

# 第 10 章：FBs-PLC 之高速計數器與高速計時器

## 10.1 FBs-PLC 之高速計數器

一般 PLC 之軟體計數器之計數頻率僅能達數十 Hz（視掃描時間而定），若超過將產生漏數甚至完全無法計數，此時必須使用高速計數器（High-Speed Counter 簡稱 HSC）才能勝任。一般 PLC 之高速計數器有兩種，一為使用專用硬體電路作成之硬體高速計數器（Hardware High-Speed Counter 簡稱 HHSC），另一種是利用計數脈波正/負緣變化時發生中斷，而由 CPU 來判斷增減，並作計數之軟體高速計數器（Software High-Speed Counter 簡稱 SHSC）。FBs-PLC 各有 4 個 HHSC（在 SoC 晶片內）及 4 個 SHSC；其皆為 32 位元高速計數器。

### 10.1.1 FBs-PLC 高速計數器之計數模式

FBs-PLC 之 4 個 HHSC 均具有 8 種計數模式可供選擇，而 SHSC 則提供 3 種計數模式，如下表所示：

計數模式 (MODE)			HHSC (HSC0~HSC3)	SHSC (HSC4~HSC7)	計數行為波形圖	
					上數 (+1)	下數 (-1)
單相獨立	MD 0	U/D	○	○	U D	
	MD 1	U/D×2	○		U D	
單相相關	MD 2	P/R	○	○	P R	
	MD 3	P/R×2	○		P R	
雙相相關	MD 4	A/B	○	○	A B	
	MD 5	A/B×2	○		A B	
	MD 6	A/B×3	○		A B	
	MD 7	A/B×4	○		A B	

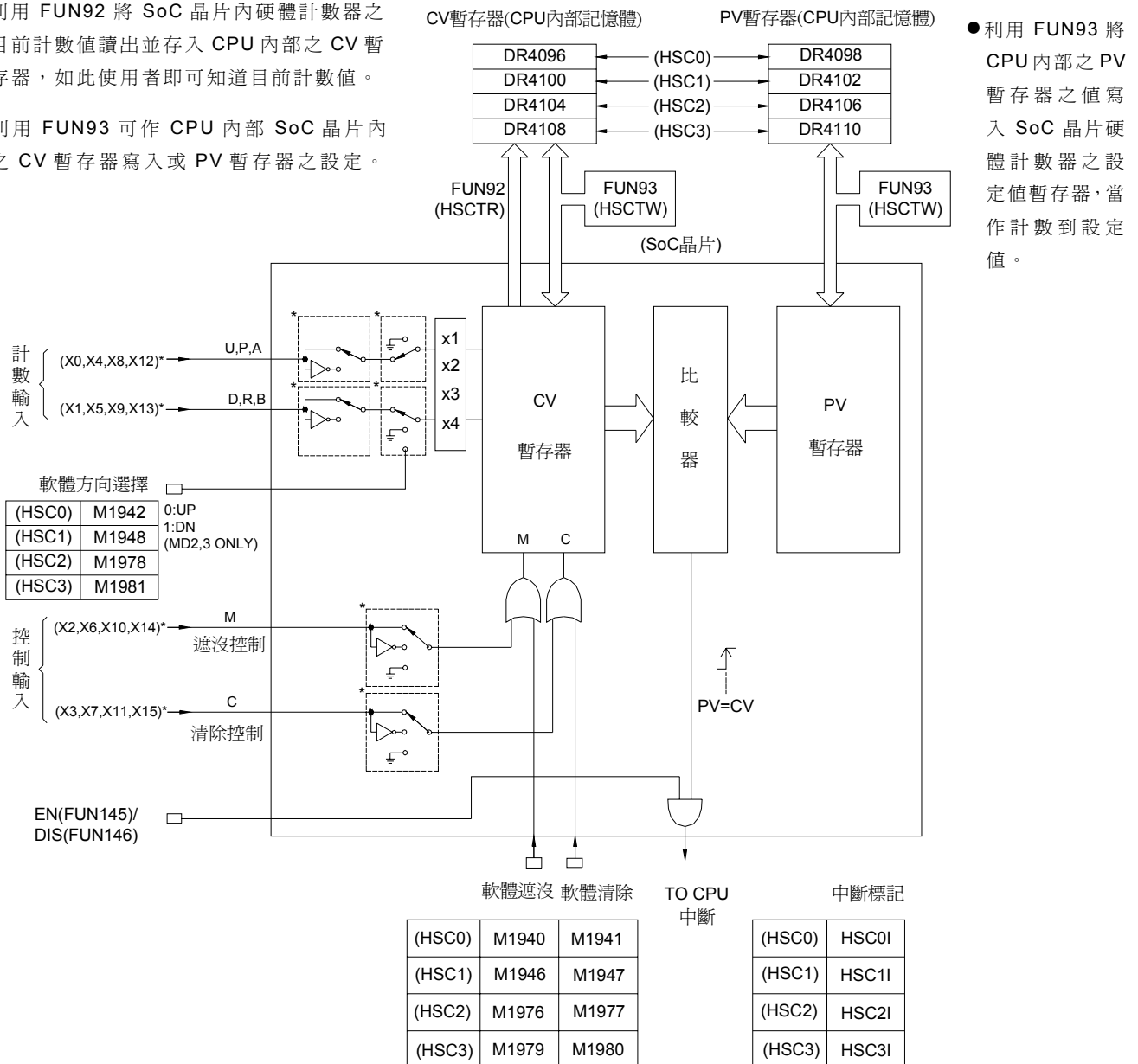
- 波形中正或負緣有上升或下降箭號（↑，↓）者表示發生計數（+1 或 -1）之處

## 10.2 FBS-PLC 高速計數器之系統架構圖

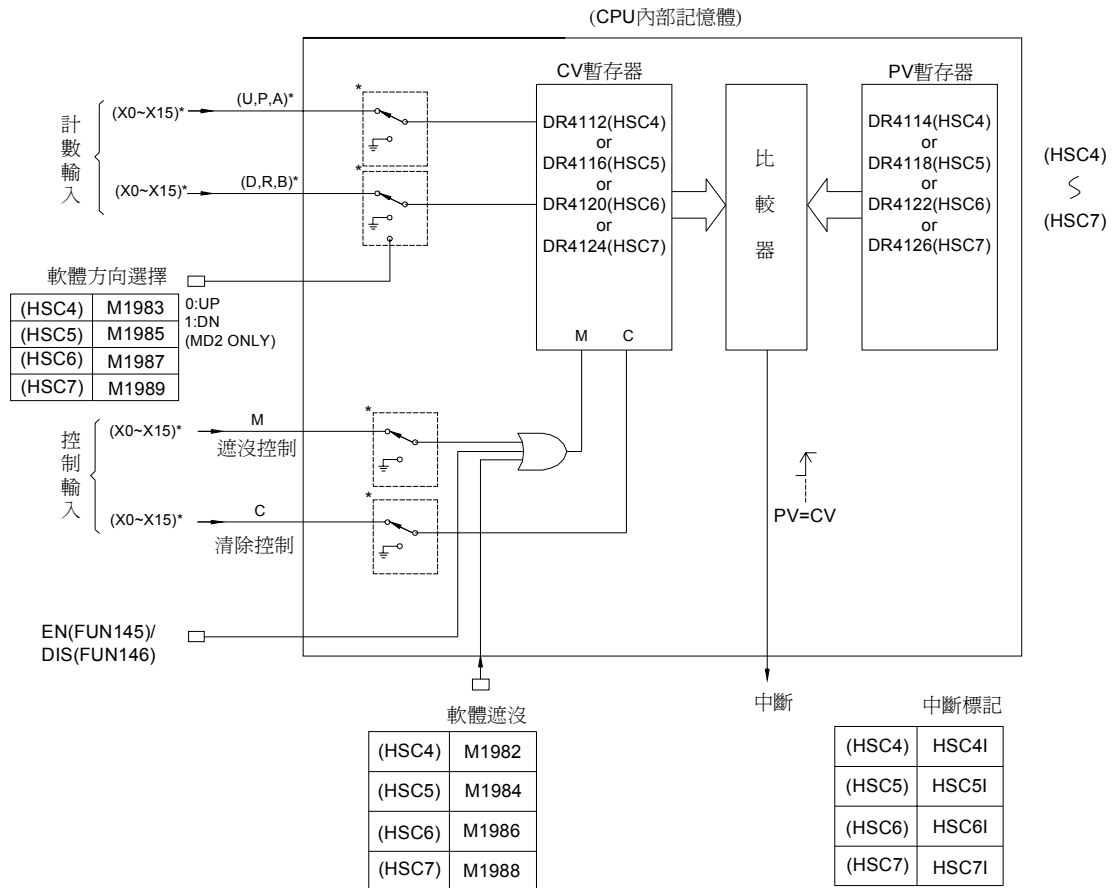
下圖為 FBS-PLC 之 HHSC 及 SHSC 之系統結構圖，兩者均具有多重用途輸入及計數功能，有些功能是內建功能（例如 CV 暫存器號碼、PV 暫存器號碼、中斷標記名稱及軟體遮沒、清除、方向選擇之繼電器號碼）無需使用者作建構指定，有些則必須由使用者利用 WinProladder 或 FP-07C 之系統模式下之第 5 項功能（Configuration）來建構該 HSC 之組態（例如 HSC 之用途選擇，計數模式，各功能輸入是否使用，極性是否反相，對應之輸入點號碼 Xn 之指定……等）。在下圖中標有“\*”記號者，表示此部份需由使用者來建構之。經建構指定之 8 種計數模式之細部結構與計數行為，請參閱 10.2.1~10.2.3 小節之說明。

註：CV (Current Value)，目前值；PV (Preset Value)，設定值。

- 利用 FUN92 將 SoC 晶片內硬體計數器之目前計數值讀出並存入 CPU 內部之 CV 暫存器，如此使用者即可知道目前計數值。
- 利用 FUN93 可作 CPU 內部 SoC 晶片內之 CV 暫存器寫入或 PV 暫存器之設定。



硬體高速計數器 (HSC0~HSC3) 之系統結構圖



軟體高速計數器 (HSC4~HSC7) 之系統結構圖

- 上圖 HHSC 與 SHSC 之所有控制信號均內定為 Active High(亦即在狀態為 1 時發生作用, 0 無影響)。但 HHSC 之計數輸入 (U, D, P, R, A, B) 及控制輸入 (M, C) 均可作極性反相選擇, 以匹配 Sensor 極性。
- 遮沒控制 M (MASK) 係指當此信號為 1 時, HSC 計數脈波將被遮沒不計數, HSC 內部狀態 (CV, PV...) 均保持不變。當 M 回到 "0", HSC 才能正常工作。有些 Sensor 則為致能 (Enable) 輸出, 其功能正好和 MASK 相反, 當 Enable=0 時, 計數器將不計數, 必須在 Enable 回到 1 時始能正常工作, 此時您可利用 MASK 極性選擇倒相輸入, 便可匹配具有 Enable 輸出之 Sensor。
- 清除控制 C (CLEAR) 係指當此信號為 1 時, HSC 內部之 CV 暫存器將被清為 0, 且無法計數, 一直要等到 C 回到 0 後, HSC 才由 0 開始計數。Ladder 程式亦可直接將 CV 暫存器 (DR4112、DR4116、DR4120、DR4124) 清為 0, 將目前計數值清除為 0。
- FBs-PLC 之 4 組 HHSC 均在 SoC 晶片中, 其 CV 暫存器和 PV 暫存器使用者是無法直接存取的, 使用者能直接存取的是 CPU 內部記憶體中對應之 CV 暫存器 (DR4096~DR4110)。理想上, 晶片上之 CV 與 PV 暫存器內容值與 CPU 內部記憶體中之 CV 與 PV 暫存器之內容值應是同步更新, 隨時均為相同的, 但因兩者分屬於不同之硬體電路, 兩者之對應必須靠 CPU 來作載入或讀取。CPU 可利用 FUN93 將目前值或設定值寫入到晶片內之 CV 暫存器 (使 HHSC 由此初始值開始計數) 或 PV 暫存器, 而利用 FUN92 將晶片內 HHSC 之 CV 暫存器之目前計數值讀入並存放於 CPU 內部之 CV 暫存器。但因讀取動作只有在 FUN92 被執行到時才進行 (亦即為 "取樣" 讀取), 因此晶片中 HHSC 之 CV 值和 CPU 內部之 CV 值可能會有落差, 尤其在計數頻率高時誤差更大。

