

SYSMAC CJ Series

CJ1G/H-CPU□□H, CJ1M-CPU□□, CJ1G-CPU□□

Programmable Controllers

**CJ1/CJ1M 系列
設定操作及指令中文技術手冊**

OMRON

目錄

第 1 章	
簡介	1
1-1 CJ1 與 CJ1-H CPU 模組比較.....	2
第 2 章	
規格與系統架構	7
2-1 規格.....	8
2-1-1 效能規格.....	8
2-1-2 一般規格.....	15
2-2 CPU 模組組件與功能.....	16
2-2-1 CPU 模組組件.....	16
2-3 CPU 匯流排模組設定區域容量.....	18
2-4 I/O 表設定表列.....	19
2-4-1 CJ 系列基本 I/O 模組.....	19
2-4-2 CJ 系列特殊 I/O 模組.....	20
2-4-3 CJ 系列 CPU 匯流排模組.....	20
第 3 章	
型號說明、功能、與尺寸規格	21
3-1 CPU 模組.....	22
3-1-1 型號.....	22
3-1-2 組件.....	23
3-2 RS-232C 埠規格.....	26
3-3 電源供應器模組.....	28
3-3-1 電源供應器模組型號.....	28
3-3-2 組件與開關設定.....	28
3-4 CJ 系列基本 I/O 模組.....	29
3-4-1 CJ 系列基本 I/O 模組附端子區塊.....	29
3-4-2 CJ 系列 32/64 點基本 I/O 模組附連接器.....	31
第 4 章	
指撥開關設定	37
4-1 概觀.....	40

目錄

4-2 詳情	41
第 5 章	
PLC 設定	41
5-1 PLC 設定	42
5-1-1 PLC 設定概觀	42
5-1-2 PLC 設定	44
5-2 PLC 系統設定之解說	70
第 6 章	
I/O 配置與資料交換	79
6-1 I/O 配置	80
6-1-1 基本 I/O 模組	80
6-1-2 基本 I/O 模組的 I/O 配置	81
6-1-3 I/O 配置至特殊 I/O 模組	83
6-1-4 I/O 配置至 CPU 匯流排模組	83
6-1-5 內藏 I/O 的資料區域配置 (僅 CJ1M CPU 模組)	84
6-2 以 CPU 匯流排模組進行資料交換	85
6-2-1 特殊 I/O 模組	85
6-2-2 CPU 匯流排模組	87
第 7 章	
記憶體區域	89
7-1 I/O 記憶體區域	90
7-1-1 I/O 記憶體區域結構	90
7-1-2 資料區域概觀	92
7-1-3 資料區域性質	96
7-2 I/O 區域 (輸出 / 輸出 relay)	97
7-3 資料 Link 區域 (Data Link Area)	102
7-4 CPU 匯流排模組區域	103
7-5 特殊 I/O 模組區域	104
7-6 串列 PLC Link 區域	105
7-7 DeviceNet 區域 (DeviceNet Area)	106
7-8 內部 I/O 區域	107

目錄

7-9 保持區域 (Holding Area).....	108
7-10 輔助區域 (Auxiliary Area).....	109
7-11 TR (暫時記憶繼電器) 區域.....	129
7-12 計時器區域.....	130
7-13 計數器區域.....	131
7-14 資料記憶體 (DM) 區域.....	132
7-15 擴充資料記憶體 (EM) 區域.....	133
7-16 工作單旗標 (Task Flags).....	134
7-17 條件旗標 (Condition Flags).....	135
7-18 時序脈衝.....	136
7-19 參數區域.....	137
7-19-1 PLC 設定.....	137
7-19-2 登錄 I/O 表.....	137
7-19-3 路由表 (Routing Tables).....	138
7-19-4 CPU 匯流排模組設定.....	139
第 8 章	
指令功能.....	141
8-1 程序輸入指令.....	142
8-2 程序輸出指令.....	144
8-3 程序控制指令.....	146
8-4 計時器與計數器指令.....	150
8-5 比較指令.....	154
8-6 資料搬移指令.....	157
8-7 資料移位指令.....	160
8-8 遞增 / 遞減指令.....	164
8-9 四則運算指令.....	165
8-10 轉換指令.....	170
8-11 邏輯 (Logic) 指令.....	176
8-12 特殊算術指令.....	178
8-13 浮點運算指令.....	179
8-14 雙精度浮點指令點指令 (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D).....	183
8-15 表格資料處理指令.....	188
8-16 資料控制指令.....	192
8-17 副程式指令.....	195

目錄

8-18 中斷控制指令	197
8-19 高速計數器與脈衝輸出指令 (僅 CJ1M-CPU22/23)	199
8-20 步進指令	200
8-21 基本 I/O 模組指令	201
8-22 串列通訊指令	202
8-23 網路指令	203
8-24 檔案記憶體指令	205
8-25 顯示指令	206
8-26 時鐘指令	206
8-27 除錯指令	207
8-28 故障診斷指令	208
8-29 其他指令	209
8-30 區塊規劃指令	210
8-31 字串處理指令	216
8-32 工作單控制指令	220

第 1 章 簡介

本章介紹 CJ 系列 PLC 與以前的 C200HX/HG/HE PLC 之相異處。

1-1 CJ1 與 CJ1-H CPU 模組比較	2
--------------------------------	---

1-1 CJ1 與 CJ1-H CPU 模組比較

項目		CJ1-H CPU 模組 (CJ1H-CPU6 □H)	CJ1M CPU 模組 (CJ1M-CPU □□□)	CJ1 CPU 模組 (CJ1G-CPU4 □)
指令執行時間	基本指令	LD : 0.02 μs	LD : 0.10 μs	0.08 μs
		OUT : 0.02 μs	OUT : 0.35 μs	0.21 μs
	特殊指令	範例		
		XFER : 300 μs (對於 1,000 個字組)	XFER : 650 μs (對於 1,000 個字組)	XFER : 633 μs (對於 1,000 個字組)
		BSET : 200 μs (對於 1,000 個字組)	BSET : 400 μs (對於 1,000 個字組)	BSET : 278 μs (對於 1,000 個字組)
		BCD 運算 : 8.2 μs 以上	BCD 運算 : 18.9 μs 以上	14 μs 以上
		BIN 運算 : 0.18 μs 以上	BIN 運算 : 0.30 μs 以上	0.37 μs 以上
		浮點運算 : 8 μs 以上	浮點運算 : 13.3 μs 以上	10 μs 以上
SBS/RET : 2.1 μs	SBS/RET : 3.8 μs	37 μs		
監督處理時間		一般模式 : 0.3 ms 平行模式 : 0.2 ms	一般模式 : 0.5 ms	0.5 ms
執行時序	CPU 執行處理模式	以下四種模式之一： 1. 一般 (指令與週邊服務連續進行) 2. 週邊服務優先模式 (在特定週期與時間下中斷指令執行而進行週邊服務；接著會進行更新) 3. I/O 記憶存取同步之並行處理模式 (指令執行與週邊服務並行，並同步存取 I/O 記憶體) 4. I/O 記憶存取非同步之並行處理模式 (指令執行與週邊服務並行，但沒有同步存取 I/O 記憶體)	以下二種模式之一： 1. 一般 (指令與週邊服務連續進行) 2. 週邊服務優先模式 (在特定週期與時間下中斷指令執行而進行週邊服務；接著會進行更新)	以下二種模式之一： 1. 一般 (指令與週邊服務連續進行) 2. 週邊服務優先模式 (在特定週期與時間下中斷指令執行而進行週邊服務；接著會進行更新)
	CPU 匯流排模組特殊更新	資料 Link DeviceNet (裝置網路) 遠端 I/O 協定巨集送出 / 接收資料 位於 CPU 匯流排模組的 CIO 與 DM 區域字組之更新	在 I/O 更新週期期間或經由特殊 CPU 匯流排模組 I/O 更新指令 (DLNK(226))	在 I/O 更新週期期間

項目		CJ1-H CPU 模組 (CJ1H-CPU6 □H)	CJ1M CPU 模組 (CJ1M-CPU □□□)	CJ1 CPU 模組 (CJ1G-CPU4 □)
工作單 (Tasks)	循環執行的中斷工作單經由 TKON 指令 (稱為 "額外循環工作單")	支援 (至多 256 個額外循環工作, 最多增加循環工作總數至 288)	支援 (至多 256 個額外循環工作, 最多增加循環工作總數至 288)	不支援 (無額外循環工作; 最多 32 個循環工作)
	索引與資料暫存器的個別 / 共通規格	支援 若使用共通暫存器, 則可減少工作間接換的時間。	支援 若使用共通暫存器, 則可減少工作間接換的時間。	不支援 (每個工作均只有個別暫存器)
	工作起動時之初值化	支援 支援工作啓動旗標	支援 支援工作啓動旗標	第一次執行時只有工作旗標。
	由多個工作起動副程式	可以定義被多個工作呼叫的整體副程式。	可以定義被多個工作呼叫的整體副程式。	不支援
	排定中斷工作之排定中斷間隔	1 ms 至 9,999 ms, 或 10 ms 至 99,990 ms, 單位為 1 ms 或 10 ms	除原支援的間隔 (1 ms 至 9,999 ms, 或 10 ms 至 99,990 ms, 單位為 1 ms 或 10 ms) 外, 也支援 0.5 ms 至 999.9 ms 單位為 0.1 ms 的間隔。	1 ms 至 9,999 ms, 或 10 ms 至 99,990 ms, 單位為 1 ms 或 10 ms
在指令執行期間中斷工作的執行時序	用於下述以外的指令	當符合中斷工作條件時起動中斷工作, 任何指令的執行都會中斷。若循環工作 (包括額外循環工作) 存取與所中斷指令相同的資料區塊字組, 則資料可能會不一致。為確保資料的一致性, 必需使用 DI 與 EI 指令, 以便在程式的特定部份中將中斷致能或除能。		
	用於 BIT 計數器 (BCNT) 或 BLOCK TRANSFER (XFER) 指令	只有在指令被完全執行後才會起動中斷工作, 以便能夠即使於指令與中斷工作存取相同的資料區塊字組時也能確保資料的一致性。		
除錯	備份至記憶卡 (簡易備份功能)	除了右方所列的資料外, 連接到 CPU 插槽或延伸插槽上裝置的資料也能備份至記憶卡 (經由面板上的按鈕)。這在更記模組時是非常有用的。備份資料包含掃描裝置網路模組的清單、串列通訊模組的協定巨集等。	除了右方所列的資料外, 連接到 CPU 插槽或延伸插槽上裝置的資料也能備份至記憶卡 (經由面板上的按鈕)。這在更記模組時是非常有用的。備份資料包含掃描裝置網路模組的清單、串列通訊模組的協定巨集等。	在 CPU 模組中只有使用者程式參數與 I/O 記憶。
	自動把使用者程式與參數區塊備份至快閃記憶體	支援 (無記憶卡時可以無電池操作) 使用者程式與參數區塊資料由 CX-P 軟體、檔案記憶體等裝置轉移到 CPU 模組時會自動備份到快閃記憶體。	支援 (無記憶卡時可以無電池操作) 使用者程式與參數區塊資料由 CX-P 軟體、檔案記憶體等裝置轉移到 CPU 模組時會自動備份到快閃記憶體。	不支援
I/O 表	I/O 表建立錯誤的詳細資訊	當 I/O 表因故無法建立時, 詳細的 I/O 表錯誤資訊是儲存於 A261。	當 I/O 表因故無法建立時, 詳細的 I/O 表錯誤資訊是儲存於 A261。	不支援
	在規劃主控器上顯示第一軌字組設定	若規劃主控器上已詳述第一軌字組, 則可以加以確認。第一軌字組由 CX-P 軟體所指定, 原來是無法由規劃主控器加以確認的。	若規劃主控器上已詳述第一軌字組, 則可以加以確認。第一軌字組由 CX-P 軟體所指定, 原來是無法由規劃主控器加以確認的。	不支援

項目		CJ1-H CPU 模組 (CJ1H-CPU6 □ H)	CJ1M CPU 模組 (CJ1M-CPU □ □ □)	CJ1 CPU 模組 (CJ1G-CPU4 □)
循序指令	微分型 LD, NOT, AND, NOT, OR, NOT 指令	支援	支援	不支援 (可藉由將微分型 LD, AND, OR 指令和 NOT 指令加以組合而達到相同的效果。)
	OUTB, SETB, 與 RSTB 指令以修改 DM 與 EM 區塊字組中的個別位元	支援	支援	不支援
計時器/計數器指令	更新 TIM, TIMH, TMHH, TTIM, TIML, MTIM, CNT, CNTR, CNR, TIMW, TMHW, CNTW 指令的 PV 之格式	可選擇 BCD 或 BIN (以 3.0 版或更新的 CX-P 軟體)。	可選擇 BCD 或 BIN (以 3.0 版或更新的 CX-P 軟體)。	僅 BCD
特殊運算指令	由 APR 指令指定 32-位元的有號資料線座標與 X 軸起始點	支援	支援	不支援
浮點運算指令	單精度計算與轉換	支援 (可計算標準差)	支援 (可計算標準差)	不支援
	在單精度浮點數與 ASCII 碼間轉換	支援 浮點數可轉換為 ASCII 碼以顯示於 PT 上。 量測裝置上的 ASCII 字串可以轉換為可計算的浮點數。	支援 浮點數可轉換為 ASCII 碼以顯示於 PT 上。 量測裝置上的 ASCII 字串可以轉換為可計算的浮點數。	不支援
	雙精度計算與轉換	支援 (可定位高精度)	支援 (可定位高精度)	不支援
字串、表格資料與資料移位指令	字串與表格資料指令執行	資料處理可以正常或在背景進行 (由各指令指定)。 (使用時間片段來在數個循環中處理指令以減少這些指令在循環時間中的效應。)	資料處理可以正常或在背景進行 (由各指令指定)。 (使用時間片段來在數個循環中處理指令以減少這些指令在循環時間中的效應。)	僅一般處理
	以表格處理指令進行堆疊插入/刪除與置換及堆疊計數	支援 有效追蹤輸送帶上的工件。	支援 有效追蹤輸送帶上的工件。	不支援
資料控制指令	附自我調整的 PID	支援 (省略 PID 常數的調整)	支援 (省略 PID 常數的調整)	不支援
副程式指令	整體副程式	支援 (GSBS, GSBN, 與 GRET 指令) 使副程式結構化更容易。	支援 (GSBS, GSBN, 與 GRET 指令) 使副程式結構化更容易。	不支援
失敗診斷指令	FAL 的錯誤記錄儲存	支援 FAL 可以在不指定錯誤記錄下執行 (只有系統 FAL 錯誤會記錄於錯誤記錄中)	支援 FAL 可以在不指定錯誤記錄下執行 (只有系統 FAL 錯誤會記錄於錯誤記錄中)	不支援
	以 FAL/FALS 模擬錯誤	支援 在除錯中可以模擬系統的嚴重錯誤與非嚴重錯誤。	支援 在除錯中可以模擬系統的嚴重錯誤與非嚴重錯誤。	不支援
資料比較指令	區域 RANGE COMPARE (ZCP) 與 DOUBLE RANGE COMPARE (ZCPL)	支援	支援	不支援
CVM1/CV 的索引暫存器真實 I/O 位址轉換	程式與真實 I/O 記憶體位址相容於 CVM1/CV- 系列 PLC	CVM1/CV- 系列真實 I/O 記憶體位址可以轉換為 CJ 系列位址並置入索引暫存器中, 而 CJ 系列索引暫存器中真實 I/O 記憶體位址也可以轉換為 CVM1/CV- 系列位址。	CVM1/CV- 系列真實 I/O 記憶體位址可以轉換為 CJ 系列位址並置入索引暫存器中, 而 CJ 系列索引暫存器中真實 I/O 記憶體位址也可以轉換為 CVM1/CV- 系列位址。	不支援

項目		CJ1-H CPU 模組 (CJ1H-CPU6 □ H)	CJ1M CPU 模組 (CJ1M-CPU □ □ □)	CJ1 CPU 模組 (CJ1G-CPU4 □)
狀態旗標 儲存與載 入	與CVM1/CV-系列PLC 相容	可以使用 SAVE CONDITION FLAGS (CCS) 與 LOAD CONDITION FLAGS (CCL) 指令來儲存或載入狀 態旗標的狀態值，以便能應 用於必須把狀態旗標在不同的 程式位置、工作或循環間 傳送的情況。	可以使用 SAVE CONDITION FLAGS (CCS) 與 LOAD CONDITION FLAGS (CCL) 指令來儲存或載入狀 態旗標的狀態值，以便能應 用於必須把狀態旗標在不同的 程式位置、工作或循環間 傳送的情況。	不支援
於模組起 動處理後 的操作	CPU 模組起動	在監看或執行模式均可以在 PLC設定中指定起動或不起 動(待命)CPU 模組，即使 是某模組沒有完全起動也可 以。	在監看或執行模式均可以在 PLC設定中指定起動或不起 動(待命)CPU 模組，即使 是某模組沒有完全起動也可 以。	CPU 模組待命(固定)
在程式區段使電源不會中斷		支援 即使檢測到電源中斷並經確 認，也可以執行 DI 與 EI 間 的指令而不會進行電源? 閉 之處理。	支援 即使檢測到電源中斷並經確 認，也可以執行 DI 與 EI 間 的指令而不會進行電源? 閉 之處理。	不支援
狀態旗標操作		等於、反向與錯誤旗標在執 行以下指令時均會維持。 TIM, TIMH, TMHH, CNT, IL, ILC, JMP0, JME0, XCHG, XCGL, MOVR, 輸 入比較指令, CMP, CMPL, CPS, CPSL, TST, TSTN, STC, 與 CLC。	等於、反向與錯誤旗標在執 行以下指令時均會維持。 TIM, TIMH, TMHH, CNT, IL, ILC, JMP0, JME0, XCHG, XCGL, MOVR, 輸 入比較指令, CMP, CMPL, CPS, CPSL, TST, TSTN, STC, 與 CLC。	等於、反向與錯誤旗標在執 行以下指令時均會變為 OFF。 TIM, TIMH, TMHH, CNT, IL, ILC, JMP0, JME0, XCHG, XCGL, MOVR, 輸入比較指 令, CMP, CMPL, CPS, CPSL, TST, TSTN, STC, 與 CLC。
內建 I/O		不支援	CJ1M-CPU2 □	不支援
串列 PLC Link		不支援	支援	不支援
0.1-ms 排程中斷		不支援	支援	不支援
電池		CPM2A-BAT01	CJ1W-BAT01	CPM2A-BAT01

第 2 章 規格與系統架構

本章表列各種不同模組的標準型號、模組規格、系統架構及其間的比較。

2-1	規格	8
2-1-1	效能規格	8
2-1-2	一般規格	15
2-2	CPU 模組組件與功能	16
2-2-1	CPU 模組組件	16
2-3	CPU 匯流排模組設定區域容量	18
2-4	I/O 表設定表列	19
2-4-1	CJ 系列基本 I/O 模組	19
2-4-2	CJ 系列特殊 I/O 模組	20
2-4-3	CJ 系列 CPU 匯流排模組	20

2-1 規格

2-1-1 效能規格

CJ1-H CPU 模組

CPU	CJ1H-CPU66H	CJ1H-CPU65H	CJ1G-CPU45H	CJ1G-CPU44H	CJ1G-CPU43H	CJ1G-CPU42H
入出力點數	2,560		1,280		960	
程式容量 (參見附註)	120 k steps	60 k steps	60 k steps	30 k steps	20 k steps	10 k steps
資料記憶體	32 k words					
延伸資料記憶體	32 k words x 7 banks E0_00000 至 E6_32767	32 k words x 3 banks E0_00000 至 E2_32767	32 k words x 3 banks E0_00000 至 E2_32767	32 k words x 1 bank E0_00000 至 E0_32767		
消耗電流	0.82 A 於 5V DC		0.78 A 於 5V DC			

CJ1M CPU 模組

CPU	CJ1M-CPU23	CJ1M-CPU22	CJ1M-CPU13	CJ1M-CPU12
入出力點數	640	320	640	320
使用者程式記憶體 (參見附註)	20k steps	10k steps	20k steps	10k steps
資料記憶體	32k words			
延伸資料記憶體	不支援			
內藏 I/O	支援		不支援	
消耗電流	0.64 A 於 5V DC		0.58 A 於 5V DC	

CJ1 CPU 模組

CPU	CJ1G-CPU45	CJ1G-CPU44
入出力點數	1,280	
使用者程式記憶體 (參見附註)	60k steps	30 k steps
資料記憶體	32 k words	
延伸資料記憶體	32 k words x 3 banks E0_00000 至 E2_32767	32 k words x 1 bank E0_00000 至 E0_32767
消耗電流	0.91 A 於 5V DC	

附註 程式中的 Steps 與指令數是不相同的。例如，LD 與 OUT 均需要一個 Steps，而 MOV(021) 則需要三個 Steps。程式容量表示程式中所有指令的總數。

共通規格

項目	規格
控制方法	儲存的程式
I/O 控制方法	循環掃描與立即處理均可
規劃	階梯圖
CPU 處理模式	CJ1-H CPU 模組：一般模式，同步記憶體存取的並行處理模式，非同步記憶體存取的並行處理模式，或週邊服務優先模式 CJ1M CPU 模組：一般模式或週邊服務優先模式 CJ1 CPU 模組：一般模式或週邊服務優先模式
指令長度	每個指令 1 至 7 個 Steps
指令種類	約 400 個 (3- 位元功能碼)

項目	規格
執行時間	CJ1-H CPU 模組： 基本指令：0.02 μ s 以上 特殊指令：0.06 μ s 以上 CJ1M CPU 模組： 基本指令：0.10 μ s 以上 特殊指令：0.15 μ s 以上 CJ1 CPU 模組： 基本指令：0.08 μ s 以上 特殊指令：0.12 μ s 以上
整體時間	CJ1-H CPU 模組： 一般模式：0.3 ms 以上 平行處理：0.3 ms 以上 CJ1M CPU 模組：0.5 ms 以上 CJ1 CPU 模組：0.5 ms 以上
模組連接方式	無背板：模組彼此直接連接
安裝方式	DIN 軌道（不能用螺絲安裝）
最大連接模組數	<ul style="list-style-type: none"> • CJ1-H 與 CJ1 CPU 模組： 每個 CPU 或擴充 Rack：10 個模組，包括基本 I/O 模組，特殊 I/O 模組，與 CPU 匯流排模組。 每個 PLC 總數：在 CPU rack 上 10 個模組，在 3 個擴充 rack 上各有 10 個模組 = 總共 40 個模組 • CJ1M CPU 模組： 系統中共有 20 個模組，包括 CPU rack 上 10 個模組，一個擴充 rack 上 10 個模組。
最大擴充底架數	<ul style="list-style-type: none"> • CJ1-H 與 CJ1 CPU 模組： 最多 3 個 (CPU rack 上需要一個 I/O 控制模組，每個擴充 rack 上需要一個 I/O 介面模組。) • CJ1M CPU 模組： 最多 1 個 (CPU rack 上需要一個 I/O 控制模組，每個擴充 rack 上需要一個 I/O 介面模組。)
工作單數 (Number of tasks)	288 (循環工作區：32，中斷工作：256) 在 CJ1-H 或 CJ1M CPU 模組，中斷工作可以定義為循環工作，稱為 "額外循環工作"。包括這些，最多可以使用 288 個循環工作。 附註 每個循環都會執行循環工作，並受 TKON(820) 與 TKOF(821) 指令所控制。 附註 支援以下 2 種中斷工作。 支援以下 4 種中斷工作。 斷電中斷工作：1 以下 排程中斷工作：2 以下 I/O 中斷工作：32 以下 外部中斷工作：256 以下
中斷型式 (Interrupt Types)	排程中斷： 由 CPU 模組的內藏計時器在預定時間所產生的中斷。(參見附註) I/O 中斷： 由中斷輸入模組產生的中斷。 斷電中斷： 當 CPU 模組的電源變為 OFF 時所產生的中斷。 外部 I/O 中斷： 由特殊 I/O 模組或 CPU 匯流排模組所產生的中斷。 附註 CJ1-H 與 CJ1 CPU 模組：排程中斷時間間隔為 1 ms 至 9,999 ms 或 10 ms 至 99,990 ms，單位為 1 ms 或 10 ms。 CJ1M CPU 模組：除以上之外，排程中斷時間間隔也可以是 0.5 ms 至 999.9 ms，單位為 0.1 ms。
由多個工作單呼叫副程式	CJ1-H CPU 模組：支援 (稱為全域副程式)。 CJ1 CPU 模組：不支援

項目		規格												
CIO (核心 I/O) 區域	I/O 區域	1,280 : CIO 000000 至 CIO 007915 (從 CIO 0000 至 CIO 0079 共 80 個字組) 第一個字組的設定可由內定值(CIO 0000)更改為 CIO 0000 至 CIO 0999 間的值。出力點數位於基本 I/O 模組。												
	Link 區域 (Link Area)	3,200 (200 字組) : CIO 10000 至 CIO 119915 (字組 CIO 1000 至 CIO 1199) 位於 Controller Link 系統中的 Link 位元用於資料 Link。												
	CPU 匯流排模組區域	6,400 (400 字組) : CIO 150000 至 CIO 189915 (字組 CIO 1500 至 CIO 1899) CPU 匯流排模組位元儲存 CPU 匯流排模組的操作狀態。 (每模組 25 個字組, 最多 16 個模組)												
	特殊 I/O 模組區域	15,360 (960 字組) : CIO 200000 至 CIO 295915 (字組 CIO 2000 至 CIO 2959) 特殊 I/O 模組位元位於特殊 I/O 模組中。 (每模組 10 個字組, 最多 96 個模組) 附註 特殊 I/O 模組為屬於所謂 "特殊 I/O 模組" 的特殊族群。例如: CJ1W-AD081 類比輸入模組												
	串列 PLC Link 區域 (僅 CJ1M CPU 有此區域)	1,440 (90 字組) : CIO 310000 至 CIO 318915 (字組 CIO 3100 至 CIO 3189)												
	DeviceNet 區域	9,600 (600 字組) : CIO 320000 至 CIO 379915 (字組 CIO 3200 至 CIO 3799) 當主局以固定位置使用時, DeviceNet 子局區域分配於固定位置區域。 <table border="1" data-bbox="443 927 1011 1120"> <tr> <td>DeviceNet 固定位置設定 1</td> <td>輸出: CIO 3200 至 CIO 3263 輸入: CIO 3300 至 CIO 3363</td> </tr> <tr> <td>DeviceNet 固定位置設定 2</td> <td>輸出: CIO 3400 至 CIO 3463 輸入: CIO 3500 至 CIO 3563</td> </tr> <tr> <td>DeviceNet 固定位置設定 3</td> <td>輸出: CIO 3600 至 CIO 3663 輸入: CIO 3700 至 CIO 3763</td> </tr> </table> 以下的字組位於主功能, 即使 DeviceNet 模組用來做為從屬端也是一樣。 <table border="1" data-bbox="443 1189 1011 1382"> <tr> <td>DeviceNet 固定位置設定 1</td> <td>輸出: CIO 3370(子局至主局) 輸入: CIO 3270(主局至子局)</td> </tr> <tr> <td>DeviceNet 固定位置設定 2</td> <td>輸出: CIO 3570(子局至主局) 輸入: CIO 3470(主局至子局)</td> </tr> <tr> <td>DeviceNet 固定位置設定 3</td> <td>輸出: CIO 3770(子局至主局) 輸入: CIO 3670(主局至子局)</td> </tr> </table>		DeviceNet 固定位置設定 1	輸出: CIO 3200 至 CIO 3263 輸入: CIO 3300 至 CIO 3363	DeviceNet 固定位置設定 2	輸出: CIO 3400 至 CIO 3463 輸入: CIO 3500 至 CIO 3563	DeviceNet 固定位置設定 3	輸出: CIO 3600 至 CIO 3663 輸入: CIO 3700 至 CIO 3763	DeviceNet 固定位置設定 1	輸出: CIO 3370(子局至主局) 輸入: CIO 3270(主局至子局)	DeviceNet 固定位置設定 2	輸出: CIO 3570(子局至主局) 輸入: CIO 3470(主局至子局)	DeviceNet 固定位置設定 3
DeviceNet 固定位置設定 1	輸出: CIO 3200 至 CIO 3263 輸入: CIO 3300 至 CIO 3363													
DeviceNet 固定位置設定 2	輸出: CIO 3400 至 CIO 3463 輸入: CIO 3500 至 CIO 3563													
DeviceNet 固定位置設定 3	輸出: CIO 3600 至 CIO 3663 輸入: CIO 3700 至 CIO 3763													
DeviceNet 固定位置設定 1	輸出: CIO 3370(子局至主局) 輸入: CIO 3270(主局至子局)													
DeviceNet 固定位置設定 2	輸出: CIO 3570(子局至主局) 輸入: CIO 3470(主局至子局)													
DeviceNet 固定位置設定 3	輸出: CIO 3770(子局至主局) 輸入: CIO 3670(主局至子局)													
內部 I/O 區域	4,800 (300 字組) : CIO 120000 至 CIO 149915 (字組 CIO 1200 至 CIO 1499) 37,504 (2,344 字組) : CIO 380000 至 CIO 614315 (字組 CIO 3800 至 CIO 6143) 在 CIO 區域中的這些位元用來作為規劃中的工作位元, 以控制程式的執行。它們不能用來作為外部 I/O。													
工作區域 (Work Area)	8,192 位元 (512 字組) : W00000 至 W51115 (W000 至 W511) 僅控制此程式。(I/O 不可能由外部 I/O 端子過來。) 附註 當在規劃中使用工作位元時, 在使用其他區域的位元前應先使用工作區域的位元。													
保持區域 (Holding Area)	8,192 位元 (512 字組) : H00000 至 H51115(H000 至 H511) 保持位元用於控制程式的執行, 與在 PLC 變為 OFF 或操作模式改變時能夠維持它們的 ON/OFF 狀態。													
輔助區域 (Auxiliary Area)	唯讀: 7,168 位元 (448 字組) : A00000 至 A44715(字組 A000 至 A447) 讀 / 寫: 8,192 位元 (512 字組) : A44800 至 A95915(字組 A448 至 A959) 輔助位元位於特殊功能。													
暫存區域 (Temporary Area)	16 位元 (TR0 至 TR15) 暫存位元用於在程式分枝暫時 ON/OFF 執行條件。													
計時器區域 (Timer Area)	4,096 : T0000 至 T4095(僅作為計時器)													
計數器區域 (Counter Area)	4,096 : C0000 至 C4095(僅作為計數器)													

項目	規格
DM 區域	<p>32K 字組：D00000 至 D32767</p> <p>作為一般用途資料區域以讀出或寫入字組模組的資料 (16 位元)。DM 區域內的字組在 PLC 變為 OFF 或操作模式改變時會維持它們的狀態。</p> <p>內部特殊 I/O 模組 DM 區域：D20000 至 D29599 (100 字組 x 96 模組)</p> <p>用於設定特殊 I/O 模組的參數。</p> <p>CPU 匯流排模組 DM 區域：D30000 至 D31599 (100 字組 x 16 模組)</p> <p>用於設定 CPU 匯流排模組的參數。</p>
EM 區域 (僅 CJ1-H 與 CJ1 CPU 模組)	<p>每 bank 32K 字組，最多 3 banks：E0_00000 最多至 E2_32767 (依 CPU 模組型號而不同)</p> <p>作為一般用途資料區域以讀出或寫入字組模組的資料 (16 位元)。EM 區域內的字組在 PLC 變為 OFF 或操作模式改變時會維持它們的狀態。</p> <p>EM 區域分為若干 bank，其地址可用以下的方法設定。</p> <p>用 EMBC(281) 指令改變現在的 bank 與設定本 bank 的地址。</p> <p>設定 bank 數與直接設定地址。</p> <p>EM 資料可藉由指定第一 bank 的數量而儲存於檔案中。</p>
索引暫存器 (Index Registers)	<p>IR0 至 IR15</p> <p>儲存 PLC 記憶體位址供間接地址使用。索引暫存器可在每項工作中獨立地使用。每個暫存器都是 32 位元 (2 字組)。</p> <p>CJ1-H 與 CJ1M CPU 模組：在每項工作中可設定索引暫存器為獨立使用或是共用。</p> <p>CJ1 CPU 模組：在每項工作中獨立使用索引暫存器。</p>
工作旗標區域 (TaskFlag Area)	<p>32 (TK0000 至 TK0031)</p> <p>工作旗標為唯讀的旗標，其在相關的循環工作可執行時為 ON，當相關的工作為不可執行或待命中時為 OFF。</p>
追蹤記憶體 (Trace Memory)	<p>4,000 字組 (追蹤資料：31 位元，6 字組)</p>
檔案記憶體 (File Memory)	<p>記憶卡：可使用 OFF (MS-DOS 格式)。</p> <p>EM 檔案記憶體 (僅 CJ1-H 與 CJ1 CPU 模組)：部份的 EM 區域可以轉換到檔案記憶體 (MS-DOS 格式)。</p> <p>可以使用 OMRON 記憶卡。</p>

功能規格

項目	規格	
固定循環時間 (Constant cycle time)	1 至 32,000 ms (單位: 1 ms) 當以並行處理模式用於 CJ1-H CPU 模組時, 執行指令的循環時間為常數。	
循環時間監控 (Cycle time monitoring)	可能的 (若循環太長時, 模組會停止操作): 10 至 40,000 ms (單位: 10 ms) 當 CJ1-H CPU 模組使用並行處理模式時, 指令執行循環是受到監控的。若週邊服務循環時間超過 2 s (固定) 時, CPU 模組操作將會停止。	
I/O 更新	循環更新、立即更新、以 IORF(097) 更新。 IORF(097) 更新位於基本 I/O 模組與特殊 I/O 模組的位元。 加上 CJ1-H 與 CJ1M CPU 模組, CPU 匯流排模組 I/O REFRESH (DLNK(226)) 指令可用來更新 CIO 與 DM 區塊內位於 CPU 匯流排模組的位元。	
CPU 匯流排模組的特殊更新時間	Controller Link 與 SYSMAC LINK 模組的資料 Link、DeviceNet 模組的遠端 I/O、與 CPU 匯流排模組的其他特殊更新是在以下的時間進行的: CJ1 與 CJ1M CPU 模組: I/O 更新週期 CJ1-H CPU 模組: I/O 更新週期與當 CPU 匯流排模組 I/O REFRESH (DLNK(226)) 指令執行時。	
改變操作模式時 I/O 記憶體保持	決定於輔助區域內 IOM 保持位元的 ON/OFF 狀態。	
載入 OFF	輸出模組上所有的輸出在 CPU 模組操作於 RUN、MONITOR、或 PROGRAM 模式時都可以變為 OFF。	
計時器 / 計數器 PV 更新方法	CJ1-H 與 CJ1M CPU 模組: BCD 或 BIN (CX-P 軟體 3.0 版以上)。 CJ1 CPU 模組: 僅 BCD。	
輸入響應時間設定	可由基本 I/O 模組設定輸入的時間常數。時間常數的增加可以減少雜訊與顫動的影響, 而減少時間常數則可偵測輸入中較短的脈衝。	
送電時的模式設定	可能的 (若沒有連接程式書寫器, 則內定的 CPU 模組會啟動於 RUN 模式。)	
快閃記憶體 (僅 CJ1-H 與 CJ1M CPU 模組)	使用者程式與參數區域的資料 (例如 PLC 設定) 在快閃記憶體中永遠都會自動備份。(自動備份與回復)	
記憶卡功能	當電源變為 ON 時會自動由記憶卡中讀出程式 (自動開機)。	可能的
	程式在 PLC 操作期間置換。	可能的
	儲存於記憶卡的資料格式	使用者程式: 程式檔格式 PLC 設定與其他參數: 資料檔格式 I/O 記憶體: 資料檔格式 (BIN 格式)、純文字格式、或 CSV 格式
	支援讀 / 寫的記憶卡功能	使用者程式指令、規劃裝置 (包括 CX-P 軟體與程式書寫器)、Host Link 電腦、AR 區域控制位元、簡易備份操作
歸檔 (Filing)	記憶卡資料與 EM (延伸資料記憶體) 區域可當作檔案來處理。	
除錯 (Debugging)	當程式錯誤發生時, 控制設定 / 重置、監控、資料追蹤 (排程、每個循環或當指令執行時)、指令錯誤追蹤、儲存產生錯誤的位置。	
線上編輯 (Online editing)	當 CPU 模組在 MONITOR 或 PROGRAM 模式時, 使用者程式可以覆寫到程式區塊模組。此功能在區塊規劃區域不適用。透過 CX-P 軟體, 一次可以規劃好幾個程式。	
程式保護 (Program protection)	防止寫入保護: 以指撥開關進行設定。 防止複製保護: 用 CX-P 軟體或程式書寫器設定密碼。	
錯誤檢查 (Error check)	使用者定義錯誤 (即使用者可以定義嚴重錯誤與非嚴重錯誤) FPD(269) 指令可用來檢查每個程式區塊的執行時間與邏輯。 FAL 與 FALS 指令可與 CJ1-H 及 CJ1M CPU 模組共同使用以模擬錯誤狀態。	
錯誤記錄 (Error log)	錯誤記錄最多儲存 20 個錯誤。其各式包括錯誤碼、錯誤細節與錯誤發生的時間。 可以將 CJ1-H 或 CJ1M CPU 模組設定為使用者定義 FAL 錯誤不會存入錯誤記錄。	
串列通訊 (Serial communication)	內藏週邊埠: 規劃裝置 (包括程式書寫器) 連接、Host Link、NT Link。	
	內藏 RS-232C 埠: 規劃裝置 (不含程式書寫器) 連接、Host Link、無協定通訊、NT Link。	
	串列通訊模組 (另售): 協定巨集、Host Link、NT Link。	

項目	規格
時鐘 (Clock)	所有型號均提供。 正確性：環境溫度 55°C 月誤差 -3.5 min 至 +0.5 min 25°C -1.5 min 至 +1.5 min 0°C -3 min 至 +1 min 附註 用於記錄電源 ON 與發生錯誤的時間。
電源 OFF 檢測時間	10 至 25 ms (非固定)
電源 OFF 檢測延遲時間	0 至 10 ms (使用者定義, 內定值: 0 ms)
記憶保護	保持區域: 保持位元、資料記憶體與延伸資料記憶體的內容、及計數器完成旗標與現值的狀態。 附註 若輔助區域內的 IOM 保持位元變為 ON, 而 PLC 設定為當 PLC 的電源變為 ON 時要維持 IOM 保持位元狀態, 則 CIO 區域的內容、工作區域、部份的輔助區域、計時器完成旗標與 PV、索引暫存器、及資料暫存器將會保存最多 20 天。
送指令到上位電腦	FINS 指令可以藉由從 PLC 執行網路通訊指令而經由 Host Link 系統送到連接的電腦。
遙控程式與監控	Host Link 通訊可透過 Controller Link 系統或乙太網路而用於遙控規劃與遙控監控。
三階層通訊	Host Link 通訊可用於由最多兩階層網路上的裝置進行遙控規劃與遙控監控 (Controller Link 網路、乙太網路、或其他網路)。
儲存註解於 CPU 模組	I/O 註解可儲存於記憶卡或 EM 檔案記憶體 (僅 CJ1-H 或 CJ1M) 內的 CPU 模組。
程式檢查	程式檢查是在程式開始操作前檢查類似沒有 END 指令的項目與指令錯誤。 CX-P 軟體也可以用來檢查程式。
控制輸出訊號	RUN 輸出: 當 CPU 模組操作時內部接點會變為 ON (閉合) (CJ1W-PA205R)。
電池壽命	參見 12-2 Replacing User-serviceable Parts. CJ1-H 與 CJ1 CPU 模組的電池組: CPM2A-BAT01 CJ1M CPU 模組電池組: CJ1W-BAT01
自我診斷	CPU 錯誤 (Watchdog timer)、I/O 匯流排錯誤、記憶體錯誤、與電池錯誤。
其他功能	儲存電源中斷的次數。(儲存於 A514)

僅 CJ1M CPU 模組所提供的功能

項目		規格	
內藏 I/O	內藏輸入	一般用途輸入	當由輸入模組輸入時，一般輸入訊號會依 I/O 更新時序處理，並反射到 I/O 記憶體。
		中斷輸入	中斷輸入 (直接模式)： 中斷工作數 140 至 143 是由 CIO 2960 的位元 00 至 03 的上升緣或下降緣開始的。 響應時間：0.3 ms 中斷輸入 (計數器模式)： 中斷工作數 140 至 143 是由 CIO 2960 的位元 00 至 03 的計數器遞增或遞減時開始的。 響應頻率：1 kHz
		高速計數器	輸入至內藏輸入端子的訊號會被計數。高速計數器可使用的輸入有以下四種： 相位差脈衝輸入： 30 kHz (開集極) 與 50 kHz (線驅動器) 脈衝 + 方向輸入： 60 kHz (開集極) 與 100 kHz (線驅動器) 上 / 下脈衝輸入： 60 kHz (開集極) 與 100 kHz (線驅動器) 遞增脈衝輸入： 60 kHz (開集極) 與 100 kHz (線驅動器) 中斷工作可開始於高速計數器的比較條件符合之時。高速計數器的 PV 有兩種比較的方法： 1) 目標值比較 2) 範圍比較 也可以禁止對輸入訊號計數 (閉道功能)。
		快速響應輸入	當作輸入訊號讀入較循環時間短的脈衝訊號 (最小脈衝寬度：50 μs)。
	內藏輸出	一般用途輸出	當作輸出模組，I/O 記憶體的內容會依據 I/O 更新時序輸出。
		脈衝輸出	固定功率脈衝訊號 (功率：50%) 由內藏輸出端子輸出。速度控制 (連續輸出指定頻率的脈衝) 與定位 (輸出特定個數的指定頻率脈衝後停止) 是可能的。
		可變負載脈衝輸出 (PWM(891)輸出)	以設定功率 (一脈衝時間之 ON 時間與 OFF 時間的比值) 執行脈衝輸出。
	建立原點	原點搜尋	建立一基於原點搜尋參數中指定樣式的脈衝輸出之機械原點。
		原點回復	由任意位置回到原點。
	串列 PLC Link		使用內藏 RS-232C 埠來在 CPU 模組間交換資料而不需要程式。PT 為 NT Link 而設定 (1:N 模式)
排程中斷	排程中斷單位為 0.1 ms	以最小的 0.5 ms 時間間隔在 0.1 ms 的精度下開始排程中斷工作 (在 PLC 設定中進行設定)。	
	以 MSKS 指令開始重置	當執行 MSKS 時，重置開始內部計時器並把到首次中斷的時間標準化。	
	由 MSKS 指令讀出內部計時器 PV	當執行 MSKS 時，讀出由排程中斷或前一個排程中斷的開始流逝的時間。	

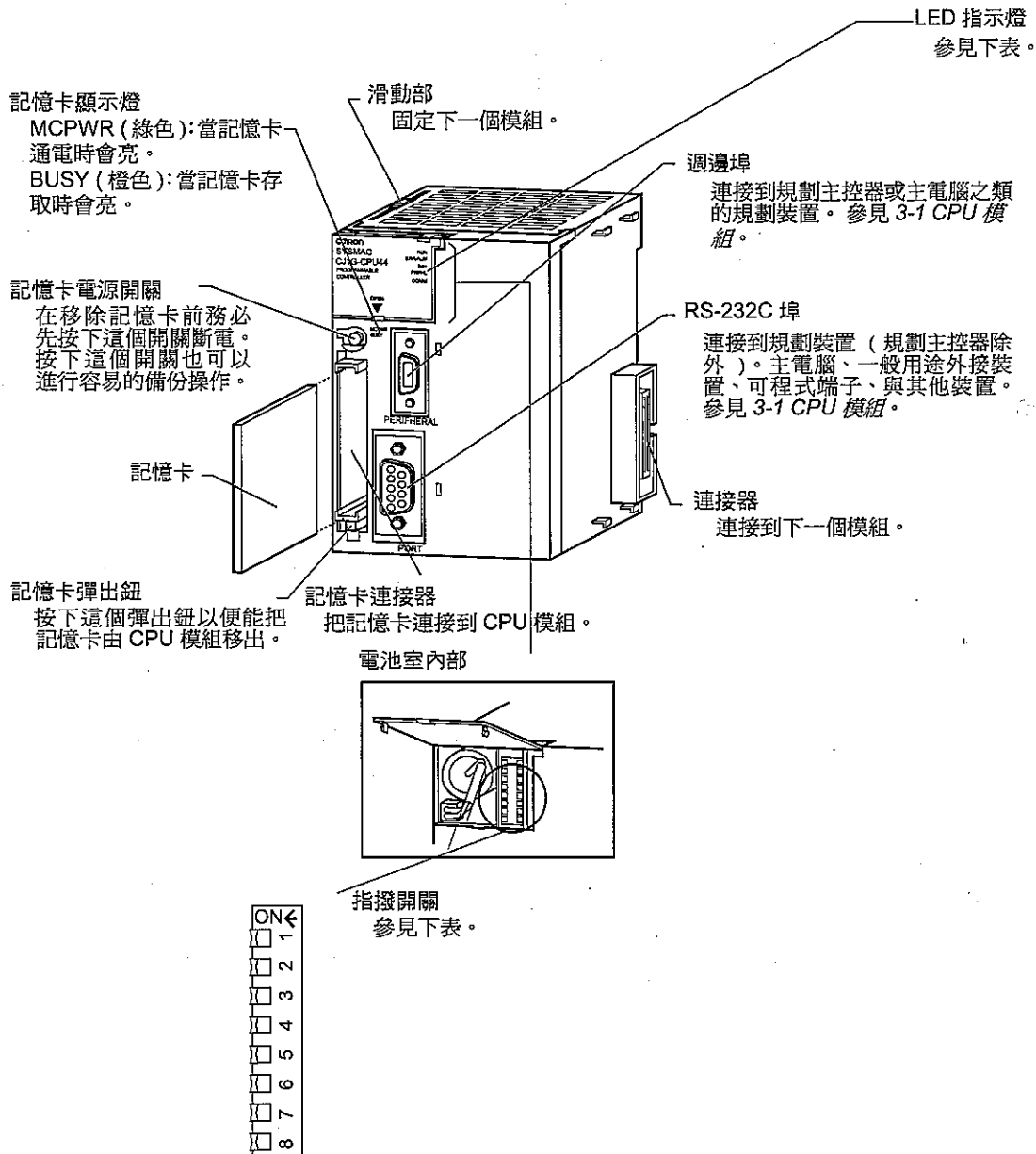
2-1-2 一般規格

項目	規格		
電源供應器	CJ1W-PA205R	CJ1W-PA202	CJ1W-PD025
電源電壓	100 至 240 V AC, 50/60 Hz		24 V DC
操作電壓與頻率範圍	85 至 264 V AC, 47 至 63 Hz		19.2 至 28.8 V DC
消耗功率	100 VA 以下	50 VA 以下	50 W 以下
突入電流 (參見附註 3.)	於 100 至 120 V AC : 在室溫下冷起動最多 15 A/8 ms 於 200 至 240 V AC : 在室溫下冷起動最多 30 A/8 ms	於 100 至 120 V AC : 在室溫下冷起動最多 20 A/8 ms 於 200 至 240 V AC : 在室溫下冷起動最多 40 A/8 ms	於 24 V DC : 在室溫下冷起動最多 30 A/2 ms
輸出容量	5.0 A, 5 V DC (包括供應給 CPU 模組)	2.8 A, 5 V DC (包括供應給 CPU 模組)	5.0 A, 5 V DC (包括供應給 CPU 模組)
	0.8 A, 24 V DC 總共: 25 W 以下	0.4 A, 24 V DC 總共: 14 W 以下	0.8 A, 24 V DC 總共: 25 W 以下
輸出端子 (服務提供)	未提供		
RUN 輸出 (參見附註 2.)	接點架構: SPST-NO 開關容量: 250 V AC, 2 A (電阻性負載) 120 V AC, 0.5 A (電感性負載), 24 V DC, 2 A (電阻性負載) 24 V DC, 2 A (電感性負載)	未提供	
絕緣電阻	20 M Ω 以上 (於 500 V DC) 在 AC 外部與 GR 端子間 (參見附註 1.)		20 MW 以上 (於 500 V DC) 在 DC 外部與 GR 端子間 (參見附註 1.)
介電強度	2,300 V AC 50/60 Hz 1 分鐘在 AC 外部與 GR 端子間 (參見附註 1.) 漏電流: 10 mA 以下		
	1,000 V AC 50/60 Hz 1 分鐘在 DC 外部與 GR 端子間 (參見附註 1.) 漏電流: 10 mA 以下		
耐雜訊	2 kV 在電源線上 (符合 IEC61000-4-4)		
耐震性	10 至 57 Hz, 0.075-mm 振幅, 57 至 150 Hz, 加速度: 9.8 m/s ² in X, Y, 與 Z 方向 80 分鐘 (時間係數: 8 分鐘 x 係數因子 10 = 總時間 80 以上) (依據 JIS C0040)		
耐衝擊性	在 X, Y, 與 Z 方向 147 m/s ² 各 3 次 (繼電器輸出單位: 100 m/s ²) (依據 JIS C0041)		
操作環境溫度	0 至 55°C		
操作環境濕度	10% 至 90% (無凝結)		
空氣	必須沒有腐蝕性氣體		
存放環境溫度	-20 至 70°C (不含電池)		
接地電阻	小於 100 Ω		
外殼	安裝於面板上		
重量	所有的型號均在 5 kg 以下		
CPU 架尺寸規格	90.7 至 466.7 x 90 x 65 mm (W x H x D) (不包含纜線) 附註: W = a + b + 20 x n + 31 x m + 14.7 a: 電源供應器: PA205R = 80°F PA202 = 45°F PD025 = 60 b: CPU 模組: CJ1-H 或 CJ1 = 62; CJ1M-CPU1 □ = 31°F CJ1M-CPU2 □ = 49 總寬度: W = 156.7 + n x 20 + m x 31, 其中 n 為 32 點 I/O 模組或 I/O 控制模組的數量, 而 m 為其他模組的個數。		
安全標準認證	符合 cULus 與 EC 規範。		

- 附註
1. 當測試絕緣與介電強度時從 GR 端子上拆除電源供應器的 LG 端子。在 LG 端子與 GR 端子連接下測試絕緣與介電強度將會使 CPU 內部電路受損。
 2. 僅在安裝於 CPU 架時支援。
 3. 湧入電流是定義於室溫下冷起動的狀況。湧入控制電路使用一具低溫電流控制特性的電熱調節元件。若環境溫度偏高或 PLC 為熱起動, 電熱調節器將不夠冷, 表中的湧入電流可能會達到給定值的兩倍。在選擇外部電路的保險絲或繼電器時, 請保留足夠的斷離餘裕。

2-2 CPU 模組組件與功能

2-2-1 CPU 模組組件



ON	1
	2
	3
	4
	5
	6
	7
	8

顯示燈

以下的表格描述 CPU 模組面板上的 LED 顯示燈之狀況。

顯示燈	意義
RUN(綠色)	當 PLC 在 MONITOR 或 RUN 模式正常動作時會亮。
ERR/ALM(紅色)	當 CPU 模組發生非嚴重錯誤時 ERR/ALM 燈會閃爍。若發生非嚴重錯誤，CPU 模組仍將繼續操作。 當 CPU 模組發生嚴重錯誤或硬體發生錯誤時 ERR/ALM 燈會亮。若發生非嚴重錯誤，CPU 模組仍將繼續操作。
INH(橙色)	當輸出 OFF 位元 (A50015) 變為 ON 時會亮。若輸出 OFF 位元變為 ON，則所有輸出模組的輸出都會變為 OFF。
PRPHL(橙色)	當 CPU 模組與週邊埠進行通訊時會閃爍。
BKUP(橙色；僅 CJ1-H 與 CJ1M CPU 模組)	當資料由 RAM 備份至快閃記憶體時會亮。 當此燈亮時不要把 CPU 模組變為 OFF。
COMM(橙色)	當 CPU 模組與 RS-232C 埠進行通訊時會閃爍。
MCPWR(綠色)	當記憶卡供電時會亮。
BUSY	當記憶卡存取時會亮。

指撥開關

CJ 系列 CPU 模組有一個 8 針腳的指撥開關可用來設定 CPU 模組的基本操作參數。指撥開關位於電池蓋的下方。指撥開關針腳的設定如下表所示：

接腳	設定	功能
1	ON	使用者程式記憶體不可寫入。
	OFF	使用者程式記憶體可以寫入。
2	ON	當電源變為 ON 時使用者程式自動傳送。
	OFF	當電源變為 ON 時使用者程式不會自動傳送。
3	ON	未使用
4	ON	在 PLC 設定中使用週邊埠參數。
	OFF	自動檢測程式書寫器或 CX-P 軟體在週邊埠的參數。
5	ON	自動檢測 CX-P 軟體 RS-232C 埠上的參數。
	OFF	在 PLC 設定中使用 RS-232C 埠參數。
6	ON	使用者定義接腳。 使用者指撥開關接腳旗標 (A39512) 變為 OFF。
	OFF	使用者定義接腳。 使用者指撥開關接腳旗標 (A39512) 變為 ON。
7	ON	簡易備份：讀出 / 寫入記憶卡。
	OFF	簡易備份：驗證記憶卡的內容。
8	OFF	永遠為 OFF

2-3 CPU 匯流排模組設定區域容量

大部份 CPU 匯流排模組的設定是儲存於 CPU 模組內的 CPU 匯流排模組設定區域。CPU 匯流排模組配置需要的工作數以便能由這個區域設定。

CPU 匯流排模組設定區域的容量有 10,752 bytes (10k bytes) 的限制。系統在設計上必需讓所有 CPU 匯流排模組所使用 CPU 匯流排模組設定區域中使用的字組數不會超過這個容量。若模組的組合不當，容量將會超出，且各模組將會以預設值來操作，甚至不會動作。

以下的表格顯示在 CPU 匯流排模組設定區域中，各模組所需用到的位元組數。使用 0 的模組不能使用 CPU 匯流排模組設定區域。

區分	名稱	型號	容量 (位元組)
CPU 匯流排模組	Controller Link 模組	CJ1W-CLK21 型	512
	串列通訊模組	CJ1W-SCU41 型 CJ1W-SCU21 型	0
	乙太網路模組	CJ1W-ETN11 型	412
	DeviceNet 模組	CJ1W-DRM21 型	0

2-4 I/O 表設定表列

以編輯 CX-P 軟體 I/O 表的設定內容如下所示。

2-4-1 CJ 系列基本 I/O 模組

模組名稱	型號	模組型式	定位模組數	定位字組數	
				輸入	輸出
DC 輸入模組	CJ1W-ID211 型	16 點輸入模組	---	---	---
	CJ1W-ID231 型	32 點輸入模組	---	---	---
	CJ1W-ID232 型	32 點輸入模組	---	---	---
	CJ1W-ID261 型	64 點輸入模組	---	---	---
	CJ1W-ID262 型	64 點輸入模組	---	---	---
AC 輸入模組	CJ1W-IA111 型	16 點輸出模組	---	---	---
	CJ1W-IA201 型	16 點輸出模組	---	---	---
中斷輸入模組	CJ1W-INT01 型	16 點輸出模組 (16 個中斷點)	---	---	---
快速響應輸入模組	CJ1W-IDP01 型	16 點輸入模組	---	---	---
繼電器輸出模組 (Relay Output)	CJ1W-OC201 型	16 點輸出模組	---	---	---
	CJ1W-OC211 型	16 點輸出模組	---	---	---
Triac 輸出模組	CJ1W-OA201 型	16 點輸出模組	---	---	---
電晶體輸出模組 (NPN 輸出)	CJ1W-OD201 型	16 點輸出模組	---	---	---
	CJ1W-OD211 型	16 點輸出模組	---	---	---
	CJ1W-OD231 型	32 點輸出模組	---	---	---
	CJ1W-OD233 型	32 點輸出模組	---	---	---
	CJ1W-OD261 型	64 點輸出模組	---	---	---
	CJ1W-OD263 型	64 點輸出模組	---	---	---
電晶體輸出模組 (PNP 輸出)	CJ1W-OD202 型	16 點輸出模組	---	---	---
	CJ1W-OD212 型	16 點輸出模組	---	---	---
	CJ1W-OD232 型	32 點輸出模組	---	---	---

附註 若所選的模組不正確，將會產生 I/O 表設定錯誤。

2-4-2 CJ 系列特殊 I/O 模組

模組名稱	型號	模組型式	定位模組數	定位字組數	
				輸入	輸出
類比輸入模組	CJ1W-AD041 型	SIOU (特殊 I/O 模組)	1	9	1
	CJ1W-AD081 (-V1) 型		1	9	1
類比輸出模組	CJ1W-DA021 型		1	1	9
	CJ1W-DA041 型		1	1	9
溫度控制模組	CJ1W-TC001 型		2	14	6
	CJ1W-TC002 型		2	14	6
	CJ1W-TC003 型		2	14	6
	CJ1W-TC004 型		2	14	6
	CJ1W-TC101 型		2	14	6
	CJ1W-TC102 型		2	14	6
	CJ1W-TC103 型		2	14	6
	CJ1W-TC104 型		2	14	6
定位控制模組	CJ1W-NC113 型		1	3	2
	CJ1W-NC213 型		1	6	4
	CJ1W-NC413 型		2	12	8
	CJ1W-NC133 型		1	3	2
	CJ1W-NC233 型	1	6	4	
	CJ1W-NC433 型	2	12	8	
高速計數器模組	CJ1W-CT021 型	4	26	14	
CompoBus/S 主要模組	CJ1W-SRM21 型	1	6	4	
		2	12	8	

附註 若所選的模組、輸入字組數、或輸出字組數不正確，將會產生特殊 I/O 模組設定錯誤。

2-4-3 CJ 系列 CPU 匯流排模組

模組名稱	型號	模組型式	定位模組數	定位字組數	
				輸入	輸出
Controller Link 模組	CJ1W-CLK21 型	Controller Link 模組	---	---	---
串列通訊模組	CJ1W-SCU41 型 CJ1W-SCU21 型	串列通訊模組	---	---	---
乙太網路模組	CJ1W-ETN11 型	乙太網路模組	---	---	---
DeviceNet 模組	CJ1W-DRM21 型	無法註記	---	---	---

附註 DeviceNet 模組不是僅相容於 2.0 之前的 CX-P 軟體版本，所以它無法在 I/O 表中註記。請線上建立 I/O 表。

第 3 章 型號說明、功能、與尺寸規格

本章說明多種不同模組的組件名稱與它們的功能。各模組的尺寸規格也有提及。

3-1	CPU 模組.....	22
3-1-1	型號.....	22
3-1-2	組件.....	23
3-2	RS-232C 埠規格.....	26
3-3	電源供應器模組.....	28
3-3-1	電源供應器模組型號.....	28
3-3-2	組件與開關設定.....	28
3-4	CJ 系列基本 I/O 模組.....	29
3-4-1	CJ 系列基本 I/O 模組附端子區塊.....	29
3-4-2	CJ 系列 32/64 點基本 I/O 模組附連接器.....	31

3-1 CPU 模組

3-1-1 型號

CJ1-H CPU 模組

I/O 點	擴充架	程式容量	資料記憶體 (DM + EM)	LD 指令處理時間	型號	重量
2,560	3 以下	120 k steps	256 k words	0.02 μ s	CJ1H-CPU66H	200 g 以下
		60 k steps	128 k words		CJ1H-CPU65H	
		60 k steps	128 k words	0.04 μ s	CJ1G-CPU45H	190 g 以下
1,280	3 以下	30 k steps	64 k words		CJ1G-CPU44H	
960	2 以下	20 k steps	64 k words		CJ1G-CPU43H	
		10 k steps	64 k words		CJ1G-CPU42H	

CJ1M CPU 模組

I/O 點	擴充架	程式容量	資料記憶體 (DM)	LD 指令處理時間	脈衝 I/O	型號	重量
640	1	20 k steps	32 k words	0.1 μ s	是	CJ1M-CPU23	170 g 以下
320	無	10 k steps				CJ1M-CPU22	
640	1	20 k steps			否	CJ1M-CPU13	120 g 以下
320	無	10 k steps				CJ1M-CPU12	

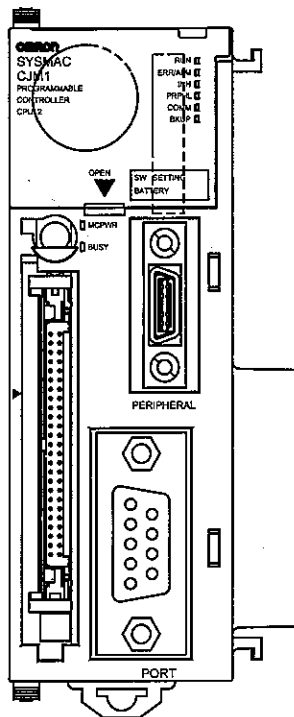
CJ1 CPU 模組

I/O 點	擴充架	程式容量	資料記憶體 (DM + EM)	LD 指令處理時間	型號	重量
1,280	3 以下	60 k steps	128 k words	0.08 μ s	CJ1G-CPU45	200 g 以下
		30 k steps	64 k words		CJ1G-CPU44	

3-1-2 組件

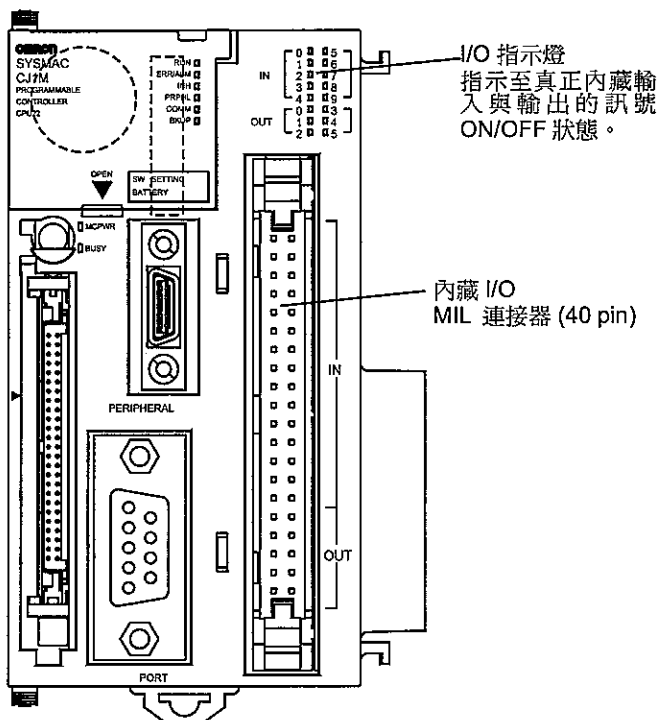
CJ1M CPU 模組

無內藏 I/O 機能機種 (CJ1M-CPU1 □)



附註 型號說明及功能與先前的 CJ1 及 CJ1-H CPU 模組相同。

有內藏 I/O 機能機種 (CJ1M-CPU2 □)

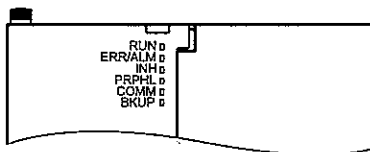


附註 上圖所示以外的型號說明及功能與 CJ1 與 CJ1-H CPU 模組相同。

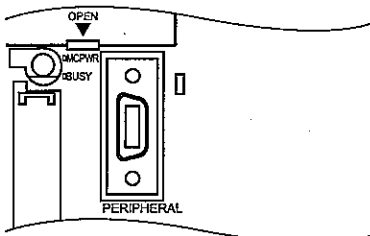
指示燈

下表格描述位於 CPU 模組面板的 LED 指示燈。

指示燈	顏色	狀態	意義
RUN	綠色	ON	PLC 在 MONITOR 或 RUN 模式正常動作。
		閃爍	系統下載模式錯誤或指撥開關設定錯誤。
		OFF	PLC 在 PROGRAM 模式已經停止動作，或已因嚴重錯誤而停止動作，或正從系統下載資料。
ERR/ALM	紅色	ON	發生嚴重錯誤（包括 FALS 指令執行），或發生硬體錯誤（看門狗計時器錯誤）。CPU 模組將會停止動作，所有輸出模組的輸出將會變為 OFF。
		閃爍	發生非嚴重錯誤（包括 FAL 指令執行）CPU 模組將會繼續動作。
		OFF	CPU 模組正常動作中。
INH	橙色	ON	輸出 OFF 位元（A50015）已經變為 ON。所有輸出模組的輸出將會變為 OFF。
		OFF	輸出 OFF 位元（A50015）已經變為 OFF。
PRPHL	橙色	閃爍	CPU 模組正經由經由週邊埠通訊（發送或接收）。
		OFF	CPU 模組不是經由週邊埠通訊。
COMM	橙色	閃爍	CPU 模組正經由 RS-232C 埠通訊（發送或接收）。
		OFF	CPU 模組不是經由 RS-232C 埠通訊。
BKUP (僅 CJ1-H CPU 模組)	橙色	ON	使用者程式與參數區域資料正在備份到 CPU 模組的快閃記憶體內或正從快閃記憶體載出。 附註 當指示燈亮時不要關閉 PLC 的電源。
		OFF	資料沒有寫入快閃記憶體。



指示燈	顏色	狀態	意義
MCPWR	綠色	ON	有供電給記憶卡。
		閃爍	閃爍一次：簡易備份讀出、寫入或驗證正常 閃爍五次：簡易備份寫入誤動作 閃爍三次：簡易備份寫入警告 連續閃爍：簡易備份讀出或驗證誤動作
		OFF	沒有供電給記憶卡。
BUSY	橙色	閃爍	記憶卡存取中。
		OFF	記憶卡非存取中。



指撥開關

CJ 系列 CPU 模組有一個 8 支接腳的指撥開關可用來設定 CPU 模組的基本操作參數。這個指撥開關位於電池室的蓋子下方。指撥開關接腳的設定如下表所述。

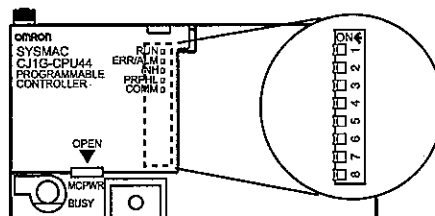
接腳編號	設定	功能	使用	預設值
1	ON	使用者程式記憶體寫入禁止 (參見附註)	用來防止程式不小心被規劃裝置 (包括程式書寫器) 所覆寫。	OFF
	OFF	使用者程式記憶體允許寫入		
2	ON	當打開電源時使用者程式自動由記憶體傳送。	用來把程式存入記憶卡以切換操作、或在送電時自動傳送程式 (記憶卡 ROM 操作)。 附註 當接腳 7 為 ON 而接腳 8 為 OFF 時, 由記憶卡簡易備份讀取會給定優先序, 所以即使接腳 2 為 ON, 當打開電源時使用者程式也不會自動由記憶卡傳送。	OFF
	OFF	當打開電源時使用者程式不會自動由記憶體傳送。		
3	---	未使用	---	OFF
4	ON	使用 PLC 內設定的週邊埠通訊參數。	變為 ON 以使用程式書寫器或 CX-P 軟體以外的週邊埠 (僅週邊匯流排)。	OFF
	OFF	使用以程式書寫器或 CX-P 軟體 (僅週邊匯流排) 設定的週邊埠通訊參數。		
5	ON	使用以 CX-P 軟體 (僅週邊匯流排) 設定的 RS-232C 埠通訊參數。	變為 ON 以使用規劃裝置的 RS-232C 埠。	OFF
	OFF	使用 PLC 內設定的 RS-232C 埠通訊參數。		
6	ON	使用者定義接腳, 關閉使用者指撥開關接腳旗標 (A39512)。	設定接腳 6 為 ON 或 OFF 並在程式中使用 A39512 以便能不用 I/O 模組建立使用者定義條件。	OFF
	OFF	使用者定義接腳, 開啓使用者指撥開關接腳旗標 (A39512)。		
7	ON	由 CPU 模組寫入記憶卡。	按住記憶卡電源開關 3 秒鐘。	OFF
		由記憶卡載出至 CPU 模組。	打開 PLC 電源以便能由記憶卡讀出至 CPU 模組。 這項操作給定的優先序高過當電源為 ON 時的自動傳送 (接腳 2 為 ON)。	
	OFF	驗證記憶卡的內容。	按住記憶卡電源開關 3 秒鐘。	
8	OFF	永遠為 OFF。		OFF

