



台達電子工業股份有限公司
33068 桃園縣桃園市興隆路18號
TEL:886-3-3626301
FAX:886-3-3716301

*本使用手冊內容若有變更，恕不另行通知

DVP-4959600-11

ISPSoft 軟體使用手冊



ISPSoft 軟體使用手冊



I SPSoft 軟體使用手冊

版本修訂一覽表

版本	變更內容	發行日期
第一版	第一版發行	2012/01/06
第二版	第二版發行	2012/06/30
第三版	第三版發行	2012/10/31
第四版	增加附錄 D 及附錄 E	2015/03/18
第五版	<ol style="list-style-type: none">1.第 3 章加入 AHxxEMC 系列與 AS300 系列機種相關內容。2.新增章節：第 8 章自定義資料型態、第 9 章軸、第 15 章連續功能圖 CFC、第 21 章 G-Code 編輯器與 E-CAM 編輯器及第 22 章精靈工具。3.附錄 B 加入 AHxxEMC 系列與 AS300 系列機種相關內容。4.刪除附錄 D 及附錄 E。5.更新所有章節軟體畫面。	2016/05/18
第六版	<ol style="list-style-type: none">1.第 2 章加入 AS300 模擬器功能介紹。2.第 3 章 AHxxEMC 系列與 AS300 系列增加及修改功能。3.第 9 章軸參數增加及修改功能。4.第 14 章 SFC 增加折疊 Action 列表功能。5.第 16 章增加 DVP 轉 AS 功能及裝置註解新增功能。6.第 17 章線上監看模式顯示內容修改。7.第 21 章 ECAM 表新增檔案模式等多項功能及新增 Polynomial 函數功能。8.第 22 章定位規畫表軸設定畫面變更及各種軸輸出表格項目修改；資料示波器、資料記錄器介面修改及新增雙垂直量測線。9.附錄 A_USB 連線新增 WINDOWS7、WINDOWS8 及 WINDOWS10 作業系統 USB 連線方式。10.附錄 B 各系列機種注意事項修改 AS300 系列 S 裝置範圍。	2016/11/30
第七版	<ol style="list-style-type: none">1.第 2 章自動儲存內容增加備份檔數量設定及修改 AH 模擬器啟動數量與限制內容。	2017/09/30

版本	變更內容	發行日期
第七版	<p>2.第 3 章增加 AH500 備援系列機種的 HWCONFIG 相關操作介紹及新增系統配置區備援延伸背板相關操作介紹；AH500 機種之 CPU 基本參數 - 名稱/註解內容修改；AHxxEMC 機種 HWCONFIG 之 PLC 主機參數設定內，多增加本體 IO 濾波輸入。</p> <p>3.第 5 章新增 AH500 備援系列機種的工作 (Task) 管理工具內容。</p> <p>4.第 6 章 AS 系列機種支援資料型態內容修改；AH 系列機種 S 裝置支援內容修改。</p> <p>5.第 7 章新增 AS 系列機種功能塊內部區域符號新增不支援的資料型態；功能塊記憶體配置內容新增 AH500 備援系列機種；功能塊的基本規格內容修改。</p> <p>6.第 8 章新增結構項宣告結構及列舉數量限制；新增結構改變列舉及列舉改變結構內容。</p> <p>7.第 9 章新增軸項目右側週期時間的顯示；新增「與 ECAT Builder」同步選項介紹；軸參數設定之輸出裝置 ID 設定內容修改。</p> <p>8.第 10 章 MPS 指令介紹；階梯圖的致能與失效內容修改。</p> <p>9.第 13 章新增 AH/AS 系列機種 ST 語言不支援的指令列表及 AS 系列機種之 ST 語言有支援的指令但有所限制的規範事項。</p> <p>10.第 14 章修改 STEP 符號占用 PLC 內部的 S 裝置內容；增加 AH500 備援系列機種執行「Final Scan」內容。</p> <p>11.第 16 章修改轉換機種可能導致部分程式遺失需存檔之提示字樣。</p> <p>12.第 17 章新增 DVP 機種線上編輯介紹；PLC 系統資訊及 PLC 系統記錄內容修改增加 AH500 備援機種相關顯示資訊及操作內容。</p> <p>13.第 20 章新增 CARD Utility 對 AH500 備援系列機種相關內容。</p> <p>14.第 21 章敘述本版不支援 Gcode 功能；ECAM 檔案模式新增圖表檢視功能。</p> <p>15.第 22 章定位規畫表之定位表格內，位址模式新增相對模式 (符號/位址)、絕對模式 (符號/位址) 相關設定內容；定位規畫表新增符號裝置模擬數值設定內容。</p>	2017/09/30

版本	變更內容	發行日期
第七版	16.附錄 B 新增 AHCPU560-EN 機種內容。	2017/09/30
第八版	<ol style="list-style-type: none"> 1.修改第 1 章，新增 DVP15MC 機種內容，以及安裝/移除 ISPSOFT 內容修改。 2.修改第 2 章，新增 DVP15MC 機種內容；DVP 模擬器內容及 AS 模擬器啟動數量限制等內容修改；「封包繞送模式」功能內容修改。 3.修改第 3 章，新增 DVP15MC 機種內容，DVP15MC 機種 HWCONFIG 環境介紹及 PLC 主機參數設定介紹等新增內容；硬體規劃工具 - HWCONFIG 內容修改；AS 系列機種的 PLC 主機參數設定內容修改；HWCONFIG 硬體參數管理及線上診斷內容修改。 4.修改第 4 章，(致能/失效) 區段相關內容修改。 5.修改第 5 章，新增 DVP15MC 機種工作管理及 DVP15MC 機種工作管理工具操作等相關內容修改；POU 的管理操作相關內容修改；新增功能塊 POU 鎖定 (永久) 功能介紹；修改範例實作內容。 6.修改第 6 章，新增 DVP15MC 機種內容；變數符號的資料型態內容修改，新增 DVP15MC 機種資料型態內容、資料型態說明表格及相關內容修改；變數符號的位址配置與初始值內容修改；變數符號的間接指定暫存器修飾內容修改；新增變數符號，對於字串、陣列型態符號使用原則等內容修改；下載符號的初始值內容修改，新增 DVP15MC 機種相關內容；符號表的匯入內容修改；範例實作內容修改。 7.修改第 7 章，新增 DVP15MC 機種內容，功能塊的輸入與輸出針對 DVP15MC 機種等相關內容修改；範例與實作內容修改；DVP15MC 機種對函示庫內容新增。 8.修改第 8 章，新增 DVP15MC 機種內容等相關內容。 9.修改第 9 章，新增 DVP15MC 機種內容，軸操作及相關內容修改；AHxxEMC 建立軸相關內容修改；軸參數設定及監控與調測相關內容修改。 10.修改第 10 章，新增 DVP15MC 機種內容；設置標籤 (Label) 內容修改。 11.修改第 11 章，設置標籤 (Label) 內容修改。 12.修改第 14 章，新增 SFC 的系統結構內容。 	2018/05/31

版本	變更內容	發行日期
第八版	<p>13.修改第 16 章·新增 DVP15MC 機種內容；修改 DVP 轉 AS 內容；修改專案下載與上傳介紹內容；修改 DVP、AH/AS 機種的裝置註解表相關內容；修改裝置使用狀況內容。</p> <p>14.修改第 17 章·新增 DVP15MC 機種內容；程式監控之改變現在值內容修改；DVP 模擬器內容修改；除錯模式內容修改；PLC 系統資訊·AH500 備援系列「Redundant」分頁等內容修改；軸記錄內容修改。</p> <p>15.修改第 18 章·新增 DVP15MC 機種內容。</p> <p>16.修改第 19 章·NWCONFIG 簡介內容修改。</p> <p>17.修改第 20 章·AS 系列機種備份檔名稱、路徑內容修改。</p> <p>18.修改第 21 章·ECAM 編輯器使用內容修改。</p> <p>19.修改第 22 章·定位規劃表支援機種及相關內容修改；資料示波器內容修改；資料記錄器觸發取樣等內容修改；高速計數精靈內容修改。</p> <p>20.修改附錄 A·新增 AH560EN2 機種之驅動程式路徑內容。</p> <p>21.修改附錄 B·新增 DVP15MC 機種內容；各機種資料交換及 NWCONFIG 支援狀況內容新增修改；編譯與上下載時間內容修改；AHCPU501-EN、AHCPU501-RS2、AHCPU521-RS2、AHCPU531-RS2、AS200 機種內容新增；AHCPU511-EN/ AHCPU511-RS2 及 AHCPU531-EN 等機種定址範圍修改。</p>	2018/05/31
第九版	<p>1.修改第 1 章·新增 DVPxxMC 機種敘述；安裝 ISPSOft 內容及移除 ISPSOft 內容修改；安裝 COMMGR 內容及移除 COMMGR 內容修改；新增 ISPSOft PLC 機種支援的語言支援列表及功能支援列表。</p> <p>2.修改第 2 章·進入 ISPSOft 的第一步及視窗標題與狀態列內容修改；功能工具列與圖示工具列內容修改；專案管理區與訊息顯示區及編輯工作區內容修改；ISPSOft 的專案架構·單機專案與群組專案、整合運動模組專案內容修改；系統及環境設定內容修改；通訊管理員 COMMGR 的啟動與關閉及驅動管理、建立連線通道內容修改；建立 ISPSOft 與 COMMGR 之間的連結與連接主機與通訊埠口內容修改。</p>	2019/04/08

版本	變更內容	發行日期
第九版	<p>3.修改第 3 章·DVPxxMC 機種 PLC 主機參數設定內容修改；AH560 備援機種參數設定與相關說明內容修改；硬體規劃工具 - HWCONFIG 內容修改。</p> <p>4.修改第 4 章·新增專案及硬體組態與模組配置等相關內容修改；主機與模組參數設定及建立程式內容修改；基本編輯-階梯圖區段選取與操作功能增加；機本編輯-插入 API 應用指令、程式的檢查與編譯內容修改；測試與除錯建立連線、下載程式與組態參數修、連線測試範例實作內容修改。</p> <p>5.修改第 5 章·ISPSOft 的 POU 架構與 ISPSOft 中的工作(TASK)管理內容修改；專案管理區的工作(TASK)項目與工作 (TASK) 中的 POU 執行順序內容修改；POU 的建立與屬性設定及 POU 的啟動狀態、POU 的刪除與複製、POU 密碼的設定與解除、匯出程式 POU、匯入程式 POU、匯出功能塊 POU、匯入功能塊 POU 等內容修改；工作 (TASK) 管理的操作及工作 (TASK) 屬性與中斷條件的設定、POU 的配置、POU 的排序的內容修改；範例實作 POU 與 TASK 的編程範例、DVP 機種的中斷程示範例、AH500 機種的中斷程式範例等內容修改。</p> <p>6.修改第 6 章·變數符號的類別與變數符號的資料型態及變數符號的位置配置與初始值內容修改；變數符號表與新增變數符號及陣列與字串型態符號的使用原則、變數符號的修改與符號表編輯、下載變數符號的初始值、裝置資源分配設定內容修改。</p> <p>7.修改第 7 章·功能塊的變數符號與功能塊的輸出與輸入內容修改；功能塊呼叫功能塊的巢式結構與功能塊的記憶體配置及功能塊的基本規格內容修改；建立範例程式內容修改；新增使用者自訂函式庫及引用使用者自訂函式庫、引用台達函式庫內容修改。</p> <p>8.修改第 8 章·建立結構 (STRUCTURE) 內容修改。</p> <p>9.修改第 9 章·AHxxEMC 機種的建立軸內容修改；DVPxxMC 機種的建立軸與軸基本設定內容修改；全域符號表的主符號表與軸符號表內容修改；AHxxEMC 機種的軸參數設定與 DVPxxEMC 機種的軸參數設定內容修改；AHxxEMC 機種的監控與調測與 DVPxxEMC 機種的監控與調測內容修改。</p>	2019/04/08

版本	變更內容	發行日期
第九版	<p>10.修改第 10 章· ISPSOft 的 LD 編輯環境內容修改；於 LD 中使用裝置變數符號及常數、插入 API 應用指令與功能塊、編輯區塊邏輯指令、符號模式與位置模式顯示切換內容修改。</p> <p>11.修改第 11 章· ISPSOft 的 FBD 編輯環境及於 FBD 中使用裝置變數符號及常數、API 應用指令比較接點與功能塊內容修改；符號模式與位址模式的顯示切換內容修改。</p> <p>12.修改第 12 章· 編輯 IL 指令、插入 API 應用指令與功能塊內容修改。</p> <p>13.修改第 13 章· ISPSOft 的 ST 編輯環境、編輯 ST 敘述、插入 API 應用指令與功能塊、建立範例程式內容修改。</p> <p>14.修改第 14 章· 步進點與 Action、轉換點(Transition)、跳躍點、Action 的修飾條件、條件定義、初始步進點等內容修改；內部屬性、Step 屬性參數、Action 與 Transition 屬性參數內容修改；ISPSOft 的 SFC 編輯環境、Action 與 Transition 的建立與管理、指定步進點與轉換點符號、配置 Action/摺疊 Action 列表內容修改；建立範例程式內容修改。</p> <p>15.修改第 15 章· 指令與功能塊內容。</p> <p>16.修改第 16 章· 新增變更 PLC 機種列表，新增專案的下載與上載列表、DVP 機種的裝置註解表及裝置使用狀況、程式比對內容修改。</p> <p>17.修改第 17 章· DVP 機種的除錯模式、AH500 機種的模擬器內容修改；PLC 系統資訊、PLC 系統紀錄內容修改。</p> <p>18.修改第 18 章· ISPSOft 的密碼架構、POU 密碼、副程式密碼、其他密碼的資料保護功能內容修改。</p>	2019/04/08

版本	變更內容	發行日期
第九版	<p>19.修改第 19 章，NWCONFIG 簡介、相關基礎介紹、設定通訊參數、工作流程內容修改；部署節點、連結網路、節點與網路屬性的設定、裝置或網路的隱藏/顯示、合法的網路架構、下載繞送路徑表、繞送路徑測試內容修改；選擇主站裝置、設定通訊參數、資料交換表簡介、設定區塊、資料交換表的機種同步性、資料交換表的管理操作、PLC Link 的監控功能、設置 PLC Link 的注意事項內容修改；Ether Link 簡介、開啟 Ether Link 的設置視窗、資料交換表的建立與管理、裝置清單與圖示的操作、Ether Link 的運作模式、下載 Ether Link 設置、上載 Ether Link 設置、刪除非同步裝置、線上啟動/停止 Ether Link、整體下載、ISPSOft 中的繞送應用內容修改。</p> <p>20.修改第 20 章，資料備份記憶卡操作、PLC 永久備份設定、記憶卡工具 CARD Utility 軟體介紹、備份操作、還原操作內容修改。</p> <p>21.修改第 21 章，刪除 G-Code 編輯器內容；E-CAM 編輯器、關於 E-CAM(電子凸輪)、電子凸輪意義說明、使用 E-CAM 編輯器內容修改。</p> <p>22.修改第 22 章，設定定位規劃表、定位規劃表的模擬、上載與下載定位規劃表內容修改；軸控模式-單軸點對點模式、單軸多段輸出模式、雙軸直線補間模式、雙軸圓弧補間模式內容修改；關於資料示波器、開啟資料示波器、設定取樣參數、取樣模式、量測內容修改；關於資料紀錄器、開啟資料紀錄器、設定取樣參數、監看與錄製模式、量測內容修改；關於高數計數器精靈內容修改。</p> <p>23.修改附錄 A，在 Windows10 環境下安裝 USB 驅動程式內容。</p> <p>24.修改附錄 B，DVPxxMC 機種的裝置種類、DVPxxMC 機種之裝置的定址格式、DVPxxMC 機種之停電保持裝置、DVPxxMC 系列機種內容修改。</p>	2019/04/08

版本	變更內容	發行日期
第十版	<ol style="list-style-type: none"> 1.所有章節因增加第 16 章，內容提及第 16 章-第 22 章內容更新為調整後章節。 2.修改第 1 章功能支援列表更新。 3.修改第 2 章最近使用專案的顯示方式變更，增加功能列 PLC 主機裡的初始值選項。 4.修改第 3 章增加 HWCONFIG 4.0 的使用說明。 5.修改第 4 章 IEC61131-3 文字。 6.更新第 5.3.2 節 EI 程式圖。 7.修改第 6 章全域變數表增加分類功能與宣告變量的初始值說明。 8.修改第 10 章書籤定位功能會顯示連結在專案樹內。 9.修改第 13 章支援單行註解與/* */註解說明，輸入可以自動帶出結構。 10.增加第 16 章 C 語言。 11.修改第 17 章增加搜尋裝置與搜尋變數說明。 12.修改第 18 章增加線上編輯可以暫時記住密碼選項。 13.修改第 22 章 G-Code 支援。 	2019/11/30
第十一版	<ol style="list-style-type: none"> 1.在各章節增加 AS516E-B、AS524C-B (簡稱為 AS5xx 機種) 相關資料。 2.修改第 1 章系統需求、語言支援列表、功能支援列表更新。 3.修改第 2 章工具>選項裡面的 ISPSOFT 預載項目內的功能項目，增加編輯器設定的 IL 及 ST 和 C 支援提示的選項設定，增加通訊設定的 COMMGR 按鈕說明，增加匯出程式到文字格式。 4.第 3 章將第 3.1.2 節及第 3.1.3 節內容合併，增加第 3.1.2.8 節功能工具列-常用，增加 IO-Link 及 HWCONFIG 符號設定說明。 5.增加第 5.2.1 節與第 5.4.1 節對函式的介紹。 6.增加第 6.1.2 節變數符號表增加 VAR_STATIC 與 VAR_STATIC_RETAIN 類別的說明，第 6.1.3 節增加函式可以使用的資料型態與指標 (Pointer) 及參考 (Reference) 資料型態說明，第 6.2.6 節停電保持符號初始值生效說明。 7.增加第 7 章函式的相關說明。 8.增加第 8 章聯合 (UNION) 相關說明，修改結構、列舉章節的圖片及相關說明。 	2020/06/30

版本	變更內容	發行日期
第十一版	<p>9.增加第 9 章軸支援列表，增加伺服參數備份還原章節，增加自動增益調整章節。</p> <p>10.增加第 10.2.4 節在 DVPxxMC 與 AS5xx 機種的功能塊引腳支援 ST 運算式的寫法。</p> <p>11.修改第 15.2.10 節變更程式的執行順序說明</p> <p>12.修改第 17.1.1 節變更 PLC 機種支援列表內容，修改程式上下載圖片，變更尋找取代章節說明內容與圖片</p> <p>13.增加第 21.3 節 CARD Utility 備份還原支援列表。</p> <p>14.增加第 23.5 節 NTC 精靈介紹。</p> <p>15.增加附錄 B 章的 AS5xx 相關說明。</p>	2020/06/30

ISPSOft 使用手冊

目錄

前言	I
第 1 章 軟體簡介	
1.1 ISPSOft 簡介及系統需求	1-2
1.1.1 特色簡介	1-2
1.1.2 系統需求	1-3
1.1.3 安裝 ISPSOft	1-3
1.1.4 移除 ISPSOft	1-8
1.2 通訊管理員 COMMGR 簡介	1-9
1.2.1 COMMGR 的工作模式	1-9
1.2.2 安裝 COMMGR	1-10
1.2.3 移除 COMMGR	1-13
1.3 ISPSOft PLC 機種支援	1-14
1.3.1 ISPSOft 語言支援列表	1-14
1.3.2 ISPSOft 功能支援列表	1-14
第 2 章 啟動與設定	
2.1 啟始導覽與環境介紹	2-3
2.1.1 進入 ISPSOft 的第一步	2-3
2.1.2 視窗標題與狀態列	2-8
2.1.3 功能工具列	2-9
2.1.4 圖示工具列	2-11
2.1.5 專案管理區與訊息顯示區	2-12
2.1.6 編輯工作區	2-14
2.2 ISPSOft 的專案架構	2-17
2.2.1 單機專案與群組專案	2-17
2.2.2 整合運動模組專案	2-18
2.2.3 ISPSOft 的單機專案管理	2-19
2.2.4 ISPSOft 的群組專案管理	2-24
2.2.5 ISPSOft 的運動模組專案管理	2-31

2.3 ISPSOft 中的基本設定.....	2-36
2.3.1 系統及環境設定	2-36
2.3.2 使用者編輯格式的匯入與匯出.....	2-42
2.3.3 匯出程式到文字格式.....	2-43
2.4 通訊設定	2-44
2.4.1 通訊管理員 COMMGR 的啟動與關閉	2-44
2.4.2 COMMGR 的驅動管理	2-46
2.4.3 建立連線通道 - 驅動的建立	2-46
2.4.3.1 RS232/422/485 通訊參數設定.....	2-48
2.4.3.2 USB (Virtual COM) 通訊參數設定	2-49
2.4.3.3 DirectLink (USB) 通訊參數設定	2-50
2.4.3.4 Ethernet 通訊參數設定	2-50
2.4.3.5 DirectLink (Ethernet) 通訊參數設定	2-51
2.4.3.6 DVP Simulator 通訊參數設定	2-52
2.4.3.7 AH5x0 與 AH5x1 Simulator 通訊參數設定.....	2-53
2.4.3.8 AS Simulator 通訊參數設定.....	2-53
2.4.4 建立連線通道 - 驅動的啟動與停止	2-54
2.4.5 建立連線通道 - 驅動的修改與刪除	2-55
2.4.6 建立連線通道 - 模擬器的面板操作	2-56
2.4.7 建立 ISPSOft 與 COMMGR 之間的連結.....	2-59
2.4.8 連接主機與通訊埠口.....	2-62
2.4.9 實際連線測試.....	2-65
第 3 章 PLC 系統的配置與設定	
3.1 硬體規劃工具 - HWCONFIG 4.0.....	3-6
3.1.1 進入 HWCONFIG 的第一步.....	3-7
3.1.2 HWCONFIG 的環境介紹.....	3-10
3.1.2.1 視窗標題與狀態列	3-11
3.1.2.2 功能工具列-檔案.....	3-11
3.1.2.3 功能工具列-編輯.....	3-18
3.1.2.4 功能工具列-檢視.....	3-24
3.1.2.5 功能工具列-通訊.....	3-24
3.1.2.6 功能工具列-工具.....	3-32

3.1.2.7 功能工具列-說明.....	3-41
3.1.2.8 功能工具列-常用.....	3-42
3.1.2.9 專案樹.....	3-42
3.1.2.10 產品列表.....	3-43
3.1.2.11 輸出視窗.....	3-44
3.1.2.12 硬體配置區.....	3-45
3.1.2.13 編輯區.....	3-46
3.1.3 進階功能操作.....	3-47
3.1.3.1 新增模組.....	3-47
3.1.3.2 變更模組.....	3-52
3.1.3.3 模組的搬移.....	3-53
3.1.3.4 備註編輯.....	3-54
3.1.3.5 硬體配置區-重設配置.....	3-56
3.1.3.6 硬體配置區-顯示模組詳細訊息.....	3-57
3.1.3.7 硬體配置區-輸出/輸入裝置重新排序.....	3-57
3.1.3.8 硬體配置區-改變遠端裝置位址.....	3-58
3.1.3.9 硬體配置區-重置大小.....	3-59
3.1.3.10 編輯區-模組內部參數的匯出與匯入.....	3-60
3.1.3.11 編輯區-模組內部參數的更新.....	3-61
3.1.3.12 顯示/隱藏功能.....	3-61
3.1.3.13 資訊檢視與模組診斷.....	3-62
3.1.3.14 線上變更模組狀態.....	3-62
3.1.3.15 遠端模組配置.....	3-64
3.1.3.16 開啟外部軟體.....	3-66
3.1.3.17 IO-Link.....	3-66
3.1.3.18 符號設定.....	3-85
3.1.4 PLC 主機參數設定.....	3-89
3.1.4.1 系統設定-系統資訊.....	3-90
3.1.4.2 系統設定-系統參數.....	3-91
3.1.4.3 系統設定-裝置元件範圍設定.....	3-95
3.1.4.4 系統設定-輸入點濾波時間.....	3-96
3.1.4.5 系統設定-定位軸參數.....	3-97
3.1.4.6 系統設定-日光節約時間.....	3-98

3.1.4.7	COM 通訊埠設定	3-99
3.1.4.8	乙太網路基本設定	3-100
3.1.4.9	乙太網進階設定-IP 過濾器	3-101
3.1.4.10	乙太網進階設定-網路時間校正	3-102
3.1.4.11	乙太網進階設定-郵件	3-103
3.1.4.12	乙太網進階設定-通訊接口	3-108
3.1.4.13	乙太網進階設定- RTU 對應	3-109
3.1.4.14	功能卡 1 設定	3-111
3.1.4.15	功能卡 2 設定	3-111
3.1.4.16	台達設備參數還原功能	3-115
3.1.4.17	資料交換	3-120
3.2	硬體規劃工具 - HWCONFIG 3.0	3-121
3.2.1	HWCONFIG 的環境介紹	3-121
3.2.2	模組配置	3-125
3.2.2.1	新增模組	3-125
3.2.2.2	模組的位址配置	3-127
3.2.2.3	註解編輯	3-131
3.2.2.4	刪除模組	3-132
3.2.2.5	變更模組	3-134
3.2.2.6	模組的搜尋/取代功能	3-136
3.2.2.7	模組的複製/貼上	3-139
3.2.2.8	模組的剪下/貼上	3-141
3.2.2.9	模組的搬移	3-142
3.2.2.10	新增延伸背板及備援延伸背板	3-143
3.2.2.11	刪除延伸背板及備援延伸背板	3-147
3.2.2.12	變更背板	3-149
3.2.2.13	延伸背板及備援延伸背板的剪下/複製/貼上	3-151
3.2.2.14	延伸背板圖示及備援延伸背板圖示的搬移	3-156
3.2.2.15	重新排序輸入/輸出裝置範圍	3-157
3.2.3	主機與模組的版本管理	3-157
3.3	AH500 機種的 PLC 主機參數設定	3-160
3.3.1	開啟 PLC 主機的參數設定頁面	3-160
3.3.2	CPU 基本參數設定	3-161

3.3.2.1 CPU 基本參數 - 名稱	3-161
3.3.2.2 CPU 基本參數 - 系統	3-162
3.3.2.3 CPU 基本參數 - 停電保持裝置範圍	3-166
3.3.2.4 CPU 進階設定.....	3-168
3.3.3 序列埠 (COM Port) 參數設定.....	3-169
3.3.4 乙太網路 (Ethernet) 基本參數設定.....	3-170
3.3.5 乙太網路 (Ethernet) 進階參數設定.....	3-171
3.3.5.1 乙太網路進階參數 - 過濾器	3-171
3.3.5.2 乙太網路進階參數 - 網路時間校正	3-173
3.3.5.3 乙太網路進階參數 - 郵件	3-174
3.3.5.4 乙太網路進階參數 - 郵件發送條件	3-175
3.3.5.5 乙太網路進階參數 - 郵件位址與發送條件設定	3-180
3.3.5.6 乙太網路進階參數 - 通訊接口 (Socket)	3-181
3.3.5.7 乙太網路進階參數 - 網頁功能	3-183
3.3.5.8 乙太網路進階參數 - 資料交換	3-184
3.3.6 資料交換 - COM 與資料交換 - Ethernet	3-187
3.3.6.1 資料交換-COM1、資料交換-COM2 或資料交換-COM.....	3-188
3.3.6.2 資料交換-Ethernet	3-189
3.4 AHxxEMC 機種的 PLC 主機參數設定	3-190
3.4.1 開啟 PLC 主機的參數設定頁面.....	3-190
3.4.2 一般設定	3-192
3.4.2.1 一般設定 - 系統資訊	3-192
3.4.2.2 一般設定 - COM Port Setting.....	3-197
3.4.2.3 一般設定 - 乙太網路 - 基本設定	3-198
3.4.2.4 一般設定 - 乙太網進階設定	3-199
3.4.3 資料交換-COM2 與資料交換-Ethernet.....	3-200
3.4.3.1 資料交換-COM2.....	3-202
3.4.3.2 資料交換-Ethernet	3-203
3.5 DVP 系列機種的參數設定	3-203
3.5.1 DVP 主機的系統管理工具	3-203
3.5.2 停電保持設定.....	3-204
3.5.3 連線資訊	3-208
3.5.4 DVP-ES3 機種的主機參數設定	3-209

3.6 DVPxxMC 機種的 PLC 主機參數設定	3-211
3.6.1 開啟 PLC 主機的參數設定頁面	3-211
3.6.2 一般設定	3-212
3.6.2.1 一般設定-系統資訊	3-212
3.6.2.2 一般設定-CANopen	3-213
3.6.2.3 一般設定-CANMotion	3-214
3.6.2.4 一般設定-RS232	3-214
3.6.2.5 一般設定-RS485	3-215
3.6.2.6 一般設定-增量型編碼器 1	3-215
3.6.2.7 一般設定-增量型編碼器 2	3-216
3.6.2.8 一般設定-絕對型編碼器	3-217
3.6.2.9 一般設定-輸入點過濾時間	3-218
3.6.2.10 一般設定-運轉 / 停止設定	3-219
3.6.2.11 一般設定-乙太網路基本設定	3-220
3.7 模組參數設定	3-221
3.7.1 設定 AH5x0 與 AH5x1 模組內部參數	3-221
3.7.2 設定 AHxxEMC 機種模組內部參數	3-226
3.7.3 模組內部參數的匯出與匯入	3-229
3.7.4 模組內部參數的更新	3-229
3.7.5 智慧型模組參數設定	3-229
3.8 HWCONFIG 硬體參數管理及線上診斷	3-230
3.8.1 硬體規劃的儲存與列印	3-230
3.8.2 訂購清單	3-232
3.8.3 AH 系列機種的背板資訊列表	3-233
3.8.4 系統參數上下載	3-234
3.8.5 AH 系列機種的模組布局偵測	3-235
3.8.6 AH 系列機種的線上診斷	3-239
3.8.6.1 啟動線上操作模式	3-239
3.8.6.2 資訊檢視與模組診斷	3-240
3.8.6.3 線上變更模組狀態	3-241
3.8.6.4 監控表	3-244
3.9 萬年曆設定	3-245
3.10 PLC 記憶體設定 (清除程式&回復出廠值)	3-246

第 4 章 快速入門

4.1 快速入門	4-2
4.1.1 範例功能說明	4-2
4.1.2 硬體規劃	4-2
4.1.3 程式規劃	4-3
4.2 ISPSoft 的專案開發程序	4-3
4.3 新增專案	4-4
4.4 硬體組態	4-5
4.4.1 模組配置	4-6
4.4.2 主機與模組參數設定	4-8
4.5 建立程式	4-10
4.5.1 新增階梯圖程式	4-10
4.5.2 基本編輯 - 建立接點與線圈	4-12
4.5.3 基本編輯 - 階梯圖區段的新增/插入與指令編輯模式	4-15
4.5.4 基本編輯 - 階梯圖區段的選取與操作	4-17
4.5.5 基本編輯 - 並聯接點裝置	4-20
4.5.6 基本編輯 - 編輯註解	4-21
4.5.7 基本編輯 - 插入 API 應用指令	4-22
4.5.8 基本編輯 - 比較接點的建立與常數的輸入格式	4-24
4.5.9 完成範例程式	4-25
4.5.10 程式的檢查與編譯	4-26
4.6 測試與除錯	4-27
4.6.1 建立連線	4-27
4.6.2 下載程式與組態參數	4-30
4.6.3 連線測試	4-32

第 5 章 POU 與 TASK

5.1 關於程式組織單元 (POU) 與工作 (TASK)	5-2
5.2 認識程式組織單元 (POU)	5-3
5.2.1 POU 的編程架構與種類	5-3
5.2.2 ISPSoft 的 POU 架構	5-4
5.3 認識工作 (TASK)	5-5

5.3.1 ISPSoft 中的工作 (TASK) 管理	5-5
5.3.2 專案管理區的工作 (TASK) 項目	5-10
5.3.3 工作 (TASK) 中的 POU 執行順序	5-12
5.4 POU 的管理操作	5-14
5.4.1 POU 的建立與屬性設定	5-14
5.4.2 POU 的啟動狀態	5-18
5.4.3 POU 的刪除與複製	5-19
5.4.4 POU 密碼的設定與解除	5-21
5.4.5 匯出程式 POU	5-23
5.4.6 匯入程式 POU	5-24
5.4.7 匯出功能塊 POU	5-25
5.4.8 匯入功能塊 POU	5-26
5.5 工作 (TASK) 管理的操作	5-27
5.5.1 工作 (TASK) 屬性與中斷條件的設定	5-29
5.5.2 POU 的配置	5-34
5.5.3 POU 的排序	5-36
5.6 範例實作	5-37
5.6.1 POU 與 TASK 的編程範例	5-37
5.6.2 DVP 機種的中斷程式範例	5-41
5.6.3 AH5x0 機種的中斷程式範例	5-46
第 6 章 變數符號	
6.1 認識變數符號	6-2
6.1.1 變數符號的作用範圍與命名原則	6-2
6.1.2 變數符號的類別	6-3
6.1.3 變數符號的資料型態	6-5
6.1.4 變數符號的位址配置與初始值	6-10
6.1.5 變數符號的間接指定暫存器修飾	6-12
6.1.6 變數符號的位元操作 (AH/AS 機種限定)	6-14
6.2 ISPSoft 中的變數符號管理	6-15
6.2.1 變數符號表	6-15
6.2.2 新增變數符號	6-18
6.2.3 陣列與字串型態符號的使用原則	6-23

6.2.4 變數符號的修改與符號表編輯.....	6-26
6.2.5 清除變數符號的配置位址	6-28
6.2.6 下載變數符號的初始值	6-29
6.2.7 符號表的匯出	6-31
6.2.8 符號表的匯入	6-32
6.2.9 符號表的排序	6-35
6.2.10 變數符號的篩選	6-36
6.2.11 裝置資源分配設定	6-37
6.3 範例實作	6-38
6.3.1 變數符號規劃	6-39
6.3.2 建立範例程式	6-40

第 7 章 功能塊與函式庫

7.1 認識功能塊 (Function Block)	7-2
7.1.1 功能塊簡介	7-2
7.1.2 功能塊的特性與優點	7-3
7.2 ISPSOFT 功能塊 (Function Block) 架構	7-4
7.2.1 功能塊的致能 (En) 端點	7-4
7.2.2 功能塊的變數符號	7-5
7.2.3 功能塊的輸出與輸入	7-6
7.2.4 指標型態的變數符號	7-12
7.2.5 功能塊的定義與實例	7-16
7.2.6 功能塊呼叫功能塊的巢式結構	7-19
7.2.7 功能塊的記憶體配置	7-21
7.3 功能塊的使用	7-26
7.3.1 功能塊的基本規格	7-26
7.3.2 功能塊專用的脈波型指令 (AH/AS 機種限定)	7-27
7.3.3 功能塊的程式監控	7-29
7.3.4 功能塊的修改	7-30
7.4 範例實作	7-31
7.4.1 程式規劃	7-31
7.4.2 建立範例程式	7-31
7.4.3 函式 (Function) 範例程式	7-41

7.5 認識函式庫 (Library)	7-42
7.5.1 新增使用者自訂函式庫	7-42
7.5.2 引用使用者自訂函式庫	7-44
7.5.3 引用台達函式庫	7-46
第 8 章 自定義資料型態	
8.1 自定義資料型態	8-2
8.2 結構 (STRUCTURE)	8-2
8.2.1 結構 (STRUCTURE) 的意義	8-2
8.2.2 建立結構 (STRUCTURE)	8-3
8.2.3 使用結構 (STRUCTURE) 型態變數符號	8-5
8.2.4 結構 (STRUCTURE) 的常見用法	8-7
8.2.5 結構 (STRUCTURE) 改變為列舉 (ENUMERATION)	8-10
8.3 列舉 (ENUMERATION)	8-11
8.3.1 列舉 (ENUMERATION) 的意義	8-11
8.3.2 建立列舉 (ENUMERATION)	8-11
8.3.3 使用列舉 (ENUMERATION) 型態變數符號	8-13
8.3.4 列舉 (ENUMERATION) 的常見用法	8-15
8.3.5 列舉 (ENUMERATION) 改變為結構 (STRUCTURE)	8-16
8.4 聯合 (UNION)	8-17
8.4.1 聯合 (UNION) 的意義	8-17
8.4.2 建立聯合 (UNION)	8-17
8.4.3 使用聯合 (UNION) 型態變數符號	8-20
8.4.4 聯合 (UNION) 的常見用法	8-21
8.4.5 內建聯合的支援與名稱	8-23
第 9 章 軸	
9.1 軸	9-2
9.1.1 軸的意義	9-2
9.1.2 建立軸	9-3
9.1.2.1 AHxxEMC 機種的建立軸	9-3
9.1.2.2 DVPxxMC / AS5xx 機種的建立軸與軸基本設定	9-5
9.1.3 全域符號表的主符號表與軸符號表	9-9

9.2 軸的參數設定、監控與調測	9-10
9.2.1 軸參數設定.....	9-10
9.2.1.1 AHxxEMC 機種的軸參數設定	9-10
9.2.1.2 DVPxxMC / AS5xx 機種的軸參數設定.....	9-14
9.2.2 監控與調測.....	9-21
9.2.2.1 AHxxEMC 機種的監控與調測	9-21
9.2.2.2 DVPxxMC/AS5xx 機種的監控與調測	9-26
9.2.3 伺服參數備份還原	9-33
9.2.4 自動增益調整.....	9-36
第 10 章 階梯圖 LD	
10.1 階梯圖 (LD) 簡介	10-2
10.1.1 ISPSOft 的 LD 編輯環境.....	10-2
10.1.2 階梯圖區段 (NETWORK)	10-4
10.1.3 選取物件.....	10-5
10.2 於 ISPSOft 中建立 LD 程式.....	10-6
10.2.1 建立接點 (Contact)、線圈 (Coil) 及 MPS.....	10-6
10.2.1.1 插入接點 (Contact) 與變更接點型態	10-6
10.2.1.2 插入線圈 (Coil) 與變更線圈型態.....	10-8
10.2.1.3 新增 MPS	10-9
10.2.2 於 LD 中使用裝置、變數符號及常數	10-10
10.2.3 指令編輯模式	10-11
10.2.4 插入 API 應用指令與功能塊	10-12
10.2.5 編輯比較接點	10-16
10.2.6 編輯區塊邏輯指令.....	10-19
10.2.7 編輯多重輸出	10-21
10.2.8 設置標籤 (Label)	10-22
10.2.9 編輯註解.....	10-23
10.2.10 符號模式與位址模式的顯示切換	10-25
10.2.11 書籤功能.....	10-26
10.2.12 階梯圖區段的致能與失效.....	10-28

第 11 章 功能塊圖 FBD

11.1 功能塊圖 FBD 簡介	11-2
11.1.1 關於功能塊圖 (FBD)	11-2
11.1.2 ISPSOft 的 FBD 編輯環境.....	11-2
11.1.3 功能塊圖區段 (NETWORK)	11-4
11.1.4 選取物件.....	11-5
11.2 於 ISPSOft 中建立 FBD 程式.....	11-6
11.2.1 輸入節點與輸出節點.....	11-7
11.2.2 於 FBD 中使用裝置、變數符號及常數	11-10
11.2.3 AND 與 OR 方塊.....	11-11
11.2.4 反相邏輯.....	11-14
11.2.5 上緣觸發輸入與下緣觸發輸入.....	11-16
11.2.6 設置 (SET) 輸出與重置 (RESET) 輸出.....	11-18
11.2.7 API 應用指令、比較接點與功能塊	11-20
11.2.8 設置標籤 (Label)	11-24
11.2.9 註解與提示.....	11-25
11.2.10 符號模式與位址模式的顯示切換	11-27
11.2.11 書籤功能.....	11-28
11.2.12 功能塊圖區段的致能與失效.....	11-29
第 12 章 指令列表 IL	
12.1 指令列表 (IL) 簡介	12-2
12.1.1 IL 程式的結構	12-2
12.1.2 IL 的功能塊 (FB) 呼叫指令	12-4
12.1.3 IL 編程的注意事項.....	12-5
12.2 於 ISPSOft 中建立 IL 程式.....	12-6
12.2.1 ISPSOft 的 IL 編輯環境.....	12-6
12.2.2 編輯 IL 指令	12-7
12.2.3 插入 API 應用指令與功能塊	12-8
12.2.4 書籤功能.....	12-10
12.2.4.1 加入書籤	12-10
12.2.4.2 取消書籤	12-11
12.2.4.3 移至書籤位置.....	12-12