- 當計數頻率不高或定位精度要求不高時，在主程式利用 FUN92 讀取目前計數值，然後再配合比較指令，即可作一般簡單之計數定位控制。
- 當定位精度要求較高或多段計數設定控制時，可在定時中斷處理副程式裡，利用 FUN92 較頻繁去讀取目前計數值，然後配合比較指令作較精緻之計數定位控制。
- 當定位精度要求極高時，在計數前須先將預設之中斷值以 FUN93 寫入 SoC 晶片內 HHSC 之 PV 暫存器，等 HHSC 之 CV 計數值到達此預設值時，HHSC 內之硬體比較器將會在 CV=PV 瞬間，向 CPU 發出中斷，而能立即跳到中斷副程式中作及時之控制或處置。
- SHSC 則是利用中斷方式在計數輸入之上緣時向 CPU 發出中斷，再由 CPU 判斷其為加 1 或減 1，而直接在其內部 CV 暫存器上更新（亦即 CPU 內部之 CV 暫存器本身即為 SHSC 之 CV 暫存器，故無需 FUN92、FUN93 指令）。在每次 CV 更新之同時，CPU 均會比較是否與其 PV 暫存器值相等，若是，立即跳到其對應之 SHSC 之中斷服務程式作及時之處理。因 SHSC 之每一計數輸入及控制輸入之變化都會造成 CPU 中斷，當計數頻率高時，將嚴重佔用 CPU 時間，大幅降低 CPU 之反應速度，甚至造成 Watchdog Time-out，使 PLC 停機。因此應儘量優先使用 HHSC，如需使用 SHSC，所有 SHSC 之輸入頻率總和請勿超過 8KHz。
- 所有軟體遮沒、清除、方向控制等特殊繼電器之控制時效均受限於 PLC 掃描時間，因此並不適合在高精度要求 HSC 之即時控制（主要當作在 HSC 運作前之初始設定）。若需作即時控制，請用硬體控制輸入或使用 FUN145 (EN)、FUN146 (DIS)、FUN92 (HSCTR)、FUN93 (HSCTW) 等指令來控制。
- 所有 HSC 均附加有 Enable (FUN145) 及 Disable (FUN146) 功能，HSC 在 Enable 下能計數且計數到時可產生中斷信號運作；在 Disable 時，則 HHSC 雖能繼續計數，但計數到時，不會產生中斷，而 SHSC 則保持在停滯狀態。在 Configure HSC 時 HSC 是內定為 Enable，程式中可依控制需要隨時 Disable 或 Enable。

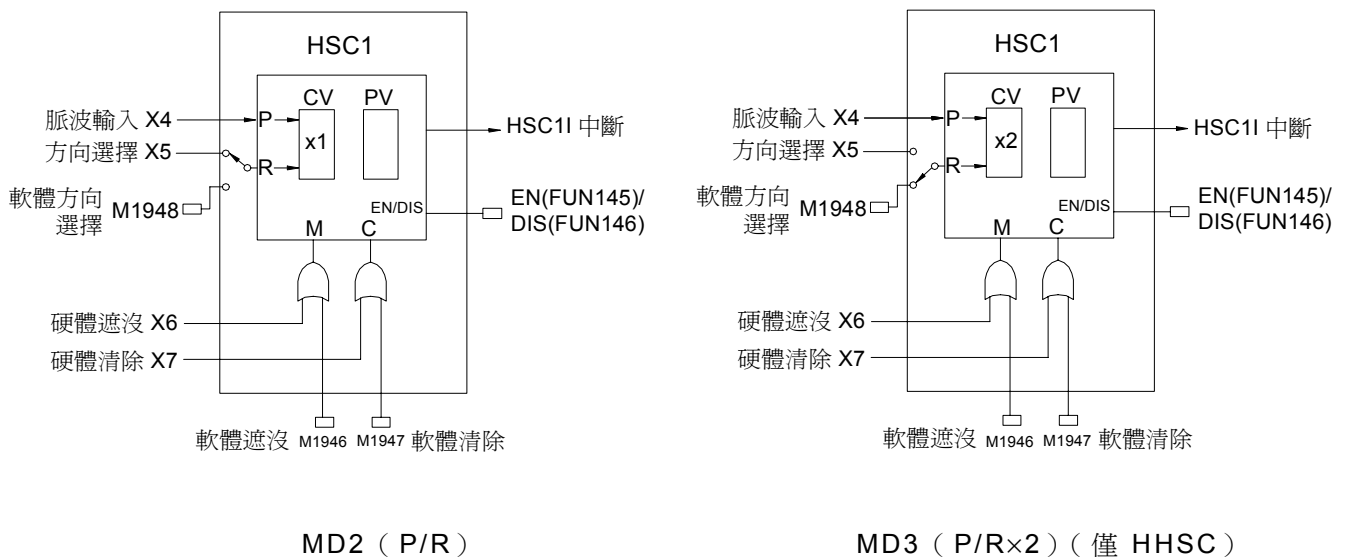
### 10.2.1 單相獨立之上／下數高速計數器 (MD0, MD1)

單相獨立上／下數高速計數器具有兩個獨立之上數脈波輸入 (U) 和下數脈波輸入 (D)，兩者彼此獨立沒有相位關係，各自於其脈波輸入之正緣  $\uparrow$  (MD1 則為正/負緣兩者) 到來時將 CV 值加 1 (U) 或減 1 (D)，即使 U 和 D 脈波正緣 (負緣) 同時發生亦允許 (此時相互抵消)，兩種模式均內建有軟體遮沒和軟體清除 (SHSC 無軟體清除)，當控制功能不使用時使其狀態 (如本例之 M1940、M1941) 保持為 "0" 即可。除內建之軟體遮沒與軟體清除外，尚可規劃使用硬體遮沒與硬體清除控制。軟／硬體遮沒是先 OR 起來，再接到 HSC 之遮沒控制 M 去，軟／硬體清除亦相同方式。下圖係以 HSC0 為例分別建構為 MD0 與 MD1 HSC 之功能示意圖。



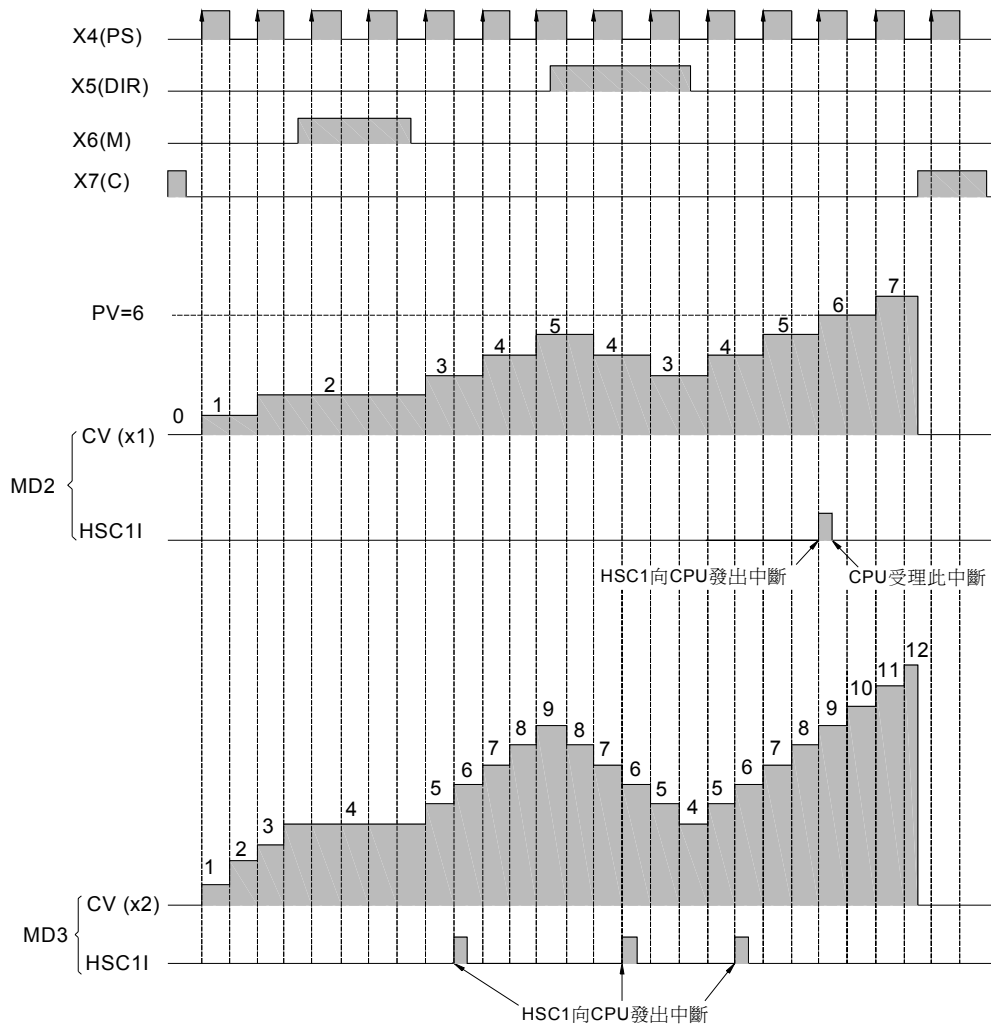
## 10.2.2 單相相關之上／下數高速計數器（MD2，MD3）

單相相關上／下數高速計數器，僅有一個計數脈波輸入 P (Pulse)，而其上／下數則必須由另一個方向輸入 R (Direction) 來決定在計數脈波正緣 (MD3 時則正/負緣兩者) 到來時 CV 值是要 +1 (R=0) 或 -1 (R=1)。MD2 和 MD3 之計數行為類似，差異只在於 MD2 只在 PS 脈波之正緣計數 (+1 或 -1)，而 MD3 則在 PS 之正/負緣均計數 (亦即 MD3 之計數值為 MD2 之兩倍)。兩種模式均內建有軟體遮沒及軟體清除 (SHSC 無軟體清除)，當控制功能不使用時必須使其狀態 (如本例之 M1946 與 M1947) 保持為 0。除內建之軟體遮沒與軟體清除外，尚可建構使用硬體遮沒與硬體清除控制，軟/硬體遮沒是先 OR 起來，再接至 HSC 之遮沒控制 M 去，軟/硬體清除亦同方式，以下為將 HSC1 分別建構為 MD2 與 MD3 HSC 之功能示意圖。



HHSC 工作在 MD2 或 MD3 模式時，可選擇由外界輸入 (如本例之 X5) 來作方向選擇或由 CPU 內部之特殊繼電器 (如本例之 M1948) 來作方向選擇。SHSC 工作在 MD2 模式時，必須透過 CPU 內部之特殊繼電器來做方向選擇。

下圖為本例之兩模式之 HSC 在設定值 PV 為 6 時之計數與控制關係波形圖。

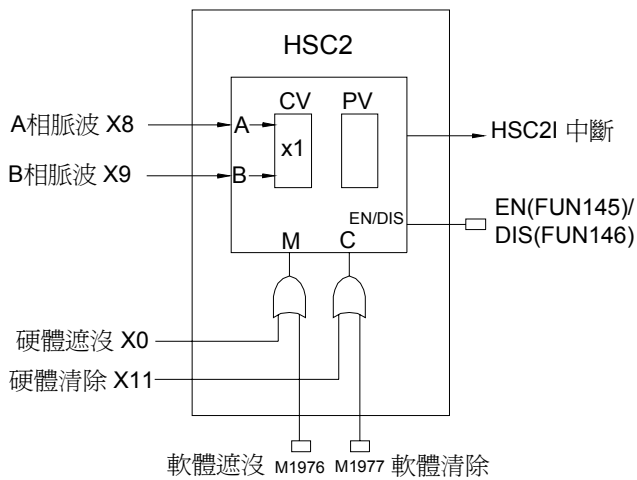


### 10.2.3 雙相高速計數器 ( MD4 , MD5 , MD6 , MD7 )

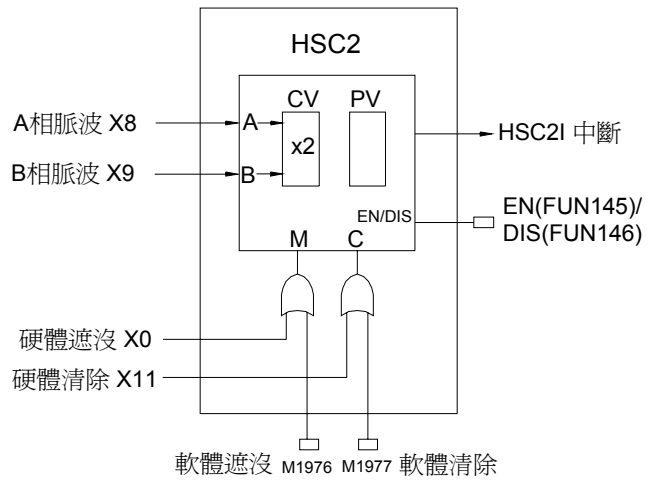
雙相高速計數器具有 A 相與 B 相兩個脈波輸入，其計數值之+1 或-1，係以兩者之相位關係作判斷，亦即為兩相相關之計數。若 A 相超前 B 相則 CV 值+1，反之則-1。雙相 HSC 之四種模式 MD4 (A/B)、MD5 (A/B×2)、MD6 (A/B×3)、MD7 (A/B×4) 之計數行為均相似，其差異在於：

- ① MD4 (A/B) : 在 A 超前 B 時在 A 之正緣+1，而在 A 落後 B 時在 A 之負緣-1。
- ② MD5 (A/B×2) : 在 A 超前 B 時在 A 之正/負緣均+1，而在 A 落後 B 時在 A 之正/負緣均-1 (計數值為 MD4 之 2 倍)。
- ③ MD6 (A/B×3) : 在 A 超前 B 時在 A 之正/負緣及 B 之正緣均+1，而在 A 落後 B 時在 A 之正/負緣及 B 之負緣均-1 (計數值為 MD4 之 3 倍)。
- ④ MD7 (A/B×4) : 在 A 超前 B 時在 A 及 B 之正/負緣均+1，而在 A 落後 B 時在 A 及 B 之正/負緣均-1 (計數值為 MD4 之 4 倍)。

