

## 目錄

第一章：機種構成及規格

第二章：基本指令

第三章：步階指令

第四章：各種要素功能細述

第五章：應用命令

第六章：特殊暫存器與資料暫存器

附錄 A 通訊介面 RS422 腳點陣圖

附錄 B 故障排除方法及異常碼一覽表

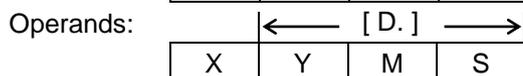
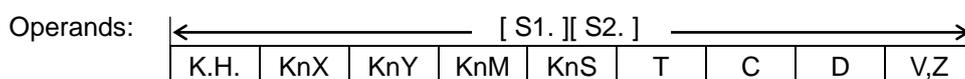
## 應用命令通則

- ◆ 應用命令被設計成以功能數位表示( FNC 00-99 )每一應用命令被冠以一個”符號”。  
例：FNC 12 被冠以“MOV”
- ◆ 有些應用命令的格式僅須指定 FNC 數字的部份，有些須再加入運算元。

## COMPARE

FNC(10)			16 bits: CMP & CMP(P) -----7 steps			J2n--	J3n--
D	CMP	P	32 bits:(D)CMP & (D)CMP(P) -----13 steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●		●	●			●	●		
[S2.]								●	●	●	●	●	●	●		●	●			●	●		
[D.]		●	●	●																			



[S.]:代表一個運算元，其被稱為來源”Source”運算元，如果”Source”超過一個時，則表示成[S1.] [S2.]等。

[D.]:代表一個運算元，其被稱為目的”Destination”運算元，如果”Destination”超過一個時，則表示成[D1.] [D2.] 等。

- ◆ 指令部份佔用一個程式步序，運算元則佔用 1-2 程式步序，佔用程式步序的多寡完全由指令是 16 位元或 32 位元指令來決定。

- ◆ X,Y,M,S,等位要素(bit device)亦可當作運算元處理，經組合後，以字元要素(word device)的形態表示。
- ◆ 資料暫存器(D)皆為 16 位，當處理 32 位元資料時將自動成對使用。  
例:指定 D0 執行 32 位運算時，將會取(D1,D0)的資料來做處理(D1 為上 16 位，D0 為下 16 位)。
- ◆ 32 位的計數器則不能當作 16 位命令的運算元。

## 資料長度及命令執行格式

- ◆ 32 位元的指令被表示成在指令前加入(D)。例:(D) MOV
- ◆ 不管指令的要素號碼是偶數還是奇數皆可使用，但為避免混亂儘量使用偶數要素號碼來配合 32 位使用。

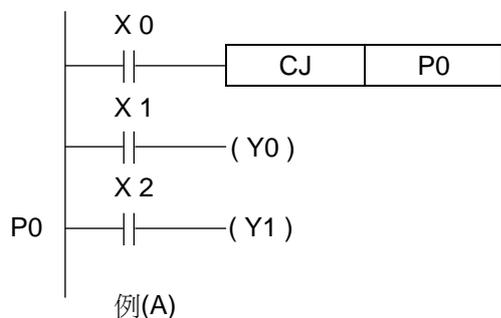
## 位元要素與字元要素

- ◆ 諸如 X,Y,M 及 S 等要素僅能以 ON/OFF 狀態表示，因此被稱為”位要素”，其他要素如 T,C 及 D 等要素能表示數值資料，因此被稱為”字元要素”，但對於 X,Y,M,S 組成的位元要素亦可以字元的狀態來表現，只要在前面加上指定位數的 Kn 符號即可。
- ◆ 位元要素能組合以 4 個位為單位的要素，K1~K4 允許 16 位元資料運算，K1~K8 允許 32 位元資料運算。  
例:K2M0 表示由 M0 開始 2 個 4 位元為單位的要素，即 M0~M7 組合成的字元。

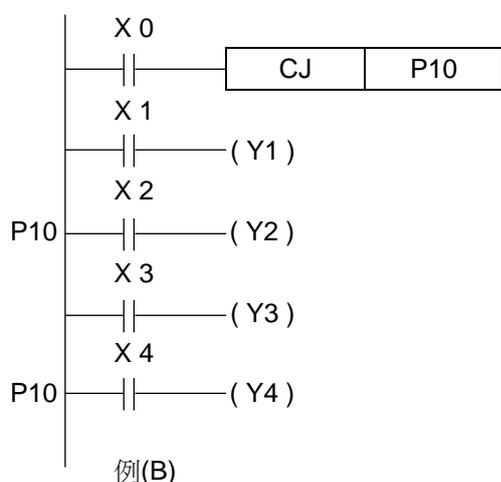
## FNC(00)條件跳躍 CONDITION JUMP

FNC(00)		16 bits: CJ & CJ(P) ----- 3 Steps				J2n--	J3n--
CJ	P						

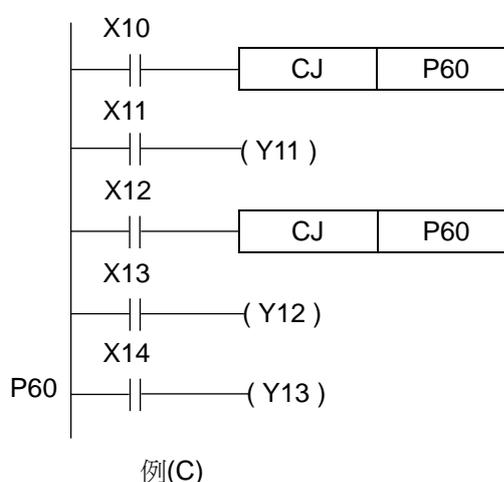
Operand: P00 ~ P63



例(A)



例(B)



例(C)

- ◆ 使用 CJ(P) 隔開部分程式，能減少演算週期且允許雙重線圈線路。
- ◆ 例 (A): 假如 X0 ON 強迫程式跳至 LAB P0，被隔開的指令將不被執行，即使輸入條件動作，輸出狀態亦不會改變。
- ◆ 例 (A): 假如未寫 LAB P0，X0 ON 則直接跳至 END。
- ◆ 當使用逆向跳躍時，須注意 watchdog timer overrun。
- ◆ 假如 P 標記號碼重複時，僅最後的標記號碼有效。
- ◆ 例 (B): X0 ON 則程式跳至第二 LAB P10。
- ◆ 例 (C): 多個 CJ(P) 指令可指定跳至同一個標記號碼。

## FNC(01)呼叫副程式 Subroutine Call

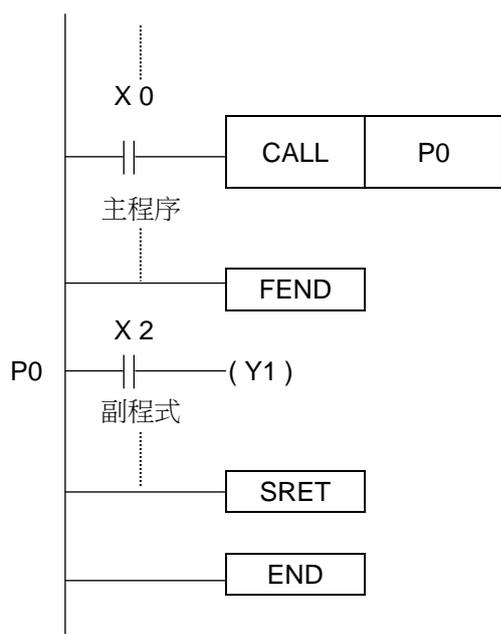
FNC(01)		16 bits: CALL & CALL(P) ----- 3 Steps			J2n--	J3n--
CALL	P					

Operand: P00~P63

## FNC(02)副程式返回 Subroutine Return

FNC(02)		16 bits: SRET ----- 1 Steps			J2n--	J3n--
SRET						

Operand: None



◆ 當 X0 ON，程式會跳至 LAB P0 執行副程式，直到 SRET 被執行，則程式返回至主程序繼續執行。

◆ 副程式必須寫在 FEND 之後。

◆ 多個 CALL 指令可指定至同一個標記號碼。

◆ 在同一個 CALL 指令下最多四層的副程式被使用。

◆ 副程式之後須寫 SRET 指令。

## FNC(03)中斷返回 Interrupt Return

FNC(03)	16 bits: IRET ----- 1 Steps			J2n--	J3n--
IRET					

Operand: None

## FNC(04)中斷致能 Enable Interrupt

FNC(04)	16 bits: EI ----- 1 Steps			J2n--	J3n--
EI					

Operand: None

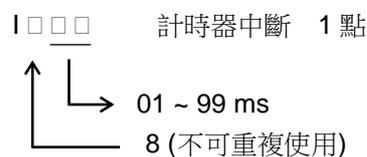
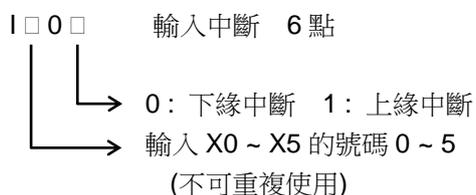
## FNC(05)中斷禁止 Disable Interrupt

FNC(05)	16 bits: DI ----- 1 Steps			J2n--	J3n--
DI					

Operand: None



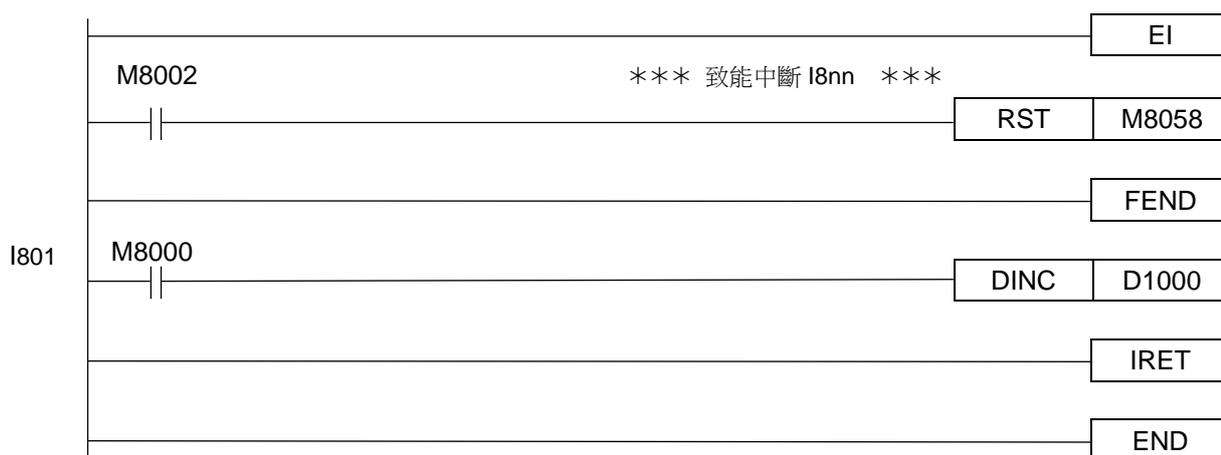
## 中斷指針號碼



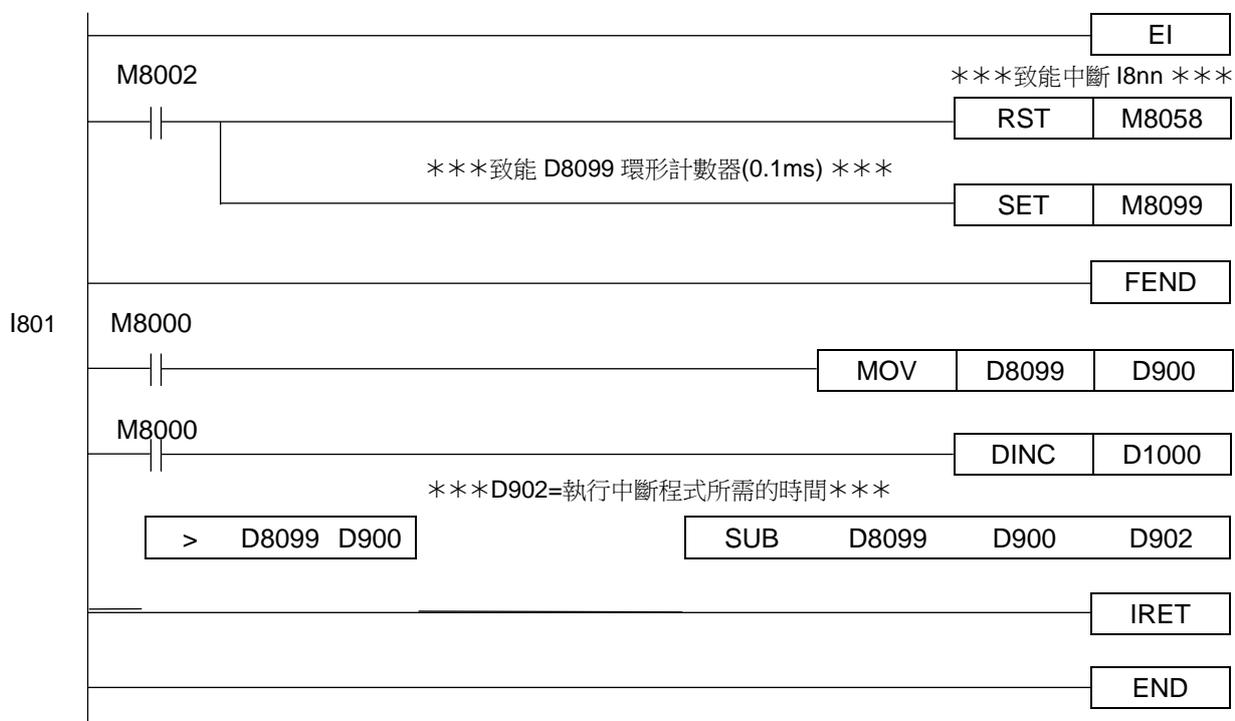
### <<注意事項>>

- ◆ 當一中斷程式執行時，其他中斷呼叫視為無效。
- ◆ 假如中斷發生在中斷禁止範圍內( DI ~ EI )時，這中斷要求信號暫時被儲存，待中斷致能範圍內( EI ~ DI )再執行。
- ◆ 中斷禁止旗標 M805n 動作時(n=0 ~ 5)，相對應的中斷輸入將不被執行。
- ◆ 中斷程式內不可使用 FNC(50) REF 指令。(如上述範常式式中的 A 區段)

## 時間中斷範例



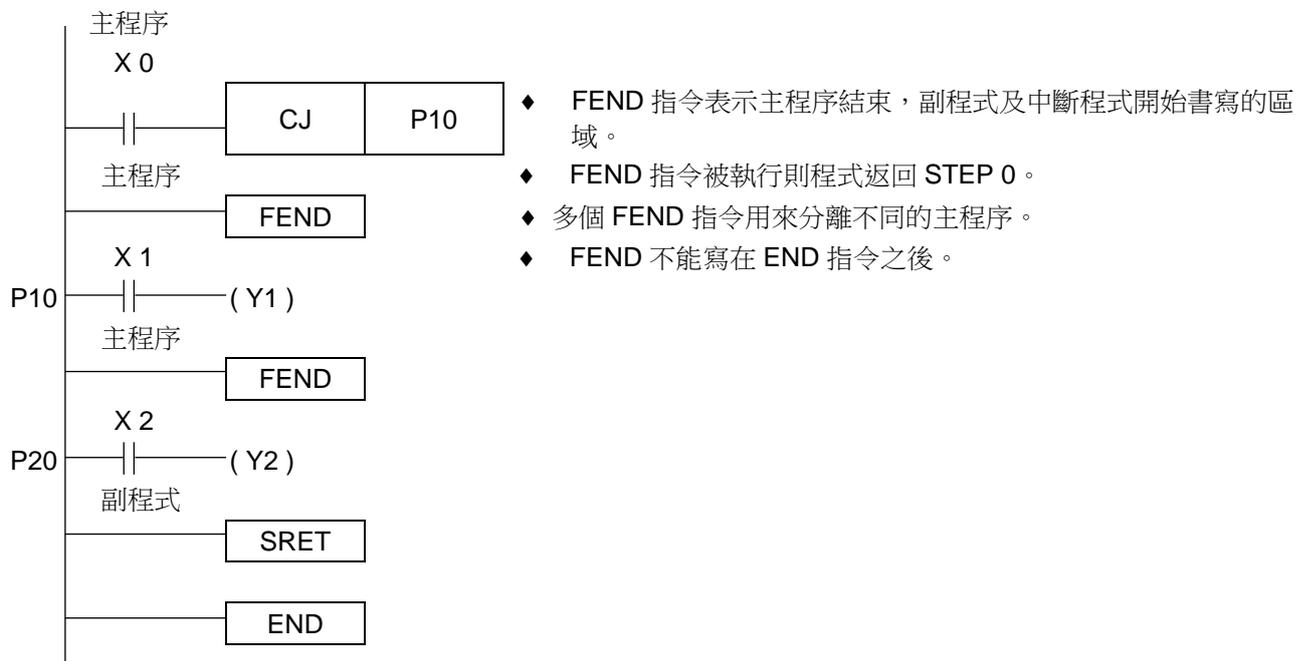
## 計算時間中斷程式執行時間範例



## FNC(06)主程序結束 First End

FNC(06)		16 bits: FEND ----- 1 Steps			J2n--	J3n--
	FEND					

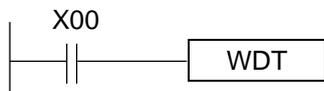
Operand: None



## FNC(07)計時器 Watch Dog Timer

FNC(07)		16 bits: WDT ----- 1 Steps				J2n--	J3n--
	WDT	P					

Operand: None



- ◆ 此命令主要在檢查掃描週期的時間是否大於特殊資料暫存器 D8000 的內容值。
- ◆ 若大於 D8000 的內容值，則會產生錯誤，錯誤碼 6309。
- ◆ 特殊資料暫存器 D8000 的內容值可利用 MOV 命令來變更。
- ◆ 未寫入此命令則不執行時間的比對。

## FNC(08)重複起始點 FOR

FNC(08)		16 bits: FOR ----- 3 Steps																J2n--	J3n--		
	FOR																				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●		●	●				●	●	

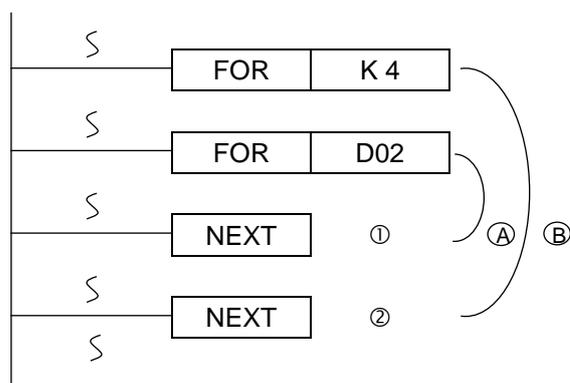
Operands: ← [ S. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

## FNC(09)重複結束點 NEXT

FNC(09)		16 bits: NEXT ----- 1 Steps																J2n--	J3n--		
	NEXT																				

Operand: None



- ◆ 程式 B 被執行 4 次後，再執行第②個 NEXT 以下的程式。
- ◆ 若資料暫存器 D0Z 中的值為 5，則程式 B 每一次執行時，程式 A 將執行 5 次，也就是說程式 A 總共要執行 20 次。
- ◆ FOR-NEXT 回路至多 5 層。

## FNC(10)比較 COMPARE

FNC(10)			16 bits: CMP & CMP(P) ----- 7 Steps			J2n--	J3n--
D	CMP	P	32 bits:(D)CMP&(D)CMP(P) ----- 13 Steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●		●	●			●	●		
[S2.]								●	●	●	●	●	●	●		●	●			●	●		
[D.]		●	●	●																			

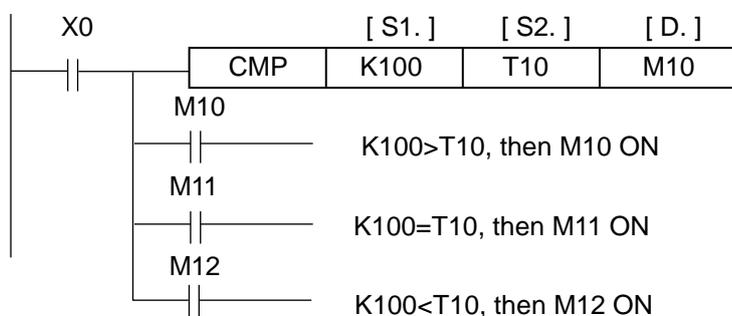
Operands: ← [ S1. ][ S2. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

Operands: ← [ D. ] →

X	Y	M	S
---	---	---	---

影響旗號:



- ◆ 來源[S1.]與[S2.]內的資料互相比較，[D.]則依據比較的結果產生變化。本指令目的位元將自動佔據 3bits，本例為 (M10-M12)。
- ◆ 來源資料以代數方式做比較。例:  $-10 < 2$ 。
- ◆ 當 X0 OFF，則[D.]位元狀態不改變。

## FNC(11)區域比較 ZONE COMPARE

FNC(11)			16 bits: ZCP & ZCP(P) ----- 9 Steps			J2n--	J3n--
D	ZCP	P	32 bits: (D)ZCP&(D)ZCP(P) -----17 Steps				

元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[S1.]								●	●	●	●	●	●	●		●	●			●	●	
[S2.]								●	●	●	●	●	●	●		●	●			●	●	
[S3.]								●	●	●	●	●	●	●		●	●			●	●	
[D.]		●	●	●																		

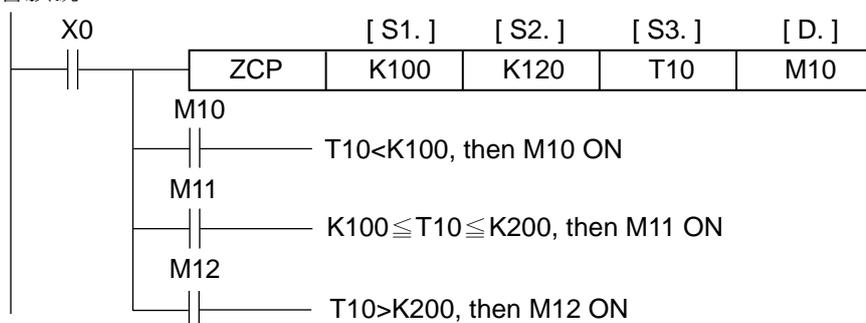
Operands: ← [ S1. ] [ S2. ] [ S3. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

Operands: ← [ D. ] →

X	Y	M	S
---	---	---	---

影響旗號:

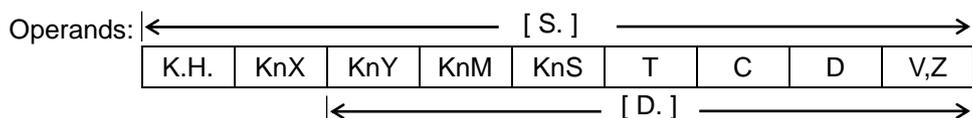


- ◆ 來源[S3.]內的資料與來源[S1.]及[S2.]內的資料範圍做比較，[D.]則依據比較的結果產生變化。本指令目的位元將自動佔據 3bits，本例為(M10-M12)。
- ◆ 請設定 [S1.] ≤ [S2.]。若 [S1.] > [S2.] 時，則 [S2.] 的值視為與 [S1.] 的數值相同。
- ◆ 來源資料以代數方式做比較。例: -10 < 2。
- ◆ 當 X0 OFF，則[D.]位元狀態不改變。

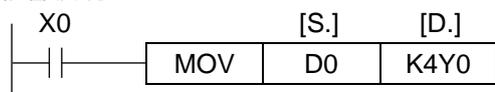
## FNC(12)搬移 MOVE

FNC(12)			16 bits: MOV & MOV(P) ----- 5 Steps			J2n--	J3n--
D	MOV	P	32 bits:(D)MOV&(D)MOV(P) ----- 9 Steps				

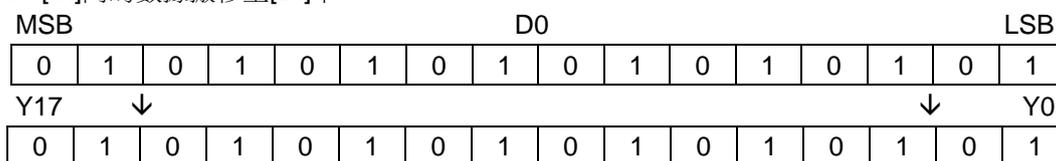
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]								●	●	●	●	●	●	●		●	●			●	●		
[D.]									●	●	●	●	●	●		●	●						



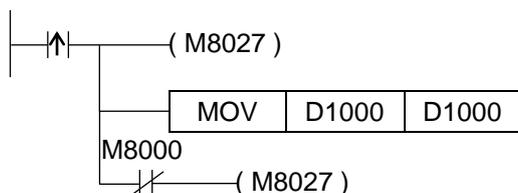
影響旗號:



- ◆ 當 X0 ON，[S.]內的數據搬移至[D.]中。



- ◆ 當 M8027 ON 時，系統同時將 [S.] 之資料寫入 EEPROM 相對應之 [D.] 中。



注意事項：當 M8027 ON 時，為避免破壞 EEPROM，請使用脈波指令且目的字元僅 D 暫存器有效。

## FNC(13)移位搬移 Shift Move

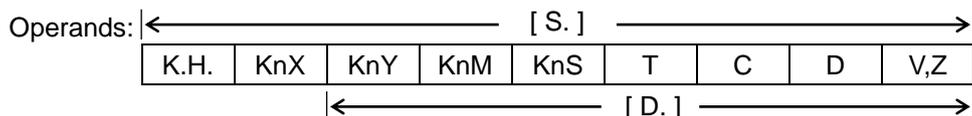
FNC(13)			16 bits: SMOV & SMOV(P) ----- 7 Steps				
	SMOV	P					

Reserved

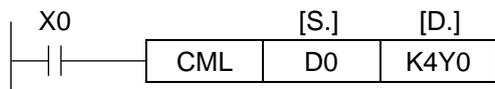
## FNC(14)互補 COMPLEMENT

FNC(14)			16 bits: CML & CML(P) ----- 5 Steps			J2n--	J3n--
D	CML	P	32 bits: (D)CML & (D)CML(P) -----9 Steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●		●	●						



影響旗號:



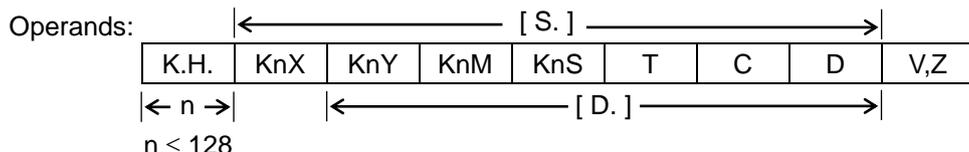
◆ [S.] 的每一位數值被反相 (0→1,1→0) 搬移至 [D.] 中。

MSB	D0														LSB	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Y17	↓														Y0	
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

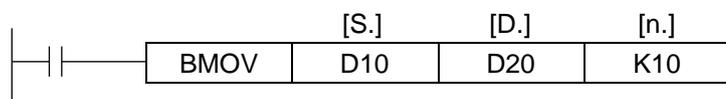
## FNC(15) 區塊搬移 BLOCK MOVE

FNC(15)		16 bits: BMOV & BMOV(P) ----- 7 Steps																	J2n--	J3n--
BMOV	P																			

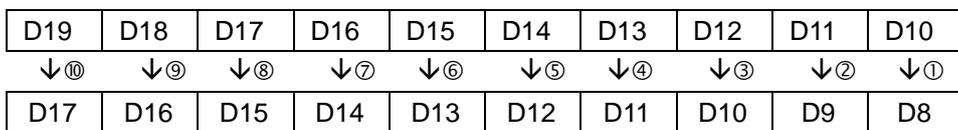
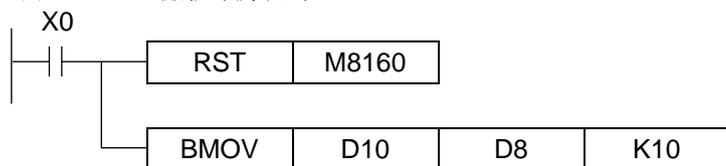
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●									
[D.]									●	●	●	●	●	●									
[n.]																					●	●	



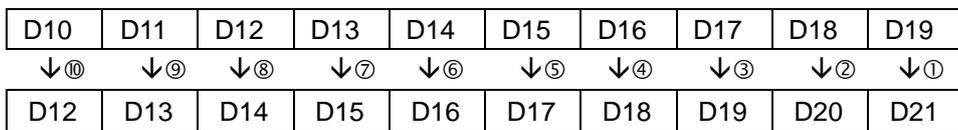
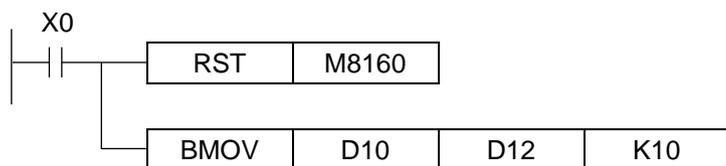
影響旗號: 無



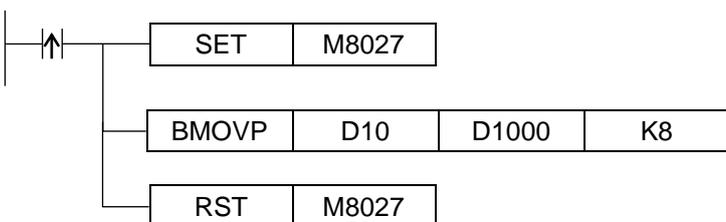
◆當 X0 ON，搬移順序如下



◆當轉送編號重複時，搬移順序如下



◆當 M8027 ON 時，系統同時將[S.]區塊資料寫入 EEPROM 相對應之[D.]中，僅 D 暫存器有效。

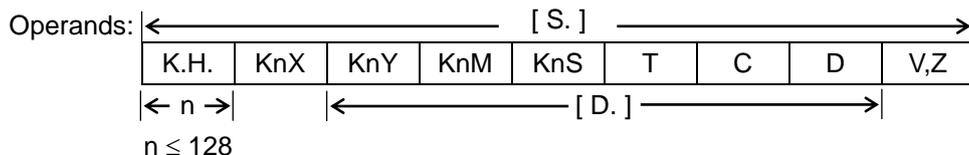


注意事項：當 M8027 ON 時，為避免破壞 EEPROM，請使用脈波指令且目的字元僅 D 暫存器有效。

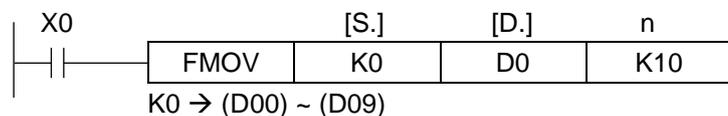
### FNC(16)多點搬移 FILL MOVE

FNC(16)			16 bits: FMOV & FMOV(P) ----- 7 Steps			J2n--	J3n--	
D	FMOV	P	32 bits: (D)FMOV & (D)FMOV(P) -----13 Steps					

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●									
[n.]																					●	●	



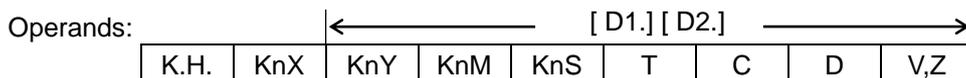
影響旗號:



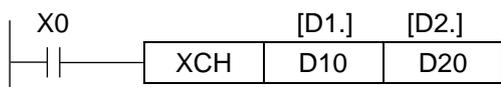
## FNC(17) 互換 EXCHANGE

FNC(17)			16 bits: XCH & XCH(P) ----- 5 Steps			J2n--	J3n--
D	XCH	P	32 bits: (D)XCH & (D)XCH(P) ----- 9 Steps				

元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[D1.]									●	●	●	●	●	●		●	●						
[D2.]									●	●	●	●	●	●		●	●						

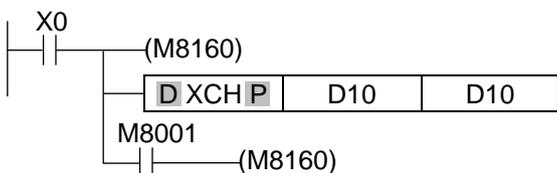


影響旗號:

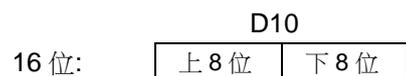


執行前 : (D10)=100      執行後 : (D10)=200  
 (D20)=200              (D20)=100

&lt;&lt; 擴充機能 &gt;&gt; SWAP



- ◆ M8160 為 ON，[D1.]與[D2.]又為同一要素時，將會執行上 8 位元與下 8 位元資料的交換。
- ◆ 若[D1.]與[D2.]不為同一要素時，錯誤旗標 M8067 ON，錯誤碼 6705，錯誤步序存入 D8069，且命令不執行。



執行前(D10)=0050H=80，執行後(D10)=5000H=20480

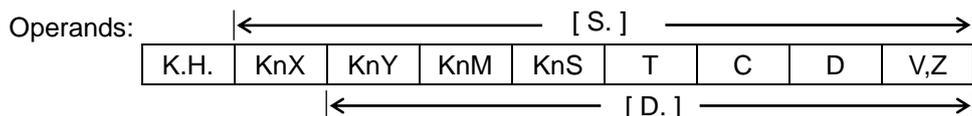


執行前(D11,D10)=87654321H=80，執行後 65872143H

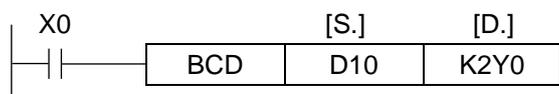
## FNC(18)BCD 變換 (BINARY CODE TO DECIMAL)

FNC(18)			16 bits: BCD & BCD(P) ----- 5 Steps			J2n--	J3n--
D	BCD	P	32 bits: (D)BCD & (D)BCD(P) ----- 9 Steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]								●	●	●	●	●	●	●		●	●						
[D.]									●	●	●	●	●	●		●	●						



影響旗號:

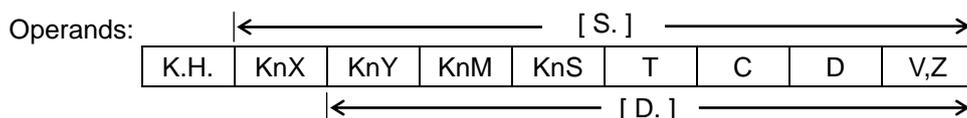


- ◆ 來源 [S.] 二進位 (BIN) 的資料轉換成十進位 (BCD) 存入 [D.] 。
- ◆ 假如十進位 BCD 的數值超過 0 - 9999 (16 bits operation)或 0 - 99999999 (32 bits operation)則錯誤旗標 M8067ON，錯誤代碼 6705，錯誤步序存入 D8069，程式繼續執行，但演算結果不存入[D.]中。
- ◆ 此指令可用來驅動七段顯示器。

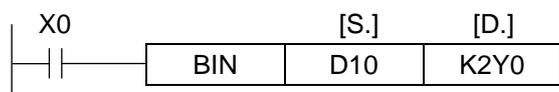
## FNC(19)BIN 變換 (DECIMAL CODE TO BINARY)

FNC(19)			16 bits: BIN & BIN(P) ----- 5 Steps			J2n--	J3n--
D	BIN	P	32 bits: (D)BIN & (D)BIN(P) ----- 9 Steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●		●	●						
[D.]									●	●	●	●	●	●		●	●						



影響旗號:

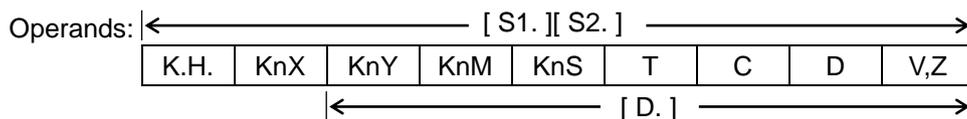


- ◆ 來源[S.]十進位(BCD)的資料轉換成二進位(BIN)存入[D.]。
- ◆ 若來源[S.]資料不符合 BCD 的格式，則錯誤旗標 M8067 ON，錯誤代碼 6705，錯誤步序存入 D8069，程式繼續執行，但不執行演算。
- ◆ 來源[S.]不可指定為常數 K/H。

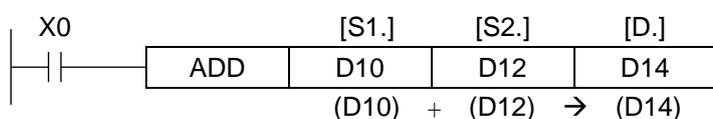
## FNC(20)加法 ADDITION

FNC(20)			16 bits: ADD & ADD(P) ----- 7 Steps			J2n--	J3n--
D	ADD	P	32 bits: (D)ADD & (D)ADD(P) -----13 Steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S1.]								●	●	●	●	●	●	●		●	●				●	●	
[S2.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●		●	●						



影響旗號: M8020, M8021, M8022

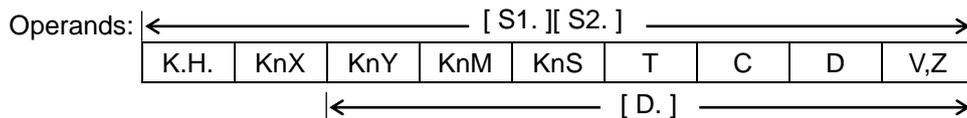


- ◆ 來源[S1.]與[S2.] BIN 資料相加結果存入[D.]中。
- ◆ 所有運算均以代數進行，i.e. 5+(-8) = -3.
- ◆ 最高位代表正負號 (0:正，1:負)。
- ◆ 運算結果等於“0”，則零位旗標 M8020 ON。
- ◆ 運算結果大於 32,767 (16 bit operation) 或 3,147,483,647 (32 bit operation)，則進位旗標 M8022 ON。
- ◆ 運算結果小於 -32,767 (16 bit)或 -2,147,483,647 (32 bit)，則借位旗標 M8021 ON。

