

## 第 9 章：FBs-PLC 之中斷

### 9.1 中斷之原理與架構

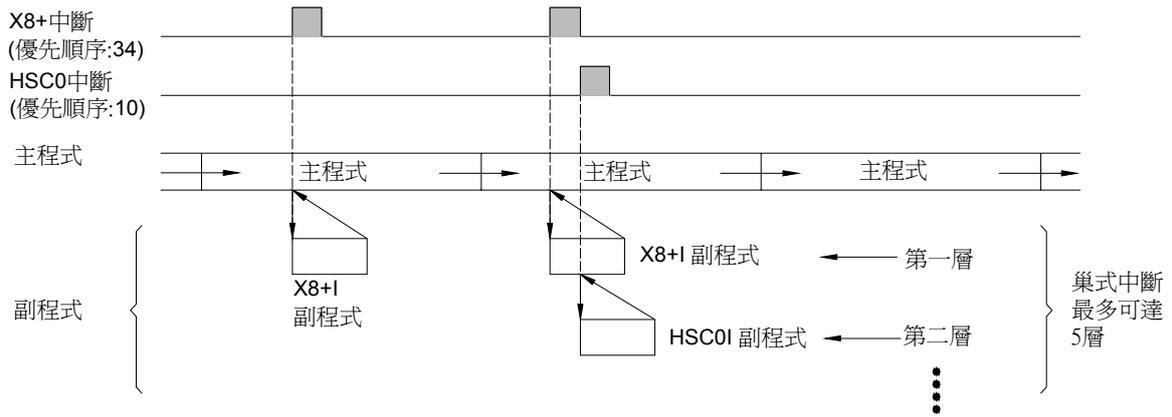
FBs-PLC 所需執行之工作繁複眾多，有 20K Words 的使用者程式要解析，512 點 I/O 狀態要抓取或更新，有 5 個通訊埠需服務…，但 CPU 只有一個，故任一時間只能執行一項工作，因此 PLC 只能依序地將上述所有工作由第一項工作開始逐一地執行到最後一項工作為止，再循環回到第一項工作重覆同樣的工作循環，如此週而復始地作掃描 (Scan) 服務工作，每一項工作在一次掃描循環中均被執行一次，每一次被執行之間隔時間即所謂 PLC 之掃描時間 (Scan Time)。因 CPU 之工作速度較之人類反應，可謂極端快速，上述龐大的工作量通常在數毫秒到數十毫秒 (mS) 即可完成，因此就人類的感覺，PLC 幾乎是在同一時間完成所有工作，而能達到實用之控制效果。

對大部份之應用，上述依序掃描之控制方式均已足夠，但對某些需要高速反應之應用場合 (例如定位控制…等)，掃描時間之延遲即代表誤差之擴大，其反應甚至要求到微秒 (uS) 的速度，才能達到精度要求。在此情形下，唯有利用中斷 (Interrupt) 功能才能達到。

所謂中斷係指 PLC 在平常依序執行之掃描循環中，當有需立即反應之需求發生時，馬上對 CPU 發出中斷要求 (Interrupt Request)；CPU 在收到中斷要求後，立即停止其正在執行之掃描工作，優先地去執行該中斷要求所指定之服務工作；等該服務工作完成後，再回到剛才被中斷之處 (稱之為中斷返回：Return from Interrupt，簡稱 RTI)，繼續執行其未完成之掃描工作。

上述所謂之“中斷要求所指定之服務工作”，即所謂“中斷服務程式”(Interrupt Service Routine)。它是由一連串在中斷發生時“所需執行動作之階梯圖程式”所組成之副程式。放置於副程式區，並以其中斷訊號名稱為其標記 (LABEL) 名稱 (請參閱 9.3 節之說明)。因其放置於副程式區，故在正常之 PLC 掃描循環中是不會被執行到 (PLC 只掃描主程式區，不掃描副程式區)。

雖然 CPU 能在中斷要求發生時，在數十微秒內立即去執行對應之控制動作，但當中斷輸入不祇 1 個時 (如 FBs-PLC 多達 49 個中斷)，只有在其所對應之中斷發生時，才會跳入執行，因 CPU 任一時間只能執行一次動作，因此同樣的問題仍將出現，必須等一個中斷服務程式執行完畢後，才能執行下一個中斷服務程式；如此可能造成數百微秒甚至毫秒之反應延遲，因此在多重中斷輸入結構時，會將各中斷輸入依其重要性給予其不同之中斷優先順序 (Interrupt Priority)。當 PLC 接受某一中斷要求而正執行該中斷之服務程式的當時，若有另一個中斷要求發生，而其優先順序低於正在執行之中斷，CPU 將不理會該中斷，須等 CPU 執行完副程式返回後才會接受，但若其優先順序高於正在執行者，CPU 將立即停止其正在執行之中斷服務程式之執行，而立即跳入該更高順位中斷之中斷服務程式去執行，俟其完成後，再回到剛才被中斷之較低順位服務程式中去繼續完成未完成之工作，此種中斷執行中又被中斷之情形稱之為巢式中斷 (Nested Interrupt)。FBs-PLC 之巢式中斷最多可達 5 層，下圖為單一中斷與巢式中斷之範例：

