ON	E
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	F

API	指令碼			運算元							功能						
1205	D	SORT		S · m₁ · m₂ · D · n							資料排序						

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

脈波執行型	16 位元指令 (11 steps)	32 位元指令 (11 steps)
-	AH500	AH500

符號：



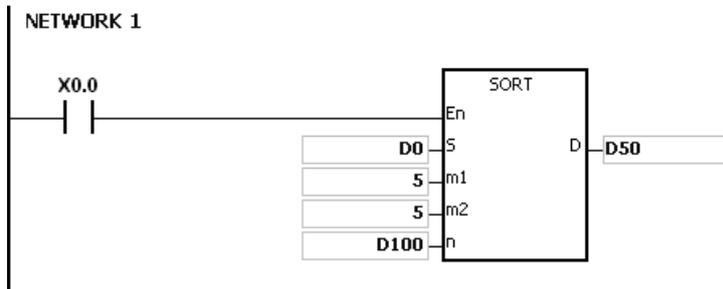
- S** : 原始資料區塊之起始裝置 Word/Double Word
- m₁** : 被排序之資料組數 Word/Double Word
- m₂** : 每筆資料之欄位數 Word/Double Word
- D** : 存放排序結果資料區塊之起始裝置 Word/Double Word
- n** : 資料排序的參考值 Word/Double Word

指令說明：

1. 排序結果顯示於 **D** 所指定的起始號碼開始算的 **m₁ × m₂** 個暫存器當中，因此，**S** 與 **D** 指定同一個暫存器的話，排序結果將與原來被排序的資料 **S** 內容相同。
2. **m₁ = 1~32**，**m₂ = 1~6**，**n = 1~m₂**。
3. **SM604=OFF** 時為升冪排序，**SM604=ON** 時為降冪排序。
4. 32 位元指令才可以使用 **HC** 裝置。

程式範例：

1. 將 **SM604=OFF**，當 **X0.0=OFF→ON** 時，指定執行資料進行升冪排序作業。



6

2. 排序資料構成例

		資料數： m_2 個				
		資料欄位				
	行	1	2	3	4	5
	列	學生編號	國文	英文	數學	理化
↑ 資料 個數 : m_1 個 ↓	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 時的排序後資料

		資料數： m_2 個				
		資料欄位				
	行	1	2	3	4	5
	列	學生編號	國文	英文	數學	理化
↑ 資料 個數 : m_1 個 ↓	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

4. D100=5 時的排序後資料。

		資料數：m ₂ 個				
		資料欄位				
	行	1	2	3	4	5
	列	學生編號	國文	英文	數學	理化
↑ 資料 個 數 : m ₁ 個 ↓	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

補充說明：

1. 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. m₁、m₂ 或 n 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

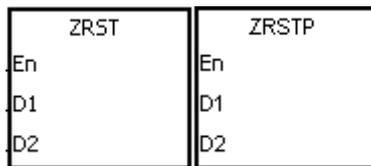
6

API	指令碼		運算元				功能					
1206	ZRST	P	D ₁ · D ₂				區域清除					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令
AH500	AH500	-

符號：



D₁：區域清除起始裝置 Bit/Word

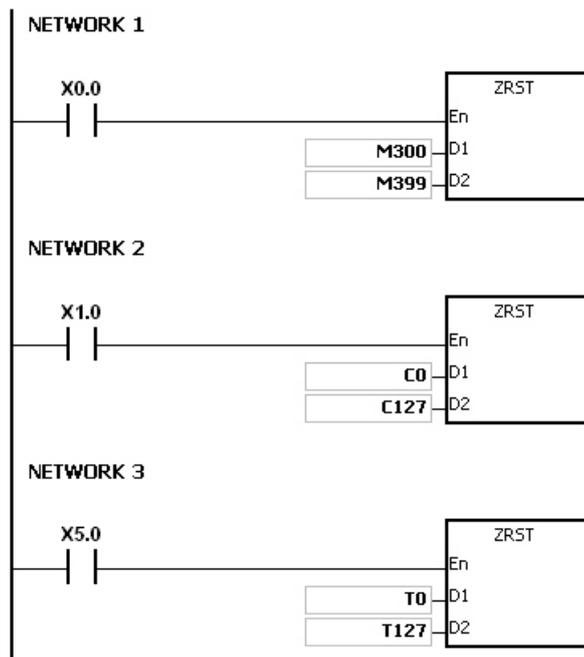
D₂：區域清除結束裝置 Bit/Word

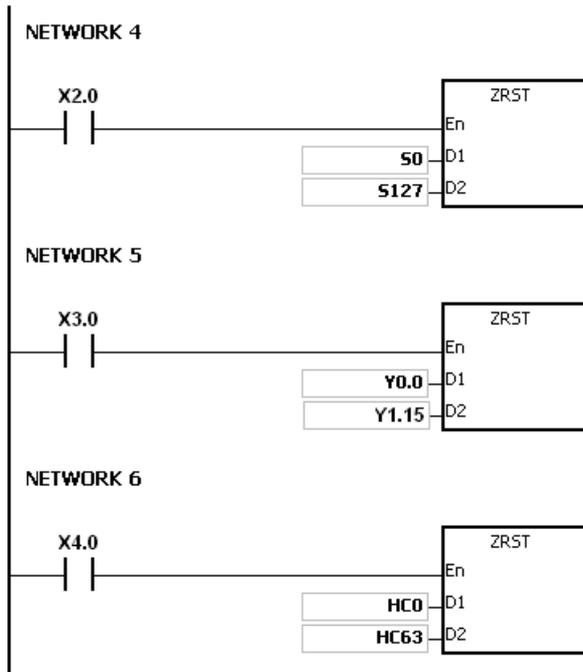
指令說明：

1. 清除 D₁~D₂ 的運算元內容值。
2. 當 D₁ 運算元編號 > D₂ 運算元編號時，只有 D₂ 指定之運算元被清除。

程式範例：

1. 當 X0.0 為 ON 時，輔助繼電器 M300~M399 被清除成 OFF。
2. 當 X1.0 為 ON 時，16 位元計數器 C0~C127 全部清除。(寫入 0，並將接點及線圈清除成 OFF)。
3. 當 X5.0 為 ON 時，計時器 T0~T127 全部清除。(寫入 0，並將接點及線圈清除成 OFF)。
4. 當 X2.0 為 ON 時，步進點 S0~S127 被清除成 OFF。
5. 當 X3.0 為 ON 時，資料暫存器 Y0.0~Y1.15 被清除成 OFF。
6. 當 X4.0 為 ON 時，32 位元計數器 HC0~HC63 全部清除。(寫入 0，並將接點及線圈清除成 OFF)。





補充說明：

1. D_1 、 D_2 運算元裝置類別不相同，指令不執行，SM0=ON，錯誤碼 SR0=16#2007。
2. D_1 、 D_2 運算元資料格式不相同，指令不執行，SM0=ON，錯誤碼 SR0=16#2007。

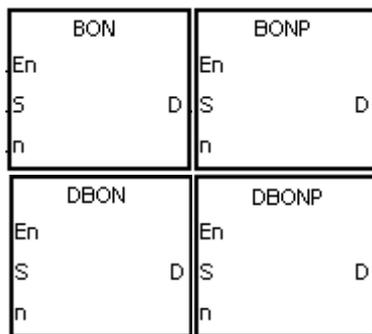
6

API	指令碼			運算元								功能				
1207	D	BON	P	S · D · n								ON 位元判定				

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



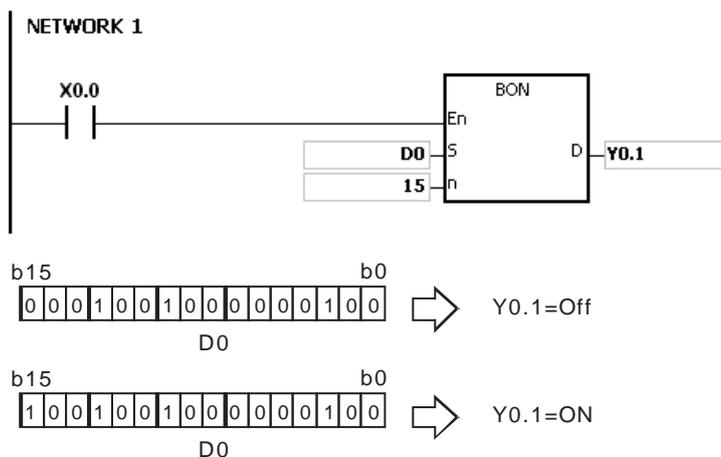
- S**：來源裝置 Word/Double Word
- D**：存放判定結果之裝置 Bit
- n**：指定判定之位元 (自 0 開始編號) Word/Double Word

指令說明：

- 判斷 **S** 內容值的第 **n** 個位元的狀態，並將結果存放到 **D**。
- 16 位元指令 **n=0~15**，32 位元指令 **n=0~31**。
- 32 位元指令才可以使用 **HC** 裝置。

程式範例：

- 當 X0.0=ON 時，若是 D0 的第 15 個位元為“1”時，Y0.1=ON，為“0”時，Y0.1=OFF。
- X0.0 變成 OFF 時，Y0.1 仍保持之前的狀態。



補充說明：

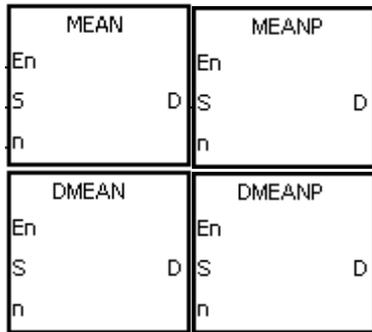
n 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元							功能						
1208	D	MEAN	P	S · D · n							平均值						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：



- S** : 欲取平均值之起始裝置 Word/Double Word
- D** : 存放平均值之裝置 Word/Double Word
- n** : 取平均值之裝置個數 Word/Double Word

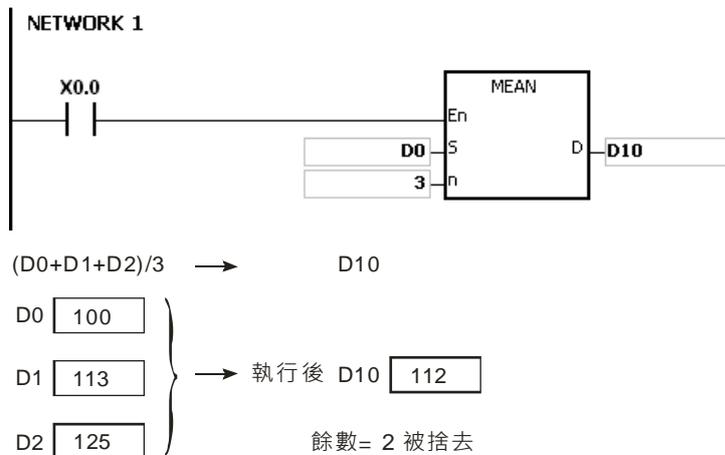
指令說明：

1. 將 **S** 起始之 **n** 個裝置內容值相加後取平均值存入 **D** 中。
2. 如果計算中出現餘數時，餘數會被捨去。
3. 16 位元指令中，**n** 運算元的內容值為 1~256；32 位元指令中，**n** 運算元的內容值為 1~128。
4. 32 位元指令才可以使用 HC 裝置。

6

程式範例：

當 X0.0=ON 時，D0 開始算的 3 個 (n=3) 暫存器的內容全部相加，相加之後再除以 3 以求得平均值並存於指定的 D10 當中，餘數被捨去。



補充說明：

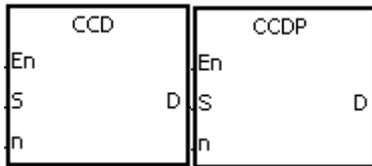
1. 16 位元指令中，**n**<1 或 **n**>256 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

2. 32 位元指令中， $n < 1$ 或 $n > 128$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
3. 若 $S+n-1$ 裝置，超過裝置範圍時， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。

API	指令碼			運算元							功能						
1209		CCD	P	S · D · n							總和檢查						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●	○	●	○	○		

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：



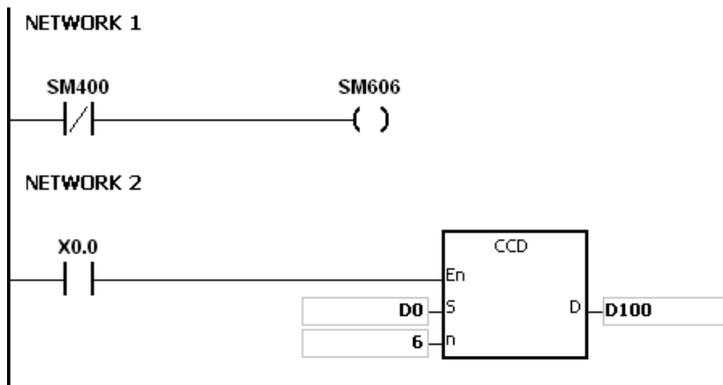
- S**：資料來源起始裝置 Word
- D**：存放總和檢查之結果 Word
- n**：資料個數 Word

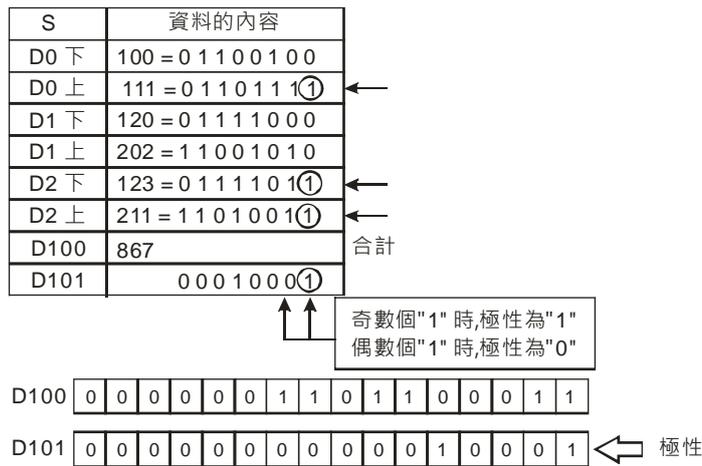
指令說明：

1. 本指令用來作通信時，為了確保資料傳輸時之正確性所做的字串總和檢查 (Sum Check)。
2. 16 位元轉換模式：當 SM606=OFF 時，指定為 16 位元轉換模式。將 **S** 所指定暫存器起始號碼開始算的 **n** 個資料 (以 8 位元為單位) 內容作加總，加總結果存放於 **D** 所指定的暫存器當中，而極性位元存放於 **D+1** 當中。
3. 8 位元轉換模式：當 SM606=ON 時，指定為 8 位元轉換模式。將 **S** 所指定暫存器起始號碼開始算的 **n** 個資料 (以 8 位元為單位，只有下 8 位元有效) 內容作加總，加總結果存放於 **D** 所指定的暫存器當中，而極性位元存放於 **D+1** 當中。
4. **n=1~256**。

程式範例一：

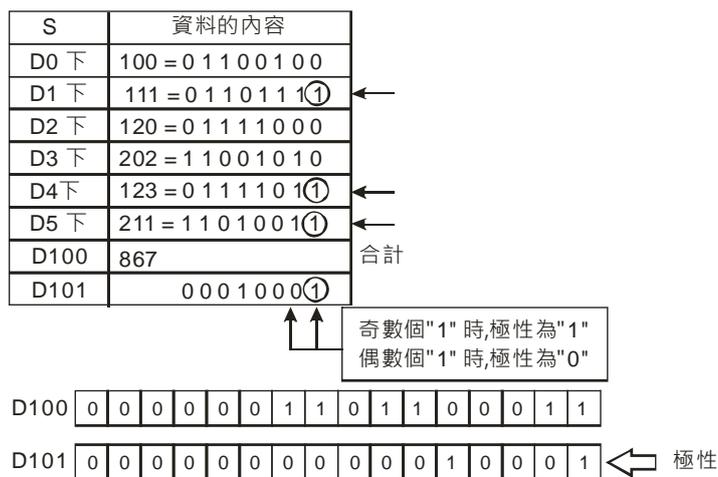
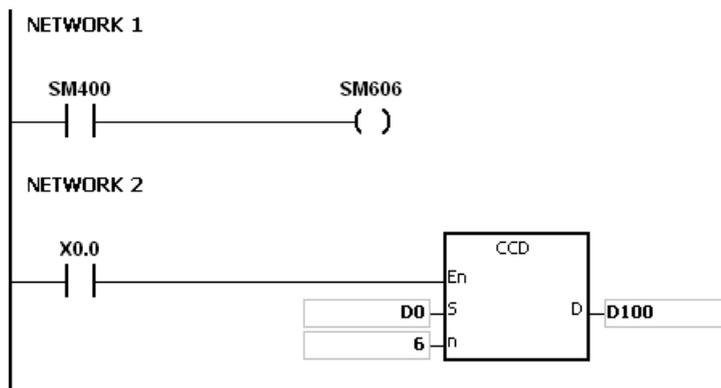
1. 當 SM606=OFF 時，指定為 16 位元轉換模式。
2. 當 X0.0=ON 時，將 D0 所指定暫存器起始號碼開始算的 6 個資料 (以 8 位元為單位 n=6 代表指定 D0~D2) 內容作加總，加總結果存放於 D100 所指定的暫存器當中，而極性位元存放於 D101 當中。





程式範例二：

1. 當 SM606=ON 時，指定為 8 位元轉換模式。
2. 當 X0.0=ON 時，將 D0 所指定暫存器起始號碼開始算的 6 個資料 (以 8 位元為單位 n=6 代表指定 D0~D5) 內容作加總，加總結果存放於 D100 所指定的暫存器當中，而極性位元存放於 D101 當中。



補充說明：

1. SM606=ON (8 位元模式) 時，S+n-1 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

2. SM606=OFF (16 位元模式) 時，S+n/2-1 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. n<1 或 n>256 時，指令不執行，SM0= ON，錯誤碼 SR0=16#200B。
4. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。

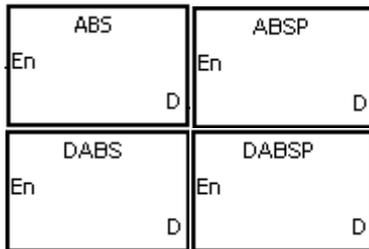
6

API	指令碼			運算元					功能				
1210	D	ABS	P	D					絕對值				

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

脈波執行型	16 位元指令 (3 steps)	32 位元指令 (3 steps)
AH500	AH500	AH500

符號：



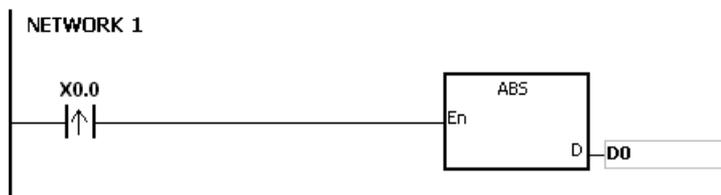
D：欲取絕對值之裝置 Word/Double Word

指令說明：

1. 當 ABS 指令執行時，被指定的元件 D 取絕對值。
2. 本指令一般都是使用脈波執行型指令 (ABSP)。
3. 32 位元指令才可以使用 HC 裝置。

程式範例：

假設執行前 D0=-1234，當 X0.0=OFF→ON 時，D0 內容取絕對值，執行後 D0=1234。



API	指令碼		運算元				功能					
1211	MINV	P	S · D · n				矩陣反相					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：



- S**：矩陣來源裝置 Word
- D**：運算結果 Word
- n**：陣列長度 Word

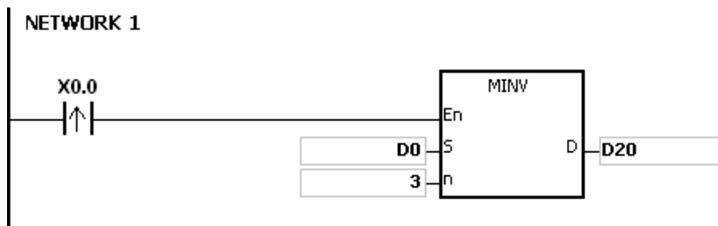
指令說明：

1. 矩陣來源 **S** 依陣列長度 **n** 作矩陣的反相運算並將結果存於 **D**。
2. **n** 運算元的範圍值為 1~256。

程式範例：

當 X0.0=ON 時，16 位元暫存器 D0~D2 共 3 列作 MINV，矩陣反相運算，將結果存於 16 位元暫存器 D20~D22 共 3 列中。

6



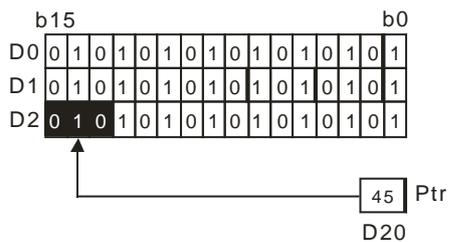
	b15																	b0				
D0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
D2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

執行後 ↓

	b15																	b0				
D20	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D21	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
D22	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

補充說明：

1. **S+n-1**、**D+n-1** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n<1** 或 **n>256** 時，指令不執行，SM0=ON，SR0 Error Code=16#200B。



補充說明：

1. **S+n-1** 裝置超出範圍時，指令不執行，SM0=ON，SR0 Error Code=16#2003。
2. **n<1** 或 **n>256** 時，指令不執行，SM0=ON，SR0 Error Code=16#200B。
3. 旗標信號說明：
 - SM608： 矩陣搜尋結束旗標，當比較到最後一個 bit 時，SM608=1
 - SM611： 矩陣指標錯誤旗標，指標 Pr 值超出此範圍則 SM611=1
 - SM612： 矩陣指標遞增旗標，將指標目前值+1
 - SM613： 矩陣指標清除旗標，將指標目前值清除為 0
 - SM614： 矩陣旋轉位移輸出進位旗標

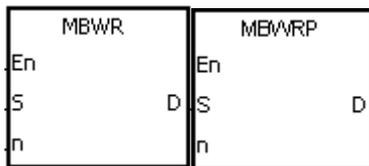
6

API	指令碼			運算元							功能				
1213	MBWR	P		S · n · D							矩陣位元寫入				

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
AH500	AH500	-

符號：



- S**：矩陣來源裝置 Word
- n**：陣列長度 Word
- D**：指標 Pr，用以存放目標之位置值 Word

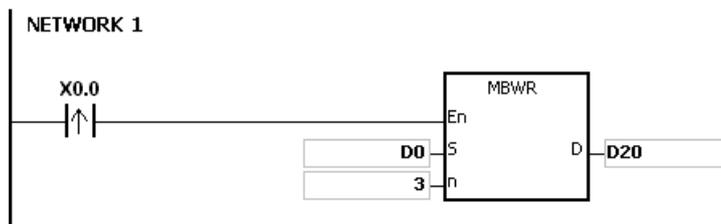
指令說明：

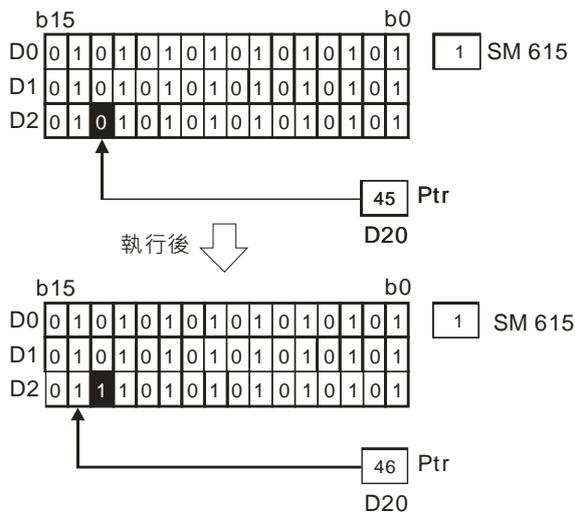
- 當指令執行時，一開始判斷矩陣指標清除旗標 SM613 是否為 ON，若為 ON，指標 D 清除為 0，把矩陣位移輸入補位旗標 SM615 的值，由指標 D 所指定的位元寫入 S 矩陣，當寫完一個 bit 時，判斷矩陣指標遞增旗標 SM612 是否 ON，若 ON 把指標 D 的值加 1，然後結束本指令之執行。
- 當寫到最後一個 bit 時矩陣搜尋結束旗標 SM608=ON，指標 D 記錄著讀取的 bit 之編號，然後結束本指令之執行。若 D 的值超過範圍則 SM611=ON。
- n 運算元的範圍值為 1~256。
- 矩陣之指標 Pr (Pointer)，由使用者於指令中指定，其有效範圍為 0~16n-1，分別對應至矩陣中之位元 b0~b16n-1。若 Pr 值超出此範圍則矩陣指標錯誤旗標 SM611 設為 1，且本指令不執行。

6

程式範例：

- 當 X0.0 由 OFF→ON 時，設指標清除旗標 SM613=0、矩陣指標遞增旗標 SM612=1，所以每寫入一次指標 Pr 增加 1。
- 設指標當時值 D20=45，矩陣位移輸入補位旗標 SM615 狀態為 1 當 X0.0 由 OFF→ON 時動作 1 次，可得到如下執行結果。D20=45，矩陣位移輸入補位旗標 SM615=1，矩陣搜尋結束旗標 SM608=0。





補充說明：

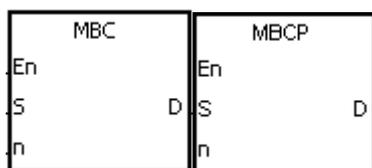
1. **S+n-1** 裝置超出範圍時，指令不執行，SM0=ON，SR0 Error Code=16#2003。
2. **n<1** 或 **n>256** 時，指令不執行，SM0=ON，SR0 Error Code=16#200B。
3. 旗標信號說明：
 - SM608：矩陣搜尋結束旗標，當比較到最後一個 bit 時，SM608=1
 - SM611：矩陣指標錯誤旗標，指標 Pr 值超出此範圍則 SM611=1
 - SM612：矩陣指標遞增旗標
 - SM613：矩陣指標清除旗標
 - SM615：矩陣位移輸入補位旗標

API	指令碼			運算元								功能					
1214		MBC	P	S · n · D								矩陣位元狀態計數					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：



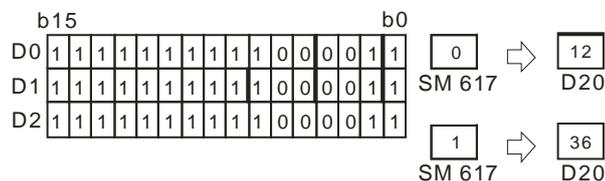
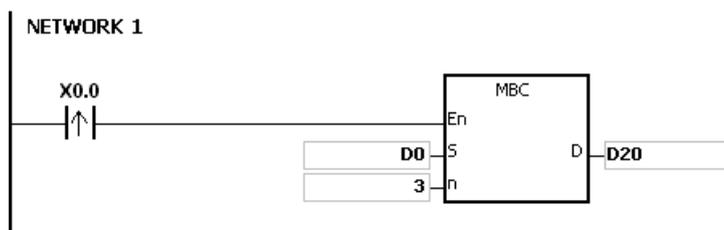
S：矩陣來源裝置 Word
n：陣列長度 Word
D：運算結果 Word

指令說明：

1. 依陣列長度 **n** 計算 **S** 矩陣中所有位元為 1 或為 0 的個數，並將數目存於 **D** 中。
2. 當 **SM617=1** 時計算矩陣位元為 1 的個數，**SM617=0** 時，計算矩陣位元為 0 的個數。當計算出來的結果為 0 時，**SM618=1**。
3. **n** 運算元的範圍值為 1~256。

程式範例：

當 X0.0 為 ON 時，D0~D2 的矩陣中，當 **SM617=1** 時計算矩陣位元為 1 的位元總數被存於 D20 當中。當 **SM617=0** 時計算矩陣位元為 0 的位元總數被存於 D20 當中。



補充說明：

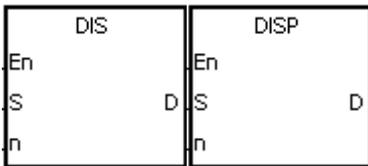
1. **S+n-1** 裝置超出範圍時，指令不執行，**SM0=ON**，**SR0 Error Code=16#2003**。
2. **n<1** 或 **n>256** 時，指令不執行，**SM0=ON**，**SR0 Error Code=16#200B**。
3. 旗標信號說明：
SM617： 矩陣計數位元為 0 或位元為 1 旗標
SM618： 矩陣計數結果為 0 時 ON

API	指令碼			運算元							功能				
1215		DIS	P	S · n · D							16 位元資料的 4 位元分組				

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

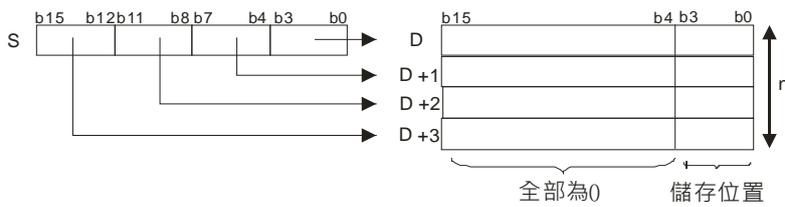
符號：



S：資料來源裝置 Word
n：儲存長度 Word
D：運算結果 Word

指令說明：

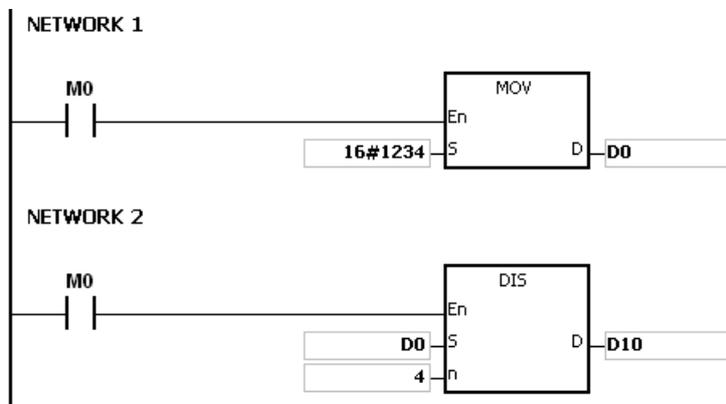
- 將 **S** 所指定暫存器內容值，從 b_0 開始以 4 個位元為 1 組，分別存放於 **D~D+(n-1)** 所指定的暫存器的低 4 位元中 ($b_0 \sim b_3$)。

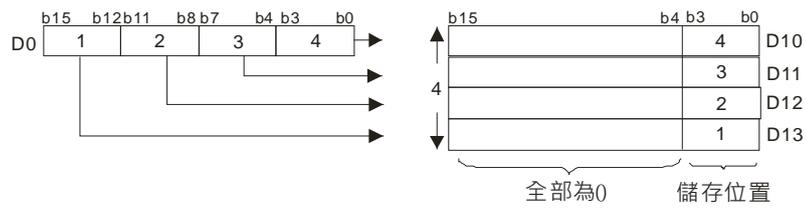


- $n=1 \sim 4$ 。

程式範例：

D0 的內容為 16#1234。啟動 M0，執行 DIS 指令時，將 D0 的 $b_0 \sim b_{15}$ 內容分為 4 個 Bit 一組，分別移動至 D10~D13 的 $b_0 \sim b_3$ 當中。





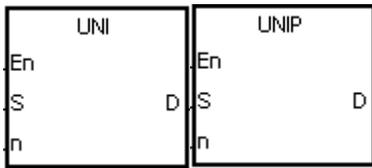
補充說明：

1. $D \sim D + (n - 1)$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n < 1$ 或 $n > 4$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1216		UNI	P	S · n · D								16 位元資料的 4 位元鏈結					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
n	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●		●				

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

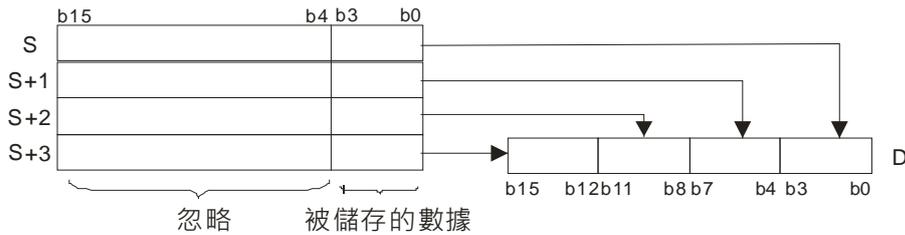
符號：



- S : 資料來源裝置 Word
- n : 資料來源長度 Word
- D : 運算結果 Word

指令說明：

- 將 S~S+ (n-1) 所指定暫存器內容值 (b₀~b₃) · 依續存放於 D (b₀~b₁₅) 所指定的暫存器中。

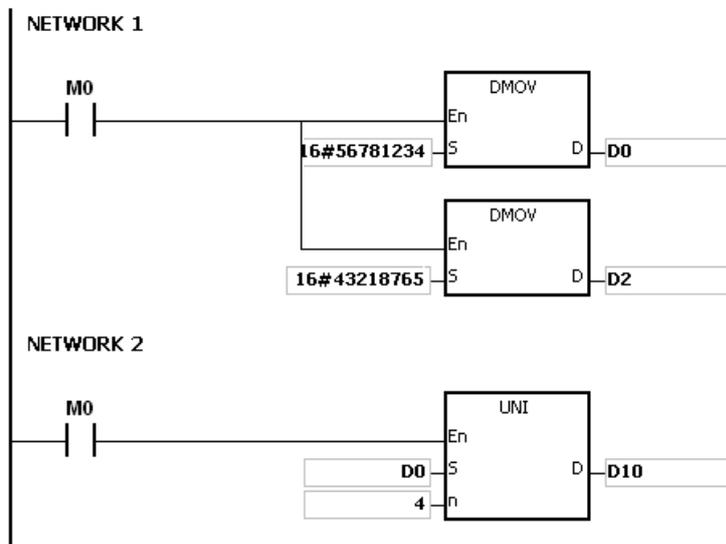


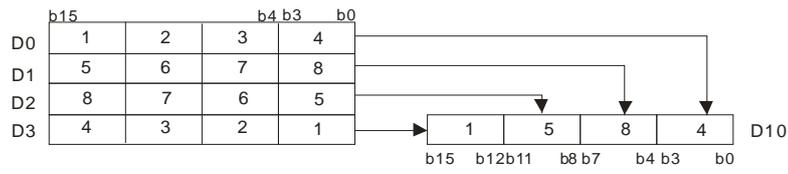
6

- n=1 ~ 4。

程式範例：

D0~D3 的內容為分別為 16#1234、16#5678、16#8765、16#4321。啟動 M0，執行 UNI 指令時，將 D0~D3 的 b₀~b₃ 內容，分別儲存至 D10 的 b₀~b₁₅ 當中。





補充說明：

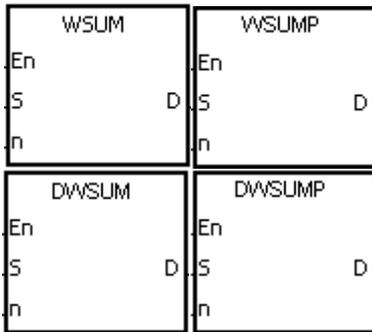
1. **S~S+ (n-1)** 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **n<1** 或 **n>4** 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元							功能						
1217	D	WSUM	P	S · n · D							求和						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

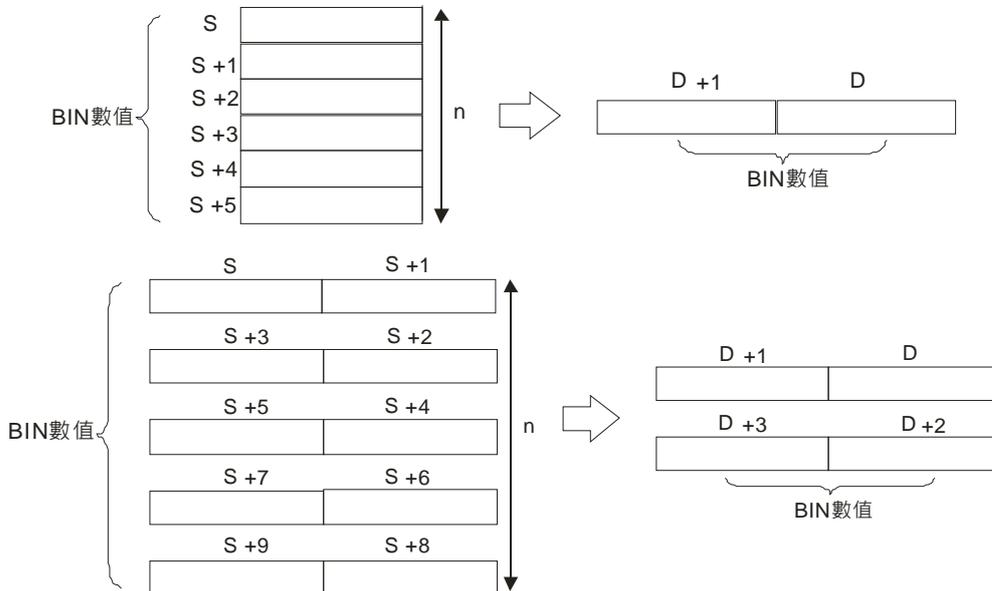


S：資料來源裝置 Word/Double Word
n：資料來源長度 Word/Double Word
D：運算結果 Double Word/Long Word

指令說明：

- 將 **S~S+n-1** 所指定暫存器內容值，以有號數十進制作相加，存放於 **D** 所指定的暫存器中。

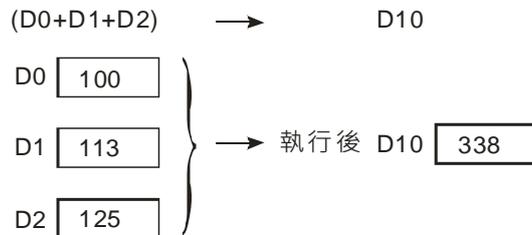
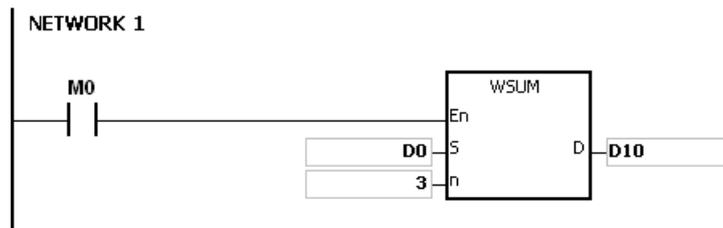
6



- 16 位元指令中，**n** 運算元的內容值為 1~256；32 位元指令中，**n** 運算元的內容值為 1~128。
- 32 位元指令才可以使用 HC 裝置。

程式範例：

當執行 WSUM 指令時，會將 D0~D2 的內容相加後，計算結果儲存至 D10 當中。



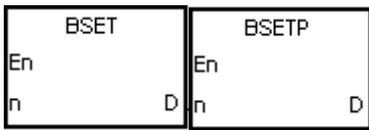
補充說明：

1. 16 位元指令中， $n < 1$ 或 $n > 256$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
2. 32 位元指令中， $n < 1$ 或 $n > 128$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
3. 若 $S+n-1$ 或 D 裝置，超過裝置範圍時， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。

API	指令碼			運算元								功能					
1218		BSET	P	D · n								字元裝置位元動作保持 ON					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●		●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



D：裝置編號

Word

n：要設定的 bit 號碼

Word

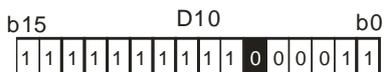
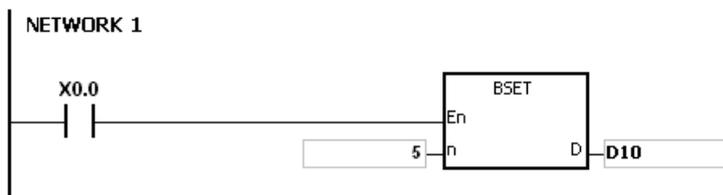
指令說明：

1. 將 D 所指定暫存器，的第 n 個位元設為 1。
2. 當 BSET 指令被驅動，其指定的位元被設定為 ON，且被設定的位元會維持 ON，不管 BSET 指令是否仍被驅動，可利用 BRST 指令將該位元設為 OFF。
3. n=0~15。

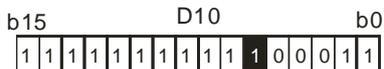
程式範例：

當 X0.0=ON 時，D10 的第 5 個 Bit 會設定為 1。

6



執行後 ↓

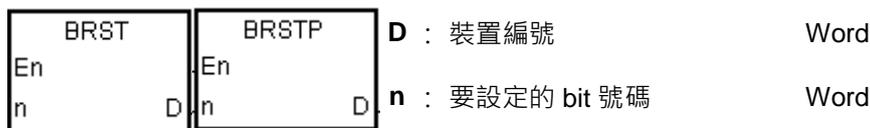


補充說明：

n<0 或 n>15 時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1219		BRST	P	D · n								字元裝置位元清除					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
n	●	●			●	●		●	●		●		●	○	○		
脈波執行型											16 位元指令 (5 steps)			32 位元指令			
AH500											AH500			-			

符號：

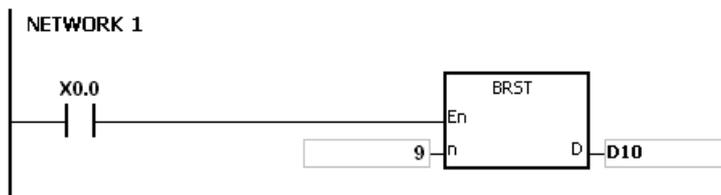


指令說明：

1. 將 D 所指定暫存器 · 的第 n 個位元設為 0。
2. 當 BRST 指令被驅動 · 其指定的位元會被設定為 OFF。
3. n=0~15。

程式範例：

當 X0.0=ON 時 · D10 的第 9 個 bit 會清除為 0。



執行後 ↓



補充說明：

n<0 或 n>15 時 · 指令不執行 · SM0=ON · 錯誤碼 SR0=16#200B。

API	指令碼			運算元								功能					
1220		BKRST	P	D · n								指定區域清除					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●			○	●				
n	●	●	●	●	●	●	●	●	●	●	●	○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：



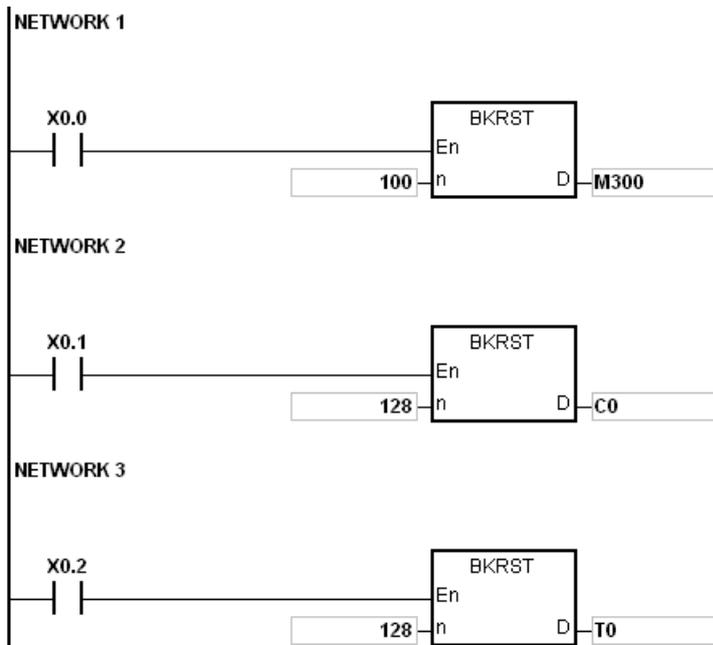
D : 裝置編號 Bit/Word
n : 長度 Word

指令說明：

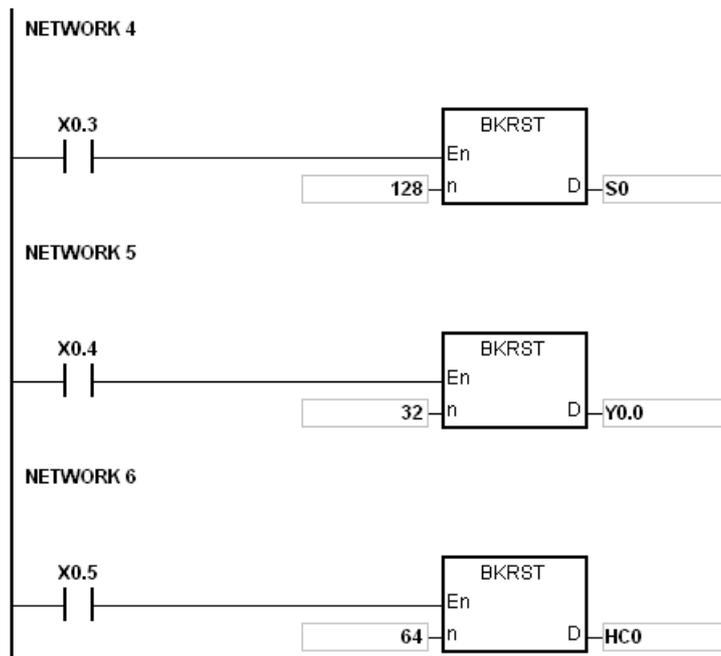
1. 清除 D~D+ (n-1) 的運算元內容值。
2. n=1~1024

程式範例：

1. 當 X0.0 為 ON 時，輔助繼電器 M300~M399 被清除成 OFF。
2. 當 X0.1 為 ON 時，計數器 C0~C127 全部清除。(寫入 0，並將接點及線圈清除成 OFF)。
3. 當 X0.2 為 ON 時，計時器 T0~T127 全部清除。(寫入 0，並將接點及線圈清除成 OFF)。
4. 當 X0.3 為 ON 時，步進點 S0~S127 被清除成 OFF。
5. 當 X0.4 為 ON 時，資料暫存器 Y0.0~Y1.15 被清除成 OFF。
6. 當 X0.5 為 ON 時，計數器 HC0~HC63 全部清除。(寫入 0，並將接點及線圈清除成 OFF)。



6



補充說明：

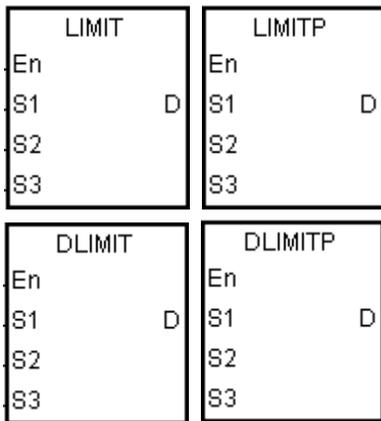
1. $D \sim D + (n - 1)$ 裝置超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. $n < 0$ 或 $n > 1024$ ，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。

API	指令碼			運算元						功能					
1221	D	LIMIT	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$						高低限控制					

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●	●	●	●		●	○	●	○	○		
S_2	●	●			●	●	●	●	●		●	○	●	○	○		
S_3	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
AH500	AH500	AH500

符號：



- S_1 : 最小輸出值 Word/Double Word
- S_2 : 最大輸出值 Word/Double Word
- S_3 : 輸入值 Word/Double Word
- D : 輸出值 Word/Double Word

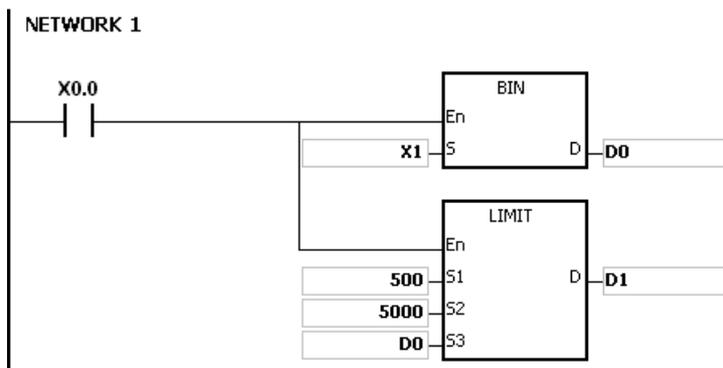
指令說明：

6

- 輸入值 S_3 與最小輸出值 S_1 及最大輸出值 S_2 作比較，比較後將結果存入輸出值 D，說明如下：
 - 若最小輸出值 $S_1 >$ 輸入值 S_3 ，輸出值 D=最小輸出值 S_1 。
 - 若最大輸出值 $S_2 <$ 輸入值 S_3 ，輸出值 D=最大輸出值 S_2 。
 - 若最小輸出值 $S_1 \leq$ 輸入值 $S_3 \leq$ 最大輸出值 S_2 ，輸出值 D=輸入值 S_3 。
 - 若最小輸出值 $S_1 >$ 最大輸出值 S_2 ，輸出值 D 不執行。
- 32 位元指令才可以使用 HC 裝置。

程式範例：

- 當 X0.0=ON 時，X1 的裝置狀態轉為 BIN 值儲存在 D0，D0 內的數值經 500~5000 範圍限制計算後，將結果儲存到 D1。



最小輸出值	最大輸出值	輸入值 D0	功能	輸出值 D1
500	5000	499	$D0 < 500$	500
		5001	$D0 > 5000$	5000
		600	$500 \leq D0 \leq 5000$	600

補充說明：

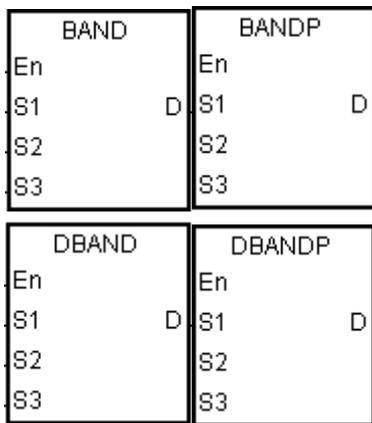
$S_1 > S_2$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元							功能						
1222	D	BAND	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							死區控制						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
AH500	AH500	AH500

符號：

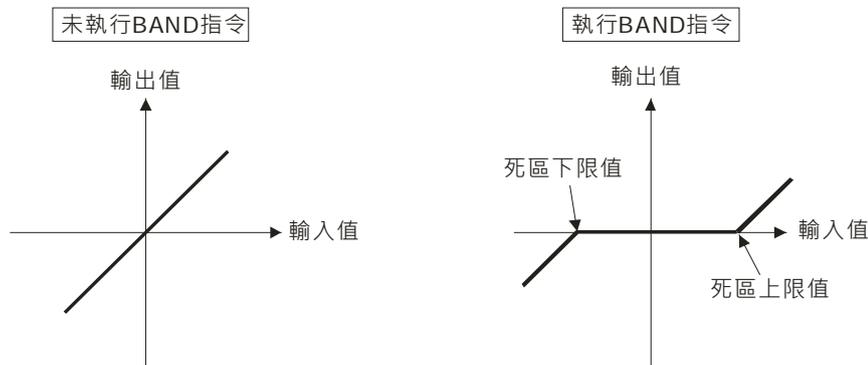


- S_1 : 死區下限值 Word/Double Word
- S_2 : 死區上限值 Word/Double Word
- S_3 : 輸入值 Word/Double Word
- D : 輸出值 Word/Double Word

指令說明：

6

- 輸入值 S_3 與死區下限值 S_1 及死區上限值 S_2 作相減，相減後將結果存入 D。說明如下：
 - 若最小輸出值 $S_1 >$ 輸入值 S_3 ，輸出值 $D = S_3 - S_1$ 。
 - 若最大輸出值 $S_2 <$ 輸入值 S_3 ，輸出值 $D = S_3 - S_2$ 。
 - 若最小輸出值 $S_1 \leq$ 輸入值 $S_3 \leq$ 最大輸出值 S_2 ，輸出值 $D = 0$ 。
 - 若最小輸出值 $S_1 >$ 最大輸出值 S_2 ，輸出值 D 不執行。
- 32 位元指令才可以使用 HC 裝置。
- 圖形說明如下：



- S_1 、 S_2 和 S_3 的範圍：
 - BAND 指令：-32768~32767

當輸出值 **D** 運算結果超過-32768 或 32767，範例說明如下：

死區下限值 $S_1=10$ ，輸入值 $S_3=-32768$

輸出值 $D=-32768-10=16\#8000-16\#000A=16\#7FF6=32758$

- **DBAND** 指令：-2147483648~2147483647

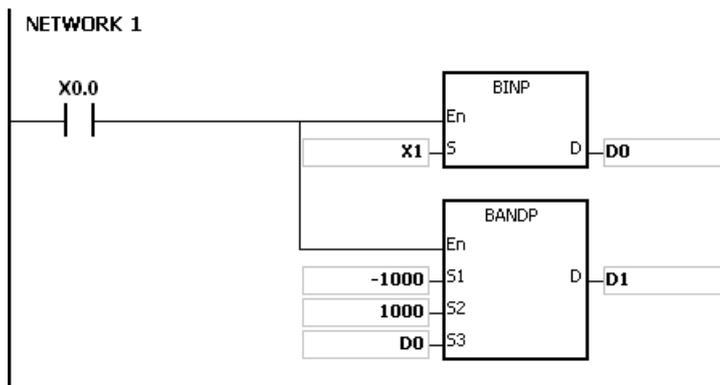
當輸出值 **D** 運算結果超過-2147483648 或 2147483647，範例說明如下：

死區下限值 (S_1+1, S_1) =1000，輸入值 (S_3+1, S_3) =-2147483648

輸出值 ($D+1, D$) =-2147483648-1000=16#80000000-16#000003E8=16#7FFFFC18
=2147482648

程式範例一：

當 X0.0 為 ON 時，X1 的 BCD 數值執行從-1000 至 1000 的範圍控制，將控制結果儲存至 D1 當中。



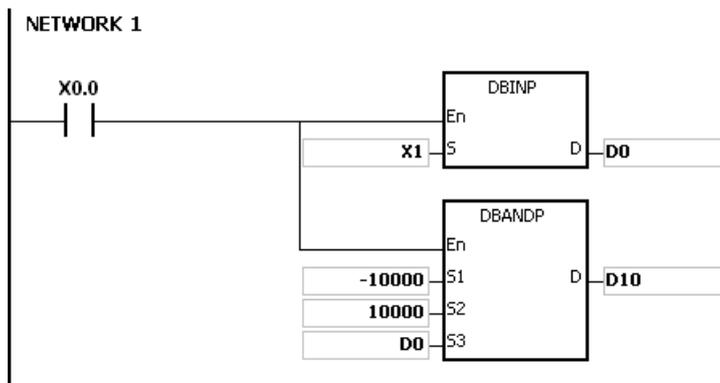
執行結果：

死區下限值	死區上限值	輸入值 D0	功能	輸出值 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

6

程式範例二：

當 X0.0 為 ON 時，(X2 · X1) 的 BCD 數值執行從-10000 至 10000 的範圍控制，將控制結果儲存至 (D11 · D10) 當中。



執行結果：

死區下限值	死區上限值	輸入值 (D1 · D0)	功能	輸出值 (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

補充說明：

$S_1 > S_2$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

6

API	指令碼			運算元						功能					
1223	D	ZONE	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$						區域控制					

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
AH500	AH500	AH500

符號：

ZONE		ZONEP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	

DZONE		DZONEP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	

S_1 ：負偏差值

Word/Double Word

S_2 ：正偏差值

Word/Double Word

S_3 ：輸入值

Word/Double Word

D：輸出值

Word/Double Word

指令說明：

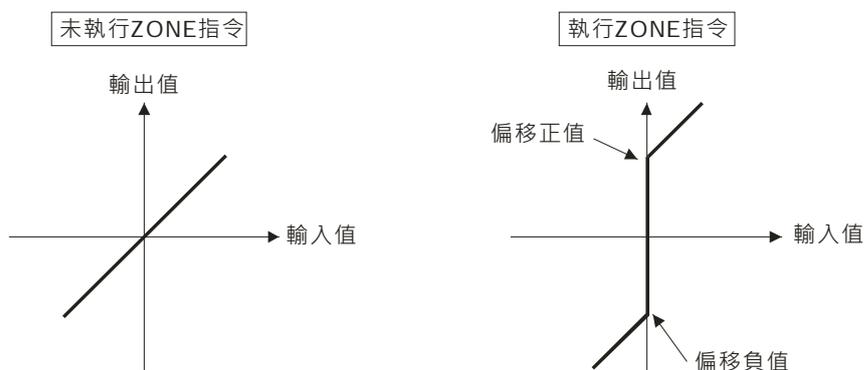
- 輸入值 S_3 與負偏差值 S_1 或正偏差值 S_2 以作相加，並將結果存入 D。說明如下：

若輸入值 $S_3 < 0$ ，輸出值 $D = S_3 + S_1$ 。

若輸入值 $S_3 > 0$ ，輸出值 $D = S_3 + S_2$ 。

若輸入值 $S_3 = 0$ ，輸出值 $D = 0$ 。

- 圖形說明如下：



- 32 位元指令才可以使用 HC 裝置。

- S_1 、 S_2 和 S_3 的範圍：

- ZONE 指令：-32768~32767

當輸出值 D 運算結果超過 -32768 或 32767，範例說明如下：

偏差負值 $S_1 = -100$ · 輸入值 $S_3 = -32768$

輸出值 $D = (-32768) + (-100) = 16\#8000 + 16\#FF9C = 16\#7F9C = 32668$

- DZONE 指令：-2147483648~2147483647

當輸出值 D 運算結果超過 -2147483648 或 2147483647 · 範例說明如下：

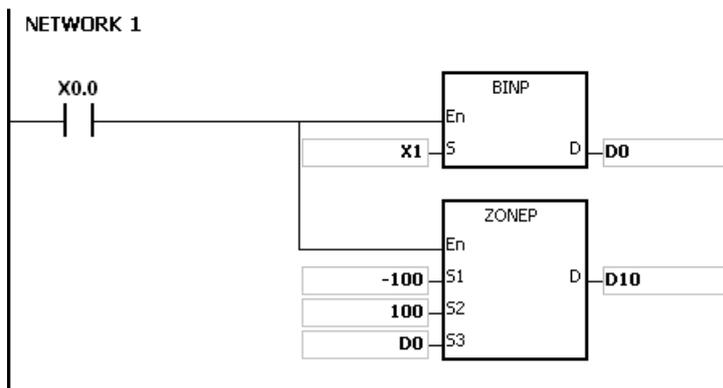
偏差負值 $(S_1 + 1) \cdot S_1 = -1000$ · 輸入值 $(S_3 + 1) \cdot S_3 = -2147483648$

輸出值 $(D + 1) \cdot D = -2147483648 + (-1000)$

$= 16\#80000000 + 16\#FFFFFFC18 = 16\#7FFFFFFC18 = 2147482648$

程式範例一：

當 X0.0 為 ON 時 · X1 的 BCD 數值執行從 -100 至 100 的偏差值相加 · 將偏移結果儲存至 D10 當中。



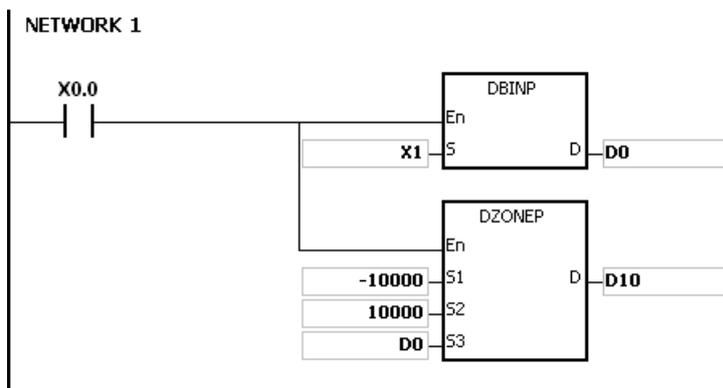
執行結果：

負偏差值	正偏差值	輸入值 D0	功能	輸出值 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

6

程式範例二：

當 X0.0 為 ON 時 · $(X2 \cdot X1)$ 的 BCD 數值執行從 -10000 至 10000 的偏差值相加 · 將偏移結果儲存至 $(D11 \cdot D10)$ 當中。



負偏差値	正偏差値	輸入値 (D1 · D0)	功能	輸出値 (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

6.14 結構建立指令

6.14.1 結構建立指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1300</u>	FOR	–	–	巢串迴路起始	3
<u>1301</u>	NEXT	–	–	巢串迴路結束	1
<u>1302</u>	BREAK	–	✓	強制結束 FOR-NEXT 迴圈	5

6.14.2 結構建立指令說明

API	指令碼			運算元								功能					
1300		FOR		S								巢串迴路起始					

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

脈波執行型	16 位元指令 (3 steps)	32 位元指令
-	AH500	-

符號：



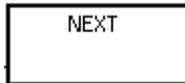
S : 迴路重複執行的次數

WORD

API	指令碼	運算元	功能
1301	NEXT	-	巢串迴路結束

脈波執行型	16 位元指令 (1 step)	32 位元指令
-	AH500	-

符號：



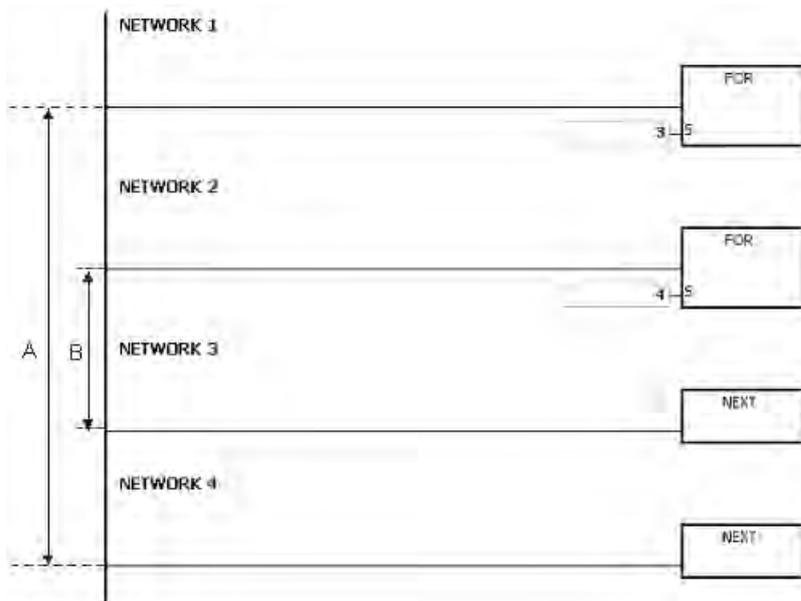
指令說明：

1. 由 FOR 指令指定 FOR~NEXT 迴圈來回執行 N 次後跳出 FOR~NEXT 迴圈往下繼續執行。
2. 指定次數範圍 N=1~32,767，當指定次數範圍 $N \leq 1$ 時，都視為是 1。
3. 當不執行 FOR~NEXT 迴路時，可使用 CJ 指令來跳出迴路。
4. 下列情形會產生錯誤：
 - NEXT 指令在 FOR 指令之前。
 - 有 FOR 指令沒有 NEXT 指令。
 - FEND 或 END 指令之後有 NEXT 指令時。
 - FOR~NEXT 指令個數不同時。
5. 巢串式 FOR~NEXT 迴路最多可使用 32 層，但要注意迴路次數過多時，會使 PLC 掃描時間增加，有可能造成逾時監視計時器動作，而導致錯誤產生。可使用 WDT 指令來改善。

6

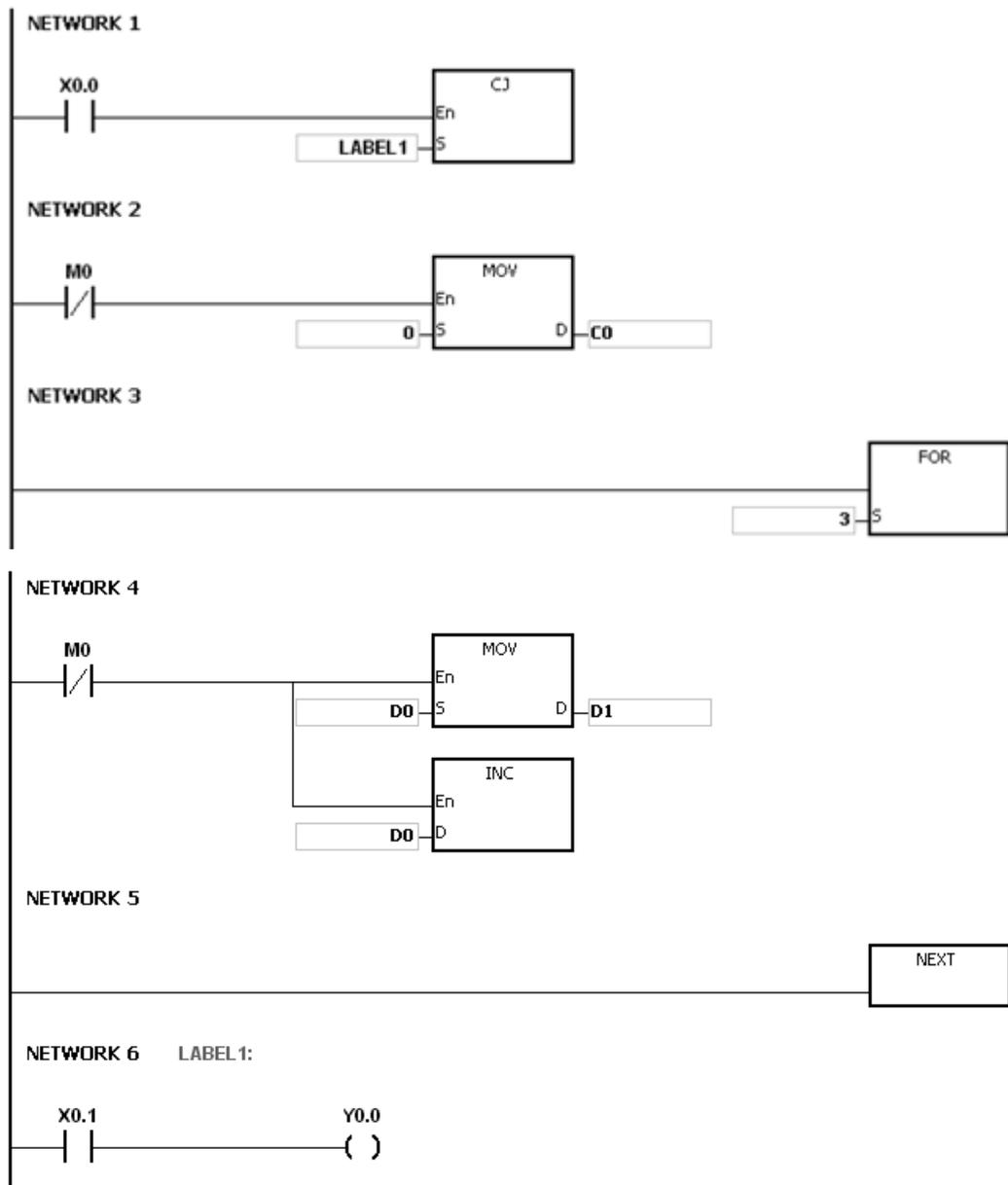
程式範例一：

A 程式執行 3 次後在到 NEXT 指令以後的程式繼續執行。而 A 程式每執行一次 B 程式會執行四次，所以 B 程式合計共執行 $3 \times 4 = 12$ 次。



程式範例二：

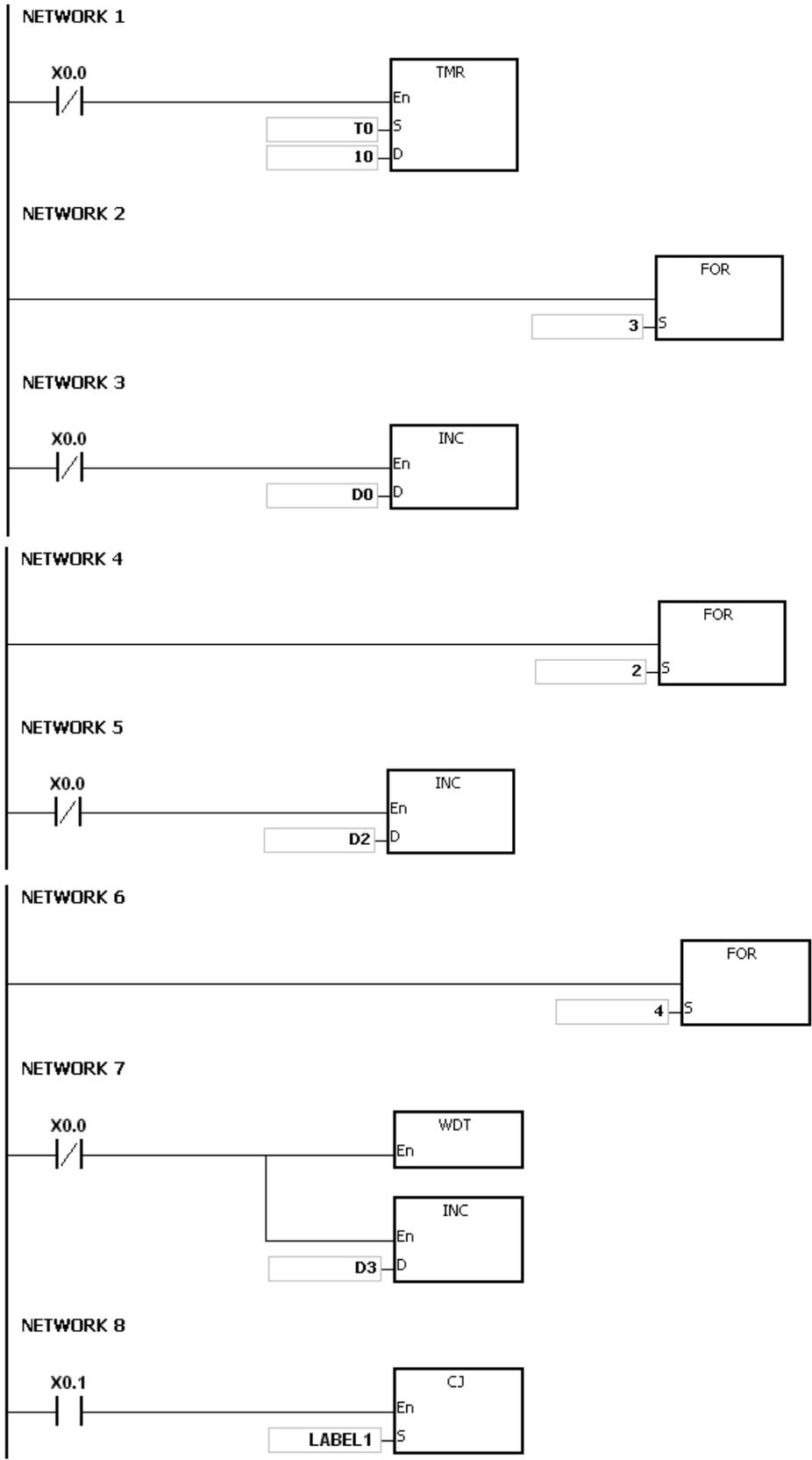
當 X0.0=OFF 時，PLC 會執行 FOR~NEXT 之間的程式，當 X0.0=ON 時，CJ 指令執行跳躍至 LABEL 1：即 NETWORK 6 處，FOR~NEXT 之間的程式 NETWORK 4~ NETWORK 5 跳過不執行。



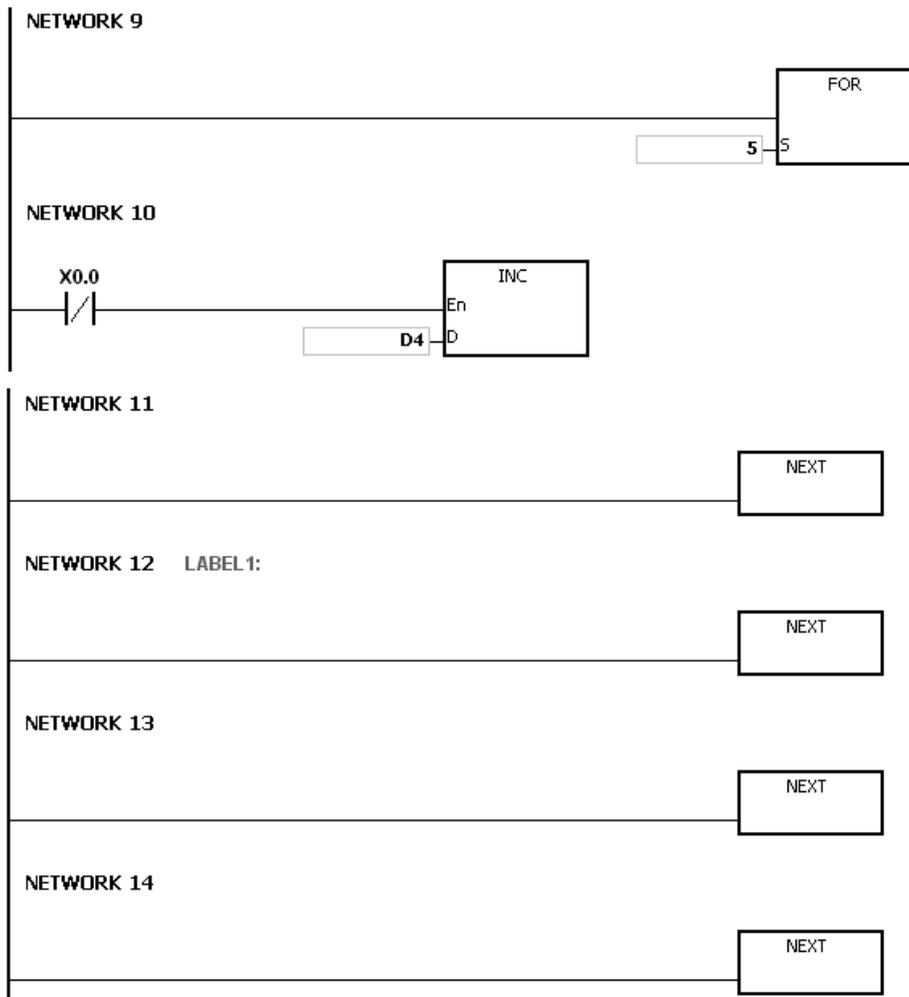
6

程式範例三：

當不執行 FOR~NEXT 時，可使用 CJ 指令來跳躍。NETWORK 8 的 X0.1=ON 時，CJ 指令執行跳躍至 Label 1：即 NETWORK 12 處，NETWORK 9~NETWORK 11 跳過不執行。



6



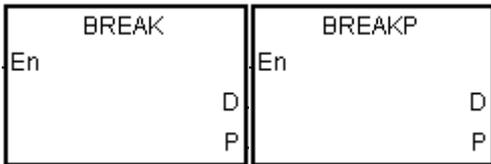
補充說明：

指標 P 在 ISPSOft 中的使用方式，請參考 ISPSOft 的使用手冊說明。

API	指令碼			運算元								功能					
1302		BREAK	P	D · P								強制結束 FOR-NEXT 迴圈					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
D	●	●			●	●		●	●		●	○	●				
P																	

脈波執行型	16 位元指令(3 steps)	32 位元指令
AH500	AH500	-

符號：



D：儲存剩餘未執行完的迴圈數 WORD

P：強制結束迴圈後要執行的指標 POINTER

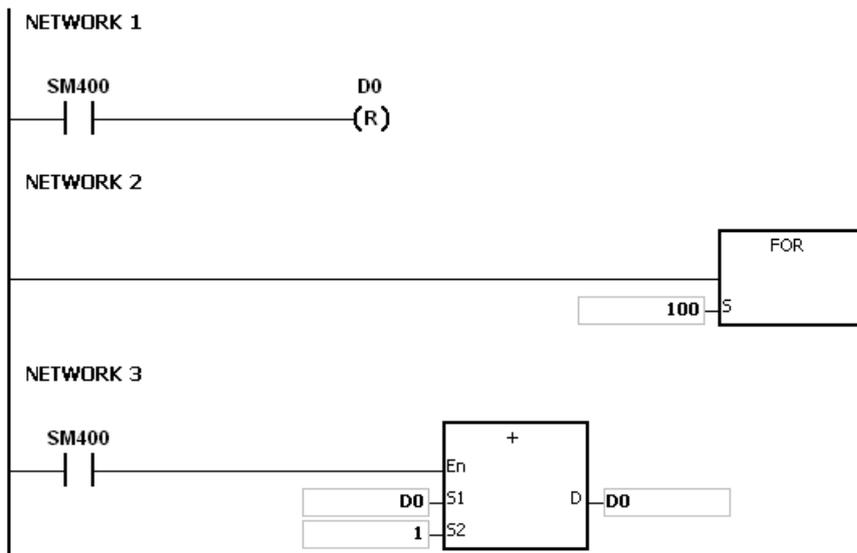
指令說明：

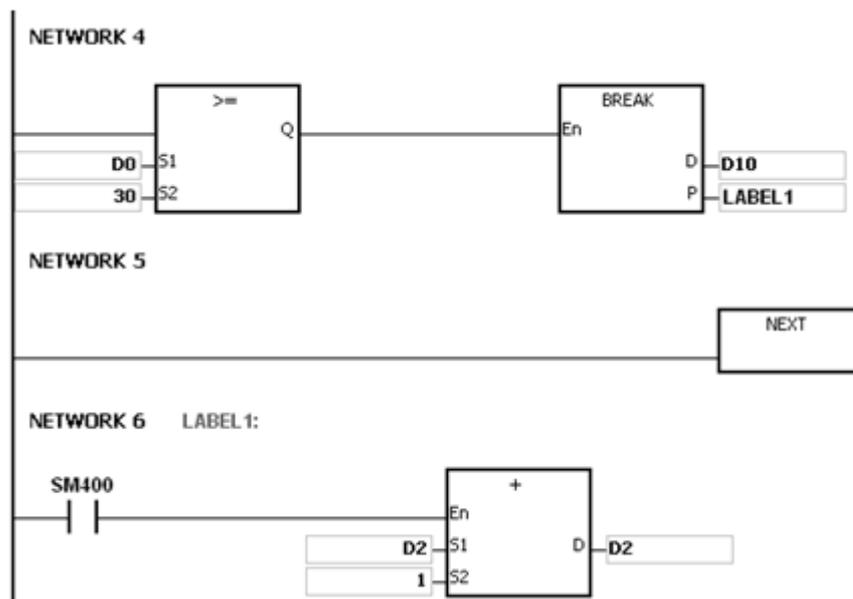
1. 強制結束 FOR-NEXT 的迴圈，將尚未執行完的迴圈數存放到 **D**，並跳至 **P** 所指定的指標開始執行。
2. BREAK 指令執行時，將 FOR-NEXT 剩餘尚未執行的迴圈數存放到 **D** 中，包含 BREAK 指令正在執行時的這一次。

程式範例：

程式開始執行 100 次 FOR-NEXT 迴圈的程式 D0 內容加 1，當 D0 內容等於 30 時，將強制結束 FOR-NEXT 迴圈，將剩餘的迴圈數 71 儲存至 D10，並跳至 LABEL1 (即 NETWORK 6 處) 執行 D2 內容加 1。

6





補充說明：

1. BREAK 指令所指定的 P 指標不存在，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2004。
2. BREAK 指令寫在 FOR-NEXT 之外，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2017。
3. 指標 P 在 ISPSOft 中的使用方式，請參考 ISPSOft 的使用手冊說明。

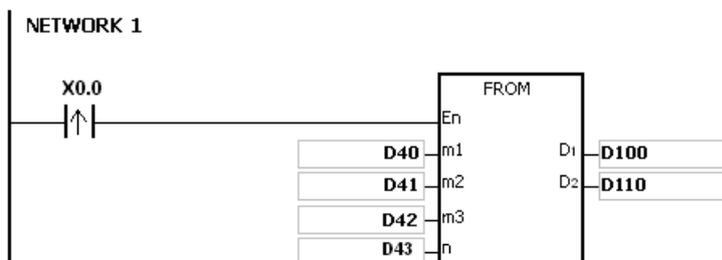
6.15 模組的資料讀寫指令

6.15.1 模組的資料讀寫指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1400</u>	FROM	DFROM	✓	特殊模組 CR 資料讀出	13
<u>1401</u>	TO	DTO	✓	特殊模組 CR 資料寫入	13

程式範例：

當啟動 X0.0=OFF→ON 時，會執行 FROM 的應用指令，讀取置放於 CPU module 右側第一個特殊模組，AH10SCM-A5 的 COM1 資料交換讀取觸發的運作模式 (CR#7)，並且將回覆的 CR#7 儲存到 D100，因為執行無誤所以 D110=16#0000。



各參數使用說明如下：

- 模組位於主背板，因此背板 (Rack) 編號 D40 設定為 16#0001。
- 模組放置在第一個插槽，因此插槽 (Slot) 編號 D41 設定為 16#0000。
- 模組之 COM1 資料交換讀取觸發設定值為 CR#7，因此 CR 編號 D42 設定為 16#0007。
- 模組之 COM1 資料交換讀取觸發設定值只佔用一個暫存器，因此讀取筆數 D43 設定為 1。
- 模組回覆 CR#7 的資料會儲存在 D100 裡。

補充說明：

1. m₁ 與 m₂ 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 當 D1~D1+n-1 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. n 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
4. 由於使用 FROM 會降低 CPU module 與 I/O Module 的執行效能，所以不建議使用。
5. 錯誤代碼說明

錯誤代碼	說明
16#2003	請參考補充說明 1 跟 2
16#200B	請參考補充說明 3
16#1400	輔助處理器存取錯誤
16#1401	I/O 模組存取錯誤
16#1402	I/O 模組不符合 I/O 配置設定
16#1407	輔助處理器通訊錯誤

6

API	指令碼			運算元							功能						
1401	D	TO	P	$m_1 \cdot m_2 \cdot m_3 \cdot S \cdot D \cdot n$							特殊模組 CR 資料寫入						

裝置	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	●	●			●	●	●	●	●				●	○	○		

脈波執行型	16 位元指令 (13 steps)	32 位元指令 (13 steps)
AH500	AH500	AH500

符號：



- m_1 : Rack 代碼 Word/DWord
- m_2 : Slot 代碼 Word/DWord
- m_3 : 欲寫入特殊模組之 CR (Controlled Register) 編號 Word/DWord
- S : 寫入 CR 的資料 Word/DWord
- D : 存放錯誤代碼的位置 Word/DWord
- n : 一次寫入之資料筆數 Word/DWord