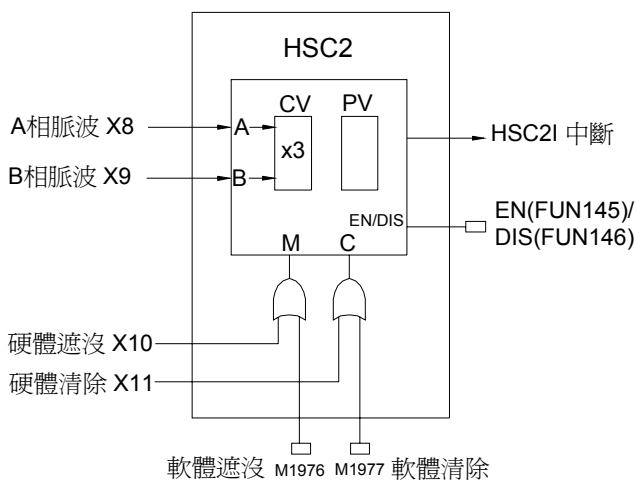
如同其他模式 MD4~MD7 HSC 均內建有軟體遮沒和軟體清除 (SHSC 無軟體清除), 當控制功能不用時必須使其狀態 (如本例之 M1976 和 M1977) 保持為 "0"。同時使用者亦可建構硬體遮沒與硬體清除控制。軟/硬體遮沒是先 OR 起來, 再接到 HSC 之遮沒控制 M 去, 軟/硬體清除亦同方式。下圖係以 HSC2 為例分別建構為 MD4、MD5、MD6、MD7 等 4 種模式之 HSC 功能示意圖。



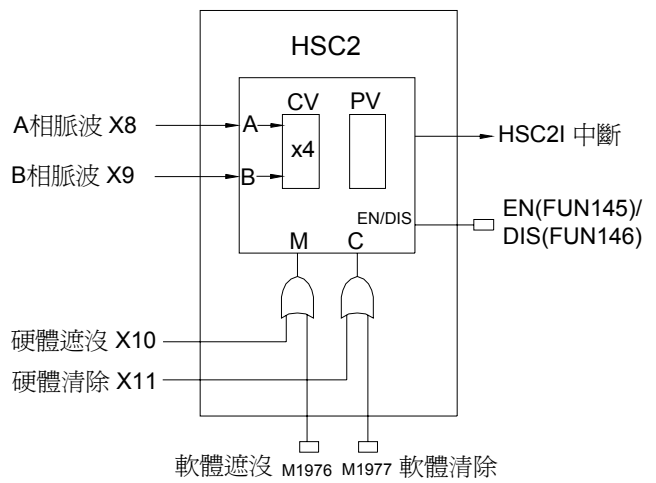
MD4 (A/B)



MD5 (A/Bx2) (僅 HHSC)



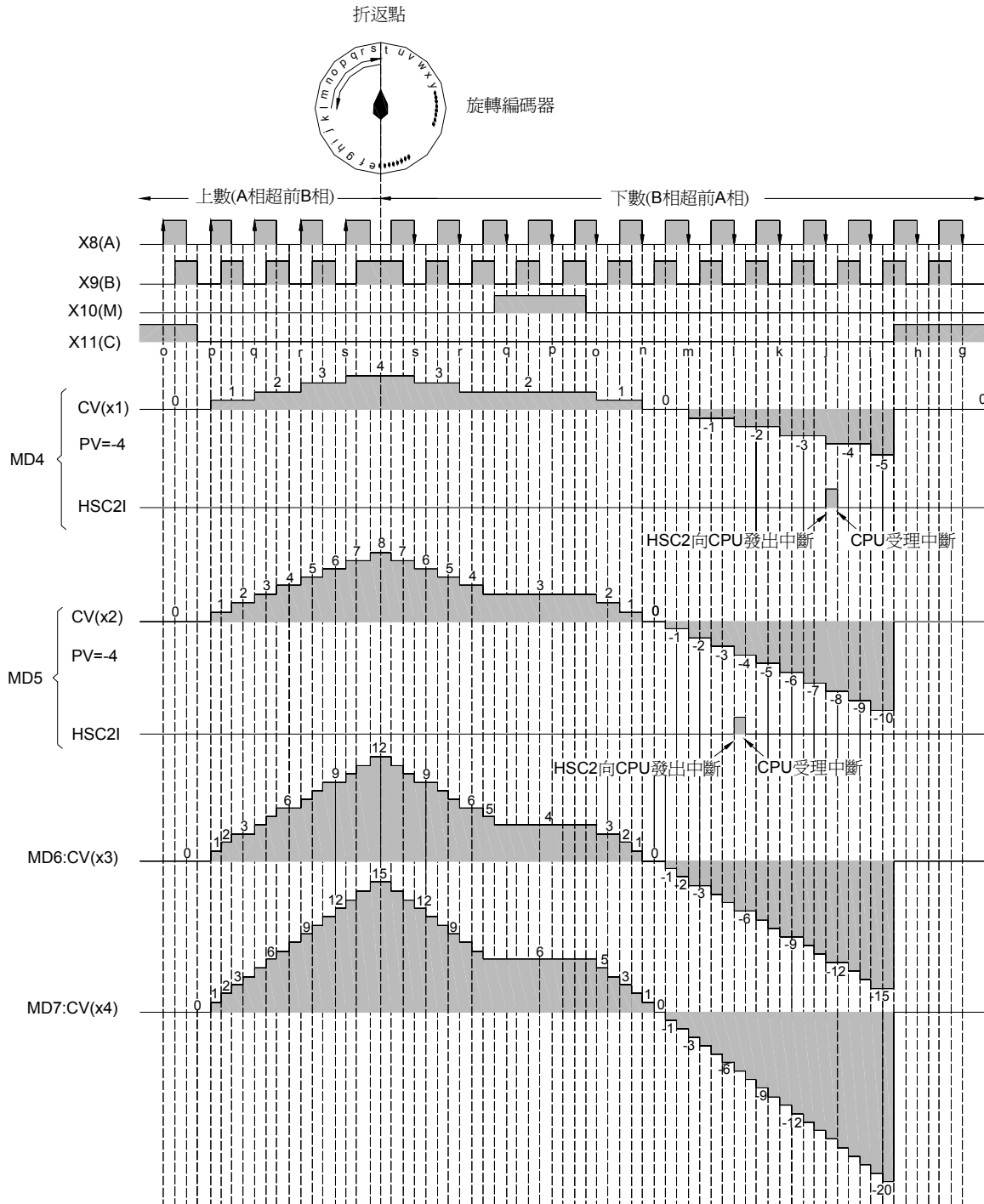
MD6 (A/Bx3) (僅 HHSC)



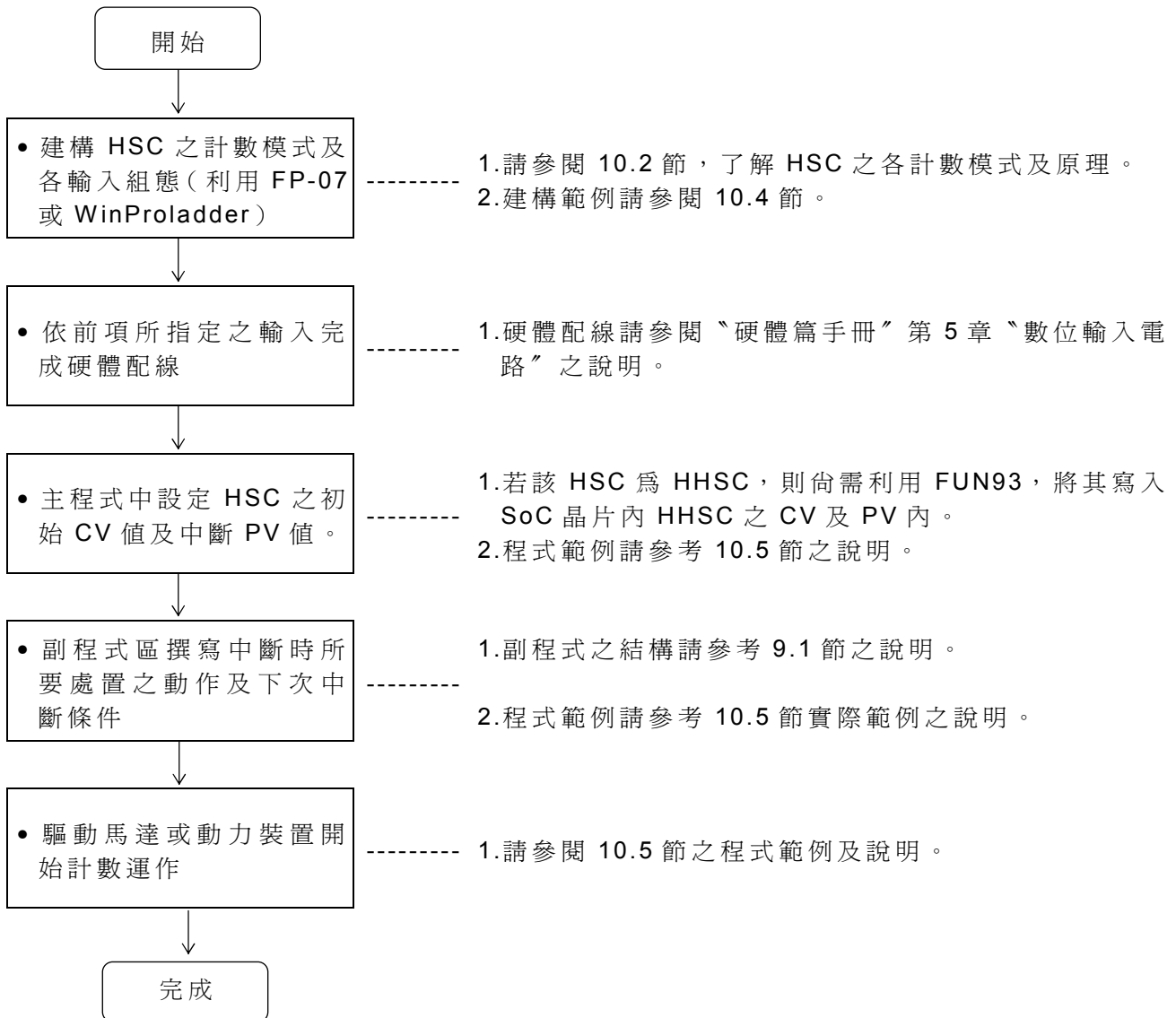
MD7 (A/Bx4) (僅 HHSC)



下圖係以本例之 4 種模式 HSC，在設定值 PV 為 -4 時之計數與控制關係波形圖。



### 10.3 使用 FBs-PLC 高速計數器之步驟



### 10.4 HSC/HST 之建構 (Configuration)

#### 10.4.1 以 FP-07C 作 HSC/HST 之建構

本節將以範例說明 HSC Configuration 之方法，在 HSC 之 Configuration 依序包括下列 5 項：

- ① HSC/HST 之選擇指定 (僅 HHSC 有此選項功能)，選擇為 HST 時無需再作任何建構指定。
- ② 指定各 HSC 之計數模式 (MD0~MD7)。鍵入模式號碼後，FP-07C 將自動顯示該模式 HSC 之計數與控制輸入名稱，並保留空格供使用者鍵入外界輸入點號碼 Xn，模式欄位為空格時表示不使用該 HSC。
- ③ 指定各計數輸入 (U, D, P, R, A, B) 及控制輸入 (M, C) 欲使用與否 (不使用使之保留空格，欲使用則需填入 Xn 值，因 HHSC 之各輸入之 Xn 值均固定，因此只需鍵入字母“X”，FP-07 將自動補上內定之 n 號碼。

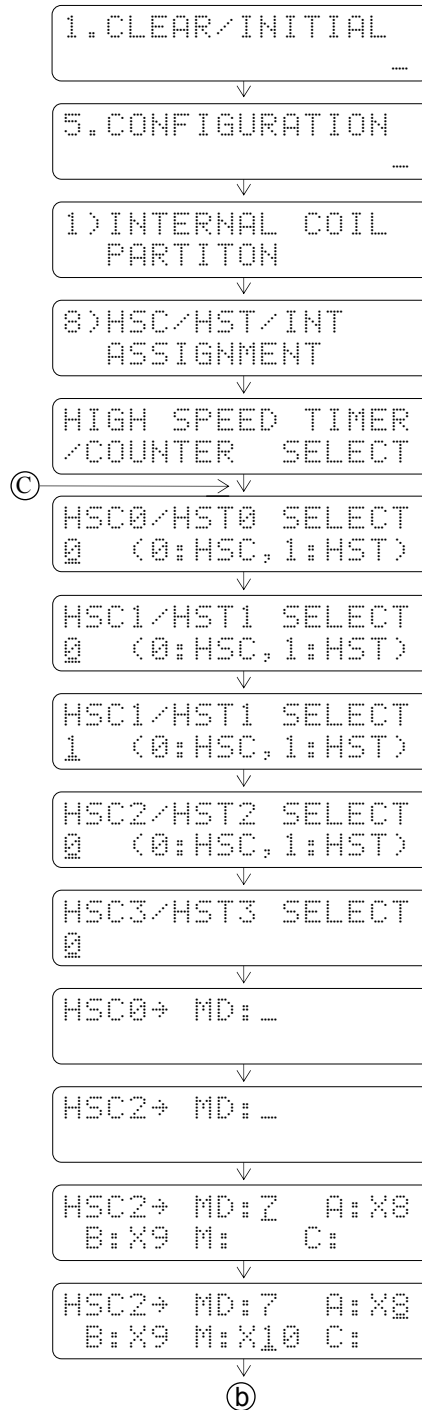
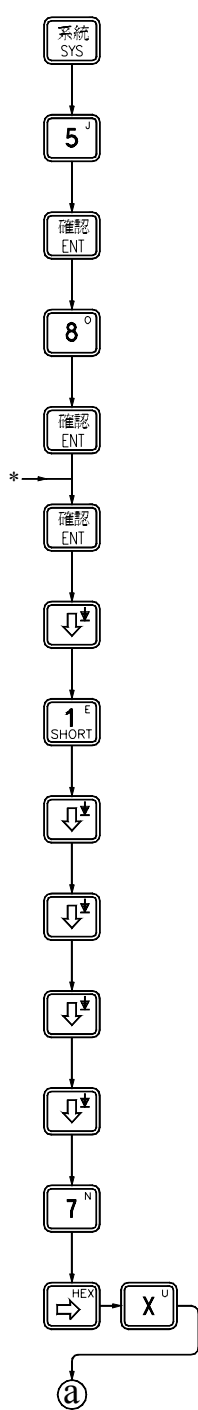
④ 選擇 HHSC 之計數輸入 (U, D, P, R, A, B) 倒相與否, 以匹配 Encoder 極性 (0: 不倒相, 1: 倒相; 內定值為 0)。

