

M Series

Programmable Controller

M PLC 指令使用手冊



NEXT Level SOLUTION

因手冊內容會隨著版本變更而做修改，此版本不一定會是最終版本。
若要下載最新版的手冊請到 www.fatek.com 的技術支援專區。

永宏電機股份有限公司

目錄

第 1 章	PLC 階梯圖程式基本原理.....	1-1
1-1	階梯圖工作原理.....	1-1
1-2	傳統階梯圖和 PLC 階梯圖之差異	1-4
1-3	階梯圖組成及其術語定義	1-6
第 2 章	PLC 內部之記憶體配置及其單點(數位)與暫存器明細	2-1
2-1	M SERIES 記憶體配置	2-2
2-2	單點 (Digital) 及暫存器之配置	2-3
2-3	CPU 模組特殊繼電器明細	2-6
2-4	CPU 模組特殊暫存器明細	2-14
2-5	Motion 模組特殊繼電器明細	2-33
2-6	Motion 模組特殊暫存器明細	2-35
2-7	Motion 錯誤與警報暫存器相關明細	2-36
第 3 章	M SERIES 指令一覽表	3-1
3-1	順序指令一覽表.....	3-1
3-2	應用指令一覽表.....	3-3
第 4 章	順序指令說明	4-1
4-1	元件指令特性說明.....	4-1
第 5 章	應用指令說明	5-1
5-1	應用指令之通則.....	5-1
5-2	利用 w 前綴字作字元組及位元存取變換	5-9

5-3	數目系統.....	5-9
5-4	利用指標暫存器 (XR) 作間接定址.....	5-12
5-5	運算元遞增/減之溢位與欠位(初學者請略過本節)	5-18
5-6	加/減運算之進位與借位.....	5-19
第 6 章 基本應用指令		6-1
6-1	一般計時器(T).....	6-1
6-2	一般計數器(C).....	6-6
6-3	設定(SET).....	6-10
6-4	清除(RST)	6-13
6-5	主控迴路開始指令(MC)	6-15
6-6	主控終止指令(MCE).....	6-18
6-7	跳過迴路開始指令(SKIP)	6-19
6-8	跳過迴路終止指令(SKPE).....	6-21
6-9	上微分指令(DIFU)	6-22
6-10	下微分指令(DIFD)	6-24
6-11	位元位移(BSHF)	6-25
6-12	上/下數計數器(UDCTR)	6-27
6-13	搬移(MOV)	6-30
6-14	倒相後搬移(MOV/)	6-32
6-15	交替開關(TOGG)	6-34
6-16	快速加法運算 F (+)	6-35
6-17	快速減法運算 F (-)	6-38
6-18	加法運算 (+)	6-41

6-19	減法運算 (-)	6-43
6-20	乘法運算 (*)	6-45
6-21	除法運算 (/)	6-48
6-22	MOD 取餘數	6-52
6-23	遞增 (+ 1)	6-54
6-24	遞減 (- 1)	6-56
6-25	數值比較 (CMP)	6-58
6-26	邏輯及運算(AND).....	6-60
6-27	邏輯或運算(OR)	6-62
6-28	BIN→BCD 變換	6-64
6-29	BCD→BIN 變換	6-66
 第 7 章 進階應用指令		7-1
7-1	數學運算指令 (FUN24 ~ 33)	7-1
7-2	邏輯運算指令 (FUN35 ~ 36)	7-29
7-3	比較指令 (FUN37)	7-33
7-4	搬移指令一 (FUN40 ~ 50)	7-35
7-5	位移 / 旋轉指令 (FUN51 ~ 54)	7-51
7-6	數碼變換指令 (FUN55 ~ 64)	7-59
7-7	流程控制指令 (FUN22 、 FUN65 ~ 71)	7-76
7-8	I / O 指令一 (FUN74 ~ 86)	7-98
7-9	PID 控制 (FUN38 、 FUN99)	7-100
7-10	積算型計時器指令 (FUN87 ~ 89)	7-116
7-11	監控計時器指令 (FUN90 ~ 91)	7-120

7-12	高速計數器 / 計時器指令 (FUN92 ~ 93)	7-125
7-13	緩升 / 緩降指令 (FUN95 ~ 98)	7-131
7-14	列表指令 (FUN100 ~ 114)	7-138
7-15	矩陣指令 (FUN120 ~ 130)	7-173
7-16	NC 定位控制指令一 (FUN137 ~ 144)	7-189
7-17	中斷控制指令 (FUN145 ~ 146)	7-226
7-18	NC 定位控制指令二 (FUN147~FUN149)	7-232
7-19	通訊指令 (FUN150 ~ 156)	7-239
7-20	搬移指令二 (FUN160 ~ 162)	7-248
7-21	接點型比較指令 (FUN170 ~ 175)	7-259
7-22	運動控制指令	7-272
7-23	其他指令 (FUN115、FUN258)	7-344
7-24	浮點運算指令 (FUN200 ~ 220)	7-350
第 8 章	步進指令說明	8-1
8-1	步進階梯圖工作原理	8-2
8-2	步進階梯圖基本組成	8-3
8-3	步進指令介紹：STP、FROM、TO、STPEND	8-8
8-4	步進階梯圖寫法	8-26
8-5	實際應用範例	8-31
8-6	步進程式語法檢查錯誤碼說明	8-38
修訂紀錄		8-0

使用本產品注意事項

符合應用之條件

FATEK 的產品適用評估並安裝於經過全面設計的設備或系統。

請使用者自行確認目前所使用的系統、機械或是裝置是否適用於 FATEK 產品。如未確認是否符合或適用時，本公司無須對產品的適用性負責。

如客戶要求，FATEK 將提供相應的第三方認證來明確適用於產品的額定值和使用限制。該認證信息本身不足以完全決定 FATEK 產品與最終產品、機器、系統及其它應用或組合的適用性。

以下為一些必須引起特別注意的應用場合，但下述內容並非為包括所有可能的產品用途，也不表示所列用途對產品均適用：戶外使用、在遭受潛在化學污染或電氣干擾處使用、或未在本手冊中提及的條件或用途。可能對生命或財產造成風險的系統、機器和設備。

務必事先確認系統整體是否有危險告示、並採用備援設計等可確保安全性的設計，否則不得將產品用於與人身財產安全密切相關的場合。FATEK 對於客戶在其應用中的產品組合或產品使用的規格、法規或限制等，不承擔任何責任。

使用本產品時，FATEK 不對用戶編輯的程式或其引起的後果承擔任何責任。

免責聲明

尺寸和重量

手冊記載的尺寸和重量僅為名義值，即使已說明了公差，也不能用於製造用途。

性能數據

本手冊中給出的這些數據僅表示在 FATEK 測試條件下的性能數據僅供用戶作為確定符合應用的參考，用戶必須將其與實際應用條件互相考慮。實際性能遵從 FATEK 保證內容和責任限制。

錯誤和疏忽

本手冊中的內容已仔細核對並認為是準確的；但對於文字、印刷和校對錯誤或疏忽不承擔任何責任。

規格變更

產品規格和附件可能會因技術改進或其它原因而隨時變更。當公佈的規格、性能改變，或者進行過重大的結構改變時，FATEK 通常會改變型號。若產品的某些規格發生變更時，以下情況不另行通知：根據客戶的要求，對客戶的應用指定特別的型號或設定特定的規格。歡迎隨時洽詢 FATEK 代理經銷商，確認所購產品的實際規格。

1

PLC 階梯圖程式基本原理

1-1	階梯圖工作原理	1-1
1-1-1	組合邏輯	1-1
1-1-2	順序邏輯	1-3
1-2	傳統階梯圖和 PLC 階梯圖之差異	1-4
1-3	階梯圖組成及其術語定義	1-6

本章將介紹 PLC 階梯圖程式之基本原理。

1-1 階梯圖工作原理

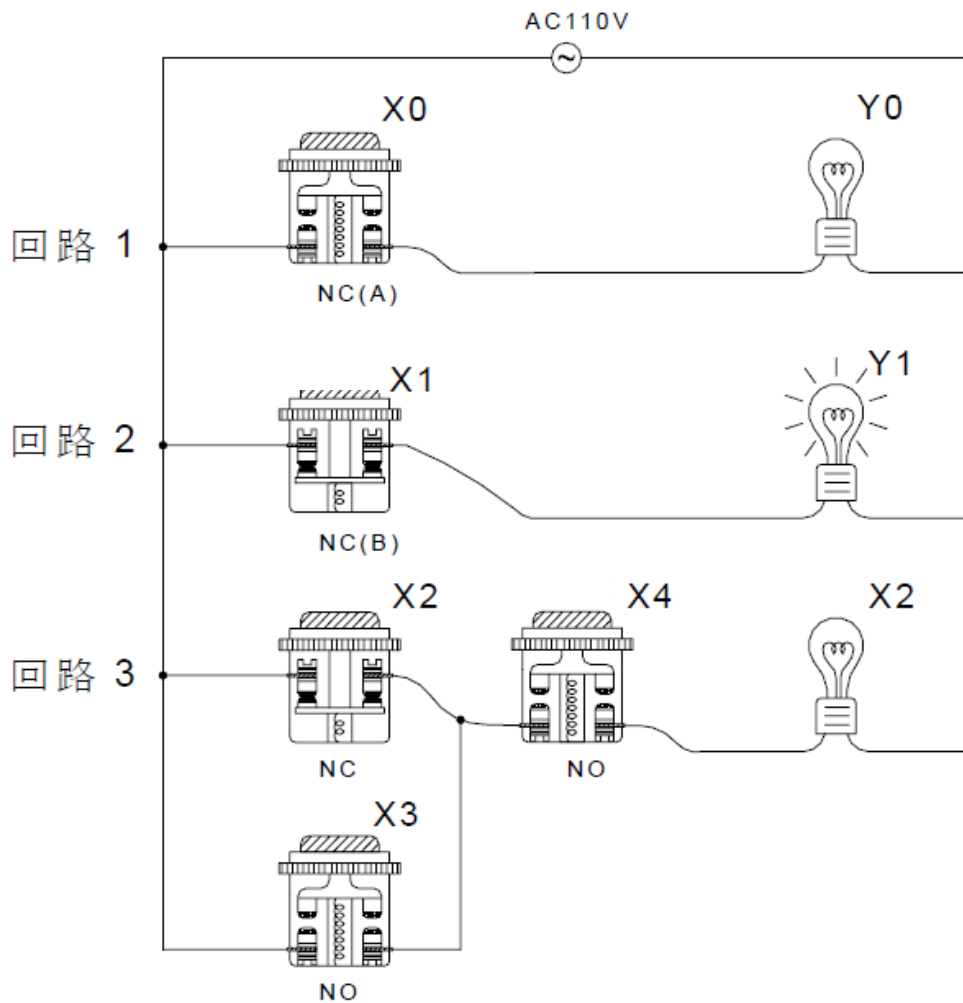
階梯圖為二次世界大戰期間所發展出來之自動控制圖形語言，是歷史最久、使用最廣之自動控制語言，最初只有 A (常開)接點、B (常閉)接點、輸出線圈、計時器、計數器等。

基本機構元件 (今日仍在使用之配電盤即是)，直到微電腦 PLC 出現後，階梯圖之元件(語言) 除上述元件外尚增加了諸如微分接點、保持線圈等元件(表二) 以及傳統配電盤無法達成之應用指令。

無論傳統階梯圖或 PLC 階梯圖其工作原理均相同，只是在符號表示上傳統階梯圖以較接近實體之符號表示，而 PLC 則採用較簡明且易於電腦或報表上表示之符號表示。在階梯圖邏輯方面可分為組合邏輯和順序邏輯兩種，茲分述如下：

1-1-1 組合邏輯

組合邏輯之階梯圖，係單純地將單一或一個以上之輸入元件組合 (串、並聯等)後再將結果送到輸出元件 (線圈、計時 / 計數器或應用指令等) 之回路結構。

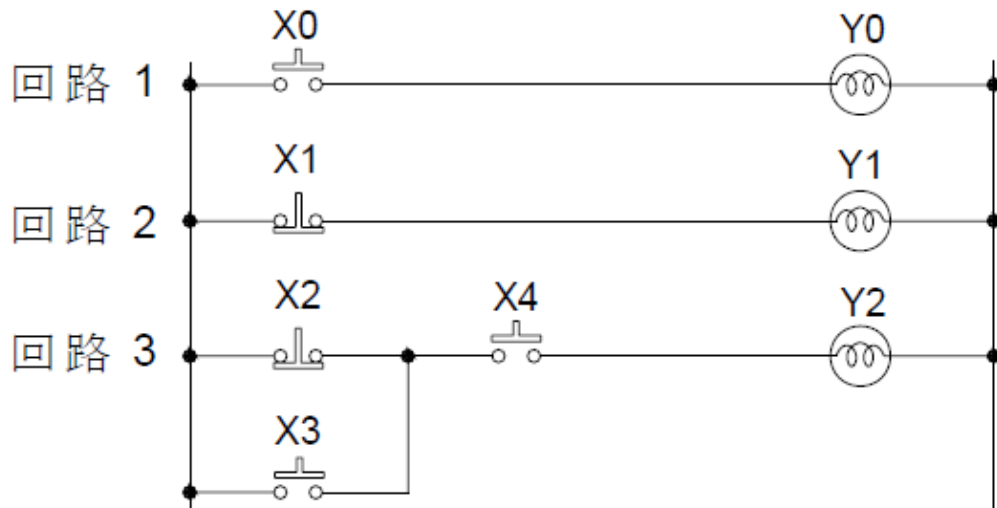


組合邏輯_實際配線圖

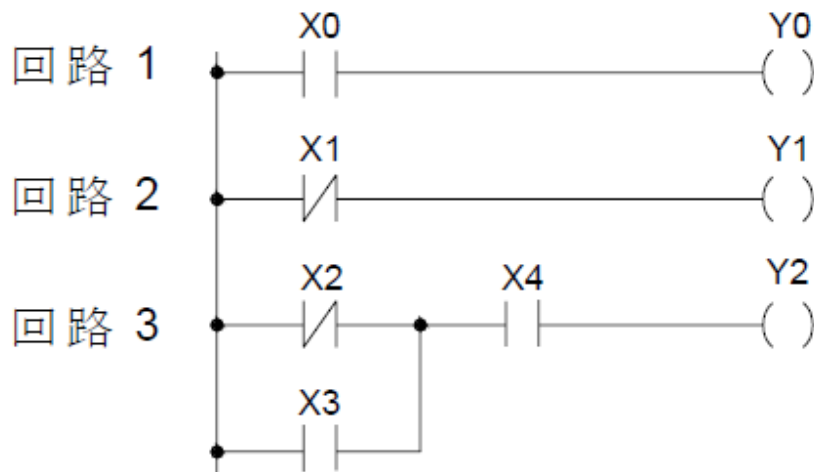
本例為組合邏輯分別以實際配線、傳統階梯圖 PLC 階梯圖表示之範例，其中回路 1 使用一常開開關 (NO：Normally Open) 亦即一般所謂之 "A" 開關或接點。其特性是在平常 (未壓下) 時，其接點為開路 (OFF) 狀態，故燈泡不亮，而在開關動作 (壓下按鈕) 時，其接點變為導通 (ON)，故燈泡點亮。

相對地，回路 2 使用一常閉開關 (NC：Normally Close) 亦即一般所稱之 "B" 開關或接點，其特性是在平常時其接點為導通，故燈泡點亮，而在開關動作時其接點反而變成開路，故燈泡熄滅。

回路 3 為一個以上輸入元件之組合邏輯輸出範例，其輸出 Y2 燈泡只有在 X2 不動作或 X3 動作且 X4 為動作時才會點亮。



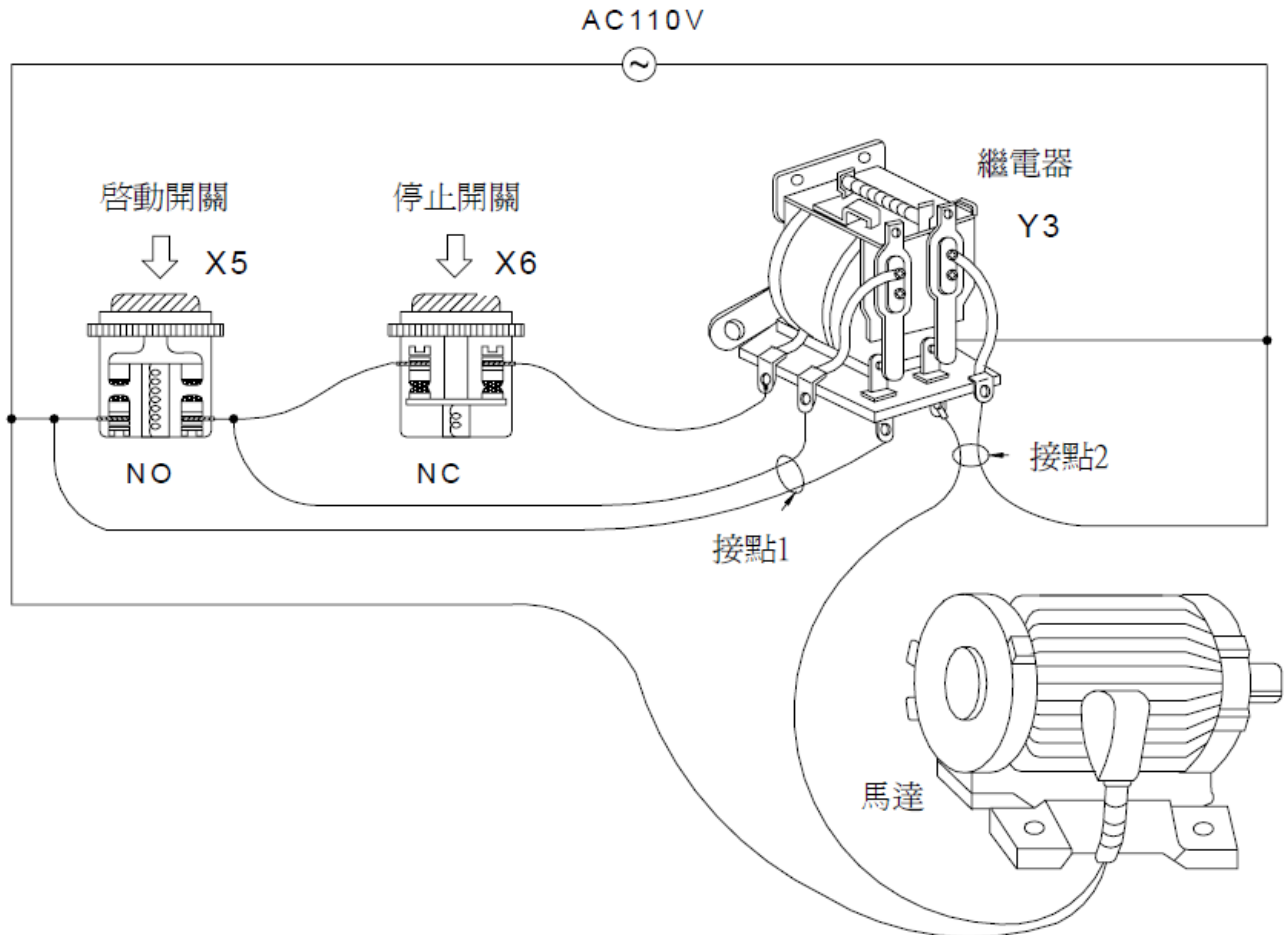
組合邏輯_傳統階梯圖



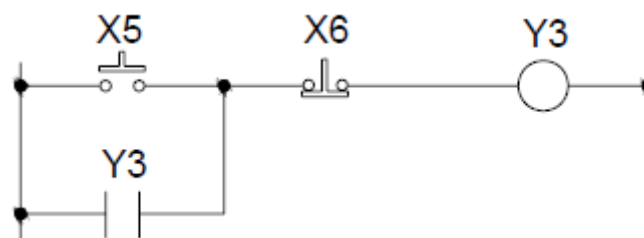
組合邏輯_PLC 階梯圖

1-1-2 順序邏輯

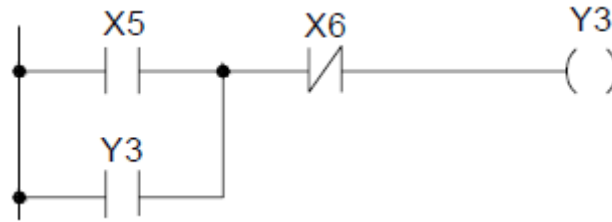
順序邏輯為具有回授結構之回路，亦即將回路輸出結果拉回當輸入條件，如此在相同輸入條件下，會因前次狀態或動作順序之不同，而得到不同之輸出結果，茲以下圖具自保功能之馬達啟動 / 停止回路作說明。



順序邏輯_實際配線圖



順序邏輯_傳統階梯圖



順序邏輯_PLC 階梯圖

在此回路剛接上電源時，雖 X6 開關為 ON，但 X5 開關為 OFF，故繼電器不動作，而繼電器之輸出接點 1 和接點 2 均為 A 接點 (繼電器動作時才 ON)，故接點 1 和接點 2 均不導通，馬達在停止狀態。在啟動開關 X5 按下後，繼電器動作，接點 1 及接點 2 同時 ON，馬達開始運轉，一旦繼電器動作後，即使放開啟動開關 (X5 變成 OFF) 繼電器電源因為自身之接點 1 回授而仍可繼續保持動作 (此即為自我保持回路)，其動作可以下表表示：

	X5 開關(NO)	X6 開關(NC)	馬達(繼電器)狀態
① ↓	放開	放開	停止
② ↓	壓下	放開	動作
③ ↓	放開	放開	動作
④ ↓	放開	壓下	停止
⑤	放開	放開	停止

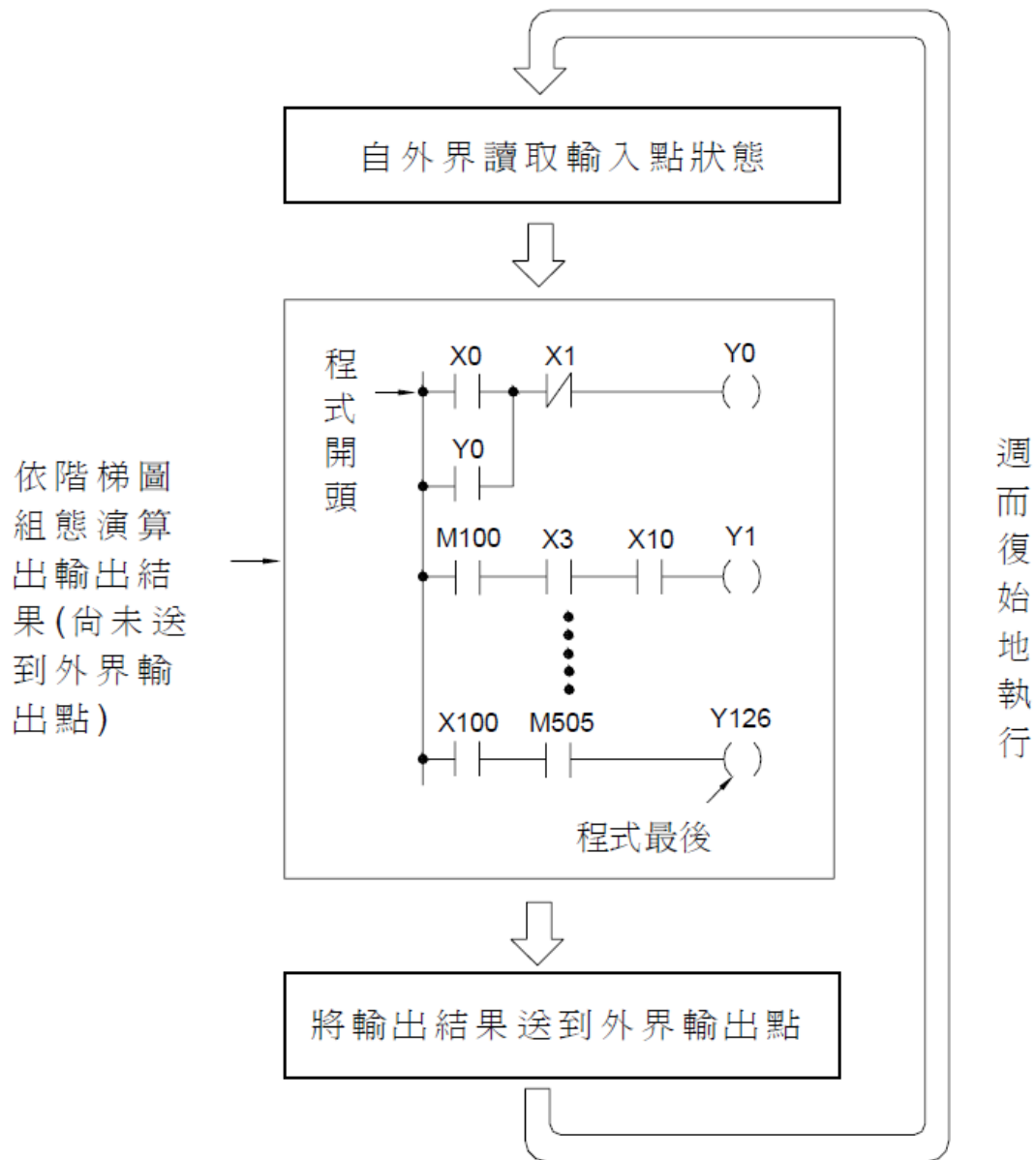
順序邏輯_動作

由上表可知在不同順序下，雖輸入狀態完全一致，其輸出結果亦可能不一樣，如表中之狀態 1 和 3 其 X5 和 X6 的開關均為放開，在狀態 1 下馬達為停止，但狀態 3 時馬達卻為運轉，此種繼電器輸出狀態拉回當輸入 (即所謂之回授) 而使回路具有順序控制效果是階梯圖回路之主要特性，因之有人稱階梯圖為 "順序控制回路"，而將 PLC 稱為順序控制器 (Sequencer)。在本節範例中僅列舉 A、B 接點和輸出線圈作說明，其他元件之用法和此相同，請參考第 5 章 "順序指令說明"。

1-2 傳統階梯圖和 PLC 階梯圖之差異

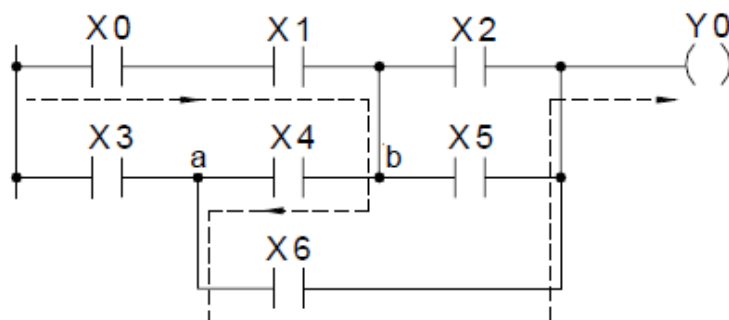
雖然傳統階梯圖和 PLC 階梯圖之工作原理是完全一致的，但實際上 PLC 僅是利用微電腦 (CPU) 來模擬傳統階梯圖之動作，亦即利用掃描的方式逐一地查看所有輸入元件及輸出線圈之狀態，再將此等狀態依階梯圖之組態邏輯來演算出和傳統階梯圖一樣之輸出結果，但因 CPU 只有一個，只能逐一地查看階梯圖程式，並依該程式及輸入/出狀態演算輸出結果，再將結果送到輸出界面，然後又重新讀取輸入狀態、演算、輸出，如此週而復始地循環執行上述動作，此一完整之循環動作所費之時間稱之為掃描時間，其時間會隨著程式之增大而加長，此掃描時間將造成 PLC 從輸入檢知到輸出反應之延遲，延遲時間愈長對控制所造成之誤差愈大，甚至造成無法勝任控制要求之情況，此時就必須選用掃描速度更快之 PLC，因此 PLC 之掃描速度是 PLC 之重要規

格，惟拜微電腦技術精進之賜，現今之 PLC 在掃描速度上均有極大之改善，以 M SERIES 為例 1K step 接點之掃描時間在連續位址讀取狀況之下只需 1us，下圖為 PLC 之階梯圖程式掃描之示意圖。



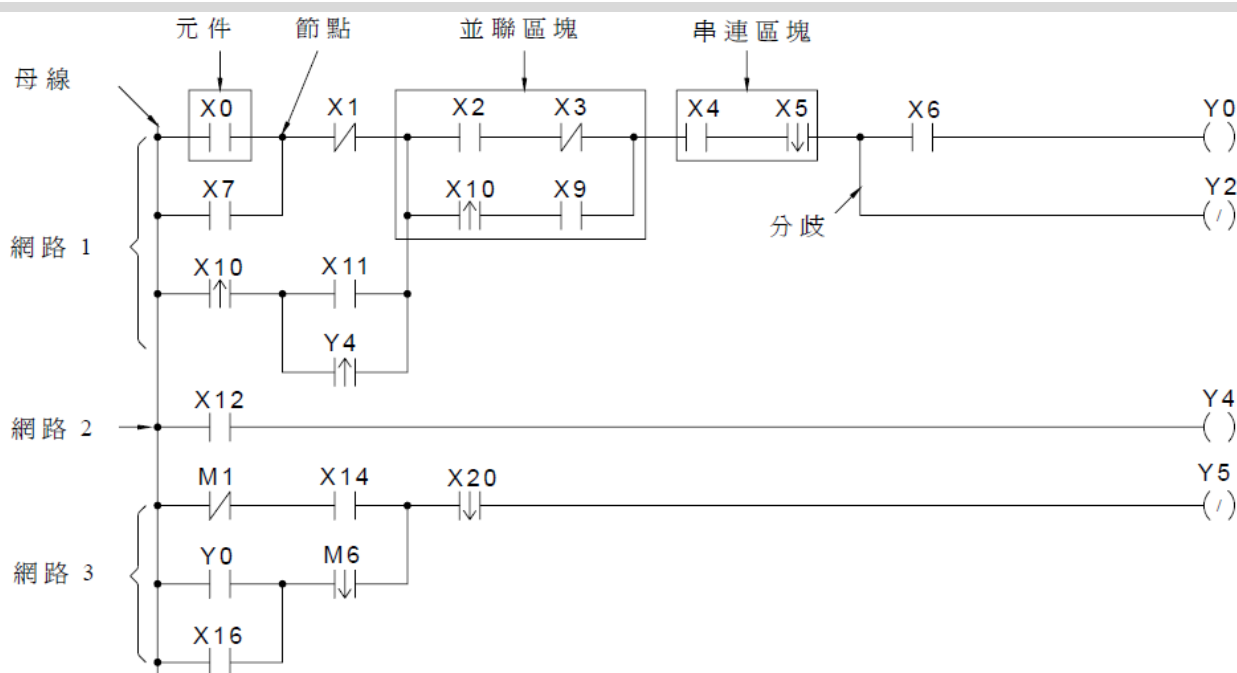
PLC 之階梯圖程式掃描之示意圖

除上述掃描時間差異外，PLC 階梯圖和傳統階梯圖尚有如下之 "逆向回流" 之差異，如下圖所示圖中若 X0、X1、X4、X6 為導通，其他為不導通，在傳統之階梯圖回路上輸出 Y0 會如虛線所示形成迴路而為 ON，但在 PLC 階梯圖因 PLC 之 CPU 在演算階梯圖程式之結果時，係由左而右，由上而下地掃描。在同樣輸入條件下，本圖例中之 a 點狀態因 X3 接點 OFF 故 CPU 認定為 OFF，雖然 a 點經由 X4 接至 b 點均為 ON，但因 PLC 階梯圖只由左至右掃描，CPU 無法查覺，故 Y0 輸出為 OFF。



傳統階梯圖之逆向回流

1-3 階梯圖組成及其術語定義



階梯圖程式範例

註：M SERIES 系列之網路最大為 22 行×16 列。

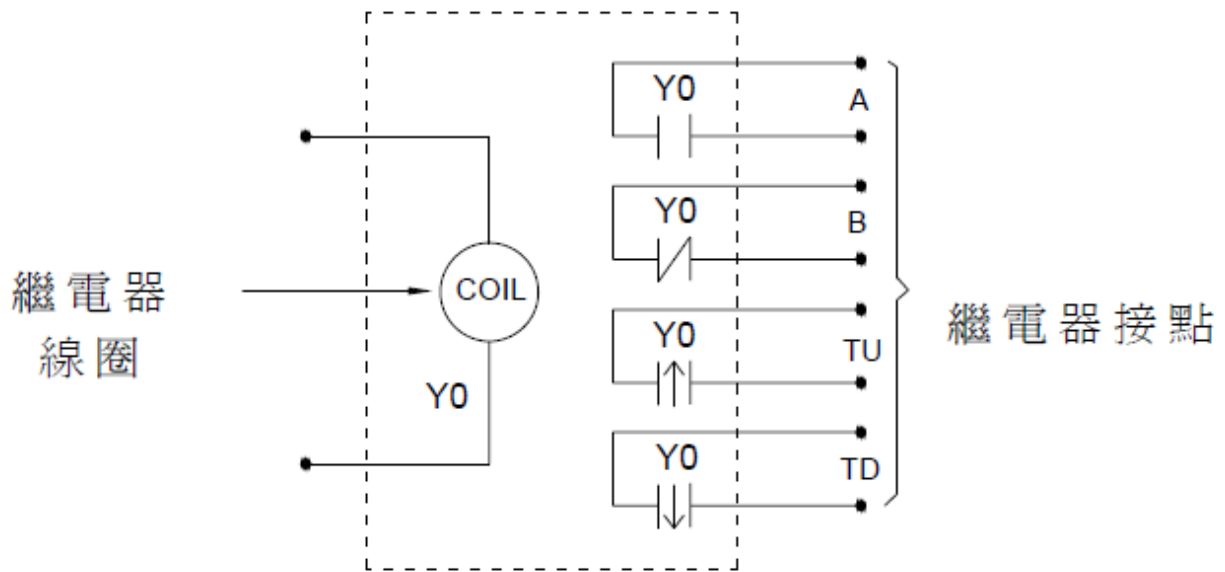
如上階梯圖程式可分為一個個小方塊 (本圖例為 8 列 × 11 行 = 88 個小方塊)，每個小方塊均可放置一個元件，將所有元件依控制需求作成各種不同之連結即構成所謂之階梯圖程式，茲就階梯圖程式相關之術語及其意義，分述如下：

①. 接點(Contact)

接點為表示導通 (ON) 與不導通 (OFF) 狀態之元件，共有兩類。一為 "輸入接點" (編號以 X 開頭者)，其狀態是來自外界 (端子上之輸入點)。另一為 "繼電器附屬之接點" (請參考第 2 項說明)，其狀態是反應 (來自) 繼電器線圈之狀態。M SERIES 系列所提供之接點有 A 接點、B 接點、上 / 下微分接點、開 / 短路接點 6 種，請參閱第 4 項元件之說明。

②. 繼電器(Relay)

正如同傳統繼電器，它包含線圈(Coil) 和接點(Contact)，如下圖例所示。



繼電器_線圈和接點

如圖示繼電器必有線圈，欲使繼電器動作，需驅動其線圈(用 OUT 指令驅動)，在線圈被驅動後，其接點狀態會受到影響。

如圖例若將 Y0 以 1 驅動(使之為 ON)，則繼電器之 A 接點為 1，B 接點為 0，TU 接點只 ON 一個掃描時間，TD 接點為 0。

當 Y0 變成 OFF 時，A 接點為 0，B 接點為 1，TU 接點為 0，而 TD 接點只 ON 一個掃描時間(A、B、TU、TD 接點之動作請參閱第 4 章 "順序指令說明")。

M SERIES 之繼電器有四種，分別為 Y $\Delta\Delta\Delta$ (輸出繼電器)，M $\Delta\Delta\Delta\Delta\Delta$ (內部輔助繼電器)，S $\Delta\Delta\Delta$ (步進繼電器)和 TR $\Delta\Delta\Delta$ (暫存繼電器)，其中輸出繼電器 Y $\Delta\Delta\Delta$ 之狀態會被送到外界(端子台上之輸出點)去。

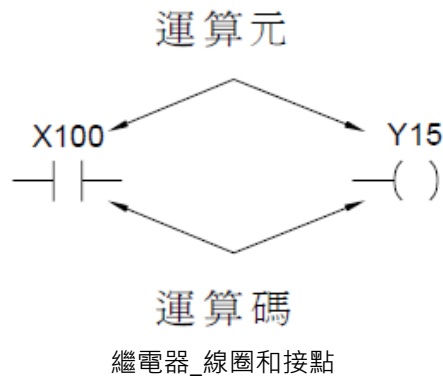
③. 母線(Origin)

階梯圖最左側之起始線。

④. 元件(Element)

元件(即線圈或接點)為組成階梯圖程式之最基本單位。

元件之表示分為兩部分，一為元件之符號，稱之為運算碼(OP Code)，另一為數字部分，稱之為運算元(Operand)，如下圖所示。



M SERIES 系列之元件有下列 8 種：

元件類別	符號	備註
A 接點 (常開接點)		□可為 X、Y、M、S、T、C (請參閱 2.2 節說明)
B 接點 (常閉接點)		
上微分接點		□可為 X、Y、M、S
下微分接點		
開路接點		
短路接點		
輸出線圈		□可為 Y、M、S
倒相輸出線圈		

M SERIES 系列之元件

註：X、Y、M、S、T、C 等接點或線圈範圍請參閱第 2.2 節、其元件特性請參閱第 4.2 節。

另外尚有一個特殊順序指令(FOn) 亦屬元件之一種，請參考第 5.1.4 節"功能輸出 FO"之說明。

⑤. 節點(Node)

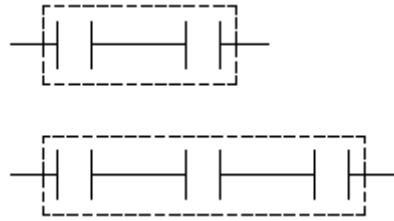
任兩個或兩個以上元件相連接之點。

⑥. 區塊(Block)

兩個或兩個以上之元件組合成之回路稱之。

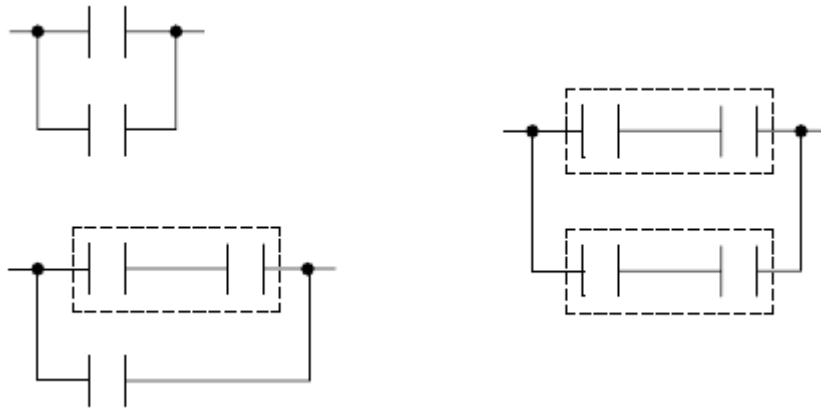
基本之區塊有兩種：

- 串聯區塊：兩個或兩個以上元件串接而成之單列回路。



串聯區塊

- 並聯區塊：由元件或串聯區塊並聯組成之平行(矩形)封閉回路稱之。

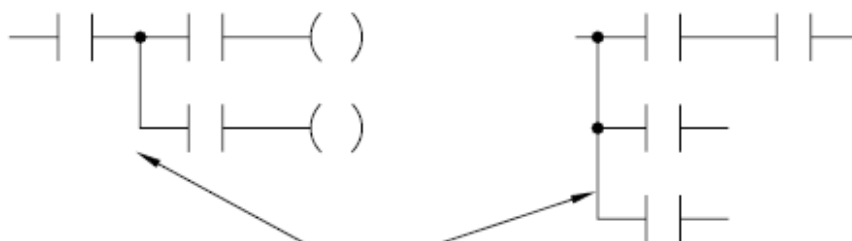


並聯區塊

註：由元件、串聯區塊及並聯區塊等三種基本單元可以組成許多更複雜之串並聯區塊回路。

⑦. 分歧(Branch)

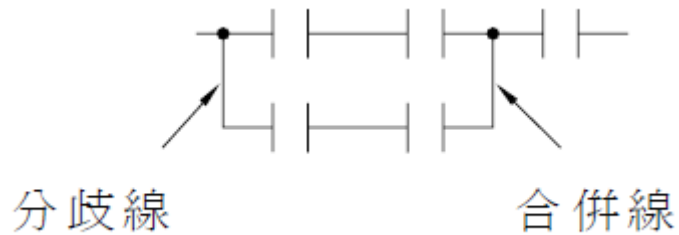
任一網路中之垂直線右方有兩列或兩列以上之回路連接，此即為分歧，而此垂直線即稱分歧線或稱為支線。



分歧

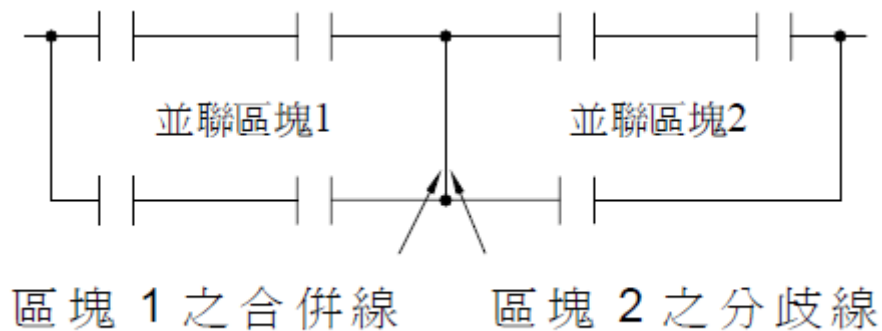
分歧和分歧線

分歧線之右邊若有另一垂直線將分歧之兩列回路予以合併(此垂直線稱為合併線)，則此回路即形成一封閉之回路(形成並聯區塊)，此回路即非分歧回路。



分歧線及合併線

若垂直線的左、右邊均有兩列以上之回路連接，則此垂直線既是合併線，又是分歧線。
如下圖：



同時為分歧線和合併線

⑧. 網路(Network)

由元件、分歧、區塊組成一能執行特定功能之回路，即稱為網路。網路是階梯圖程式中能執行完整功能之基本單位，而階梯圖程式就是由一連串網路所組成。網路之起始必須由母線開始，任一無垂直線連接之兩列回路即屬不同之兩個網路(有垂直線相連者則屬於同一網路)。依此法則，如階梯圖程式範例可區分成網路 1~3 三個網路。

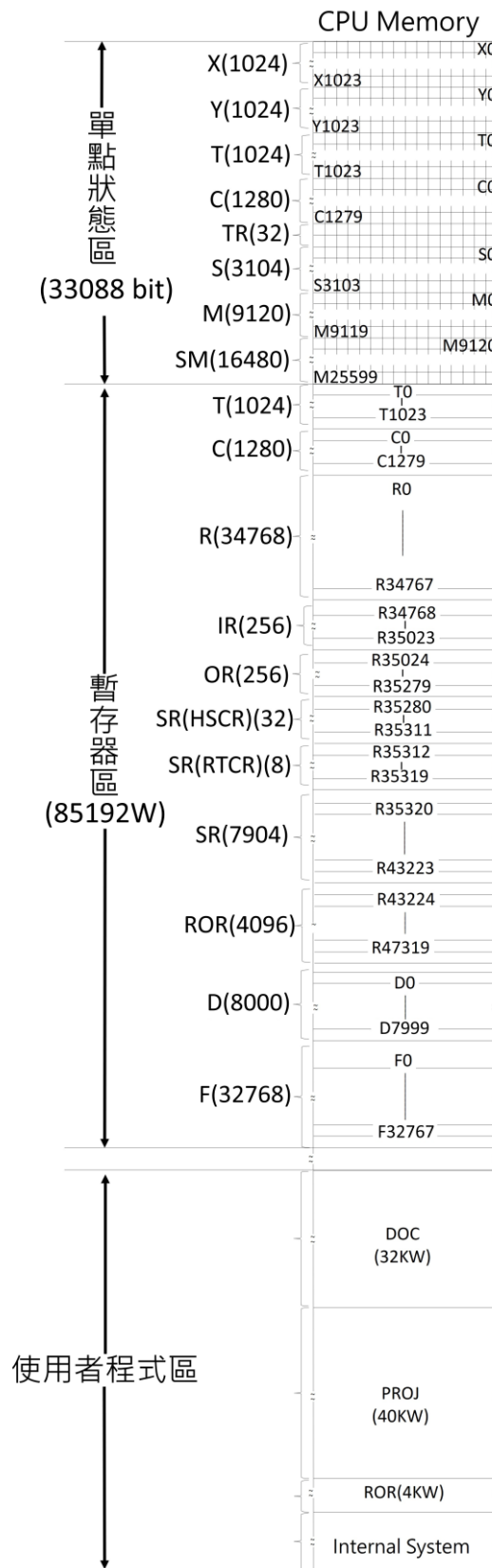
2

PLC 內部之記憶體配置及其單點(數位)與暫存器明細

2-1	M SERIES 記憶體配置	2-2
2-2	單點 (Digital) 及暫存器之配置	2-3
2-3	CPU 模組特殊繼電器明細	2-6
2-4	CPU 模組特殊暫存器明細	2-14
2-5	Motion 模組特殊繼電器明細	2-33
2-6	Motion 模組特殊暫存器明細	2-35
2-7	Motion 錯誤與警報暫存器相關明細	2-36

※※M Serial PLC 在設計上,一般暫存器區 R(34768 個 words)可以用間接定址存取,彈性很大,但也很容易因為間接定址參數值使用不當而造成誤寫資料的問題.至於唯讀暫存器區 ROR(4096 個 words), M Serial PLC 不允許間接定址進行存取, 所以若是重要的參數值,建議可以利用 ROR 唯讀暫存器區域,使用程式指令進行讀寫,避免間接定址參數誤設而導致的後續問題。

2-1 M SERIES 記憶體配置



PLC 記憶體配置圖

2-2 單點 (Digital) 及暫存器之配置

- 本配置為出廠時之設定

項 目			規 格		備 註
單 點 《 B I T 狀 態 》	X	輸入接點(DI) (最大點數 2048 點)		X0 ~ X1023 (1024)	對應至外界數位輸入
	Y	輸出繼電器(DO) (最大點數 2048 點)		Y0 ~ Y1023 (1024)	對應至外界數位輸出
	TR	暫存繼電器		TR0 ~ TR31(32) (系統保留使用)	
	M	內部繼電器		M0 ~ M9119 (9120)	M0~M9119 可規劃為保持或非保持型繼電器。
		特殊繼電器		M9120 ~ M25599 (16480)	
	S	步進繼電器		S0 ~ S3103 (3104)	S0 ~ S3103 可規劃為保持或非保持型繼電器。
	T	計時器 `計時到` 狀態接點		T0 ~ T1023 (1024)	
暫 存 器 《 W O R D 資 料 》	TMR	計時器 現在 值 暫存器	0.001S 時基	T0 ~ T255 (256) *	T0 ~ T1023 可彈性 規劃各時基之數量
			0.01S 時基	T256 ~ T511 (256) *	
			0.1S 時基	T512 ~ T767 (256) *	
			1S 時基	T768 ~ T1023 (256) *	
	CTR	計數器 現在 值	16 位元	C0 ~ C1023 (1024)	可規劃為保持型或非 保持型

	暫存器	32 位元	C1024 ~ C1279 (256)				可規劃為保持型或非保持型
HR DR	資料暫存器	保持型	R0 ~ R14999 (15000) * 註: 可規劃為非保持型 D0 ~ D7999 (8000)				
		非保持型	R15000 ~ R34767 (19768)				
	HR ROR	保持型	R43224 ~ R47319 (4096) * 註: 無被規劃為 ROR 時,可當一般暫存器使用(可讀、寫)				
		唯讀暫存器 (ROR)	R43224 ~ R47319 可規劃為唯讀暫存器, 出廠設定為 0*				ROR 存放在 ROR 專區,不佔用程式容量
		檔案暫存器	F0 ~ F32767(32768) * 註: 需透過專用指令存取				
IR+OR (總計 256 點)	輸入暫存器(AI)		R34768 ~ R35023 (256)				對應至外界類比輸入
	輸出暫存器(AO)		R35024 ~ R35279 (256)				對應至外界類比輸出
	SR		R35280 ~ R43223 (7944)				
〈特殊暫存器〉	0.1ms 定時器 HST 暫存器		R35451~R35466(16)				
	1ms 定時器 STM 暫存器		R35435~R35442(8)				
	10ms 定時器 LTM 暫存器		R35443~R35450(8)				
	0.1ms 定時 HSTA 循環計數		DR35467				
	高速計數器暫存器		R35280~R35311(32)				
	萬年曆暫存器		R35312 (秒)	R35313 (分)	R35314 (時)	R35315 (日)	
			R35316 (月)	R35317 (年)	R35318 (週)	R35319 (時 + 分)	
	FR	檔案(File)暫存器	F0~F32767(32768)				

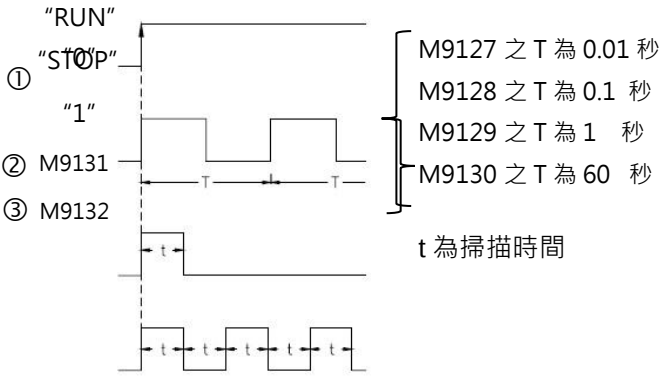
	XR	指標(Index)暫存器	V : R43214 Z : R43216 P0 ~ P9 : R43194, R43196, R43198, R43200, R43202, R43204, R43206, R43208, R43210, R43212	
--	----	--------------	--	--

單點 (Digital) 及暫存器之配置

註：非保持型繼電器或暫存器，在斷電再開機或 PLC 由 STOP→RUN 時會先被清為 0，而保持型則保持原來（斷電前或 STOP 時）狀態。

2-3 CPU 模組特殊繼電器明細

繼電器號碼	功用/標籤簡寫	說明
1.停機，禁止控制		
M9120	緊急停機控制 EMERGENCY_STOP_CTRL	1 時 PLC 停止,
2.抑制，清除控制		
M9121	保留	
M9122	抑制記憶體保持選擇 DISABLE_STATUS_RETN_CTRL	1 時抑制
M9123	非保持型繼電器清除 CLR_NON_RETENT_RELAY	1 時清除
M9124	保持型繼電器清除 CLR_RETENT_RELAY	1 時清除
M9125	非保持型暫存器清除 CLR_NON_RETENT_REG	1 時清除
M9126	保持型暫存器清除 CLR_RETENT_REG	1 時清除

3.脈波信號		
M9127 M9128 M9129 M9130	0.01 秒週期脈波 CLK_PULSE_0_01S 0.1 秒週期脈波 CLK_PULSE_0_1S 1 秒週期脈波 CLK_PULSE_1S 60 秒週期脈波 CLK_PULSE_60S	 ① “RUN” ② “STOP” ③ “1” M9127 之 T 為 0.01 秒 M9128 之 T 為 0.1 秒 M9129 之 T 為 1 秒 M9130 之 T 為 60 秒 t 為掃描時間
M9131 M9132 M9133	啟動（第一次掃描）脈波② CLK_PULSE_INIT 掃描週期波③ CLK_PULSE_SCAN PLC 工作模式 PLC_WORKING_MODE	=0,PLC 工作在 STOP 模式 =1,PLC 工作在 RUN 模式
4.錯誤訊息		
M9134	系統異常警示 CPU_ABNL_WARNING	1 :表示未接擴充機或點數超過
5.Port1~Port2 控制		
M9135 M9136 M9137	通訊埠 1 工作指示 COM_BUSY_P1 通訊埠 1 工作指示 COM_DN_P1 通訊埠 1 通訊指示 COM_STATUS_P1	0: 通訊埠 1 被佔用。 1: 通訊埠 1 Ready。 1: 完成 FUN151(CLINK)之所有通訊交易，只 ON 一個掃描。 1 通訊埠 1 接收到並回應一筆通訊信息
M9138 M9139 M9140	通訊埠 2 工作指示 COM_BUSY_P2 通訊埠 2 工作指示 COM_DN_P2 通訊埠 2 通訊指示 COM_STATUS_P2	0: 通訊埠 2 被佔用。 1: 通訊埠 2 Ready。 1: 完成 FUN151(CLINK)之所有通訊交易，只 ON 一個掃描。 1: 通訊埠 2 接收到並回應一筆通訊信息
6.HSC0~HSC7 控制		
M9141	HSC0 軟體遮沒 HSC0_MSK	1: 遮沒
M9142	HSC0 軟體清除 HSC0_CLR	1: 清除
M9143	HSC1 軟體遮沒 HSC1_MSK	1: 遮沒

M9144	HSC1 軟體清除 HSC1_CLR	1: 清除
M9145	HSC2 軟體遮沒 HSC2_MSK	1: 遮沒
M9146	HSC2 軟體清除 HSC2_CLR	1: 清除
M9147	HSC3 軟體遮沒 HSC3_MSK	1: 遮沒
M9148	HSC3 軟體清除 HSC3_CLR	1: 清除
M9149	保留	
7.Port3~Port4 控制		
M9149	通訊埠 3 工作指示 COM_BUSY_P3	0: 通訊埠 3 被佔用。 1: 通訊埠 3 Ready。
M9150	通訊埠 3 工作指示 COM_DN_P3	1: 完成 FUN151(CLINK)之所有通訊交易，只 ON 一個掃描。
M9151	通訊埠 3 通訊指示 COM_STATUS_P3	1 通訊埠 3 接收到並回應一筆通訊信息
M9152	通訊埠 4 工作指示 COM_BUSY_P4	0: 通訊埠 4 被佔用。 1: 通訊埠 4 Ready。
M9153	通訊埠 4 工作指示 COM_DN_P4	1: 完成 FUN151(CLINK)之所有通訊交易，只 ON 一個掃描。
M9154	通訊埠 4 通訊指示 COM_STATUS_P4	1: 通訊埠 4 接收到並回應一筆通訊信息
M9155 ~M9157	保留	

8.通訊/計時/計數控制		
M9158	計時器計時到後 CV 值的處理 HST_TIME_UP_MODE	0: “計時到”後，其 CV 值繼續計時，直到上限為止。 1: “計時到”後，其 CV 值停在 PV 值，不再增加（使用者可於程式中“計時器”指令執行前設定 M9158 狀態，而能多重或動態選擇計時器之計時模式）。
M9159	計數器“計數到”後，其 CV 值之計數模式選擇	0: “計數到”後，CV 值繼續計數到上限為止。 1: “計數到”後，CV 值計數不再增加（CV 值停於 PV 值）（使用者可於程式中“計數器”指令執行前，設定 M1973 狀態，而能多重或動態選擇計數器之計數模式）。
M9160	電子凸輪跨 0 度功能選擇 CAM_FUNC_SELECT	M9160=1: FUN112(BKCOMP)指令上限值小於下限值時，可執行(例如上限值為 10°，下限值為 350°，當目前角度為 350°~10°時，該比較位元為 1) M9160=0: 若上下限設定值中，上限值小於下限值，則限值錯誤旗號“ERR”設為 1，且該組比較輸出為 0。
M9161	高速脈波輸出停止選擇 HSPSO_STOP_SELECT	
M9162	MODBUS 規劃更新 MODBUS_UPDATE	
M9163	COM 設定更新 COM_UPDATE	
M9164	網路介面重啟 ETH_UPDATE	
M9165	啟用 DHCP ETH_DHCP_ENABLE	
M9166	1 ms 定時計時器 STM 0 控制 STM0_CTRL	
M9167	1 ms 定時計時器 STM 1 控制 STM1_CTRL	
M9168	1 ms 定時計時器 STM 2 控制 STM2_CTRL	
M9169	1 ms 定時計時器 STM 3 控制 STM3_CTRL	

M9170	10 ms 定時計時器 LTM 0 控制 LTM0_CTRL	
M9171	10 ms 定時計時器 LTM 1 控制 LTM1_CTRL	
M9172	10 ms 定時計時器 LTM 2 控制 LTM2_CTRL	
M9173	10 ms 定時計時器 LTM 3 控 LTM3_CTRL	
M9174	0.1 ms 定時 HST 0 控制 HST0_CTRL	
M9175	0.1 ms 定時 HST 1 控制 HST1_CTRL	
M9176	0.1 ms 定時 HST 2 控制 HST2_CTRL	
M9177	0.1 ms 定時 HST 3 控制 HST3_CTRL	
M9178	0.1 ms 定時 HSTA 循環計數控制 HSTA_CTRL	
8.RTC 控 制		
M9179	RTC 設定 RTC_UPDATE	
M9180	30 秒補正 RTC_30S_ADJUSTMENT	
M9181	RTC 安裝檢知 RTC_INSTALL_CHK	
M9182	設定值錯誤 RTC_SET_VALUE_ERROR	
9.PS0~7 控 制		
M9183	高速脈波輸出(PSO0)指示 PSO0_BUSY	
M9184	高速脈波輸出(PSO1)指示 PSO1_BUSY	
M9185	高速脈波輸出(PSO2)指示 PSO2_BUSY	
M9186	高速脈波輸出(PSO3)指示 PSO3_BUSY	

M9187	高速脈波輸出(PSO0)完成 PSO0_DN	
M9188	高速脈波輸出(PSO1)完成 PSO1_DN	
M9189	高速脈波輸出(PSO2)完成 PSO2_DN	
M9190	高速脈波輸出(PSO3)完成 PSO3_DN	
M9191	高速脈波輸出(PSO4)指示 PSO4_BUSY	
M9192	高速脈波輸出(PSO5)指示 PSO5_BUSY	
M9193	高速脈波輸出(PSO6)指示 PSO6_BUSY	
M9194	高速脈波輸出(PSO7)指示 PSO7_BUSY	
M9195	高速脈波輸出(PSO4)完成 PSO4_DN	
M9196	高速脈波輸出(PSO5)完成 PSO5_DN	
M9197	高速脈波輸出(PSO6)完成 PSO6_DN	
M9198	高速脈波輸出(PSO7)完成 PSO7_DN	

9. 擴充模組使用區域

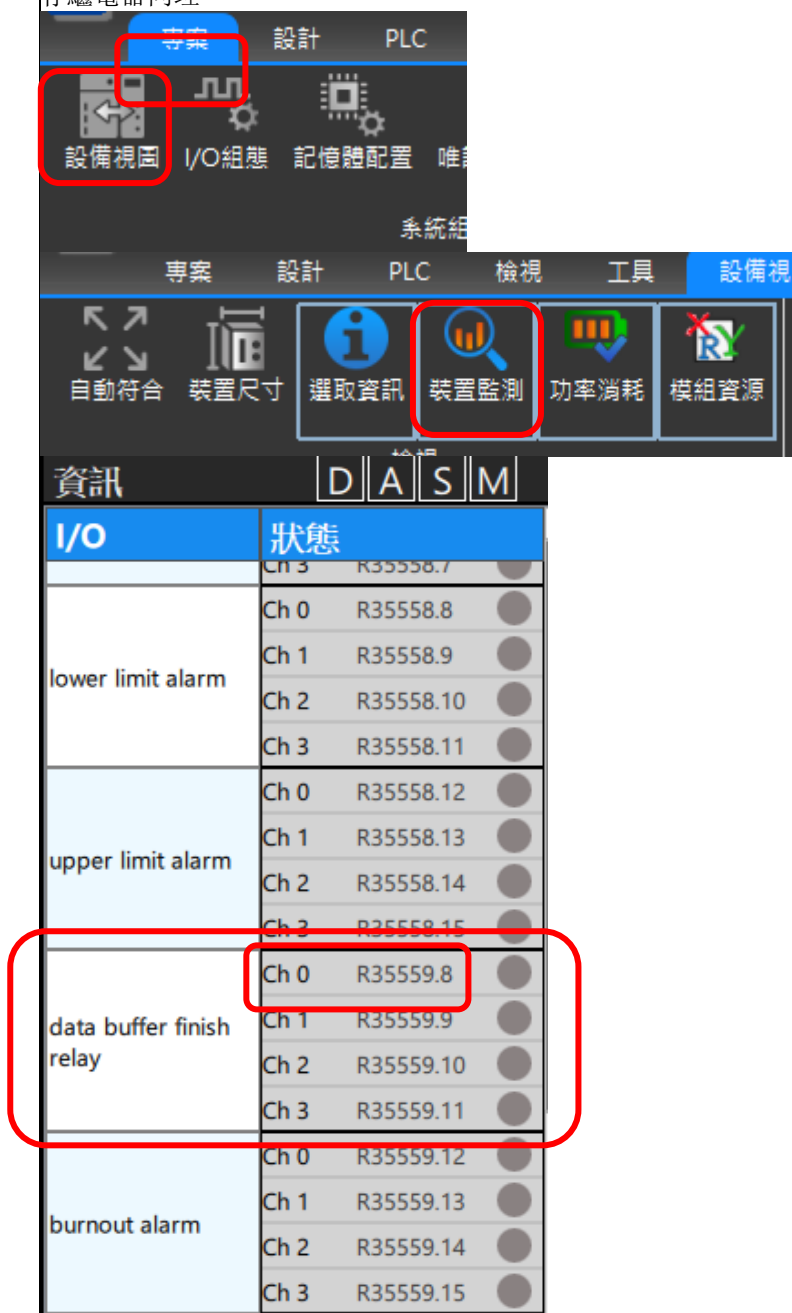
M9199
~
M10511

請參考各擴充模組使用手冊

因為其特殊暫存器的編號與使用者的擴充模組設定相關，故順序沒有固定的編號，可以藉由以下方式得知：

特殊暫存器的編號可以從專案->設備視圖->裝置監測->點選想使用的模組後顯示

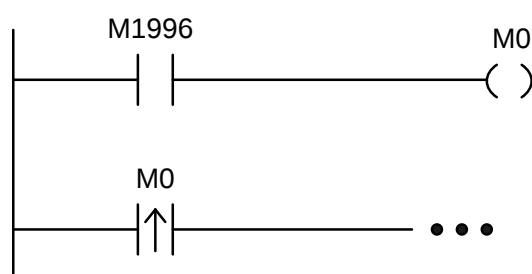
下圖以數據緩存繼電器為例，啟動數據緩存繼電器與觸發數據緩存繼電器同理



M10512 ~ M16095	Motion 相關特殊繼電器	
-----------------------	----------------	--

CPU 模組特殊繼電器明細

※所有特殊繼電器皆不提供上、下微分、接點指令(TU)，有需要對特殊繼電器作微分動作的話，可以採間接方式替代之。(參考下圖)



特殊繼電器微分動作接法

註：特殊繼電器和暫存器中標有▀符號者均為禁止寫入，同時此類繼電器尚禁止 / 抑能控制及強制設定，亦不提供 TU、TD 接點。

2-4 CPU 模組特殊暫存器明細

暫存器號碼 / 系統標籤代號	功用/標籤簡寫	說明
R35280	HSC0 目前計數值 Low word HSC0_CV	
R35281	HSC0 目前計數值 High word HSC0_CV	
R35282	HSC0 計數設定值 Low word HSC0_PV	
R35283	HSC0 計數設定值 High word HSC0_PV	
R35284	HSC1 目前計數值 Low word HSC1_CV	
R35285	HSC1 目前計數值 High word HSC1_CV	
R35286	HSC1 計數設定值 Low word HSC1_PV	
R35287	HSC1 計數設定值 High word HSC1_PV	
R35288	HSC2 目前計數值 Low word HSC2_CV	
R35289	HSC2 目前計數值 High word HSC2_CV	
R35290	HSC2 計數設定值 Low word HSC2_PV	
R35291	HSC2 計數設定值 High word HSC2_PV	
R35292	HSC3 目前計數值 Low word HSC3_CV	
R35293	HSC3 目前計數值 High word HSC3_CV	
R35294	HSC3 計數設定值 Low word HSC3_PV	
R35295	HSC3 計數設定值 High word HSC3_PV	

暫存器號碼 / 系統標籤代號	功用/標籤簡寫	說明
R35296	HSC4 目前計數值 Low word HSC4_CV	
R35297	HSC4 目前計數值 High word HSC4_CV	
R35298	保留	
R35299	保留	
R35300	HSC5 目前計數值 Low word HSC5_CV	
R35301	HSC5 目前計數值 High word HSC5_CV	
R35302	保留	
R35303	保留	
R35304	HSC6 目前計數值 Low word HSC6_CV	
R35305	HSC6 目前計數值 High word HSC6_CV	
R35306	保留	
R35307	保留	
R35308	HSC7 目前計數值 Low word HSC7_CV	
R35309	HSC7 目前計數值 High word HSC7_CV	
R35310	保留	
R35311	保留	

暫 存 器 號 碼 / 系 統 標 籤 代 號	功用/標籤簡寫	說明
R35312	秒 RTC_SECOND	
R35313	分 RTC_MINUTE	
R35314	時 RTC_HOUR	
R35315	日 RTC_DAY	
R35316	月 RTC_MONTH	
R35317	年 RTC_YEAR	
R35318	週 RTC_DAY_OF_WEEK	
R35319	時(High byte) + 分(Low byte) RTC_HOUR_MINUTE	
R35320	PSO0 錯誤碼 PSO0_ERR_CODE	
R35321	PSO1 錯誤碼 PSO1_ERR_CODE	
R35322	PSO2 錯誤碼 PSO2_ERR_CODE	
R35323	PSO3 錯誤碼 PSO3_ERR_CODE	
R35324	PSO0 每步結束之步號 PSO0_DN_STEP_NUM	
R35325	PSO1 每步結束之步號 PSO1_DN_STEP_NUM	
R35326	PSO2 每步結束之步號 PSO2_DN_STEP_NUM	
R35327	PSO3 每步結束之步號 PSO3_DN_STEP_NUM	
R35328	PSO0 輸出頻率 Low word PSO0_CUR_FREQ	-
R35329	PSO0 輸出頻率 High word PSO0_CUR_FREQ	-
R35330	PSO1 輸出頻率 Low word PSO1_CUR_FREQ	-

暫 存 器 號 碼 / 系 統 標 籤 代 號	功用/標籤簡寫	說明
R35331	PSO1 輸出頻率 High word PSO1_CUR_FREQ	-
R35332	PSO2 輸出頻率 Low word PSO2_CUR_FREQ	-
R35333	PSO2 輸出頻率 High word PSO2_CUR_FREQ	-
R35334	PSO3 輸出頻率 Low word PSO3_CUR_FREQ	-
R35335	PSO3 輸出頻率 High word PSO3_CUR_FREQ	-
R35336	PSO0 目前脈衝位置 Low word PSO0_CUR_POS	
R35337	PSO0 目前脈衝位置 High word PSO0_CUR_POS	
R35338	PSO1 目前脈衝位置 Low word PSO1_CUR_POS	
R35339	PSO1 目前脈衝位置 High word PSO1_CUR_POS	
R35340	PSO2 目前脈衝位置 Low word PSO2_CUR_POS	
R35341	PSO2 目前脈衝位置 High word PSO2_CUR_POS	
R35342	PSO3 目前脈衝位置 Low word PSO3_CUR_POS	
R35343	PSO3 目前脈衝位置 High word PSO3_CUR_POS	
R35344	PSO0 待輸出脈衝數值 Low word PSO0_REMAINING_COUNT	
R35345	PSO0 待輸出脈衝數值 High word PSO0_REMAINING_COUNT	
R35346	PSO1 待輸出脈衝數值 Low word PSO1_REMAINING_COUNT	
R35347	PSO1 待輸出脈衝數值 High word PSO1_REMAINING_COUNT	
R35348	PSO2 待輸出脈衝數值 Low word PSO2_REMAINING_COUNT	
R35349	PSO2 待輸出脈衝數值 High word PSO2_REMAINING_COUNT	

暫存器號碼 / 系統標籤代號	功用/標籤簡寫	說明
R35350	PSO3 待輸出脈衝數值 Low word PSO3_REMAINING_COUNT	
R35351	PSO3 待輸出脈衝數值 High word PSO3_REMAINING_COUNT	
R35352	COM1 通訊參數設定 COM_PARAM_P1	設定 Port 1 之 Baud Rate · Data bit...
R35353	COM2 通訊參數設定 COM_PARAM_P2	設定 Port 2 之 Baud Rate · Data bit...
R35354	COM1 與 COM2 連線設定 COM_STN_CHK_P1P2	● R35354 之低位元組定義如下： =1 · Port 1 對外部通訊命令格式(人機/圖控)不檢查站號。 ≠1 · Port 1 對外部通訊命令格式需檢查站號；可作多台 PLC 資料連線。 ● R35354 之高位元組定義如下： =1 · Port 2 對外部通訊命令格式(人機/圖控)不檢查站號。 ≠1 · Port 2 對外部通訊命令格式需檢查站號；可作多台 PLC 資料連線。
R35355	COM1 與 COM2 通訊協定設定 COM_PROTOCOL	設定 Port1、Port2 為 FATEK 或為 Modbus RTU/ASCII 通訊協定
R35356	保留	
R35357	COM1 作為主站時傳送延遲與接收異常偵測時間設定 COM_TX_DELAY_P1	
R35358	COM2 作為主站時傳送延遲與接收異常偵測時間設定 COM_TX_DELAY_P2	
R35359	保留	

暫存器號碼 / 系統標籤代號	功用/標籤簡寫	說明		
		項目	ERR1/ERR2 LED	SR/MASK
R35360	系統錯誤指示(需搭配 LED 做確認) CPU_ERROR	記憶體不足	ON/ON	保留
		初始化錯誤	ON/ON	保留
		系統錯誤	ON/ON	保留
		系統堆棧錯誤	ON/ON	R35361/0x0200
		系統校驗碼錯誤指示	ON/ON	R35361/0x0004
		上電檢測為試運行斷電	ON/ON	R35360/0x0400
		系統校驗碼錯誤指示	ON/ON	R35362/0x3E00
		擴充模組偵測錯誤	ON/OFF	R35360/0x0001
		擴充模組配置檔錯誤	ON/OFF	R35360/0x0002
		擴充模組數量與主機專案不匹配	ON/OFF	R35360/0x0004
		擴充模組 I/O 點數超出範圍	ON/OFF	R35360/0x0008
		擴充模組數量超出範圍	ON/OFF	R35361/0x0100
		運動控制單元佇列錯誤	ON/OFF	R35360/0x0080
		運動控制單元溢位錯誤	ON/OFF	R35360/0x0100
		運動控制單元緊急停機	ON/OFF	R35360/0x0200
		看門狗重啟檢測	ON/OFF	R35361/0x0010
		不合法記憶卡偵測指示	OFF/ON	R35360/0x0010
		記憶卡操作錯誤指示	OFF/ON	R35360/0x0020
		PLC ID 與 PROG ID 不匹配	OFF/ON	R35360/0x0040
		應用程式超出本 CPU 功能	OFF/ON	R35361/0x0800
		系統服務錯誤指示	OFF/ON	R35361/0x8000

R35361 ~R35362	系統狀態指示 CPU_STATUS	BIT0 : PLC 運行或停機狀態 BIT1 : 保留 BIT2 : 系統校驗碼錯誤指示 BIT3 : 記憶卡 Ready 指示 BIT4 : 看門狗重啟檢測 BIT5 : 運動控制單元偵測 BIT6 : PLC ID 保護 BIT7 : 緊急停機 BIT8 : 擴充模組數量超出範圍 BIT9 : 系統堆棧錯誤 BIT10 : 保留 BIT11 : 應用程式超出本 CPU 功能 BIT12 : 保留 BIT13 : 保留 BIT14 : RTC Ready 指示 BIT15 : 系統服務錯誤指示 BIT16 : PLC ID 設置狀態 BIT17 : Program ID 設置狀態 BIT18 : 主程序密碼設置狀態 BIT19 : 子程序密碼設置狀態 BIT20 : PLC 上載密碼設置狀態 BIT21 : PLC 下載密碼設置狀態 BIT22 : CIC 設置狀態 BIT23 : 保留 BIT24 : 保留 BIT25~29 : 系統校驗碼錯誤指示 BIT30 : 指撥開關狀態 BIT31 : 保留
R35363	PLC 站號顯示或設定 PLC_STATION_NUM	● 當暫存器高位元組不等於 55H 時，R35363 之內容顯示此 PLC 之站號 ● 當暫存器 R35363 高位元組等於 55H 時，R35363 之低位元組用來設此 PLC 之站號
R35364	PLC OS Version (MAJOR NO.) PLC_OS_VER_MAJOR	

R35365	PLC OS Version (MINOR NO. + PATCH NO.) PLC_OS_VER_MINOR PLC_OS_VER_PATCH	
R35366	主機型號資訊(UNIT ID + MODEL) MAIN_UNIT_MODEL	
R35367	開機延遲設定(0.01s unit) POWER_ON_DELAY	● 設定 PLC 開機後，延遲此段時間後才作 I/O 檢測，單位為 0.01 秒 (內定 100)
R35368	斷電計數 POWER_OFF_COUNTER	
R35369	保留	
R35370	當次掃描時間 SCAN_TIME_CURRENT	1. 誤差為±1ms 2. PLC 由 STOP→RUN 時復歸再重新計算
R35371	最大掃描時間 SCAN_TIME_MAX	
R35372	最小掃描時間 SCAN_TIME_MIN	
R35373	固定掃描時間 SCAN_TIME_SETTING	-
R35374	擴充模組心跳偵測(Rack 1) EXP_HEARTBEAT_RACK1	
R35375	擴充模組心跳偵測(Rack 2) EXP_HEARTBEAT_RACK2	
R35376	擴充模組心跳偵測(Rack 3) EXP_HEARTBEAT_RACK3	
R35377	擴充模組心跳偵測(Rack 4) EXP_HEARTBEAT_RACK4	
R35378	擴充 AI 點數 EXP_AI_POINTS	
R35379	擴充 AO 點數 EXP_AO_POINTS	
R35380	擴充 DI 點數 EXP_DI_POINTS	
R35381	擴充 DO 點數 EXP_DO_POINTS	
R35382	主機 IP 位址 OCT1(Leading) ETH_IP_OCT1	-
R35383	主機 IP 位址 OCT2 ETH_IP_OCT2	

R35384	主機 IP 位址 OCT3 ETH_IP_OCT3	
R35385	主機 IP 位址 OCT4 ETH_IP_OCT4	
R35386	主機子網路遮罩 OCT1(Leading) ETH_SUBMASK_OCT1	
R35387	主機子網路遮罩 OCT2 ETH_SUBMASK_OCT2	
R35388	主機子網路遮罩 OCT3 ETH_SUBMASK_OCT3	
R35389	主機子網路遮罩 OCT4 ETH_SUBMASK_OCT4	
R35390	主機路由 OCT1(Leading) ETH_GATEWAY_OCT1	
R35391	主機路由 OCT2 ETH_GATEWAY_OCT2	
R35392	主機路由 OCT3 ETH_GATEWAY_OCT3	
R35393	主機路由 OCT4 ETH_GATEWAY_OCT4	
R35394	主機主要 DNS 伺服器 OCT1(Leading) ETH_PRIM_DNS_OCT1	
R35395	主機主要 DNS 伺服器 OCT2 ETH_PRIM_DNS_OCT2	
R35396	主機主要 DNS 伺服器 OCT3 ETH_PRIM_DNS_OCT3	
R35397	主機主要 DNS 伺服器 OCT4 ETH_PRIM_DNS_OCT4	
R35398	主機次要 DNS 伺服器 OCT1(Leading) ETH_SEC_DNS_OCT1	
R35399	主機次要 DNS 伺服器 OCT2 ETH_SEC_DNS_OCT2	
R35400	主機次要 DNS 伺服器 OCT3 ETH_SEC_DNS_OCT3	
R35401	主機次要 DNS 伺服器 OCT4 ETH_SEC_DNS_OCT4	
R35402	Modbus 規劃：Y 起始位址 MODBUS_ADDR_Y	
R35403	Modbus 規劃：Coil 起始位址 MODBUS_COIL_Y	

R35404	Modbus 規劃：對應長度 MODBUS_COIL_TOTALS_Y	
R35405	Modbus 規劃：X 起始位址 MODBUS_COIL_ADDR_X	
R35406	Modbus 規劃：Coil 起始位址 MODBUS_COIL_X	
R35407	Modbus 規劃：對應長度 MODBUS_COIL_TOTALS_X	
R35408	Modbus 規劃：M 起始位址 MODBUS_COIL_ADDR_M	
R35409	Modbus 規劃：Coil 起始位址 MODBUS_COIL_M	
R35410	Modbus 規劃：對應長度 MODBUS_COIL_TOTALS_M	
R35411	Modbus 規劃：S 起始位址 MODBUS_COIL_ADDR_S	
R35412	Modbus 規劃：Coil 起始位址 MODBUS_COIL_S	
R35413	Modbus 規劃：對應長度 MODBUS_COIL_TOTALS_S	
R35414	Modbus 規劃：T 起始位址 MODBUS_COIL_ADDR_T	
R35415	Modbus 規劃：Coil 起始位址 MODBUS_COIL_T	
R35416	Modbus 規劃：對應長度 MODBUS_COIL_TOTALS_T	
R35417	Modbus 規劃：C 起始位址 MODBUS_COIL_ADDR_C	
R35418	Modbus 規劃：Coil 起始位址 MODBUS_COIL_C	
R35419	Modbus 規劃：對應長度 MODBUS_COIL_TOTALS_C	
R35420	Modbus 規劃：R 起始位址 MODBUS_HOLDING_ADDR_R	
R35421	Modbus 規劃：Holding 起始位址 MODBUS_HOLDING_R	
R35422	Modbus 規劃：對應長度 MODBUS_HOLDING_TOTALS_R	
R35423	Modbus 規劃：D 起始位址 MODBUS_HOLDING_ADDR_D	

R35424	Modbus 規劃：Holding 起始位址 MODBUS_HOLDING_D	
R35425	Modbus 規劃：對應長度 MODBUS_HOLDING_TOTALS_D	
R35426	Modbus 規劃：RT 起始位址 MODBUS_HOLDING_ADDR_RT	
R35427	Modbus 規劃：Holding 起始位址 MODBUS_HOLDING_RT	
R35428	Modbus 規劃：對應長度 MODBUS_HOLDING_TOTALS_RT	
R35429	Modbus 規劃：RC 起始位址 MODBUS_HOLDING_ADDR_RC	
R35430	Modbus 規劃：Holding 起始位址 MODBUS_HOLDING_RC	
R35431	Modbus 規劃：對應長度 MODBUS_HOLDING_TOTALS_RC	
R35432	Modbus 規劃：LC 起始位址 MODBUS_HOLDING_ADDR_DRC	
R35433	Modbus 規劃：Holding 起始位址 MODBUS_HOLDING_DRC	
R35434	Modbus 規劃：對應長度 MODBUS_HOLDING_TOTALS_DRC	
R35435	1 ms 定時計時器 STM 0 週期設定 STM0_PV	
R35436	1 ms 定時計時器 STM 0 當前計時 STM0_CV	
R35437	1 ms 定時計時器 STM 1 週期設定 STM1_PV	
R35438	1 ms 定時計時器 STM 1 當前計時 STM1_CV	
R35439	1 ms 定時計時器 STM 2 週期設定 STM2_PV	
R35440	1 ms 定時計時器 STM 2 當前計時 STM2_CV	
R35441	1 ms 定時計時器 STM 3 週期設定 STM3_PV	
R35442	1 ms 定時計時器 STM 3 當前計時 STM3_CV	
R35443	10 ms 定時計時器 LTM 0 週期設定 LTM0_PV	

R35444	10 ms 定時計時器 LTM 0 當前計時 LTM0_CV	
R35445	10 ms 定時計時器 LTM 1 週期設定 LTM1_PV	
R35446	10 ms 定時計時器 LTM 1 當前計時 LTM1_CV	
R35447	10 ms 定時計時器 LTM 2 週期設定 LTM2_PV	
R35448	10 ms 定時計時器 LTM 2 當前計時 LTM2_CV	
R35449	10 ms 定時計時器 LTM 3 週期設定 LTM3_PV	
R35450	10 ms 定時計時器 LTM 3 當前計時 LTM3_CV	
R35451	0.1 ms 定時 HST 0 週期設定 LOW WORD HST0_PV	
R35452	0.1 ms 定時 HST 0 週期設定 HIGH WORD HST0_PV	
R35453	0.1 ms 定時 HST 0 當前計時 LOW WORD HST0_CV	
R35454	0.1 ms 定時 HST 0 當前計時 HIGH WORD HST0_CV	
R35455	0.1 ms 定時 HST 1 週期設定 LOW WORD HST1_PV	
R35456	0.1 ms 定時 HST 1 週期設定 HIGH WORD HST1_PV	
R35457	0.1 ms 定時 HST 1 當前計時 LOW WORD HST1_CV	
R35458	0.1 ms 定時 HST 1 當前計時 HIGH WORD HST1_CV	
R35459	0.1 ms 定時 HST 2 週期設定 LOW WORD HST2_PV	
R35460	0.1 ms 定時 HST 2 週期設定 HIGH WORD HST2_PV	
R35461	0.1 ms 定時 HST 2 當前計時 LOW WORD HST2_CV	
R35462	0.1 ms 定時 HST 2 當前計時 HIGH WORD HST2_CV	
R35463	0.1 ms 定時 HST 3 週期設定 LOW WORD HST3_PV	

R35464	0.1 ms 定時 HST 3 週期設定 HIGH WORD HST3_PV	
R35465	0.1 ms 定時 HST 3 當前計時 LOW WORD HST3_CV	
R35466	0.1 ms 定時 HST 3 當前計時 HIGH WORD HST3_CV	
R35467	0.1 ms HSTA 循環當前計數 LOW WORD HSTA_CV	
R35468	0.1 ms HSTA 循環當前計數 HIGH WORD HSTA_CV	
R35469- R35478	用來指定燒錄於 SD 卡中要讀回的資料暫存器；必須在執行燒錄 SD 卡前先將此區域設定好。其後每次開機時，會根據燒錄在 SD 卡之 SR18 ~ SR27 來執行動作。 SD_GROUP_FLAG SD_GROUP_COUNT SD_GROUP_LEN1 SD_GROUP_ADDR1 SD_GROUP_LEN2 SD_GROUP_ADDR2 SD_GROUP_LEN3 SD_GROUP_ADDR3 SD_GROUP_LEN4 SD_GROUP_ADDR4	使用 ROM Pack 儲存 Ladder 程式與資料暫存器時，利用此表格決定那些暫存器需燒錄並在開機時由 ROM Pack 讀回作初始化
R35479	SD 卡暫存器讀回控制，決定了開機時是否將 PACK 中的資料暫存器讀回 SD_GROUP_LOAD_FLAG	=5530H：開機時不會將有燒錄至 ROM Pack 之資料暫存器讀回 =其它值：每次開機時，有燒錄至 ROMPack 之資料暫存器其內容會被初始化為燒錄時之值
R35480	試俾修改模式或燒錄 SD 卡命令與狀態 SD_STATE	
R35481	Fatek binary server 的使用者自定義 TCP 埠 ETH_FATEK_CUSTOM_PORT	
R35482	Modbus TCP server 的使用者自定義 TCP 埠 ETH_MODBUS_CUSTOM_PORT	

R35483	iMonitor connection status IMONITOR_STATUS	0: 離線 1: 在線 2: 正在連線中 其他: 錯誤碼
R35484- R35486	主機 MAC 碼 SYS_MAC1 SYS_MAC2 SYS_MAC3	
R35487 - R35643	系統保留使用	
R35644	SD 運行資訊字組 High byte : 狀態碼 Low byte : 操作碼 SD_OPERATION_STATUS	
R35645	內建 AI CH 0 讀值(M2 機型) PLC_AI0	
R35646	內建 AI CH 1 讀值(M2 機型) PLC_AI1	
R35647	PSO4 錯誤碼 PSO4_ERR_CODE	
R35648	PSO5 錯誤碼 PSO5_ERR_CODE	
R35649	PSO6 錯誤碼 PSO6_ERR_CODE	
R35650	PSO7 錯誤碼 PSO7_ERR_CODE	
R35651	PSO4 每步結束之步號 PSO4_DN_STEP_NUM	
R35652	PSO5 每步結束之步號 PSO5_DN_STEP_NUM	
R35653	PSO6 每步結束之步號 PSO6_DN_STEP_NUM	
R35654	PSO7 每步結束之步號 PSO7_DN_STEP_NUM	

R35655	PSO4 輸出頻率 Low word PSO4_CUR_FREQ	
R35656	PSO4 輸出頻率 High word PSO4_CUR_FREQ	
R35657	PSO5 輸出頻率 Low word PSO5_CUR_FREQ	
R35658	PSO5 輸出頻率 High word PSO5_CUR_FREQ	
R35659	PSO6 輸出頻率 Low word PSO6_CUR_FREQ	
R35660	PSO6 輸出頻率 High word PSO6_CUR_FREQ	
R35661	PSO7 輸出頻率 Low word PSO7_CUR_FREQ	
R35662	PSO7 輸出頻率 High word PSO7_CUR_FREQ	
R35663	PSO4 目前脈衝位置 Low word PSO4_CUR_POS	
R35664	PSO4 目前脈衝位置 High word PSO4_CUR_POS	
R35665	PSO5 目前脈衝位置 Low word PSO5_CUR_POS	
R35666	PSO5 目前脈衝位置 High word PSO5_CUR_POS	
R35667	PSO6 目前脈衝位置 Low word PSO6_CUR_POS	
R35668	PSO6 目前脈衝位置 High word PSO6_CUR_POS	
R35669	PSO7 目前脈衝位置 Low word PSO7_CUR_POS	
R35670	PSO7 目前脈衝位置 High word PSO7_CUR_POS	
R35671	PSO4 待輸出脈衝數值 Low word PSO4_REMAINING_COUNT	
R35672	PSO4 待輸出脈衝數值 High word PSO4_REMAINING_COUNT	
R35673	PSO5 待輸出脈衝數值 Low word PSO5_REMAINING_COUNT	
R35674	PSO5 待輸出脈衝數值 High word PSO5_REMAINING_COUNT	

R35675	PSO6 待輸出脈衝數值 Low word PSO6_REMAINING_COUNT	
R35676	PSO6 待輸出脈衝數值 High word PSO6_REMAINING_COUNT	
R35677	PSO7 待輸出脈衝數值 Low word PSO7_REMAINING_COUNT	
R35678	PSO7 待輸出脈衝數值 High word PSO7_REMAINING_COUNT	
R35679	MQTT 連接狀態	MQTT 連線成功 = 0, MQTT 版本不支援 = 1, MQTT 描述符不匹配 = 2, MQTT 伺服器拒絕存取 = 3, MQTT 帳號密碼錯誤 = 4, MQTT 身分驗證失敗 = 5, MQTT 連線已關閉 = 256, MQTT 連接逾時 = 257 MQTT 記憶體不足 = 258
R35680 - R35760	系統內部使用	
R35761	可動態更改高速脈波輸出頻率 HSPO_FREQ_CTRL	High Byte : 保留 Low Byte : 5AH
R35762	COM2 高速網路啟用/模式	Low Bit=55H, 高速僕站啟用 =Others, 關閉 High Bit=55H, 高速主站通訊一次表格循環 =Others, 連續

R35763	COM2 接收/傳送超時時間設定(高速) COM_RX_TX_TIME_P2H	當高位元組之值非 56H 時，系統會根據 R4161 通訊參數設定而產生適當設定值，使用者不必設定 當高位元組之值為 56H 時，低位元組保留給當系統設定不符合使用時之人工設定。 R4160[5-0]：保留 R4160[7-6]：設定 TX TMO, =0, 500us ; =1, 700us ; =2, 900us ; =3, 1100us FBs 高速連線 Baud Rate 與自動設定 Rx/Tx time out 值如下，供參考 Baud Rate = 38400, Rx = 8 bit time , Tx = 700us Baud Rate = 153600, Rx = 16 bit time , Tx = 700us Baud Rate = 614400, Rx = 20 bit time , Tx = 700us
R35764	FUN30 系統類比值位元(Type) PID_AI_RESOLUTION	=0, 12bit =1, 14bit =Other, Nbit
R35765	FUN30 增益常數 PID_GAIN	
R35766	FUN147 Gp0 線(向量)速度顯示暫存器 Low Word	
R35767	FUN147 Gp0 線(向量)速度顯示暫存器 High Word	
R35768	FUN147 Gp1 線(向量)速度顯示暫存器 Low Word	
R35769	FUN147 Gp1 線(向量)速度顯示暫存器 High Word	
R35770	FUN147 Gp2 線(向量)速度顯示暫存器 Low Word	
R35771	FUN147 Gp2 線(向量)速度顯示暫存器 High Word	

R35772	FUN147 Gp3 線(向量)速度顯示暫存器 Low Word	
R35773	FUN147 Gp3 線(向量)速度顯示暫存器 High Word	
R35774	FUN147 Gp0 錯誤碼暫存器	
R35775	FUN147 Gp1 錯誤碼暫存器	
R35776	FUN147 Gp2 錯誤碼暫存器	
R35777	FUN147 Gp3 錯誤碼暫存器	
R35778	FUN147 Gp0 定位完成時之步號	
R35779	FUN147 Gp1 定位完成時之步號	
R35780	FUN147 Gp2 定位完成時之步號	
R35781	FUN147 Gp3 定位完成時之步號	
R35782 - R35797	保留	
R35798	COM3 通訊參數設定	
R35799	COM4 通訊參數設定	
R35800	COM3 與 COM4 連線設定	
R35801	COM3 與 COM4 通訊協定設定	
R35802	COM3 與 COM4 回應延遲設定	
R35803	COM3 作為主站時傳送延遲與接收異常偵測時間設定	
R35804	COM4 作為主站時傳送延遲與接收異常偵測時間設定	

R35805 - R35821	保留	
R35822- R35871	擴充模組校正保留暫存器	
R35872 -R36871	擴充模組狀態起始暫存器	
R36872 -R36879	TEST RUN 保留暫存器(唯讀)	
R36880~ R36979	Motion 相關特殊暫存器	
R36980-	運動軸資料起始(唯讀)	
R43194- R43213	P0 (R43194),保留(R43195), P1(R43196), 保留(R43197), P2(R43198), ... P9 (R43212) INDEX_P0 INDEX_P1 INDEX_P2 INDEX_P3 INDEX_P4 INDEX_P5 INDEX_P6 INDEX_P7 INDEX_P8 INDEX_P9	
R43214	V INDEX_V	
R43216	Z INDEX_Z	

2-5 Motion 模組特殊繼電器明細

※ 詳細內容請參考 Motion 手冊第 2 章 “運動參數和狀態(特殊暫存器和繼電器區域)”

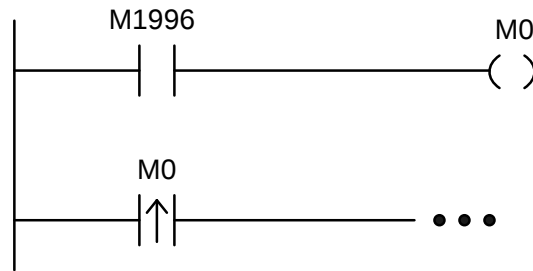
※表格只列軸 1 之暫存器編號，後續每軸暫存器編號為軸 1 之暫存器編號+40*(n-1)，n 為第幾軸

繼電器編號	定義	標籤簡寫
M10520	所有軸使能	ALL_SERVO_ON
M10521	所有軸復位	ALL_SERVO_FAULT_RST
M10522	初始化時寫入所有映射參數	MAPPING_ALL_PARM_AFTER_INIT
M10523	運動控制卡重啟	RESTART_MOTION_CARD
M10524	特殊暫存器自定義 PDO 封包	ENABLE_CUSTOM_PDO
M10600 + 40*(n-1)	軸使能	SERVO_ON
M10601+ 40*(n-1)	軸復位	FAULT_RST
M10602+ 40*(n-1)	軸減速停止	DEC_STOP
M10603+ 40*(n-1)	軸緊急停止	EMG_STOP
M10604+ 40*(n-1)	軸同步主離合器開關	SYNC_MAIN_CLU_ON
M10605+ 40*(n-1)	軸原點開關	ORIG_SIG_IN
M10606+ 40*(n-1)	軸正向極限開關	LIM_POS_IN
M10607+ 40*(n-1)	軸負向極限開關	LIM_NEG_IN
M10608+ 40*(n-1)	軸 Z 相訊號	Z_SIG_IN
M10609+ 40*(n-1)	軸同步主離合器開禁止	SYNC_MAIN_CLU_ON_DIS
M10610+ 40*(n-1)	軸同步主離合器關禁止	SYNC_MAIN_CLU_OFF_DIS
M10611+ 40*(n-1)	軸同步輔助離合器開關	SYNC_AUX_CLU_ON
M10612+ 40*(n-1)	軸同步輔助離合器開禁止	SYNC_AUX_CLU_ON_DIS
M10613+ 40*(n-1)	軸同步輔助離合器關禁止	SYNC_AUX_CLU_OFF_DIS
M10614+ 40*(n-1)	保留	
M10615+ 40*(n-1)	保留	
M10616+ 40*(n-1)	保留	
M10617+ 40*(n-1)	軸探針 1 功能開	PROBE1_EN
M10618+ 40*(n-1)	軸探針 1 功能重置	PROBE1_RST
M10619+ 40*(n-1)	軸探針 2 功能開	PROBE2_EN
M10620+ 40*(n-1)	軸探針 2 功能重置	PROBE2_RST
M10621+ 40*(n-1)	軸同步參數立即生效請求	SYNC_PARM_UP_IMMED
M10622+ 40*(n-1)	軸同步參數下個週期生效請求	SYNC_PARM_UP_NEXT
M10623+ 40*(n-1)	軸同步離合器邊緣觸發緩存開	SYNC_MAIN_CLU_BUF_ON
M10624+ 40*(n-1)	軸同步離合器 OFF 時輸出相位初始化	ECAM_PHASE_RST
M10625+ 40*(n-1)	軸旋轉角擇近	ROT_ANG_NEAR
M10626+ 40*(n-1)	軸旋轉角目標方向	ROT_ANG_DIR
M10627+ 40*(n-1)	軸同步模式開	SYNC_MODE_ON
M10628+ 40*(n-1)	軸暫停當前動作	PAUSE_ON

繼電器編號	定義	標籤簡寫
M10629+ 40*(n-1)	軸原點高速模式開	HOM_SKIP_EN
M10630+ 40*(n-1)	軸指定當前座標	SET_CUR_COOR
M10631+ 40*(n-1)	軸操作模式開	ADV_MODE_ON
M10632+ 40*(n-1)	軸操作模式單位	ADV_MODE_UNIT_PLS
M10633+ 40*(n-1)	軸操作模式絕對位置	ADV_MODE_COOR_ABS_OR_REF_MASK
M10634+ 40*(n-1)	離合器相位顯示	MAIN_CLUTCH_PHASE_INOUT_DISPLAY
M11240 + 40*(n-1)	軸狀態使能	IS_SERVO_ON
M11241+ 40*(n-1)	軸狀態準備就緒	IS_OP_READY
M11242+ 40*(n-1)	軸狀態錯誤中	IS_ERR
M11243+ 40*(n-1)	軸狀態警告中	IS_WARN
M11244+ 40*(n-1)	軸控制中	IS_CTRL
M11245+ 40*(n-1)	軸原點復歸中	IS_HOM_ACT
M11246+ 40*(n-1)	軸原點復歸完成	IS_HOM_DN
M11247+ 40*(n-1)	軸定位模式中	IS_MOV_ACT
M11248+ 40*(n-1)	軸定位模式完成	IS_MOV_DN
M11249+ 40*(n-1)	軸吋動中	IS_JOG_ACT
M11250+ 40*(n-1)	軸吋動完成	IS_JOG_DN
M11251+ 40*(n-1)	軸同步中	IS_SYNC_ACT
M11252+ 40*(n-1)	軸同步完成	IS_SYNC_DN
M11253+ 40*(n-1)	軸速度模式中	IS_VEL_ACT
M11254+ 40*(n-1)	軸速度模式完成	IS_VEL_DN
M11255+ 40*(n-1)	軸扭矩模式中	IS_TORQ_ACT
M11256+ 40*(n-1)	軸扭矩模式完成	IS_TORQ_DN
M11257+ 40*(n-1)	軸正向軟體極限狀態	IS_SOFT_LIM_POS
M11258+ 40*(n-1)	軸負向軟體極限狀態	IS_SOFT_LIM_NEG
M11259+ 40*(n-1)	軸原點極限狀態	IS_ORIG_SIG
M11260+ 40*(n-1)	軸正向極限狀態	IS_LIM_POS
M11261+ 40*(n-1)	軸負向極限狀態	IS_LIM_NEG
M11262+ 40*(n-1)	軸探針 1 觸發狀態	IS_PROBE1_TRIG
M11263+ 40*(n-1)	軸探針 2 觸發狀態	IS_PROBE2_TRIG
M11264+ 40*(n-1)	軸同步參數生效狀態	IS_SYNC_PARM_UPDN
M11265+ 40*(n-1)	軸跟蹤誤差狀態	IS_FOLLOW_ERR
M11266+ 40*(n-1)	軸暫停狀態	IS_PAUSE
M11267+ 40*(n-1)	軸原點模式 Z 向訊號	IS_Z_SIG
M11268+ 40*(n-1)	軸操作模式動作中	IS_ADV_ACT
M11269+ 40*(n-1)	軸操作模式完成	IS_ADV_DN

Motion 模組特殊繼電器明細

※所有特殊繼電器皆不提供上接點指令(TU)，有需要對特殊繼電器作微分動作的話，可以採間接方式替代之。(參考下圖)



特殊繼電器微分動作接法

註：特殊繼電器和暫存器中標有 ▣ 符號者均為禁止寫入，同時此類繼電器尚禁止 / 抑能控制及強制設定，亦不提供 TU、TD 接點。

2-6 Motion 模組特殊暫存器明細

暫存器編號	定義	標籤簡寫
R36880	運動控制器狀態	UNIT_STATE
R36881	運動控制器錯誤碼	UNIT_ERR
R36882	運動程式狀態	UNIT_PROGRAM_STATE
R36883	運動程式錯誤碼	UNIT_ERR_CODE
R36884 - 36903	運動分支流程塊編號	CURRENT_STEP_1 - CURRENT_STEP_20
R36904 - 36923	運動分支流程塊狀態	CURRENT_BLOCK_STATE_1 - CURRENT_BLOCK_STATE_20
R36924 - 36943	運動流程狀態	FLOW_STATE_ID_1 - FLOW_STATE_ID_20
DR36964 - 36970	外部編碼器數值	ENCODER_VALUE_1 - ENCODER_VALUE_4
DR36972	格雷碼編碼器數值	GRAY_CODE_ENCODER_VALUE
DR36974	格雷碼編碼器圈數	GRAY_CODE_ENCODER_TURNS

※表格只列軸 1 之暫存器編號，後續每軸暫存器編號為軸 1 之暫存器編號+150*(n-1)，n 為第幾軸

暫存器編號	定義	標籤簡寫
R36980 + 150*(n-1)	軸屬性	TYPE
R36984 + 150*(n-1)	軸當前控制模式	CTRL_MODE
R37004 + 150*(n-1)	軸錯誤資訊一	ERR_INFO_1
R37005 + 150*(n-1)	軸錯誤資訊二	ERR_INFO_2
R37006 + 150*(n-1)	軸警告資訊一	WARN_INFO_1
R37007 + 150*(n-1)	軸警告資訊二	WARN_INFO_2
R37012 + 150*(n-1)	軸控制中	CTRL_STATUS
R37013 + 150*(n-1)	軸警告中	WARN_STATUS
DR37014 + 150*(n-1)	軸指令座標	CMD_COOR
DR37016 + 150*(n-1)	軸指令速度	CMD_VEL
DR37018 + 150*(n-1)	軸指令位置	CMD_PLS
R37020 + 150*(n-1)	軸當前控制點編號	CUR_PT_OR_ECAM_NUM
DR37021 + 150*(n-1)	軸當前座標	CUR_COOR
DR37023 + 150*(n-1)	軸回授速度	CUR_VEL

暫存器編號	定義	標籤簡寫
DR37025+ 150*(n-1)	軸位置偏差監控	CUR_COOR_DEV
DR37027+ 150*(n-1)	軸驅動器數位輸入	SERVO_DI_VALUE
R37029+ 150*(n-1)	軸當前運動流程編號	CUR_FLOW_NUM
DR37030+ 150*(n-1)	軸同步接點	SYNC_CONT_OUT
R37032+ 150*(n-1)	軸當前扭矩	CUR_TORQ
DR37033+ 150*(n-1)	軸電子凸輪輸入相位	ECAM_IN_PHASE
DR37035+ 150*(n-1)	軸原點位置	ORG_PLS
R37037 - R37039+ 150*(n-1)	軸控制狀態字	CONTROL_STATUS_WORD1
DR37040+ 150*(n-1)	軸主離合器輸出相位	MAIN_CLU_INOUT_PHASE
DR37042+ 150*(n-1)	軸驅動器探針 1 座標	PROBE1_COOR
DR37044+ 150*(n-1)	軸驅動器探針 2 座標	PROBE2_COOR
DR37050+ 150*(n-1)	軸指定座標	SET_COOR_VALUE
DR37052+ 150*(n-1)	軸進階控制模式	SET_ADV_MODE
DR37054+ 150*(n-1)	軸進階控制模式指令	SET_ADV_CMD
DR37056+ 150*(n-1)	軸進階控制模式指令 2	SET_ADV_CMD2
DR37058+ 150*(n-1)	軸進階控制模式指令 3	SET_ADV_CMD3
DR37060+ 150*(n-1)	軸進階控制模式指令 4	SET_ADV_CMD4
DR37062+ 150*(n-1)	軸進階控制模式指令 5	SET_ADV_CMD5
DR37064+ 150*(n-1)	軸進階控制模式指令 6	SET_ADV_CMD6
DR37066+ 150*(n-1)	軸進階控制模式指令 7	SET_ADV_CMD7

2-7 Motion 錯誤與警報暫存器相關明細

暫存器編號	定義	標籤簡寫
R36880	運動控制器狀態	UNIT_STATE
R36881	運動控制器錯誤碼	UNIT_ERR
R36882	運動程式狀態	UNIT_PROGRAM_STATE
R36883	運動程式錯誤碼	UNIT_ERR_CODE
R36884 - 36903	運動分支流程塊編號	CURRENT_STEP_1 - CURRENT_STEP_20

※表格只列軸 1 之暫存器編號，後續每軸暫存器編號為軸 1 之暫存器編號+150*(n-1)，n 為第幾軸

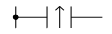
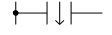
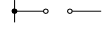
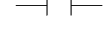
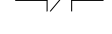
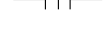
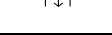
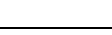
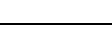
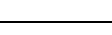
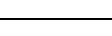
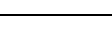
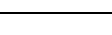
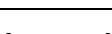
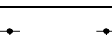
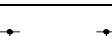
暫存器編號	定義	標籤簡寫
R37004+ 150*(n-1)	軸錯誤資訊一	ERR_INFO_1
R37005+ 150*(n-1)	軸錯誤資訊二	ERR_INFO_2
R37006+ 150*(n-1)	軸警告資訊一	WARN_INFO_1
R37007+ 150*(n-1)	軸警告資訊二	WARN_INFO_2

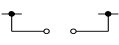
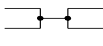
3

M SERIES 指令一覽表

3-1	順序指令一覽表.....	3-1
3-2	應用指令一覽表.....	3-3

3-1 順序指令一覽表

運算元	符號	功能	指令類別
X,Y,M, S,T,C		網路以 A 接點開始	網路起始 指令
		網路以 B 接點開始	
		網路以上微分接點開始	
		網路以下微分接點開始	
		網路以開路接點開始	
		網路以短路接點開始	
X,Y,M, S,T,C		母線或分歧線以 A 接點開始	母線或分歧 線起始指令
		母線或分歧線以 B 接點開始	
		母線或分歧線以上微分接點開始	
		母線或分歧線以下微分接點開始	
		母線或分歧線以開路接點開始	
		母線或分歧線以短路接點開始	
X,Y,M, S,T,C		電路串聯 A 接點	串聯指令
		電路串聯 B 接點	
		電路串聯上微分接點	
		電路串聯下微分接點	
		電路串聯開路接點	
		電路串聯短路接點	
X,Y,M, S,T,C		電路並聯 A 接點	並聯指令
		電路並聯 B 接點	
		電路並聯上微分接點	
		電路並聯下微分接點	

運算元	符號	功能	指令類別
		電路並聯開路接點	
		電路並聯短路接點	
		兩區塊串聯之結合	區塊合併指令
		兩區塊並聯之結合	
Y,M,S	—()	將運算結果送到線圈去	線圈輸出指令
	—(/)	將運算結果倒相後送到線圈去	
Y	—()	將運算結果送到外部輸出線圈，並指定此外部輸出線圈為保持型	
TR		將節點狀態存入暫存接點	節點運作指令
		將暫存之節點狀態取出	
	—↑—	將節點狀態取上微分	
	—↓—	將節點狀態取下微分	
	—/—	將節點狀態作倒相	
	●—(S)	設定線圈	
	●—(R)	清除線圈	

順序指令一覽表

※M SERIES 之順序指令有上列 36 種，所有機種均有此等指令功能。

3-2 應用指令一覽表

M SERIES 之應用指令總共有兩百餘種，加上 D、P 衍生指令，總數超過 300 個指令，而且許多應用指令尚具有多輸入控制（最多 8 個輸入），一個指令最多可組合出 8 種運作模式。實際上 M SERIES 之指令集已不下於大型 PLC 之指令。雖然強大之指令功能有助於複雜、巧妙之控制應用，但對一般小型 PLC 之使用者確實是一大負擔，因此我們將 M SERIES 之應用指令區分為基礎功能指令（在下表中標有“★”號者，只包含常用之 26 種應用指令和 4 個 SFC 指令）以及進階功能指令。

● 一般計時 / 計數指令

指令 號碼	指令名稱	運算元	衍生 指令	功能
★	T nnnn	PV		一般計時器指令（nnnn 為 0 ~ 1023 共 1024 個）
★	C nnnn	PV		一般計數器指令（nnnn 為 0 ~ 1279 共 1280 個）
★ 7	UDCTR	CV,PV	DP	16 位元或 32 位元上/下數計數器

一般計時 / 計數指令表

● 單點運作指令

★ 4	DIFU	D	P	取輸入之上微分狀態並將結果存入 D
★ 5	DIFD	D	P	取輸入之下微分狀態並將結果存入 D
★10	TOGG	D	P	交替 ON/OFF 指令（輸入每 P 一次，D 狀態即變換狀態一次）

單點運作指令表

● 設定 / 清除指令

★	SET	D		設定單點或暫存器之所有位元 (設為 1)
★	RST	D		清除單點或暫存器之所有位元 (設為 0)
114	Z-WR	N		區域設定或區域清除

設定 / 清除指令表

● SFC 指令

★	STP	Snnnn		定義 STEP
★	STPEND			STEP 程式之結束
★	TO	Snnnn		STEP 分歧指令
★	FROM	Snnnn		STEP 合流指令

SFC 指令表

● 數學運算指令

★11	(+)	Sa,Sb,D		Sa 加 Sb 結果存入 D(Sa+Sb→D)
★12	(-)	Sa,Sb,D		Sa 減 Sb 結果存入 D(Sa-Sb→D)
★13	(*)	Sa,Sb,D		將 Sa 乘以 Sb，結果存於 D(Sa×Sb→D)
★14	(/)	Sa,Sb,D		將 Sa 除以 Sb，結果存於 D(Sa/Sb→D)
★15	(+1)	D		將 D 之資料加 1 後結果存回 D(D+1→D)
★16	(-1)	D		將 D 之資料減 1 後結果存回 D(D-1→D)
224	Add F(+)	Sa、Sb、D		Sa 加 Sb 結果存入 D(Sa+Sb→D)
225	Subtract F(+)	Sa、Sb、D		Sa 減 Sb 結果存入 D(Sa-Sb→D)
24	SUM	S,N,D		將 S 開始連續 N 個值取總和後存入 D
25	MEAN	S,N,D		將 S 開始之連續 N 個值平均後存入 D
27	NEG	D		將 D 之值取 2 的補數 (負數) 後存回 D
28	ABS	D		將 D 之值取絕對值後存回 D
31	CRC16	MD,S,N,D		CRC16 計算指令

38	PID2	ID,CH, SR,OR, PR,WR		泛用 PID 運算指令
33	LCNV	Md,S,Ts,D,L	P	線性轉換指令
34	MLC	Rs,Sl,Tx,Ty,Tl, D	P	多段線性轉換指令

● 浮點運算指令

200	I→F	S,D	D P	S 之整數資料變成浮點數後存入 D
201	F→I	S,D	D P	S 之浮點數資料變成整數後存入 D
202	FADD	Sa,Sb,D	P	Sa 及 Sb 相加後結果存入 D(浮點數)
203	FSUB	Sa,Sb,D	P	Sa 及 Sb 相減後結果存入 D(浮點數)
204	FMUL	Sa,Sb,D	P	Sa 及 Sb 相乘後結果存入 D(浮點數)
205	FDIV	Sa,Sb,D	P	Sa 及 Sb 相除後結果存入 D(浮點數)
206	FCMP	Sa,Sb	P	比較浮點數 Sa 和 Sb，再將比較結果送到 FO0 ~ FO2
207	FZCP	S,SU,SL	P	將浮點數 S 和由上限 SU 與下限 SL 所形成之區域作比較，再將比較結果送到 FO0 ~ FO2 去
208	FSQR	S,D	P	將 S 取平方根值後結果存入 D(浮點數)
209	FSIN	S,D	P	將 S 取三角函數 SIN 值後結果存入 D(浮點數)
210	FCOS	S,D	P	將 S 取三角函數 COS 值後結果存入 D(浮點數)
211	FTAN	S,D	P	將 S 取三角函數 TAN 值後結果存入 D(浮點數)
212	FNEG	D	P	將 D 之值取 (負數) 後存回 D(浮點數)
213	FABS	D	P	將 D 之值取絕對值後存回 D(浮點數)
218	FASIN	S,D	P	將 S 取反正弦函數值後結果存入 D(浮點數)
219	FACOS	S,D	P	將 S 取反餘弦數值後結果存入 D(浮點數)
226	MOD	Sa、Sb、D		輸出 Sa/Sb 的

99	TPCTL 2	ID,CH,SR,PR,OR,WR		PID 控制便利指令
----	---------	-------------------	--	------------

數學運算指令表

● 邏輯運算指令

★18	AND	Sa,Sb,D		把 Sa 和 Sb 作邏輯 AND 後存入 D
★19	OR	Sa,Sb,D		把 Sa 和 Sb 作邏輯 OR 後存入 D
35	XOR	Sa,Sb,D		把 Sa 和 Sb 作邏輯 Exclusive NOR 運算後結果存入 D
36	XNR	Sa,Sb,D		把 Sa 和 Sb 作邏輯 Enclusive NOR 運算後結果存入 D

邏輯運算指令表

● 比較指令

★17	CMP	Sa,Sb		比較 Sa 和 Sb 資料，再將比較結果送到 FO0 ~ FO2
37	ZNCMP	S,SU,SL		將 S 和由上限 SU 與下限 SL 所形成之區域作比較，再將比較結果送到 FO0 ~ FO2 去

比較指令表

● 接點型比較指令

170	=	Sa,Sb		將 Sa 和 Sb 作相對等比較
171	>	Sa,Sb		將 Sa 和 Sb 作大於比較
172	<	Sa,Sb		將 Sa 和 Sb 作小於比較
173	< >	Sa,Sb		將 Sa 和 Sb 作不相對等比較
174	> =	Sa,Sb		將 Sa 和 Sb 作大於或等於比較
175	= <	Sa,Sb		將 Sa 和 Sb 作小於或等於比較

接點型比較指令表

● 搬移指令

★ 8	MOV	S,D		將 S 資料搬移至 D(S→D)
★ 9	MOV/	S,D		將 S 資料倒相後搬移至 D(S→D)
40	BITRD	S,N		把 S 中 N 所指位元之狀態取出送到 FO0 去
41	BITWR	D,N		把 INB 輸入之狀態寫入 D 中 N 所指之位元

42	BITMV	S,Ns,D,Nd		把 S 中之 Ns 位元狀態搬至 D 中之 Nd 位元處
43	NBMV	S,Ns,D,Nd		把 S 中 Ns 指定位數(Nibble)搬至 D 中 Nd 所指定之位數
44	BYMV	S,Ns,D,Nd		把 S 中 Ns 指定之 Byte 搬至 D 中 Nd 所指定之 Byte
45	XCHG	Da,Db		把 Da 和 Db 之內容值互換
46	SWAP	D		把 D 中之 High-Byte 和 Low-Byte 之內容值互換
115	DBUF	ID,CH,D		從指定暫存器位址開始存放，擴充模組透過緩存功能收集到的數據
160	RWFR	Sa,Sb,Pr,L,Sa		讀/寫檔案暫存器指令
161	WR-MP	S,Bk,Os,Pr,L,WR		寫入資料至 SD 卡
162	RD-MP	Bk,Os,Pr,L,D		由 SD 卡讀取資料

搬移指令表

● 位移 / 旋轉指令

★ 6	BSHF	D		將 D 資料作一位元之位移 (左或右移一位元後存回 D)
51	SHFL	D,N		把 D 作 N 位元左移 (高位元方向) 後存回 D，移出位元送到 FO0，位移造成之空位以輸入位元填補之
52	SHFR	D,N		把 D 作 N 位元右移 (低位元方向) 後存回 D，移出位元送到 FO0，位移造成之空位以輸入位元填補之
53	ROTL	D,N		把 D 作 N 位元左旋轉 (高位元方向) 後存回 D，旋出之位元送到 FO0
54	ROTR	D,N		把 D 作 N 位元右旋轉 (低位元方向) 後存回 D，旋出之位元送到 FO0

位移/旋轉表

● 數碼變換指令

★ 20	->BCD	S,D		S 資料變成等值之 BCD 值後存入 D
------	-------	-----	---	----------------------

★ 21	->BIN	S,D	D P	S 資料變成等值之二進值後存入 D
34	MLC	Rs,Sl,Tx,Ty,T I,D	P	多段線性轉換指令
55	B->G	S,D	D P	S 之二進制資料轉成格雷碼後存入 D
56	G->B	S,D	D P	S 之格雷碼資料轉成二進制值後存入 D
57	DECOD	S,Ns,NL,D	P	將 S 中 Ns 開始往左 (高位元方向) NL 個位元所形成之二進制數值解碼後，將結果存放於 D 開始之暫存器中
58	ENCOD	S,Ns,NL,D	P	將 S 中 Ns 開始往左 (高位元方向) NL 個單點作高優先或低優先編碼後，將結果存到 D 去
61	→SEC	S,D	P	將 S 開始連續三個暫存器之時分秒時間值變成秒數後存到 D 去
62	→HMS	S,D	P	將 S 之秒數值變成時分秒時間值並將之存入 D 開始之連續三個暫存器中
63	→HEX	S,N,D	P	將 S 開始連續 N 個 ASCII 碼轉為十六進制值存入 D
64	→ASCII	S,N,D	P	將 S 開始連續 N 個十六進制值轉為 ASCII 碼存入 D

數碼變換指令表

● 流程控制指令

★ 0	MC	N		主控回路之開始
★ 1	MCE	N		主控回路之結束
★ 2	SKP	N		跳過回路之開始
★ 3	SKPE	N		跳過回路之結束
	END			程式執行終止點 (除錯用)
22	BREAK		P	FOR 與 NEXT 迴圈之跳出指令
65	LBL	英文 / 數字 1 ~ 6 字		定義運算元所列之文數字為 Label

66	JMP	LBL	P	跳至 LBL 處之程式去執行
165	FLBL	LN		定義運算元所列之文數字為 FLabel
166	FJMP	LBL	P	跳至 FLBL 處之程式去執行
67	CALL	LBL	P	呼叫 LBL 之副程式
68	RTS			副程式之返回指令
69	RTI			中斷服務程式之返回指令
70	FOR	N		Loop 指令之開始點及指定 Loop N 次
71	NEXT			Loop 指令之返回指令
199	TXTDF	LN		階梯程式區塊封鎖功能

流程控制指令表

● I/O 指令

74	IMDIO	D,N	P	立即更新主機上 I/O 點之狀態
83	SPD	S,TI,D		脈波速度偵測指令

I/O 指令表

● 模組功能

258	MODCON F			
-----	-------------	--	--	--

模組功能

● 積算型計時指令

87	T1mS	CV,PV		1 毫秒時基之積算型計時器
88	T10mS	CV,PV		10 毫秒時基之積算型計時器
89	T100mS	CV,PV		100 毫秒時基之積算型計時器

積算型計時指令表

● 監控計時指令

90	WDT	N	P	設定 WDT 之計時時間為 N mS
----	-----	---	---	--------------------

91	RSWDT		P	復歸 WDT 使之重新由 0 開始計時
----	-------	--	----------	---------------------

監控計時指令表

● 高速計數 / 計時指令

92	HSCTR	CN	DP	將 SoC 上硬體高速計數器 HSC0 ~ HSC3 或 HST 之現在值 CV 讀到 PLC 內部對應的 HSC 或 HST 之 CV 暫存器中
93	HSCTW	S,CN,D	DP	將 PLC 內部 HSC0 ~ HSC3 或 HST 之 CV 或 PV 暫存器值寫到 SoC 上的硬體 HSC 或 HST 之 CV 或 PV 暫存器

高速計數 / 計時指令表

● 緩升 / 緩降指令

98	RAMP2	Om,Ta Td,Rt Rc,WR		追蹤型類比輸出緩升/緩降指令
----	-------	----------------------	--	----------------

緩升 / 緩降指令表

● 通訊指令

150	M-BUS	Pt,SR,WR	P	Modbus 通訊便利指令
151	CLINK	Pt,MD,SR,WR	P	永宏泛用通訊便利指令
152	NCR	SR、MD、WR		網路主動通訊
156	CMCTL	ID,Pt,Ts,MD,WR		通訊模組指令

通訊指令表

● 列表指令

103	BT_M	Ts,Td,L	DP	把 Ts 整個內容搬至 Td 去
107	T_FIL	Rs,Td,L	DP	將 Rs 值填入 Td 中之每個位置
113	SORT	S,D,L	DP	排序 (由大而小或由小而大) 便利指令
100	R→T	Rs,Td,L,Pr	DP	把 Rs 值放入 Td 中 Pr 所指之位置去

101	T→R	Ts,L,Pr,Rd	 	把 Ts 中 Pr 所指之位置之值放入 Rd 中
102	T→T	Ts,Td,L,Pr	 	把 Ts 中 Pr 所指位置之內容值放入 Td 中 Pr 所指之位置
105	R-T_S	Rs,Ts,L,Pr	 	由上而下自 Ts 中找出和 Rs 值不同或相同之位置，並將此位置值存入 Pr 中
106	T-T_C	Ta,Tb,L,Pr	 	由上而下自 Ta・Tb 中比較找出值不同或相同之位置，並將此位置值存入 Pr
108	T_SHF	IW,Ts,Td,L,OW	 	將 Ts 取出，位移一個位置後將結果存到 Td 去，而移出之資料送入 OW，騰出之空位以 IW 填入
109	T_ROT	Ts,Td,L	 	將 Ts 取出旋轉一個位置後將結果存到 Td 去
110	QUEUE	IW,QU,L,Pr,OW	 	將 IW 壓下貯列(QUEUE)或自貯列中取出送到 OW 去 (先進先出裝置)
111	STACK	IW,ST,L,Pr,OW	 	將 IW 壓下堆疊(STACK)或自堆疊中取出資料送到 OW 去 (後進先出裝置)
112	BKCOMP	Rs,Ts,L,D	 	將 Rs 之值和列表 Ts 所構成之 L 對上 / 下限值作比較，並將各對比較結果存到 D 所指定之繼電器去 (DRUM 指令)
113	SORT	S,D,L	 	排序 (由大而小或由小而大) 便利指令
114	Z-WR	D,N	 	將 D 為起始 N 長度的暫存器或繼電器，寫入 1 或 0

列表指令表

● 矩陣指令

130	MBCNT	Ms,L,D		計算 Ms 中所有為 1 或為 0 之位元總數，並將之存到 D
120	MAND	Ma,Mb,Md,		將 Ma 和 Mb 作邏輯 AND 運算後將結果存到 Md 去
121	MOR	Ma,Mb,Md,L		將 Ma 和 Mb 作邏輯 OR 運算後將結果存到 Md 去
122	MXOR	Ma,Mb,Md,L		將 Ma 和 Mb 作邏輯 Exclusive OR 運算後將結果存到 Md
123	MXNR	Ma,Mb,Md,L		將 Ma 和 Mb 作邏輯 Enclusive OR 運算後將結果存到 Md
124	MINV	Ms,Md,L		將 Ms 作倒相後將結果存到 Md 去
126	MBRD	Ms,L,Pr		將 Ms 中 Pr 所指位置之位元狀態取出送到 FO0 輸出
128	MBSHF	Ms,Md,L		將 Ms 位移一位元後將結果存到 Md 去，擠出之位元送到 FO0 去，空出之位元則以 INB 之輸入狀態填補

129	MBROT	Ms,Md,L		將 Ms 旋轉一個位元後將結果存到 Md 去並將旋出之位元送到 FO0 去
130	MBCNT	Ms,L,D		計算 Ms 中所有為 1 或為 0 之位元總數，並將之存到 D

矩陣指令表

● NC 定位控制指令

139	HSPWM	PW,OP,RS, PN,OR,WR		硬體脈波寬度調變指令
140	HPSO	Ps,SR,WR		NC 定位控制之高速脈波輸出指令
141	MPARA	Ps,SR		NC 定位控制之參數表指令
142	PSOFF	Ps		NC 定位控制之強制關閉脈波輸出指令
143	PSCNV	Ps,D		將 NC 定位 Ps 位置轉換為 mm · Inch · 或 Deg
147	MHSPO	Gp,SR WR,		NC 定位控制之多軸直線補間定位輸出指令
148	MPG	Sc,Ps,Fo,M r, WR		NC 定位控制之手搖輪定位控制指令
137	ICA	Ps,Is,Fo,Ag		脈衝中斷定角指令
138	ICF	Ps,Is,Fo,Fd		脈衝中斷定長指令
144	HSPWM 2	Pw,Op,HZ, OR		新硬體脈波寬度調變指令

NC 定位控制指令表

● 中斷控制指令

145	EN	LBL		啟動 HSC、HST 及外部 INT 等功能
146	DIS	LBL		關閉 HSC、HST 及外部 INT 等功能

中斷控制指令表

● 運動控制指令

176	MFFlowStart	ID		執行運動流程
177	MFSysStop			停止所有運動流程
178	MFHome	AX		原點復歸

179	MFPointMov	PT	P	位置控制
180	MFJog	AX,D,MD	P	原點復歸
181	MFChgTbPrm	TM、PN、S、PV	P	改變運動控制參數
182	MFFlowPause	ID	P	暫停運動流程
183	MFFlowResume	ID	P	恢復運動流程
184	MFFlowHalt	ID	P	中斷運動流程
185	MFSysRstAlm		P	清除運動流程警報
186	MFFlowStop	ID	P	FLOW 結束
187	MFSysInit		P	Servo 初始化
188	MFSysRCPR	Md、D、Gp	P	運動控制配方讀取
189	MFSysRCPW	Md、D、Gp	P	運動控制配方寫入
191	MFSysCAMR	Md、D、ID、L	P	運動控制凸輪讀取
192	MFSysCAMW	Md、D、ID、L	P	運動控制凸輪寫入
193	MFGearMPG	M、S、N、D、T	P	EtherCAT 手搖輪
194	MFVelCtl	S、V、MX	P	速度控制
195	MFTorqCtl	S、T、MX	P	扭矩控制
196	MFSysCAMGen	ID、MD、D、L	P	生成凸輪
197	MFAxMov	S、MD、PS、V、A、D、SA、SD、DR、BF	P	單軸定位
198	MFMapTbPrm	Gp、N、L	P	寫入映射表
235	MFSysSetVirt	AX	P	將實體軸切換成虛擬軸
236	MFSDOR	S,I,SI,LN,EC,D	P	讀取驅動器的 SDO
237	MFSDOW	S,I,SI,LN,EC,D	P	寫入驅動器的 SDO
238	MFAxLMov	TAX,MD,DR,BF,D,EC	P	多軸直線補間

239	MFAxCirMov	TAX,MD,DR, BF,D,EC	P	多軸圓弧補間
240	MFPATHMov	AX1,AX2,AX3 ,PT,V,TA,TD,S A,SD,EC	P	運行人機教導的路徑

運動控制指令表

4

順序指令說明

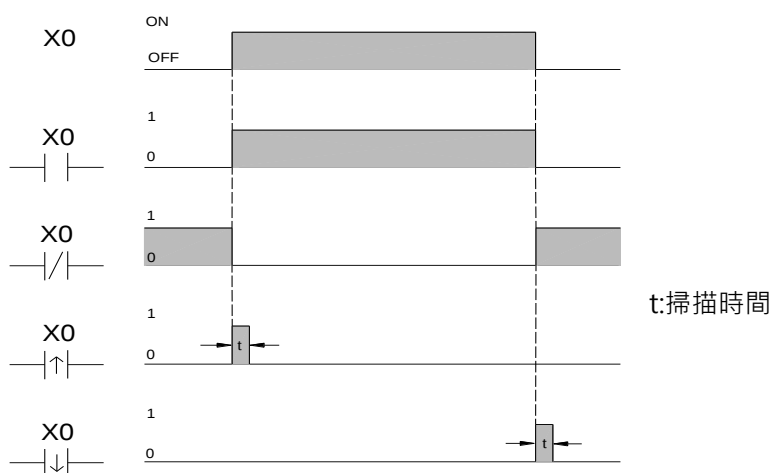
4-1	元件指令特性說明	4-1
4-1-1	A、B、TU、TD 接點元件特性	4-1
4-1-2	開路 (OPEN) 和短路 (SHORT) 接點	4-3
4-1-3	輸出線圈及倒相輸出線圈.....	4-4
4-1-4	保持型輸出線圈.....	4-4
4-1-5	設定線圈及清除線圈.....	4-5

本章僅就順序指令之元件特性及功能作說明。

4-1 元件指令特性說明

4-1-1 A、B、TU、TD 接點元件特性

- 輸入模組上 X0 輸入點之動作
- A 接點元件之狀態
- B 接點元件之狀態
- TU 接點元件之狀態
- TD 接點元件之狀態

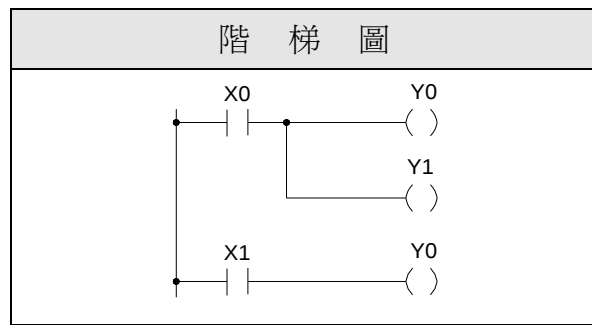


A、B、TU、TD 接點元件特性

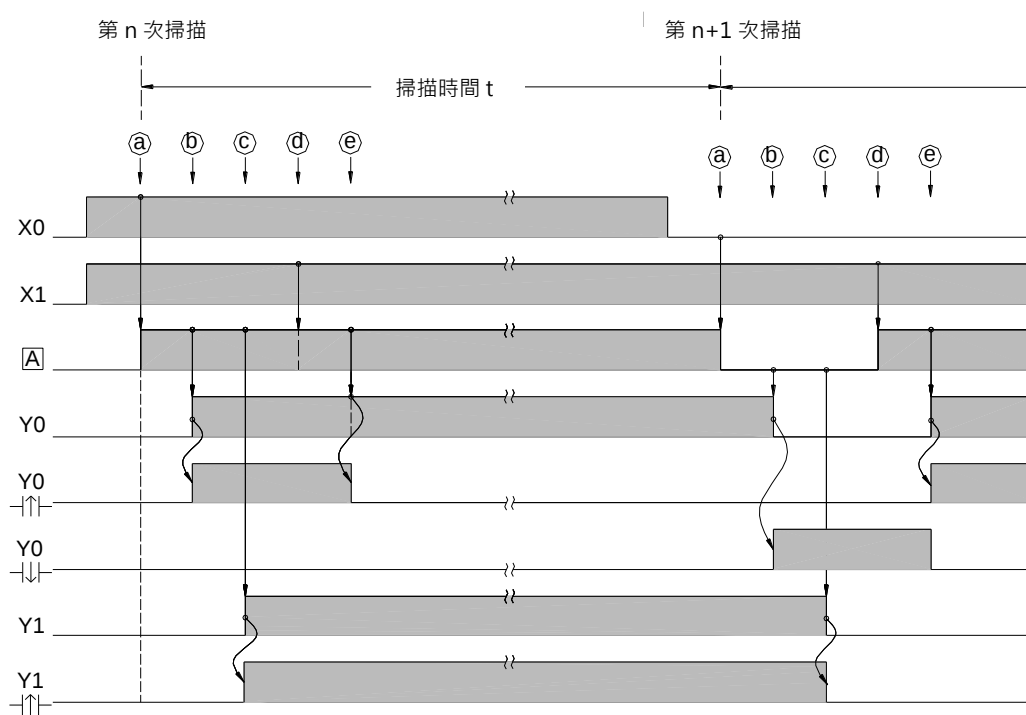
上圖例為由 PLC 輸入模組上之編號 X0 輸入點作如上圖波形之 ON / OFF 動作，在 PLC 階梯圖程式中使用 A 接點、B 接點、TU 接點、TD 接點四種元件，所獲得之狀態波形。

- TU (Transition Up)：即“上微分接點”，本元件在其運算元（本例為 X0）狀態之升緣（0→1）瞬間會產生一個隻“ON”一個掃描時間 t 之單擊脈波。
- TD (Transition Down)：即“下微分接點”，本元件在其運算元（本例為 X0）狀態之降緣（1→0）瞬間會產生一個隻“ON”一個掃描時間 t 之單擊脈波。
- TU、TD 元件對所有 X、Y、M、S、T、C 之接點或線圈狀態變化均會自動產生該接點或線圈所對應之 TU 或 TD 單擊脈波，但線圈之狀態變化若系由“應用指令”以 16 或 32 位元為單位（WY△△△△，WM△△△△，WS△△△△）運作者，則不會產生 TU 或 TD 脈波。

注：M SERIES 之繼電器之 TU、TD 元件之“ON”維持時間是以該元件“ON”條件成立（如 TU 元件由 0→1，TD 元件由 1→0）後之第一次掃描到線圈元件時設為“ON”，一旦設為“ON”後，再掃描到便立刻清為“OFF”，在大部分應用上每一元件在 CPU 解題掃描周期內只會被掃描一次，因此其 TU、TD 元件“ON”的時間必等於 CPU 之掃描時間。但若在一次 CPU 掃描周期內被掃描一次以上者（例如在程式中使用“即時輸入”或“多重線圈輸出”），則其元件之 TU、TD 狀態將在“ON”條件滿足之第一次掃描到時設為“ON”，而在第二次掃描到時立刻清為“OFF”，其“ON”之時間將小於一個 CPU 掃描時間。如下圖例之 Y0 之 TU 即是。因此客戶若須抓取 Y0 之 TU 作觸發運用，就必須將其應用程式插於 Y0 TU “ON”到“OFF”區間內（本例在 b ~ e 間），否則將抓不到任何 Y0 或 TU 之觸發信號。



接點與掃描時間關係範例圖



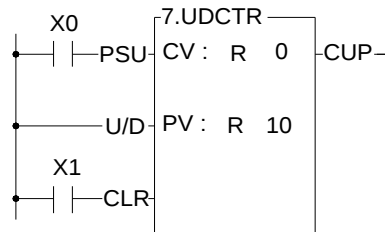
[A]: PLC 內部之累積器

PLC 接點觸發與掃描時間關係圖

- 除 TU、TD 能針對接點或線圈狀態變化自動產生該接點所對應之上微分 (TU)，及下微分 (TD) 之單擊脈波外，M SERIES 尚提供可將節點狀態轉為上下微分單擊脈波而存入線圈內之指令，亦類似 TU、TD 之功能，請參考 FUN4 (DIFU) 及 FUN5 (DIFD) 指令。

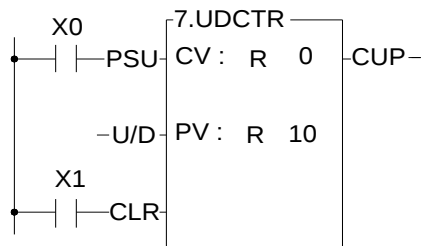
4-1-2 開路 (OPEN) 和短路 (SHORT) 接點

開路和短路接點之狀態是永遠固定不變的，不會受 PLC 之任何運算所影響，此兩接點主要用於階梯圖程式中某些節點狀態需固定不變之處，例如應用指令之輸入控制以作模式選擇。下圖例為利用 SHORT 接點將“上下數計數器” (UDCTR)變成上數計數器之範例。