第 13 章 結構化語言 ST

13.1 認識結構化語言 ST	13-3
13.1.1 ST 程式的基本結構.....	13-3
13.1.2 敘述句 (Statement)	13-4
13.1.3 表達式 (Expression)	13-5
13.1.4 運算元 (Operand) 與運算子 (Operator)	13-6
13.1.5 關鍵字 (Keyword) 與註解 (Comment)	13-7
13.1.6 於 ST 程式中使用陣列符號	13-8
13.1.7 ST 編程的注意事項.....	13-9
13.2 ISPSOft 中的 ST 敘述句結構	13-11
13.2.1 指定結構 :=	13-11
13.2.2 條件結構 IF	13-13
13.2.3 條件結構 CASE	13-15
13.2.4 迴圈結構 REPEAT	13-17
13.2.5 迴圈結構 WHILE.....	13-18
13.2.6 迴圈結構 FOR	13-19
13.2.7 API 應用指令結構	13-20
13.2.8 功能塊 (FB) 呼叫結構	13-21
13.2.9 空白敘述.....	13-23
13.2.10 RETURN 敘述	13-24
13.2.11 EXIT 敘述	13-24
13.3 於 ISPSOft 中建立 ST 程式.....	13-26
13.3.1 ISPSOft 的 ST 編輯環境.....	13-26
13.3.2 編輯 ST 敘述.....	13-27
13.3.3 插入 API 應用指令與功能塊	13-28
13.3.4 書籤功能.....	13-29
13.4 ST 的程式範例.....	13-30
13.4.1 範例說明.....	13-30
13.4.2 硬體規劃.....	13-31
13.4.3 程式規劃.....	13-31
13.4.4 建立範例程式	13-32

第 14 章 循序功能圖 SFC

14.1 認識循序功能圖 SFC.....	14-3
14.1.1 SFC 程式架構	14-3
14.1.2 SFC 動作原理	14-4
14.2 ISPSOft 中的 SFC 架構.....	14-5
14.2.1 步進點與 Action	14-5
14.2.2 轉換點 (Transition)	14-9
14.2.3 共進分歧與選擇分歧.....	14-11
14.2.4 共進合流與選擇合流.....	14-12
14.2.5 跳躍點	14-13
14.2.6 Actions 的修飾條件.....	14-15
14.2.6.1 條件定義	14-15
14.2.6.2 設定修飾條件的注意事項	14-19
14.2.7 初始步進點.....	14-20
14.2.8 內部屬性.....	14-21
14.2.8.1 Step 屬性參數	14-21
14.2.8.2 Action 與 Transition 屬性參數	14-25
14.3 於 ISPSOft 中建立 SFC 程式	14-33
14.3.1 ISPSOft 的 SFC 編輯環境.....	14-33
14.3.2 Actions 與 Transitions 的建立與管理.....	14-35
14.3.3 新增步進點結構	14-37
14.3.4 插入並聯轉換點	14-38
14.3.5 插入並聯步進點	14-39
14.3.6 插入步進點結構 - 共進分歧&選擇合流.....	14-40
14.3.7 插入步進點結構 - 選擇分歧&共進合流.....	14-41
14.3.8 插入跳躍點.....	14-42
14.3.9 指定步進點與轉換點符號	14-43
14.3.10 指定初始步進點.....	14-43
14.3.11 配置 Actions/摺疊 Action 列表.....	14-44
14.4 SFC 的程式範例.....	14-47
14.4.1 範例說明.....	14-47
14.4.2 硬體規劃.....	14-48
14.4.3 程式規劃.....	14-48
14.4.4 建立範例程式	14-49

第 15 章 連續功能圖 CFC

15.1 連續功能圖 (CFC) 簡介.....	15-2
15.1.1 關於連續功能圖	15-2
15.1.2 CFC 編程的注意事項	15-2
15.1.3 ISPSOft 的 CFC 編輯環境.....	15-3
15.1.4 CFC 的圖示工具列.....	15-3
15.1.5 CFC 編輯工作區的快捷選單.....	15-4
15.2 於 ISPSOft 中建立 CFC 程式	15-5
15.2.1 選取物件.....	15-5
15.2.2 輸入節點、輸出節點與邏輯閘	15-6
15.2.2.1 插入節點或閘	15-7
15.2.2.2 新增與刪除接腳	15-9
15.2.3 改變接腳的形式	15-10
15.2.4 物件的連線與取消.....	15-11
15.2.5 指令與功能塊	15-11
15.2.5.1 插入指令	15-13
15.2.5.2 插入功能塊.....	15-15
15.2.6 刪除物件.....	15-17
15.2.7 編輯裝置或符號	15-18
15.2.8 物件的致能與失效.....	15-19
15.2.9 插入註解.....	15-19
15.2.10 變更程式的執行順序	15-22
15.2.11 顯示裝置註解	15-22

第 16 章 C 語言

16.1 C 語言編輯 (C Language)	16-2
16.1.1 C 語言之架構說明	16-2
16.1.2 C 語言編輯環境說明.....	16-3
16.1.3 C 語言編譯與除錯	16-9
16.1.4 內建函式庫指令說明.....	16-9
16.1.5 書籤與註解	16-10
16.1.6 範例操作說明	16-12

第 17 章 編輯輔助工具

17.1 ISPSOft 的編輯工具與輔助功能	17-2
17.1.1 變更 PLC 機種	17-2
17.1.2 將 DVP 機種轉換為 AS 機種	17-3
17.1.3 專案的下載與上傳	17-15
17.1.4 LD、FBD 與 CFC 編輯環境的尋找取代與到功能	17-17
17.1.5 IL 與 ST 編輯環境的尋找取代功能	17-19
17.1.6 SFC 編輯環境的尋找取代功能	17-23
17.1.7 符號表的尋找功能	17-25
17.1.8 專案尋找與取代	17-26
17.1.9 列印功能	17-28
17.2 裝置與暫存器管理	17-30
17.2.1 DVP 機種的裝置註解表	17-30
17.2.2 AH/AS 系列機種的裝置註解表	17-34
17.2.3 裝置使用狀況	17-40
17.2.4 暫存器編輯	17-43
17.2.5 裝置狀態編輯	17-50
17.2.6 DVP 機種的檔案暫存器編輯	17-54
17.3 步序定位	17-61
17.3.1 使用步序定位	17-61
17.4 程式比對	17-62
17.4.1 程式比對簡介	17-62
17.4.2 與檔案比對	17-63
17.4.3 與 PLC 比對	17-64

第 18 章 測試與除錯工具

18.1 線上監控	18-2
18.1.1 切換主機的運轉/停止狀態	18-2
18.1.2 線上監控的功能與環境介紹	18-2
18.1.3 線上模式下變更 X 裝置的狀態	18-7
18.1.4 程式監控的線上操作	18-9
18.1.5 裝置監控表	18-16

18.1.6 線上編輯與線上更新	18-22
18.2 DVP 機種的除錯模式	18-25
18.2.1 啟動 DVP 的除錯模式	18-25
18.2.2 設定與清除中斷點	18-26
18.2.3 除錯模式下的程式執行	18-26
18.3 AH/AS 機種的除錯模式	18-28
18.3.1 進入除錯模式	18-28
18.3.2 中斷點操作	18-28
18.3.3 連續執行	18-29
18.3.4 單步執行	18-30
18.4 確認 PLC 狀態	18-31
18.4.1 PLC 系統資訊	18-31
18.4.2 PLC 系統記錄	18-33
18.5 3D 座標圖	18-40
18.5.1 功能說明	18-40
18.5.2 建立 3D 座標圖	18-41
18.5.3 加入曲線	18-42
18.5.4 顯示 3D 座標圖	18-44

第 19 章 密碼管理與資料保護機制

19.1 ISPSOft 的密碼架構	19-2
19.2 程式識別碼與 PLC 識別碼	19-3
19.2.1 程式識別碼的設定與解除	19-3
19.2.2 PLC 識別碼的設定與解除	19-4
19.3 專案密碼與 PLC 密碼	19-5
19.3.1 專案密碼的設定與解除	19-5
19.3.2 PLC 密碼的設定與解除	19-7
19.3.3 專案密碼與 PLC 密碼同步	19-8
19.4 POU 密碼	19-9
19.5 副程式密碼	19-10
19.5.1 副程式密碼簡介	19-10
19.5.2 副程式密碼的設定與解除	19-11
19.6 其它的密碼與資料保護功能	19-12

19.6.1 程式上載失效	19-12
19.6.2 唯讀裝置區域設定	19-12
19.6.3 TC-01 密碼鑰匙設定	19-13
第 20 章 網路規劃與資料交換	
20.1 網路規劃工具 - NWCONFIG	20-3
20.1.1 NWCONFIG 簡介	20-3
20.1.2 相關基礎介紹	20-4
20.1.3 NWCONFIG 的通訊設定	20-6
20.1.3.1 NWCONFIG 中的連線機制	20-6
20.1.3.2 設定通訊參數	20-8
20.1.4 工作流程	20-9
20.2 網路架構的部署	20-13
20.2.1 部署節點	20-13
20.2.2 連結網路	20-17
20.2.3 調整或刪除已部署的裝置及網路	20-21
20.2.4 節點與網路屬性的設定	20-24
20.2.5 裝置或網路的隱藏/顯示	20-28
20.2.6 合法的網路架構	20-31
20.2.7 下載繞送路徑表 (Routing Table)	20-34
20.2.8 繞送路徑測試	20-36
20.3 PLC Link 的建構	20-38
20.3.1 開啟 PLC Link 的設置視窗	20-38
20.3.2 選擇主站裝置 (步驟 1)	20-40
20.3.3 設定通訊參數 (步驟 2)	20-41
20.3.4 設置資料交換表 (步驟 3)	20-43
20.3.4.1 資料交換表簡介	20-43
20.3.4.2 設定區塊	20-46
20.3.4.3 資料交換表的機種同步性	20-48
20.3.4.4 資料交換表的管理操作	20-49
20.3.5 PLC Link 的監控功能	20-50
20.3.6 設置 PLC Link 的注意事項	20-55
20.4 Ether Link 的建構	20-56

20.4.1 Ether Link 簡介	20-56
20.4.2 開啟 Ether Link 的設置視窗	20-57
20.4.3 資料交換表的建立與管理	20-59
20.4.4 裝置清單與圖示區的操作	20-62
20.4.5 Ether Link 的運作模式	20-64
20.4.6 下載 Ether Link 設置	20-66
20.4.7 上傳 Ether Link 設置	20-68
20.4.8 刪除非同步裝置	20-70
20.4.9 啟動/關閉線上監控模式	20-71
20.4.9.1 啟動監控	20-71
20.4.9.2 監控狀態顯示	20-73
20.4.9.3 關閉監控	20-73
20.4.10 線上啟動/停止 Ether Link (旗標模式)	20-75
20.4.10.1 啟動 Ether Link	20-75
20.4.10.2 停止 Ether Link	20-76
20.4.11 監控表與錯誤記錄	20-78
20.5 NWCONFIG 的管理與應用	20-79
20.5.1 存檔與列印	20-79
20.5.2 整體下載	20-80
20.5.2.1 下載參數說明	20-80
20.5.2.2 下載操作說明	20-80
20.5.3 ISPSOFT 中的繞送應用	20-82

第 21 章 交換資料的備份與還原

21.1 資料備份記憶卡輔助工具	21-2
21.1.1 資料備份記憶卡簡介	21-2
21.1.2 資料備份記憶卡的操作	21-3
21.2 PLC 永久備份設定	21-3
21.3 記憶卡工具 - CARD Utility	21-4
21.3.1 軟體介紹	21-5
21.3.2 備份操作	21-6
21.3.3 還原操作	21-12
21.3.4 命令列模式指令操作說明	21-15

第 22 章 G-Code 編輯器與 E-CAM 編輯器

22.1 G-Code 編輯器	22-2
22.1.1 關於 G-Code	22-2
22.1.2 使用 G-Code 編輯器	22-2
22.1.3 G-Code 的指令結構.....	22-6
22.1.4 G-Code 功能詳細介紹.....	22-8
22.1.4.1 G0 (快速定位)	22-8
22.1.4.2 G1 (直線補間)	22-10
22.1.4.3 G2 (順時針圓弧補間/螺旋補間)	22-13
22.1.4.4 G3 (逆時針圓弧補間/螺旋補間)	22-15
22.1.4.5 G4 (時間延遲)	22-18
22.1.4.6 G17/G18/G19 (平面選擇)	22-19
22.1.4.7 G50 (準確停止)	22-20
22.1.4.8 G51 (圓弧交接)	22-21
22.1.4.9 G52 (圓滑交接)	22-22
22.1.4.10 G90 (絕對模式)	22-24
22.1.4.11 G91 (相對模式)	22-25
22.1.4.12 M 碼.....	22-26
22.1.5 G-Code 範例-三軸同動繪出 Delta LOGO	22-28
22.2 E-CAM 編輯器.....	22-29
22.2.1 關於 E-CAM (電子凸輪)	22-29
22.2.2 電子凸輪意義說明.....	22-29
22.2.3 使用 E-CAM 編輯器	22-30

第 23 章 精靈工具

23.1 定位規劃表	23-2
23.1.1 關於定位規劃表	23-2
23.1.2 設定定位規劃表	23-2
23.1.3 定位規劃表的模擬.....	23-6
23.1.4 下載與上載定位規劃表	23-8
23.1.5 軸控模式-單軸點對點模式	23-9
23.1.6 軸控模式-單軸多段輸出模式	23-10

23.1.7 軸控模式-雙軸直線補間模式	23-11
23.1.8 軸控模式-雙軸圓弧補間模式	23-11
23.2 資料示波器	23-13
23.2.1 關於資料示波器	23-13
23.2.2 開啟資料示波器	23-13
23.2.3 設定取樣參數	23-14
23.2.4 取樣模式.....	23-17
23.2.5 量測	23-19
23.3 資料記錄器	23-23
23.3.1 關於資料記錄器	23-23
23.3.2 開啟資料記錄器	23-24
23.3.3 設定取樣參數	23-26
23.3.4 監看與錄製模式	23-27
23.3.5 量測	23-31
23.4 高速計數器精靈.....	23-34
23.4.1 關於高速計數器精靈.....	23-34
23.4.2 使用高速計數器精靈.....	23-35
23.5 NTC 精靈.....	23-36
23.5.1 關於 NTC 精靈.....	23-36
23.5.2 使用 NTC 精靈.....	23-37

附錄 A USB 連線

A.1 安裝 PLC 的 USB 驅動程式	A-2
A.1.1 在 Windows XP SP3 環境下安裝 USB 驅動程式	A-2
A.1.2 在 Windows 7 環境下安裝 USB 驅動程式	A-6
A.1.3 在 Windows 8.1 環境下安裝 USB 驅動程式	A-11
A.1.4 在 Windows 10 環境下安裝 USB 驅動程式	A-14
A.2 於 COMMGR 中建立 USB 型態的 Driver.....	A-17
A.3 DVP-SX2 系列的 USB 設定	A-19

附錄 B 各系列機種的相關注意事項

B.1 各系列機種的裝置定址.....	B-2
B.1.1 AH/AS 系列機種的裝置種類.....	B-2

B.1.2 AH/AS 系列機種 X、Y、D、L 裝置的定址格式	B-3
B.1.3 AH/AS 系列機種 X/Y 接點狀態的即時存取	B-4
B.1.4 DVPxxMC/AS5xx 機種的裝置種類	B-5
B.1.5 DVPxxMC/AS5xx 機種之裝置的定址格式	B-5
B.1.6 DVPxxMC/AS5xx 機種之停電保持裝置	B-6
B.2 各系列機種的裝置資源	B-6
B.2.1 AHCPU500-EN/AHCPU500-RS2 機種	B-6
B.2.2 AHCPU510-EN/AHCPU510-RS2 機種	B-7
B.2.3 AHCPU520-EN/AHCPU520-RS2 機種	B-7
B.2.4 AHCPU530-EN/AHCPU530-RS2 機種	B-8
B.2.5 AHCPU501-EN/AHCPU501-RS2 機種	B-8
B.2.6 AHCPU511-EN/AHCPU511-RS2 機種	B-9
B.2.7 AHCPU521-EN/AHCPU521-RS2 機種	B-9
B.2.8 AHCPU531-EN/AHCPU531-RS2 機種	B-10
B.2.9 AHCPU560-EN2 機種	B-10
B.2.10 AHxxEMC 系列機種	B-11
B.2.11 AS200/AS300 系列機種	B-11
B.2.12 DVPxxMC/AS5xx 系列機種	B-12
B.3 編譯與上下載時間	B-12
附錄 C 列印管理工具	
C.1 環境簡介	C-2
C.2 操作功能說明	C-3

前言

ISPSOft 為台達新一代的可程式化控制器 (PLC) 開發工具，除了導入 IEC 61131-3 的編程架構外，更以多任務整合的方式來進行專案的管理，無論是單純的小型應用，或是較為複雜的中大型控制系統，ISPSOft 都能提供使用者一個高效且便利的開發環境。

關於 IEC 61131-3

早期當各家 PLC 開發商紛紛建立自己的編程語言時，其各自不同的編程方式，對於使用者而言無疑是一個相當大的負擔；有鑒於此，由國際組織 I.E.C. (International Electro-technical Commission) 所制定與推動的 IEC 61131-3 標準便由此應運而生。其主要的特色在於統整各家的編程特點，並考慮使用者的背景與習慣，而規範了一個可讓各家廠商與使用者共同依循的標準，並可讓符合 IEC61131-3 標準的程式碼具備相當高的可讀性與相容性。以下便是 IEC 61131-3 的概廓與簡述。

- 統整各家的編程方式與使用者習慣而規範了 7 種編程語言，包括階梯圖 (LD)、循序功能圖 (SFC)、功能區塊圖 (FBD)、指令列表 (IL)、結構化語言 (ST) 以及連續功能圖 (CFC) 與 C 語言 (C)。
- 使用變數的觀念，讓使用者可自行定義變數符號來取代 PLC 的裝置名稱，除了大大地提高程式的可讀性之外，更省去了使用者費心配置裝置位址的時間。
- 使用程式組織單元 (POU) 的架構，其特色在於將傳統單一的程式結構，切割成若干個可獨立開發的程序單元，並透過呼叫函式 (Function) 與功能塊 (Function Block) 的方式，讓程式的架構可以更模組化也更便於維護。
- 使用工作 (Task) 的觀念來對各個開發的程式單元進行管理與組織，將程式開發的工作提升至專案管理的層次，讓大規模的程式開發更便於管理。

手冊導覽

關於這本手冊的編寫方向，基本上是以軟體功能的介紹與操作解說做為出發點，並會在其中加入部分相關的觀念介紹，至於不同機種之間的差異性，若未造成軟體操作上的嚴重障礙，原則上僅會稍作提示，而細節的部份則請自行參照各機種的相關操作手冊或應用指令手冊。

章節的編排主要以功能操作為導向，使用者可輕易的由目錄中找到與軟體操作相關的各個主題，如設定通訊參數、於階梯圖中插入接點、線上監控、啟動模擬器...等；若章節的主題有涉及相關的觀念介紹時，原則上都會集中在該章節的前半部，而對於已有一定基礎或僅需了解如何實作的使用者，則可略過該部分的內容，並直接查閱與操作相關的節次即可。此外，部分的內容可能需要搭配其他章節的說明或操作，而此時於內容中亦會附註使用者須參考的節次。以下則為手冊中各章節內容的簡介。

- 第 1 章與第 2 章的內容主要在於軟體系統的介紹，包括軟體安裝、環境介紹與基本設定。
- 第 3 章的內容將介紹如何在 ISPSOft 中對 PLC 系統進行組態配置與參數設定。
- 第 4 章為快速入門，在章節中將會透過一個簡單的應用範例，而先以非 IEC6113-3 的編程架構來逐步帶領使用者迅速的得知如何在 ISPSOft 中建立一個傳統的階梯圖程式，並進行簡單的測試。
- 第 5 章~第 8 章將會正式介紹 IEC61131-3 的相關概念，並說明在 ISPSOft 當中的建構方式。
- 第 9 章將會介紹運動控制與軸的相關概念，並說明在 ISPSOft 當中的使用。

- 第 10 章~第 16 章將逐一介紹階梯圖 (LD)、功能塊圖 (FBD)、指令列表 (IL)、結構化語言 (ST) 及循序功能圖 (SFC)、連續功能圖 (CFC)、C 語言 (C) 等語言的編程概念，並說明如何於 ISPSOft 當中進行實作。
- 第 17 章的主要內容在介紹 ISPSOft 所提供的編輯輔助工具。
- 第 18 章將介紹如何在 ISPSOft 中進行監控、測試與除錯，其中亦包括模擬器的操作。
- 第 19 章的內容主要在介紹 ISPSOft 所提供的密碼與資料保護機制。
- 第 20 章的內容主要是介紹如何在 ISPSOft 當中部署一個網路架構，並藉以建構資料交換的機制。
- 第 21 章將介紹如何透過 ISPSOft 來對 PLC 或專案進行資料的備份與還原。
- 第 22 章將介紹運動控制中常使用到的幾種工具，即在 ISPSOft 中的操作。
- 第 23 章將介紹 ISPSOft 中的幾個輔助工具，不論編程或調試方面均大幅增加便利性。

圖示說明

下表為手冊當中搭配說明所使用的圖示。

圖示	說明
	點擊滑鼠左鍵
	點擊滑鼠右鍵
	快點兩下滑鼠左鍵
	按住滑鼠左鍵後進行拖曳
	使用鍵盤輸入
	操作順序 (當說明中需要特別表達操作順序時會配合此標註，例如  、 )
	配合圖片說明的指示編號
	重點注意事項 (該事項可能涉及設備、財產或人體的損害)

商標聲明

手冊中所提及屬於台達電子以外的產品與商標，其所有權分屬於各公司。

1

第1章 軟體簡介

目錄

1.1 ISPSOft 簡介及系統需求	1-2
1.1.1 特色簡介	1-2
1.1.2 系統需求	1-3
1.1.3 安裝 ISPSOft	1-3
1.1.4 移除 ISPSOft	1-8
1.2 通訊管理員 COMMGR 簡介	1-9
1.2.1 COMMGR 的工作模式	1-9
1.2.2 安裝 COMMGR	1-10
1.2.3 移除 COMMGR	1-13
1.3 ISPSOft PLC 機種支援	1-14
1.3.1 ISPSOft 語言支援列表	1-14
1.3.2 ISPSOft 功能支援列表	1-14

1

1.1 ISPSOft 簡介及系統需求

ISPSOft 為台達新一代的可程式化控制器 (PLC) 開發工具。採用 IEC 61131-3 的編程架構，支援七種編程語言以及豐富的應用指令集。除了具備基本的程式編輯功能之外，ISPSOft 還包含了許多的輔助工具，加上多語系的環境與親切的使用介面，提供給使用者的將會是一個便利且高效的開發環境。

1.1.1 特色簡介

- 採用國際標準 IEC 61131-3 的編程架構，並內建豐富的應用指令集。
- 支援七種語言，包括階梯圖 (LD)、循序功能圖 (SFC)、功能區塊圖 (FBD)、指令列表 (IL)、結構化語言 (ST) 以及連續功能圖 (CFC) 與 C 語言 (C)，且在一個專案中並不限定只可使用一種語言。
- 包含繁體中文、簡體中文、英文...等多語系支援。
- 提供便利的**尋找/取代**功能，其範圍可選擇在目前的工作視窗或是包含整個專案。
- 開放使用者自訂的操作環境設定。
- 專案管理採用清晰且易於操作的 Windows 階層樹狀操作介面。
- 支援專案群組的管理架構，可在同一的專案群組中，進行多個機種的專案開發。
- 提供許多的便利功能，如註解、書籤、程式區段致能/失效、裝置與符號管理、模擬器...等。
- 支援多種連線操作，如線上監控、線上編輯、符號表監控、除錯模式、PLC 操作/設定...等。
- 支援多種專案物件的**匯入/匯出**功能。
- 可直接開啟 WPLSoft 的檔案 (*.dvp)，並將其轉換另存為 ISPSOft 格式的專案檔 (*.isp)。
- 提供多種密碼及資料保護機制。
- 支援新一代的通訊管理員 COMMGR。
- 內建多種組態管理工具
 - 硬體規劃工具 (HWCONFIG)：用於對系統上的硬體架構進行組態與參數管理。
 - 網路規劃工具 (NWCONFIG)：用於管理及配置 PLC 系統的網路架構與資料交換機制。
 - 記憶卡工具 (CARD Utility)：透過管理精靈與記憶卡的操作，可對系統進行備份與還原。
- 針對運動控制提供各種解決方案，包括 PLCopen MC 功能塊、G-code 編輯器、E-CAM 編輯器、定位規劃表等工具。
- 本手冊對 DVPxxMC 機種之內容敘述，僅適用於 DVP15MC、DVP15MC-06、DVP50MC 以及 DVP50MC-06 機種。
- 本手冊對 AS 機種之內容敘述，不包含 AS5xx 機種。
- 本手冊對 AS5xx 機種之內容敘述，僅適用於 AS516E-B、AS524C-B 機種。

1.1.2 系統需求

在開始使用 ISPSOft 之前，請務必確認操作系統是否符合以下的需求。

項目	系統需求	
作業系統	Windows XP / 7 / 8 / 10	
CPU	Pentium 1.5G 以上機種	
記憶體	256MB 以上 (建議使用 512MB 以上)	
磁碟機	硬碟容量：至少須提供 5000MB 以上空間供 ISPSOft 使用	
光碟機	用於透過光碟片來進行軟體安裝的場合 (選配)	
顯示器	解析度：800 × 600 以上 (建議顯示設定：1024 x 768 / 96DPI)	
鍵盤/滑鼠	一般的鍵盤滑鼠或與 Windows 相容的裝置	
印表機	具 Windows 驅動程式的印表機 (選配，用於專案內容的列印)	
RS-232 埠	與 PLC 連線之用	三者擇一即可。但仍須依據主機本身或搭配模組所提供的通訊介面 (*1)
USB	與 PLC 連線之用	
乙太網路	與 PLC 連線之用	
連線軟體	電腦中須安裝 通訊管理員 - COMMGR 的軟體 (*2)	
支援機種	PLC：AH 全系列 / DVP 全系列 (不含 DVP-PM 系列) (*3) / AS 全系列。 交流馬達驅動器：VFD 系列內建 PLC 機種。 文本顯示器：TP 系列內建 PLC 機種。	

*1. ISPSOft 支援多種與 PLC 的連線方式，進行連線前請先確認使用機種所提供的通訊埠口及所支援的連線模式。

*2. 關於通訊管理員 COMMGR 的相關介紹請參閱第 1.2 節。

*3. AH-PM 系列與 AH-MC 系列模組的專案尚須搭配 PMSOft V2.05 版以上的軟體來進行開發。

*4. 以上所提及的功能與規格僅適用於 ISPSOft V3.00 或更高的版本，先前的版本可能不具備完整功能。

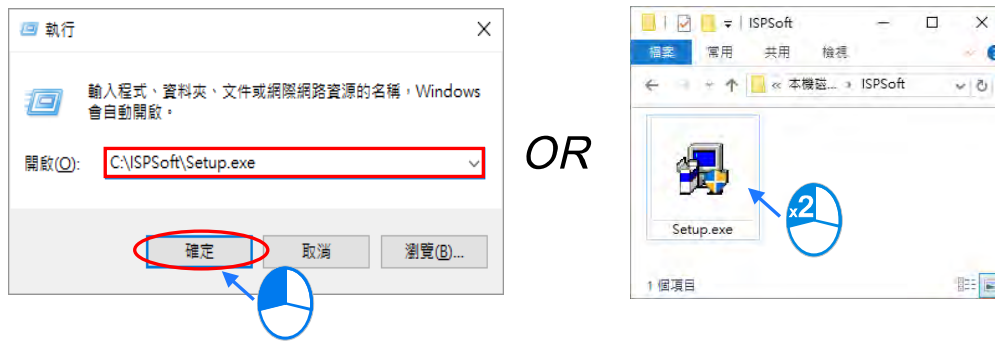
1.1.3 安裝 ISPSOft

當電腦中已有安裝舊版的 ISPSOft 時，執行 ISPSOft 安裝程式時，系統會自動偵測有無安裝舊的 ISPSOft 版本，並提示要移除舊的 ISPSOft，移除完成之後會自動接續安裝新的 ISPSOft。

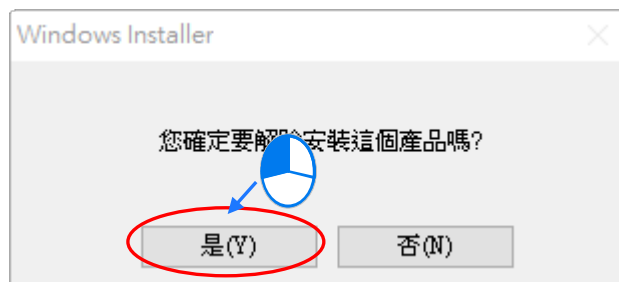
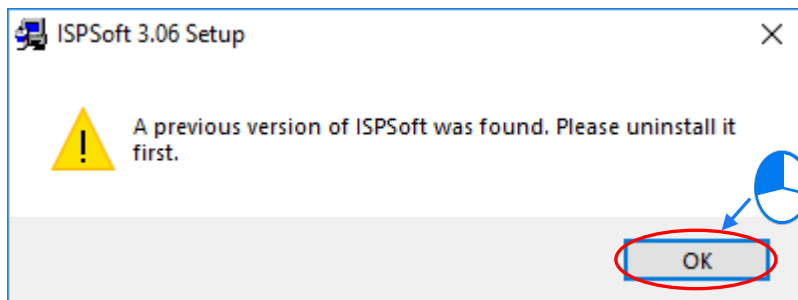
- (1) 啟動電腦並進入作業系統，且須以具備系統管理員的權限登入才可進行軟體安裝。
- (2) 將 ISPSOft 光碟片放入光碟機中或從台達官方網站 <http://www.deltaww.com/> 下載 ISPSOft 的安裝程式 (從網路下載的安裝程式必須經過解壓縮後才可進行安裝。)。

1

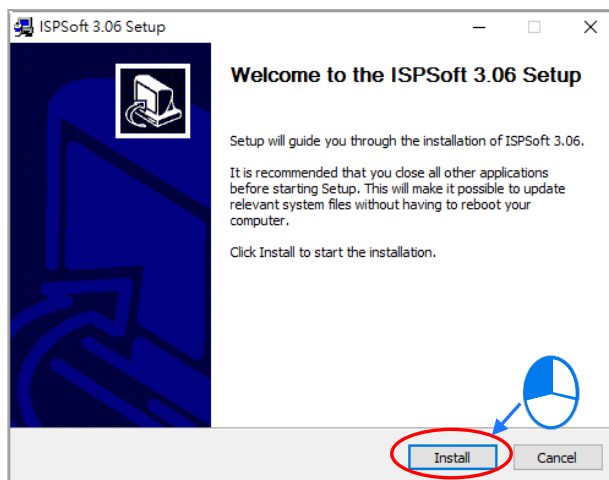
- (3) 於「開始」→「執行」的視窗中指定安裝檔路徑後按下「確定」，或是直接於安裝檔的圖示上雙擊滑鼠左鍵，以執行安裝程式。



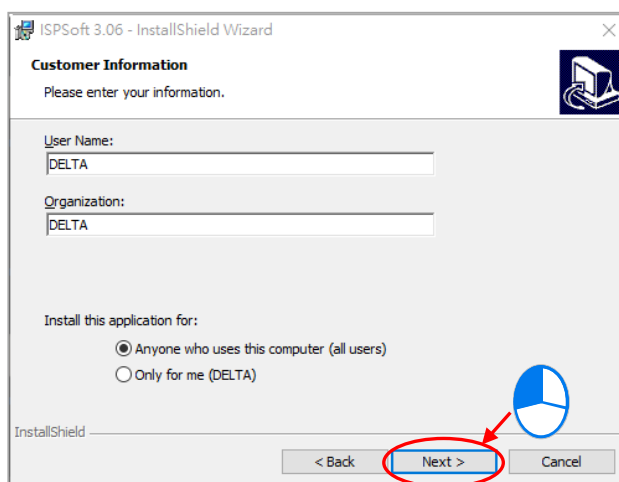
- (4) 如果系統內有安裝舊版本 ISPSoft，才會出現此畫面提示要移除舊版本的 ISPSoft，此時按下「OK」進行下一步，接著按下「是 (Y)」，系統會開始移除舊版本 ISPSoft。



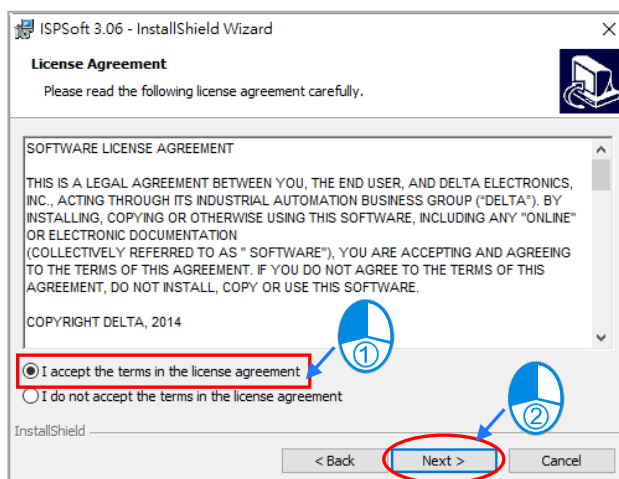
- (5) 待安裝視窗出現後，按下「Install」進行下一步。



- (6) 輸入使用者的相關資訊後，按下「Next」進行後續的安裝工作。

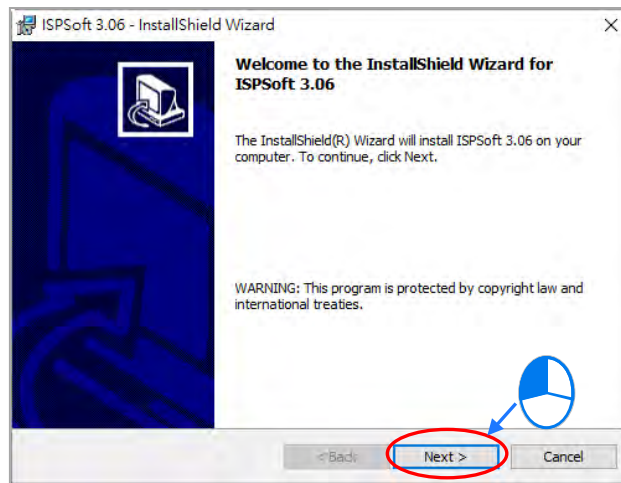


- (7) 選取「I accept the terms in the license agreement」同意安裝，完成後按下「Next」進行下一步。

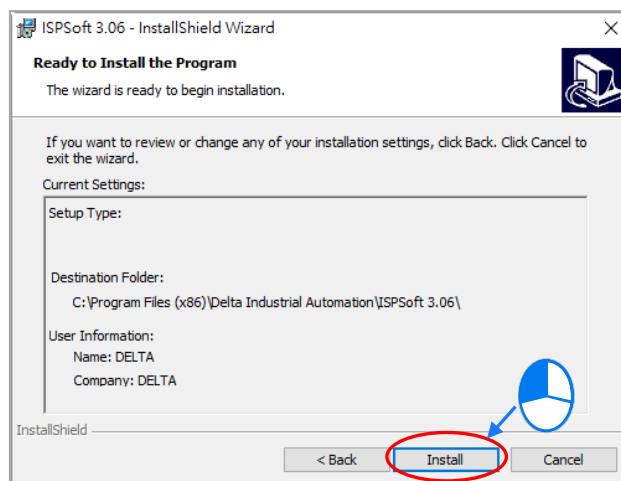


1

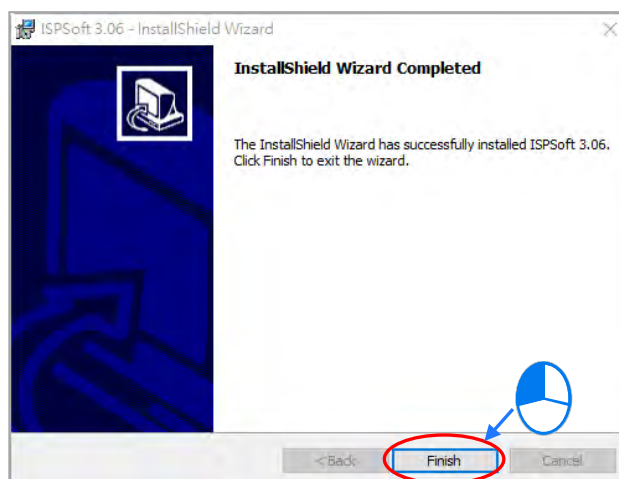
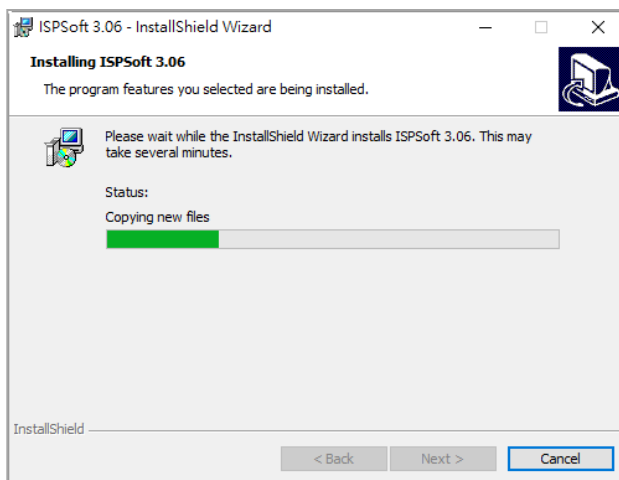
(8) 再次按下「**Next**」進行下一步。



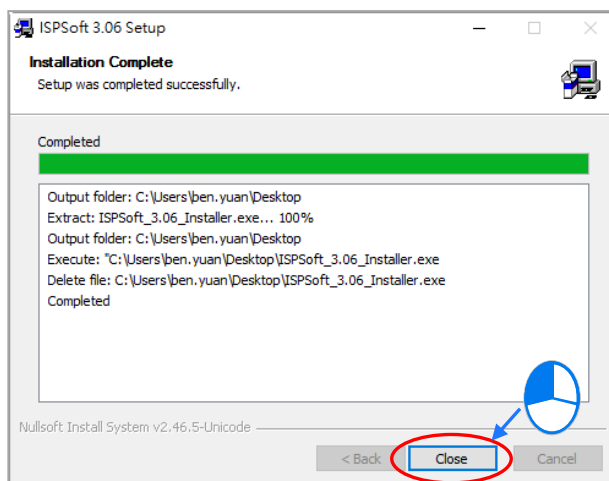
(9) 確認安裝資訊後，按下「**Install**」進行下一步。



(10) 等待安裝進度條完成時，按下「**Finish**」即可進行下一步。



(11) 完成安裝後，在桌面及開始功能表中會自動建立程式的執行捷徑，按下「**Close**」即可結束安裝。



1

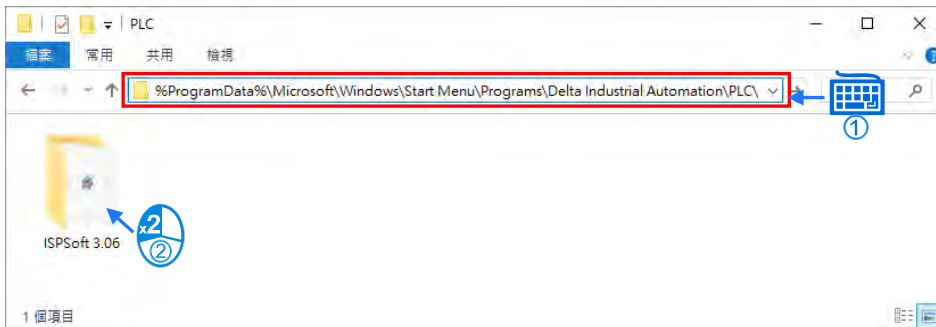
1.1.4 移除 ISPSoft

(1) 正常的情況下只要點擊 ISPSoft Uninstall 或到控制台的程式和功能項目，就可以移除 ISPSoft；在找不到 ISPSoft Uninstall 時，可以透過下面兩種方式操作：