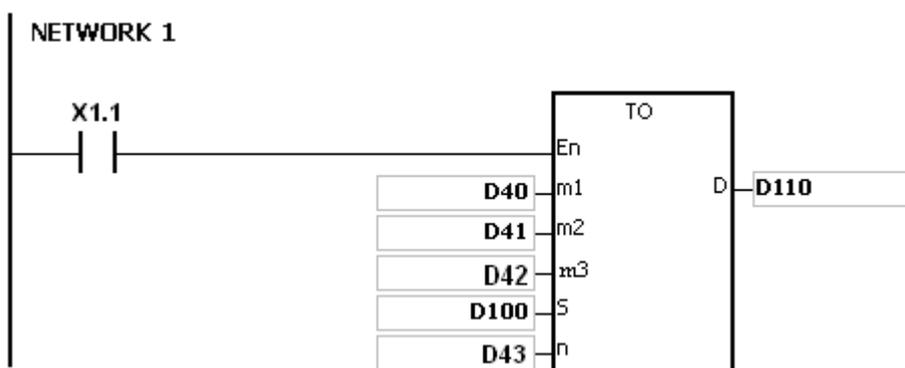
指令說明：

- AH500 系列 PLC 可利用此指令讀取特殊模組之 CR 資料。
- m_1 : 背板 Rack 代碼, $m_1=1\sim 8$; 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 代碼, $m_2=0\sim 11$; 若 $m_1=1$ (主背板), 則 m_2 (插槽) = $0\sim 11$; 若 $m_1=2\sim 8$ (延伸背板), 則 $m_2=0\sim 7$ 。
- m_3 : 欲寫入特殊模組之 CR (Controlled Register) 編號
- D : 開始執行 TO 時, D 會被設定為 0 (表示無錯誤), 當有錯誤時, D 為非 0。有關錯誤代碼說明請參考補充說明。
- n : 讀取之資料筆數
- 16 位元指令中, $n=1\sim 256$; 32 位元指令中, $n=1\sim 128$ 。
- 32 位元指令才可以使用 HC 裝置。
- 特殊模組包含 AIO、NIO 與 PIO。

10. 當 S 為 KH 時，會傳送 n 個 KH 給指定的 I/O module。例如：S 為 16#0001，n 為 3，則傳送三個 16#0001 給 I/O Module。
11. 在 AH 備援系統 (AHCPU560-EN2) 架構下，若使用本指令設定主背板模組參數，則必須考慮當發生控制器切換時，有可能導致備援系統模組參數設定不一致。建議搭配 RCS (API2901) 指令讀取控制器的 PLC ID，藉此判斷是否發生控制器切換，當控制器切換發生時，則重新設定備援系統模組參數，確保備援系統模組參數設定正確。

程式範例：

當啟動 X1.1=OFF→ON 時，會執行 TO 的應用指令，將置放於 CPU module 右側第一個特殊模組，AH-10SCM-A5 的 COM1 資料交換讀取觸發的運作模式 (CR#0007) 從不觸發轉換成觸發一次，因為執行無誤所以 D110=16#0000。



各參數使用說明如下：

- 模組位於主背板，因此背板 (Rack) 編號 D40 設定為 16#0001。
- 模組放置在第一個插槽，因此插槽 (Slot) 編號 D41 設定為 16#0000。
- 模組之 COM1 資料交換讀取觸發設定值為 CR#7，因此 CR 編號 D42 設定為 16#0007。
- 模組之 COM1 資料交換讀取觸發設定值只佔用一個暫存器，因此寫入筆數 D43 設定為 1。
- 要寫入模組 CR#0007 的資料會儲存在 D100 裡，因此 D100 設定為 16#0002。

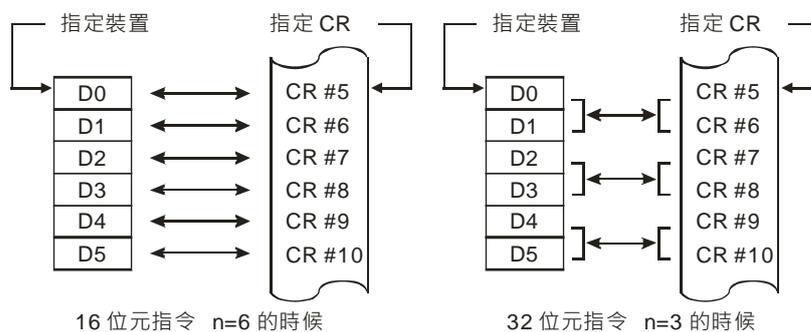
指令運算元的規則說明：

- **m₁**：背板 Rack 代碼，m₁=1~8；其中 m₁=1 代表主背板，m₁=2~8 代表延伸背板。
- **m₂**：插槽 Slot 代碼，m₂=0~11。
若 m₁=1 (主背板)，則 m₂ (插槽) =0~11；若 m₁=2~8 (延伸背板)，則 m₂=0~7。
- **m₃**：CR 的號碼，特殊模組的內部內建 16 位元長度的記憶體，稱之為 CR(Controlled Register)。CR 的編號以 10 進制編碼#0~#N，特殊模組的各種運轉情況及設定值均被包含在裡面。N 的個數依據不同模組而有所不同。
- 最多可掛 68 台特殊模組，且不佔用 I/O 點數。
- 如果使用 FROM/TO 指令時，一次以一個編號的 CR 為讀出/寫入單位，若是使用 DFROM/DTO 指令時，一次以 2 個編號的 CR 為讀出/寫入單位。



6

- 傳送組數 n ，16 位元指令的 $n=2$ 與 32 位元指令的 $n=1$ 意義相同。



補充說明：

1. $m1$ 與 $m2$ 內容值超出範圍時，視為運算錯誤，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. 當 $S\sim S+n-1$ 超出裝置範圍時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
3. n 內容值超出範圍時，視為運算錯誤，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#200B$ 。
4. 由於使用 TO 會降低 CPU module 與 I/O Module 的執行效能，所以不建議使用。
5. 錯誤說明

錯誤代碼	說明
16#2003	請參考補充說明 1 跟 2
16#200B	請參考補充說明 3
16#1400	輔助處理器存取錯誤
16#1401	I/O 模組存取錯誤
16#1402	I/O 模組不符合 I/O 配置設定
16#1407	輔助處理器通訊錯誤

6.16 浮點數指令

6.16.1 浮點數指令一覽表

API	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
<u>1500</u>	–	FSIN	DFSIN	✓	浮點數 SIN 運算	5-6
<u>1501</u>	–	FCOS	DFCOS	✓	浮點數 COS 運算	5-6
<u>1502</u>	–	FTAN	DFTAN	✓	浮點數 TAN 運算	5-6
<u>1503</u>	–	FASIN	DFASIN	✓	浮點數 ASIN 運算	5-6
<u>1504</u>	–	FACOS	DFACOS	✓	浮點數 ACOS 運算	5-6
<u>1505</u>	–	FATAN	DFATAN	✓	浮點數 ATAN 運算	5-6
<u>1506</u>	–	FSINH	DFSINH	✓	浮點數 SINH 運算	5-6
<u>1507</u>	–	FCOSH	DFCOSH	✓	浮點數 COSH 運算	5-6
<u>1508</u>	–	FTANH	DFTANH	✓	浮點數 TANH 運算	5-6
<u>1509</u>	–	FRAD	DFRAD	✓	角度→徑度	5-6
<u>1510</u>	–	FDEG	DFDEG	✓	徑度→角度	5-6
<u>1511</u>	SQR	DSQR	–	✓	BIN 開平方根	5
<u>1512</u>	–	FSQR	DFSQR	✓	浮點數開平方根	5-6
<u>1513</u>	–	FEXP	DFEXP	✓	浮點數取指數	5-6
<u>1514</u>	–	FLOG	DFLOG	✓	浮點數取對數	7-9
<u>1515</u>	–	FLN	DFLN	✓	浮點數取自然對數	5-6
<u>1516</u>	–	FPOW	DFPOW	✓	浮點數權值指令	7-9
<u>1517</u>	RAND	–	–	✓	亂數值	7
<u>1518</u>	BSQR	DBSQR	–	✓	BCD 開平方根	5
<u>1519</u>	–	BSIN	–	✓	BCD SIN 運算	5
<u>1520</u>	–	BCOS	–	✓	BCD COS 運算	5
<u>1521</u>	–	BTAN	–	✓	BCD TAN 運算	5
<u>1522</u>	–	BASIN	–	✓	BCD ASIN 運算	5
<u>1523</u>	–	BACOS	–	✓	BCD ACOS 運算	5
<u>1524</u>	–	BATAN	–	✓	BCD ATAN 運算	5

6.16.2 浮點數指令說明

API	指令碼			運算元							功能						
1500	D	FSIN	P	S · D							浮點數 SIN 運算						

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

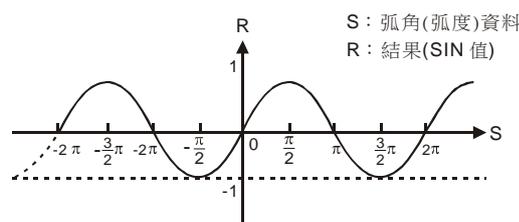
脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：



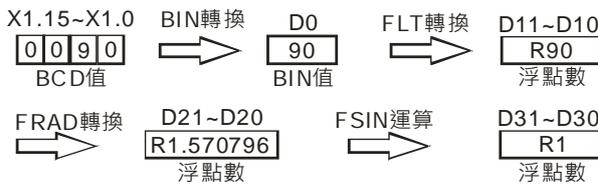
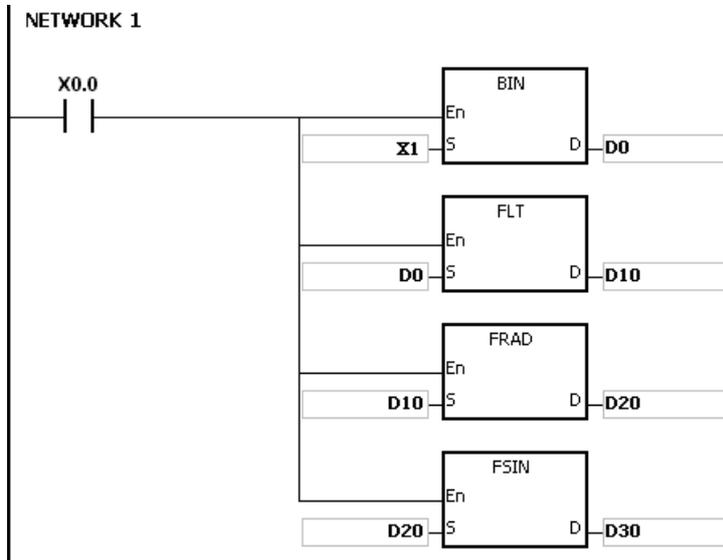
指令說明：

1. S 所指定的來源可為徑度或角度，由旗標 SM695 決定。
2. 當 SM695=OFF 時，指定為徑度模式，徑度值=角度×π/180。
3. 當 SM695=ON 時，指定為角度模式，角度值=徑度×180/π，角度範圍：0°≤角度值≤360°。
4. 若轉換結果為 0，則零旗號 SM600=ON。
5. 將 S 所指定的來源值，求取 SIN 值後存於 D 所指定的暫存器當中。
6. 下圖顯示弧角（弧度、徑度）與結果的關係：



程式範例：

當 X0.0=ON 時，將 (X1.15~X1.0) 的 BCD 值轉成 BIN 值存於 D0，將 D0 做 FLT 運算轉成浮點數值後存於 (D11，D10)，將 (D11，D10) 做 FRAD 運算轉成徑度存於 (D21，D20)，再將 (D21，D20) 的徑度做 SIN 運算後存於 (D31，D30) 當中，內容為浮點數。



6

補充說明：

1. 當 S 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 當 SM695=ON，且 S 的內容值小於 0 或大於 360，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元							功能						
1501	D	FCOS	P	S · D							浮點數 COS 運算						

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

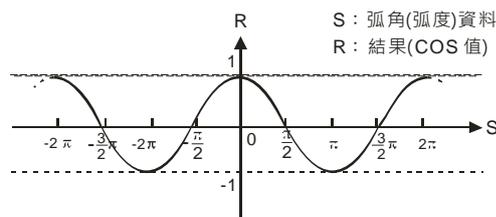
脈波執行型	32 位元指令(5 steps)	64 位元指令(5-6 steps)
AH500	AH500	AH500

符號：



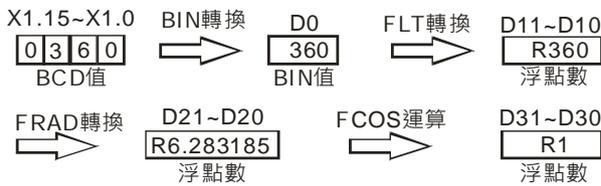
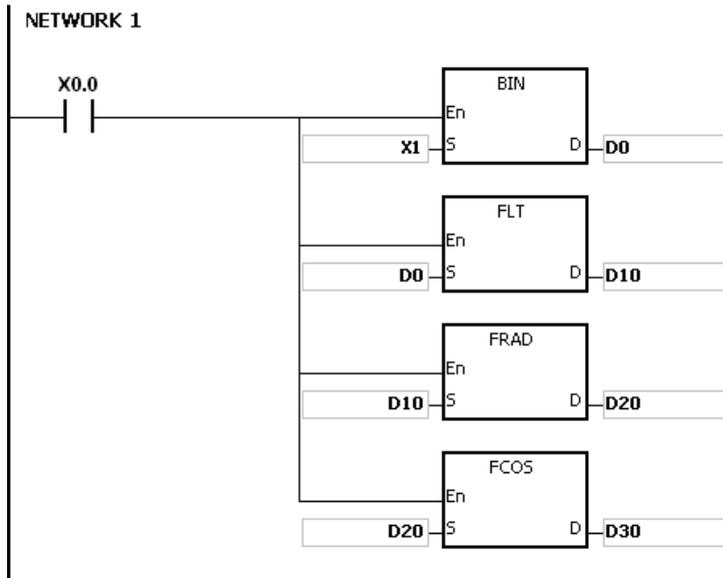
指令說明：

1. S 所指定的來源可為徑度或角度，由旗標 SM695 決定。
2. 當 SM695=OFF 時，指定為徑度模式，徑度值=角度×π/180。
3. 當 SM695=ON 時，指定為角度模式，角度值=徑度×180/π，角度範圍：0°≤角度值≤360°。
4. 若轉換結果為 0，則零旗號 SM600=ON。
5. 將 S 所指定的來源值，求取 COS 值後存於 D 所指定的暫存器當中。
6. 下圖顯示弧角與結果的關係：



程式範例：

當 X0.0=ON 時，將 (X1.15~X1.0) 的 BCD 值轉成 BIN 值存於 D0，將 D0 做 FLT 運算轉成浮點數值後存於 (D11 · D10)，將 (D11 · D10) 做 DRAD 運算轉成徑度存於 (D21 · D20)，再將 (D21 · D20) 的徑度做 COS 運算後存於 (D31 · D30) 當中，內容為浮點數。



補充說明：

1. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 當 SM695=ON，且 **S** 的內容值小於 0 或大於 360，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

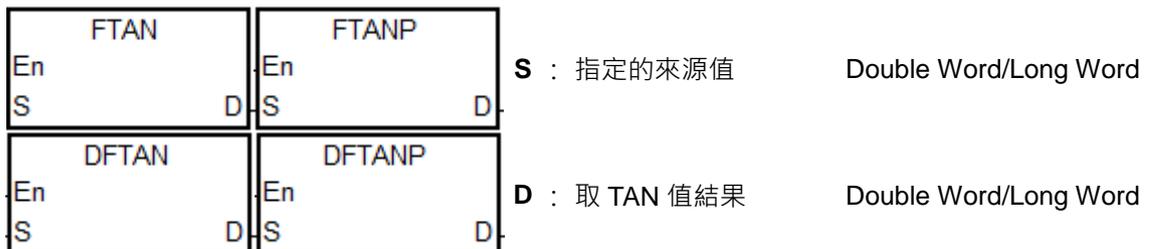
6

API	指令碼			運算元					功能				
	D	FTAN	P	S · D					浮點數 TAN 運算				

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

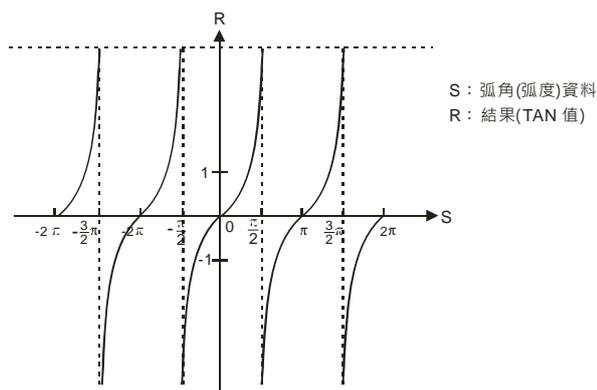
脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：



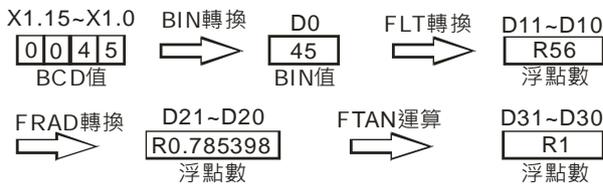
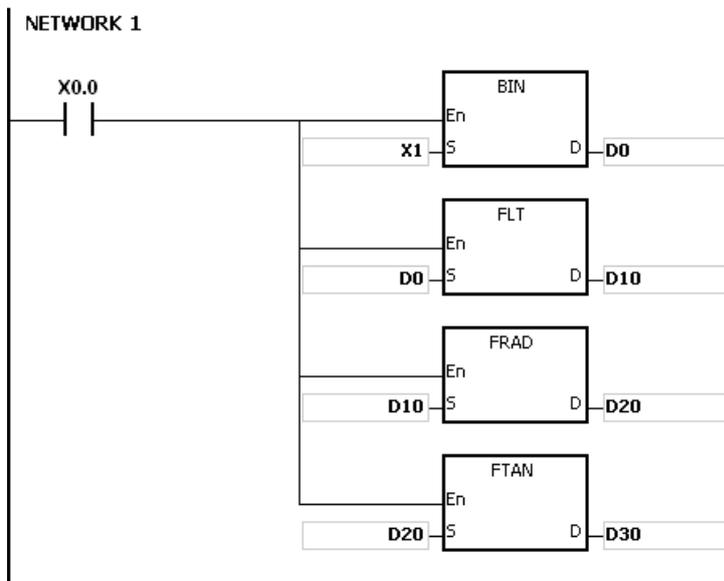
指令說明：

1. S 所指定的來源可為徑度或角度，由旗標 SM695 決定。
2. 當 SM695=OFF 時，指定為徑度模式，徑度值=角度 $\times\pi/180$ 。
3. 當 SM695=ON 時，指定為角度模式，角度值=徑度 $\times 180/\pi$ ，角度範圍： $0^\circ \leq \text{角度值} \leq 360^\circ$ 。
4. 若轉換結果為 0，則零旗號 SM600=ON。
5. 將 S 所指定的來源值，求取 TAN 值後存於 D 所指定的暫存器當中。
6. 下圖顯示弧角與結果的關係：



程式範例：

當 X0.0=ON 時，將 (X1.15~X1.0) 的 BCD 值轉成 BIN 值存於 D0，將 D0 做 FLT 運算轉成浮點數值後存於 (D11 · D10)，將 (D11 · D10) 做 DRAD 運算轉成徑度存於 (D21 · D20)，再將 (D21 · D20) 的徑度做 TAN 運算後存於 (D31 · D30) 當中，內容為浮點數。



補充說明：

1. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 當 SM695=ON，且 **S** 的內容值小於 0 或大於等於 360 或等於 90 或等於 270 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

6

API	指令碼			運算元							功能						
	D	FASIN	P	S · D							浮點數 ASIN 運算						

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

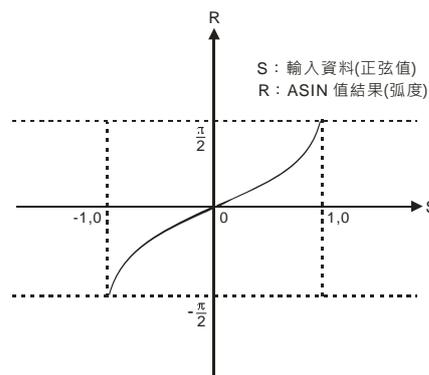
脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：



指令說明：

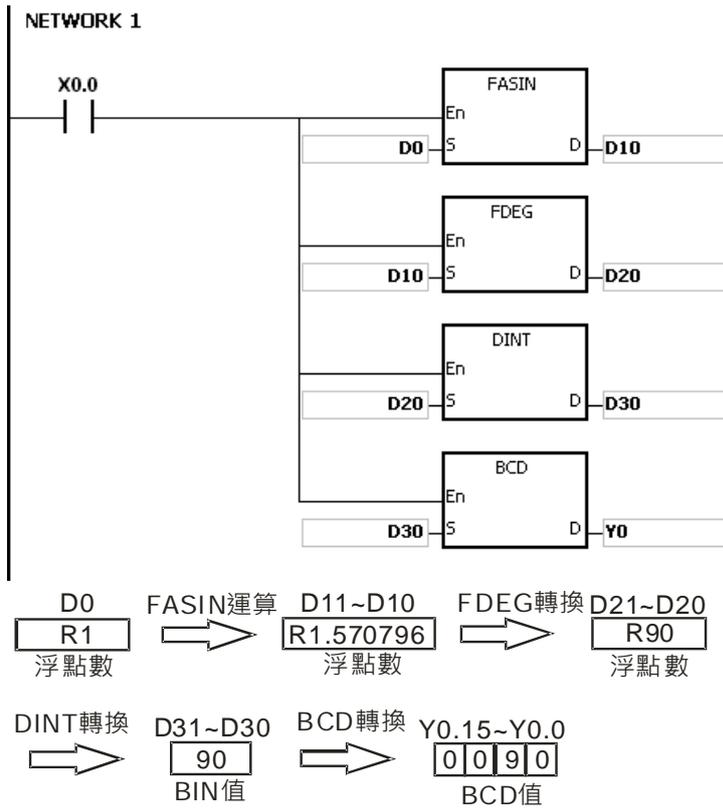
- ASIN 值 = \sin^{-1} ，下圖顯示輸入資料與結果的關係：



- 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

當 X0.0=ON 時，指定浮點數 (D1 · D0) 求取 ASIN 值後存於 (D11 · D10) 當中，將 (D11 · D10) 的結果做 FDEG 轉為角度存於 (D21 · D20)，再將 (D21 · D20) 做 DINT 轉換成整數存於 (D31 · D30)，再透過 BCD 指令將求得角度輸出於 Y0.15~Y0.0。



補充說明：

1. **S** 運算元指定的正弦值數值之十進浮點值只能介於-1.0~+1.0 之間，若不在此範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
3. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

6

API	指令碼			運算元					功能				
	D	FACOS	P	S · D					浮點數 ACOS 運算				

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

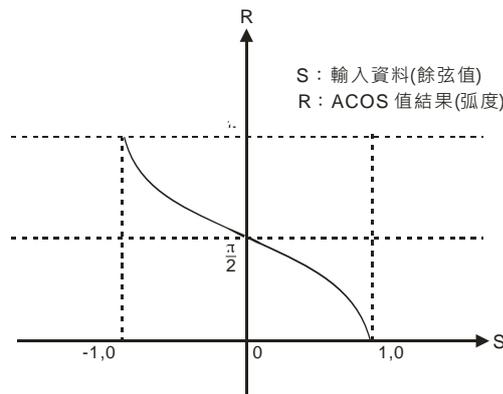
脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：



指令說明：

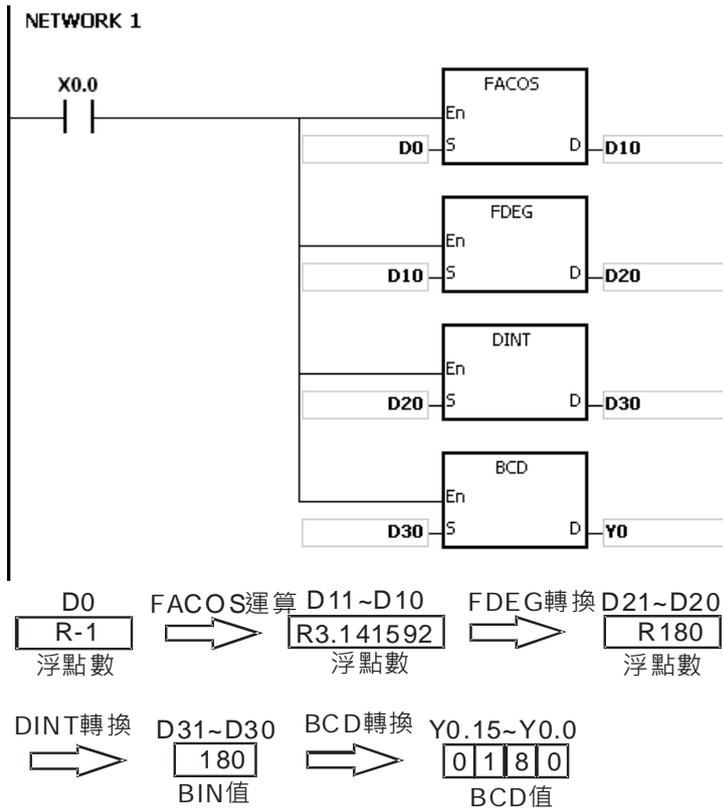
- ACOS 值 = \cos^{-1} · 下圖顯示輸入資料與結果的關係：



- 若轉換結果的絕對值大於可表示之最大浮點值，則進位旗號 SM602=ON。
- 若轉換結果的絕對值小於可表示之最小浮點值，則借位旗號 SM601=ON。
- 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

當 X0.0=ON 時，指定浮點數 (D1 · D0) 求取 ACOS 值後存於 (D11 · D10) 當中，將 (D11 · D10) 的結果做 FDEG 轉為角度存於 (D21 · D20)，再將 (D21 · D20) 做 DINT 轉換成整數存於 (D31 · D30)，再透過 BCD 指令將求得角度輸出於 Y0.15~Y0.0。



補充說明：

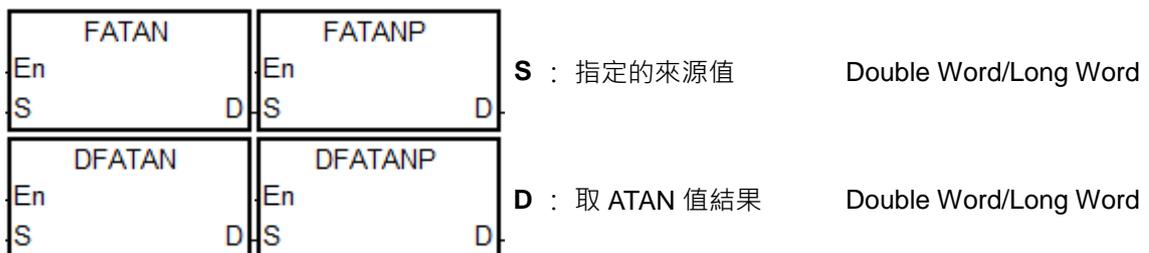
1. **S** 運算元指定的正弦值數值之十進浮點值只能介於-1.0~+1.0 之間，若不在此範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
3. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

6

API	指令碼			運算元								功能					
1505	D	FATAN	P	S · D								浮點數 ATAN 運算					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●		●				○
D	●	●			●	●	●	●	●		●		●				

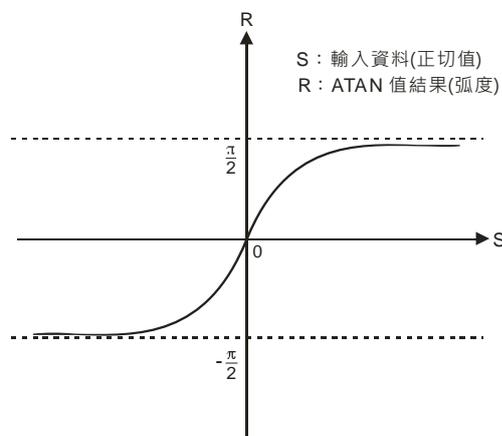
脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：



指令說明：

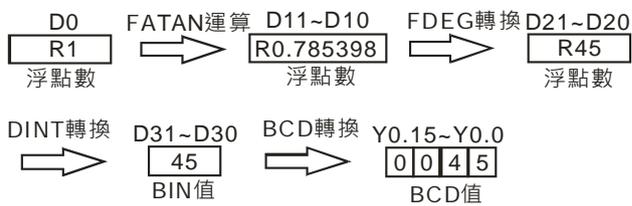
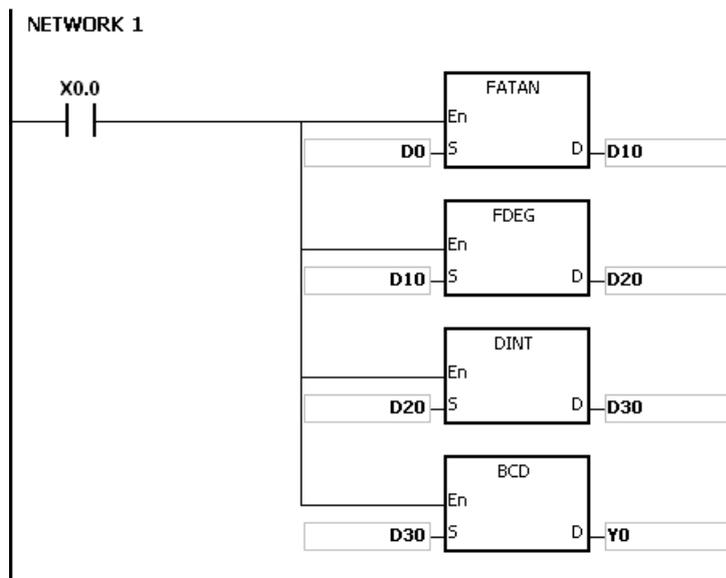
1. ATAN 值 = \tan^{-1}
2. 下圖顯示輸入資料與結果的關係：



3. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

當 X0.0=ON 時，指定浮點數 (D1 · D0) 求取 ATAN 值後存於 (D11 · D10) 當中，將 (D11 · D10) 的結果做 FDEG 轉為角度存於 (D21 · D20)，再將 (D21 · D20) 做 DINT 轉換成整數存於 (D31 · D30)，再透過 BCD 指令將求得角度輸出於 Y0.15~Y0.0。



補充說明：

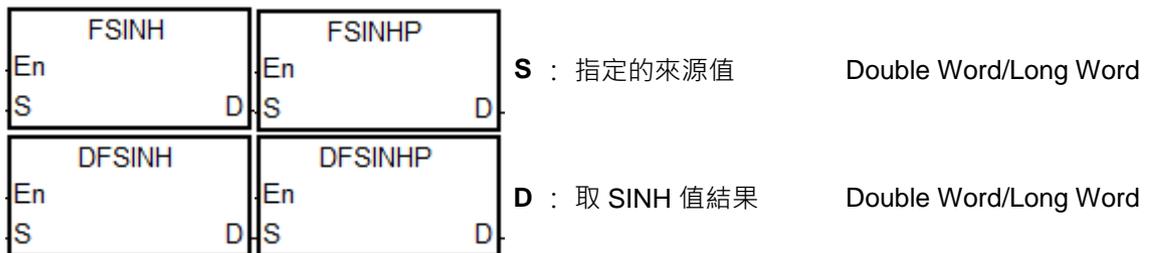
1. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

6

API	指令碼			運算元								功能						
1506	D	FSINH	P	S · D								浮點數 SINH 運算						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●		●					○
D	●	●			●	●	●	●	●		●		●					

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：

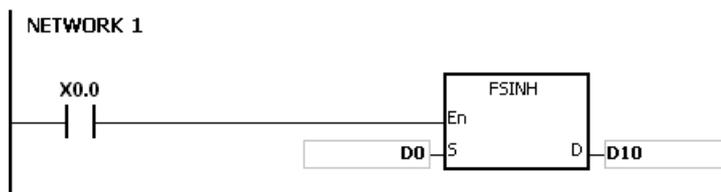


指令說明：

- SINH 值 = $(e^s - e^{-s}) / 2$ 。
- 32 位元單精度浮點數指令：
 - 若轉換結果的絕對值小於可表示之最小浮點值，則 D=16#FF800000，借位旗號 SM601=ON。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則 D=16#7F800000，進位旗號 SM602=ON。
- 64 位元雙精度浮點數指令：
 - 若轉換結果的絕對值小於可表示之最小浮點值，則 D=16#FFF0000000000000，借位旗號 SM601=ON。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則 D=16#7FF0000000000000，進位旗號 SM602=ON。
- 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

- 當 X0.0=ON 時，指定浮點數 (D11 · D10) 求取 SINH 值後存於 (D11 · D10) 當中，內容為浮點數。



S

D1	D0
----	----

 單精度浮點數



D

D11	D10
-----	-----

 FSINH值單精度浮點數

2. 若轉換結果的絕對值大於可表示之最大浮點值，則進位旗號 SM602=ON。
3. 若轉換結果的絕對值小於可表示之最小浮點值，則借位旗號 SM601=ON。
4. 若轉換結果為 0，則零旗號 SM600=ON。

補充說明：

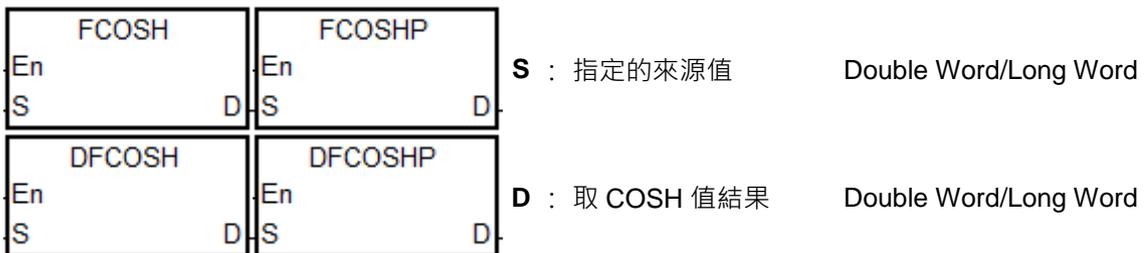
1. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元					功能				
1507	D	FCOSH	P	S · D					浮點數 COSH 運算				

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

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：

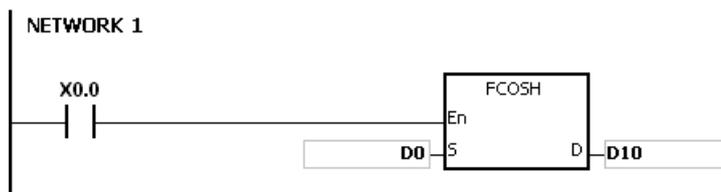


指令說明：

- COSH 值 = $(e^s + e^{-s}) / 2$ 。
- 32 位元單精度浮點數指令：
 - 若轉換結果的絕對值小於可表示之最小浮點值，則 D=16#FF800000，借位旗號 SM601=ON。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則 D=16#7F800000，進位旗號 SM602=ON。
- 64 位元雙精度浮點數指令：
 - 若轉換結果的絕對值小於可表示之最小浮點值，則 D=16#FFF0000000000000，借位旗號 SM601=ON。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則 D=16#7FF0000000000000，進位旗號 SM602=ON。
- 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

- 當 X0.0=ON 時，指定浮點數 (D11 · D10) 求取 COSH 值後存於 (D11 · D10) 當中，內容為浮點數。





2. 若轉換結果的絕對值大於可表示之最大浮點值，則進位旗號 **SM602=ON**。
3. 若轉換結果的絕對值小於可表示之最小浮點值，則借位旗號 **SM601=ON**。
4. 若轉換結果為 0，則零旗號 **SM600=ON**。

補充說明：

1. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2013**。
2. 64 位元雙精度浮點數指令僅支援於 **AHCPU5X1-EN V2.01** 與 **AHCPU5X1-RS2 V1.03** 以後之版本。

API	指令碼			運算元								功能						
1508	D	FTANH	P	S · D								浮點數 TANH 運算						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF	
S	●	●			●	●	●	●	●		●		●					○
D	●	●			●	●	●	●	●		●		●					

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：

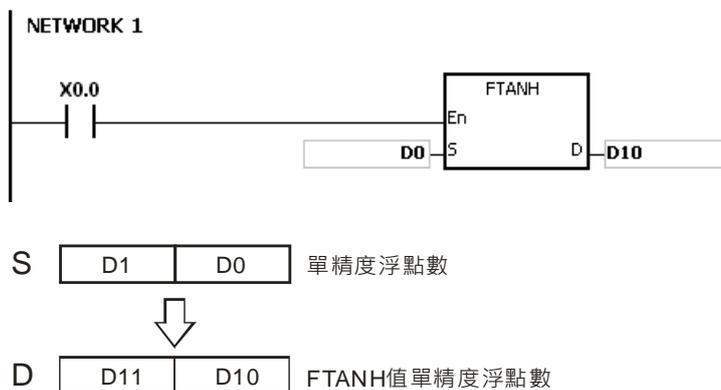


指令說明：

1. $TANH 值 = (e^s - e^{-s}) / (e^s + e^{-s})$ 。
2. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

1. 當 X0.0=ON 時，指定浮點數 (D1 · D0) 求取 TANH 值後存於 (D11 · D10) 當中，內容為浮點數。



2. 若轉換結果為 0，則零旗號 SM600=ON。

補充說明：

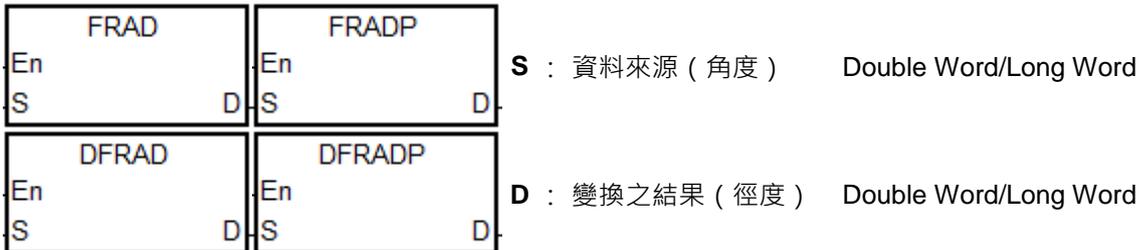
1. 當 S 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元						功能						
	1509	D	FRAD	P	S · D						角度→徑度					

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

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：

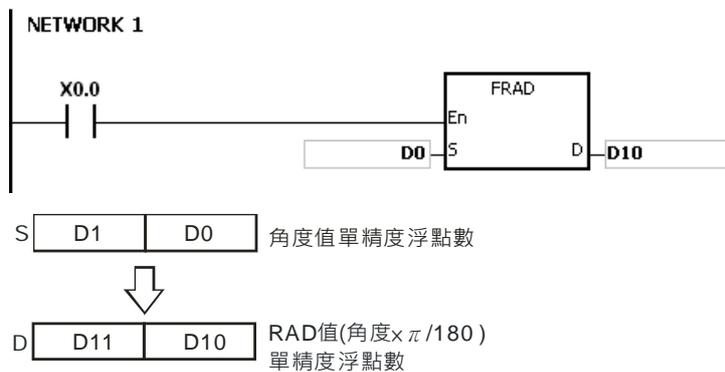


指令說明：

1. 使用下列公式將角度轉換成徑度。
2. 徑度 = 角度 × (π/180)
3. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

當 X0.0=ON 時，指定浮點數 (D1 · D0) 之角度值，將角度轉換成徑度值後存於 (D11 · D10) 當中，內容為浮點數。



補充說明：

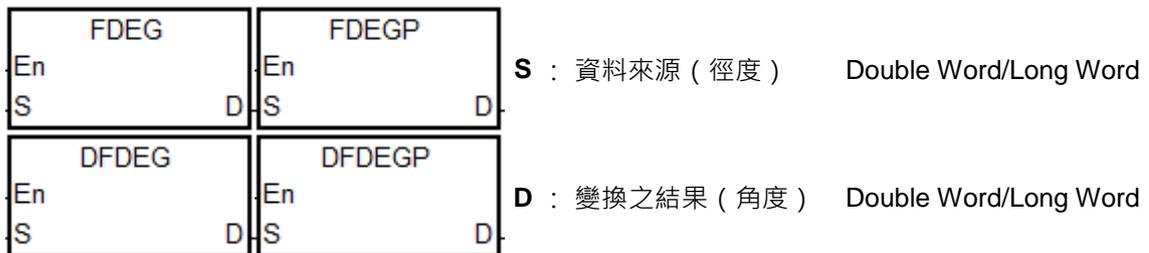
1. 當 S 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元					功能				
1510	D	FDEG	P	S · D					徑度→角度				

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

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：

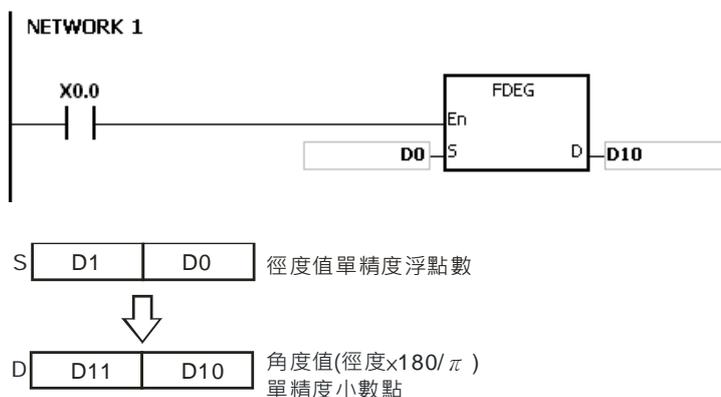


指令說明：

- 使用下列公式將徑度轉換成角度。
- 角度 = 徑度 × (180/π)
- 32 位元單精度浮點數指令：
 - 若轉換結果的絕對值小於可表示之最小浮點值，則 D=16#FF7FFFFFFF。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則 D=16#7F7FFFFFFF。
- 64 位元雙精度浮點數指令：
 - 若轉換結果的絕對值小於可表示之最小浮點值，則 D=16#FFEFFFFFFFFFFFFFFF。
 - 若轉換結果的絕對值大於可表示之最大浮點值，則 D=16#7FEFFFFFFFFFFFFFFF。
- 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

當 X0.0=ON 時，指定浮點數 (D1 · D0) 之角度值，將徑度值轉換成角度後存於 (D11 · D10) 當中，內容為浮點數。



補充說明：

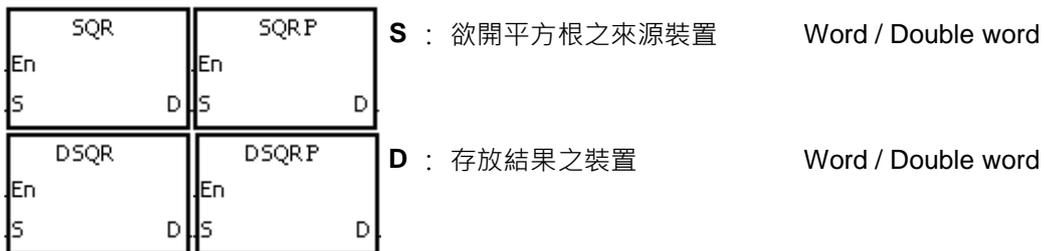
1. 當 **S** 的內容值超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元								功能				
1511	D	SQR	P	S · D								開平方根				

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：

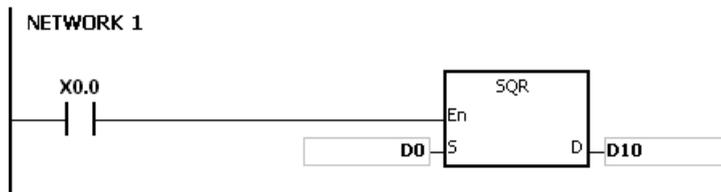


指令說明：

1. 將 **S** 所指定之裝置內容值開平方根後，存放於 **D** 所指定之裝置。
2. 運算結果 **D** 只求整數，小數點被捨棄。有小數點被捨棄時，借位旗標信號 SM601=ON。
3. 運算結果 **D** 為 0 時，零旗標信號 SM600=ON。

程式範例：

當 X0.0=ON 時，將 D0 內容值開平方根後，存放於 D10 內。



補充說明：

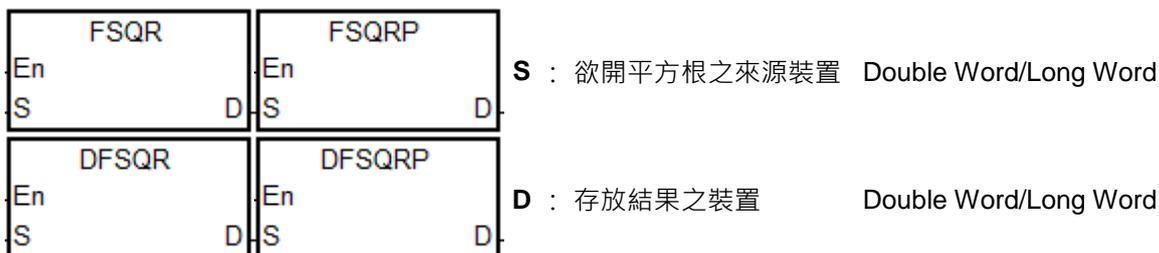
S 只可以指定正數，若指定負數時，PLC 視為“指令運算錯誤”，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元						功能					
1512	D	FSQR	P	S · D						浮點數開平方根					

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

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：



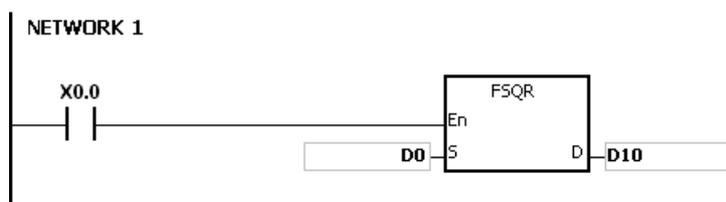
指令說明：

1. **S** 所指定的暫存器內容被開平方，所得的結果暫存於 **D** 所指定的暫存器內容，開平方的動作全部以浮點數型態進行。
2. 若運算結果為 0，則零旗號 **SM600=ON**。

程式範例一：

6

當 **X0.0=ON** 時，將浮點數 (**D1 · D0**) 取開平方根，將結果存放至 (**D11 · D10**) 所指定的暫存器當中。

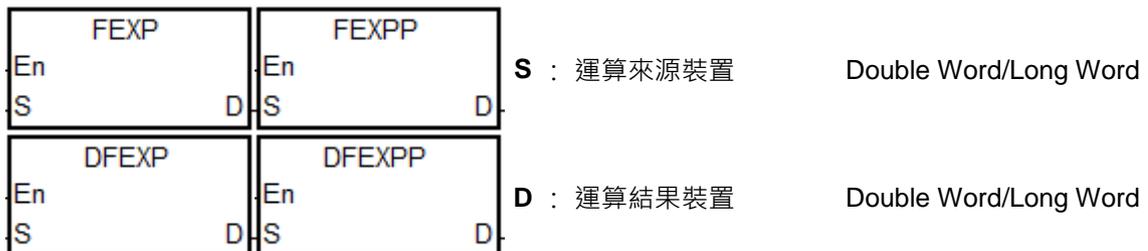


補充說明：

1. **S** 只可以指定正數，若指定負數時，PLC 視為“指令運算錯誤”，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2003**。
2. 64 位元雙精度浮點數指令僅支援於 **AHCPU5X1-EN V2.01** 與 **AHCPU5X1-RS2 V1.03** 以後之版本。

API	指令碼			運算元								功能						
1513	D	FEXP	P	S · D								浮點數取指數						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●		●				○	
D	●	●			●	●	●	●	●		●		●					
										脈波執行型			32 位元指令 (5 steps)			64 位元指令 (5-6 steps)		
										AH500			AH500			AH500		

符號：

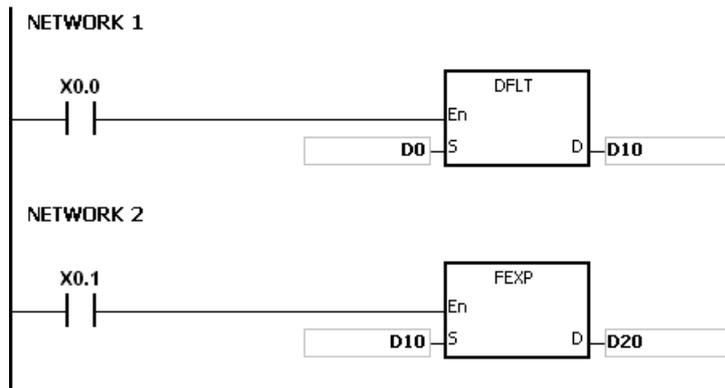


指令說明：

- 以 $e=2.71828$ 為底數，S 為指數做 EXP 運算。
- $EXP[D+1 \cdot D]=[S+1 \cdot S]$
- S 內容正負數都有效，運算時均以浮點數方式執行，故 S 需轉換為浮點數值。
- D 運算元內容值 $=e^S$ ； $e=2.71828$ ，S 為指定的來源資料。
- 32 位元單精度浮點數指令：
 - 若運算結果的絕對值大於可表示之最大浮點值，則 $D=16\#7F800000$ ，進位旗號 $SM602=ON$ 。
- 64 位元雙精度浮點數指令：
 - 若運算結果的絕對值大於可表示之最大浮點值，則 $D=16\#7FF0000000000000$ ，進位旗號 $SM602=ON$ 。
- 若運算結果為 0，則零旗號 $SM600=ON$ 。

程式範例：

- 當 X0.0 為 ON 時，將 (D1 · D0) 值轉成浮點數存於 (D11 · D10) 暫存器中。
- 當 X0.1 為 ON 時，(D11 · D10) 為指數做 EXP 運算，其值為浮點數值並存放於 (D21 · D20) 暫存器中。



補充說明：

64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

6

API	指令碼			運算元							功能				
1514	D	FLOG	P	$S_1 \cdot S_2 \cdot D$							浮點數取對數				

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

脈波執行型	32 位元指令 (7 steps)	64 位元指令 (7-9 steps)
AH500	AH500	AH500

符號：

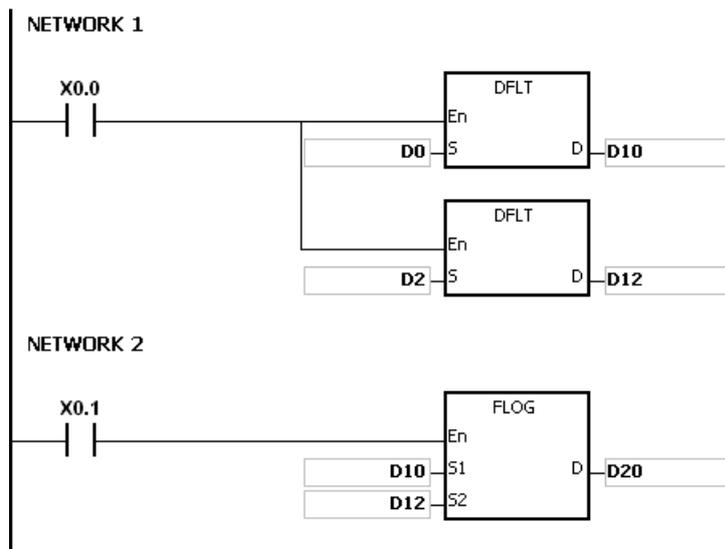


指令說明：

- 將 S_1 內容及 S_2 內容為運算元做 log 運算，結果存放於 D。
- S_1 、 S_2 內容值只有正數有效，運算時均以浮點數方式執行，故 S_1 、 S_2 需轉換為浮點數值。
- $S_1^D = S_2$ ，求 D 值 $\rightarrow \text{Log}_{S_1} S_2 = D$
- 例：已知 $S_1 = 5$ ， $S_2 = 125$ ，求 $D = \log_5 125 = ?$
- $S_1^D = S_2 \rightarrow 5^D = 125 \rightarrow D = \log_5 125 = 3$
- 若運算結果為 0，則零旗號 SM600=ON。

程式範例：

- 當 X0.0 為 ON 時，將 (D1 · D0) 內容及 (D3 · D2) 內容轉成浮點數分別存於 (D11 · D10) 及 (D13 · D12) 32 位元暫存器中。
- 當 X0.1 為 ON 時，將 (D11 · D10) 及 (D13 · D12) 32 位元暫存器浮點數值做 Log 運算並將結果存於 (D21 · D20) 32 位元暫存器中。



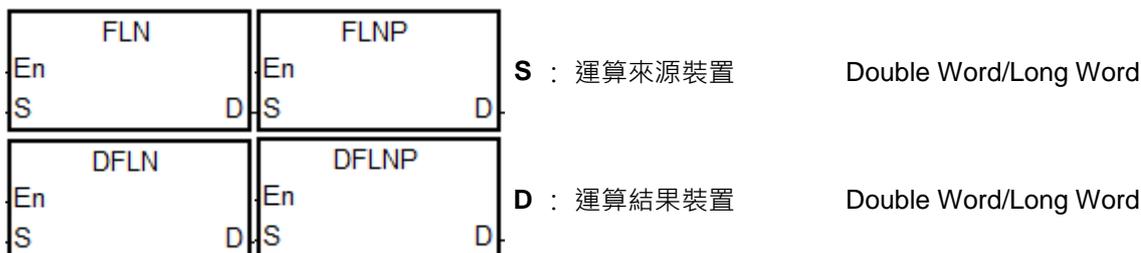
補充說明：

1. S_1 內容值 ≤ 1 或 $S_2 \leq 0$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元								功能						
	D	FLN	P	S · D								浮點數取自然對數						
1515																		
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●	●	●	●		●		●					○
D	●	●			●	●	●	●	●		●		●					

脈波執行型	32 位元指令 (5 steps)	64 位元指令 (5-6 steps)
AH500	AH500	AH500

符號：

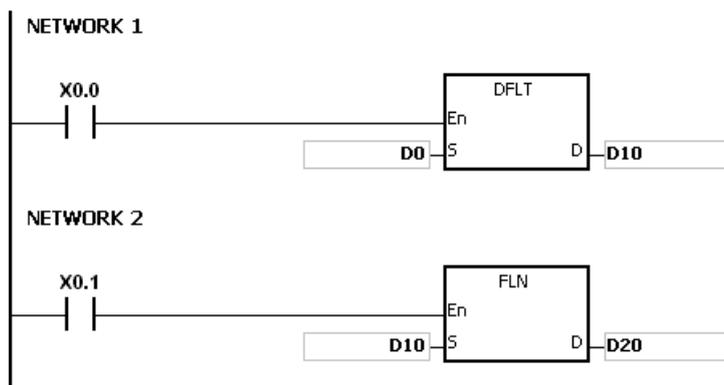


指令說明：

1. 以 **S** 為運算元做自然對數 \ln 運算。
2. $\ln[S+1 \cdot S]=[D+1 \cdot D]$
3. **S** 內容只有正數有效，運算時均以浮點數方式執行，故 **S** 需轉換為浮點數值。
4. $e D=S \rightarrow D$ 運算元內容值 $=\ln S$; **S** 為指定的來源資料。
5. 若運算結果為 0，則零旗號 $SM600=ON$ 。

程式範例：

1. 當 X0.0 為 ON 時，將 (D1 · D0) 值轉成浮點數存於 (D11 · D10) 暫存器中。
2. 當 X0.1 為 ON 時，將 (D11 · D10) 暫存器為真數做 \ln 運算，其值為浮點數並存放於 (D21 · D20) 暫存器中。



補充說明：

1. $S \leq 0$ 時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元						功能						
1516	D	FPOW	P	$S_1 \cdot S_2 \cdot D$						浮點數權值指令						

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

脈波執行型	32 位元指令 (7 steps)	64 位元指令 (7-9 steps)
AH500	AH500	AH500

符號：



指令說明：

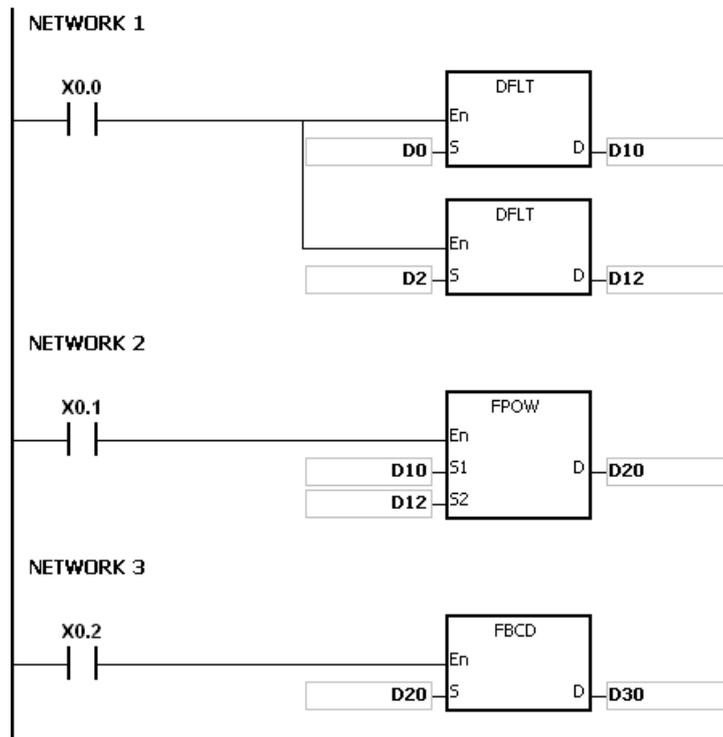
- 將浮點數資料 S_1 以 S_2 的次冪數相乘後存放於 D。
- $D = \text{POW}[S_1+1 \cdot S_1]^{[S_2+1 \cdot S_2]}$
- S_1 內容值只有正數有效， S_2 內容值正負值都有效。運算時均以浮點數方式執行，故 $S_1 \cdot S_2$ 需轉換為浮點數值。
- 例： $S_1^{S_2} = D$ ，求 D 值？
已知 $S_1=5 \cdot S_2=3$ ，則 $D=5^3=125$
- 32 位元單精度浮點數指令：
 - 若運算結果的絕對值小於可表示之最小浮點值，則 $D=16\#FF7FFFFF$ 。
 - 若運算結果的絕對值大於可表示之最大浮點值，則 $D=16\#7F7FFFFF$ 。
- 64 位元雙精度浮點數指令：
 - 若運算結果的絕對值小於可表示之最小浮點值，則 $D=16\#FFEFFFFFFFFFFFFFFF$ 。
 - 若運算結果的絕對值大於可表示之最大浮點值，則 $D=16\#7FEFFFFFFFFFFFFFFF$ 。
- 若運算結果為 0，則零旗號 $SM600=ON$ 。

程式範例：

- 當 X0.0 為 ON 時，將 (D1 · D0) 內容及 (D3 · D2) 內容轉成浮點數分別存於 (D11 · D10) 及 (D13 · D12) 32 位元暫存器中。
- 當 X0.1 為 ON 時，將 (D11 · D10) 及 (D13 · D12) 32 位元暫存器浮點數做 POW 運算並將結果存於 (D21 · D20) 32 位元暫存器中。
- 當 X0.2 為 ON 時，將 (D21 · D20) 32 位元暫存器浮點數值轉成十進浮點數值並存於 (D31 ·

6

D30) 暫存器中。(此時 D31 為表示 D30 的 10 次冪方)



補充說明：

1. $S_1 < 0$ 時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 64 位元雙精度浮點數指令僅支援於 AHCPU5X1-EN V2.01 與 AHCPU5X1-RS2 V1.03 以後之版本。

API	指令碼			運算元				功能			
1517		RAND	P	$S_1 \cdot S_2 \cdot D$				亂數值			

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">RAND</td> <td style="width: 20px;"></td> <td style="text-align: center;">RANDP</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></td> <td></td> <td>D</td> </tr> </table>	RAND		RANDP	En		En	S1	D	S1	S2		S2			D	<p>S_1 : 亂數產生的範圍下限 Word</p> <p>S_2 : 亂數產生的範圍上限 Word</p> <p>D : 亂數產生的結果 Word</p>
RAND		RANDP														
En		En														
S1	D	S1														
S2		S2														
		D														

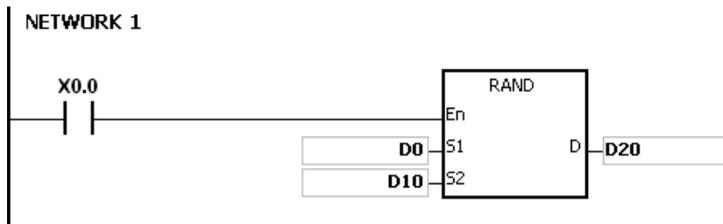
指令說明：

- 產生介於範圍下限 S_1 與範圍上限 S_2 之亂數，將結果存放到 D 內。
- 若使用者輸入 $S_1 > S_2$ ，指令執行時把 S_1 作為上限值， S_2 下限值進行比較。

程式範例：

當 X0.0=ON 時，RAND 指令產生介於範圍下限 D0 與範圍上限 D10 之亂數，將結果存放到 D20 內。

6



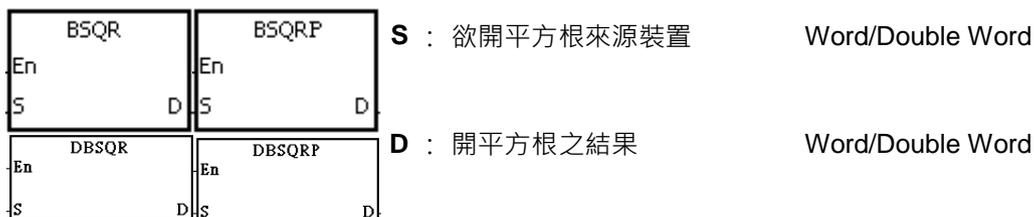
補充說明：

S_1 或 S_2 運算元內容值範圍為 0~32767，內容值超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能					
	D	BSQR	P	S · D								BCD 開平方根					
1518																	
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●	●	●	●		●	○	●	○	○		
D	●	●			●	●	●	●	●		●	○	●				

脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：

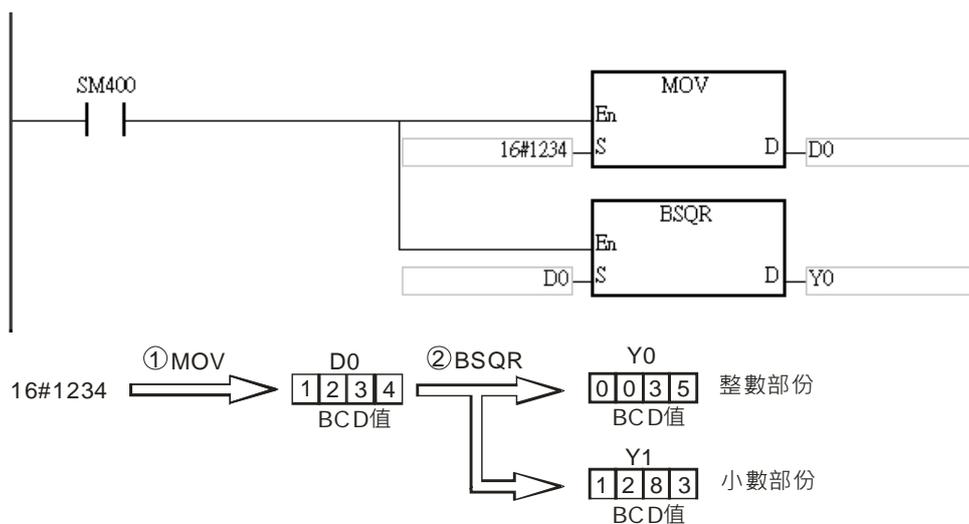


指令說明：

1. 將 **S** 所指定之裝置內容值開平方根後，將結果的整數部份存放於 **D**，小數部份存於 **D+1** 所指定之裝置。
2. 資料來源 **S** 的內容有效數值範圍：BCD (0~9,999) · DBCD (0~99,999,999)。
3. BSQR 小數部份計算到小數點後第 4 位。
4. DBSQR 小數部份計算到小數點後第 8 位。
5. 運算結果 D 為 0 時，零旗標信號 SM600=ON。

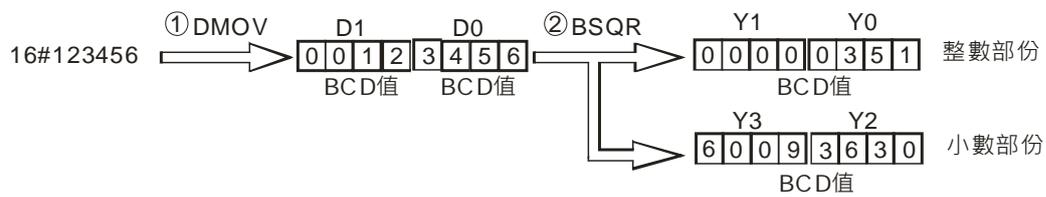
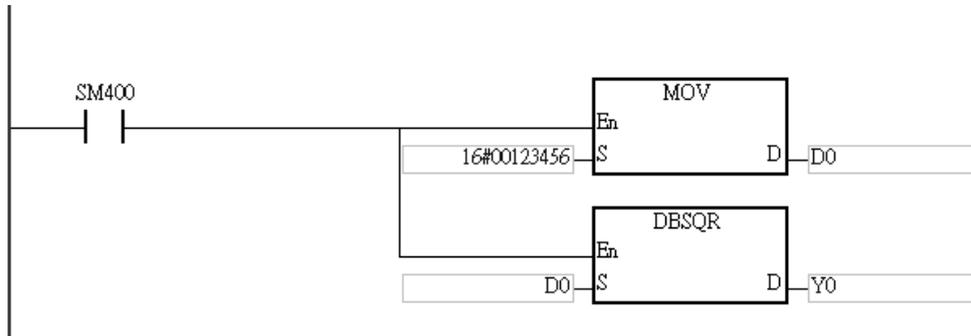
程式範例一：

D0 的內容經開根號後，將結果整數部分存放至 Y0，小數點部分存放至 Y1。



程式範例二：

D0 的內容經開根號後，將結果整數部分存放至 Y0，小數點部分存放至 Y1。



補充說明：

1. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
2. 16 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
3. 32 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of DWORD/DINT。

6

API	指令碼			運算元								功能					
1519		BSIN	P	S · D								BCD SIN 運算					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：

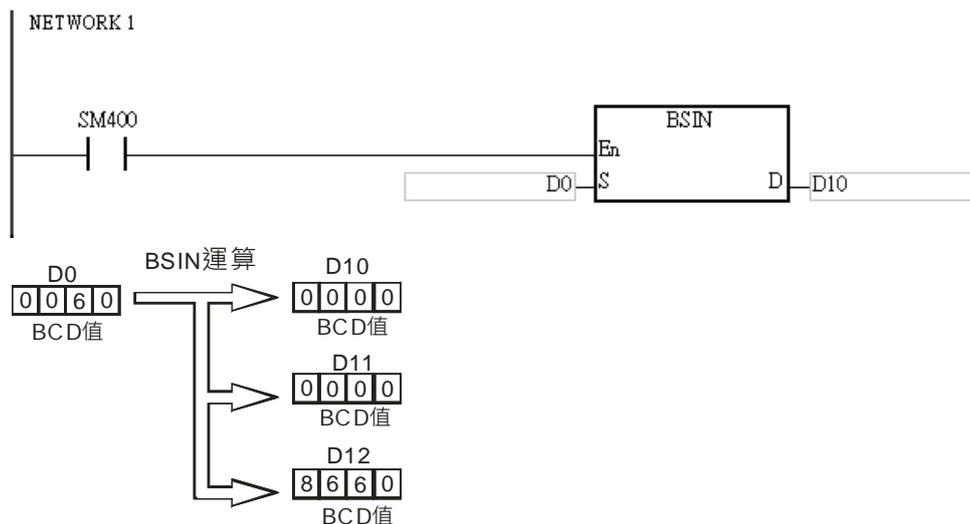


指令說明：

1. S 所指定的來源為角度，求取 SIN 值，運算結果的符號存放於 D，整數部份存於 D+1，小數部份存於 D+2。
2. 角度範圍：0° ≤ 角度值 < 360°。
3. 運算結果計算到小數點後第 5 位四捨五入。
4. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

D0 的內容經過 SIN 運算後將結果儲存至 D 運算元中，符號部份儲存於 D10；整數部份儲存於 D11；小數部份儲存於 D12。



補充說明：

1. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
2. 當 S 的資料內容不在 0° 到 360°，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#2003。
3. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元						功能							
1520		BCOS	P	S · D						BCD COS 運算							
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●		●	○	●				
脈波執行型										16 位元指令 (5 steps)				32 位元指令			
AH500										AH500				-			

符號：

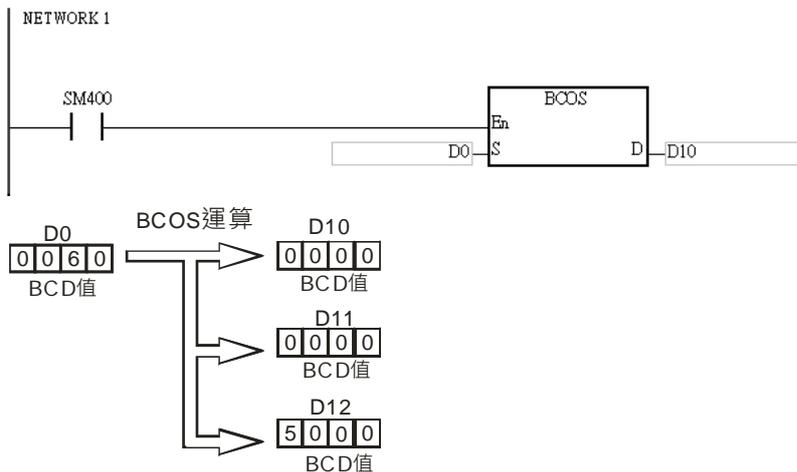


指令說明：

1. S 所指定的來源為角度，求取 COS 值，運算結果的符號存放於 D，整數部份存於 D+1，小數部份存於 D+2。
2. 角度範圍：0° ≤ 角度值 < 360°。
3. 運算結果計算到小數點後第 5 位四捨五入。
4. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

D0 的內容經過 COS 運算後將結果儲存至 D 運算元中，符號部份儲存於 D10；整數部份儲存於 D11；小數部份儲存於 D12。



補充說明：

1. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
2. 當 S 的資料內容不在 0° 到 360°，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#2003。
3. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元								功能						
1521		BTAN	P	S · D								BCD TAN 運算						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●		●	●		●	○	●	○	○			
D	●	●			●	●		●	●		●	○	●					
										脈波執行型			16 位元指令 (5 steps)			32 位元指令		
										AH500			AH500			-		

符號：

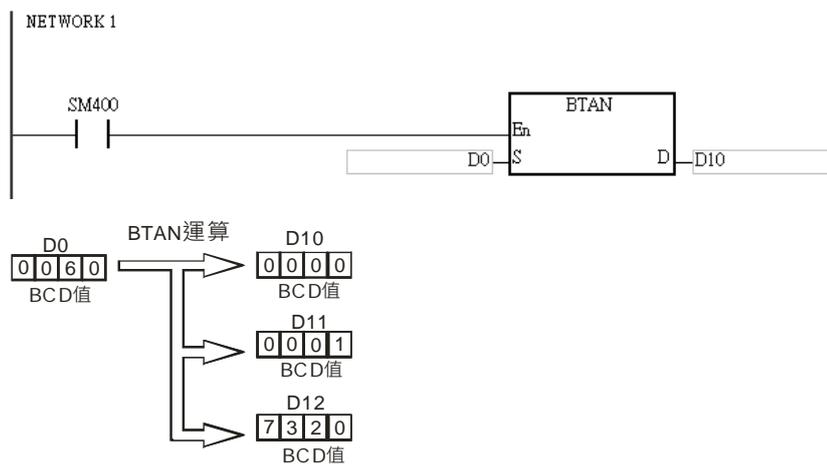


指令說明：

1. S 所指定的來源為角度，求取 TAN 值，運算結果的符號存放於 D，整數部份存於 D+1，小數部份存於 D+2。
2. 角度範圍：0° ≤ 角度值 < 360°。
3. 運算結果計算到小數點後第 5 位四捨五入。
4. 若轉換結果為 0，則零旗號 SM600=ON。

程式範例：

D0 的內容經過 TAN 運算後將結果儲存至 D 運算元中，符號部份儲存於 D10；整數部份儲存於 D11；小數部份儲存於 D12。



補充說明：

1. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
2. 當 S 的資料內容不在 0° 到 360°，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#2003。
3. 當 S 的資料內容=90° 或 270°，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#2003。
4. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元							功能						
1522		BASIN	P	S · D							BCD ASIN 運算						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				
脈波執行型										16 位元指令 (5 steps)				32 位元指令			
AH500										AH500				-			

符號：



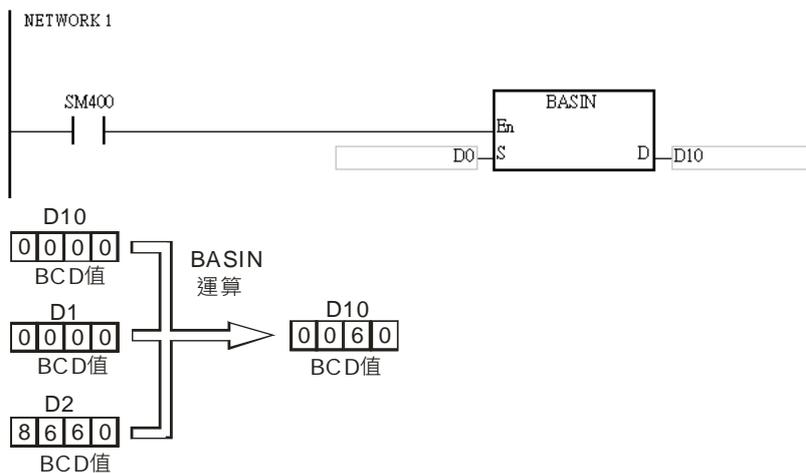
指令說明：

1. S 所指定的來源 (BCD)，求取 ASIN 值 = \sin^{-1} 值，運算的結果存放於 D (角度)。
2. S 的內容值代表符負，0=正值，1=負值，S+1 代表整數部份，S+2 代表小數部份。
3. 運算結果將小數四捨五入。
4. 運算結果為 0°到 90°和 270°到 360°之間的 BCD 值 (角度)。

程式範例：

D0 內容為 \sin^{-1} 之正負號，D1 內容為 \sin^{-1} 之整數部分，D2 值為 \sin^{-1} 小數部份。經過 BASIN 指令運算後，其結果四捨五入至整數儲存於 D10 內。

6



補充說明：

1. 小數部份輸入注意事項：以 0.5 為例，輸入時，需輸入 S=0，S+1=0，S+2=16#5000。
2. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
3. S 運算元指定的正弦值數值只能介 -1.0~+1.0 之間，若不在此範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元								功能						
1523	BACOS	P		S · D								BCD ACOS 運算						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
S	●	●			●	●		●	●		●	○	●					
D	●	●			●	●		●	●		●	○	●					
										脈波執行型			16 位元指令 (5 steps)			32 位元指令		
										AH500			AH500			-		

符號：

BACOS		BACOSP		S	：	指定的來源值	Word
En		En		D	：	取 ACOS 值結果	Word
S		S		D			

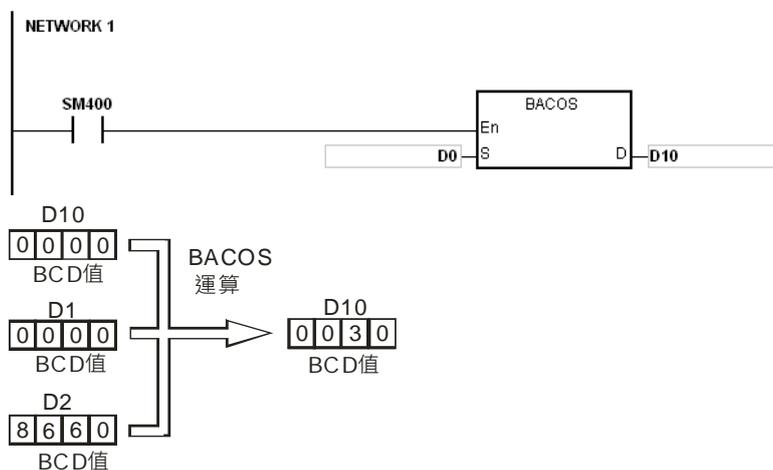
指令說明：

1. S 所指定的來源 (BCD) · 求取 ACOS 值 = \cos^{-1} 值 · 運算的結果存放於 D (角度) 。
2. S 的內容值代表符負 · 0=正值 · 1=負值 · S+1 代表整數部份 · S+2 代表小數部份 。
3. 運算結果將小數四捨五入 。
4. 運算結果為 0° 到 180° 之間的 BCD 值 (角度) 。

程式範例：

D0 內容為 \cos^{-1} 之正負號 · D1 內容為 \cos^{-1} 之整數部分 · D2 值為 \cos^{-1} 小數部份 。

經過 BACOS 指令運算後 · 其結果四捨五入至整數儲存於 D10 內 。



補充說明：

1. 小數部份輸入注意事項：以 0.5 為例 · 輸入時 · 需輸入 S=0 · S+1=0 · S+2=16#5000 。
2. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內) · 則將會產生運算錯誤 · SM0=ON · 錯誤碼 SR0=16#200D 。
3. S 運算元指定的正弦值數值只能介於 -1.0~+1.0 之間 · 若不在此範圍 · 指令不執行 · SM0=ON · 錯誤碼 SR0=16#2003 。
4. S 運算元 · 若使用 ISPSOft 宣告 · 則資料型態為 ARRAY [3] of WORD/INT 。

API	指令碼			運算元							功能						
1524		BATAN	P	S · D							BCD ATAN 運算						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
D	●	●			●	●		●	●		●	○	●				
脈波執行型										16 位元指令 (5 steps)			32 位元指令				
AH500										AH500			-				

符號：



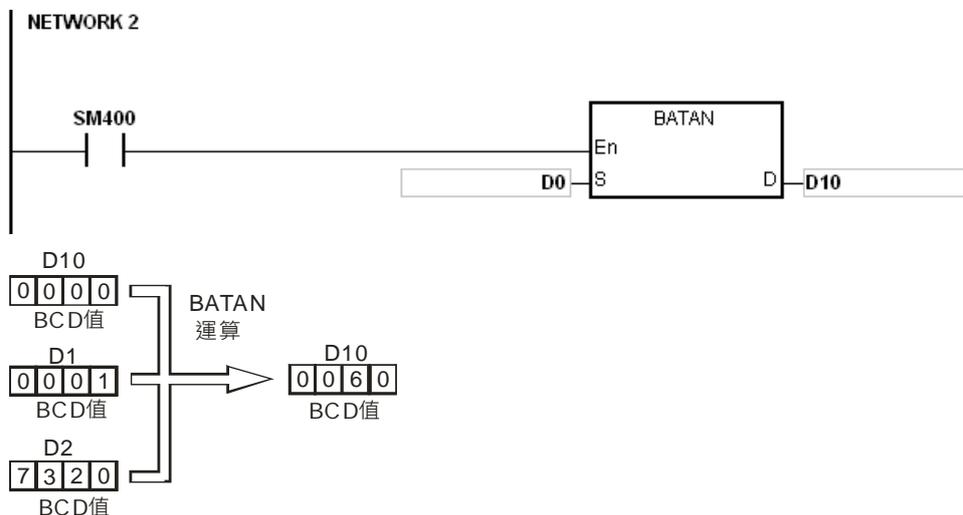
指令說明：

1. S 所指定的來源 (BCD)，求取 ATAN 值 = \tan^{-1} 值，運算的結果存放於 D (角度)。
2. S 的內容值代表符負，0=正值，1=負值，S+1 代表整數部份，S+2 代表小數部份。
3. 運算結果將小數四捨五入。
4. 運算結果為 0°到 90°和 270°到 360°之間的 BCD 值 (角度)。

程式範例：

D0 內容為 \tan^{-1} 之正負號，D1 內容為 \tan^{-1} 之整數部分，D2 值為 \tan^{-1} 小數部份。經過 BATAN 指令運算後，其結果四捨五入至整數儲存於 D10 內。

6



補充說明：

1. 小數部份輸入注意事項：以 0.5 為例，輸入時，需輸入 S=0，S+1=0，S+2=16#5000。
2. 當 S 的資料內容並非為 BCD 值 (以 Hex 表示有任一數不在 0~9 的範圍內)，則將會產生運算錯誤，SM0=ON，錯誤碼 SR0=16#200D。
3. S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

6.17 萬年曆指令

6.17.1 萬年曆指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
1600	TRD	–	✓	萬年曆資料讀出	3
1601	TWR	–	✓	萬年曆資料寫入	3
1602	T+	–	✓	萬年曆資料加算	7
1603	T-	–	✓	萬年曆資料減算	7
1604	HOUR	DHOUR	–	運轉計時器	7
1605	TCMP	–	✓	萬年曆資料比較	11
1606	TZCP	–	✓	萬年曆資料區域比較	9
1607	DST	–	✓	日光節約時間設定	15