⑤ 選擇 HHSC 之控制輸入 (M, C) 倒相與否, 以匹配 Encoder 極性 (0: 不倒相, 1: 倒相; 內定值為 0)。

以 FP-07C 作上述 ①~⑤ 項建構之範例

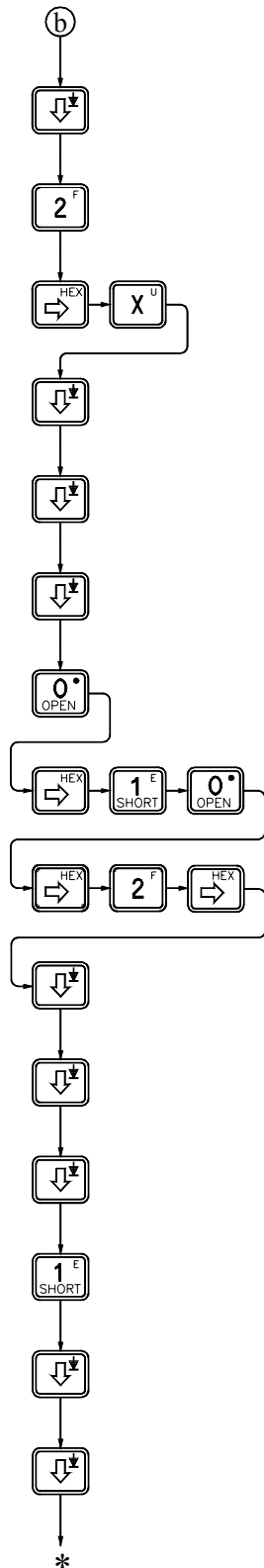
【按鍵操作】

【LCD 畫面顯示】

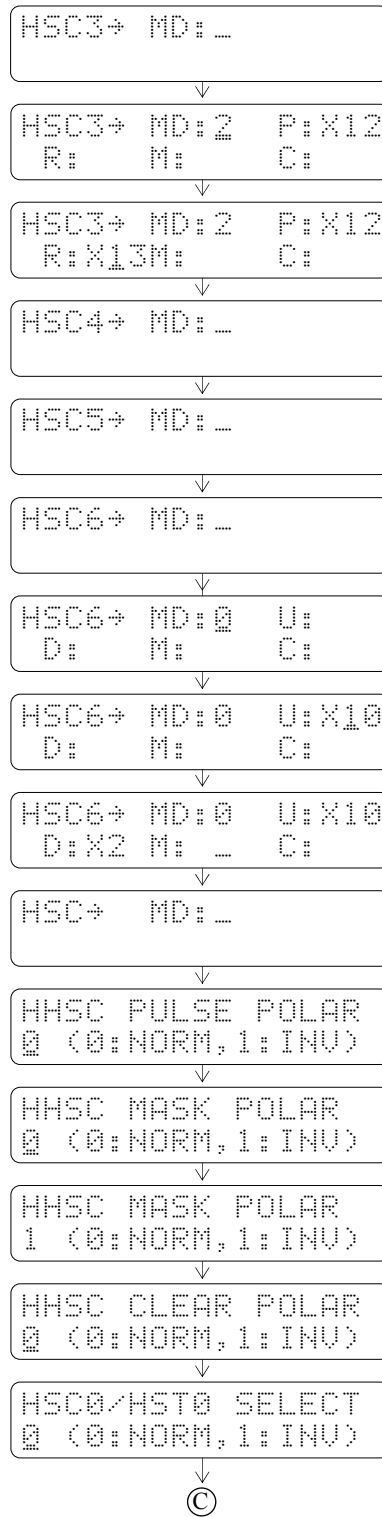


- HSC0 內定為 HSC
- HSC1 內定為 HSC
- 將 HSC1 改為高速計時器 HST1 使用
- HSC2 內定為 HSC
- HSC3 內定為 HSC
- MD 欄為空格, 表 HSC0 不用
- 因 HSC1 改為 HST 故無需再規劃 (不顯示)
- HSC2 之 MD 欄空格表不用
- \* 鍵入 7 後, 自動顯示 MD7 之各輸入名稱並自動補上 A、B 之內定值 (X8、X9)
- 欲使用 MASK 只需鍵入 X, 自動補號碼 10

【按鍵操作】



【LCD 畫面顯示】

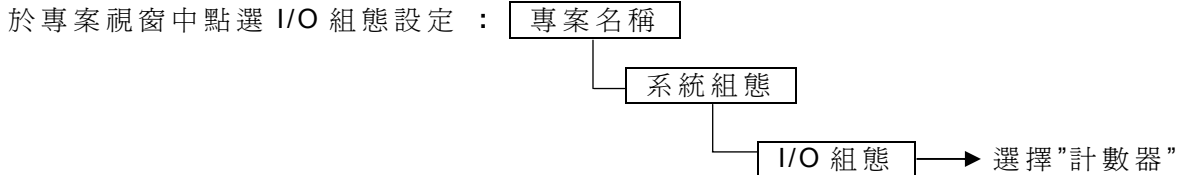


- 鍵入 2 後，自動填上 P 之內定號碼
- 游標移到 R 處只需鍵入“X”，立即改為 X13，將之改為硬體方向控制
- HSC4 不使用
- HSC5 不使用
- HSC6 設為 MD0，自動顯示 MD0 SHSC 之各輸入名稱
- 將 X10 指定為 HSC6 之上數脈波輸入“U”
- 將 X2 指定為 HSC6 之下數脈波“D”
- HSC7 不使用
- 所有 HHSC (HSC0~HSC3) 之所有計數輸入內定不倒相
- 所有 HHSC 之遮沒控制內定不倒相
- 將所有 HHSC 之 MASK 輸入改為倒相 (即 MASK 功能變 Enable)
- 所有 HHSC 之 CLEAR 控制輸入內定不倒相
- 完成整個 Configuration 循環回到最初畫面 (HSC0/HST0 選項)

- 修改輸入值可直接輸入新值覆蓋之，欲刪除輸入值則用 鍵。
- 任何欄位為空格 (不輸入任何值)，則表示不使用該 HSC 或該輸入。
- 上例中“PULSE”表 HHSC 之各種“計數輸入”，即 U，D 或 P，R 或 A，B。
- “POLAR”表極性“POLARITY”，亦即倒相或不倒相之選擇。

- HHSC 之各計數輸入及控制輸入之輸入點號碼均為固定，故在上例 Configuration 範例中，對 HHSC 之各輸入僅需鍵入 “X” 字母表示欲使用該輸入即可，FP-07C 或 WinProladder 會自動補上內定之 X 號碼，同時不容許更改。而 SHSC 之各計數或控制輸入則需由使用者在 X0~X15 間自由指定，因此 SHSC 之輸入點號碼必須鍵入 “X” 及號碼 n 始為完整。

#### 10.4.2 以 WinProladder 作 HSC/HST 之建構



- 出現計數器設定畫面後，可直接於視窗中選擇欲計數之訊號及輸入接點……等。



[計數器設定]區域：

- **使用模式** 欄位：可選擇硬體計數器或硬體計時器。
- **計數模式** 欄位：可選擇欲計數之工作模式，有 U/D、P/R、A/B……等。
- **上數(UP)** 欄位：選擇上數訊號之輸入，若計數模式為 P/R，則此欄位為 **脈波(PS)**；若計數模式為 A/B，則此欄位為 **A 相**。
- **下數(DN)** 欄位：選擇下數訊號之輸入，若計數模式為 P/R，則此欄位為 **方向(DIR)**；若計數模式為 A/B，則此欄位為 **B 相**。
- **遮沒(MSK)** 欄位：選擇遮沒訊號之輸入。
- **清除(CLR)** 欄位：選擇清除訊號之輸入。

[HSC 極性設定]區域：

- **遮沒訊號** 欄位：決定遮沒訊號為正相或是倒相。
- **清除訊號** 欄位：決定清除訊號為正相或是倒相。
- **計數訊號** 欄位：決定計數訊號為正相或是倒相。

[HSC 內容長度設定]區域：

- 可以選擇的模式有以雙字元組(32-Bit 硬體計數器)來存放計數值或 1 字元組存放計數值，1 字元組當做循環計數器(16-Bit Timer+16-Bit Counter)兩種。客戶可依自己的需求應用之。
- 以下將 FBs-PLC 之 HHSC 與 SHSC 所內定或可選擇之輸入點號碼、軟體遮沒、清除、方向選擇等相關號碼彙整如下表。

機 種		MC/MN								MA
容許 之號碼	類別	HHSC				SHSC				SHSC
		HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	HSC6	HSC7	HSC4 ~ HSC7
HSC 相關信號		HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	HSC6	HSC7	HSC4 ~ HSC7
CV 暫存器號碼		DR4096	DR4100	DR4104	DR4108	DR4112	DR4116	DR4120	DR4124	同 MC/MN 之 SHSC
PV 暫存器號碼		DR4098	DR4102	DR4106	DR4110	DR4114	DR4118	DR4122	DR4126	同 MC/MN 之 SHSC
計數 輸入	U 或 P 或 A	X0	X1/X4	X4/X5/X8	X5/X12	X0~X15	X0~X15	X0~X15	X0~X15	同 MC/MN 之 SHSC
	D 或 R 或 B	X1	X5	X9	X13	X0~X15*	X0~X15*	X0~X15*	X0~X15*	同 MC/MN 之 SHSC
控制 輸入	M	X2	X6	X10	X14	X0~X15	X0~X15	X0~X15	X0~X15	同 MC/MN 之 SHSC
	C	X3	X7	X11	X15	X0~X15	X0~X15	X0~X15	X0~X15	同 MC/MN 之 SHSC
軟體遮沒繼電器		M1940	M1946	M1976	M1979	M1982	M1984	M1986	M1988	同 MC/MN 之 SHSC
軟體清除繼電器		M1941	M1947	M1977	M1980	直接清除 CV 暫存器即可				同左所述
軟體方向選擇 (僅 MD2, 3)		M1942	M1948	M1978	M1981	M1983	M1985	M1987	M1989	同 MC/MN 之 SHSC
中斷副程式 標記名稱		HSC0I	HSC1I	HSC2I	HSC3I	HSC4I	HSC5I	HSC6I	HSC7I	同 MC/MN 之 SHSC

\*：SHSC 工作於 MD2(P/R)時，方向選擇由特殊繼電器 M1983、M1985、M1987、M1989 來決定。

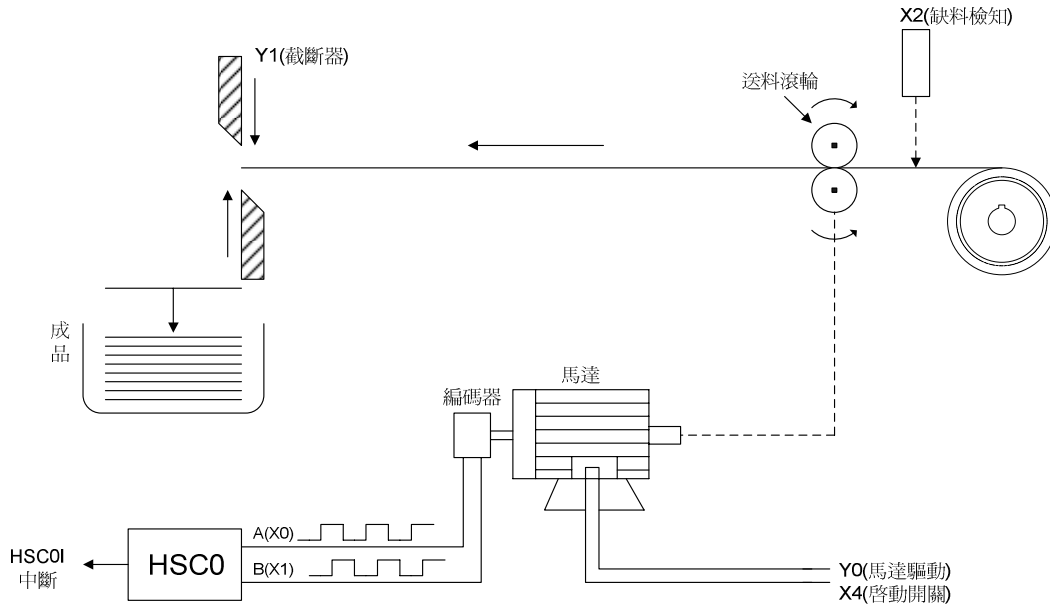
- 當工作於雙向模式時(HHSC 為 MD4~MD7、SHSC 為 MD4)，輸入點必須以連續號碼配對使用，並且以偶數號碼為 A 相輸入(例 X4)，奇數號碼為 B 相輸入(例 X5)。