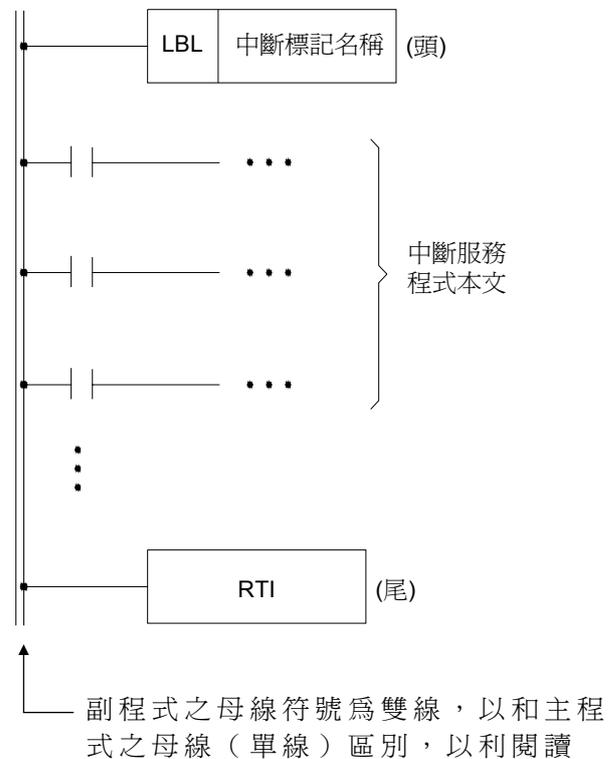


## 9.2 中斷服務程式之結構與其應用

中斷和呼叫雖然同樣都用到副程式，但其召用（跳到副程式去執行）之方式卻不同。呼叫係在主程式中利用執行到 **CALL** 指令（**FUN67**）時，CPU 會記下 **CALL** 指令所指定之副程式名稱，並至副程式區執行該標記名稱之副程式，一直執行到 **RTS**（**Return from Subroutine**）指令後，才返回主程式。

中斷之召用則非利用軟體指令，而是由硬體電路發出中斷信號給 CPU，而由 CPU 自行辨別該中斷之名稱而自動跳入副程式中以該中斷名稱為標記之“中斷服務程式”去執行，一直執行到 **RTI** 指令（**Return from Interrupt**）後，才返回主程式。

如前述中斷服務程式必須放置於副程式區，其結構如右圖所示，有“頭”有“尾”及服務程式本文。“頭”即為該中斷之“中斷標記名稱”，請參考下節之說明，而“尾”即是 **RTI** 指令（**FUN69**），是告訴 CPU 中斷副程式之結束，而跳回主程式或上一層中斷服務程式（巢式中斷時），請參閱 **FUN69**（**RTI**）指令之說明。而頭尾中間則為中斷服務程式本體，用以告知 CPU 在該中斷發生時須執行那些控制動作。



### 9.3 FBs-PLC 之中斷元件、標記名稱與優先順序

如前節所述，任一中斷之“中斷服務程式”均必須有一唯一之“中斷標記名稱”。FP-08 或 WinProladder 在 FBs-PLC 之副程式區內，為 FBs-PLC 所有 49 個中斷保留 49 種對應之“中斷標記名稱”，我們稱之為中斷保留字（Interrupt Reserved Word），在副程式區內這 49 種保留字當作“中斷標記”使用，其他標記名稱不能與之重覆。

“中斷標記名稱”（保留字）之取名原則是將各中斷元件之硬體名稱（例如 HSTA、HSC0、X0+、X0-、…）後面加一個 I 字即得，例如高速計數器 HSC0 之中斷標記名稱為“HSC0I”，X0+之中斷標記名稱為“X0+I”，以下為 FBs-PLC 之 49 種中斷元件之“中斷標記名稱”及其優先順序。