利用短路生成上數計數器範例階梯圖

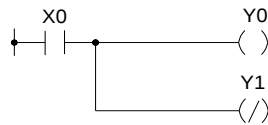
- 因 FUN7 (UDCTR) 指令之“U/D 輸入” 狀態為 1 時 FUN7 作上數計數，為 0 則作下數計數，本例因“U/D” 自母線接 SHORT 接點，因之永遠為 1，故 FUN7 變成上數計數器，相反地，若“U/D” 串接 OPEN 接點則 U/D 永遠為 0，FUN7 即變成下數計數器了。



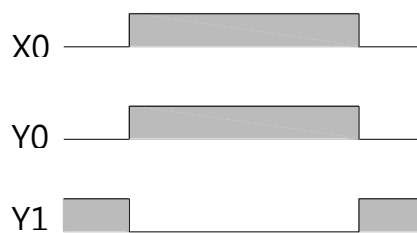
1 利用開路生成下數計數器範例階梯圖

4-1-3 輸出線圈及倒相輸出線圈

輸出線圈系將節點狀態寫入該線圈指令所指定之運算元內，倒相輸出線圈則將節點狀態先經倒相後再寫入該指令所指定之運算元內。其特性如下圖：



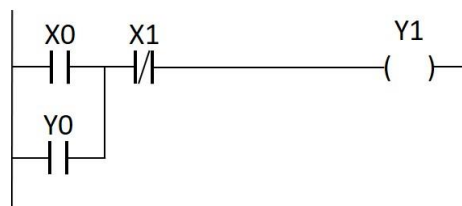
輸出線圈及倒相輸出線圈階梯圖



輸出線圈及倒相輸出線圈

4-1-4 保持型輸出線圈

對內部線圈而言，可設定為保持或非保持兩種（為二分法，如內部線圈 M0 ~ M9119 之 M0 ~ M8519 為非保持，則 M8520 ~ M9119 為保持），但對輸出點，則因實用上不適合以二分法設定保持或非保持，因之大部分 PLC 若需輸出點保持，則必需先將結果送至內部保持線圈，再由此內部保持線圈送至輸出點的間接作法，M SERIES 則提供您在 I/O 組態->輸出停電保持頁面下選擇要保留的輸出點的做法，如下之自保電路：

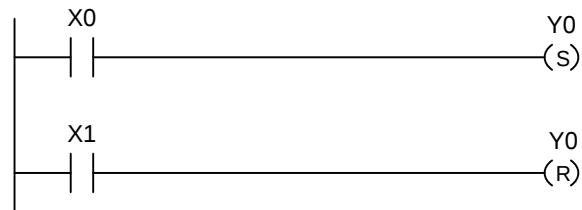


保持型輸出線圈階梯圖

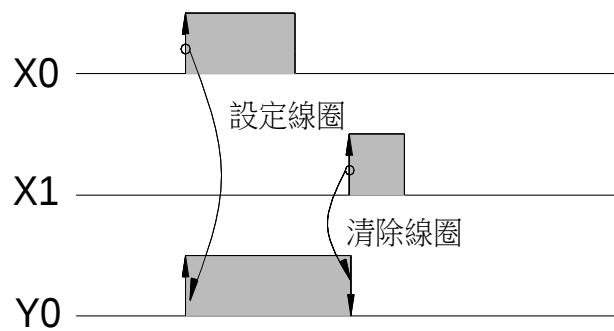
上圖例，X1 為緊急停止開關，使 X0 只 ON 一下再放開(OFF)Y0 將一直保持 ON，斷電後再開或將 PLC STOP 後再 RUN Y0 仍為 ON。但若您使用的是 OUT Y0 指令，只要斷電再開或 PLC STOP 後再 RUN，就必須重新觸發 (X0→ON)，Y0 才會 ON。

4-1-5 設定線圈及清除線圈

設定線圈係將所指定之運算元設定為“ 1 ”，清除線圈則將其所指定之運算元清除為“ 0 ”。特性如下圖所示:



設定線圈及清除線圈範例階梯圖



設定線圈及清除線圈

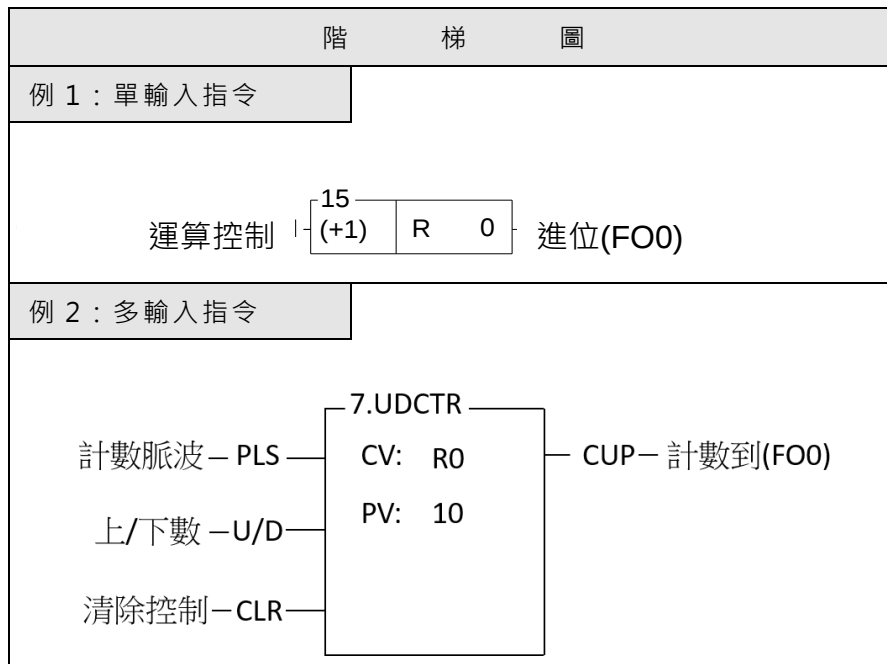
5

應用指令說明

5-1	應用指令之通則	5-1
5-1-1	輸入控制態	5-2
5-1-2	指令號碼與衍生指令	5-3
5-1-3	運算元	5-5
5-1-4	功能輸出 (FO)	5-8
5-2	利用 W 前綴字作字元組及位元存取變換	5-9
5-3	數目系統	5-9
5-3-1	二進制數值及其術語	5-9
5-3-2	M SERIES 之數碼	5-10
5-3-3	數值之範圍	5-10
5-3-4	數值之表示 (初學者請略過本節)	5-10
5-4	利用指標暫存器 (XR) 作間接定址	5-12
5-4-1	指標暫存器 P0~P9 應用說明	5-12
5-4-2	間接定位程式範例	5-13
5-4-3	負數之表示及取得 (初學者請略過本節)	5-16
5-4-4	浮點數之表示 (初學者請略過本節)	5-16
5-5	運算元遞增/減之溢位與欠位 (初學者請略過本節)	5-18
5-6	加/減運算之進位與借位	5-19

5-1 應用指令之通則

M 系列 PLC 之應用指令可分為輸入控制、指令號碼名稱、運算元及功能輸出四部分。而各指令之輸入控制、運算元、及功能輸出之數目不盡相同（請參閱各指令說明）。



應用指令之通則

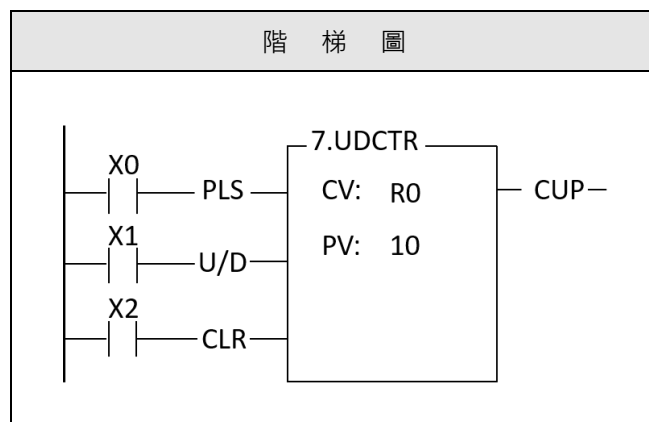
5-1-1 輸入控制態

M SERIES 除 7 個無輸入控制之應用指令外，其他應用指令至少有一個輸入控制，最多為八個。應用指令係依輸入控制信號之組合以決定該指令是否執行，以及執行何種運算。在 UperLogic 套裝軟體上及階梯圖程式印出時，所有之應用指令符號之輸入控制及功能輸出端子上均有加註英文註解簡寫，以註明該端子是何種功能控制或輸出，以利於記憶和閱讀，如上圖例 2 第一個輸入標註“PLS”，表示計數脈波 Pulse 由 0→1 (升緣) 時，該計數器才計數一次，第二個輸入標註“U/D”斜線上方 U 表上數 Up，下方 D 表示下數 Down，若此輸入為 1 則當計數脈波 PLS 來時，該計數器值會加 1，反之若為 0 則減 1，第三個輸入標示“CLR”，表示清除 Clear，即當此輸入為 1 時，該計數器之計數值會被清為 0。其他應用指令之輸入控制註解請參閱各指令說明。

註：無輸入控制指令係指該指令需直接接於母線，不能串接輸入控制元件，亦無功能輸出。該指令本身單獨形成一個網路。有 MCE、SKPE、LBL、RTS、FOR、NEXT 等 6 個無輸入控制指令，請參閱第 6 及 7 章各該指令之說明。

所有應用指令之各“輸入控制”均應有元件連接，否則會出現語法錯誤。如下圖例 3，FUN7 為三輸入之應用指令，在 FUN7 指令前面之三個元件 (X0, X1, X2) 分別對應到 FUN7 之第一個輸入 PLS，第二個輸入 U/D 和第三個輸入 CLR。

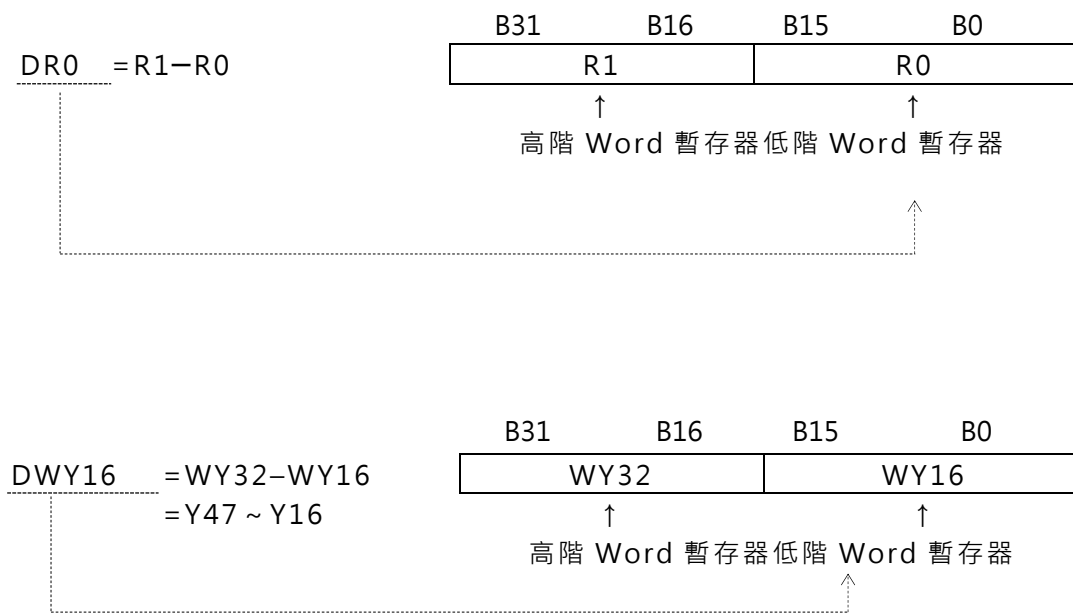
例 3：



應用指令之輸入控制範例

5-1-2 指令號碼與衍生指令

D：表示 Double Word，雙字元組（32 位元）之意。在 M SERIES 中之暫存器均以字元組 WORD（16 位元）為基本單位，即所有 R、T、C 暫存器（C1024~C1063 除外）均為 16 位元長度，例如 R0、R1、T0、．．．．．等。若需 32 位元長度之暫存器，則必須由兩個連續之 16 位元暫存器合併起來而形成如 R1-R0、R3-R2、．．．．．等，針對這種連續兩個 16 位元暫存器組成之雙字元組暫存器，我們以該雙字元組暫存器之低階暫存器號碼（如 R1-R0 取 R0，R3-R2 取 R2）加上 D 表示之（如 DR0 表示 R1-R0，DR2 表示 R3-R2），例如您在監視模式(MON)下鍵入如下之 DR0 或 DWY16，將會顯示 32 位元（R1-R0，或 WY32-WY16）長度之數值。



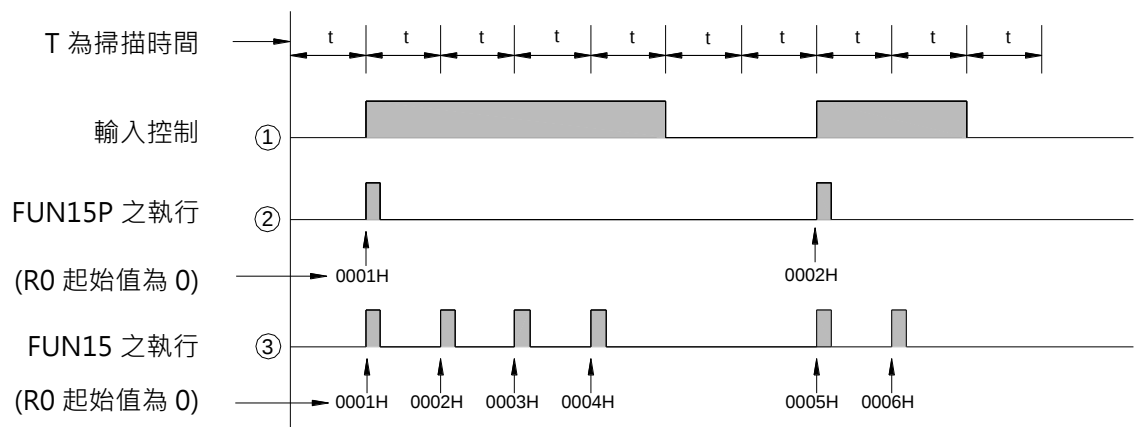
註 1：在階梯圖之表示，為方便區別 16 位元或 32 位元指令，我們在“指令號碼”後面加 D 尾碼以表示 32 位元指令，其來源或目的運算元的長度當然也是 32 位元。但在運算元（如 S、D……）欄上只標示 Double Word 之低階暫存器號碼，如 5.1.3 章節中例 4 中“被加數”Sa：R0 因該指令 FUN11D，有 D 尾碼，故所有來源或目的運算元均無冠上 D 字，亦即 Sa 為 DR0=R1-R0，Sb 為 DR2=R3-R2……但非來源或目的運算元如指標 Pr、數值 N、長度 L……等則無論 D 或非 D 指令均固定只有 16 位元長度，請特別注意。在 16 位元指令因其運算元長度只有一個 Word，也正是運算元欄上所標示之暫存器號碼，如例 1 中之 D=R0。

註 2：在階梯圖存取 BIT 狀態時，可以使用前綴字“W”來存取連續 16 位元之 BIT 狀態，例如，WX0 代表 X0~X15，WM32 代表 M32~M47，但存取之 BIT 位址必須為 16 位元之倍數位址，例如 WM16 是合法位址，但 WM8 是不合法位址。

P：表示 Pulse（脈沖）模式運作，也就是每當輸入控制由 0→1 瞬間（升緣↑）該指令即執行一次，如上圖例 1 若指令號碼加上 P 碼（即 FUN15P），則只有在輸入控制信號之升緣（0→1）時 FUN15P 才執行一次。若指令號碼後無 P 尾碼，則為連續執行模式，即只輸入控制為 1，PLC 每次掃描到該指令均會執行一次，一直至輸入控制變為 0 為止。在本手冊之應用指令說明中之輸入敘述有如下之敘述例：

- 當運算控制“EN”=1 或由 0→1（**P** 指令）時，……………

前者即表非 P 指令（連續模式）之執行條件，後者即為 P 指令（脈沖模式）之執行條件。下列波形圖為上節範例 1（FUN15）工作在 P 模式和非 P 模式下其執行結果(R0)之比較。



上在 P 模式和非 P 模式下其執行結果(R0)之比較圖

D P：表該指令為 32 位元指令，且為脈沖模式運作。

註：實際控制應用上大部分之應用指令都可使用 **P** 指令，在程式設計時請儘可能使用 **P** 指令，以節省程式執行時間。

5-1-3 運算元

運算元為指令運算時之參考或寫入之對象。可分為只供參考，內容不會因指令運算而改變之來源運算元 (Source，簡稱 S) 及用來儲存運算結果之目的運算元 (Destination 簡稱 D) 兩大類。以下就 M 系列 PLC 應用指令中，主要之運算元名稱及性質作說明，並將可當運算元之接點、線圈或暫存器之類別範圍分述如下：

● 主要運算元名稱及性質：

簡 寫	名 稱	說 明
S	來源運算元 (Source)	S 為指令運算中之資料讀取、參考的對象，其內容不會因運算而改變，若不只一個以上，則以註腳區分，如 Sa、Sb。
D	目的運算元 (Destination)	用以存放指令運算結果之區域，其原始資料在運算後會破壞，只有能寫入之線圈或暫存器才能當目的運算元。
L	長 度 (Length)	用以表示一連串資料或列表(Table)之長度 (範圍)，可為常數或變數。
N	數 值 (Number)	用以指定次數，個數 (如第 N 個位元) 等之固定數字，若不只一個，則以註腳區分如 Na、Nb、Ns、Nd 等。
Pr	指 標 (Point)	用以指定一串資料或列表中之某個資料或暫存器，通常 Pr 值為可變，故不能為常數或輸入暫存器。
CV	現在值	用於 T、C 中，只能為可寫入之暫存器。
PV	設定值	用於 T、C 中，只供參考比較用。
T	列 表 (Table)	列表是一連續暫存器組合之稱，其運作單位係以字元組或雙字元組為單位，若不只一個，則以註腳區分，如 Ta、Tb、Ts、Td 等。
M	矩 陣 (Matrix)	矩陣亦是一連續暫存器組合之稱，但是其運作單位是以位元為單位。若不只一個則以 Ms、Md、Ma、Mb 等表示。

主要運算元名稱及性質對照表

除上述主要運算元外，尚有用以指定特定用途之運算元，如 Fr 表頻率、ST 表堆疊、QU 表示 QUEUE 等，請參閱各指令之說明。

● 運算元類別與範圍：應用指令之運算元類別有 a.單點(數位) b.暫存器 c.常數 三種

a. 單點 (數位) 運算元：

在應用指令中，有單點運算元者（即其運算元只影響某一單點者）僅有 SET、RST、DIFU、DIFD、TOGG 四個指令，而且只能對 Y△△△△（外部輸出）、M△△△△△（內部及特殊）、S△△△△（步進）三類型之繼電器運作。下表為可當此五指令之單點運算元之種類及範圍，細部解釋請參閱此五指令之說明。

運算元 範圍	Y	M	SM	S
	Y0 Y1023	M0 M9119	M9120 M25599	S0 S3103
D	○	○	○*	○

單點運算元可使用範圍表

“○”符號表 D 可用該類別之線圈當運算元。在 SM 欄位中“○”上方標有“*”符號，表示在 SM 中禁止寫入之特殊繼電器不得當作 D 運算元，請參考 2.3 章“CPU 模組特殊繼電器明細”。

b. 暫存器運算元：

應用指令中之運算元主要為暫存器運算元。暫存器運算元又分為兩類，一為原本就以 Word 或 2 Words 為單位之暫存器(R、T、C)。另一則為由 16 或 32 個單點(X、Y、M、S)組成 Word 或 2 Words 之暫存器(WX、WY、WM、WS)。下表為在本手冊中用以表示各指令之運算元所能容許之暫存器類別及其範圍之範例：

運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位 元正、負數	V·Z P0~P9
S	○	○	○	○	○	○	○	○	○	○	○*	○	○	○
D		○	○	○	○	○	○		○	○*	○*	○		○
⋮														

各指令之運算元的暫存器與範圍範例表

“○”符號表示可以以該類別之暫存器當運算元。在 SR 和 D 交會之欄位中，“○”符號上方標有“*”符號，表示 D 運算元若為特殊暫存器 SR 時，應扣除不可寫入之暫存器，請參閱 2.4 章“CPU 特殊暫存器明細”。

R43224~R47319 不是規劃為唯讀暫存器時，可當一般暫存器使用（可讀寫）

註 1：凡有 W 開頭之暫存器（WX、WY、WM、WS）表示此暫存器係由 16 個單點組成 Word 之暫存器。例如 WX0 表示由 X0（位元 0）~X15（位元 15）組成之暫存器，WY144 表示由 Y144（位元 0）~Y159（位元 15）所組成之暫存器。但注意號碼必須為 16 之倍數（如 0、16、32……等）才允許，不支援 8 之倍數

註 2：表中最後一個暫存器（Word），不能當 32 位元運算元，因 32 位元運算元需有連續兩個 Word 的長度才行，WM, WX, WY 的使用必須以 16 的倍數，譬如：WM16, WM32 是支援的；WM8 並不支援。

註 3：TMR（T0~T1023）和 CTR（C0~C1279）為計時器和計數器專用之暫存器，雖亦可當一般暫存器使用，但會造成系統複雜，除錯困難，因此除 T 或 C 指令外，其他指令應避免寫入 TMR 或 CTR。

註 4：T0~T1023 和 C0~C1023 均為 16 位元長度，而 C1024~C1279 限定為 32 位元長度，故不能當 16 位元運算元。

註 5：暫存器運算元除如上述直接以暫存器號碼（位址）來指定外，對於 RXXXXX 及 DXXXXX 暫存器尚可結合指標暫存器 V、Z 或 P0~P9 來作間接定址指定。利用指標暫存器（XR）作間接定址之說明請參考下節（5.3 節）之範例。

c. 常數運算元：

在 16 位元中之常數範圍最大為-32768 ~ 32767，32 位元之範圍為-2147483648 ~ 2147483647，而某些指令只能為正常數，因此我們以下列敘述表示 16 或 32 位元之常數範圍。

常數類別	範圍
16 位元正負數	-32768 ~ 32767
16 位元正數	0 ~ 32767
32 位元正負數	-2147483648 ~ 2147483647
32 位元正數	0 ~ 2147483647
16 / 32 位元正負數	-32768 ~ 32767 或 -2147483648 ~ 2147483647
16 / 32 位元正數	0 ~ 32767 或 0 ~ 2147483647

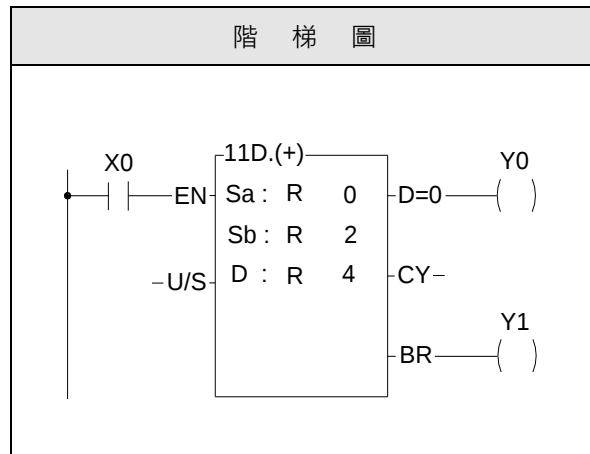
常數類別及其範圍對照表

此外有某些特定之運算元長度大小不一（如長度 L、位元數 N 等）將於各該運算元之欄位上直接標示範圍，請參考個別之指令說明。

5-1-4 功能輸出 (FO)

簡稱 FO(Function Output)為應用指令運算結果或狀態之輸出，如同「輸入控制」一樣。在 UperLogic 及程式印出之階梯圖應用指令上，其功能輸出上亦有英文註解說明該 FO 為何種功能，如上圖例 1 之 CY，例 2、例 3 之 CUP 及下圖例 4 之 D=0、CY、BR 均是。功能輸出 FO 最多只有 4 個 (即 FO0~FO3)，其編號順序是由上而下，第一個 FO 為 FO0，第二個為 FO1，最後一個為 FO3。FO 狀態之取出必須用 FO 指令，不使用之功能輸出可空著不接任何元件，如下圖例 4 之 FO1(CY)即是。

例 4：



以 FUN11 為範例之功能輸出圖

5-2 利用 W 前綴字作字元組及位元存取變換

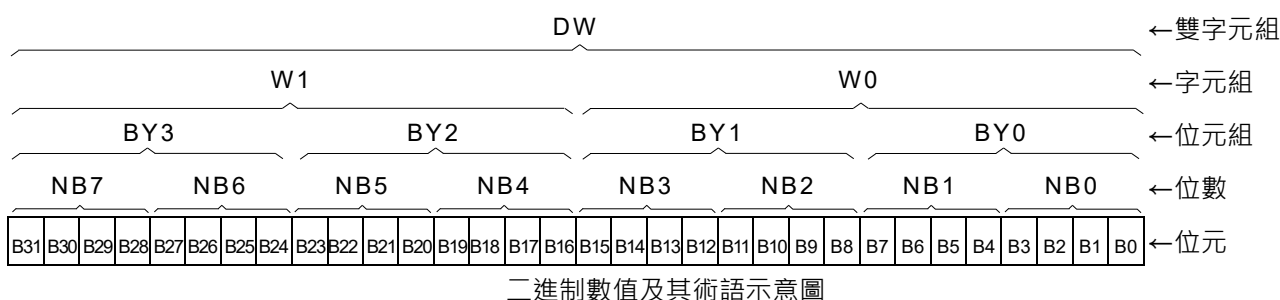
M SERIES 之單點 (BIT 狀態) 記憶體，皆可使用 W 前綴字來做字元組 (word) 存取，也就是一次存取 16 個單點，例如 WX0 意指一次存取 X0~X15 之狀態；反之，也可以利用此技巧來存取字元組資料之任一單點狀態，例如可將字元組資料放置在 WM0，則若要讀取此字元組之第 6 個位元狀態，只要直接讀取 M6 即可。

5-3 數目系統

5-3-1 二進制數值及其術語

二進制(Binary)為數位計算機之基本數目系統，PLC 是由數位計算機所構成，自然亦採用二進制，為便於表示及掌握二進數值，首先需了解如下之術語：

- 位元：(Bit 簡寫 B，如 B0，B1，．．．．．) 位元為二進制數值之最基本單位，其狀態非 1 即 0。
- 位數：(Nibble 簡寫 NB，如 NB0，NB1，．．．．．) 由連續的 4 個位元所組成 (如 B3 ~ B0) 可用以表示一個位數之 10 進制數字 0 ~ 9 或 16 進制之 0 ~ F。
- 位元組：(Byte 簡寫 BY，如 BY0，BY1，．．．．．) 是由連續之兩個位數所組成 (亦即 8 個位元，例如 B7 ~ B0)。可表示 16 進制之兩個位數值 00 ~ FF。
- 字元組：(Word 簡寫 W，如 W0，W1，．．．．．) 是由連續之兩個位元所組成 (亦即 16 個位元例如 B15 ~ B0) 可表示 16 進制之 4 個位數值 0000 ~ FFFF。
- 雙字元組：(Double word 簡寫 DW，如 DW0，DW1，．．．．．) 是由連續之兩個字位元組所組成 (亦即 32 個位元，例如 B31 ~ B0) 可表示 16 進制之 8 個位數值 00000000 ~ FFFFFFFF。



二進制數值及其術語示意圖

- 浮點數：(Floating Point Number) 也是由連續之兩個字位元組所組成。浮點數所能表示的最大範圍為 $\pm(1.8 \times 10^{-38} \sim 3.4 \times 10^{38})$ ，有關詳細之格式說明請參考 5.3.6 節。

5-3-2 M SERIES 之數碼

M SERIES 內部之數值運算或儲存全部採用二進值 (Binary)，因此外界輸入 PLC 內部之數值必須轉換成二進碼 PLC 才能處理，同樣地自 PLC 內部取出之數值結果亦均為二進值，因此 UperLogic 其所有數目最終均須化成二進制才能輸入 PLC。但因二進值極難輸入和閱讀，因此 UperLogic 在人機界面 (數值輸入或顯示) 部分均提供使用者以人們熟悉之 10 進值 (Decimal) 或 16 進制 (Hexadecimal) 來輸入或顯示，但實際上之數值處理全部都以二進碼來進行。

註：若您的數值輸入或顯示不是透過 UperLogic (例如以指撥開關或 7 段顯示器透過 I/O 點而輸入 PLC 或自 PLC 取出)，那麼您就得自己藉著階梯圖程式指令來處理二進制和 10 進制間之轉換，使您雖不透過 UperLogic 亦能以 10 進制來輸入及得到 10 進制之輸出顯示，請參考 FUN20(BIN→BCD)和 FUN21(BCD→BIN)之說明。

5-3-3 數值之範圍

如前述 M SERIES 內部全部採用二進制 (BCD 值只是為適合人們習慣，而由二進值轉成適合人們閱讀之顯示用數碼而已)。在 PLC 之數值有 16 位元、32 位元以及浮點數三種數值，分別能表示如下之範圍。

16 位元	-32768 ~ 32767
32 位元	-2147483648 ~ 2147483647
浮點數	$\pm(1.8 \times 10^{-38} \sim 3.4 \times 10^{38})$

5-3-4 數值之表示 (初學者請略過本節)

以下各節將敘述 16 位元及 32 位元數值之表示方式與格式。以供使用者能深入了解數值之運算過程及結果而能應付各種複雜之應用需求。

無論是 16 位元或 32 位元之數值，均以其最高位元 MSB (16 位元之 B15，32 位元之 B31) 表示該數值之正負 (0：正數，1：負數)，剩下之位元 (B14 ~ B0 或 B30 ~ B0) 才真正用以表示數值大小，茲以 16 位元為例說明如下： (32 位元之作法亦同，只是長度倍增而已)。

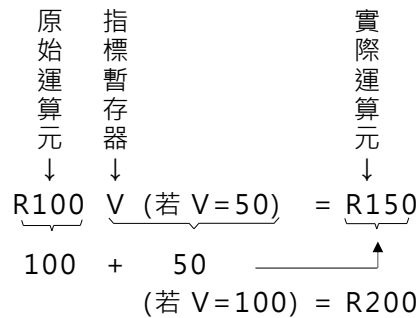


如上例，無論 16 位元或 32 位元，其二進制之位元由最低位元 LSB (B0) 開始，B0 代表 1，B1 代表 2，B2 代表 4，B3 代表 8，……餘此類推，而其數值則為所有為 1 之位元所代表數值之總和。

5-4 利用指標暫存器 (XR) 作間接定址

在 M SERIES 應用指令中，有些運算元可以結合指標暫存器 (V 或 Z) 而作間接定址之指定 (在每個指令說明欄之運算元敘述中會註明) 但能夠結合 V 或 Z 作間接位址指定之運算元只限定為 RXXXXX 範圍內之暫存器，若 DXXXXX 範圍內暫存器欲使用間接定址則需使用指標暫存器 P0~P9 (其他運算元如單點、常數等均不能作間接位址指定)。

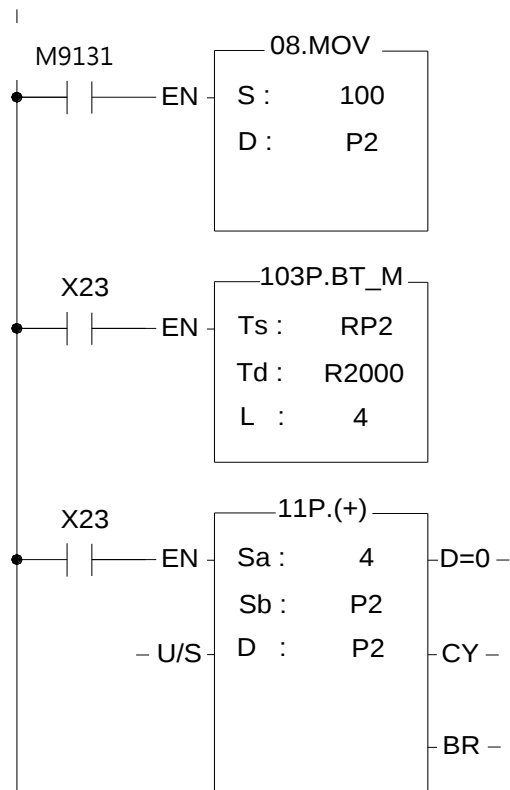
指標暫存器 XR 共有 12 個 (V、Z、P0~P9)，實際上在 M SERIES 之 V 為 R43214，而 Z 暫存器則為 R43216。運算元結合指標暫存器作間接定址之表示方式是原運算元後緊接 V 或 Z 來表示：



如上圖示，只要變更 V 之值即可變更運算元之位址，利用此功能結合 M SERIES 之應用指令，您可以極簡易之指令，達成功能強大、極具效率之控制應用，如下圖程式例，您只須以一個區塊搬移指令 (BT_M) 即達成諸如停車管理系統之動態區塊資料顯示。

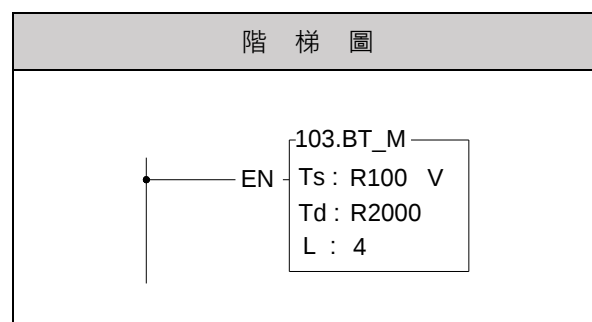
5-4-1 指標暫存器 P0~P9 應用說明

- 在間接定址應用中，RXXXXX 暫存器可以結合指標暫存器 V、Z 和 P0~P9 作間接定址應用；DXXXXX 暫存器不可以結合指標暫存器 V、Z 作間接定址應用，但可以結合 P0~P9 作間接定址應用。
- 當 RXXXXX 暫存器要結合 V、Z 作間接定址應用時，例如 R0 要結合 V、Z 做間接定址應用，則所輸入之格式為 R0V(當 V=100 時，則指向 R100)或 R0Z(當 Z=500 時，則指向 R500)；而欲結合 P0~P9 作間接定址應用時，則所輸入之格式為 RPn (n 為 0~9)或為 RPmPn (m,n 為 0~9)，例如 RP5 (若 P5 內容為 100，則指向 R100) 或 RP0P1(若 P0 內容為 100，P1 內容為 50，則指向 R150)。
- 當 DXXXXX 暫存器要結合 P0~P9 作間接定址應用時，則所輸入之格式為 DPn (n 為 0~9)或為 DPmPn (m,n 為 0~9)，例如 DP3 (若 P3 內容為 10，則指向 D10) 或 DP4P5(若 P4 內容為 100，P5 內容為 1，則指向 D101)。
- P0~P9 指標暫存器可同時結合運用，例如 P2=20、P3=30，當 RXXXXX 或 DXXXXX 暫存器一次結合兩個指標暫存器時，RP2P3 就會指向 R50，DP2P3 就會指向 D50；也就是說兩個指標值之間的關係是相加的。

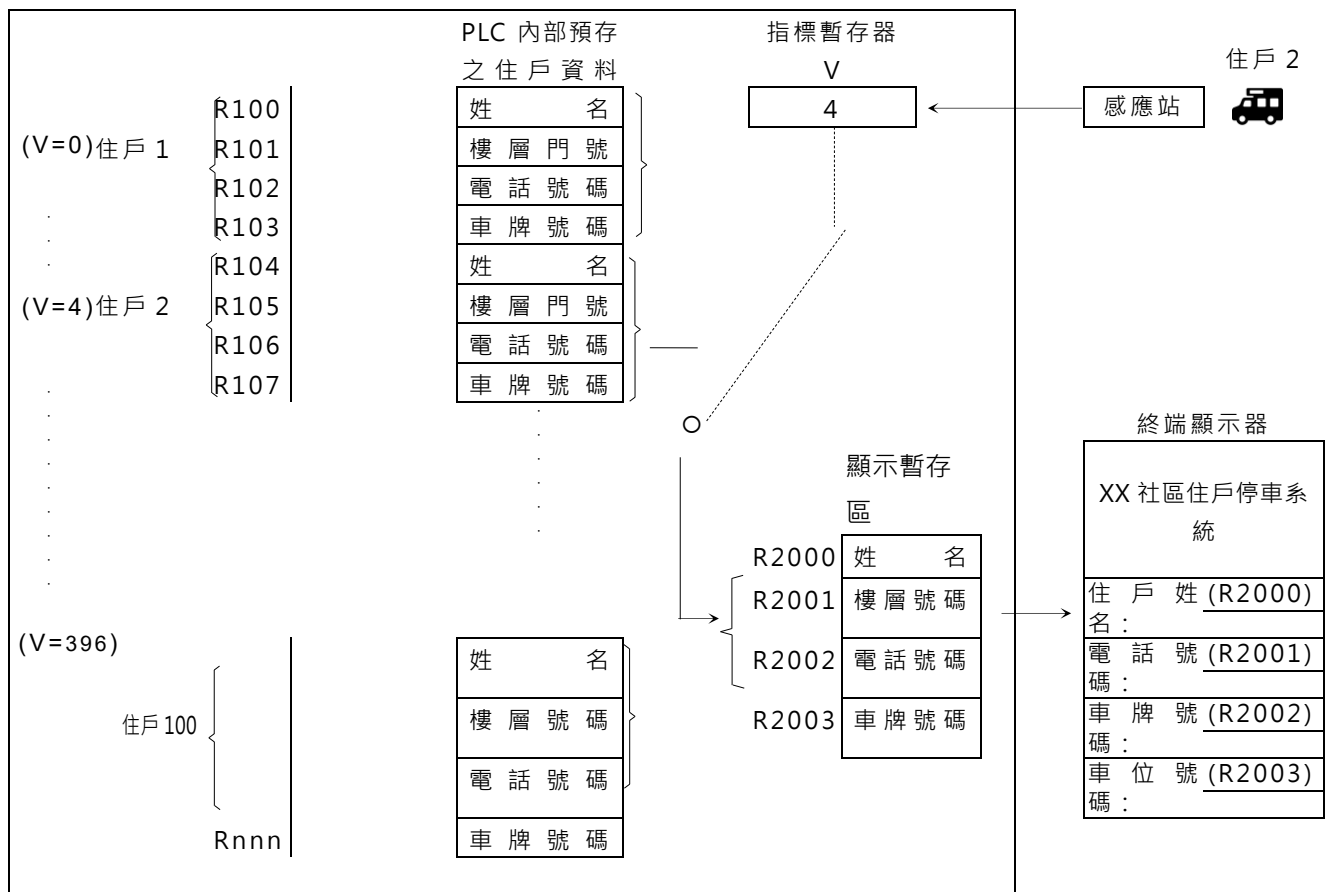


1. 開機時 M9131 起始脈波將 100 搬入指標暫存器 P2。
2. 當 X23 由 0→1 時，Fun103 將由 R100(因為 P2=100)開始，一次 4 個暫存器的長度，依序搬到 R2000。也就是說第一次將 R100~R103 搬到 R2000~R2003，第二次將 R104~R107 搬到 R2000~R2003，第三次將 R108~R111 搬到 R2000~R2003 依此類推。
3. Fun11 用來將指標每次增加 4 個 word 用，亦即 X23 每“ON”一次，P2 之指標值便加上 4。

5-4-2 間接定位程式範例



FUN103 使用間接定址 V 之範例



自動化停車場管理系統示意圖

程式說明

上例假設某社區住戶之自動化停車場管理系統，共有 100 個住戶停車位，每個住戶均有 1 組基本資料，分別為住戶姓名，電話號碼，車牌號碼，停車位號碼等 4 個，每組資料如上圖示佔用連續 4 個 PLC 內部暫存器，共佔用 R100 ~ R499 等 400 個暫存器，每個住戶均有一不同卡號之卡片，以供進出口門禁及停車場進出之感應通行使用，其卡號為 0，4，……，396 等 100 種，在 PLC 感應到卡號後，將之存入指標暫存器“V”，而在管理員處之終端機（LCD 或 CRT）顯示資料則固定由 PLC 內部之 R2001 ~ R2003 來抓取並顯示，如本例感應到住戶 2 之卡片，其值=4，因此 V 暫存器=4，PLC 立即將 R104 ~ R107 之資料搬移至顯示暫存區（R2000 ~ R2003），因此管理員處之終端機可在感應到住戶 2 之卡片之同時，立即將其資料顯示在終端機上。

警告

1. 運用指標暫存器作間接定址之應用雖然功能強大、彈性方便，但相對地 V、Z 內容值之任意變化可能對正常資料區之錯誤寫入有著極大之殺傷力，因此使用者在使用時應特別小心。
2. 在間接定址所能定址之 RXXXXX 及 DXXXXX 資料暫存器範圍中，暫存器 R34768 ~ R47319 等之 12552 個暫存器（即 IR、OR、SR）為系統或 I/O 用之重要暫存器，任意對此等暫存器之寫入，將可能使系統或 I/O 錯誤，造成重大之災害。鑑於 V、Z 值對暫存器位址變化之靈活變化，使用者可能不易察覺或掌握，因此 M SERIES 對間接定址之寫入動作會自動檢查寫入目標位址（Destination）是否在上述之 R34768 ~ R47319 範圍內，若是則不執行該寫入動作。若應用上確實需要對 R34768 ~ R47319 之暫存器作寫入，請使用非間接定址之指令來執行，但讀取動作不在此限制內。

5-4-3 負數之表示及取得(初學者請略過本節)

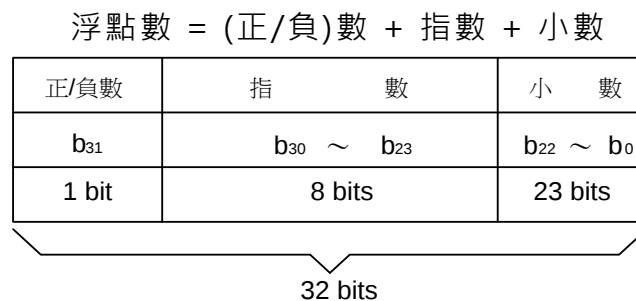
如前述當 MSB 為 1，則此數為負數，M SERIES 之負數係以“2 的補數”(2'S Complement)來表示。所謂 2 的補數，係將等值正數之所有位元 (B15 ~ B0 或 B31 ~ B0) 倒相 (為 1 之位元變 0，為 0 之位元變 1，亦即所謂 1 的補數)，然後再加上 1 即變成 2 的補數，茲以上例正數 12345，取其 2 的補數 (即-12345) 為例說明如下：

$$\begin{array}{rcl}
 12345 & \Rightarrow & \boxed{0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1} \Rightarrow 3039H \\
 \\
 12345\text{ 之 }1\text{ 的補數} & \Rightarrow & \boxed{1\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0} \Rightarrow CFC6H \\
 \\
 \begin{array}{c} + \\ \hline \end{array} & & \begin{array}{c} 1 \\ \hline \end{array} \\
 12345\text{ 之 }2\text{ 的補數} & \Rightarrow & \boxed{1\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1} \Rightarrow CFC7H \\
 (-12345) & &
 \end{array}$$

負數的表示及取得示意圖

5-4-4 浮點數之表示(初學者請略過本節)

永宏 M SERIES 浮點數格式同 IEEE-754 所制定之標準。儲存之格式共佔用 32 個位元(雙字元組)其說明如下所示：



浮點數之表示示意圖

若(正/負)位元之值為 0，表示此浮點數為正值，反之如果若(正/負)位元之值為 1，代表此浮點數為負值。指數表示法為超 127 法，舉例來說，若指數之值為 128 就代表 1 次方，指數之值為 129 就代表 2 次方……依此類推。若想表示指數為負之值，則 126 就是-1 次方，125 為-2 次方……依此類推。

小數位數有 23bits，用來存放小數點以下之數字，正規化格式要求小數點之前一個位數，必須為 1，且不必儲存(稱為隱藏位元)。

整數轉換浮點數之規則為如下：

$$N = (-1)^S * 2^{(E-127)} * (1.M) \quad 0 < E < 255$$

轉換範例 1

$$1 = (-1)^0 * 2^{(01111111)} * (1.000 \dots 0)$$

此範例中(正/負)數位元為 0，指數部分為 127(超 127 法)=01111111，隱藏位元為 1，而小數位數則全部為 0，因此經過轉換後的浮點數表示法如下所示：

$$\text{Code}(1) = \begin{array}{|c|} \hline 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \dots\dots\dots 0 & 0 & 0 & 0 & 0 \\ \hline s & e & e & e & e & e & e & e & e & e & m & m & m & m & m \dots\dots\dots m & m & m & m & m \\ \hline \end{array}$$

= 3F800000H

轉換範例 2

$$0.5 = (-1)^0 * 2^{(01111110)} * (1.000 \dots 0)$$

此範例中(正/負)數位元為 0，指數部分以超 127 法表示為 126=01111110(因指數部分為-1，126-127=-1，故指數部份在超 127 法中要轉換成 126)，隱藏位元為 1，而小數位數則全部為 0，因此經過轉換後的浮點數表示法如下所示：

$$\text{Code}(0.5) = \begin{array}{|c|} \hline 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \dots\dots\dots 0 & 0 & 0 & 0 & 0 \\ \hline s & e & e & e & e & e & e & e & e & e & m & m & m & m & m \dots\dots\dots m & m & m & m & m \\ \hline \end{array}$$

= 3F000000H

轉換範例 3

$$-500.125 = (-1)^1 * 2^{(10000111)} * (1.111010000100000000000000)$$

此範例中(正/負)數位元為 1，指數部分以超 127 法表示為 135=10000111(因指數部分為 8，135-127=8，故指數部份在超 127 法中要轉換成 135)，隱藏位元部分為 1，而小數位數則為 111101000010000000000000，因此經過轉換後的浮點數表示法如下所示：

$$\text{Code}(-500.125) = \begin{array}{|c|} \hline 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 00 \dots\dots 0 & 0 \\ \hline s & e & e & e & e & e & e & e & e & e & m & m & m & m & m & m & m & m & m & m & m & m & m & m \dots\dots\dots m & m \\ \hline \end{array}$$

= C3FA1000H

5-5 運算元遞增/減之溢位與欠位(初學者請略過本節)

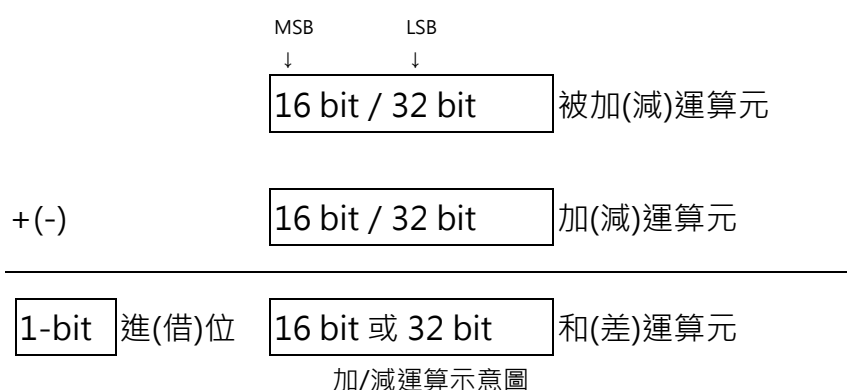
16 位元及 32 位元運算元所能表示之數位極限正數最上限為 32767 和 2147483647，而負數之最下限為-32768 和-2147483648。當對一運算元作遞增或遞減動作（例如計數器上/下數或使暫存器值+1 或-1 時，若結果使其數值超過該運算元正數之最上限，則稱之為溢位（Overflow：OVF），溢位之結果會使數值循環至負數之最下限（例如 16 位元上限 32767 再加 1，變成-32768）。若遞增(減)結果使其數值小於負數最下限，則稱之欠位（Underflow：UDF），而欠位之結果會使數值循環至正數之最上限（例如-32768 若再減 1 變成 32767），如下表所示。在 M SERIES 之遞增指令之功能輸出（FO）均有溢位或欠位之旗號輸出，可供您作串聯（Cascade）應用而獲得超過 16 或 32 位元之運算結果。

遞增(減) 結果 溢/欠位	16 bit 運算元	32 bit 運算元
遞增		
遞減		

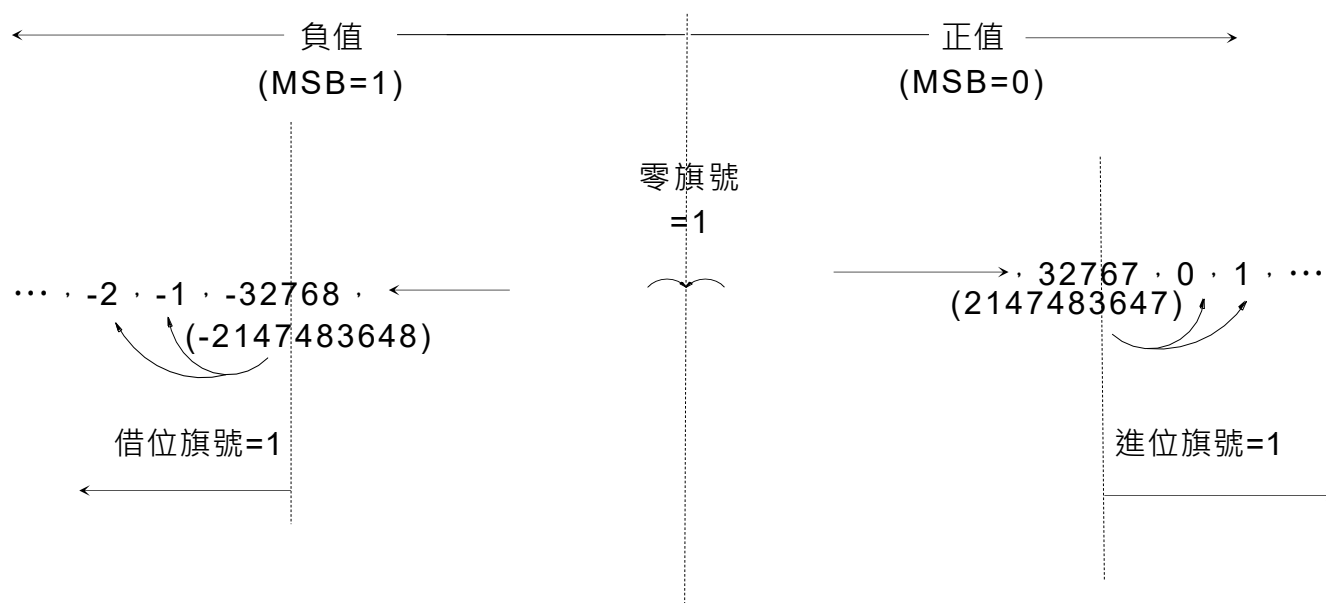
32 位元與 16 位元溢位、欠位對照表

5-6 加/減運算之進位與借位

溢/欠位之發生係針對單一運算元之遞增/減致使該運算元之值超出其所能表示之正/負值極限時，產生溢/欠位旗號。而進/借位則不同於溢/欠位，首先其必有兩個運算元作加(減)運算，而得到和(差)結果與進/借位旗號。因被加(減)、加(減)及和(差)之位數(bit 數)均一樣(16 bit 或 32 bit)，因此相加(減)之結果將可能造成和(差)之值超出 16 或 32 bit，因此需以進(借)位旗號配合和(差)運算元來表示真正之數值。而進位旗號發生在加(減)結果超出該和(差)運算元所能表示之正值最大極限(32767 或 2147483647)時，而借位則發生在加(減)結果超出該和(差)運算元所能表示之負值最大極限(-32768 或 -2147483648)時，因此加(減)運算後之真正結果為進(借)位再加上和(差)運算元之值，M SERIES 之加減指令之功能輸出(FO)均有進位與借位旗號輸出，可供您獲得真正之結果。



因 M SERIES 之數值運算均採用 2 的補數，因此加（減）運算所得之和（差）值之負值之表示將不同於我們一般習慣之負值表示方式。運算結果為負值時，其和（差）運算元將永遠不可能出現 0 值。進位旗號代表正值 32768(2147483648)，借位旗號則代表負值-32768(-2147483648)。



[illegible]

加/減運算及進位與借位示意圖

※若使用上不會需要進借位處理，建議可以使用 Fun224 快速加法和 Fun225 快速減法，因與 Fun11 加法和 Fun12 減法相比，不需做進借位。

6

基本應用指令

6-1	一般計時器(T)	6-1
6-2	一般計數器(C)	6-6
6-3	設定(SET)	6-10
6-4	清除(RST)	6-13
6-5	主控迴路開始指令(MC)	6-15
6-6	主控終止指令(MCE)	6-18
6-7	跳過迴路開始指令(SKIP)	6-19
6-8	跳過迴路終止指令(SKPE)	6-21
6-9	上微分指令(DIFU)	6-22
6-10	下微分指令(DIFD)	6-24
6-11	位元位移(BSHF)	6-25
6-12	上/下數計數器(UDCTR)	6-27
6-13	搬移(MOV)	6-30
6-14	倒相後搬移(MOV/)	6-32
6-15	交替開關(TOGG)	6-34

6-16	快速加法運算 F (+)	6-35
6-17	快速減法運算 F (-)	6-38
6-18	加法運算 (+)	6-41
6-19	減法運算 (-)	6-43
6-20	乘法運算 (*)	6-45
6-21	除法運算 (/)	6-48
6-22	MOD 取餘數	6-52
6-23	遞增 (+ 1)	6-54
6-24	遞減 (- 1)	6-56
6-25	數值比較 (CMP)	6-58
6-26	邏輯及運算(AND)	6-60
6-27	邏輯或運算(OR)	6-62
6-28	BIN→BCD 變換	6-64
6-29	BCD→BIN 變換	6-66

6-1 一般計時器(T)

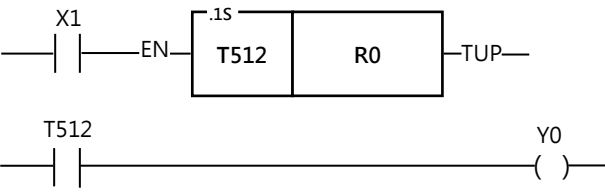
T	一般計時器 (TIMER)													T
指令說明														
階段圖符號 計時控制 — EN — TB Tn PV — TUP — 計時到(FO0) TB：時基 (Time Base) · 有 0.01S 、 0.1S · 1S 三種。							運算元 Tn：計時器號碼，為儲存累計之計時時間 (即現在值 CV) 。 PV： (Preset Value) 為計時器之設定值。							
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	WX0 WX100 8	WY0 WY100 8	WM0 WM9104	WSM9120 WM2558 4	WS0 WS308 8	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	0 32767
	Tn					○								
	PV	○	○	○	○	○	○	○	○	○	○	○	○	○
功能敘述														

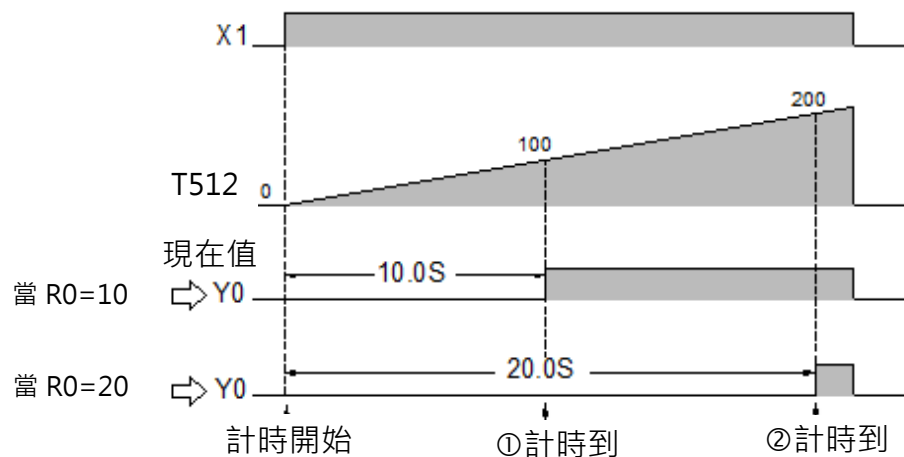
- 計時器總數為 1024 個 (T0 ~ T1023)，共有 0.001 秒、0.01 秒、0.1、1 秒四種時基 (Time Base)。PLC 在出廠時已將這四種時基之計時器之個數、編號分配如下：【若這不符您的需要可利用“專案” -> “記憶體配置” (CONFIGURATION) 功能自行調整】。
T0 ~ T255：0.001 秒計時器 (設定值 0.00 ~ 32.767 秒)。
T256 ~ T511：0.01 秒計時器 (設定值 0.0 ~ 327.67 秒)。
T512 ~ T767：0.1 秒計時器 (設定值 0 ~ 3276.7 秒)。
T768 ~ T1023：1 秒計時器 (設定值 0 ~ 32767 秒)。
- M SERIES 副程式若觸發 Timer, Timer 就開始運作, 要停止 Timer 運作, 就必須 Disable Timer, 不呼叫副程式, 不等於 Disable Timer, 若沒 Disable Timer，就會繼續計時。
- PV 若為暫存器，則計時時間 = 時基 X 暫存器內容值，只要變化暫存器內容值即可改變該計時器之計時時間，請參考例 2。
- 計時器之最大誤差為一個時基加一個掃描時間，為了減少應用上之計時誤差，請儘量使用時基愈小之計時器。
- 當計時控制“EN”為 1 時，計時器開始計時 (現在值由 0 開始累加) 直至“計時到” (Time Up，即現在值 $\geq$ 設定值) 後，該計時器之 Tn 接點和計時到旗號 TUP (F00) 均會變為 1，表示計時到。只要計時控制“EN”輸入一直保持 1，即使計時器 Tn 之現在值 CV 已到達或超過設定值，計時器之現在值 CV 仍會持續累加計時 (M9158=0 時)，一直累積到最上限 (32767) 為止，而 Tn 接點狀態和旗號則只要 CV $\geq$ PV 就會為 1，除非 EN 輸入為 0。當計時控制“EN”為 0 時，Tn 現在值 CV 會立刻被清為 0，同時 Tn 接點和計時到旗號 TUP 均將變回 0 (請參閱下圖①)。
- M 主機可控制 M9158 為 1，使計時器“計時到”時，現在值 CV 不再累加而保持在設定值。M9158 之出廠設定為 0，您可在程式中任一計時器指令執行之前，設入 M9158 之狀態，而能個別設定該計時器在“計時到”後 CV 為繼續計時累加，或保持在設定值 (請參閱下圖②)。

T	一般計時器 (TIMER)	T			
程式範例 1	固定時間計時器				
階段圖 “計時到”信號直接由 F00 取出之範例。 ST <pre>Timer(EN:= X1, T:= 0, PV:= 1000, IsTimeout=> Y0); M9158 := TRUE; Timer(EN:= X1, T:= 1, PV:= 1000, IsTimeout=> T1);</pre> ① M9158=0 (出廠設定) Y0 或 ② M9158=1 <tr><td>程式範例 2</td><td colspan="2">可變時間計時器</td></tr>			程式範例 2	可變時間計時器	
程式範例 2	可變時間計時器				

T	一般計時器 (TIMER)	T
---	--------------------	---

上例之計時器設定值 PV=10 為常數，故一旦程式輸入完成後即固定無法改變。但在許多應用場合，計時器之計時時間需要隨時地動態改變。欲達成此要求，可將 PV 改成暫存器 (R 或 D 等)，再改變暫存器之內容值，則可動態地改變計時時間。如本例 R0 值若設為 10，則 T 為 10 秒計時器，若將 R0 值變成 20，則 T 變為 20 秒之計時器。如此您可很容易地在 PLC 運轉 (RUN) 中動態地改變計時時間。

階梯圖	ST
	<pre> Timer(EN:= X1, T:= 512, PV:= R0, IsTimeout=> T512); IF T512 THEN Y0 := TRUE; ELSE Y0 := FALSE; END_IF </pre>
“計時到”信號間接由 T512 接點取出之範例。	



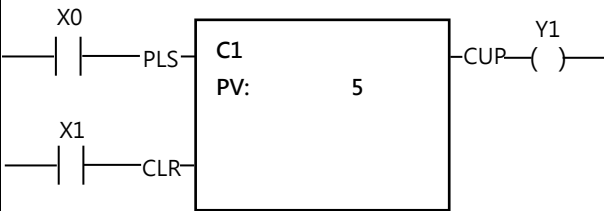
T	一般計時器 (TIMER)	T
註：計時器之設定值 PV 若為 0，則此計時器在 PLC 一開始 RUN 時計時到輸出接點立即為 1 (但 EN 輸入必須為 1)，且一直為 1，而不管 CV 值如何變化，直到 EN 輸入變為 0 止。		

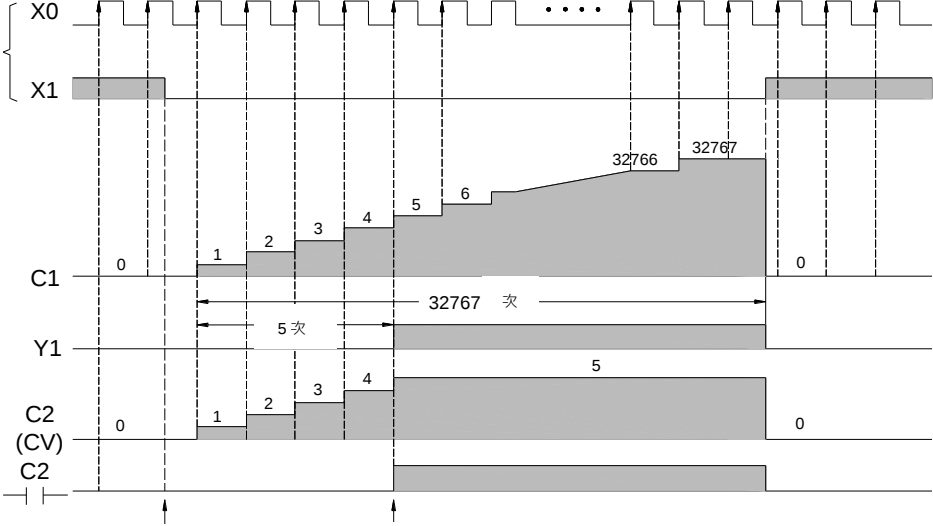
6-2 一般計數器(C)

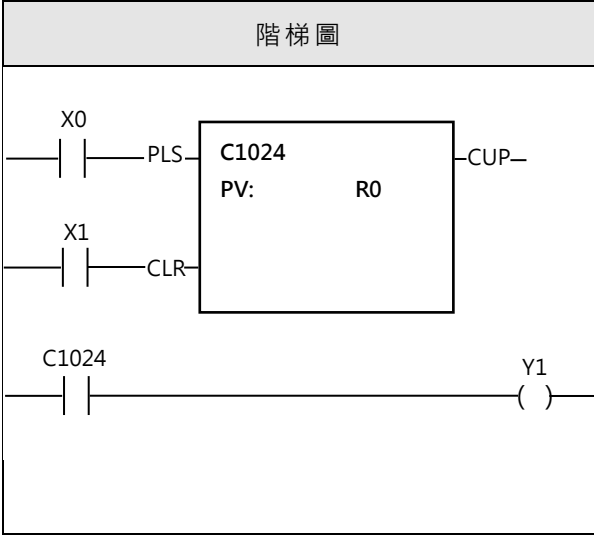
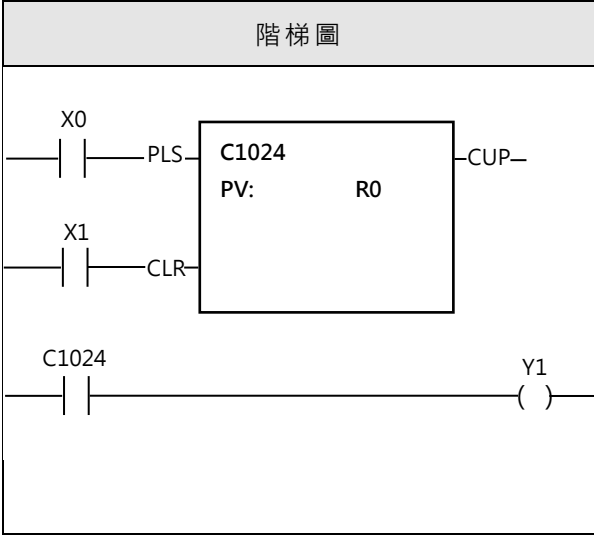
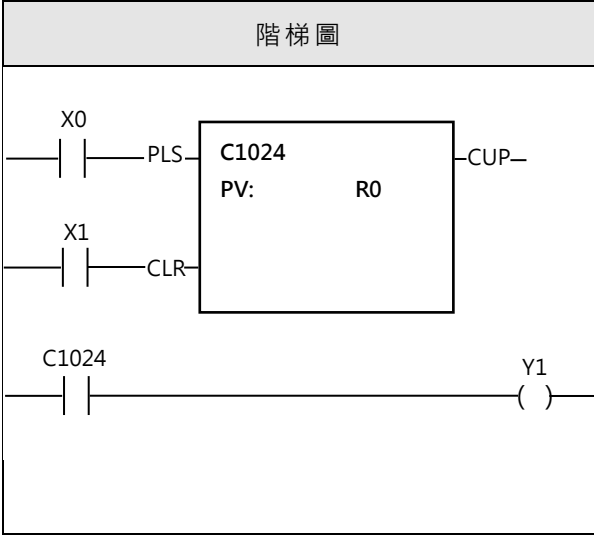
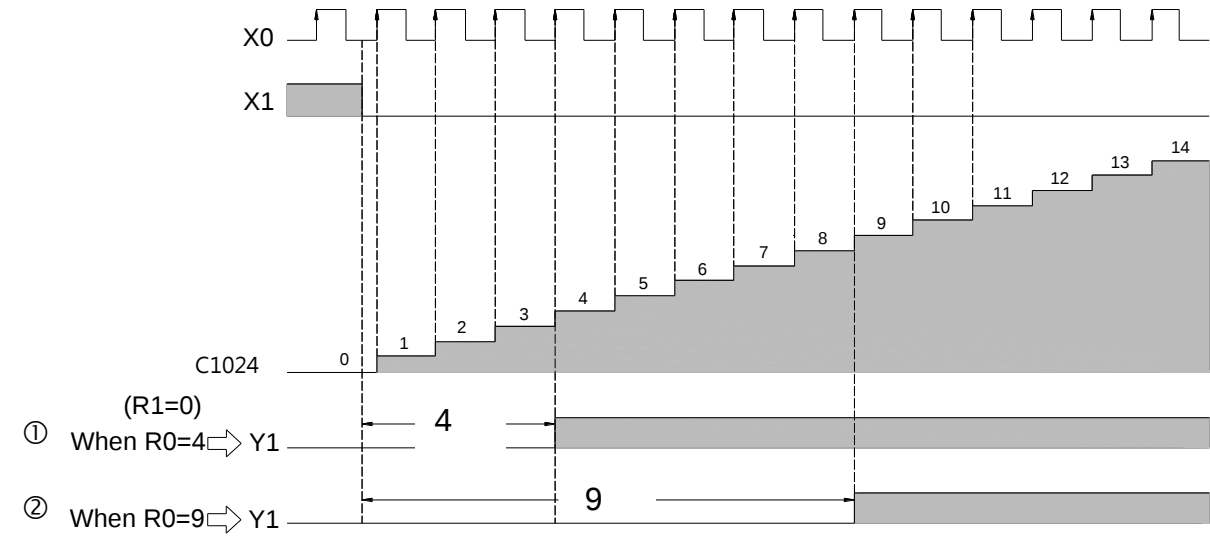
C	一般計數器 (COUNTER) (16 位元 : C0 ~ C1023 , 32 位元 : C1024~1279)														C
指令說明															
階段圖符號 計數脈波 — PLS — Cn PV : 清除控制 — CLR — CUP — 計數到 (FO0)										運算元 Cn : 計數器號碼 , 為儲存累計之計數值 (即現在值 CV) 。 PV : (Preset Value) 為計數器之設定值 。					
範圍 運算元		WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	0 32767
Cn								○							
PV		○	○	○	○	○	○	○	○	○	○	○	○	○	○

- C0~C1023 為 16 位元計數器 (共 1024 個) , 其設定值可為 0~32767 次。其中 C0~C139 為保持型 (斷電後再開或 PLC STOP 後再 RUN 計數值仍保留) , C140~C1023 為非保持型 (斷電再開或 PLC STOP 後再 RUN 計數值變為 0) 。
- C1024 ~ C1279 為 32 位元計數器 (共 56 個) , 其設定值可為 0 ~ 2147483647 次。其中 C1024~C1063 為保持型 , C1064~C1279 為非保持型 。
- 上列 16 位元及 32 位元計數器之保持 / 非保持之個數分配為出廠之原始設定 , 若這不符您的需求 , 可利用 “ 架構配置 ” 功能自行調整 。
- 為確保 C0~C1024 能夠正確計數 , 計數脈波之狀態為 1 時或為 0 時 , 皆必須大於一個掃描時間 , 否則會造成計數不正確 。
- 本指令在每一個掃描週期會依據 PLS 訊號的變化做計數動作 , 因此只要訊號變化後持續時間超過掃描週期 , 此指令就可以偵測到變化。但掃描週期時間會依據使用者程式的多寡而變化 , 若要得到精確的計數值 , 請使用軟體或硬體之高速計數器 。

C	一般計數器 (COUNTER) (16 位元 : C0 ~ C1023 , 32 位元 : C1024 ~ 1279)	C
功能敘述	● 當清除控制 “CLR” 為 1 時 , Cn 計數器之現在值 CV 和 Cn 接點以及計數到旗號 CUP (FO0) 均將被清為 0 , 計數器無法計數。	
程式範例 1	固定時間計數器	

階梯圖	ST
	<pre>Counter(Pulse:= X0, Clr:= X1, C:= 1, PV:= 5, IsUp=> Y1);</pre>
“計數到” 信號直接由 FO0 取出之範例。	

C	一般計數器 (COUNTER) (16 位元 : C0~C1023 · 32 位元 : C1024~1279)	C
	 The diagram illustrates the operation of a general counter. It shows the timing of input X0 (a series of pulses), input X1 (a pulse that starts the counter), counter C1 (which counts from 0 to 32767 and then resets to 0), output Y1 (which is high for 5 cycles), and counter C2 (CV) (which counts from 0 to 4 and then resets to 0). The counter C2 is the output of C1, which is high for 5 cycles.	
程式範例 2	32 位元可變計數器	
	如同計時器一樣，計數器之設定值 PV 若改成暫存器 (R、D.....等)，則計數器會以暫存器之內容為計數設定值。因此只要改變暫存器之內容值，即可動態地 (在 PLC RUN 中) 改變計數器之計數設定值。以下為 32 位元計數器以資料暫存器 R0 (實際上為 R1 與 R0 組成 32 位元設定值) 為設定值之範例。	

C	一般計數器 (COUNTER) (16 位元 : C0 ~ C1023 · 32 位元 : C1024~1279)	C						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>Counter(Pulse:= X0, Clr:= X1, C:= 1024, PV:= R0, IsUp=> C1024); IF C1024 THEN Y1 := TRUE; ELSE Y1 := FALSE; END_IF</pre></td></tr><tr><td colspan="2">“計數到” 信號間接由 C1024 接點取出之範例</td></tr></table>			階段圖	ST		<pre>Counter(Pulse:= X0, Clr:= X1, C:= 1024, PV:= R0, IsUp=> C1024); IF C1024 THEN Y1 := TRUE; ELSE Y1 := FALSE; END_IF</pre>	“計數到” 信號間接由 C1024 接點取出之範例	
階段圖	ST							
	<pre>Counter(Pulse:= X0, Clr:= X1, C:= 1024, PV:= R0, IsUp=> C1024); IF C1024 THEN Y1 := TRUE; ELSE Y1 := FALSE; END_IF</pre>							
“計數到” 信號間接由 C1024 接點取出之範例								
								
註：計數器之設定值 PV 若為 0，則 PLC 一開始 RUN 其計數到輸出接點立刻為 1 (但必須 CLR 輸入為 0)，且一直為 1 而不管 CV 值如何變化，直到 CLR 輸入為 1 才會清除。								

6-3 設定(SET)

SET DP		設定 (SET) (將線圈或暫存器之所有位元設為 1)												SET DP			
指令說明																	
階段圖符號 設定控制 — EN — DP SET D								運算元 D：設定之對象 (線圈或暫存器號碼) 。									
範圍 運算元	Y	M	SM	S	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	
	Y0 Y1023	M0 M9119	M9120 M25599	S0 S3103	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	
	D	○	○	○*	○	○			○	○	○	○	○	○	○*	○*	○
功能敘述																	
● 當設定控制“EN” =1 或由 0→1 (P 指令) 時，執行設定動作 (將線圈或暫存器之所有位元設為 1) 。																	
程式範例 1		單一線圈設定															
階段圖 X0 — EN — SET Y0 X1 — EN — RST Y0								ST IF X0 THEN Y0 := TRUE; END_IF IF X1 THEN Y0 := FALSE; END_IF									

SET

DP

設定 (SET)

(將線圈或暫存器之所有位元設為 1)

SET

DP

X0

X1

Y0

SET

RST

程式範例 2

16 位元暫存器設定例

階段圖

ST

X0

EN

SET

R0

IF X0 THEN

R0 := -1;

END_IF

B15

B0

↓

D

R0

1

0

1

1

0

1

1

0

1

1

1

0

1

1

0

0

↕ X0 = ↗

B15

B0

↓

D

R0

1

1

1

1

1

1

1

1

1

1

1

1

1

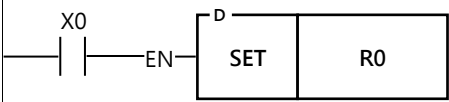
1

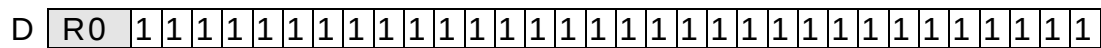
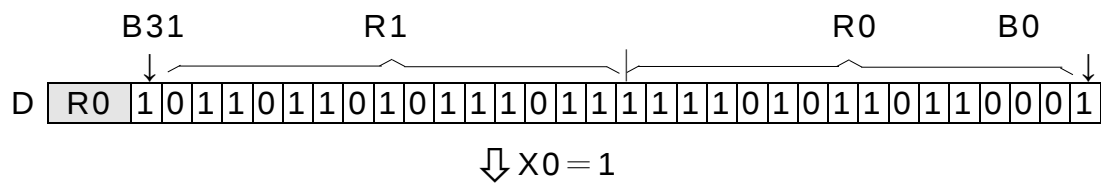
1

1

1

程式範例 3 32 位元暫存器設定例

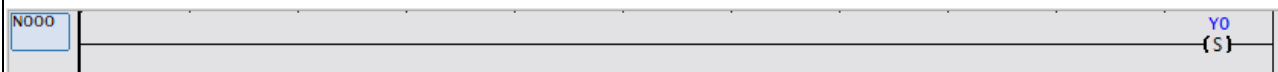
階段圖	ST
	<pre>IF X0 THEN DR0 := -1; END_IF</pre>



註:

如果是使用單一線圈的 set (支援 Y、M、S)，建議可以使用“線圈設定”，使用上 PLC 的效率會比指令的 set 好

範例如下圖:



為將 Y0 SET 住

6-4 清除(RST)

RST DP		清除 (RESET) (將線圈或暫存器之所有位元清為 0)												RST DP			
指令說明																	
階梯圖符號 清除控制 — EN — DP RST D								運算元 D：清除之對象 (線圈或暫存器號碼)									
範圍 運算元		Y	M	SM	S	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	
		Y0 Y1023	M0 M9119	M9120 M25599	S0 S3103	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	
D		○	○	○*	○	○	○	○	○	○	○	○	○	○*	○*	○	

功能敘述																	

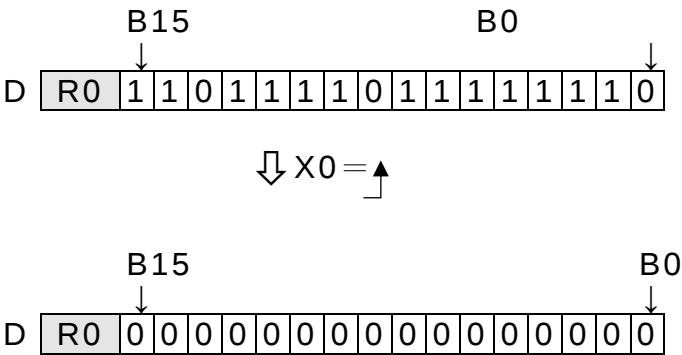
- 當清除控制“EN” =1 或由 0→1 (**P** 指令) 時，將 D (線圈或暫存器) 之所有位元清除為 0。

程式範例 1		單一線圈清除例															

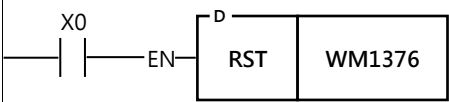
請參考 SET 指令程式範例 1 之說明。

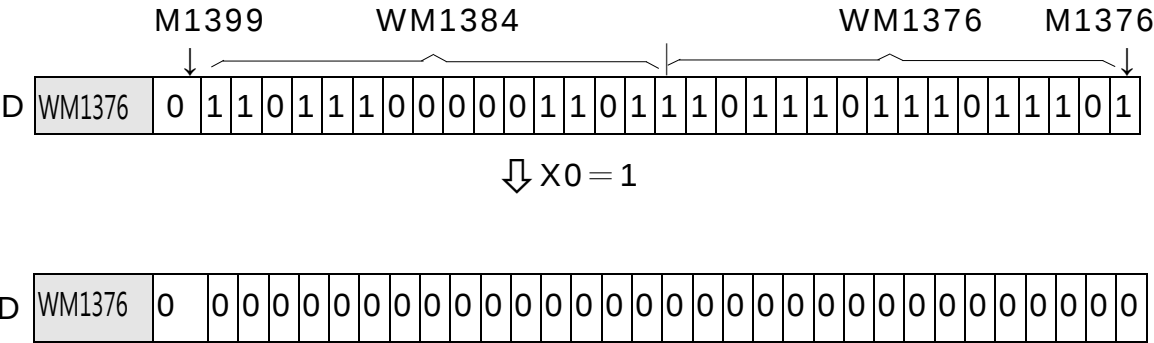
程式範例 2		16 位元暫存器清除例															

階梯圖								ST							
X0 — EN — RST R0								IF X0 THEN R0 := 0; END_IF							

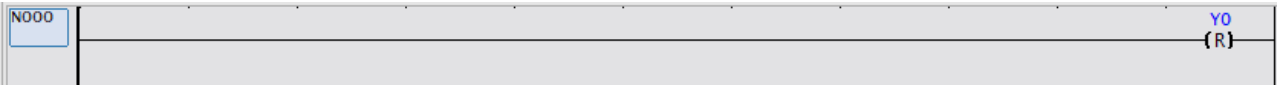


程式範例 3	32 位元暫存器清除例
--------	-------------

階梯圖	ST
	<pre>IF X0 THEN WM1376 := 0; END_IF</pre>



註:
如果是使用單一線圈的 reset (支援 Y、M、S)，建議可以使用” 線圈復歸 ”，使用上 PLC 的效率會比指令的 reset 好
範例如下圖:

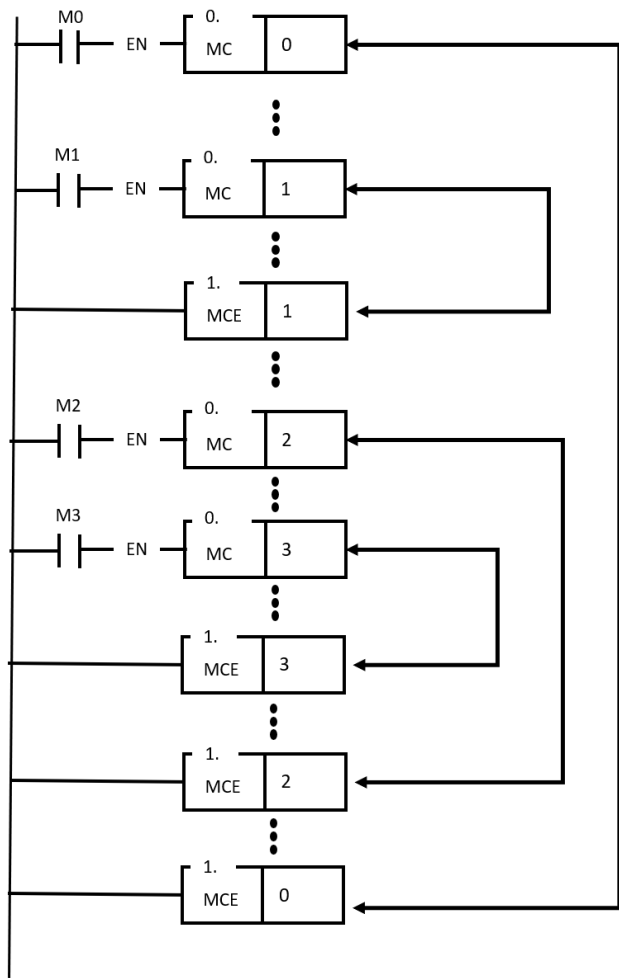


為將 Y0 Reset 掉

6-5 主控迴路開始指令(MC)

FUN0 MC	主控 (MASTER CONTROL) 回路開始指令		FUN0 MC
指令說明			
階段圖符號 主控輸入 — EN / — 0. MC N		運算元 N：主控回路之號碼 (N=0 ~ 127)，號碼不得重複。	
功能敘述			
● MC 回路總共可有 128 個 (N=0 ~ 127)。每個主控回路 MC N 指令均要有一個相同號碼之主控回路終止指令 MCE N 與之對應 (但需確保 MCE N 指令要在 MC N 指令之後)。 ● 當主控輸入 “EN/” 為 1 則此指令不執行 (等效 MC N 指令不存在)。 ● 當主控輸入 “EN/” 為 0 則主控回路動作，由 MC N 指令開始，一直到相同號碼之 MCE N 指令間 (稱之為主控回路動作區) 之程式，若為 OUT 線圈或一般計時器將其狀態均清為 0，其他指令則不執行。			
程式範例			

FUN0 MC	主控 (MASTER CONTROL) 回路開始指令	FUN0 MC				
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table> 			階梯圖	ST		
階梯圖	ST					

FUN0 MC	主控 (MASTER CONTROL) 回路開始指令	FUN0 MC
註 1：MC/MCE 指令可作多層巢狀使用。如右例：		

6-6 主控終止指令(MCE)

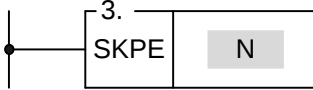
FUN1 MCE	主控終止 (MASTER CONTROL END) 指令		FUN1 MCE
指令說明			
階段圖符號 1. MCE N		運算元 N：終止之主控回路號碼 (N=0 ~ 127)，號碼不得重複。	
功能敘述			
● MCE N 是搭配 MC N 使用，單獨存在並無意義。在 MC N 指令動作後，其後之程式若為 OUT 線圈或一般計時器狀態均被清為 0，其他指令則不執行，直到遇到相同號碼 (N) 之 MCE 指令，才會解除主控動作，恢復正常之程式執行動作。 ● MCE 指令無需輸入控制，其本身自成一個網路，不能串接其他指令，在程式執行中只要遇到 MCE N 指令，若已發生 MC N 主控動作則此指令會將主控動作解除，若未發生則此指令無效 (無任何影響)。			
程式範例 1			
● 請參閱 MC 指令之範例說明。			

6-7 跳過迴路開始指令(SKIP)

FUN2 SKP	跳過 (SKIP) 回路之開始指令		FUN2 SKP
指令說明			
跳過控制 — EN — 2. SKP N		階段圖符號 運算元 N：跳過回路之號碼 (N=0 ~ 127)，號碼不得重複。	
功能敘述	● SKP 回路共有 128 個 (N=0 ~ 127)。每個跳過回路開始指令 SKP N 至少要有一個相同號碼之跳過回路終止指令 SKPE N 與之對應 (但需確保 SKPE N 指令要在 SKP N 指令之後)。 ● 當跳過控制“EN”為 0，此指令不執行 (等效 SKP N 指令不存在)。 ● 當跳過控制“EN”為 1 則跳過回路動作，在 SKP N 以後一直到遇到相同號碼之 SKPE N 指令間 (稱之為跳過回路動作區) 之程式均不執行 (跳過)。故此區域內所有之單點或暫存器狀態均保持不變。		

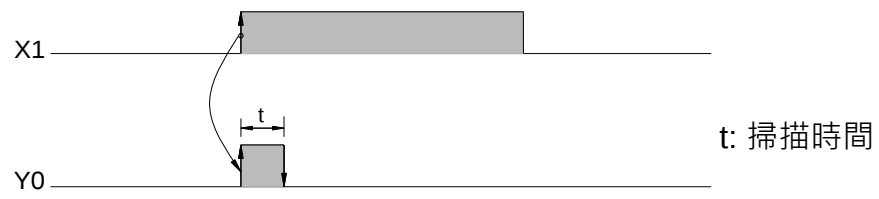
FUN2 SKP	跳過 (SKIP) 回路之開始指令	FUN2 SKP				
程式範例						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td> The diagram shows three rungs of a ladder logic circuit. Rung 1: Input X0 (normally open) is connected to the EN (enable) input of an SKP (Skip) instruction. The SKP instruction has a step number of 02 and a coil output Y0 with a step number of 1. Rung 2: Input X1 (normally open) is connected to the coil output Y0. Rung 3: Input X2 (normally open) is connected to the EN input of a T1023 (Timer) instruction. The T1023 instruction has a step number of 15 and a coil output Y1 with a step number of 10. Below the rungs, there is a SKPE (Skip) instruction with a step number of 03 and a coil output Y2 with a step number of 1. Input X1 (normally open) is also connected to the coil output Y2. </td><td></td></tr></table>			階段圖	ST	The diagram shows three rungs of a ladder logic circuit. Rung 1: Input X0 (normally open) is connected to the EN (enable) input of an SKP (Skip) instruction. The SKP instruction has a step number of 02 and a coil output Y0 with a step number of 1. Rung 2: Input X1 (normally open) is connected to the coil output Y0. Rung 3: Input X2 (normally open) is connected to the EN input of a T1023 (Timer) instruction. The T1023 instruction has a step number of 15 and a coil output Y1 with a step number of 10. Below the rungs, there is a SKPE (Skip) instruction with a step number of 03 and a coil output Y2 with a step number of 1. Input X1 (normally open) is also connected to the coil output Y2.	
階段圖	ST					
The diagram shows three rungs of a ladder logic circuit. Rung 1: Input X0 (normally open) is connected to the EN (enable) input of an SKP (Skip) instruction. The SKP instruction has a step number of 02 and a coil output Y0 with a step number of 1. Rung 2: Input X1 (normally open) is connected to the coil output Y0. Rung 3: Input X2 (normally open) is connected to the EN input of a T1023 (Timer) instruction. The T1023 instruction has a step number of 15 and a coil output Y1 with a step number of 10. Below the rungs, there is a SKPE (Skip) instruction with a step number of 03 and a coil output Y2 with a step number of 1. Input X1 (normally open) is also connected to the coil output Y2.						
The timing diagram illustrates the sequence of events. X0 is active for a short duration. X1 is active for a longer duration. X2 is active after X0. T1023 starts a 10-second ramp when X2 is active. Y0 is active during the first pulse of X0 and the first pulse of X1. Y1 is active during the 10-second ramp of T1023. Y2 is active during the first pulse of X1 and the first pulse of X2.						

6-8 跳過迴路終止指令(SKPE)

FUN3 SKPE	跳過回路終止 (SKIP END) 指令		FUN3 SKPE
指令說明			
階段圖符號 		運算元 N：終止之跳過回路號碼 (N=0 ~ 127)，號碼不得重複。	
功能敘述			
● SKPE N 指令係搭配 SKP N 指令使用，單獨存在並無意義。在 SKP N 跳過指令動作後，其後之程式即跳過不執行，所有狀態亦不變，一直要到遇到相同號碼 (N) 之 SKPE 指令，才會解除跳過動作，恢復正常之程式執行動作。 ● SKPE 指令無需輸入控制，其本身即是一個網路，不能串接其他指令，在程式執行中只要遇到 SKPE N 指令，若已發生 SKP N 之跳過動作，則此指令會立即將跳過動作解除。若未發生，則此指令無效 (無任何影響)。			
程式範例 1			
請參閱 SKP 指令之範例說明。 註：SKP/SKPE 指令可作多層巢式，其用法規則和 MC/MCE 指令之巢式或交錯用法完全相同，請參閱 MC/MCE 指令。			

6-9 上微分指令(DIFU)

FUN4 P DIFU	上微分 (DIFFERENTIAL UP) 指令	FUN4 P DIFU																		
指令說明																				
階段圖符號 狀態輸入—TGU— 4. DIFU D		運算元 D：存放上微分結果之繼電器線圈號碼。																		
<table><tr><td rowspan="4">範圍 運算元</td><td>Y</td><td>M</td><td>SM</td><td>S</td></tr><tr><td>Y0</td><td>M0</td><td>M9120</td><td>S0</td></tr><tr><td> Y1023</td><td> M9119</td><td> M25599</td><td> S3103</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○*</td><td>○</td></tr></table>			範圍 運算元	Y	M	SM	S	Y0	M0	M9120	S0	 Y1023	 M9119	 M25599	 S3103	D	○	○	○*	○
範圍 運算元	Y	M		SM	S															
	Y0	M0		M9120	S0															
	 Y1023	 M9119		 M25599	 S3103															
	D	○	○	○*	○															
功能敘述																				
● DIFU 指令係將狀態輸入“TGU”之狀態取上微分（在 TGU 升緣變換時產生一波寬為掃描時間 T 之單擊脈波）後，將此單擊脈波信號存入 D 所指定之線圈。 ● 本指令可以用順序指令之 TU 接點直接取代，其較本指令方便好用。																				
程式範例	以下兩例結果完全相同																			
<table><tr><td>階段圖</td><td>ST</td></tr><tr><td> 例 1 X1 — —TGU— 04 DIFU Y0 </td><td> R_TRIG(S:= X1, D=>Y0); </td></tr><tr><td> 例 2 X1 —↑↑— Y0 () </td><td> R_TRIG(S:= X1, D=>Y0); </td></tr></table>			階段圖	ST	例 1 X1 — —TGU— 04 DIFU Y0	R_TRIG(S:= X1, D=>Y0);	例 2 X1 —↑↑— Y0 ()	R_TRIG(S:= X1, D=>Y0);												
階段圖	ST																			
例 1 X1 — —TGU— 04 DIFU Y0	R_TRIG(S:= X1, D=>Y0);																			
例 2 X1 —↑↑— Y0 ()	R_TRIG(S:= X1, D=>Y0);																			

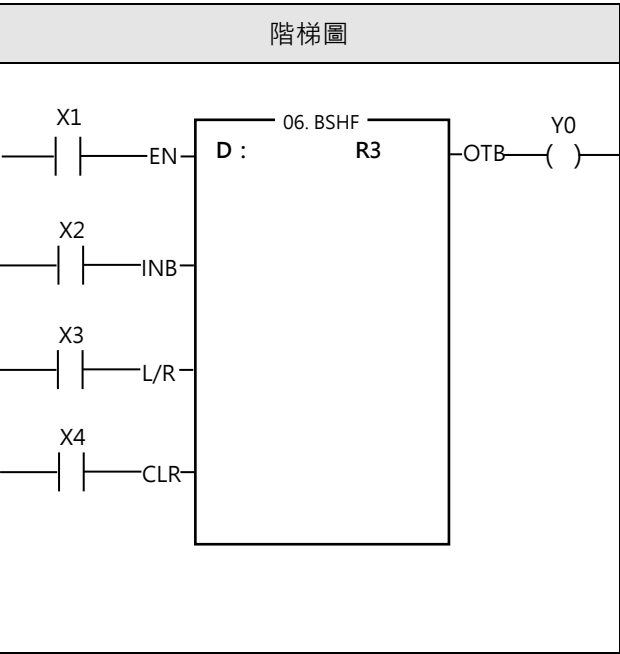
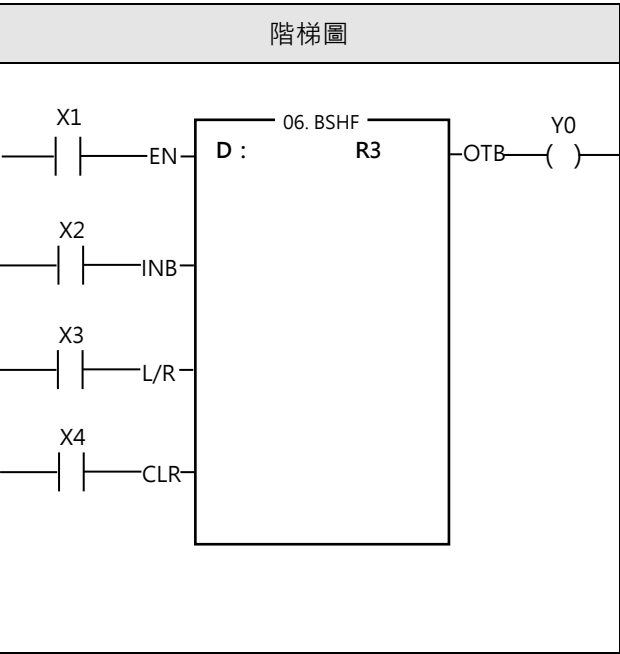
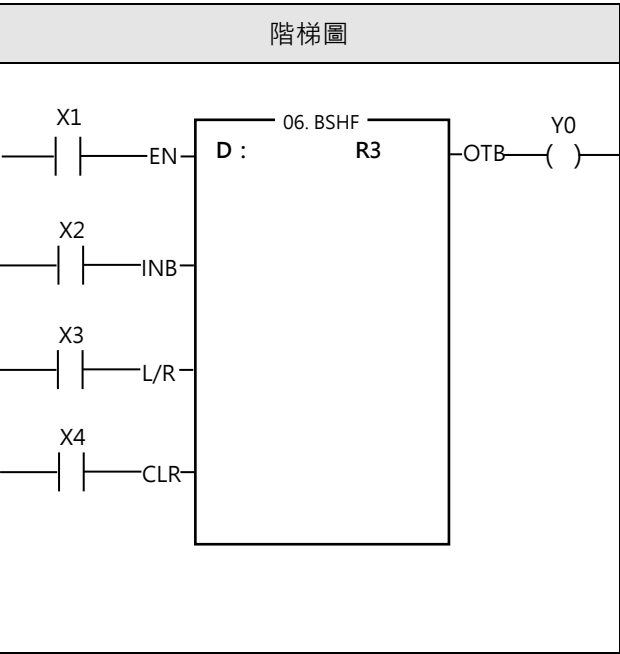
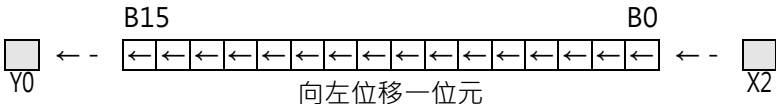
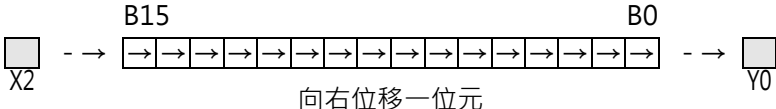


6-10 下微分指令(DIFD)

FUN5 P DIFD	下微分 (DIFFERENTIAL DOWN) 指令				FUN5 P DIFD																						
指令說明																											
階段圖符號			運算元																								
狀態輸入 — TGD — 5. <table><tr><td>DIFD</td><td>D</td></tr></table>			DIFD	D	D：存放上微分結果之繼電器線圈號碼。																						
DIFD	D																										
<table><tr><td rowspan="4">範圍 運算元</td><td>Y</td><td>M</td><td>SM</td><td>S</td></tr><tr><td>Y0</td><td>M0</td><td>M9120</td><td>S0</td></tr><tr><td> </td><td> </td><td> </td><td> </td></tr><tr><td>Y1023</td><td>M9119</td><td>M25599</td><td>S3103</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○*</td><td>○</td></tr></table>						範圍 運算元	Y	M	SM	S	Y0	M0	M9120	S0					Y1023	M9119	M25599	S3103	D	○	○	○*	○
範圍 運算元	Y	M	SM	S																							
	Y0	M0	M9120	S0																							
	Y1023	M9119	M25599	S3103																							
D	○	○	○*	○																							
功能敘述																											
- DIFD 指令係將狀態輸入“TGD”之狀態取下微分（在 TGD 降緣變換時產生一波寬為掃描時間 T 之單擊脈波）後，將此單擊脈波信號存入 D 所指定之線圈內。 - 本指令可以用順序指令之 TD 接點直接取代，其較本指令方便好用																											
程式範例																											
以下兩例結果完全相同																											
階段圖			ST																								
例 1 X1 TGD 5P. DIFD Y 0			F_TRIG(S:= X1, D=>Y0);																								
例 2 X1 Y0			F_TRIG(S:= X1, D=>Y0);																								

6-11 位元位移(BSHF)

FUN6 DP BSHF	位元位移 (BIT SHIFT) (將 16 或 32 位元暫存器資料向左或右位移一位元)										FUN6 DP BSHF																																				
指令說明																																															
階梯圖符號 位移控制 — EN 填補位元 — INB 位移方向 — L/R 清除控制 — CLR D : OTB — 移出位元 (FO0)						運算元 D : 位移之暫存器號碼。																																									
<table><tr><th>範圍 運算元</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th></tr><tr><td></td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C12789</td><td>R0 R34767</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>○</td></tr></table>												範圍 運算元	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR		WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C12789	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	D	○	○	○	○	○	○	○	○	○*	○*	○
範圍 運算元	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR																																				
	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C12789	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999																																				
D	○	○	○	○	○	○	○	○	○*	○*	○																																				
功能敘述																																															
● 當清除控制“CLR”為 1 時，D 之資料及 FO0 均清為 0，其他輸入信號均無效。 ● 當清除控制“CLR”為 0 時，則允許位移動作。當位移控制“EN” =1 或由 0→1 (P 指令) 時將 D 之資料向左或向右位移一位元 (方向由位移方向“L/R”之輸入來控制，1 為左移，0 為右移)。而位移所擠出之位元則送到 FO0 去 (此位元在左移時為 MSB，右移時為 LSB)。而位移所空出之位置 (左移為 LSB，右移為 MSB) 則以填補位元“INB”輸入之狀態填補之。																																															

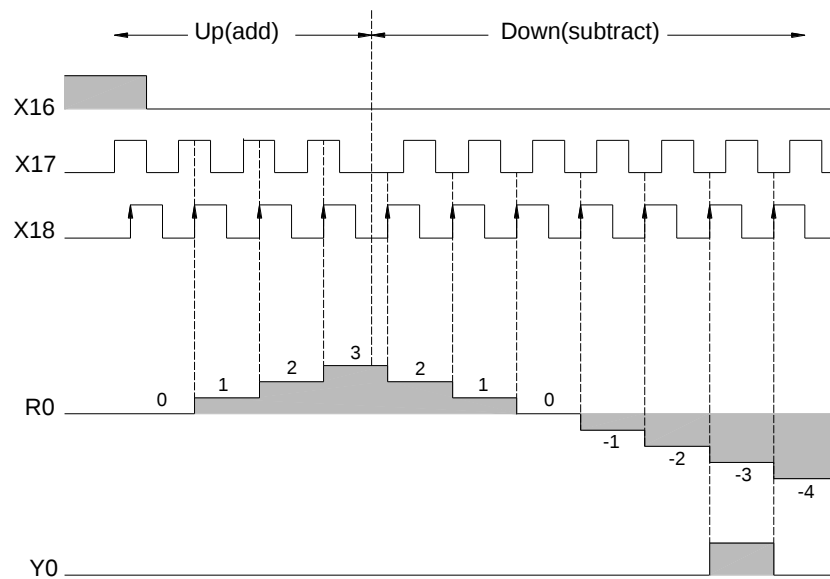
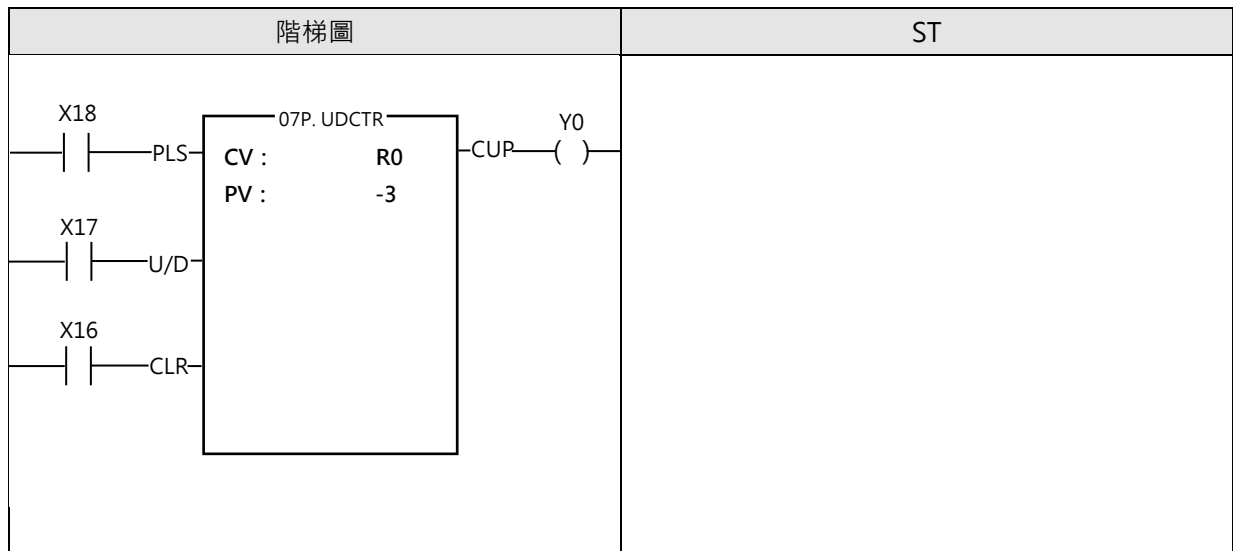
FUN6 DP BSHF	位元位移 (BIT SHIFT) (將 16 或 32 位元暫存器資料向左或右位移一位元)	FUN6 DP BSHF				
程式範例	16 位元暫存器資料位移例					
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table>			階梯圖	ST		
階梯圖	ST					
						
X3=1 (左移)						
X3=0 (右移)						

6-12 上/下數計數器(UDCTR)

FUN7 D P UDCTR		上 / 下數計數器 (UP/DOWN COUNTER) (16 位元或 32 位元可上數、下數之雙相計數器)										FUN7 D P UDCTR			
指令說明															
		</													

- 上數計數值之上限為 32767 (16 位元) 或 2147483647 (32 位元)，到達上限後，若再來個上數計數脈波，計數值將會變成-32768 或-2147483648 (下數之最下限)。
- 下數之最下限為-32768 或-2147483648，達此值後，若再來一個下數計數脈波，則計數值會跳到 32767 或 2147483647 (上數之最上限)。
- 若將 U/D 固定為 1，則本指令將變成單相上數計數器，反之若固定為 0 則變成單相下數計數器。

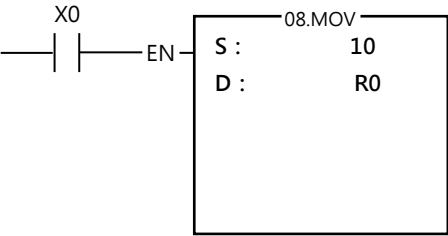
程式範例



FUN7  UDCTR	上 / 下數計數器 (UP/DOWN COUNTER) (16 位元或 32 位元可上數、下數之雙相計數器)	FUN7  UDCTR
	● 註 1：本指令在每一個掃描週期會依據 PLS 訊號的變化做計數動作，因此只要訊號變化後持續時間超過掃描週期，此指令就可以偵測到變化。但掃描週期時間會依據使用者程式的多寡而變化，若要得到精確的計數值，請使用軟體或硬體之高速計數器。 ● 註 2：為確保本指令能夠正確計數，計數脈波之波寬無論為 1 或 0，皆必須大於一個掃描時間。	

6-13 搬移(MOV)

FUN8 DP MOV		搬移 (MOVE) (將 S 之資料搬移到 D 去)												FUN8 DP MOV																																																																		
指令說明																																																																																
階段圖符號 搬移控制 — EN — 8DP.MOV S : D : <td colspan="8"> 運算元 S : 來源 (Source) 資料或其暫存器號碼。 D : 搬移之目的 (Destination) 暫存器號碼。 S,D 可結合 V、Z、P0~P9 作間接定址應用。 </td>										運算元 S : 來源 (Source) 資料或其暫存器號碼。 D : 搬移之目的 (Destination) 暫存器號碼。 S,D 可結合 V、Z、P0~P9 作間接定址應用。																																																																						
<table><tr><td rowspan="4">範圍 運算元</td><td>WX</td><td>WY</td><td>WM</td><td>WSM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>WS0 WS3008</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 或 32 位元 正、負 數</td><td>V、Z P0-P9</td></tr><tr><td>S</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td></tr><tr><td>D</td><td></td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td>☐</td><td></td><td>☐</td><td>☒</td><td>☒</td><td>☐</td><td></td><td>☐</td></tr></table>																		範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3008	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9	S	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	D		☐	☐	☐	☐	☐	☐	☐		☐	☒	☒	☐		☐
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																	
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3008	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9																																																																	
	S	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐																																																																
	D		☐	☐	☐	☐	☐	☐	☐		☐	☒	☒	☐		☐																																																																
功能敘述																																																																																
● 當搬移控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 之資料搬移 (寫入) 到 D 去。																																																																																
程式範例		16 位元暫存器搬移例																																																																														

FUN8 DP MOV	搬移 (MOVE) (將 S 之資料搬移到 D 去)	FUN8 DP MOV
階段圖 		ST <pre> IF X0 THEN R0 := 10; END_IF </pre>
S 常數 10 ⇩ X0 = ⇧ D R0 10		

6-14 倒相後搬移(MOV/)

FUN9 D P MOV/		倒相後搬移 (MOVE INVERSE) (將 S 之資料倒相後搬移到 D 去)										FUN9 D P MOV/				
指令說明																
搬移控制 — EN — 階段圖符號 9DP.MOV/ S : D :										運算元 S : 來源 (Source) 資料或其暫存器號碼。 D : 搬移之目的 (Destination) 暫存器號碼。 S,D 可結合 V、Z、P0~P9 作間接定址應用。						
範圍 運算元		WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3008	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9
S																
D												*	*			
功能敘述																
		● 當搬移控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 之資料倒相 (狀態為 0 之位元變為 1，為 1 者變為 0) 後，搬移 (寫入) 到 D 去。														
程式範例		16 位元暫存器倒相搬移例														

FUN9 DP MOV/	倒相後搬移 (MOVE INVERSE) (將 S 之資料倒相後搬移到 D 去)	FUN9 DP MOV/
階段圖 X0 — —EN— S : R0 D : WY16		ST IF X0 THEN WY16 := NOT R0; END_IF

S

B15

R0

0

1

0

1

0

1

0

1

0

1

0

1

0

1

B0

5555H

⇓ X0=1

D

Y31

WY16

1

0

1

0

1

0

1

0

1

0

1

0

1

0

Y16

AAAAH

6-15 交替開關(TOGG)

FUN10 TOGG	交替開關 (TOGGLE SWITCH) (輸入每觸動一次，輸出 D 轉態一次)				FUN10 TOGG														
指令說明																			
階段圖符號 觸發輸入—TGU 10. TOGG D			運算元 D：交替開關線圈號碼																
<table><tr><td rowspan="4">範圍 運算元</td><td>Y</td><td>M</td><td>SM</td><td>S</td></tr><tr><td>Y0 Y1023</td><td>M0 M9119</td><td>M9120 M25599</td><td>S0 S3103</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○*</td><td>○</td></tr></table>						範圍 運算元	Y	M	SM	S	Y0 Y1023	M0 M9119	M9120 M25599	S0 S3103	D	○	○	○*	○
範圍 運算元	Y	M	SM	S															
	Y0 Y1023	M0 M9119	M9120 M25599	S0 S3103															
	D	○	○	○*	○														
	功能敘述																		
● 當觸發輸入“TGU” 每由 0→1 一次，D 線圈之狀態就交替轉態一次（ 0 變 1，1 則變 0 ）。																			
程式範例																			
階段圖			ST																
			<pre>IF X0 THEN Y0 := NOT Y0; END_IF</pre>																

6-16 快速加法運算 F (+)

FUN224 DP F (+)		快速加法運算 (ADDITION) (將 Sa 加 Sb 之和存入 D)												FUN224 DP F (+)																																																											
指令說明		於 UperLogic: v_1.0.0 以後支援																																																																							
		階段圖符號 224DPU/DPS.F(+) 執行控制 —EN— Sa: Sb: D:								運算元 Sa：被加數或其暫存器號碼。 Sb：加數或其暫存器號碼。 D：存放結果(和)之暫存器號碼。 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用。																																																															
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																										
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9																																																										
	Sa	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐																																																										
	Sb	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐																																																										
	D	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐																																																										
功能敘述		● 當加算控制 “EN” =1 或由 0→1 (P 指令) 而指令中設定為 signed (S 指令) 時，將 Sa 與 Sb 以正負數 (Sign) 運算法則作加法運算並將結果寫入 D 去。 ● 當加算控制 “EN” =1 或由 0→1 (P 指令) 而指令中設定為 unsigned (U 指令) 時，將 Sa 與 Sb 以正整數 (Unsign) 運算法則作加法運算並將結果寫入 D 去。 ● 與 FUN11 的加法運算相比,快速加法運算因為其省去了溢位及欠位的運算及旗標,因此其程式執行所需時間會較 FUN11 的加法運算快速,其運算結果也會與一般在 PC 使用高階語言如 C 語言的結果一樣如下圖左側結果, 另外也會與 FUN11 的加法運算的計算結果在數值邊界時會不相同(以正負數 (Sign) 運算法為例)。 <table><tr><td>R10</td><td>十六進制</td><td>7FFFH</td><td>被加數</td><td></td><td></td><td></td><td></td></tr><tr><td>R11</td><td>十六進制</td><td>0001H</td><td>加數</td><td></td><td></td><td></td><td></td></tr><tr><td>R12</td><td>十六進制</td><td>8000H</td><td>快速加法運算結果</td><td>R15</td><td>十六進制</td><td>0000H</td><td>加法運算結果</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>R10</td><td>十進制</td><td>32767</td><td>被加數</td><td></td><td></td><td></td><td></td></tr><tr><td>R11</td><td>十進制</td><td>1</td><td>加數</td><td></td><td></td><td></td><td></td></tr><tr><td>R12</td><td>十進制</td><td>-32768</td><td>快速加法運算結果</td><td>R15</td><td>十進制</td><td>0</td><td>加法運算結果</td></tr></table>																R10	十六進制	7FFFH	被加數					R11	十六進制	0001H	加數					R12	十六進制	8000H	快速加法運算結果	R15	十六進制	0000H	加法運算結果									R10	十進制	32767	被加數					R11	十進制	1	加數					R12	十進制	-32768	快速加法運算結果	R15	十進制	0	加法運算結果
R10	十六進制	7FFFH	被加數																																																																						
R11	十六進制	0001H	加數																																																																						
R12	十六進制	8000H	快速加法運算結果	R15	十六進制	0000H	加法運算結果																																																																		
R10	十進制	32767	被加數																																																																						
R11	十進制	1	加數																																																																						
R12	十進制	-32768	快速加法運算結果	R15	十進制	0	加法運算結果																																																																		

FUN224 DP F (+)	快速加法運算 (ADDITION) (將 Sa 加 Sb 之和存入 D)	FUN224 DP F (+)
程式範例	Signed 範例	

階段圖	ST
X0 — — EN 224S.F(+) Sa : R0 Sb : R1 D : R2	IF X0 THEN R2 := R0 + R1; END_IF

Sa	R0	12345
Sb	R1	20425

R0 + R1 = 32770(32767+3)

↓ X0 = ↑

D	R2	-32766
---	----	--------

32767+3>>-32768+2>>-32766

※當加超過 32767(0x7FFF)的時候會變成
-32768(0x8000)、-32767(0x8001)、-32766(0x8002)...-1(0xFFFF)、
0(0x0000)

FUN224 DP F (+)	快速加法運算 (ADDITION) (將 Sa 加 Sb 之和存入 D)	FUN224 DP F (+)
程式範例	Unsigned 範例	

階梯圖	ST
X0 — — EN 224S.F(+) Sa : R0 Sb : R1 D : R2	IF X0 THEN R2 := R0 + R1; END_IF

Sa	<table><tr><td>R0</td><td>12345</td></tr><tr><td>R1</td><td>20425</td></tr></table>	R0	12345	R1	20425	R0 + R1 = 32770
R0	12345					
R1	20425					

⇓ X0 =

⇑

D	<table><tr><td>R2</td><td>32770</td></tr></table>	R2	32770
R2	32770		

6-17 快速減法運算 F (-)

FUN225 D P F (-)		快速減法運算 (SUBTRACTION) (將 Sa 減 Sb 之和存入 D)														FUN225 D P F (-)																																																																														
指令說明		於 UperLogic: v_1.0.0 以後支援																																																																																												
		階段圖符號 225DPU/DPS.F(-) 執行控制—EN— Sa: Sb: D:										運算元 Sa：被減數或其暫存器號碼。 Sb：減數或其暫存器號碼。 D：存放結果(差)之暫存器號碼。 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用																																																																																		
<table><tr><th rowspan="4">範圍 運算元</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>WS0 WS3008</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 或 32 位元 正、負 數</td><td>V、Z P0-P9</td></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>																		範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3008	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○		○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																															
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3008	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9																																																																															
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																															
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																															
D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																															
功能敘述		- 當減算控制“EN” =1 或由 0→1 (P 指令) 而指令中設定為 signed (S 指令) 時，將 Sa 與 Sb 以正負數 (Sign) 運算法則作減法運算並將結果寫入 D 去。 - 當減算控制“EN” =1 或由 0→1 (P 指令) 而指令中設定為 unsigned (U 指令) 時，將 Sa 與 Sb 以正整數 (Unsigned) 運算法則作減法運算並將結果寫入 D 去。 - 與 FUN12 的減法運算相比,快速減法運算因為其省去了溢位及欠位的運算及旗標,因此其程式執行所需時間會較 FUN12 的減法運算快速,其運算結果也會與一般電腦計算後的結果一樣如下一頁的圖左側結果,另外也會與 FUN12 的減法運算的計算結果在數值邊界時會不相同																																																																																												

FUN225 D P F (-)		快速減法運算 (SUBTRACTION) (將 Sa 減 Sb 之和存入 D)				FUN225 D P F (-)	
十六進制	8000H	被減數					
十六進制	0001H	減數					
十六進制	7FFFH	快速減法運算結果	R5	十六進制	FFFFH	減法運算結果	
十進制	-32768	被減數					
十進制	1	減數					
十進制	32767	快速減法運算結果	R5	十進制	-1	減法運算結果	

程式範例	Signed 範例
------	-----------

階梯圖	ST
X0 — — EN 225S.F(-) Sa : R0 Sb : R1 D : R2	IF X0 THEN R2 := R0 - R1; END_IF

Sa	R0	- 5
Sb	R1	32767

↓ X0 = 1

D	R2	32764
---	----	-------

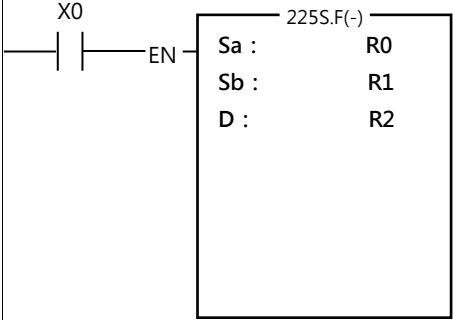
- 32772>>-32768-4>>32767-3

>>32764

※ 當減超過 -32768(0x8000) 會變成 32767(0x7FFF) 、32766(0x7FFE)....0(0x000)

FUN225 D P F (-)	快速減法運算 (SUBTRACTION) (將 Sa 減 Sb 之和存入 D)	FUN225 D P F (-)
------------------------------	--	------------------------------

程式範例	Unsigned 範例
------	-------------

階梯圖	ST
	<pre>IF X0 THEN R2 := R0 - R1; END_IF</pre>

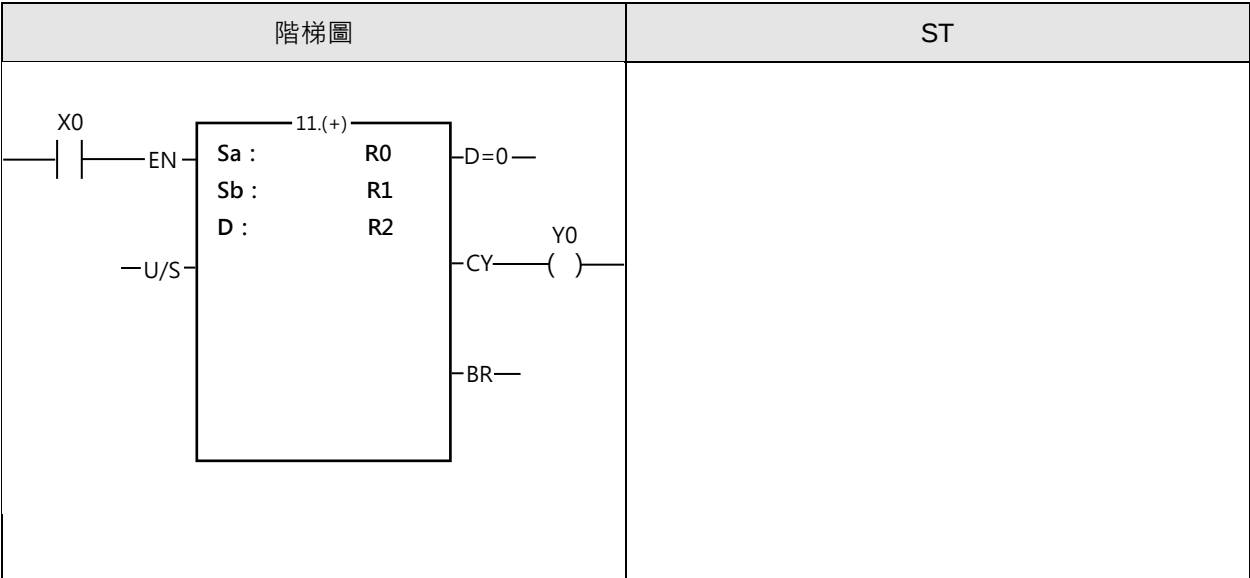
Sa	R0	-5(65531)	R0 - R1 = 32764
Sb	R1	32767	
⇓ X0 = 1			
D	R2	32764	

6-18 加法運算 (+)

FUN11 DP (+)		加法運算 (ADDITION) (將 Sa 加 Sb 之和存入 D)														FUN11 DP (+)																																																																																	
指令說明																																																																																																	
		階段圖符號 加算控制 — EN 正/負數選擇 — U/S 11DP.(+) Sa : Sb : D : D=0 — 和=0 (FO0) CY — 進位 (FO1) BR — 借位 (FO2)										運算元 Sa：被加數或其暫存器號碼。 Sb：加數或其暫存器號碼。 D：存放結果(和)之暫存器號碼。 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用。																																																																																					
		<table><tr><th>範圍 運算元</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 或 32 位元 正、負 數</td><td>V、Z P0-P9</td></tr><tr><td>fSa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>																範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9	fSa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○		○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																		
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9																																																																																		
fSa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																		
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																		
D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																		
功能敘述		● 當加算控制“EN” =1 或由 0→1 (P 指令) 而“U/S” =0 時，將 Sa 與 Sb 以正負數 (Sign) 運算法則作加法運算並將結果寫入 D 去。同時若和為 0，則 FO0 (D=0) 設為 1，若發生進位 (和超過 32767 或 2147483647)，則將 FO1 (CY) 設為 1，若發生借位 (負數加負數，使和小於-32768 或 -2147483648)，則 FO2 (BR) 設為 1。所有 FO 之狀態均維持到本指令下次執行時才被新執行結果所取代。 ● 當加算控制“EN” =1 或由 0→1 (P 指令) 而“U/S” =1 時，將 Sa 與 Sb 以正整數 (Unsign) 運算法則作加法運算並將結果寫入 D 去。同時若和為 0，則 FO0 (D=0) 設為 1，若發生進位 (和超過 65535 或 4294967295)，則進位 FO1 (CY) 設為 1。																																																																																															

FUN11 DP (+)	加法運算 (ADDITION) (將 Sa 加 Sb 之和存入 D)	FUN11 DP (+)
--------------------------	---	--------------------------

程式範例



Sa	R0	12345
Sb	R1	20425

R0 + R1 = 32770

↓ X0 = ↑

D	R2	2
---	----	---

32768 + 2 = 32770

Y0 = 1 (進位 1 代表 + 32768)

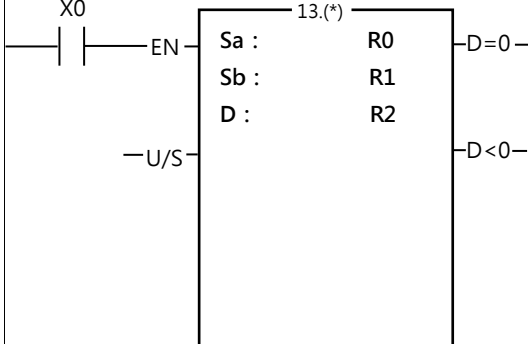
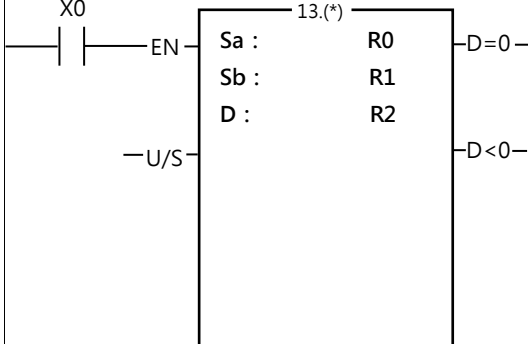
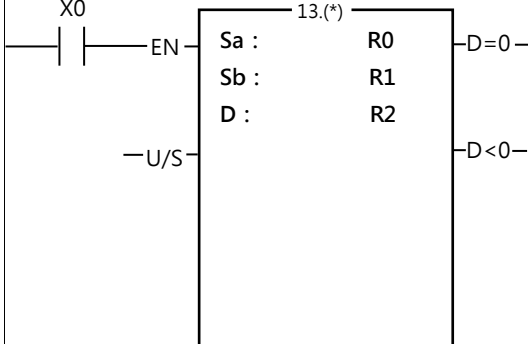
6-19 減法運算 (-)

FUN12 D P (-)		減法運算 (SUBTRACTION) (將 Sa 減 Sb 之和存入 D)														FUN12 D P (-)			
指令說明																			
		階梯圖符號 12DP.(.) 減算控制 — EN — Sa : — D=0 — 差=0 (FO0) Sb : 正/負數選擇 — U/S — D : — CY — 進位 (FO1) BR — 借位 (FO2)										運算元 Sa：被減數或其暫存器號碼。 Sb：減數或其暫存器號碼。 D：存放結果(差)之暫存器號碼。 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用							
範圍 暫存器	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR				
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3008	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0-P9				
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○				
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○				
	D		○	○	○	○	○	○	○		○	○*	○*	○		○			
功能敘述																			
		● 當減算控制 “EN” =1 或由 0→1 (P 指令) 而 “U/S” =0 時，將 Sa 與 Sb 以正負數 (Sign) 運算法則作減法運算並將結果寫入 D 去。同時若差為 0，則 FO0																	

FUN12 D P (-)	減法運算 (SUBTRACTION) (將 Sa 減 Sb 之和存入 D)	FUN12 D P (-)										
(D=0) 設為 1，若發生進位 (正數減負數，使差超過+32767 或+2147483647) 則 FO1 (CY) 設為 1，若借位發生 (負數減正數，使差小於-32768 或-2147483648) 則 FO2 (BR) 設為 1。所有 FO 之狀態均維持到本指令下一次執行時才被新執行結果所取代。 當減算控制“EN” =1 或由 0→1 (P 指令) 而“U/S” =1 時，將 Sa 與 Sb 以正整數 (Unsign) 運算法則作減法運算並將結果寫入 D 去。同時若差為 0，則 FO0 (D=0) 設為 1，若借位發生 (Sa - Sb < 0) 則 FO2 (BR) 設為 1。												
程式範例												
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table> D <table><tr><td>R2</td><td>- 4</td></tr></table> - 32768 - 4 = - 32772 Y2 = 1 (借位 1 代表 - 32768) 請參閱 5.5 節之說明 Sa <table><tr><td>R0</td><td>- 5</td></tr><tr><td>Sb</td><td>32767</td></tr></table> R0 - R1 = - 32772 ↓ X0 = 1			階段圖	ST			R2	- 4	R0	- 5	Sb	32767
階段圖	ST											
R2	- 4											
R0	- 5											
Sb	32767											

6-20 乘法運算 (*)

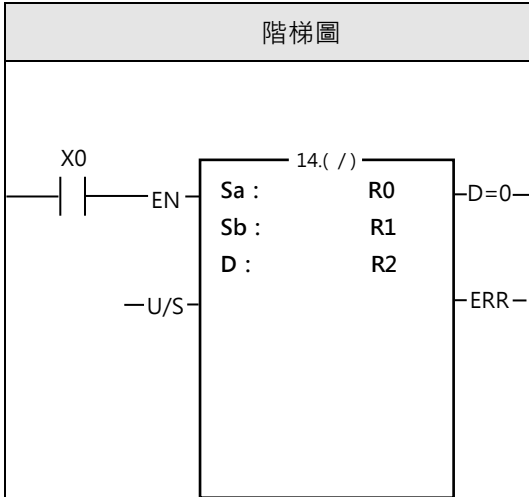
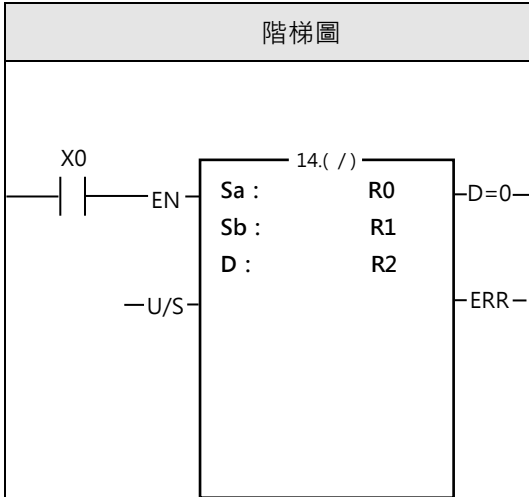
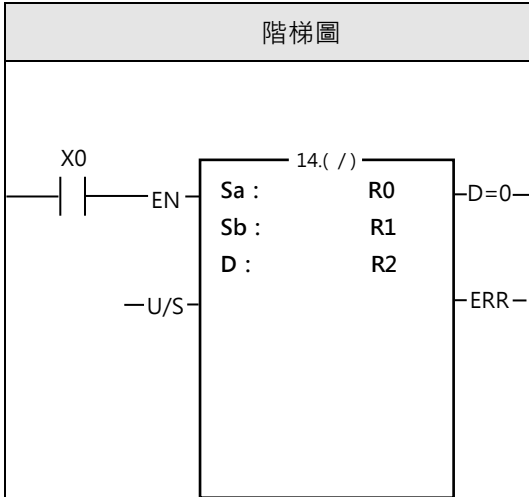
FUN13 D P (*)		乘法運算 (MULTIPLICATION) (將 Sa 乘 Sb 之積存入 D)										FUN13 D P (*)			
指令說明															

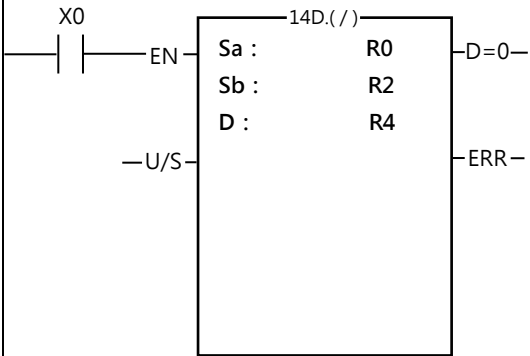
FUN13 D P (*)	乘法運算 (MULTIPLICATION) (將 Sa 乘 Sb 之積存入 D)	FUN13 D P (*)				
程式範例 1	16 位元乘法例					
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table> 被乘數 Sa 12345 × 乘數 Sb 4567 積 D R3R2 56379615			階段圖	ST		
階段圖	ST					
						
程式範例 2	32 位元乘法例					

FUN13 D P (*)	乘法運算 (MULTIPLICATION) (將 Sa 乘 Sb 之積存入 D)	FUN13 D P (*)														
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td> X0 — EN — 13D.(*) Sa : R0 Sb : R2 D : R4 — U/S — D=0 D<0 </td><td></td></tr></table>			階段圖	ST	X0 — EN — 13D.(*) Sa : R0 Sb : R2 D : R4 — U/S — D=0 D<0											
階段圖	ST															
X0 — EN — 13D.(*) Sa : R0 Sb : R2 D : R4 — U/S — D=0 D<0																
<table><tr><td>被乘數 Sa</td><td><table><tr><td>R1</td><td>R0</td></tr><tr><td colspan="2">12345678</td></tr></table></td></tr><tr><td>×</td><td>乘數 Sb</td></tr><tr><td></td><td><table><tr><td>R3</td><td>R2</td></tr><tr><td colspan="2">456</td></tr></table></td></tr></table>			被乘數 Sa	<table><tr><td>R1</td><td>R0</td></tr><tr><td colspan="2">12345678</td></tr></table>	R1	R0	12345678		×	乘數 Sb		<table><tr><td>R3</td><td>R2</td></tr><tr><td colspan="2">456</td></tr></table>	R3	R2	456	
被乘數 Sa	<table><tr><td>R1</td><td>R0</td></tr><tr><td colspan="2">12345678</td></tr></table>	R1	R0	12345678												
R1	R0															
12345678																
×	乘數 Sb															
	<table><tr><td>R3</td><td>R2</td></tr><tr><td colspan="2">456</td></tr></table>	R3	R2	456												
R3	R2															
456																
<table><tr><td>積 D</td><td><table><tr><td>R7</td><td>R6</td><td>R5</td><td>R4</td></tr><tr><td colspan="4">5629629168</td></tr></table></td></tr></table>			積 D	<table><tr><td>R7</td><td>R6</td><td>R5</td><td>R4</td></tr><tr><td colspan="4">5629629168</td></tr></table>	R7	R6	R5	R4	5629629168							
積 D	<table><tr><td>R7</td><td>R6</td><td>R5</td><td>R4</td></tr><tr><td colspan="4">5629629168</td></tr></table>	R7	R6	R5	R4	5629629168										
R7	R6	R5	R4													
5629629168																

6-21 除法運算 (/)

FUN14 D P (/)		除法運算 (DIVISION) (將 Sa 除以 Sb 所得之商和餘數存到 D 去)										FUN14 D P (/)			
指令說明															
階梯圖符號 除算控制 — EN — 14DP.(/) 正/負數選擇 — U/S — Sa : Sb : D : D=0 — 商=0 (FO0) ERR — 除數為0 (FO1)								運算元 Sa：被除數或其暫存器號碼。 Sb：除數或其暫存器號碼。 D：存放結果(商和餘數)之暫存器號碼。 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用。							
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16、32 位元 正、負 數	V、Z P0-P9
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	D		○	○	○	○	○	○	○		○	○*	○*	○	
功能敘述		● 當除算控制“EN”=1 或由 0→1 (P 指令) 而“U/S”=0 時，將 Sa 與 Sb 以正負數 (Sign) 運算法則作除法運算並將結果寫入 D 去。同時若商為 0，則 FO0 (商=0) 設為 1，若除數 Sb=0 則錯誤旗號 FO1 (除數=0) 設為 1 且本指令不執行。 ● 當除算控制“EN”=1 或由 0→1 (P 指令) 而“U/S”=1 時，將 Sa 與 Sb 以正整數 (Unsign) 運算法則作除法運算並將結果寫入 D 去。同時若商為 0，則 FO0 (商=0) 設為 1，若除數 Sb=0 則錯誤旗號 FO1 (除數=0) 設為 1 且本指令不執行。													

FUN14 D P (/)	除法運算 (DIVISION) (將 Sa 除以 Sb 所得之商和餘數存到 D 去)	FUN14 D P (/)												
程式範例 1	16 位元除法例													
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN R2 := R0 / R1; END_IF</pre></td></tr></table> 被除數 Sa <table><tr><td>R0</td></tr><tr><td>256</td></tr></table> ÷ 除數 Sb <table><tr><td>R1</td></tr><tr><td>12</td></tr></table> D <table><tr><td>R3</td><td>R2</td></tr><tr><td>4</td><td>21</td></tr></table> 餘數 商			階段圖	ST		<pre>IF X0 THEN R2 := R0 / R1; END_IF</pre>	R0	256	R1	12	R3	R2	4	21
階段圖	ST													
	<pre>IF X0 THEN R2 := R0 / R1; END_IF</pre>													
R0														
256														
R1														
12														
R3	R2													
4	21													
程式範例	32 位元除法例													

階段圖	ST
 Diagram illustrating the Ladder Logic (階段圖) for the D=0 instruction. The instruction is represented by a box with inputs EN (Enabled) and U/S (Underflow/Overflow). The box contains three lines: Sa : R0, Sb : R2, and D : R4. The output is D=0. The instruction is labeled 14D.(/).	