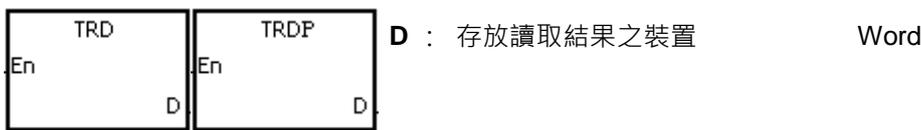
6.17.2 萬年曆指令說明

API	指令碼			運算元								功能			
1600		TRD	P	D								萬年曆資料讀出			

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

脈波執行型	16 位元指令 (3 steps)	32 位元指令
AH500	AH500	-

符號：

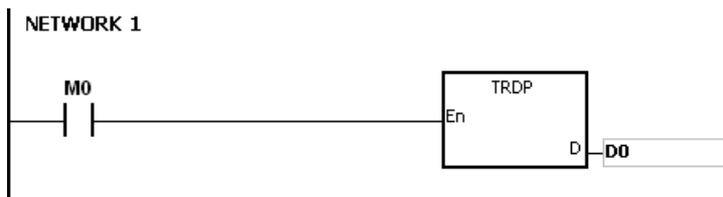


指令說明：

1. D：萬年曆現在時間讀出後存放之裝置。
2. D 運算元會佔用連續 7 個裝置。
3. 主機內建萬年曆時鐘共提供年、星期、月、日、時、分、秒及共 7 組資料存放於 SR391~SR397 當中，TRD 指令的功能就是讓程式設計者直接將萬年曆現在時間讀出至指定的 7 個暫存器當中。
4. SR391 只讀取西元年份的右 2 位。

程式範例：

當 M0=ON 時，將萬年曆現在時間讀出至指定的 D0~D6 暫存器當中，SR397 之內容 1 表星期一、2 表星期二，類推，7 表星期日。



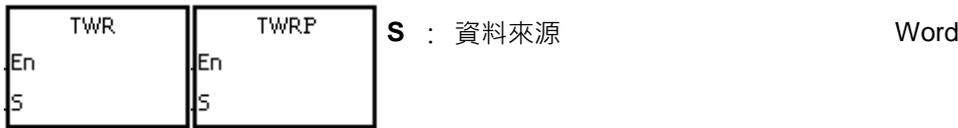
SR	項目	內容		一般 D	項目
SR391	年 (西元)	00~99	→	D0	年 (西元)
SR392	月	1~12	→	D1	月
SR393	日	1~31	→	D2	日
SR394	時	0~23	→	D3	時
SR395	分	0~59	→	D4	分
SR396	秒	0~59	→	D5	秒
SR397	星期	1~7	→	D6	星期

補充說明：

1. **D+6** 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. SM220：±30 秒補正，ON 時作補正。(0~29 秒時歸 0，30~59 秒時，分加 1、秒歸 0)。
3. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [7] of WORD/INT。

API	指令碼			運算元							功能						
1601		TWR	P	S							萬年曆資料寫入						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●	○	●				
脈波執行型											16 位元指令 (3 steps)			32 位元指令			
AH500											AH500			-			

符號：



指令說明：

1. **S**：存放欲寫入萬年曆新設定值之裝置。
2. **S** 運算元會佔用連續 7 個裝置。
3. 要調整主機內建萬年曆時鐘的時候，可使用本指令將正確的現在時間寫入至內藏萬年曆時鐘當中。
4. 本指令被執行時，新的設定時間立刻被寫入至 PLC 內部的萬年曆時鐘當中，因此，執行本指令時，請注意所寫入的新設定時間與寫入當時的現在時間是否吻合。

程式範例：

當 M0=ON 時，將正確的現在時間寫入至內藏萬年曆時鐘當中。



一般 D	項目	內容	→	SR	項目
D20	年 (西元)	00~99	→	SR391	年 (西元)
D21	月	1~12	→	SR392	月
D22	日	1~31	→	SR393	日
D23	時	0~23	→	SR394	時
D24	分	0~59	→	SR395	分
D25	秒	0~59	→	SR396	秒
D26	星期	1~7	→	SR397	星期

新的設定時間

萬年曆時鐘

補充說明：

1. 若 **S** 內容值超出範圍，則視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 若 **S+6** 超出裝置範圍，則視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

3. **S** 運算元，若使用 ISPSoft 宣告，則資料型態為 ARRAY [7] of WORD/INT。

API	指令碼			運算元							功能						
1602		T+	P	S ₁ · S ₂ · D							萬年曆資料加算						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

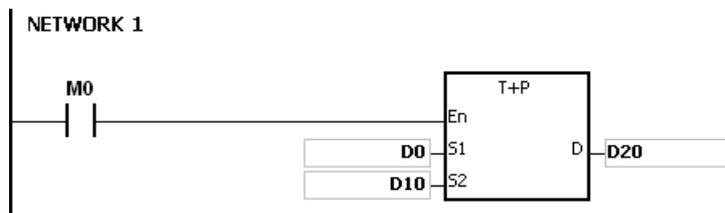


指令說明：

1. 將 S₁ 所指定的萬年曆資料時、分、秒與 S₂ 所指定的萬年曆資料時、分、秒相加，所得到的結果存於指定 D 所指定的暫存器時、分、秒當中。
2. S₁、S₂、D 運算元，皆連續佔用 3 個裝置。
3. 加算結果若大於等於 24 小時的話，進位旗標 SM602=ON、D 顯示加算總值減掉 24 小時所得的結果。
4. 加算結果若是等於 0 (0 時 0 分 0 秒)，零旗標 SM600=ON。

程式範例：

當 M0=ON 時，T+指令執行，將 D0~D2 所指定的萬年曆資料時、分、秒與 D10~D12 所指定的萬年曆資料時、分、秒相加，所得到的結果存於 D20~D22 所指定的暫存器中得到加總後之時、分、秒。



D0 8(時)	+	D10 6(時)	→	D20 14(時)
D1 10(分)		D11 40(分)		D21 50(分)
D2 20(秒)		D12 6(秒)		D22 26(秒)

8 時 10 分 20 秒 6 時 40 分 6 秒 14 時 50 分 26 秒

補充說明：

1. 若 **S₁**、**S₂** 內容值超出範圍，則視為運算錯誤，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2003**。
2. 若 **S₁+2**、**S₂+2**、**D+2** 超出裝置範圍，則視為運算錯誤，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2003**。
3. **S₁** 運算元，若使用 ISPSoft 宣告，則資料型態為 **ARRAY [3] of WORD/INT**。
4. **S₂** 運算元，若使用 ISPSoft 宣告，則資料型態為 **ARRAY [3] of WORD/INT**。
5. **D** 運算元，若使用 ISPSoft 宣告，則資料型態為 **ARRAY [3] of WORD/INT**。

API	指令碼			運算元						功能							
1603		T-	P	$S_1 \cdot S_2 \cdot D$						萬年曆資料減算							

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

T-		T-P			
En		En		S ₁	: 變換來源裝置 Word
S1	D	S1	D	S ₂	: 變換來源裝置 Word
S2		S2		D	: 存放變換結果之裝置 Word

指令說明：

- 將 S₁ 所指定的萬年曆資料時、分、秒減掉 S₂ 所指定的萬年曆資料時、分、秒，所得到的結果存於指定 D 所指定的暫存器時、分、秒當中。
- S₁、S₂、D 運算元，皆連續佔用 3 個裝置。
- 減算結果若為負數時，借位旗號 SM601=ON、該負數再加上 24 小時所得的結果顯示 D 所指定的暫存器當中。
- 減算結果若是等於 0 (0 時 0 分 0 秒)，零旗標 SM600=ON。

程式範例：

- 當 M0=ON 時，T-指令執行，將 D0~D2 所指定的萬年曆資料時、分、秒與 D10~D12 所指定的萬年曆資料時、分、秒相減，所得到的結果存於指定 D20~D22 所指定的暫存器時、分、秒當中。

D0 20(時)	-	D10 14(時)	→	D20 5(時)
D1 20(分)		D11 30(分)		D21 49(分)
D2 5(秒)		D12 8(秒)		D22 57(秒)

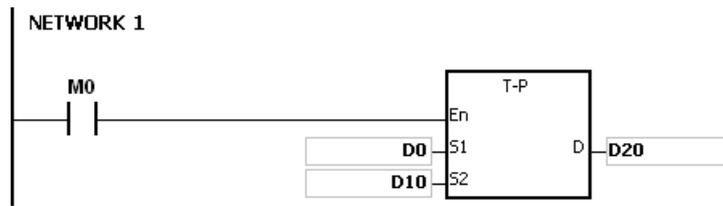
20 時 20 分 5 秒 14 時 30 分 8 秒 5 時 49 分 57 秒

- 減算結果若為負數時，借位旗號 SM601=ON。

5(時)	-	19(時)	→	10(時)
20(分)		11(分)		9(分)
30(秒)		15(秒)		15(秒)

5 時 20 分 30 秒 19 時 11 分 15 秒 10 時 9 分 15 秒

6



補充說明：

1. 若 S_1 、 S_2 內容值超出範圍，則視為運算錯誤，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. 若 S_1+2 、 S_2+2 、 $D+2$ 超出裝置範圍，則視為運算錯誤，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
3. S_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
4. S_2 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
5. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元							功能						
1604	D	HOUR		S · D ₁ · D ₂							運轉計時器						

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

脈波執行型	16 位元指令(7 steps)	32 位元指令(7 steps)
-	AH500	AH500

符號：

HOUR		D HOUR		S	Word/Double Word
En		En		D ₁	Word/Double Word
S	D1	S	D1	D ₂	bit
	D2		D2		

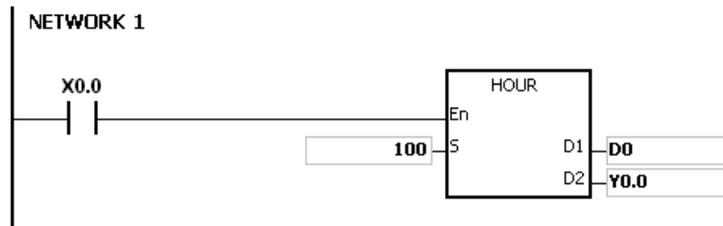
指令說明：

- S：設定導通時間，單位：小時。D₁：測量中之現在時間值，單位：小時。D₂：輸出裝置。
- S 為設定導通時間，單位：小時。
16 位元指令：1~32,767
32 位元指令：1~2,147,483,647
- HOUR：
D₁ 為測量中之現在時間值（0~32,767），單位：小時。
D₁+1 為未滿 1 個小時的現在時間值，設定範圍 0~3,599，單位：秒。
D₁+2 系統內部使用，指令執行時不可更改，否則會造成時間計算錯誤。
測量中之現在時間值到達最大數值 32,767 小時、3,599 秒時會停止計時測量，要重新計時須將現在時間值 D₁、D₁+1 清除為 0。
- D HOUR：
(D₁+1、D₁) 為測量中之現在時間值（0~2,147,483,647），單位：小時，連續佔用兩個裝置。
D₁+2 為未滿 1 個小時的現在時間值，設定範圍 0~3,599，單位：秒，佔用一個裝置。
D₁+3 系統內部使用，指令執行時不可更改，否則會造成時間計算錯誤，佔用一個裝置。
測量中之現在時間值到達最大數值 2,147,483,647 小時、3,599 秒時會停止計時測量，要重新計時須將現在時間值 D₁、D₁+1、D₁+2 清除為 0。
- 將輸入接點導通時間做計時，當到達設定時間時（以小時為單位），會將輸出裝置導通，計數未到達時，輸出裝置不導通。可提供使用者管理機械的運作計時或維修。
- 當輸出裝置導通後，計時器會繼續計時。
- 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。

6

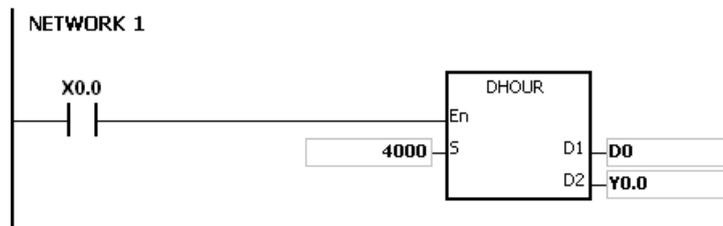
程式範例一：

16 位元指令當 X0.0=ON 時，開始計時，當到達 100 小時 Y0.0 導通，而 D0 會記錄測量中之現在時間值（單位：小時），D1 會記錄測量中不足 1 小時之現在時間值 0~3599（單位：秒），D2 程式保留使用，執行中不得更改內容值，否則會造成錯誤。



程式範例二：

32 位元指令當 X0.0=ON 時，開始計時，當到達 4000 小時 Y0.0 導通，而 D1、D0 會記錄測量中之現在時間值（單位：小時），D2 會記錄測量中不足 1 小時之現在時間值 0~3599（單位：秒），D3 程式保留使用，執行中不得更改內容值，否則會造成錯誤。



補充說明：

1. $S \leq 0$ 時指令不執行，輸出裝置狀態不變。
2. HOUR 中 $D_1 < 0$ 時指令不執行，輸出裝置狀態不變。
3. HOUR 中 $D_1 + 2$ 超出裝置範圍，則視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. D HOUR 中 $(D_1 + 1, D_1) < 0$ 時指令不執行，輸出裝置狀態不變。
5. D HOUR 中 $D_1 + 3$ 超出裝置範圍，則視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
6. 16 位元指令 D_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
7. 32 位元指令 D_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of DWORD/DINT。

API	指令碼			運算元							功能						
1605		TCMP	P	$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot D$							萬年曆資料比較						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●		●	○	●	○	○		
S_2	●	●			●	●		●	●		●	○	●	○	○		
S_3	●	●			●	●		●	●		●	○	●	○	○		
S	●	●			●	●		●	●		●	○	●				
D	●	●	●	●				●	●	●			●				

脈波執行型	16 位元指令 (11 steps)	32 位元指令
AH500	AH500	-

符號：

TCMP		TCMPP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	
S		S	

- S_1 ：設定比較時間之“時”，設定範圍為「K0~K23」 Word
- S_2 ：設定比較時間之“分”，設定範圍為「K0~K59」 Word
- S_3 ：設定比較時間之“秒”，設定範圍為「K0~K59」 Word
- S ：萬年曆現在時間 Word
- D ：比較結果 bit

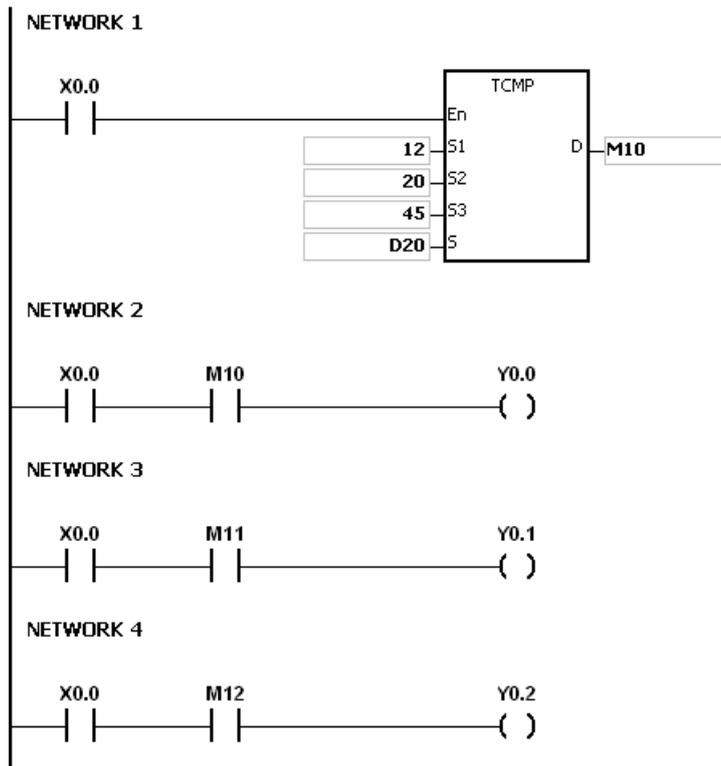
指令說明：

- 將由 $S_1 \sim S_3$ 所指定的時、分、秒設定值與 S 起始之萬年曆時、分、秒現在值做比較，其比較結果在 D 作表示。
- S 為萬年曆現在時間之“時”，內容為 0~23。 $S+1$ 為萬年曆現在時間之“分”，內容為 0~59。 $S+2$ 為萬年曆現在時間之“秒”，內容為 0~59。
- 運算元 D 佔用 3 個連續裝置。 D 、 $D+1$ 、 $D+2$ 用於儲存比較結果。
- 通常 S 所指定的萬年曆現在時間通常是預先使用 TRD 指令將萬年曆現在時間讀入後再使用 TCMP 指令進行比較。
- 設定值 ($S_1 \sim S_3$) > S 現在值 $D=ON$ 、 $D+1=OFF$ 、 $D+2=OFF$ 。
- 設定值 ($S_1 \sim S_3$) = S 現在值 $D=OFF$ 、 $D+1=ON$ 、 $D+2=OFF$ 。
- 設定值 ($S_1 \sim S_3$) < S 現在值 $D=OFF$ 、 $D+1=OFF$ 、 $D+2=ON$ 。

程式範例：

- 當 $X0.0=ON$ 時，指令執行，將 $D20 \sim D22$ 萬年曆現在時間與設定值 12 時 20 分 45 秒做比較，將結果顯示到 $M10 \sim M12$ 。當 $X0.0$ 由 $ON \rightarrow OFF$ 變化時，指令不被執行，但是 $M10 \sim M12$ 之前的 ON/OFF 狀態仍被保持住。
- 若需要得到 \geq 、 \leq 、 \neq 之結果時，可將 $M10 \sim M12$ 串並聯即可取得。

6



補充說明：

1. **S+2** 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. **D+2** 超出裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 若 **S** 內容值超出範圍，則視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. 若 **S₁~S₃** 內容值超出範圍，則視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
5. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of BOOL。

6

API	指令碼		運算元							功能							
1606		TZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$							萬年曆資料區域比較						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

TZCP		TZCPP	
En		En	
S1	D	S1	D
S2		S2	
S		S	

- S_1 : 設定比較時間之下限值 Word
- S_2 : 設定比較時間之上限值 Word
- S : 萬年曆現在時間 Word
- D : 比較結果 bit

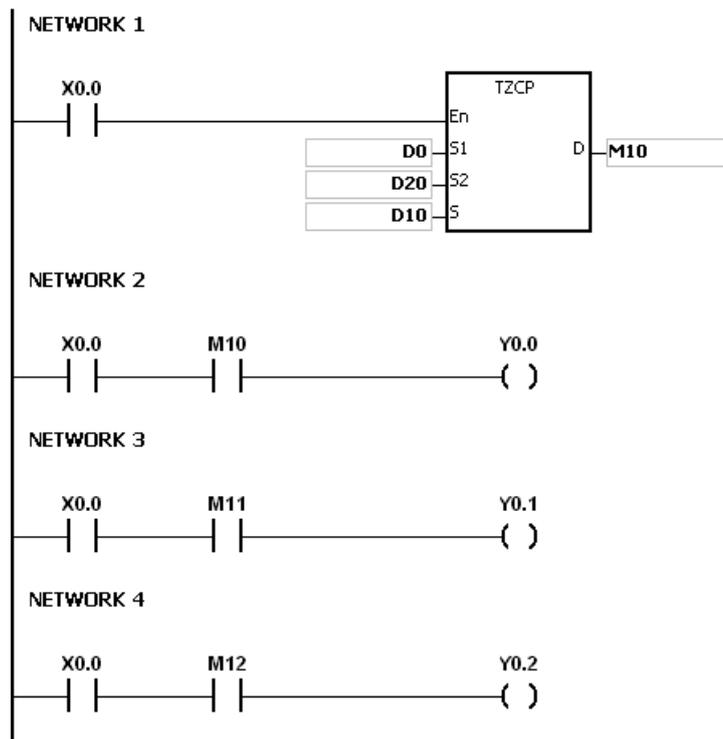
指令說明：

1. 將由 S 所指定的萬年曆現在時間時、分、秒值與 S_1 所指定設定比較時間之下限值及 S_2 所指定設定比較時間之上限值做區域比較，其比較結果在 D 作表示。
2. S_1 、 S_1+1 、 S_1+2 ：設定比較時間下限值的“時”、“分”、“秒”。
3. S_2 、 S_2+1 、 S_2+2 ：設定比較時間上限值的“時”、“分”、“秒”。
4. S 、 $S+1$ 、 $S+2$ ：為萬年曆現在時間的“時”、“分”、“秒”。
5. 運算元 S_1 時間須比 S_2 時間小。當 $S_1 > S_2$ ，指令執行時把 S_1 時間作為上/下限值進行比較。
6. 通常 S 所指定的萬年曆現在時間通常是預先使用 TRD 指令將萬年曆現在時間讀入後再使用 TZCP 指令進行比較。
7. 當現在時間 S 小於下限值 S_1 且 S 小於上限值 S_2 時，則 D 為 ON，當現在時間 S 大於下限值 S_1 且 S 大於上限值 S_2 時，則 $D+2$ 為 ON，其餘狀態則 $D+1$ 為 ON。

程式範例：

當 X0.0=ON 時，TZCP 指令執行，M10~M12 其中之一會 ON，當 X0.0=OFF 時，TZCP 指令不執行，M10~M12 狀態保持在 X0.0=OFF 之前的狀態。

6



補充說明：

1. S_{i+2} 、 S_{2+2} 、 $S+2$ 、 $D+2$ 超出裝置範圍時，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
2. S_1 、 S_2 、 S 內容值超出範圍，則視為運算錯誤，指令不執行， $SM0=ON$ ，錯誤碼 $SR0=16\#2003$ 。
3. S_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
4. S_2 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
5. S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
6. D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼		運算元					功能					
1607		DST	P	S · S₁ · S₂ · S₃ · S₄ · S₅ · D					日光節約時間				

裝置	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S					●	●		●					○	○		
S₁					●	●		●					○	○		
S₂					●	●		●					○	○		
S₃					●	●		●					○	○		
S₄					●	●		●					○	○		
S₅					●	●		●					○	○		
D		●	●	●				●								

脈波執行型	16 位元指令 (15 steps)	32 位元指令
AH500	AH500	-

符號：

DST		DSTP	
En		En	
S	D	S	D
S1		S1	
S2		S2	
S3		S3	
S4		S4	
S5		S5	

S : 日光節約時間功能代碼
S₁ : 節約開始時間之月份
S₂ : 節約開始時間之日期
S₃ : 節約結束時間之月份
S₄ : 節約結束時間之日期
S₅ : 節約時間 (分鐘)
D : 節約功能狀態

6

指令說明：

1. DST 指令中各運算元的說明如下：

S : 日光節約時間功能代碼：

功能代碼	功能
0	關閉日光節約時間
1	啟動模式 1 日光節約時間
2	讀取日光節約時間
3	啟動模式 2 日光節約時間
4	系統設定，表示目前節約功能為關閉
5	系統設定，表示目前節約功能為啟動模式 1
7	系統設定，表示目前節約功能為啟動模式 2
6, 8~	保留，自動視為讀取日光節約時間

註 1：當 **S**=4、5、7 時，此指令不會執行任何動作

註 2：各功能代碼的使用方式後續會有詳細說明

S₁ : 設定日光節約時間 (Daylight Saving Time) 的開始“月份”

S₂：設定日光節約時間 (D.S.T.) 的開始日期，會根據 **S** (啟動模式) 的值而有所不同。

設定 **S=1** (啟動模式 1) 時，**S₂** 設定節約開始之日期為“日”

設定 **S=3** (啟動模式 2) 時，**S₂** 設定節約開始之日期為“第 **S₂** 個星期 **S₂+1**”

S₃：設定日光節約時間 (Daylight Saving Time) 的結束“月份”

S₄：設定日光節約時間 (D.S.T.) 的結束日期，會根據 **S** (啟動模式) 的值而有所不同。

設定 **S=1** (啟動模式 1) 時，**S₄** 設定節約結束之日期為“日”

設定 **S=3** (啟動模式 2) 時，**S₄** 設定節約結束之日期為“第 **S₄** 個星期 **S₄+1**”

S₅：設定節約時間，其單位為“分鐘”。

D：儲存指令執行後之狀態，OFF 表示功能關閉，ON 表示功能啟動。

2. DST 指令支援 D.S.T (日光節約時間) 的功能，其使用說明如下：

D.S.T 功能	S 功能代碼	說明
關閉	0	關閉日光節約時間
啟動	1、3	啟動日光節約時間
讀取	2	讀取日光節約時間

● **關閉日光節約時間功能**：(參考範例一)

設定 **S** 運算元為 0 表“關閉”功能，其餘參數 **S₁~S₅** 運算元都為無意義，可輸入任意值，並將目前的節約狀態輸出 **D** 運算元為 OFF。

● **啟動日光節約時間功能**：(參考範例二、三)

設定 **S** 運算元為 1 或 3 (決定啟動模式) 表“啟動”功能，其中 **S₁、S₂** 運算元為開始之日期，其中 **S₃、S₄** 運算元為結束之日期，**S₅** 運算元為節約時間，並將目前的節約狀態輸出 **D** 運算元。

當日光節約時間為“啟動”功能時，並且系統時間之日期第一次進入開始日期 (**S₁、S₂**) 時，其系統內存時間將自動加上節約時間 **S₅** 一次；反之，系統時間之日期第一次進入結束日期 (**S₃、S₄**) 時，其系統內存時間將自動減去節約時間 **S₅** 一次。

啟動日光節約時間的功能有下面幾種：

啟動模式 1 (**S=1**)：依“月”、“日”做啟動，各運算元設定如下：(參考範例二)

運算元	說明
S₁	節約開始時間之月份 限制輸入數值範圍 1~12
S₂	節約開始時間之日期“日” 限制輸入數值範圍 1~31
S₃	節約結束時間之月份 限制輸入數值範圍 1~12
S₄	節約結束時間之日期“日” 限制輸入數值範圍 1~31

運算元	說明
S₅	節約時間 (分鐘) 限制輸入數值範圍 1~1439 (表限制 1 天內)

註 1：若啟動成功，目前的節約狀態輸出 **D** 運算元為 ON。

註 2：若是設定日期錯誤 (如：4 月 31 日)，或者開始日期大於等於結束日期 (如：開始 10 月 1 日、結束 4 月 1 日)，則“啟動”功能將無法成功設定，且會設定 **SM0=ON**，錯誤碼 **SR0=16#200B**。

註 3：若 **S₅** 超過限制輸入數值，則“啟動”功能將無法成功設定，且會設定 **SM0=ON**，錯誤碼 **SR0=16#200B**。

啟動模式 2 (**S=3**)：依“星期”做啟動，各運算元設定如下：(參考範例三)

運算元	說明
S₁	節約開始時間之月份 限制輸入數值範圍 1~12
S₂~S₂₊₁	節約開始時間之日期“星期” (第 S₂ 個星期 S₂₊₁) S₂ 限制輸入數值範圍 1~4 或 -1 ~ -4 S₂₊₁ 限制輸入數值範圍 1~7
S₃	節約結束時間之月份 限制輸入數值範圍 1~12
S₄~S₄₊₁	節約結束時間之日期“星期” (第 S₄ 個星期 S₄₊₁) S₄ 限制輸入數值範圍 1~4 或 -1 ~ -4 S₄₊₁ 限制輸入數值範圍 1~7 (星期一~星期日)
S₅	節約時間 (分鐘)

註 1：若啟動成功，目前的節約狀態輸出 **D** 運算元為 ON。

註 2：**S₂** 與 **S₄** 限制輸入數值為 1~4 或 -1 ~ -4。其中 -1 表示當月份倒數第一個星期 **S₂₊₁** 與 **S₄₊₁**，-2 表示倒數第 2 個星期 **S₂₊₁** 與 **S₄₊₁**，舉例：倒數第 1 個週日是 28 號，那倒數第 2 個週日就是 21 號。若輸入數值超出限制範圍，則“啟動”功能將無法成功設定，且會設定 **SM0=ON**，錯誤碼 **SR0=16#200B**。

註 4：若 **S₂₊₁** 與 **S₄₊₁** 輸入數值超出限制範圍，皆預設為星期日。

註 5：若 **S₅** 超過限制輸入數值，則“啟動”功能將無法成功設定，且會設定 **SM0=ON**，錯誤碼 **SR0=16#200B**。

註 6：若運算元 **S₂** 與 **S₄** 選擇裝置為 K 或 16#元件，則無法被儲存，且會被設定 **SM0=ON**，錯誤碼 **SR0=16#2003**。

- 讀取日光節約時間功能：(參考範例一~三)

設定 **S** 運算元為 2 表“讀取”功能，其中 **S₁**、**S₂** 運算元將儲存開始之日期，**S₃**、**S₄** 運算元將儲存結束之日期，**S₅** 運算元將儲存節約時間，並將目前的節約狀態輸出 **D** 運算元。若設定為“讀取”後，運算元 **S**，**S₁~S₅** 選擇裝置為 **K** 或 **16#**元件，則無法被儲存，且會被設定 **SM0=ON**，錯誤碼 **SR0=16#2003**。

在讀取動作完成時，會自動將 **DST** 現有設定之 **S** 功能碼加 4 顯示，之後不再執行任何動作。
舉例：0、1、3 這三種功能碼，在被讀取完成後將分別顯示 4、5、7。

如果讀出目前 **DST** 狀態為“關閉”，**S₁~S₅** 及 **D** 各運算元儲存的資料如下：

運算元	說明
S	系統自動修正功能代碼為 4，表示目前節約狀態為關閉
S₁~S₅	無效運算元，不用理會
D	節約狀態輸出 OFF (關閉)

如果讀出目前 **DST** 狀態為“啟動模式 1”，**S₁~S₅** 及 **D** 各運算元儲存的資料如下：

運算元	說明
S	系統自動修正功能代碼為 5，表示目前節約狀態為啟動模式 1
S₁	節約開始時間之月份
S₂	節約開始時間之日期“日”
S₃	節約結束時間之月份
S₄	節約結束時間之日期“日”
S₅	節約時間 (分鐘)
D	節約狀態輸出 ON (啟動)

如果讀出目前 **DST** 狀態為“啟動模式 2”，**S₁~S₅** 及 **D** 各運算元儲存的資料如下：

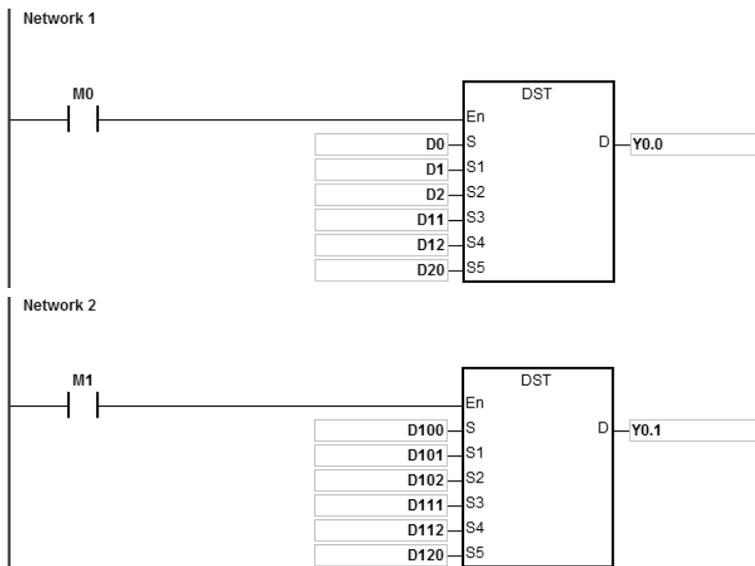
運算元	說明
S	系統自動修正功能代碼為 7，表示目前節約狀態為啟動模式 2
S₁	節約開始時間之月份
S₂~S₂₊₁	節約開始時間之日期“星期” (第 S₂ 個星期 S₂₊₁)
S₃	節約結束時間之月份
S₄~S₄₊₁	節約結束時間之日期“星期” (第 S₄ 個星期 S₄₊₁)
S₅	節約時間 (分鐘)
D	節約狀態輸出 ON (啟動)

3. 此指令僅做為日光節約功能的開啟與關閉，因此指令前面的條件接點被啟動或關閉，都不會重複設定日光節約功能的時間 (參考範例二中重新啟動條件接點 **M0 OFF=>ON** 的部分)。若功能為已啟動狀態，並且要修改設定開始與結束時間，則可直接重新執行指令的啟動功能即可，不需要先關閉日光節約功能再重新啟動日光節約功能。

4. 若設定日光節約時間日期為 4 月 1 日~9 月 1 日，節約 60 分鐘，則萬年曆的時間變化如下：

關閉啟動日光節約功能 (預設)	啟動日光節約功能
3 月 01 日 3 時	3 月 01 日 3 時
3 月 31 日 3 時	3 月 31 日 3 時
4 月 01 日 3 時	4 月 01 日 4 時
5 月 01 日 3 時	5 月 01 日 4 時
6 月 01 日 3 時	6 月 01 日 4 時
7 月 01 日 3 時	7 月 01 日 4 時
8 月 01 日 3 時	8 月 01 日 4 時
8 月 31 日 3 時	8 月 31 日 4 時
9 月 01 日 3 時	9 月 01 日 3 時

範例一、關閉 D.S.T.功能，並讀取目前 D.S.T 的狀態。



6

給定下面設定值：

裝置	內容值	說明
D0	0	關閉 D.S.T.功能
D1	X	無意義運算元，可輸入任意值
D2	X	無意義運算元，可輸入任意值
D11	X	無意義運算元，可輸入任意值
D12	X	無意義運算元，可輸入任意值
D20	X	無意義運算元，可輸入任意值

啟動條件接點 M0

監控 Y0.0=OFF 表示節約功能關閉中

給定 D100=K2，表示要讀取 DST 目前的節約狀態。

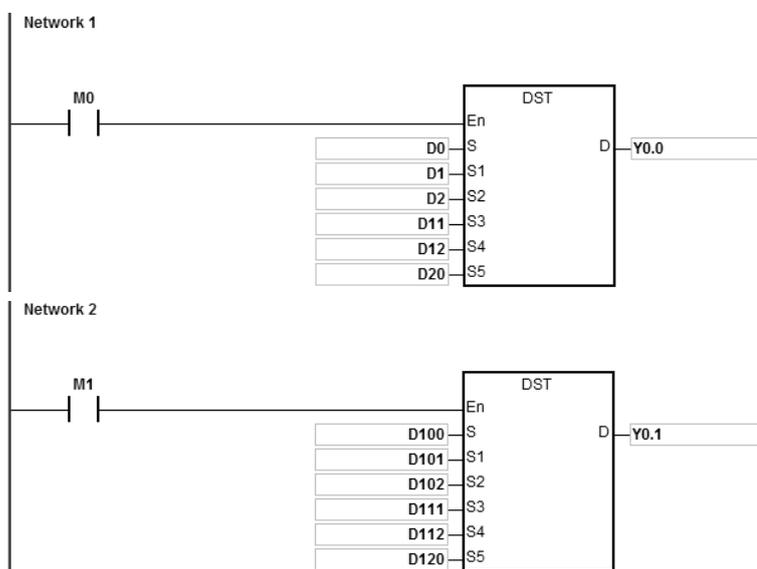
啟動條件接點 M1

讀出各裝置對應的內容值如下：

裝置	內容值	說明
D100	4	系統自行變更為 4，表示已經關閉 D.S.T 功能
D101	X	無意義運算元，不用理會
D102	X	無意義運算元，不用理會
D111	X	無意義運算元，不用理會
D112	X	無意義運算元，不用理會
D120	X	無意義運算元，不用理會
Y0.1	OFF	表示節能狀態為關閉

範例二、啟動模式 1 開啟 D.S.T.功能，並讀取目前 D.S.T 的狀態。

設定日光節約時間為 4 月 1 日開始，9 月 3 日結束（這天會恢復），且節約時間為 60 分鐘。



給定下面設定值：

裝置	內容值	說明
D0	1	啟動模式 1 開啟 D.S.T.功能
D1	4	起始月份：4 月
D2	1	起始日期：1 日
D11	9	結束月份：9 月
D12	3	結束日期：3 日
D20	60	單位節約時間為 60 分鐘

啟動條件接點 M0

監控 Y0.0=ON 表示節約功能啟動中

也就是，當 PLC 的萬年曆時間執行到 4 月 1 日的 0 時 0 分 0 秒時會自動增加 60 分鐘，但到 9 月 3 日的 0 時 0 分 0 秒時會自動恢復為未節約的時間。

給定 D100=K2，表示要讀取 DST 目前的節約狀態。

啟動條件接點 M1

讀出各裝置對應的內容值如下：

裝置	內容值	說明
D100	5	系統自行變更為 5，表示啟動模式 1 開啟 D.S.T.功能
D101	4	起始月份：4 月
D102	1	起始日期：1 日
D111	9	結束月份：9 月
D112	3	結束日期：3 日
D120	60	單位節約時間為 60 分鐘
Y0.1	ON	表示節能狀態為 ON

若重新啟動條件接點 M0 OFF=>ON，不會因為重複執行條件接點就重複設定日光節約的時間

讀出各裝置對應的內容值如下：

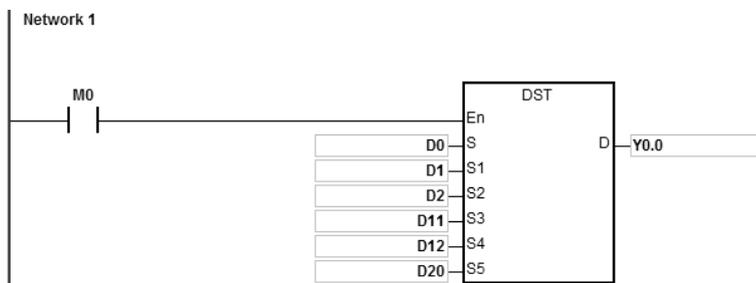
裝置	內容值	說明
D100	5	系統自行變更為 5，表示啟動模式 1 開啟 D.S.T.功能
D101	4	起始月份：4 月
D102	1	起始日期：1 日
D111	9	結束月份：9 月
D112	3	結束日期：3 日
D120	60	單位節約時間為 60 分鐘
Y0.1	ON	表示節能狀態為 ON

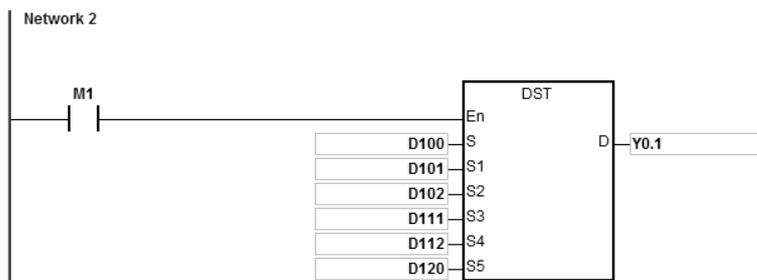
6

如果 4 月 1 日 3 時節約後為 4 月 1 日 4 時，此時不管重複幾次條件接點 M0 OFF=>ON，則節約後還是 4 月 1 日 4 時，並不會因為重複執行條件接點就重複設定日光節約的時間。

範例三、啟動模式 2 開啟 D.S.T.功能

設定日光節約時間為 5 月的第 2 個星期三開始，9 月的第 3 個星期五結束（這天會恢復），且節約時間為 60 分鐘。





給定下面設定值：

裝置	內容值	說明
D0	3	啟動模式 2 開啟 D.S.T.功能
D1	5	起始月份：5 月
D2	2	起始日期：第 2 個星期
D3	3	起始日期：星期三
D11	9	結束月份：9 月
D12	3	結束日期：第 3 個星期
D13	5	結束日期：星期五
D20	60	單位節約時間為 60 分鐘

啟動條件接點 M0

監控 Y0.0=ON 表示節約功能啟動中

如果是在 2017 年時執行這個命令，則 5 月的第 2 個星期三就是 5 月 10 日，9 月的第 3 個星期五就是 9 月 15 日；也就是，當 PLC 的萬年曆時間執行到 5 月 10 日的 0 時 0 分 0 秒時會自動增加 60 分鐘，但到 9 月 15 日的 0 時 0 分 0 秒時會自動恢復為未節約的時間。

給定 D100=K2，表示要讀取 DST 目前的節約狀態。

啟動條件接點 M1

讀出各裝置對應的內容值如下：

裝置	內容值	說明
D100	7	系統自行變更為 7，表示啟動模式 2 開啟 D.S.T.功能。
D101	5	起始月份：5 月
D102	2	起始日期：第 2 個星期
D103	3	起始日期：星期三
D111	9	結束月份：9 月
D112	3	結束日期：第 3 個星期
D113	5	結束日期：星期五
D120	60	單位節約時間為 60 分鐘
Y0.1	ON	表示節能狀態為 ON

補充說明：

DST 指令支援機種如下

1. AHCPU5X1-EN V2.01 以後之版本
2. AHCPU5X1-RS2 V1.03 以後之版本
3. AHCPU560-EN2 V1.10 以後之版本

6

6.18 週邊設備指令

6.18.1 週邊設備指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1700</u>	TKY	DTKY	—	10 鍵盤輸入	7
<u>1701</u>	HKY	DHKY	—	16 鍵盤輸入	9
<u>1702</u>	DSW	—	—	指撥開關輸入	9
<u>1703</u>	ARWS	—	—	箭頭鍵盤輸入	9
<u>1704</u>	SEGL	—	—	七段顯示器掃描輸出	7

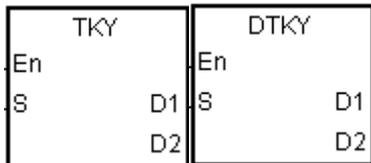
6.18.2 週邊設備指令說明

API	指令碼			運算元								功能					
1700	D	TKY		S · D ₁ · D ₂								10 鍵鍵盤輸入					

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
-	AH500	AH500

符號：



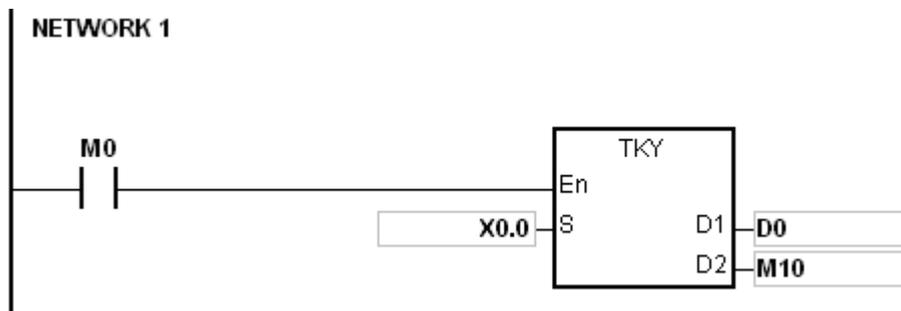
- S** : 按鍵輸入起始裝置 Bit
- D₁** : 按鍵輸入值存放處 Word/Double Word
- D₂** : 按鍵輸出信號 Bit

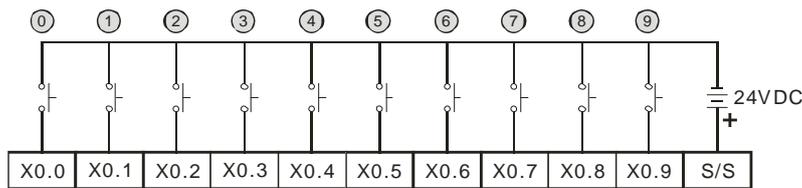
指令說明：

1. 本指令指定由 **S** 開始之 10 個外部輸入點，依序代表 10 進位數字之 0~9。這 10 個外部輸入點分別接上 10 個按鍵，依據這 10 個按鍵被壓下之先後順序可輸入 4 位 10 進位數字 0~9,999 (16 位元指令)，或 8 位 10 進位數字 0~99,999,999 (32 位元指令)，並將輸入之數值存放在 **D₁**，而 **D₂** 則存放鍵盤之按鍵情形。
2. **S** 連續佔用 10 個 bit。
3. **D₂** 連續佔用 11 個 bit，指令執行中，請勿改變 bit 的狀態。
4. 當條件接點未啟動時，**D₂** 開始的連續 11 個 bit 輸出點會保持 OFF。
5. 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。

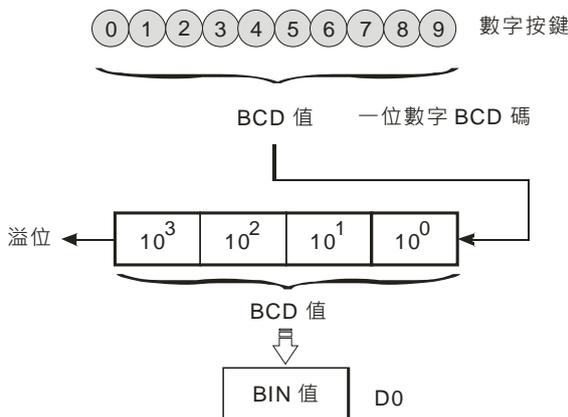
程式範例：

1. 指令指定 X0.0 開始的 10 個輸入端與 0~9 的 10 個按鍵連接，當 M0=ON 時，指令執行，將鍵盤輸入的數值以 BIN 值的形態存入 D0 中，而按鍵之情況則放在 M10~M19。

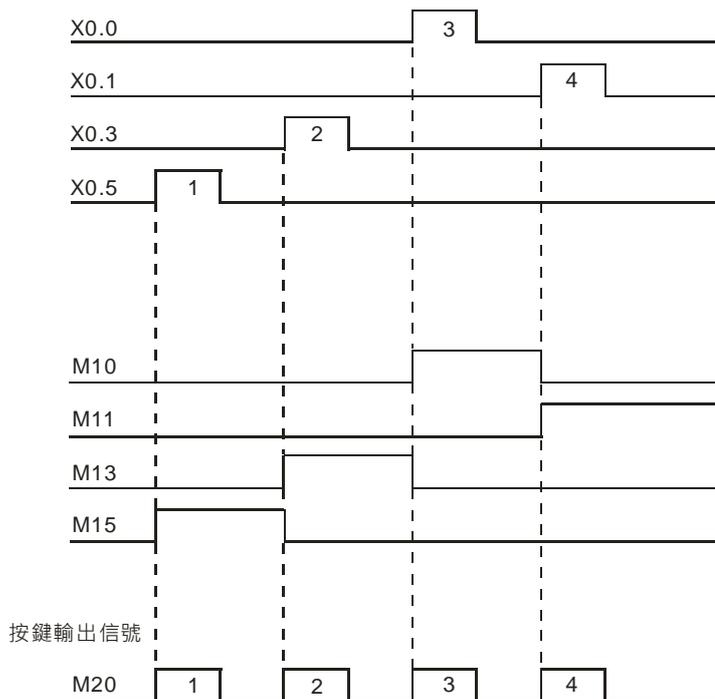




註: 此範例使用AH16AM10N數位輸入模組



2. 如下列時序圖所示，連接於數字鍵盤 X0.5、X0.3、X0.0、X0.1 的 4 個按鍵以①、②、③、④ 的順序作打入的動作，結果為 5,301 被暫存於 D0 當中，D0 最大可容納 9,999，超過 4 位數時，最前面的位數溢位。
3. X0.2 被按下後，至別的按鍵被按之前，M12=ON，其他的數按鍵亦相同。
4. 當 X0.0~X0.9 當中任何一個按鍵被按下時，M10~M19 當中一點對應 ON。
5. 任何一個按鍵被按下時，M20=ON。
6. 當條件接點 M0 變成 OFF 時，D0 之前的值無變化，但是，M10~M20 全部變成 OFF。



補充說明：

1. **S** 運算元，若使用 ISPSoft 宣告，則資料型態為 ARRAY [10] of BOOL。
2. **D₂** 運算元，若使用 ISPSoft 宣告，則資料型態為 ARRAY [11] of BOOL。

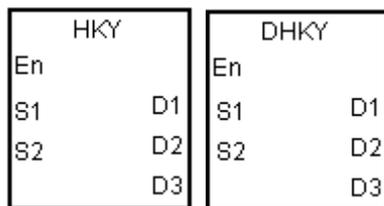
6

API	指令碼			運算元						功能							
1701	D	HKY		$S_1 \cdot S_2 \cdot D_1 \cdot D_2 \cdot D_3$						16 鍵鍵盤輸入							

裝置	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		●	●	●				●	●				●				

脈波執行型	16 位元指令 (9 steps)	32 位元指令 (9 steps)
-	AH500	AH500

符號：



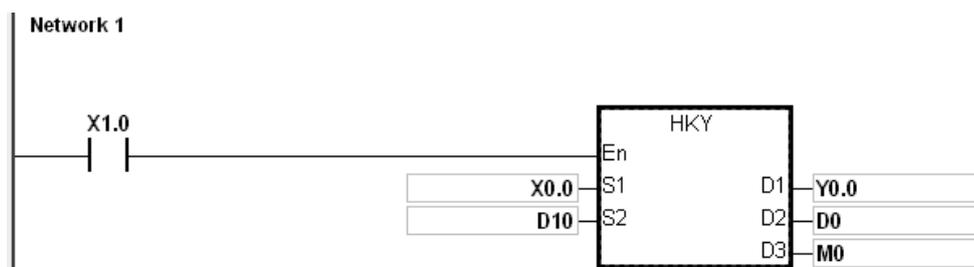
- S₁ : 按鍵掃描輸入起始裝置 Bit
- S₂ : 系統內部使用 Word
- D₁ : 按鍵掃描輸出起始裝置 Bit
- D₂ : 按鍵輸入值存放處 Word/Double Word
- D₃ : 按鍵輸出信號 Bit

指令說明：

- 本指令指定由 S₁ 開始之連續 4 個外部輸入點及由 D₁ 開始之連續 4 個外部輸出點以矩陣掃描之方式構成 16 鍵之鍵盤。鍵盤輸入之數值存放在 D₂，而 D₃ 則存放鍵盤之按鍵情形。如果有數個按鍵同時被按下時，以 S₁ 的編號小的為優先。
- 由數字鍵盤所打入的值被暫存於 D₂ 當中，使用 16 位元指令 HKY 時，D₂ 最大可容納 9,999，超過 4 位數時，最前面的位數溢位。使用 32 位元指令 DHKY 時，D₂ 最大可容納 99,999,999，超過 8 位數時，最前面的位數溢位。
- 指令執行完畢 SM692 會 ON；每執行完一次矩陣掃描，SM692 會 ON 一個掃描週期。

程式範例：

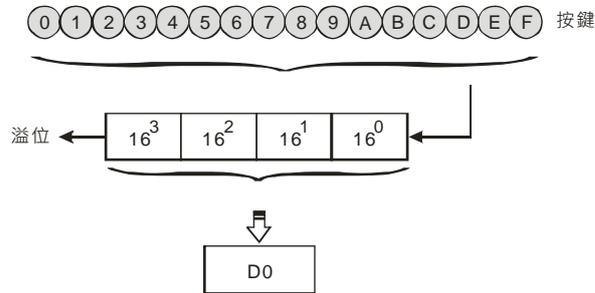
- 指令指定 X0.0~X0.3 等 4 個輸入端與 Y0.0~Y0.3 等 4 個輸出端構成掃描 16 鍵之鍵盤。當 X1.0=ON 時，指令執行，由鍵盤輸入的數值以 BIN 值的形態存入 D0 中，而按鍵之情況則放在 M0~M7。



旗標 SM691 之功能：

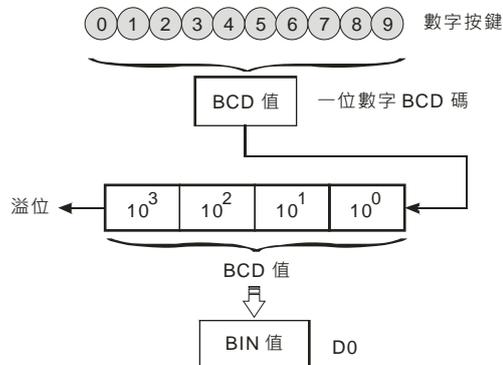
- SM691=ON 時，則 HKY 指令可以輸入 0~F 的 16 進位數值。

■ 數字輸入：



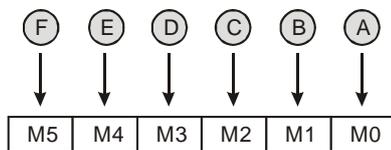
- SM691=OFF 時，則 HKY 指令 A~F 當成功能鍵使用。

■ 數字輸入：



■ 功能鍵輸入：

- ◆ 按 A 鍵時，M0=ON 並保持，接著再按 D 鍵時，M0 變成 OFF、M3=ON 並保持。
- ◆ 複數個按鍵同時按，以先按者優先。



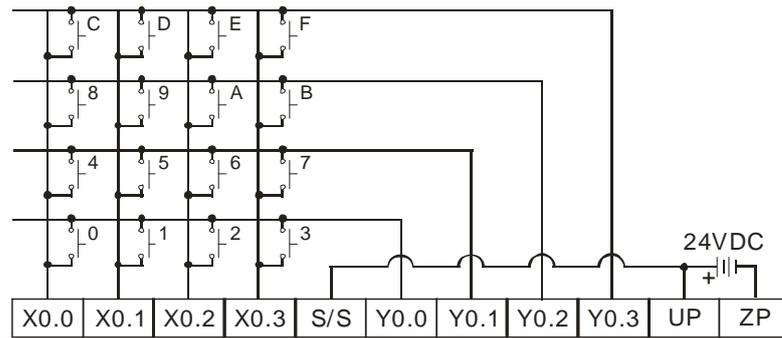
2. 按鍵輸出信號：

- A~F 當中任何一個按鍵被按時，M6=ON 一次。
- 0~9 當中任何一個按鍵被按時，M7=ON 一次。

3. 當條件接點 X1.0 變成 OFF 時，D0 之前的輸入值無變化，但是 M0~M7 全部變成 OFF。

4. 外部配線：

6



註: 本範例使用AH16AP11T電晶體輸出模組

補充說明：

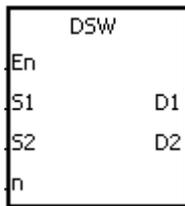
1. 本指令被執行時，當掃描週期太長或太短都可能造成按鍵輸入不良，因此可運用下列技巧來克服。
2. 當掃描週期太短時，可能造成 I/O 來不及反應而無法讀取正確之按鍵輸入，此時，可將掃描時間加以固定。
3. 當掃描週期太長時，可能會使按鍵反應變得遲鈍，可將此指令寫在時間中斷工作 (TASK) 內，固定時間執行此指令。
4. **S₁** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of BOOL。
5. **D₁** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of BOOL。
6. **D₃** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [8] of BOOL。

API	指令碼		運算元								功能				
1702		DSW	$S_1 \cdot S_2 \cdot D_1 \cdot D_2 \cdot n$								指撥開關輸入				

裝置	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	●	●						●	●		●		●	○	○		

脈波執行型	16 位元指令 (9 steps)	32 位元指令
-	AH500	-

符號：



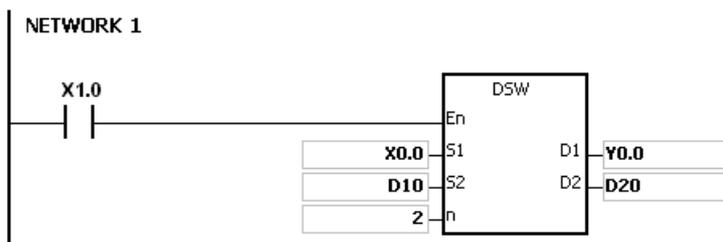
- S₁ : 指撥開關掃描輸入起始裝置 Bit
- S₂ : 系統內部使用 Word
- D₁ : 指撥開關掃描輸出起始裝置 Bit
- D₂ : 指撥開關設定值存放處 Word
- n : 指撥開關所連接之組數 Word

指令說明：

- 本指令由 S₁ 開始的連續 4 個或 8 個外部輸入點及由 D₁ 開始的連續 4 個外部輸出點掃描讀取 1 組或 2 組 4 位數指撥開關，指撥開關設定值存放在 D₂，由 n 決定讀取 4 位數指撥開關有 1 組或 2 組。
- 當 n=1 時，D₂ 運算元佔用一個暫存器。n=2 時，D₂ 運算元會連續佔用 2 個暫存器。
- S₂、S₂₊₁ 佔用兩個裝置給系統內部使用，使用者請勿變更內容值。
- 每循環掃描一次完畢旗標信號 SM694=ON 一個掃描週期。
- 當條件接點未啟動時，D₁ 開始的連續 4 個外部輸出點會保持 OFF。
- 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。

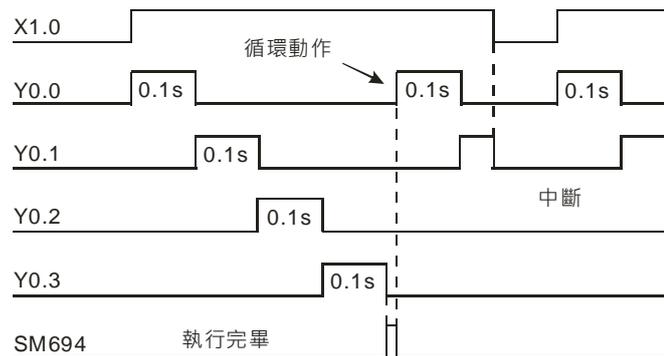
程式範例：

- 由 X0.0~X0.3 及 Y0.0~Y0.3 組成第一組指撥開關迴路，由 X0.4~X0.7 及 Y0.4~Y0.7 組成第二組指撥開關迴路。當 X1.0=ON 時，指令開始執行，第一組指撥開關的設定值被讀入並轉換成 BIN 值後存放至 D20 中，第二組指撥開關的設定值被讀入並轉換成 BIN 值後存放至 D21 中。

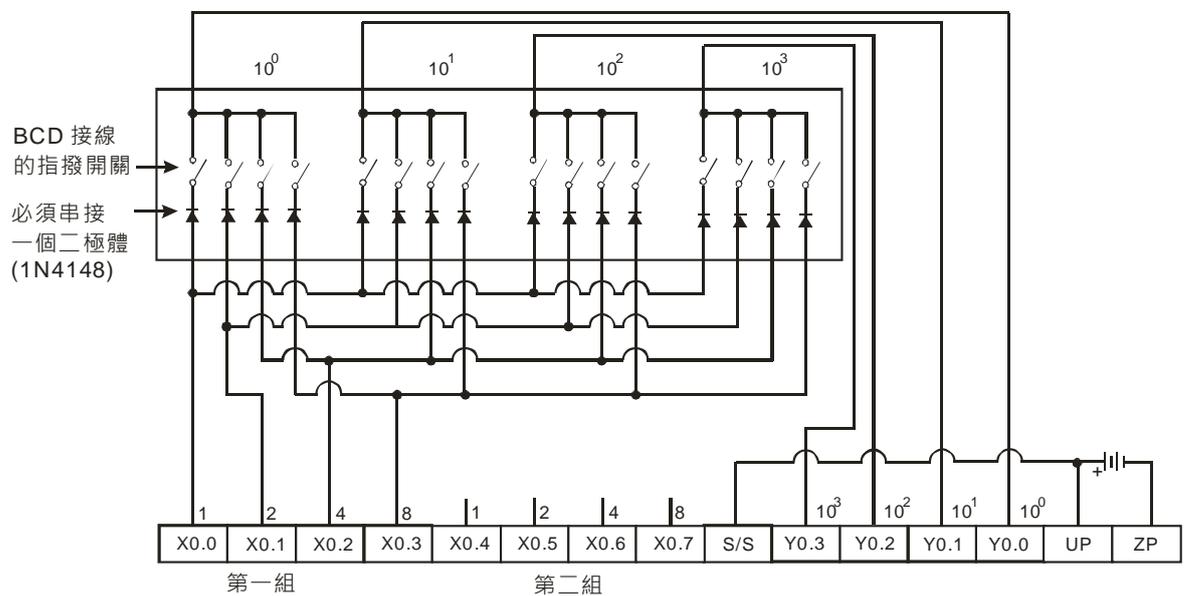


- 當 X1.0=ON 時，Y0.0~Y0.3 自動循環掃描 ON，每循環一次，執行完畢旗標信號 SM694=ON 一個掃描週期。

3. 掃描用輸出端 Y0.0~Y0.3 請使用電晶體輸出。



4. 指撥開關輸入



註: 此範例使用AH16AP11T電晶體輸出模組

補充說明：

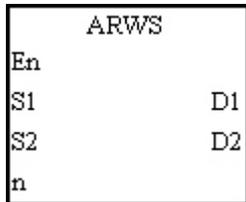
1. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
2. **D₁** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of BOOL。

API	指令碼			運算元							功能						
1703	ARWS			$S_1 \cdot S_2 \cdot D_1 \cdot D_2 \cdot n$							箭頭鍵盤輸入						

裝置	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	●	●						●	●		●		●	○	○		

脈波執行型	16 位元指令 (9 steps)	32 位元指令
-	AH500	-

符號：



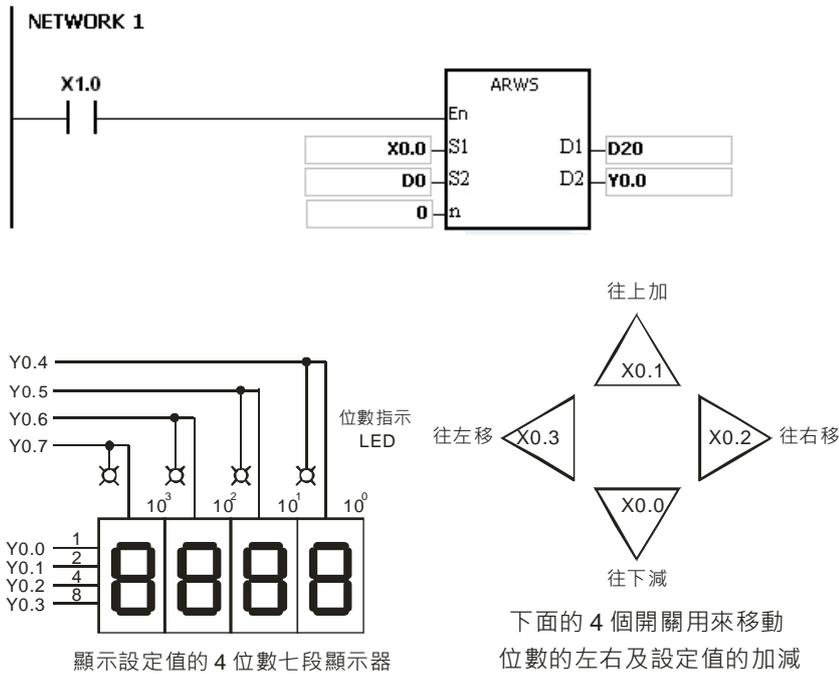
- S_1 : 按鍵輸入起始裝置 Bit
- S_2 : 系統內部使用 Word
- D_1 : 欲顯示於七段顯示器之裝置 Word
- D_2 : 七段顯示器掃描輸出起始裝置 Bit
- n : 輸出信號及掃描信號之極性指示 Word

指令說明：

- 本指令執行， S_1 定義為下鍵， S_1+1 定義為上鍵， S_1+2 定義為右鍵， S_1+3 定義為左鍵，利用上下左右鍵來執行外部設定值的操作及顯示。將設定值存放於 D_1 當中，設定值範圍：0~9,999。
- S_1 連續佔用 4 個 bit 裝置。
- S_2 系統內部使用，使用者請勿變更內容值。
- D_2 連續佔用 8 個 bit 裝置。
- 條件接點未啟點時， D_2 連續的 8 個 bit 裝置保持 OFF。
- n 的範圍：0~3，功能請參考 API1704 SEGL 的補充說明。
- 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。

程式範例：

- 本指令執行，X0.0 定義為下鍵，X0.1 定義為上鍵，X0.2 定義為右鍵，X0.3 定義為左鍵，利用上下左右鍵來執行外部設定值的操作及顯示。將設定值存放於 $D20$ 當中，設定值範圍：0~9,999。
- 當 X1.0=ON 時，位數 10^3 為有效設定位數，如果按左按鍵時，則有效設定位數呈現 $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$ 的方向循環跳動。
- 如果按右移按鍵，則有效設定位數呈現 $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3 \rightarrow 10^2$ 的方向循環跳動。在循環的同時，由 Y0.4~Y0.7 所連接的位數指示燈亦循環 ON 作有效設定位數的指示。
- 如果按往上加按鍵時，則有效設定位數的內容由 0→1→2→...8→9→0→1 作變化。如果按往下減按鍵時，則有效設定位數的內容由 0→9→8→...1→0→9 作變化，同時，變化值亦被顯示在七段顯示器上。



補充說明：

1. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B。
2. **S₁** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of BOOL。
3. **D₂** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [8] of BOOL。

API	指令碼		運算元										功能				
1704		SEGL	$S_1 \cdot S_2 \cdot D \cdot n$										7 段顯示器掃描輸出				

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：



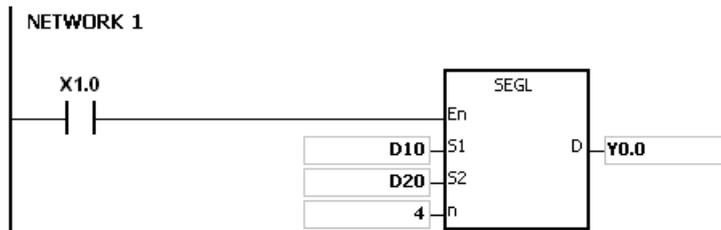
- S_1 : 欲顯示於七段顯示器之來源裝置 Word
- S_2 : 系統內部使用 Word
- D : 七段顯示器掃描輸出起始裝置 Bit
- n : 輸出信號及掃描信號之正負邏輯設定 Word

指令說明：

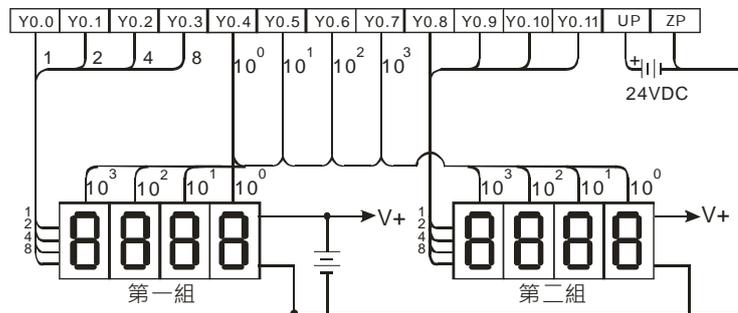
1. 本指令佔用 D 開始的連續 8 個或 12 個外部輸出點，作為 1 組或 2 組 4 位數七段顯示器的顯示資料及掃描信號輸出。每個位數均帶有 7 段顯示器驅動器，該驅動器是將輸入的 BCD 碼轉換 7 段顯示器的驅動信號；驅動器並帶有栓鎖控制信號，可將 7 段顯示器顯示保持。
2. S_1 中之數值為欲顯示到第一組七段顯示器之數值， S_{1+1} 中之數值為欲顯示到第二組七段顯示器之數值。
3. S_2 系統內部使用，使用者請勿變更內容值。
4. $n=0\sim7$ 。請參考補充說明。
5. 由 n 決定掃描輸出 4 位數七段顯示器有 1 組或 2 組，且 n 也用來指定 PLC 輸出端的正負邏輯輸出。
6. 4 位數 1 組時，佔用輸出點 8 個，4 位數 2 組時，佔用輸出點 12 個。
7. 本指令執行時，掃描輸出端順序循環動作，指令執行中條件接點變成 OFF 再 ON 時，掃描輸出端重新執行。
8. 每循環掃描一次完畢旗標信號 SM693=ON 一個掃描週期。

程式範例：

1. 當 X1.0=ON 時，指令開始執行，由 Y0.0~Y0.7 構成七段顯示器掃描迴路，D10 中之數值被轉換成 BCD 碼後送到第一組七段顯示器顯示出來，D11 中之數值被轉換成 BCD 碼後送到第二組七段顯示器顯示出來，若 D10 或 D11 中之數值超過 9,999 將發生運算錯誤。



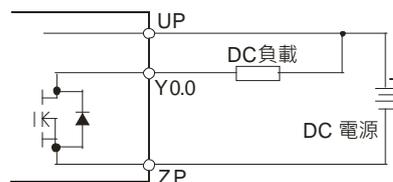
2. 當 X1.0=ON 時，Y0.4~Y0.7 會自動循環掃描，每循環掃描一次需 12 個掃描時間，每循環掃描一次完畢旗標信號 SM693=ON 一個掃描週期。
3. 一組 4 位數的時候 n=0~3。
 - 將已經解碼的 7 段顯示模組 1、2、4、8 各端各自並接後連接至 PLC 的 Y0.0~Y0.3，而各位數的 Latch 端單獨連接至 PLC 的 Y0.4~Y0.7。
 - 當 X1.0=ON 時，指令被執行，D10 的內容隨著 Y0.4~Y0.7 的循環動作被順序傳送至七段顯示器作顯示。
4. 二組 4 位數的時候 n=4~7。
 - 將已經解碼的七段顯示器 1、2、4、8 各端各自並接後連接至 PLC 的 Y0.8~Y0.11，而各位數的 Latch 端與第一組共用 Y0.4~Y0.7。
 - D10 的內容被傳送到第一組七段顯示器上、D11 的內容被傳送到第二組七段顯示器作顯示。若 D10=1234，D11=4321，則第一組將會顯示 1 2 3 4，第二組顯示 4 3 2 1。
5. 七段顯示器掃描輸出接線圖：



註: 此範例使用AH16AN01T電晶體輸出模組

補充說明：

1. n 的設定值：是用來設定電晶體輸出為正極性或負極性回路，連接的七段顯示器是一組 4 位數或者是二組 4 位數。
2. PLC 輸出點必須選用為電晶體模組，輸出為 NPN 型式，採開集極式輸出，在電路的連接上，輸出必須連接一提升電阻至 DC 電源 (小於 30VDC)，因此當輸出點 Y 導通時，信號輸出為低電位。



- BCD 碼正邏輯 (負極性) 輸出

BCD 數值				Y 輸出 (BCD 碼)				信號輸出			
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 碼負邏輯 (正極性) 輸出

BCD 數值				Y 輸出 (BCD 碼)				信號輸出			
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

- 顯示掃描栓鎖 (Latch) 信號

正邏輯 (負極性)		負邏輯 (正極性)	
Y 輸出 (Latch)	輸出控制信號	Y 輸出 (Latch)	輸出控制信號
1	0	0	1

- 參數 n 的設定值

7 段顯示器組數	一組				二組			
BCD 碼資料 Y 輸出	+		-		+		-	
顯示掃描栓鎖信號	+	-	+	-	+	-	+	-
n	0	1	2	3	4	5	6	7

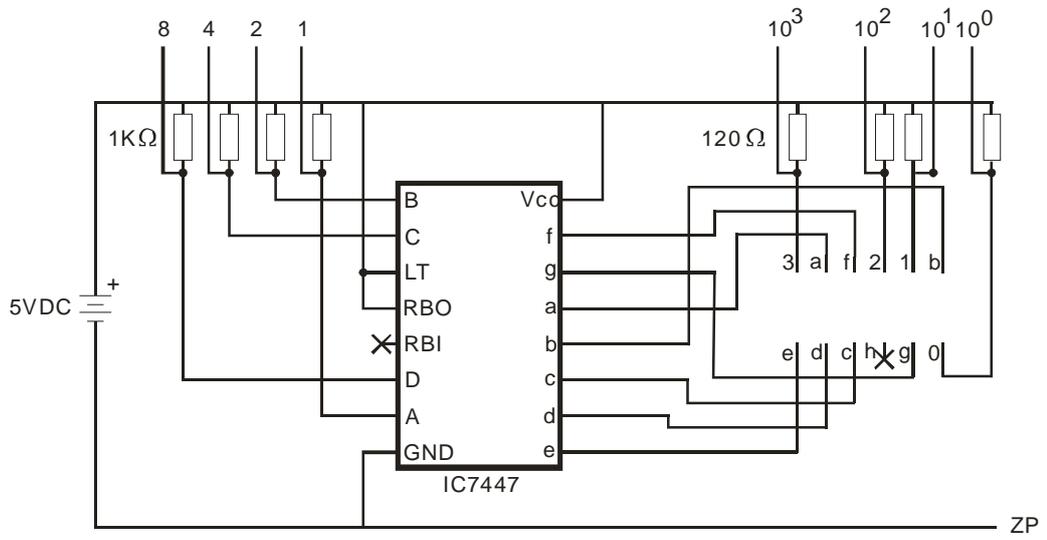
‘+’: 正邏輯 (負極性) 輸出

‘-’: 反邏輯 (正極性) 輸出

- PLC 的電晶體輸出極性與 7 段顯示器的輸入極性是否相同或者是不同時，可透過可參數 n 的設定值來相互匹配。

6

- 採用共陽極四位數 7 段顯示器搭配 IC7447 配置如下圖



6.19 通訊指令

6.19.1 通訊指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1800</u>	RS	—	—	串列資料傳輸	9
<u>1801</u>	FWD	—	—	交流馬達驅動器正轉指令	7
<u>1802</u>	REV	—	—	交流馬達驅動器反轉指令	7
<u>1803</u>	STOP	—	—	交流馬達驅動器停止指令	3
<u>1804</u>	RDST	—	—	交流馬達驅動器狀態讀取	5
<u>1805</u>	RSTEF	—	—	交流馬達驅動器異常重置	3
<u>1806</u>	LRC	—	✓	和檢查 LRC 模式	7
<u>1807</u>	CRC	—	✓	和檢查 CRC 模式	7
<u>1808</u>	MODRW	—	—	MODBUS 資料讀寫	11
<u>1812</u>	COMRS	—	—	通訊傳送與接收指令	11

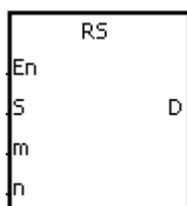
6.19.2 通訊指令說明

API	指令碼		運算元				功能						
1800		RS	S · m · D · n				傳送使用者自行定義的通訊命令						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
-	AH500	-

符號：



S :	傳送資料的起始裝置	Word
m :	傳送資料的筆數	Word
D :	接收資料的起始裝置	Word
n :	接收資料的筆數	Word

指令說明：

- 此指令專為主機使用串聯式通訊介面 (RS-232/422/485) 所提供的便利指令，使用者只要將其相關參數 (S、m、D、n) 設定完成，即可發送與接收命令。若使用 E 修飾裝置起始位置時，請勿在指令執行期間變更 E 的設定值，否則容易造成資料讀取或寫入錯誤。
- 若僅發送或接收資料，可將裝置長度 (m、n) 設定為 0 即可。
- 本指令於程式中使用次數並無限制，但是不同指令 (如 MODRW、FWD...) 在同一時間使用相同的通訊埠時，僅會只有一個指令被執行。
- RS 指令傳送或接收未完成時，無法改變傳送或接收裝置資料內容。
- 傳送或接收裝置最大長度 (m、n) 為 500 words。
- 本指令可以藉由設定特殊暫存器來選擇 8 位元或 16 位元模式 (SM106 或 SM107)。
- 若周邊設備之通訊格式符合 MODBUS 之通訊格式，PLC 提供通訊便利指令 MODRW 供使用者使用。詳細使用說明請參考個別指令之說明。

通訊能力設定

在執行串列通訊命令之前，使用者必須先行設定通訊能力 (如 RS232/485、傳輸速率...)，設定通訊能力的方式有兩種，使用者可以在 HWCONFIG 內針對 PLC 通訊埠直接進行設定，或於程式內針對其相對應的特殊輔助繼電器來進行設定。

- 利用 HWCONFIG 來針對通訊埠進行通訊能力設定 (請參照 ISPSOFT 使用手冊)
- 於程式內針對各個特殊暫存器，來進行通訊埠通訊能力設定，其設定方式如範例，暫存器內容格式請參照本章節補充說明。

傳輸資料格式說明

資料的傳送格式分成幾個部分 8 位元與 16 位元模式。8 位元模式是將 16 位元資料分成上 8 位元、下 8 位元，上 8 位元被省略，僅下 8 位元為有效資料可做資料的傳送和接收。如模式為 16 位元時要特別注意高低位元的部分，請參照下表。

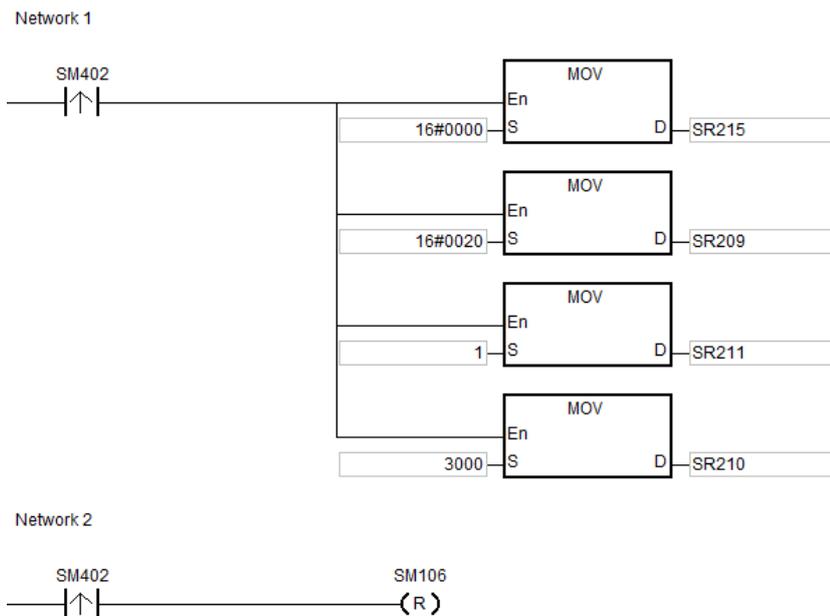
8 位元模式 · 傳送資料 0x01234567							
D10 (上)	D10 (下)	D11 (上)	D11 (下)	D12 (上)	D12 (下)	D13 (上)	D13 (下)
*	16#01	*	16#23	*	16#45	*	16#67
16 位元模式 · 傳送資料 0x1234567							
D10		D11		D12		D13	
16#2301		16#6745					

通訊協定設定範例：

1. 本範例為將通訊埠的通訊協定設定為 RS232 9600 · 7 · N · 1。
2. 設定通訊埠傳輸方式為 RS232 (SR215=0)。
3. 設定通訊埠其通訊速率與格式為 9600 · 7 · N · 1 (SR209=16#0020)。
4. 設定其自動詢問次數為 1 (SR211=1)。
5. 設定其逾時時間為 3000ms (SR210=3000)。
6. 設定為 16 位元模式 (SM106=OFF)。
7. 通訊埠通訊能力設定生效 (SM209=ON)。

使用者若於 ISPSOft - > HWCONFIG - > COM Port 內直接對通訊埠進行設定，則可以省略此步驟。

6

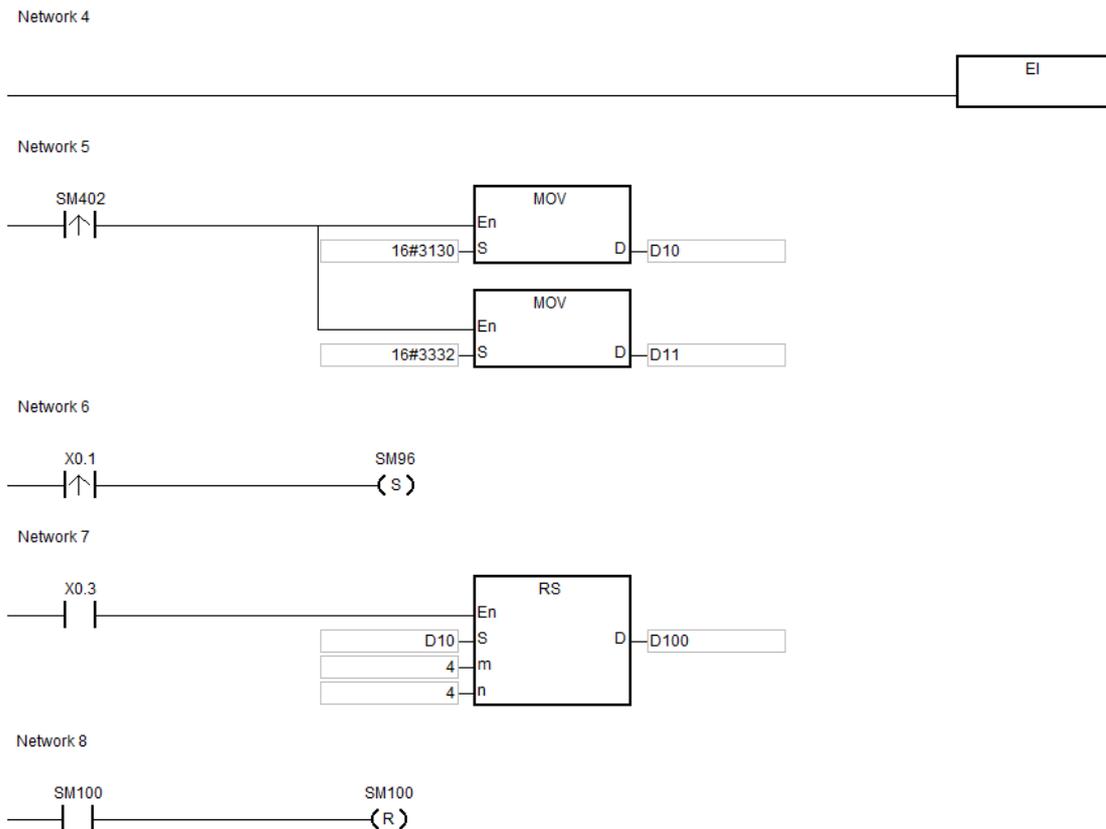




程式範例：

1. 利用 HWCONFIG 或是特殊輔助暫存器，來設定通訊埠通訊能力 (如設定範例所述)。
2. 先將傳送資料內容預先寫入 D10 與 D11 暫存器內，再將 SM96 (送信要求旗標) 設為 ON。
3. 當 X0.1 與 X0.3=ON 時，RS 指令執行 PLC 即進入等待傳送、接收資料的狀態，開始執行後，D10 開始連續傳送 n 個 bytes 到外部機器，在傳送結束時，SM96 會自動 RESET 成 OFF (請勿利用程式執行 RST SM96)，當有資料回覆回來的時候，將其存入由 D100 開始之連續暫存器內。
4. 當資料接收完畢旗標 (SM100) 自動 ON，程式中處理完接收資料後，須將 SM100 RESET 為 OFF，再度進入等待傳送接收的狀態。但請勿利用 PLC 程式連續執行 RST SM100。
5. 當 PLC 接收到的回覆命令中，含有特殊字元 (SR621) 時，將會觸發 I32，使得 D30 加一。

範例程式



範例中斷程式 (I32)