下表為中斷元件及其中斷標記名稱，而為了與之前版本相容，除了 HSC/HST 外，之前舊版本中斷標記名稱亦列入表內（標記名稱有小括號者）；使用原則以新標記名稱為優先（優先使用 HSTAI、1MSI~100MSI、X0+I~X15-I），當使用中斷標記名稱而無中斷處理服務時，可更改中斷標記名稱為舊版本之中斷標記名稱，如 ATMRI、1MS~100MS、INT0~INT15-，如正常，則建議儘快更新 WinProladder 或 FP-08 版本。

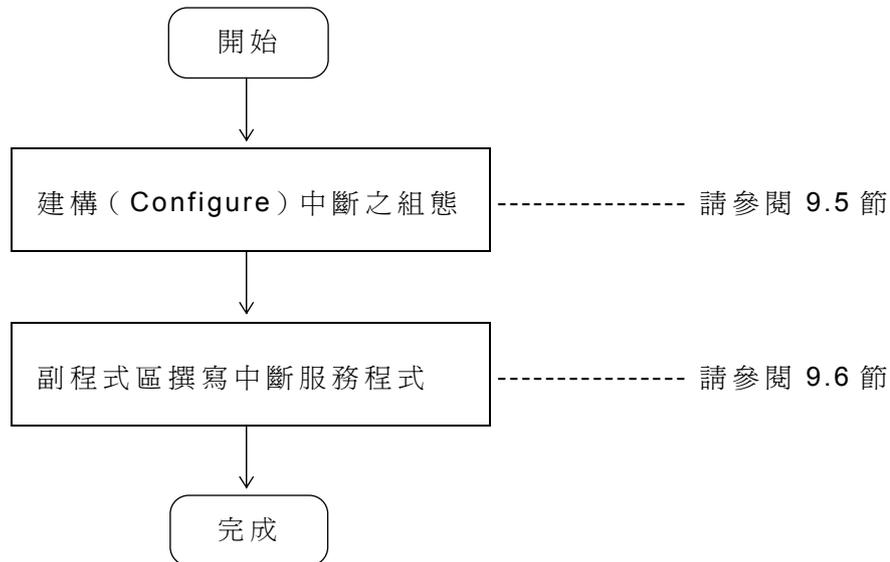
（越上層優先度越高）

中斷元件	優先順序	中斷標記名稱	中斷發生之條件	備註
高速計時器	1	HSTAI (ATMRI)	HSTA 計時到 (CV=PV)	當循環計時器用時無中斷
內部 定 時 時 基	2	1MSI (1MS)	每 1mS 週期發出一中斷	任一時間只能一種時基之中斷作動，請參閱 9.5.2 節之說明，故實際中斷數為 42 個
	3	2MSI (2MS)	每 2mS 週期發出一中斷	
	4	3MSI (3MS)	每 3mS 週期發出一中斷	
	5	4MSI (4MS)	每 4mS 週期發出一中斷	
	6	5MSI (5MS)	每 5mS 週期發出一中斷	
	7	10MSI (10MS)	每 10mS 週期發出一中斷	
	8	50MSI (50MS)	每 50mS 週期發出一中斷	
	9	100MSI (100MS)	每 100mS 週期發出一中斷	
HSC / HST	10	HSC0I/HST0I	HSC0/HST0 計數/時到(CV=PV)	HSC0~HSC3 建構為高速計數器時，其中斷標記名稱為 HSC0I~HSC3I； HSC0~HSC3 建構為高速計時器時，其中斷標記名稱為 HST0I~HST3I
	11	HSC1I/HST1I	HSC1/HST1 計數/時到(CV=PV)	
	12	HSC2I/HST2I	HSC2/HST2 計數/時到(CV=PV)	
	13	HSC3I/HST3I	HSC3/HST3 計數/時到(CV=PV)	
PSO	14	PSO0I	PSO0 脈波輸出完成	
	15	PSO1I	PSO1 脈波輸出完成	
	16	PSO2I	PSO2 脈波輸出完成	
	17	PSO3I	PSO3 脈波輸出完成	