- 上表中之 X0~X15 輸入點只能被指定一次（亦即只能當作一種功能），不能重覆使用。
- FBs MN 之 HHSC 最高計數頻率單／雙相最高可達到 920KHz。
- FBs MC 機種之 HHSC 計數頻率單／雙相最高可達到 120KHz。
- 軟體高速計數器之輸入頻率總和不得大於 8KHz，頻率愈高愈佔系統時間，掃描時間會暴增！
- MA 機種只提供軟體高速計數器。

## 10.5 高速計數器之應用範例

範例 1 · 係利用高速計數器作等長度之截斷控制

機構



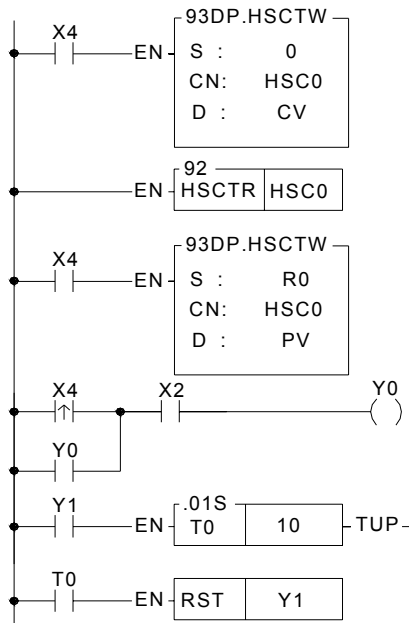
HSC 建構 ( Configuration ) ( 將 HSC0 設為 MD7 以提高截斷解析度 4 倍 )





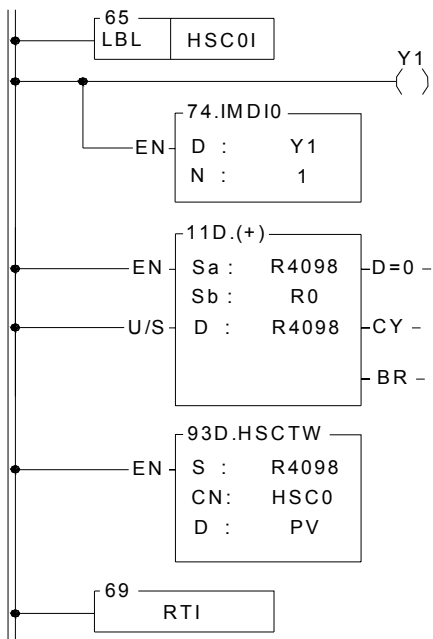
## 控制程式

### 【主程式】



- 利用 FUN93 將 SoC 晶片內 HSC0 之 CV 暫存器及目前值暫存器 DR4096 清為 0  
CN=0 表 HSC0  
D=0 表 CV
- 利用 FUN92 將 SoC 晶片內 HSC0 之 CV 暫存器之計數值讀出 (存入 DR4096)
- 將計數行程 DR0 利用 FUN93 將其值寫入 SoC 晶片內 HSC0 之 PV 暫存器及寫入 DR4098 設定值暫存器  
CN=0 表 HSC0  
D=1 表 PV
- 起動馬達
- 截斷器 Y1 ON 0.1 秒

### 【副程式】



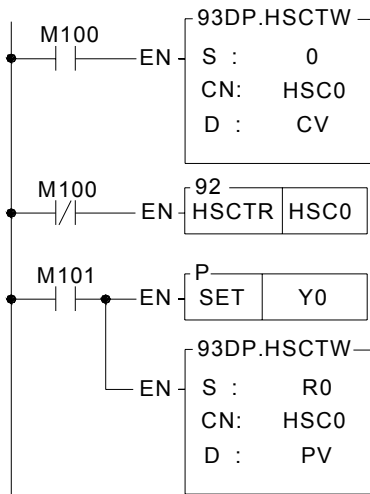
- 當 SoC 晶片內 HSC0 之 CV=PV 時，硬體自動執行此標名為 HSC0I 之中斷副程式
- 計數到時，將 Y1 ON (截斷物料)
- 將 Y1 立即輸出，才能減少因掃描時間所引起之誤差
- 計算出新的行程，並載入 HSC0 之 PV

### 【說明】

- 1.主程式先將 HSC0 CV 初始化 (CV 值為 0)，並將截斷長度值 (DR0) 搬入 HSC0 之 PV 後啟動 Y0，馬達轉動開始送料。
- 2.當 CV 值達 PV 值時，啟動裁刀並將 PV 值再加長度值 DR0 重新載入 HSC0 之 PV 去後返回。
- 3.當物料捲完時，缺料檢知 X2 ON，馬達停止。

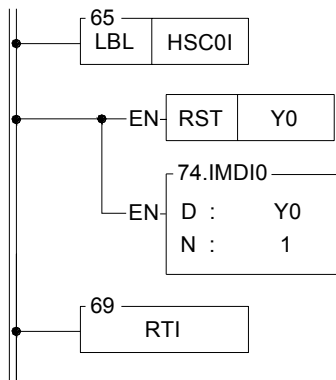
範例 2 · 固定行程高速計數到中斷立即處理範例

【主程式】



- 當 M100 由 0→1 時，利用 FUN93 將 SoC 晶片內 HSC0 本體之 CV 以及目前值暫存器 DR4096 歸零  
CN=0，代表 HSC0  
D =0，代表 CV
- 利用 FUN92 將 SoC 晶片內 HSC0 之目前計數值讀出，並存入目前值暫存器 DR4096  
CN=0，代表 HSC0
- 當 M101 由 0→1 時，啓動 Y0 ON (開始運轉)
- 利用 FUN93 將設定值暫存器(DR0)之內容寫入 SoC 晶片內 HSC0 之 PV，當作計數到中斷設定值。  
CN=0，代表 HSC0  
D =1，代表 PV

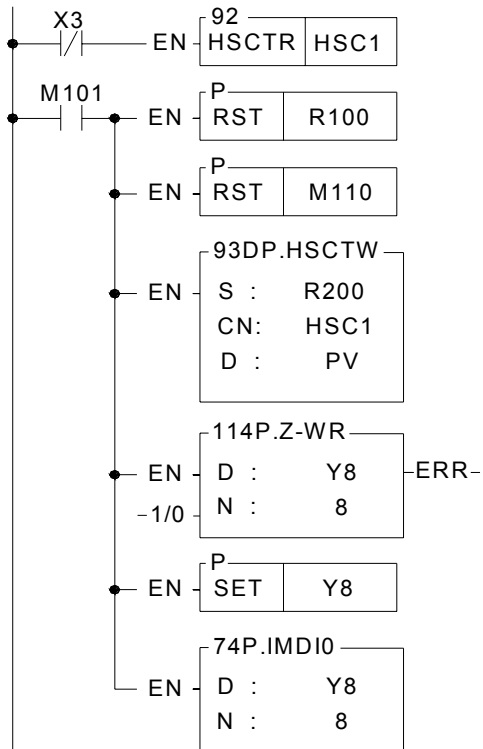
【副程式】



- 標名為 HSC0I 之硬體高速計數器中斷服務副程式
- 計數到時，將 Y0 OFF (停止運轉)
- 將 Y0 立即輸出，才能立即停止運轉 (否則 Y0 會有掃描時間之輸出延遲)

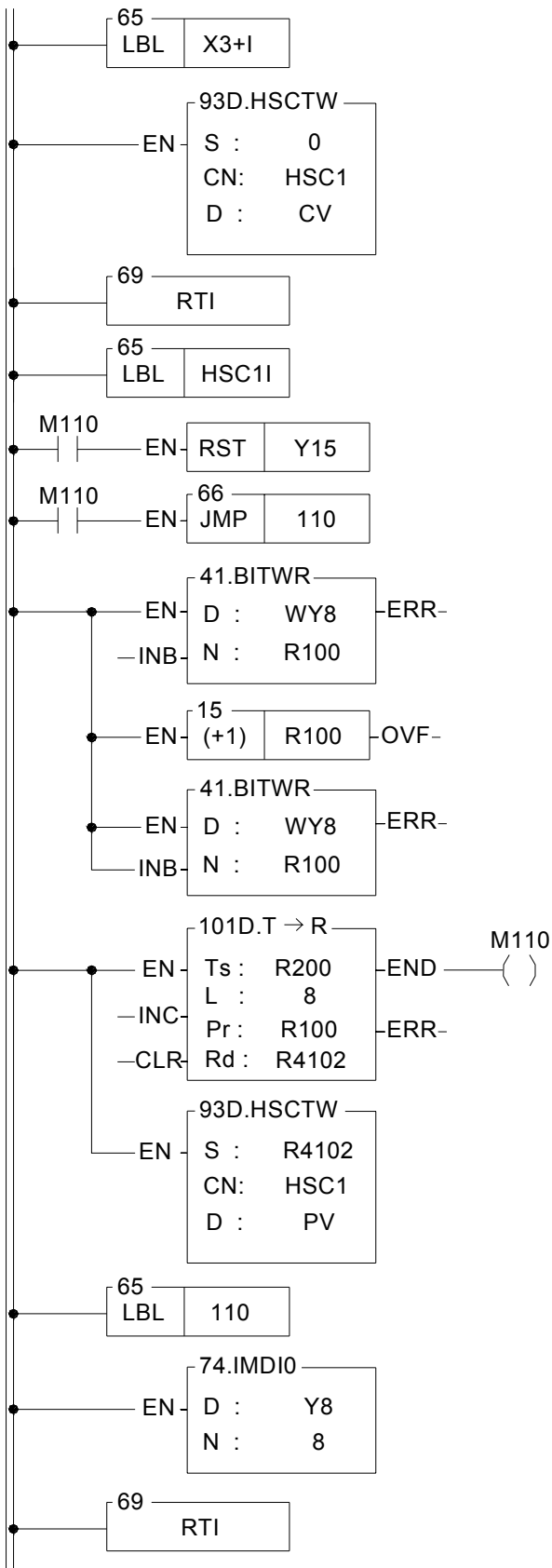
範例 3 · 多段行程高速計數到中斷立即處理範例

【主程式】



- 利用 FUN92 將 SoC 晶片內 HSC1 之目前值讀出，並存入目前值暫存器 DR4100  
CN=1，代表 HSC1
- 當 M101 由 0→1 時，將指標暫存器清除為 0
- 將最後一段旗標清除為 OFF
- 利用 FUN93 將計數行程 DR200（第 0 段）之內容寫入 SoC 晶片內 HSC1 之 PV，當作計數到中斷設定值  
CN=1，代表 HSC1  
D =1，代表 PV
- 將 Y8~Y15 清除為 OFF
- 設定 Y8 ON，表示目前正位於第 0 段
- Y8~Y15 立即輸出

【副程式】



- 標名為 X3+I 之 X3 正緣中斷服務副程式（需指定 X3 為正緣中斷輸入）
- 當 X3 由 0→1 時，利用 FUN93 將 SoC 晶片內 HSC1 之 CV 以及目前值暫存器 DR4096 歸零  
CN=1，代表 HSC1  
D=1，代表 CV
- 標名為 HSC1I 之硬體高速計數器中斷處理副程式
- 最後一段結束時，將 Y15 OFF
- 將前一段輸出 OFF
- 將指標指到下一段
- 將下一段輸出 ON
- 將下一段之計數行程（DR200 開始，指標指到之暫存器）搬至設定值暫存器 DR4102
- 最後一段時 M110 ON
- 利用 FUN93 將設定值暫存器之內容寫入 SoC 晶片內 HSC1 之 PV，當作計數到中斷設定值  
CN=1，代表 HSC1  
D =1，代表 PV
- 將 Y8~Y15 立即輸出

## 10.6 FBs-PLC 之高速計時器

一般 PLC 計時器之計時單位（時基）最小只能達 1mS，且需加上掃描時間之誤差，因此對需要較精密之計時（如利用計時器配合 HSC 作成頻率計）便無法達成，必須使用高速計時器（High-Speed Timer 簡稱 HST）才能勝任。

FBs-PLC 內建有一個 16 位元/0.1mS 時基（Time base）之高速計時器（HSTA）。另外，如前述 HHSC 之 4 個 32 位元高速計數器（HSC0~HSC3）均可轉化為 32 位元/0.1ms 時基之高速計時器（HST0~HST3）使用，因此 FBs-PLC 最多可有 5 個高速計時器。如同 HSC 及 INT，所有 HST 均可以 EN（FUN145）及 DIS（FUN146）指令將其開啓或關閉（內定爲 EN 開啓）。下面就 HSTA 與 HST0~HST3 分別敘述如下。