FUN14 D P (/)	除法運算 (DIVISION) (將 Sa 除以 Sb 所得之商和餘數存到 D 去)	FUN14 D P (/)								
被除數 Sa <table><tr><td>R1</td><td>R0</td></tr><tr><td colspan="2">2147483647</td></tr></table> ÷ 除數 Sb <table><tr><td>R3</td><td>R2</td></tr><tr><td colspan="2">1234567</td></tr></table>			R1	R0	2147483647		R3	R2	1234567	
R1	R0									
2147483647										
R3	R2									
1234567										
D <table><tr><td>R7</td><td>R6</td></tr><tr><td colspan="2">571634</td></tr></table> 餘數 <table><tr><td>R5</td><td>R4</td></tr><tr><td colspan="2">1739</td></tr></table> 商			R7	R6	571634		R5	R4	1739	
R7	R6									
571634										
R5	R4									
1739										

6-22 MOD 取餘數

FUN226 D P MOD		取餘數 (取出)												FUN226 D P MOD			
指令說明																	
執行控制 EN 226DP.MOD Sa : Sb : D :										運算元 Sa：被除數或其暫存器號碼。 Sb：除數或其暫存器號碼。 D：存放結果(餘數)之暫存器號碼。 Sa、Sb、D 可結合 V、Z、P0~P9 作間接定址應用。							
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16、32 位元 正、負 數	V、Z P0-P9	
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
	D		○	○	○	○	○	○	○		○	○*	○*	○		○	
功能敘述																	
		● 當除算控制“EN” =1 或由 0→1 (P 指令) 而“U/S” =0 時，將 Sa 與 Sb 以正負數 (Sign) 運算法則作餘數運算並將結果寫入 D 去。 ● 當除算控制“EN” =1 或由 0→1 (P 指令) 而“U/S” =1 時，將 Sa 與 Sb 以正整數 (Unsign) 運算法則作餘數運算並將結果寫入 D 去。 ※當除數為 0 時，會當做商為 0，並餘數就等於被除數。使用時須注意，除數(分母)不應輸入 0，否則會得到餘數=被除數。															

FUN226 D P MOD	取餘數 (取出)	FUN226 D P MOD
程式範例		

6-23 遞增 (+ 1)

FUN15 D P (+ 1)	遞增 (加 1) (將 D 之內容值加 1)											FUN15 D P (+ 1)																																								
指令說明																																																				
階段圖符號 遞增控制 — EN — 15DP. (+1) D — OVF — 溢位 (FOO)							運算元 D：遞增之暫存器號碼。 D：可結合 V、Z、P0~P9 作間接定址應用。																																													
<table><tr><th>範圍 運算元</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>XR</th></tr><tr><td></td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>V、Z P0-P9</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>○</td><td>○</td></tr></table>														範圍 運算元	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR		WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0-P9	D	○	○	○	○	○	○	○	○	○*	○*	○	○
範圍 運算元	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR																																								
	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0-P9																																								
D	○	○	○	○	○	○	○	○	○*	○*	○	○																																								
功能敘述																																																				
● 當遞增控制“EN” =1 或由 0→1 (P 指令) 時，將 D 之內容值加 1。若 D 之值已在正數之最上限 (32767 或 2147483647)，加 1 之結果會使 D 之值變成負數之最下限 (-32768 或 -2147483648)，同時 FOO 溢位旗號 (OVF) 設為 1。 ● 溢位之詳細說明請參考 5.4 節之敘述。																																																				
程式範例	16 位元暫存器遞增例 (間接定址)																																																			

FUN15 D P (+ 1)	遞增 (加 1) (將 D 之內容值加 1)	FUN15 D P (+ 1)								
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>IF R_TRIG(S:= X0) THEN R0V++ ; END_IF</pre></td></tr></table> 當 $V = 100 \cdot 0 + 100 = 100$ D <table><tr><td>R100</td><td>1</td></tr></table> ↓ $X0 = 1$ D <table><tr><td>R100</td><td>2</td></tr></table>			階段圖	ST		<pre>IF R_TRIG(S:= X0) THEN R0V++ ; END_IF</pre>	R100	1	R100	2
階段圖	ST									
	<pre>IF R_TRIG(S:= X0) THEN R0V++ ; END_IF</pre>									
R100	1									
R100	2									

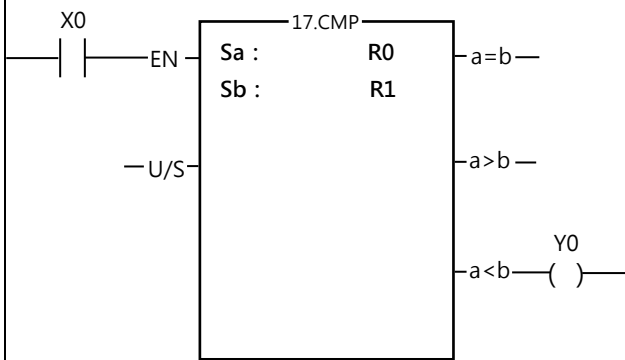
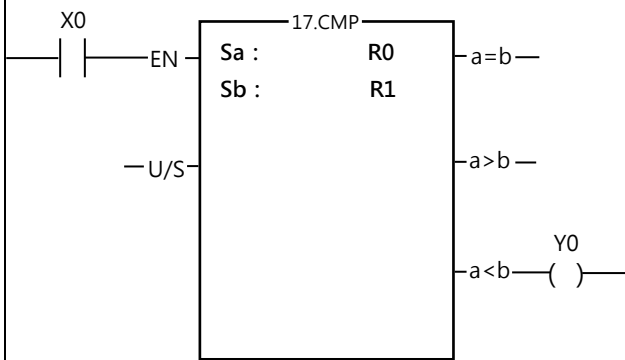
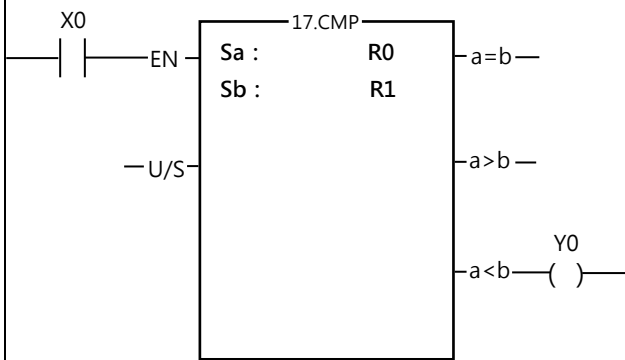
6-24 遞減 (- 1)

FUN16 DP (- 1)	遞減 (減 1) (將 D 之內容值減 1)	FUN16 DP (- 1)																																						
指令說明																																								
階梯圖符號 遞減控制 — EN — 16DP. (-1) D — UDF — 欠位 (FO0)		運算元 D：遞減之暫存器號碼。 D：可結合 V、Z、P0~P9 作間接定址應用。																																						
<table><tr><th rowspan="2">範圍 運算元</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>XR</th></tr><tr><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>V、Z P0-P9</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>○</td><td>○</td></tr></table>			範圍 運算元	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0-P9	D	○	○	○	○	○	○	○	○	○*	○*	○	○
範圍 運算元	WY	WM		WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR																											
	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0-P9																												
D	○	○	○	○	○	○	○	○	○*	○*	○	○																												
功能敘述																																								
● 當遞減控制“EN” =1 或由 0→1 (P 指令) 時，將 D 之內容值減 1。若 D 之值已在負數之最下限 (- 32768 或 -2147483648)，減 1 之結果會使 D 之值變成正數之最上限 (32767 或 2147483647)，同時欠位 (UDF) 旗號 FO0 變成 1。 ● 欠位之詳細說明請參考 5.4 節之敘述。																																								
程式範例	16 位元暫存器遞減例																																							
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td> X0 — — EN— 16. (-1) R0 —OVF— </td><td> IF X0 THEN R0-- ; END_IF </td></tr></table>			階梯圖	ST	X0 — — EN— 16. (-1) R0 —OVF—	IF X0 THEN R0-- ; END_IF																																		
階梯圖	ST																																							
X0 — — EN— 16. (-1) R0 —OVF—	IF X0 THEN R0-- ; END_IF																																							

FUN16 D P (- 1)	遞減 (減 1) (將 D 之內容值減 1)	FUN16 D P (- 1)
D R0 0 ↓ X0 = └─┐ D R0 - 1		

6-25 數值比較 (CMP)

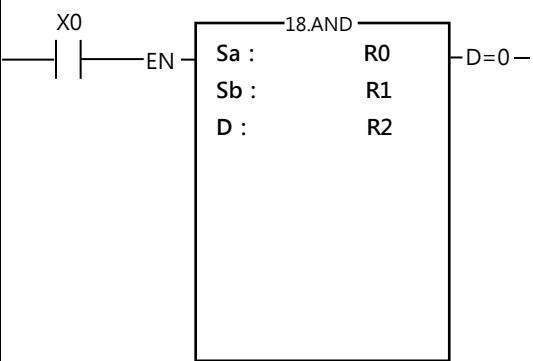
FUN17 DP CMP		數值比較 (COMPARE) (比較 Sa 與 Sb 數值之大小並產出結果)												FUN17 DP CMP	
指令說明															

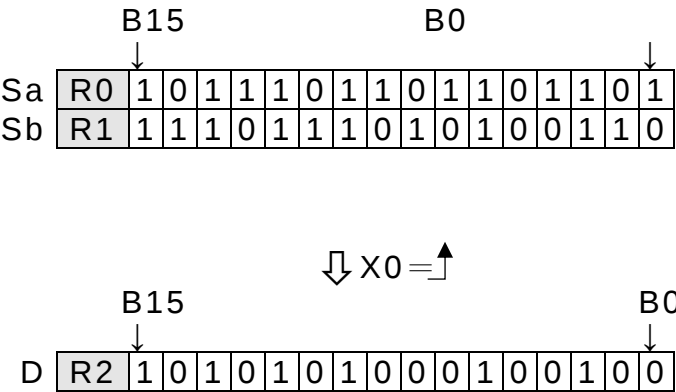
FUN17 CMP	數值比較 (COMPARE) (比較 Sa 與 Sb 數值之大小並產出結果)	FUN17 CMP				
程式範例	16 位元暫存器數值比較例					
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table>			階段圖	ST		
階段圖	ST					
						
- 上例假若 R0 之值為 1，R1 之值為 2，則當 X0=1 時，CMP 指令執行比較工作，並得出 a < b 之結果，故會將 FO0 及 FO1 設為 0，FO2 (a < b) 設為 1。 - 若您需要複合結果，如$\geq$、$\leq$、< > 等，請先將 =、>、< 等結果送到繼電器再由繼電器取出 OR 起來即可。						

6-26 邏輯及運算(AND)

FUN18 D P AND		邏輯 及 (AND) 運算												FUN18 D P AND	
指令說明															

FUN18 D P AND	邏輯及 (AND) 運算	FUN18 D P AND
-------------------------	----------------	-------------------------

階段圖	ST
	



6-27 邏輯或運算(OR)

FUN19 D P OR		邏輯或 (OR) 運算												FUN19 D P OR	
指令說明															

FUN19

D

P

OR

邏輯或 (OR) 運算

FUN19

D

P

OR

程式範例

16 位元邏輯 OR 運算例

階段圖

X0

●

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

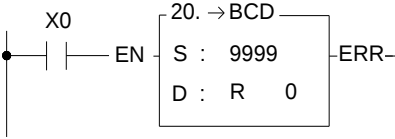
|

|

6-28 BIN→BCD 變換

FUN20 D P →BCD		BIN→BCD 變換 (將 S 之二進碼轉成 BCD 碼後存入 D)														FUN20 D P →BCD																																																																
指令說明																																																																																
階梯圖符號 轉換控制 — EN — 20DP.→ BCD — ERR — 錯誤訊息(FO0) S : D :										運算元 S：被轉換之資料或其暫存器號碼。 D：存放轉換結果(BCD 碼)之暫存器號碼。 S、D 可結合 V、Z、P0~P9 作間接定址應用。																																																																						
<table><tr><td rowspan="4">範圍 運算元</td><td>WX</td><td>WY</td><td>WM</td><td>WSM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>WX0 WX100 8</td><td>WY0 WY100 8</td><td>WM0 WM91 04</td><td>WM91 20 WM25 584</td><td>WS0 WS308 8</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R3476 7</td><td>R3476 8 R3489 5</td><td>R3502 4 R3515 1</td><td>R3528 0 R4322 3</td><td>R4322 4 R4731 9</td><td>D0 D7999</td><td>16 位 元 正、 負數</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>																				範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX100 8	WY0 WY100 8	WM0 WM91 04	WM91 20 WM25 584	WS0 WS308 8	T0 T1023	C0 C1279	R0 R3476 7	R3476 8 R3489 5	R3502 4 R3515 1	R3528 0 R4322 3	R4322 4 R4731 9	D0 D7999	16 位 元 正、 負數	V、Z P0~P9	S	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○		○	○*	○*	○		○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																	
	WX0 WX100 8	WY0 WY100 8	WM0 WM91 04	WM91 20 WM25 584	WS0 WS308 8	T0 T1023	C0 C1279	R0 R3476 7	R3476 8 R3489 5	R3502 4 R3515 1	R3528 0 R4322 3	R4322 4 R4731 9	D0 D7999	16 位 元 正、 負數	V、Z P0~P9																																																																	
	S	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																	
	D		○	○	○	○	○	○		○	○*	○*	○		○																																																																	
功能敘述																																																																																
● 因 PLC 內部之數值均以二進碼 (Binary) 來儲存和演算，因此若欲將 PLC 內部資料送到外界之顯示器去 (如 7 段顯示器)，需利用此指令將之變換成合乎人們習慣之 BCD 碼後再送到顯示器去，以方便閱讀 (例如二進碼 “1100”，若將之轉換為 BCD 碼 “12” 就容易閱讀多了)。 ● 當變換控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 之二進碼轉成等值之 BCD 碼後寫入 D 去。若 S 之數值非 BCD 之 0~9999 或 99999999，則 FO0 錯誤旗號設為 1，D 維持舊值。																																																																																

FUN20 D P →BCD	BIN→BCD 變換 (將 S 之二進碼轉成 BCD 碼後存入 D)	FUN20 D P →BCD
程式範例	16 位元 BIN→BCD 轉換例	

階段圖	ST
	<pre> IF X0 THEN ToBCD(S:= 9999, D=> R0); END_IF </pre>

B15
B0

S 常數
0
0
1
0
0
1
1
1
0
0
0
0
1
1
1
1

⇓ X0 = 1

B15
B0

D R0
1
0
0
1
1
0
0
1
1
0
0
1
1
0
0
1

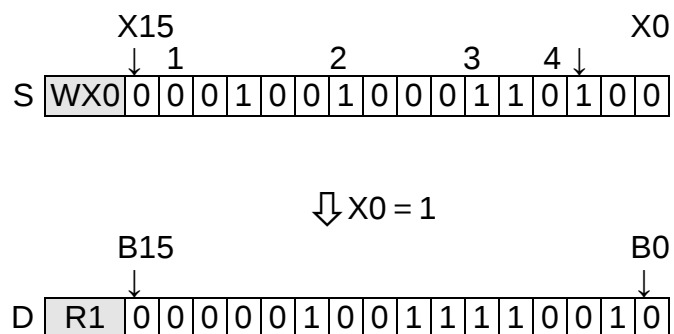
6-29 BCD→BIN 變換

FUN21 D P →BIN		BCD→BIN 變換 (將 S 之 BCD 碼轉成二進碼後存入 D)														FUN21 D P →BIN																																																																										
指令說明																																																																																										
階段圖符號 轉換控制 — EN — 21DP.→BIN S : D : — ERR — 錯誤訊息 (FO0)										運算元 S：被轉換之資料或其暫存器號碼。 D：存放轉換結果(BIN 碼)之暫存器號碼。 S，D 可結合 V、Z、P0~P9 作間接定址應用。																																																																																
<table><tr><td rowspan="5">範圍 運算元</td><td>WX</td><td>WY</td><td>WM</td><td>WSM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>XR</td></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																				範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0~P9	S	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○														
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	XR																																																																												
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0~P9																																																																												
	S	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																												
	D		○	○	○	○	○	○	○		○	○*	○*	○																																																																												
功能敘述																																																																																										
欲由外界以人們習用之 10 進制（BCD 碼）輸入裝置（如指撥開關）來輸入數值到 PLC 內部，需利用此指令將 BCD 數字轉成 BIN 碼，PLC 才能處理。 當變換控制“EN”=1 或由 0→1（P 指令）時，將 S 之 BCD 碼轉換成等值之 BIN 碼後寫入 D 去。若 S 值非 BCD 碼，則 FO0 錯誤旗號變為 1，D 則維持原值。 常數數值在輸入時就已轉換成二進碼儲存起來，故不能再以常數當作 S 轉成二進碼。																																																																																										

FUN21 D P →BIN	BCD→BIN 變換 (將 S 之 BCD 碼轉成二進碼後存入 D)	FUN21 D P →BIN
---------------------------------	---	---------------------------------

程式範例	16 位元 BCD→BIN 轉換例
------	-------------------

階段圖	ST
	<pre>IF R_TRIG(S:= X0) THEN ToBIN(S:= WX0 , D=> R1); END_IF</pre>



7

進階應用指令

7-1	數學運算指令 (FUN24 ~ 33)	7-1
7-1-1	總和計算 (SUM)	7-1
7-1-2	取平均值 (MEAN)	7-4
7-1-3	取負數 (NEGATION)	7-6
7-1-4	取絕對值 (ABSOLUTE)	7-8
7-1-5	線性轉換指令 (LCNV)	7-10
7-1-6	多段線性轉換指令 (MLC)	7-19
7-2	邏輯運算指令 (FUN35 ~ 36)	7-29
7-2-1	邏輯互斥或 (XOR) 運算 (EXCLUSIVE OR)	7-29
7-2-2	邏輯互斥容 (XNR) 運算 (ENCLUSIVE OR)	7-31
7-3	比較指令 (FUN37)	7-33
7-3-1	區域比較 (ZONE COMPARE)	7-33
7-4	搬移指令一 (FUN40 ~ 50)	7-35
7-4-1	位元資料讀取 (BIT READ)	7-35
7-4-2	位元資料寫入 (BIT WRITE)	7-38
7-4-3	位元資料搬移 (BIT MOVE)	7-40
7-4-4	位數 (NIBBLE) 搬移 (NIBBLE MOVE)	7-42
7-4-5	位元組 (BYTE) 搬移 (BYTE MOVE)	7-44
7-4-6	資料互換 (EXCHANGE)	7-46
7-4-7	位元組 (BYTE) 資料對換 (BYTE SWAP)	7-49
7-5	位移 / 旋轉指令 (FUN51 ~ 54)	7-51
7-5-1	向左位移 (SHIFT LEFT)	7-51
7-5-2	向右位移 (SHIFT RIGHT)	7-53
7-5-3	向左旋轉 (ROTATE LEFT)	7-55

7-5-4 向右旋轉 (ROTATE RIGHT)	7-57
7-6 數碼變換指令 (FUN55 ~ 64)	7-59
7-6-1 二進制碼轉換格雷碼(B→G).....	7-59
7-6-2 格雷碼轉換二進制碼(G→B).....	7-61
7-6-3 時：分：秒→秒 (HOUR：MINUTE：SECOND→SECOND)	7-63
7-6-4 秒數→時：分：秒 (SECOND→HOUR：MINUTE：SECOND)	7-66
7-6-5 ASCII 碼轉換為十六進制值 (ASCII→HEX)	7-69
7-6-6 十六進制值轉換為 ASCII 碼 (HEX→ASCII)	7-73
7-7 流程控制指令 (FUN22、FUN65 ~ 71)	7-76
7-7-1 BREAK (BREAK)	7-76
7-7-2 標記 (LABEL)	7-78
7-7-3 跳躍 (JUMP)	7-81
7-7-4 功能區塊圖標記 (FUNTION BLOCK LABEL)	7-83
7-7-5 功能區塊圖跳躍 (FUNTION BLOCK JUMP)	7-86
7-7-6 呼叫 (CALL)	7-88
7-7-7 副程式返回 (RETURN FROM SUBROUTINE)	7-90
7-7-8 中斷返回 (RETURN FROM INTERRUPT)	7-91
7-7-9 迴圈開始 (FOR)	7-93
7-7-10 迴圈結束(NEXT)	7-95
7-8 I / O 指令一 (FUN74 ~ 86)	7-98
7-8-1 即時 I / O 更新 (IMMEDIATE I/O REFRESH)	7-98
7-9 PID 控制 (FUN38、FUN99)	7-100
7-9-1 溫控便利指令(TPCTL2)	7-100
7-9-2 泛用 PID 運算指令(PID)	7-107
7-9-3 泛用 PID 控制指令(PID2).....	7-109
7-10 積算型計時器指令 (FUN87 ~ 89)	7-116
7-10-1 積算型計時器 (10 毫秒，100 毫秒，1 秒) (ACCUMULATIVE TIMER)	7-116
7-11 監控計時器指令 (FUN90 ~ 91)	7-120
7-11-1 監控定時器 (WATCHDOG TIMER) 設定時間設定	7-120
7-11-2 清除監控計時器 (RESET WATCHDOG TIMER)	7-122
7-12 高速計數器 / 計時器指令 (FUN92 ~ 93)	7-125
7-12-1 硬體高速計數器現在值 (CV) 讀取	7-125
7-12-2 硬體高速計數器 CV 或 PV 值寫入	7-128
7-13 緩升 / 緩降指令 (FUN95 ~ 98)	7-131
7-13-1 追蹤型類比輸出緩升 / 緩降指令	7-131

7-14 列表指令 (FUN100 ~ 114)	7-138
7-14-1 暫存器→列表搬移 (REGISTER TO TABLE MOVE)	7-138
7-14-2 列表→暫存器搬移 (TABLE TO REGISTER MOVE)	7-140
7-14-3 列表→列表搬移 (TABLE TO TABLE MOVE)	7-142
7-14-4 整個列表搬移 (BLOCK TABLE MOVE)	7-144
7-14-5 暫存器對列表找尋異同 (REGISTER TO TABLE SEARCH)	7-146
7-14-6 列表對列表比較異同 (TABLE TO TABLE COMPARE)	7-149
7-14-7 列表填塞 (TABLE FILL)	7-152
7-14-8 列表位移 (TABLE SHIFT)	7-154
7-14-9 列表旋轉 (TABLE ROTATE)	7-156
7-14-10 貯列 (QUEUE)	7-158
7-14-11 堆疊 (STACK)	7-161
7-14-12 區塊比較 (凸輪開關 DRUM) (BLOCK COMPARE)	7-164
7-14-13 大小排序便利指令 (SORTING)	7-167
7-14-14 區域寫入 (ZONE WRITE)	7-169
7-15 矩陣指令 (FUN120 ~ 130)	7-173
7-15-1 矩陣邏輯及 (AND) 運算 (MATRIX AND)	7-173
7-15-2 矩陣邏輯或 (OR) 運算 (MATRIX OR)	7-175
7-15-3 矩陣邏輯互斥或 (XOR) 運算 (MATRIX EXCLUSIVE OR)	7-177
7-15-4 矩陣互容或 (XNR) 運算 (MATRIX ENCLUSIVE OR)	7-179
7-15-5 矩陣倒相 (MATRIX INVERSE)	7-181
7-15-6 矩陣位元位移 (MATRIX BIT SHIFT)	7-183
7-15-7 矩陣位元旋轉 (MATRIX BIT ROTATE)	7-185
7-15-8 矩陣位元狀態數量計算 (MATRIX BIT STATUS COUNT)	7-187
7-16 NC 定位控制指令一 (FUN137 ~ 144)	7-189
7-16-1 中斷定角指令 (ICA)	7-189
7-16-2 中斷定長指令 (ICF)	7-191
7-16-3 高速脈波寬度調變 (HSPWM)	7-193
7-16-4 高速脈波輸出 (HSPSO)	7-197
7-16-5 定位程式參數設定指令 (MPARA)	7-214
7-16-6 強制停止 HSPSO 脈波輸出 (PSOFF)	7-222
7-16-7 目前脈波值轉換為顯示值 (PSCNV)	7-223
7-16-8 高速脈波寬度調變 2 (HSPWM2)	7-225
7-17 中斷控制指令 (FUN145 ~ 146)	7-226
7-17-1 允許外界輸入或週邊中斷作動指令 (EN)	7-226
7-17-2 禁止外界輸入或週邊中斷作動指令 (DIS)	7-229
7-18 NC 定位控制指令二 (FUN147 ~ FUN149)	7-232
7-18-1 FUN147 多軸直線補間定位輸出指令	7-232

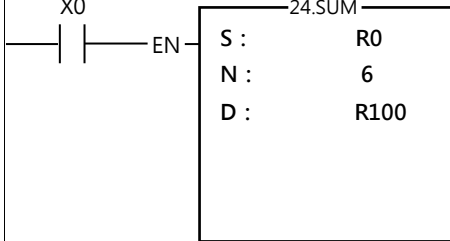
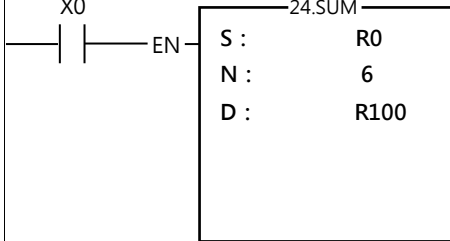
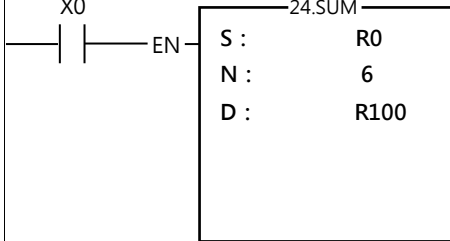
7-18-2 FUN148 手搖輪定位控制 (MPG)	7-234
7-18-3 FUN149 多軸圓弧補間定位輸出指令	7-238
7-19 通訊指令 (FUN150 ~ 156)	7-239
7-19-1 Modbus RTU 通訊協定(主站)通訊連線便利指令	7-239
7-19-2 通訊連線便利指令(CLINK)	7-241
7-19-3 Modbus 網路通訊連線便利指令(NCR)	7-243
7-19-4 串列通訊模組控制指令(CMCTL)	7-245
7-20 搬移指令二 (FUN160 ~ 162)	7-248
7-20-1 讀/寫檔案暫存器 (Read/Write File Register)	7-248
7-20-2 寫入資料至資料記憶匣 (Write SD Card)	7-251
7-20-3 由資料記憶匣讀取資料 (Read SD Card)	7-255
7-21 接點型比較指令 (FUN170 ~ 175)	7-259
7-21-1 相等比較指令 (比較 Sa 是否等於 Sb)	7-259
7-21-2 大於比較指令 (比較 Sa 是否大於 Sb)	7-261
7-21-3 小於比較指令 (比較 Sa 是否小於 Sb)	7-263
7-21-4 不相等比較指令 (比較 Sa 是否不等於 Sb)	7-266
7-21-5 大於或等於比較指令 (比較 Sa 是否大於或等於 Sb)	7-268
7-21-6 小於或等於比較指令 (比較 Sa 是否小於或等於 Sb)	7-270
7-22 運動控制指令	7-272
7-22-8 執行運動流程 (MFFlowStart)	7-272
7-22-9 運動系統緊急停止 MFSysStop)	7-274
7-22-10 原點復歸(MFHome)	7-276
7-22-11 位置控制(MFPointMov)	7-278
7-22-12 吋動控制(MFJog)	7-280
7-22-13 改變運動控制參數(MFChgTbPrm)	7-282
7-22-14 暫停運動流程(MFFlowPause)	7-284
7-22-15 恢復運動流程(MFFlowResume)	7-286
7-22-16 終止運動流程(MFFlowHalt)	7-287
7-22-17 重置運動錯誤警報(MFSysRstAlm)	7-289
7-22-18 運動流程結束(MFFlowStop)	7-290
7-22-19 Servo 初始化(MFSysInit)	7-292
7-22-20 運動控制配方表讀取(MFSysRCPR)	7-294
7-22-21 運動控制配方表寫入(MFSysRCPW)	7-300
7-22-22 凸輪讀取(MFSysCAMR)	7-307
7-22-23 凸輪寫入(MFSysCAMW)	7-310
7-22-24 手搖輪 (MFGearMPG)	7-312
7-22-25 速度控制模式(MFVelCtl)	7-315
7-22-26 扭矩控制模式(MFTorqCtl)	7-318

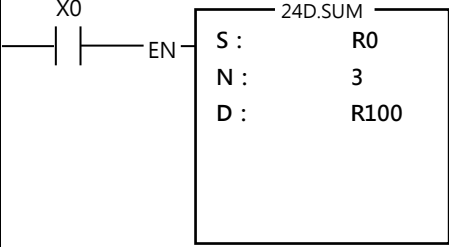
7-22-27 生成凸輪(MFSysCAMGen)	7-321
7-22-28 單軸定位 (MFAxMov)	7-324
7-22-29 寫入映射表 (MFMapTbPrm)	7-327
7-22-30 實軸轉換成虛擬軸 (MFSysSetVirt)	7-329
7-22-31 運動控制 SDO 映射讀取 MFSDOR	7-330
7-22-32 運動控制 SDO 映射寫入 MFSDOW	7-332
7-22-33 直線補間運動 MFAxLMov	7-334
7-22-34 圓弧補間運動 MFAxCirMov	7-338
7-23 其他指令 (FUN115 、 FUN258)	7-344
7-23-1 DBUF 數據緩存	7-344
7-23-2 皮重扣除指令	7-347
7-23-3 皮重偏移指令	7-348
7-24 浮點運算指令 (FUN200 ~ 220)	7-350
7-24-1 整數轉換浮點數(CONVERSION OF INTEGER TO FLOATING POINT NUMBER)	7-350
7-24-2 浮點數轉換整數(CONVERSION OF FLOATING POINT NUMBER TO INTEGER)	7-352
7-24-3 浮點數加法運算(FLOATING POINT NUMBER ADDITION)	7-354
7-24-4 浮點數減法運算(FLOATING POINT NUMBER SUBTRACTION).....	7-356
7-24-5 浮點數乘法運算(FLOATING POINT NUMBER MULTIPLICATION)	7-358
7-24-6 浮點數除法運算(FLOATING POINT NUMBER DIVIDION)	7-360
7-24-7 浮點數比較運算(FLOATING POINT NUMBER COMPARE)	7-362
7-24-8 浮點數區域比較運算(FLOATING POINT NUMBER ZONE COMPARE)	7-364
7-24-9 浮點數開根號運算(FLOATING POINT NUMBER SQUARE ROOT)	7-366
7-24-10 浮點數表示法取三角函數(SIN)運算	7-368
7-24-11 浮點數表示法取三角函數(COS)運算	7-370
7-24-12 浮點數表示法取三角函數(TAN)運算	7-372
7-24-13 浮點數取負值運算 (CHANGE SIGN OF THE FLOATING POINT NUMBER)	7-374
7-24-14 浮點數取絕對值運算(FLOATING POINT NUMBER ABSOLUTE VALUE)	7-376
7-24-15 浮點數反正弦函數值運算(FLOATING POINT ARC SINE FUNTION).....	7-378
7-24-16 浮點數反餘弦函數值運算(FLOATING POINT ARC COSINE FUNTION).....	7-381

7-1 數學運算指令 (FUN24 ~ 33)

7-1-1 總和計算 (SUM)

FUN24 D P SUM		總和計算 (SUM)														FUN24 D P SUM																																																																																			
指令說明																																																																																																			
		階梯圖符號 24DP.SUM 運算控制 — EN — S : N : D :										S：來源暫存器之起頭號碼 N：欲總和之暫存器個數 (由 S 開始連續 N 個) D：存放結果 (總和) 之暫存器號碼 S · N · D 可結合 V、Z、P0~P9 作間接定址應用																																																																																							
		<table><tr><th>範圍 運算元</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WSM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>1 511</td><td>V、Z P0-P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>N</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>																		範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	1 511	V、Z P0-P9	S	○	○	○	○	○	○	○	○	○	○	○	○	○		○	N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○		○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																				
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	1 511	V、Z P0-P9																																																																																				
S	○	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																				
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																				
D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																				
功能敘述																																																																																																			
		● 當運算控制“EN”=1 或由 0→1 (P 指令) 時，將 S 開始之連續 N 個 16 位元或 32 位元 (P 指令) 暫存器作加法運算，得出總和，並將結果存入 D 所指定之暫存器。 ● 當 N 之值為 0 或大於 511 時，運算不執行。 ● 如欲通訊對象之資料錯誤查核方式為總和 (Checksum) 查核，則可使用此指令來產生總和值或利用此指令計算總和值並比對看是否資料有誤。																																																																																																	

FUN24 D P SUM	總和計算 (SUM)	FUN24 D P SUM				
程式範例 1	X0 由 OFF→ON 時，計算 16 位元總和					
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td>IF X0 THEN SUM(S:= R0, N:= 6, D:= R100); END_IF</td></tr></table> R0=0030H R1=0031H R2=0032H R3=0033H R4=0034H R5=0035H } → R100=012FH 上圖範例係將 R0 開始之 6 個暫存器以 16 位元方式計算總和值，並將結果存入 R100 暫存器。			階段圖	ST		IF X0 THEN SUM(S:= R0, N:= 6, D:= R100); END_IF
階段圖	ST					
	IF X0 THEN SUM(S:= R0, N:= 6, D:= R100); END_IF					
程式範例 2	X0 ON 時，計算 32 位元總和					

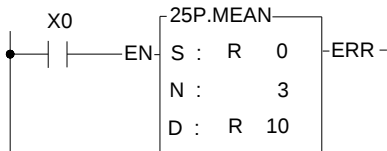
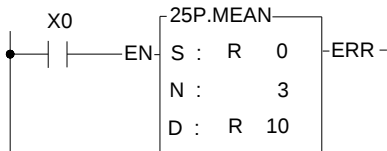
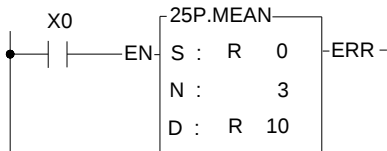
階段圖	ST
	<pre>IF X0 THEN SUM_D(S:= R0, N:= 6, D:= R100); END_IF</pre>

R1 ~ R0=00310030H
R3~R2=00330032H ➔ R101 ~ R100=00A5009BH
R5 ~ R4=00410039H

上圖範例係將 DR0 開始，以 32 位元方式計算總和值，並將結果存入 DR1，00 (32 位元) 暫存器內。

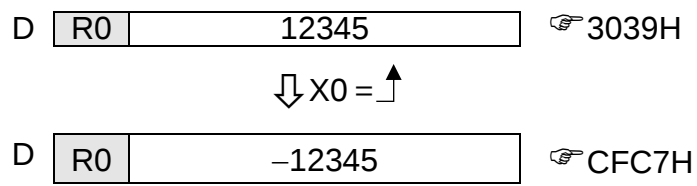
7-1-2 取平均值 (MEAN)

FUN25 DP MEAN		取平均值 (MEAN)										FUN25 DP MEAN			
指令說明															

FUN25 DP MEAN	取平均值 (MEAN)	FUN25 DP MEAN									
程式範例											
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>IF R_TRIG(S:= X0) THEN MEAN(S:= R0, N:= 3, D:= R10); END_IF</pre></td></tr></table>			階段圖	ST		<pre>IF R_TRIG(S:= X0) THEN MEAN(S:= R0, N:= 3, D:= R10); END_IF</pre>					
階段圖	ST										
	<pre>IF R_TRIG(S:= X0) THEN MEAN(S:= R0, N:= 3, D:= R10); END_IF</pre>										
上圖範例為求取自 R0 開始連續 3 個 16 位元暫存器之平均值，再將結果存於 16 位元暫存器 R10 中。 S (N = 3) <table><tr><td>R0</td><td>123</td></tr><tr><td>R1</td><td>9</td></tr><tr><td>R2</td><td>788</td></tr></table> ↓ X0 = <table><tr><td>D</td><td>R10</td><td>306</td></tr></table> $\frac{123 + 9 + 788}{3} = 306 \text{ (餘數 2 捨棄)}$			R0	123	R1	9	R2	788	D	R10	306
R0	123										
R1	9										
R2	788										
D	R10	306									

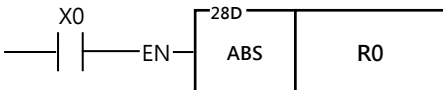
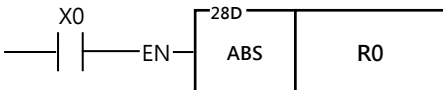
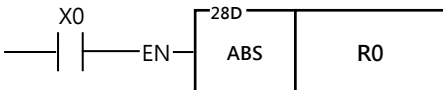
7-1-3 取負數 (NEGATION)

FUN27 DP NEG		取負數 (NEGATION)										FUN27 DP NEG			
指令說明															



7-1-4 取絕對值 (ABSOLUTE)

FUN28 DP ABS		取絕對值 (ABSOLUTE)												FUN28 DP ABS			
指令說明																	

FUN28 DP ABS	取絕對值 (ABSOLUTE)	FUN28 DP ABS																		
程式範例																				
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN ABS_D(D:= R0); END_IF</pre></td></tr></table>			階段圖	ST		<pre>IF X0 THEN ABS_D(D:= R0); END_IF</pre>														
階段圖	ST																			
	<pre>IF X0 THEN ABS_D(D:= R0); END_IF</pre>																			
上圖程式例係將暫存器 DR0 之值取其絕對值後再存回 DR0 (R1,R0) 去。																				
<table><tr><td>D</td><td><table><tr><td>R1</td><td>R0</td></tr></table></td><td><table><tr><td>-12345</td></tr></table></td><td>⇨ CFC7H</td></tr><tr><td colspan="4">↓ X0 = ↑</td></tr><tr><td>D</td><td><table><tr><td>R1</td><td>R0</td></tr></table></td><td><table><tr><td>12345</td></tr></table></td><td>⇨ 3039H</td></tr></table>			D	<table><tr><td>R1</td><td>R0</td></tr></table>	R1	R0	<table><tr><td>-12345</td></tr></table>	-12345	⇨ CFC7H	↓ X0 = ↑				D	<table><tr><td>R1</td><td>R0</td></tr></table>	R1	R0	<table><tr><td>12345</td></tr></table>	12345	⇨ 3039H
D	<table><tr><td>R1</td><td>R0</td></tr></table>	R1	R0	<table><tr><td>-12345</td></tr></table>	-12345	⇨ CFC7H														
R1	R0																			
-12345																				
↓ X0 = ↑																				
D	<table><tr><td>R1</td><td>R0</td></tr></table>	R1	R0	<table><tr><td>12345</td></tr></table>	12345	⇨ 3039H														
R1	R0																			
12345																				

7-1-5 線性轉換指令 (LCNV)

FUN33 P LCNV	線性轉換指令 (LCNV)	FUN33 P LCNV																																																
指令說明																																																		
階段圖符號 執行控制 — EN — 33P.LCNV Md : S : Ts : D : L :		Md：運算模式選擇・0~3 S：欲轉換之來源暫存器起始號碼 Ts：轉換表格起始暫存器起始號碼 D：存放轉換結果之起始暫存器號碼 L：欲轉換之長度・1~64																																																
<table><tr><th>範圍</th><th>HR</th><th>IR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td rowspan="2">運算元</td><td>R0 R34767</td><td>R34768 R34895</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V・Z P0-P9</td></tr><tr><td>Md</td><td></td><td></td><td></td><td>0~3</td><td></td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>Ts</td><td>○</td><td></td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>D</td><td>○</td><td></td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>L</td><td>○</td><td></td><td>○</td><td>○</td><td>1~64</td><td></td></tr></table>			範圍	HR	IR	ROR	DR	K	XR	運算元	R0 R34767	R34768 R34895	R43224 R47319	D0 D7999		V・Z P0-P9	Md				0~3		S	○	○	○	○		○	Ts	○		○	○		○	D	○		○*	○		○	L	○		○	○	1~64	
範圍	HR	IR	ROR	DR	K	XR																																												
運算元	R0 R34767	R34768 R34895	R43224 R47319	D0 D7999		V・Z P0-P9																																												
	Md				0~3																																													
S	○	○	○	○		○																																												
Ts	○		○	○		○																																												
D	○		○*	○		○																																												
L	○		○	○	1~64																																													
功能敘述	● 當使用類比輸入模組讀取外界類比信號時，可以利用本指令將原始類比讀值轉換為相對應之工程讀值以作為實際工程值之顯示或作為控制之比較、運算等應用。 ● 當使用溫度或類比模組來作溫度或類比量測應用時，如果 PLC 所量測之溫度或工程讀值與標準溫度計或相關標準儀表所量測之結果有偏差時，可以利用本指令來作線性修正以作為實際量測值之校正。 ● 當執行控制`EN`=1 或由 0→1(P 指令)時，將以 S 為起始之 L 個資料暫存器根據運算模式選擇，以轉換表格內容所設定之參數值執行線性轉換運算，並將運算結果存放到以 D 為起始之暫存器群中。 ● 本指令共提供二種線性轉換公式如下所示以適合各種不同之應用:																																																	

(公式一) 兩點校正法:

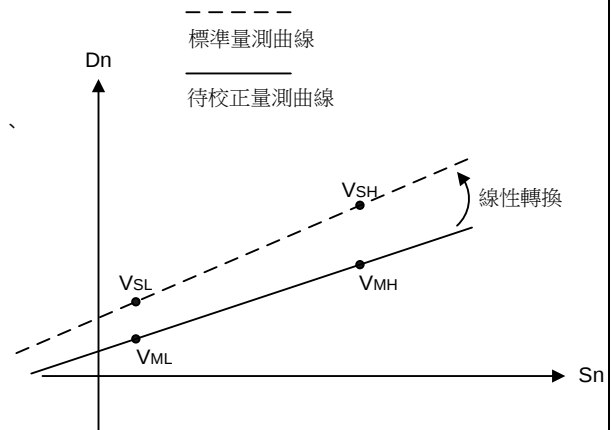
在轉換表格內填入低點測量值(VML)、高點測量值(VMH)及對應之低點標準值(VSL)與高點標準值(VSH)；執行線性轉換時，來源資料(Sn)經由下列運算產生對應之目標值(Dn)。

$$A = (VSL - VSH / VML - VMH) \times 10000$$

$$B = VSL - (VML \times A / 10000)$$

$$Dn = (Sn \times A / 10000) + B$$

- 本運算式所有運算元(VSL、VSH、VML、VMH、Sn、Dn)之值範圍為-32768 ~ 32767。
- 本運算式亦可用來將原始類比輸入讀值量化為實際工程值；在此應用時，VML=類比輸入最小值、VMH=類比輸入最大值，VSL=工程最小值，VSH=工程最大值。

**(公式二) 倍率 + 偏置量:**

在轉換表格內填入倍率分子值(A)、倍率分母值(B)及偏置值(C)；執行線性轉換時，來源資料(Sn)經由下列運算產生對應之目標值(Dn)。

$$Dn = [(Sn \times A) / B] + C$$

本運算式各運算元之值範圍如下:

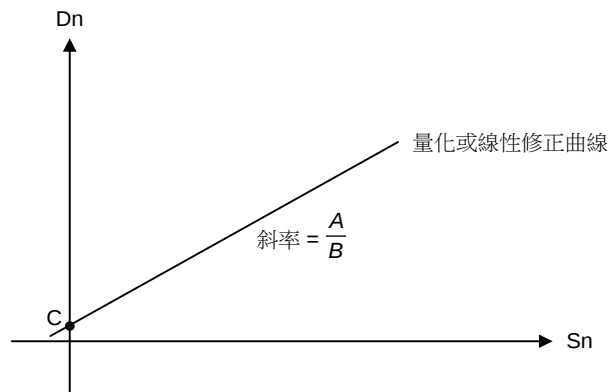
$$A = 1 \sim 65535$$

$$B = 1 \sim 65535$$

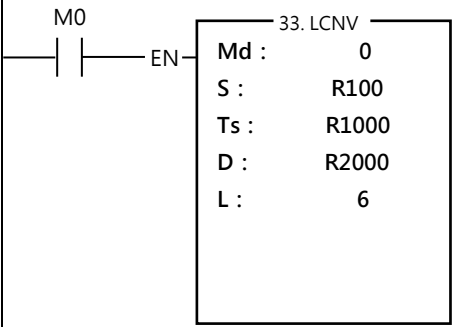
$$C = -32768 \sim 32767$$

$$Sn = 0 \sim 65535$$

$$Dn = -32768 \sim 32767$$



FUN33 P LCNV	線性轉換指令 (LCNV)	FUN33 P LCNV
	運算模式說明: 1. 運算模式為 0 時，使用公式一運算式；所有來源資料共用轉換表格內同一組 VML、VMH、VSL 與 VSH 參數值作線性轉換運算。 2. 運算模式為 1 時，使用公式一運算式；每個來源資料獨立使用轉換表格內相對應之一組 VML、VMH、VSL 與 VSH 參數值作線性轉換運算。如果有 N 個來源資料需轉換，則轉換表格內需有 N 組 VML、VMH、VSL 與 VSH 參數值，共佔用 $N \times 4$ 個暫存器。 3. 運算模式為 2 時，使用公式二運算式；所有來源資料共用轉換表格內同一組 A、B 與 C 參數值作線性轉換運算。 4. 運算模式為 3 時，使用公式二運算式；每個來源資料獨立使用轉換表格內相對應之一組 A、B 與 C 參數值作線性轉換運算。如果有 N 個來源資料需轉換，則轉換表格內需有 N 組 A、B 與 C 參數值，共佔用 $N \times 3$ 個暫存器。	
程式範例 1	運算模式為 0 之線性轉換運算	

階段圖	ST
	<pre>LCNV(EN:= M0, Md:= 0, S:= R100, Ts:= R1000, D=> R2000, L:= 6);</pre>

- 當 M0=1 時，以暫存器 R100 為起始來源資料，根據 R1000 為起始之轉換表格內 VML、VMH、VSL、VSH 參數值作 6 點線性轉換運算，並將轉換結果存放至暫存器 R2000~R2005。

FUN33 LCNV	線性轉換指令 (LCNV)	FUN33 LCNV
Ts R1000 282 VML R1001 3530 VMH R1002 260 VSL R1003 3650 VSH S R100 282 R101 3530 R102 1906 R103 0 R104 5000 R105 -115 D R2000 260 R2001 3650 R2002 1955 R2003 -34 R2004 5184 R2005 -154 程式範例 2 運算模式為 1 之線性轉換運算 階段圖 M0 — — EN 33. LCNV Md : 1 S : R100 Ts : R1000 D : R2000 L : 3 ST LCNV(EN:= M0, Md:= 1, S:= R100, Ts:= R1000, D=> R2000, L:= 3); 範例說明： ● 當 M0=1 時，以暫存器 R100 為起始來源資料，根據 R1000 為起始之轉換表格內各組 VML、VMH、VSL、VSH 參數值作 3 點線性轉換運算，並將轉換結果存放至暫存器 R2000 ~ R2002。		

FUN33P LCNV	線性轉換指令 (LCNV)		FUN33P LCNV
	Ts		
R1000	282	VML_0	
R1001	3530	VMH_0	
R1002	260	VSL_0	
R1003	3650	VSH_0	
R1004	-52	VML_1	
R1005	1208	VMH_1	
R1006	-38	VSL_1	
R1007	1101	VSH_1	
R1008	235	VML_2	
R1009	4563	VMH_2	
R1010	264	VSL_2	
R1011	4588	VSH_2	
	S		D
R100	282	⇒	R2000 260
R101	1208		R2001 1100
R102	2399		R2002 2426

FUN33 P LCNV	線性轉換指令 (LCNV)	FUN33 P LCNV				
程式範例 3	運算模式為 2 之線性轉換運算					
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td> M0 — — EN 33. LCNV Md : 2 S : R100 Ts : R1000 D : R2000 L : 6 </td><td> LCNV(EN:= M0, Md:= 2, S:= R100, Ts:= R1000, D=> R2000, L:= 6); </td></tr></table>			階段圖	ST	M0 — — EN 33. LCNV Md : 2 S : R100 Ts : R1000 D : R2000 L : 6	LCNV(EN:= M0, Md:= 2, S:= R100, Ts:= R1000, D=> R2000, L:= 6);
階段圖	ST					
M0 — — EN 33. LCNV Md : 2 S : R100 Ts : R1000 D : R2000 L : 6	LCNV(EN:= M0, Md:= 2, S:= R100, Ts:= R1000, D=> R2000, L:= 6);					
範例說明： 當 M0=1 時，以暫存器 R100 為起始來源資料，根據 R1000 為起始之轉換表格內 A、B、C 參數值作 6 點線性轉換運算，並將轉換結果存放至暫存器 R2000~R2005。						

FUN33 LCNV	線性轉換指令 (LCNV)	FUN33 LCNV
Ts R1000 985 A R1001 1000 B R1002 20 C S R100 1000 R101 2345 R102 3560 R103 401 R104 568 R105 2680 ⇒ D R2000 1005 R2001 2329 R2002 3526 R2003 414 R2004 579 R2005 2659		
程式範例 4	運算模式為 3 之線性轉換運算	
階段圖 M0 — — EN 33. LCNV Md : 3 S : R100 Ts : R1000 D : R2000 L : 4 ST LCNV(EN:= M0, Md:= 3, S:= R100, Ts:= R1000, D=> R2000, L:= 4);		
範例說明： ● 當 M0=1 時，以暫存器 R100 為起始來源資料，根據 R1000 為起始之轉換表格內各組 A、B、C 參數值作 4 點線性轉換運算，並將轉換結果存放至暫存器 R2000 ~ R2003。		

FUN33 LCNV	線性轉換指令 (LCNV)		FUN33 LCNV

7-1-6 多段線性轉換指令 (MLC)

FUN34P MLC	多段線性轉換指令 (Multiple Linear Conversion)	FUN34P MLC																																																																
指令說明																																																																		
階段圖符號 34P.MLC 執行控制 — EN — 轉換選擇 — X/Y — RS: SI: Tx: Ty: TI: D: OVR — Rs : 來源資料起始暫存器號碼 SI : 欲轉換之來源資料長度 · 1 ~ 64 Tx : X 轉換表格起始暫存器起始號碼 Ty : Y 轉換表格起始暫存器起始號碼 TI : 轉換表格長度 · 2 ~ 255 D : 存放轉換結果之起始暫存器號碼																																																																		
<table><tr><th> 運算元 範圍 </th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><td></td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td></tr><tr><td>Rs</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>SI</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>1~64</td></tr><tr><td>Tx</td><td>○</td><td></td><td></td><td></td><td>○</td><td>○</td><td></td></tr><tr><td>Ty</td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td></td></tr><tr><td>TI</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>2 ~ 255</td></tr><tr><td>D</td><td>○</td><td></td><td></td><td></td><td>○</td><td>○</td><td></td></tr></table>			運算元 範圍	HR	IR	OR	SR	ROR	DR	K		R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		Rs	○	○	○	○	○	○		SI	○	○	○	○	○	○	1~64	Tx	○				○	○		Ty	○				○*	○		TI	○	○	○	○	○	○	2 ~ 255	D	○				○	○	
運算元 範圍	HR	IR	OR	SR	ROR	DR	K																																																											
	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999																																																												
Rs	○	○	○	○	○	○																																																												
SI	○	○	○	○	○	○	1~64																																																											
Tx	○				○	○																																																												
Ty	○				○*	○																																																												
TI	○	○	○	○	○	○	2 ~ 255																																																											
D	○				○	○																																																												

FUN34 P MLC	多段線性轉換指令 (Multiple Linear Conversion)	FUN34 P MLC
功能敘述	● 當使用類比輸入模組讀取外界類比信號時，可以利用本指令將原始類比讀值作多段線性修正並轉換為相對應之工程讀值以作為實際工程值之顯示或作為控制之比較、運算等應用。 ● 當使用溫度或類比模組來作溫度或類比量測應用時，如果 PLC 所量測之溫度或工程讀值與標準溫度計或相關標準儀表所量測之結果有偏差時，可以利用本指令來作多段線性修正以作為實際量測值之校正。 ● 當執行控制“EN”=1 或由 0→1(P 指令)時，將以 Rs 為起始之 SI 個資料暫存器根據轉換選擇 X/Y，以轉換表格內容所設定之參數值執行線性轉換運算，並將運算結果存放到以 D 為起始之暫存器群中。 ● 轉換選擇 X/Y=0 時，來源資料根據 Tx 轉換表格(內容值必須由小而大排列)作搜尋找到正確線段、以 Ty 轉換表格(內容值可由小而大、或由大而小排列)相對應線段作線性轉換運算；轉換選擇 X/Y=1 時，來源資料根據 Ty 轉換表格作搜尋找到正確線段、以 Tx 轉換表格相對應線段作線性轉換運算。 ● 來源資料超出轉換表格之最小或最大值時，OVR 輸出為 1。 ● 來源資料長度或轉換表格長度錯誤時，本指令不執行。 線性轉換公式: 轉換選擇 X/Y=0 時，在 Tx 轉換表格內(內容值必須由小而大排列)填入各點測量值、及對應之 Ty 轉換表格內(內容值可由小而大、或由大而小排列)填入各點標準值；執行線性轉換時，來源資料根據 Tx 轉換表格作搜尋找到正確線段、以 Ty 轉換表格相對應線段作線性轉換運算，產生對應之目標值。轉換選擇 X/Y=1 時，作反方向運算。 $Vy = (Vx - Tx_n) \times (Ty_n+1 - Ty_n / Tx_n+1 - Tx_n) + Ty_n \quad (X/Y=0)$ $Vx = (Vy - Ty_n) \times (Tx_n+1 - Tx_n / Ty_n+1 - Ty_n) + Tx_n \quad (X/Y=1)$ 運算元 Vy、Vx、Tx_n、Tx_n+1、Ty_n、Ty_n+1 之值範圍為-32768 ~ 32767。	

FUN34 P MLC	多段線性轉換指令 (Multiple Linear Conversion)	FUN34 P MLC
多段線性轉換示意圖：		
程式範例 1		
階段圖	ST <pre> MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100); </pre>	

FUN34 
MLC多段線性轉換指令
(Multiple Linear Conversion)FUN34 
MLC

範例說明：

當 M10=1、M11=0 時，以暫存器 R0 為起始來源資料、R99 為來源資料長度，根據 R1000 為起始之 Tx 轉換表格與 R2000 為起始之 Ty 轉換表格、R199 為轉換表格長度，將 R0~R5 等來源資料作 4 段線性轉換運算，並將轉換結果存放至暫存器 D0~D5。

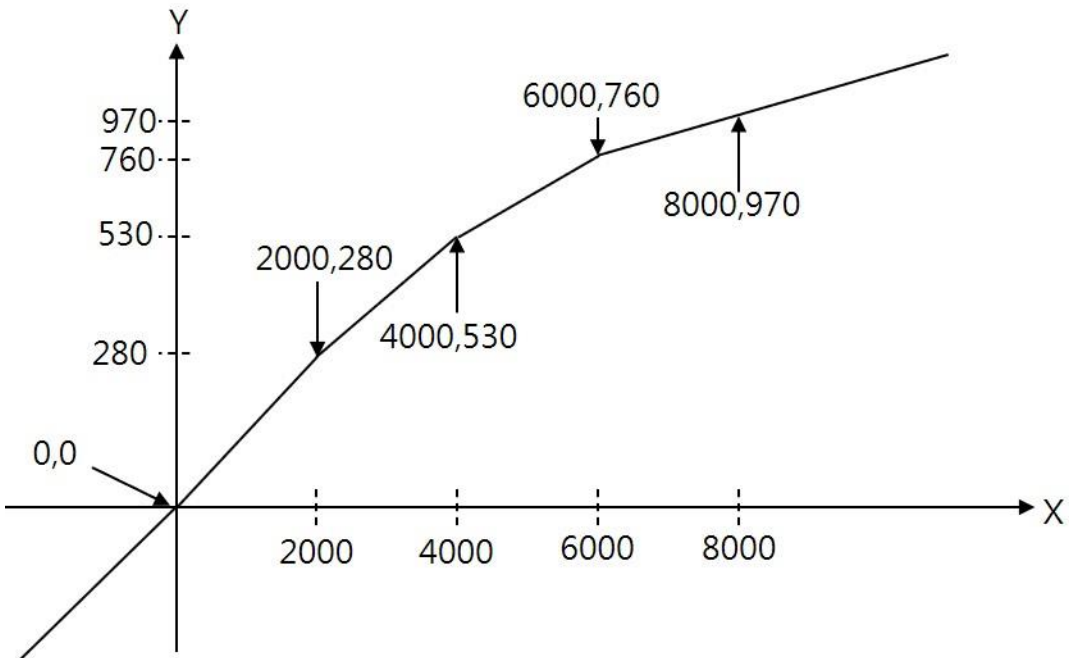
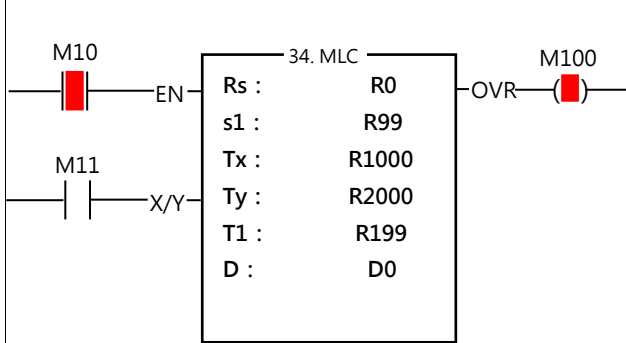
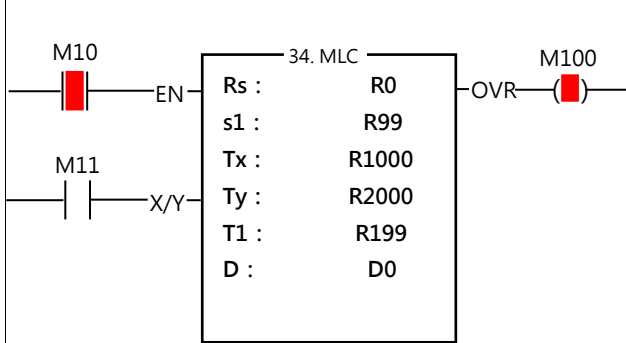
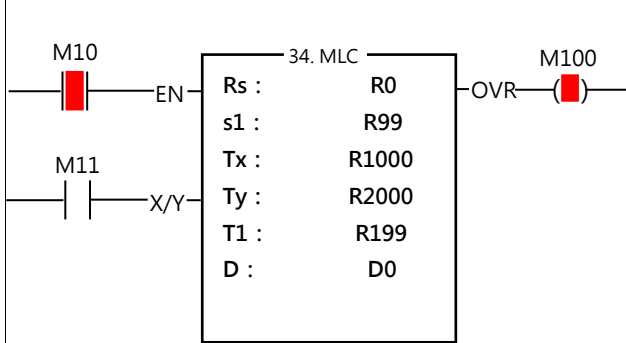
監視頁

 編號
  註解
  全部
  二進制
  十進制
  十六進制
  十進制 (正數)
  重新整理
  刪除
  全部清除

編號	狀態	資料	編號	狀態	資料	編號	狀態	資料	編號	狀態	資料
R1000	十進制	0	R2000	十進制	0	R0	十進制	1000			
R1001	十進制	2000	R2001	十進制	280	R1	十進制	2500			
R1002	十進制	4000	R2002	十進制	530	R2	十進制	5600			
R1003	十進制	6000	R2003	十進制	760	R3	十進制	7500			
R1004	十進制	8000	R2004	十進制	970	R4	十進制	8000			
R199	十進制	5				R5	十進制	10000			
M10	致能	ON	M11	致能	OFF	R99	十進制	6			
D0	十進制	140									
D1	十進制	342									
D2	十進制	714									
D3	十進制	917									
D4	十進制	970									
D5	十進制	1180									

StatusPage0

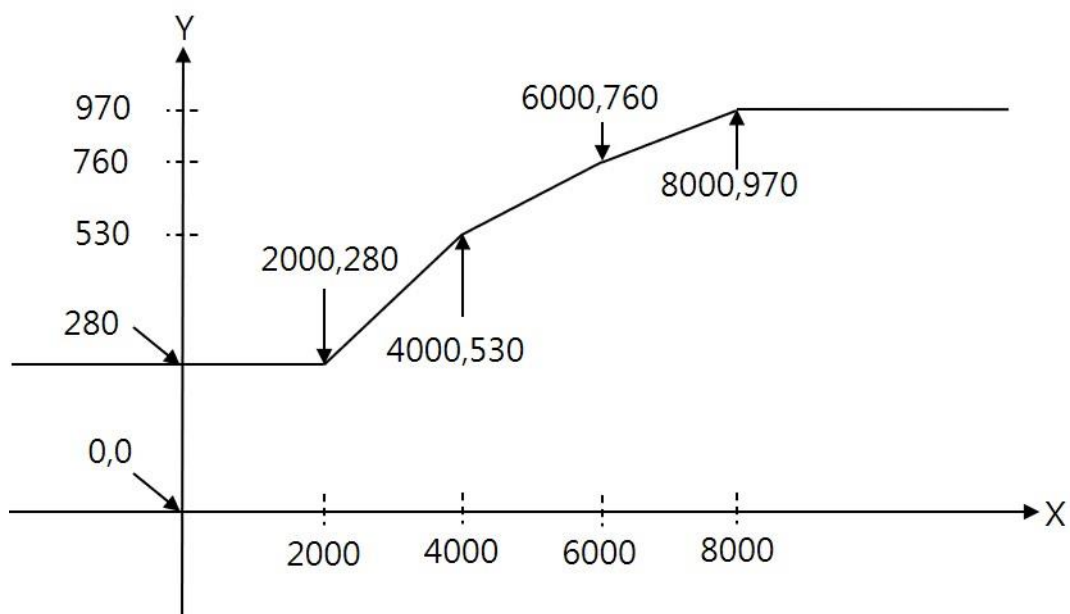
N008

FUN34 P MLC	多段線性轉換指令 (Multiple Linear Conversion)	FUN34 P MLC				
						
程式範例 2						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td>  </td><td><pre>MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100);</pre></td></tr></table>			階梯圖	ST		<pre>MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100);</pre>
階梯圖	ST					
	<pre>MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100);</pre>					
範例說明： 當 M10=1、M11=0 時，以暫存器 R0 為起始來源資料、R99 為來源資料長度，根據 R1000 為起始之 Tx 轉換表格與 R2000 為起始之 Ty 轉換表格、R199 為轉換表格長度，將 R0 ~ R5 等來源資料作 5 段線性轉換運						

FUN34 P
MLC多段線性轉換指令
(Multiple Linear Conversion)FUN34 P
MLC

監視頁

編號	狀態	資料	編號	狀態	資料	編號	狀態	資料
R1000	十進制	2000	R2000	十進制	280	R0	十進制	1000
R1001	十進制	2000	R2001	十進制	280	R1	十進制	2000
R1002	十進制	4000	R2002	十進制	530	R2	十進制	3800
R1003	十進制	6000	R2003	十進制	760	R3	十進制	7500
R1004	十進制	8000	R2004	十進制	970	R4	十進制	8000
R1005	十進制	8000	R2005	十進制	970	R5	十進制	10000
D0	十進制	280	M10	致能	ON	R99	十進制	6
D1	十進制	280	M11	致能	OFF	R199	十進制	6
D2	十進制	505						
D3	十進制	917						
D4	十進制	970						
D5	十進制	970						



FUN34 P MLC	多段線性轉換指令 (Multiple Linear Conversion)	FUN34 P MLC
----------------	--	----------------

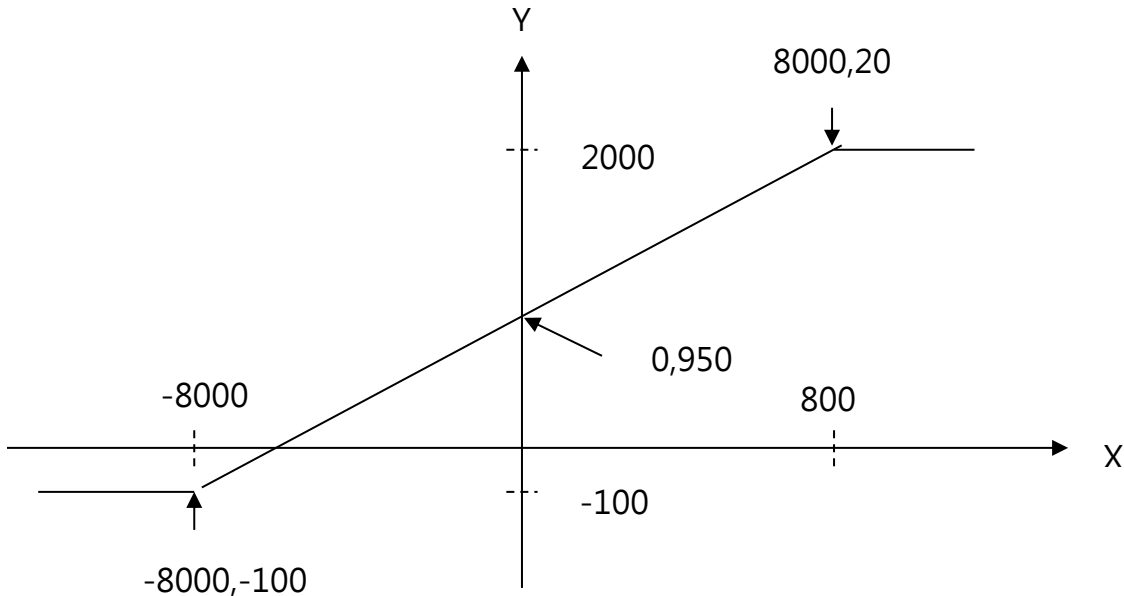
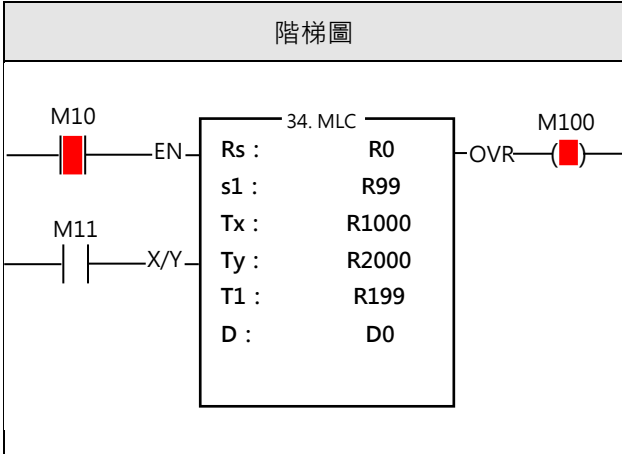
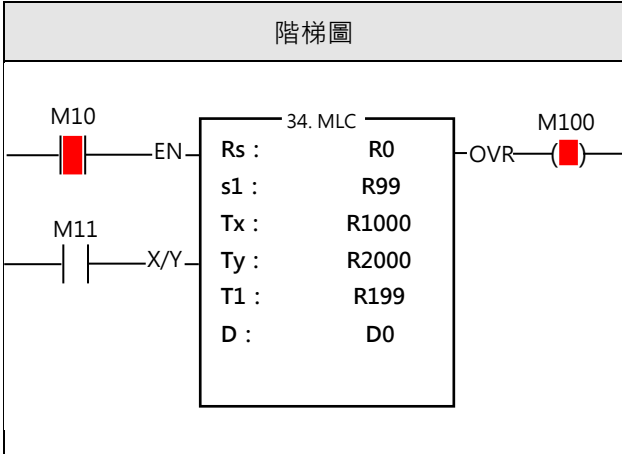
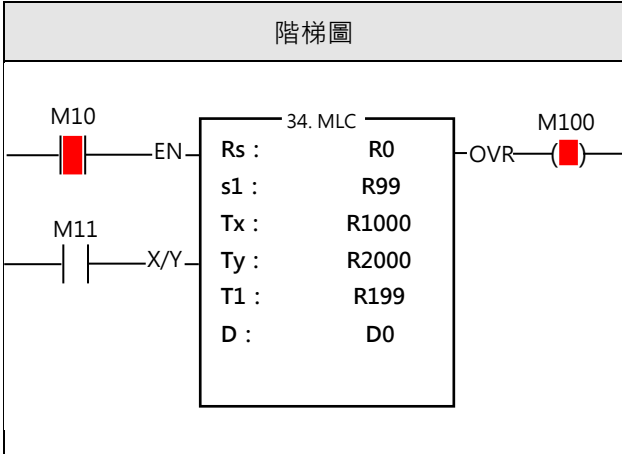
程式範例 3

階梯圖	ST
	<pre> MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100); </pre>

監視頁									
編號	註解								
編號	狀態	資料	編號	狀態	資料	編號	狀態	資料	編號
R1000	十進制	-8000	R2000	十進制	-100	R0	十進制	-8100	
R1001	十進制	-8000	R2001	十進制	-100	R1	十進制	0	
R1002	十進制	8000	R2002	十進制	2000	R2	十進制	4000	
R1003	十進制	8000	R2003	十進制	2000	R3	十進制	8100	
						R4	十進制	-10000	
						R5	十進制	10000	
D0	十進制	-100	M10	致能	ON	R99	十進制	6	
D1	十進制	950	M11	致能	OFF	R199	十進制	4	
D2	十進制	1474							
D3	十進制	2000							
D4	十進制	-100							
D5	十進制	2000							

範例說明：

當 M10=1、M11=0 時，以暫存器 R0 為起始來源資料、R99 為來源資料長度，根據 R1000 為起始之 Tx 轉換表格與 R2000 為起始之 Ty 轉換表格、R199 為轉換表格長度，將 R0~R5 等來源資料作如下圖示線性轉換運算，並將轉換結果存放至暫存器 D0~D5。此範例為來源資料之值為-8000~8000 時，根據下圖線性轉換得出對應值-100~2000；來源資料之值 ≥ 8000 時，對應值皆為 2000；來源資料之值 ≤ -8000 時，對應值皆為-100。

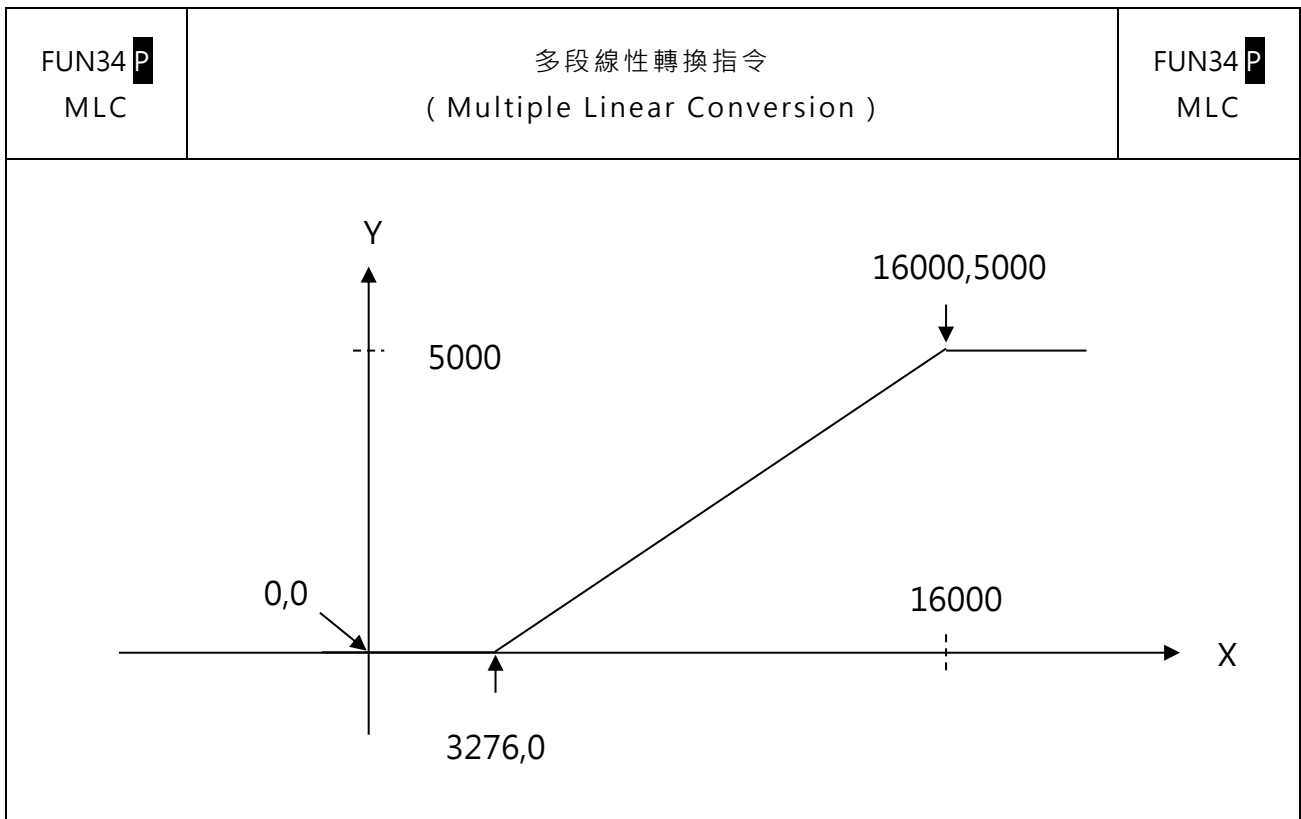
FUN34 P MLC	多段線性轉換指令 (Multiple Linear Conversion)	FUN34 P MLC				
						
程式範例 4						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td>  </td><td><pre>MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100);</pre></td></tr></table>			階段圖	ST		<pre>MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100);</pre>
階段圖	ST					
	<pre>MLC (EN:= M10, XY:= M11, Rs:= R0, S1:= R99, Tx:= R1000, Ty:= R2000, T1:= R199, D:= D0, OVR=> M100);</pre>					

FUN34 P
MLC多段線性轉換指令
(Multiple Linear Conversion)FUN34 P
MLC

監視頁									
編號		註解							
編號	狀態	資料	編號	狀態	資料	編號	狀態	資料	編
R1000	十進制	3276	R2000	十進制	0	R0	十進制	0	
R1001	十進制	3276	R2001	十進制	0	R1	十進制	3276	
R1002	十進制	16000	R2002	十進制	5000	R2	十進制	4095	
R1003	十進制	16000	R2003	十進制	5000	R3	十進制	9638	
						R4	十進制	16000	
						R5	十進制	16380	
D0	十進制	0	M10	致能	ON	R99	十進制	6	
D1	十進制	0	M11	致能	OFF	R199	十進制	4	
D2	十進制	321							
D3	十進制	2500							
D4	十進制	5000							
D5	十進制	5000							

範例說明：

當 M10=1、M11=0 時，以暫存器 R0 為起始來源資料、R99 為來源資料長度，根據 R1000 為起始之 Tx 轉換表格與 R2000 為起始之 Ty 轉換表格、R199 為轉換表格長度，將 R0 ~ R5 等來源資料作如下圖示線性轉換運算，並將轉換結果存放至暫存器 D0 ~ D5。此範例為來源資料之值為 3276~16000 時，根據下圖線性轉換得出對應值 0~5000；來源資料之值 ≥ 16000 時，對應值皆為 5000；來源資料之值 ≤ 3276 時，對應值皆為 0。



7-2 邏輯運算指令 (FUN35 ~ 36)

7-2-1 邏輯互斥或 (XOR) 運算 (EXCLUSIVE OR)

FUN35 DP XOR		邏輯互斥或 (XOR) 運算 (EXCLUSIVE OR)														FUN35 DP XOR																																																																																		
指令說明																																																																																																		
		階段圖符號 35DP.XOR 運算控制 — EN — Sa : Sb : D : — D=0 — 結果為0										Sa : XOR 資料 a 或其暫存器號碼 Sb : XOR 資料 b 或其暫存器號碼 D : 存放 XOR 結果之暫存器號碼 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用																																																																																						
		<table><tr><td rowspan="3">範圍 運算元</td><td>WX</td><td>WY</td><td>WM</td><td>WSM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WSM WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 位元 正、負 數</td><td>V、Z P0~P9</td></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>																		範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 位元 正、負 數	V、Z P0~P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○		○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																			
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 位元 正、負 數	V、Z P0~P9																																																																																			
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																		
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																			
D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																			
功能敘述		● 當運算控制 “EN” =1 或由 0→1 (P 指令) 時，將 Sa 和 Sb 資料作邏輯 XOR (Exclusive OR) 運算，亦即將 Sa 和 Sb 之各對應位元 (B0~B15 或 B0~B31) 作比較，任一對應位元之狀態若不相同，則在 D 之該對應位元設為 1，相同則為 0。 ● 若運算結果 D 之所有位元均為 0，則結果為 0 旗號 “D=0” 設為 1。																																																																																																

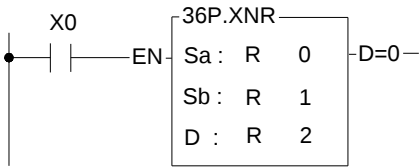
FUN35 DP XOR	邏輯互斥或 (XOR) 運算 (EXCLUSIVE OR)	FUN35 DP XOR																																																			
程式範例																																																					
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td> X0 ● EN 35P.XOR Sa : R 0 Sb : R 1 D : R 2 D=0 </td><td></td></tr></table>			階梯圖	ST	X0 ● EN 35P.XOR Sa : R 0 Sb : R 1 D : R 2 D=0																																																
階梯圖	ST																																																				
X0 ● EN 35P.XOR Sa : R 0 Sb : R 1 D : R 2 D=0																																																					
● 上圖程式例係將暫存器 R0 和 R1 作邏輯互斥或運算後將結果存到 R2 去 Sa Sb <table><tr><td>R0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>R1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table> ⇓ XOR = ⇓ D <table><tr><td>R2</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr></table>			R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0	R2	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1
R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1																																					
R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0																																					
R2	0	1	0	1	0	1	0	1	1	1	0	0	1	0	1	1																																					

7-2-2 邏輯互斥容 (XNR) 運算 (ENCLUSIVE OR)

FUN36 DP XNR		邏輯互斥容 (XNR) 運算 (ENCLUSIVE OR)												FUN36 DP XNR																																																																																	
指令說明																																																																																															
		階段圖符號 36DP.XNR 運算控制 — EN — Sa : — D=0 — 結果為0 Sb : D :								Sa : XNR 資料 a 或其暫存器號碼 Sb : XNR 資料 b 或其暫存器號碼 D : 存放 XNR 結果之暫存器號碼 Sa · Sb · D 可結合 V、Z、P0~P9 作間接定址應用																																																																																					
		<table><tr><th>範圍 運算元</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WSM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 位元 正、負 數</td><td>V、Z P0~P9</td></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>														範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 位元 正、負 數	V、Z P0~P9	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○		○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 位元 正、負 數	V、Z P0~P9																																																																																
Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																
Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																
D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																
功能敘述																																																																																															
		● 當運算控制 “EN” =1 或由 0→1 (P 指令) 時，將 Sa 和 Sb 資料作邏輯 XNR (Exclusive NOR，即 Exclusive NOR 之相反)，亦即將 Sa 和 Sb 之各對應位元 (B0~B15 或 B0~B31) 作比較，任一對應位元之狀態相同，則 D 之該對應位元設為 1，若不同則設為 0。 ● 若運算結果 D 之所有位元均為 0，則結果為 0 旗號 “D=0” 設為 1。																																																																																													

FUN36 DP XNR	邏輯互斥容 (XNR) 運算 (ENCLUSIVE OR)	FUN36 DP XNR
------------------------	--------------------------------------	------------------------

程式範例	
------	--

階梯圖	ST
	

- 上圖程式範例係將暫存器 R0 和 R1 作邏輯互斥容運算後，再將所得結果存到暫存器 R2 去

S a S b	R0	1	0	1	1	1	0	1	1	0	1	1	0	1	1	0	1
	R1	1	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0
⇓ X0 = ⇓																	
D	R2	1	0	1	0	1	0	1	0	0	0	1	1	0	1	0	0

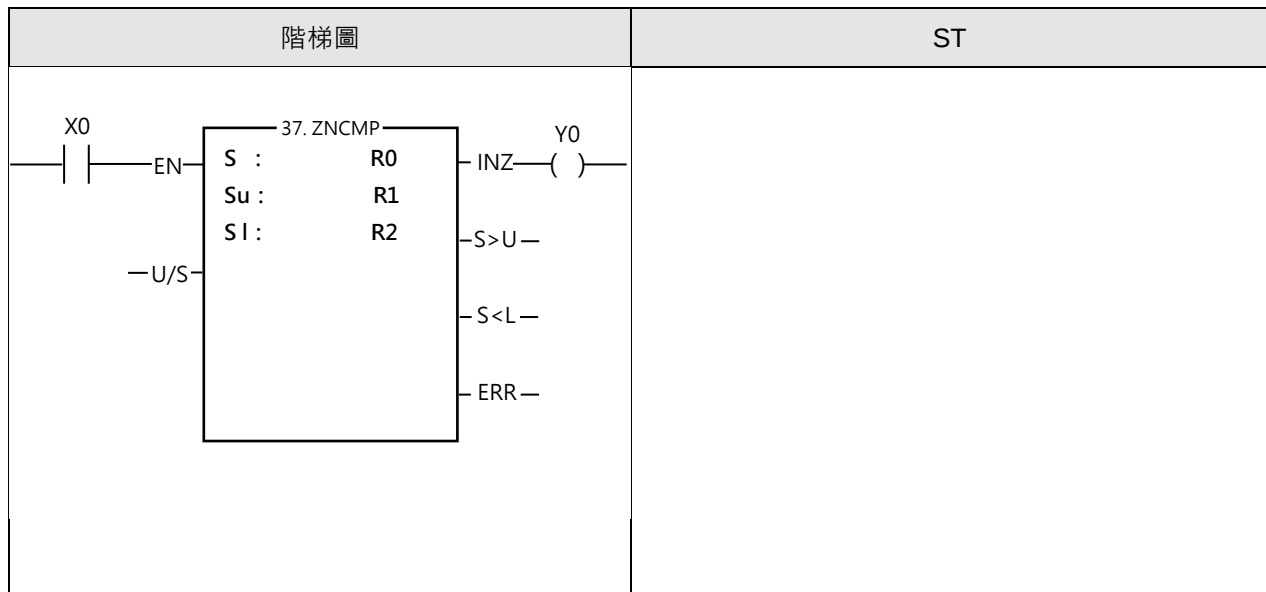
7-3 比較指令 (FUN37)

7-3-1 區域比較 (ZONE COMPARE)

FUN37 D P ZNCMP		區域比較 (ZONE COMPARE)												FUN37 D P ZNCMP			
指令說明																	

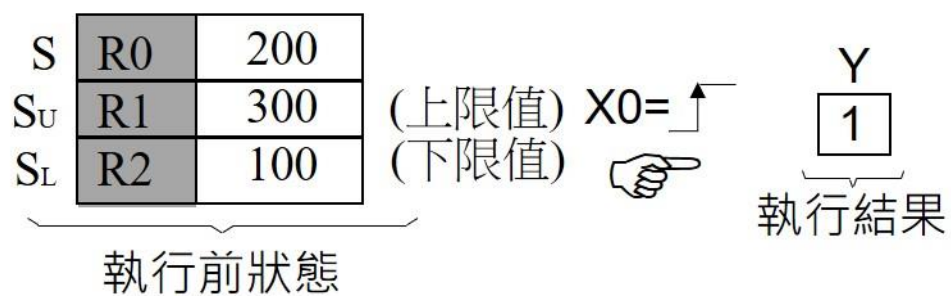
FUN37 D P ZNCMP	區域比較 (ZONE COMPARE)	FUN37 D P ZNCMP
---------------------------	--------------------------	---------------------------

程式範例



- 上圖程式範例係將 R0 之值和由 R1 和 R2 所構成之上、下限區域作比較，設 R0 ~ R2 之數值如下圖左，則可獲得如下圖右之執行結果。

若輸出結果需要為不在區域內，則可用 OUT NOT Y0 即可。



7-4 搬移指令一 (FUN40 ~ 50)

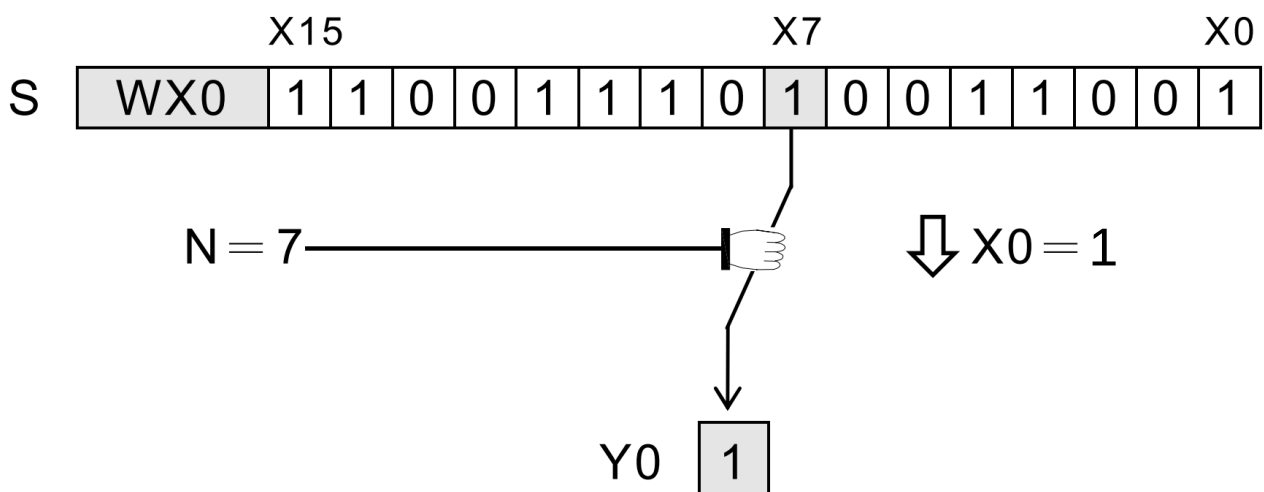
7-4-1 位元資料讀取 (BIT READ)

FUN40 D P BITRD		位元資料讀取 (BIT READ)														FUN40 D P BITRD			
指令說明																			
		階段圖符號 運算控制 — EN — 40DP.BITRD S : N : OTB — 輸出位元 ERR — N值錯誤										S：欲讀取位元之資料或其暫存器號碼 N：指定 S 資料中第 N 個位元資料被讀出 S · N 可結合 V、Z、P0~P9 作間接定址應用							
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR			
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9			
	S	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
	N	○	○	○	○	○	○	○	○	○	○	○	○	○	0~31	○			
	功能敘述																		
		● 當讀取控制“EN”=1 或由 0→1 (P 指令) 時，將 S 資料中之第 N 個位元 (BIT) 取出送到輸出位元“OTB”去。 ● N 之值在 16 位元指令時有效範圍為 0~15，在 32 位元 (D 指令) 時則為 0~31，超出此範圍則 N 值錯誤旗號“ERR”設為 1，且本指令不執行。																	

FUN40 D P BITRD	位元資料讀取 (BIT READ)	FUN40 D P BITRD
程式範例		

階段圖	ST
<pre> graph LR X0[X0] --> EN[EN] subgraph 40_BITRD [40.BITRD] S[S: WX0] N[N: 7] end OTB[OTB] --> Y0[Y0] ERR[ERR] </pre>	

上圖程式範例係自 WX0 (X0 ~ X15) 中讀取第 7 個位元 (X7) 之狀態，再將之送到 Y0 去，其結果如下：



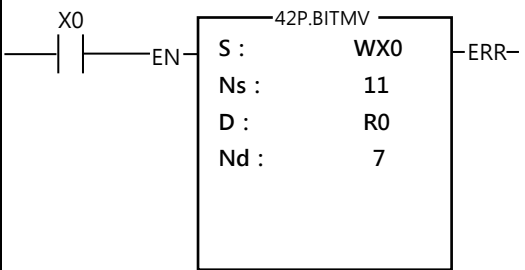
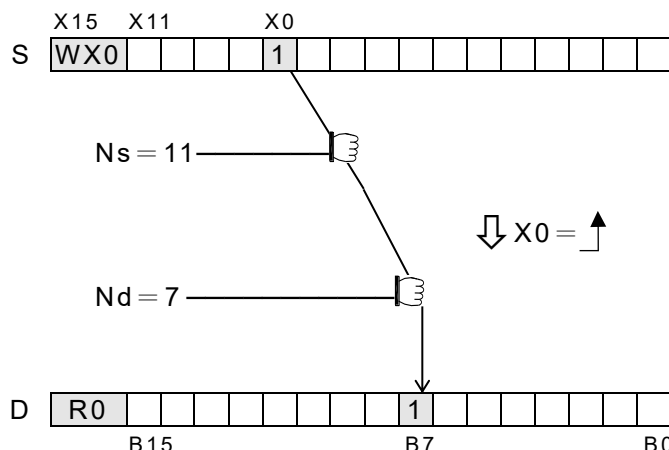
7-4-2 位元資料寫入 (BIT WRITE)

FUN41 DP BITWR		位元資料寫入 (BIT WRITE)												FUN41 DP BITWR	
指令說明															
		</													

FUN41 DP BITWR	位元資料寫入 (BIT WRITE)	FUN41 DP BITWR
階段圖 X0 X1 41. BITWR ERR S : R0 N : 3 EN INB ST		
● N 之值在 16 位元指令時有效範圍為 0 ~ 15，在 32 位元 (D 指令) 時則為 0 ~ 31，若超出此範圍則 N 值錯誤旗號“ERR”設為 1，且本指令不執行。 ● 上圖程式範例係將寫入位元 INB 之狀態寫到 R0 中之 B3 去，假設 X=1，其執行結果如下圖。 X1 1 N = 3 X0 = 1 D R0 B15 B3 B0 1 B3 以外之位元保持不變		

7-4-3 位元資料搬移 (BIT MOVE)

FUN42 DP BITMV		位元資料搬移 (BIT MOVE)												FUN42 DP BITMV			
指令說明																	
階段圖符號 搬移控制 — EN — 42DP.BITMV S : Ns : D : Nd : — ERR — N值錯誤								S : 搬移之來源資料或其暫存器號碼 Ns : 指定 S 中之 Ns 位元為來源位元 D : 搬移之目的暫存器號碼 Nd : 指定 D 中之 Nd 位元為目的位元 S · Ns · D · Nd 可結合 V、Z、P0~P9 作間接定址應用									
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	
		WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	16 或 32 位元	V、Z	
															正、負數	P0~P9	
		WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999			
	S	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
	Ns	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0~31	○
D		○	○	○	○	○	○	○			○	○*	○*	○		○	
Nd	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	0~31	○
功能敘述																	
		● 當搬移控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 中 Ns 所指定之位元狀態搬移至 D 中 Nd 所指定之位元去。 ● Ns 或 Nd 之值在 16 位元指令時有效範圍為 0~15，在 32 位元 (D 指令) 時則為 0~31，超出此範圍，則 N 值錯誤旗號 “ERR” 設為 1，且本指令不執行。															

FUN42 DP BITMV	位元資料搬移 (BIT MOVE)	FUN42 DP BITMV
程式範例		
階梯圖  ST		
上圖程式範例係將 S 中之 B11 (即 X11) 之狀態搬移到 D 中 B7 之位置去，D 中除被寫入之位元 B7 外，其他位元狀態不變。		

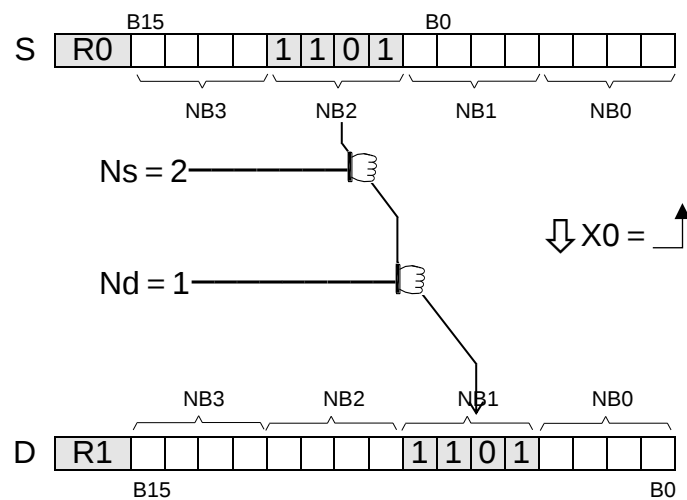
7-4-4 位數 (NIBBLE) 搬移 (NIBBLE MOVE)

FUN43 DP NBMV		位數 (NIBBLE) 搬移 (NIBBLE MOVE)												FUN43 DP NBMV			
指令說明																	
階段圖符號 搬移控制 — EN — 43DP.NBMV — S : — ERR — N值錯誤 Ns : D : Nd :								S : 搬移之來源資料或其暫存器號碼 Ns : 指定 S 中之第 Ns 個位數為來源位數 D : 搬移之目的暫存器號碼 Nd : 指定 D 中之第 Nd 個位數為目的位數 S · Ns · D · Nd 可結合 V、Z、P0~P9 作間接定址應用									
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9		
	S																
	Ns														0~7		
	D											*	*				
	Nd														0~7		
功能敘述																	
● 當搬移控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 中第 Ns 個位數 (Nibble：為 4 個位元所組合。由暫存器之最低位元 B0 起每連續 4 個位元形成一個 Nibble，即 B0~B3 為第 0 個位數，B4~B7 為第 1 個位數，．．．．) 搬移到 D 中 Nd 所指定的那個位數去。 ● Ns 或 Nd 在 16 位元指令時，有效範圍為 0~3，在 32 位元 (D 指令) 時則為 0~7，超出此範圍則 N 值錯誤旗號 “ERR” 設為 1，且本指令不執行。																	

FUN43 DP NBMV	位數 (NIBBLE) 搬移 (NIBBLE MOVE)	FUN43 DP NBMV
程式範例		

階梯圖	ST

- 上圖程式範例係將 S 中之第 2 個位數 NB2 (即 B8~B11) 搬到 D 中之第 1 個位數 NB1 (即 B4~B7) 去，D 中之其他位數則保持不變。



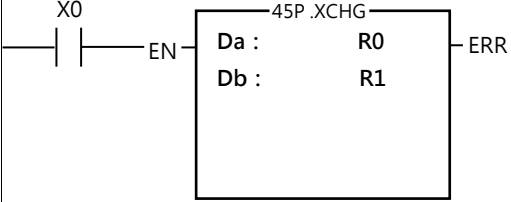
7-4-5 位元組 (BYTE) 搬移 (BYTE MOVE)

FUN44 DP BYMV		位元組 (BYTE) 搬移 (BYTE MOVE)												FUN44 DP BYMV		
指令說明																
		階梯圖符號 搬移控制 — EN — 44DP.BYMV — S : Ns : D : Nd : ERR — N值錯誤								S : 搬移之來源資料或其暫存器號碼 Ns : 指定 S 中之第 Ns 個位元組為來源位元組 D : 搬移之目的暫存器號碼 Nd : 指定 D 中之第 Nd 個位元組為目的位元組 S · Ns · D · Nd 可結合 V、Z、P0~P9 作間接定址應用						
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9
	S	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Ns	○	○	○	○	○	○	○	○	○	○	○	○	○	0~3	○
	D		○	○	○	○	○	○	○		○	○*	○*	○		○
	Nd	○	○	○	○	○	○	○	○	○	○	○	○	○	0~3	○
功能敘述																
		● 當搬移控制“EN” =1 或由 0→1 (P 指令) 時，將 S 中第 Ns 個位元組 (Byte：為 8 個位元組成，由暫存器之最低位元 B0 起每連續 8 個位元形成一個 Byte，即 B0~B7 為第 0 個位元組，B8~B15 為第 1 個位元組．．．．．) 搬移到 D 中第 Nd 個位元組處。 ● Ns 或 Nd 在 16 位元指令時有效範圍為 0~1，在 32 位元 (D 指令) 時則為 0~3，超出此範圍則 N 值錯誤旗號“ERR” 設為 1，且本指令不執行。														

FUN44 DP BYMV	位元組 (BYTE) 搬移 (BYTE MOVE)	FUN44 DP BYMV
程式範例		
階段圖 X0 EN 44DP.BYMV S : R0 Ns : 2 D : R2 Nd : 1 ERR- ST		
● 上圖程式範例係將 S (由 R1R0 所構成之 32 位元暫存器) 中之第 2 個位元組 (即 B16 ~ B23) 搬移到 D (由 R3R2 構成之 32 位元暫存器) 中之第 1 個位元組去。D 中之其他位元組則保持不變。 B15 B0 S R1 R0 Byte3 Byte2 Byte1 Byte0 Ns = 2 Nd = 1 Byte3 Byte2 Byte1 Byte0 B31 B0 D R3 R2 Byte3 Byte2 Byte1 Byte0 ↓ X0 = ↗		

7-4-6 資料互換 (EXCHANGE)

FUN45 DP XCHG		資料互換 (EXCHANGE)										FUN45 DP XCHG			
指令說明															

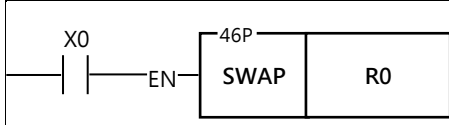
階段圖	ST
	
上圖程式範例係將 16 位元暫存器 R0 和 R1 之資料內容互換。	

FUN45 DP XCHG	資料互換 (EXCHANGE)	FUN45 DP XCHG																																																																								
B15 Da Db <table><tr><td>R0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>R1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> ⇩ X0 = ⇧ B15 B0 Da Db <table><tr><td>R0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>R1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>			R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	R1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	R0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																									
R1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																									
R0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1																																																									
R1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																									

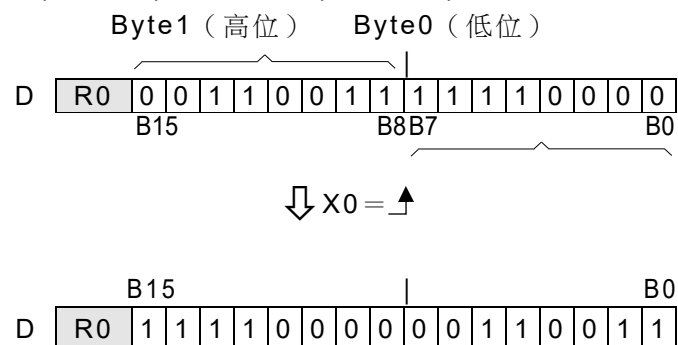
7-4-7 位元組 (BYTE) 資料對換 (BYTE SWAP)

FUN46 P SWAP		位元組 (BYTE) 資料對換 (BYTE SWAP)										FUN46 P SWAP																																						
指令說明																																																		
階段圖符號 對換控制 — EN — 46P. SWAP D						D：執行位元組資料對換之暫存器號碼 D 可結合 V、Z、P0~P9 作間接定址應用																																												
<table><tr><th rowspan="3">運算元 範圍</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>XR</th></tr><tr><td>WX0 WX1008</td><td>WM0 WM9104</td><td>WM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>V、Z P0~P9</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>○</td><td>○</td></tr></table>														運算元 範圍	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR	WX0 WX1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0~P9	D	○	○	○	○	○	○	○	○*	○*	○	○
運算元 範圍	WY	WM	WSM	WS	TMR	CTR	HR	OR	SR	ROR	DR	XR																																						
	WX0 WX1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	V、Z P0~P9																																						
	D	○	○	○	○	○	○	○	○*	○*	○	○																																						
功能敘述																																																		
B15 B8 B7 B0 Byte 1 (high) Byte 0 (low)																																																		

程式範例

階梯圖	ST
	<pre>IF X0 THEN SWAP(D:= R0); END_IF</pre>

上圖程式係將 R0 之低位元組 (B0 ~ B7) 和高位元組 (B8 ~ B15) 之資料互換，其結果如下。



7-5 位移 / 旋轉指令 (FUN51 ~ 54)

7-5-1 向左位移 (SHIFT LEFT)

FUN51 DP SHFL		向左位移 (SHIFT LEFT)														FUN51 DP SHFL																																																																					
指令說明																																																																																					
		階段圖符號 51DP.SHFL 位移控制 — EN — D : — OTB — 移出位元 移入位元 — INB — N : — ERR — N值錯誤										D：被位移之暫存器號碼 N：位移之位元數 D · N 可結合 V、Z、P0~P9 作間接定址應用																																																																									
		<table><tr><th rowspan="3">運算元</th><th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>0 0 或 16 32</td><td>V、Z P0~P9</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>N</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>																		運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	0 0 或 16 32	V、Z P0~P9	D		○	○	○	○	○	○	○		○	○*	○*	○		○	N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																					
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	0 0 或 16 32	V、Z P0~P9																																																																					
	D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																					
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																					
功能敘述																																																																																					
		● 當位移控制“EN” =1 或由 0→1 (P 指令) 時，將 D 暫存器之資料向左 (由低位往高位) 連續移動 N 個位元，在最低位元 B0 左移後，其空位以移入位元 INB 填補之，同時將移出位元 B15 或 B31 (D 指令) 之狀態送至移出位元“OTB”去。 ● N 之有效範圍在 16 位元指令為 0~16，輸入 0 不位移，在 32 位元 (D 指令) 則為 0~32，輸入 0 不位移，若超出此範圍則 N 值錯誤旗號“ERR”設為 1，且本指令不執行。																																																																																			

FUN51 DP SHFL	向左位移 (SHIFT LEFT)	FUN51 DP SHFL
程式範例		
階段圖 X0 — —EN 51P. SHFL D : R0 N : 4 Y0 —OTB—() INB —ERR— ST ● 上圖程式範例係將暫存器 R0 之資料連續向左位移 4 個位元 (4 次)，下圖為其執行結果。 Y0 □ ← B15 0 0 1 1 0 0 1 0 1 1 1 1 0 0 0 0 R0 B0 ← INB 1 △ * ⇩ X0 = ⇧ Y0 1 * B15 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1 1 R0 B0 △ △ △ △ INB 1 △		

7-5-2 向右位移 (SHIFT RIGHT)

FUN52 DP SHFR		向右位移 (SHIFT RIGHT)														FUN52 DP SHFR																																																																																				
指令說明																																																																																																				
		階段圖符號 52DP.SHFR 位移控制 — EN↑ 移出位元 — INB D : N : OTB — 移出位元 ERR — N值錯誤										D：被位移之暫存器號碼 N：位移之位元數 D，N 可結合 V、Z、P0~P9 作間接定址應用																																																																																								
		<table><tr><th rowspan="3">運算元</th><th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>WX0</td><td>WY0</td><td>WM0</td><td>WM9120</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td>0 0</td><td>V、Z</td></tr><tr><td>WX1008</td><td>WY1008</td><td>WM9104</td><td>WM25584</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td>16 32</td><td>P0~P9</td></tr><tr><td>D</td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>N</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>																		運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	0 0	V、Z	WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	16 32	P0~P9	D			○	○	○	○	○	○	○		○	○*	○*	○		○	N		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																				
	WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	0 0	V、Z																																																																																					
	WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	16 32	P0~P9																																																																																					
D			○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																				
N		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																				
功能敘述																																																																																																				
		當位移控制“EN”=1 或 “EN↑”（P 指令）由 0→1 時，將 D 暫存器之資料向右（由高位往低位）連續移動 N 個位元，在最高位元 B15 或 B31（D 指令）右移後，其空位由移入位元 INB 填補之，同時將移出位元 B0 之狀態送至移出位元“OTB”去。 N 之有效範圍在 16 位元指令為 0~16，輸入 0 不位移，在 32 位元（D 指令）則為 0~32，輸入 0 不位移，超出此範圍則 N 值錯誤旗號“ERR”設為 1，且本指令不執行。																																																																																																		

FUN52 DP SHFR	向右位移 (SHIFT RIGHT)	FUN52 DP SHFR
程式範例		

階梯圖	ST

● 上圖程式範例係將暫存器 R0 之資料連續向右位移 15 個位元 (15 次)，下圖為其執行結果。

INB

B15

0

→

R0

B0

10101010101010

→

Y0

△

*

⇩ X0 = 1 ⇧

INB

B15

0

R0

B0

000000000000001

Y0

0

△

△

△

△

△

△

△

△

△

△

△

△

△

△

△

*

7-5-3 向左旋轉 (ROTATE LEFT)

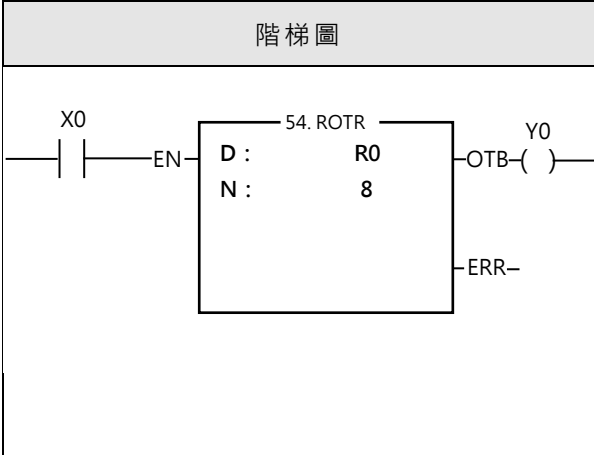
FUN53 DP ROTL		向左旋轉 (ROTATE LEFT)														FUN53 DP ROTL																																																																		
指令說明																																																																																		
		階梯圖符號 旋轉控制 — EN↑ 53DP.ROTL D : N : OTB — 旋出位元 ERR — N值錯誤										D：被旋轉之暫存器號碼 N：旋轉之位元數 D・N 可結合 V、Z、P0~P9 作間接定址應用																																																																						
		<table><tr><th rowspan="3">運算元 範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th>WX0 WX1008</th><th>WY0 WY1008</th><th>WM0 WM9104</th><th>WM9120 WM25584</th><th>WS0 WS3088</th><th>T0 T1023</th><th>C0 C1279</th><th>R0 R34767</th><th>R34768 R34895</th><th>R35024 R35151</th><th>R35280 R43223</th><th>R43224 R47319</th><th>D0 D7999</th><th>0 0 或 16 32</th><th>V・Z P0~P9</th></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>N</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>																		運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	0 0 或 16 32	V・Z P0~P9	D		○	○	○	○	○	○	○		○	○*	○*	○		○	N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																			
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	0 0 或 16 32	V・Z P0~P9																																																																			
	D		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																		
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																			
功能敘述																																																																																		
		● 當旋轉控制“EN”=1 或“EN↑”(P 指令) 由 0→1 時，將 D 暫存器之資料向左 (低位往高位，即 16 位元指令 B0→B1，B1→B2，．．．．．，B14→B15，B15→B0。32 位元指令則為 B0→B1，B1→B2，．．．．．，B30→B31，B31→B0) 連續旋轉 N 位元，同時並將旋出之 B15 或 B31 (D 指令) 位元狀態送到旋出位元“OTB”去。 ● N 之有效範圍在 16 位元指令為 0~16，輸入 0 不旋轉，在 32 位元 (D 指令) 則為 0~32，輸入 0 不旋轉，超出此範圍則 N 值錯誤旗號“ERR”設為 1，且本指令不執行。																																																																																

FUN53 DP ROTL	向左旋轉 (ROTATE LEFT)	FUN53 DP ROTL
程式範例		
階梯圖 X0 — —EN 53P. ROTL D : R0 N : 9 OTB—()—Y0 ERR— ST		
● 上圖程式範例係將暫存器 R0 之資料連續向左旋轉 9 次，下圖為其執行結果。 R0 B0 11110000101010 * ↘ Y0 ↓ X0 = ↑ B15 R0 B0 0101010111100001 * 1 Y0 *		

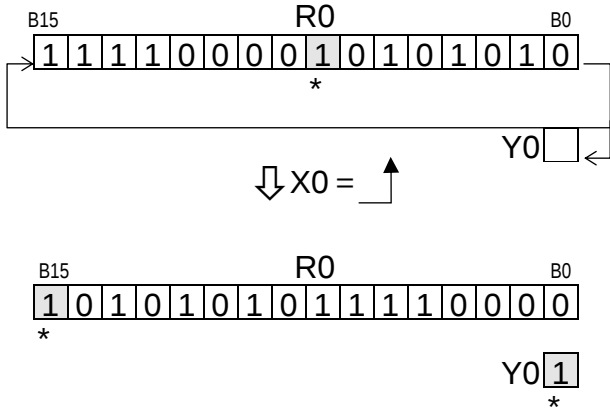
7-5-4 向右旋轉 (ROTATE RIGHT)

FUN54 DP ROTR		向右旋轉 (ROTATE RIGHT)												FUN54 DP ROTR																																																																																						
指令說明																																																																																																				
		階段圖符號 54DP.ROTR 旋轉控制 — EN↑ D : N : OTB — 旋出位元 ERR — N值錯誤										D：被旋轉之暫存器號碼 N：旋轉之位元數 D、N 可結合 V、Z、P0~P9 作間接定址應用																																																																																								
		<table><tr><th rowspan="3">運算元</th><th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0</td><td>WY0</td><td>WM0</td><td>WM9120</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td>0 0</td><td>V、Z</td></tr><tr><td></td><td>WX1008</td><td>WY1008</td><td>WM9104</td><td>WM25584</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td>16 32</td><td>P0~P9</td></tr><tr><td>D</td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>N</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>																運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	0 0	V、Z		WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	16 32	P0~P9	D			○	○	○	○	○	○	○		○	○*	○*	○		○	N		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																				
		WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	0 0	V、Z																																																																																				
		WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	16 32	P0~P9																																																																																				
D			○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																				
N		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																				
功能敘述																																																																																																				
		● 當旋轉控制“EN”=1 或“EN↑”(P 指令) 由 0→1 時，將 D 暫存器之位元資料向右 (高位往低位，即 16 位元指令 B15→B14，B14→B13，．．．．．，B1→B0，B0→B15。32 位元指令則為 B31→B30，B30→B29，．．．．．，B1→B0，B0→B31) 連續旋轉 N 位元，同時並將旋出之 B0 位元狀態送到旋出位元“OTB”去。 ● N 之有效範圍在 16 位元指令為 0~16，輸入 0 不旋轉，在 32 位元 (D 指令) 則為 0~32，輸入 0 不旋轉，超出此範圍則 N 值錯誤旗號“ERR”設為 1，且本指令不執行。																																																																																																		

FUN54 DP ROTR	向右旋轉 (ROTATE RIGHT)	FUN54 DP ROTR
程式範例		

階梯圖	ST
	

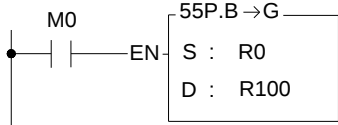
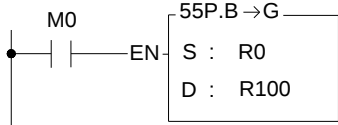
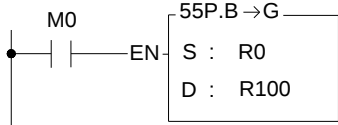
● 上圖程式範例係將暫存器 R0 之資料連續向右旋轉 8 次，下圖為其執行結果。



7-6 數碼變換指令 (FUN55 ~ 64)

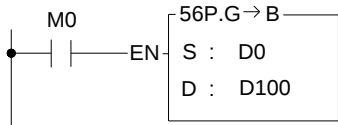
7-6-1 二進制碼轉換格雷碼(B→G)

FUN55 DP B→G		二進制碼轉換格雷碼 (BINARY-CODE TO GRAY-CODE CONVERSION)												FUN55 DP B→G			
指令說明																	

FUN55 DP B→G	二進制碼轉換格雷碼 (BINARY-CODE TO GRAY-CODE CONVERSION)	FUN55 DP B→G				
程式範例一						
當 M0 由 OFF→ON 時，將 R0(二進制碼)轉換為格雷碼，然後存入 R100。						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>IF R_TRIG(S:= M0) THEN BintoGray(S:= R0, D:= R100); END_IF</td></tr></table>			階梯圖	ST		IF R_TRIG(S:= M0) THEN BintoGray(S:= R0, D:= R100); END_IF
階梯圖	ST					
	IF R_TRIG(S:= M0) THEN BintoGray(S:= R0, D:= R100); END_IF					
1001010101010011B ➔ R100 = 1101111111111010B						
程式範例二						
當 M0 ON 時，將 DR0(二進制碼)轉換為格雷碼，然後存入 DR100。						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>IF R_TRIG(S:= M0) THEN BintoGray(S:= DR0, D:= DR100); END_IF</td></tr></table>			階梯圖	ST		IF R_TRIG(S:= M0) THEN BintoGray(S:= DR0, D:= DR100); END_IF
階梯圖	ST					
	IF R_TRIG(S:= M0) THEN BintoGray(S:= DR0, D:= DR100); END_IF					
DR0 = 00110111001001000010111100010100B ➔ DR100 = 00101100101101100011100010011110B						

7-6-2 格雷碼轉換二進制碼(G→B)

[illegible]

FUN56 DP G→B	格雷碼轉換二進制碼 (GRAY-CODE TO BINARY-CODE CONVERSION)	FUN56 DP G→B
程式範例一		
當 M0 由 OFF→ON 時，將 D0(格雷碼)轉換為二進制碼，然後存入 D100。		
階梯圖 		ST IF R_TRIG(S:= M0) THEN GraytoBin(S:= D0, D:= D100); END_IF
D0 = 1001010101010011B ➔ D100 = 1110011001100010B		
程式範例二		
當 M0 ON 時，將 DD0(格雷碼)轉換為二進制碼，然後存入 DD100。		
階梯圖 		ST IF R_TRIG(S:= M0) THEN GraytoBin_D(S:= D0, D:= D100); END_IF
DD0 = 00110111001001000010111100010100B ➔ DD100 = 00100101110001111100101000011000B		

7-6-3 時：分：秒→秒 (HOUR : MINUTE : SECOND→SECOND)

FUN61 P →SEC	時：分：秒→秒 (HOUR：MINUTE：SECOND→SECOND)												FUN61 P →SEC																																																							
指令說明																																																																				
階段圖符號 變換控制 — EN — 61P.→ SEC S： D： — D=0 — 結果為0 S：欲變換之時間資料暫存器起頭號碼 D：存放結果之暫存器起頭號碼																																																																				
<table><tr><td rowspan="4">範圍 運算元</td><td>WX</td><td>WY</td><td>WM</td><td>WSM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WSM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td></tr></table>															範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	S	○	○	○	○	○	○	○	○	○	○	○	○	D		○	○	○	○	○	○	○		○	○*	○*	○
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR																																																							
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999																																																							
	S	○	○	○	○	○	○	○	○	○	○	○	○																																																							
	D		○	○	○	○	○	○	○		○	○*	○*	○																																																						
功能敘述																																																																				

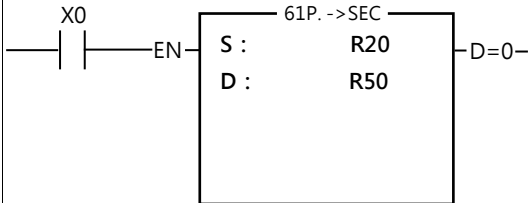
FUN61 P →SEC	時：分：秒→秒 (HOUR : MINUTE : SECOND→SECOND)	FUN61 P →SEC
● 當變換控制“EN”=1 或由 0→1 (P 指令) 時，將 S~S+2 的「時：分：秒」資料轉換為等值之秒數值後存入由 D 和 D+1 合併而成之 32 位元暫存器中。若結果=0，則“D=0”旗號設為 1。 ● M SERIES 指令中，和「時：分：秒」時間相關之指令 (FUN61 和 62)，其時間資料之格式如下圖所示將自動合併連續 3 個暫存器 (Word) 來當時間值使用，其起頭第一個為秒數 (Second) 暫存器，次一個為分數 (Minute) 暫存器，最後一個則為時數 (Hour) 暫存器。每個暫存器之 16 個位元中只有 B14~B0 用以表示時間值，而其最高位元 B15 則用以表示各該時間值之正、負。B15 為 0 表示該時間為正，B15 若為 1 則表示該時間值為負，B14~B0 之時間值是以二進制表示，當時間值為負時，B14~B0 則以 2 的補數表示。運算之秒數結果為「時：分：秒」三個暫存器之秒數加減結果。		
● 任一「時：分：秒」時間資料，除了以 FUN61 或 62 兩指令去存取時才會自動合併使用，其他指令均會將之視為個別之一般暫存器，不會自動合併使用，3 個暫存器間沒有任何關係，因此您可個別對時、分、秒之任一資料運算，結果互不影響。		
程式範例		

FUN61 **P**
→SEC

時：分：秒→秒
(HOUR : MINUTE : SECOND→SECOND)

FUN61 **P**
→SEC

- 下圖程式例，本指令會將 R20 開始之 3 個資料視為「時：分：秒」資料而將之轉換成等值之秒數值後再存入 R50~R51 所組成之 32 位元暫存器中，其結果如下圖右。

階梯圖	ST
	<pre>IF R_TRIG(S:= X0) THEN ToSEC(S:= R20, D:=R50); END_IF</pre>



S {	R20	0E11H	= 3601 秒
	R21	FD2FH	= -721 分
	R22	03F3H	= 1011 小時
↓ X0 = ↑			
D {	R50	EE45H	= 3599941 秒
	R51	0036H	

7-6-4 秒數→時：分：秒 (SECOND→HOUR : MINUTE : SECOND)

FUN62 P →HMS		秒數→時：分：秒 (SECOND→HOUR：MINUTE：SECOND)													FUN62 P →HMS			
指令說明																		
階段圖符號 變換控制 — EN — 62P.→HMS S： D=0 — 結果為0D： OVR — 超出範圍										S：欲變換之秒數資料暫存器起頭號碼 D：變換結果〔時：分：秒〕存放之暫存器起頭號碼								
範圍 運算元		WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K			
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WSM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	- 117968399 117964799			
S		○	○	○	○	○	○	○	○	○	○	○	○	○	○			
D			○	○	○	○	○	○	○		○	○*	○*	○				
功能敘述																		

FUN62 P →HMS	秒數→時：分：秒 (SECOND→HOUR : MINUTE : SECOND)	FUN62 P →HMS												
● 當變換控制“EN” =1 或由 0→1 (P 指令) 時，將 S~S+1 之 32 位元秒數資料轉換為等值之〔時：分：秒〕時間值存入 D~D+2 三個連續之暫存器中，本指令所有資料均以二進碼表示 (若為負值則以 2 的補數表示)。														
B15 B0 B15 S 秒數值 B15 S + B31 B16 ↑ 秒數暫存器之最高位元 B31 表 該秒數值之正負號 ⇒ B0 <table style="border-collapse: collapse; width: 200px;"> <tr> <td style="width: 30px; text-align: center;">D</td> <td style="text-align: center;">(時)</td> <td style="text-align: center;">B0</td> <td style="text-align: center;">- 59 秒 ~ 59 秒</td> </tr> <tr> <td style="text-align: center;">D+1</td> <td style="text-align: center;">(分)</td> <td style="text-align: center;">B16</td> <td style="text-align: center;">- 59 分 ~ 59 分</td> </tr> <tr> <td style="text-align: center;">D+2</td> <td style="text-align: center;">(時)</td> <td style="text-align: center;">B32</td> <td style="text-align: center;">- 32768 小時 ~ 32767 小時</td> </tr> </table> ↑ 每個暫存器之最高位元 B15 均用 以表示該〔時：分：秒〕值之正 負號			D	(時)	B0	- 59 秒 ~ 59 秒	D+1	(分)	B16	- 59 分 ~ 59 分	D+2	(時)	B32	- 32768 小時 ~ 32767 小時
D	(時)	B0	- 59 秒 ~ 59 秒											
D+1	(分)	B16	- 59 分 ~ 59 分											
D+2	(時)	B32	- 32768 小時 ~ 32767 小時											
● 如上圖所示本指令轉成〔時：分：秒〕時間後，其〔分：秒〕值只可能為-9~59，而時數則可為-32768~32767 小時，因此 D 之最大極限是-32768 小時-59 分-59 秒至 32767 小時 59 分 59 秒，分別對應到 S 之秒數為-117968399 秒~117964799 秒。若 S 值超出此範圍 D 將放不下，此時本指令便不執行，並將超出範圍旗號“OVR”設為 1。若 S 為 0 則結果為零旗號“D=0”會設為 1。														

FUN62 P →HMS	秒數→時：分：秒 (SECOND→HOUR : MINUTE : SECOND)	FUN62 P →HMS
程式範例		

- 下圖程式為本指令執行之結果範例，注意暫存器內容值均為二進制值，其右邊為其等效之 10 進制表示值。

R0	5D17H	} 6315287 秒
R1	0060H	

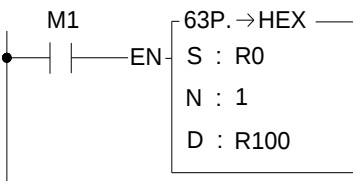
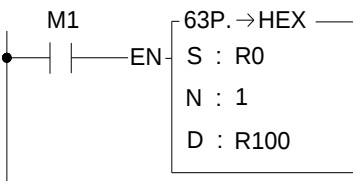
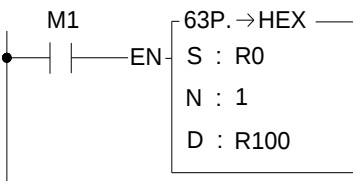
⇓ X0 = ⬆

R10	002FH	47 秒
R11	000EH	14 分
R12	06DAH	1754 小時

階梯圖	ST
	<pre> IF R_TRIG(S:= X0) THEN ToHMS(S:= R0, D:=R10); END_IF </pre>

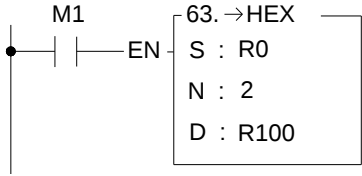
7-6-5 ASCII 碼轉換為十六進制值 (ASCII→HEX)

FUN63 P →HEX		ASCII 碼轉換為十六進制值												FUN63 P →HEX			
指令說明																	
		階段圖符號 變換控制 — EN — 63P. → HEX — S : N : D : ERR — 錯誤訊息								S : 來源暫存器之起始號碼 N : 欲轉 ASCII 碼為十六進制值之個數 D : 存放結果 (十六進制值) 之暫存器起始號碼 S · N · D 可結合 V、Z、P0~P9 作間接定址應用							
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9	
	S	○	○	○	○	○	○	○	○	○	○	○	○	○		○	
	N	○	○	○	○	○	○	○	○	○	○	○	○	○	1~511	○	
	D		○	○	○	○	○	○	○		○	○*	○*	○		○	
功能敘述																	
		● 當換變控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 開始之連續 N 個 16 位元暫存器 (Low Byte 有效) 之 ASCII 碼轉換為十六進制值，並將結果存入由 D 所指定開始之暫存器。每 4 個 ASCII 碼由一個暫存器儲存，未對應到 ASCII 碼之暫存器內容維持原值不變。 ● 當 N 之值為 0 或大於 511 時，運算不執行。 ● 當 ASCII 碼錯誤時 (非 30H ~ 39H 或 41H ~ 46H)，輸出 “ERR” ON。 ● 此指令最大用途係將通訊埠 1~2 所接收到外界 ASCII 週邊 (以 ASCII 碼傳送數值給 PLC) 之 ASCII 數碼轉換為 CPU 能夠直接處理之十六進制值。															

FUN63 P →HEX	ASCII 碼轉換為十六進制值	FUN63 P →HEX				
程式範例一						
M1 由 OFF→ON 時，轉 ASCII 碼為十六進制值 將 R0 之 ASCII 碼轉換為十六進制值並存入 R100 之 Nibble 0 (Nibble1 ~ Nibble3 不變) <table><tr><th>階 梯 圖</th><th>ST</th></tr><tr><td></td><td><pre>IF R_TRIG(S:= M1) THEN ToHEX(S:= R0, N:= 1, D:= R100); END_IF</pre></td></tr></table> 原 R100 = 0000H R0 = 0039H (9) → R100 = 0009H			階 梯 圖	ST		<pre>IF R_TRIG(S:= M1) THEN ToHEX(S:= R0, N:= 1, D:= R100); END_IF</pre>
階 梯 圖	ST					
	<pre>IF R_TRIG(S:= M1) THEN ToHEX(S:= R0, N:= 1, D:= R100); END_IF</pre>					
程式範例二						

M1 ON 時・轉 ASCII 碼為十六進制值

將 R0 與 R1 之 ASCII 碼轉換為十六進制值並存入 R100 之低位元組 (高位元組不變)

階梯圖	ST
	<pre>IF M1 THEN ToHEX(S:= R0, N:= 2, D:= R100); END_IF</pre>

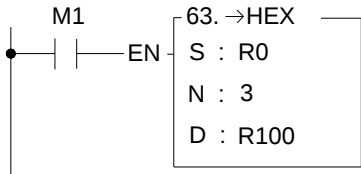
R0 = 0039H (9) 原 R100 = 0000H

R1 = 0041H (A) → R100 = 009AH

FUN63  →HEX	ASCII 碼轉換為十六進制值	FUN63  →HEX
程式範例三		

M1 ON 時・轉 ASCII 碼為十六進制值

將 R0 ~ R2 之 ASCII 碼轉換為十六進制值並存入 R100 (Nibble 3 不變)

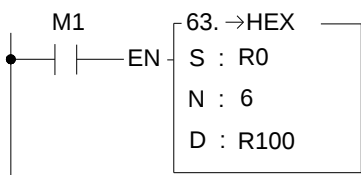
階梯圖	ST
	<pre>IF M1 THEN ToHEX(S:= R0, N:= 3, D:= R100); END_IF</pre>

R0 = 0039H (9) 原 R100 = 0000H
 R1 = 0041H (A)
 R2 = 0045H (E) → R100 = 09AEH

程式範例四	
-------	--

M1 ON 時・轉 ASCII 碼為十六進制值

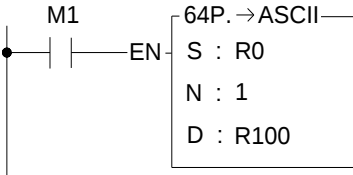
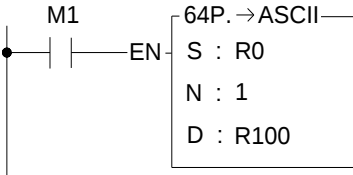
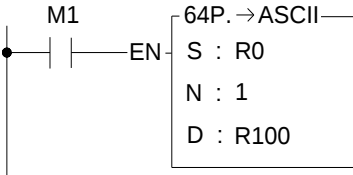
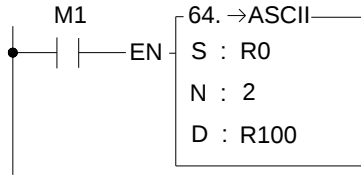
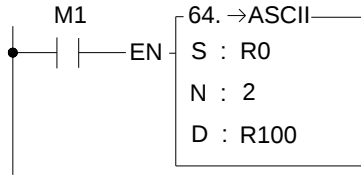
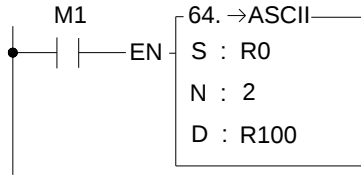
將 R0 ~ R5 之 ASCII 碼轉換為十六進制值並存入 R100 ~ R101

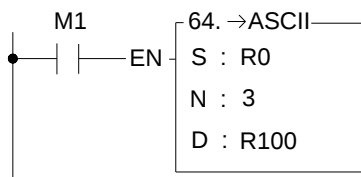
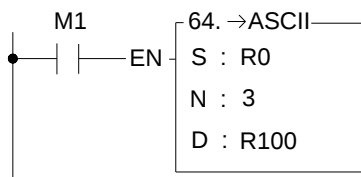
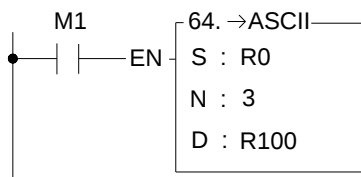
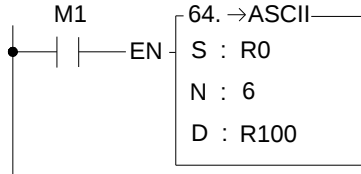
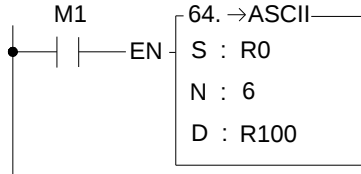
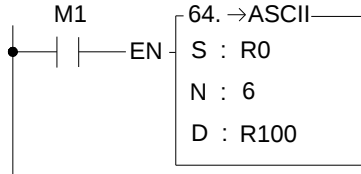
階梯圖	ST
	<pre>IF M1 THEN ToHEX(S:= R0, N:= 6, D:= R100); END_IF</pre>

R0 = 0031H (1) 原 R100 = 0000H
 R1 = 0032H (2) R101 = 0000H
 R2 = 0033H (3)
 R3 = 0034H (4)
 R4 = 0035H (5) → R100 = 3456H
 R5 = 0036H (6) R101 = 0012H

7-6-6 十六進制值轉換為 ASCII 碼 (HEX→ASCII)

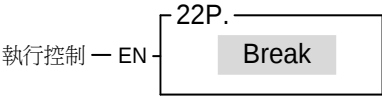
FUN64 P →ASCII		十六進制值轉換為 ASCII 碼														FUN64 P →ASCII			
指令說明																			
		階段圖符號 變換控制 — EN — 64P.→ ASCII — S : N : D :														S : 來源暫存器之起始號碼 N : 欲轉十六進制值為 ASCII 碼之個數 D : 存放結果 (ASCII 碼) 之暫存器起始號碼 S · N · D 可結合 V、Z、P0~P9 作間接定址應用			
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR			
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 位元 正數	V、Z P0~P9			
	S	○	○	○	○	○	○	○	○	○	○	○	○	○		○			
	N	○	○	○	○	○	○	○	○	○	○	○	○	○	1~511	○			
	D		○	○	○	○	○	○	○		○	○*	○*	○		○			
功能敘述																			
		● 當變換控制 “EN” =1 或由 0→1 (P 指令) 時，將 S 開始之暫存器連續 N 個位數 (4 位元) 之十六進制值轉換為 ASCII 碼，並將結果存入 D 所指定開始暫存器之低位元組。(高位元組維持原值不變)。 ● 當 N 之值為 0 或大於 511 時，運算不執行。 ● 此指令最大用途係將 PLC 處理完之數值資料轉換為 ASCII 碼透過通訊埠 1~2 傳送給 ASCII 週邊設備。																	

FUN64 P →ASCII	十六進制值轉換為 ASCII 碼	FUN64 P →ASCII				
程式範例一						
M1 由 OFF→ON 時，轉十六進制值為 ASCII 碼 將 R0 之 Nibble 0 轉換為 ASCII 碼並存入 R100 (高位元組不變) <table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF R_TRIG(S:= M1) THEN ToASCII (S:= R0, N:= 1, D:= R100); END_IF</pre></td></tr></table> R0 = 0009H → R100 = 0039H (9)			階梯圖	ST		<pre>IF R_TRIG(S:= M1) THEN ToASCII (S:= R0, N:= 1, D:= R100); END_IF</pre>
階梯圖	ST					
	<pre>IF R_TRIG(S:= M1) THEN ToASCII (S:= R0, N:= 1, D:= R100); END_IF</pre>					
程式範例二						
M1 ON 時，轉十六進制值為 ASCII 碼 將 R0 之 NB0 ~ NB1 轉換為 ASCII 碼並存入 R100 ~ R101 (高位元組皆維持原值不變) <table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF M1 THEN ToASCII (S:= R0, N:= 2, D:= R100); END_IF</pre></td></tr></table> R0 = 009AH → R100 = 0039H (9) R101 = 0041H (A)			階梯圖	ST		<pre>IF M1 THEN ToASCII (S:= R0, N:= 2, D:= R100); END_IF</pre>
階梯圖	ST					
	<pre>IF M1 THEN ToASCII (S:= R0, N:= 2, D:= R100); END_IF</pre>					