中斷元件	優先順序	中斷標記名稱	中斷發生之條件	備註
外界硬體輸入中斷或軟體高速計數器中斷	18	X0+I (INT0)	X0 由 0→1 (↑) 發出中斷	HSC4~HSC7 之計數輸入與控制輸入可任意指定為 X0~X15 之任一輸入，因此其中斷優先順序需視其指定之 X 號碼決定
	19	X0-I (INT0-)	X0 由 1→0 (↓) 發出中斷	
	20	X1+I (INT1)	X1 由 0→1 (↑) 發出中斷	
	21	X1-I (INT1-)	X1 由 1→0 (↓) 發出中斷	
	22	X2+I (INT2)	X2 由 0→1 (↑) 發出中斷	
	23	X2-I (INT2-)	X2 由 1→0 (↓) 發出中斷	
	24	X3+I (INT3)	X3 由 0→1 (↑) 發出中斷	
	25	X3-I (INT3-)	X3 由 1→0 (↓) 發出中斷	
	26	X4+I (INT4)	X4 由 0→1 (↑) 發出中斷	
	27	X4-I (INT4-)	X4 由 1→0 (↓) 發出中斷	
	28	X5+I (INT5)	X5 由 0→1 (↑) 發出中斷	
	29	X5-I (INT5-)	X5 由 1→0 (↓) 發出中斷	
	30	X6+I (INT6)	X6 由 0→1 (↑) 發出中斷	
	31	X6-I (INT6-)	X6 由 1→0 (↓) 發出中斷	
	32	X7+I (INT7)	X7 由 0→1 (↑) 發出中斷	
	33	X7-I (INT7-)	X7 由 1→0 (↓) 發出中斷	
	34	X8+I (INT8)	X8 由 0→1 (↑) 發出中斷	
	35	X8-I (INT8-)	X8 由 1→0 (↓) 發出中斷	
	36	X9+I (INT9)	X9 由 0→1 (↑) 發出中斷	
	37	X9-I (INT9-)	X9 由 1→0 (↓) 發出中斷	
	38	X10+I (INT10)	X10 由 0→1 (↑) 發出中斷	
	39	X10-I (INT10-)	X10 由 1→0 (↓) 發出中斷	
	40	X11+I (INT11)	X11 由 0→1 (↑) 發出中斷	
	41	X11-I (INT11-)	X11 由 1→0 (↓) 發出中斷	
	42	X12+I (INT12)	X12 由 0→1 (↑) 發出中斷	
	43	X12-I (INT12-)	X12 由 1→0 (↓) 發出中斷	
	44	X13+I (INT13)	X13 由 0→1 (↑) 發出中斷	
	45	X13-I (INT13-)	X13 由 1→0 (↓) 發出中斷	
	46	X14+I (INT14)	X14 由 0→1 (↑) 發出中斷	
	47	X14-I (INT14-)	X14 由 1→0 (↓) 發出中斷	
	48	X15+I (INT15)	X15 由 0→1 (↑) 發出中斷	
	49	X15-I (INT15-)	X15 由 1→0 (↓) 發出中斷	

## 9.4 如何使用 FBs-PLC 之中斷

中斷之用法無論是內部定時中斷、外界輸入中斷、HSC/HST 中斷或 PSO 中斷，其用法均類似，因 HSC/HST 及 PSO 已有專門章節敘述其用法，本節僅對內部定時中斷及外界輸入中斷作範例說明。