補充說明：

1. **M** 或 **n** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#200B。
2. COM1 與 COM2 通訊 RS / MODRW 指令相關旗標信號：

6

旗標信號		功能說明	動作
COM1	COM2		
SM96	SM97	送信要求，當使用者要利用 RS/MODRW 指令將資料傳送與接收，必須用脈波指令將 SM96/SM97 設為 ON，若上述指令開始執行，則 PLC 執行資料傳送接收的動作。當上述指令執行資料傳送完畢後會自動將 SM96/SM97 清除。	使用者設定，系統自動清除
SM98	SM99	接收等待，當 SM98、SM99 為 ON 時，表示 PLC 目前正等待接收資料中。	系統產生以及系統自動清除
SM100	SM101	接收完畢，當 RS/MODRW 指令接收完畢後會將 SM100/SM101 設為 ON，使用者在程式中可利用 SM100 為 ON 時，處理所接收到的資料。當接收到的資料處理完畢後，必須將 SM100、SM101 清除為 OFF。	系統自動設定，使用者清除
SM102	SM103	MODRW、RS...等資料接收錯誤，錯誤代碼紀錄在 Error Log 中。	系統產生以及系統自動清除
SM104	SM105	接收逾時，使用者若有設定通訊逾時(SR210*SR211 或 SR213*SR214 的時間為逾時時間)，若超出設定值資料尚未接收完畢則會啟動此旗標。若狀態解除後必須將 SM104/SM105 清除為 OFF。	系統自動設定，使用者清除
SM106	SM107	8/16 位元處理模式選擇，ON 為 8 位元模式，OFF 為 16 位元模式。	使用者設定及清除
SM108	SM109	接收結束旗標。 ON：當 PLC 在接收的字元時，會去判斷是否有通訊中斷字元 (SR621/SR622 low byte) 存在，若有檢查到中斷字元時，執行通訊中斷副程式，執行完後結束接收資料。 OFF：當 PLC 在接收的字元時，會去判斷是否有通訊中斷字元 (SR621/SR622 low byte) 存在，若有檢查到中斷字元時，執行通訊中斷副程式，執行完後繼續接收資料，直到完成接收 (接收資料長度為 n)。(預設值)。	使用者設定及清除

旗標信號		功能說明	動作
COM1	COM2		
SM209	SM211	通訊協定變更改用，根據特殊資料暫存器 SR201、SR202、SR209、SR210、SR211、SR212、SR213、SR214、SR215、SR216、SM210、SM212 的設定，作為通訊協定設定的重置。可將 SM209/SM211 設為 ON，此時，COM1 或 COM2 的通訊協定即依 SR201、SR202、SR209、SR210、SR211、SR212、SR213、SR214、SR215、SR216、SM210、SM212 的設定改變，且 SM209/SM211 會自動設定成 OFF。	使用者設定，系統自動清除

3. COM1 或 COM2 通訊 RS/MODRW 指令相關設定之特殊資料暫存器：

特 D		功能說明
COM1	COM2	
SR201	SR202	PLC 主機通訊位址，當 PLC 主機當從站時的通訊位址。
SR210	SR213	通訊逾時異常，時間定義 (ms)，設定 Timeout 時間。當設定值大於 0 時，則 RS/MODRW 指令執行，進入接收模式後，若在指定的時間內沒有收到第一個字元或任二字元之間的時間超過此設定值，PLC 會將 SM104/SM105 設為 ON，使用者可利用此旗標作通訊逾時的處理。但必須記得：處理完後，必須將 SM104/SM105 清除，此暫存器在 RS 指令可被設為 0，此時逾時偵測功能被取消，但在 MODRW 指令則為限定在 100~65535 (ms) 區間內。
SR621	X	RS 指令特定字元通訊接收中斷請求 (I32)，當通訊接收的字元=SR621 的 Low Byte 時，觸發中斷 I32。當 n=0 時，中斷不反應。
X	SR622	RS 指令特定長度通訊接收中斷請求 (I33)，當通訊接收的字元=SR622 的 Low Byte 時，觸發中斷 I33。當 n=0 時，中斷不反應。

6

4. SR215、SR216 用來紀錄 PLC COM 通訊介面的代號，代號代表的介面功能如下表：

代號	0	1	2
介面功能	RS232	RS485	RS422

5. SR209、SR212：RS-485/RS-232 通信協定，其設定方法請參考下表：

b0	資料長度		7 (內容值=0)	8 (內容值=1)
b1 b2	同位元		00	無 (None)
			01	奇同位 (Odd)
			10	偶同位 (Even)
b3	stop bits		1 bit (內容值=0)	2 bits (內容值=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	無定義 (保留)				

6. PLC 相關通訊傳送速率如下：

鮑率 (bps)	RTU 逾時時間間隔 (ms)	鮑率 (bps)	RTU 逾時時間間隔 (ms)
4800	9	115200	1
9600	5	230400	1
19200	3	460800	1
38400	2	921600	1
57600	1		

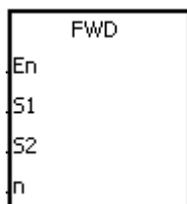
6

API	指令碼			運算元						功能							
1801		FWD		$S_1 \cdot S_2 \cdot n$						交流馬達驅動器正轉							

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：



S_1 : 連線裝置位址 Word

S_2 : 交流馬達驅動器運轉頻率 Word

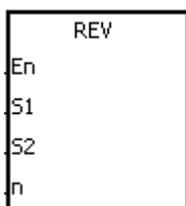
n : 模式選擇 Word

API	指令碼		運算元							功能						
1802		REV	$S_1 \cdot S_2 \cdot n$							交流馬達驅動器反轉						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：



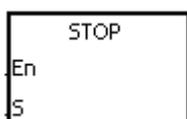
- S_1 : 連線裝置位址 Word
- S_2 : 交流馬達驅動器運轉頻率 Word
- n : 模式選擇 Word

API	指令碼	運算元	功能
1803	STOP	S ₁	交流馬達驅動器停止

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

脈波執行型	16 位元指令 (3 steps)	32 位元指令
-	AH500	-

符號：



S₁：連線裝置位址 Word

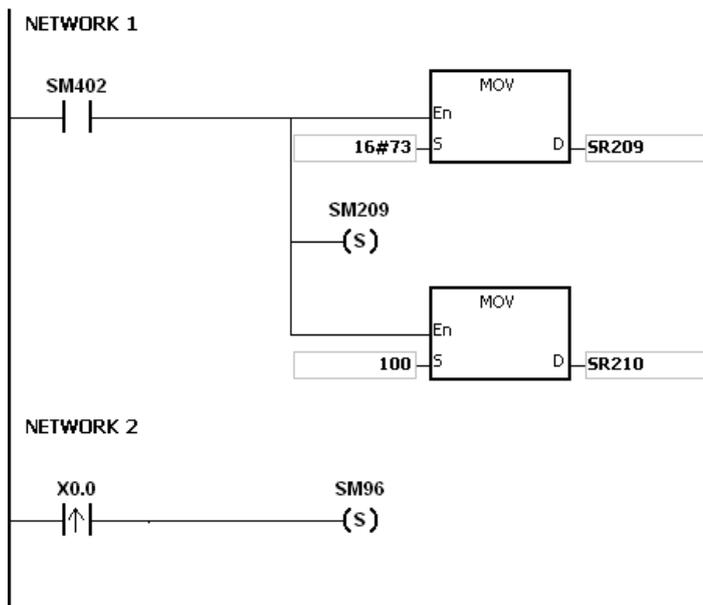
指令說明：

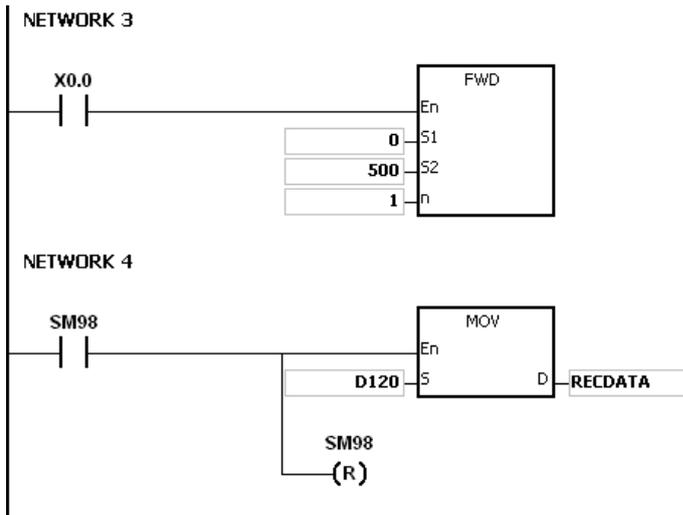
1. FWD/REV/STOP 為台達交流馬達驅動器 VFD 系列專用的通訊便利指令，對交流馬達驅動器下達正轉/反轉/停止的指令。此指令在應用時，必須配合通訊逾時設定 (SR210、SR212)。
2. S₁：範圍 0~255，當站號為 0 時表示對所有交流馬達驅動器進行廣播通訊。
3. S₂ 頻率值之設定值與數值單位，請參考交流馬達驅動器使用手冊，但在 STOP 指令為無此接腳。
4. n 模式選擇依指令說明如下：
 FWD 指令：正轉模式 n=0：一般正轉模式，n=1：JOG 正轉模式，其餘數值皆無法支援。
 REV 指令：反轉模式 n=0：一般反轉模式，n=1：JOG 反轉模式，其餘數值皆無法支援。
 STOP 指令：無。
5. 當正轉模式為 JOG 正轉模式時，其 S₂ 的頻率值將會無效，如須修改 JOG 頻率請參考交流馬達驅動器使用手冊。

6

程式範例：

1. PLC 與 VFD 系列交流馬達驅動器連線，通訊逾時及接收資料錯誤之重傳資料。





PLC⇒VFD · PLC 傳送：“ : 01 10 2000 0002 04 0012 01F4 C2 CR LF”

VFD⇒PLC · PLC 接收：“ : 01 10 2000 0002 CD CR LF”

PLC 傳送資料 (傳送訊息)

Data		說明		
'0'	16#30	ADR 1	ADR (10) 為交流馬達驅動器位址	
'1'	16#31	ADR 0		
'1'	16#31	CMD 1	CMD (10) 為功能碼	
'0'	16#30	CMD 0		
'2'	16#32	資料位址 Data Address		
'0'	16#30			
'0'	16#30			
'0'	16#30			
'0'	16#30	資料內容 Data contents		
'0'	16#30			
'0'	16#30			
'2'	16#32			
'0'	16#30	Byte 數目		
'4'	16#34			
'0'	16#30			資料內容 1 16#12 為正轉啟動
'0'	16#30			
'1'	16#31			
'2'	16#32	資料內容 2 運轉頻率=500Hz 16#01F4		
'0'	16#30			
'1'	16#31			
'F'	16#46			
'4'	16#34	LRC CHK (01) 為錯誤檢查碼		
'C'	16#43			LRC CHK 1
'2'	16#32	LRC CHK 0		

6

PLC 接收資料 (回應訊息)

Data		說明	
'0'	16#30	ADR 1	ADR (10) 為交流馬達驅動器位址
'1'	16#31	ADR 0	
'1'	16#31	CMD 1	CMD (10) 為功能碼
'0'	16#30	CMD 0	
'2'	16#32	資料位址 Data Address	
'0'	16#30		
'0'	16#30		
'0'	16#30		
'0'	16#30	暫存器數目 Number of Register	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'C'	16#43	LRC CHK 1	
'D'	16#44	LRC CHK 0	

補充說明：

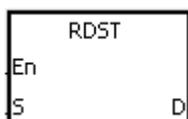
1. 相關旗標信號與相關設定之特殊暫存器請參考 API1800 RS 指令補充說明。
2. FWD、REV、STOP、RDST、RSTEF 指令於程式中使用次數並無限制，但是同時間僅有一個指令被執行。
3. **S₁** 內容值超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003
4. **n** 超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#200B
5. FWD、REV、STOP、RDST、RSTEF 指令，符合 MODBUS 通訊格式。

API	指令碼		運算元							功能						
1804		RDST	S · D							交流馬達驅動器狀態讀取						

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

脈波執行型	16 位元指令 (5 steps)	32 位元指令
-	AH500	-

符號：



S：連線裝置位址 Word

D：儲存裝置起始位址 Word

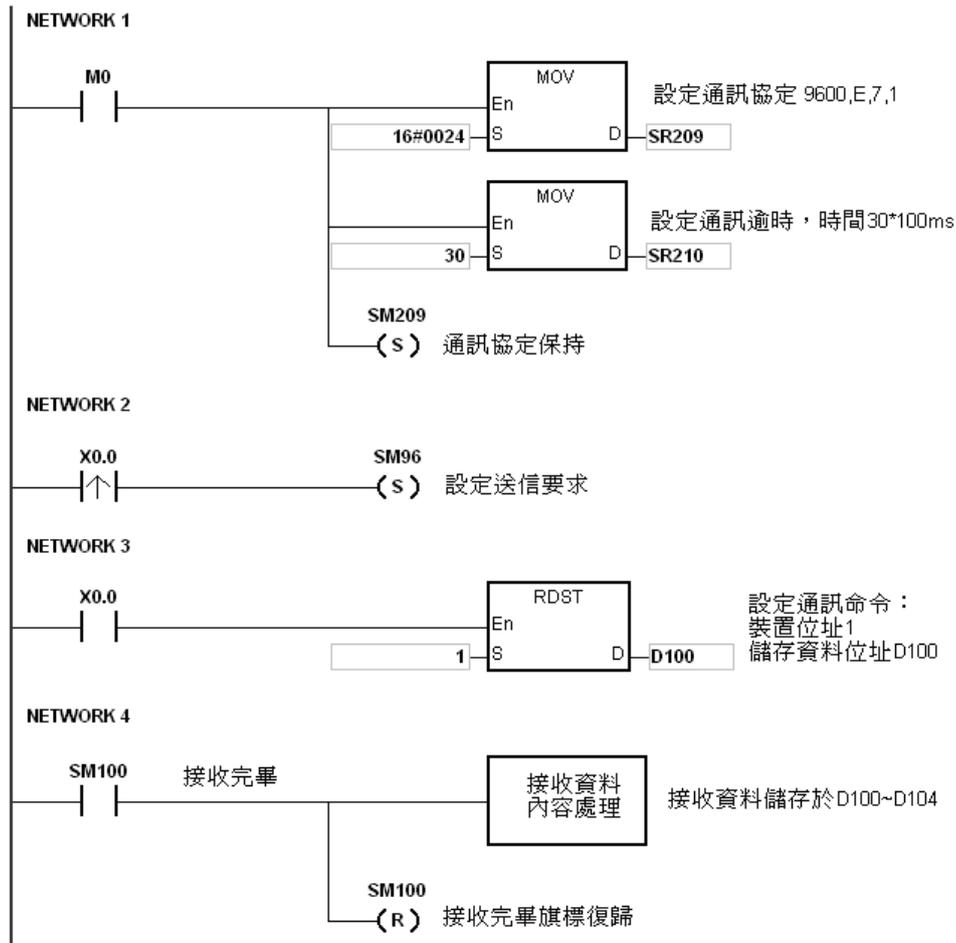
指令說明：

1. **S**：範圍 1~255。不能使用 0。
2. **D**：將交流馬達驅動器回覆的資料，儲存在使用者指定的裝置位址裡。
3. 此指令將會讀取交流馬達驅動器參數位址 16#2100~16#2104 之狀態(詳細狀態說明請參考交流馬達驅動器使用手冊)，並於接收完成後儲存於 **D** 參數所指定的裝置位址裡；不過當讀取通訊接收訊息錯誤或接收逾時發生時，此 **D** 內容將不會被更改，因此建議要判斷交流馬達驅動器狀態訊息前，請先確認接收完成旗標是否已有被設定。
4. 裝置起始位址會佔用到 5 個裝置，例如：**D_n**、**D_{n+1}**、**D_{n+2}**、**D_{n+3}**、**D_{n+4}**。所以裝置起始位址須要保留 5 個裝置的位址。

程式範例：

1. PLC 與 VFD 系列交流馬達驅動器連線 (ASCII Mode · SM210 · SM212=OFF)，當通訊逾時之重傳資料。
2. 讀取交流馬達驅動器參數位址 16#2100~16#2104 之狀態，儲存於 D100~D104 之中。

6



PLC⇒VFD PLC 傳送：“ : 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 傳送資料 (傳送訊息)

Data		說明	
'0'	16#30	ADR 1	ADR (10) 為交流馬達驅動器位址
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	CMD (10) 為命令碼
'3'	16#33	CMD 0	
2'	16#32	起始資料位址 Starting Data Address	
'1'	16#31		
'0'	16#30		
'0'	16#30		
'0'	16#30	資料 (word) 個數 Number of Data (count by word)	
'0'	16#30		
'5'	16#35		
'D'	16#44	LRC CHK 1	LRC CHK (01) 為錯誤檢查碼 code
'6'	16#36	LRC CHK 0	

PLC 接收資料暫存器 (回應訊息)

Data		說明	
'0'	16#30	ADR 1	
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	
'3'	16#33	CMD 0	
'0'	16#30	資料 (BYTE) 個數	
'A'	16#41	Number of Data (count by Byte)	
'0'	16#30	位址 16#2100 的內容	PLC 自動將 ASCII 字元轉換為數值儲存於 D100=16#0000
'0'	16#30		
'0'	16#30		
'0'	16#30		
'0'	16#30	位址 16#2101 的內容	PLC 自動將 ASCII 字元轉換為數值儲存於 D101=16#0500
'5'	16#35		
'0'	16#30		
'0'	16#30		
'0'	16#33	位址 16#2102 的內容	PLC 自動將 ASCII 字元轉換為數值儲存於 D1072=16#01F4
'1'	16#45		
'F'	16#30		
'4'	16#30		
'0'	16#30	位址 16#2103 的內容	PLC 自動將 ASCII 字元轉換為數值儲存於 D1073=16#0000
'0'	16#30		
'0'	16#30		
'0'	16#30		
'0'	16#30	位址 16#2104 的內容	PLC 自動將 ASCII 字元轉換為數值儲存於 D1074=16#0000
'0'	16#30		
'0'	16#30		
'0'	16#30		
'2'	16#32	LRC CHK 1	
'A'	16#41	LRC CHK 0	

補充說明：

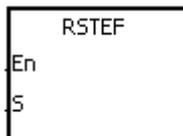
1. 相關旗標信號與相關設定之特殊暫存器請參考 API1800 RS 指令補充說明。
2. FWD、REV、STOP、RDST、RSTEF 指令於程式中使用次數並無限制，但是同時間僅有一個指令被執行。
3. **S** 內容值超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. **D+4** 超過裝置範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
5. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [5] of WORD/INT。

API	指令碼	運算元	功能
1805	RSTEF	S	交流馬達驅動器異常重置

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

脈波執行型	16 位元指令 (3 steps)	32 位元指令
-	AH500	-

符號：



S：連線裝置位址

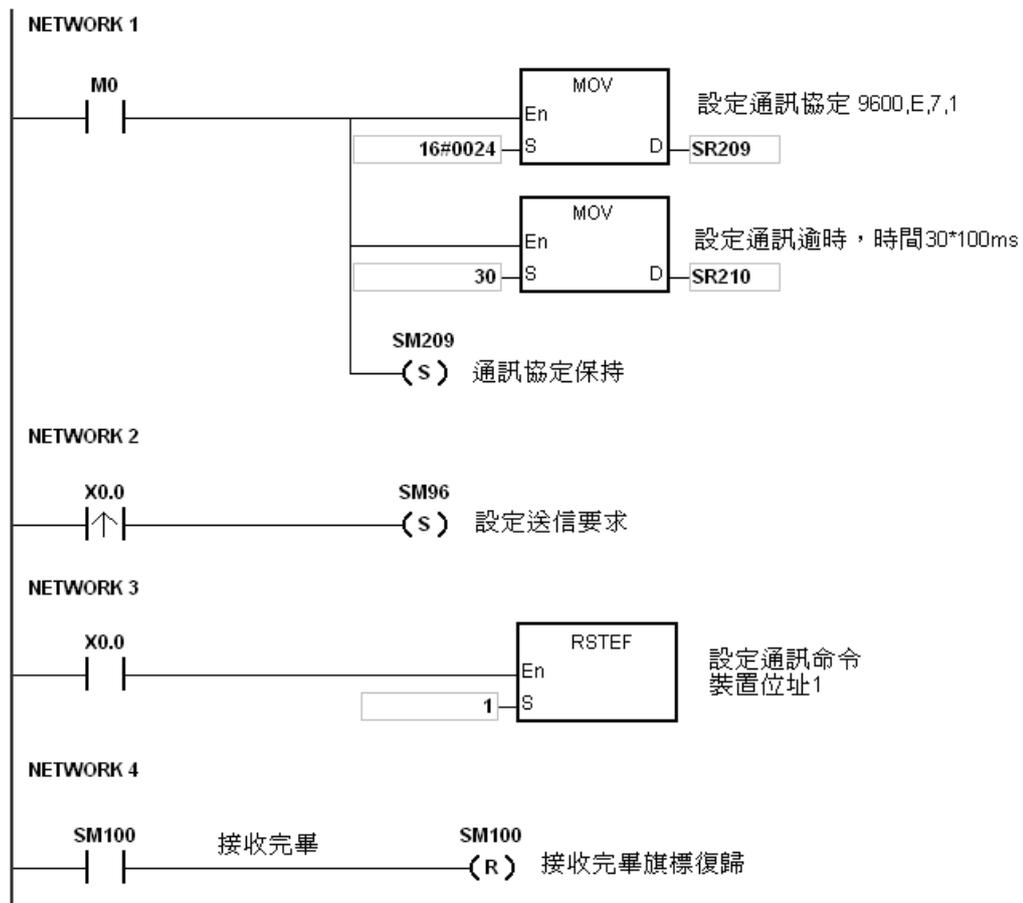
Word

指令說明：

1. S：範圍 0~255，當站號為 0 時表示對所有交流馬達驅動器進行廣播通訊。

程式範例：COM2 (RS-485)

PLC 與 VFD 系列交流馬達驅動器連線 (ASCII 模式，SM210、SM212=OFF)，當通訊逾時之重傳資料。



PLC⇒VFD · PLC 傳送：“ 01 06 2002 0002 D5 CR LF” (ASCII)

VFD⇒PLC · PLC 接收：“ : 01 06 2002 0002 D5 CR LF” (ASCII)

PLC 傳送資料 (傳送訊息)

Data		說明	
'0'	16#30	ADR 1	ADR (10) 為交流馬達驅動器位址
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	CMD (10) 為命令碼
'6'	16#36	CMD 0	
'2'	16#32	資料位址 Data Address	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'0'	16#30	資料內容 Data contents	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'D'	16#44	LRC CHK 1	LRC CHK (01) 為錯誤檢查碼
'5'	16#35	LRC CHK 0	

PLC 接收資料 (回應訊息)

Data		說明	
'0'	16#30	ADR 1	
'1'	16#31	ADR 0	
'0'	16#30	CMD 1	
'6'	16#36	CMD 0	
'2'	16#32	資料位址 Data Address	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'0'	16#30	暫存器數目 Number of Register	
'0'	16#30		
'0'	16#30		
'2'	16#32		
'D'	16#44	LRC CHK 1	
'5'	16#35	LRC CHK 0	

補充說明：

1. 相關旗標信號與相關設定之特殊暫存器請參考 API1800 RS 指令補充說明。
2. FWD、REV、STOP、RDST、RSTEF 指令於程式中使用次數並無限制，但是同時間僅有一個指令被執行。
3. **S** 內容值超出範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

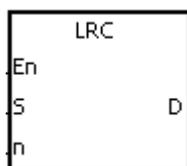
6

API	指令碼		運算元							功能						
1806		LRC	S · n · D							和檢查 LRC 模式						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：



S：ASCII 模式和檢查運算起始裝置 Word

n：運算組數 Word

D：存放運算結果之起始裝置 Word

指令說明：

1. LRC 檢查碼：請參考補充說明。
2. **n**：運算組數須為偶數，範圍 1~1000 不在此範圍則視為運算錯誤，指令不執行。SM0、SM1=ON，SR0 記錄錯誤碼 16#200B。
3. 16 位元轉換模式：當 SM606=OFF 時，S 起始裝置將其 16 進位資料區分為上 8 位元、下 8 位元，將每一個位數做 LRC 檢查碼運算，傳送到 D 的上 8 位元及下 8 位元中，運算的位數以 n 來設定。
4. 8 位元轉換模式：當 SM606=ON 時，S 起始裝置將其 16 進位資料區分為上 8 位元(無效資料)、下 8 位元，將每一個位數做 LRC 檢查碼運算，傳送到 D 的下 8 位元中佔用 2 個暫存器，運算的位數以 n 來設定。(D 的上 8 位元全部為 0)

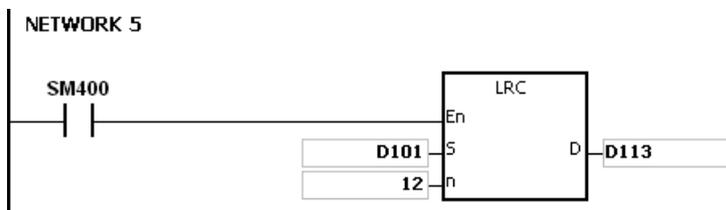
程式範例：

1. PLC 與 VFD-S 系列交流馬達驅動器連線 (ASCII 模式，SM210 =OFF)、(8 位元模式，SM606=ON)，傳送資料預先寫入讀取 VFD-S 參數位址 16#2101 開始之 6 筆資料。
 PLC⇒VFD-S，PLC 傳送：“：01 03 2101 0006 D4 CR LF”
 PLC 傳送資料暫存器 (PLC 傳送訊息)

暫存器	DATA		說明	
D100 下	‘:’	16#3A	STX	
D101 下	‘0’	16#30	ADR 1	ADR (10) 為交流馬達驅動器位址
D102 下	‘1’	16#31	ADR 0	
D103 下	‘0’	16#30	CMD 1	CMD (10) 為命令碼
D104 下	‘3’	16#33	CMD 0	
D105 下	‘2’	16#32	起始資料位址 Starting Data Address	
D106 下	‘1’	16#31		
D107 下	‘0’	16#30		
D108 下	‘1’	16#31		
D109 下	‘0’	16#30	資料 (word) 個數 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) 為錯誤檢查碼
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) 為錯誤檢查碼可由指令 LRC 算出 (8 位元 Mode · SM606=ON)

6



LRC 檢查碼：16#01+16#03+16#21+16#01+16#00+16#06=16#2C，然後取 2 的補數=16#D4。
 此時，‘D’ (16#44) 存於 D113 下 8 位元內，‘4’ (16#34) 存於 D114 下 8 位元內。

補充說明：

1. 有一通訊資料之 ASCII 模式，格式如下：

STX	‘：’	起始字元=‘：’ (16#3A)
Address Hi	‘0’	通信位址：
Address Lo	‘1’	8-bit 位址由 2 個 ASCII 碼組合
Function Hi	‘0’	功能碼：
Function Lo	‘3’	8-bit 功能碼由 2 個 ASCII 碼組合
DATA (n-1) DATA 0	‘2’	資料內容：
	‘1’	n×8-bit 資料內容由 2n 個 ASCII 碼組合
	‘0’	
	‘2’	
	‘0’	
	‘0’	
	‘0’	
‘2’		
LRC CHK Hi	‘D’	LRC 檢查碼：
LRC CHK Lo	‘7’	8-bit 檢查碼由 2 個 ASCII 碼組合
END Hi	CR	結束字元：
END Lo	LF	END Hi=CR (16#0D) · END Lo=LF (16#0A)

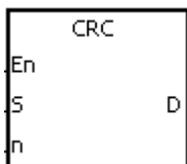
2. LRC 檢查碼：由通信位址到資料內容結束加起來的值取 2 的補數即為檢查碼 (LRC Check)。例如： $16\#01+16\#03+16\#21+16\#02+16\#00+16\#02=16\#29$ ，然後取 2 的補數 =16#D7。

API	指令碼		運算元							功能								
1807		CRC																檢查 CRC 模式

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：



S：RTU 模式和檢查運算起始裝置

n：運算組數

D：存放運算結果之起始裝置。

指令說明：

- CRC 檢查碼：請參考補充說明
- n**：範圍 1~1000 不在此範圍則視為運算錯誤，指令不執行，SM0、SM1=ON，SR0 記錄錯誤碼 16#200B。
- 16 位元轉換模式：當 SM606=OFF 時，S 起始裝置其資料區分為上 8 位元、下 8 位元，將每一個位數做 CRC 檢查碼運算，傳送到 D 的上 8 位元及下 8 位元中，運算的位數以 n 來設定。
- 8 位元轉換模式：當 SM606=ON 時，S 起始裝置其資料區分為上 8 位元 (無效資料)、下 8 位元，將每一個位數做 CRC 檢查碼運算，傳送到 D 的下 8 位元中佔用 2 個暫存器，運算的位數以 n 來設定。(D 的上 8 位元全部為 0)

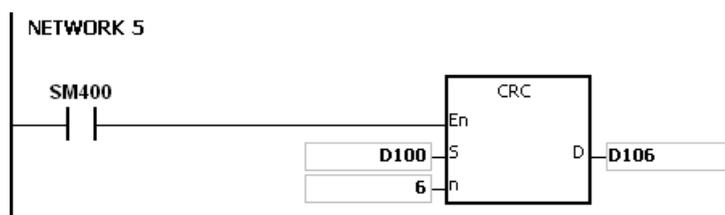
6

程式範例：

- PLC 與 VFD-S 系列交流馬達驅動器連線 (RTU 模式，SM210=ON)、(16 位元模式，SM606=ON)，傳送資料預先寫入欲寫入 VFD-S 參數位址 16#2000 寫入內容為 16#12。
 PLC VFD-S · PLC 傳送：01 06 2000 0012 02 07
 PLC 傳送資料暫存器 (PLC 傳送訊息)

暫存器	DATA	說明
D100 下	16#01	Address
D101 下	16#06	Function
D102 下	16#20	資料位址
D103 下	16#00	Data Address
D104 下	16#00	資料內容
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) 為錯誤檢查碼可由指令 CRC 算出 (8 位元 Mode，SM606=ON)



CRC 檢查碼：此時，16#02 存於 D106 下 8 位元內，16#07 存於 D107 下 8 位元內。

補充說明：

1. 有一通訊資料之 RTU 模式，格式如下：

START	時間間隔
Address	通信位址：8-bit 二進制位址
Function	功能碼：8-bit 二進制
DATA (n-1)	資料內容：
.....	n×8-bit 資料
DATA 0	
CRC CHK Low	CRC 檢查碼：
CRC CHK High	16-bit CRC 檢查碼由 2 個 8-bit 二進制組合
END	時間間隔

2. CRC 檢查碼：檢查碼由 Address 到 Data content 結束。其運算規則如下：

步驟一：令 16-bit 暫存器 (CRC 暫存器) =16#FFFF

步驟二：Exclusive OR 第一個 8-bit Byte 的訊息指令與低位元 16-bit CRC 暫存器，Exclusive OR，將結果存入 CRC 暫存器內。

步驟三：右移一位 CRC 暫存器，將 0 填入高位元處。

步驟四：檢查右移的值，如果是 0 將步驟 3 的新值存入 CRC 暫存器內否則 Exclusive OR 16#A001 與 CRC 暫存器，將結果存入 CRC 暫存器內。

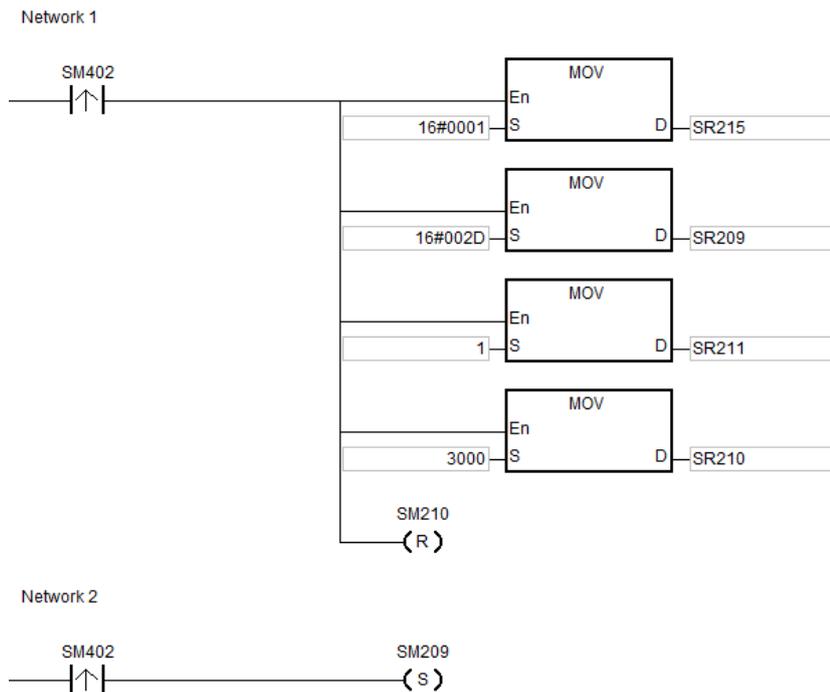
步驟五：重複步驟 3~步驟 4，將 8-bit 全部運算完成。

步驟六：重複步驟 2~步驟 5，取下一個 8-bit 的訊息指令，直到所有訊息指令運算完成。最後，得到的 CRC 暫存器的值，即是 CRC 的檢查碼。值得注意的是 CRC 的檢查碼必須交換放置於訊息指令的檢查碼中。

5. **n**：讀寫資料長度 (Data Length)，設定的資料量不可以大於 240 bytes。Coil 相關的通訊命令，其資料單位為 Bit，**n** 的範圍為 1~1920。Register 相關的通訊命令，其資料單位為 Word，則 **n** 的範圍為 1~120。
6. 本指令於程式中使用次數並無限制，但是不同指令 (如 MODRW、FWD...) 在同一時間使用相同的通訊埠時，僅會只有一個指令被執行。
7. 當發生通訊逾時時，逾時旗標會 ON 起來 (SM104、SM105)。若此問題已解除，則可以將此逾時旗標 (SM104、SM105) 設定為 OFF，在使用 MODRW 指令時，逾時時間不可設為 0，必須在 100 ~ 65535ms 區間內。
8. 於 MODBUS ASCII 模式時，使用者只需設定好，所欲傳送之資料，本指令會自動加入啟始字元 (:)、檢查碼 (LRC) 和結束字元 (CR LF)，其接收資料以 ASCII 字元形式儲存於內部暫存器內，AH500 自動將其內容轉換為 HEX 數值存放在 **S** 中。
9. 於 MODBUS RTU 模式時，使用者只需設定好，所欲傳送之資料，本指令會自動加入檢查碼 (CRC)，其接收資料以 HEX 數值形式儲存於 **S** 中。

通訊協定設定範例：

1. 通訊協定設定成 RS485 ASCII，9600，8，E，1，使用者可利用 HWCONFIG 或是特殊暫存器來設定通訊埠通訊能力。(HWCONFIG 相關設定請參照 ISPSOFT 使用手冊，通訊相關 SR、SM 暫存器設定可參考 API1800 RS 通訊應用指令)。
2. 設定通訊埠傳輸方式為 RS485 (SR215=1)。
3. 設定通訊埠其通訊速率與格式為 9600，8，E，1 (SR209=16#002D)。
4. 設定其自動詢問次數為 1 (SR211=1)。
5. 設定其逾時時間為 3000ms (SR210=3000)。
6. 設定為 ASCII 模式 (SM210=OFF)
7. 通訊埠通訊能力設定生效 (SM209=ON)。

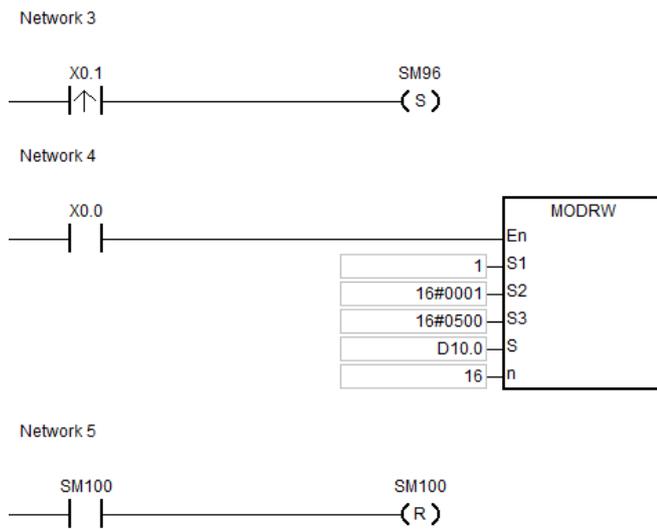


使用者若於 ISPSOft - > HWCONFIG - > COM Port 內直接對通訊埠進行設定，則可以省略此步驟。

程式範例一：

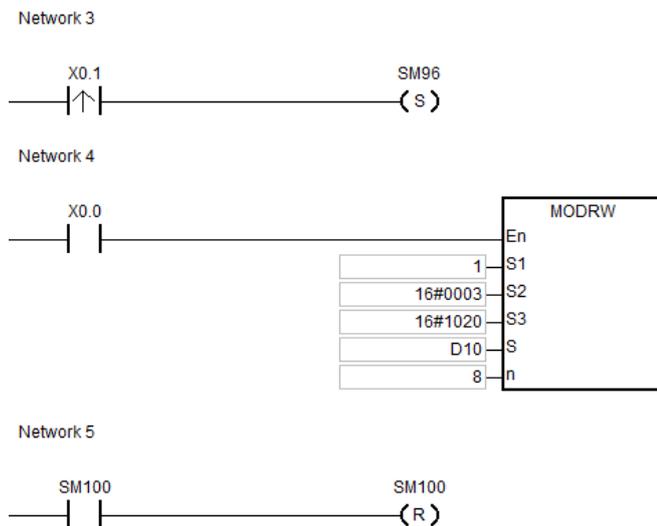
6

1. 功能碼 01 (16#01)：讀取多筆位元資料。(此範例讀取 16 筆)
2. AH500 與 DVP-ES2 連線：
當 SM96 為 ON 且 X0.0 也為 ON 的狀態時，將會對 DVP-ES2 進行資料的讀取，從 Y0 開始至 Y15 (Y0 的位址為 16#0500)。
3. DVP-ES2 所回覆的內容會儲存在 D10.0 到 D10.15。
4. DVP-ES2 所回傳的資料接收完畢後，會對 DVP-ES2 所回傳的資料格式進行確認，是否符合資料格式，如果正確無誤則 SM100=ON，如果回傳資料格式錯誤則 SM102=ON。



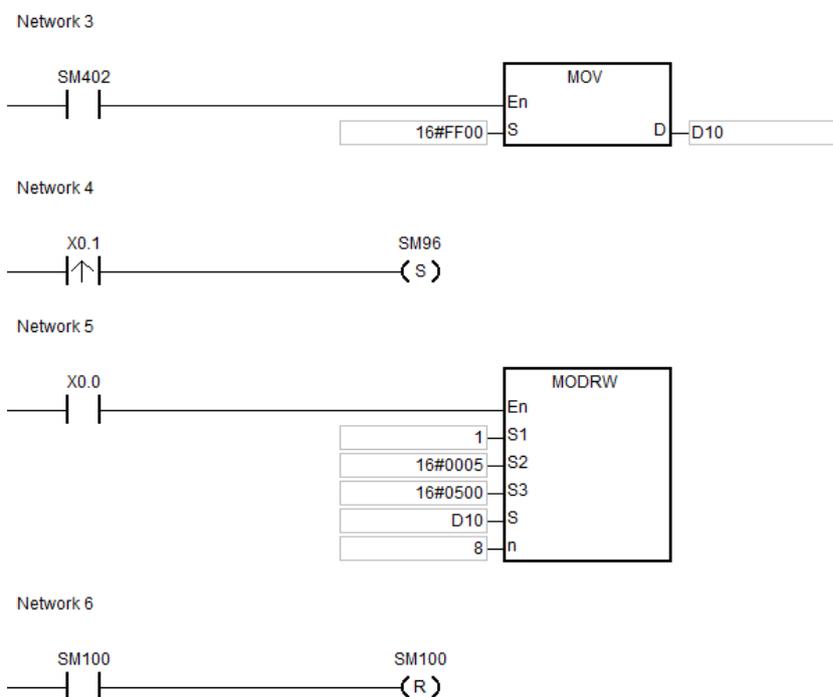
程式範例二：

1. 功能碼 03 (16#03): 讀取多筆暫存器資料。(此範例讀取 8 筆)
2. AH500 與 DVP-ES2 連線
當 SM96 為 ON 且 X0.0 也為 ON 的狀態時，將會對 DVP-ES2 進行資料的讀取，從 D20 開始至 D27 (D20 的位址為 16#1020)。
3. DVP-ES2 所回傳的資料會儲存在 D10 至 D17 裝置內。
4. DVP-ES2 所回傳的資料接收完畢後，會對 DVP-ES2 所回傳的資料格式進行確認，是否符合資料格式，如果正確無誤則 SM100=ON，如果回傳資料格式錯誤則 SM102=ON。



程式範例三：

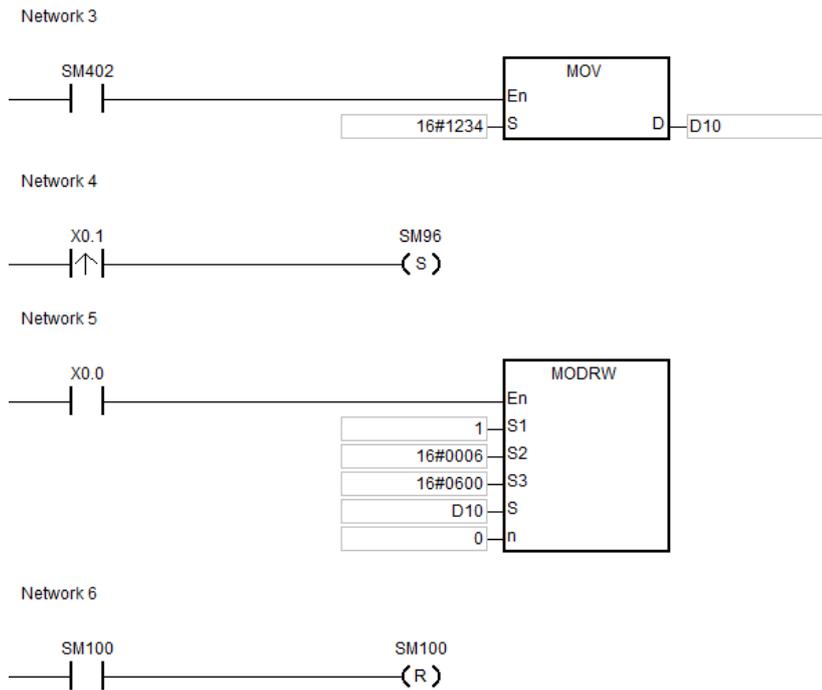
1. 功能碼 05 (16#05)：寫入單個位元資料至 DVP-ES2。
2. ASCII Mode：AH500 與 DVP-ES2 連線
當 SM96 為 ON 且 X0.0 也為 ON 的狀態時，將會對 DVP-ES2 的 Y0 寫入資料 (Y0 的位址為 16#0500)。
3. DVP-ES2 所回傳的資料接收完畢後，會對 DVP-ES2 所回傳的資料格式進行確認，是否符合資料格式，如果正確無誤則 SM100=ON，如果回傳資料格式錯誤則 SM102=ON。
4. 當 DVP-ES2 接收到此通訊命令的時候，會強制 Y0 為 ON 的狀態。
5. 此功能碼為寫入功能，所以於本指令中的 n 值，並不會被使用到。



6

程式範例四：

1. 功能碼 06 (16#06)：寫入單個暫存器資料至 DVP-ES2。
2. ASCII Mode：AH500 與 DVP-ES2 連線
當 SM96 為 ON 且 X0.0 也為 ON 的狀態時，將會對 DVP-ES2 的 T0 寫入資料 (T0 的位址為 16#0600)。
3. DVP-ES2 所回傳的資料接收完畢後，會對 DVP-ES2 所回傳的資料格式進行確認，是否符合資料格式，如果正確無誤則 SM100=ON，如果回傳資料格式錯誤則 SM102=ON。
4. 當 DVP-ES2 接收到此通訊命令的時候，會對 T0 寫入 D10 裝置內的資料。
5. 此功能碼為寫入功能，所以於本指令中的 n 值，並不會被使用到。



補充說明：

1. **S₁** 或 **S₂** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. **S** 裝置不足以寫入或讀取 **n** 所指定的長度時，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
3. **n** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#200B。
4. 當 **S₂** 通訊功能碼指定為 Bit 讀寫時 **S** 必須指定為 Bit 裝置，否則視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
5. 當 **S₂** 通訊功能碼指定為字元裝置讀寫時 **S** 必須指定為字元裝置，否則視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
6. 當通訊命令為 0x05 與 0x06 時，**n** 無作用，指令只會寫入 1 個 Bit 或 Word，惟 **n** 仍不能大於讀寫資料長度範圍。
7. 若執行 MODRW 的時候，送信旗標 (SM96、SM97) 都沒有 ON 起來的話，則不執行。
8. 當發生通訊逾時 (Timeout) 的時候，通訊逾時旗標 (SM104、SM105) 會被設定為 ON，等待 (Waiting) 旗標 (SM98、SM99) 會被設定為 OFF。
9. 當接收指令有錯誤 (Error) 時，錯誤旗標 (SM102、SM103) 會被設定為 ON，等待 (Waiting) 旗標 (SM98、SM99) 會被設定為 OFF。
10. 當 **S₂** 為讀寫 Word 裝置，則讀寫對方設備裝置也必須是 Word 裝置，當 **S₂** 為讀寫位元裝置，則讀寫對方設備裝置也必須為位元裝置。
11. RS-485/RS-232 通訊 MODRW 指令相關旗標信號與特殊暫存器：

旗標信號		功能說明
COM1	COM2	
SM96	SM97	送信要求

旗標信號		功能說明
COM1	COM2	
SM98	SM99	接收等待
SM100	SM101	接收完成
SM102	SM103	接收發生錯誤
SM104	SM105	通訊逾時
SM209	SM211	通訊協定變更改用

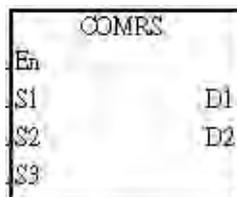
旗標信號的功能詳細說明請參考 API1800 RS 通訊應用指令。

API	指令碼	運算元	功能
1812	COMRS	S ₁ · S ₂ · S ₃ · D ₁ · D ₂	通訊傳送與接收指令

裝置	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	●	●			●	●		●	●							

脈波執行型	16 位元指令 (11steps)	32 位元指令
-	AH500	-

符號：



- S₁ : 通訊埠編號 (1~2) Word
- S₂ : 傳送資料來源 Word
- S₃ : 傳送長度設定值 Word
- D₁ : 接收通訊資料的起始裝置元件 Word
- D₂ : 接收資料結束條件參數設定 Word

指令說明：

- 僅適用於版本 V1.03 以上。
- 當接收結束條件有指定特定字元時，建議使用於 ASCII 編碼之通訊資料，若非 ASCII 編碼之通訊資料，則建議改用通訊逾時時間做為結束條件。
- COMRS 指令僅支援 AH500 主機內建的 COM·AHCPU5xx-EN 系列為 COM1·AHCPU5xx-RS2 系列為 COM1 與 COM 2。
- S₁ : 通訊埠編號範圍。
S₁ = 1 (COM1)，S₁ = 2 (COM2)，若超出 PLC 通訊埠範圍時，則指令將不會執行任何通訊資料接收。
- S₂ 通訊資料傳送來源與 S₃ 傳送長度設定值，其範例說明如下：假設 S₂ 選擇 D100 及 S₃ 為 10，並設定為 8 位元模式，則指令將會傳送 D100~D109 的下 8 位元數值共 10 個字，由 S₁ 指定的通訊埠傳送出去。
- 當 S₃ 傳送長度設定值為 0 時，即表示不需要傳送字串；傳送長度最大值為 1000 個字元。
- D₁ 為已接收資料長度值，D₁+1~ D₁+n 為存放接收資料位址，說明如下：假設 D₁ 設為 D200，D₂=3 及 D₂+1 設定結束字元 16#0D0A，則指令將會依照接收資料順序存放於 D201 之後的下 8 位元中 (上 8 位元不變化)，直到兩個連續結束位元為 16#0D 與 16#0A 時，即停止接收資料，並將接收長度填入 D200 內，接著設定完成旗標。

8. **D₂+0** 為接收模式參數設定，**D₂+1** 為接收條件參數設定，其詳細說明如下表所示。

說明列表如下：

D₂	D₂ 接收模式說明	D₂+1 參數內容說明	備註
0	不接收通訊資料	不使用	即傳送完成後，設定完成旗標。
1	通訊資料不連續時間超出 D₂+1 設定值時，即為接收完成。	時間長度設定值，時間單位為 1ms，設定值範圍為 2~3000。	設定值超出 3000 時，其值為 3000；小於 2 時，其值為 2。
2	接收資料有一個特定字元為結尾字元	特定結尾字元設定值	如特定字元為 16#0A，則 D₂+1 設定為 16#000A
3	接收資料有兩個連續特定字元為結尾字元	兩個特定結尾字元設定值	如兩個特定字元順序為 16#0D 與 16#0A，則 D₂+1 設定為 16#0D0A
4	接收資料有一個特定字元為開始字元，以及不連續時間超出 D₂+1 設定值時，即為接收完成。	上 8 位元為特定開始字元，下 8 位元為不連續時間設定值 (範圍 2~255ms)。	如開始字元為 16#3A 以及不連續時間 15ms，則 D₂+1 設定為 16#3A0F
5	接收資料有一個特定字元為開始字元與一個結尾字元	特定開始與結尾字元設定值	如開始字元為 16#3A，結尾字元為 16#0A，則 D₂+1 設定為 16#3A0A
6	接收資料有一個特定長度為接收完成	D₂+1 為接收長度設定值	如接收 10 個字元，則 D₂+1 設定為 10
其他	超出現有支援模式範圍，則指令將不會被啟動。		

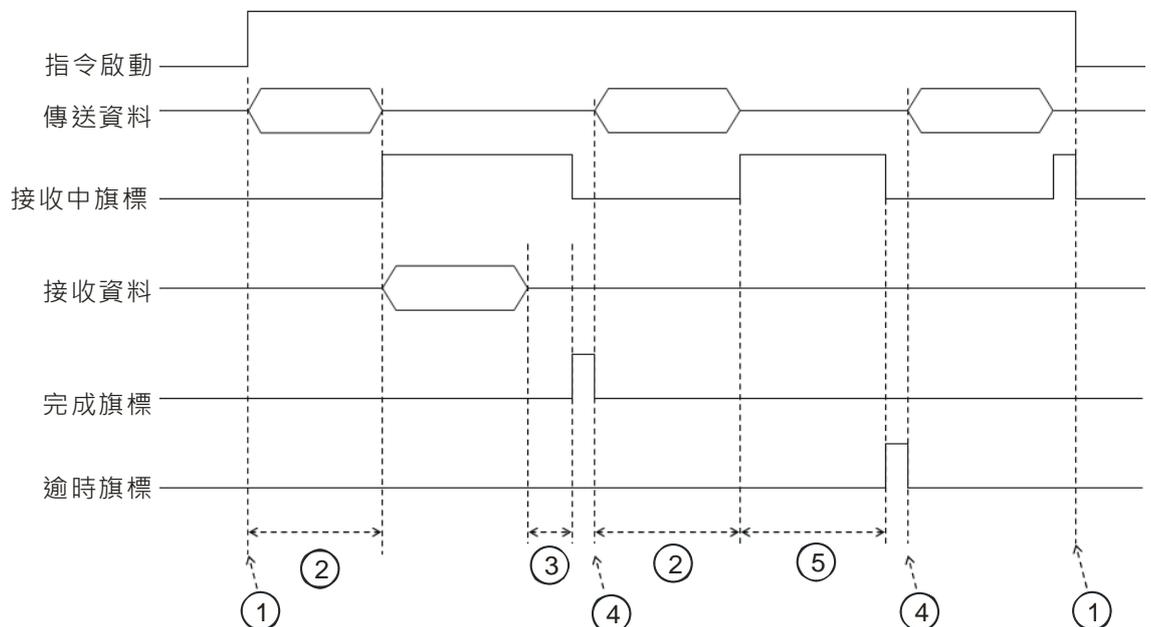
9. 當 **D₂** 內容值為 1~5 時，最大接收長度為 1000 字元，到達 1000 字元時，指令自動結束。

10. 通訊埠對應的特殊輔助繼電器 **SM** 與特殊資料暫存器 **SR** 如下：

通訊埠編號	COM1	COM2	備註
接收中旗標	SM98	SM99	由 PLC 設定 ON 為接收中，接收結束也由 PLC 清除為 OFF。
完成旗標	SM100	SM101	由 PLC 設定 ON 為完成，但須由使用者程式清除 OFF，然後等待下一筆通訊資料。
通訊逾時旗標	SM104	SM105	由 PLC 設定 ON 為逾時，但須由使用者程式清除 OFF，然後等待下一筆通訊資料。
逾時時間設定	SR210	SR213	數值為 0 表示不啟動通訊逾時功能，時間單位為 1ms。
資料傳送/接收模式	SM106	SM107	傳送及接收時的資料格式，ON 時為 8 位元，OFF 時為 16 位元。

11. 通訊傳送與接收時序圖如下：

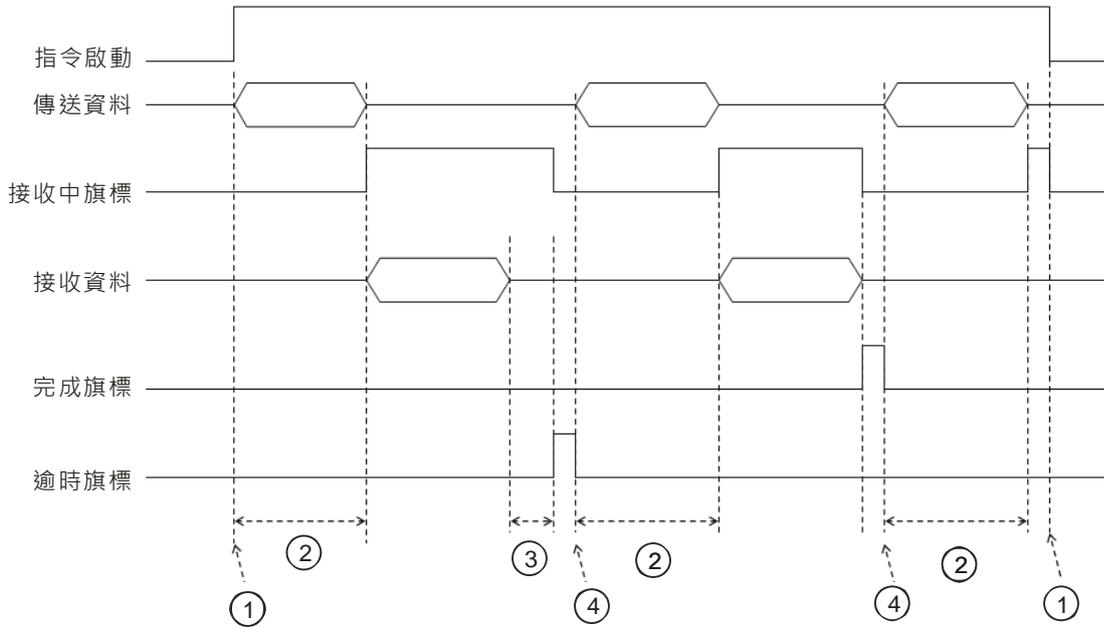
- 接收模式為 0 時
需注意資料傳送時無法中途取消，此時即使指令啟動條件不成立，仍會繼續傳送資料，惟傳送完成後不會將完成旗標設為 ON。
- 接收模式為 1 或 4 時



圖示符號說明：

- ① → 由使用者設定啟動/關閉
- ② → 傳送資料時間，此期間通訊逾時不會被計時。
- ③ → 接收到第 1 個字元時，即開始累積計時結束時間，每接收一個字元即清除累積計時值一次，直到累積計時超出 D_2+1 設定值之後，則產生通訊完成旗標。

- ④ → 由使用者清除完成或逾時旗標，若指令還處於開啟狀態，則下個週期掃描到指令時，將自動重新傳送下一筆通訊資料。
- ⑤ → 當接收開始時即開始累積計時，直到累積計時超出通訊逾時設定值之後，則產生通訊逾時旗標。建議通訊逾時時間最好設定比 D_2+1 結束時間長。
- 接收模式為 2、3、5 或 6 時



圖示符號說明：

- ① → 由使用者設定啟動/關閉
- ② → 傳送資料時間，此期間通訊逾時不會被計時。
- ③ → 接收開始時即開始累積計時，每接收一個字元即清除累積計時值一次，直到累積計時超出通訊逾時設定值之後，則產生通訊逾時旗標。
- ④ → 由使用者清除完成或逾時旗標，若指令還處於開啟狀態，則下個週期掃描到指令時，將自動重新傳送下一筆通訊資料。

12. 資料傳送/接收模式：

《8 位元模式》：使用者將編輯好的命令儲存在傳送裝置起始位址上，傳送出去的命令包含頭尾碼。將 16 位元資料分成上 8 位元，下 8 位元，上 8 位元被省略，僅下 8 位元為有效資料可做資料的傳送和接收。(以標準 MODBUS 為例)

傳送資料：(PLC→外部機器)

D10下	D11下	D12下	D13下	D14下	D15下	D16下
頭碼		來源資料暫存器由 D10下 8 位元開始 長度=7			尾碼 1	尾碼 2

接收資料：(外部機器→PLC)

D100下	D101下	D102下	D103下	D104下	D105下	D106下
頭碼		來源資料暫存器由 D100 下 8 位元開始 長度=7			尾碼 1	尾碼 2

《16 位元模式》：使用者將編輯好的命令儲存在傳送裝置起始位址上，傳送出去的命令包含頭尾碼。當 SM106/SM107=OFF 時，指定為 16 位元轉換模式，將 16 位元資料分成上 8 位元與下 8 位元做資料的傳送和接收。

傳送資料：(PLC→外部機器)

D10下	D10上	D11下	D11上	D12下	D12上	D13下
頭碼		來源資料暫存器由 D10下 8 位元開始 長度=7			尾碼 1	尾碼 2

接收資料：(外部機器→PLC)

D100下	D100上	D101下	D101上	D102下	D102上	D103下
頭碼		來源資料暫存器由 D100 下 8 位元開始 長度=7			尾碼 1	尾碼 2

PLC 接收資料會將外部機器傳入資料包含頭碼、尾碼一起接收，所以長度之設定要注意。

6

補充說明：

1. 此 COMRS 通訊指令無使用次數限制，但每個通訊埠都僅能被一條通訊指令啟動，因此後面啟動之通訊指令將不會被執行。
2. 通訊指令包含有 RS、MODRW、FWD 及 REV 等等都會佔用通訊埠之通訊指令。
3. 此 COMRS 指令無檢查碼機制之設計，須此功能者可搭配現有提供之便利指令自行組合運用。
4. D2 內容值為 2、3、5、6 時，建議設定通訊逾時。若有設定通訊逾時，當指令沒有收到結束字元時，才能重新進行重傳機制。
5. 每次剛啟動指令或重新開始接收新的下一筆通訊資料時，此指令並不會自動清除 $D_{1+0} \sim D_{1+n}$ 裝置內的數值，相關的接收資料訊息 (如接收資料是否有接收？與接收長度為何？) 皆以完成旗標由 OFF 變為 ON 時為基準。若需清除這些接收資料區域時，建議可搭配 ZRST 指令清除。
6. 當 S_1 的內容值超出範圍時 (1~2)，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
7. S_2 裝置不足以讀取 S_3 所指定的長度時，指令不執行，SM0=ON，錯誤碼為 SR0=16#2003。
8. 當 D_2 的內容值超出範圍時 (0~6)，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

9. 當 **D₂** 的內容值為 6，**D₁** 裝置不足以寫入 **D₂+1** 所指定的長度時，指令不執行，**SM0=ON**，**SR0** 錯誤碼為 16#2003。
10. 當 **D₂** 的內容為 1~5 時，接收資料超出 **D₁** 最大裝置範圍，則超出裝置範圍的接收資料將會被忽略。
11. 當接收完成旗標變為 **ON** 時，其通訊資料將暫停接收，若此期間通訊埠還有資料傳送，將不再接收。
12. 當 **S₃** 傳送長度設定值為小於 0 或大於 1000 時，指令不執行，**SM0=ON**，錯誤碼為 **SR0=16#2003**。

6

6.20 其他指令

6.20.1 其他指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>1900</u>	WDT	–	✓	逾時監視計時器	1
<u>1901</u>	DELAY	–	✓	延遲指令	3
<u>1902</u>	GPWM	–	–	一般用脈波波寬調變	7
<u>1903</u>	TIMCHK	–	–	時間檢查	7
<u>1904</u>	EPUSH	–	✓	間接指定暫存器存入	3
<u>1905</u>	EPOP	–	✓	間接指定暫存器讀出	3

6.20.2 其他指令說明

API	指令碼		運算元	功能
1900	WDT	P	—	逾時監視計時器

脈波執行型	16 位元指令 (1 step)	32 位元指令
—	AH500	—

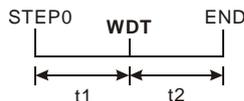
符號：



指令說明：

- AH500 系列 PLC 系統中有一逾時監視計時器 (Watchdog Timer)，用來監視 PLC 系統的正常運轉。
- WDT 指令可用來清除 PLC 中之監控定時器之計時時間。當 PLC 的掃描程式時間過長，超過使用者設定的 WDT 時間(預設值 200ms)時，PLC ERROR 的指示燈會亮，PLC 會自動變 STOP，使用者只要重新做 STOP=>RUN 就可以恢復正常。
- 令逾時監視計時器動作的時機：
 - PLC 系統發生異常。
 - 程式執行時間太長，造成掃描時間大於 (Watchdog Timeout) 的內容值。可以下列 2 種方法來改善。

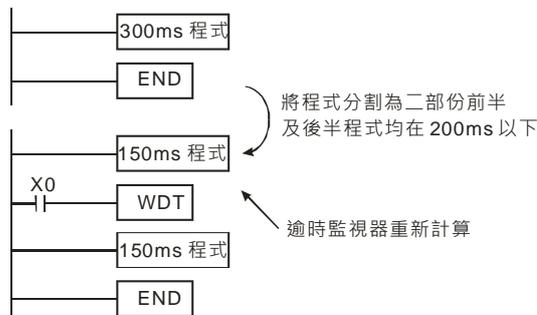
(a) 使用 WDT 指令。



(b) 要改變逾時監視時間請參考 ISPSOft 手冊中的硬體組態工具說明。

程式範例：

若程式掃描時間為 300ms，此時，可將程式分割為 2 部份，並在中間放入 WDT 指令，使得前半及後半程式都在 200ms 以下。

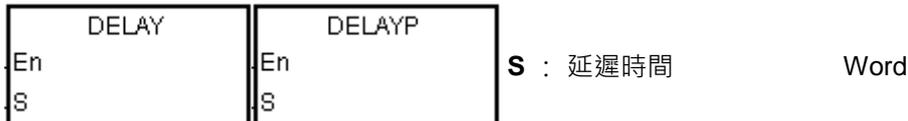


補充說明：

設定 WDT 時間的方式，請參考 ISPSOft 的使用手冊說明。

API	指令碼			運算元								功能					
1901		DELAY	P	S								延遲指令					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●		●		●	○	○		
脈波執行型											16 位元指令 (3 steps)			32 位元指令			
AH500											AH500			—			

符號：



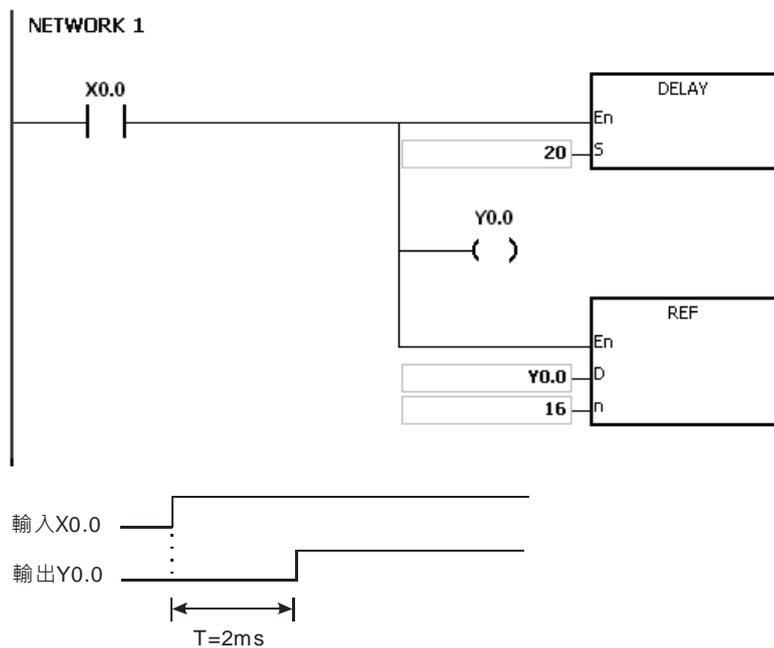
指令說明：

執行 DELAY 指令後，在每次掃描週期 DELAY 指令後面之程式執行會依使用者指定之時間作延遲。

S 延遲時間的單位：0.1ms

程式範例：

當 X0.0=ON 時，執行 DELAY 指令延遲 2ms 後才執行後面程式，Y0.0 導通且更新 Y0.0~Y0.15 的狀態。



補充說明：

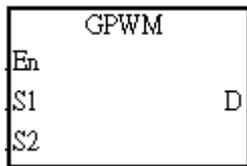
1. **S** ≤ 0 時，不會有延遲時間。
2. **S** > 1000，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 使用者可依實際狀況來調整延遲時間。
4. DELAY 指令會受到通訊或其它影響而增加延遲時間。

API	指令碼			運算元								功能					
1902		GPWM		$S_1 \cdot S_2 \cdot D$								基本脈波寬度調變					

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

脈波執行型	16 位元指令(7 steps)	32 位元指令
—	AH500	—

符號：



S_1 : 脈波輸出寬度 Word
 S_2 : 脈波輸出周期 Word
D : 脈波輸出裝置 Bit

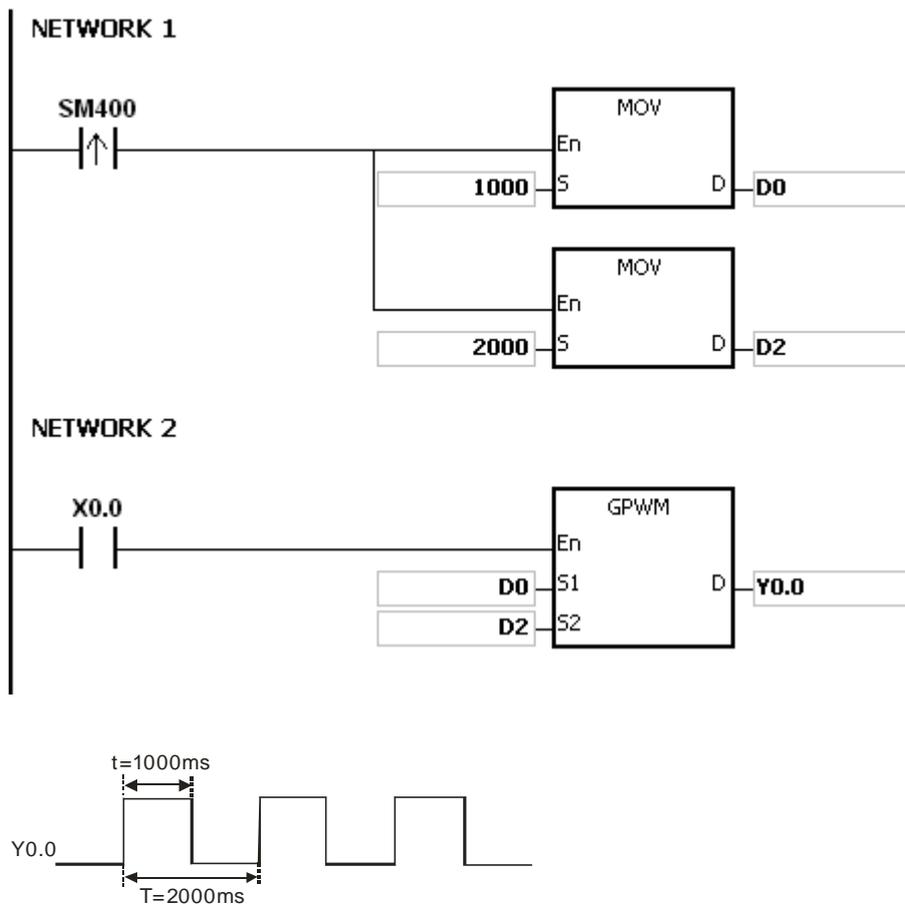
指令說明：

1. GPWM 指令執行時，指定 S_1 脈波輸出寬度與 S_2 脈波輸出周期，並由 D 脈波輸出裝置輸出。
2. S_1 脈波輸出寬度指定 t：0~32,767ms。
3. S_2 脈波輸出周期指定為 T：1~32,767ms，但 $S_1 \leq S_2$ 。
4. S_{2+1} 、 S_{2+2} 為系統用參數，請勿佔用。
5. 當 $S_1 \leq 0$ 時，脈波輸出裝置無輸出，當 $S_1 \geq S_2$ 時，脈波輸出裝置一直為 ON。
6. S_1 、 S_2 可在 GPWM 指令執行時更改。
7. 當條件接點未啟動時，D 裝置輸出 0。
8. 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。

程式範例：

程式啟動時，D0=1000，D2=2000，當 X0.0=ON 時，Y0.0 輸出以下脈波，當 X0.0=OFF 時，Y0.0 輸出也變成 OFF。



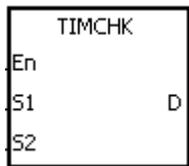


補充說明：

1. 此指令是以掃描週期去計數，因此最大誤差為 1 個 PLC 掃描週期。 S_1 、 S_2 與 $(S_2 - S_1)$ 的值必須 $>$ PLC 掃描週期，否則 GPWM 輸出會有誤動作。
2. 若將此指令置於功能塊或中斷中使用，則會產生 GPWM 輸出不準確的情況發生，請特別注意。
3. S_2 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元							功能								
1903		TIMCHK		$S_1 \cdot S_2 \cdot D$							時間檢查								
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF		
S_1	●	●			●	●		●	●				●						
S_2	●	●			●	●		●	●		●	○	●	○	○				
D	●	●	●	●				●	●				●						
											脈波執行型			16 位元指令 (7 steps)			32 位元指令		
											-			AH500			-		

符號：



- S_1 : 存放測量值所經過的時間 Word
- S_2 : 測量設定值 Word
- D : 測量時間到達時的輸出裝置 Bit

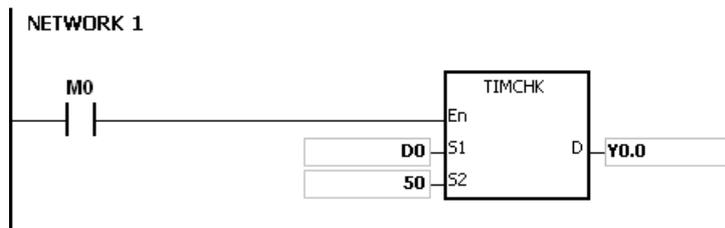
指令說明：

1. 條件接點 ON 時， S_1 開始計數，直到 $S_1 \geq S_2$ ，D 輸出裝置=ON，之後就算條件接點變成 OFF， S_1 內容值不變且 D 裝置一樣為 ON。
2. 條件接點由 OFF→ON 時， S_1 被清除為 0，D 輸出裝置=OFF。
3. 計數的單位為 100ms。
4. S_{1+1} 、 S_{1+2} 為系統用參數，請勿佔用。
5. 使用線上編輯時，請重新啟動條件接點，以達到指令的初始化。

6

程式範例：

M0=ON 時，D0 開始計數，直到 $D0 \geq 50$ (5 秒)，Y0.0=ON，之後就算條件接點變成 OFF，D0 內容值不變且 Y0.0 一樣為 ON。

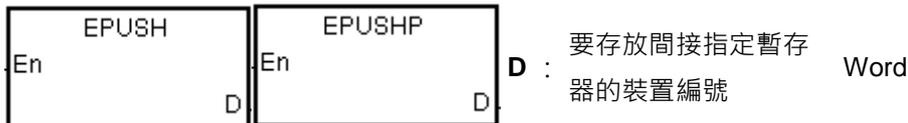


補充說明：

1. 若 S_{1+2} ，裝置超過範圍，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. S_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

API	指令碼			運算元								功能						
1904		EPUSH	P	D								間接指定暫存器存入						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF	
D	●	●			●	●		●	●				●					
										脈波執行型			16 位元指令 (3 steps)			32 位元指令		
										AH500			AH500			-		

符號：



指令說明：

1. 將 E0~E31 的內容值存放到 D 的內容 (保存的次數) 所指定的裝置。
2. 每次執行存放需 34 個裝置 (最後兩個為系統保留用)，因此若保存次數為 n (D 的內容值)，則執行時 E0~E31 會存放在 $D+34*n+1 \sim D+34*n+32$ 中，且保存次數 $D=n+1$ 。
3. 如果 EPUSH 執行多次，則 E0~E31 會被依序往下存放，所以 D 的容量要夠大。
4. 配合 EPOP 指令使用，可以達到 E 裝置的堆疊使用。



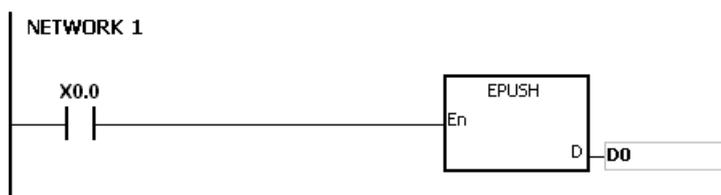
程式範例：

給定 $D0=0$ ，當第一次 $X0.0=ON$ 時，E0~E31 的內容會分別傳送至 $D1 \sim D32$ ， $D0=1$ 。

第二次 $X0.0$ 由 OFF→ON 時，E0~E31 的內容會分別傳送至 $D35 \sim D66$ ， $D0=2$ 。

第 N 次 $X0.0$ 由 OFF→ON，E0~E31 的內容會分別傳送至：

裝置編號 ($D0$ 的內容值 * 34) + 1 ~ ($D0$ 的內容值 * 34) + 32

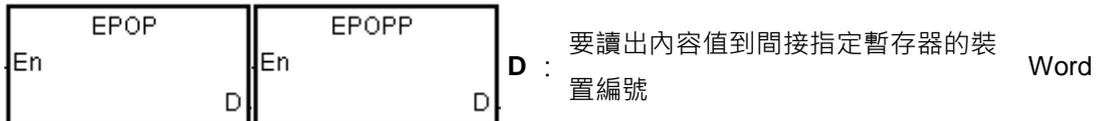


補充說明：

1. **D** 內容值 <0 時，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2003**。
2. **D+** (**D** 內容值+1) *34-1 裝置容量不足時，指令不執行，**SM0=ON**，錯誤碼 **SR0=16#2003**。

API	指令碼			運算元							功能									
1905		EPOP	P	D							間接指定暫存器讀出									
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF			
D	●	●			●	●		●	●				●							
										脈波執行型			16 位元指令 (3 steps)				32 位元指令			
										AH500			AH500				—			

符號：



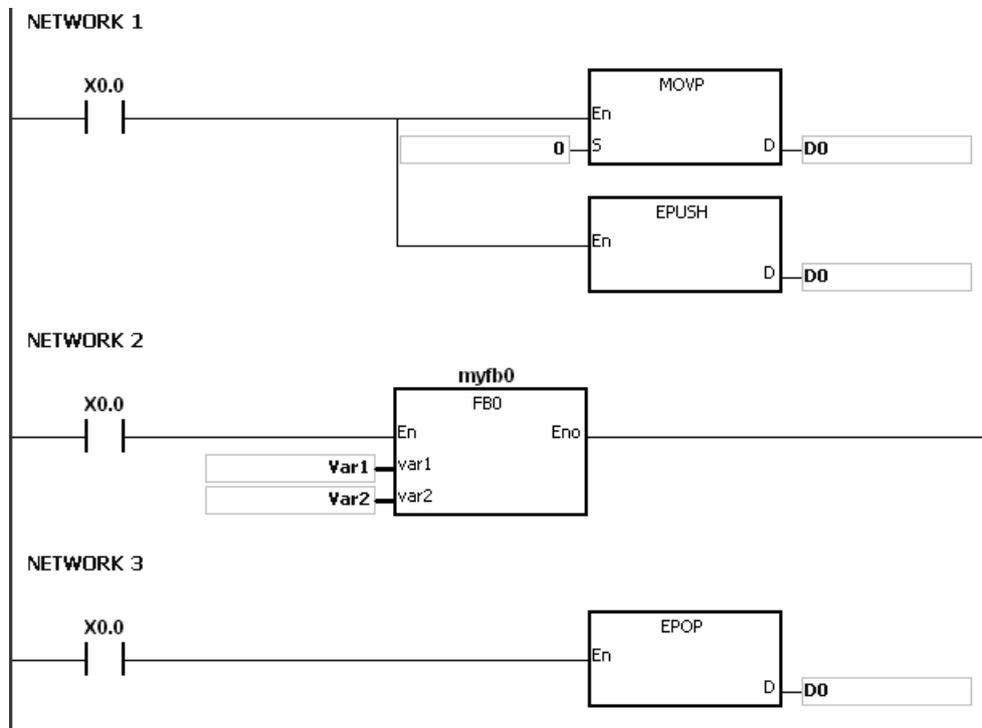
指令說明：

1. 將 **D** 的內容值讀出到裝置 E0~E31，並且將 **D** 的內容值（保存的次數）減 1。
2. 每次執行存放需 34 個裝置（最後的兩個裝置為系統保留用），因此若保存次數為 n (D 的內容值)，則該次會讀取 $D+34*(n-1)+1 \sim D+34*(n-1)+32$ 中的內容值寫入 E0~E31，且保存次數 $D=n-1$ 。
3. 取出的順序為最後放入 **D** 中的值。



程式範例：

當 X0.0=ON 時，先將 D0 內容值設為 0，然後使用 EPUSH 將 E0~E31 的內容分別傳送至 D1~D32，再開始執行功能塊 FB0，當功能塊 FB0 執行完畢之後，再用 EPOP 將 D1~D32 的值讀出至 E0~E31。



6

補充說明：

1. D 內容值 ≤ 0 時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. D+D 內容值 * 34-1 裝置容量不足時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。

6.21 字串處理指令

6.21.1 字串處理指令一覽表

API	指令碼 (位元)			P 指令	功能	STEPS
	16	32	64			
<u>2100</u>	BINDA	DBINDA	–	✓	有號數十進制→ASCII 變換	5
<u>2101</u>	BINHA	DBINHA	–	✓	BIN 十六進制→十六進制 ASCII 變換	5
<u>2102</u>	BCDDA	DBCDDA	–	✓	BCD→ASCII 變換	5
<u>2103</u>	DABIN	DDABIN	–	✓	有號數十進制 ASCII→有號數十進制 BIN 變換	5-11
<u>2104</u>	HABIN	DHABIN	–	✓	十六進制 ASCII→十六進制 BIN 變換	5-11
<u>2105</u>	DABCD	DDABCD	–	✓	BCD ASCII→BCD 變換	5-11
<u>2106</u>	\$LEN	–	–	✓	計算字串長度	5-11
<u>2107</u>	\$STR	\$DSTR	–	✓	BIN→String	7
<u>2108</u>	\$VAL	\$DVAL	–	✓	String→BIN	7-13
<u>2109</u>	\$FSTR	–	–	✓	Float→String	7-8
<u>2110</u>	\$FVAL	–	–	✓	String→Float	5-11
<u>2111</u>	\$RIGHT	–	–	✓	從右邊擷取字串	7-13
<u>2112</u>	\$LEFT	–	–	✓	從左邊擷取字串	7-13
<u>2113</u>	\$MIDR	–	–	✓	區段擷取字串	7-13
<u>2114</u>	\$MIDW	–	–	✓	區段字串取代	7-13
<u>2115</u>	\$SER	–	–	✓	字串搜尋	9-21
<u>2116</u>	\$RPLC	–	–	✓	字串取代	11-17
<u>2117</u>	\$DEL	–	–	✓	指定字串刪除	9
<u>2118</u>	\$CLR	–	–	✓	字串清除	3
<u>2119</u>	\$INS	–	–	✓	字串插入	9-15
<u>2120</u>	–	FMOD	–	✓	浮點數轉 BCD 浮點數	7-8
<u>2121</u>	FREXP	–	–	✓	BCD 浮點數轉浮點數	7

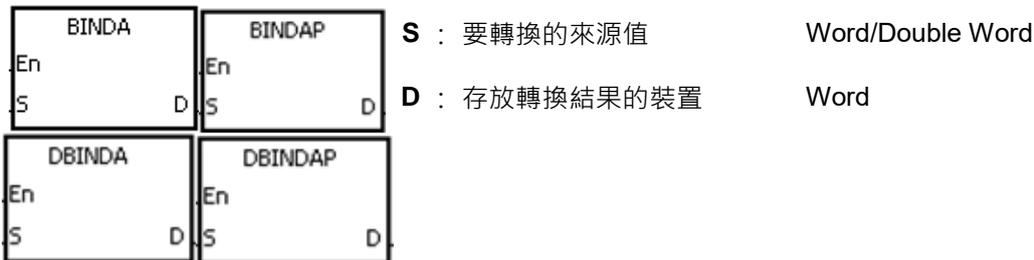
6.21.2 字串處理指令說明

API	指令碼			運算元				功能			
2100	D	BINDA	P	S · D				有號數十進制→ASCII 變換			

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

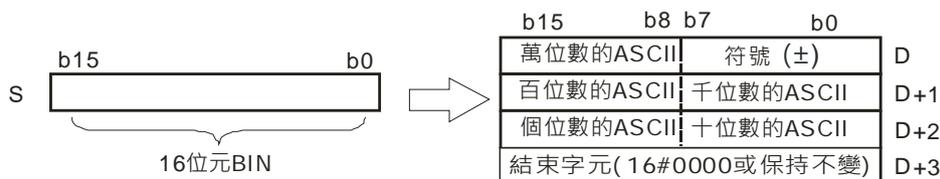
脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：