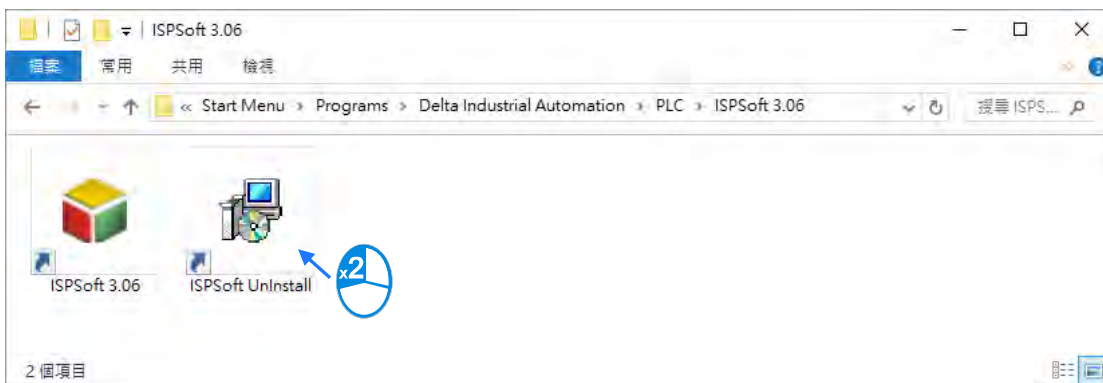
方法一：從選單的 ISPSoft x.xx 點擊右鍵然後更多開啟檔案位置。

方法二：從檔案總管視窗的位址列輸入

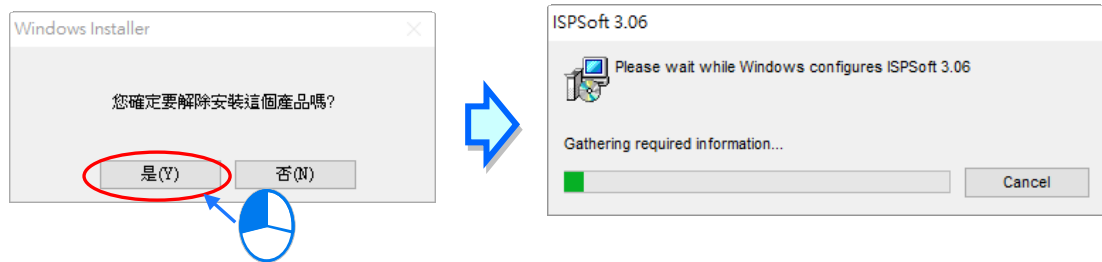
`%ProgramData%\Microsoft\Windows\Start Menu\Programs\Delta Industrial Automation\PLC\`
按下 **Enter** 之後，在 ISPSoft x.xx 資料夾雙擊滑鼠左鍵。



(2) 在「ISPSoft UnInstall」雙擊滑鼠左鍵，移除 ISPSoft。



- (3) 待 ISPSOft 移除程式視窗出現之後點選「是」確認動作後即可開始進行移除，移除完成之後視窗會自動關閉。

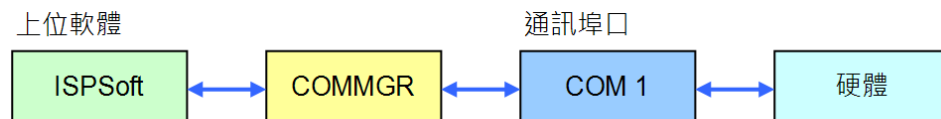


1.2 通訊管理員 COMMGR 簡介

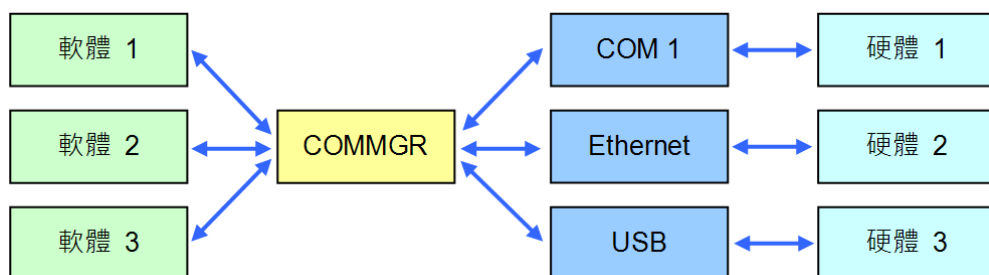
COMMGR 是台達電子的通訊管理工具，其主要的功能在於扮演台達軟體與硬體之間的通訊橋樑，而透過 COMMGR 的管理，連線的工作亦將變得更加便利也更有效率。

1.2.1 COMMGR 的工作模式

對於所有須要設定的連線參數，我們都可以事先將它們建立在 COMMGR 的管理列表中，而這些事先建立好的連線參數我們便稱之為 **Driver**。依實際的使用狀況，我們可以直接在 COMMGR 當中設定每組 **Driver** 的連線參數，而當該組 **Driver** 的狀態為「OK」時，實際的連線通道便會自動建立，接下來只須在 ISPSOft 中指定欲使用的 **Driver** 即可進行連線。

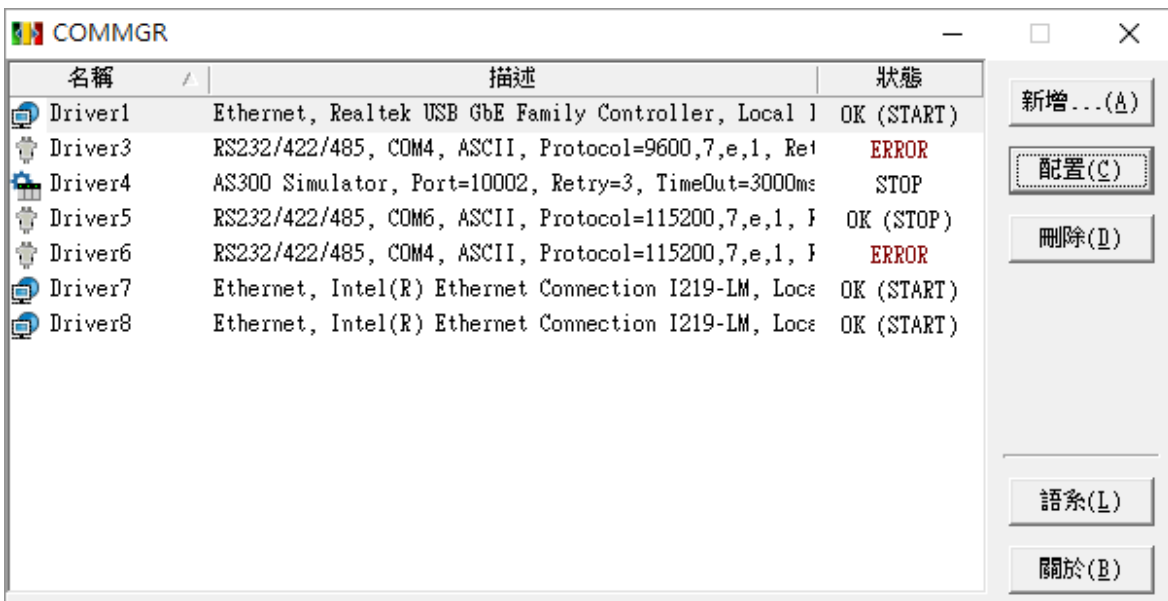


除了 ISPSOft 之外，其他透過 COMMGR 來進行連線的軟體也可在同一時間一起運作，COMMGR 會主動管理所有的通訊命令，並可讓各軟體與硬體之間保持最佳的連線狀態。



1

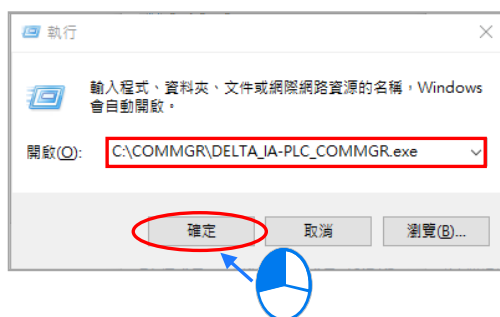
下圖為 COMMGR 的主畫面及管理列表，其中「名稱」欄位可由使用者自行設定，「描述」欄位則會顯示各組 Driver 的相關參數，而「狀態」則顯示該 Driver 目前的狀態。



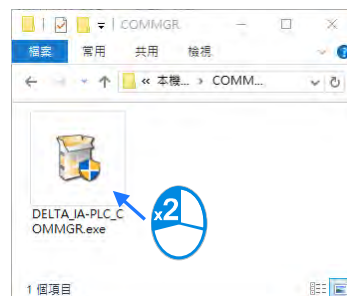
1.2.2 安裝 COMMGR

COMMGR 為獨立於 ISPSOft 的工具軟體，因此使用者必須另外對其進行安裝，而當電腦中已有安裝舊版的 COMMGR 時，執行 COMMGR 安裝程式，系統會自動偵測有無安裝舊的 COMMGR 版本，並提示要移除舊的 COMMGR，移除完成之後會自動接續安裝新的 COMMGR。

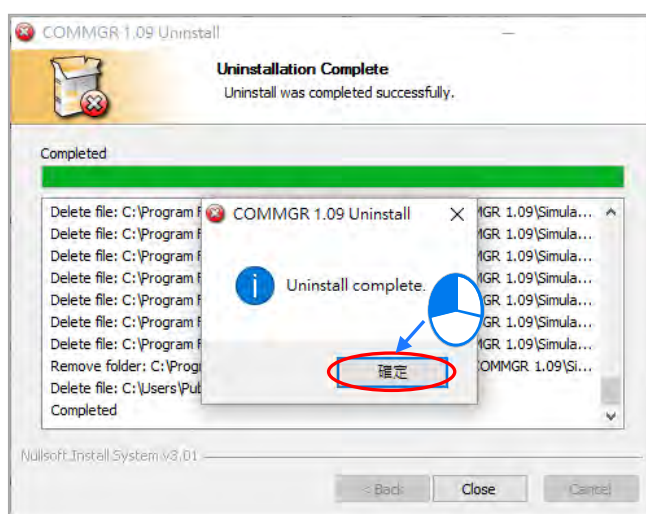
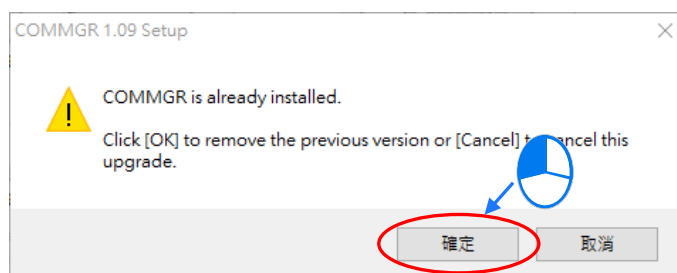
- (1) 啟動電腦並進入作業系統，且須以具備系統管理員的權限登入才可進行軟體安裝。
- (2) 將含 COMMGR 安裝檔的光碟片放入光碟機或從台達網站 <http://www.deltaww.com/> 下載安裝程式（從網路下載的安裝程式必須經過解壓縮後才可進行安裝）。
- (3) 於「開始」→「執行」的視窗中指定安裝檔路徑後按下「確定」，或是直接於安裝檔的圖示上雙擊滑鼠左鍵，以執行安裝程式。



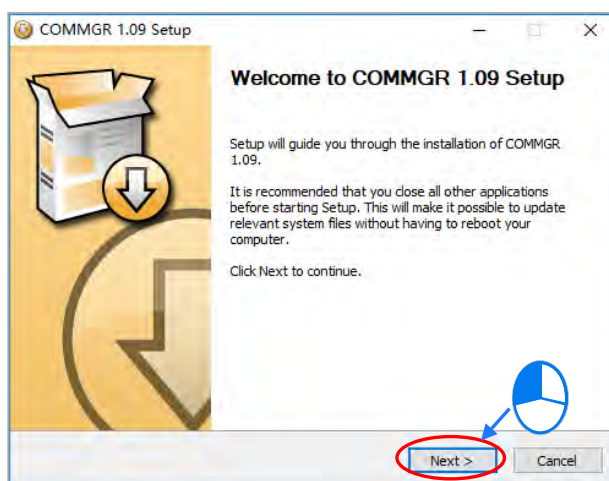
OR



- (4) 如果系統內有安裝舊版本 COMMGR，才會出現此畫面提示要移除舊版本的 COMMGR，此時按下「確定」系統會開始移除舊版本 COMMGR，等待移除完成視窗出現後按下「確定」。

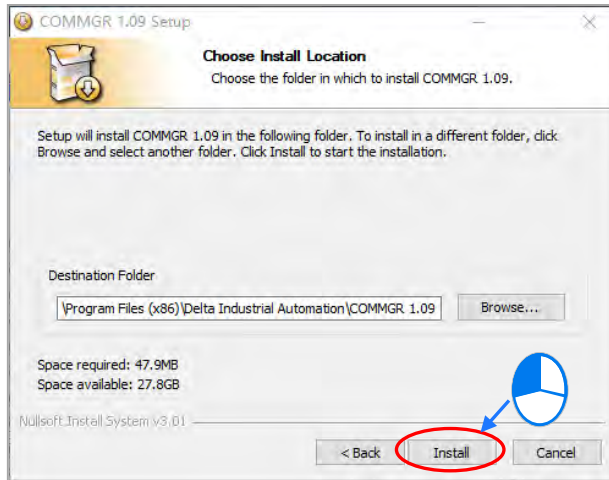


- (5) 待安裝視窗出現後，按下「Next」。

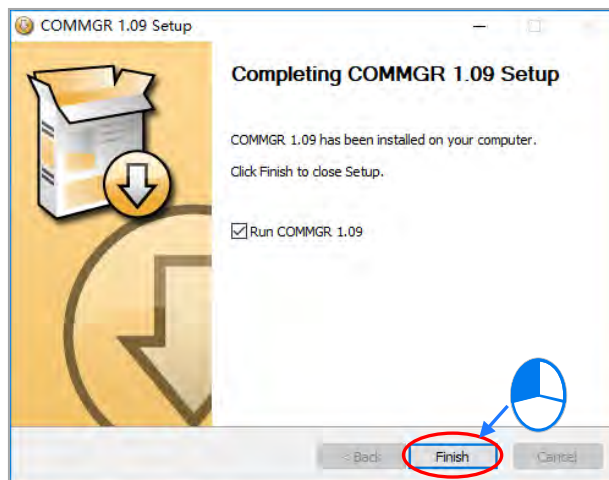
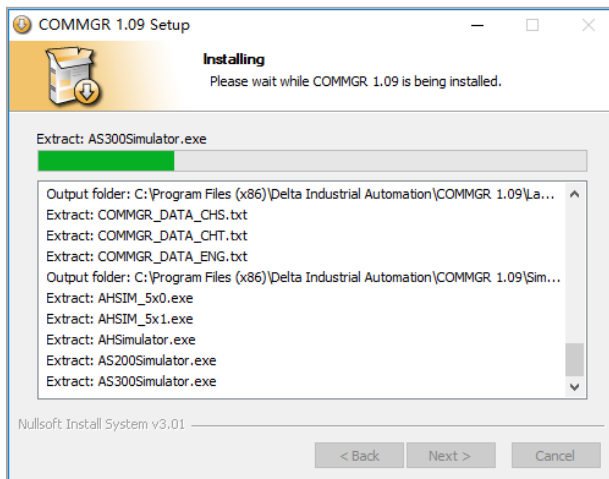


1

(6) 建議使用預設路徑安裝，輸入完成安裝路徑後，按下「**Install**」進行後續的安裝工作。



(7) 完成安裝後，在開始功能表中便會自動建立程式的執行捷徑，而按下「**Finish**」後即可結束安裝。

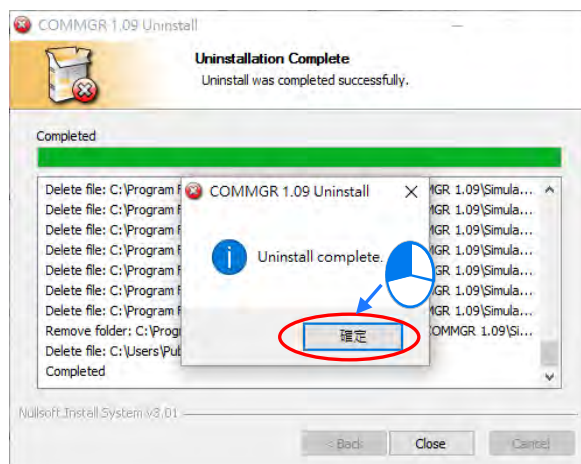
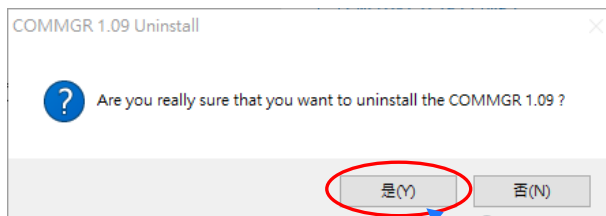


1.2.3 移除 COMMGR

- (1) 進入 Windows 設定中的「應用程式與功能」，選取「**COMMGR x.xx**」後按下「解除安裝」。



- (2) 按下「是 (Y)」系統會開始移除 COMMGR，等待移除完成視窗出現後按下「確定」，即可完成移除。



1

1.3 ISPSOft PLC 機種支援

ISPSOft 支援台達多款 PLC 設備，提供的功能也非常的多，為了方便使用者查詢，這個章節將會介紹 ISPSOft 對於 AS、AH5x0、AH5x1、AH560、AHxxEMC、DVP、DVPxxMC、AS5xx，這八種系列的 PLC 機種，所支援的功能做一個整理。

1.3.1 ISPSOft 語言支援列表

ISPSOft 支援七種語言，包括階梯圖 (LD)、循序功能圖 (SFC)、功能區塊圖 (FBD)、指令列表 (IL)、結構化語言 (ST) 以及連續功能圖 (CFC) 與 C 語言 (C)，機種與其語言支援列表如下所示。

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
階梯圖 (LD)	V	V	V	V	V	V	V
功能塊圖 (FBD)		V	V	V			
結構化語言 (ST)	V	V	V	V	V	只支援 ES3	V
循序功能圖 (SFC)	V	V	V	V	V	V	
連續功能圖 (CFC)	V				V	只支援 ES3	
C 語言 (C)	V						

1.3.2 ISPSOft 功能支援列表

ISPSOft 支援的功能非常的多，下表針對較大的功能項目做機種支援的整理。

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
運動模組		V	V	V	V		
HWCONFIG	V	V	V	V	V	只支援 ES3	V
HWCONFIG 列印功能	V	V	V	V	V		
訂購清單	V	V	V	V	V		
自訂函式庫	V	V	V	V	V	V	
台達函式庫	V	V	V	V	V	V	V

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
自定義	V	V	V	V	V	V	V
軸					V		V
變更 PLC 機種	V	V	V	V	V	V	V
裝置註解表	V	V	V	V	V	V	
裝置使用狀況	V	V	V	V	V	V	
暫存器編輯	V	V	V	V	V	V	
裝置狀態編輯	V	V	V	V	V	V	
檔案暫存器 編輯	V					只支援 SA/SX/SC/EH/EH2/ SV/EH2-L/ES2/ ES2-E/EX2/SX2/SA2 /EH3/SV2/EH3-L/ ES3	
步序定位	V	V	V	V	V	V	
程式比對	V	V	V	V	V	V	
線上編輯	V	V	V	V	V	只支援 EH3/EH3-L/SV2/ES2 /ES2-E/EX2/SA2/ SX2/MC/SS2	
除錯工具列	V	V	V	V	V	V	
系統紀錄	V	V	V	V	V	只支援 ES3	V
3D 座標圖					V		
程式識別碼	V	V	V	V	V	V	V
專案密碼	V	V	V	V	V	V	V
POU 密碼	V	V	V	V	V	V	V
副程式密碼						不支援 ES3	
NWCONFIG		V				不支援 ES3	
資料備份記憶 卡輔助工具						只支援 EH3/EH2	

1

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
PLC 永久 備份設定						只支援 EH2/EH2-L/EH3/ EH3-L/SV/SV2	
CARD Utility	V	V	V	V	V	只支援 ES3	只支援 DVPxxMC
G-Code 編輯器							V
E-CAM 編輯器	V				V		V
定位規劃表	V					只支援 ES3	
資料示波器	V		V		V	只支援 ES3	
資料記錄器	V		V			只支援 ES3	
高速計數器 精靈	V					只支援 ES3	
AIO 精靈						不支援 ES3	
溫度監控						不支援 ES3	
秤重模組						不支援 ES3	
擴充模組輔助 設計精靈						不支援 ES3	
擴充模組 監控精靈						不支援 ES3	
NTC 精靈						只支援 SX、SE、SS2、SX2、 SA2、SV2	只支援 DVPxxMC

2

第2章 啟動與設定

目錄

2.1 啟始導覽與環境介紹.....	2-3
2.1.1 進入 ISPSOft 的第一步.....	2-3
2.1.2 視窗標題與狀態列.....	2-8
2.1.3 功能工具列.....	2-9
2.1.4 圖示工具列.....	2-11
2.1.5 專案管理區與訊息顯示區.....	2-12
2.1.6 編輯工作區.....	2-14
2.2 ISPSOft 的專案架構.....	2-17
2.2.1 單機專案與群組專案.....	2-17
2.2.2 整合運動模組專案.....	2-18
2.2.3 ISPSOft 的單機專案管理.....	2-19
2.2.4 ISPSOft 的群組專案管理.....	2-24
2.2.5 ISPSOft 的運動模組專案管理.....	2-31
2.3 ISPSOft 中的基本設定.....	2-36
2.3.1 系統及環境設定.....	2-36
2.3.2 使用者編輯格式的匯入與匯出.....	2-42
2.3.3 匯出程式到文字格式.....	2-43
2.4 通訊設定.....	2-44
2.4.1 通訊管理員 COMMGR 的啟動與關閉.....	2-44
2.4.2 COMMGR 的驅動管理.....	2-46
2.4.3 建立連線通道 - 驅動的建立.....	2-46
2.4.3.1 RS232/422/485 通訊參數設定.....	2-48
2.4.3.2 USB (Virtual COM) 通訊參數設定.....	2-49
2.4.3.3 DirectLink (USB) 通訊參數設定.....	2-50
2.4.3.4 Ethernet 通訊參數設定.....	2-50
2.4.3.5 DirectLink (Ethernet) 通訊參數設定.....	2-51
2.4.3.6 DVP Simulator 通訊參數設定.....	2-52
2.4.3.7 AH5x0 與 AH5x1 Simulator 通訊參數設定.....	2-53
2.4.3.8 AS Simulator 通訊參數設定.....	2-53

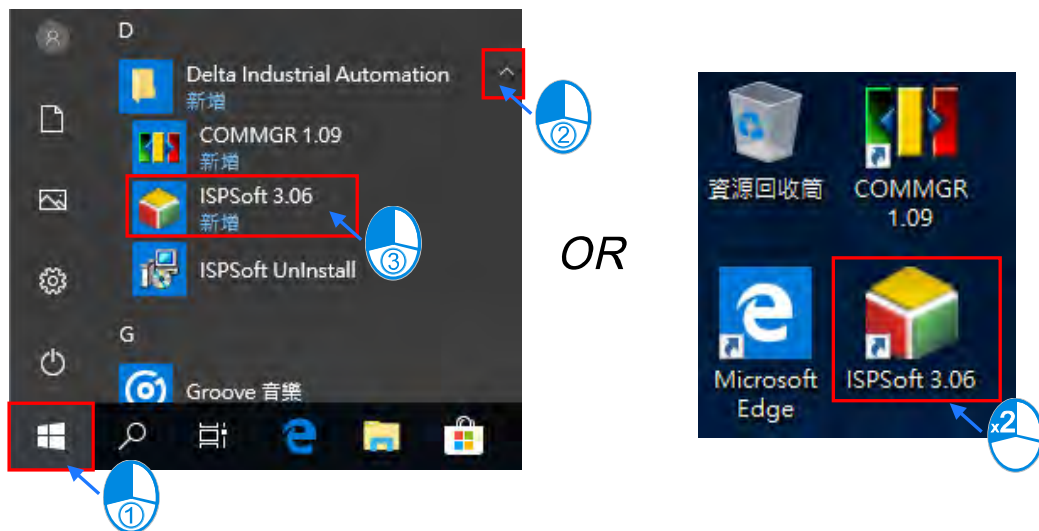
2.4.4	建立連線通道 - 驅動的啟動與停止	2-54
2.4.5	建立連線通道 - 驅動的修改與刪除	2-55
2.4.6	建立連線通道 - 模擬器的面板操作	2-56
2.4.7	建立 ISPSOFT 與 COMMGR 之間的連結	2-59
2.4.8	連接主機與通訊埠口	2-62
2.4.9	實際連線測試	2-65

2.1 啟始導覽與環境介紹

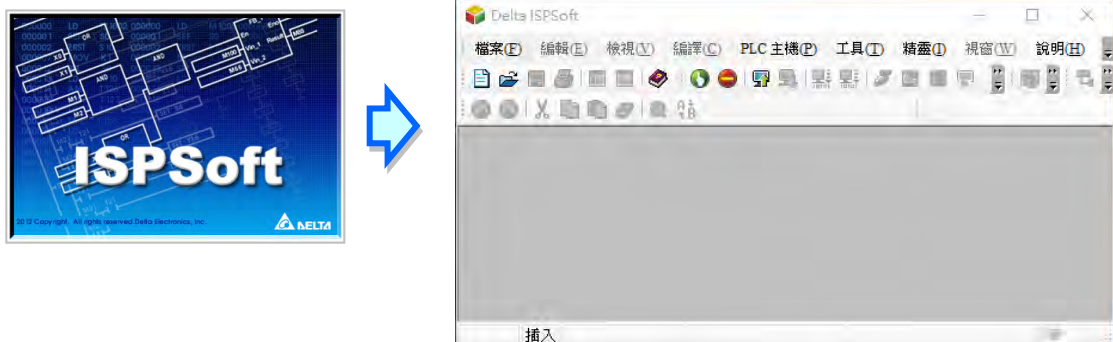
2.1.1 進入 ISPSOft 的第一步


成功安裝 ISPSOft 之後，系統便會自動在桌面及程式集中建立捷徑，點擊該捷徑即可啟動 ISPSOft。ISPSOft 也支援同時開啟多個軟體視窗，只要依據相同步驟，另外再啟動 ISPSOft 即可。

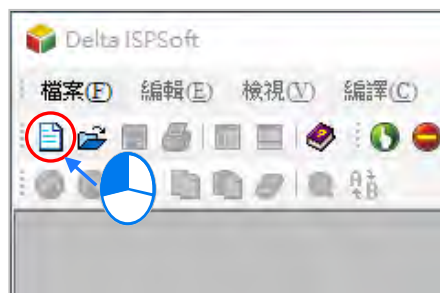
2



待歡迎畫面結束後便可正式進入 ISPSOft，而剛進入時，視窗中僅有一些基本的功能可供使用。

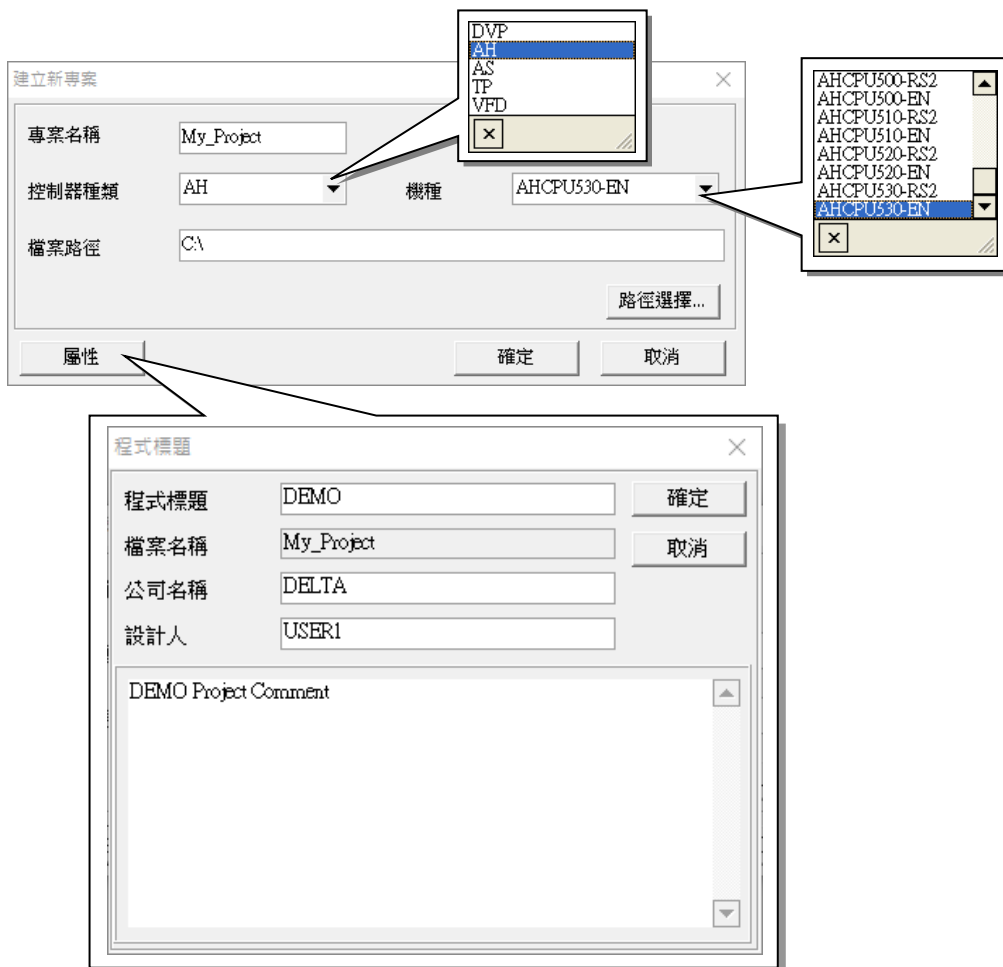




直接點擊圖示工具列中的  圖示即可建立一個新的專案。

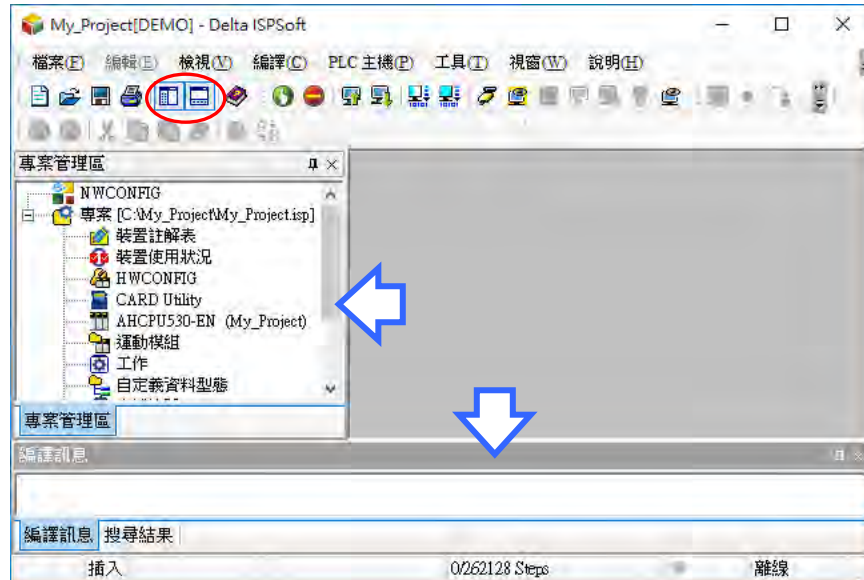


於設定視窗中依序輸入「專案名稱」及「檔案路徑」，之後於「控制器種類」及「機種」的下拉選單中選擇機種型式；而點擊「屬性」則可為此專案加入相關的資訊及描述。完成後按下「確定」即可。

2



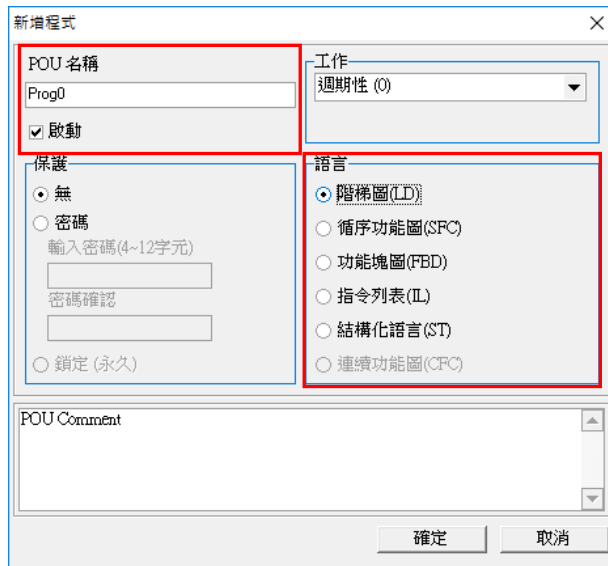
成功建立專案後，畫面左側會開啟一個「專案管理區」，並以階層樹狀圖的方式列出所有物件；若未出現該區域，請點擊功能工具列的 **檢視 (V) > 工作區 (W)** 或圖示工具列的  圖示即可。若點擊 **檢視 (V) > 輸出視窗 (M)** 或圖示工具列的  圖示則可開啟「訊息顯示區」。



剛建立專案時，主要的工作區因尚未進行任何編輯而處於空白狀態；若要開始建立 PLC 程式時，可於「專案管理區」的「程式」項目上點擊滑鼠右鍵，之後於選單點選 **新增 (N)**。



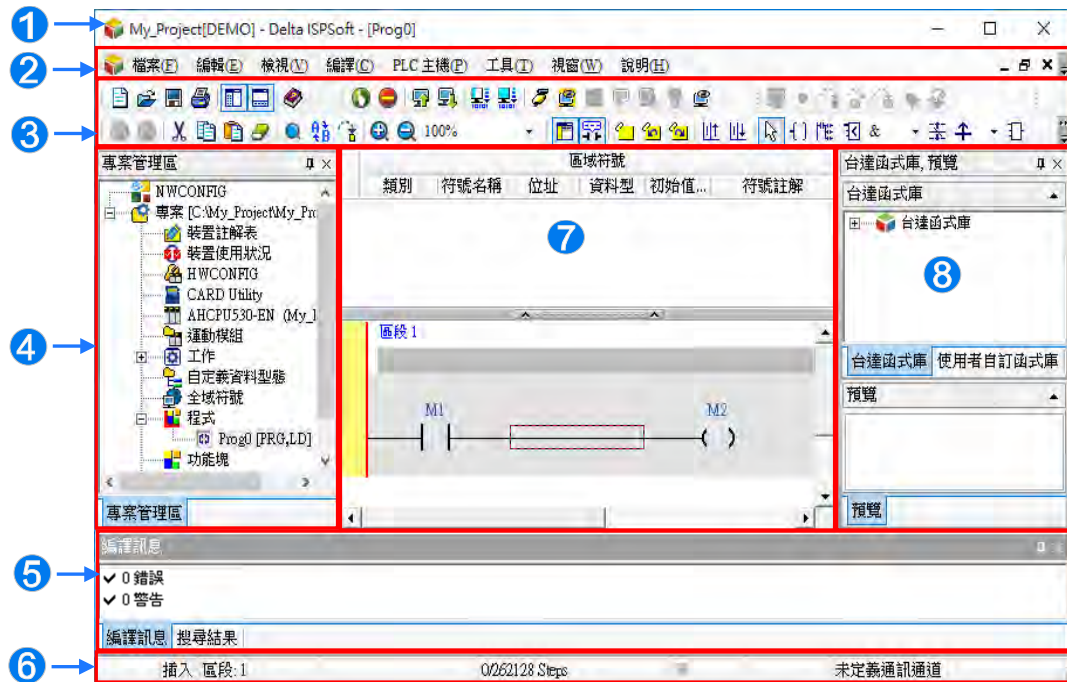
於新增程式視窗中，使用者可在「**POU 名稱**」欄位中輸入該程式的命名，之後再於「**語言**」欄位中選擇欲使用的編程語言；至於其他參數則可先延用預設值，其意義將於**第 5.4 節**的內容當中另行介紹。



成功新增一個程式單元後，於「專案管理區」的「程式」項目下便會多出一個物件，而在主要的工作區中也會自動開啟一個程式編輯視窗。




下圖便是一個完整的 ISPSOft 工作畫面。



- ❶ 視窗標題：顯示目前所編輯的專案名稱與程式標題。
- ❷ 功能工具列：共有 8 類的功能操作選項。
- ❸ 圖示工具列：共有 5 類的圖示工具列可供操作。
- ❹ 專案管理區：階層樹狀結構的專案物件管理介面。
- ❺ 訊息顯示區：顯示編譯結果及專案搜尋的相關訊息。
- ❻ 狀態列：顯示目前編輯或連線狀態的相關資訊。
- ❼ 編輯工作區：顯示程式編輯區、符號表及監控表...等工作視窗。
- ❽ 函式庫區：列出台達函式庫與使用者自訂函式庫的內容。

在使用 ISPSoft 的過程中，使用者可透過功能工具列的「說明 (H)」項目來啟動各種的線上輔助。

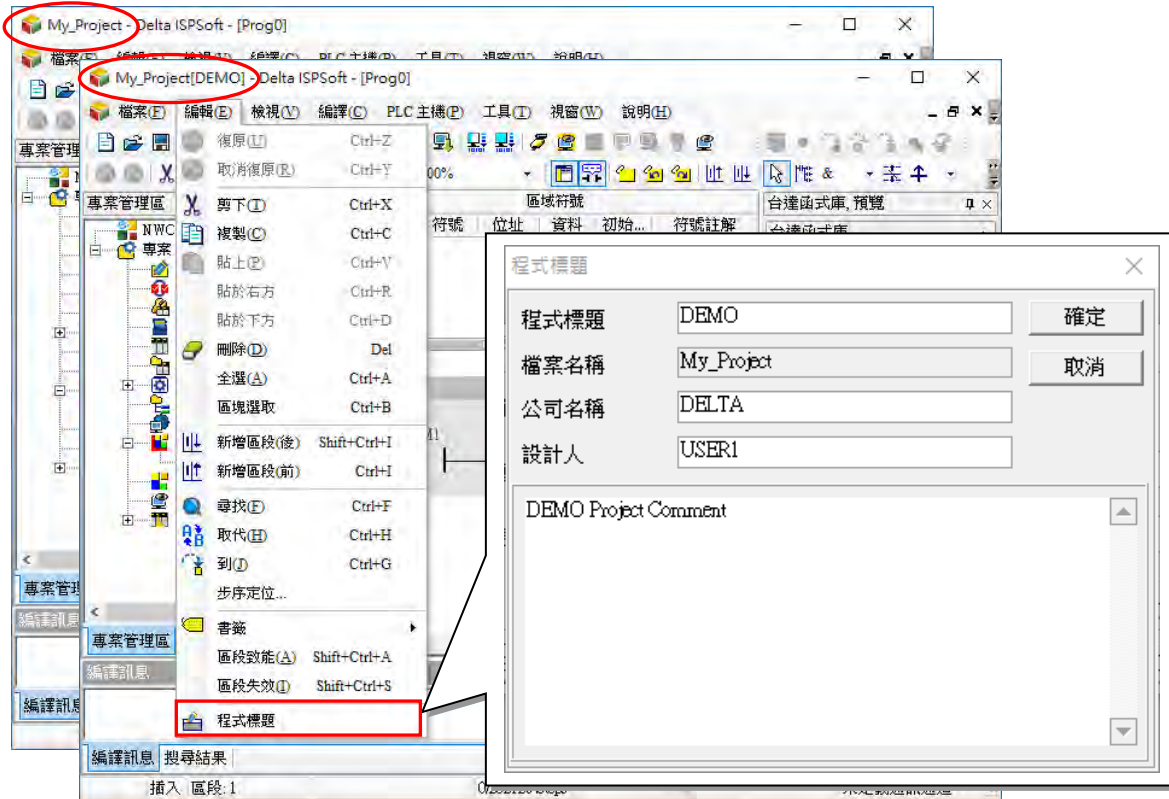


操作項	功能說明
關於	查詢軟體版本及發行日期等相關資訊。
輔助編輯	輔助編輯工具如通訊格式<=>參數碼。
ISPSoft 使用索引	查詢軟體的使用說明。(亦可透過圖示工具列的  圖示來開啟)
PLC 指令及特殊暫存器索引	查詢各種指令及暫存器的說明。而當於程式編輯區中點選某一指令後，按下鍵盤的【F1】鍵便可直接開啟該指令的說明頁面。
版本變更紀錄	列出此版本軟體的新增或修改功能。