附註 1. 以下的資料在接腳 1 為 ON 時不能覆寫：

- 所有使用者程式的部份 (所有工作單內的程式)
- 所有參數區域內的資料 (例如 PLC 設定與 I/O 表)

當接腳 1 為 ON 時, 在由規劃裝置進行記憶體清除操作時, 使用者程式與參數區域將不會清除。

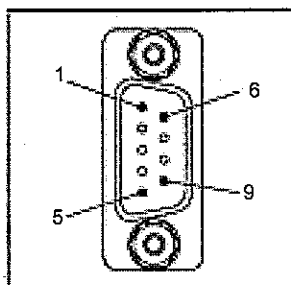
2. 在使用指撥開關接腳 7 把資料備份至記憶卡後, CPU 模組除了 PROGRAM 模式之外將不會進入任何模式。要進入 RUN 或 MONITOR 模式, 請關閉電源後, 將接腳 7 設定為 OFF, 然後重新起動 PLC。這將使操作模式回復正常。



附註 CJ 系列 CPU 模組的顯示語言不是在指撥開關上設定, 而是要用程式書寫器的按鍵設定。

3-2 RS-232C 埠規格

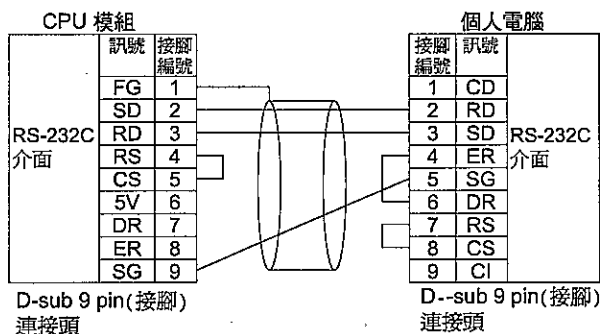
連接器接腳排列



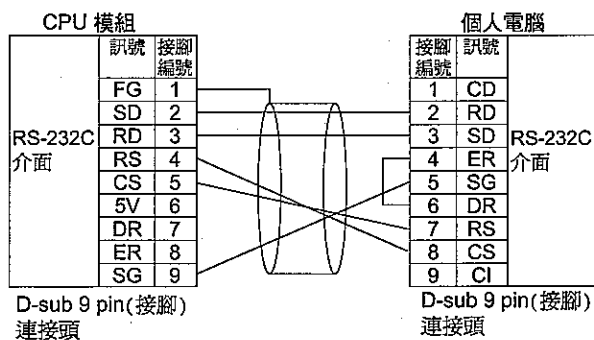
接腳編號	訊號	名稱	說明
1	FG	保護接地	—
2	SD (TXD)	發送資料	輸出
3	RD (RXD)	接收資料	輸入
4	RS (RTS)	請求發送	輸出
5	CS (CTS)	清除發送	輸入
6	5 V	電源	—
7	DR (DSR)	資料設定妥當	輸入
8	ER (DTR)	資料端子妥當	輸出
9	SG (0 V)	訊號接地	—
連接器罩	FG	保護接地	—

CJ 系列 CPU 模組與個人電腦間的連接

以下的連接採 Host Link 串列通訊模式。



以下的連接採週邊匯流排串列通訊模式。



可用的連接器

CPU 模組連接器

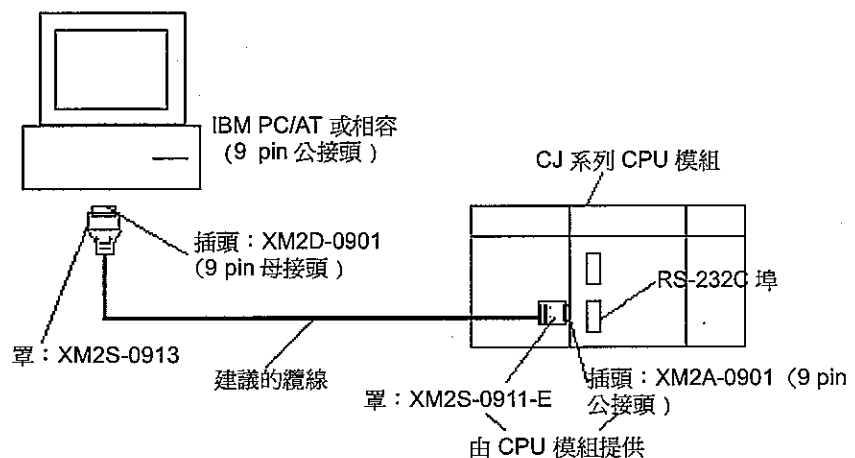
項目	型號	規格
插頭	XM2A-0901	9 pin(接腳)公接頭
罩	XM2S-0911-E	9 pin(接腳)、毫米螺絲、抗靜電

一起使用 (CPU 模組
附有各一個)

個人電腦連接器

項目	型號	規格
插頭	XM2D-0901	9 pin(接腳)母接頭
罩	XM2S-0913	9 pin(接腳)、英吋螺絲

一起使用



附註 儘可能使用 OMRON 所提供之特殊纜線。若纜線為自製，請確認接線之正確性。若使用一般用途（例如電腦至數據機）之纜線或纜線之接線不正確，則可能會造成外部裝置與 CPU 模組損壞。

建議的纜線

藤倉公司: UL2464 AWG28 × 5P IFS-RVV-SB (UL 產品)
AWG 28 × 5P IFVV-SB (非 UL 產品)

日立纜線公司: UL2464-SB(MA) 5P × 28AWG (7/0.127) (UL 產品)
CO-MA-VV-SB 5P × 28AWG (7/0.127) (非 UL 產品)

RS-232C 埠規格

項目	規格
通訊方法	半雙工
同步化	起動 - 停止
速率	0.3/0.6/1.2/2.4/4.8/9.6/19.2/38.4/57.6/115.2 kbps (參見附註)
傳輸距離	15 m 以下
介面	EIA RS-232C
協定	Host Link, NT Link, 1 : N, 無協定 (No-protocol), 或週邊匯流排

附註 RS-232C 的速率只有到 19.2 kbps。CJ 系列支援由 38.4 kbps 至 115.2 kbps 的串列通訊，不過某些電腦並不支援這個速度。必要時請降低速率。

協定 PLC 設定與指撥開關設定

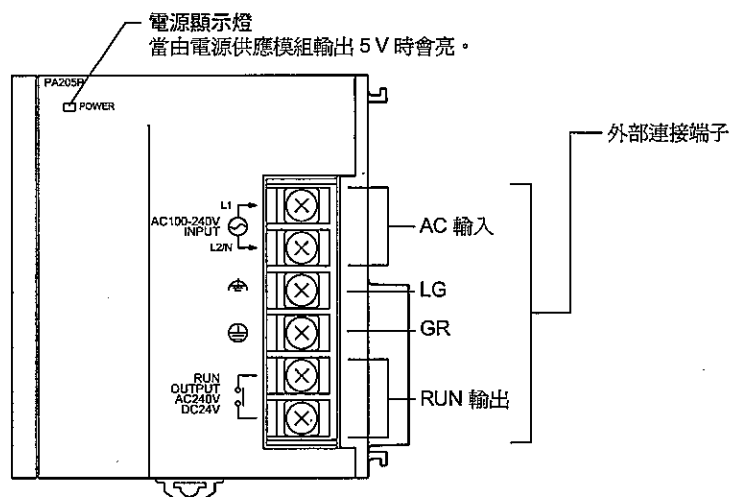
接腳編號 5	RS-232C 埠設定 (PLC 設定內)				
	預設值： 0 (Hex)	NT Link： 2 (Hex)	無協定： 3 (Hex)	週邊匯流排： 4 (Hex)	Host Link： 5 (Hex)
OFF	主電腦 (Host Link)	PT (NT Link)	汎用外部裝置 (無協定)	CX-P 軟體 (週邊匯流排)	主電腦或 CX-P 軟體 (Host Link)
ON	CX-P 軟體 (不是程式書寫器) 透過週邊匯流排連接。(規劃裝置的通訊參數為自動檢測。)				

3-3 電源供應器模組

3-3-1 電源供應器模組型號

電源電壓	輸出	電源輸出端子	RUN 輸出	型號	重量
100 至 240 V AC (容許值：85 至 264 V AC) 50/60 Hz (容許值：47 至 63 Hz)	5 A 於 5V DC 0.8 A 於 24 V DC 總共：25 W	否	是	CJ1W-PA205R	350 g 以下
	2.8 A 於 5V DC 0.4 A 於 24 V DC 總共：14 W	否	否	CJ1W-PA202	200 g 以下
24 V DC (容許值：19.2 至 28.8 V DC)	5 A 於 5V DC 0.8 A 於 24 V DC 總共：25 W	否	否	CJ1W-PD025	300 g 以下

3-3-2 組件與開關設定



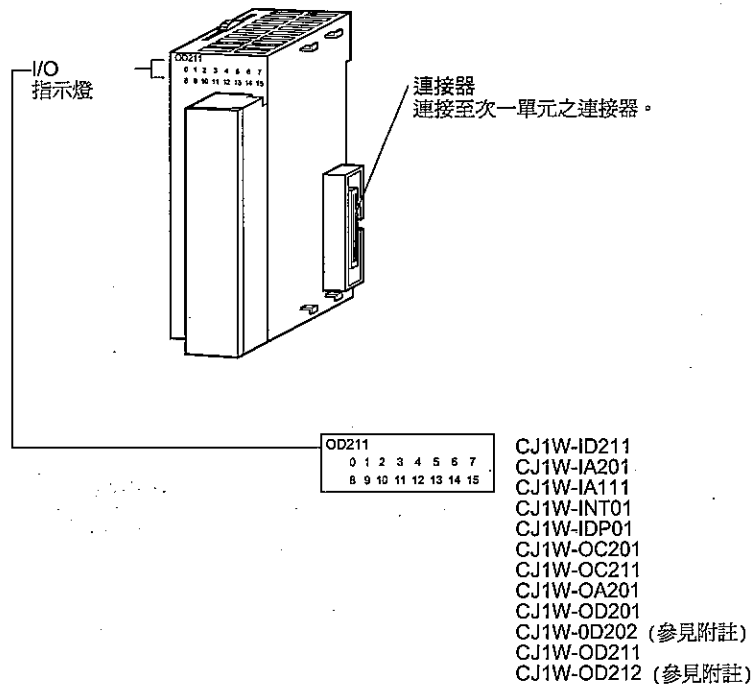
- AC 輸入 提供 100 至 240 V AC (容許值：85 至 264 V AC)。(不需選擇電壓)
- DC 輸入 提供 24 V DC (容許值：19.2 至 28.8 V DC)。
- LG 接地至 100 Ω 以下的電阻以增加雜訊容忍度與避免電擊。
- GR 接地至 100 Ω 以下的電阻以避免電擊。
- RUN 輸出 (僅 CJ1W-PA205R) 當 CPU 模組動作 (RUN 或 MONITOR 模式) 時，內部接點變為 ON。電源供應器模組必須在 CPU 架中以使用這個輸出。

3-4 CJ 系列基本 I/O 模組

3-4-1 CJ 系列基本 I/O 模組附端子區塊

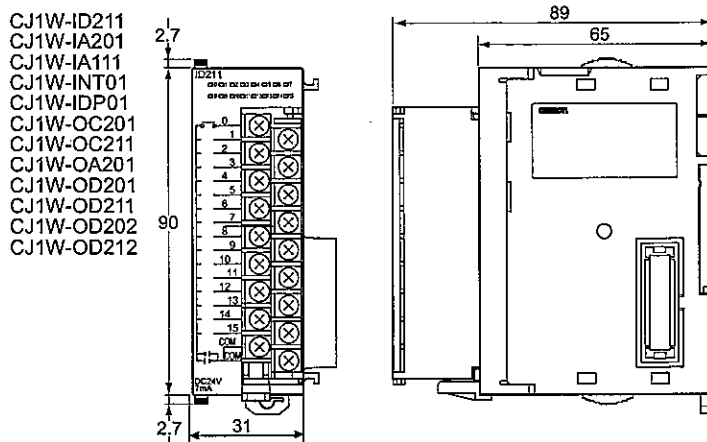
區分	名稱	規格	配置位元數	型號	頁碼
附端子區塊之基本輸入模組	DC 輸入模組	24 V DC	16	CJ1W-ID211	410
	AC 輸入模組	200 至 240 V DC	8	CJ1W-IA201	418
		100 至 120 V DC	16	CJ1W-IA111	419
	快速回應模組	24 V DC	16	CJ1W-IDP01	421
	中斷輸入模組	24 V DC	16	CJ1W-INT01	420
附端子區塊之基本輸出模組	繼電器輸出模組	250 V AC/24 V DC, 2 A: 8 獨立接點	8	CJ1W-OC201	422
		250 V AC/24 V DC, 2 A: 16 輸出	16	CJ1W-OC211	423
	Triac 輸出模組	250 V AC, 0.5 A	8	CJ1W-OA201	424
	電晶體輸出模組 NPN 輸出 (sinking outputs)	12 至 24 V DC, 2.0 A	8	CJ1W-OD201	425
		12 至 24 V DC, 0.5 A	16	CJ1W-OD211	426
	電晶體輸出模組 PNP 輸出 (sourcing outputs)	24 V DC, 2 A, 8 輸出, 負載短路保護與斷線偵測	8	CJ1W-OD202	433
		24 V DC, 0.5 A, 16 輸出, 負載短路保護	16	CJ1W-OD212	434

附 18 點端子區塊模組之各部名稱



附註 CJ1W-OD202 與 CJ1W-OD212 也有負載短路警報之 ERR 指示燈。

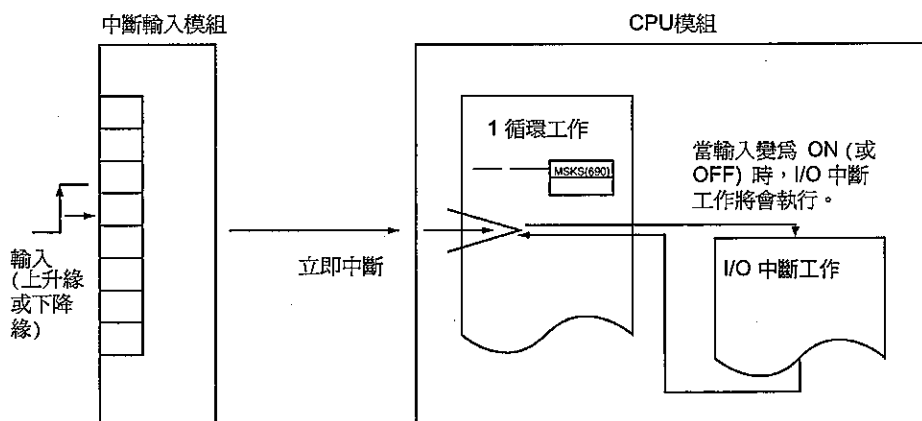
附 18 點端子區塊模組之尺寸規格



中斷輸入模組

功能

中斷輸入模組用來在輸入訊號的上升緣或下降緣時執行中斷程式。當指定的中斷輸入變為 ON (或 OFF) 時，CPU 模組中循環程式的執行會被中斷而執行 I/O 中斷工作單 (工作單編號 100 至 131)。當完成執行 I/O 中斷工作單後，循環程式會再由中斷處的指令開始執行。



可應用的模組

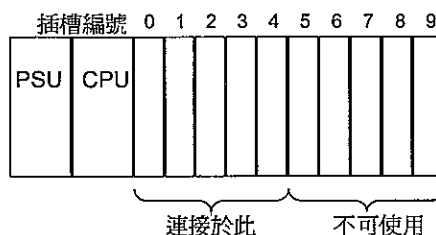
以下的中斷輸入模組可以用於 CJ1-H 或 CJ1M CPU 模組。(中斷輸入模組不能安裝至 CJ1 CPU 模組。)

型號	規格	安裝至 CPU 架的模組數
CJ1W-INT01	24 V DC, 16 輸入	2 以下

應用注意事項

- 中斷輸入模組必須安裝於以下所述的位置中。
 - CJ1-H 與 CJ1 CPU 模組
CPU 架中所有的中斷輸入模組都必須連接，並且必須連接到 CPU 模組右側的五個位置之一。若中斷輸入模組連接到擴充架，則中斷輸入功能將無

法支援。若連接到任何其他的位置或擴充架，則將會發生 I/O 設定錯誤（嚴重錯誤）。



● CJ1M CPU 模組

所有的中斷輸入模組必須連接到 CPU 架且必須連接到 CPU 模組右側的三個位置內。若中斷輸入模組連接到擴充架，則中斷輸入功能將無法支援。若連接到任何其他的位置或擴充架，則將會發生 I/O 設定錯誤（嚴重錯誤）。

- 若中斷輸入模組沒有連接到正確的位置，在由 CX-P 軟體產生 I/O 表時將會發生錯誤。A40110 將會變為 ON 以表示 I/O 設定錯誤，而 A40508 將會變為 ON 以表示中斷輸入模組是在錯誤的位置。

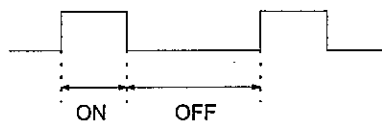
附註 即使一個模組實際上是在正確的位置，還是可以在 I/O 表中註記一個虛擬模組，讓它定義一個不同於實體位置的位置。

可以安裝的中斷輸入模組個數是有限的。（參見上表）

CJ1W-INT01 的輸入響應時間不能改變，PLC 設定中基本 I/O 模組輸入時間常數的相關部份，與 A220 至 A259 中的設定狀態將不會合法。

輸入訊號寬度

輸入訊號必須符合以下的條件：



模組	ON 時間	OFF 時間
CJ1W-INT01	0.05 ms 以上	0.5 ms 以上

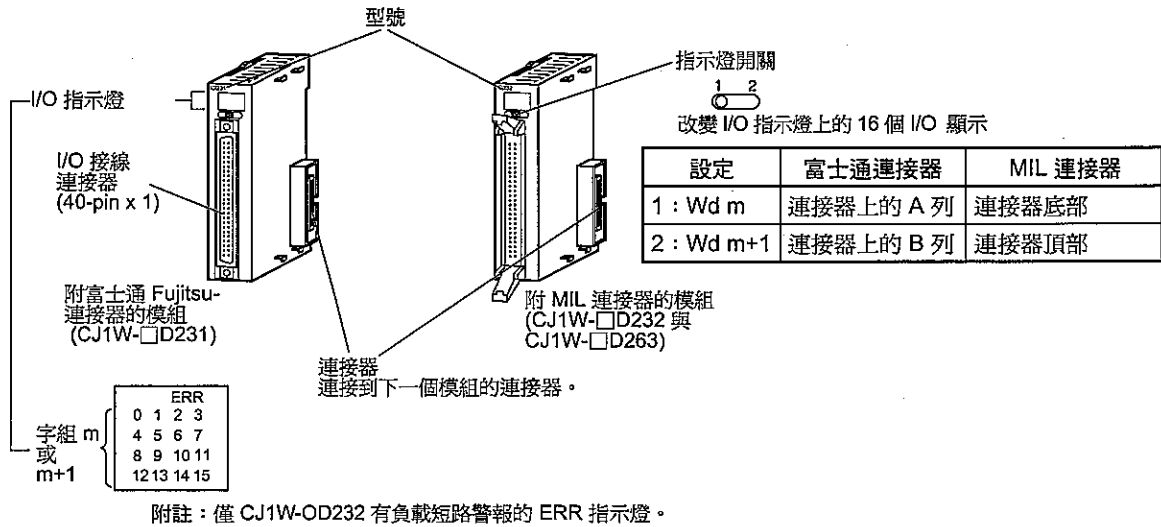
3-4-2 CJ 系列 32/64 點基本 I/O 模組附連接器

模組可連接富士通 - 連接器 (CJ1W-□D□□1) 或 MIL 連接器 (CJ1W-□D□□2/3)。

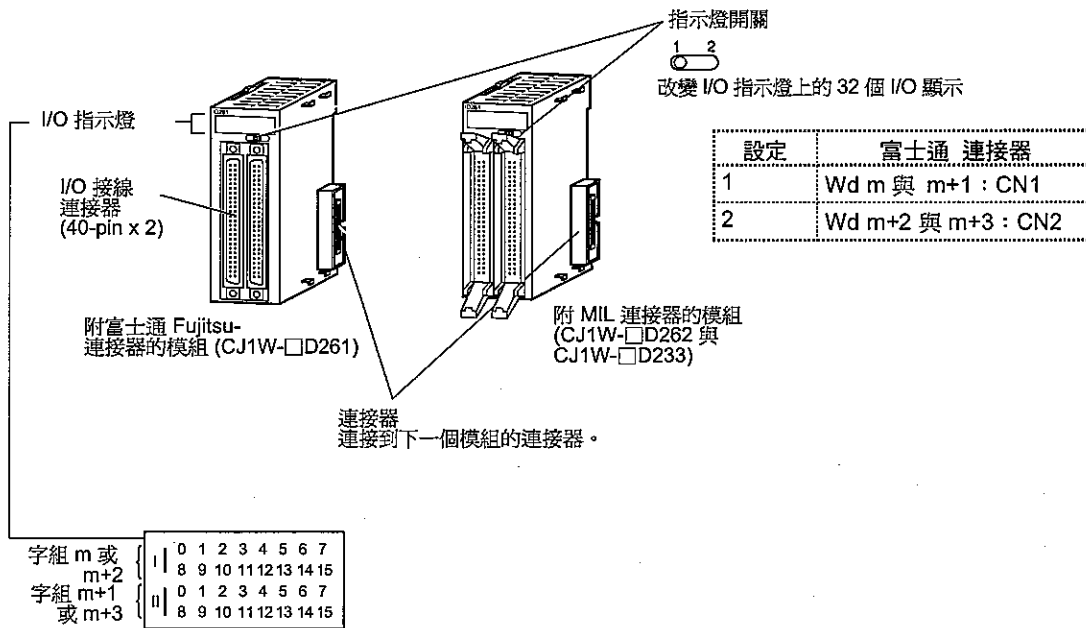
名稱	規格	型號	配置位元數
DC 輸入模組	富士通 - 連接器 24 V DC, 32 輸入	CJ1W-ID231	32
	富士通 - 連接器 24 V DC, 64 輸入	CJ1W-ID261	64
	MIL 連接器 24 V DC, 32 輸入	CJ1W-ID232	32
	MIL 連接器 24 V DC, 64 輸入	CJ1W-ID262	64
電晶體輸出模組 NPN 輸出 (sinking outputs)	富士通 - 連接器 12 至 24 V DC, 0.5 A, 32 輸出	CJ1W-OD231	32
	富士通 - 連接器 12 至 24 V DC, 0.3 A, 64 輸出	CJ1W-OD261	64
	MIL 連接器 12 至 24 V DC, 0.5 A, 32 輸出	CJ1W-OD233	32
	MIL 連接器 12 至 24 V DC, 0.3 A, 64 輸出	CJ1W-OD263	64
電晶體輸出模組 PNP 輸出 (sourcing outputs)	MIL 連接器 24 V DC, 0.5 A, 32 輸出，負載短路保護	CJ1W-OD232	32

組件名稱

32 點模組 (CJ1W-□D23□)



64 點模組 (CJ1W-□D26□)

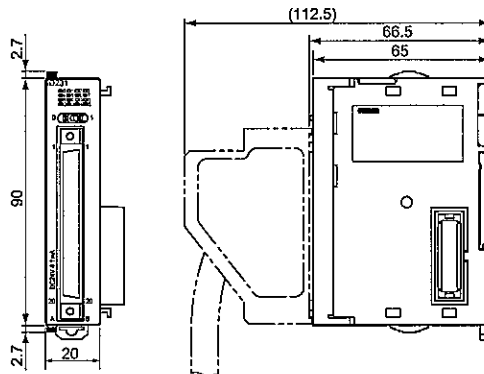


尺寸規格

32 點模組 (40 pin (接腳) x 1)

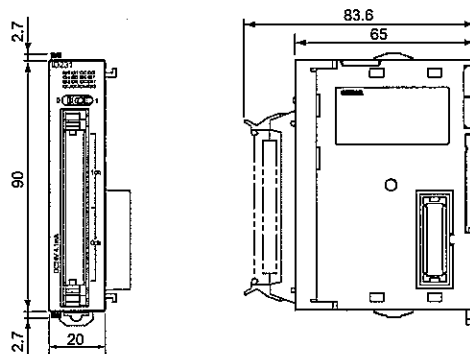
■ 富士通 - 連接器的模組

- CJ1W-ID231 (32 點輸入模組)
- CJ1W-OD231 (32 點輸出模組)



■ MIL 連接器的模組

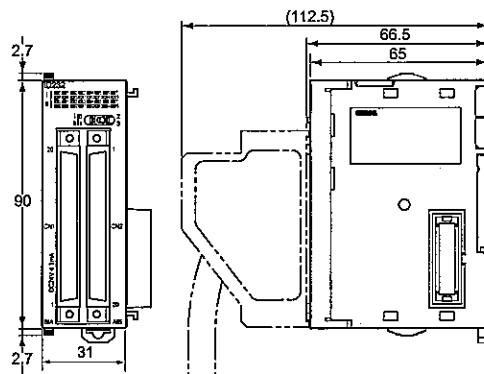
- CJ1W-ID232 (32 點輸入模組)
- CJ1W-OD232 (32 點輸出模組)
- CJ1W-OD233 (32 點輸出模組)



64 點模組 (40 pin(接腳) x 2)

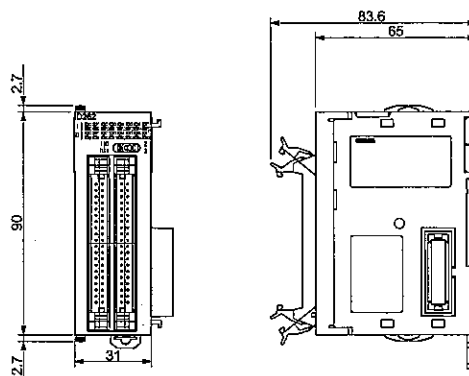
■ 富士通 - 連接器的模組

- CJ1W-ID261 (64 點輸入模組)
- CJ1W-OD261 (64 點輸出模組)



■ 附 MIL 連接器的模組

CJ1W-ID262 (64 點輸入模組)
 CJ1W-OD263 (64 點輸出模組)



連接到連接器 - 端子區塊模組

CJ 系列 32/64 點基本 I/O 模組可以以下表的方式連接到連接器 - 端子區塊轉換模組。

附富士通 - 相容連接器的模組

基本 I/O 模組		連接纜線	連接器 - 端子區塊轉換模組		連接上需要
型號	規格		型號	規格	
CJ1W-ID231	32 點 24-V DC 輸入模組	XW2Z-□□□ B	XW2B-40G5	標準型, M3.5 螺絲端子區塊	1 連接纜線與 1 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
			XW2D-40G6-RF	細長型, M3 螺絲端子區塊, 內藏滋生電阻	
		XW2Z-□□□ D	XW2C-20G5-IN16	16 點輸入共用, M3.5 螺絲端子區塊	1 連接纜線與 2 轉換模組
CJ1W-ID261	64 點 24-V DC 輸入模組	XW2Z-□□□ B	XW2B-40G5	標準型, M3.5 螺絲端子區塊	2 連接纜線與 2 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
			XW2D-40G6-RF	細長型, M3 螺絲端子區塊, 內藏滋生電阻	
		XW2Z-□□□ D	XW2C-20G5-IN16	16 點輸入共用, M3.5 螺絲端子區塊	2 連接纜線與 4 轉換模組
CJ1W-OD231	32 點電晶體輸出模組, NPN 輸出 (sinking outputs)	XW2Z-□□□ B	XW2B-40G5	標準型, M3.5 螺絲端子區塊	1 連接纜線與 1 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
CJ1W-OD261	64 點電晶體輸出模組, NPN 輸出 (sinking outputs)	XW2Z-□□□ B	XW2B-40G5	標準型, M3.5 螺絲端子區塊	2 連接纜線與 2 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	

附 MIL 連接器的模組

基本 I/O 模組		連接纜線	連接器 - 端子區塊轉換模組		連接上需要
型號	規格		型號	規格	
CJ1W-ID232	32 點 24-V DC 輸入模組	XW2Z-□□□K	XW2B-40G5	標準型, M3.5 螺絲端子區塊	1 連接纜線與 1 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
			XW2D-40G6-RM	細長型, M3 螺絲端子區塊, 內藏滋生電阻	
		XW2Z-□□□N	XW2C-20G5-IN16	16 點輸入共用, M3.5 螺絲端子區塊	1 連接纜線與 2 轉換模組
			XW2D-20G6-IO16	細長型, M3 螺絲端子區塊, 內藏滋生電阻	
CJ1W-ID262	64 點 24-V DC 輸入模組	XW2Z-□□□K	XW2B-40G5	標準型, M3.5 螺絲端子區塊	2 連接纜線與 2 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
			XW2D-40G6-RM	細長型, M3 螺絲端子區塊, 內藏滋生電阻	
		XW2Z-□□□N	XW2C-20G5-IN16	16 點輸入共用, M3.5 螺絲端子區塊	2 連接纜線與 4 轉換模組
			XW2D-20G6-IO16	細長型, M3 螺絲端子區塊	
CJ1W-OD232	32 點電晶體輸出模組, PNP (sourcing)	XW2Z-□□□K	XW2B-40G5	標準型, M3.5 螺絲端子區塊	1 連接纜線與 1 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
		XW2Z-□□□N	XW2D-20G6-IO16	細長型, M3 螺絲端子區塊	1 連接纜線與 2 轉換模組
CJ1W-OD233	32 點電晶體輸出模組, PNP (sourcing)	XW2Z-□□□K	XW2B-40G5	標準型, M3.5 螺絲端子區塊	1 連接纜線與 1 轉換模組
			XW2B-40G6	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
		XW2Z-□□□N	XW2D-20G6-IO16	細長型, M3 螺絲端子區塊	1 連接纜線與 2 轉換模組
CJ1W-OD263	64 點電晶體輸出模組, NPN (sinking)	XW2Z-□□□K	XW2B-40G5	標準型, M3.5 螺絲端子區塊	2 連接纜線與 2 轉換模組
			XW2B-40G4	標準型, M3 螺絲端子區塊	
			XW2D-40G6	細長型, M3 螺絲端子區塊	
		XW2Z-□□□N	XW2D-20G6-IO16	細長型, M3 螺絲端子區塊	2 連接纜線與 4 轉換模組

連接至 I/O 端子

CJ 系列 32/64 點基本 I/O 模組可以如下表般連接到 I/O 端子。

附富士通相容連接器的模組

基本 I/O 模組		連接纜線	I/O 端子			連接上需要
型號	規格		型號	型式	輸入電壓 / 輸出型式	
CJ1W-ID231	32 點 24-V DC 輸入模組	G79-I□C-□	G7TC-ID16	輸入區塊	輸入: 24 V DC 輸出: 繼電器	1 連接纜線與 2 I/O 端子
			G7TC-IA16		輸入: 100/200 V AC 輸出: 繼電器	
CJ1W-ID261	64 點 24-V DC 輸入模組	G79-I□C-□	G7TC-ID16		輸入: 24 V DC 輸出: 繼電器	2 連接纜線與 4 I/O 端子
			G7TC-IA16		輸入: 100/200 V AC 輸出: 繼電器	

基本 I/O 模組		連接纜線	I/O 端子			連接上需要
型號	規格		型號	型式	輸入電壓 / 輸出型式	
CJ1W-OD231	32 點電晶體輸出模組，NPN 輸出 (sinking Outputs)	G79-O □ C- □	G7TC-OC16	輸出區塊	輸入：24 V DC 輸出：繼電器	1 連接纜線與 2 I/O 端子
			G70D-SOC16	輸出端子，細長型	輸入：24 V DC 輸出：繼電器	
			G70D-FOM16	輸出端子	輸入：24 V DC 輸出：MOS FET	
			G70A-ZOC16-3 + 繼電器	繼電器端子插槽 (NPN) + 繼電器	—	
CJ1W-OD261	64 點電晶體輸出模組，NPN 輸出 (sinking Outputs)	G79-O □ C- □	G7TC-OC16	輸出區塊	輸入：24 V DC 輸出：繼電器	2 連接纜線與 4 I/O 端子
			G70D-SOC16	輸出端子，細長型	輸入：24 V DC 輸出：繼電器	
			G70D-FOM16	輸出端子	輸入：24 V DC 輸出：MOS FET	
			G70A-ZOC16-3 + 繼電器	繼電器端子插槽 + 繼電器	—	

附 MIL 連接器的模組

基本 I/O 模組		連接纜線	I/O 端子			連接上需要
型號	規格		型號	型式	輸入電壓 / 輸出型式	
CJ1W-ID232	32 點 24-V DC 輸入模組	G79-O □ - □ -DI	G7TC-ID16	輸入區塊	輸入：24 V DC 輸出：繼電器	1 連接纜線與 2 I/O 端子
			G7TC-IA16		輸入：100/200 V AC 輸出：繼電器	
CJ1W-ID262	64 點 24-V DC 輸入模組	G79-O □ - □ -DI	G7TC-ID16	輸入區塊	輸入：24 V DC 輸出：繼電器	2 連接纜線與 4 I/O 端子
			G7TC-IA16		輸入：100/200 V AC 輸出：繼電器	
CJ1W-OD232	32 點電晶體輸出模組，PNP (sourcing)	G79-O □ - □ -DI	G70D-SOC16-1	輸出區塊	輸入：24 V DC 輸出：繼電器	1 連接纜線與 2 I/O 端子
			G70A-ZOC16-4 + 繼電器	繼電器端子插槽 (NPN) + 繼電器	—	
CJ1W-OD233	32 點電晶體輸出模組，NPN (sinking)	G79-O □ - □ -DI	G7TC-OC16	輸出區塊	輸入：24 V DC 輸出：繼電器	1 連接纜線與 2 I/O 端子
			G70D-SOC16 G70D-VSOC16	輸出區塊 (細長型)	輸入：24 V DC 輸出：繼電器	
			G70A-ZOC16-3 + 繼電器	繼電器端子插槽 (NPN) + 繼電器	—	
CJ1W-OD263	64 點電晶體輸出模組，NPN (sinking)	G79-O □ - □ -DI	G7TC-OC16	輸出區塊	輸入：24 V DC 輸出：繼電器	2 連接纜線與 4 I/O 端子
			G70D-SOC16 G70D-VSOC16	輸出區塊 (細長型)	輸入：24 V DC 輸出：繼電器	
			G70A-ZOC16-3 + 繼電器	繼電器端子插槽 (NPN) + 繼電器	—	

第 4 章 指撥開關設定

本章描述 CPU 模組指撥開關的硬體初始設定。

4-1 概觀	38
4-2 詳情	39

4-1 概觀

CJ 系列 PLC 的初始設定有兩種：硬體設定與軟體設定。硬體是以 CPU 模組的指撥開關來設定，而軟體設定則是在 PLC 設定（使用規劃裝置）中進行。打開 CPU 模組的電池室蓋子即可調整指撥開關。

附註 在改變指撥開關設定前務必要先把 PLC 關掉，否則可能會因靜電而造成 PLC 功能不正常。

外觀	pin(接腳)編號	設定	功能
	1	ON	使用者程式記憶體禁止寫入。
		OFF	使用者程式記憶體可以寫入。
	2	ON	當打開電源時，使用者程式會自動傳送並執行。
		OFF	當打開電源時，使用者程式不會自動傳送與執行。
	3	--	未使用
	4	ON	使用內定的週邊埠通訊參數。
		OFF	使用 PLC 設定中的週邊埠通訊參數。
	5	ON	使用內定的 RS-232C 埠通訊參數。
		OFF	使用 PLC 設定中的 RS-232C 埠通訊參數。
	6	ON	使用者定義接腳。把使用者指撥開關接腳旗標變為 OFF (A39512)。
		OFF	使用者定義接腳。把使用者指撥開關接腳旗標變為 ON (A39512)。
	7	ON	由 CPU 模組寫入資料至記憶卡或由記憶卡回復資料至 CPU 模組。
		OFF	驗證記憶卡的內容。
	8	OFF	永遠為 OFF

附註 程式書寫器的顯示語言不是以 CJ 系列 CPU 模組中的指撥開關來設定，而是使用程式書寫器的鍵盤。

4-2 詳情

接腳	功能	設定		描述
1	使用者程式記憶體的寫入保護 (UM) (參見附註 1.)	ON	寫入保護	當這個接腳為 ON 時，使用者程式記憶體為防止寫入保護。把它變為 ON 以防止程式被意外修改。
		OFF	讀 / 寫	
2	程式於起動時自動傳送	ON	是	當此接腳為 ON 時，程式 (AUTOEXEC.OBJ) 與 PLC 設定 (AUTOEXEC.STD) 將於起動時自動由記憶卡傳送至 CPU 模組。(參見附註 4.) 只要插入新的記憶卡並打開電源，即可將 PLC 的軟體 (程式與 PLC 設定) 完全初始化。這可用來快速地把系統切換到一個新的排列方式。 附註 當接腳 7 為 ON 且接腳 8 為 OFF 時，由記憶卡讀取來進行簡易備份是優先的；即使接腳 2 為 ON，程式也不會自動傳送。
		OFF	否	
3	未使用	---	---	---
4	週邊埠通訊參數	ON	使用 PLC 設定內的參數設定	當使用連接至週邊埠的程式書寫器或 CX-P 軟體 (週邊匯流排設定) 時，讓此接腳維持在 OFF。 當週邊埠用於程式書寫器或 CX-P 軟體 (週邊匯流排設定) 以外的裝置時，把這個接腳變為 ON。
		OFF (預設值)	自動檢測規劃裝置 (參見附註 2)	
5	RS-232C 埠通訊參數	ON	自動檢測規劃裝置 (參見附註 3)	當使用 RS-232C 埠連接 CX-P 軟體 (週邊匯流排設定) 以外的裝置 (如規劃端子或主電腦) 時，讓此接腳維持在 OFF。 當使用連接至 RS-232C 埠的 CX-P 軟體 (週邊匯流排設定) 時，把這個接腳變為 ON。
		OFF (預設值)	使用 PLC 設定內的參數設定	
6	使用者定義接腳	ON	A39512 ON	此接腳的 ON/OFF 狀態反應於 A39512。當您想要在程式中不用輸入模組而建立一個永遠為 ON 或永遠為 OFF 的條件時，請使用此一功能。
		OFF (預設值)	A39512 OFF	
7	簡易備份設定	ON	由 CPU 模組寫入記憶卡	按住記憶卡電源開關 3 秒鐘。
			由記憶卡回復至 CPU 模組	要由記憶卡讀出至 CPU 模組，打開 PLC 的電源。當電源 ON 時此動作將優先自動傳送 (接腳 2 為 ON)。
		OFF (預設值)	驗證記憶卡的內容	按住記憶卡電源開關 3 秒鐘。
8	未使用	OFF (預設值)	永遠為 OFF	

- 附註**
- 當接腳 1 為 ON 時，以下的資料為寫入保護：使用者程式與所有 PLC 設定與註記的 I/O 表等參數區域的資料。而且當接腳 1 為 ON 時，即使由規劃裝置進行記憶體清除動作，使用者程式與參數區域仍將無法清除。
 - 自動檢測依以下順序的速率進行：程式書寫器 → 週邊匯流排為 9,600 bps、19,200 bps、38,400 bps 與 115,200 bps。不在週邊匯流排模式的規劃裝置及在週邊匯流排模式卻以 51,200 bps 之速率操作的裝置將無法檢測。
 - 自動檢測操作依以下順序的速率進行：週邊匯流排為 9,600 bps、19,200 bps、38,400 bps 與 115,200 bps。不在週邊匯流排模式的規劃裝置及在週邊匯流排模式卻以其他速率操作的裝置將無法檢測。
 - 當接腳 2 為 ON 且打開電源時，任何 I/O 記憶體檔案 (AUTOEXEC.IOM, ATEXEC □□.IOM) 將無法自動傳送。程式 (AUTOEXEC.OBJ) 與參數區域 (AUTOEXEC.STD) 均必須存在於記憶卡。I/O 記憶體檔案 (AUTOEXEC.IOM, ATEXEC □□.IOM) 為選擇性的。

5. 在進行簡單備份操作後，CJ1-H CPU 模組將維持在 PROGRAM 模式而無法變為 MONITOR 或 RUN 模式，直到供應電源循環為止。在完成備份操作之後，關閉 CPU 模組的電源，改變接腳 7 的設定，然後再打開電源。

指撥開關設定		PLC 設定								
		週邊埠設定 (位址 144 位元 8 至 11)				RS-232C 埠設定 (位址 160 位元 8 至 11)				
		預設值 (0)	NT Link (2)	週邊匯流排 (4)	Host Link (5)	預設值 (0)	NT Link (2)	無協定 (3)	週邊匯流排 (4)	Host Link (5)
接腳 4	OFF	程式書寫器或 CX-P 軟體在週邊匯流排模式 (自動檢測連接裝置的飽率)				---				
	ON	主電腦或 CX-P 軟體 在 Host Link 模式	PT (NT Link)	CX-P 軟體 在週邊匯流 排模式	主電腦或 CX-P 軟體 在 Host Link 模式	---				
接腳 5	OFF	---				主電腦或 CX-P 軟體 在 Host Link 模式	PT (NT Link)	標準的外部 裝置	CX-P 軟體 在週邊匯流 排模式	主電腦或 CX-P 軟體 在 Host Link 模式
	ON	---				CX-P 軟體在週邊匯流排模式 (自動檢測連接裝置的飽率)				

附註 當經由週邊埠或 RS-232C 埠連接 CX-P 軟體時，對於 CX-P 軟體的網路與指撥開關的接腳 4 使用以下的設定。

CX-P 軟體網路設定	週邊埠連接	RS-232C 埠連接	PLC 設定
Toolbus (週邊匯流排)	接腳 4 變為 OFF	接腳 5 變為 ON	無
SYSMAC WAY (Host Link)	接腳 4 變為 ON	接腳 5 變為 OFF	設定為 Host Link

當 CX-P 軟體設定為 Host Link 模式時，在以下的狀況下將無法通訊 (線上進行)：

- 電腦連接到 CPU 模組的週邊埠且接腳 4 為 OFF。
- 電腦連接到 CPU 模組的 RS-232C 埠且接腳 5 為 ON。

要線上進行，設定 CX-P 軟體為週邊匯流排模式，把接腳 4 變為 ON (若是 RS-232C 埠則把接腳 5 變為 OFF)，並設定 PLC 設定的通訊模式為 Host Link 模式。

第 5 章 PLC 設定

本章描述 PLC 設定中的初始軟體設定。

5-1	PLC 設定	42
5-1-1	PLC 設定概觀	42
5-1-2	PLC 設定	44
5-2	PLC 系統設定之解說	70

5-1 PLC 設定

5-1-1 PLC 設定概觀

PLC 設定包括使用者可以改變的基本 CPU 模組軟體設定以訂製 PLC 的操作。這些設定可以用程式書寫器或其他規劃裝置來加以改變。

下表列出 PLC 設定必須改變的狀況。在其他的狀況下，PLC 可用預設的設定來操作。

必須更改設定的狀況	要更改的設定
CJ 系列基本 I/O 模組的輸入回應時間設定在以下的狀況下必須改變： <ul style="list-style-type: none"> 基本 I/O 模組發生顫動或雜訊。 在比循環時間長的時間內接收到短的脈衝輸入。 	基本 I/O 模組輸入回應時間
當 PLC 的的電源打開時，I/O 記憶體 (包括 CIO 區域、工作區域、計時器旗標與 PV、工作單旗標、索引暫存器及資料暫存器) 的所有區域中之資料都必須保持。	起動時的 IOM 保持位元狀態
當 PLC 的的電源打開時，由規劃裝置 (包括程式書寫器) 來的位元強制設定或強制重置的狀態都必須保持。	起動時的強制狀態保持位元狀態
<ul style="list-style-type: none"> 您不想讓操作模式在起動時被程式書寫器的模式開關設定所決定。 您希望 PLC 在起動後立即進入 RUN 模式或 MONITOR 模式並開始操作。 您希望當電源變為 ON 時操作模式為 PROGRAM 以外的模式。 	起動模式
當使用無電池操作時，不需要檢測低電量錯誤。	檢測電池低電量
不需要檢測中斷工作單錯誤。	檢測中斷工作單錯誤
需要資料檔但記憶卡不能使用或那些檔案經常寫入。(部份的 EM 區域將被用為檔案記憶體。)	EM 檔案記憶體
週邊埠將不會以程式書寫器或 CX-P 軟體 (週邊匯流排) 通訊速度自動檢測來使用，也將不會使用如 9,600 bps 的預設 Host Link 通訊設定。 附註 CPU 模組面板的指撥開關之接腳 4 必需為 OFF 以改變 PLC 設定。	週邊埠設定
RS-232C 埠將不會以程式書寫器或 CX-P 軟體 (週邊匯流排) 通訊速度自動檢測來使用，也將不會使用如 9,600 bps 的預設 Host Link 通訊設定。 附註 CPU 模組面板的指撥開關之接腳 5 必需為 OFF 以改變 PLC 設定。	RS-232C 埠設定
您想要經由 NT LinkPT 加快通訊。	設定週邊埠或 RS-232C 埠通訊速率至 "高速 NT Link"。
您想要排定之中斷的時間間隔以 1 ms (或 0.1 ms) 為單位而非以 10 ms 為單位。	排定的中斷時間模組
您想要在指令錯誤時 CPU 模組的操作會停止，即當 ER 旗標或 AER 旗標變為 ON (您要讓錯誤變為嚴重錯誤)。 您想要在發生指令錯誤時 (當 ER 旗標變為 ON 時) 找出錯誤的指令。	指令錯誤操作
您想要設定最小的循環時間以建立一致的 I/O 更新循環。	最小的循環時間
您想要設定 1 秒 (10 ms 至 40,000 ms) 以外的最大循環時間。	觀察循環時間
您想要延遲週邊服務以讓它能執行數個循環。	固定週邊服務時間
您想要指定程式執行的服務週邊之優先序。此處所稱的 "週邊" 包括 CPU 匯流排模組、特殊 I/O 模組、內藏 RS-232C 埠、與週邊埠。	週邊服務優先模式
當電源中斷時進行特殊處理。	電源 OFF 中斷工作單
您想要延遲電源中斷的檢測。	電源 OFF 檢測延遲時間

必須更改設定的狀況	要更改的設定
您想要執行中斷工作單中的 IORF。 您想要縮短使用許多特殊 I/O 模組時的平均循環時間。 您想要延長特殊 I/O 模組的 I/O 更新間隔。	特殊 I/O 模組循環更新
您想要同時增進程式執行與週邊服務回應。	CPU 處理模式 (僅 CJ1-H CPU 模組)
您不想要把 FAL(006) 與 FPD(269) 等使用者定義錯誤記錄到錯誤記錄中。	FAL 錯誤記錄註記 (僅 CJ1-H CPU 模組)
您想要減少因字串處理所造成循環時間中的顫動。	表格資料、字串、與資料位移指令 (僅 CJ1-H CPU 模組) 的背景執行。
您不想等待模組完成起動程序來完成 CPU 模組操作。	起動條件 (僅 CJ1-H CPU 模組)
您想要使用內藏 I/O 的高速計數器。(參見附註 4.)	使用 / 不使用高速計數器 0。 使用 / 不使用高速計數器 1。
您想要使用內藏 I/O 的中斷輸入。(參見附註 4.)	IN0 至 IN3 輸入操作設定。
您想要使用內藏 I/O 的快速回應輸入。(參見附註 4.)	IN0 至 IN3 輸入操作設定。
您想要使用一般用途輸入以內藏 I/O 作為輸入濾波器功能。(參見附註 4.)	一般用途 IN0 至 IN9 輸入時間常數。
您想要使用內藏 I/O 的原點搜尋功能。(參見附註 4.)	脈衝輸出 0/1：原點搜尋操作設定
您想要為原點搜尋功能設定不同的參數。(參見附註 4.)	<ul style="list-style-type: none"> • 脈衝輸出 0/1：原點搜尋，原點返回初速 • 脈衝輸出 0/1：原點搜尋高速 • 脈衝輸出 0/1：原點搜尋近接速度 • 脈衝輸出 0/1：原點補償值 • 脈衝輸出 0/1：原點搜尋加速率 • 脈衝輸出 0/1：原點搜尋減速率 • 脈衝輸出 0/1：極限輸入訊號型式 • 脈衝輸出 0/1：近傍原點輸入訊號型式 • 脈衝輸出 0/1：原點輸入訊號型式 • 脈衝輸出 0/1：定位監控時間 • 脈衝輸出 0/1：原點返回目標速度 • 脈衝輸出 0/1：原點返回加速率 • 脈衝輸出 0/1：原點返回減速率
您想要使用串列 PLC Link。(參見附註 4.)	<ul style="list-style-type: none"> • RS-232C 通訊埠：串列通訊模式 • 串列 PLC Link：格式 • 串列 PLC Link：要傳送的字組數 • 串列 PLC Link：最大模組數 • 串列 PLC Link：輸詢模組之模組數

- 附註
1. 僅 CJ1-H CPU 模組。
 2. 僅 CJ1-H 與 CJ1M CPU 模組。
 3. 僅 CJ1-H 與 CJ1 CPU 模組。
 4. 僅 CJ1M CPU 模組。

5-1-2 PLC 設定

以下各表格中所有非 BIN 的設定除另有說明外均為 Hex。

5-1-2-1 起動標籤 (在 CX-P 軟體上)

起動保持設定

強制狀態保持位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組 (word)	位元 (bit)				
80	14	0:清除 1:維持 預設值:0	這項設定決定起動時強制狀態保持位元的狀態 (A50013) 是否要維持。 當您想要在電源變為 ON 時, 所有強制設定或強制重置的位元維持它們的強制狀態, 把強制狀態保持位元變為 ON 並將這項設定設定為 1 (ON)。	A50013 (強制狀態保持位元)	起動時

IOM 保持位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
80	15	0:清除 1:維持 預設值:0	這項設定決定 IOM 保持位元的狀態 (A50012) 在起動時是否維持。 當您想要在電源變為 ON 時, I/O 記憶體中所有的資料都能維持, 把 IOM 保持位元變為 ON 並將這項設定設定為 1 (ON)。	A50012 (IOM 保持位元)	起動時

模式設定

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
81	---	程式: PROGRAM 模式 監控: MONITOR 模式 執行: RUN 模式 使用程式書寫器: 程式書寫器的模式開關 預設值: 程式	這項設定決定起動模式將會是程式書寫器的模式開關所設定的模式還是此處 PLC 設定所設定的模式。 若這項設定為 PRCN 而程式書寫器沒有連接, 起動模式將會是 RUN 模式。	---	起動時

執行設定

起動條件 (僅 CJ1-H CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
83	15	0:等待模組。 1:不等待。 預設值:0	要使 CPU 模組在一個或更多的模組未完成起動程序下依然起動於 MONITOR 或 PROGRAM 模式, 把這項設定為 1 (不要等待模組)。 要等待所有的模組都完成起動程序, 把這項設定為 0 (等待模組)。	---	起動時

附註 這項設定僅適用於特定模組。

5-1-2-2 CPU 設定標籤 (在 CX-P 軟體上)

執行程序

檢測電池低電量

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
128	15	0: 檢測 1: 不檢測 預設值: 0	這項設定決定是否檢測 CPU 模組電池錯誤。若設定為 0 並檢測到電池錯誤, CPU 模組上的 ERR/ALM 指示燈將會閃爍且電池錯誤旗標 (A40204) 將會變為 ON, 但 CPU 模組仍將繼續操作。	A40204 (電池錯誤旗標)	於下一個循環。

檢測中斷工作單錯誤

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
128	14	0: 檢測 1: 不檢測 預設值: 0	這項設定決定是否檢測中斷工作單錯誤。若設定為 0 並檢測到中斷工作單錯誤, CPU 模組上的 ERR/ALM 指示燈將會閃爍且中斷工作單錯誤旗標 (A40213) 將會變為 ON, 但 CPU 模組仍將繼續操作。	A40213 (中斷工作單錯誤旗標)	於下一個循環。

指令錯誤時停止 CPU (指令錯誤操作)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
197	15	0: 繼續 1: 停止 預設值: 0	這項設定決定指令錯誤 (指令處理錯誤 (ER) 與非法存取錯誤 (AER)) 是否要視為非嚴重錯誤還是視為嚴重錯誤。當這項設定是設定為 1 時, 若 ER 或 AER 旗標變為 ON (即使 AER 旗標只是因為間接 DM/EM BCD 錯誤而變為 ON), 則 CPU 模組將會停止操作。 相關旗標: A29508 (指令處理錯誤旗標) A29509 (間接 DM/EM BCD 錯誤旗標) A29510 (非法存取錯誤旗標)	A29508, A29509, A29510 (若設定為 0, 則即使發生指令錯誤, 這些旗標也不會變為 ON。)	開始動作時。

不要把 FAL 註記至錯誤記錄 (使用者定義 FAL 錯誤儲存。僅 CJ1-H 與 CJ1M CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
129	15	0: 記錄使用者定義 FAL 錯誤於錯誤記錄 (errorlog) 中。 1: 不記錄使用者定義 FAL 錯誤於錯誤記錄中。 預設值: 0	這項設定決定使用者以 FAL(006) 所建立的使用者定義 FAL 錯誤及 FPD(269) 的時間監控是否要記錄到錯誤記錄 (A100 至 A199) 中。把它設定為 1 可讓這些錯誤不被記錄起來。	---	FAL(006) 執行時。(每個循環)

記憶體配置設定

EM 檔案設定致能 (僅 CJ1-H 與 CJ1 CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
136	7	0: 無 1: EM 檔案記憶體致能。 預設值: 0	這項設定決定是否部份的 EM 區域將作為檔案記憶體。	---	由規劃裝置或經由 FINS 指令初值化之後。

EM 起動檔案數 (起動記憶體起動 bank) (僅 CJ1-H 與 CJ1 CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
136	0 至 3	0 至 6 預設值: 0	若位元 7 (以上) 設定為 1, 此處的設定指定檔案記憶體所在的 EM bank。所指定的 EM bank 及後續的 bank 將會當做檔案記憶體使用。若位元 7 設定為 0, 則這項設定將會失效。	A344 (EM 檔案記憶體起動 bank)	由規劃裝置或經由 FINS 指令初值化之後。

背景執行設定

表格資料處理指令 (僅 CJ1-H 與 CJ1M CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
198	15	0: 不在背景中執行。 1: 在背景中執行。 預設值: 0	這項設定決定表格資料指令是否將執行於多個循環時間 (即於背景處理)。	---	開始動作時

字串資料處理指令 (僅 CJ1-H 與 CJ1M CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
198	14	0: 不在背景中執行。 1: 在背景中執行。 預設值: 0	這項設定決定字串資料指令是否將執行於多個循環時間 (即於背景處理)。	---	開始動作時

資料移位處理指令 (僅 CJ1-H 與 CJ1M CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
198	13	0: 不在背景中執行。 1: 在背景中執行。 預設值: 0	這項設定決定資料位移指令是否將執行於多個循環時間 (即於背景處理)。	---	開始動作時

背景執行的通訊埠個數 (僅 CJ1-H 與 CJ1M CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
198	0 至 3	0 至 7: 通訊埠 0 至 7 (內部邏輯埠)	作為背景執行的通訊埠個數 (內部邏輯埠)。	---	開始動作時。

5-1-2-3 計時標籤 (在 CX-P 軟體上)

觀察循環時間設定致能

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
209	15	0:預設值 1:位元 0 至 14 預設值: 0	把位元 0 至 14 設定為 1 可進行觀察循環時間的設定。讓這項設定維持在 0 可得最大的循環時間 1 s。	A40108 (循環時間過長旗標)	開始動作時。 (操作期間不能改變。)

觀察循環時間

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
209	0 至 14	001 至 FA0: 10 至 40,000 ms (以 10 ms 為單位) 預設值: 001 (1 s)	這項設定只有在 209 的位元 15 設定為 1 時有效。若循環時間超過這項設定, 則循環時間過長旗標 (A40108) 將會變為 ON。	A264 與 A265 (現在的循環時間)	開始動作時 (操作期間不能改變。)

循環時間 (最小的循環時間)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
208	0 至 15	0001 至 7D00: 1 至 32,000 ms (以 1 ms 為單位) 預設值: 0000 (無最小值)	設定為 0001 至 7D00 以指定最小的循環時間。若循環時間小於這項設定, 它將會延長到這個時間。讓這項設定為 0000 可得到可變循環時間。(操作期間不能改變。) 當使用並行處理模式時, 這個循環時間將會應用於程式的執行循環。	---	開始動作時。

排定的中斷間隔

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
195	0 至 3	0: 10 ms 1: 1.0 ms 2: 0.1 ms (僅 CJ1M CPU 模組) 預設值: 0	這項設定決定使用於排定的中斷間隔設定之時間模組。 (這項設定在操作期間不能改變。)	---	開始動作時。

電源 OFF 檢測時間 (電源 OFF 檢測延遲時間)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
225	0 至 7	00 至 0A: 0 至 10 ms (以 1 ms 為單位) 預設值: 00	這項設定決定由檢測到電源中斷 (大約在電源電壓降至額定值的 85% 後之 10 至 25 ms (AC 電源) 或 2 至 5 ms (DC)) 至確認電源中斷之間需要多少的時間延遲。預設的設定為 0 ms。 在電源 OFF 中斷工作單致能下, 電源中斷確認時它將會執行。若電源 OFF 中斷工作單除能, 則 CPU 將會重置並停止操作。	---	起動時或開始動作時。 (操作期間不能改變。)

電源 OFF 中斷除能

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
225	15	0:除能 (Disabled) 1:致能 (Enabled) 預設值: 0	在這項設定設定為 1 下, 電源中斷時將會執行電源 OFF 中斷工作單。	---	起動時或開始動作時。 (操作期間不能改變。)

5-1-2-4 SIOU 更新標籤 (在 CX-P 軟體上)

特殊 I/O 模組循環更新

項目	程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
	字組	位元				
模組 0 至 15 的循環更新	226	0 至 15	0:致能 (Enabled) 1:除能 (Disabled) 預設值: 0	這些設定決定在特殊 I/O 模組的循環更新期間, 資料是否要在指定的模組與特殊 I/O 模組的配置字組 (10 字組 / 模組) 間進行交換。 把相對應的位元變為 ON 以便在中斷工作單中將被以 IORF(097)、數個特殊 I/O 模組更新的模組在使用中而您又不想延長循環時間時, 或循環時間太短以致於特殊 I/O 模組的內部處理無法跟上時, 能使循環更新除能。 (特殊 I/O 模組可以由程式中以 IORF(097) 更新。)	---	開始動作時。
模組 16 至 31 的循環更新	227	0 至 15	0:致能 (Enabled) 1:除能 (Disabled) 預設值: 0			
模組 32 至 47 的循環更新	228	0 至 15	0:致能 (Enabled) 1:除能 (Disabled) 預設值: 0			
模組 48 至 63 的循環更新	229	0 至 15	0:致能 (Enabled) 1:除能 (Disabled) 預設值: 0			
模組 64 至 79 的循環更新	230	0 至 15	0:致能 (Enabled) 1:除能 (Disabled) 預設值: 0			
模組 80 至 95 的循環更新	231	0 至 15	0:致能 (Enabled) 1:除能 (Disabled) 預設值: 0			

5-1-2-5 模組設定標籤 (在 CX-P 軟體上)

基本 I/O 模組輸入 (Rack) 回應時間

項目	程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
	字組	位元				
Rack 0, 插槽 0	10	0 至 7	00 : 8 ms	設定 CJ 系列基本 I/O 模組的輸入回應時間 (ON 回應時間 = OFF 回應時間)。預設的設定為 8 ms 而設定範圍為 0 ms 至 32 ms。 可以增加這個值以減少顫動及雜訊的影響，也可以減少這個值以接受較短的輸入脈衝。	A220 至 A259 : 基本 I/O 模組真正的輸入回應時間	起動時。
Rack 0, 插槽 1		8 至 15	10 : 0 ms			
Rack 0, 插槽 2	11	0 至 7	11 : 0.5 ms			
Rack 0, 插槽 3		8 至 15	12 : 1 ms			
Rack 0, 插槽 4	12	0 至 7	13 : 2 ms			
Rack 0, 插槽 5		8 至 15	14 : 4 ms			
Rack 0, 插槽 6	13	0 至 7	15 : 8 ms			
Rack 0, 插槽 7		8 至 15	16 : 16 ms			
Rack 0, 插槽 8	14	0 至 7	17 : 32 ms			
Rack 0, 插槽 9		8 至 15	預設值 : 00 (8 ms)			
Rack 1, 插槽 0 至 9	15 至 19	參見 Rack 0				
Rack 2, 插槽 0 至 9	20 至 24					
Rack 3, 插槽 0 至 9	25 至 29					
Rack 4, 插槽 0 至 9	30 至 34					
Rack 5, 插槽 0 至 9	35 至 39					
Rack 6, 插槽 0 至 9	40 至 44					
Rack 7, 插槽 0 至 9	45 至 49					

5-1-2-6 Host Link 埠標籤 (在 CX-P 軟體上)

以下的設定適用於當 CPU 模組的指撥開關之接腳 5 為 OFF 時。

Host Link 設定

通訊設定

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	15	0:標準的* 1:PLC 設定 (客製的) 預設值:0	* 預設的設定適用於 1 起動位元、7 資料位元、偶同位、2 停止位元、與鮑率為 9,600 bps 時。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

鮑率 (bps)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
161	0 至 7	00:9,600 bps 01:300 bps 02:600 bps 03:1,200 bps 04:2,400 bps 05:4,800 bps 06:9,600 bps 07:19,200 bps 08:38,400 bps 09:57,600 bps 0A:115,200 bps 預設值:00	這些設定只在通訊模式設定為 Host Link 或 no-protocol 時有效。 這些設定只在 RS-232C 埠設定選擇設定為 1: PLC 設定時有效。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

格式: 資料位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	3	0:7 位元 1:8 位元 預設值:0	這些設定只在通訊模式設定為 Host Link 或 no-protocol 時有效。 這些設定在 RS-232C 埠設定選擇設定為 1: PLC 設定時也有效。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

格式: 停止位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	2	0:2 位元 1:1 位元 預設值:0	這些設定只在通訊模式設定為 Host Link 或 no-protocol 時有效。 這些設定在 RS-232C 埠設定選擇設定為 1: PLC 設定時也有效。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

格式: 同位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	0 至 1	00:偶 01:奇 10:無 預設值:00	這些設定只在通訊模式設定為 Host Link 或 no-protocol 時有效。 這些設定在 RS-232C 埠設定選擇設定為 1: PLC 設定時也有效。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

模式：通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	8 至 11	00 : Host Link 05 : Host Link 預設值 : 0	這項設定決定 RS-232C 埠將會操作於 Host Link 模式還是另一個串列通訊模式。(Host Link 不能指定為 00 或 05。) 這個週邊匯流排模式用於與程式書寫器以外的規劃裝置建立通訊。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

模組數 (用於 Host Link 模式下的 CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
163	0 至 7	00 至 1F : (0 至 31) 預設值 : 00	這項設定決定當 CPU 模組連接到 1 對 N (N=2 至 32) Host Link 的模組數。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

NT Link 設定

模式：通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	8 至 11	02: 1:N NT Link 預設值 : 0	這項設定決定 RS-232C 埠將操作於 Host Link 模式還是另一個串列通訊模式。 附註 在 1:1 NT Link 下,有可能無法與 PT 建立通訊。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

鮑率 (bps)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
161	0 至 7	00 : 標準的 0A : 高速 NT Link* 預設值 : 00	* 當由 CX-P 軟體設定這個值時,設定為 115,200 bps。要回復標準的設定,讓此設定維持為 "PLC 設定" 並設定鮑率至 9,600 bps。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

NT Link 最大數 (NT Link 模式下的最大模組數)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
166	0 至 3	0 至 7 預設值 : 0	這項設定決定可以連接到 PLC 的最大 PT 模組數。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

週邊匯流排設定

通訊設定

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	15	0 : 標準的 * 1 : PLC 設定 (客製的) 預設值 : 0	* 預設的設定適用於鮑率為 9,600 bps 時。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

模式：通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	8 至 11	04：週邊匯流排 預設值：0	這項設定決定 RS-232C 埠將會操作於 Host Link 模式還是另一個串列通訊模式。(Host Link 不能指定為 00 或 05。) 這個週邊匯流排模式用於與程式書寫器以外的規劃裝置建立通訊。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

速率 (bps)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
161	0 至 7	00：9,600 bps 06：9,600 bps 07：19,200 bps 08：38,400 bps 09：57,600 bps 0A：115,200 bps 預設值：00	當通訊模式設定為週邊匯流排時可設定 00 或 06 至 0A。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

串列 PLC Link (僅 CJ1M CPU 模組)

輸詢模組：串列通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	8 至 11	8：串列 PLC Link 輸詢模組 預設值：0	這項設定指明 RS-232C 埠要使用串列通訊模式。它也指出以本地節點作為串列 PLC Link 輸詢模組。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：連接埠速率

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
161	0 至 7	00：9,600 bps 0A：115,200 bps 預設值：00	這項設定指定串列 PLC Link 速率。 附註：這項設定對於使用串列 PLC Link 的所有被輸詢模組與輸詢模組均必須相同。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：Link 方法

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
166	15	0：完全 Link 方法 1：輸詢模組 Link 方法 預設值：0	這項設定指定串列 PLC Link 的 Link 方法。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：Link 字組數

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
166	4 至 7	1 至 A 預設值：0 (參見附註) 附註：若設定為預設值，字組數將自動變為 10 (A Hex)。	這項設定指定用於串列 PLC Link 的串列 PLC Link 區域中每個節點的字組數。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：串列 PLC Link 中的最大模組數

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
166	0 至 3	0 至 7 預設值：0	這項設定指定可以連接到串列 PLC Link 中的輸詢模組之最大模組數。 附註：在計算模組數時必須把要連接的 PT 計算在內。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：串列通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
160	8 至 11	7：串列 PLC Link 輸詢模組 預設值：0	這項設定指明 RS-232C 埠要使用串列通訊模式。它也指出以本地節點作為串列 PLC Link 輸詢模組。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：連接埠速率