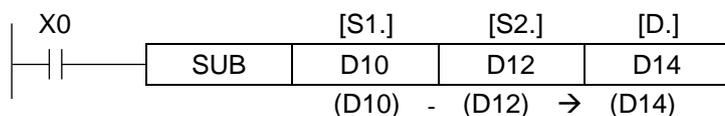
## FNC(21)減法 SUBTRACTION

FNC(21)			16 bits: SUB & SUB(P) ----- 7 Steps			J2n--	J3n--
D	SUB	P	32 bits: (D)SUB & (D)SUB(P) ----- 13 Steps				

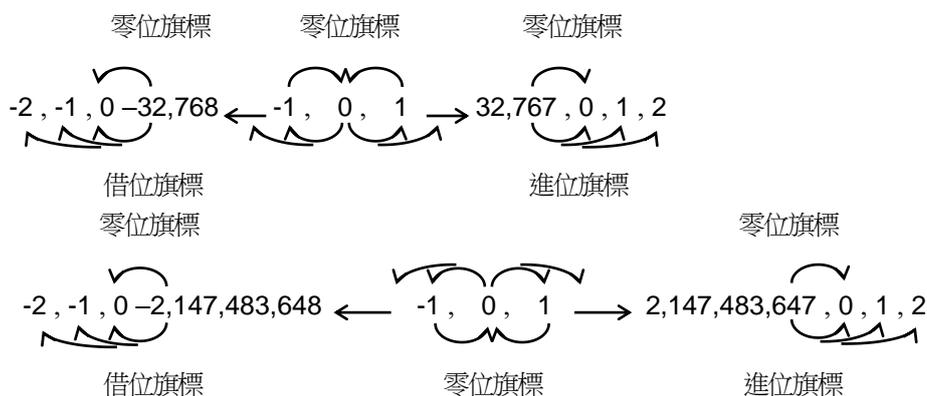
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S1.]								●	●	●	●	●	●	●		●	●				●	●	
[S2.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●		●	●						



影響旗號: M8020, M8021, M8022



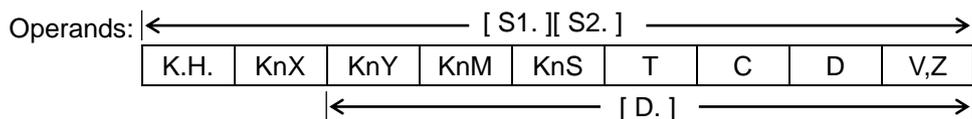
- ◆ 來源 [S1.] 與 [S2.] 資料內容相減結果存入 [D.]。
- ◆ 所有運算均以代數進行，i.e.  $5+(-8) = -3$ 。
- ◆ 最高位代表正負號 (0:正，1:負)。
- ◆ 旗號設定及正負數間的關係。



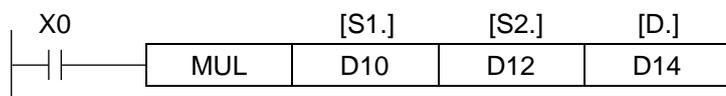
## FNC(22)乘法 MULTIPLICATION

FNC(22)			16 bits: MUL & MUL(P) ----- 7 Steps			J2n--	J3n--
D	MUL	P	32 bits: (D)MUL & (D)MUL(P) -----13 Steps				

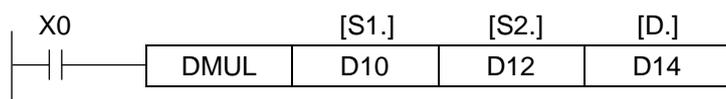
元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S1.]								●	●	●	●	●	●	●		●	●				●	●	
[S2.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●		●	●						



影響旗號:



16 bits: (D10) × (D12) → (D15,D14)



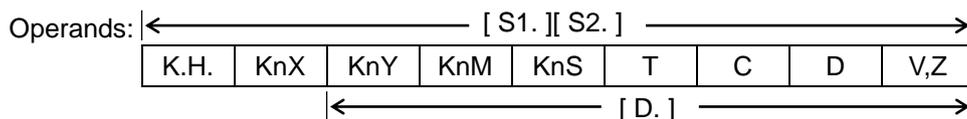
32 bits: (D11,D10) × (D13,D12) → (D17,D16,D15,D14)

- ◆ 來源 [S1.] 乘以 [S2.] 結果存入 [D.] 中。

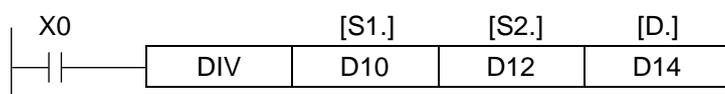
## FNC(23)除法 DIVISION

FNC(23)			16 bits: DIV & DIV(P) ----- 7 Steps			J2n--	J3n--
D	DIV	P	32 bits: (D)DIV & (D)DIV(P) -----13 Steps				

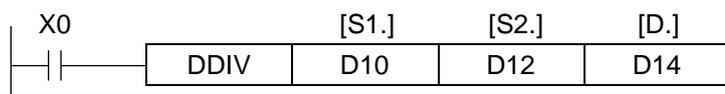
元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S1.]								●	●	●	●	●	●	●		●	●				●	●	
[S2.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●		●	●						



影響旗號:



Dividend    divisor    quotient    remainder  
 (D10) ÷ (D12) → (D14) ..... (D15)  
 16 bits    16 bits    16 bits    16 bits



Dividend    divisor    quotient    remainder  
 (D11,D10) ÷ (D13,D12) → (D15,D14) ..... (D17,D16)  
 32 bits    32 bits    32 bits    32 bits

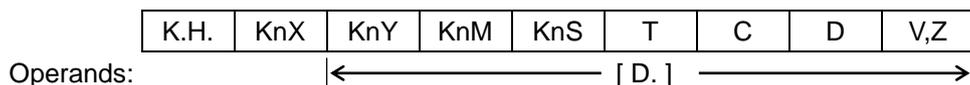
- ◆ 來源 [S1.] 除以 [S2.] 結果存入 [D.] 中。
- ◆ 若來源 [S2.] 等於 "0" (zero)，則程式停止運轉，錯誤旗標 M8067 ON，錯誤代碼 6706。
- ◆ V1.17 版：若來源 [S2.] = "0"，則不予處理，直接跳至下一個指令。



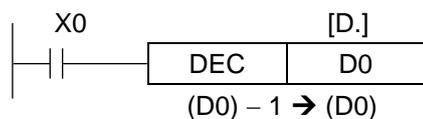
### FNC(25)遞減 DECREMENT

FNC(25)			16 bits: DEC & DEC(P) ----- 3 Steps			J2n--	J3n--
D	DEC	P	32 bits: (D)DEC & (D)DEC(P) -----5 Steps				

元件 種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[D.]									●	●	●	●	●	●		●	●						



影響旗號:



- ◆ 每次執行目的字元 [D.] 的內容減”1”。
- ◆ 16 bits 演算，當到達 -32,767，則下次執行會將 +32,768 寫入 [D.] 中。
- ◆ 32 bits 演算，當到達 -2,147,483,647，則下次執行會將+2,147,483,648 寫入 [D.] 中。
- ◆ 演算過程進位旗標、零位旗標、借位旗標均不變。

## FNC(26)邏輯及 LOGICAL AND

FNC(26)			16 bits: WAND & WAND(P) ----- 7 Steps			J2n--	J3n--
D	WAND	P	32 bits: (D)WAND & (D)WAND(P) -----13 Steps				

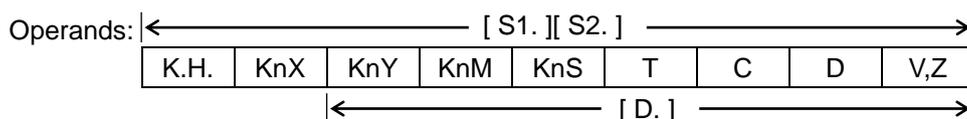
## FNC(27)邏輯或 LOGICAL OR

FNC(27)			16 bits: WOR & WOR(P) ----- 7 Steps			J2n--	J3n--
D	WOR	P	32 bits: (D)WOR & (D)WOR(P) -----13 Steps				

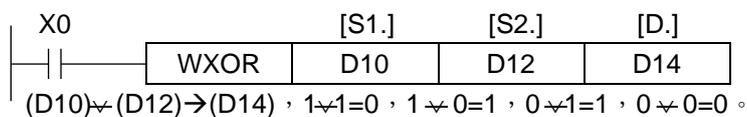
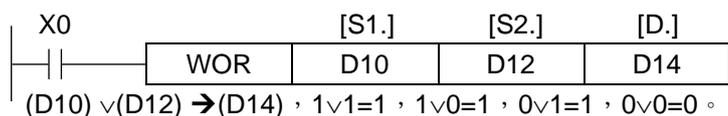
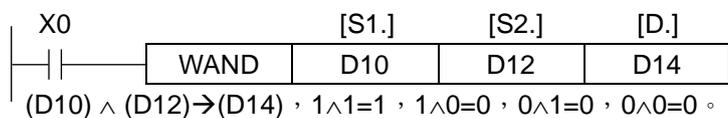
## FNC(28)互斥或 XOR

FNC(28)			16 bits: WXOR & WXOR(P) ----- 7 Steps			J2n--	J3n--
D	WXOR	P	32 bits: (D)WXOR & (D)WXOR(P) ----- 13 Steps				

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●		●	●			●	●		
[S2.]								●	●	●	●	●	●	●		●	●			●	●		
[D.]									●	●	●	●	●	●		●	●						



影響旗號:

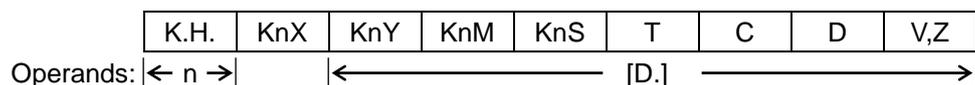




## FNC(30)右迴旋 ROTATION RIGHT

FNC(30)			16 bits: ROR & ROR(P) ----- 5 Steps			J2n--	J3n--
D	ROR	P	32 bits: (D)ROR & (D)ROR(P) -----9 Steps				

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[D.]									●	●	●	●	●	●		●	●						
[n.]																				●	●		

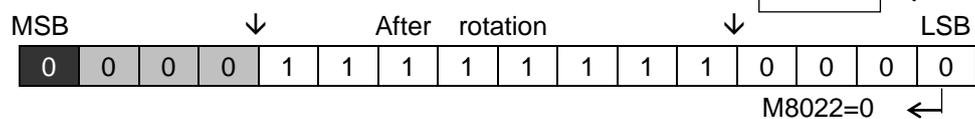
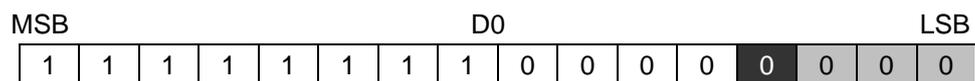
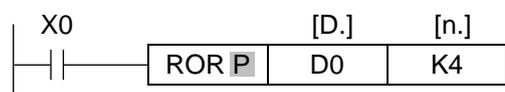


16bit :  $n \leq 16$

32bit :  $n \leq 32$

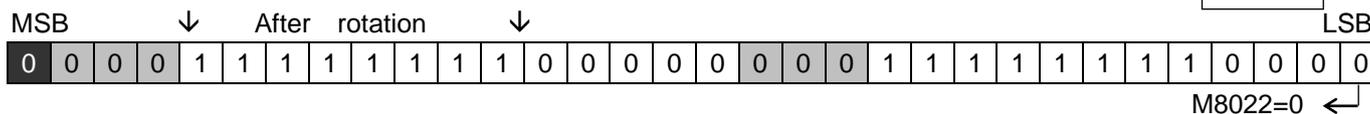
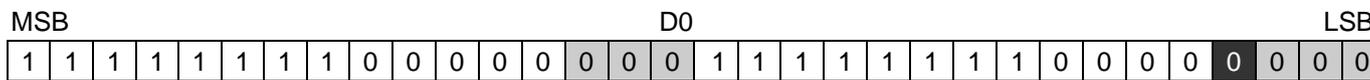
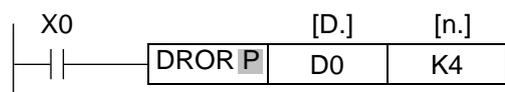
影響旗號: M8022

16bit



- ◆ 右旋後的最右位被存入進位旗標。
- ◆ 註: 16BIT 組成的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)
- ◆ 註: 32BIT 組成的元件 只能使用 K8 修飾(例:K8Y0 K8M0 K8S0)

32bit

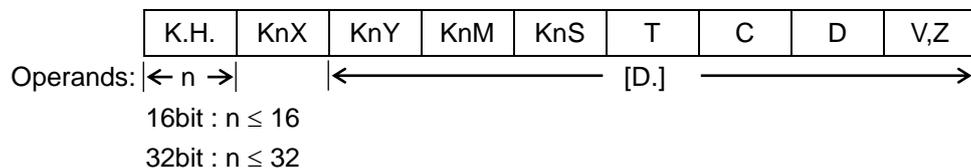


- ◆ 右旋後的最右位被存入進位旗標。
- ◆ 註: 16BIT 組成的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)
- ◆ 註: 32BIT 組成的元件 只能使用 K8 修飾(例:K8Y0 K8M0 K8S0)



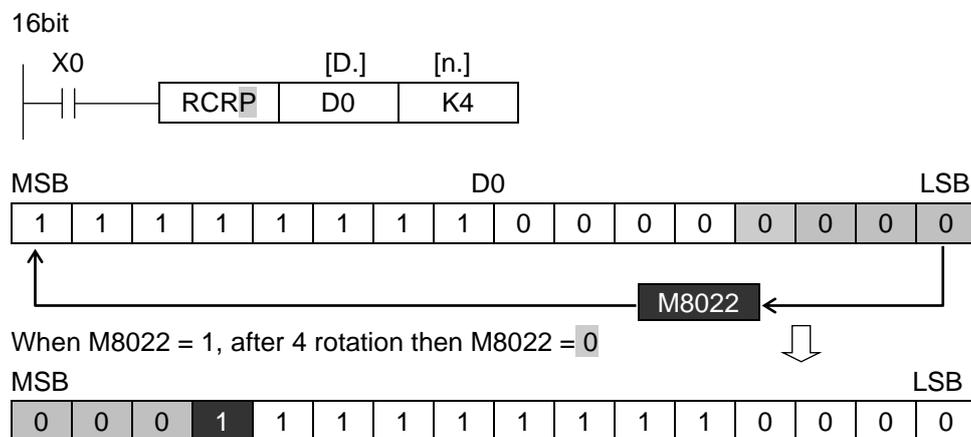
### FNC(32)右迴旋含進位旗號 ROTATION RIGHT WITH CARRY

FNC(32)			16 bits: RCR & RCR(P) ----- 5 Steps			J2n--	J3n--
D	RCR	P	32 bits: (D)RCR & (D)RCR(P) ----- 9 Steps				

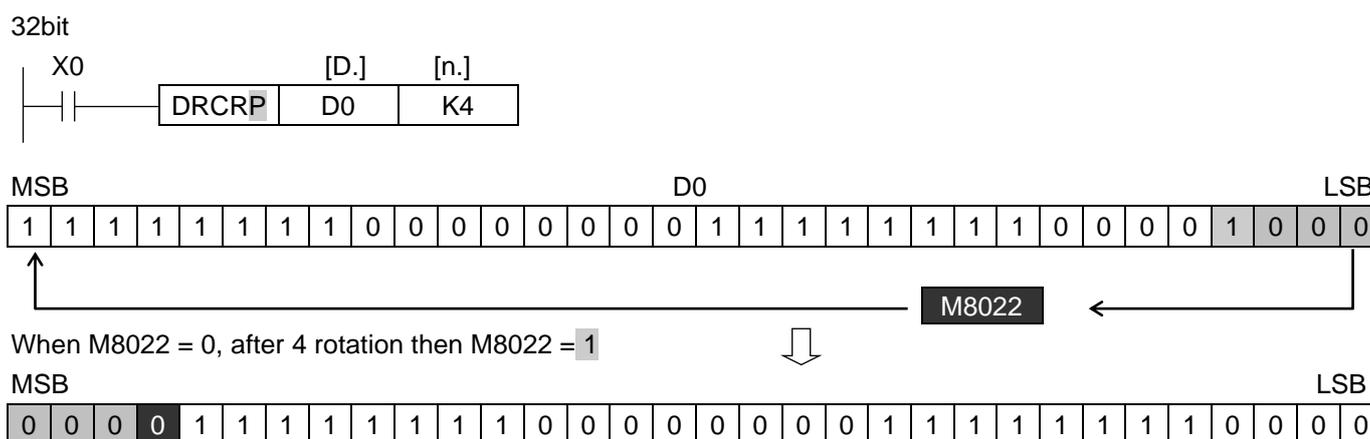


元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[D.]									●	●	●	●	●	●		●	●						
[n.]																				●	●		

影響旗號:



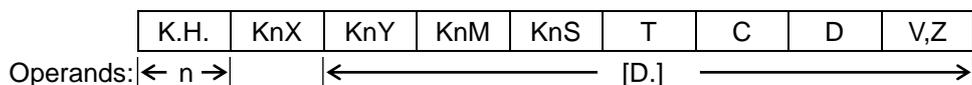
- ◆ 註: 16BIT 組成的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)
- ◆ 註: 32BIT 組成的元件 只能使用 K8 修飾(例:K8Y0 K8M0 K8S0)



- ◆ 註: 16BIT 組成的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)
- ◆ 註: 32BIT 組成的元件 只能使用 K8 修飾(例:K8Y0 K8M0 K8S0)

## FNC(33)左迴旋含進位旗號 ROTATION LEFT WITH CARRY

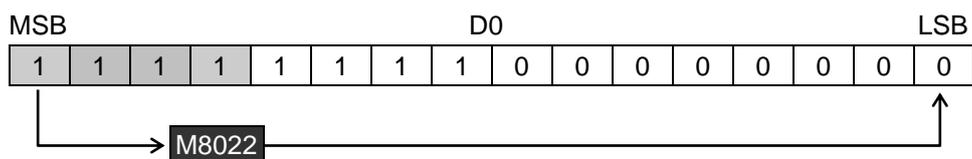
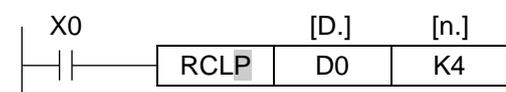
FNC(33)			16 bits: RCL & RCL(P) ----- 5 Steps			J2n--	J3n--
D	RCL	P	32 bits: (D)RCL & (D)RCL(P) ----- 9 Steps				

16bit :  $n \leq 16$ 32bit :  $n \leq 32$ 

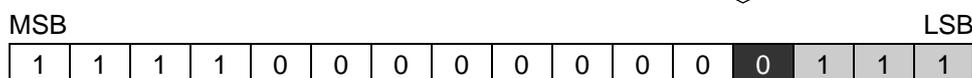
元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[D.]									●	●	●	●	●	●		●	●					
[n.]																				●	●	

影響旗號:

16bit



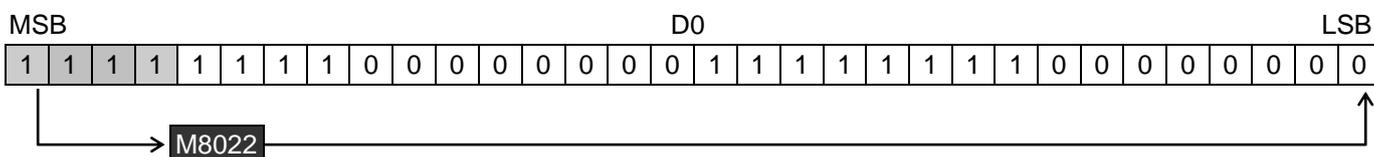
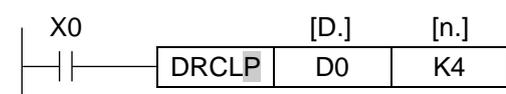
When M8022 = 0, after 4 rotation then M8022 = 1



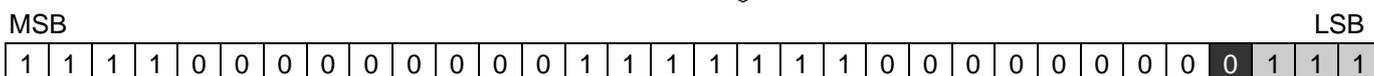
◆ 註: 16BIT 組成的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)

◆ 註: 32BIT 組成的元件 只能使用 K8 修飾(例:K8Y0 K8M0 K8S0)

32bit



When M8022 = 0, after 4 rotation then M8022 = 1



◆ 註: 16BIT 組成的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)

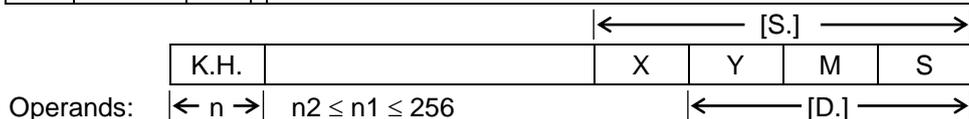
◆ 註: 32BIT 組成的元件 只能使用 K8 修飾(例:K8Y0 K8M0 K8S0)

### FNC(34)位右位移 SHIFT RIGHT

FNC(34)		16 bits: SFTR & SFTR(P) ----- 9 steps																J2n--	J3n--	
	SFTR	P																		

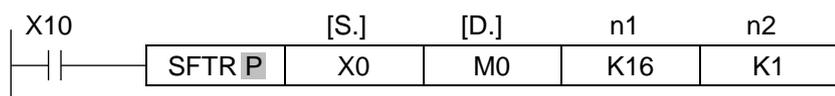
### FNC(35)位左位移 SHIFT LEFT

FNC(35)		16 bits: SFTL & SFTL(P) ----- 9 steps																J2n--	J3n--	
	SFTL	P																		



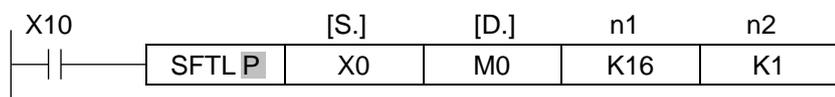
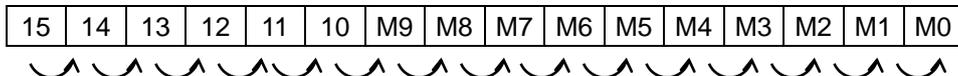
元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E	
[S.]	●	●	●	●																		
[D.]		●	●	●																		
[n.]																			●	●		

影響旗號:



X0

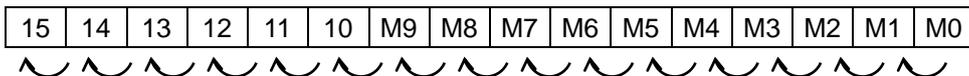
<< BIT SHIFT RIGHT >>



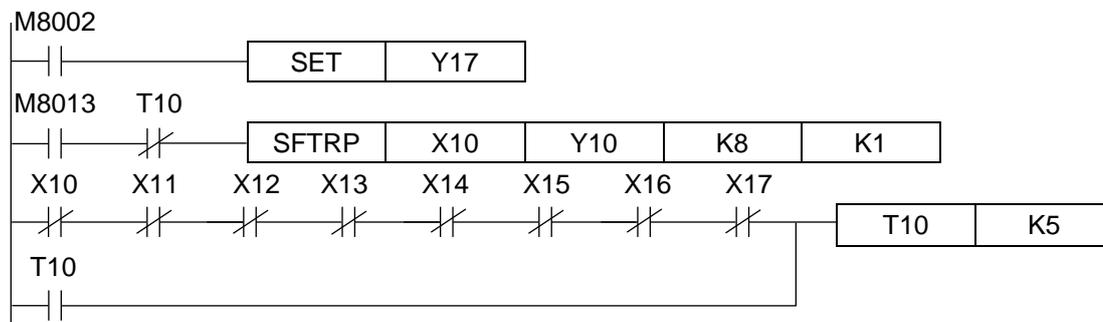
X0

<< BIT SHIFT LEFT >>

↓



範例 I/O 測試: 接線 X10 ↔Y10 ... X17 ↔Y17

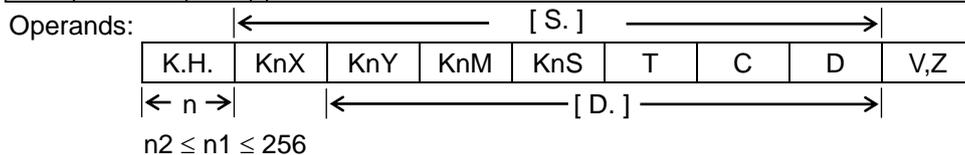


### FNC(36)字元右位移 WORD SHIFT RIGHT

FNC(36)		16 bits: WSFR & WSFR(P) ----- 9 steps												J2n--	J3n--
WSFR	P														

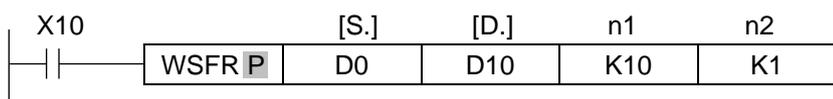
### FNC(37)字元左位移 WORD SHIFT LEFT

FNC(37)		16 bits: WSFL & WSFL(P) ----- 9 steps												J2n--	J3n--
WSFL	P														



元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[S.]							●	●	●	●	●	●	●									
[D.]								●	●	●	●	●	●									
[n.]																				●	●	

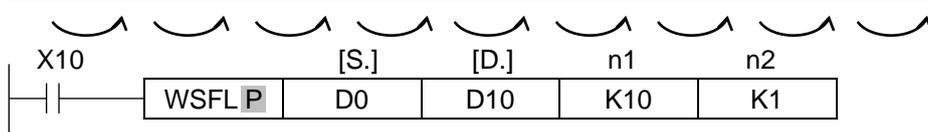
影響旗號:



D0

↓ << WORD SHIFT RIGHT >> n2 =< n1 =< 255

D19	D18	D17	D16	D15	D14	D13	D12	D11	D10
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----



D0

<< WORD SHIFT LEFT >> n2 =< n1 =< 255

D19	D18	D17	D16	D15	D14	D13	D12	D11	D10
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

◆ 註:16BIT 組成的元件 只能使用 K4 (例:K4Y0 K4M0 K4S0)

## FNC(38)移位暫存器寫入 SHIFT REGISTER WRITE

FNC(38)		16 bits: SFWR & SFWR(P) ----- 7 Steps												J2n--	J3n--				
SFWR	P																		

Operands: ← [ S. ] →

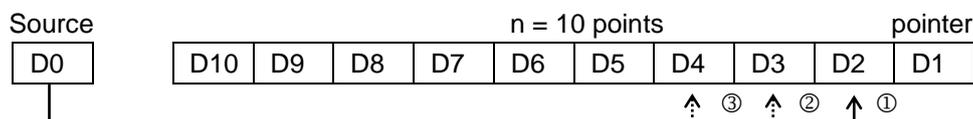
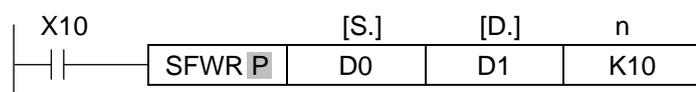
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

← n → [ D. ]

$2 \leq n \leq 256$

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]								●	●	●	●	●	●	●		●	●					●	●
[D.]									●	●	●	●	●	●									
[n.]																						●	●

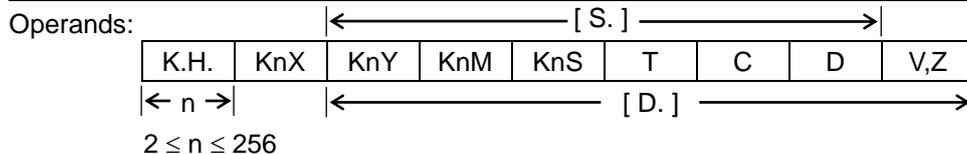
影響旗號:



- ◆ 當 X10 OFF → ON，來源 D0 的內容存入 D2 且堆疊指標 D1="1"，當下一個 OFF → ON 的脈波，D0 的內容存入 D3 且堆疊指標 D1="2"，[S.] 的內容依序存入目的暫存器且堆疊指標 [D.] 的內容自動加"1"。
- ◆ 假如 [D.] 的內容超過"n-1" (n 是 FIFO 堆疊長度)時，則無法處理且進位旗標 M8022 ON。
- ◆ 註:BIT 組成 WORD 的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)
- ◆ [D.]的堆疊次數不能為負 否則出現 Error

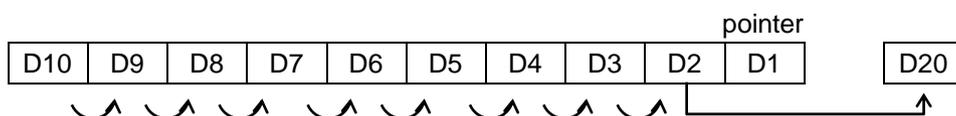
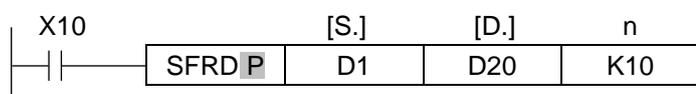
## FNC(39)移位暫存器讀取 SHIFT REGISTER READ

FNC(39)		16 bits: SFRD & SFRD(P) ----- 7 Steps												J2n--	J3n--
SFRD	P														



元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S.]									●	●	●	●	●	●										
[D.]									●	●	●	●	●	●		●	●							
[n.]																				●	●			

影響旗號:



- ◆ 當 X10 OFF → ON，D2 的內容存入 D20 且堆疊指標 D1 的內容自動減“1”。
- ◆ 當 [S.] = “0”， i.e. FIFO 堆疊無數據，零位元旗標 M8020 ON。
- ◆ 數據一律由 [S.]+1 處讀出，且 D10 的內容保持不變。
- ◆ 註:BIT 組成 WORD 的元件 只能使用 K4 修飾(例:K4Y0 K4M0 K4S0)
- ◆ [D.]的堆疊次數不能為負 否則出現 Error
- ◆ [D.]的堆疊次數不能大於 n 否則出現 Error

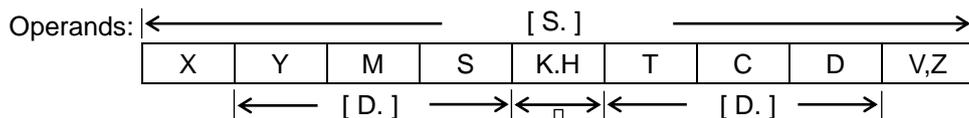


## FNC(41)解碼 DECODE

FNC(41)		16 bits: DECO(P) ----- 7 steps												J2n--	J3n--
DECO	P														

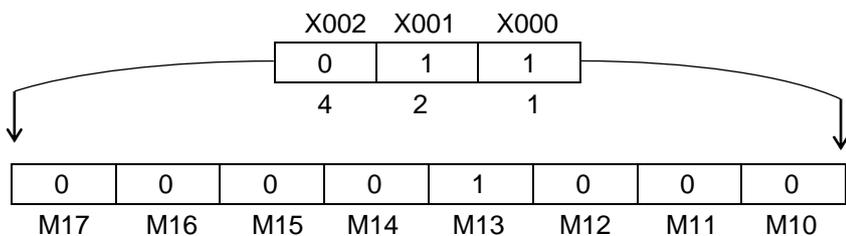
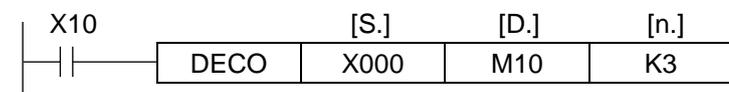
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E		
[S.]	●	●	●	●								●	●	●		●	●		●	●			
[D.]		●	●	●								●	●	●									
[n.]																			●	●			

n=1-8



影響旗號:

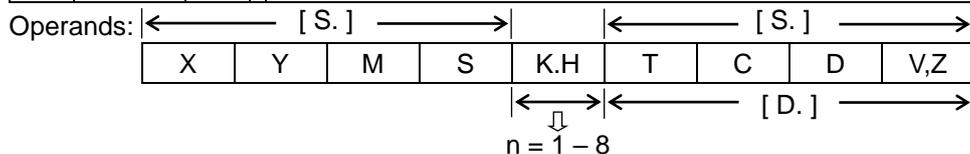
n = 1 - 8



- ◆ [D.] 所指定的要素若為 T, C 或 D 時，則  $n \leq 4$ 。
- ◆ 若來源全部為“0”，則 M10 被設為“1”。

## FNC(42)編碼 ENCODE

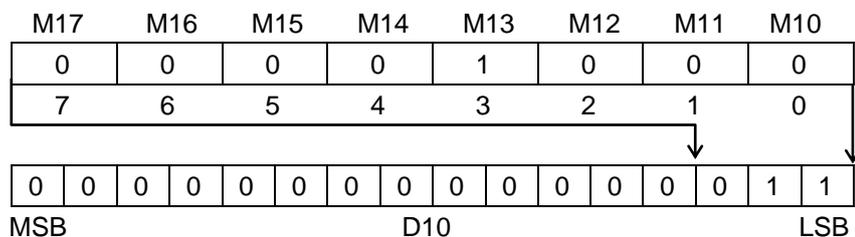
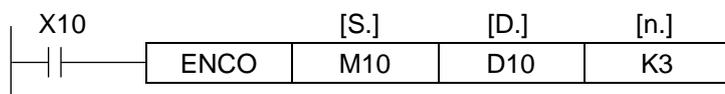
FNC(42)		16 bits: ENCO(P) ----- 7 steps												J2n--	J3n--				
ENCO	P																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E			
[S.]	●	●	●	●								●	●	●		●	●							
[D.]												●	●	●		●	●		●	●				
[n.]																			●	●				

n=1 - 8

影響旗號:



- ◆ [S.] 所指定的要素若為 T,C 或 D 時，則  $n \leq 4$ 。
- ◆ 若來源要素中“1”的位超過 1 個時，僅最低位的“1”視為有效。
- ◆ 若來源要素的每一位都為“0”，則將發生錯誤。

## FNC(43)和 SUM

FNC(43)			16 bits: SUM(P) ----- 5 steps			J2n--	J3n--
D	SUM	P	32 bits: (D)SUM(P) ----- 9 steps				

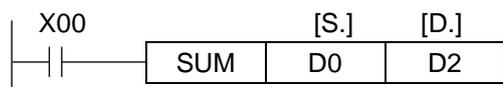
Operands: ← [ S. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

← [ D. ] →

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]								●	●	●	●	●	●	●		●	●			●	●		
[D.]									●	●	●	●	●	●		●	●						

影響旗號:



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

D 0



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

D 2

- ◆ [S.] 資料中“1”的個數存入[D.]中。
- ◆ 若[S.]中任一為“0”，則零位旗標 M8020 ON。

### FNC(44)位檢查 BIT ON CHECK

FNC(44)			16 bits: BON(P) ----- 7 steps			J2n--	J3n--
D	BON	P	32 bits: (D)BON(P) ----- 13 steps				

Operands: ← [ S. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

← n → [ n. ] = 0 - 15 或 0 - 31

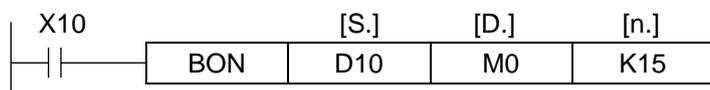
Operands: ← [ D. ] →

X	Y	M	S
---	---	---	---

元件 種類	位元元件							位元組合元件				字元元件				指標			字串	常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[S.]								●	●	●	●	●	●	●		●	●			●	●	
[D.]		●	●	●																		
[n.]																				●	●	

n = 0-15 或 0-31

影響旗號:



0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Bit15, D10=0, then M0 = OFF															
LSB															

1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Bit15, D10=1, then M0 = ON															
LSB															

### FNC(45)平均值 MEAN

FNC(45)		16 bits: MEAN(P) ----- 7 steps									J2n--	J3n--			
MEAN	P														

Operands:      ←----- [ S. ] ----->

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

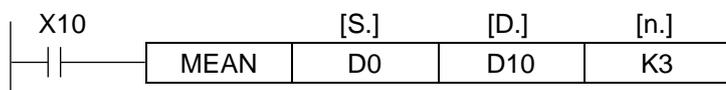
Operands: ← n →      ←----- [D.] ----->

[n]=1-64

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E	
[S.]								●	●	●	●	●	●	●										
[D.]									●	●	●	●	●	●		●	●							
[n.]																						●	●	

n= 1-64

影響旗號:



$$[(D0)+(D1)+(D2)] / 3 \rightarrow (D10)$$

**FNC(46)故障指示器設定 ANNUNCIATOR SET**

FNC(46)			16 bits: ANS ----- 7 steps													
	ANS															

Reserved

**FNC(47)故障指示器複置 ANNUNCIATOR RESET**

FNC(47)			16 bits: ANR(P) ----- 1 steps													
	ANR															

Reserved

**FNC(48)開平方根 SQUARE ROOT**

FNC(48)			16 bits: SQR(P) ----- 5 steps												J2n	J3n--
D	SQR	P	32 bits: (D)SQR(P) ----- 9 steps													

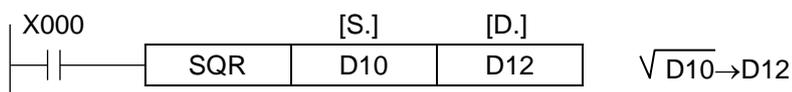
Operands:  $\longleftrightarrow$  [S.] [S.]  $\longleftrightarrow$ 

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

Operands: [D.]  $\longleftrightarrow$ 

元件 種類	位元元件							位元組合元件				字元元件				指標				字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E	
[S.]														●							●	●		
[D.]														●										

影響旗號: M8020, M8021, M8022



- ◆[S.]之內容必須為正數，若為負數時，錯誤旗號 M8067 將會動作，且命令不會執行。
- ◆演算結果若有小數點將被舍去，若結果未滿 1 而被舍去時，借位旗號 M8021 將會動作。
- ◆演算結果正好為 0 時，零旗號 M8020 將會動作。

## FNC(49)浮動小數點 FLOAT

FNC(49)			16 bits: FLT(P) -----5 steps			J2n	J3n--
D	FLT	P	32 bits: (D)FLT(P) -----9 steps				

Operands: [S.] ←→

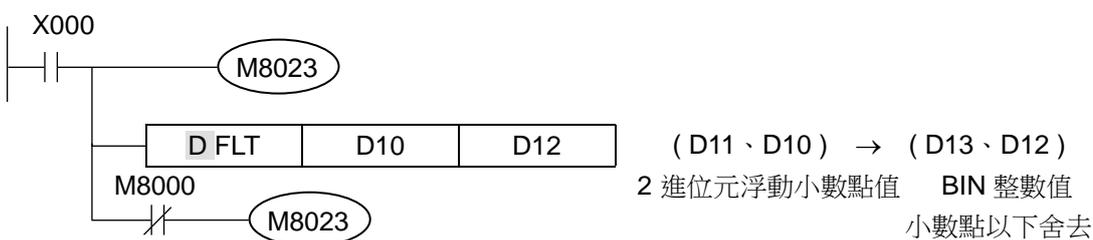
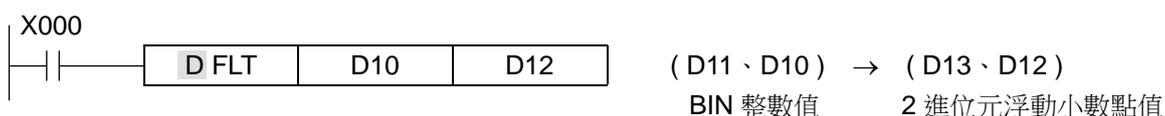
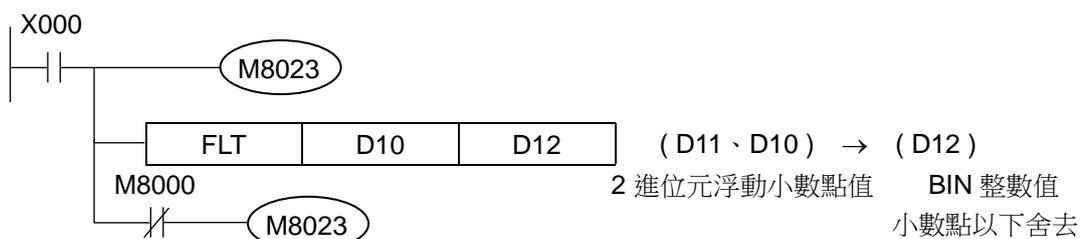
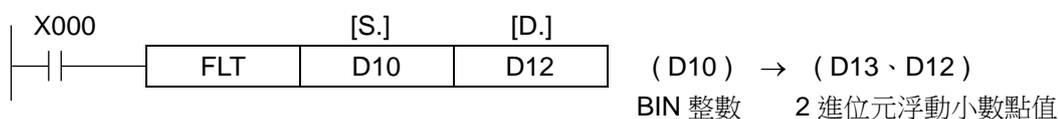
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

Operands: [D.] ←→

元件種類	位元元件						位元組合元件				字元元件				指標		字串		常數		實數		
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]														●									
[D.]														●									

影響旗號: M8020, M8021, M8022

- ◆ FLT 指令為 BIN 之整數值與 2 進位元浮動小數點值間的轉換命令，由於常數 K,H 值於浮動小數點運算時會自動轉換，因此不適用於此命令。



- ◆ M8023 = ON 時，執行 2 進位浮動小數點值→BIN 整數值之轉換。  
M8023 = OFF 時，則執行反方向之轉換。
- ◆ 2 進位浮動小數點值→BIN 整數值之轉換結果若未滿 1，而造成被舍去為 0 或產生溢位時，M8021 / M8022 將分別 ON，結果若正好為 0 時，M8020 將會 ON。

## FNC(50)輸出入更新 REFRESH

FNC(50)			16 bits: REF(P) ----- 5 steps															J2n--	J3n--		
REF	X	P																			

Operands:

K,H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
← n →								

Operands:

X	Y	M	S
← [D.] →			

元件 種類	位元元件							位元組合元件					字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E	
[D.]	●	●																						
[n.]																					●	●		

[D.] X/Y 的號碼應指定為 10 的倍數，X00, X10,.....