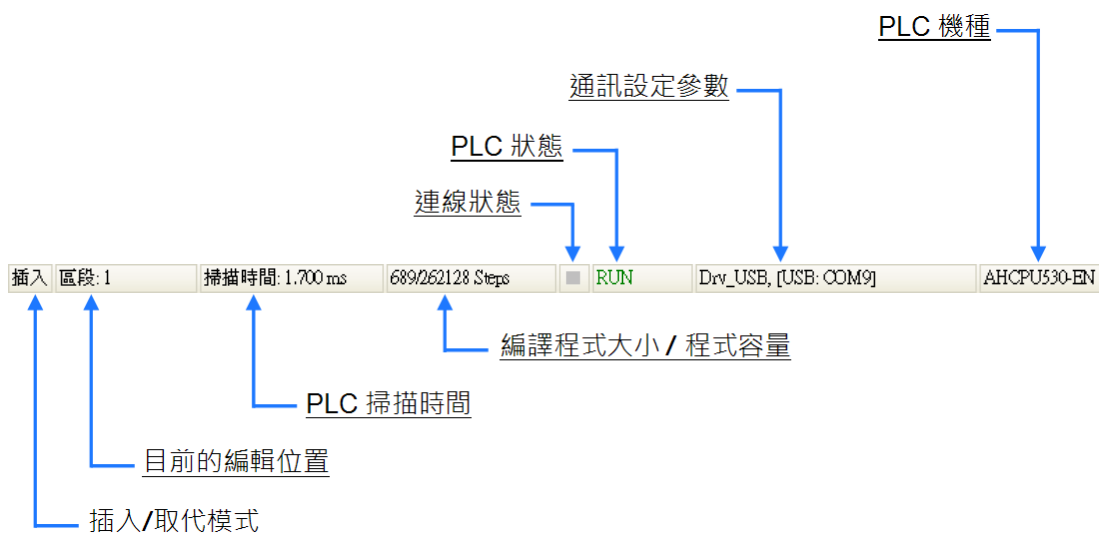
2.1.2 視窗標題與狀態列

成功建立專案之後，ISPSOft 的視窗標題便會顯示該專案的名稱，其格式為 **專案名稱[程式標題]**，而當沒有設定「程式標題」時，視窗標題則只會顯示「專案名稱」；當要檢視「程式標題」等資訊時，只要在功能工具列上依序點選 **編輯 (E) > 程式標題** 即可。

2

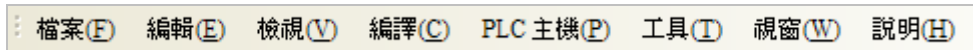


視窗下方的狀態列主要在於顯示目前的各種工作狀態，包括編輯模式透過鍵盤上的 **Insert** 按鍵可以切換（插入/取代）、目前編輯位置、PLC 掃描時間、程式大小/容量、連線狀態、PLC 狀態、通訊設定參數以及 PLC 機種等資訊。



2.1.3 功能工具列

ISPSoft 的功能工具列共有 8 類功能項，且依據目前所進行的編輯工作與選用的機種，各類功能項的下層項目也會有所差異，在此先稍做瀏覽介紹，詳細的操作方式則將於之後的章節當中陸續介紹。



- **檔案**：主要功能為基本的專案存取操作。



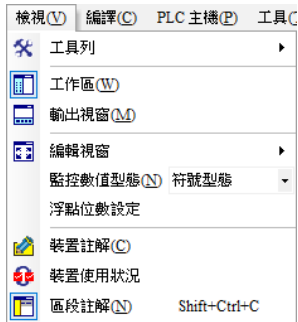
- **編輯**：主要提供進行編輯工作時所需的操作功能。



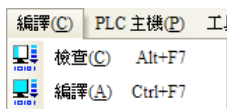
2

2

- **檢視**：主要功能為檢視專案中的資訊以及設定工作環境的顯示方式。



- **編譯**：主要功能為檢查程式語法或將專案程式編譯為可讓 PLC 運行的執行碼。



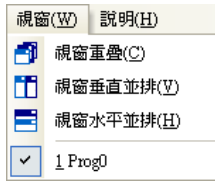
- **PLC 主機**：主要為透過 ISPSOft 對 PLC 主機進行的連線操作或設定功能。



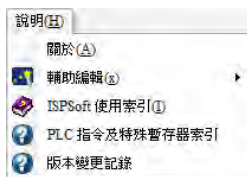
- **工具**：主要為 ISPSOft 所提供的一些便利功能。



- **視窗**：主要功能為管理編輯工作區中的各個視窗。

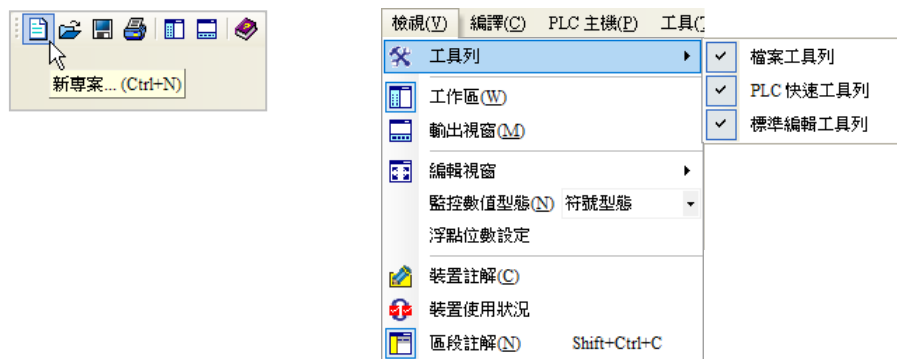


- **說明**：主要為 ISPSOft 所提供的一些輔助功能。



2.1.4 圖示工具列

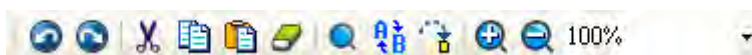
ISPSOft 中共有 5 類的圖示工具列，而且同樣會依據目前的編輯內容與選擇機種而有不同的顯示。當滑鼠移至圖示的上方並稍作停留時，畫面便會顯示該圖示的功能與快捷鍵提示；而在功能工具列的 **檢視 (V) > 工具列** 中，使用者可自行勾選所要顯示或隱藏的圖示工具列。



- **檔案工具列**：主要為專案操作的相關功能。



- **標準編輯工具列**：主要為與編輯工作相關的操作選項。

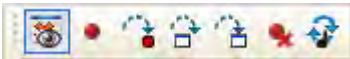


- **PLC 快速工具列**：主要為與 PLC 操作相關的功能。



- **除錯工具列**：主要為除錯功能的相關操作選項，且不同機種的操作圖示也會有所差異。


*.目前 DVPxxMC/AS5xx 不支援除錯模式。


AH/AS 機種：


DVP 機種：


2


- **編程工具列**：主要為編輯程式時所需的各項功能，且不同的編輯語言會有不同的操作圖示。


LD：

FBD：

IL：



ST：

SFC：



CFC：

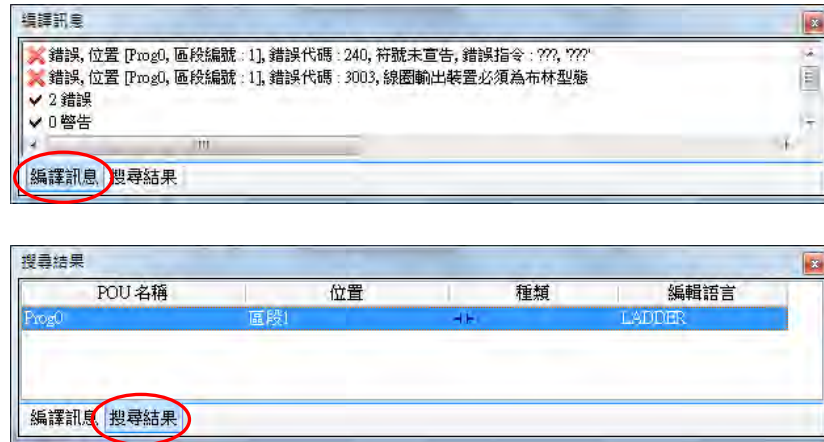
2.1.5 專案管理區與訊息顯示區



專案管理區的主要內容為與專案開發相關的一些物件，並以階層樹狀結構做為其管理介面；內容包括由系統提供的組態管理工具、裝置資訊與應用指令，同時也包含使用者自行建立的變數符號、程式項目、監控表...等，且隨著選用機種的不同，其所顯示的項目，如工具、指令...等，亦會有所不同。

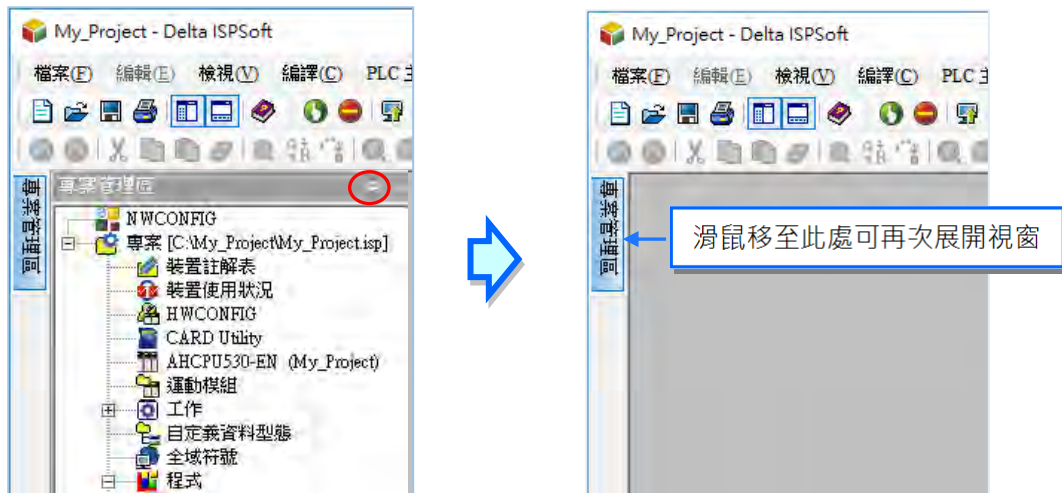
預設的**專案管理區**會固定在 ISPSOft 視窗的左側，點擊此區右上角的  圖示即可將其關閉，而透過切換圖示工具列上的  圖示或是功能工具列的 **檢視 (V) > 工作區 (W)** 項目則可開啟或關閉**專案管理區**的顯示。當在某個專案物件上按下滑鼠右鍵時，畫面便會出現對應的快捷選單，利用選單上的功能便可進行相關的物件操作。

訊息顯示區主要有**編譯訊息**與**搜尋結果**兩個頁面，而兩者之間則可透過**訊息顯示區**下方的標籤來進行切換。**編譯訊息**頁面主要是顯示進行程式編譯或語法檢查時的結果或提示訊息，而**搜尋結果**頁面則是用來條列執行專案搜尋後的搜尋結果。

訊息顯示區的預設位置是固定在 ISPSOft 視窗的下方，而切換圖示工具列上的  圖示，或是切換功能工具列的 **檢視 (V)** > **輸出視窗 (M)** 項目便可開啟或關閉訊息顯示區的顯示，而點擊此區右上角的  圖示亦可將其關閉。

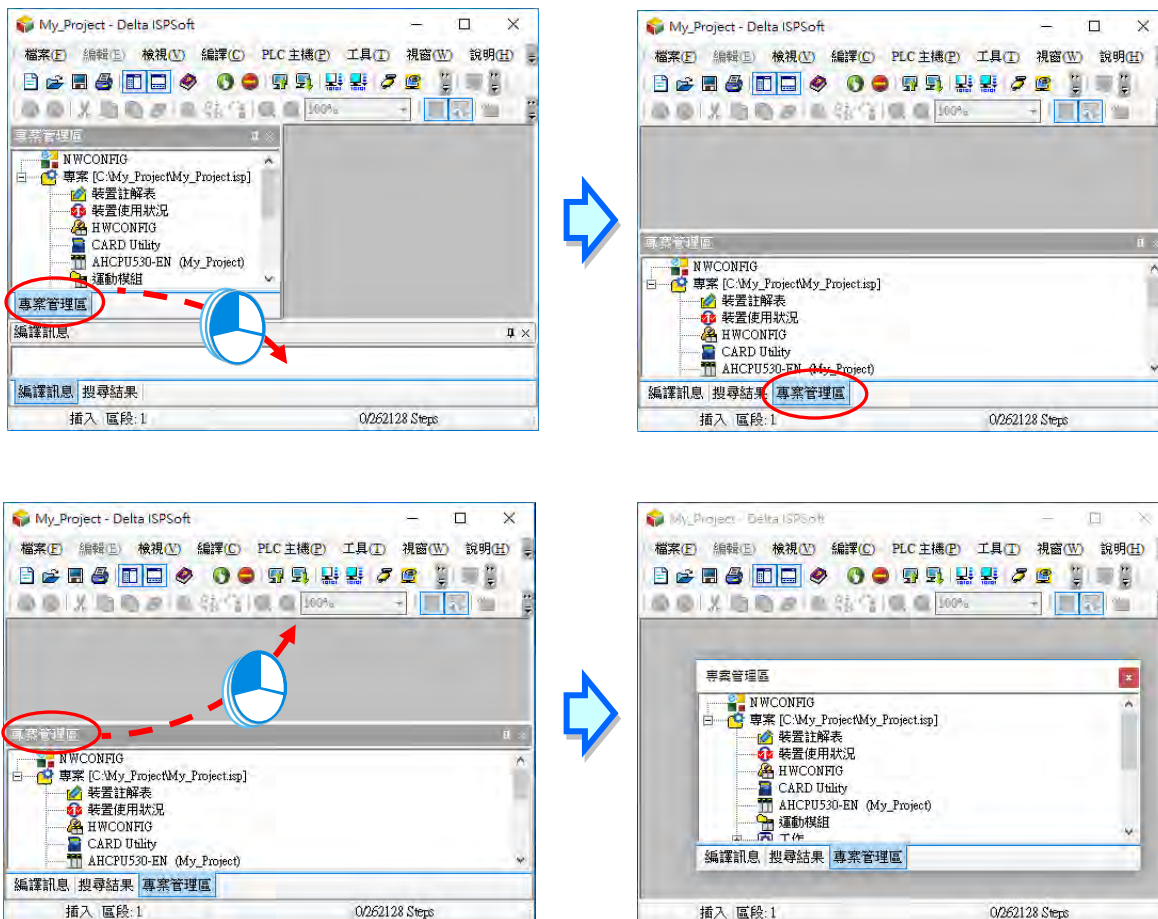


不論是專案管理區或訊息顯示區，切換兩者視窗右上角的  圖示便可改變視窗的顯示方式；當圖示切換成  狀態時便可將該視窗設為自動隱藏，只要鼠標離開該區域一段時間之後，該區域的視窗便會自動收藏至邊緣，而當將滑鼠移至收藏邊緣上的標籤時，鼠標所指的視窗便會再次展開。



當視窗未設定為自動隱藏的狀態時，使用者可在專案管理區或訊息顯示區的頁面切換標籤上按住滑鼠左鍵，之後便可將該頁面拖曳至任何位置，甚至可拖曳至另一區的位置上與其合併；而要拖曳整個區域時，只要先按住該區上方的視窗標題之後再進行拖曳即可。

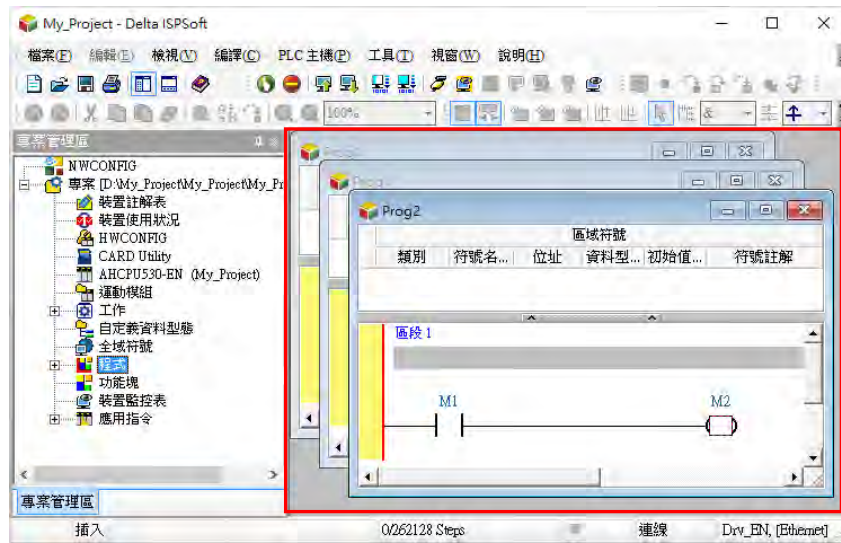
2



2.1.6 編輯工作區

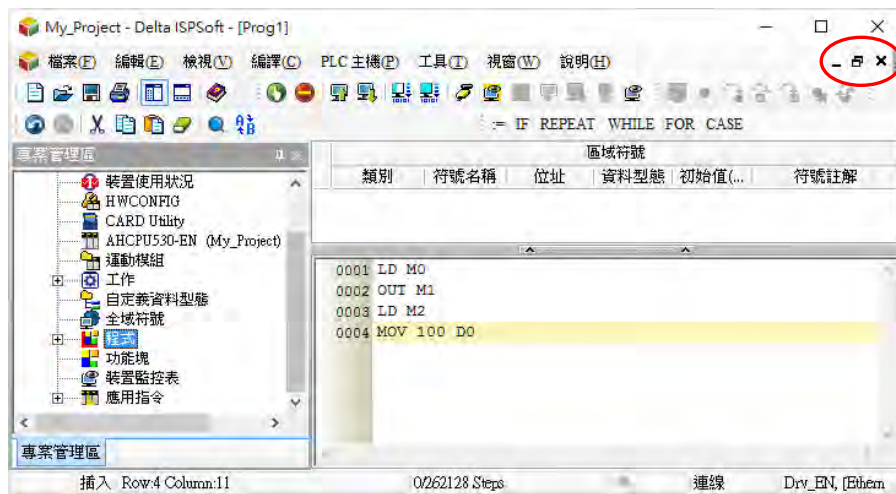
此區為各個工作視窗的所在位置，當將工作視窗設為最大化或最小化時，該視窗也會在此區域中進行縮放；而當工作視窗最大化時，該視窗的狀態操作按鈕則會顯示在功能工具列的右側。

● 工作視窗陳列

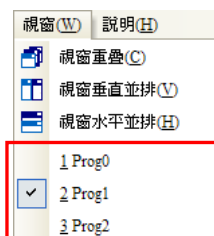


2

● 工作視窗最大化



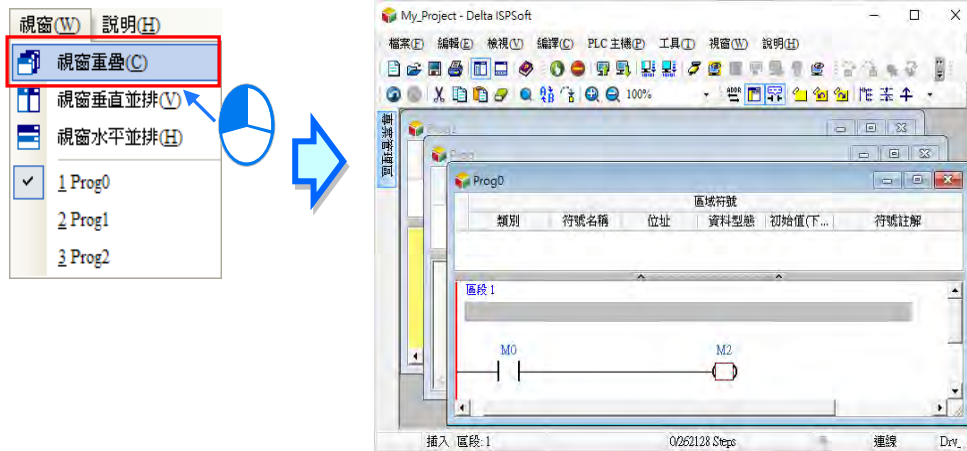
透過功能工具列中的「視窗 (W)」功能項即可對這些工作視窗進行管理與切換。



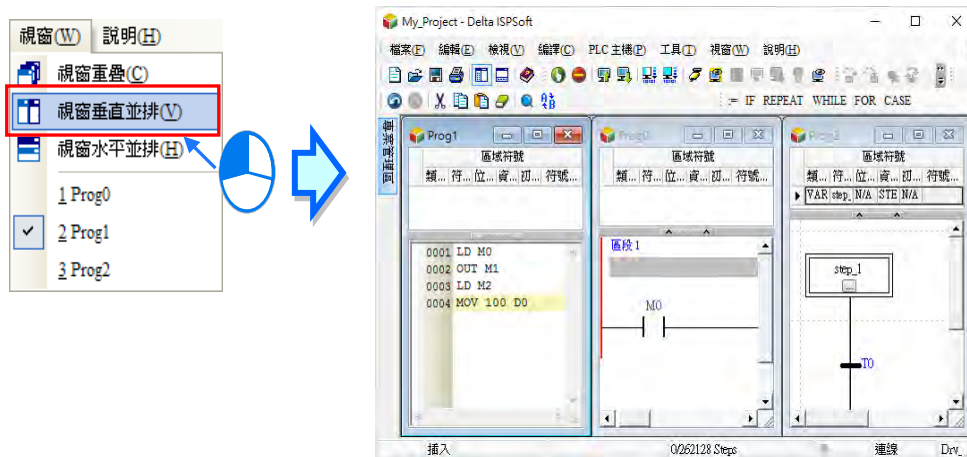
← 直接勾選以切換目前的工作視窗

- 視窗重疊：點選此項後，所有的視窗會以重疊的方式排列，其中目前的工作視窗會被排在最前方。

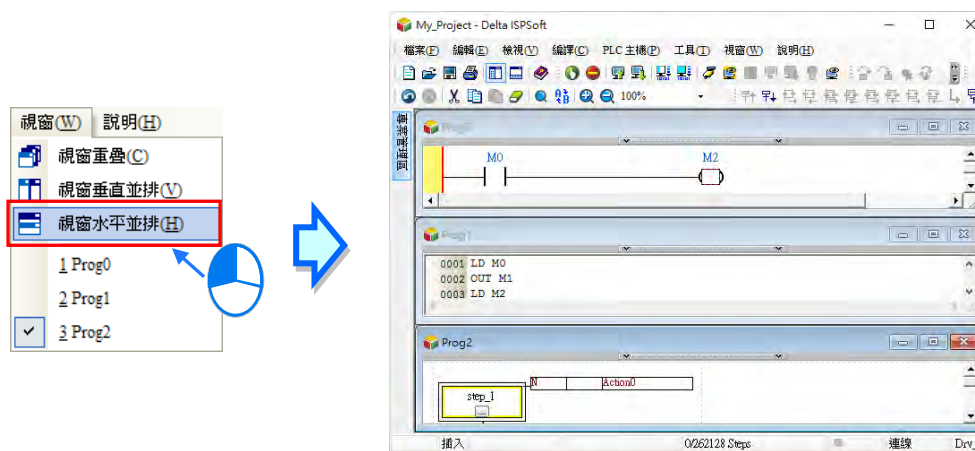
2



- 視窗垂直並排：所有的視窗會以垂直對齊的方式並行排列，其中目前的工作視窗會被排在最左側。



- 視窗水平並排：所有的視窗會以水平對齊的方式並行排列，其中目前的工作視窗會被排在最上方。



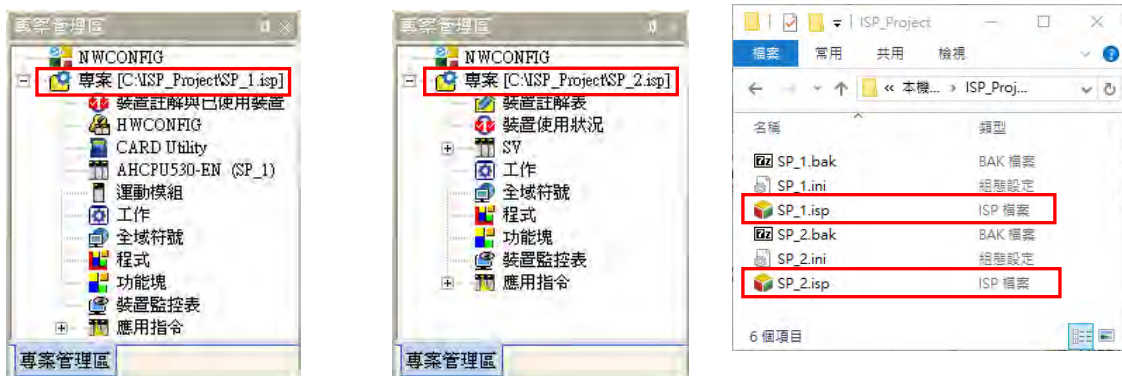
2.2 ISPSOft 的專案架構

2.2.1 單機專案與群組專案

ISPSOft 的專案架構共可分為單機專案與群組專案兩種，說明如下。

● 單機專案

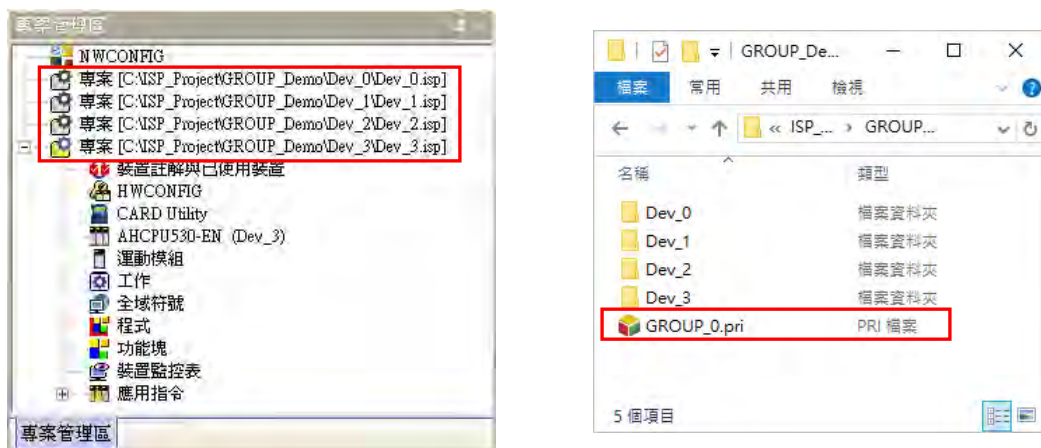
即最基本的專案類型，在每個專案中只可進行一個機種的開發，主要用於單純的單機應用，而專案檔的檔案類型則為 *.isp 檔。



● 群組專案

用於聯網應用的多主機系統，在一個專案群組中可建立多個單機專案，且不限機種系列；一個群組中的專案數量並沒有上限，使用者可依照實際應用的狀況加以規劃，而透過 ISPSOft 中的網路規劃工具 - NWCONFIG，即可簡單且快速的對同一群組中的各個裝置進行網路架構的規劃。

群組專案檔的檔案類型為 *.pri 檔，而群組下的每個單機專案的檔案類型則同樣為 *.isp 檔，且會置於與專案名稱相同的資料夾中進行分類管理；對於先前已建立好的單機專案，後續亦可透過匯入的方式來加入專案群組當中。

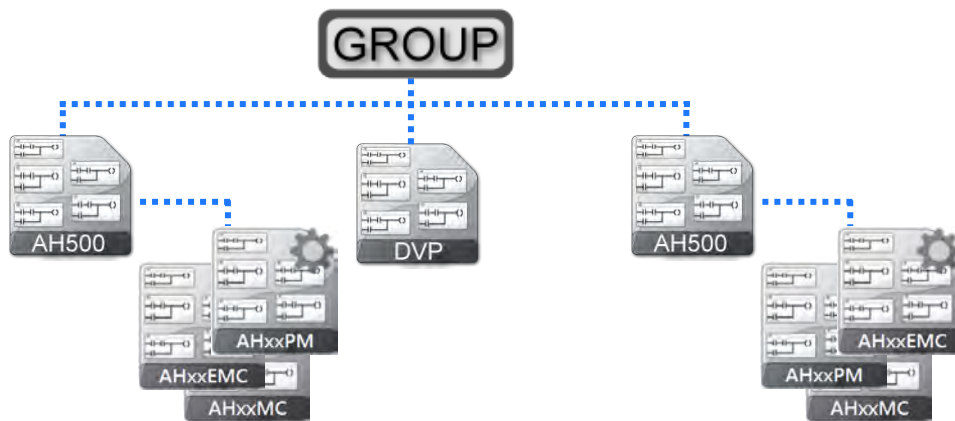


*.關於網路規劃工具 - NWCONFIG 的相關說明將會於第 20 章的內容當中有詳盡的介紹。

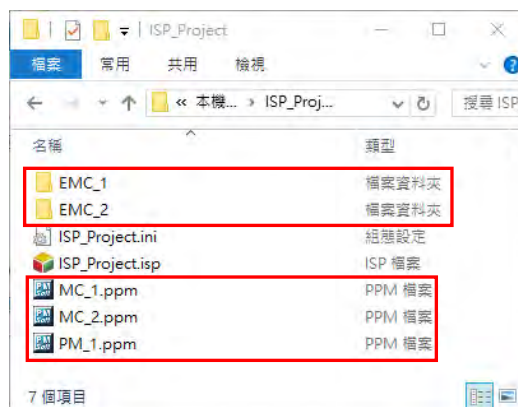
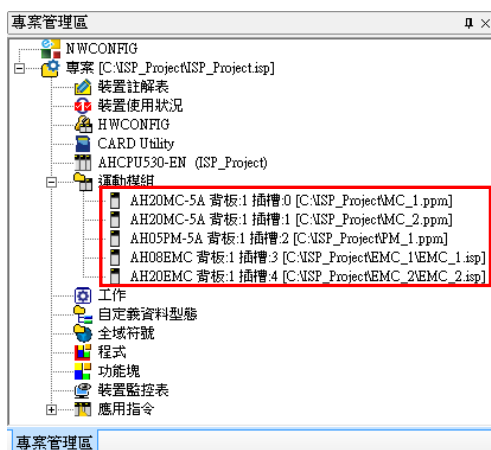
2.2.2 整合運動模組專案

AH 系列的系統提供了 AH05PM-5A、AH10PM-5A、AH15PM-5A 與 AH20MC-5A 及 AH08EMC、AH10EMC、AH20EMC 等運動控制模組，且使用者可於該模組的內部佈署獨立的運動控制程式，而 AHxxPM 與 AHxxMC 其開發的軟體為台達電子所推出的 **PMSoft**，相關細節與使用方式請參考各機種與 PMSoft 的使用手冊。AHxxEMC 則是使用 ISPSoft 開發，可以當主機也可以當運動模組使用。

使用者可直接在 ISPSoft 中建立運動模組專案，且可透過**硬體規劃工具 - HWCONFIG** 來進行相關的硬體配置與參數設定；而不分**單機專案**或是**群組專案**的架構，在每個 AH 系列機種的專案中皆可建立一個以上的**運動模組專案**，還需要在 **HWCONFIG** 中的硬體配置增加對應的運動模組，相關細節將於第 2.2.5 節當中另行介紹。




AHxxPM 與 AHxxMC 為 **PMSoft** 專案，檔案類型為 **.ppm** 檔，而 AHxxEMC 則是 **ISPSoft** 專案，會多一個專案資料夾，檔案類型為 **.isp** 檔，且會與 AH 機種的**單機專案**置於同一路徑下，而對於先前已建立好的 **PMSoft** 專案或 **ISPSoft** 專案，則可透過匯入的方式來加入 ISPSoft 的專案中。

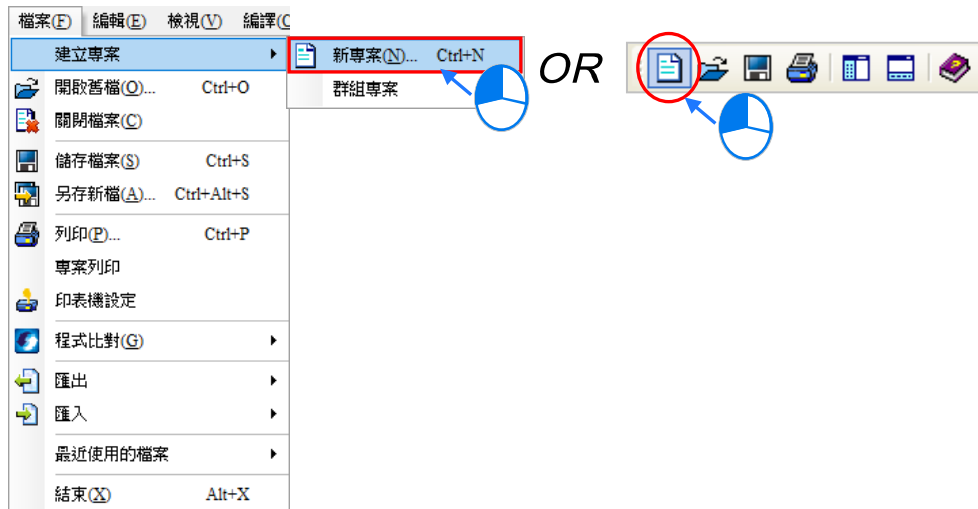


*1. 匯入的 PM 專案必須為 **PMSoft V2.05** 版以上的檔案格式，且不支援 **DVP-PM** 機種的專案。

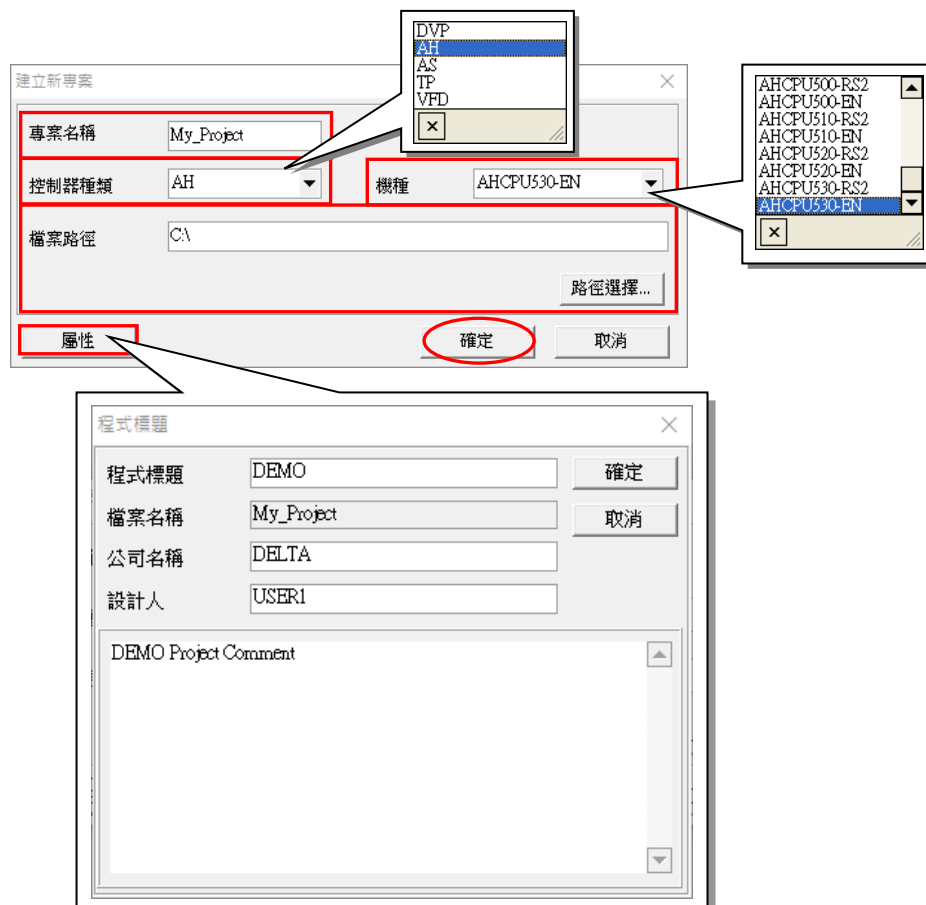
2.2.3 ISPSoft 的單機專案管理

● 建立新單機專案

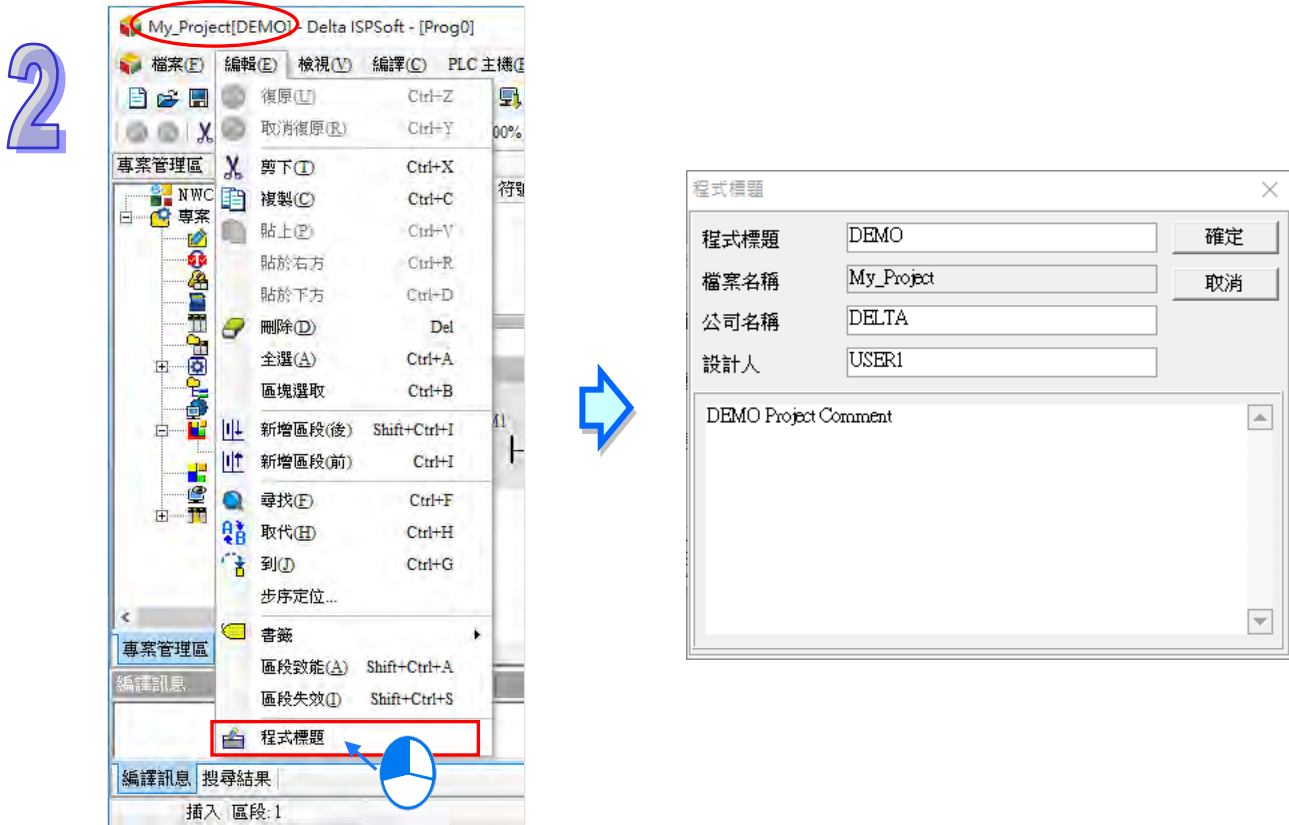
首先於功能工具列中點選 **檔案 (F) > 建立專案 > 新專案 (N)** 或點擊圖示工具列的  圖示。