## 9.5 中斷之建構(Configuration)

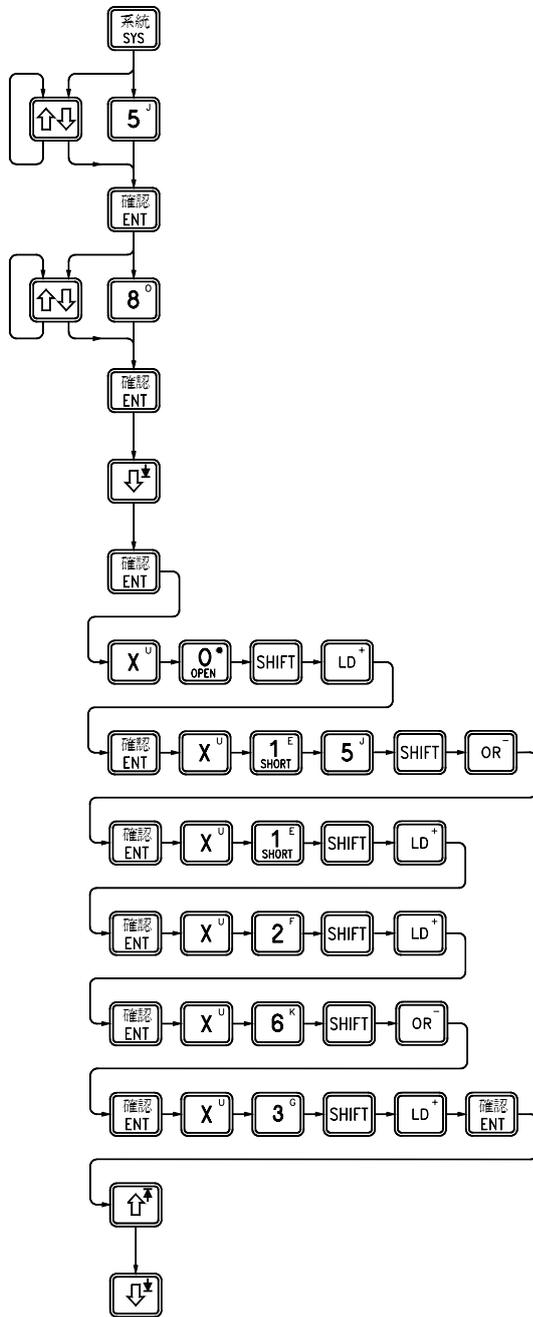
中斷之建構實質上僅僅是指定某一中斷要不要使用而已。

中斷之建構區分為和 I/O 有關之建構及和 I/O 無關之建構兩種。與 I/O 有關之 HSC/HST、PSO 及外界中斷，必須在 WinProladder 或 FP-08 之系統模式之第 5 項(建構:Configuration) 功能下來執行。祇要建構完成，便自動啓動 (Enable) 該裝置之中斷。

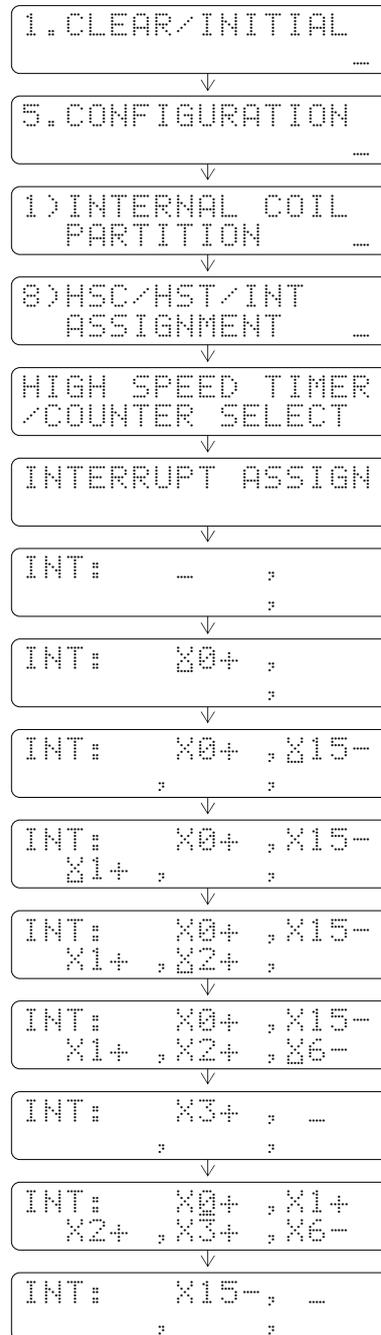
另外和 I/O 無關之“內部定時中斷”則不需作中斷建構；祇要在副程式區有出現定時中斷保留字所起始之中斷處理副程式，即代表該中斷已被規劃，而利用特殊暫存器 R4162 之低位元組之 B0~B7 來彈性指定 1MSI~100MSI 定時中斷是否被允許執行。

### 9.5.1 以 FP-08 作“外界中斷”之建構範例

【按鍵操作】



【LCD 畫面顯示】



- 外界中斷係和 HSC 及 SPD 指令共用 X0~X15 等 16 個高速輸入點，故 HSC 或 SPD 使用過之輸入點號碼，即不能建構於外界中斷。  
註：SPD 指令僅能使用 X0~X7 等 8 個輸入點來作平均速度偵測。
- 中斷建構指定一旦完成便無法在 PLC RUN 中變更，但 FBs-PLC 提供 EN 指令 (FUN145) 及 DIS 指令 (FUN146)，可對外界中斷、HSC 及 HSTA 之中斷作允許或禁止之控制，使之能在 PLC RUN 中亦能動態變化控制，請參考該兩指令之說明。

### 9.5.2 以 WinProladder 作“外界中斷”之建構範例

於專案視窗中點選 I/O 組態設定：

專案名稱

系統組態

I/O 組態 → 選擇“中斷設定”

出現中斷設定畫面後，可直接於視窗中點選欲發生中斷之外界輸入：



### 9.5.3 以 R4162 來指定內部定時中斷

當副程式區有出現內部定時中斷保留字（8 種，1MSI~100MSI）所起始之中斷處理副程式時，即代表該中斷已被規劃，而根據實際應用需求，可以特殊暫存器 R4162 低位元組之 8 個 Bit 來指定要遮沒那個時基之定時中斷，其分配如下：