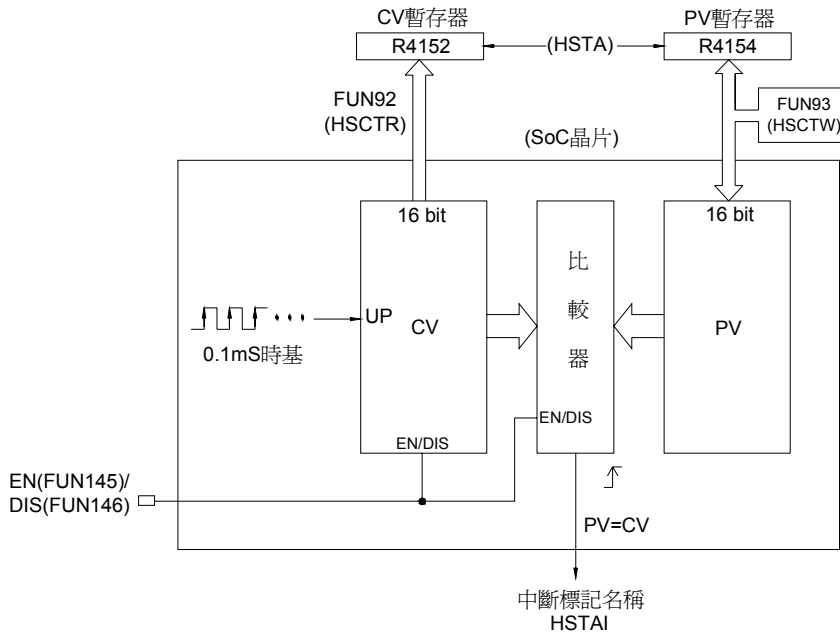
大部份 PLC 計時器之最小時基爲 10mS，即使有部份 PLC 能提供 1mS 時基之計時器，也因 PLC 掃描時間之誤差，使其 1mS 之時基失去意義（例如雖時間爲 1mS，但若掃描時間爲 10mS，總誤差仍超過 10mS），因此無法應用在需要高精度之計時應用。FBs-PLC 之時基爲 0.1mS，且其 Time up（計時到）係以中斷發出，因此無掃描時間誤差，較之一般 PLC 之計時器，其精度提高 100 倍，可使用許多需精密計時之應用。

### 10.6.1 HSTA 高速計時器

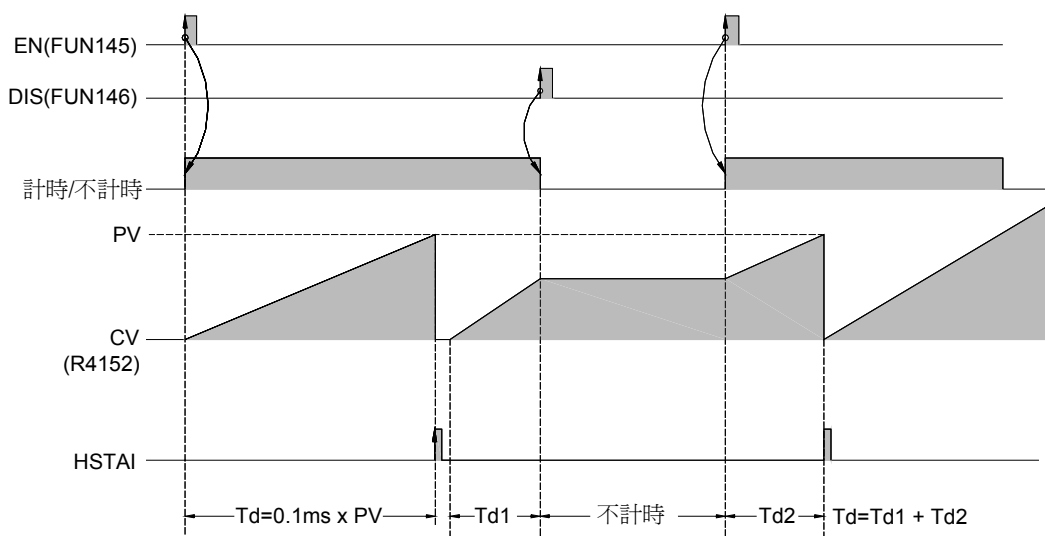
HSTA 是內建於 SoC 晶片內之 16 位元硬體計時器，因此如同 HHSC 一般必須利用 FUN93（HSCTW）指令將計時設定值寫入到晶片內 HSTA 之 PV 去，而 CV 之讀出則使用 FUN92（HSCTR）。HSTA 可當成兩種功能不同之計時器，當其 PV 值 $\geq 2$ 時，FBs-PLC 會將 HSTA 當作一般標準功能之 16 位元延遲計時器（Delay Timer）；當 PV 值=0 時，則將 HSTA 當作 32 位元之循環計時器（Cyclic Timer）。

## A. HSTA 16 位元高速延遲計時器 (定時中斷計時器)

延遲計時器是在 HSTA 計時開始起，延遲  $PV \times 0.1\text{ms}$  之時間後發出中斷。因 HSTA 當延遲計時器時為 16 位元，故其 PV 值可設為 0002H~FFFFH (無正負號之 65535)，亦即可設之延遲時間為 0.2ms~6.5535 秒，其功能使用方法和一般延遲計時器一樣，差異只是 HSTA 之時基較精細，且計時到 (Time-Up) 時，將立即發出中斷，計時精度相差甚大。下圖為 HSTA 當作延遲計時器時之結構圖。詳細功能及使用方法請參考 10.6.3 小節之程式範例。



- 利用 FUN93 將設定值寫入 SoC 晶片內之 HSTA 高速計時器設定值暫存器 R4154，作為定時中斷計時器 (每隔設定值時間，執行標名為 "HSTAI" 之定時中斷副程式)。

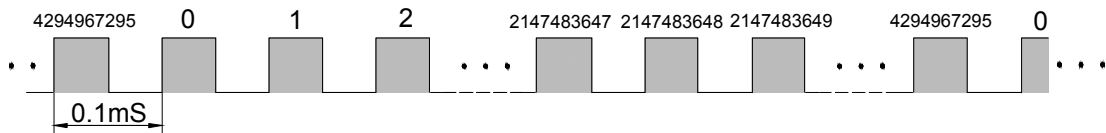
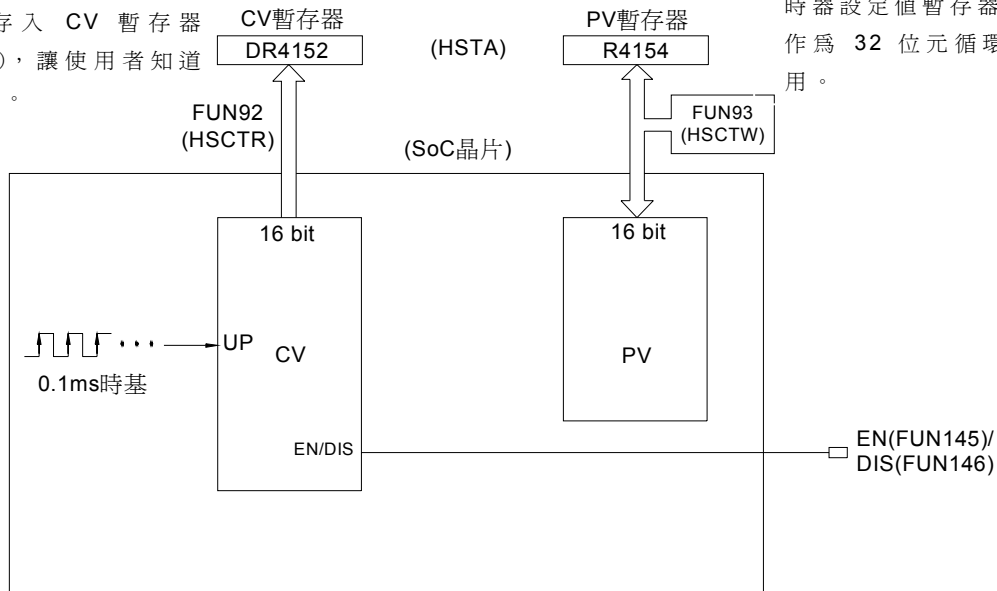


## B. HSTA 32 位元循環計時器

所謂循環計時器是該計時器每間隔固定時間即將目前計時值加 1，而永遠不停地上數循環計時。其 CV 值由 0, 1, 2……, 2147483647, 2147483648, 2147483649, …… 4294967295, 0, 1, 2……週而復始地循環（因時基為 0.1mS，CV 值×0.1mS 即為其累計之時間值）。實質上，循環計時器為一恆久運轉之 0.1mS 時基之上數循環計時時鐘，可供任兩事件（Event）發生時讀取，而求得兩事件發生之間隔時間。下圖 B 為 HSTA 當作 32 位元循環計時器時之結構圖，如圖示，循環計時器 PV=0 時，不會發出中斷，欲得知計時值必須利用 FUN92 將之自 SoC 晶片讀取 CV 值再存放到 PLC 內部之 32 位元 CV 暫存器（DR4152）中。循環計時器典型之應用是可作較準確之轉速偵測，在轉速變化極大或極低場合下作轉速偵測，請參考第 10.6.3 小節範例說明。

- 利用 FUN92 將 SoC 晶片內之 HSTA 高速計時器目前計時值讀出並存入 CV 暫存器（DR4152），讓使用者知道計時經過值。

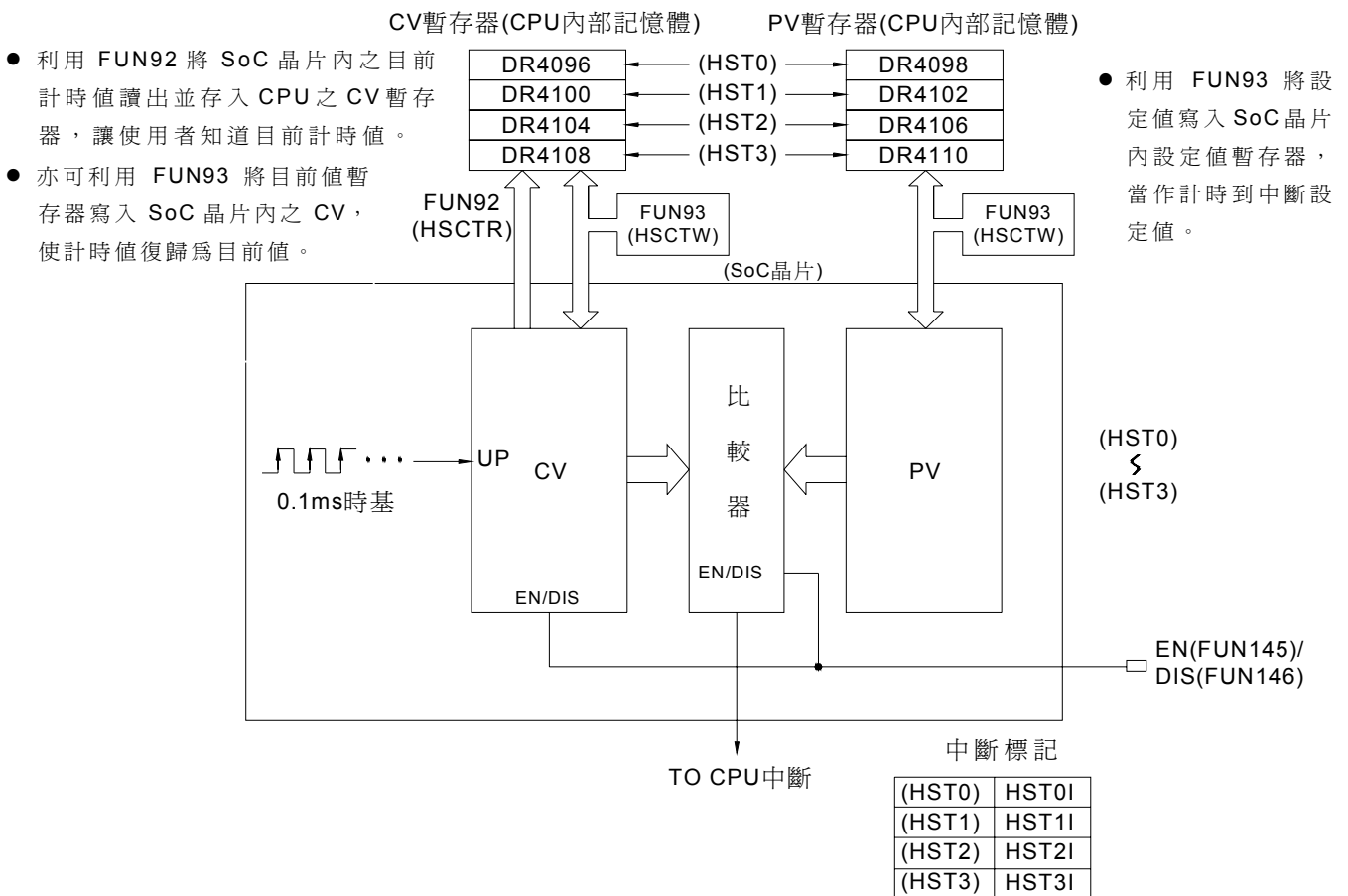
- 利用 FUN93 將設定值=0 寫入 SoC 晶片內之 HSTA 高速計時器設定值暫存器 R4154，作為 32 位元循環計時器使用。