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
161	0 至 7	00：9,600 bps 0A：115,200 bps 預設值：00	這項設定指定串列 PLC Link 速率。 附註：這項設定對於使用串列 PLC Link 的所有被輸詢模組與輸詢模組均必須相同。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

輸詢模組：串列 PLC Link 輸詢模組之模組數

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
167	0 至 3	0 至 7 預設值：00	這項設定指定串列 PLC Link 上本地節點的輸詢模組之模組數。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。

無協定 (No-protocol) 設定

插入中文起始碼 / 結束碼

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
164	8 至 15	00 至 FF 預設值：00	起始碼：只有在 165 的位元 12 至 15 中之起始碼為致能 (1) 時才能設定這個起始碼。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)
	0 至 7	00 至 FF 預設值：00	結束碼：只有在 165 的位元 8 至 11 中之結束碼為致能 (1) 時才能設定這個結束碼。		
165	12	0：無 1：164 中的碼 預設值：0	起始碼設定： 1 的設定使 164 位元 8 至 15 中的起始碼致能。		
	8 至 9	0：無 1：164 中的碼 2：CR+LF 預設值：0	結束碼設定： 設定為 0 時，必須指定接收到的資料量。設定為 1 允許 164 的位元 0 至 7 為結束碼。設定為 2 允許 CR+LF 的結束碼。		
	0 至 7	00：256 位元組 01 至 FF：1 至 255 位元組 預設值：00	設定以 no-protocol 通定發要及接收的資料長度。結束碼與起始碼不包含在資料長度中。只有在 165 的位元 8 至 11 中的結束碼設定為 "0：無" 時才要設定這個值。 這項設定可用來改變以 TXD(236) 或 RXD(235) 一次可傳送的資料量。預設的設定為 256 位元的最大值。		

延遲

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
162	0 至 15	0000 至 270F : 0 至 99990 ms (以 10-ms 為單位) 預設值 : 0000	這項設定決定從執行 TXD(236) 到資料真正由指定的連接埠傳送的時間延遲。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

5-1-2-7 週邊埠標籤 (在 CX-P 軟體上)

以下的設定適用於當 CPU 模組的指撥開關之接腳 4 為 ON 時。

Host Link 設定

通訊設定

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	15	0 : 標準的 * 1 : PLC 設定 (客製的) 預設值 : 0	* 預設的設定適用於 1 起動位元、7 資料位元、偶同位、2 停止位元、與鮑率為 9,600 bps 時。	A61902 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

鮑率 (bps)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
145	0 至 7	00 : 9,600 bps 01 : 300 bps 02 : 600 bps 03 : 1,200 bps 04 : 2,400 bps 05 : 4,800 bps 06 : 9,600 bps 07 : 19,200 bps 08 : 38,400 bps 09 : 57,600 bps 0A : 115,200 bps 預設值 : 00	這些設定只在通訊模式設定為 Host Link 或 no-protocol 時有效。 這些設定在週邊埠設定選擇設定為 1:PLC 設定時也有效。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

格式 : 資料位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	3	0 : 7 位元 1 : 8 位元 預設值 : 0	這些設定只在通訊模式設定為 Host Link 時有效。 這些設定在週邊埠設定選擇設定為 1:PLC 設定時也有效。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

格式：停止位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	2	0:2 位元 1:1 位元 預設值:0	這些設定只在通訊模式設定為 Host Link 時有效。 這些設定在週邊埠設定選擇設定為 1:PLC 設定時也有效。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

格式：同位元

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	0 與 1	00:偶 01:奇 10:無 預設值:00	這些設定只在通訊模式設定為 Host Link 時有效。 這些設定在週邊埠設定選擇設定為 1:PLC 設定時也有效。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

模式：通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	8 至 11	00: Host Link 05: Host Link 預設值:0	這項設定決定是週邊埠將操作於 Host Link 模式還是另一個串列通訊模式。(Host Link 可以指定為 00 或 05) 此週邊匯流排模式用於以程式書寫器以外之規劃裝置建立通訊。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

模組數 (用於 Host Link 模式中的 CPU 模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
147	0 至 7	00 至 1F (0 至 31) 預設值:00	這項設定決定當 CPU 模組連接到 1 對 N (N=2 至 32) Host Link 時的模組數。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

NT Link 設定

模式：通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	8 至 11	02: 1:N NT Link 預設值:0	這項設定決定 RS-232C 埠將操作於 Host Link 模式還是另一個串列通訊模式。 附註 在 1:1 NT Link 下,有可能無法與 PT 建立通訊。	A61902 (RS-232C 埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

鮑率 (bps)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
145	0 至 7	00: 標準的 0A: 高速 NT Link* 預設值: 00	* 當由 CX-P 軟體設定這個值時, 設定為 115,200 bps。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

NT Link 最大數 (NT Link 模式的最大模組數)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
150	0 至 3	0 至 7 預設值: 0	這項設定決定可以連接到 NT Link 模式下之 PLC 的最大 PT 模組數。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

週邊匯流排設定

通訊設定

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	15	0: 預設值 (標準的)* 1: PLC 設定 (客製的) 預設值: 0	* 預設的設定適用於鮑率為 9,600 bps 時。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

模式: 通訊模式

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	8 至 11	4: 週邊匯流排 預設值: 0	這項設定決定週邊埠是否在通訊模式。週邊匯流排模式用於程式書寫器以外所有的規劃裝置。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

鮑率 (bps)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
144	0 至 7	00: 9,600 bps 06: 9,600 bps 07: 19,200 bps 08: 38,400 bps 09: 57,600 bps 0A: 115,200 bps 預設值: 00	以下的設定適用於週邊匯流排模式: 00 與 06 至 0A Hex。	A61901 (週邊埠設定改變旗標)	於下一個循環。 (也可以用 STUP (237) 來改變。)

5-1-2-8 週邊服務標籤 (在 CX-P 軟體上) (CPU 處理模式設定)

週邊服務模式 (週邊服務優先模式)

指令執行時間

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
219	08 至 15	00 05 至 FF (Hex) 預設值: 00 (Hex)	00: 優先服務除能 05 至 FF: 指令執行的時間截分 (5 至 255 ms, 增量為 1 ms)	A266 與 A267	開始動作時 (操作期間不能改變。)

週邊服務執行時間

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
219	00 至 07	00 至 FF (Hex) 預設值: 00 (Hex)	00: 優先服務除能 01 至 FF: 週邊服務的時間截分 (0.1 至 25.5 ms, 增量為 0.1 ms)	A266 與 A267	開始動作時 (操作期間不能改變。)

目標模組 (優先服務的模組)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定時
字組	位元				
220	08 至 15	00 10 至 1F 20 至 2F E1 FC FD 預設值: 00	最多可以指定五個優先服務模組。 00: 優先服務除能 10 至 1F: CPU 匯流排模組模組數 (0 至 15) + 10 (Hex) 20 至 2F: CJ 系列 特殊 I/O 模組模組數 (0 至 96) + 20 (Hex) FC: RS-232C 埠 FD: 週邊埠	--	開始動作時 (操作期間不能改變。)
221	08 至 15				
222	08 至 15				

同步 / 非同步通訊 (並行處理模式)

以下的設定僅支援 CJ1-H CPU 模組

執行模式 (並行處理模式)

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定的時間
字組	位元				
219	08 至 15	00 01 02 預設值: 00	00: 未指定 (並行處理除能) 01: 同步 (同步記憶體存取) 02: 非同步 (非同步記憶體存取)	--	開始動作時 (操作期間不能改變。)

對所有的事件設定時間 (固定週邊服務時間)

固定服務時間致能

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定的時間
字組	位元				
218	15	0: 預設值 * 1: 位元 0 至 7 預設值: 0	位元 0 至 7 設定為 1 以使固定週邊服務時間致能。 * 預設值: 循環時間的 4%	--	開始動作時 (操作期間不能改變。)

固定服務時間

程式書寫器中的位址		設定	功能	相關旗標與字組	由 CPU 模組讀取設定的時間
字組	位元				
218	0 至 7	00 至 FF : 0.0 至 25.5 ms (以 0.1 ms 為單位) 預設值 : 00	設定週邊服務時間。 這項設定只在 218 的位元 15 設定為 1 時為有效。	—	開始動作時 (操作期間不能改變。)

5-1-2-9 內藏 I/O 設定標籤 (在 CX-P 軟體上)

以下各表顯示 CX-P 軟體的內藏 I/O 設定標籤中的各項設定。這些設定適用於配備內藏 I/O 的 CJ1M CPU 模組。

高速計數器 0 操作設定

高速計數器 0 致能 / 除能

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
50	12 至 15	0 Hex : 不使用計數器 1 Hex* : 使用計數器 (60 kHz). 2 Hex* : 使用計數器 (100 kHz).	0 Hex	指定是否使用高速計數器 0。 附註 當高速計數器 0 為致能 (設定 1 或 2) 時, IN 的輸入操作設定 8 與 IN9 為除能的。若重置方法 設定為相位 Z 訊號 + 軟體重置, 則 IN 的輸入操作設定 3 也是除能的。	---	開始動作時

高速計數器 0 計數模式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
50	08 至 11	0 Hex : 線性模式 1 Hex : 環狀模式	0 Hex	指定高速計數器 0 的計數模式。	---	開始動作時

高速計數器 0 循環最大計數 (環狀計數器最大值)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
51	00 至 15	00000000 至 FFFFFFFF Hex (參見附註)	00000000 Hex	設定高速計數器 0 的最大環狀計數。 在高速計數器 0 計數模式 設定為環狀模式下, 當計數器 PV 超過最大的環狀計數時, 將會自動重置為 0。	A270 (高速計數器 0 PV 的最右邊 4 位元)	開始動作時
52	00 至 15				A271 (高速計數器 0 PV 的最左邊 4 位元)	

高速計數器 0 重置方法

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
50	04 至 07	0 Hex : Z 相位與軟體重置 1 Hex : 軟體重置	0 Hex	指定高速計數器 0 的重置方法。	---	開始動作時

高速計數器 0 脈衝輸入設定 (脈衝輸入模式)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
50	00 至 03	0 Hex : 差動相位輸入 1 Hex : 脈衝 + 方向輸入 2 Hex : 上數 / 下數輸入 3 Hex : 遞增脈衝輸入	0 Hex	指定高速計數器 0 的脈衝輸入方法。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

高速計數器 1 操作設定

高速計數器 1 致能 / 除能

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
53	12 至 15	0 Hex : 不使用計數器 1 Hex* : 使用計數器 (60 kHz). 2 Hex* : 使用計數器 (100 kHz).	0 Hex	指定是否使用高速計數器 1。 附註 當高速計數器 1 為致能 (設定 1 或 2) 時，IN 的輸入操作設定 IN6 與 IN7 為除能的。若重置方法設定為相位 Z 訊號 + 軟體重置，則 IN 的輸入操作設定 2 也是除能的。	---	開始動作時

高速計數器 1 計數模式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
53	08 至 11	0 Hex : 線性模式 1 Hex : 環狀模式	0 Hex	指定高速計數器 1 的計數模式。	---	開始動作時

高速計數器 1 循環最大計數 (環狀計數器最大值)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
54	00 至 15	00000000 至 FFFFFFFF Hex (參見附註)	00000000 Hex	設定高速計數器 1 的最大環狀計數。在高速計數器 1 計數模式設定為環狀模式下，當計數器 PV 超過最大的環狀計數時，將會自動重置為 0。	A272 (高速計數器 1 PV 的最右邊 4 位元)	開始動作時
55	00 至 15				A273 (高速計數器 1 PV 的最左邊 4 位元)	

高速計數器 0 重置方法

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
53	04 至 07	0 Hex : Z 相位與軟體重置 1 Hex : 軟體重置	0 Hex	指定高速計數器 1 的重置方法。	---	開始動作時

高速計數器 1 脈衝輸入設定 (脈衝輸入模式)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
53	00 至 03	0 Hex : 差動相位輸入 1 Hex : 脈衝 + 方向輸入 2 Hex : 上數 / 下數輸入 3 Hex : 遞增脈衝輸入	0 Hex	指定高速計數器 1 的脈衝輸入方法。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

內藏輸入 IN0 至 IN3 的輸入操作設定

IN0 的輸入操作設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
60	00 至 03	0 Hex : 一般 (一般用途輸入) 1 Hex : 中斷 (中斷輸入) (參見附註) 2 Hex : 快速 (快速回應輸入)	0 Hex	指定於內藏輸入 IN0 所接收的輸入種類。	---	開始動作時

附註 當 IN0 設定為中斷輸入 (1 Hex) 時，用 MSKS(690) 指令選擇直接模式或計數器模式操作。

IN1 的輸入操作設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
60	04 至 07	0 Hex : 一般(一般用途輸入) 1 Hex : 中斷(中斷輸入)(參見附註) 2 Hex : 快速(快速回應輸入)	0 Hex	指定於內藏輸入 IN1 所接收的輸入種類。	---	開始動作時

附註 當 IN1 設定為中斷輸入 (1 Hex) 時, 用 MSKS(690) 指令選擇直接模式或計數器模式操作。

IN2 的輸入操作設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
60	08 至 11	0 Hex : 一般(一般用途輸入) 1 Hex : 中斷(中斷輸入)(參見附註) 2 Hex : 快速(快速回應輸入)	0 Hex	指定於內藏輸入 IN2 所接收的輸入種類。 當使用高速計數器 1 且重置方法設定為相位 Z 訊號 + 軟體重置時, IN2 的輸入操作設定為除能的。	---	開始動作時

附註 當 IN2 設定為中斷輸入 (1 Hex) 時, 使用 MSKS(690) 指令選擇直接模式或計數器模式操作。

IN3 的輸入操作設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
60	12 至 15	0 Hex : 一般(一般用途輸入) 1 Hex : 中斷(中斷輸入)(參見附註) 2 Hex : 快速(快速回應輸入)	0 Hex	指定於內藏輸入 IN3 所接收的輸入種類。 附註 當使用高速計數器 0 且重置方法設定為相位 Z 訊號 + 軟體重置時, IN3 的輸入操作設定為除能的。	---	開始動作時

附註 當 IN3 設定為中斷輸入 (1 Hex) 時, 用 MSKS(690) 指令選擇直接模式或計數器模式操作。

一般用途輸入的輸入時間常數設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
61	00 至 07	00 Hex : 預設值 (8 ms) 10 Hex : 0 ms (無濾波器) 11 Hex : 0.5 ms 12 Hex : 1 ms 13 Hex : 2 ms 14 Hex : 4 ms 15 Hex : 8 ms 16 Hex : 16 ms 17 Hex : 32 ms	0 Hex	指定一般用途輸入 IN0 至 IN9 輸入的時間常數。 附註 這項設定在輸入設定為中斷輸入、快速回應輸入、或高速計數器時是沒有影響的。	---	開始動作時

5-1-2-10 定義原點 1/2 標籤的定義原點操作設定欄位 (在 CX-P 軟體上)(原點搜尋)

以下各表顯示定義原點 1/2 標籤的 CX-P 軟體的定義原點操作設定欄位中原點搜尋功能的各項設定。這些設定適用於配備內藏 I/O 的 CJ1M CPU 模組。

原點搜尋 0 設定 (在定義原點 1 標籤上的 CX-P 軟體的定義原點操作設定)

脈衝輸出 0 使用原點操作設定 (原點搜尋功能致能 / 除能)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
256	00 至 03	0 Hex : 除能 1 Hex* : 致能	0 Hex	指定原點搜尋功能是否作為脈衝輸出 0。 附註 當原點搜尋功能對於脈衝輸出 0 為致能 (設定 1) 時, 中斷輸入 0 與 1 及 PWM(891) 輸出 0 不能使用。高速計數器 0 與 1 可以使用。	---	開始動作時

脈衝輸出 0 原點搜尋方向設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
257	12 至 15	0 Hex : 順時針方向 1 Hex : 反時針方向	0 Hex	指定脈衝輸出 0 原點搜尋方向。	---	開始動作時

脈衝輸出 0 原點檢測方法

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
257	08 至 11	0 Hex: 方法 0 (原點檢測方法 0) 1 Hex: 方法 1 (原點檢測方法 1) 2 Hex: 方法 2 (原點檢測方法 2)	0 Hex	指定用於脈衝輸出 0 的原點檢測方法。	---	開始動作時

脈衝輸出 0 原點搜尋操作設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
257	04 至 07	0 Hex: 反向 1 (反轉模式 1) 1 Hex: 反向 2 (反轉模式 2)	0 Hex	指定用於脈衝輸出 0 的原點搜尋操作。	---	開始動作時

脈衝輸出 0 原點搜尋操作模式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
257	00 至 03	0 Hex: 模式 0 1 Hex: 模式 1 2 Hex: 模式 2	0 Hex	指定原點搜尋模式用於脈衝輸出 0。	---	開始動作時

脈衝輸出 0 原點輸入訊號型式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
268	08 至 11	0 Hex: NC 1 Hex: NO	0 Hex	指定脈衝輸出 0 的原點輸入訊號是常閉還是常開。	---	開始動作時

脈衝輸出 0 近傍原點輸入訊號型式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
268	04 至 07	0 Hex: NC 1 Hex: NO	0 Hex	指定脈衝輸出 0 的近傍原點輸入訊號是常閉還是常開。	---	開始動作時

脈衝輸出 0 極限輸入訊號型式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
268	00 至 03	0 Hex: NC 1 Hex: NO	0 Hex	指定脈衝輸出 0 的原點極限輸入訊號是常閉還是常開。	---	開始動作時

脈衝輸出 0 原點搜尋 / 原點復歸起動速度

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
258	00 至 15	00000000 至 000186A0 Hex (參見附註)	00000000 Hex	指定脈衝輸出 0 原點搜尋與原點返回操作的起動速度 (0 至 100,000 pps)。	---	開始動作時
259	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 0 原點搜尋高速

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
260	00 至 15	00000001 至 000186A0 Hex (參見附註)	00000000 Hex	指定脈衝輸出 0 原點搜尋與原點返回操作的高速設定 (1 至 100,000 pps)。	---	開始動作時
261	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 0 近傍原點搜尋速度

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
262	00 至 15	00000001 至 000186A0 Hex (參見附註)	00000000 Hex	指定脈衝輸出 0 原點搜尋與原點返回操作的近接速度設定 (1 至 100,000 pps)。	---	開始動作時
263	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 0 搜尋補償值 (原點補償)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
264	00 至 15	80000000 至 7FFFFFFF Hex (參見附註)	---	設定脈衝輸出 0 原點補償 (-2,147,483,648 至 2,147,483,647)。	---	開始動作時
265	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 0 原點搜尋加速比率

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
266	00 至 15	0001 至 07D0 Hex (參見附註)	---	設定原點搜尋加速率用於脈衝輸出 0 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 0 原點搜尋減速比率

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
267	00 至 15	0001 至 07D0 Hex(參見附註)	---	設定用於脈衝輸出 0 (1 至 2,000 脈衝 / 4 ms) 的原點搜尋減速率。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 0 定位監控時間

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
269	00 至 15	0000 至 270F Hex (參見附註)	0000 Hex	指定用於脈衝輸出 0 的定位監控時間 (0 至 9,999 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

原點搜尋 1 設定 (在定義原點 2 標籤上的 CX-P 軟體的定義原點操作設定)

脈衝輸出 1 使用原點操作設定 (原點搜尋功能致能 (Enabled)/ 除能 (Disabled))

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
274	00 至 03	0 Hex : 除能 1 Hex* : 致能	0 Hex	指定原點搜尋功能是否作為脈衝輸出 1。 附註 當原點搜尋功能對於脈衝輸出 1 為致能 (設定 1) 時，中斷輸入 2 與 3 及 PWM(891) 輸出 1 不能使用。高速計數器 0 與 1 可以使用。	---	開始動作時

脈衝輸出 1 原點搜尋方向設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
275	12 至 15	0 Hex : 順時針方向 1 Hex : 反時針方向	0 Hex	指定脈衝輸出 1 的原點搜尋方向。	---	開始動作時

脈衝輸出 1 原點檢測方法

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
275	08 至 11	0 Hex : 方法 0 (原點檢測方法 0) 1 Hex : 方法 1 (原點檢測方法 1) 2 Hex : 方法 2 (原點檢測方法 2)	0 Hex	指定用於脈衝輸出 1 的原點檢測方法。	---	開始動作時

脈衝輸出 1 原點搜尋操作設定

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
275	04 至 07	0 Hex : 反向 1 (反轉模式 1) 1 Hex : 反向 2 (反轉模式 2)	0 Hex	指定原點搜尋操作用於脈衝輸出 1。	---	開始動作時

脈衝輸出 1 原點搜尋操作模式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
275	00 至 03	0 Hex : 模式 0 1 Hex : 模式 1 2 Hex : 模式 2	0 Hex	指定原點搜尋模式用於脈衝輸出 1。	---	開始動作時

脈衝輸出 1 原點輸入訊號型式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
286	08 至 11	0 Hex : NC 1 Hex : NO	0 Hex	指定脈衝輸出 1 的原點輸入訊號為常閉或常開。	---	開始動作時

脈衝輸出 1 近傍原點輸入訊號型式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
286	04 至 07	0 Hex : NC 1 Hex : NO	0 Hex	指定脈衝輸出 1 的近傍原點輸入訊號為常閉或常開。	---	開始動作時

脈衝輸出 1 限動輸入訊號型式

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
286	00 至 03	0 Hex : NC 1 Hex : NO	0 Hex	指定脈衝輸出 1 的極限輸入訊號為常閉或常開。	---	開始動作時

脈衝輸出 1 原點搜尋 / 原點復歸起動速度

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
276	00 至 15	00000000 至 000186A0 Hex (參見附註)	00000000 Hex	指定脈衝輸出 1 原點搜尋與原點返回操作的起動速度 (0 至 100,000 pps)。	---	開始動作時
277	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 1 原點搜尋高速

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
278	00 至 15	00000001 至 000186A0 Hex (參見附註)	00000001 Hex	指定脈衝輸出 1 原點搜尋操作的高速設定 (1 至 100,000 pps)。	---	開始動作時
279	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 1 原點搜尋近接速度

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
280	00 至 15	00000001 至 000186A0 Hex (參見附註)	00000000 Hex	指定脈衝輸出 1 原點搜尋操作的近接速度設定 (1 至 100,000 pps)。	---	開始動作時
281	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 1 搜尋補償值 1 (原點補償)

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
282	00 至 15	80000000 至 7FFFFFFF Hex(參見附註)	---	設定脈衝輸出 1 原點補償 (-2,147,483,648 至 2,147,483,647)。	---	開始動作時
283	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 1 原點搜尋加速比率

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
284	00 至 15	0001 至 07D0 Hex (參見附註)	---	設定原點搜尋加速率用於脈衝輸出 1 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 1 原點搜尋減速比率

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
285	00 至 15	0001 至 07D0 Hex (參見附註)	---	設定脈衝輸出 1 的原點搜尋減速率 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

脈衝輸出 1 定位監控時間

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
287	00 至 15	0000 至 270F Hex (參見附註)	0000 Hex	指定脈衝輸出 1 的定位監控時間 (0 至 9,999 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

5-1-2-11 在定義原點 1/2 標籤上的原點返回欄位（在 CX-P 軟體上）（原點返回）

以下各表顯示定義原點 1/2 標籤的 CX-P 軟體的定義原點操作設定欄位中原點返回功能的各項設定。這些設定適用於配備內藏 I/O 的 CJ1M CPU 模組。

原點搜尋 0 設定（在定義原點 1 標籤上的 CX-P 軟體的原點返回欄位）

速度（脈衝輸出 0 原點返回的目標速度）

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
270	00 至 15	00000001 至 000186A0 Hex (參見附註)	00000000 Hex	指定目標速度 (1 至 100,000 pps) 用於脈衝輸出 0 原點返回操作。	---	開始動作時
271	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

加速比率（脈衝輸出 0 原點復歸加速比率）

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
272	00 至 15	0001 至 07D0 Hex (參見附註)	0000 Hex	設定原點返回加速率用於脈衝輸出 0 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

減速比率（脈衝輸出 0 原點復歸減速比率）

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
273	00 至 15	0001 至 07D0 Hex (參見附註)	0000 Hex	設定原點返回減速率用於脈衝輸出 0 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

原點搜尋 1 設定（在定義原點 2 標籤上的 CX-P 軟體的原點返回欄位）

速度（目標速度用於脈衝輸出 1 原點返回）

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
288	00 至 15	00000001 至 000186A0 Hex (參見附註)	00000000 Hex	指定目標速度 (1 至 100,000 pps) 用於脈衝輸出 1 原點返回操作。	---	開始動作時
289	00 至 15					

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

加速比率（脈衝輸出 1 原點復歸加速比率）

程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
290	00 至 15	0001 至 07D0 Hex (參見附註)	0000 Hex	設定脈衝輸出 1 的原點返回加速率 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 輸入設定。

減速（脈衝輸出 1 原點復歸減速比率）

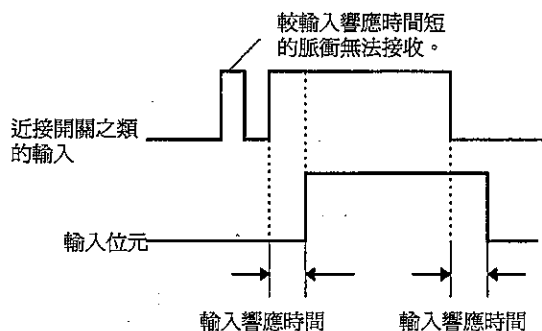
程式書寫器設定位址		設定	預設值	功能	相關的輔助區域旗標 / 位元	由 CPU 模組讀取設定的時間
字組	位元					
291	00 至 15	0001 至 07D0 Hex (參見附註)	0000 Hex	設定脈衝輸出 1 的原點返回減速率 (1 至 2,000 脈衝 / 4 ms)。	---	開始動作時

附註 當以 CX-P 軟體來進行設定時，是以 BCD 數輸入設定。

5-2 PLC 系統設定之解說

基本 I/O 模組輸入回應時間

基本 I/O 模組的輸入回應時間可以以 Rack 數或插槽數來設定。增加這個值會減少顫動與雜訊的影響。減少這個值可以接收較短的輸入脈衝，(但不能把 ON 回應時間或 OFF 回應時間設定為小於循環時間)。



輸入回應時間的預設設定為 8 ms 且設定範圍為 0 至 32 ms。當輸入回應時間設定為 0 ms 時，唯一的延遲將會是模組內部元件的延遲。

當 PLC 電源打開時，輸入回應時間設定會傳送至基本 I/O 模組。

當模組的設定改變時，它們是儲存於 A220 至 A259 (基本 I/O 模組真正的輸入回應時間)。當 PLC 設定中的設定以在 PROGRAM 模式中的 PLC 加以改變時，PLC 設定將與模組中真正的設定不同。在這個情形下，可以檢查 A220 至 A259 中的值以查看真正設定在模組中的輸入回應時間。

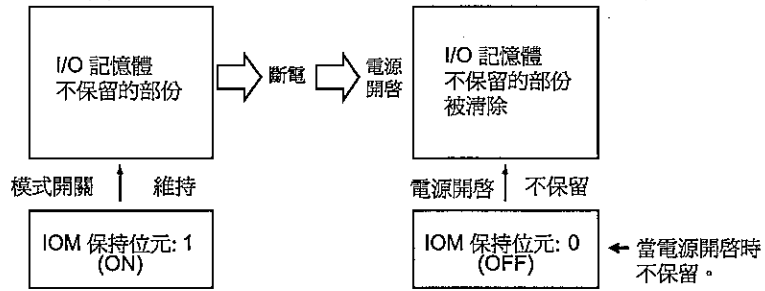
起動時 IOM 保持位元狀態

當 CPU 模組的操作模式在 PROGRAM 模式與 RUN/MONITOR 模式間切換時，IOM 保持位元 (A50012) 可以變為 ON 以保持 I/O 記憶體中所有的資料。當 PLC 電源打開時，IOM 保持位元本身將會被清除 (OFF)，除非它有受到 PLC 設定的保護。

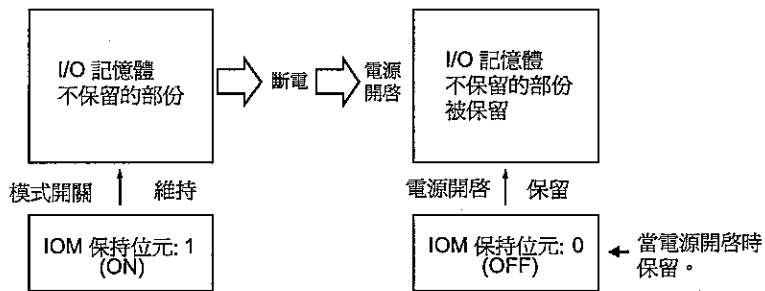
若起動時 IOM 保持位元狀態設定為 ON，當 PLC 電源打開時，IOM 保持位元的狀態將受到保護。若這項設定為 ON 且 IOM 保持位元本身也為 ON，則當 PLC 電源打開時，I/O 記憶體中的資料將會維持。

附註 若備份電池失效或失去連接，則不論這項設定為何，IOM 保持位元都將會被清除。

OFF (0): 起動時清除 IOM 保持位元



ON (1): 起動時保護 IOM 保持位元

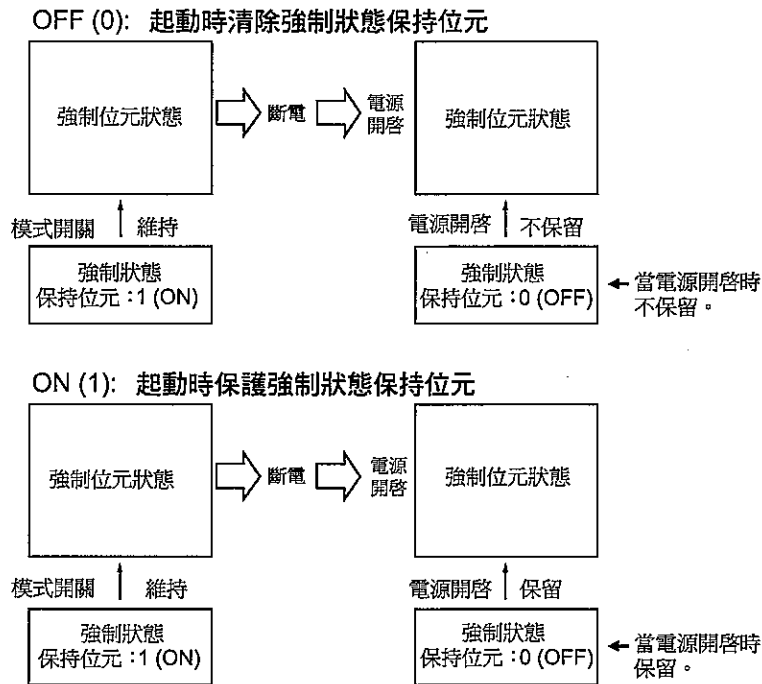


起動時的強制狀態保持位元

強制狀態保持位元 (A50013) 可以變為 ON 以便在當 CPU 模組的操作模式在 PROGRAM 模式與 RUN/MONITOR 模式間切換時，能夠讓所有被強制設定或強制重置的位元維持在強制狀態。當 PLC 電源打開時，強制狀態保持位元本身將會被清除 (OFF)，除非它被以這個 PLC 設定加以保護。

若起動時強制狀態保持位元設定為 ON，當 PLC 電源打開時，強制狀態保持位元的狀態將會被保護。若這項設定為 ON 且強制狀態保持位元本身也為 ON，當 PLC 電源打開時，所有的強制設定與強制重置位元將會維持它們的強制狀態。

附註 若備份電池失效或失去連接，則不論這項設定為何，IOM 保持位元都將會被清除。

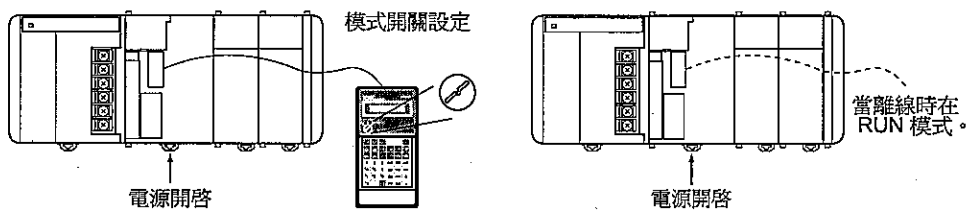


起動模式設定

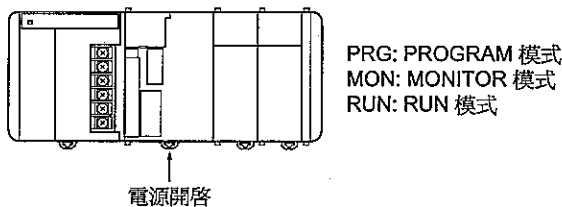
這項設定決定起動模式為程式書寫器的模式開關所設定的模式還是此處 PLC 設定所設定的模式。

附註 若這項設定指定為程式書寫器的模式開關 (0) 但卻沒有連接程式書寫器，則起動時 CPU 模組將會自動進入 RUN 模式。(這與 CS 系列 CPU 模組的預設值操作不同。)

PRCN: 程式書寫器的模式開關



其他: PC 設定的起動模式設定

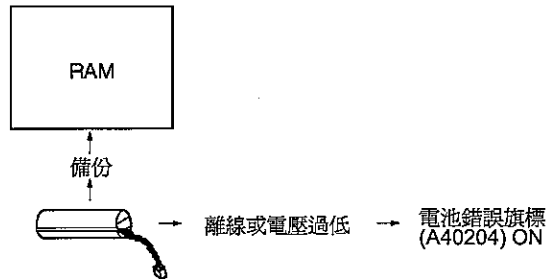


檢測電池低電量

這項設定決定是否檢測到 CPU 模組電池錯誤。設定 PLC 為當使用無電池操作時不檢測電池錯誤。

若設定為檢測錯誤 (0) 且檢測到電池錯誤，電池錯誤旗標 (A40204) 將會變為 ON。

1. CPU 模組中 DM、EM、與 HR 區域的內容不是備份到快閃記憶體；它們只能以電池備份。若電池電壓下降，則資料可能會遺失。在程式中使用電池錯誤旗標 (A40204) 以提供對策，以便在電池電壓下降時重新初始化資料或採取其他的措施。
2. 當電池未連線或其電壓低於可容許的最小值時，將會檢測到電池錯誤。



檢測中斷工作單錯誤

若設定為檢測錯誤 (0)，在以下的狀況中將會檢測出中斷工作單錯誤：

- 在中斷工作單中執行 IORF(097) 以讓一特殊 I/O 模組的 I/O 在循環更新期間更新時能夠更新該模組的 I/O。

EM 檔案記憶體設定 (僅 CJ1 與 CJ1-H CPU 模組)

這些設定用來把部份的 EM 區域轉換至檔案記憶體。

程式書寫器

特定的 EM bank 與其後續的 bank 將被指定為檔案記憶體。使用程式書寫器改變這些設定無法格式化這些特定的 EM bank；必須要在改變這些 PLC 設定後以規劃裝置來使這些 EM bank 格式化。當以程式書寫器格式化 EM bank 時，請參見 *程式書寫器操作手冊 (W341-E1-1)* 中的 7-2 記憶卡 格式。

CX-P 軟體

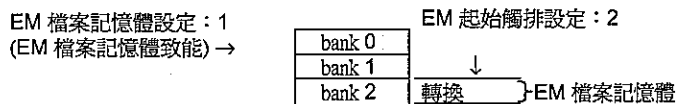
在傳送 PLC 設定時，當檔案記憶體轉換與要轉換的 bank 數被指定的情形下，可以用 CX-P 軟體將檔案記憶體格式化。(除非 EM bank 已經在 PLC 設定中被指定為檔案記憶體，否則它們不能被格式化為檔案記憶體。)

一旦部份的 EM 區域已經被格式化為檔案記憶體，可以藉把這些 PLC 設定改回它們原先的值並以規劃裝置將 EM bank 反格式化來把它轉換為正常的 EM 區域使用。

附註

1. 真正的起動檔案記憶體 bank 是儲存於 A344 (EM 檔案記憶體起動 bank)。當 PLC 設定中的這項設定被改變而 EM 區域尚未格式化時，PLC 設定將會與 EM 區域中真正的檔案記憶體設定不同。在這個情形下，可以檢查 A344 中的值以查看檔案記憶體真正的設定。
2. 若現在的 EM bank 為正在轉換到檔案記憶體的 bank 之一，則 EM 區域不能夠格式化。

以下的例子說明 EM bank 2 如何轉換到檔案記憶體。



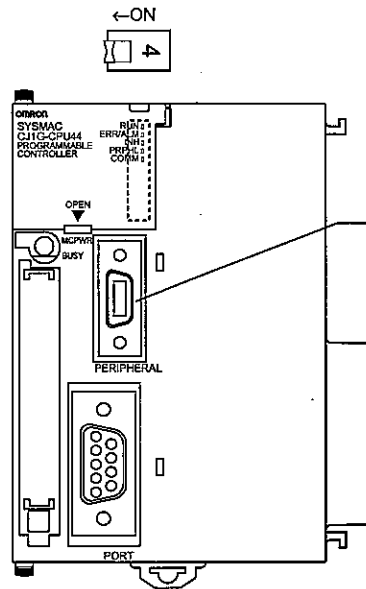
週邊埠設定

只有在 CPU 模組面板之指撥開關的接腳 4 為 ON 時這些設定才是有效的。

週邊埠的預設設定為：Host Link 模式、1 起動位元、7 資料位元、偶同位、2 停止位元與 9,600 bps 的鮑率。當您要改變這些設定時，把週邊埠設定設定在 PLC 設定。

附註

當 CPU 模組面板之指撥開關的接腳 4 為 OFF 時，CPU 模組會自動檢測所連接規劃裝置 (包括程式書寫器) 的通訊參數。所檢測到的參數會儲存於 PLC 設定中。



週邊埠通訊設定在指撥開關的針腳 4 為 ON 時:

預設設定:
主鏈結模式、1 起動位元、7 資料位元、偶同位、
2 停止位元、與速率 9,600 bps。

使用者定義設定:
設定通訊模式 (主鏈結、NT 鏈結、或週邊匯流排)
與速率等其他的設定。

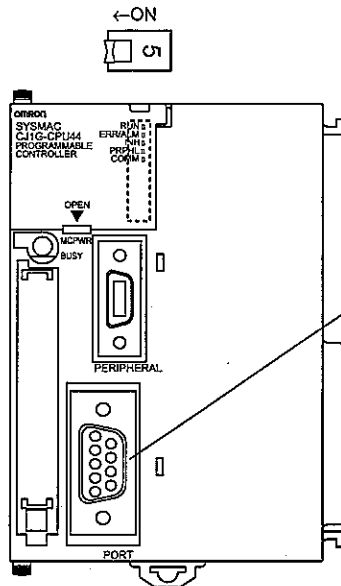
RS-232C 埠設定

這些設定僅在 CPU 模組面板之指撥開關的接腳 5 為 OFF 時方為有效的。

RS-232C 埠預設的設定為：Host Link 模式、1 起動位元、7 資料位元、偶同位、
2 停止位元與 9,600 bps 的速率。當您要改變這些設定時，把 RS-232C 埠設定
在 PLC 設定。當選擇 no-protocol 模式時，請指明框架格式。

RS-232C 埠設定也可以用 STUP(237) 改變。當執行 STUP(237) 時，RS-232C
埠設定改變旗標 (A61902) 變為 ON，當 RS-232C 埠設定改變時則會變為 OFF。

附註 當 CPU 模組面板之指撥開關的接腳 5 設定為 ON 時，CPU 模組會自動來測連接至 RS-232C 埠的規劃裝置（包括程式書寫器）之通訊參數。那些自動檢測的參數並不是儲存於 PLC 設定中。



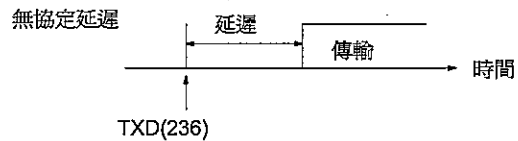
RS-232C 埠通訊設定在指撥開關的針腳 5 為 OFF 時:

預設設定:
Host Link 模式、1 起動位元、7 資料位元、偶同位、
2 停止位元、與速率 9,600 bps。

使用者定義設定:
設定通訊模式 (Host Link、NT 鏈結、no-protocol*、或週邊
匯流排) 與速率等其他的設定。

*有關無協定模式的詳情請參見附註 1 與 2。

附註 1. 在 no-protocol 模式中設定 no-protocol 傳輸延遲 (位址 162)。這個延遲的操作如下圖所示。



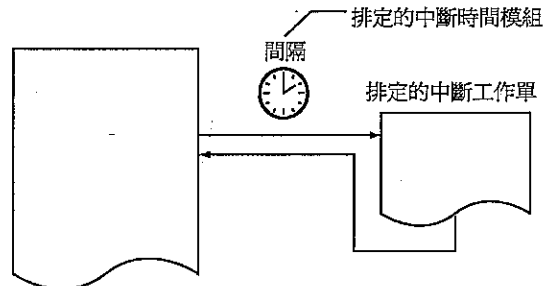
2. 下表顯示在 no-protocol 模式中可設定為傳送與接收的訊息格式。這個格式由設定的起始碼 (ST) 與結束碼 (ED) 設定來決定。(在 no-protocol 模式中可接收 1 至 256 位元組。)

起始碼設定	結束碼設定		
	無	是	CR+LF
無	DATA	DATA+ED	DATA+CR+LF
是	ST+DATA	ST+DATA+ED	ST+DATA+CR+LF

排定的中斷時間模組

這項設定決定排定的中斷間隔設定之時間模組。以 MSKS(690) 由程式中來設定排定的中斷間隔。

附註 當 CPU 模組在 RUN 或 MONITOR 模式時，這項設定不能改變。



指令錯誤操作

這項設定決定指令執行錯誤要視為非嚴重錯誤 (0) 還是嚴重錯誤 (1)。若以下的旗標變為 ON，則將產生程式錯誤。

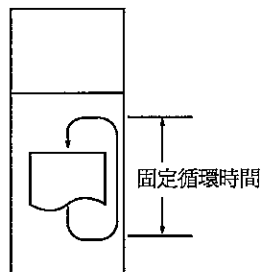
指令錯誤旗標	位址	原因
指令處理錯誤旗標	A29508	ER 旗標變為 ON。
間接 DM/EM BCD 錯誤旗標	A29509	當需要 BCD 作間接定址時，DM/EM 字組的內容不是 BCD。
非法存取錯誤旗標	A29510	意圖存取程式中禁止部份的記憶體。

若這項設定為 OFF (0)，在發生這些錯誤之一後 PLC 操作將會繼續。
 若這項設定為 ON (1)，在發生這些錯誤之一後 PLC 操作將會停止。

最小的循環時間

設定最小的循環時間為一個非零的值以消除 I/O 回應的不一致性。這項設定只有在真正的循環時間小於設定的最小的循環時間時才是有效的。若真正的循環時間較設定的最小的循環時間為長，則真正的循環時間將維持不變。

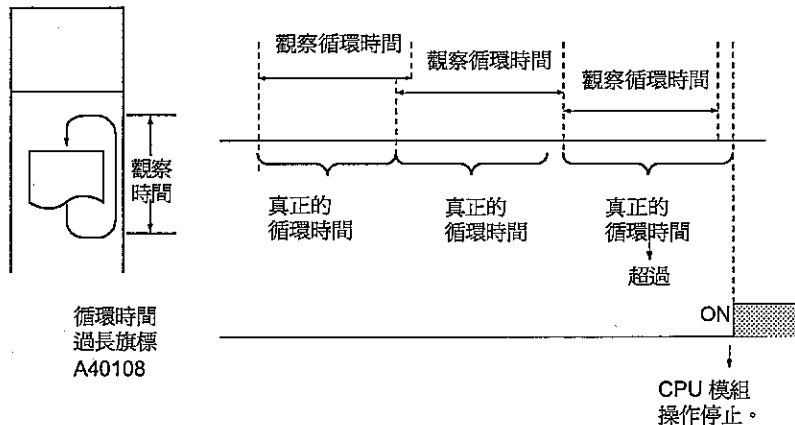
附註 當 CPU 模組在 RUN 或 MONITOR 模式時，最小循環時間的設定不能改變。



觀察循環時間

若循環時間超過觀察 (最大的) 循環時間設定，循環時間過長旗標 (A40108) 將會變為 ON 且 PLC 操作將會停止。若一般循環時間超過 1 s 的觀察循環時間設定預設值，則這項設定必需改變。

附註 當 CPU 模組在 RUN 或 MONITOR 模式時，不能改變觀察循環時間。



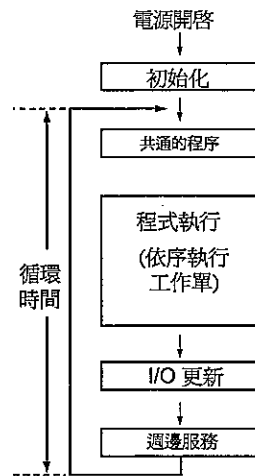
附註 預設的觀察循環時間為 1 s (1,000 ms)。

固定週邊服務時間

這項設定決定以下的程序的週邊服務是要以預設的設定 (循環時間的 4%) 進行還是在一固定服務時間內一起進行。

- 必要時與特殊 I/O 模組交換資料
- 必要時與 CPU 匯流排模組交換資料
- 與週邊埠交換資料
- 與串列通訊埠交換資料
- 服務檔案存取操作 (記憶卡)

週邊服務是在循環的結束時進行的，就在 I/O 更新之後。



下表為週邊服務時間的分類。

週邊服務時間	預設值	設定範圍
特殊 I/O 模組的事件服務時間	前一循環的循環時間之 4%	一致的服務時間，單位為 ms：0.0 至 25.5 ms 以 0.1 ms 為單位
CPU 匯流排模組的事件服務時間	同上	
週邊埠的事件服務時間	同上	
RS-232C 埠的事件服務時間	同上	
記憶卡的檔案存取服務時間	同上	

每項服務程序的預設值為前一循環的循環時間之 4%。

一般而言，會建議使用預設值。只有在週邊服務因每個服務程序都要橫跨數個循環而延遲時才需要設定一致的服務時間。

附註

1. 當週邊服務時間設定得較預設值長時，循環時間也將會變長。
2. 當 CPU 模組在 RUN 模式或 MONITOR 模式時不能更改固定的週邊服務時間之設定。
3. 使用週邊服務優先模式以指定服務週邊在程式執行期間的優先序。

電源 OFF 中斷工作單

這項設定決定檢測到電源中斷時是否要執行電源 OFF 中斷工作單。(當這項設定是設定為 0 時，在檢測到電源中斷時將只會停止常規的程式。)

當電源保持時間(電源中斷後之處理時間 + 電源 OFF 檢測延遲時間)過了之後，電源 OFF 中斷工作單將會停止。電源保持時間最多為 10 ms。

當電源 OFF 檢測延遲時間已設定，請確定電源 OFF 中斷工作單在可用的時間內可以執行(10 ms 的電源 OFF 檢測延遲時間)。

附註

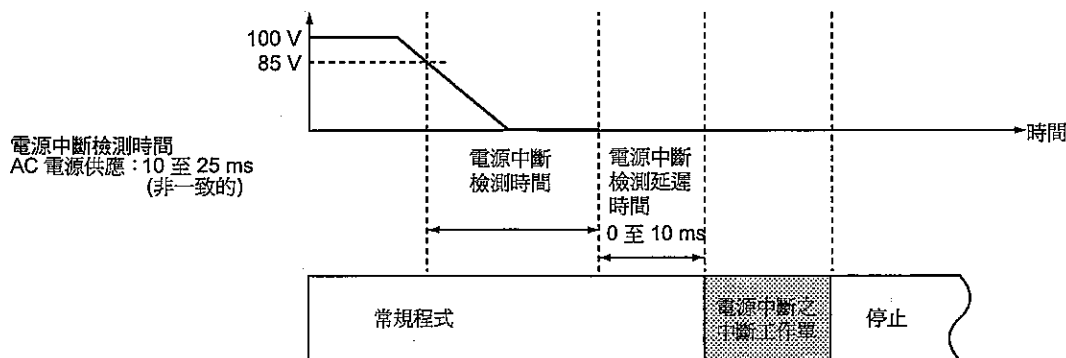
當 CPU 模組在 RUN 模式或 MONITOR 模式時，無法改變電源 OFF 中斷工作單設定。

電源 OFF 檢測延遲時間

這項設定決定由檢測到電源中斷(約在電源電壓降到額定值的 85% 後)到確定電源中斷且常規程式停止之間的延遲要多久。這項設定在 0 與 10 ms 之間。

內部的 5 V DC 電源供應器在電源中斷檢測時時後至降到 0 V DC 最多要花費 10 ms 的時間。讓這個時間延長到當電源供應器不良導致電源暫時中斷而使 PLC 停止操作所花費的時間。

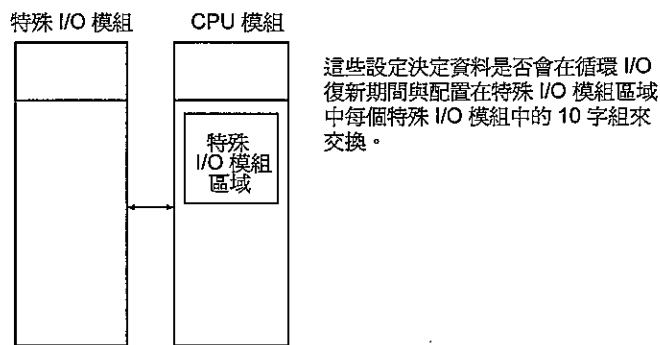
附註 當 CPU 模組在 RUN 模式或 MONITOR 模式時，電源 OFF 檢測延遲時間設定不能改變。



附註 電源 OFF 中斷工作單的執行時間必須小於可用的最大時間，即：10 ms -- 電源 OFF 檢測延遲時間。

特殊 I/O 模組循環更新

當一個特殊 I/O 模組將在中斷工作單中以 IORF(097) 進行更新時，永遠要解除有這項設定之模組的循環更新功能。若於正常的 I/O 更新期間在中斷工作單中執行 IORF(097)，則不僅無法達成所預期的結果，中斷工作單錯誤旗標 (A40213) 也將會變為 ON。



附註 要解除特殊 I/O 模組的循環更新的功能前，請確認該模組的 I/O 在操作期間至少每 11 秒由程式中的 IORF(097) 更新一次。若沒有每 11 秒更新一次，在特殊 I/O 模中將會發生 CPU 模組服務監控錯誤。

第 6 章 I/O 配置與資料交換

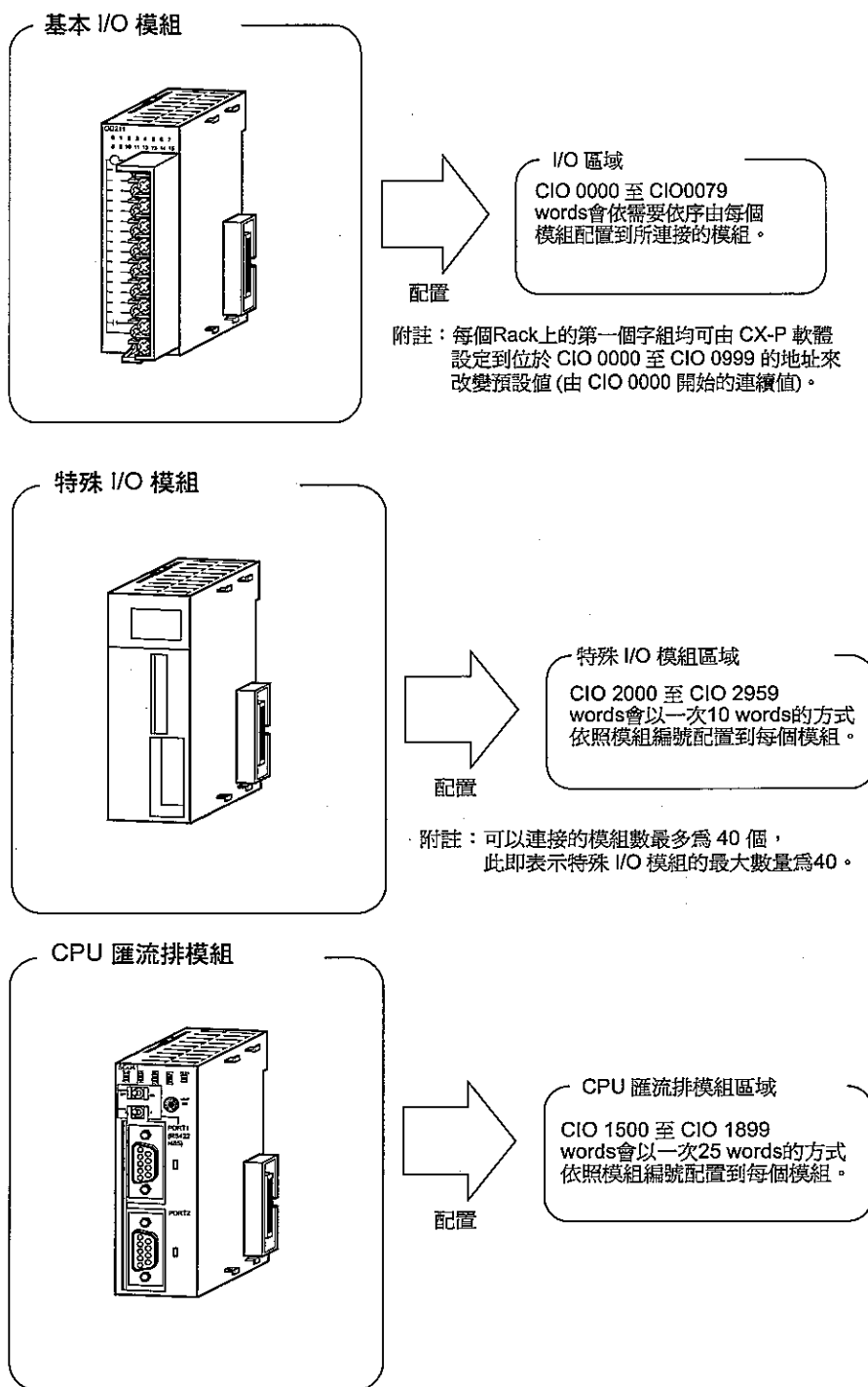
本章描述基本 I/O 模組與 CPU 匯流排模組的 I/O 配置及以 CPU 匯流排模組進行資料交換。

6-1	I/O 配置	80
6-1-1	基本 I/O 模組	80
6-1-2	基本 I/O 模組的 I/O 配置	81
6-1-3	I/O 配置至特殊 I/O 模組	83
6-1-4	I/O 配置至 CPU 匯流排模組	83
6-1-5	內藏 I/O 的資料區域配置 (僅 CJ1M CPU 模組)	84
6-2	以 CPU 匯流排模組進行資料交換	85
6-2-1	特殊 I/O 模組	85
6-2-2	CPU 匯流排模組	87

6-1 I/O 配置

在 CJ 系列 PLC 中，部份的 I/O 記憶體是配置到每個模組。記憶體分別配置到基本 I/O 模組、特殊 I/O 模組、與 CJ 系列 CPU BUS(匯流排) 模組。

6-1-1 基本 I/O 模組

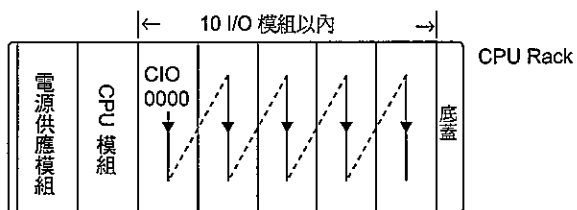


6-1-2 基本 I/O 模組的 I/O 配置

CJ 系列基本 I/O 模組為 I/O 區域 (CIO 0000 至 CIO 0079) 中的配置 words，並可安裝到 CPU rack 或擴充 rack。

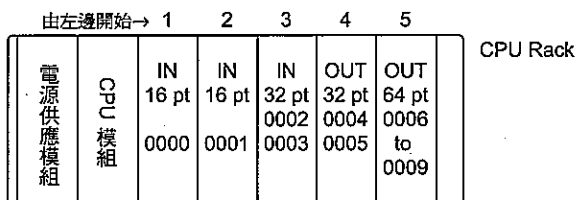
CPU Rack 上的基本 I/O 模組 CPU Rack 上的基本 I/O 模組是從最靠近 CPU 模組開始由左至右來配置 words。每個模組均會依需要儘量配置 words。

附註 有 1 至 16 I/O 點的模組及有 17 至 32 I/O 點均配置 32 位元。例如：一個 8 點的模組 16 位元 (1 word) 而 word 的位元 00 至 07 是配置給這個模組的 8 個點。



範例

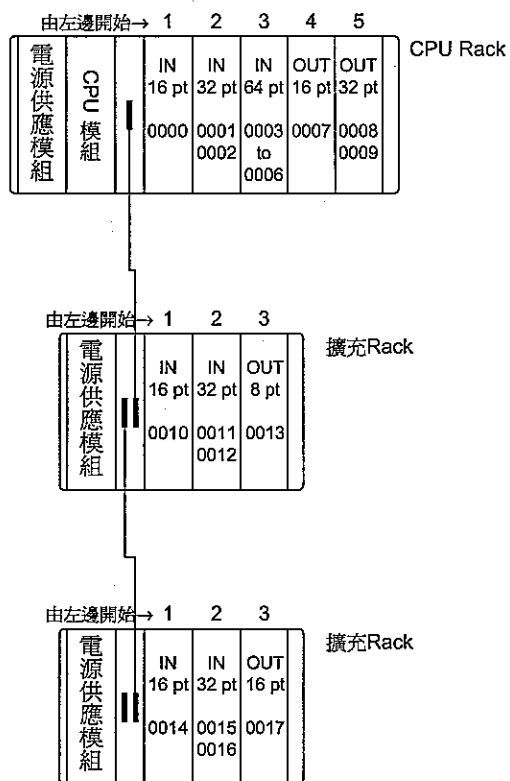
以下的例子顯示配置到 CPU Rack 上的 5 個基本 I/O 模組之 I/O 配置。



CPU 模組右方的位置	模組	所需 words	配置的 words
1	CJ1W-ID211 16 點 DC 輸入模組	1	CIO 0000
2	CJ1W-ID211 16 點 DC 輸入模組	1	CIO 0001
3	CJ1W-ID231 32 點 DC 輸入模組	2	CIO 0002 與 CIO 0003
4	CJ1W-OD231 32 點電晶體輸出模組	2	CIO 0004 與 CIO 0005
5	CJ1W-OD261 64 點電晶體輸出模組	4	CIO 0006 至 CIO 0009

範例

以下的例子顯示 CPU rack 上與兩個 CJ 系列擴充 rack 上的基本 I/O 模組之 I/O 配置。



Rack	CPU 模組右方的位置	模組	所需 words	配置的 words
CPU Rack	1	CJ1W-ID211 16 點 DC 輸入模組	1	CIO 0000
	2	CJ1W-ID231 3 2點 DC 輸入模組	2	CIO 0001 與 CIO0002
	3	CJ1W-ID261 6 4點 DC 輸入模組	4	CIO 0003 至 CIO0006
	4	CJ1W-OD211 1 6點電晶體輸出模組	1	CIO 0007
	5	CJ1W-OD231 3 2點電晶體輸出模組	2	CIO 0008 與 CIO0009
擴充 Rack	1	CJ1W-ID211 16 點 DC 輸入模組	1	CIO 0010
	2	CJ1W-ID231 3 2點 DC 輸入模組	2	CIO 0011 與 CIO 0012
	3	CJ1W-OC201 8 點繼電器輸出模組	1	CIO 0013
擴充 Rack	1	CJ1W-ID211 16 點 DC 輸入模組	1	CIO 0014
	2	CJ1W-ID231 3 2點 DC 輸入模組	1	CIO 0015 與 CIO 0016
	3	CJ1W-OC211 1 6點繼電器輸出模組	1	CIO 0017

6-1-3 I/O 配置至特殊 I/O 模組

每個 CJ 系列特殊 I/O 模組會依設定在模組上的模組數在特殊 I/O 模組區域 (CIO 2000 至 CIO 2959) 中各配置 10 個 words。特殊 I/O 模組可以安裝至 CJ 系列 CPU Rack 或 CJ 系列擴充 Rack。

words 配置

下表顯示在特殊 I/O 模組區域中的那些 words 會配置到每個模組。

模組數	配置的 words
0	CIO 2000 至 CIO 2009
1	CIO 2010 至 CIO 2019
2	CIO 2020 至 CIO 2029
:	:
15	CIO 2150 至 CIO 2159
:	:
:	:
95	CIO 2950 至 CIO 2959

特殊 I/O 模組在 I/O 配置至基本 I/O 模組期間是被被忽略的。包含特殊 I/O 模組的位置在 I/O 區域中將不會配置任何 words。

範例

以下的例子顯示 I/O words 配置至 CPU Rack 上的基本 I/O 模組與特殊 I/O 模組。

插槽	模組	所需 words	配置的 words	模組數	群組
0	CJ1W-ID211 1 6點 DC 輸入模組	1	CIO 0000	---	基本 I/O 模組
1	CJ1W-AD081 類比 輸入模組	10	CIO 2000 至 CIO 2009	0	特殊 I/O 模組
2	CJ1W-OD211 1 6點電晶體輸出模組	1	CIO 0001	---	基本 I/O 模組
3	CJ1W-TC001 溫度控制模組	20	CIO 2010 至 CIO 2029	1	特殊 I/O 模組
4	CJ1W-OD231 3 2點電晶體輸出模組	2	CIO 0002 與 CIO 0003	---	基本 I/O 模組

	0	1	2	3	4
電源供應模組					
CPU 模組					
IN 16 pt CIO 0000					
特殊 I/O 模組 CIO 2000 至 2009					
OUT 16 pt CIO 0001					
特殊 I/O 模組 CIO 2010 至 2019					
OUT 32 pt CIO 0002 CIO 0003					

6-1-4 I/O 配置至 CPU 匯流排模組

每個 CJ 系列 CPU 匯流排模組會依設定在模組上的模組數在 CPU 匯流排模組區域 (CIO 1500 至 CIO 1899) 中各配置 25 個 words。CJ 系列 CPU 匯流排模組可以安裝至 CJ 系列 CPU Rack 或 CJ 系列擴充 Rack。

words 配置

下表顯示在 CJ 系列 CPU 匯流排模組區域中的那些 words 會配置到每個模組。

模組數	配置的 words
0	CIO 1500 至 CIO 1524
1	CIO 1525 至 CIO 1549
2	CIO 1550 至 CIO 1574
:	:
15	CIO 1875 至 CIO 1899

CPU 匯流排模組在 I/O 配置至基本 I/O 模組期間是被被忽略的。包含 CJ 系列 CPU 匯流排模組的位置在 I/O 區域中將不會配置任何 words。

範例

以下的例子顯示 I/O words 配置至 CPU Rack 上的基本 I/O 模組、特殊 I/O 模組、與 CPU 匯流排模組。

	0	1	2	3	4	
電源供應模組	CPU 模組	IN 16 pt CIO 0000	特殊 I/O 模組 CIO 2000 至 2009	特殊匯流排模組 CIO 1500 至 1524	OUT 16 pt CIO 0001	特殊匯流排模組 CIO 1525 至 1549

插槽	模組	所需 words	配置的 words	模組數	群組
0	CJ1W-ID211 16點 DC 輸入模組	1	CIO 0000	---	基本 I/O 模組
1	CJ1W-AD081 類比 輸入模組	10	CIO 2000 至 CIO 2009	0	特殊 I/O 模組
2	CJ1W-SCU41 串列通訊模組	25	CIO 1500 至 CIO 1524	0	CPU 匯流排模組
3	CJ1W-OD211 16點電晶體輸出模組	1	CIO 0001	---	基本 I/O 模組
4	CJ1W-CLK21 Controller Link 模組	25	CIO 1525 至 CIO 1549	1	CPU 匯流排模組

6-1-5 內藏 I/O 的資料區域配置 (僅 CJ1M CPU 模組)

I/O 碼	位址	CIO 2960										CIO 2961						
		位元	00	01	02	03	04	05	06	07	08	09	00	01	02	03	04	05
輸入	汎用輸入	汎用輸入 0	汎用輸入 1	汎用輸入 2	汎用輸入 3	汎用輸入 4	汎用輸入 5	汎用輸入 6	汎用輸入 7	汎用輸入 8	汎用輸入 9	---	---	---	---	---	---	
	中斷輸入	中斷輸入 0	中斷輸入 1	中斷輸入 2	中斷輸入 3	---	---	---	---	---	---	---	---	---	---	---	---	
	快速回應輸入	快速回應輸入 0	快速回應輸入 1	快速回應輸入 2	快速回應輸入 3	---	---	---	---	---	---	---	---	---	---	---	---	
	高速計數器	---	---	高速計數器 1 (乙相/置置)	高速計數器 0 (乙相/置置)	---	---	高速計數器 1 (A 相加或輸入)	高速計數器 0 (B 相減或輸入)	高速計數器 1 (A 相加或輸入)	高速計數器 0 (B 相減或輸入)	---	---	---	---	---	---	
輸出	汎用輸出	---	---	---	---	---	---	---	---	---	---	汎用輸出 0	汎用輸出 1	汎用輸出 2	汎用輸出 3	汎用輸出 4	汎用輸出 5	
	脈衝輸出	CW/CCW 輸出	---	---	---	---	---	---	---	---	---	脈衝輸出 0 (CW)	脈衝輸出 0 (CCW)	脈衝輸出 1 (CW)	脈衝輸出 1 (CCW)	---	---	
	脈衝 + 方向輸出	---	---	---	---	---	---	---	---	---	---	脈衝輸出 0 (脈衝)	脈衝輸出 1 (脈衝)	脈衝輸出 0 (方向)	脈衝輸出 1 (方向)	---	---	
	可變功率輸出	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	PWM(89 1)輸出 0	PWM(89 1)輸出 1
原點搜尋	原點搜尋 0 (原入訊號)	原點搜尋 1 (原入訊號)	原點搜尋 0 (原入訊號)	原點搜尋 1 (原入訊號)	原點搜尋 0 (原入訊號)	原點搜尋 1 (原入訊號)	---	---	---	---	---	---	---	---	---	---	原點搜尋 0 (計量器輸出)	原點搜尋 1 (計量器輸出)

6-2 以 CPU 匯流排模組進行資料交換

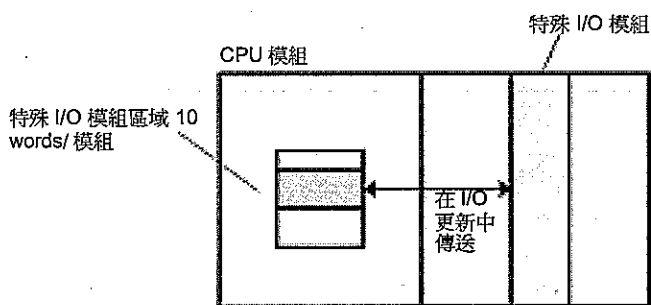
本節描述資料如何在特殊 I/O 模組或 CPU 匯流排模組與 CPU 模組之間進行交換。