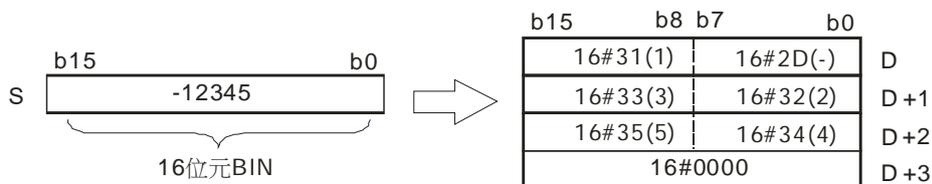
指令說明：

- 將有號數十進制 BIN 值的資料來源 S 做 ASCII 的轉換，並儲存於 D。
- 本指令支援字串結束模式旗標 SM690，來控制字串的結束字元。
- 16 位元指令中，S 的範圍-32768~32767，且固定為 6 位數的 BIN 值；D 連續佔用四個 Word 裝置，資料轉換方式如圖說明：

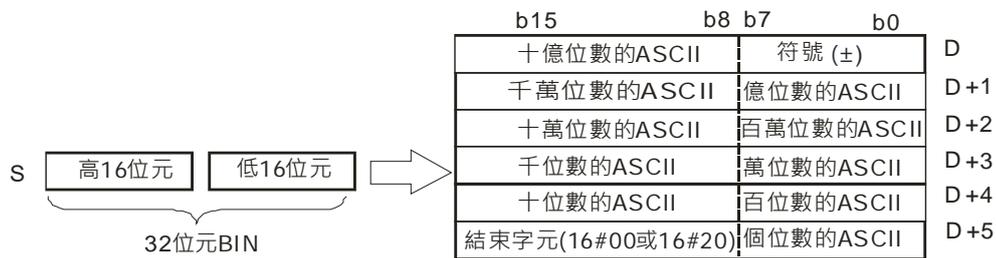


其中 SM690=OFF 時，D+3 填入字串結束模式 16#0000；SM690=ON 時，不填入字串結束模式，因此 D+3 的值保持不變。

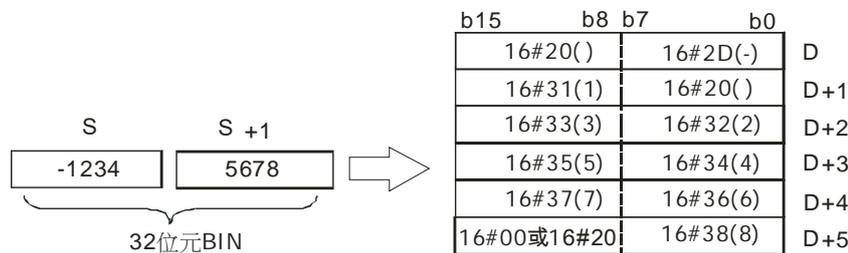
如果 S 的內容值為正數，則 D 的符號會被填入 16#20；如果 S 的內容值為負數，則 D 的符號會被填入 16#2D。如下範例：S=-12345 且 SM690=OFF 則轉換後的值如圖說明：



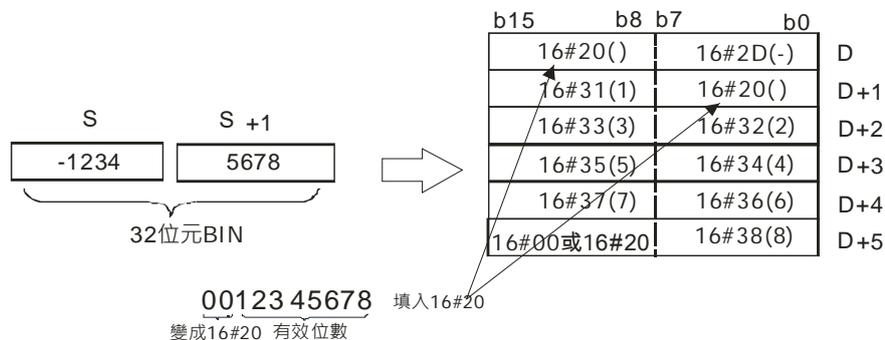
4. 32 位元指令中，S 的範圍-2147483648~2147483647，且固定為 11 位數的 BIN 值；D 連續佔用六個 Word 裝置，資料轉換方式如圖說明：



其中 SM690=OFF 時，D+5 的上 8 位元填入 16#00；SM690=ON 時，D+5 的上 8 位元填入 16#20。如果 S 的內容值為正數，則 D 的符號會被填入 16#20；如果 S 的內容值為負數，則 D 的符號會被填入 16#2D。如下範例：S=-12345678 則轉換後的值如圖說明：

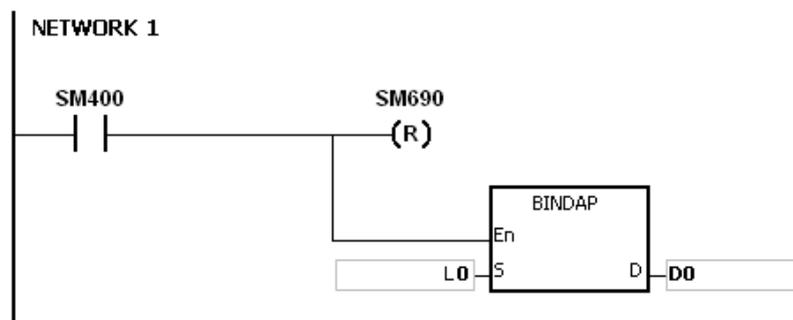


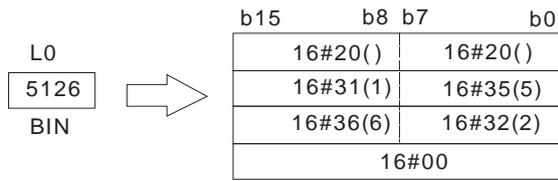
5. 若 S 內容值 (BIN) 如果有位數不足時，則 D 相對應的存放位址，會被補 16#20，例如：



程式範例一：

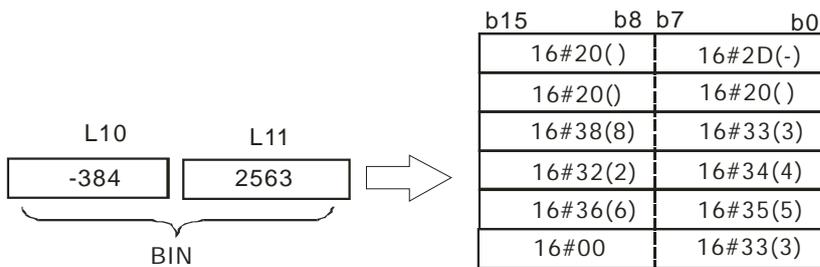
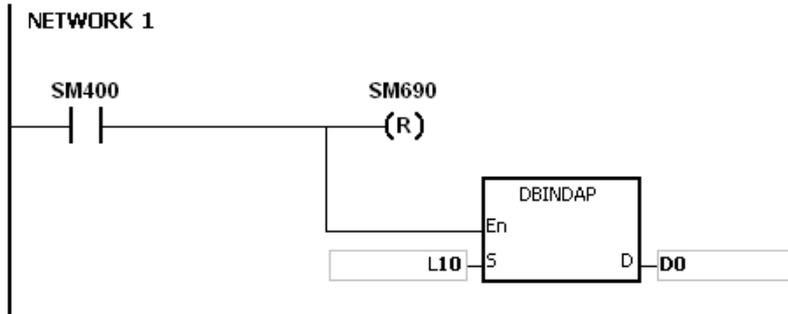
給定 L0=5126 且 SM690=OFF，PLC RUN 時 D0=16#2020、D1=16#3135、D2=16#3632、D3=16#0000。





程式範例二：

給定 L10=-3842563 且 SM690=OFF · PLC RUN 時 D0=16#202D、D1=16#2020、D2=16#3833、D3=16#3234、D4=16#3635、D5=16#0033。



6

補充說明：

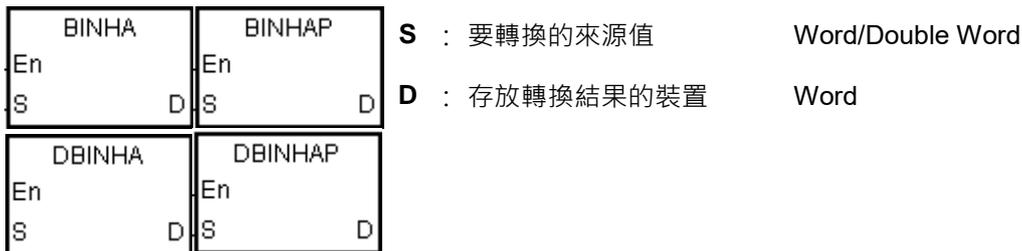
1. 16 位元指令中 D+3 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
2. 32 位元指令中 D+5 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
3. 16 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of WORD/INT。
4. 32 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [6] of WORD/INT。

API	指令碼			運算元								功能				
2101	D	BINHA	P	S · D								BIN 十六進制→十六進制 ASCII 變換				

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

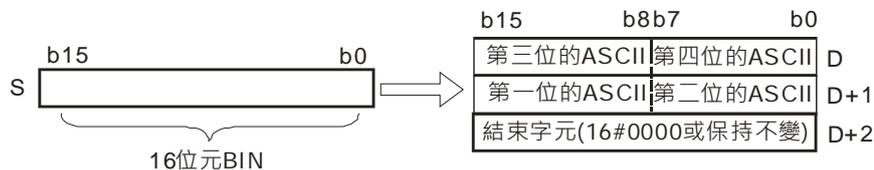
脈波執行型	16 位元指令 (5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

符號：

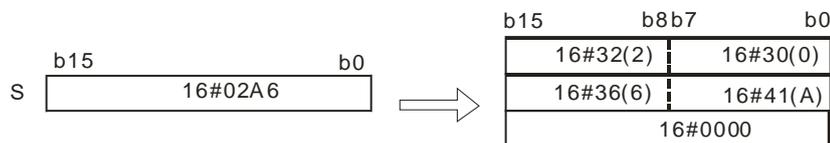


指令說明：

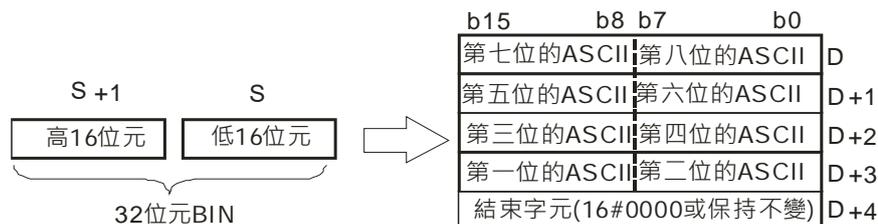
- 將十六進制 BIN 值的資料來源 S 做 ASCII 的轉換，並儲存於 D。
- 本指令支援字串結束模式旗標 SM690，來控制字串的結束字元。
- 16 位元指令中，S 的範圍 16#0000~16#FFFF，且固定為 4 位數的 BIN 值；D 連續佔用三個 Word 裝置，資料轉換方式如圖說明：



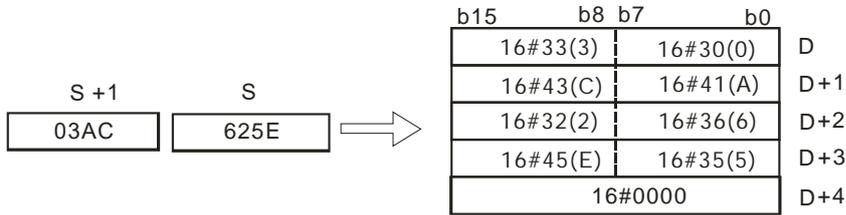
其中 SM690=OFF 時，D+2 填入字串結束模式 16#0000；SM690=ON 時，不填入字串結束模式，因此 D+2 的值保持不變。如下範例：S=16#02A6 且 SM690=OFF，則轉換後的結果如下圖所示。



- 32 位元指令中，S 的範圍 16#00000000~16#FFFFFFFF，且固定為 8 位數的 BIN 值；D 連續佔用五個 Word 裝置，資料轉換方式如圖說明：

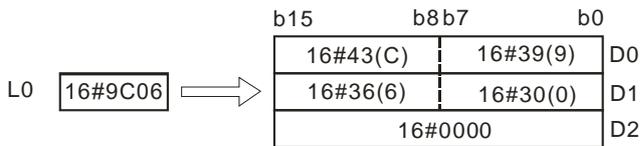
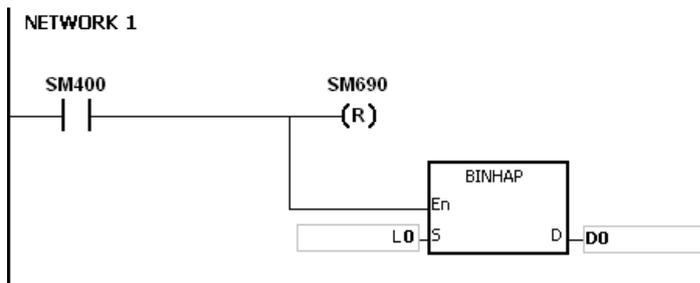


其中 SM690=OFF 時，D+4 填入字串結束模式 16#0000；SM690=ON 時，不填入字串結束模式，因此 D+4 的值保持不變。例如 S=16#03AC625E 且 SM690=OFF，則轉換後的結果如下圖所示。



程式範例一：

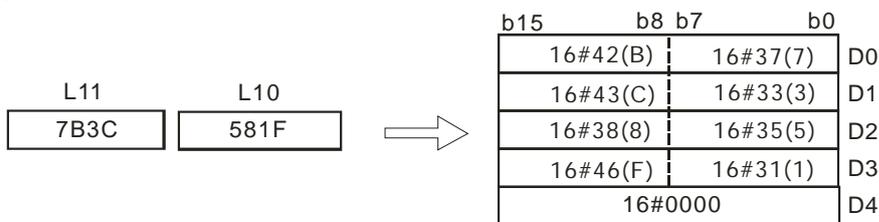
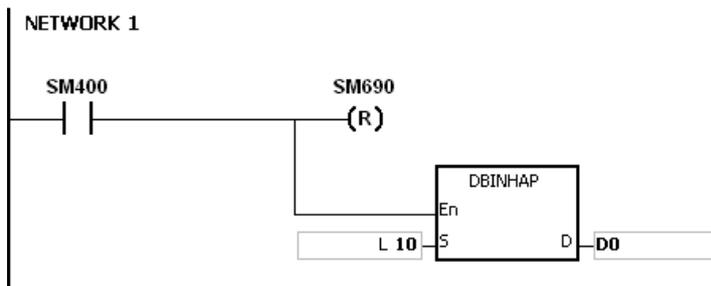
給定 L0=16#9C06 且 SM690=OFF，PLC RUN 時 D0=16#4239、D1=16#3630、D2=16#0000。



6

程式範例二：

給定 L10=16#7B3C581F 且 SM690=OFF，PLC RUN 時 D0=16#4237、D1=16#4333、D2=16#3835、D3=16#4631、D4=16#0000。



補充說明：

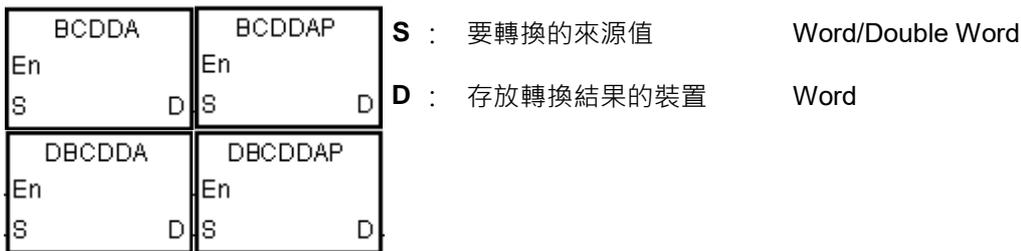
1. 16 位元指令中 **D+2** 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
2. 32 位元指令中 **D+4** 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
3. 16 位元指令 **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
4. 32 位元指令 **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [5] of WORD/INT。

API	指令碼			運算元						功能					
2102	D	BCDDA	P	S · D						BCD→ASCII 變換					

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

脈波執行型	16 位元指令(5 steps)	32 位元指令 (5 steps)
AH500	AH500	AH500

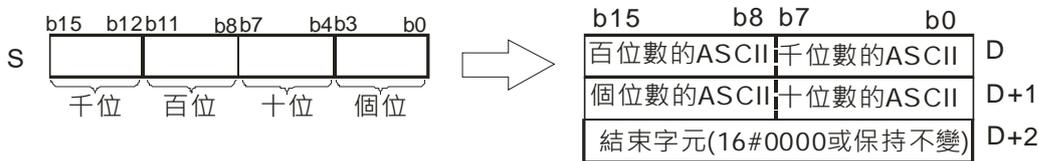
符號：



指令說明：

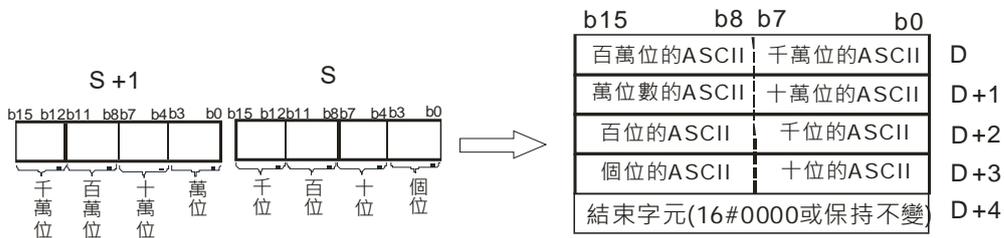
- 將 BCD 值的資料來源 **S** 做 ASCII 的轉換，並儲存於 **D**。
- 本指令支援字串結束模式旗標 **SM690**，來控制字串的結束字元。
- 16 位元指令中，**S** 的範圍 BCD 0~9999，且固定為 4 位數的 BCD 值；**D** 連續佔用三個 Word 裝置，資料轉換方式如圖說明：

6



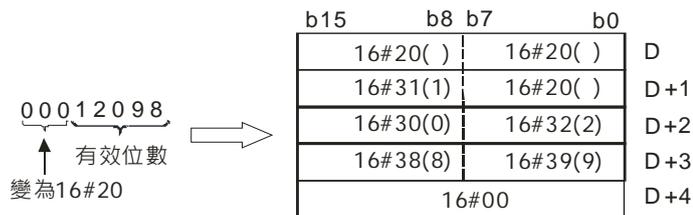
其中 **SM690=OFF** 時，**D+2** 填入字串結束模式 **16#0000**；**SM690=ON** 時，不填入字串結束模式，因此 **D+2** 的值保持不變。

- 32 位元指令中，**S** 的範圍 BCD 0~99999999，且固定為 8 位數的 BCD 值；**D** 連續佔用五個 Word 裝置，資料轉換方式如圖說明：



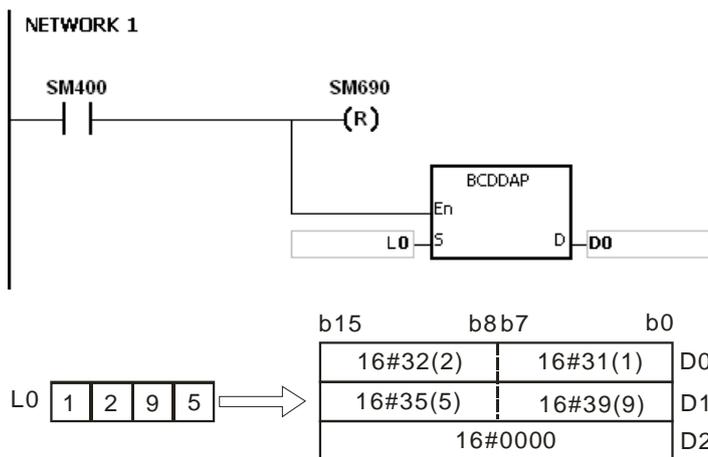
其中 **SM690=OFF** 時，**D+5** 填入字串結束模式 **16#0000**；**SM690=ON** 時，不填入字串結束模式，因此 **D+5** 的值保持不變。

5. 若 S 內容值如果有位數不足，則 D 相對應的存放位址，會被補 16#20，例如：



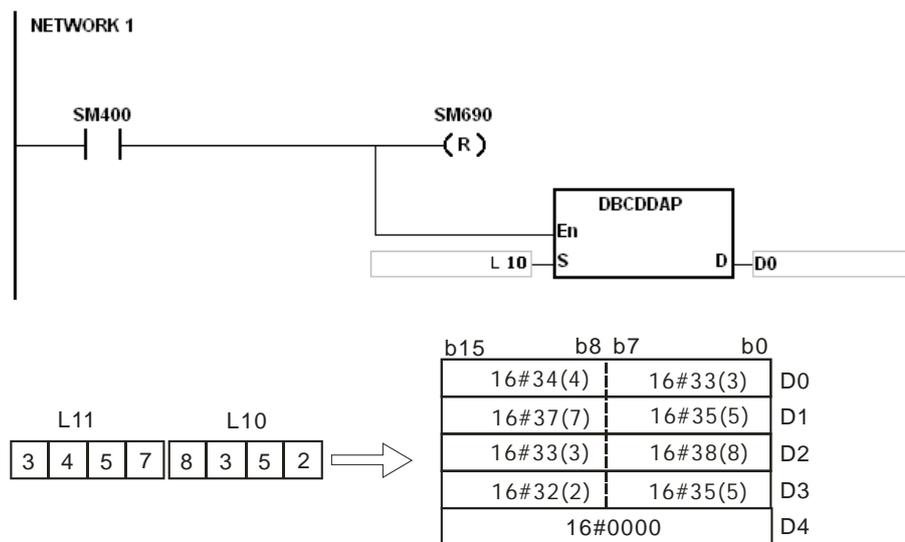
程式範例一：

給定 L0=BCD 1295 且 SM690=OFF，PLC RUN 時 D0=16#3231、D1=16#3539、D2=16#0000。



程式範例二：

給定 L10=BCD 34578352 且 SM690=OFF，PLC RUN 時 D0=16#3433、D1=16#3735、D2=16#3338、D3=16#3235、D4=16#0000。



補充說明：

1. 16 位元指令中，若 **S** 內容值不在 0~9999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D。(BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
2. 32 位元指令中，若 **S** 內容值不在 0~99999999，指令不執行，SM0=ON，錯誤碼 SR0=16#200D。(BCD 值以 Hex 表示有任一位數不在 0~9 的範圍內)。
3. 16 位元指令中 **D+2** 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
4. 32 位元指令中 **D+4** 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
5. 16 位元指令 **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
6. 32 位元指令 **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [5] of WORD/INT。

6

API	指令碼			運算元								功能				
2103	D	DABIN	P	S · D								有號數十進制 ASCII→有號數十進制 BIN 變換				

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

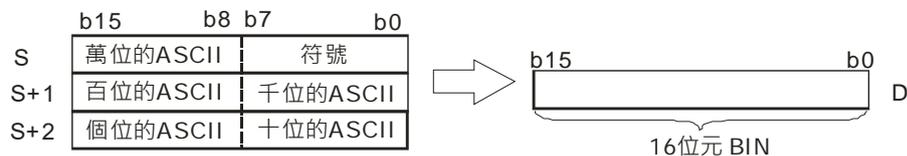
脈波執行型	16 位元指令 (5-11 steps)	32 位元指令 (5-11 steps)
AH500	AH500	AH500

符號：

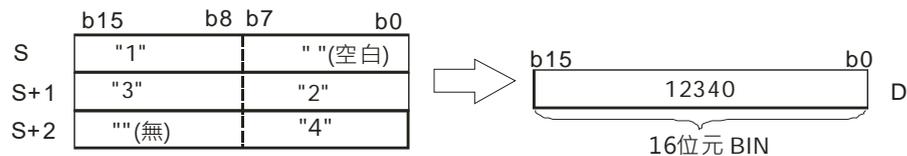
<table border="1" style="width: 100%; text-align: center;"> <tr><td colspan="2">DABIN</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	DABIN		En		S	D	<table border="1" style="width: 100%; text-align: center;"> <tr><td colspan="2">DABINP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	DABINP		En		S	D	S：要轉換的來源值	Word
DABIN															
En															
S	D														
DABINP															
En															
S	D														
<table border="1" style="width: 100%; text-align: center;"> <tr><td colspan="2">DDABIN</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	DDABIN		En		S	D	<table border="1" style="width: 100%; text-align: center;"> <tr><td colspan="2">DDABINP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	DDABINP		En		S	D	D：存放轉換結果的裝置	Word/Double Word
DDABIN															
En															
S	D														
DDABINP															
En															
S	D														

指令說明：

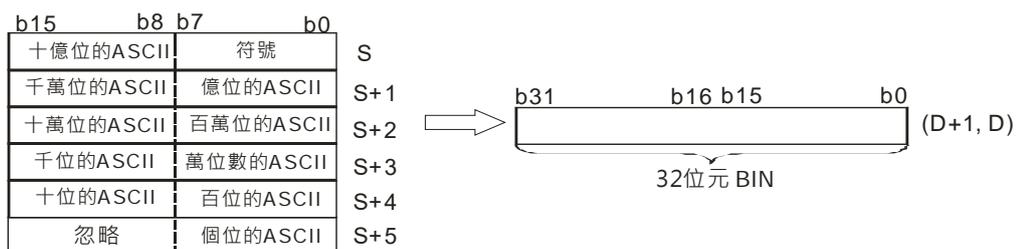
- 將有號十進制 ASCII 值的資料來源 S 轉換成有號數十進制 BIN，並儲存於 D。
- 16 位元指令中，S 連續佔用三個 Word 裝置，且組成的 S 內容值範圍 ASCII-32768~32767；若 S 來源如輸入字串，範圍為“-32768”~“32767”。



- 16 位元指令中，若 S 來源為字串，且不足 6 個字串，則不足處視為“0”；第一個字元用來表示正負號，“ ” (空白) 表示正數，“-”表示負數。如下範例“1234”：

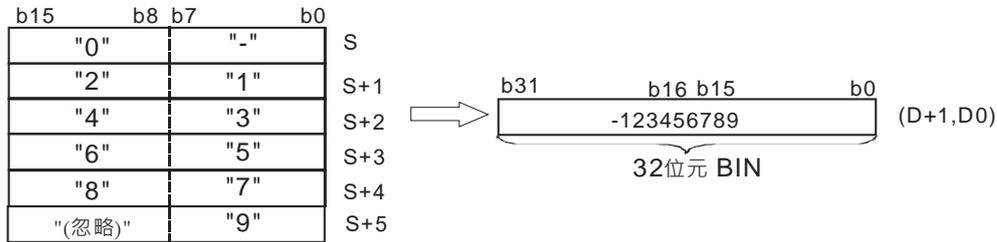


- 32 位元指令中，S 連續佔用六個 Word 裝置，且組成的 S 內容值範圍 ASCII -2147483648~2147483647；若 S 來源如輸入字串，範圍為“-2147483648”~“2147483647”。



- 32 位元指令中，若 **S** 來源為字串，且不足 11 個字串，則不足處視為“0”；第一個字元用來表示正負號，“ ” (空白) 表示正數，“-”表示負數。

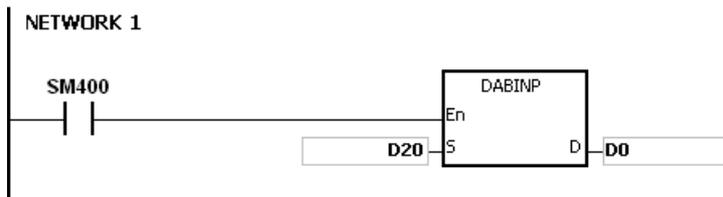
如下範例“-0123456789”：



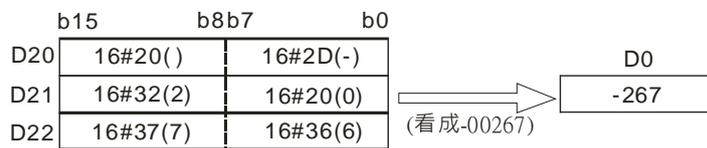
- S** 內容為 16#20 或 16#00 時，將當做 16#30 來做處理。
- 符號為 16#20、16#30 或 16#2B 時，轉換結果為正數，符號為 16#2D 時，轉換結果為負數。
- S** 來源為字串時，16 位元指令長度為 1~6，32 位元指令長度為 1~11。

程式範例一：

給定 D20=16#202D、D21=16#3220、D22=16#3736 (即 ASCII-00267) · PLC RUN 時，D0=-267。

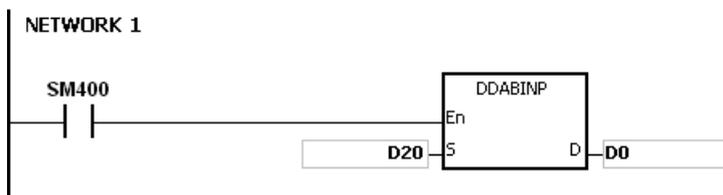


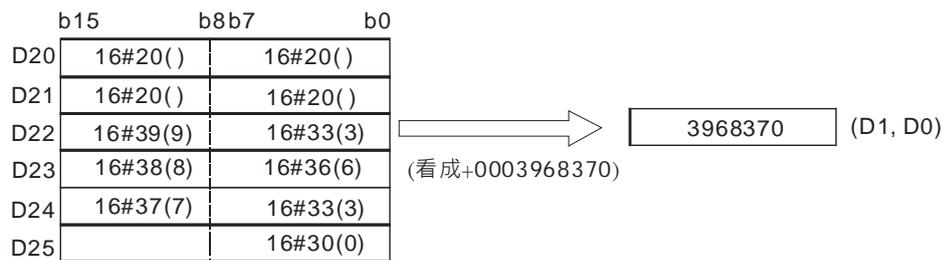
6



程式範例二：

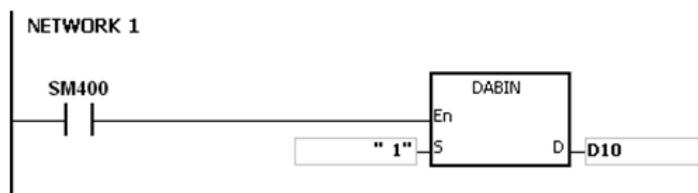
給定 D20=16#2020、D21=16#2020、D22=16#3933、D23=16# 3836、D24=16#3733、D25=16#3330 (即 ASCII 0003968370) · PLC RUN 時，D0=3968370。





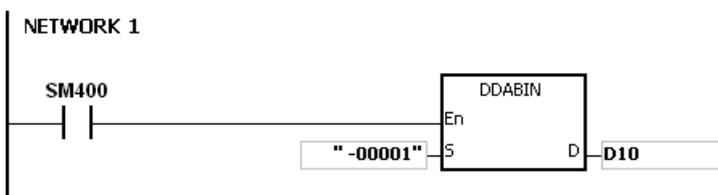
程式範例三：

S 資料來源為“1” (String 型態)，第一個字元需為“ ” (空白) 表示正數，且“1”因為不足 6 個字串，因此視為“10000”，PLC RUN 後 D10=10000。



程式範例四：

S 資料來源為“-00001” (String 型態)，第一個字元為“ ” (空白) 表示正數，且“-00001”因為不足 11 個字串，因此視為“-0000100000”，PLC RUN 後 (D11 · D10) =-100000。



補充說明：

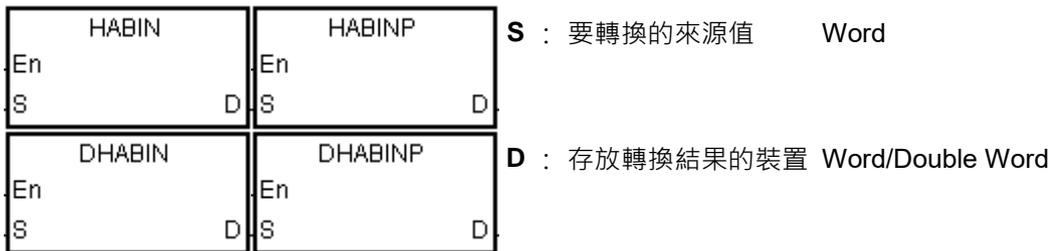
1. 若 S 所指定的符號不是 16#20、16#30、16#2B 或 16#2D，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 若 S 所指定的內容值不是 ASCII 碼 16#30~16#39、16#20 或 16#0，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. S 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
4. 16 位元指令中 S+2 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
5. 32 位元指令中 S+5 裝置，超過裝置範圍時，SM0=ON，錯誤碼 SR0=16#2003。
6. 16 位元指令 S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。
7. 32 位元指令 S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [6] of WORD/INT。

API	指令碼			運算元								功能				
2104	D	HABIN	P	S · D								十六進制 ASCII→十六進制 BIN 變換				

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

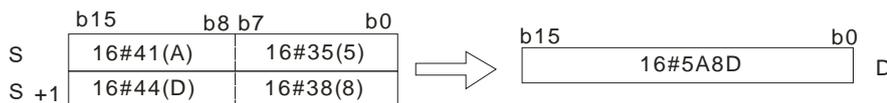
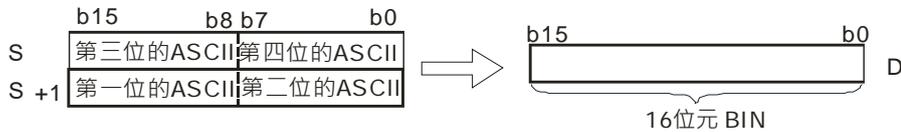
脈波執行型	16 位元指令 (5-11 steps)	32 位元指令 (5-11 steps)
AH500	AH500	AH500

符號：

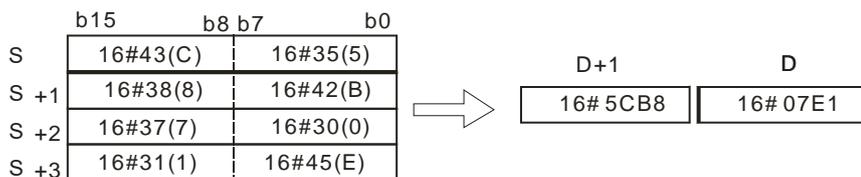
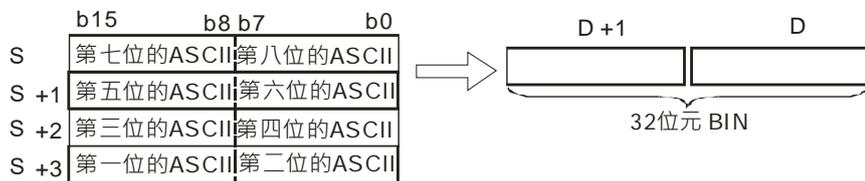


指令說明：

- 資料來源 **S** 的內容十六進制 (ASCII 值) 轉換成十六進制 BIN，存於 **D**。
- 16 位元指令 **S** 連續佔用二個 Word 裝置，S 內容值範圍 ASCII 0000~FFFF；若 **S** 來源如輸入字串，範圍為“0”~“FFFF”。



- 32 位元指令 **S** 連續佔用四個 Word 裝置，S 內容值範圍 ASCII 00000000~FFFFFFFF；若 **S** 來源如輸入字串，範圍為“0”~“FFFFFFFF”。

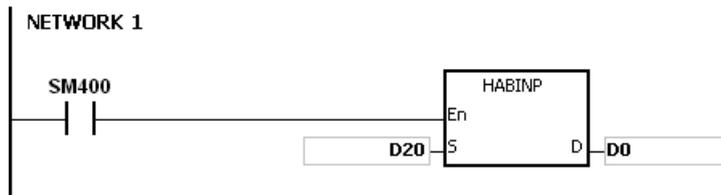


- S** 來源為字串時，16 位元指令長度為 1~4，32 位元指令長度為 1~8。

6

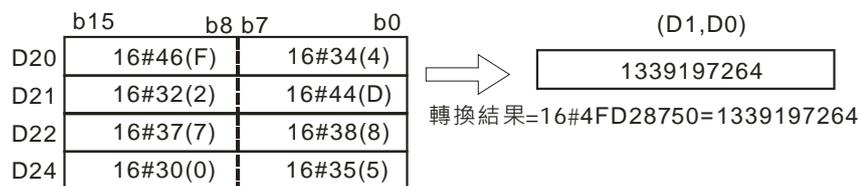
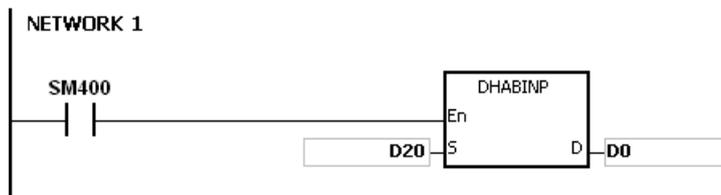
程式範例一：

給定 D20=16#3641、D21=16#4633 (即 ASCII 16#A63F=ASCII -22977)、PLC RUN 時、D0=-22977



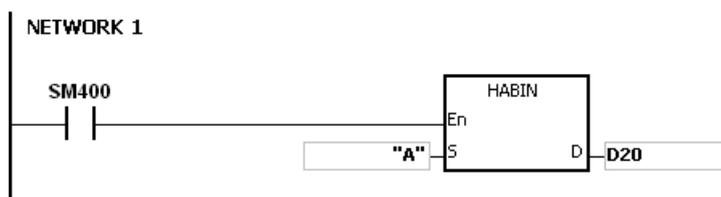
程式範例二：

給定 D20=16#4634、D21=16#3244、D22=16#3738、D23=16#3035 (即 ASCII 16#4FD28750=ASCII 1339197264)、PLC RUN 時、(D1 · D0) =1339197264。



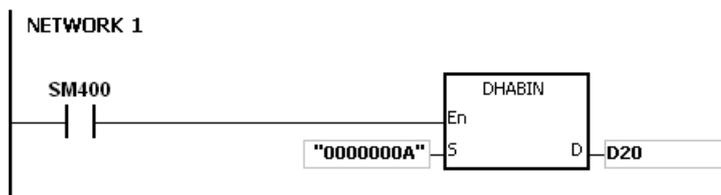
程式範例三：

S 資料來源為“A” (String 型態)、因為不足 4 個字串、因此視為“A000”、PLC RUN 後 D20=16#A000=-24576。



程式範例四：

S 資料來源為“0000000A” (String 型態)、PLC RUN 後 (D21 · D20) =16#0000000A=10。



補充說明：

1. **S** 內容值不是 ASCII 碼 16#30~ 16#39 或 16#41~ 16#46，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. 16 位元指令 **S** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
3. 32 位元指令 **S** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of WORD/INT。

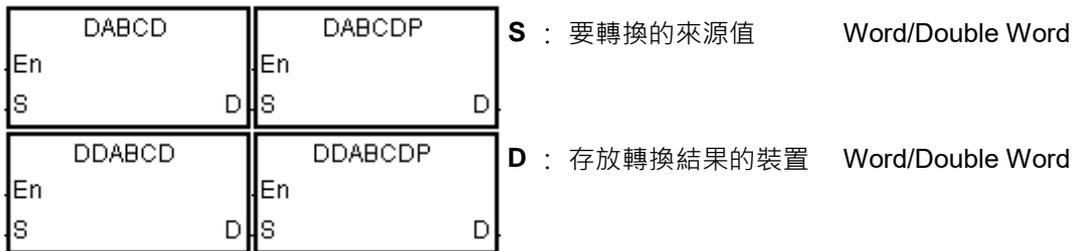
6

API	指令碼			運算元								功能					
2105	D	DABCD	P	S · D								BCD ASCII→BCD 變換					

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

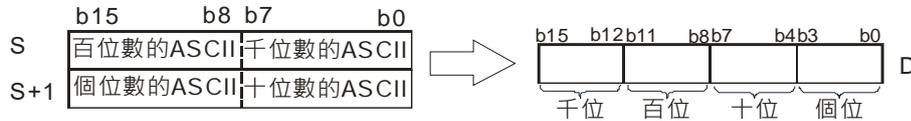
脈波執行型	16 位元指令 (5-11 steps)	32 位元指令 (5-11 steps)
AH500	AH500	AH500

符號：

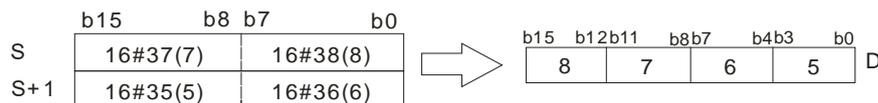


指令說明：

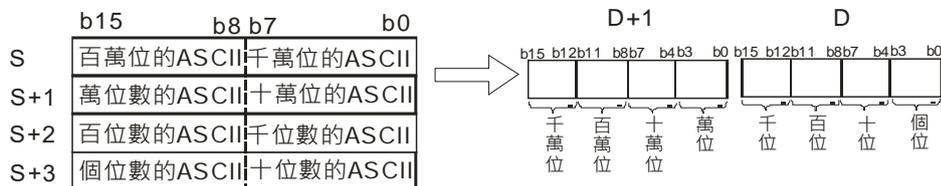
- 資料來源 **S** 的內容 (ASCII 值) 轉換成 BCD · 存於 **D** 。
- 16 位元指令 S** 連續佔用二個 Word 裝置 · S 內容值範圍 ASCII 0000~9999 ; 若 **S** 來源如輸入字串 · 範圍為“0”~“9999”。



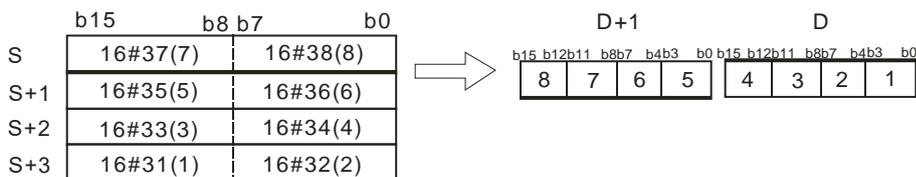
- 例如 · S~S+N 的內容分別為 ASCII 碼 8765 · 則轉換結果如下圖所示。



- 32 位元指令 S** 連續佔用四個 Word 裝置 · S 內容值範圍 ASCII 0000000~99999999 ; 若 **S** 來源如輸入字串 · 範圍為“0”~“99999999”。



例如 · S~S+N 的內容分別為 ASCII 碼 87654321 · 則轉換結果如下圖所示。

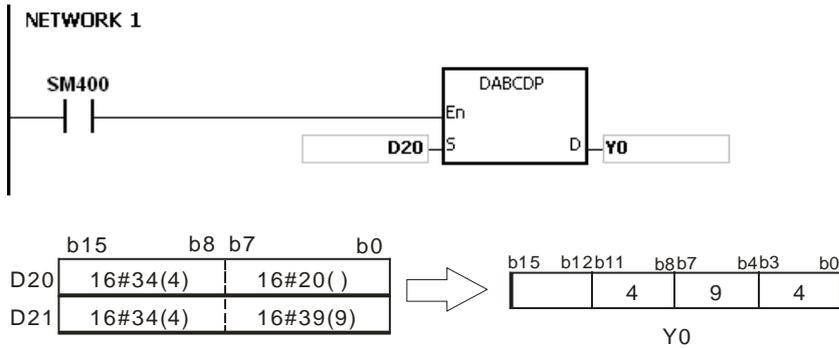


- S** 內容為 16#20 或 16#00 時 · 將當做 16#30 來做處理。

6. S 來源為字串時，16 位元指令長度為 1~4，32 位元指令長度為 1~8。

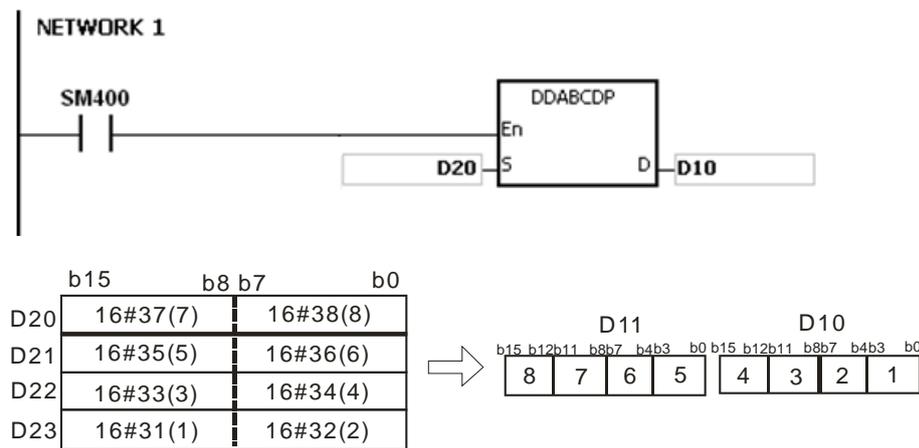
程式範例一：

給定 D20=16#3420、D21=16#3439 (即 ASCII 494)，PLC RUN 時，轉換為 Y0=16#494。



程式範例二：

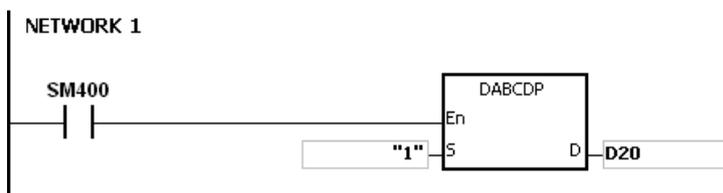
給定 D20=16#3738、D21=16#3536、D22=16#3334、D23=16#3132 (即 ASCII 87654321)，PLC RUN 時，轉換為 (D11 · D10) = 16#87654321。



6

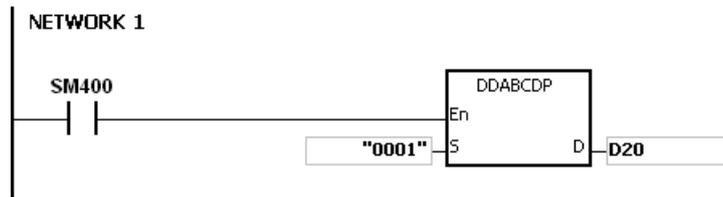
程式範例三：

S 資料來源為“1” (String 型態)，因為不足 4 個字串，因此視為“1000”，PLC RUN 時，轉換為 D20=16#1000。



程式範例四：

S 資料來源為“0001” (String 型態)，因為不足 8 個字串，因此視為“00010000”，PLC RUN 時，轉換為 (D21 · D20) =16#10000。



補充說明：

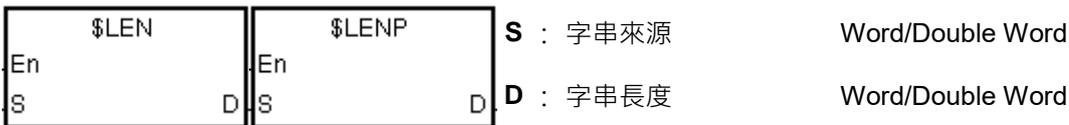
1. S 內容值不是 ASCII 碼 16#30~16#39 指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. S 字串長度超出範圍時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. 16 位元指令 S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
4. 32 位元指令 S 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of WORD/INT。

API	指令碼			運算元								功能				
2106		\$LEN	P	S · D								計算字串長度				

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

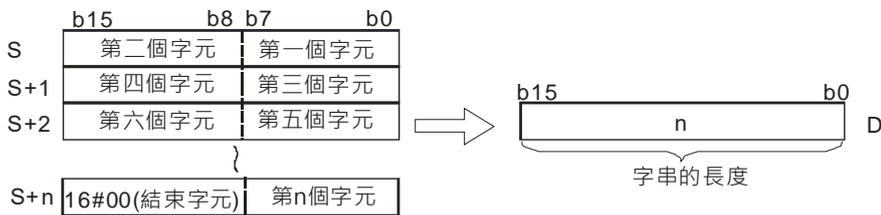
脈波執行型	16 位元指令 (5-11 steps)	32 位元指令
AH500	AH500	-

符號：

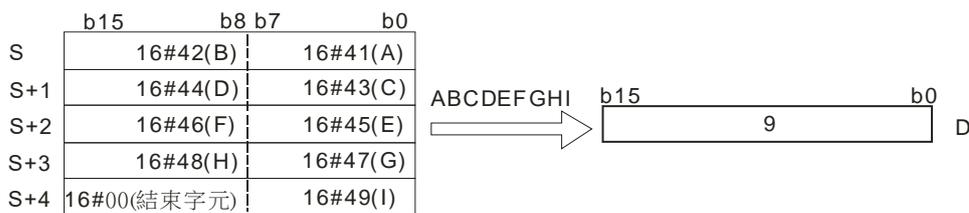


指令說明：

1. 計算資料來源 **S** 的內容 (字串) 長度，一直計算到字串結尾 16#00 之前的長度，並將字串長度存於 **D**。
2. 字串長度=0~65535。
 如果字串長度=65536=16#10000，則 D=0。
 如果字串長度=65537=16#10001，則 D=1。

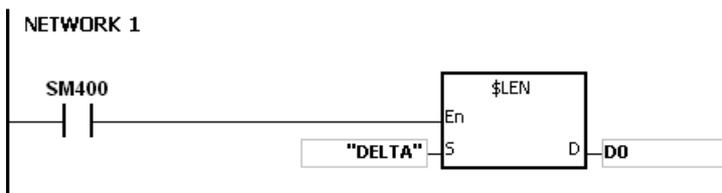


例如 S~S+n 的內容為 ABCDEFGHI，則計算結果如下圖所示。



程式範例一：

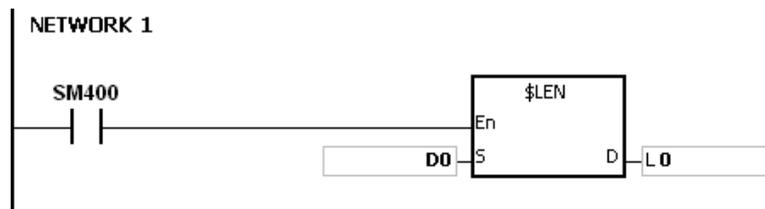
S 資料來源給定 'DELTA'，則 PLC 執行後 D0=5。



程式範例二：

S 資料來源給定 D0~D2 如下表，則 PLC 執行後 L0=5。

D0	16#45 (E)	16#44 (D)
D1	16#54 (T)	16#4C (L)
D2	16#00 (結束字元)	16#41 (A)



補充說明：

字串的字尾無 16#00 當做結束符號，指令不執行，SM0=ON，錯誤碼 SR0=16#200E。

API	指令碼			運算元							功能						
2107	D	\$STR	P	$S_1 \cdot S_2 \cdot D$							BIN→String						

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令 (7 steps)
AH500	AH500	AH500

符號：

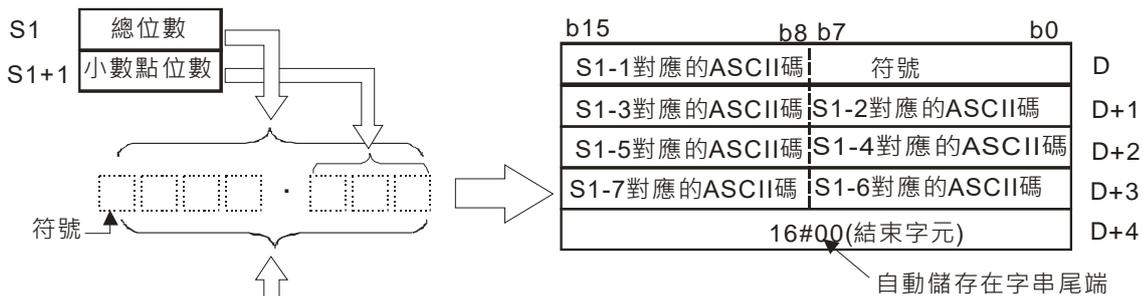
En	\$STR	En	\$STRP	D	S1	S2	D												
S1		S1																	
S2		S2																	

S_1 : 要轉換的位數裝置起始編號 Word/Double Word
 S_2 : 要被轉換的來源 Word/Double Word
D : 存放轉換結果的裝置起始編號 Word

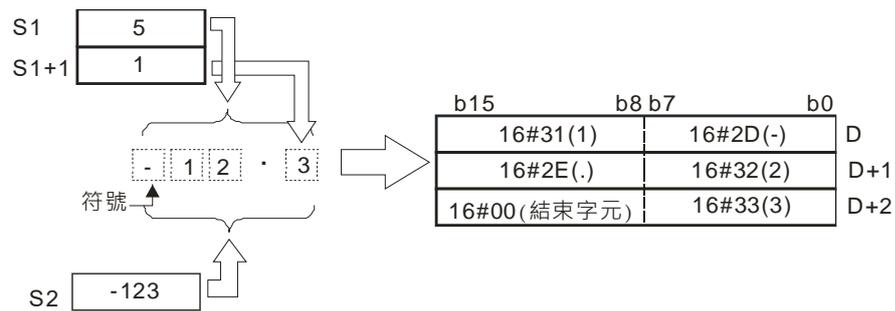
指令說明：

- S_2 的內容加上一個小數點，小數點後的位數由 S_{1+1} 所指定，並將轉換後的結果依照 S_1 所指定的總長度存於 D。
- \$STR** :
 S_1 總長度範圍=2~8。
 S_{1+1} 所指定的小數點後的位數，範圍=0~5，而且 S_{1+1} 必須 $\leq S_1-3$ 。
 S_2 的範圍=-32768~32767。

6



例如總位數為 5，小數點位數為 1，內容為-123，轉換後的結果如下圖所示。

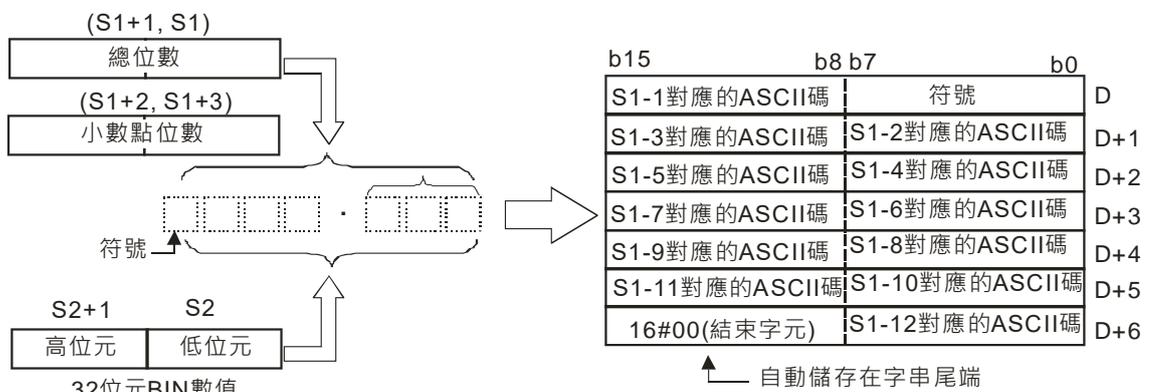


3. **D\$STR** :

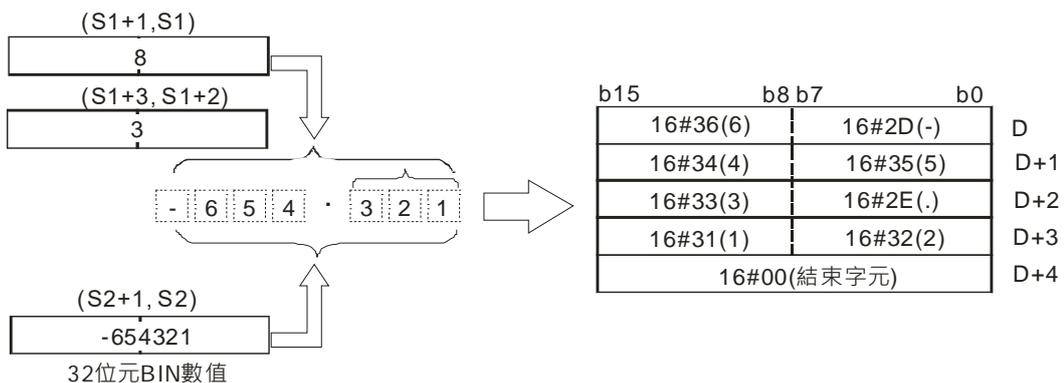
S_1 總長度範圍=2~13。

S_1+1 所指定的小數點後的位數，範圍=0~10，而且 S_1+1 必須 $\leq S_1-3$ 。

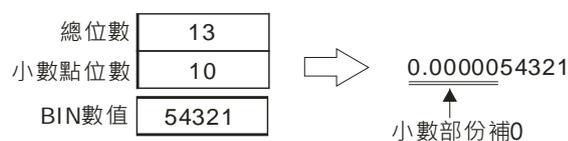
S_2 的範圍=-2147483648~2147483647。



例如總位數為 8，小數點位數為 3，內容為-654321，轉換後的結果如下圖所示。

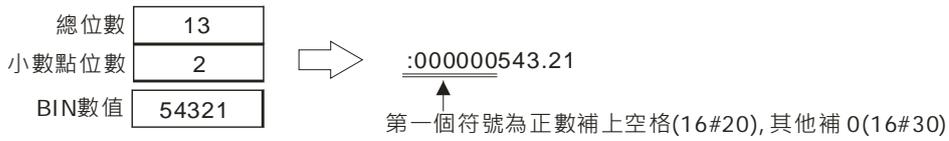


4. 如果 S_2 為正數，則 D 的符號位置被填入 16#20，為負數，填入 16#2D。
5. D 的小數點位置被填入 16#2E。
6. 如果 S_1+1 的位數 $> S_2$ 的位數時，空出的位數自動補 0。

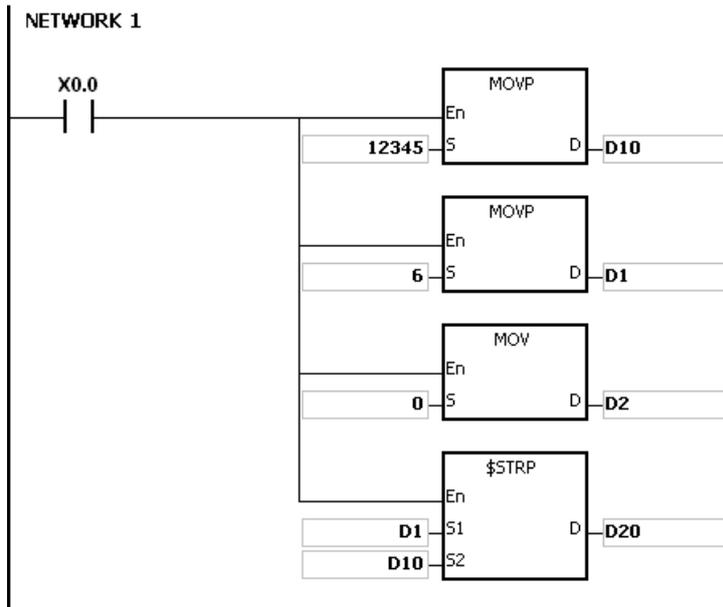


6

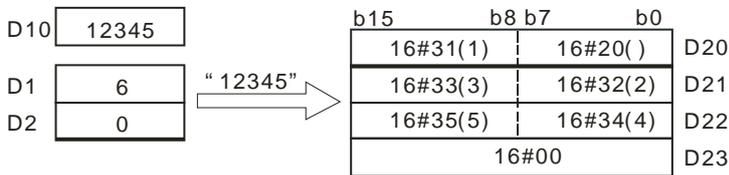
7. 如果 S_1 總位數 $>$ S_2 加上小數點跟符號的位數時，空出的位數自動補上‘0’。



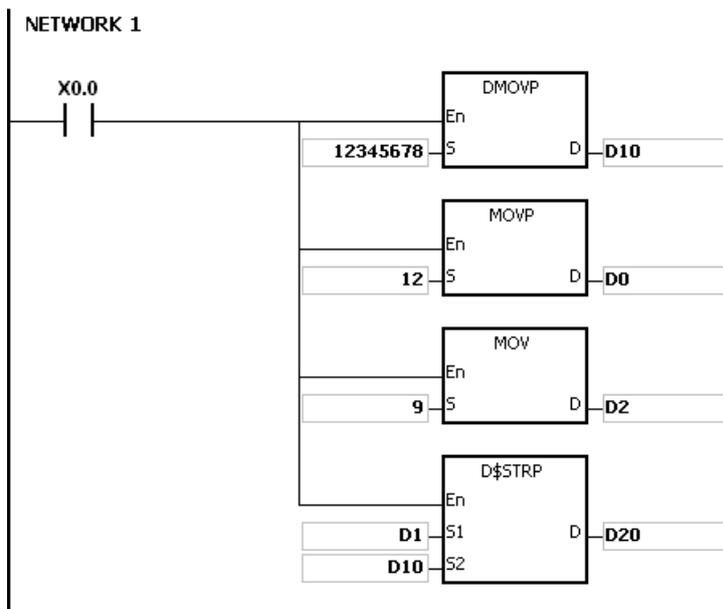
程式範例一：

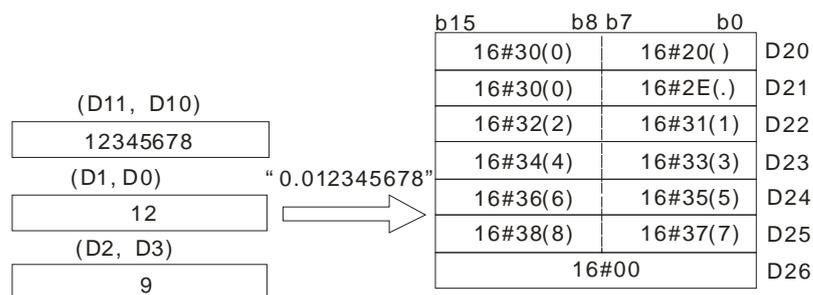


6



程式範例二：





補充說明：

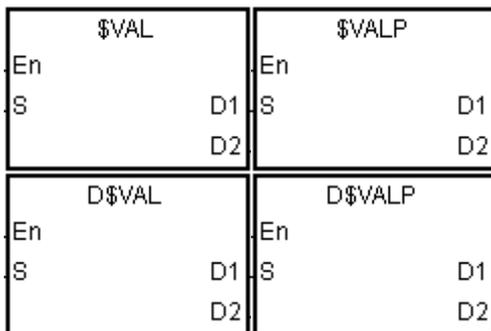
1. S_1 總長度超出範圍時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. S_{1+1} 位數超出範圍時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. S_{1+1} 必須 $\leq S_1-3$ ，不符合此範圍時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
4. S_1 總位數 $< S_2$ 加上小數點跟符號的位數時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
5. 16 位元指令 S_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
6. 32 位元指令 S_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of DWORD/DINT。

API	指令碼			運算元								功能				
2108	D	\$VAL	P	S · D ₁ · D ₂								String→BIN				

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

脈波執行型	16 位元指令 (7-13 steps)	32 位元指令 (7-13 steps)
AH500	AH500	AH500

符號：



S : 要轉換的來源值 Word
 D₁ : 存放轉換後的位數 Word/Double Word
 D₂ : 存放轉換後的 BIN 值 Word/Double Word

指令說明：

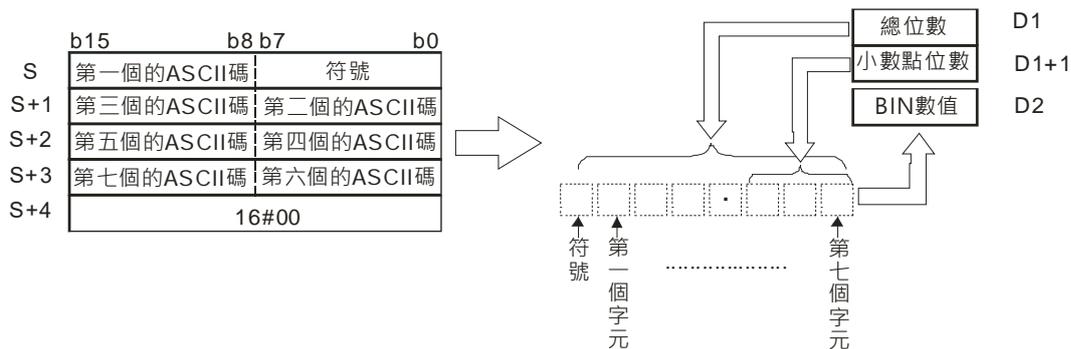
1. S 的內容字串，轉換成 BIN 值，並將總位數存放在 D₁，小數點後有幾位數存放在 D₁+1，BIN 值存放在 D₂。

2. \$VAL：

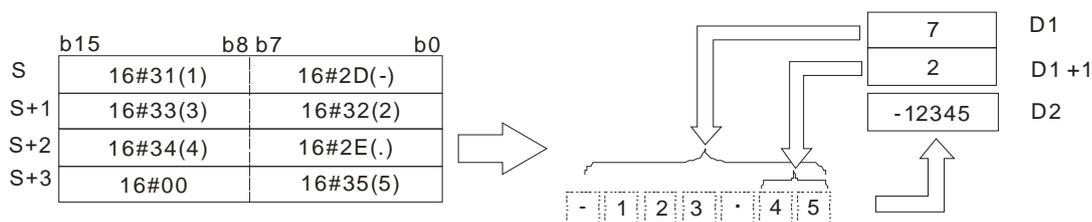
S 最多佔用五個 Word 裝置。

S 字串總長度範圍=2~8。

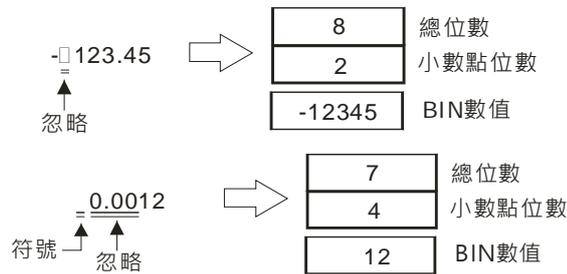
如果 S 字串中有小數點，則小數點 16#2E 的位置，需在 3~ (字串總長度-1) 之中。



例如 S~S+3 的內容分別為-123.45，則運算結果如下圖所示。



S 字串在符號與第一個非零數值之間，如果有 16#20 或 16#30，則在轉換成 BIN 數值時，會忽略 16#20 或 16#30。



忽略小數點的點數 (16#2E) 不看時，則 **S** 字串的範圍=-32768~32767。例如：“1235.3”就是要檢查“12353”是否在 **S** 的範圍內。

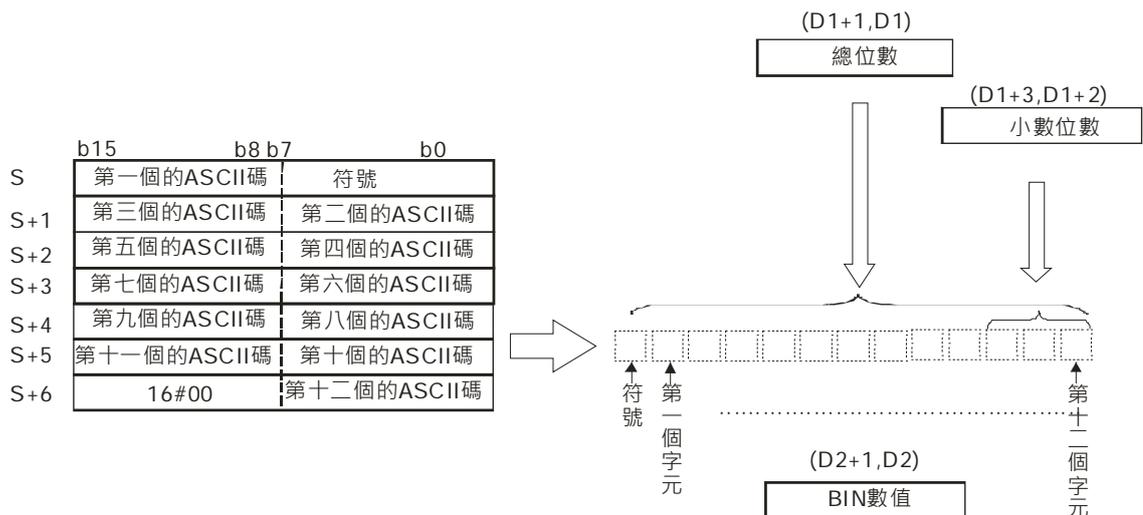
3. **D\$VAL** :

S 最多佔用七個 Word 裝置。

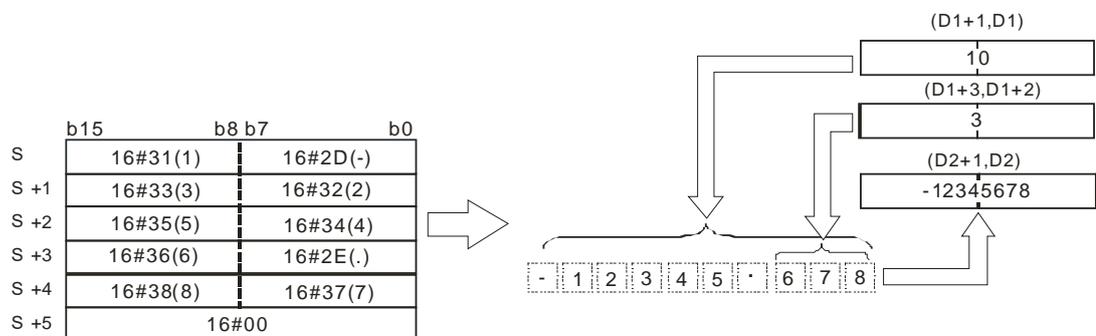
字串總長度範圍=2~13。

如果 **S** 字串中有小數點，則小數點 16#2E 的位置，需在 3~ (字串總長度-1) 之中。

忽略小數點的點 (16#2E) 不看時，則 **S** 字串的範圍=-2147483648~2147483647。

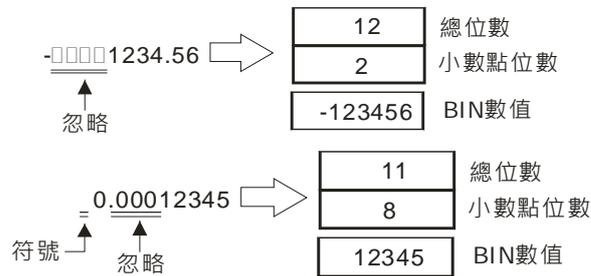


例如 **S~S+n** 的內容分別為-12345.678，則運算結果如下圖所示。



6

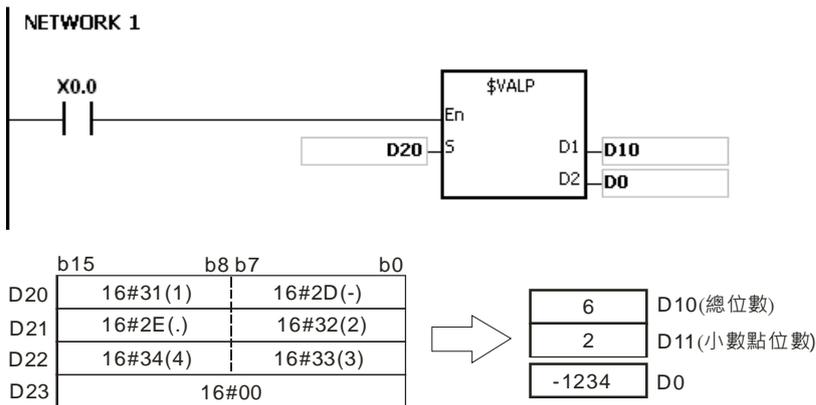
S 字串在符號與第一個非零數值之間，如果有 16#20 或 16#30，則在轉換成 BIN 數值時，會忽略 16#20 或 16#30。



忽略小數點的點 (16#2E) 不看時，則 S 字串的範圍=-2147483648~2147483647。例如：“1234567.8”就是要檢查“12345678”是否在 S 的範圍內。

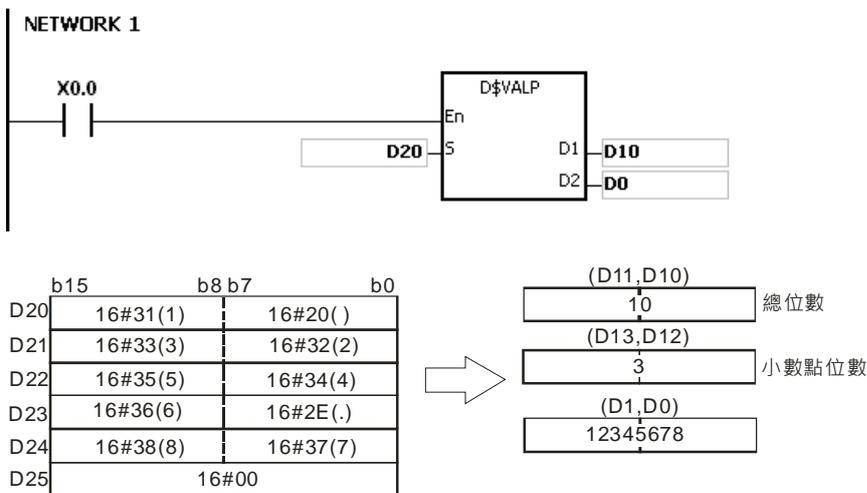
4. 如果 S 的符號位置被填入 16#20、16#2B、16#30 則轉換結果為正數，填入 16#2D，則轉換結果為負數。
5. S 字串除了符號、小數點，還有可能會被忽略的部份可以填入 16#20 或 16#30，其餘只能填入 16#30~16#39。

程式範例一：



6

程式範例二：



補充說明：

1. **S** 字串長度超出範圍時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. **S** 所指定的符號不是 16#20、16#30、16#2B 或 16#2D，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. **S** 字串中的小數點符號位置不正確，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
4. **S** 字串轉換後的 BIN 值超出範圍，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
5. **S** 字串除了符號、小數點，還有可能會被忽略的部份可以填入 16#20 或 16#30，其餘要轉成 BIN 值的部份不在 16#30~16#39 範圍內，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
6. 16 位元指令 **D₁** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
7. 32 位元指令 **D₁** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of DWORD/DINT。

API	指令碼			運算元							功能						
2109		\$FSTR	P	$S_1 \cdot S_2 \cdot D$							Float→String						

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

脈波執行型	16 位元指令 (7-8 steps)	32 位元指令
AH500	AH500	-

符號：

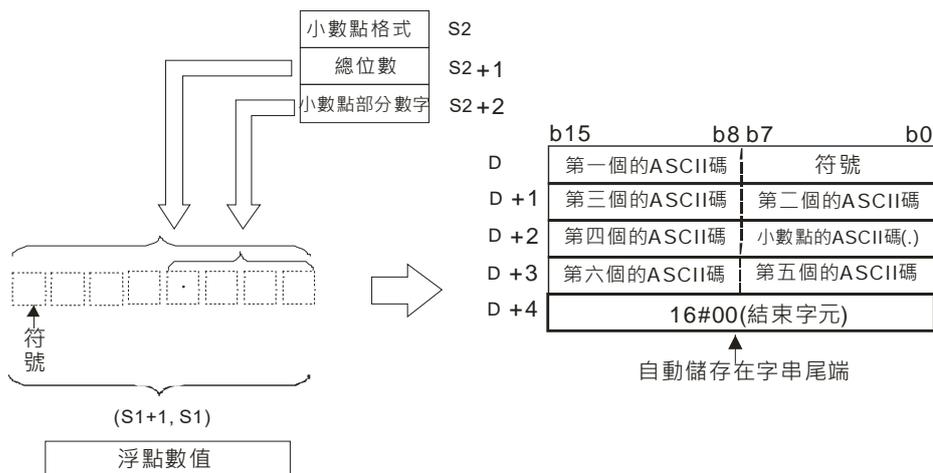
\$FSTR		\$FSTRP		S_1 ：	要被轉換的來源浮點數值	Double Word
En		En		S_2 ：	要轉換的顯示設定裝置起始編號	Word
S1		S1	D	D：	存放轉換結果的裝置起始編號	Word
S2		S2				

指令說明：

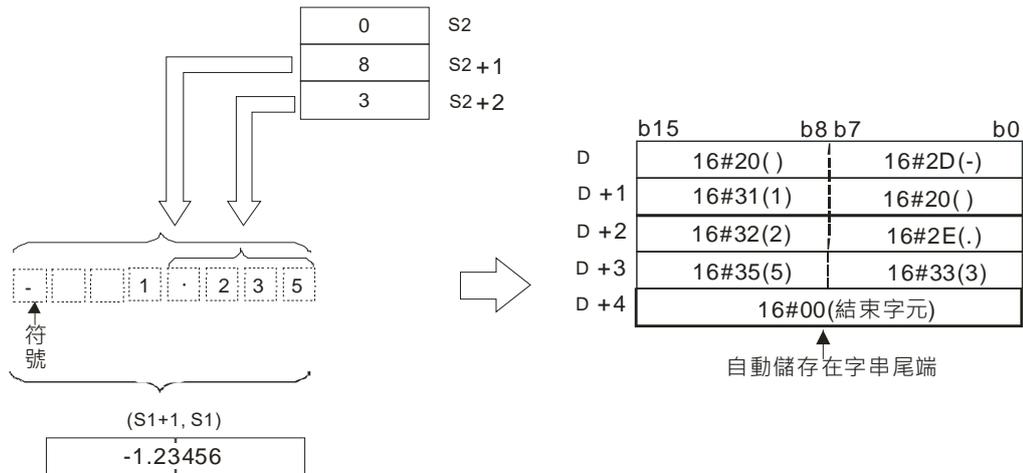
1. 依據 S_2 的顯示設定，將 S_1 的浮點值轉換成字串，並將結果存於 D。
2. 轉換的結果依據 S_2 的顯示設定而有所不同。
3. 總位數 S_{2+1} 設定範圍為 2~24。

S_2	0: 小數格式 1: 指數格式
S_{2+1}	總位數
S_{2+2}	小數部分位數

4. 當使用小數格式 ($S_2=0$) 時：

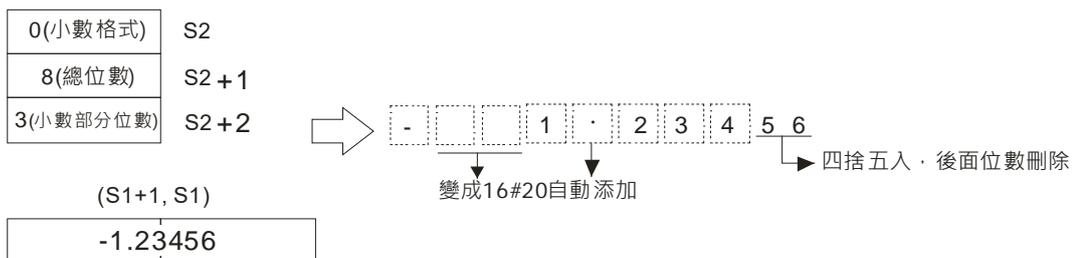


舉例說明，總位數 8 位，小數點位數 3 位，數值為-1.23456，則運算結果如下所示：

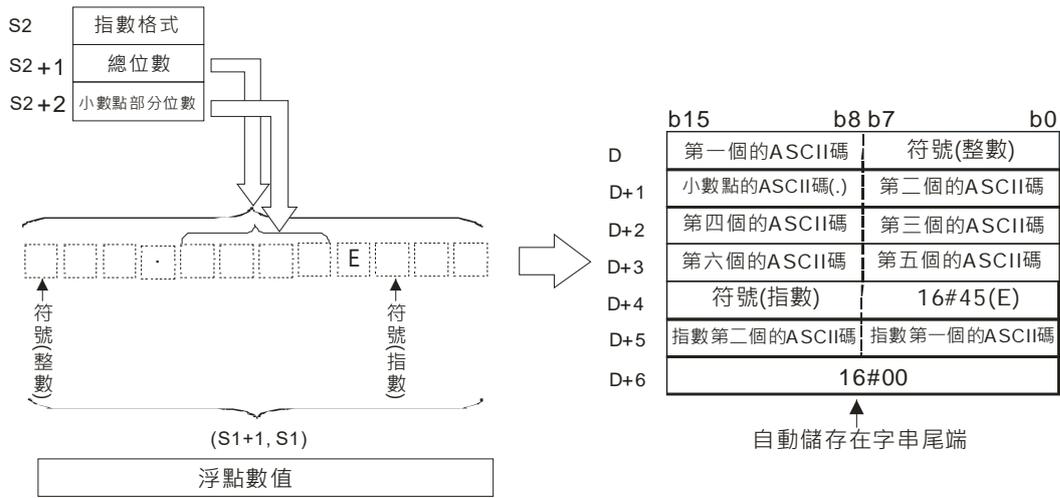


- 總位數 S_{2+1} 可以指定的範圍如下：
 當小數點後的位數 $S_{2+2}=0$ 時， $2 \leq S_{2+1} \leq 24$ ，整數部分的位數 ≤ 23 。
 當小數點後的位數 S_{2+2} 不等於 0 時， $S_{2+2}+3 \leq S_{2+1} \leq 24$ ，整數部分的位數 $\leq 22-S_{2+2}$ 。
- 小數點後的位數 S_{2+2} 可以指定的範圍為：0~7，但是 S_{2+2} 不等於 0 時， S_{2+2} 必需要 $\leq (S_{2+1} \text{ 總位數}) - 3$ 。
- 如果來源浮點數 S_1 為正數，則 D 的符號位置被填入 16#20，為負數，填入 16#2D。
- 如果浮點數的長度超出總位數 S_{2+1} ，則低位元的數字會被刪除，並做四捨五入。
- 如果小數點後的位數 $S_{2+2} > 0$ ，則 16#2E (".")，會自動存放在指定的位數之前。
- 轉換後的結果，如果長度小於總位數，則從符號之後到浮點數實數之前的位數會被填入 16#20。
- 在轉換結果的字串最後面，將自動加上結束字元 16#00。