接著於設定視窗中依序輸入「專案名稱」及「檔案路徑」，之後於「控制器種類」及「機種」的下拉選單中選擇機種型式；而點擊「屬性」則可為此專案加入相關的資訊及描述。完成後按下「確定」鍵即可。



完成新專案的建立後，於 ISPSOft 的視窗標題中便會顯示專案的名稱，格式為 **專案名稱【程式標題】**，而當沒有設定「程式標題」時，視窗標題則只會顯示「專案名稱」；當要檢視或修改「程式標題」等資訊時，只要在功能工具列上依序點選 **編輯 (E) > 程式標題** 即可。




在 **專案管理區** 中可見到專案檔的路徑與所選的專案機種。而當所建立的為 AH 及 AS 系列的機種專案時，於機種類型的後方還會以括號顯示該 PLC 的標籤名稱，且會以專案名稱做為其預設值，但之後仍可在 **硬體規劃工具 - HWCONFIG** 中重新設定，詳細說明將於 **第 3.3.2.1 節** 的內容當中另行介紹。

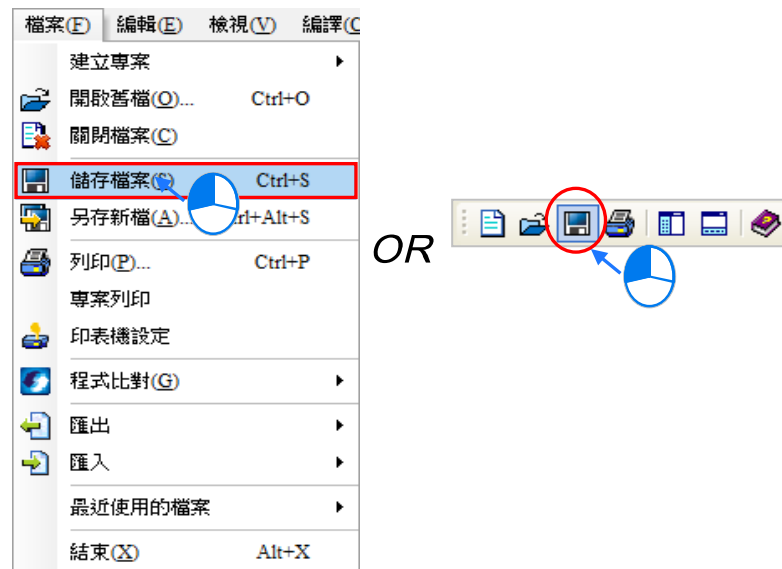


*. PLC 的標籤名稱僅有 AH/AS 系列、DVPxxMC/AS5xx 系列的機種才有，DVP 系列的機種則不會有此資訊，而其主要的功能是用來做為裝置識別，尤其是在聯網系統的應用或是在進行連線操作時，此資訊可用來確認實際的操作對象是否與我們所預期的相同。

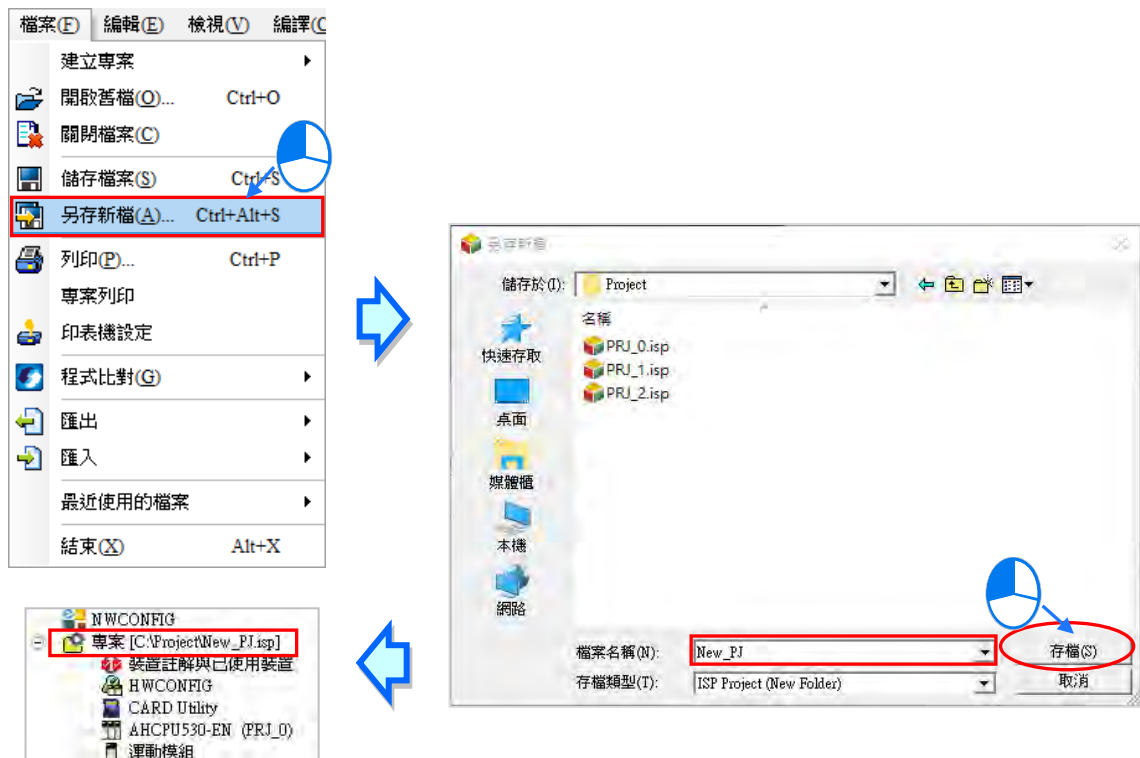
● 儲存專案

剛建立專案時，於指定的路徑中並不會馬上產生實際的專案檔，而必須在執行儲存後才會真正的產生指定的檔案。儲存專案的操作方式如下。

於功能工具列中點選 **檔案 (F) > 儲存檔案 (S)** 或直接點擊圖示工具列中的  圖示即可。




若要以其他的名稱或路徑另外儲存時，請確認 HWCONFIG、EthernetIP、EtherCAT 等相關編輯軟體已經存檔並關閉，接著於功能工具列中點選 **檔案 (F) > 另存新檔 (A)**，接著於檔案視窗中設定新的路徑及檔名，完成後按下「存檔」按鈕即可。



*.另存新檔的操作並不會變更 PLC 的標籤名稱，因標籤名稱僅會在一開始建立新專案時以專案名稱做為其預設值，但之後仍可在硬體規劃工具 - HWCONFIG 中重新設定。

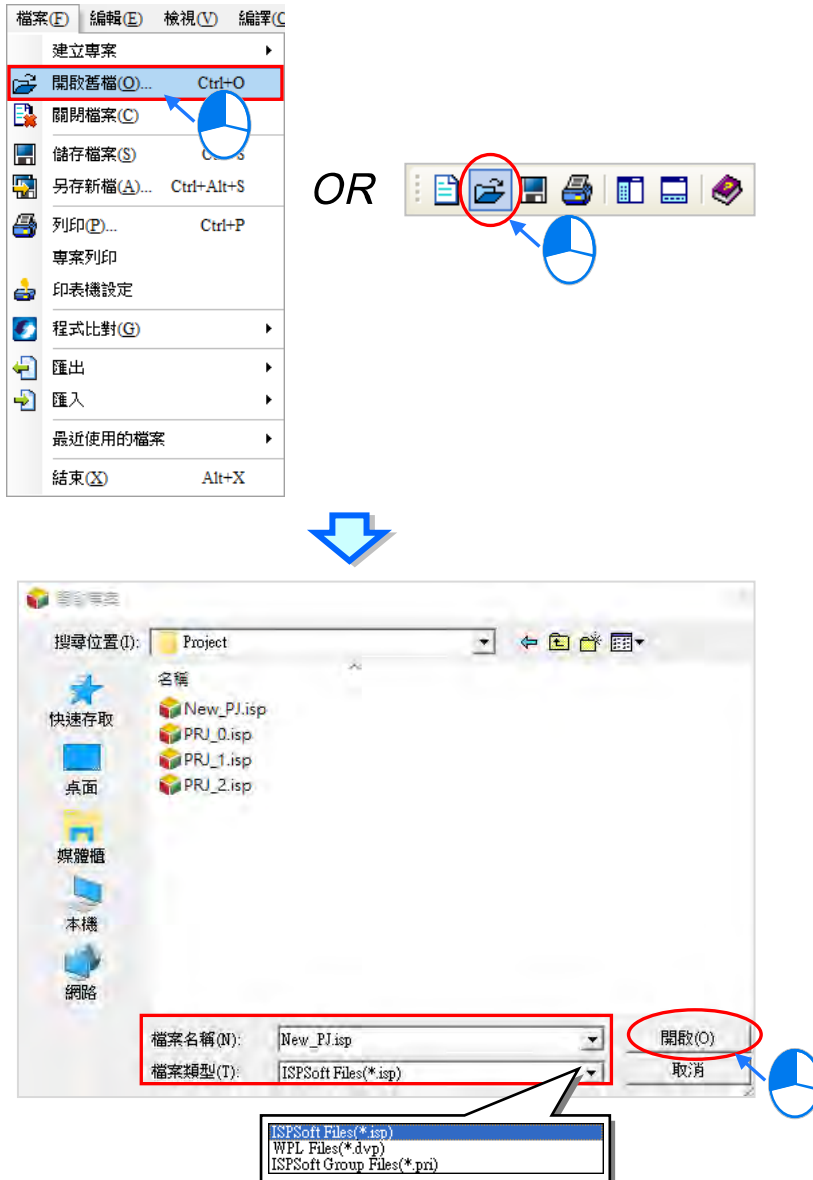
● 開啟舊專案

若要在 ISPSOft 當中開啟先前已建立的舊專案時，請於功能工具列中點選 **檔案(F) > 開啟舊檔(O)** 或直接點擊圖示工具列中的  圖示，接著於檔案視窗的「**檔案類型**」中下拉選擇欲開啟的檔案類型，之後選取欲開啟的檔案後按下「**開啟**」按鈕即可。下表則為可開啟的檔案類型。

2

檔案類型	說明
ISPSOft Files (*.isp)	此為 ISPSOft 的 單機專案檔 ，而若開啟了舊格式的檔案，則於儲存專案時，ISPSOft 將會自動將其轉換為新格式的專案檔。
WPL Files (*.dvp)	此為 WPLSoft 所建立的專案檔，而開啟此類型的檔案後，於儲存專案時，ISPSOft 會自動將其轉換為 .isp 的 ISPSOft 專案檔。
ISPSOft Group Files (*.pri)	此為 ISPSOft 的 群組專案檔 。

*.WPLSoft 亦為台達電子所推出的 PLC 編程軟體，相關說明請參考 **WPLSoft 使用手冊**。

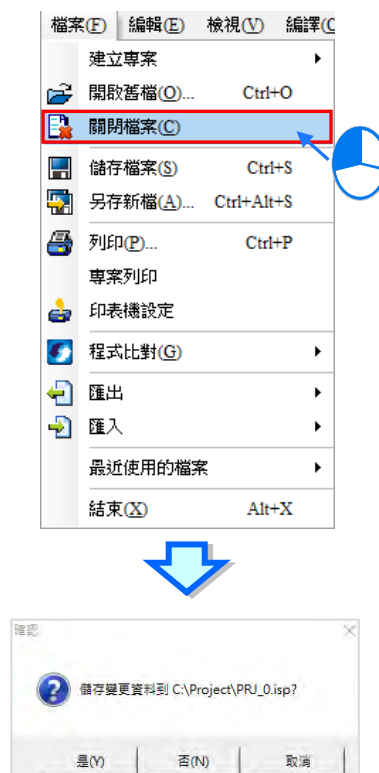


也可以在功能工具列中直接點擊「檔案 (F)」項目下的專案清單中的「最近使用的檔案」來開啟最近編輯過的專案。



● 關閉專案

若已完成專案的編輯時，請於功能工具列中點選 **檔案 (F) > 關閉檔案 (C)** 即可將專案關閉，而若尚未完成存檔時，系統便會出現提示視窗詢問是否要進行存檔。



補充說明

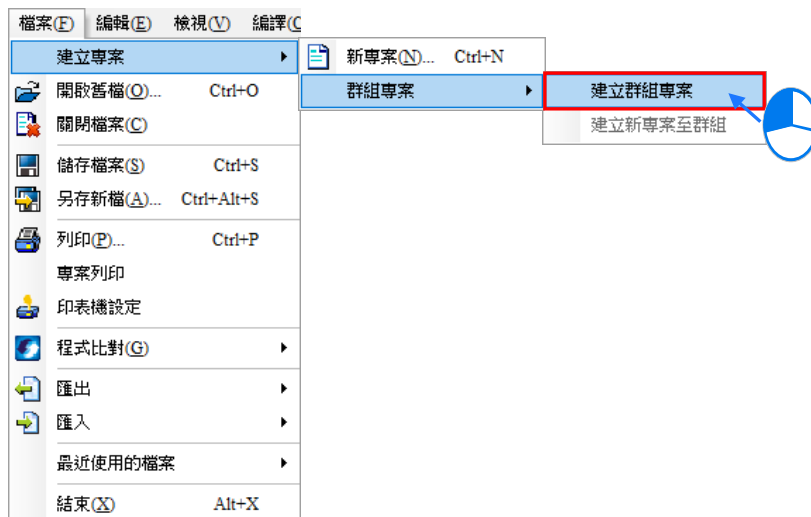
若離開 ISPSOFT 之前未先將專案關閉，則下次再啟動 ISPSOFT 時，系統預設將自動開啟上次所編輯的專案。(自動開啟前次專案的功能可由使用者自行設定啟動與否，相關細節請參考第 2.3.1 節。)

2.2.4 ISPSOft 的群組專案管理

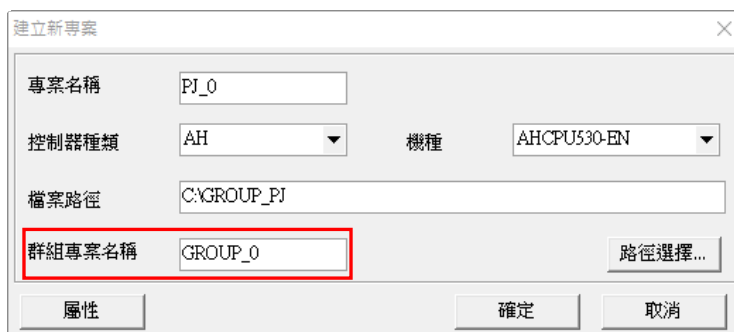
● 建立新群組專案

首先於功能工具列中點選 **檔案 (F) > 建立專案 > 群組專案 > 建立群組專案**。

2



而建立新群組時必須連帶建立該群組下的第一個單機專案，其設定方式則與建立**單機專案**時相同，分別於設定視窗中依序輸入「專案名稱」及「檔案路徑」，於「控制器種類」、「機種」的下拉選單中選擇機種型式，而點擊「屬性」則可為此專案加入相關的資訊及描述；除此之外，還必須於「**群組專案名稱**」的欄位中輸入群組名稱，完成後按下「確定」即可。



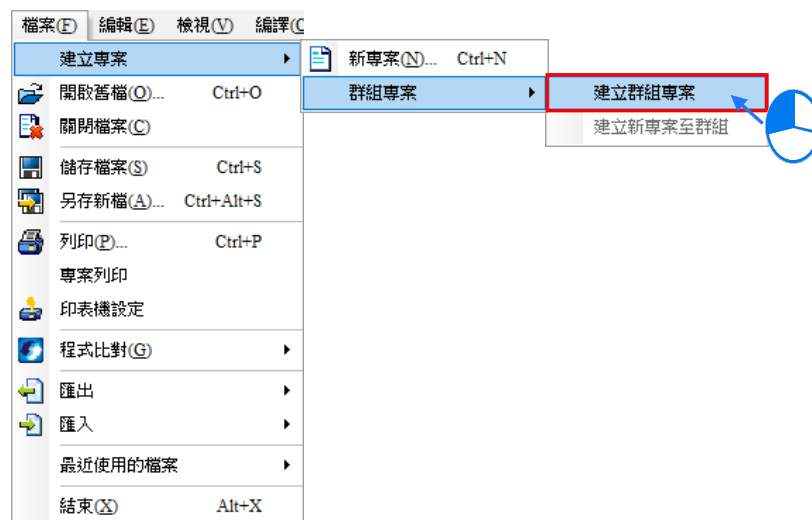
完成群組專案的建立後，於專案管理區中便會顯示連帶建立的單機專案資訊，且因一開始建立時，群組中僅有一個單機專案，所以其顯示與單獨建立單機專案時相同，而所有相關的檔案操作也都與在單機專案架構下的方式無異，唯一不同的是，於群組專案的架構下無法執行另存新檔的功能。



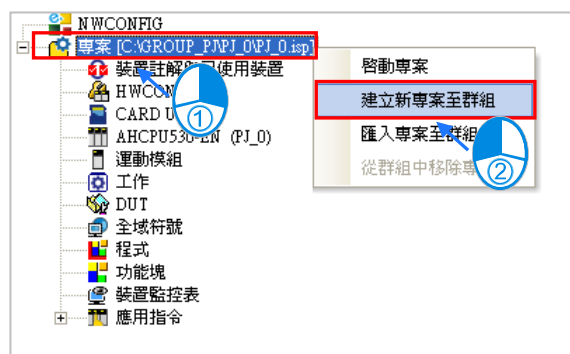
2

● 新增單機專案至群組

欲新增其他單機專案至群組中時，請於功能工具列中依序點選 **檔案 (F) > 建立專案 > 群組專案 > 建立新專案至群組**，亦或是於專案管理區中點選任一專案後按下滑鼠右鍵，之後於快捷選單中點選 **建立新專案至群組**。



OR



因在群組專案的架構下，同時間只能有一個單機專案處於編輯狀態，因此當要新增一個專案至群組中時，原來正在編輯的專案便會被要求關閉，而此時若該專案尚未存檔時，系統便會出現確認訊息。

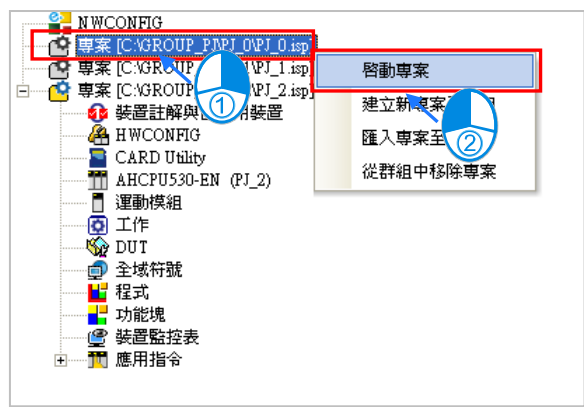
新專案的設定視窗與一般建立新單機專案時相同，唯一不同的是無法設定檔案路徑。

2



● 啟動專案

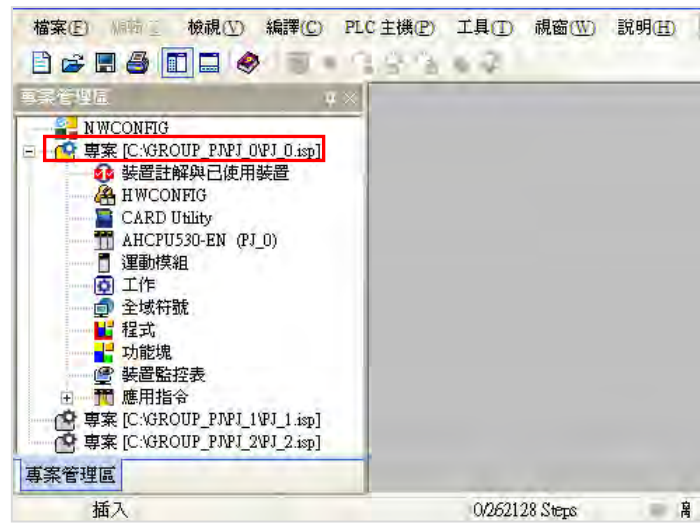
在群組專案的架構下，同時間只能有一個單機專案處於編輯狀態，而要切換編輯的專案時，只要在專案物件上點擊滑鼠右鍵，之後於快捷選單中點選 **啟動專案**，或是直接在專案物件上雙擊滑鼠左鍵即可。



OR



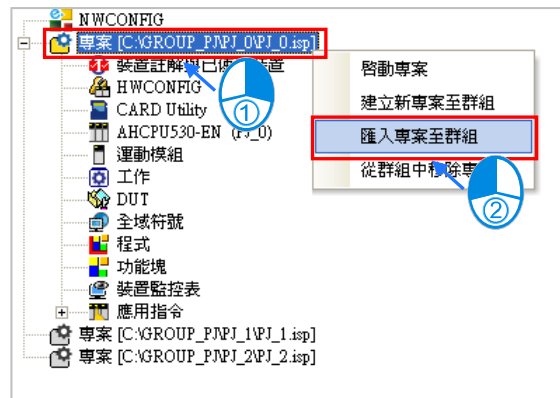
切換專案後，專案管理區中可編輯的專案會變成剛剛選取的專案，其餘處於關閉狀態的專案則會以灰階的圖示顯示。



2

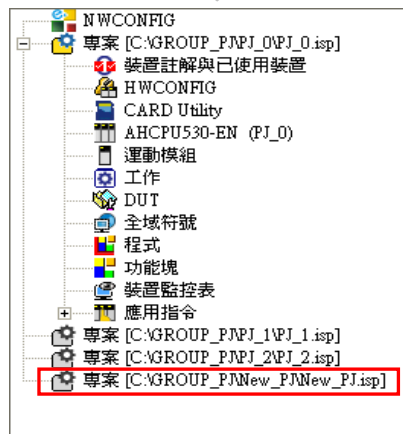
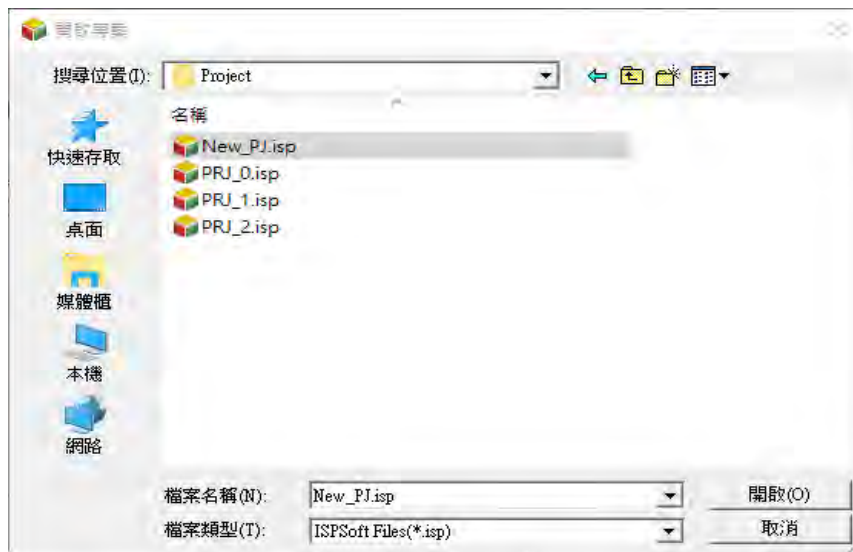
● 匯入單機專案檔至群組

若有某個單機專案檔 (*.isp) 是在非群組專案的架構下所建立的，透過匯入的方式亦可將其加入專案群組當中；欲進行匯入時，請於專案管理區中點選任一專案後按下滑鼠右鍵，之後於快捷選單中點選匯入專案至群組，接著便可於檔案視窗中選取欲匯入的專案檔。



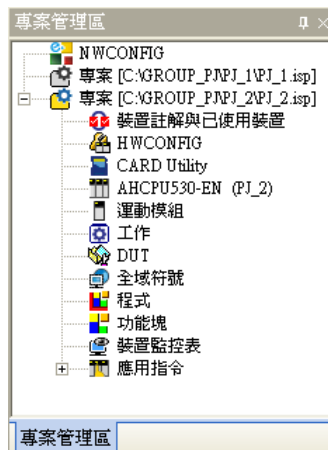
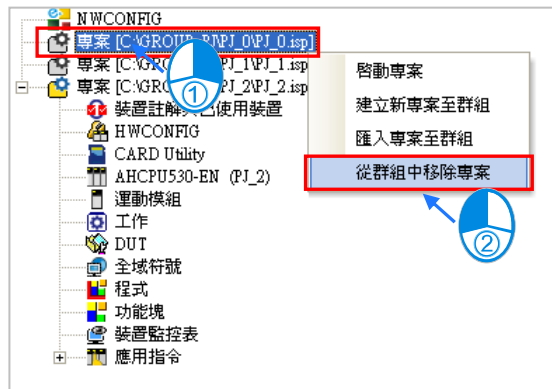
允許匯入的檔案類型只可為 *.isp 的單機專案檔，而匯入專案檔時，系統僅會將該單機專案檔複製至專案群組的目錄下，而原單機專案檔則仍會被保留在原来的路徑中。

2

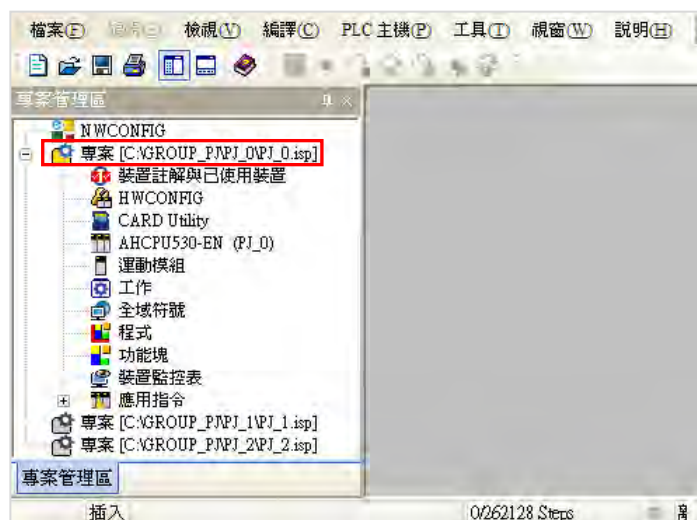


● 移除群組中的單機專案


若要將某個單機專案從群組中移除時，請於**專案管理區**中點選欲移除的專案後按下滑鼠右鍵，之後於快捷選單中點選**從群組中移除專案**；不過若該群組中僅剩一個專案時便無法移除。



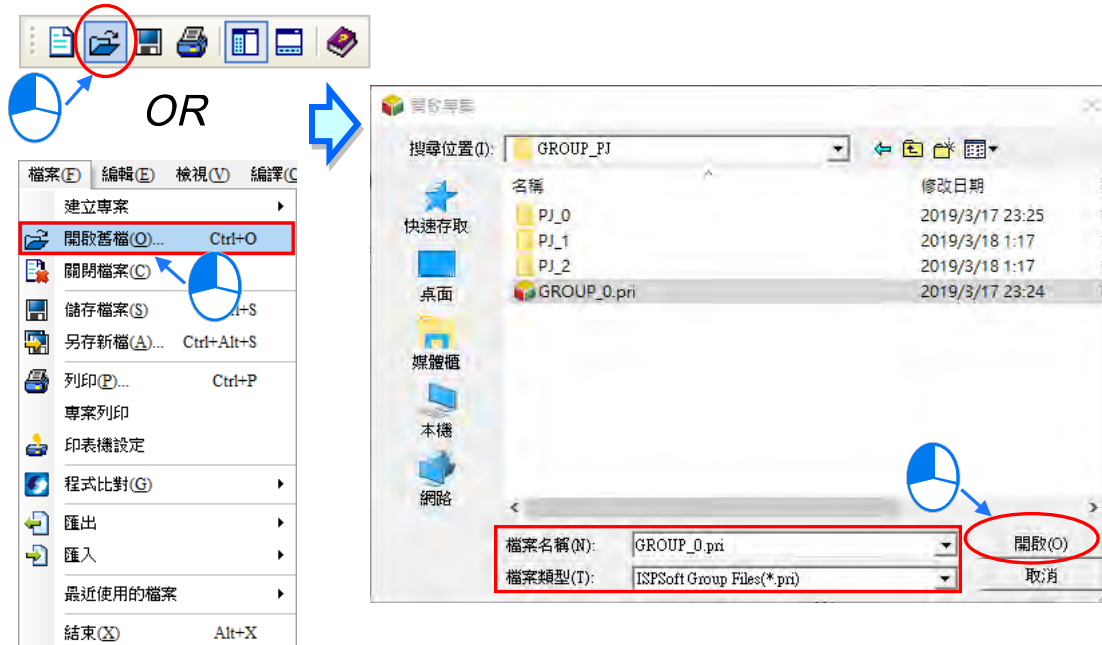
此操作僅會將該項目於群組專案的管理列表中移除而已，實際存放在原路徑的**單機專案檔**並不會被刪除，而之後若要重新將該專案加入群組時，只要透過匯入的方式重新將其加入即可。



● 開啟群組專案

若要開啟之前建立的群組專案時，請於功能工具列中點選 **檔案 (F) > 開啟舊檔 (O)** 或直接點擊圖示工具列中的  圖示，接著再於檔案視窗的「檔案類型」下拉選擇 **ISPSOft Group Files (*.pri)**，之後選取欲開啟的檔案後按下「**開啟**」按鈕即可。

2



而若開啟的群組專案中有出現紅色叉號的圖示時，代表位於此群組路徑下的該單機專案檔出現異常，有可能已被刪除、變更路徑或是變更名稱。



- 關閉群組專案

於功能工具列中點選 **檔案 (F) > 關閉檔案 (C)** 可將目前正在編輯的單機專案關閉，但此時仍可再啟動其他處於關閉狀態的單機專案，而若要關閉整個群組專案時，請點選 **關閉全部專案**。



2

2.2.5 ISPSOft 的運動模組專案管理

在 ISPSOft 的專案管理區中，針對 AH 系列機種的專案會多出一個「運動模組」的項目，若使用者要增加 AHxxPM 與 AHxxMC 運動模組專案，則必須要安裝 PMSOft V2.05 版以上的軟體。

- 建立新 AHxxPM 與 AHxxMC 運動模組專案

要在 ISPSOft 中建立 PMSOft 專案時，請於專案管理區中點選「運動模組」項目後按下滑鼠右鍵，之後於快捷選單中點選 **建立運動模組**。



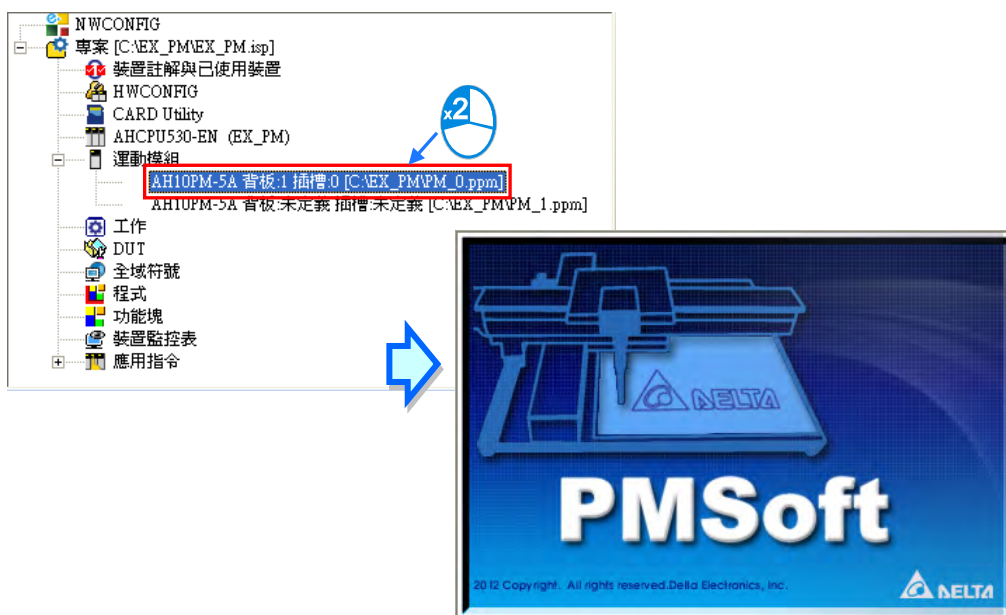
接著請於設定視窗的「檔案名稱」中輸入運動模組專案的名稱，於「型號」欄位下拉選擇模組型號，之後依據 HWCONFIG 中的配置，於「背板編號」及「插槽編號」欄位中下拉選擇該模組安裝的位置，且每個位置僅可對應一個運動模組專案，而若尚未確定安裝位置亦或是之後要重新調整插槽時，可先於背板編號下拉選單中選擇「未定義」。完成設定之後按下「確定」即可。

2



● 開啟 AHxxPM 與 AHxxMC 運動模組專案

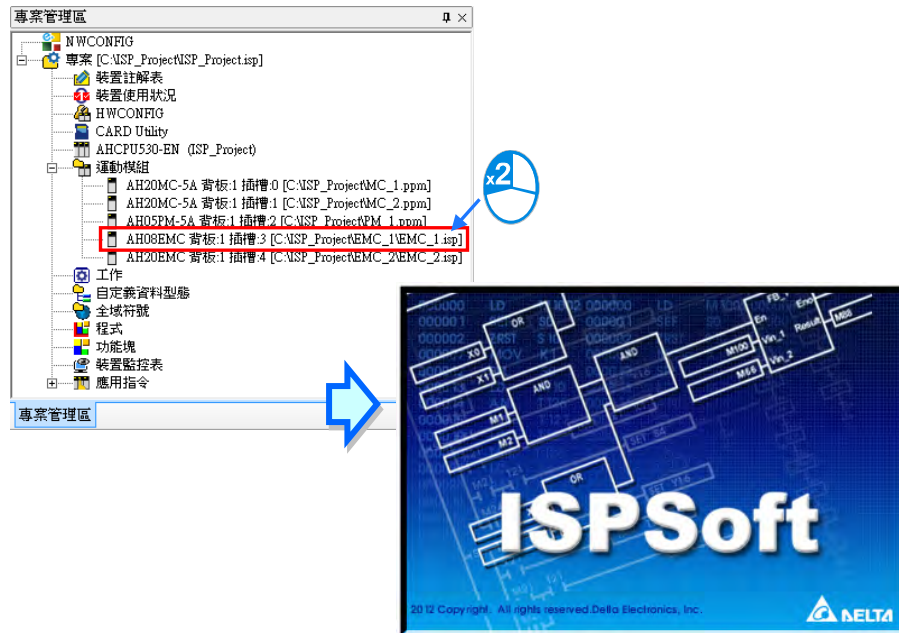
當要對已建立好的 AHxxPM 與 AHxxMC 運動模組專案進行編輯時，請直接於該專案物件上雙擊滑鼠左鍵，之後系統便會自動開啟 PMSOft，而使用者後續即可直接於 PMSOft 當中進行該運動模組的程式開發。



*.關於 PMSOft 的操作請參考 PMSOft 使用手冊，而透過 ISPSOft 啟動的 PMSOft，只可以針對該運動模組編輯，且編輯完成之後仍必須於 PMSOft 中執行「儲存」功能才可完成運動模組編輯。

● 開啟 AHxxEMC 運動模組專案

當要對已建立好的 AHxxEMC 運動模組專案進行編輯時，請直接於該專案物件上雙擊滑鼠左鍵，之後系統便會自動開啟 ISPSoft，而使用者後續即可直接於 ISPSoft 當中進行該運動模組的程式開發。

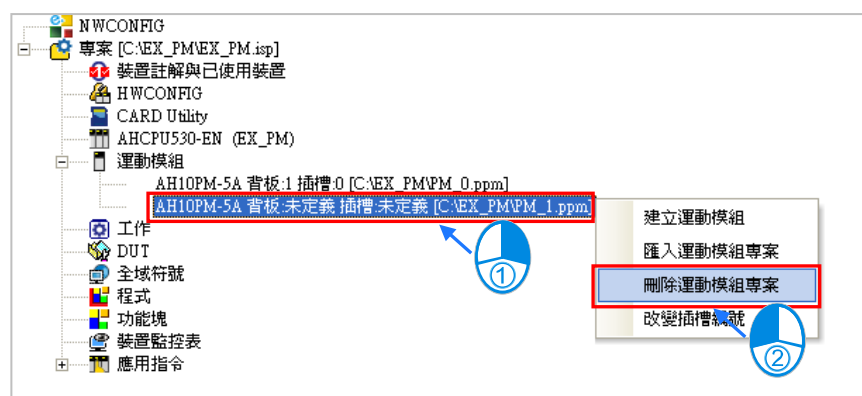


2

*透過 ISPSoft 啟動的 ISPSoft，只可以針對該運動模組編輯，且編輯完成之後仍必須於 ISPSoft 中執行「儲存」功能才可完成運動模組編輯。

● 刪除運動模組專案

欲刪除已建立好的運動模組專案時，請先點選欲刪除的專案後按下滑鼠右鍵，之後於快捷選單中點選刪除運動模組專案；而刪除運動模組專案時，系統僅會將該專案由專案管理清單中移除，實際存放的運動模組專案檔 (*.ppm) 或 (*.isp) 則仍會保留在原來的路徑中。



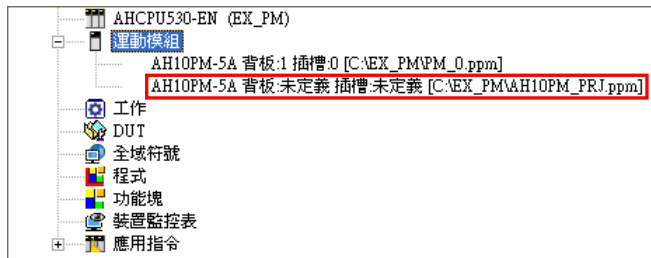
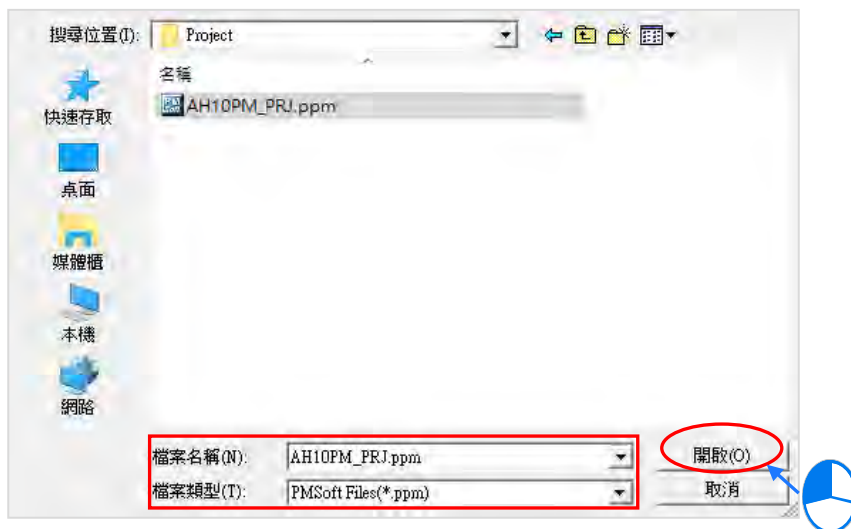
● 匯入運動模組專案