6-2-1 特殊 I/O 模組

特殊 I/O 模組區域
(I/O 更新)

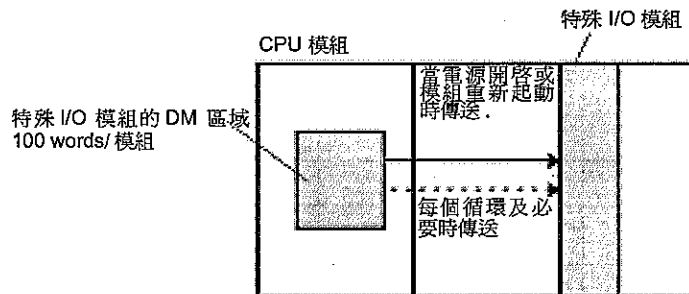
資料交換是在特殊 I/O 模組區域的 I/O 更新期間之每個循環中進行的。基本而言，會依每個特殊 I/O 模組的模組數設定各配置 10 words。有關個別特殊 I/O 模組的詳情請參見操作手冊。

特殊 I/O 模組區域之範圍為由 CIO 2000 至 CIO 2959 (10 words × 96 模組)。



DM 區域

每個特殊 I/O 模組在 D20000 至 D29599 (100 words × 96 模組) 的範圍中的 DM 區域中配置有 100 words。這 100 個 words 一般用於保持特殊 I/O 模組的初值設定。當這個區域的內容為反應系統的改變而由程式進行改變時，受影響的模組之重新起動位元必需變為 ON 以重新起動該模組。



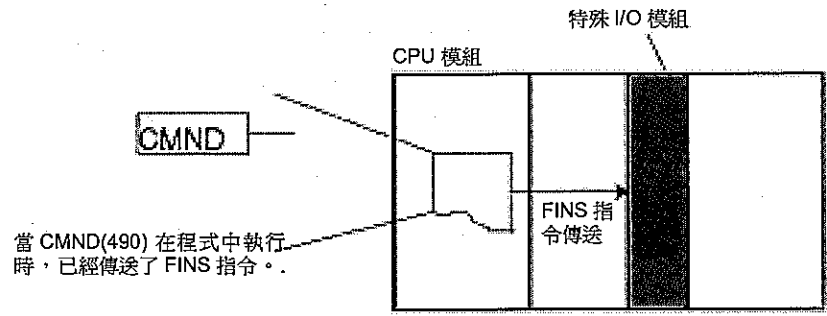
有三種時機資料會透過配置於每個模組的 words 來進行傳送。資料傳送的時機會依使用的型號而有所不同。

- 1,2,3...
1. 當 PLC 變為 ON 時傳送資料。
 2. 當模組重新起動時傳送資料。
 3. 當必要時傳送資料。

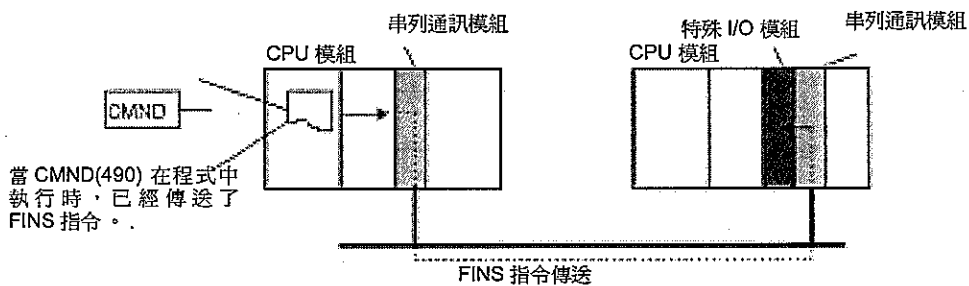
某些型號會雙向傳送資料，由 DM 區域至模組與由模組至 DM 區域。有關資料傳送的詳情請參見模組的操作手冊。

FINS 指令

CMND(490) 指令可以加到階梯程式以發佈 FINS 指令至特殊 I/O 模組。



FINS 指令不只可以傳送至本地的 PLC，還可以傳送到網路上其他 PLC 中的特殊 I/O 模組。



特殊 I/O 模組初始化

當 PLC 的電源打開時或模組的重新起動位元變為 ON 時，特殊 I/O 模組會初始化。當模組初始化時，模組的特殊 I/O 模組初始化旗標 (A33000 至 A33515) 將會為 ON。

當特殊 I/O 模組的初始化旗標為 ON 時，將不會進行 I/O 更新 (循環 I/O 更新或以 IORF(097) 更新)。

特殊 I/O 模組循環更新有無使定

特殊 I/O 模組區域 (CIO 2000 至 CIO 2959) 中的每個特殊 I/O 模組會依據每個模組面板上的模組編號設定配置 10 個 words。特殊 I/O 模組區域中的資料在 I/O 更新 (就在 END(001) 指令執行之後) 期間的每個循環都會更新於 CPU 模組。若安裝了太多個特殊 I/O 模組，則 I/O 更新可能要花費很長的時間。若 I/O 更新花費太多的時間，依照 PLC 系統的設定，到指定的高機能模組及 END 指令後的 I/O 更新可以停止。(特殊 I/O 模組循環更新除能位元位於 PLC 設定位址 226 至 231。)

若 I/O 更新時間太短，模組的內部處理將無法跟上，於是特殊 I/O 模組錯誤旗標 (A40206) 將會變為 ON，且特殊 I/O 模組將無法適當地操作。在這個情形下，可以藉由在 PLC 設定中設定最小的循環時間來延伸循環時間，否則以特殊 I/O 模組進行循環 I/O 更新可能會被除能。當循環更新被除能時，特殊 I/O 模組的資料可以在程式執行期間以 IORF(097) 來進行更新。

附註

1. 若模組的 I/O 將在中斷 task 中以 IORF(097) 來更新，務必要將特殊 I/O 模組的循環更新除能。若同時進行循環更新與 IORF(097) 更新，將會發生中斷 task 錯誤 (A40213)。
2. 在將特殊 I/O 模組的循環更新除能後，務必確認該模組的 I/O 會在操作期間每 11 秒就於程式中以 IORF(097) 更新一次。若特殊 I/O 模組沒有每 11 秒更新一次，則將會在其中發生 CPU 模組服務監控錯誤。

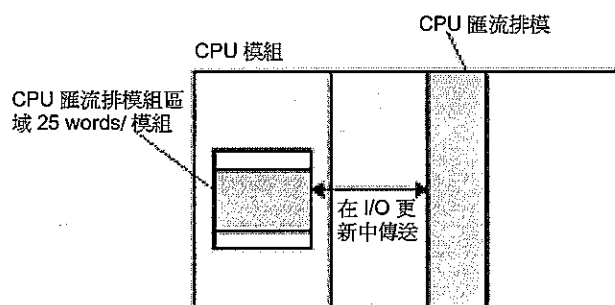
6-2-2 CPU 匯流排模組

資料可以透過 CPU 匯流排模組區域、DM 區域、或 FINS 指令在 CPU 匯流排模組與 CPU 模組之間進行交換。

CPU 匯流排模組區域 (I/O 更新)

資料交換是在 CPU 匯流排模組區域的 I/O 更新期間之每個循環中進行的。基本而言，會依每個 CPU 匯流排模組的模組數設定各配置 25 words。CPU 匯流排模組真正使用的 words 數是會改變的。

特殊 I/O 模組區域範圍自 CIO 1500 至 CIO 1899 (25 words × 16 模組)。



附註 有了 CJ1-H 與 CJ1M CPU 模組，可以在階梯程式中執行 CPU BUS I/O 更新指令 (DLNK(226)) 以更新配置至特定模組編號的 CPU 匯流排模組之 CIO 區塊 words。

DM 區域

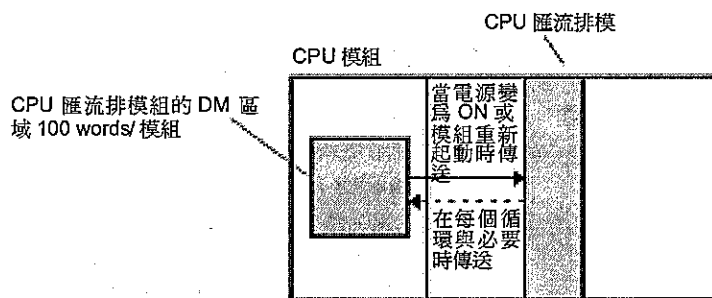
每個 CPU 匯流排模組會在 D30000 至 D31599 (100 words × 16 模組) 的範圍中配置 100 words 於 DM 區域。有三種時機資料會透過配置於每個模組的 words 來進行傳送。資料傳送的時候會依使用的型號而有所不同。

- 1,2,3...
1. 當 PLC 變為 ON 時傳送資料。
 2. 每個循環傳送資料。
 3. 必要時傳送資料。

附註 有了 CJ1-H CPU 模組，可以在階梯程式中執行 CPU BUS I/O 更新指令 (DLNK(226)) 以更新配置至特定模組編號的 CPU 匯流排模組之 DM 區塊 words。

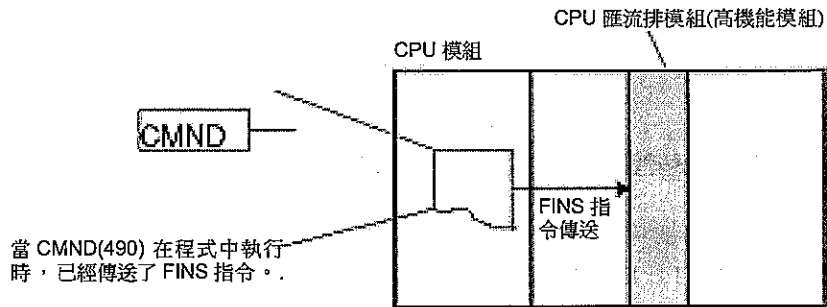
某些型號會雙向傳送資料，由 DM 區域至模組與由模組至 DM 區域。有關資料傳送的詳情請參見模組的操作手冊。

這 100 個 words 通常用來保持 CPU 匯流排模組設定的初值。當這個區域的內容為反應系統的改變而由程式加以改變時，受影響的模組之重新起動位元 (A50100 至 A50115) 必需變為 ON 以重新起動該模組。

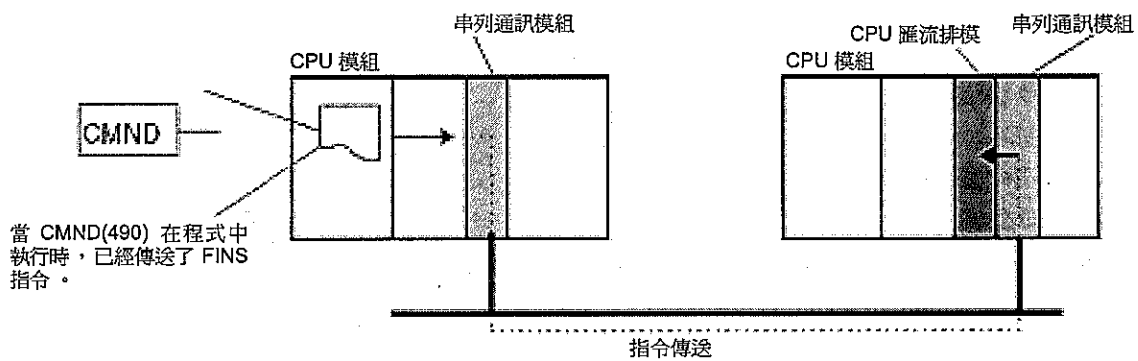


FINS 指令

CMND(490) 指令可以加到階梯程式以傳送一個 FINS 指令至 CPU 匯流排模組。



FINS 指令不只可以傳送至本地的 PLC，還可以傳送至網路中其他 PLC 的 CPU 匯流排模組。



CPU 匯流排模組初值化

CPU 匯流排模組在當 PLC 的的電源打開時或模組的重新起動位元變為 ON 時會初值化。在模組初值化時，這個模組的 CPU 匯流排模組初值化旗標 (A30200 至 A30215) 將會為 ON。

在 CPU 匯流排模組的初值化旗標為 ON 時，將不會進行循環 I/O 更新。

第 7 章 記憶體區域

本章描述 I/O 記憶體區域及參數區域的結構與功能。

7-1	I/O 記憶體區域	90
7-1-1	I/O 記憶體區域結構	90
7-1-2	資料區域概觀	92
7-1-3	資料區域性質	96
7-2	I/O 區域 (輸出 / 輸出 relay)	97
7-3	資料 Link 區域 (Data Link Area)	102
7-4	CPU 匯流排模組區域	103
7-5	特殊 I/O 模組區域	104
7-6	串列 PLC Link 區域	105
7-7	DeviceNet 區域 (DeviceNet Area)	106
7-8	內部 I/O 區域	107
7-9	保持區域 (Holding Area)	108
7-10	輔助區域 (Auxiliary Area)	109
7-11	TR (暫時記憶繼電器) 區域	129
7-12	計時器區域	130
7-13	計數器區域	131
7-14	資料記憶體 (DM) 區域	132
7-15	擴充資料記憶體 (EM) 區域	133
7-16	工作單旗標 (Task Flags)	134
7-17	條件旗標 (Condition Flags)	135
7-18	時序脈衝	136
7-19	參數區域	137
7-19-1	PLC 設定	137
7-19-2	登錄 I/O 表	137
7-19-3	路由表 (Routing Tables)	138
7-19-4	CPU 匯流排模組設定	139

7-1 I/O 記憶體區域

7-1-1 I/O 記憶體區域結構

下表顯示 I/O 記憶體區域的基本結構。

區域	大小	範圍	工作單 (task) 用法	外部 I/O 配置	位元 存取	words 存取	存取		由 CX-P 軟體改 變	起動時或 模式改變 時之狀態	強制 位元 狀態	
							讀取	寫入				
C I O 區域	I/O 區域	1,280 位 元 (80 words)	CIO 0000 至 CIO 0079 (附註 1)	由所有 的工作 單分享	基本 I/O 模 組	OK	OK	OK	OK	OK	清除	OK
	資料 Link 區 域	3,200 位 元 (200 words)	CIO 1000 至 CIO 1199		資料 Link	OK	OK	OK	OK	OK	清除	OK
	CPU 匯流排 模組區域	6,400 位 元 (400 words)	CIO 1500 至 CIO 1899		CPU 匯流 排模組	OK	OK	OK	OK	OK		OK
	特殊 I/O 模組 區域	15,360 位 元 (960 words)	CIO 2000 至 CIO 2959		特殊 I/O 模 組	OK	OK	OK	OK	OK		OK
	內藏 I/O 區 域 (僅內藏 I/ O 之 CJ1M CPU 模組)	10 位元 + 6 位元 (1 word) + 1 word)	CIO 2960 至 CIO 2961		內藏 I/O 埠	OK	OK	OK	OK	OK		OK
	串列 PLC Link 區域 (僅 CJ1M CPU 模組)	1,440 位 元 (90 words)	CIO 3100 至 CIO 3189		串列 PLC Link	OK	OK	OK	OK	OK		OK
	DeviceNet 區域	9,600 位 元 (600 words)	CIO 3200 至 CIO 3799		DeviceNet 主局 (固定 配置)	OK	OK	OK	OK	OK		OK
	內部 I/O 區域	37,504 位 元 (2,344 words) 4,800 位 元 (300 words)	CIO 1200 至 CIO 1499 CIO 3800 至 CIO 6143		—	OK	OK	OK	OK	OK		OK

區域	大小	範圍	工作單 (task) 用法	外部 I/O 配置	位元存取	words 存取	存取		由 CX-P 軟體改變	起動時或模式改變時之狀態	強制位元狀態	
							讀取	寫入				
工作區域 (Work Area)	8,192 位元 (512 words)	W000 至 W511	由所有的工作單分享	—	OK	OK	OK	OK	OK	清除	OK	
保持區域 (Holding Area)	8,192 位元 (512 words)	H000 至 H511		—	OK	OK	OK	OK	OK	維持	OK	
輔助區域 (Auxiliary Area)	15,360 位元 (960 words)	A000 至 A959		—	OK	OK	OK	A000 至 A447 否 A448 至 A959 OK	A000 至 A447 否 A448 至 A959 OK	依位址而改變	否	
TR 區域	16 位元	TR0 至 TR15		—	OK	—	OK	OK	OK	否	清除	否
DM 區域	32,768 words	D00000 至 D32767		—	否 (附註 2)	OK	OK	OK	OK	OK	維持	否
EM 區域 (附註 6.)	32,768 words 每個 bank (0 至 2, 3 以下)	E0_00000 至 E2_32767		—	否 (附註 2)	OK	OK	OK	OK	OK	維持	否
計時器完成旗標	4,096 位元	T0000 至 T4095		—	OK	—	OK	OK	OK	OK	清除	OK
計數器完成旗標	4,096 位元	C0000 至 C4095		—	OK	—	OK	OK	OK	OK	維持	OK
計時器現在值 (PV)	4,096 words	T0000 至 T4095		—	—	OK	OK	OK	OK	OK	清除	否 (附註 4)
計數器現在值 (PV)	4,096 words	C0000 至 C4095		由所有的工作單分享	—	—	OK	OK	OK	OK	維持	否 (附註 5)
工作單旗標區域 (Task Flag Area)	32 位元	TK00 至 TK31	—		OK	—	OK	否	否	否	清除	否
索引暫存器 (附註 3)	16 暫存器	IR0 至 IR15	每個工作單分別使用	—	OK	OK	僅間接定址	僅特定指令	否	清除	否	
資料暫存器 (附註 3)	16 暫存器	DR0 至 DR15		—	否	OK	OK	OK	OK	否	清除	否

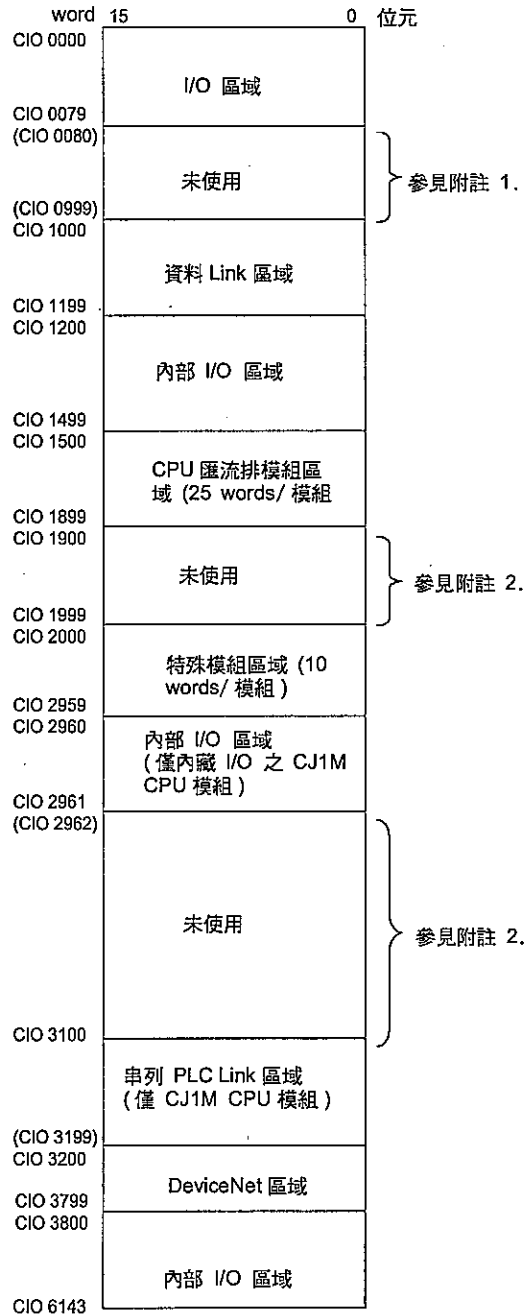
- 附註
1. 可藉由改變配置到 Racks 上的第一個 words 把 I/O 區域擴充到 CIO 0000 至 CIO 0999。
 2. 可使用 TST(350)、TSTN(351)、SET、SETB(532)、RSTB(533)、OUTB(534) 來操縱位元。
 3. 索引暫存器與資料暫存器可在工作單中個別使用，也可以由所有的工作單共同分享 (僅 CJ1-H 與 CJ1M CPU 模組)。
 4. 計時器 PV 可間接由強制設定 / 重置計時器完成旗標更新。
 5. 計數器 PV 可間接由強制設定 / 重置計數器完成旗標更新。
 6. 僅 CJ1-H 與 CJ1 CPU 模組。

7-1-2 資料區域概觀

有關 I/O 記憶體區域中資料區域的詳情描述如下。

CIO 區域

要指定 CIO 區域的位址在輸入時必須以 "CIO" 開頭來輸入。CIO 區域一般作為不同模組間 I/O 更新等資料交換之用。沒有配置到模組的 words 只能用於程式中作為工作 words 或工作位元。



- 附註 1. 可能會透過對 Racks 上的第一個 words 進行適當的設定來使用 CIO 0080 至 CIO 0999 作為 I/O words。可以使用 CX-P 軟體來進行 Racks 上第一個 words 的設定，以設定 I/O 表中的第一 Rack 位址。第一 Racks 位址的設定範圍為 CIO 0000 至 CIO 0900。

2. 標示 "未使用" 之 CIO 區域的部份可在程式中當作工作位元。未來未使用的 CIO 區域位元將可作為擴充功能。永遠要優先使用工作區域位元。

I/O 區域

這些 words 配置到基本 I/O 模組上的外部 I/O 端子。不是配置到外部 I/O 端子的 words 僅能於程式中使用。

資料 Link 區域

這些 words 用來作為 Controller Link 網路中的資料 Link。未在資料 Link 中使用的 words 僅能於程式中使用。

CPU 匯流排模組區域

這些 words 配置到 CPU 匯流排模組以傳送狀態資訊。每個模組配置 25 words，最多可以使用 16 模組 (模組編號 0 至 15)。不是由 CPU 匯流排模組使用的 words 僅能於程式中使用。

特殊 I/O 模組區域

這些 words 配置到特殊 I/O 模組。每個模組配置 10 words，最多可以使用 96 模組 (模組編號 0 至 95)。不是由特殊 I/O 模組使用的 words 僅能於程式中使用。

內藏 I/O 區域 (僅內藏 I/O 之 CJ1M CPU 模組)

這些 words 配置到 CPU 模組的內藏 I/O 埠。配置為固定不變的。這個區域僅能用於內藏 I/O 之 CJ1M CPU 模組。其他 CPU 模組只能依以下 " 內部 I/O 區域 " 所述來進行規劃。

串列 PLC Link 區域 (Serial PLC Link Area)

為串列 PLC Link 所使用的 Relay 區域和其他 PLC 進行資料 Link。串列 PLC Link 未使用之 Relay 號碼可作為僅供程式上使用之內部補助 Relay。

DeviceNet 區域 (Device Net Area)

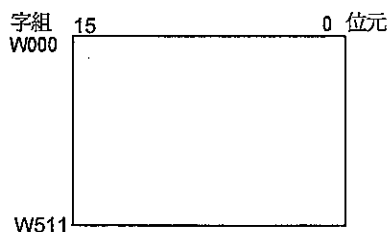
為 Device Net (CompoBus/D) 之遠端 I/O 通訊，各子局所分配的區域。其未使用之 Relay 號碼同樣地可作為內部補助 Relay 使用。

內部 I/O 區域 (內部補助 Relay)

這些 words 僅能用於程式中；它們不能用於與外部 I/O 端子進行 I/O 交換。在配置內部 I/O 區域中的 words 或 CIO 區域中其他未使用的 words 之前，務必使用工作區域 (WR) 中所提供的工作 words。這些 words 在未來版本的 CJ 系列 CPU 模組中可能會被指定新的功能，因而若 CIO 區塊 words 被用為程式中的工作 words，程式在用於新的 CJ 系列 PLC 之前可能必須加以修改。

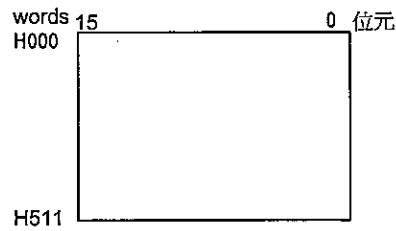
工作區域 (WR)

工作區域中的 words 僅能用於程式中；它們不能用於與外部 I/O 端子進行 I/O 交換。未來版本的 CJ 系列 PLC 不會指定新的功能至這個區域，所以請在使用 CIO 區域中的任何 words 之前，先使用這個區域作為工作 words。



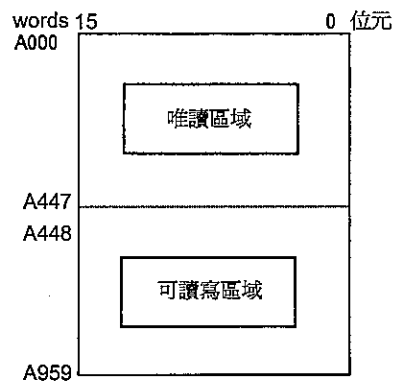
保持區域 (HR)

保持區域中的 words 僅能用於程式中。當 PLC 電源打開時，或操作模式在 PROGRAM 模式與 RUN 或 MONITOR 模式之間切換時，這些 words 會保持它們的內容。



輔助區域 (AR)

輔助區域包含用於監控與控制 PLC 操作的旗標與控制位元。這個區域分為兩個部份：A000 至 A447 是唯讀的，而 A448 至 A959 是可以讀寫的。有關輔助區域的詳情請參見 7-10 輔助區域。

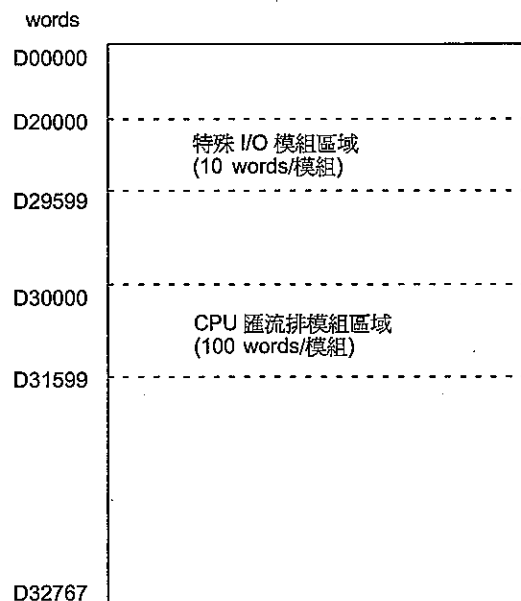


暫時記憶繼電器區域 (TR)

TR 區域包含用來記錄程式分歧之 ON/OFF 狀態的位元。TR 位元僅使用助碼。

資料記憶體區域 (DM)

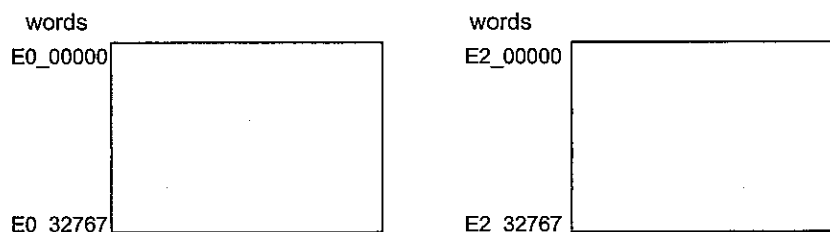
DM 區域是一個僅能在 words 模組中存取之多用途的資料區域。當 PLC 電源打開時，或操作模式在 PROGRAM 模式與 RUN 或 MONITOR 模式之間切換時，這些 words 會保持它們的內容。



擴充資料記憶體區域 (EM)
(僅 CJ1 與 CJ1-H CPU 模
組)

EM 區域是一個多用途的資料區域僅能在 words 模組中存取之多用途的資料區域。當 PLC 電源打開時，或操作模式在 PROGRAM 模式與 RUN 或 MONITOR 模式之間切換時，這些 words 會保持它們的內容。

EM 區域區分為 32,767 個稱為 bank 的 words 範圍。EM bank 的編號與 CPU 模組的型號相關，最多編到 13 banks (0 至 C)。有關於每個 CPU 模組的型號所提供的 EM bank 的編號請參見 2-1 規格。



計時器區域

有兩個計時器資料區域，分別是計時器完成旗標與計時器現在值 (PV)。最多有 4,096 計時器，可使用計時器編號 T0000 至 T4095。計時器完成旗標與 PV 之存取使用相同的編號。

計時器完成旗標

這些旗標以位元為單位來讀取。當相對應的計時器時間到 (過了設定時間) 時，完成旗標被系統變為 ON。

計時器 PV

PV 是以 words (16 位元) 為單位來讀寫的。在計時器操作的同時，PV 會往上計數或往下計數。

計數器區域

有兩個計數器資料區域，分別是計數器完成旗標與計數器顯示值 (PV)。最多有 4,096 計數器，可使用計數器編號 C0000 至 C4095。計數器完成旗標與 PV 之存取使用相同的編號。

計數器完成旗標

這些旗標以位元為單位來讀取。當相對應的計數器數到 (達到設定的計數值) 時，完成旗標被系統變為 ON。

計數器 PV

PV 是以 words (16 位元) 為單位來讀寫的。在計數器操作的同時，PV 會往上計數或往下計數。

條件旗標

這些包括表示指令執行結果的錯誤旗標與相等旗標及常時 ON 與常時 OFF 旗標等的算術旗標。條件旗標是以標記 (符號) 而非位址來指定的。

時序脈衝 (clock pulses)

時序脈衝由 CPU 模組的內部計時器來變為 ON 與 OFF。這些位元是以標記 (符號) 而非位址來指定的。

工作單旗標區域 (TK)
(Task Flag Area)

工作單旗標範圍為 TK00 至 TK31 並對應到循環工作單 0 至 31。當相對應的循環工作單在可執行 (RUN) 狀態時，工作單旗標將為 ON；而當循環工作單還沒有被執行 (INI) 或在待命 (WAIT) 狀態時，工作單旗標將為 OFF。

索引暫存器 (IR)

這些暫存器 (IR0 至 IR15) 用來把 PLC 記憶體位址 (RAM 中的絕對記憶體位址) 儲存至 I/O 記憶體中的間接位址 words。索引暫存器可在每個工作單分別使用，或者對於 CJ1-H 或 CJ1M CPU 模組，它們可以由所有的工作單分享。

資料暫存器 (DR)

這些暫存器 (DR0 至 DR15) 與索引暫存器要一起使用。當資料暫存器在索引暫存器之前輸入時，資料暫存器的內容會加到索引暫存器中 PLC 記憶體位址以偏移該位址。資料暫存器為每個工作單個別使用，而對於 CJ1-H 或 CJ1M CPU 模，它們可以由所有的工作單分享。

7-1-3 資料區域性質

嚴重錯誤、強制設定 / 重置後內容的用法

區域		產生的嚴重錯誤				強制設定 / 強制重置功能能否使用？
		FALS(007) 的執行		其他的嚴重錯誤		
		IOM 保持位元 OFF	IOM 保持位元 ON	IOM 保持位元 OFF	IOM 保持位元 ON	
CIO 區域	I/O 區域	維持	維持	清除	維持	是
	資料 Link 區域					
	CPU 匯流排模組區域					
	特殊 I/O 模組區域					
	DeviceNet 區域					
內部 I/O 區域						
工作區域 (W)		維持	維持	清除	維持	是
保持區域 (H)		維持	維持	維持	維持	是
輔助區域 (A)		狀態依位址而改變				否
資料記憶體區域 (D)		維持	維持	維持	維持	否
擴充資料記憶體區域 (E)		維持	維持	維持	維持	否
計時器完成旗標 (T)		維持	維持	清除	維持	是
計時器 PV (T)		維持	維持	清除	維持	否
計數器完成旗標 (C)		維持	維持	維持	維持	是
計數器 PV (C)		維持	維持	維持	維持	否
工作單旗標 (TK)		清除	清除	維持	維持	否
索引暫存器 (IR)		維持	維持	清除	維持	否
資料暫存器 (DR)		維持	維持	清除	維持	否

模式改變或電源中斷後的內容

區域		模式改變 1		PLC 電源中斷再 ON 時			
				IOM 保持位元清除		IOM 保持位元保持	
		IOM 保持位元 OFF	IOM 保持位元 ON	IOM 保持位元 OFF	IOM 保持位元 ON	IOM 保持位元 OFF	IOM 保持位元 ON
CIO 區域	I/O 區域	清除	維持	清除	清除	清除	維持
	資料 Link 區域						
	CPU 匯流排模組區域						
	特殊 I/O 模組區域						
	內藏 I/O 區域 (僅內藏 I/O 之 CJ1M CPU 模組)						
	串列 PLC Link 區域 (僅 CJ1M CPU 模組)						
	DeviceNet 區域						
內部 I/O 區域							
工作區域 (W)		清除	維持	清除	清除	清除	維持
保持區域 (H)		維持	維持	維持	維持	維持	維持
輔助區域 (A)		狀態依位址而改變					
資料記憶體區域 (D)		維持	維持	維持	維持	維持	維持
延伸資料記憶體區域 (E)		維持	維持	維持	維持	維持	維持

區域	模式改變 1		PLC 電源中斷再 ON 時			
			IOM 保持位元清除		IOM 保持位元保持	
	IOM 保持位元 OFF	IOM 保持位元 ON	IOM 保持位元 OFF	IOM 保持位元 ON	IOM 保持位元 OFF	IOM 保持位元 ON
計時器完成旗標 (T)	清除	維持	清除	清除	清除	維持
計時器 PV (T)	清除	維持	清除	清除	清除	維持
計數器完成旗標 (C)	維持	維持	維持	維持	維持	維持
計數器 PV (C)	維持	維持	維持	維持	維持	維持
工作單旗標 (TK)	清除	清除	清除	清除	清除	清除
索引暫存器 (IR)	清除	維持	清除	清除	清除	維持
資料暫存器 (DR)	清除	維持	清除	清除	清除	維持

- 附註
1. 模式由 PROGRAM 變為 RUN/MONITOR，或是反過來的狀況。
 2. PLC 設定的 " 起動時之 IOM 保持位元狀態 " 設定決定在 PLC 送電時 IOM 保持位元的狀態是保持還是清除。
 3. IOM 意指 I/O 記憶體。保持位元同保持旗標。

7-2 I/O 區域 (輸出 / 輸出 kelay)

I/O 區域位址範圍為 CIO 0000 至 CIO 0079 (CIO 位元 000000 至 007915 1280 點, 80CH), 但這個區域可藉以程式書寫器以外的任何 CX-P 軟體來改變第一個 Rack words 來擴充為 CIO 0000 至 CIO 0999。即使 I/O 區域擴充了, 外部 I/O 可以配置的最大位元數仍為 1,280 (80 words)。

附註 外部 I/O 點的最大數量決定於所使用的 CPU 模組。

I/O 區域中的 words 是配置到基本 I/O 模組上的 I/O 端子。

配置到基本 I/O 模組的 words 是依據插槽位置 (由左至右) 與所需要的 words 數而定的。words 是依序配置, 且空的插槽會跳過。沒有配置到基本 I/O 模組的 I/O 區域 words 僅能用於程式中。

強制位元狀態

I/O 區域中的位元可以強制設定與強制重置。

I/O 區域初值化

I/O 區域的內容在以下的狀況下將會被清除：

- 1,2,3... 1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
(參見以下 IOM 保持位元操作的解說。)
2. PLC 的電源斷電復歸時 (ON → OFF → ON)。
(參見以下 IOM 保持位元操作的解說。)
3. I/O 區域為 CX-P 軟體所清除。
4. 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(若執行了 FALS(007), 則 I/O 區域的內容將會維持。)

IOM 保持位元操作

若 IOM 保持位元 (A50012) 為 ON, 當發生嚴重錯誤或操作模式由 PROGRAM 模式改變為 RUN 或 MONITOR 模式, 或反過來的狀況時, I/O 區域的內容將不會被清除。

若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的 " 起動時之 IOM 保持位元狀態 " 設定為保護 IOM 保持位元, 則當 PLC 的電源循環時, I/O 區域的內容將不會被清除。所有的 I/O 位元, 包括輸出, 將會保持它們在 PLC 關掉之前的狀態。

附註 若 I/O 保持位元變為 ON, 則當 PLC 由 RUN 或 MONITOR 模式切換到 PROGRAM 模式時, PLC 的輸出將無法變為 OFF 並將維持它們先前的狀態。當這個狀況發生時, 務必確認外部負載將不會產生嚴重的狀況。(當因嚴重錯誤而導致操作

停止時，包括由 FALS(007) 指令所造成的狀況，輸出模組的所有輸出將會變為 OFF 且只有內部輸出狀態會被維持。)

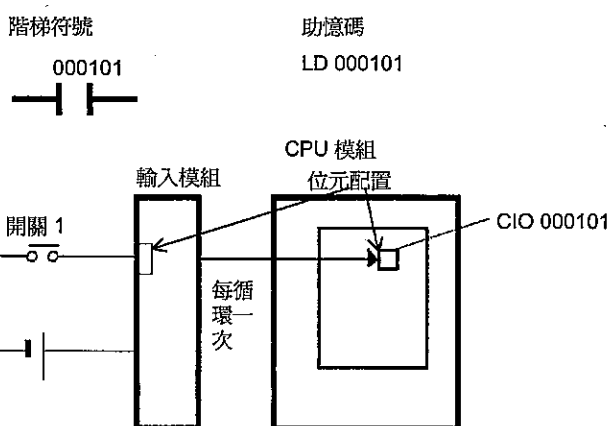
輸入位元

I/O 區域中配置到輸入模組的位元被稱為輸入位元。輸入位元反應按鈕開關、限動開關、與光電開關等裝置的 ON/OFF 狀態。有三種方法可以把 PLC 中的輸入點狀態加以更新：正常 I/O 更新、立即更新、與 IORF(097) 更新。

正常 I/O 更新

程式執行後，每個循環都會讀取一次外部裝置上之 I/O 點的狀態。

在以下的例子中，CIO 000101 配置到開關 1，一外部開關連接到輸入模組的輸入端子。開關 1 的 ON/OFF 狀態在每個循環都會反應於 CIO 000101 一次。



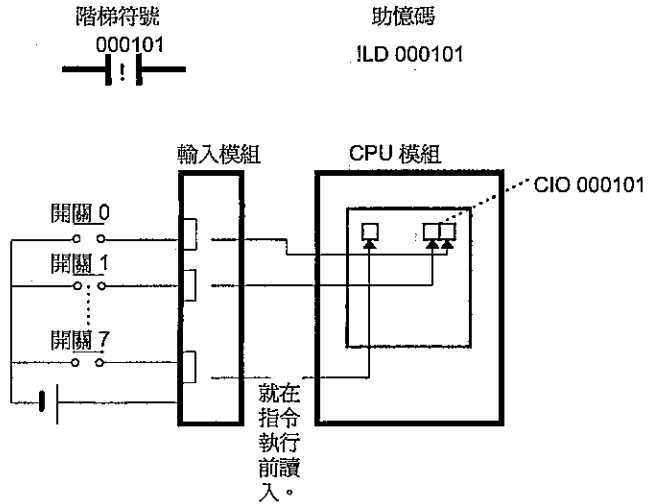
立即更新

當在指令前加上驚嘆號來指定該指令的立即更新變化量，且指令的運算元為輸入位元或 words，這個包含位元的 words 或 words 本身將在指令執行前更新。這個立即更新是除了每個循環進行一次的正常 I/O 更新之外額外進行的。

1,2,3...

1. 位元運算元
就在指令執行前，配置到包含指定位元的 words 之 16 個 I/O 點的 ON/OFF 狀態將會被讀到 PLC。
2. words 運算元
就在指令執行前，配置到包含指定 words 之 16 個 I/O 點的 ON/OFF 狀態將會被讀到 PLC。

在以下的例子中，CIO 000101 配置到開關 1，一外部開關連接到輸入模組的輸入端子。就在 !LD 000101 執行之前，開關 1 的 ON/OFF 狀態會被讀取並反應到 CIO 000101。

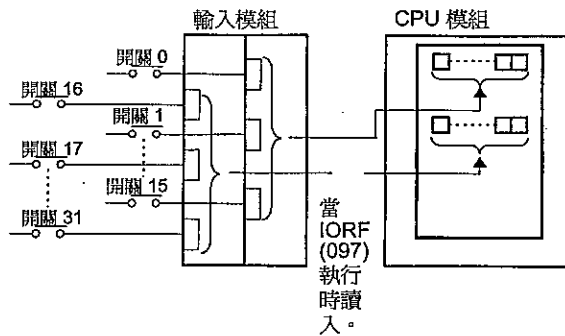


IORF(097) 更新

當 IORF(097) (I/O REFRESH) 執行時，words 指定範圍中的輸入位元會更新。這個 I/O 更新是除了每個循環進行一次的正常 I/O 更新之外額外進行的。以下的 IORF(097) 指令會更新 I/O 區塊 words CIO 0000 至 CIO 0003 中所有 I/O 點的狀態。輸入點的狀態是由輸入模組讀入，而輸出位元的狀態則是寫入至輸出模組。



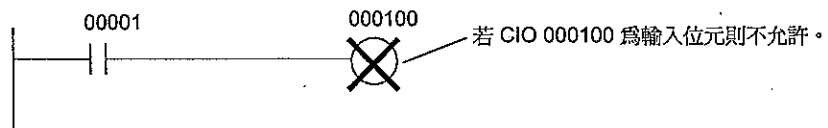
在以下的例子中，配置到 CIO 0000 與 CIO 0001 的輸入點狀態是由輸入模組讀入的。(CIO 0002 與 CIO 0003 配置到輸出模組)



輸入位元的限制

輸入位元可以在程式中當作常開或常閉使用的次數是沒有限制的，而且位址可以以任何順序來規劃。

輸入位元不能在輸出指令中當作運算元使用。

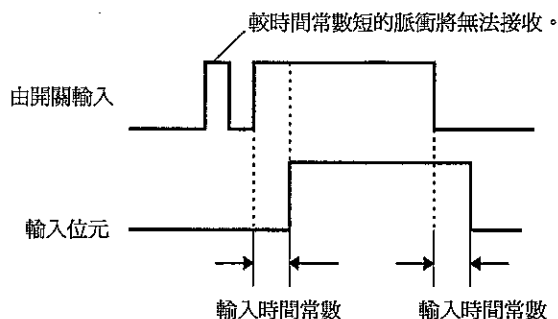


輸入回應時間設定

每個輸入模組的輸入回應時間可以在 PLC 設定中設定。增加輸入回應時間將減少顫動與雜訊的影響，而減少輸入回應時間則可接收較高速的輸入脈衝。

輸入回應時間的預設值為 8 ms，而設定範圍為 0.5 ms 至 32 ms。

附註 若時間設定為 0 ms，將還是會有因內部元件所造成最多 20 μ s 的 ON 延遲時間與 300 μ s 的 OFF 延遲時間。



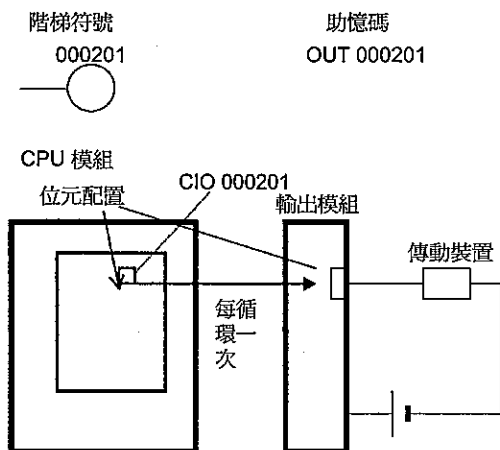
輸出位元

I/O 區域中配置到輸出模組的位元被稱為輸出位元。輸出位元的 ON/OFF 狀態會輸出到傳動裝置 (actuator)。有三種方法可以把 PLC 中的輸出位元狀態加以更新：正常 I/O 更新、立即更新、與 IORF(097) 更新。

正常 I/O 更新

程式執行後，每個循環都會把輸出位元的狀態輸出至外部裝置。

在以下的例子中，CIO 000201 配置到一個傳動裝置，一外部開關連接到輸出模組的輸出端子。CIO 000201 的 ON/OFF 狀態在每個循環都會輸出至該傳動裝置。



立即更新

當以在指令前加上驚嘆號來指定該指令的立即更新變化量，且指令的運算元為輸出位元或 words，這個包含位元的 words 或 words 本身將在指令執行前更新。這個立即更新是除了每個循環進行一次的正常 I/O 更新之外額外進行的。

1,2,3...

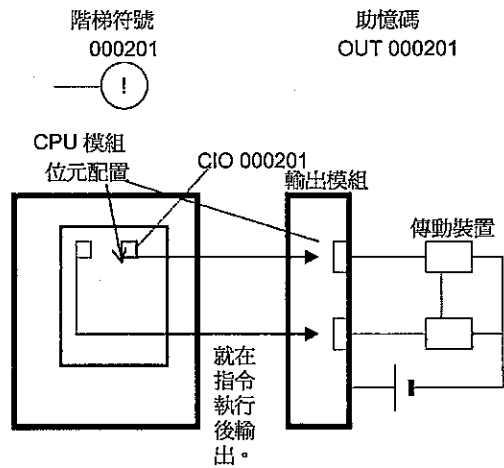
1. 位元運算元

就在指令執行後，配置到包含指定位元的 words 之 16 個 I/O 點的 ON/OFF 狀態將會被輸出至輸出裝置。

2. words 運算元

就在指令執行後，配置到包含指定 words 之 16 個 I/O 點的 ON/OFF 狀態將會被輸出至輸出裝置。

在以下的例子中，CIO 000201 配置到一個傳動裝置，一外部裝置連接到輸出模組的輸出端子。就在 !OUT 000201 執行之後，CIO 000201 的 ON/OFF 狀態會輸出至傳動裝置。

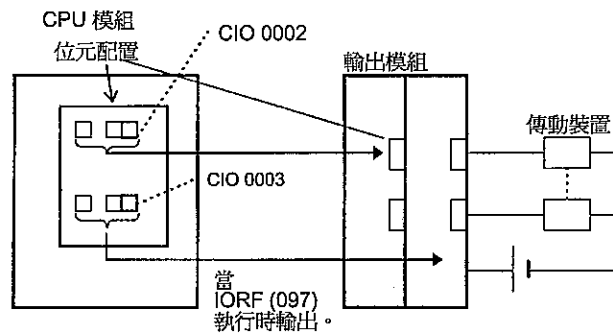
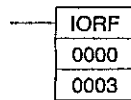


IORF(097) 更新

當 IORF(097) (I/O REFRESH) 執行時，words 指定範圍中輸出位元的 ON/OFF 狀態會輸出至它們的外部裝置。這個 I/O 更新是除了每個循環進行一次的正常 I/O 更新之外額外進行的。

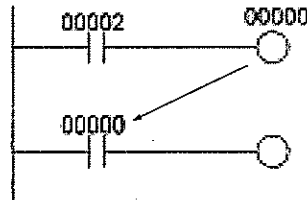
以下的 IORF(097) 指令會更新 I/O 區塊 words CIO 0000 至 CIO 0003 中所有 I/O 點的狀態。輸入點的狀態是由輸入模組讀入，而輸出位元的狀態則是寫入至輸出模組。

在這個例子中，配置到 CIO 0002 與 CIO 0003 的輸出點狀態是輸出至輸出模組。(CIO 0000 與 CIO 0001 是配置到輸入模組)

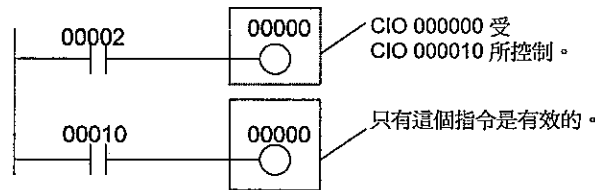


輸出位元的限制

輸出位元可依任何順序規劃。輸出位元可在輸入指令中當作運算元使用，且輸出位元當作常開或常閉條件使用的次數是沒有限制的。



一輸出位元只能使用於一個控制其狀態的輸出指令。若一輸出位元使用於二個或以上的輸出指令，則只有最後一個指令是有效的。

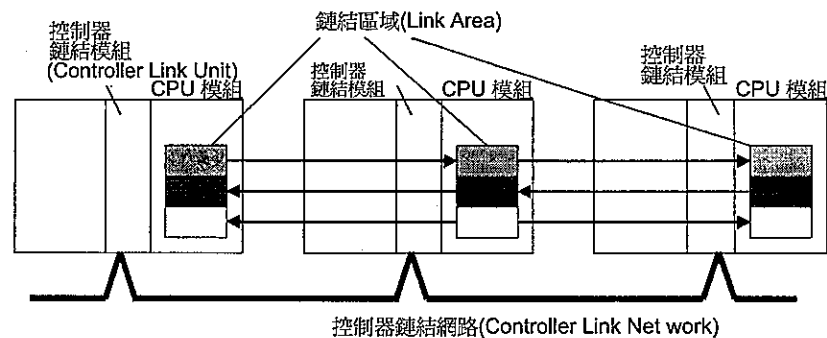


附註 可以藉著把輸出 OFF 位元 (A50015) 變成 ON 來把所有基本 I/O 模組與特殊 I/O 模組上的輸出都變為 OFF。即使真正的輸出是變為 OFF，也不會影響到輸出位元的狀態。

7-3 資料 Link 區域 (Data Link Area)

資料 Link 區域位址範圍為 CIO 1000 至 CIO 1199 (CIO 位元 100000 至 119915) 3200 點 (200CH)。Link 區域中的 words 可用於 Controller Link 網路中的資料 Link。一資料 Link 會透過安裝於 PLC 的 CPU Rack 上的 Controller Link 模組自動 (與程式無關) 與網路中其他 CJ 系列 CPU 模組的 Link 區域分享資料。

資料 Link 可以是自動 (每個節點使用相同的 words 數) 或手動產生的。當一使用者手動定義了資料的 Link，他可以任意指定每個節點的 words 數與讓節點為僅能接收或僅能發送。更多的詳情請參見 Controller Link *模組操作手冊* (W309)。Link 區域中未作為資料 Link 的 words 僅能用於程式中。



強制位元狀態

Link 至 C200HX/HG/HE、C200HS、與 C200H PLC

資料 Link 區域中的位元可以強制設定與強制重置。

CJ 系列 PLC 中的 Link 區塊 words CIO 1000 至 CIO 1063 對應到在 C200HX/HG/HE PLC 中所建立資料 Link 的 Link 中繼區塊 words LR 00 至 LR 63。當為 CJ 系列 PLC 轉換 C200HX/HG/HE、C200HS、或 C200H 程式時，把 LR 00 到 LR 63 的位址轉換至它們的等效 Link 區域位址之 CIO 1000 到 CIO 1063。

Link 區域初值化

1,2,3...

在以下的情形下，Link 區域的內容將會被清除：

1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
2. PLC 的電源斷電復歸時 (ON → OFF → ON)。
3. Link 區域為 CX-P 軟體所清除。
4. 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(若執行了 FALS(007)，則 Link 區域的內容將會維持。)

IOM 保持位元操作

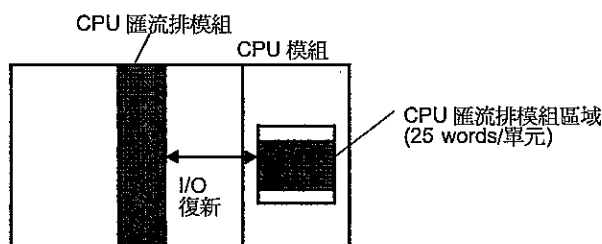
若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的 " 起動時之 IOM 保持位元狀態 " 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，Link 區域的內容將不會被清除。

若 IOM 保持位元 (A50012) 為 ON，當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，或是反過來的狀況，Link 區域的內容將不會被清除。

7-4 CPU 匯流排模組區域

CPU 匯流排模組區域包含位址範圍由 CIO 1500 至 CIO 1899 的 400 words。CPU 匯流排模組區域中的 words 可以配置到 CPU 匯流排模組以傳送模組操作狀態之類的資料。每個模組會依模組的編號設定配置 25 words。

在 I/O 更新期間，會於程式執行後的每個循環與 CPU 匯流排模組交換一次資料。(這個資料區域中的 words 無法立即更新或以 IORF(097) 更新。)



每個 CPU 匯流排模組會依模組的編號配置 25 words，如下表所示。

模組編號	配置的 words
0	CIO 1500 至 CIO 1524
1	CIO 1525 至 CIO 1549
2	CIO 1550 至 CIO 1574
3	CIO 1575 至 CIO 1599
4	CIO 1600 至 CIO 1624
5	CIO 1625 至 CIO 1649
6	CIO 1650 至 CIO 1674
7	CIO 1675 至 CIO 1699
8	CIO 1700 至 CIO 1724
9	CIO 1725 至 CIO 1749
A	CIO 1750 至 CIO 1774
B	CIO 1775 至 CIO 1799
C	CIO 1800 至 CIO 1824
D	CIO 1825 至 CIO 1849
E	CIO 1850 至 CIO 1874
F	CIO 1875 至 CIO 1899

25 words 的功能會依所使用的 CPU 匯流排模組之不同而有所不同。詳情請參見模組的操作手冊。

CPU 匯流排模組區域中沒有配置到 CPU 匯流排模組的 words 僅能用於程式中。

強制位元狀態

CPU 匯流排模組區域中的位元可以強制設定與強制重置。

CPU 匯流排模組區域初值化

在以下的狀況下，CPU 匯流排模組區域的內容將會被清除：

- 1,2,3... 1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
2. PLC 的電源斷電復歸時 (ON → OFF → ON)。
3. CPU 匯流排模組區域為 CX-P 軟體所清除。

- 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(若執行了 FALS(007)，則 CPU 匯流排模組區域的內容將會維持。)

IOM 保持位元操作

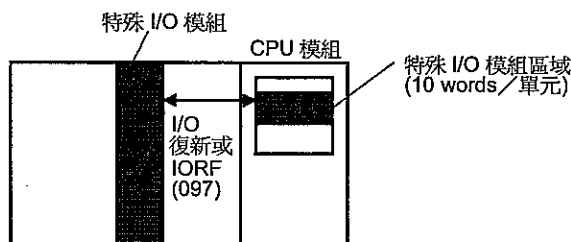
若 IOM 保持位元 (A50012) 為 ON，當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，CPU 匯流排模組區域的內容將不會被清除。

若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的 " 起動時之 IOM 保持位元狀態 " 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，CPU 匯流排模組區域的內容將不會被清除。

7-5 特殊 I/O 模組區域

特殊 I/O 模組區域包含位址範圍由 CIO 2000 至 CIO 2959 的 960 words。特殊 I/O 模組區域中的 words 可以配置到 CJ 以傳送模組操作狀態之類的資料。每個模組會依模組的編號設定配置 10 words。

在 I/O 更新期間，會於程式執行後的每個循環與特殊 I/O 模組交換一次資料。這些 words 也可以用 IORF(097) 來更新。



每個特殊 I/O 模組依模組的編號配置 10 words，如下表所示。

模組編號	配置的 words
0	CIO 2000 至 CIO 2009
1	CIO 2010 至 CIO 2019
2	CIO 2020 至 CIO 2029
3	CIO 2030 至 CIO 2039
4	CIO 2040 至 CIO 2049
5	CIO 2050 至 CIO 2059
6	CIO 2060 至 CIO 2069
7	CIO 2070 至 CIO 2079
8	CIO 2080 至 CIO 2089
9	CIO 2090 至 CIO 2099
10 (A)	CIO 2100 至 CIO 2109
11 (B)	CIO 2110 至 CIO 2119
12 (C)	CIO 2120 至 CIO 2129
13 (D)	CIO 2130 至 CIO 2139
14 (E)	CIO 2140 至 CIO 2149
15 (F)	CIO 2150 至 CIO 2159
16	CIO 2160 至 CIO 2169
17	CIO 2170 至 CIO 2179
95	CIO 2950 至 CIO 2959

這 10 個配置到模組的 words 功能會依所使用的特殊 I/O 模組之不同而有所不同。詳情請參見模組的操作手冊。

特殊 I/O 模組區域中沒有配置到特殊 I/O 模組的 words 僅能用於程式中。

強制位元狀態

特殊 I/O 模組區域中的位元可以強制設定與強制重置。

特殊 I/O 模組區域初值化

特殊 I/O 模組區域的內容將會被清除 在以下的狀況下：

1,2,3...

1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
2. PLC 的電源斷電復歸時 (ON → OFF → ON)。
3. 特殊 I/O 模組區域為 CX-P 軟體所清除。
4. 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(當執行了 FALS(007)，則特殊 I/O 模組區域的內容將會維持。)

IOM 保持位元操作

若 IOM 保持位元 (A50012) 為 ON，當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，特殊 I/O 模組區域的內容將不會被清除。

若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的"起動時之 IOM 保持位元狀態" 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，特殊 I/O 模組區域的內容將不會被清除。

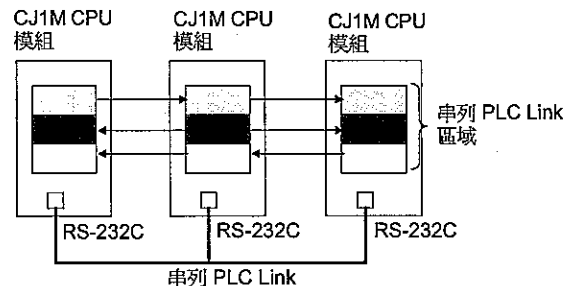
7-6 串列 PLC Link 區域

串列 PLC Link 區域包含位址範圍由 CIO 3100 至 CIO 3189 的 90 CH。串列 PLC Link 區域中的 words 可用於與其他 PLC 資料 Link。

串列 PLC Link 利用經由內藏的 RS-232C 埠，不需要特殊的程式即可交換 CPU 模組間的資料。

串列 PLC Link 配置藉由以下在主局模組的 PLC 設定自動進行設定。

- 串列 PLC Link 模式
- 串列 PLC Link 傳送的 words 數
- 串列 PLC Link 模組數的最大值



與工作區域相同的，串列 PLC Link 未使用的位址僅能用於程式中。

強制位元狀態

串列 PLC Link 區域中的位元可以強制設定與強制重置。

串列 PLC Link 區域初值化

在以下的狀況下，串列 PLC Link 區域的內容將會被清除：

1,2,3...

1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
2. PLC 的電源斷電復歸時 (ON → OFF → ON)。
3. 串列 PLC Link 區域為 CX-P 軟體所清除。
4. 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(當執行了 FALS(007)，則串列 PLC Link 區域的內容將會維持。)

IOM 保持位元操作

若 IOM 保持位元 (A50012) 為 ON，當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，串列 PLC Link 區域的內容將不會被清除。

若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的"起動時之 IOM 保持位元狀態" 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，串列 PLC Link 區域的內容將不會被清除。

7-7 DeviceNet 區域 (DeviceNet Area)

DeviceNet 區域包含位址範圍由 CIO 3200 至 CIO 3799 的 600 CH。DeviceNet 區域中的 words 配置到 DeviceNet 遠端 I/O 通訊的子局。資料有條不紊地透過 DeviceNet 模組與網路中的子局 (與程式無關) 進行交換。

使用固定配置的 words 會依固定配置設定 1, 2, 與 3 配置到子局。會選擇這些固定區域之一。

區域	輸出區域 (主局到子局)	輸入區域 (主局到子局)
固定配置區域 1	CIO 3200 至 CIO 3263	CIO 3300 至 CIO 3363
固定配置區域 2	CIO 3400 至 CIO 3463	CIO 3500 至 CIO 3563
固定配置區域 3	CIO 3600 至 CIO 3663	CIO 3700 至 CIO 3763

當遠端 I/O 從屬功能搭配固定配置使用時，以下的 words 會配置到 DeviceNet 模組。

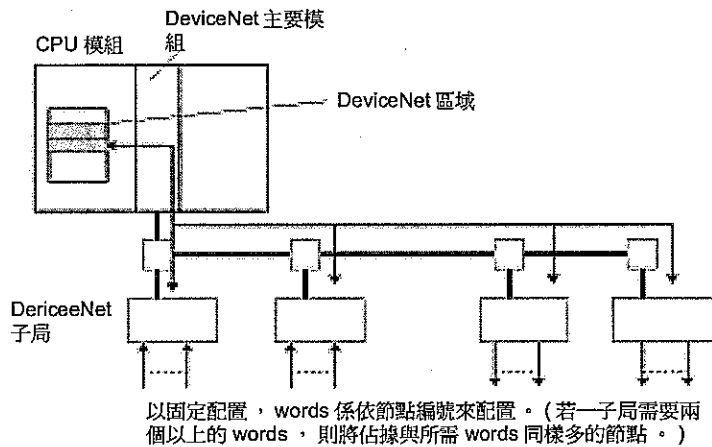
區域	輸出區域 (主局到子局)	輸入區域 (主局到子局)
固定配置區域 1	CIO 3370	CIO 3270
固定配置區域 2	CIO 3570	CIO 3470
固定配置區域 3	CIO 3770	CIO 3670

DeviceNet 區域中的位元可以強制設定與強制重置。

附註 有兩種方法可以配置 DeviceNet 中的 I/O: 依據節點位址固定配置與使用者設定配置。

- 以固定配置，words 會依據節點位址自動配置到指定的固定配置區域中的子局。
- 以使用者設定配置，使用者可由以下的 words 配置 words 到子局。
CIO 0000 至 CIO 0235, CIO 0300 至 CIO 0511, CIO 1000 至 CIO 1063
W000 至 W511
H000 至 H511
D00000 至 D32767
E00000 至 E32767, bank 0 至 2

有關 words 配置的詳情，請參見 *DeviceNet (CompoBus/D) 操作手冊 (W267)*。



DeviceNet 區域初值化

在以下的狀況下，DeviceNet 區域的內容將會被清除：

1,2,3...

1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
2. PLC 的電源斷電復歸時 (ON → OFF → ON) 供應為循環的。
3. DeviceNet 區域為 CX-P 軟體所清除。
4. 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(當執行了 FALS(007)，則 DeviceNet 區域的內容將會維持。)

IOM 保持位元操作

若 IOM 保持位元 (A50012) 為 ON，則當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式，或是反過來的狀況時，DeviceNet 區域的內容將不會被清除。

若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的 " 起動時之 IOM 保持位元狀態 " 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，DeviceNet 區域的內容將不會被清除。

7-8 內部 I/O 區域

內部 I/O (工作) 區域包含位址範圍由 W000 至 W511 的 512 CH。這些 words 僅能當作工作 words 用於程式中。

在 CIO 區域 (CIO 1200 至 CIO 1499 與 CIO 3800 至 CIO 6143) 中有些未使用的 words 可以用於程式中，但請優先使用工作區域中可用的 words，這是因為 CIO 區域中未使用的 words 在未來版本的 CJ 系列 CPU 模組中可能會被賦予新的功能。

工作區域中的位元可以強制設定與強制重置。

工作區域初值化

在以下的狀況下，工作區域中的內容將會被清除：

1,2,3...