## 10.6.2 HST0~HST3 高速計時器

### A.HST0~HST3 高速延遲計時器 (定時中斷時器)

HHSC (HSC0~HSC3) 可將之規劃為 HST0~HST3 4 個 32 位元之高速延遲計時器，其功能及時基和 16 位元之 HSTA 延遲計時器完全一樣，差異僅在 HST0~HST3 為 32 位元。將 HHSC 規劃為 HST 僅需在 WinProladder 或 FP-07C 之系統模式下之“Configuration”功能下之第 8 項“HSC/HST/INT”指定項下之 HSC/HST 選項中選擇“1”即完成，請參考 10.4 節“HSC/HST 之 Configuration”之範例（在該範例中係將 HSC1 建構為 HST1）。下圖為 HHSC 規劃為 HST 高速延遲計時器之功能結構圖。其使用方法同 HSTA 16bit 高速延遲計時器，請參閱 10.6.4 小節之程式範例。



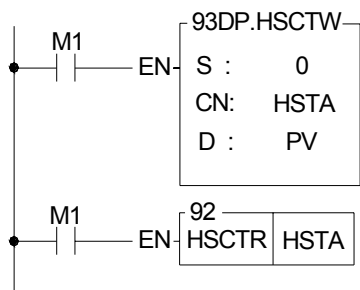
### B.HST0~HST3 32 位元循環計時器

將 HHSC (HSC0~HSC3) 視需要規劃為 HST0~HST3 等 32 位元計時器。每間隔 0.1mS，SoC 晶片內之目前計時值暫存器會加 1；使用者可利用 FUN92 指令將 SoC 晶片內之目前計時值讀出並存入 CPU 之 CV 暫存器 (DR4096、DR4100、DR4104、DR4108)。因此 CPU 之 CV 暫存器內容即為 0, 1, 2, ……7FFFFFFFH, 80000000H, ……FFFFFFFFH, 0, 1, ……等 32 位元之變化值。利用二事件間之間隔時間計算技巧，可得到無限多個 0.1mS 之 32 位元計時器。



### 10.6.3 高速計時器 HSTA 之使用範例

#### 範例 1 · HSTA 當作 32 位元循環計時器



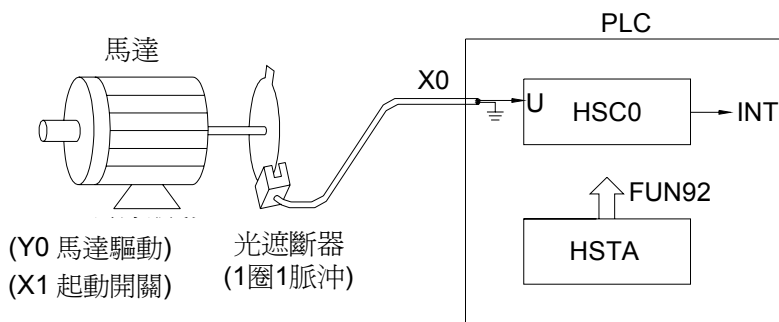
- 利用 FUN93 將 SoC 晶片內 HSTA 之 PV 設定為 0 (當作循環計數器用)  
CN=4, 代表 HSTA  
D=1, 代表 PV
- 利用 FUN 92 將 SoC 晶片內 HSTA 之目前計時值讀出並存入 DR4152 (DR4152 之值由 0, 1, 2, ... .., FFFFFFFF, 0, 1, 2, ... .. 循環變化, 單位為 0.1mS)
- CN=4, 代表 HSTA

#### 範例 2 · 循環計時器之應用例

本例以 HSTA 當作循環計時器，配合 HSC0，以每累積 10 個脈沖發出一次中斷，讀取累積該 10 次脈沖所歷經之時間，而反向地求出 RPM (脈沖數固定，時間變化)。

註：本例適合高速 RPM 應用 (300~6000RPM)，不適合低速 RPM (低速時，RPM 值更新太慢)。

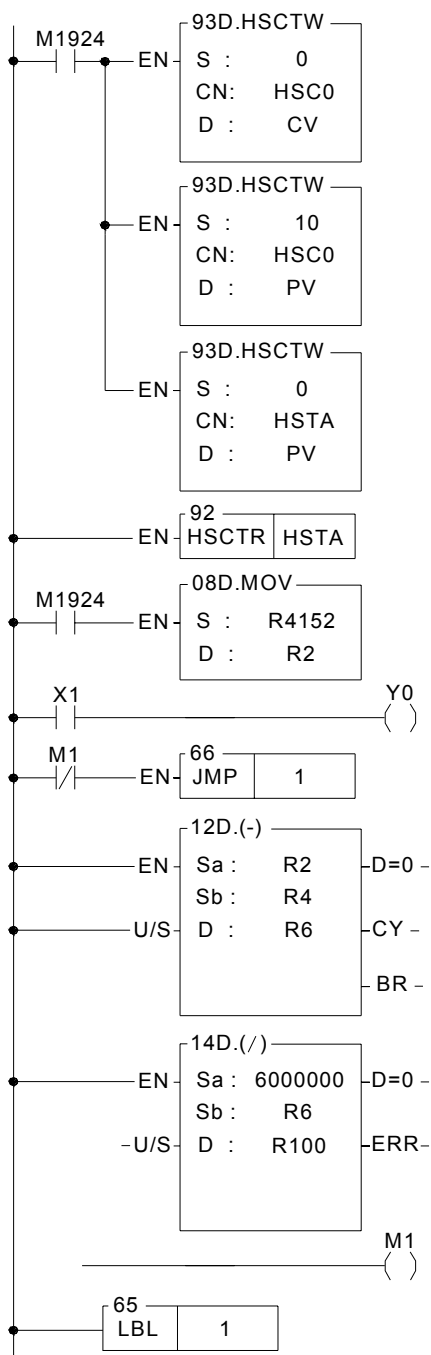
#### 機構



#### HSC 與 HST 之建構

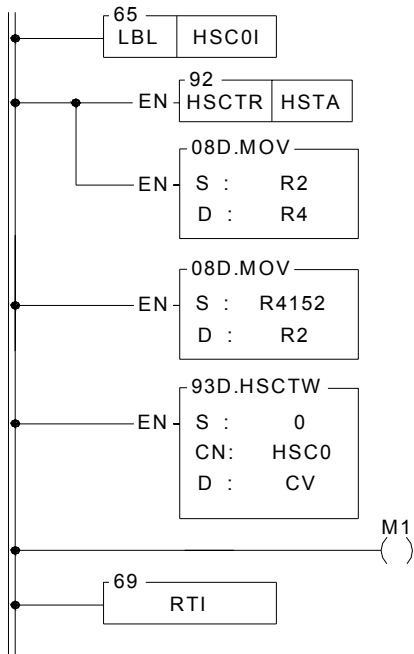
- ① 因 HSTA 為內建，無需任何建構，只要寫入 PV = 0 即可將之變成 32 位元循環計時器。
- ② 配合光遮斷器，將 HSC0 設為單輸入之上數計數器 (MD0, 但只使用 U 輸入) 其他設定 (計數輸入與控制輸入極性) 均為內定 (不倒相)，不必更動。

【主程式】

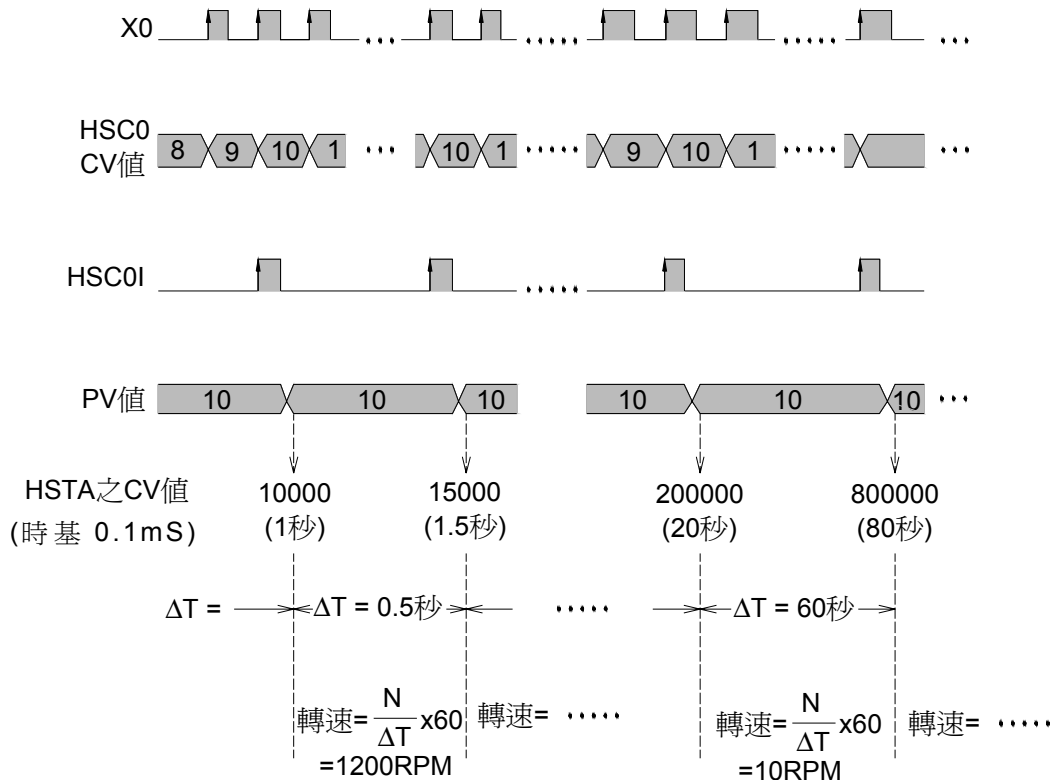


- 利用 FUN93 將 SoC 晶片內之目前值暫存器計數歸零；  
FUN93 之 CN=0 表 HSC0；D=0 表 CV
- 利用 FUN93 將 10 寫入 SoC 晶片內之設定值暫存器，當作計數到中斷值；  
FUN93 之 CN=0 表 HSC0；D=1 表 PV
- 將 0 利用 FUN93 寫入 SoC 晶片內之設定值暫存器，將 HSTA 規劃為 32 位元循環計時器；  
FUN93 之 CN=4 表 HSTA；D=1 表 PV
- 讀取目前計時值並存放至 DR4152
- HSTA 之 CV 暫存器初值存入 DR2
- 求出 HSC0 每次中斷的間隔時間  
 $\Delta T$  (DR6×0.1mS)
- 轉速 =  $\frac{N}{\Delta T} \times 60$  RPM  
 $N=10, \Delta T = \Delta CV \times 0.1mS = \frac{(此次CV - 前次CV)}{10000}$  秒  
，故轉速 =  $\frac{6000000}{\Delta CV}$  RPM
- R100=RPM
- 清除 RPM 計算旗標(將 M1 清除為 0)

【副程式】

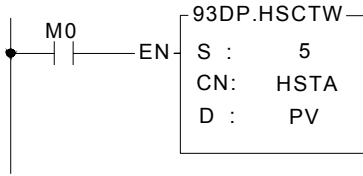


- HSC0 每累計 10 個脈沖，硬體即自動執行此中斷副程式
- 讀取 HSTA 之 CV 值
- 將目前計數值復歸為 0
- M1=ON，RPM 計算旗標



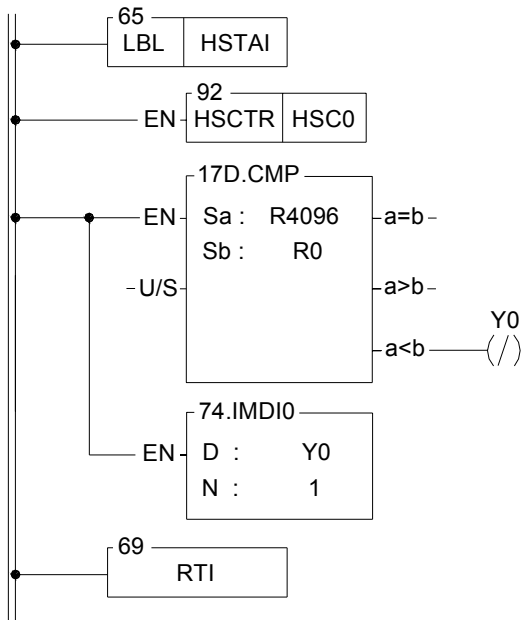
範例 3 · HSTA 當作定時中斷計時器程式

【主程式】



- 設定定時中斷之時間；S=5 代表每 0.5mS 執行標名為 HSTAI 之定時中斷服務副程式
- 利用 FUN93 將設定值寫入 SoC 晶片內 HSTA 之 PV，當作計時到中斷設定值 CN=4，代表 HSTA  
D =1，代表 PV

【副程式】



- 標名為 HSTA 之定時中斷服務副程式
- 每 0.5mS 讀取硬體高速計數器 HSC0 之目前計數值，並存放於 DR4096
- 判斷目前計數值是否大於等於 DR0，如是，則 Y0 ON
- 將 Y0 立即輸出，才能達到高速輸出反應（否則 Y0 會有掃描時間之延遲）

## 10.6.4 HST0~HST3 高速計時器使用範例

### HSC 與 HST 之建構(使用 WinProladder)

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"計數器"

- 出現計數器設定畫面後，於「使用模式」欄位點選「硬體計時器」選項即可將 HHSC(硬體高速計數器)建構為 HHT(硬體高速延遲計時器)。
- 使用者並不需要去建構 HSTA，因為 HSTA 本身即為內建的。只有要將 HHSC(硬體高速計數器)拿來當做 HHT(硬體高速延遲計時器)使用時，才需以上述方式建構之。