要匯入先前所建立的運動模組專案檔 (*.ppm) 或 (*.isp) 時，請先於專案管理區中點選「運動模組」項目後按下滑鼠右鍵，之後於快捷選單中點選 匯入運動模組專案。

2



接著請於檔案視窗中選取欲匯入的運動模組專案檔，完成後按下「開啟」即可。而匯入的運動模組專案，插槽編號將預設為未定義。



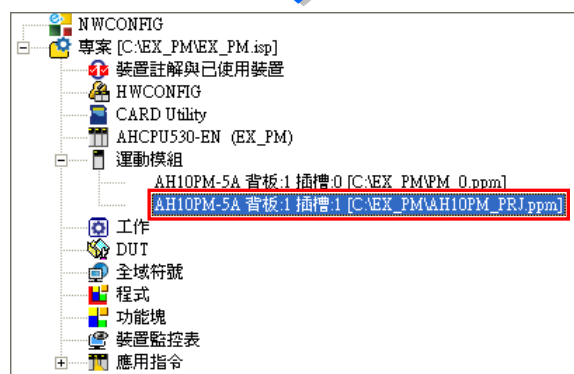
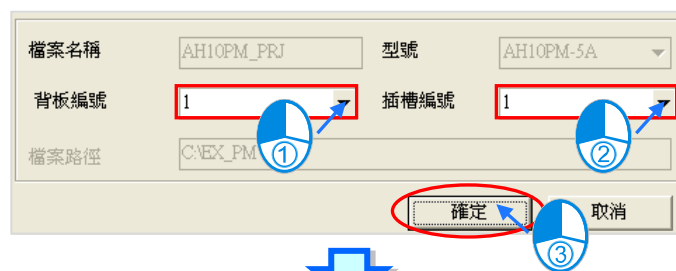
匯入 PM 專案檔時，系統僅會將該 PMSOft 專案檔複製至 ISPSOft 的專案路徑下，而原始的 PMSOft 專案檔則仍會被保留在原來的路徑中，且匯入的 PMSOft 專案檔必須為 PMSOft V2.05 版以上的檔案格式，且支援 DVP-PM 機種的專案；匯入 ISP 專案檔時，系統會將該 ISPSOft 專案檔複製至 ISPSOft 的專案路徑下並建立一個資料夾。

● 改變插槽編號

欲變更運動模組專案所對應的背板與插槽編號時，請先點選該模組的專案後按下滑鼠右鍵，之後於快捷選單中點選 **改變插槽編號**。



接著請於設定視窗的「背板編號」與「插槽編號」欄位下拉選擇欲變更的編號，而選擇時請注意不可選用已被佔用的插槽；完成後按下「確定」即可。



2.3 ISPSOft 中的基本設定

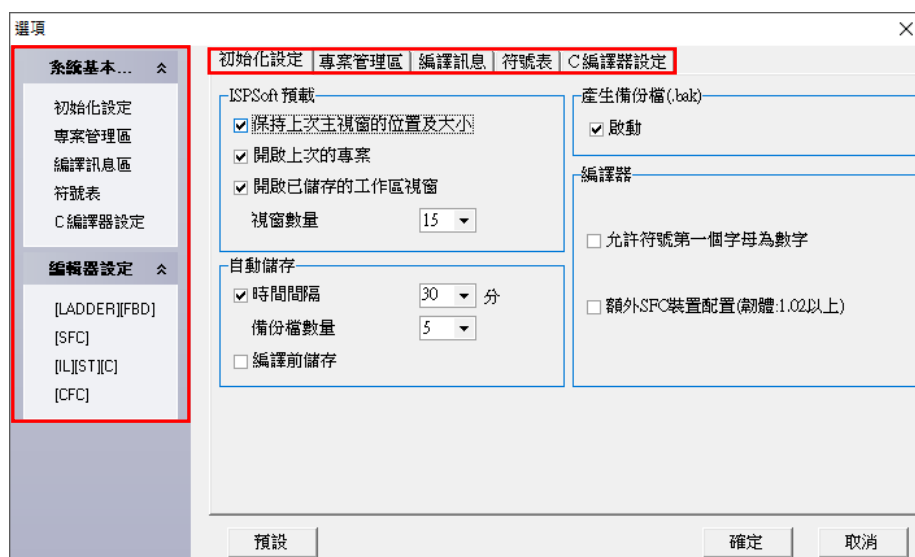
2.3.1 系統及環境設定

於功能工具列中點選 **工具 (I)** > **語系**，之後於下拉選單中便可選擇欲使用的操作語系。

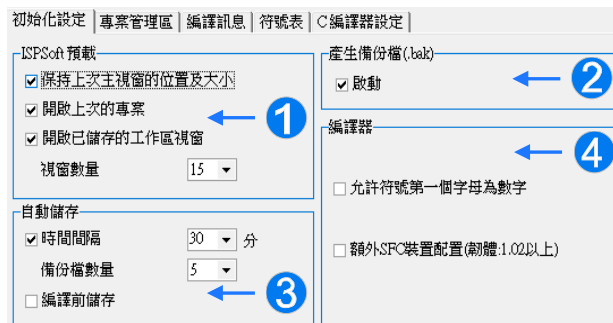
2



於功能工具列點選 **工具 (I)** > **選項 (O)** 之後便會出現操作環境的設定視窗，而透過視窗上方或左側的標籤便可切換各個設定頁。



● 系統基本設定 - 初始化設定



- ❶ 保持上次主視窗的位置及大小：**啟動 ISPSOFT 時自動開啟最後一次編輯的專案。

開啟上次的專案：自動記憶最後一次編輯的主視窗位置及大小，使用雙螢幕編輯的使用者可以取消此項目，以使主視窗每次都會出現在主螢幕，可以避免無第二個螢幕時，主視窗還停留在第二個螢幕的問題。

開啟已儲存的工作區視窗：啟動 ISPSOFT 時自動開啟最後一次存檔未關閉之工作區視窗，也可透過**視窗數量**選取要顯示的視窗數量，最多為 15 個。
- ❷ 產生備份檔啟動：**存檔時會自動產生一個備份檔案。
- ❸ 時間間隔：**依照右側的欄位所選取的時間，系統便會自動將目前所編輯的專案依照系統的時間存成專案檔，使用者可由「備份檔數量」設定最大的備份數量，若自動存檔超過最大數量，將依序覆蓋先最早的存檔；若 ISPSOFT 發生不正常關閉，則下次再開啟該專案檔時，系統便會詢問使用者是否要載入最後備份的編輯內容。

編譯前儲存：每次編譯前會自動儲存目前編輯的專案檔。
- ❹ 允許符號表中位址欄位裝置名稱重覆：**允許不同的變數符號指定到同一個裝置位址。

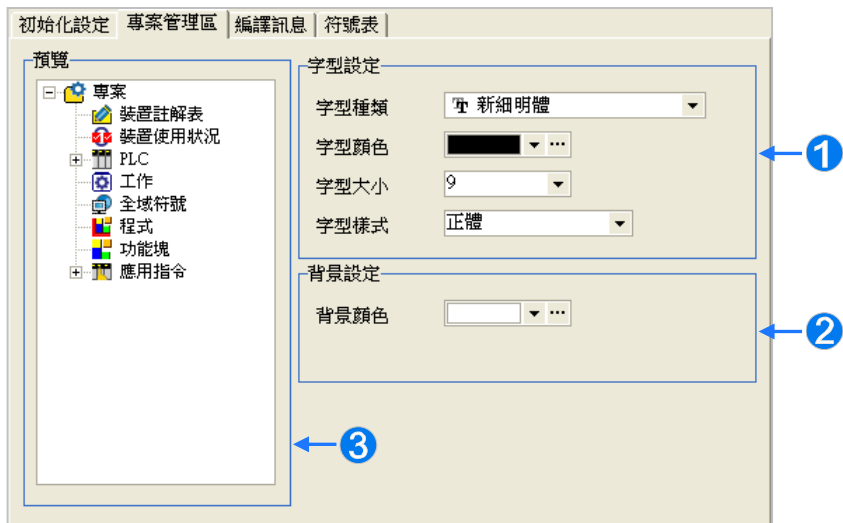
允許符號第一個字母為數字：允許變數符號名為數字開頭。

舊版變數陣列編譯：支援舊版變數陣列編譯。

*功能項目可能會隨著專案機種而有所不同。

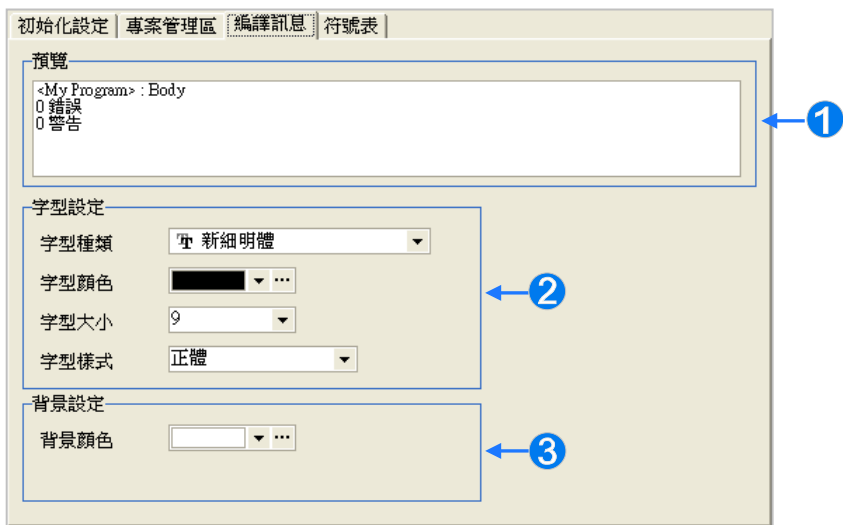
2

● 系統基本設定 - 專案管理區



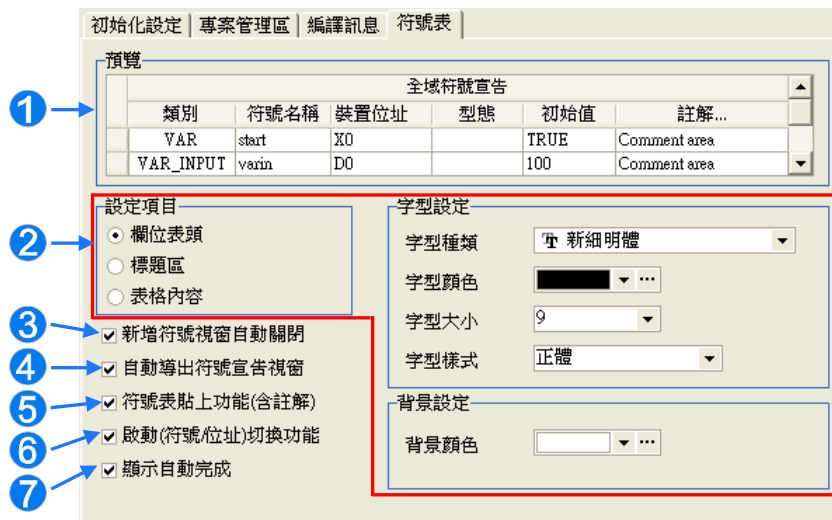
- ① 設定專案管理區中，各項目的顯示字型。
- ② 設定專案管理區的背景底色。
- ③ 此處顯示的為變更後的預覽畫面。


● 系統基本設定 - 編譯訊息

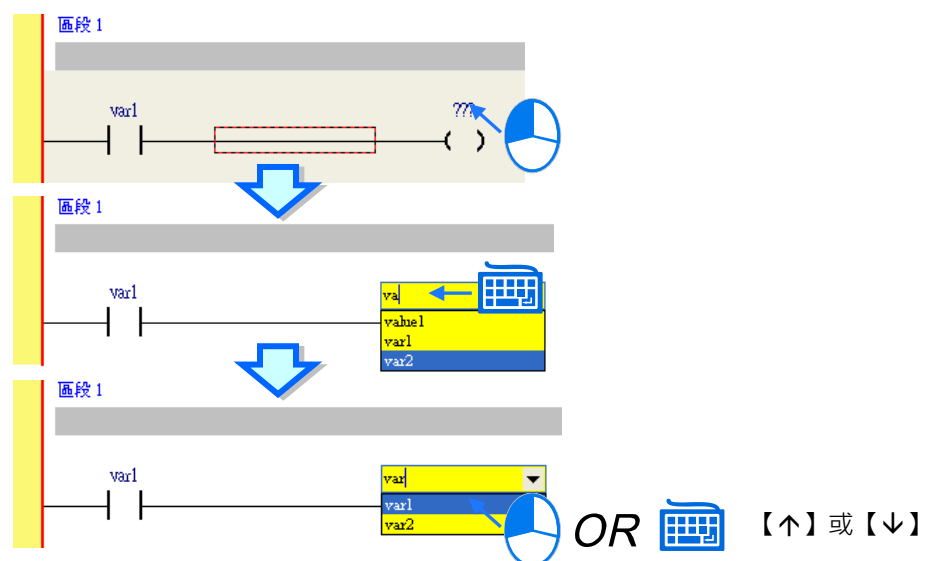


- ① 此處顯示的為變更後的預覽畫面。
- ② 設定訊息的顯示字型。
- ③ 設定訊息顯示區的背景底色。

● 系統基本設定 - 符號表

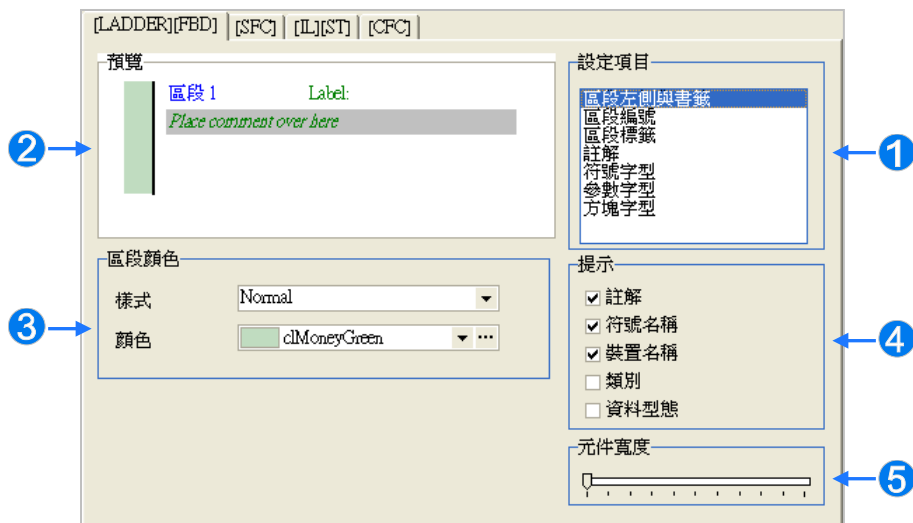


- ❶ 此處顯示的為變更後的預覽畫面。
- ❷ 先於左側選擇設定項目後，再於右側欄位中設定該項目的顯示樣式。
- ❸ 符號宣告視窗在完成宣告後的狀態會自動關閉。
- ❹ 於 LD、FBD 或 SFC 等環境中輸入未定義的變數符號，並在輸入完畢後直接按下【Enter】鍵，系統會自動導出符號宣告視窗。
- ❺ 符號表之間進行複製/貼上時，內容是否要包含變數符號的註解。
- ❻ 啟用圖示工具列的位址模式  按鈕，可以切換地址模式或位址模式。
- ❼ 在程式中輸入變數符號時，會自動列出所有符合的已建立變數符號項目，使用者可直接點選滑鼠左鍵或使用鍵盤上下鍵由清單中挑選，如下圖。



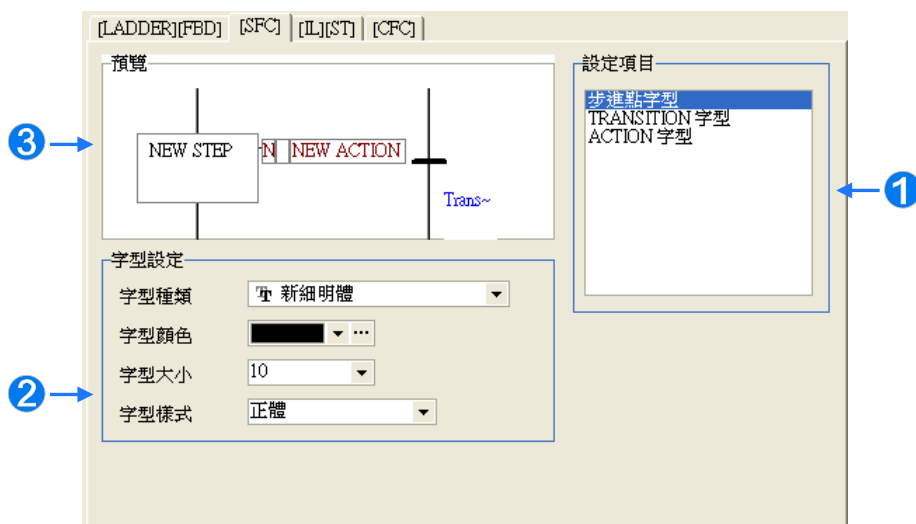
● 編輯器設定 - LADDER 及 FBD

2



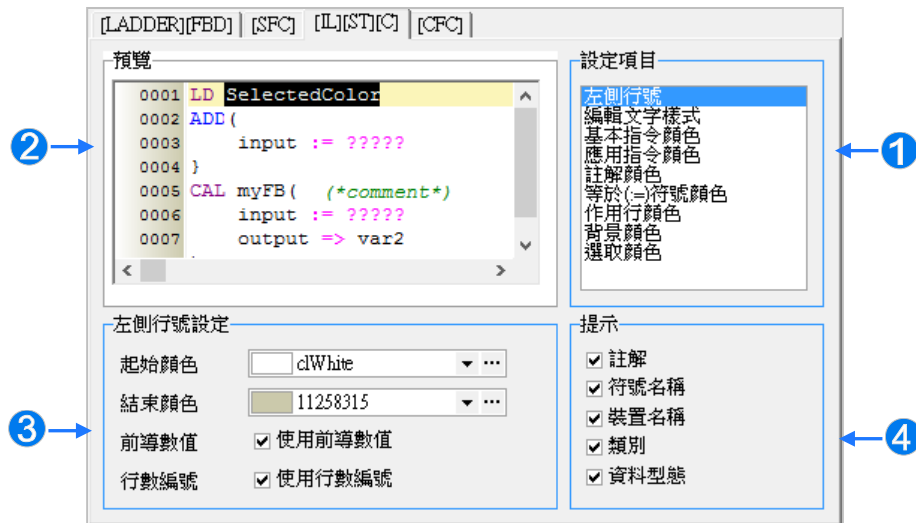
- ① 先於此處選擇設定項目。
- ② 此處為變更後的預覽畫面
- ③ 於此處設定所選項目的顯示樣式。
- ④ 在 LD 及 FBD 的程式編輯區中，當滑鼠停留在某個裝置或變數符號上時，畫面將會顯示該裝置或變數符號的提示，而於此便可勾選將會顯示的提示項目。
- ⑤ 於此處設定 LD 及 FBD 中的元件顯示寬度，而此設定將會影響編程環境中，功能塊圖示內部可顯示的字數長短。

● 編輯器設定 - SFC



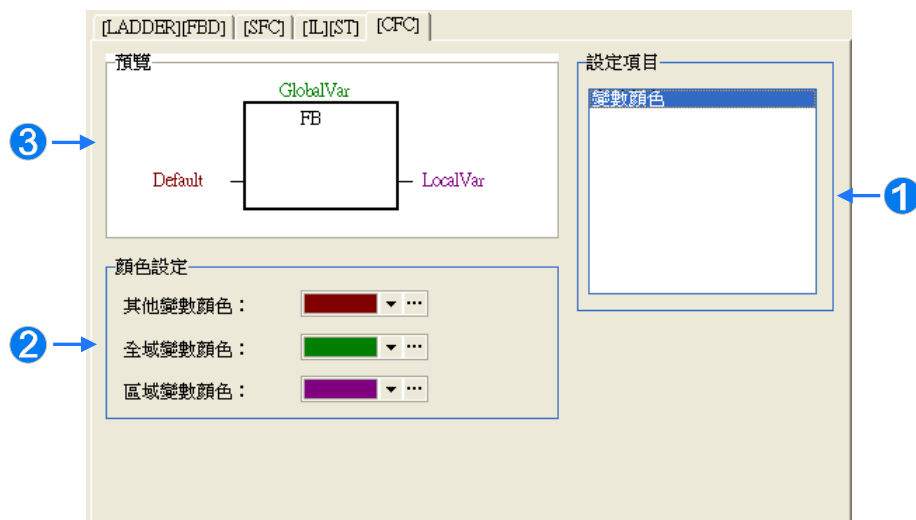
- ① 先於此處選擇設定項目。
- ② 於此處設定所選項目的顯示樣式。
- ③ 此處顯示的為變更後的預覽畫面。

● 編輯器設定 - IL 及 ST 和 C



- ❶ 先於此處選擇設定項目。
- ❷ 此處顯示的為變更後的預覽畫面。
- ❸ 於此處設定所選項目的顯示樣式。
- ❹ 在 IL 及 ST 的程式編輯區中，當滑鼠停留在某個裝置或變數符號上時，畫面將會顯示該裝置或變數符號的提示，而於此便可勾選將會顯示的提示項目。

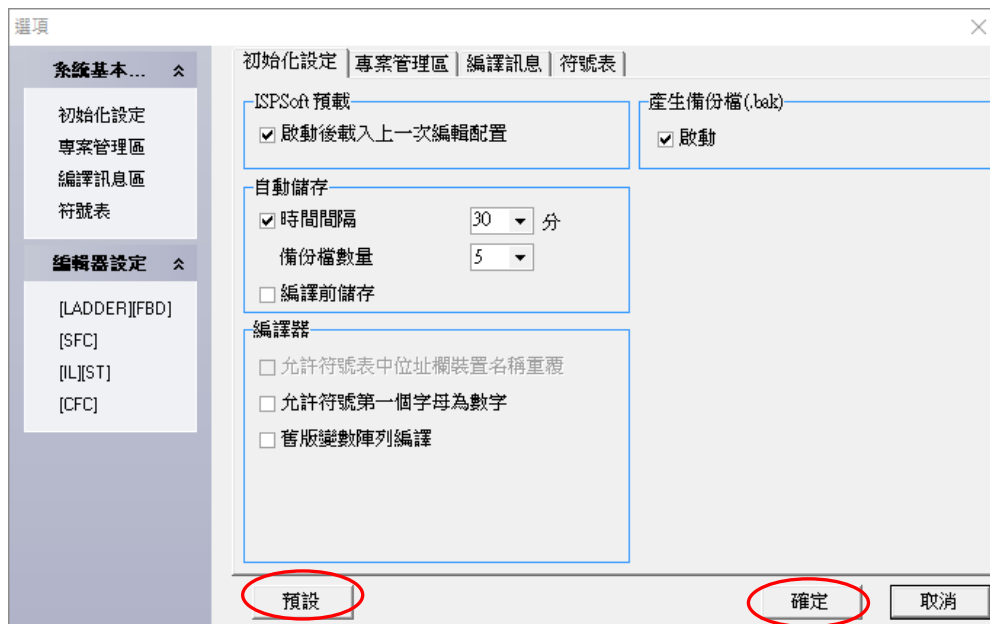
● 編輯器設定 - CFC



- ❶ 先於此處選擇設定項目。
- ❷ 於此處設定所選項目的顯示樣式。
- ❸ 此處顯示的為變更後的預覽畫面。

當按下「預設」按鈕時，所有的設定值便可恢復為系統最初的預設值，而完成所有的設定之後，按下「確定」按鈕即可套用所做的變更。

2



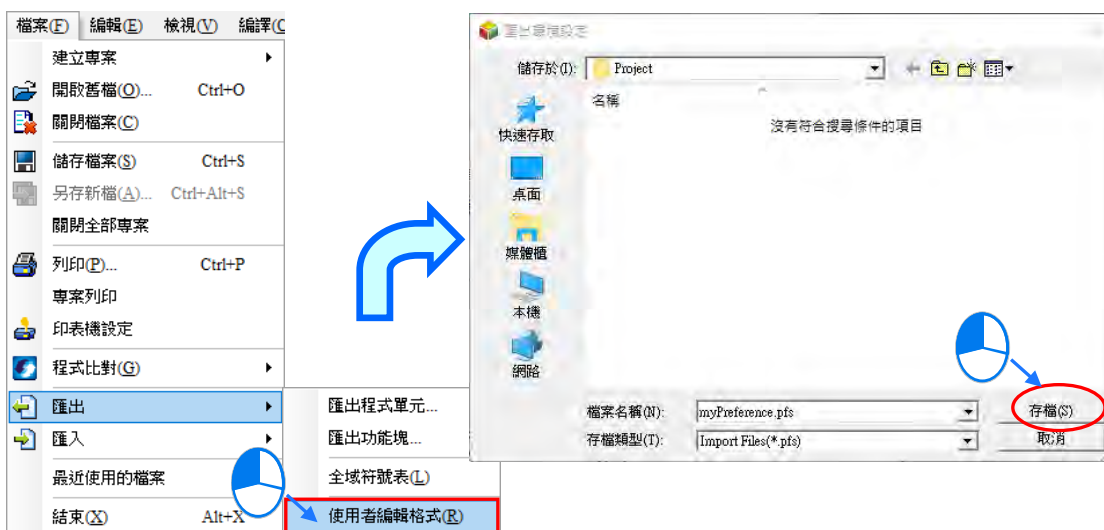
*若同時開啟多個 ISPSOft 視窗時，不同的 ISPSOft 視窗可做不同的設定，不過系統只會記憶最後一個關閉視窗的環境設定，因此當下次重新啟動 ISPSOft 時，系統便會自動套用最後一個關閉視窗中的設定值。

2.3.2 使用者編輯格式的匯入與匯出

透過 ISPSOft 所提供的功能，使用者可將自己設定好的環境設定進行匯出與匯入的動作。

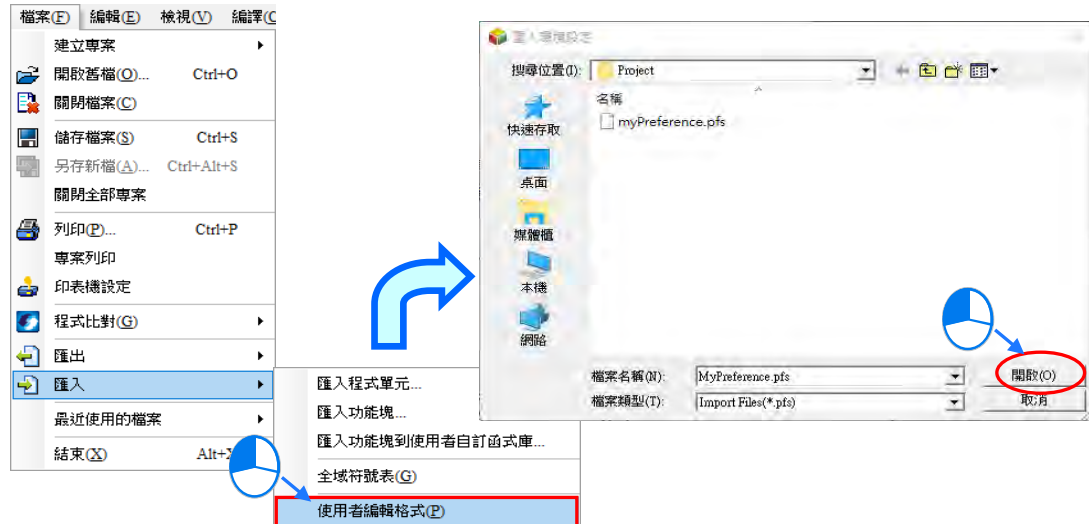
● 匯出使用者編輯格式

於功能工具列中點選 **檔案 (F) > 匯出 > 使用者編輯格式 (R)**，之後於檔案視窗中輸入欲匯出的檔名後按下「存檔」即可。



● 匯入使用者編輯格式

於功能工具列中點選 **檔案 (F)** > **匯入** > **使用者編輯格式 (P)**，之後於檔案視窗中選擇欲匯入的檔案後按下「開啟」即可。

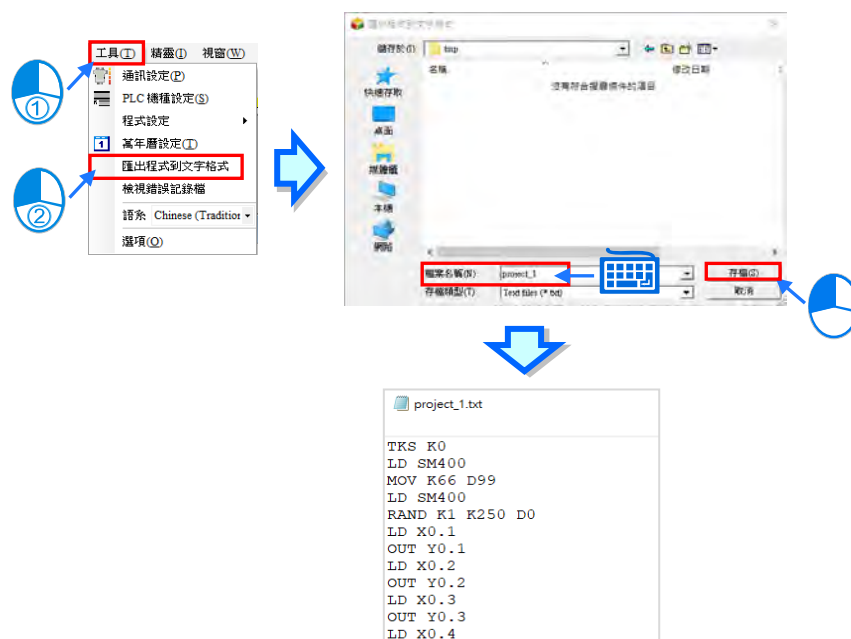


補充說明

不同版本的 ISPSOft，因其環境參數可能會有所不同，所以匯入時不見得可以完全套用，因此若不確定所匯入的**使用者編輯格式檔 (*.pfs)**是由哪個版本的 ISPSOft 所匯出的，匯入後請再次確認所匯入的內容是否與預期相符。

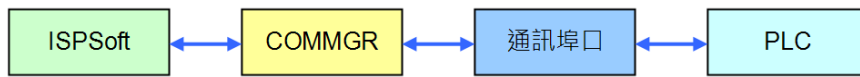
2.3.3 匯出程式到文字格式

透過 ISPSOft 所提供的功能，使用者可將程式匯出到文字格式以便於申請專利，匯出的文字格式無法再轉回程式，於功能工具列中點選 **工具 (I)** > **匯出程式到文字格式**，之後於檔案視窗中輸入欲匯出的檔名後按下「存檔」即可，如下圖。



2.4 通訊設定

下圖為 ISPSOft 與台達 PLC 之間的通訊架構，有別於以往的連線方式，ISPSOft 則是統一透過**通訊管理員 - COMMGR** 來做為通訊的介面。而在此節的內容中，我們將會介紹如何在 ISPSOft 與 PLC 之間逐步地建立起正常的通訊連線，並完成基本的測試。

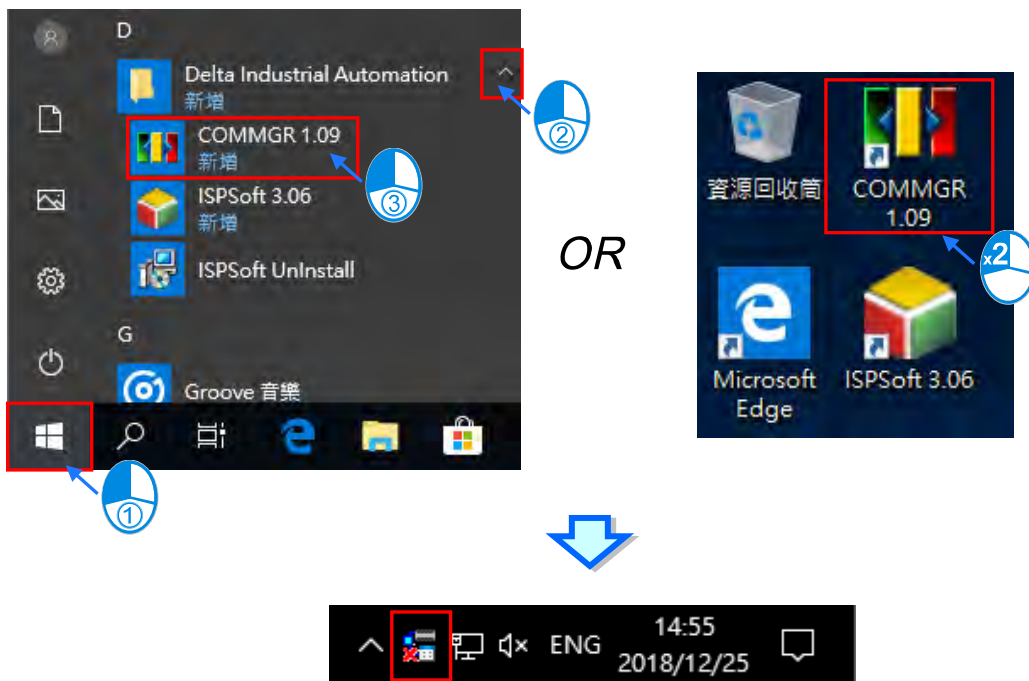


- *1. 關於通訊管理員 - COMMGR 的簡介與安裝方式，請參考手冊第 1.2 節的相關說明。
- *2. ISPSOft 在 V2.0 版以上才開始採用 COMMGR 的通訊架構，較早期的版本則仍使用傳統的連線方式。

2.4.1 通訊管理員 COMMGR 的啟動與關閉



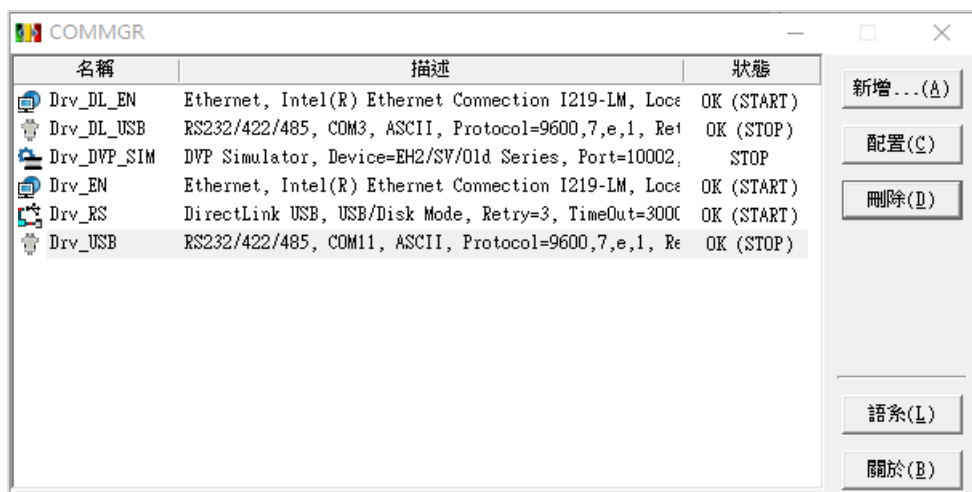
順利完成 **COMMGR** 的安裝後，系統便會自動於**程式集**當中建立一個執行捷徑，直接點擊該捷徑即可啟動 **COMMGR**；而除了首次安裝完成後，使用者需自行將其啟動外，之後只要每次電腦重新開機，系統便會自動啟動 **COMMGR**，並將其常駐在 Windows 的系統列當中；但若系統列中沒有出現該圖示時，請直接經由程式集當中的捷徑自行啟動 **COMMGR**。



當順利啟動 **COMMGR** 之後，於 Windows 的系統列當中便可發現 **COMMGR** 的圖示，於該圖示上雙擊滑鼠左鍵，或是點擊滑鼠右鍵後點選「開啟」即可開啟 **COMMGR** 的管理視窗。



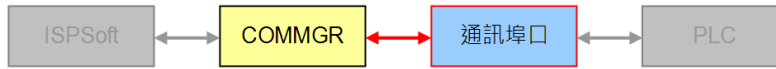
下圖為 **COMMGR** 的操作畫面，視窗的中間部份會列出所有已經建立好的通訊驅動，ISPSOft 便是透過指定其中的某個驅動來與 **PLC** 之間建立連結；而視窗右側則是管理這些驅動的操作項，詳細操作方式將於下一節的內容當中另行介紹。



當使用者按下視窗右上角的 **X** 鍵或是 **-** 鍵時都可將操作視窗關閉，不過 **COMMGR** 仍會繼續常駐在 Windows 的系統列當中；若要完全關閉 **COMMGR** 時，請於系統列的 **COMMGR** 圖示上點擊滑鼠右鍵，之後選擇「退出」即可將其關閉。

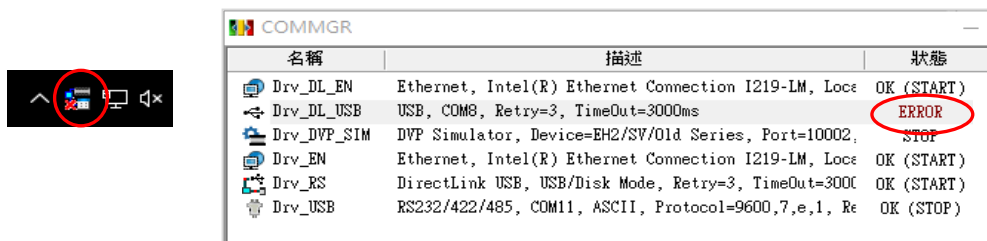


2.4.2 COMMGR 的驅動管理



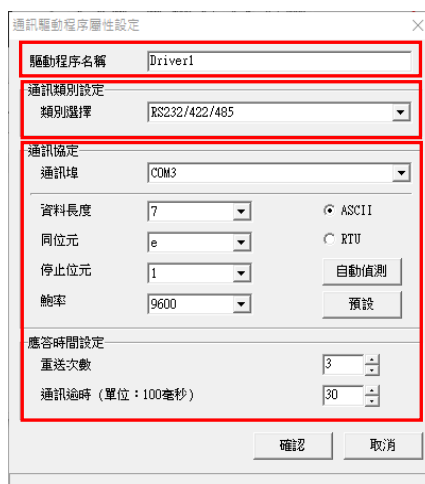
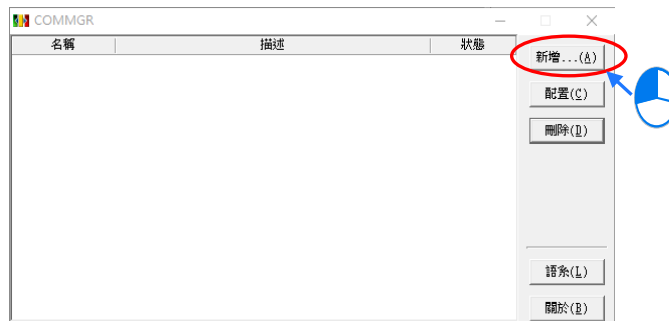
2

COMMGR 中的**驅動**所代表的意義為程式與電腦通訊埠之間的連線通道。當使用者新增了一個**驅動**且設定好通訊參數，**COMMGR** 便會與該**驅動**中所指定的通訊埠建立起一個連線通道，且之後只要電腦重新啟動時，**COMMGR** 便會將該**驅動**自動啟動；而當所建立的**驅動**發生連線通道無法正常使用的情況，例如原本存在的網路卡或是 USB 連接線被移除，**COMMGR** 便會自動將該**驅動**停止並標示為**ERROR** 的狀態，而此時位於 Windows 系統列上的 **COMMGR** 圖示也會出現一個紅色的 X 號；而一旦連線恢復正常，**驅動**的狀態也將恢復為 **OK**。



2.4.3 建立連線通道 - 驅動的建立

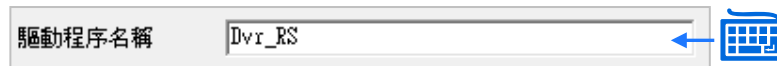
於 **COMMGR** 右側的「新增」點擊滑鼠左鍵即可開啟通訊參數的設定視窗。