1. 操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式。
2. PLC 的電源斷電復歸時 (ON → OFF → ON)。
3. 工作區域為 CX-P 軟體所清除。
4. 當發生 FALS(007) 錯誤以外的嚴重錯誤時 PLC 停止操作。(若執行了 FALS(007)，則工作區域的內容將會維持。)

IOM 保持位元操作

若 IOM 保持位元 (A50012) 為 ON，當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，工作區域的內容將無法清除。

若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的"起動時之 IOM 保持位元狀態" 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，工作區域的內容將無法清除。

7-9 保持區域 (Holding Area)

保持區域包含位址範圍由 H000 至 H511 (位元 H00000 至 H51115) 的 512 CH。這些 words 僅能用於程式中。

保持區域位元在程式中可依任何順序使用，必要時也可以當作常開或常閉的條件來使用。

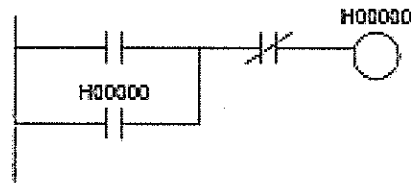
保持區域初值化

當 PLC 的電源供應是循環的，或 PLC 的操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式，或是反過來的狀況，保持區域中的資料不會清除。

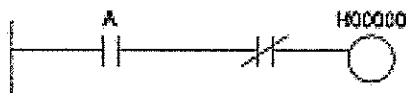
若一保持區域位元規劃在 IL(002) 與 ILC(003) 之間且 IL(002) 的執行條件為 OFF，則它將會被清除。要保持一個位元在 IL(002) 的執行條件為 OFF 下還要為 ON，請在 IL(002) 之前以 SET 指令把這個位元變為 ON。

自我保持位元

保持區域被規劃時，自我保持位元即使電源重置也將無法清除。

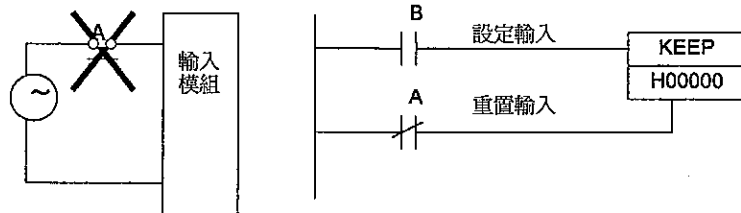


- 附註
1. 若一保持區域位元不是使用自我保持位元，則當電源重置時，這個位元將會變為 OFF 且此一位置元將會被清除。
 2. 若一保持區域位元不是以下圖的方式來規劃為自我保持位元使用，則當電源重置時，這個位元變為 OFF。

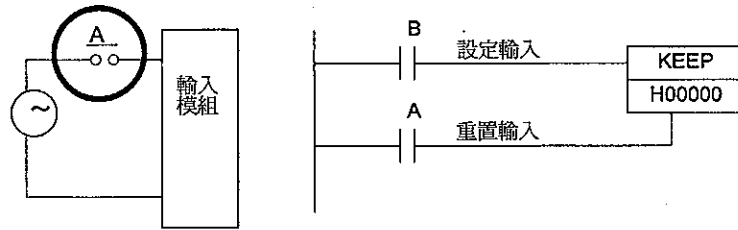


注意事項

當在 KEEP(011) 指令中使用保持區域位元時，若輸入裝置使用 AC 電源，則切勿使用常閉條件來作為重置輸入。當供應的電源變為 OFF 或是暫時中斷，則輸入將會在 PLC 的內部電源供應之前變為 OFF 且保持區域位元將會重置。



而應以下圖所示的方式來架構。



使用位元位址的順序與可規劃的 N.C. 或 N.O. 條件數是沒有任何限制的。

7-10 輔助區域 (Auxiliary Area)

輔助區域包含位址範圍為 A000 至 A959 的 960 CH。這些 words 預先設定為旗標與控制位元以監控與控制操作。

A000 到 A447 是唯讀的，但 A448 到 A959 則是可以由程式或 CX-P 軟體來讀取或寫入的。

強制位元狀態

輔助區域的位元不能連續地強制設定與強制重置。

寫入輔助區域資料

以下的操作可由 CX-P 軟體來進行以寫入資料到輔助區域。

- 使用 CX-P 軟體：當監控規劃位址 (設定值對話框)，或在編輯 PLC 資料表後傳送資料至 PLC 時，進行線上設定 / 重置 (非強制設定 / 強制重置)、改變顯示值。參見 CX-P 軟體使用者手冊 (W361-E2)。
- 使用程式書寫器：由位元 / words 監控器或 3-words 監控器操作進行暫時地強制設定 / 強制重置位元 (參見程式書寫器操作手冊)。

功能

以下各表列出輔助區域旗標與控制位元的功能。表格是依據旗標與位元的功能來編排的。

初始設定

名稱	位址	描述	存取
基本 I/O 模組中的 I/O 回應時間	A22000 至 A25915	包含 CJ 系列基本 I/O 模組現行的 I/O 回應時間。	唯讀
IOM 保持位元	A50012	決定當 PLC 的電源重置或 PLC 的操作模式改變 (由 PROGRAM 至 RUN/MONITOR 或反過來的狀況) 時 I/O 記憶體的內容是否要維持。 把這個位元變為 ON 以便在 PROGRAM 與 RUN 或 MONITOR 模式之間改變時能維持 I/O 記憶體。 把這個位元變為 OFF 以便在 PROGRAM 與 RUN 或 MONITOR 模式之間改變時能清除 I/O 記憶體。	讀 / 寫
強制狀態保持位元	A50013	決定當 PLC 的電源重置或 PLC 的操作模式改變 (在 PROGRAM 與 RUN 或 MONITOR 模式) 時，強制設定與強制重置位元的狀態是否要維持。	讀 / 寫
模式改變除能設定 (僅 CJ1-H 與 CJ1M CPU 模組)	A530	設定為 A5A5 Hex 以使 DI(693) 與 EI(694) 指令之間的電源中斷 (除了電源 OFF 中斷工作單以外)。 A5A5 Hex：電源禁止區間設定有效。 A5A5 Hex 以外：電源禁止區間設定無效。	讀 / 寫

CPU 模組設定

名稱	位址	描述	存取
指撥開關接腳 6 的狀態	A39512	包含 CPU 模組指撥開關接腳 6 的狀態 (每循環更新)	唯讀

基本 I/O 模組設定

名稱	位址	描述	存取
基本 I/O 模組狀態區域	A05000 至 A08915	指示基本 I/O 模組中的保險絲是完好的還是燒掉的。這個旗標對應到 Rack 0 插槽 0 到 Rack 7 插槽 9。	唯讀
I/O 配置狀態	A260	指示 I/O 配置的現行狀態，即是起動時自動 I/O 配置還是使用者設定 I/O 配置。	唯讀
起動時模組檢測 (Rack 0 至 3) (僅 CJ1-H 與 CJ1M CPU 模組)	Rack 0 : A33600 至 A33603 Rack 1 : A33604 至 A33607 Rack 2 : A33608 至 A33611 Rack 3 : A33612 至 A33615	檢測所得每 Rack 的模組數以 Hex 的 1 位數儲存 (0 至 A Hex)。範例：若 Rack 0 有 1 模組，Rack 1 有 4 模組，Rack 2 有 8 模組而 Rack 3 有 10 模組，則將會儲存如下： A336 = A 8 4 1	唯讀

CPU 匯流排模組旗標 / 位元

名稱	位址	描述	存取
CPU 匯流排模組初值旗標	A30200 至 A30215	這些對應到 CPU 匯流排模組 0 至 15。當模組在電源變為 ON 後或模組的重新起動位元 (in A501) 變為 ON 後進行初值化，則相對應的旗標會為 ON。	唯讀
CPU 匯流排模組重新起動位元	A50100 至 A50115	這些位元對應到 CPU 匯流排模組 0 至 15。把一位元由 OFF 變為 ON 以重新起動相對應的模組。	讀 / 寫

特殊 I/O 模組旗標 / 位元

名稱	位址	描述	存取
特殊 I/O 模組初值旗標	A33000 至 A33515	這些對應到特殊 I/O 模組 0 至 95。當模組在電源變為 ON 後或模組的重新起動位元變為 ON 後進行初值化，則相對應的旗標將會為 ON。(重新起動位元 A50200 至 A50715 對應到模組 0 至 95。)	唯讀
特殊 I/O 模組重新起動位元	A50200 至 A50715	這些位元對應到特殊 I/O 模組 0 至 95。把一位元由 OFF 變為 ON 以重新起動相對應的模組。	讀 / 寫

系統旗標

名稱	位址	描述	存取
第一循環旗標	A20011	當程式開始執行時這個旗標有一個循環變為 ON (操作模式由 PROGRAM 切換至 RUN/MONITOR)。	唯讀
初始工作單 (task) 執行旗標	A20015	當一工作單第一次由 INI 切換至 RUN 狀態時，這個旗標將會在工作單的一個循環中變為 ON。	唯讀
工作單起動旗標 (僅 CJ1-H 與 CJ1M CPU 模組)	A20014	當一工作單由 WAIT 或 INI 切換至 RUN 狀態時，這個旗標將會在工作單的一個循環中變為 ON。 這個旗標與 A20015 唯一的差別就是當工作單由 WAIT 切換至 RUN 狀態時這個旗標也會變為 ON。	唯讀
最大循環時間	A262 至 A263	這些 words 包含最大循環時間，單位為 0.1 ms。在平行處理模式中，程式執行循環的最大循環時間將是給定的。這個時間每循環都會更新，並以 32 位元的 BIN 數記錄 (0 至 FFFF FFFF，或 0 至 429,496,729.5 ms)。(A263 為最左邊的 words)	唯讀
顯示循環時間	A264 至 A265	這些 words 包含顯示循環時間，單位為 0.1 ms。在平行處理模式中，程式執行循環的最大循環時間將是給定的。這個時間每循環都會更新，並以 32 位元的 BIN 數記錄 (0 至 FFFF FFFF，或 0 至 429,496,729.5 ms)。(A265 為最左邊的 words)	唯讀
週邊服務循環時間 (僅 CJ1-H CPU 模組)	A268	在平行處理中以同步或非同步記憶體存取時，這個 words 包含週邊服務循環時間，單位為 0.1 ms。這個時間每循環都會更新，並以 16 位元的 BIN 數記錄 (0 至 4E20 Hex，或 0.0 至 2,000.0 ms)。	唯讀

工作單資訊 (Task Information)

名稱	位址	描述	存取
程式停止時的工作單編號	A294	這個 words 包含程式因錯誤而停止執行時所執行的工作單編號。	唯讀
中斷工作單最大處理時間	A440	包含中斷工作單最大處理時間，單位為 0.1 ms。	唯讀
最大處理時間之中斷工作單	A441	包含處理時間最長之中斷工作單編號。Hex 數 8000 至 80FF 對應到工作單編號 00 至 FF。當發生中斷時，位元 15 變為 ON。	唯讀
在工作單間進行 IR/DR 操作 (僅 CJ1-H 與 CJ1M CPU 模組)	A09914	把這個位元變為 ON 以便能與所有的工作單間之資料暫存器分享索引。把這個位元變為 OFF 以便能在每個工作單間之資料暫存器個別使用索引。	唯讀

除錯資訊

■線上編輯

名稱	位址	描述	存取
線上編輯等待旗標	A20110	ON 當線上編輯處理等待中。 (當線上編輯除能時，會接到一個線上編輯請求)	唯讀
線上編輯處理旗標	A20111	ON 當線上編輯處理執行中。	唯讀
線上編輯除能位元確認器	A52700 至 A52707	只有在這個位元包含 5A 時，這個線上編輯除能位元 (A52709) 才是正當的。	讀 / 寫
線上編輯除能位元	A52709	把這個位元變為 ON 以使線上編輯除能。	讀 / 寫

■輸出控制

名稱	位址	描述	存取
輸出 OFF 位元	A50015	把這個位元變為 ON 以把所有基本 I/O 模組、輸出模組、與特殊 I/O 模組的輸出都變為 OFF。	讀 / 寫

■ 微分監控

名稱	位址	描述	存取
微分監控完成旗標	A50809	ON 當已經建立微分監控執行期間的微分監控條件時。	讀 / 寫

■ 資料追蹤

名稱	位址	描述	存取
取樣起動位元	A50815	當由 CX-P 軟體把個位元由 OFF 變為 ON 而開始資料追蹤時, PLC 將會以下列三種方法開始儲存追蹤記憶體之資料: 1) 週期性取樣 (10 至 2,550 ms) 2) TRSM(045) 執行時取樣 3) 每個循環結束時取樣	讀 / 寫
追蹤起動位元	A50814	把個位元由 OFF 變為 ON 以建立觸發條件。這個以延遲值 (正值或負值) 表示的偏量決定那一個資料取樣是正當的。	讀 / 寫
追蹤忙碌旗標	A50813	當取樣起動位元 (A50815) 由 OFF 變為 ON 時為 ON。當追蹤完成時為 OFF。	讀 / 寫
追蹤完成旗標	A50812	當追蹤期間完成追蹤記憶體範圍的取樣時為 ON。當下一次取樣起動位元 (A50815) 由 OFF 變為 ON 時為 OFF。	讀 / 寫
追蹤觸發監控旗標	A50811	當由追蹤起動位元 (A50814) 建立一觸發條件時為 ON。當取樣起動位元 (A50815) 開始下一次資料追蹤時為 OFF。	讀 / 寫

檔案記憶體資訊

名稱	位址	描述	存取
記憶卡型式	A34300 至 A34302	若有安裝記憶卡, 會指出記憶卡的型式。	唯讀
記憶卡格式錯誤旗標	A34307	當記憶卡未格式化或發生格式化錯誤時為 ON。	唯讀
檔案傳送錯誤旗標	A34308	當發生資料寫入檔案記憶體錯誤時為 ON。	唯讀
檔案寫入錯誤旗標	A34309	當因寫入保護或資料超過檔案記憶體容量而導致資料無法寫入檔案記憶體時為 ON。	唯讀
檔案讀取錯誤	A34310	當因動作不正常 (檔案損壞或資料衝突) 而無法讀取檔案時為 ON。	唯讀
檔案遺失旗標	A34311	當意圖讀取一個不存在的檔案或把檔案寫入一個不存在的目錄時為 ON。	唯讀
檔案記憶體操作旗標	A34313	當執行以下任一項操作時為 ON。OFF 當沒有任何一項執行時為 OFF。 記憶卡檢測開始。 CMND 指令發送一 FINS 指令至本地的 CPU 模組。 FREAD/FWRIT 指令。 使用輔助區域中的控制位元進行程式置換。 簡易備份操作。 若此旗標為 ON, 將無法對記憶卡執行寫入與比較操作。	唯讀
記憶卡檢測旗標	A34315	ON 當檢測到記憶卡時。 OFF 當檢測不到記憶卡時。	唯讀
要傳送的項目數	A346 至 A347	這些 words 包含還需要傳送的 words 數或欄位數 (32 位元)。對於 BIN 檔案 (.IOM), 每個 words 讀取時會使這個值減 1。對於純文字檔案 (.TXT) 或 CSV (.CSV) 資料, 每個欄位讀取時會使這個值減 1。	唯讀
存取檔案資料旗標	A34314	當資料存取時為 ON。	唯讀
EM 檔案記憶體格式 錯誤旗標 (僅 CJ1 與 CJ1-H CPU 模組)	A34306	當檔案記憶體的第一個 EM bank 發生格式錯誤時會變為 ON。當格式化完全正常時會變為 OFF。	唯讀

名稱	位址	描述	存取
EM 檔案記憶體起動 bank (僅 CJ1 與 CJ1-H CPU 模組)	A344	包含 EM 檔案記憶體的起動 bank 編號 (第一個已格式化 bank 的編號)。 當開始由記憶卡寫入資料時會讀取這個編號。若有要進行簡易備份之 EM 檔案的最大的 bank 數 (BACKUPE.IOM, 其中表示連續的 bank 編號) 與 CPU 模組支援的最大 bank 編號相同, 則這個 EM 區域將會使用 A344 中的值來把它格式化為檔案記憶體。若最大的 bank 編號不同, 則 EM 區域將回到其未格式化 (非檔案記憶體) 的狀態。	唯讀
檔案刪除旗標	A39506	系統在發生電源中斷時自動刪除已更新之 EM 檔案記憶檔案的剩餘部分。	唯讀
	A39507	系統在發生電源中斷時自動刪除已更新之記憶卡檔案的剩餘部分。	唯讀
簡易備份寫入容量	A397	若簡易備份操作的寫入失敗, A397 將包含完成這項寫入操作所需之記憶卡容量。這個值的單位是千位元組。(這表示在開始寫入時記憶卡沒有足夠的容量。)。 0001 至 FFFF Hex: 寫入錯誤 (值表示所需的容量, 由 1 至 65,535 千位元組)。 當簡易備份操作的寫入完成時 A397 將會清除為 0000 Hex。	唯讀
程式置換結束碼	A65000 至 A65007	正常結束 (即 A65014 為 OFF) 01 Hex: 程式檔案 (.OBJ) 已置換。 錯誤結束 (即 A65014 為 ON) 00 Hex: 嚴重錯誤 01 Hex: 記憶體錯誤 11 Hex: 寫入保護 12 Hex: 程式置換密碼錯誤 21 Hex: 無記憶卡 22 Hex: 無此檔案 23 Hex: 指定的檔案超過容量 (記憶體錯誤)。 31 Hex: 以下處理之一: 檔案記憶體操作 使用者程式寫入 操作模式改變	唯讀
置換錯誤旗標	A65014	當置換起動位元 (A65015) 已經變為 ON 要置換檔案卻發生錯誤時為 ON。若置換起動位元再次變為 ON, 置換錯誤旗標將會變為 OFF。	讀 / 寫
置換起動位元	A65015	若程式密碼 (A651) 為正當的 (A5A5 Hex), 當置換起動位元變為 ON 時開始程式置換。在程式置換期間不可以把置換起動位元變為 OFF。 當電源變為 ON 或程式置換完成時, 不論置換是否正常完成或有錯誤, 置換起動位元都將會變為 OFF。 可以用 CX-P 軟體、PT、或主電腦讀取置換起動位元來確認程式置換是否正在執行。	讀 / 寫

名稱	位址	描述	存取															
程式密碼	A651	輸入置換程式的密碼。 A5A5 Hex：置換起動位元 (A65015) 為有效。 其他任何值：置換起動位元 (A65015) 為無效。 當電源變為 ON 或程式置換完成時，不論置換是否正常完成或有錯誤，置換起動位元都將會變為 OFF。	讀 / 寫															
程式檔案名稱	A654 至 A657	當程式置換開始，程式檔案名稱將以 ASCII 碼儲存。檔案名稱不含附檔名最多可以八個字元。 檔案名稱會以以下的順序儲存：A654 至 A657 (即由最低的 words 至最高的 words)，且是由最高位元組至最低位元組。 若檔案名稱小於八個字元，則最低的剩餘位元組與最高的剩餘 words 將會以空白 (20 Hex) 填滿。空字元與空白不能使用於檔案名稱。 範例：檔案名稱為 ABC.OBJ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td style="text-align: center;">15</td> <td style="text-align: center;">0</td> </tr> <tr> <td>A654</td> <td style="text-align: center;">41</td> <td style="text-align: center;">42</td> </tr> <tr> <td>A655</td> <td style="text-align: center;">43</td> <td style="text-align: center;">20</td> </tr> <tr> <td>A656</td> <td style="text-align: center;">20</td> <td style="text-align: center;">20</td> </tr> <tr> <td>A657</td> <td style="text-align: center;">20</td> <td style="text-align: center;">20</td> </tr> </table>		15	0	A654	41	42	A655	43	20	A656	20	20	A657	20	20	讀 / 寫
	15	0																
A654	41	42																
A655	43	20																
A656	20	20																
A657	20	20																

程式錯誤資訊

名稱	位址	描述	存取
程式錯誤旗標 (嚴重錯誤)	A40109	當程式內容不正確時為 ON。CPU 模組操作將會停止。	唯讀
程式錯誤工作單	A294	當程式因錯誤而停止執行時，提供所執行的工作單型式與編號。	唯讀
指令處理錯誤旗標	A29508	當發生指令處理錯誤且 PLC 設定已經設定為指令錯誤時要停止的狀況下，這個旗標與錯誤旗標 (ER) 都將會變為 ON。	唯讀
間接 DM/EM BCD 錯誤旗標	A29509	當發生間接 DM/EM BCD 錯誤且 PLC 設定已經設定為間接 DM/EM BCD 錯誤時要停止的狀況下，這個旗標與存取錯誤旗標 (AER) 都將會變為 ON。	唯讀
非法存取錯誤旗標	A29510	當發生非法存取錯誤且 PLC 設定已經設定為非法存取錯誤時要停止的狀況下，這個旗標與存取錯誤旗標 (AER) 都將會變為 ON。	唯讀
無結束錯誤旗標	A29511	當不是工作單中的每個程式都有 END(001) 指令時為 ON。	唯讀
工作單錯誤旗標	A29512	當發生工作單錯誤時為 ON。以下的狀況將產生工作單錯誤。 1) 沒有可執行的循環工作單。 2) 沒有程式配置到工作單。	唯讀
微分溢位錯誤旗標	A29513	當指定的微分旗標編號超過容許值時為 ON。	唯讀
不當指令錯誤旗標	A29514	當已儲存的程式無法執行時為 ON。	唯讀
UM 溢位錯誤旗標	A29515	當 UM (使用者程式記憶體) 中最後的位址超過時為 ON。	唯讀
程式停止位址	A298 與 A299	這些 words 包含程式因錯誤而停止執行之指令的 8 位元 Hex 程式位址。 (A299 為最左邊的位元)	唯讀

錯誤資訊

■ 錯誤記錄，錯誤碼

名稱	位址	描述	存取
錯誤記錄區域	A100 至 A199	當發生錯誤時，錯誤碼、錯誤內容、及發生的時間與日期會儲存於這個錯誤記錄區域。	唯讀
錯誤記錄指標	A300	當發生錯誤時，錯誤記錄指標會加 1 以指出下一個錯誤記錄的位置，這個位址是由錯誤記錄區域 (A100) 開始之一個偏量。	唯讀
錯誤記錄指標重置位元	A50014	把這個位元變為 ON 以重置錯誤記錄指標 (A300) 為 00。	讀 / 寫
錯誤碼	A400	當發生非嚴重錯誤 (使用者定義 FALS(006) 或系統錯誤) 或嚴重錯誤 (使用者定義 FALS(007) 或系統錯誤) 時，一個 4 位元的 Hex 錯誤碼會寫入這個 words。	唯讀

■ FAL/FALS 錯誤資訊

名稱	位址	描述	存取
FAL 錯誤旗標 (非嚴重錯誤)	A40215	當執行 FAL(006) 而產生非嚴重錯誤時為 ON。	唯讀
執行 FAL 編號 旗標	A360 至 A391	當 FAL(006) 執行時，對應到指定 FAL 編號的旗標將會變為 ON。位元 A36001 至 A39115 對應到 FAL 編號 001 至 511。	唯讀
FALS 錯誤旗標 (嚴重錯誤)	A40106	當執行 FAL(007) 而產生嚴重錯誤時為 ON。	唯讀
系統錯誤模擬的 FAL/FALS 編號 (僅 CJ1-H 與 CJ1M CPU 模組)	A053	設定一個假的 FAL/FALS 編號以使用 FAL(006) 或 FALS(007) 來模擬系統錯誤。 0001 至 01FF Hex: FAL/FALS 編號 1 至 511 0000 或 0200 至 FFFF Hex: 無系統錯誤模擬的 FAL/FALS 編號 (將不會產生錯誤)	讀 / 寫

■ 記憶體錯誤資訊

名稱	位址	描述	存取
記憶體錯誤旗標 (嚴重錯誤)	A40115	當發生記憶體錯誤時，或在電源開啓時由記憶卡自動傳送發生錯誤時會為 ON。 當這個旗標變為 ON 時，CPU 模組面板之 ERR/ALM 指示燈將會亮，且 CPU 模組操作將會停止。 若起動時自動資料傳送失敗，A40309 將會變為 ON。若在起動時自動資料傳送發生錯誤時，這個錯誤將不能被清除。	唯讀
記憶體錯誤位置	A40300 至 A40308	當發生記憶體錯誤時，記憶體錯誤旗標 (A40115) 會變為 ON 且以下的旗標之一也會變為 ON 以指出記憶體區域發生錯誤的地方。 A40300: 使用者程式 A40304: PLC 設定 A40305: 註記 I/O 表 A40307: 路由表 A40308: CJ 系列 CPU 匯流排模組設定	唯讀
起動記憶卡傳送錯誤旗標	A40309	當起動時由記憶卡自動傳送檔案至 CPU 模組發生錯誤，包括檔案遺失或記憶卡未安裝會為 ON。 關掉電源可清除這項錯誤。(這項錯誤在當電源為 ON 時無法清除)	唯讀
快閃記憶體錯誤 (僅 CJ1-H 與 CJ1M CPU 模組)	A40310	當快閃記憶體失效時變為 ON	唯讀

■ PLC 設定錯誤資訊

名稱	位址	描述	存取
PLC 設定錯誤旗標 (非嚴重錯誤)	A40210	當 PLC 設定中有設定錯誤時會為 ON。	唯讀
PLC 設定錯誤位置	A406	當 PLC 設定中有設定錯誤時，錯誤位置會以 16 位元的 BIN 數寫入 A406。這個位置就是程式書寫器所設定的位址。	唯讀

■ 中斷工作單錯誤資訊

名稱	位址	描述	存取
中斷工作單錯誤旗標 (非嚴重錯誤)	A40213	當 PLC 設定中的檢測中斷工作單錯誤設定是設定為檢測而發生以下的狀況之一時會為 ON。 循環工作單中的 IORD(222) 或 IOWR(223) 與中斷工作單中的 IORD(222) 或 IOWR(223) 發生競爭。 I/O 更新時在中斷工作單中執行 IORD(222) 或 IOWR(223)。	唯讀
中斷工作單錯誤原因旗標	A42615	指出中斷工作單錯誤的原因。	唯讀
中斷工作單錯誤，工作單編號	A42600 至 A42611	這些位元的功能決定於 A42615 的狀態 (中斷工作單錯誤旗標)。 A42615 ON： 當一特殊 I/O 模組的 I/O 正由循環 I/O 更新進行更新工作時，意圖由中斷工作單以 IORF(097) 更新該模組的 I/O (重覆更新)，它會包含這個特殊 I/O 模組的模組編號。	唯讀

■ I/O 資訊

名稱	位址	描述	存取
基本 I/O 模組錯誤旗標 (非嚴重錯誤)	A40212	當基本 I/O 模組 (包括 C200H 第 2 群組 高密度 I/O 模組與 C200H 中斷輸入模組) 發生錯誤時會為 ON。	唯讀
基本 I/O 模組錯誤，插槽編號	A40800 至 A40807	包含當基本 I/O 模組 (包括 C200H 第 2 群組 高密度 I/O 模組與 C200H 中斷輸入模組) 發生錯誤時錯誤發生的 BIN 插槽編號。	唯讀
基本 I/O 模組錯誤，Rack 編號	A40808 至 A40815	包含當基本 I/O 模組 (包括 C200H 第 2 群組 高密度 I/O 模組與 C200H 中斷輸入模組) 發生錯誤時錯誤發生的 BIN Rack 編號。	唯讀
I/O 設定錯誤旗標 (嚴重錯誤)	A40110	當一輸入模組被安裝到輸出模組的插槽，或反過來的狀況，以致於註記 I/O 表中的輸入與輸出模組不一致時會為 ON。	唯讀
擴充 I/O Rack 編號重複旗標	A40900 至 A40903	當由 CX-P 軟體設定一擴充 I/O Racks 的起動 words 位址且兩個 Racks 有重疊的 words 配置或一 Racks CIO 0901 時，相對應的旗標將會變為 ON。位元 00 至 07 對應到 Rack 0 至 3。	唯讀
I/O 點過多旗標 (嚴重錯誤)	A40111	當基本 I/O 模組中已使用的 I/O 點編號超過 PLC 的最大容許值時將會變為 ON。	唯讀
I/O 點過多，詳情	A40700 至 A40712	I/O 點過多錯誤的三個可能原因列舉於下。 A40713 至 A40715 中的 3 位元 BIN 值表示錯誤的原因。 當設定於 I/O 表 (子局 Racks 除外) 的 I/O 點總數超過 CPU 模組的最大容許值時，I/O 點的數量將會寫入於此。 當超過 32 個中斷輸入時，中斷輸入數將會寫入於此。 當擴充 I/O Racks 的數量超過最大值時，Racks 數將會寫入於此。	唯讀
I/O 點過多原因	A40713 至 A40715	這三個位元造成 I/O 點過多錯誤的原因。(參見 A40700 至 A40712) 000 (0): I/O 點過多。 001 (1): 太多的中斷輸入點。 101 (5): 連接太多的擴充 Racks。 111 (7): 一個 Rack 子連接太多的模組 (超過 10)。	唯讀

名稱	位址	描述	存取
I/O 匯流排錯誤旗標 (嚴重錯誤)	A40114	當在 CPU 模組與安裝至插槽的模組之間傳送發生錯誤時，或底蓋沒有連接到 CPU Rack 或擴充 Rack 時將會變為 ON。	唯讀
I/O 匯流排錯誤插槽編號	A40400 至 A40407	包含發生 I/O 匯流排錯誤之 8 位元 BIN 插槽編號 (00 至 09)。底蓋沒有連接到 CPU Rack 或擴充 Rack 時內容為 0E Hex。	唯讀
I/O 匯流排錯誤 Rack 編號	A40408 至 A40415	包含發生 I/O 匯流排錯誤之 8 位元 BIN Rack 編號 (00 至 07)。	唯讀
I/O 表錯誤 (僅 CJ1-H 與 CJ1M CPU 模組)	A26100	CPU 匯流排模組設定區域初值初始化錯誤旗標 CPU 匯流排模組設定錯誤時為 ON。 當 I/O 表正常產生時變為 OFF。	唯讀
	A26102	I/O 溢位旗標 I/O 點的最大編號溢位時為 ON。 當 I/O 表正常產生時變為 OFF。	唯讀
	A26103	重覆錯誤旗標 相同的模組編號使用超過一次時為 ON。 當 I/O 表正常產生時變為 OFF。	唯讀
	A26104	I/O 匯流排錯誤旗標 I/O 匯流排錯誤時為 ON。 當 I/O 表正常產生時變為 OFF。	唯讀
	A26107	特殊 I/O 模組錯誤旗標 特殊 I/O 模組有錯誤時為 ON。 當 I/O 表正常產生時變為 OFF。	唯讀
	A26109	I/O 未確認錯誤旗標 I/O 檢測未完成時為 ON。 當 I/O 表正常產生時變為 OFF。	唯讀
重覆錯誤旗標 (嚴重錯誤)	A40113	在以下的狀況下時為 ON： 兩個 CPU 匯流排模組被指定了相同的模組編號。 兩個特殊 I/O 模組被指定了相同的模組編號。 兩個基本 I/O 模組被指定了相同的資料區塊 words。 同樣的 Rack 編號被設定到多個擴充 Rack。	唯讀
中斷輸入模組位置錯誤旗標 (僅 CJ1-H 與 CJ1M CPU 模組)	A40508	CJ1-H CPU 模組： 當中斷輸入模組沒有連到 CPU Rack 上鄰接 CPU 模組的五個位置 (槽位 (slots) 0 至 4) 之中時會為 ON。 CJ1M CPU 模組： 當中斷輸入模組沒有連到 CPU Rack 上鄰接 CPU 模組的三個位置 (槽位 (slots) 0 至 2) 之中時會為 ON。	唯讀

■ CPU 匯流排模組資訊

名稱	位址	描述	存取
CPU 匯流排模組編號重覆旗標	A41000 至 A41015	當 CPU 匯流排模組的模組編號重覆時，重覆錯誤旗標 (A40113) 與 A410 中相對應的旗標將會變為 ON。 位元 00 至 15 對應到模組編號 0 至 F。	唯讀
CPU 匯流排模組錯誤，模組編號旗標	A41700 至 A41715	當 CPU 模組與 CPU 匯流排模組間的資料交換發生錯誤時，CPU 匯流排模組錯誤旗標 (A40207) 與 A410 中相對應的旗標將會變為 ON。 位元 00 至 15 對應到模組編號 0 至 F。	唯讀
CPU 匯流排模組設定錯誤，模組編號旗標	A42700 至 A42715	當發生 CPU 匯流排模組設定錯誤時，A40203 與 A410 中相對應的旗標將會變為 ON。 位元 00 至 15 對應到模組編號 0 至 F。	唯讀
CPU 匯流排模組設定錯誤旗標 (非嚴重錯誤)	A40203	當一已安裝的 CPU 匯流排模組與 I/O 表中 CPU 匯流排模組註記不匹配時將會變為 ON。	唯讀
CPU 匯流排模組錯誤旗標 (非嚴重錯誤)	A40207	當 CPU 模組與 CPU 匯流排模組間的資料交換發生錯誤 (包括 CPU 匯流排模組本身的錯誤) 時，將會變為 ON。	唯讀

■ 特殊 I/O 模組資訊

名稱	位址	描述	存取
特殊 I/O 模組編號重覆旗標	A41100 至 A41615	當特殊 I/O 模組的模組編號有重覆時，重覆錯誤旗標 (A40113) 與 A411 至 A416 中相對應的旗標將會變為 ON。 (位元 A41100 至 A41615 對應到模組編號 0 至 95)	唯讀
特殊 I/O 模組設定錯誤旗標 (非嚴重錯誤)	A40202	當一已安裝的特殊 I/O 模組與 I/O 表中特殊 I/O 模組註記不匹配時將會變為 ON。	唯讀
特殊 I/O 模組設定錯誤，模組編號旗標	A42800 至 A43315	當發生特殊 I/O 模組設定錯誤時，A40202 與這些 words 中相對應的旗標都將會變為 ON。(位元 A42800 至 A43315 對應到模組編號 0 至 95)	唯讀
特殊 I/O 模組錯誤旗標 (非嚴重錯誤)	A40206	當 CPU 模組與一特殊 I/O 模組間的資料交換發生錯誤 (包括特殊 I/O 模組本身的錯誤) 時，將會變為 ON。	唯讀
特殊 I/O 模組錯誤，模組編號旗標	A41800 至 A42315	當 CPU 模組與一特殊 I/O 模組間的資料交換發生錯誤時，特殊 I/O 模組錯誤旗標 (A40206) 與這些 words 中相對應的旗標將會變為 ON。(位元 A42800 至 A43315 對應到模組編號 0 至 95)	唯讀

■ 其他 PLC 操作資訊

名稱	位址	描述	存取
電池錯誤旗標 (非嚴重錯誤)	A40204	若 CPU 模組的電池未連接或其電壓不足且 PLC 設定為要檢測這項錯誤時會為 ON。(檢測電池低電量)	唯讀
循環時間過長旗標 (嚴重錯誤)	A40108	若循環時間超過 PLC 設定中所設定的最大循環時間時會為 ON。在並行處理模式中，將會用到程式執行循環時間。(觀察循環時間)	唯讀
週邊服務太長旗標 (嚴重錯誤，僅 CJ1-H CPU 模組)	A40515	當並行處理模式中的週邊服務時間超過 2 s 會變為 ON。這也會造成循環時間錯誤而且操作將會停止。	唯讀
FPD 教導位元	A59800	把這個位元變為 ON 以便能以教導功能自動設定 FPD(269) 中的監控時間。	讀 / 寫
記憶體備份電池失效旗標	A39511	來自電源變為 OFF (HR、DM 等) 時維護的 I/O 記憶體區域之資料會以電池備份。若電池電壓下降且資料無法再維護，則 A39511 會變為 ON。當這個狀況發生時，I/O 記憶體中的資料將不再可靠。	唯讀

時鐘

■時鐘資訊

名稱	位址	描述	存取
時鐘資料		內藏於 CPU 模組的時鐘資料會以 BCD 的型式儲存於此。	唯讀
	A35100 至 A35107	秒：00 至 59 (BCD)	唯讀
	A35108 至 A35115	分：00 至 59 (BCD)	唯讀
	A35200 至 A35207	時：00 至 23 (BCD)	唯讀
	A35208 至 A35215	日：01 至 31 (BCD)	唯讀
	A35300 至 A35307	月：01 至 12 (BCD)	唯讀
	A35308 至 A35315	年：00 至 99 (BCD)	唯讀
	A35400 至 A35407	星期：00：星期日，01：星期一， 02：星期二，03：星期三，04：星期四， 05：星期五，06：星期六	唯讀

■電源資訊

名稱	位址	描述	存取
起動時間	A510 與 A511	這些 words 包含電源變為 ON 的時間 (以 BCD)。其內容在每次電源變為 ON 時都會更新。 A51000 至 A51007：秒 (00 至 59) A51008 至 A51015：分 (00 至 59) A51100 至 A51107：時 (00 至 23) A51108 至 A51115：日 (00 至 31)	讀 / 寫
模式改變時間	A512 與 A513	這些 words 包含斷電的時間 (以 BCD)。其內容在每次電源中斷時都會更新。 A51200 至 A51207：秒 (00 至 59) A51208 至 A51215：分 (00 至 59) A51300 至 A51307：時 (00 至 23) A51308 至 A51315：日 (00 至 31)	讀 / 寫
斷電次數	A514	包含自電源首次開啓後電源中斷的次數 (以 BIN 的型式)。要重置這個值時，請把現行值以 0000 覆寫即可。	讀 / 寫
電源 ON 總時間	A523	包含以 10 小時為單位之 PLC 為 ON 的總時間 (以 BIN 的型式)。這個資料每 10 小時會更新一次。要重置這個值時，請把現行值以 0000 覆寫即可。	讀 / 寫

快閃記憶體備份資訊

名稱	位址	描述	存取
使用者程式日期 (僅 CJ1-H 與 CJ1M CPU 模組)	A090 至 A093	這些 words 以 BCD 表示使用者程式最後一次寫入的日期與時間。 A09000 至 A09007 : 秒 (00 至 59) A09008 至 A09015 : 分 (00 至 59) A09100 至 A09107 : 時 (00 至 23) A09108 至 A09115 : 日 (00 至 31) A09200 至 A09207 : 月 (01 至 12) A09208 至 A09215 : 年 (00 至 99) A09308 至 A09307 : 星期 (00 : 星期日, 01 : 星期一, 02 : 星期二, 03 : 星期三, 04 : 星期四, 05 : 星期五, 06 : 星期六)	唯讀
參數日期 (僅 CJ1-H 與 CJ1M CPU 模組)	A094 至 A0947	這些 words 以 BCD 表示參數最後一次寫入的日期與時間。 A09400 至 A09407 : 秒 (00 至 59) A09408 至 A09415 : 分 (00 至 59) A09500 至 A09507 : 時 (00 至 23) A09508 至 A09515 : 日 (00 至 31) A09600 至 A09607 : 月 (01 至 12) A09608 至 A09615 : 年 (00 至 99) A09708 至 A09707 : 星期 (00 : 星期日, 01 : 星期一, 02 : 星期二, 03 : 星期三, 04 : 星期四, 05 : 星期五, 06 : 星期六)	唯讀

通訊

■ 網路通訊資訊

名稱	位址	描述	存取
通訊埠致能旗標	A20200 至 A20207	當網路指令 (SEND, RECV, CMND, 或 PMCR) 能與相對應的埠編號執行或可以以相對應的埠編號進行背景執行 (僅 CJ1-H 與 CJ1M CPU 模組) 時會為 ON。位元 00 至 07 對應到通訊埠 0 至 7。 當簡易備份操作用來進行 CJ1-H 或 CJ1M CPU 模組上之記憶卡的寫入或比較操作時, 會自動配置一個通訊埠, 且在操作期間相對應的旗標會變為 ON 而在操作完成後會變為 OFF。	唯讀
通訊埠完成碼	A203 至 A210	當網路指令 (SEND, RECV, CMND, 或 PMCR) 執行時, 這些 words 包含相對應的埠編號之完成碼。當背景執行完成 (僅適用於 CJ1-H 與 CJ1M CPU 模組) 時, 這些內容會被清除。words A203 至 A210 對應到通訊埠 0 至 7。 當用簡易備份操作來對 CJ1-H 或 CJ1M CPU 模組上的記憶卡進行寫入或比較時, 會自動配置一個通訊埠, 且會把完成碼儲存於相對應的 words。	唯讀
通訊埠錯誤旗標	A21900 至 A21907	當執行網路指令 (SEND, RECV, CMND, 或 PMCR) 期間發生錯誤時會為 ON。當正常完成執行時變為 OFF。位元 00 至 07 對應到通訊埠 0 至 7。 當用簡易備份操作來對 CJ1-H 或 CJ1M CPU 模組上的記憶卡進行寫入或比較時, 會自動配置一個通訊埠。若發生錯誤, 則相對應的旗標 將會變為 ON; 而若簡易備份操作正常結束, 則它將會變為 OFF。	唯讀

■週邊埠通訊資訊

名稱	位址	描述	存取
週邊埠通訊錯誤旗標	A39212	當週邊埠發生通訊錯誤時會變為 ON。	唯讀
週邊埠重新起動位元	A52601	把這個位元變為 ON 以重新起動週邊埠。	讀 / 寫
週邊埠設定改變位元	A61901	當週邊埠的通訊設定改變時會變為 ON。	讀 / 寫
週邊埠錯誤旗標	A52808 至 A52815	這些指示出週邊埠發生的是那一種錯誤。	讀 / 寫
週邊埠 PT 通訊旗標	A39400 至 A39407	當週邊埠以 NT Link 模式與 PT 構成通訊時，相對應的位元將會為 ON。位元 0 至 7 對應到模組 0 至 7。	唯讀
週邊埠 PT 優先註記旗標	A39408 至 A39415	當週邊埠以 NT Link 模式通訊時，有優先權之 PT 所對應的位元將會為 ON。位元 0 至 7 對應到模組 0 至 7。	唯讀

■RS-232C 埠通訊資訊

名稱	位址	描述	存取
RS-232C 埠通訊錯誤旗標	A39204	當 RS-232C 埠發生通訊錯誤時會變為 ON。	唯讀
RS-232C 埠重新起動位元	A52600	把這個位元變為 ON 以重新起動 RS-232C 埠。	讀 / 寫
RS-232C 埠設定改變位元	A61902	當 RS-232C 埠的通訊設定改變時會變為 ON。	讀 / 寫
RS-232C 埠錯誤旗標	A52800 至 A52807	這些指示出 RS-232C 埠發生的是那一種錯誤。	讀 / 寫
RS-232C 埠傳送備妥旗標 (no-protocol 模式)	A39205	當 RS-232C 埠能夠以 no-protocol 模式傳送資料時為 ON。	唯讀
RS-232C 埠接收完成旗標 (no-protocol 模式)	A39206	當 RS-232C 埠以 no-protocol 模式完成接收時會變為 ON。	唯讀
RS-232C 埠接收溢位旗標 (no-protocol 模式)	A39207	當透過 RS-232C 埠以 no-protocol 模式接收期間發生資料溢位時會變為 ON。	唯讀
RS-232C 埠 PT 通訊旗標	A39300 至 A39307	當 RS-232C 埠以 NT Link 模式與 PT 構成通訊時，相對應的位元將會為 ON。位元 0 至 7 對應到模組 0 至 7。	唯讀
RS-232C 埠 PT 優先註記旗標	A39308 至 A39315	當 RS-232C 埠以 NT Link 模式通訊時，有優先權之 PT 所對應的位元將會為 ON。位元 0 至 7 對應到模組 0 至 7。	唯讀
RS-232C 埠接收計數器 (no-protocol 模式)	A39300 至 A39315	指出 (以 BIN 的型式) RS-232C 埠在 no-protocol 模式所接收資料的位元組數。	唯讀

■串列裝置通訊資訊

名稱	位址	描述	存取
通訊模組 0 至 15，埠 1 至 4 設定改變位元	A62001 至 A63504	當連接埠的設定改變時，相對應的旗標將會變為 ON。 (A620 至 A635 中的位元 1 至 4 對應到通訊模組 0 至 15 的埠 1 至 4。)	讀 / 寫

指令相關資訊

名稱	位址	描述	存取
step 旗標	A20012	當以 STEP(008) 進行單步執行時會有一循環為 ON。	唯讀
現行的 EM bank (僅 CJ1 與 CJ1-H CPU 模組)	A301	這個 words 內容為 4 位元的 Hex EM bank 編號。	唯讀
巨集區域輸入 words	A600 至 A603	當 MCRO(099) 執行時，會由指定的來源 words (輸入參數 words) 複製輸入資料到 A600 至 A603。	讀 / 寫
巨集區域輸出 words	A604 至 A607	在 MCRO(099) 所指定的副程式執行之後，副程式的結果會由 A604 至 A607 傳送到指定的目的 words (輸出參數 words)。	讀 / 寫

背景執行資訊

名稱	位址	描述	存取
背景執行的 DR00 輸出 (僅 CJ1-H 與 CJ1M CPU 模組)	A597	當一資料暫存器被指定為背景執行指令的輸出時，A597 會替代 DR00 來接收輸出。 0000 至 FFFF Hex	唯讀
背景執行的 IR00 輸出 (僅 CJ1-H 與 CJ1M CPU 模組)	A595 與 A596	當一索引暫存器被指定為背景執行指令的輸出時，A595 與 A596 會替代 IR00 來接收輸出。 0000 0000 至 FFFF FFFF Hex (A596 包含最左邊的位元)	唯讀
背景執行的相等旗標 (僅 CJ1-H 與 CJ1M CPU 模組)	A59801	若 SRCH(181) 指令在背景執行時發現符合的資料，則會變為 ON。	唯讀
背景執行的 ER/AER 旗標 (僅 CJ1-H 與 CJ1M CPU 模組)	A39510	若在背景執行期間發生錯誤或非法存取，則會變為 ON。當電源變為 ON 或操作開始時會變為 OFF。	唯讀

內藏輸入的輔助區域旗標與位元

以下各表顯示與 CJ1M CPU 模組的內藏輸入相關之輔助區塊 words 與位元。這些配置僅能應用於內藏 I/O 的 CPU 模組。

■ 中斷輸入

名稱	位址	描述	讀 / 寫	資料存取時
中斷計數器 0 計數器 SV	A532	作為計數器模式中的中斷輸入 0。 設定將起動之中斷工作單的計數值。當中斷計數器 0 數到這個數量的脈衝時，中斷工作單 140 將會起動。	讀 / 寫	<ul style="list-style-type: none"> 當電源為 ON 時會維持。 當操作開始時會維持。
中斷計數器 1 計數器 SV	A533	作為計數器模式中的中斷輸入 1。 設定將起動之中斷工作單的計數值。當中斷計數器 1 數到這個數量的脈衝時，中斷工作單 141 將會起動。	讀 / 寫	
中斷計數器 2 計數器 SV	A534	作為計數器模式中的中斷輸入 2。 設定將起動之中斷工作單的計數值。當中斷計數器 2 數到這個數量的脈衝時，中斷工作單 142 將會起動。	讀 / 寫	
中斷計數器 3 計數器 SV	A535	作為計數器模式中的中斷輸入 3。 設定將起動之中斷工作單的計數值。當中斷計數器 3 數到這個數量的脈衝時，中斷工作單 143 將會起動。	讀 / 寫	
中斷計數器 0 計數器 PV	A536	這些 words 包含在計數器模式中中斷輸入操作的中斷計數器 PV。	讀 / 寫	<ul style="list-style-type: none"> 當電源為 ON 時會維持。 當操作開始時會清除。 當產生中斷時會更新。 當 INI(880) 指令執行時會更新。
中斷計數器 1 計數器 PV	A537	在遞增模式中，計數器 PV 會由 0 開始遞增。當計數器 PV 到達計數器 SV 時，PV 會自動重置為 0。	讀 / 寫	
中斷計數器 2 計數器 PV	A538	在遞減模式中，計數器 PV 會由計數器 SV 開始遞減。當計數器 PV 到達 0 時，PV 會自動重置為 SV。	讀 / 寫	
中斷計數器 3 計數器 PV	A539		讀 / 寫	

■ 高速計數器

名稱	位址	描述	讀 / 寫	資料存取時
高速計數器 0 PV	A270 至 A271	內容為高速計數器 0 的 PV。A271 為最左邊 4 位元而 A270 為最右邊 4 位元。	唯讀	<ul style="list-style-type: none"> 當電源為 ON 時會清除。 當操作開始時會清除。 在監督程序進行期間，每個循環都會更新。 當 PRV(881) 指令為相對應的計數器執行時會更新。
高速計數器 1 PV	A272 至 A273	內容為高速計數器 1 的 PV。A273 為最左邊 4 位元而 A272 為最右邊 4 位元。	唯讀	

名稱	位址	描述	讀 / 寫	資料存取時
高速計數器 0 範圍 1 比較條件滿足 旗標	A27400	這些表示當高速計數器 0 操作於範圍比較模式時， PV 是否在指定的範圍內。 0：PV 不在範圍內 1：PV 在範圍內	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 在監督程序進行期間，每個循環都會更新。 當 PRV(881) 指令為相對應的計數器執行時會更新。
高速計數器 0 範圍 2 比較條件滿足 旗標	A27401		唯讀	
高速計數器 0 範圍 3 比較條件滿足 旗標	A27402		唯讀	
高速計數器 0 範圍 4 比較條件滿足 旗標	A27403		唯讀	
高速計數器 0 範圍 5 比較條件滿足 旗標	A27404		唯讀	
高速計數器 0 範圍 6 比較條件滿足 旗標	A27405		唯讀	
高速計數器 0 範圍 7 比較條件滿足 旗標	A27406		唯讀	
高速計數器 0 範圍 8 比較條件滿足 旗標	A27407		唯讀	
高速計數器 0 比較處理中旗標	A27408	這個旗標表示高速計數器 0 是否執行比較操作模式。 0：停止 1：執行中	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當比較操作開始或停止時更新。
高速計數器 0 溢位 / 欠位旗標	A27409	這個旗標表示高速計數器 0 PV 何時發生溢位或欠位。(僅適用於計數模式設定為線性模式時。) 0：正常 1：溢位或欠位	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當 PV 改變時清除。 當發生溢位或欠位時更新。
高速計數器 0 CTBL(882) 指令執行 旗標	A27415	只有當 CTBL(882) 指令為高速計數器 0 執行時， 即比較表為高速計數器 0 註記時才會變為 ON。 為防止中斷衝突，系統在執行 INI(880) 指令 (指 定一高速計數器) 或 CTBL(882) 指令之前要先檢 查這個旗標的狀態。	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當 CTBL(882) 指令執行時更新。

名稱	位址	描述	讀 / 寫	資料存取時
高速計數器 1 範圍 1 比較條件滿足 旗標	A27500	這些表示當高速計數器 1 操作於範圍比較模式時，PV 是否在指定的範圍內。 0：PV 不在範圍內 1：PV 在範圍內	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 在監督程序進行期間，每個循環都會更新。 當 PRV(881) 指令為相對應的計數器執行時會更新。
高速計數器 1 範圍 2 比較條件滿足 旗標	A27501		唯讀	
高速計數器 1 範圍 3 比較條件滿足 旗標	A27502		唯讀	
高速計數器 1 範圍 4 比較條件滿足 旗標	A27503		唯讀	
高速計數器 1 範圍 5 比較條件滿足 旗標	A27504		唯讀	
高速計數器 1 範圍 6 比較條件滿足 旗標	A27505		唯讀	
高速計數器 1 範圍 7 比較條件滿足 旗標	A27506		唯讀	
高速計數器 1 範圍 8 比較條件滿足 旗標	A27507		唯讀	
高速計數器 1 比較處理中旗標	A27508	這個旗標表示高速計數器 1 是否執行比較操作模式。 0：停止。 1：執行中	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當比較操作開始或停止時更新。
高速計數器 1 溢位 / 欠位旗標	A27509	這個旗標表示高速計數器 1 PV 何時發生溢位或欠位。(僅適用於計數模式設定為線性模式時。) 0：正常 1：溢位或欠位	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當 PV 改變時清除。 當發生溢位或欠位時更新。
高速計數器 0 重置位元	A53100	當重置方法設定為相位 Z 訊號 + 軟體重置時，若在這個位元為 ON 時接收到相位 Z 訊號，則相對應的高速計數器的 PV 將會重置。 當重置方法設定為軟體重置時，在此位元由 OFF 變為 ON 的循環中，相對應的高速計數器的 PV 將會重置。	讀 / 寫	<ul style="list-style-type: none"> 當電源變為 ON 時清除。
高速計數器 1 重置位元	A53101		讀 / 寫	
高速計數器 0 闔道位元	A53102	當一計數器的闔道位元為 ON 時，即使計數器接收到脈衝輸入，計數器的 PV 也不會改變。	讀 / 寫	<ul style="list-style-type: none"> 當電源變為 ON 時清除。
高速計數器 1 闔道位元	A53103	當位元再次變為 OFF，將會再開始計數，且高速計數器的 PV 也將會更新。 當重置方法設定為 Z 相訊號 + 軟體重置時，在相對應的重置位元 (A53100 或 A53101) 為 ON 時，闔道位元為除能。	讀 / 寫	

內藏輸出的輔助區域旗標與位元

以下各表顯示與 CJ1M CPU 模組的內藏輸出相關之輔助區塊 words 與位元。這些配置僅能應用於內藏 I/O 的 CPU 模組。

名稱	位址	描述	讀 / 寫	資料存取時
脈衝輸出 0 PV	A276 至 A277	內容為相對應的脈衝輸出埠之脈衝輸出數。 PV 範圍：80000000 至 7FFFFFFF Hex	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 在監督程序進行期間，每個循環都會更新。 當 INI(880) 指令為相對應的脈衝輸出執行時會更新。
脈衝輸出 1 PV	A278 至 A279	(-2,147,483,648 至 2,147,483,647) 當脈衝以順時針方向輸出，PV 會每個脈衝加 1。 當脈衝以反時針方向輸出，PV 會每個脈衝減 1。 溢位後的 PV：7FFFFFFF Hex 欠位後的 PV：80000000 Hex A277 為脈衝輸出 0 PV 的最左邊 4 位元而 A276 為其最右邊 4 位元。 A279 為脈衝輸出 1 PV 的最左邊 4 位元而 A278 為其最右邊 4 位元。 附註 若座標系統為相對座標（未定義原點），則當脈衝輸出開始時，即當脈衝輸出指令 (SPED(885), ACC(888), 或 PLS2(887)) 執行時，PV 將會清除為 0。		
脈衝輸出 0 加速/減速旗標	A28000	當脈衝依據 ACC(888) 或 PLS2(887) 指令由脈衝輸出 0 輸出且輸出頻率逐步改變（加速或減速）時，這個旗標將會為 ON。 0：固定速度 1：加速或減速	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 在監督程序進行期間，每個循環都會更新。
脈衝輸出 0 溢位 / 欠位旗標	A28001	這個旗標表示脈衝輸出 0 PV 何時發生溢位或欠位。 0：正常 1：溢位或欠位	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當 PV 被 INI(880) 指令改變時清除。 當發生溢位或欠位時更新。
脈衝輸出 0 輸出量設定旗標	A28002	當脈衝輸出 0 的輸出脈衝數已被以 PULS(886) 指令設定時為 ON。 0：未設定 1：已設定	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 當 PULS(886) 指令執行時更新。 當脈衝輸出停止時更新。
脈衝輸出 0 輸出完成旗標	A28003	當以 PULS(886) 指令設定的輸出脈衝數已被脈衝輸出 0 輸出時為 ON。 0：輸出未完成 1：輸出完成	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 當以獨立模式開始或完成脈衝輸出時更新。
脈衝輸出 0 輸出處理中旗標	A28004	當脈衝由脈衝輸出 0 輸出時為 ON。 0：停止 1：輸出脈衝	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 當脈衝輸出開始或停止時更新。

名稱	位址	描述	讀 / 寫	資料存取時
脈衝輸出 0 無原點旗標	A28005	當脈衝輸出 0 的原點未建立時為 ON，而當原點已建立時則為 OFF。 0：原點已建立。 1：原點未建立。	唯讀	<ul style="list-style-type: none"> 當打開電源時變為 ON。 當操作開始時會變為 ON。 當脈衝輸出開始或停止時更新。 在監控程序進行期間每個循環都更新。
脈衝輸出 0 在原點旗標	A28006	當脈衝輸出 PV 符合原點 (0) 時為 ON。 0：不停止於原點。 1：停止於原點。	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 在監控程序進行期間每個循環都更新。
脈衝輸出 0 輸出停止錯誤旗標	A28007	在脈衝輸出 0 原點搜尋功能中輸出脈衝時發生錯誤則為 ON。 脈衝輸出 0 輸出停止錯誤碼將會寫入 A444。 0：無錯誤 1：發生停止錯誤	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當原點搜尋開始時更新。 當發生脈衝輸出停止錯誤時更新。
脈衝輸出 1 加速 / 減速旗標	A28100	當脈衝依據 ACC(888) 或 PLS2(887) 指令由脈衝輸出 1 輸出且輸出頻率逐步改變 (加速或減速) 時，這個旗標將會為 ON。 0：固定速度 1：加速或減速	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 在監督程序進行期間，每個循環都會更新。
脈衝輸出 1 溢位 / 欠位旗標 (Overflow/Underflow Flag)	A28101	這個旗標表示脈衝輸出 1 PV 何時發生溢位或欠位。 0：正常 1：溢位或欠位	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始時清除。 當 PV 被 INI(880) 指令改變時清除。 當發生溢位或欠位時更新。
脈衝輸出 1 輸出量設定旗標	A28102	當脈衝輸出 1 的輸出脈衝數已被以 PULS(886) 指令設定時為 ON。 0：未設定 1：已設定	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 當 PULS(886) 指令執行時更新。 當脈衝輸出停止時更新。
脈衝輸出 1 輸出完成旗標	A28103	當以 PULS(886) 指令設定的輸出脈衝數已被脈衝輸出 1 輸出時為 ON。 0：輸出未完成 1：輸出完成	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 當 PULS(886) 指令執行時更新。 當以獨立模式開始或完成脈衝輸出時更新。
脈衝輸出 1 輸出處理中旗標	A28104	當脈衝由脈衝輸出 1 輸出時為 ON。 0：停止 1：輸出脈衝	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。 當脈衝輸出開始或停止時更新。

名稱	位址	描述	讀 / 寫	資料存取時
脈衝輸出 1 無原點旗標	A28105	當脈衝輸出 1 的原點未建立時為 ON, 而當原點已建立時則為 OFF。 0: 原點已建立。 1: 原點未建立。	唯讀	<ul style="list-style-type: none"> 當打開電源時變為 ON。 當操作開始時會變為 ON。 當脈衝輸出開始或停止時更新。 在監控程序進行期間每個循環都更新。
脈衝輸出 1 在原點旗標	A28106	當脈衝輸出 PV 符合原點 (0) 時為 ON。 0: 不停止於原點。 1: 停止於原點。	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 在監控程序進行期間每個循環都更新。
脈衝輸出 1 輸出停止錯誤旗標	A28107	在脈衝輸出 1 原點搜尋功能中輸出脈衝時發生錯誤則為 ON。 脈衝輸出 1 輸出停止錯誤碼將會寫入 A445。 0: 無錯誤 1: 發生停止錯誤	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當原點搜尋開始時更新。 當發生脈衝輸出停止錯誤時更新。
PWM(891) 輸出 0 輸出處理中旗標	A28300	當脈衝由 PWM(891) 輸出 0 輸出時為 ON。 0: 停止 1: 輸出脈衝	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。 當操作開始或停止時清除。
PWM(891) 輸出 1 輸出處理中旗標	A28308	當脈衝由 PWM(891) 輸出 1 輸出時為 ON。 0: 停止 1: 輸出脈衝	唯讀	<ul style="list-style-type: none"> 當脈衝輸出開始或停止時更新。
脈衝輸出 0 停止錯誤碼	A444	當脈衝輸出 0 發生脈衝輸出停止錯誤時, 相對應的錯誤碼會寫入這個 words。	唯讀	<ul style="list-style-type: none"> 當電源變為 ON 時清除。
脈衝輸出 1 停止錯誤碼	A445	當脈衝輸出 1 發生脈衝輸出停止錯誤時, 相對應的錯誤碼會寫入這個 words。	唯讀	<ul style="list-style-type: none"> 當原點搜尋開始時更新。 當發生脈衝輸出停止錯誤時更新。
脈衝輸出 0 重置位元	A54000	當這個位元由 OFF 變為 ON 時, 脈衝輸出 0 PV (包含於 A276 與 A277) 會被清除。	讀 / 寫	<ul style="list-style-type: none"> 當電源變為 ON 時清除。
脈衝輸出 0 CW 限動輸入訊號旗標	A54008	這是脈衝輸出 0 的順時針方向限動輸入訊號, 用於原點搜尋。要使用這個訊號, 請由真正的感測器寫入輸入作為階梯程式的輸入條件並把結果輸出至這個旗標。	讀 / 寫	當打開電源時清除。
脈衝輸出 0 CCW 限動輸入訊號旗標	A54009	這是脈衝輸出 0 的反時針方向限動輸入訊號, 用於原點搜尋。要使用這個訊號, 請由真正的感測器寫入輸入作為階梯程式的輸入條件並把結果輸出至這個旗標。	讀 / 寫	
脈衝輸出 1 重置位元	A54100	當這個位元由 OFF 變為 ON 時, 脈衝輸出 1 PV (包含於 A278 與 A279) 會被清除。	讀 / 寫	
脈衝輸出 1 CW 限動輸入訊號旗標	A54108	這是脈衝輸出 1 的順時針方向限動輸入訊號, 用於原點搜尋。要使用這個訊號, 請由真正的感測器寫入輸入作為階梯程式的輸入條件並把結果輸出至這個旗標。	讀 / 寫	
脈衝輸出 1 CCW 限動輸入訊號旗標	A54109	這是脈衝輸出 1 的反時針方向限動輸入訊號, 用於原點搜尋。要使用這個訊號, 請由真正的感測器寫入輸入作為階梯程式的輸入條件並把結果輸出至這個旗標。	讀 / 寫	

串列 PLC Link (僅 CJ1M CPU 模組)

名稱	位址	功能說明	讀 / 寫	時間反應
RS-232C 埠通訊錯誤旗標	A39204	ON 當 RS-232C 埠發生通訊錯誤時 ON：錯誤 OFF：正常	唯讀	當打開電源時清除。 當 RS-232C 埠發生通訊錯誤時為 ON。 當通訊埠重新起動時為 OFF。 在 Tool bus 模式與 NT Link 模式時除能。
RS-232C 埠 PT 通訊旗標	A39300 至 A39307	當 RS-232C 埠以 NT Link 模式與一 PT 通訊時，相對應的位元將會為 ON。位元 0 至 7 對應到模組 0 至 7。 ON：通訊 OFF：無通訊	唯讀	當打開電源時清除。 當 RS-232C 埠在 NT Link 模式或串列 PLC Link 模式時，對應到通訊 PT 或子局子局的位元會變為 ON。 位元 0 至 7 對應到模組 0 至 7。
RS-232C 埠重新起動位元	A52600	把這個位元變為 ON 以重新起動 RS-232C 埠。	讀 / 寫	當打開電源時清除。 把這個位元變為 ON 以重新起動 RS-232C 埠。 在重新起動處理完成後，會自動被系統變為 OFF。
RS-232C 埠錯誤旗標	A52800 至 A52807	這些表示 RS-232C 埠曾發生那一種的錯誤。 位元 0：未使用 位元 1：未使用 位元 2：同位元錯誤 位元 3：框架錯誤 位元 4：超越錯誤 位元 5：逾時錯誤 位元 6：未使用 位元 7：未使用	讀 / 寫	當打開電源時清除。 當 RS-232C 埠發生錯誤時，錯誤碼會儲存起來。 在 Tool bus 模式時會除能。 在 NT Link 模式時只有位元 5 (逾時錯誤) 會致能。 在串列 PLC Link 模式時只有以下位元會致能： 輸出模組： 位元 5：逾時錯誤 被輸出模組： 位元 5：逾時錯誤 位元 4：超越錯誤 位元 3：框架錯誤
RS-232C 埠設定改變位元	A61902	當 RS-232C 埠的通訊設定改變時為 ON。 ON：改變 OFF：未改變	讀 / 寫	當打開電源時清除。 當 RS-232C 埠通訊設定開始改變時為 ON。 當 STUP(237) 執行時為 ON；而設定改變後為 OFF。

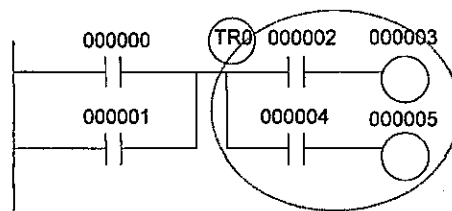
7-11 TR (暫時記憶繼電器) 區域

TR 區域包含位址範圍為 TR0 至 TR15 的 16 位元。它們為程式的分歧暫時儲存指令的 ON/OFF 狀態。當有數個輸出分歧與連鎖不能使用时，TR 位元是有用的。TR 位元的使用次數與順序沒有限制，只要同樣的 TR 位元沒有在同一個指令區塊中使用兩次以上即可。

TR 位元只能使用於 OUT 與 LD 指令。OUT 指令 (OUT TR0 至 OUT TR15) 儲存分歧點的 ON OFF 狀態，而 LD 指令則是叫出所儲存的分歧點 ON OFF 狀態。TR 位元不能由 CX-P 軟體改變。

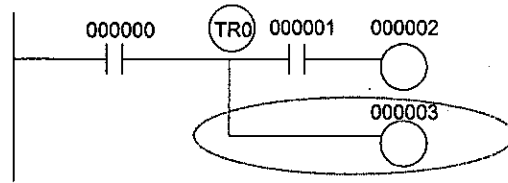
範例

在這個例子中，TR 位元使用於兩個輸出直接連接到一個分歧點的狀況。



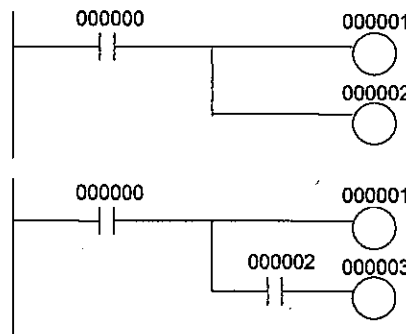
指令	運算元
LD	000000
OR	000001
OUT	TR 0
AND	000002
OUT	000003
LD	TR 0
AND	000004
OUT	000005

在這個例子中，TR 位元使用於一個輸出連接到一個沒有個別執行條件的分岐點之狀況。



指令	運算元
LD	000000
OUT	TR 0
AND	000001
OUT	000002
LD	TR 0
OUT	000003

附註 在分岐點後沒有執行條件或只有指令區塊的最後一行有執行條件的狀況下，是不需要 TR 位元的。



指令	運算元
LD	000000
OUT	000001
OUT	000002

指令	運算元
LD	000000
OUT	000001
AND	000002
OUT	000003

7-12 計時器區域

4,096 計時器 編號 (T0000 至 T4095) 由 TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TIMHHX(552), TTIM(087), TTIMX(555), TIMW(813), TIMWX(816), TMHW(815), 與 TIMHWX(817) 指令所共同分。這些指令的計時器完成旗標與顯示值 (PV) 是以計時器編號來存取的。(TIML(542), TIMLX(553), MTIM(543), 與 MTIMX(554) 指令不使用計時器編號。)

當計時器編號使用於一個需要位元資料的運算元時，計時器編號存取計時器的完成旗標。當計時器編號使用於一個需要 words 資料的運算元時，計時器編號存取計時器的 PV。只要需要即可使用計時器完成旗標作為常開與常閉條件，且計時器 PV 的值可以當作一般 words 資料來讀取。

在 CJ1-H 與 CJ1M CPU 模組中，計時器 PV 的更新方法可以由 CX-P 軟體設定為 BCD 或 BIN。對於 CJ1 CPU 模組中，它只能夠設定為 BIN。

附註 在兩個計時器指令中不建議使用相同的計時器編號，這是因為若它們同時計時則計時器將無法正確地動作。

(若兩個以上的計時器指令使用同一個計時器編號，在程式檢查時將會產生錯誤，但只要指令沒有在同一個循環內執行，則計時器還是可以動作。)

下表顯示何時計時器 PV 與完成旗標將會重置。

指令名稱	PV 與完成旗標的影響			在跳躍與連鎖中操作	
	模式改變 (註 1.)	PLC 起動 (註 1.)	CNR(545)/ CNRX(547)	跳躍 (JMP-JME) 或待命中的工作單	連鎖 (IL-ILC)
TIMER: TIM/TIMX(550)	PV → 0 旗標 → OFF	PV → 0 旗標 → OFF	PV → 9999 旗標 → OFF	PV 在操作計時器 中更新	PV = SV (重置為 SV.) 旗標 → OFF
HIGH-SPEED TIMER : TIMH(015)/TIMHX(551)					
ONE-MS TIMER : TMHH(540)/TMHHX(552)					
ACCUMULATIVE TIMER : TTIM(087)/TTIMX(555)				PV 維持	PV 維持
TIMER WAIT : TIMW(813)/TIMWX(816)				PV 在操作計時器 中更新	--
HIGH-SPEED TIMER WAIT : TMHW(815)/TMHWX(817)					--

- 附註
1. 若 I/O memory 保持位元 (A50012) 為 ON，當發生嚴重錯誤或操作模式由 PROGRAM 改變為 RUN 或 MONITOR 模式，或是反過來的狀況時，PV 與完成旗標將會維持。當電源 ON → OFF → ON 時，PV 與完成旗標將會清除。
 2. 若 IOM 保持位元 (A50012) 為 ON 且 PLC 設定的 " 起動時之 IOM 保持位元狀態 " 設定為保護 IOM 保持位元，則當 PLC 的電源循環時，Link 區域的內容將會維持。
 3. 由於 TIML(542), TIMLX(553), MTIM(543), 與 MTIMX(554) 指令不使用計時器編號，它們會在不同的條件下重置。詳情請參見這些指令的描述。
 4. 以計時器編號 0000 至 2047 規劃的 TIM, TIMX(550), TIMH(015), TIMHX(551), TMHH(540), TMHHX(552), TIMW(813), TIMWX(816), TMHW(815) 與 TMHWX(817) 計時器之顯示值於在 JMP 與 JME 指令之間跳躍或於待命中的工作單時將會更新。以計時器編號 2048 至 4095 規劃的計時器之 PV 值在跳躍或於待命中的工作單中將會保持。

計時器完成旗標可以強制設定與強制重置。

計時器 PV 不能強制設定與強制重置，雖然 PV 可以間接由強制設定 / 重置完成旗標更新。

計時器編號使用的順序或其可以在程式中當作常開或常閉使用的次數是沒有限制的。計時器 PV 可以當作 words 資料讀取與在程式中使用。

7-13 計數器區域

4,096 計數器編號 (C0000 至 C4095) 由 CNT, CNTX(546), CNTR(012), CNTRX(548), CNTW(814) 與 CNTWX(818) 指令所共同分享。這些指令的計數器完成旗標與顯示值 (PV) 是以計數器編號來存取。

當一計數器編號使用於一個需要位元資料的運算元時，計數器編號存取計數器的完成旗標。當計數器編號使用於一個需要 words 資料的運算元時，計數器編號存取計數器的 PV。

在 CJ1-H 與 CJ1M CPU 模組中，計數器 PV 的更新方法可以由 CX-P 軟體設定為 BCD 或 BIN。對於 CJ1 CPU 模組中，它只能夠設定為 BIN。

在兩個計數器指令中不建議使用相同的計數器編號，這是因為若它們同時計數則計數器將無法正確地動作。若兩個以上的計數器指令使用同一個計數器編號，在程式檢查時將會產生錯誤，但只要指令沒有在同一個循環內執行，則計數器還是可以動作。

下表顯示何時計數器 PV 與完成旗標將會重置。

指令名稱	PV 與完成旗標的影響					
	重置	模式改變	PLC 起動	重置輸入	CNR(545)/ CNRX(547)	連鎖 (IL-ILC)
COUNTER : CNT/ CNTX(546)	PV → 0000 旗標 → OFF	維持	維持	重置	重置	維持
REVERSIBLE COUNTER : CNTR(012)/ CNTRX(548)						
COUNTER WAIT : CNTW(814)/CNTWX(818)						

計數器的完成旗標可以被強制設定與強制重置。

計數器的現在值，不可以強制 set 及 Reset，(但是依計數器完成旗標的強制 set/Reset 可間接地更新)。

計數器編號使用的順序或其可以在程式中當作常開或常閉使用的次數是沒有限制的。計數器的 PV 可以在程式中以通道 (CH) 方式來讀取數值。

7-14 資料記憶體 (DM) 區域

DM 區域包含位址範圍為 D00000 至 D32767 的 32,768 words。這個資料區域是作為一般資料儲存與操縱之用，並且只能以 words 存取。

當 PLC 的電源 ON → OFF → ON 或 PLC 的操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，或是反過來的狀況時，DM 區域中的資料將會維持。

雖然 DM 區域中的位元無法被直接存取，這些位元的狀態可以用 BIT TEST 指令 TST(350) 與 TSTN(351) 來存取。

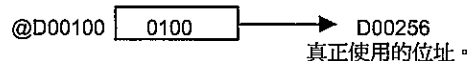
DM 區域中的位元不能強制設定與強制重置。

間接定址

DM 區域中的 words 可以用兩種方式來間接定址：BIN 模式與 BCD 模式。

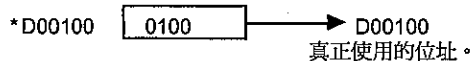
BIN 模式定址 (@D)

當 DM 位址前加上一個 "@" 符號時，DM words 的內容被視為 BIN，且指令將在該 BIN 位址的 DM words 上操作。可以用 Hex 數 0000Hex 至 7FFFHex 來對整個 DM 區域 (D00000 至 D32767) 進行間接定址。



BCD 模式定址 (*D)

當 DM 位址前加上一個 "*" 符號時，DM words 的內容被視為 BCD，且指令將在該 BCD 位址的 DM words 上操作。可以用 BCD 數 0000 至 9999 來對部份的 DM 區域 (D00000 至 D09999) 進行間接定址。



特殊模組 DM 區的配置

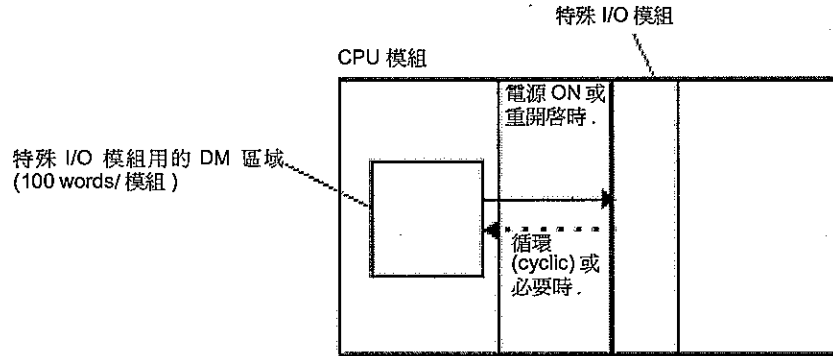
部份的 DM 區域為了初始模組設定的功能配置到特殊 I/O 模組與 CPU 匯流排模組。這些模組之資料傳送時序是不同的，不過將是以下的三種狀況之一。

- 1,2,3... 1. 當 PLC 的電源變為 ON 或模組重新起動時傳送資料。
2. 每個循環傳送一次資料。
3. 資料的傳送。

有關資料傳送時序的詳情，請參見模組的操作手冊。

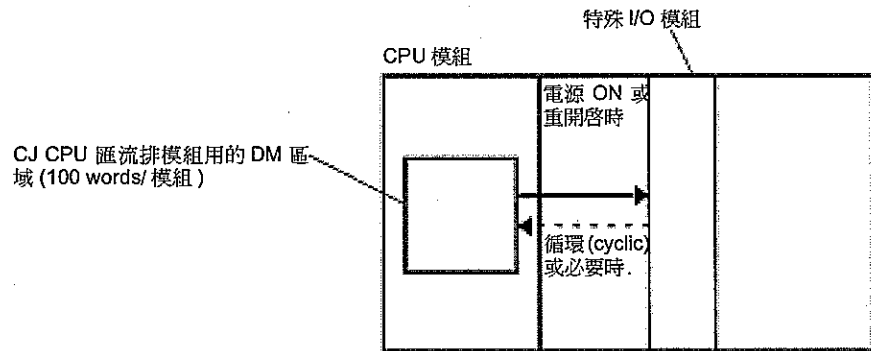
特殊 I/O 模組 (D20000 至 D29599)

每個特殊 I/O 模組配置 100 words (依據模組編號 0 至 95)。有關這些 words 的功能之詳情，請參見模組的操作手冊。



CPU 匯流排模組 (D30000 至 D31599)

每個 CPU 匯流排模組配置 100 words (依據模組編號 0 至 F)。有關這些 words 的功能之詳情，請參見模組的操作手冊。有些諸如乙太網路模組的 CPU 匯流排模組，初始設定必需登錄於 CPU 模組的參數區域；這個資料可以用程式書寫器以外的 CX-P 軟體來加以登錄。



7-15 擴充資料記憶體 (EM) 區域

僅 CJ1 與 CJ1-H CPU 模組支援 EM 區域。共區分為 13 bank (0 至 2)，每個 bank 包含 32,768 words。EM 區域位址範圍為 E0_00000 至 E2_32767。這個資料區域是作為一般資料儲存與操縱之用，並且只能以 words 存取。

PLC 的電源 ON → OFF → ON 或 PLC 的操作模式由 PROGRAM 改變為 RUN/MONITOR 模式，或是反過來的狀況時，EM 區域中的資料會維持不變。

雖然 EM 區域中的位元無法被直接存取，這些位元的狀態可以用 BIT TEST 指令 TST(350) 與 TSTN(351) 來存取。

EM 區域中的位元不能強制設定與強制重置。

指定 EM 位址

有兩種方法可以指定 EM 位址：可同時指定 bank 與位址，或指定現行 bank 中的一個位址 (必要時要先改變現行的 bank)。一般而言，我們會建議同時指定 bank 與位址。

1,2,3...