[n.] K/H 的號碼應指定為 8 的倍數，i.e. 8,16,24,....

影響旗號:

- ◆ PLC 輸入全部更新均在程式 STEP 0 執行前，而輸出均在 END 或 FEND 命令之後執行，在演算過程中不變更。  
若演算過程中需即時的輸入資料或輸出演算的結果，則須用輸出入更新命令。

&lt;&lt; 輸入更新 &gt;&gt; 僅 X10 – X17 被更新



&lt;&lt; 輸出更新 &gt;&gt; Y00-Y07，Y10-Y17，Y20-Y27 被更新。

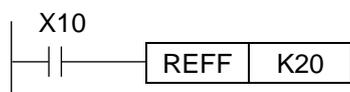


- ◆ 中斷程式內不可使用 FNC(50) REF 指令。

## FNC(51)更新及時間濾波器 REFRESH AND FILTER ADJUSTMENT

FNC(51)		16 bits: REFF(P) ----- 3 steps				J2n--	J3n--
	REFF	P					

Operand: [n.] = 0 - 60



- ◆ 為防止不要雜訊干擾，一般 PLC 輸入繼電器都有硬體 RC 濾波器的設計，及可用來調整軟體濾波器的時間。
- ◆ 此命令只變更 X00-X07 軟體濾波時間，即 D8020 的內容。若須變更其他輸入點的濾波時間，請利用 MOV 指令。

## FNC(52)陣列 MATRIX

FNC(52)		16 bits: MTR ----- 9 Steps																J2n--	J3n--
MTR																			

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]	●																						
[D1.]		●																					
[D2.]		●	●	●																			
[n.]																					●	●	

Operands:

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
← n →								

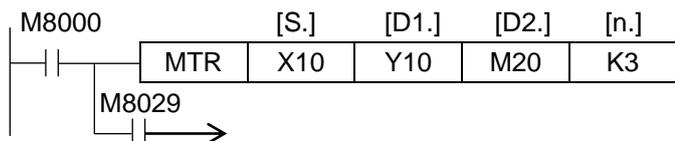
Operands ←[S.]→←[D1.]→

X	Y	M	S
← [ D2. ] →			

(S.): X00, X10, X20, X30 -----X160, X170。

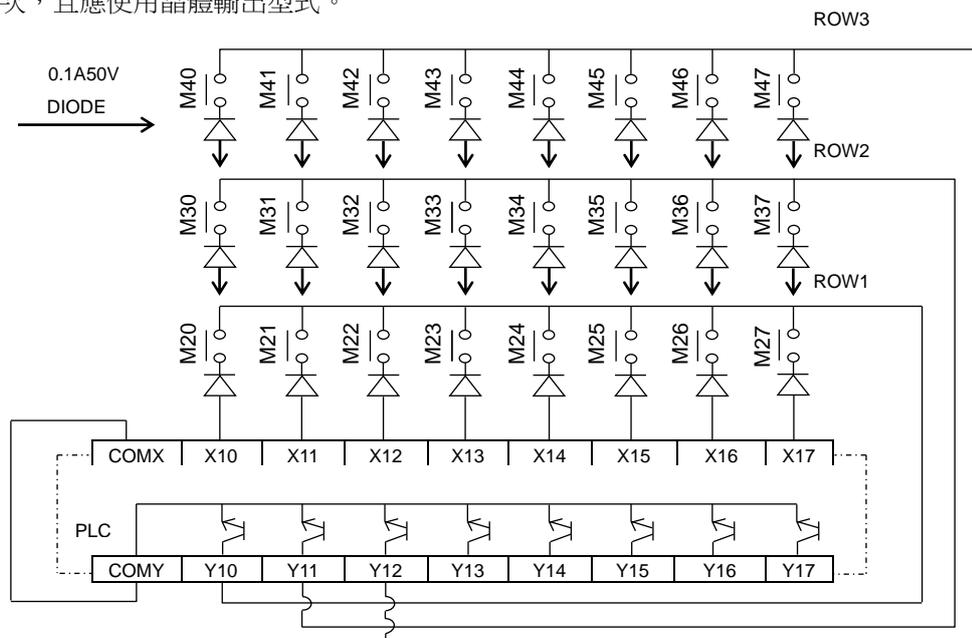
(D1.): Y00, Y10, Y20, Y30 -----Y160, Y170。 (D2.): Y, M, S multiple of 10, i.e. 00, 10, 20 etc。

(n.): K, H n=2~8。



- ◆ MTR 指令用來讀取 8 點 n 列的輸入信號(8 個輸入點, n 個輸出點)。8 個輸入點的起始由(S.)指定, n 個輸出點的起始由(D1.)指定, 如例所示, 輸出點 Y10, Y11, Y12 依序且重複為 ON, 則第一列、第二列、第三列的輸入點被依序且重複的讀入, 並存放在 M10~M17, M20~M27, M30~M37 中。
- ◆ (D2.)為陣列的起始位址。
- ◆ 當命令執行完畢, 完成旗號 M8029 設為 ON。當命令再度被執行時, 完成旗號 M8029 自動復置。

- ◆ 此命令只能使用一次，且應使用晶體輸出型式。



M8000

ROW1 X10 ~ X17 STATUS -> M20 ~ M27



ROW2 X10 ~ X17 STATUS -> M30 ~ M37



ROW3 X10 ~ X17 STATUS -> M40 ~ M47



## FNC(53)高速計數器設定 SET BY HIGH SPEED COUNTER

FNC(53)		32 bits: HSCS ----- 13 Steps													J2n--		J3n--	
D	HSCS																	

Operands: ← [ S1. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

↔ [ S2. ] : C235~C255

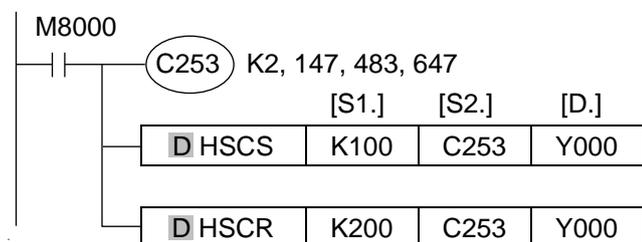
Operands:

X	Y	M	S
---	---	---	---

← [ D. ] → [D.]時，I010~I060 中斷可使用索引來指定。

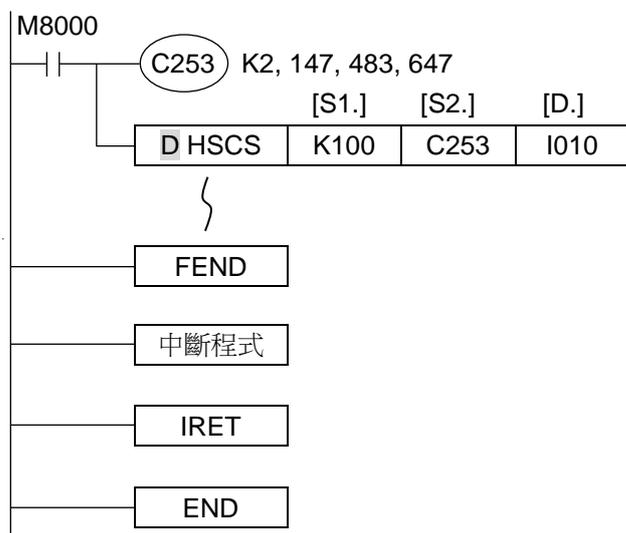
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●	●	●	●				●	●	
[S2.]													●										
[D.]		●	●	●																			

影響旗號:



◆ 使用 FNC53 命令時，外部輸出動作以中斷來處理，C253 的現在值 99→100 和 101→100 變化時，Y000 被 SET。當 C253 的現在值 199→200 和 201→200 變化時，Y000 OFF。

- ◆ 此命令為 32 位之專用命令，請輸入 D HSCS 命令。
- ◆ FNC53, FNC54, FNC55 只可以使用一次。



◆ D HSCS 命令的 [D.] 可指定 I0 □ 0 = (□=1~6)(□=1~6 不可重複使用編號)

◆ 因此 [S2.] 所指定的高速計數器的現在值與 [S1.] 所指定的值相同時，中斷主程序，立即跳往指針 I0 □ 0 中斷程式執行。

◆ 特殊輔助繼電器 M8059 ON 時，I010~I060 中斷全部禁止。

### FNC(54)高速計數器複置 RESET BY HSC

FNC(54)						J2n--	J3n--
D	HSCR	32 bits: HSCR ----- 13 Steps					

Operands: ← [ S1. ] →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

↔ [ S2. ] : C235~C255

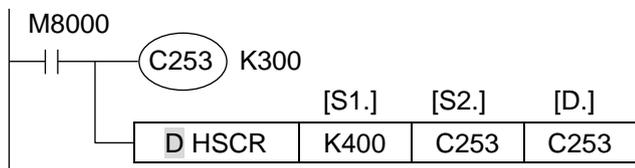
Operands:

X	Y	M	S
---	---	---	---

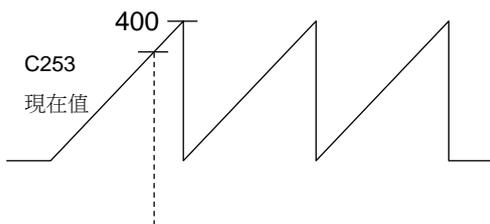
← [D.] → 可指定 [D.] 與 [S2.] 為同一高速計數器。

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S1.]								●	●	●	●	●	●	●	●	●	●					●	●	
[S2.]													●											
[D.]		●	●	●																				

影響旗號:



◆ C253 的現在值為 400 時，C253 立即清除，現在值變成 0，輸出接點不動作。



◆ 此命令為 32 位專用命令，請務必用 **D HSCR** 命令。

### FNC(55)高速計數器區域比較 ZONE COMPARE FOR HSC

FNC(55)							
D	HSZ	32 bits: HSZ----- 17 Steps					

Reserved

## FNC(56)速度偵測 SPEED DETECT

FNC(56)		16 bits: SPD ----- 7 Steps												J2n--		J3n--	
SPD																	

Operands: (S1.): X000~X005

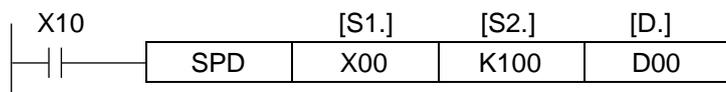
Operands: ←----- [ S2. ] -----&gt;

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

←----- [D.] -----&gt;

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S2.]								●	●	●	●	●	●	●	●	●	●					●	●	
[D.]												●	●	●		●	●							

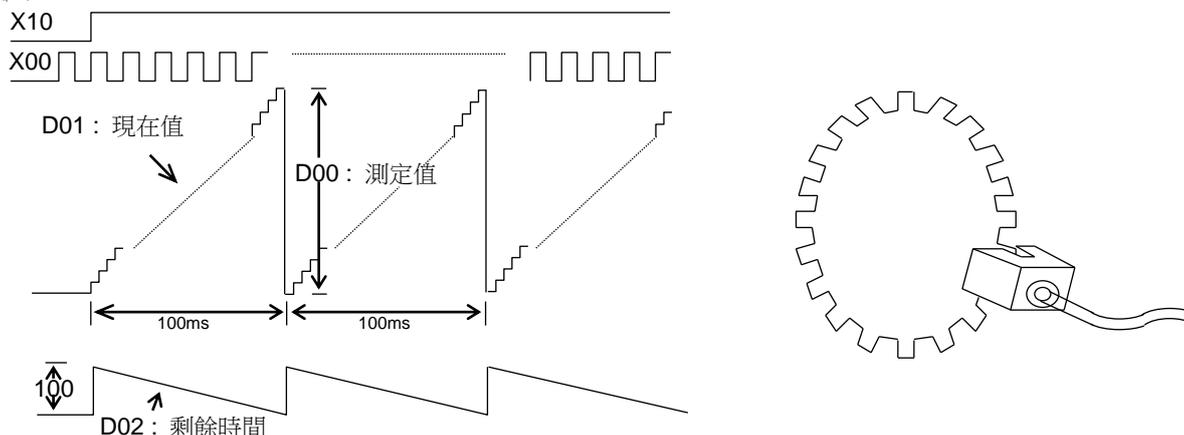
影響旗號:M8029



- ◆ 輸入脈波由 [S1.] 指定，[S2.] 指定計數時間，隨時把結果存放在 [D.] 中。
- ◆ 本指令[D.]會佔用 3 個目的要素。(本例為 D00-D02)
- ◆ 本例 D01 計算 X00 (OFF→ON)的次數，100msec 後，把計數結果存入 D00 中，然後 D01 被複置再重新開始計數。
- ◆ D02 用來測量剩餘時間。
- ◆ 指定的時間所計數的脈波量不可超過 65535。
- ◆ 脈波密度與 RPM 成比例，下列公式可求得回轉數。  

$$\text{RPM} : N = (D00 \times 60) \times 1000 / n \times t \quad n = \text{每轉脈波數}, t = \text{測量時間}。$$
- ◆ 輸入(X00-X05) ON/OFF 最大頻率與單相高速計數器相同。
- ◆ SPD 指令所使用的輸入點(X00-X05)，不得再作為其他高速處理或是插斷信號。當 C251 使用時，X00,X01 不可再做為速度偵測點。
- ◆ 利用完成旗號 M8029，輕易達到連續測得多筆資料，再求平均值。

### (i) 量頻率模式



### (ii) 量脈波寬度模式

- ◆ [S2.] 的內容值為"0"時，只需 1 脈波寬度即可測出速度 pps(pulse/second)。
- ◆ 本例速度值存放在 D01,D00 中。

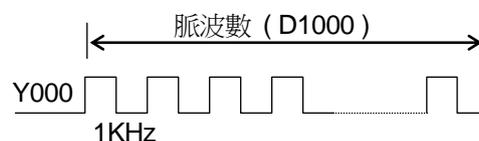
## FNC(57)脈波輸出 PULSE OUTPUT

FNC(57)												J2n--	J3n--
D	PLSY	32 bits: (D)PLSY----- 13 steps											
Operands:		← [ S1. ] →											
		K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z			
[D.] : Y00 – Y03										←[S2.]→			

元件 種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●	●	●	●				●	●	
[S2.]														●									
[D.]																							

[D.]: Y00~Y03

影響旗號: M8029



- ◆ 此指令為無加減速發送指定頻率和指定數量脈衝的指令。
- ◆ [D.] 指定脈波輸出點
  - [S1.] 指定輸出頻率(10~200,000Hz).
  - [S2.] 從指定的[S2.]開始會連續佔用 100 個 words。在本例中，佔用 D1000~D1099。
  - [S2.]+1, [S2.]+0：輸出脈波數
  - [S2.]+3, [S2.]+2：系統保留
  - [S2.]+5, [S2.]+4：啟始地址
  - [S2.]+7, [S2.]+6：絕對位址(監視用)
  - [S2.]+9, [S2.]+8：相對位址(監視用)
- ◆ DPLSY 用以產生一指定脈波數，32 位: 1 ~ 2,147,483,647 個脈波。
- ◆ 若[S2.]+1, [S2.]+0 指定為“0”，則無限制的產生脈波。
- ◆ 固定為 32 位運算。若指定 16 位元運轉模式，則產生 error 6509。
- ◆ 脈波導通週期 (duty cycle) 50% ON 50% OFF。
- ◆ 執行過程中，若更動 [S2.]+1, [S2.]+0 的數值不予考慮，須待下一次輸出才有效。指令執行完畢 M8029 ON。
- ◆ 此命令只能使用一次，且應使用晶體輸出型式。

## FNC(58)脈波寬度調變 PULSE WIDTH MODULATION

FNC(58)		16 bits: PWM ----- 7 steps												J2n--	J3n--				
PWM																			

Operands: ← [ S1. ] [ S2. ] →

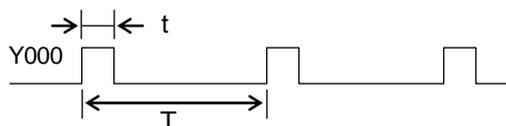
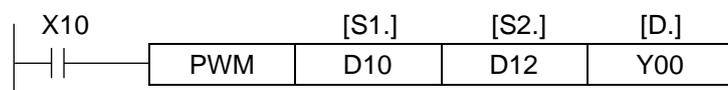
K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

[D.] : Y00 – Y07

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●	●	●	●			●	●		
[S2.]								●	●	●	●	●	●	●	●	●	●			●	●		
[D.]																							

[D.] : Y00~Y07

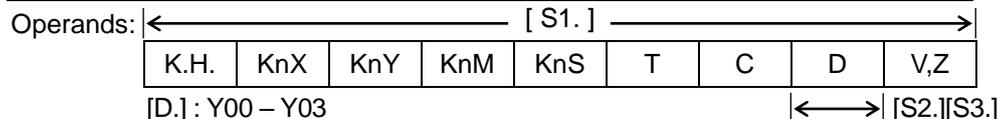
影響旗號: 無



- ◆ [S1.] 指定(t)導通脈波寬度，Y00 - Y01 值範圍(0 - 32,767) x 0.01ms；Y02 - Y07 值範圍 (0 - 32,767 msec)
- ◆ [S2.] 指定(T)週期，Y00 - Y01 值範圍(0 - 32,767) x 0.01ms；Y02 - Y07 值範圍 (0 - 32,767 msec)  
J34MT，ML 系列 Y00 - Y03 值範圍(0 - 32,767) x 0.01ms；Y04 - Y07 值範圍 (0 - 32,767 msec)
- ◆ [D.] 指定輸出點。(以中斷方式輸出)
- ◆ 若 [S1.] 的數值 > [S2.] 的數值，CPU 判定錯誤。
- ◆ 此指令外加濾波回路即可仿真為類比輸出。
- ◆ 此命令應使用晶體輸出型式。

## FNC(59)付加減速脈波輸出 PULSE OUTPUT WITH SLOPE

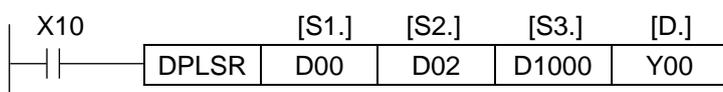
FNC(59)												J2n--	J3n--
D	PLSR	32 bits: (D)PLSR----- 17 steps											



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S1.]								●	●	●	●	●	●	●	●	●	●					●	●	
[S2.]														●										
[S3.]														●										

[D.]: Y00~Y03

影響旗號: M8029



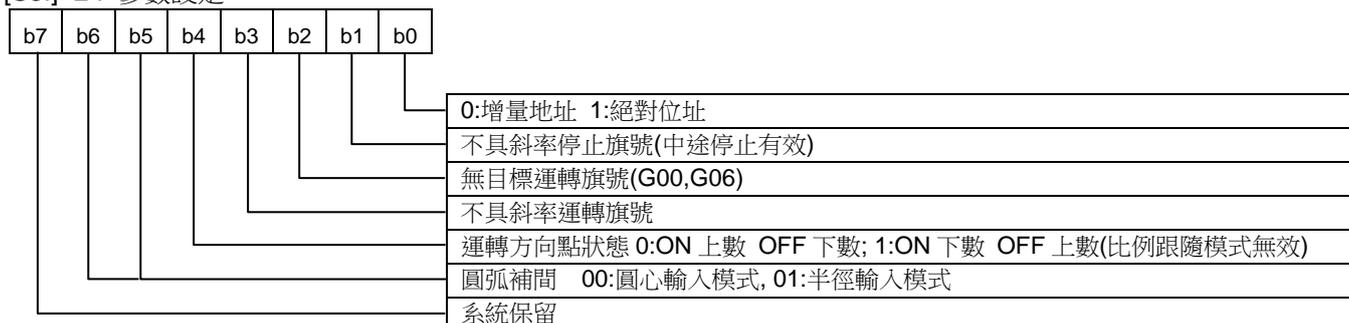
- ◆ [D.] 指定脈波輸出點，若指定 Y04 則為虛擬軸 無實際脈波輸出。
- [S1.] 指定輸出頻率。(10 ~ 200,000 pps)
- [S2.] 指定輸出脈波數。從指定的[S2.]開始會連續佔用 8 個 words。本例中，佔用 D02~D09。
- [S3.] 從指定的[S3.]開始會連續佔用 100 個 words。在本例中，佔用 D1000~D1099。
- [S3.]+0：運動模式: 命令碼 0~99 相當於 G00~G99

命令碼	內容
00	一段位置運動
01	直線補間(B type only)
02	圓弧補間 CW (B type only)
03	圓弧補間 CCW (B type only)
04	凸輪運動
05	區間運動
06	比例跟隨(硬體追隨):齒輪比必須為分數，分子<分母
07	飛剪
08	追剪
09	保留
28	原點復歸

[S3.]+1：運轉方向控制點: Y02~Y07



[S3.]+2：參數設定





[S3.] +3 : 系統保留

[S3.] +5, [S3.] +4 : 啟始位址(監視用)

[S3.] +9, [S3.] +8 : 相對位址(監視用)

[S3.] +13, [S3.] +12 : 目標位址(監視用)

[S3.] +17, [S3.] +16 : 最高速度

[S3.] +20 : 啟始速度(pps)

[S3.] +22 : 加速時間(1ms - 50,000ms)

[S3.] +24 : DOG(近點信號)

[S3.] +7, [S3.] +6 : 絕對位址(監視用)

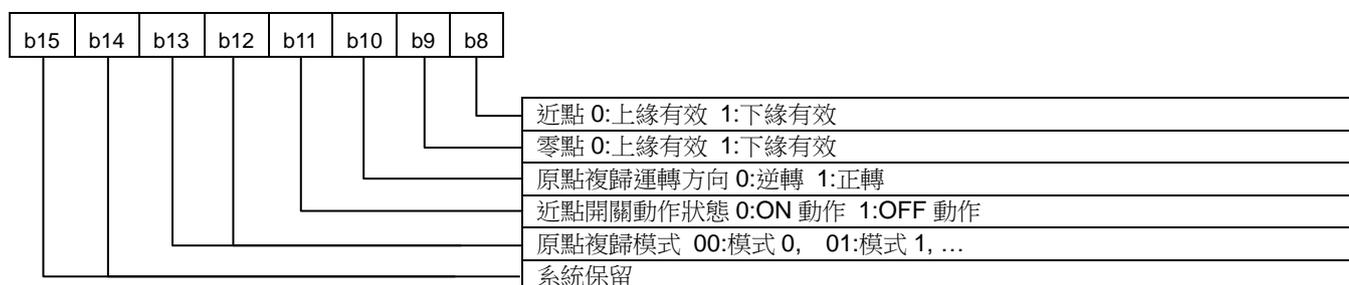
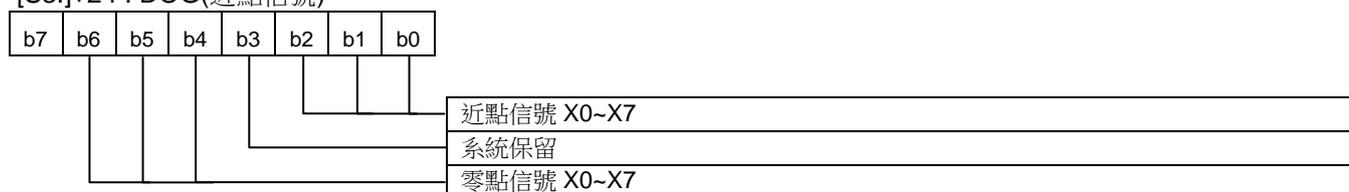
[S3.] +11, [S3.] +10 : 剩餘脈波(監視用)

[S3.] +15, [S3.] +14 : 現在速度(監視用)

[S3.] +19, [S3.] +18 : 系統保留

[S3.] +21 : 系統保留

[S3.] +23 : 減速時間(1ms - 50,000ms)



[S3.] +25 : 零點信號設定值。歸原點時，若無零點信號(步進馬達時)，則將找尋零點次數設為"0"即可。

[S3.] +26 : 零點信號計數值(監視用)

[S3.] +27 : 系統保留

[S3.] +28 : 電子齒輪比(分子)

[S3.] +29 : 電子齒輪比(分母)

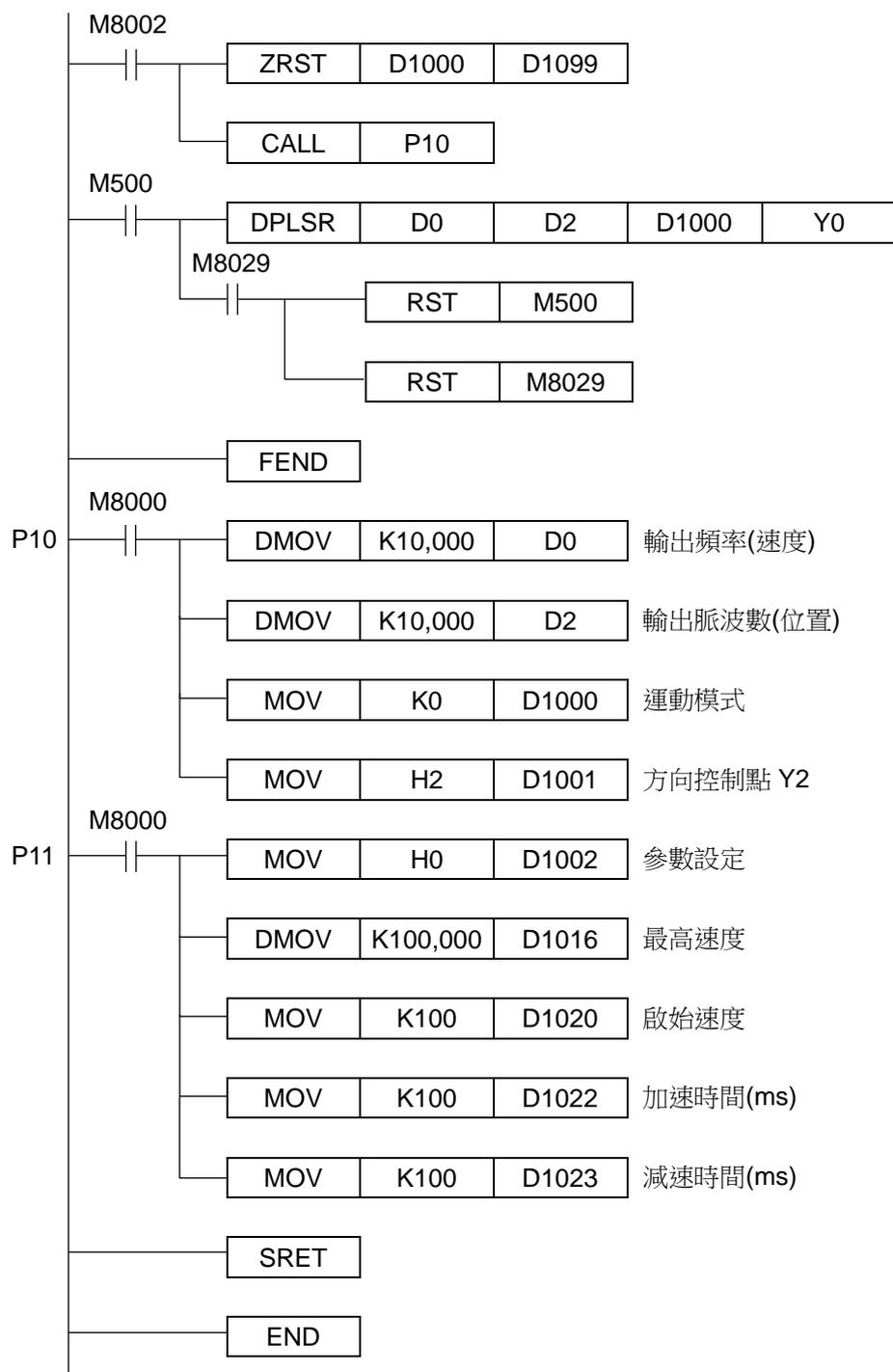
[S3.] +30 : 系統保留

[S3.] +32 : 系統保留

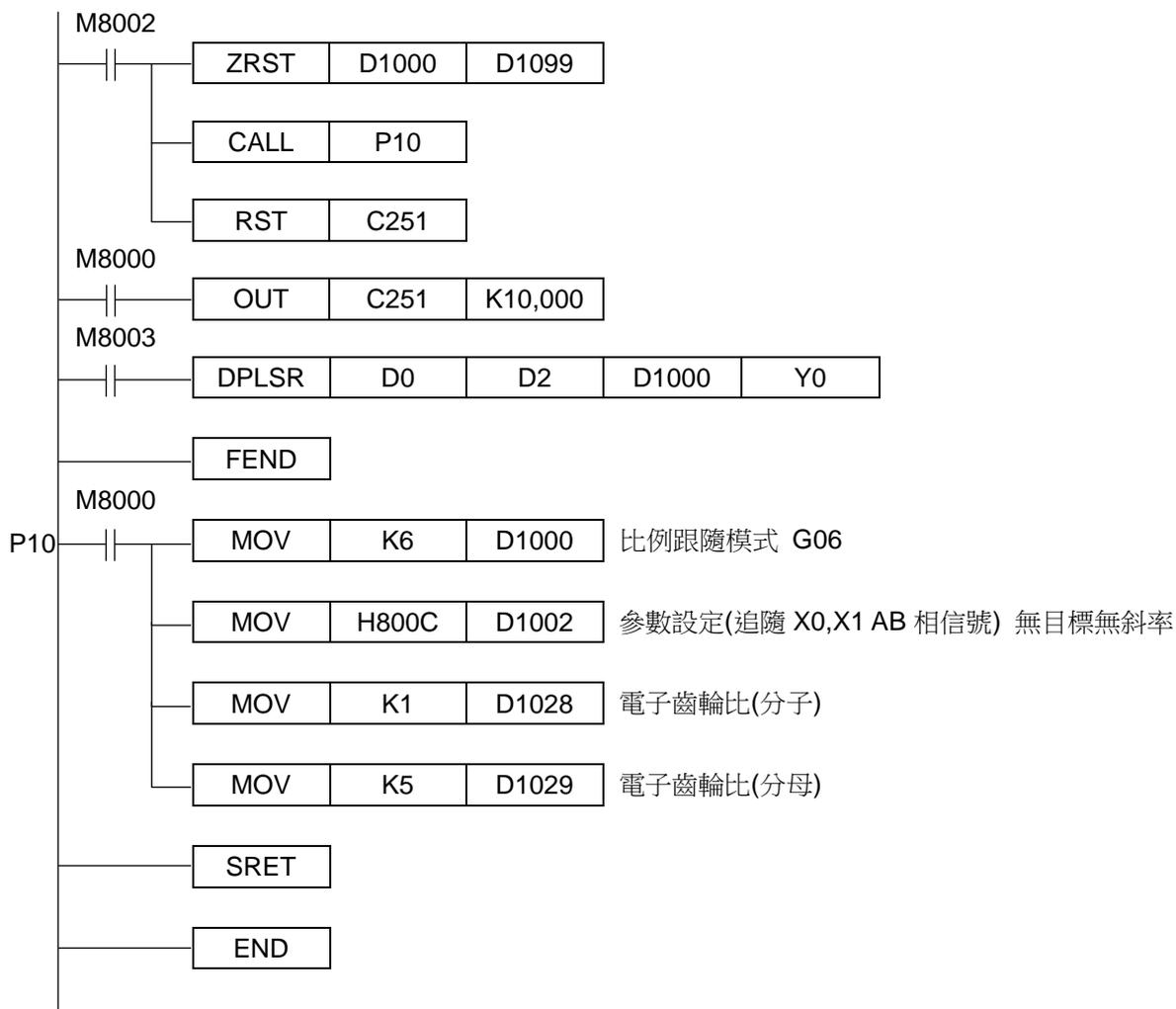
- ◆ 多軸同動：先驅動虛擬軸 其他軸設為[G06]比例跟隨模式且追隨虛擬軸 Y4 的脈波信號
- ◆ JOG+ JOG- 設定無目標運轉旗號，相對位置模式 利用位置正負值 控制運轉方向
- ◆ 使用此命令，須先將相對距離或絕對位址換算為脈波數再存入 [S2.] 中。
- ◆ 脈波輸出中，X10 OFF，脈波依停止旗號[S3.] +2, b1 的設定狀態停止輸出。
- ◆ 脈波導通週期( duty cycle) 50% ON, 50% OFF。
- ◆ 固定為 32 位運算，且應選擇為晶體輸出型式。若指定 16 位元運轉模式，則產生 error 6509
- ◆ 此命令脈波輸出型式只有一種 (Negative Logic Type, Pulse & Sign)，可用來控制步進或伺服馬達。