	B7	B6	B5	B4	B3	B2	B1	B0
R4162 :	100MS	50MS	10MS	5MS	4MS	3MS	2MS	1MS

- Bit 狀態=0 時：允許該時基之定時中斷（不遮沒）
- Bit 狀態=1 時：關閉該時基之定時中斷（遮沒）

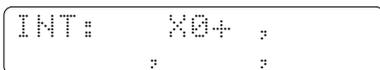
- B0~B7 間如同時有多個 Bit 為 0 時，則 FBs-PLC 將只啟動時基最小且其中斷處理副程式有被規劃者，而關閉其他時基（例如內容值為 00H 時，全部定時中斷皆未被遮沒；但如 1MS 與 2MS~100MS 定時中斷處理副程式皆出現在副程式區時，祇有 1MS 定時中斷能被執行，其它不被執行）。

- 因 R4162 之值可在 PLC RUN 中由使用者以階梯圖程式隨時改變，故能動態地改變時基，或暫停、啓動中斷，彈性甚大。
- R4162 內定值為 0，代表 1MS~100MS 定時中斷皆未被遮沒，祇要副程式區有任一個定時中斷處理副程式，則該中斷副程式會定時被執行。
- 因 CPU 每次被中斷均固定要花費相當之執行時間，故定時中斷時基愈小，中斷愈頻繁，則佔用 CPU 之時間愈多，因此應用上應持適用即可之原則，以免劣化 CPU 之效能。

## 9.6 中斷程式之範例

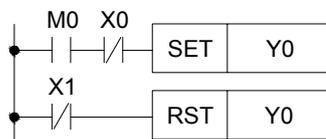
**範例 1** 以外界硬體中斷 X0+作單接點之精密定位控制範例  
 X0：定位到感應器  
 X1：緊急停止  
 Y0：運轉控制馬達

### 【外界中斷建構】



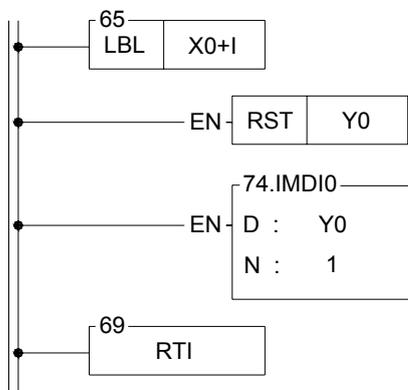
- 建構輸入 X0 由 0→1 時發出中斷

### 【主程式】



- M0（啓動）由 0→1 時，運轉控制馬達 ON

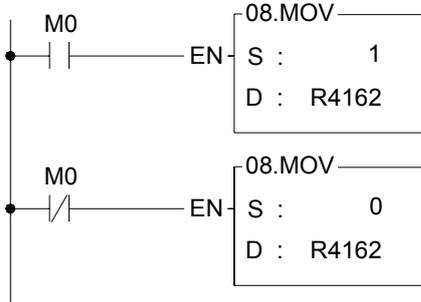
### 【副程式】



- 當感應器 X0 檢知定位位置到時，即 X0 由 0→1，硬體自動執行此中斷副程式
- 運轉控制馬達 Y0 變 0，立即停止馬達運轉
- 將 Y0 立即輸出，減少因掃描時間引起之延遲
- 在中斷處理副程式中必須利用立即輸入／出指令才能達到及時高速精密控制之效果。

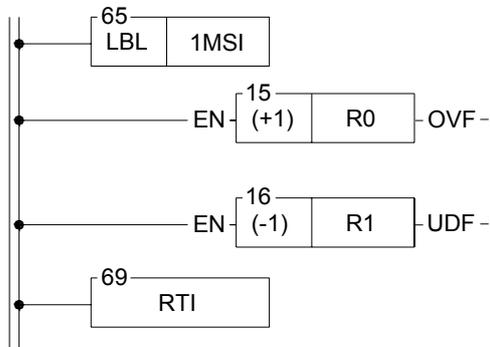
**範例 2** 1MS 內部定時中斷

**【主程式】**



- 當 M0=1 時，關閉 1MS 定時中斷  
(1MS 定時中斷被遮沒)
- 當 M0=0 時，允許 1MS 定時中斷

**【副程式】**

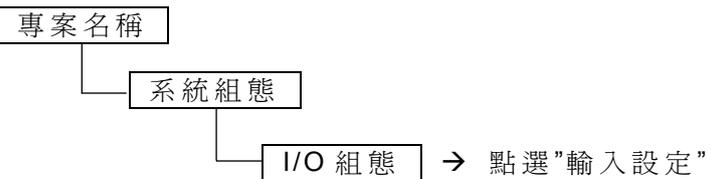


- 啓動 1MS 定時中斷後，則系統自動每隔 1MS 執行此中斷副程式
- R0 當成每 1MS 時基之上數循環計時器
- R1 當成每 1MS 時基之下數循環計時器