1. bank 與位址之指定

用這個方法時，要在 EM 位址之前先指定 bank 編號。舉例來說，E2_00010 表示 EM 位址為 bank 2 中的 00010。

2. 現行 bank 的位址之指定

用這個方法時，只要指定 EM 位址。舉例來說，E00010 表示 EM 位址為現行 bank 中的 00010。(必需以 EMBC(281) 來改變現行的 bank 以存取另一 bank 的資料。A301 內容為現行 EM bank 的編號。)

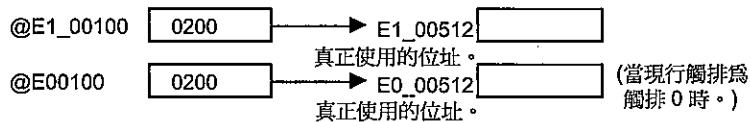
當操作模式由 PROGRAM 模式改變為 RUN/MONITOR 模式時，除非 IOM 保持位元 (A50012) 為 ON，否則現行的 bank 將會重置為 0。當程式在循環工作單中進行時，現行的 bank 不會改變；若程式在中斷工作單中改變了，則現行的 bank 將會回到它原來的值 (在原始的循環工作單中)。

間接定址

EM 區域中的 words 可以用兩種方式來間接定址：BIN 模式與 BCD 模式。

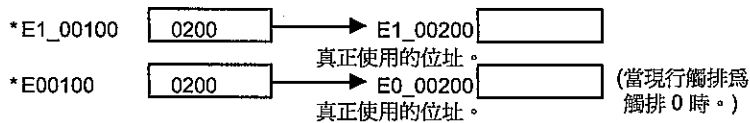
BIN 模式定址 (@E)

當 EM 位址前加上一個 "@" 符號時，EM words 的內容被視為 BIN，且指令將在與該 BIN 位址相同 bank 之 EM words 上操作。EM bank (E00000 至 E32767) 中所有的 words 可以用 0000Hex 至 7FFFHex 來間接定址，下一個 EM bank (E00000 至 E32767) 可以用 8000Hex 至 FFFFHex 來間接定址。



BCD 模式定址 (*E)

當 EM 位址前加上一個 "*" 符號時，EM words 的內容被視為 BCD，且指令將在與該 BIN 位址相同 bank 之 EM words 上操作。只有部份的 EM 區域 (E00000 至 E09999) 可以用 BCD 數 0000 至 9999 來進行間接定址。



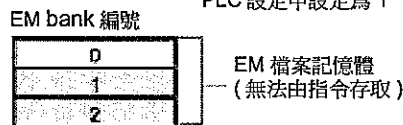
檔案記憶體轉換

部份的 EM 區域可以被 PLC 設定中的設定轉換為檔案記憶體使用。由指定的 bank (EM 檔案記憶體起動 bank) 開始至最後的 EM bank 間所有的 EM bank 都被轉換為檔案記憶體。

一旦 EM bank 被轉換為檔案記憶體，它們就無法被指令所存取 (讀取或寫入)。若指定一檔案記憶體 bank 為指令中的運算元，則將會發生非法存取錯誤。

以下的範例顯示當 EM 檔案記憶體起動 bank 已經在 PLC 設定中設定為 1 時的 EM 檔案記憶體。

範例：
EM 檔案記憶體起動 bank 在 PLC 設定中設定為 1



7-16 工作單旗標 (Task Flags)

工作單旗標範圍為 TK00 至 TK31 並對應到循環工作單 0 至 31。當相對應的循環工作單在可執行 (RUN) 狀態時，工作單旗標將會為 ON；當循環工作單還沒有被執行 (INI) 或在待命 (WAIT) 狀態時，工作單旗標將會為 OFF。

附註 這些僅表示循環工作單的狀態，不能反應中斷工作單的狀態。

工作單旗標初值化

在以下的狀況下，不論 IOM 保持位元為何狀態，工作單旗標將會被清除。

- 1,2,3... 1. 操作模式由 PROGRAM 模式變為 RUN/MONITOR 模式。
2. PLC 的電源由 ON → OFF → ON。

強制位元狀態

工作單旗標 (A50012) 不能被強制設定或強制重置。

7-17 條件旗標 (Condition Flags)

這些旗標包括表示指令執行結果的錯誤旗標與相等旗標之類的算術旗標。在早期的 PLC 中，這些都是在 SR 區域中。

條件旗標是以 CY 與 ER 之類的標記、或以 P_Carry 與 P_Instr_Error 之類的符號來指定的，而不是以位址指定的。這些旗標的狀態反應指令執行的結果，不過都是唯讀的；它們不能夠以指令或 CX-P 軟體 (CX-P 軟體或程式書寫器) 直接寫入，只能由 CX-P 軟體中叫出這些旗標，規劃於程式中。

附註 CX-P 軟體把條件旗標全都視為以 P_ 開頭的全域符號。

當程式切換工作單時，所有的條件旗標都會清除，所以 ER 與 AER 旗標的狀態。條件旗標不能被強制設定與強制重置。

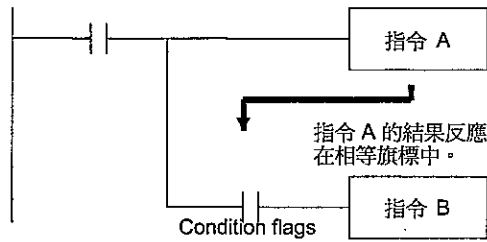
條件旗標一覽表

下表總結條件旗標的功能，雖然這些旗標的功能會因指令的不同而有些許的差異。有關特殊指令之條件旗標操作的詳情，請參見該指令的描述。

名稱	標記	符號	功能
錯誤旗標	ER	P_ER	當指令中的運算元資料不正確 (指令處理錯誤) 時會變為 ON，以表示因有錯誤而造成指令結束。 當 PLC 設定為在指令錯誤 (指令錯誤操作) 時停止操作，則在錯誤旗標變為 ON 時，程式將會停止執行且指令處理錯誤旗標 (A29508) 也會變為 ON。
存取錯誤旗標	AER	P_AER	當發生非法存取錯誤時會變為 ON。非法存取錯誤表示一指令意圖存取一塊不應被存取的記憶體區域。 當 PLC 設定為在指令錯誤 (指令錯誤操作) 時停止操作，則在存取錯誤旗標變為 ON 時，程式將會停止執行且指令處理錯誤旗標 (A429510) 也會變為 ON。
進位旗標	CY	P_CY	當算術運算的結果有進位或有 "1" 被資料移位指令移位到進位旗標時會變為 ON。 進位旗標是可以與移位指令、四則運算運算指令，搭配運用。
大於旗標	>	P_GT	當比較指令的第一個運算元大於第二個或其中一個值超過指定範圍時會變為 ON。
相等旗標	=	P_EQ	當比較指令的兩個運算元相等或演算結果為 0 時會變為 ON。
小於旗標	<	P_LT	當比較指令的第一個運算元小於第二個或其中一個值低於指定範圍時會變為 ON。
負數旗標	N	P_N	當結果的最高有效位元 (符號位元) 為 ON 時會變為 ON。
溢位旗標	OF	P_OF	當計算結果超過結果 words 的容量時會變為 ON。
欠位旗標	UF	P_UF	當計算結果不足結果 words 的容量時會變為 ON。
大於等於旗標	>=	P_GE	當比較指令的第一個運算元大於等於第二個時會變為 ON。
不等於	<>	P_NE	當比較指令的兩個運算元不相等時會變為 ON。
小於等於旗標	<=	P_LE	當比較指令的第一個運算元小於等於第二個時會變為 ON。
常時 ON 旗標	ON	P_On	永遠為 ON。(永遠為 1.)
常時 OFF 旗標	OFF	P_Off	永遠為 OFF。(永遠為 0.)

使用條件旗標

條件旗標是由所有的指令所共同分享的。所以它們的狀態可能在單一循環就會改變。務必在指令執行後，最好在同一個執行條件的分歧中，立即設置條件旗標。



指令	運算元
LD	
指令 A	
AND	=
指令 B	

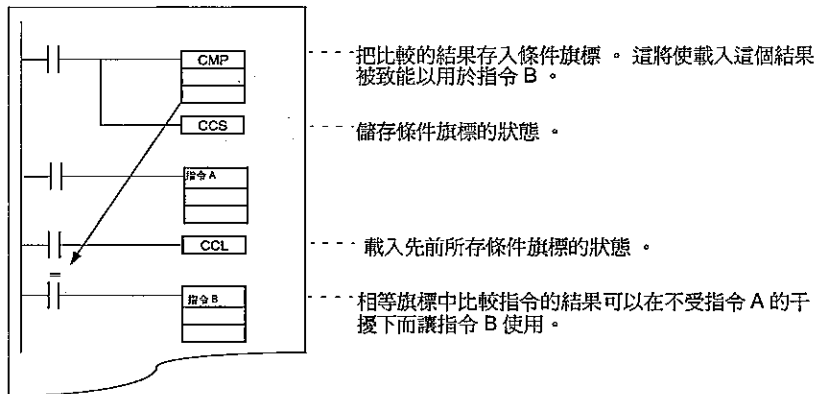
由於條件旗標是由所有的指令所分享，單一工作單的中斷就可能讓程式操作由原來所預期的順序改變到另一個順序。在撰寫程式時，務必要考慮清楚中斷可能造成的影響。詳情請參見 第 2 章 CS/CJ 系列可程式控制器程式手冊(W394)。當程式切換工作單時會清除條件旗標，所以條件旗標的狀態無法傳送到另一個工作單。舉例來說，工作單 2 無法讀取工作單 1 中旗標的狀態。

儲存與載入條件旗標狀態

CJ1-H 與 CJ1M CPU 模組支援儲存及載入條件旗標狀態 (CCS(282) 與 CCL(283)) 的指令。這些可用來存取工作單中其他位置或不同工作單中的條件旗標之狀態。

以下的範例顯示如何在同一工作單中的不同位置使用相等旗標。

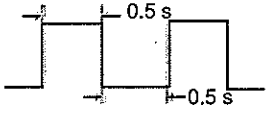
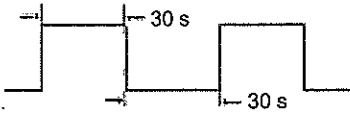
Task



7-18 時序脈衝

時序脈衝是會被系統在規律的時間間隔下變為 ON 與 OFF 的旗標。

名稱	標記	符號	操作	
0.02 s 時序脈衝	0.02s	P_0_02_s		ON 維持 0.01 s OFF 維持 0.01 s
0.1 s 時序脈衝	0.1s	P_0_1s		ON 維持 0.05 s OFF 維持 0.05 s
0.2 s 時序脈衝	0.2s	P_0_2s		ON 維持 0.1 s OFF 維持 0.1 s

名稱	標記	符號	操作	
1 s 時序脈衝	1s	P_1s		ON 維持 0.5 s OFF 維持 0.5 s
1 分鐘時序脈衝	1min	P_1min		ON 維持 30 s OFF 維持 30 s

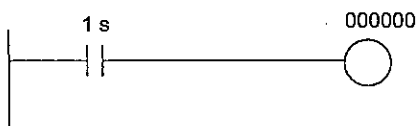
時序脈衝是以標記 Label (或符號) 而不是以位址來指定的。

附註 CX-P 軟體把條件旗標全都視為以 P_ 開頭的全域符號。

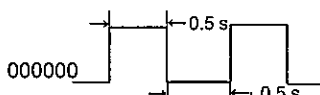
時序脈衝是唯讀的；它們不能被指令或 CX-P 軟體 (CX-P 軟體或程式書寫器) 所覆寫。

時序脈衝會在操作開始時清除。

使用時序脈衝



以下的範例在 0.5 s 間隔中把 CUT 000000 作 ON / OFF 的輸出。



指令	運算元
LD	1 s
OUT	000000

7-19 參數區域

參數區域不像 I/O 記憶體中的資料區域般可以使用於指令運算元，它只能由 CX-P 軟體所存取。參數區域由以下幾個部份所組成。

- PLC 設定
- 登錄 I/O 表
- 路由表 (Routing Tables)
- CPU 高機能模組系統設定區

7-19-1 PLC 設定

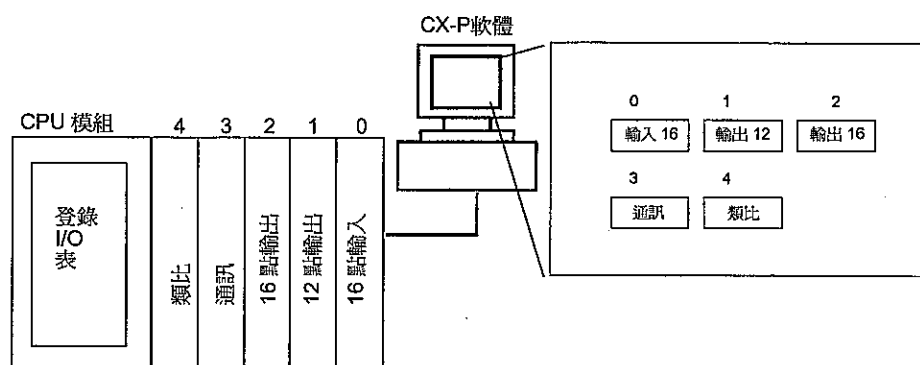
使用者可以以 CX-P 來定義 CPU 模組的基本規格。PLC 設定包含 串列埠通訊設定與最小循環時間 cycle time 設定等各項設定。

附註 有關於改變這些設定的詳情，請參見 CX-P 軟體的操作手冊。

7-19-2 登錄 I/O 表

登錄 I/O 表為 CPU 模組中包含所有安裝到 CPU Rack 與擴充 Rack 的模組之型號與插槽位置等資訊的一個表。I/O 表是以 CX-P 軟體之操作來寫入 CPU 模組的。

CPU 模組依據登錄 I/O 表中的資料把 I/O 記憶體配置到基本 I/O 模組與 CPU 匯流排模組上的 I/O 點。有關註記 I/O 表的詳情，請參見 CX-P 軟體的操作手冊。



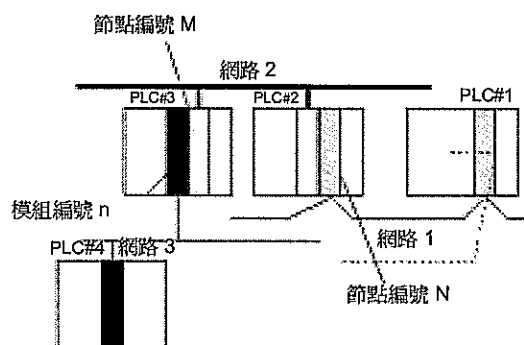
若真正安裝至 PLC (CPU Rack 與擴充 Rack) 的模組型號與位置與登錄 I/O 表中的模組形號不一致，則 I/O 設定錯誤旗標 (A40110) 將會變為 ON。

初始狀態下，CJ 系列 CPU 模組將於電源 ON 時自動建立 I/O 表並依其中的資訊操作。因此，使用者不需要自己建立 I/O 表。

7-19-3 路由表 (Routing Tables)

當在網路間傳送資料時，必需要在每個 CPU 模組中建立一個表來顯示由本地通訊模組的 PLC 到其他網路之通訊路由。這些通訊路由的表格就稱為路由表。

以 CX-P 軟體或 Controller Link 支援軟體來建立路由表，並把這個表傳送到每個 CPU 模組。下圖所示為用來把資料由 PLC #1 傳送至 PLC #4 的路由表。



1,2,3... 1. PLC #1 的中繼網路表：

目的網路	中繼網路	中繼節點
3	1	N

2. PLC #2 的中繼網路表：

目的網路	中繼網路	中繼節點
3	2	M

3. PLC #3 的本地網路表：

本地網路	模組編號
3	n

中繼網路表

本表列出網路位址與要達到目的網路所需連接的第一個中繼節點之節點編號。透過這些中繼節點便可以到達目的網路。

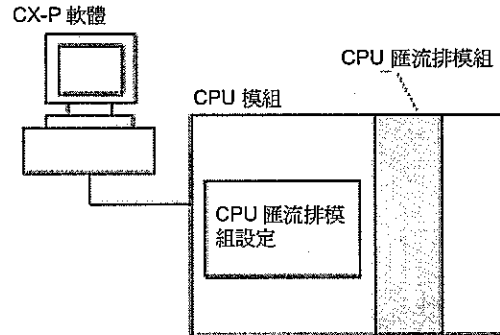
本地網路表

本表列出網路位址與连接到本地 PLC 的通訊模組編號。

這些是受 CPU 模組所控制的 CPU 匯流排模組之設定。真正的設定與所使用的 CPU 匯流排模組之型號有關；詳情請參見模組的操作手冊。

7-19-4 CPU 匯流排模組設定

這些設定不像 I/O 記憶體的资料區域般地直接管理，卻像註記 I/O 表般地由 CX-P 軟體 (CX-P 軟體或程式書寫器) 來設定。有關這些設定的改變之詳情請參見 CX-P 軟體的操作手冊。




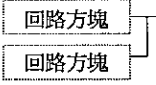
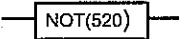
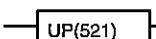
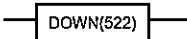
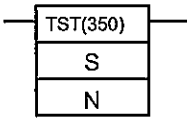
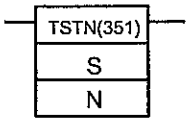
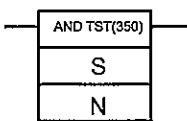
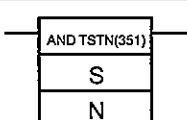
第 8 章 指令功能

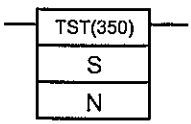
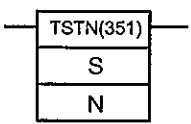
本章列出可以用來撰寫使用者程式的指令。

8-1	程序輸入指令	142
8-2	程序輸出指令	144
8-3	程序控制指令	146
8-4	計時器與計數器指令	150
8-5	比較指令	154
8-6	資料搬移指令	157
8-7	資料移位指令	160
8-8	遞增 / 遞減指令	164
8-9	四則運算指令	165
8-10	轉換指令	170
8-11	邏輯 (Logic) 指令	176
8-12	特殊算術指令	178
8-13	浮點運算指令	179
8-14	雙精度浮點指令點指令 (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	183
8-15	表格資料處理指令	188
8-16	資料控制指令	192
8-17	副程式指令	195
8-18	中斷控制指令	197
8-19	高速計數器與脈衝輸出指令 (僅 CJ1M-CPU22/23)	199
8-20	步進指令	200
8-21	基本 I/O 模組指令	201
8-22	串列通訊指令	202
8-23	網路指令	203
8-24	檔案記憶體指令	205
8-25	顯示指令	206
8-26	時鐘指令	206
8-27	除錯指令	207
8-28	故障診斷指令	208
8-29	其他指令	209
8-30	區塊規劃指令	210
8-31	字串處理指令	216
8-32	工作單控制指令	220



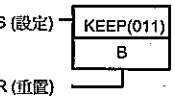
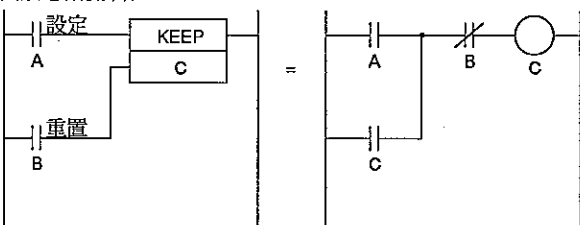
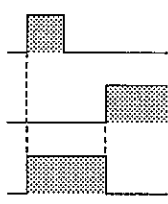
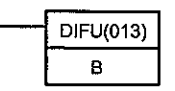
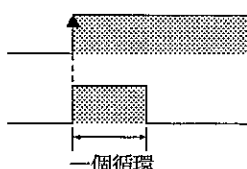
8-1 程序輸入指令

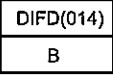
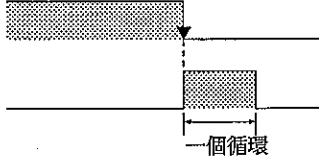
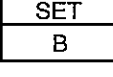
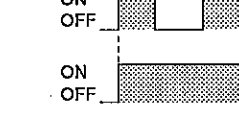
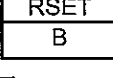
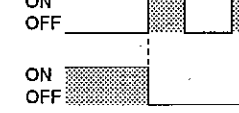
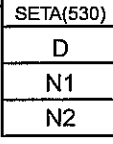
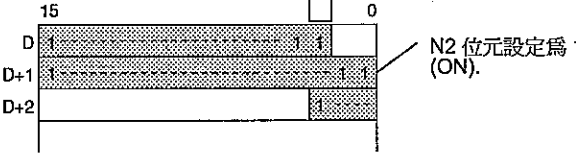
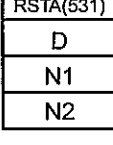
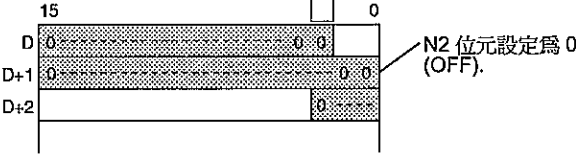
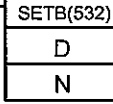

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
LOAD LD @LD %LD !LD !@LD !%LD	<p>母線</p> <p>方塊之 起始點</p>	指出邏輯上的起始點並依據指定的運算元位元之 ON/OFF 狀態建立 ON/OFF 執行條件。	不需要
LOAD NOT LD NOT @LD NOT %LD NOT !LD NOT !@LD NOT !%LD NOT CS1-H, CJ1-H, 或 僅 CJ1M CPU 模 組 : @LD NOT %LD NOT !@LD NOT !%LD NOT	<p>母線</p> <p>方塊之 起始點</p>	指出邏輯上的起始點並依據指定的運算元位元之 ON/OFF 狀態的反相建立 ON/OFF 執行條件。	不需要
AND AND @AND %AND !AND !@AND !%AND		對指定的運算元位元之狀態與現行的執行條件取邏輯 AND 運算。	需要
AND NOT AND NOT @AND NOT %AND NOT !AND NOT !@AND NOT !%AND NOT CS1-H, CJ1-H, 或 僅 CJ1M CPU 模 組 : @AND NOT %AND NOT !@AND NOT !%AND NOT		把指定的運算元位元之狀態反相再與現行的執行條件取邏輯 AND 運算。	需要
OR OR @OR %OR !OR !@OR !%OR	<p>母線</p>	對指定的運算元位元之狀態與現行的執行條件取邏輯 OR 運算。	需要
OR NOT OR NOT @OR NOT %OR NOT !OR NOT !@OR NOT !%OR NOT CS1-H, CJ1-H, 或 僅 CJ1M CPU 模 組 : @OR NOT %OR NOT !@OR NOT !%OR NOT	<p>母線</p>	把指定的運算元位元之狀態反相再與現行的執行條件取邏輯 OR 運算。	需要

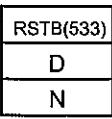
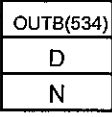
指令	助憶碼	符號 / 運算元	功能	位置 執行條件
AND LOAD	AND LD		<p>對回路方塊間取回路作AND運算。</p> <p>LD 至 } 回路方塊 A</p> <p>LD 至 } 回路方塊 B</p> <p>AND LD 回路方塊 A 與回路方塊 B 之間為串聯。</p>	需要
OR LOAD	OR LD		<p>對回路方塊間取回路OR運算。</p> <p>LD 至 } 回路方塊 A</p> <p>LD 至 } 回路方塊 B</p> <p>OR LD 回路方塊 A 與回路方塊 B 之間為並聯。</p>	需要
NOT	NOT 520		把執行條件反相。	需要
CONDITION ON	UP 521		UP(521) 在執行條件從 OFF 變化為 ON 時，把執行條件變為 ON 一個循環。	需要
CONDITION OFF	DOWN 522		DOWN(522) 在執行條件從 ON 變化為 OFF 時，把執行條件變為 ON 一個循環。	需要
BIT TEST	LD TST 350	 S: 來源words N: 位元編號	LD TST(350), AND TST(350), 與 OR TST(350) 在程式中使用如同 LD, AND, 與 OR；其執行條件為當指定 words 中的指定位元為 ON 時為 ON；而當該位元為 OFF 時為 OFF。	不需要
BIT TEST	LD TSTN 351	 S: 來源words N: 位元編號	LD TSTN(351), 與 TSTN(351), 與 OR TSTN(351) 在程式中使用如同 LD NOT, 與 NOT, 與 OR NOT；其執行條件為當指定 words 中的指定位元為 ON 時為 OFF；而當該位元為 OFF 時為 ON。	不需要
BIT TEST	AND TST 350	 S: 來源words N: 位元編號	LD TST(350), AND TST(350), 與 OR TST(350) 在程式中使用如同 LD, AND, 與 OR；其執行條件為當指定 words 中的指定位元為 ON 時為 ON；而當該位元為 OFF 時為 OFF。	需要
BIT TEST	AND TSTN 351	 S: 來源words N: 位元編號	LD TSTN(351), 與 TSTN(351), 與 OR TSTN(351) 在程式中使用如同 LD NOT, 與 NOT, 與 OR NOT；其執行條件為當指定 words 中的指定位元為 ON 時為 OFF；而當該位元為 OFF 時為 ON。	需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
BIT TEST OR TST 350	 <p>S: 來源 words N: 位元編號</p>	LD TST(350), AND TST(350), 與 OR TST(350) 在程式中使用如同 LD, AND, 與 OR; 其執行條件為當指定 words 中的指定位元為 ON 時為 ON; 而當該位元為 OFF 時為 OFF。	需要
BIT TEST OR TSTN 351	 <p>S: 來源 words N: 位元編號</p>	LD TSTN(351), AND TSTN(351), 與 OR TSTN(351) 在程式中使用如同 LD NOT, 與 NOT, 與 OR NOT; 其執行條件為當指定 words 中的指定位元為 ON 時為 OFF; 而當該位元為 OFF 時為 ON。	需要

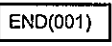
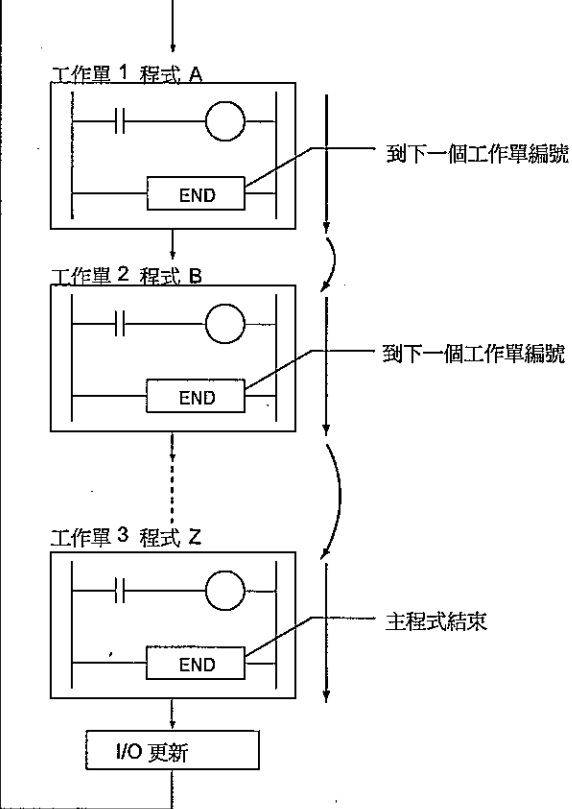
8-2 程序輸出指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
OUTPUT OUT IOUT		把邏輯處理的結果 (執行條件) 輸出至指定的位元。	輸出 需要
OUTPUT NOT OUT NOT IOUT NOT		把邏輯處理的結果 (執行條件) 反相, 並輸出至指定的位元。	輸出 需要
KEEP KEEP !KEEP 011	 <p>S (設定) R (重置) B: 位元</p>	<p>如門鎖電驛般操作。</p>  <p>S 執行條件</p>  <p>R 執行條件</p> <p>B 的狀態</p>	輸出 需要
DIFFERENTIATE UP DIFU IDIFU 013	 <p>B: 位元</p>	<p>DIFU(013) 在執行條件由 OFF 改變為 ON (上升緣) 時把指定的位元變為 ON 一個週期。</p>  <p>執行條件</p> <p>B 的狀態</p> <p>一個循環</p>	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件
DIFFERENTIATE DOWN DIFD IDIFD 014 B: 位元		DIFD(014) 在執行條件由 ON 改變為 OFF (下降緣) 時把指定的位元變為 ON 一個週期。 執行條件 	輸出 需要
SET SET @SET %SET !SET !@SET !%SET B: 位元		SET 在執行條件為 ON 時把運算元位元變為 ON。 SET 的執行條件 	輸出 需要
RESET RSET @RSET %RSET !RSET !@RSET !%RSET B: 位元		RSET 在執行條件為 ON 時把運算元位元變為 OFF。 RSET 的執行條件 	輸出 需要
MULTIPLE BIT SET SETA @SETA 530 D: 開始words N1: 開始位元 N2: 位元編號		SETA(530) 把指定數量的連續位元都設定為 ON。 	輸出 需要
MULTIPLE BIT RESET RSTA @RSTA 531 D: 開始字組 N1: 開始位元 N2: 位元編號		RSTA(531) 把指定數量的連續位元都設定為 OFF。 	輸出 需要
SINGLE BIT SET (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) SETB @SETB !SETB 532 D: words位址 N: 位元編號		SETB(532) 在執行條件為 ON 時把指定 words 中的指定位元變為 ON。 與 SET 指令不同的是, SETB(532) 可用來設定 DM 或 EM words 中的位元。 	輸出 需要

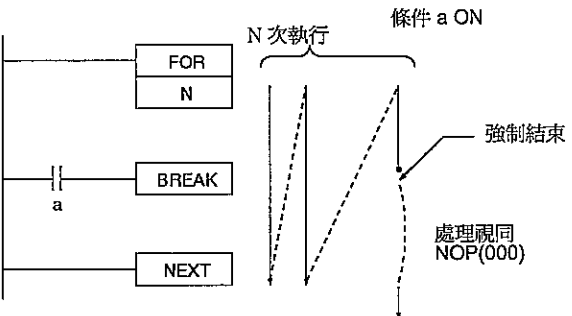
指令 助憶碼	符號 / 運算元	功能	位置 執行條件
SINGLE BIT RESET (僅 CS1-H, CJ1- H, CJ1M, 或 CS1D) RSTB @RSTB !RSTB 533	 D: words位址 N: 位元編號	RSTB(533) 在執行條件為 ON 時把指定 words 中的指定位元變為 OFF。 與 RSET 指令不同的是, RSETB(533) 可用來重置 DM 或 EM words 中的位元。	輸出 需要
SINGLE BIT OUTPUT (僅 CS1-H, CJ1- H, CJ1M, 或 CS1D) OUTB @OUTB !OUTB 534	 D: words位址 N: 位元編號	OUTB(534) 輸出指定位元經邏輯處理的結果 (執行條件)。 與 OUT 指令不同的是, OUTB(534) 可用來控制 DM 或 EM words 中的位元。	輸出 需要

8-3 程序控制指令

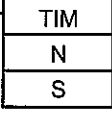
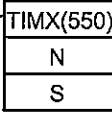
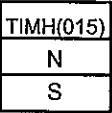
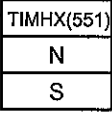
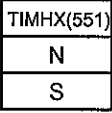
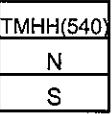
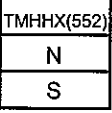
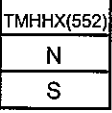
指令 助憶碼	符號 / 運算元	功能	位置 執行條件
END END 001		<p>表示程式的結束。 END(001) 完成程式該循環的執行。位於 END(001)後的指令將不會被執行。執行工作會進入到下一個工作單的程式。當被執行的程式有程式中最大的工作單編號時, END(001) 會標記整個主程式的結束。</p> 	輸出 不需要
NO OPERATION NOP 000		這個指令沒有功能。(NOP(000) 不做處理。)	輸出 不需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
INTERLOCK IL 002	IL(002)	<p>當 IL(002) 的執行條件為 OFF 時，把 IL(002) 與 ILC(003) 間的所有輸出連鎖。IL(002) 與 ILC(003) 通常都是成對使用的。</p>	輸出 需要
INTERLOCK CLEAR ILC 003	ILC(003)	<p>表示輸出連鎖的結束。</p>	輸出 不需要
JUMP JMP 004	JMP(004) N N: 跳躍編號	<p>當 JMP(004) 的執行條件為 OFF 時，程式會直接跳到程式中第一個具有相同跳躍編號的 JME(005) 繼續執行。JMP(004) 與 JME(005) 需成對使用。</p>	輸出 需要
JUMP END JME 005	JME(005) N N: 跳躍編號	<p>表示 JMP 或 CTP 指令的結束。</p>	輸出 不需要
CONDITIONAL JUMP CJP 510	CJP(510) N N: 跳躍編號	<p>CJP(510) 的操作與 JMP(004) 恰恰相反。當 CJP(510) 的執行條件為 ON 時，程式會直接跳到程式中第一個具有相同跳躍編號的 JME(005) 繼續執行。CJP(510) 與 JME(005) 需成對使用。</p>	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件
助憶碼 CONDITIONAL JUMP CJPN 511	<div style="border: 1px solid black; padding: 2px; display: inline-block;">CJPN(511)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">N</div> N: 跳躍編號	CJPN(511) 的操作與 JMP(004) 幾乎相同。當 CJPN(511) 的執行條件為 OFF 時，程式會直接跳到程式中第一個具有相同跳躍編號的 JME(005) 繼續執行。CJPN(511) 與 JME(005) 需成對使用。 	輸出 不需要
MULTIPLE JUMP JMP0 515	<div style="border: 1px solid black; padding: 2px; display: inline-block;">JMP0(515)</div>	當 JMP0(515) 的執行條件為 OFF 時，JMP0(515) 至下一個 JME0(516) 之間的所有指令都視同 NOP(000) 般處理。JMP0(515) 與 JME0(516) 為成對使用。這組指令在程式中的使用次數沒有限制。 	輸出 需要
MULTIPLE JUMP END JME0 516	<div style="border: 1px solid black; padding: 2px; display: inline-block;">JME0(516)</div>	表示 JMP 0 指令的結束。	輸出 不需要
FOR-NEXT LOOPS FOR 512	<div style="border: 1px solid black; padding: 2px; display: inline-block;">FOR(512)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">N</div> N: 迴圈次數	FOR(512) 與 NEXT(513) 之間的指令會重複執行指定的次數。FOR(512) 與 NEXT(513) 需成對使用。 	輸出 不需要

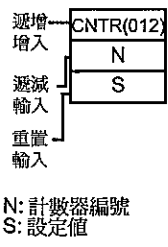
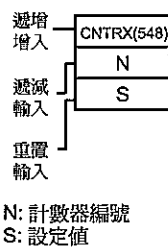
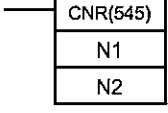
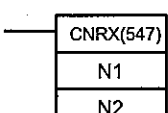
指令 助億碼	符號 / 運算元	功能	位置 執行條件
BREAK LOOP BREAK 514	BREAK(514)	<p>用於 FOR-NEXT 迴圈的程式中在特定條件滿足時取消迴圈的執行。迴圈中剩下的指令會被當做 NOP(000) 指令來處理。</p> 	輸出 需要
FOR-NEXT LOOPS NEXT 513	NEXT(513)	表示 FOR- NEXT 指令的結束。	輸出 不需要

8-4 計時器與計數器指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
TIMER TIM (BCD)		TIM 為以 0.1 s 為單位的遞減計時器。BCD 方式時:0~999.9 秒 設定值 (SV) 的範圍為 0 至 999.9 s。BIN 方式時:0~6553.3 秒 (BCD 或 Hex)	輸出 需要
	N: 計時器編號 S: 設定值		
(僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)		計時器輸入在 completion flag 變為 ON 之前就變為 OFF。	
	N: 計時器編號 S: 設定值		
HIGH-SPEED TIMER TIMH 015 (BCD)		TIMH(015) 為以 10 ms 為單位的遞減計時器。 設定值 (SV) 的範圍為 0 至 99.99 s。BCD 方式時:0~99 秒 BIN 方式時:0~655.35 秒 (BCD 或 Hex)	輸出 需要
	N: 計時器編號 S: 設定值		
TIMHX 551 (BIN)		計時器輸入在 completion flag 變為 ON 之前就變為 OFF。	
	N: 計時器編號 S: 設定值		
(僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)		計時器輸入在 completion flag 變為 ON 之前就變為 OFF。	
	N: 計時器編號 S: 設定值		
ONE-MS TIMER TMHH 540 (BCD)		TMHH(540) 為以 1 ms 為單位的遞增計時器。設定值(SV) 的範圍為 0 至 9.999 s。 TMHH(540) 的時序圖與 TIMH(015) 的相同。 BCD 方式時:0~9.999 秒 BIN 方式時:0~65.535 秒 (BCD 或 Hex)	輸出 需要
	N: 計時器編號 S: 設定值		
TMHHX 552 (BCD)			
	N: 計時器編號 S: 設定值		
(僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)			
	N: 計時器編號 S: 設定值		

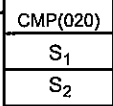
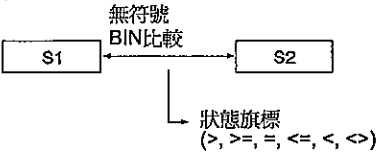
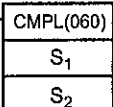
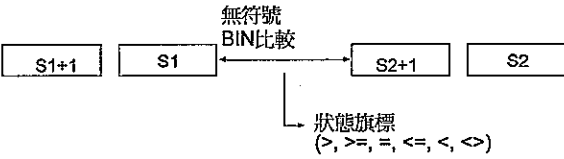
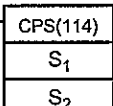
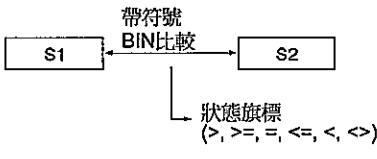
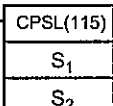
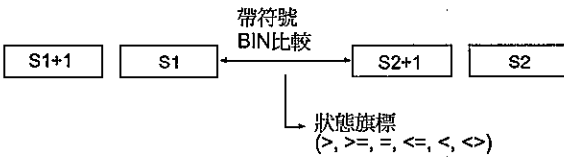
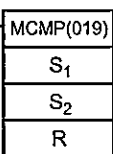
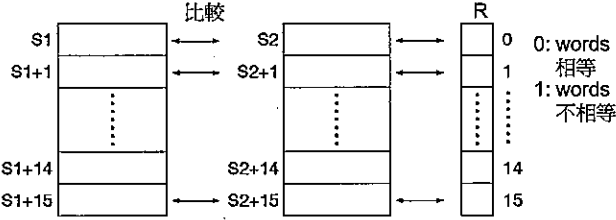
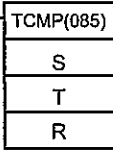
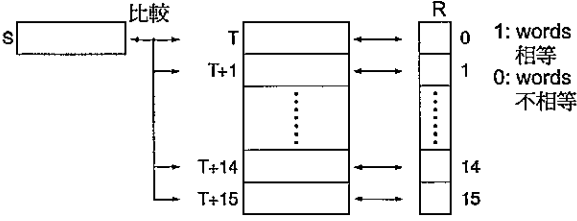
指令	符號 / 運算元	功能	位置 執行條件
ACCUMULATIVE TIMER TTIM 087 (BCD) 計時器 輸入 重置 輸入 N: 計時器編號 S: 設定值 TTIMX 555 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) 計時器 輸入 重置 輸入 N: 計時器編號 S: 設定值		TTIM(087) 為以 0.1 s 為單位的遞增計時器。 設定值 (SV) 的範圍為 0 至 999.9 s。 BCD 方式時: 0~999.9 秒 BIN 方式時: 0~6553.3 秒 (BCD 或 Hex) 	輸出 需要
LONG TIMER TIML 542 (BCD) 計時器 輸入 重置 輸入 D1: 完成旗標 D2: PV words S: SV words TIMLX 553 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) 計時器 輸入 重置 輸入 D1: 完成旗標 D2: PV words S: SV words		TIML(542) 為以 100 ms 為單位的遞減計時器， 計時範圍達 9999999.9 s (約 115 天)。 	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件
MULTI-OUTPUT TIMER MTIM 543 (BCD) MTIMX 554 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> MTIM(543) D1 D2 S </div> D1: 完成旗標 D2: PV words S: 第一個 SV words <div style="border: 1px solid black; padding: 5px;"> MTIMX(554) D1 D2 S </div> D1: 完成旗標 D2: PV words S: 第一個 SV words	<p>MTIM(543) 為以 0.1 s 為單位且具有八組獨立的 SV 與完成期標之遞減計時器。BCD 方式時:0~999.9 秒 設定值 (SV) 的範圍為 0 至 999.9 s。BIN 方式時:0~6553.5 秒 (BCD 或 Hex)</p> <p>計時器輸入 ON OFF</p> <p>計時器 PV (D2) SV 7 至 SV 2 SV 1 SV 0 0</p> <p>完成旗標 (D1) 位元 7 至 位元 2 位元 1 位元 0</p>	輸出需要
COUNTER CNT (BCD) CNTX 546 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> 計數器輸入 CNT N S 重置輸入 </div> N: 計數器編號 S: 設定值 <div style="border: 1px solid black; padding: 5px;"> 計數器輸入 CNTX(546) N S 重置輸入 </div> N: 計數器編號 S: 設定值	<p>CNT 作為一遞減計數器使用。設定值 (SV) 的範圍為 0 至 9,999。</p> <p>計數輸入 ON OFF</p> <p>重置輸入 ON OFF</p> <p>計數器 PV SV 0</p> <p>完成旗標 ON OFF</p> <p>BCD 方式時:0~9999 次 BIN 方式時:0~65535 次 (BCD 或 Hex)</p>	輸出需要

指令	符號 / 運算元	功能	位置 執行條件
REVERSIBLE COUNTER 助憶碼 CNTR 012 (BCD) 遞增 輸入 遞減 輸入 重置 輸入 N: 計數器編號 S: 設定值 CNTRX 548 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) 遞增 輸入 遞減 輸入 重置 輸入 N: 計數器編號 S: 設定值	 	CNTR(012) 為一可加減的計數器。 遞增輸入 遞減輸入 計數器 PV 0 • 加算執行時： 計數器 PV SV 0 +1 完成旗標 ON OFF • 減算執行時： 計數器 PV SV 0 -1 完成旗標 ON OFF	輸出 需要
RESET TIMER/ COUNTER CNR @CNR 545 (BCD) N ₁ : 範圍內第一個 編號 N ₂ : 範圍內最後一 個編號 CNRX @CNRX 547 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) N ₁ : 範圍內第一個 編號 N ₂ : 範圍內最後一 個編號	 	在指定的計時器範圍或計數器數字範圍內重置計時器或計數器。設定值 (SV) 最大可以設定到 9999 (BCD) , FFFF (BIN) 。	輸出 需要

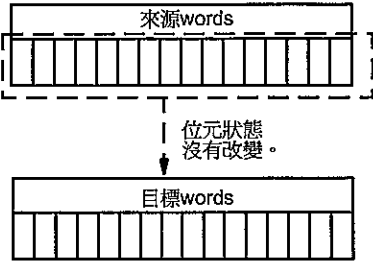
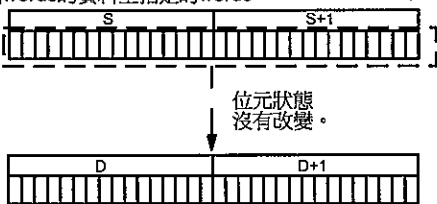
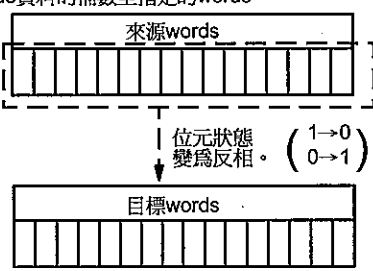
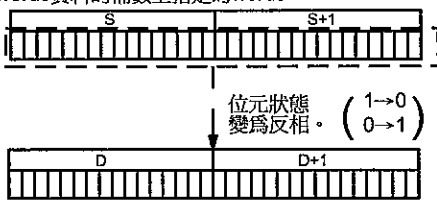
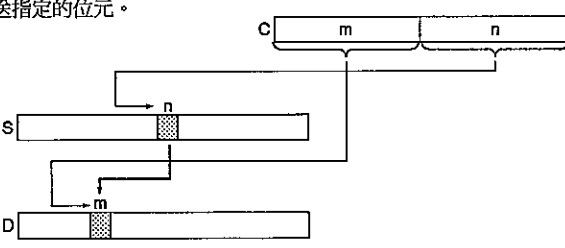
8-5 比較指令

指令	助憶碼	符號 / 運算元	功能	位置 執行條件
符號比較 (無符號)	LD, AND, OR +=, <>, <, <=, >, >=	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">符號與選項</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">S₁</div> <div style="border: 1px solid black; padding: 2px;">S₂</div> <p>S₁: 比較資料 1 S₂: 比較資料 2</p>	<p>符號比較指令 (無符號) 比較兩個 16 位元 (Hex 的四位數) 無符號 BIN 資料的值 (常數及/或指定字組的內容), 當比較條件為真時並建立一個 ON 的執行條件。有三種符號比較指令: LD (LOAD)、AND、與 OR。</p> <p>LD: 當比較條件成立時, 執行條件為 ON。</p> <p>AND: 當比較條件成立時, 執行條件為 ON。</p> <p>OR: 當比較條件成立時, 執行條件為 ON。</p>	LD: 不需要 AND, OR: 需要
符號比較 (雙 words, 無符號)	LD, AND, 或 +=, <>, <, <=, >, >= + L	S ₁ : 比較資料 1 S ₂ : 比較資料 2	符號比較指令 (雙 words, 無符號) 比較兩個 32 位元無符號 BIN 資料的值 (常數及/或指定雙 words 資料的內容), 當比較條件成立時並建立一個 ON 的執行條件。有三種符號比較指令: LD (LOAD)、AND、與 OR。	LD: 不需要 AND, OR: 需要
符號比較 (帶符號)	LD, AND, 或 +=, <>, <, <=, >, >= + S	S ₁ : 比較資料 1 S ₂ : 比較資料 2	符號比較指令 (帶符號) 比較兩個 16 位元 (Hex 的四位數) 帶符號 BIN 資料的值 (常數及/或指定 words 的內容), 當比較條件成立時並建立一個 ON 的執行條件。有三種符號比較指令: LD (LOAD)、AND、與 OR。	LD: 不需要 AND, OR: 需要
符號比較 (雙 words, 帶符號)	LD, AND, 或 +=, <>, <, <=, >, >= + SL	S ₁ : 比較資料 1 S ₂ : 比較資料 2	符號比較指令 (雙 words, 帶符號) 比較兩個 32 位元 (Hex 的八位數) 帶符號 BIN 資料的值 (常數及/或指定雙 words 資料的內容), 當比較條件成立時並建立一個 ON 的執行條件。有三種符號比較指令: LD (LOAD)、AND、與 OR。	LD: 不需要 AND, OR: 需要

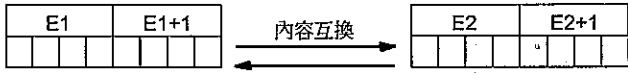
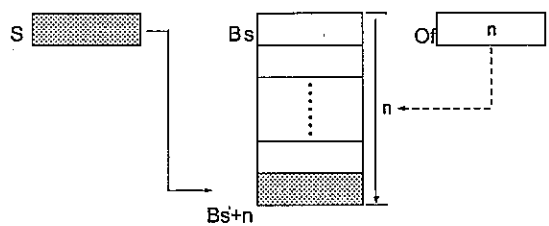
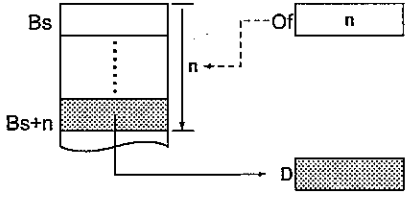
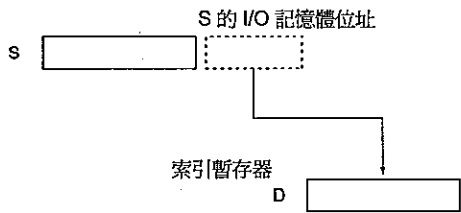
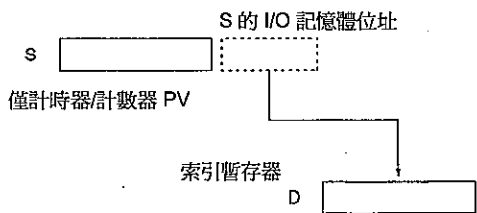
指令 助億碼	符號 / 運算元	功能	位置 執行條件
UNSIGNED COMPARE CMP !CMP 020	 <p>S₁: 比較資料 1 S₂: 比較資料 2</p>	<p>比較兩個無符號的BIN值 (常數及/或指定words的內容) 並把結果輸出到輔助區域的狀態旗標。</p> 	輸出 需要
DOUBLE UNSIGNED COMPARE CMPL 060	 <p>S₁: 比較資料 1 S₂: 比較資料 2</p>	<p>比較兩個雙精度無符號的BIN值 (常數及/或指定words的內容) 並把結果輸出到輔助區域的狀態旗標。</p> 	輸出 需要
SIGNED BINARY COMPARE CPS !CPS 114	 <p>S₁: 比較資料 1 S₂: 比較資料 2</p>	<p>比較兩個帶符號的BIN值 (常數及/或指定words的內容) 並把結果輸出到輔助區域的狀態旗標。</p> 	輸出 需要
DOUBLE SIGNED BINARY COMPARE CPSL 115	 <p>S₁: 比較資料 1 S₂: 比較資料 2</p>	<p>比較兩個雙精度帶符號的BIN值 (常數及/或指定words的內容) 並把結果輸出到輔助區域的狀態旗標。</p> 	輸出 需要
MULTIPLE COMPARE MCMP @MCMP 019	 <p>S₁: 集合 1 的第一個words S₂: 集合 2 的第一個words R: 結果words</p>	<p>把連續的 16 個words與另外 16 個連續的words相比較，並把輸出words中兩個比較words內容不相等的位元變為 ON。</p> 	輸出 需要
TABLE COMPARE TCMP @TCMP 085	 <p>S: 來源words T: 表中的第一個words R: 結果words</p>	<p>把來源資料與另外 16 個連續位元的words相比較，並把輸出words中比較字組與來源words內容相等的位元變為 ON。</p> 	輸出 需要

指令	助憶碼	符號 / 運算元	功能	位置 執行條件
UNSIGNED BLOCK COMPARE	BCMP @BCMP 068	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BCMP(068) <hr/> S <hr/> T <hr/> R </div> <p>S: 來源words T: 表中的第一個 words R: 結果words</p>	<p>把來源資料與另外 16 個範圍 (以 16 個下限與 16 個上限來定義) 的 words 相比較, 並把輸出 words 中來源 words 所落在的範圍之相對應位置的位元變為 ON。</p> <p style="text-align: right;">1: 在範圍內 0: 不在範圍內</p>	輸出 需要
EXPANDED BLOCK COMPARE	BCMP2 @BCMP2 502 (僅 CJ1M)	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> BCMP2(502) <hr/> S <hr/> T <hr/> R </div> <p>S: 來源words T: 區段中的第一個 words R: 結果words</p>	<p>把來源資料與最多 256 個範圍 (以上限與下限來定義) 之資料相比較, 並把結果 words 中來源資料所落範圍之對應位元變為 ON。</p> <p style="text-align: right;">1: 在範圍內 0: 不在範圍內</p> <p style="text-align: right;">D 位元</p> <p style="text-align: right;">最多 D+15</p> <p>附註: A 可以小於等於 B 或大於 B。</p>	輸出 需要 127
AREA RANGE COMPARE (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	ZCP @ZCP 088	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> ZCP(088) <hr/> CD <hr/> LL <hr/> UL </div> <p>CD: Compare data (1 word) LL: 下限值 UL: 上限值</p>	<p>把位於 CD (words 內容或常數) 中的 16 位元無符號 BIN 數值與定義於 LL 與 UL 間的範圍作比較, 並把結果輸出到輔助區域中的狀態旗標。</p>	輸出 需要
DOUBLE AREA RANGE COMPARE (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	ZCPL @ZCPL 116	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> ZCPL(116) <hr/> CD <hr/> LL <hr/> UL </div> <p>CD: Compare data (2 words) LL: 下限值 UL: 上限值</p>	<p>把位於 CD 與 CD+1 (words 內容或常數) 中的 32 位元無符號 BIN 數值與定義於 LL 與 UL 間的範圍作比較, 並把結果輸出到輔助區域中的狀態旗標。</p>	輸出 需要

8-6 資料搬移指令

指令	助憶碼	符號 / 運算元	功能	位置 執行條件
MOVE	MOV @MOV !MOV !@MOV 021	MOV(021) S D	傳送一個words的資料至指定的words。 	輸出 需要
DOUBLE MOVE	MOVL @MOVL 498	MOVL(498) S D	傳送兩個words的資料至指定的words。 	輸出 需要
MOVE NOT	MVN @MVN 022	MVN(022) S D	傳送一個words資料的補數至指定的words。 	輸出 需要
DOUBLE MOVE NOT	MVNL @MVNL 499	MVNL(499) S D	傳送兩個words資料的補數至指定的words。 	輸出 需要
MOVE BIT	MOVB @MOVB 082	MOVB(082) S C D	傳送指定的位元。 	輸出 需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件				
MOVE DIGIT MOVD @MOVD 083	<table border="1"> <tr><td>MOVD(083)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: 來源words 或資料 C: 控制words D: 目標words</p>	MOVD(083)	S	C	D	<p>傳送一個或多個指定的位數 (每位數為 4 個位元)。</p>	輸出 需要
MOVD(083)							
S							
C							
D							
MULTIPLE BIT TRANSFER XFRB @XFRB 062	<table border="1"> <tr><td>XFRB(062)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制words S: 第一個來源 words D: 第一個目標 words</p>	XFRB(062)	C	S	D	<p>傳送指定數量的連續位元。</p>	輸出 需要
XFRB(062)							
C							
S							
D							
BLOCK TRANSFER XFER @XFER 070	<table border="1"> <tr><td>XFER(070)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>N: words數 S: 第一個來源 words D: 第一個目標 words</p>	XFER(070)	N	S	D	<p>傳送指定數量的連續words。</p>	輸出 需要
XFER(070)							
N							
S							
D							
BLOCK SET BSET @BSET 071	<table border="1"> <tr><td>BSET(071)</td></tr> <tr><td>S</td></tr> <tr><td>St</td></tr> <tr><td>E</td></tr> </table> <p>S: 來源words St: 開始words E: 結束words</p>	BSET(071)	S	St	E	<p>把一個words複製到連續範圍的words內。</p>	輸出 需要
BSET(071)							
S							
St							
E							
DATA EXCHANGE XCHG @XCHG 073	<table border="1"> <tr><td>XCHG(073)</td></tr> <tr><td>E1</td></tr> <tr><td>E2</td></tr> </table> <p>E1: 第一個交換 words E2: 第二個交換 words</p>	XCHG(073)	E1	E2	<p>把兩個指定words的內容互換。</p>	輸出 需要	
XCHG(073)							
E1							
E2							

指令	符號 / 運算元	功能	位置 執行條件
DOUBLE DATA EXCHANGE XCGL @XCGL 562	XCGL(562) E1 E2 E1: 第一個交換 words E2: 第二個交換 words	把連續兩個 words 的內容與另外連續兩個 words 的內容互換。 	輸出 需要
SINGLE WORD DISTRIBUTE DIST @DIST 080	DIST(080) S Bs Of S: 來源 words Bs: 目標基底位址 Of: 偏量	把來源 words 傳送到目標 words，目標 words 的位置為基底位址加上一個偏量值。 	輸出 需要
DATA COLLECT COLL @COLL 081	COLL(081) Bs Of D Bs: 來源基底位址 Of: 偏量 D: 目標 words	把來源 words 傳送到目標 words，來源 words 的位置為基底位址加上一個偏量值。 	輸出 需要
MOVE TO REGISTER MOVR @MOVR 560	MOVR(560) S D S: 來源 (想要的 words 或位元) D: 目標 (索引暫存器)	設定指定的索引暫存器中指定 words、位元、或計時器/計數器完成旗標之 PLC 記憶體位址。(使用 MOVRW(561)來設定索引暫存器中計時器/計數器的 PV 之記憶體位址。 	輸出 需要
MOVE TIMER/COUNTER PV TO REGISTER MOVRW @MOVRW 561	MOVRW(561) S D S: 來源 (想要 TC 編號) D: 目標 (索引暫存器)	設定指定的索引暫存器中指定計時器或計數器的 PV 之 PC 記憶體位址。(使用 MOVR(560)來設定索引暫存器中 words、位元、或計時器/計數器完成旗標的 PLC 記憶體位址。) 	輸出 需要

8-7 資料移位指令

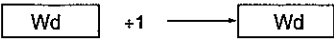
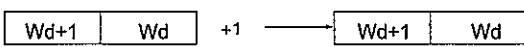
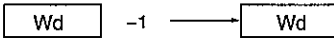
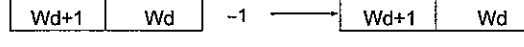
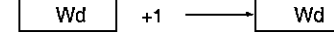
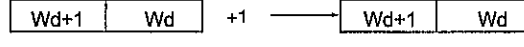
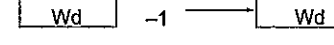
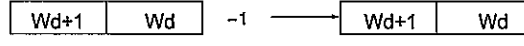
指令	符號 / 運算元	功能	位置 執行條件
SHIFT REGISTER SFT 010	<p>資料輸入 移位輸入 重置輸入</p> <p>St: 開始words E: 結束words</p>	操作一移位暫存器。 	輸出 需要
REVERSIBLE SHIFT REGISTER SFTR @SFTR 084	<p>C: 控制words St: 開始words E: 結束words</p>	建立一個會把資料往右或往左移位的移位暫存器。 	輸出 需要
ASYNCHRO- NOUS SHIFT REGISTER ASFT @ASFT 017	<p>C: 控制words St: 開始words E: 結束words</p>	把所有在指定words範圍中的的非零words資料往 St 或往 E 方向 移位，取代 0000 Hex的words值。 	輸出 需要
WORD SHIFT WSFT @WSFT 016	<p>S: 來源words St: 開始words E: 結束words</p>	把words單元 St 與 E 之間的資料移位。 	輸出 需要
ARITHMETIC SHIFT LEFT ASL @ASL 025	<p>Wd: words</p>	把 Wd 的內容向左移位一個位元。 	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件
DOUBLE SHIFT LEFT ASLL @ASLL 570	ASLL(570) Wd	把 Wd 與 Wd + 1 的內容向左移位一個位元。 	輸出 需要
ARITHMETIC SHIFT RIGHT ASR @ASR 026	ASR(026) Wd	把 Wd 的內容向右移位一個位元。 	輸出 需要
DOUBLE SHIFT RIGHT ASRL @ASRL 571	ASRL(571) Wd	把 Wd 與 Wd + 1 的內容向右移位一個位元。 	輸出 需要
ROTATE LEFT ROL @ROL 027	ROL(027) Wd	把 Wd 所有的位元含進位旗標 (CY) 共同向左移位一個位元。 	輸出 需要
DOUBLE ROTATE LEFT ROLL @ROLL 572	ROLL(572) Wd	把 Wd 與 Wd + 1 所有的位元含進位旗標 (CY) 共同向左移位一個位元。 	輸出 需要
ROTATE LEFT WITHOUT CARRY RLNC @RLNC 574	RLNC(574) Wd	把 Wd 所有的位元不含進位旗標 (CY) 共同向左移位一個位元。 	輸出 需要
DOUBLE ROTATE LEFT WITHOUT CARRY RLNL @RLNL 576	RLNL(576) Wd	把 Wd 與 Wd + 1 所有的位元不含進位旗標 (CY) 共同向左移位一個位元。 	輸出 需要
ROTATE RIGHT ROR @ROR 028	ROR(028) Wd	把 Wd 與 Wd + 1 所有的位元含進位旗標 (CY) 共同向右移位一個位元。 	輸出 需要
DOUBLE ROTATE RIGHT RORL @RORL 573	RORL(573) Wd	把 Wd 與 Wd + 1 所有的位元含進位旗標 (CY) 共同向右移位一個位元。 	輸出 需要