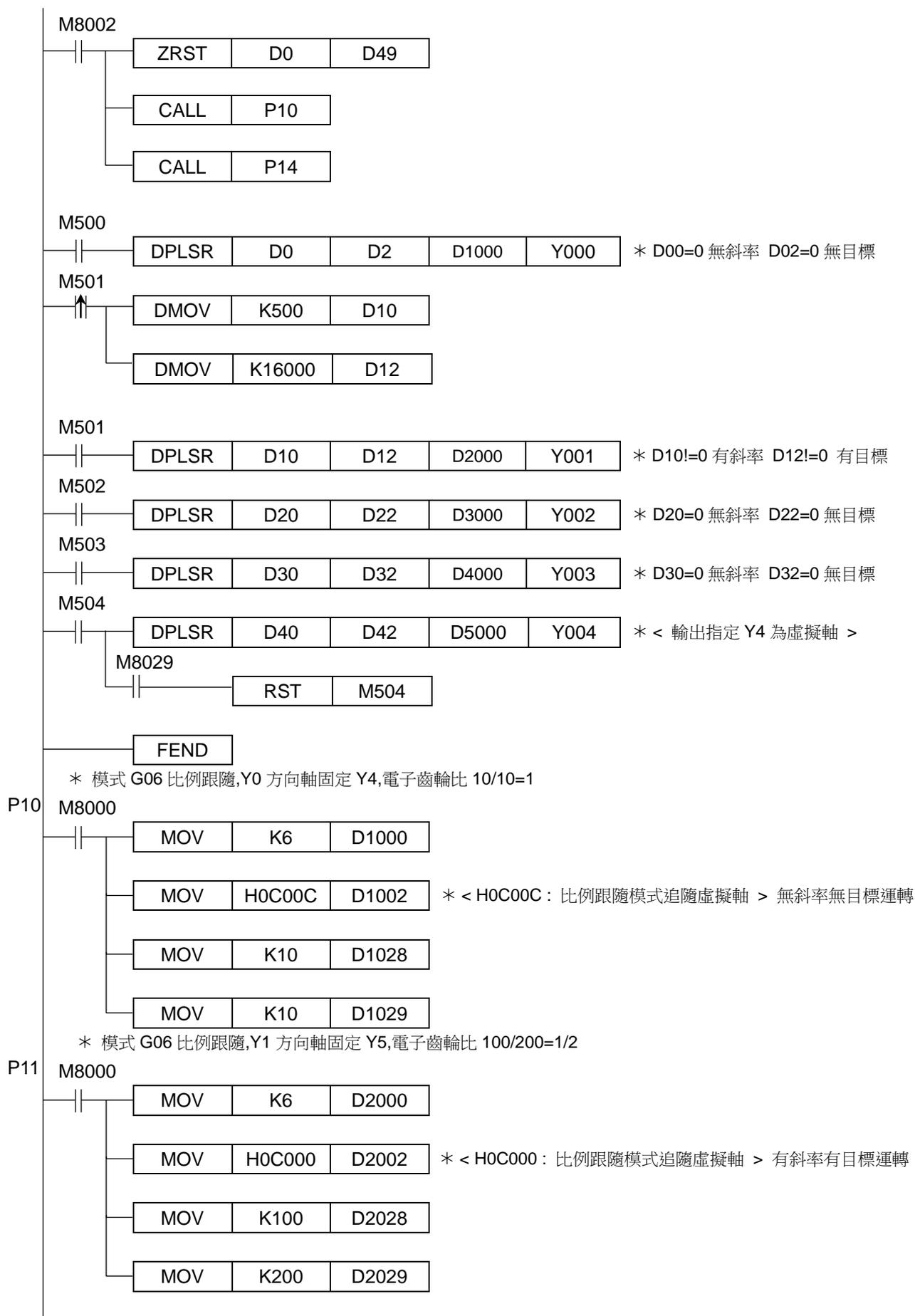
## ※ 命令碼 00 [G00] 一段位置



※ 命令碼 06 [G06] 比例跟隨(Y0 軸方向固定為 Y2，Y1 軸方向固定為 Y3)



※ 多軸同動 (若虛擬軸參數 bit2,D5002=1 則為無目標運轉) 實體軸須先驅動



\* 模式 G06 比例跟隨, Y2 方向軸固定 Y6, 電子齒輪比  $320/1280=1/4$

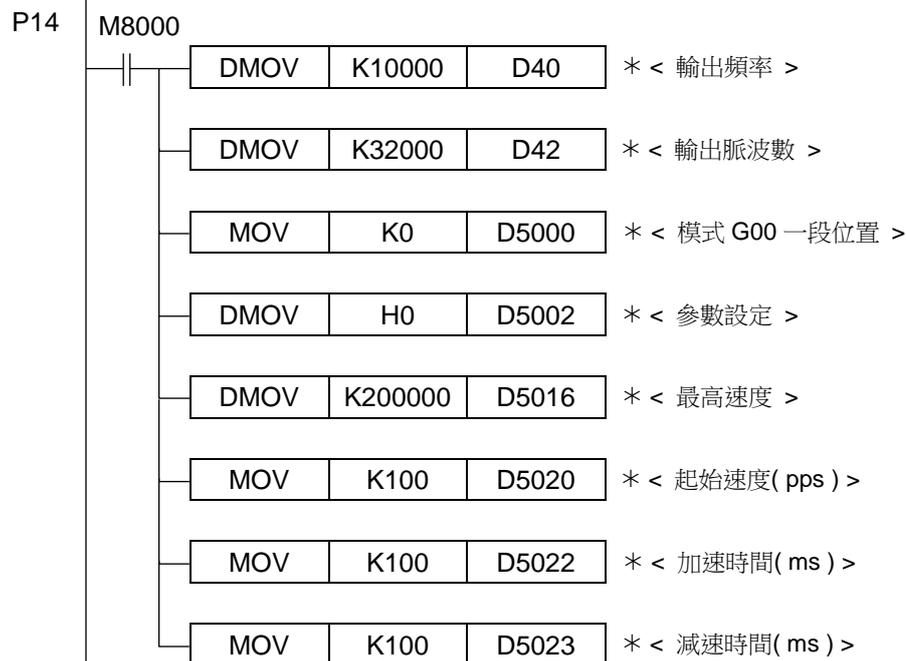


\* 模式 G06 比例跟隨, Y3 方向軸固定 Y7, 電子齒輪比  $401/3208=1/8$



SRET

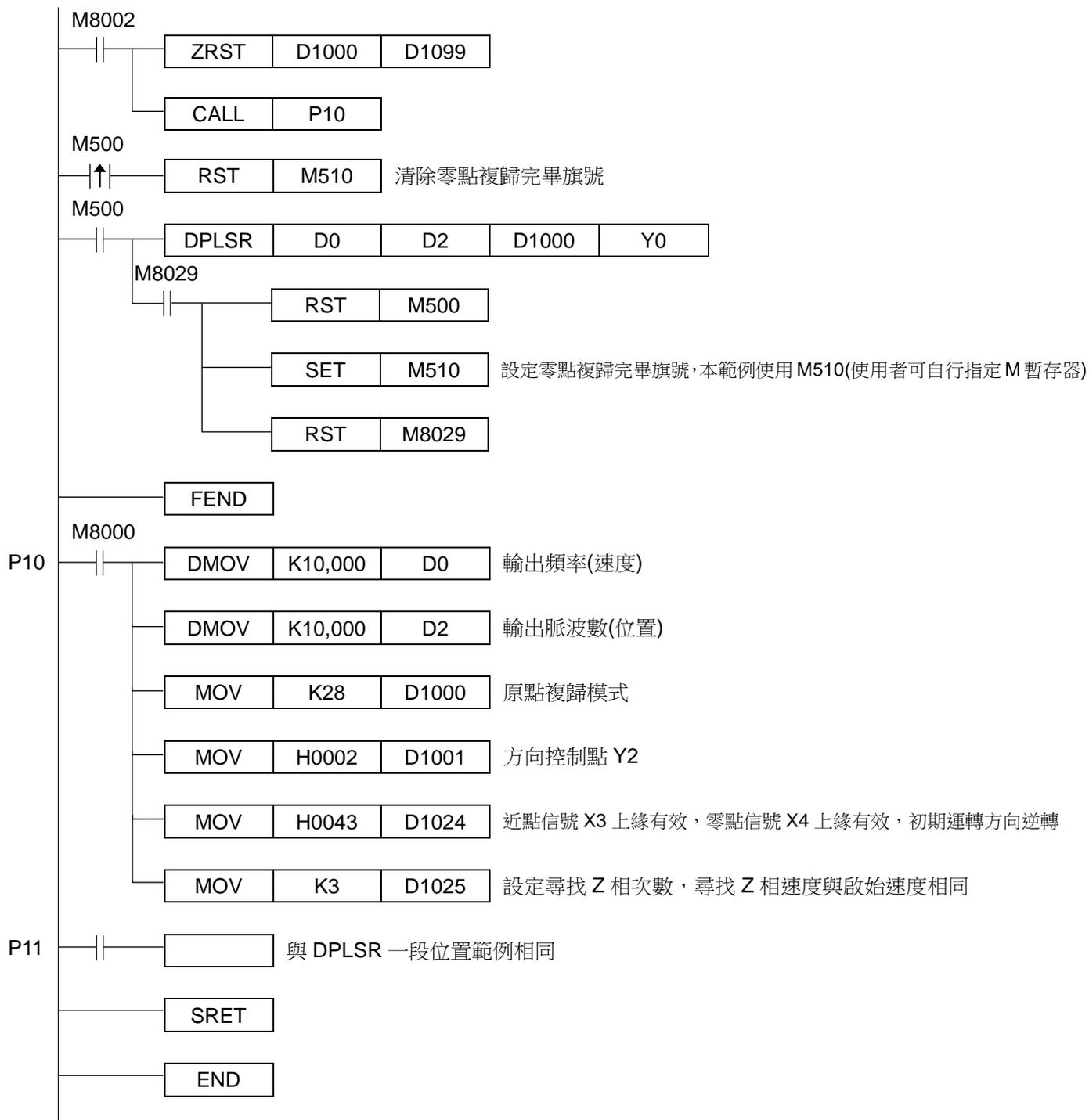
\* 虛擬軸



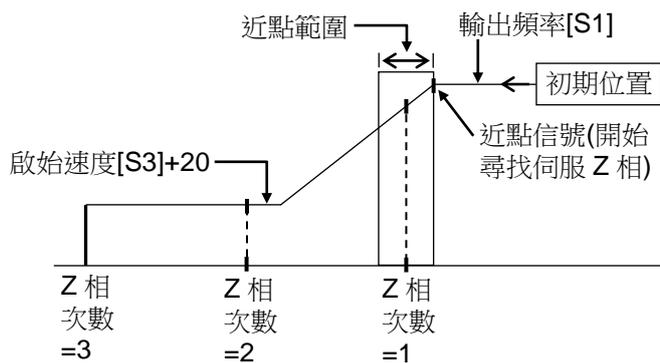
SRET

END

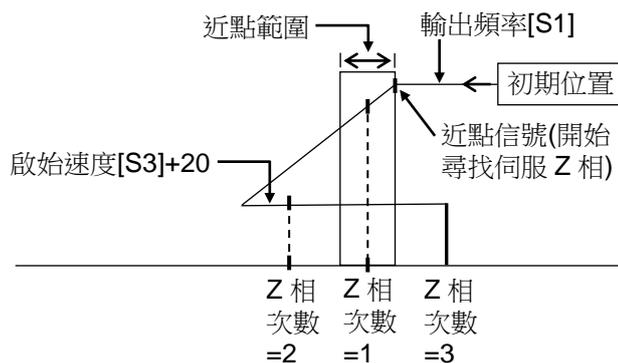
※ 命令碼 28 [G28] 零點復歸(尋找 Z 相次數不為 0)



< 模式 0 > D1024=H0043 相同方向尋找零點信號



< 模式 1 > D1024=H1043 相反方向尋找零點信號



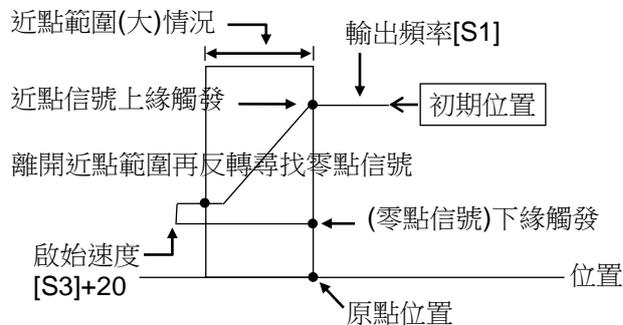
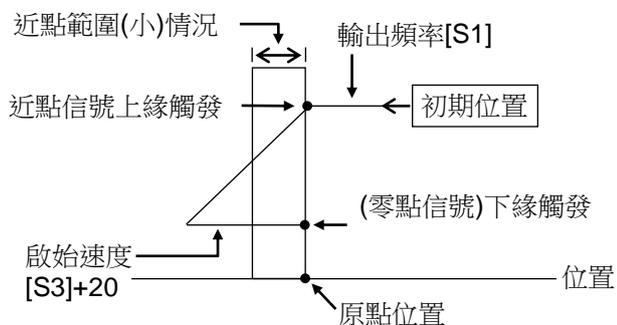
命令碼 28 [G28] 零點複歸(尋找 Z 相次數為 0，近點信號與零點信號需設定成同一點)

程式碼與零點複歸(尋找 Z 相次數不為 0 的模式)相同，只有 D1024, D1025 設定值不同

### << 模式 0 >> 近點確認 減速至起始速度 且需離開近點範圍 立即反轉開始尋找零點

D1024 = H0233 (近點信號 X3 上緣有效，零點信號 X3 下緣有效，初期運轉方向逆轉)

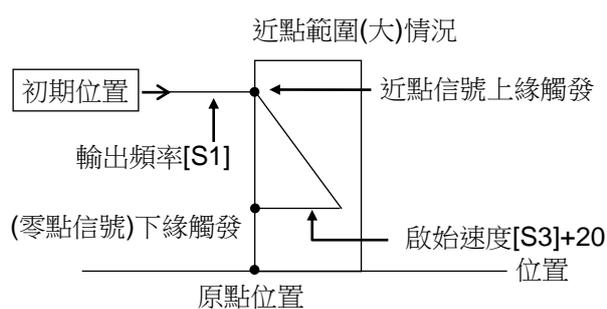
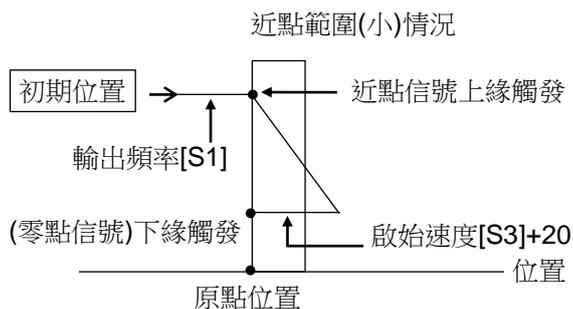
D1025 = K0 (Z 相次數 = 0)



### << 模式 1 >> 近點確認 減速至起始速度 不需離開近點範圍 立即反轉開始尋找零點

D1024 = H1633 (近點信號 X3 上緣有效，零點信號 X3 下緣有效，初期運轉方向正轉)

D1025 = K0 (Z 相次數 = 0)



**FNC(60) 初始狀態 INITIAL STATE**

FNC(60)		16 bits: IST ----- 7 steps
IST		

--	--	--	--

Reserved

**FNC(61) 資料搜尋 DATA SEARCH**

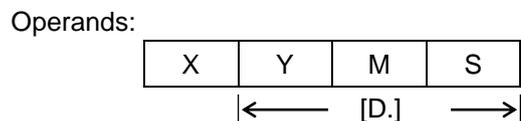
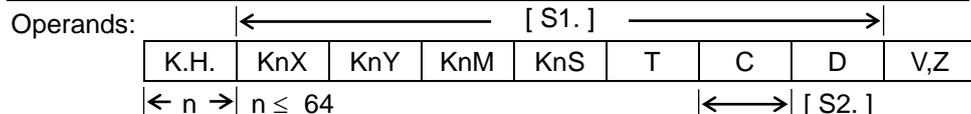
FNC(61)			16 bits: SER(P)----- 9 steps
D	SER	P	32 bits: (D)SER(P) -----17 steps

--	--	--	--

Reserved

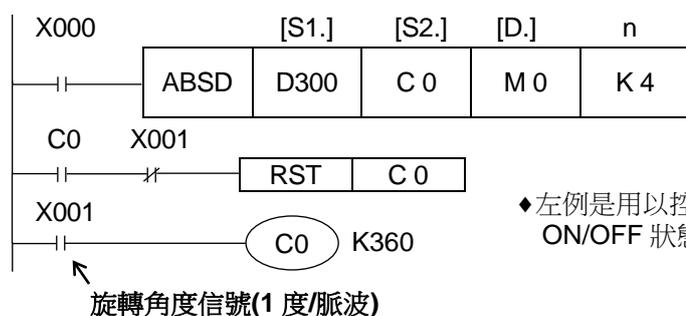
## FNC(62)鼓輪控制 ABSOLUTE DRUM SEQUENCE

FNC(62)		16 bits: ABSD ----- 9 steps			J2n--	J3n--
D	ABSD	32 bits: (D)ABSD ----- 17 steps				



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●	●								
[S2.]													●										
[D.]		●	●	●										●									
[n.]																				●	●		

$n \leq 64$



本命令用以對計數器值產生一多變的輸出型式。可做角度檢出凸輪控制動作。

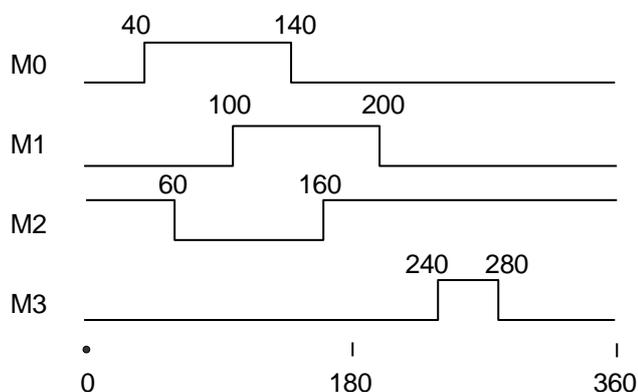
◆左例是用以控制旋轉台，在旋轉一圈內補助繼電器 M0~M3 的 ON/OFF 狀態。

◆ 使用 MOVE 命令把下列的數值寫入 D300~D307

ON 設定數據	OFF 設定數據	輸出點
D300= 40	D301= 140	M0
D302= 100	D303= 200	M1
D304= 160	D305= 60	M2
D306= 240	D307= 280	M3

把 Turn ON 點的數值放在編號為偶數的 D 要素中，並且把 Turn OFF 點的數值放在編號為奇數的 D 要素中。

◆ X0 為 ON 時，M0~M3 的變化為如下所描述。Turn ON 點及 Turn OFF 點的數值能夠重新更改寫入到 D300~D307。



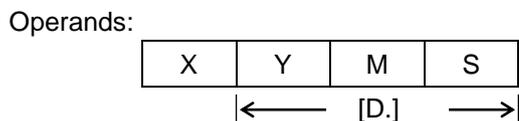
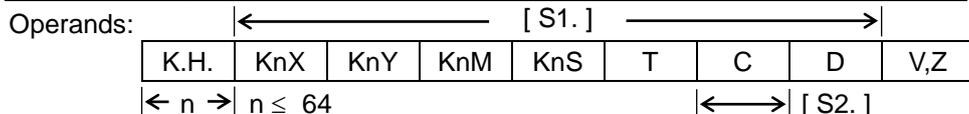
◆輸出點的號碼由[D.]的設定值決定。  
◆當 X0 變為 OFF 時，輸出保持不變。

◆ ABSD 命令在一程式中僅能使用一次。

◆ 若在[S.]中指定高速計數器時，亦可使用(D)ABSD 命令。但此時對於計數器之現在值，其輸出型態將因其掃描週期而產生延遲現象，建議使用 HSZ 命令中之 Table 高速比較模式。

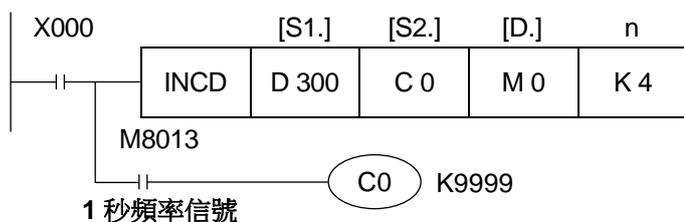
## FNC(63)鼓輪控制 INCREMENTAL DRUM SEQUENCE

FNC(63)		16 bits: INCD ----- 9 steps												J2n--		J3n--	
INCD																	



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●									
[S2.]													●										
[D.]		●	●	●																			
[n.]																				●	●		

$n \leq 64$

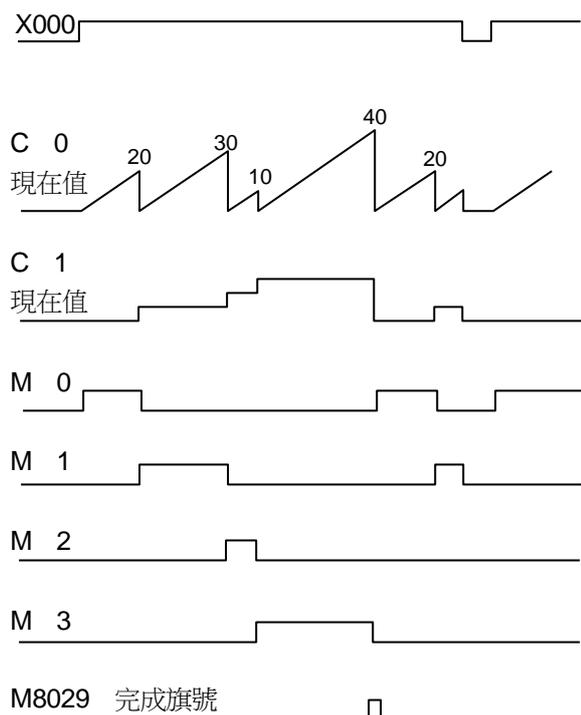


本命令在使用一對計數器情況下用以產生一多變的輸出。

如下為四點 (M0~M3) 的控制範圍。

◆使用 MOVE 命令把下列的數值預先寫入[S1.]中。

D300 = 20      D302 = 10  
D301 = 30      D303 = 40



- ◆當 C0 的計數值達到 D300~D303 的設定值時，C0 自動地依序被複置。
- ◆C1 計算 C0 複置發生的次數。
- ◆M0~M3 依據 C1 的計數值依序動作。
- ◆在完成由“n”所設定次數的最後一次處理後，旗號 M8029 變為 ON。如此相定的迴圈將一再重複。
- ◆在 X0 為 OFF 時 C0 及 C1 被清除，M0~M3 變為 OFF，然後在 X0 又變為 ON 時重新運作。
- ◆INCD 命令在一程式中僅能使用一次。

**FNC(64)教導式計時器 TECHING TIMER**

FNC(64)		16 bits: TTMR ----- 5 steps							
	TTMR								

Reserved

**FNC(65)特殊計時器 SPECIAL TIMER**

FNC(65)		16 bits: STMR ----- 7 steps							
	STMR								

Reserved

**FNC(66)交替式輸出 ALTERNATE OUTPUT**

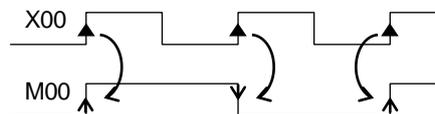
FNC(66)			16 bits: ALT(P) ----- 3 steps						J2n--	J3n--
	ALT	P								

Operands:      ← [ D. ] →

X	Y	M	S
---	---	---	---

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[D.]		●	●	●																			

影響旗號:



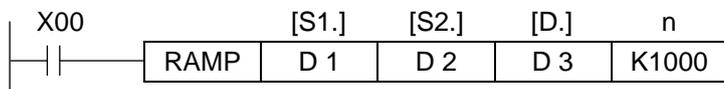
◆ 連續使用此指令，可得到多段除頻

## FNC(67)斜率信號 RAMP

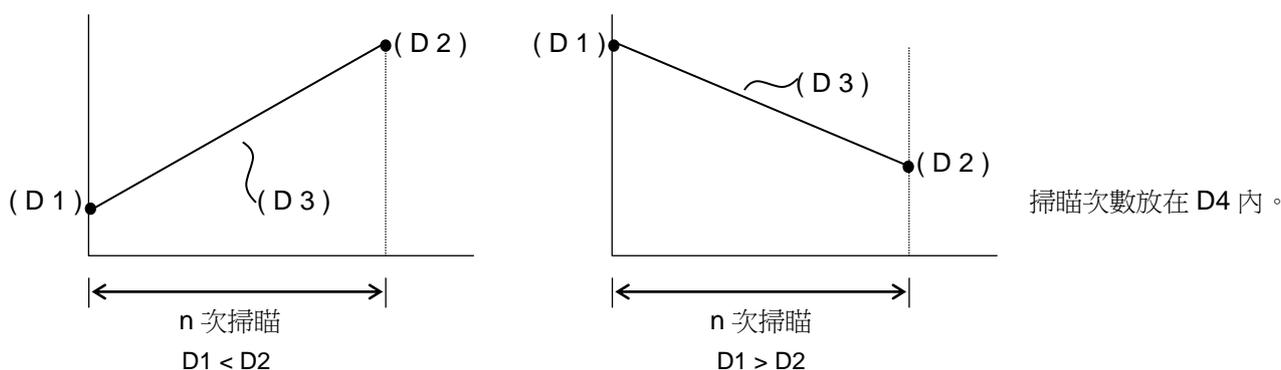
FNC(67)	16 bits: RAMP ----- 9 steps			J2n--	J3n--
RAMP					

Operands: [ S1. ][ S2. ][ D. ]:   
 n:  n = 1 to 32,767

影響旗號: M8029



- 在 X0 為 ON 時分別把 D1 及 D2 寫入初值及終值。D3 內的數值則由 D1 的設定值逐漸向 D2 的設定值遞增，而所需的時間(掃瞄次數)，則由“n”指定。



- 在 M8029 被驅動後把 1 次掃瞄時間值(比實際掃瞄時間稍長)寫入 D8039 中，則 PLC 將進入定掃瞄模式。例如，上例中 n = K1000，若掃瞄週期設定為 20msec，則 D3 內的數值將在 20 秒內由 D1 的設定值變為 D2 的設定值。
- 假如 X0 在動作途中變為 OFF，則傾斜信號的動作也半途停止，若 X0 又再次為 ON 時，D4 被清除，D3 由 D1 設定值重新開始。
- 在執行結束後，旗號 M8029 動作，然後 D3 的值也復歸為 D1 的值。
- 在起始 / 結束點的控制可以由結合 RAMP 命令及模擬輸出加以執行。
- 在 X0 為 ON 時進入 RUN 狀態，若 D4 具停電保持作用，則應先將 D4 清除。

## FNC(68)旋轉控制 ROTARY CONTROL

FNC(68)	16 bits: ROTC ----- 9 steps				
ROTC					

Reserved

## FNC(69)資料排列 SORT

FNC(69)	16 bits: SORT ----- 11 steps				
SORT					

Reserved





## FNC(72)指撥開關 DIGITAL SWITCH

FNC(72)		16 bits: DSW ----- 9 Steps										J2n--	J3n--
DSW													

Operands:  $\leftarrow [n] \rightarrow = 1 \sim 8$        $\leftarrow [D2.] \rightarrow$

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

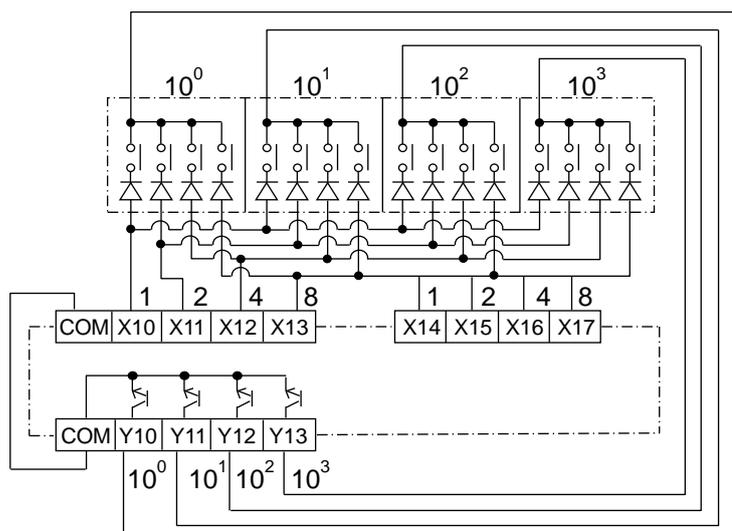
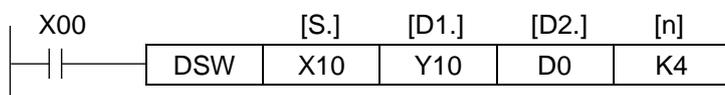
Operands:  $\leftarrow [S.] \rightarrow$   $\leftarrow [D1.] \rightarrow$

X	Y	M	S
---	---	---	---

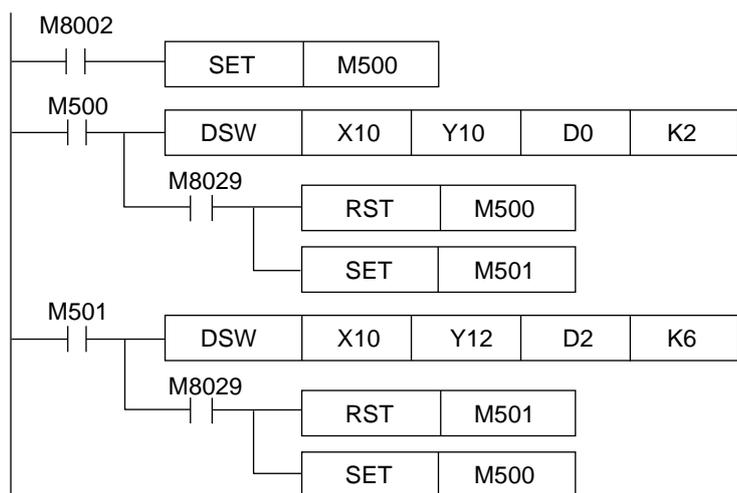
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]	●																						
[D1.]		●																					
[D2.]												●	●	●		●	●						
[n.]																				●	●		

n=1~8

影響旗號:M8029

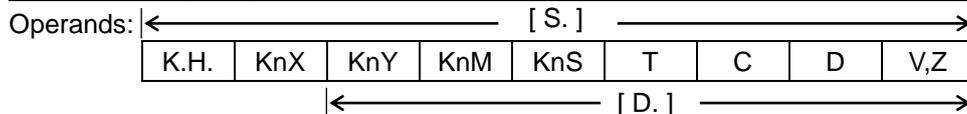


- ◆ 本命令可用 n=1~8 點輸出點，以及 4 點輸入點讀取 n 個(n=1~8)指撥開關的數值，若讀入數大於 32 位(n≥5)則[D2.]自動佔用下一組暫存器。
- ◆ BCD 4 位元數指撥開關 1,2,4,8 接腳連接到 X10~X13 或 X14~X17，[S.]須以 X10,X14,X20,X24...為起始點，[D1.]則可任意指定，但不可超出 1 個 CHANNEL 值，如 n=2 [D1.]不可指定為 Y17，Y10~Y16 都可被指定。
- ◆ 一旦命令被執行則 M8029 清除為"0"，Y10~Y13 依序動作，待執行完畢(數個演算週期後)，M8029 被設定為"1"。
- ◆ 指撥開關上之每 1,2,4,8 接腳須再外接二極體(0.1A/50V)
- ◆ 允許多個 DSW 命令，但一次僅允許一個 DSW 命令及動作，如下圖利用 M8029 來控制兩組 DSW 的命令。



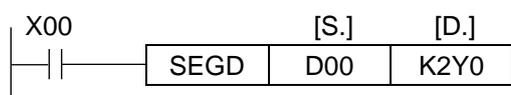
## FNC(73)七段顯示器解碼器 SEVEN SEGMENT DECODER

FNC(73)		16 bits: SEGD(P) ----- 5 steps												J2n--	J3n--				
SEGD	P																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S.]	●							●	●	●	●	●	●	●		●	●			●	●			
[D.]									●	●	●	●	●	●		●	●							

影響旗號:



◆ [S.] 中下 4 位所指定的數值 0~F (十六進位)，被解碼為 7 段顯示用數值後放入[D.]中。[D.]的上 8 位不變。

( S. )		7 段顯示器之構成	( D. )								顯示資料		
16 進制	Bit 組合		b7	b6	b5	b4	b3	b2	b1	b0			
0	0000		0	0	1	1	1	1	1	1	0	0	
1	0001		0	0	0	0	0	1	1	0			
2	0010		0	1	0	1	1	0	1	1			
3	0011		0	1	0	0	1	1	1	1			
4	0100		0	1	1	0	0	1	1	0			
5	0101		0	1	1	0	1	1	0	1			
6	0110		0	1	1	1	1	1	0	1			
7	0111		0	0	1	0	0	1	1	1			
8	1000		0	1	1	1	1	1	1	1			
9	1001		0	1	1	0	1	1	1	1			
A	1010		0	1	1	1	0	1	1	1			
B	1011		0	1	1	1	1	1	0	0			
C	1100		0	0	1	1	1	0	0	1			
D	1101		0	1	0	1	1	1	1	0			
E	1110		0	1	1	1	1	0	0	1			
F	1111		0		1	1	0	0	0	1			

◆ 位要素的開頭(本例為 Y0)或字元要素的最低位元(LSB)均對稱到 b0，且依此類推。

**FNC(74) 栓鎖式七段顯示器 SEVEN SEGMENT WITH LATCH**

FNC(74)		16 bits: SEGL(P) ----- 5 steps													
SEGL	P														

Reserved

**FNC(75) 箭號開關 ARROW SWITCH**

FNC(75)		16 bits: ARWS(P) ----- 9 steps													
ARWS															

Reserved

**FNC(76) ASCII 轉換 ASCII CODE CONVERSION**

FNC(76)		----- 11 steps												J2n--		J3n--	
ASC																	

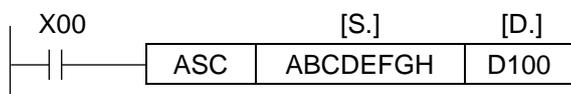
Operands: [S.]: 8 個英文字或阿拉伯數字。

Operands:      |← [ D. ] →|

	T	C	D
--	---	---	---

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[D.]												●	●	●									

影響旗號:



- ◆ 字元“A”~“H”被轉換成 ASCII 碼並放在 D100~D103 中。

M8161 OFF 時

M8161=OFF	上 8 位	下 8 位
D100	“B”	“A”
D101	“D”	“C”
D102	“F”	“E”
D103	“H”	“G”

M8161 ON 時

	上 8 位	下 8 位		上 8 位	下 8 位
D100	0	“A”	D104	0	“E”
D101	0	“B”	D105	0	“F”
D102	0	“C”	D106	0	“G”
D103	0	“D”	D107	0	“H”

**FNC(77) 列印 PRINT**

FNC(77)		16 bits: PR ----- 5 steps													
PR															

Reserved

## FNC(78)FROM 命令

FNC(78)			16 bits: FROM(P) ----- 9 steps			J2n--	J3n--
D	FROM	P	32 bits: (D)FROM(P) ----- 17 steps				

Operands:  $\leftarrow$  [D.]  $\rightarrow$

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

Operands:  $\leftarrow$   $\rightarrow$  m1 = 0 ~ 7 特殊模組號碼  
 m2 = 0 ~ 31 緩衝記憶體 (BFM) 號碼  
 n = 1 ~ 31 轉送點數 (D 命令時=1 ~ 15)

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[m1]														●	●					●	●		
[m2]														●	●					●	●		
[D.]									●	●	●	●	●	●	●	●	●						
[n.]														●	●					●	●		

m1 = 0 ~ 7 特殊模組號碼

m2 = 0 ~ 31 緩衝記憶體 (BFM) 號碼

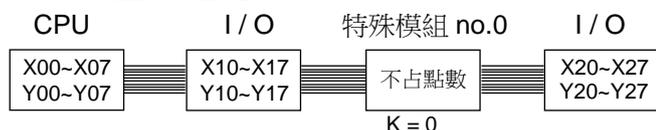
n = 1 ~ 31 轉送點數 (D 命令時=1 ~ 15)

影響旗號:



◆ 當 X00 ON 時，將特殊模組 NO.1 之緩衝記憶體 BFM#29 讀出，轉送到可程式控制器之 M00~M15。

<<特殊裝置 模組號碼 m1>>



- ◆ 特殊模組號碼的排列依靠近主機的順序分別為 NO.0~NO.7
- ◆ 特殊模組不占 I/O 點數且最多可擴充 8 台。
- ◆ 所謂緩衝記憶體 BFM 即為特殊模組與可程式控制器溝通之資料暫存器。