FUN64 P →ASCII	十六進制值轉換為 ASCII 碼	FUN64 P →ASCII				
程式範例三						
M1 ON 時・轉十六進制值為 ASCII 碼 將 R0 之 NB0 ~ NB2 轉換為 ASCII 碼並存入 R100 ~ R102						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>IF M1 THEN ToASCII (S:= R0, N:= 3, D:= R100); END_IF</td></tr></table>		階梯圖	ST		IF M1 THEN ToASCII (S:= R0, N:= 3, D:= R100); END_IF	
階梯圖	ST					
	IF M1 THEN ToASCII (S:= R0, N:= 3, D:= R100); END_IF					
R0 = 0123H → R100 = 0031H (1) R101 = 0032H (2) R102 = 0033H (3)						
程式範例四						
M1 ON 時・轉十六進制值為 ASCII 碼 將 R0 ~ R1 之 NB0 ~ NB5 轉換為 ASCII 碼並存入 R100 ~ R105						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>IF M1 THEN ToASCII (S:= R0, N:= 6, D:= R100); END_IF</td></tr></table>		階梯圖	ST		IF M1 THEN ToASCII (S:= R0, N:= 6, D:= R100); END_IF	
階梯圖	ST					
	IF M1 THEN ToASCII (S:= R0, N:= 6, D:= R100); END_IF					
R0 = 3456H → R100 = 0031H (1) R1 = 0012H R101 = 0032H (2) R102 = 0033H (3) R103 = 0034H (4) R104 = 0035H (5) R105 = 0036H (6)						

7-7 流程控制指令 (FUN22、FUN65 ~ 71)

7-7-1 BREAK (BREAK)

FUN22 P BREAK	FOR 與 NEXT 迴圈之跳出指令 (BREAK)	FUN22 P BREAK
指令說明	階段圖符號 	
功能敘述	● 當執行控制“EN” =1 或由 0→1 (P 指令) 時，將跳出該 FOR 與 NEXT 構成之迴圈程式。 ● 在 FOR 與 NEXT 指令所構成之迴圈程式內，如需提前跳出該迴圈，則可使用本指令而不必等到指定之迴圈次數執行完畢才能跳出該迴圈。 ● 本指令必須使用於 FOR 與 NEXT 指令所構成之迴圈內。	

FUN22 P BREAK	FOR 與 NEXT 迴圈之跳出指令 (BREAK)	FUN22 P BREAK
------------------	-------------------------------	------------------

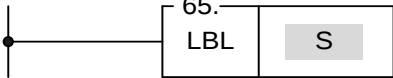
程式範例

階梯圖	ST
	<pre> V := 0; FOR D10 := 0 TO 10 BY 1 DO IF D100 = R0V THEN M200 := TRUE; END_IF IF M200 THEN BREAK; END_IF V := V + 1; END_FOR D1000 := V; </pre>

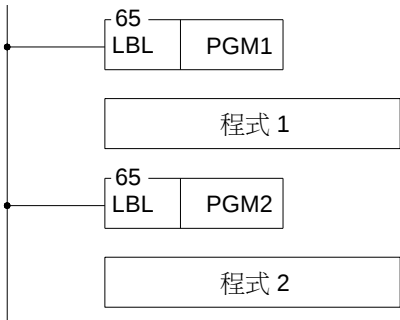
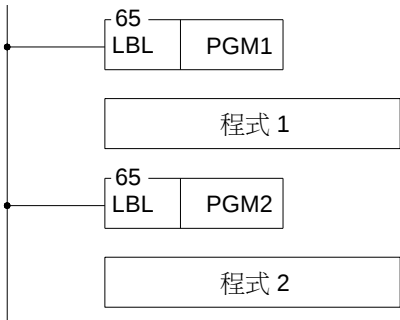
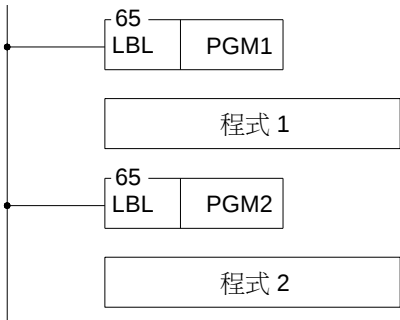
範例說明:

暫存器 D10 之值決定 FOR 與 NEXT 指令所構成之迴圈程式應被執行次數;迴圈內程式用來找尋由 R0 為起始之表格內是否有與 D100 相同內容者，如有，則停止找尋並跳出迴圈程式;如沒有，則該迴圈程式會被執行 D10 所指定之次數。M200 之狀態反應找尋結果，暫存器 D100 為找尋結果之指標暫存器。

7-7-2 標記 (LABEL)

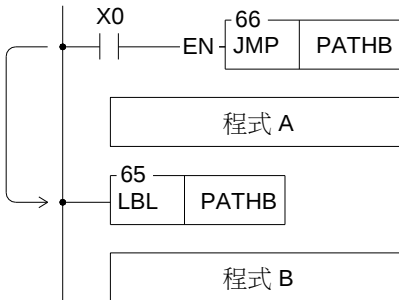
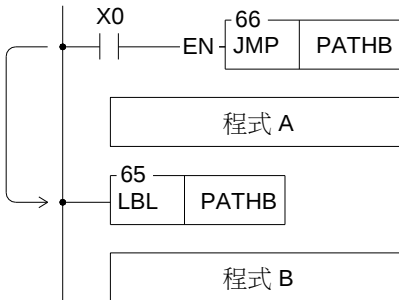
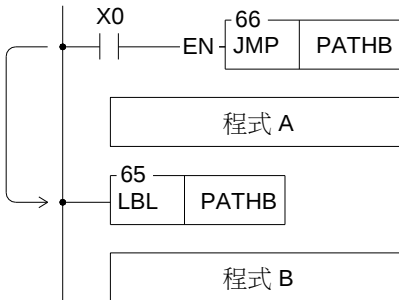
FUN65 LBL	標記 (LABEL)	FUN65 LBL
指令說明		
階梯圖符號 		S : 英文 / 數字 1 ~ 6 字
功能敘述	※僅支援在主程式與副程式中使用	
● 本指令用於標示程式中某一特定位址，以供程式跳躍 (JUMP) 到此標記所在之位址來執行，或當作中斷服務程式或副程式之名稱，以供中斷或呼叫 (CALL) 之用。若不需作跳躍或呼叫等之流程控制，亦可作標記來對程式作註解，以利程式之辨識或提高可讀性。 ● 本指令只當程式位址標記以供流程控制或註解用，指令本身不會執行任何動作，程式中有沒有本指令，程式執行結果均不受其影響。 ● 標記名稱可以 1 ~ 6 個任意英文字母或數字組成，但不得重複，且下列標記名稱是保留給中斷功能使用，稱之為“保留字”，一般程式標記不得使用：		

FUN65 LBL	標記 (LABEL)	FUN65 LBL										
<table><tr><th>保留字</th><th>中斷服務程式名稱</th></tr><tr><td>X0+I ~ X7+I (INT0 ~ INT7) X0-I ~ X7-I (INT0- ~ INT7-)</td><td>外部 X0 ~ X7 之中斷服務程式名稱</td></tr><tr><td>HSC0I ~ HSC7I</td><td>HSC0 ~ HSC7 之中斷服務程式名稱</td></tr><tr><td>STM0I (1MS) · STM1I (1MS) · STM2I (1MS) · STM3I (1MS) · LTM0I (10MS) · LTM1I (10MS) · LTM2I (10MS) · LTM3I (10MS) ·</td><td>PLC 內部 1mS · 10mS · 2 種定時中斷之 服務程式名稱</td></tr><tr><td>HSTAI (ATMRI) · HST0I~HST3I</td><td>高速定時中斷服務程式名稱 以 0.1mS 為單位</td></tr></table>			保留字	中斷服務程式名稱	X0+I ~ X7+I (INT0 ~ INT7) X0-I ~ X7-I (INT0- ~ INT7-)	外部 X0 ~ X7 之中斷服務程式名稱	HSC0I ~ HSC7I	HSC0 ~ HSC7 之中斷服務程式名稱	STM0I (1MS) · STM1I (1MS) · STM2I (1MS) · STM3I (1MS) · LTM0I (10MS) · LTM1I (10MS) · LTM2I (10MS) · LTM3I (10MS) ·	PLC 內部 1mS · 10mS · 2 種定時中斷之 服務程式名稱	HSTAI (ATMRI) · HST0I~HST3I	高速定時中斷服務程式名稱 以 0.1mS 為單位
保留字	中斷服務程式名稱											
X0+I ~ X7+I (INT0 ~ INT7) X0-I ~ X7-I (INT0- ~ INT7-)	外部 X0 ~ X7 之中斷服務程式名稱											
HSC0I ~ HSC7I	HSC0 ~ HSC7 之中斷服務程式名稱											
STM0I (1MS) · STM1I (1MS) · STM2I (1MS) · STM3I (1MS) · LTM0I (10MS) · LTM1I (10MS) · LTM2I (10MS) · LTM3I (10MS) ·	PLC 內部 1mS · 10mS · 2 種定時中斷之 服務程式名稱											
HSTAI (ATMRI) · HST0I~HST3I	高速定時中斷服務程式名稱 以 0.1mS 為單位											
除非您所標註的程式確實是上述中斷所對應之服務程式才可用上述之名稱，其他地方不能使用，否則當中斷發生時，PLC 會把您標記之一般程式當作中斷程式執行，而造成錯誤或當機。												
程式範例												

FUN65 LBL	標記 (LABEL)	FUN65 LBL			
下圖例為標記只當作程式註解 (未被呼叫或跳躍至此標記) 之範例，至於標記在跳躍控制之應用請參閱跳躍 (JMP) 指令之說明，標記當副程式名稱則請參閱呼叫 (CALL) 指令之說明。					
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>LBL (PGM1) 程式 1 LBL (PGM2) 程式 2</td></tr></table>		階梯圖	ST		LBL (PGM1) 程式 1 LBL (PGM2) 程式 2
階梯圖	ST				
	LBL (PGM1) 程式 1 LBL (PGM2) 程式 2				

7-7-3 跳躍 (JUMP)

FUN66 P JMP	跳躍 (JUMP)		FUN66 P JMP
指令說明			
階梯圖符號 跳躍控制 — EN — 66P. JMP LBL		LBL：欲跳躍之程式標記	
功能敘述	※僅支援在主程式與副程式中使用		
● 當跳躍控制“EN” =1 或由 0→1 (P 指令) 時，PLC 直接跳到其後標記 LBL 所在位置，繼續往下執行程式。 ● 本指令之應用特別適合在特定狀況發生才需執行某部分程式的應用，平常不執行以節省時間。以及在線圈多重輸出之應用場合，再以輸入控制選擇執行某一段程式之應用。 ● 本指令程式跳躍可往回跳 (即跳回之 LBL 位址比該 JMP 指令所在之位址要小)，但需注意如往回跳致使掃描時間延長超過 Watchdog Timer 所設定之時間，則 PLC 會發生 WDT 中斷，而停止運轉，並發出錯誤訊號。 ● 跳躍指令只限於主程式區跳主程式區，或副程式區跳副程式區，不能跨越主 / 副程式區作跳躍。			

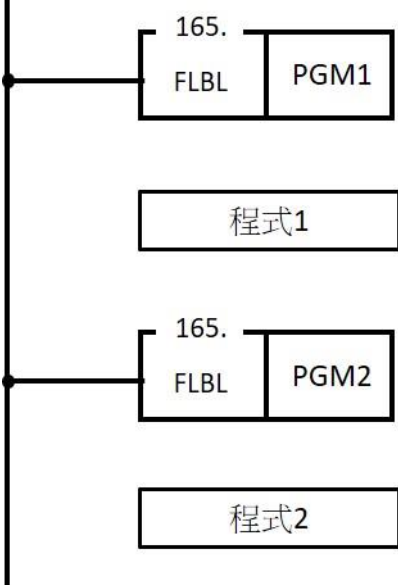
FUN66 P JMP	跳躍 (JUMP)	FUN66 P JMP				
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table>			階梯圖	ST		
階梯圖	ST					
						

上圖中當 X0=1，則程式執行將由 JMP 指令所在處直接跳到 LBL 名稱為 PATHB 之地方往下執行，故程式 A 被跳過，A 中所有指令均不執行，和程式 A 相關之單點或暫存器狀態均保持不變（如同無 A 這段程式）。

7-7-4 功能區塊圖標記 (FUNTION BLOCK LABEL)

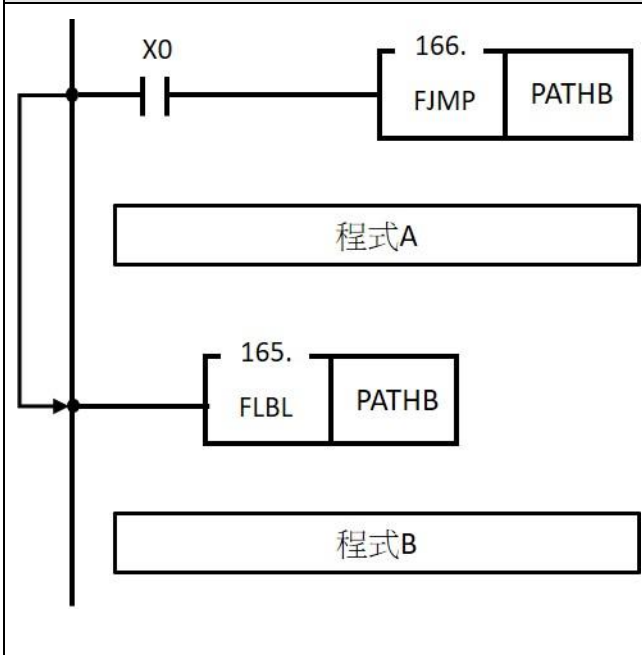
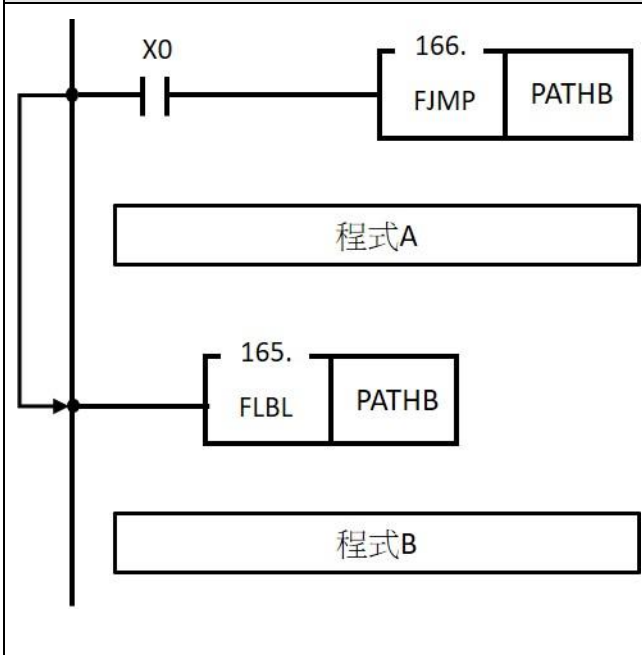
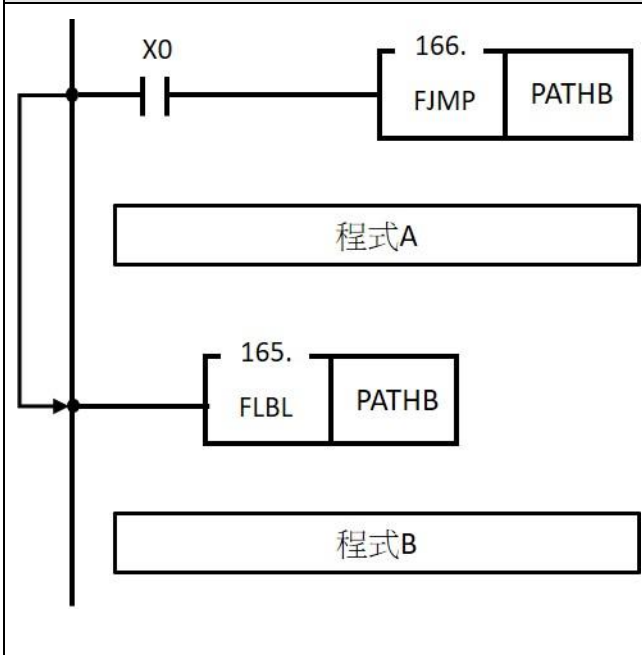
FUN165 FLBL	功能區塊圖標記 (FUNTION BLOCK LABEL)	FUN165 FLBL
指令說明		
階段圖符號 165. FLBL S		S：英文 / 數字 1 ~ 6 字
功能敘述	※僅支援在功能區塊圖(Funtion Block)程式塊中使用	
<		

FUN165 FLBL	功能區塊圖標記 (FUNTION BLOCK LABEL)	FUN165 FLBL												
<table><tr><th>保留字</th><th>中斷服務程式名稱</th></tr><tr><td>X0+I ~ X7+I (INT0 ~ INT7) X0-I ~ X7-I (INT0- ~ INT7-)</td><td>外部 X0 ~ X7 之中斷服務程式名稱</td></tr><tr><td>HSC0I ~ HSC7I</td><td>HSC0 ~ HSC7 之中斷服務程式名稱</td></tr><tr><td>STM0I (1MS) · STM1I (1MS) · STM2I (1MS) · STM3I (1MS) · LTM0I (10MS) · LTM1I (10MS) · LTM2I (10MS) · LTM3I (10MS) ·</td><td>PLC 內部 1mS · 10mS · 2 種定時中斷之 服務程式名稱</td></tr><tr><td>HSTAI (ATMRI) · HST0I~HST3I</td><td>高速定時中斷服務程式名稱 以 0.1mS 為單位</td></tr><tr><td>COCPUI · LHMI · RHM0I · RHM1I · RHM2I · RHM3I · RHM4I · RHM5I</td><td>擴充模組事件中斷</td></tr></table>			保留字	中斷服務程式名稱	X0+I ~ X7+I (INT0 ~ INT7) X0-I ~ X7-I (INT0- ~ INT7-)	外部 X0 ~ X7 之中斷服務程式名稱	HSC0I ~ HSC7I	HSC0 ~ HSC7 之中斷服務程式名稱	STM0I (1MS) · STM1I (1MS) · STM2I (1MS) · STM3I (1MS) · LTM0I (10MS) · LTM1I (10MS) · LTM2I (10MS) · LTM3I (10MS) ·	PLC 內部 1mS · 10mS · 2 種定時中斷之 服務程式名稱	HSTAI (ATMRI) · HST0I~HST3I	高速定時中斷服務程式名稱 以 0.1mS 為單位	COCPUI · LHMI · RHM0I · RHM1I · RHM2I · RHM3I · RHM4I · RHM5I	擴充模組事件中斷
保留字	中斷服務程式名稱													
X0+I ~ X7+I (INT0 ~ INT7) X0-I ~ X7-I (INT0- ~ INT7-)	外部 X0 ~ X7 之中斷服務程式名稱													
HSC0I ~ HSC7I	HSC0 ~ HSC7 之中斷服務程式名稱													
STM0I (1MS) · STM1I (1MS) · STM2I (1MS) · STM3I (1MS) · LTM0I (10MS) · LTM1I (10MS) · LTM2I (10MS) · LTM3I (10MS) ·	PLC 內部 1mS · 10mS · 2 種定時中斷之 服務程式名稱													
HSTAI (ATMRI) · HST0I~HST3I	高速定時中斷服務程式名稱 以 0.1mS 為單位													
COCPUI · LHMI · RHM0I · RHM1I · RHM2I · RHM3I · RHM4I · RHM5I	擴充模組事件中斷													
除非您所標註的程式確實是上述中斷所對應之服務程式才可用上述之名稱，其他地方不能使用，否則當中斷發生時，PLC 會把您標記之一般程式當作中斷程式執行，而造成錯誤或當機。														
程式範例														

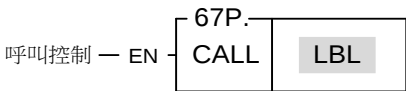
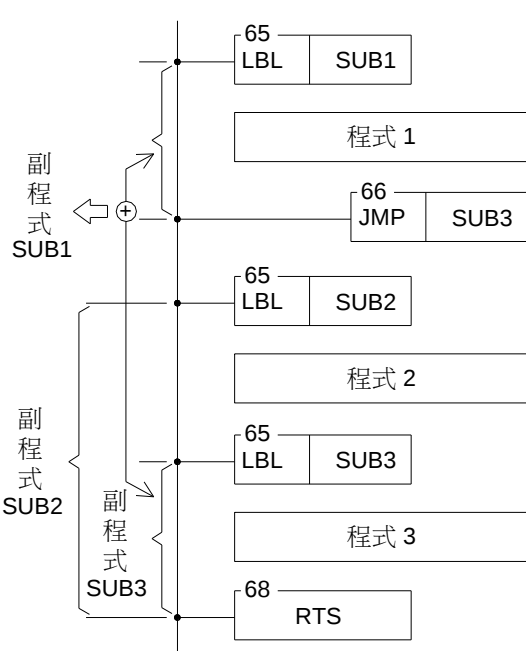
FUN165 FLBL	功能區塊圖標記 (FUNTION BLOCK LABEL)	FUN165 FLBL
下圖例為標記只當作程式註解（未被呼叫或跳躍至此標記）之範例，至於標記在跳躍控制之應用請參閱跳躍（ JMP ）指令之說明，標記當副程式名稱則請參閱呼叫（ CALL ）指令之說明。		
階梯圖 		ST LBL_F ("PGM1") 程式 1 LBL_F ("PGM2") 程式 2

7-7-5 功能區塊圖跳躍 (FUNTION BLOCK JUMP)

FUN166 P FJMP	功能區塊圖跳躍 (FUNTION BLOCK JUMP)	FUN166 P FJMP
指令說明		
階梯圖符號 166. P FJMP FLBL		FLBL：欲跳躍之程式標記
功能敘述	※僅支援在功能區塊圖(Funtion Block)程式塊中使用	
● 當功能區塊圖跳躍控制“EN” =1 或由 0→1 (P 指令) 時，PLC 直接跳到其後標記 FLBL 所在位置，繼續往下執行程式。 ● 本指令之應用特別適合在特定狀況發生才需執行某部分程式的應用，平常不執行以節省時間。以及在線圈多重輸出之應用場合，再以輸入控制選擇執行某一段程式之應用。 ● 本指令程式功能區塊圖跳躍可往回跳（即跳回之 FLBL 位址比該 FJMP 指令所在之位址要小），但需注意如往回跳致使掃描時間延長超過 Watchdog Timer 所設定之時間，則 PLC 會發生 WDT 中斷，而停止運轉，並發出錯誤訊號。 ● 功能區塊圖跳躍指令僅限於同一個功能區塊圖內。		

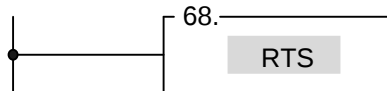
FUN166 P FJMP	功能區塊圖跳躍 (FUNTION BLOCK JUMP)	FUN166 P FJMP				
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN FJMP_166 ("PATHB") END_IF 程式 A LBL ("PATHB") 程式 B</pre></td></tr></table>			階梯圖	ST		<pre>IF X0 THEN FJMP_166 ("PATHB") END_IF 程式 A LBL ("PATHB") 程式 B</pre>
階梯圖	ST					
	<pre>IF X0 THEN FJMP_166 ("PATHB") END_IF 程式 A LBL ("PATHB") 程式 B</pre>					
上圖中當 X0=1，則程式執行將由 JMP 指令所在處直接跳到 FLBL 名為 PATHB 之地方往下執行，故程式 A 被跳過，A 中所有指令均不執行，和程式 A 相關之單點或暫存器狀態均保持不變（如同無 A 這段程式）。						

7-7-6 呼叫 (CALL)

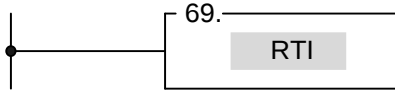
FUN67 P CALL	呼叫 (CALL)	FUN67 P CALL
指令說明		
階梯圖符號  呼叫控制 — EN		LBL：被呼叫之副程式標記名稱
功能敘述	當呼叫控制“EN”=1 或由 0→1 (P 指令) 時，PLC 將呼叫 (執行) 標記名稱與被呼叫之標記名稱相同之副程式，在副程式執行前 PLC 會先將副程式執行完後所需返回之位址 (該 CALL 指令之下一位址) 存入 CPU 內部之堆疊 (STACK) 內，然後再去執行呼叫之副程式，直到遇到副程式中之“副程式返回指令 RTS”後才將先前存入堆疊之返回位址取回，而從返回位址處之指令往下繼續執行程式。	
階梯圖  ST <pre> LBL ("SUB1") 程式 1 JMP_66 ("SUB3") LBL ("SUB2") 程式 2 LBL ("SUB3") 程式 3 RETURN; </pre>		
副程式之最後均要有“副程式返回指令 RTS”，否則將造成執行錯誤或當機，但多個副程式可共用一個 RTS 指令 (此即所謂之多進入點副程式，此種副程式之進入點不同，返回點卻一致)，如右圖例之副程式 SUB1 ~ 3。		

FUN67 P CALL	呼叫 (CALL)	FUN67 P CALL
- 主程式呼叫副程式後副程式尚可呼叫其他副程式（即所謂巢式副程式），最多可達 8 層（中斷 + 呼叫）。 - 中斷服務程式（HSC0I ~ HSC7I、HST0I ~ HST3I、PSO0I ~ PSO3I、X0+I ~ X15+I / INT0 ~ INT15、X0-I ~ X15-I / INT0- ~ INT15-、HSTAI / ATMRI、STM0I ~ STM3I、LTM0I ~ LTM3I、COCPI、LHMI、RHM0I ~ RHM5I）也算是副程式之一種，亦存放在副程式區內，但中斷服務程式之呼叫，是利用硬體觸發信號促使 CPU 去執行對應之中斷服務程式（我們稱之為中斷服務程式之召用）。但使用者設計系統時應該盡量減少中斷服務程式中之工作，因中斷服務程式在系統中擁有比較高的優先權，過長的佔用系統時間會有不良的影響，請參閱 RTI 指令之說明。 - 亦可使用 FCM 實做 Function 或 Function Block，用 FCM 實做之 Function，有如自製的指令，和所有內建指令一樣可加入到階梯圖的網路中。詳情請參閱“使用 FCM 實作 Function 與 Function Block 手冊”。		

7-7-7 副程式返回 (RETURN FROM SUBROUTINE)

FUN68 RTS	副程式返回 (RETURN FROM SUBROUTINE)	FUN68 RTS
指令說明		
階梯圖符號 		
功能敘述		
● 本指令用於表示一副程式之終了，因此只能出現在副程式區內，其輸入側無控制信號，故無法串聯任何元件，本指令單獨就是一完整指令，係直接接到母線上。 ● 當 PLC 執行到本指令時，表示副程式已執行完畢，因此會將先前存入堆疊中之返回位址取回，以便 PLC 回到先前呼叫副程式之下一個指令，繼續往下執行程式。 ● 若在副程式中執行不到 RTS 指令，則程式流程將不再正確，系統堆疊也會被破壞 (DR35361 的 Bit9 系統堆棧錯誤 ON)，並造成系統失控。因此，無論流程如何控制，均需確保所有副程式均會執行到 RTS 指令。 ● RTS 指令之應用請參考 CALL 指令之說明。		

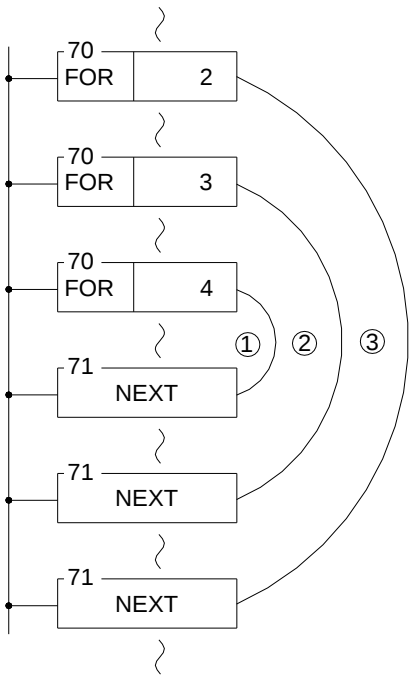
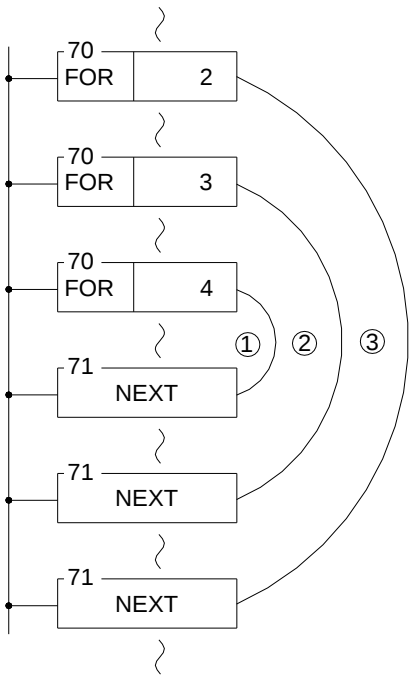
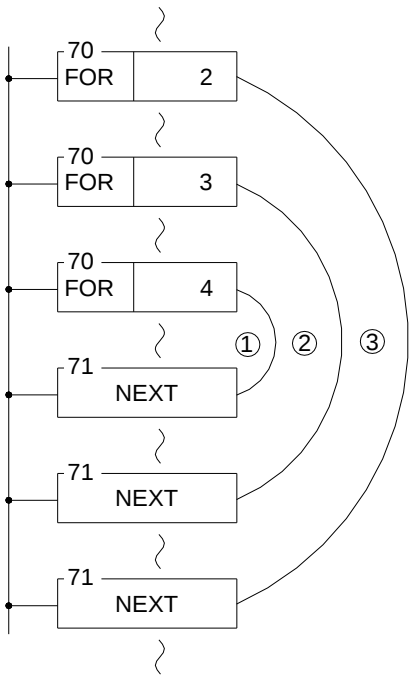
7-7-8 中斷返回 (RETURN FROM INTERRUPT)

FUN69 RTI	中斷返回 (RETURN FROM INTERRUPT)	FUN69 RTI
指令說明	階段圖符號 	
功能敘述	● 本指令之功能和 RTS 類似，只是 RTS 是用於副程式之最後，而 RTI 則用於中斷服務程式之最後，請參閱 RTS 指令之說明。 ● 多個中斷服務程式可共用一個 RTI 指令，其用法和多個副程式可共用一個 RTS 指令一樣，請參考呼叫 (CALL) 指令之說明。 ● 中斷和呼叫之差異只有在呼叫係由使用者自行定義副程式之名稱 (標記 LBL)，然後在主程式或其他副程式中有呼叫指令並指名該副程式之標記，如此當 PLC 執行到該呼叫指令 (CALL)，且其輸入“EN”=1 或由 0→1 (P 指令) 時，PLC 即會去呼叫 (執行) 此副程式。而中斷服務程式之執行則是直接以硬體訊號來中斷 CPU，要 CPU 暫停其他較次要之工作，而來執行該硬體信號所對應之中斷服務程式 (我們稱為中斷服務程式召用)。因此較之呼叫必須掃描到該呼叫指令才會執行之作法，中斷則為更即時 (Real Time) 之作法。此外因中斷服務程式無法指名呼叫，因此我們以特定之“保留字”標記名稱來對應 PLC 所提供之各種中斷 (詳見 FUN65 說明)，例如保留字 X0+I 指定給輸入點 X0 所發生之中斷，只要副程式中有標記為 X0+I 之程式，當輸入點 X0 中斷允許發生 (X0 :)，PLC 就會立即暫停其他較不優先之程式掃描工作，而馬上跳到副程式中標記為 X0+I 的位址去執行程式。	

FUN69 RTI	中斷返回 (RETURN FROM INTERRUPT)	FUN69 RTI
功能敘述		
	● 若中斷發生之時，CPU 正在處理比此中斷優先度更高（如硬體高速計數器中斷）或優先度一樣之中斷（請參考軟體進階使用手冊-第五章之中斷之優先順序章節），則 PLC 會等執行完上述所有中斷服務程式後才會處理此中斷。 ● 若在中斷服務程式中執行不到 RTI 指令，則 PLC 之系統堆疊會被破壞、程式流程錯亂，而有可能引起嚴重當機。因此，無論流程如何控制，均需確保任一中斷服務程式均會執行到 RTI 指令。 ● 關於中斷之詳細說明與使用方法範例請參閱軟體進階使用手冊-第五章之說明。	

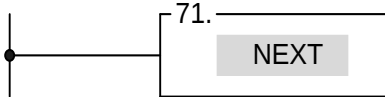
7-7-9 迴圈開始 (FOR)

FUN70 FOR	迴圈開始 (FOR)														FUN70 FOR																																																																								
指令說明																																																																																							
階段圖符號 70. FOR N								N：迴圈執行次數																																																																															
<table><tr><th rowspan="4">運算元 範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><td>WX0</td><td>WY0</td><td>WM0</td><td>WM9120</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td>1</td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td>WX1008</td><td>WY1008</td><td>WM9104</td><td>WM25584</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td>16383</td></tr><tr><td>N</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>																運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	1															WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	16383	N	○	○	○	○	○	○	○	○	○	○	○	○	○	○
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K																																																																									
	WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	1																																																																									
	WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	16383																																																																									
N	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																									
功能敘述																																																																																							
● 本指令無輸入控制，係直接接於母線，不能串接任何元件。 ● FOR 指令和 NEXT 指令所包夾之程式形成一程式迴圈（迴圈程式之開頭為 FOR 之次一個指令，結尾為 NEXT 之前一個指令），當 PLC 執行到 FOR 指令時，首先記下該指令後之 N（迴圈執行次數），然後將此迴圈內的程式從頭到尾連續執行 N 次後，跳離該迴圈，繼續往下（NEXT 之次一指令開始）執行。 ● 迴圈可為巢式結構，即迴圈內包含著迴圈，猶如洋蔥一般，一個迴圈稱為一層最多可達 4 層。FOR 和 NEXT 指令必須成對使用，第一個 FOR 指令和最後一個 NEXT 為巢式迴圈之最外（第一）層。而第二個 FOR 指令和倒數第二個 NEXT 指令為第二層，.....最後一個 FOR 指令和第一個 NEXT 指令形成最內層之迴圈。																																																																																							
程式範例																																																																																							

FUN70 FOR	迴圈開始 (FOR)	FUN70 FOR				
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>FOR 0 TO 2 BY 1 DO ~ FOR 0 TO 3 BY 1 DO ~ FOR 0 TO 4 BY 1 DO ~ END_FOR ~ END_FOR ~ END_FOR</pre></td></tr></table>			階梯圖	ST		<pre>FOR 0 TO 2 BY 1 DO ~ FOR 0 TO 3 BY 1 DO ~ FOR 0 TO 4 BY 1 DO ~ END_FOR ~ END_FOR ~ END_FOR</pre>
階梯圖	ST					
	<pre>FOR 0 TO 2 BY 1 DO ~ FOR 0 TO 3 BY 1 DO ~ FOR 0 TO 4 BY 1 DO ~ END_FOR ~ END_FOR ~ END_FOR</pre>					

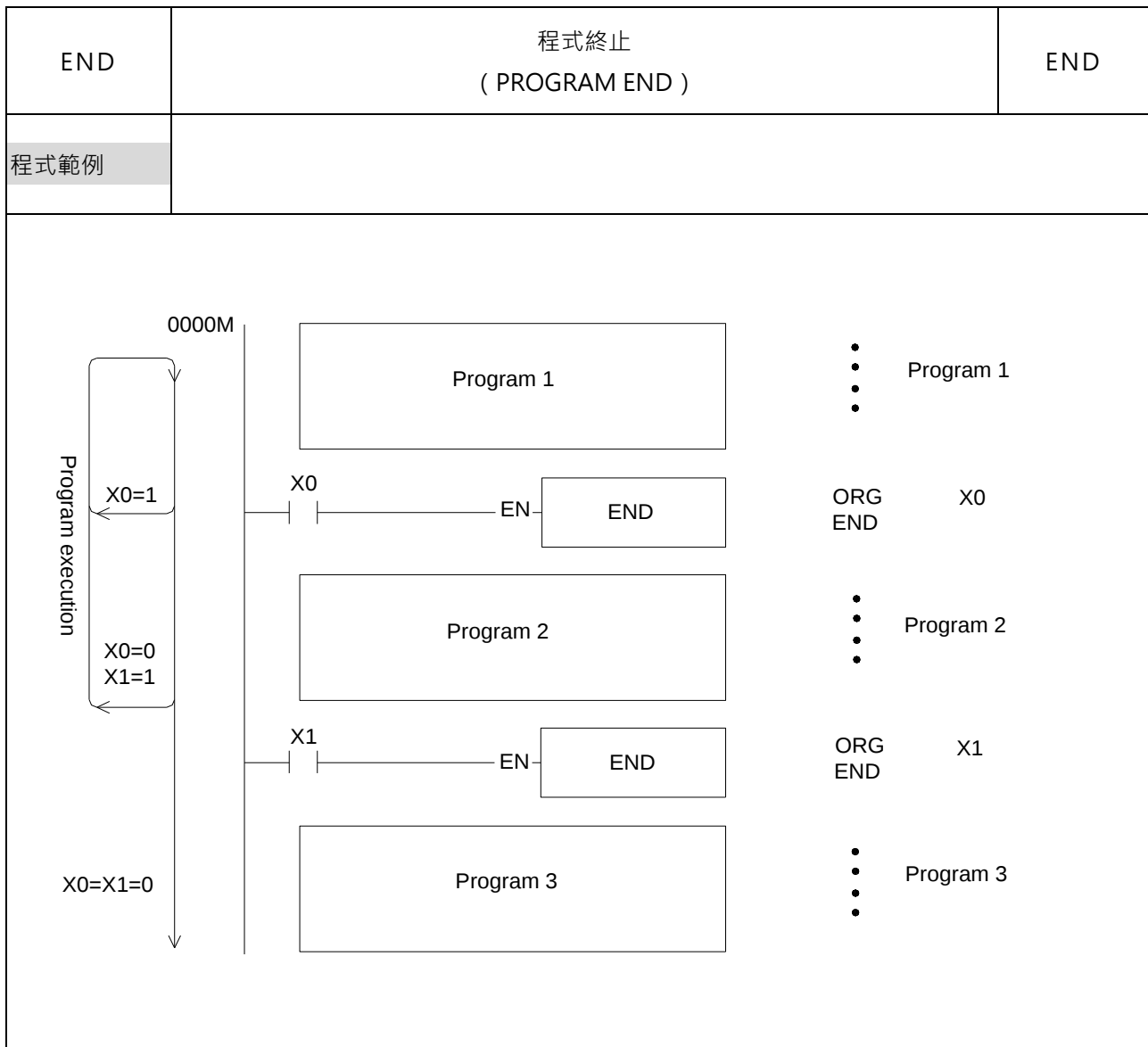
- 上圖例迴圈 將被執行 $4 \times 3 \times 2 = 24$ 次，迴圈 將被執行 $3 \times 2 = 6$ 次，而迴圈 則會執行 2 次。
- 若有 FOR 指令而無 NEXT 指令與之對應，或巢式迴圈之 FOR 和 NEXT 指令未配對使用，或 FOR、NEXT 順序顛倒，均將造成語法錯誤，程式無法執行。
- 迴圈中不可使用 JMP 指令跳出迴圈，否則 PLC 之系統堆疊會被破壞、程式流程錯亂，而有可能引起嚴重當機。
- N 之有效範圍為 1 ~ 16383 次，超出此範圍，PLC 均將之視為 1 次。N 之次數若太大，或迴圈程式太長，可能造成 WATCHDOG 發生，請注意。

7-7-10 迴圈結束(NEXT)

FUN71 NEXT	迴圈結束	FUN71 NEXT
指令說明	階段圖符號 	
功能敘述	● 本指令和 FOR 指令配合形成一個程式迴圈。指令本身無輸入控制，係直接接於母線，不能串接任何元件。 ● 未執行到 FOR 指令，絕不可以執行到 NEXT 指令，否則有可能造成 PLC 當機。 ● 其應用請參閱前頁 FOR 指令之說明。	

程式終止 (PROGRAM END)

END	程式終止 (PROGRAM END)		END
指令說明			
階段圖符號 終止控制 — EN — END		無運算元	
功能敘述			
● 當終止控制“EN” =1 時，本指令動作，立即結束本次之程式掃描，亦即在 END 指令後之程式雖然存在，但卻不會被執行。當“EN” =0 時本指令不執行（當作無此指令），在 END 指令後之程式會繼續被執行。 ● 本指令可在程式中多點放置，而以其輸入（終止控制“EN”）來控制程式執行之終止處，特別有利於除錯或測試。 ● 程式之最後並不一定要有 END 指令，CPU 會自動偵察程式之結束。			



7-8 I / O 指令一 (FUN74 ~ 86)

7-8-1 即時 I / O 更新 (IMMEDIATE I/O REFRESH)

FUN74 P IMDIO	即時 I / O 更新 (IMMEDIATE I/O REFRESH)	FUN74 P IMDIO																			
指令說明																					
階段圖符號 更新控制 — EN — 74P.IMDIO D : N :		D : 欲更新之 I/O 點起頭號碼 N : 欲更新之 I/O 點數																			
<table><tr><th rowspan="2">運算元</th><th>範圍</th><th>X</th><th>Y</th><th>K</th></tr><tr><td></td><td>主機上 之 XN</td><td>主機上 之 YN</td><td>1 36</td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td></td></tr><tr><td>N</td><td></td><td></td><td></td><td>○</td></tr></table>			運算元	範圍	X	Y	K		主機上 之 XN	主機上 之 YN	1 36	D		○	○		N				○
運算元	範圍	X		Y	K																
		主機上 之 XN	主機上 之 YN	1 36																	
D		○	○																		
N				○																	
功能敘述																					
● PLC 系統之輸入 / 輸出信號更新通常在程式執行前先一次抓取全部之輸入信號，然後開始掃描程式，等全部掃描結束才將所有輸出結果一次送到輸出點，如此之輸入動作至輸出反應至少會有一個掃描時間之延遲（最大為 2 個掃描時間）。本指令之作法則為遇到本指令便立即去抓取或送出指令所指定之輸入信號或輸出信號，如此可獲得最即時快速之輸入 / 輸出反應。																					

FUN74 P IMDIO	即時 I / O 更新 (IMMEDIATE I/O REFRESH)	FUN74 P IMDIO									
● 當更新控制“EN”=1 或由 0→1 (P 指令) 時，將 D 所指定之 I (輸入點) 或 O (輸出點) 開始之 N 個輸入點或輸出點 (即 D~D+N-1) 狀態更新。 ● PLC 之即時 I/O 更新之 I/O 點僅限於在主機上之 I/O 點。下表為 MA、ME/MS 主機可容許之即時 I/O 號碼：											
<table border="1"> <tr> <th data-bbox="419 499 815 607">容許號碼 \ 主機 I/O 點數</th><th data-bbox="815 499 991 607">MA</th><th data-bbox="991 499 1161 607">ME/MS</th></tr> <tr> <td data-bbox="419 607 815 658">輸入點</td><td data-bbox="815 607 991 658">X0 ~ X15</td><td data-bbox="991 607 1161 658">X0 ~ X7</td></tr> <tr> <td data-bbox="419 658 815 712">輸出點</td><td data-bbox="815 658 991 712">Y0 ~ Y15</td><td data-bbox="991 658 1161 712">Y0 ~ Y15</td></tr> </table>			容許號碼 \ 主機 I/O 點數	MA	ME/MS	輸入點	X0 ~ X15	X0 ~ X7	輸出點	Y0 ~ Y15	Y0 ~ Y15
容許號碼 \ 主機 I/O 點數	MA	ME/MS									
輸入點	X0 ~ X15	X0 ~ X7									
輸出點	Y0 ~ Y15	Y0 ~ Y15									
● 如果程式中之即時 I/O 點之範圍超出主機之輸入點或輸出點號碼 (例如程式中 D=X7，N=9，則表示要即時抓取 X7~X15 等 9 個輸入點信號，而假設主機為 ME/MS 機種，其輸入點最大為 X7，明顯地 X15 已超出該主機之輸入點號碼) 則 PLC 將無法運轉 (STOP，ERR 燈亮) ● 本指令執行時，雖然 PLC 會立即去抓取或送出即時輸入 / 輸出信號，但在輸入點上之硬體或軟體積分之延遲或輸出點之動作延遲 (如繼電器或電晶體等輸出元件之動作反應時間) 仍然存在，請特別注意。											

7-9 PID 控制 (FUN38 、 FUN99)

7-9-1 溫控便利指令(TPCTL2)

FUN99 ^P TPCTL2	PID 溫控便利指令 2 (PID TEMPERATURE CONTROL INSTRUCTION)		FUN99 ^P TPCTL2																																																																						
指令說明																																																																									
執行控制 —EN— 更新參數 —UPD— 自動輸出 /手動輸出 —A/M— 加熱冷卻 —H/C— 99P.TPCTL2 ID: CH: SR: PR: OR: WR: ERR — 溫控錯誤 ALM— 溫控警報 ID：作輸入訊號的擴充模組的編號 CH: 作輸入訊號的通道編號 SR：程控設定起始暫存器 PR：增益設定起始暫存器 OR :輸出起始暫存器 WR: 工作起始暫存器																																																																									
<table><tr><th rowspan="2">運算元</th><th rowspan="2">範圍</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><th>R0 R34767</th><th>R34768 R34895</th><th>R35024 R35151</th><th>R35280 R43223</th><th>R43224 R47319</th><th>D0 D7999</th><th></th></tr><tr><td>ID</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>0 ~ 127</td></tr><tr><td>CH</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~63</td></tr><tr><td>SR</td><td>○</td><td></td><td></td><td></td><td></td><td>○</td><td>○</td><td></td></tr><tr><td>PR</td><td>○</td><td></td><td></td><td></td><td></td><td>○*</td><td>○</td><td></td></tr><tr><td>OR</td><td>○</td><td></td><td></td><td></td><td></td><td>○*</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td></td><td></td><td></td><td></td><td>○*</td><td>○</td><td></td></tr></table>				運算元	範圍	HR	IR	OR	SR	ROR	DR	K	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		ID								0 ~ 127	CH	○	○	○	○	○	○	○	0~63	SR	○					○	○		PR	○					○*	○		OR	○					○*	○		WR	○					○*	○	
運算元	範圍	HR	IR			OR	SR	ROR	DR	K																																																															
		R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999																																																																		
ID								0 ~ 127																																																																	
CH	○	○	○	○	○	○	○	0~63																																																																	
SR	○					○	○																																																																		
PR	○					○*	○																																																																		
OR	○					○*	○																																																																		
WR	○					○*	○																																																																		

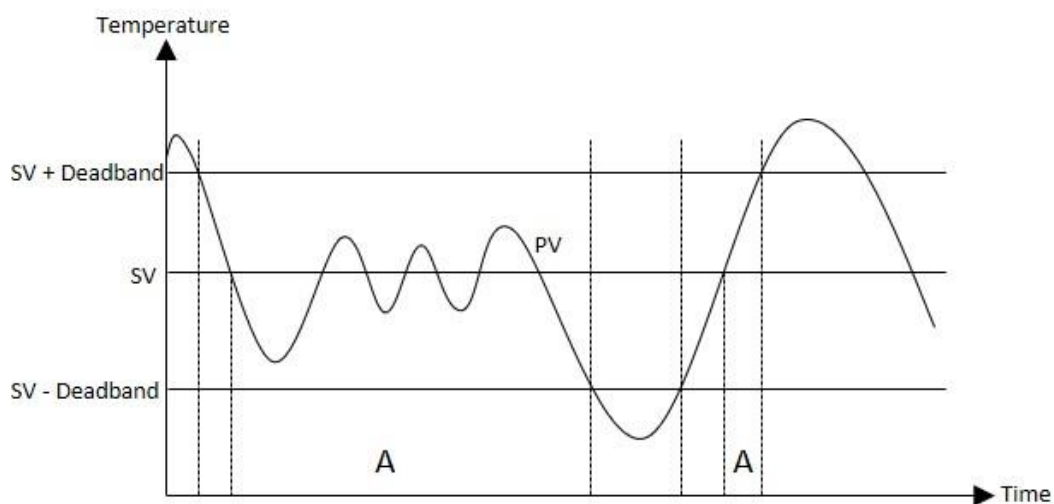
FUN99 ^P TPCTL2	PID 溫控便利指令 2 (PID TEMPERATURE CONTROL INSTRUCTION)	FUN99 ^P TPCTL2
功能敘述	● PID 溫控 (FUN99) 係利用溫度模組配合溫度規劃表格將外界目前之溫度值量測進來當作程控變數 (Process Variable · 簡稱 PV) · 並將使用者所設定之溫度設定值 (Set Point · 簡稱 SP) 與程控變數經由軟體 PID 數學式運算後 · 得到適宜之輸出控制值以控制溫度在使用者所期望之溫度範圍內。 ● 將 PID 運算後之數值結果轉換為時間比例 ON/OFF (PWM) 輸出 · 經由電晶體式接點輸出控制 SSR 所串接之加熱或冷卻迴路 · 即可得到相當精準之控制結果。亦可將 PID 運算後之數值結果經由 D/A 類比輸出模組 · 控制 SCR 導通角度或比例閥以作溫度精準控制。 ● EN : ON 時執行溫度控制 · OFF 時停止 ● UPD : ON 時將參數更新至模組指定通道 ● A/M : PID manual mode(PID 手動模式) · 啟用的話輸出量將為手動控制模式 · 會將 MOUT 值自動拷貝至 MV · 而非使用 PID 運算結果做為輸出。 ● H/C : 執行加熱或冷卻控制	

FUN99 ^P TPCTL2	PID 溫控便利指令 2 (PID TEMPERATURE CONTROL INSTRUCTION)	FUN99 ^P TPCTL2
PID control ● PID 控制係由模組獨立運算，不因多台模組同時進行 PID 而增加 PLC 掃描周期。 ● 每個通道皆可以執行各自的 PID 運算，溫度控制模式需要設為 PID 控制，利用比例項(P)、積分動作(I)和微分動作(D)可以更有效地進行溫度控制，可以依照使用需求進行 P、PI、PD、PID 控制。 ● Proportional(比例項)，輸出量(MV)的大小會隨著測量值(PV)和設定值(SV)間的誤差(E)成一個輸出的比例，比例項設定小時波動較大，反之波動則較小。 ● Integral time(積分時間)，根據測量值(PV)和設定值(SV)之間的誤差(E)量來增減輸出量，以減少 P 動作最後產生的穩態誤差，積分時間設定越小則波動較大上升較快，反之則較小較慢，範圍為 0~3600s，若積分時間為 0 則不進行積分控制。 ● Derivative time(微分時間)，根據測量值(PV)和設定值(SV)之間的誤差(E)變化率來增減輸出量，即使因為雜訊影響造成急遽的變化，或是控制上過衝，透過微分動作可以在短時間內恢復到穩定狀態，微分時間設定越小則波動較小反應較慢，反之則較大較快，範圍為 0~3600s，若微分時間為 0 則不進行微分控制。		

FUN99^P
TPCTL2PID 溫控便利指令 2
(PID TEMPERATURE CONTROL INSTRUCTION)FUN99^P
TPCTL2

SR 參數	Word size	說明
TS	1	Time cycle 大小，單位為 0.1s(0.1s~30.0s)
SV	2	Set value，單位為 0.1 度
DEAD BAND	1	到達 SV 附近的不作用區，範圍為 0.1%~10.0%
DOUT	1	輸出點數
PERIOD	1	PWM 週期，單位為 1s
Out mode	1	0, PWM 輸出 1, else

- PID_Deadband(PID 死區)，設定範圍為 0~10.0%(輸入範圍)，在 PID 控制中此區為一個偏差量(E)不作用區，當 temperature process value(PV，溫度程控值)一開始進入死區後仍會正常 PID 運算直到穿越過 set value(SV，設定值)，此時的 E 才會用 0 代入算式，直到穿越此區後恢復正常直行 PID 運算，例如圖中 A 區的 E 皆視為 0。



FUN99P
TPCTL2PID 溫控便利指令 2
(PID TEMPERATURE CONTROL INSTRUCTION)FUN99P
TPCTL2

PR	Word size	說明
Kp	2(floating point)	比例項，實數
Ti	1	積分時間，0~3600s
Td	1	微分時間，0~3600s
Bias	2(floating point)	輸出偏差值，實數
High output limit	2(floating point)	輸出上限
Low output limit	2(floating point)	輸出下限
PID Method	1	0: 標準 PID 1: 最小超越法
AT	1	AT 是否啟用
MAUTO	1	MOUT 值是否隨著 MV 改變

- Kp、Ti、Td：PID 參數，可指定或開啟 AT 自動產生後調整使用。
- Bias：輸出偏差值，使用者可以用來對輸出值做一個增減，但還是會受到輸出範圍的設定的限定。
- High/Low output limit：輸出範圍的限定，設定 PID 輸出上下限值，若輸出下限大於等於輸出上限則會發出錯誤警報。
- PID Method：選擇適合的 PID 演算法
- AT：是否啟用 Autotuning 取得 PID 調控參數
- MAUTO：將 MV 值拷貝至 MOUT

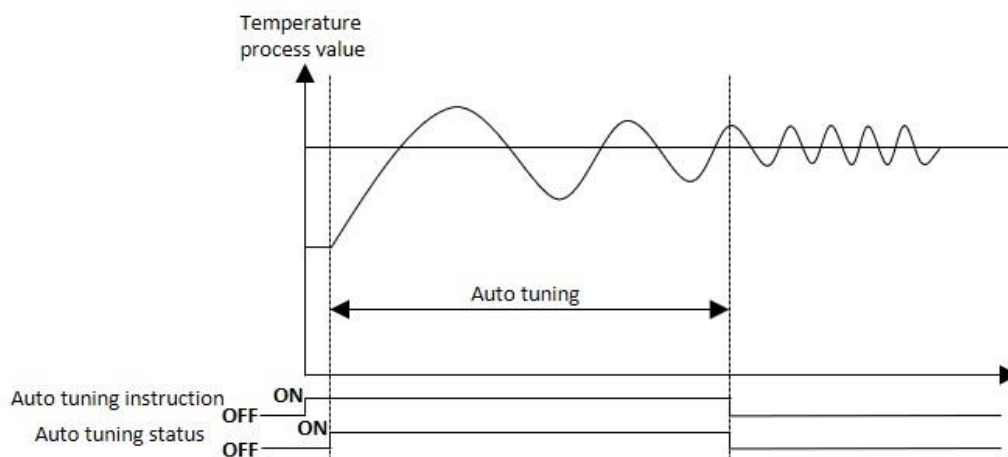
OR	Word size	說明
MV	2(floating point)	輸出值回報
MOUT	2(floating point)	MV 手動輸出值設定

FUN99P
TPCTL2PID 溫控便利指令 2
(PID TEMPERATURE CONTROL INSTRUCTION)FUN99P
TPCTL2

WR	Word size	說明
PID 運行狀態	1	=0, Idle =1, Working =2, Error =3, AT now
AT 工作狀態	1	=0, Idle =1, Running =2, Error =3, Finish =4, Time out
PV	2	程控值回報

Auto tuning

- 此功能可以依據控制系統環境自動計算出適合的比例項(P)、積分時間(I)和微分時間(D)的 PID 參數，需選擇 PID 控制模式下並開始執行溫度控制後才能使用，暫時透過 ON/OFF 控制後得到的數個波形進行計算以獲得最佳的 PID 參數，結束後將參數自動寫入到 PID 各自的記憶體中，並轉為 PID 控制模式進行溫度控制。



FUN99 ^P TPCTL2	PID 溫控便利指令 2 (PID TEMPERATURE CONTROL INSTRUCTION)	FUN99 ^P TPCTL2
	● 在 auto tuning 的期間會參考輸出上限和輸出下限做為輸出的參考依據，輸出週期的設定需要不為 0 才能進行 auto tuning。 ● SV 設定若超出溫度範圍值時，auto tuning 不會被執行。 ● 若經過 2 小時 auto tuning 尚未完成的話，則發出 auto tuning 的逾時錯誤。 ● 被設為關閉的通道無法執行 auto tuning 功能。 ● 若在 auto tuning 期間更改 SV、死區、TC 模組校正、輸出上限、輸出週期、控制模式和關閉通道的設定值，則 auto tuning 將停止並發將錯誤繼電器 ON。 ● 執行方式: 透過溫控指令 ● 結束方式: Auto tuning 完成回報	

7-9-2 泛用 PID 運算指令(PID)

FUN30 PID	泛用 PID 運算指令 (功能簡述) (PID)		FUN30 PID																																		
指令說明																																					
執行控制 — EN 平順轉移 — BUM 運作方向 — D/R 30. PID Ts : SR : OR : PR : WR : ERR — 設定錯誤 HA — 上限警告 LA — 下限警告		Ts : PID 運算間隔時間 SR : 程控設定值起始暫存器號碼， 共佔用 8 個暫存器 OR : PID 輸出暫存器號碼 PR : 參數設定值起始暫存器號碼， 共佔用 7 個暫存器 WR: 本指令所需使用之工作暫存器起 始號碼，共佔用 5 個暫存器，其 它地方不可重覆使用。																																			
<table><tr><th rowspan="2">運算元 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><th>R0 R34767</th><th>R43224 R47319</th><th>D0 D7999</th><th></th></tr><tr><td>Ts</td><td>○</td><td>○</td><td>○</td><td>0 ~ 3000</td></tr><tr><td>SR</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>OR</td><td>○</td><td>○*</td><td>○</td><td></td></tr><tr><td>PR</td><td>○</td><td>○*</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>				運算元 範圍	HR	ROR	DR	K	R0 R34767	R43224 R47319	D0 D7999		Ts	○	○	○	0 ~ 3000	SR	○	○	○		OR	○	○*	○		PR	○	○*	○		WR	○	○*	○	
運算元 範圍	HR	ROR	DR		K																																
	R0 R34767	R43224 R47319	D0 D7999																																		
Ts	○	○	○	0 ~ 3000																																	
SR	○	○	○																																		
OR	○	○*	○																																		
PR	○	○*	○																																		
WR	○	○*	○																																		

FUN30 PID	泛用 PID 運算指令 (功能簡述) (PID)	FUN30 PID
功能敘述	● 泛用 PID 指令(FUN30) 係將目前所量測之外界類比輸入值當作程控變數 (Process Variable , 簡稱 PV) , 將使用者所設定之設定值 (Set Point , 簡稱 SP) 與程控變數經由軟體 PID 數學式運算後 , 得到適宜之輸出控制值經由 D/A 類比輸出模組或再處理經由其它界面以控制受控程序在使用者所期望之設定範圍內。 ● 數位化 PID 運算式如下 : $Mn = [(D4005/Pb) \times En] + \sum_{n=0}^{n} [(D4005/Pb) \times Ti \times Ts \times En] - [(D4005/Pb) \times Td \times (PVn - PVn-1)/Ts] + Bias$ Mn = : “n” 時之控制輸出量 D4005 : 增益常數 , 內定值為 1000 ; 可設定範圍為 1 ~ 5000 Pb : 比例帶 (範圍 : 2 ~ 5000 , 單位為 0.1% ; Kc (增益) = 1000 / Pb) Ti : 積分時間常數 (範圍 : 0 ~ 9999 , 相當於 0.00 ~ 99.99 Repeats/Minute) Td : 微分時間常數 (範圍 : 0 ~ 9999 , 相當於 0.00 ~ 99.99 Minutes) PVn : “n” 時之程控變數值 PV n - 1 : “n” 之上一次之程控變數值 En : “n” 時之誤差 = 設定值 (SP) - “n” 時之程控變數值 (PVn) Ts : PID 運算之間隔時間 (範圍 : 1 ~ 3000 , 單位 : 0.01S) Bias : 偏置輸出量 (範圍 : 0 ~ 16380) ● 詳細之功能說明與程式範例 , 請參考功能與應用手冊第 13 章 “PID 控制” 之敘述。	

7-9-3 泛用 PID 控制指令(PID2)

FUN38 PID2	PID 2					FUN38 PID2																																																	
指令說明																																																							
階段圖符號 執行控制 — EN 參數更新 — UPD 手動輸出/PID輸出 — A/M 正向輸出/反向輸出 — D/R 38P.PID2 ID CH SR OR PR WR ERR — 通訊錯誤				ID ：作輸入訊號的擴充模組的編號 CH ：作輸入訊號的通道編號 SR ：程控設定值起始暫存器號碼， 共佔用 8 個暫存器 OR ：PID 輸出暫存器號碼 PR ：參數設定值起始暫存器號碼， 共佔用 7 個暫存器 WR ：本指令所需使用之工作暫存器起始號碼，共佔 用 5 個暫存器，其它地方不可重覆使用。																																																			
<table><tr><td rowspan="8">範圍 運算元</td><td>HR</td><td>IR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td></tr><tr><td>R0 R34767</td><td>R34768 R34895</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td></tr><tr><td>ID</td><td></td><td></td><td></td><td></td><td>0 ~ 127</td></tr><tr><td>CH</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0 ~ 63</td></tr><tr><td>SR</td><td>○</td><td></td><td>○</td><td>○</td><td></td></tr><tr><td>OR</td><td>○</td><td></td><td>○*</td><td>○</td><td></td></tr><tr><td>PR</td><td>○</td><td></td><td>○</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td></td><td>○</td><td>○</td><td></td></tr></table>							範圍 運算元	HR	IR	SR	ROR	DR	K	R0 R34767	R34768 R34895	R35280 R43223	R43224 R47319	D0 D7999		ID					0 ~ 127	CH	○	○	○	○	0 ~ 63	SR	○		○	○		OR	○		○*	○		PR	○		○	○		WR	○		○	○	
範圍 運算元	HR	IR	SR	ROR	DR	K																																																	
	R0 R34767	R34768 R34895	R35280 R43223	R43224 R47319	D0 D7999																																																		
	ID					0 ~ 127																																																	
	CH	○	○	○	○	0 ~ 63																																																	
	SR	○		○	○																																																		
	OR	○		○*	○																																																		
	PR	○		○	○																																																		
	WR	○		○	○																																																		
功能敘述																																																							

FUN38 PID2	PID 2	FUN38 PID2
	● PID2 的使用與 PID 的不同之處在於，PID2 將運算分佈在支援 PID 功能的模組上，而不是使用 CPU 單元的運算能力。 ● 泛用 PID2 指令(FUN38)係將目前所量測之外界類比輸入值當作程控變數 (Process Variable，簡稱 PV)，將使用者所設定之設定值 (Set Point，簡稱 SP) 與程控變數經由軟體 PID 數學式運算後，得到適宜之輸出控制值經由 D/A 類比輸出模組或再處理經由其它介面以控制受控程序在使用者所期望之設定範圍內。 ● 數位化 PID 運算式如下： $Mn = [Kp \times En] \sum_0^n [Kp \times Ti \times Ts \times En] - [Kp \times Tdx(PVn - PVn-1)/Ts] + Bias$ $Mn = \quad : \text{“n” 時之控制輸出量}$ $Kp : \text{比例項實數 (範圍：} \pm(1.8 \times 10^{-38} \sim 3.4 \times 10^{38}))$ $Ti : \text{積分時間常數 (範圍：} 0 \sim 3600 \cdot \text{相當於 } 0 \sim 3600 \text{ Repeats/Seconds)}$ $Td : \text{微分時間常數 (範圍：} 0 \sim 3600 \cdot \text{相當於 } 0 \sim 3600 \text{ Seconds)}$ $PVn : \text{“n” 時之程控變數值}$ $PVn-1 : \text{“n” 之上一次之程控變數值}$ $En : \text{“n” 時之誤差} = \text{設定值 (SP)} - \text{“n” 時之程控變數值 (PVn)}$ $Ts : \text{PID 運算之間隔時間 (範圍：} 1 \sim 300 \cdot \text{單位：} 0.1S)$ $Bias : \text{偏置輸出量 (範圍：} \pm(1.8 \times 10^{-38} \sim 3.4 \times 10^{38}))$	

FUN38 PID2	PID 2	FUN38 PID2
	● 當控制選擇“A/M”=0 時，代表手動控制模式，將不會使用 PID 運算結果，手動輸出值 MOUT 將自動拷貝至 MV。 ● 當控制選擇“A/M”=1 時，代表自動控制模式，MV 值由 PID 運算而來，若 MAUTO=1 則 MV 值會自動拷貝至 MOUT。 ● 當控制選擇“A/M”=1 且運作方向“D/R”=1 時，程序控制為順向 PID 控制；亦即誤差 (SP-PVn) 為正時，PID 運算結果之控制輸出量越大；誤差為負時，PID 運算結果之控制輸出量越小。 ● 當控制選擇“A/M”=1 且運作方向“D/R”=0 時，程序控制為反向 PID 控制；亦即誤差 (SP-PVn) 為正時，PID 運算結果之控制輸出量越小；誤差為負時，PID 運算結果之控制輸出量越大。 ● 當程控設定值或參數設定值錯誤時，PID 指令不會執行，設定錯誤指示“ERR”=1。 ● 如需更新參數的話，更新相關暫存器內容後將 UPD OFF->ON 即可更新參數。	

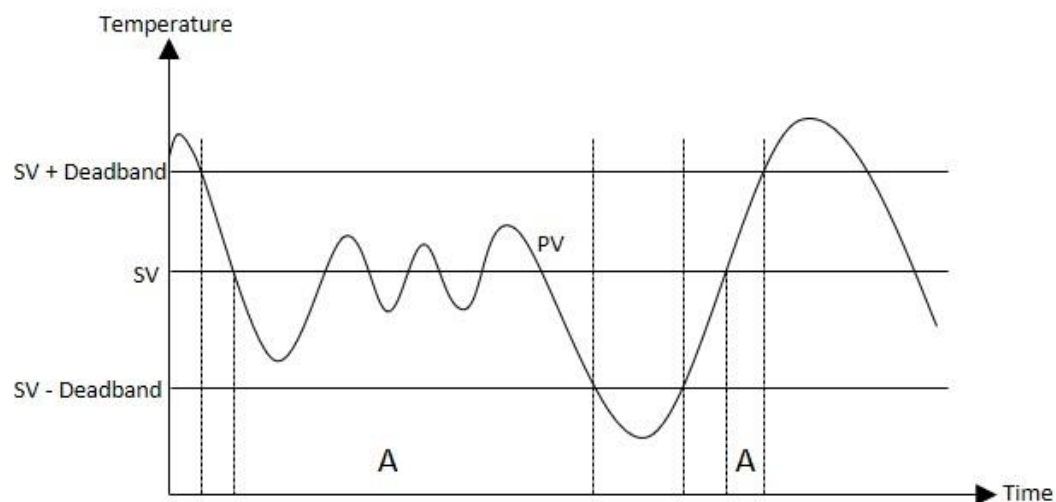
FUN38
PID2

PID 2

FUN38
PID2

SR 參數	Word size	說明
TS	1	Time cycle 大小，單位為 0.1s(0.1s~30.0s)
SV	2	Set value，單位為 0.1 度
DEAD BAND	1	到達 SV 附近的不作用區，範圍為 0.1%~10.0%

- PID_Deadband(PID 死區)·設定範圍為 0~10.0%(輸入範圍)·在 PID 控制中此區為一個偏差量(E)不作用區·當 temperature process value(PV·溫度程控值)一開始進入死區後仍會正常 PID 運算直到穿越過 set value(SV·設定值)·此時的 E 才會用 0 代入算式·直到穿越此區後恢復正常直行 PID 運算·例如圖中 A 區的 E 皆視為 0。



FUN38 PID2	PID 2	FUN38 PID2																																	
<table border="1"> <thead> <tr> <th>PR</th><th>Word size</th><th>說明</th></tr> </thead> <tbody> <tr> <td>Kp</td><td>2(floating point)</td><td>比例項，實數</td></tr> <tr> <td>Ti</td><td>1</td><td>積分時間，0~3600s</td></tr> <tr> <td>Td</td><td>1</td><td>微分時間，0~3600s</td></tr> <tr> <td>Bias</td><td>2(floating point)</td><td>輸出偏差值，實數</td></tr> <tr> <td>High output limit</td><td>2(floating point)</td><td>輸出上限</td></tr> <tr> <td>Low output limit</td><td>2(floating point)</td><td>輸出下限</td></tr> <tr> <td>PID Method</td><td>1</td><td>0: 標準 PID 1: 最小超越法</td></tr> <tr> <td>BUM</td><td>2</td><td>平順轉移啟用(1 啟用,0 關閉),為手動轉自動控制模式時能平順的轉換的功能</td></tr> <tr> <td>AT</td><td>1</td><td>AT 是否啟用</td></tr> <tr> <td>MAUTO</td><td>1</td><td>MOUT 值是否隨著 MV 改變</td></tr> </tbody> </table> ● Kp、Ti、Td：PID 參數，可指定或開啟 AT 自動產生後調整使用。 ● Bias：輸出偏差值，使用者可以用來對輸出值做一個增減，但還是會受到輸出範圍的設定的限定。 ● High/Low output limit：輸出範圍的限定，設定 PID 輸出上下限值，若輸出下限大於等於輸出上限則會發出錯誤警報。 ● PID Method：選擇適合的 PID 演算法 ● BUM：啟用平順轉移，用於從手動控制切換至自動控制模式時，使控制輸出平順銜接 ● AT：是否啟用 Autotuning 取得 PID 調控參數 ● MAUTO：將 MV 值拷貝至 MOUT			PR	Word size	說明	Kp	2(floating point)	比例項，實數	Ti	1	積分時間，0~3600s	Td	1	微分時間，0~3600s	Bias	2(floating point)	輸出偏差值，實數	High output limit	2(floating point)	輸出上限	Low output limit	2(floating point)	輸出下限	PID Method	1	0: 標準 PID 1: 最小超越法	BUM	2	平順轉移啟用(1 啟用,0 關閉),為手動轉自動控制模式時能平順的轉換的功能	AT	1	AT 是否啟用	MAUTO	1	MOUT 值是否隨著 MV 改變
PR	Word size	說明																																	
Kp	2(floating point)	比例項，實數																																	
Ti	1	積分時間，0~3600s																																	
Td	1	微分時間，0~3600s																																	
Bias	2(floating point)	輸出偏差值，實數																																	
High output limit	2(floating point)	輸出上限																																	
Low output limit	2(floating point)	輸出下限																																	
PID Method	1	0: 標準 PID 1: 最小超越法																																	
BUM	2	平順轉移啟用(1 啟用,0 關閉),為手動轉自動控制模式時能平順的轉換的功能																																	
AT	1	AT 是否啟用																																	
MAUTO	1	MOUT 值是否隨著 MV 改變																																	

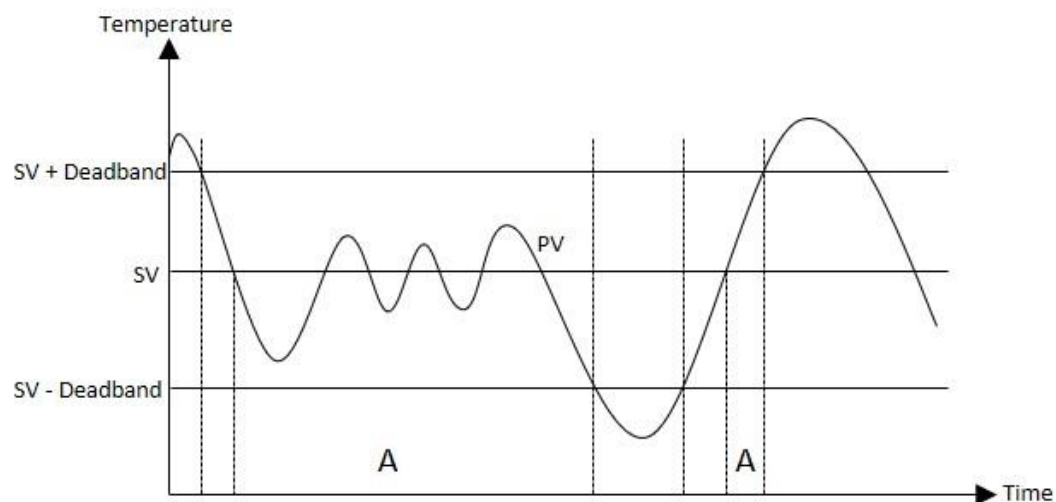
FUN38
PID2

PID 2

FUN38
PID2

SR 參數	Word size	說明
TS	1	Time cycle 大小，單位為 0.1s(0.1s~30.0s)
SV	2	Set value，單位為 0.1 度
DEAD BAND	1	到達 SV 附近的不作用區，範圍為 0.1%~10.0%

- PID_Deadband(PID 死區)，設定範圍為 0~10.0%(輸入範圍)，在 PID 控制中此區為一個偏差量(E)不作用區，當 temperature process value(PV，溫度程控值)一開始進入死區後仍會正常 PID 運算直到穿越過 set value(SV，設定值)，此時的 E 才會用 0 代入算式，直到穿越此區後恢復正常直行 PID 運算，例如圖中 A 區的 E 皆視為 0。



FUN38 PID2	PID 2	FUN38 PID2																					
<table border="1"> <thead> <tr> <th>WR</th><th>Word size</th><th>說明</th></tr> </thead> <tbody> <tr> <td>PID 運行狀態</td><td>1</td><td>=0, Idle =1, Working =2, Error =3, AT now</td></tr> <tr> <td>AT 工作狀態</td><td>1</td><td>=0, Idle =1, Running =2, Error =3, Finish =4, Time out</td></tr> <tr> <td>PV</td><td>2</td><td>程控值回報</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th>OR</th><th>Word size</th><th>說明</th></tr> </thead> <tbody> <tr> <td>MV</td><td>2(floating point)</td><td>輸出值回報</td></tr> <tr> <td>MOUT</td><td>2(floating point)</td><td>MV 手動輸出值設定</td></tr> </tbody> </table>			WR	Word size	說明	PID 運行狀態	1	=0, Idle =1, Working =2, Error =3, AT now	AT 工作狀態	1	=0, Idle =1, Running =2, Error =3, Finish =4, Time out	PV	2	程控值回報	OR	Word size	說明	MV	2(floating point)	輸出值回報	MOUT	2(floating point)	MV 手動輸出值設定
WR	Word size	說明																					
PID 運行狀態	1	=0, Idle =1, Working =2, Error =3, AT now																					
AT 工作狀態	1	=0, Idle =1, Running =2, Error =3, Finish =4, Time out																					
PV	2	程控值回報																					
OR	Word size	說明																					
MV	2(floating point)	輸出值回報																					
MOUT	2(floating point)	MV 手動輸出值設定																					

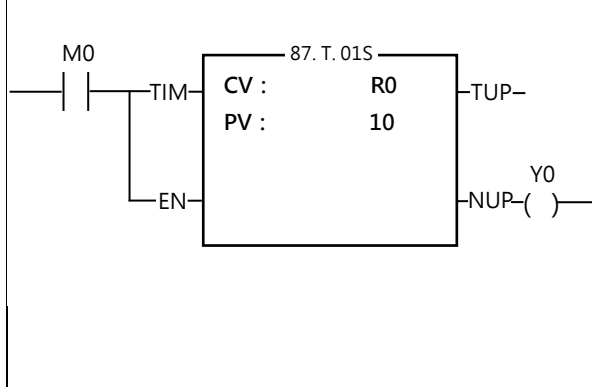
7-10 積算型計時器指令 (FUN87 ~ 89)

7-10-1 積算型計時器 (10 毫秒，100 毫秒，1 秒) (ACCUMULATIVE TIMER)

FUN87 T.01S FUN88 T.1S FUN89 T1S	積算型計時器 (0.01 秒 · 0.1 秒 · 1 秒) (ACCUMULATIVE TIMER)												FUN87 T.01S FUN88 T.1S FUN89 T1S		
指令說明															
階段圖符號 89.T1S 88.T.1S 87.T.01S 計時控制 — TIM — CV : — TUP — 計時到 致能控制 — EN — PV : — NUP — 計時未到								CV：存放計時時間 (即現在值) 之暫存器號碼 PV：計時器之設定或存放設定值之暫存器號碼							
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 位元 32 位元 0 0 或 32767 2147483647	
	CV	○	○	○	○	○	○	○		○	○*	○*	○		
	PV	○	○	○	○	○	○	○	○	○	○	○	○	○	
功能敘述															

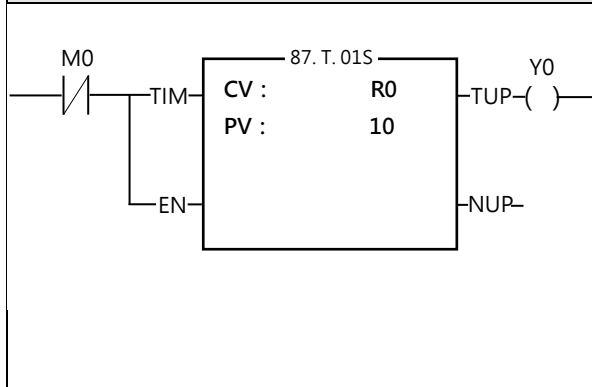
FUN87 T.01S FUN88 T.1S FUN89 T1S	積算型計時器 (0.01 秒 · 0.1 秒 · 1 秒) (ACCUMULATIVE TIMER)	FUN87 T.01S FUN88 T.1S FUN89 T1S				
● 本指令之工作原理和一般計時器 (T0 ~ T1023) 相同，只是一般計時器只有一個計時控制“EN”輸入，在其輸入為 1 時計時，為 0 時則清除，每次輸入變動都會重新計時，無法累計。而本指令之計時需在致能控制“EN”=1 之條件下才允許，此時其計時控制“TIM”為 1 時和一般計時器一樣，但為 0 時，則不清除而保持現值。若需清除則使致能控制“EN”變為 0 即可，若您不清除之，等計時控制“TIM”再度為 1 時，計時器將由上次暫停時之計時值繼續累加。此外本指令尚有計時到“TUP”（計時到時為 1，平時為 0），及計時未到“NUP”（平時為 1，計時到為 0）兩個輸出，使用者可利用輸入和輸出組合出各種不同功能需求之計時器。						
程式範例 1	ON DELAY ENERGIZING (ON 延遲供電) 計時器					
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>ACTimer_10MS(TIM:= M0, EN:= M0, CV:= R0, PV:= 10, TUP=> Y0);</td></tr></table>			階梯圖	ST		ACTimer_10MS(TIM:= M0, EN:= M0, CV:= R0, PV:= 10, TUP=> Y0);
階梯圖	ST					
	ACTimer_10MS(TIM:= M0, EN:= M0, CV:= R0, PV:= 10, TUP=> Y0);					
● 係指該計時器輸出 (本例為 Y0) 平常不供電，而在該計時器輸入控制 (本例為 X0) 動作 (ON) 後延遲 100 毫秒後，輸出 Y0 才供應電能 (ON)。						
程式範例 2	ON DELAY DE-ENERGIZING (ON 延遲斷電) 計時器					

FUN87 T.01S FUN88 T.1S FUN89 T1S	積算型計時器 (0.01 秒 · 0.1 秒 · 1 秒) (ACCUMULATIVE TIMER)	FUN87 T.01S FUN88 T.1S FUN89 T1S
--	---	--

階梯圖	ST
	<pre>ACTimer_10MS(TIM:= M0, EN:= M0, CV:= R0, PV:= 10, NUP=> Y0);</pre>

- 係指該計時器輸出 Y0 平常就在供電狀態下，而在該計時器之輸入控制 X0 ON 後延遲 100 毫秒後，輸出才斷電 (OFF)。

程式範例 3	OFF DELAY ENERGIZING (OFF 延遲供電) 計時器
--------	---------------------------------------

階梯圖	ST
	<pre>IF M0 = 0 Then M1 := 1; ELSE M1 := 0; END_IF ACTimer_10MS(TIM:= M1 , EN:= M1, CV:= R0, PV:= 10, TUP=> Y0);</pre>

- 係指該計時器之輸出 Y0 平常在斷電狀態下，在該計時器之輸入控制 X0 OFF 後延遲 100 毫秒後，輸出 Y0 才供電 (ON)。

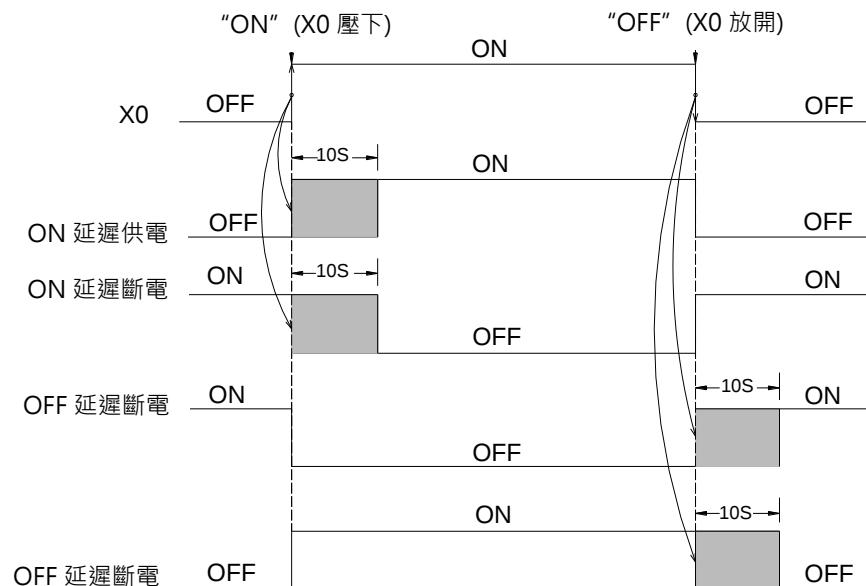
程式範例 4

OFF DELAY DE-ENERGIZING (OFF 延遲斷電) 計時器

階梯圖	ST
	<pre> IF M0 = 0 Then M1 := 1; ELSE M1 := 0; END_IF ACTimer_10MS(TIM:= M1 , EN:= M1, CV:= R0, PV:= 10, NUP=> Y0); </pre>

- 係指該計時器之輸出 Y0 平常在供電狀態下，在該計時器之計時控制 X0 OFF 後延遲 100 毫秒後輸出 Y0 才斷電 (OFF) 。

下圖為以上四種計時器之輸入與輸出之對應結果。

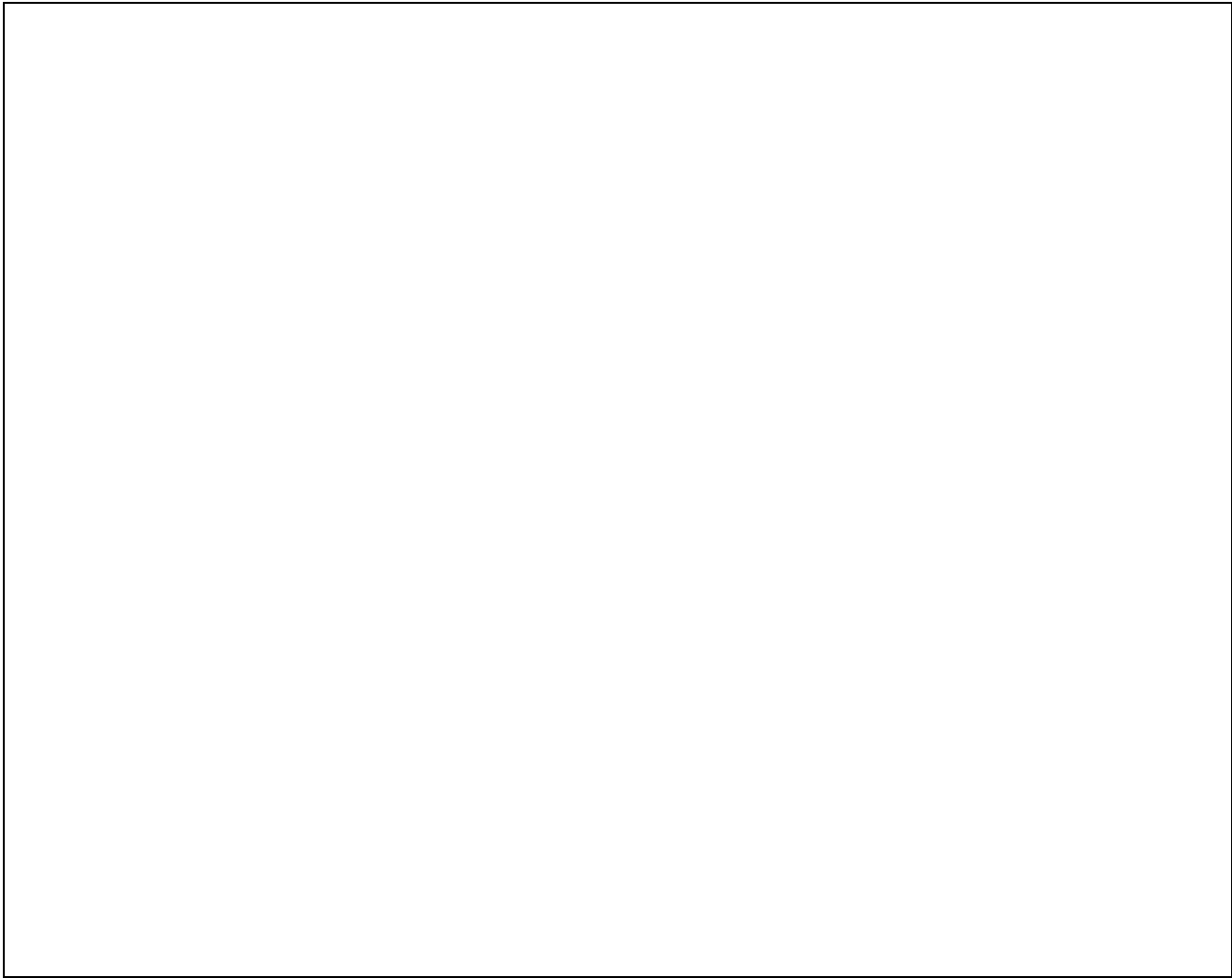


7-11 監控計時器指令 (FUN90 ~ 91)

7-11-1 監控定時器 (WATCHDOG TIMER) 設定時間設定

FUN90 	監控定時器 (WATCHDOG TIMER) 設定時間設定	FUN90 
指令說明		
階段圖符號 執行控制 — EN 90P. WDT 		N：監控定時器之設定時間。 其值祇能為 50、60、70.....990， 單位為 10MS，即設定之時間範圍為(50 ~ 990)×10MS，即 0.05 秒 ~ 9.9 秒。
程式敘述		

- 當執行控制“EN”=1 或由 0→1 (P 指令) 時，將監控定時器之設定時間改為 NX10MS。一旦設定後，WATCHDOG TIMER (WDT) 即以此為計時時間，若掃描時間超過此設定時間，PLC 將會停機不執行。
- WDT 設定時間主要是以系統應用上之安全考量而特別設計的，例如 PLC CPU 若突然損壞，無法執行程式或 I/O 更新時，經過 WDT 所設定之時間後，WDT 會自動從硬體上將 I/O 完全關閉以確保安全。在某些應用上若掃描時間太長亦可能造成某些安全上或不符控制要求之問題，亦可利用本指令設定您所要求之掃描時間極限值，超過您的設定值，PLC 會立刻停機，以確保安全。
- 設定值一旦設定後，即永遠保存，無需每次掃描均設定一次，因此本指令實用上應使用 P 指令。
- WDT 時間內定為 0.25 秒。
- WDT 之工作原理請參考 FUN91 (RSWDT) 指令。



7-11-2 清除監控計時器 (RESET WATCHDOG TIMER)

FUN91 P RSWDT	清除監控計時器 (RESET WATCHDOG TIMER)		FUN91 P RSWDT
指令說明			
階段圖符號 執行控制 — EN — 91P. RSWDT	本指令無運算元		
程式敘述			

- 當執行控制“EN”=1 或由 0→1 (P 指令) 時，將 WDT 計時器清除 (亦即使 WDT 重新由 0 開始計時)。
- WATCHDOG TIMER 之功能已在 FUN90 (WDT 指令) 中敘述，其原理如下：
- 監控計時器一般均為硬體單擊 (ONE-SHOT) 計時器 (不能用軟體作，否則 CPU 若當機，該計時器便失效，當然談不上能保護了)，所謂單擊，意思就是只要您觸發計時器一下，該計時器之計時值便立刻清為 0 再重新計時。若您在 WDT 開始計時後都未去觸發它，則 WDT 計時時間繼續累增至設定值 N 後 WDT 即動作，而將 PLC 停機。若您每次在 WDT 計時時間 N 尚未到達前就觸發 WDT 一次，則 WDT 永遠都不會發生，PLC 即利用此原理來確保系統安全，因為 PLC 一般會在程式掃描和 I/O 更新後，進入系統服務 (HOUSEKEEPING) 時觸發 WDT 一次，若系統正常且掃描時間未超出 WDT 之設定時間 N，就一定來得及在 WDT 未動作前清掉 WDT 而使之不動作，但若 CPU 損壞而無法觸發 WDT 或掃描時間過長致來不及在 N 時間內觸發 WDT，WDT 即會動作而關掉 PLC。

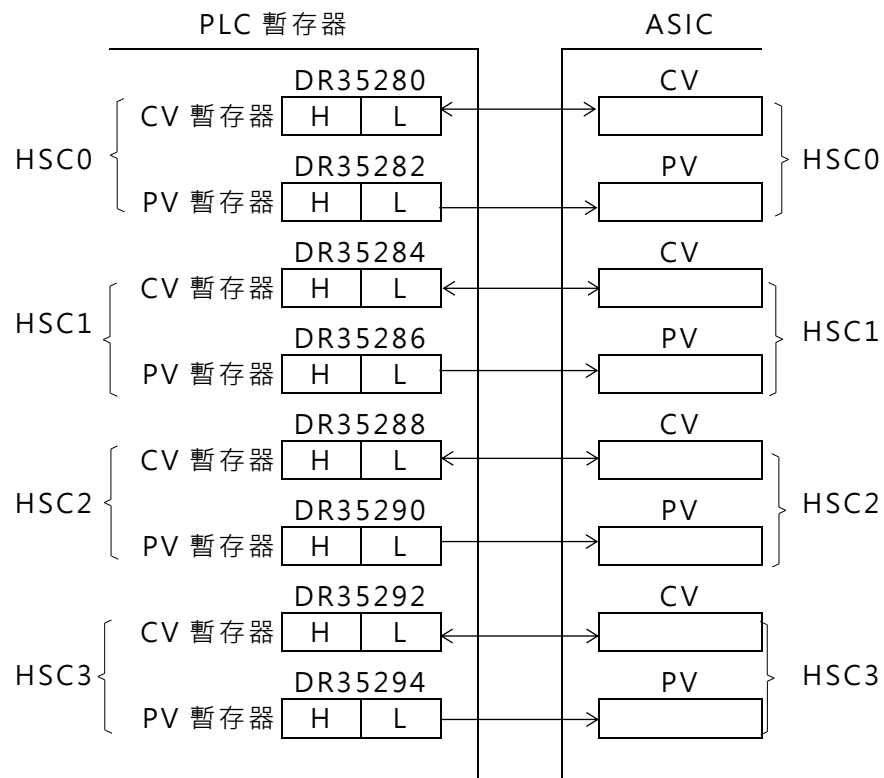
FUN91  RSWDT	清除監控計時器 (RESET WATCHDOG TIMER)	FUN91  RSWDT
● 在某些應用下，您已設好了 WDT 時間 (FUN90)，而您的程式在某些情況下掃描時間可能會暫時超出 WDT 之設定時間，這情況是您所能預期且允許的，您當然不希望因此而 PLC 停機，此時您可用本指令觸發 WDT 一下即可避免 WDT 發生，此即本指令之主要目的。		

7-12 高速計數器 / 計時器指令 (FUN92 ~ 93)

7-12-1 硬體高速計數器現在值 (CV) 讀取

FUN92D  HSCTR	硬體高速計數器現在值 (CV) 讀取	FUN92D  HSCTR
指令說明	*高速計數器作為 32BITS 時只能下數，PV 只能為 0 不開放設定。	
階段圖符號 讀取控制 — EN 92P. HSCTR CN CN：硬體高速計數器號碼 0：HSC0 1：HSC1 2：HSC2 3：HSC3 4：HSC4 5：HSC5 6：HSC6 7：HSC7		
功能敘述		

- M SERIES 之硬體高速計數器 HSC0~HSC3 是四組 32 位元可作雙相或單相上下數之高速計數器，是由 ASIC 內部硬體電路所組成，能獨立進行計數、比較，發出中斷，而不會佔用 CPU 的時間。〔軟體高速計數器 HSC4~HSC7 是用中斷方式由 CPU 來處理，因此當使用個數多或計數頻率高時，整個 PLC 之性能（掃描速度）將嚴重劣化〕。因為 HSC0~HSC3 之現在值 CV 是在 ASIC 內部之硬體電路中，使用者之控制（階梯圖）程式無法直接到 ASIC 去抓取，因此需利用本指令將硬體 HSC 之 CV 值讀出並放至控制程式能抓取到的暫存器去，以下為 ASIC 中 CV、PV 和 PLC 內部相對應之 CV、PV 暫存器之安排。

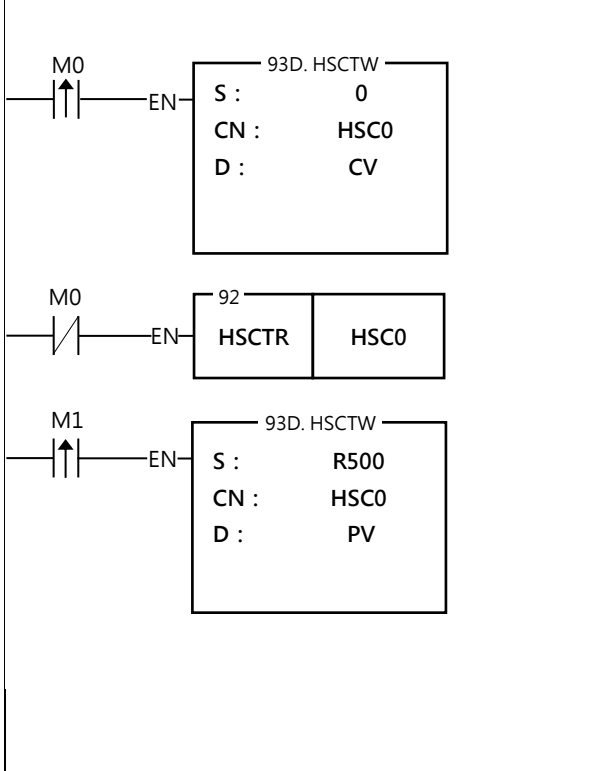
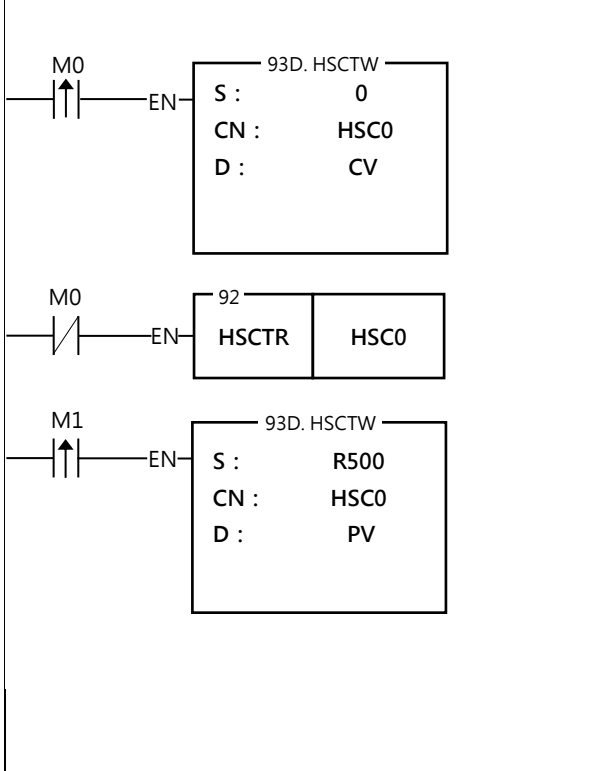
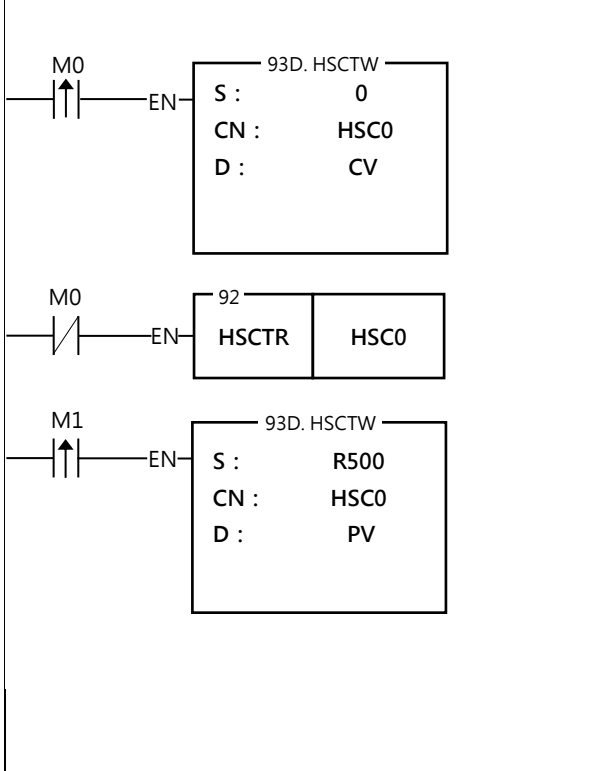


FUN92D  HSCTR	硬體高速計數器現在值 (CV) 讀取	FUN92D  HSCTR
	● 當讀取控制 “EN” =1 或由 0→1 ( 指令) 時，將 ASIC 中，CN 所指定的 HSC 之 CV 值讀到 PLC 內部該 HSC 所對應之 CV 暫存器去。(即將 HSC0 之 CV 讀到 DR35280 或將 HSC1 之 CV 讀到 DR35284)。 ● 雖然 ASIC 中之 PV 在 PLC 內部亦有 PV 暫存器與之對應，但卻不需讀取，因 ASIC 中之 PV 值是來自 PLC 內部之 PV 暫存器。 ● HSTA 係以 0.1MS 為時基之計時器，CV 之內容，代表經過多少個 0.1MS 時間。 ● 詳細之應用請參考軟體使用進階功能手冊第 8 章 “M SERIES 之高速計數器與高速計時器”。	

7-12-2 硬體高速計數器 CV 或 PV 值寫入

FUN93D  HSCTW	硬體高速計數器 CV 或 PV 值寫入	FUN93D  HSCTW
指令說明	※高速計數器作為 32BITS 時只能下數・PV 只能為 0 不開放設定。	
階段圖符號 寫入控制 — EN — 93DP.HSCTW S : CN : D : CN : 硬體高速計數器號碼 0 : HSC0 1 : HSC1 2 : HSC2 3 : HSC3 4 : HST4 2 : HSC2 3 : HSC3 4 : HST4 D : 寫入對象 (0 : 表示 CV・1 : 表示 PV)		
功能敘述		

FUN93D  HSCTW	硬體高速計數器 CV 或 PV 值寫入	FUN93D  HSCTW
	● 請先參閱 FUN92 有關 ASIC 中 HSC0 ~ HSC7 之 CV 或 PV 值和 PLC 內部相對應之 CV 暫存器和 PV 暫存器之關係。 ● 當寫入控制“EN”=1 或由 0→1 (P 指令) 時，將 PLC 內部 CN 所指定的那個高速計數器之 CV 暫存器或 PV 暫存器之內容值寫到 ASIC 內部相對應 HSC 的 CV 或 PV 去。 ● 一般應用常需寫入 PV，亦即將您所預設之設定值寫到 ASIC 中之 PV 去，當計數值到達您的設定值時，該計數器立即發出中斷，透過中斷服務程式您可作各種精密之計數或定位控制。 ● M SERIES 在電源斷電時會自動將當時 ASIC 內部 HSC0 ~ HSC3 之現在值暫存器 CV 之值讀出再將之寫入 PLC 內部 HSC0 ~ HSC3 之 CV 暫存器 (具斷電保持功能) 中，而在 PLC 復電時則會反向地將 PLC 內部之 CV 暫存器寫回 ASIC 內部之 CV 暫存器，因此每次 PLC 斷電再復電，ASIC 內部 HSC0 ~ HSC3 之 CV 暫存器內容值將會自動回復到上次斷電前之數值，但若您的控制應用在復電時需清為 0 或自某一特定值開始計數，您就必須利用本指令來作 ASIC 內部 HSC 之 CV 值寫入。 ● 詳細之應用請參閱軟體使用進階手冊第 7 章“M SERIES 之高速計數器與高速計時器”。	

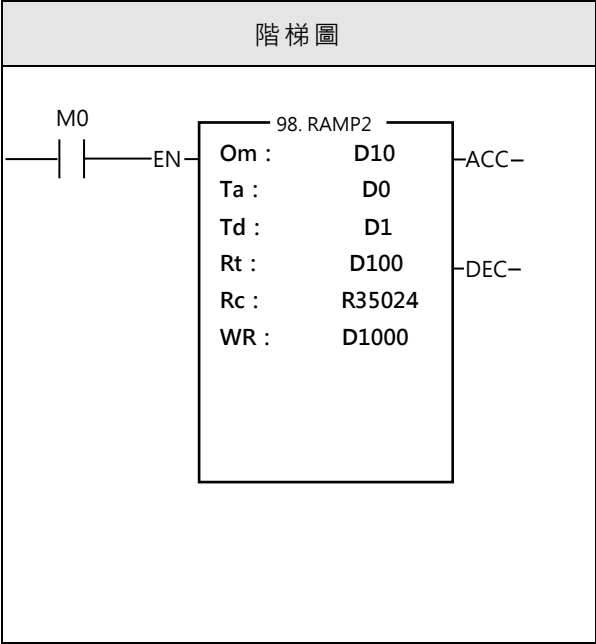
FUN93D P HSCTW	硬體高速計數器 CV 或 PV 值寫入	FUN93D P HSCTW				
範例程式						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF R_TRIG(S:= M0) THEN HSCTW(S:= 0, CN:= HSC_HSC0, D:= HSC_CV); END_IF IF M0 = FALSE THEN HSCTR(CN:= 0); END_IF IF R_TRIG(S:= M1) THEN HSCTW(S:= R500, CN:= HSC_HSC0, D:= HSC_PV); END_IF</pre></td></tr></table>			階梯圖	ST		<pre>IF R_TRIG(S:= M0) THEN HSCTW(S:= 0, CN:= HSC_HSC0, D:= HSC_CV); END_IF IF M0 = FALSE THEN HSCTR(CN:= 0); END_IF IF R_TRIG(S:= M1) THEN HSCTW(S:= R500, CN:= HSC_HSC0, D:= HSC_PV); END_IF</pre>
階梯圖	ST					
	<pre>IF R_TRIG(S:= M0) THEN HSCTW(S:= 0, CN:= HSC_HSC0, D:= HSC_CV); END_IF IF M0 = FALSE THEN HSCTR(CN:= 0); END_IF IF R_TRIG(S:= M1) THEN HSCTW(S:= R500, CN:= HSC_HSC0, D:= HSC_PV); END_IF</pre>					
上頁程式，M0 由 0→1 時，將 HSC0 目前值清除為 0，並透過 FUN93 寫入硬體 ASIC 中 ● M0 為 0 時，隨時讀出目前之計數值 ● M1 由 0→1 時，將 DR500 之計數設定值搬至 DR35282，並透過 FUN93 寫入硬體 ASIC 中 ● 當計數值等於 DR500 中之設定值時，立即執行 HSC0I 中斷處理副程式						

7-13 緩升 / 緩降指令 (FUN95 ~ 98)

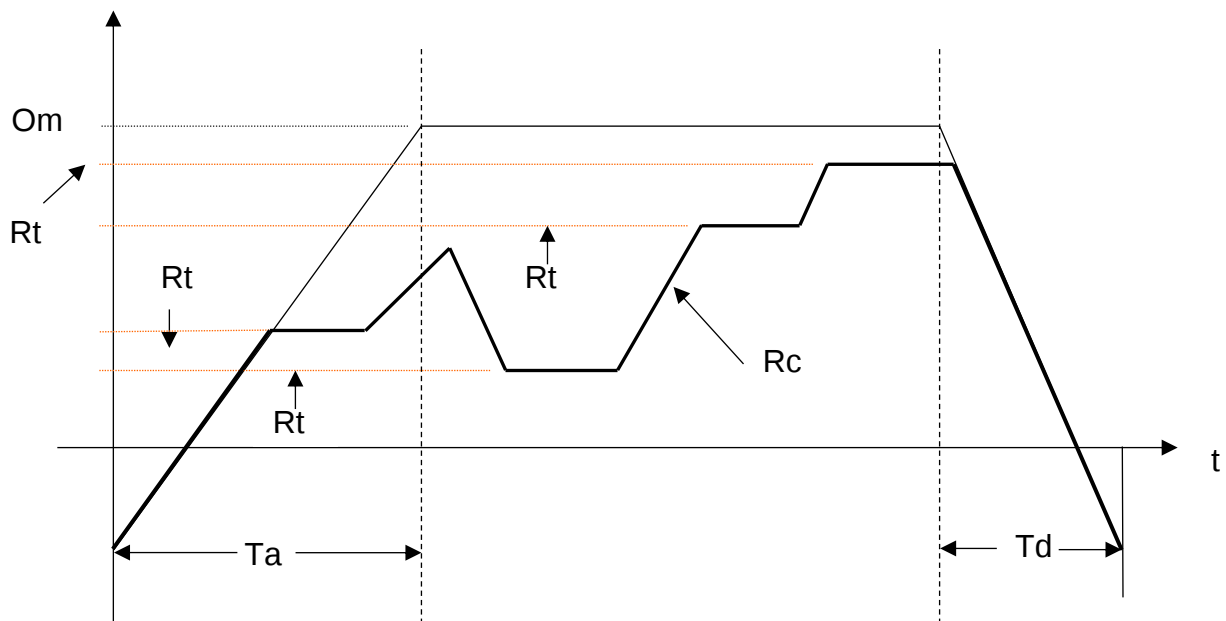
7-13-1 追蹤型類比輸出緩升 / 緩降指令

FUN98 RAMP2	追蹤型類比輸出緩升 / 緩降指令	FUN98 RAMP2																																															
指令說明																																																	
EN 98.RAMP2 Om : Ta : Td : Rt : Rc : WR : ACC DEC		Om：最大輸出設定；設定範圍 0~65535 Ta：由 0 到最大輸出之上升時間設定； 設定範圍 0~65000，單位為 mS Td：由最大輸出到 0 之下降時間設定； 設定範圍 0~65000，單位為 mS Rt：目標輸出量設定暫存器； 設定範圍 0~65535 Rc：目前輸出量暫存器，用來作 D/A 輸出 WR：工作暫存器起始位址，共佔用 4 個暫存器																																															
<table><tr><th rowspan="2">運算元 範圍</th><th>HR</th><th>OR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><td>R0 R3839</td><td>R35024 R3967</td><td>R5000 R8071</td><td>D0 D3999</td><td>16 位元</td></tr><tr><td>OM</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~65535</td></tr><tr><td>TA</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~65000</td></tr><tr><td>TD</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~65000</td></tr><tr><td>RT</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>RC</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>			運算元 範圍	HR	OR	ROR	DR	K	R0 R3839	R35024 R3967	R5000 R8071	D0 D3999	16 位元	OM	○	○	○	○	0~65535	TA	○	○	○	○	0~65000	TD	○	○	○	○	0~65000	RT	○	○	○	○		RC	○	○	○	○		WR	○	○	○*	○	
運算元 範圍	HR	OR		ROR	DR	K																																											
	R0 R3839	R35024 R3967	R5000 R8071	D0 D3999	16 位元																																												
OM	○	○	○	○	0~65535																																												
TA	○	○	○	○	0~65000																																												
TD	○	○	○	○	0~65000																																												
RT	○	○	○	○																																													
RC	○	○	○	○																																													
WR	○	○	○*	○																																													

FUN98 RAMP2	追蹤型類比輸出緩升 / 緩降指令	FUN98 RAMP2
功能敘述	● 當執行控制“EN”為 0 時，目前輸出量(RC)立即輸出為 0；輸出指示 ACC=0，DEC=0 ● 當執行控制“EN”為 1 時，首先以目前輸出量(RC)輸出，然後隨時比較目標輸出量(RT)與目前輸出量(RC)；如果目標輸出量設定值大於目前輸出量，則目前輸出量會以上升時間(TA)與最大輸出(OM)所決定之斜率增加輸出量，直到等於目標輸出量設定值為止(輸出量增加中，輸出指示 ACC=1)；如果目標輸出量設定值小於目前輸出量，則目前輸出量會以下降時間(TD)與最大輸出(OM)所決定之斜率減少輸出量，直到等於目標輸出量設定值為止(輸出量減少中，輸出指示 DEC=1)。 ● 目標輸出量(RT)設定值大於最大輸出(OM)設定時，會以最大輸出作為輸出控制。 ● 經本指令之執行，將目前輸出量(RC)之值搬至類比輸出暫存器(R35024~R35151)，可使受此類比輸出控制之動作較為平順。 ● 為確保本指令正確執行，目標輸出量(RT)設定值必須最少維持 2 個或以上掃描時間。 ● 本指令會佔用 4 個工作暫存器(WR)，其它程式不可重複使用。 ● 本指令雖為單向正值計算，但配合簡短的應用程式，也可作雙向正、負輸出緩升/緩降控制；請參考程式範例 2。	
程式範例 1	單向加/減速控制	

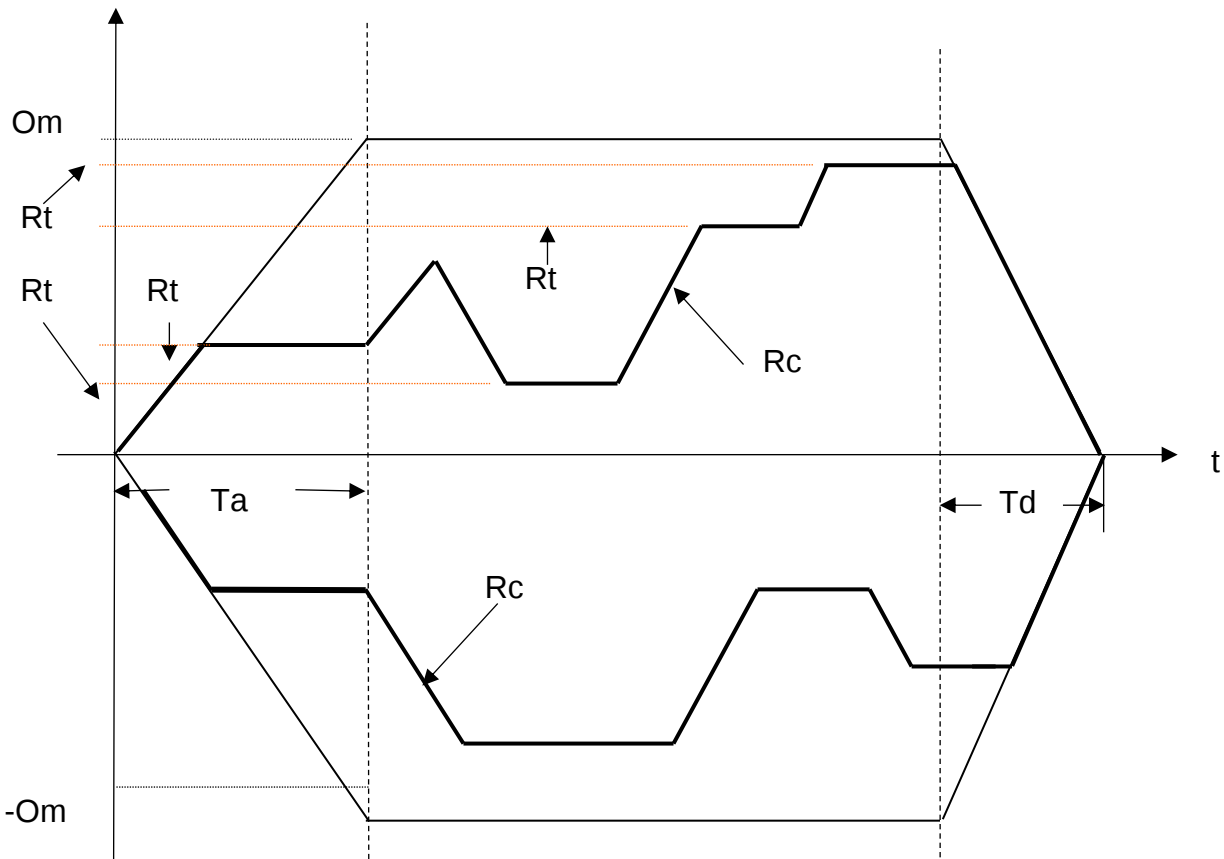
階梯圖	ST
	<pre>RAMP2(EN:= M0, Om:= D10, Ta:= D0, Td:= D1, Rt:= D100, Rc:= R35024, WR:= D1000, ACC=> M0, DEC=> M1);</pre>

FUN98 RAMP2	追蹤型類比輸出緩升 / 緩降指令	FUN98 RAMP2
D10：最大輸出設定，設定值為 16383 D0：由 0 到最大輸出之上升時間，設定值為 30000MS D1：由最大輸出到 0 之下降時間，設定值為 20000MS D100：目標輸出量設定暫存器，設定值為 8192 R35024：目前輸出量暫存器，用來作 D/A 輸出 D1000~D1003：工作暫存器 範例說明： 當 M0=0，目前輸出量立即輸出為 0(無緩降輸出)； 當 M0=1，首先以 R35024 目前值輸出，然後隨時比較目標輸出量(D100)與目前輸出量(R35024)；如果 D100 設定值大於 R35024 目前輸出值，則 R35024 目前值會以 16383/30000 (最大輸出=16383，上升時間=30000)之斜率增加輸出量，直到等於 D100 設定值為止(輸出量增加中，輸出指示 ACC=1)；如果 D100 設定值小於 R35024 目前輸出值，則 R35024 目前值會以 16383/20000 (最大輸出=16383，下降時間=20000)之斜率減少輸出量，直到等於 D100 設定值為止(輸出量減少中，輸出指示 DEC=1)。		



FUN98 RAMP2	追蹤型類比輸出緩升 / 緩降指令	FUN98 RAMP2			
範例程式 2	雙向加/減速控制				
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td> M0 — —EN 98. RAMP2 Om : D10 Ta : D0 Td : D1 Rt : D100 Rc : R35024 WR : D1000 M1 ACC—() M2 DEC—() M100 — —EN 08. MOV S : D200 D : R35024 M101 — —EN 08. MOV S : D200 D : D201 27 NEG D201 08. MOV S : D201 D : R35024 </td></tr></table>			階梯圖	ST	M0 — —EN 98. RAMP2 Om : D10 Ta : D0 Td : D1 Rt : D100 Rc : R35024 WR : D1000 M1 ACC—() M2 DEC—() M100 — —EN 08. MOV S : D200 D : R35024 M101 — —EN 08. MOV S : D200 D : D201 27 NEG D201 08. MOV S : D201 D : R35024
階梯圖	ST				
M0 — —EN 98. RAMP2 Om : D10 Ta : D0 Td : D1 Rt : D100 Rc : R35024 WR : D1000 M1 ACC—() M2 DEC—() M100 — —EN 08. MOV S : D200 D : R35024 M101 — —EN 08. MOV S : D200 D : D201 27 NEG D201 08. MOV S : D201 D : R35024					
D10：最大輸出設定，設定值為 8191 D0：由 0 到最大輸出之上升時間，設定值為 20000MS D1：由最大輸出到 0 之下降時間，設定值為 10000MS D100：目標輸出量設定暫存器，設定值為 0 D200：目前輸出量暫存器，用來作 D/A 輸出 D1000~D1003：工作暫存器					

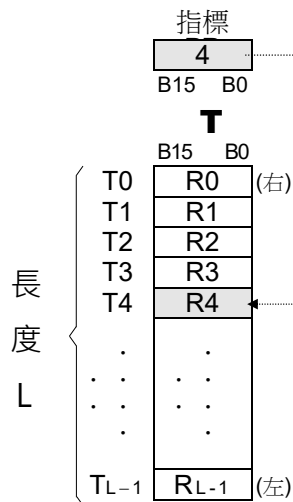
FUN98 RAMP2	追蹤型類比輸出緩升 / 緩降指令	FUN98 RAMP2
範例說明： 當 M0=0，目前輸出量立即輸出為 0(無緩降輸出)； 當 M0=1，首先以 D200 目前值輸出，然後隨時比較目標輸出量(D100)與目前輸出量(D200)；如果 D100 設定值大於 D200 目前輸出值，則 D200 目前值會以 8191/20000 (最大輸出=8191，上升時間=20000)之斜率增加輸出量，直到等於 D100 設定值為止(輸出量增加中，輸出指示 ACC=1)；如果 D100 設定值小於 D200 目前輸出值，則 D200 目前值會以 8191/10000 (最大輸出=8191，下降時間=10000)之斜率減少輸出量，直到等於 D100 設定值為止(輸出量減少中，輸出指示 DEC=1)。 正、負輸出控制，M100=1，正量輸出；M101=1，負量輸出。 不管正、負量輸出，目標輸出量(D100)設定值都需為正值(0~65535)。		



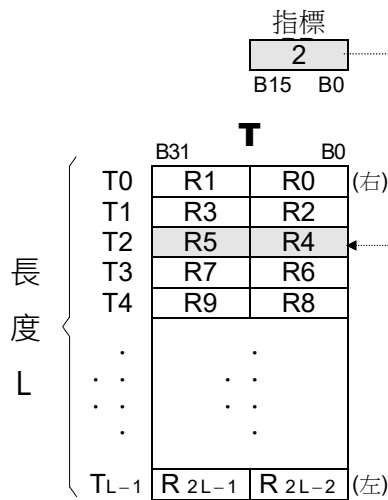
列 表 (T A B L E) 指 令

100. R→T	107. T_FIL
101. T→R	108. T_SHF
102. T→T	109. T_ROT
103. BT_M	110. QUEUE
104. T_SWP	111. STACK
105. R-T_S	112. BKCMP
106. T-T_C	

- 列表是 2 個以上連續之暫存器 (16 或 32 位元) 所組成，組成列表之暫存器個數稱為列表的長度 L (LENGTH)，列表指令運算每次均以列表之一個暫存器為單位 (即 16 或 32 位元資料)。
- 列表指令主要在處理列表和暫存器或列表和列表間之資料處理，諸如搬移、拷貝、比較、搜尋等，是極為方便和重要之應用指令。
- 在列表指令運作中通常需要有一指引器來指定列表中之某一暫存器當作運算對象，此指引器我們稱之為指標 PR (POINTER)。無論 16 或 32 位元列表指令其指標都只是一個 16 位元之暫存器。指標之有效範圍為 0~L-1，分別用以對應 (指引) 至列表之暫存器 T0~TL-1 (共 L 個)，以下為 16 位元及 32 位元列表之示意圖。
- 在列表運算中有左、右移或旋轉，我們定義高序號者為左，低序號者為右，如下圖示。



16 位元列表



32 位元列表

7-14 列表指令 (FUN100 ~ 114)

7-14-1 暫存器→列表搬移 (REGISTER TO TABLE MOVE)

FUN100 DP R→T		暫存器→列表搬移 (REGISTER TO TABLE MOVE)														FUN100 DP R→T			
指令說明																			
		階梯圖符號 100DP.R→T 搬移控制 — EN 指標遞增 — INC 指標清除 — CLR Rs : Td : L : Pr : END — 搬至最後 ERR — 指標錯誤										Rs：來源資料或其暫存器號碼 Td：目的列表之起頭暫存器號碼 L：目的列表之長度 Pr：指標暫存器號碼 Rs、Td 可結合 V、Z、P0~P9 作間接定址應用							
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR				
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9				
	Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○				
	Td		○	○	○	○	○	○	○		○	○*	○*	○	○				
	L							○				○*	○	2~2048					
	Pr		○	○	○	○	○	○	○		○	○*	○*	○					
功能敘述																			
		● 當搬移控制“EN”=1 或由 0→1 (P 指令) 時，將來源暫存器 Rs 的內容寫到目的列表 Td (長度為 L) 中指標 Pr 所指之暫存器 Tdpr 去。在執行搬移前本指令會先檢視指標清除“CLR”之輸入信號，若“CLR”為 1，則會先將指標 Pr 之內容清除為 0 後再作搬移。在作完搬移動作後接著檢視 Pr 值，若 Pr 值已達 L-1 (已指在列表之最後一個暫存器) 則將搬至最後旗號“END”設為 1 後結束本指令之執行；若 Pr 小於 L-1，則再檢視指標遞增“INC”之狀態，若“INC”為 1，則再將 Pr 加 1 後才結束執行。此外，指標清除“CLR”可單獨執行，不受其他輸入影響。																	

FUN100 DP R→T	暫存器→列表搬移 (REGISTER TO TABLE MOVE)	FUN100 DP R→T																	
指標之有效範圍為 0 ~ L-1，超出此範圍則指標錯誤“ERR”設為 1，且本指令不執行。																			
程式範例																			
<table> <tr> <th>階梯圖</th> <th>ST</th> </tr> <tr> <td> </td> <td></td> </tr> </table>			階梯圖	ST															
階梯圖	ST																		
- 左圖程式例假設剛開始指標 Pr=4，列表 Td 內容全部為 0，而 Rs 值為 8888，下圖為當 X0 連續由 0→1 變換 2 次所得之運算結果。 - 因 INC 為 1，故每執行一次 Pr 即加 1 一次。																			
<table> <tr> <td> Rs R0 8888 </td> <td> Pr R50 4 </td> <td> Td R10(T0) 0000 R11(T1) 0000 R12(T2) 0000 R13(T3) 0000 R14(T4) 0000 R15(T5) 0000 R16(T6) 0000 R17(T7) 0000 </td> <td> X0 = 1 (第一次) </td> <td> Pr R50 5 </td> <td> Td R10 0000 R11 0000 R12 0000 R13 0000 R14 8888 R15 0000 R16 0000 R17 0000 </td> <td> X0 = 1 (第二次) </td> <td> Pr R50 6 </td> <td> Td R10 0000 R11 0000 R12 0000 R13 0000 R14 8888 R15 8888 R16 0000 R17 0000 </td> </tr> <tr> <td colspan="3">執行前狀態</td> <td></td> <td colspan="2">第一次執行結果</td> <td colspan="2">第二次執行結果</td> </tr> </table>			Rs R0 8888	Pr R50 4	Td R10(T0) 0000 R11(T1) 0000 R12(T2) 0000 R13(T3) 0000 R14(T4) 0000 R15(T5) 0000 R16(T6) 0000 R17(T7) 0000	X0 = 1 (第一次)	Pr R50 5	Td R10 0000 R11 0000 R12 0000 R13 0000 R14 8888 R15 0000 R16 0000 R17 0000	X0 = 1 (第二次)	Pr R50 6	Td R10 0000 R11 0000 R12 0000 R13 0000 R14 8888 R15 8888 R16 0000 R17 0000	執行前狀態				第一次執行結果		第二次執行結果	
Rs R0 8888	Pr R50 4	Td R10(T0) 0000 R11(T1) 0000 R12(T2) 0000 R13(T3) 0000 R14(T4) 0000 R15(T5) 0000 R16(T6) 0000 R17(T7) 0000	X0 = 1 (第一次)	Pr R50 5	Td R10 0000 R11 0000 R12 0000 R13 0000 R14 8888 R15 0000 R16 0000 R17 0000	X0 = 1 (第二次)	Pr R50 6	Td R10 0000 R11 0000 R12 0000 R13 0000 R14 8888 R15 8888 R16 0000 R17 0000											
執行前狀態				第一次執行結果		第二次執行結果													

7-14-2 列表→暫存器搬移 (TABLE TO REGISTER MOVE)

FUN101 DP T→R		列表→暫存器搬移 (TABLE TO REGISTER MOVE)										FUN101 DP T→R																																																																																																			
指令說明																																																																																																															
		階梯圖符號 搬移控制 — EN 指標遞增 — INC 指標清除 — CLR 101DP.T→R Ts : L : Pr : Rd : END — 搬至最後 ERR — 指標錯誤										Ts：來源列表之起頭暫存器號碼 L：來源列表之長度 Pr：指標暫存器號碼 Rd：目的暫存器之起頭號碼 Ts · Rd 可結合 V、Z、P0~P9 作間接定址應用																																																																																																			
運算元		<table><tr><th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WSM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WSM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 或 32 位元 正、負 數</td><td>V、Z P0~P9</td></tr><tr><td>Ts</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td></td><td></td></tr><tr><td>Pr</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td>2~2048</td><td></td></tr><tr><td>Rd</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr></table>														範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9	Ts	○	○	○	○	○	○	○	○	○	○	○	○	○		○	L								○				○*	○			Pr		○	○	○	○	○	○	○		○	○*	○*	○	2~2048		Rd		○	○	○	○	○	○	○		○	○*	○*	○		○
範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																																
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V、Z P0~P9																																																																																																
Ts	○	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																																
L								○				○*	○																																																																																																		
Pr		○	○	○	○	○	○	○		○	○*	○*	○	2~2048																																																																																																	
Rd		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																																
功能敘述																																																																																																															
		● 當搬移控制“EN”=1 或由 0→1 (P 指令) 時，將來源列表 Ts (長度為 L) 中指標 Pr 所指之暫存器 Tspr 之內容值寫到目的暫存器 Rd 去。在執行搬移前本指令會先檢視指標清除“CLR”之輸入信號，若“CLR”為 1，則會先將指標 Pr 之內容清除為 0 後再作搬移。在作完搬移動作後接著檢視 Pr 值，若 Pr 值已達 L-1 (已指在列表之最後一個暫存器) 則將搬至最後旗號“END”設為 1 後結束本指令之執行；若 Pr 小於 L-1，則再檢視指標遞增“INC”之狀態，若“INC”為 1，則再將 Pr 加 1 後才結束執行。此外，指標清除“CLR”可單獨執行，不受其他輸入影響。																																																																																																													

FUN101 T→R	列表→暫存器搬移 (TABLE TO REGISTER MOVE)	FUN101 T→R
● 指標之有效範圍為 0 ~ L-1，超出此範圍則指標錯誤“ERR” 設為 1，且本指令不執行。		
程式範例		
階段圖 X0 ● — EN ● — INC — CLR 101P.T→R Ts : R 0 L : 9 Pr : R 19 Rd : R 20 — END — ERR ST		
● 左圖程式例假設剛開始指標 Pr=7，而 Ts 和 Rd 內容如下圖左所示，當 X0 連續由 0→1 變換 2 次後，可得到下圖右方之兩個結果。 ● 第二次執行時指標已至最後，故不再增加。 Ts R0(T0) 1111 R1(T1) 2222 R2(T2) 3333 R3(T3) 4444 R4(T4) 5555 R5(T5) 6666 R6(T6) 7777 R7(T7) 8888 R8(T8) 9999 Pr 7 R19 Rd 0000 R20 END 0 執行前狀態 Pr 8 R19 X0 = 1 (第一次) ⇒ Rd 8888 R20 END 0 第一次執行結果 Pr 8 R19 X0 = 1 (第二次) ⇒ Rd 9999 R20 END 1 第二次執行結果		

7-14-3 列表→列表搬移 (TABLE TO TABLE MOVE)

FUN102 T→T		列表→列表搬移 (TABLE TO TABLE MOVE)												FUN102 T→T		
指令說明																
		階梯圖符號 搬移控制 — EN 指標遞增 — INC 指標清除 — CLR 102DP.T→T Ts : Td : L : Pr : END — 搬至最後 ERR — 指標錯誤												Ts：來源列表之暫存器起頭號碼 Td：目的列表之暫存器起頭號碼 L：列表 (Ts 和 Td) 之長度 Pr：指標暫存器號碼 Ts、Td 可結合 V、Z、P0~P9 作間接定址應用		
運算元 範圍		WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	2 2048	V、Z P0~P9
Ts		○	○	○	○	○	○	○	○	○	○	○	○	○		○
Td			○	○	○	○	○	○	○		○	○*	○*	○		○
L									○				○*	○	○	
Pr			○	○	○	○	○	○	○		○	○*	○*	○		
功能敘述																
		● 當搬移控制“EN”=1 或由 0→1 (P 指令) 時，將來源列表中指標 Pr 所指的那一個暫存器資料 Tspr 搬到目的列表中同樣是 Pr 所指之暫存器 Tdpr 去。在執行搬移前本指令會先檢視指標清除輸入“CLR”信號，若其為 1，會先清除 Pr 為 0 後再搬移 (此時為 Ts0→Td0)。在作完搬移動作後接著檢視指標 Pr 之值，若 Pr 值已達 L-1 (已指到列表之最後一對暫存器)，則將搬至最後旗號“END”設為 1 後結束指令之執行；若 Pr 值小於 L-1，將再檢視指標遞增“INC”之狀態，若“INC”為 1，則再將 Pr 值加 1 後才結束指令之執行。此外，指標清除“CLR”可單獨執行，不受其他輸入影響。														

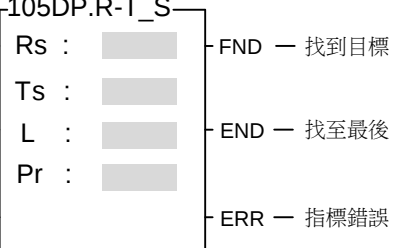
FUN102 T→T	列表→列表搬移 (TABLE TO TABLE MOVE)	FUN102 T→T																																																														
● 指標之有效範圍為 0 ~ L - 1，超出此範圍則指標錯誤“ERR” 設為 1，且本指令不執行。																																																																
程式範例																																																																
<table><tr><td>階梯圖</td><td>ST</td></tr><tr><td></td><td></td></tr></table>			階梯圖	ST																																																												
階梯圖	ST																																																															
● 上圖程式範例起始狀態如下圖左之執行前狀態，當 X0 由 0→1 時，將得到下圖右之結果，Ts 為來源列表不受指令執行之影響。																																																																
<table><tr><td colspan="2"> Pr R205 Ts <table><tr><td>R0</td><td>1111</td></tr><tr><td>R1</td><td>1111</td></tr><tr><td>R2</td><td>1111</td></tr><tr><td>R3</td><td>1111</td></tr><tr><td>R4</td><td>1111</td></tr><tr><td>R5</td><td>1111</td></tr><tr><td>R6</td><td>1111</td></tr><tr><td>R7</td><td>1111</td></tr><tr><td>R8</td><td>1111</td></tr><tr><td>R9</td><td>1111</td></tr></table> Td <table><tr><td>R10</td><td>0000</td></tr><tr><td>R11</td><td>0000</td></tr><tr><td>R12</td><td>0000</td></tr><tr><td>R13</td><td>0000</td></tr><tr><td>R14</td><td>8888</td></tr><tr><td>R15</td><td>0000</td></tr><tr><td>R16</td><td>0000</td></tr><tr><td>R17</td><td>0000</td></tr><tr><td>R18</td><td>0000</td></tr><tr><td>R19</td><td>0000</td></tr></table> X0 = 1 ⇒ Pr R206 Td <table><tr><td>R10</td><td>0 0 0 0</td></tr><tr><td>R11</td><td>0 0 0 0</td></tr><tr><td>R12</td><td>0 0 0 0</td></tr><tr><td>R13</td><td>0 0 0 0</td></tr><tr><td>R14</td><td>8 8 8 8</td></tr><tr><td>R15</td><td>1 1 1 1</td></tr><tr><td>R16</td><td>0 0 0 0</td></tr><tr><td>R17</td><td>0 0 0 0</td></tr><tr><td>R18</td><td>0 0 0 0</td></tr><tr><td>R19</td><td>0 0 0 0</td></tr></table> </td></tr></table> 執行前狀態 執行結果			Pr R205 Ts <table><tr><td>R0</td><td>1111</td></tr><tr><td>R1</td><td>1111</td></tr><tr><td>R2</td><td>1111</td></tr><tr><td>R3</td><td>1111</td></tr><tr><td>R4</td><td>1111</td></tr><tr><td>R5</td><td>1111</td></tr><tr><td>R6</td><td>1111</td></tr><tr><td>R7</td><td>1111</td></tr><tr><td>R8</td><td>1111</td></tr><tr><td>R9</td><td>1111</td></tr></table> Td <table><tr><td>R10</td><td>0000</td></tr><tr><td>R11</td><td>0000</td></tr><tr><td>R12</td><td>0000</td></tr><tr><td>R13</td><td>0000</td></tr><tr><td>R14</td><td>8888</td></tr><tr><td>R15</td><td>0000</td></tr><tr><td>R16</td><td>0000</td></tr><tr><td>R17</td><td>0000</td></tr><tr><td>R18</td><td>0000</td></tr><tr><td>R19</td><td>0000</td></tr></table> X0 = 1 ⇒ Pr R206 Td <table><tr><td>R10</td><td>0 0 0 0</td></tr><tr><td>R11</td><td>0 0 0 0</td></tr><tr><td>R12</td><td>0 0 0 0</td></tr><tr><td>R13</td><td>0 0 0 0</td></tr><tr><td>R14</td><td>8 8 8 8</td></tr><tr><td>R15</td><td>1 1 1 1</td></tr><tr><td>R16</td><td>0 0 0 0</td></tr><tr><td>R17</td><td>0 0 0 0</td></tr><tr><td>R18</td><td>0 0 0 0</td></tr><tr><td>R19</td><td>0 0 0 0</td></tr></table>		R0	1111	R1	1111	R2	1111	R3	1111	R4	1111	R5	1111	R6	1111	R7	1111	R8	1111	R9	1111	R10	0000	R11	0000	R12	0000	R13	0000	R14	8888	R15	0000	R16	0000	R17	0000	R18	0000	R19	0000	R10	0 0 0 0	R11	0 0 0 0	R12	0 0 0 0	R13	0 0 0 0	R14	8 8 8 8	R15	1 1 1 1	R16	0 0 0 0	R17	0 0 0 0	R18	0 0 0 0	R19	0 0 0 0
Pr R205 Ts <table><tr><td>R0</td><td>1111</td></tr><tr><td>R1</td><td>1111</td></tr><tr><td>R2</td><td>1111</td></tr><tr><td>R3</td><td>1111</td></tr><tr><td>R4</td><td>1111</td></tr><tr><td>R5</td><td>1111</td></tr><tr><td>R6</td><td>1111</td></tr><tr><td>R7</td><td>1111</td></tr><tr><td>R8</td><td>1111</td></tr><tr><td>R9</td><td>1111</td></tr></table> Td <table><tr><td>R10</td><td>0000</td></tr><tr><td>R11</td><td>0000</td></tr><tr><td>R12</td><td>0000</td></tr><tr><td>R13</td><td>0000</td></tr><tr><td>R14</td><td>8888</td></tr><tr><td>R15</td><td>0000</td></tr><tr><td>R16</td><td>0000</td></tr><tr><td>R17</td><td>0000</td></tr><tr><td>R18</td><td>0000</td></tr><tr><td>R19</td><td>0000</td></tr></table> X0 = 1 ⇒ Pr R206 Td <table><tr><td>R10</td><td>0 0 0 0</td></tr><tr><td>R11</td><td>0 0 0 0</td></tr><tr><td>R12</td><td>0 0 0 0</td></tr><tr><td>R13</td><td>0 0 0 0</td></tr><tr><td>R14</td><td>8 8 8 8</td></tr><tr><td>R15</td><td>1 1 1 1</td></tr><tr><td>R16</td><td>0 0 0 0</td></tr><tr><td>R17</td><td>0 0 0 0</td></tr><tr><td>R18</td><td>0 0 0 0</td></tr><tr><td>R19</td><td>0 0 0 0</td></tr></table>		R0	1111	R1	1111	R2	1111	R3	1111	R4	1111	R5	1111	R6	1111	R7	1111	R8	1111	R9	1111	R10	0000	R11	0000	R12	0000	R13	0000	R14	8888	R15	0000	R16	0000	R17	0000	R18	0000	R19	0000	R10	0 0 0 0	R11	0 0 0 0	R12	0 0 0 0	R13	0 0 0 0	R14	8 8 8 8	R15	1 1 1 1	R16	0 0 0 0	R17	0 0 0 0	R18	0 0 0 0	R19	0 0 0 0			
R0	1111																																																															
R1	1111																																																															
R2	1111																																																															
R3	1111																																																															
R4	1111																																																															
R5	1111																																																															
R6	1111																																																															
R7	1111																																																															
R8	1111																																																															
R9	1111																																																															
R10	0000																																																															
R11	0000																																																															
R12	0000																																																															
R13	0000																																																															
R14	8888																																																															
R15	0000																																																															
R16	0000																																																															
R17	0000																																																															
R18	0000																																																															
R19	0000																																																															
R10	0 0 0 0																																																															
R11	0 0 0 0																																																															
R12	0 0 0 0																																																															
R13	0 0 0 0																																																															
R14	8 8 8 8																																																															
R15	1 1 1 1																																																															
R16	0 0 0 0																																																															
R17	0 0 0 0																																																															
R18	0 0 0 0																																																															
R19	0 0 0 0																																																															