指令	助憶碼	符號 / 運算元	功能	位置 執行條件
ROTATE RIGHT WITHOUT CARRY	RRNC @RRNC 575	RRNC(575) Wd	把 Wd 所有的位元含進位旗標 (CY) 都向右移位一個位元。 Wd 最右邊的位元內容會移到最左邊的位元與進位旗標 (CY)。	輸出 需要
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL @RRNL 577	RRNL(577) Wd	把 Wd 與 Wd + 1 所有的位元含進位旗標 (CY) 都向右移位一個位元。 Wd + 1 最右邊的位元內容會移到 Wd 最左邊的位元與進位旗標 (CY)。	輸出 需要
ONE DIGIT SHIFT LEFT	SLD @SLD 074	SLD(074) St E	把資料往左移位一個位數 (4 位元)。 遺失	輸出 需要
ONE DIGIT SHIFT RIGHT	SRD @SRD 075	SRD(075) St E	把資料往右移位一個位數 (4 位元)。 遺失	輸出 需要
SHIFT N-BIT DATA LEFT	NSFL @NSFL 578	NSFL(578) D C N	往左移位指定的位元數。 	輸出 需要
SHIFT N-BIT DATA RIGHT	NSFR @NSFR 579	NSFR(579) D C N	往右移位指定的位元數。 	輸出 需要

指令	助憶碼	符號 / 運算元	功能	位置 執行條件			
SHIFT N-BITS LEFT	NASL @NASL 580	<table border="1"> <tr><td>NASL(580)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: 移位words C: 控制words</p>	NASL(580)	D	C	<p>把指定words資料的 16 個位元往左移位指定的位元數。</p>	輸出 需要
NASL(580)							
D							
C							
DOUBLE SHIFT N-BITS LEFT	NSLL @NSLL 582	<table border="1"> <tr><td>NSLL(582)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: 移位words C: 控制words</p>	NSLL(582)	D	C	<p>把指定words資料的 32 個位元往左移位指定的位元數。</p>	輸出 需要
NSLL(582)							
D							
C							
SHIFT N-BITS RIGHT	NASR @NASR 581	<table border="1"> <tr><td>NASR(581)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: 移位words C: 控制words</p>	NASR(581)	D	C	<p>把指定words資料的 16 個位元往右移位指定的位元數。</p>	輸出 需要
NASR(581)							
D							
C							
DOUBLE SHIFT N-BITS RIGHT	NSRL @NSRL 583	<table border="1"> <tr><td>NSRL(583)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>D: 移位words C: 移位words</p>	NSRL(583)	D	C	<p>把指定words資料的 32 個位元往右移位指定的位元數。</p>	輸出 需要
NSRL(583)							
D							
C							

8-8 遞增 / 遞減指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
INCREMENT BINARY ++ @++ 590	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> ++(590) Wd </div> Wd: words	把指定words的 4 位元Hex數之內容加 1。 	輸出 需要
DOUBLE INCREMENT BINARY ++L @++L 591	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> ++L(591) Wd </div> Wd: words	把指定words的 8 位元Hex位數之內容加 1。 	輸出 需要
DECREMENT BINARY -- @-- 592	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> --(592) Wd </div> Wd: words	把指定words的 4 位元Hex數之內容減 1。 	輸出 需要
DOUBLE DECREMENT BINARY --L @--L 593	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> --L(593) Wd </div> Wd: 第一個words	把指定words的 8 位元Hex數之內容減 1。 	輸出 需要
INCREMENT BCD ++B @++B 594	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> ++B(594) Wd </div> Wd: words	把指定words的 4 位元 BCD 之內容加 1。 	輸出 需要
DOUBLE INCREMENT BCD ++BL @++BL 595	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> ++BL(595) Wd </div> Wd: 第一個words	把指定words的 8 位元 BCD 之內容加 1。 	輸出 需要
DECREMENT BCD --B @--B 596	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> --B(596) Wd </div> Wd: words	把指定words的 4 位元 BCD 之內容減 1。 	輸出 需要
DOUBLE DECREMENT BCD --BL @--BL 597	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> --BL(597) Wd </div> Wd: 第一個words	把指定words的 8 位元 BCD 之內容減 1。 	輸出 需要

8-9 四則運算指令

指令	符號 / 運算元	功能	位置 執行條件
SIGNED BINARY ADD WITHOUT CARRY 助億碼 + @+ 400	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +(400) <hr/> Au <hr/> Ad <hr/> R </div> Au: 被加數words Ad: 加數words R: 結果words	把 4 位數 (單words) 的Hex資料及/或常數相加。 <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au</div> (帶符號BIN) + <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad</div> (帶符號BIN) <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R</div> (帶符號BIN) </div> 當產生進位時 CY 將會變為 ON。	輸出 需要
DOUBLE SIGNED BINARY ADD WITHOUT CARRY +L @+L 401	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +L(401) <hr/> Au <hr/> Ad <hr/> R </div> Au: 被加數第一個數words Ad: 加數第一個words R: 結果第一個words	把 8 位數 (雙words) 的Hex資料及/或常數相加。 <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au</div> (帶符號BIN) + <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad</div> (帶符號BIN) <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R</div> (帶符號BIN) </div> 當產生進位時 CY 將會變為 ON。	輸出 需要
SIGNED BINARY ADD WITH CARRY +C @+C 402	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +C(402) <hr/> Au <hr/> Ad <hr/> R </div> Au: 被加數words Ad: 加數words R: 結果words	把 4 位數 (單words) 的Hex資料及/或常數與進位旗標 (CY) 一同相加。 <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au</div> (帶符號BIN) + <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad</div> (帶符號BIN) <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R</div> (帶符號BIN) </div> 當產生進位時 CY 將會變為 ON。	輸出 需要
DOUBLE SIGNED BINARY ADD WITH CARRY +CL @+CL 403	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +CL(403) <hr/> Au <hr/> Ad <hr/> R </div> Au: 被加數第一個words Ad: 加數第一個words R: 結果第一個words	把 8 位數 (雙words) 的Hex資料及/或常數與進位旗標 (CY) 一同相加。 <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au</div> (帶符號BIN) + <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad</div> (帶符號BIN) <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R+1</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R</div> (帶符號BIN) </div> 當產生進位時 CY 將會變為 ON。	輸出 需要
BCD ADD WITHOUT CARRY +B @+B 404	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> +B(404) <hr/> Au <hr/> Ad <hr/> R </div> Au: 被加數words Ad: 加數words R: 結果words	把 4 位數 (雙words) 的BCD資料及/或常數相加。 <div style="text-align: center;"> <div style="border: 1px solid black; padding: 2px; display: inline-block;">Au</div> (BCD) + <div style="border: 1px solid black; padding: 2px; display: inline-block;">Ad</div> (BCD) <hr style="width: 50%; margin: 0 auto;"/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">CY</div> <div style="border: 1px solid black; padding: 2px; display: inline-block;">R</div> (BCD) </div> 當產生進位時 CY 將會變為 ON。	輸出 需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
DOUBLE BCD ADD WITHOUT CARRY +BL @+BL 405	+BL(405) Au Ad R	把 8 位數 (雙words) 的BCD資料及/或常數與進位旗標 (CY) 一同相加。 $\begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (BCD) \\ + \quad \boxed{Ad+1} \quad \boxed{Ad} \quad (BCD) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (BCD) \end{array}$ 當產生進位 時 CY 將會 變為 ON。	輸出 需要
BCD ADD WITH CARRY +BC @+BC 406	+BC(406) Au Ad R	把 4 位數 (單words) 的BCD資料及/或常數與進位旗標 (CY) 一同相加。 $\begin{array}{r} \boxed{Au} \quad (BCD) \\ + \quad \boxed{Ad} \quad (BCD) \\ + \quad \boxed{CY} \\ \hline \boxed{CY} \quad \boxed{R} \quad (BCD) \end{array}$ 當產生進位 時 CY 將會 變為 ON。	輸出 需要
DOUBLE BCD ADD WITH CARRY +BCL @+BCL 407	+BCL(407) Au Ad R	把 8 位數 (雙words) 的BCD資料及/或常數與進位旗標 (CY) 一同相加。 $\begin{array}{r} \boxed{Au+1} \quad \boxed{Au} \quad (BCD) \\ + \quad \boxed{Ad+1} \quad \boxed{Ad} \quad (BCD) \\ + \quad \boxed{CY} \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (BCD) \end{array}$ 當產生進位 時 CY 將會 變為 ON。	輸出 需要
SIGNED BINARY SUBTRACT WITHOUT CARRY - @- 410	-(410) Mi Su R	把 4 位數 (單words) 的Hex資料及/或常數相減。 $\begin{array}{r} \boxed{Mi} \quad (帶符號BIN) \\ - \quad \boxed{Su} \quad (帶符號BIN) \\ \hline \boxed{CY} \quad \boxed{R} \quad (帶符號BIN) \end{array}$ 當產生借位 時 CY 將會 變為 ON。	輸出 需要
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY -L @-L 411	-L(411) Mi Su R	把 8 位數 (雙words) 的Hex資料及/或常數相減。 $\begin{array}{r} \boxed{Mi+1} \quad \boxed{Mi} \quad (帶符號BIN) \\ - \quad \boxed{Su+1} \quad \boxed{Su} \quad (帶符號BIN) \\ \hline \boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (帶符號BIN) \end{array}$ 當產生借位 時 CY 將會 變為 ON。	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件
SIGNED BINARY SUBTRACT WITH CARRY -C @-C 412	$\begin{array}{ c } \hline -C(412) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ Mi: 被減數words Su: 減數words R: 結果words	把 4 位數 (單words) 的Hex資料及/或常數相減再減去進位旗標 (CY)。 $\begin{array}{r} \boxed{Mi} \text{ (帶符號BIN)} \\ \boxed{Su} \text{ (帶符號BIN)} \\ - \\ \hline \boxed{CY} \\ \boxed{CY} \ \boxed{R} \text{ (帶符號BIN)} \end{array}$ 當產生借位時 CY 將會變為 ON。	輸出需要
DOUBLE SIGNED BINARY WITH CARRY -CL @-CL 413	$\begin{array}{ c } \hline -CL(413) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ Mi: 被減數words Su: 減數words R: 結果words	把 8 位數 (雙words) 的Hex資料及/或常數相減再減去進位旗標 (CY)。 $\begin{array}{r} \boxed{Mi+1} \ \boxed{Mi} \text{ (帶符號BIN)} \\ \boxed{Su+1} \ \boxed{Su} \text{ (帶符號BIN)} \\ - \\ \hline \boxed{CY} \\ \boxed{CY} \ \boxed{R+1} \ \boxed{R} \text{ (帶符號BIN)} \end{array}$ 當產生借位時 CY 將會變為 ON。	輸出需要
BCD SUBTRACT WITHOUT CARRY -B @-B 414	$\begin{array}{ c } \hline -B(414) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ Mi: 被減數words Su: 減數words R: 結果words	把 4 位數 (單words) 的BCD資料及/或常數相減。 $\begin{array}{r} \boxed{Mi} \text{ (BCD)} \\ \boxed{Su} \text{ (BCD)} \\ - \\ \hline \boxed{CY} \ \boxed{R} \text{ (BCD)} \end{array}$ 當產生借位時 CY 將會變為 ON。	輸出需要
DOUBLE BCD SUBTRACT WITHOUT CARRY -BL @-BL 415	$\begin{array}{ c } \hline -BL(415) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ Mi: 被減數第一個words Su: 減數第一個words R: 結果第一個words	把 8 位數 (雙words) 的BCD資料及/或常數相減。 $\begin{array}{r} \boxed{Mi+1} \ \boxed{Mi} \text{ (BCD)} \\ \boxed{Su+1} \ \boxed{Su} \text{ (BCD)} \\ - \\ \hline \boxed{CY} \ \boxed{R+1} \ \boxed{R} \text{ (BCD)} \end{array}$ 當產生借位時 CY 將會變為 ON。	輸出需要
BCD SUBTRACT WITH CARRY -BC @-BC 416	$\begin{array}{ c } \hline -BC(416) \\ \hline Mi \\ \hline Su \\ \hline R \\ \hline \end{array}$ Mi: 被減數words Su: 減數words R: 結果words	把 4 位數 (單words) 的Hex資料及/或常數相減再減去進位旗標 (CY)。 $\begin{array}{r} \boxed{Mi} \text{ (BCD)} \\ \boxed{Su} \text{ (BCD)} \\ - \\ \hline \boxed{CY} \\ \boxed{CY} \ \boxed{R} \text{ (BCD)} \end{array}$ 當產生借位時 CY 將會變為 ON。	輸出需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件															
DOUBLE BCD SUBTRACT WITH CARRY -BCL @-BCL 417	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> -BCL(417) <hr/> Mi <hr/> Su <hr/> R </div> <p>Mi: 被減數第一個 words Su: 減數第一個 words R: 結果第一個 words</p>	<p>把 8 位數 (雙 words) 的 BCD 資料及/或常數相減再減去進位旗標(CY)。</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">Mi + 1</td> <td style="border: 1px solid black; padding: 2px;">Mi</td> <td>(BCD)</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Su + 1</td> <td style="border: 1px solid black; padding: 2px;">Su</td> <td>(BCD)</td> </tr> <tr> <td colspan="3" style="text-align: center;">-</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">CY</td> <td style="border: 1px solid black; padding: 2px;">R + 1</td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> </table> <p>(BCD)</p> <p>當產生借位時 CY 將會變為 ON。</p> </div>	Mi + 1	Mi	(BCD)	Su + 1	Su	(BCD)	-			CY	R + 1	R	輸出 需要			
Mi + 1	Mi	(BCD)																
Su + 1	Su	(BCD)																
-																		
CY	R + 1	R																
SIGNED BINARY MULTIPLY * @* 420	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *(420) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘數 words Mr: 乘數 words R: 結果 words</p>	<p>把 4 位數帶符號的 Hex 資料或常數相乘。</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">Md</td> <td>(帶符號 BIN)</td> </tr> <tr> <td style="text-align: center;">×</td> <td style="border: 1px solid black; padding: 2px;">Mr</td> </tr> <tr> <td colspan="2" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 1</td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> </table> <p>(帶符號 BIN)</p> </div>	Md	(帶符號 BIN)	×	Mr	-----		R + 1	R	輸出 需要							
Md	(帶符號 BIN)																	
×	Mr																	

R + 1	R																	
DOUBLE SIGNED BINARY MULTIPLY *L @*L 421	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *L(421) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘數第一個 words Mr: 乘數第一個 words R: 結果第一個 words</p>	<p>把 8 位數帶符號的 Hex 資料或常數相乘。</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">Md + 1</td> <td style="border: 1px solid black; padding: 2px;">Md</td> <td>(帶符號 BIN)</td> </tr> <tr> <td style="text-align: center;">×</td> <td style="border: 1px solid black; padding: 2px;">Mr + 1</td> <td style="border: 1px solid black; padding: 2px;">Mr</td> </tr> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 3</td> <td style="border: 1px solid black; padding: 2px;">R + 2</td> <td style="border: 1px solid black; padding: 2px;">R + 1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;"></td> <td style="border: 1px solid black; padding: 2px;"></td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> </table> <p>(帶符號 BIN)</p> </div>	Md + 1	Md	(帶符號 BIN)	×	Mr + 1	Mr	-----			R + 3	R + 2	R + 1			R	輸出 需要
Md + 1	Md	(帶符號 BIN)																
×	Mr + 1	Mr																

R + 3	R + 2	R + 1																
		R																
UNSIGNED BINARY MULTIPLY *U @*U 422	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *U(422) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘數 words Mr: 乘數 words R: 結果 words</p>	<p>把 4 位數無符號的 Hex 資料或常數相乘。</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">Md</td> <td>(無符號 BIN)</td> </tr> <tr> <td style="text-align: center;">×</td> <td style="border: 1px solid black; padding: 2px;">Mr</td> </tr> <tr> <td colspan="2" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 1</td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> </table> <p>(無符號 BIN)</p> </div>	Md	(無符號 BIN)	×	Mr	-----		R + 1	R	輸出 需要							
Md	(無符號 BIN)																	
×	Mr																	

R + 1	R																	
DOUBLE UNSIGNED BINARY MULTIPLY *UL @*UL 423	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *UL(423) <hr/> Md <hr/> Mr <hr/> R </div> <p>Md: 被乘數第一個 words Mr: 乘數第一個 words R: 結果第一個 words</p>	<p>把 8 位數無符號的 Hex 資料或常數相乘。</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">Md + 1</td> <td style="border: 1px solid black; padding: 2px;">Md</td> <td>(無符號 BIN)</td> </tr> <tr> <td style="text-align: center;">×</td> <td style="border: 1px solid black; padding: 2px;">Mr + 1</td> <td style="border: 1px solid black; padding: 2px;">Mr</td> </tr> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">R + 3</td> <td style="border: 1px solid black; padding: 2px;">R + 2</td> <td style="border: 1px solid black; padding: 2px;">R + 1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;"></td> <td style="border: 1px solid black; padding: 2px;"></td> <td style="border: 1px solid black; padding: 2px;">R</td> </tr> </table> <p>(無符號 BIN)</p> </div>	Md + 1	Md	(無符號 BIN)	×	Mr + 1	Mr	-----			R + 3	R + 2	R + 1			R	輸出 需要
Md + 1	Md	(無符號 BIN)																
×	Mr + 1	Mr																

R + 3	R + 2	R + 1																
		R																

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
BCD MULTIPLY *B @*B 424	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *B(424) Md Mr R </div> <p>Md: 被乘數words Mr: 乘數words R: 結果words</p>	<p>把 4 位數 (單words) 的BCD資料或常數相乘。</p> $ \begin{array}{r} \text{Md} \text{ (BCD)} \\ \times \text{Mr} \text{ (BCD)} \\ \hline \text{R+1} \quad \text{R} \text{ (BCD)} \end{array} $	輸出 需要
DOUBLE BCD MULTIPLY *BL @*BL 425	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> *BL(425) Md Mr R </div> <p>Md: 被乘數第一個 words Mr: 乘數第一個 words R: 結果第一個 words</p>	<p>把 8 位數 (雙words) Hex資料或常數相乘。</p> $ \begin{array}{r} \text{Md+1} \quad \text{Md} \text{ (BCD)} \\ \times \text{Mr+1} \quad \text{Mr} \text{ (BCD)} \\ \hline \text{R+3} \quad \text{R+2} \quad \text{R+1} \quad \text{R} \text{ (BCD)} \end{array} $	輸出 需要
SIGNED BINARY DIVIDE /I @/I 430	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /I(430) Dd Dr R </div> <p>Dd: 被除數words Dr: 除數words R: 結果words</p>	<p>把 4 位數 (單words) 的帶符號Hex資料或常數相除。</p> $ \begin{array}{r} \text{Dd} \text{ (帶符號BIN)} \\ \div \text{Dr} \text{ (帶符號BIN)} \\ \hline \text{R+1} \quad \text{R} \text{ (帶符號BIN)} \\ \text{餘數} \quad \text{商} \end{array} $	輸出 需要
DOUBLE SIGNED BINARY DIVIDE /L @/L 431	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /L(431) Dd Dr R </div> <p>Dd: 被除數第一個 words Dr: 除數第一個 words R: 結果第一個 words</p>	<p>把 8 位數 (雙words) 帶符號的Hex資料或常數相除。</p> $ \begin{array}{r} \text{Dd+1} \quad \text{Dd} \text{ (帶符號BIN)} \\ \div \text{Dr+1} \quad \text{Dr} \text{ (帶符號BIN)} \\ \hline \text{R+3} \quad \text{R+2} \quad \text{R+1} \quad \text{R} \text{ (帶符號BIN)} \\ \text{餘數} \quad \text{商} \end{array} $	輸出 需要
UNSIGNED BINARY DIVIDE /U @/U 432	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> /U(432) Dd Dr R </div> <p>Dd: 被除數words Dr: 除數words R: 結果words</p>	<p>把 4 位數 (單words) 無符號的Hex資料或常數相除。</p> $ \begin{array}{r} \text{Dd} \text{ (無符號BIN)} \\ \div \text{Dr} \text{ (無符號BIN)} \\ \hline \text{R+1} \quad \text{R} \text{ (無符號BIN)} \\ \text{餘數} \quad \text{商} \end{array} $	輸出 需要

指令 助億碼	符號 / 運算元	功能	位置 執行條件																										
DOUBLE UNSIGNED BINARY DIVIDE /UL @/UL 433	<table border="1"> <tr><td>/UL(433)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除數第一個 words Dr: 除數第一個 words R: 結果第一個 words</p>	/UL(433)	Dd	Dr	R	<p>把 8 位數 (雙words) 無符號的Hex資料或常數相除。</p> <table border="1"> <tr> <td>Dd + 1</td> <td>Dd</td> <td>(無符號BIN)</td> </tr> <tr> <td colspan="3" style="text-align: center;">+</td> </tr> <tr> <td>Dr + 1</td> <td>Dr</td> <td>(無符號BIN)</td> </tr> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td>R + 3</td> <td>R + 2</td> <td>R + 1</td> <td>R</td> <td>(無符號BIN)</td> </tr> <tr> <td colspan="2" style="text-align: center;">餘數</td> <td colspan="2" style="text-align: center;">商</td> <td></td> </tr> </table>	Dd + 1	Dd	(無符號BIN)	+			Dr + 1	Dr	(無符號BIN)	-----			R + 3	R + 2	R + 1	R	(無符號BIN)	餘數		商			輸出 需要
/UL(433)																													
Dd																													
Dr																													
R																													
Dd + 1	Dd	(無符號BIN)																											
+																													
Dr + 1	Dr	(無符號BIN)																											

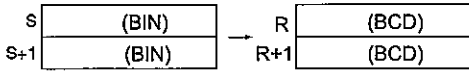
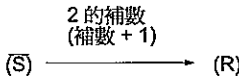
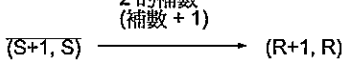
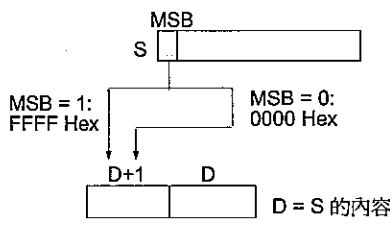
R + 3	R + 2	R + 1	R	(無符號BIN)																									
餘數		商																											
BCD DIVIDE /B @/B 434	<table border="1"> <tr><td>/B(434)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除數 words Dr: 除數 words R: 結果 words</p>	/B(434)	Dd	Dr	R	<p>把 4 位數 (單words) 的BCD資料或常數相除。</p> <table border="1"> <tr> <td>Dd</td> <td>(BCD)</td> </tr> <tr> <td colspan="2" style="text-align: center;">+</td> </tr> <tr> <td>Dr</td> <td>(BCD)</td> </tr> <tr> <td colspan="2" style="text-align: center;">-----</td> </tr> <tr> <td>R + 1</td> <td>R</td> <td>(BCD)</td> </tr> <tr> <td style="text-align: center;">餘數</td> <td style="text-align: center;">商</td> <td></td> </tr> </table>	Dd	(BCD)	+		Dr	(BCD)	-----		R + 1	R	(BCD)	餘數	商		輸出 需要								
/B(434)																													
Dd																													
Dr																													
R																													
Dd	(BCD)																												
+																													
Dr	(BCD)																												

R + 1	R	(BCD)																											
餘數	商																												
DOUBLE BCD DIVIDE /BL @/BL 435	<table border="1"> <tr><td>/BL(435)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 被除數第一個 words Dr: 除數第一個 words R: 結果第一個 words</p>	/BL(435)	Dd	Dr	R	<p>把 8 位數 (雙words) 的BCD資料或常數相除。</p> <table border="1"> <tr> <td>Dd + 1</td> <td>Dd</td> <td>(BCD)</td> </tr> <tr> <td colspan="3" style="text-align: center;">+</td> </tr> <tr> <td>Dr + 1</td> <td>Dr</td> <td>(BCD)</td> </tr> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td>R + 3</td> <td>R + 2</td> <td>R + 1</td> <td>R</td> <td>(BCD)</td> </tr> <tr> <td colspan="2" style="text-align: center;">餘數</td> <td colspan="2" style="text-align: center;">商</td> <td></td> </tr> </table>	Dd + 1	Dd	(BCD)	+			Dr + 1	Dr	(BCD)	-----			R + 3	R + 2	R + 1	R	(BCD)	餘數		商			輸出 需要
/BL(435)																													
Dd																													
Dr																													
R																													
Dd + 1	Dd	(BCD)																											
+																													
Dr + 1	Dr	(BCD)																											

R + 3	R + 2	R + 1	R	(BCD)																									
餘數		商																											

8-10 轉換指令

指令 助億碼	符號 / 運算元	功能	位置 執行條件													
BCD-TO-BINARY BIN @BIN 023	<table border="1"> <tr><td>BIN(023)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 來源 words R: 結果 words</p>	BIN(023)	S	R	<p>把十進位資料轉為BIN資料。</p> <p>S (BCD) → R (BIN)</p>	輸出 需要										
BIN(023)																
S																
R																
DOUBLE BCD- TO-DOUBLE BINARY BINL @BINL 058	<table border="1"> <tr><td>BINL(058)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	BINL(058)	S	R	<p>把八位數的BCD資料轉為八位數的Hex (32 位元BIN) 資料。</p> <table border="1"> <tr> <td>S</td> <td>(BCD)</td> <td>→</td> <td>R</td> <td>(BIN)</td> </tr> <tr> <td>S+1</td> <td>(BCD)</td> <td></td> <td>R+1</td> <td>(BIN)</td> </tr> </table>	S	(BCD)	→	R	(BIN)	S+1	(BCD)		R+1	(BIN)	輸出 需要
BINL(058)																
S																
R																
S	(BCD)	→	R	(BIN)												
S+1	(BCD)		R+1	(BIN)												
BINARY-TO-BCD BCD @BCD 024	<table border="1"> <tr><td>BCD(024)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 來源 words R: 結果 words</p>	BCD(024)	S	R	<p>把 words 的BIN資料轉為 words 的BCD資料。</p> <p>S (BIN) → R (BCD)</p>	輸出 需要										
BCD(024)																
S																
R																

指令 助億碼	符號 / 運算元	功能	位置 執行條件
DOUBLE BINARY-TO- DOUBLE BCD BCDL @BCDL 059	BCDL(059) S R	把八位數的Hex (32 位元BIN) 資料轉為八位數的BCD資料。 	輸出 需要
2'S COMPLEMENT NEG @NEG 160	NEG(160) S R	計算一個Hex words資料之 2 的補數。 	輸出 需要
DOUBLE 2'S COMPLEMENT NEGL @NEGL 161	NEGL(161) S R	計算一個Hex雙words資料之 2 的補數。 	輸出 需要
16-BIT TO 32-BIT SIGNED BINARY SIGN @SIGN 600	SIGN(600) S R	把一個 16 位元的帶符號BIN值擴充為 32 位元的等效值。 	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件
DATA DECODER MLPX @MLPX 076	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> MLPX(076) <hr/> S <hr/> C <hr/> R </div> <p>S: 來源words C: 控制words R: 第一個結果words</p>	<p>讀入來源words中指定位元 (或位元組) 的數值，再把結果words (或 16 words 範圍) 中相對應的位元變為 ON，並把結果words (或 16 words範圍) 中其他的位元變為 OFF。 4 至 16 位元轉換</p> <p>4 至 16 位元解碼器 (R 的第 m 個位元變為 ON)</p> <p>8 至 256 位元轉換</p> <p>8 至 256 位元解碼器 (R 至 R+15 的第 m 個位元變為 ON)</p> <p>當 l 指定 2 個位元組時， 會使用兩個 16 words 範圍。</p>	輸出 需要

指令	符號 / 運算元	功能	位置 執行條件				
<p>DATA ENCODER 助憶碼 DMPX @DMPX 077</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>DMPX(077)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> <tr><td>C</td></tr> </table> <p>S: 第一個來源 words R: 結果 words C: 控制 words</p>	DMPX(077)	S	R	C	<p>找出來源 words (或 16 words 範圍) 中第一個或最後一個 ON 的位置，並把這個值寫入結果 words 中指定的位元 (或位元組)。 16 至 4 位元轉換</p> <p>16 至 4 解碼 (最左邊的位元會寫入到 R。)</p> <p>256 至 8 位元轉換</p> <p>256 至 8 解碼 (16 個 words 範圍 (m) 中的最左位元會寫入到 R。)</p>	<p>輸出 需要</p>
DMPX(077)							
S							
R							
C							
<p>ASCII CONVERT ASC @ASC 086</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>ASC(086)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> <p>S: 來源 words Di: 位數指定器 D: 第一個目的 words</p>	ASC(086)	S	Di	D	<p>把來源 words 中的 4 位元 Hex 數轉換為它們的 8 位元 ASCII 碼。</p> <p>HEX ↓ ASCII</p>	<p>輸出 需要</p>
ASC(086)							
S							
Di							
D							

指令 助憶碼	符號 / 運算元	功能	位置 執行條件																																																																																																																																																								
ASCII TO HEX HEX @HEX 162	<table border="1" style="margin-bottom: 10px;"> <tr><td>HEX(162)</td></tr> <tr><td>S</td></tr> <tr><td>Di</td></tr> <tr><td>D</td></tr> </table> S: 第一個來源 words Di: 位數指定器 D: 目的 words	HEX(162)	S	Di	D	把來源 words 中最多 4 個位元組的 ASCII 碼數轉換為 Hex 數並把這些位元寫入指定的目的位址。 <div style="text-align: center; margin-top: 10px;"> <table style="margin-left: auto; margin-right: auto;"> <tr><td colspan="4">C: 0021</td></tr> <tr><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0/1</td><td style="border: 1px solid black;">n</td><td style="border: 1px solid black;">m</td></tr> </table> </div>	C: 0021				0	0/1	n	m	輸出 需要																																																																																																																																												
HEX(162)																																																																																																																																																											
S																																																																																																																																																											
Di																																																																																																																																																											
D																																																																																																																																																											
C: 0021																																																																																																																																																											
0	0/1	n	m																																																																																																																																																								
COLUMN TO LINE LINE @LINE 063	<table border="1" style="margin-bottom: 10px;"> <tr><td>LINE(063)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>D</td></tr> </table> S: 第一個來源 words N: 位元數 D: 目的 words	LINE(063)	S	N	D	把 16 words 範圍中的一行位元 (即 16 個連續 words 中相同位元的數) 轉換為目的 words 的 16 個位元。 <div style="text-align: center; margin-top: 10px;"> <table style="margin-left: auto; margin-right: auto;"> <tr><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td></tr> <tr><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td></tr> <tr><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td></tr> <tr><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td></tr> <tr><td colspan="16" style="text-align: center;">⋮</td></tr> <tr><td style="border: 1px solid black;">S+15</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td></tr> </table> </div> <div style="text-align: center; margin-top: 20px;"> <table style="margin-left: auto; margin-right: auto;"> <tr><td style="border: 1px solid black;">D</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td></tr> </table> </div>	0	0	0	1	1	1	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	1	1	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1	⋮																S+15	0	1	1	0	0	0	1	1	0	0	0	1	0	1	0	D	0	0	0	1	1	1	1	輸出 需要																																													
LINE(063)																																																																																																																																																											
S																																																																																																																																																											
N																																																																																																																																																											
D																																																																																																																																																											
0	0	0	1	1	1	0	0	0	1	0	0	0	0	1																																																																																																																																													
1	1	0	1	0	0	1	0	0	1	1	1	0	0	0	1																																																																																																																																												
0	0	0	1	1	0	1	0	0	1	0	0	1	1	1	1																																																																																																																																												
1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	1																																																																																																																																												
⋮																																																																																																																																																											
S+15	0	1	1	0	0	0	1	1	0	0	0	1	0	1	0																																																																																																																																												
D	0	0	0	1	1	1	1																																																																																																																																																				
LINE TO COLUMN COLM @COLM 064	<table border="1" style="margin-bottom: 10px;"> <tr><td>COLM(064)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>N</td></tr> </table> S: 來源 words D: 第一個目的 words N: 位元數	COLM(064)	S	D	N	把來源 words 的 16 個位元轉換為目的 words 的 16 words 範圍中的一行位元 (即 16 個連續 words 中相同位元的數)。 <div style="text-align: center; margin-top: 10px;"> <table style="margin-left: auto; margin-right: auto;"> <tr><td style="border: 1px solid black;">S</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td></tr> </table> </div> <div style="text-align: center; margin-top: 20px;"> <table style="margin-left: auto; margin-right: auto;"> <tr><td style="border: 1px solid black;">D</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td></tr> <tr><td style="border: 1px solid black;">D+1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td></tr> <tr><td style="border: 1px solid black;">D+2</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td></tr> <tr><td style="border: 1px solid black;">D+3</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td></tr> <tr><td colspan="22" style="text-align: center;">⋮</td></tr> <tr><td style="border: 1px solid black;">D+15</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td><td style="border: 1px solid black;">1</td><td style="border: 1px solid black;">0</td></tr> </table> </div>	S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	D	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	D+1	1	1	0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	1	D+2	0	0	0	1	1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	D+3	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	⋮																						D+15	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0	輸出 需要
COLM(064)																																																																																																																																																											
S																																																																																																																																																											
D																																																																																																																																																											
N																																																																																																																																																											
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1																																																																																																																																								
D	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1																																																																																																																																							
D+1	1	1	0	1	0	0	1	0	0	1	1	1	0	0	0	1	0	0	0	1																																																																																																																																							
D+2	0	0	0	1	1	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1																																																																																																																																							
D+3	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1																																																																																																																																							
⋮																																																																																																																																																											
D+15	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0																																																																																																																																						

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
SIGNED BCD-TO-BINARY BINS @BINS 470	<table border="1"> <tr><td>BINS(470)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制words S: 來源words D: 目的words</p>	BINS(470)	C	S	D	把一個帶符號 BCD 資料轉換為一個帶符號BIN(二進制) 資料的words。 	輸出 需要
BINS(470)							
C							
S							
D							
DOUBLE SIGNED BCD-TO-BINARY BISL @BISL 472	<table border="1"> <tr><td>BISL(472)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制words S: 第一個來源words D: 第一個目的words</p>	BISL(472)	C	S	D	把一個帶符號雙words BCD 資料轉換為一個帶符號BIN資料的雙words。 	輸出 需要
BISL(472)							
C							
S							
D							
SIGNED BINARY-TO-BCD BCDS @BCDS 471	<table border="1"> <tr><td>BCDS(471)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制words S: 來源words D: 目的words</p>	BCDS(471)	C	S	D	把一個帶符號BIN資料轉換為一個帶符號 BCD 資料的words。 	輸出 需要
BCDS(471)							
C							
S							
D							
DOUBLE SIGNED BINARY-TO-BCD BDSL @BDSL 473	<table border="1"> <tr><td>BDSL(473)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制words S: 第一個來源words D: 第一個目的words</p>	BDSL(473)	C	S	D	把一個帶符號雙字組BIN資料轉換為一個帶符號 BCD 資料的雙words。 	輸出 需要
BDSL(473)							
C							
S							
D							

8-11 邏輯 (Logic) 指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件																			
LOGICAL AND ANDW @ANDW 034	<table border="1"> <tr><td>ANDW(034)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	ANDW(034)	I ₁	I ₂	R	<p>對words資料或常數中的單words的對應位元作邏輯 AND 運算。</p> <p>$I_1, I_2 \rightarrow R$</p> <table border="1"> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I ₁	I ₂	R	1	1	1	1	0	0	0	1	0	0	0	0	輸出 需要
ANDW(034)																						
I ₁																						
I ₂																						
R																						
I ₁	I ₂	R																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	0																				
DOUBLE LOGICAL AND ANDL @ANDL 610	<table border="1"> <tr><td>ANDL(610)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	ANDL(610)	I ₁	I ₂	R	<p>對words資料或常數中的雙words的對應位元作邏輯 AND 運算。</p> <p>$(I_1, I_1+1), (I_2, I_2+1) \rightarrow (R, R+1)$</p> <table border="1"> <tr><th>I_{1, I₁+1}</th><th>I_{2, I₂+1}</th><th>R, R+1</th></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I _{1, I₁+1}	I _{2, I₂+1}	R, R+1	1	1	1	1	0	0	0	1	0	0	0	0	輸出 需要
ANDL(610)																						
I ₁																						
I ₂																						
R																						
I _{1, I₁+1}	I _{2, I₂+1}	R, R+1																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	0																				
LOGICAL OR ORW @ORW 035	<table border="1"> <tr><td>ORW(035)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	ORW(035)	I ₁	I ₂	R	<p>對words資料或常數中的單words的對應位元作邏輯 OR 運算。</p> <p>$I_1 + I_2 \rightarrow R$</p> <table border="1"> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I ₁	I ₂	R	1	1	1	1	0	1	0	1	1	0	0	0	輸出 需要
ORW(035)																						
I ₁																						
I ₂																						
R																						
I ₁	I ₂	R																				
1	1	1																				
1	0	1																				
0	1	1																				
0	0	0																				
DOUBLE LOGICAL OR ORWL @ORWL 611	<table border="1"> <tr><td>ORWL(611)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	ORWL(611)	I ₁	I ₂	R	<p>對words資料或常數中的雙words的對應位元作邏輯 OR 運算。</p> <p>$(I_1, I_1+1) + (I_2, I_2+1) \rightarrow (R, R+1)$</p> <table border="1"> <tr><th>I_{1, I₁+1}</th><th>I_{2, I₂+1}</th><th>R, R+1</th></tr> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I _{1, I₁+1}	I _{2, I₂+1}	R, R+1	1	1	1	1	0	1	0	1	1	0	0	0	輸出 需要
ORWL(611)																						
I ₁																						
I ₂																						
R																						
I _{1, I₁+1}	I _{2, I₂+1}	R, R+1																				
1	1	1																				
1	0	1																				
0	1	1																				
0	0	0																				
EXCLUSIVE OR XORW @XORW 036	<table border="1"> <tr><td>XORW(036)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	XORW(036)	I ₁	I ₂	R	<p>對words資料或常數中的單words的對應位元作邏輯互斥 OR 運算。</p> <p>$I_1, \bar{I}_2 + \bar{I}_1, I_2 \rightarrow R$</p> <table border="1"> <tr><th>I₁</th><th>I₂</th><th>R</th></tr> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	I ₁	I ₂	R	1	1	0	1	0	1	0	1	1	0	0	0	輸出 需要
XORW(036)																						
I ₁																						
I ₂																						
R																						
I ₁	I ₂	R																				
1	1	0																				
1	0	1																				
0	1	1																				
0	0	0																				

指令 助憶碼	符號 / 運算元	功能	位置 執行條件																			
DOUBLE EXCLUSIVE OR XORL @XORL 612	<table border="1"> <tr><td>XORL(612)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	XORL(612)	I ₁	I ₂	R	<p>對words資料或常數中的雙words的對應位元作邏輯互斥 OR 運算。</p> $(I_1.I_1+1). (\overline{I_2.I_2+1}) + (\overline{I_1.I_1+1}). (I_2.I_2+1) \rightarrow (R, R+1)$ <table border="1"> <thead> <tr> <th>I₁.I₁+1</th> <th>I₂.I₂+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	I ₁ .I ₁ +1	I ₂ .I ₂ +1	R, R+1	1	1	0	1	0	1	0	1	1	0	0	0	輸出 需要
XORL(612)																						
I ₁																						
I ₂																						
R																						
I ₁ .I ₁ +1	I ₂ .I ₂ +1	R, R+1																				
1	1	0																				
1	0	1																				
0	1	1																				
0	0	0																				
EXCLUSIVE NOR XNRW @XNRW 037	<table border="1"> <tr><td>XNRW(037)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 結果words</p>	XNRW(037)	I ₁	I ₂	R	<p>對words資料或常數中的單words的對應位元作邏輯互斥 NOR 運算。</p> $I_1.I_2 + \overline{I_1.I_2} \rightarrow R$ <table border="1"> <thead> <tr> <th>I₁</th> <th>I₂</th> <th>R</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I ₁	I ₂	R	1	1	1	1	0	0	0	1	0	0	0	1	輸出 需要
XNRW(037)																						
I ₁																						
I ₂																						
R																						
I ₁	I ₂	R																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	1																				
DOUBLE EXCLUSIVE NOR XNRL @XNRL 613	<table border="1"> <tr><td>XNRL(613)</td></tr> <tr><td>I₁</td></tr> <tr><td>I₂</td></tr> <tr><td>R</td></tr> </table> <p>I₁: 輸入 1 I₂: 輸入 2 R: 第一個結果words</p>	XNRL(613)	I ₁	I ₂	R	<p>對words資料或常數中的雙words的對應位元作邏輯互斥 NOR 運算。</p> $(I_1.I_1+1). (I_2.I_2+1) + (\overline{I_1.I_1+1}). (\overline{I_2.I_2+1}) \rightarrow (R, R+1)$ <table border="1"> <thead> <tr> <th>I₁.I₁+1</th> <th>I₂.I₂+1</th> <th>R, R+1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </tbody> </table>	I ₁ .I ₁ +1	I ₂ .I ₂ +1	R, R+1	1	1	1	1	0	0	0	1	0	0	0	1	輸出 需要
XNRL(613)																						
I ₁																						
I ₂																						
R																						
I ₁ .I ₁ +1	I ₂ .I ₂ +1	R, R+1																				
1	1	1																				
1	0	0																				
0	1	0																				
0	0	1																				
COMPLEMENT COM @COM 029	<table border="1"> <tr><td>COM(029)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: words</p>	COM(029)	Wd	<p>把 Wd 中所有 ON 的位元變為 OFF 而把所有 OFF 的位元變為 ON。</p> $\overline{Wd} \rightarrow Wd: 1 \rightarrow 0 \text{ and } 0 \rightarrow 1$	輸出 需要																	
COM(029)																						
Wd																						
DOUBLE COMPLEMENT COML @COML 614	<table border="1"> <tr><td>COML(614)</td></tr> <tr><td>Wd</td></tr> </table> <p>Wd: words</p>	COML(614)	Wd	<p>把 Wd 與 Wd+1 中所有 ON 的位元變為 OFF 而把所有 OFF 的位元變為 ON。</p> $\overline{(Wd+1, Wd)} \rightarrow (Wd+1, Wd)$	輸出 需要																	
COML(614)																						
Wd																						

8-12 特殊算術指令


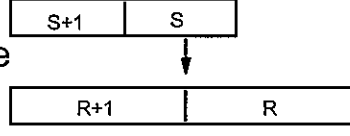
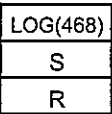
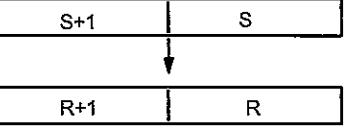
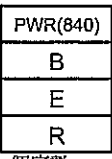
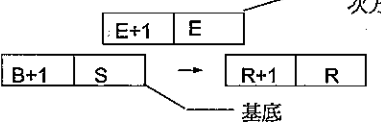



指令 助憶碼	符號 / 運算元	功能	位置 執行條件				
BINARY ROOT ROT B @ROT B 620	<table border="1"> <tr><td>ROT B(620)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 結果 words</p>	ROT B(620)	S	R	<p>計算指定 words 之 32 位元 Hex 數的平方根並把結果的整數部份輸出到指定的結果 words。</p>	輸出 需要	
ROT B(620)							
S							
R							
BCD SQUARE ROOT ROOT @ROOT 072	<table border="1"> <tr><td>ROOT(072)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 結果 words</p>	ROOT(072)	S	R	<p>計算 8 位元 BCD 數的平方根並把結果的整數部份輸出到指定的結果 words。</p>	輸出 需要	
ROOT(072)							
S							
R							
ARITHMETIC PROCESS APR @APR 069	<table border="1"> <tr><td>APR(069)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>C: 控制 words S: 來源 words R: 結果 words</p>	APR(069)	C	S	R	<p>計算來源資料的正弦、餘弦或線性外插值。 線性外插功能可接受任何能夠以線段概括的 X 與 Y 之間的關係。</p>	輸出 需要
APR(069)							
C							
S							
R							
FLOATING POINT DIVIDE FDIV @FDIV 079	<table border="1"> <tr><td>FDIV(079)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 第一個被除數 words Dr: 第一個除數 words R: 第一個結果 words</p>	FDIV(079)	Dd	Dr	R	<p>把兩個 7 位浮點數相除。其中浮點數是以 BCD 8 個 words 來表示。</p>	輸出 需要
FDIV(079)							
Dd							
Dr							
R							
BIT COUNTER BCNT @BCNT 067	<table border="1"> <tr><td>BCNT(067)</td></tr> <tr><td>N</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>N: words 數 S: 第一個來源 words R: 結果 words</p>	BCNT(067)	N	S	R	<p>計算指定 words 中 ON 位元的總數。</p>	輸出 需要
BCNT(067)							
N							
S							
R							

8-13 浮點運算指令

指令	助憶碼	符號 / 運算元	功能	位置執行條件
FLOATING TO 16-BIT	FIX @FIX 450	FIX(450) S R	把一個 32 位元的浮點數值轉換為 16 位元的帶符號BIN值，並把結果放到指定的結果words。 浮點資料 (32 位元) 帶符號BIN資料 (16 位元)	輸出需要
FLOATING TO 32-BIT	FIXL @FIXL 451	FIXL(451) S R	把一個 32 位元的浮點數值轉換為 32 位元的帶符號BIN值，並把結果放到指定的結果words。 浮點資料 (32 位元) 帶符號BIN資料 (32 位元)	輸出需要
16-BIT TO FLOATING	FLT @FLT 452	FLT(452) S R	把一個 16 位元的帶符號BIN值轉換為 32 位元的浮點數值，並把結果放到指定的結果words。 帶符號BIN資料 (16 位元) 浮點資料 (32 位元)	輸出需要
32-BIT TO FLOATING	FTL @FTL 453	FTL(453) S R	把一個 32 位元的帶符號BIN值轉換為 32 位元的浮點數值，並把結果放到指定的結果words。 帶符號BIN資料 (32 位元) 浮點資料 (32 位元)	輸出需要
FLOATING-POINT ADD	+F @+F 454	+F(454) Au Ad R	把兩個 32 位元的浮點數值相加，並把結果放到指定的結果words。 被加數 (浮點資料，32 位元) 加數 (浮點資料，32 位元) 結果 (浮點資料，32 位元)	輸出需要
FLOATING-POINT SUBTRACT	-F @-F 455	-F(455) Mi Su R	把兩個 32 位元的浮點數值相減，並把結果放到指定的結果words。 被減數 (浮點資料，32 位元) 減數 (浮點資料，32 位元) 結果 (浮點資料，32 位元)	輸出需要

指令 助億碼	符號 / 運算元	功能	位置 執行條件
FLOATING-POINT MULTIPLY *F @*F 456	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">*F(456)</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">Md</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">Mr</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">R</div> </div> <p>Md: 第一個被乘數 words Mr: 第一個乘數 words R: 第一個結果 words</p>	<p>把兩個 32 位元的浮點數值相乘，並把結果放到指定的結果words。</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Md+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Md</div> <div style="margin-left: 20px;">被乘數 (浮點資料, 32 位元)</div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 5px;">×</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Mr+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Mr</div> <div style="margin-left: 20px;">乘數 (浮點資料, 32 位元)</div> </div> <hr style="width: 100%;"/> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R</div> <div style="margin-left: 20px;">結果 (浮點資料, 32 位元)</div> </div>	輸出需要
FLOATING-POINT DIVIDE /F @/F 457	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">/F(457)</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">Dd</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">Dr</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">R</div> </div> <p>Dd: 第一個被除數 words Dr: 第一個除數 words R: 第一個結果 words</p>	<p>把兩個 32 位元的浮點數值相除，並把結果放到指定的結果words。</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Dd+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Dd</div> <div style="margin-left: 20px;">被除數 (浮點資料, 32 位元)</div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 5px;">÷</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Dr+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">Dr</div> <div style="margin-left: 20px;">除數 (浮點資料, 32 位元)</div> </div> <hr style="width: 100%;"/> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R</div> <div style="margin-left: 20px;">結果 (浮點資料, 32 位元)</div> </div>	輸出需要
DEGREES TO RADIANS RAD @RAD 458	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">RAD(458)</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">R</div> </div> <p>S: 第一個來源 words R: 第一個結果 words</p>	<p>把一個 32 位元的浮點數由度度量轉換為弧度量，並把結果放到指定的結果words。</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">來源 (度度量, 32 位元浮點資料)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R</div> <div style="margin-left: 20px;">結果 (弧度量, 32 位元浮點資料)</div> </div>	輸出需要
RADIANS TO DEGREES DEG @DEG 459	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">DEG(459)</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">R</div> </div> <p>S: 第一個來源 words R: 第一個結果 words</p>	<p>把一個 32 位元的浮點數由弧度量轉換為度度量，並把結果放到指定的結果words。</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">來源 (弧度量, 32 位元浮點資料)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R</div> <div style="margin-left: 20px;">結果 (度度量, 32 位元浮點資料)</div> </div>	輸出需要
SINE SIN @SIN 460	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">SIN(460)</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">R</div> </div> <p>S: 第一個來源 words R: 第一個結果 words</p>	<p>計算一個 32 位元浮點數 (弧度量) 的正弦值，並把結果放到指定的結果words。</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 5px;">SIN (</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">來源 (32 位元浮點資料)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R</div> <div style="margin-left: 20px;">結果 (32 位元浮點資料)</div> </div>	輸出需要
COSINE COS @COS 461	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">COS(461)</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">S</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">R</div> </div> <p>S: 第一個來源 words R: 第一個結果 words</p>	<p>計算一個 32 位元浮點數 (弧度量) 的餘弦值，並把結果放到指定的結果words。</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 5px;">COS (</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">S</div> <div style="margin-left: 20px;">來源 (32 位元浮點資料)</div> </div> <div style="text-align: center; margin-bottom: 10px;">↓</div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R+1</div> <div style="border: 1px solid black; padding: 2px; margin-right: 10px;">R</div> <div style="margin-left: 20px;">結果 (32 位元浮點資料)</div> </div>	輸出需要

指令 助億碼	符號 / 運算元	功能	位置 執行條件			
TANGENT TAN @TAN 462	<table border="1"> <tr><td>TAN(462)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	TAN(462)	S	R	<p>計算一個 32 位元浮點數 (經度量) 的正切值，並把結果放到指定的結果 words。</p> $\text{TAN} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>來源 (32 位元浮點資料)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>結果 (32 位元浮點資料)</p>	輸出需要
TAN(462)						
S						
R						
ARC SINE ASIN @ASIN 463	<table border="1"> <tr><td>ASIN(463)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	ASIN(463)	S	R	<p>計算一個 32 位元浮點數的反正弦值，並把結果放到指定的結果 words。 (反正弦函數是正弦函數的反函數；回傳的是能產生所給定介於 -1 與 1 間之正弦值的值。)</p> $\text{SIN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>來源 (32 位元浮點資料)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>結果 (32 位元浮點資料)</p>	輸出需要
ASIN(463)						
S						
R						
ARC COSINE ACOS @ACOS 464	<table border="1"> <tr><td>ACOS(464)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	ACOS(464)	S	R	<p>計算一個 32 位元浮點數的反餘弦值，並把結果放到指定的結果 words。 (反餘弦函數是餘弦函數的反函數；回傳的是能產生所給定介於 -1 與 1 間之餘弦值的值。)</p> $\text{COS}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>來源 (32 位元浮點資料)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>結果 (32 位元浮點資料)</p>	輸出需要
ACOS(464)						
S						
R						
ARC TANGENT ATAN @ATAN 465	<table border="1"> <tr><td>ATAN(465)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	ATAN(465)	S	R	<p>計算一個 32 位元浮點數的反正切值，並把結果放到指定的結果 words。 (反正切函數是正切函數的反函數；回傳的是能產生所給定正切值的值。)</p> $\text{TAN}^{-1} \left(\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$ <p>來源 (32 位元浮點資料)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>結果 (32 位元浮點資料)</p>	輸出需要
ATAN(465)						
S						
R						
SQUARE ROOT SQRT @SQRT 466	<table border="1"> <tr><td>SQRT(466)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	SQRT(466)	S	R	<p>計算一個 32 位元浮點數的開根號，並把結果放到指定的結果 words。</p> $\sqrt{\begin{array}{ c c } \hline \text{S+1} & \text{S} \\ \hline \end{array}}$ <p>來源 (32 位元浮點資料)</p> $\begin{array}{ c c } \hline \text{R+1} & \text{R} \\ \hline \end{array}$ <p>結果 (32 位元浮點資料)</p>	輸出需要
SQRT(466)						
S						
R						

指令	助憶碼	符號 / 運算元	功能	位置 執行條件
EXPONENT	EXP @EXP 467	 <p>S:第一個來源 words R:第一個結果 words</p>	<p>計算一個 32 位元浮點數的自然指數 (以 e 為底) 值, 並把結果放到指定的結果 words。</p>  <p>來源 (32 位元浮點資料)</p> <p>結果 (32 位元浮點資料)</p>	輸出 需要
LOGARITHM	LOG @LOG 468	 <p>S:第一個來源 words R:第一個結果 words</p>	<p>計算一個 32 位元浮點數的自然對數 (以 e 為底) 值, 並把結果放到指定的結果 words。</p>  <p>來源 (32 位元浮點資料)</p> <p>結果 (32 位元浮點資料)</p>	輸出 需要
EXPONENTIAL POWER	PWR @PWR 840	 <p>B:第一個底數 words E:第一個指數 words R:第一個結果 words</p>	<p>把一個 32 位元浮點數作為另一個 32 位元浮點數的次方。</p> 	輸出 需要
浮點符號比較 (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	LD, AND, 或 OR + =F (329), <>F (330), <F (331), <=F (332), >F (333), 或 >=F (334)	<p>使用 LD:</p>  <p>使用 AND:</p>  <p>使用 OR:</p>  <p>S1: 比較資料 1 S2: 比較資料 2</p>	<p>比較指定的單精度資料 (32 位元) 或常數並且在結果成立時建立 ON 的執行條件。</p> <p>有三種符號可以用於浮點符號比較指令: LD (載入), AND, 與 OR。</p>	LD: 不需要 AND 或 或: 需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件				
FLOATING- POINT TO ASCII (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) FSTR @FSTR 448	<table border="1"> <tr><td>FSTR(448)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: 第一個來源 words C: 控制 words D: 目的 words</p>	FSTR(448)	S	C	D	把指定的 32 位元單精度浮點資料轉變為字串 (ASCII) 的單精度浮點資料(小數點或指數型式) 並把結果輸出到目的 words。	輸出需要
FSTR(448)							
S							
C							
D							
ASCII TO FLOATING POINT (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) FVAL @FVAL 449	<table border="1"> <tr><td>FVAL(449)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 來源 words D: 第一個目的 words</p>	FVAL(449)	S	D	把指定字串 (ASCII) 的單精度浮點資料 (有小數點或指數型式) 轉變為 32 位元單精度浮點資料並把結果輸出到目的 words。	輸出需要	
FVAL(449)							
S							
D							

8-14 雙精度浮點指令點指令 (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)

指令 助憶碼	符號 / 運算元	功能	位置 執行條件			
DOUBLE FLOATING TO 16-BIT BINARY FIXD @FIXD 841	<table border="1"> <tr><td>FIXD(841)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 第一個來源 words D: 目的 words</p>	FIXD(841)	S	D	把指定的雙精度浮點資料 (64 位元) 轉換為 16 位元帶符號 BIN 資料並把結果輸出到目的 words。	輸出需要
FIXD(841)						
S						
D						
DOUBLE FLOATING TO 32-BIT BINARY FIXLD @FIXLD 842	<table border="1"> <tr><td>FIXLD(842)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 第一個來源 words D: 第一個目的 words</p>	FIXLD(842)	S	D	把指定的雙精度浮點資料 (64 位元) 轉換為 32 位元帶符號 BIN 資料並把結果輸出到目的 words。	輸出需要
FIXLD(842)						
S						
D						
16-BIT BINARY TO DOUBLE FLOATING DBL @DBL 843	<table border="1"> <tr><td>DBL(843)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 來源 words D: 第一個目的 words</p>	DBL(843)	S	D	把指定的 16 位元帶符號 BIN 資料轉換為雙精度浮點資料 (64 位元) 並把結果輸出到目的 words。	輸出需要
DBL(843)						
S						
D						

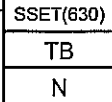
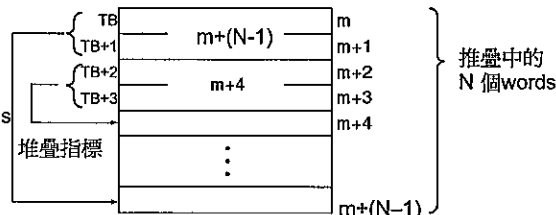
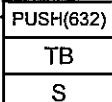
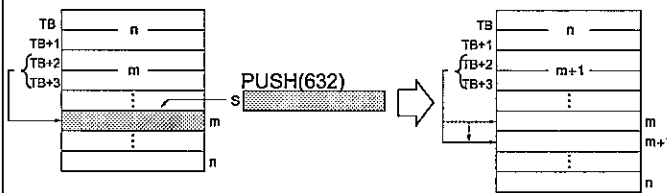
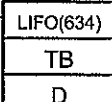
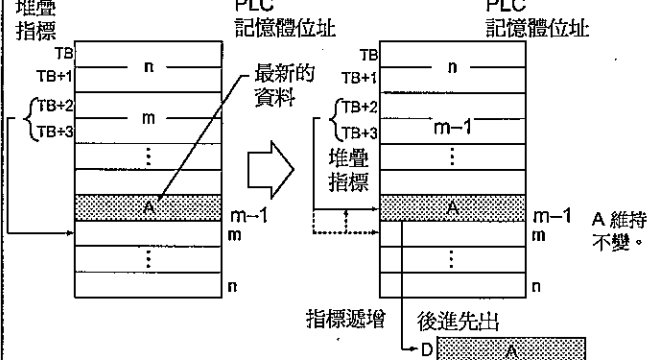
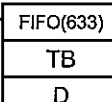
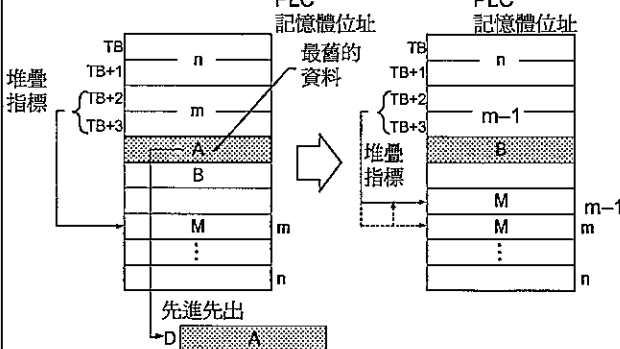
指令 助億碼	符號 / 運算元	功能	位置 執行條件				
32-BIT BINARY TO DOUBLE FLOATING DBLL @DBLL 844	<table border="1"> <tr><td>DBLL(844)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S:第一個來源 words D:第一個目的 words</p>	DBLL(844)	S	D	把指定的 32 位元帶符號 BIN 資料轉換為雙精度浮點資料 (64 位元) 並把結果輸出到目的 words。	輸出 需要	
DBLL(844)							
S							
D							
DOUBLE FLOATING POINT ADD +D @+D 845	<table border="1"> <tr><td>+D(845)</td></tr> <tr><td>Au</td></tr> <tr><td>Ad</td></tr> <tr><td>R</td></tr> </table> <p>Au:第一個被加數 words Ad:第一個加數 words R:第一個結果 words</p>	+D(845)	Au	Ad	R	把指定的雙精度浮點數值 (均為 64 位元) 相加, 並把結果輸出到結果 words。	輸出 需要
+D(845)							
Au							
Ad							
R							
DOUBLE FLOATING POINT SUBTRACT -D @-D 846	<table border="1"> <tr><td>-D(846)</td></tr> <tr><td>Mi</td></tr> <tr><td>Su</td></tr> <tr><td>R</td></tr> </table> <p>Mi: 第一個被減數 words Su: 第一個減數 words R:第一個結果 words</p>	-D(846)	Mi	Su	R	把指定的雙精度浮點數值 (均為 64 位元) 相減, 並把結果輸出到結果 words。	輸出 需要
-D(846)							
Mi							
Su							
R							
DOUBLE FLOATI NGPOINT MULTIPLY *D @*D 847	<table border="1"> <tr><td>*D(847)</td></tr> <tr><td>Md</td></tr> <tr><td>Mr</td></tr> <tr><td>R</td></tr> </table> <p>Md: 第一個被乘數 words Mr: 第一個乘數 words R:第一個結果 words</p>	*D(847)	Md	Mr	R	把指定的雙精度浮點數值 (均為 64 位元) 相乘, 並把結果輸出到結果 words。	輸出 需要
*D(847)							
Md							
Mr							
R							
DOUBLE FLOATING POINT DIVIDE /D @/D 848	<table border="1"> <tr><td>/D(848)</td></tr> <tr><td>Dd</td></tr> <tr><td>Dr</td></tr> <tr><td>R</td></tr> </table> <p>Dd: 第一個被除數 words Dr: 第一個除數 words R:第一個結果 words</p>	/D(848)	Dd	Dr	R	把指定的雙精度浮點數值 (均為 64 位元) 相除, 並把結果輸出到結果 words。	輸出 需要
/D(848)							
Dd							
Dr							
R							

指令 助憶碼	符號 / 運算元	功能	位置 執行條件			
DOUBLE DEGREES TO RADIANS RADD @RADD 849	<table border="1"> <tr><td>RADD(849)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	RADD(849)	S	R	把指定的雙精度浮點資料 (64 位元) 由度度量轉換為弧度量，並把結果輸出到結果 words。	輸出需要
RADD(849)						
S						
R						
DOUBLE RADIANS TO DEGREES DEGD @DEGD 850	<table border="1"> <tr><td>DEGD(850)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	DEGD(850)	S	R	把指定的雙精度浮點資料 (64 位元) 由弧度量轉換為度度量，並把結果輸出到結果 words。	輸出需要
DEGD(850)						
S						
R						
DOUBLE SINE SIND @SIND 851	<table border="1"> <tr><td>SIND(851)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	SIND(851)	S	R	計算所指定的雙精度浮點資料 (64 位元) 角度 (弧度量) 之正弦值，並把結果輸出到結果 words。	輸出需要
SIND(851)						
S						
R						
DOUBLE COSINE COSD @COSD 852	<table border="1"> <tr><td>COSD(852)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	COSD(852)	S	R	計算所指定的雙精度浮點資料 (64 位元) 角度 (弧度量) 之餘弦值，並把結果輸出到結果 words。	輸出需要
COSD(852)						
S						
R						
DOUBLE TANGENT TAND @TAND 853	<table border="1"> <tr><td>TAND(853)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	TAND(853)	S	R	計算所指定的雙精度浮點資料 (64 位元) 角度 (弧度量) 之正切值，並把結果輸出到結果 words。	輸出需要
TAND(853)						
S						
R						
DOUBLE ARC SINE ASIND @ASIND 854	<table border="1"> <tr><td>ASIND(854)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S: 第一個來源 words R: 第一個結果 words</p>	ASIND(854)	S	R	由指定的正弦值之雙精度浮點資料 (64 位元) 計算角度 (弧度量)，並把結果輸出到結果 words。(反正弦函數為正弦函數的反函數；它會傳回產生所給定介於 -1 與 1 之間的正弦值之角度。)	輸出需要
ASIND(854)						
S						
R						

指令 助億碼	符號 / 運算元	功能	位置 執行條件			
DOUBLE ARC COSINE ACOSD @ACOSD 855	<table border="1"> <tr><td>ACOSD(855)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	ACOSD(855)	S	R	由指定的餘弦值之雙精度浮點資料 (64 位元) 計算角度 (弧度量), 並把結果輸出到結果 words。(反餘弦函數為餘弦函數的反函數: 它會傳回產生所給定介於 -1 與 1 之間的餘弦值之角度。)	輸出 需要
ACOSD(855)						
S						
R						
DOUBLE ARC TANGENT ATAND @ATAND 856	<table border="1"> <tr><td>ATAND(856)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	ATAND(856)	S	R	由指定的正切值之雙精度浮點資料 (64 位元) 計算角度 (弧度量), 並把結果輸出到結果 words。(反正切函數為正切函數的反函數: 它會傳回產生所給定介於 -1 與 1 之間的正切值之角度。)	輸出 需要
ATAND(856)						
S						
R						
DOUBLE SQUARE ROOT SQRTD @SQRTD 857	<table border="1"> <tr><td>SQRTD(857)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	SQRTD(857)	S	R	計算指定的雙精度浮點資料 (64 位元) 之平方根, 並把結果輸出到結果 words。	輸出 需要
SQRTD(857)						
S						
R						
DOUBLE EXPONENT EXPD @EXPD 858	<table border="1"> <tr><td>EXPD(858)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	EXPD(858)	S	R	計算指定的雙精度浮點資料 (64 位元) 的自然指數 (以 e 為底), 並把結果輸出到結果 words。	輸出 需要
EXPD(858)						
S						
R						
DOUBLE LOGARITHM LOGD @LOGD 859	<table border="1"> <tr><td>LOGD(859)</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>S:第一個來源 words R:第一個結果 words</p>	LOGD(859)	S	R	計算指定的雙精度浮點資料 (64 位元) 的自然對數 (以 e 為底), 並把結果輸出到結果 words。	輸出 需要
LOGD(859)						
S						
R						

指令 助億碼	符號 / 運算元	功能	位置 執行條件
DOUBLE EXPONENTIAL POWER PWRD @PWRD 860	<p>B:第一個底數 words E:第一個指數 words R:第一個結果 words</p>	把一個雙精度浮點資料 (64 位元) 作為另一個雙精度浮點資料(64 位元) 的指數, 並把結果輸出到結果 words。	輸出 需要
DOUBLE SYMBOL COMPARISON LD, AND, OR 或 + =D (335), <>D (336), <D (337), <=D (338), >D (339), 或 >=D (340)	<p>使用 LD:</p> <p>使用 AND:</p> <p>使用 OR:</p> <p>S1: 比較資料 1 S2: 比較資料 2</p>	比較指定的雙精度浮點資料 (64 位元), 若比較結果為成立時則建立一個 ON 執行條件。 有三種符號可用於浮點數符號比較指令: LD (載入), AND, 與 OR。	LD : 不需要 AND 或 OR : 需要