## FNC(79)TO 命令

FNC(79)			16 bits: TO(P) ----- 9 steps			J2n--	J3n--
D	TO	P	32 bits: (D)TO(P) ----- 17 steps				

Operands: |←----- [S.] ----->|

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
------	-----	-----	-----	-----	---	---	---	-----

Operands: |← →| m1 = 0 ~ 7 特殊模組號碼  
 m2 = 0 ~ 31 緩衝記憶體(BFM) 號碼      n = 1 ~ 31 轉送點數(D 命令時=1 ~ 15)

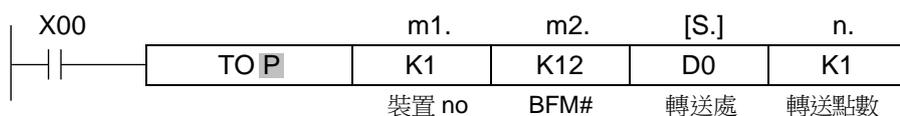
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[m1]														●	●					●	●		
[m2]														●	●					●	●		
[D.]								●	●	●	●	●	●	●	●	●	●						
[n.]														●	●					●	●		

m1 = 0 ~ 7 特殊模組號碼

m2 = 0 ~ 31 緩衝記憶體(BFM) 號碼

n = 1 ~ 31 轉送點數(D 命令時=1 ~ 15)

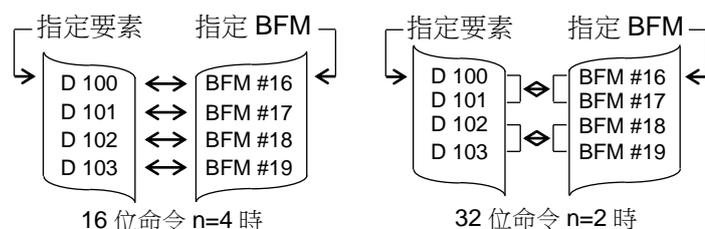
影響旗號:



◆ 當 X00 ON 時，將 D0 的 16 位元資料寫入特殊模組 NO.1 之緩衝記憶體 BFM#12。

◆ 此命令儘量使用脈波命令，低掃瞄週期時間。

<< 轉送點數 n >>



## FNC(80)資料通訊 COMMUNICATION

FNC(80)		16 bits: RS ----- 9 steps									J2n--		J3n--	
RS														

Operands: ← S, →

K.H.	KnX	KnY	KnM	KnS	T	C	D	V,Z
← m,n=1~128							← D.→ m,n	

元件種類	位元元件							位元組合元件					字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S.]														●	●			●						
[m.]														●	●					●	●			
[D.]														●	●			●						
[n.]														●	●					●	●			

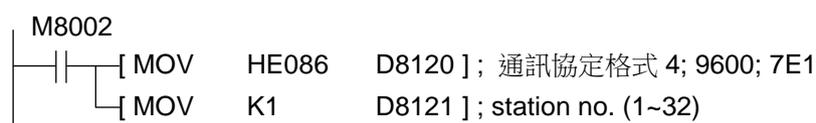
影響旗號:

&lt;&lt; 通訊格式 &gt;&gt; D8120

	內容	0	1
Bit0	數據長度	7 bit	8 bit
Bit1	同位	(00):none, (01):odd, (11):even	
Bit2			
Bit3	停止位	1 bit	2 bit
Bit4	傳送速率 (bps)	(0011):300, (0100):600	
Bit5		(0101):1200, (0110):2400	
Bit6		(0111):4800, (1000):9600	
Bit7		(1001):19200	
Bit8	前端 1	無	D8124
Bit9	終端 1	無	D8125
Bit10	Reserved	-	-
Bit11	Reserved	-	-
Bit12	終端 2	無	D8126
Bit13	通訊模式	自定模式	ModBus
Bit14	ModBus Mode	Ascii Mode	RTU Mode 或 Computer Link
Bit15	Protocol	Format 1	Format 4

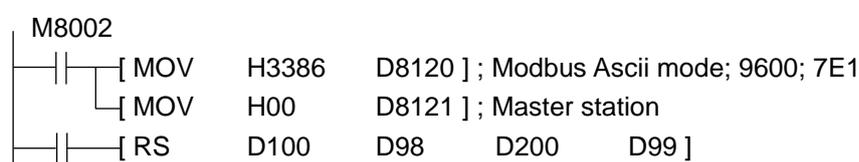
- ◆ 此命令乃利用主機第二通訊埠連接 EXADP232/422/485 通訊板來執行資料的傳送與接收，通訊協定由 D8120 來設定。
- ◆ 由於通訊格式與資料框架(FRAME)均由使用者自行規劃，且可選用不同的通訊介面板，所以此通訊 Port 可連接多種不同通訊格式的機器。
- ◆ 主機運轉開始，先自行檢查是否有書寫 RS 指令。若有，則 Computer Link 模式即無效。通訊協定模式變更為 user define mode 或 Modbus mode。
- ◆ Computer link mode:此模式程式不可書寫 RS 指令，亦即均為被動站，僅設定 D8120 及 D8121 的內容值(D8120 的 bit14 須設為 1)，即可架構多站連結的系統。

例:

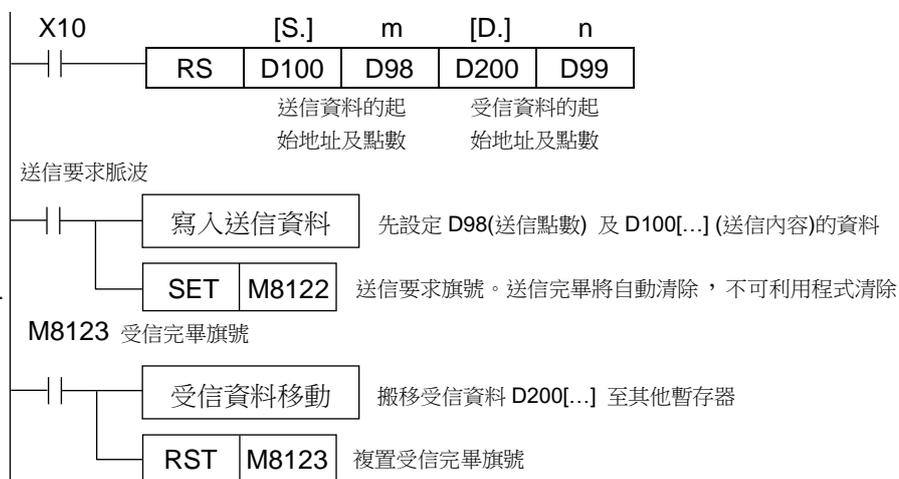


- ◆ Modbus mode:此模式程式須利用 RS 指令來變更通訊協定(D8120 的 bit13 須設為 1)。由於有 RS 指令所以可當為主動站，亦可為被動站。利用 M8122 及 M8123 控制資料傳送與接收。

例:



- ◆ RS 命令被驅動後，若修改 D8120 之設定值，將不被接受。
- ◆ 此命令在程式中不限使用次數，但一掃描週期僅可用一驅動命令，且於切換時須設計 1 個掃描週期以上之 OFF 時間。
- ◆ 此通訊 Port 可為主動模式亦可為被動模式。所以 RS 指令一旦被驅動，則 PLC 即處於等待送信與受信的狀態。
- ◆ 若 RS 指令已使用，則不可再使用 PRUN 指令。



#### << 送信要求 >> M8122

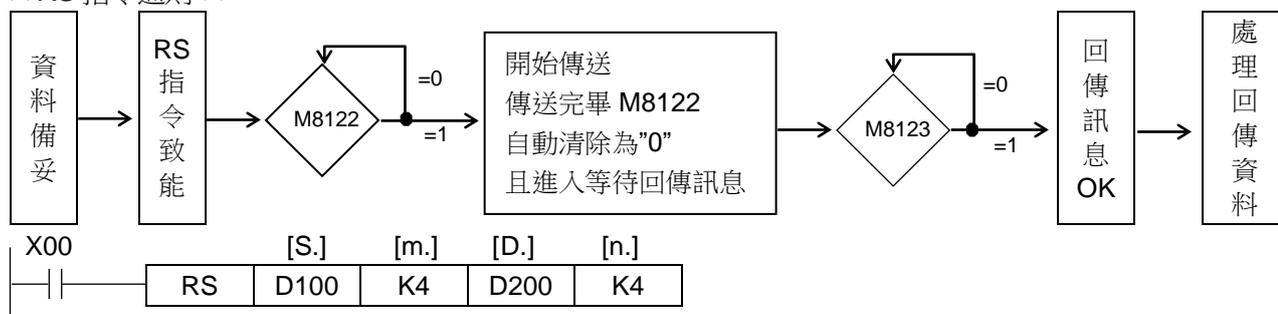
- ◆ 無論在等待受信或受信完畢的狀態，若以脈波指令驅動 M8122 送信要求旗號，則 PLC 將從 D100 起始之 D98 點的資料傳送出，且於送信完畢後 M8122 將自動 Reset。
- ◆ PLC 於資料受信結束後才會執行資料送信，在此期間，若要求送信則等待送信旗號 M8121 將被設定。

#### << 受信完畢 >> M8123

- ◆ PLC 接收資料完畢，則受信完畢旗號 M8123 被設定。請利用程式將 M8123 複置，此時 PLC 將在處於等待受信狀態。
- ◆ M8123 動作中，若接獲送信要求，M8123 不會被清除，但仍會執行資料送信。

#### << 載波檢出 >> M8124

#### << RS 指令通則 >>



< 8 位元資料處理模式 > M8161=ON 為 8 位元模式



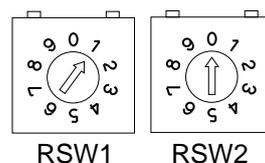
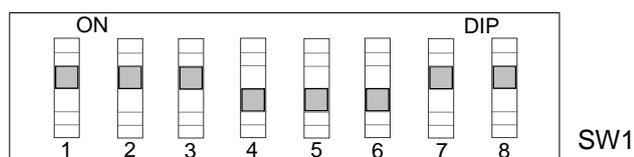
< 16 位元資料處理模式 > M8161=OFF 為 16 位元模式



- ◆ 送信/受信中有錯誤發生，M8063 將被設定，且錯誤內容將會寫入 D8063 中。

## &lt;&lt; MODBUS RTU 模式的應用 &gt;&gt; CRC 偵誤方式

## ◆ EXRM0808R/T 開關



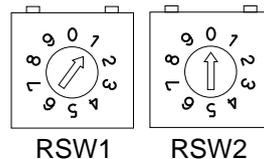
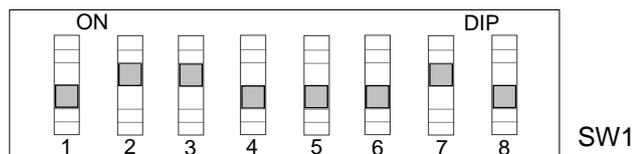
例: 主站與 Remote I/O 模組聯機範常式



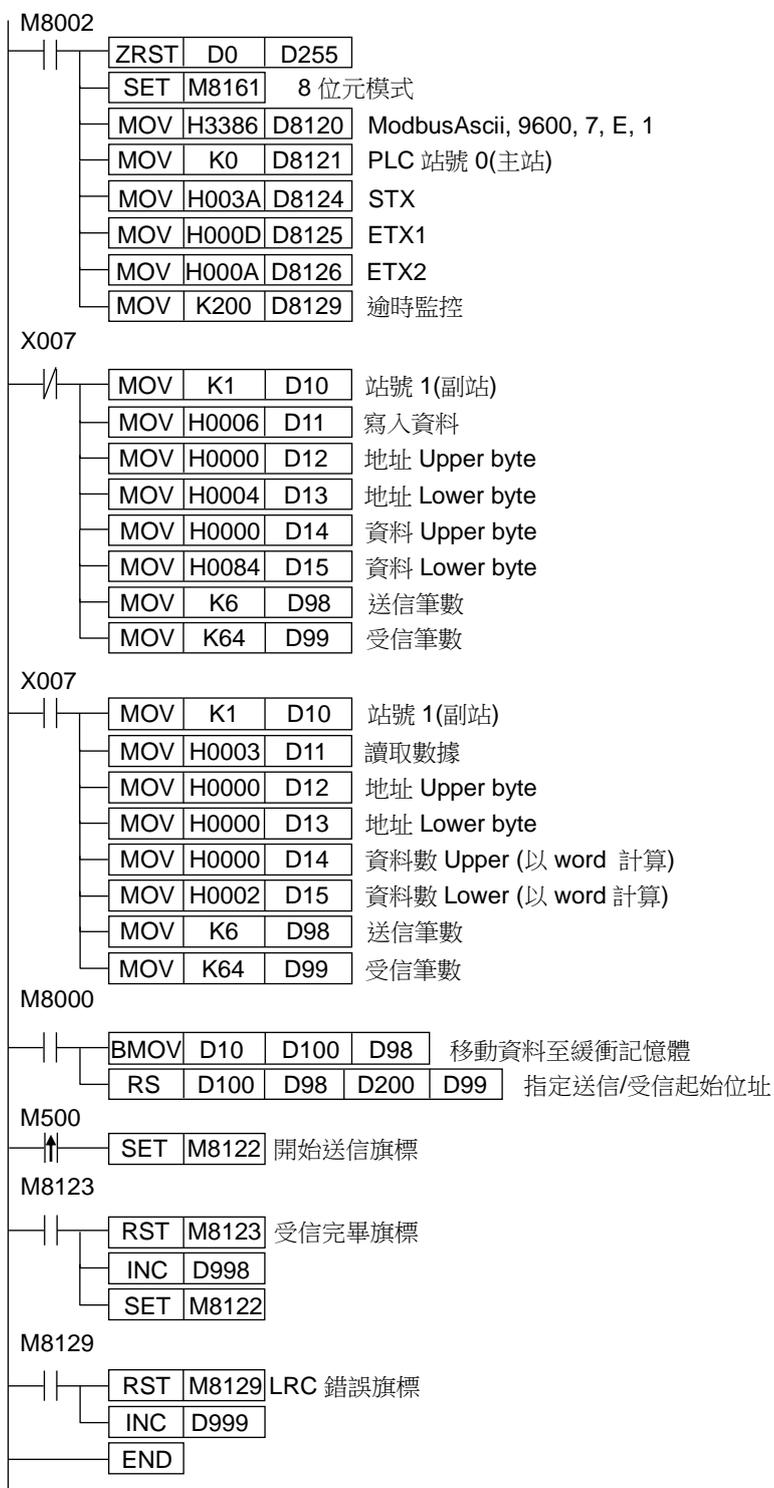
- ◆ 在 ModBus RTU 模式下，請將 D8120 的通訊格式設為無前/終端(無 STX/ETX)，且送信資料點數必須正確。
- ◆ 偵誤值不列入送信筆數，且由 PLC 自動計算，並將結果存入下 2 個暫存器。

## &lt;&lt; MODBUS ASCII 模式的應用 &gt;&gt; LRC 偵誤方式

## ◆ EXRM0808R/T 開關



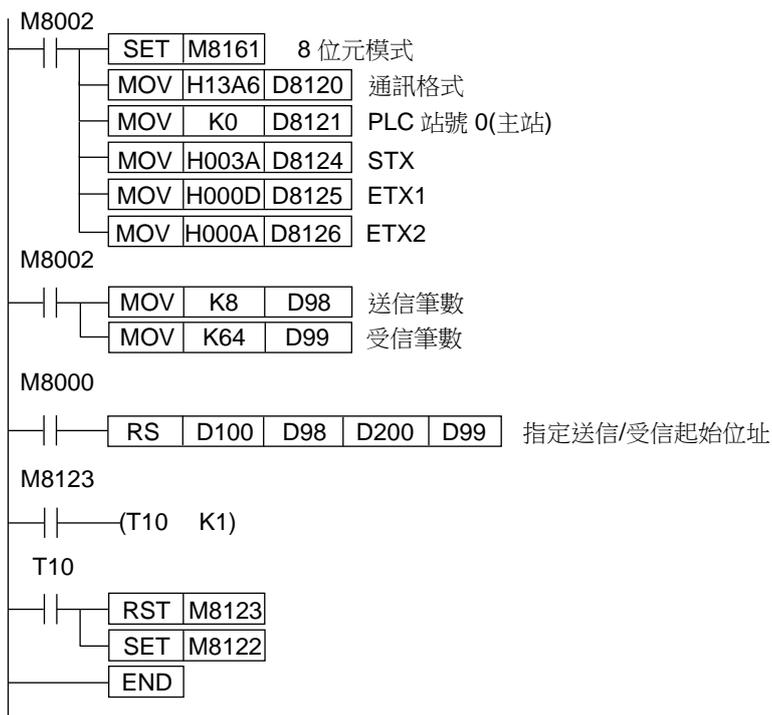
例：主站與 Remote I/O 模組聯機範常式



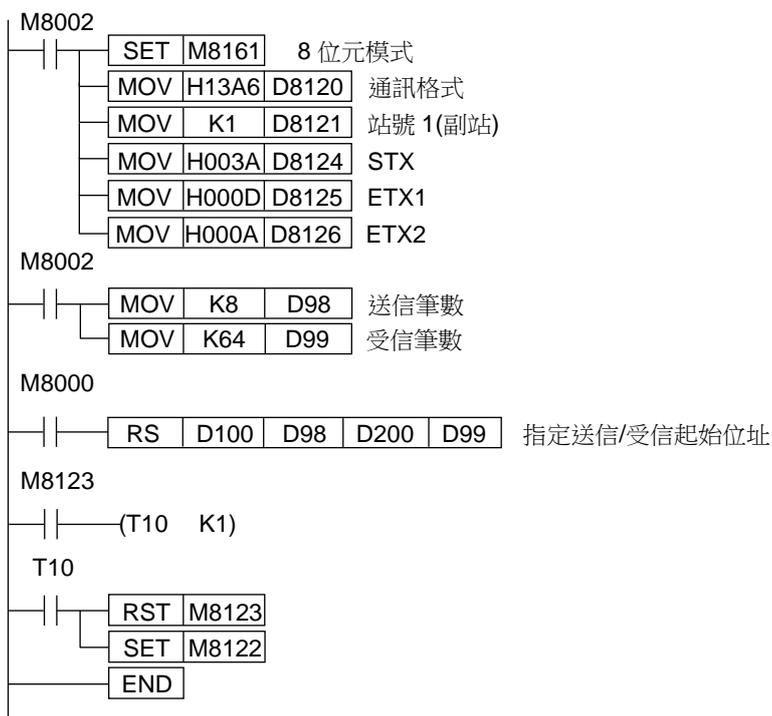
- ◆ 在 ModBus Ascii 模式下，請將 D8120 的通訊格式設為有前/終端(STX/ETX1,2)，且送信資料點數必須正確。
- ◆ 偵誤值不列入送信筆數，且由 PLC 自動計算，並將結果存入下 2 個暫存器。

## &lt;&lt; 使用者自定模式的應用 &gt;&gt; 自定偵誤方式

## 例 1: Ascii 模式主站範常式式

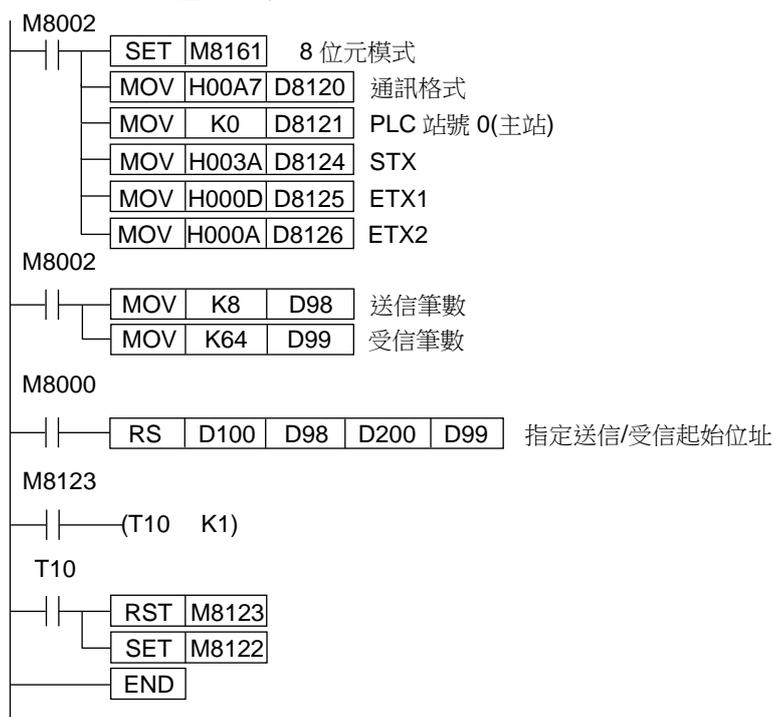


## 副站範常式式

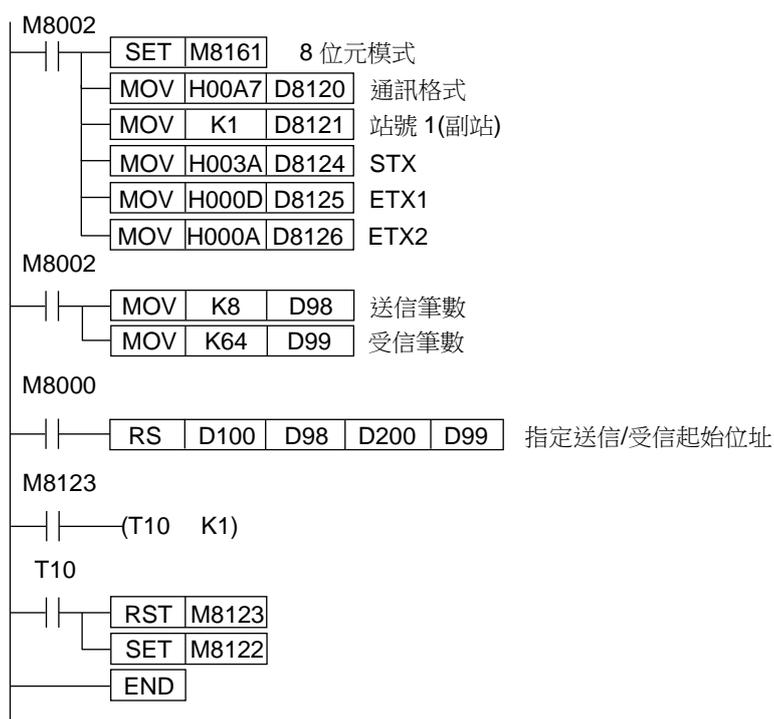


- ◆ 偵誤值須由程式設計人員計算，系統不自動演算。
- ◆ 傳送資料須轉換為 Ascii 存入傳送區。

## 例 2: HEX 模式主站範常式式



## 副站範常式式



◆ 偵誤值須由設計人員計算，系統不自動演算。

## FNC(81)雙機並聯運轉 PARALLEL RUNNING

FNC(81)			16 bits: PRUN(P) ----- 5 Steps			J2n--	J3n--
D	PRUN	P	32 bits: (D)PRUN(P) ----- 9 Steps				

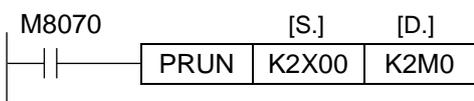
Operands: [S.]: KnX, KnM 最低位元數字須為"0"

[D.]: KnM, KnY 最低位元數字須為"0"

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]								●		●													
[D.]									●	●													

影響旗號: M8073, M8129

親局程式 M8070=1, [S.] [D.]為虛運算元

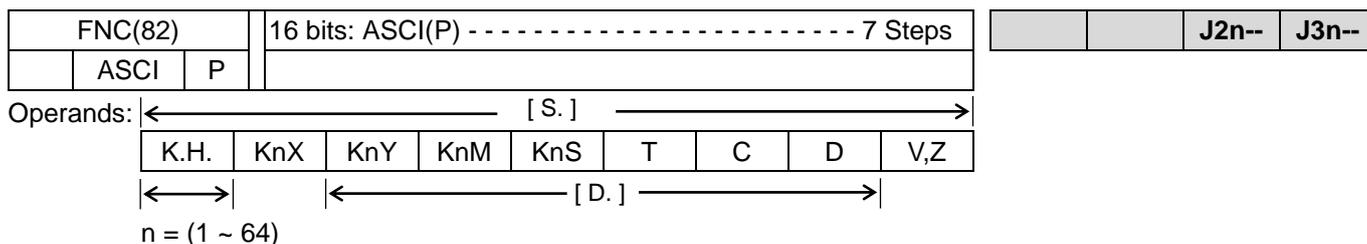


子局程式 M8071=1, [S.] [D.]為虛運算元



- ◆ 親局會將 D490~D497 傳入子局的 D490~D497(M8070=1)。
- ◆ 子局會將 D500~D507 傳入親局的 D500~D507(M8070=0)。
- ◆ 此命令只須設定 M8070 及 M8071, 數據暫存器(D)不須指定, 兩者即會自動通訊。
- ◆ 由於輔助暫存器(M)不對傳, 必要時利用 MOV 命令轉換。
- ◆ 相關參數
  - M8122: 啟動通訊傳輸旗號
  - M8123: 接收完畢旗號
  - M8070: 親局旗號
  - M8071: 子局旗號
  - M8129: 和檢查錯誤旗號
  - M8073: 逾時旗號
  - D8070: 逾時暫存器(ms)
  - D8072: 通訊所費時間(ms)
- ◆ 範常式請參閱力揚應用範例 F081。
- ◆ 若 PRUN 指令已使用, 則不可再使用 RS 指令。

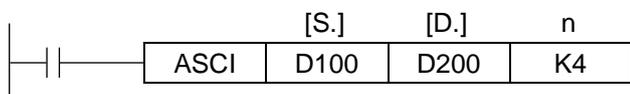
### FNC(82)HEX TO ASCII 轉換



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●		●	●				●	●	
[D.]									●	●	●	●	●	●									
[n.]																					●	●	

n=(1~64)

影響旗號:

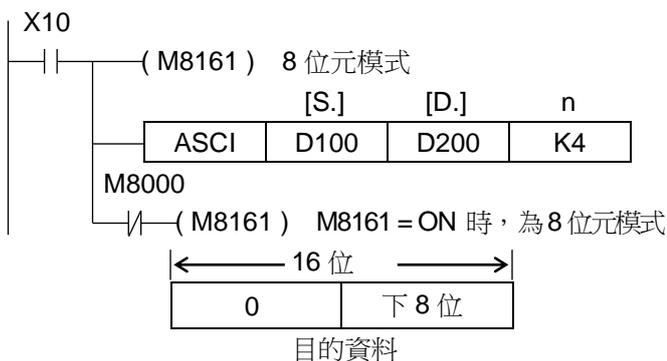


◆ [S.]內的 16 進制資料被轉換為 ASCII 碼傳送到 [D.]的上/下 8 位中。轉換的文字數由 n 指定。

◆ 當 M8161=OFF 時，為 16 位元模式。

例: (D100)=0ABCH，(D101)=1234H

	K1	K2	K3	K4	K5	K6	K7	K8
D200 下	"C"	"B"	"A"	"0"	"4"	"3"	"2"	"1"
D200 上		"C"	"B"	"A"	"0"	"4"	"3"	"2"
D201 下			"C"	"B"	"A"	"0"	"4"	"3"
D201 上				"C"	"B"	"A"	"0"	"4"
D202 下					"C"	"B"	"A"	"0"
D202 上						"C"	"B"	"A"
D203 下							"C"	"B"
D203 上								"C"



◆ [S.]內的 16 進制資料被轉換為 ASCII 碼傳送到[D.]的下 8 位中。轉換的文字數由 n 指定。

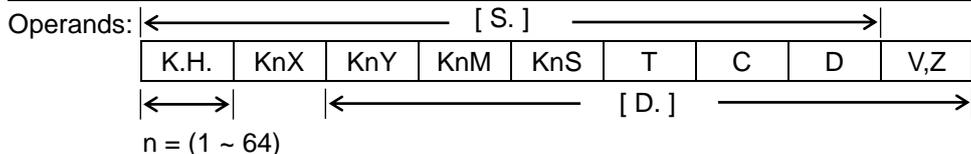
◆ 當 M8161=ON 時，為 8 位元模式。

例: (D100)=0ABCH, (D101)=1234H

	K1	K2	K3	K4	K5	K6	K7	K8
D200 下	"C"	"B"	"A"	"0"	"4"	"3"	"2"	"1"
D201 下		"C"	"B"	"A"	"0"	"4"	"3"	"2"
D202 下			"C"	"B"	"A"	"0"	"4"	"3"
D203 下				"C"	"B"	"A"	"0"	"4"
D204 下					"C"	"B"	"A"	"0"
D205 下						"C"	"B"	"A"
D206 下							"C"	"B"
D207 下								"C"

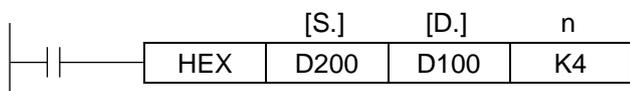
## FNC(83)ASCII TO HEX 轉換

FNC(83)		16 bits: HEX(P) ----- 7 Steps									J2n--	J3n--
HEX	P											



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數		
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E			
[S.]								●	●	●	●	●	●	●									●	●	
[D.]									●	●	●	●	●	●		●	●								
[n.]																							●	●	

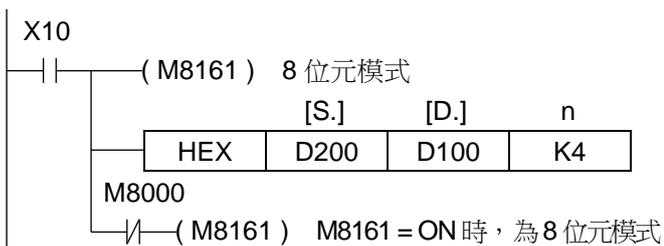
影響旗號:



- ◆ [S.]內上/下 8 位的 ASCII 碼被轉換為 16 進制資料，以每 4 筆為單位傳送到[D.]中。轉換的文字數由 n 指定。
- ◆ 當 M8161=OFF 時，為 16 位元模式。

例: D200 下="0"，D200 上="A"，D201 下="B"，D201 上="C"  
D202 下="1"，D202 上="2"，D203 下="3"，D203 上="4"

	D102	D101	D100
K1			0H
K2			0AH
K3			0ABH
K4			0ABCH
K5		0H	ABC1H
K6		0AH	BC12H
K7		0ABH	C123H
K8		0ABCH	1234H



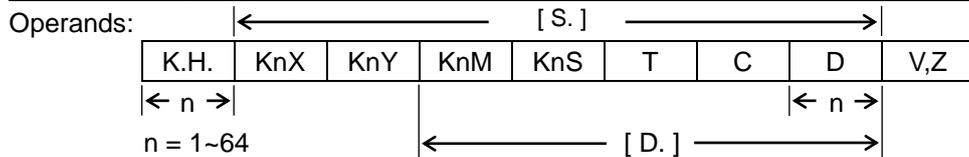
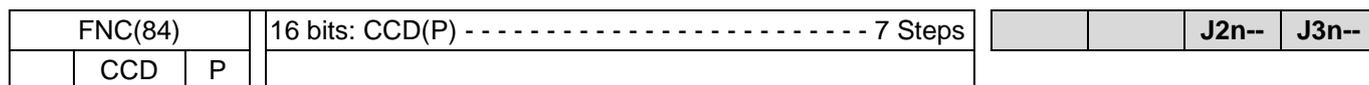
- ◆ [S.]內下 8 位的 ASCII 碼被轉換為 16 進制資料，以每 4 筆為單位傳送到[D.]中。轉換的文字數由 n 指定。
- ◆ 當 M8161=ON 時，為 8 位元模式。

例: D200="0", D201="A", D202="B", D203="C"

D204="1", D205="2", D206="3", D207="4"

	D102	D101	D100
K1			0H
K2			0AH
K3			0ABH
K4			0ABCH
K5		0H	ABC1H
K6		0AH	BC12H
K7		0ABH	C123H
K8		0ABCH	1234H

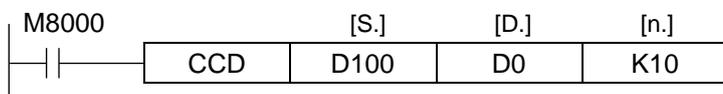
### FNC(84)檢查碼 CHECK CODE



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]								●	●	●	●	●	●	●	●								
[D.]										●	●	●	●	●	●								
[n.]														●	●						●	●	

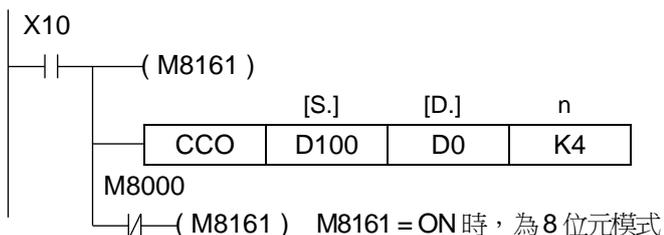
n=1~64

影響旗號:



◆ 將所指定之起始位址之 n 筆資料(8 位元)，其總和→D00，水準極性 Vertical Parity→D01([D.]+1)。

M8161=OFF 16 位元模式									
(S.)		Bit Pattern							
D100 L	K100	0	1	1	0	0	1	0	0
D100 H	K111	0	1	1	0	1	1	1	1
D101 L	K100	0	1	1	0	0	1	0	0
D101 H	K98	0	1	1	0	0	0	1	0
D102 L	K123	0	1	1	1	1	0	1	1
D102 H	K66	0	1	0	0	0	0	1	0
D103 L	K100	0	1	1	0	0	1	0	0
D103 H	K95	0	1	0	1	1	1	1	1
D104 L	K210	1	1	0	1	0	0	1	0
D104 H	K88	0	1	0	1	1	0	0	0
Vertical parity		1	0	0	0	0	1	0	1
Sum	K1091								

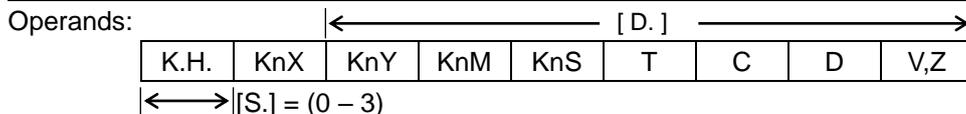


- ◆ 將所指定之起始位址之 n 筆資料(僅 8 位元)，其總和→D00，水準極性 Vertical Parity→D01([D.] +1).