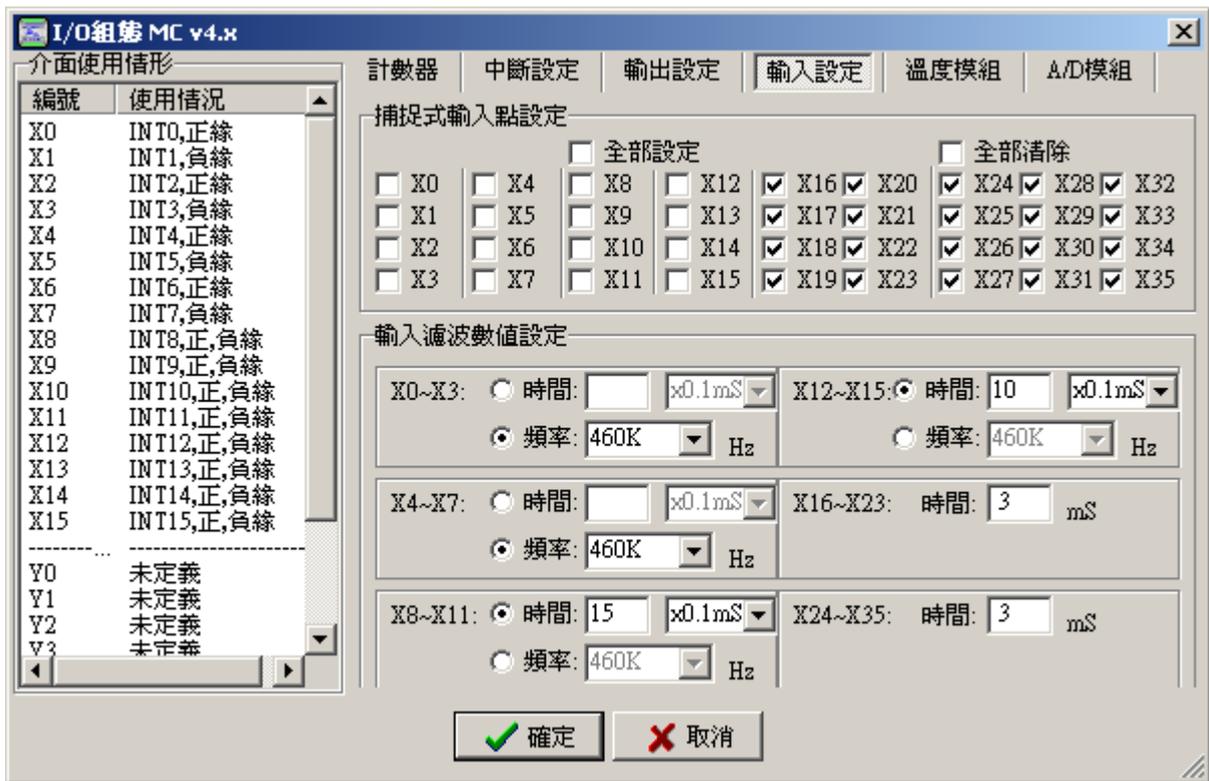
**9.7 捕捉式輸入及數位濾波**

針對許多高速應用的場合，除了可以使用中斷輸入方式來防止信號遺漏外，亦可以將主機內含之輸入點設定為捕捉式輸入來捕捉其一閃即逝的信號。使用捕捉式輸入的設定方式相當的簡單：