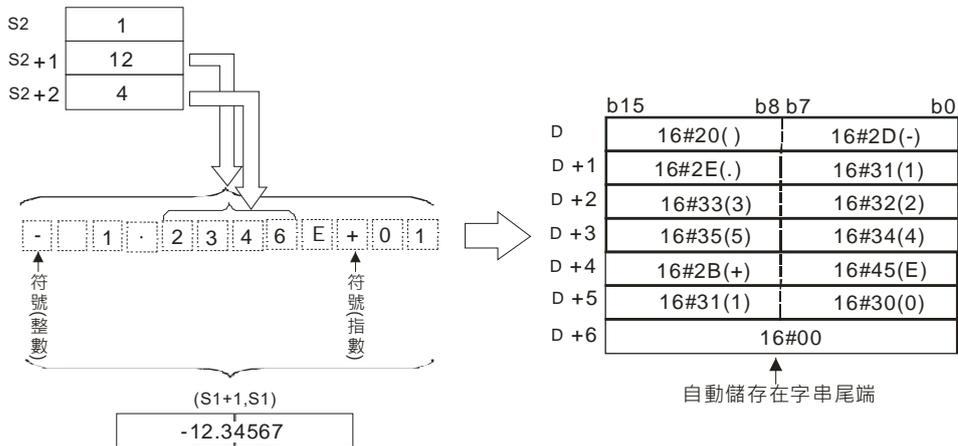
6



5. 當使用指數格式 ($S_2=1$) 時 :

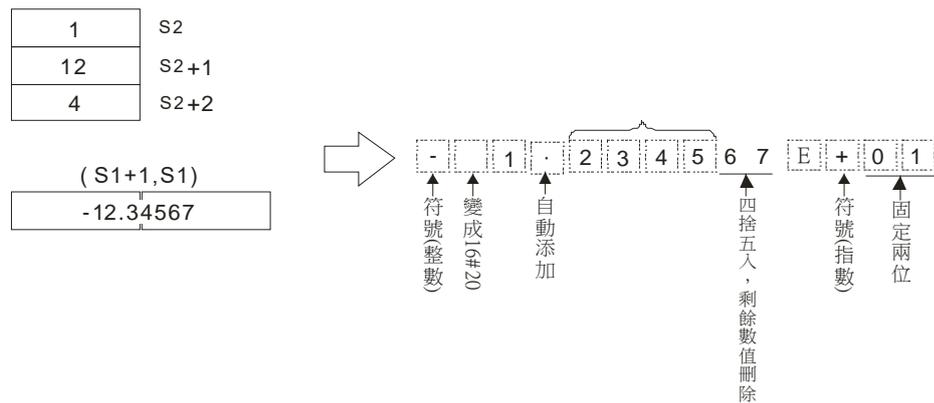


舉例說明·總位數 12 位·小數點位數 4 位·數值為-12.34567·則運算結果如下所示 :



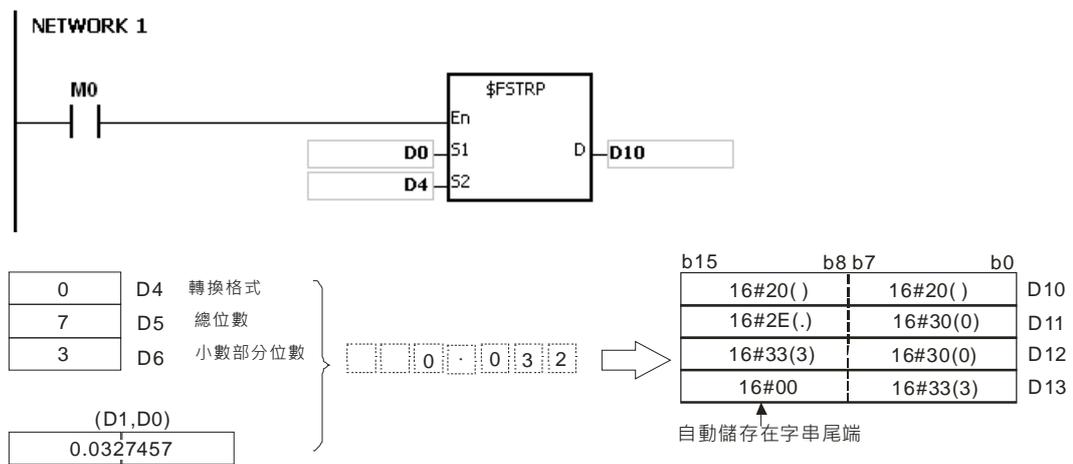
6

- 總位數 S_{2+1} 可以指定的範圍如下 :
 當小數點後的位數 $S_{2+2}=0$ 時 , $6 \leq S_{2+1} \leq 24$ 。
 當小數點後的位數 S_{2+2} 不等於 0 時 , $S_{2+2}+7 \leq S_{2+1} \leq 24$ 。
- 小數點後的位數 S_{2+2} 可以指定的範圍為 : 0~7 , 但是 S_{2+2} 不等於 0 時 , S_{2+2} 必需要 $\leq (S_{2+1} \text{ 總位數 }) - 7$ 。
- 如果來源浮點數 S_1 為正數 , 則 D 的符號位置被填入 16#20 , 為負數 , 填入 16#2D 。
- 整數部份固定為 1 個位數 , 並且在符號與整數之間自動填入 16#20 , 以補足總位數 。
- 如果小數點後的位數 $S_{2+2} > 0$, 則 16#2E (".") , 會自動存放在指定的位數之前 。
- 如果指數值為正數 , 則 D 的指數符號位置被填入 16#2B , 為負數 , 填入 16#2D 。
- 指數部份固定為 2 位數 , 如果只有 1 位數 , 則被填入 16#30 ("0") 。
- 在轉換結果的字串最後面 , 將自動加上 16#00 當做字串的結尾 。



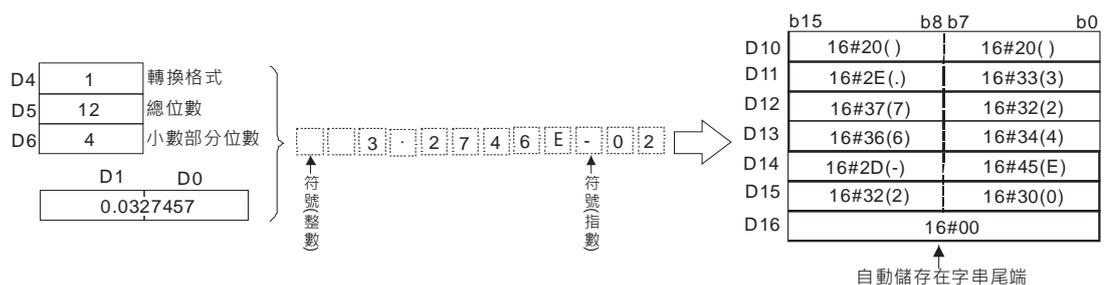
程式範例一：

將浮點值內容 (D1 · D0) 轉為小數形式 (D4=0) 的字串。



程式範例二：

將浮點值內容 (D1 · D0) 轉為指數形式 (D4=1) 的字串。



補充說明：

1. 當 S₁ 超出浮點數可以表示的範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2013。
2. 當 S₂ 符號所指定的值不是 0 也不是 1，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. 當總位數 S₂₊₁ 超出以下範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
 - 當使用小數格式時
 - 當小數點後的位數 S₂₊₂=0 時，2 ≤ S₂₊₁ ≤ 24，整數部分的位數 ≤ 23。
 - 當小數點後的位數 S₂₊₂ 不等於 0 時，S₂₊₂+3 ≤ S₂₊₁ ≤ 24，整數部分的位數 ≤ 22-S₂₊₂。

- 當使用指數格式時
當小數點後的位數 $S_{2+2}=0$ 時， $6 \leq S_{2+1} \leq 24$ 。
當小數點後的位數 S_{2+2} 不等於 0 時， $S_{2+2}+7 \leq S_{2+1} \leq 24$ 。
4. 當小數點後的位數 S_{2+2} 超出以下範圍時，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
- 當使用小數格式時：
0~7，但是 S_{2+2} 必需要 $\leq (S_{2+1} \text{ 總位數}) - 3$ 。
 - 當使用指數格式時：
0~7，但是 S_{2+2} 必需要 $\leq (S_{2+1} \text{ 總位數}) - 7$ 。
5. S_2 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [3] of WORD/INT。

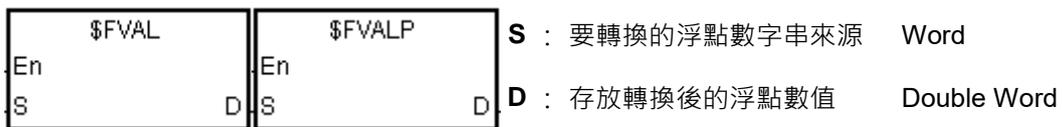
6

API	指令碼		運算元								功能				
2110	\$FVAL	P	S · D								String→Float				

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

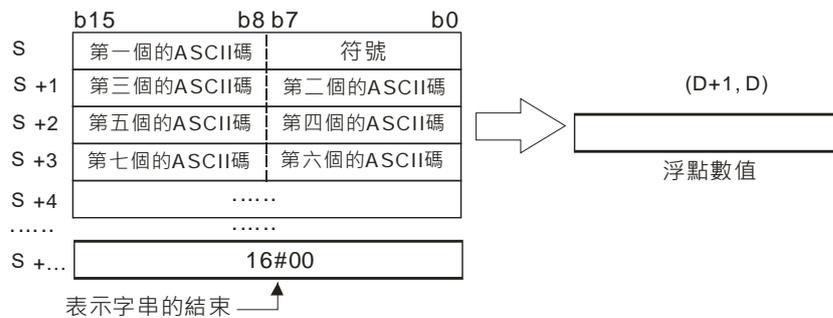
脈波執行型	16 位元指令 (5-11 steps)	32 位元指令
AH500	AH500	-

符號：



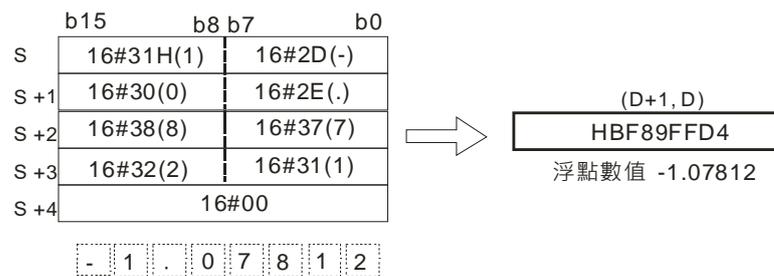
指令說明：

- 將 **S** 的浮點數字串轉換成浮點值，並將結果存於 **D**。

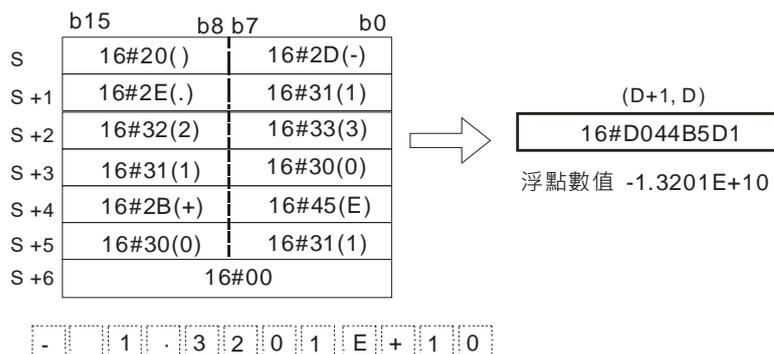


- S** 的浮點數字串分為小數格式跟指數格式兩種，格式會由主機自動根據 **S** 的輸入判別是哪一種。

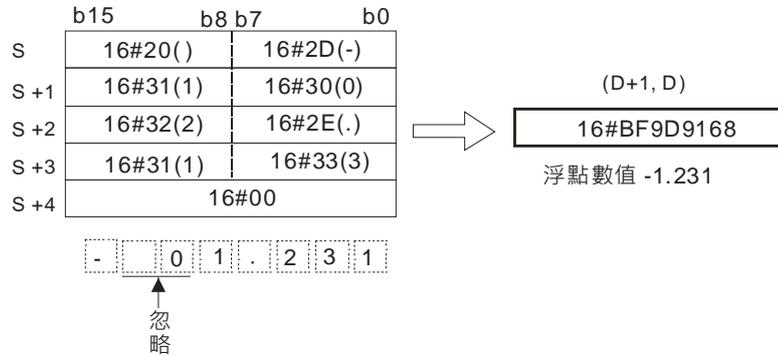
- 當使用小數格式時，範例如下：



- 當使用指數格式時，範例如下：

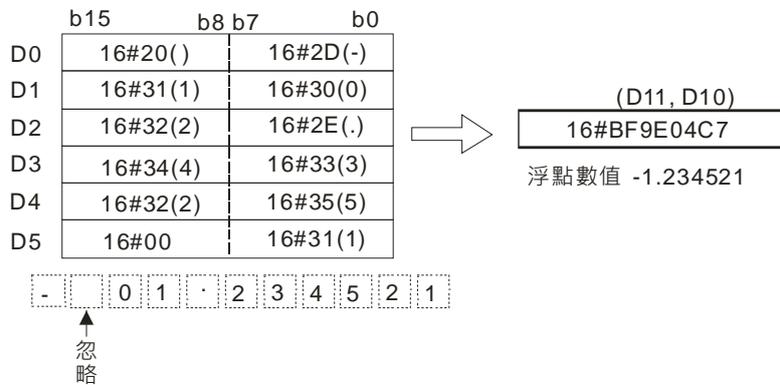
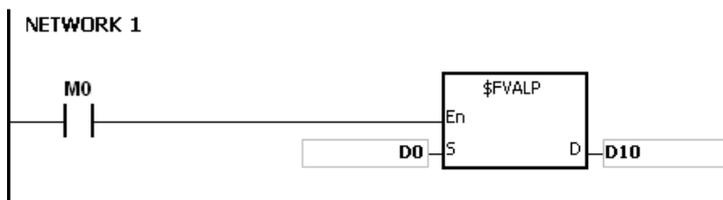


3. 如果 **S** 的符號位置被填入 16#20、16#30 或 16#2B，則轉換結果為正數，填入 16#2D，則轉換結果為負數。
4. **S** 的字串中除了起始的零之外，其餘的 16#20 和 16#30 在轉換時全部忽略。

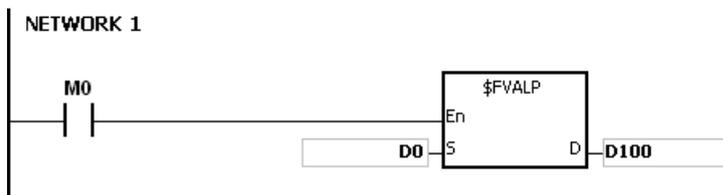


5. **S** 的浮點數字串長度，最長為 24 個字。

程式範例一：



程式範例二：



6

API	指令碼			運算元							功能						
2111		\$RIGHT	P	S · n · D							從右邊擷取字串						

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

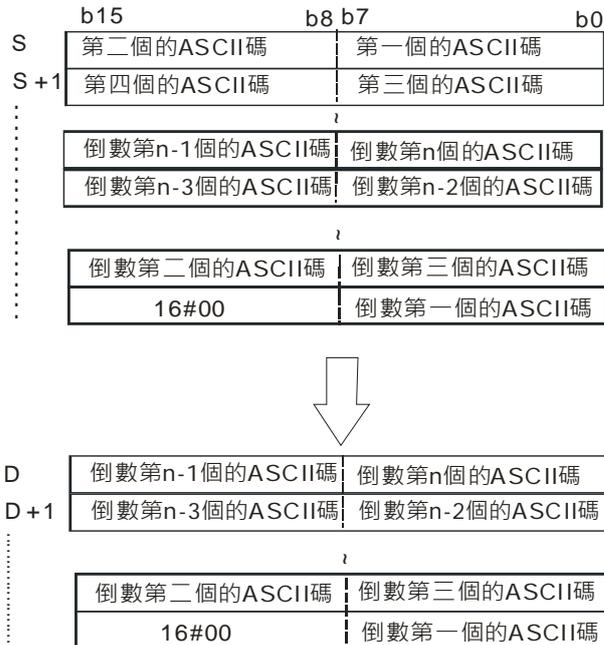
脈波執行型	16 位元指令 (7-13 steps)	32 位元指令
AH500	AH500	-

符號：

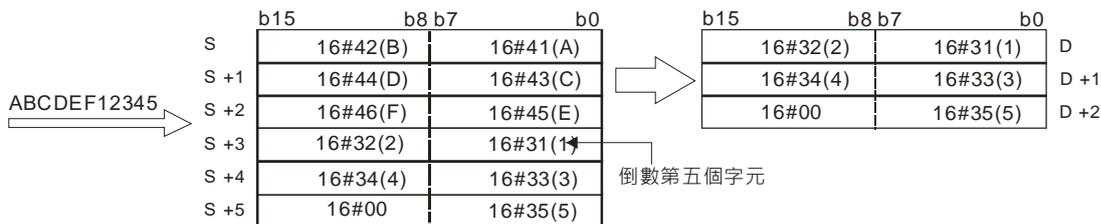
\$RIGHT	\$RIGHTP	S : 字串來源 Word n : 要擷取的長度 Word D : 存放擷取到的字串 Word
En	En	
S	S	

指令說明：

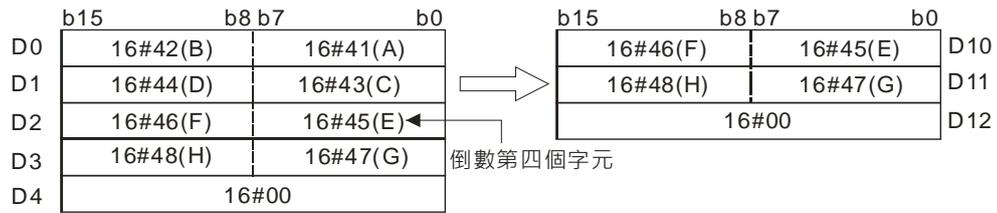
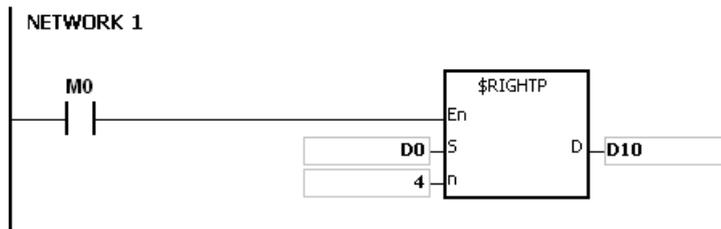
1. 從 **S** 字串的最右邊開始擷取 n 個字，並將擷取出來的字串存放到 **D**。
2. 當 n=0 時，**D** 裝置會存放 0。



例如，S 的內容為 ABCDEF12345，當 n=5 時從最右邊開始擷取 5 個字元，轉換結果如下圖所示。



程式範例：



補充說明：

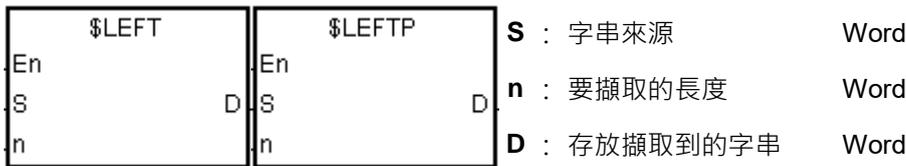
1. **S** 字串中沒有 16#00 當結尾，SM0=ON，SR0 記錄錯誤碼 16#200E。
2. **n**<0 或 **n**>**S** 的字串長度時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. **D** 不足 **n** 填入字串長度時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。

API	指令碼			運算元								功能				
2112	\$LEFT	P		S · n · D								從左邊擷取字串				

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

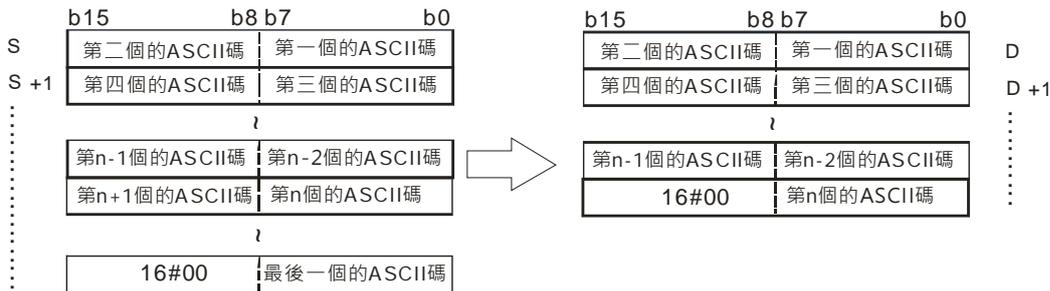
脈波執行型	16 位元指令 (7 -13 steps)	32 位元指令
AH500	AH500	-

符號：

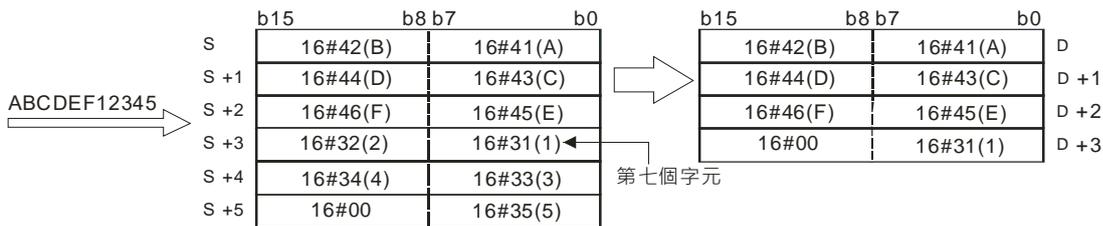


指令說明：

1. 從 **S** 字串的最左邊開始擷取 n 個字，並將擷取出來的字串存放到 **D**。
2. 當 n=0 時，**D** 裝置會存放 0。

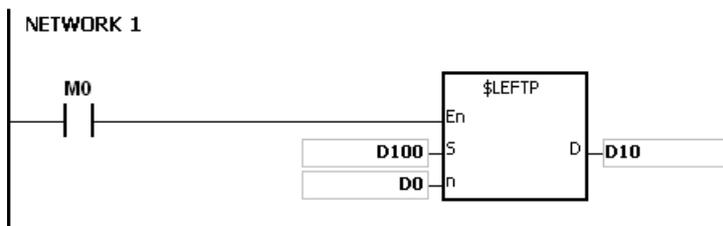


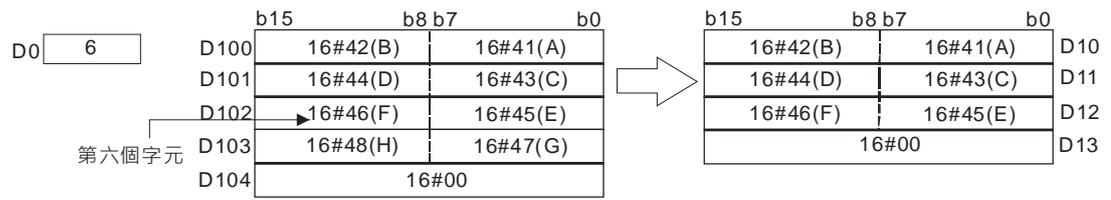
例如，S=ABCDEF12345，當 n=7 時從最左邊開始擷取 7 個字元，轉換結果如下圖所示。



程式範例：

M0=ON 時，執行\$LEFT 指令，從 D100 開始的資料開始擷取，取 6 個字元後將結果儲存至 D10。





補充說明：

1. **S** 字串中沒有 16#00 當結尾，SM0=ON，SR0 記錄錯誤碼 16#200E。
2. **n**<0 或 **n**>**S** 的字串長度時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. **D** 不足 **n** 填入字串長度時，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。

API	指令碼			運算元							功能						
2113		\$MIDR	P	$S_1 \cdot S_2 \cdot D$							區段擷取字串						

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

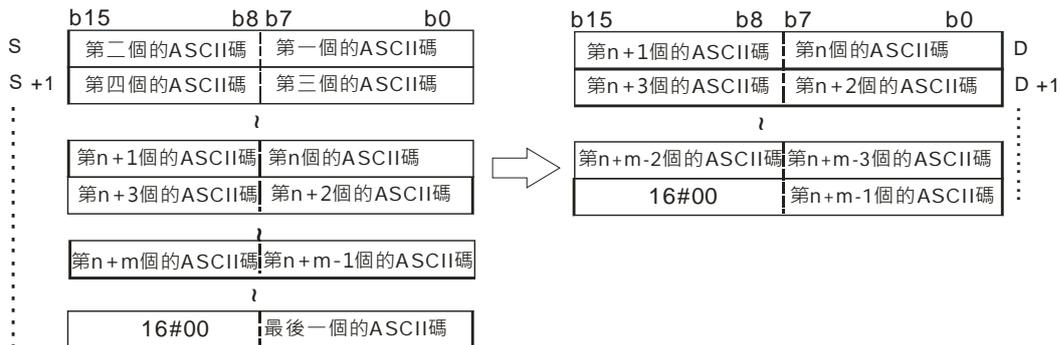
脈波執行型	16 位元指令 (7 -13 steps)	32 位元指令
AH500	AH500	-

符號：

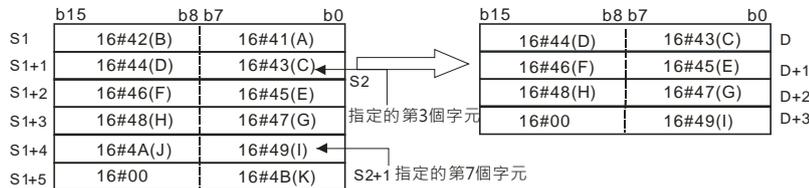
\$MIDR		\$MIDRP		S_1	字串來源	Word
En		En		S_2	要擷取的區段設定值	Word
S1		S1	D	D	存放擷取到的字串	Word
S2		S2				

指令說明：

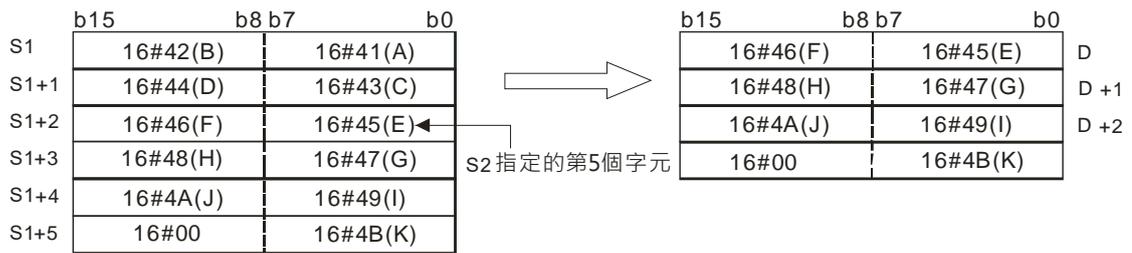
- 將 S_1 字串從 S_2 所指定的字開始擷取 S_2+1 個字，並將擷取出來的字串存放到 D。當 $S_1=n \cdot S_2+1=m$ 如圖說明。



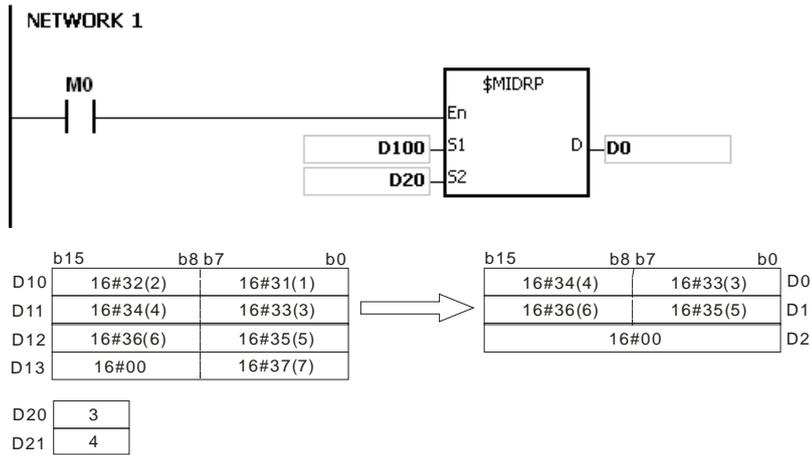
- 例如 $S=ABCDEFGHIJK$ ，當 $S_2=3$ 且 $S_2+1=7$ 時，從最左邊第 3 個字元開始擷取 7 個字，轉換結果如下圖所示。



- 當 $S_2+1=0$ 時，指令不動作。
- 當 $S_2+1=-1$ 時，擷取 S_2 開始到 S_1 結尾的所有字串。
- 例如 $S_1=ABCDEFGHIJK$ ，當 $S_2=5$ 且 $S_2+1=-1$ 時，如下圖說明。



程式範例：



補充說明：

1. S_1 字串中沒有 16#00 當結尾，SM0=ON，SR0 記錄錯誤碼 16#200E。
2. $S_2 \leq 0$ 或 $S_2+1 < -1$ 時，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. S_2 超出 S_1 的字串長度，SM0=ON，SR0 記錄錯誤碼 16#2003。
4. S_2+1 所指定的長度超出 S_1 字串的範圍，SM0=ON，SR0 記錄錯誤碼 16#2003。
5. 16 位元指令 S_2 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。

API	指令碼			運算元								功能				
2114	\$MIDW	P		$S_1 \cdot S_2 \cdot D$								區段字串取代				

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

脈波執行型	16 位元指令 (7-13 steps)	32 位元指令
AH500	AH500	-

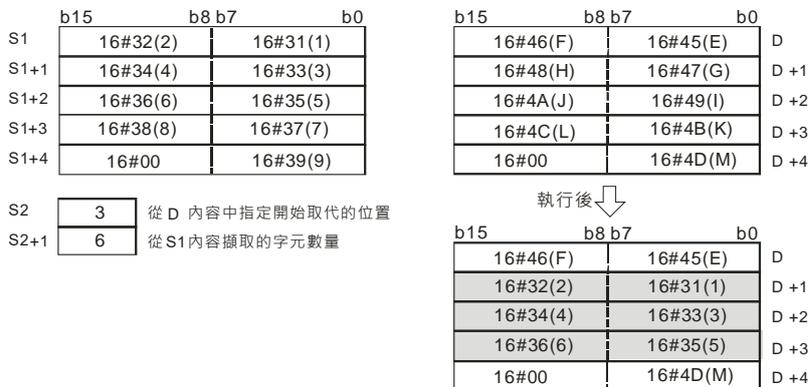
符號：

\$MIDW		\$MIDWP		S_1	Word
En		En		S_1	字串來源
S1	D	S1	D	S_2	要取代的區段設定值
S2		S2		D	要被取代的字串

指令說明：

- S_2 : D 內容中開始被取代的起始位置。 S_2+1 : S_1 內容被擷取出的字元數量。
- 從 S_1 字串的第 1 個字開始擷取 S_2+1 個字，並將擷取出來的字串，從 D 字串中 S_2 所指定的字開始取代原本 D 中的字。

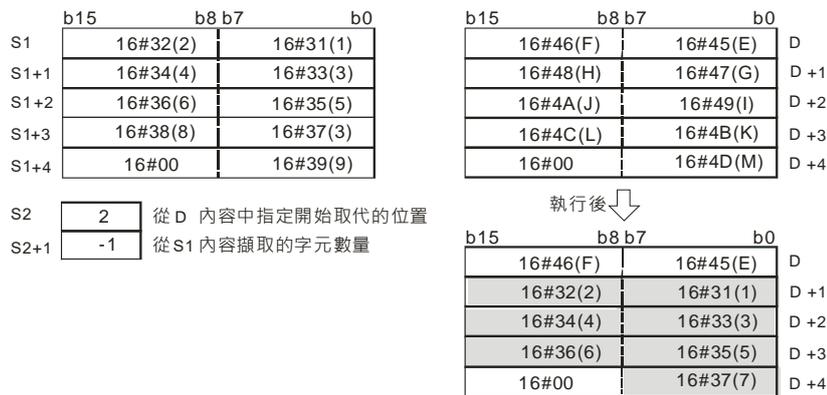
6



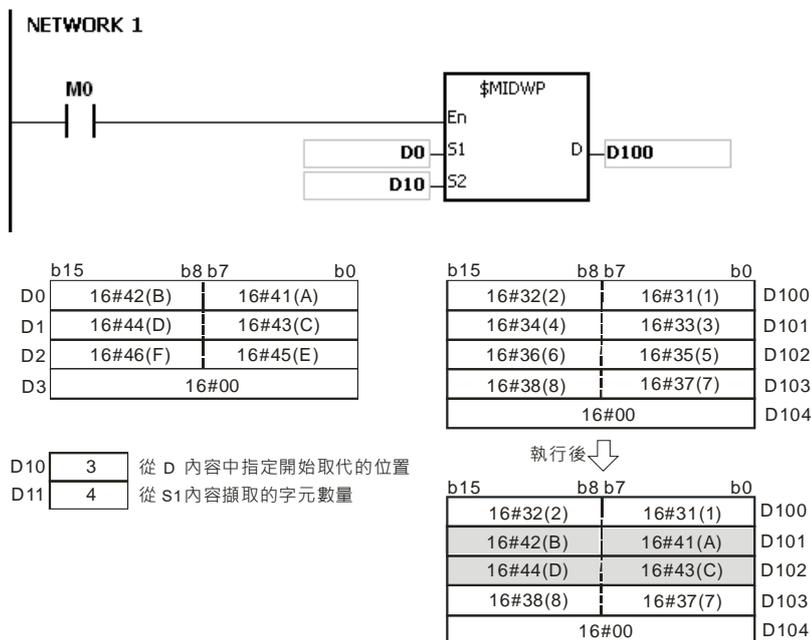
- 當 $S_2+1=0$ 時，指令不動作。
- 如果 S_2+1 所指定的長度大於 D 字串的長度，則只取代到 D 字串的最後一個字。



5. 當 $S_2+1=-1$ 時，擷取 S_1 的所有字串。



程式範例：



補充說明：

1. S_1 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
2. D 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
3. $S_2 \leq 0$ 或 $S_2 > D$ 的字串長度，SM0=ON，記錄錯誤碼 SR0=16#2003。
4. $S_2+1 < -1$ 或 S_2+1 所指定的長度超出 S_1 字串的範圍，SM0=ON，記錄錯誤碼 SR0=16#2003。
5. 16 位元指令 S_2 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。

API	指令碼			運算元							功能						
2115		\$SER	P	$S_1 \cdot S_2 \cdot n \cdot D$							字串搜尋						

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

脈波執行型	16 位元指令 (9-21 steps)	32 位元指令
AH500	AH500	-

符號：

<table border="0" style="width: 100%;"> <tr><td style="text-align: center;">\$SER</td><td></td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td style="text-align: center;">D</td></tr> <tr><td>S2</td><td></td></tr> <tr><td>N</td><td></td></tr> </table>	\$SER		En		S1	D	S2		N		<table border="0" style="width: 100%;"> <tr><td style="text-align: center;">\$SERP</td><td></td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td style="text-align: center;">D</td></tr> <tr><td>S2</td><td></td></tr> <tr><td>N</td><td></td></tr> </table>	\$SERP		En		S1	D	S2		N		<p>S_1 : 要搜尋的字串 Word</p> <p>S_2 : 要被搜尋的字串 Word</p> <p>n : 從 S_2 的第 n 個字開始搜尋 Word</p> <p>D : 搜尋結果 Word</p>
\$SER																						
En																						
S1	D																					
S2																						
N																						
\$SERP																						
En																						
S1	D																					
S2																						
N																						

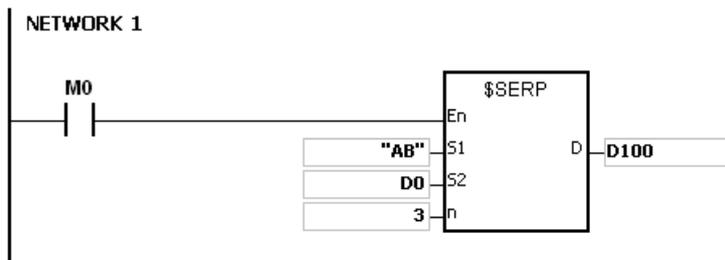
指令說明：

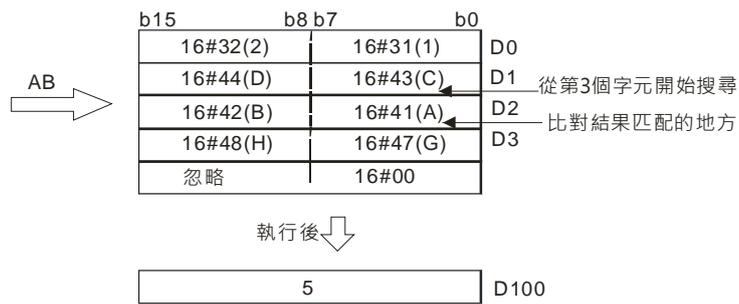
- 從 S_2 字串的第 n 個字開始搜尋與 S_1 相同的字串，如果搜尋到相同的字串，則將此字串的起始位置位於 S_2 字串的第幾個字，存於 D 中，如果沒搜尋到相同的字串，則 D=0。
- 例如，在 S_2 ="ABCDEFGHIJK"中搜尋 S_1 ="EFGH"，從 S_2 的第 3 個字元 (n=3) 開始搜尋，搜尋結果為 S_1 存在 S_2 的起始位置為 5 並存在 D=5。

6



程式範例：





補充說明：

1. S_1 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
2. S_2 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
3. $n \leq 0$ 或 $n > S_2$ 的字串長度，SM0=ON，記錄錯誤碼 SR0=16#2003。

API	指令碼			運算元						功能					
2116		\$RPLC	P	$S_1 \cdot S_2 \cdot S_3 \cdot S_4 \cdot D$						字串取代					

裝置	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	●	●			●	●		●	●				●				

脈波執行型	16 位元指令 (11-17 steps)	32 位元指令
AH500	AH500	-

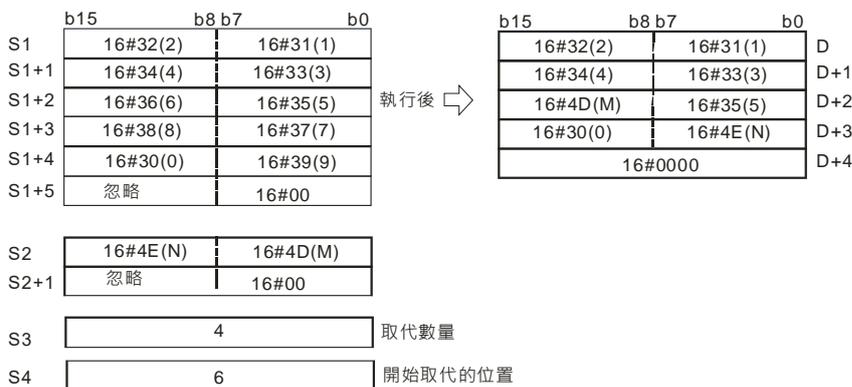
符號：

\$RPLC		\$RPLCP		S ₁ : 要被取代的字串	Word
En		En		S ₂ : 要取代的新字串	Word
S1	D	S1	D	S ₃ : S ₁ 要被取代掉的長度	Word
S2		S2		S ₄ : 從 S ₁ 的第 S ₄ 個字開始取代	Word
S3		S3		D : 取代後的字串存放位置	Word
S4		S4			

指令說明：

1. 從 S₁ 字串的第 S₄ 個字開始，用 S₂ 的字串取代 S₁ 中的字串，取代的長度為 S₃，並將結果存於 D 中，如下圖範例說明。

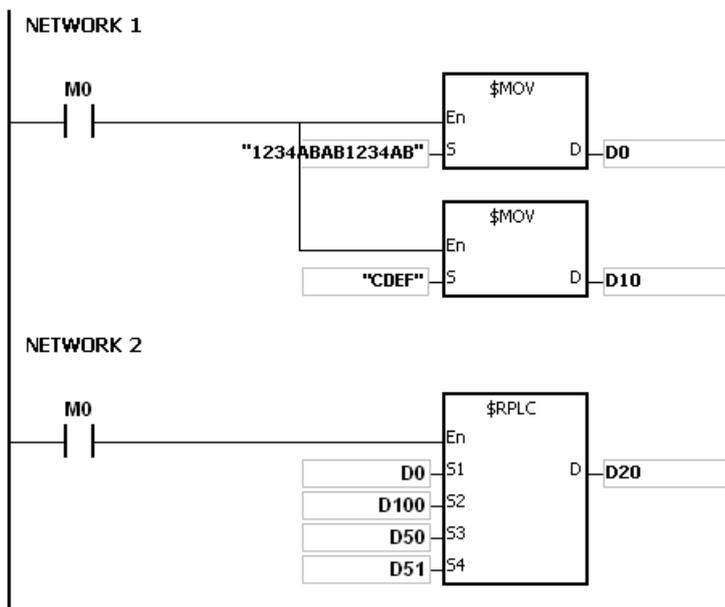
將字串“1234567890”中的第 6 個字開始的 4 個字取代為“MN”，取代後的結果為“12345MNO”。



2. S₂ 字串如果為 16#00，則此指令功能相當於將字串刪除。
3. 若欲取代的字串長度 S₃>S₁ 可被取代的長度，則只取代到 S₁ 的最後一個字。
4. S₃=0，指令不動作。

程式範例：

當 M0=ON 時，D0~D7 的內容為“1234ABAB1234AB”，D10~D11 的內容為“CDEF”；當 \$RPLC 指令執行，將用 D10~D11 的內容取代 D0~D7 中由 D51 指定的起始位置取代 D50 所指定的個數，結果儲存於 D20~D27 當中。



當 D50 指定值為 3，D51 指定值為 4 時，執行的結果如下圖所示。

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)		D21
D2	16#42(B)	16#41(A)		D22
D3	16#42(B)	16#41(A)		D23
D4	16#32(2)	16#31(1)		D24
D5	16#34(4)	16#33(3)		D25
D6	16#42(B)	16#41(A)		D26
D7	忽略	16#00		D27

執行後 ⇨

	b15	b8 b7	b0	
D20	16#32(2)	16#31(1)		D20
D21	16#43(C)	16#33(3)		D21
D22	16#45(E)	16#44(D)		D22
D23	16#41(A)	16#46(F)		D23
D24	16#31(1)	16#42(B)		D24
D25	16#33(3)	16#32(2)		D25
D26	16#41(A)	16#34(4)		D26
D27	16#00	16#42(B)		D27

D10	16#44(D)	16#43(C)
D11	16#45(F)	16#45(E)
D12	忽略	16#00

D50	3	取代數量
D51	4	開始取代的位置

當 D50 指定值為 4，D51 指定值為 4 時，執行的結果如下圖所示。

	b15	b8 b7	b0		b15	b8 b7	b0	
D0	16#32(2)		16#31(1)	執行後 →	16#32(2)		16#31(1)	D20
D1	16#34(4)		16#33(3)		16#43(C)		16#33(3)	D21
D2	16#42(B)		16#41(A)		16#45(E)		16#44(D)	D22
D3	16#42(B)		16#41(A)		16#42(B)		16#46(F)	D23
D4	16#32(2)		16#31(1)		16#32(2)		16#31(1)	D24
D5	16#34(4)		16#33(3)		16#34(4)		16#33(3)	D25
D6	16#42(B)		16#41(A)		16#42(B)		16#41(A)	D26
D7	忽略		16#00		16#0000			D27

D10	16#44(D)	16#43(C)
D11	16#45(F)	16#45(E)
D12	忽略	16#00

D50	4	取代數量
D51	4	開始取代的位置

當 D50 指定值為 20，D51 指定值為 4 時，執行的結果如下圖所示。

	b15	b8 b7	b0		b15	b8 b7	b0	
D0	16#32(2)		16#31(1)	執行後 →	16#32(2)		16#31(1)	D20
D1	16#34(4)		16#33(3)		16#43(C)		16#33(3)	D21
D2	16#42(B)		16#41(A)		16#45(E)		16#44(D)	D22
D3	16#42(B)		16#41(A)		16#00		16#46(F)	D23
D4	16#32(2)		16#31(1)					
D5	16#34(4)		16#33(3)					
D6	16#42(B)		16#41(A)					
D7	忽略		16#00					

D10	16#44(D)	16#43(C)
D11	16#45(F)	16#45(E)
D12	忽略	16#00

D50	20	取代數量
D51	4	開始取代的位置

6

當 D50 指定值為 3，D51 指定值為 4，且 D10=16#00 表示未指定取代字串時，執行的結果如下圖所示。沒有指定取代字串，執行後原本的字串被刪除。

	b15	b8 b7	b0		b15	b8 b7	b0	
D0	16#32(2)		16#31(1)	執行後 →	16#32(2)		16#31(1)	D20
D1	16#34(4)		16#33(3)		16#41(A)		16#33(3)	D21
D2	16#42(B)		16#41(A)		16#31(1)		16#42(B)	D22
D3	16#42(B)		16#41(A)		16#33(3)		16#32(2)	D23
D4	16#32(2)		16#31(1)		16#41(A)		16#34(4)	D24
D5	16#34(4)		16#33(3)		16#00		16#42(B)	D25
D6	16#42(B)		16#41(A)					
D7	16#00							

D10	16#00
-----	-------

D50	3	取代數量
D51	4	開始取代的位置

補充說明：

1. S_1 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
2. S_2 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
3. $S_3 < 0$ 或 $S_4 \leq 0$ 或 S_4 指定的位置超出 S_1 的長度，SM0=ON，記錄錯誤碼 SR0=16#2003。

API	指令碼			運算元							功能						
2117		\$DEL	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							指定字串刪除						

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

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

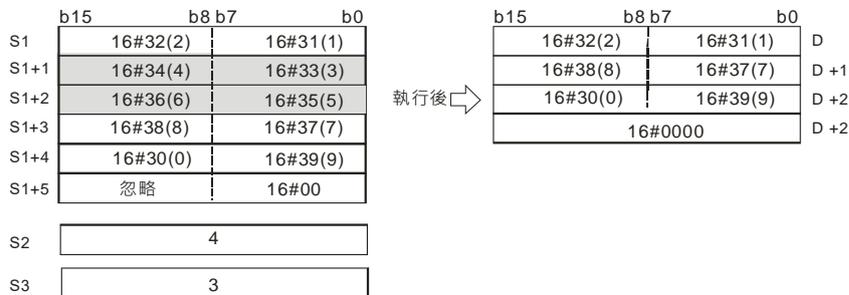
符號：

\$DEL	\$DELP	S_1 : 字串來源	Word
En	En	S_2 : 要刪除的個數	Word
S1	S1	S_3 : 從 S_1 的第 S_3 個字開始刪除	Word
S2	S2	D : 刪除後的字串	Word
S3	S3		

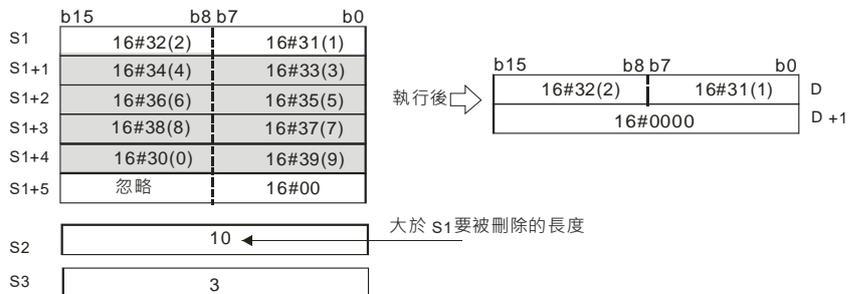
指令說明：

- 從 S_1 字串的第 S_3 個字開始，刪除 S_2 個字串，並將結果存於 D 中。
- 如下圖說明，將 S_1 的字串“1234567890”，刪除掉第 3 個開始共 4 個字串，刪除結果為“127890”並存在 D 中

6



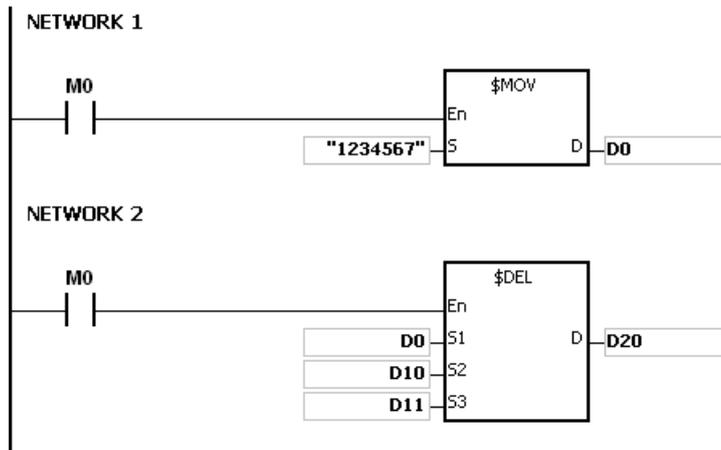
- 如果欲刪除的字串長度 $S_2 > S_1$ 要被刪除的字串長度，則只刪除到 S_1 的最後一個字，並將結束字元 16#00 存入 D 中。



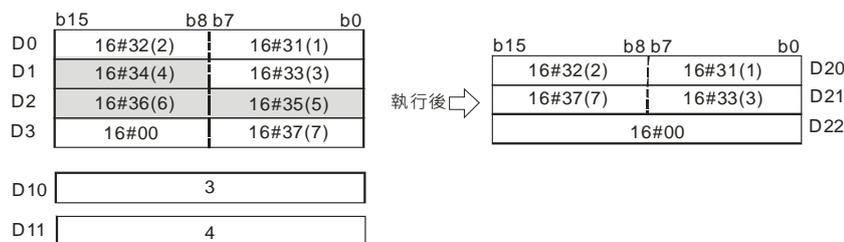
- $S_2=0$ ，指令不動作。

程式範例：

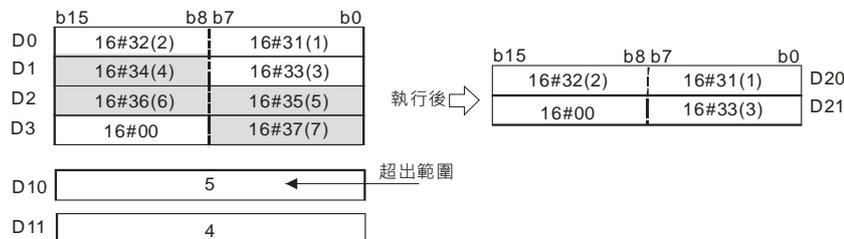
當 M0=ON 時，D0~D3 的內容為“1234567”；當\$DEL 指令執行，將從 D11 內容的指定位置開始，刪除 D0~D3 內的字串，刪除的字元數為 D10 內容所指定的數值，結果儲存於 D20~D22 當中。



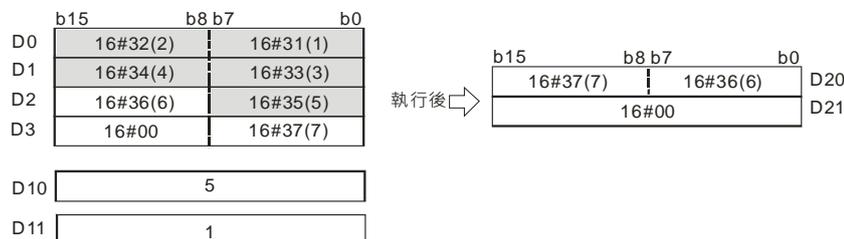
當 D10 指定刪除數量為 3，D11 指定從第 4 個字串開始刪除時，執行的結果如下圖所示。



當 D10 指定刪除數量為 5，D11 指定從第 4 個字串開始刪除時，執行的結果如下圖所示。由於指定的刪除數量超出範圍，故刪除到最後一個字後結束。



當 D10 指定刪除數量為 5，D11 指定從第 1 個字串開始刪除時，執行的結果如下圖所示。



補充說明：

1. S₁ 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
2. S₂<0 或 S₃<=0 或 S₃ 指定的位置超出 S₁ 的長度，SM0=ON，記錄錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能					
2118		\$CLR	P	S								字串清除					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S	●	●			●	●		●	●				●				
脈波執行型										16 位元指令 (3 steps)				32 位元指令			
AH500										AH500				-			

符號：



指令說明：

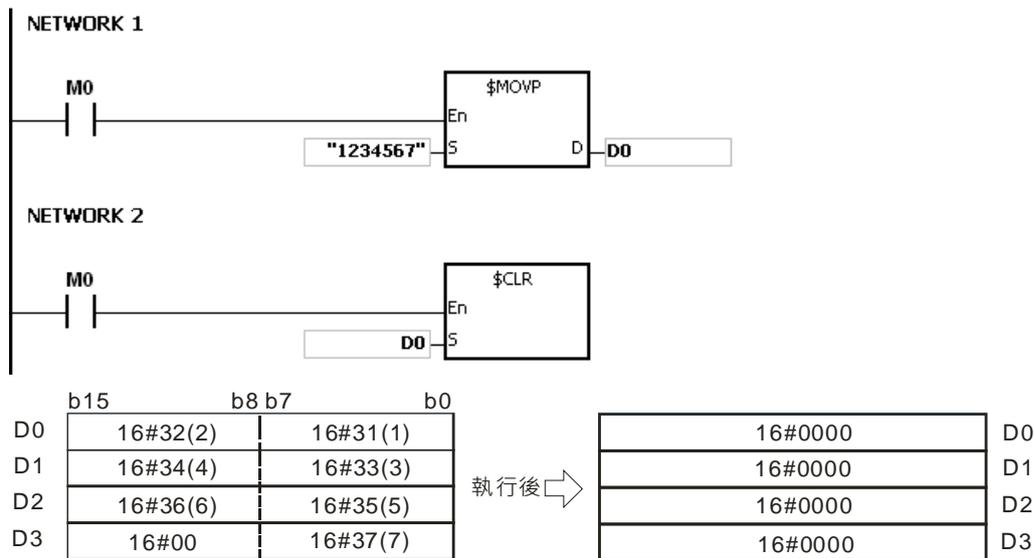
- 將 **S** 中的字串全部清除為 0。



程式範例：

6

將 D0 的字串內容清除，如下圖說明。



補充說明：

- S** 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。

API	指令碼			運算元							功能				
2119		\$INS	P	S₁ · S₂ · S₃ · D							字串插入				

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁	●	●			●	●		●	●		●		●				
S ₂	●	●			●	●		●	●		●		●			○	
S ₃	●	●			●	●		●	●		●	○	●	○	○		
D	●	●			●	●		●	●				●				

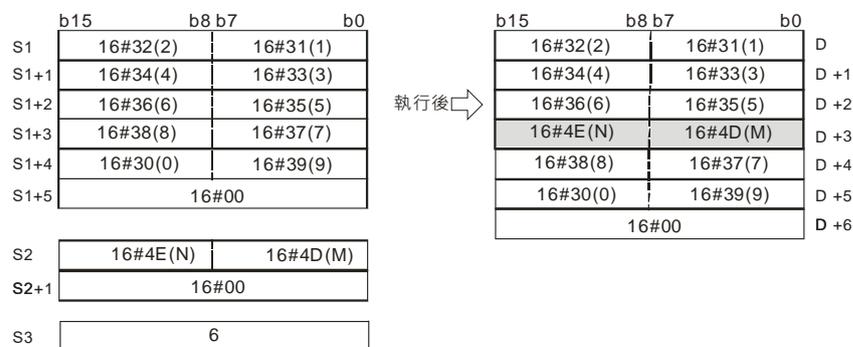
脈波執行型	16 位元指令 (9-15 steps)	32 位元指令
AH500	AH500	-

符號：

\$INS		\$INSP		S ₁ : 來源字串	Word
En		En		S ₂ : 要插入的字串	Word
S1		S1	D	S ₃ : 從 S ₁ 的第 S ₃ 個字之後開始插入	Word
S2		S2		D : 存放插入之後的字串	Word
S3		S3			

指令說明：

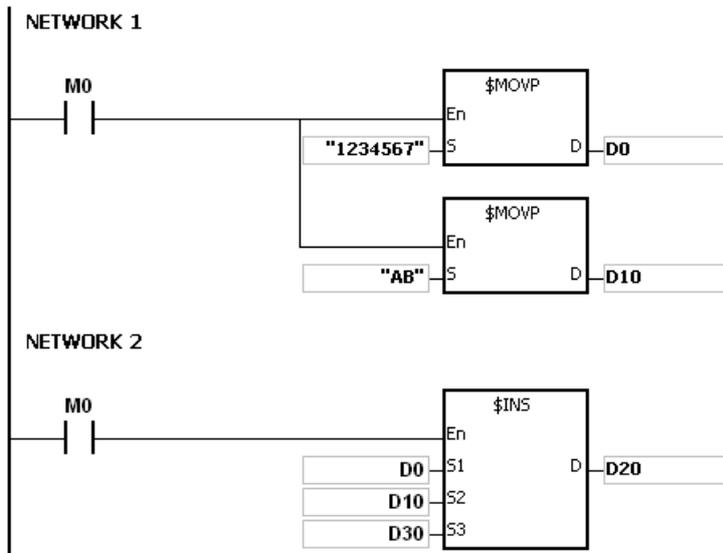
- 從 S₁ 字串的第 S₃ 個字之後開始，插入 S₂ 的字串，並將結果存於 D 中。
- 如果 S₁ 或 S₂ 有任何一個字串為空字串，則另一個不是空字串的值會被存於 D 中。
- 如果 S₁ 和 S₂ 都為空字串，則 D 為 16#0000。



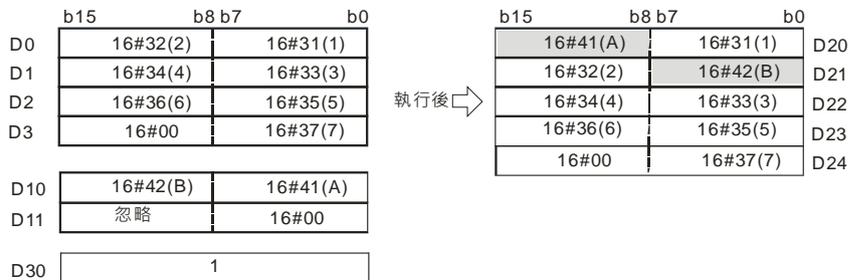
6

程式範例：

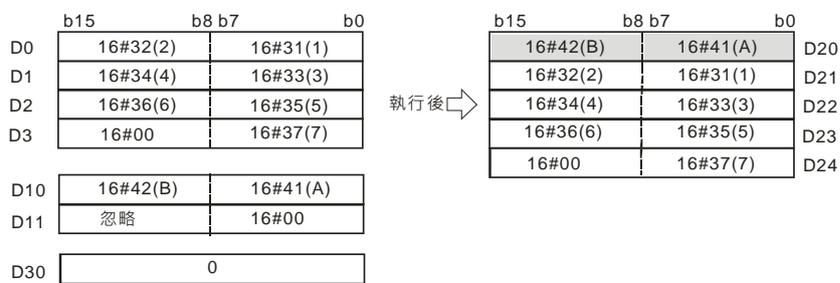
當 M0=ON 時，D0~D3 的內容為“1234567”，D10 的內容為“AB”；當 \$INS 指令執行，將從 D30 內容所指定的位置之後開始，插入 D10 的內容，結果儲存於 D20~D24 當中。



D30 指定從第 1 個字元後開始插入，執行的結果如下圖所示。



D30 指定從第 0 個字元後開始插入，執行的結果如下圖所示。



6

補充說明：

1. S_1 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
2. S_2 字串中沒有 16#00 當結尾，SM0=ON，記錄錯誤碼 SR0=16#200E。
3. $S_3 < 0$ 或 S_3 指定的位置超出 S_1 的長度，SM0=ON，記錄錯誤碼 SR0=16#2003。

API	指令碼			運算元								功能					
2120	FMOD	P		$S_1 \cdot S_2 \cdot D$								浮點數轉 BCD 浮點數					

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

脈波執行型	32 位元指令 (7-8 steps)	64 位元指令
AH500	AH500	-

符號：

FMOD		FMODP		S_1 :	來源浮點數值	Double Word
En		En		S_2 : <td>小數點位移位數</td> <td>Word</td>	小數點位移位數	Word
S1	D	S1	D	D : <td>轉換後的結果</td> <td>Word</td>	轉換後的結果	Word
S2		S2				

指令說明：

- 將浮點數 S_1 中的小數點右移 S_2 的位數後，再轉成 BCD 浮點數格式，並將結果存於 D 中。



BCD 浮點數格式說明如下：

S_2 ：小數點指定位移位數， S_2 的範圍為 0~7。

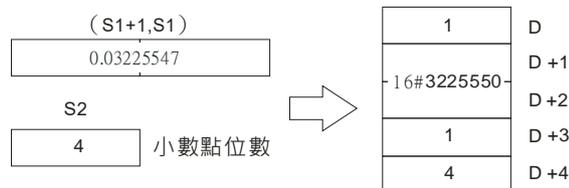
D：紀錄符號，若 S_1 是正數，則 $D=0$ ；若 S_1 是負數，則 $D=1$ 。

($D+2$, $D+1$)：紀錄浮點數轉換後的 7 位數 BCD 值。

$D+3$ ：紀錄浮點數轉成 BCD 浮點數格式之後的指數符號。若指數符號是正數，則 $D+3=0$ ；若指數符號是負數，則 $D+3=1$ 。

$D+4$ ：紀錄 BCD 浮點數格式的指數值。

例如 $S_1=-0.03225547$ ，指定小數點右移位數為 4，經轉換後的結果如下圖所示。



S_2 指定小數點右移位數為 4，所以 S_1 的小數點往右移 4 位後為 -322.5547。

將 -322.5547 視為 -3225547E-4，轉換為 BCD 浮點數格式後為

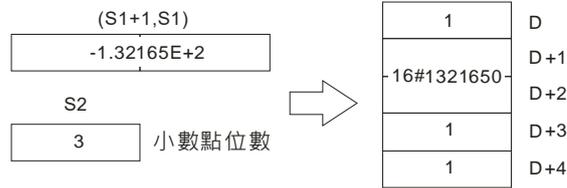
$D=1$ (因為 S_1 是負數)。

($D+2$, $D+1$) = 16#3225550 (轉成 7 位數的 BCD，第 7 位數被四捨五入)。

$D+3=1$ (因為指數的符號是負數)。

$D+4=4$ (因為指數的數值為 4)。

例如 $S_1=-1.32165E+2$ ，指定小數點右移位數為 3，經轉換後的結果如下圖所示。



原本的 S_1 浮點數為 $-1.32165E+2 = -132.165$ 。

S_2 指定小數點右移 3 位，所以 S_1 的小數點往右移 3 位後為 -132165 。

將 -132165 視為 $-1321650E-1$ (因為需固定 7 位數 BCD)，轉換為 BCD 浮點數格式後為

$D = 1$ (因為 S_1 是負數)。

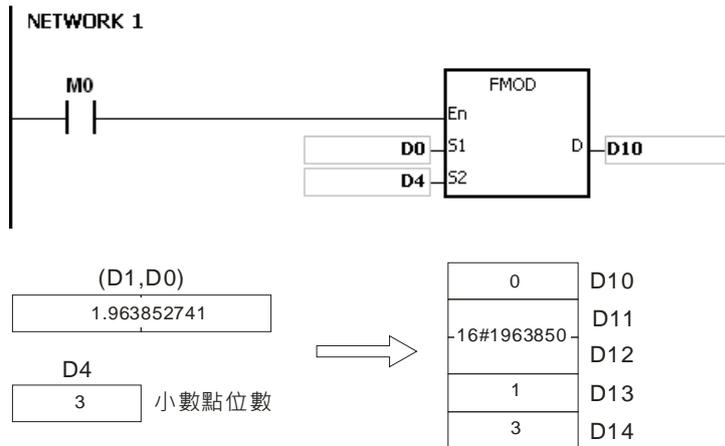
($D+2$ 、 $D+1$) = $16\#1321650$ (轉成 7 位數的 BCD，第 7 位數被四捨五入)。

$D+3=1$ (因為指數的符號是負數)。

$D+4=1$ (因為指數的數值為 1)。

程式範例：

6



(D_1 、 D_0) = 1.963852741 ， $D_4=3$ ，所以右移 3 個位數之後 = 1963.852741 。

$D_{10}=0$ ，因為來源 (D_1 、 D_0) 值為正數。

將 1963.852741 視為 $1963852E-3$ (因為需固定 7 位數 BCD)，轉換為 BCD 浮點數格式後為

(D_{12} 、 D_{11}) = $16\#1963850$ (取七位數，第七位四捨五入)。

$D_{13}=1$ (指數符號為負數)。

$D_{14}=3$ (指數值為 3)。

補充說明：

1. 當 S_1 超出浮點數可以表示的範圍時，指令不執行， $SM_0=ON$ ，錯誤碼 $SR_0=16\#2013$ 。
2. S_2 的內容值超出範圍時，指令不執行， $SM_0=ON$ ，錯誤碼 $SR_0=16\#2003$ 。
3. 32 位元指令 D 運算元，若使用 ISPSOft 宣告，則資料型態為 $ARRAY [5] \text{ of } WORD/INT$ 。

API	指令碼			運算元								功能					
2121		FREXP	P	$S_1 \cdot S_2 \cdot D$								BCD 浮點數轉浮點數					

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

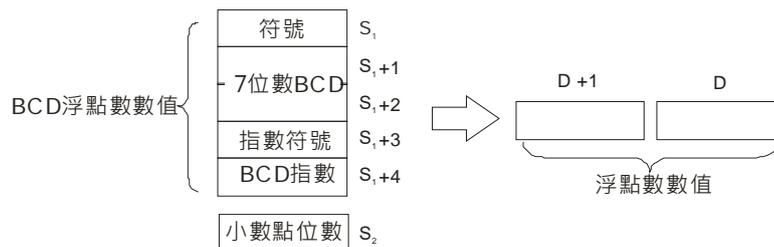
脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

FREXP		FREXPP			
En		En		S_1 :	BCD 浮點數來源值 Word
S1	D	S1	D	S_2 :	小數點位移位數 Word
S2		S2		D :	轉換後的結果 Double Word

指令說明：

將 BCD 浮點數格式數值 S_1 ，依照 S_2 設定的小數點左移的位數，轉成浮點數，並將結果存於 D 中。

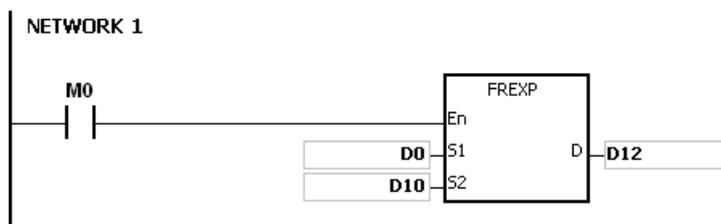


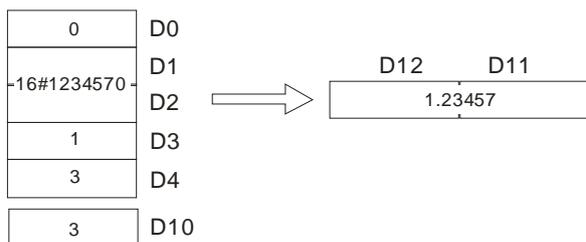
BCD 浮點數格式說明如下：

- S_1 紀錄浮點數的符號，若符號為 0 表浮點數為正數，若符號為 1 表浮點數為負數。
- ($S_{1+2} \cdot S_{1+1}$) 紀錄 BCD 浮點數表示值的 7 位數整數值。
- S_{1+3} 紀錄指數部分的符號，若指數符號為 0 表示指數值為正數，若指數符號為 1 表示指數值為負數。
- S_{1+4} ，紀錄 BCD 的指數值，BCD 指數的範圍為 0~38。
- S_2 ：紀錄小數點指定左移位數， S_2 的範圍為 0~7。(來源轉成浮點數之後，小數點再左移 S_2 個位數)

程式範例：

啟動條件接點 M0，將 BCD 浮點數轉成浮點數值，說明如下：





D0=0 表示浮點數值為正數.

(D2 · D1) =16#1234570 表示浮點數的整數部分

D3=1 表示指數符號為負數

D4=3 表示指數值為 3

將 D0~D4 轉換成浮點數後為 1234570E-3=1234.57

D10=3 表示小數點需左移 3 位數，轉移後的浮點數為 1.23457 並紀錄在 (D12 · D11)

補充說明：

1. 當 S_1 符號所指定的值不是 0 也不是 1，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
2. S_{1+1} · S_{1+2} 所指定的 BCD 位數>7，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
3. S_{1+1} · S_{1+2} 的資料內容並非為 BCD 值 (以 Hex 表示有任一位數不在 0~9 的範圍內)，則將會產生運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#200D。
4. 當 S_{1+3} 符號所指定的值不是 0 也不是 1，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
5. S_{1+4} · BCD 指數內容值<0 或>38，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
6. S_2 的內容值<0 或>7，指令不執行，SM0=ON，錯誤碼 SR0=16#2003。
7. 32 位元指令 S_1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [5] of WORD/INT。



6.22 乙太網控制指令

6.22.1 乙太網控制指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>2200</u>	SOPEN	–	✓	開啟 Socket	7
<u>2201</u>	SSEND	–	✓	透過已開啟的 Socket 傳送資料	5
<u>2202</u>	SRCVD	–	✓	透過已開啟的 Socket 接收資料	5
<u>2203</u>	SCLOSE	–	✓	關閉 Socket	5
<u>2204</u>	MSEND	–	✓	寄送電子郵件	9
<u>2205</u>	EMDRW	–	✓	MODBUS TCP 資料讀寫	11
<u>2206</u>	–	DINTOA	✓	IP→字串變換	5
<u>2207</u>	–	DIATON	✓	字串→IP 變換	5-11
<u>2208</u>	EIPRW	–	–	EtherNet/IP 讀寫通訊指令	23

6.22.2 乙太網控制指令說明

API	指令碼			運算元							功能						
2200		SOPEN	P	$S_1 \cdot S_2 \cdot S_3$							開啟 Socket						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●	○	○		
S_3	●	●			●	●		●	●			○	●	○	○		

脈波執行型	16 位元指令 (7 steps)	32 位元指令
AH500	AH500	-

符號：

SOPEN		SOPENP		S_1 : Socket 模式.	Word
En		En		S_2 : Socket 編號	Word
S1		S1		S_3 : 開啟模式	Word
S2		S2			
S3		S3			

指令說明：

- S_1 為 Socket 模式，欲開啟 TCP Socket 時， S_1 為 1 時，開啟 UDP Socket 時， S_1 為 0， S_2 為 Socket 編號，對應乙太網路中 Socket 設定，範圍為 1~8。
- 開啟 TCP 連線時， S_3 為 1 和 3 時，表示 AH500 為 Client 端，主動送出連線要求給 Server 端， S_3 為 0 和 2 時，表示 AH500 為 Server 端，等待來自 Client 端的連線要求，欲開啟 UDP 連線時， S_3 可設為 0~2， S_3 模式說明如下。

S_3 設定值	Socket 模式	開啟模式	傳送資料上限 (Words)	接收資料上限 (Words)
0	TCP	Client	500	500
1	TCP	Client	500	500
2 (V2.0x)	TCP	Server	4096	4096
3 (V2.0x)	TCP	Server	4096	4096
0~1	UDP	N/A	500	500
2 (V2.0x)	UDP	N/A	4096	4096

- 使用本指令前，需用 ISPSOft - > HWCONFIG 完成以下設定，或者用 MOV 指令將 Socket 資料搬入對應的 SR 暫存器中。
 - 在 PLC 參數設定 - > 基本設定 - > 設定本機的 IP 位址、網路遮罩。
 - 在 PLC 參數設定 - > 進階設定 - > 通訊接口 - > 啟動通訊接口功能。
 - 在 PLC 參數設定 - > 進階設定 - > 通訊接口 - > TCP/UDP 通訊接口連線設定欲使用的 Socket

4. TCP Socket 設定對應暫存器，其中除了傳送資料計數器及接收資料計數器是唯讀，其他暫存器的值皆可更改。

Socket 編號 項目	1	2	3	4	5	6	7	8
本機通訊埠	SR1118	SR1131	SR1144	SR1157	SR1170	SR1183	SR1196	SR1209
遠端 IP H	SR1119	SR1132	SR1145	SR1158	SR1171	SR1184	SR1197	SR1210
遠端 IP L	SR1120	SR1133	SR1146	SR1159	SR1172	SR1185	SR1198	SR1211
遠端通訊埠	SR1121	SR1134	SR1147	SR1160	SR1173	SR1186	SR1199	SR1212
傳送長度	SR1122	SR1135	SR1148	SR1161	SR1174	SR1187	SR1200	SR1213
傳送位址 H	SR1123	SR1136	SR1149	SR1162	SR1175	SR1188	SR1201	SR1214
傳送位址 L	SR1124	SR1137	SR1150	SR1163	SR1176	SR1189	SR1202	SR1215
接收長度	SR1125	SR1138	SR1151	SR1164	SR1177	SR1190	SR1203	SR1216
接收位址 H	SR1126	SR1139	SR1152	SR1165	SR1178	SR1191	SR1204	SR1217
接收位址 L	SR1127	SR1140	SR1153	SR1166	SR1179	SR1192	SR1205	SR1218
保持連線時間 (Sec)	SR1128	SR1141	SR1154	SR1167	SR1180	SR1193	SR1206	SR1219
接收資料計數器	SR1129	SR1142	SR1155	SR1168	SR1181	SR1194	SR1207	SR1220
傳送資料計數器	SR1130	SR1143	SR1156	SR1169	SR1182	SR1195	SR1208	SR1221

5. 連線保持時間若設為 0，則表示沒有該連線會保持等待狀態，不會逾時。
6. UDP Socket 設定對應暫存器，其中除了傳送資料計數器及接收資料計數器是唯讀，其他暫存器的值皆可更改。

Socket 編號 項目	1	2	3	4	5	6	7	8
本機通訊埠	SR1222	SR1234	SR1246	SR1258	SR1270	SR1282	SR1294	SR1306
遠端 IP H	SR1223	SR1235	SR1247	SR1259	SR1271	SR1283	SR1295	SR1307
遠端 IP L	SR1224	SR1236	SR1248	SR1260	SR1272	SR1284	SR1296	SR1308
遠端通訊埠	SR1225	SR1237	SR1249	SR1261	SR1273	SR1285	SR1297	SR1309
傳送長度	SR1226	SR1238	SR1250	SR1262	SR1274	SR1286	SR1298	SR1310
傳送位址 H	SR1227	SR1239	SR1251	SR1263	SR1275	SR1287	SR1299	SR1311
傳送位址 L	SR1228	SR1240	SR1252	SR1264	SR1276	SR1288	SR1300	SR1312
接收長度	SR1229	SR1241	SR1253	SR1265	SR1277	SR1289	SR1301	SR1313
接收位址 H	SR1230	SR1242	SR1254	SR1266	SR1278	SR1290	SR1302	SR1314
接收位址 L	SR1231	SR1243	SR1255	SR1267	SR1279	SR1291	SR1303	SR1315
接收資料計數器	SR1232	SR1244	SR1256	SR1268	SR1280	SR1292	SR1304	SR1316
傳送資料計數器	SR1233	SR1245	SR1257	SR1269	SR1281	SR1293	SR1305	SR1317

7. 開啟 TCP Socket 時，Socket IP 和通訊埠編號設定。

遠端 IP	本機通訊埠	遠端通訊埠	說明
0.0.0.0	0	0	不合法
0.0.0.0	不等於 0	0	僅用於 Server 端應用 1. 接受同一個本機通訊埠編號的連線要求 2. 透過本機通訊埠接收任何裝置的封包
0.0.0.0	0	不等於 0	不合法
特定 IP 位址	0	0	不合法
特定 IP 位址	不等於 0	0	僅用於 Server 端應用 1. 透過本機通訊埠接收特定 IP 位址的封包
特定 IP 位址	0	不等於 0	僅用於 Client 端應用 1. 當要求建立連線時，系統會指定未使用的通訊埠為本機通訊埠。 2. 透過遠端通訊埠傳送資料至特定 IP 位址
特定 IP 位址	不等於 0	不等於 0	1. 接受設定之本機通訊埠編號、遠端通訊埠和特定 IP 位址的連線要求 2. 透過遠端通訊埠傳送資料至特定 IP 位址啟動 3. 透過本機通訊埠接收特定 IP 位址所傳送的封包

8. 開啟 TCP Socket 時，當此指令執行後，若無錯誤，則開始與遠端裝置進行連線，開啟中旗標為 ON，若連線成功則已連線旗標為 ON，開啟中旗標為 OFF，反之若有錯誤則對應錯誤旗標 ON。

6

TCP Socket 編號	開啟中旗標	已連線旗標	錯誤旗標
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

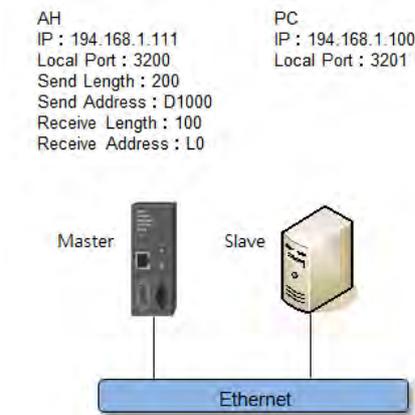
9. 開啟 UDP Socket 時，當此指令執行後，若無錯誤，則連線已開啟之旗標為 ON，反之若有錯誤則對應錯誤旗標 ON。

UDP Socket 編號	連線已開啟旗標	錯誤旗標
1	SM1334	SM1338
2	SM1339	SM1343
3	SM1344	SM1348
4	SM1349	SM1353
5	SM1354	SM1358
6	SM1359	SM1363
7	SM1364	SM1368
8	SM1369	SM1373

10. 本指令一般都是使用脈波執行型指令 (**SOPENP**) 。

程式範例：

1. 此程式範例的系統架構如下圖，此範例說明如何以 PC 為 Server 端，AH500 為 Client 端建立一個 TCP 連線，此範例為使用第一組 TCP 連線。



2. ISPSOft -> HWCONFIG (乙太網路-基本設定)

Ethernet Configuration

IP Addressing Mode: Refresh

IP Address:

Netmask Address:

Gateway Address:

Keep Alive Timer: sec (1 ~ 65535 sec)

3. ISPSOft -> HWCONFIG (乙太網路-進階設定 -> 通訊接口 -> TCP)

TCP Socket Connection

Remote IP:

Remote Port:

Local Port:

Send Data Length: (0 ~ 500 words)

Send Data Address:

Receive Data Length: (0 ~ 500 words)

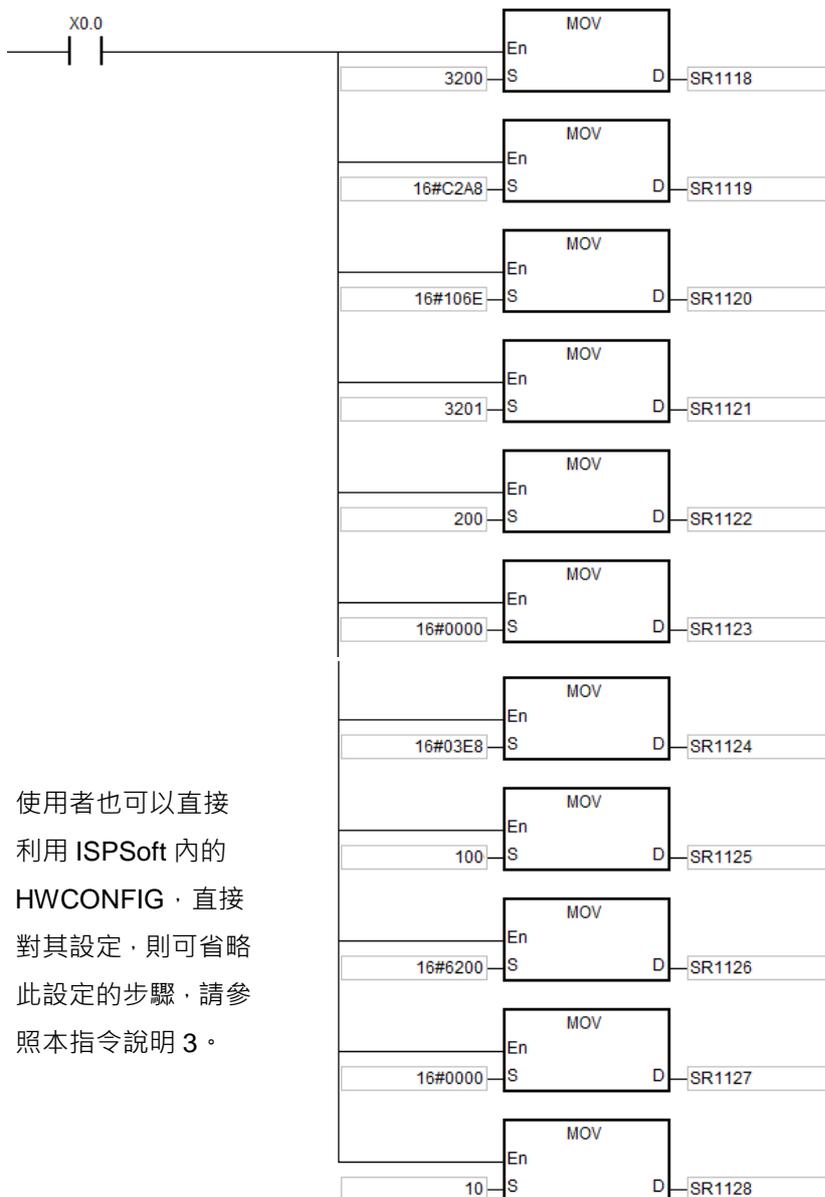
Receive Data Address:

Keep Alive Timer: sec (0 ~ 65535 sec)