(S.)		M8161=ON 8 位元模式							
		Bit Pattern							
D100	K100	0	1	1	0	0	1	0	0
D101	K111	0	1	1	0	1	1	1	1
D102	K100	0	1	1	0	0	1	0	0
D103	K98	0	1	1	0	0	0	1	0
D104	K123	0	1	1	1	1	0	1	1
D105	K66	0	1	0	0	0	0	1	0
D106	K100	0	1	1	0	0	1	0	0
D107	K95	0	1	0	1	1	1	1	1
D108	K210	1	1	0	1	0	0	1	0
D109	K88	0	1	0	1	1	0	0	0
Vertical parity		1	0	0	0	0	1	0	1
SUM	K1091								

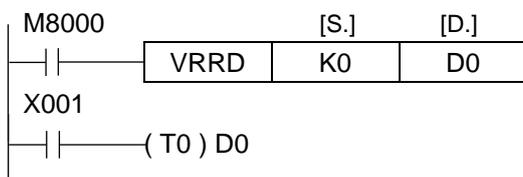
## FNC(85)旋鈕讀取 VOLUME READ

FNC(85)		16 bits: VRRD(P) ----- 5 Steps												J2n--	J3n--				
VRRD	P																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]														●	●						●	●	
[D.]									●	●	●	●	●	●	●	●	●						

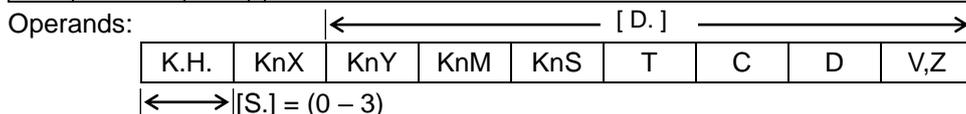
影響旗號:



- ◆ 讀取被指定的 VR 類比輸入且轉換為 8 位二進位碼(0 ~ 255)再存入目的字元要素中。(K=0 時讀取 VR1，K=1 時讀取 VR2，K=2 時讀取 VR3，K=3 時讀取 VR4)
- ◆ 目的要素[D.]的內容可作為 Timer 或 Counter 的設定值。

## FNC(86)旋鈕刻度 VOLUME SCALE

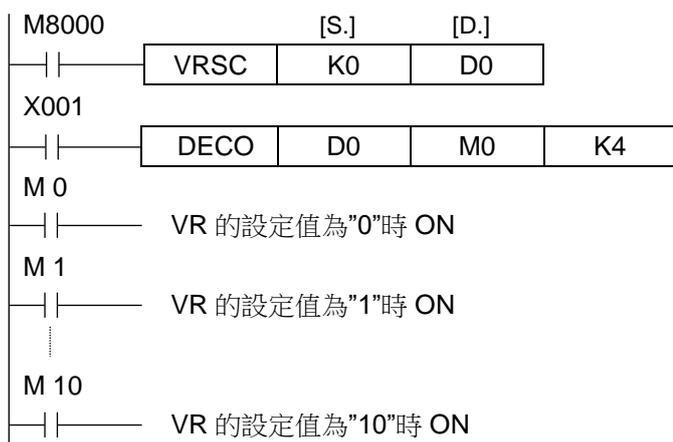
FNC(86)		16 bits: VRSC(P) ----- 5 Steps												J2n--	J3n--				
VRSC	P																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S.]																					●	●	
[D.]									●	●	●	●	●	●	●	●	●						

[S.] = (0-3)

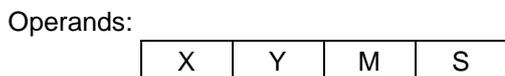
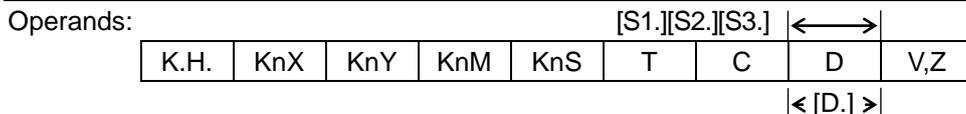
影響旗號:



- ◆ 讀取被指定的 VR 類比輸入且轉換為 8 位二進位碼(0 ~ 255)除以 16 取四捨五入再存入目的字元要素中(0~15)。
- ◆ 此指令可將 VR 當作為 16 段的旋轉開關。

## FNC(88)PID 演算

FNC(88)		16 bits: PID ----- 9 Steps												J2n--		J3n--	
PID																	



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S1.]														●	●								
[S2.]														●	●								
[S3.]														●	●								
[D.]														●	●								

影響旗號:



[S1.]: 為目標值

[S2.]: 為測定值

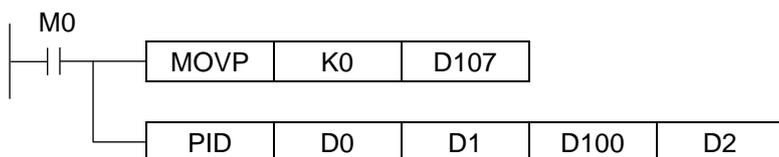
[S3.] ~ [S3.]+6: 為控制參數

[D.]: 輸出值資料暫存器

} 以左邊的設定執行程式，並將演算結果(MV)存入[D.]中

◆ 從指定的[S3.]開始會佔用連續 25 點的組件。在本例中，則佔用 D100~D124。

◆ 初次執行時，須先清除[S3.]+7 的內容為 0



◆ 在開始執行 PID 演算前，必須先以 MOV 等命令寫入 PID 控制用的參數設定值。

[S3.]	取樣時間(Ts)	1~32767 (ms) (不可設定比掃描時間短)
[S3.] + 1	動作方向(ACT)	BIT0: 0: 正動作; 1: 逆動作 BIT1: 0: 無輸入變動量警報; 1: 具輸入變動量警報 BIT2: 0: 無輸出變動量警報; 1: 具輸出變動量警報 BIT3: 保留 BIT4: 保留 BIT5: 0: 無輸出限制; 1: 具輸出限制 BIT6 ~ BIT15: 保留
[S3.] + 2	輸入濾波常數( $\alpha$ )	0 ~ 99 (%)
[S3.] + 3	比例增益(Kp)	1 ~ 32767 (%)
[S3.] + 4	積分時間(Ti)	1 ~ 32767 ( x 100ms ), 0 時為無窮大(無積分動作)
[S3.] + 5	微分增益(Kd)	0 ~ 100 (%)
[S3.] + 6	微分時間(Td)	1 ~ 32767 ( x 10ms ), 0 為無微分動作

- [S3.] + 7 } 執行 PID 演算時，內部處理用  
 [S3.] + 19 }  
 [S3.] + 20 系統保留  
 [S3.] + 21 系統保留  
 [S3.] + 22 輸出最大值限制，[S3.]+1 的 BIT5=1 時有效  
 [S3.] + 23 輸出最小值限制，[S3.]+1 的 BIT5=1 時有效  
 [S3.] + 24 系統保留

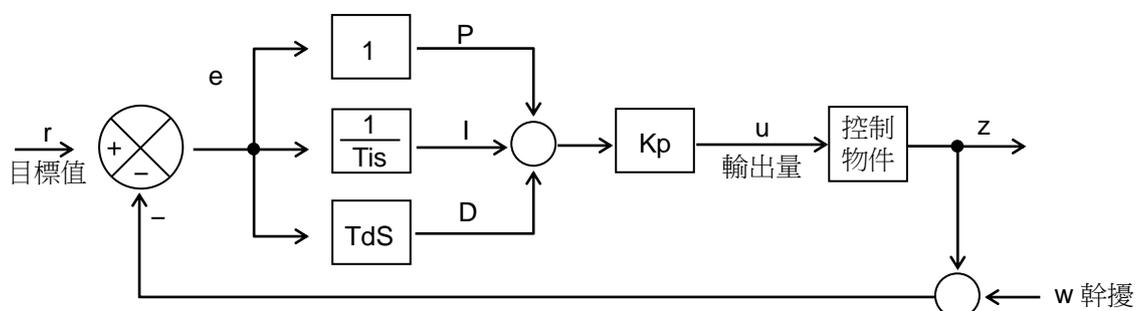
◆ PID 指令的基本演算：

本指令依照速度形，測定微分形演算式，執行 PID 演算。

在 PID 的控制，依[S3.]所指定“動作方向”的內容，執行正動作或逆動作的演算式。

PID 基本型式:

$$\text{輸出 } u(t) = K_p \left\{ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right\} \quad e(t) = \text{偏差值}$$



FNC(89)					

FNC(90)					

FNC(91)					

FNC(92)					

FNC(93)					

FNC(94)					

FNC(95)					

FNC(96)					

FNC(97)					

FNC(98)					

FNC(99)					

## FNC(102)ZPUSH/ (V, Z)暫存器保存

FNC(102)		16bits: ZPUSH & ZPUSH(P) - - - - - 3 steps																			J3n--	
ZPUSH	P																					

Operands:

							←[D.]→	
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z

Operands:

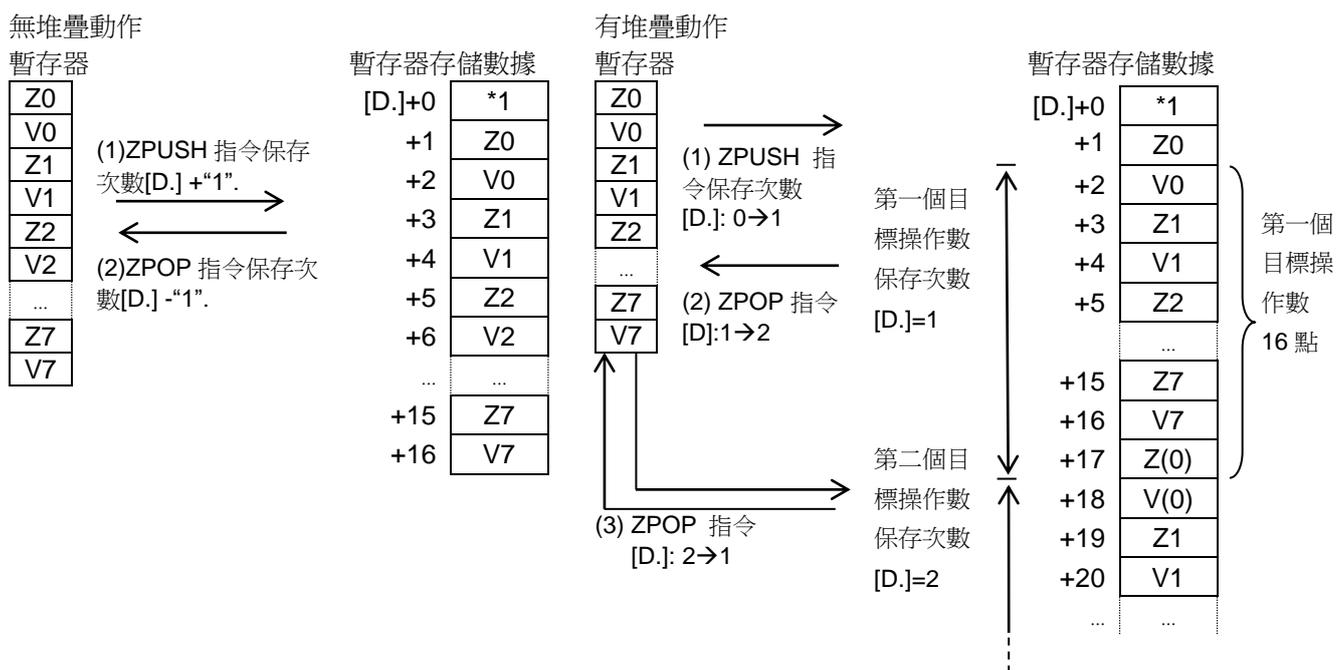
X	Y	M	S
---	---	---	---

元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E	
[D.]														●	●							

影響旗號: None



- 1) 將暫存器 Z0~Z7、V0~V7 的內容全部保存到[D.] 指定暫存器中。保存了暫存器的內容後，保存次數[D.] 就被+1。
- 2) 使用 ZPOP(FNC103) 指令使 V,Z 暫存器數據返回。  
成對使用 ZPUSH(FNC102)、ZPOP (FNC103) 指令。
- 3) 通過對 [D.] 指定相同的暫存器，可以堆疊使用 ZPUSH (FNC102)~ ZPOP(FNC 103)指令。此時，每次執行 ZPUSH (FNC102) 指令時，[D.] 開始使用的區域會每次增加 16 點，因此，請事先確保堆疊使用的次數的區域
- 4) [D.] 以後被保存的數據的結構如下所示。



## 相關指令

指令	內容
ZPOP (FNC103)	使因 ZPUSH (FNC102) 指令而暫時保存的暫存器 V0 to V7、 Z0 to Z7 恢復的指令。

## 注意

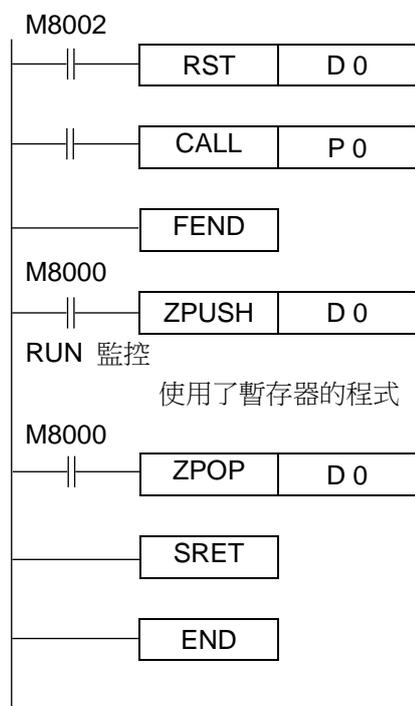
- 無堆疊動作時，執行 ZPUSH (FNC102) 指令前請清除保存次數[D.]
- 有堆疊動作時，請第一次執行前清除保存次數[D.]

## 錯誤

以下情況會發生運算錯誤，錯誤標誌位置 M8067 ON，錯誤代碼保存在 D8067 中。

- ZPUSH (FNC102) 指令中，當[D.] 開始使用的點數範圍超出相應的 D 暫存器的範圍時。(錯誤代碼:K6706)
- 執行 ZPUSH(FNC102) 指令時，[D.] (保存次數) 為負時(錯誤代碼:K6707)

## 範例



## FNC(103)ZPOP/ (V, Z)暫存器的讀取

FNC(103)		16 bits: ZPOP & (D)ZPOP(P) ----- 3 steps													J3n--	
ZPOP	P															

Operands: ←[D.]→

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z
-----	-----	-----	-----	-----	---	---	---	-----

Operands:

X	Y	M	S
---	---	---	---

元件 種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E	
[D.]														●	●							

影響旗號: None



- 1) 將使用ZPUSH(FNC 102)指令被暫時保存到[D.]開始的軟元件中的暫存器V0~V7、Z0~Z7的內容恢復到原來的暫存器中。暫存器的內容恢復後，保存次數[D.]被-1。
- 2) 使用ZPUSH(FNC 102)指令，暫時保存暫存器V,Z的資料。  
ZPUSH(FNC 102)、ZPOP(FNC 103)指令都成對使用。

### 相關指令

指令	內容
ZPUSH (FNC102)	暫時保存暫存器 V0~V7、Z0~Z7 的當前值的指令。

### 錯誤

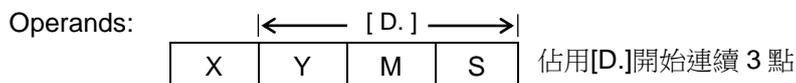
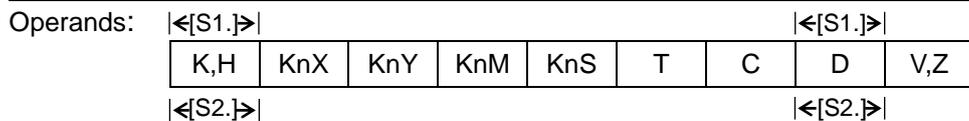
- 以下一些情況下會發生運算錯誤，錯誤標誌位元M8067置ON，錯誤代碼保存在D8067中。
- 執行 ZPOP(FNC 103)指令時，保存次數[D.]的內容為 0 或是負數時。(錯誤代碼: K6706)

### 範例

參考上一指令 ZPUSH 之範例

## 2 進制浮動小數點比較

FNC(110)			32 bits:(D)ECMP & (D)ECMP(P) ----- 13 steps										J2n	J3n--
D	ECMP	P												



元件種類	位元元件						位元組合元件					字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]														●	●					●	●	●	
[S2.]														●	●					●	●	●	
[D.]		●	●	●																			

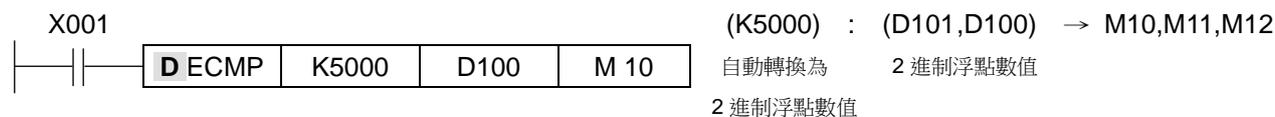
[D.]: 佔用[D.]開始連續 3 點

影響旗號: None



即使將 X001 OFF，不執行 ECMP 命令，但是 M0-M2 仍會保持 X001 OFF 前的狀態。

- ◆ 將 2 個來源組件[S1.]及[S2.]的 2 進制浮點數值做比較，依比較大小的結果，[D.]開始連續 3 點自動 ON/OFF。
- ◆ 來源運算元，若以常數 K 或 H 指定時，則自動地轉換為 2 進制浮點數值作比較處理。



## FNC(111)2 進制浮動小數點區域比較

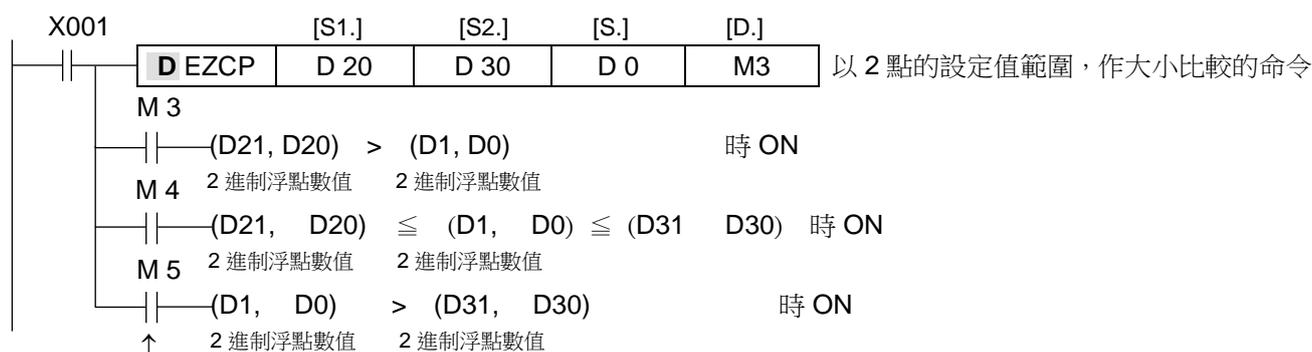
FNC(111)																				J2n	J3n--
D	EZCP	P	32 bits:(D)EZCP & (D)EZCP(P) ----- 17 steps																		



元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E	
[S1.]														●	●					●	●	
[S2.]														●	●					●	●	
[S.]														●	●					●	●	
[D.]		●	●	●																		

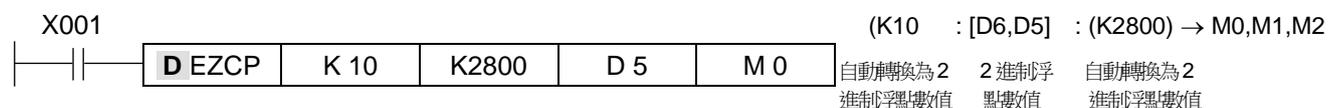
[D.]佔用開始連續 3 點，[S1.], [S2.]請設定為[S1.]≤ [S2.]

影響旗號: None



即使將 X001 OFF，不執行 ECMP 命令，但是 M3~M5 仍會保持 X001 OFF 前的狀態。

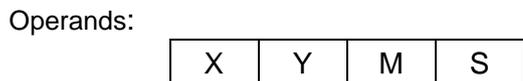
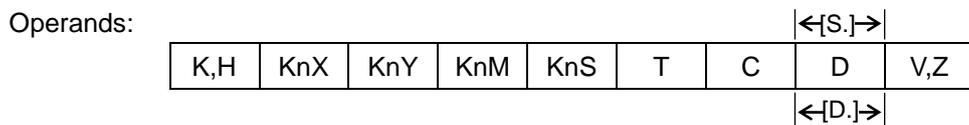
- ◆ 以 2 進制浮點數值所指定的上下 2 點的範圍和 [S1.], [S2.]+1 的內容作比較，依比較大小的結果 [D.] 開始連續 3 點自動 ON/OFF。
- ◆ 來源的運算元若以常數 K 或 H 指定時，則自動地將其轉換成 2 進制浮點數值作比較處理。



- ◆ 請設定 [S1.] ≤ [S2.]。若 [S1.] > [S2.] 時，則 [S2.] 的值視為與 [S1.] 的數值相同。

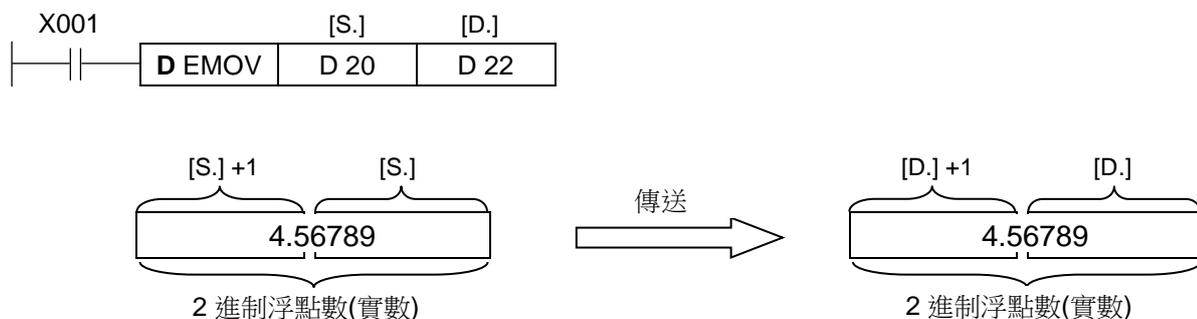
## FNC(112) Floating Point Move 2 進制浮點數資料傳送

FNC(112)																J3n--
D	EMOV	P	32 bits:(D)EMOV & (D)EMOV(P) ----- 9 steps													

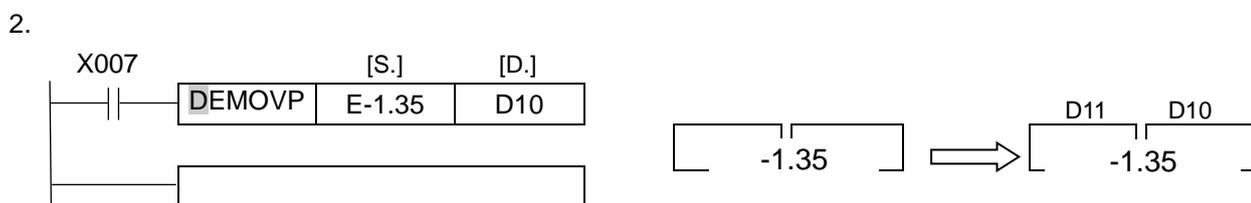
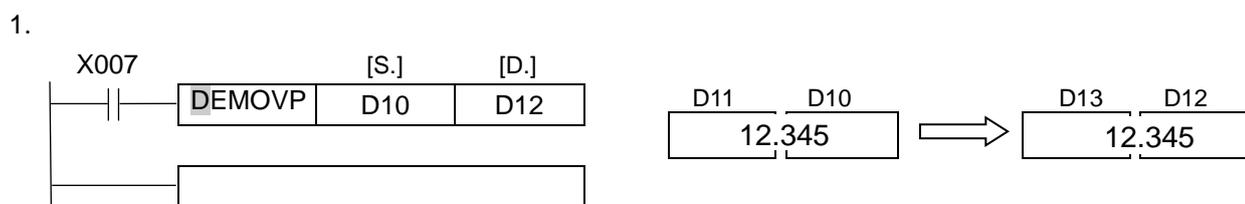


元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]														●									●
[D.]														●									

影響旗號:



### 範例



**FNC(116)ESTR 2 進制浮點數→字串的轉換**

FNC(116)			
D	ESTR	P	32 bits:(D)ESTR & (D)ESTR(P) ----- 13 steps

--	--	--	--

Reserved

**FNC(117)EVAL/字串→2 進制浮點數的轉換**

FNC(117)			
D	EVAL	P	32 bits:(D)EVAL & (D)EVAL(P) ----- 9 steps

--	--	--	--

Reserved

**FNC(118)2 進制浮動小數點 → 10 進制浮動小數點轉換**

FNC(118)			
D	EBCD	P	32 bits:(D)EBCD & (D)EBCD(P) ----- 9 steps

--	--	--	--

Reserved

**FNC(119)10 進制浮動小數點 → 2 進制浮動小數點轉換**

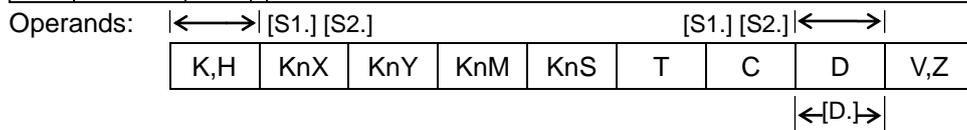
FNC(119)			
D	EBIN	P	32 bits:(D)EBIN & (D)EBIN(P) ----- 9 steps

--	--	--	--

Reserved

## FNC(120)2 進制浮動小數點加算

FNC(120)			32 bits:(D)EADD & (D)EADD(P) ----- 13 steps													J2n--		J3n--	
D	EADD	P																	

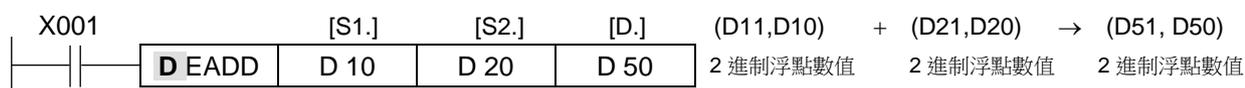


Operands:

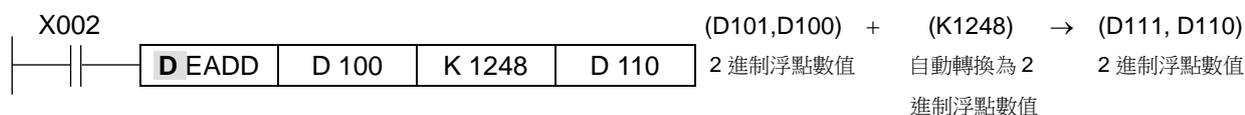
X	Y	M	S
---	---	---	---

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]														●	●					●	●		
[S2.]														●	●					●	●		
[D.]														●	●								

影響旗號: 無



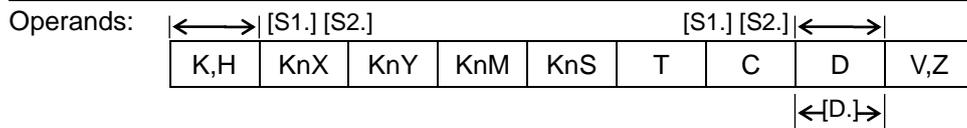
- ◆ 將 2 個儲存 2 進制浮點數值的組件作加算後，以 2 進制浮點數值的形式存入目的地組件中。
- ◆ 來源運算元，若以常數 K 或 H 指定時，則自動地將其轉換為 2 進制浮點數值作處理。



- ◆ 可以指定來源運算元[S.]和目的地運算元[D.]為相同的組件編號。此時，若使用連續執行型命令，則因每個掃描週期均會執行加算。

## FNC(121)2 進制浮動小數點減算

FNC(121)			32 bits:(D)ESUB & (D)ESUB(P) ----- 13 steps													J2n--		J3n--	
D	ESUB	P																	

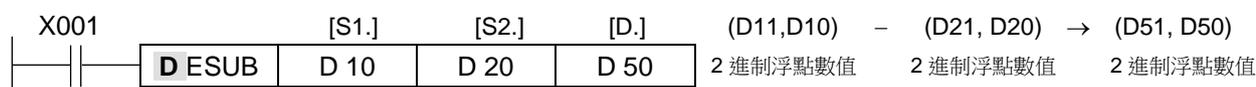


Operands:

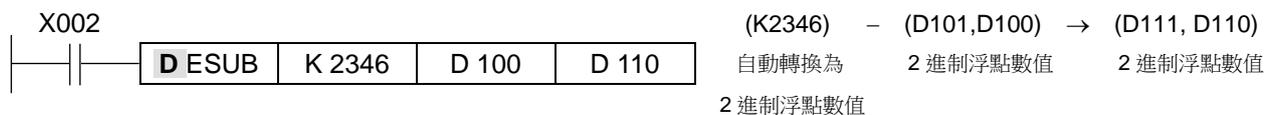
X	Y	M	S
---	---	---	---

元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E	
[S1.]														●	●					●	●	
[S2.]														●	●					●	●	
[D.]														●	●							

影響旗號: 無



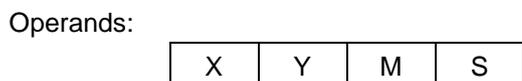
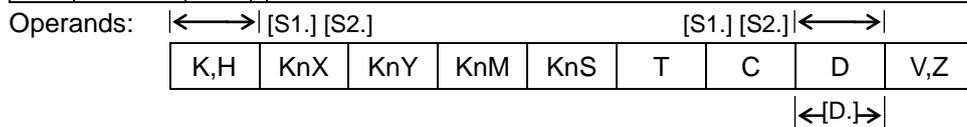
- ◆ [S1.]的 2 進制浮點數值減去[S2.]的 2 進制浮點數值，將其結果以 2 進制浮點數值形式存放在目的地組件[D.]。
- ◆ 來源運算元，若以常數 K 或 H 指定時，則自動地將其轉換為 2 進制浮點數值來作處理。



- ◆ 可以指定來源運算元[S.]和目的地運算元[D.]為相同的組件編號。此時，若使用連續執行型命令，則因每個掃描週期均會執行減算。

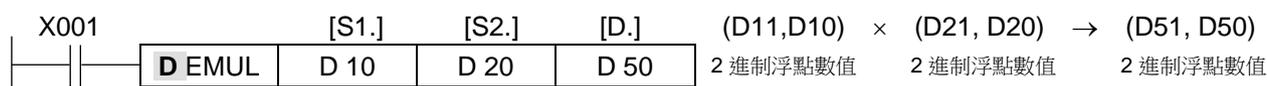
## FNC(122) 2 進制浮動小數點乘算

FNC(122)			32 bits:(D)EMUL & (D)EMUL(P) ----- 13 steps													J2n--		J3n--	
D	EMUL	P																	

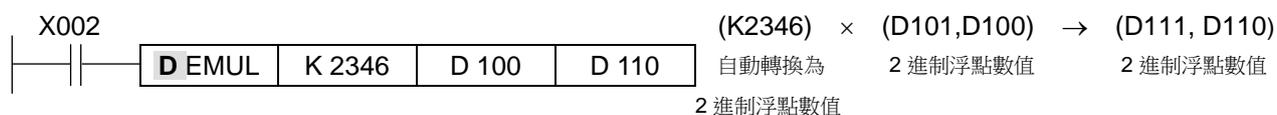


元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[S1.]														●	●					●	●	
[S2.]														●	●					●	●	
[D.]														●	●							

影響旗號: 無

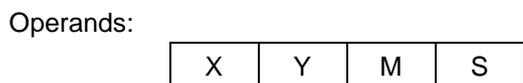
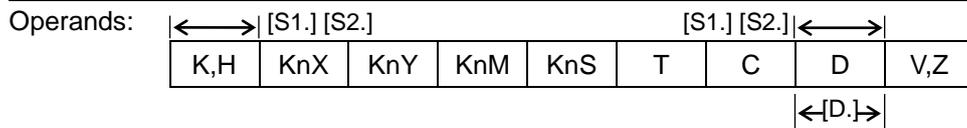


- ◆ 將 2 個來源組件[S1.]和[S2.]的 2 進制浮點數值相乘後，將其結果以 2 進制浮點數值的形式存放在目的地組件[D.]。
- ◆ 若來源運算元以常數 K 或 H 指定時，則自動地將其轉換為 2 進制浮點數值來作處理。



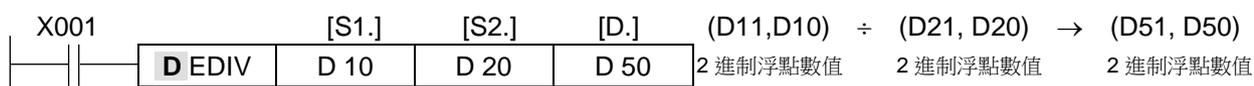
## FNC(123)2 進制浮動小數點除算

FNC(123)			32 bits:(D)EDIV & (D)EDIV(P) ----- 13 steps													J2n--		J3n--	
D	EDIV	P																	

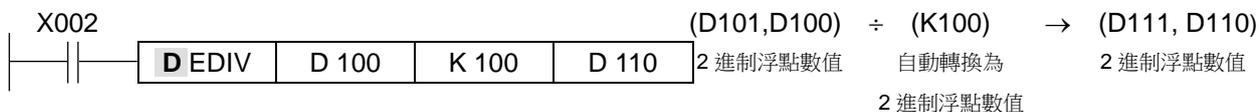


元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P			K	H	E
[S1.]														●	●						●	●	
[S2.]														●	●						●	●	
[D.]														●	●								

影響旗號: 無



- ◆ 將所指定組件[S1.]的 2 進制浮點數值，除以[S2.]指定組件的 2 進制浮點數值，並將其結果以 2 進制浮點數值的形式存放在目的地組件[D.]。
- ◆ 來源運算元，若以常數 K 或 H 指定時，則自動地將其轉換為 2 進制浮點數值來作處理。



**FNC(124)EXP/2 進制浮點數指數運算**

FNC(124)						
D	EXP	P	32 bits:(D)EXP & (D)EXP(P) ----- 9 steps			

Reserved

**FNC(125)LOGE/2 進制浮點數自然對數運算**

FNC(125)						
D	LOGE	P	32 bits:(D)CMP & (D)CMP(P) ----- 9 steps			

Reserved

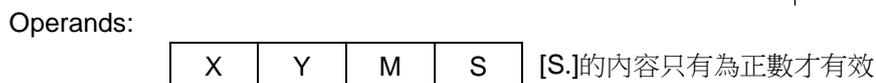
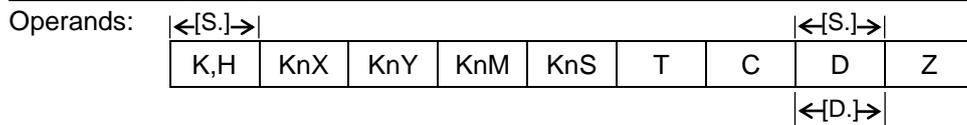
**FNC(126)LOG10/2 進制浮點數常用對數運算**

FNC(126)						
D	LOG10	P	32 bits:(D)LOG10 & (D)LOG10(P) ----- 9 steps			

Reserved

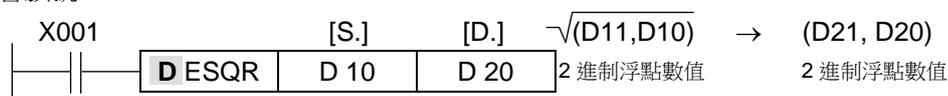
## FNC(127)2 進制浮動小數點開平方根

FNC(127)			32 bits:(D)ESQR & (D)ESQR(P) ----- 13 steps													J2n--		J3n--	
D	ESQR	P																	

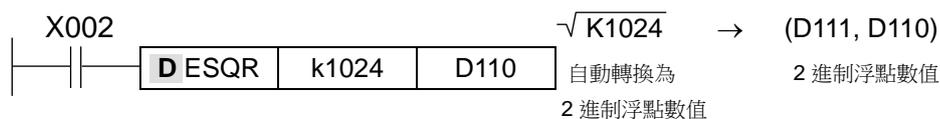


元件種類	位元元件						位元組合元件					字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]														●	●					●	●		
[D.]														●	●								

影響旗號: M8020



- ◆ 將來源[S.]的 2 進制浮點數值的組件做開平方根演算，並將其結果以 2 進制浮點數值儲放在[D.]。
- ◆ 若來源運算元，以常數 K 或 H 指定時，則自動地將其轉換為 2 進制浮點數值來作處理。



- ◆ 演算結果若確實為 0，則零旗標 M8020 會 ON。
- ◆ 來源運算元的內容，僅在為正數時才有效。若為負數時，則演算異常 M8067 會 ON，並停止執行命令。

## FNC(128)ENEG/2 進制浮點數符號翻轉

FNC(128)																		J3n--		
D	ENEG	P	32 bits:(D)ENEG & (D)ENEG(P) ----- 5 steps																	

Operands:

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z	
							←[D.]→		

Operands:

X	Y	M	S
---	---	---	---

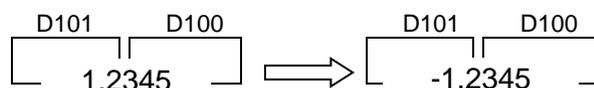
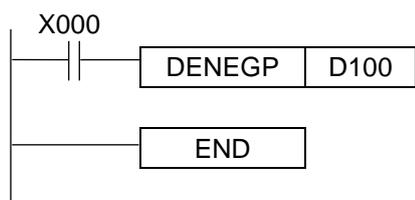
元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[D.]														●	●							

影響旗號: 無



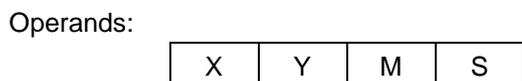
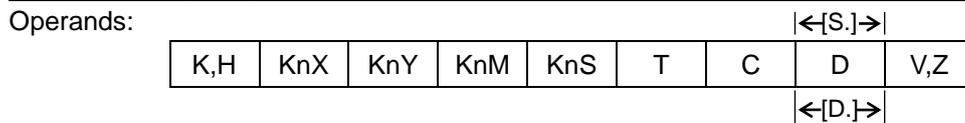
### 範例

X000 為 ON 時，將 D100、D101 的 2 進制浮點數資料的符號翻轉，並且保存到 D100、D101 中的程式。



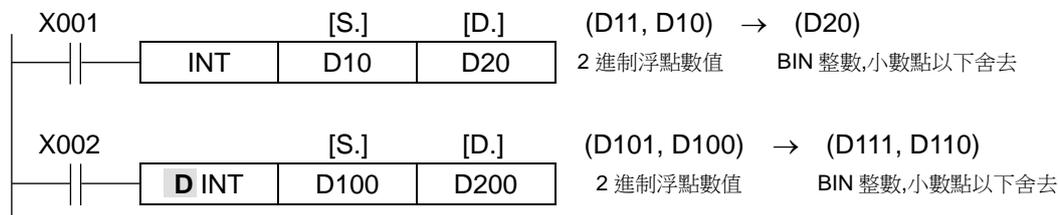
## FNC(129)2 進制浮動小數點→BIN 整數變換

FNC(129)			16 bits:INT & INT ----- 5 steps			J2n--	J3n--
D	INT	P	32 bits:(D)INT & (D)INT(P) -----9 steps				



元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[S.]														●	●							
[D.]														●	●							

影響旗號:



- ◆ 將所指定組件[S.]組件的 2 進制浮點數值轉換成 BIN 整數並存放在目的地元件[D.]。此時小數點以下的數值被舍去。
- ◆ 本命令為 FNC49 (FLT) 命令的相反轉換命令。
- ◆ 演算結果為 0 時，則零旗標 M8020 為 ON。  
轉換時因不足 1 而舍去成為 0 時，則負號旗標 M8021 會 ON。  
演算結果若超出下記範圍會發生溢位，且進位旗標 M8022 會 ON。  
16 位演算時：-32,768~32,767  
32 位演算時：-2,147,483,648~2,147,483,647

## FNC(130)浮動小數點 SIN 演算

FNC(130)																	J2n--		J3n--	
D	SIN	P	32 bits:(D)SIN & (D)SIN(P) ----- 9 steps																	

Operands:

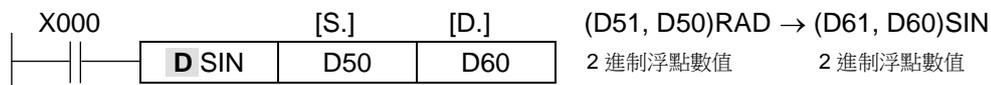
							←[S.]→				
K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z			
							←[D.]→				

Operands:

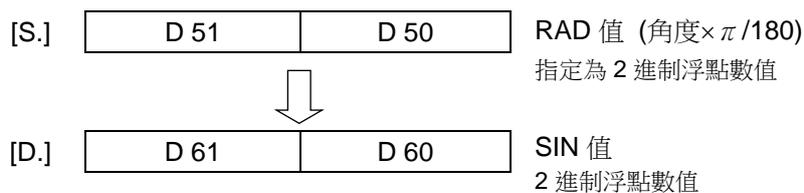
X	Y	M	S	$0^\circ \leq \text{角度} < 360^\circ$
---	---	---	---	--------------------------------------

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S.]														●	●									
[D.]														●	●									

影響旗號:



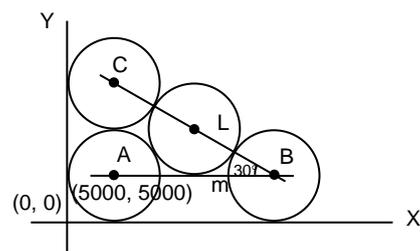
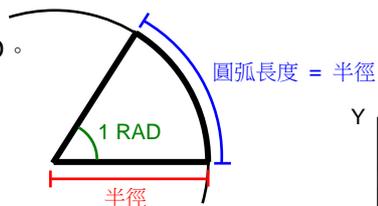
- ◆ 將來源 [S.] 所指定的弧度 (RAD) ，求取其 SIN 值，並將結果存放在目的地組件 [D.]。



◆單位弧度定義: 圓弧長度=半徑時的圓心角。記為 RAD。

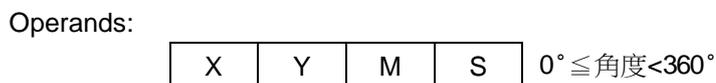
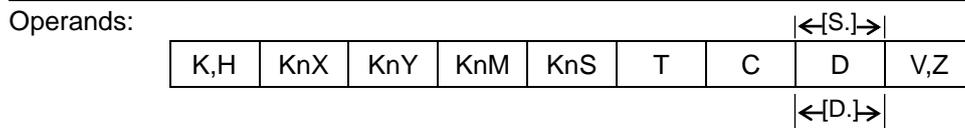
◆ $1 \text{ rad} = 180/\pi^\circ$ ;  $1^\circ = \pi/180 \text{ rad}$

◆求 m 的長度



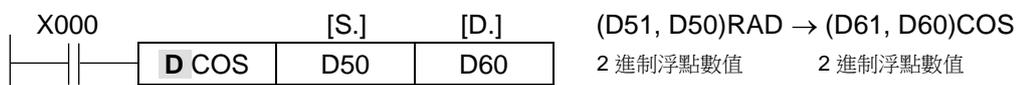
### FNC(131)浮動小數點 COS 演算

FNC(131)			32 bits:(D)COS & (D)COS(P) ----- 9 steps												J2n--		J3n--	
D	COS	P																

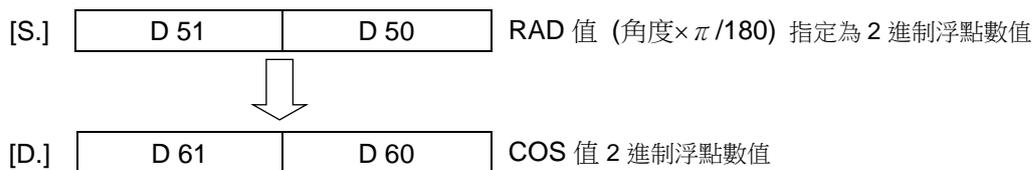


元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E		
[S.]														●	●								
[D.]														●	●								

影響旗號:



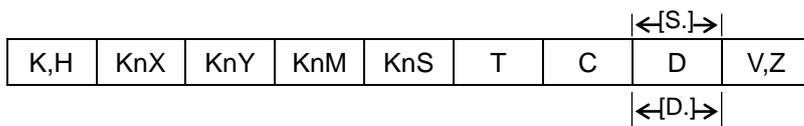
- ◆ 將來源組件[S.]所指定的角度(RAD)，求取其 COS 值，並將結果存放在目的地組件[D.]。



## FNC(132)浮動小數點 TAN 演算

FNC(132)											J2n--	J3n--
D	TAN	P	32 bits:(D)TAN & (D)TAN(P) ----- 9 steps									

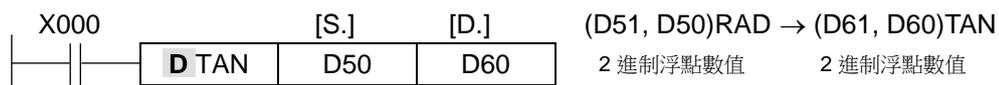
Operands:



Operands:



影響旗號:

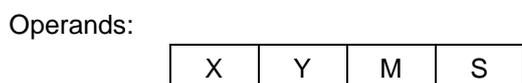
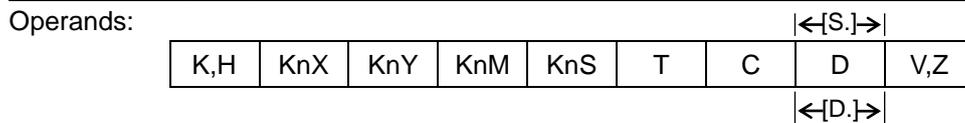


- ◆ 將來源組件[S.]所指定的角度(RAD) ，求取其 TAN 值，並將結果存放在目的地組件[D.]。



## FNC(133)ASIN/2 進制浮點數 SIN-1 運算

FNC(133)															J3n--		
D	ASIN	P	32 bits:(D)ASIN & (D)ASIN(P) ----- 9 steps														

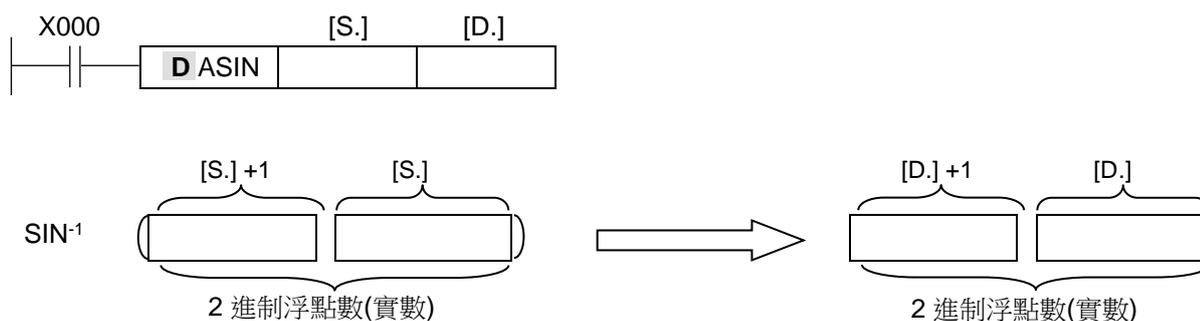


元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E
[S.]														●	●							
[D.]														●	●							

影響旗號:

### 1. 32位運算(DASIN/DASINP)

[S.] +1, [S.]的SIN值求出角度, 將運算結果保存到[D.] +1, [D.]中。此外, 可以在[S.]中直接指定實數。



- ◆ [S.] +1, [S.]的SIN值, 可以在-1.0~1.0的範圍內設定。
  - ◆ [D.] +1, [D.]中保存的角度(運算結果)是保存弧度( $-\pi/2$ )~( $\pi/2$ )的值。
- 關於弧度與角度之間的轉換, 請參考RAD(FNC 136)命令、DEG(FNC 137)指令。

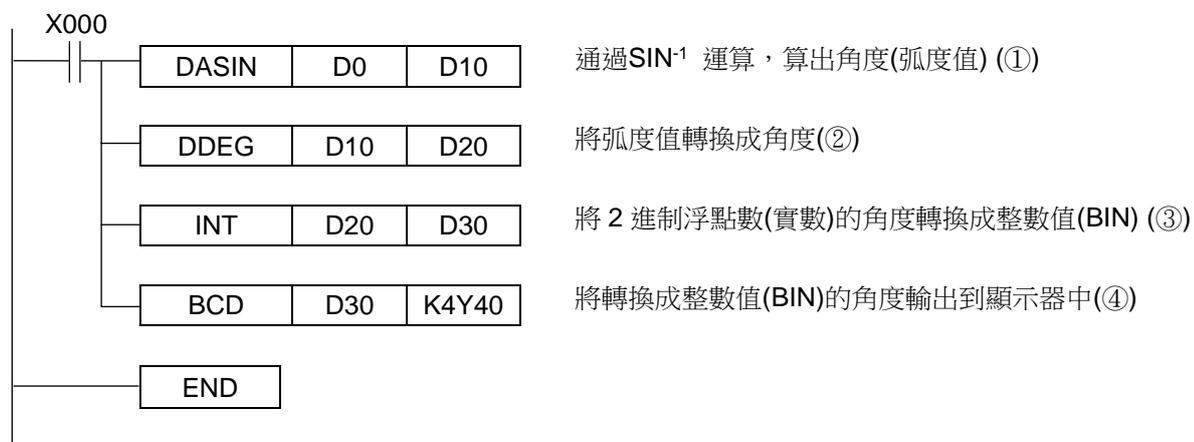
### 錯誤

下面的情況下會發生運算錯誤, 錯誤標誌位元(M8067)為ON, 錯誤代碼保存在D8067中。

- ◆ [S.]中指定的值不在-1.0~1.0的範圍內時。(錯誤代碼: K6706)

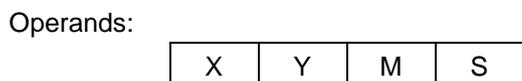
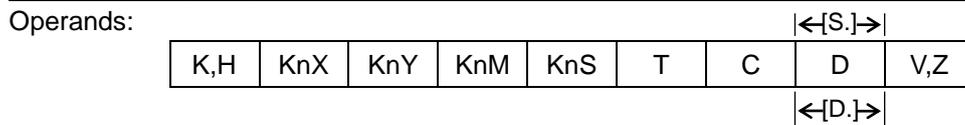
## 範例

X000 為ON時，求出D0、D1(2 進制浮點數) 的 $\text{SIN}^{-1}$ ，然後將其角度以BCD4位數形式輸出到Y040 ~Y057 中的程式。



## FNC(134)ACOS/2 進制浮點數 COS-1 運算

FNC(134)																		J3n--		
D	ACOS	P	32 bits:(D)ACOS & (D)ACOS(P) ----- 9 steps																	

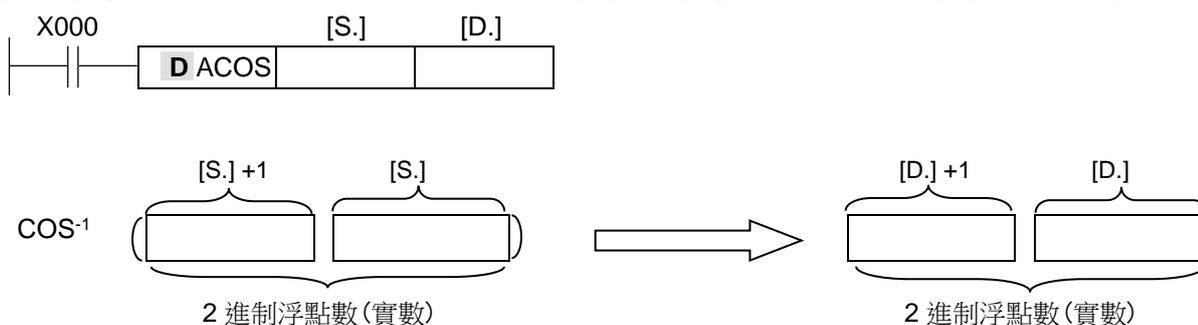


元件種類	位元元件						位元組合元件					字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]														●	●								
[D.]														●	●								

影響旗號:

### 1. 32位運算(DACOS/DACOSP)

[S.] +1, [S.]的COS值求出角度, 將運算結果保存到[D.] +1, [D.]中。此外, 可以在[S.]中直接指定實數。



- ◆ [S.] +1, [S.]的COS值, 可以在-1.0~1.0的範圍內設定。
- ◆ [D.] +1, [D.]中保存的角度(運算結果)是保存弧度0~π的值。  
關於弧度與角度之間的轉換, 請參考 RAD(FNC 136)指令、DEG(FNC 137)指令。

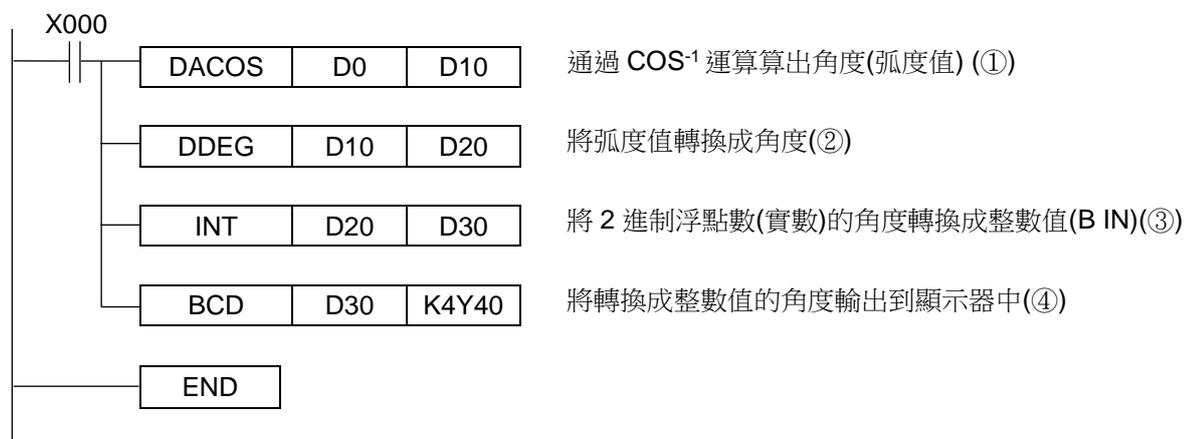
### 錯誤

下面的情況下會發生運算錯誤, 錯誤標誌位元(M8067)為ON, 錯誤代碼保存在D8067中。

- ◆ [S.]中指定的值不在-1.0~1.0的範圍內時。(錯誤代碼: K6706)

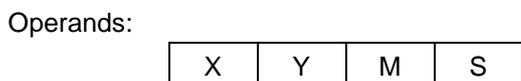
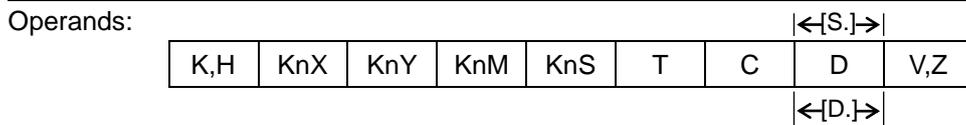
## 範例

X000為ON時，求出D0、D1(2進制浮點數)的 $\text{COS}^{-1}$ ，然後將其角度以BCD4位數形式輸出到Y040~Y057中的程式。



## FNC(135)ATAN/2 進制浮點數 TAN-1 運算

FNC(135)															J3n--		
D	ATAN	P	32 bits:(D)ATAN & (D)ATAN(P) ----- 9 steps														

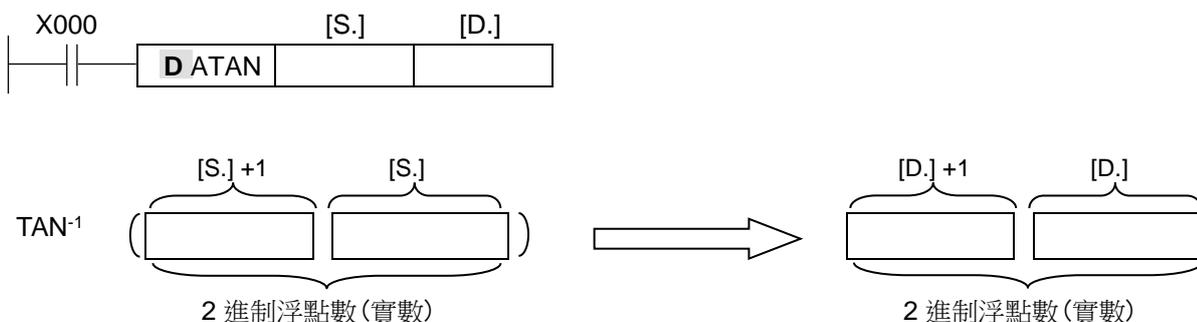


元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]														●	●								
[D.]														●	●								

影響旗號:

### 1. 32位運算(DATAN/DATANP)

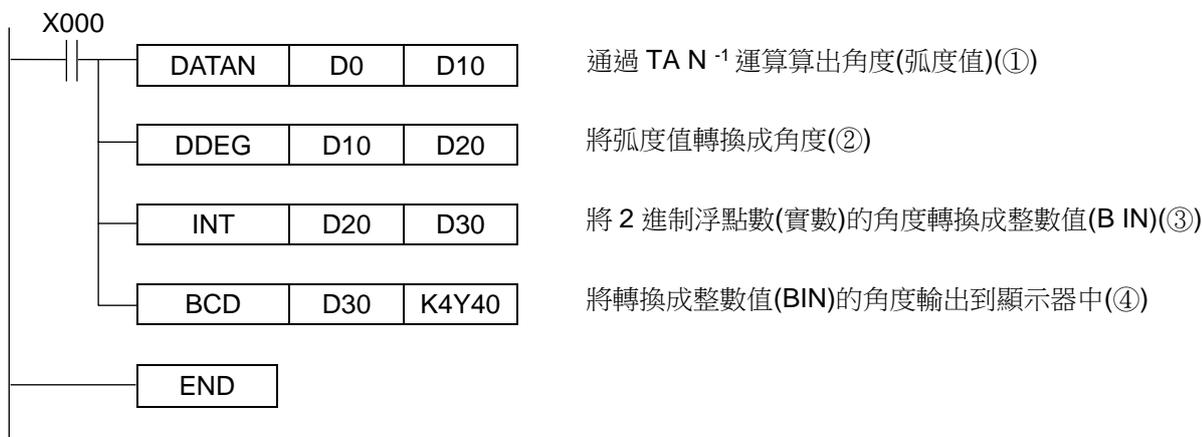
[S.]+1, [S.]的TAN值求出角度，將運算結果保存到[D.]+1, [D.]中。此外，可以在[S.]中直接指定實數。



◆ [D.]+1, [D.]中的角度(運算結果)，保存以弧度為單位的比(- $\pi/2$ )大、比( $\pi/2$ )小的數值。  
關於弧度與角度之間的轉換，請參考 RAD(FNC 136)指令、DEG(FNC 137)指令。

### 範例

X000為ON時，求出D0、D1(2 進制浮點數)的TAN<sup>-1</sup>，然後將其角度以BCD4位數形式輸出到Y040~Y057中的程式。

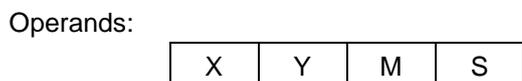
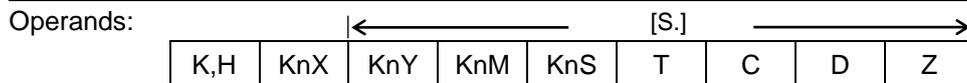






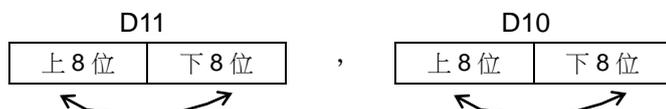
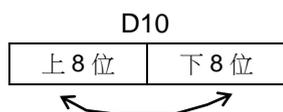
## FNC(147)上下位元組(Byte)交換

FNC(147)			16 bits:SWAP & SWAP(P) ----- 5 steps			J2n--	J3n--
D	SWAP	P	32 bits:(D)SWAP & (D)SWAP(P) -----9 steps				



元件 種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]									●	●	●	●	●	●	●		●						

影響旗號:



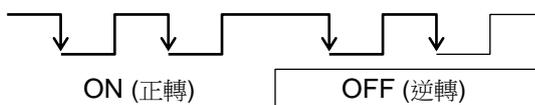
- ◆ 若使用連續執行命令，因每個掃描週期均會執行交換處理，請特別注意。
- ◆ 本命令和 FNC17 (XCH)命令的擴張機能動作相同。

## FNC150 – 159 定位控制概述

◆ 此系列控制器脈波輸出信號是以“脈波列(負邏輯)+符號”的形態，如下圖

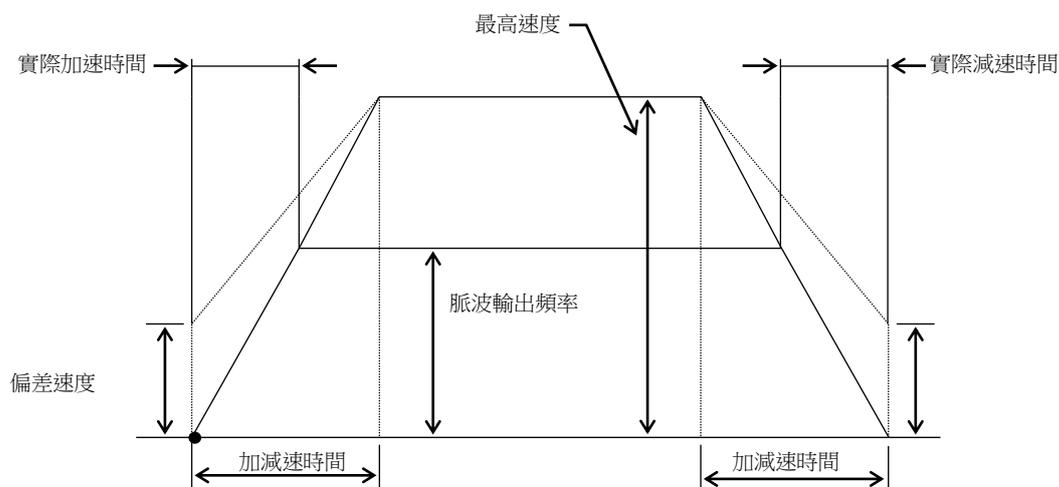
固定以 Y00, Y01 為脈波輸出點

固定 Y02, Y03 為方向輸出點



◆ 脈波導通週期(duty cycle) 50% ON 50% OFF。

◆ 1 段位置驅動曲線情形(定斜率模式)及相關元件



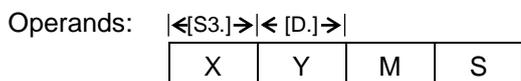
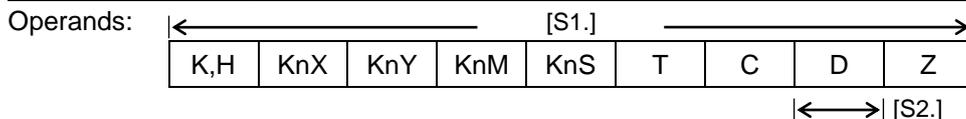
## FNC(155)ABS 現在值讀出

FNC(155)		16 bits:ABS ----- 7 steps				
D	ABS	32 bits:(D)ABS ----- 11 steps				

Reserved

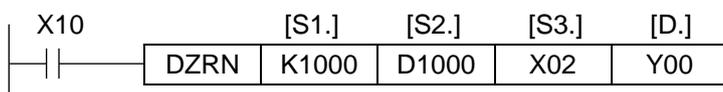
## FNC(156)原點複歸 Zero Return

FNC(156)				J2n--	J3n--
D	ZRN	32 bits:(D)ZRN ----- 17 steps			



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●	●		●			●	●		
[S2.]														●									
[S3.]	●																						
[D.]		●																					

影響旗號: M8029



◆ [D.] 指定脈波輸出點。

[S1.] 指定原點複歸找尋近點速度(Home Speed) 10 ~ 200,000 pps。

[S2.] 從指定的[S2.]開始會連續佔用 100 個 words。在本例中，佔用 D1000~D1099

[S2.]+0：尋找零點速度 10~32,767 pps

[S2.]+1：運轉方向控制點 Y2~Y7

b7	b6	b5	b4	b3	b2	b1	b0
----	----	----	----	----	----	----	----

指定運轉方向輸出點(Y)  
系統保留

[S2.]+2：參數設定

b7	b6	b5	b4	b3	b2	b1	b0
----	----	----	----	----	----	----	----

系統保留  
不具斜率停止旗號(X10 OFF 停止有效)  
系統保留  
系統保留  
變換運轉控制點狀態 0:ON 上數 OFF 下數; 1:ON 下數 OFF 上數  
系統保留

[S2.]+3 ~ [S2.]+99：與 FNC(59) PLSR 的[S3.]+3 ~ [S3.]+99 相同

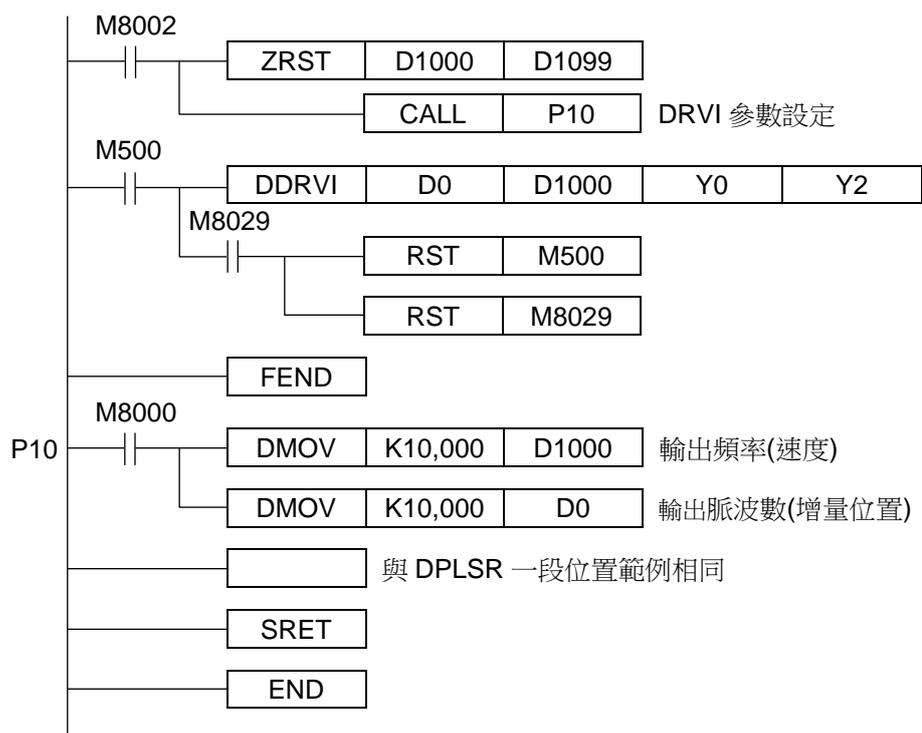
[S3.] 指定近點輸入信號，有效範圍 X00~X07 (脈波截取信號 M8170~M8177)。

零點信號由[S2.]+24 設定。

- ◆ 當執行 ZRN 指令時，歸零點忙碌旗標 M8154~M8157 將被自動設定，避免同時驅動 DRVI,DRVA。
- ◆ 這個指令 Y00 到 Y03 只能使用一次而且必須選擇電晶體輸出模組。
- ◆ 固定為 32 位運算。若指定 16 位元運轉模式，則產生 error 6509。

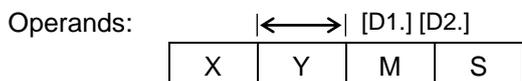
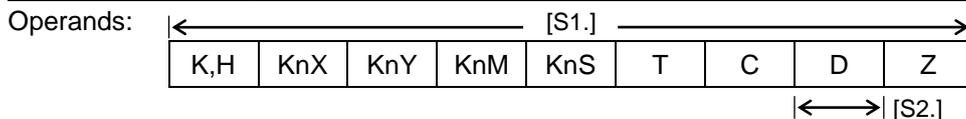






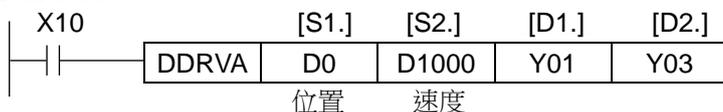
## FNC(159)絕對位址定位控制

FNC(159)				J2n--	J3n--
D	DRVA	32 bits:(D)DRVA ----- 17 steps			

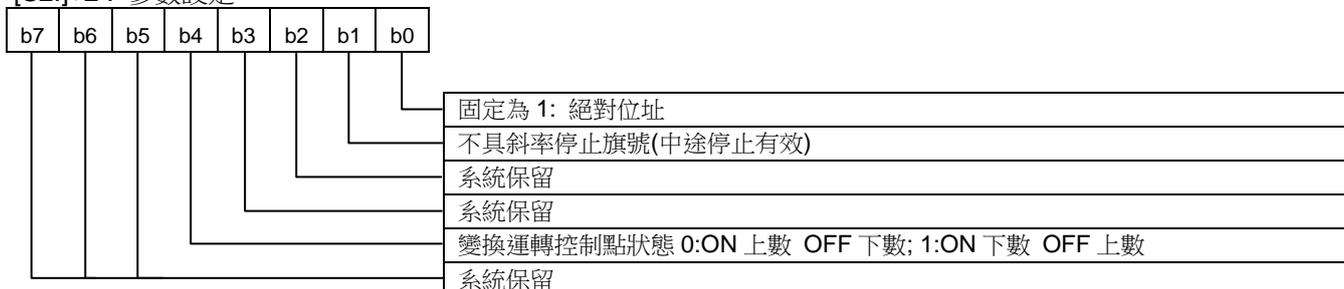


元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●			●			●	●		
[S2.]														●									
[D1.]	●																						
[D2.]	●																						

影響旗號: M8029

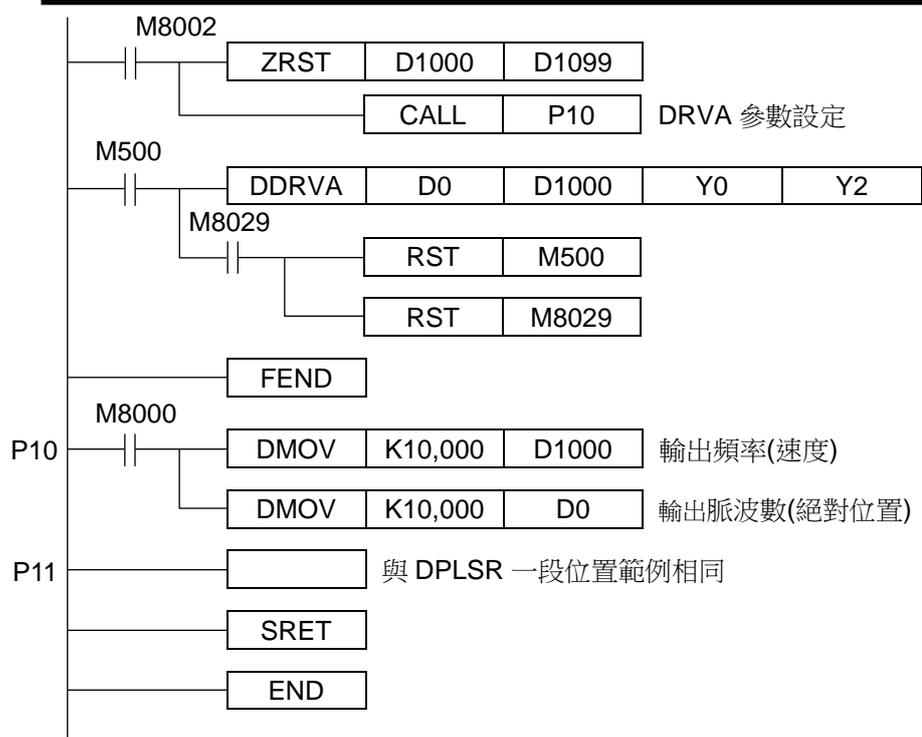


- ◆ [D1.] 指定運轉脈波輸出點。(固定以 Y00~Y03 為輸出點)。
- ◆ [D2.] 指定運轉方向輸出點。(固定以 Y02~Y07 為輸出點)。
- ◆ [S1.] 指定絕對位置輸出脈波數。(與啟始位址比較，決定正、逆轉)
- ◆ [S2.] 從指定的[S2.]開始會佔用連續 100 個 words。在本例中，佔用 D1000~D1099。
- ◆ [S2.]+1, [S2.]+0：指定輸出頻率。[32bits]:10 ~ 200,000 Hz。
- ◆ [S2.]+2：參數設定



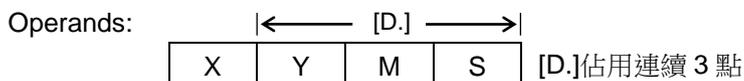
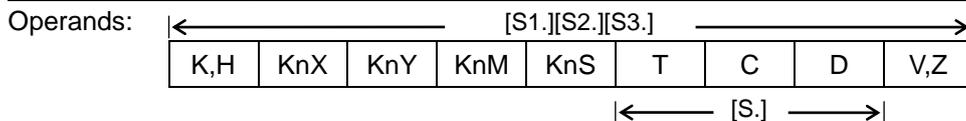
[S2.]+3 ~ [S2.]+99：與 FNC(59) PLSR 的[S3.]+3 ~ [S3.]+99 相同

- ◆ 這個指令 Y00 到 Y03 只能使用一次，而且必須選擇電晶體輸出模組。
- ◆ 當執行 DDRVA 指令時，則忙碌旗標 M8150~M8153 將會被自動設定。
- ◆ 輸出脈波中，修改 [S1], [S2.]+1, [S2.]+0 的內容值無效。
- ◆ 固定為 32 位運算。若指定 16 位元運轉模式，則產生 error 6509。



## FNC(160)時鐘資料比較

FNC(160)		16 bits:TCMP & TCMP(P) ----- 5 steps									J2n--	J3n--
TCMP	P											



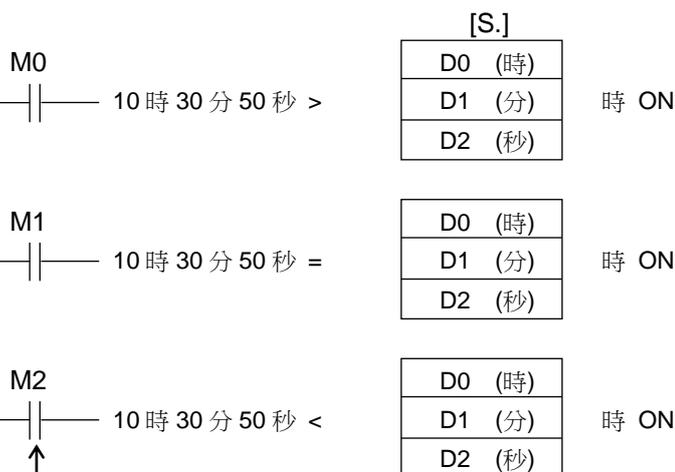
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●		●	●				●	●	
[S2.]								●	●	●	●	●	●	●		●	●				●	●	
[S3.]								●	●	●	●	●	●	●		●	●				●	●	
[S.]												●	●	●									
[D.]	●	●	●																				

[D.]佔用連續 3 點

影響旗號: M8020, M8021, M8022



執行指定時間和時鐘資料做大小比較的指令。



若將 X000 OFF 時，則不執行 TCMP 命令，但 M0~M2 仍保持 OFF 前的狀態。

◆ 來源組件「[S1.],[S2.],[S3.]」的時間和儲存在[S.]開始連續 3 點的時間資料做比較，依比較結果，[D.]開始連續 3 點的組件，自動地 ON/OFF。

[S1.] : 比較時間的“時”指定為「0~23」時。

[S2.] : 比較時間的“分”指定為「0~59」分。

[S3.] : 比較時間的“秒”指定為「0~59」秒。

[S.] : 比較時間的“時”指定為「0~23」時。

[S.] + 1 : 比較時間的“分”指定為「0~59」分。

[S.] + 2 : 比較時間的“秒”指定為「0~59」秒。

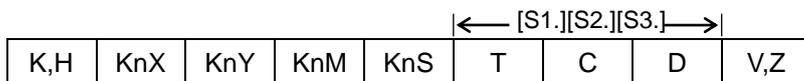
[D.], [D.] + 1, [D.] + 2 : 依比較結果 [D.] 開始連續 3 點的組件自動 ON/OFF。

◆ 時鐘資料可以利用控制器的內部時鐘(Real Time Clock)資料，時鐘資料儲存在下記特殊暫存器 D8015(時)，D8014(分)，D8013(秒)。

## FNC(161)時鐘資料比較

FNC(161)		16 bits:TZCP & TZCP(P) ----- 9 steps									J2n--	J3n--
TZCP	P											

Operands:



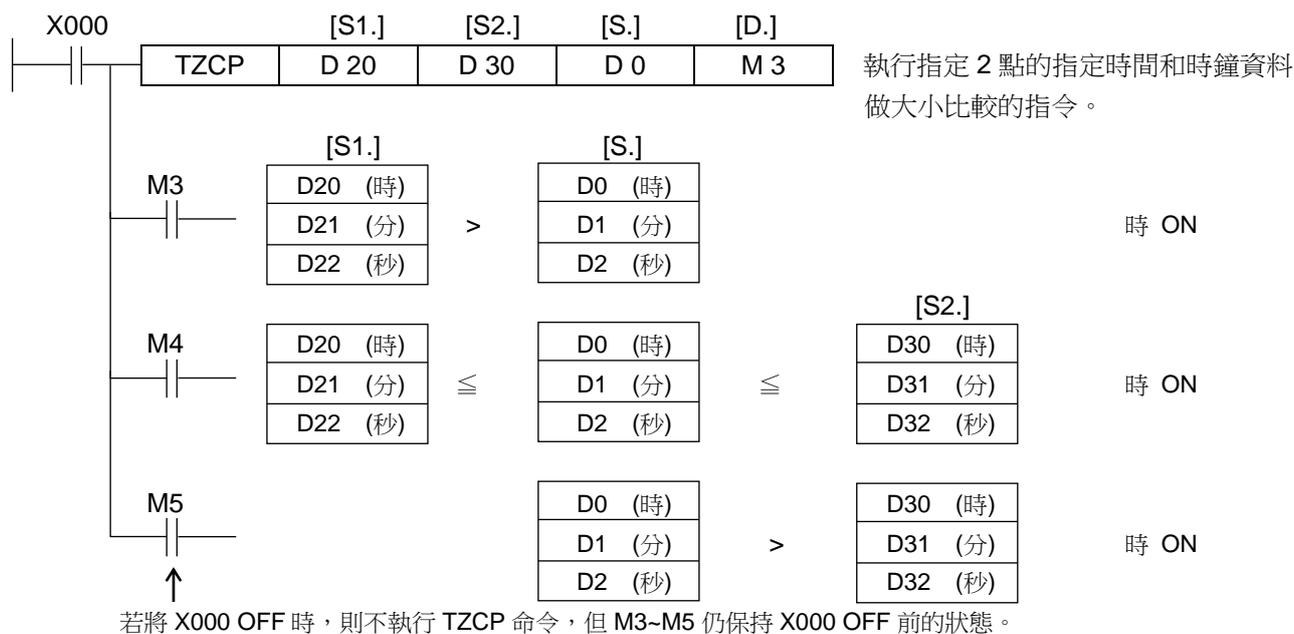
Operands:



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S1.]												●	●	●										
[S2.]												●	●	●										
[S3.]												●	●	●										
[S.]												●	●	●										
[D.]	●	●	●																					

佔用[D.]開始連續 3 點，請設定為[S1.]≤[S2.]