於專案視窗中點選 I/O 組態設定：

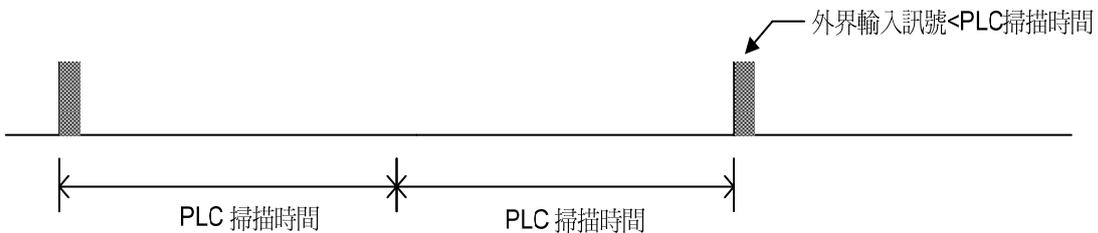


出現捕捉式輸入設定畫面後，直接點選欲捕捉之輸入點數即可。



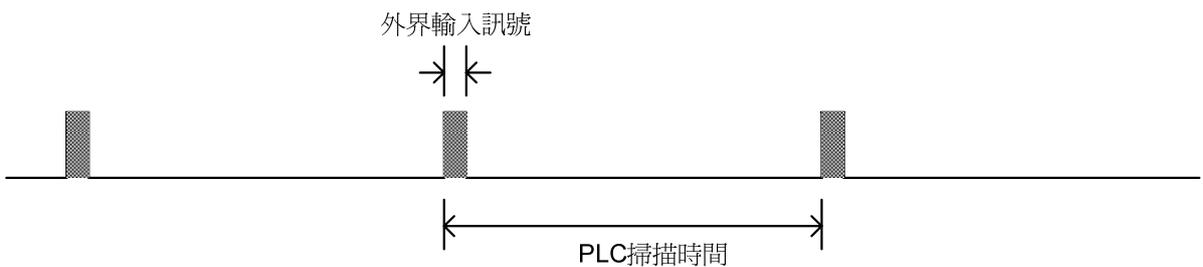
範例說明一

如果欲將捕捉式輸入用於計數時，其輸入週期至少需大於掃描週期的兩倍以上，才不會有遺漏的現象。例如一輸入信號之頻率為 50Hz，則其掃描時間至少須小於 10mS 以上，輸入計數才會正確。



範例說明二

下述之範例即可設定捕捉式輸入來捕捉比 PLC 掃描時間 (Scan Time) 更短暫之外界輸入信號，另外必須注意的是，捕捉式輸入的設定只針對主機之內含點數，擴充點數則無效果。

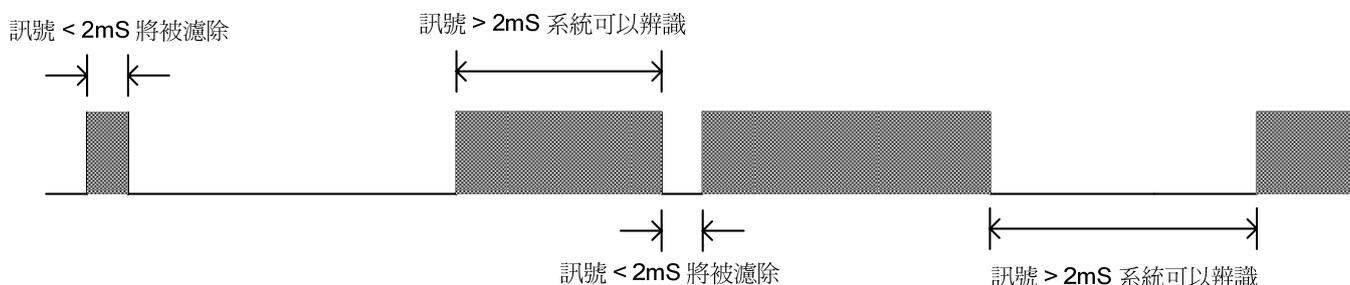


FBs 主機內含之輸入點除了可設定為捕捉式輸入之外，還可將其分成 6 組{ (X0~X3)、(X4~X7)、(X8~X11)、(X12~X15)、(X16~X23)、(X24~X35) }做輸入濾波條件設定。

數位濾波設定可分成對時間或是對頻率設定濾波條件。前四組輸入信號(X0~X15)可作濾波頻率或濾波時間之設定；以頻率作設定時，可設定 14KHz~1.8MHz 等 8 種選擇；以時間作設定時，可設定(1~15)×1mS 或(1~15)×0.1mS。後兩組輸入信號(X16~X35)則只可作時間設定，其設定範圍為(1~15)×1mS。設定輸入濾波時間用於消除低於設定時間之雜訊；設定輸入濾波頻率用於消除高於設定頻率之雜訊。

#### 範例說明一

濾波時間設定為 2mS 時，當信號 ON 或 OFF 時間低於 2mS，則該 ON 或 OFF 信號無法被偵測到。



#### 範例說明二

濾波頻率設定為 28KHz 時，當輸入頻率大於 28KHz 時，則該輸入訊號無法被偵測到。