8-15 表格資料處理指令

指令	符號 / 運算元	功能	位置 執行條件
SET STACK SSET @SSET 630	 <p>TB: 第一個堆疊位址 N: words數</p>	在指定的words上定義一指定長度的堆疊，並將資料範圍內的words均初始化為零。 PLC 記憶體位址 	輸出需要
PUSH ONTO STACK PUSH @PUSH 632	 <p>TB: 第一個堆疊位址 S: 來源words</p>	把一個words的資料寫入堆疊。 PLC 記憶體位址 	輸出需要
LAST IN FIRST OUT LIFO @LIFO 634	 <p>TB: 第一個堆疊位址 D: 目的words</p>	把最後寫入指定堆疊的資料 (堆疊中最新的資料) words讀出來。 堆疊指標 PLC 記憶體位址 	輸出需要
FIRST IN FIRST OUT FIFO @FIFO 633	 <p>TB: 第一個堆疊位址 D: 目的words</p>	把最先寫入指定堆疊的資料 (堆疊中最舊的資料) words讀出來。 堆疊指標 PLC 記憶體位址 	輸出需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件					
DIMENSION RECORD TABLE DIM @DIM 631	<table border="1"> <tr><td>DIM(631)</td></tr> <tr><td>N</td></tr> <tr><td>LR</td></tr> <tr><td>NR</td></tr> <tr><td>TB</td></tr> </table> <p>N: 表格編號 LR: 每筆記錄的長度 NR: 記錄編號 TB: 表格的第一個words</p>	DIM(631)	N	LR	NR	TB	<p>藉由宣告每筆記錄的長度與記錄數來定義記錄表格。最多可以定義 16 個記錄表格。</p>	輸出 需要
DIM(631)								
N								
LR								
NR								
TB								
SET RECORD LOCATION SETR @SETR 635	<table border="1"> <tr><td>SETR(635)</td></tr> <tr><td>N</td></tr> <tr><td>R</td></tr> <tr><td>D</td></tr> </table> <p>N: 表格編號 R: 記錄編號 D: 目標索引暫存器</p>	SETR(635)	N	R	D	<p>把指定記錄 (記錄開頭的 PLC 記憶體位址) 之位置寫入指定的索引暫存器中。</p> <p>SETR(635) 把記錄 R 的第一個字組之 PLC 記憶體位址 (m) 寫入索引暫存器 D。</p>	輸出 需要	
SETR(635)								
N								
R								
D								
GET RECORD NUMBER GETR @GETR 636	<table border="1"> <tr><td>GETR(636)</td></tr> <tr><td>N</td></tr> <tr><td>IR</td></tr> <tr><td>D</td></tr> </table> <p>N: 表格編號 IR: 索引暫存器 D: 目標words</p>	GETR(636)	N	IR	D	<p>傳回包含於指定的索引暫存器中之 PLC 記憶體位址上的記錄之記錄編號。</p> <p>GETR(636) 把包含 I/O 記憶體位址 (m) 的記錄之記錄編號寫入到 D。</p>	輸出 需要	
GETR(636)								
N								
IR								
D								
DATA SEARCH SRCH @SRCH 181	<table border="1"> <tr><td>SRCH(181)</td></tr> <tr><td>C</td></tr> <tr><td>R1</td></tr> <tr><td>Cd</td></tr> </table> <p>C: 第一個控制 words R1: 範圍內的第一個words Cd: 比較資料</p>	SRCH(181)	C	R1	Cd	<p>在某範圍的words內搜尋一資料words。</p> <p>搜尋</p> <p>相同 → IR00</p>	輸出 需要	
SRCH(181)								
C								
R1								
Cd								

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
SWAP BYTES SWAP @SWAP 637	<table border="1"> <tr><td>SWAP(637)</td></tr> <tr><td>N</td></tr> <tr><td>R1</td></tr> </table> <p>N: words數 R1: 範圍內的 第一個words</p>	SWAP(637)	N	R1	把一個範圍內所有words的最左位元組與最右位元組交換。 高位元與低位元資料互換。 	輸出 需要	
SWAP(637)							
N							
R1							
FIND MAXIMUM MAX @MAX 182	<table border="1"> <tr><td>MAX(182)</td></tr> <tr><td>C</td></tr> <tr><td>R1</td></tr> <tr><td>D</td></tr> </table> <p>C: 第一個控制 words R1: 範圍內的 第一個words D: 目標words</p>	MAX(182)	C	R1	D	尋找一範圍內之最大值。 PLC 記憶體位址 	輸出 需要
MAX(182)							
C							
R1							
D							
FIND MINIMUM MIN @MIN 183	<table border="1"> <tr><td>MIN(183)</td></tr> <tr><td>C</td></tr> <tr><td>R1</td></tr> <tr><td>D</td></tr> </table> <p>C: 第一個控制 words R1: 範圍內的 第一個words D: 目標words</p>	MIN(183)	C	R1	D	尋找一範圍內之最小值。 PLC 記憶體位址 	輸出 需要
MIN(183)							
C							
R1							
D							
SUM SUM @SUM 184	<table border="1"> <tr><td>SUM(184)</td></tr> <tr><td>C</td></tr> <tr><td>R1</td></tr> <tr><td>D</td></tr> </table> <p>C: 第一個控制 words R1: 範圍內的 第一個words D: 第一個目標 words</p>	SUM(184)	C	R1	D	把範圍內的位元組或words相加並把結果輸出到兩個words。 	輸出 需要
SUM(184)							
C							
R1							
D							
FRAME CHECKSUM FCS @FCS 180	<table border="1"> <tr><td>FCS(180)</td></tr> <tr><td>C</td></tr> <tr><td>R1</td></tr> <tr><td>D</td></tr> </table> <p>C: 第一個控制 words R1: 範圍內的 第一個words D: 第一個目標 words</p>	FCS(180)	C	R1	D	計算指定範圍的FCS值，再以ASCII方式輸出。 	輸出 需要
FCS(180)							
C							
R1							
D							

指令 助憶碼	符號 / 運算元	功能	位置 執行條件				
STACK SIZE READ (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) SNUM @SNUM 638	<table border="1"> <tr><td>SNUM(638)</td></tr> <tr><td>TB</td></tr> <tr><td>D</td></tr> </table> TB: 第一個堆疊 位址 D: 目標words	SNUM(638)	TB	D	計算指定之堆疊中堆疊資料的量 (words 數)。	輸出需要	
SNUM(638)							
TB							
D							
STACK DATA READ (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) SREAD @SREAD 639	<table border="1"> <tr><td>SREAD(639)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> TB: 第一個堆疊 位址 C: 偏量值 D: 目標words	SREAD(639)	TB	C	D	由堆疊中的指定資料元素讀出資料。偏量值表示所要的資料元素之位置 (在現行指標位置之前多少個資料元素)。	輸出需要
SREAD(639)							
TB							
C							
D							
STACK DATA OVERWRITE (僅 CS1-H, CJ1-H; CJ1M, 或 CS1D) SWRIT @SWRIT 640	<table border="1"> <tr><td>SWRIT(640)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> TB: 第一個堆疊 位址 C: 偏量值 S: 來源資料	SWRIT(640)	TB	C	S	把來源資料寫入堆疊中的指定資料元素 (覆蓋現有的資料)。偏量值表示所要的資料元素之位置 (在現行指標位置之前多少個資料元素)。	輸出需要
SWRIT(640)							
TB							
C							
S							
STACKDATAINSE RT (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) SINS @SINS 641	<table border="1"> <tr><td>SINS(641)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> TB: 第一個堆疊 位址 C: 偏量值 S: 來源資料	SINS(641)	TB	C	S	把來源資料插入到堆疊中的指定位置並把堆疊中剩下的資料往下移位。偏量值表示所要的資料元素之位置 (在現行指標位置之前多少個資料元素)。	輸出需要
SINS(641)							
TB							
C							
S							
STACKDATADEL ETE (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) SDEL @SDEL 642	<table border="1"> <tr><td>SDEL(642)</td></tr> <tr><td>TB</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> TB: 第一個堆疊 位址 C: 偏量值 D: 目標words	SDEL(642)	TB	C	D	刪除堆疊中指定位置的資料元素並把堆疊中剩下的資料往上移位。偏量值表示所要的資料元素之位置 (在現行指標位置之前多少個資料元素)。	輸出需要
SDEL(642)							
TB							
C							
D							

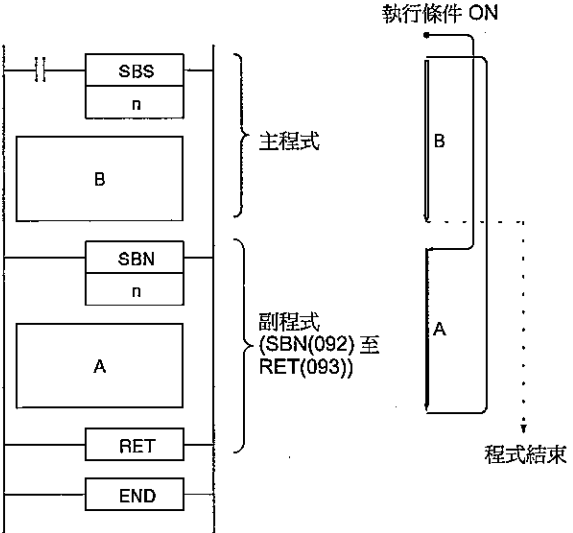
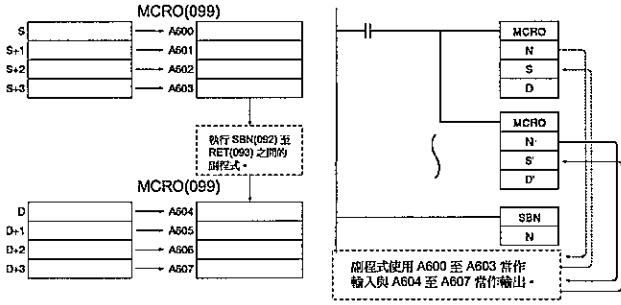
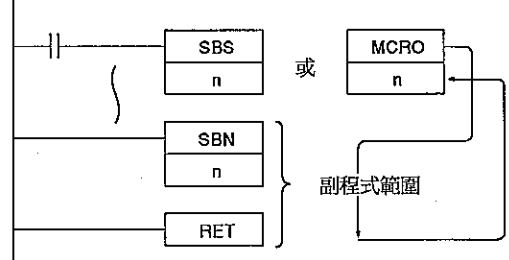
8-16 資料控制指令

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
PID CONTROL PID 190	<table border="1"> <tr><td>PID(190)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: PV值輸入CH的編號 C: 設定PID參數的CH編號 D: MV值輸出CH的編號</p>	PID(190)	S	C	D	<p>依據指定的參數執行 PID 控制。</p>	輸出 需要
PID(190)							
S							
C							
D							
PID CONTROL WITH AUTO TUNING (僅 CS1-H, CJ1- H, CJ1M, 或 CS1D) PIDAT 191	<table border="1"> <tr><td>PIDAT(191)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: PV值輸入CH的 編號 C: 設定PID參數的 CH編號 D: MV值輸出CH的 編號</p>	PIDAT(191)	S	C	D	<p>依據指定的參數執行 PID 控制。以 PIDAT(191) 可以自動調諧 Auto tuning PID 參數。</p>	輸出 需要
PIDAT(191)							
S							
C							
D							
LIMIT CONTROL LMT @LMT 680	<table border="1"> <tr><td>LMT(680)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: 輸入的CH編號 C: 上下限值的CH 編號 D: 輸出的CH編號</p>	LMT(680)	S	C	D	<p>依據輸入資料是否在上下限之間而控制輸出的資料。</p>	輸出 需要
LMT(680)							
S							
C							
D							
DEAD BAND CONTROL BAND @BAND 681	<table border="1"> <tr><td>BAND(681)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: 輸入的CH編號 C: 上下限值的CH 編號 D: 輸出的CH編號</p>	BAND(681)	S	C	D	<p>依據輸入資料是否在無效區 DEAD BAND 段範圍內而決定輸出資料。</p>	輸出 需要
BAND(681)							
S							
C							
D							

指令	符號 / 運算元	功能	位置 執行條件				
DEAD ZONE CONTROL ZONE @ZONE 682	<table border="1"> <tr><td>ZONE(682)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>S: 輸入 words 的 CH 編號 C: 第一個限制 words bios 資料的 CH 編號 D: 輸出 words 的 CH 編號</p>	ZONE(682)	S	C	D	<p>把輸入資料加上指定的偏壓並輸出到結果。</p>	輸出需要
ZONE(682)							
S							
C							
D							
SCALING SCL @SCL 194	<table border="1"> <tr><td>SCL(194)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: 被轉換的 CH 編號 P1: 設定參數的 CH 編號 R: 轉換值輸出的 CH 編號</p>	SCL(194)	S	P1	R	<p>依據指定的線性函數把無符號 BIN 資料轉換為無符號 BCD 資料。</p>	輸出需要
SCL(194)							
S							
P1							
R							
SCALING 2 SCL2 @SCL2 486	<table border="1"> <tr><td>SCL2(486)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: 被轉換的 CH 編號 P1: 設定參數的 CH 編號 R: 轉換值輸出 CH 編號</p>	SCL2(486)	S	P1	R	<p>依據指定的線性函數把帶符號 BIN 資料轉換為帶符號 BCD 資料。在定義線性函數時可以輸入一偏量。</p>	輸出需要
SCL2(486)							
S							
P1							
R							

指令 助憶碼	符號 / 運算元	功能	位置 執行條件				
SCALING 3 SCL3 @SCL3 487	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>SCL3(487)</td></tr> <tr><td>S</td></tr> <tr><td>P1</td></tr> <tr><td>R</td></tr> </table> <p>S: 被轉換的CH編號 P1: 設定參數的CH編號 R: 轉換值輸出的CH編號</p>	SCL3(487)	S	P1	R	<p>依據指定的線性函數把帶符號BCD資料轉換為帶符號BIN資料。 在定義線性函數時可以輸入一偏量。</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>正偏量</p> </div> <div style="text-align: center;"> <p>負偏量</p> </div> </div> <div style="text-align: center; margin-top: 20px;"> <p>0000 的偏量</p> </div>	<p>輸出 需要</p>
SCL3(487)							
S							
P1							
R							
AVERAGE AVG 195	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>AVG(195)</td></tr> <tr><td>S</td></tr> <tr><td>N</td></tr> <tr><td>R</td></tr> </table> <p>S: 現在值輸入的CH編號 N: 平均值計算cycle數 R: 平均值的CH編號</p>	AVG(195)	S	N	R	<p>計算特定循環數的輸入words之平均值。</p> <div style="margin-left: 40px;"> <p>S: 來源words</p> <p>N: 循環數</p> <p>R</p> <p>R + 1 指標</p> <p>平均合法旗標</p> <p>R + 2</p> <p>R + 3</p> <p>...</p> <p>R + N + 1</p> <p>N 個</p> <p>平均值</p> </div>	<p>輸出 需要</p>
AVG(195)							
S							
N							
R							

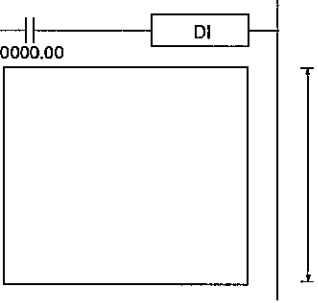
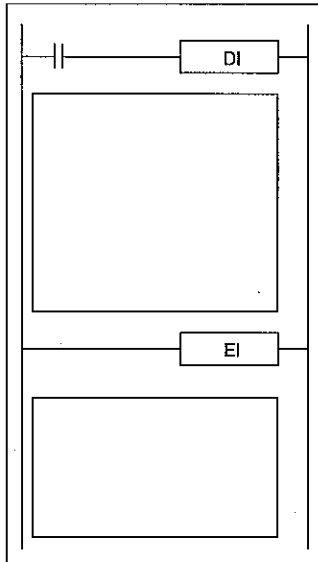
8-17 副程式指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
<p>SUBROUTINE CALL</p> <p>SBS @SBS 091</p>	<p>SBS(091) N</p> <p>N: 副程式編號</p>	<p>以指定的副程式編號來呼叫副程式並執行該程式。</p> 	<p>輸出 需要</p>
<p>MACRO</p> <p>MCRO @MCRO 099</p>	<p>MCRO(099) N S D</p> <p>N: 副程式編號 S: 第一個輸入 參數words D: 第一個輸出 參數words</p>	<p>以指定的副程式編號來呼叫副程式並使用 S 至 S+3 中的輸入參數及 D 至 D+3 中的輸出參數來執行該程式。</p> 	<p>輸出 需要</p>
<p>SUBROUTINE ENTRY</p> <p>SBN 092</p>	<p>SBN(092) N</p> <p>N: 副程式編號</p>	<p>表示編號為指定值之副程式的開頭。</p> 	<p>輸出 不需要</p>
<p>SUBROUTINE RETURN</p> <p>RET 093</p>	<p>RET(093)</p>	<p>表示副程式的結束。</p>	<p>輸出 不需要</p>

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
GLOBAL SUBROUTINE CALL (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) GSBS 750	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GSBS(750)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 40px;">N</div> N: 副程式編號	以指定的副程式編號呼叫副程式並執行該程式。	輸出 不需要
GLOBAL SUBROUTINE ENTRY (僅 CS1-H, CJ1-H, CJ1M; 或 CS1D) GSBN 751	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GSBN(751)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 40px;">N</div> N: 副程式編號	表示以指定值為編號之副程式程式的開頭。	輸出 不需要
GLOBAL SUBROUTINE RETURN (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) GRET 752	<div style="border: 1px solid black; padding: 2px; display: inline-block;">GRET(752)</div>	表示副程式的結束。	輸出 不需要

8-18 中斷控制指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
<p>SET INTERRUPT MASK (CS1D 不支援)</p> <p>MSKS @MSKS 690</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> MSKS(690) N S </div> <p>N: 控制資料1 S: 控制資料2</p>	<p>設定 I/O 中斷或定時中斷之中斷程序。I/O 中斷工作單與定時中斷工作單在 PC 第一次開啓時都是被除能的。可以用 MSKS(690)來解除或加於 I/O 中斷與設定定時中斷的時間間隔。(I/O 中斷不支援 CJ1 CPU 模組。)</p>	<p>輸出 需要</p>
<p>READ INTERRUPT MASK (CS1D 不支援)</p> <p>MSKR @MSKR 692</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> MSKR(692) N D </div> <p>N: 控制資料 D: 輸出的CH編號</p>	<p>讀取 MSKS(690) 所設定之現行的中斷處理設定。</p>	<p>輸出 需要</p>
<p>CLEAR INTERRUPT (CS1D 不支援)</p> <p>CLI @CLI 691</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> CLI(691) N S </div> <p>N: 控制資料1 S: 控制資料2</p>	<p>清除或維持所記錄的 I/O 中斷之中斷輸入，或設定定時中斷之到第一次定時中斷的時間。 N = 0 至 3 (I/O 中斷不支援 CJ1 CPU 模組。)</p> <p>N = 4 至 5</p>	<p>輸出 需要</p>

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
DISABLE INTERRUPTS DI @DI 693	DI(693)	把除了電源 OFF 中斷之外的其他中斷工作單之執行均除能。  <p>把所有的中斷工作單除能 (除了電源 OFF 中斷之外)。</p>	輸出 需要
ENABLE INTERRUPTS EI 694	EI(694)	把被 DI(693) 所除能之中斷工作單之執行全部致能。  <p>把所有的中斷工作單除能 (除了電源 OFF 中斷之外)。</p> <p>把所有被除能之中斷工 作單之執行全部致能。</p>	輸出 不需要

8-19 高速計數器與脈衝輸出指令 (僅 CJ1M-CPU22/23)

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
MODE CONTROL INI @INI 880	<table border="1"> <tr><td>INI</td></tr> <tr><td>P</td></tr> <tr><td>C</td></tr> <tr><td>NV</td></tr> </table> <p>P: 指定輸出埠 C: 控制資料 NV: 新 PV 的第一個 words</p>	INI	P	C	NV	INI(880) 用來開始與停止目標值比較、改變高速計數器的顯示值 (PV)、改變中斷輸入 (計數器模式) 的 PV、改變脈衝輸出的 PV、或停止脈衝輸出。	輸出 需要
INI							
P							
C							
NV							
HIGH-SPEED PV READ PRV @PRV 881	<table border="1"> <tr><td>PRV</td></tr> <tr><td>P</td></tr> <tr><td>C</td></tr> <tr><td>D</td></tr> </table> <p>P: 指定輸出埠 C: 控制資料 D: 輸出現在值的 CH 編號</p>	PRV	P	C	D	PRV(881) 用來讀取高速計數器、脈衝輸出、或中斷輸入 (計數器模式) 的顯示值 (PV)。	輸出 需要
PRV							
P							
C							
D							
COMPARISON TABLE LOAD CTBL @CTBL 882	<table border="1"> <tr><td>CTBL</td></tr> <tr><td>P</td></tr> <tr><td>C</td></tr> <tr><td>TB</td></tr> </table> <p>P: 指定輸出埠 C: 控制資料 TB: 比較表第一個 words</p>	CTBL	P	C	TB	CTBL(882) 用於進行高速計數器的顯示值 (PV) 之目標值或範圍比較。	輸出 需要
CTBL							
P							
C							
TB							
SPEED OUTPUT SPED @SPED 885	<table border="1"> <tr><td>SPED</td></tr> <tr><td>P</td></tr> <tr><td>M</td></tr> <tr><td>F</td></tr> </table> <p>P: 指定輸出埠 M: 輸出模式 F: 第一個脈衝頻率 words</p>	SPED	P	M	F	SPED(885) 用來指定頻率與進行脈衝輸出而不會加速或減速。	輸出 需要
SPED							
P							
M							
F							
SET PULSES PULS @PULS 886	<table border="1"> <tr><td>PULS</td></tr> <tr><td>P</td></tr> <tr><td>T</td></tr> <tr><td>N</td></tr> </table> <p>P: 指定輸出埠 T: 脈衝型式 N: 設定脈衝數的 CH 編號</p>	PULS	P	T	N	PULS(886) 用於設定脈衝輸出的脈衝數。	輸出 需要
PULS							
P							
T							
N							

指令 助憶碼	符號 / 運算元	功能	位置 執行條件					
PULSE OUTPUT PLS2 @PLS2 887	<table border="1"> <tr><td>PLS2</td></tr> <tr><td>P</td></tr> <tr><td>M</td></tr> <tr><td>S</td></tr> <tr><td>F</td></tr> </table> <p>P: 指定輸出埠 M: 輸出模式 S: 設定表第一個 words F: 起動頻率第一個 words</p>	PLS2	P	M	S	F	PLS2(887) 用來設定脈衝頻率與加速/減速率,並以加速/減速進行脈衝輸出(以不同的加速/減速率)。只能用於定位。	輸出需要
PLS2								
P								
M								
S								
F								
ACCELERATION CONTROL ACC @ACC 888	<table border="1"> <tr><td>ACC</td></tr> <tr><td>P</td></tr> <tr><td>M</td></tr> <tr><td>S</td></tr> </table> <p>P: 指定輸出埠 M: 輸出模式 S: 設定表的 CH 編號</p>	ACC	P	M	S	ACC(888) 用來設定脈衝頻率與加速/減速率,並以加速/減速進行脈衝輸出(以相同的加速/減速率)。可用於定位與速度控制。	輸出需要	
ACC								
P								
M								
S								
ORIGIN SEARCH ORG @ORG 889	<table border="1"> <tr><td>ORG</td></tr> <tr><td>P</td></tr> <tr><td>C</td></tr> </table> <p>P: 指定輸出埠 C: 控制資料</p>	ORG	P	C	ORG(889) 用來執行原點搜尋與返回。	輸出需要		
ORG								
P								
C								
PULSE WITH VARIABLE DUTY FACTOR PWM @ 891	<table border="1"> <tr><td>PWM</td></tr> <tr><td>P</td></tr> <tr><td>F</td></tr> <tr><td>D</td></tr> </table> <p>P: 指定輸出埠 F: 設定脈衝頻率 D: 設定 duty</p>	PWM	P	F	D	PWM(891) 用於以可變的功率因數輸出脈衝。	輸出需要	
PWM								
P								
F								
D								

8-20 步進指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件		
STEP DEFINE STEP 008	<table border="1"> <tr><td>STEP(008)</td></tr> <tr><td>B</td></tr> </table> <p>B: 位元</p>	STEP(008)	B	STEP(008) 依其位置或控制位元是否指定而有以下 2 種動作方式: (1) 開始一指定的步進。 (2) 結束步進規劃區域 (即步進執行)。	輸出需要
STEP(008)					
B					
STEP START SNXT 009	<table border="1"> <tr><td>SNXT(009)</td></tr> <tr><td>B</td></tr> </table> <p>B: 位元</p>	SNXT(009)	B	SNXT(009) 用於以下的三種方式: (1) 開始步進規劃執行。 (2) 繼續下一個步進控制位元。 (3) 結束步進規劃執行。	輸出需要
SNXT(009)					
B					

8-21 基本 I/O 模組指令

指令 助憶碼	符號 / 運算元	功能	定位 執行條件
I/O REFRESH IORF @IORF 097	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> IORF(097) St E </div> <p>St: 更新低四位元的CH編號 E: 更新高四位元的CH編號</p>	<p>更新指定的 I/O words。</p> <p>I/O 位元區域或 特殊 I/O 模組位元區域</p>	輸出 需要
7-SEGMENT DECODER SDEC @SDEC 078	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> SDEC(078) S Di D </div> <p>S: 來源 words Di: 位元指定器 D: 第一個目標 words</p>	<p>把指定的位數之Hex內容轉換為 8 位元、7 段顯示碼並把它放入指定的目的 words 之上 8 位元或下 8 位元。</p>	輸出 需要
INTELLIGENT I/O IORD @IORD 222	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> IORD(222) C S D </div> <p>C: 控制資料 S: 轉換來源與 words 數 D: 傳送CH編號</p>	<p>讀取 I/O 模組的記憶體區域之內容。</p>	輸出 需要

指令 助億碼	符號 / 運算元	功能	定位 執行條件				
INTELLIGENT I/O WRITE IOWR @IOWR 223	<table border="1"> <tr><td>IOWR(223)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制資料 S: 轉換來源與 words 數 D: 轉換標的與 words 數</p>	IOWR(223)	C	S	D	<p>把 CPU 模組之 I/O 記憶體區域的內容輸出到特殊 I/O 模組。</p>	輸出需要
IOWR(223)							
C							
S							
D							
CPU BUS UNIT I/O REFRESH (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) DLNK @DLNK 226	<table border="1"> <tr><td>DLNK(226)</td></tr> <tr><td>N</td></tr> </table> <p>N: 模組編號</p>	DLNK(226)	N	<p>立即以指定的模組編號更新 CPU 匯流排模組中的 I/O。</p>	輸出需要		
DLNK(226)							
N							

8-22 串列通訊指令

指令 助億碼	符號 / 運算元	功能	位置 執行條件					
PROTOCOL MACRO PMCR @PMCR 260	<table border="1"> <tr><td>PMCR(260)</td></tr> <tr><td>C1</td></tr> <tr><td>C2</td></tr> <tr><td>S</td></tr> <tr><td>R</td></tr> </table> <p>C1: 控制資料通埠的編號及設定外部裝置的機號 C2: 控制資料傳送與接收序列數的NO S: 傳送資料的起始CH編號 R: 接收資料的起始CH編號</p>	PMCR(260)	C1	C2	S	R	<p>呼叫並執行設定於串列通訊卡 (僅 CS 系列) 或通信模組中的通訊序列。</p>	輸出需要
PMCR(260)								
C1								
C2								
S								
R								
TRANSMIT TXD @TXD 236	<table border="1"> <tr><td>TXD(236)</td></tr> <tr><td>S</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>S: 接收資料的起始 CH 編號 C: 控制 words N: 接收的 Byte 數 (BCD 四個 words) 0000 至 0256</p>	TXD(236)	S	C	N	<p>由內藏於 CPU 模組的 RS-232C 埠輸出指定之位元組數的資料。</p>	輸出需要	
TXD(236)								
S								
C								
N								

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
RECEIVE RXD @RXD 235	<table border="1"> <tr><td>RXD(235)</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> <tr><td>N</td></tr> </table> <p>D: 接收資料的起始 CH 編號 C: 控制 words N: 位元組數 0000 至 0100 Hex (0 至 256 BCD)</p>	RXD(235)	D	C	N	由內藏於 CPU 模組的 RS-232C 埠讀出指定之位元組數的資料。	輸出 需要
RXD(235)							
D							
C							
N							
CHANGE SERIAL PORT SETUP STUP @STUP 237	<table border="1"> <tr><td>STUP(237)</td></tr> <tr><td>C</td></tr> <tr><td>S</td></tr> </table> <p>C: 埠的指定 S: 設定資料的起始 CH 編號</p>	STUP(237)	C	S	改變 CPU 模組、串列通訊 模 組(CPU 匯流排模組)、或串列通訊卡 (僅 CS 系列) 上之串列埠的通訊參數。因而 STUP(237) 可以在 PLC 操作期間改變通訊協定模式。	輸出 需要	
STUP(237)							
C							
S							

8-23 網路指令

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
NETWORK SEND SEND @SEND 090	<table border="1"> <tr><td>SEND(090)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>S: 傳送資料 CH 編號(本地) D: 接收資料 CH 編號(目的地) C: 控制資料的低四位元的 CH 編號</p>	SEND(090)	S	D	C	<p>傳送資料到網路中的節點。</p> <p>本地節點</p> <p>目的節點</p> <p>n: 傳送字組數</p>	輸出 需要
SEND(090)							
S							
D							
C							

指令 助憶碼	符號 / 運算元	功能	位置 執行條件				
<p>NETWORK RECEIVE</p> <p>RECV @RECV 098</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>RECVD(098)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>S: 傳送資料CH編號 (目的) D: 接收資料CH編號 (本地) C: 控制資料的低四位 元的CH編號</p>	RECVD(098)	S	D	C	<p>由網路中的節點請求傳輸的資料並接收該資料。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>本地節點</p> </div> <div style="text-align: center;"> <p>←</p> </div> <div style="text-align: center;"> <p>來源節點</p> </div> </div>	<p>輸出 需要</p>
RECVD(098)							
S							
D							
C							
<p>DELIVER COMMAND</p> <p>CMND @CMND 490</p>	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>CMND(490)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> <tr><td>C</td></tr> </table> <p>S: 設定指令語的 起始CH編號 D: 接收反應資料 的起始CH編號 C: 控制資料的低四 位元的CH編號</p>	CMND(490)	S	D	C	<p>送出 FINS 指令並接收反應。</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>本地節點</p> </div> <div style="text-align: center;"> <p>指令語 →</p> </div> <div style="text-align: center;"> <p>目的節點</p> </div> <div style="text-align: center;"> <p>← 反應</p> </div> </div>	<p>輸出 需要</p>
CMND(490)							
S							
D							
C							

8-24 檔案記憶體指令

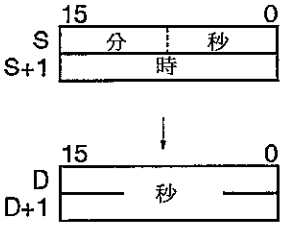
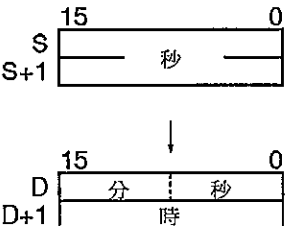
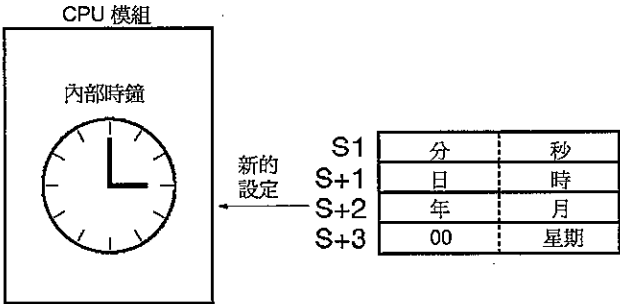
指令	符號 / 運算元	功能	位置 執行條件					
READ DATA FILE FREAD @FREAD 700	<table border="1"> <tr><td>FREAD(700)</td></tr> <tr><td>C</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>C: 控制words S1: 第一個來源 words S2: 檔名 D: 第一個目的 words</p>	FREAD(700)	C	S1	S2	D	<p>由指定的資料檔讀入指定的資料或指定量的資料到 CPU 模組中的指定資料區域內。</p> <p>指定於 S1+2 與 S1+3 中的開始讀出位址</p> <p>S2 中指定的檔案</p> <p>CPU 模組</p> <p>指定於 S1 與 S1+1 的 words 數</p> <p>要寫入 D 與 D+1 的 words 數。</p> <p>記憶卡或 EM 檔案記憶體 (由 C 的第四個位元指定)</p> <p>words 數</p> <p>記憶卡或 EM 檔案記憶體 (由 C 的第四個位元指定)</p>	輸出需要
FREAD(700)								
C								
S1								
S2								
D								
WRITE DATA FILE FWRIT @FWRIT 701	<table border="1"> <tr><td>FWRIT(701)</td></tr> <tr><td>C</td></tr> <tr><td>D1</td></tr> <tr><td>D2</td></tr> <tr><td>S</td></tr> </table> <p>C: 控制words D1: 第一個目的 words D2: 檔名 S: 第一個來源 words</p>	FWRIT(701)	C	D1	D2	S	<p>把 CPU 模組中資料區域的指定資料附加於或覆寫於檔案記憶體中指定資料檔的資料之後或之上。若指定的檔案不存在，將會以該檔名建立一個新的檔案。</p> <p>CPU 模組</p> <p>指定於 D1+2 與 D1+3 的開始 words</p> <p>D2 中指定的檔案</p> <p>S 中指定的開始位址</p> <p>指定於 D1 與 D1+1 的 words 數</p> <p>覆寫</p> <p>記憶卡或 EM 檔案記憶體 (由 C 的第四個位元指定)</p> <p>CPU 模組</p> <p>S 中指定的開始位址</p> <p>檔案結尾</p> <p>指定於 D1 與 D1+1 的 words 數</p> <p>D2 中指定的檔案</p> <p>既存資料</p> <p>附加</p> <p>記憶卡或 EM 檔案記憶體 (由 C 的第四個位元指定)</p> <p>CPU 模組</p> <p>S 中指定的開始位址</p> <p>檔案開頭</p> <p>指定於 D1 與 D1+1 的 words 數</p> <p>D2 中指定的檔案</p> <p>建立的新檔案</p> <p>記憶卡或 EM 檔案記憶體 (由 C 的第四個位元指定)</p>	輸出需要
FWRIT(701)								
C								
D1								
D2								
S								

8-25 顯示指令

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
DISPLAY MESSAGE MSG @MSG 046	MSG(046) N M N: 訊息編號 M: 接收訊息的CH 編號	讀出指定之延伸ASCII 的十六個words 並把此訊息顯示於諸如程式書寫器等之週邊裝置上。	輸出 需要

8-26 時鐘指令

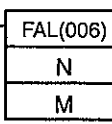
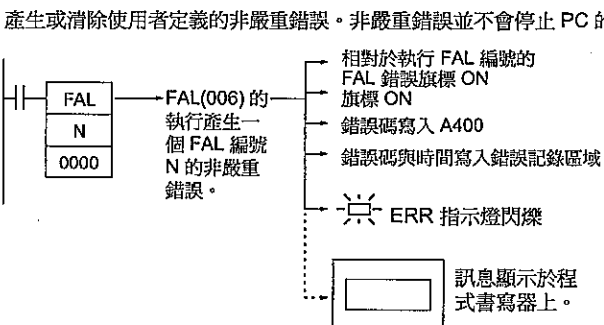
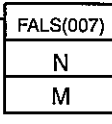
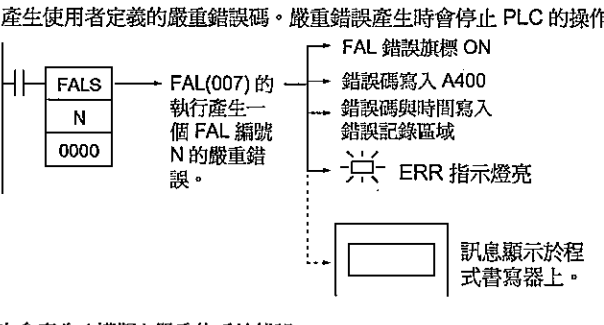
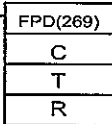
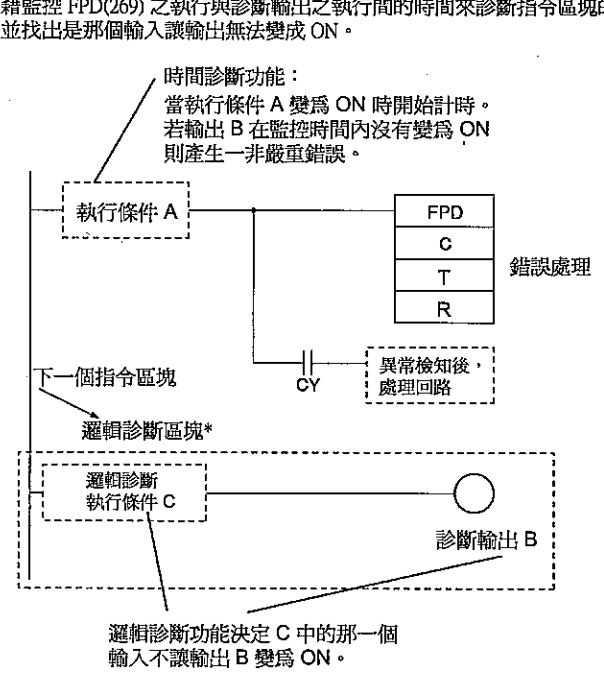
指令 助憶碼	符號 / 運算元	功能	位置 執行條件
CALENDAR ADD CADD @CADD 730	C T R C: 第一個日曆 words T: 第一個時間 words R: 第一個結果 words	把時間加到指定words 中的日曆資料。 $ \begin{array}{r} \begin{array}{ccc} 15 & 87 & 0 \\ \hline C & \text{分} & \text{秒} \\ C+1 & \text{日} & \text{時} \\ C+2 & \text{年} & \text{月} \end{array} \\ + \\ \begin{array}{ccc} 15 & 87 & 0 \\ \hline T & \text{分} & \text{秒} \\ T+1 & \text{時} & \end{array} \\ \downarrow \\ \begin{array}{ccc} 15 & 87 & 0 \\ \hline R & \text{分} & \text{秒} \\ R+1 & \text{日} & \text{時} \\ R+2 & \text{年} & \text{月} \end{array} \end{array} $	輸出 需要
CALENDAR SUBTRACT CSUB @CSUB 731	T R C: 第一個日曆 words T: 第一個時間 words R: 第一個結果 words	把指定words 中的日曆資料減去某個時間。 $ \begin{array}{r} \begin{array}{ccc} 15 & 87 & 0 \\ \hline C & \text{分} & \text{秒} \\ C+1 & \text{日} & \text{時} \\ C+2 & \text{年} & \text{月} \end{array} \\ - \\ \begin{array}{ccc} 15 & 87 & 0 \\ \hline T & \text{分} & \text{秒} \\ T+1 & \text{時} & \end{array} \\ \downarrow \\ \begin{array}{ccc} 15 & 87 & 0 \\ \hline R & \text{分} & \text{秒} \\ R+1 & \text{日} & \text{時} \\ R+2 & \text{年} & \text{月} \end{array} \end{array} $	輸出 需要

指令 助億碼	符號 / 運算元	功能	位置 執行條件
HOURS TO SECONDS SEC @SEC 065	SEC(065) S D S:第一個來源 words D:第一個目標 words	把時/分/秒格式表示的時間資料轉換為只有以秒表示。 	輸出 需要
SECONDS TO HOURS HMS @HMS 066	HMS(066) S D S:第一個來源 words D:第一個目標 words	把以秒表示的時間資料轉換為時/分/秒的格式。 	輸出 需要
CLOCK ADJUSTMENT DATE @DATE 735	DATE(735) S S:第一個來源 words	把內部時鐘的設定改為指定之來源 words 的設定。 	輸出 需要

8-27 除錯指令

指令 助億碼	符號 / 運算元	功能	位置 執行條件
TRACE MEMORY SAMPLING TRSM 045	TRSM(045)	當 TRSM(045) 執行時，會取樣一預先選定之位元或 words 的狀態並將其儲存於 Trace 記憶體。TRSM(045) 可以在任何時間使用於程式中的任何地方。	輸出 不需要

8-28 故障診斷指令

指令	符號 / 運算元	功能	位置 執行條件
助憶碼 FAILURE ALARM FAL @FAL 006	 <p>N: FAL 編號 M: 要產生之訊息 words 或錯誤碼的第一位位置 (#0000 至 #FFFF)</p>	<p>產生或消除使用者定義的非嚴重錯誤。非嚴重錯誤並不會停止 PC 的操作。</p>  <p>FAL(006) 的執行產生一個 FAL 編號 N 的非嚴重錯誤。</p> <ul style="list-style-type: none"> 相對於執行 FAL 編號的 FAL 錯誤旗標 ON 錯誤碼寫入 A400 錯誤碼與時間寫入錯誤記錄區域 ERR 指示燈閃爍 訊息顯示於程式書寫器上。 <p>也會產生 (模擬) 嚴重的系統錯誤。</p>	輸出需要
SEVERE FAILURE ALARM FALS 007	 <p>N: FALS 編號 M: 要產生之訊息 words 或錯誤碼的第一位位置 (#0000 至 #FFFF)</p>	<p>產生使用者定義的嚴重錯誤碼。嚴重錯誤產生時會停止 PLC 的操作。</p>  <p>FALS(007) 的執行產生一個 FAL 編號 N 的嚴重錯誤。</p> <ul style="list-style-type: none"> FAL 錯誤旗標 ON 錯誤碼寫入 A400 錯誤碼與時間寫入錯誤記錄區域 ERR 指示燈亮 訊息顯示於程式書寫器上。 <p>也會產生 (模擬) 嚴重的系統錯誤。</p>	輸出需要
FAILURE POINT DETECTION FPD 269	 <p>C: 控制 words T: 異常監視時間的設定值 R: 第一個註記 words</p>	<p>藉監控 FPD(269) 之執行與診斷輸出之執行間的時間來診斷指令區塊的錯誤，並找出是哪個輸入該輸出無法變成 ON。</p> <p>時間診斷功能： 當執行條件 A 變為 ON 時開始計時。 若輸出 B 在監控時間內沒有變為 ON 則產生一非嚴重錯誤。</p>  <p>錯誤處理</p> <p>異常檢知後，處理回路</p> <p>邏輯診斷功能決定 C 中的那一個輸入不讓輸出 B 變為 ON。</p>	輸出需要

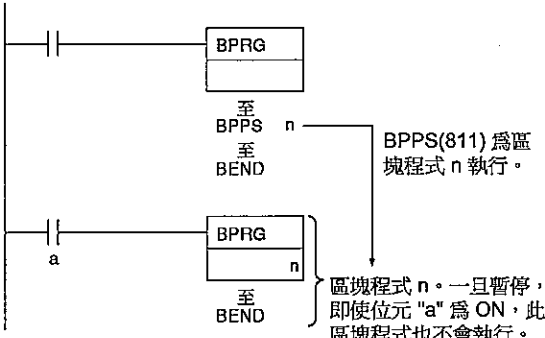
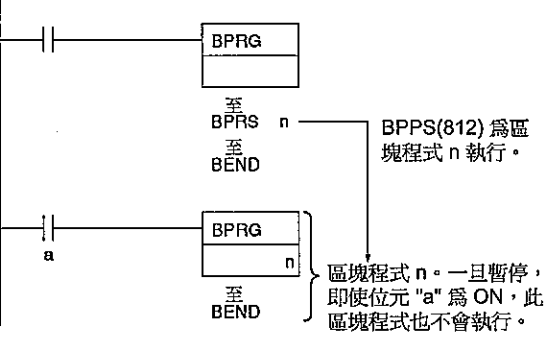
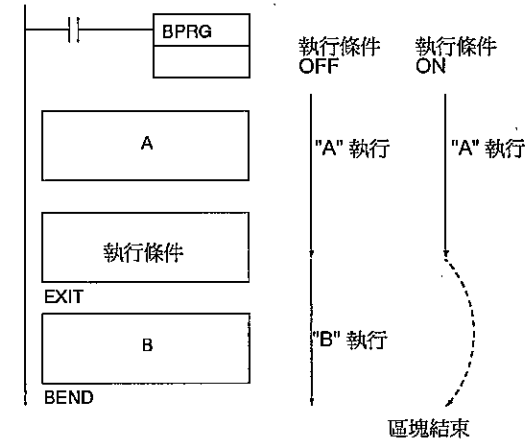
8-29 其他指令

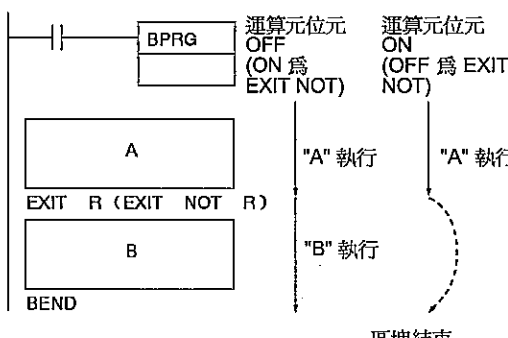
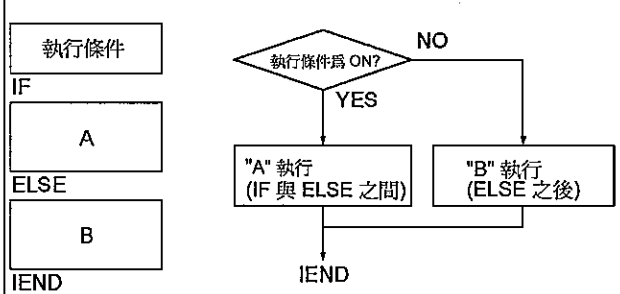
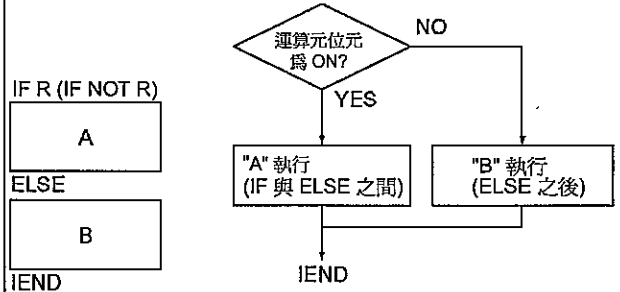
指令 助憶碼	符號 / 運算元	功能	位置 執行條件
SET CARRY STC @STC 040	STC(040)	設定進位旗標 (CY)。	輸出 需要
CLEAR CARRY CLC @CLC 041	CLC(041)	把進位旗標 (CY) 變為 OFF。	輸出 需要
SELECT EM BANK EMBC @EMBC 281	EMBC(281) N N: EM 觸排編號	改變現行的 EM bank。	輸出 需要
EXTEND MAXIMUM CYCLE TIME WDT @WDT 094	WDT(094) T T: 設定 Cycle time 的時間	延長最大循環時間，不過僅適用於此指令執行的循環。	輸出 需要
SAVE CONDITIONAL FLAG (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) CCS @CCS 282	CCS(282)	儲存狀態旗標的狀態。	輸出 需要
LOAD CONDITIONAL FLAG (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) CCL @CCL 283	CCL(283)	讀出所儲存之狀態旗標的狀態。	輸出 需要
CONVERT ADDRESS FROM CV (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) FRMCV @FRMCV 284	FRMCV(284) S D S: 存放 CV 系列 記憶體位址之 words D: 輸出轉換結果 IR	把 CV 系列 PLC 記憶體位址轉換為 CS 系列 PLC 記憶體位址。	輸出 需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
CONVERT ADDRESS TO CV (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D) TOCV @TOCV 285	<div style="border: 1px solid black; padding: 2px; display: inline-block;">TOCV(285)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">S</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">D</div> S: 存放 CS 系列 記憶體位址之 索引暫存器 D: 輸出轉換結果	把 CS 系列 PLC 記憶體位址轉換為 CV 系列 PLC 記憶體位址。	輸出 需要
DISABLE PERIPHERAL SERVICING (僅 CS1-H, CJ1-H, 或 CJ1M) IOISP @IOISP 287	<div style="border: 1px solid black; padding: 2px; display: inline-block;">IOISP(287)</div>	使並行處理模式或週邊服務優先模式中的程式執行期間不能進行週邊服務。	輸出 需要
ENABLE PERIPHERAL SERVICING (僅 CS1-H, CJ1-H, 或 CJ1M) IORS 288	<div style="border: 1px solid black; padding: 2px; display: inline-block;">IORS(288)</div>	使被 IOISP(287) 除能之並行處理模式或週邊服務優先模式中的程式執行期間之週邊服務可以進行。	輸出 不需要

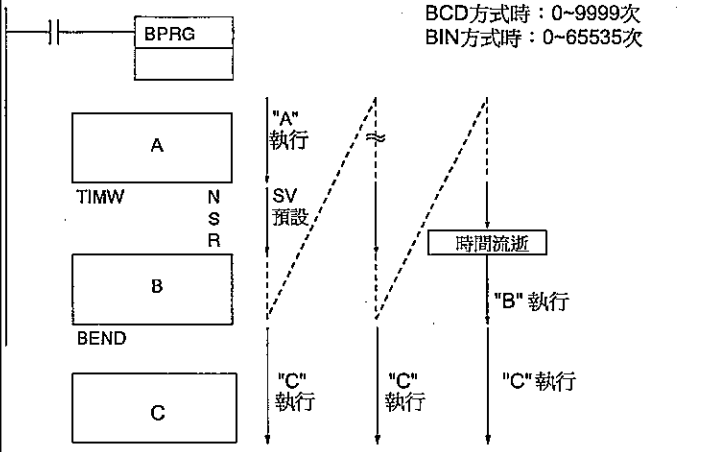
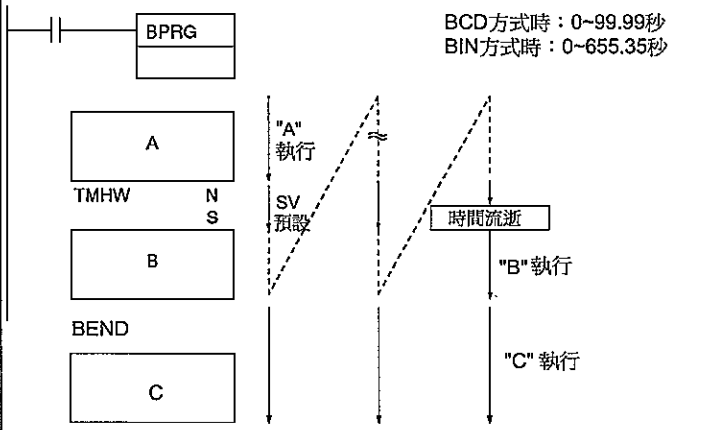
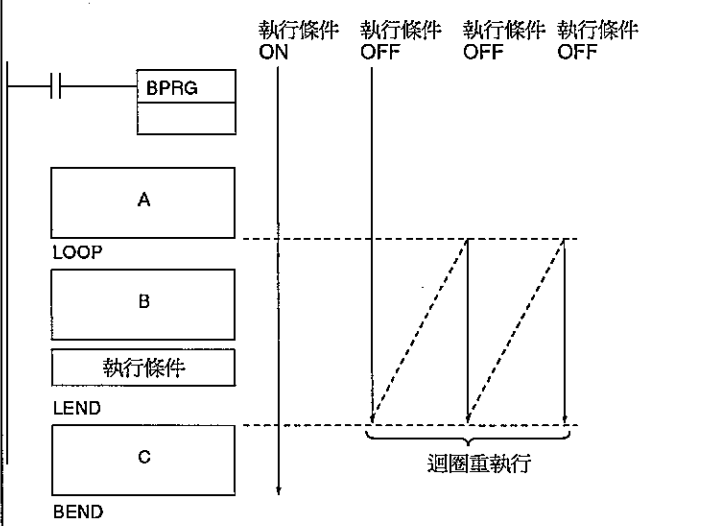
8-30 區塊規劃指令

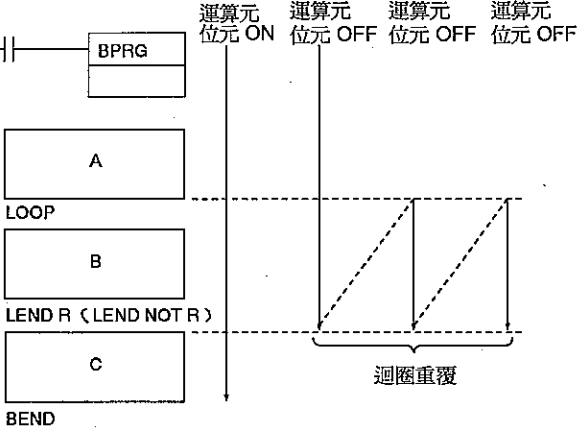
指令 助憶碼	符號 / 運算元	功能	位置 執行條件
BLOCK PROGRAM BEGIN BPRG 096	<div style="border: 1px solid black; padding: 2px; display: inline-block;">BPRG(096)</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">N</div> N: 區塊程式編號	定義一區塊規劃區域。對於每個 BPRG(096) 都需要一個相對應的 BEND(801)。 	輸出 需要
BLOCK PROGRAM END BEND 801		程式執行至 BEND 時，為 BPRG~BEND 區的結束。對於每個 BPRG(096) 都需要一個相對應的 BEND(801)。	區塊程式 需要

指令 助億碼	符號 / 運算元	功能	位置 執行條件
BLOCK PROGRAM PAUSE BPPS 811	BPPS (811) N N: 區塊程式編號	由另一個區塊程式暫停與重新開始指定的區塊程式。 	區塊程式需要
BLOCK PROGRAM RESTART BPRS 812	BPRS (812) N N: 區塊程式編號	由另一個區塊程式暫停與重新開始指定的區塊程式。 	區塊程式需要
CONDITIONAL BLOCK EXIT EXIT 806	EXIT(806) B: 位元運算元	EXIT執行條件 ON 時，EXIT到BEND (B段程式) 不執行。 	區塊程式需要

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
CONDITIONAL BLOCK EXIT EXIT 806	EXIT(806) B B: 位元運算元	EXIT執行條件 ON 時，EXIT到BEND (B段程式) 不執行。 	區塊程式 需要
CONDITIONAL BLOCK EXIT (NOT) EXIT NOT 806		EXIT 執行條件 ON 時，EXIT 到 BEND (B 段程式) 不執行。	區塊程式 需要
CONDITIONAL BLOCK BRANCHING IF 802	IF (802) B B: 位元運算元	若執行條件為 ON，將執行 IF(802) 與 ELSE(803) 之間的指令； 而若執行條件為 OFF，將執行 ELSE(803) 與 IEND(804) 之間的指令。 	區塊程式 需要
CONDITIONAL BLOCK BRANCHING IF 802	IF (802) B B: 位元運算元	若運算元位元為 ON，將執行 IF(802) 與 ELSE(803) 之間的指令。 若運算元位元為 OFF，將執行 ELSE(803) 與 IEND(804) 之間的指令。 	區塊程式 需要
CONDITIONAL BLOCK BRANCHING (NOT) IF NOT 802	IF (802) NOT B B: 位元運算元	若運算元位元為 ON，在 IF(802) 與 ELSE(803) 之間的指令將會執行。若運算元位元為 OFF，在 ELSE(803) 與 IEND(804) 之間的指令將會執行。	區塊程式 需要
CONDITIONAL BLOCK BRANCHING (ELSE) ELSE 803		若 ELSE(803) 指令省略而運算元位元為 ON，則在 IF(802) 與 IEND(804) 之間的指令將會執行。	區塊程式 需要

指令 助億碼	符號 / 運算元	功能	位置 執行條件
CONDITIONAL BLOCK BRANCHING END IEND 804	---	若運算元位元為 OFF，只有 IEND(804) 之後的指令才會執行。	區塊程式 需要
ONE CYCLE AND WAIT WAIT 805	WAIT(805)	<p>若 WAIT(805) 的執行條件為 ON，則區塊程式中其餘的指令將會被跳過，也就是不執行B段承程式。</p>	區塊程式 需要
ONE CYCLE AND WAIT WAIT 805	WAIT(805) B B: 位元運算元	若運算元位元為 OFF (ON 對於 WAIT NOT(805))，程式區塊中其餘的指令將會被跳過。在下一個循環中，除了 WAIT(805) 或 WAIT(805) NOT 的執行條件之外程式區塊將完全不會被執行。當執行條件變為 ON (OFF 對於 WAIT(805) NOT)，由 WAIT(805) 或 WAIT(805) NOT 至程式結尾的指令才會被執行。	區塊程式 需要
ONE CYCLE AND WAIT (NOT) WAIT NOT 805	WAIT(805) NOT B B: 位元運算元	若運算元位元為 OFF (ON 對於 WAIT NOT(805))，程式區塊中其餘的指令將會被跳過。在下一個循環中，除了 WAIT(805) 或 WAIT(805) NOT 的執行條件之外程式區塊將完全不會被執行。當執行條件變為 ON (OFF 對於 WAIT(805) NOT)，由 WAIT(805) 或 WAIT(805) NOT 至程式結尾的指令才會被執行。	區塊程式 需要
TIMER WAIT TIMW 813 (BCD)	TIMW(813) N SV	<p>程式剩餘的部份會被延遲到指定的時間過去了才會繼續執行。當計時器的時間到時，將會由 TIMW(813) 後的下一個指令繼續執行。</p> <p>BCD方式時：0~999.9秒 BIN方式時：0~6553.5秒</p>	區塊程式 需要
TIMWX 816 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)	TIMWX(816) N SV	<p>N: 計時器編號 SV: 設定值</p> <p>N: 計時器編號 SV: 設定值</p>	

指令 助億碼	符號 / 運算元	功能	位置 執行條件
<p>COUNTER WAIT CNTW 814 (BCD)</p> <p>CNTWX 817 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)</p>	<p>CNTW(814) N SV</p> <p>N: 計數器編號 SV: 設定值 I: 計數輸入</p> <p>CNTWX(817) N SV</p> <p>N: 計數器編號 SV: 設定值 I: 計數輸入</p>	<p>程式剩餘的部份會被延遲到指定的計數量到達後才會繼續執行。當計數器的數到時，將會由 CNTW (814) 後的下一個指令繼續執行。</p> <p>BCD方式時：0~9999次 BIN方式時：0~65535次</p> 	<p>區塊程式 需要</p>
<p>HIGH-SPEED TIMER WAIT TMHW 815 (BCD)</p> <p>TMHWX 818 (BIN) (僅 CS1-H, CJ1-H, CJ1M, 或 CS1D)</p>	<p>TMHW(815) N SV</p> <p>N: 計時器編號 SV: 設定值</p> <p>TMHWX(818) N SV</p> <p>N: 計時器編號 SV: 設定值</p>	<p>程式剩餘的部份會被延遲到指定的時間到達後才會繼續執行。當計時器的時間到時，將會由 TMHW(815) 後的下一個指令繼續執行。SV = 0 至 99.99 s</p> <p>BCD方式時：0~99.99秒 BIN方式時：0~655.35秒</p> 	<p>區塊程式 需要</p>
<p>LOOP LOOP 809</p>	<p>---</p>	<p>LOOP(809) 表示迴圈程式的開頭。</p> 	<p>區塊程式 需要</p>

指令 助憶碼	符號 / 運算元	功能	位置 執行條件
LEND LEND 810	LEND(810)	LEND(810) 或 LEND(810) NOT 表示迴圈的結束。當程式執行到 LEND(810) 或 LEND(810) NOT 時，會回到 LOOP(809) 的下一個指令，直到 LEND(810) 或 LEND(810) NOT 的運算元位元分別變為 ON 或 OFF 或是直到 LEND(810) 的執行條件變為 ON 為止。	區塊程式 需要
LEND LEND 810	LEND (810) B B: 位元運算元	<p>若 LEND(810) 的運算元位元為 OFF (或 LEND(810) NOT 的為 ON)，迴圈的執行會由 LOOP(809) 後的下一個指令開始重覆。若 LEND(810) 的運算元位元為 ON (或 LEND(810) NOT 的為 OFF)，則迴圈結束並且會由 LEND(810) 或 LEND(810) NOT 後的下一個指令繼續程式的執行。</p>  <p>附註：對於 LEND(810) NOT，運算元位元的狀態應該要相反。</p>	區塊程式 需要
LEND NOT LEND NOT 810	LEND(810) NOT B: 位元運算元	LEND(810) 或 LEND(810) NOT 表示迴圈的結束。當程式執行到 LEND(810) 或 LEND(810) NOT 時，會回到 LOOP(809) 的下一個指令，直到 LEND(810) 或 LEND(810) NOT 的運算元位元分別變為 ON 或 OFF 或是直到 LEND(810) 的執行條件變為 ON 為止。	區塊程式 需要

8-31 字串處理指令

指令 助億碼	符號 / 運算元	功能	位置 執行條件				
MOV STRING MOV\$ @MOV\$ 664	<table border="1"> <tr><td>MOV\$(664)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 傳送字串的低四位元CH編號 D: 輸出低四位元CH編號</p>	MOV\$(664)	S	D	<p>傳送一文字字串。</p>	輸出 需要	
MOV\$(664)							
S							
D							
CONCATENATE STRING +\$ @+\$ 656	<table border="1"> <tr><td>+(656)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 設定字串低四位元CH編號1 S2: 設定字串低四位元CH編號2 D: 輸出整合後的低四位元CH編號</p>	+(656)	S1	S2	D	<p>把一個字串整合到另一個字串之後。</p>	輸出 需要
+(656)							
S1							
S2							
D							
GET STRING LEFT LEFT\$ @LEFT\$ 652	<table border="1"> <tr><td>LEFT\$(652)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 設定字串的低四位元CH編號 S2: 設定取出字串的資料 D: 表示輸出字串的低四位元CH編號</p>	LEFT\$(652)	S1	S2	D	<p>由一文字字串的左邊 (開頭) 取出指定數量的字元。</p>	輸出 需要
LEFT\$(652)							
S1							
S2							
D							
GET STRING RIGHT RGHT\$ @RGHT\$ 653	<table border="1"> <tr><td>RGHT\$(653)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 設定字串的低四位元CH編號 S2: 設定取出字串的資料 D: 表示輸出字串的低四位元CH編號</p>	RGHT\$(653)	S1	S2	D	<p>由一文字字串的右邊 (結尾) 讀出指定數量的字元。</p>	輸出 需要
RGHT\$(653)							
S1							
S2							
D							

指令	符號 / 運算元	功能	位置 執行條件						
GET STRING MIDDLE MID\$ @MID\$ 654	<table border="1"> <tr><td>MID\$(654)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 設定字串的低四位元CH編號 S2: 設定取出字串的資料 S3: 定義開始位置 D: 表示輸出字串的低四位元CH編號</p>	MID\$(654)	S1	S2	S3	D	由一文字字串的中間任何位置讀出指定數量的字元。 	輸出需要	
MID\$(654)									
S1									
S2									
S3									
D									
FIND IN STRING FIND @FIND\$ 660	<table border="1"> <tr><td>FIND\$(660)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>D</td></tr> </table> <p>S1: 被尋找字串的低四位元CH編號 S2: 字串的低位元CH編號 D: 輸出尋找結果的CH編號</p>	FIND\$(660)	S1	S2	D	由一文字字串中尋找一指定的字串。 	輸出需要		
FIND\$(660)									
S1									
S2									
D									
STRING™™™™ LENS\$ @LENS\$ 650	<table border="1"> <tr><td>LENS\$(650)</td></tr> <tr><td>S</td></tr> <tr><td>D</td></tr> </table> <p>S: 字串列低位元CH編號 D: 輸出演算結果的CH編號</p>	LENS\$(650)	S	D	計算一文字字串的長度。 	輸出需要			
LENS\$(650)									
S									
D									
REPLACE IN STRING RPLC\$ @RPLC\$ 661	<table border="1"> <tr><td>RPLC\$(661)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>S4</td></tr> <tr><td>D</td></tr> </table> <p>S1: 字串列低位元CH編號 S2: 定義要取代的字串CH編號 S3: 定義要取代的字串數量 S4: 開始位置 D: 輸出取代結果的CH編號</p>	RPLC\$(661)	S1	S2	S3	S4	D	以指定位置的指定文字字串來取代一文字字串。 	輸出需要
RPLC\$(661)									
S1									
S2									
S3									
S4									
D									

指令	符號 / 運算元	功能	位置 執行條件					
DELETE STRING DEL\$ @DEL\$ 658	<table border="1"> <tr><td>DEL\$(658)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 字串低位元的CH編號 S2: 定義刪除起始位置 S3: 開始位置 D: 輸出刪除結果的CH編號</p>	DEL\$(658)	S1	S2	S3	D	<p>從一文字字串的中間位置刪除一指定的文字字串。</p> <p>要刪去的字元數 (由 S2 指定)</p>	輸出 需要
DEL\$(658)								
S1								
S2								
S3								
D								
EXCHANGE STRING XCHG\$ @XCHG\$ 665	<table border="1"> <tr><td>XCHG\$(665)</td></tr> <tr><td>Ex1</td></tr> <tr><td>Ex2</td></tr> </table> <p>Ex1: 取代CH編號1 Ex2: 取代CH編號2</p>	XCHG\$(665)	Ex1	Ex2		輸出 需要		
XCHG\$(665)								
Ex1								
Ex2								
CLEAR STRING CLR\$ @CLR\$ 666	<table border="1"> <tr><td>CLR\$(666)</td></tr> <tr><td>S</td></tr> </table> <p>S: 文字字串的第一個words</p>	CLR\$(666)	S	<p>以 NULL (00 Hex) 取代整個文字字串。</p>	輸出 需要			
CLR\$(666)								
S								

指令 助憶碼	符號 / 運算元	功能	位置 執行條件					
INSERT INTO STRING INSS @INSS 657	<table border="1" style="margin-left: 20px;"> <tr><td>INS\$(657)</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> <tr><td>S3</td></tr> <tr><td>D</td></tr> </table> <p>S1: 基底文字字串的第一個 words S2: 要插入的文字字串之第一個 words S3: 開始位置 D: 第一個目標 words</p>	INS\$(657)	S1	S2	S3	D	<p>由一文字字串中插入指定字串。</p>	輸出 需要
INS\$(657)								
S1								
S2								
S3								
D								
String comparison LD, AND, 或 + =\$, <>\$, <\$, <=\$, >\$, >=\$ 670 (=)\$ 671 (<>\$) 672 (<\$) AND 673 (<=\$) 674 (>\$) 675 (>=\$) OR <table border="1" style="margin-left: 20px;"> <tr><td>Symbol</td></tr> <tr><td>S1</td></tr> <tr><td>S2</td></tr> </table> <p>S1: 文字字串 1 S2: 文字字串 2</p>	Symbol	S1	S2	<p>字串比較指令 (=, <>, <, <=, >, >=) 根據 ASCII 碼的值由兩個字串開始比較。若比較的結果成立時，則會為 LOAD, AND, 或 OR 建立 ON 的執行條件。</p>	LD: 不需要 AND, 或: 需要			
Symbol								
S1								
S2								

8-32 工作單控制指令

指令	符號 / 運算元	功能	位置 執行條件
TASK ON TKON @TKON 820	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> TKON(820) N </div> N: 工作單編號	<p>使指定的工作單可執行。</p> <p>指定之工作單的編號大於本地工作單的編號 ($m < n$)。</p> <p>指定之工作單的編號小於本地工作單的編號 ($m > n$)。</p>	輸出需要
TASK OFF TKOF @TKOF 821	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> TKOF(821) N </div> N: 工作單編號	<p>使指定的工作單進入待命狀態。</p> <p>指定之工作單的編號大於本地工作單的編號 ($m < n$)。</p> <p>指定之工作單的編號小於本地工作單的編號 ($m > n$)。</p>	輸出需要