影響旗號: M8020, M8021, M8022



- ◆ 上下點的比較範圍和[S.]開始連續 3 點的時間資料區域做比較，對應比較大小的結果，[D.]開始連續 3 點自動地 ON/OFF。

[S1.], [S.] +1, [S.] +2 : 比較範圍的下限，指定“時”，“分”，“秒”。

[S2.], [S2.] +1, [S2.] +2 : 比較範圍的上限，指定“時”，“分”，“秒”。

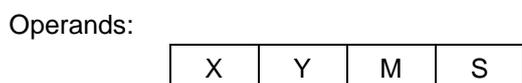
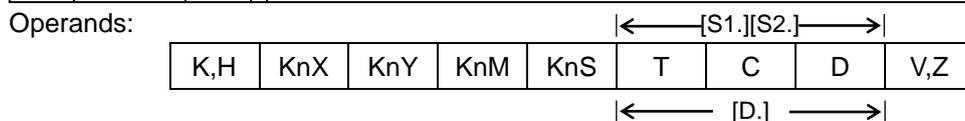
[S.], [S.] +1, [S.] +2 : 時鐘資料，指定“時”，“分”，“秒”。

[D.], [D.] +1, [D.] +2 : 依區域比較結果，[D.]開始連續 3 點的位元組件自動 ON/OFF。

“時”，“分”，“秒”的設定範圍和控制器的內部時鐘的處理，請參考 FNC160 (TCMP)命令。

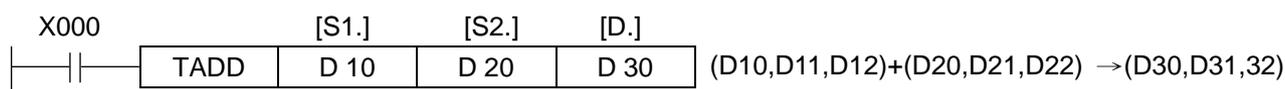
## FNC(162)時鐘資料的加算

FNC(162)		16 bits: TADD & TADD(P) ----- 7 steps												J2n--	J3n--				
TADD	P																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]												●	●	●									
[S2.]												●	●	●									
[D.]												●	●	●									

影響旗號: M8020, M8021, M8022



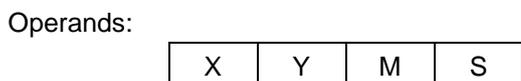
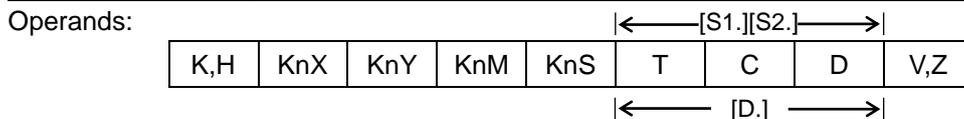
- ◆ 將儲存在[S1.]開始連續 3 點的時間資料加上[S2.]開始連續 3 點的時間資料，並將加算結果存放在[D.]開始連續 3 點的組件。
- ◆ 演算結果若超過(24)時，則進位旗標 M8022 ON。並將加總數值減去 24 後的數值存放在[D.]。



- ◆ 演算結果為 0 (0 時 0 分 0 秒)，則零旗標 M8020 會 ON。
- ◆ “時”，“分”，“秒”的設定範圍和控制器的內部時鐘的處理，請參考 FNC160 (TCMP)命令。

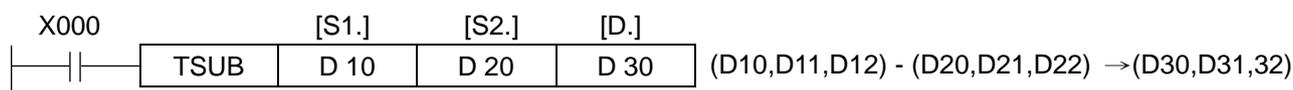
## FNC(163)時鐘資料的減算

FNC(163)			16 bits: TSUB & TSUB(P) ----- 7 steps												J2n--		J3n--		
TSUB	P																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S1.]												●	●	●										
[S2.]												●	●	●										
[D.]												●	●	●										

影響旗號: M8020, M8021, M8022



◆ 將儲存在[S1.]開始連續 3 點的時間資料，減去[S2.]開始連續 3 點的時間資料，並將結果存放在[D.]開始連續 3 點的組件。

◆ 演算結果為 0 時以下負位旗標 ON，並將減算結果加上 24 後的數值存入[D.]中。



◆ 演算結果為 0 (0 時 0 分 0 秒)，則零旗標 M8020 會 ON。

◆ “時”，“分”，“秒”的設定範圍和控制器的內部時鐘的處理，請參考 FNC160(TCMP)命令。

## FNC(166)時鐘資料的讀出

FNC(166)			16 bits: TRD & TRD(P) ----- 5 steps															J2n--	J3n--
	TRD	P																	

Operands:

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z
								← [D.] →

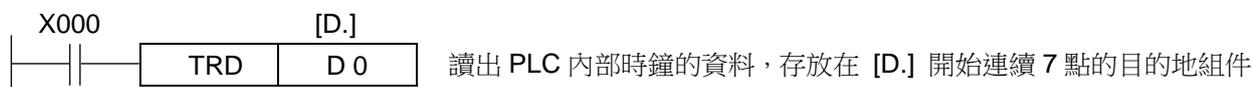
Operands:

X	Y	M	S	佔用[D.]開始連續 7 點
---	---	---	---	----------------

元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[D.]												●	●	●									

佔用[D.]開始連續 7 點

影響旗號:

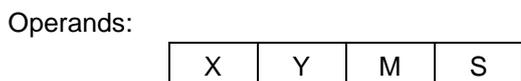


- ◆ 控制器內部時鐘的時間資料，以如下格式讀出。  
讀出的時間資料系儲存在特殊暫存器 D8013~D8019。

特 內 部 暫 時 存 鐘 器 用	元件	項目	時 鐘 資 料	→	元件	項目
	D8018	年(西曆)	0~99(西曆下 2 位)	→	D0	年(西曆)
	D8017	月	1~12	→	D1	月
	D8016	日	1~31	→	D2	日
	D8015	時	0~23	→	D3	時
	D8014	分	0~59	→	D4	分
	D8013	秒	0~59	→	D5	秒
	D8019	星期	0(日)~6(六)	→	D6	星期

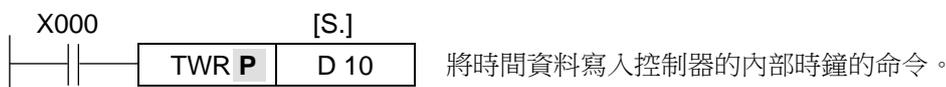
## FNC(167)時鐘資料的寫入

FNC(167)			16 bits: TWR & TWR(P) ----- 5 steps												J2n--	J3n--
TWR	P															



元件 種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]												●	●	●									

影響旗號:



- ◆ 將時間資料寫入控制器的內部時鐘。  
將欲寫入的時間資料先存放在所指定 [S.] 的連續 7 點組件中。

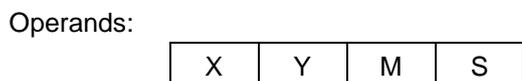
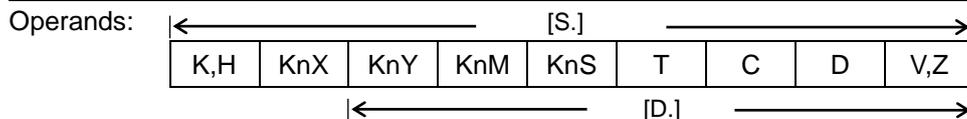
時間 設定 用 資料	元件	項目	時 鐘 資 料	→	元件	項目	特 內 殊 部 資 時 料 鐘 暫 用 存 器
	D10	年(西曆)	0~99(西曆下 2 位)		D8018	年(西曆)	
	D11	月	1~12		D8017	月	
	D12	日	1~31		D8016	日	
	D13	時	0~23		D8015	時	
	D14	分	0~59		D8014	分	
	D15	秒	0~59		D8013	秒	
	D16	星期	0(日)~6(六)		D8019	星期	

執行 FNC167 (TWR)命令後，內部時鐘的時間資料，立即變更為所寫入的新設時間值。因此，請先將數分鐘後數值寫入[S.]組件，待實際時間到達時再執行 TWR 命令，而且利用本命令來校正時間，則不需再使用特殊繼電器 M8015(時間停止和時間校正)。



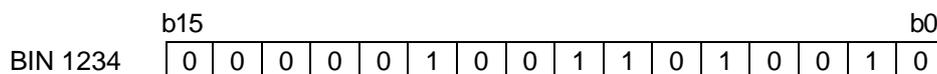
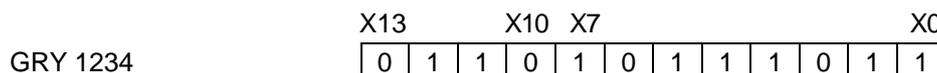
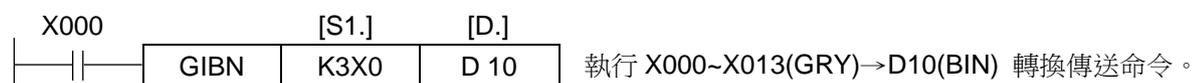
## FNC(171) 格雷碼(GRAY CODE)相反轉換

FNC(171)			16 bits:GBIN & GBIN(P) ----- 5 steps															J2n--		J3n--	
D	GBIN	P	32 bits:(D)GBIN & (D)GBIN(P) -----9 steps																		



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S.]								●	●	●	●	●	●	●		●	●			●	●		
[D.]									●	●	●	●	●	●		●	●						

影響旗號:



- ◆ 將 GRAY CODE 轉換成 BIN 數值的轉換傳送命令，可利用 GRAY CODE 方式的編碼器，作絕對位置的檢出。
- ◆ 指定 [S.] 為輸入(X)時，會有「掃瞄時間+輸入濾波器常數」的反應延遲時間。
- ◆ 使用 FNC51 (REFF)命令或改變輸入濾波器常數值 D8020 (濾波器調整)，則可消除濾波器常數部分的反應延遲時間。
- ◆ 使用 **D** GBIN 命令時，可執行最多 32 位的 GRAY CODE 相反轉換。
- ◆ 對於 [S.] 有效的數值範圍如下，
  - 16 位演算時：0~32,767
  - 32 位演算時：0~2,147,483,647

**FNC(192)BK+/資料塊的加法運算**

FNC(192)			16 bits: BK+ & BK+P(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BK+	P	32 bits:(D)BK+ & (D)BK+(P) ----- 17 steps				

Reserved

**FNC(193)BK-/資料塊的減法運算**

FNC(193)			16 bits: BK- & BK-(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BK-	P	32 bits:(D)BK- (D)BK-(P) -----17 steps				

Reserved

**FNC(194)BKCMP=, >, <, < >, <=, >= / 資料塊比較**

FNC(194)			16 bits: BKCMP= & BKCMP=P(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BKCMP=	P	32 bits:(D)BKCMP= (D)BKCMP=P(P) -- - - - - 17				

Reserved

FNC(195)			16 bits: BKCMP> & BKCMP>P(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BKCMP>	P	32 bits:(D)BKCMP> (D)BKCMP>P(P) --- - - - - 17				

Reserved

FNC(196)			16 bits: BKCMP< & BKCMP<(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BKCMP<	P	32 bits:(D)BKCMP< (D)BKCMP<(P) -- - - - - 17				

Reserved

FNC(197)			16 bits: BKCMP<> & BKCMP<>P(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BKCMP<	P	32 bits:(D)BKCMP<> (D)BKCMP<>(P) -- - - - - 17				

Reserved

FNC(198)			16 bits: BKCMP<= & BKCMP<=P(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BKCMP<	P	32 bits:(D)BKCMP<= (D)BKCMP<=P(P) -- - - - - 17				

Reserved

FNC(199)			16 bits: BKCMP>= & BKCMP>=P(P) ----- 9 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
D	BKCMP>	P	32 bits:(D)BKCMP>= (D)BKCMP>=P(P) -- - - - - 17				

Reserved

**FNC(210)FDEL /資料表的資料刪除**

FNC(210)			16 bits: FDEL & FDEL(P) ----- 7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	FDEL	P					

Reserved

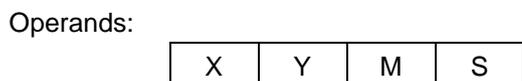
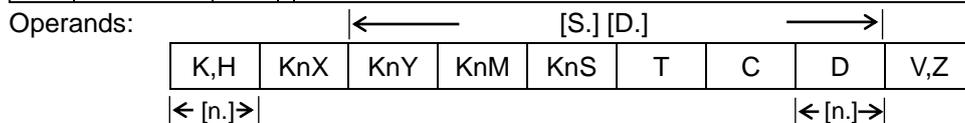
**FNC(211)FINS / 資料表的資料插入**

FNC(211)			16 bits: FINS & FINS(P) ----- 7 steps	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	FINS	P					

Reserved

## FNC(212)POP/讀取後入的資料[先入後出控制用]

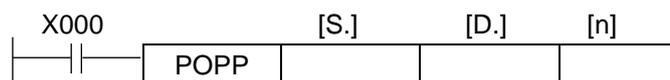
FNC(212)		16 bits: POP & POP(P) ----- 7 steps												J3n--	
POP	P														



元件種類	位元元件				位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E
[S.]									●	●	●	●	●	●							
[D.]									●	●	●	●	●	●							
[n.]														●					●	●	

影響旗號:

### 1. 16 位運算(POP/POPP)

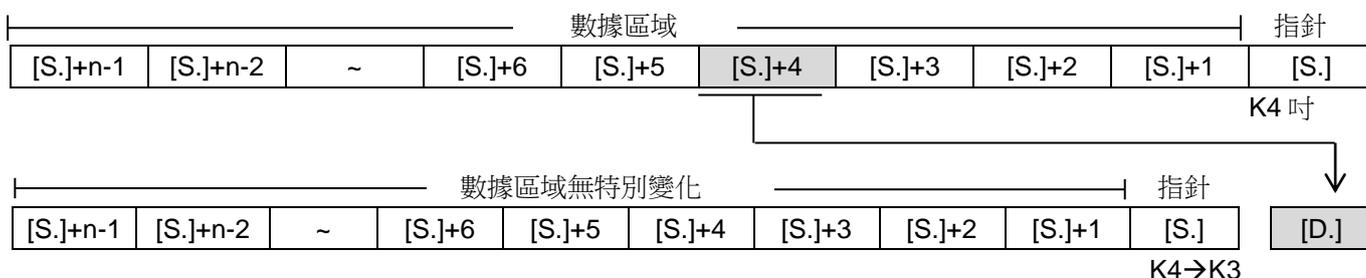


先入後出控制用資料

	內容
[S.]	指標資料(被保存的資料個數)
[S.]+1	資料區域 (使用移位元元寫入指令(SFWR)被先入的資料)
[S.]+2	
[S.]+3	
...	
[S.]+n-3	
[S.]+n-2	
[S.]+n-1	

◆ 對於[S.]~[S.] +n-1的字軟元件，每次執行指令時，讀出([S.] +指標資料) 的軟元件保存到[D.]中( 使用先入先出控制用的移位元元寫入指令(SFWR)寫入的最後的資料讀出到[D.]中)。n可以指定2~512。

◆ 指標數據[D.]的值減 1

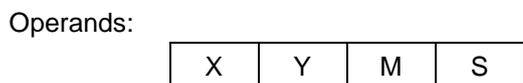
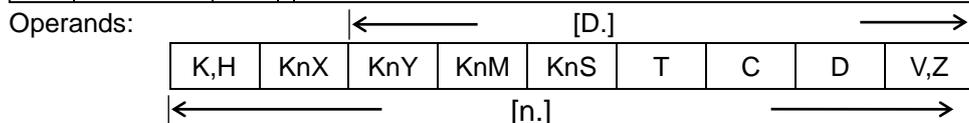
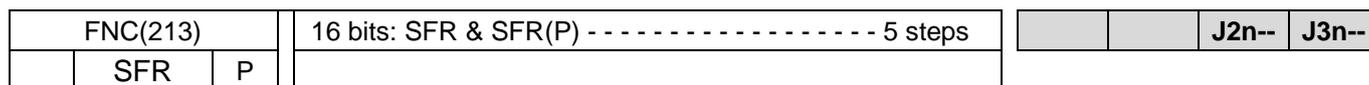


### 注意要點

◆ 用連續執行型編寫該指令時，每個運算週期都會處理指令，所以請注意有時可能會出現意外的動作。通常，編程時使用[脈衝執行型]或是通過[脈衝化的指令觸點]執行。

- 
- ◆ 指針的當前值為0時，零位元標誌位元M8020為ON，不處理指令。此時，請先使用比較指令確認的當前值是否 $1 \leq \leq (n-1)$ ，然後執行這個指令。
  - ◆ 指針的當前值為1時，中被寫入0，零位元標誌位元M8020為ON。錯誤
  - ◆ 以下一些情況下會發生運算錯誤，錯誤標誌位元M8067置ON，錯誤代碼保存在D8067中。
    - $> n-1$ 時(錯誤代碼: K6706) —  $< 0$ 時(錯誤代碼: K6706)
  - ◆ [D.]的堆疊次數不能為負 否則出現 **Error**
  - ◆ [D.]的堆疊次數不能大於 n 否則出現 **Error**

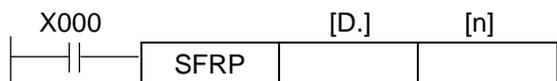
### FNC(213)SFR/16 位元數據 n 位元右移(帶進位)



元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E		
[D.]									●	●	●	●	●	●		●	●						
[n.]								●	●	●	●	●	●	●		●	●		●	●			

影響旗號:

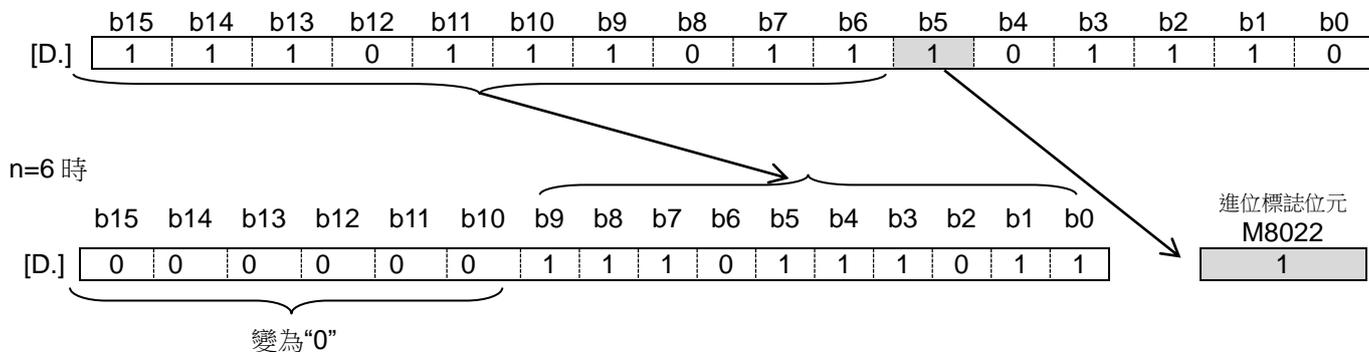
#### 1. 16 位運算(SFR/SFRP)



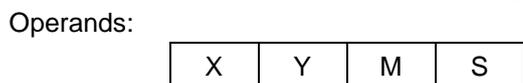
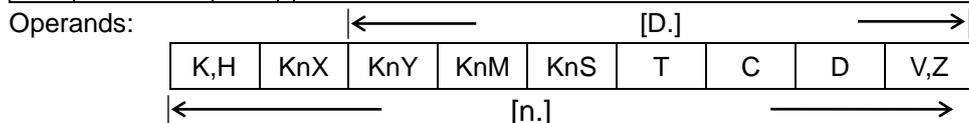
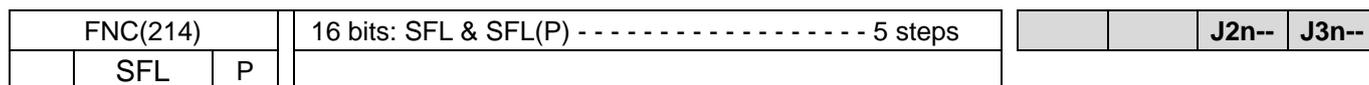
(1) 字軟元件中[D.]的16位右移n位。0n指定00~15的數字。當n中指定了16以上的數值時，根據n/16的餘數移動。例如，如n=18時，18/16=1餘2，所以右移2位。

(2) 將字軟元件中[D.]的第n位(n-1位)的ON(1)/OFF(0)狀態轉移到進位元標誌位元M8022中。

(3) 最高位開始的 n 位變為 0。



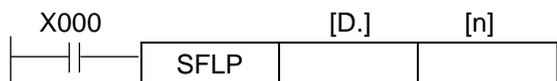
### FNC(214)SFL/16 位元數據 n 位元左移(帶進位)



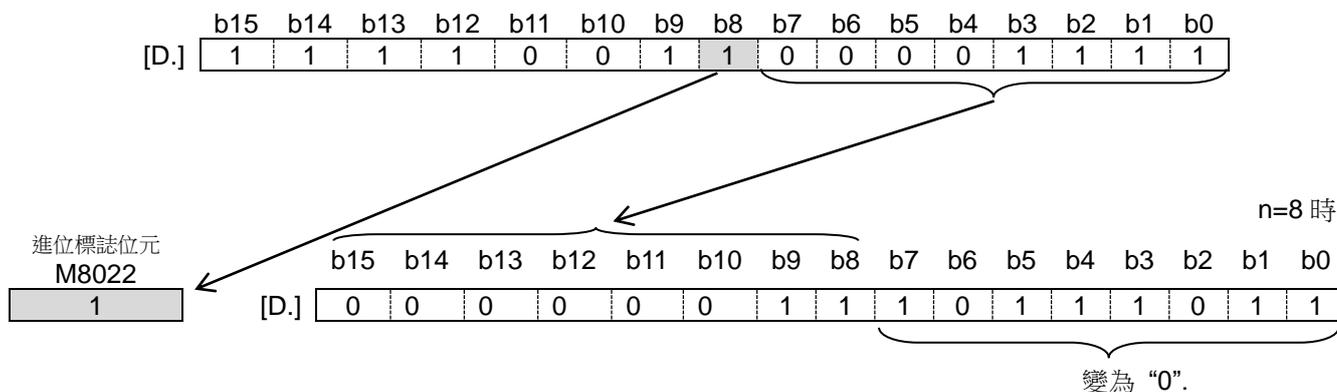
元件種類	位元元件						位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P	K	H	E	
[D.]								●	●	●	●	●	●	●		●	●					
[n.]							●	●	●	●	●	●	●	●		●	●		●	●		

影響旗號:

#### 16 位運算(SFL/SFLP)



- (1) 字軟元件[D.]中的16位左移n位。n指定0~15的數字。當n中指定了16以上的數值時，根據n/16的餘數移動。例如，如n=18時，18/16=1餘2，所以左移2位。
- (2) 將字軟元件[D.]中的第n+1位(n位)的ON(1)/OFF(0)狀態轉移到進位元標誌位元M8022中。
- (3) 最低位開始的 n 位變為 0。



## FNC(224)接點型比較命令演算開始 LD ※

FNC(224~230)		16 bits: ----- 5 steps			J2n--	J3n--
D	LD ※	32 bits: ----- 9 steps				

※ : =, >, <, <>, ≤, ≥

Operands:  $\leftarrow$  [S1.][S2.]  $\rightarrow$

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z
-----	-----	-----	-----	-----	---	---	---	-----

Operands:

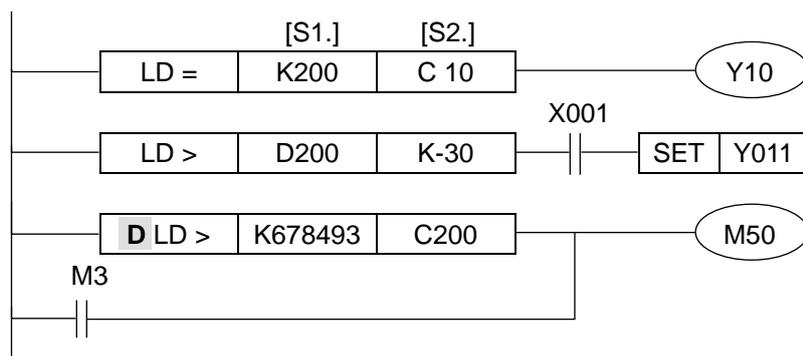
X	Y	M	S
---	---	---	---

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數	
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E		
[S1.]								●	●	●	●	●	●	●		●	●					●	●	
[S2.]								●	●	●	●	●	●	●		●	●					●	●	

影響旗號:

◆ 2 個來源運算元的內容做 BIN 比較，對應比較的結果，執行後段的順序程式。LD ※ 為連接母線的接點型比較命令。

FNC No.	16 位命令	32 位命令	導通條件	非導通條件
224	LD =	<b>D</b> LD =	[S1.] = [S2.]	[S1.] ≠ [S2.]
225	LD >	<b>D</b> LD >	[S1.] > [S2.]	[S1.] ≤ [S2.]
226	LD <	<b>D</b> LD <	[S1.] < [S2.]	[S1.] ≥ [S2.]
228	LD <>	<b>D</b> LD <>	[S1.] ≠ [S2.]	[S1.] = [S2.]
229	LD ≤	<b>D</b> LD ≤	[S1.] ≤ [S2.]	[S1.] > [S2.]
230	LD ≥	<b>D</b> LD ≥	[S1.] ≥ [S2.]	[S1.] < [S2.]



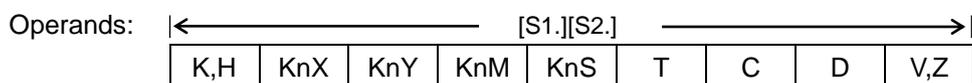
- ◆ 當來源資料 [S1.][S2.] 的最上位位(16 位命令: b15, 32 位命令: b31)為 1 時，視為負值來執行比較。
- ◆ 使用 32 位的計數器 (C200~) 做比較時，必須使用 32 位命令來執行。  
若使用 16 位命令做比較，則會發生程式異常或演算異常。

## FNC(232)接點型比較命令串聯接續 AND=, AND>, AND<, AND<>, AND<=,

### AND>=

FNC(232~238)		16 bits: ----- 5 steps			J2n--	J3n--
D	AND ※	32 bits: ----- 9 steps				

※ :=, >, <, <>, ≤, ≥



Operands:

X	Y	M	S
---	---	---	---

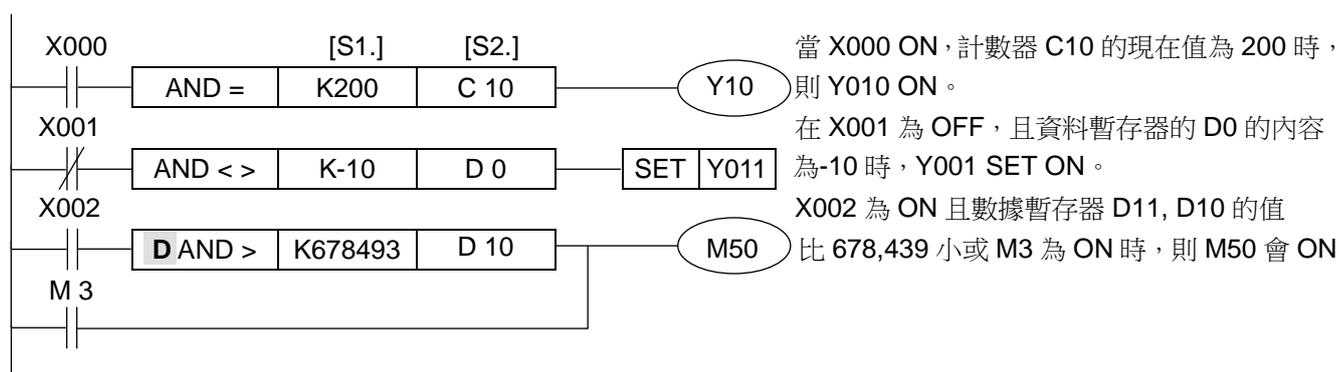
元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●		●	●			●	●		
[S2.]								●	●	●	●	●	●	●		●	●			●	●		

影響旗號:

- ◆ 2 個來源運算元的內容做 BIN 比較，對應比較的結果，執行後段的順序程式。

AND ※ 為和其他接點串聯連接的接點型比較命令。

FNC No.	16 位命令	32 位命令	導通條件	非導通條件
232	AND =	<b>D</b> AND =	[S1.] = [S2.]	[S1.] ≠ [S2.]
233	AND >	<b>D</b> AND >	[S1.] > [S2.]	[S1.] ≤ [S2.]
234	AND <	<b>D</b> AND <	[S1.] < [S2.]	[S1.] ≥ [S2.]
236	AND <>	<b>D</b> AND <>	[S1.] ≠ [S2.]	[S1.] = [S2.]
237	AND ≤	<b>D</b> AND ≤	[S1.] ≤ [S2.]	[S1.] > [S2.]
238	AND ≥	<b>D</b> AND ≥	[S1.] ≥ [S2.]	[S1.] < [S2.]



- ◆ 當來源資料 [S1.][S2.] 的最上位位(16 位命令:b15，32 位命令:b31)為 1 時，視為負值來執行比較。
- ◆ 使用 32 位的計數器(C200~)做比較時，必須使用 32 位命令來執行。若使用 16 位命令做比較，則會發生程式異常或演算異常。

## FNC(240)接點型比較命令並聯接續 OR=, OR>, OR<, OR<>, OR<=, OR>=

FNC(240~246)		16 bits: ----- 5 steps			J2n--	J3n--
D	OR ※	32 bits: ----- 9 steps				

※ :=, >, <, <>, ≤, ≥

Operands:  $\leftarrow$  [S1.][S2.]  $\rightarrow$

K,H	KnX	KnY	KnM	KnS	T	C	D	V,Z
-----	-----	-----	-----	-----	---	---	---	-----

Operands:

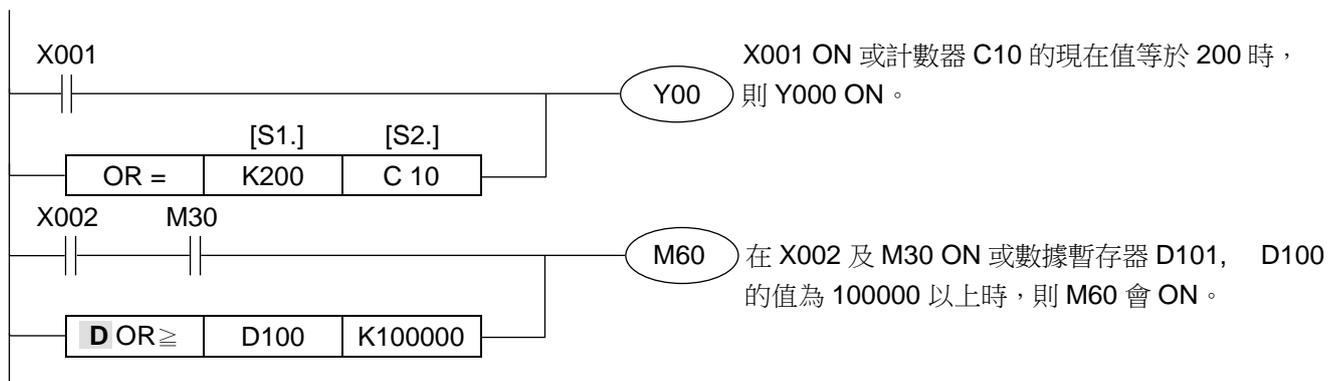
X	Y	M	S
---	---	---	---

元件種類	位元元件							位元組合元件				字元元件				指標			字串		常數		實數
	X	Y	M	S	T	C	Dn.b	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	P		K	H	E	
[S1.]								●	●	●	●	●	●	●		●	●				●	●	
[S2.]								●	●	●	●	●	●	●		●	●				●	●	

影響旗號:

◆ 2 個來源運算元的內容做 BIN 比較，對應比較的結果，執行後段的順序程式。OR ※ 為連接母線的接點型比較命令。

FNC No.	16 位命令	32 位命令	導通條件	非導通條件
240	OR =	D OR =	[S1.] = [S2.]	[S1.] ≠ [S2.]
241	OR >	D OR >	[S1.] > [S2.]	[S1.] ≤ [S2.]
242	OR <	D OR <	[S1.] < [S2.]	[S1.] ≥ [S2.]
244	OR <>	D OR <>	[S1.] ≠ [S2.]	[S1.] = [S2.]
245	OR ≤	D OR ≤	[S1.] ≤ [S2.]	[S1.] > [S2.]
246	OR ≥	D OR ≥	[S1.] ≥ [S2.]	[S1.] < [S2.]



- ◆ 當來源資料 [S1.][S2.] 的最上位位 (16 位命令:b15, 32 位命令:b31) 為 1 時，視為負值來執行比較。
- ◆ 使用 32 位的計數器 (C200~) 做比較時，必須使用 32 位命令來執行。若使用 16 位命令做比較，則會發生程式異常或演算異常。

**FNC(256)LIMIT / 上下限限位控制**

FNC(256)			16 bits: LIMIT & LIMIT(P) ----- 9 steps				
D	LIMIT	P	32 bits: DLIMIT & DLIMIT(P) ----- 17 steps				

Reserved

**FNC(257)BAND/ 死區控制**

FNC(257)			16 bits: BAND & BAND(P) ----- 9 steps				
D	BAND	P	32 bits: DBAND & DBAND(P) ----- 17 steps				

Reserved

**FNC(258)ZONE / 區域控制**

FNC(258)			16 bits: ZONE & ZONE(P) ----- 9 steps				
D	ZONE	P	32 bits: DZONE & DZONE (P) ----- 17 steps				

Reserved

**FNC(259)SCL / 定座標(不同點座標資料)**

FNC(259)			16 bits: SCL & SCL (P) ----- 7 steps				
D	SCL	P	32 bits: DSCL & DSCL (P) ----- 13 steps				

Reserved

**FNC(260)DABIN / 10 進制 ASCII→BIN 的轉換**

FNC(260)			16 bits: DABIN & DABIN (P) ----- 5 steps				
D	DABIN	P	32 bits: DDABIN & DDABIN (P) ----- 9 steps				

Reserved

**FNC(261)BINDA / BIN 10 進制 ASCII 的轉換**

FNC(261)			16 bits: BINDA & BINDA (P) ----- 5 steps				
D	BINDA	P	32 bits: BINDA & BINDA (P) ----- 9 steps				

Reserved

**FNC(269)SCL2 / Scaling 2 定座標 2(X/Y 座標資料)**

FNC(269)			16 bits: SCL2 & SCL2 (P) ----- 7 steps				
D	SCL2	P	32 bits: DSCL2 & DSCL2 (P) ----- 13 steps				

Reserved

**FNC(280)HSCT / High 高速計數器表比較**

FNC(280)							
D	HSCT	P	32 bits: DHSCT & ----- 21 steps				

Reserved

## Note