7-14-4 整個列表搬移 (BLOCK TABLE MOVE)

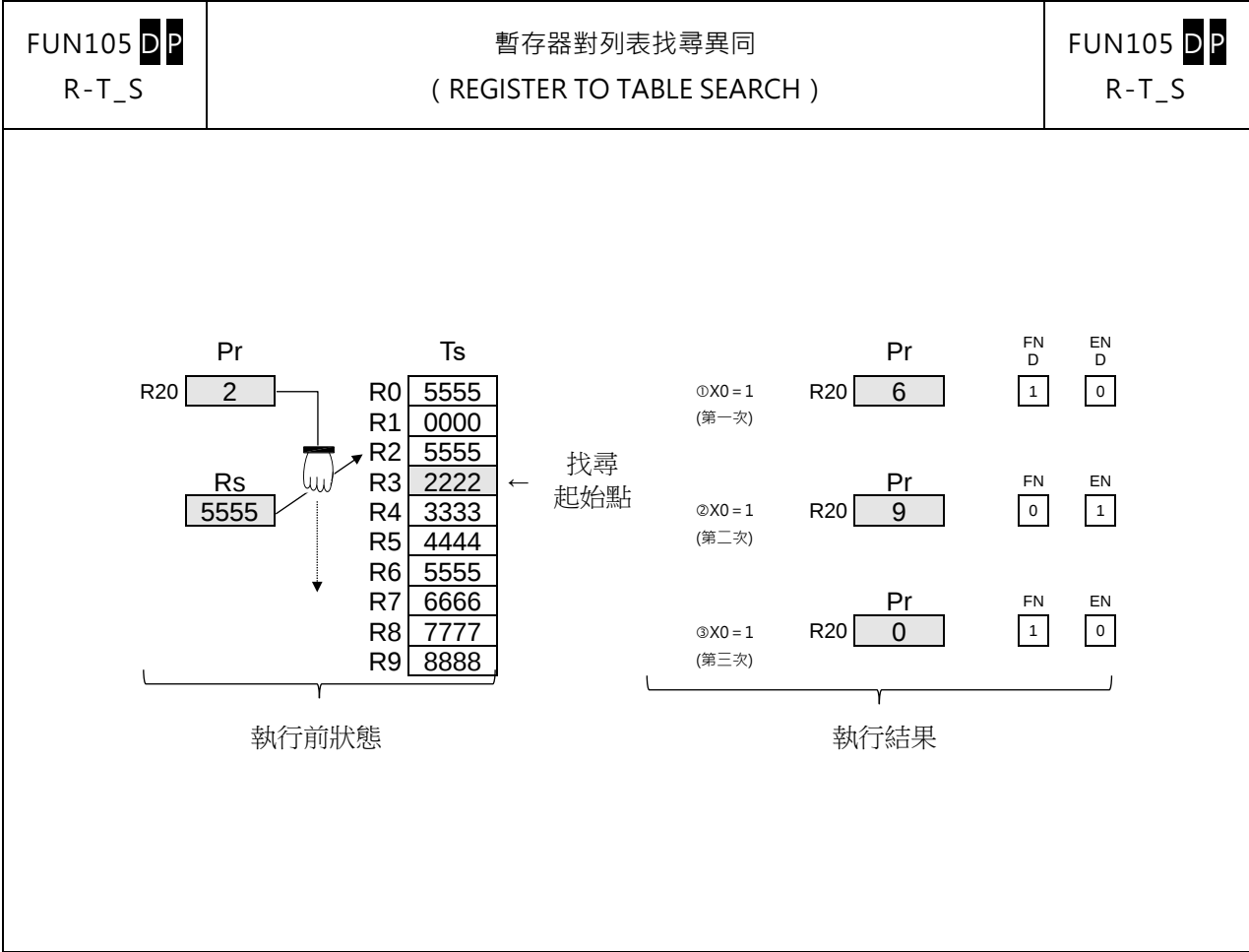
FUN103 DP BT_M	整個列表搬移 (BLOCK TABLE MOVE)		FUN103 DP BT_M																																																																																															
指令說明																																																																																																		
階段圖符號 搬移控制 — EN — 103DP.BT_M — Ts : Td : L : 		TS : 來源列表之起頭暫存器號碼 TD : 目的列表之起頭暫存器號碼 L : 來源和目的列表之長度 TS、TD 可結合 V、Z、P0~P9 作間接定址應用																																																																																																
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="width: 5%;">範圍</th><th style="width: 5%;">WX</th><th style="width: 5%;">WY</th><th style="width: 5%;">WM</th><th style="width: 5%;">WSM</th><th style="width: 5%;">WS</th><th style="width: 5%;">TMR</th><th style="width: 5%;">CTR</th><th style="width: 5%;">HR</th><th style="width: 5%;">IR</th><th style="width: 5%;">OR</th><th style="width: 5%;">SR</th><th style="width: 5%;">ROR</th><th style="width: 5%;">DR</th><th style="width: 5%;">K</th><th style="width: 5%;">XR</th></tr> <tr> <td rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">運算元</td><td>WX0</td><td>WY0</td><td>WM0</td><td>WM9120</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td>2</td><td>V、Z</td></tr> <tr> <td>WX1008</td><td>WY1008</td><td>WM9104</td><td>WM25584</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td>256</td><td>P0~P9</td></tr> <tr> <td>TS</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr> <tr> <td>TD</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr> <tr> <td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td>○</td><td></td></tr> </table>				範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	運算元	WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	2	V、Z	WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	256	P0~P9	TS	○	○	○	○	○	○	○	○	○	○	○	○	○		○	TD		○	○	○	○	○	○	○		○	○*	○*	○		○	L								○				○*	○	○	
範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																			
運算元	WX0	WY0	WM0	WM9120	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	2	V、Z																																																																																			
	WX1008	WY1008	WM9104	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	256	P0~P9																																																																																			
TS	○	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																			
TD		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																			
L								○				○*	○	○																																																																																				
功能敘述																																																																																																		
● 本指令來源和目的列表長度相同，且在一次指令執行中即將整個 TS 列表資料全部搬至 TD 中，故無需指標來指定列表中之某一暫存器。 ● 當搬移控制“EN”=1 或由 0→1 (P 指令) 時，將來源列表 TS (長度為 L) 的所有資料整個一次搬至相同長度之目的列表 TD 去。 ● 因本指令在每次執行中是整個列表一次搬完，若列表長度較長時，將會耗費較多的時間，實用上應使用 P 指令，以免每次掃描均重複同樣之搬移動作而浪費時間。																																																																																																		

FUN103 BT_M	整個列表搬移 (BLOCK TABLE MOVE)	FUN103 BT_M
程式範例		
階段圖 X0 — —EN 103.BT_M Ts : R0 Td : R10 L : 10 ST IF X0 THEN BT_M(Ts:= R0, Td:= R10, L:= 10); END_IF		
● 上圖程式範例，假設 TS 和 TD 列表之狀態如下圖左之執行前狀態，當 X0 由 0→1 時，可得到下圖右之執行結果。 TS R0000 R1111 R2222 R3333 R4444 R5555 R6666 R7777 R8888 R9999 → → → → → → → → → TD R10000 R110000 R120000 R130000 R140000 R150000 R160000 R170000 R180000 R190000 X0 = 1 ⇒ TD R10000 R11111 R12222 R13333 R14444 R15555 R16666 R17777 R18888 R19999 執行前狀態 執行後狀態		

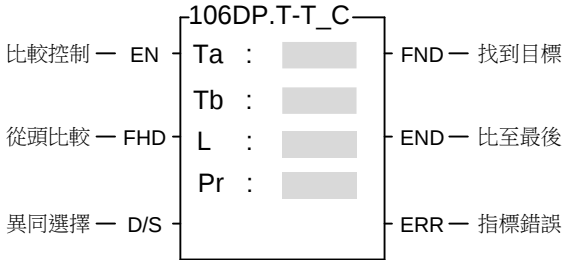
7-14-5 暫存器對列表找尋異同 (REGISTER TO TABLE SEARCH)

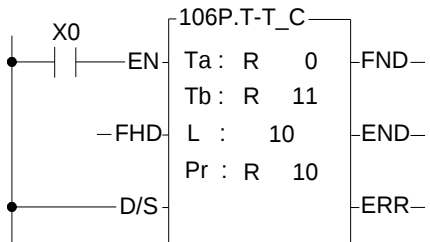
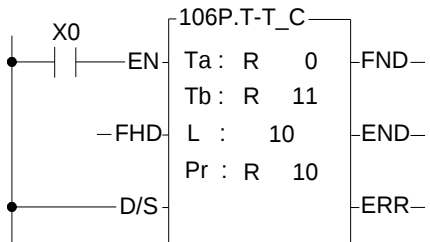
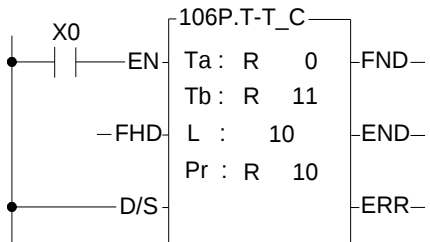
FUN105  R-T_S		暫存器對列表找尋異同 (REGISTER TO TABLE SEARCH)										FUN105  R-T_S				
指令說明																
		階梯圖符號 						Rs：找尋之來源（樣本）資料或其暫存器號碼 Ts：被找尋之列表起頭暫存器號碼 L：列表長度 Pr：指標，用以記錄目標所在之位置值 Rs，Ts 可結合 V、Z、P0~P9 作間接定址應用								
運算元	範圍	WX WX1008	WY WY1008	WM WM9104	WSM WM25584	WS WS3088	TMR T1023	CTR C1279	HR R34767	IR R34895	OR R35024 R35151	SR R35280 R43223	ROR R43224 R47319	DR D0 D7999	K 16 或 32 位元 正、負 數	XR V、Z P0~P9
	Rs	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Ts	○	○	○	○	○	○	○	○	○	○	○	○	○		○
	L								○				○*	○	2~256	
	Pr		○	○	○	○	○	○	○		○	○*	○*	○		

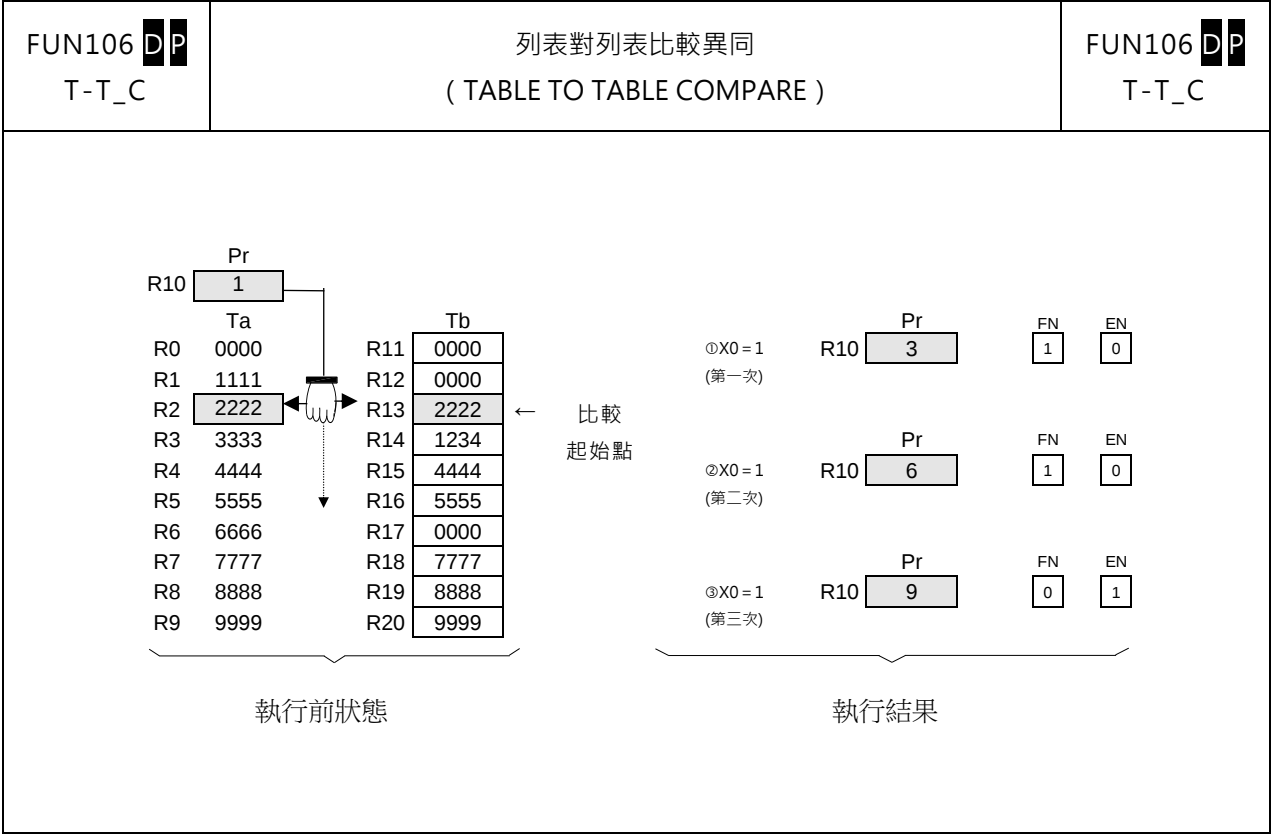
FUN105 DP R-T_S	暫存器對列表找尋異同 (REGISTER TO TABLE SEARCH)	FUN105 DP R-T_S				
功能敘述						
● 當找尋控制“EN” =1 或由 0→1 (P 指令) 時，自列表 Ts 之開頭第一個暫存器開始 (“FHD” =1 或 Pr 值已達 L-1 時) 或自列表中當時指標所指那個暫存器之下一個暫存器 Tspr+1 開始 (“FHD” =0 同時 Pr 值小於 L-1) 往下找尋和樣本資料 Rs 不同 (D/S=1 時) 或相同 (D/S=0 時) 之暫存器。若找到目標 (不同或相同者)，則立即停止找尋動作，並將該目標在列表之位置序號值存到指標 Pr 去，同時將找到目標旗號“FND” 設為 1 後結束本指令之執行。當找到列表之最後一個暫存器時，無論是否找到目標均將結束該次指令執行，並將找至最後旗號“END” 設為 1，而 Pr 值則停在 L-1。當本指令下次再度被執行時，Pr 將會自動循環至列表之最開頭 (Pr=0) 開始往下找尋。 ● 指標值之有效範圍為 0 ~ L-1，若其值超出此範圍，則指標錯誤旗號“ERR” 變為 1，且本指令不執行。						
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN T_Search(FHD:= FALSE, DS:= FALSE, Rs:= 5555, Ts:= R0, L:= 10, Pr:= R20, FND=> M0, END=> M1, ERR=> M2); END_IF</pre></td></tr></table> ● 上圖程式範例在列表中找尋數值同為 5555 之暫存器 (因 D/S=0，為找相同)，執行前指標指在 R2，但開始找尋點是 Pr+1 (即 R3 開始)，在 X0 連續 3 次由 0→1 動作後，可得到如下圖①、②、③三個結果。			階梯圖	ST		<pre>IF X0 THEN T_Search(FHD:= FALSE, DS:= FALSE, Rs:= 5555, Ts:= R0, L:= 10, Pr:= R20, FND=> M0, END=> M1, ERR=> M2); END_IF</pre>
階梯圖	ST					
	<pre>IF X0 THEN T_Search(FHD:= FALSE, DS:= FALSE, Rs:= 5555, Ts:= R0, L:= 10, Pr:= R20, FND=> M0, END=> M1, ERR=> M2); END_IF</pre>					



7-14-6 列表對列表比較異同 (TABLE TO TABLE COMPARE)

FUN106 DP T-T_C	列表對列表比較異同 (TABLE TO TABLE COMPARE)														FUN106 DP T-T_C
指令說明															
階段圖符號 								Ta：列表 a 之暫存器起頭號碼 Tb：列表 b 之暫存器起頭號碼 L：列表 a 和 b 之長度 Pr：指標，用以記錄目標所在之位置值 Ta，Tb 可結合 V、Z、P0~P9 作間接定址應用							
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WSM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	2 256	V、Z P0~P9
	Ta	○	○	○	○	○	○	○	○	○	○	○	○		○
	Tb	○	○	○	○	○	○	○	○	○	○	○	○		○
	L							○				○*	○	○	
Pr															
功能敘述															

FUN106 DP T-T_C	列表對列表比較異同 (TABLE TO TABLE COMPARE)	FUN106 DP T-T_C				
● 當比較控制“EN” =1 或由 0→1 時 (P 指令)，自 Ta 和 Tb 兩列表中之最開頭那對暫存器 (Ta0 和 Tb0) 開始 (“FHD” =1 或 Pr 值已達 L-1 時) 或自當時 Pr 所指那對暫存器的下一對暫存器 (Tapr+1 和 Tbpr+1) 開始 (“FHD” =0 同時 Pr 值小於 L-1) 往下雙雙成對比較找尋內容值不同 (“D/S” =1 時) 或相同 (“D/S” =0 時) 之暫存器對 (Pair)，當找到目標 (不同或相同者) 後立即停止比較找尋，同時將該對目標在列表中之位置序號值存到指標 Pr 去，並將找到目標旗號“FND”設為 1，然後結束本指令之執行。當找到列表之最後一對暫存器，無論其是否為所要找尋之目標，均將結束本指令之執行，同時將比至最後旗號“END”設為 1，而指標值則停在 L-1。而當本指令下一次再度被執行時，Pr 將會自動循環至列表之最開頭開始往下找起。指標 Pr 之範圍為 0~L-1，在運作中應避免更動到 Pr 值，以免影響其正確之比較找尋，若 Pr 值超出此範圍，則指標錯誤旗號“ERR”變為 1，且本指令不執行。						
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN T_Compare(FHD:= FALSE, DS:= TRUE, Ta:= R0, Tb:= R11, L:= 10, Pr:= R10, FND=> M0, END=> M1, ERR=> M2); END_IF</pre></td></tr></table> ● 右圖程式範例為自指標所指之下一個暫存器開始往下比較找尋 (因“FHD”為 0) 兩列表中資料不相同之暫存器對 (因“D/S”=1)。剛開始 Pr 指在 Ta1 和 Tb1，雖然兩列表中分別在列表位置 1、3、6 三處有資料不同，但因其非從頭比較，故本指令將先找到位置 3，下圖右顯示 X0 由 0→1 變換 3 次之結果。			階梯圖	ST		<pre>IF X0 THEN T_Compare(FHD:= FALSE, DS:= TRUE, Ta:= R0, Tb:= R11, L:= 10, Pr:= R10, FND=> M0, END=> M1, ERR=> M2); END_IF</pre>
階梯圖	ST					
	<pre>IF X0 THEN T_Compare(FHD:= FALSE, DS:= TRUE, Ta:= R0, Tb:= R11, L:= 10, Pr:= R10, FND=> M0, END=> M1, ERR=> M2); END_IF</pre>					

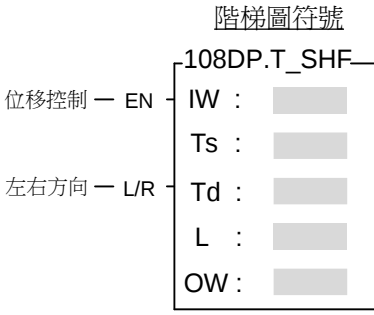


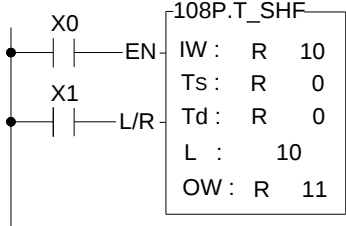
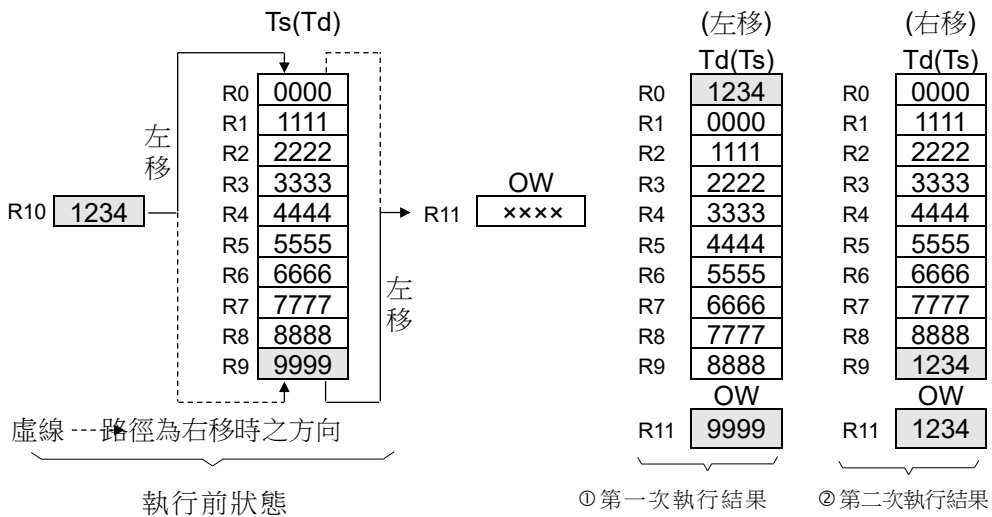
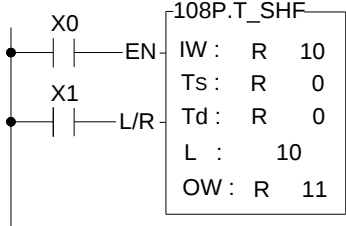
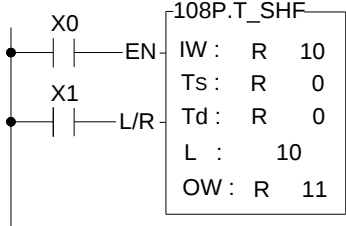
7-14-7 列表填塞 (TABLE FILL)

FUN107 T_FIL		列表填塞 (TABLE FILL)												FUN107 T_FIL																																																																																		
指令說明																																																																																																
		階段圖符號 107DP.T_FIL 填塞控制 — EN — Rs : Td : L :										RS : 欲填入列表之來源資料或其暫存器號碼 TD : 列表之起頭暫存器號碼 L : 列表之長度 RS · TD 可結合 V、Z、P0~P9 作間接定址應用																																																																																				
		<table><tr><td rowspan="2">運算元 範圍</td><td>WX</td><td>WY</td><td>WM</td><td>WSM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>16 或 32 位元 正、負數</td><td>V、Z P0~P9</td></tr><tr><td>RS</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>TD</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td>2~256</td><td></td></tr></table>																運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9	RS	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	TD		○	○	○	○	○	○	○		○	○*	○*	○		○	L								○				○*	○	2~256	
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																	
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9																																																																																	
RS	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○																																																																																	
TD		○	○	○	○	○	○	○		○	○*	○*	○		○																																																																																	
L								○				○*	○	2~256																																																																																		
功能敘述																																																																																																
		● 當填塞控制 “EN” =1 或由 0→1 (P 指令) 時，將 RS 之資料填寫到列表 TD 中所有的暫存器中。 ● 本指令主要用於列表之清除 (填 0) 或一致化 (全部填相同值) 用，實用上應使用 指令。																																																																																														

FUN107 T_FIL	列表填塞 (TABLE FILL)	FUN107 T_FIL
程式範例		
階段圖 X0 107.T_FIL EN Ts : 555 Td : R0 L : 10 ST IF X0 THEN T_Fill(Rs:= 555, Td:= R0, L:= 10); END_IF ● 上圖程式範例，將列表 TD 全部填入 5555，如下圖結果。 RS 5555 R0 1547 R1 2314 R2 7725 R3 0013 R4 5247 R5 1925 R6 6744 R7 5319 R8 9788 R9 2796 X0 =  ⇒ R0 5555 R1 5555 R2 5555 R3 5555 R4 5555 R5 5555 R6 5555 R7 5555 R8 5555 R9 5555 執行前狀態 執行結果		

7-14-8 列表位移 (TABLE SHIFT)

FUN108  T_SHF		列表位移 (TABLE SHIFT)										FUN108  T_SHF				
指令說明																
		階段圖符號 						IW：填補位移空位之資料或其暫存器號碼 Ts：被位移之來源列表 Td：存放結果之目的列表 L：列表 Ts 和 Td 長度 OW：存放自列表移出資料之暫存器號碼 Ts・Td 可結合 V、Z、P0~P9 作間接定址應用								
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負 數	V・Z P0~P9
	IW	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
	Ts	○	○	○	○	○	○	○	○	○	○	○	○	○		○
	Td		○	○	○	○	○	○	○		○	○*	○*	○		○
	L								○				○*	○	2~256	
	OW		○	○	○	○	○	○	○		○	○*	○*	○		
功能敘述																
		當位移控制“EN”=1 或由 0→1 (P 指令) 時，將列表 Ts 之資料全部取出，整個向左 (“L/R” =1 時) 或向右 (“L/R” =0 時) 位移一個位置，因位移造成之空位，以 IW 填補之，再將此位移過並填補之結果，寫到列表 Td 去，而因位移而擠出之資料則寫到 OW 去。														

FUN108 DP T_SHF	列表位移 (TABLE SHIFT)	FUN108 DP T_SHF				
程式範例						
<table border="1"> <thead> <tr> <th>階梯圖</th><th>ST</th></tr> </thead> <tbody> <tr> <td>  </td><td></td></tr> </tbody> </table> ● 左圖程式範例 Ts 和 Td 相同，因此就是把列表自己位移後再寫回自己（列表必須為可寫入之暫存器），如下圖執行前之狀態，先執行一次左移（使 X1=1，再使 X0 由 0→1），然後再作一次右移（使 X1=0，再使 X0 由 0→1），將可得到下圖右方之兩個結果。 			階梯圖	ST		
階梯圖	ST					
						

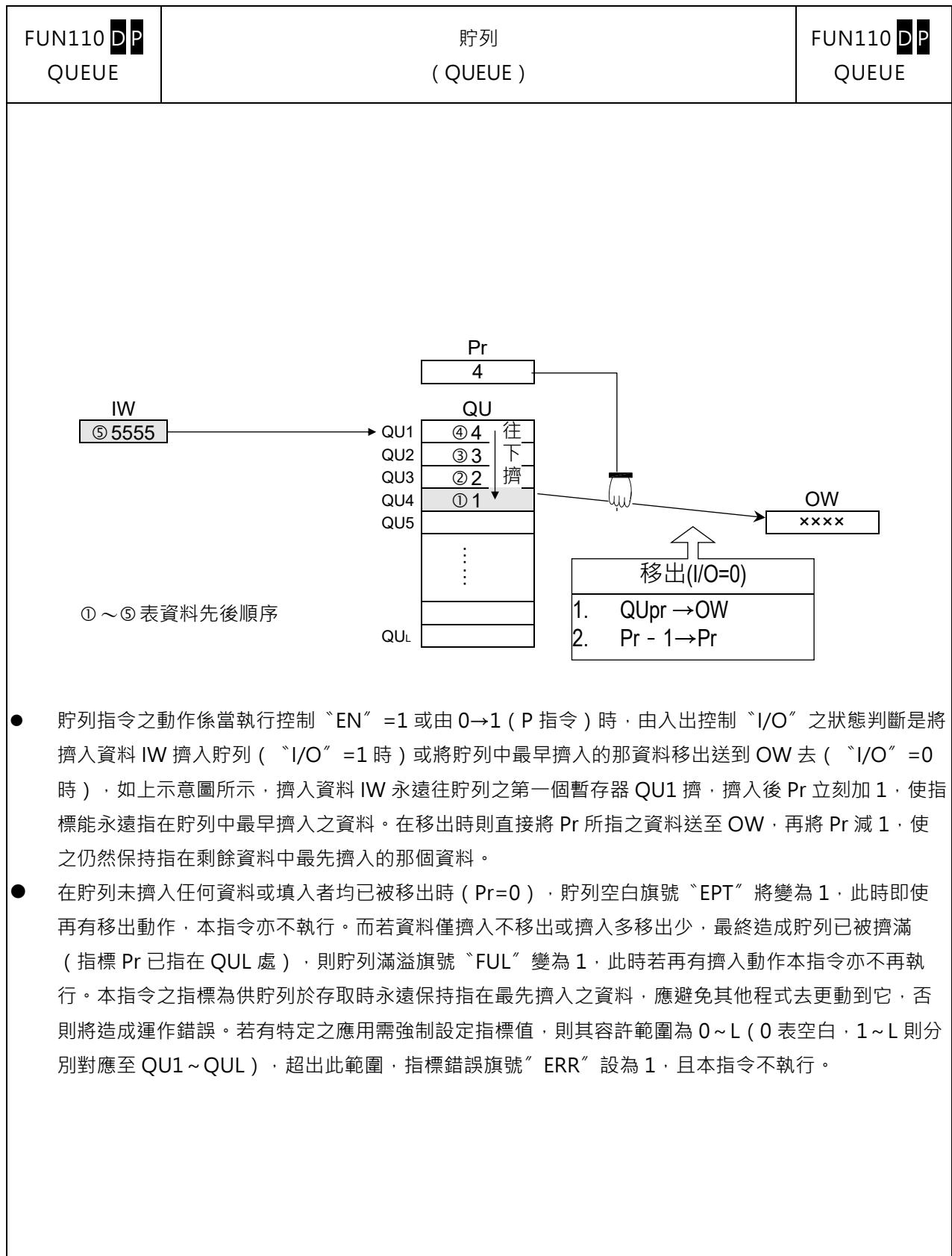
7-14-9 列表旋轉 (TABLE ROTATE)

FUN109 DP T_ROT	列表旋轉 (TABLE ROTATE)												FUN109 DP T_ROT																																																																																																																					
指令說明																																																																																																																																		
階段圖符號 旋轉控制 — EN — 109DP.T_ROT 左/右方向 — L/R — <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>Ts :</td><td></td></tr> <tr><td>Td :</td><td></td></tr> <tr><td>L :</td><td></td></tr> </table>						Ts :		Td :		L :		Ts : 被旋轉之來源列表 Td : 存放旋轉結果之列表 L : 列表 Ts 和 Td 之長度 Ts · Td 可結合 V、Z、P0~P9 作間接定址應用																																																																																																																						
Ts :																																																																																																																																		
Td :																																																																																																																																		
L :																																																																																																																																		
<table border="1"> <tr> <th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr> <tr> <td rowspan="4">運算元</td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WM9120 WM2558 4</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>2 256</td><td>V · Z P0~P9</td></tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>Ts</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr> <tr> <td>Td</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr> <tr> <td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td>○</td><td></td></tr> </table>														範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	運算元	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	2 256	V · Z P0~P9																																											Ts	○	○	○	○	○	○	○	○	○	○	○	○		○	Td		○	○	○	○	○	○		○	○*	○*	○		○	L							○				○*	○	○	
範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																																																				
運算元	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM2558 4	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	2 256	V · Z P0~P9																																																																																																																				
Ts	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																																																				
Td		○	○	○	○	○	○		○	○*	○*	○		○																																																																																																																				
L							○				○*	○	○																																																																																																																					
功能敘述																																																																																																																																		
當旋轉控制“EN”=1 或由 0→1 (P 指令) 時，將列表 Ts 之資料取出後向左 (“L/R”=1 時) 或向右 (“L/R”=0 時) 旋轉一個位置後，將此旋轉過之結果寫到 Td 去。																																																																																																																																		

FUN109 T_ROT	列表旋轉 (TABLE ROTATE)	FUN109 T_ROT
程式範例		
階段圖 X0 X1 109P.T_ROT Ts: R 0 Td: R 0 L : 10 ST ● 左圖程式例，Ts 和 Td 相同，故是將列表自己取出旋轉後再寫回自己，如下圖執行前之狀態，先執行一次左旋（使 X1=1，再使 X0 由 0→1），然後再作一次右旋（使 X1=0，再使 X0 由 0→1）後，可得到下圖右方之①、②兩個結果。 左旋 右旋 Ts(Td) R0000 R1111 R2222 R3333 R4444 R5555 R6666 R7777 R8888 R9999 (右) (左) 執行前 (左旋) (右旋) Td(Ts) R09999 R10000 R21111 R32222 R43333 R54444 R65555 R76666 R87777 R98888 ① 第一次執行結果 Td(Ts) R00000 R11111 R22222 R33333 R44444 R55555 R66666 R77777 R88888 R99999 ② 第二次執行結果		

7-14-10 貯列 (QUEUE)

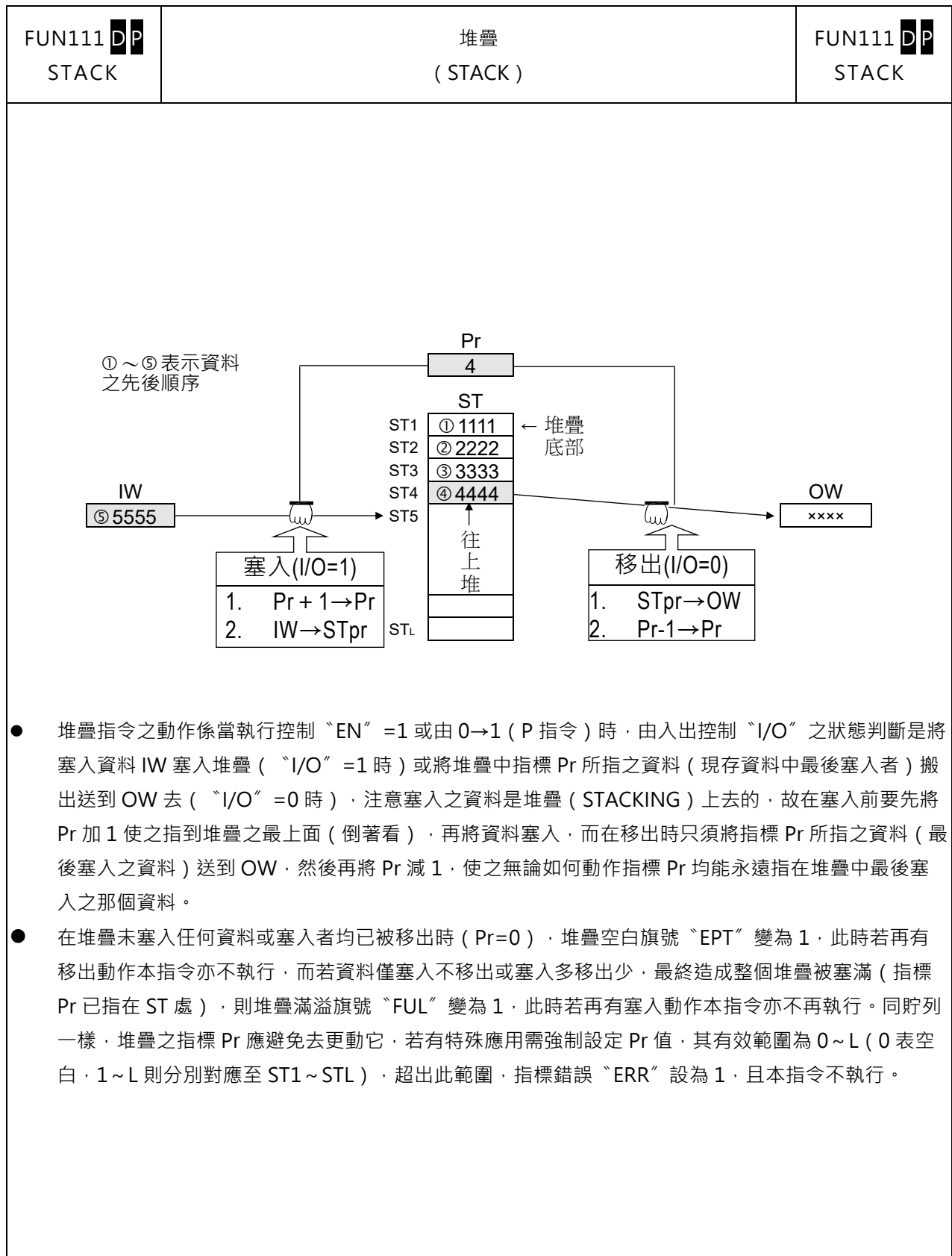
FUN110 DP QUEUE		貯列 (QUEUE)										FUN110 DP QUEUE		
指令說明														
		階段圖符號 執行控制 — EN — 110DP.QUEUE — EPT — 貯列空白 入/出控制 — I/O — QU : — FUL — 貯列滿溢 Pr : — ERR — 指標錯誤 OW : —										IW : 擠入貯列之資料或其暫存器號碼 QU : 貯列之起頭暫存器號碼 L : 貯列之長度 PR : 指標暫存器號碼 OW : 接收自貯存器移出資料之暫存器號碼 QU 可結合 V、Z、P0~P9 作間接定址應用		
運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS WM25584	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9
	IW	○	○	○	○	○	○	○	○	○	○	○	○	
	QU		○	○	○	○	○		○	○	○*	○		○
	L						○				○*	○	2~256	
	Pr		○	○	○	○	○		○	○*	○*	○		
	OW		○	○	○	○	○		○	○*	○*	○		
功能敘述														
		- 貯列 (QUEUE) 亦屬列表之一種，其有別於一般列表的是其貯列暫存器序號是由 1~L 而非 0~L-1，亦即 QU1~QUL，分別以指標 Pr=1~L 來對應，而指標 Pr=0 則用以表示該貯列為空白。 注意: 系統使用序號 L+1 暫存器做內部運算，使用者必須避開此暫存器 - 貯列 (QUEUE) 是一種先進先出裝置，即最先擠入 (PUSH) 貯列之資料，在移出 (POP) 時要最先移出。本指令之貯列是由 QU 暫存器開始之連續 L 個 16 位元或 32 位元 (D 指令) 暫存器所組成。												



FUN110 DP QUEUE	貯列 (QUEUE)	FUN110 DP QUEUE				
程式範例						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td> X0 — —EN X1 — —I/O 110. QUEUE IW : R0 Qu : R2 L : 10 Pr : R1 OW : R20 EPT— FUL— ERR— </td><td> IF X0 THEN QUEUE(InOut:= X1, IW:= R0, Qu:= R2, L:= 10, Pr:= R1, OW:= R20, EPT=> M0, FUL=> M1); END_IF </td></tr></table>			階段圖	ST	X0 — —EN X1 — —I/O 110. QUEUE IW : R0 Qu : R2 L : 10 Pr : R1 OW : R20 EPT— FUL— ERR—	IF X0 THEN QUEUE(InOut:= X1, IW:= R0, Qu:= R2, L:= 10, Pr:= R1, OW:= R20, EPT=> M0, FUL=> M1); END_IF
階段圖	ST					
X0 — —EN X1 — —I/O 110. QUEUE IW : R0 Qu : R2 L : 10 Pr : R1 OW : R20 EPT— FUL— ERR—	IF X0 THEN QUEUE(InOut:= X1, IW:= R0, Qu:= R2, L:= 10, Pr:= R1, OW:= R20, EPT=> M0, FUL=> M1); END_IF					
● 上圖範例程式，假設其起始狀態如上頁之貯列示意圖範例所示，先將之作一次貯列擠入動作，再作一次移出動作，將可得到如下兩個執行結果。無論如何 Pr 總是指向在 QUEUE 中最先擠入之資料。 Pr 5 QU QU15555R2 QU24444R3 QU33333R4 QU42222R5 QU51111R6 QU6R7 QU7R8 QU8R9 QU9R10 QU10R11 OW XXXXR20 ↑ OW 值不變 Pr 4 QU QU15555R2 QU24444R3 QU33333R4 QU42222R5 QU5R6 QU6R7 QU7R8 QU8R9 QU9R10 QU10R11 OW 1111R20 執行擠入動作(X1=1，X0 由 0→1)後 執行移出動作(X1=0，X0 由 0→1)後						

7-14-11 堆疊 (STACK)

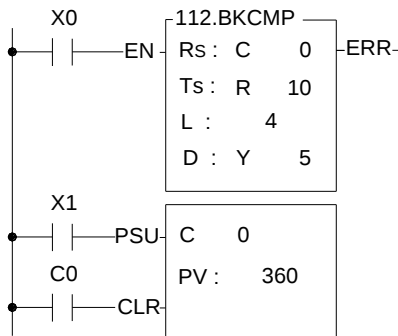
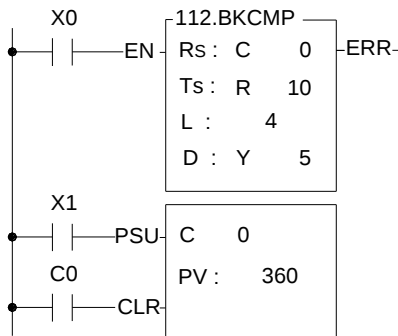
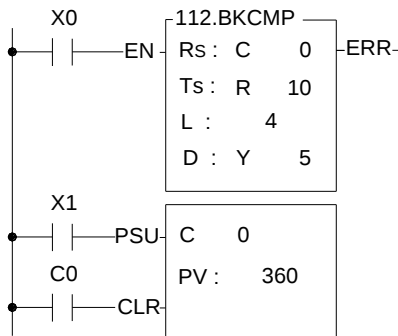
FUN111 D P STACK		堆疊 (STACK)										FUN111 D P STACK			
指令說明															
		階梯圖符號 執行控制 — EN — 111DP.STACK — EPT — 堆疊空白 入/出控制 — I/O — ST : — FUL — 堆疊滿溢 Pr : — ERR — 指標錯誤 OW : —										IW : 塞入堆疊之資料或其暫存器號碼 ST : 堆疊之起頭暫存器號碼 L : 堆疊之長度 PR : 指標暫存器號碼 OW : 接收堆疊移出資料之暫存器號碼 ST 可結合 V、Z、P0~P9 作間接定址應用			
範圍 運算 元	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9	
	IW	○	○	○	○	○	○	○	○	○	○	○	○		
	ST		○	○	○	○	○		○	○*	○*	○		○	
	L						○				○*	○	2~256		
	Pr		○	○	○	○	○		○	○*	○*	○			
	OW		○	○	○	○	○		○	○*	○*	○			
功能敘述															
		- 堆疊和貯列一樣同屬列表之一種，其指標序號性質和貯列完全相同，以 Pr=1~L 來對應 ST1~STL，而 Pr=0 則用以表示該堆疊為空白。 - 堆疊和貯列正好相反，是一種後進先出之裝置，即最後塞入 (PUSH) 堆疊之資料，在移出 (POP) 時要最先移出，堆疊是由 ST 開始之連續 L 個 16 位元或 32 位元 (D 指令) 暫存器所組成，如下之示意圖所示：													



FUN111 DP STACK	堆疊 (STACK)	FUN111 DP STACK				
程式範例						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN STACK(InOut:= X1, IW:= R0, ST:= R2, L:= 10, Pr:= R1, OW:= R20, EPT=> M0, FUL=> M1); END_IF</pre></td></tr></table>			階段圖	ST		<pre>IF X0 THEN STACK(InOut:= X1, IW:= R0, ST:= R2, L:= 10, Pr:= R1, OW:= R20, EPT=> M0, FUL=> M1); END_IF</pre>
階段圖	ST					
	<pre>IF X0 THEN STACK(InOut:= X1, IW:= R0, ST:= R2, L:= 10, Pr:= R1, OW:= R20, EPT=> M0, FUL=> M1); END_IF</pre>					
● 上圖範例程式，假設堆疊狀況正好如上頁之堆疊示意圖所示，先將之作一次塞入動作，再執行一次移出動作，可得到如下兩個結果，無論如何動作 Pr 始終指在堆疊中最後塞入的那個資料。 <table><tr><td> Pr 5 R1 ST ST1 1111 R2 ST2 2222 R3 ST3 3333 R4 ST4 4444 R5 ST5 5555 R6 ST6 R7 ST7 R8 ST8 R9 ST9 R10 ST10 R11 OW XXXX R20 ↑ OW 值不變 </td><td> Pr 4 QU ST1 1111 R2 ST2 2222 R3 ST3 3333 R4 ST4 4444 R5 ST5 R6 ST6 R7 ST7 R8 ST8 R9 ST9 R10 ST10 R11 OW 5555 R20 </td></tr><tr><td>執行塞入動作(X1=1，X0 由 0→1)後</td><td>執行移出動作(X1=0，X0 由 0→1)後</td></tr></table>			Pr 5 R1 ST ST1 1111 R2 ST2 2222 R3 ST3 3333 R4 ST4 4444 R5 ST5 5555 R6 ST6 R7 ST7 R8 ST8 R9 ST9 R10 ST10 R11 OW XXXX R20 ↑ OW 值不變	Pr 4 QU ST1 1111 R2 ST2 2222 R3 ST3 3333 R4 ST4 4444 R5 ST5 R6 ST6 R7 ST7 R8 ST8 R9 ST9 R10 ST10 R11 OW 5555 R20	執行塞入動作(X1=1，X0 由 0→1)後	執行移出動作(X1=0，X0 由 0→1)後
Pr 5 R1 ST ST1 1111 R2 ST2 2222 R3 ST3 3333 R4 ST4 4444 R5 ST5 5555 R6 ST6 R7 ST7 R8 ST8 R9 ST9 R10 ST10 R11 OW XXXX R20 ↑ OW 值不變	Pr 4 QU ST1 1111 R2 ST2 2222 R3 ST3 3333 R4 ST4 4444 R5 ST5 R6 ST6 R7 ST7 R8 ST8 R9 ST9 R10 ST10 R11 OW 5555 R20					
執行塞入動作(X1=1，X0 由 0→1)後	執行移出動作(X1=0，X0 由 0→1)後					

7-14-12 區塊比較 (凸輪開關 DRUM) (BLOCK COMPARE)

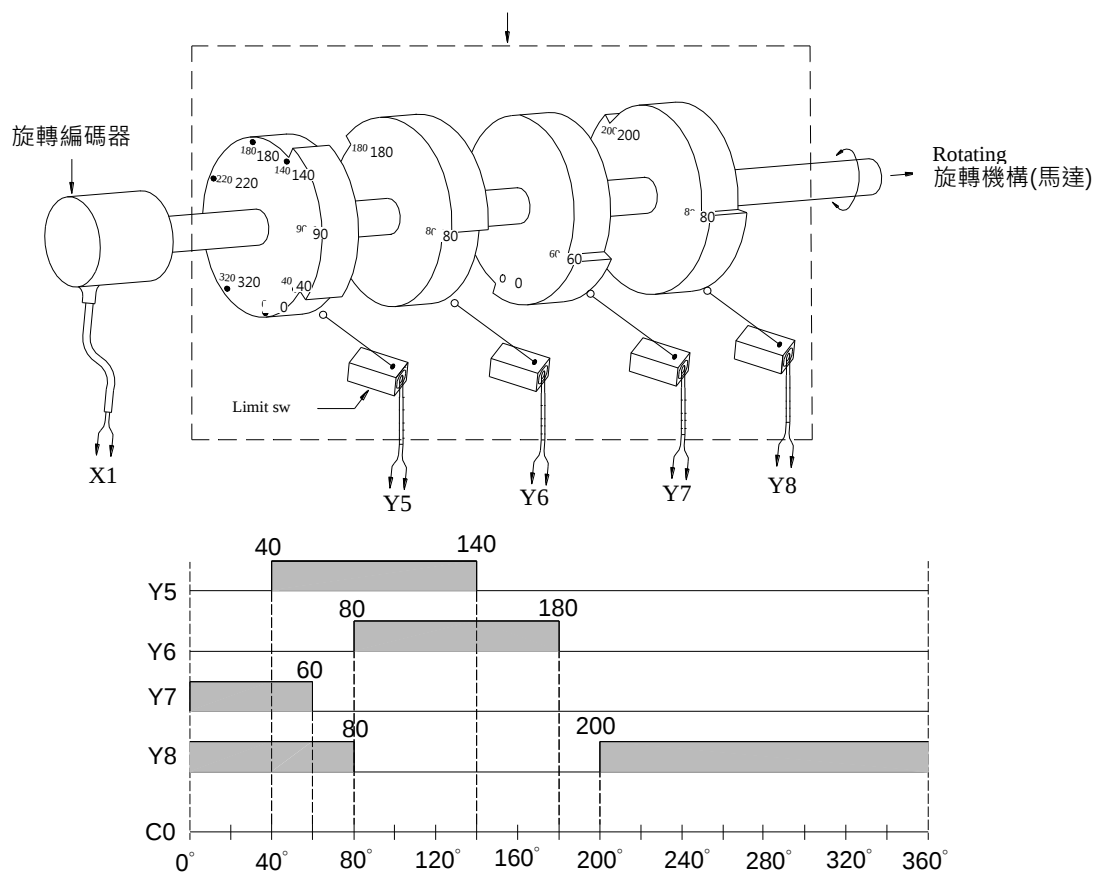
FUN112 DP BKCMP		區塊比較 (凸輪開關 DRUM) (BLOCK COMPARE)										FUN112 DP BKCMP				
指令說明																
階段圖符號 112DP.BKCMP 比較控制 — EN — Rs : Ts : L : D : — ERR — 限值錯誤								RS : 被比較之資料或其暫存器號碼 TS : 上下限值暫存器區塊之起頭號碼 L : 上下限值之組數 D : 存放比較結果之繼電器起頭號碼								
範圍 運算元	Y	M	S	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	Y0 Y1023	M0 M19583	S0 S3103	WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R3476 7	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數
	Rs			○	○	○	○	○	○	○	○	○	○	○	○	○
	Ts			○	○	○	○	○	○	○	○	○	○	○	○	
	L										○				○*	1~256
	D	○	○	○												
功能敘述																
● 當比較控制“EN”=1 或由 0→1 (P 指令) 時，將 Rs 之內容值逐一和由暫存器 Ts 開始之 L 組 16 或 32 (D 指令) 位元之上下限值 (由 T0 開始每相鄰之兩暫存器形成一組上下限設定值) 作比較，若 Rs 值落於該組設定值範圍內，則將比較結果之繼電器 D 中對應於該組之位元設為 1，否則為 0，直到比完所有 L 組上下限設定值為止。 ● 當 M9160=0 時，若上下限設定值中有任一組之上限值小於下限值，則限值錯誤旗號“ERR”設為 1，且該組比較輸出為 0。 ● 當 M9160=1 時，下限值可大於上限值之設定，而適用於 360°圓周運動，當有跨 0°之電子凸輪角度之應用。																
	上限值	下限值	上下限 比較	被比較值	比較 結果	輸出										
0	Ts1	Ts0	← →	Rs	← →	D0										
1	Ts3	Ts2	←		←	D1										
?	?	?	?		?	?										
L-1	TS2L-1	TS2L-2				DL-1										

FUN112 DP BKCMP	區塊比較 (凸輪開關 DRUM) (BLOCK COMPARE)	FUN112 DP BKCMP				
程式範例						
● 本指令實質上為一機械式之絕對是凸輪開關(DRUM)指令;其亦可置放於定時中斷程式中，配合 FUN74 (IMDIO)可得到較準確之電子凸輪角度輸出。						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN BKCMP(Rs:= C0, Ts:= R10, L:= 4, D:= Y5); END_IF Counter(Pulse:= X1, Clr:= C0, C:= 0, PV:=360, IsUp=> C0);</pre></td></tr></table>			階梯圖	ST		<pre>IF X0 THEN BKCMP(Rs:= C0, Ts:= R10, L:= 4, D:= Y5); END_IF Counter(Pulse:= X1, Clr:= C0, C:= 0, PV:=360, IsUp=> C0);</pre>
階梯圖	ST					
	<pre>IF X0 THEN BKCMP(Rs:= C0, Ts:= R10, L:= 4, D:= Y5); END_IF Counter(Pulse:= X1, Clr:= C0, C:= 0, PV:=360, IsUp=> C0);</pre>					
● 本程式範例指定 C0 值作為凸輪軸之旋轉角度(Rs)，並藉由本區塊比較指令，將 Rs 和 R10&R11、R12&R13、R14&R15、R16&R17 等 4 組(因 R=4)上下限設定值作比較，而得到 Y5~Y8 四個凸輪輸出點之輸出結果。 ● 程式中輸入接點 X1 為一旋轉角度感知器，係安裝於凸輪軸上，凸輪軸角度每轉動一度，X1 就會產生一個脈波，凸輪軸每轉動一周，X1 會產生 360 個脈波。						

FUN112 **DP**
BKCMPI區塊比較 (凸輪開關 DRUM)
(BLOCK COMPARE)FUN112 **DP**
BKCMPI

- 上圖程式配合一旋轉編碼器或其他轉動角度檢知裝置(直接連結至旋轉機構)即可組成和實際凸輪機械結構等效之機構裝置(如下圖虛線標示之機構)。同時透過上下限值之修改，您更可隨時改變凸輪觸動之角度範圍，為傳統凸輪機構所無法達成。

本指令範例所取
代之實際凸輪機械結構



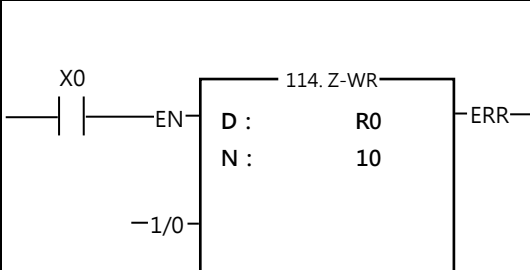
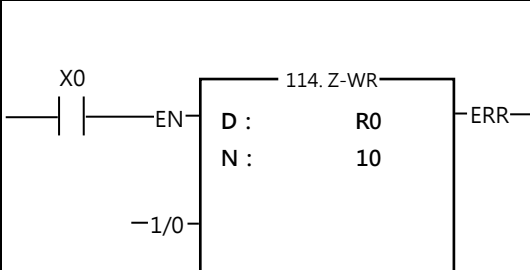
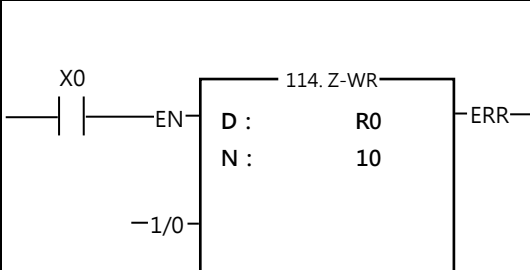
7-14-13 大小排序便利指令(SORTING)

FUN113 DP SORT		SORTING										FUN113 DP SORT			
指令說明															

FUN113 DP SORT	大小排序便利指令 (SORTING)	FUN113 DP SORT
階段圖 X0 — — EN 113. SORT S : R0 D : R10 L : 10 —ERR— A/D ST IF X0 THEN SORT(AD:= TRUE, S:= R0, D:= R10, L:= 10); END_IF		
● 上圖程式範例，將 R0 為起始之暫存器列表由小而大排序，並將排序結果存放到以 R10 為起始之暫存器列表中，如下圖結果。 S R01547 R12314 R27725 R30013 R45247 R51925 R66744 R75319 R89788 R92796 執行前狀態 X0 = ⇨ ↑ D R100013 R111547 R121925 R132314 R142796 R155247 R165319 R176744 R187725 R199788 執行結果		

7-14-14 區域寫入(ZONE WRITE)

FUN114 DP Z-WR		區域寫入 (ZONE WRITE)														FUN114 DP Z-WR			
指令說明																			
		階段圖符號 執行控制 — EN 寫入選擇 — 1/0 114P.Z-WR D : N : ERR — 長度錯誤 <td colspan="8"> D : 欲寫入或清除區域之起始位址 N : 欲寫入或清除區域之長度:1~511 D、N 可結合 V、Z、P0~P9 作間接定址應用 </td>										D : 欲寫入或清除區域之起始位址 N : 欲寫入或清除區域之長度:1~511 D、N 可結合 V、Z、P0~P9 作間接定址應用							
範圍 運算元		Y	M	SM	S	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	XR		
		Y0 Y1023	M0 M19583	M9120 M29599	S0 S3103	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0~P9		
D		○	○		○	○	○	○	○	○	○	○	○	○	○		○		
N											○			○	○	1-511	○		
功能敘述																			
		● 當執行控制“EN”=1 或由 0→1 (P 指令) 時，將以 D 為起始之 N 個暫存器，依據寫入選擇(1 或 0)，將其區域資料覆寫之。 ● 當 N 之長度設定不正確時(N=0 或 N>511)，則錯誤旗標“ ERR ”設定為 1。																	
程式範例 1		(程式範例當 X0 “ON” 時，將 R0~R9 寫入為 0)																	

FUN114 DP Z-WR	區域寫入 (ZONE WRITE)	FUN114 DP Z-WR																																														
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN ZoneWR(Val:= FALSE, D:= R0, N:= 10); END_IF</pre></td></tr></table>			階梯圖	ST		<pre>IF X0 THEN ZoneWR(Val:= FALSE, D:= R0, N:= 10); END_IF</pre>																																										
階梯圖	ST																																															
	<pre>IF X0 THEN ZoneWR(Val:= FALSE, D:= R0, N:= 10); END_IF</pre>																																															
<table><tr><td><table><tr><td>R0</td><td>1111</td></tr><tr><td>R1</td><td>2222</td></tr><tr><td>R2</td><td>3333</td></tr><tr><td>R3</td><td>4444</td></tr><tr><td>R4</td><td>5555</td></tr><tr><td>R5</td><td>6666</td></tr><tr><td>R6</td><td>7777</td></tr><tr><td>R7</td><td>8888</td></tr><tr><td>R8</td><td>9999</td></tr><tr><td>R9</td><td>1234</td></tr></table></td><td>X0 =  </td><td><table><tr><td>R0</td><td>0000</td></tr><tr><td>R1</td><td>0000</td></tr><tr><td>R2</td><td>0000</td></tr><tr><td>R3</td><td>0000</td></tr><tr><td>R4</td><td>0000</td></tr><tr><td>R5</td><td>0000</td></tr><tr><td>R6</td><td>0000</td></tr><tr><td>R7</td><td>0000</td></tr><tr><td>R8</td><td>0000</td></tr><tr><td>R9</td><td>0000</td></tr></table></td></tr><tr><td>執行前狀態</td><td></td><td>執行結果</td></tr></table>			<table><tr><td>R0</td><td>1111</td></tr><tr><td>R1</td><td>2222</td></tr><tr><td>R2</td><td>3333</td></tr><tr><td>R3</td><td>4444</td></tr><tr><td>R4</td><td>5555</td></tr><tr><td>R5</td><td>6666</td></tr><tr><td>R6</td><td>7777</td></tr><tr><td>R7</td><td>8888</td></tr><tr><td>R8</td><td>9999</td></tr><tr><td>R9</td><td>1234</td></tr></table>	R0	1111	R1	2222	R2	3333	R3	4444	R4	5555	R5	6666	R6	7777	R7	8888	R8	9999	R9	1234	X0 =  	<table><tr><td>R0</td><td>0000</td></tr><tr><td>R1</td><td>0000</td></tr><tr><td>R2</td><td>0000</td></tr><tr><td>R3</td><td>0000</td></tr><tr><td>R4</td><td>0000</td></tr><tr><td>R5</td><td>0000</td></tr><tr><td>R6</td><td>0000</td></tr><tr><td>R7</td><td>0000</td></tr><tr><td>R8</td><td>0000</td></tr><tr><td>R9</td><td>0000</td></tr></table>	R0	0000	R1	0000	R2	0000	R3	0000	R4	0000	R5	0000	R6	0000	R7	0000	R8	0000	R9	0000	執行前狀態		執行結果
<table><tr><td>R0</td><td>1111</td></tr><tr><td>R1</td><td>2222</td></tr><tr><td>R2</td><td>3333</td></tr><tr><td>R3</td><td>4444</td></tr><tr><td>R4</td><td>5555</td></tr><tr><td>R5</td><td>6666</td></tr><tr><td>R6</td><td>7777</td></tr><tr><td>R7</td><td>8888</td></tr><tr><td>R8</td><td>9999</td></tr><tr><td>R9</td><td>1234</td></tr></table>	R0	1111	R1	2222	R2	3333	R3	4444	R4	5555	R5	6666	R6	7777	R7	8888	R8	9999	R9	1234	X0 =  	<table><tr><td>R0</td><td>0000</td></tr><tr><td>R1</td><td>0000</td></tr><tr><td>R2</td><td>0000</td></tr><tr><td>R3</td><td>0000</td></tr><tr><td>R4</td><td>0000</td></tr><tr><td>R5</td><td>0000</td></tr><tr><td>R6</td><td>0000</td></tr><tr><td>R7</td><td>0000</td></tr><tr><td>R8</td><td>0000</td></tr><tr><td>R9</td><td>0000</td></tr></table>	R0	0000	R1	0000	R2	0000	R3	0000	R4	0000	R5	0000	R6	0000	R7	0000	R8	0000	R9	0000						
R0	1111																																															
R1	2222																																															
R2	3333																																															
R3	4444																																															
R4	5555																																															
R5	6666																																															
R6	7777																																															
R7	8888																																															
R8	9999																																															
R9	1234																																															
R0	0000																																															
R1	0000																																															
R2	0000																																															
R3	0000																																															
R4	0000																																															
R5	0000																																															
R6	0000																																															
R7	0000																																															
R8	0000																																															
R9	0000																																															
執行前狀態		執行結果																																														
程式範例 2	(程式範例當 X0 “ON” 時，將 M5~M11 清除為 0)																																															

階梯圖	ST
	<pre> IF X0 THEN ZoneWR(Val:= FALSE, D:= M5, N:= 7); END_IF </pre>

M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

X0 =

執行前狀態

↓

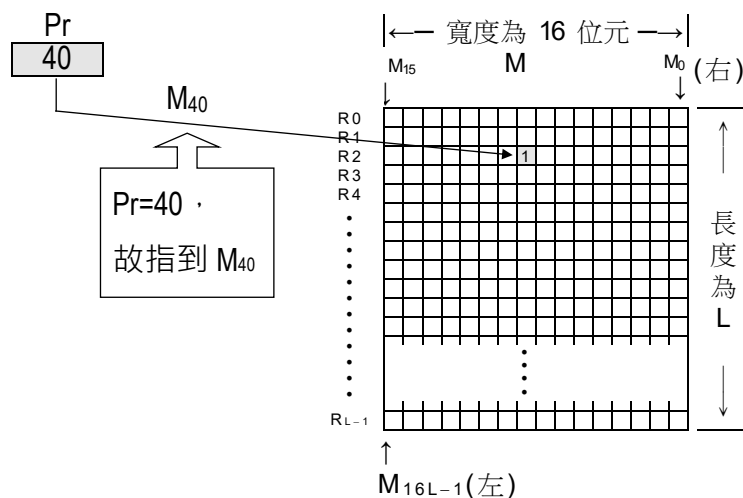
M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1

執行結果

矩 陣 (M A T R I X) 指 令

120. MAND	126. MBRD
121. MOR	127. MBWR
122. MXOR	128. MBSHF
123. MXNR	129. MBROT
124. MINV	130. MBCNT
125. MCMP	

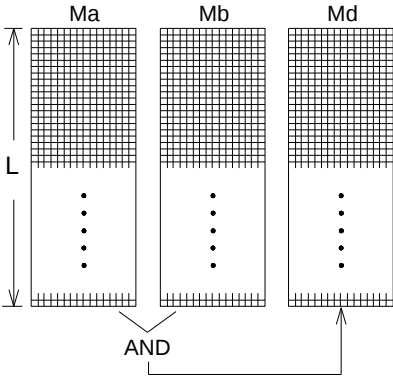
- 矩陣是 2 個以上連續之 16 位元暫存器所組成，組成矩陣之暫存器個數稱為矩陣之長度 L，一個矩陣共有 $L \times 16$ 個位元（點），其運算單位一次只有一個位元（點）。
- 矩陣指令是將 $16 \times L$ 個矩陣位元（序號由 $M_0 \sim M_{16L-1}$ ）當作一連串單點之集合，而自此集合中指定某一單點作運作，而不將之當作數值看待。
- 矩陣指令主要在處理單點對多點（矩陣）或多點對多點之狀態處理，如搬移、拷貝、比較、搜尋等，為極為方便和重要之應用指令。
- 在矩陣指令運作中，通常需要有一個 16 位元暫存器來指定矩陣中 16L 個單點之某個單點當作運算對象，此暫存器稱為矩陣之指標 PR（POINTER），其有效範圍為 $0 \sim 16L-1$ ，分別對應至矩陣中之位元 $M_0 \sim M_{16L-1}$ 。
- 矩陣運作中有左、右位移或旋轉，我們定義高序號者為左，低序號者為右，如下圖示。



7-15 矩陣指令 (FUN120 ~ 130)

7-15-1 矩陣邏輯及 (AND) 運算(MATRIX AND)

FUN120 P MAND		矩陣邏輯及 (AND) 運算 (MATRIX AND)												FUN120 P MAND	
指令說明															

FUN120 P MAND	矩陣邏輯及 (AND) 運算 (MATRIX AND)	FUN120 P MAND																																																																																																																																																																																																																																																																																																																																																																																																														
																																																																																																																																																																																																																																																																																																																																																																																																																
範例程式																																																																																																																																																																																																																																																																																																																																																																																																																
<table border="1" data-bbox="185 963 671 1370"> <thead> <tr> <th colspan="2">階梯圖</th></tr> </thead> <tbody> <tr> <td data-bbox="325 1115 357 1160">X0</td><td data-bbox="373 1115 405 1160"> </td></tr> <tr> <td data-bbox="405 1115 437 1160"> </td><td data-bbox="437 1115 469 1160"> </td></tr> <tr> <td data-bbox="469 1115 501 1160"> </td><td data-bbox="501 1115 533 1160"> </td></tr> <tr> <td data-bbox="533 1115 564 1160"> </td><td data-bbox="564 1115 596 1160"> </td></tr> <tr> <td data-bbox="596 1115 628 1160"> </td><td data-bbox="628 1115 660 1160"> </td></tr> <tr> <td data-bbox="660 1115 692 1160"> </td><td data-bbox="692 1115 724 1160"> </td></tr> <tr> <td data-bbox="724 1115 756 1160"> </td><td data-bbox="756 1115 788 1160"> </td></tr> <tr> <td data-bbox="788 1115 820 1160"> </td><td data-bbox="820 1115 852 1160"> </td></tr> <tr> <td data-bbox="852 1115 884 1160"> </td><td data-bbox="884 1115 916 1160"> </td></tr> <tr> <td data-bbox="916 1115 948 1160"> </td><td data-bbox="948 1115 979 1160"> </td></tr> <tr> <td data-bbox="979 1115 1011 1160"> </td><td data-bbox="1011 1115 1043 1160"> </td></tr> <tr> <td data-bbox="1043 1115 1075 1160"> </td><td data-bbox="1075 1115 1107 1160"> </td></tr> <tr> <td data-bbox="1107 1115 1139 1160"> </td><td data-bbox="1139 1115 1171 1160"> </td></tr> <tr> <td data-bbox="1171 1115 1203 1160"> </td><td data-bbox="1203 1115 1235 1160"> </td></tr> <tr> <td data-bbox="1235 1115 1267 1160"> </td><td data-bbox="1267 1115 1299 1160"> </td></tr> <tr> <td data-bbox="1299 1115 1331 1160"> </td><td data-bbox="1331 1115 1362 1160"> </td></tr> <tr> <td data-bbox="1362 1115 1394 1160"> </td><td data-bbox="1394 1115 1426 1160"> </td></tr> <tr> <td data-bbox="1426 1115 1458 1160"> </td><td data-bbox="1458 1115 1490 1160"> </td></tr> <tr> <td data-bbox="1490 1115 1522 1160"> </td><td data-bbox="1522 1115 1554 1160"> </td></tr> <tr> <td data-bbox="1554 1115 1586 1160"> </td><td data-bbox="1586 1115 1596 1160"> </td></tr> <tr> <td data-bbox="1618 1115 1596 1160"> </td><td data-bbox="1650 1115 1596 1160"> </td></tr> <tr> <td data-bbox="1682 1115 1596 1160"> </td><td data-bbox="1714 1115 1596 1160"> </td></tr> <tr> <td data-bbox="1746 1115 1596 1160"> </td><td data-bbox="1777 1115 1596 1160"> </td></tr> <tr> <td data-bbox="1809 1115 1596 1160"> </td><td data-bbox="1841 1115 1596 1160"> </td></tr> <tr> <td data-bbox="1873 1115 1596 1160"> </td><td data-bbox="1905 1115 1596 1160"> </td></tr> <tr> <td data-bbox="1937 1115 1596 1160"> </td><td data-bbox="1969 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2001 1115 1596 1160"> </td><td data-bbox="2033 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2065 1115 1596 1160"> </td><td data-bbox="2097 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2129 1115 1596 1160"> </td><td data-bbox="2160 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2192 1115 1596 1160"> </td><td data-bbox="2224 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2256 1115 1596 1160"> </td><td data-bbox="2288 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2320 1115 1596 1160"> </td><td data-bbox="2352 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2384 1115 1596 1160"> </td><td data-bbox="2416 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2448 1115 1596 1160"> </td><td data-bbox="2480 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2512 1115 1596 1160"> </td><td data-bbox="2544 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2575 1115 1596 1160"> </td><td data-bbox="2607 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2639 1115 1596 1160"> </td><td data-bbox="2671 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2703 1115 1596 1160"> </td><td data-bbox="2735 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2767 1115 1596 1160"> </td><td data-bbox="2799 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2831 1115 1596 1160"> </td><td data-bbox="2863 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2895 1115 1596 1160"> </td><td data-bbox="2927 1115 1596 1160"> </td></tr> <tr> <td data-bbox="2958 1115 1596 1160"> </td><td data-bbox="2990 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3022 1115 1596 1160"> </td><td data-bbox="3054 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3086 1115 1596 1160"> </td><td data-bbox="3118 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3150 1115 1596 1160"> </td><td data-bbox="3182 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3214 1115 1596 1160"> </td><td data-bbox="3246 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3278 1115 1596 1160"> </td><td data-bbox="3310 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3342 1115 1596 1160"> </td><td data-bbox="3373 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3405 1115 1596 1160"> </td><td data-bbox="3437 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3469 1115 1596 1160"> </td><td data-bbox="3501 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3533 1115 1596 1160"> </td><td data-bbox="3565 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3597 1115 1596 1160"> </td><td data-bbox="3629 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3661 1115 1596 1160"> </td><td data-bbox="3693 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3725 1115 1596 1160"> </td><td data-bbox="3756 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3788 1115 1596 1160"> </td><td data-bbox="3820 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3852 1115 1596 1160"> </td><td data-bbox="3884 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3916 1115 1596 1160"> </td><td data-bbox="3948 1115 1596 1160"> </td></tr> <tr> <td data-bbox="3980 1115 1596 1160"> </td><td data-bbox="4012 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4044 1115 1596 1160"> </td><td data-bbox="4076 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4108 1115 1596 1160"> </td><td data-bbox="4140 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4171 1115 1596 1160"> </td><td data-bbox="4203 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4235 1115 1596 1160"> </td><td data-bbox="4267 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4299 1115 1596 1160"> </td><td data-bbox="4331 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4363 1115 1596 1160"> </td><td data-bbox="4395 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4427 1115 1596 1160"> </td><td data-bbox="4459 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4491 1115 1596 1160"> </td><td data-bbox="4523 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4554 1115 1596 1160"> </td><td data-bbox="4586 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4618 1115 1596 1160"> </td><td data-bbox="4650 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4682 1115 1596 1160"> </td><td data-bbox="4714 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4746 1115 1596 1160"> </td><td data-bbox="4778 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4810 1115 1596 1160"> </td><td data-bbox="4842 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4874 1115 1596 1160"> </td><td data-bbox="4906 1115 1596 1160"> </td></tr> <tr> <td data-bbox="4938 1115 1596 1160"> </td><td data-bbox="4969 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5001 1115 1596 1160"> </td><td data-bbox="5033 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5065 1115 1596 1160"> </td><td data-bbox="5097 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5129 1115 1596 1160"> </td><td data-bbox="5161 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5193 1115 1596 1160"> </td><td data-bbox="5225 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5257 1115 1596 1160"> </td><td data-bbox="5289 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5321 1115 1596 1160"> </td><td data-bbox="5352 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5384 1115 1596 1160"> </td><td data-bbox="5416 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5448 1115 1596 1160"> </td><td data-bbox="5480 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5512 1115 1596 1160"> </td><td data-bbox="5544 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5576 1115 1596 1160"> </td><td data-bbox="5608 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5640 1115 1596 1160"> </td><td data-bbox="5672 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5704 1115 1596 1160"> </td><td data-bbox="5736 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5767 1115 1596 1160"> </td><td data-bbox="5799 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5831 1115 1596 1160"> </td><td data-bbox="5863 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5895 1115 1596 1160"> </td><td data-bbox="5927 1115 1596 1160"> </td></tr> <tr> <td data-bbox="5959 1115 1596 1160"> </td><td data-bbox="5991 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6023 1115 1596 1160"> </td><td data-bbox="6055 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6087 1115 1596 1160"> </td><td data-bbox="6119 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6150 1115 1596 1160"> </td><td data-bbox="6182 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6214 1115 1596 1160"> </td><td data-bbox="6246 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6278 1115 1596 1160"> </td><td data-bbox="6310 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6342 1115 1596 1160"> </td><td data-bbox="6374 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6406 1115 1596 1160"> </td><td data-bbox="6438 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6470 1115 1596 1160"> </td><td data-bbox="6502 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6534 1115 1596 1160"> </td><td data-bbox="6565 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6597 1115 1596 1160"> </td><td data-bbox="6629 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6661 1115 1596 1160"> </td><td data-bbox="6693 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6725 1115 1596 1160"> </td><td data-bbox="6757 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6789 1115 1596 1160"> </td><td data-bbox="6821 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6853 1115 1596 1160"> </td><td data-bbox="6885 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6917 1115 1596 1160"> </td><td data-bbox="6948 1115 1596 1160"> </td></tr> <tr> <td data-bbox="6980 1115 1596 1160"> </td><td data-bbox="7012 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7044 1115 1596 1160"> </td><td data-bbox="7076 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7108 1115 1596 1160"> </td><td data-bbox="7140 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7172 1115 1596 1160"> </td><td data-bbox="7204 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7236 1115 1596 1160"> </td><td data-bbox="7268 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7300 1115 1596 1160"> </td><td data-bbox="7332 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7363 1115 1596 1160"> </td><td data-bbox="7395 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7427 1115 1596 1160"> </td><td data-bbox="7459 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7491 1115 1596 1160"> </td><td data-bbox="7523 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7555 1115 1596 1160"> </td><td data-bbox="7587 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7619 1115 1596 1160"> </td><td data-bbox="7651 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7683 1115 1596 1160"> </td><td data-bbox="7715 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7746 1115 1596 1160"> </td><td data-bbox="7778 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7810 1115 1596 1160"> </td><td data-bbox="7842 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7874 1115 1596 1160"> </td><td data-bbox="7906 1115 1596 1160"> </td></tr> <tr> <td data-bbox="7938 1115 1596 1160"> </td><td data-bbox="7970 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8002 1115 1596 1160"> </td><td data-bbox="8034 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8066 1115 1596 1160"> </td><td data-bbox="8098 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8130 1115 1596 1160"> </td><td data-bbox="8161 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8193 1115 1596 1160"> </td><td data-bbox="8225 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8257 1115 1596 1160"> </td><td data-bbox="8289 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8321 1115 1596 1160"> </td><td data-bbox="8353 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8385 1115 1596 1160"> </td><td data-bbox="8417 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8449 1115 1596 1160"> </td><td data-bbox="8481 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8513 1115 1596 1160"> </td><td data-bbox="8544 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8576 1115 1596 1160"> </td><td data-bbox="8608 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8640 1115 1596 1160"> </td><td data-bbox="8672 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8704 1115 1596 1160"> </td><td data-bbox="8736 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8768 1115 1596 1160"> </td><td data-bbox="8800 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8832 1115 1596 1160"> </td><td data-bbox="8864 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8896 1115 1596 1160"> </td><td data-bbox="8928 1115 1596 1160"> </td></tr> <tr> <td data-bbox="8959 1115 1596 1160"> </td><td data-bbox="8991 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9023 1115 1596 1160"> </td><td data-bbox="9055 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9087 1115 1596 1160"> </td><td data-bbox="9119 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9151 1115 1596 1160"> </td><td data-bbox="9183 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9215 1115 1596 1160"> </td><td data-bbox="9247 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9279 1115 1596 1160"> </td><td data-bbox="9311 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9342 1115 1596 1160"> </td><td data-bbox="9374 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9406 1115 1596 1160"> </td><td data-bbox="9438 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9470 1115 1596 1160"> </td><td data-bbox="9502 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9534 1115 1596 1160"> </td><td data-bbox="9566 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9598 1115 1596 1160"> </td><td data-bbox="9630 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9662 1115 1596 1160"> </td><td data-bbox="9694 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9726 1115 1596 1160"> </td><td data-bbox="9757 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9789 1115 1596 1160"> </td><td data-bbox="9821 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9853 1115 1596 1160"> </td><td data-bbox="9885 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9917 1115 1596 1160"> </td><td data-bbox="9949 1115 1596 1160"> </td></tr> <tr> <td data-bbox="9981 1115 1596 1160"> </td><td data-bbox="10013 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10045 1115 1596 1160"> </td><td data-bbox="10077 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10109 1115 1596 1160"> </td><td data-bbox="10140 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10172 1115 1596 1160"> </td><td data-bbox="10204 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10236 1115 1596 1160"> </td><td data-bbox="10268 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10300 1115 1596 1160"> </td><td data-bbox="10332 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10364 1115 1596 1160"> </td><td data-bbox="10396 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10428 1115 1596 1160"> </td><td data-bbox="10460 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10492 1115 1596 1160"> </td><td data-bbox="10524 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10555 1115 1596 1160"> </td><td data-bbox="10587 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10619 1115 1596 1160"> </td><td data-bbox="10651 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10683 1115 1596 1160"> </td><td data-bbox="10715 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10747 1115 1596 1160"> </td><td data-bbox="10779 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10811 1115 1596 1160"> </td><td data-bbox="10843 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10875 1115 1596 1160"> </td><td data-bbox="10907 1115 1596 1160"> </td></tr> <tr> <td data-bbox="10938 1115 1596 1160"> </td><td data-bbox="10970 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11002 1115 1596 1160"> </td><td data-bbox="11034 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11066 1115 1596 1160"> </td><td data-bbox="11098 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11130 1115 1596 1160"> </td><td data-bbox="11162 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11194 1115 1596 1160"> </td><td data-bbox="11226 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11258 1115 1596 1160"> </td><td data-bbox="11290 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11322 1115 1596 1160"> </td><td data-bbox="11353 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11385 1115 1596 1160"> </td><td data-bbox="11417 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11449 1115 1596 1160"> </td><td data-bbox="11481 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11513 1115 1596 1160"> </td><td data-bbox="11545 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11577 1115 1596 1160"> </td><td data-bbox="11609 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11641 1115 1596 1160"> </td><td data-bbox="11673 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11705 1115 1596 1160"> </td><td data-bbox="11736 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11768 1115 1596 1160"> </td><td data-bbox="11800 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11832 1115 1596 1160"> </td><td data-bbox="11864 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11896 1115 1596 1160"> </td><td data-bbox="11928 1115 1596 1160"> </td></tr> <tr> <td data-bbox="11960 1115 1596 1160"> </td><td data-bbox="11992 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12024 1115 1596 1160"> </td><td data-bbox="12056 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12088 1115 1596 1160"> </td><td data-bbox="12120 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12151 1115 1596 1160"> </td><td data-bbox="12183 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12215 1115 1596 1160"> </td><td data-bbox="12247 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12279 1115 1596 1160"> </td><td data-bbox="12311 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12343 1115 1596 1160"> </td><td data-bbox="12375 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12407 1115 1596 1160"> </td><td data-bbox="12439 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12471 1115 1596 1160"> </td><td data-bbox="12503 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12534 1115 1596 1160"> </td><td data-bbox="12566 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12598 1115 1596 1160"> </td><td data-bbox="12630 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12662 1115 1596 1160"> </td><td data-bbox="12694 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12726 1115 1596 1160"> </td><td data-bbox="12758 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12790 1115 1596 1160"> </td><td data-bbox="12822 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12854 1115 1596 1160"> </td><td data-bbox="12886 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12918 1115 1596 1160"> </td><td data-bbox="12949 1115 1596 1160"> </td></tr> <tr> <td data-bbox="12981 1115 1596 1160"> </td><td data-bbox="13013 1115 1596 1160"> </td></tr> <tr> </tr></tbody></table>	階梯圖		X0																																																																																																																																																																																																																																																																																																																																																																																																													
階梯圖																																																																																																																																																																																																																																																																																																																																																																																																																
X0																																																																																																																																																																																																																																																																																																																																																																																																																

7-15-2 矩陣邏輯或 (OR) 運算(MATRIX OR)

FUN121 P MOR		矩陣邏輯或 (OR) 運算 (MATRIX OR)												FUN121 P MOR	
指令說明															

FUN121 P MOR	矩陣邏輯或 (OR) 運算 (MATRIX OR)	FUN121 P MOR

7-15-3 矩陣邏輯互斥或 (XOR) 運算(MATRIX EXCLUSIVE OR)

FUN122 P MXOR		矩陣邏輯互斥或 (XOR) 運算 (MATRIX EXCLUSIVE OR)												FUN122 P MXOR	
指令說明															

FUN122 P MXOR	矩陣邏輯互斥或 (XOR) 運算 (MATRIX EXCLUSIVE OR)	FUN122 P MXOR
Ma Mb Md L ⋮ ⋮ ⋮ ⋮ XOR		
程式範例		
階段圖 X0 EN 122P.MXOR Ma : R 0 Mb : R 10 Md : R 20 L : 5	ST IF X0 THEN MXOR(Ma:= R0, Mb:= R10, Md:= R20, L:= 5); END_IF	
Ma₁₅ Ma Ma₀ R0 0000000000000000 R1 111111111100000000 R2 000000000011111111 R3 000000000000000000 R4 111111111111111111 Ma₇₉ Ma₆₄ Mb₁₅ Mb Mb₀ R10 111111111111111111 R11 00000000000111111111 R12 00000000000111111111 R13 00000000000000000000 R14 11111111111111111111 Mb₇₉ Mb₆₄ 執行前狀態 Md₁₅ Md Md₀ R20 11111111111111111111 R21 11111111111111111111 R22 00000000000000000000 R23 00000000000000000000 R24 00000000000000000000 Md₇₉ Md₆₄ 執行結果		

7-15-4 矩陣互容或 (XNR) 運算 (MATRIX EXCLUSIVE OR)

FUN123 P MXNR	矩陣互容或 (XNR) 運算 (MATRIX EXCLUSIVE NOR)													FUN123 P MXNR																																																																																																							
指令說明																																																																																																																					
階梯圖符號 運算控制 — EN 123P.MXNR — Ma : Mb : Md : L : 		Ma : 來源矩陣 a 之起頭暫存器號碼 Mb : 來源矩陣 b 之起頭暫存器號碼 Md : 存放結果之目的矩陣起頭暫存器號碼 L : 矩陣 (Ma · Mb · Md) 之長度 Ma · Mb · Md 可結合 V、Z、P0~P9 作間接定址應用																																																																																																																			
<table border="1" style="border-collapse: collapse; text-align: center; font-size: 0.8em;"> <tr> <th rowspan="3" style="writing-mode: vertical-rl; transform: rotate(180deg);">運算元</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr> <tr> <td>WX0</td><td>WY0</td><td>WM0</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td>2</td><td>V、Z</td></tr> <tr> <td>WX1008</td><td>WY1008</td><td>WM25584</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td>256</td><td>P0~P9</td></tr> <tr> <td>Ma</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr> <tr> <td>Mb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr> <tr> <td>Md</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr> <tr> <td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td>○</td><td></td></tr> </table>		運算元	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	2	V、Z	WX1008	WY1008	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	256	P0~P9	Ma	○	○	○	○	○	○	○	○	○	○	○	○		○	Mb	○	○	○	○	○	○	○	○	○	○	○	○		○	Md		○	○	○	○	○	○		○	○*	○*	○		○	L							○				○*	○	○														
運算元	WX		WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																																						
	WX0		WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	2	V、Z																																																																																																						
	WX1008	WY1008	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	256	P0~P9																																																																																																							
Ma	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																																							
Mb	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																																							
Md		○	○	○	○	○	○		○	○*	○*	○		○																																																																																																							
L							○				○*	○	○																																																																																																								
功能敘述																																																																																																																					
		● 當運算控制 “EN” =1 或由 0→1 (P 指令) 時，將長度為 L 之兩來源矩陣 Ma 與 Mb 整個作邏輯 XNR 運算 (兩位元相同則結果為 1，否則為 0) 後，再將結果存到長度同為 L 之目的矩陣 Md 去 (相同序號之位元作 XNR，例如 Ma0=0，Mb0=1，則 Md0=0；Ma1=0，Mb1=0 則 Md1=1；..... 一直 XNR 至 Ma16L-1 和 Mb16L-1 止)。																																																																																																																			

FUN123 P MXNR	矩陣互容或 (XNR) 運算 (MATRIX EXCLUSIVE NOR)	FUN123 P MXNR
Ma Mb Md L ⋮ ⋮ ⋮ ⋮ XNR		
程式範例		
X0 EN 123P.MXNR Ma : R 0 Mb : R 10 Md : R 10 L : 5	ST IF X0 THEN MXNR(Ma:= R0, Mb:= R10, Md:= R10, L:= 5); END_IF	
Ma15 Ma Ma0 0000000000000000 1111111111000000 0000000001111111 0000000000000000 1111111111111111 Ma79 Ma64 Mb15 Mb Mb0 R1011111111111111 R1100000000000111 R1200000000000111 R130000000000000000 R141111111111111111 Mb79 Mb64 Md15 Md Md0 R200000000000000000 R210000000000000000 R221111111111111111 R231111111111111111 R241111111111111111 Md79 Md64 執行前狀態 執行結果		

7-15-5 矩陣倒相 (MATRIX INVERSE)

FUN124 P MINV		矩陣倒相 (MATRIX INVERSE)												FUN124 P MINV			
指令說明																	

FUN124 P MINV	矩陣倒相 (MATRIX INVERSE)	FUN124 P MINV
範例程式		

階段圖	ST
X0 EN 124P.MINV Ms : R 0 Md : R 0 L : 5	<pre>IF X0 THEN MINV(Ms:= R0, Md:= R0, L:= 5); END_IF</pre>

●

左圖程式範例，當 X0 由 0→1 時，將 R0~R4 所組成之矩陣反相後存回自己（因本例 Ms 和 Md 為同一矩陣）。所獲得之結果如下圖右。

Ms₁₅

Ms

Ms₀

R0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
R2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

↑ Ms₇₉

Ms₆₄ ↑

執行前狀態

Md₁₅

Md

Md₀

R0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R2	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
R3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

↑ Md₇₉

Md₆₄ ↑

執行結果

7-15-6 矩陣位元位移 (MATRIX BIT SHIFT)

FUN128 P MBSHF	矩陣位元位移 (MATRIX BIT SHIFT)												FUN128 P MBSHF																																																																																											
指令說明																																																																																																								
階梯圖符號 128P.MBSHF 位移控制 — EN — Ms : OTB — 移出位元 填補位元 — INB — Md : 左/右方向 — L/R — L :								Ms：來源矩陣之起頭暫存器號碼 Md：目的矩陣之起頭暫存器號碼 L：矩陣 (Ms 和 Md) 長度 Ms、Md 可結合 V、Z、P0~P9 作間接定址應用																																																																																																
<table><tr><th>圖 號</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0</td><td>WY0</td><td>WM0</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td>2</td><td>V、Z</td></tr><tr><td></td><td>WX1008</td><td>WY1008</td><td>WM25584</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td>256</td><td>P0~P9</td></tr><tr><td>Ms</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>Md</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td>○</td></tr><tr><td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○*</td><td>○</td><td>○</td><td></td></tr></table>															圖 號	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	2	V、Z		WX1008	WY1008	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	256	P0~P9	Ms	○	○	○	○	○	○	○	○	○	○	○	○		○	Md		○	○	○	○	○	○		○	○*	○*	○		○	L											○*	○	○	
圖 號	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																										
	WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0	2	V、Z																																																																																										
	WX1008	WY1008	WM25584	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999	256	P0~P9																																																																																										
Ms	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																										
Md		○	○	○	○	○	○		○	○*	○*	○		○																																																																																										
L											○*	○	○																																																																																											
功能敘述																																																																																																								
● 當位移控制“EN”=1 或由 0→1 (P 指令) 時，將矩陣 Ms 整個取出向左 (L/R=1 時) 或向右 (L/R=0 時) 位移一個位置，因位移而騰出之空位 (左移時為 M0，右移時為 M16L-1) 則以填補位元“INB”之狀態填補。而因位移而擠出之位元 (左移時為 M16L-1，右移時為 M0) 狀態則送到移出位元“OTB”去，然後再將此位移過之矩陣結果填入目的矩陣 Md 去。																																																																																																								

FUN128 P MBSHF	矩陣位元位移 (MATRIX BIT SHIFT)	FUN128 P MBSHF				
Ms INB OTB Shift left 1 bit ⇒ Md L Ms OTB INB Shift right 1 bit ⇒ Md L						
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td> X0 EN X0 INB L/R 128P.MBSHF Ms : R 0 Md : R 0 L : 5 OTB </td><td><pre>IF X0 THEN MBSHF(INB:= X0, L_R:= TRUE, Ms:= R0, Md:= R0, L:= 5, OTB=> M0); END_IF</pre></td></tr></table>			階梯圖	ST	X0 EN X0 INB L/R 128P.MBSHF Ms : R 0 Md : R 0 L : 5 OTB	<pre>IF X0 THEN MBSHF(INB:= X0, L_R:= TRUE, Ms:= R0, Md:= R0, L:= 5, OTB=> M0); END_IF</pre>
階梯圖	ST					
X0 EN X0 INB L/R 128P.MBSHF Ms : R 0 Md : R 0 L : 5 OTB	<pre>IF X0 THEN MBSHF(INB:= X0, L_R:= TRUE, Ms:= R0, Md:= R0, L:= 5, OTB=> M0); END_IF</pre>					
● 上圖程式範例之 Ms 和 Md 為同一矩陣之範例，當 X0 由 0→1 動作時，將 Ms 整個取出作左移 (因 L/R=1) 一位元後，再存回 Md 而得到如下圖右之結果。 Ms₁₅ Ms Ms₀ R0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 R1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 R2 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 R3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 R4 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 Ms₇₉ Ms₆₄ X0=1 Md₁₅ Md Md₀ R0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 R1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 R2 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 R3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 R4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 Md₇₉ Md₆₄ 執行前狀態 執行結果						

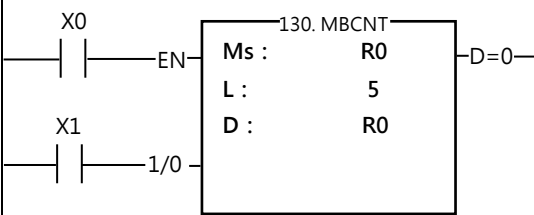
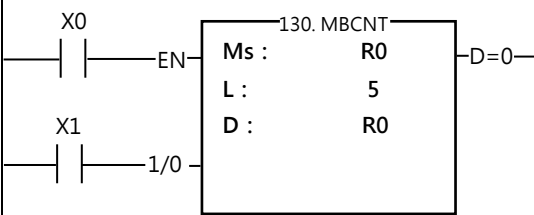
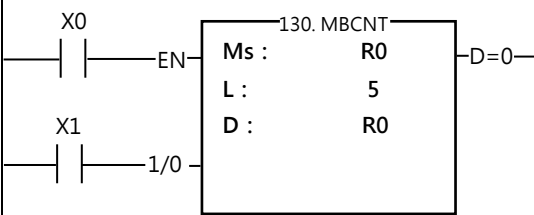
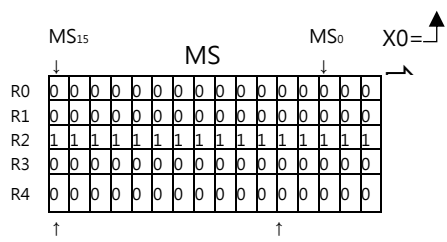
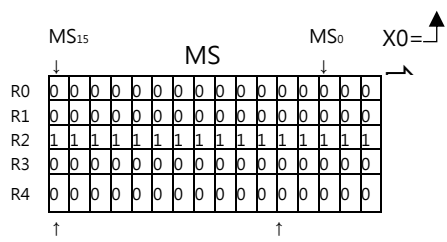
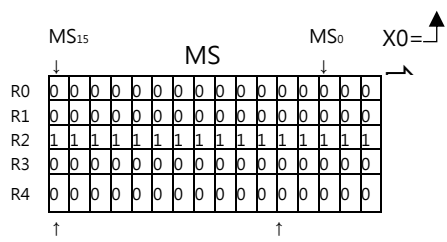
7-15-7 矩陣位元旋轉 (MATRIX BIT ROTATE)

FUN129 P MBROT		矩陣位元旋轉 (MATRIX BIT ROTATE)												FUN129 P MBROT	
指令說明															

FUN129 P MBROT	矩陣位元旋轉 (MATRIX BIT ROTATE)	FUN129 P MBROT
L/R=1 Ms Rotate left 1 bit OTB Md L/R=0 Ms Shift right 1 bit OTB Md		
程式範例		
階梯圖 X0 EN 129P.MBROT Ms: R 0 Md: R 0 L : 5 OTB	ST IF X0 THEN MBROT(L_R:= FALSE, Ms:= R0, Md:= R0, L:= 5, OTB=> M0); END_IF	
Ms Ms₁₅ Ms₀ R0 R1 R2 R3 R4 Ms₇₉ Ms₆₄ X0=1 Md Md₁₅ Md₀ R0 R1 R2 R3 R4 Md₇₉ Md₆₄ OTB 0		
執行前狀態 執行結果		

7-15-8 矩陣位元狀態數量計算 (MATRIX BIT STATUS COUNT)

FUN130 P MBCNT	矩陣位元狀態數量計算 (MATRIX BIT STATUS COUNT)													FUN130 P MBCNT																																																																										
指令說明																																																																																								
階梯圖符號 計算控制 — EN — 130P.MBCNT — D=0 — 結果為0 1或0選擇 — 1/0 — Ms : L : D :							MS : 矩陣之起頭暫存器號碼 L : 矩陣之長度 D : 存放數量結果之暫存器號碼 MS 可結合 V、Z、P0~P9 作間接定址應用																																																																																	
<table><tr><th rowspan="3">運算元 範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td>2 256</td><td>V、Z P0~P9</td></tr><tr><td>MS</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td>○</td><td></td></tr><tr><td>D</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td></td></tr></table>															運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	2 256	V、Z P0~P9	MS	○	○	○	○	○	○	○	○	○	○	○	○		○	L							○				○*	○	○		D		○	○	○	○	○	○		○	○*	○*	○		
運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																										
	WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	2 256	V、Z P0~P9																																																																										
	MS	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																									
L							○				○*	○	○																																																																											
D		○	○	○	○	○	○		○	○*	○*	○																																																																												
功能敘述																																																																																								
● 當計算控制 “EN” =1 或由 0→1 (P 指令) 時，統計矩陣 MS 之 16L 個位元中，所有狀態為 “1” (ON) 之位元總數目 (1 或 0 選擇輸入 “1/0” =1 時) 或所有狀態為 “0” (OFF) 之位元總數目 (1 或 0 選擇輸入 “1/0” =0 時)。再將統計所得之數目結果存到 D 所指定之暫存器去，若此數目值為 0，則將結果為 0 旗號 “D=0” 設為 1。																																																																																								

FUN130 P MBCNT	矩陣位元狀態數量計算 (MATRIX BIT STATUS COUNT)	FUN130 P MBCNT							
範例程式									
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF X0 THEN MBCNT(OnOff:= X1, Ms:= R0, L:= 5, D:= R0); END_IF</pre></td></tr></table>			階梯圖	ST		<pre>IF X0 THEN MBCNT(OnOff:= X1, Ms:= R0, L:= 5, D:= R0); END_IF</pre>			
階梯圖	ST								
	<pre>IF X0 THEN MBCNT(OnOff:= X1, Ms:= R0, L:= 5, D:= R0); END_IF</pre>								
● 上圖程式範例分別將 X1 設為 0 (統計狀態為 0 之位元數) 及設為 1 (統計狀態為 1 者) 兩種條件，再分別於此兩條件下使 X0 由 0→1 各一次，可獲得如下圖右①、②兩個執行結果。									
<table><tr><td> 來源矩陣狀態 </td><td><table><tr><td>①</td><td> D R20 64 X1=0 時 計算狀態為 0 之位元數 </td><td><table><tr><td>②</td><td> D R20 16 X1=1 時 計算狀態為 1 之位元數 </td></tr></table></td></tr></table></td></tr></table>			 來源矩陣狀態	<table><tr><td>①</td><td> D R20 64 X1=0 時 計算狀態為 0 之位元數 </td><td><table><tr><td>②</td><td> D R20 16 X1=1 時 計算狀態為 1 之位元數 </td></tr></table></td></tr></table>	①	D R20 64 X1=0 時 計算狀態為 0 之位元數	<table><tr><td>②</td><td> D R20 16 X1=1 時 計算狀態為 1 之位元數 </td></tr></table>	②	D R20 16 X1=1 時 計算狀態為 1 之位元數
 來源矩陣狀態	<table><tr><td>①</td><td> D R20 64 X1=0 時 計算狀態為 0 之位元數 </td><td><table><tr><td>②</td><td> D R20 16 X1=1 時 計算狀態為 1 之位元數 </td></tr></table></td></tr></table>	①	D R20 64 X1=0 時 計算狀態為 0 之位元數	<table><tr><td>②</td><td> D R20 16 X1=1 時 計算狀態為 1 之位元數 </td></tr></table>	②	D R20 16 X1=1 時 計算狀態為 1 之位元數			
①	D R20 64 X1=0 時 計算狀態為 0 之位元數	<table><tr><td>②</td><td> D R20 16 X1=1 時 計算狀態為 1 之位元數 </td></tr></table>	②	D R20 16 X1=1 時 計算狀態為 1 之位元數					
②	D R20 16 X1=1 時 計算狀態為 1 之位元數								

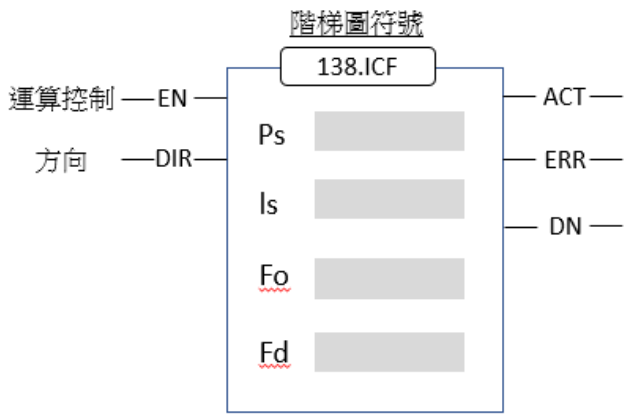
7-16 NC 定位控制指令一 (FUN137 ~ 144)

7-16-1 中斷定角指令(ICA)

FUN137 ICA		中斷定角 (INTERRUPT CONSTANT ANGLE)												FUN137 ICA	
指令說明															

FUN137 ICA	中斷定角 (INTERRUPT CONSTANT ANGLE)	FUN137 ICA
功能敘述		
1. 定位軸最多可以控制到 PSO7，唯實際能夠控制的最大軸號依主機機型而有差異。 2. 目標工作速度最大頻率依照主機機型而有 100K 與 200K 兩種差異。 3. 在通用與進階運動型主機中，外界輸入 X 點 8~15 將保留給運動功能使用而不支援於此指令。 4. 外界輸入點不需額外於 I/O 組態 中之中斷設定特別配置。指令執行時會自動進行相關設定。		

7-16-2 中斷定長指令(ICF)

FUN138 ICF	中斷定長 (INTERRUPT CONSTANT FEED)	FUN138 ICF
指令說明		
		Ps : 第幾組 Pulse 輸出 (0 ~ 7) 0 : Y0 & Y1 1 : Y2 & Y3 2 : Y4 & Y5 3 : Y6 & Y7 4 : MQ: Y16 & Y17; ME、MS、MA: Y8 & Y9 5 : MQ: Y18 & Y19; ME、MS、MA: Y10 & Y11 6 : MQ: Y20 & Y21; ME、MS、MA: Y12 & Y13 7 : MQ Y22 & Y23; ME、MS、MA: Y14 & Y15 Ls : MQ: X0~X23 or Y0~Y23; MA: X0~X15 or Y0~Y15; ME、MS: X0~X7 or Y0~Y7 Fo : 目標工作速度(1~2147483647) Fd : 中斷捕捉後輸出之脈波移動量(1~2147483647)

範圍 運算元	X	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K
	主 機 上 之 Xn	WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	
	Pw													0 ~ 7
	Ls	○												
	Fo	○	○	○	○	○	○	○	○	○	○	○	○	○
Fd	○	○	○	○	○	○	○	○	○	○	○	○	○	1- 2147483647
FUN138 ICF		中斷定長 (INTERRUPT CONSTANT FEED)											FUN138 ICF	
功能敘述														
1. 定位軸最多可以控制到 PSO7 ，唯實際能夠控制的 最大軸號 依主機機型而有差異。 2. 目標工作速度 最大頻率 依照主機機型而有 100K 與 200K 兩種差異。 3. 在通用與進階運動型主機中，外界輸入 X 點 8~15 將保留給運動功能使用而不支援於此指令。 4. 外界輸入點不需額外於 I/O 組態 中之 中斷設定 特別配置。指令執行時會自動進行相關設定。														

7-16-3 高速脈波寬度調變(HSPWM)

FUN139 HSPWM		高速脈波寬度調變 (HIGH SPEED PULSE WIDTH MODULATION)										FUN139 HSPWM			
指令說明															

FUN139 HSPWM	高速脈波寬度調變 (HIGH SPEED PULSE WIDTH MODULATION)	FUN139 HSPWM
功能敘述		
● 第 0 點(Y0)與第 1 點(Y2)PWM 輸出, 其解析度必須相同, 同為 1/100 或 1/1000; 而輸出頻率參數設定也必須相同, 只有輸出寬度(PULSE WIDTH)可以不一樣; 同理第 2 點(Y4)與第 3 點(Y6)PWM 輸出亦如上述規定。 ● 當執行控制“EN”=1 時, 本指令所指定之輸出點將依下列公式所決定之頻率以指定之脈波寬度輸出。 1. $f_{pwm} = \frac{184320}{(P_n + 1)}$ 當 Rs(解析度)設定為 1/100 時 2. $f_{pwm} = \frac{18432}{(P_n + 1)}$ 當 Rs(解析度)設定為 1/1000 時		

程式範例一

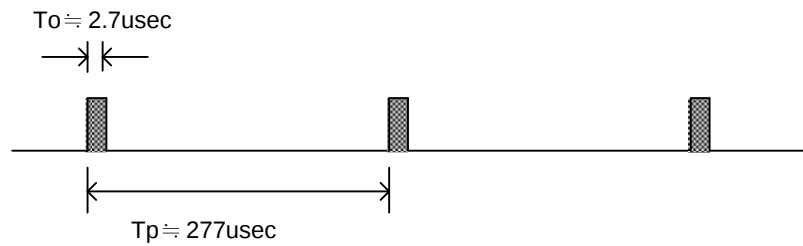
假設 Pn(輸出頻率參數)設為 50，Rs(解析度)=0 則

$$f_{pwm} = \frac{184320}{(50+1)} = 3614.117 \dots \approx 3.6 \text{ KHz}$$

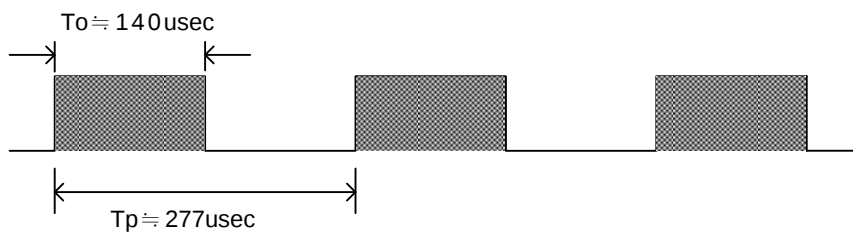
$$T(\text{週期}) = \frac{1}{f_{pwm}} \approx 277 \mu\text{s}$$

因為解析度為 1/100，所以 OR(輸出寬度)若為 1 則 $T_o \approx 2.7 \mu\text{s}$ ，OR(輸出寬度)若為 50 則 $T_o \approx 140 \mu\text{s}$ 。圖形如下：

(1). Pn(輸出頻率參數)=50，Rs=0(解析度=1/100)，OR(輸出寬度)=1 之輸出波形：



(2). Pn(輸出頻率參數)=50，Rs=0(解析度=1/100)，OR(輸出寬度)=50 之輸出波形：



FUN139 HSPWM	高速脈波寬度調變 (HIGH SPEED PULSE WIDTH MODULATION)	FUN139 HSPWM
程式範例二	假設 Pn(輸出頻率參數)設為 200 · Rs(解析度)=1 則。	
$f_{pwm} = \frac{18432}{(200 + 1)} \div 91.7\text{Hz} ; T(\text{週期}) = \frac{1}{f_{pwm}} \div 10.9\text{mS}$ 因為 Rs(解析度)為 1/1000 · 所以 OR(輸出寬度)若為 10 則 To$\div$109uS · OR(輸出寬度)若為 800 則 To$\div$8.72mS · 圖形如下: (1).Pn(輸出頻率參數)=200 · Rs=1(解析度=1/1000) · OR(輸出寬度)=10 之輸出波形： To $\div$ 109uS Tp $\div$ 10.90msec (2).Pn(輸出頻率參數)=200 · Rs=1(解析度=1/1000) · OR(輸出寬度)=800 之輸出波形： To $\div$ 8.72msec Tp $\div$ 10.90msec		

7-16-4 高速脈波輸出(HSPSO)

FUN140 HSPSO	高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO																									
指令說明																											
階段圖符號 140.HSPSO 執行控制 — EN 暫停輸出 — PAU 放棄輸出 — ABT Ps : SR : WR : ACT — 脈波輸出中 ERR — 指令執行錯誤 DN — 定位點完成		Ps : 第幾組 Pulse 輸出 (0 ~ 7) 0 : Y0 & Y1 1 : Y2 & Y3 2 : Y4 & Y5 3 : Y6 & Y7 4 : Y8 & Y9 5 : Y10 & Y11 6 : Y12 & Y13 7 : Y14 & Y15 SR : 定位程式起始暫存器 WR : 指令運作起始暫存器，共佔用 7 個暫存器，其它程式不可重覆使用																									
<table><tr><th> 範圍 運算元 </th><th>HR</th><th>DR</th><th>ROR</th><th>K</th></tr><tr><td></td><td> R0 R34767 </td><td> D0 D7999 </td><td> R43224 R47319 </td><td></td></tr><tr><td>Ps</td><td></td><td></td><td></td><td>0~7</td></tr><tr><td>SR</td><td> ○ </td><td> ○ </td><td> ○ </td><td></td></tr><tr><td>WR</td><td> ○ </td><td> ○ </td><td> ○* </td><td></td></tr></table>			範圍 運算元	HR	DR	ROR	K		R0 R34767	D0 D7999	R43224 R47319		Ps				0~7	SR	○	○	○		WR	○	○	○*	
範圍 運算元	HR	DR	ROR	K																							
	R0 R34767	D0 D7999	R43224 R47319																								
Ps				0~7																							
SR	○	○	○																								
WR	○	○	○*																								
功能敘述																											
1. 定位軸最多可以控制到 PSO7，唯實際能夠控制的最大軸號依主機機型而有差異。 2. FUN140 (HSPSO) 指令之 NC 定位程式是以文字之程式書寫方式來編輯；每一定位點我們稱一步（含輸出頻率、動作行程、轉移條件），一個 FUN140 最多可編 250 步定位點，每一步定位點需佔用 9 個暫存器。 3. 將定位程式存在暫存器最大好處是，如果結合人機作機台操控設定，則可將定位程式存入人機，更換模具時，可直接由人機操作存取該副模具之定位程式。																											

FUN140 HSPSO	高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO
	4. 當執行控制輸入“EN”=1 時，如 Ps0 ~ Ps7 沒有被其它 FUN140 指令佔用 (Ps0=M9183、Ps1=M9184、Ps2=M9185、Ps3=M9186、Ps4=M9191、Ps5=M9192、Ps6=M9193、Ps7=M9194 之狀態為 ON，反之則為 OFF)，則由下一步定位點開始執行 (如已至最後一步，則重新由第 1 步開始執行)；如 Ps0 ~ 7 被其它 FUN140 指令佔用，則待佔用之 FUN140 釋出控制權，本指令取得定位控制之脈波輸出權。 5. 當執行控制“EN”=0 時，馬上停止脈波輸出。 6. 當暫停輸出“PAU”=1，且執行控制“EN”先前為 1 時，則暫停脈波輸出； 當暫停輸出“PAU”=0，而執行控制“EN”仍為 1 時，繼續輸出未完成之脈波數。 7. 當放棄輸出“ABT”=1 時，馬上停止脈波輸出。(下一次當執行控制輸入“EN”=1 時，重新由第一步定位點開始執行) 8. 當脈波輸出中，輸出指示“ACT”ON。 9. 當指令執行錯誤時，輸出指示“ERR”ON。(錯誤代碼存放於錯誤碼暫存器) 10. 當每一步定位點完成時，輸出指示“DN”ON。	

FUN140 HSPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO																				
*** 務必設定 Pulse 輸出之工作模式 (不設定時，Y0~Y15 當作一般輸出) 為 U/D，P/R 或 A/B 等三種模式之一，Pulse 輸出才能正常輸出。 U/D 模式：Y0 (Y2，Y4，Y6，Y8，Y10，Y12，Y14) 送出上數脈波。 Y1 (Y3，Y5，Y7，Y9，Y11，Y13，Y15) 送出下數脈波。 P/R 模式：Y0 (Y2，Y4，Y6，Y8，Y10，Y12，Y14) 送出脈波。 Y1 (Y3，Y5，Y7，Y9，Y11，Y13，Y15) 送出方向信號；ON=上數，OFF=下數。 A/B 模式：Y0 (Y2，Y4，Y6，Y8，Y10，Y12，Y14) 送出 A 相脈波。 Y1 (Y3，Y5，Y7，Y9，Y11，Y13，Y15) 送出 B 相脈波。 Pulse Output 輸出極性可選擇 Normal ON 或 Normal OFF。 ※FUN140 不支援 脈波模式(U)，如需使用，請以 FUN139 做使用 【介面處理信號】 <table><tr><td rowspan="2">M9183</td><td>ON：代表 Ps0 Ready。</td></tr><tr><td>OFF：代表 Ps0 作動中。</td></tr><tr><td rowspan="2">M9184</td><td>ON：代表 Ps1 Ready。</td></tr><tr><td>OFF：代表 Ps1 作動中。</td></tr><tr><td rowspan="2">M9185</td><td>ON：代表 Ps2 Ready。</td></tr><tr><td>OFF：代表 Ps2 作動中。</td></tr><tr><td rowspan="2">M9186</td><td>ON：代表 Ps3 Ready。</td></tr><tr><td>OFF：代表 Ps3 作動中。</td></tr><tr><td>M9187</td><td>ON：Ps0 完成最後一步。</td></tr><tr><td>M9188</td><td>ON：Ps1 完成最後一步。</td></tr><tr><td>M9189</td><td>ON：Ps2 完成最後一步。</td></tr><tr><td>M9190</td><td>ON：Ps3 完成最後一步。</td></tr></table>			M9183	ON：代表 Ps0 Ready。	OFF：代表 Ps0 作動中。	M9184	ON：代表 Ps1 Ready。	OFF：代表 Ps1 作動中。	M9185	ON：代表 Ps2 Ready。	OFF：代表 Ps2 作動中。	M9186	ON：代表 Ps3 Ready。	OFF：代表 Ps3 作動中。	M9187	ON：Ps0 完成最後一步。	M9188	ON：Ps1 完成最後一步。	M9189	ON：Ps2 完成最後一步。	M9190	ON：Ps3 完成最後一步。
M9183	ON：代表 Ps0 Ready。																					
	OFF：代表 Ps0 作動中。																					
M9184	ON：代表 Ps1 Ready。																					
	OFF：代表 Ps1 作動中。																					
M9185	ON：代表 Ps2 Ready。																					
	OFF：代表 Ps2 作動中。																					
M9186	ON：代表 Ps3 Ready。																					
	OFF：代表 Ps3 作動中。																					
M9187	ON：Ps0 完成最後一步。																					
M9188	ON：Ps1 完成最後一步。																					
M9189	ON：Ps2 完成最後一步。																					
M9190	ON：Ps3 完成最後一步。																					

FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HPSO																																													
<table><tr><td rowspan="2">M9191</td><td>ON : 代表 Ps4 Ready 。</td></tr><tr><td>OFF : 代表 Ps4 作動中 。</td></tr><tr><td rowspan="2">M9192</td><td>ON : 代表 Ps5 Ready 。</td></tr><tr><td>OFF : 代表 Ps5 作動中 。</td></tr><tr><td rowspan="2">M9193</td><td>ON : 代表 Ps6 Ready 。</td></tr><tr><td>OFF : 代表 Ps6 作動中 。</td></tr><tr><td rowspan="2">M9194</td><td>ON : 代表 Ps7 Ready 。</td></tr><tr><td>OFF : 代表 Ps7 作動中 。</td></tr><tr><td>M9195</td><td>ON : Ps4 完成最後一步 。</td></tr><tr><td>M9196</td><td>ON : Ps5 完成最後一步 。</td></tr><tr><td>M9197</td><td>ON : Ps6 完成最後一步 。</td></tr><tr><td>M9198</td><td>ON : Ps7 完成最後一步 。</td></tr></table>			M9191	ON : 代表 Ps4 Ready 。	OFF : 代表 Ps4 作動中 。	M9192	ON : 代表 Ps5 Ready 。	OFF : 代表 Ps5 作動中 。	M9193	ON : 代表 Ps6 Ready 。	OFF : 代表 Ps6 作動中 。	M9194	ON : 代表 Ps7 Ready 。	OFF : 代表 Ps7 作動中 。	M9195	ON : Ps4 完成最後一步 。	M9196	ON : Ps5 完成最後一步 。	M9197	ON : Ps6 完成最後一步 。	M9198	ON : Ps7 完成最後一步 。																									
M9191	ON : 代表 Ps4 Ready 。																																														
	OFF : 代表 Ps4 作動中 。																																														
M9192	ON : 代表 Ps5 Ready 。																																														
	OFF : 代表 Ps5 作動中 。																																														
M9193	ON : 代表 Ps6 Ready 。																																														
	OFF : 代表 Ps6 作動中 。																																														
M9194	ON : 代表 Ps7 Ready 。																																														
	OFF : 代表 Ps7 作動中 。																																														
M9195	ON : Ps4 完成最後一步 。																																														
M9196	ON : Ps5 完成最後一步 。																																														
M9197	ON : Ps6 完成最後一步 。																																														
M9198	ON : Ps7 完成最後一步 。																																														
<table><tr><th>Ps No.</th><th>目前輸出頻率</th><th>目前 PS 位置</th><th>剩餘待輸出 PS 數</th><th>錯誤碼</th></tr><tr><td>Ps0</td><td>DR35328</td><td>DR35336</td><td>DR35344</td><td>R35320</td></tr><tr><td>Ps1</td><td>DR35330</td><td>DR35338</td><td>DR35346</td><td>R35321</td></tr><tr><td>Ps2</td><td>DR35332</td><td>DR35340</td><td>DR35348</td><td>R35322</td></tr><tr><td>Ps3</td><td>DR35334</td><td>DR35342</td><td>DR35350</td><td>R35323</td></tr><tr><td>Ps4</td><td>DR35655</td><td>DR35663</td><td>DR35671</td><td>R35647</td></tr><tr><td>Ps5</td><td>DR35657</td><td>DR35665</td><td>DR35673</td><td>R35648</td></tr><tr><td>Ps6</td><td>DR35659</td><td>DR35667</td><td>DR35675</td><td>R35649</td></tr><tr><td>Ps7</td><td>DR35661</td><td>DR35669</td><td>DR35677</td><td>R35650</td></tr></table>			Ps No.	目前輸出頻率	目前 PS 位置	剩餘待輸出 PS 數	錯誤碼	Ps0	DR35328	DR35336	DR35344	R35320	Ps1	DR35330	DR35338	DR35346	R35321	Ps2	DR35332	DR35340	DR35348	R35322	Ps3	DR35334	DR35342	DR35350	R35323	Ps4	DR35655	DR35663	DR35671	R35647	Ps5	DR35657	DR35665	DR35673	R35648	Ps6	DR35659	DR35667	DR35675	R35649	Ps7	DR35661	DR35669	DR35677	R35650
Ps No.	目前輸出頻率	目前 PS 位置	剩餘待輸出 PS 數	錯誤碼																																											
Ps0	DR35328	DR35336	DR35344	R35320																																											
Ps1	DR35330	DR35338	DR35346	R35321																																											
Ps2	DR35332	DR35340	DR35348	R35322																																											
Ps3	DR35334	DR35342	DR35350	R35323																																											
Ps4	DR35655	DR35663	DR35671	R35647																																											
Ps5	DR35657	DR35665	DR35673	R35648																																											
Ps6	DR35659	DR35667	DR35675	R35649																																											
Ps7	DR35661	DR35669	DR35677	R35650																																											

FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HPSO
R35324 : Ps0 每步結束時之步號。 R35325 : Ps1 每步結束時之步號。 R35326 : Ps2 每步結束時之步號。 R35327 : Ps3 每步結束時之步號。 R35651 : Ps4 每步結束時之步號。 R35652 : Ps5 每步結束時之步號。 R35653 : Ps6 每步結束時之步號。 R35654 : Ps7 每步結束時之步號。		
● 定位程式格式：		
SR：定位程式起始暫存器，說明如下：		
SR	A55AH	；有效定位程式，起始暫存器旗標必為 A55AH
SR+1	總步數	；1 ~ 250
SR+2		}
SR+3		
SR+4		
SR+5		
SR+6		
SR+7		} 第一步定位點程式 (每步佔用 9 個暫存器)
SR+8		
SR+9		
SR+10		
	· · ·	}
		}
		第 N 步定位點程式
SR+N×9+2		

FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HPSO														
指令運作工作暫存器說明： WR 為起始暫存器 <table><tr><td>WR+0</td><td>目前工作或停留步數</td></tr><tr><td>WR+1</td><td>工作旗標</td></tr><tr><td>WR+2</td><td>系統使用</td></tr><tr><td>WR+3</td><td>系統使用</td></tr><tr><td>WR+4</td><td>系統使用</td></tr><tr><td>WR+5</td><td>系統使用</td></tr><tr><td>WR+6</td><td>系統使用</td></tr></table> WR+0：如果本指令正執行中，則該暫存器之內容值即為正執行之步數（ 1 ~ N ）； 如果本指令未執行中，則該暫存器之內容值代表目前所停留之步數。 WR+1：B0 ~ B7，總步數。 B8 = 保留。 B9 = 保留。 B10 = 保留。 B11 = 保留。 B12 = ON，脈波輸出中（輸出指示“ACT”）。 B13 = ON，指令執行錯誤（輸出指示“ERR”）。 B14 = ON，一步定位點完成（輸出指示“DN”）。 ***每一步定位點完成後，輸出指示“DN”會一直維持 ON；如果不想讓輸出指示一直維持 ON，則在每步定位點完成後，利用輸出指示線圈所控制之上緣接點指令將 WR+1 暫存器內容清除為 0，即可達成。			WR+0	目前工作或停留步數	WR+1	工作旗標	WR+2	系統使用	WR+3	系統使用	WR+4	系統使用	WR+5	系統使用	WR+6	系統使用
WR+0	目前工作或停留步數															
WR+1	工作旗標															
WR+2	系統使用															
WR+3	系統使用															
WR+4	系統使用															
WR+5	系統使用															
WR+6	系統使用															

FUN140 HSPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO
錯誤碼 0 : 無錯誤 1 : 參數 0 錯誤 2 : 參數 1 錯誤 3 : 參數 2 錯誤 4 : 參數 3 錯誤 5 : 參數 4 錯誤 6 : 參數 5 錯誤 7 : 參數 6 錯誤 8 : 參數 7 錯誤 9 : 參數 8 錯誤 10 : 參數 9 錯誤 13 : 參數 12 錯誤 14 : 參數 13 錯誤 15 : 參數 14 錯誤 30 : 速度設定變數號碼錯誤 31 : 速度設定值錯誤 32 : 行程設定變數號碼錯誤 33 : 行程設定值錯誤 34 : 不合法定位程式 35 : 步數長度錯誤 36 : 超過最大步數 37 : 最高頻率錯誤 38 : 起始 / 停止頻率錯誤 39 : 移動量補正值太大 40 : 移動量超出範圍 41 : DRVC 內不允許 ABS 定址 42 : DRVC 不可銜接 DRVZ 命令 43 : 驅動命令碼錯誤 50 : DRVZ 工作模式錯誤 51 : 近點 DOG 輸入點錯誤 52 : 零點信號 PG0 輸入點錯誤 53 : 歸零清除 CLR 輸出點錯誤 54 : I/O 組態錯誤 (Ex :以不支援的模式使用, 如 FUN140 配脈波模式; 應改為 FUN139 配脈波模式) 60 : 不合法補間驅動命令		
執行 FUN141 時 可能之錯誤碼		
執行 FUN140 與 執行 FUN147 時 可能之錯誤碼		
註：錯誤指示暫存器內容會保持最近一次之錯誤碼，如需確認不再有錯誤發生，可將錯誤指示暫存器內容清除為 0，祇要其內容一直維持 0 不變，即代表無錯誤發生。		

FUN140 HSPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO
-----------------	-------------------------	-----------------

使用 UperLogic 編輯伺服命令表格

於專案視窗中點選伺服命令表格: 專案名稱

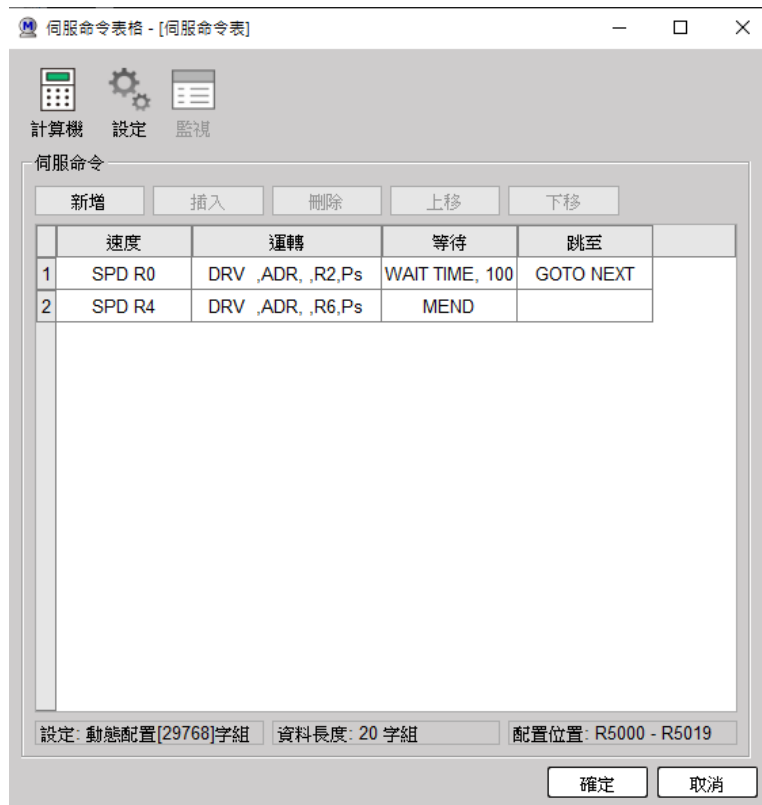
表格編輯

伺服命令表格 → 按右鍵後，點選“新增伺服命令表格”。

FUN140 HSPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO
-----------------	-------------------------	-----------------

- 表格類別：固定為“ 伺服命令表格” 。
- 表格名稱：可為伺服命令表格輸入一容易辨識之名稱，方便日後修改或除錯用。
- 表格起始位置：伺服命令指令(FUN140)所用之資料表格起始暫存器 SR 之起始位置。

※ 有關伺服命令表格之建立，請參考 UperLogic 介面手冊第 7 章(表格輸入及編輯)，或按一下指令，在按下 Z(快捷鍵)進行建立。

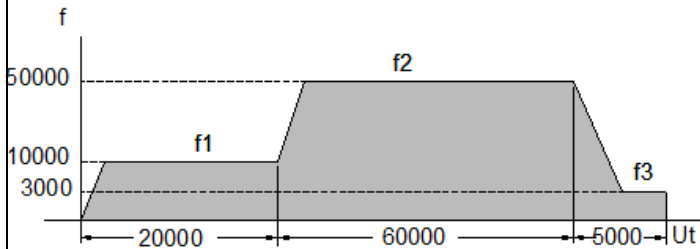


FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HPSO
● 為了讓定位程式易編、易讀、易維護，在 FUN140 指令下我們衍生出下列幾個相關指令，使用者在 Uperlogic 下即可直接編輯、修改定位程式。 ● 定位衍生指令列表如下：		
指令	運算元	說明
SPD	XXXXXX 或 Rxxxxx 或 Dxxxxx	● 脈波輸出之頻率或速度 (FUN141 參數 0=0 時為速度；參數 0=1 或 2 時為頻率，系統內定為頻率)；運算元可直接輸入常數或變數 (Rxxxx · Dxxxx)；當運算元為變數時共需使用二個暫存器，例如 D10，即代表 D10 (Low Word) 與 D11 (High Word) 為頻率或速度設定值。 ● 當選擇使用速度設定時，系統會自動將速度設定值轉換為頻率輸出。 ● 頻率輸出範圍：1≤頻率輸出≤100000 or 200000Hz *** 當頻率設定值=0 時，本指令等待設定值不等於 0 時，才會執行定位脈波輸出。

FUN140 HSPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO			
	<table border="1" data-bbox="181 483 1410 1581"> <tr> <td data-bbox="181 483 277 1581">DRV</td><td data-bbox="277 483 684 1581"> ADR · + · XXXXXXXX · Ut ADR · + · XXXXXXXX · Ps ADR · - · XXXXXXXX · Ut ADR · - · XXXXXXXX · Ps ADR · · XXXXXXXX · Ut ADR · · -XXXXXXX · Ut ADR · · XXXXXXXX · Ps ADR · · -XXXXXXX · Ps ADR · + · Rxxxx · Ut ADR · + · Rxxxx · Ps ADR · - · Rxxxx · Ut ADR · - · Rxxxx · Ps ADR · · Rxxxx · Ut ADR · · Rxxxx · Ps ADR · + · Dxxxx · Ut ADR · + · Dxxxx · Ps ADR · - · Dxxxx · Ut ADR · - · Dxxxx · Ps ADR · · Dxxxx · Ut ADR · · Dxxxx · Ps ABS · · XXXXXXXX · Ut ABS · · -XXXXXXX · Ut ABS · · XXXXXXXX · Ps ABS · · -XXXXXXX · Ps ABS · · Rxxxx · Ut ABS · · Rxxxx · Ps ABS · · Dxxxx · Ut ABS · · Dxxxx · Ps </td><td data-bbox="684 483 1410 1581"> 脈波輸出量 (FUN141 參數 0=1 時，單位為 Ps；參數 0=0 或 2 時，單位為 mm · Deg · Inch；系統內定為 Ps) 當脈波輸出單位不為 Ps 時，系統會根據 FUN141 之參數 1，2，3 設定轉換為 Ps 數輸出。 DRV 指令運算元共有四項，說明如下： 第一項運算元：定位座標選擇 ADR 或 ABS： ADR · 相對值座標定位。 ABS · 絕對值座標定位。 第二項運算元：運轉方向選擇 (相對值座標有效) ' + ' 或 ' - '： ' + '，正轉或上數。 ' - '，反轉或下數。 或 ' ' '： ' ' '，運轉方向由行程設定值決定 (正值：正轉；負值：反轉) 第三項運算元：行程設定值 (脈波輸出量)。 XXXXXXXX 可直接輸入常數或變數 (Rxxxx， 或 Dxxxx)；當使用變數時共需使用二個 -XXXXXXX 暫存器，例如 R0，即代表 R0 (Low 或 Rxxxx Word) 與 R1 (High Word) 為行程 或 Dxxxx 設定值。 *** 當行程設定值=0，不管座標為 ADR 或 ABS，代表連續運轉不停。 行程設定範圍：-99999999 ≤ 行程設定值 ≤ 99999999 第四項運算元：行程設定值解析度 Ut 或 Ps：Ut 時，解析度為一個單位；(由 FUN141 之參數 0，3 決定) Ps 時，強制解析度為一個 Ps。 </td></tr> </table>	DRV	ADR · + · XXXXXXXX · Ut ADR · + · XXXXXXXX · Ps ADR · - · XXXXXXXX · Ut ADR · - · XXXXXXXX · Ps ADR · · XXXXXXXX · Ut ADR · · -XXXXXXX · Ut ADR · · XXXXXXXX · Ps ADR · · -XXXXXXX · Ps ADR · + · Rxxxx · Ut ADR · + · Rxxxx · Ps ADR · - · Rxxxx · Ut ADR · - · Rxxxx · Ps ADR · · Rxxxx · Ut ADR · · Rxxxx · Ps ADR · + · Dxxxx · Ut ADR · + · Dxxxx · Ps ADR · - · Dxxxx · Ut ADR · - · Dxxxx · Ps ADR · · Dxxxx · Ut ADR · · Dxxxx · Ps ABS · · XXXXXXXX · Ut ABS · · -XXXXXXX · Ut ABS · · XXXXXXXX · Ps ABS · · -XXXXXXX · Ps ABS · · Rxxxx · Ut ABS · · Rxxxx · Ps ABS · · Dxxxx · Ut ABS · · Dxxxx · Ps	脈波輸出量 (FUN141 參數 0=1 時，單位為 Ps；參數 0=0 或 2 時，單位為 mm · Deg · Inch；系統內定為 Ps) 當脈波輸出單位不為 Ps 時，系統會根據 FUN141 之參數 1，2，3 設定轉換為 Ps 數輸出。 DRV 指令運算元共有四項，說明如下： 第一項運算元：定位座標選擇 ADR 或 ABS： ADR · 相對值座標定位。 ABS · 絕對值座標定位。 第二項運算元：運轉方向選擇 (相對值座標有效) ' + ' 或 ' - '： ' + '，正轉或上數。 ' - '，反轉或下數。 或 ' ' '： ' ' '，運轉方向由行程設定值決定 (正值：正轉；負值：反轉) 第三項運算元：行程設定值 (脈波輸出量)。 XXXXXXXX 可直接輸入常數或變數 (Rxxxx， 或 Dxxxx)；當使用變數時共需使用二個 -XXXXXXX 暫存器，例如 R0，即代表 R0 (Low 或 Rxxxx Word) 與 R1 (High Word) 為行程 或 Dxxxx 設定值。 *** 當行程設定值=0，不管座標為 ADR 或 ABS，代表連續運轉不停。 行程設定範圍：-99999999 ≤ 行程設定值 ≤ 99999999 第四項運算元：行程設定值解析度 Ut 或 Ps：Ut 時，解析度為一個單位；(由 FUN141 之參數 0，3 決定) Ps 時，強制解析度為一個 Ps。	
DRV	ADR · + · XXXXXXXX · Ut ADR · + · XXXXXXXX · Ps ADR · - · XXXXXXXX · Ut ADR · - · XXXXXXXX · Ps ADR · · XXXXXXXX · Ut ADR · · -XXXXXXX · Ut ADR · · XXXXXXXX · Ps ADR · · -XXXXXXX · Ps ADR · + · Rxxxx · Ut ADR · + · Rxxxx · Ps ADR · - · Rxxxx · Ut ADR · - · Rxxxx · Ps ADR · · Rxxxx · Ut ADR · · Rxxxx · Ps ADR · + · Dxxxx · Ut ADR · + · Dxxxx · Ps ADR · - · Dxxxx · Ut ADR · - · Dxxxx · Ps ADR · · Dxxxx · Ut ADR · · Dxxxx · Ps ABS · · XXXXXXXX · Ut ABS · · -XXXXXXX · Ut ABS · · XXXXXXXX · Ps ABS · · -XXXXXXX · Ps ABS · · Rxxxx · Ut ABS · · Rxxxx · Ps ABS · · Dxxxx · Ut ABS · · Dxxxx · Ps	脈波輸出量 (FUN141 參數 0=1 時，單位為 Ps；參數 0=0 或 2 時，單位為 mm · Deg · Inch；系統內定為 Ps) 當脈波輸出單位不為 Ps 時，系統會根據 FUN141 之參數 1，2，3 設定轉換為 Ps 數輸出。 DRV 指令運算元共有四項，說明如下： 第一項運算元：定位座標選擇 ADR 或 ABS： ADR · 相對值座標定位。 ABS · 絕對值座標定位。 第二項運算元：運轉方向選擇 (相對值座標有效) ' + ' 或 ' - '： ' + '，正轉或上數。 ' - '，反轉或下數。 或 ' ' '： ' ' '，運轉方向由行程設定值決定 (正值：正轉；負值：反轉) 第三項運算元：行程設定值 (脈波輸出量)。 XXXXXXXX 可直接輸入常數或變數 (Rxxxx， 或 Dxxxx)；當使用變數時共需使用二個 -XXXXXXX 暫存器，例如 R0，即代表 R0 (Low 或 Rxxxx Word) 與 R1 (High Word) 為行程 或 Dxxxx 設定值。 *** 當行程設定值=0，不管座標為 ADR 或 ABS，代表連續運轉不停。 行程設定範圍：-99999999 ≤ 行程設定值 ≤ 99999999 第四項運算元：行程設定值解析度 Ut 或 Ps：Ut 時，解析度為一個單位；(由 FUN141 之參數 0，3 決定) Ps 時，強制解析度為一個 Ps。			

FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)				FUN140 HPSO
指 令	運	算	元	明	明
DRVC	ADR, +, 或 或 ABS, -, 或 Dxxxx,	xxxxx, 或 Rxxxx, 或 Dxxxx,	Ut 或 Ps	DRVC 之使用及運算元說明與 DRV 指令相同。 ***DRVC 用來做連續多段速度變換控制 (最多 8 段) ***DRVC 所構成之連續多段速度變換控制，祇有第一個 DRVC 指令能夠使用絕對值座標定位。 ***DRVC 之運轉方向決定祇能由 ' + ' 或 ' - ' 決定 ***連續多段速度控制之方向 (正、反轉) 祇能由第一段之方向決定，後面指令之方向運算元無效；亦即多段速度變換控制祇能同方向。 例：連續三段速度控制 <pre> 001 ; 脈波頻率=10KHz SPD 10000 ; 正轉 20000 個單位 DRVC ADR · + · 20000 · Ut GOTO NEXT ; 脈波頻率=50 KHz 002 ; 正轉 60000 個單位 SPD 50000 DRVC ADR · + · 60000 · Ut ; 脈波頻率=3 KHz GOTO NEXT ; 正轉 5000 個單位 003 ; 等待 X0 ON · 重新由第 SPD 3000 一步執行 DRV ADR · + · 5000 · Ut WAIT X0 GOTO 1 ***注意：DRVC 指令之個數必須比連續之段數少一，亦即最後一段必須使用 DRV 指令。 </pre>	

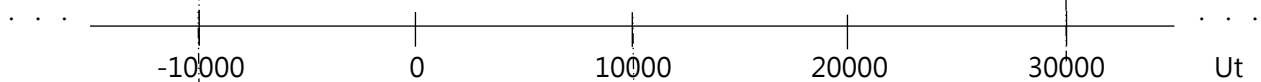
FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HPSO
----------------	-----------------------	----------------

指 令	運 算 元	說 明
		上例為連續三段速度控制，DRVC 指令使用二個，第三段必須使用 DRV 指令。 上例圖示： 
DRVZ	MD1	DRVZ 為原點復歸便利指令，支援 MD1 原點復歸方式。 ***有關 DRVZ 之 MD1 之詳細說明，請參閱 9.6 節(機械原點復歸)。

註: 相對值座標定位(ADR)與絕對值座標定位(ABS)比較說明

欲由位置 30000 移至-10000 時，程式寫法：

← DRV ADR,-,40000,Ut 或 DRV ABS, ,-10000,Ut



欲由位置-10000 移至 10000 時，程式寫法：DRV ADR,+,20000,Ut 或

DRV ABS, ,10000,Ut

FUN140 HSPSO		單軸高速脈波輸出 (含衍生定位指令)	FUN140 HSPSO
指令	運算元	說明	
WAIT	TIME · XXXXX 或 Rxxxxx 或 Dxxxxx 或 X0~X1023 或 Y0~Y1023 或 M0~M29599 或 S0~S3103	當脈波輸出完成時，欲執行下一步之等待指令； 運算元共有五種，說明如下： Time：等待時間（單位為 0.01 秒），可直接輸入常數或變數（Rxxxx 或 Dxxxx）；當計時到，則執行 GOTO 所指之步數。 X0~X1023：等待輸入接點信號 ON，執行 GOTO 所指之步數。 Y0~Y1023：等待輸出接點信號 ON，執行 GOTO 所指之步數。 M0~M29599：等待內部繼電器 ON，執行 GOTO 所指之步數。 S0~S3103：等待步進繼電器 ON，執行 GOTO 所指之步數。	
ACT	TIME · XXXXX 或 Rxxxxx 或 Dxxxxx	脈波輸出 ACT 所描述之作動時間後，立即執行 GOTO 所指之步數；亦即脈波輸出一段時間後，立即執行下一步。作動時間（單位為 0.01 秒），可直接輸入常數或變數（Rxxxxx 或 Dxxxxx）；當作動時間到，則執行 GOTO 所指之步數。	
EXT	X0~X1023 或 Y0~Y1023 或 M0~M29599 或 S0~S999	○ 外部觸發指令，當脈波輸出中（脈波個數尚未送完），如果外部觸發信號作動（ON），則立即執行 GOTO 所指之步數；如果脈波輸出已完成，外部觸發信號尚未作動，則與 WAIT 指令相同，信號（ON）時，才會執行 GOTO 所指之步數。	
GOTO	NEXT 或 1~N 或 Rxxxxx 或 Dxxxxx	當 WAIT · ACT · EXT 等指令條件滿足時，利用 GOTO 指令描述將要執行之步數。 NEXT：代表執行下一步 1~N：執行第幾步 Rxxxxx：欲執行之步數存放於暫存器 Rxxxxx Dxxxxx：欲執行之步數存放於暫存器 Dxxxxx	
MEND		定位程式結束	

FUN140 HPSO	單軸高速脈波輸出 (含衍生定位指令)	FUN140 HPSO
● 定位程式之書寫： 定位程式在編輯之前，必須要先完成 FUN140 指令，並在 FUN140 指令指定欲存放定位程式之起始暫存器號碼；編輯定位程式時，會將新編之定位程式存入所指定之暫存器區塊，每編 1 個定位點（稱為 1 步）會佔用 9 個暫存器，如有 N 個定位點（N 步），共佔用 $N \times 9 + 2$ 個暫存器。 *** 注意：儲存定位程式之暫存器不可被重複使用！ 程式格式與範例： <pre> 001 SPD 5000 ; 脈波頻率=5KHz DRV ADR,+,10000,Ut ; 正轉 10000 個單位 WAIT TIME,100 ; 等待 1 秒 GOTO NEXT ; 執行下一步 002 SPD R1000 ; 脈波頻率存放在 DR1000 (R1001 與 R1000) DRV ADR,+,D100,Ut ; 正轉行程存放在 DD100 (D101 與 D100) WAIT TIME,R500 ; 等待時間存放在 R500 GOTO NEXT ; 執行下一步 003 SPD R1002 ; 脈波頻率存放在 DR1002 (R1003 與 R1002) DRV ADR,-,D102,Ut ; 反轉行程存放在 DD102 (D103 與 D102) EXT X0 ; 外部觸發 X0 (減速點) ON 時，馬上執行下一步 GOTO NEXT 004 SPD 2000 ; 脈波頻率=2K HZ DRV ADR,-,R4072,Ps ; 繼續執行第 3 步未完成之 PS 數 (存放於 DR4072) WAIT X1 ; 等待 X1 ON 時 GOTO 1 ; 執行第一步 </pre>		

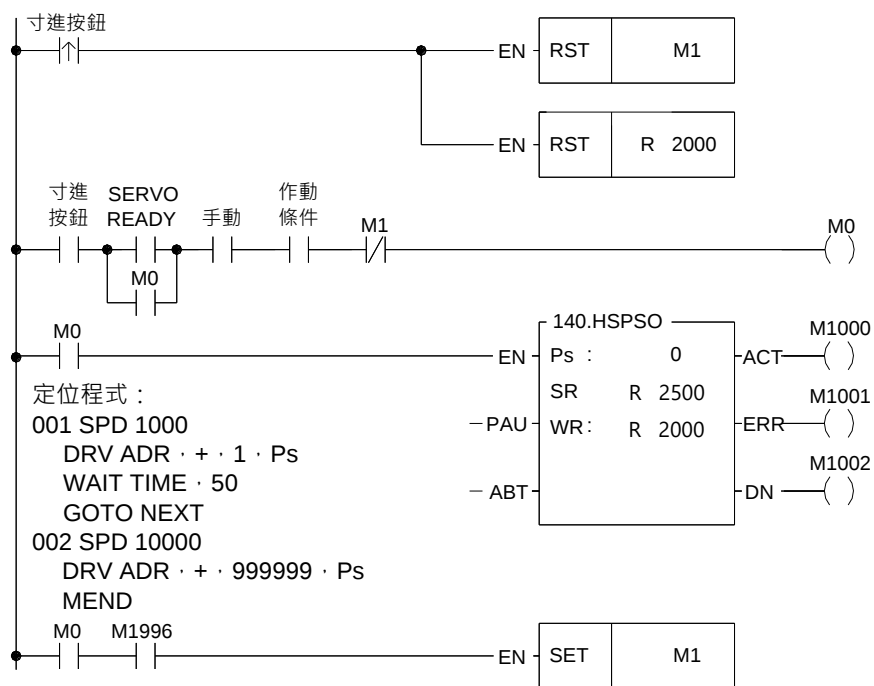
FUN140
HSPSO單軸高速脈波輸出
(含衍生定位指令)FUN140
HSPSO

程式範例：寸動進 (Jog Forward)

當寸進按鈕被按時間小於 0.5 秒 (可變) 時，祇輸出一個 (可變) 脈波；

當寸進按鈕被按時間大於 0.5 秒 (可變) 時，連續輸出脈波 (頻率為 10KHz，可變)，直到放開寸進按鈕才停止輸出；或可設計成最多祇能輸出 N 個脈波數。

階段圖



- 清除結束訊號
- 每次由第一步開始執行

- 最後一步執行完，設定結束信號

ST

```

IF R_TRIG( S:=寸進按鈕) THEN
M1 := FALSE;
R2000 := 0;
END_IF
IF 寸進按鈕 AND (SERVO READY OR M0) AND 手動 AND 作動條件 AND M1 = FALSE THEN
M0 := TRUE;
ELSE
M0 := FALSE;
END_IF
HSPSO( EN:= M0, PAU:= FALSE, ABT:= FALSE, Ps:= 0, SR:= R2500, WR:= R2000,
ACT=> M1000, ERR=> M1001, DN=> M1002);
IF M0 AND M1996 Then
M1 := FALSE;
END_IF

```

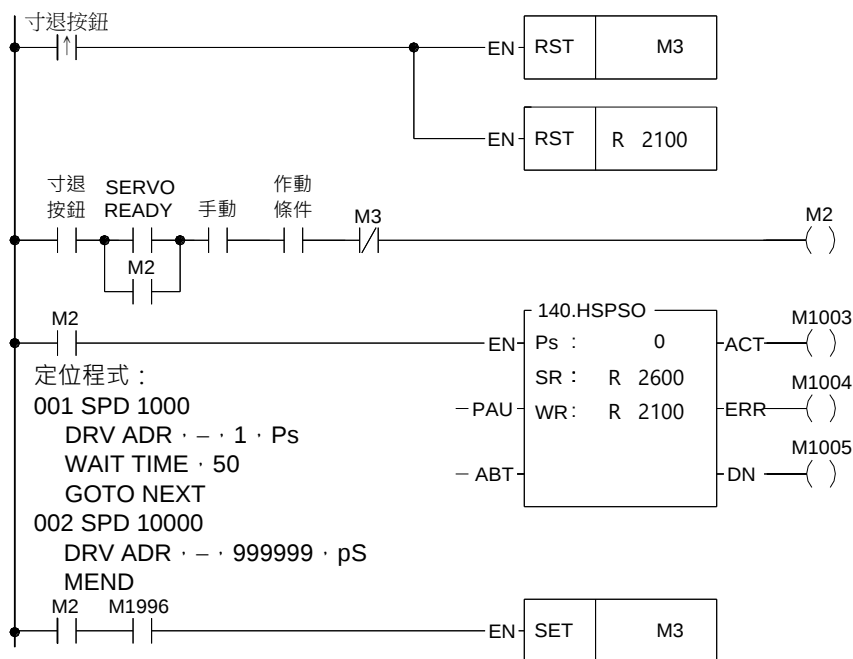
FUN140
HSPSO單軸高速脈波輸出
(含衍生定位指令)FUN140
HSPSO

程式範例：寸動退 (Jog Backward)

當寸退按鈕被按時間小於 0.5 秒 (可變) 時，祇輸出一個 (可變) 脈波；

當寸退按鈕被按時間大於 0.5 秒 (可變) 時，連續輸出脈波 (頻率為 10KHz，可變)，直到放開寸退按鈕才停止輸出；或可設計成最多祇能輸出 N 個脈波數。

階段圖



- 清除結束訊號
- 每次由第一步開始執行

- 最後一步執行完，設定結束信號

ST

```

IF R_TRIG( S:=寸退按鈕) THEN
M3 := FALSE;
R2100 := 0;
END_IF

IF 寸退按鈕 AND (SERVO READY OR M2) AND 手動 AND 作動條件 AND M3 = FALSE THEN
M2 := TRUE;
ELSE
M2 := FALSE;
END_IF

HSPSO( EN:= M2, PAU:= FALSE, ABT:= FALSE, Ps:= 0, SR:= R2500, WR:= R2000,
ACT=> M1003, ERR=> M1004, DN=> M1005);
IF M2 AND M1996 Then
M3 := FALSE;
END_IF

```

7-16-5 定位程式參數設定指令(MPARA)

FUN141 MPARA	定位程式參數設定指令				FUN141 MPARA																			
指令說明																								
階段圖符號 執行控制 — EN — 141.MPARA Ps : SR : — ERR — 錯誤訊息				Ps : 第幾組 Pulse 輸出 (0 ~ 7) SR : 參數表起始暫存器，共 18 個參數，佔用 24 個暫存器																				
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>DR</th><th>ROR</th><th>K</th></tr><tr><td>R0 R34767</td><td>D0 D7999</td><td>R43224 R47319</td><td>2 256</td></tr><tr><td>Ps</td><td></td><td></td><td></td><td>0 ~ 7</td></tr><tr><td>SR</td><td>○</td><td>○</td><td>○</td><td></td></tr></table>						運算元 \ 範圍	HR	DR	ROR	K	R0 R34767	D0 D7999	R43224 R47319	2 256	Ps				0 ~ 7	SR	○	○	○	
運算元 \ 範圍	HR	DR	ROR	K																				
	R0 R34767	D0 D7999	R43224 R47319	2 256																				
Ps				0 ~ 7																				
SR	○	○	○																					
功能敘述																								
1. 定位軸最多可以控制到 PS07，唯實際能夠控制的最大軸號依主機機型而有差異。 2. 本指令並不一定要使用；如果系統內定之參數值已符合使用者需求，則可不必用到此指令；如果需開放參數值作動態修改，則需有此指令。 3. 本指令配合 FUN140 作定位控制使用，每一軸只可使用一個本指令 4. 無論控制輸入 ``EN`` =0 或 1 時，本指令皆會被執行。 5. 當參數值有錯誤時，輸出指示 ``ERR`` ON，且錯誤代碼會出現在錯誤碼暫存器。SR=參數表起始暫存器，假設為 R2000																								

FUN141 MPARA	定位程式參數設定指令				FUN141 MPARA
R2000	0 ~ 2		參數 0	系統內定值=1	
R2001	1 ~ 65535 Ps/Rev		參數 1	系統內定值=2000	
DR2002	1 ~ 999999 μM/Rev 1 ~ 999999 mDeg/Rev 1 ~ 99999930.1 mInch/Rev		參數 2	系統內定值=2000	
R2004	0 ~ 3		參數 3	系統內定值=2	
DR2005	1 ~ 200000 Ps/sec 1 ~ 153000		參數 4	系統內定值=460000	
DR2007	0~921600 Ps/sec 1~153000		參數 5	系統內定值=141	
R2009	1~65535 Ps/sec		參數 6	系統內定值=1000	
R2010	0~32767		參數 7	系統內定值=0	
R2011	0~30000		參數 8	系統內定值=5000	
R2012	0~1	0~1	參數 9	系統內定值=0100H	
R2013	-32768~32767		參數 10	系統內定值=0	
R2014	-32768~32767		參數 11	系統內定值=0	
R2015	0~30000		參數 12	系統內定值=0	
R2016	0~30000		參數 13	系統內定值=500	
DR2017	0~1999999		參數 14	系統內定值=0	
DR2019	00H~FFH	00H~FFH	參數 15	系統內定值=FFFFFFFFH	
	00H~FFH	00H~FFH			
DR2021	-999999~999999		參數 16	系統內定值=0	
R2023	0~255		參數 17	系統內定值=1	

FUN141
MPARA

定位程式參數設定指令

FUN141
MPARA

使用 UperLogic 編輯伺服參數表格

於專案視窗中點選伺服參數表格：

專案名稱

表格編輯

伺服參數表格

→ 按右鍵後，點選“新增伺服參數表格”。

表格屬性

表格類別：伺服命令表格

表格名稱：Servo_TEST

表格起始位址：R5000

長度：20

表格容量設定：
☒ 動態配置
☐ 固定配置長度(字組)

☐ 資料從 PLC 載入
☐ 從 ROR 載入資料

說明

確認 取消

- 表格類別：固定為“伺服參數表格”。
- 表格名稱：可為伺服參數表格輸入一容易辨識之名稱，方便日後修改或除錯用。
- 表格起始位置：伺服參數指令(FUN141)所用之資料表格起始暫存器 SR 之起始位置。

FUN141 MPARA	定位程式參數設定指令	FUN141 MPARA
-----------------	------------	-----------------

伺服參數表格-[rty]
?
×

計算機
設定

參數設定

R5000	0 單位設定	1: 脈波	R5013	10 正轉移動量補正值	0Ps
R5001	1 脈波數/轉 (16Bit)	2000	R5014	11 反轉移動量補正值	0Ps
DR5002	2 移動量/轉	2000	R5015	12 減速時間設定	0ms
R5004	3 最小設定單位	2	R5016	13 補間加減速時間設定	500ms
DR5005	4 最高速度	200000	DR5017	14 脈波數/轉 (32Bit)	0
DR5007	5 起始/結束速度	141	R5019 LB	15-0近點 DOG 輸入接點	不使用 0 [X0]
R5009	6 原點復歸減速速度	1000	R5019 HB	15-1行程極限輸入接點	不使用 0 [X0]
R5010	7 齒輪間隙補償值	0Ps	R5020 LB	15-2零點信號 PG0 輸入接點	不使用 0 [X0]
R5011	8 加減速時間設定	5000ms	R5020 HB	15-3歸零清除信號 CLR 輸出接點	不使用 0 [Y0]
R5012 LB	9-0運轉方向	0: Up	DR5021	16 機械原點位置值	0Ps
R5012 HB	9-1原點復歸方向	1: Down (左)	R5023	17 零點信號數	1

設定: 動態配置[29768]字組
資料長度: 24 字組
配置位置: R5000 - R5023

預設值
確定
取消

參數說明

- 參數 0：單位設定，內定值為 1
 - 設定值為 0 時，程式內所使用之行程與速度設定值均被指定以 mm・Deg・Inch 為單位，稱為機械單位。
 - 設定值為 1 時，程式內所使用之行程與速度設定值均被指定以 Pulse 為單位，稱為馬達單位。
 - 設定值為 2 時，程式內所使用之行程設定值均被指定以 mm・Deg・Inch 為單位，而速度設定均被指定以 Pulse 為單位，稱為複合單位。

參數 0・單位	“0” 機械單位	“1” 馬達單位	“2” 複合單位
參數 1・2	必須設定	不必設定	必須設定
參數 3・7・10・11	mm・Deg・Inch	Ps	mm・Deg・Inch
參數 4・5・6・15・16	Cm/Min・Deg/Min・Inch/Min	Ps/Sec	Ps/Sec

FUN141 MPARA	定位程式參數設定指令			FUN141 MPARA																												
● 參數 1：脈波數/1 轉，內定值為 2000，亦即 2000 Ps/Rev 馬達轉一圈所需之脈波數 (A) A=1 ~ 65535 (32767 以上時，以十進位正數設定) Ps/Rev 當參數 14 = 0 時，以參數 1 為脈波數/1 轉。 當參數 14 ≠ 0 時，以參數 14 為脈波數/1 轉。 ● 參數 2：移動量/1 轉，內定值為 2000，亦即 2000 Ps/Rev 馬達轉一圈所帶動之距離 (B) B=1 ~ 999999 μM/Rev 1 ~ 999999 mDeg/Rev 1 ~ 999999x0.1 mInch/Rev ● 參數 3：最小設定單位，內定值為 2，相當於小數點二位<table><tr><th rowspan="2">參數 3 \ 參數 0</th><th colspan="3">設定值=0，機械單位；設定值=2，複合單位；</th><th rowspan="2">設定值=1 馬達單位 Ps</th></tr><tr><th>mm</th><th>Deg</th><th>Inch</th></tr><tr><td>設定值=0</td><td>x1</td><td>x1</td><td>x0.1</td><td>x1000</td></tr><tr><td>設定值=1</td><td>x0.1</td><td>x0.1</td><td>x0.01</td><td>x100</td></tr><tr><td>設定值=2</td><td>x0.01</td><td>x0.01</td><td>x0.001</td><td>x10</td></tr><tr><td>設定值=3</td><td>x0.001</td><td>x0.001</td><td>x0.0001</td><td>x1</td></tr></table> ● 參數 4：最高速度設定，內定值為 200000，亦即 200000 Ps/Sec ○ 馬達及複合單位：1 ~ 200000 Ps/Sec ○ 機械單位：1 ~ 153000 (cm/Min · x10 Deg/Min · Inch/Min) 但最高頻率不能大於 200000 Ps/Sec $f_max = (V_max \times 1000 \times A) / (6 \times B) \leq 200000 \text{ Ps/Sec}$ $f_min \geq 1 \text{ Ps/Sec}$ 註：A=參數 1 · B=參數 2 ● 參數 5：起始 / 結束速度，內定值=141 ○ 馬達及複合單位：1~200000 Ps/Sec。 ○ 機械單位：1 ~ 15300 (cm/Min · x10 Deg/Min · Inch/Min) 但最高頻率不能大於 200000 Ps/Sec。 ● 參數 6：原點復歸減速速度，內定值為 1000 馬達及複合單位：1 ~ 65535 Ps/Sec 機械單位：1 ~ 15300 (Cm/Min, x10 Deg/Min, Inch/Min) ● 參數 7：齒輪間隙補正值，內定值=0 設定範圍：0 ~ 32767 Ps。 反相行走時，行走距離會自動加上此值。 註：多軸直線補間命令無效					參數 3 \ 參數 0	設定值=0，機械單位；設定值=2，複合單位；			設定值=1 馬達單位 Ps	mm	Deg	Inch	設定值=0	x1	x1	x0.1	x1000	設定值=1	x0.1	x0.1	x0.01	x100	設定值=2	x0.01	x0.01	x0.001	x10	設定值=3	x0.001	x0.001	x0.0001	x1
參數 3 \ 參數 0	設定值=0，機械單位；設定值=2，複合單位；			設定值=1 馬達單位 Ps																												
	mm	Deg	Inch																													
設定值=0	x1	x1	x0.1	x1000																												
設定值=1	x0.1	x0.1	x0.01	x100																												
設定值=2	x0.01	x0.01	x0.001	x10																												
設定值=3	x0.001	x0.001	x0.0001	x1																												

FUN141 MPARA	定位程式參數設定指令	FUN141 MPARA								
● 參數 8：加減速時間設定，內定值=5000，單位為 mS 註：多軸直線補間命令無效 設定範圍：0~30000 mS。 此時間代表由靜止加速至最高速度(參數 4)，或由最高速度減至靜止所需時間。 本系統之加減速乃為等斜率控制。 當參數 12=0 時，以本參數為減速時間。 本系統之加減速控制，依實際動作行程會自動以三角波型或梯形波型運動。 ● 參數 9：原點復歸方向與運轉方向設定，內定值為 0100H 註：多軸直線補間命令無效 <table><tr><td></td><td>b15</td><td>b8 b7</td><td>b0</td></tr><tr><td>SR+12</td><td colspan="2">參數 9-1</td><td>參數 9-0</td></tr></table> ● 參數 9-0：運轉方向設定，內定值為 0 設定值=0 時，正轉脈波輸出、目前 Ps 值往上加 反轉脈波輸出、目前 Ps 值往下減 設定值=1 時，正轉脈波輸出、目前 Ps 值往下減 反轉脈波輸出、目前 Ps 值往上加 ● 參數 9-1：原點復歸方向設定，內定值為 1 設定值=0 時，原點復歸方向為目前 Ps 值往上加方向 (原點在右) 設定值=1 時，原點復歸方向為目前 Ps 值往下減方向 (原點在左) ● 參數 10：正轉移動量補正值，內定值=0 註：多軸直線補間命令無效 ● 設定範圍：-32768~32767 Ps。 ● 正轉脈波輸出時，會自動加上此值作為移動距離。 ● 參數 11：反轉移動量補正值，內定值=0 註：多軸直線補間命令無效 ● 設定範圍：-32768~32767 Ps。 ● 反轉脈波輸出時，會自動加上此值作為移動距離。				b15	b8 b7	b0	SR+12	參數 9-1		參數 9-0
	b15	b8 b7	b0							
SR+12	參數 9-1		參數 9-0							

- 參數 12：減速時間設定，內定值=0，單位為 mS 註：多軸直線補間命令無效

- 設定範圍：0~30000 mS。
- 當參數 12 = 0 時，以參數 8 為減速時間。
- 當參數 12 ≠ 0 時，以參數 12 為減速時間。

- 參數 13：補間加減速時間(定數)設定，內定值為 500

註：多軸直線補間命令專用

- 設定範圍：0~30000 mS
- 用來設定直線補間運動時，由靜止(速度=0)加速至工作頻率所需時間；本時間亦用來作減速停止控制

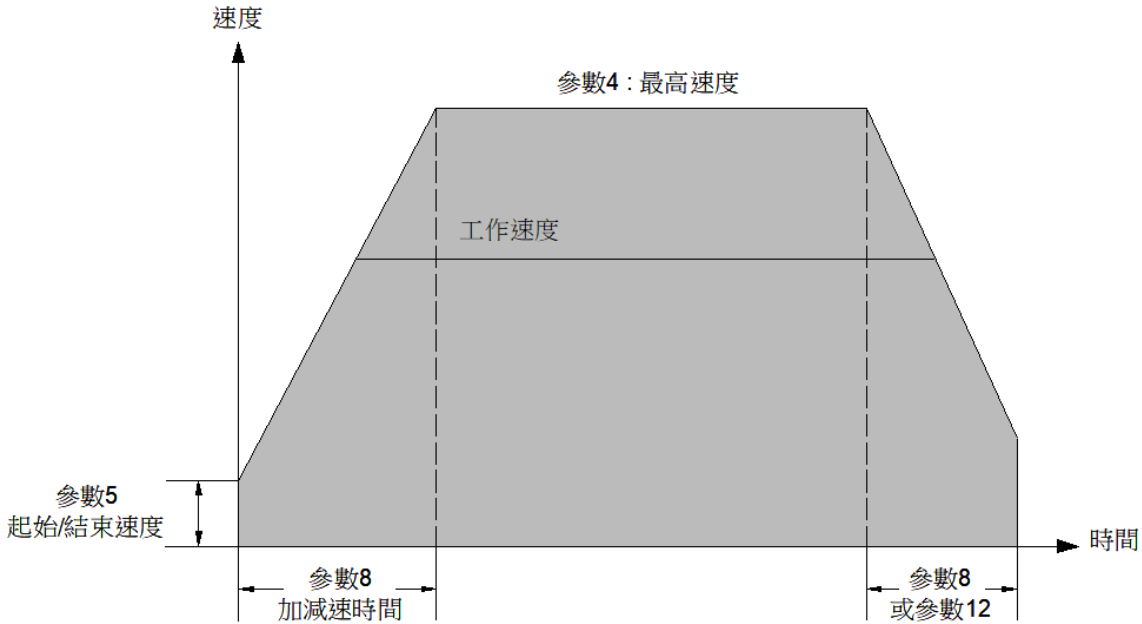
- 參數 14：脈波數/1 轉，內定值為 0

- 設定範圍：0~1999999。
- 當參數 14 = 0 時，以參數 1 為脈波數/1 轉。
- 當參數 14 ≠ 0 時，以參數 14 為脈波數/1 轉。

- 參數 15：控制介面 I/O 設定，內定值為 FFFFFFFFH

	b15	b8 b7	b0
SR+19	參數 15-1	參數 15-0	
SR+20	參數 15-3	參數 15-2	

- 參數 15-0：近點 DOG 輸入接點設定; 必須為主機之輸入點(SR+19)
b6 ~ b0：近點 DOG 輸入接點號碼 (0 ~ 15，亦即 X0 ~ X15)
b7 = 0：近點 DOG 輸入為常開接點 (A 或 NO 接點)
= 1：近點 DOG 輸入為常閉接點 (B 或 NC 接點)
b7 ~ b0=FFH，無近點 DOG 輸入
- 參數 15-1：行程極限輸入接點設定 (SR+19)
b14 ~ b8：行程極限輸入接點號碼(0 ~ 125，亦即 X0 ~ X125)
b15 = 0：行程極限輸入為常開接點 (A 或 NO 接點)
= 1：行程極限輸入為常閉接點 (B 或 NC 接點)
b15 ~ b8=FFH：無行程極限輸入

FUN141 MPARA	定位程式參數設定指令	FUN141 MPARA
	● 參數 15-2：零點信號 PG0 輸入接點設定；必須為主機之輸入點(SR+20) b6 ~ b0：零點信號 PG0 輸入接點號碼 (0 ~ 15，亦即 X0 ~ X15) b7 = 0：近點 DOG 前緣開始計數零點信號 = 1：近點 DOG 後緣開始計數零點信號 b7 ~ b0 = FFH：無零點信號 PG0 輸入 ● 參數 15-3：歸零清除信號 CLR 輸出接點設定；必須為主機之輸出點(SR+20) b15 ~ b8：歸零清除信號 CLR 輸出接點號碼 (0 ~ 23，亦即 Y0 ~ Y23) b15 ~ b8=FFH：無歸零清除信號 CLR 輸出 ● 參數 16：機械原點位置值，內定值為 0 -999999 ~ 999999 Ps ● 參數 17：零點信號數，內定值為 1 0 ~ 255 Count  The diagram illustrates a trapezoidal velocity profile over time. The vertical axis is labeled '速度' (Velocity) and the horizontal axis is labeled '時間' (Time). The profile starts at a low velocity (labeled '參數5 起始/結束速度'), rises linearly to a peak velocity (labeled '參數4：最高速度'), remains constant at that peak velocity for a duration, and then falls linearly back to the low velocity. The rising and falling slopes are labeled '工作速度' (Working speed). The time taken for the acceleration and deceleration phases is indicated by two horizontal double-headed arrows, both labeled '參數8 加減速時間'. The total time for the deceleration phase is also labeled '參數8 或參數12'.	

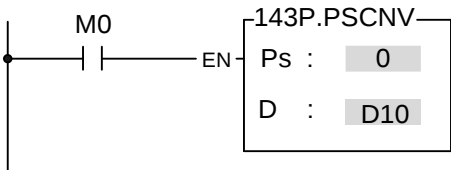
7-16-6 強制停止 HSPSO 脈波輸出(PSOFF)

FUN142 P PSOFF	強制停止 HSPSO 脈波輸出指令		FUN142 P PSOFF		
指令說明					
階段圖符號 執行控制 — EN — <table><tr><td>142P. PSOFF</td><td>PsN</td></tr></table>		142P. PSOFF	PsN	N : 0 ~ 7 強制第幾組 Pulse 輸出停止輸出	
142P. PSOFF	PsN				
功能敘述	1. 定位軸最多可以控制到 PS07，唯實際能夠控制的最大軸號依主機機型而有差異。 2. 當停止控制“EN” =1 或“EN” 由 0→1 (P 指令) 時，本指令將強制所指定之第幾組 Pulse 輸出停止輸出。 3. 在執行機械原點復歸之應用時，當原點條件滿足時，利用本指令可快速停止脈波輸出，讓每次作機械原點復歸時，都停在同一個位置。				
程式範例					
M0 ● — — EN — <table><tr><td>142P. PSOFF</td><td>0</td></tr></table>		142P. PSOFF	0	當 M0 由 0→1 時，強制 Ps0 停止脈波之輸出。	
142P. PSOFF	0				

7-16-7 目前脈波值轉換為顯示值(PSCNV)

FUN143  PSCNV	目前脈波值轉換為顯示值 (mm · Deg · Inch · PS) 指令	FUN143  PSCNV																		
指令說明																				
階段圖符號 執行控制 — EN — 143P.PSCNV Ps : D :		Ps : 0 ~ 7 ; 將第幾組脈波位置轉換為與設定值同單位之 mm (Deg , Inch , PS) , 以作為目前位置顯示。 D : 儲存轉換後目前位置之暫存器 , 共需使用二個暫存器 ; 例如 D10 , 即代表 D10 (Low Word) 與 D11 (High Word) 二個暫存器。																		
<table><tr><th rowspan="3">運算元 \ 範圍</th><th>HR</th><th>DR</th><th>ROR</th><th>K</th></tr><tr><td>R0 R34767</td><td>D0 D7999</td><td>R43224 R47319</td><td>2 256</td></tr><tr><td>Ps</td><td></td><td></td><td>0 ~ 7</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○*</td><td></td></tr></table>			運算元 \ 範圍	HR	DR	ROR	K	R0 R34767	D0 D7999	R43224 R47319	2 256	Ps			0 ~ 7	D	○	○	○*	
運算元 \ 範圍	HR	DR		ROR	K															
	R0 R34767	D0 D7999		R43224 R47319	2 256															
	Ps			0 ~ 7																
D	○	○	○*																	
功能敘述																				
1. 定位軸最多可以控制到 PSO7 , 唯實際能夠控制的最大的軸號依主機機型而有差異。 2. 當執行控制 “EN” =1 或 “EN” 由 0→1 ( 指令) 時 , 本指令將所指定之目前脈波位置 (Ps) 轉換為與設定值同單位之 mm (或 Deg 或 Inch 或 Ps) , 以作為目前位置顯示。 3. FUN140 或 FUN147 指令執行過後 , 執行本指令時 , 才會得到正確之轉換值。																				

FUN143 P PSCNV	目前脈波值轉換為顯示值 (mm · Deg · Inch · PS) 指令	FUN143 P PSCNV
程式範例		

階梯圖	ST
	PSCNV(EN:= M0, Ps:= 0, D:= D10);

- 當 M0 由 0→1 時，將 Ps0 目前脈波位置 (DR4088)，轉換為與設定值同單位之 mm (或 Deg 或 Inch 或 PS)，並存入 DD10 以作為目前位置顯示。

7-16-8 高速脈波寬度調變 2(HSPWM2)

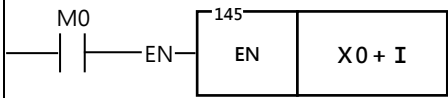
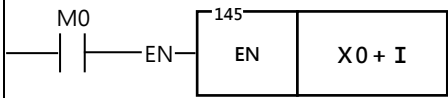
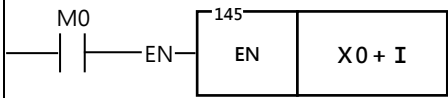
FUN144 HSPWM2	高速脈波寬度調變 2 (HIGH SPEED PULSE WIDTH MODULATION 2)												FUN144 HSPWM2		
指令說明															
輸出控制—EN XXX D.PWM PW: Op: Hz: OR: ACT—脈波輸出 ERR—錯誤										PW：脈波寬度調變輸出點(0=Y0, 1=Y2, 2=Y4, 3=Y6, 4=Y8, 5=Y10, 6=Y12, 7=Y14) Op：輸出極性(0=正相, 1=倒相) Hz：輸出頻率(1~100000000 or 1~200000000 · 單位 0.001Hz) OR：脈波輸出寬度(0~100 · 單位%)					
範圍 運算元	Y	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR		DR	K
	主機上之 Yn	WX0 WX1008	WY0 WY1008	WM0 WM19578	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R34768 R34895	RR3355022840 RR343453125213	R43224 R47319	D0 D7999		
	Pw									○			○	○	
	Op														0 ~ 1
	Hz										○		○	○	
	OR										○		○	○	0 ~ 100
1. 相較於 FUN139，FUN144 提供了更直接便利的高速 PWM 輸出控制，無須透過內建公式推算參數。 2. 最大輸出頻率依照機型可能為 100K 或 200K，如超過最大輸出頻率將無法執行。															

7-17 中斷控制指令 (FUN145 ~ 146)

7-17-1 允許外界輸入或週邊中斷作動指令(EN)

FUN145 P EN	允許外界輸入或週邊中斷作動指令		FUN145 P EN
指令說明			
階段圖符號 允許中斷控制 — EN — 145P. EN LBL		LBL：允許中斷作動之外界輸入或週邊標記名稱。	
功能敘述			
● 當允許控制“EN” =1 或由 0→1 (P 指令) 時，允許 LBL 所指定之外界輸入或週邊中斷作動。 ● 可允許之中斷標記名稱如下： (詳細請參考軟體進階使用手冊第 5 章 5-3 節之說明)			

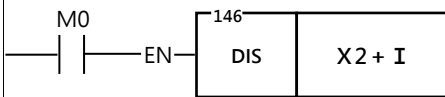
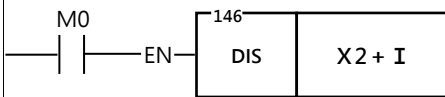
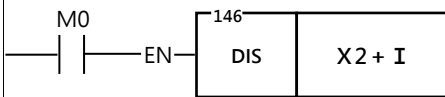
FUN145 P EN	允許外界輸入或週邊中斷作動指令				FUN145 P EN
功能敘述					
LBL 名稱	敘 述	LBL 名稱	中斷控制點	LBL 名稱	中斷控制點
HSC0I	HSC0 高速計數器中斷	X4-I	X4 負緣中斷	LTM2I	10 MS 定時計時器 LTM2 中斷
HSC1I	HSC1 高速計數器中斷	X5+I	X5 正緣中斷	LTM3I	10 MS 定時計時器 LTM3 中斷
HSC2I	HSC2 高速計數器中斷	X5-I	X5 負緣中斷	HST0I	HST0 高速計時器中斷
HSC3I	HSC3 高速計數器中斷	X6+I	X6 正緣中斷	HST1I	HST1 高速計時器中斷
X0+I	X0 正緣中斷	X6-I	X6 負緣中斷	HST2I	HST2 高速計時器中斷
X0-I	X0 負緣中斷	X7+I	X7 正緣中斷	HST3I	HST3 高速計時器中斷
X1+I	X1 正緣中斷	X7-I	X7 負緣中斷		
X1-I	X1 負緣中斷	STM0I	1 MS 定時計時器 STM0 中斷		
X2+I	X2 正緣中斷	STM1I	1 MS 定時計時器 STM1 中斷		
X2-I	X2 負緣中斷	STM2I	1 MS 定時計時器 STM2 中斷		
X3+I	X3 正緣中斷	STM3I	1 MS 定時計時器 STM3 中斷		
X3-I	X3 負緣中斷	LTM0I	10 MS 定時計時器 LTM0 中斷		
X4+I	X4 正緣中斷	LTM1I	10 MS 定時計時器 LTM1 中斷		

FUN145 P EN	允許外界輸入或週邊中斷作動指令	FUN145 P EN				
在實際應用上，有些中斷信號在有些時候不可讓它發生作用，而在有些時候卻必須讓它發生作用；利用 FUN146 (DIS) 和 FUN145 (EN) 指令即可達成上述需求。						
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF M0 THEN INTEnable(LB:= X0 + I); END_IF</pre></td></tr></table>			階梯圖	ST		<pre>IF M0 THEN INTEnable(LB:= X0 + I); END_IF</pre>
階梯圖	ST					
	<pre>IF M0 THEN INTEnable(LB:= X0 + I); END_IF</pre>					
當 M0 由 0→1 時，允許 X0 由 0→1 時發出中斷；CPU 可立即快速處理 X0+I 之中斷服務程式。						

7-17-2 禁止外界輸入或週邊中斷作動指令(DIS)

FUN146 DIS	禁止外界輸入或週邊中斷作動指令		FUN146 DIS
指令說明			
階段圖符號 禁止中斷控制 — EN — 146P. DIS LBL		LBL：禁止作動之外界輸入或週邊之中斷標記名稱。	
功能敘述			
● 當禁止控制“EN” =1 或由 0→1 (P 指令) 時，禁止 LBL 所指定之外界輸入或週邊中斷或週邊功能作動。 ● 可禁止之中斷標記名稱如下： (同可允許者)			
功能敘述			

FUN146 P DIS		禁止外界輸入或週邊中斷作動指令				FUN146 P DIS	
LBL 名稱	敘 述	LBL 名稱	中斷控制點	LBL 名稱	中斷控制點		
HSC0I	HSC0 高速計數器中斷	X4-I	X4 負緣中斷	LTM2I	10 MS 定時計時器 LTM2 中斷		
HSC1I	HSC1 高速計數器中斷	X5+I	X5 正緣中斷	LTM3I	10 MS 定時計時器 LTM3 中斷		
HSC2I	HSC2 高速計數器中斷	X5-I	X5 負緣中斷	HST0I	HST0 高速計時器中斷		
HSC3I	HSC3 高速計數器中斷	X6+I	X6 正緣中斷	HST1I	HST1 高速計時器中斷		
X0+I	X0 正緣中斷	X6-I	X6 負緣中斷	HST2I	HST2 高速計時器中斷		
X0-I	X0 負緣中斷	X7+I	X7 正緣中斷	HST3I	HST3 高速計時器中斷		
X1+I	X1 正緣中斷	X7-I	X7 負緣中斷				
X1-I	X1 負緣中斷	STM0I	1 MS 定時計時器 STM0 中斷				
X2+I	X2 正緣中斷	STM1I	1 MS 定時計時器 STM1 中斷				
X2-I	X2 負緣中斷	STM2I	1 MS 定時計時器 STM2 中斷				
X3+I	X3 正緣中斷	STM3I	1 MS 定時計時器 STM3 中斷				
X3-I	X3 負緣中斷	LTM0I	10 MS 定時計時器 LTM0 中斷				
X4+I	X4 正緣中斷	LTM1I	10 MS 定時計時器 LTM1 中斷				

FUN146 ^P DIS	禁止外界輸入或週邊中斷作動指令	FUN146 ^P DIS				
在實際應用上，有些中斷信號在有些時候不可讓它發生作用，即可利用此指令將該中斷信號禁止而不會產生中斷處理。						
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td><pre>IF M0 THEN INTDisable(LB:= X2+I); END_IF</pre></td></tr></table>			階梯圖	ST		<pre>IF M0 THEN INTDisable(LB:= X2+I); END_IF</pre>
階梯圖	ST					
	<pre>IF M0 THEN INTDisable(LB:= X2+I); END_IF</pre>					
當 M0 由 0→1 時，禁止 X2 由 0→1 時發出中斷處理。						

7-18 NC 定位控制指令二 (FUN147~FUN149)

7-18-1 FUN147 多軸直線補間定位輸出指令

FUN147 MHSP0		多軸直線補間定位輸出指令				FUN147 MHSP0																								
		階段圖符號 執行控制 — EN 暫停輸出 — PAU 放棄輸出 — ABT 147.MHSP0 Gp : SR : WR : ACT — 脈波輸出 ERR — 錯誤訊息 DN — 定位完成				Gp : 第幾個群組 (0 ~ 1) SR : 定位程式起始暫存器 WR : 指令運作起始暫存器，共佔用 9 個暫存器， 其它程式不可重覆使用																								
		<table><tr><td rowspan="3">範圍 運算元</td><td>HR</td><td>DR</td><td>ROR</td><td>K</td></tr><tr><td>R0 R34767</td><td>D0 D7999</td><td>R43224 R47319</td><td></td></tr><tr><td>Gp</td><td></td><td></td><td>0 ~ 1</td></tr><tr><td>SR</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td>○</td><td>○*</td><td></td></tr></table>						範圍 運算元	HR	DR	ROR	K	R0 R34767	D0 D7999	R43224 R47319		Gp			0 ~ 1	SR	○	○	○		WR	○	○	○*	
範圍 運算元	HR	DR	ROR	K																										
	R0 R34767	D0 D7999	R43224 R47319																											
	Gp			0 ~ 1																										
SR	○	○	○																											
WR	○	○	○*																											
指令說明																														

1. FUN147 (MHSP0) 指令之直線補間定位程式是以文字之程式書寫方式來編輯；每一定位點我們稱一步 (含輸出頻率、動作行程、轉移條件)，每一步定位點需佔用 15 個暫存器。
2. FUN147 (MHSP0) 直線補間定位指令最多可同時作四軸直線補間，或兩組獨立之二軸直線補間運動。
3. 將定位程式存在暫存器最大好處是，如果結合人機作機台操控設定，則可將定位程式存入人機，更換模具時，可直接由人機操作存取該副模具之定位程式。
4. 當執行控制輸入“EN”=1 時，如 Gp(群組)內的各軸(Ps0~Ps3)沒有被其它 FUN140 或 FUN147 指令佔用 (Ps0=M9183、Ps1=M9184、Ps2=M9185、Ps3=M9186 之狀態為 ON)，則由下一步定位點開始執行 (如已至最後一步，則重新由第 1 步開始執行)；如 Gp(群組)內的各軸(Ps0~3)被其它 FUN140 或 FUN147 指令佔用 (Ps0=M9183、Ps1=M9184、Ps2=M9185、Ps3=M9186 之狀態為 OFF)，則等佔用之 FUN140 或 FUN147 釋出控制權，本指令取得定位控制之脈波(Pulse)輸出權。
5. 當執行控制“EN”=0 時，馬上停止脈波輸出。
6. 當暫停輸出“PAU”=1，且執行控制“EN”先前為 1 時，則暫停脈波輸出；當暫停輸出“PAU”=0，而執行控制“EN”仍為 1 時，繼續輸出未完成之脈波數。
7. 當放棄輸出“ABT”=1 時，馬上停止脈波輸出。
(下一次當執行控制輸入“EN”=1 時，重新由第一步定位點開始執行)
8. 當脈波輸出中，輸出指示“ACT”ON。
9. 當指令執行錯誤時，輸出指示“ERR”ON。
(錯誤代碼存放於錯誤碼暫存器)
10. 當每一步定位點完成時，輸出指示“DN”ON。
11. 請參考功能與應用手冊第九章。

*** 務必設定 Pulse Output 之工作模式 (不設定時，Y0 ~ Y7 當作一般輸出)
為 U/D 或 A/B 等二種模式之一，Pulse Output 才能正常輸出。

U/D 模式：Y0 (Y2 , Y4 , Y6) 送出上數脈波。

Y1 (Y3 , Y5 , Y7) 送出下數脈波。

A/B 模式：Y0 (Y2 , Y4 , Y6) 送出 A 相脈波。

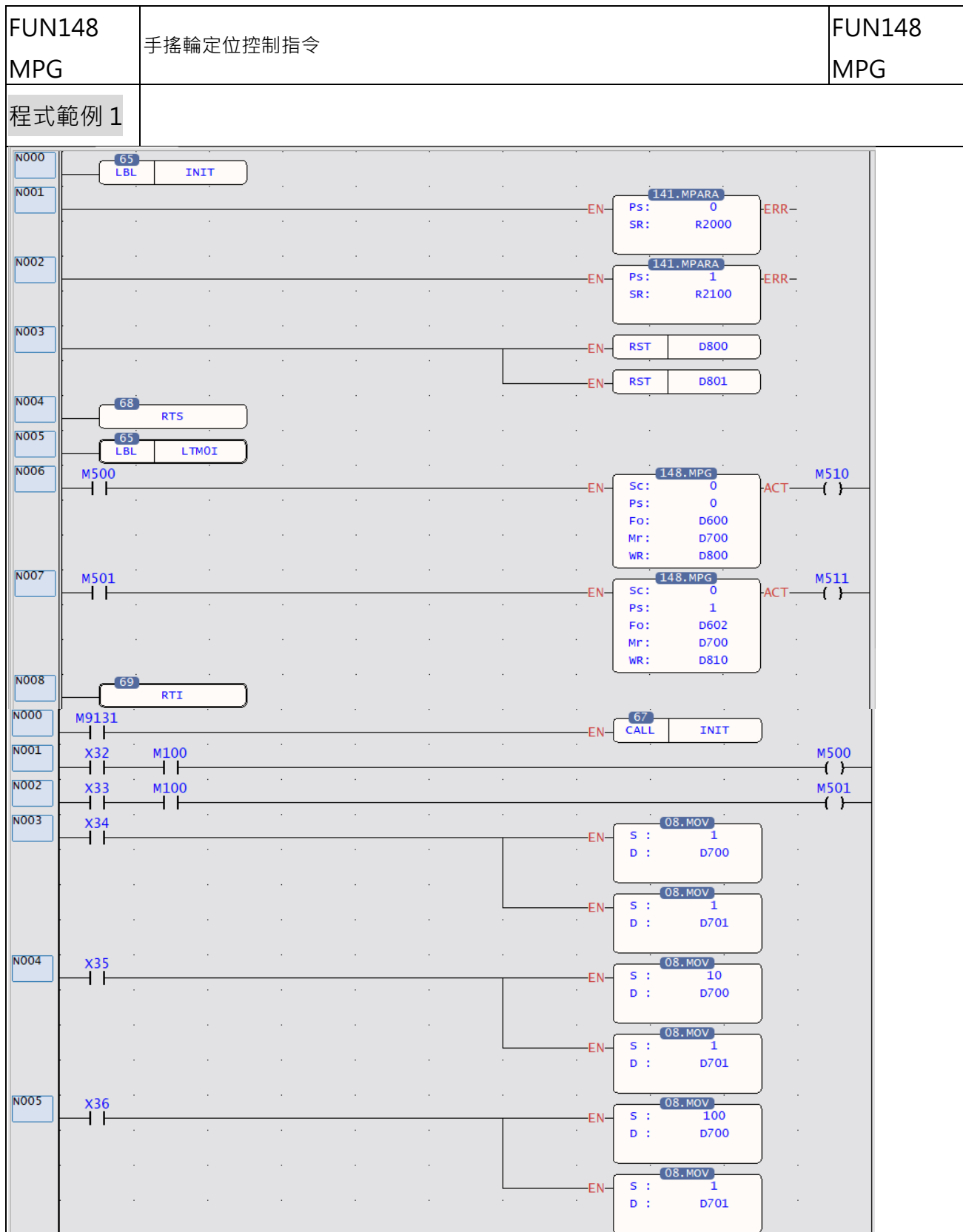
Y1 (Y3 , Y5 , Y7) 送出 B 相脈波。

● Pulse Output 輸出極性可選擇 Normal ON 或 Normal OFF。

7-18-2 FUN148 手搖輪定位控制 (MPG)

FUN148 MPG	手搖輪定位控制指令	FUN148 MPG																																		
指令說明																																				
階段圖符號 148.MPG Sc Ps Fo Mr WR 運算控制 — EN — — ACT — Sc：指定接手搖輪之來源高速計數器；0~7 Ps：指定反應手搖輪之脈波輸出軸；0~7 Fo：輸出頻率設定暫存器(2 個暫存器) Mr：倍率設定暫存器(2 個暫存器) Mr+0：倍率被乘數(Fa) Mr+1：倍率被除數(Fb) WR：工作暫存器起始位址，共佔用 4 個暫存器 輸出脈波數=(手搖輪輸入脈波數×Fa)/Fb																																				
<table><tr><th rowspan="2">範圍 運算元</th><th>HR</th><th>DR</th><th>ROR</th><th>K</th></tr><tr><td>R0 R34767</td><td>D0 D7999</td><td>R43224 R47319</td><td>16 位元</td></tr><tr><td>Sc</td><td>○</td><td>○</td><td>○</td><td>0~7</td></tr><tr><td>Ps</td><td>○</td><td>○</td><td>○</td><td>0~7</td></tr><tr><td>Fo</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>Mr</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>			範圍 運算元	HR	DR	ROR	K	R0 R34767	D0 D7999	R43224 R47319	16 位元	Sc	○	○	○	0~7	Ps	○	○	○	0~7	Fo	○	○	○		Mr	○	○	○		WR	○	○*	○	
範圍 運算元	HR	DR		ROR	K																															
	R0 R34767	D0 D7999	R43224 R47319	16 位元																																
Sc	○	○	○	0~7																																
Ps	○	○	○	0~7																																
Fo	○	○	○																																	
Mr	○	○	○																																	
WR	○	○*	○																																	

FUN148 MPG	手搖輪定位控制指令	FUN148 MPG
功能敘述	● 定位軸最多可以控制到 PSO7，唯實際能夠控制的最大軸號依主機機型而有差異。 ● 將此指令放在 10mS 單位時間定時中斷處理程式(LTM0)，並利用特殊暫存器 R35443、或利用 0.1mS 高速計時器產生 50mS 定時中斷來執行此指令，以便以較準確的時間間隔對手搖輪輸入脈波作取樣、並依倍率設定(Mr+0 與 Mr+1)計算輸出脈波數；同時在此間隔時間內以 Fo 所設定的頻率，將計算出來的輸出脈波數作輸出。 輸出頻率(Fo)設定值必須夠高，加減速(由 FUN141 指令之參數 4 與參數 8 設定)也必須夠快才足夠在高放大倍率(100 或 200 倍)的情況下，在間隔時間內將計算出來的輸出脈波數輸出完畢；否則會有失步現象。 ● 當執行控制“EN”為 1 時，每個間隔時間會對手搖輪輸入脈波作取樣；如果沒有取樣到有脈波輸入，則本指令不會有輸出；如果有取樣到有脈波輸入，則會根據倍率設定(Mr+0 與 Mr+1)計算輸出脈波數，然後以 Fo 所設定的輸出頻率，將計算出來的輸出脈波數作輸出。 輸出脈波數=(間隔時間內手搖輪輸入脈波數×Fa)/Fb ● 如果指定輸出軸被其它定位指令使用中(FUN140/FUN147)，則就算有取樣到有手搖輪脈波輸入，也不會有輸出。 ● 當脈波輸出中，輸出指示 ACT=1；否則為 0。 ● 本指令會佔用 4 個工作暫存器(WR)，其它程式不可重複使用。 ← 間隔時間 → ... ↓ · 取樣手搖輪脈波輸入 · 以 Fo 所設定的頻率，輸出脈波 ← 間隔時間 → ↓ · 取樣手搖輪脈波輸入 · 以 Fo 所設定的頻率，輸出脈波	



FUN148 MPG	手搖輪定位控制指令	FUN148 MPG
---------------	-----------	---------------

編號	狀態	資料	編號	狀態	資料	編號	狀態	資料	編號	狀態	資料
DR4080	十進制	0	DR4082	十進制	0	D800	十進制	1	D810	十進制	1
DR4088	十進制	114200	DR4090	十進制	-24300	D801	十六進制	0100H	D811	十六進制	0001H
						DD802	十進制	11250	DD812	十進制	11250
DR2005	十進制	200000	DR2105	十進制	200000	DR4096	十進制	11250	M1992	致能	ON
R2011	十進制	30	R2111	十進制	30				M1993	致能	ON
DD600	十進制	200000	DD602	十進制	200000	D700	十進制	100	D701	十進制	1
M500	致能	ON	M501	致能	OFF	X34	致能	OFF			
X32	致能	ON	X33	致能	OFF	X35	致能	OFF	X36	致能	ON

X32 : 選擇第 0 軸 (Ps0)

X33 : 選擇第 1 軸 (Ps1)

X34 : 輸出倍率為 1

X35 : 輸出倍率為 10

X36 : 輸出倍率為 100

M100 : 手搖輪作動選擇

DR2005 : 第 0 軸最高輸出頻率(FUN141 指令之參數 4) ; 200K Hz

R2011 : 第 0 軸加減速時間(FUN141 指令之參數 8) ; 30mS

DD600 : 第 0 軸手搖輪作動輸出頻率 ; 200K Hz

DR2105 : 第 1 軸最高輸出頻率(FUN141 指令之參數 4) ; 200K Hz

R2111 : 第 1 軸加減速時間(FUN141 指令之參數 8) ; 30mS

DD602 : 第 1 軸手搖輪作動輸出頻率 ; 200K Hz

範例說明：在 50MSI 定時中斷處理程式裏放入 Ps0 與 Ps1 之手搖輪定位處理指令。

當 X32=1 且 M100=1 時，啟動 Ps0 手搖輪定位處理；每個間隔時間(50mS)會對手搖輪輸入脈波(來自 HSC0)作取樣；如果沒有取樣到有脈波輸入，則 FUN148 指令不會有輸出；如果有取樣到有脈波輸入，則會根據倍率設定(D700 與 D701)計算輸出脈波數，然後以 DD600 所設定的輸出頻率，將計算出來的輸出脈波數作輸出。

輸出脈波數=(間隔時間內 HSC0 輸入脈波數×D700)/D701

7-18-3 FUN149 多軸圓弧補間定位輸出指令

FUN149 AHSP0	多軸圓弧補間定位輸出指令 Arc High Speed Pulse Output	FUN149 AHSP0
階段圖符號 執行控制—EN 暫停輸出—PAU 放棄輸出—ABT 149. AHSP0 SR: WR: ACT—動作中 ERR—錯誤 DN—執行完成		運算元 SR: 定位程式起始暫存器 WR: 指令運作工作暫存器
功能敘述		
● Function 149 (AHSP0) 指令是編輯表格中的運動參數，可以使用 Ps0~Ps7 中的任意兩個軸作為脈衝輸出軸，當它作為 fun149 的輸出軸時，將不使用當執行控制 “En” = 1 或由0→1 (P 指令) 變更時，此指令將指定的目前脈衝位置 (PS) 轉換為與設定值相同單位的毫米 (或度數、英吋、PS)，以顯示目前位置。 ● 此指令在表格中設定目標點，分為半徑、圓心、圓弧插補運動點三種模式。 ● 運動插補中的每個目標點稱為一個步 (包括運動模式、距離、速度向量、變化條件等參數)，每步需要 40個16位元暫存器，最多可編輯250個步。 ● 參數設定完成後，透過引腳啟動脈衝輸出，並在此過程中即時顯示輸出引腳、WR 暫存器以及相關繼電器 /暫存器。 ● 在脈搏波的運動插補過程中，使用者還可以利用過渡條件和變化條件來實現特定的目標。 ● 當輸入「EN」=1 時，開始脈衝輸出，完成一步後依照變化條件執行下一步。 ● 當輸入「EN」=0 時，停止脈衝輸出，下一次「EN」ON 將從第一步開始重新執行。 ● 當「PAU」=1且「EN」先前為1時，暫時輸出脈衝。 ● 當「PAU」=0且「EN」仍為1時，繼續上一個未完成的脈衝輸出。 ● 當輸入 “ABT” =1 時，停止脈衝輸出。 ● 當輸入「ABT」=0 且「EN」=1 時，從第一步開始執行。 ● 輸出脈衝波時，輸出指示 “ACT” ON。 ● 當脈衝執行出錯時，輸出指示「ERR」ON，脈衝停止輸出。 ● 每一步完成後，輸出指示「DN」ON。 ● 更多詳細信息，請參閱功能和應用程式使用手冊。		

7-19 通訊指令 (FUN150 ~ 156)

7-19-1 Modbus RTU 通訊協定(主站)通訊連線便利指令

FUN150 M-BUS	MODBUS RTU 通訊協定(主站)通訊連線便利指令 (使 PLC 經由 PORT 1,2 當作 MODBUS RTU 通訊協定之主站)	FUN150 M-BUS																									
指令說明																											
階段圖符號 執行控制 — EN ASCII/RTU — A/R 放棄運作 — ABT 150.M-BUS Pt : SR : WR : ACT — 通訊傳輸 ERR — 通訊錯誤 DN — 通訊完成		PT : 1~4 · 透過該通訊埠,以 MODBUS RTU 通訊協定作資料傳輸 SR : 通訊程式起始暫存器 WR : 指令運作起始暫存器 · 共佔用 8 個暫存器 · 其它程式不可重複使用。																									
<table><tr><th>運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><td></td><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td></td></tr><tr><td>PT</td><td></td><td></td><td></td><td>1 ~ 4</td></tr><tr><td>SR</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K		R0 R34767	R43224 R47319	D0 D7999		PT				1 ~ 4	SR	○	○	○		WR	○	○*	○	
運算元 \ 範圍	HR	ROR	DR	K																							
	R0 R34767	R43224 R47319	D0 D7999																								
PT				1 ~ 4																							
SR	○	○	○																								
WR	○	○*	○																								
功能敘述																											

- FUN150 (M-BUS) 指令提供 M SERIES(主站)透過 PORT 1~4 以 MODBUS RTU/ASCII 通訊協定與具有該通訊協定之智慧型週邊(僕站)連線。
- 一個通訊埠可經由 RS-485 介面最多與 247 台僕站連線互享資料。
- 被 FUN150 指定使用之通訊埠即為該 MODBUS RTU/ASCII 網路之主站。
- 採用程序編碼法或填表法來規劃資料流控制；即從哪一個從站獲取哪種類型的資料並保存到主站 PLC 中，或從主站 PLC 中寫入哪種類型的資料到指定的從站中。只需要 7 個註冊表即可定義；每 7 個暫存器定義一包資料傳輸。
- 當執行控制“EN”由 0→1 且放棄運作“ABT”為 0 時，若 PORT 1,2 未被其他通訊指令佔用 [M9135(PORT1),M9138(PORT2)= 1]，則本指令立即掌控 PORT 1,2，並將 M9135,M9138 設為 0 (表示佔用)，然後立即進行一筆資料傳輸交易。若 PORT 1,2 已被佔用 (M9135,M9138 =0)，則本指令進入等待狀態，一直等到佔用之通訊指令傳送完畢或放棄運作，釋出掌控權 (M9135,M9138=1) 後，本指令立即脫離等待狀態，將 M9135,M9138 設為 0 並立即進行傳輸交易。
- 在傳輸交易進行中，若放棄運作“ABT”變為 1，則本指令將立即停止傳輸，並釋出掌控權 (將 M9135,M9138 設為 1)。當本指令回復運作，並再次掌控 PORT 1,2 時，會重頭由第一筆資料開始傳輸。
- “A/R”=0，MODBUS RTU 通訊協定；“A/R”=1，MODBUS ASCII 通訊協定 (保留)。
- 當資料交易傳輸中，輸出指示“ACT”ON。
- 當一筆資料交易傳輸完，如有錯誤發生，則輸出指示“DN”與“ERR”同時 ON。
- 當一筆資料交易傳輸完，如無錯誤發生，則輸出指示“DN”ON。
- 詳細應用範例請參考功能與應用手冊第 11 章“乙太網路功能及乙太網路通訊”。

7-19-2 通訊連線便利指令(CLINK)

FUN151 CLINK	FUN151 (CLINK) : 通訊連線便利指令 (PLC 透過連接埠 1~4 作為 CPU Link 網路中的主站)	FUN151 CLINK																													
指令說明																															
階段圖符號 151P.CLINK 執行控制 — EN 暫停運作 — PAU 放棄運作 — ABT Pt : MD : SR : WR : ACT — 通訊傳輸 ERR — 通訊錯誤 DN — 通訊完成		Pt : 指定通訊埠 , 1~4 MD : 通訊模式選擇(MD0~MD5) #MD3 只支援 Port2. SR : 存放通訊程式起始暫存器 WR : 指令運作起始暫存器 , 共佔 用 8 個暫存器 , 其它程式不可重複使用。																													
<table><tr><td rowspan="2">運算元 範圍</td><td>HR</td><td>ROR</td><td>DR</td><td>K</td></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td></td></tr><tr><td>Pt</td><td></td><td></td><td></td><td>1 ~ 4</td></tr><tr><td>MD</td><td></td><td></td><td></td><td>0 ~ 5</td></tr><tr><td>SR</td><td>○</td><td>○</td><td>○</td><td></td></tr><tr><td>WR</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>			運算元 範圍	HR	ROR	DR	K	R0 R34767	R43224 R47319	D0 D7999		Pt				1 ~ 4	MD				0 ~ 5	SR	○	○	○		WR	○	○*	○	
運算元 範圍	HR	ROR		DR	K																										
	R0 R34767	R43224 R47319	D0 D7999																												
Pt				1 ~ 4																											
MD				0 ~ 5																											
SR	○	○	○																												
WR	○	○*	○																												
功能敘述																															
● 本指令為 MD0~MD5 通用通訊連線便利指令 , 客戶可自視需求 , 指定通訊模式 (MD0~MD5)。 ● FUN151 (CLINK) : MD 0 , 使 PLC 透過 Port 1~2 作為 FATEK CPU Link Network 的主站。 ● 一台主 PLC 可經由 RS-485 介面最多與 254 台僕 PLC 連線互享資料。 ● 僅主 PLC 需使用 CLINK 指令 (設為階段圖指令掌控界面) , 其他所有僕 PLC 均不必 (設在標準界面)。 ● 利用程式書寫方式或填表格方式來規劃資料流控制 ; 亦即要從那一台僕 PLC 讀取何種型態資料存放到主 PLC , 或從主 PLC 要寫何種資料至僕 PLC , 僅需利用七個暫存器來定義 , 每七個暫存器定義一筆傳輸交易。																															
FUN151 CLINK	FUN151 (CLINK) : 通訊連線便利指令 (使 PLC 經由 Port 1,2 當作永宏通訊協定之主站)	FUN151 CLINK																													

- 當執行控制“EN”由 0→1 且暫停運作“PAU”與放棄運作“ABT”均為 0 時，若指定之通訊埠未被其他通訊指令佔用 [M9135(Port1),M9138(Port2) = 1]，則本指令立即掌控該通訊埠，並將 M9135,M9138 設為 0 (表示佔用)，然後立即進行一筆資料傳輸交易。若指定之通訊埠已被佔用 (M9135,M9138 =0)，則本指令進入等待狀態，一直等到佔用之通訊指令傳送完畢或暫停 / 放棄運作，釋出掌控權後 (M9135,M9138 =1)，本指令立即脫離等待狀態，將 M9135,M9138 設為 0，並立即進行傳輸交易。
- 在傳輸交易進行中，若暫停運作“PAU”變為 1，則本指令將在當時正在傳輸之那筆交易資料傳輸完畢後，暫停運作並釋出掌控權。而等到本指令回復運作並再次掌控傳輸權時，將會接續上次暫停傳輸之下一筆資料開始傳輸 (亦即暫停是以一筆完整之交易資料為單位)。
- 傳輸交易進行中，若放棄運作“ABT”變為 1，則本指令將立即停止傳輸，並釋出掌控權。當本指令回復運作，並再次掌控通訊埠時，會重頭由第一筆資料開始傳輸。
- 當資料交易傳輸中，輸出指示“ACT”ON。
- 當一筆資料交易傳輸完，如有錯誤發生，則輸出指示“DN”與“ERR”同時 ON。
- 當一筆資料交易傳輸完，如無錯誤發生，則輸出指示“DN”ON。
- 詳細應用範例請參考功能與應用手冊第 10.4 章“Fatek binary 主從通訊協定用”。

7-19-3 Modbus 網路通訊連線便利指令(NCR)

FUN152 NCR	Modbus 網路(主站)通訊連線便利指令										FUN152 NCR																																																										
指令說明																																																																					
執行傳輸 – EN 152P.NCR SR: MD: WR: ACT – 傳輸中 ERR – 錯誤 DN – 完成						SR: 表格起始暫存器位址。 MD: Modbus TCP 主動通訊(=1)。 WR: 工作暫存器。																																																															
<table><tr><td rowspan="3">範圍 運算元</td><td>WY</td><td>WM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td></tr><tr><td>WY0 WY1008</td><td>WM0 WM25584</td><td>WS0 WS3088</td><td>T0 T1024</td><td>C0 C1280</td><td>R0 R34767</td><td>R35024 R35151</td><td>R35280 R40279</td><td>R40280 R48471</td><td>D0 D7999</td><td></td></tr><tr><td>SR</td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td></tr><tr><td>MD</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr><tr><td>WR</td><td></td><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td></tr></table>												範圍 運算元	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1024	C0 C1280	R0 R34767	R35024 R35151	R35280 R40279	R40280 R48471	D0 D7999		SR			○	○	○	○	○*	○*	○		MD											1	WR				○	○	○	○	○*	○*	○	
範圍 運算元	WY	WM	WS	TMR	CTR	HR	OR	SR	ROR	DR	K																																																										
	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1024	C0 C1280	R0 R34767	R35024 R35151	R35280 R40279	R40280 R48471	D0 D7999																																																											
	SR			○	○	○	○	○*	○*	○																																																											
MD											1																																																										
WR				○	○	○	○	○*	○*	○																																																											
功能敘述																																																																					
1. FUN152 (NCR) 指令透過乙太網路埠以 Modbus 通訊協定與智慧外設 (從站) 連線。 2. 此指令主要是根據指定的形式，如使用 Modbus Master TCP 形式，依照指定的形式進行讀取或寫入，主動進行網路通訊。使用前必須設定通訊形式；僅定義了 6 個暫存器，每 6 個暫存器定義一個傳輸事務。 3. 當此指令的 EN 為 ON 時，通訊將會繼續。 4. 當資料傳輸過程中，輸出指示「ACT」為 ON。 5. 更多詳細信息，請參閱功能和應用程式使用手冊。																																																																					

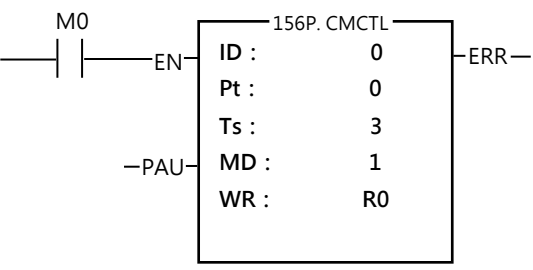
FUN152 NCR	Modbus 網路通訊協定(主站)通訊連線便利指令	FUN152 NCR						
程式範例	僕站 (IP : 192.168.0.151) 400101~ 400105 -> 主站 (IP : 192.168.0.150) R100~R104							
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td><pre>graph LR M1((M1)) -- EN --> NCR[152. NCR] NCR -- ACT --> M11((M11)) NCR -- ERR --> M12((M12)) NCR -- DN --> M13((M13)) subgraph NCR_Parameters [] direction TB SR[SR : R1000] MD[MD : 1] WR[WR : R1500] end</pre></td><td><pre>NCR(EN:= M1, SR:= R1000, MD:= 1, WR:= R1500, ACT=> M11, ERR=> M12, DN=> M13);</pre></td></tr></table>			階段圖	ST	<pre>graph LR M1((M1)) -- EN --> NCR[152. NCR] NCR -- ACT --> M11((M11)) NCR -- ERR --> M12((M12)) NCR -- DN --> M13((M13)) subgraph NCR_Parameters [] direction TB SR[SR : R1000] MD[MD : 1] WR[WR : R1500] end</pre>	<pre>NCR(EN:= M1, SR:= R1000, MD:= 1, WR:= R1500, ACT=> M11, ERR=> M12, DN=> M13);</pre>		
階段圖	ST							
<pre>graph LR M1((M1)) -- EN --> NCR[152. NCR] NCR -- ACT --> M11((M11)) NCR -- ERR --> M12((M12)) NCR -- DN --> M13((M13)) subgraph NCR_Parameters [] direction TB SR[SR : R1000] MD[MD : 1] WR[WR : R1500] end</pre>	<pre>NCR(EN:= M1, SR:= R1000, MD:= 1, WR:= R1500, ACT=> M11, ERR=> M12, DN=> M13);</pre>							
<table><tr><td>程式範例說明</td></tr><tr><td>當輸入控制“EN”由 0→1，基於 Modbus TCP 表格中的設定，從遠端 IP 僕站讀取暫存器資料存放到 PLC 主站，連續達成資料的交易。</td></tr><tr><td>設定步驟，如下說明。</td></tr><tr><td>首先新增資料表格中的 Modbus Master 表格。</td></tr><tr><td>在 Modbus Master 表格中定義遠端 IP 和 Port，及欲讀取和寫入的位址，包含主站資料和僕站資料。</td></tr><tr><td>在主站 Ladder 編輯 Fun152 NCR 指令。</td></tr></table>			程式範例說明	當輸入控制“EN”由 0→1，基於 Modbus TCP 表格中的設定，從遠端 IP 僕站讀取暫存器資料存放到 PLC 主站，連續達成資料的交易。	設定步驟，如下說明。	首先新增資料表格中的 Modbus Master 表格。	在 Modbus Master 表格中定義遠端 IP 和 Port，及欲讀取和寫入的位址，包含主站資料和僕站資料。	在主站 Ladder 編輯 Fun152 NCR 指令。
程式範例說明								
當輸入控制“EN”由 0→1，基於 Modbus TCP 表格中的設定，從遠端 IP 僕站讀取暫存器資料存放到 PLC 主站，連續達成資料的交易。								
設定步驟，如下說明。								
首先新增資料表格中的 Modbus Master 表格。								
在 Modbus Master 表格中定義遠端 IP 和 Port，及欲讀取和寫入的位址，包含主站資料和僕站資料。								
在主站 Ladder 編輯 Fun152 NCR 指令。								

7-19-4 串列通訊模組控制指令(CMCTL)

FUN 156P CMCTL	CMCTL	FUN156P CMCTL
指令說明		
執行控制 — EN 通訊暫停 — PAU 156P.CMCTL ID  Pt  Ts  MD  WR  ERR — 通訊錯誤 ID:使用的模組編號 Pt:指定 COMA/COMB (A= 0;B=1) Ts:通訊表格遮罩 Bit 1:表格 0 Bit 2:表格 1 Bit 15:表格 15 Bit 16~Bit 31: 保留，不可使用 MD:設定模式 0:RUN ONCE 1:RUN CYCLING 2:STOP WR:存放工作狀態 Bit0~Bit1:表格 0 state Bit2~Bit3:表格 1 state Bit30~Bit31:表格 15 state (= 0:RUN_ONCE = 1:RUN_CYCLING = 2:STOP)		

FUN156P CMCTL	CMCTL						FUN156P CMCTL																																															
<table><tr><th rowspan="2">運算元 範圍</th><th>HR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><td>R0 R34767</td><td>R35024 R35151</td><td>R35280 R43323</td><td>R43224 R47319</td><td>D0 D7999</td><td></td></tr><tr><td>ID</td><td></td><td></td><td></td><td></td><td></td><td>0~127</td></tr><tr><td>Pt</td><td></td><td></td><td></td><td></td><td></td><td>0~23</td></tr><tr><td>Ts</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0 ~ 63</td></tr><tr><td>MD</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0 ~ 63</td></tr><tr><td>WR</td><td>○</td><td></td><td></td><td>○</td><td>○</td><td>5word</td></tr></table>							運算元 範圍	HR	OR	SR	ROR	DR	K	R0 R34767	R35024 R35151	R35280 R43323	R43224 R47319	D0 D7999		ID						0~127	Pt						0~23	Ts	○	○	○	○	○	0 ~ 63	MD	○	○	○	○	○	0 ~ 63	WR	○			○	○	5word
運算元 範圍	HR	OR	SR	ROR	DR	K																																																
	R0 R34767	R35024 R35151	R35280 R43323	R43224 R47319	D0 D7999																																																	
ID						0~127																																																
Pt						0~23																																																
Ts	○	○	○	○	○	0 ~ 63																																																
MD	○	○	○	○	○	0 ~ 63																																																
WR	○			○	○	5word																																																
功能敘述	此指令為搭配通訊模組 HCM25 與 HCM55 使用，使用前需使用軟體配置相關設定																																																					
● EN OFF->ON 時會進行通訊控制，ON->OFF 會停止 ● PAU 尚未支援 ● 各表格的通訊狀態碼會更新於分配的狀態暫存器中，可使用設備視圖確認位址																																																						
FUN156：WR 之說明																																																						
Bit0	Table0 狀態		Table0 = 0(00): RUN_ONCE = 1(01): RUN_CYCLING = 2(10): STOP																																																			
Bit1																																																						
Bit2	Table1 狀態		Table1 = 0(00): RUN_ONCE = 1(01): RUN_CYCLING = 2(10): STOP																																																			
Bit3																																																						
Bit4	Table2 狀態		Table2 = 0(00): RUN_ONCE = 1(01): RUN_CYCLING = 2(10): STOP																																																			
Bit5																																																						
Bit6	Table3 狀態		Table3 = 0(00): RUN_ONCE = 1(01): RUN_CYCLING = 2(10): STOP																																																			
Bit7																																																						
Bit30	Table15 狀態		Table15 = 0(00): RUN_ONCE = 1(01): RUN_CYCLING = 2(10): STOP																																																			
Bit31																																																						
Bit32 以後保留																																																						

程式範例

階段圖	ST
	<pre>CMCTL(EN:= M0, PAU:= M1, ID:= 0, Pt:= 0, Ts:= 3, MD:= 1, WR:= R0, ERR=> M2);</pre>

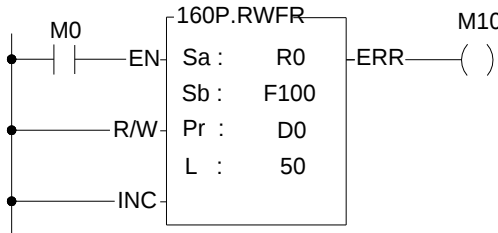
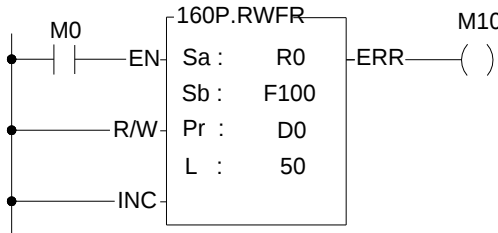
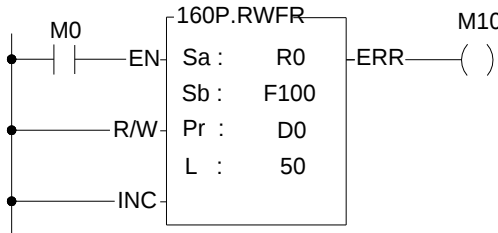
以上圖為例，當 M0 為 1 時，會將第 1 個模組的 Port 0 開啟，並控制表格 1 與表格 2 進行通訊 (0001b+0010b=0011b 等於 3(10))，MD 選擇 RUN CYCLING 模式表示持續通訊，R0 則存放各表格當前模式。

7-20 搬移指令二 (FUN160 ~ 162)

7-20-1 讀/寫檔案暫存器 (Read/Write File Register)

FUN160DP RWFR		讀/寫檔案暫存器 (Read/Write File Register)												FUN160DP RWFR																																																																																							
指令說明																																																																																																					
		階段圖符號 執行控制 — EN 讀取/寫入選擇 — R/W 指標遞增 — INC 160DP.RWFR Sa : Sb : Pr : L : ERR — 範圍錯誤								Sa：暫存器列表之起始暫存器號碼 Sb：檔案暫存器之起始號碼 Pr：指標暫存器號碼 L：列表之長度 1~511 Sa 可結合 V、Z、P0~P9 作間接定址應用																																																																																											
運算元 範圍 <table><tr><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th><th>FR</th></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM25584</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V、Z P0 ~ P9</td><td>F0 F32767</td></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td></td></tr><tr><td>Sb</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td></tr><tr><td>Pr</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td><td>○*</td><td>○*</td><td>○</td><td></td><td></td></tr><tr><td>L</td><td></td><td></td><td></td><td></td><td></td><td></td><td>○</td><td></td><td></td><td></td><td>○*</td><td>○</td><td>1~511</td><td></td></tr></table>		WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	FR	WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0 ~ P9	F0 F32767	Sa	○	○	○	○	○	○	○	○	○	○	○		○		Sb														○	Pr		○	○	○	○	○	○		○	○*	○*	○			L							○				○*	○	1~511											
WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	FR																																																																																							
WX0 WX1008	WY0 WY1008	WM0 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0 ~ P9	F0 F32767																																																																																							
Sa	○	○	○	○	○	○	○	○	○	○	○		○																																																																																								
Sb														○																																																																																							
Pr		○	○	○	○	○	○		○	○*	○*	○																																																																																									
L							○				○*	○	1~511																																																																																								
功能敘述																																																																																																					
		● 當執行控制“EN” =1 或由 0→1 (P 指令) 時，自暫存器 Sa 開始，將長度 L 的資料依讀寫控制“R/W”決定對檔案暫存器進行讀出或寫入之動作；“R/W” =1 為讀出檔案暫存器，“R/W” =0 為寫入檔案暫存器。讀取時，從 Sa 開始的資料暫存器的內容將被基檔暫存器 Sb 和記錄指標 Pr 所定址的檔案暫存器的內容覆蓋；寫入時，基檔暫存器 Sb 和記錄指標 Pr 所定址的檔案暫存器的內容將會被從 Sa 開始的資料暫存器的內容覆蓋。本指令以資料結構之 Record 觀念執行之，亦即 Pr 指標所指的是每筆長度為 L 的區塊，舉例來說若 Sa=R0，Sb=F0，L=10，當 Pr 指標值為 2 時，那麼執行讀寫的區域就是 F20~F29，示意圖如下：																																																																																																			

FUN160DPP RWFR	讀/寫檔案暫存器 (Read/Write File Register)	FUN160DPP RWFR
Sa R0 ~ R9 (L=10) ↔ Sb F0 ~ F9 (L=10) F10 ~ F19 (L=10) F20 ~ F29 (L=10) F30 ~ F39 (L=10) • • • • • • ← Pr = 0 ← Pr = 1 ← Pr = 2 ← Pr = 3		
● 本指令讀寫之暫存器，為系統內部之“檔案暫存器”，這些暫存器無法由其它的功能指令存取，唯有本指令才可對其讀寫。 ● 若指標遞增“INC”=1，則每次執行完本指令之後，指標暫存器 Pr 之內容值加 1，也就是說指向下一個長度為 L 之記憶體區塊。 ● 若長度為 0 或大於 511 或指標長度超出檔案暫存器範圍 F0~F32767，則“指標錯誤”ERR 設為 1，本指令不執行。		
程式範例一		
階梯圖 M0 ● EN 160P.RWFR Sa: R0 Sb: F100 Pr: D0 L: 50 ERR () M10 INC ● —R/W ST ReadWriteFileReg(EN:=M0, R_W:= FALSE, INC:= TRUE, Sa:= R0, Sb:= F100, Pr:= D0, L:= 50);		
● 當 M0 由 0→1 時，若 D0=2，則將 R0~R49 內的資料，搬移至 F200~F249 內覆蓋之。 ● 執行完本指令之後，指標暫存器自動加 1。		

FUN160DPRWFR	讀/寫檔案暫存器 (Read/Write File Register)	FUN160DPRWFR				
程式範例二						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>ReadWriteFileReg(EN:=M0, R_W:= TRUE, INC:= TRUE, Sa:= R0, Sb:= F100, Pr:= D0, L:= 50);</pre></td></tr></table>			階段圖	ST		<pre>ReadWriteFileReg(EN:=M0, R_W:= TRUE, INC:= TRUE, Sa:= R0, Sb:= F100, Pr:= D0, L:= 50);</pre>
階段圖	ST					
	<pre>ReadWriteFileReg(EN:=M0, R_W:= TRUE, INC:= TRUE, Sa:= R0, Sb:= F100, Pr:= D0, L:= 50);</pre>					
● 當 M0 由 0→1 時，若 D0=1，則將 F150~F199 內之資料。搬移至 R0~R49 內覆蓋之。 ● 執行完本指令之後，指標暫存器自動加 1。						

7-20-2 寫入資料至資料記憶匣 (Write SD Card)

FUN161P WR-MP	寫入資料至 SD 卡 (Write SD Card)	FUN161P WR-MP																																																
指令說明																																																		
階梯圖符號 161P.WR-MP 執行控制 — EN — S : Bk : 指標遞增 — INC — Os : Pr : L : WR : ACT — ERR — DN —		S：寫入資料之來源起始暫存器號碼 BK：SD 卡之區塊號碼，0 ~ 1 Os：分區資料起始位置 Pr：指標暫存器號碼 L：寫入資料長度 1 ~ 128 WR：工作暫存器起始號碼，共佔用 2 個暫存器 S 可結合 V、Z、P0 ~ P9 作間接定址應用																																																
<table><tr><th> 範圍 運算元 </th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V、Z P0 ~ P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>Bk</td><td></td><td></td><td></td><td>0 ~ 1</td><td></td></tr><tr><td>Os</td><td>○</td><td>○</td><td>○</td><td>0 ~ 32510</td><td></td></tr><tr><td>Pr</td><td>○</td><td>○*</td><td>○</td><td></td><td></td></tr><tr><td>L</td><td>○</td><td>○*</td><td>○</td><td>1 ~ 128</td><td></td></tr><tr><td>WR</td><td>○</td><td>○*</td><td>○</td><td></td><td></td></tr></table>			範圍 運算元	HR	ROR	DR	K	XR		R0 R34767	R43224 R47319	D0 D7999		V、Z P0 ~ P9	S	○	○	○		○	Bk				0 ~ 1		Os	○	○	○	0 ~ 32510		Pr	○	○*	○			L	○	○*	○	1 ~ 128		WR	○	○*	○		
範圍 運算元	HR	ROR	DR	K	XR																																													
	R0 R34767	R43224 R47319	D0 D7999		V、Z P0 ~ P9																																													
S	○	○	○		○																																													
Bk				0 ~ 1																																														
Os	○	○	○	0 ~ 32510																																														
Pr	○	○*	○																																															
L	○	○*	○	1 ~ 128																																														
WR	○	○*	○																																															
功能敘述																																																		

FUN161P WR-MP	寫入資料至 SD 卡 (Write SD Card)	FUN161P WR-MP
------------------	---------------------------------	------------------

以S為起始，長度為L之資料RECORD

寫入

Os = 0 →

區塊0	區塊1
區塊0檔頭資料	區塊1檔頭資料
長度為L之資料 RECORD 0	長度為L之資料 RECORD 0
長度為L之資料 RECORD 1	長度為L之資料 RECORD 1
長度為L之資料 RECORD 2	長度為L之資料 RECORD 2
• • • • •	• • • • •

← Pr = 0

← Pr = 1

← Pr = 2

← Pr = N

Os = 32510 →

● M Serial PLC 之 SD 卡除了可用來儲存階梯圖控制程式外，尚可透過本指令用來當作 SD 卡(SD Card)以作為可攜式機台生產成型資料之存取裝置。當執行控制“EN”由 0→1 時，自暫存器 S 開始，將長度 L 的資料寫入所指定 SD 卡之區塊(Bk)內，由分區資料起始位置(Os)加指標所指位址開始寫入。本指令以資料結構之 RECORD 觀念執行之，亦即 Pr 指標所指的是每筆長度為 L 的 RECORD，透過本指令將其儲存至 SD 卡內。本指令執行示意圖如下：

● 若指標遞增“INC”=1，則每次執行完本指令之後，指標暫存器 Pr 之內容值加 1，也就是說指向下下一個長度為 L 之 RECORD。

● 若長度為 0 或大於 128 或指標所指超出範圍，則錯誤指示"ERR"設為 1，本指令不執行。

FUN161P WR-MP	寫入資料至 SD 卡 (Write SD Card)	FUN161P WR-MP				
● 本指令在執行資料寫入與寫入資料比對過程中有可能會需要多次掃描時間才能完成；在寫入執行過程中時，輸出指示"ACT"為 1；當寫入完成且寫入資料比對無誤時，輸出指示"DN"為 1；當寫入完成但寫入資料比對有誤時，輸出指示"ERR"為 1。 ● M series 之 SD 卡可規劃為程式儲存裝置或當作機台生產成型資料記憶裝置，或兩者兼具；階梯圖控制程式固定儲存在區塊 0，而生產成型資料則可選擇儲存在區塊 0 或區塊 1；每個區塊之記憶體容量為 32K Word。						
程式範例一	寫入兩種不同長度之 RECORD 至 SD 卡區塊 1					
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td> M0 M1 ENU INC 161P.WR-MP S : R0 Bk : 1 Os: 0 Pr: D1 L : 20 WR: R2900 ACT ERR DN M100 M101 M102 M2 M3 ENU INC 161P.WR-MP S : R100 Bk : 1 Os: 10000 Pr: D2 L : 50 WR: R2910 ACT ERR DN M103 M104 M105 </td><td> WriteSDMem(EN:= M0, INC:= M1, S:= R0, Bk:= 1, Os:= 0, Pr:= D1, L:= 20, WR:= R2900, ACT=> M100, ERR=> M101, DN=> M102); WriteSDMem(EN:= M2, INC:= M3, S:= R100, Bk:= 1, Os:= 10000, Pr:= D2, L:= 50, WR:= R2910, ACT=> M103, ERR=> M104, DN=> M105); </td></tr></table>			階梯圖	ST	M0 M1 ENU INC 161P.WR-MP S : R0 Bk : 1 Os: 0 Pr: D1 L : 20 WR: R2900 ACT ERR DN M100 M101 M102 M2 M3 ENU INC 161P.WR-MP S : R100 Bk : 1 Os: 10000 Pr: D2 L : 50 WR: R2910 ACT ERR DN M103 M104 M105	WriteSDMem(EN:= M0, INC:= M1, S:= R0, Bk:= 1, Os:= 0, Pr:= D1, L:= 20, WR:= R2900, ACT=> M100, ERR=> M101, DN=> M102); WriteSDMem(EN:= M2, INC:= M3, S:= R100, Bk:= 1, Os:= 10000, Pr:= D2, L:= 50, WR:= R2910, ACT=> M103, ERR=> M104, DN=> M105);
階梯圖	ST					
M0 M1 ENU INC 161P.WR-MP S : R0 Bk : 1 Os: 0 Pr: D1 L : 20 WR: R2900 ACT ERR DN M100 M101 M102 M2 M3 ENU INC 161P.WR-MP S : R100 Bk : 1 Os: 10000 Pr: D2 L : 50 WR: R2910 ACT ERR DN M103 M104 M105	WriteSDMem(EN:= M0, INC:= M1, S:= R0, Bk:= 1, Os:= 0, Pr:= D1, L:= 20, WR:= R2900, ACT=> M100, ERR=> M101, DN=> M102); WriteSDMem(EN:= M2, INC:= M3, S:= R100, Bk:= 1, Os:= 10000, Pr:= D2, L:= 50, WR:= R2910, ACT=> M103, ERR=> M104, DN=> M105);					

FUN161P WR-MP	寫入資料至 SD 卡 (Write SD Card)	FUN161P WR-MP
------------------	---------------------------------	------------------

以R0為起始，長度為20
(R0~R19)之資料RECORD

寫入

Os = 0

以R100為起始，長度為50
(R100~R149)之資料RECORD

寫入

Os = 9999
Os = 10000

SD 卡

區塊1

區塊1檔頭資料

長度為20之資料
RECORD 0

長度為20之資料
RECORD 1

•
•
•
•

長度為20之資料
RECORD 499

長度為50之資料
RECORD 0

•
•
•
•

長度為50之資料
RECORD 449

← Pr = 0

← Pr = 1

← Pr = 499

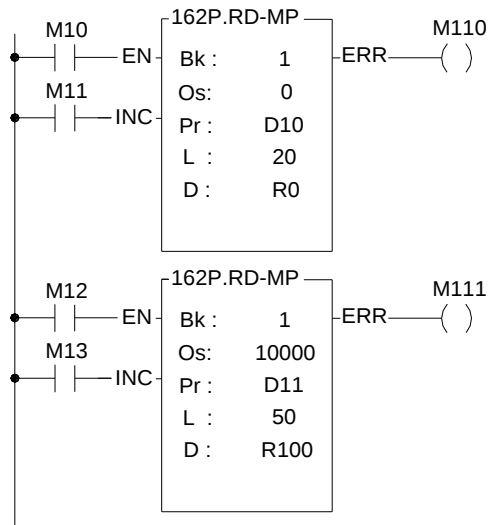
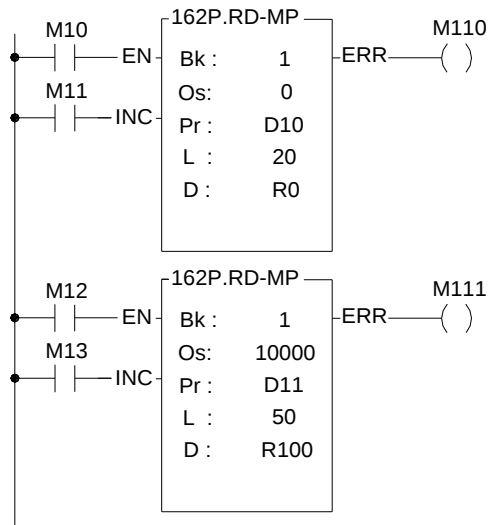
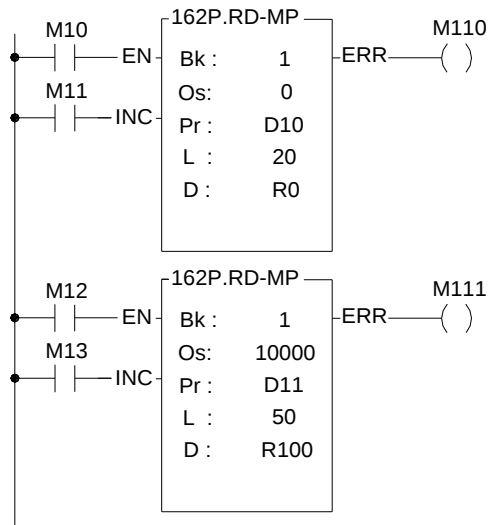
← Pr = 0

← Pr = 449

7-20-3 由資料記憶匣讀取資料 (Read SD Card)

FUN162 P RD-MP	由 SD 卡讀取資料 (Read SD Card)	FUN162 P RD-MP																																		
指令說明																																				
階段圖符號 執行控制 — EN — 指標遞增 — INC — 162P.RD-MP Bk : Os : Pr : L : D : ERR —		BK : SD 卡之區塊號碼 · 0 ~ 1 Os : 分區資料起始位置 Pr : 指標暫存器號碼 L : 讀取資料長度 1 ~ 128 D : 存放讀取資料之暫存器起始號碼																																		
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th></tr><tr><th>R0 R34767</th><th>R43224 R47319</th><th>D0 D7999</th><th></th></tr><tr><td>Bk</td><td></td><td></td><td></td><td>0 ~ 1</td></tr><tr><td>Os</td><td>○</td><td>○</td><td>○</td><td>0 ~ 32510</td></tr><tr><td>Pr</td><td>○</td><td>○*</td><td>○</td><td></td></tr><tr><td>L</td><td>○</td><td>○*</td><td>○</td><td>1 ~ 128</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○</td><td></td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	R0 R34767	R43224 R47319	D0 D7999		Bk				0 ~ 1	Os	○	○	○	0 ~ 32510	Pr	○	○*	○		L	○	○*	○	1 ~ 128	D	○	○*	○	
運算元 \ 範圍	HR	ROR		DR	K																															
	R0 R34767	R43224 R47319	D0 D7999																																	
Bk				0 ~ 1																																
Os	○	○	○	0 ~ 32510																																
Pr	○	○*	○																																	
L	○	○*	○	1 ~ 128																																
D	○	○*	○																																	
功能敘述																																				
● M Serial PLC 之 SD 卡如果儲存有 FUN161 指令所寫入之機台生產成型資料，則可透過本指令將儲存之資料讀出再用，以減少生產調機時間。 ● 當執行控制 “EN” =1 或由 0→1(P 指令)時，將所指定 SD 卡之區塊(BK)內，由分區資料起始位置(Os)加指標所指位址開始，長度為 L 的資料 RECORD 讀出。本指令以資料結構之 RECORD 觀念執行之，亦即 Pr 指標所指的是每筆長度為 L 的 RECORD，透過本指令將其由 SD 卡內讀出。本指令執行示意圖如下：																																				

FUN162 P RD-MP	由 SD 卡讀取資料 (Read SD Card)	FUN162 P RD-MP																		
SD 卡 <table><thead><tr><th>區塊0</th><th>區塊1</th></tr></thead><tbody><tr><td>區塊0檔頭資料</td><td>區塊1檔頭資料</td></tr><tr><td>長度為L之資料 RECORD 0</td><td>長度為L之資料 RECORD 0</td></tr><tr><td>長度為L之資料 RECORD 1</td><td>長度為L之資料 RECORD 1</td></tr><tr><td>長度為L之資料 RECORD 2</td><td>長度為L之資料 RECORD 2</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>⋮</td><td>⋮</td></tr><tr><td>⋮</td><td>⋮</td></tr></tbody></table> Os = 0 → ← Pr = 0 ← Pr = 1 ← Pr = 2 ← Pr = N Os = 32510 → 讀取 以D為起始，長度為L之資料RECORD			區塊0	區塊1	區塊0檔頭資料	區塊1檔頭資料	長度為L之資料 RECORD 0	長度為L之資料 RECORD 0	長度為L之資料 RECORD 1	長度為L之資料 RECORD 1	長度為L之資料 RECORD 2	長度為L之資料 RECORD 2	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
區塊0	區塊1																			
區塊0檔頭資料	區塊1檔頭資料																			
長度為L之資料 RECORD 0	長度為L之資料 RECORD 0																			
長度為L之資料 RECORD 1	長度為L之資料 RECORD 1																			
長度為L之資料 RECORD 2	長度為L之資料 RECORD 2																			
⋮	⋮																			
⋮	⋮																			
⋮	⋮																			
⋮	⋮																			
● 若指標遞增“INC”=1，則每次執行完本指令之後，指標暫存器 Pr 之內容值加 1，也就是說指向下一個長度為 L 之 RECORD。 ● 若長度為 0 或大於 128 或指標所指超出範圍，則錯誤指示"ERR"設為 1，本指令不執行。當 SD 卡內無資料或資料格式不正確以致 FUN162 無法讀取資料時，錯誤指示"ERR"亦設為 1，且本指令不執行。																				

FUN162 P RD-MP	由 SD 卡讀取資料 (Read SD Card)	FUN162 P RD-MP				
程式範例一	由 SD 卡區塊 1 讀取兩種不同長度之 RECORD					
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td><pre>ReadSDMem (EN:=M10, INC:= M11, Bk:= 1, Os:=0, Pr:= D10, L:= 20, D:= R0, ERR=> M110); ReadSDMem (EN:=M12, INC:= M13, Bk:= 1, Os:= 10000, Pr:= D11, L:= 50, D:= R100, ERR=> M111);</pre></td></tr></table>			階段圖	ST		<pre>ReadSDMem (EN:=M10, INC:= M11, Bk:= 1, Os:=0, Pr:= D10, L:= 20, D:= R0, ERR=> M110); ReadSDMem (EN:=M12, INC:= M13, Bk:= 1, Os:= 10000, Pr:= D11, L:= 50, D:= R100, ERR=> M111);</pre>
階段圖	ST					
	<pre>ReadSDMem (EN:=M10, INC:= M11, Bk:= 1, Os:=0, Pr:= D10, L:= 20, D:= R0, ERR=> M110); ReadSDMem (EN:=M12, INC:= M13, Bk:= 1, Os:= 10000, Pr:= D11, L:= 50, D:= R100, ERR=> M111);</pre>					

FUN162 P RD-MP	由 SD 卡讀取資料 (Read SD Card)	FUN162 P RD-MP
程式範例一	由 SD 卡區塊 1 讀取兩種不同長度之 RECORD	

※ SD 卡 內需有相符資料格式之資料，否則本範例無法執行。

以R0為起始，長度為20
(R0~R19)之資料RECORD

讀取

以R100為起始，長度為
50(R100~R149)之資料
RECORD

讀取

SD 卡

區塊1

區塊1檔頭資料

長度為20之資料
RECORD 0

長度為20之資料
RECORD 1

⋮

長度為20之資料
RECORD 499

長度為50之資料
RECORD 0

⋮

長度為50之資料
RECORD 449

Os = 0

Os = 9999

Os = 10000

Os = 32510

← Pr = 0

← Pr = 1

← Pr = 499

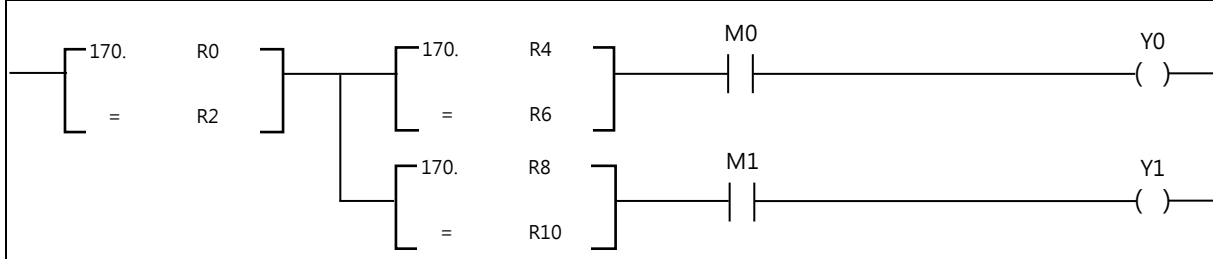
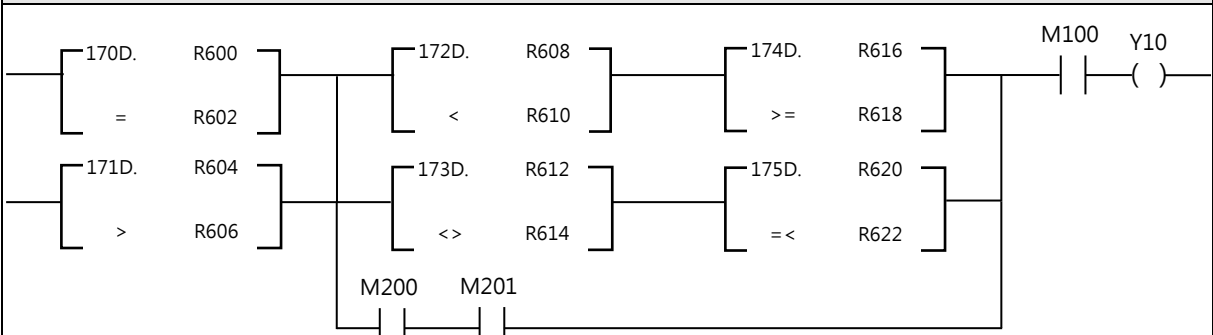
← Pr = 0

← Pr = 449

7-21 接點型比較指令 (FUN170 ~ 175)

7-21-1 相等比較指令 (比較 Sa 是否等於 Sb)

FUN170 D = <	
---	--

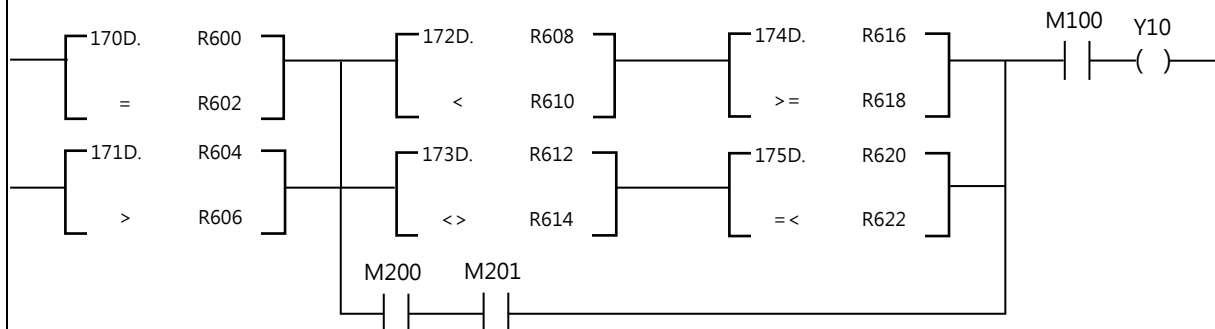
FUN170 D	相等比較指令 (比較 Sa 是否等於 Sb)	FUN170 D
=		=
程式範例 1	當 R0=R2、R4=R6 且 M0=1 時，Y0 輸出為 1；否則，Y0 輸出為 0。 R0=R2、R8=R10 且 M1=1 時，Y1 輸出為 1；否則，Y1 輸出為 0。	
階段圖  ST IF R0 = R2 AND R4 = R6 AND M0 Then Y0 := TRUE; ELSEIF R0 = R2 AND R8 = R10 AND M1 Then Y1 := TRUE; ELSE Y0 := FALSE; Y1 := FALSE; END_IF		
程式範例 2	當 DR600 = DR602 或 DR604 > DR606、且 DR608 < DR610 及 DR616 ≥ DR618 或 DR612 ≠ DR614 及 DR620 ≤ DR622 或 M200 = 1 及 M201 = 1、且 M100 = 1 時，Y10 輸 出 1；否則，Y10 輸出為 0。	
階段圖  ST IF R600 = R602 OR R604 > R606 Then IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 <= R622) OR (M200 AND M201) Then IF M100 Then Y10 := TRUE; END_IF END_IF ELSE Y10 := FALSE; END_IF		

7-21-2 大於比較指令 (比較 Sa 是否大於 Sb)

FUN171 D >		大於比較指令 (比較 Sa 是否大於 Sb)														FUN171 D >			
指令說明																			
		執行控制 EN 171D. > Sa Sb										Sa : 比較值 a 或其暫存器號碼 Sb : 比較值 b 或其暫存器號碼 Sa、Sb 可結合 V、Z、P0~P9 作間接定址應用							
運算元	範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR			
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9			
		Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
		Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
功能敘述																			
		● 當執行控制“EN”=1 時，本指令以正負數(Signed)運算法則執行 Sa 與 Sb 的比較。若 Sa > Sb，則解題輸出為 1；若否，則解題輸出為 0。																	
程式範例 1																			
		階段圖 M10 171. > R20 R22 M11 Y2 () ST IF M10 AND (R20 > R22 OR M11) Then Y2 := TRUE; ELSE Y2 := FALSE; END_IF																	
		範例說明：當 M10=1、R20 > R22 或 M11=1 時，Y2 輸出為 1；否則，Y2 輸出為 0。																	

程式範例 2

階梯圖



ST

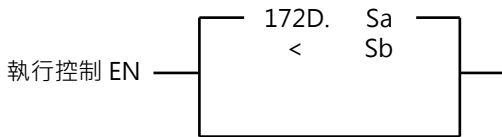
```

IF R600 = R602 OR R604 > R606 Then
IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 =< R622) OR
(M200 AND M201) Then
IF M100 Then
Y10 := TRUE;
END_IF
END_IF
ELSE
Y10 := FALSE;
END_IF

```

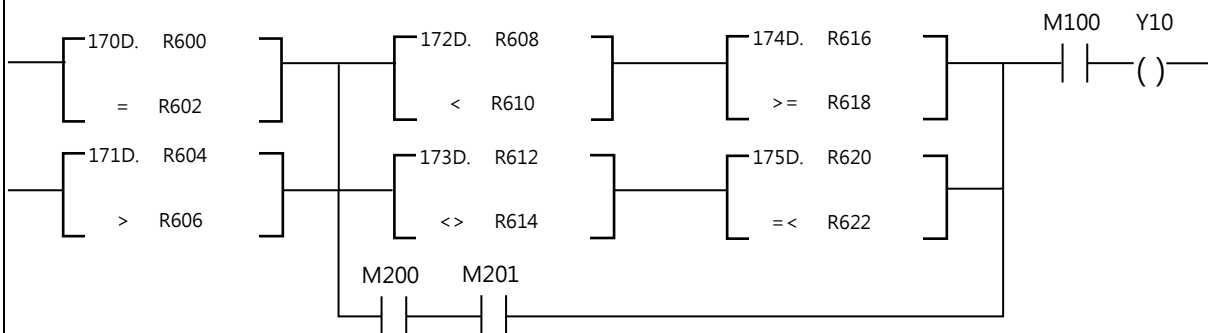
範例說明：當 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≥DR618 或 DR612≠DR614 及 DR620≤DR622 或 M200=1 及 M201=1、而且 M100=1 時，Y10 輸出為 1；否則，Y10 輸出為 0。

7-21-3 小於比較指令 (比較 Sa 是否小於 Sb)

FUN172D <		小於比較指令 (比較 Sa 是否小於 Sb)														FUN172D <			
指令說明																			
												Sa : 比較值 a 或其暫存器號碼 Sb : 比較值 b 或其暫存器號碼 Sa、Sb 可結合 V、Z、P0 ~ P9 作間接定址應用							
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR				
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9				
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○				
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○				
功能敘述																			
		當執行控制“EN”=1 時，本指令以正負數(Signed)運算法則執行 Sa 與 Sb 的比較。若 Sa < Sb，則解題輸出為 1；若否，則解題輸出為 0。																	
程式範例 1																			
階梯圖  ST <pre>IF M10 AND (R20 < R22 OR M11) Then Y2 := TRUE; ELSE Y2 := FALSE; END_IF</pre>																			
範例說明：當 M10=1、R20 < R22 或 M11=1 時，Y2 輸出為 1；否則，Y2 輸出為 0。																			

程式範例 2

階梯圖



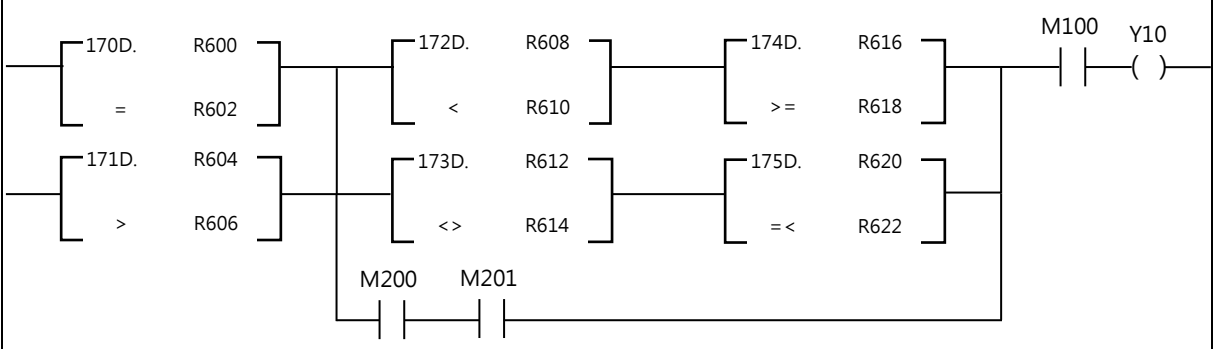
ST

```

IF R600 = R602 OR R604 > R606 Then
IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 =<
R622) OR (M200 AND M201) Then
IF M100 Then
Y10 := TRUE;
END_IF
END_IF
ELSE
Y10 := FALSE;
END_IF

```

範例說明：當 DR600=DR602 或 DR604>DR606 時，其後 DR608<DR610 且 DR616≥DR618，或 DR612≠DR614 且 DR620≤DR622，或 M200=1 且 M201.為 1;否則為 0。

FUN172 D <	小於比較指令 (比較 Sa 是否小於 Sb)	FUN172 D <
階梯圖  ST <pre> IF R600 = R602 OR R604 > R606 Then IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 <= R622) OR (M200 AND M201) Then IF M100 Then Y10 := TRUE; END_IF END_IF ELSE Y10 := FALSE; END_IF </pre> 範例說明：當 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≥DR618 或 DR612≠DR614 及 DR620≤DR622 或 M200=1 及 M201=1、而且 M100=1 時，Y10 輸出為 1；否則，Y10 輸出為 0。		

7-21-4 不相等比較指令 (比較 Sa 是否不等於 Sb)

FUN173 D		不相等比較指令 (比較 Sa 是否不等於 Sb)												FUN173 D	
<>															
指令說明															
執行控制 EN		173D. Sa <> Sb												Sa：比較值 a 或其暫存器號碼 Sb：比較值 b 或其暫存器號碼 Sa、Sb 可結合 V、Z、P0~P9 作間接定址應用	
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9
	Sa	○	○	○	○	○	○	○	○	○	○	○	○	○	○
	Sb	○	○	○	○	○	○	○	○	○	○	○	○	○	○
功能敘述															
		● 當執行控制“EN”=1 時，本指令以正負數(Signed)運算法則執行 Sa 與 Sb 的比較。若 Sa≠Sb，則解題輸出為 1；若否，則解題輸出為 0。													
程式範例一															

FUN173 D <>	不相等比較指令 (比較 Sa 是否不等於 Sb)	FUN173 D <>
程式範例 2		
階梯圖 ST <pre> IF R600 = R602 OR R604 > R606 Then IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 <= R622) OR (M200 AND M201) Then IF M100 Then Y10 := TRUE; END_IF END_IF ELSE Y10 := FALSE; END_IF </pre>		
範例說明：當 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≥DR618 或 DR612≠DR614 及 DR620≤DR622 或 M200=1 及 M201=1、而且 M100=1 時，Y10 輸出為 1；否則，Y10 輸出為 0。		

7-21-5 大於或等於比較指令 (比較 Sa 是否大於或等於 Sb)

FUN174 D >=		大於或等於比較指令 (比較 Sa 是否大於或等於 Sb)														FUN174 D >=	
指令說明																	
		執行控制 EN 174D. >= Sa Sb								Sa：比較值 a 或其暫存器號碼 Sb：比較值 b 或其暫存器號碼 Sa、Sb 可結合 V、Z、P0~P9 作間接定址應用							
範圍 運算元	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9		
	Sa																
	Sb																
功能敘述																	
		● 當執行控制“EN”=1 時，本指令以正負數(Signed)運算法則執行 Sa 與 Sb 的比較。若 Sa≥ Sb，則解題輸出為 1；若否，則解題輸出為 0。															
程式範例 1																	
階段圖																	
M10 171. >= R20 R22 M11 Y2 ()																	
ST																	
IF M10 AND (R20 >= R22 OR M11) Then Y2 := TRUE; ELSE Y2 := FALSE; END_IF																	
範例說明：當 M10=1、R20≥R22 或 M11=1 時，Y2 輸出為 1；否則，Y2 輸出為 0。																	

FUN174 D >=	大於或等於比較指令 (比較 Sa 是否大於或等於 Sb)	FUN174 D >=
程式範例 2		
階梯圖		
ST IF R600 = R602 OR R604 > R606 Then IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 =< R622) OR (M200 AND M201) Then IF M100 Then Y10 := TRUE; END_IF END_IF ELSE Y10 := FALSE; END_IF		
範例說明：當 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≥DR618 或 DR612≠DR614 及 DR620≤DR622 或 M200=1 及 M201=1、而且 M100=1 時，Y10 輸出為 1；否則，Y10 輸出為 0。		

7-21-6 小於或等於比較指令 (比較 Sa 是否小於或等於 Sb)

FUN175 D =<		小於或等於比較指令 (比較 Sa 是否小於或等於 Sb)												FUN175 D =<			
指令說明																	
		執行控制 EN 175D. =< Sa Sb										Sa：比較值 a 或其暫存器號碼 Sb：比較值 b 或其暫存器號碼 Sa、Sb 可結合 V、Z、P0~P9 作間接定址應用					
運算元 範圍	WX	WY	WM	WSM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WSM9120 WM25584	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999	16 或 32 位元 正、負數	V、Z P0~P9		
	Sa																
	Sb																
功能敘述																	
		● 當執行控制“EN”=1 時，本指令以正負數(Signed)運算法則執行 Sa 與 Sb 的比較。若 Sa≤Sb，則解題輸出為 1；若否，則解題輸出為 0。															
程式範例 1																	
		階段圖 M10 171. =< R20 R22 M11 Y2 () ST IF M10 AND (R20 >= R22 OR M11) Then Y2 := TRUE; ELSE Y2 := FALSE; END_IF															
		範例說明：當 M10=1、R20≤R22 或 M11=1 時，Y2 輸出為 1；否則，Y2 輸出為 0。															

FUN175 D =<	小於或等於比較指令 (比較 Sa 是否小於或等於 Sb)	FUN175 D =<
程式範例 2		
階梯圖 ST <pre> IF R600 = R602 OR R604 > R606 Then IF (R608 < R610 AND R616 >= R618) OR (R612 <> R614 AND R620 <= R622) OR (M200 AND M201) Then IF M100 Then Y10 := TRUE; END_IF END_IF ELSE Y10 := FALSE; END_IF </pre> 範例說明：當 DR600=DR602 或 DR604>DR606、且 DR608<DR610 及 DR616≥DR618 或 DR612≠DR614 及 DR620≤DR622 或 M200=1 及 M201=1、而且 M100=1 時，Y10 輸出為 1；否則，Y10 輸出為 0。		

7-22 運動控制指令

7-22-8 執行運動流程 (MFFlowStart)

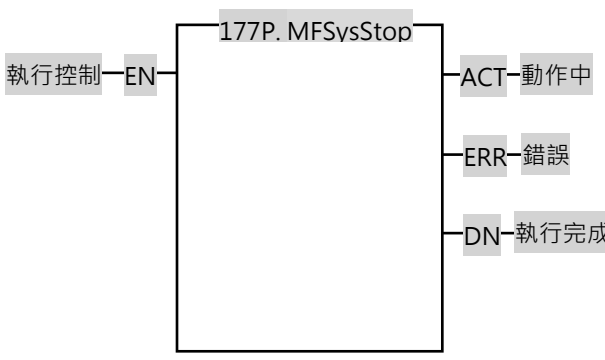
Fun176 P MFFlowStart		【運動流程開始】										Fun176 P MFFlowStart			
指令說明															
階段圖符號 執行控制—EN 176P. MFFlowStart ID : —ACT—動作中 —ERR—錯誤 —DN—執行完成 運算元 ID : 運動流程 ID															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
	ID	○	○	○	○	○	○	○	○	○	○	○	○	○	1~16
功能描述															
啟動指定 ID 的運動流程															
● EN = 1 : 啟動運動流程 ● ACT = 1 : 運動流程動作中 ● ERR = 1 : 運動流程錯誤 ● DN = 1 : 運動流程動作完畢 ● ID : 要啟動的運動流程 UID															

程式範例

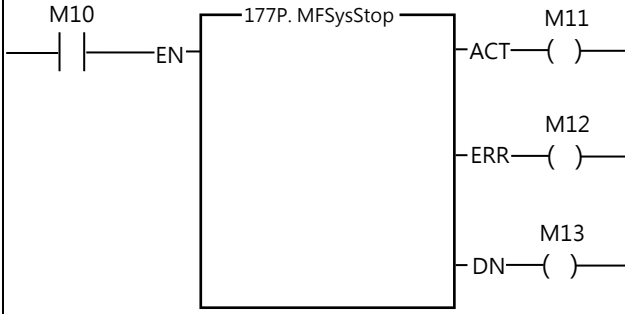
階段圖	ST
	<pre> MFFlowStart (EN:= M70, ID:= 1, ACT=> M71, ERR=> M72, DN=> M73); IF M73 Then M70 := FALSE; END_IF </pre>

- 當執行控制 “EN” = 1 時，將執行對應 UID 的運動流程。
- 若 ID 沒有對應到運動流程，將會觸發 ERR = 1。

7-22-9 運動系統緊急停止 MFSysStop)

Fun177 P MFSysStop	【運動系統緊急停止】	Fun177 P MFSysStop
指令說明	階段圖符號 無運算元 	
功能描述	● 中斷所有運動流程，並停止 EtherCAT 通訊。若要再啟動流程，則需再 MFSysInit 啟動 EtherCAT 通訊。 ● EN = 1：中斷所有運動流程 ● EN = 1：運動控制系統急緊急停止 ● ACT = 1：系統急停動作中 ● ERR = 1：系統急停錯誤 ● DN = 1：系統急停完畢	

程式範例

階梯圖	ST
	<pre>MFSysStop (EN:= M10, ACT=> M11, ERR=> M12, DN=> M13);</pre>

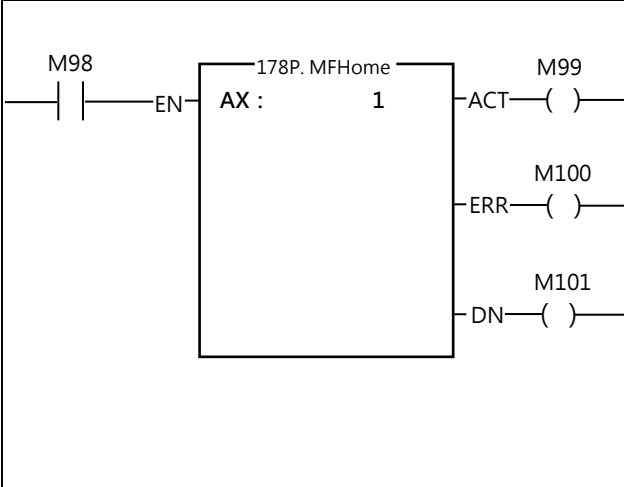
- 當執行控制 “EN” = 1 時，將執行中的運動控制緊急停止動作。
- 執行後若要重新開始操作，需執行初始化及啟動。

7-22-10 原點復歸(MFHome)

Fun178 P MFHome	【原點復歸】	Fun178 P MFHome												
指令說明														
階段圖符號 執行控制 EN 178P.MFHome AX : ACT動作中 ERR錯誤 DN執行完成 運算元 AX : 執行原點復歸的軸														
繼電器和暫存器														
類型	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
範圍	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
AX							○	○	○	○	○	○	1~16	○

| 功能描述 | | |
| 指定運動軸執行原點復歸。 - EN = 1：觸發原點復歸 - ACT = 1：原點復歸動作中 - ERR = 1：原點復歸動作錯誤 - DN = 1：原點復歸動作完畢 - AX：要執行的軸 特殊暫存器 Axis 1：原點復歸動作中 M10621 Axis 1：原點復歸動作完畢 M10622 - 關於 HOME 指令各模式與細節請參閱第十章。 | | |

程式範例

階梯圖	ST
	<pre>MFHome (EN:= M98, AX:= 1, ACT=> M99, ERR=> M100, DN=> M101);</pre>

- 當執行控制 “EN” = 1 時，將依運動軸設定頁面的參數進行原點復歸。

7-22-11 位置控制(MFPointMov)

Fun179 P MFPointMov		【位置控制】												Fun179 P MFPointMov	
指令說明															
階段圖符號 執行控制 —EN— 179P. MFPointMov ID : AX : —ACT—動作中 —ERR—錯誤 —DN—執行完成								運算元 PT : 執行位置控制點表的第幾點 AX : 執行位置控制的主軸							
繼電器和暫存器															
類型	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	
範圍	WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z P0 ~ P9	
	WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999			
ID	○	○	○	○	○	○	○	○	○	○	○	○	1~256	○	
AX	○	○	○	○	○	○	○	○	○	○	○	○	1~16	○	
功能描述															
執行點表位置控制指令。															
● EN = 1 : 觸發位置控制 ● ACT = 1 : 位置控制動作中 ● ERR = 1 : 位置控制錯誤 ● DN = 1 : 位置控制動作完畢 ● PT : 選擇運動點參數的點位 ● AX : 要執行的軸 ● 特殊暫存器： ● Axis 1 : 位置控制動作中 M10623 ● Axis 1 : 位置控制動作完畢 M10624															

程式範例

階梯圖	ST
	<pre>MFPointMov (EN:= M85, PT:= 3, AX:= 1, ACT=> M86, ERR=> M87, DN=> M88);</pre>

- 當執行控制 “EN” = 1 時，將以 AX 指定的軸執行第 PT 指定編號的點表，
- 當執行控制 “EN” = 0 時，運動會立即停止。
- 以下表當範例，當 PT = 1、AX = 1 時則會使軸 1 依照點表 1 的參數運行；但如果設定 PT=2、AX=1 則會因與點表設定不同造成失敗，使 ERR 被觸發。

	Axis
1	M : Axis_1
2	M : Axis_2
3	M : Axis_1

7-22-12 吋動控制(MFJog)

Fun180 P MFJog		【JOG】										Fun180 P MFJog		
指令說明														
階段圖符號 執行控制 EN 180P.MFJog AX : MD: ACT動作中 ERR錯誤 DN執行完成 運算元 AX : 執行時動控制的軸 MD : 執行的模式														
繼電器和暫存器														
類型	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
範圍	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
AX	○	○	○	○	○	○	○	○	○	○	○	○	1~16	○
MD	○	○	○	○	○	○	○	○	○	○	○	○	0~3	○
功能描述														
依照 JOG 參數及模式設定，指定運動軸執行時動功能。														
● EN = 1：觸發時動控制 ● D/R = 1 正轉 / = 0 反轉 ● ACT = 1：JOG 動作中 ● ERR = 1：JOG 錯誤 ● DN = 1：JOG 動作完畢 ● AX：要執行的軸 ● MD：模式 0~模式 3														
模式 0：以 JOG 啟動速度持續前進。														
模式 1：以 JOG 啟動速度前進，前進時動距離後停止。														
模式 2：以 JOG 啟動速度起步，以 JOG 加速度加速到 JOG 速度後持續前進。														
模式 3：以 JOG 啟動速度起步，以 JOG 加速度加速到 JOG 速度，前進時動距離後停止。														
● 特殊暫存器 ● Axis 1：JOG 動作中 M10625 ● Axis 1：JOG 動作完畢 M10626 ● 關於 JOG 指令各模式與細節請參閱第十一章。														

程式範例

階梯圖	ST
	<pre>MFJog (EN:= M93, D_R:= M94, AX:= 1, MD:= 1, ACT=> M95, ERR=> M96, DN=> M97);</pre>

- 當執行控制 “EN” = 1 時，將以 AX 指定的軸執行 MD 指定的模式。
- 當執行控制 “EN” = 0 時，運動會立即停止。
- 以下表為例，當 AX = 1、MD = 1 時，代表軸 1 將以 1mm/s 的速度跑 100mm 的距離，

	Axis 1
JOG 啟動速度	1mm/s
JOG 速度	10mm/s
JOG 加速度	1000mm/s ²
JOG 減速度	1000mm/s ²
寸動距離	100mm

7-22-13 改變運動控制參數(MFChgTbPrm)

Fun181 P MFChgTbPrm	【改變運動控制參數】	Fun181 P MFChgTbPrm																																																												
指令說明																																																														
階段圖符號 執行控制—EN 181P.MFChgTbPrm TM: PN: S: ERR—錯誤 DN—執行完成 運算元 TM: 表格編號 PN: 點編號 S: 項目編號 PV: 寫入值 繼電器和暫存器 <table><tr><th>範圍</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>運算元</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V, Z P0 ~ P9</td></tr><tr><td>TM</td><td></td><td></td><td></td><td></td><td></td><td></td><td>0~128</td><td></td></tr><tr><td>PN</td><td></td><td></td><td></td><td></td><td></td><td></td><td>1~4096</td><td></td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~50</td><td>○</td></tr><tr><td>PV</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>-2147483648 ~21474826477</td><td>○</td></tr></table> <tr><td colspan="3">功能描述</td></tr> <tr><td colspan="3"> ●【Fun181 改變運動控制參數】用來改變單個或少數運動控制參數，如果需要大量讀取或寫入運動控制參數，可以使用【Fun188 配方讀取】和【Fun189 配方寫入】。 ●運算元 TM 表格編號: 0 點表、1 軸表、2 同步表、128 流程表 PN 點編號: 依據 TM 要修改的表對應到不同類型的編號，點表編號、軸編號、流程塊編號 S 項目編號: 請參考後面表格 PV 寫入值: 要寫入的數值，固定 Double Word ●當執行控制【EN】由上微分觸發，Fun181 將 PV 值寫入指定的運動控制參數 當執行控制【EN】由下微分觸發，復位所有輸出指示 ●當寫入運動控制參數中，如有錯誤發生，則輸出指示【ERR】ON。 ●當寫入運動控制參數完成，輸出指示【DN】ON。 </td></tr>			範圍	HR	IR	OR	SR	ROR	DR	K	XR	運算元	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9	TM							0~128		PN							1~4096		S	○	○	○	○	○	○	0~50	○	PV	○	○	○	○	○	○	-2147483648 ~21474826477	○	功能描述			●【Fun181 改變運動控制參數】用來改變單個或少數運動控制參數，如果需要大量讀取或寫入運動控制參數，可以使用【Fun188 配方讀取】和【Fun189 配方寫入】。 ●運算元 TM 表格編號: 0 點表、1 軸表、2 同步表、128 流程表 PN 點編號: 依據 TM 要修改的表對應到不同類型的編號，點表編號、軸編號、流程塊編號 S 項目編號: 請參考後面表格 PV 寫入值: 要寫入的數值，固定 Double Word ●當執行控制【EN】由上微分觸發，Fun181 將 PV 值寫入指定的運動控制參數 當執行控制【EN】由下微分觸發，復位所有輸出指示 ●當寫入運動控制參數中，如有錯誤發生，則輸出指示【ERR】ON。 ●當寫入運動控制參數完成，輸出指示【DN】ON。		
範圍	HR	IR	OR	SR	ROR	DR	K	XR																																																						
運算元	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9																																																						
TM							0~128																																																							
PN							1~4096																																																							
S	○	○	○	○	○	○	0~50	○																																																						
PV	○	○	○	○	○	○	-2147483648 ~21474826477	○																																																						
功能描述																																																														
●【Fun181 改變運動控制參數】用來改變單個或少數運動控制參數，如果需要大量讀取或寫入運動控制參數，可以使用【Fun188 配方讀取】和【Fun189 配方寫入】。 ●運算元 TM 表格編號: 0 點表、1 軸表、2 同步表、128 流程表 PN 點編號: 依據 TM 要修改的表對應到不同類型的編號，點表編號、軸編號、流程塊編號 S 項目編號: 請參考後面表格 PV 寫入值: 要寫入的數值，固定 Double Word ●當執行控制【EN】由上微分觸發，Fun181 將 PV 值寫入指定的運動控制參數 當執行控制【EN】由下微分觸發，復位所有輸出指示 ●當寫入運動控制參數中，如有錯誤發生，則輸出指示【ERR】ON。 ●當寫入運動控制參數完成，輸出指示【DN】ON。																																																														

程式範例

階段圖	ST
<pre> graph LR M1000 --> EN subgraph 181DP_MFChgTbPrm [181DP. MFChgTbPrm] TM[0] PN[1] S[2] PV[1000000] end ERR --> M1001 DN --> M1002 </pre>	<pre>MFChgTbPrm (EN:= M1000, TM:= 0, PN:= 1, S:= 2, PV:= 1000000, ERR=> M1001, DN=> M1002);</pre>