OK Cancel

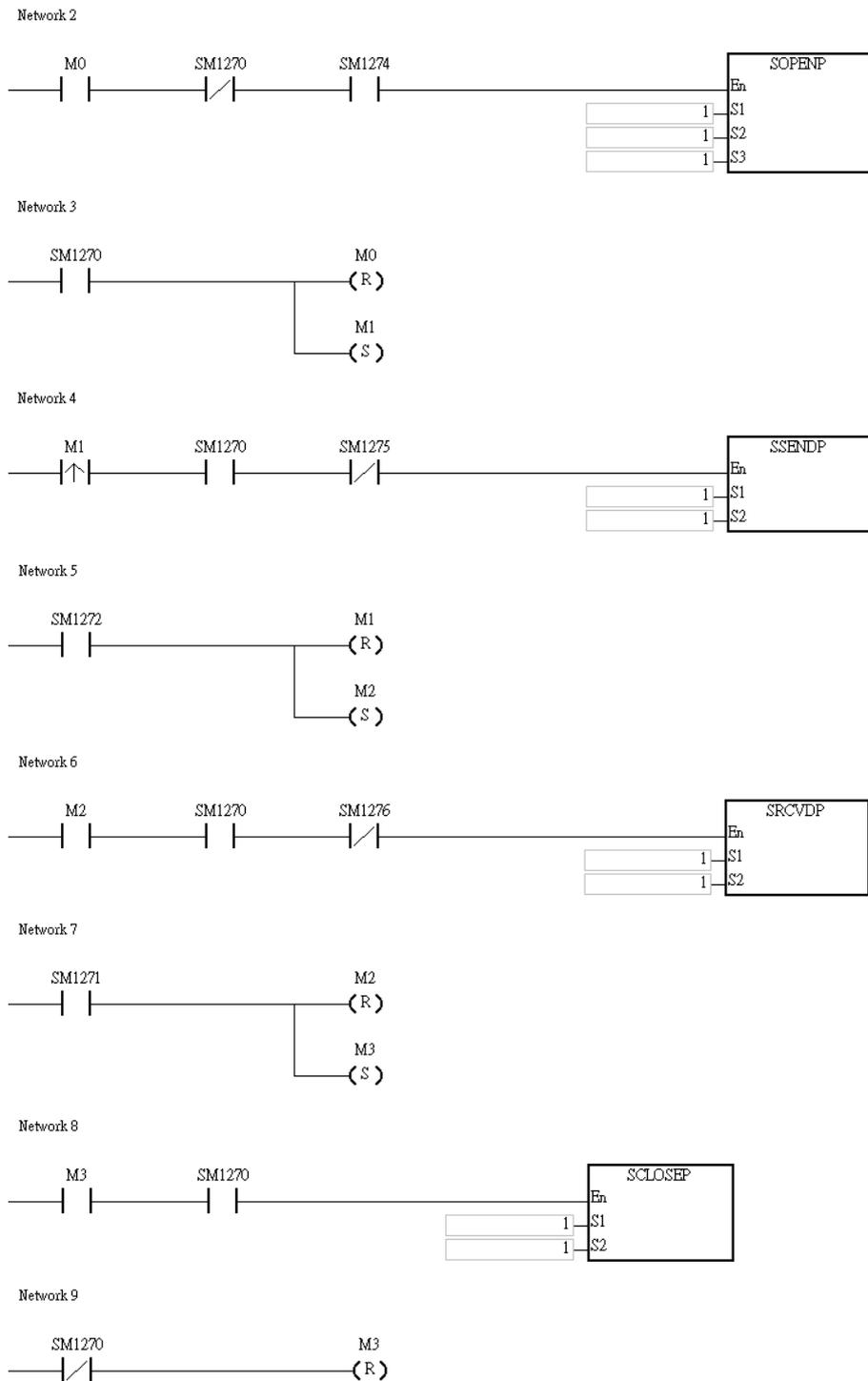
4. 當 X0.0=ON 時，設定 TCP Socket1 的資訊，亦可先行在 ISPSOft PLC 參數設定中設定，即可省略 Network 1。
5. 當 M0=ON 時，檢查欲連線的 Socket 是否已連線或是連線中的狀況，如果不是則正式進入連線程序，連線完成後清除 M0，並使 M1=ON。
6. 當 M1=ON 時，準備傳送資料，檢查欲傳送的 Socket 是否為已連線，並且無資料在傳送中的狀態，若是則開始傳送資料，若否則此指令無法運行，傳送完成後清除 M1，並使 M2=ON。
7. 當 M2=ON 時，準備接收資料，檢查欲接收的 Socket 是否為已連線，並且不是在接收中的狀態；若已是，則此 Socket 轉為接收中狀態；若否，則此指令無法運行。當接收完成後清除 M2，並使 M3=ON。
8. 當 M3=ON 時，準備關閉連線，檢查欲關閉的 Socket 是否為已連線狀態，若已連線則開始關閉連線，若尚未連線則此指令無法運行，當連線關閉後，清除 M3。

Network 1



使用者也可以直接利用 ISPSOft 內的 HWCONFIG，直接對其設定，則可省略此設定的步驟，請參照本指令說明 3。

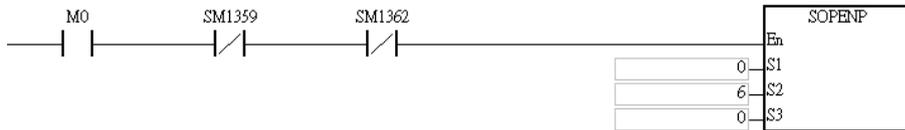
6



9. 此程式另一範例如下，此範例說明如何使用 AH 建立一個 UDP 連線，此範例為使用 UDP 第六組連線。
10. 當 M0=ON 時，檢查欲連線的 Socket 是否處於連線已開啟或是正在接收中狀態，如果不是則正式進入連線程序，連線完成後清除 M0，並使 M1=ON。
11. 當 M1=ON 時，開始傳送資料，傳送完成後清除 M1，並使 M2=ON。

12. 當 M2=ON 時，準備接收資料，檢查欲接收的 Socket 是否為已連線，並且檢查是否在接收中狀態，若於接收中狀態則此 Socket 轉為接收中狀態，若不是接收中狀態，則此指令無法運行，當資料接收完成後清除 M2，並使 M3=ON。
13. 當 M3=ON 時，開始關閉連線，檢查欲關閉的 Socket 是否為已連線狀態，若已連線則開始關閉連線，若尚未連線則此指令無法運行，連線關閉後清除 M3。

Network 1



Network 2



Network 3



Network 4



Network 5



Network 6



Network 7



Network 8



6

補充說明：

1. 如果 Client 與 Server 端皆為使用 AH 系列 PLC，且逾時時間設定相同，如 Server 端先發生逾時，則 Server 端會自行切斷連線，所以 Client 端不會有錯誤旗標發生，反之 Client 端先發生逾時，則 Client 端會發生錯誤旗標，並切斷連線。
2. TCP 連線執行錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600A	TCP 連線建立失敗	SM1090
16#600C	Socket 通訊接口已被使用	SM1109
16#600D	RJ45 埠未連接	SM1100
16#6200	TCP 通訊接口 (Socket) 功能的遠端 IP 位址不合法	註 1
16#6201	TCP 通訊接口 (Socket) 功能的本地埠不合法	註 1
16#6202	TCP 通訊接口 (Socket) 功能的遠端埠不合法	註 1
16#6214	遠端裝置拒絕連線 (Socket)	註 1
16#6217	目前通訊接口 (Socket) 已開啟	註 1
16#621A	目前通訊接口 (Socket) 關閉中	註 1

註 1：

SM1277：TCP Socket 1 錯誤旗標。

SM1285：TCP Socket 2 錯誤旗標。

~

SM1325：TCP Socket 7 錯誤旗標。

SM1333：TCP Socket 8 錯誤旗標。

3. UDP 連線執行錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600B	UDP 連線建立失敗	SM1090
16#600C	Socket 通訊接口已被使用	SM1109
16#600D	RJ45 埠未連接	SM1100
16#6209	UDP 通訊接口 (Socket) 功能的遠端 IP 位址不合法	註 2
16#620A	UDP 通訊接口 (Socket) 功能的本地埠不合法	註 2
16#620B	UDP 通訊接口 (Socket) 功能的遠端埠不合法	註 2
16#6217	目前通訊接口 (Socket) 已開啟	註 2
16#621A	目前通訊接口 (Socket) 關閉中	註 2

註 2 :

SM1338 : UDP Socket 1 錯誤旗標。

SM1343 : UDP Socket 2 錯誤旗標。

~

SM1368 : UDP Socket 7 錯誤旗標。

SM1373 : UDP Socket 8 錯誤旗標。

6

API	指令碼			運算元								功能					
2201		SSEND	P	$S_1 \cdot S_2$								透過已開啟的 Socket 傳送資料					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●	○	○		

脈波執行型	16 位元指令 (5 steps)	32 位元指令
AH500	AH500	-

符號：

SSSEND		SSENDP	
En		En	
S1		S1	
S2		S2	

 S_1 : Socket 模式. Word S_2 : Socket 編號 Word

指令說明：

- S_1 為 Socket 模式，欲開啟 TCP Socket 時， S_1 為 1，開啟 UDP Socket 時， S_1 為 0， S_2 為 Socket 編號對應乙太網路中 Socket 設定。
- $S_1=0\sim 1$ ， $S_2=1\sim 8$ 。
- 使用本指令前，需先以 SOPEN 開啟 Socket 連線，當已連線旗標 (TCP) 或是已開啟旗標 (UDP) 為 ON 時，方可使用本指令進行資料傳輸。
- 透過 TCP Socket 傳送資料時，當此指令執行後，若無錯誤，則開始傳送資料，傳送中旗標為 ON，若傳送成功則已傳送旗標為 ON，傳送中旗標為 OFF，反之若有錯誤則對應錯誤旗標 ON。

TCP Socket 編號	傳送中旗標	已傳送旗標	錯誤旗標
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

- 透過 UDP Socket 傳送資料時，當此指令執行後，若無錯誤，則資料已傳送旗標為 ON，反之若有錯誤則對應錯誤旗標 ON。

UDP Socket 編號	已傳送旗標	錯誤旗標
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脈波執行型指令 (SSENDP)。

7. 使用此指令時，如欲傳送奇數位元組長度位元組資料，可使用下列旗標進行設定。

Socket 編號	奇數位元組啟動旗標 (TCP)	奇數位元組啟動旗標 (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

奇數位元組啟動旗標功能說明：

- 當傳送的資料長度設定為 4 個 Word 時。(其相關設定與程式範例，請參考 SOPEN 指令章節)

- 當未使用奇數位元組啟動旗標時，則實際傳送的資料內容為 8 個 Bytes，若使用數位元組啟動旗標時，則實際傳送的資料內容為 7 個 Bytes，其實際輸出的資料內容如下表。

主機傳送資料位址內容							
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
實際傳送資料內容 (奇數位元組啟動旗標 = OFF)							
0102 0304 0506 0708							
實際傳送資料內容 (奇數位元組啟動旗標 = ON)							
0102 0304 0506 07							

補充說明：

1. TCP 發送錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍。	SM0/SM5
16#600D	RJ45 埠未連接。	SM1100
16#6203	TCP 通訊接口 (Socket) 功能的傳送資料位址不合法	註 1
16#6204	TCP 通訊接口 (Socket) 功能的傳送資料長度不合法	註 1
16#6205	TCP 通訊接口 (Socket) 功能的傳送資料裝置超出範圍	註 1
16#6212	遠端裝置回應逾時 (Socket)	註 1
16#6214	遠端裝置拒絕連線 (Socket)	註 1
16#6215	目前通訊接口 (Socket) 未開啟	註 1
16#6218	目前通訊接口 (Socket) 傳送中	註 1

註 1：

SM1277：TCP Socket 1 錯誤旗標

SM1285：TCP Socket 2 錯誤旗標

~

SM1325：TCP Socket 7 錯誤旗標

SM1333：TCP Socket 8 錯誤旗標

2. UDP 發送錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600D	RJ45 埠未連接	SM1100
16#620C	UDP 通訊接口 (Socket) 功能的傳送資料位址不合法	註 2
16#620D	UDP 通訊接口 (Socket) 功能的傳送資料長度不合法	註 2
16#620E	UDP 通訊接口 (Socket) 功能的傳送資料裝置超出範圍	註 2
16#6218	目前通訊接口 (Socket) 傳送中	註 2

註 2：

SM1338：UDP Socket 1 錯誤旗標。

SM1343：UDP Socket 2 錯誤旗標。

~

SM1368：UDP Socket 7 錯誤旗標。

SM1373：UDP Socket 8 錯誤旗標。

API	指令碼			運算元							功能						
2202		SRCVD	P	S₁ · S₂							透過已開啟的 Socket 接收資料						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S₁	●	●			●	●		●	●			○	●	○	○		
S₂	●	●			●	●		●	●			○	●	○	○		
脈波執行型										16 位元指令 (5 steps)				32 位元指令			
AH500										AH500				-			

符號：

SRCVD		SRCVDP		S₁ : Socket 模式	Word
En		En			
S1		S1			
S2		S2		S₂ : Socket 編號	Word

指令說明：

- S₁** 為 Socket 模式，欲開啟 TCP Socket 時，**S₁** 為 1，開啟 UDP Socket 時，**S₁** 為 0，**S₂** 為 Socket 編號對應乙太網路中 Socket 設定，範圍為 1~8。
- 使用本指令前，需先以 SOPEN 開啟 Socket 連線，當已連線旗標 (TCP) 或是已開啟旗標 (UDP) 為 ON 時，方可使用本指令進行資料傳輸。
- 若 SOPEN 開啟連線時開啟模式 (**S₃**) 設定為 2 或 3，PLC 接收位址第一個位置將會儲存接收長度，第二個位置開始存放接收資料。未將第一個位置的長度值清除，接收到新的資料時，將會順序的往後放，直到將第一個位置的值被寫為 0 或超出最大接收資料長度。
- 透過 TCP Socket 接收資料時，當此指令執行後，若無錯誤，則開始接收資料，接收中旗標為 ON，若接收成功則已接收旗標為 ON，接收中旗標為 OFF，反之若有錯誤則對應錯誤旗標 ON。

TCP Socket 編號	接收中旗標	已接收旗標	錯誤旗標
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

- 透過 UDP Socket 接收資料時，當此指令執行後，若無錯誤，則資料接收中旗標為 ON，等到收到資料後，已接收旗標為 ON，反之若有錯誤則對應錯誤旗標 ON。

UDP Socket 編號	接收中旗標	已接收旗標	錯誤旗標
1	SM1337	SM1335	SM1338
2	SM1342	SM1340	SM1343
3	SM1347	SM1345	SM1348
4	SM1352	SM1350	SM1353
5	SM1357	SM1355	SM1358
6	SM1362	SM1360	SM1363

UDP Socket 編號	接收中旗標	已接收旗標	錯誤旗標
7	SM1367	SM1365	SM1368
8	SM1372	SM1370	SM1373

6. 本指令一般都是使用脈波執行型指令 (SRCVDP) 。

程式範例：

請參考 SOPEN 程式範例。

補充說明：

1. TCP 接收錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600D	RJ45 埠未連接	SM1100
16#6206	TCP 通訊接口 (Socket) 功能的傳送資料位址不合法	註 1
16#6207	TCP 通訊接口 (Socket) 功能的傳送資料長度不合法	註 1
16#6208	TCP 通訊接口 (Socket) 功能的傳送資料裝置超出範圍	註 1
16#6212	遠端裝置回應逾時 (Socket)	註 1
16#6213	接收資料超過限制 (Socket)	註 1
16#6214	遠端裝置拒絕連線 (Socket)	註 1
16#6215	目前通訊接口 (Socket) 未開啟	註 1
16#6219	目前通訊接口 (Socket) 接收中	註 1

註 1：

SM1277：TCP Socket 1 錯誤旗標。

SM1285：TCP Socket 2 錯誤旗標。

~

SM1325：TCP Socket 7 錯誤旗標。

SM1333：TCP Socket 8 錯誤旗標。

2. UDP 接收錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600D	RJ45 埠未連接	SM1100
16#620F	UDP 通訊接口 (Socket) 功能的傳送資料位址不合法	註 2
16#6210	UDP 通訊接口 (Socket) 功能的傳送資料長度不合法	註 2
16#6211	UDP 通訊接口 (Socket) 功能的傳送資料裝置超出範圍	註 2
16#6213	接收資料超過限制 (Socket)	註 2
16#6219	目前通訊接口 (Socket) 接收中	註 2

註 2 :

SM1338 : UDP Socket 1 錯誤旗標。

SM1343 : UDP Socket 2 錯誤旗標。

~

SM1368 : UDP Socket 7 錯誤旗標。

SM1373 : UDP Socket 8 錯誤旗標。

6

API	指令碼			運算元							功能									
2203		SCLOSE	P	$S_1 \cdot S_2$							關閉 Socket									
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF			
S_1	●	●			●	●		●	●			○	●	○	○					
S_2	●	●			●	●		●	●			○	●	○	○					
										脈波執行型			16 位元指令 (5 Steps)				32 位元指令			
										AH500			AH500				-			

符號：

SCLOSE		SCLOSEP		S_1 : Socket 模式	Word
En		En			
S1		S1		S_2 : Socket 編號	Word
S2		S2			

指令說明：

- S_1 為 Socket 模式，欲關閉 TCP Socket 時， S_1 為 1，關閉 UDP Socket 時， S_1 為 0； S_2 為 Socket 編號對應乙太網路中 Socket 設定。
- $S_1=0\sim 1$ ， $S_2=1\sim 8$ 。
- 關閉 Socket 前必須確認此 Socket 為連線狀態，否則此指令不會執行。
- TCP Socket 若由 Client 端執行關閉指令，則 Server 端會繼續對本機通訊埠進行監聽（連線中旗標為 ON）；若由 Server 端執行關閉指令，則 Server 端在關閉完成後也取消對本機通訊埠的監聽，除上述之外，關閉後對應的旗標皆為 OFF。
- UDP Socket 執行關閉指令後，對應的旗標皆為 OFF。
- 關閉 TCP Socket 時，當此指令執行後，若無錯誤，則開始關閉與遠端通訊埠的連線，關閉中旗標為 ON，若關閉成功則關閉中旗標為 OFF，反之若有錯誤則對應錯誤旗標 ON。

Socket 編號	關閉中旗標	錯誤旗標
1	SM1274	SM1277
2	SM1282	SM1285
3	SM1290	SM1293
4	SM1298	SM1301
5	SM1306	SM1309
6	SM1314	SM1317
7	SM1322	SM1325
8	SM1330	SM1333

- 關閉 UDP Socket 時，當此指令執行後，若無錯誤，則已連線旗標為 OFF，反之若有錯誤則對應錯誤旗標 ON。

Socket 編號	錯誤旗標
1	SM1338
2	SM1343
3	SM1348
4	SM1353
5	SM1358

Socket 編號	錯誤旗標
6	SM1363
7	SM1368
8	SM1373

8. 本指令一般都是使用脈波執行型指令 (SCLOSEP) 。

程式範例：

請參考 SOPEN 程式範例。

補充說明：

1. TCP 關閉錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600D	RJ45 埠未連接	SM1100
16#6212	遠端裝置回應逾時 (Socket)	註 1
16#6214	遠端裝置拒絕連線 (Socket)	註 1
16#621A	目前通訊接口 (Socket) 關閉中	註 1

註 1：

SM1277：TCP Socket 1 錯誤旗標。

SM1285：TCP Socket 2 錯誤旗標。

~

SM1325：TCP Socket 7 錯誤旗標。

SM1333：TCP Socket 8 錯誤旗標。

6

2. UDP 關閉錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#600D	RJ45 埠未連接	SM1100
16#6212	遠端裝置回應逾時 (Socket)	註 2
16#6214	遠端裝置拒絕連線 (Socket)	註 2
16#621A	目前通訊接口 (Socket) 關閉中	註 2

註 2：

SM1338：UDP Socket 1 錯誤旗標。

SM1343：UDP Socket 2 錯誤旗標。

~

SM1368：UDP Socket 7 錯誤旗標。

SM1373：UDP Socket 8 錯誤旗標。

API	指令碼		運算元							功能						
2204	MSEND	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							寄送電子郵件						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S_1	●	●			●	●		●	●			○	●	○	○		
S_2	●	●			●	●		●	●			○	●				
S_3	●	●			●	●		●	●			○	●				
D	●	●	●	●				●	●				●				

脈波執行型	16 位元指令 (9 steps)	32 位元指令
AH500	AH500	-

符號：

MSEND		MSENDP			
En		En		S_1 ：	郵件組態 Word
S1	D	S1	D	S_2 ：	郵件主旨 Word
S2		S2		S_3 ：	郵件信息 Word
S3		S3		D：	指令完成旗標 Bit

指令說明：

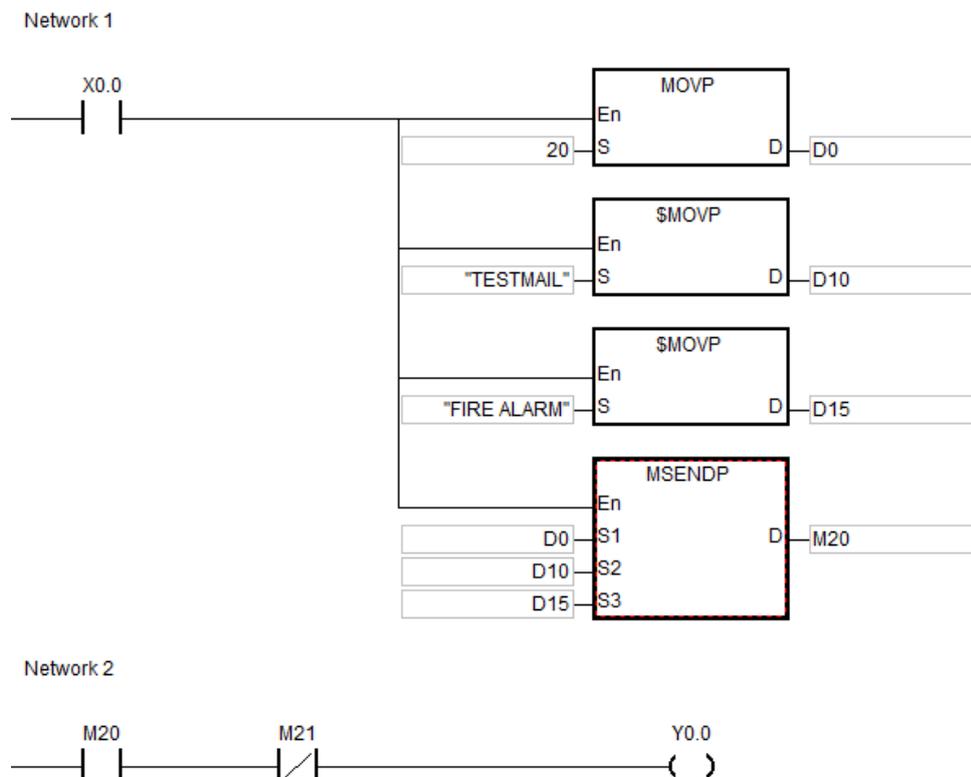
- 利用 S_1 、 S_2 和 S_3 的設定轉為所需的資料，寄出 Email。
- 使用本指令前，需用 ISPSOFT 完成以下設定。
 - 在 PLC 參數設定 -> 基本設定中設定本機的 IP 位址、網路遮罩。
 - 在 PLC 參數設定 -> 進階設定 -> 郵件下設定郵件伺服器的 IP 位址、通訊埠、本機郵件位址和主旨。
 - 在 PLC 參數設定 -> 進階設定 -> 郵件位址與發送條件設定 -> 郵件位址設定要欲寄送的郵件位置。
 - 若外寄伺服器需要身分驗證，可在 PLC 參數設定 -> 進階設定 -> 郵件下設定使用者名稱和密碼。
- 郵件設定如下：

運算元	描述	設定範圍
S_1	遠端郵件編號	1-256。 bit0~bit7 分別代表在 ISPSOFT 中設定的遠端郵件位址 (ISPSOFT 中郵件編號為 8 組)。bit0 代表遠端位址 1，以此類推。 若欲傳送至此郵件位址，須建立對應的位元。
S_2	主旨	郵件主旨最大支援 16 字元。
S_3	信息	郵件信息最大支援 64 字元。
D	指令完成旗標	指令完成後，該位元為 ON。若指令完成異常，下一個位元為 ON。

- 本指令一般都是使用脈波執行型指令 (MSENDP)。

程式範例：

當 X0.0=ON 時，寄一封郵件給遠端郵件編號 3 和編號 5。(D0=00010100)。當與 SMTP 伺服器通訊完畢後 M20=ON，若通訊過程中無錯誤 M21=OFF，輸出 Y0.0=ON。



補充說明：

1. **D+1** 裝置超出範圍時，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
2. **S₁<1** 或 **S₁>256**，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
3. **S₂** 或 **S₃** 的字串長度，系統以抓取結束字元 (16#00) 為主，若達長度上限仍未讀取結束字元，則字串長度等於上限值。
4. 當執行此指令時，若系統對 SMTP 伺服器的連線已達上限，則會發生錯誤，錯誤碼 16#6100。
5. 若 PLC 參數中設定寄送郵件需經過身分驗證，又輸入的帳號經 SMTP 伺服器驗證後無效，則會發生錯誤，錯誤碼 16#6108；如帳號有效密碼錯誤時，錯誤碼 16#6109。
6. 如遠端郵件位址被 SMTP 伺服器判別為無效，則會發生錯誤，錯誤碼 16#6111。
7. 如寄送過程中，SMTP 伺服器回應逾時，則會發生錯誤，錯誤碼 16#6107，寄送動作取消。
8. **D** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD/INT。
9. S₂ 與 S₃ 之間請保留一個字元，供本指令產生中斷字元使用。
10. 當執行此指令時，若網路線未連接，則 SM1100 為 ON，錯誤碼 16#600D。

API	指令碼		運算元								功能				
2205	EMDRW	P	$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot n$								MODBUS TCP 資料讀寫				

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

脈波執行型	16 位元指令 (11 steps)	32 位元指令
AH500	AH500	-

符號：

EMDRW		EMDRWP		符號	說明
En		En		S_1	連線裝置位址 Word
S1		S1		S_2	通訊功能碼 Word
S2		S2		S_3	欲讀寫資料的位址 Word
S3		S3		S	欲讀寫之資料存放暫存器 Bit/ Word
S		S		n	讀寫資料長度 Word
n		n			

指令說明：

- 使用本指令前，需用 ISPSOft 完成以下設定。
 - 在 PLC 參數設定 -> 基本設定中設定本機的 IP 位址、網路遮罩。
- S_1 設定：

位址	描述	設定範圍
S_1	通訊站號	0-255
S_{1+1}	遠端 IP High Word	範例：遠端 IP 位址=172.16.144.230 $S_{1+1}=16\#AC10$ $S_{1+2}=16\#90E6$
S_{1+2}	遠端 IP Low Word	
S_{1+3}	指令完成後是否關閉連線	0：關閉連線 · 1：保持連線 (依 PLC 參數設定中 TCP 連線保持時間決定連線何時關閉)

S_2 ：通訊功能碼 (Function Code)。

例如：

- AH500 的讀取多筆位元裝置 (Bit) (非 Discrete Input 裝置) 命令為 1 (16#01)。
- AH500 的讀取多筆位元裝置 (Bit) (僅 Discrete Input 裝置) 命令為 2 (16#02)。
- AH500 的讀取多筆字元裝置 (Word) (非 Input Register 裝置) 命令為 3 (16#03)。
- AH500 的讀取多筆字元裝置 (Word) (僅 Input Register 裝置) 命令為 4 (16#04)。
- AH500 的單筆位元裝置 (Bit) 狀態寫入命令為 5 (16#05)。
- AH500 的單筆字元裝置 (Word) 資料寫入命令為 6 (16#06)。
- AH500 的多筆位元裝置 (Bit) 狀態寫入命令為 15 (16#0F)。
- AH500 的多筆字元裝置 (Word) 資料寫入命令為 16 (16#10)。

目前僅支援上述功能碼，其餘功能碼將無法執行。請參考下列程式範例。

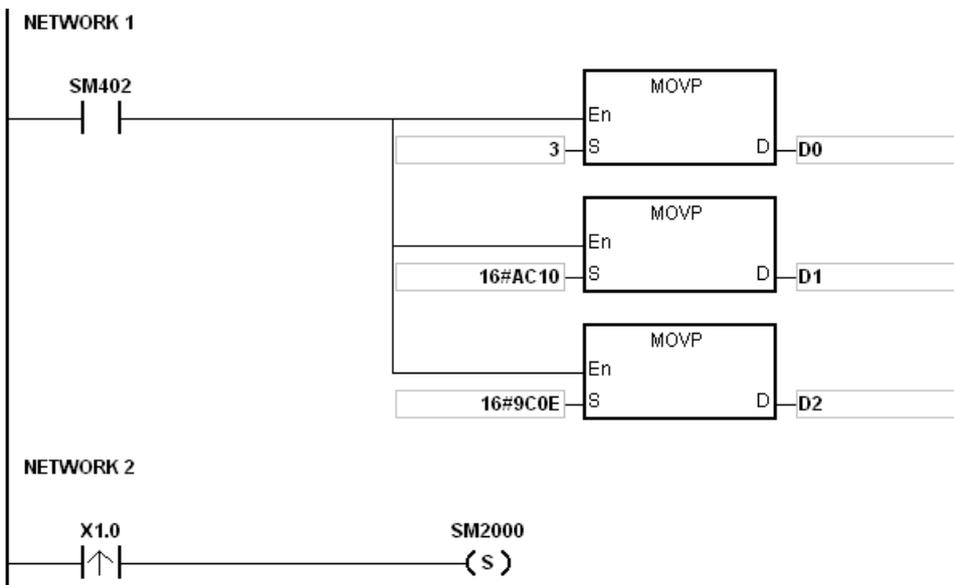
3. **S₃**：欲讀寫資料的位址（裝置位址）。連線裝置的內部裝置位址。
4. **S**：欲讀寫之資料（來源或目的）。由使用者設定暫存器，將欲寫入資料長度的資料事先存入暫存器內。或資料讀取後存放之暫存器。
5. **n**：讀寫資料長度（資料長度）。指定範圍 1~240（Byte）。設定的資料量不可以大於 240 字元。Coil 相關的通訊命令，其資料單位為 Bit，n 的範圍為 1~1920。Register 相關的通訊命令，其資料單位為 Word，則 n 的範圍為 1~120。

旗標 EMDRW	要求旗標	等待中旗標	已接收旗標	錯誤旗標	逾時旗標	連線已關閉旗標
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

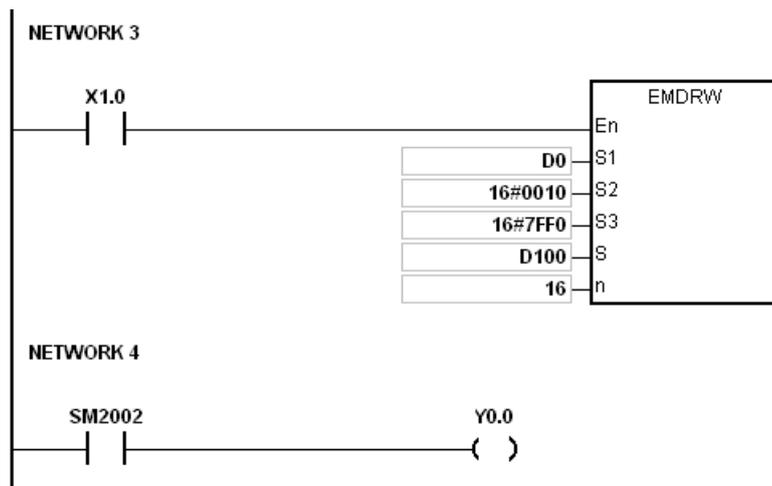
6. 本指令於程式中使用次數並無限制，但是同時間僅有 8 個指令被執行。
7. 若指令執行時，同時有不只一個送信旗標為 ON，則以旗標數字最小者優先執行。
8. 本指令一般都是使用脈波執行型指令（EMDRWP）。

程式範例：

1. 設定遠端站號為 3。



6



補充說明：

1. 當 S2 通訊功能碼指定為 Bit 讀取時 S 必需指定為 Bit 裝置，否則視為運算錯誤，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
2. 當 S2 通訊功能碼指定為 Word 讀寫時 S 必需指定為 Word 裝置，否則視為運算錯誤，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
3. 當通訊命令為 0x05 與 0x06 時，n 無作用，指令只會寫入 1 個 Bit 或 Word，惟 n 仍不能超出讀寫資料長度的範圍。
4. 當指令執行時，若某要求旗標為 ON，但其對應的連線已關閉旗標不為 ON，則系統會自動尋找是否有其他要求旗標為 ON 且其對應的連線已關閉旗標為 ON 來執行指令，若無則此次指令不執行。
5. S1 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [4] of WORD/INT。
6. EMDRW 指令執行錯誤說明

錯誤代碼	說明	錯誤旗標
16#2003	程式中使用的裝置超過可用範圍	SM0/SM5
16#200B	n 運算元或其它 K/H 運算元超出範圍 (錯誤旗標 SM0/SM5)	SM0/SM5
16#600D	RJ45 埠未連接	SM1100
16#6400	連線數超出限制或未設定送信旗標 (EMDRW)	註 1
16#6401	遠端裝置中止連線 (EMDRW)	註 1
16#6402	遠端裝置回應逾時 (EMDRW)	註 1
16#6403	遠端裝置回應逾時 (EMDRW)	註 1
16#6405	MODBUS 回覆訊息的 Byte Count 與實際的資料長度不符 (EMDRW)	註 1

註 1：

SM2003：EMDRW 1 錯誤旗標。

...

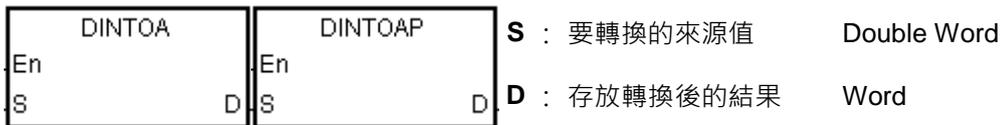
SM2045：EMDRW 8 錯誤旗標。

API	指令碼			運算元								功能					
2206	D	INTOA	P	S · D								IP→字串變換					

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

脈波執行型	16 位元指令	32 位元指令 (5 steps)
AH500	-	AH500

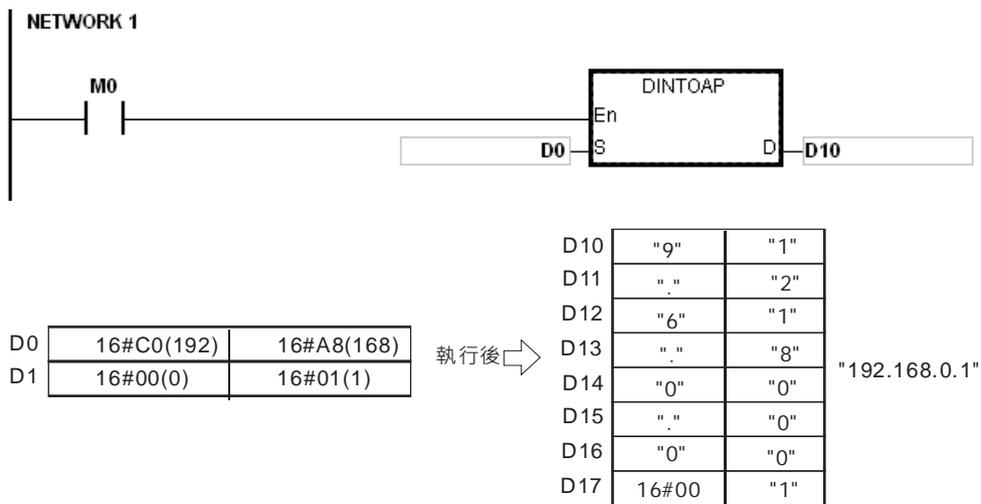
符號：



指令說明：

1. 將 **S** 的內容值轉換成字串型態的 IP 位址，並且將結果存到 **D**。
2. **D** 連續佔用 8 個裝置。

程式範例一：



程式範例二：



C0A80001(Hex) 執行後 →

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"

補充說明：

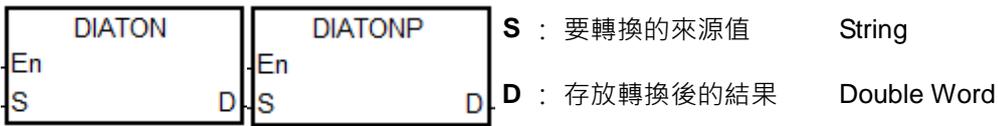
D 運算元，若使用 ISPSoft 宣告，則資料型態為 ARRAY [8] of WORD/INT。

API	指令碼			運算元								功能				
2207	D	IATON	P	S · D								字串→IP 變換				

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

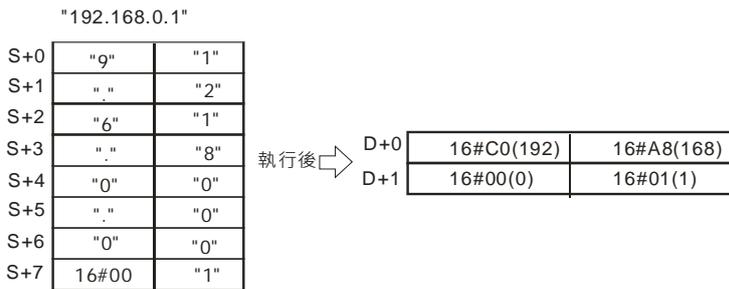
脈波執行型	16 位元指令	32 位元指令 (5-11 steps)
AH500	-	AH500

符號：



指令說明：

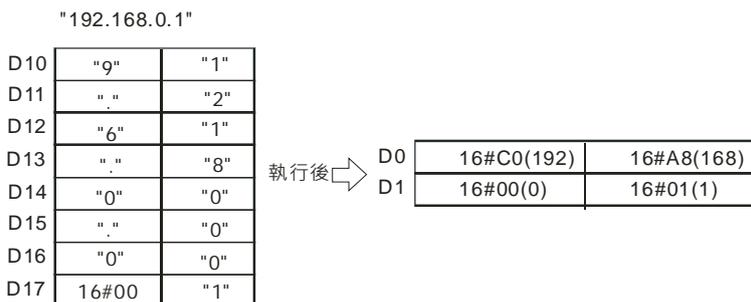
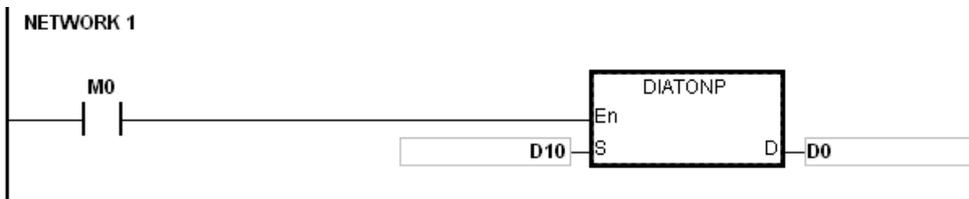
- 將 S 的字串 IP 位址轉換成整數型態，並且將結果存到 D。
- S 連續佔用 8 個裝置。



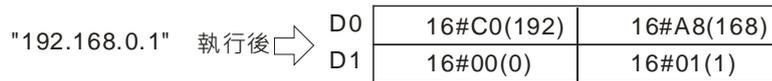
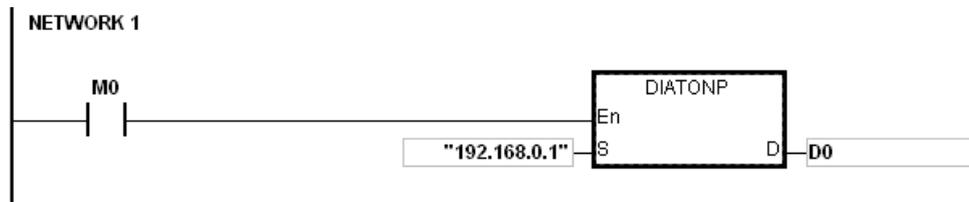
6

- S 的字串 IP 位址來源，每個區段數值固定為 3 個字元，中間以一個"."16#2E 隔開。
- S 的字串 IP 位址來源，每個區段數值轉換出來的結果不能>255。
- 如果 S 是由使用者直接輸入字串，則每個區段不必固定輸入 3 個字元，例如："192.168.0.1"，只要直接輸入即可，不必輸入"192.168.000.001"。

程式範例一：



程式範例二：



補充說明：

1. **S** 字串中沒有 16#00 當結尾，SM0=ON，SR0 錯誤碼為 16#200E。
2. **S** 字串除了小數點，其餘要轉成 BIN 值的部份不在 16#30~16#39 範圍內，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
3. **S** 字串中小數點的位址“.”符號不為 16#2E 時，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
4. **S** 字串中小數點“.”符號的個數不等於 3 個時，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
5. **S** 的字串來源，若有任一個區段的數值轉換出來的結果>255，指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
6. **S** 的字串 IP 位址來源，每個區段數值固定 1~3 個字元，若超過則指令不執行，SM0=ON，SR0 錯誤碼為 16#2003。
7. **S** 運算元，若使用 ISPSOft 宣告，則資料型態為 ARRAY [8] of WORD/INT。

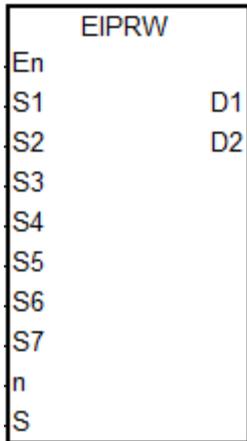
API	指令碼			運算元							功能						
2208	EIPRW			$S_1 \sim S_7 \cdot n \cdot S \cdot D_1 \cdot D_2$							EtherNet/IP 讀寫通訊指令						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
S ₁								●	●								
S ₂								●	●					○	○		
S ₃								●	●					○	○		
S ₄								●	●					○	○		
S ₅								●	●					○	○		
S ₆	●	●	●	●				●	●								
S ₇								●	●					○	○		
n								●	●					○	○		
S								●	●								
D ₁								●	●								
D ₂								●	●								

脈波執行型	16 位元指令 (23 steps)	32 位元指令
-	AH500	-

符號：

- S₁ : IP 位址 Word[2]
- S₂ : 連線模式 Word
- S₃ : 功能碼 Word
- S₄ : 類別 (Class) ID Word
- S₅ : 需求 (Instance) ID Word
- S₆ : 屬性 (Attribute) ID 開關 Bit
- S₇ : 屬性 (Attribute) ID Word
- n 讀寫資料長度 n Word
- S 讀寫資料目的或來源 Word[n]
- D₁ 通訊執行狀態 Word
- D₂ : 錯誤碼 Word[2]



指令說明：

- S₁~S₇ · n · S 與 D₁~D₂ 裝置名稱與說明如下表所示。

裝置	名稱	使用說明
S ₁	IP 位址	第 1 個 word 為前兩組 IP 位址，第 2 個 word 為後兩組 IP 位址。舉例：IP=192.168.1.5，則 S ₁ =16#C0A8 · S ₁ +1=16#0105

裝置	名稱	使用說明
S₂	連線模式	0 : UCMM , 不建立 CIP 連線。 1 : Connected then close CIP connection , 傳送資料後關閉連線。 2 : Connected then keep CIP connection , 傳送資料後保持連線。
S₃	功能碼 (Service Code)	功能碼範圍 16#0000~00FF , 超出此範圍不執行 , 請參照所欲連接的裝置其支援的 Service Code 類型。
S₄	類別 (Class) ID	請參照所欲連接的裝置 , 其支援的 Class 類型。
S₅	實例 (Instance) ID	請參照所欲連接的裝置 , 其支援的 Instance 實例。
S₆	屬性 (Attribute) ID 開關	ON 為啟動 , OFF 為關閉 , 當此欄位為 OFF 時 , 表示此次通訊 S₇ 欄位不使用。
S₇	屬性 (Attribute) ID	請參照所欲連接的裝置 , 其支援的 Attribute 屬性。
n	讀寫資料長度 n	讀寫資料長度設定值 , 單位 bytes , 最大長度限制為 200 bytes
S	讀寫資料目的或來源	發送資料的來源位置 , 或者接收資料的存放位置 , 為長度 n 個 Word
D₁	通訊執行狀態	0 : 未觸發執行 1 : 通訊執行中 2 : 通訊完成且無發生錯誤 3 : 通訊發生錯誤 4 : 參數設定錯誤
D₂	錯誤碼	D ₂ 紀錄主錯誤碼 , D ₂ +1 記錄延伸錯誤碼

- 指令初次被啟動時 , 即表示發送此通訊指令 , 若此時已達最大連線數時 , 則 **D₁** 通訊執行狀態為 0 , 表示還未被觸發執行。當設定的參數數值超出限制範圍時 , 此通訊指令不執行 , 並且設定 **D₁** 通訊執行狀態為 4。

補充說明 :

- S₁** 運算元使用 ISPSOFT 宣告 , 則資料型態為 ARRAY [2] of WORD。
- 如不需提供 Attribute 屬性時 , 則 **S₆** 運算元為 OFF , 如需提供 Attribute 屬性時 , 則 **S₆** 運算元為 ON。
- S** 運算元使用 ISPSOFT 宣告 , 則資料型態請以讀寫長度 n 設 ARRAY [n] of WORD。

4. **n** 運算元在寫入所連結的裝置內的 Object 時，必須正確的填寫此 Object 的資料長度，在讀取裝置內的 Object 時，則可以填寫 0 或是小於等於此 Object 的資料長度。(0：自動填入此 Object 資料長度)
5. 如欲寫入 2 Bytes 資料長度至所連結的裝置，則 **n** 運算元為 2，並以 **S** 運算元內第一個 Word 的 Low Byte 為第一筆資料，而 High Byte 為第二筆資料。
6. **D₂** 運算元使用 ISPSOft 宣告，則資料型態為 ARRAY [2] of WORD。
7. 指令執行錯誤說明：

錯誤碼	錯誤旗標	錯誤說明
16#2003	SM0 / D ₁	(1) S ₂ 、S ₃ 內容值超出範圍 (2) S + n 超出裝置位址範圍
16#200B	SM0	n 運算元不在範圍
16#600D	SM1100	Ethernet 網路未連接
16#6701	D ₁	遠端設備中斷連線
16#6702	D ₁	遠端設備回應逾時
16#6703	D ₁	IP 位址不合法
16#6704	D ₁	回應命令 Service Code 錯誤
16#6705	D ₁	回應命令長度錯誤
16#6706	D ₁	所有連線忙碌中

8. D2 主錯誤碼狀態說明

狀態碼	狀態名稱	錯誤處置
16#00	成功	
16#01	連線錯誤	檢查從站 EDS File 內容是否正確
16#02	資源不可用	1. 檢查主站連接設備是否超過限制 2. 檢查從站連接設備是否超過限制
16#03	參數設定錯誤	檢查讀寫資料內容 (S) 是否正確
16#04	路徑段落錯誤	檢查類別 ID (S ₄)、需求 ID (S ₅) 和屬性 ID (S ₇) 設定是否正確
16#05	目的路徑不存在	檢查類別 ID (S ₄)、需求 ID (S ₅) 和屬性 ID (S ₇) 設定是否正確
16#07	連線中斷	1. 檢查從站設備 Ethernet 端口連接是否正確 2. 檢查從站設備 Keep alive timer 設定
16#08	服務碼不支援	檢查功能碼 (S ₃) 設定是否正確
16#09	錯誤屬性值	檢查讀寫資料來源 (S) 與內容設定是否正確
16#0A	屬性列表錯誤	檢查從站設備物件屬性是否允許執行 Get_Attribute_List 和 Set_Attribute_List 功能碼

狀態碼	狀態名稱	錯誤處置
16#0B	發送衝突	檢查服務是否有重複設定
16#0C	物件狀態衝突	檢查是否有建立 Owner IO 連線
16#0D	物件已存在	檢查從站是否已支援設定物件，若是已支援則不需執行服務
16#0E	屬性不可寫	檢查物件屬性是否支援寫入功能
16#0F	權限不足	檢查從站設備是否允許執行此服務碼
16#10	設備目前狀態無法執行服務	檢查是否有建立 Owner IO 連線
16#11	回應資料長度過大	檢查物件屬性資料長度是否超過限制（100 words）
16#12	存取 Tag 時使用，存取資料長度與資料型態不符	檢查存取資料長度與資料型態設定是否正確
16#13	傳送資料不足	檢查讀寫長度（n）設定是否正確
16#14	屬性值不支援	檢查屬性 ID 開關（S ₆ ）和屬性 ID（S ₇ ）設定是否正確
16#15	傳送資料過多	檢查讀寫長度（n）設定是否正確
16#16	物件不存在	檢查類別 ID（S ₄ ）設定是否正確
16#17	存取 Tag 時使用，存取資料順序錯誤	1. 檢查 Ethernet 網路連接是否正確 2. 檢查 Ethernet 通訊環境是否發生封包遺失
16#18	屬性值未儲存	檢查從站設備狀態是否發生錯誤
16#19	屬性值執行儲存錯誤	檢查從站設備硬體是否發生錯誤
16#1A	路由錯誤，需求封包長度超過	檢查讀寫資料長度（n）是否超過路由器限制
16#1B	路由錯誤，回應封包長度超過	檢查讀寫資料長度（n）是否超過路由器限制
16#1F	自定義物件存取錯誤	請參考從站設備錯誤定義
16#20	不合法的參數	檢查讀寫資料內容（S）是否正確

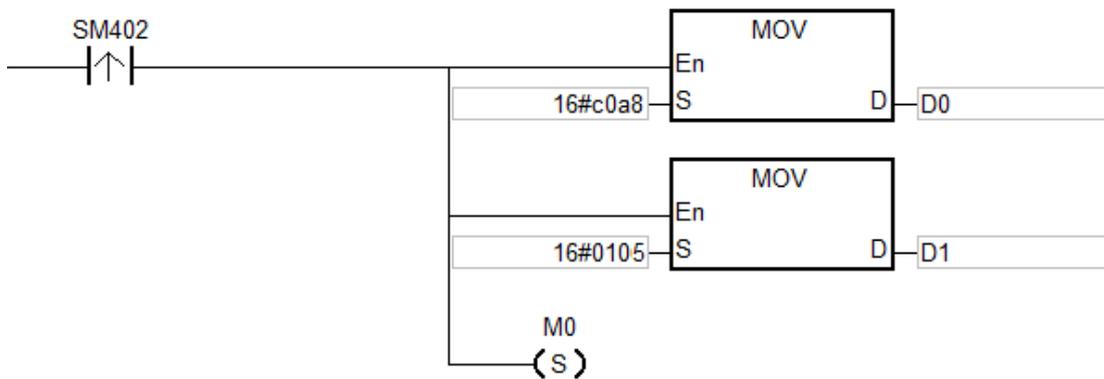
9. D2+1 延伸錯誤碼，依連接從站裝置有不同的錯誤碼定義，如果從站為台達相關設備，可參考 EtherNet/IP 操作手冊第 6.2 節錯誤碼及排除方法。

程式範例一：

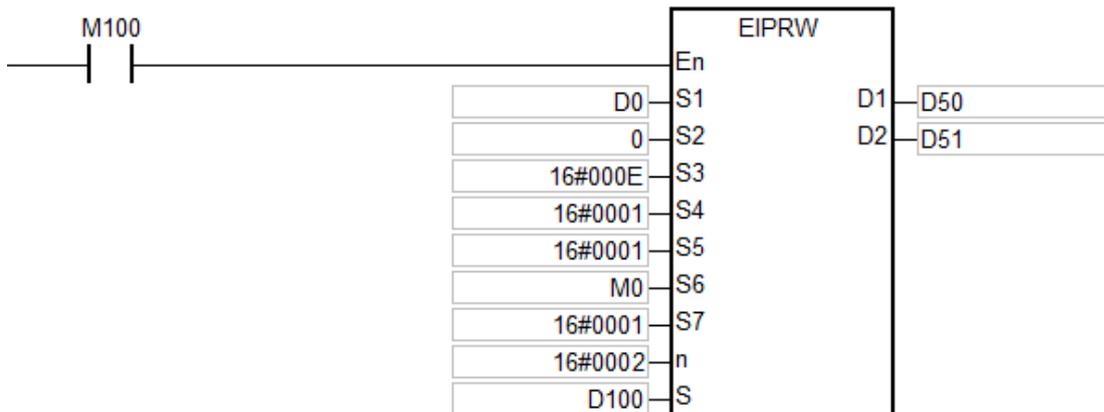
讀取設備 (192.168.1.5) 的廠商代碼，資料讀取結果存放於 D100，EtherNet/IP Object 參數如下：

- (1) Service Code = 16#000E (對單個屬性進行讀取)。
- (2) Identity Object : Class ID = 16#0001。
- (3) 廠商代碼 : Instance ID = 1 與 Attribute ID = 1。

Network 1



Network 2

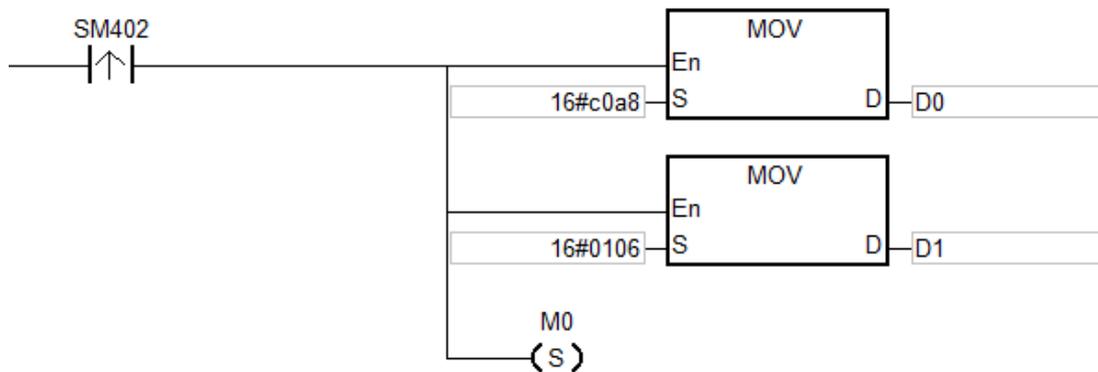


程式範例二：

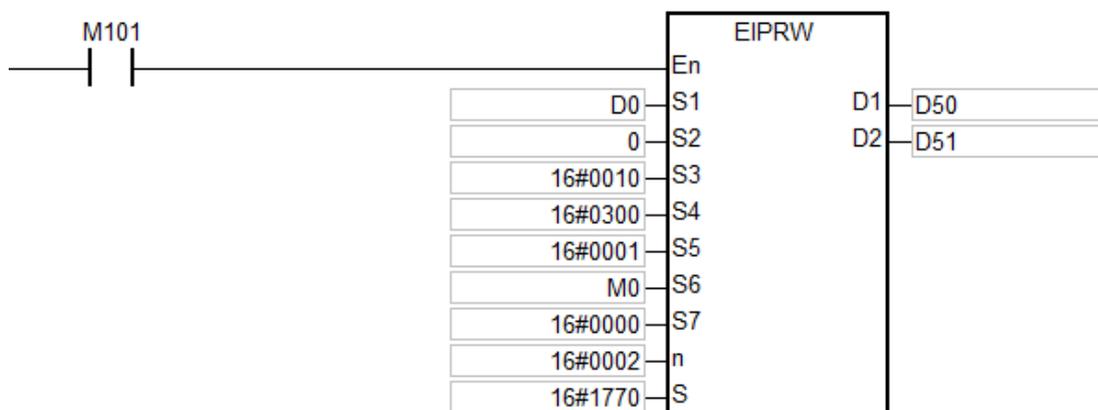
設定變頻器 (192.168.1.6) 最高操作頻率數值 (01-00) 為 60.00Hz，EtherNet/IP Object 參數定義如下：

- (1) Service Code = 16#0010 (對單個屬性進行寫入)。
- (2) 自定義 Object : Class ID = 16#0300。
- (3) 最高操作頻率數值 : Instance ID = 16#0001，Attribute ID = 16#0000。

Network 1



Network 2



補充說明：

EIPRW 指令支援機種說明如下：

1. AHCPU5x1-EN V2.02 以後
2. AHCPU560-EN2 V1.10 以後

6.23 記憶卡讀寫指令

6.23.1 記憶卡讀寫指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>2300</u>	MWRIT	–	✓	將 PLC 數據資料寫入到記憶卡裡	13
<u>2301</u>	MREAD	–	✓	將數據資料從記憶卡裡讀取到 PLC	13
<u>2302</u>	MTWRIT	–	✓	將文字寫入到記憶卡裡	11

6.23.2 記憶卡讀寫指令說明

API	指令碼			運算元								功能											
2300		MWRIT	P	C	S	S ₁	S ₂	S ₃	S ₄									將 PLC 數據資料寫入到記憶卡裡					

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	"\$"	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●				
S₄								●	●				●	○	○		

脈波執行型	16 位元指令(13 steps)	32 位元指令
AH500	AH500	-

符號：

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** : 控制參數 Word
- S** : 資料來源 Word
- S₁** : 欲寫入的資料長度 Double Word
- S₂** : 換行 Word
- S₃** : 檔案名稱 Word
- S₄** : 檔案內的資料位址 Double Word

指令說明：

1. 運算元說明：

- 控制參數 **C**：



說明項目	代碼	說明
檔案資料格式	0	二進制。預設值。
		副檔案名稱為.dmd。
		每個數據的單位為 Word。
	1	每個數據之間用逗點做為區隔。
		每個數據的單位為 Word。
		副檔案名稱為.csv。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。

6

說明項目	代碼	說明
檔案資料格式	2	每個數據之間用逗點做為區隔。
		每個數據的單位為 DWord。
		副檔案名稱為.csv。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
	3	每個數據之間用 TAB 做為區隔。
		每個數據的單位為 Word。
		副檔案名稱為.txt。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
	4	每個數據之間用 TAB 做為區隔。
		每個數據的單位為 DWord。
		副檔案名稱為.txt。
		採用 ASCII 的字元。
	5	數據與數據之間沒有區隔符號。
		每個數據的單位為 Word。
		副檔案名稱為.txt。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
	6	數據與數據之間沒有區隔符號。
每個數據的單位為 DWord。		
副檔案名稱為.txt。		
採用 ASCII 的字元。		
功能選項	0	附加 (Append)。將使用者欲寫入的數據，從檔案的最後一個數據開始寫入。預設值。
		若檔案不存在，則會自動產生。
	1	覆寫 (overwrite)。將使用者欲寫入的數據，從指定的檔案的數據位址開始寫入。
		若檔案不存在，則會自動產生。
保留	-	固定為 0。

- 資料來源 **S**：資料來源的起始位址。
- 欲寫入的資料長度 **S₁**：用來設定欲寫入檔案的數據資料長度，0 表示不寫入。

項目	說明
計算單位	計算的單位需依照運算元 C 的檔案資料格式來做為辨別，若檔案資料格式為 0、1、3 或 5 (請參考運算元 C)，則計算單位為 Word；若檔案資料格式為 2、4 或 6 (請參考運算元 C)，則計算單位為 DWord。
參數單位	DWord。
數據資料長度	設定長度不可以超出該裝置範圍 (有關裝置範圍請參考 Ch2 裝置說明)，檔案寫入的資料量不可以超出 4GB。

- 換行 **S₂**: 可用來設定寫入多少數據 (Word/DWord) 後，換到下一行，可設定範圍為 0~256。(0 表示不換行)
- 檔案名稱 **S₃~S₃+4**: **S₃** 佔用連續 5 個裝置來指定寫入資料的檔案名稱，且檔案名稱不可以超過九個字元 (包含終止位元 16#00)，如果沒有填入字串結束字元會有錯誤。當有讀取到終止位元時，將會停止讀取字元，且檢查檔案名稱是否合法。合法命名的字元為 A~Z、a~z 與 0~9。副檔案名稱會依據 **C** 的檔案資料格式，而自動給於副檔案名稱，生成的檔案會在預設資料夾裡。若檔案名稱為 "Test1"，則寫入裝置樣式如下：

S3	'e'	'T'	ASCII代碼 ➔	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

- 預設資料夾路徑：

機種名稱	資料夾路徑
AHCPU530-RS2	PLC CARD\AH500\UserProg
AHCPU530-EN	
AHCPU530-EN/RM	
AHCPU533-EN	

- 檔案內的資料位址 **S₄**: 用來指定寫入檔案的起始位址。

說明項目	說明
計算單位	計算的單位需依照運算元 C 的檔案資料格式來做為辨別，若檔案資料格式為 0、1、3 或 5，則計算單位為 Word；若檔案資料格式為 2、4 或 6，則計算單位為 DWord。(請參考運算元 C)
S₄ 的參數單位	此參數單位為 Dword。
S₄ 使用方式	若 C 的功能項目為 0 (附加)，則此運算元不會被使用到。
	若 C 的功能項目為 1 (覆寫)，則會將數據寫入到 S₄ 指定在檔案裡的數據位址。
	此 S₄ 指所定的數據位址不可以超出檔案所擁有的數據總個數。當 S₄ 為 0 時，表示為第一個位置。

2. 相關旗標

旗標	說明
SM450	若記憶卡有存在於 CPU module 上，則此旗標會為 ON 的狀態。

旗標	說明
SM451	記憶卡在寫保護的模式裡
SM452	記憶卡正在讀/寫的動作
SM453	當記憶卡有發生錯誤的時候，此 SM 會設定為 ON。當此 SM 為 ON 時，需自行設定為 OFF。錯誤代碼請參考 SR453。

相關錯誤代碼 (SR453)

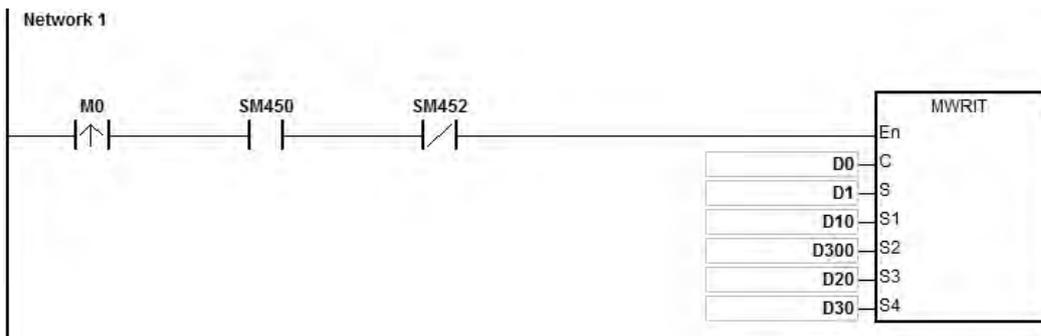
錯誤代碼	說明
16#005E	當 CPU module 在進行記憶卡初始化流程時，發生錯誤而無法進行初始化流程。
16#005F	路徑錯誤或檔案不存在。
16#0060	無法產生預設資料夾。
16#0061	記憶卡記憶體空間不足。
16#0062	記憶卡為防寫保護。
16#0063	當資料寫入到檔案時，發生錯誤。
16#0064	當要讀取檔案資料時，無法讀取檔案的資料。
16#0065	檔案屬性為唯讀狀態。

- 當寫入檔案資料格式為 0 時，則讀取檔案資料格式也一定要為 0，不然無法讀取，且 SM453 會 ON 起來。其他檔案資料格式依此類推。

程式範例：

當記憶卡有插入 CPU module 的時候，SM450 為 ON 的狀態；當 MWRIT 開始運作的時候，SM452 為 ON；當 MWRIT 完成寫入動作的時候，SM452 為 OFF。可使用脈波指令 MWRITP，避免指令持續執行，持續寫入記憶卡，導致寫入次數超過記憶卡限制而記憶卡故障。

6



運算元	設定值	說明
D0	16#0011	寫入檔案。 檔案資料格式：每個數據之間有用逗點來作為區隔。每個數據的單位為 Word。副檔案名稱為.csv。採用 ASCII 的字元。
D1	-	寫入資料內容。
D10、D11	16#00000030	總共寫入長度為 48 個 Word。
D300	16#000A	每寫入 10 個 Word 換下一行。

運算元	設定值	說明
D20	D20=16#6554 D21=16#7473 D22=16#0031	寫入檔案名稱為“Test1”。
D30、D31	16#00000000	從檔案最初的位址開始寫入。

補充說明：

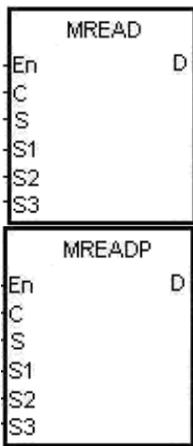
1. **C** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. **S₁** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. **S₂** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
4. **S₃** 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。

API	指令碼			運算元							功能						
2301		MREAD	P	C · S · S₁ · S₂ · S₃ · D							將數據資料從記憶卡裡讀取到 PLC						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●	○	○		
D								●	●				●				

脈波執行型	16 位元指令 (13 steps)	32 位元指令
AH500	AH500	-

符號：



- C** : 控制參數 Word
- S** : 檔案名稱 Word
- S₁** : 檔案的資料位址 Double Word
- S₂** : 系統內容使用 Word
- S₃** : 欲讀取的資料長度 Double Word
- D** : 資料目的地 Word

指令說明：

1. 運算元說明：

- 控制參數 **C**：



說明項目	代碼	說明
檔案資料格式	0	二進制 (0 為預設值)。
		副檔案名稱為 .dmd。
		每個數據的單位為 Word。
	1	每個數據之間用逗點做為區隔。
		每個數據的單位為 Word。
		副檔案名稱為 .csv。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。

說明項目	代碼	說明
檔案資料格式	2	每個數據之間用逗點做為區隔。
		每個數據的單位為 DWord。
		副檔案名稱為.csv。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
	3	每個數據之間用 TAB 做為區隔。
		每個數據的單位為 Word。
		副檔案名稱為.txt。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
	4	每個數據之間用 TAB 做為區隔。
		每個數據的單位為 DWord。
		副檔案名稱為.txt。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
	5	數據與數據之間沒有區隔符號。
		每個數據的單位為 Word。
		副檔案名稱為.txt。
		採用 ASCII 的字元。
		儲存的數據採用 16 進制來表示。
6	數據與數據之間沒有區隔符號。	
	每個數據的單位為 DWord。	
	副檔案名稱為.txt。	
	採用 ASCII 的字元。	
	儲存的數據採用 16 進制來表示。	
功能項目	0	讀取指定檔案內的數據資料 (0 為預設值)。
	1	計算檔案內有多少個 Words/DWords。此計算結果會寫入 D 與 D+1 裡。 若在運算元 C 中的檔案資料格式為 0、1、3、5，則計算單位為 Word；若檔案資料格式為 2、4、6，則計算單位為 DWord。
保留	-	固定為 0

- 檔案名稱 **S~S+4** : **S** 佔用連續 5 個裝置來指定寫入資料的檔案名稱，且檔案名稱不可以超過九個字元 (包含終止位元 16#00)，如果沒有填入字串結束字元會有錯誤。當有讀取到終止位元時，將會停止讀取字元，且檢查檔案名稱是否合法。合法命名的字元為 A~Z、a~z 與 0~9。副檔案名稱會依據 **C** 的檔案資料格式，而自動給於副檔案名稱，生成的檔案會在預設資料夾裡。若檔案名稱為“Test1”，則寫入裝置樣式如下：



- 預設資料夾路徑：

機種名稱	資料夾路徑
AHCPU530-RS2	PLC CARD\AH500\UserProg
AHCPU530-EN	
AHCPU530-EN/RM	
AHCPU533-EN	

- 檔案的資料位址 **S₁**：用來指定讀取檔案的起始位址。

項目	說明
計算單位	計算的單位需依照運算元 C 的檔案資料格式來做為辨別，若檔案資料格式為 0、1、3 或 5，則計算單位為 Word；若檔案資料格式為 2、4 或 6，則計算單位為 DWord。(請參考運算元 C)
參數單位	此參數單位為 DWord。
使用方式	所設定的起始數據位址不可以超過檔案內所擁有的數據總個數。當 S₁ 為 0 時，表示為第一個位置。

- 欲讀取的資料長度 **S₃**：設定的數據資料長度不可以超出裝置最大範圍。若超出檔案所記載的數據 (Word/DWord) 個數，讀取的數據各數會以檔案記載的為主，0 表示不讀取，單位為 DWord。
- 資料目的地 **D**：數據 (Word/DWord) 儲存的起始裝置位址。

6

2. 相關旗標

旗標	說明
SM450	若記憶卡有存在於 CPU module 上，則此旗標會為 ON 的狀態。
SM451	記憶卡在寫保護的模式裡。
SM452	記憶卡正在讀/寫的動作。
SM453	當記憶卡有發生錯誤的時候，此 SM 會設定為 ON。當此 SM 為 ON 時，需自行設定為 OFF。錯誤代碼請參考 SR453。

相關錯誤代碼 (SR453)

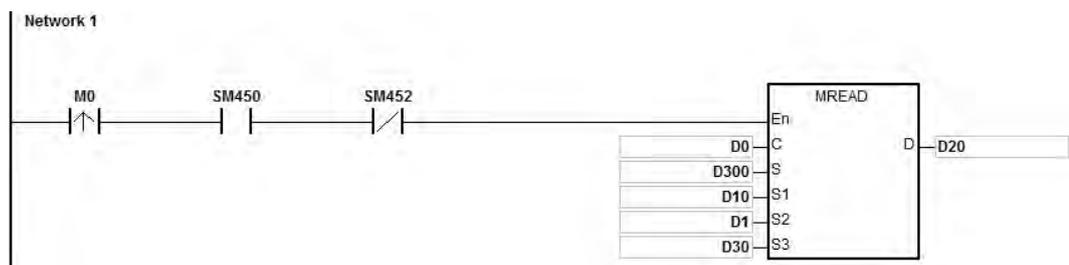
錯誤代碼	說明
16#005E	當 CPU module 再進行記憶卡初始化流程時，發生錯誤而無法進行初始化流程。
16#005F	路徑錯誤或檔案不存在。
16#0060	無法產生預設資料夾。
16#0061	記憶卡記憶體空間不足。
16#0062	記憶卡為防寫保護。

錯誤代碼	說明
16#0063	當資料寫入到檔案時，發生錯誤。
16#0064	當要讀取檔案資料時，無法讀取檔案的資料。

3. 當寫入檔案資料格式為 0 時，則讀取檔案資料格式也一定要為 0，不然無法讀取，且 SM453 會 ON 起來。其他檔案資料格式依此類推。

程式範例：

當記憶卡有插入 CPU module 的時候，SM450 為 ON。當 MREAD 開始運作的時候，SM452 為 ON 的狀態，當 MREAD 完成讀取動作的時候，SM452 為 OFF。



運算元	設定值	說明
D0	16#0011	讀取檔案資料 檔案資料格式：每個數據之間有用逗點來作為區隔。每個數據的單位為 Word。副檔案名稱為.csv。採用 ASCII 的字元。
D300	D300=16#6554 D301=16#7473 D302=16#0031	讀取檔案名稱為“Test1”
D10、D11	16#00000000	從檔案最初的位址開始讀取
D1	16#000A	每讀取 10 個 Word 換下一行
D30、D31	16#00000020	總共讀取長度為 32 個 Word
D20	-	將讀取的資料儲存至 D20

補充說明：

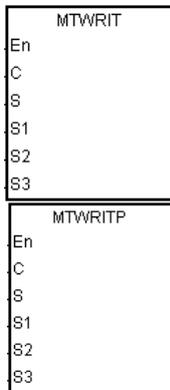
1. C 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. S₂ 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. S₃ 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
4. D 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。

API	指令碼			運算元							功能						
2302		MTWRIT	P	C · S · S₁ · S₂ · S₃							將文字寫入到記憶卡裡						

裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
C								●	●				●	○	○		
S								●	●				●				
S₁								●	●				●	○	○		
S₂								●	●				●	○	○		
S₃								●	●				●				

脈波執行型	16 位元指令 (11 steps)	32 位元指令
AH500	AH500	-

符號：



- C** : 控制參數 Word
- S** : 資料來源 Word
- S₁** : 資料長度 Word
- S₂** : 分隔符號 Word
- S₃** : 檔案名稱 Word

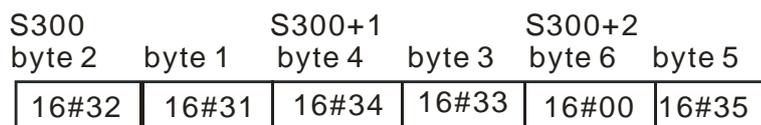
指令說明：

1. 運算元說明：

- 控制參數 **C**：

參數數值	說明
0 (append)	若檔案存在，將裝置的資料寫在檔案的最後一個 byte。
	若檔案不存在，會自動產生出來。
1 (overwrite)	若檔案存在，則新的資料會覆蓋到舊的資料，檔案的大小以新的資料為主。
	若檔案不存在，會自動產生出來。

- 資料來源 **S**：指定要寫入記憶卡的裝置資料來源。若欲寫入“12345”的字串到文字檔裡的話，則儲存在裝置裡的資料如下圖所示。由於以 byte 做為基本單位，所以儲存在 Word 的時候，第一個 byte 會在 Word 的 low byte 裡，第二個 byte 會在 Word 的 High byte 裡，其於以此類推。最後一個 byte 為終止符號“16#00”，表示寫入結束。



- 資料長度 **S₁**：用來決定寫入到記憶卡的資料長度，單位為 byte。不能超過指定裝置的最大範圍。每次寫入的長度不可以超過 255 bytes，0 表示不寫入。

- 分隔符號 S_2 ：每次寫完資料到記憶卡的時候，會將此符號加入到資料裡的最後一個 byte 的下兩個 bytes。例： $S_1=N$ ， S_2 分隔符號之情況如下方表格，則 S_2 分隔符號寫入到記憶卡的情況會如下方表格之說明。

S_2 運算元		說明
High byte	Low byte	
16#00	16#00 或非 16#00	寫入 N 個 Bytes 到檔案裡。
非 16#00	16#00	寫入 N+1 個 Bytes 到檔案裡，其中 N+1 為 S_2 運算元的 High Byte。
非 16#00	非 16#00	寫入 N+2 個 Bytes 到檔案裡，其中 N+1 為 S_2 運算元的 High Byte，N+2 為 S_2 運算元的 Low Byte。

- 檔案名稱 $S_3\sim S_3+4$ ： S_3 佔用連續 5 個裝置來指定寫入資料的檔案名稱，且檔案名稱不可以超過九個字元（包含終止位元 16#00），如果沒有填入字串結束字元會有錯誤。當有讀取到終止位元時，將會停止讀取字元，且檢查檔案名稱是否合法。合法命名的字元為 A~Z、a~z 與 0~9。副檔案名稱會依據 C 的檔案資料格式，而自動給於副檔案名稱，生成的檔案會在預設資料夾裡。若檔案名稱為“Test1”，則寫入裝置樣式如下：

S_3	'e'	'T'	ASCII代碼	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

- 預設資料夾路徑：

機種名稱	資料夾路徑
AHCPU530-RS2	PLC CARD\AH500\UserProg
AHCPU530-EN	
AHCPU530-EN/RM	
AHCPU533-EN	

2. 相關旗標

旗標	說明
SM450	若記憶卡有存在於 CPU module 上，則此旗標會為 ON 的狀態。
SM451	記憶卡在寫保護的模式裡
SM452	記憶卡正在讀/寫的動作
SM453	當記憶卡有發生錯誤的時候，此 SM 會設定為 ON。當此 SM 為 ON 時，需自行設定為 OFF。錯誤代碼請參考 SR453。

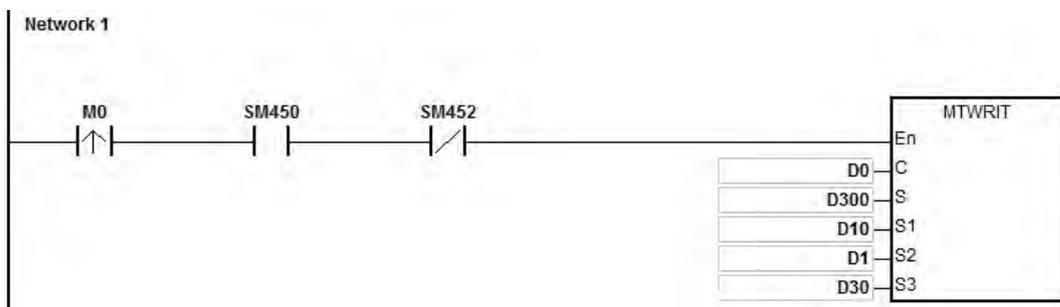
相關錯誤代碼 (SR453)

錯誤代碼	說明
16#005E	當 CPU module 再進行記憶卡初始化流程時，發生錯誤而無法進行初始化流程。
16#005F	路徑錯誤或檔案不存在
16#0060	無法產生預設資料夾

錯誤代碼	說明
16#0061	記憶卡記憶體空間不足
16#0062	記憶卡為防寫保護
16#0063	當資料寫入到檔案時，發生錯誤
16#0064	當要讀取檔案資料時，無法讀取檔案的資料
16#0065	檔案屬性為唯讀狀態

程式範例：

當記憶卡有插入 CPU module 的時候，SM450 會為 ON 的狀態，且接點會閉合。當 MTWRIT 開始運作的時候，SM452 會為 ON 的狀態，且接點會打開，當 MTWRIT 完成寫入動作的時候，SM452 會為 OFF 的狀態，且接點會閉合。



運算元	設定值	說明
D0	16#0001	覆寫檔案
		檔案資料格式：每個字元的單位為 byte。副檔案名稱為.txt。採用 ASCII 的字元。將 D300 的內容寫入到檔案裡。
D300	-	寫入資料內容
D10	16#000A	寫入字串長度為 10 個 bytes。
D1	16#0A00	每次寫完資料後，將此符號加入到最後一個 byte
D30	D30=16#6554 D31=16#7473 D32=16#0031	寫入之檔案名稱為 Test1

補充說明：

1. C 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
2. S₁ 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。
3. S₃ 內容值超出範圍時，視為運算錯誤，指令不執行，SM0=ON，SR0 記錄錯誤碼 16#2003。

6.24 任務控制指令

6.24.1 任務控制指令一覽表

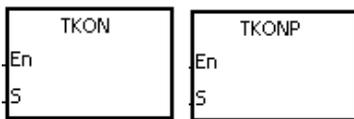
API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>2400</u>	TKON	–	✓	Cyclic 任務 Task 啟動	3
<u>2401</u>	TKOFF	–	✓	Cyclic 任務 Task 關閉	3

6.24.2 任務控制指令說明

API	指令碼				運算元								功能			
2400		TKON	P		S								Cyclic 任務 Task 啟動			
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	“\$”	DF
S	●	●						●	●		●		○			

脈波執行型	16 位元指令 (3 steps)	32 位元指令
AH500	AH500	-

符號：



S : 任務編號

Word

指令說明：

1. 啟動 S 所指定的 Cyclic Task。
2. 當 PLC 由 STOP→RUN 時，會依據 ISPSOft 程式下載時 Cyclic Task 是否啟動來執行。
3. 運算元說明：
 - S 的範圍為 0~31。
 - Tasks 的建立與啟動請參考 ISPSOft 手冊。

程式範例：

6

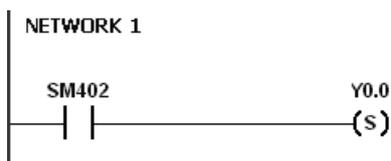
程式寫入主機時，只指定啟動 Task Cyclic_0 任務，PLC RUN 後，因為程式中指示 TKON 1 表示執行 Task Cyclic_1，所以 Y0.0=ON。

在 ISPSOft 中先建立兩個 Cyclic Tasks，只先指定 Cyclic_0 在 PLC RUN 時為 On，Cyclic_1 先預設為不執行。

Cyclic_0 的 Prog0 中 TKON 1 啟動 Cyclic_1



Cyclic_1 的 Prog1 的程式就會被執行



補充說明：

有關任務的說明，請參考 ISPSOft 使用說明。

API	指令碼			運算元								功能						
2401		TKOFF	P	S								Cyclic 任務 Task 關閉						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	"\$"	DF		
S	●	●						●	●		●		○					
										脈波執行型			16 位元指令 (3 steps)			32 位元指令		
										AH500			AH500			-		

符號：



S : 任務編號

Word

指令說明：

1. 關閉 **S** 所指定的 Cyclic Task。
2. 當 PLC 由 STOP→RUN 時，會依據 ISPSOFT 程式下載時 Cyclic Task 是否啟動來執行。
3. 運算元說明：
 - **S** 的範圍為 0~31。
 - Tasks 的建立與啟動請參考 ISPSOFT 手冊。

程式範例：

程式寫入主機時，就指定啟動 Task Cyclic_0 及 Tasks Cyclic_1 任務，PLC RUN 後，因為 TKOFF1 表示關閉 Task Cyclic_1 因此 Y0.0=OFF。

在 ISPSOFT 中先建立兩個 Cyclic Tasks，先指定 Cyclic_0 和 Cyclic_1 在 PLC RUN 時為執行。

Cyclic_0 的 Prog0 中 TKOFF 1 關閉 Cyclic_1



Cyclic_1 的 Prog1 的程式不會執行



補充說明：

有關任務的說明，請參考 ISPSOFT 使用說明。

6.25 SFC 控制指令

6.25.1 SFC 控制指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
<u>2500</u>	SFCRUN	–	–	SFC 啟動	7
<u>2501</u>	SFCPSE	–	–	SFC 暫停	5
<u>2502</u>	SFCSTP	–	–	SFC 停止	3