設定的步驟簡述如下：

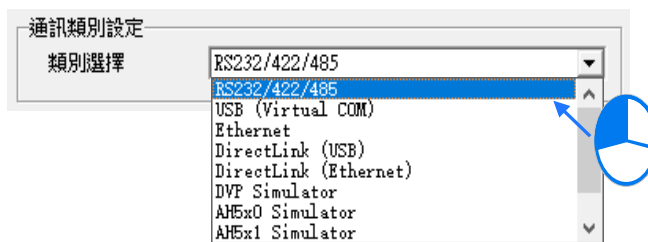
(1) 設定驅動名稱

於「驅動程序名稱」欄位中輸入此驅動的識別名稱，至多可輸入 31 個字元，而輸入的內容除了底線之外，不可使用其它如*、#、?、\、%、@ ...等特殊符號。



(2) 設定連線型態

於「通訊類別設定」的「類別選擇」欄位上點擊滑鼠左鍵，之後於下拉選單中選擇欲使用的連線型態即可。下列則是目前 **COMMGR** 所支援的幾種連線型態。



➤ RS232/422/485

透過電腦的 COM Port 與 PLC 主機進行通訊。

➤ USB (Virtual COM)

部分 PLC 主機本身有提供 USB 的接口，因此電腦與 PLC 主機之間可直接透過 USB 連接。使用此種連線方式之前，請確認電腦中已安裝該主機的 USB 驅動程式，安裝方式請參考 附錄 A 的相關說明或各機種的使用手冊。

➤ Ethernet

透過電腦的乙太網路與 PLC 主機進行通訊。

➤ DirectLink (USB) & DirectLink (Ethernet)

此為台達 HMI 所提供的接續功能。當 PLC 與 HMI 在正常連線的狀態下，電腦便可透過 USB 或乙太網路與 HMI 連線後，再間接地與 PLC 主機進行通訊。詳細的連線設定方式以及使用上的注意事項，請參考台達 HMI 的相關使用手冊。

➤ DVP Simulator & AH5x0 Simulator & AH5x1 Simulator & AS300 Simulator & AS200 Simulator

各機種模擬器的虛擬通道，必須在 ISPSOFT 中使用相對應的機種才能進行連接。

(3) 設定通訊參數

針對所選的連線型態來設定其通訊參數，而不同的連線型態，其可設定的通訊參數也會不同。關於各種型態的參數設定方式將會於接下來的幾個小節當中陸續介紹。

通訊協定

通訊埠: COM3

資料長度: 7

同位元: e

停止位元: 1

鮑率: 9600

ASCII

RTU

自動偵測

預設

應答時間設定

重送次數: 3

通訊逾時 (單位: 100毫秒): 30

2.4.3.1 RS232/422/485 通訊參數設定

欲建立 RS232/422/485 型態的驅動時，在開啟驅動設定視窗前請先確認下列兩點事項：

- (a) 電腦中已成功安裝相關驅動程式。
- (b) 電腦與 PLC 主機之間目前已連接在一起，且已正常運作。

① 驅動程序名稱: Driver1

通訊類別設定

② 類別選擇: RS232/422/485

通訊協定

③ 通訊埠: COM3

資料長度: 7

同位元: e

停止位元: 1

鮑率: 9600

ASCII

RTU

④

自動偵測

預設

應答時間設定

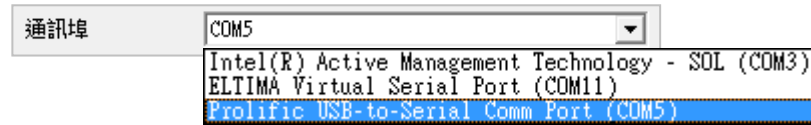
⑤

重送次數: 3

⑥ 通訊逾時 (單位: 100毫秒): 30

- (1) 於「驅動程序名稱」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「通訊類別設定」的「類別選擇」欄位下拉選擇「RS232/422/485」的連線型態。

- (3) 於「**通訊埠**」欄位下拉選擇欲建立連線通道的通訊埠口，而下拉選單中將會顯示完整的 COM Port 編號與裝置名稱，顯示的內容則會與 Windows **裝置管理員**中的內容一致，因此，需先確認上述 (a)(b) 兩點事項之後才能選擇到此 COM Port。



- (4) 選擇通訊的資料格式為 **ASCII** 或 **RTU**。
- (5) 設定埠口的通訊協定，且所設定的通訊協定並須與連接的連線裝置一致，而按下「**預設**」可將通訊協定參數回到系統預設值。

若不確定連線裝置的通訊協定時，可在該裝置與電腦連線時按下「**自動偵測**」即可自動偵測連線裝置的參數，當偵測成功時，該連線裝置的通訊協定參數，會被回填於視窗中的對應欄位；不過進行自動偵測時，系統將不會偵測 **COM Port** 與 **ASCII / RTU** 這兩個參數，因此在執行自動偵測前，必須先確定這兩個參數才可進行操作。

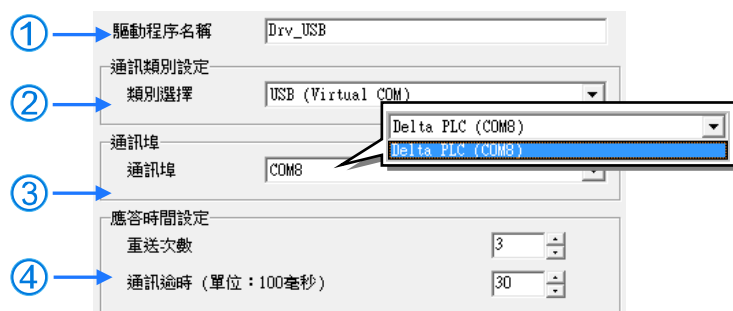
- (6) 設定通訊回應的相關參數。其中「**重送次數**」參數是設定連線錯誤時的重試次數，而「**通訊逾時**」參數則是設定重試的時間間隔。

*.若有變更電腦連接的硬體裝置，則必須關掉通訊驅動程序屬性設定頁面，再重新開啟才能取得新的硬體通訊裝置。

2.4.3.2 USB (Virtual COM) 通訊參數設定

欲建立 USB 型態的驅動時，在開啟驅動設定視窗前請先確認下列兩點事項：

- (a) 電腦中已成功安裝該主機的 USB 驅動程式。
- (b) 電腦與 PLC 主機之間目前已透過 USB 連接線連接在一起，且已正常運作。



- (1) 於「**驅動程序名稱**」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「**通訊類別設定**」的「**類別選擇**」欄位下拉選擇「**USB (Virtual COM)**」的連線型態。
- (3) 於「**通訊埠**」欄位下拉選擇欲建立連線通道的通訊埠口，而若上述 (a)(b) 兩點事項已確認無誤，則此時於下拉選單中便會顯示已連線的主機裝置名稱與其 COM Port 編號。
- (4) 設定通訊回應的相關參數。其中「**重送次數**」參數是設定連線錯誤時的重試次數，而「**通訊逾時**」參數則是設定重試的時間間隔。

*.關於 PLC 主機的 USB 驅動程式，其相關說明請參考手冊中的 附錄 A 或各主機的操作手冊。

2.4.3.3 DirectLink (USB) 通訊參數設定

① 驅動程序名稱: Drv_DL_USB

② 通訊類別設定
類別選擇: DirectLink (USB)

HMI USB Mode
 USB / Disk Mode
 CDC Mode
 COM Port: []

③

應答時間設定
 重送次數: 3
 通訊逾時 (單位: 100毫秒): 30

④

- (1) 於「驅動程序名稱」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「通訊類別設定」的「類別選擇」欄位下拉選擇「**DirectLink (USB)**」的連線型態。
- (3) 選擇 HMI 的 USB 傳輸模式。若選擇「**CDC Mode**」時，於「**COM Port**」欄位下拉選擇欲建立連線通道的通訊埠口。詳細的連線設定方式以及使用上的注意事項，請參考台達 HMI 的相關使用手冊。
- (4) 設定通訊回應的相關參數。其中「**重送次數**」參數是設定連線錯誤時的重試次數，而「**通訊逾時**」參數則是設定重試的時間間隔。

2.4.3.4 Ethernet 通訊參數設定

① 驅動程序名稱: Drv_EN

② 通訊類別設定
類別選擇: Ethernet

乙太網路卡
 描述: Intel(R) Ethernet Connection I219-LM
 192.168.2.95

③

IP 位址設定
 增加 刪除 搜尋

IP位址	埠號	名稱	機種
192.168.2.56	502	Main_Controller	AHCPU530-EN
192.168.2.40	502	CMC-EIP01	CMC-EIP01
192.168.2.211	502	DVP32ES2-E	ES2-E
192.168.2.210	502	DVP12SE	SE

④

應答時間設定
 重送次數: 3
 通訊逾時 (單位: 100毫秒): 30

⑤

- (1) 於「驅動程序名稱」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「通訊類別設定」的「類別選擇」欄位下拉選擇「**Ethernet**」的連線型態。

- (3) 於「乙太網路卡」的欄位下拉選擇欲建立連線通道的網路卡，欄位左下會顯示目前的 IP 位址。
- (4) 於「IP 位址設定」的區域中，建立通訊裝置的 IP 位址與名稱。
 - 按下「搜尋」後便會開始搜尋連接在相同網域的裝置，並將結果顯示於列表中。
 - 按下「增加」按鈕可於下方的列表中新增一組 IP 位址，之後便可於各欄位中輸入相關資訊。
 - ❶ IP 位址：裝置的 IP 位址。
 - ❷ 埠號：通訊時使用的埠號，預設為 502。
 - ❸ 名稱：裝置的名稱。
 - ❹ 機種：裝置的機種，只能由搜尋功能顯示。
 - 選取某組 IP 後按下「刪除」按鈕或直接按下鍵盤的【DEL】鍵即可將該組 IP 於列表中刪除。
- (5) 設定通訊回應的相關參數。其中「重送次數」參數是設定連線錯誤時的重試次數，而「通訊逾時」參數則是設定重試的時間間隔。

*.若有變更電腦連接的網路裝置，則必須關掉通訊驅動程序屬性設定頁面，再重新開啟才能取得新的網路裝置。

2.4.3.5 DirectLink (Ethernet) 通訊參數設定

① 驅動程序名稱: Drv_DL_EN

② 通訊類別設定: 類別選擇: DirectLink (Ethernet)

③ 乙太網路卡: 描述: Intel(R) Ethernet Connection I219-LM
192.168.2.95

④ IP 位址設定

IP位址	埠號	名稱	機種
192.168.2.220	502	HMI_1	

⑤ 應答時間設定: 重送次數: 3, 通訊逾時 (單位: 100毫秒): 30

- (1) 於「驅動程序名稱」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「通訊類別設定」的「類別選擇」欄位下拉選擇「DirectLink (Ethernet)」的連線型態。
- (3) 於「乙太網路卡」的欄位下拉選擇欲建立連線通道的網路卡，欄位左下會顯示目前的 IP 位址。
- (4) 於「IP 位址設定」的區域中，建立通訊裝置的 IP 位址與名稱。
 - 按下「搜尋」後便會開始搜尋連接在相同網域的裝置，並將結果顯示於列表中。
 - 按下「增加」按鈕可於下方的列表中新增一組 IP 位址，之後便可於各欄位中輸入相關資訊。
 - ❶ IP 位址：裝置的 IP 位址。
 - ❷ 埠號：通訊時使用的埠號，預設為 502。

- ③ 名稱：裝置的名稱。
- ④ 機種：裝置的機種，只能由搜尋功能顯示。

➢ 選取某組 IP 後按下「刪除」按鈕或直接按下鍵盤的【DEL】鍵即可將該組 IP 於列表中刪除。

- (5) 設定通訊回應的相關參數。其中「重送次數」參數是設定連線錯誤時的重試次數，而「通訊逾時」參數則是設定重試的時間間隔。

2

*.若有變更電腦連接的網路裝置，則必須關掉通訊驅動程序屬性設定頁面，再重新開啟才能取得新的網路裝置。

2.4.3.6 DVP Simulator 通訊參數設定

① 驅動程序名稱 Drv_DVP_SIM

② 通訊類別設定
類別選擇 DVP Simulator

模擬器
③ 通訊埠號 10003

裝置設定
④ 裝置 EH2/SV/Old Series

應答時間設定
⑤ 重送次數 3
通訊逾時 (單位: 100毫秒) 30

- (1) 於「驅動程序名稱」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「通訊類別設定」的「類別選擇」欄位下拉選擇「DVP Simulator」的型態。
- (3) 於此輸入指定的通訊埠號。
- (4) 設定 DVP 模擬器的種類，分為「EH2/SV/Old Series」、「EH3/ EH3-L/SV2」、「ES2/EX2/SA2/SX2/MC」、「SE」及「SS2」。
- (5) 設定通訊回應的相關參數。其中「重送次數」參數是設定連線錯誤時的重試次數，而「通訊逾時」參數則是設定重試的時間間隔。

*. ES、SS、EX、VFD、EC 機種專案，下載進 DVP 擬器器，上載回來時會變為 EH 機種專案。

*. TP 機種專案請選擇 SS2 機種模擬器。

2.4.3.7 AH5x0 與 AH5x1 Simulator 通訊參數設定

- (1) 於「**驅動程序名稱**」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「**通訊類別設定**」的「**類別選擇**」欄位下拉選擇「**AH5x0 Simulator**」或「**AH5x1 Simulator**」的機種。
- (3) 於此輸入指定的通訊埠號。
- (4) 設定通訊回應的相關參數。其中「**重送次數**」參數是設定連線錯誤時的重試次數，而「**通訊逾時**」參數則是設定重試的時間間隔。

2.4.3.8 AS Simulator 通訊參數設定

- (1) 於「**驅動程序名稱**」欄位輸入識別名稱，而輸入的內容除了底線之外，不可使用其它特殊符號。
- (2) 於「**通訊類別設定**」的「**類別選擇**」欄位下拉選擇「**AS200 Simulator**」或「**AS300 Simulator**」的機種。
- (3) 於此輸入指定的通訊埠號。
- (4) 設定通訊回應的相關參數。其中「**重送次數**」參數是設定連線錯誤時的重試次數，而「**通訊逾時**」參數則是設定重試的時間間隔。

補充說明

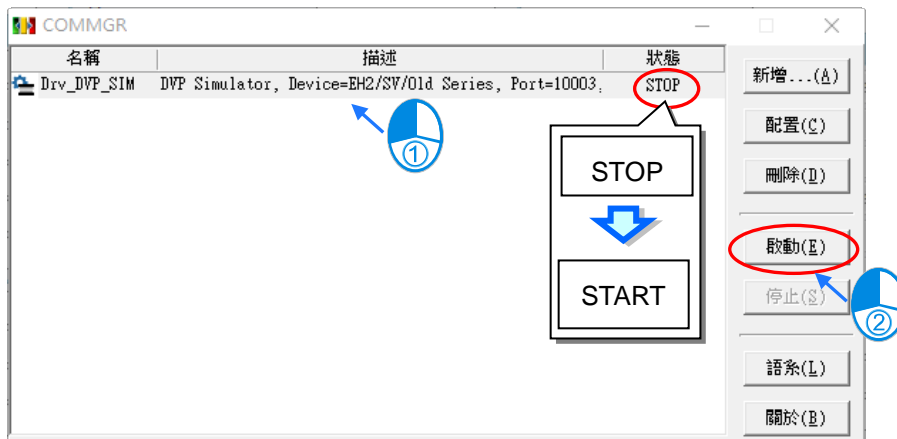
- (a) 模擬器的功能建議搭配 ISPSOFT V3.05 (含) 以上版本使用。
- (b) 模擬器不支援完整的功能跟指令，使用上需要注意。

2.4.4 建立連線通道 - 驅動的啟動與停止

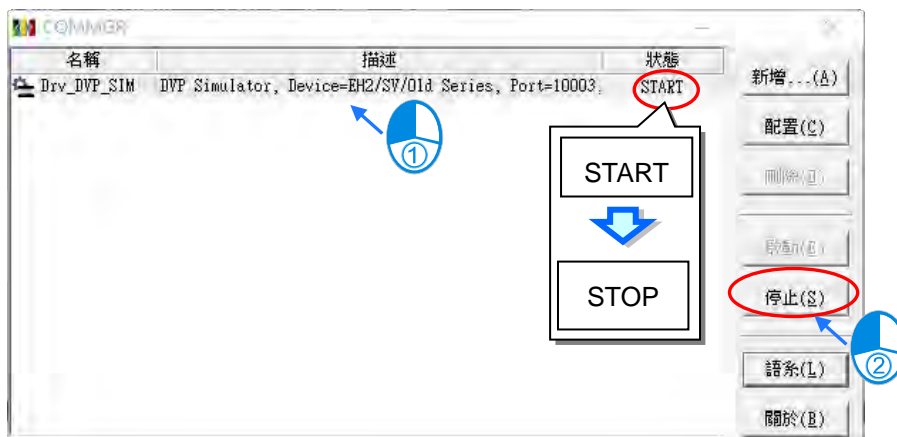
對於 USB 連接的驅動，只要裝置與電腦的連線正常，COMMGR 即會自動偵測連線，使用者無須任何操作。



模擬器驅動必須先啟動才能使用，需要先在 COMMGR 的管理列表上點選該模擬器驅動，之後再於右側的「啟動 (E)」點擊滑鼠左鍵，模擬器驅動啟動成功會顯示 **START**。

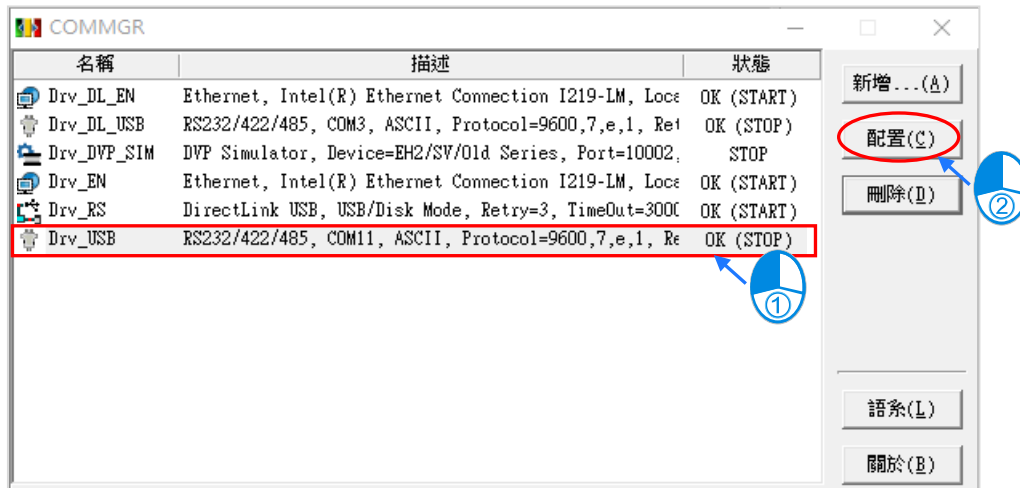


而欲停止模擬器驅動時，只要先在 COMMGR 的管理列表上點選該驅動，之後再於右側的「停止 (S)」鍵上點擊滑鼠左鍵，待該驅動的 Status 顯示為 **STOP** 後即代表已經停止。

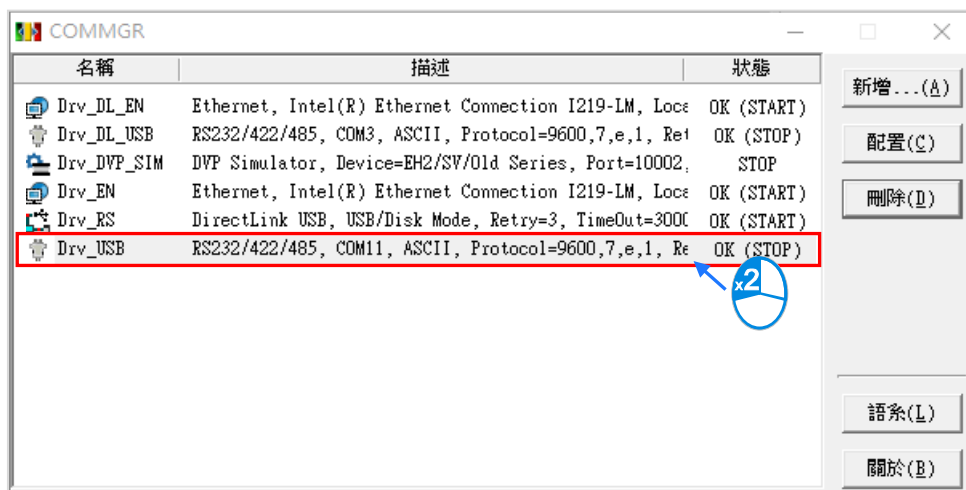


2.4.5 建立連線通道 - 驅動的修改與刪除

欲修改某個驅動的參數時，請先於列表上點選該驅動後點擊右側的「配置」按鈕，或是直接於該驅動上雙擊滑鼠左鍵，之後便會開啟設定視窗，即可修改此筆資料。

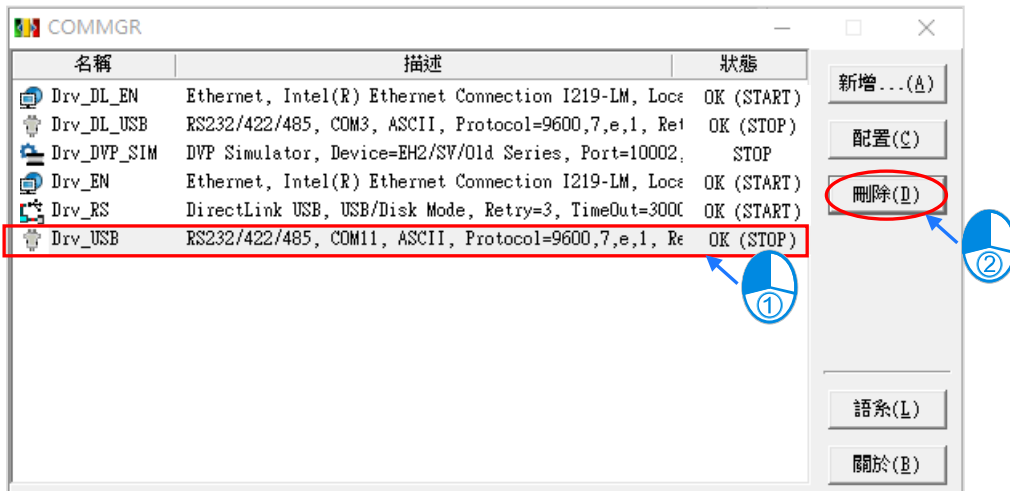


OR



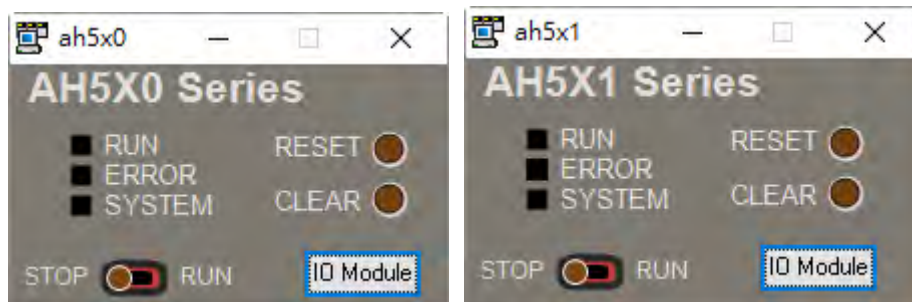
欲刪除某個驅動時，請先於列表上點選該驅動後點擊右側「刪除」按鈕或直接按下鍵盤的【DEL】鍵即可將其刪除。

2



2.4.6 建立連線通道 - 模擬器的面板操作

當模擬器的狀態顯示為 **START** 時，AH5x0 與 AH5x1 模擬器會彈出一個操作面板，用來查看目前模擬器的狀態燈號，並且可以透過 **STOP** 和 **RUN** 來控制模擬器的停止與運行，如下圖。

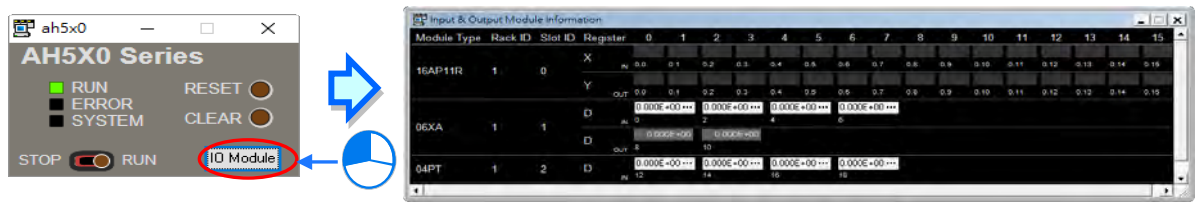


模擬器的「IO Module」按鈕則可開啟**模組 IO 面板**，使用此功能之前，必須先在 HWCONFIG 中配置模組並參考第 3.8.4 節將參數下載至主機，如下圖。



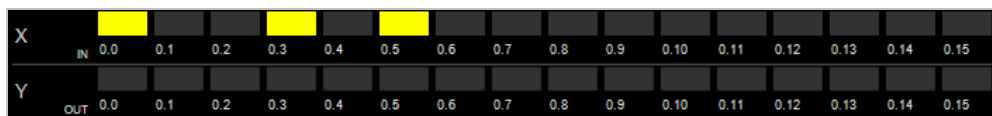
資訊: 背板 1						
插槽...	名稱	MDS版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHP805-5A	-	電源模組	None	None	
-	AHCPU530-EN	01.00.00	基本型 CPU 模組，內建 Ethernet、RS485、USB 與	None	None	
0	AH16AP11R-5A	-	8 點數位直流輸入；8 點數位交/直流輸出	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1	AH06XA-5A	01.00.00	4 通道類比輸入 16 位元，2 通道類比輸出 16 位元	D0 ~ D7	D8 ~ D11	
2	AH04PT-5A	01.00.00	4 通道 3/4 線式鉑金電阻溫度感測器 0.1°C/0.1°F	D12 ~ D19		

再將模擬器切換至 RUN 運行，接著就可以在**模組 IO 面板**看到跟 HWCONFIG 一樣配置的模組與其狀態，如下圖。

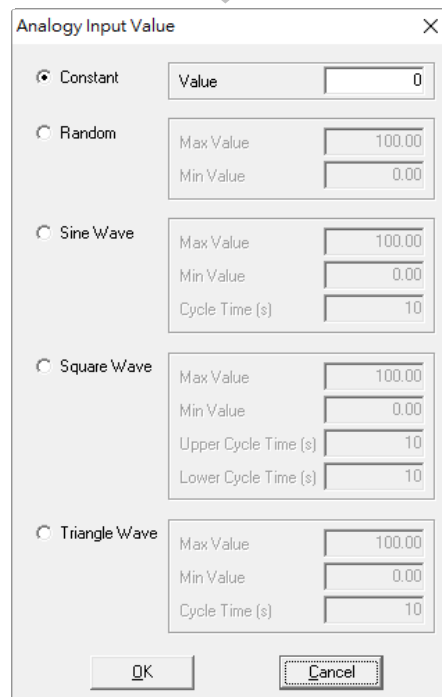


2

下圖是一個數位 IO 的暫存器操作區域，其中 Y 裝置的部份僅可用來檢視輸出狀況，但無法進行任何操作；而 X 裝置的部份則可直接用滑鼠點擊各個接點而變更其 ON/OFF 狀態，以模擬外部的輸入。



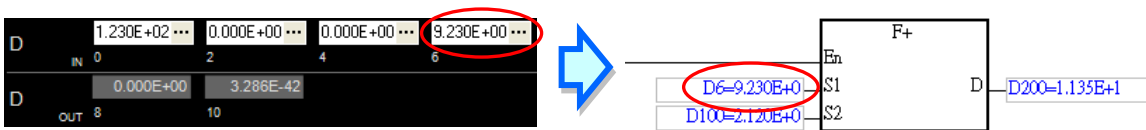
下圖是一個類比 IO 的暫存器操作區域，其中的每個欄位皆是 32 位元浮點數的格式，並對應實際模組的各個通道；下方的 D8 與 D10 是對應類比輸出通道的 D 裝置，輸出僅可用來檢視；上方的 D0、D2、D4 與 D6 則是對應類比輸入通道的 D 裝置，可於此處點擊 按鈕，開啟設定視窗輸入數值，如下圖。



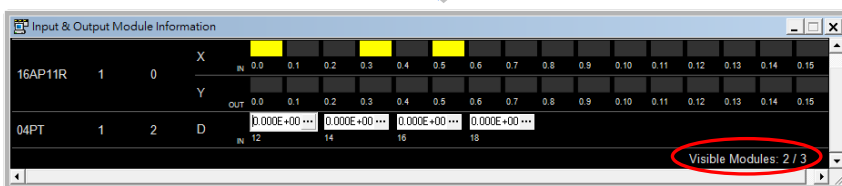
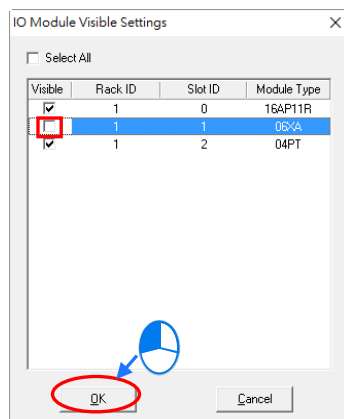
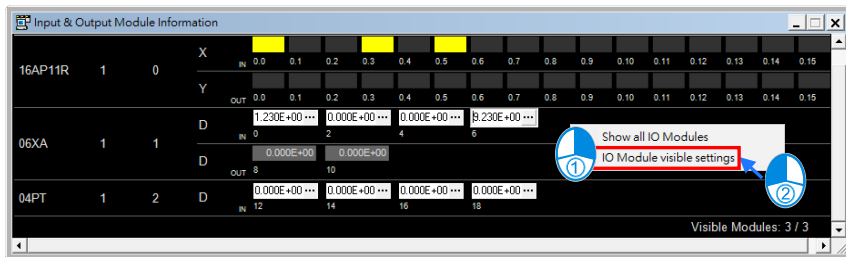
2

信號種類	說明
Constant	為一個固定數值的常數信號。
Random	可指定最大值與最小值的隨機信號。
Sine Wave	可指定最大值、最小值與信號週期的正弦波信號。
Square Wave	可指定最大值、最小值、ON 時間與 OFF 時間的週期性矩形波。
Triangle Wave	可指定最大值、最小值與信號週期的三角波信號。

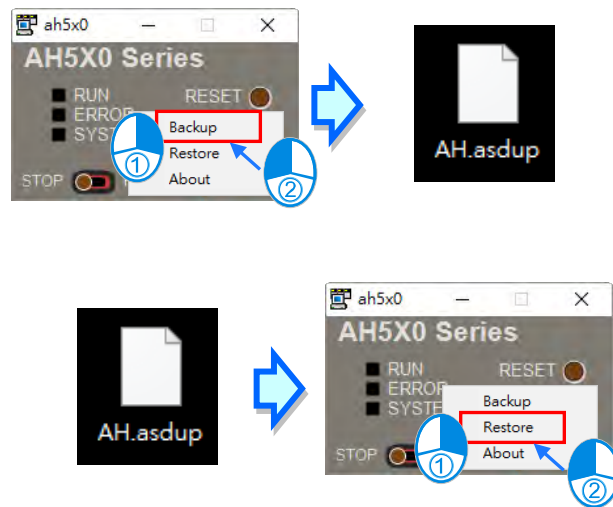
設定完畢後，該欄位（通道）便會依據所設定的信號模式而持續送出資料給 COMMGR，而 ISPSOft 便可透過 COMMGR 來擷取到模擬器的信號，並可藉此進行相關的測試。



於模組 IO 面板按下滑鼠右鍵，選取「Show All IO Module」可以顯示所有模組，若要隱藏模組就點選「IO Module Visible settings」，接著於設定視窗中取消勾選欲隱藏的模組項目，完成後按下「OK」即可由視窗的右下角，得知目前的顯示狀況，如下圖。



在 AH5x0 與 AH5x1 模擬器，還可以透過模擬器本身所提供的備份功能，將參數與程式備份成一個 .asdup 檔案，再透過這個檔案還原到其他相同的模擬器如下圖。



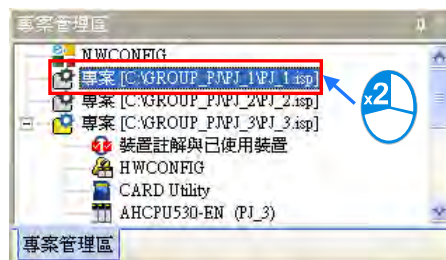
2

2.4.7 建立 ISPSOft 與 COMMGR 之間的連結



在 **COMMGR** 中成功建立驅動並啟動後，接下來便可於 **ISPSOft** 中指定欲使用的驅動，而完成設定後，在 **ISPSOft** 與 **COMMGR** 之間便建立了連結。設定時，每個**單機專案**都必須單獨指定其通訊方式，因此在**群組專案**的架構下，使用者必須逐一開啟每個**單機專案**並完成各自的設定。

(1) 首先請於群組中將欲設定通訊方式的專案啟動，但若是在**單機專案**的架構下則可略過此步驟。

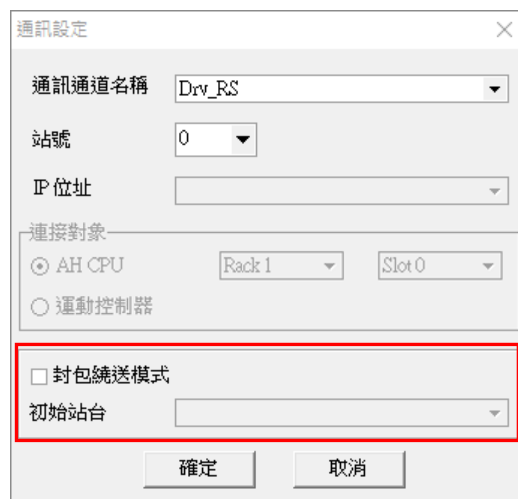


- (2) 於 ISPSOft 中點擊功能工具列的 **工具 (I)** > **通訊設定 (P)**，之後便會出現通訊設定視窗。

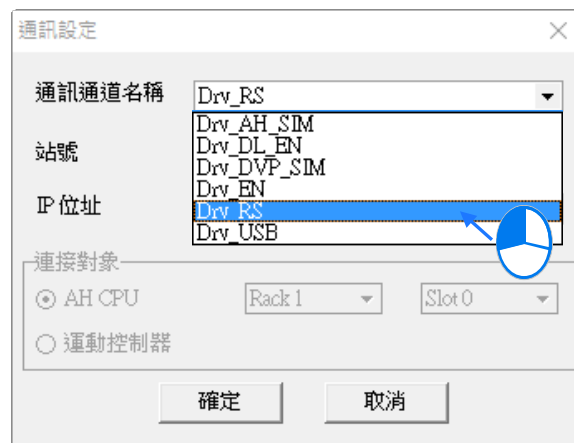
2



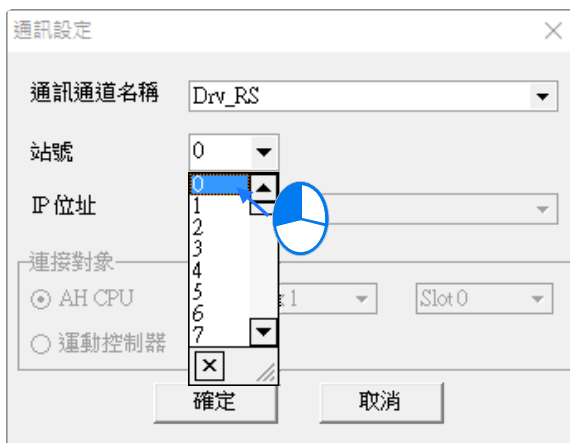
- (3) 當所設定的對象為**群組專案**下的 AH5x0 系列機種時，會增加「**封包繞送模式**」的項目，該選項需搭配 NWCONFIG 使用，相關操作與說明請參考第 20 章。



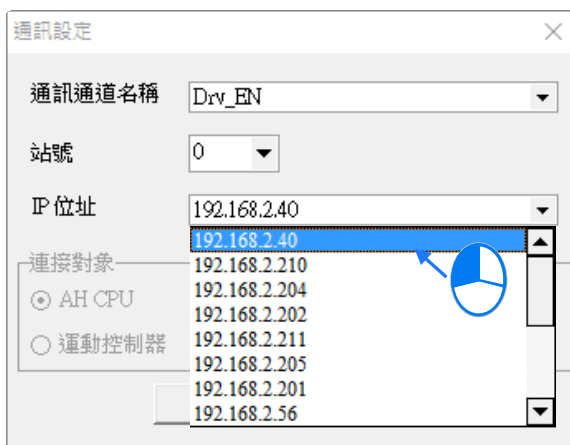
- (4) 在通訊設定視窗中的「**通訊通道名稱**」，下拉選擇欲連結的驅動。



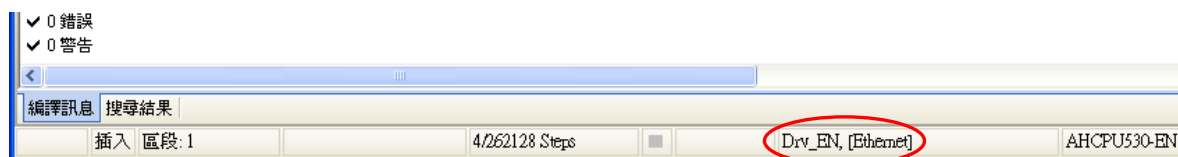
- (5) 於「站號」下拉選擇與電腦實際連線的 PLC 主機站號，也可直接使用站號 0 (廣播站號)。



- (6) 若驅動型態為 Ethernet 或 DirectLink (Ethernet) 時，於「IP 位址」下拉選擇欲連接 PLC 主機的 IP 位址。



- (7) 完成所有設定後按下「確定」。同時於 ISPSOft 的狀態列中，會顯示目前所連結的驅動資訊。



補充說明

如果 COMMGR 設定有變更，則 ISPSOft 的通訊設定，必須要重新選擇一次，才能正常連接，也可以透過 **COMMGR** 按鈕來呼叫 COMMGR 設定視窗，COMMGR 版本須提升至 V1.09.05 以上，才能搭配此便利功能。

2



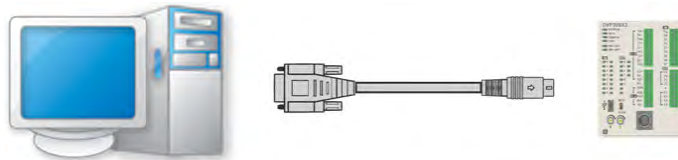
2.4.8 連接主機與通訊埠口



接下來便可透過通訊連接線將主機與指定的通訊埠口連接在一起。下述將列舉一些常用的連接方式與提醒事項，而更進一步的接線與設定方式則請自行參考各機種的操作手冊。

1. DVP 系列主機 (RS232)

直接以台達專用的 **PLC 通訊連接線** 與主機連接，而 COMMGR 驅動的型態則請選擇 RS232。**注意事項**：連線前請確認 COMMGR 驅動中所設定的 RS232 參數與 PLC 主機的設定一致。



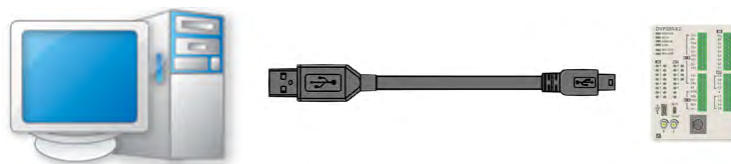
2. DVP - SX2 主機 (USB)

DVP 系列的 SX2 主機本身有提供 Mini USB (B-Type) 的通訊介面，使用者可直接以 USB 連接線將主機與電腦進行接續，但因 SX2 機種的 USB 架構是將 USB 轉 RS232 的電路內建於主機中，因此其內部運作仍是採用 RS232 的模式，所以 COMMGR 驅動的型態請選擇 RS232。