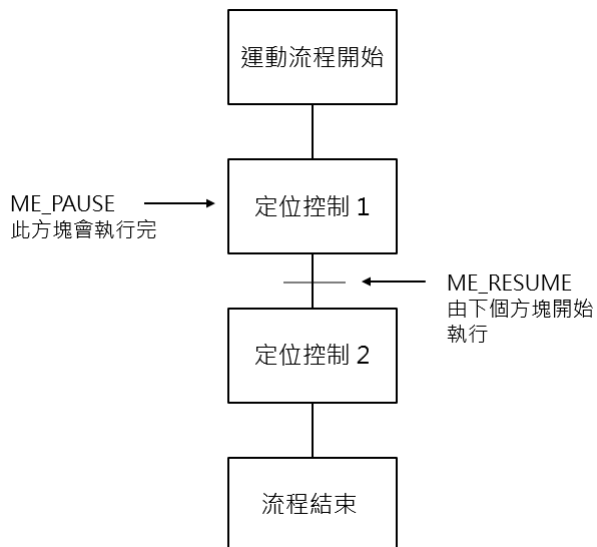
- M1000 OFF→ON 時，變更點表參數(TM：0 點表、PN：1 點表 1、S：2 主軸座標、PV：變更為 1000.000mm)，點表 1 的主軸移動距離變更為 1000.000mm。

7-22-14 暫停運動流程(MFFlowPause)

Fun182 P MFFlowPause	【暫停運動流程】												Fun182 P MFFlowPause	
指令說明														
階段圖符號 執行控制 EN 182P. MFFlowPause ID : ACT動作中 ERR錯誤 DN執行完成 運算元 ID : 運動流程 ID														
繼電器和暫存器														
類型	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
範圍	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
ID	○	○	○	○	○	○	○	○	○	○	○	○	- 32767~32767	○
功能描述														
暫停指定 ID 之運動流程，且執行完當前流程方塊才停止， 若要恢復暫停的運動流程，可以使用 Fun183 MFFlowResume 來恢復執行。 ● EN = 1：執行完目前流程方塊後停止進入下一步 ● ACT = 1：暫停動作中 ● ERR = 1：暫停錯誤 ● DN = 1：暫停完畢 ● ID：要暫停的運動流程 UID														

程式範例

階梯圖	ST
	<pre> MFFlowPause (EN:= M73, ID:= 1, ACT=> M74, ERR=> M75, DN=> M76); IF M76 Then M73 := FALSE; END_IF </pre>



- 當執行控制 “EN” = 1 時，將會在執行完當前的運動流程方塊後，暫停不去執行下一步。

7-22-15 恢復運動流程(MFFlowResume)

Fun183 P MFFlowResume		【恢復運動流程】										Fun183 P MFFlowResume																																												
指令說明																																																								
階段圖符號 執行控制 EN 183P. MFFlowResume ID : ACT-動作中 ERR-錯誤 DN-執行完成 運算元 ID : 運動流程 ID 繼電器和暫存器 <table><tr><td rowspan="3">範圍 運算元</td><td>WX</td><td>WY</td><td>WM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V, Z P0 ~ P9</td></tr><tr><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>- 32767~32767</td><td>○</td></tr></table>														範圍 運算元	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9	○	○	○	○	○	○	○	○	○	○	○	○	- 32767~32767	○
範圍 運算元	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K		XR																																									
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999			V, Z P0 ~ P9																																									
	○	○	○	○	○	○	○	○	○	○	○	○	- 32767~32767	○																																										
功能描述																																																								
恢復暫停或是中斷的運動流程，繼續執行。 ● EN = 1：恢復運動流程 ● ACT = 1：恢復運動流程動作中 ● ERR = 1：恢復運動流程錯誤 ● DN = 1：恢復運動流程完畢																																																								
程式範例																																																								
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td> M77 EN 183P. MFFlowResume ID : 1 ACT () M75 ERR () M76 DN (R) M77 </td><td><pre>MFFlowResume (EN:= M77, ID:= 1, ACT=> M75, ERR=> M76, DN=> M78); IF M78 Then M77 := FALSE; END_IF</pre></td></tr></table> ● 當執行控制“EN” = 1 時，會將因執行 Fun182(MFFlowPause)或 Fun184(MFFlowHalt)暫停的運動流程恢復運行。														階段圖	ST	M77 EN 183P. MFFlowResume ID : 1 ACT () M75 ERR () M76 DN (R) M77	<pre>MFFlowResume (EN:= M77, ID:= 1, ACT=> M75, ERR=> M76, DN=> M78); IF M78 Then M77 := FALSE; END_IF</pre>																																							
階段圖	ST																																																							
M77 EN 183P. MFFlowResume ID : 1 ACT () M75 ERR () M76 DN (R) M77	<pre>MFFlowResume (EN:= M77, ID:= 1, ACT=> M75, ERR=> M76, DN=> M78); IF M78 Then M77 := FALSE; END_IF</pre>																																																							

7-22-16 終止運動流程(MFFlowHalt)

Fun184 P MFFlowHalt	【中斷運動流程】												Fun184 P MFFlowHalt		
指令說明															
階段圖符號 執行控制 EN 184P. MFFlowHalt ID : ACT動作中 ERR錯誤 DN執行完成 運算元 ID : 運動流程 ID															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
	ID	○	○	○	○	○	○	○	○	○	○	○	○	- 32767~32767	○
功能描述															
立即停止當前執行的流程方塊。 若要繼續停止的運動流程，可以使用 Fun183 MFFlowResume 來恢復執行。 ● EN = 1：停止運動流程 ● ACT = 1：停止動作中 ● ERR = 1：停止錯誤 ● DN = 1：停止完畢 ● ID：要中斷的運動流程 UID															

程式範例

階梯圖	ST
	<pre> MFFlowHalt (EN:= M77, ID:= 1, ACT=> M78, ERR=> M79, DN=> M80); IF M80 Then M77 := FALSE; END_IF </pre>

- 當執行控制 “EN” = 1 時，將會立刻暫停執行中的運動流程方塊。

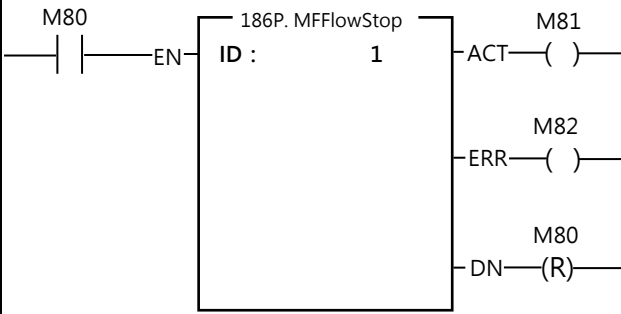
7-22-17 重置運動錯誤警報(MFSysRstAlm)

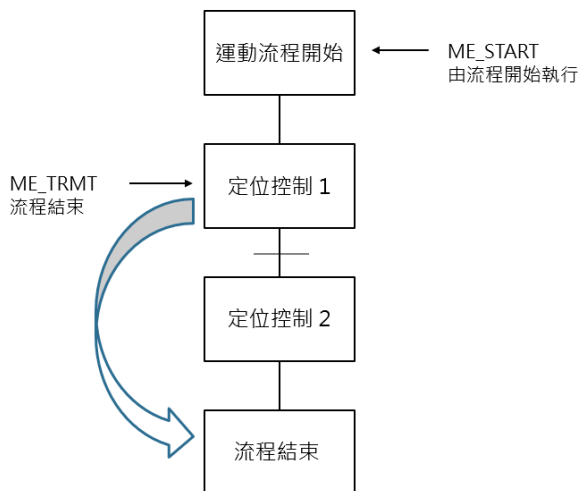
Fun185 P MFSysRstAlm	【重置運動錯誤警報】	Fun185 P MFSysRstAlm
指令說明		
階段圖符號 運算元		
功能描述		
清除所有運動流程及驅動器錯誤警報， 但驅動器通訊警報無法透過此命令清除，需重新上電。 ● EN = 1：上緣觸發清除運動錯誤警報 ● ACT = 1：清除運動錯誤警報動作中 ● ERR = 1：清除運動錯誤警報錯誤 ● DN = 1：清除運動錯誤警報完畢		
程式範例		
階段圖		ST <pre>MFSysRstAlm (EN:= M101, ACT=> M102, ERR=> M103, DN=> M104);</pre>
● 當執行控制“EN” = 1 時，會清除運動流程及驅動器發生的錯誤。		

7-22-18 運動流程結束(MFFlowStop)

Fun186 P MFFlowStop		【運動流程結束】										Fun186 P MFFlowStop																																															
指令說明																																																											
		階段圖符號 執行控制EN 186P. MFFlowStop ID : ACT動作中 ERR錯誤 DN執行完成										運算元 ID : 運動流程 ID																																															
		繼電器和暫存器 <table><tr><th> 範圍 </th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td rowspan="2">運算元</td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V, Z P0 ~ P9</td></tr><tr><td>ID</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>- 32767~32767</td><td>○</td></tr></table>														範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	運算元	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9	ID	○	○	○	○	○	○	○	○	○	○	○	- 32767~32767	○
範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																													
運算元	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9																																													
	ID	○	○	○	○	○	○	○	○	○	○	○	- 32767~32767	○																																													
功能描述																																																											
		立即結束所指定 ID 的運動流程。 當執行完此指令時，無法使用 MFFlowResume 來恢復執行。需使用 MFFlowStart 來重新啟動流程。 ● EN = 1：上緣觸發運動流程結束 ● ACT = 1：運動流程結束動作中 ● ERR = 1：運動流程結束錯誤 ● DN = 1：運動流程結束完畢																																																									

程式範例

階段圖	ST
	<pre> MFFlowStop (EN:= M80, ID:= 1, ACT=> M81, ERR=> M82, DN=> M83); IF M83 Then M80 := FALSE; END_IF </pre>

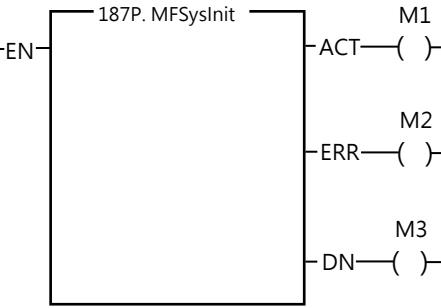


- 當執行控制 “EN” = 1 時，會將指定 ID 的運動流程立即結束。

7-22-19 Servo 初始化(MFSysInit)

Fun187 P MFSysInit	【系統初始化】													Fun187 P MFSysInit
指令說明														
階段圖符號 無運算元 執行控制 EN 187P. MFSysInit ACT動作中 ERR錯誤 DN執行完成														
繼電器和暫存器														
運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
	ID	○	○	○	○	○	○	○	○	○	○	○	○	1~16
功能描述														
● 欲透過 EtherCAT 通訊控制伺服，在執行任何運動控制前必需先執行此指令 ● 欲使用 Fun235 實體軸轉虛軸指令，必需在此指令前執行 ● EN = 1：啟動運動控制初始化 (觸發條件支援上下微分輸入) ● ACT = 1：運動控制初始化動作中 ● ERR = 1：運動控制初始化錯誤 ● DN = 1：運動控制初始化完畢														

程式範例

階梯圖	ST
	<pre>MFSysInit (EN:= TRUE, ACT=> M1, ERR=> M2, DN=> M3);</pre>

- 當執行控制 “EN” = 1 時，將執行運動控制功能初始化動作。
- 若執行發現無反應時請確認運動連結設定是否與實際連結相符。
- 初始化後，需將伺服啟動方可繼續後續操作，如所有軸使能 (Servo on) 暫存器 (M10520)。

7-22-20 運動控制配方表讀取(MFSysRCPR)

Fun188 P MFSysRCPR		【運動控制配方讀取】												Fun188 P MFSysRCPR	
指令說明															
階段圖符號 執行控制 EN 188P. MFSysRCPR ID : D : Gp: ACT-動作中 ERR-錯誤 DN-執行完成 運算元 Md : 模式 D : 配方起始暫存器 Gp : 讀取配方表第幾列															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z P0 ~ P9
		WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		
ID		○	○	○	○	○	○	○	○	○	○	○	○	0~1	
D		○	○	○	○	○	○	○	○	○	○	○	○		○
Gp		○	○	○	○	○	○	○	○	○	○	○	○	0~100	

功能描述	
● 【Fun188 配方讀取】和【Fun189 配方寫入】用來大量讀取或寫入運動控制參數，如果只需要修改單一或少數參數，可以使用【Fun181 改變運動控制參數】或【Fun198 映射表】。 ● 軸動作停止時才能讀取參數。 ● 運算元 Md 模式：0 使用 PLC 暫存器 D 配方起始暫存器：讀取配方表後要存放的暫存器起始位址 Gp 讀取配方表第幾列：讀取配方表第幾列，0 全部讀取 ● 當執行控制【EN】由上微分觸發，Fun188 將指定配方讀取至指定的暫存器 ● 當執行控制【EN】由下微分觸發，復位所有輸出指示 ● 當讀取配方中，則輸出指示【ACT】ON。 ● 當讀取配方中，如有錯誤發生，則輸出指示【ERR】ON。 ● 當讀取配方完成，輸出指示【DN】ON。	

配方表

【專案管理】>【運動控制】>【運動配方】

運動配方表 ×					
	表	索引	長度	起始位址	結束位址
1	點表	1	1	R0	R49
2	軸表	1	1	R50	R119
3	同步表	1	1	R120	R269

- 運動配方表
表：點表、軸表、同步表
索引：點表(第幾點)、軸表(第幾軸)、同步表(第幾軸)
長度：連續點表或連續軸
起始位址：讀寫配方的暫存器起始位址
- 運動配方表暫存器數值定義請參考後面表格

配方點表

序號	項目	資料大小	資料類型	長度	定義
R+0	運行模式	WORD	INT	1	0. 未使用 1. 單軸/絕對 2. 單軸/相對 3. 直線(2軸)/絕對 4. 直線(2軸)/相對 5. 直線(3軸)/絕對 6. 直線(3軸)/相對 7. 直線(4軸)/絕對 8. 直線(4軸)/相對 9. 圓弧/絕對 10. 圓弧/相對 11. 螺旋/絕對 12. 螺旋/相對 13. 單軸速度
R+1	加速類型	WORD	INT	1	0. T 曲線 1. S 曲線
R+2	主軸	WORD	INT	1	1~16 不使用 = 0
R+3	補間軸 1	WORD	INT	1	1~16 不使用 = 0
R+4	補間軸 2	WORD	INT	1	1~16 不使用 = 0
R+5	補間軸 3	WORD	INT	1	1~16 不使用 = 0
R+6	目標位置 主軸	DWORD	INT	2	精度：小數點位置(可負數)
R+8	目標位置 補間軸 1	DWORD	INT	2	精度：小數點位置(可負數)
R+10	目標位置 補間軸 2	DWORD	INT	2	精度：小數點位置(可負數)
R+12	目標位置 補間軸 3	DWORD	INT	2	精度：小數點位置(可負數)
R+14	速度	DWORD	INT	2	精度：小數點位置(只能正數)
R+16	加速度	DWORD	INT	2	精度：小數點位置(只能正數)
R+18	減速度	DWORD	INT	2	精度：小數點位置(只能正數)
R+20	S 加速度曲線	WORD	INT	1	精度：0.1
R+21	S 減速度曲線	WORD	INT	1	精度：0.1
R+22	圓弧模式	WORD	INT	1	0. 通過點 1. 中心 2. 半徑
R+23	圓弧方向	WORD	INT	1	0. 逆時針 1. 順時針
R+24	圓弧(通過點/圓心) X 座標	DWORD	INT	2	精度：小數點位置(可負數)
R+26	圓弧(通過點/圓心) Y 座標	DWORD	INT	2	精度：小數點位置(可負數)
R+28	圓弧半徑	DWORD	INT	2	精度：小數點位置(只能正數)
R+30	輔助半徑	DWORD	INT	2	精度：小數點位置(只能正數)
R+32	待機時間	DWORD	UINT	2	單位 ms
R+34	連續點	WORD	INT	1	1~1024 結束 = 0
R+35	圓弧圈數	WORD	UINT	1	0~65535
R+36	連續模式	WORD	INT	1	0. 待機 1. 下一點速度連續 2. 當前點速度連續 3. 開始速度連續
R+42	圓弧(通過點/圓心) Z 座標	DWORD	INT	2	精度：小數點位置(可負數)

配方軸表

序號	項目	資料大小	資料類型	長度	定義
R+0	編碼器類型	WORD		1	0. 增量型 1. 絕對型
R+1	單位	WORD		1	0. PLS 1. mm 2. deg 3. inch
R+2	小數點位置	WORD		1	1000. 1 100. 0.1 10. 0.01 1. 0.001
R+3	每圈脈波數	DWORD		2	精度：小數點位置
R+5	每圈單位長度	DWORD		2	精度：小數點位置
R+7	速度單位	WORD			0. PLS/Sec 1. PLS/min 2. RPM
R+8	速度增益	DWORD		2	精度：0.001
R+10	開始速度	DWORD		2	精度：小數點位置
R+12	最大馬達速度	DWORD		2	精度：1 單位 RPM
R+14	預設加速度	DWORD		2	精度：小數點位置
R+16	預設減速度	DWORD		2	精度：小數點位置
R+18	軟限制 +	DWORD		2	精度：小數點位置 可負數
R+20	軟限制 -	DWORD		2	精度：小數點位置 可負數
R+22	跟蹤誤差容許範圍	DWORD		2	精度：小數點位置
R+24	跟蹤誤差容許時間	DWORD		2	單位ms
R+26	定位完成容許誤差	DWORD		2	精度：小數點位置
R+28	定位完成容許時間	DWORD		2	單位ms
R+30	最大馬達扭矩	WORD		1	精度：0.1
R+31	最大扭矩限制 +	WORD		1	精度：0.1
R+32	最大扭矩限制 -	WORD		1	精度：0.1
R+41	停止模式	WORD		1	5. 減速停止 7. 立即停止
R+42	停止減速度	DWORD		2	精度：小數點位置
R+44	復歸模式	WORD		1	99. 當前位置為原點 100. Dog Forward 101. 近點復歸 102. Dog-z-sig Forward 103. Dog-z-sig Backward
R+45	復歸 IO 來源	WORD		1	0. 從伺服驅動器 1. 從PLC
R+46	復歸開始方向	WORD		1	0. 負方向 1. 正方向
R+47	原點復歸偏移	DWORD		2	精度：小數點位置 (可負數)
R+49	復歸搜尋速度	DWORD		2	精度：小數點位置
R+51	復歸爬行速度	DWORD		2	精度：小數點位置
R+53	復歸減速度	DWORD		2	精度：小數點位置
R+55	極限開關 - 位元	WORD		1	
R+56	極限開關 + 位元	WORD		1	
R+57	原點開關位元	WORD		1	
R+58	原點零點訊號數	DWORD		2	
R+60	JOG 啟動速度	DWORD		2	精度：小數點位置
R+62	JOG 速度	DWORD		2	精度：小數點位置
R+64	JOG 加速度	DWORD		2	精度：小數點位置
R+66	JOG 減速度	DWORD		2	精度：小數點位置
R+68	寸動距離	DWORD		2	精度：小數點位置

配方同步表

序號	項目	資料大小	資料類型	長度	定義
R+0	輸入軸座標單位	WORD		1	
R+1	輸入軸小數點位置	WORD		1	
R+2	輸入軸週期	DWORD		2	精度：輸入軸小數點位置
R+4	減速停止滑動時間	DWORD		2	
R+6	輸入軸相位初始化方法	WORD		1	
R+7	同步主軸相位預設值	DWORD		2	精度：輸入軸小數點位置
R+9	相位補償後主軸相位預設值	DWORD		2	精度：輸入軸小數點位置
R+11	主離合器輸入項預設值	DWORD		2	精度：輸入軸小數點位置
R+13	輔助離合器輸入相位預設值	DWORD		2	精度：輸入軸小數點位置
R+15	凸輪輸入軸相位初始化方法	WORD		1	
R+16	主離合器輸出相位預設值	DWORD		2	精度：輸入軸小數點位置
R+18	輔助離合器輸出相位預設值	DWORD		2	精度：輸入軸小數點位置
R+20	保留	DWORD		2	
R+22	凸輪輸入相位預設值	DWORD		2	精度：輸入軸小數點位置
R+24	凸輪輸出基準座標	DWORD		2	精度：輸入軸小數點位置
R+26	主軸 1 選擇輸入軸	WORD		1	
R+27	主軸 1 外部參照編號	WORD		1	
R+28	主軸 1 防止逆轉	WORD		1	
R+29	主軸 1 座標轉換設定	WORD		1	
R+30	主軸 1 座標轉換分子	DWORD		2	
R+32	主軸 1 座標轉換分母	DWORD		2	
R+34	主軸 2 選擇輸入軸	WORD		1	
R+35	主軸 2 外部參照編號	WORD		1	
R+36	主軸 2 防止逆轉	WORD		1	
R+37	主軸 2 座標轉換設定	WORD		1	
R+38	主軸 2 座標轉換分子	DWORD		2	
R+40	主軸 2 座標轉換分母	DWORD		2	
R+42	輔助軸選擇輸入軸	WORD		1	
R+43	輔助軸外部參照編號	WORD		1	
R+44	輔助軸防止逆轉	WORD		1	
R+45	輔助軸座標轉換設定	WORD		1	
R+46	輔助軸座標轉換分子	DWORD		2	
R+48	輔助軸座標轉換分母	DWORD		2	
R+50	主軸相位補償指令量	DWORD		2	精度：輸入軸小數點位置
R+52	主軸相位補償更改模式	WORD		1	
R+53	主軸相位補償更改時間	DWORD		2	
R+55	輔助軸相位補償指令量	DWORD		2	精度：輸入軸小數點位置
R+57	輔助軸相位補償更改模式	WORD		1	
R+58	輔助軸相位補償更改時間	DWORD		2	
R+60	可變齒輪比分子	DWORD		2	
R+62	可變齒輪比分母	DWORD		2	
R+64	可變齒輪比更改模式	WORD		1	
R+65	可變齒輪比更改時間	DWORD		2	
R+67	主離合器 ON 條件	WORD		1	
R+68	主離合器 ON 設定值	DWORD		2	精度：輸入軸小數點位置
R+70	主離合器 ON 延遲	DWORD		2	精度：輸入軸小數點位置
R+73	主離合器 ON 連接方式	WORD		1	
R+75	主離合器 ON 滑動曲線	WORD		1	
R+78	主離合器 ON 滑動時間	DWORD		2	
R+80	主離合器 ON 隨動時間	DWORD		2	
R+82	主離合器 ON 隨動量	DWORD		2	精度：輸入軸小數點位置
R+84	主離合器 OFF 條件	WORD		1	
R+85	主離合器 OFF 設定值	DWORD		2	精度：輸入軸小數點位置
R+87	主離合器 OFF 延遲	DWORD		2	精度：輸入軸小數點位置
R+90	主離合器 OFF 連接方式	WORD		1	
R+92	主離合器 OFF 滑動曲線	WORD		1	
R+95	主離合器 OFF 滑動時間	DWORD		2	
R+97	輔助離合器 ON 條件	WORD		1	
R+98	輔助離合器 ON 設定值	DWORD		2	精度：輸入軸小數點位置
R+100	輔助離合器 ON 延遲	DWORD		2	精度：輸入軸小數點位置
R+103	輔助離合器 ON 連接方式	WORD		1	
R+105	輔助離合器 ON 滑動曲線	WORD		1	
R+108	輔助離合器 ON 滑動時間	DWORD		2	
R+110	輔助離合器 ON 隨動時間	DWORD		2	
R+112	輔助離合器 ON 隨動量	DWORD		2	精度：輸入軸小數點位置
R+114	輔助離合器 OFF 條件	WORD		1	
R+115	輔助離合器 OFF 設定值	DWORD		2	精度：輸入軸小數點位置
R+117	輔助離合器 OFF 延遲	DWORD		2	精度：輸入軸小數點位置
R+120	輔助離合器 OFF 連接方式	WORD		1	
R+122	輔助離合器 OFF 滑動曲線	WORD		1	
R+125	輔助離合器 OFF 滑動時間	DWORD		2	
R+132	步進角補償基準速度	DWORD		2	精度：輸入軸小數點位置
R+134	步進角補償基準量	DWORD		2	精度：輸入軸小數點位置
R+136	步進角補償更改方式	WORD		1	
R+137	步進角補償更改時間	DWORD		2	
R+139	凸輪資料編號	WORD		1	
R+140	凸輪行程	DWORD		2	精度：軸表小數點位置
R+142	同步接點編號	WORD		1	
R+143	輸出濾波器時間常數	DWORD		2	

程式範例

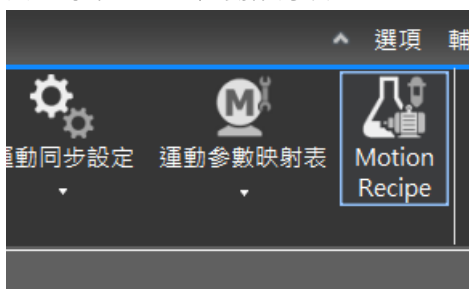
階梯圖	ST
	<pre> MFSysRCPR (EN:= M1000, Md:= 0, D:= R1000, Gp:= 0, ACT=> M1001, ERR=> M1002, DN=> M1003); </pre>

運動配方表

表	索引	長度	起始位址	結束位址
1 點表	1	1	R1000	R1049
2 軸表	1	1	R1050	R1119
3 同步表	1	1	R1120	R1269

- M1000 由 OFF→ON 時，讀取全部配方表存放至 R1000。
- 讀取 PLC 點表 1 的參數，存放至 R1000~R1049
- 讀取 PLC 軸表(軸 1)的參數，存放至 R1050~R1119
- 讀取 PLC 同步表(軸 1)的參數，存放至 R1120~R1269

於“專案” -> “運動配方表”



此以點表的目標位置做映射為例,下圖為設定頁面:

表	索引	長度	起始位址	結束位址
1 點表	1	1	R0	R49

FUN189 MFSysRCPW	配方表寫入	FUN189 MFSysRCPW
---------------------	-------	---------------------

範例程式為以點表為例,以下為程式:

階梯圖(副程式)	ST
	<pre> Timer(EN:= M50 = FALSE, T:= 0, PV:= 1000, IsTimeout=> T0); MFSysInit (EN:= T0, ACT=> M100, ERR=> M101, DN=> M102); Timer(EN:= T0, T:= 1, PV:= 1000, IsTimeout=> T1); IF T1 AND M50 = FALSE Then M10520 := TRUE; END_IF MFPointMov (EN:= M0, PT:= 1, AX:= 1, ACT=> M103, ERR=> M104, DN=> M105); MFSysStop (EN:= M50, ACT=> M106, ERR=> M107, DN=> M108); MFSysRCPR (EN:= M60, Md:= 1, D:= R0, Gp:= 1, ACT=> M109, ERR=> M110, DN=> M111); </pre>

此範例程式在一開始執行後一秒後會初始化運動控制系統,再 1 秒後會將其他軸 Servo On,此時將 M60 On 起,即會將中點表的全部內容移入配方表 1 中,此為 R0 為被移入點表中的起始的暫存器,再將 M0 On 起來,執行點表,其設定將會依照 DR6 的位置走;如需恢復配方表使用前,可將 M50 On 起後關閉以重啟 Motion 控制系統。

7-22-21 運動控制配方表寫入(MFSysRCPW)

Fun189 P MFSysRCPW		【運動控制配方寫入】										Fun189 P MFSysRCPW																																																																																																									
指令說明																																																																																																																					
		階段圖符號 執行控制 EN 189P. MFSysRCPW MD : D : Gp: ACT 動作中 ERR 錯誤 DN 執行完成										運算元 Md : 模式 D : 配方起始暫存器 Gp : 寫入配方表第幾列																																																																																																									
		繼電器和暫存器 <table><tr><td rowspan="4"> 運算元 範圍 </td><td>WX</td><td>WY</td><td>WM</td><td>WS</td><td>TMR</td><td>CTR</td><td>HR</td><td>IR</td><td>OR</td><td>SR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>WX0</td><td>WY0</td><td>WM0</td><td>WS0</td><td>T0</td><td>C0</td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td></td><td>V, Z</td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td></td><td>P0 ~</td></tr><tr><td>WX1008</td><td>WY1008</td><td>WM9104</td><td>WS3088</td><td>T1023</td><td>C1279</td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td></td><td>P9</td></tr><tr><td>ID</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>0~1</td><td></td></tr><tr><td>D</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td></td><td> </td></tr><tr><td>Gp</td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td>0~100</td><td></td></tr></table>														運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z														P0 ~	WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		P9	ID													0~1		D															Gp													0~100	
運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																																							
	WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z																																																																																																							
														P0 ~																																																																																																							
	WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		P9																																																																																																							
ID													0~1																																																																																																								
D																																																																																																																					
Gp													0~100																																																																																																								
功能描述																																																																																																																					

- 【Fun188 配方讀取】和【Fun189 配方寫入】用來大量讀取或寫入運動控制參數，如果只需要修改單一或少數參數，可以使用【Fun181 改變運動控制參數】或【Fun198 映射表】。
- 軸動作停止時才能寫入參數。
- 運算元
Md 模式 : 0 使用 PLC 暫存器
D 配方起始暫存器 : 要寫入配方表的暫存器起始位址
Gp 寫入配方表第幾列 : 寫入配方表第幾列，0 全部寫入
- 當執行控制【EN】由上微分觸發，Fun188 將指定暫存器寫入至指定的配方
- 當執行控制【EN】由下微分觸發，復位所有輸出指示
- 當寫入配方中，則輸出指示【ACT】ON。
- 當寫入配方中，如有錯誤發生，則輸出指示【ERR】ON。
- 當寫入配方完成，輸出指示【DN】ON。

配方表

【專案管理】>【運動控制】>【運動配方】

運動配方表 ×					
	表	索引	長度	起始位址	結束位址
1	點表	1	1	R0	R49
2	軸表	1	1	R50	R119
3	同步表	1	1	R120	R269

● 運動配方表

表：點表、軸表、同步表

索引：點表(第幾點)、軸表(第幾軸)、同步表(第幾軸)

長度：連續點表或連續軸

起始位址：配方的暫存器起始位址

● 運動配方表暫存器數值定義請參考後面表格

配方點表

序號	項目	資料大小	資料類型	長度	定義
R+0	運行模式	WORD	INT	1	0. 未使用 1. 單軸/絕對 2. 單軸/相對 3. 直線(2軸)/絕對 4. 直線(2軸)/相對 5. 直線(3軸)/絕對 6. 直線(3軸)/相對 7. 直線(4軸)/絕對 8. 直線(4軸)/相對 9. 圓弧/絕對 10. 圓弧/相對 11. 螺旋/絕對 12. 螺旋/相對 13. 單軸速度
R+1	加速類型	WORD	INT	1	0. T 曲線 1. S 曲線
R+2	主軸	WORD	INT	1	1~16 不使用 = 0
R+3	補間軸 1	WORD	INT	1	1~16 不使用 = 0
R+4	補間軸 2	WORD	INT	1	1~16 不使用 = 0
R+5	補間軸 3	WORD	INT	1	1~16 不使用 = 0
R+6	目標位置 主軸	DWORD	INT	2	精度：小數點位置(可負數)
R+8	目標位置 補間軸 1	DWORD	INT	2	精度：小數點位置(可負數)
R+10	目標位置 補間軸 2	DWORD	INT	2	精度：小數點位置(可負數)
R+12	目標位置 補間軸 3	DWORD	INT	2	精度：小數點位置(可負數)
R+14	速度	DWORD	INT	2	精度：小數點位置(只能正數)
R+16	加速度	DWORD	INT	2	精度：小數點位置(只能正數)
R+18	減速度	DWORD	INT	2	精度：小數點位置(只能正數)
R+20	S 加速度曲線	WORD	INT	1	精度：0.1
R+21	S 減速度曲線	WORD	INT	1	精度：0.1
R+22	圓弧模式	WORD	INT	1	0. 通過點 1. 中心 2. 半徑
R+23	圓弧方向	WORD	INT	1	0. 逆時針 1. 順時針
R+24	圓弧(通過點/圓心) X 座標	DWORD	INT	2	精度：小數點位置(可負數)
R+26	圓弧(通過點/圓心) Y 座標	DWORD	INT	2	精度：小數點位置(可負數)
R+28	圓弧半徑	DWORD	INT	2	精度：小數點位置(只能正數)
R+30	輔助半徑	DWORD	INT	2	精度：小數點位置(只能正數)
R+32	待機時間	DWORD	UINT	2	單位 ms
R+34	連續點	WORD	INT	1	1~1024 結束 = 0
R+35	圓弧圈數	WORD	UINT	1	0~65535
R+36	連續模式	WORD	INT	1	0. 待機 1. 下一點速度連續 2. 當前點速度連續 3. 開始速度連續
R+42	圓弧(通過點/圓心) Z 座標	DWORD	INT	2	精度：小數點位置(可負數)



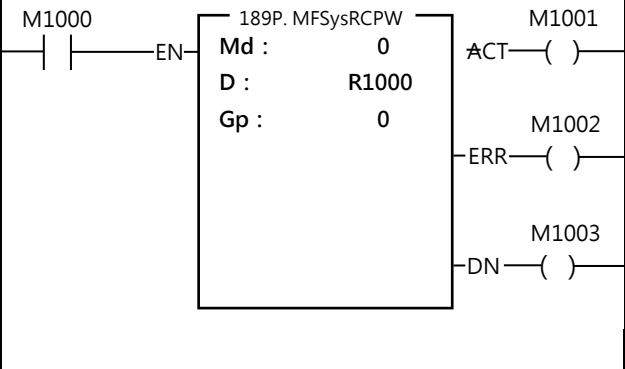
配方軸表

序號	項目	資料大小	資料類型	長度	定義
R+0	編碼器類型	WORD		1	0. 增量型 1. 絕對型
R+1	單位	WORD		1	0. PLS 1. mm 2. deg 3. inch
R+2	小數點位置	WORD		1	1000. 1 100. 0.1 10. 0.01 1. 0.001
R+3	每圈脈波數	DWORD		2	精度：小數點位置
R+5	每圈單位長度	DWORD		2	精度：小數點位置
R+7	速度單位	WORD			0. PLS/Sec 1. PLS/min 2. RPM
R+8	速度增益	DWORD		2	精度：0.001
R+10	開始速度	DWORD		2	精度：小數點位置
R+12	最大馬達速度	DWORD		2	精度：1 單位 RPM
R+14	預設加速度	DWORD		2	精度：小數點位置
R+16	預設減速度	DWORD		2	精度：小數點位置
R+18	軟限制 +	DWORD		2	精度：小數點位置 可負數
R+20	軟限制 -	DWORD		2	精度：小數點位置 可負數
R+22	跟蹤誤差容許範圍	DWORD		2	精度：小數點位置
R+24	跟蹤誤差容許時間	DWORD		2	單位ms
R+26	定位完成容許誤差	DWORD		2	精度：小數點位置
R+28	定位完成容許時間	DWORD		2	單位ms
R+30	最大馬達扭矩	WORD		1	精度：0.1
R+31	最大扭矩限制 +	WORD		1	精度：0.1
R+32	最大扭矩限制 -	WORD		1	精度：0.1
R+41	停止模式	WORD		1	5. 減速停止 7. 立即停止
R+42	停止減速度	DWORD		2	精度：小數點位置
R+44	復歸模式	WORD		1	99. 當前位置為原點 100. Dog Forward 101. 近點復歸 102. Dog-z-sig Forward 103. Dog-z-sig Backward
R+45	復歸 IO 來源	WORD		1	0. 從伺服驅動器 1. 從PLC
R+46	復歸開始方向	WORD		1	0. 負方向 1. 正方向
R+47	原點復歸偏移	DWORD		2	精度：小數點位置 (可負數)
R+49	復歸搜尋速度	DWORD		2	精度：小數點位置
R+51	復歸爬行速度	DWORD		2	精度：小數點位置
R+53	復歸減速度	DWORD		2	精度：小數點位置
R+55	極限開關 - 位元	WORD		1	
R+56	極限開關 + 位元	WORD		1	
R+57	原點開關位元	WORD		1	
R+58	原點零點訊號數	DWORD		2	
R+60	JOG 啟動速度	DWORD		2	精度：小數點位置
R+62	JOG 速度	DWORD		2	精度：小數點位置
R+64	JOG 加速度	DWORD		2	精度：小數點位置
R+66	JOG 減速度	DWORD		2	精度：小數點位置
R+68	寸動距離	DWORD		2	精度：小數點位置

配方同步表

序號	項目	資料大小	資料類型	長度	定義
R+0	輸入軸座標單位	WORD		1	
R+1	輸入軸小數點位置	WORD		1	
R+2	輸入軸週期	DWORD		2	精度：輸入軸小數點位置
R+4	減速停止滑動時間	DWORD		2	
R+6	輸入軸相位初始化方法	WORD		1	
R+7	同步主軸相位預設值	DWORD		2	精度：輸入軸小數點位置
R+9	相位補償後主軸相位預設值	DWORD		2	精度：輸入軸小數點位置
R+11	主離合器輸入項預設值	DWORD		2	精度：輸入軸小數點位置
R+13	輔助離合器輸入相位預設值	DWORD		2	精度：輸入軸小數點位置
R+15	凸輪輸入軸相位初始化方法	WORD		1	
R+16	主離合器輸出相位預設值	DWORD		2	精度：輸入軸小數點位置
R+18	輔助離合器輸出相位預設值	DWORD		2	精度：輸入軸小數點位置
R+20	保留	DWORD		2	
R+22	凸輪輸入相位預設值	DWORD		2	精度：輸入軸小數點位置
R+24	凸輪輸出基準座標	DWORD		2	精度：輸入軸小數點位置
R+26	主軸 1 選擇輸入軸	WORD		1	
R+27	主軸 1 外部參照編號	WORD		1	
R+28	主軸 1 防止逆轉	WORD		1	
R+29	主軸 1 座標轉換設定	WORD		1	
R+30	主軸 1 座標轉換分子	DWORD		2	
R+32	主軸 1 座標轉換分母	DWORD		2	
R+34	主軸 2 選擇輸入軸	WORD		1	
R+35	主軸 2 外部參照編號	WORD		1	
R+36	主軸 2 防止逆轉	WORD		1	
R+37	主軸 2 座標轉換設定	WORD		1	
R+38	主軸 2 座標轉換分子	DWORD		2	
R+40	主軸 2 座標轉換分母	DWORD		2	
R+42	輔助軸選擇輸入軸	WORD		1	
R+43	輔助軸外部參照編號	WORD		1	
R+44	輔助軸防止逆轉	WORD		1	
R+45	輔助軸座標轉換設定	WORD		1	
R+46	輔助軸座標轉換分子	DWORD		2	
R+48	輔助軸座標轉換分母	DWORD		2	
R+50	主軸相位補償指令量	DWORD		2	精度：輸入軸小數點位置
R+52	主軸相位補償更改模式	WORD		1	
R+53	主軸相位補償更改時間	DWORD		2	
R+55	輔助軸相位補償指令量	DWORD		2	精度：輸入軸小數點位置
R+57	輔助軸相位補償更改模式	WORD		1	
R+58	輔助軸相位補償更改時間	DWORD		2	
R+60	可變齒輪比分子	DWORD		2	
R+62	可變齒輪比分母	DWORD		2	
R+64	可變齒輪比更改模式	WORD		1	
R+65	可變齒輪比更改時間	DWORD		2	
R+67	主離合器 ON 條件	WORD		1	
R+68	主離合器 ON 設定值	DWORD		2	精度：輸入軸小數點位置
R+70	主離合器 ON 延遲	DWORD		2	精度：輸入軸小數點位置
R+73	主離合器 ON 連接方式	WORD		1	
R+75	主離合器 ON 滑動曲線	WORD		1	
R+78	主離合器 ON 滑動時間	DWORD		2	
R+80	主離合器 ON 隨動時間	DWORD		2	
R+82	主離合器 ON 隨動量	DWORD		2	精度：輸入軸小數點位置
R+84	主離合器 OFF 條件	WORD		1	
R+85	主離合器 OFF 設定值	DWORD		2	精度：輸入軸小數點位置
R+87	主離合器 OFF 延遲	DWORD		2	精度：輸入軸小數點位置
R+90	主離合器 OFF 連接方式	WORD		1	
R+92	主離合器 OFF 滑動曲線	WORD		1	
R+95	主離合器 OFF 滑動時間	DWORD		2	
R+97	輔助離合器 ON 條件	WORD		1	
R+98	輔助離合器 ON 設定值	DWORD		2	精度：輸入軸小數點位置
R+100	輔助離合器 ON 延遲	DWORD		2	精度：輸入軸小數點位置
R+103	輔助離合器 ON 連接方式	WORD		1	
R+105	輔助離合器 ON 滑動曲線	WORD		1	
R+108	輔助離合器 ON 滑動時間	DWORD		2	
R+110	輔助離合器 ON 隨動時間	DWORD		2	
R+112	輔助離合器 ON 隨動量	DWORD		2	精度：輸入軸小數點位置
R+114	輔助離合器 OFF 條件	WORD		1	
R+115	輔助離合器 OFF 設定值	DWORD		2	精度：輸入軸小數點位置
R+117	輔助離合器 OFF 延遲	DWORD		2	精度：輸入軸小數點位置
R+120	輔助離合器 OFF 連接方式	WORD		1	
R+122	輔助離合器 OFF 滑動曲線	WORD		1	
R+125	輔助離合器 OFF 滑動時間	DWORD		2	
R+132	步進角補償基準速度	DWORD		2	精度：輸入軸小數點位置
R+134	步進角補償基準量	DWORD		2	精度：輸入軸小數點位置
R+136	步進角補償更改方式	WORD		1	
R+137	步進角補償更改時間	DWORD		2	
R+139	凸輪資料編號	WORD		1	
R+140	凸輪行程	DWORD		2	精度：軸表小數點位置
R+142	同步接點編號	WORD		1	
R+143	輸出濾波器時間常數	DWORD		2	

程式範例

階梯圖	ST
	<pre>MFSysRCPW (EN:= M1000, Md:= 0, D:= R1000, Gp:= 0, ACT=> M1001, ERR=> M1002, DN=> M1003);</pre>

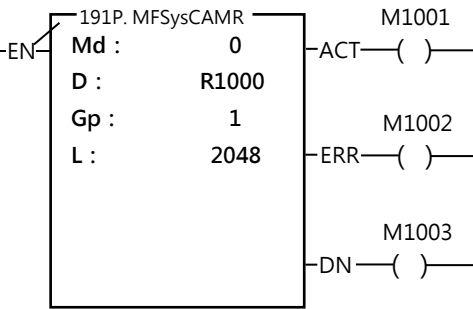
- M1000 由 OFF→ON 時，從 R1000 寫入全部配方表。

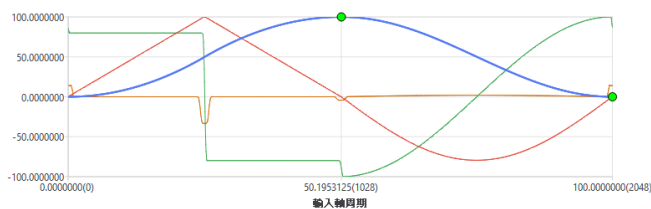
7-22-22 凸輪讀取(MFSysCAMR)

Fun191 P MFSysCAMR		【運動控制凸輪讀取】										Fun191 P MFSysCAMR																																																																																													
指令說明																																																																																																									
		階段圖符號 執行控制 EN 191P. MFSysCAMR MD : D : ID : ACT動作中 ERR錯誤 DN執行完成										運算元 Md : 模式 D : 凸輪起始暫存器 ID : 凸輪編號 L : 凸輪解析度																																																																																													
		繼電器和暫存器 <table><tr><th> 範圍 運算元 </th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>WX0 WX1008</td><td>WY0 WY1008</td><td>WM0 WM9104</td><td>WS0 WS3088</td><td>T0 T1023</td><td>C0 C1279</td><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V, Z P0 ~ P9</td></tr><tr><td>Md</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~1</td><td></td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>ID</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>1~16</td><td></td></tr><tr><td>L</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>2048~32767</td><td></td></tr></table>														範圍 運算元	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9	Md	○	○	○	○	○	○	○	○	○	○	○	○	0~1		D	○	○	○	○	○	○	○	○	○	○	○	○		○	ID	○	○	○	○	○	○	○	○	○	○	○	○	1~16		L	○	○	○	○	○	○	○	○	○	○	○	○	2048~32767	
範圍 運算元	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																											
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9																																																																																											
Md	○	○	○	○	○	○	○	○	○	○	○	○	0~1																																																																																												
D	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																																											
ID	○	○	○	○	○	○	○	○	○	○	○	○	1~16																																																																																												
L	○	○	○	○	○	○	○	○	○	○	○	○	2048~32767																																																																																												
功能描述																																																																																																									

- 運算元
Md 模式 : 0 使用 PLC 暫存器
D 凸輪起始暫存器 : 讀取凸輪後要存放的暫存器起始位址
ID 凸輪編號 : 凸輪編號
L 凸輪解析度 : 讀取凸輪後要存放的暫存器長度
- 當執行控制【EN】由上微分觸發，Fun191 將指定凸輪讀取至指定的暫存器
當執行控制【EN】由下微分觸發，復位所有輸出指示
- 當讀取凸輪中，則輸出指示【ACT】ON。
- 當讀取凸輪中，如有錯誤發生，則輸出指示【ERR】ON。
- 當讀取凸輪完成，輸出指示【DN】ON。

程式範例

階梯圖	ST
	<pre>MFSysCAMR (EN:= TRUE, Md:= 0, D:= R1000, ID:= 1, L:= 2048, ACT=> M1001, ERR=> M1002, DN=> M1003);</pre>

CAM

- M1000 由 OFF→ON 時，讀取凸輪 ID：1 資料表 2048 個存放至 DR1000~DR5094。

資料表

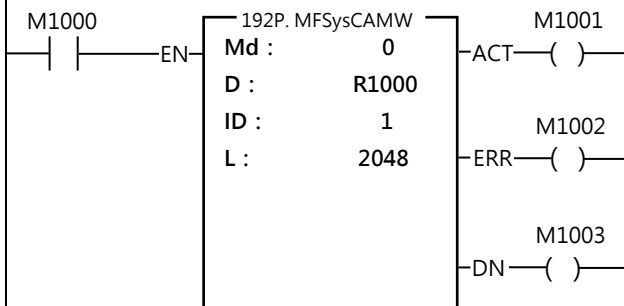
凸輪資料表		
相位	編號	位移
99.6582031(2041)	2041	0.0116203
99.7070312(2042)	2042	0.0085375
99.7558594(2043)	2043	0.0059289
99.8046875(2044)	2044	0.0037945
99.8535156(2045)	2045	0.0021344
99.9023438(2046)	2046	0.0009486
99.9511719(2047)	2047	0.0002372
100.0000000(2048)	2048	0.0000000

7-22-23 凸輪寫入(MFSysCAMW)

Fun192 P MFSysCAMW		【運動控制凸輪寫入】										Fun192 P MFSysCAMW		
指令說明														
階段圖符號 執行控制 EN 192P. MFSysCAMW MD : D : ID : ACT動作中 ERR錯誤 DN執行完成 運算元 Md : 模式 D : 凸輪起始暫存器 ID : 凸輪編號 L : 凸輪解析度														
繼電器和暫存器														
運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
	Md	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	0~1
D	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐		☐
ID	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	1~16	
L	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	2048~32767	
功能描述														
● 運算元 Md 模式 : 0 使用 PLC 暫存器 D 凸輪起始暫存器 : 寫入凸輪的暫存器起始位址 ID 凸輪編號 : 凸輪編號 L 凸輪解析度 : 寫入凸輪的暫存器起始長度 ● 當執行控制【EN】由上微分觸發，Fun191 將指定暫存器寫入至指定的凸輪 當執行控制【EN】由下微分觸發，復位所有輸出指示 ● 當寫入凸輪中，則輸出指示【ACT】ON。 ● 當寫入凸輪中，如有錯誤發生，則輸出指示【ERR】ON。 ● 當寫入凸輪完成，輸出指示【DN】ON。														

程式範例

階梯圖

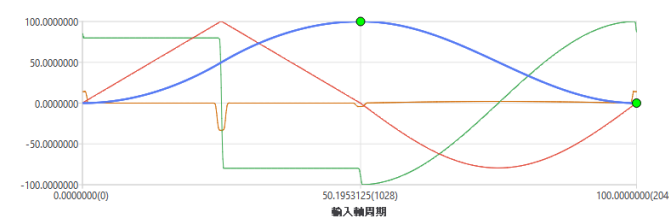


ST

```

MFSysCAMW ( EN:= M1000, Md:= 0, D:=
R1000, ID:= 1, L:= 2048, ACT=> M1001,
ERR=> M1002, DN=> M1003);
  
```

CAM



- M1000 由 OFF→ON 時，從 DR1000~DR5094 寫入凸輪 ID：1 資料表 2048 個。

資料表

相位	編號	位移
99.6582031(2041)	2041	0.0116203
99.7070312(2042)	2042	0.0085375
99.7558594(2043)	2043	0.0059289
99.8046875(2044)	2044	0.0037945
99.8535156(2045)	2045	0.0021344
99.9023438(2046)	2046	0.0009486
99.9511719(2047)	2047	0.0002372
100.0000000(2048)	2048	0.0000000

確定 取消

7-22-24 手搖輪 (MFGearMPG)

Fun193 DP MFGearMPG		【EtherCAT 手搖輪】												Fun193 DP MFGearMPG	
指令說明															
階段圖符號 執行控制 更新參數 193P. MFGearMPG M : S : N : D : ACT動作中 ERR錯誤 DN執行完成 UPD更新完成 運算元 M : 主軸輸入來源 S : 從軸輸出目標 N : 可變齒輪比分子 D : 可變齒輪比分母 T : 轉換時間(ms)															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
M		○	○	○	○	○	○	○	○	○	○	○	○	1~16,100~108	
S		○	○	○	○	○	○	○	○	○	○	○	○	1~16	○
N		○	○	○	○	○	○	○	○	○	○	○	○		
D		○	○	○	○	○	○	○	○	○	○	○	○		
T		○	○	○	○	○	○	○	○	○	○	○	○		
功能描述															

- Fun193 (EtherCAT 手搖輪)整合位置同步的手搖輪相關設定，提供使用者快速設定手搖輪。
- 運算元
 - M 主軸輸入來源 : EtherCAT_軸編號 1~16
 - : 編碼器_格雷碼 100 (X8~X15)
 - : 編碼器_硬體高速計數編號 101 ~ 104 (HSC4~HSC7)
 - S 從軸輸出目標 : EtherCAT_軸編號 1~16 (【M 主軸輸入來源】不能跟【S 從軸輸出目標】相同)
 - N 可變齒輪比分子 : 正負數，包含【運動控制】中【運動軸設定】的【小數點位置】
(【軸單位】設定 mm，【小數點位置】設定 0.001，N : DR0 = 1000 等於 1.000mm)
 - D 可變齒輪比分母 : 正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】
 - T 轉換時間(ms) : 正數 (大於零的實數)，單位為 ms
- 當執行控制【EN】由上微分觸發，Fun193 使用當前參數啟動手搖輪位置同步控制
- 當執行控制【EN】由下微分觸發，Fun193 停止手搖輪位置同步控制，並復位所有輸出指示
- 在手搖輪同步控制中，若更新參數【UPD】變為 1，則本指令將立即更新手搖輪參數(N、D、T)。
- 當手搖輪同步控制中，輸出指示【ACT】ON。
- 當手搖輪同步控制中，如有錯誤發生，則輸出指示【ERR】ON。
- 當更新手搖輪參數完成，輸出指示【UPD】ON。

程式範例

階梯圖	ST
	<pre> MFGearMPG_D(EN:= M1000, UPD_in:= M1005, M:= 1, S:= 2, N:= R1000, D:= R1002, T:= R1004, ACT=> M1001, ERR=> M1002, DN=> M1003, UPD_out=> M1004); IF M1004 THEN M1005 := FALSE; END_IF </pre>

軸參數設定

		1	2
基本設定	軸名稱	軸_1	軸_2
	軸類型	伺服	伺服
	編碼器類型	增量型	增量型
單位設定	單位	mm	mm
	小數點位置	0.001	0.001
	脈波/圈	131072 PLS/Rev	131072 PLS/Rev
	單位/圈	1.000 mm/Rev	1.000 mm/Rev
	速度單位	指令位置/秒	指令位置/秒
	速度增益	1.000	1.000

- M1000 由 OFF→ON 時，依照當前 Fun193 參數(M：EtherCAT 軸 1、N：EtherCAT 軸 2、N：可變齒輪比分子 0.001、D 可變齒輪比分母 0.001、T：1ms)啟動手搖輪同步。

階梯圖	ST
	<pre> MFGearMPG_D(EN:= M1000, UPD_in:= M1005, M:= 1, S:= 2, N:= R1000, D:= R1002, T:= R1004, ACT=> M1001, ERR=> M1002, DN=> M1003, UPD_out=> M1004); </pre>

- 變更參數(D 可變齒輪比分母 0.002)後，M1005 由 OFF→ON 時，依照變更後的參數更新手搖輪，參數更新完成，輸出指示【UPD】ON，手搖輪從軸行程減半。

7-22-25 速度控制模式(MFVelCtl)

Fun194 DP MFVelCtl		【速度控制模式】										Fun194 DP MFVelCtl																																																																													
指令說明																																																																																									
		階段圖符號 執行控制—EN 更新參數—UPD 194P.MFVelCtl S : V : MX : ACT—動作中 ERR—錯誤 DN—執行完成 UPD—更新完成										運算元 S : EtherCAT 速度控制軸 V : 速度 MX : 最大扭矩限制																																																																													
		繼電器和暫存器 <table><tr><th rowspan="3">運算元 範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th>WX0 WX1008</th><th>WY0 WY1008</th><th>WM0 WM9104</th><th>WS0 WS3088</th><th>T0 T1023</th><th>C0 C1279</th><th>R0 R34767</th><th>R34768 R34895</th><th>R35024 R35151</th><th>R35280 R43223</th><th>R43224 R47319</th><th>D0 D7999</th><th></th><th>V, Z P0 ~ P9</th></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>1~16</td><td></td></tr><tr><td>V</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr><tr><td>MX</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr></table>														運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9	S	○	○	○	○	○	○	○	○	○	○	○	○	1~16		V	○	○	○	○	○	○	○	○	○	○	○	○		○	MX	○	○	○	○	○	○	○	○	○	○	○	○		
運算元 範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																											
	WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9																																																																											
	S	○	○	○	○	○	○	○	○	○	○	○	○	1~16																																																																											
V	○	○	○	○	○	○	○	○	○	○	○	○		○																																																																											
MX	○	○	○	○	○	○	○	○	○	○	○	○																																																																													
功能描述																																																																																									

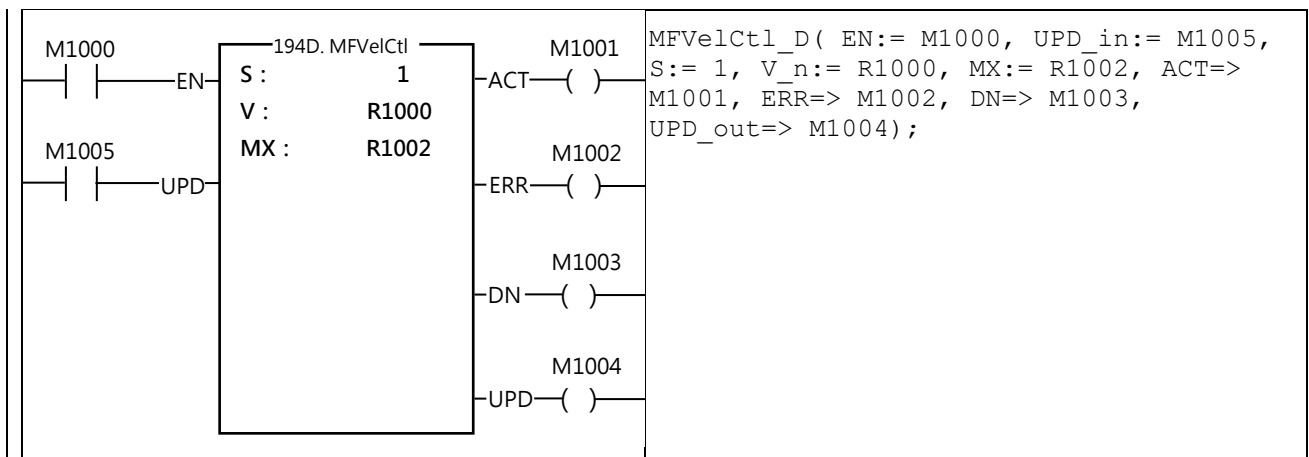
- 運算元
 - S 速度控制軸 : EtherCAT_軸編號 1~16
 - V 速度 : 速度設定值，單位 Pulses/s
 - MX 最大扭矩限制 : 當速度無法到達速度設定值，最大的扭矩限制，0 等於沒有限制，單位 0.0%
- 當執行控制【EN】由上微分觸發，Fun194 使用當前參數啟動軸速度控制
- 當執行控制【EN】由下微分觸發，Fun194 停止軸速度控制，並復位所有輸出指示
- 在軸速度控制中，若更新參數【UPD】變為 1，則本指令將立即更新速度控制參數(V、MX)。
- 當軸速度控制中，輸出指示【ACT】ON。
- 當軸速度控制中，如有錯誤發生，則輸出指示【ERR】ON。
- 當更新速度控制參數完成，輸出指示【UPD】ON。

程式範例

階梯圖	ST
<pre> M1000 --- --- EN M1005 --- --- UPD S: 1 V: R1000 MX: R1002 ACT --- () --- M1001 ERR --- () --- M1002 DN --- () --- M1003 UPD --- () --- M1004 </pre>	<pre> MFVelCtl_D(EN:= M1000, UPD_in:= M1005, S:= 1, V_n:= R1000, MX:= R1002, ACT=> M1001, ERR=> M1002, DN=> M1003, UPD_out=> M1004); </pre>

- M1000 由 OFF→ON 時，依照當前 Fun194 參數(S：EtherCAT 軸 1、V：每秒 131072 個 Pulse、MX：沒有扭矩限制)啟動速度控制。

階梯圖	ST
-----	----



- 變更參數參數(V：每秒 262144 個 Pulse)後，M1005 由 OFF→ON 時，依照變更後的參數更新速度，參數更新完成，輸出指示 M1004【UPD】ON，速度變成 2 倍。

7-22-26 扭矩控制模式(MFTorqCtl)

Fun195 DP MFTorqCtl	【 扭矩控制模式 】	Fun195 DP MFTorqCtl													
指令說明															
階段圖符號 執行控制 更新參數 195P. MFTorqCtl S : T : MX : ACT-動作中 ERR-錯誤 DN-執行完成 UPD-更新完成 運算元 S : EtherCAT 扭矩控制軸 T : 設定扭矩 MX : 最大速度限制															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
	S	○	○	○	○	○	○	○	○	○	○	○	○	1~16	
	T	○	○	○	○	○	○	○	○	○	○	○	○		○
	MX	○	○	○	○	○	○	○	○	○	○	○	○		
功能描述															

- 運算元
 - S 扭矩控制軸 : EtherCAT_軸編號 1~16
 - T 扭矩 : 扭矩設定值，單位 0.0%
 - MX 最大速度限制 : 當扭矩無法到達扭矩設定值，最大的速度限制，0 等於沒有限制，單位 rpm
- 當執行控制【EN】由上微分觸發，Fun195 使用當前參數啟動軸扭矩控制
當執行控制【EN】由下微分觸發，Fun195 停止軸扭矩控制，並復位所有輸出指示
- 在軸扭矩控制中，若更新參數【UPD】變為 1，則本指令將立即更新扭矩控制參數(T、MX)。
- 當軸扭矩控制中，輸出指示【ACT】ON。
- 當軸扭矩控制中，如有錯誤發生，則輸出指示【ERR】ON。
- 當更新扭矩控制參數完成，輸出指示【UPD】ON。

程式範例

階梯圖	ST
	<pre> MFTorqCtl (EN:= M1000, UPD_in:= M1005, S:= 1, V_n:= R1000, MX:= R1002, ACT=> M1001, ERR=> M1002, DN=> M1003, UPD_out=> M1004); </pre>

- M1000 由 OFF→ON 時，依照當前 Fun194 參數(S：EtherCAT 軸 1、T：5.0%、MX：沒有速度限制)啟動扭矩控制。

階梯圖	ST
	<pre> MFTorqCtl (EN:= M1000, UPD_in:= M1005, S:= 1, V_n:= R1000, MX:= R1002, ACT=> M1001, ERR=> M1002, DN=> M1003, UPD_out=> M1004); </pre>

- 變更參數參數(T：10.0%)後，M1005 由 OFF→ON 時，依照變更後的參數更新扭矩，參數更新完成，輸出指示 M1004【UPD】ON，扭矩變成 2 倍。

7-22-27 生成凸輪(MFSysCAMGen)

Fun196 DP MFSysCAMGen		【生成凸輪】												Fun196 DP MFSysCAMGen	
指令說明															
階段圖符號 執行控制 EN 196P. MFSysCAMGen ID : Md : D : ACT動作中 ERR錯誤 DN執行完成 運算元 ID : 凸輪編號 Md : 凸輪生成模式 D : 暫存器起始位址 L : 凸輪曲線段數															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z
															P0 ~
		WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		P9
	ID	○	○	○	○	○	○	○	○	○	○	○	○	1~16	
	Md													0~1	
D	○	○	○	○	○	○	○	○	○	○	○	○			
L	○	○	○	○	○	○	○	○	○	○	○	○			
功能描述															

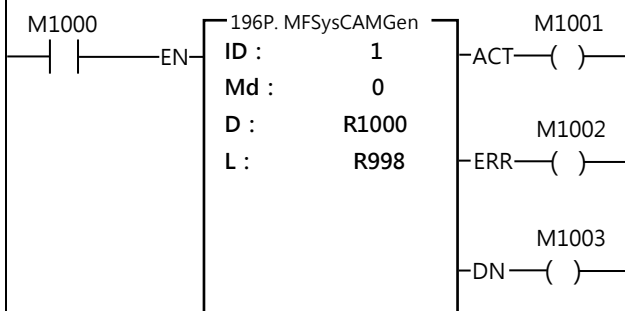
- 運算元
ID 凸輪編號 : 1~16
Md 凸輪生成模式 : 0 同凸輪表、1 追剪曲線
D 暫存器起始位 : 設定凸輪的起始暫存器
L 凸輪曲線段數 : 模式 0 才有每一段凸輪設定，其他模式不用設定
- 當執行控制【EN】由上微分觸發，Fun196 將依照設定模式生成凸輪。
- 當執行控制【EN】由下微分觸發，復位所有輸出指示
- 當凸輪生成中，輸出指示【ACT】ON。
- 當凸輪生成中，如有錯誤發生，則輸出指示【ERR】ON。
- 當凸輪生成完成，則輸出指示【DN】ON。

模式 0

暫存器	項目	定義	
D+0	開始相位	0~凸輪解析度	第一段凸輪
D+2	結束相位	0~凸輪解析度 (必須銜接下一段開始相位)	
D+4	偏移	0~1000000000 (0~100.00000000%)	
D+6	凸輪輪廓	0:等速度 1:等加速度 2:擺線 3:雙躍 4:變形等速度 5:變形梯形 6:變形正弦 7:梯形 8:無停置擺線, M=1 9:無停置擺線, M=2/3 10:無停置梯形 11:無停置變形正弦 12:無停置梯形 13:無停置變形梯形 14:無停置變形等速度 15:NC2 16:非對稱擺線 17:非對稱變形梯形 18:三次曲線 19:五次曲線	
D+8	開始速度	固定3位小數點	
D+10	結束速度	固定3位小數點	
D+12	開始加速度	固定3位小數點	
D+14	結束加速度	固定3位小數點	
D+15	開始相位	0~凸輪解析度	
D+16	結束相位	0~凸輪解析度 (必須銜接下一段開始相位)	
		:	第二段凸輪

程式範例

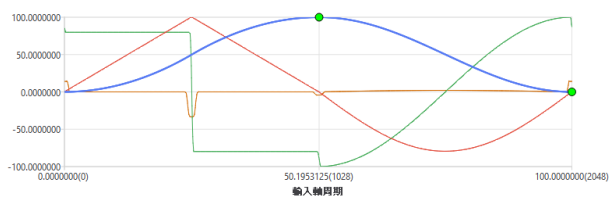
階梯圖



ST

```
MFSysCAMGen ( EN:= M1000, ID:= 1, Md:=
0, D:= R1000, L:= R998, ACT=> M1001,
ERR=> M1002, DN=> M1003);
```

CAM



D

暫存器	項目	定義	
R1000	開始相位	0	第一段凸輪
R1001	結束相位	1028	
R1002	偏移	1000000000	
R1003	凸輪輪廓	1:等加速度	
R1004	開始速度	0	
R1005	結束速度	0	
R1006	開始加速度	0	
R1007	結束加速度	0	第二段凸輪
R1008	開始相位	1028	
R1009	結束相位	2048	
R1010	偏移	0	
R1011	凸輪輪廓	3:簡諧	
R1012	開始速度	0	
R1013	結束速度	0	
R1014	開始加速度	0	
R1015	結束加速度	0	

凸輪參數

	開始相位	結束相位	偏移	凸輪輪廓
1	0.0000000(0)	50.1953125(1028)	100.0000000	等加速度
2	50.1953125(1028)	100.0000000(2048)	0.0000000	簡諧

- M1000 由 OFF→ON 時，依照當前 Fun196 數(ID：凸輪編號 1、Md：模式 0、D：從 R1000 開始設定凸輪生成參數、L：2 段凸輪曲線)執行凸輪生成。

7-22-28 單軸定位 (MFAXMov)

Fun197 DP MFAxMov		【 軸移動 】										Fun197 DP MFAxMov			
指令說明															
		階段圖符號						運算元							
執行控制—EN		197P. MFAxMov						S : EtherCAT 控制軸							
		S :						MD : 運行模式							
		MD :						PS : 目標位置							
更新參數—UPD		Ps :						V : 速度							
		V :						A : 加速度							
		A :						D : 減速度							
		D :						SA : S 加速度曲線%							
		SA :						SD : S 減速度曲線%							
		SD :						DR : 方向							
		DR :						BF : 速度連續模式							
		BF :													
		繼電器和暫存器													
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
S		○	○	○	○	○	○	○	○	○	○	○	○	1~16	
MD														0~1	
Ps		○	○	○	○	○	○	○	○	○	○	○	○		
V		○	○	○	○	○	○	○	○	○	○	○	○		
A		○	○	○	○	○	○	○	○	○	○	○	○		
D		○	○	○	○	○	○	○	○	○	○	○	○		
SA		○	○	○	○	○	○	○	○	○	○	○	○		
SD		○	○	○	○	○	○	○	○	○	○	○	○		
DR		○	○	○	○	○	○	○	○	○	○	○	○	1~2	
BF														0~5	
功能描述															

- 運算元
 - S EtherCAT 控制軸 : EtherCAT_軸編號 1~16
 - MD 運行模式 : 0 絕對、1 相對
 - PS 目標位置 : 正負數，包含【運動控制】中【運動軸設定】的【小數點位置】
(【軸單位】設定 mm，【小數點位置】設定 0.001，PS : DR0 = 1000 等於 1.000mm)
 - V 速度 : 正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】
 - A 加速度 : 正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】
 - D 減速度 : 正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】
 - SA S 加速度曲線% : 正整數，0~1000 ‰
 - SD S 減速度曲線% : 正整數，0~1000 ‰
 - DR 方向 : 1 正方向、2 負方向
 - BF : 速度連續模式 : 0 立即執行當前指令、1 等待前一指令結束、2 選擇較低速度連續、3 選擇前一指令速度連續、4 選擇當前指令速度連續、5 選擇較高速度連續
- 當執行控制【EN】由上微分觸發，Fun197 執行軸位置控制
- 當執行控制【EN】由下微分觸發，Fun197 停止軸位置控制，並復位所有輸出指示
- 在軸位置控制中，若更新參數【UPD】變為 1，則本指令將立即更新位置控制參數(S、PS、V、A、D、SA、SD、DR)。
- 當軸位置控制中，輸出指示【ACT】ON。
- 當軸位置控制中，如有錯誤發生，則輸出指示【ERR】ON。
- 當軸位置控制完成，則輸出指示【DN】ON。
- 當更新位置控制參數完成，輸出指示【UPD】ON。

程式範例

階梯圖	ST																																
M1000 M1005 197D. MFAxMov S : 1 MD : 1 Ps : R1000 V : R1002 A : R1004 D : R1006 SA : R1008 SD : R1010 DR : R1012 BF : 0 M1001 M1002 M1003 M1004	MFAxMov_D(EN:= M1000, UPD_in:= M1005, S:= 1, MD:= 1, Ps:= R1000, V_n:= R1002, A:= R1004, D:= R1006, SA:= R1008, SD:= R1010, DR:= R1012, BF:= 0, ACT=> M1001, ERR=> M1002, DN=> M1003, UPD_out=> M1004); 軸參數設定 <table><tr><td></td><td>1</td><td>2</td></tr><tr><td rowspan="3">基本設定</td><td>軸名稱</td><td>軸_1</td><td>軸_2</td></tr><tr><td>軸類型</td><td>伺服</td><td>伺服</td></tr><tr><td>編碼器類型</td><td>增量型</td><td>增量型</td></tr><tr><td rowspan="6">單位設定</td><td>單位</td><td>mm</td><td>mm</td></tr><tr><td>小數點位置</td><td>0.001</td><td>0.001</td></tr><tr><td>脈波/圈</td><td>131072 PLS/Rev</td><td>131072 PLS/Rev</td></tr><tr><td>單位/圈</td><td>1.000 mm/Rev</td><td>1.000 mm/Rev</td></tr><tr><td>速度單位</td><td>指令位置/秒</td><td>指令位置/秒</td></tr><tr><td>速度增益</td><td>1.000</td><td>1.000</td></tr></table>		1	2	基本設定	軸名稱	軸_1	軸_2	軸類型	伺服	伺服	編碼器類型	增量型	增量型	單位設定	單位	mm	mm	小數點位置	0.001	0.001	脈波/圈	131072 PLS/Rev	131072 PLS/Rev	單位/圈	1.000 mm/Rev	1.000 mm/Rev	速度單位	指令位置/秒	指令位置/秒	速度增益	1.000	1.000
	1	2																															
基本設定	軸名稱	軸_1	軸_2																														
	軸類型	伺服	伺服																														
	編碼器類型	增量型	增量型																														
單位設定	單位	mm	mm																														
	小數點位置	0.001	0.001																														
	脈波/圈	131072 PLS/Rev	131072 PLS/Rev																														
	單位/圈	1.000 mm/Rev	1.000 mm/Rev																														
	速度單位	指令位置/秒	指令位置/秒																														
	速度增益	1.000	1.000																														

●

M1000 由 OFF→ON 時，依照當前 Fun197 參數(S：EtherCAT 軸 1、MD：相對位置、PS：移動到 10.000mm、V：速度1.000mm/s、A：加速度100.000mm/s²、D：減速度100.000mm/s²、SA：S 加速度曲線0.0%、SD：S 減速度曲線0.0%、DR：正方向、BF：立即執行當前指令)執行位置控制。

7-22-29 寫入映射表 (MMapTbPrm)

Fun198 P MMapTbPrm		【寫入映射表】												Fun198 P MMapTbPrm		
指令說明																
		階段圖符號										運算元				
執行控制		198P. MMapTbPrm Gp : N : L : ACT-動作中 ERR-錯誤 DN-執行完成										Gp : 映射表群組編號 N : 映射起始表編號 L : 映射連續長度				
繼電器和暫存器																
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR	
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9	
		Gp													0~64	
		N	○	○	○	○	○	○	○	○	○	○	○	○	0~1024	
		L	○	○	○	○	○	○	○	○	○	○	○	○	1~1024	
功能描述																
● 【Fun198 寫入映射表】用來改變單個或少數運動控制參數，如果需要大量讀取或寫入運動控制參數，可以使用【Fun188 配方讀取】和【Fun189 配方寫入】。 ● 運算元 Gp 映射表群組編號：群組編號 1~16，0 表示全部群組。 N 映射表起始表編號：映射表編號 1~1024，0 表示整張映射表 L 映射連續長度：連續映射項目數量 ● 當執行控制【EN】由上微分觸發，Fun198 將 PLC 暫存器映射(寫入)到運動控制參數 當執行控制【EN】由下微分觸發，復位所有輸出指示 ● 當映射寫入中，輸出指示【ACT】ON。 ● 當映射寫入中，如有錯誤發生，則輸出指示【ERR】ON。 ● 當映射寫入完成，則輸出指示【DN】ON。																

程式範例

階梯圖	ST
	<pre> MMapTbPrm (EN:= M1000, Gp:= 1, N:= 1, L:= 2, ACT=> M1001, ERR=> M1002, DN=> M1003); </pre>

映射表

1:PM1 ×					
	註解	表	索引	項目	地址
1		軸表	1	19JOG速度	R9000
2		軸表	1	22吋動距離	R9002

運動軸設定表

點動	JOG啟動速度	0.100 mm/s
	JOG速度	1.000 mm/s[2.000 mm/s]
	JOG加速度	1000.000 mm/s ²
	JOG減速度	1000.000 mm/s ²
	吋動距離	5.000 mm[6.000 mm]

- M1000 由 OFF→ON 時，依照當前 Fun198 參數(Gp 1：映射表單 1(1：PM)、N：從映射表單 (1：PM1)第一行開始、L：長度 1)執行映射表寫入，可以從運動軸設定表看到 JOG 速度被修改成 2.000mm/s²，吋動距離被修改成 6.000mm。

7-22-30 實軸轉換成虛擬軸 (MFSysSetVirt)

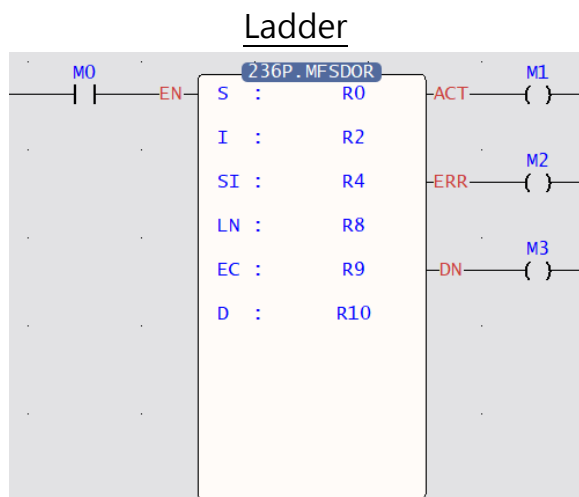
Fun235 P MFSysSetVirt		【實軸轉換成虛擬軸】										Fun235 P MFSysSetVirt			
指令說明															
階段圖符號 執行控制 EN 235P. MFSysSetVirt AX : ACT動作中 ERR錯誤 DN執行完成 運算元 AX：欲轉換之軸編號 EN：觸發指令 ACT：執行中 ERR：轉換失敗 DN：執行完成															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z P0 ~ P9
		WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		
ID		○	○	○	○	○	○	○	○	○	○	○	○	1~16	○
功能描述															
- 此指令是將實軸轉換成虛擬軸的指令。 - 使用前須確保運動控制系統為停止狀態,如果在初始化狀態,則 ERR 將輸出 1。 - 如需將已初始化的系統停下來,可以參考 FUN177 停止所有運動流程指令的說明。 - 此指令詳細資訊，請參考運動控制手冊中的說明。															
程式範例															
Ladder 待補充															

7-22-31 運動控制 SDO 映射讀取 MFSDOR

Fun236 P MFSDOR		【運動控制 SDO 映射讀取】												Fun236 P MFSDOR																																																																																																																																															
指令說明																																																																																																																																																													
階段圖符號 執行控制 EN 236P. MFSDOR S : I : SI : LN : EC : D : ACT動作中 ERR錯誤 DN執行完成 運算元 S : 控制軸編號 I : 索引 SI : 子索引 LN : 資料長度 EC : 錯誤碼站存器 D : 讀取資料 (DWORD) 繼電器和暫存器 <table><tr><th rowspan="3">運算元</th><th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th></th><th>WX0 WX1008</th><th>WY0 WY1008</th><th>WM0 WM9104</th><th>WS0 WS3088</th><th>T0 T1023</th><th>C0 C1279</th><th>R0 R34767</th><th>R34768 R34895</th><th>R35024 R35151</th><th>R35280 R43223</th><th>R43224 R47319</th><th>D0 D7999</th><th></th><th>V, Z P0 ~ P9</th></tr><tr><th></th><th>S</th><th>I</th><th>SI</th><th>LN</th><th>EC</th><th>D</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr><tr><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td></td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr></table>																運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9		S	I	SI	LN	EC	D											○	○	○	○	○	○	○	○	○	○	○	○					○	○	○	○	○	○	○	○	○	○	○	○					○	○	○	○	○	○	○	○	○	○	○	○					○	○	○	○	○	○	○	○	○	○	○	○					○	○	○	○	○	○	○	○	○	○	○	○					○	○	○	○	○	○	○	○	○	○	○	○		
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR																																																																																																																																														
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9																																																																																																																																														
		S	I	SI	LN	EC	D																																																																																																																																																						
		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																
		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																
		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																
		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																
		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																
		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																
功能描述																																																																																																																																																													

- 運算元
 - S 控制軸編號：EtherCAT 從站編號
 - I 索引：EtherCAT 從站記憶體索引
 - SI 子索引：EtherCAT 從站記憶體子索引
 - LN 資料長度：資料長度
 - EC 錯誤碼暫存器：SDO 錯誤碼存放位置
 - D 讀取資料：讀取資料存放位置
- 當執行控制【EN】由上微分觸發，指令執行 SDO 讀取。
- 當指令執行中，輸出指示【ACT】ON。
- 當指令執行中，如有錯誤發生，則輸出指示【ERR】ON。
- 當指令執行完成，則輸出指示【DN】ON。

程式範例



- 假設要讀取軸一伺服驅動器狀態字 (0x6041)，此時 S 輸入 1、I 輸入 0x6041、SI 輸入 0、LN 輸入 2。
- 執行指令後，讀取料存放在 R10 的位置。

7-22-32 運動控制 SDO 映射寫入 MFSDOW

Fun237 P MFSDOW		【運動控制 SDO 映射寫入】												Fun237 P MFSDOW	
指令說明															
階段圖符號 執行控制 EN 236P. MFSDOW S : I : SI : LN : EC : D : ACT動作中 ERR錯誤 DN執行完成 運算元 S : 控制軸編號 I : 索引 SI : 子索引 LN : 資料長度 EC : 錯誤碼暫存器 D : 寫入資料															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
S		○	○	○	○	○	○	○	○	○	○	○	○		
I		○	○	○	○	○	○	○	○	○	○	○	○		
SI		○	○	○	○	○	○	○	○	○	○	○	○		
LN		○	○	○	○	○	○	○	○	○	○	○	○		
EC		○	○	○	○	○	○	○	○	○	○	○	○		
D		○	○	○	○	○	○	○	○	○	○	○	○		
功能描述															

- 運算元
 - S 控制軸編號：EtherCAT 從站編號
 - I 索引：EtherCAT 從站記憶體索引
 - SI 子索引：EtherCAT 從站記憶體子索引
 - LN 資料長度：資料長度
 - EC 錯誤碼暫存器：SDO 錯誤碼存放位置
 - D 寫入資料：寫入資料存放位置
- 當執行控制【EN】由上微分觸發，指令執行 SDO 寫入。
- 當指令執行中，輸出指示【ACT】ON。
- 當指令執行中，如有錯誤發生，則輸出指示【ERR】ON。
- 當指令執行完成，則輸出指示【DN】ON。

程式範例

Ladder

- 假設要寫入軸一伺服驅動器位置 0x3000 的參數，寫入 0x1011，此時 S 輸入 1、I 輸入 0x3000、SI 輸入 0、LN 輸入 2、R10 輸入 0x1011。
- 執行指令後，可使用 SDO 讀取指令驗證，或使用驅動器軟體查看參數是否寫入。

7-22-33 直線補間運動 MFAxLMov

Fun238 DP MFAxLMov		【直線補間運動】										Fun238 DP MFAxLMov			
指令說明															
階段圖符號 執行控制—EN 更新參數—UPD 238P. MFAxLMov TAX : MD : DR : BF : D : EC : ACT—動作中 ERR—錯誤 DN—執行完成 UPD—更新完成 運算元 TAX : 總軸數 MD : 運行模式 DR : 方向 BF : 速度連續模式 D : 運動資料暫存器起始位置 EC : 錯誤碼															
繼電器和暫存器															
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR
		WX0 WX1008	WY0 WY1008	WM0 WM9104	WS0 WS3088	T0 T1023	C0 C1279	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V, Z P0 ~ P9
TAX		○	○	○	○	○	○	○	○	○	○	○	○	1~4	
MD														0~1	
DR		○	○	○	○	○	○	○	○	○	○	○	○	1~2	
BF		○	○	○	○	○	○	○	○	○	○	○	○		
D		○	○	○	○	○	○	○	○	○	○	○	○		
EC		○	○	○	○	○	○	○	○	○	○	○	○		
功能描述															

- 運算元
TAX 總軸數：直線補間運動所需軸數 1~4 軸
MD 運行模式：0 絕對、1 相對
DR 方向：1 正方向、2 負方向
BF 速度連續模式：0 立即執行當前指令、1 等待前一指令結束、2 選擇較低速度連續、3 選擇前一指令速度連續、4 選擇當前指令速度連續、5 選擇較高速度連續
D 運動資料暫存器起始位置：設定直線補間運動的起始暫存器
EC 錯誤碼：存放錯誤碼暫存器
- 當執行控制【EN】由上微分觸發，Fun238 執行直線補間運動
當執行控制【EN】由下微分觸發，Fun238 停止直線補間運動，並復位所有輸出指示
- 在直線補間運動中，若更新參數【UPD】變為 1，則本指令將立即更新位置控制參數(D、DR)。
- 當直線補間運動中，輸出指示【ACT】ON。
- 當直線補間運動中，如有錯誤發生，則輸出指示【ERR】ON。
- 當直線補間運動完成，則輸出指示【DN】ON。
- 當更新位置控制參數完成，輸出指示【UPD】ON。

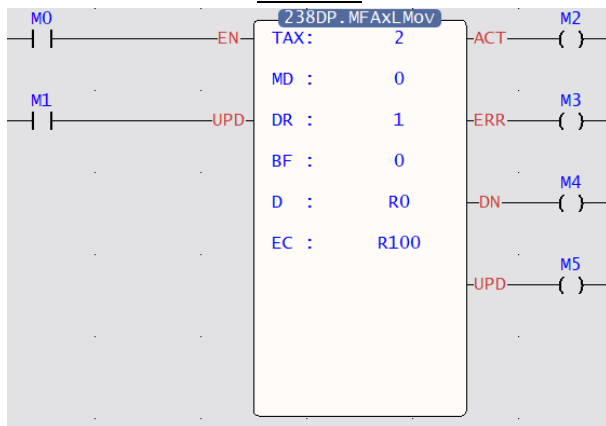
運動資料

暫存器	項目	資料長度	注意
R+0	主軸	WORD	
R+1	軸 2	WORD	
R+2	軸 3	WORD	
R+3	軸 4	WORD	
R+4	主軸目標位置	DWORD	帶正負號包含主軸小數點位
R+6	軸 2 目標位置	DWORD	帶正負號包含軸 2 小數點位
R+8	軸 3 目標位置	DWORD	帶正負號包含軸 3 小數點位
R+10	軸 4 目標位置	DWORD	帶正負號包含軸 4 小數點位
R+12	最高速度	DWORD	正數包含主軸小數點

			位
R+14	加速時間 (ms)	DWORD	從 0 加速至最高速所需時間
R+16	減速時間 (ms)	DWORD	從最高速減速至 0 所需時間
R+18	加速 S 曲線 (‰)	WORD	正整數範圍 0~1000 ‰
R+19	減速 S 曲線 (‰)	WORD	正整數範圍 0~1000 ‰

程式範例

Ladder



軸參數設定

		1	2
基本設定	軸名稱	軸_1	軸_2
	軸類型	伺服	伺服
	編碼器類型	增量型	增量型
單位設定	單位	mm	mm
	小數點位置	0.001	0.001
	脈波/圈	131072 PLS/Rev	131072 PLS/Rev
	單位/圈	1.000 mm/Rev	1.000 mm/Rev
	速度單位	指令位置/秒	指令位置/秒
	速度增益	1.000	1.000

- M0 由 OFF→ON 時，依照當前 Fun238 參數 TAX：直線補間運動所需軸數 2、MD：絕對位置、DR：正方向、BF：立即執行當前指令、D：運動資料暫存器起始位置 R0、EC：存放錯誤碼之暫存器 R100)執行直線補間運動。

7-22-34 圓弧補間運動 MFAxCirMov

Fun239D P MFAxCirMov		【圓弧補間運動】										Fun239D P MFAxCirMov	
指令說明													

- 運算元
TAX 總軸數：圓弧補間運動所需軸數 2~3 軸
MD 運行模式：0 絕對、1 相對
DR 方向：1 正方向、2 負方向
BF 速度連續模式：0 立即執行當前指令、1 等待前一指令結束、2 選擇較低速度連續、3 選擇前一指令速度連續、4 選擇當前指令速度連續、5 選擇較高速度連續
D 運動資料暫存器起始位置：設定圓弧補間運動的起始暫存器
EC 錯誤碼：存放錯誤碼暫存器
- 當執行控制【EN】由上微分觸發，Fun239 執行圓弧補間運動
當執行控制【EN】由下微分觸發，Fun239 停止圓弧補間運動，並復位所有輸出指示
- 在圓弧補間運動中，若更新參數【UPD】變為 1，則本指令將立即更新位置控制參數(D、DR)。
- 當圓弧補間運動中，輸出指示【ACT】ON。
- 當圓弧補間運動中，如有錯誤發生，則輸出指示【ERR】ON。
- 當圓弧補間運動完成，則輸出指示【DN】ON。
- 當更新位置控制參數完成，輸出指示【UPD】ON。

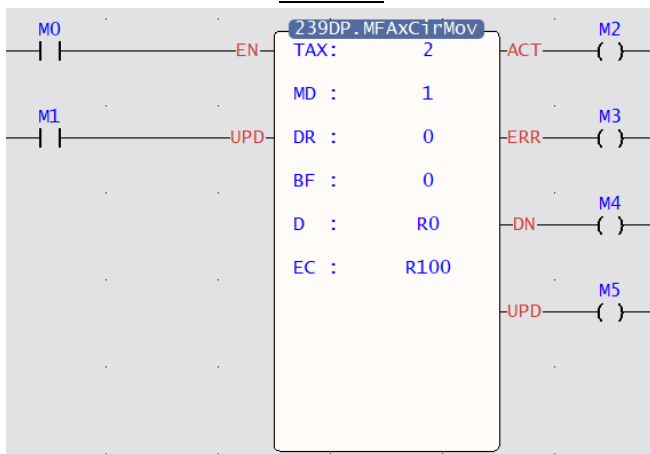
運動資料

暫存器	項目	資料長度	注意
R+0	主軸	WORD	
R+1	軸 2	WORD	
R+2	軸 3	WORD	
R+3	主軸目標位置	WORD	帶正負號包含主軸小數點位
R+5	軸 2 目標位置	DWORD	帶正負號包含軸 2 小數點位
R+7	軸 3 目標位置	DWORD	帶正負號包含軸 3 小數點位
R+9	最高速度	DWORD	正數包含主軸小數點位
R+11	加速時間 (ms)	DWORD	從 0 加速至最高速所需時間

R+13	減速時間 (ms)	DWORD	從最高速減速至 0 所需時間
R+15	加速 S 曲線 (‰)	WORD	正整數範圍 0~1000 ‰
R+16	減速 S 曲線 (‰)	WORD	正整數範圍 0~1000 ‰
R+17	圓弧模式	WORD	0: 通過點, 1: 圓心 順時針, 2: 圓心 逆時針, 3: 半徑 順時針, 4: 半徑 逆時針
R+18	X 座標 / 半徑	DWORD	帶正負號包含主軸小數點位
R+20	Y 座標	DWORD	帶正負號包含主軸小數點位
R+22	Z 座標	DWORD	帶正負號包含主軸小數點位

程式範例

Ladder



軸參數設定

		1	2
基本設定	軸名稱	軸_1	軸_2
	軸類型	伺服	伺服
	編碼器類型	增量型	增量型
單位設定	單位	mm	mm
	小數點位置	0.001	0.001
	脈波/圈	131072 PLS/Rev	131072 PLS/Rev
	單位/圈	1.000 mm/Rev	1.000 mm/Rev
	速度單位	指令位置/秒	指令位置/秒
	速度增益	1.000	1.000

- M0 由 OFF→ON 時，依照當前 Fun239 參數 TAX：圓弧補間運動所需軸數 2、MD：絕對位置、DR：正方向、BF：立即執行當前指令、D：資料暫存器起始位置 R0、EC：存放錯誤碼之暫存器 R100)執行直線補間運動。

連續路徑運動 MFPPathMov

Fun240 DP MFPPathMov		【連續路徑運動】										Fun240 DP MFPPathMov																																																																																																																																																																																																															
指令說明																																																																																																																																																																																																																											
階梯圖符號																																																																																																																																																																																																																											
運算元																																																																																																																																																																																																																											
移動到起始點—EN 開始動作—ACT 更新參數—UPD 240P. MFPPathMov AX1 : AX2 : AX3 : PT : V : TA : TD : SA : SD : EC : —ACT—動作中 —ERR—錯誤 —DN—執行完成 —UPD—更新完成																																																																																																																																																																																																																											
AX1 : 軸 1 編號 AX2 : 軸 2 編號 AX3 : 軸 3 編號 PT : 路徑編號 V : 速度 TA : 加速度 TD : 減速度 SA : S 加速度曲線% SD : S 減速度曲線% EC : 錯誤碼																																																																																																																																																																																																																											
繼電器和暫存器																																																																																																																																																																																																																											
<table><tr><th rowspan="3">運算元</th><th>範圍</th><th>WX</th><th>WY</th><th>WM</th><th>WS</th><th>TMR</th><th>CTR</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th></th><th>WX0</th><th>WY0</th><th>WM0</th><th>WS0</th><th>T0</th><th>C0</th><th>R0</th><th>R34768</th><th>R35024</th><th>R35280</th><th>R43224</th><th>D0</th><th></th><th>V, Z</th></tr><tr><th></th><th>WX1008</th><th>WY1008</th><th>WM9104</th><th>WS3088</th><th>T1023</th><th>C1279</th><th>R34767</th><th>R34895</th><th>R35151</th><th>R43223</th><th>R47319</th><th>D7999</th><th></th><th>P0 ~ P9</th></tr><tr><td>AX1</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~16</td><td></td></tr><tr><td>AX2</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~16</td><td></td></tr><tr><td>AX3</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~16</td><td></td></tr><tr><td>PT</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>0~10</td><td></td></tr><tr><td>V</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td>TA</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td>TD</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td>SA</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td>SD</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr><tr><td>EC</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td></td><td></td></tr></table>														運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR	K	XR		WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0		V, Z		WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		P0 ~ P9	AX1		○	○	○	○	○	○	○	○	○	○	○	○	0~16		AX2		○	○	○	○	○	○	○	○	○	○	○	○	0~16		AX3		○	○	○	○	○	○	○	○	○	○	○	○	0~16		PT		○	○	○	○	○	○	○	○	○	○	○	○	0~10		V		○	○	○	○	○	○	○	○	○	○	○	○			TA		○	○	○	○	○	○	○	○	○	○	○	○			TD		○	○	○	○	○	○	○	○	○	○	○	○			SA		○	○	○	○	○	○	○	○	○	○	○	○			SD		○	○	○	○	○	○	○	○	○	○	○	○			EC		○	○	○	○	○	○	○	○	○	○	○	○		
運算元	範圍	WX	WY	WM	WS	TMR	CTR	HR	IR	OR	SR	ROR	DR		K	XR																																																																																																																																																																																																											
		WX0	WY0	WM0	WS0	T0	C0	R0	R34768	R35024	R35280	R43224	D0			V, Z																																																																																																																																																																																																											
		WX1008	WY1008	WM9104	WS3088	T1023	C1279	R34767	R34895	R35151	R43223	R47319	D7999		P0 ~ P9																																																																																																																																																																																																												
AX1		○	○	○	○	○	○	○	○	○	○	○	○	0~16																																																																																																																																																																																																													
AX2		○	○	○	○	○	○	○	○	○	○	○	○	0~16																																																																																																																																																																																																													
AX3		○	○	○	○	○	○	○	○	○	○	○	○	0~16																																																																																																																																																																																																													
PT		○	○	○	○	○	○	○	○	○	○	○	○	0~10																																																																																																																																																																																																													
V		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																														
TA		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																														
TD		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																														
SA		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																														
SD		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																														
EC		○	○	○	○	○	○	○	○	○	○	○	○																																																																																																																																																																																																														
功能描述		10																																																																																																																																																																																																																									
● 運算元																																																																																																																																																																																																																											
AX1 軸 1 編號：EtherCAT_軸編號 0~16																																																																																																																																																																																																																											
AX2 軸 2 編號：EtherCAT_軸編號 0~16																																																																																																																																																																																																																											
AX3 軸 3 編號：EtherCAT_軸編號 0~16																																																																																																																																																																																																																											

PT 路徑編號：欲使用路徑之編號 0~10

V 速度：正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】

TA 加速度：正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】

TD 減速度：正數 (大於零的實數)，包含【運動控制】中【運動軸設定】的【小數點位置】

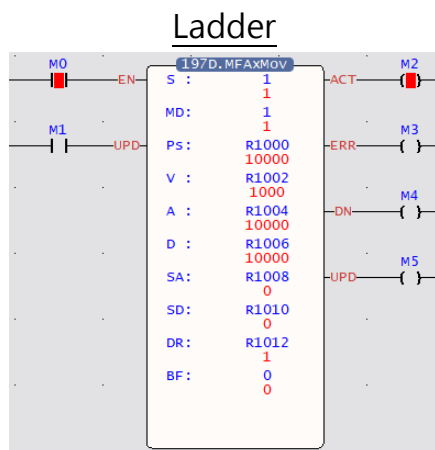
SA S 加速度曲線%：正整數，0~1000 %。

SD S 減速度曲線%：正整數，0~1000 %。

EC 錯誤碼：存放錯誤碼暫存器

- 當執行控制【EN】由上微分觸發，Fun240 執行連續路徑運動
當執行控制【EN】由下微分觸發，Fun240 停止連續路徑運動，並復位所有輸出指示
- 在連續路徑運動中，若更新參數【UPD】變為 1，則本指令將立即更新位置控制參數 (S、PS、V、A、D、SA、SD、DR)。
- 當軸位置控制中，輸出指示【ACT】ON。
- 當軸位置控制中，如有錯誤發生，則輸出指示【ERR】ON。
- 當軸位置控制完成，則輸出指示【DN】ON。
- 當更新位置控制參數完成，輸出指示【UPD】ON。

程式範例



軸參數設定

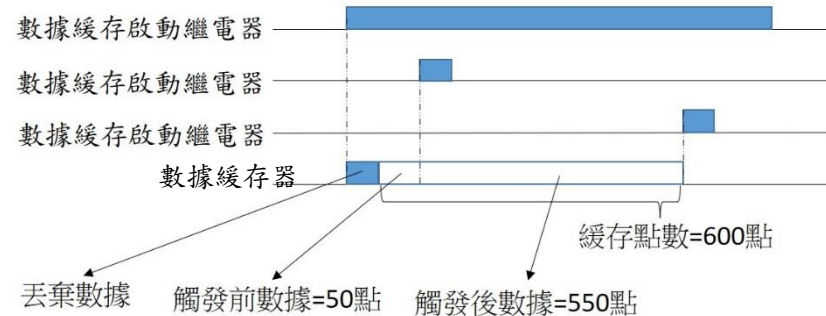
		1	2
基本設定	軸名稱	軸_1	
	軸類型	伺服	
	編碼器類型	增量型	
單位設定	單位	mm	
	小數點位置	0.001	
	脈波/圈	131072 PLS/Rev	131072 P
	單位/圈	1.000 mm/Rev	1.000 m
	速度單位	指令位置/秒	指令
	速度增益	1.000	

- M1000 由 OFF→ON 時，依照當前 Fun197 參數 (S：EtherCAT 軸 1、MD：相對位置、PS：移動到10.000mm、V：速度1.000mm/s、A：加速度100.000mm/s²、D：減速度100.000mm/s²、SA：S 加速度曲線0.0%、SD：S 減速度曲線0.0%、DR：正方向、BF：立即執行當前指令)執行位置控制。

7-23 其他指令 (FUN115 、 FUN258)

7-23-1 DBUF 數據緩存

FUN115P DBUF	數據緩存	FUN115P DBUF																																												
指令說明																																														
執行控制 — EN — 115P.DBUF ID: CH: D : — DN — 數據緩存完畢		ID : 擴充模組 ID 編號 CH : 擴充模組通道索引 (0 ~ 3) D : 存放緩存數據起始暫存器																																												
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V、Z P0~P9</td></tr><tr><th>ID</th><td></td><td></td><td></td><td></td><td></td><td></td><td>0~127</td><td>○</td></tr><tr><th>CH</th><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>0~63</td><td>○</td></tr><tr><th>D</th><td>○</td><td></td><td></td><td></td><td>○</td><td>○</td><td></td><td></td></tr></table>			運算元 \ 範圍	HR	IR	OR	SR	ROR	DR	K	XR	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0~P9	ID							0~127	○	CH	○	○	○	○	○*	○*	0~63	○	D	○				○	○		
運算元 \ 範圍	HR	IR		OR	SR	ROR	DR	K	XR																																					
	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0~P9																																						
ID							0~127	○																																						
CH	○	○	○	○	○*	○*	0~63	○																																						
D	○				○	○																																								
功能敘述																																														
用於獲取在模組上緩存完成的數據資料，適用於類比輸入且支援數據緩存功能之擴充模組。透過此指令收集到的緩存數據，不會受到程式掃描週期的限制，而是基於模組設定的採樣週期收集。																																														

FUN115P DBUF	數據緩存	FUN115P DBUF						
可透過繼電器控制數據緩存功能，儲存數位運算值至數據緩存區，用以觀察數位運算值變化。 使用方法與說明 每一個緩存點根據 A/D 轉換模式的處理時間將數位運算值更新至數據緩存區。 每個通道可以儲存最多 600 points/ch。 範例: 將緩存點數設定為 600，觸發前資料點數設定為 50 時  圖 137 數據緩存功能的範例圖 設定與預設值 <table><tr><th>設置</th><th>預設值</th></tr><tr><td>觸發前緩存點數</td><td>200</td></tr><tr><td>緩存點數</td><td>600</td></tr></table> 表 65 數據緩存功能的設置			設置	預設值	觸發前緩存點數	200	緩存點數	600
設置	預設值							
觸發前緩存點數	200							
緩存點數	600							

FUN115P DBUF	數據緩存	FUN115P DBUF												
下表為使用數據緩存功能的使用方法 Run-time 繼電器控制 <table> <tr> <th>數據緩存繼電器</th><th>敘述</th><th>設定</th></tr> <tr> <td>數據緩存請求繼電器</td><td>緩存請求</td><td>Off->On: 開始緩存。 On->Off: 中止緩存。</td></tr> <tr> <td>數據緩存觸發繼電器</td><td>觸發</td><td>Off->On: 觸發數據緩存繼電器。</td></tr> <tr> <td>數據緩存完成狀態繼電器</td><td>緩存完成 資料</td><td>Off->On: 指定緩存點數完成，可透過指令 115(DBUF 功能)來讀取緩存。 On->Off: 數據緩存請求繼電器: On -> Off，關閉緩存時則 Off。 數據緩存觸發繼電器: On->Off->On 重新觸發時則 Off，直到緩存點數完成後 Off->On。</td></tr> </table> 表 66 數據緩存功能的使用步驟 當數據緩存完成之後，再利用 Fun115 DBUF 將存在模組中的緩存數據讀取至 PLC 指定暫存器位址中。			數據緩存繼電器	敘述	設定	數據緩存請求繼電器	緩存請求	Off->On: 開始緩存。 On->Off: 中止緩存。	數據緩存觸發繼電器	觸發	Off->On: 觸發數據緩存繼電器。	數據緩存完成狀態繼電器	緩存完成 資料	Off->On: 指定緩存點數完成，可透過指令 115(DBUF 功能)來讀取緩存。 On->Off: 數據緩存請求繼電器: On -> Off，關閉緩存時則 Off。 數據緩存觸發繼電器: On->Off->On 重新觸發時則 Off，直到緩存點數完成後 Off->On。
數據緩存繼電器	敘述	設定												
數據緩存請求繼電器	緩存請求	Off->On: 開始緩存。 On->Off: 中止緩存。												
數據緩存觸發繼電器	觸發	Off->On: 觸發數據緩存繼電器。												
數據緩存完成狀態繼電器	緩存完成 資料	Off->On: 指定緩存點數完成，可透過指令 115(DBUF 功能)來讀取緩存。 On->Off: 數據緩存請求繼電器: On -> Off，關閉緩存時則 Off。 數據緩存觸發繼電器: On->Off->On 重新觸發時則 Off，直到緩存點數完成後 Off->On。												

7-23-2 皮重扣除指令

FUN258P MODCONF	皮重扣除指令	FUN258P MODCONF																																																																				
指令說明																																																																						
階段圖符號 258DP.MODCONF 運算控制 — EN 回復初始 — RST OP TARESUB ID CH SB ERR — OP: TARESUB ID:擴充模組 ID 編號 CH:通道索引(0 ~ 3) SB: 為欲扣除之皮重 32BIT 數值																																																																						
<table><tr><th rowspan="3">運算元</th><th>範圍</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>R0</td><td>R34768</td><td>R35024</td><td>R35280</td><td>R43224</td><td>D0</td><td></td><td>V、Z</td></tr><tr><td></td><td>R34767</td><td>R34895</td><td>R35151</td><td>R43223</td><td>R47319</td><td>D7999</td><td></td><td>P0-P9</td></tr><tr><td>OP</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>TARESUB</td><td>○</td></tr><tr><td>ID</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>0~63</td><td>○</td></tr><tr><td>CH</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>SB</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>			運算元	範圍	HR	IR	OR	SR	ROR	DR	K	XR		R0	R34768	R35024	R35280	R43224	D0		V、Z		R34767	R34895	R35151	R43223	R47319	D7999		P0-P9	OP								TARESUB	○	ID		○	○	○	○	○*	○*	0~63	○	CH		○	○	○	○	○	○	○	○	SB		○	○	○	○	○	○	○	○
運算元	範圍	HR		IR	OR	SR	ROR	DR	K	XR																																																												
		R0		R34768	R35024	R35280	R43224	D0		V、Z																																																												
		R34767	R34895	R35151	R43223	R47319	D7999		P0-P9																																																													
OP								TARESUB	○																																																													
ID		○	○	○	○	○*	○*	0~63	○																																																													
CH		○	○	○	○	○	○	○	○																																																													
SB		○	○	○	○	○	○	○	○																																																													
功能敘述	● 若要減去自訂皮重，需將 config 設定改為“ 數位模式” 。“ 輕觸模式” 時，會將當前毛重視為皮重，並直接扣除當前毛重。 ● 移除固定的皮重重量，並重新校正，可能得到精準度提升的好處。 ● 啟用皮重去除的指令時，若指令模式設定為“ 輕觸” ，為自動設定參數模式，減去當前的秤重讀值。 ● 啟用皮重去除的指令時，若指令模式設定為“ 數位” ，為手動設定參數模式，此時使用者可自行設定要扣除的皮重，當發送啟用皮重去除的指令時，指令會依照使用者設定的參數，減去相應重量。 ● RST OFF->ON 時，會恢復控制前設定																																																																					

7-23-3 皮重偏移指令

FUN258P MODCONF	皮重偏移指令		FUN258P MODCONF																																																					
指令說明																																																								
階段圖符號 258DP.MODCONF 运算控制 — EN — 手動模式 — MD — OP TAREZROFFSET ID CH WR ERR — DN — OP: TAREZROFFSET ID:擴充模組 ID 編號(0 ~ N) CH:通道索引(0 ~ N) WR: 偏移參數設定起始暫存器																																																								
<table><tr><th rowspan="2">運算元 範圍</th><th>HR</th><th>IR</th><th>OR</th><th>SR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>R0 R34767</td><td>R34768 R34895</td><td>R35024 R35151</td><td>R35280 R43223</td><td>R43224 R47319</td><td>D0 D7999</td><td></td><td>V、Z P0-P9</td></tr><tr><td>OP</td><td></td><td></td><td></td><td></td><td></td><td></td><td>TAREZROFFSET</td><td>○</td></tr><tr><td>ID</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○*</td><td>○*</td><td>0~63</td><td>○</td></tr><tr><td>CH</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>SB</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>				運算元 範圍	HR	IR	OR	SR	ROR	DR	K	XR	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0-P9	OP							TAREZROFFSET	○	ID	○	○	○	○	○*	○*	0~63	○	CH	○	○	○	○	○	○	○	○	SB	○	○	○	○	○	○	○	○
運算元 範圍	HR	IR	OR		SR	ROR	DR	K	XR																																															
	R0 R34767	R34768 R34895	R35024 R35151	R35280 R43223	R43224 R47319	D0 D7999		V、Z P0-P9																																																
OP							TAREZROFFSET	○																																																
ID	○	○	○	○	○*	○*	0~63	○																																																
CH	○	○	○	○	○	○	○	○																																																
SB	○	○	○	○	○	○	○	○																																																
功能敘述																																																								
WR 表格: <table><tr><td>WR+0</td><td>INA 增益</td></tr><tr><td>WR+1</td><td>ADC 增益</td></tr><tr><td>WR+2</td><td rowspan="2">數位值,為 32 位元</td></tr><tr><td>WR+3</td></tr></table>				WR+0	INA 增益	WR+1	ADC 增益	WR+2	數位值,為 32 位元	WR+3																																														
WR+0	INA 增益																																																							
WR+1	ADC 增益																																																							
WR+2	數位值,為 32 位元																																																							
WR+3																																																								

FUN258P MODCONF	皮重偏移指令	FUN258P MODCONF
功能敘述	● 移除固定的皮重重量，透過設定 Instrumentation amplifier gain 和 ADC gain 可以得到轉換精準度提升的好處。 ● 自動設定參數模式，MD=0，發送啟用 tare zero function 的指令，模組會自動計算適合的 Instrumentation amplifier gain、ADC gain 和 Digital value 後並回傳給 PLC。 ● 使用者手動設定模式，MD=1，發送啟用 tare zero function 的指令，指令會附帶使用者設定的 Instrumentation amplifier gain、ADC gain 和 Digital value。Instrumentation amplifier gain 為 433.92, 216.96 和 108.48，ADC gain 為 1,2,4,8 和 16，Digital value 設定範圍為 1~ 56874 (2.1696 V)。 ● 計算固定重量(Fixed weight)的公式， $\text{Fixed tare weight} = \left(\frac{\text{DAC digital value}}{65535} \times 2.5 \right) \times \frac{\text{Rated capacity} \times \text{Number of LC sensors}}{\text{INA gain} \times \text{Excitation voltage} \times \text{Rated output}}$ ● ADC/INA Gain 設定建議公式， $\frac{\text{Max weighting capacity}}{\text{Rated capacity} \times \text{Number of LC sensors}} \times \text{INA gain} \times \text{ADC gain} \leq 500 = \left(\frac{\text{DAC digital value}}{65535} \times 2.5 \right) \times \frac{\text{Rated capacity} \times \text{Number of LC sensors}}{\text{INA gain} \times \text{Excitation voltage} \times \text{Rated output}}$ *皮重偏移指令僅 LCR 模組支援，LC 不支援	

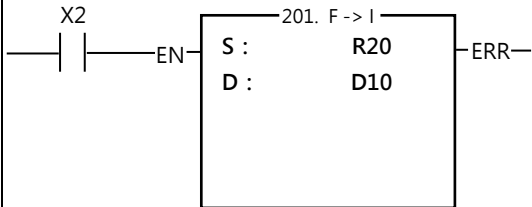
7-24 浮點運算指令 (FUN200 ~ 220)

7-24-1 整數轉換浮點數(CONVERSION OF INTEGER TO FLOATING POINT NUMBER)

FUN200DP I→F	整數轉換浮點數 (CONVERSION OF INTEGER TO FLOATING POINT NUMBER)	FUN200DP I→F																							
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																								
階梯圖符號 200DP.I→F 轉換控制 — EN S： D： S：來源暫存器之起始號碼 D：存放結果(浮點數)之暫存器起始號碼 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																									
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>16/32 位元 正、負數</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	16/32 位元 正、負數	V、Z P0~P9	S	○	○	○	○	○	D	○	○*	○*		○
運算元 \ 範圍	HR	ROR		DR	K	XR																			
	R0 R34767	R43224 R47319	D0 D7999	16/32 位元 正、負數	V、Z P0~P9																				
S	○	○	○	○	○																				
D	○	○*	○*		○																				
功能敘述																									
● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 ● 當執行控制“EN” =1 或由 0→1 (P 指令) 時，將 S 暫存器內之整數數值資料，轉換成浮點數格式資料之後，存放於 D 暫存器中。																									

7-24-2 浮點數轉換整數(CONVERSION OF FLOATING POINT NUMBER TO INTEGER)

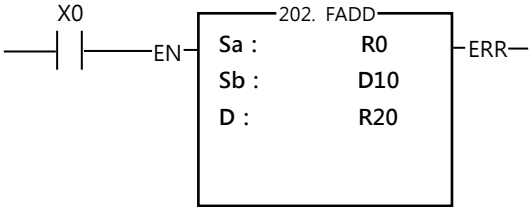
FUN201D F→I		浮點數轉換整數 (CONVERSION OF FLOATING POINT NUMBER TO INTEGER)				FUN201D F→I																				
指令說明		※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																								
		階梯圖符號 201DP.F→I 轉換控制 — EN — S : — ERR — 超出範圍 D :				S：來源暫存器之起始號碼 D：存放結果(整數)之暫存器起始號碼 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																				
		<table><tr><th rowspan="3">運算元 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>XR</th></tr><tr><td>R0 R347 67</td><td>R43224 R47319</td><td>D0 D11998</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td>○</td></tr></table>						運算元 範圍	HR	ROR	DR	XR	R0 R347 67	R43224 R47319	D0 D11998	V、Z P0~P9	S	○	○	○	○	D	○	○*	○*	○
運算元 範圍	HR	ROR	DR	XR																						
	R0 R347 67	R43224 R47319	D0 D11998	V、Z P0~P9																						
	S	○	○	○	○																					
D	○	○*	○*	○																						
功能敘述																										
		● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P.118。 ● 當執行控制“EN”=1 或由 0→1 (P 指令) 時，將 S 暫存器內之浮點數數值資料，轉換成整數格式資料之後，存放於 D 暫存器中。 ● 若 D 暫存器(目的暫存器)存放之轉換結果，超出有效範圍，則錯誤旗號“ERR”設為 1，且本指令不執行，而 D 暫存器之內容維持不變。																								
程式範例																										

FUN201D F→I	浮點數轉換整數 (CONVERSION OF FLOATING POINT NUMBER TO INTEGER)	FUN201D F→I																																																																																																																																
階段圖		ST																																																																																																																																
		先在標籤產生對應的 Tag -> IF X2 Then FtoI(S:= Tag_FlR20, D=> D10); END_IF																																																																																																																																
● DR20 = 123.45 → X2= ● (經整數轉換之後) → D10 = 007BH <table border="1" data-bbox="165 1023 1399 1133"><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>s</td><td>e</td><td>e</td><td>e</td><td>e</td><td>e</td><td>e</td><td>e</td><td>e</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td><td>m</td></tr></table> F → I <table border="1" data-bbox="178 1245 817 1319"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr></table>			0	1	0	0	0	0	1	0	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	1	0	0	0	0	1	0	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0																																																																																																			
b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																			
s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m																																																																																																			
0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1																																																																																																																			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																			

7-24-3 浮點數加法運算(FLOATING POINT NUMBER ADDITION)

FUN202 FADD	浮點數加法運算 (FLOATING POINT NUMBER ADDITION)	FUN202 FADD																														
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																															
階段圖符號 加算控制 — EN — 202P.FADD Sa : Sb : D : — ERR — 範圍錯誤 (FO0) Sa：被加數或其暫存器號碼。 Sb：加數或其暫存器號碼。 D：存放結果(和)之暫存器起始號碼 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																																
<table><tr><td rowspan="5">運算元</td><td rowspan="2">範圍</td><td>HR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>浮點數</td><td>V、Z P0~P9</td></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元	範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	Sa	○	○	○	○	○	Sb	○	○	○	○	○	D	○	○*	○*		○
運算元	範圍	HR			ROR	DR	K	XR																								
		R0 R34767		R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																									
	Sa	○		○	○	○	○																									
	Sb	○		○	○	○	○																									
	D	○	○*	○*		○																										
功能敘述																																
● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 ● 當加算控制“EN”=1 或由 0→1 (P 指令) 時，將 Sa 與 Sb 作浮點數加法運算並將結果寫入 D 去。假若執行結果超出浮點數可表示之範圍(+ -3.4*10 38)"，則錯誤旗號"ERR"設為 1，且 D 暫存器之值為無效之值，應忽視不計。																																
程式範例																																

FUN202 P FADD	浮點數加法運算 (FLOATING POINT NUMBER ADDITION)	FUN202 P FADD
-------------------------	---	-------------------------

階梯圖	ST
	先在標籤產生對應的 Tag -> IF X0 Then Tag_FlR20 := Tag_FlD10 + Tag_FlR0 ; END_IF

- 當 X0=ON，將 Sa 與 Sb 作浮點數加法運算：

DR0	200	⇒ 浮點數:	DR0	43480000H
DR10	150	⇒ 浮點數:	DR10	43160000H
+				
—————				
			DR20	43AF0000H

7-24-4 浮點數減法運算(FLOATING POINT NUMBER SUBTRACTION)

FUN203 P FSUB	浮點數減法運算 (FLOATING POINT NUMBER SUBTRACTION)	FUN203 P FSUB																													
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																														
階梯圖符號 減算控制 — EN — 203P.FSUB Sa : Sb : D : — ERR — 範圍錯誤 (FO0)		Sa：被減數或其暫存器號碼。 Sb：減數或其暫存器號碼。 D：存放結果(差)之暫存器起始號碼 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																													
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th>R0 R34767</th><th>R43224 R47319</th><th>D0 D7999</th><th>浮點數</th><th>V、Z P0~P9</th></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	Sa	○	○	○	○	○	Sb	○	○	○	○	○	D	○	○*	○*		○
運算元 \ 範圍	HR	ROR		DR	K	XR																									
	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																										
Sa	○	○	○	○	○																										
Sb	○	○	○	○	○																										
D	○	○*	○*		○																										
功能敘述	● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 ● 當加算控制“EN” =1 或由 0→1 (P 指令) 時，將 Sa 與 Sb 作浮點數減法運算並將結果寫入 D 去。假若執行結果超出浮點數可表示之範圍(+ -3.4*10 38)"，則錯誤旗號"ERR"設為 1，且 D 暫存器之值為無效之值，應忽視不計。																														
程式範例																															

FUN203 P FSUB	浮點數減法運算 (FLOATING POINT NUMBER SUBTRACTION)	FUN203 P FSUB
階段圖 X0 — — EN 203. FSUB Sa : R0 Sb : R4 D : R10 —ERR— 先在標籤產生對應的 Tag -> IF X0 Then Tag_FlR10 := Tag_FlR0 - Tag_FlR4 ; END_IF		
● 當 X0=ON，將 Sa 與 Sb 作浮點數減法運算： DR0 2 0 0 ⇒ 浮點數： DR0 4 3 4 8 0 0 0 0 H DR4 5 0 0 ⇒ 浮點數： DR4 4 3 F A 0 0 0 0 H DR10 C 3 9 6 0 0 0 0 H		

7-24-5 浮點數乘法運算(FLOATING POINT NUMBER MULTIPLICATION)

FUN204 P FMUL	浮點數乘法運算 (FLOATING POINT NUMBER MULTIPLICATION)	FUN204 P FMUL																													
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																														
階段圖符號 乘算控制 — EN — 204P.FMUL Sa : Sb : D : — ERR — 範圍錯誤 (FO0)		Sa：被乘數或其暫存器號碼。 Sb：乘數或其暫存器號碼。 D：存放結果(積)之暫存器起始號碼 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																													
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th>R0 R34767</th><th>R43224 R47319</th><th>D0 D7999</th><th>浮點數</th><th>V、Z P0~P9</th></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	Sa	○	○	○	○	○	Sb	○	○	○	○	○	D	○	○*	○*		○
運算元 \ 範圍	HR	ROR		DR	K	XR																									
	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																										
Sa	○	○	○	○	○																										
Sb	○	○	○	○	○																										
D	○	○*	○*		○																										
功能敘述																															
● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 ● 當加算控制“EN” =1 或由 0→1 (P 指令) 時，將 Sa 與 Sb 作浮點數乘法運算並將結果寫入 D 去。假若執行結果超出浮點數可表示之範圍(+ -3.4*10 38)"，則錯誤旗號"ERR"設為 1，且 D 暫存器之值為無效之值，應忽視不計。																															
程式範例																															

FUN204 P FMUL	浮點數乘法運算 (FLOATING POINT NUMBER MULTIPLICATION)	FUN204 P FMUL
階段圖 M10 — — EN 204. FMUL Sa : R10 Sb : R12 D : R14 ERR—		ST 先在標籤產生對應的 Tag -> IF M10 Then Tag_FlR14 := Tag_FlR10 * Tag_FlR12 ; END_IF 或 IF M10 Then FMUL(Sa:= Tag_FlR10, Sb:= Tag_FlR12, D:= Tag_FlR14); END_IF
當 M10= 1 時，將 Sa 與 Sb 作浮點數乘法運算： DR10 1 2 3 . 4 5 ⇒ 浮點數： DR10 4 2 F 6 E 6 6 6 H DR12 6 7 8 . 5 4 ⇒ 浮點數： DR12 4 4 2 9 A 2 8 F H × DR14 4 7 A 3 9 A E 2 H		

7-24-6 浮點數除法運算(FLOATING POINT NUMBER DIVISION)

FUN205 P FDIV	浮點數除法運算 (FLOATING POINT NUMBER DIVISION)						FUN205 P FDIV
指令說明		※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。					
階段圖符號 除算控制 — EN — 205P.FDIV Sa : Sb : D : — ERR — 範圍錯誤 (FOO)				Sa：被除數或其暫存器號碼。 Sb：除數或其暫存器號碼。 D：存放結果(商)之暫存器起始號碼 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用			
		範圍 運算	HR R0 ↓ R34 767	ROR R43 224 ↓ R47 319	DR D0 ↓ D119 98	K 浮點數	XR V、 Z P0~
功能敘述							
● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 ● 當加算控制“EN” =1 或由 0→1 (P 指令) 時，將 Sa 與 Sb 作浮點數除法運算並將結果寫入 D 去。假若執行結果超出浮點數可表示之範圍(+ -3.4*10 38)"，則錯誤旗號"ERR"設為 1，且 D 暫存器之值為無效之值，應忽視不計。							
程式範例							

FUN205 P FDIV	浮點數除法運算 (FLOATING POINT NUMBER DIVISION)	FUN205 P FDIV
階段圖 X5 — — EN 205. FDIV Sa : R0 Sb : R2 D : R4 —ERR— 先在標籤產生對應的 Tag -> IF X5 Then Tag_FlR4 := Tag_FlR0 / Tag_FlR2 ; END_IF 或 IF X5 Then FDIV(Sa:= Tag_FlR0, Sb:= Tag_FlR2, D:= Tag_FlR4); END_IF		
● 當 X5=ON · 將 Sa 與 Sb 作浮點數除法運算: DR0 125.25 ⇒ 浮點數: DR0 42FA8000H DR2 5 ⇒ 浮點數: DR2 40A00000H ÷ DR4 41C86666H		

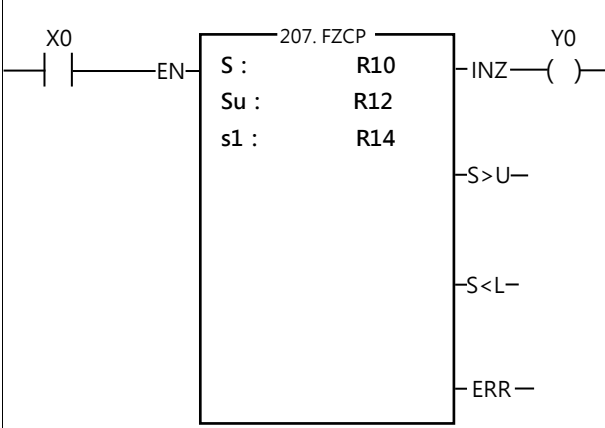
7-24-7 浮點數比較運算(FLOATING POINT NUMBER COMPARE)

FUN206 P FCMP	浮點數比較運算 (FLOATING POINT NUMBER COMPARE)	FUN206 P FCMP																								
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																									
階段圖符號 比較控制 — EN — 206P.FCMP Sa： a = b — Sa等於Sb (FO0) Sb： a > b — Sa大於Sb (FO1) a < b — Sa小於Sb (FO2)		Sa：比較值 a 或其暫存器號碼。 Sb：比較值 b 或其暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																								
<table><tr><th>運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td></td><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>浮點數</td><td>V、Z P0~P9</td></tr><tr><td>Sa</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>Sb</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	XR		R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	Sa	○	○	○	○	○	Sb	○	○	○	○	○
運算元 \ 範圍	HR	ROR	DR	K	XR																					
	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																					
Sa	○	○	○	○	○																					
Sb	○	○	○	○	○																					
功能敘述																										
- M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 - 當比較控制“EN” =1 或由 0→1 (P 指令) 時，將 Sa 與 Sb 作浮點數比較運算，若 Sa=Sb 則 FO0 設為 1，若 Sa > Sb 則 FO1 設為 1，若 Sa < Sb 則 FO2 設為 1。																										
程式範例																										

FUN206 FCMP	浮點數比較運算 (FLOATING POINT NUMBER COMPARE)	FUN206 FCMP
階段圖 X0 EN 206. FCMP Sa : Sb : R0 R2 a=b a>b a<b () Y0 ST DR0 200.1 ⇒ 浮點數： DR0 4348199AH DR2 200.2 ⇒ 浮點數： DR2 43483333H ● 上例假若 DR0 之值為 200.1，DR2 之值為 200.2，則當 X0=ON時，CMP 指令執行比較工作，並得出 a<b 之結果，故會將 FO0 及 FO1 設為 0，FO2 (a<b) 設為 1。 ● 若您需要複合結果，如≥、≤、<>等，請先將=、>、<等結果送到繼電器再由繼電器取出 OR 起來即可。		

7-24-8 浮點數區域比較運算(FLOATING POINT NUMBER ZONE COMPARE)

FUN207 P FZCP	浮點數區域比較運算 (FLOATING POINT NUMBER ZONE COMPARE)				FUN207 P FZCP																														
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																																		
階段圖符號 比較控制 — EN — 207P.FZCP S : Su : SL : INZ — 在區域內 S>U — 高於上限 S<L — 低於下限 ERR — 限值錯誤			S：存放比較資料(浮點數)之暫存器號碼。 SU：區域上限值或上限值暫存器號碼。 SL：區域下限值或下限值暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																																
			<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <th style="width: 10%;">範圍</th><th style="width: 10%;">HR</th><th style="width: 10%;">ROR</th><th style="width: 10%;">DR</th><th style="width: 10%;">K</th><th style="width: 10%;">XR</th></tr> <tr> <td></td><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>浮點數</td><td>V、Z P0~P9</td></tr> <tr> <td>S</td><td>○</td><td>○</td><td>○</td><td></td><td>○</td></tr> <tr> <td>Su</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr> <tr> <td>SL</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr> </table>			範圍	HR	ROR	DR	K	XR		R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	S	○	○	○		○	Su	○	○	○	○	○	SL	○	○	○	○	○
範圍	HR	ROR	DR	K	XR																														
	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																														
S	○	○	○		○																														
Su	○	○	○	○	○																														
SL	○	○	○	○	○																														
功能敘述	● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P. 118。 ● 當比較控制“EN”=1 或由 0→1 (P 指令) 時，執行 S 與上限 SU 及下限 SL 之比較，若 S 介於上限值與下限值之間 ($SL \leq S \leq SU$)，則在區域內旗號“INZ”設為 1，若 S 之值大於上限 SU，則高於上限旗號“S>U”設為 1，若 S 之值小於下限 SL，則低於下限旗號“S<L”設為 1。 ● 上限 SU 應大於下限 SL，若 $SU < SL$，則限值錯誤旗號“ERR”設為 1，且本指令不執行。																																		
FUN207 P FZCP	浮點數區域比較運算 (FLOATING POINT NUMBER ZONE COMPARE)				FUN207 P FZCP																														
程式範例																																			

階梯圖	ST
	<pre> IF X0 THEN FZCP(S:= R10, Su:= R12, Sl:= R14, INZ=> Y0, S_Gt_U=> M0, S_Less_L=> M1, ERR=> M2); END_IF </pre>

- 上圖程式範例係將 DR10 之值和由 DR12 和 DR14 所構成之上、下限區域作比較，假設 DR10 ~ DR14 之數值如下圖左，則可獲得如下圖右之執行結果。
- 若輸出結果需要為不在區域內，則可用 OUT NOT Y0 即可。

S	DR10	2 0 0 0 . 2
Su	DR12	3 0 0 0 . 3
Sl	DR14	1 0 0 0 . 1

⇒ 浮點數：

⇒ 浮點數：

⇒ 浮點數：

DR10	4 4 F A 0 6 6 6 H
DR12	4 5 3 B 8 4 C D H
DR14	4 4 7 A 0 6 6 6 H

(上限值)

(下現值)

執行前

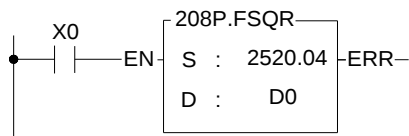
X0 = ☐ 浮點數區域比較 → Y0 = ☒

執行後結果

7-24-9 浮點數開根號運算(FLOATING POINT NUMBER SQUARE ROOT)

FUN208P FSQR	浮點數開根號運算 (FLOATING POINT NUMBER SQUARE ROOT)	FUN208P FSQR																							
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																								
階段圖符號 208P.FSQR 運算控制 — EN — S : D : ERR — S 範圍錯誤 S：求平方根之來源數值或暫存器號碼。 D：存放結果(平方根值)之暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																									
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>浮點數</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	S	○	○	○	○	○	D	○	○*	○*		○
運算元 \ 範圍	HR	ROR		DR	K	XR																			
	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																				
S	○	○	○	○	○																				
D	○	○*	○*		○																				
功能敘述																									
● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P.118。 ● 當運算控制“EN” =1 或由 0→1 (P 指令) 時，將 S 值或 S 所指定之暫存器內容值取平方根值後存入 D 所指定之暫存器內。 ● 當 S 值為暫存器內容值，而值為負數，則 S 值錯誤旗號“ERR”設為 1，且本指令不執行。																									

FUN208 P FSQR	浮點數開根號運算 (FLOATING POINT NUMBER SQUARE ROOT)	FUN208 P FSQR
程式範例		

階梯圖	ST
	先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FSQR(S:= 2520.04, D:=Tag_F1D0); END_IF

當 X0=ON，將 S 作浮點數開根號運算：

S :

K	2520.04
---	---------

↓ X0 = ↑

D :

D1	D0	50.2
----	----	------

 ⇒ 浮點數：

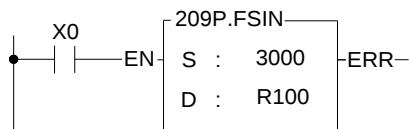
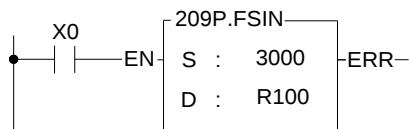
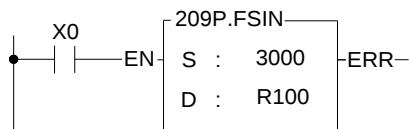
4	2	4	8	C	C	C	D
⏟ D1				⏟ D0			

H

$\sqrt{2520.04} = 50.2$

7-24-10 浮點數表示法取三角函數(SIN)運算

FUN209P FSIN	浮點數表示法取三角函數(SIN)運算 (SIN TRIGONOMETRIC INSTRUCTION)	FUN209P FSIN																											
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																												
階段圖符號 運算控制 — EN — 209P.FSIN — S : D : — ERR — S 範圍錯誤		S：求 SIN 值之來源數值或暫存器號碼。 D：存放結果之暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																											
<table><tr><th rowspan="2">運算元</th><th>範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><th></th><th>R0 R34767</th><th>R43224 R47319</th><th>D0 D7999</th><th>整數 16 位元</th><th>V、Z P0~P9</th></tr><tr><td>S</td><td></td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td></td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元	範圍	HR	ROR	DR	K	XR		R0 R34767	R43224 R47319	D0 D7999	整數 16 位元	V、Z P0~P9	S		○	○	○	○	○	D		○	○*	○*		○
運算元	範圍	HR		ROR	DR	K	XR																						
		R0 R34767	R43224 R47319	D0 D7999	整數 16 位元	V、Z P0~P9																							
S		○	○	○	○	○																							
D		○	○*	○*		○																							
功能敘述	- M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5.3 章(數目系統)-P.118。 - 當運算控制“EN” =1 或由 0→1 (P 指令) 時，將 S 值或 S 所指定之暫存器內容值取 SIN 函數後存入 D 所指定之暫存器內。S 之有效範圍為-18000~+18000，單位為 0.01 度。 - 若 S 值或 S 所指定之暫存器內容值超過其有效範圍(-18000~+18000)，則錯誤旗號“ERR”設為 1，且本指令不執行。																												

FUN209 P FSIN	浮點數表示法取三角函數(SIN)運算 (SIN TRIGONOMETRIC INSTRUCTION)	FUN209 P FSIN				
程式範例						
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FSIN(S:= 3000, D:= Tag_F1R100); END_IF</td></tr></table> ● 上圖範例，當 X0=ON，將 SIN∠30 之值存入 DR100 之中。 30 ↓ × 100 (偏移值) S 3000 X0 =  浮點數： DR100 3 F 0 0 0 0 0 H SIN(30) = 0.5			階梯圖	ST		先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FSIN(S:= 3000, D:= Tag_F1R100); END_IF
階梯圖	ST					
	先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FSIN(S:= 3000, D:= Tag_F1R100); END_IF					

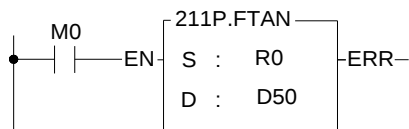
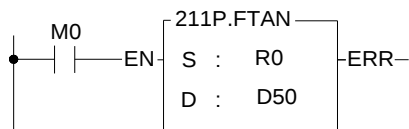
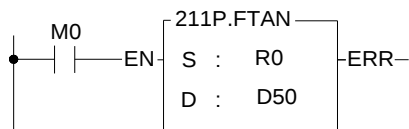
7-24-11 浮點數表示法取三角函數(COS)運算

FUN210P FCOS	浮點數表示法取三角函數(COS)運算 (COS TRIGONOMETRIC INSTRUCTION)	FUN210P FCOS																							
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																								
階段圖符號 運算控制 — EN — 210P.FCOS S : D : — ERR — S 範圍錯誤		S：求 COS 值之來源數值或暫存器號碼。 D：存放結果之暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																							
<table><tr><th rowspan="2">運算元 \ 範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>整數 16 位元</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr></table>			運算元 \ 範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	整數 16 位元	V、Z P0~P9	S	○	○	○	○	○	D	○	○*	○*		○
運算元 \ 範圍	HR	ROR		DR	K	XR																			
	R0 R34767	R43224 R47319	D0 D7999	整數 16 位元	V、Z P0~P9																				
S	○	○	○	○	○																				
D	○	○*	○*		○																				
功能敘述																									
● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5-3 章(數目系統) 5-9 頁。 ● 當運算控制“EN” =1 或由 0→1 (P 指令) 時，將 S 值或 S 所指定之暫存器內容值取 COS 函數後存入 D 所指定之暫存器內。S 之有效範圍為-18000 ~ +18000，單位為 0.01 度。 ● 若 S 值或 S 所指定之暫存器內容值超過其有效範圍(-18000 ~ +18000)，則錯誤旗號“ERR” 設為 1，且本指令不執行。																									

FUN210P FCOS	浮點數表示法取三角函數(COS)運算 (COS TRIGONOMETRIC INSTRUCTION)	FUN210P FCOS				
程式範例						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td>先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FCOS(S:= R0, D:= Tag_F1R200); END_IF</td></tr></table>			階段圖	ST		先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FCOS(S:= R0, D:= Tag_F1R200); END_IF
階段圖	ST					
	先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then FCOS(S:= R0, D:= Tag_F1R200); END_IF					
● 上圖範例，當 X0=ON，將 COS∠ 60 之值存入 DR200 之中。 60 ↓ × 100 (偏移值) R0 6000 X0 = 浮點數： DR200 3F000000H COS(60) = 0.5						