### HSC 與 HST 之建構(使用 FP-07C)

```
HSC0/HST0 SELECT
1 (0:HSC,1:HST)
```

- 將 HSC0 設為 HST0

```
HSC1/HST SELECT
0 (0:HSC,1:HST)
```

- HSC1 內定為 HSC

```
HSC1 → MD:0 U:X4
D: M: C:
```

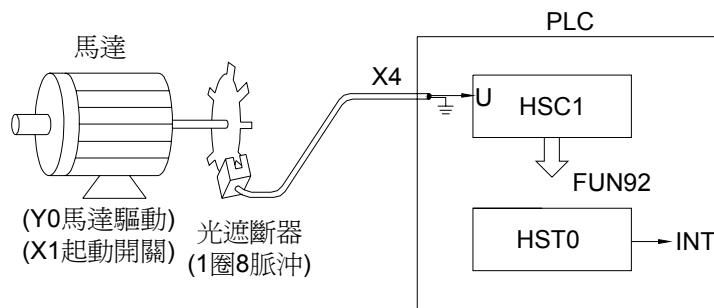
- 將 HSC1 設為 MD0，為單輸入上數計數器；其他輸入不用

- 其他設定（計數輸入，控制輸入極性）均為內定（不倒相），不必更動。

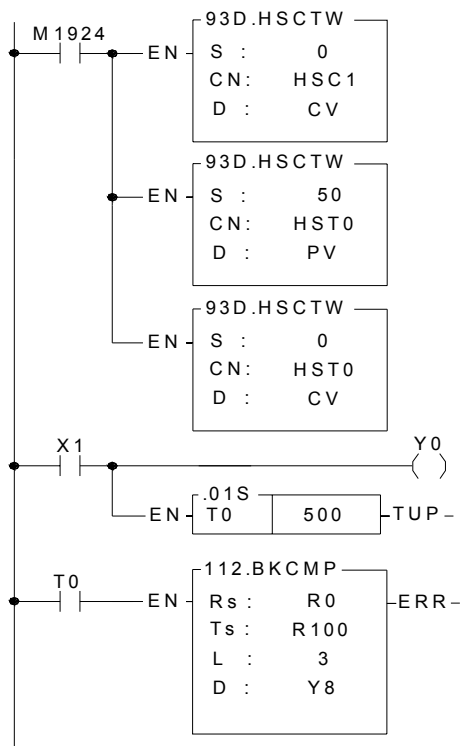
### 範例 1 · 延遲計時器之應用例

本例將 HSC0 高速計數器規劃為 HST0 延遲計時器，同時以 HSC1 高速計數器連接自動木工鑽床之轉動馬達以其旋轉圈數，於每一固定時間週期即發出中斷，讀取在此週期內所計數到之馬達回轉圈數，以未加載（空轉未作鑽孔動作）時之轉數，比較當鑽頭下壓（鑽孔）時之速度變化，而能得知鑽頭狀況；因馬達在鑽頭正常（銳利）時之阻力較小，轉速適中，而在鑽頭變鈍時，阻力較大，轉速變慢，而在鑽頭折斷時阻力無，轉速將同空轉速度，是為最快，因快、中、慢速之差異通常不大，若以一般計時器取樣檢知，因誤差高達數十 mS，光是誤差值就超過快、中、慢速之差異，根本無法判知速度變化，但以 0.1mS 高速計時器配合中斷，即能以相當低之成本，達到能檢知鑽頭正常、變鈍或折斷，而能及換頭警示或停機更換之動作。【時間固定，脈沖數變化】

### 機構

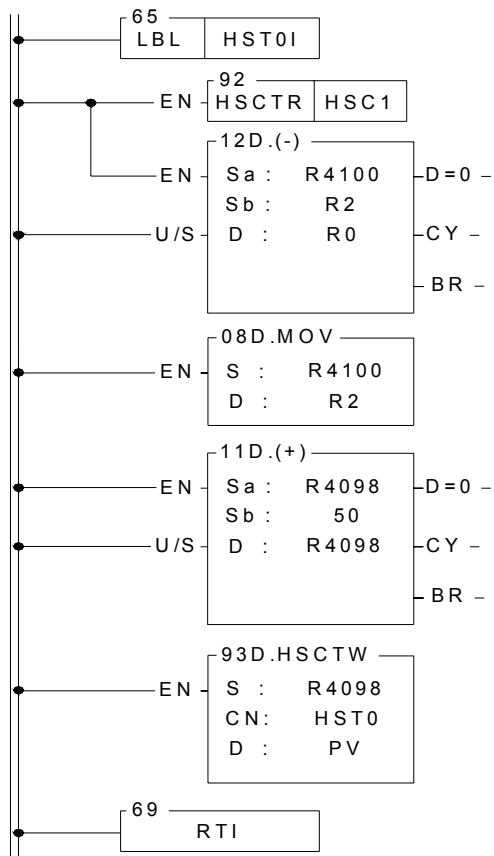


### 【主程式】



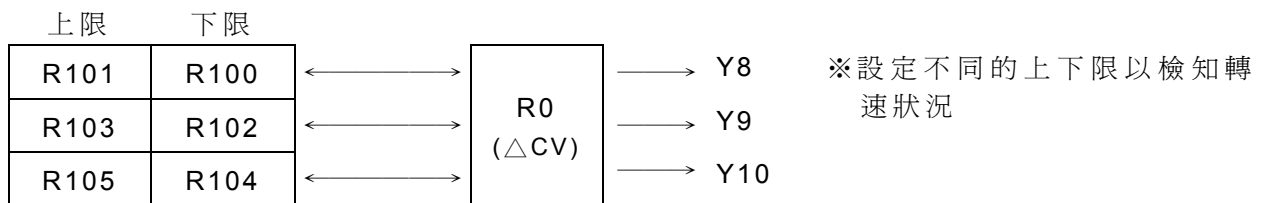
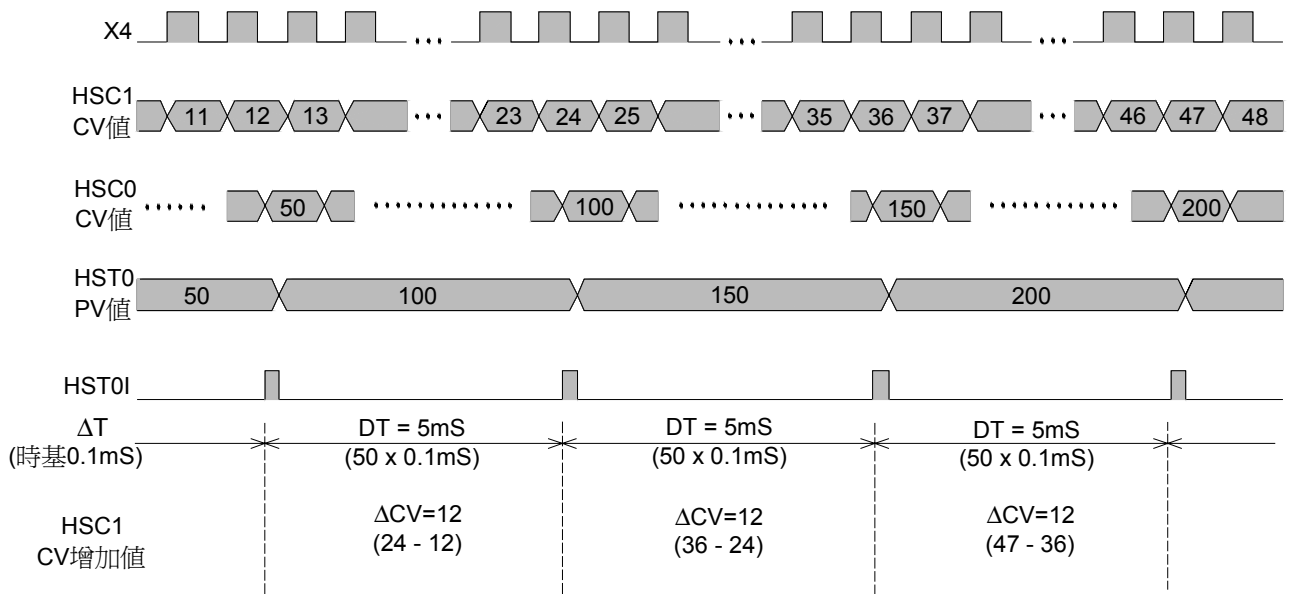
- 利用 FUN93 將 SoC 晶片內之目前值暫存器清除為 0  
FUN93 之 CN=1 表 HSC1；D=0 表 CV
- HST0 之 PV 值設為 50，即每隔 5mS (50×0.1mS) 中斷一次
- HST0 之 CV 暫存器初值為 0
- 起動馬達 5 秒後才利用 FUN112 比較鑽頭轉速狀況  
R0：每隔 5mS 取得 HST1 之脈波數

【副程式】

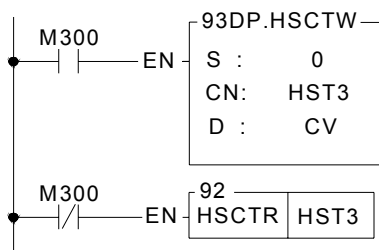


- 每隔 5mS 硬體即自動執行一次此副程式
- 讀取 HSC1 之目前計數值並存放於 DR4100
- 求取此次 5mS 內 HSC1 CV 之增加值，並存入 DR0（在實用上 R1=0）
- 計算 HST0 之新 PV 值

【說明】假設正常時，鑽頭轉速為 18000RPM，而 1 轉會使光遮斷器產生 8 個脈沖，則 HSC1 之 U 腳頻率= $18000/60 \times 8 = 2400\text{Hz}$ ，亦即 5mS 內會有 12 個脈沖。因此可利用 HST0 固定每隔 5mS 中斷一次，而讀取 HSC1 之 CV 值，便可知道轉速狀況。



**範例 2 · 硬體高速計時器 HST3 當作 32 位元循環計時器**

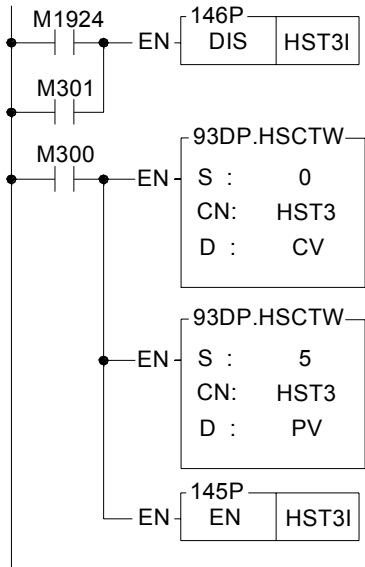


- 當 M300 由 0→1 時，利用 FUN 93 將 SoC 晶片內 HST3 之 CV (歸零)  
CN=3，代表 HST3  
D=0，代表 CV
- 利用 FUN 92 將 SoC 晶片內 HST3 之目前計時值讀出並存入目前值暫存器 DR4108  
(DR4108 之值由 0, 1, 2, ... , FFFFFFFF, 0, 1, 2, ... 循環變化，單位為 0.1mS)  
CN=3，代表 HST3



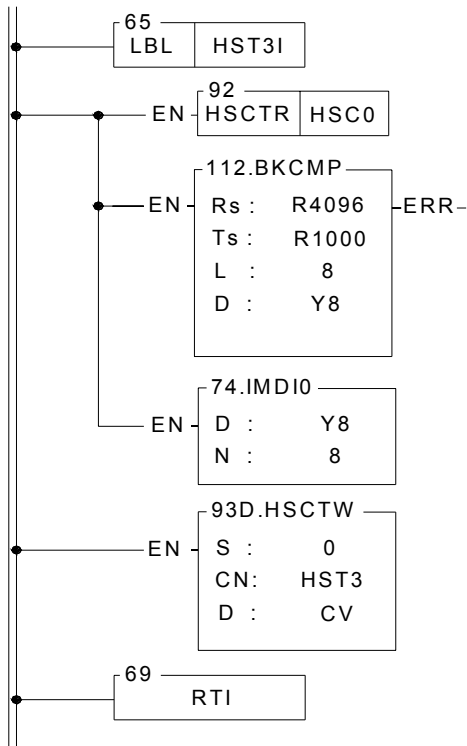
範例 3 · 硬體高速計時器 HST3 當作定時中斷計時器程式

【主程式】



- 開機或 M301 ON 時，禁止 HST3 發出定時中斷
- 當 M300 由 0→1 時，利用 FUN93 將 SoC 晶片內 HST3 之 CV 歸零  
CN=3，代表 HST3；D=0，代表 CV
- 設定定時中斷之時間；S=5，代表每 0.5mS 執行標記名稱爲 HST3I 之定時中斷服務副程式
- 利用 FUN93 將設定值寫入 SoC 晶片內 HST3 之 PV，當作計時到中斷設定值  
CN=3，代表 HST3；D=1，代表 PV
- 啓動 HST3 定時中斷

【副程式】



- 標名爲 HST3I 之硬體高速計時器中斷服務副程式
- 每 0.5mS 讀取硬體高速計數器 HSC0 之目前計數值
- 判斷目前計數值落於電子凸輪之那一段，並將相對應之輸出點 ON
- 將 Y8~Y15 立即輸出
- 利用 FUN93 將 SoC 晶片內 HST3 之 CV 歸零  
CN=3，代表 HST3；D=0，代表 CV



# MEMO