6.25.2 SFC 控制指令說明

API	指令碼			運算元							功能				
2500	SFCRUN			$S_1 \cdot S_2 \cdot S_3$							SFC 啟動				

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

脈波執行型	16 位元指令 (7 steps)	32 位元指令
-	AH500	-

符號：

SFCRUN	
En	
S1	
S2	
S3	

 S_1 : SFC 名稱 S_2 : 控制參數 Word S_3 : 要啟動的 STEP 名稱

指令說明：

1. 依照 S_2 的設定，啟動 S_1 所指定的 SFC 程式。
2. 指令執行時， S_1 所指定的 SFC 程式並不會立即啟動，要等到掃描到 SFC 程式時才會啟動。
3. 運算元說明：
 - S_1 為 SFC 程式的名稱。
 - $S_2=0$ ，啟動 S_1 指定的 SFC 程式，SFC 中的 SFC/STEP/ACTION/TRANSITION 狀態跟參數會被清除，並從初始 STEP 開始執行。
 - $S_2=1$ ，啟動 S_1 指定的 SFC 程式，SFC 中的 SFC/STEP/ACTION/TRANSITION 狀態跟參數會被清除，並從 S_3 指定的 STEP 開始執行。
 - $S_2=2$ ，啟動 S_1 指定的 SFC 程式，不會清除 SFC 中的 SFC/STEP/ACTION/TRANSITION 狀態跟參數，並從原本暫停的位置開始執行。
 - S_3 為 S_1 指定的 SFC 程式中的 STEP 名稱。
4. S_2 的範圍 0~2，超出範圍時，視為 0。
5. 當 SFC 程式狀態為 RUN 時，執行此指令無效。

程式範例：

在程式中建立一個 LD 語言 POU 名稱為主 Main 及兩個 SFC 語言 POU 名稱為主 TestSFC1 與 TestSFC2。

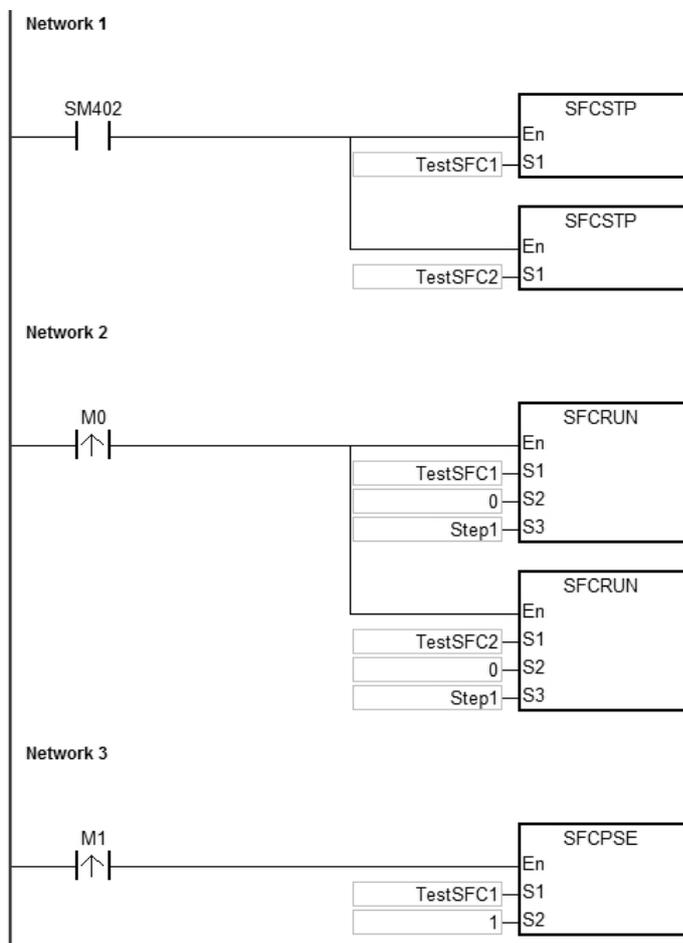
1. 系統執行 (RUN) 時，TestSFC1 與 TestSFC2 立刻執行 SFCSTP 動作，兩個 SFC POU 會停止執行。

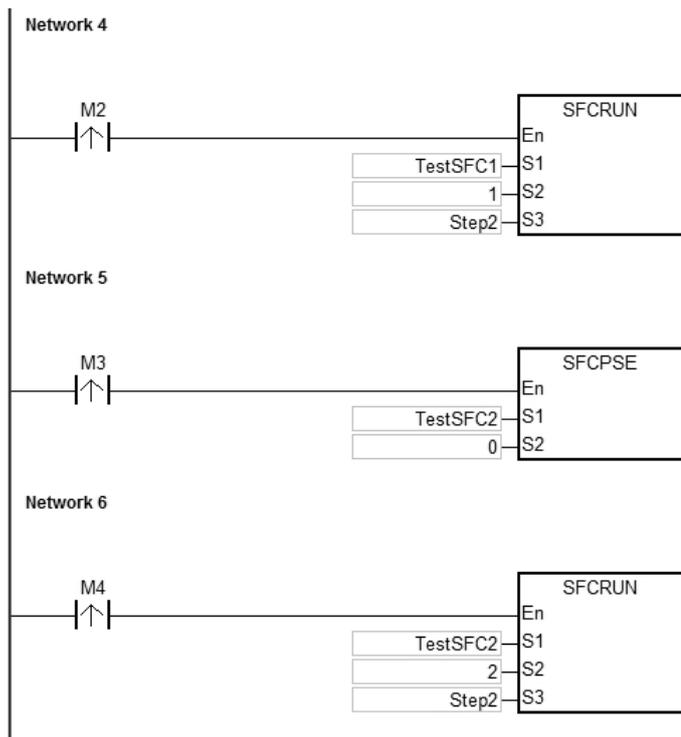
2. 當 M0 由 OFF→ON 時，TestSFC1/ TestSFC2 POU 執行動作 (SFCRUN)^{註一}，執行動作請參考 TestSFC1/ TestSFC2 POU 程式內容。設定 **S₂** 控制參數 0，會清除 SFC 中之狀態及參數，並且從 Step1 開始執行。設定 **S₂** 控制參數 1，會清除 SFC 中之狀態及參數，並且從 **S₃** 指定的 Step 開始執行。
3. 當 M1 由 OFF→ON 時，TestSFC1 POU 會暫停，控制參數設為 1，SFC 中正在執行中的 ACTION，程式中的輸出會被清除 (做 Final scan)。
4. 當 M2 由 OFF→ON 時，TestSFC1 POU 執行動作，執行動作請參考 TestSFC1 POU 程式內容。設定控制參數 1，會清除 SFC 中之狀態及參數，並且從 Step2 開始執行。
5. 當 M3 由 OFF→ON 時，TestSFC2 POU 會暫停，控制參數設為 0，SFC 中之輸出保持不變，不做 Final scan。
6. 當 M4 由 OFF→ON 時，TestSFC2 POU 執行動作，執行動作請參考 TestSFC2 POU 程式內容。設定控制參數 2，會保持 SFC 中之狀態及參數，並且從暫停時執行到之 Step 開始執行。

註一：SFCRUN 要等到下一個掃描時間才會啟動指定的 SFC POU。

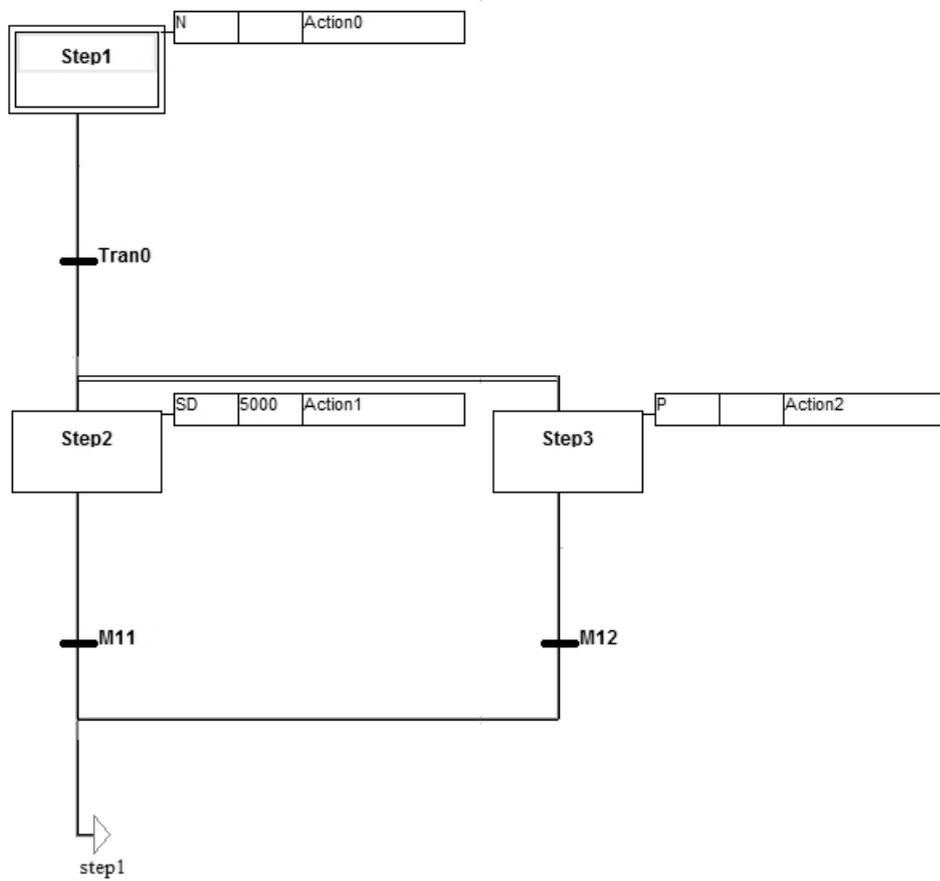
Main POU

6



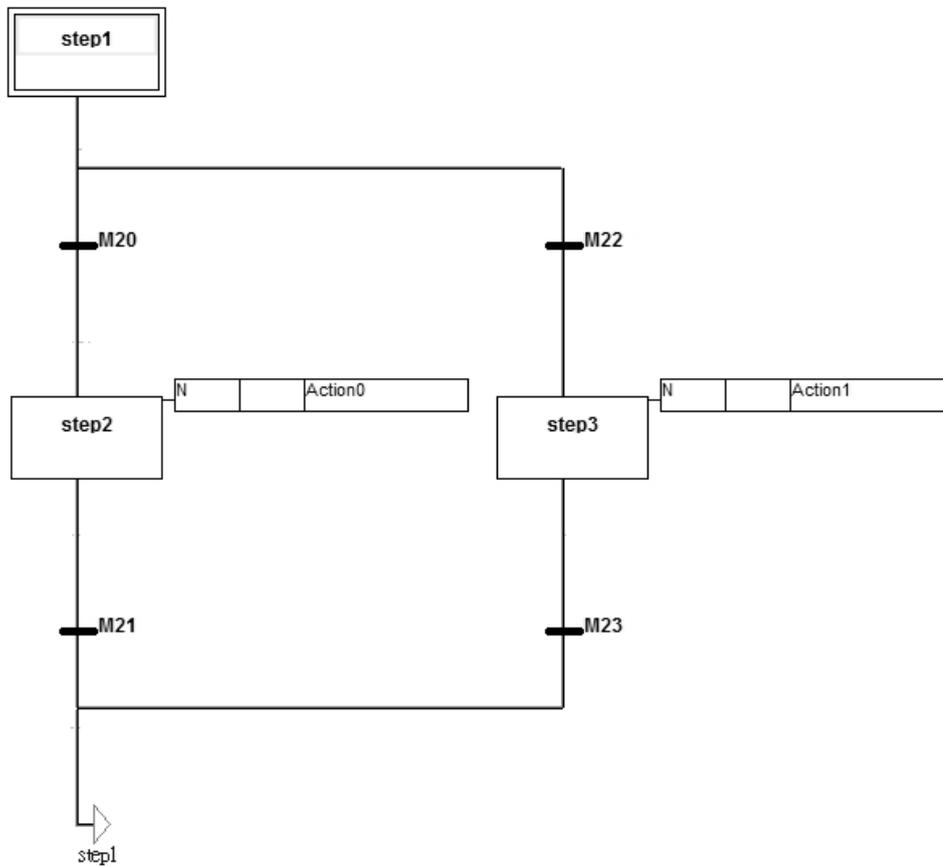


TestSFC1 POU



6

TestSFC2 POU



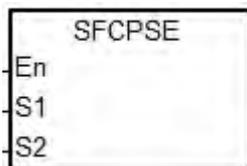
6

補充說明：

有關 SFC 的說明，請參考 ISPSOft 使用說明。

API	指令碼			運算元								功能					
2501	SFCPSE			$S_1 \cdot S_2$								SFC 暫停					
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	K	16#	"\$"	DF	
S_1																	
S_2	●	●						●	●		●		○	○			
										脈波執行型		16 位元指令 (5 steps)			32 位元指令		
										-		AH500			-		

符號：



S_1 : SFC 名稱

S_2 : 控制參數 Word

指令說明：

1. 依照 S_2 的設定，暫停 S_1 所指定的 SFC 程式。
2. 指令執行時， S_1 所指定的 SFC 程式並不會立即暫停，要等到掃描到 SFC 程式時才會暫停。
3. 進入暫停時，SFC 中的 STEP/ACTION/TRANSITION 狀態跟參數不變。
4. 運算元說明：
 - S_1 為 SFC 程式的名稱。
 - $S_2=0$ ，暫停 S_1 指定的 SFC 程式，SFC 程式中的輸出保持不變，正在執行中的 ACTION，不執行 Final Scan。
 - $S_2=1$ ，暫停 S_1 指定的 SFC 程式，正在執行中的 ACTION，程式中的輸出會被清除（執行 Final Scan）。
5. S_2 的範圍 0~1，超出範圍時，視為 0。
6. 當 SFC 程式狀態為 PAUSE/STOP，執行此指令無效。

程式範例：

請參考 SFCRUN 程式範例。

補充說明：

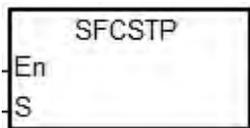
有關 SFC 的說明，請參考 ISPSOFT 使用說明。

API	指令碼			運算元						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																

脈波執行型	16 位元指令 (3 steps)	32 位元指令
-	AH500	-

符號：



S : SFC 名稱

指令說明：

1. 停止 **S** 所指定的 SFC 程式。
2. 指令執行時，**S** 所指定的 SFC 程式並不會立即停止，要等到掃描到 SFC 程式時才會停止。
3. **S** 指定的 SFC 程式，正在執行中的 ACTION，程式中的輸出會被清除（執行 Final Scan）。
4. 當 SFC 程式狀態為 STOP，執行此指令無效。

程式範例：

請參考 SFCRUN 程式範例。

6

補充說明：

有關 SFC 的說明，請參考 ISPSOFT 使用說明。

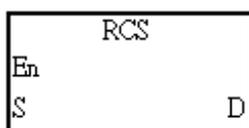
6.26 備援系統控制指令

6.26.1 備援系統控制指令一覽表

API	指令碼 (位元)		P 指令	功能	STEPS
	16	32			
2900	SSO	–	✓	備援系統切換	3
2901	RCS	–	–	讀取/設定備援系統資訊	5

API	指令碼			運算元							功能						
2901		RCS		S · D							讀取/設定備援系統資訊						
裝置	X	Y	M	S	T	C	HC	D	L	SM	SR	E	PR	K	16#	“\$”	DF
S	●	●			●	●		●	●		●		●				
D	●	●			●	●		●	●		●		●				
脈波執行型											16 位元指令 (5 steps)			32 位元指令			
-											AH500			-			

符號：



S : 備援系統資訊類別 Word

D : 備援系統資訊與執行結果 Word

指令說明：

1. 本指令為 AHCPU560-EN2 備援系統主機專用指令。
2. 運算元 S 佔用 2 個連續裝置。S · S+1。
3. 運算元 D 佔用 4 個連續裝置。D · D+1 · D+2 · D+3。
4. S：範圍 0 ~ 5。
5. 在使用本指令之前，需用 ISPSOft -> HWCONFIG 啟動備援模式。
 - PLC 參數設定 -> CPU -> 系統 勾選啟動備援模式。

說明列表：

S	說明	S+1	說明	D	說明	
0	系統 ID	0	讀取當前主機 ID	1	ID 為 A	
				2	ID 為 B	
		1	設定主機 ID 為 A	不使用		
				2	設定主機 ID 為 B	不使用
1	備援系統	不使用				1
					2	備援成功
					3	備援功能未啟動
					4	未通過資格檢定
					5	不存在備援 CPU
2	主控制器 電源模組	不使用			1	電源模組正常
					2	左側電源模組異常
					3	右側電源模組異常

6

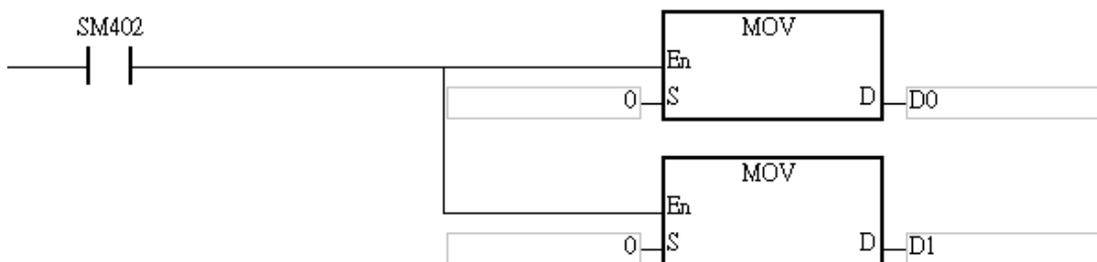
S	說明	S+1	說明	D	說明
3	控制器燈號		不使用	D：主控制器 · D+2：備援控制器	
				Bit0	D：備援控制器不存在 D+2：主控制器不存在
				Bit1	RUN 燈恆亮
				Bit2	RUN 燈熄滅
				Bit3	RUN 燈閃爍
				Bit4	ERROR 燈恆亮
				Bit5	ERROR 燈熄滅
				Bit6	ERROR 燈閃爍
				Bit7	BUS FAULT 燈恆亮
				Bit8	BUS FAULT 燈熄滅
				Bit9	BUS FAULT 燈閃爍
				Bit10	SYSTEM 燈恆亮
				Bit11	SYSTEM 燈熄滅
				Bit12	SYSTEM 燈閃爍
				Bit13~ Bit15	不使用
				D+1：主控制器 · D+3：備援控制器	
				Bit0	MASTER 燈恆亮
				Bit1	MASTER 燈熄滅
				Bit2	MASTER 燈閃爍
				Bit3	SYNC 燈橘恆亮
Bit4	SYNC 燈橘閃爍				
Bit5	SYNC 燈綠恆亮				
Bit6	SYNC 燈綠閃爍				
Bit7	SYNC 燈紅恆亮				
Bit8	SYNC 燈紅閃爍				
Bit9	SYNC 燈熄滅				
Bit10~ Bit15	不使用				
4	備援控制器 電源模組		不使用	0	備援控制器不存在
				1	電源模組正常
				2	左側電源模組異常
				3	右側電源模組異常

S	說明	S+1	說明	D	說明
5	延伸背板 電源模組	1	第一塊延伸背板	0	延伸背板不存在
		2	第二塊延伸背板	1	電源模組正常
		3	第三塊延伸背板	2	左側電源模組異常
		4	第四塊延伸背板		
		5	第五塊延伸背板	3	右側電源模組異常
		6	第六塊延伸背板		
		7	第七塊延伸背板		

程式範例：

此範例為讀取當前主機 ID，當 M0=ON 時，如 D10=1，則此主機 ID 為 A，D10=2 時，此主機 ID 為 B。

Network 1



Network 2



6

MEMO

6



第7章 錯誤代碼

目錄

7.1	模組錯誤代碼對應燈號及狀態說明.....	7-2
7.1.1	CPU 模組錯誤代碼對應燈號及狀態	7-2
7.1.2	類比模組與溫度模組錯誤代碼對應燈號	7-20
7.1.3	AH02HC-5A/AH04HC-5A 錯誤代碼對應燈號	7-21
7.1.4	AH05PM-5A/AH10PM-5A/AH15PM-5A 錯誤代碼對應燈號	7-22
7.1.5	AH20MC-5A 錯誤代碼對應燈號	7-23
7.1.6	AH10EN-5A/AH15EN-5A 錯誤代碼對應燈號	7-24
7.1.7	AH10SCM-5A/AH15SCM-5A 錯誤代碼對應燈號	7-24
7.1.8	AH10DNET-5A 錯誤代碼對應燈號.....	7-25
7.1.9	AH10PFBM-5A 錯誤代碼對應燈號.....	7-26
7.1.10	AH10PFBS-5A 錯誤代碼對應燈號	7-26
7.1.11	AH10COPM-5A 錯誤代碼對應燈號	7-27

7.1 模組錯誤代碼對應燈號及狀態說明

A. 欄位簡介

- a. 錯誤代碼：該錯誤發生時系統所產生之錯誤代碼。
- b. 說明：該錯誤之說明。
- c. CPU 狀態：該錯誤發生時，CPU 主機的狀態變化。
 - 停止：發生該錯誤時 CPU 停止運行。
 - 持續：發生該錯誤時 CPU 持續運行。
 - 自訂：CPU 狀態的變化可讓使用者自行定義，請參考 AH500 操作手冊第 8.2.1 節。
- d. 燈號狀態：該錯誤發生時的主機燈號變化。
 - ERROR：系統錯誤燈號。
 - BUS FAULT：I/O 匯流排錯誤燈號。
 - MODULE ERROR：模組錯誤燈號。

● 各燈號說明

	燈號	說明
CPU	Error LED	指示 CPU 的錯誤狀態。 恆亮：系統嚴重錯誤發生。 熄滅：系統正常。 閃爍：系統非嚴重錯誤發生。
	Bus Fault LED	指示 I/O Bus 的錯誤狀態。 恆亮：I/O Bus 嚴重錯誤發生。 熄滅：I/O Bus 正常。 閃爍：I/O Bus 非嚴重錯誤發生。
MODULE	ERROR	指示 MODULE 的錯誤狀態。 恆亮：MODULE 嚴重錯誤發生。 熄滅：MODULE 正常。 閃爍：MODULE 非嚴重錯誤發生。

7.1.1 CPU 模組錯誤代碼對應燈號及狀態

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#000A	掃描逾時 (錯誤旗標 SM8)	停止	閃爍	保持
16#000B	PLC 程式毀損	停止	恆亮	保持
16#000C	下載 PLC 程式校驗錯誤	停止	閃爍	保持
16#000D	CPU 參數毀損	停止	恆亮	保持
16#000E	程式或參數下載中，PLC 無法切換至 RUN	停止	閃爍	保持
16#000F	PLC 原始程式毀損	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#0010	CPU 記憶體存取被拒	停止	恆亮	保持
16#0011	PLC ID 錯誤 (錯誤旗標 SM9)	持續	恆亮	保持
16#0012	PLC 密碼錯誤 (錯誤旗標 SM9)	持續	恆亮	保持
16#0013	I/O 模組無法設置運行/停止 (錯誤旗標 SM10)	停止	保持	恆亮
16#0014	無法執行系統還原程序 (錯誤旗標 SM9)	停止	恆亮	恆亮
16#0015	模組配置資料錯誤 (錯誤旗標 SM10)	停止	恆亮	保持
16#0016	IO 模組設定資料錯誤 (錯誤旗標 SM10)	停止	恆亮	保持
16#0017	D 對應裝置設定錯誤 (錯誤旗標 SM10)	停止	恆亮	保持
16#0018	序列埠異常 (錯誤旗標 SM9)	持續	閃爍	保持
16#0019	USB 異常 (錯誤旗標 SM9)	持續	閃爍	保持
16#001A	系統備份檔案 (DUP) 內容錯誤	持續	閃爍	保持
16#001B	定時中斷 (編號 0) 時間設置錯誤	停止	恆亮	保持
16#001C	定時中斷 (編號 1) 時間設置錯誤	停止	恆亮	保持
16#001D	定時中斷 (編號 2) 時間設置錯誤	停止	恆亮	保持
16#001E	定時中斷 (編號 3) 時間設置錯誤	停止	恆亮	保持
16#001F	程式掃描逾時定時器設置錯誤	停止	恆亮	保持
16#0020	固定掃描時間設置錯誤	停止	恆亮	保持
16#0021	固定掃描時間設置錯誤	停止	恆亮	保持
16#0022	下載 CPU 模組參數校驗錯誤	停止	恆亮	保持
16#0023	系統 PLC 參數設定內 · Y 裝置狀態 ((STOP -> RUN) 設定選項錯誤	停止	恆亮	保持
16#0024	背板無 IO 模組	持續	保持	保持
16#0025	符號初始值與程式不符合	停止	恆亮	保持
16#0026	通訊能力佔用掃描時間比率設定錯誤	停止	恆亮	保持
16#0027	M 裝置停電保持區範圍設定錯誤	停止	恆亮	保持
16#0028	D 裝置停電保持區範圍設定錯誤	停止	恆亮	保持
16#0029	T 裝置停電保持區範圍設定錯誤	停止	恆亮	保持
16#002A	C 裝置停電保持區範圍設定錯誤	停止	恆亮	保持
16#002B	HC 裝置停電保持區範圍設定錯誤	停止	恆亮	保持
16#0033	COM 1 通訊設定設置錯誤 (錯誤旗標 SM9)	持續	閃爍	保持
16#0034	COM 1 站號設置錯誤 (錯誤旗標 SM9)	持續	閃爍	保持
16#0035	COM 1 傳輸方式設置錯誤 (錯誤旗標 SM9)	持續	閃爍	保持
16#0038	COM 2 通訊設定設置錯誤 (錯誤旗標 SM9)	持續	閃爍	保持
16#0039	COM 2 站號設置錯誤 (錯誤旗標 SM9)	持續	閃爍	保持
16#003A	COM 2 傳輸方式設置錯誤 (錯誤旗標 SM9)	持續	閃爍	保持
16#0050	停電保持區 SM 記憶區塊異常	持續	恆亮	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#0051	停電保持區 SR 暫存器異常	持續	恆亮	保持
16#0052	停電保持區 M 記憶區塊異常	持續	恆亮	保持
16#0053	停電保持區 T 暫存器異常	持續	恆亮	保持
16#0054	停電保持區 C 暫存器異常	持續	恆亮	保持
16#0055	停電保持區 HC 暫存器異常	持續	恆亮	保持
16#0056	停電保持區 T 記憶區塊異常	持續	恆亮	保持
16#0057	停電保持區 C 記憶區塊異常	持續	恆亮	保持
16#0058	停電保持區 HC 記憶區塊異常	持續	恆亮	保持
16#0059	停電保持區 D 暫存器異常	持續	恆亮	保持
16#005A	停電保持區 W 暫存器異常	持續	恆亮	保持
16#005D	CPU 模組偵測不到記憶卡 (錯誤旗標 SM453)	持續	保持	保持
16#005E	記憶卡的初始程序錯誤 (錯誤旗標 SM453)	持續	保持	保持
16#005F	於記憶卡中，欲讀取不存在的檔案，或寫入不存在路徑的檔案。(錯誤旗標 SM453)	持續	保持	保持
16#0060	CPU 模組無法建立預設資料夾 (錯誤旗標 SM453)	持續	保持	保持
16#0061	記憶卡容量不足 (錯誤旗標 SM453)	持續	保持	保持
16#0062	記憶卡為防寫模式 (錯誤旗標 SM453)	持續	保持	保持
16#0063	資料寫入記憶卡的檔案時有錯誤 (錯誤旗標 SM453)	持續	保持	保持
16#0064	記憶卡的檔案無法被讀取 (錯誤旗標 SM453)	持續	保持	保持
16#0065	記憶卡中的檔案為唯讀狀態 (錯誤旗標 SM453)	持續	保持	保持
16#0066	系統備份時錯誤	持續	閃爍	保持
16#0067	系統還原的系統參數長度超出 CPU 模組的系統參數長度	持續	閃爍	保持
16#0068	符號初始值表格毀損	停止	閃爍	保持
16#1400	輔助處理器存取錯誤	停止	保持	恆亮
16#1401	模組存取錯誤 (錯誤旗標 SM9)	停止	保持	恆亮
16#1402	實際模組不符合配置設定 (錯誤旗標 SM9)	停止	保持	恆亮
16#1403	從模組讀取資料錯誤 (錯誤旗標 SM9)	停止	保持	恆亮
16#1405	搜尋不到 I/O 模組的設定參數 (錯誤旗標 SM9)	停止	保持	恆亮
16#1407	輔助處理器通訊錯誤 (錯誤旗標 SM9)	持續	恆亮	保持
16#1409	延伸背板連線中斷 (錯誤旗標 SM9)	停止	保持	恆亮
16#140A	延伸背板通訊錯誤 (錯誤旗標 SM9)	停止	保持	恆亮
16#140B	通訊模組數量超過上限 (錯誤旗標 SM9)	停止	保持	恆亮
16#140C	高速資料交換校驗錯誤 (錯誤旗標 SM9)	停止	保持	恆亮
16#140D	實際電源 ID 不符合配置設定 (錯誤旗標 SM9)	停止	保持	恆亮
16#140E	模組高速資料交換數量超出支援最大範圍 (錯誤旗標 SM10)	停止	保持	恆亮

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#140F	高速資料交換錯誤 (錯誤旗標 SM11)	停止	保持	恆亮
16#1410	RTU IO 模組發生錯誤	停止	保持	恆亮
16#1411	RTU IO 模組發出警告	持續	保持	恆亮
16#1420	模組之 Ethernet port 發生 Link off 狀態	持續	保持	保持
16#1421	主機讀取模組之智慧型模組設定資訊發生錯誤	停止	保持	恆亮
16#1422	主機寫入模組之智慧型模組設定資訊發生錯誤	停止	保持	恆亮
16#1801	CPU 模組未設定中斷工作	持續	保持	保持
16#2000	PLC 程式無 END 指令 (錯誤旗標 SM5)	停止	閃爍	保持
16#2001	專案程式內容有誤，程式語法錯誤 (錯誤旗標 SM5)	停止	閃爍	保持
16#2002	GOEND 使用的地方錯誤 (錯誤旗標 SM5)	停止	閃爍	保持
16#2003	程式中使用的裝置超過可用範圍 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#2004	CJ/JMP 指令跳躍的 P 位址錯誤，或是 P 裝置重覆使用。 (錯誤旗標 SM0/SM5)	停止	閃爍	保持
16#2005	MC/MCR 相對應的 N 值不同，或數量不一樣多 (錯誤旗標 SM5)	停止	閃爍	保持
16#2006	n 不是從 0 開始或是 n 的值不連續 (錯誤旗標 SM5)	停止	閃爍	保持
16#2007	ZRST 指令運算元使用不當 (錯誤旗標 SM5)	停止	閃爍	保持
16#200A	無效的指令 (錯誤旗標 SM5)	停止	閃爍	保持
16#200B	n 運算元或其它 K/H 運算元超出範圍 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#200C	部份指令不允許運算元發生重疊 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#200D	BIN 轉成 BCD 時發生錯誤 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#200E	字串沒有 0x00 當做結尾 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#200F	指令不支援 E 裝置修飾 (錯誤旗標 SM5)	停止	閃爍	保持
16#2010	指令不支援該裝置類別/編碼錯誤/16 位元指令但 K·H 卻是 32 位元的編碼 (錯誤旗標 SM5)	停止	閃爍	保持
16#2011	運算元的數目錯誤 (錯誤旗標 SM5)	停止	閃爍	保持
16#2012	除法運算錯誤 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#2013	浮點數格式錯誤，超出可轉換範圍(錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#2014	TKON/TKOFF 指令所指定的 TASK 編號錯誤或超出範圍 (錯誤旗標 SM5)	停止	閃爍	保持
16#2015	CALL 指令超過 32 層 (錯誤旗標 SM0)	自訂	閃爍	保持
16#2016	FOR-NEXT 指令超過 32 層 (錯誤旗標 SM0/SM5)	自訂	閃爍	保持
16#2017	FOR 跟 NEXT 的指令數目不同 (錯誤旗標 SM5)	停止	閃爍	保持
16#2018	在 FEND 之後的 P 指標沒有相對應的 SRET，或是有 SRET 但沒有 P 指標 (錯誤旗標 SM5)	停止	閃爍	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#2019	Interrupt I 的位址不是在 FEND 之後 (錯誤旗標 SM5)	停止	閃爍	保持
16#201A	IRET/SRET 的位址不是在 FEND 之後 (錯誤旗標 SM5)	停止	閃爍	保持
16#201B	I 沒有相對應的 IRET, 或是有 IRET 但沒有 I (錯誤旗標 SM5)	停止	閃爍	保持
16#201C	END 指令不是在程式的最後一個位址 (錯誤旗標 SM5)	停止	閃爍	保持
16#201D	有 CALL 指令但沒有 MAR 指令 (錯誤旗標 SM5)	停止	閃爍	保持
16#201E	MODRW 指令中的功能代碼錯誤 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#201F	MODRW 指令中的資料長度錯誤 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2020	MODRW 的回覆命令錯誤 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2021	MODRW 回覆命令校驗碼錯誤 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2022	MODRW 指令的命令不符合 ASCII 格式 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2023	MODRW 指令的通訊逾時 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2024	RS 指令的通訊逾時數值無效 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2025	RS 指令的通訊逾時 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2026	RS 指令的中斷指標異常 (錯誤旗標 SM102/104)	自訂	閃爍	保持
16#2027	FWD 應用指令異常 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2028	REV 應用指令異常 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#2029	STOP 應用指令異常 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#202A	RSDT 應用指令異常 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#202B	RSTEF 應用指令異常 (錯誤旗標 SM102/103)	自訂	閃爍	保持
16#202C	IO 中斷服務程式 0 不存在	停止	閃爍	保持
16#202D	IO 中斷服務程式 1 不存在	停止	閃爍	保持
16#202E	IO 中斷服務程式 2 不存在	停止	閃爍	保持
16#202F	IO 中斷服務程式 3 不存在	停止	閃爍	保持
16#2030	IO 中斷服務程式 4 不存在	停止	閃爍	保持
16#2031	IO 中斷服務程式 5 不存在	停止	閃爍	保持
16#2032	IO 中斷服務程式 6 不存在	停止	閃爍	保持
16#2033	IO 中斷服務程式 7 不存在	停止	閃爍	保持
16#2034	IO 中斷服務程式 8 不存在	停止	閃爍	保持
16#2035	IO 中斷服務程式 9 不存在	停止	閃爍	保持
16#2036	IO 中斷服務程式 10 不存在	停止	閃爍	保持
16#2037	IO 中斷服務程式 11 不存在	停止	閃爍	保持
16#2038	IO 中斷服務程式 12 不存在	停止	閃爍	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#2039	IO 中斷服務程式 13 不存在	停止	閃爍	保持
16#203A	IO 中斷服務程式 14 不存在	停止	閃爍	保持
16#203B	IO 中斷服務程式 15 不存在	停止	閃爍	保持
16#203C	IO 中斷服務程式 16 不存在	停止	閃爍	保持
16#203D	IO 中斷服務程式 17 不存在	停止	閃爍	保持
16#203E	IO 中斷服務程式 18 不存在	停止	閃爍	保持
16#203F	IO 中斷服務程式 19 不存在	停止	閃爍	保持
16#2040	IO 中斷服務程式 20 不存在	停止	閃爍	保持
16#2041	IO 中斷服務程式 21 不存在	停止	閃爍	保持
16#2042	IO 中斷服務程式 22 不存在	停止	閃爍	保持
16#2043	IO 中斷服務程式 23 不存在	停止	閃爍	保持
16#2044	IO 中斷服務程式 24 不存在	停止	閃爍	保持
16#2045	IO 中斷服務程式 25 不存在	停止	閃爍	保持
16#2046	IO 中斷服務程式 26 不存在	停止	閃爍	保持
16#2047	IO 中斷服務程式 27 不存在	停止	閃爍	保持
16#2048	IO 中斷服務程式 28 不存在	停止	閃爍	保持
16#2049	IO 中斷服務程式 29 不存在	停止	閃爍	保持
16#204A	IO 中斷服務程式 30 不存在	停止	閃爍	保持
16#204B	IO 中斷服務程式 31 不存在	停止	閃爍	保持
16#2054 16#2127	外部中斷服務程式 40 不存在 外部中斷服務程式 251 不存在	停止	閃爍	保持
16#2128	SFC Action 時間屬性設定錯誤 (錯誤旗標 SM0/SM1)	自訂	閃爍	保持
16#2129	SFC Action 重置屬性設置錯誤 (錯誤旗標 SM0/SM1)	自訂	閃爍	保持
16#212A	MC/MCR 指令不能在中斷或副程式使用 (錯誤旗標 SM5)	自訂	閃爍	保持
16#6000	乙太網偵測速率失敗 (錯誤旗標 SM1106)	持續	閃爍	保持
16#6001	IP 位址不合法 (錯誤旗標 SM1107)	持續	閃爍	保持
16#6002	網路遮罩位址不合法 (錯誤旗標 SM1107)	持續	閃爍	保持
16#6003	閘道位址不合法 (錯誤旗標 SM1107)	持續	閃爍	保持
16#6004	乙太網路的 IP 位址過濾設置錯誤 (旗標 SM1108)	持續	閃爍	保持
16#6006	乙太網路的靜態 ARP 表設置錯誤 (旗標 SM1108)	持續	閃爍	保持
16#6007	NTP 設置錯誤 (錯誤旗標 SM1380)	持續	閃爍	保持
16#6008	網路編號不合法 (錯誤旗標 SM1107)	持續	閃爍	保持
16#6009	節點編號不合法 (錯誤旗標 SM1107)	持續	閃爍	保持
16#600A	TCP 連線建立失敗 (錯誤旗標 SM1090)	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#600B	UDP 連線建立失敗 (錯誤旗標 SM1091)	持續	保持	保持
16#600C	Socket 通訊接口已被使用 (錯誤旗標 SM1109)	持續	保持	保持
16#600D	RJ45 埠未連接 (錯誤旗標 SM1100)	持續	保持	保持
16#600E	AH10EN 上 RJ45 埠未連接網路線	持續	保持	保持
16#600F	MODBUS TCP 伺服器連線已滿 (錯誤旗標 SM1089)	持續	閃爍	保持
16#6010	BOOTP IP 設定錯誤 (錯誤旗標 SM1107)	持續	保持	保持
16#6011	BOOTP Gateway 設定錯誤 (錯誤旗標 SM1107)	持續	保持	保持
16#6012	IP 位址重複錯誤 (錯誤旗標 SM1107)	持續	閃爍	保持
16#6013	DNS 位址設定錯誤 (錯誤旗標 SM1107)	持續	保持	保持
16#6100	E-mail 連線忙碌 (錯誤旗標 SM1113)	持續	保持	保持
16#6101	E-mail 發送條件的觸發設定錯誤 (旗標 SM1112)	持續	閃爍	保持
16#6102	E-mail 發送條件的發送時間間隔設定錯誤 (錯誤旗標 SM1112)	持續	閃爍	保持
16#6103	E-mail 附件中的裝置位址設定錯誤 (旗標 SM1112)	持續	閃爍	保持
16#6104	E-mail 附件不存在 (錯誤旗標 SM1113)	持續	保持	保持
16#6105	E-mail 附件超過容量 (錯誤旗標 SM1113)	持續	保持	保持
16#6106	SMTP 伺服器位址錯誤 (錯誤旗標 SM1112)	持續	閃爍	保持
16#6107	SMTP 伺服器逾時 (錯誤旗標 SM1113)	持續	保持	保持
16#6108	寄件伺服器驗證錯誤 (錯誤旗標 SM1112)	持續	閃爍	保持
16#6110	SMTP 伺服器需要進行驗證 (錯誤旗標 SM1112)	持續	閃爍	保持
16#6111	指定的 E-mail 位址不存在 (錯誤旗標 SM1112)	持續	閃爍	保持
16#6200	TCP 通訊接口 (Socket) 功能的遠端 IP 位址不合法 (錯誤旗標 SM1196)	持續	閃爍	保持
16#6201	TCP 通訊接口 (Socket) 功能的本地埠不合法	持續	保持	保持
16#6202	TCP 通訊接口 (Socket) 功能的遠端埠不合法	持續	保持	保持
16#6203	TCP 通訊接口 (Socket) 功能的傳送資料位址不合法	持續	保持	保持
16#6204	TCP 通訊接口 (Socket) 功能的傳送資料長度不合法	持續	保持	保持
16#6205	TCP 通訊接口 (Socket) 功能的傳送資料裝置超出範圍	持續	保持	保持
16#6206	TCP 通訊接口 (Socket) 功能的接收資料位址不合法	持續	保持	保持
16#6207	TCP 通訊接口 (Socket) 功能的接收資料長度不合法	持續	保持	保持
16#6208	TCP 通訊接口 (Socket) 功能的接收資料裝置超出範圍	持續	保持	保持
16#6209	UDP 通訊接口 (Socket) 功能的遠端 IP 位址不合法 (錯誤旗標 SM1196)	持續	閃爍	保持
16#620A	UDP 通訊接口 (Socket) 功能的本地埠不合法	持續	保持	保持
16#620B	UDP 通訊接口 (Socket) 功能的遠端埠不合法	持續	保持	保持
16#620C	UDP 通訊接口 (Socket) 功能的傳送資料位址不合法	持續	保持	保持

7

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#620D	UDP 通訊接口 (Socket) 功能的傳送資料長度不合法	持續	保持	保持
16#620E	UDP 通訊接口 (Socket) 功能的傳送資料裝置超出範圍	持續	保持	保持
16#620F	UDP 通訊接口 (Socket) 功能的接收資料位址不合法	持續	保持	保持
16#6210	UDP 通訊接口 (Socket) 功能的接收資料長度不合法	持續	保持	保持
16#6211	UDP 通訊接口 (Socket) 功能的接收資料裝置超出範圍	持續	保持	保持
16#6212	遠端裝置回應逾時 (Socket)	持續	保持	保持
16#6213	接收資料超過限制 (Socket)	持續	保持	保持
16#6214	遠端裝置拒絕連線 (Socket)	持續	保持	保持
16#6215	目前通訊接口 (Socket) 未開啟	持續	保持	保持
16#6217	目前通訊接口 (Socket) 已開啟	持續	保持	保持
16#6218	目前通訊接口 (Socket) 傳送中	持續	保持	保持
16#6219	目前通訊接口 (Socket) 接收中	持續	保持	保持
16#621A	目前通訊接口 (Socket) 關閉中	持續	保持	保持
16#6300	Ether Link 只可用於裝置 M、D、L	持續	閃爍	保持
16#6301	Ether Link 裝置位址設定超過可用的裝置範圍	持續	閃爍	保持
16#6302	Ether Link 的資料長度超過限制	持續	閃爍	保持
16#6303	Ether Link 的遠端裝置中止連線	持續	保持	保持
16#6304	Ether Link 連線忙碌	持續	保持	保持
16#6305	Ether Link 通訊命令中的節點與本地節點不同	持續	閃爍	保持
16#6309	Ether Link 回應逾時	持續	保持	保持
16#630A	模組的 ID 或設定與 Ether Link 中的設定不同	持續	閃爍	保持
16#630B	CPU 或模組的網路遮罩設定與 Ether Link 設定不同	持續	閃爍	保持
16#6400	連線數超出限制或未設定送信旗標 (EMDRW)	持續	保持	保持
16#6401	遠端裝置中止連線 (EMDRW)	持續	保持	保持
16#6402	遠端裝置回應逾時 (EMDRW)	持續	保持	保持
16#6403	遠端 IP 位址不合法 (EMDRW)	持續	保持	保持
16#6404	不支援的功能代碼 (EMDRW)	持續	保持	保持
16#6405	MODBUS 回覆訊息的 Byte Count 與實際的資料長度不符 (EMDRW)	持續	保持	保持
16#6500	初始化資料交換功能時錯誤 (錯誤旗標 SM699)	持續	閃爍	OFF
16#6501	遠端裝置回應逾時 (錯誤旗標 SM828-SM955)	持續	OFF	OFF
16#6502	遠端裝置回覆封包錯誤 (錯誤旗標 SM828-SM955)	持續	OFF	OFF
16#6700	MODBUS TCP 資料交換初始化錯誤	持續	保持	保持
16#6701	MODBUS TCP 資料交換逾時	持續	保持	保持
16#6702	MODBUS TCP 資料交換接收錯誤	持續	保持	保持
16#7002	CPU 模組不支援此功能	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#7203	無效的存取碼 (Access Code)	持續	保持	保持
16#7401	功能碼 (Function Code) 錯誤	持續	保持	保持
16#7402	封包超出最大資料長度	持續	保持	保持
16#7404	封包格式錯誤	持續	保持	保持
16#7405	位元組長度 (Byte Length) 的資料錯誤	持續	保持	保持
16#7406	校驗 (Checksum) 錯誤	持續	保持	保持
16#7407	命令中包含非 ASCII 字元	持續	保持	保持
16#7408	PLC 處於運行 (RUN) 模式	持續	保持	保持
16#740A	主機記憶體正在寫入或寫入失敗	持續	保持	保持
16#740B	清除或重置動作正在進行中	持續	保持	保持
16#740C	通訊命令中的背板編號不正確	持續	保持	保持
16#740D	通訊命令中的插槽編號不正確	持續	保持	保持
16#740E	清除記憶體的過程發生錯誤	持續	保持	保持
16#740F	通訊逾時	持續	保持	保持
16#7410	回覆命令的功能碼 (Function Code) 不一致	持續	保持	保持
16#7412	因 SW1 ON 所以資料無法下載至 CPU 模組	持續	保持	保持
16#757D	輸入 PLC 密碼的剩餘次數為 0	持續	保持	保持
16#757E	輸入的 PLC 密碼錯誤	持續	保持	保持
16#8105	下載的專案程式內容有誤：下載的程語法錯誤	持續	保持	保持
16#8106	下載的專案程式內容有誤：執行碼超過限制長度	持續	保持	保持
16#8107	下載的專案程式內容有誤：原始碼超過限制長度	持續	保持	保持
16#8230	下載的主機參數有誤：IP 位址不合法	持續	保持	保持
16#8231	下載的主機參數有誤：網路遮罩位址不合法	持續	保持	保持
16#8232	下載的主機參數有誤：閘道位址不合法	持續	保持	保持
16#8233	下載的主機參數有誤：IP 位址過濾設定錯誤	持續	保持	保持
16#8235	下載的主機參數有誤：靜態 ARP 表錯誤	持續	保持	保持
16#8236	下載的主機參數有誤：NTP 設定錯誤	持續	保持	保持
16#8239	下載的主機參數有誤：Email 設定錯誤	持續	保持	保持
16#823A	下載的主機參數有誤：Email 觸發設定錯誤	持續	保持	保持
16#823B	下載的主機參數有誤：TCP 通訊接口 (Socket) 設定錯誤	持續	保持	保持
16#823C	下載的主機參數有誤：UDP 通訊接口 (Socket) 設定錯誤	持續	保持	保持
16#823E	下載的主機參數有誤：Web 設定錯誤	持續	保持	保持

7

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#8240	下載的主機參數有誤：Ether Link	持續	保持	保持
16#8241	DNS 設定錯誤	持續	保持	保持
16#8522	自動掃描偵測執行中	持續	保持	保持
16#853B	IO 模組未配置	持續	保持	保持
16#853C	IO 模組不存在	持續	保持	保持
16#854B	IO 模組未配置	持續	保持	保持
16#854C	IO 模組不存在	持續	保持	保持
16#8572	模組配置表檢查碼錯誤	持續	保持	保持
16#8576	模組參數設定檢查碼錯誤	持續	保持	保持
16#857A	模組參數映射表檢查碼錯誤	持續	保持	保持
16#85E1	IO 中斷編號不正確	持續	保持	保持
16#85E2	IO 中斷服務程式不存在	持續	保持	保持
16#860F	系統還原錯誤	持續	閃爍	閃爍
16#8611	記憶卡不存在或記憶卡格式錯誤	持續	保持	保持
16#8612	記憶卡存取錯誤或記憶卡是唯讀模式	持續	保持	保持
16#9A01	PLC Link / COM1 MODBUS 從站 1 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A02	PLC Link / COM1 MODBUS 從站 2 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A03	PLC Link / COM1 MODBUS 從站 3 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A04	PLC Link / COM1 MODBUS 從站 4 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A05	PLC Link / COM1 MODBUS 從站 5 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A06	PLC Link / COM1 MODBUS 從站 6 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A07	PLC Link / COM1 MODBUS 從站 7 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A08	PLC Link / COM1 MODBUS 從站 8 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A09	PLC Link / COM1 MODBUS 從站 9 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A0A	PLC Link / COM1 MODBUS 從站 10 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#9A0B	PLC Link / COM1 MODBUS 從站 11 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A0C	PLC Link / COM1 MODBUS 從站 12 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A0D	PLC Link / COM1 MODBUS 從站 13 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A0E	PLC Link / COM1 MODBUS 從站 14 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A0F	PLC Link / COM1 MODBUS 從站 15 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A10	PLC Link / COM1 MODBUS 從站 16 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A11	PLC Link / COM1 MODBUS 從站 17 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A12	PLC Link / COM1 MODBUS 從站 18 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A13	PLC Link / COM1 MODBUS 從站 19 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A14	PLC Link / COM1 MODBUS 從站 20 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A15	PLC Link / COM1 MODBUS 從站 21 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A16	PLC Link / COM1 MODBUS 從站 22 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A17	PLC Link / COM1 MODBUS 從站 23 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A18	PLC Link / COM1 MODBUS 從站 24 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A19	PLC Link / COM1 MODBUS 從站 25 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A1A	PLC Link / COM1 MODBUS 從站 26 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A1B	PLC Link / COM1 MODBUS 從站 27 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A1C	PLC Link / COM1 MODBUS 從站 28 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持

7

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#9A1D	PLC Link / COM1 MODBUS 從站 29 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A1E	PLC Link / COM1 MODBUS 從站 30 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A1F	PLC Link / COM1 MODBUS 從站 31 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A20	PLC Link / COM1 MODBUS 從站 32 的資料交換設定錯誤 (錯誤旗標 SM1590)	持續	保持	保持
16#9A21	PLC Link / COM1 MODBUS 從站 1 通訊錯誤	持續	保持	保持
16#9A22	PLC Link / COM1 MODBUS 從站 2 通訊錯誤	持續	保持	保持
16#9A23	PLC Link / COM1 MODBUS 從站 3 通訊錯誤	持續	保持	保持
16#9A24	PLC Link / COM1 MODBUS 從站 4 通訊錯誤	持續	保持	保持
16#9A25	PLC Link / COM1 MODBUS 從站 5 通訊錯誤	持續	保持	保持
16#9A26	PLC Link / COM1 MODBUS 從站 6 通訊錯誤	持續	保持	保持
16#9A27	PLC Link / COM1 MODBUS 從站 7 通訊錯誤	持續	保持	保持
16#9A28	PLC Link / COM1 MODBUS 從站 8 通訊錯誤	持續	保持	保持
16#9A29	PLC Link / COM1 MODBUS 從站 9 通訊錯誤	持續	保持	保持
16#9A2A	PLC Link / COM1 MODBUS 從站 10 通訊錯誤	持續	保持	保持
16#9A2B	PLC Link / COM1 MODBUS 從站 11 通訊錯誤	持續	保持	保持
16#9A2C	PLC Link / COM1 MODBUS 從站 12 通訊錯誤	持續	保持	保持
16#9A2D	PLC Link / COM1 MODBUS 從站 13 通訊錯誤	持續	保持	保持
16#9A2E	PLC Link / COM1 MODBUS 從站 14 通訊錯誤	持續	保持	保持
16#9A2F	PLC Link / COM1 MODBUS 從站 15 通訊錯誤	持續	保持	保持
16#9A30	PLC Link / COM1 MODBUS 從站 16 通訊錯誤	持續	保持	保持
16#9A31	PLC Link / COM1 MODBUS 從站 17 通訊錯誤	持續	保持	保持
16#9A32	PLC Link / COM1 MODBUS 從站 18 通訊錯誤	持續	保持	保持
16#9A33	PLC Link / COM1 MODBUS 從站 19 通訊錯誤	持續	保持	保持
16#9A34	PLC Link / COM1 MODBUS 從站 20 通訊錯誤	持續	保持	保持
16#9A35	PLC Link / COM1 MODBUS 從站 21 通訊錯誤	持續	保持	保持
16#9A36	PLC Link / COM1 MODBUS 從站 22 通訊錯誤	持續	保持	保持
16#9A37	PLC Link / COM1 MODBUS 從站 23 通訊錯誤	持續	保持	保持
16#9A38	PLC Link / COM1 MODBUS 從站 24 通訊錯誤	持續	保持	保持
16#9A39	PLC Link / COM1 MODBUS 從站 25 通訊錯誤	持續	保持	保持
16#9A3A	PLC Link / COM1 MODBUS 從站 26 通訊錯誤	持續	保持	保持
16#9A3B	PLC Link / COM1 MODBUS 從站 27 通訊錯誤	持續	保持	保持
16#9A3C	PLC Link / COM1 MODBUS 從站 28 通訊錯誤	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#9A3D	PLC Link / COM1 MODBUS 從站 29 通訊錯誤	持續	保持	保持
16#9A3E	PLC Link / COM1 MODBUS 從站 30 通訊錯誤	持續	保持	保持
16#9A3F	PLC Link / COM1 MODBUS 從站 31 通訊錯誤	持續	保持	保持
16#9A40	PLC Link / COM1 MODBUS 從站 32 通訊錯誤	持續	保持	保持
16#9A41	PLC Link / COM1 MODBUS 從站 1 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A42	PLC Link / COM1 MODBUS 從站 2 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A43	PLC Link / COM1 MODBUS 從站 3 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A44	PLC Link / COM1 MODBUS 從站 4 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A45	PLC Link / COM1 MODBUS 從站 5 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A46	PLC Link / COM1 MODBUS 從站 6 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A47	PLC Link / COM1 MODBUS 從站 7 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A48	PLC Link / COM1 MODBUS 從站 8 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A49	PLC Link / COM1 MODBUS 從站 9 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A4A	PLC Link / COM1 MODBUS 從站 10 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A4B	PLC Link / COM1 MODBUS 從站 11 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A4C	PLC Link / COM1 MODBUS 從站 12 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A4D	PLC Link / COM1 MODBUS 從站 13 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A4E	PLC Link / COM1 MODBUS 從站 14 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A4F	PLC Link / COM1 MODBUS 從站 15 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A50	PLC Link / COM1 MODBUS 從站 16 無回應 (錯誤旗標 SM1591)	持續	保持	保持

7

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#9A51	PLC Link / COM1 MODBUS 從站 17 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A52	PLC Link / COM1 MODBUS 從站 18 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A53	PLC Link / COM1 MODBUS 從站 19 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A54	PLC Link / COM1 MODBUS 從站 20 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A55	PLC Link / COM1 MODBUS 從站 21 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A56	PLC Link / COM1 MODBUS 從站 22 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A57	PLC Link / COM1 MODBUS 從站 23 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A58	PLC Link / COM1 MODBUS 從站 24 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A59	PLC Link / COM1 MODBUS 從站 25 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A5A	PLC Link / COM1 MODBUS 從站 26 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A5B	PLC Link / COM1 MODBUS 從站 27 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A5C	PLC Link / COM1 MODBUS 從站 28 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A5D	PLC Link / COM1 MODBUS 從站 29 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A5E	PLC Link / COM1 MODBUS 從站 30 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A5F	PLC Link / COM1 MODBUS 從站 31 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A60	PLC Link / COM1 MODBUS 從站 32 無回應 (錯誤旗標 SM1591)	持續	保持	保持
16#9A61	PLC Link Mode 設定錯誤 (錯誤旗標 SM1589)	持續	保持	保持
16#9A62	PLC Link 輪詢次數設定錯誤 (錯誤旗標 SM1596)	持續	保持	保持
16#9A63	主機與通訊模組交握逾時 (錯誤旗標 SM1596)	持續	保持	保持
16#9A64	主機內無通訊模組參數 (錯誤旗標 SM1596)	持續	保持	保持

7

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#9B21	COM2 MODBUS 從站 1 通訊錯誤	持續	保持	保持
16#9B22	COM2 MODBUS 從站 2 通訊錯誤	持續	保持	保持
16#9B23	COM2 MODBUS 從站 3 通訊錯誤	持續	保持	保持
16#9B24	COM2 MODBUS 從站 4 通訊錯誤	持續	保持	保持
16#9B25	COM2 MODBUS 從站 5 通訊錯誤	持續	保持	保持
16#9B26	COM2 MODBUS 從站 6 通訊錯誤	持續	保持	保持
16#9B27	COM2 MODBUS 從站 7 通訊錯誤	持續	保持	保持
16#9B28	COM2 MODBUS 從站 8 通訊錯誤	持續	保持	保持
16#9B29	COM2 MODBUS 從站 9 通訊錯誤	持續	保持	保持
16#9B2A	COM2 MODBUS 從站 10 通訊錯誤	持續	保持	保持
16#9B2B	COM2 MODBUS 從站 11 通訊錯誤	持續	保持	保持
16#9B2C	COM2 MODBUS 從站 12 通訊錯誤	持續	保持	保持
16#9B2D	COM2 MODBUS 從站 13 通訊錯誤	持續	保持	保持
16#9B2E	COM2 MODBUS 從站 14 通訊錯誤	持續	保持	保持
16#9B2F	COM2 MODBUS 從站 15 通訊錯誤	持續	保持	保持
16#9B30	COM2 MODBUS 從站 16 通訊錯誤	持續	保持	保持
16#9B31	COM2 MODBUS 從站 17 通訊錯誤	持續	保持	保持
16#9B32	COM2 MODBUS 從站 18 通訊錯誤	持續	保持	保持
16#9B33	COM2 MODBUS 從站 19 通訊錯誤	持續	保持	保持
16#9B34	COM2 MODBUS 從站 20 通訊錯誤	持續	保持	保持
16#9B35	COM2 MODBUS 從站 21 通訊錯誤	持續	保持	保持
16#9B36	COM2 MODBUS 從站 22 通訊錯誤	持續	保持	保持
16#9B37	COM2 MODBUS 從站 23 通訊錯誤	持續	保持	保持
16#9B38	COM2 MODBUS 從站 24 通訊錯誤	持續	保持	保持
16#9B39	COM2 MODBUS 從站 25 通訊錯誤	持續	保持	保持
16#9B3A	COM2 MODBUS 從站 26 通訊錯誤	持續	保持	保持
16#9B3B	COM2 MODBUS 從站 27 通訊錯誤	持續	保持	保持
16#9B3C	COM2 MODBUS 從站 28 通訊錯誤	持續	保持	保持
16#9B3D	COM2 MODBUS 從站 29 通訊錯誤	持續	保持	保持
16#9B3E	COM2 MODBUS 從站 30 通訊錯誤	持續	保持	保持
16#9B3F	COM2 MODBUS 從站 31 通訊錯誤	持續	保持	保持
16#9B40	COM2 MODBUS 從站 32 通訊錯誤	持續	保持	保持
16#9B41	COM2 MODBUS 從站 1 無回應	持續	保持	保持
16#9B42	COM2 MODBUS 從站 2 無回應	持續	保持	保持
16#9B43	COM2 MODBUS 從站 3 無回應	持續	保持	保持
16#9B44	COM2 MODBUS 從站 4 無回應	持續	保持	保持

7

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#9B45	COM2 MODBUS 從站 5 無回應	持續	保持	保持
16#9B46	COM2 MODBUS 從站 6 無回應	持續	保持	保持
16#9B47	COM2 MODBUS 從站 7 無回應	持續	保持	保持
16#9B48	COM2 MODBUS 從站 8 無回應	持續	保持	保持
16#9B49	COM2 MODBUS 從站 9 無回應	持續	保持	保持
16#9B4A	COM2 MODBUS 從站 10 無回應	持續	保持	保持
16#9B4B	COM2 MODBUS 從站 11 無回應	持續	保持	保持
16#9B4C	COM2 MODBUS 從站 12 無回應	持續	保持	保持
16#9B4D	COM2 MODBUS 從站 13 無回應	持續	保持	保持
16#9B4E	COM2 MODBUS 從站 14 無回應	持續	保持	保持
16#9B4F	COM2 MODBUS 從站 15 無回應	持續	保持	保持
16#9B50	COM2 MODBUS 從站 16 無回應	持續	保持	保持
16#9B51	COM2 MODBUS 從站 17 無回應	持續	保持	保持
16#9B52	COM2 MODBUS 從站 18 無回應	持續	保持	保持
16#9B53	COM2 MODBUS 從站 19 無回應	持續	保持	保持
16#9B54	COM2 MODBUS 從站 20 無回應	持續	保持	保持
16#9B55	COM2 MODBUS 從站 21 無回應	持續	保持	保持
16#9B56	COM2 MODBUS 從站 22 無回應	持續	保持	保持
16#9B57	COM2 MODBUS 從站 23 無回應	持續	保持	保持
16#9B58	COM2 MODBUS 從站 24 無回應	持續	保持	保持
16#9B59	COM2 MODBUS 從站 25 無回應	持續	保持	保持
16#9B5A	COM2 MODBUS 從站 26 無回應	持續	保持	保持
16#9B5B	COM2 MODBUS 從站 27 無回應	持續	保持	保持
16#9B5C	COM2 MODBUS 從站 28 無回應	持續	保持	保持
16#9B5D	COM2 MODBUS 從站 29 無回應	持續	保持	保持
16#9B5E	COM2 MODBUS 從站 30 無回應	持續	保持	保持
16#9B5F	COM2 MODBUS 從站 31 無回應	持續	保持	保持
16#9B60	COM2 MODBUS 從站 32 無回應	持續	保持	保持
16#B100	I/O Connection 重複建立	持續	保持	保持
16#B106	多 Scanner 建立 I/O Connection 衝突	持續	保持	保持
16#B110	Adapter configuration 參數設定錯誤	持續	保持	保持
16#B111	Adapter RPI 參數設定錯誤	持續	保持	保持
16#B113	I/O connection 連線數不足	持續	保持	保持
16#B119	Non-Listen only 連線建立失敗	持續	保持	保持
16#B127	Adapter Input size 參數錯誤	持續	保持	保持
16#B128	Adapter output size 設定錯誤	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#B129	EDS 檔 Configuration path 參數錯誤	持續	保持	保持
16#B12D	Consumed tag 參數錯誤	持續	保持	保持
16#B12E	Produced tag 參數錯誤	持續	保持	保持
16#B203	I/O connection 通訊逾時	持續	保持	保持
16#B204	建立 I/O Connection 時通訊逾時	持續	保持	保持
16#B302	網路配置超過產品 PPS 規格	持續	保持	保持
16#B315	Adapter input/output instance 參數設定錯誤	持續	保持	保持
16#E206	備援控制器與主控制型號不一致	持續	保持	保持
16#E207	韌體版本不相容	持續	保持	保持
16#E208	備援控制器和主控制器乙太網路不在相同的實體網域	持續	保持	保持
16#E209	主系統和備援系統實際 I/O 配置不相符 (資格檢定期間)	持續	保持	保持
16#E20A	備援系統和主系統實際 I/O 配置不相符 (資格檢定過後)	持續	保持	保持
16#E20B	系統錯誤，請參考錯誤紀錄	持續	保持	保持
16#E20C	下載中，無法同步	持續	保持	保持
16#E20D	請參考備援主機的錯誤紀錄	持續	保持	保持
16#E20E	I/O 匯流排錯誤	持續	保持	保持
16#E20F	Heart beat 錯誤，請參考備援主機的錯誤紀錄	持續	保持	保持
16#E210	Heart beat 通訊逾時	持續	保持	保持
16#E211	同步資料失敗	持續	保持	保持
16#E212	備援系統切換中	持續	保持	保持
16#E213	PLC 無程式	持續	保持	保持
16#E214	PLC 程式毀損	持續	保持	保持
16#E215	掃描逾時	持續	保持	保持
16#E216	CPU 記憶體存取被拒	持續	保持	保持
16#E217	系統忙碌 RST	持續	保持	保持
16#E218	系統忙碌 CLR	持續	保持	保持
16#E219	系統開機未完成	持續	保持	保持
16#E21A	系統開機失敗	持續	保持	保持
16#E21B	CPU 參數毀損,請參考錯誤紀錄	持續	保持	保持
16#E21C	停電保持區塊異常,請參考錯誤紀錄	持續	保持	保持
16#E21D	CPU EIP 參數毀損,請參考錯誤紀錄	持續	保持	保持
16#E21E	I/O 配置表不存在	持續	保持	保持
16#E21F	I/O 配置表損毀	持續	保持	保持
16#E221	PLC 程式執行錯誤，請參考錯誤紀錄	持續	保持	保持
16#E230	備援系統乙太網路連線異常	持續	保持	保持
16#E260	主背板第 0 槽模組不支援備援系統	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#E261	主背板第 1 槽模組不支援備援系統	持續	保持	保持
16#E262	主背板第 2 槽模組不支援備援系統	持續	保持	保持
16#E263	主背板第 3 槽模組不支援備援系統	持續	保持	保持
16#E264	主背板第 4 槽模組不支援備援系統	持續	保持	保持
16#E265	主背板第 5 槽模組不支援備援系統	持續	保持	保持
16#E266	主背板第 6 槽模組不支援備援系統	持續	保持	保持
16#E267	主背板第 7 槽模組不支援備援系統	持續	保持	保持
16#E268	主背板第 8 槽模組不支援備援系統	持續	保持	保持
16#E269	主背板第 9 槽模組不支援備援系統	持續	保持	保持
16#E26A	主背板第 10 槽模組不支援備援系統	持續	保持	保持
16#E26B	主背板第 11 槽模組不支援備援系統	持續	保持	保持
16#E270	主背板第 0 槽網路模組網路線未連接	持續	保持	保持
16#E271	主背板第 1 槽網路模組網路線未連接	持續	保持	保持
16#E272	主背板第 2 槽網路模組網路線未連接	持續	保持	保持
16#E273	主背板第 3 槽網路模組網路線未連接	持續	保持	保持
16#E274	主背板第 4 槽網路模組網路線未連接	持續	保持	保持
16#E275	主背板第 5 槽網路模組網路線未連接	持續	保持	保持
16#E276	主背板第 6 槽網路模組網路線未連接	持續	保持	保持
16#E277	主背板第 7 槽網路模組網路線未連接	持續	保持	保持
16#E278	主背板第 8 槽網路模組網路線未連接	持續	保持	保持
16#E279	主背板第 9 槽網路模組網路線未連接	持續	保持	保持
16#E27A	主背板第 10 槽網路模組網路線未連接	持續	保持	保持
16#E27B	主背板第 11 槽網路模組網路線未連接	持續	保持	保持
16#E280	主背板第 0 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E281	主背板第 1 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E282	主背板第 2 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E283	主背板第 3 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E284	主背板第 4 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E285	主背板第 5 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E286	主背板第 6 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E287	主背板第 7 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E288	主背板第 8 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E289	主背板第 9 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E28A	主背板第 10 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E28B	主背板第 11 槽網路通訊模組網路模組 IP 偵測失敗	持續	保持	保持
16#E290	主背板第 0 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持

錯誤代碼	說明	CPU 狀態	燈號狀態	
			ERROR	BUS FAULT
16#E291	主背板第 1 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E292	主背板第 2 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E293	主背板第 3 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E294	主背板第 4 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E295	主背板第 5 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E296	主背板第 6 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E297	主背板第 7 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E298	主背板第 8 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E299	主背板第 9 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E29A	主背板第 10 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E29B	主背板第 11 槽網路通訊模組通訊口 heart beat 偵測失敗	持續	保持	保持
16#E2A0	主背板第 0 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A1	主背板第 1 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A2	主背板第 2 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A3	主背板第 3 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A4	主背板第 4 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A5	主背板第 5 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A6	主背板第 6 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A7	主背板第 7 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A8	主背板第 8 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2A9	主背板第 9 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2AA	主背板第 10 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持
16#E2AB	主背板第 11 槽網路通訊模組網路模組 IP 偵測尚未執行	持續	保持	保持

7

7.1.2 類比模組與溫度模組錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU BUS FAULT	MODULE ERROR
16#A000	CH0 輸入信號超出硬體規格	閃爍	
16#A001	CH1 輸入信號超出硬體規格	閃爍	
16#A002	CH2 輸入信號超出硬體規格	閃爍	
16#A003	CH3 輸入信號超出硬體規格	閃爍	
16#A004	CH4 輸入信號超出硬體規格	閃爍	
16#A005	CH5 輸入信號超出硬體規格	閃爍	
16#A006	CH6 輸入信號超出硬體規格	閃爍	
16#A007	CH7 輸入信號超出硬體規格	閃爍	

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A400	CH0 輸入信號超出硬體規格	恆亮	
16#A401	CH1 輸入信號超出硬體規格	恆亮	
16#A402	CH2 輸入信號超出硬體規格	恆亮	
16#A403	CH3 輸入信號超出硬體規格	恆亮	
16#A404	CH4 輸入信號超出硬體規格	恆亮	
16#A405	CH5 輸入信號超出硬體規格	恆亮	
16#A406	CH6 輸入信號超出硬體規格	恆亮	
16#A407	CH7 輸入信號超出硬體規格	恆亮	
16#A600	插槽電源異常	恆亮	
16#A601	電源異常	恆亮	
16#A602	內部錯誤 · CJC 補償異常	恆亮	
16#A603	內部錯誤 · 出廠校正異常	恆亮	
16#A800	CH0 輸入信號超出硬體規格	OFF	
16#A801	CH1 輸入信號超出硬體規格	OFF	
16#A802	CH2 輸入信號超出硬體規格	OFF	
16#A803	CH3 輸入信號超出硬體規格	OFF	
16#A804	CH4 輸入信號超出硬體規格	OFF	
16#A805	CH5 輸入信號超出硬體規格	OFF	
16#A806	CH6 輸入信號超出硬體規格	OFF	
16#A807	CH7 輸入信號超出硬體規格	OFF	

- *1. 關於輸入信號超出硬體規格與工程值超出極限兩種錯誤，模組會依據使用者所自訂的亮燈方式，來決定送出的錯誤代碼是使用#A000~16#A00F、#A400~16#A40F、#A800~16#A80F 的那個區段。

7.1.3 AH02HC-5A/AH04HC-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A001	CH0 線性累加超過範圍	閃爍	
16#A002	CH0 前置比例值設定超過範圍	閃爍	
16#A003	CH0 移動平均值設定超過範圍	閃爍	
16#A004	CH0 比較值設定超過範圍	閃爍	
16#A005	CH0 警報輸出設定極限值錯誤	閃爍	
16#A006	CH0 中斷編號設定超過範圍	閃爍	
16#A011	CH1 線性累加超過範圍	閃爍	
16#A012	CH1 前置比例值設定超過範圍	閃爍	

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A013	CH1 移動平均值設定超過範圍	閃爍	
16#A014	CH1 比較值設定超過範圍	閃爍	
16#A015	CH1 警報輸出設定極限值錯誤	閃爍	
16#A016	CH1 中斷編號設定超過範圍	閃爍	
16#A021	CH2 線性累加超過範圍	閃爍	
16#A022	CH2 前置比例值設定超過範圍	閃爍	
16#A023	CH2 移動平均值設定超過範圍	閃爍	
16#A024	CH2 比較值設定超過範圍	閃爍	
16#A025	CH2 警報輸出設定極限值錯誤	閃爍	
16#A026	CH2 中斷編號設定超過範圍	閃爍	
16#A031	CH3 線性累加超過範圍	閃爍	
16#A032	CH3 前置比例值設定超過範圍	閃爍	
16#A033	CH3 移動平均值設定超過範圍	閃爍	
16#A034	CH3 比較值設定超過範圍	閃爍	
16#A035	CH3 警報輸出設定極限值錯誤	閃爍	
16#A036	CH3 中斷編號設定超過範圍	閃爍	

7.1.4 AH05PM-5A/AH10PM-5A/AH15PM-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A002	使用副程式無內容	閃爍	
16#A003	CJ、CJN、JMP 指令缺少對應的指標	閃爍	
16#A004	主程式中有副程式指標	閃爍	
16#A005	缺少副程式	閃爍	
16#A006	同一程式中的指標重複	閃爍	
16#A007	副程式指標重複	閃爍	
16#A008	不同副程式中的跳躍指令指標重複	閃爍	
16#A009	跳躍指令與呼叫副程式指令使用相同指標	閃爍	
16#A00B	單段速目標位置 (I) 錯誤	閃爍	
16#A00C	單軸運動目標位置 (II) 錯誤	閃爍	
16#A00D	單軸運轉速度 (I) 設定錯誤	閃爍	
16#A00E	單軸運轉速度 (II) 設定錯誤	閃爍	
16#A00F	原點復歸速度 (V _{RT}) 設定錯誤	閃爍	
16#A010	原點復歸減速速度 (V _{CR}) 設定錯誤	閃爍	
16#A011	寸動 JOG 速度設定錯誤	閃爍	

7

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A012	單軸正轉運動正向脈波禁止輸出	閃爍	
16#A013	單軸反向運動反向脈波禁止輸出	閃爍	
16#A014	到達極限	閃爍	
16#A015	裝置元件使用範圍錯誤	閃爍	
16#A017	V/Z 修飾錯誤	閃爍	
16#A018	浮點數轉換錯誤	閃爍	
16#A019	BCD 轉換錯誤	閃爍	
16#A01A	除法演算錯誤 (除數 = 0)	閃爍	
16#A01B	一般程式錯誤	閃爍	
16#A01C	LD/LDI 指令連續使用 9 次以上	閃爍	
16#A01D	RPT~RPE 超過 1 層以上	閃爍	
16#A01E	SRET 使用在 RPT~RPE 之間	閃爍	
16#A01F	主程式沒有 M102 結束指令或運動程式沒有 M2 結束指令	閃爍	
16#A020	使用錯誤指令或是使用裝置超過範圍	閃爍	

7.1.5 AH20MC-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A002	使用副程式無內容	閃爍	
16#A003	CJ、CJN、JMP 指令缺少對應的指標	閃爍	
16#A004	主程式中有副程式指標	閃爍	
16#A005	缺少副程式	閃爍	
16#A006	同一程式中的指標重複	閃爍	
16#A007	副程式指標重複	閃爍	
16#A008	不同副程式中的跳躍指令指標重複	閃爍	
16#A009	跳躍指令與呼叫副程式指令使用相同指標	閃爍	
16#A00B	單段速目標位置 (I) 錯誤	閃爍	
16#A00C	單軸運動目標位置 (II) 錯誤	閃爍	
16#A00D	單軸運轉速度 (I) 設定錯誤	閃爍	
16#A00E	單軸運轉速度 (II) 設定錯誤	閃爍	
16#A00F	原點復歸速度 (V _{RT}) 設定錯誤	閃爍	
16#A010	原點復歸減速速度 (V _{CR}) 設定錯誤	閃爍	
16#A011	寸動 JOG 速度設定錯誤	閃爍	
16#A012	單軸正轉運動正向脈波禁止輸出	閃爍	

7

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A013	單軸反向運動反向脈波禁止輸出	閃爍	
16#A014	到達極限	閃爍	
16#A015	裝置元件使用範圍錯誤	閃爍	
16#A017	V/Z 修飾錯誤	閃爍	
16#A018	浮點數轉換錯誤	閃爍	
16#A019	BCD 轉換錯誤	閃爍	
16#A01A	除法演算錯誤 (除數 = 0)	閃爍	
16#A01B	一般程式錯誤	閃爍	
16#A01C	LD/LDI 指令連續使用 9 次以上	閃爍	
16#A01D	RPT~RPE 超過 1 層以上	閃爍	
16#A01E	SRET 使用在 RPT~RPE 之間	閃爍	
16#A01F	主程式沒有 M102 結束指令或運動程式沒有 M2 結束指令	閃爍	
16#A020	使用錯誤指令或是使用裝置超過範圍	閃爍	

7.1.6 AH10EN-5A/AH15EN-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A001	Host 1 IP 位址衝突	閃爍	
16#A002	Host 2 IP 位址衝突	閃爍	
16#A003	Host 1 DHCP 失敗	閃爍	
16#A004	Host 2 DHCP 失敗	閃爍	
16#A401	硬體錯誤	恆亮	
16#A402	系統初始化失敗	恆亮	

7.1.7 AH10SCM-5A/AH15SCM-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A002	UD Link 設定錯誤或是通訊失敗	閃爍	
16#A401	硬體發生錯誤	恆亮	
16#A804	COM Port 通訊錯誤	OFF	
16#A808	MODBUS 通訊錯誤	OFF	

7.1.8 AH10DNET-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態		
		CPU	MODULE	
		BUS FAULT	MS	NS
16#A0F0	AH10DNET-5A 掃描模組的站號與其他節點重複，或超出範圍	紅燈閃爍	綠燈閃爍	紅燈恆亮
16#A0F1	沒有將任何從站配置到 AH10DNET-5A 掃描列表中	紅燈閃爍	綠燈閃爍	綠燈恆亮
16#A0F2	AH10DNET-5A 掃描模組的工作電壓過低	紅燈閃爍	紅燈閃爍	紅燈閃爍
16#A0F3	AH10DNET-5A 掃描模組進入測試模式	紅燈閃爍	橙燈恆亮	橙燈恆亮
16#A0F4	AH10DNET-5A 掃描模組進入 Bus-OFF 狀態	紅燈閃爍	綠燈恆亮	紅燈恆亮
16#A0F5	AH10DNET-5A 掃描模組檢測到 DeviceNet 網路沒有電源	紅燈閃爍	紅燈閃爍	紅燈恆亮
16#A0F6	AH10DNET-5A 掃描模組的內部存儲單元出錯	紅燈閃爍	紅燈恆亮	綠燈閃爍
16#A0F7	AH10DNET-5A 掃描模組的資料交換單元出錯	紅燈閃爍	紅燈恆亮	綠燈閃爍
16#A0F8	AH10DNET-5A 掃描模組序列號檢測出錯	紅燈閃爍	紅燈恆亮	綠燈閃爍
16#A0F9	AH10DNET-5A 掃描模組讀取或寫入配置資料出錯	紅燈閃爍	紅燈恆亮	紅燈恆亮
16#A0FA	AH10DNET-5A 掃描模組的站號與掃描列表中配置的從站站號重複	紅燈閃爍	綠燈恆亮	紅燈恆亮
16#A0FB	AH10DNET 和 AH CPU 之間資料交換失敗	紅燈閃爍	綠燈恆亮	綠燈恆亮
16#A0FC	從站出錯、AHRTU-DNET 背板插槽上的模組出錯或 AHRTU-DNET 從背板連接不正常	紅燈閃爍	紅燈閃爍	綠燈恆亮

7.1.9 AH10PFBM-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態			
		CPU	MODULE		
		BUS FAULT	RUN	SYS	DP
16#A001	主站設定為空	紅燈閃爍	綠燈 恆亮	綠燈 恆亮	綠燈 閃爍
16#A003	主站進入檢測模式	紅燈閃爍	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A005	主站內部晶片通訊逾時	紅燈閃爍	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A00B	與 PLC 資料交換逾時	紅燈閃爍	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A402	PLC 沒有為主站分配 I/O 映射區	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A404	主站初始化錯誤	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A406	內部儲存單元出錯	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A407	資料交換單元出錯	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A408	主站序列號偵測出錯	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A4E2	主站偵測到有從站全部掉線	紅燈 恆亮	OFF	綠燈 恆亮	紅燈 恆亮
	主站偵測到有部分從站掉線	紅燈 恆亮	OFF	綠燈 恆亮	紅燈 閃爍
16#A4E6	主站檢測到 AHRTU-PFBS-5A 連接的模組出錯	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮	綠燈 恆亮

7

7.1.10 AH10PFBS-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態		
		CPU	MODULE	
		BUS FAULT	RUN	NET
16#A4F0	AH10PFBS-5A 節點地址超出範圍	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮

錯誤代碼	說明	燈號狀態		
		CPU	MODULE	
		BUS FAULT	RUN	NET
16#A4F1	內部硬體錯誤	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A4F2	參數化錯誤	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A4F3	組態錯誤	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A4F4	GPIO 檢測出錯	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A4F5	AH10PFBS-5A 進入工廠測試模式	紅燈 恆亮	綠燈 恆亮	綠燈 恆亮
16#A4F6	1. AH10PFBS-5A 未接入 PFOFIBUS-DP 網路 2. PFOFIBUS-DP 主站沒有配置 AH10PFBS-5A 從站 或配置 AH10PFBS-5A 節點位址與實際連接的不符	紅燈 恆亮	綠燈 恆亮	紅燈 恆亮

7.1.11 AH10COPM-5A 錯誤代碼對應燈號

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A0B0	心跳訊息逾時	閃爍	紅燈雙閃
16#A0B1	從站返回的 PDO 長度與與節點列表中設定的 PDO 資料長度不符	閃爍	OFF
16#A0B2	主站 NodeGuard 訊息逾時	閃爍	紅燈雙閃
16#A0E0	AH10COPM-5A 接收到從站發送的緊急資訊	閃爍	OFF
16#A0E1	從站返回的 PDO 資料長度與節點列表中設定的 PDO 資料長度不符	閃爍	OFF
16#A0E2	未接收到從站 PDO	閃爍	OFF
16#A0E3	自動 SDO 下載失敗	閃爍	OFF
16#A0E4	PDO 參數設定失敗	閃爍	OFF
16#A0E5	關鍵參數設定有誤	閃爍	OFF
16#A0E6	實際網路配置與設定配置不符	閃爍	OFF
16#A0E7	從站錯誤控制逾時	閃爍	紅燈雙閃
16#A0E8	主從站站號重複	閃爍	OFF

錯誤代碼	說明	燈號狀態	
		CPU	MODULE
		BUS FAULT	ERROR
16#A0F1	CANopen Builder 軟體節點列表沒有增加從站	閃爍	OFF
16#A0F3	AH10COPM-5A 處於錯誤狀態	閃爍	OFF
16#A0F4	檢測到匯流排脫離 (Bus-off)	閃爍	紅燈常亮
16#A0F5	AH10COPM-5A 節點位址設定錯誤	閃爍	OFF
16#A0F6	內部錯誤：工廠製造流程出錯	閃爍	OFF
16#A0F7	內部錯誤：GPIO 檢測出錯	閃爍	OFF
16#A0F8	內部錯誤：內部記憶體檢測出錯	閃爍	OFF
16#A0F9	低電壓檢測錯誤	閃爍	OFF
16#A0FA	AH10COPM-5A 韌體內部處於錯誤狀態	閃爍	OFF
16#A0FB	AH10COPM-5A 的發送暫存區已滿	閃爍	OFF
16#A0FC	AH10COPM-5A 的接收暫存區已滿	閃爍	OFF