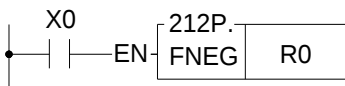
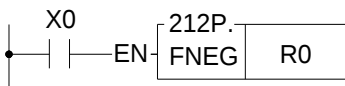
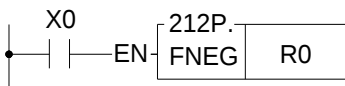
7-24-12 浮點數表示法取三角函數(TAN)運算

FUN211 FTAN	浮點數表示法取三角函數(TAN)運算 (TAN TRIGONOMETRIC INSTRUCTION)	FUN211 FTAN																					
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																						
階段圖符號 運算控制 — EN — 211P.FTAN S : D : — ERR — S 範圍錯誤		S：求 TAN 值之來源數值或暫存器號碼。 D：存放結果之暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用																					
<table><tr><td rowspan="4">運算元 範圍</td><td>HR</td><td>ROR</td><td>DR</td><td>K</td><td>XR</td></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>整數 16 位元</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td>○</td></tr></table>			運算元 範圍	HR	ROR	DR	K	XR	R0 R34767	R43224 R47319	D0 D7999	整數 16 位元	V、Z P0~P9	S	○	○	○	○	D	○	○*	○*	○
運算元 範圍	HR	ROR		DR	K	XR																	
	R0 R34767	R43224 R47319		D0 D7999	整數 16 位元	V、Z P0~P9																	
	S	○		○	○	○																	
	D	○	○*	○*	○																		
功能敘述	● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5-4 章(數目系統)。 ● 當運算控制“EN” =1 或由 0→1 (P 指令) 時，將 S 值或 S 所指定之暫存器內容值取 TAN 函數後存入 D 所指定之暫存器內。S 之有效範圍為-18000 ~ +18000，單位為 0.01 度。 ● 若 S 值或 S 所指定之暫存器內容值超過其有效範圍(-18000 ~ +18000)，則錯誤旗號“ERR”設為 1，且本指令不執行。																						

FUN211 P FTAN	浮點數表示法取三角函數(TAN)運算 (TAN TRIGONOMETRIC INSTRUCTION)	FUN211 P FTAN				
程式範例						
<table><tr><th>階段圖</th><th>ST</th></tr><tr><td></td><td>先在標籤產生對應的 Tag -> IF R_TRIG(S:= M0) Then FTAN(S:= R0, D:= Tag_F1D50); END_IF</td></tr></table> ● 上圖範例，當 M0= 將 TAN∠45 之值存入 DD50 之中。 45 ↓ × 100 (偏移值) R0 4500 M0 =  浮點數： DD50 3F800000H TAN(45) = 1			階段圖	ST		先在標籤產生對應的 Tag -> IF R_TRIG(S:= M0) Then FTAN(S:= R0, D:= Tag_F1D50); END_IF
階段圖	ST					
	先在標籤產生對應的 Tag -> IF R_TRIG(S:= M0) Then FTAN(S:= R0, D:= Tag_F1D50); END_IF					

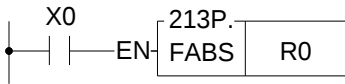
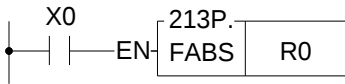
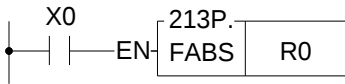
7-24-13 浮點數取負值運算 (CHANGE SIGN OF THE FLOATING POINT NUMBER)

FUN212P FNEG	浮點數取負值運算 (CHANGE SIGN OF THE FLOATING POINT NUMBER)	FUN212P FNEG														
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。															
階梯圖符號 運算控制 — EN — 212P. FNEG D		D：存放取負值運算結果之暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用														
<table><tr><td rowspan="3">運算元 範圍</td><td>HR</td><td>ROR</td><td>DR</td><td>XR</td></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>V、Z P0~P9</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td><td>○</td></tr></table>			運算元 範圍	HR	ROR	DR	XR	R0 R34767	R43224 R47319	D0 D7999	V、Z P0~P9	D	○	○	○	○
運算元 範圍	HR	ROR		DR	XR											
	R0 R34767	R43224 R47319		D0 D7999	V、Z P0~P9											
	D	○	○	○	○											
功能敘述																
M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5-4 章(數目系統)。 當運算控制“EN” =1 或由 0→1 (P 指令) 時，將 D 所指定之暫存器內容取其負數後存回原暫存器 D(浮點數)。 若 D 之內容值原為負數，取負數之結果將變為正數。																

FUN212P FNEG	浮點數取負值運算 (CHANGE SIGN OF THE FLOATING POINT NUMBER)	FUN212P FNEG																	
程式範例																			
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td>先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then Tag_FlR0 := - Tag_FlR0; END_IF</td></tr></table>			階梯圖	ST		先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then Tag_FlR0 := - Tag_FlR0; END_IF													
階梯圖	ST																		
	先在標籤產生對應的 Tag -> IF R_TRIG(S:= X0) Then Tag_FlR0 := - Tag_FlR0; END_IF																		
上圖範例，將 DR0 之內容值取負值運算之後，回存至 DR0 內(浮點數)。																			
<table><tr><td><table><tr><td>DR0</td><td>1 2 3 . 4 5</td></tr></table></td><td>⇒ 浮點數：</td><td><table><tr><td>DR0</td><td>4 2 F 6 E 6 6 6 H</td></tr></table></td></tr><tr><td>↓ (取負值)</td><td></td><td>↓ x0=↗</td></tr><tr><td><table><tr><td>DR0</td><td>- 1 2 3 . 4 5</td></tr></table></td><td></td><td><table><tr><td>DR0</td><td>C 2 F 6 E 6 6 6 H</td></tr></table></td></tr></table>			<table><tr><td>DR0</td><td>1 2 3 . 4 5</td></tr></table>	DR0	1 2 3 . 4 5	⇒ 浮點數：	<table><tr><td>DR0</td><td>4 2 F 6 E 6 6 6 H</td></tr></table>	DR0	4 2 F 6 E 6 6 6 H	↓ (取負值)		↓ x0=↗	<table><tr><td>DR0</td><td>- 1 2 3 . 4 5</td></tr></table>	DR0	- 1 2 3 . 4 5		<table><tr><td>DR0</td><td>C 2 F 6 E 6 6 6 H</td></tr></table>	DR0	C 2 F 6 E 6 6 6 H
<table><tr><td>DR0</td><td>1 2 3 . 4 5</td></tr></table>	DR0	1 2 3 . 4 5	⇒ 浮點數：	<table><tr><td>DR0</td><td>4 2 F 6 E 6 6 6 H</td></tr></table>	DR0	4 2 F 6 E 6 6 6 H													
DR0	1 2 3 . 4 5																		
DR0	4 2 F 6 E 6 6 6 H																		
↓ (取負值)		↓ x0=↗																	
<table><tr><td>DR0</td><td>- 1 2 3 . 4 5</td></tr></table>	DR0	- 1 2 3 . 4 5		<table><tr><td>DR0</td><td>C 2 F 6 E 6 6 6 H</td></tr></table>	DR0	C 2 F 6 E 6 6 6 H													
DR0	- 1 2 3 . 4 5																		
DR0	C 2 F 6 E 6 6 6 H																		

7-24-14 浮點數取絕對值運算(FLOATING POINT NUMBER ABSOLUTE VALUE)

FUN213 P FABS	浮點數取絕對值運算 (FLOATING POINT NUMBER ABSOLUTE VALUE)	FUN213 P FABS													
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。														
階段圖符號 運算控制 — EN — <table><tr><td>213P. FABS</td><td>D</td></tr></table>		213P. FABS	D	D : 存放取絕對值運算結果之暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 S、D 運算元可結合 V、Z、P0~P9 指標作間接定址應用											
213P. FABS	D														
<table><tr><td rowspan="3">運算元 範圍</td><td>HR</td><td>ROR</td><td>DR</td><td>XR</td></tr><tr><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>V、Z P0~P9</td></tr><tr><td>D</td><td>○</td><td>○</td><td>○</td></tr></table>			運算元 範圍	HR	ROR	DR	XR	R0 R34767	R43224 R47319	D0 D7999	V、Z P0~P9	D	○	○	○
運算元 範圍	HR	ROR		DR	XR										
	R0 R34767	R43224 R47319		D0 D7999	V、Z P0~P9										
	D	○	○	○											
功能敘述	- M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5-3 章(數目系統) 5-9 頁。 - 當運算控制“EN” =1 或由 0→1 (P 指令) 時，將 D 所指定之暫存器內容取其絕對值後存回原暫存器 D(浮點數)。														

FUN213P FABS	浮點數取絕對值運算 (FLOATING POINT NUMBER ABSOLUTE VALUE)	FUN213P FABS															
程式範例																	
<table><tr><th>階梯圖</th><th>ST</th></tr><tr><td></td><td></td></tr></table>			階梯圖	ST													
階梯圖	ST																
																	
● 上圖範例，將 DR0 之內容值取絕對值運算之後，回存至 DR0 內(浮點數)。 <table><tr><td>DR0</td><td>-100.25</td><td>⇒ 浮點數：</td><td>DR0</td><td>C2C88000H</td></tr><tr><td colspan="2">↓ (取絕對值)</td><td></td><td colspan="2">↓ x0=↗</td></tr><tr><td>DR0</td><td>100.25</td><td></td><td>DR0</td><td>42C88000H</td></tr></table>			DR0	-100.25	⇒ 浮點數：	DR0	C2C88000H	↓ (取絕對值)			↓ x0=↗		DR0	100.25		DR0	42C88000H
DR0	-100.25	⇒ 浮點數：	DR0	C2C88000H													
↓ (取絕對值)			↓ x0=↗														
DR0	100.25		DR0	42C88000H													

7-24-15 浮點數反正弦函數值運算(FLOATING POINT ARC SINE FUNTION)

FUN218 P FASIN	浮點數反正弦函數值運算 (FLOATING POINT ARC SINE FUNTION)				FUN218 P FASIN																								
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																												
執行控制 — EN — 218P.FASIN S: D: MD: — ERR —			S：求反正弦函數值之來源數值或暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 D：存放結果(反正弦函數值)之暫存器號碼。 S、D 可結合 V、Z、P0~P9 作間接定址應用。 MD:為了讓使用者在使用上較為直觀，MD 可以選擇輸出模式， MD 為 0:輸出的數值為弧度，且輸出為浮點數(32bit); MD 為 1:輸出的數值為角度，且輸出為正整數(16bit)。																										
			<table border="1" style="border-collapse: collapse; text-align: center; width: 100%;"> <thead> <tr> <th style="width: 10%;">運算元 範圍</th><th style="width: 10%;">HR R0 R34767</th><th style="width: 10%;">ROR R43224 R47319</th><th style="width: 10%;">DR D0 D7999</th><th style="width: 10%;">K 浮點數</th><th style="width: 10%;">XR V、Z P0~P9</th></tr> </thead> <tbody> <tr> <td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr> <tr> <td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr> <tr> <td>MD</td><td></td><td></td><td></td><td>0、1</td><td></td></tr> </tbody> </table>			運算元 範圍	HR R0 R34767	ROR R43224 R47319	DR D0 D7999	K 浮點數	XR V、Z P0~P9	S	○	○	○	○	○	D	○	○*	○*		○	MD				0、1	
運算元 範圍	HR R0 R34767	ROR R43224 R47319	DR D0 D7999	K 浮點數	XR V、Z P0~P9																								
S	○	○	○	○	○																								
D	○	○*	○*		○																								
MD				0、1																									
功能敘述																													

- M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5-3 章(數目系統) 5-9 頁。
- 當運算控制“EN”=1 或由 0→1 (**P** 指令) 時，將 S 值或 S 所指定之暫存器內容值取反正弦函數值(單位為弧度,Radian)後存入 D 所指定之暫存器內。
- S 之有效範圍為-1 ~ +1；D 之有效範圍為- $\pi/2$ ~ $\pi/2$ (單位為弧度, Radian)。
- 若 S 值超出有效範圍、或間接定址錯誤，則錯誤旗號“ERR”設為 1，且 D 所指定之暫存器內容值不會被更新。
- 所有浮點運算指令不可在中斷處理副程式裡執行。

FUN218 P FASIN	浮點數反正弦函數值運算 (FLOATING POINT ARCSINE VALUE)	FUN218 P FASIN
程式範例		

- 若 S 值超出有效範圍、或間接定址錯誤，則錯誤旗號“ERR”設為 1，且 D 所指定之暫存器內容值不會被更新。
- 所有浮點運算指令不可在中斷處理副程式裡執行。

階梯圖	ST
The diagram shows a ladder logic network. A normally open contact labeled M0 is connected to the EN (Enable) input of two function blocks. The first function block is labeled '218. FASIN' and contains 'S: R0' and 'D: R4'. Its ERR output is connected to coil M1. The second function block is also labeled '218. FASIN' and contains 'S: R4', 'D: R6', and 'MD: 1'. Its ERR output is connected to coil M2.	先在標籤產生對應的 Tag -> <pre> IF M0 Then FSIN(S:= R0, D:= Tag_FlR4); FASIN_deg(S:= Tag_FlR4, D:= R6); END_IF </pre>

名稱	狀態	資料	註解
DR0	十進制	30	[R0]
DR4	浮點數	0.005235963	[R4]
DR6	十進制	30	[R6]

7-24-16 浮點數反餘弦函數值運算(FLOATING POINT ARC COSINE FUNTION)

FUN219P FACOS	浮點數反正弦函數值運算 (FLOATING POINT ARC COSINE FUNTION)	FUN219P FACOS																														
指令說明	※因浮點數占兩個暫存器，使用間接定址時，需注意不可用到奇數暫存器。																															
執行控制 — EN — 階段圖符號 219P.FACOS S: D: MD: ERR —		S：求反正弦函數值之來源數值或暫存器號碼。 運算元使用之暫存器必須是偶數位址，例 R8 是合法的，R7 是不合法的。 D：存放結果(反正弦函數值)之暫存器號碼。 S、D 可結合 V、Z、P0~P9 作間接定址應用。 MD:為了讓使用者在使用上較為直觀，MD 可以選擇輸出模式， MD 為 0:輸出的數值為弧度，且輸出為浮點數(32bit); MD 為 1:輸出的數值為角度，且輸出為正整數(16bit)。																														
<table><tr><th>範圍</th><th>HR</th><th>ROR</th><th>DR</th><th>K</th><th>XR</th></tr><tr><td>運算元</td><td>R0 R34767</td><td>R43224 R47319</td><td>D0 D7999</td><td>浮點數</td><td>V、Z P0~P9</td></tr><tr><td>S</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr><tr><td>D</td><td>○</td><td>○*</td><td>○*</td><td></td><td>○</td></tr><tr><td>MD</td><td></td><td></td><td></td><td>0、1</td><td></td></tr></table>			範圍	HR	ROR	DR	K	XR	運算元	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9	S	○	○	○	○	○	D	○	○*	○*		○	MD				0、1	
範圍	HR	ROR	DR	K	XR																											
運算元	R0 R34767	R43224 R47319	D0 D7999	浮點數	V、Z P0~P9																											
S	○	○	○	○	○																											
D	○	○*	○*		○																											
MD				0、1																												
功能敘述	● M SERIES 之浮點數格式符合 IEEE-754 所制定之標準，有關浮點數格式之詳細說明請參閱 5-3 章(數目系統)．．．．． 5-9 頁。 ● 當運算控制“EN”=1 或由 0→1 (P 指令) 時，將 S 值或 S 所指定之暫存器內容值取反正弦函數值(單位為弧度,Radian)後存入 D 所指定之暫存器內。 ● S 之有效範圍為-1~+1；D 之有效範圍為-$\pi/2 \sim \pi/2$ (單位為弧度, Radian)。 ● 若 S 值超出有效範圍、或間接定址錯誤，則錯誤旗號“ERR”設為 1，且 D 所指定之暫存器內容值不會被更新。 ● 所有浮點運算指令不可在中斷處理副程式裡執行。																															

FUN218 P FASIN	浮點數反正弦函數值運算 (FLOATING POINT ARCSINE VALUE)	FUN218 P FASIN
-------------------	---	-------------------

程式範例

- 若 S 值超出有效範圍、或間接定址錯誤，則錯誤旗號“ERR”設為 1，且 D 所指定之暫存器內容值不會被更新。
- 所有浮點運算指令不可在中斷處理副程式裡執行。

階梯圖	ST
	先在標籤產生對應的 Tag -> <pre> IF M0 Then FCOS(S:= R0, D:= Tag_FlR4); FACOS_deg(S:= Tag_FlR4, D:= R6); END_IF </pre>

- 當 M0=1，求取 DR4 內容值之反正弦函數值，並將結果以角度(MD=1)存放至 DR6 暫存器內。

DR0	十進制	30	[R0]
DR4	浮點數	0.99998629	[R4]
DR6	十進制	30	[R6]

8

步進指令說明

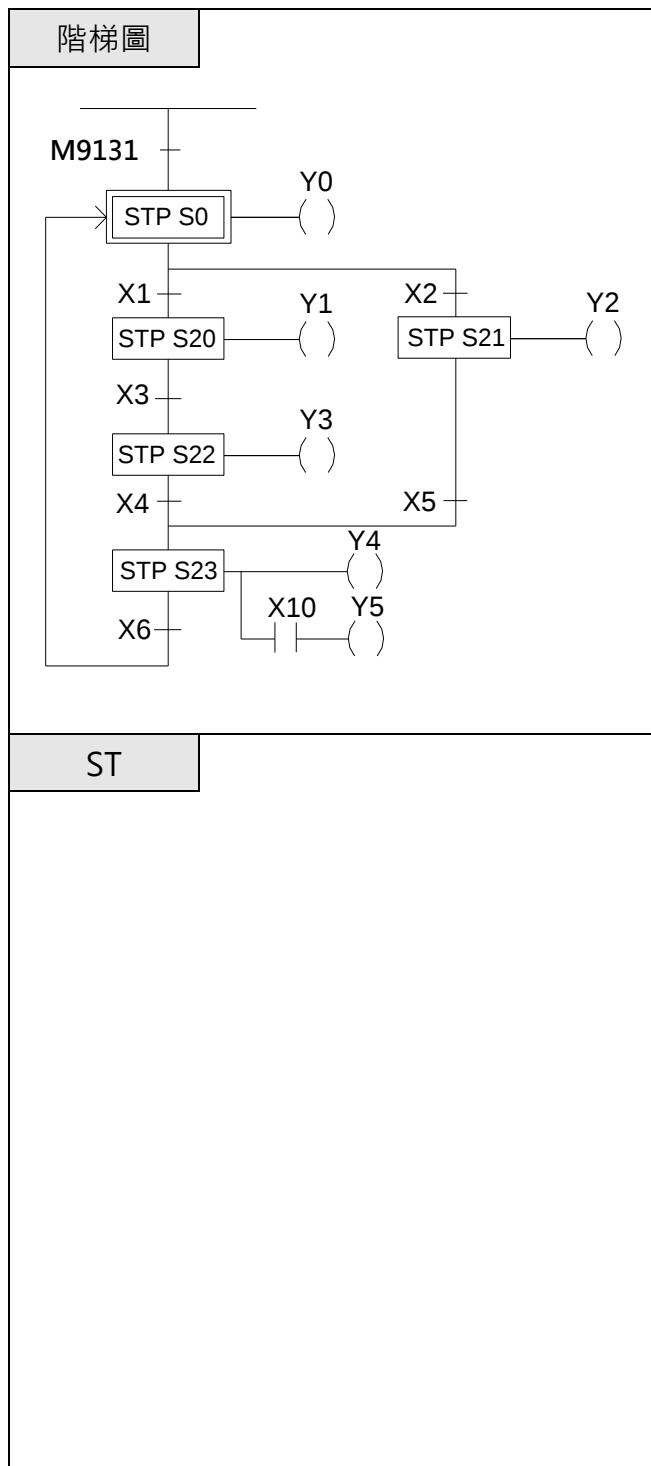
8-1	步進階梯圖工作原理	8-2
8-2	步進階梯圖基本組成	8-3
8-3	步進指令介紹：STP、FROM、TO、STPEND	8-8
8-3-1	STP	8-9
8-3-2	FROM	8-13
8-3-3	TO	8-19
8-3-4	STPEND	8-25
8-4	步進階梯圖寫法	8-26
8-5	實際應用範例	8-31
8-6	步進程式語法檢查錯誤碼說明	8-38

結構化程式設計是軟體設計的一大趨勢，它的特點是可讀性高、維護、更新容易，軟體品質可靠性大大提昇。尤其當控制偏向機械流程控制時，使用傳統的設計方式來設計程式，往往令第三者難以接手，也就是程式可讀性低、更新、維護風險較高。因此，專門針對機械動作流程之順序控制，如能結合現有廣泛之階梯圖語言，再加上步進執行指令輔助，將使這方面的設計工作更省時，更省力，且軟體掌握度更高。我們將這種結合流程控制與階梯圖語言的設計方式稱為步進階梯 (STEP LADDER) 語言。

步進階梯圖是以一個步進點 (STEP) 為最小單元。一個步進點相當於機械運件中的一個步序 (站)，每個步序皆有動作輸出，整台機械或是整個順序控制的流程，便是一個一個步進點邏輯串聯或並聯組成，其一步接一步循序執行之環境，使人對機械之運作一目了然，在設計、操作、維護上皆相當便捷容易。

8-1 步進階梯圖工作原理

【範例】



【說明】

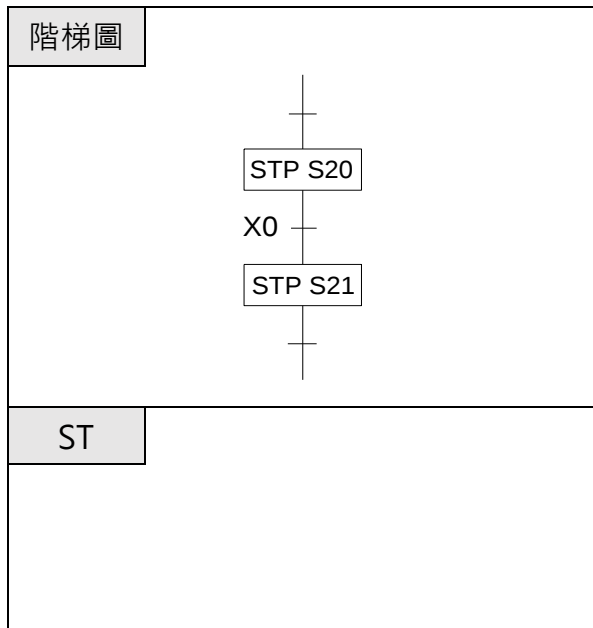
1. **STP Sxxxx** 是步進點 (STEP) Sxxxx 的表示符號，其中 Sxxxx 可從 S0 ~ S3103。當執行到此步進點時 (此點 ON)，便會執行連在右邊的階梯圖，而前一個步進點及輸出皆會變 OFF。
2. M9131 為一開機 ON 一個掃描時間的接點，因此，一開機即進入初始步進點 S0 (S0 ON) 這一站，而其他步進點皆不動作，Y1 ~ Y5 皆 OFF。即 M9131 ON $\rightarrow$ S0 ON $\rightarrow$ Y0 ON，Y0 會維持到 X1 或 X2 其中一接點先 ON 為止。
3. 假設 X2 先 ON，就會執行 S21 這條路徑，即 X2 ON $\Rightarrow \left\{ \begin{matrix} S21 \text{ ON} \\ S0 \text{ OFF} \end{matrix} \right. \Rightarrow \left\{ \begin{matrix} Y2 \text{ ON} \\ Y0 \text{ OFF} \end{matrix} \right.$ ，而 Y2 會維持到 X5 ON 為止。
4. 假設 X5 ON，就會前進至步進點 S23 這一站，即 X5 ON $\Rightarrow \left\{ \begin{matrix} S23 \text{ ON} \\ S21 \text{ OFF} \end{matrix} \right. \Rightarrow \left\{ \begin{matrix} Y4 \text{ ON} \\ Y2 \text{ OFF} \end{matrix} \right.$ ，Y4 和 Y5 會維持到 X6 ON 為止。

※如 X10 ON，則 Y5 也會 ON。

5. 假設 X6 ON，就會前進至步進點 S0 這一站。
- 即 $X6\ ON \Rightarrow \begin{cases} S0\ ON \\ S23\ OFF \end{cases} \Rightarrow \begin{cases} Y0\ ON \\ Y4, Y5\ OFF \end{cases}$
- ，如此便完成一個循環的控制流程，而進入下一循環的控制流程。

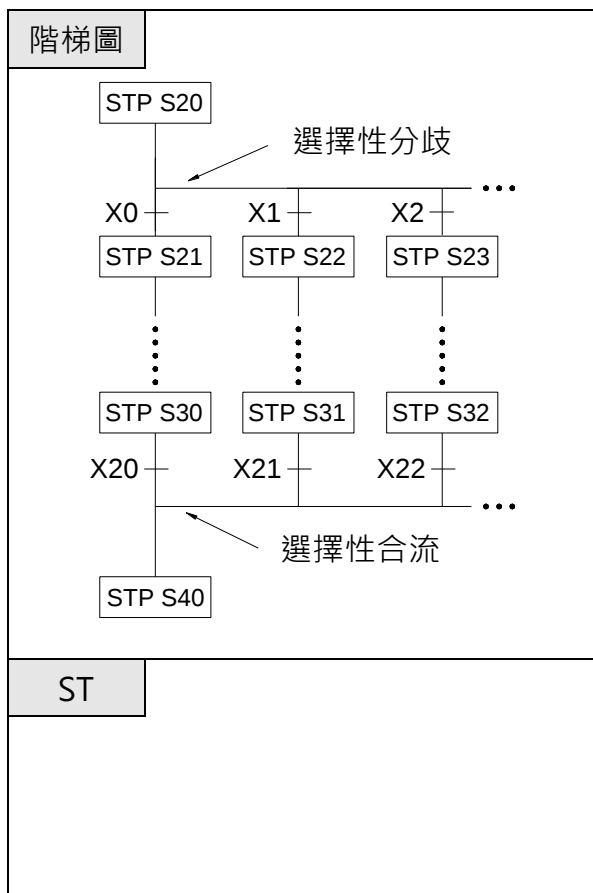
8-2 步進階梯圖基本組成

① 單一回路



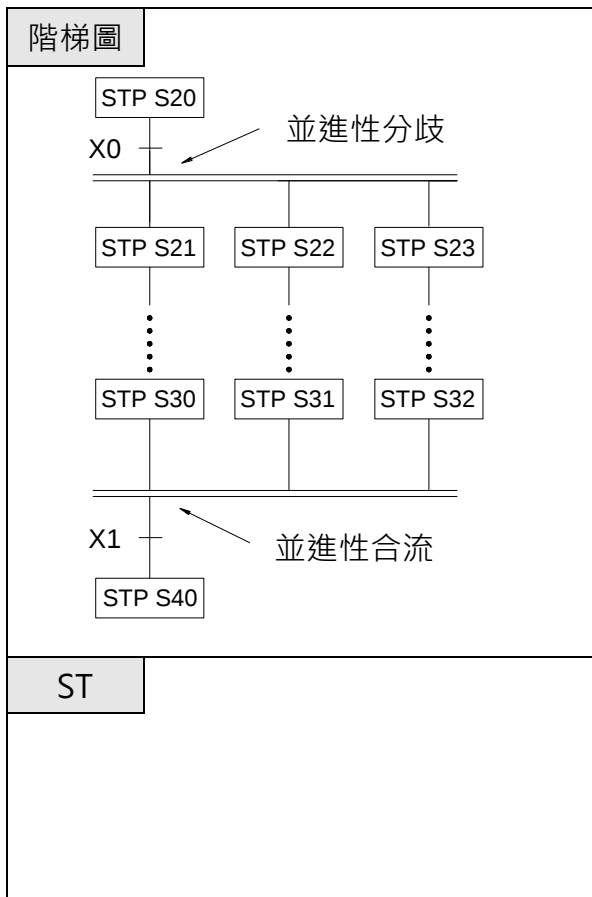
- 步進點 S20 單獨經 X0 前進至步進點 S21。
- X0 可改為其他接點的串、並聯組合。

② 選擇性分歧/合流



- 步進點 S20 選擇其下先 ON 之接點，當作唯一執行的回路，例如 X2 先 ON，則只執行步進點 S23 這條回路。
- 一個分歧最多有 8 條回路。
- X1、X2.....X22 皆可改為其他接點的串、並聯組合。

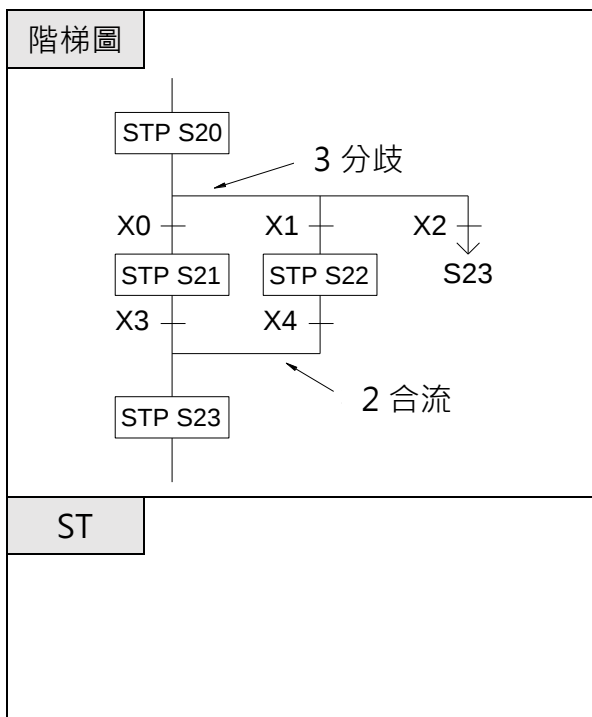
③ 並進式分歧/合流



- 步進點 S20 在 X0 ON 後，同時執行其下的所有回路，即 S21、S22、S23.....皆動作。
- 在合流處上之所有分歧回路皆執行至最後一個步進點 (如 S30、S31、S32)，在等到 X1 ON，即可轉至步進點 S40 執行。
- 並進式分歧的分歧回路數和合流回路數需一致，且最多是 8 條回路。

④ 跳躍

a. 同一步進流程



- 如左邊步進點 S20 下有三條路徑，假設 X2 ON，則直接跳躍至步進點 S23 執行，不需經選擇性合流的過程。
- 但並進式分歧的路徑不能跳躍執行。

b.不同步進流程

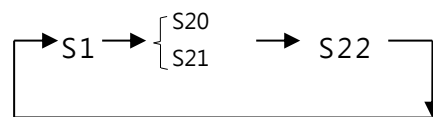
階段圖	ST

⑤ 閉回路型和單循環型

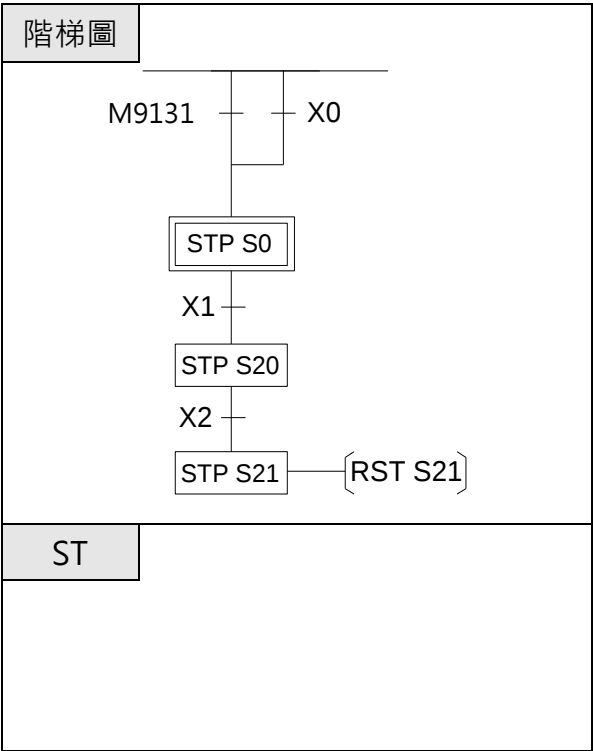
a. 閉回路型

階段圖	ST
ST	

- 一開機初始步進點 S1 ON，往後只會如下做無限次循環。

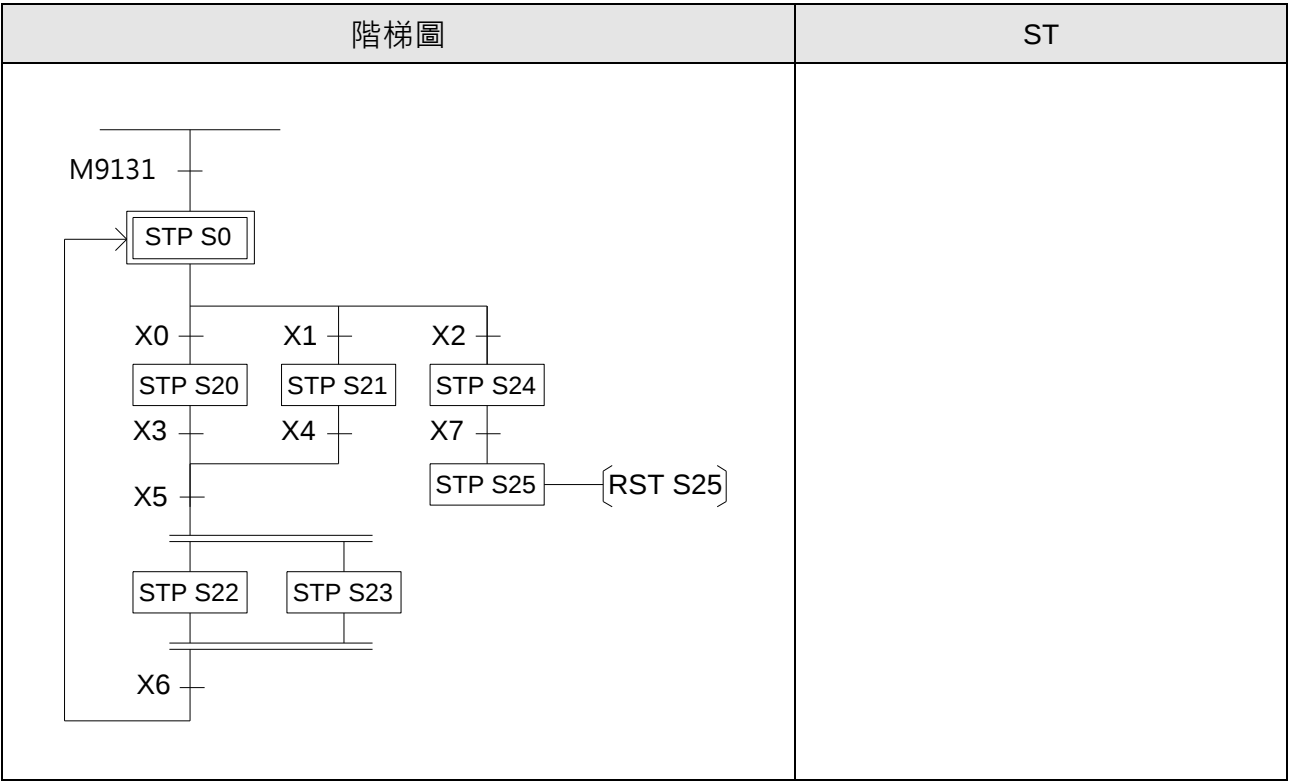


b.單循環型

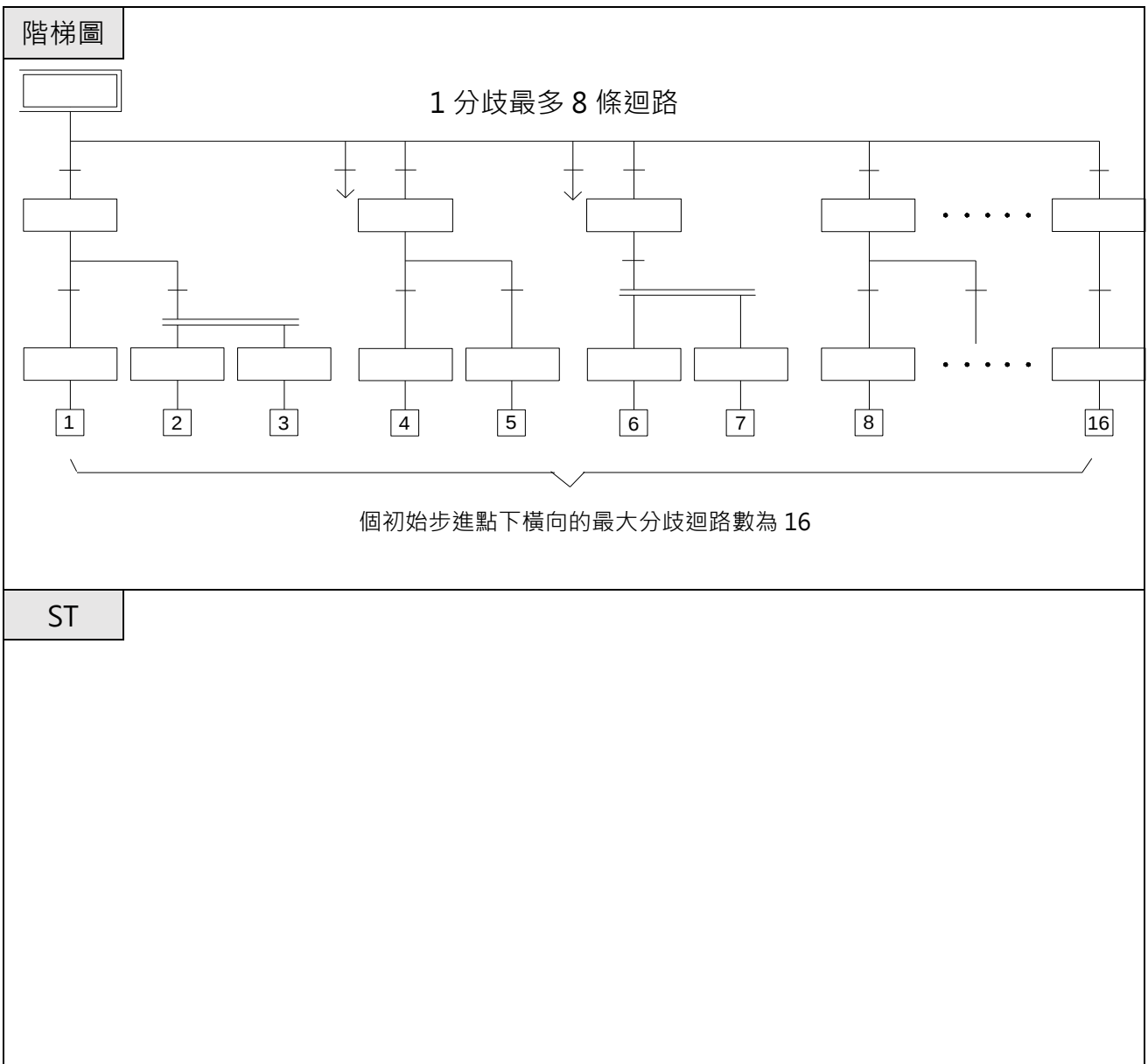


- 在步進點 S20 ON 時，如 X2 ON，S21 本應 ON，但被“RST S21”給 OFF，而結束此步進流程。

c.混合型流程



⑥綜合應用



8-3 步進指令介紹：STP、FROM、TO、STPEND

本小節將介紹步進指令，以及在 UperLogic 中如何呼叫指令，並介紹如何使用。

步進指令可以藉由以下方法呼叫：

選取功能列 **階段** → **功能指令**；或點選元件盤圖示；或於階段圖程式區按右鍵出現快顯功能表選取 **功能指令** → **功能指令**，於階段圖程式區要輸入步進指令之位置點選一下，會出現所有類別的功能指令，於類別項下選擇 **[SFC 指令]**，其右邊的指令名稱即出現 **[STP]**、**[FROM]**、**[TO]**、**[STPEND]** 四種步進指令，如下圖：



8-3-1 STP

- **STP Sx** : $S0 \leq Sx \leq S7$ (UperLogic 輸入/顯示格式) 或 **STP Sx** : $S0 \leq Sx \leq S7$

此指令為初始步進點 (Initial Step) 指令，由此指令才可往下衍生出各個機械流程之步進控制。M 系列最多可提供 8 個初始步進點，也就是說一台 PLC 最多可同時作 8 個流程控制。每一步進流程可獨立運作或是產生運作結果供其他流程參考使用。

【範例一】每次開機啟動初始步進點 S0

階梯圖	ST

【範例二】每次開機或按手動鈕或自動生產異常發生而在某特定時間內無人員處置自動進入初始步進點 S0 待機

階梯圖	ST

【說明】X0：手動鈕；M0：異常之接點

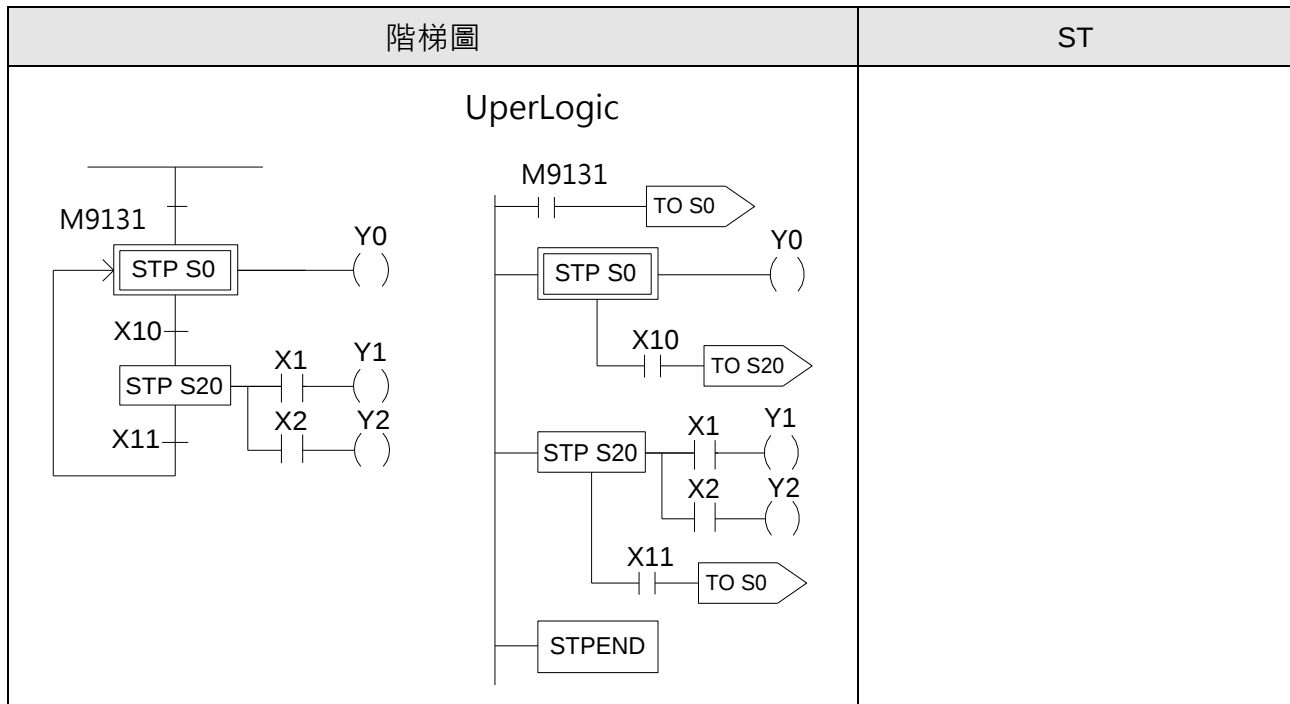
- **STP Sxxxx** : $S20 \leq Sxxxx \leq S3103$ (UperLogic 輸入/顯示格式)

或

STP Sxxxx : $S20 \leq Sxxxx \leq S3103$

此指令為流程中之步進點指令，每一步進點代表一個步序 (站)，ON 代表該步序作動，並會執行該步序下的階梯程式。

【範例】



【說明】

- 1.開機時，初始步進點 S0 ON、Y0 ON。
- 2.當轉進條件 X10 (實際使用時，轉進條件可由 X、Y、M、T、C 各接點之串、並聯組合而成) ON 時，則步進點 S20 作動，當次掃描時間內系統會自動將 S0 OFF，且 Y0 自動清除為 OFF。

即 X10 ON $\Rightarrow$ $\begin{cases} S20 \text{ ON} \\ S0 \text{ OFF} \end{cases} \Rightarrow \begin{cases} X1 \text{ ON} \rightarrow Y1 \text{ ON} \\ X2 \text{ ON} \rightarrow Y2 \text{ ON} \\ Y0 \text{ OFF} \end{cases}$

- 3.當轉進條件 X11 ON 時，則步進點 S0 ON，Y0 也 ON，同時 S20、Y1 和 Y2 變 OFF。

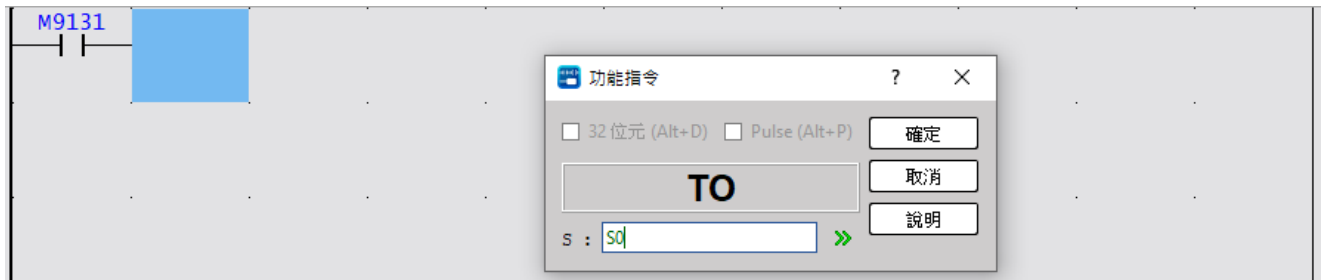
即 X11 ON $\Rightarrow$ $\begin{cases} S0 \text{ ON} \\ S20 \text{ OFF} \end{cases} \Rightarrow \begin{cases} Y0 \text{ ON} \\ Y1 \text{ OFF} \\ Y2 \text{ OFF} \end{cases}$

- 輸入步進點 (STP 指令)

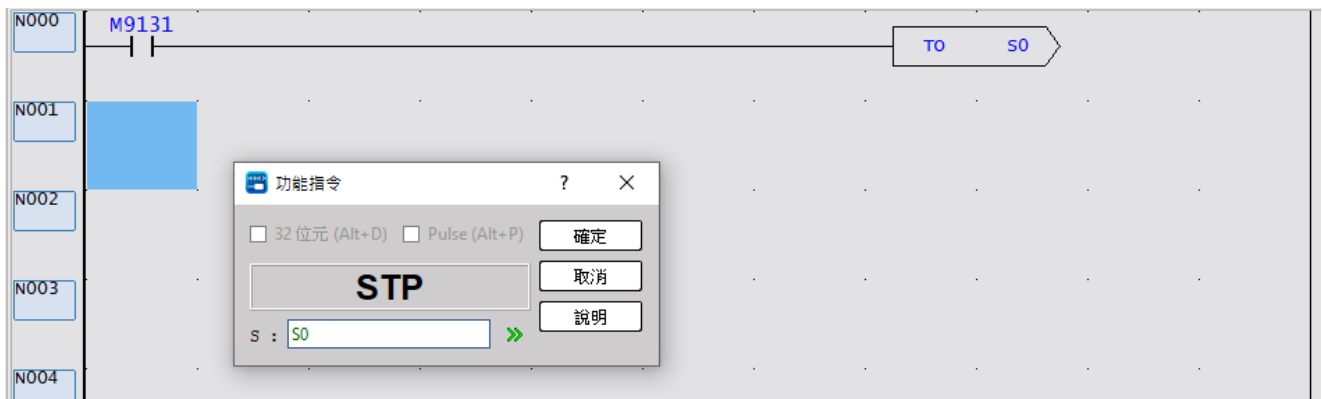
若我們要設定一每次開機啟動初始步進點 S0，其作法如下：選取元件盤上 A 接點元件，於階梯圖網路上點選一下，於編號輸入框輸入 “M9131”：



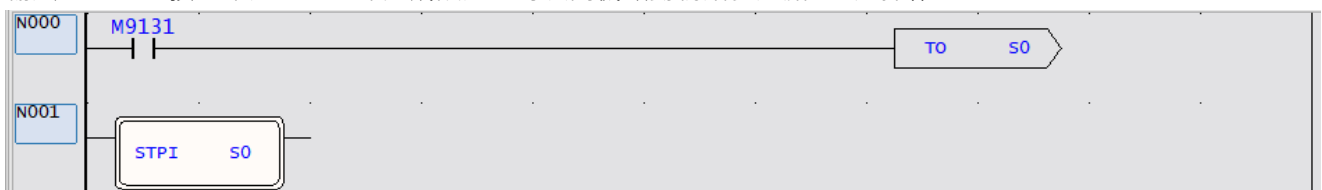
點選元件盤圖示，於 “M9131” 接點之後點選一下，出現〔應用指令〕視窗，〔類別〕項下選擇 “SFC 指令”，〔指令名稱〕選擇 “TO”，按 “確定” 鈕，出現如下視窗：



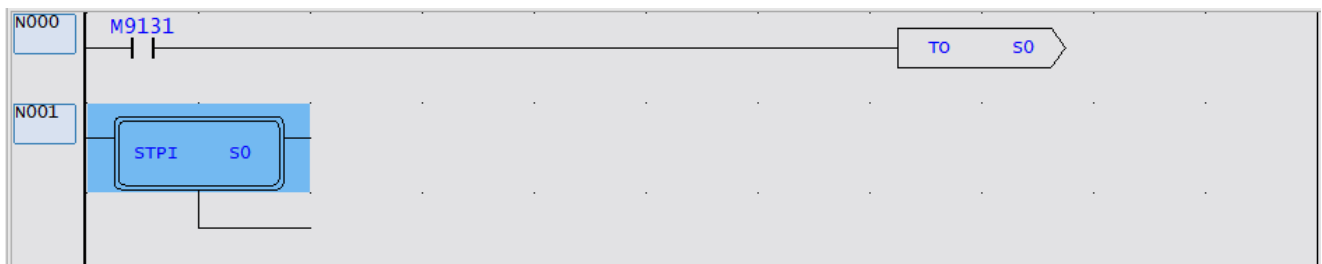
輸入 “S0”，按 “確定” 鈕，再重複 “SF 指令”，此次〔指令名稱〕選擇 “STP”，出現如下圖：



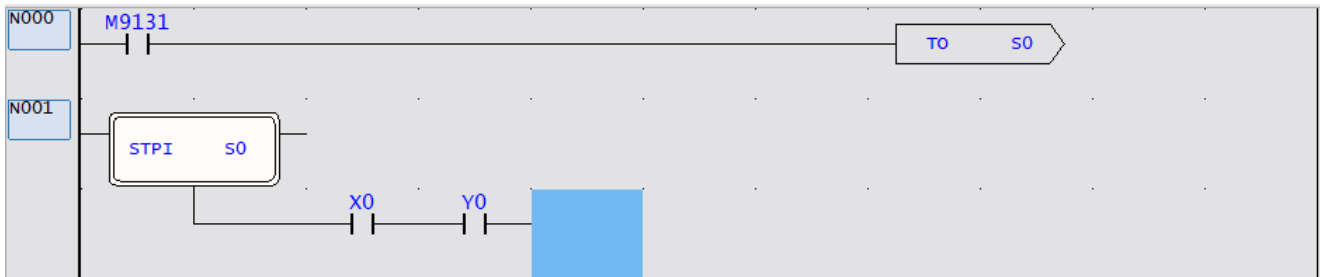
輸入 “S0”，按 “確定” 鈕，即完成設定一每次開機啟動初始步進點 S0 之操作。



亦可為初始步進點加入狀態轉移條件，首先將游標於元件盤選擇〔垂直線〕元件，然後在 “STPI S0” 上點一下；或將游標停於 “STPI S0” 上，再按快速鍵 “V” 也可以。



出現分歧線之後，加上轉移條件，例如我們加上 “X0” 和 “Y0” 兩個轉移條件。



之後再加上欲轉移之狀態點，我們假設當 " X0 " 和 " Y0 " 兩個轉移條件滿足時 (ON)，便轉移至狀態

點 " S21 "。叫出〔SFC 功能指令〕類別，指令名稱選擇〔TO〕，輸入 " S21 "，便可完成下列範例：



8-3-2 FROM

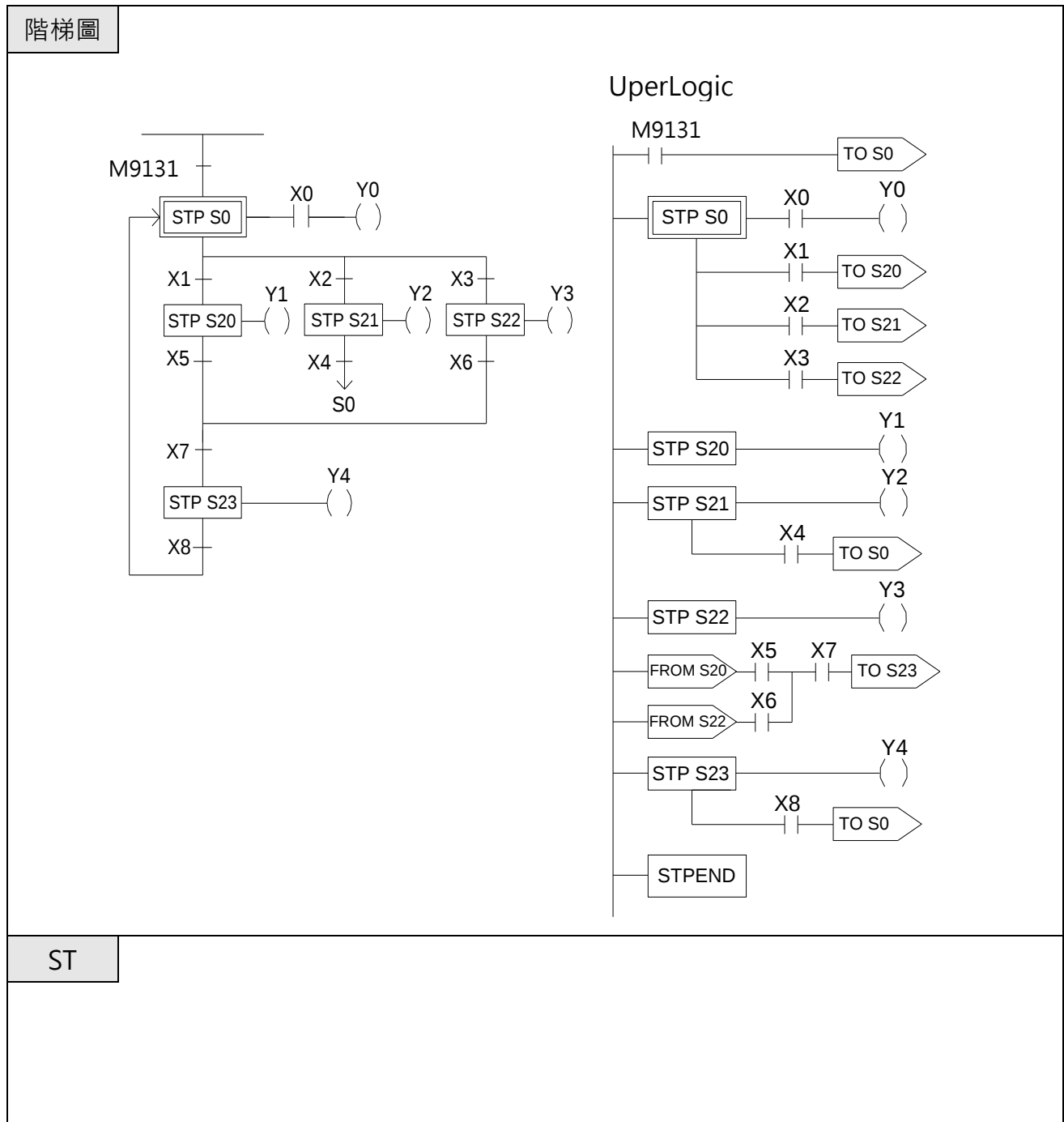
- FROM Sxxxx → $S0 \leq Sxxxx \leq S3103$ (UperLogic 輸入/顯示格式)

或

FROM Sxxxx : $S0 \leq Sxxxx \leq S3103$

此指令描述轉進之來源步進點，亦即要由步進點 Sxxx 配合轉進條件前進至下一個步進點。

【範例】



【說明】：

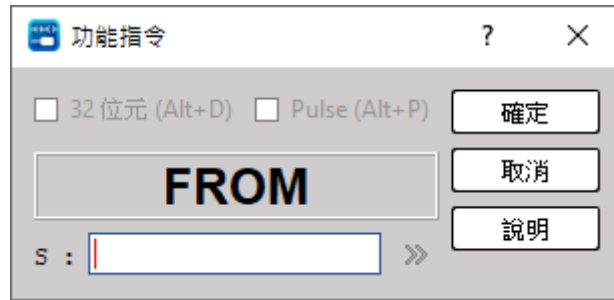
- 1.開機時進入初始步進點 S0 ON；X0 ON 則 Y0 ON。
- 2.S0 ON 時，
 - a.當 X1 ON 時，則步進點 S20 ON、Y1 ON。
 - b.當 X2 ON 時，則步進點 S21 ON、Y2 ON。
 - c.當 X3 ON 時，則步進點 S22 ON、Y3 ON。
 - d.如果 X1、X2 和 X3 同時 ON，則步進點 S20 優先 ON，S21 或 S22 不會 ON。
 - e.如果 X2 與 X3 同時 ON，則步進點 S21 優先 ON，S22 不會 ON。
- 3.S20 ON，當 X5 和 X7 同時 ON 時，則步進點 S23 ON、Y4 ON、S20 OFF、Y1 OFF。
- 4.S21 ON，當 X4 ON 時，則步進點 S0 ON、S21 OFF、Y2 OFF。
- 5.S22 ON，當 X6 和 X7 同時 ON 時，則步進點 S23 ON、Y4 ON、S22 OFF、Y3 OFF。
- 6.S23 ON，當 X8 ON 時，則步進點 S0 ON、S23 OFF、Y4 OFF。

- 輸入合流點(FROM 指令)

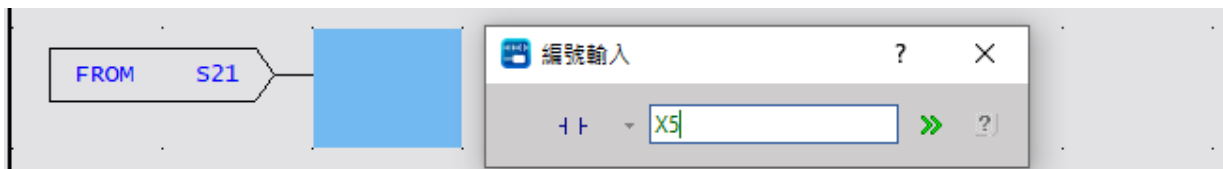
1. 選擇性合流



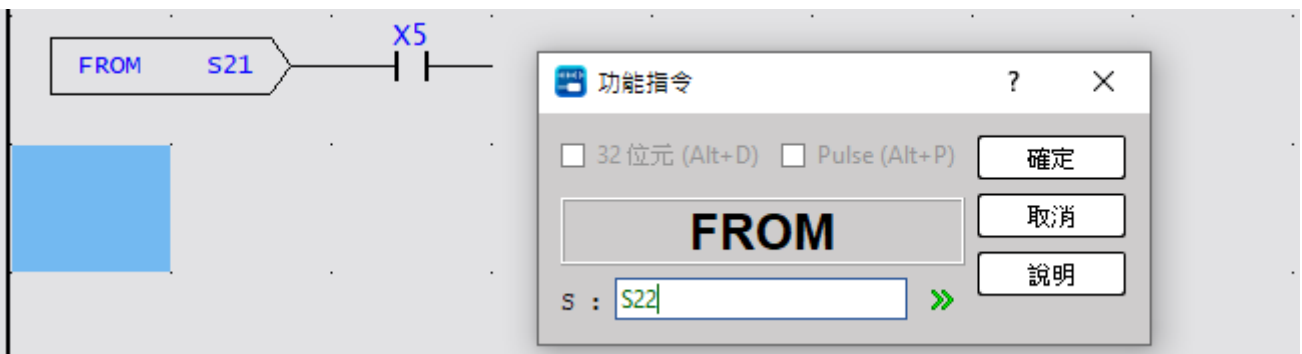
若我們要做出以上結果時的作法如下：我們先參見 8-3 節的操作方法叫出〔SFC 功能指令〕類別，指令名稱選擇〔FROM〕，按“確定”之後出現如下視窗：



輸入“S21”，按“確定”鈕，將游標於元件盤選擇〔A 接點〕元件於其後點一下出現如下視窗：



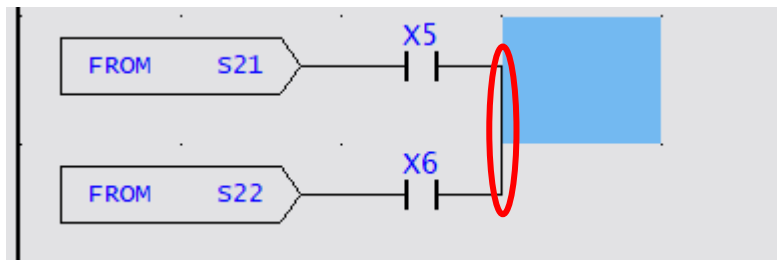
輸入“X5”，按“ENTER”鍵，再一次使用功能指令，叫出〔SFC 功能指令〕類別，指令名稱選擇〔FROM〕，按“確定”：



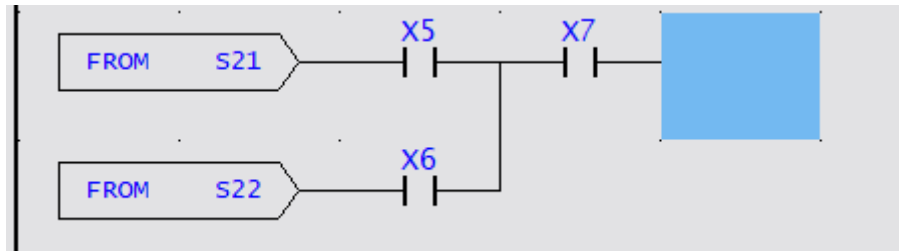
輸入“S22”，按“確定”鈕，將游標於元件盤選擇〔A 接點〕元件於其後點一下出現如下視窗：



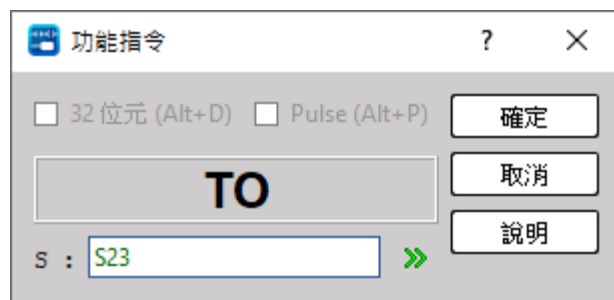
輸入“X6”，按“ENTER”鍵，此時游標於元件盤選擇〔垂直線〕元件，緊接著 X5 接點之後點一下；或游標放置於 X5 之後按快速鍵“V”，即出現一垂直線，如下圖所示：



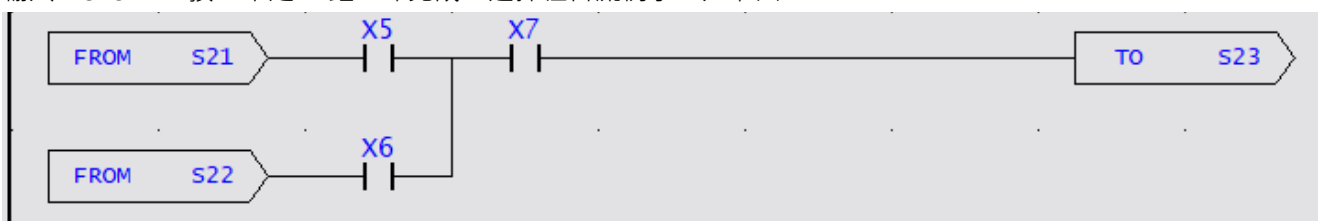
接下來輸入 “X7”，如下圖所示：



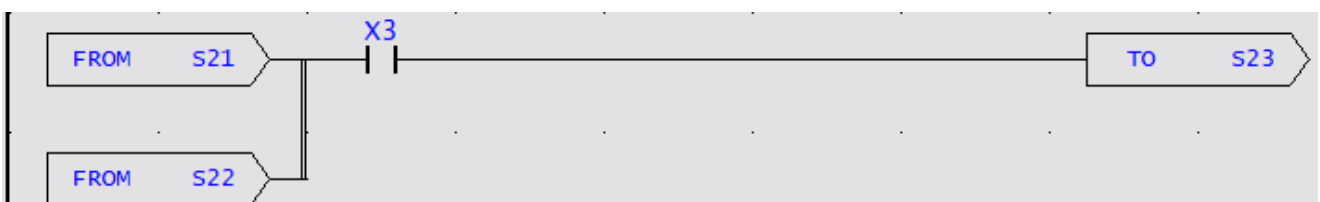
再一次使用功能指令，叫出〔SFC 功能指令〕類別，指令名稱選擇〔TO〕按“確定”之後出現如下視窗：



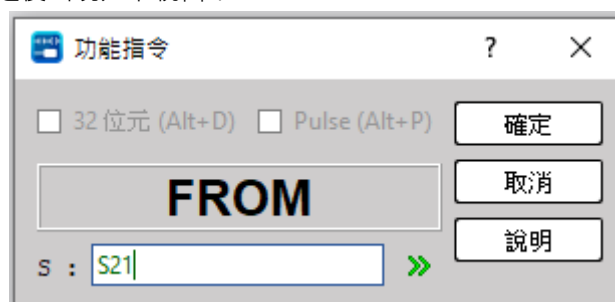
輸入 “S23”，按“確定”鈕，即完成一選擇性合流例子。如下圖。



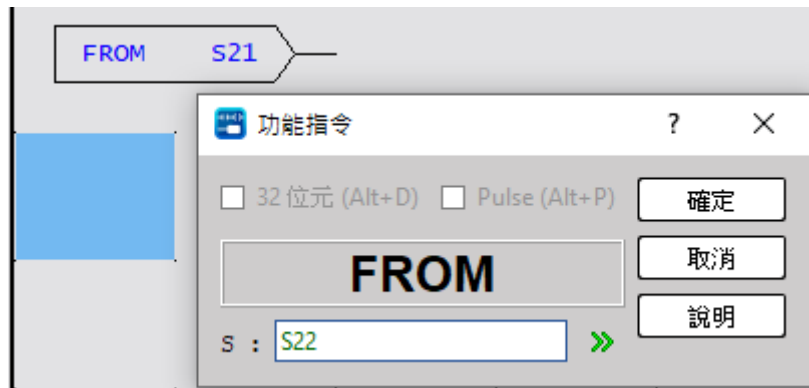
2. 並進性合流

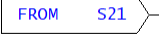


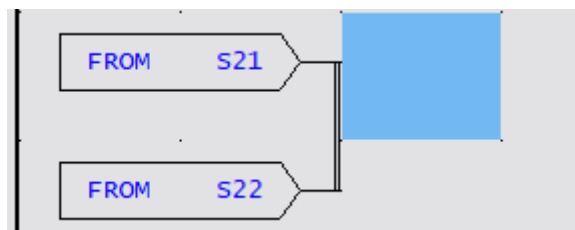
若我們要做以上結果時的作法如下：我們先參見 8-3 節的操作方法叫出〔SFC 功能指令〕類別，指令名稱選擇〔FROM〕，按“確定”之後出現如下視窗：



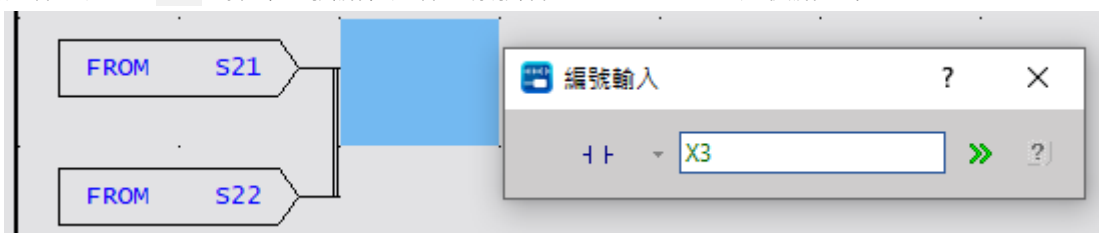
輸入“S21”，按“確定”鈕，再次叫出〔SFC 功能指令〕類別，指令名稱選擇〔FROM〕，按“確定”之後出現如下視窗：



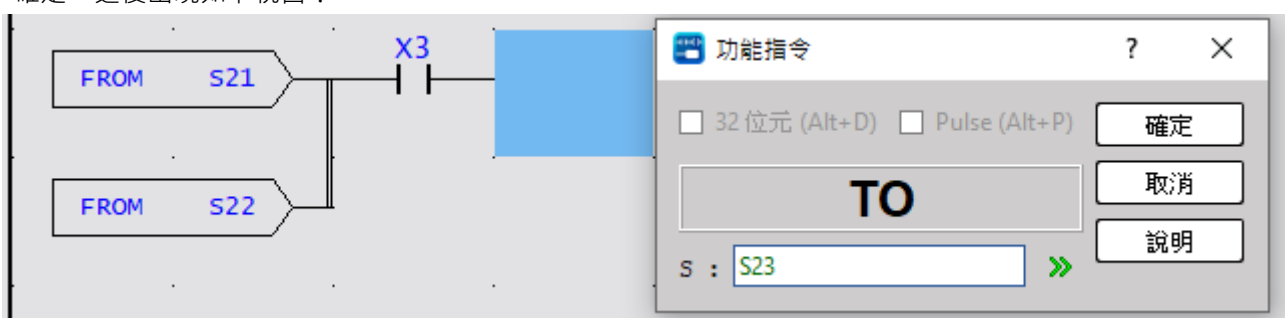
輸入“S22”，按“確定”鈕，將游標於元件盤選擇〔垂直線〕元件，緊接著  之後點一下，即完成並進性合流的階梯圖程式的表現方式。



將游標於元件盤選  擇〔A 接點〕元件，緊接著  之後點一下：

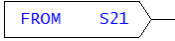


輸入“X3”，按“ENTER”鍵。再一次使用功能指令，叫出〔SFC 功能指令〕類別，指令名稱選擇〔TO〕按“確定”之後出現如下視窗：



輸入“S23”，按“確定”鈕，即完成一並進性合流例子。如下圖：



要特別注意的是若要完成並進性合流其  [垂直線] 元件，一定要緊鄰著 ，一但中間有空格，就會變成選擇性合流，如下所示：



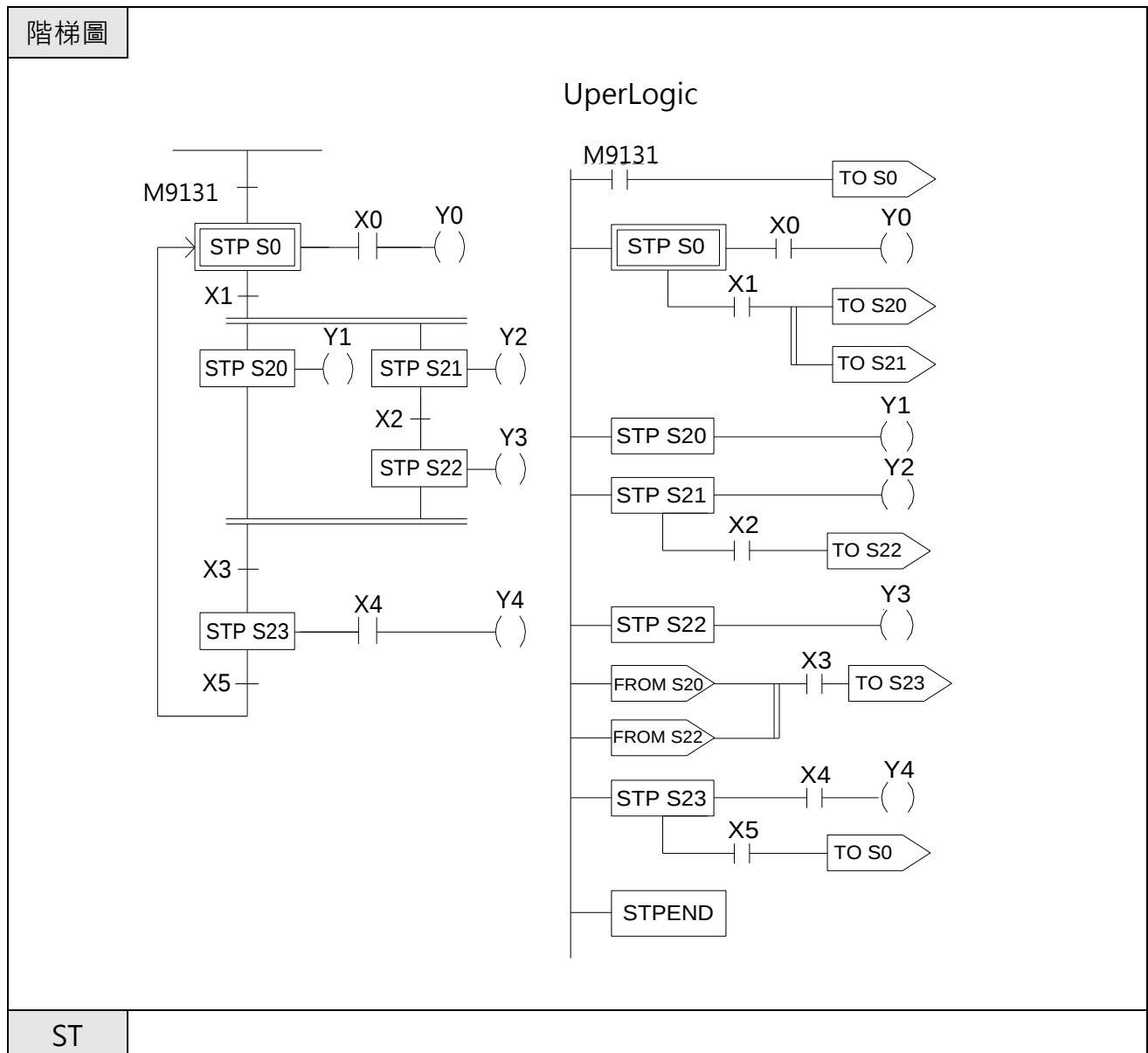
8-3-3 TO

- **TO Sxxxx** $\rightarrow$ $S0 \leq Sxxxx \leq S3103$ (UperLogic 輸入/顯示格式)
或

TO Sxxxx : $S0 \leq Sxxxx \leq S999$

此指令描述欲轉往之步進點。

【範例】



ST

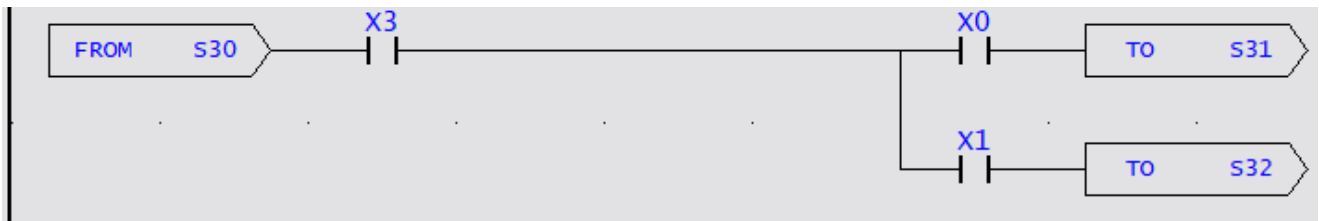


【說明】：

1. 開機時進入初始步進點 S0 ON；X0 ON 則 Y0 ON。
 2. S0 ON，當 X1 ON 時，則同時步進點 S20 ON、S21 ON，兩路並進；Y1 ON、Y2 ON。
 3. S21 ON，當 X2 ON 時，步進點 S22 ON、Y3 ON、S21 OFF、Y2 OFF。
 4. S20 和 S22 同時 ON 且轉進條件 X3 ON 時，則步進點 S23 ON (X4 ON 時 Y4 ON)；而 S20 和 S22 自動 OFF、Y1 和 Y3 變為 OFF。
 5. S23 ON，當 X5 ON 時，則轉進回復至初始步進點，即 S0 ON、S23 OFF、Y4 OFF。
- 輸入分歧點(TO 指令)

舉例說明在使用 UperLogic 階梯圖程式的操作方法及步驟如下：

1. 選擇性分歧

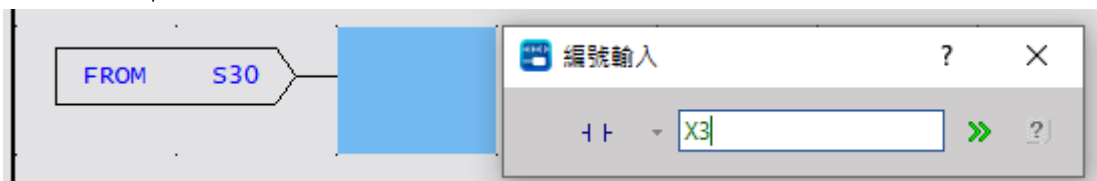


若我們要做出以上結果時的作法如下：

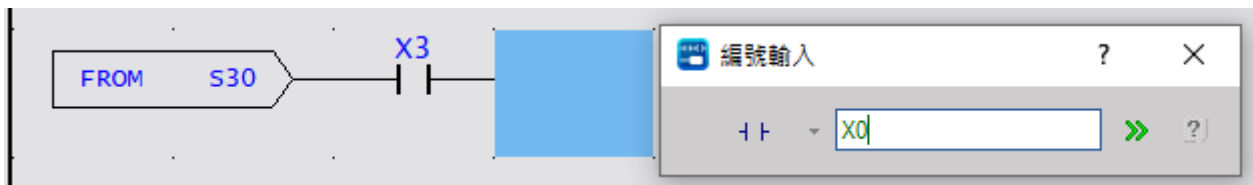
於游標放置於程式區中欲輸入處位置，叫出〔SFC 功能指令〕類別，指令名稱選擇〔FROM〕：



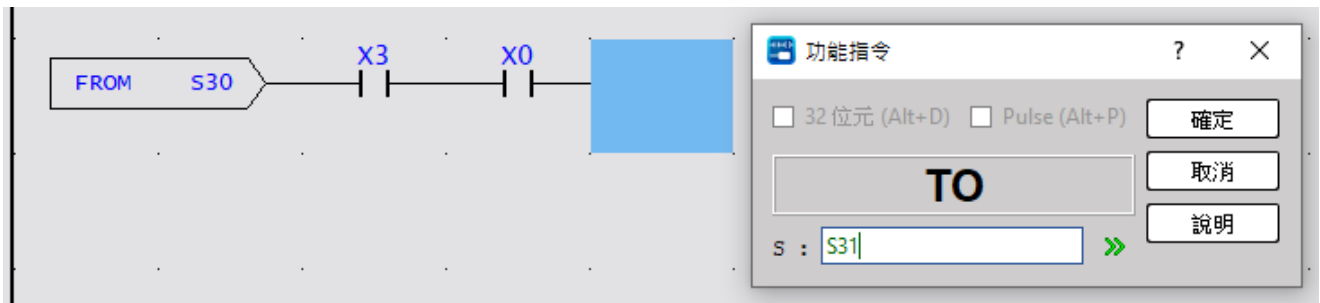
輸入“S30”，按“確定”鈕後，程式區即出現 FROM 指令 S30 元件。游標選擇 A 接點元件緊接其後點選一下，輸入“X3”編號；或緊接其後直接輸入“AX3”，如下視窗：



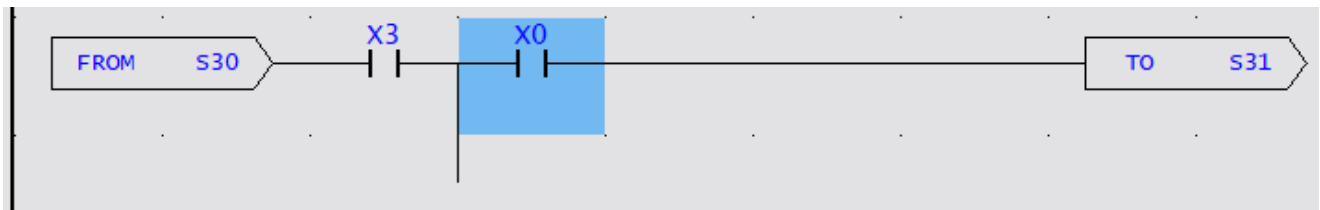
接下來在 X3 之後鍵入 X0 接於其後，



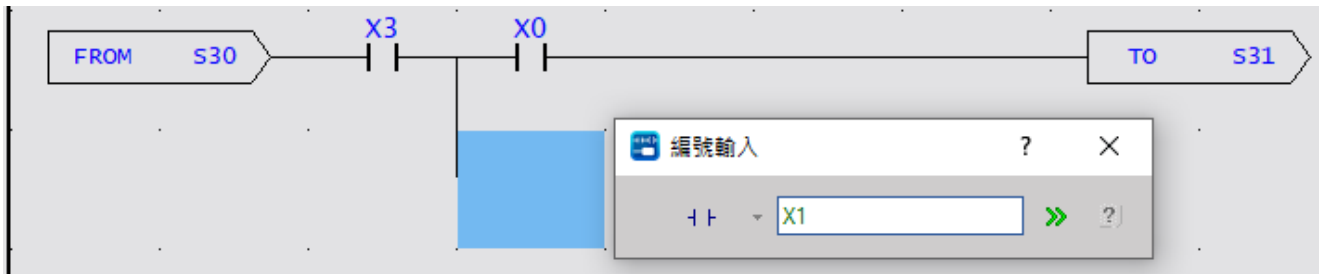
於 X0 之後，游標放置於程式區欲輸入處位置，再叫出〔SFC 功能指令〕類別，指令名稱選擇〔TO〕；



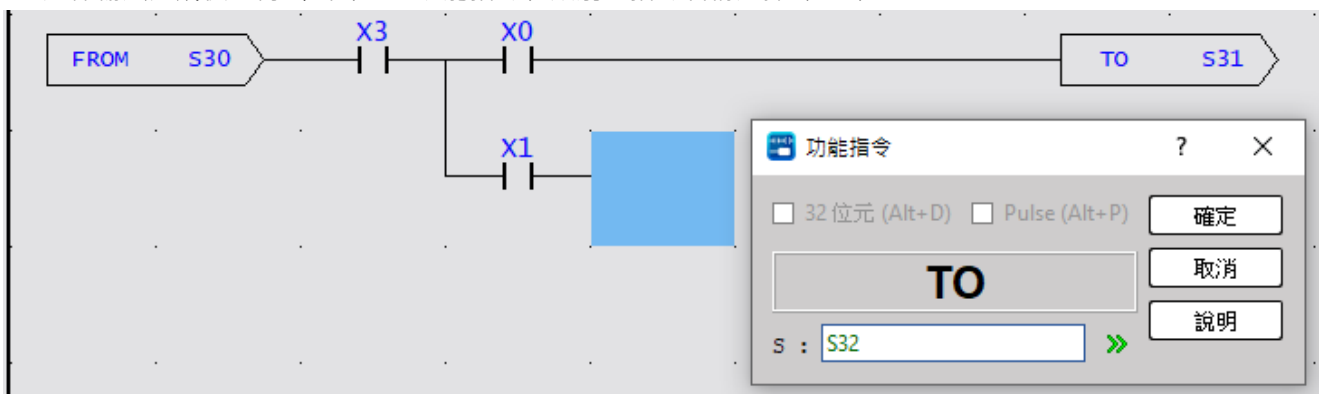
輸入 "S31"，按 "確定" 鈕，游標置於 X0 位置，輸入 "V"，增加一垂直線，如下圖：



再將游標置於 X0 下方位置，輸入 "X1" 或 "AX1" (A 為 A 接點的快捷鍵)：



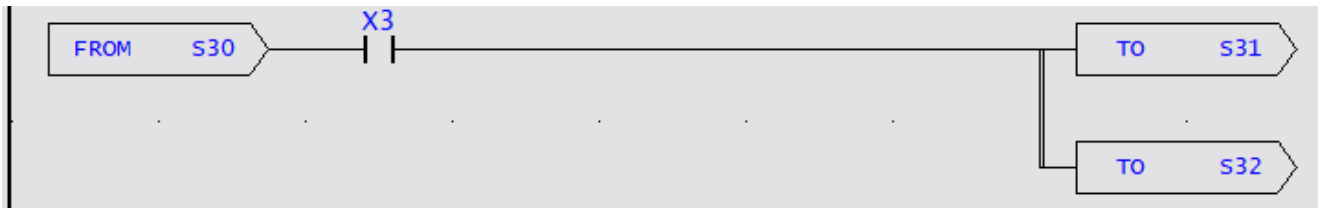
X1 元件輸入完成後，再叫出 [SFC 功能指令] 類別，指令名稱選擇 [TO]：



輸入 "S32"，按 "確定" 鈕後，即完成一選擇性分歧例子。如下圖：



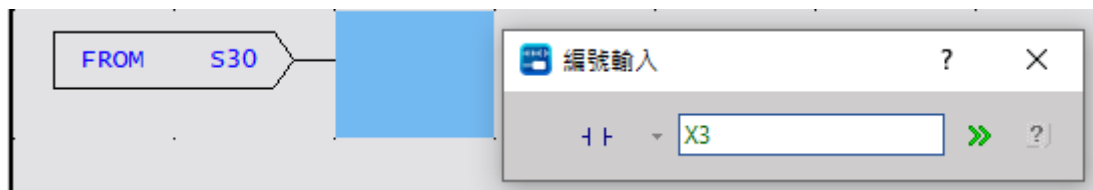
2. 並進性分歧



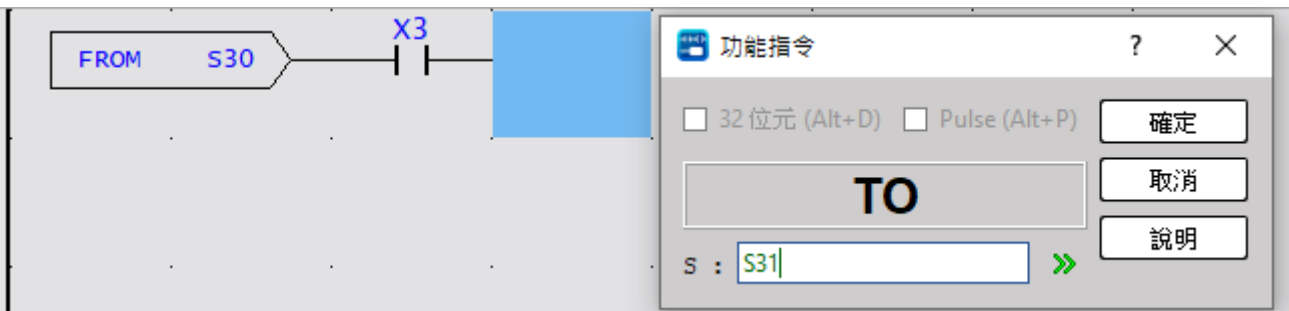
若我們要做出以上結果時的作法如下：於游標放置於程式區欲輸入處位置，叫出〔SFC 功能指令〕類別，指令名稱選擇〔FROM〕：



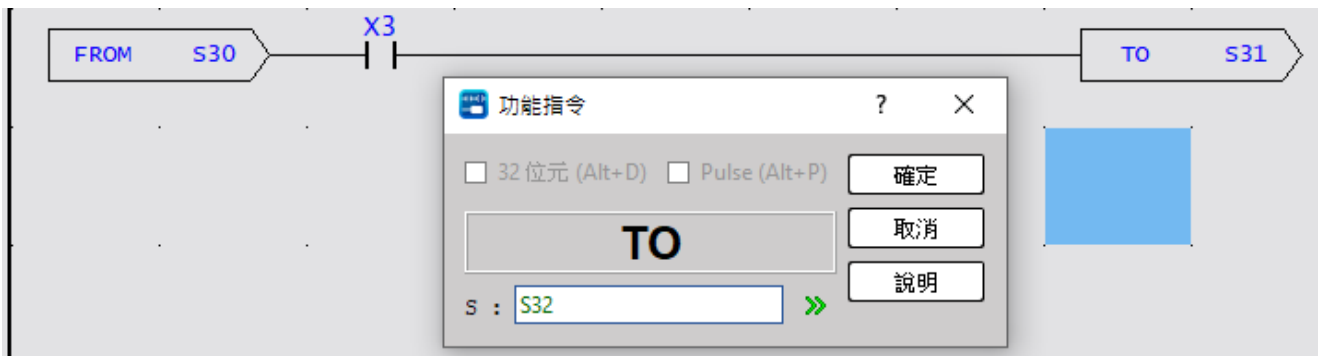
輸入“S30”，按“確定”鈕後，即出現 FROM 指令。游標選擇 A 接點元件緊接其後點選一下，輸入“X3”編號；或緊接其後直接輸入“AX3”，如下視窗：



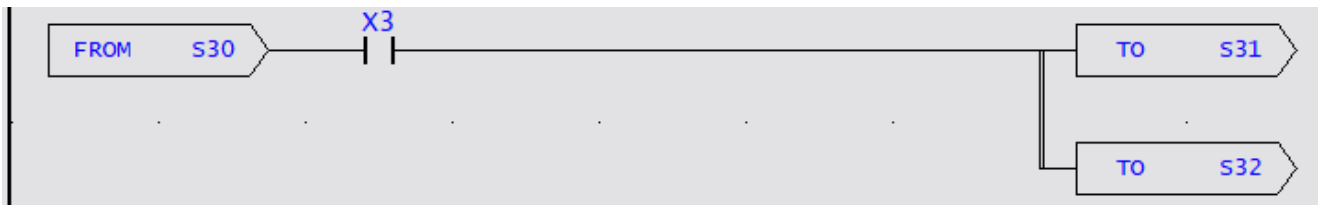
將游標放置於程式區欲輸入處位置，再叫出〔SFC 功能指令〕類別，指令名稱選擇〔TO〕：



輸入“S31”，按“確定”鈕後，即出現 TO 指令。於完成之 TO 指令 S31 下方位置，再叫出〔SFC 功能指令〕類別，指令名稱選擇〔TO〕：



輸入“S32”按“確定”鈕。游標選擇垂直線元件，點選程式區中圖示 ；或按快速鍵“V”，即出現如下圖所示：



即完成一並進性分歧例子。

8-3-4 STPEND

- **STPEND** : (UperLogic 輸入/顯示格式)

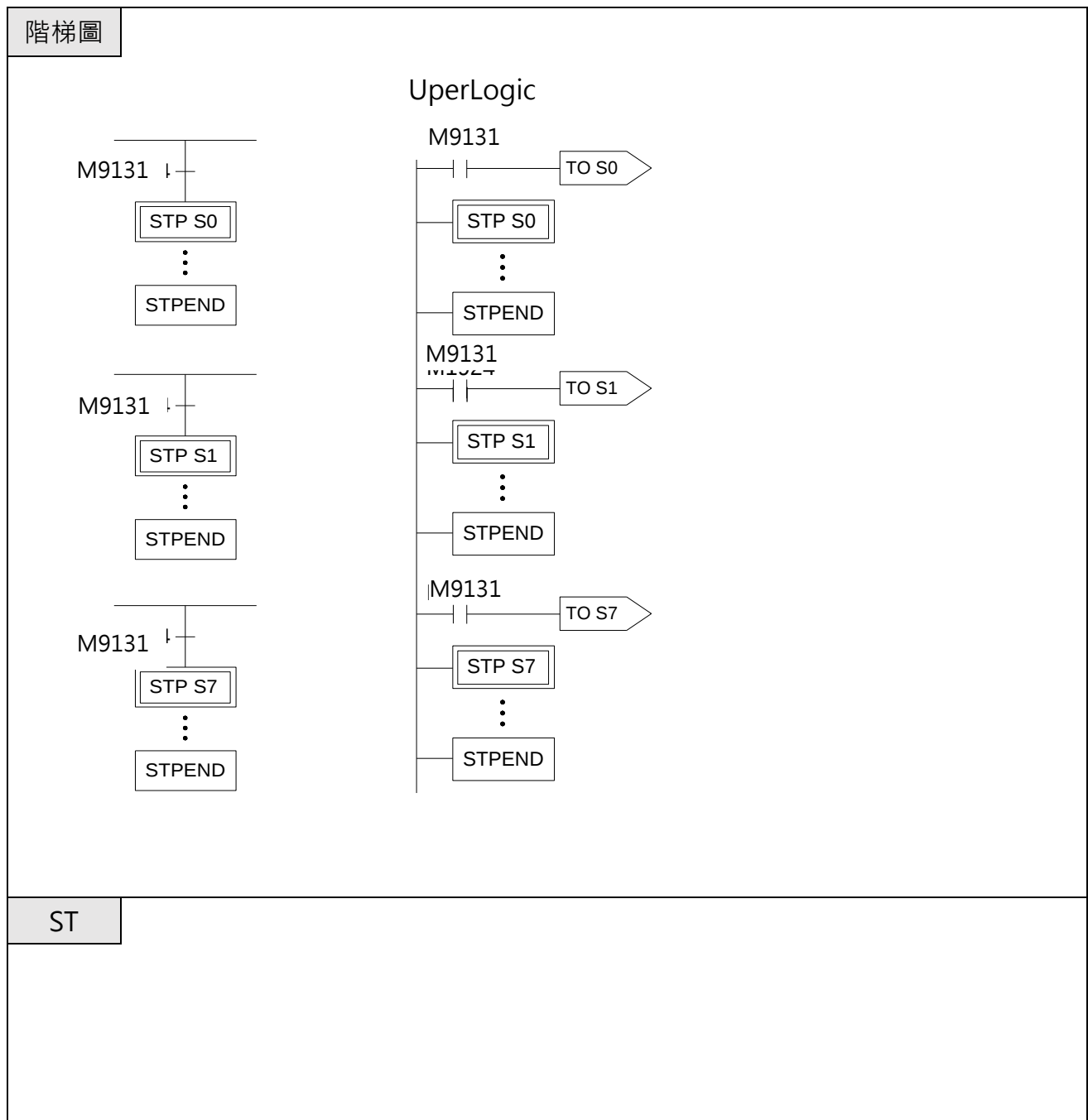
或

STPEND

此指令代表一流程指令之結束，必須有此指令，所有流程才會正確運作。

PLC 最多有 8 個步進流程 (S0 ~ S7) 可同時控制，所以最多有 8 個 STPEND 指令。

【範例】



【說明】開機時 8 個步進流程同時作動。

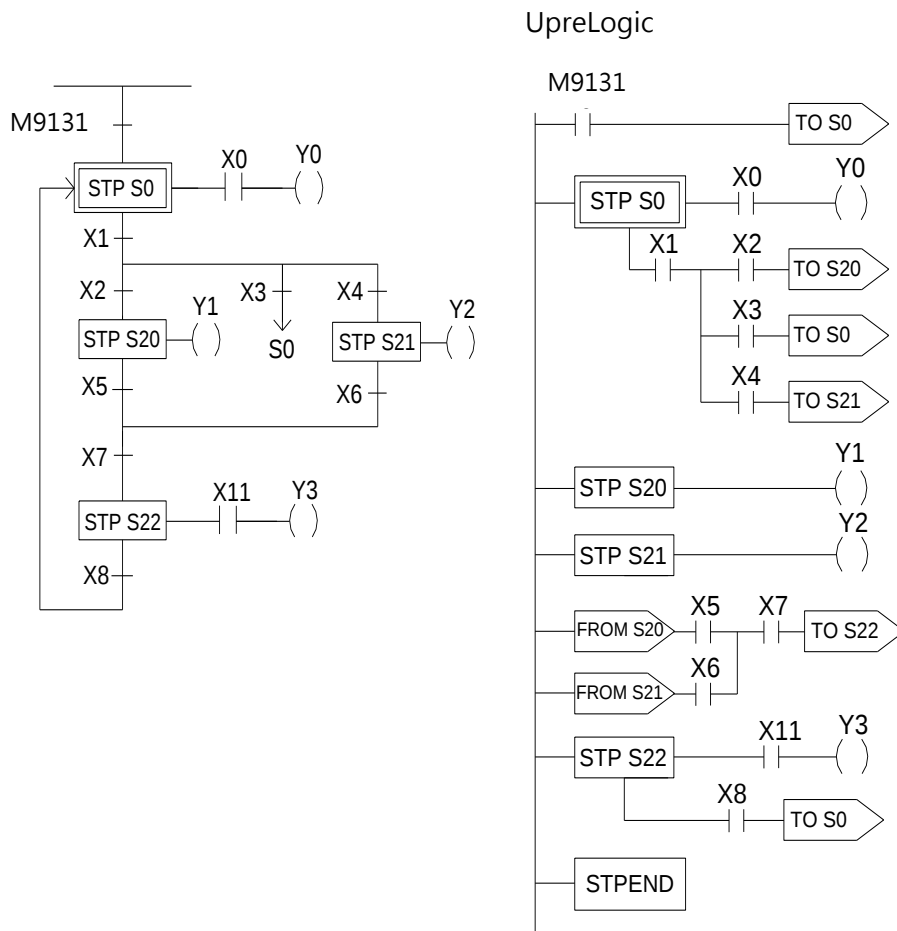
8-4 步進階梯圖寫法

【注意事項】

- 在實際的應用上，可將單純的階梯圖和步進階梯圖組合使用。
- 作為開頭的步進點我們稱為初始步進點，共有 8 點，即 S0 ~ S7。
- 要讓初始步進點作動當然可以由任何一個步進點來加以觸發，但 PLC 開始運轉時，必須讓初始步進點 ON；我們可利用系統提供之 M9131（第一次掃描 ON 信號）來觸發初始步進點 ON。
- 除了初始步進點用上述方法觸發啟動外，其他的步進點的觸發必須由另外一個步進點來驅動。
- 在步進階梯圖程式當中必須有開頭之初始步進點，及最後之 STPEND 指令，才算一完整之步進流程程式。
- 一般步進點共有 3085 點，由 S20 ~ S3103，可任意使用，無須按順序，但號碼絕不可重覆使用。系統內定 S2064 ~ S3103 為停機保持型（當然可由 USER 修改），機械流程在斷電後如想繼續斷電前之動作，則可使用這些步進點。
- 一個步進點在基本上必須具備驅動步進點內輸出負載、指定轉進條件及轉進目的地等三個功能。
- 在步進程式中不可使用 MC，SKP 指令；副程式區不能輸入步進程式。當然 JMP 指令儘量少用。
- 若步進轉進後，輸出點仍需保持 ON 則需用 SET 指令推動該輸出點；要清除該輸出點為 OFF，則需用 RST 指令。
- 從一個初始步進點往下看，橫向分歧步進點最大 16 點，但一個分歧點最多只可作 8 個分歧回路。

【範例 1】

階梯圖

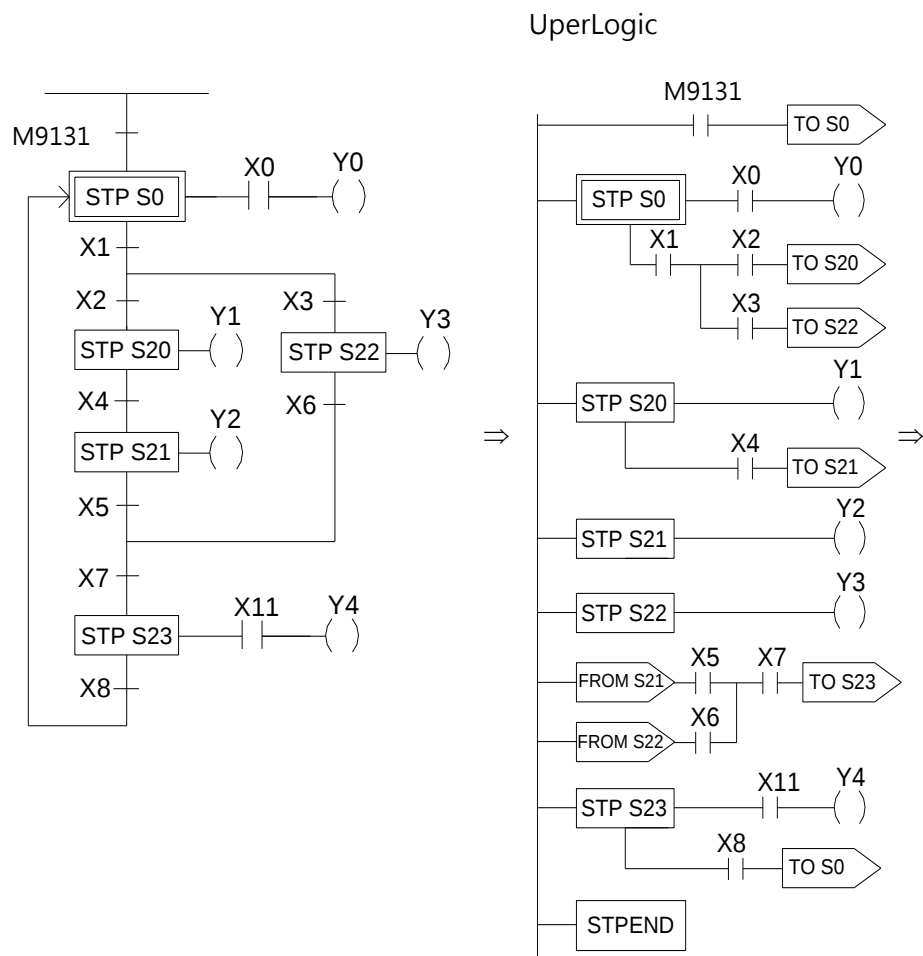


ST

- 【說明】：
- 1.編輯初始步進點 S0
 - 2.編輯 S20、S0、S21 之分歧
 - 3.編輯 S20
 - 4.折返編輯 S21
 - 5.編輯 S20、S21 之合流
 - 6.往下編輯 S22

【範例 2】

階段圖

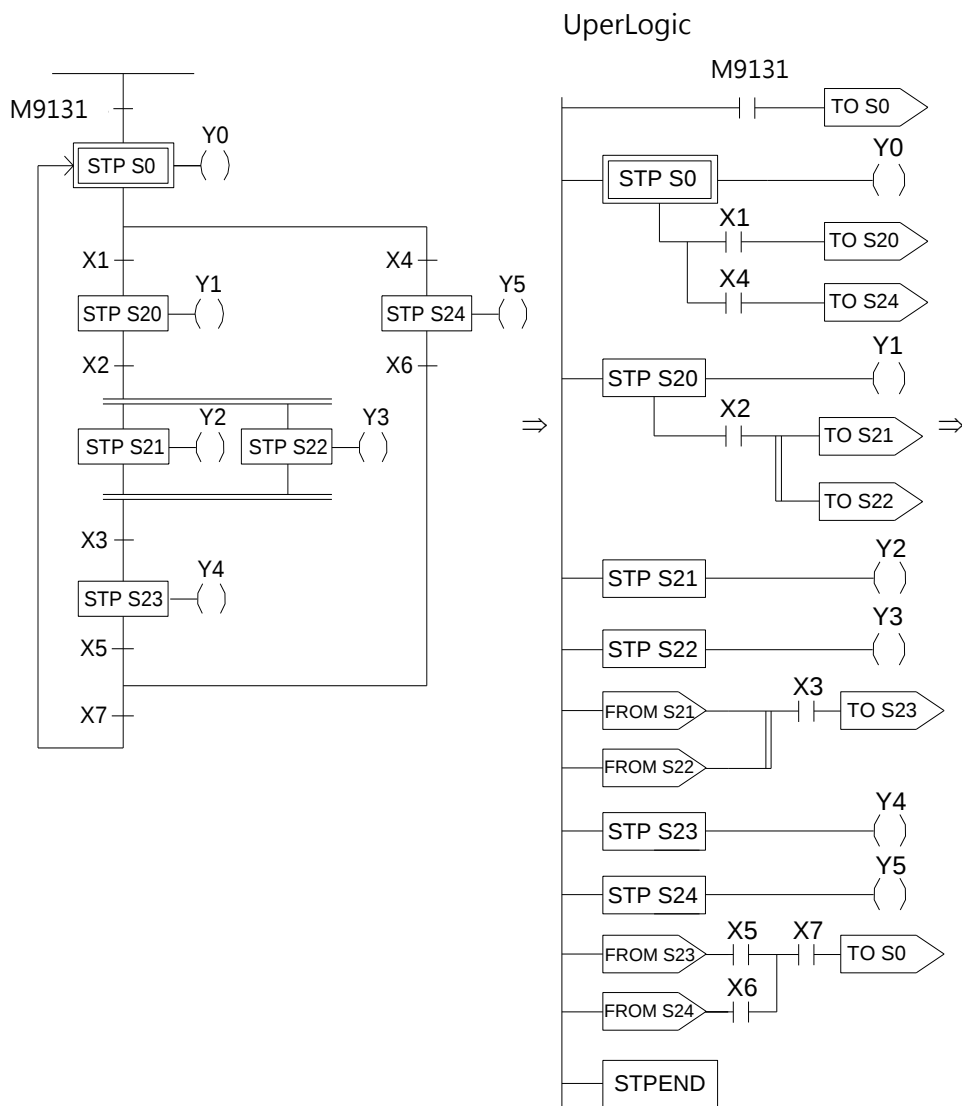


ST

- 【說明】：
- 1.編輯初始步進點 S0
 - 2.編輯 S20、S22 之分歧
 - 3.編輯 S20
 - 4.編輯 S21
 - 5.折返編輯 S22
 - 6.編輯 S21、S22 之合流
 - 7.往下編輯 S23

【範例 3】

階段圖

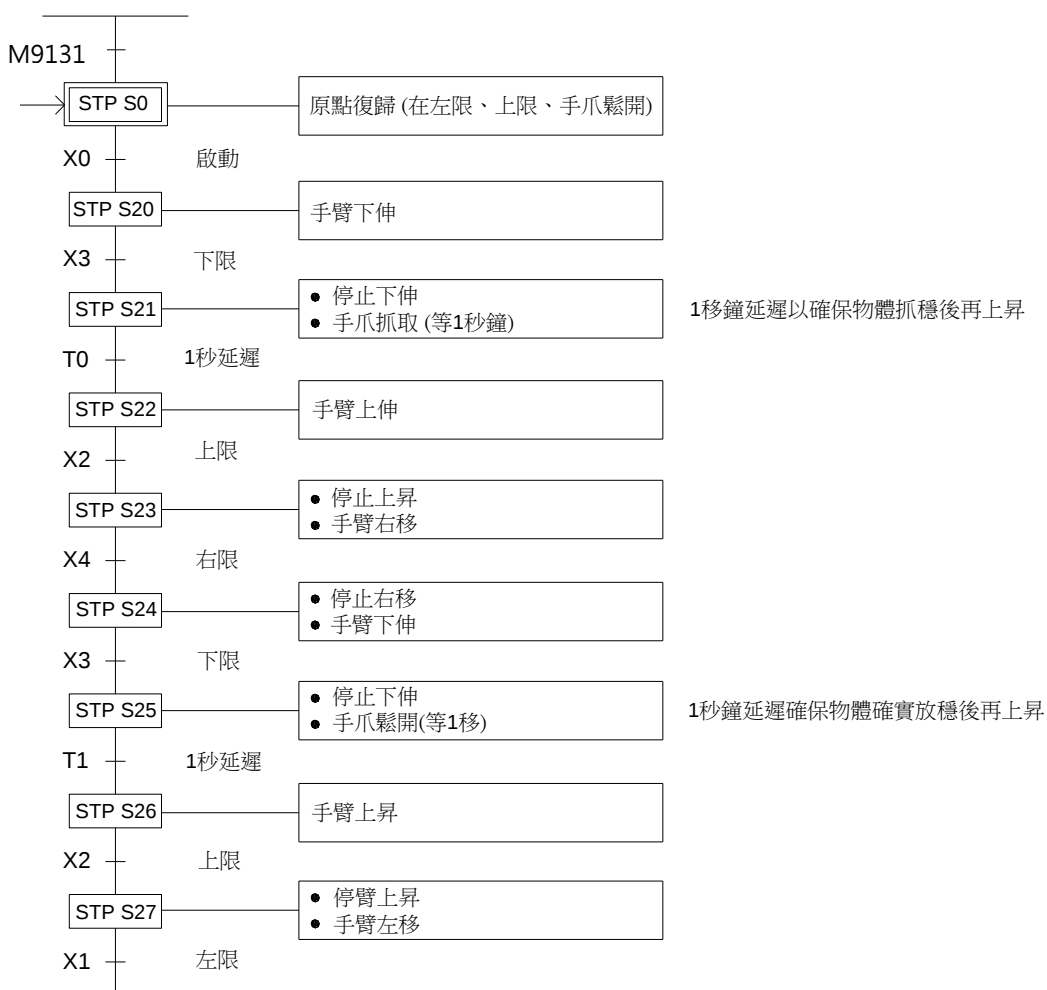
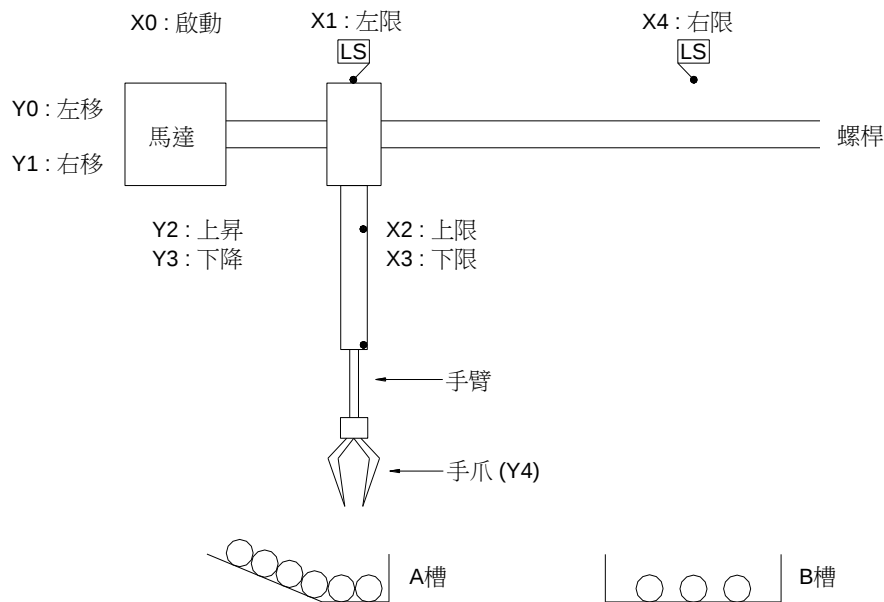


ST

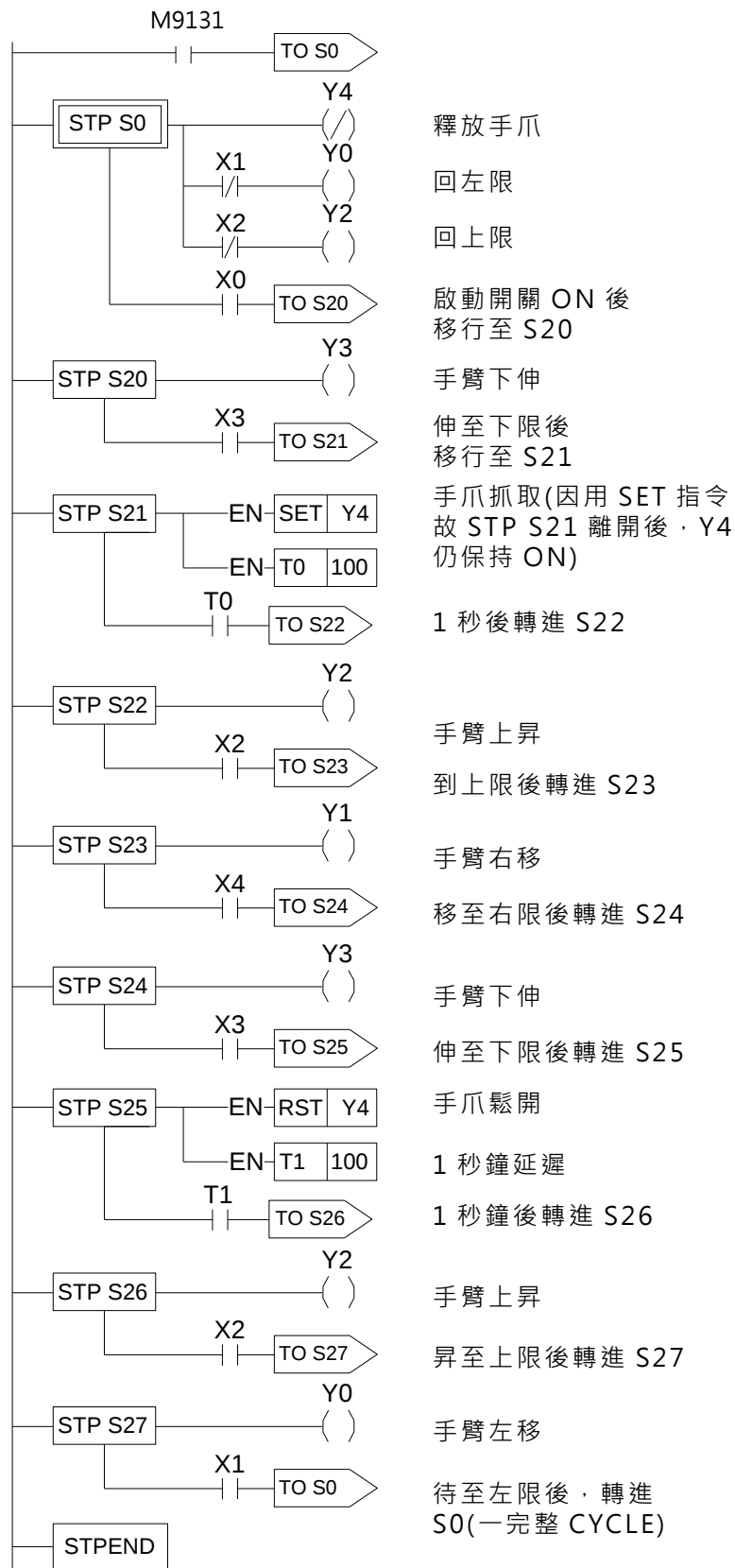
- 【說明】：
- 1.編輯初始步進點 S0
 - 2.編輯 S20、S24 之分歧
 - 3.編輯 S20
 - 4.編輯 S21、S22 之分歧
 - 5.編輯 S21
 - 6.折返編輯 S22
 - 7.編輯 S21、S22 之合流
 - 8.編輯 S23
 - 9.返回上層編輯 S24
 - 10.編輯 S23、S24 之合流

8-5 實際應用範例

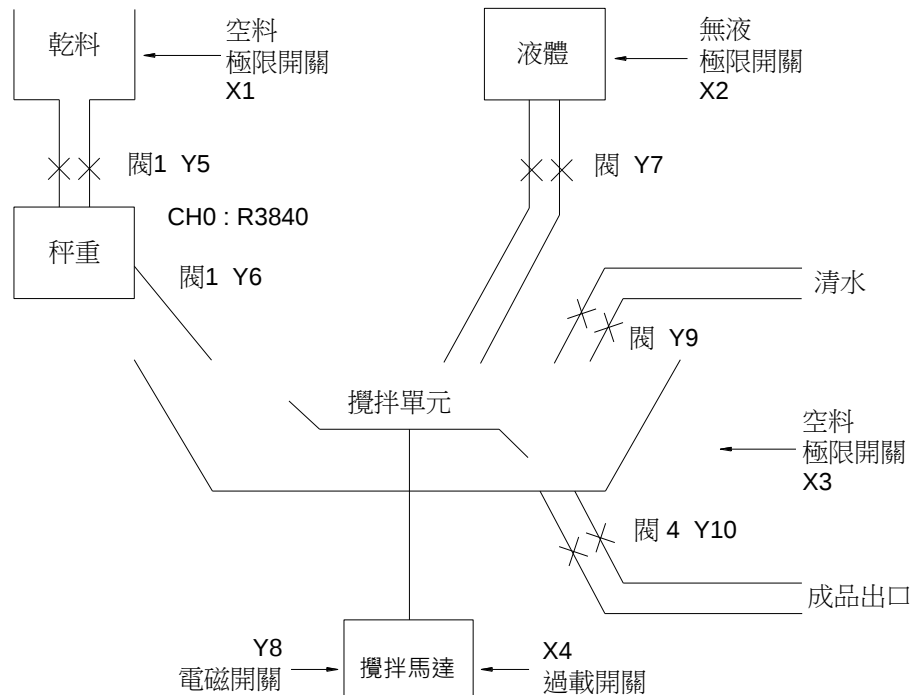
【範例 1】自 A 槽抓取物體放至 B 槽內



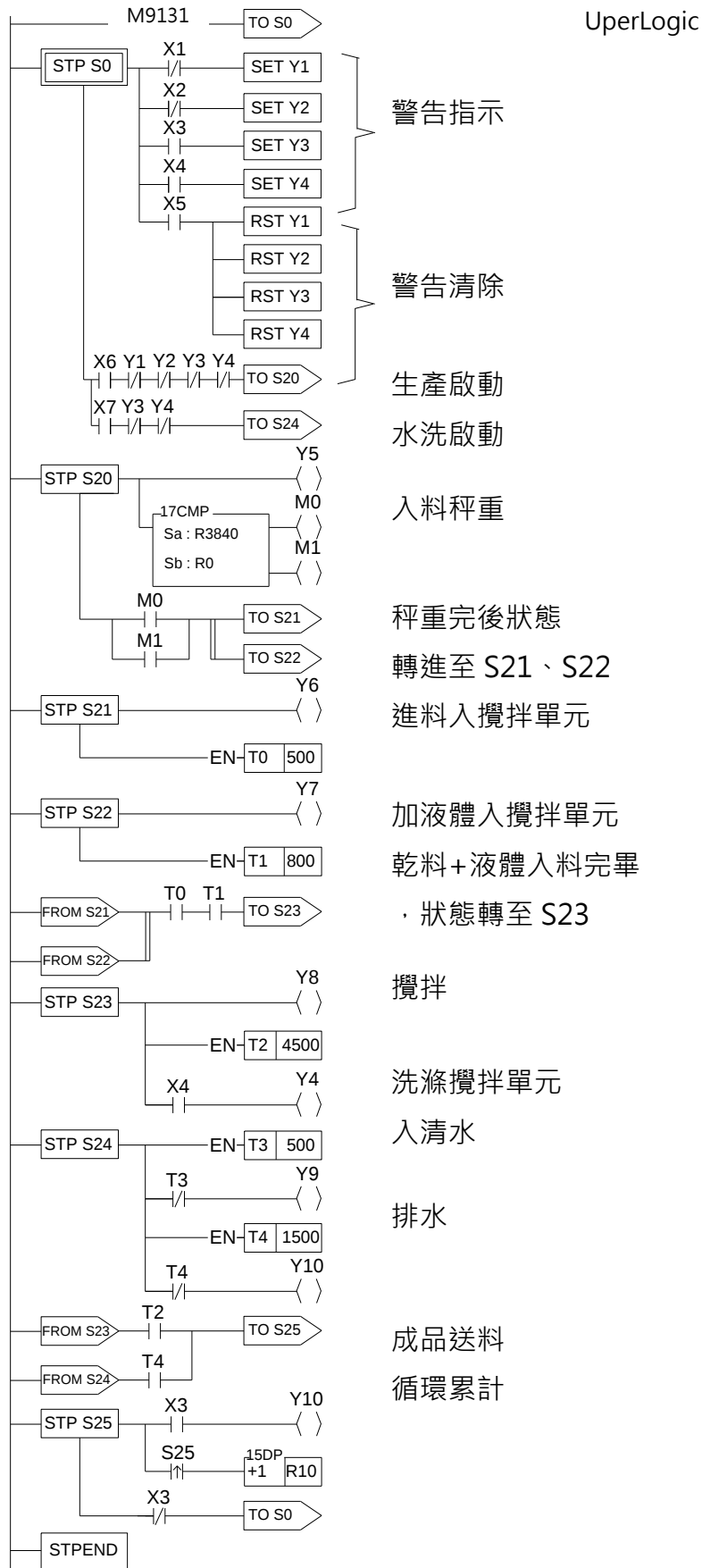
UperLogic



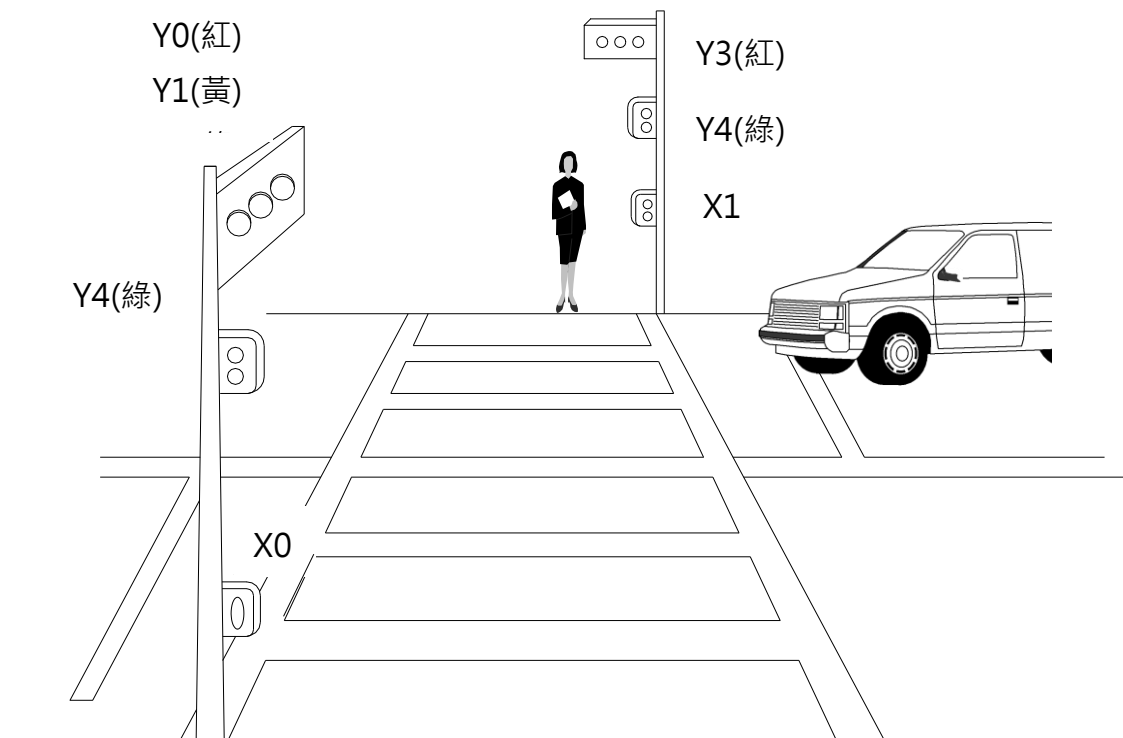
【範例 2】液體攪拌處理



- ◆ 輸 入 點：
 - 空料極限開關 X1
 - 無液極限開關 X2
 - 空料極限開關 X3
 - 過載開關 X4
 - 警告清除鈕 X5
 - 啟動鈕 X6
 - 水洗鈕 X7
- ◆ 警告指示燈：
 - 乾料空料 Y1
 - 液體缺液 Y2
 - 攪拌單元空料 Y3
 - 馬達過載 Y4
- ◆ 輸 出 點：
 - 乾料入料閥 Y5
 - 乾料入料閥 Y6
 - 液體入料閥 Y7
 - 啟動馬達電磁閥 Y8
 - 清水入水閥 Y9
 - 成品送料閥 Y10
- ◆ 秤 重 輸 入：CH0 (R3840)



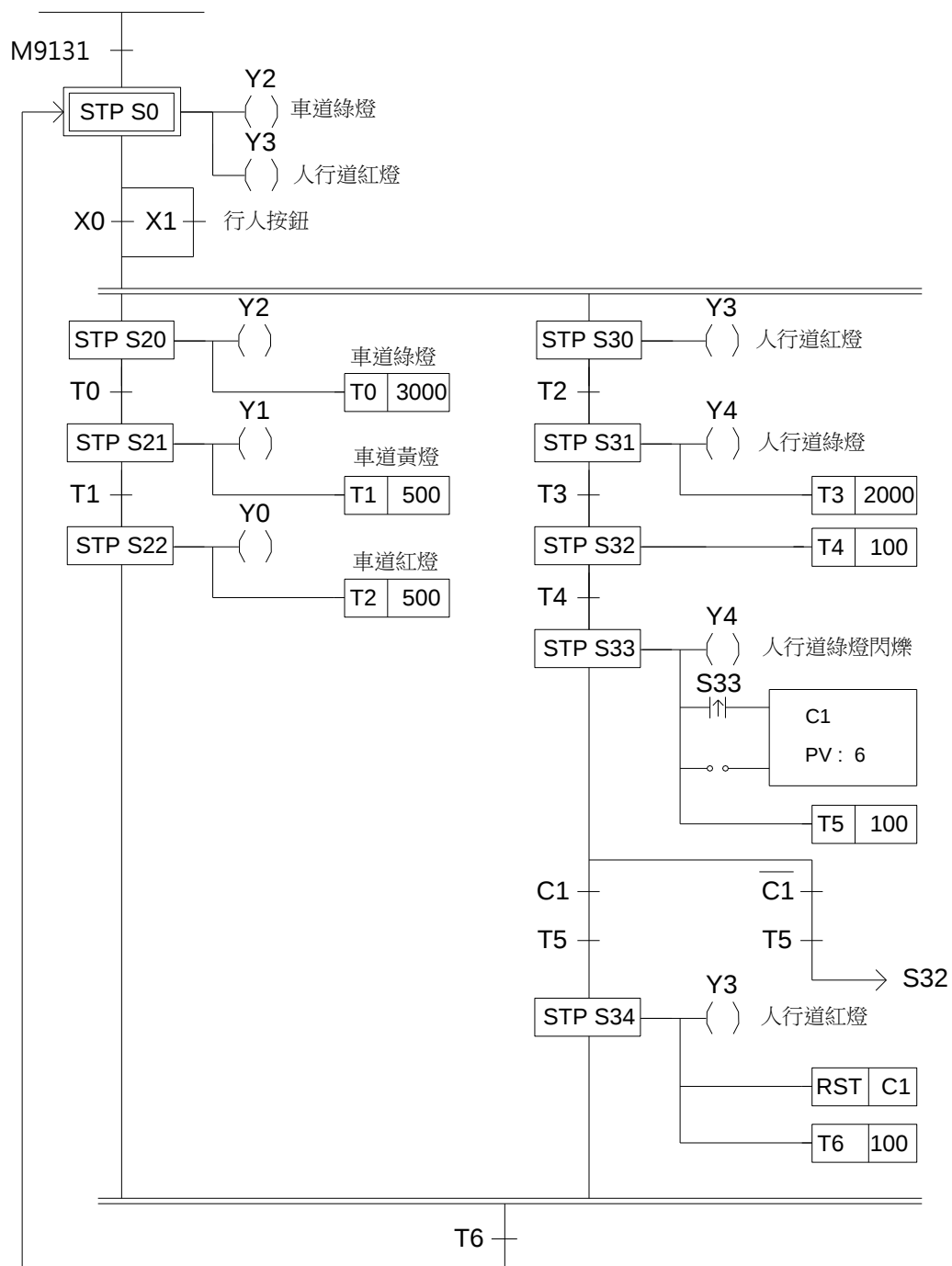
【範例 3】人行道紅綠燈



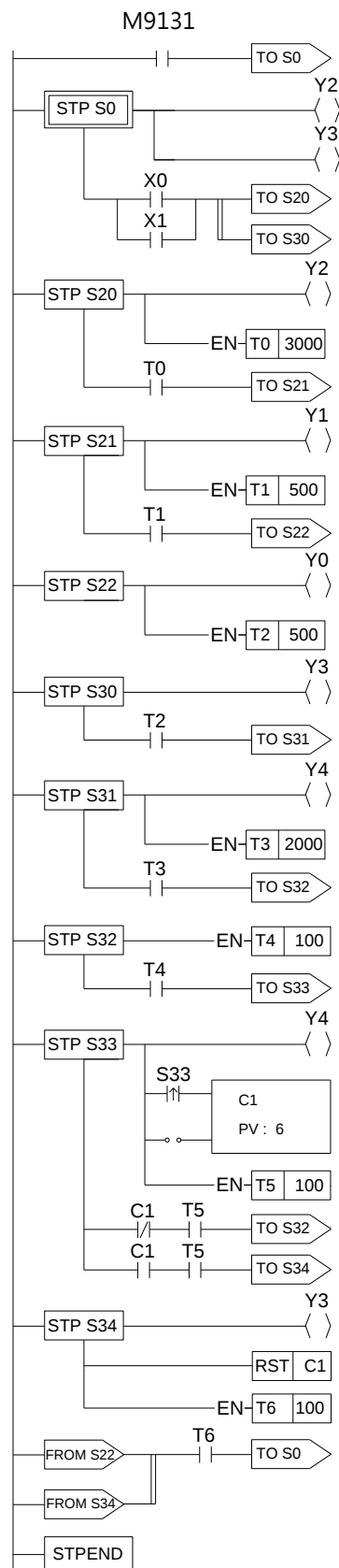
◆ 輸入點： 行人按鈕 X0
行人按鈕 X1

◆ 輸出點： 車道紅燈 Y0
車道黃燈 Y1
車道綠燈 Y2
人行道紅燈 Y3
人行道綠燈 Y4

人行道紅綠燈控制流程圖



● 人行道紅綠燈控制程式 UperLogic



步進程式語法檢查錯誤碼說明

步進語言程式、語法檢查錯誤之編號如下：

- E51 : TO(S0~S20)必需以 ORG 為起始指令
- E52 : TO(S20~S3103)不得以 ORG 為起始指令
- E53 : 同一網路中，TO(S20~S3103)前，必需在有 FROM
- E54 : TO 之前一指令，必需為 TO、AND、OR、ANDLD、ORLD
- E56 : 此時 FROM 之前一指令，必需為 FROM 或 AND、OR、ANDLD、ORLD
- E57 : OUT、TMR、CTR、FUN 不與 TO(S0~S19)並存於同一網路中
- E58 : OUT、TMR、CTR、FUN 前一 STEP 指令必需為 STP
- E59 : 同一網路中，TO 超過 8 個
- E60 : 同一網路中，FROM 超過 8 個
- E61 : TO(S0~S19) 必需為網路第一列
- E62 : 接點佔據 TO 位置
- E71 : 連繫不完整(理應不會發生)
- E72 : TO Sxxxx 重複
- E73 : STP Sxxxx 重複
- E74 : FROM Sxxxx 重複
- E76 : 上一個 STP(S0~S19)缺乏 STPEND 或 STPEND 往前找不到相對應之 STP(S0~S19)
- E77 : STP(S0~S19)的前一個網路並非是以 ORG 為起始唯一之 TO(S0~S19)
- E78 : 尚未使用 STP(S0~S19)就使用 TO(S20~S3103)、STP (S20~S3103)、FROM
- E79 : 尚未使用 TO Sxxxx 就使用 STP Sxxxx 或 FROM Sxxxx
- E80 : 尚未使用 STP Sxxxx 就使用 FROM Sxxxx
- E81 : 同一時間，尚未處理之分歧層數不得大於 16
- E82 : 同一時間，分歧中尚未處理之分枝不得大於 16
- E83 : 單一步進點，未依照 TO Sxxxx→STP Sxxxx→FROM Sxxxx 之順序且緊密連繫
- E84 : 進入分歧後，需依照由左至右之順序來處理分枝
- E85 : 合流時，與先前之分歧情況不對應
- E86 : 尚未利用 TO 來完成上一個合流，就使用 STP 或 FROM
- E87 : 尚未利用 FROM+TO 來轉移上一個 STP，就使用 STP 或 FROM
- E88 : 分歧中，STP Sxxxx 或 FROM Sxxxx，在此分歧內，往前找不到相對應之 TO Sxxxx
- E89 : 尚未利用 STP 來承接 TO 之處理，就使用 FROM
- E90 : 並進式分歧之轉接不合法
- E91 : 上一個 STP(S0~S19)尚未處理完全，就使用 ORG、LBL、RTS、RTI、MCE、SKPE、
FOR、NEXT、END

修訂紀錄

版本	修訂日期	修訂內容	頁數	修改者
V0.0.01	2020/11/02	第一版		
V0.0.02	2020/12/07	第二版		
V0.0.02	2021/01/05	第三版		
V1.0	2021/04/26	第四版	502	