注意事項：

- (a) 請確認電腦中已正確安裝 SX2 機種的 USB 驅動程式，相關說明請參考 **附錄 A**。

- (b) 連線前請確認 COMMGR 驅動中所設定的 RS232 參數與 PLC 主機的設定一致。



3. AH/AS 系列主機 (USB)

直接透過 USB 連接線與主機進行接續，而其中主機的 USB 埠為 B-Type 的 Mini USB 型式；COMMGR 驅動的連線型態則請選擇 USB (Virtual COM)。

注意事項：請確認電腦中已正確安裝 AH/AS 機種的 USB 驅動程式，相關說明請參考 附錄 A。

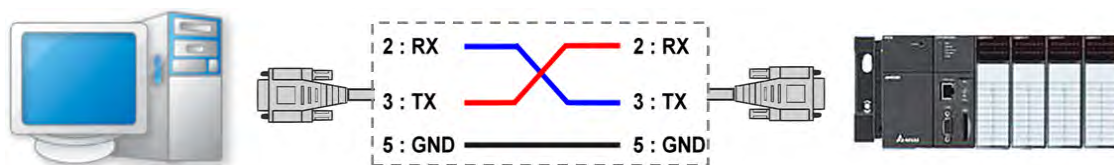


4. AH 系列主機 (RS232/422/485)

AH 系列主機的 COM 端口為標準型式，因此請使用 RX 與 TX 跳線的 RS232 連接線，或透過具備跳線功能的轉接器來與電腦進行連接，而 COMMGR 驅動的連線型態則請選擇 RS232/422/485。

注意事項：

- (a) 連線前請確認 COMMGR 驅動中所設定的 RS232 參數與 PLC 主機的設定一致。
- (b) AH 系列機種的 COM 介面可規劃為 RS232 / RS485 / RS422 等多種模式，連線前請確認該介面目前所規劃的模式是否正確。(請參考後續的補充說明)



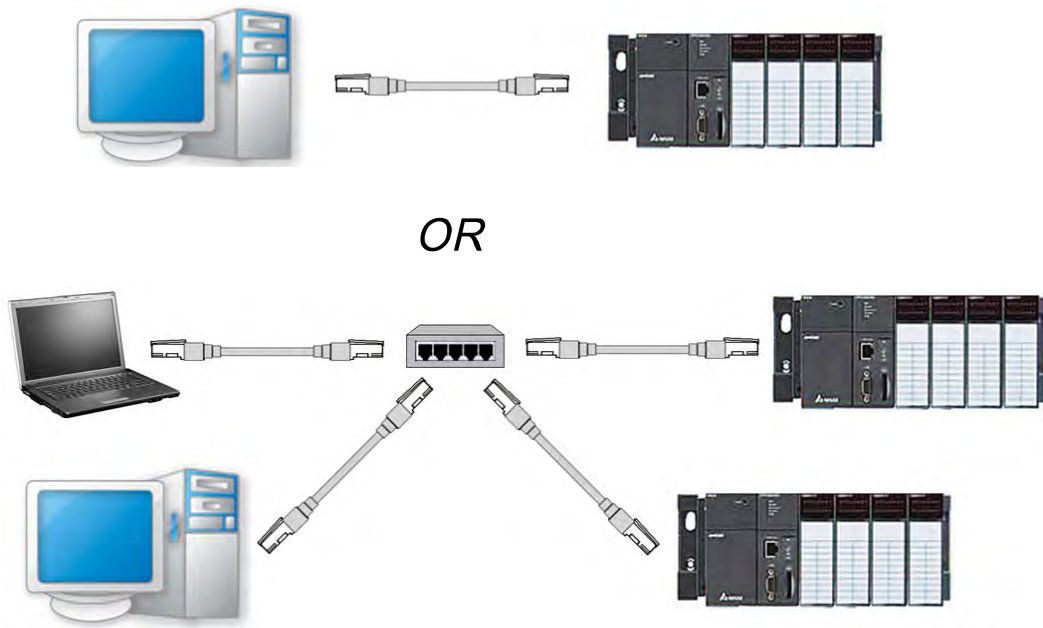
5. AH/AS 系列主機、DVPxxMC、AS5xx 與 DVP-SE (Ethernet)

AHCPU5xx-EN 系列、AHxxEMC 系列、AS 系列、DVPxxMC、AS5xx 及 DVP-SE 主機有內建 Ethernet 的 RJ - 45 端口，並可透過乙太網路與電腦進行連接，可以透過 HUB 將電腦與 PLC 主機連接至同一網域上，也可以用網路線直接將電腦與 PLC 主機直接對接。至於 COMMGR 驅動的型式則請選擇 Ethernet。

注意事項：

- (a) 連線前請確認所架構的網路處於正常工作的狀態。
- (b) 確認 CPU 主機本身的乙太網路參數是否正確，請參考後續的補充說明。

2

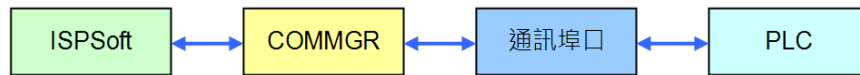


補充說明

各機種通訊參數的出廠預設值請參閱各機種操作手冊。

2.4.9 實際連線測試

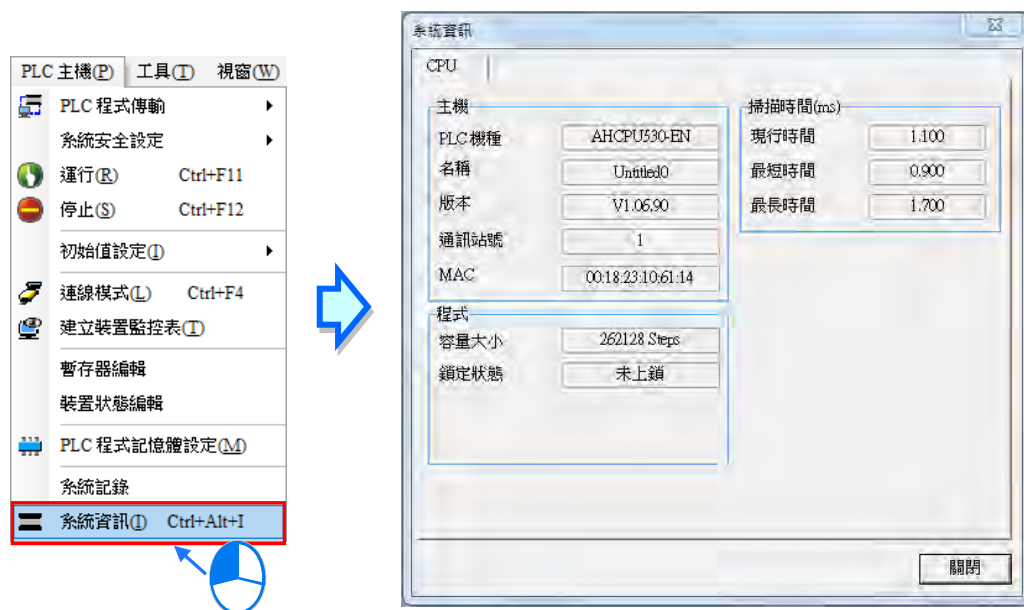
若已完成上述的所有步驟後，接著便可透過簡單的測試來驗證是否可正常與主機進行連線。



連線之前請再次確認下列事項。

- PLC 主機與電腦已正確透過通訊連接線連接，主機也已送電並處於正常運作的狀態。
- 提供通訊的裝置，如電腦的網路卡、網路 HUB、序列埠...等，已處於正常工作的狀態。
- 針對欲使用的 COMMGR 驅動，請確認相關參數的設定正確無誤動，且驅動處於 **OK** 狀態。
- 確認在 ISPSOft 的「通訊設定」中，其所指定的驅動、站號與 IP 位址等設定皆正確無誤。

確認完上述事項後，接著請於 ISPSOft 的功能工具列中，點選 **PLC 主機 (P)** > **系統資訊 (I)**，此時便會彈出系統資訊，若無彈出系統資訊表示連接失敗，請檢查(a)~(d)項目後再試。



MEMO

2

3

第3章 PLC 系統的配置與設定

目錄

3.1 硬體規劃工具 - HWCONFIG 4.0.....	3-6
3.1.1 進入 HWCONFIG 的第一步	3-7
3.1.2 HWCONFIG 的環境介紹	3-10
3.1.2.1 視窗標題與狀態列	3-11
3.1.2.2 功能工具列-檔案	3-11
3.1.2.3 功能工具列-編輯	3-18
3.1.2.4 功能工具列-檢視	3-24
3.1.2.5 功能工具列-通訊	3-24
3.1.2.6 功能工具列-工具	3-32
3.1.2.7 功能工具列-說明	3-41
3.1.2.8 功能工具列-常用	3-42
3.1.2.9 專案樹	3-42
3.1.2.10 產品列表	3-43
3.1.2.11 輸出視窗	3-44
3.1.2.12 硬體配置區	3-45
3.1.2.13 編輯區	3-46
3.1.3 進階功能操作	3-47
3.1.3.1 新增模組	3-47
3.1.3.2 變更模組	3-52
3.1.3.3 模組的搬移	3-53
3.1.3.4 備註編輯	3-54
3.1.3.5 硬體配置區-重設配置	3-56
3.1.3.6 硬體配置區-顯示模組詳細訊息	3-57
3.1.3.7 硬體配置區-輸出/輸入裝置重新排序	3-57
3.1.3.8 硬體配置區-改變遠端裝置位址	3-58
3.1.3.9 硬體配置區-重置大小	3-59
3.1.3.10 編輯區-模組內部參數的匯出與匯入	3-60
3.1.3.11 編輯區-模組內部參數的更新	3-61
3.1.3.12 顯示/隱藏功能	3-61

3.1.3.13	資訊檢視與模組診斷	3-62
3.1.3.14	線上變更模組狀態	3-62
3.1.3.15	遠端模組配置	3-64
3.1.3.16	開啟外部軟體	3-66
3.1.3.17	IO-Link	3-66
3.1.3.18	符號設定	3-85
3.1.4	PLC 主機參數設定	3-89
3.1.4.1	系統設定-系統資訊	3-90
3.1.4.2	系統設定-系統參數	3-91
3.1.4.3	系統設定-裝置元件範圍設定	3-95
3.1.4.4	系統設定-輸入點濾波時間	3-96
3.1.4.5	系統設定-定位軸參數	3-97
3.1.4.6	系統設定-日光節約時間	3-98
3.1.4.7	COM 通訊埠設定	3-99
3.1.4.8	乙太網路基本設定	3-100
3.1.4.9	乙太網進階設定-IP 過濾器	3-101
3.1.4.10	乙太網進階設定-網路時間校正	3-102
3.1.4.11	乙太網進階設定-郵件	3-103
3.1.4.12	乙太網進階設定-通訊接口	3-108
3.1.4.13	乙太網進階設定- RTU 對應	3-109
3.1.4.14	功能卡 1 設定	3-111
3.1.4.15	功能卡 2 設定	3-111
3.1.4.16	台達設備參數還原功能	3-115
3.1.4.17	資料交換	3-120
3.2	硬體規劃工具 - HWCONFIG 3.0	3-121
3.2.1	HWCONFIG 的環境介紹	3-121
3.2.2	模組配置	3-125
3.2.2.1	新增模組	3-125
3.2.2.2	模組的位址配置	3-127
3.2.2.3	註解編輯	3-131
3.2.2.4	刪除模組	3-132
3.2.2.5	變更模組	3-134
3.2.2.6	模組的搜尋/取代功能	3-136

3.2.2.7	模組的複製/貼上.....	3-139
3.2.2.8	模組的剪下/貼上.....	3-141
3.2.2.9	模組的搬移	3-142
3.2.2.10	新增延伸背板及備援延伸背板	3-143
3.2.2.11	刪除延伸背板及備援延伸背板	3-147
3.2.2.12	變更背板	3-149
3.2.2.13	延伸背板及備援延伸背板的剪下/複製/貼上.....	3-151
3.2.2.14	延伸背板圖示及備援延伸背板圖示的搬移	3-156
3.2.2.15	重新排序輸入/輸出裝置範圍.....	3-157
3.2.3	主機與模組的版本管理	3-157
3.3	AH500 機種的 PLC 主機參數設定	3-160
3.3.1	開啟 PLC 主機的參數設定頁面	3-160
3.3.2	CPU 基本參數設定	3-161
3.3.2.1	CPU 基本參數 - 名稱	3-161
3.3.2.2	CPU 基本參數 - 系統	3-162
3.3.2.3	CPU 基本參數 - 停電保持裝置範圍	3-166
3.3.2.4	CPU 進階設定	3-168
3.3.3	序列埠 (COM Port) 參數設定	3-169
3.3.4	乙太網路 (Ethernet) 基本參數設定	3-170
3.3.5	乙太網路 (Ethernet) 進階參數設定	3-171
3.3.5.1	乙太網路進階參數 - 過濾器	3-171
3.3.5.2	乙太網路進階參數 - 網路時間校正.....	3-173
3.3.5.3	乙太網路進階參數 - 郵件.....	3-174
3.3.5.4	乙太網路進階參數 - 郵件發送條件.....	3-175
3.3.5.5	乙太網路進階參數 - 郵件位址與發送條件設定.....	3-180
3.3.5.6	乙太網路進階參數 - 通訊接口 (Socket)	3-181
3.3.5.7	乙太網路進階參數 - 網頁功能	3-183
3.3.5.8	乙太網路進階參數 - 資料交換	3-184
3.3.6	資料交換 - COM 與資料交換 - Ethernet	3-187
3.3.6.1	資料交換-COM1、資料交換-COM2 或資料交換-COM.....	3-188
3.3.6.2	資料交換-Ethernet	3-189
3.4	AHxxEMC 機種的 PLC 主機參數設定	3-190
3.4.1	開啟 PLC 主機的參數設定頁面	3-190

3.4.2	一般設定	3-192
3.4.2.1	一般設定 - 系統資訊	3-192
3.4.2.2	一般設定 - COM Port Setting	3-197
3.4.2.3	一般設定 - 乙太網路 - 基本設定	3-198
3.4.2.4	一般設定 - 乙太網進階設定	3-199
3.4.3	資料交換-COM2 與資料交換-Ethernet	3-200
3.4.3.1	資料交換-COM2	3-202
3.4.3.2	資料交換-Ethernet	3-203
3.5	DVP 系列機種的參數設定	3-203
3.5.1	DVP 主機的系統管理工具	3-203
3.5.2	停電保持設定	3-204
3.5.3	連線資訊	3-208
3.5.4	DVP-ES3 機種的主機參數設定	3-209
3.6	DVPxxMC 機種的 PLC 主機參數設定	3-211
3.6.1	開啟 PLC 主機的參數設定頁面	3-211
3.6.2	一般設定	3-212
3.6.2.1	一般設定-系統資訊	3-212
3.6.2.2	一般設定-CANopen	3-213
3.6.2.3	一般設定-CANMotion	3-214
3.6.2.4	一般設定-RS232	3-214
3.6.2.5	一般設定-RS485	3-215
3.6.2.6	一般設定-增量型編碼器 1	3-215
3.6.2.7	一般設定-增量型編碼器 2	3-216
3.6.2.8	一般設定-絕對型編碼器	3-217
3.6.2.9	一般設定-輸入點過濾時間	3-218
3.6.2.10	一般設定-運轉 / 停止設定	3-219
3.6.2.11	一般設定-乙太網路基本設定	3-220
3.7	模組參數設定	3-221
3.7.1	設定 AH5x0 與 AH5x1 模組內部參數	3-221
3.7.2	設定 AHxxEMC 機種模組內部參數	3-226
3.7.3	模組內部參數的匯出與匯入	3-229
3.7.4	模組內部參數的更新	3-229
3.7.5	智慧型模組參數設定	3-229

3.8 HWCONFIG 硬體參數管理及線上診斷	3-230
3.8.1 硬體規劃的儲存與列印	3-230
3.8.2 訂購清單	3-232
3.8.3 AH 系列機種的背板資訊列表	3-233
3.8.4 系統參數上下載	3-234
3.8.5 AH 系列機種的模組布局偵測	3-235
3.8.6 AH 系列機種的線上診斷	3-239
3.8.6.1 啟動線上操作模式	3-239
3.8.6.2 資訊檢視與模組診斷	3-240
3.8.6.3 線上變更模組狀態	3-241
3.8.6.4 監控表	3-244
3.9 萬年曆設定	3-245
3.10 PLC 記憶體設定 (清除程式&回復出廠值)	3-246

在 ISPSOft 中，DVP 系列主機可透過內建的功能或工具來對 PLC 主機或整個系統進行組態配置與參數設定；而其他系列的主機，則使用硬體規劃工具 HWCONFIG 來對 PLC 主機與模組作進行設定，基於 PLC 主機產品種類眾多，舊 HWCONFIG 的介面都沒有統一，新一代的 HWCONFIG 將提供全新的整合操作介面，讓使用者更快速方便的完成操作設定，新一代的 HWCONFIG 將會持續增加支援更多的機種，目前由 ISPSOft 開啟的 HWCONFIG 將依照下列表格自動開出相對應的版本，接著我們將逐一介紹關於系統組態與參數設定等功能的操作步驟與設定方式。

機種 軟體	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
HWCONFIG 4.0	V						
HWCONFIG 3.0		V	V	V	V	只支援 ES3	V

3.1 硬體規劃工具 - HWCONFIG 4.0

HWCONFIG 是 ISPSOft 的硬體規劃工具之一，其功能包括 CPU 主機參數的設定、模組背板的配置、模組參數的設定、硬體參數上下載及線上偵測與診斷功能。**HWCONFIG4.0** 提供使用者自行選擇顯示的主題風格，預設為淺色系，本章節使用淺色系做介紹，詳細設定方式，可以參考第 3.1.2.6 節環境設定。

⚠ 請注意！ 所有硬體規劃的參數於設定完畢後，必須被下載至主機中才可生效。

3.1.1 進入 HWCONFIG 的第一步

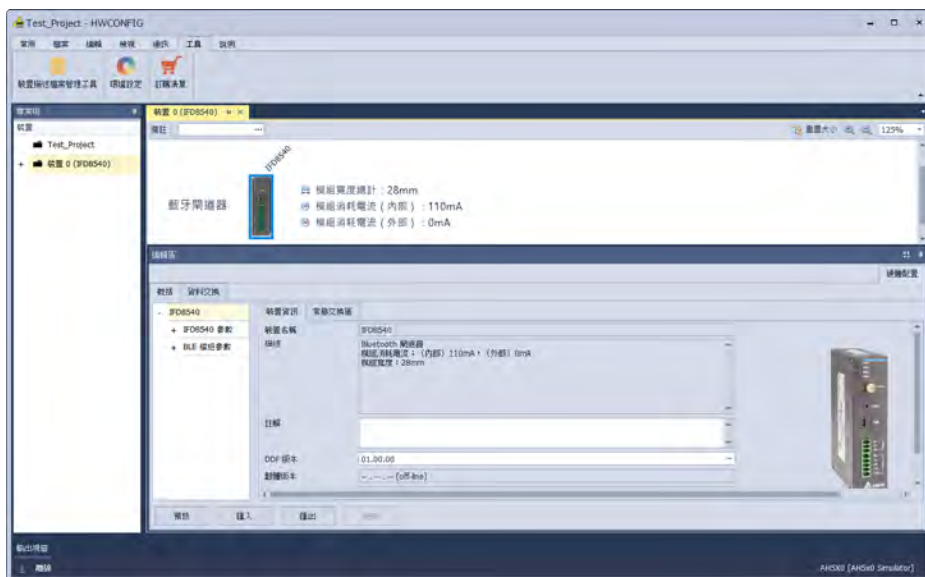
新的 HWCONFIG 支援獨立開啟，在安裝完成 ISPSoft 之後，便會自動在桌面中建立捷徑，點擊該捷徑即可啟動單機版 HWCONFIG 如下圖。開啟單機版 HWCONFIG，必須先建立一個專案，系統會自動跳出建立新專案的頁面，須於設定視窗中依序輸入「專案名稱」及「檔案路徑」，完成後按下「確定」。



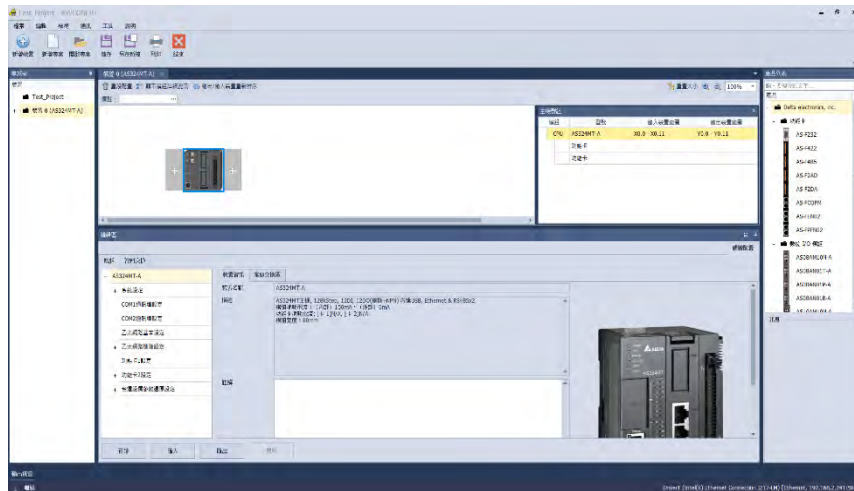
3

接著在選擇裝置視窗，點選「裝置類型」的下拉選單並選擇機種型式，再於要建立的 PLC 機種上雙擊滑鼠右鍵，也可以在選取 PLC 機種之後按下「確定」，即可進入操作畫面，如下圖。

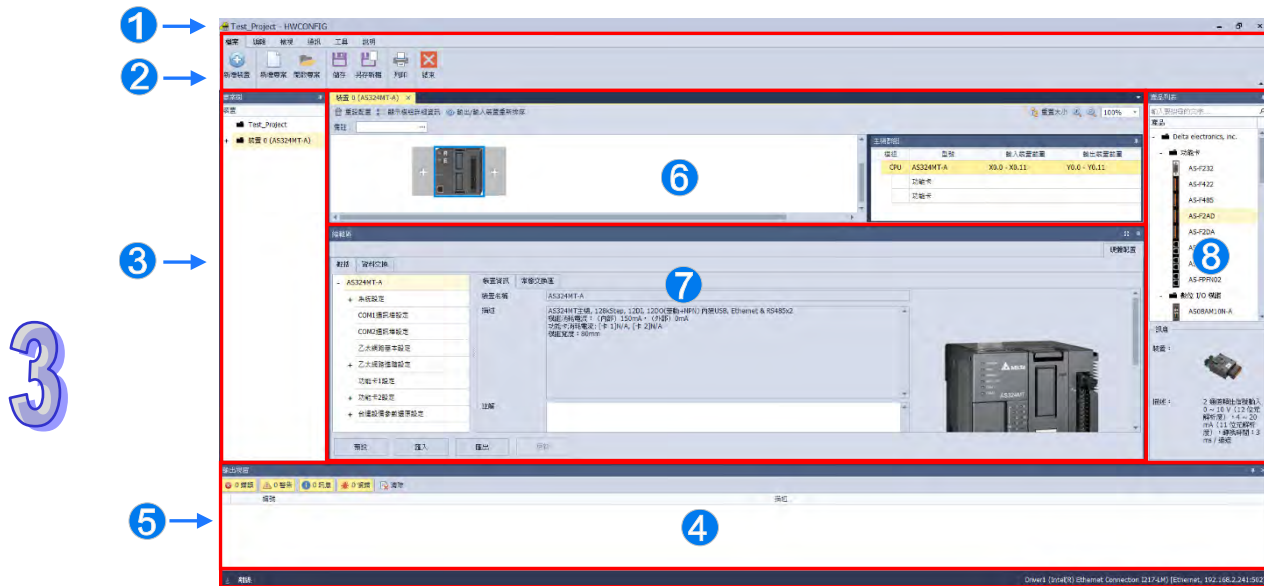
3



而在 ISPSOft 軟體的專案中，雙擊滑鼠左鍵開啟 HWCONFIG，則專案機種與通訊設定，將會自動帶入 HWCONFIG 中。



下圖是一個完整的 HWCONFIG 工作畫面。



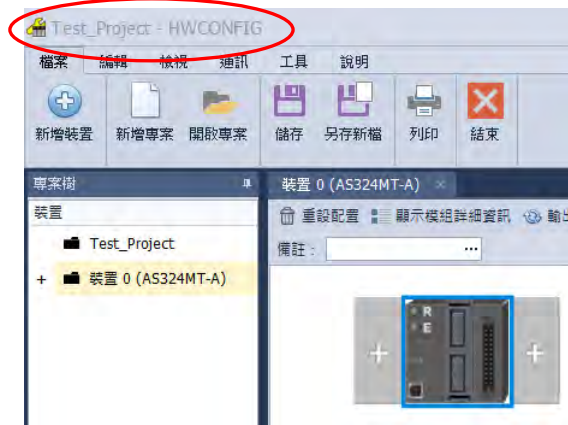
- ❶ 視窗標題：顯示目前所編輯的專案名稱與程式標題。
- ❷ 功能工具列：共有 6 類的功能操作選項。
- ❸ 專案樹：階層樹狀結構的專案物件管理介面。
- ❹ 輸出視窗：顯示專案的相關訊息。
- ❺ 狀態列：顯示目前連線狀態的相關資訊。
- ❻ 硬體配置區：使用者可於此處進行硬體的配置。
- ❼ 編輯區：使用者可於此處進行主機與各模組的參數設定。
- ❽ 產品列表：階層樹狀結構的方式呈現所有可用的硬體資源。

3.1.2 HWCONFIG 的環境介紹

HWCONFIG 的環境有分視窗標題與狀態列及功能工具列的 6 類功能項等，依據所選用的機種，各類功能項的下層項目也會有所差異，在此先稍做瀏覽介紹，詳細的操作方式則將於下方的章節中陸續介紹。

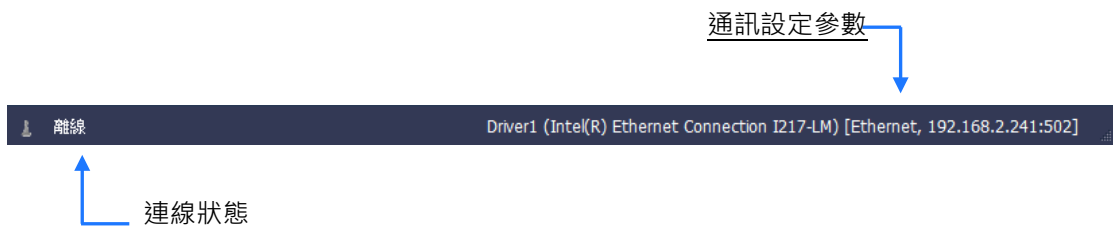
3.1.2.1 視窗標題與狀態列

成功建立專案之後，HWCONFIG 的視窗標題便會顯示該專案的名稱，其格式為 **專案名稱[程式標題]**。



3

視窗下方的狀態列主要在於顯示目前的各種工作狀態，包括連線狀態以及通訊設定參數。



3.1.2.2 功能工具列-檔案

檔案：主要功能為基本的專案存取操作。



操作項	功能說明
新增裝置	加入新的 PLC 主機至專案中
新增專案	開啟新專案
開啟專案	開啟舊專案
儲存	儲存專案

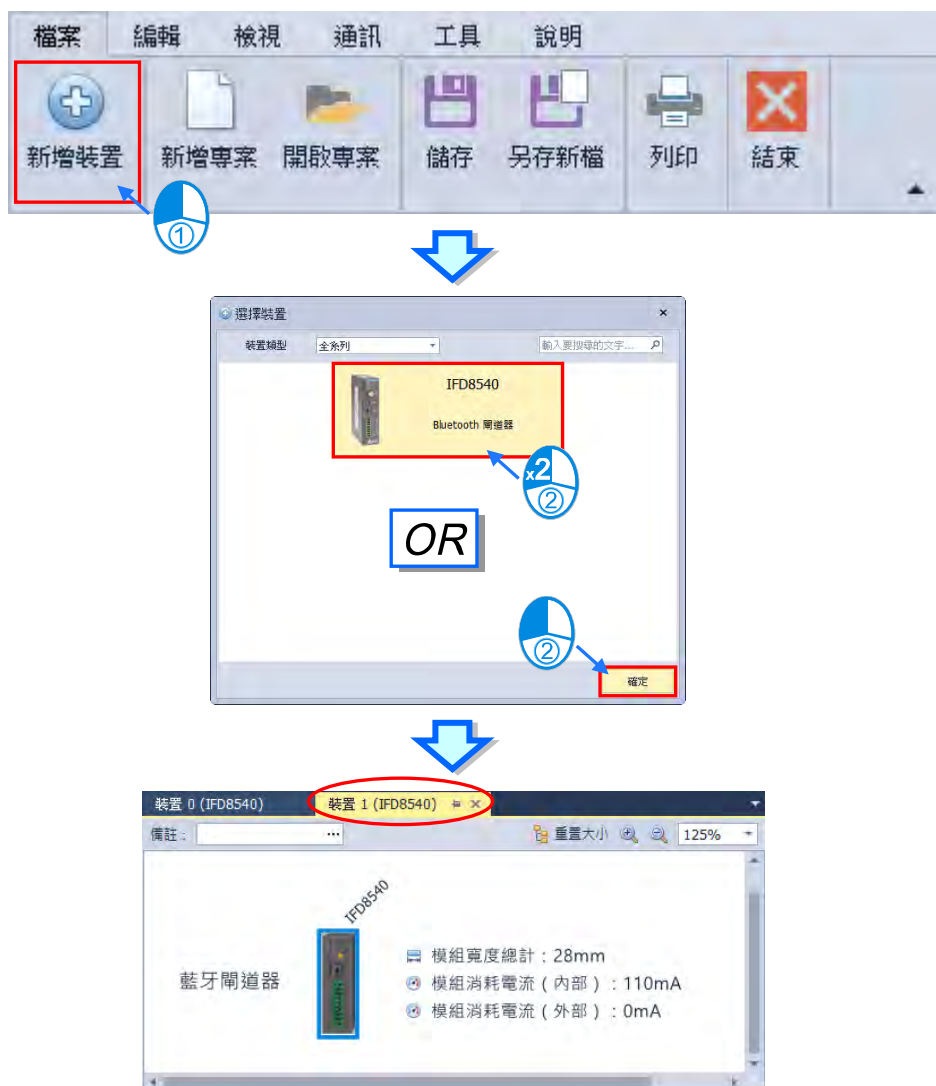
操作項	功能說明
另存新檔	另存專案於新的位置
列印	列印專案硬體配置圖
結束	關閉 HWCONFIG 軟體

● 新增裝置

新增裝置只支援單機版。新增裝置可在專案中增加另一個主機。

於「檔案」內點選「新增裝置」，會跳出選擇裝置視窗，選定好裝置後可以按雙擊滑鼠左鍵或按下「確定」鍵，即可在專案增加該裝置，在硬體配置區會產生新裝置的分頁。

3

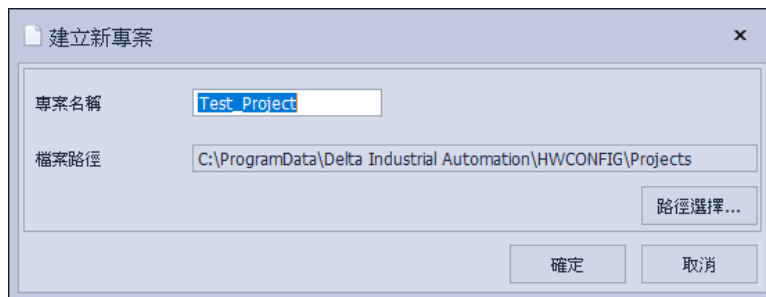


● 新增專案

新增專案只支援單機版。開啟新專案。

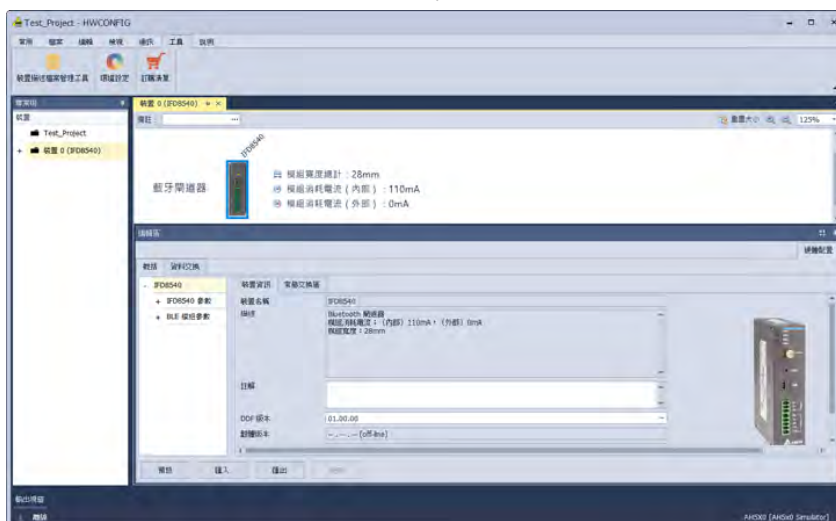
於單機版軟體中點擊「功能工具列」中的「檔案」的「新增專案」，或使用快捷鍵 **Ctrl+N**，點擊後會先詢問是否要儲存現在的專案「是 (Y)」為儲存，「否 (N)」為不儲存，「取消」為取消開啟新專案。

待建立新專案的視窗出現，於設定視窗中依序輸入「專案名稱」及「檔案路徑」，完成後按下「確定」。



3

即會出現選擇裝置視窗，在「裝置類型」的下拉選單中選擇機種型式後，在所選取的裝置雙擊或按下 Enter 鍵，也可以在選定裝置後按下「確定」，即可進入操作畫面，如下圖。

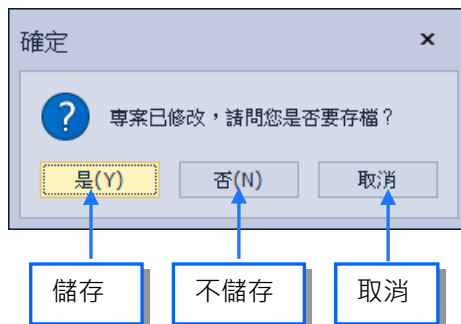
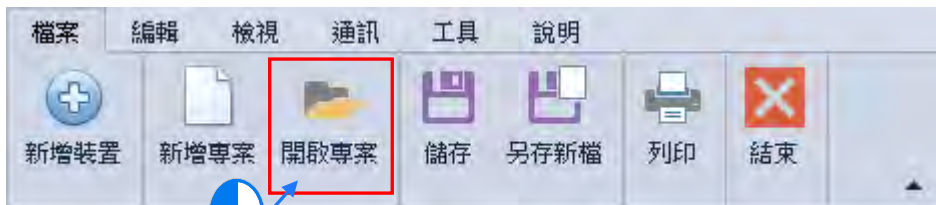


● 開啟專案

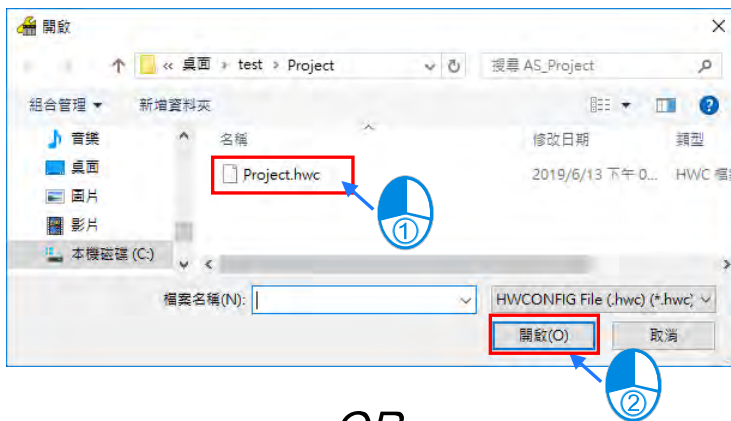
開啟專案只支援單機版，開啟舊專案。

於單機版軟體中點擊「功能工具列」中的「檔案」的「開啟」，或使用快捷鍵 **Ctrl+O**，點擊後會先詢問是否要儲存現在的專案「是 (Y)」為儲存，「否 (N)」為不儲存，「取消」為取消開啟舊專案。

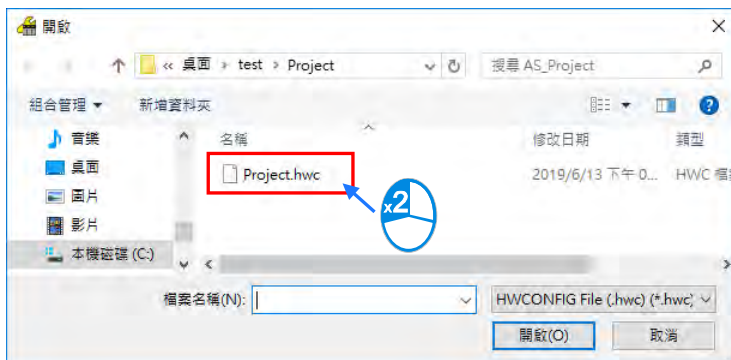
3



找到檔案路徑選擇.hwc 檔後，按下確「開啟」或是雙擊點選該檔案，即可開啟舊檔。

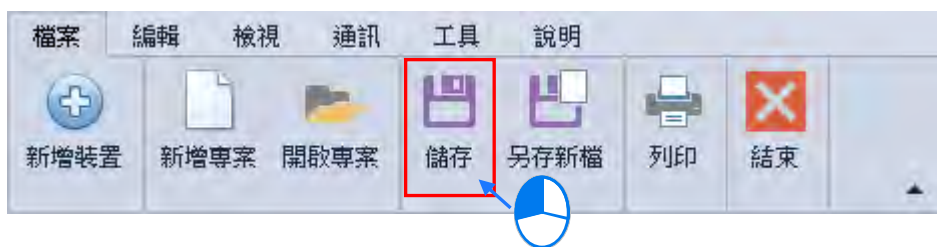


OR



- 儲存

點擊「功能工具列」中「檔案」的「儲存」，或使用快捷鍵 **Ctrl+S**。



儲存成功後，下方輸出視窗會顯示專案存檔成功以及存檔目錄。



- 另存新檔

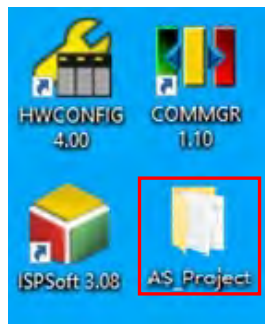
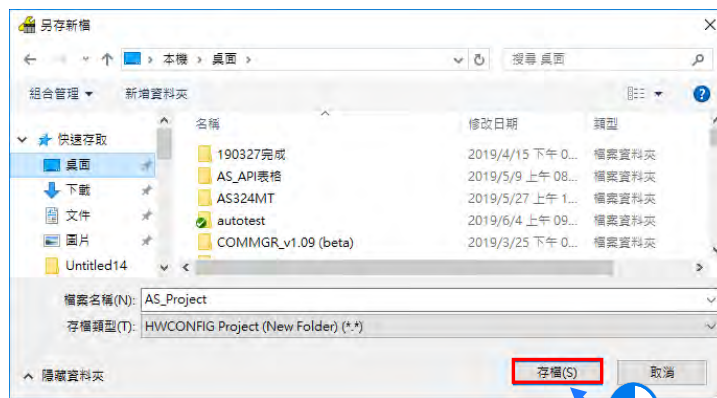
另存新檔只支援單機版。將專案儲存於新的位置。

點擊「功能工具列」中的「檔案」的「另存新檔」，或使用快捷鍵 **Ctrl+Alt+S**。



3

等待另存新檔畫面跳出後，選擇要存檔的位置，於「檔案名稱 (N)」內輸入檔名後，按下「存檔 (S)」，即會將檔案儲存在新的資料夾。

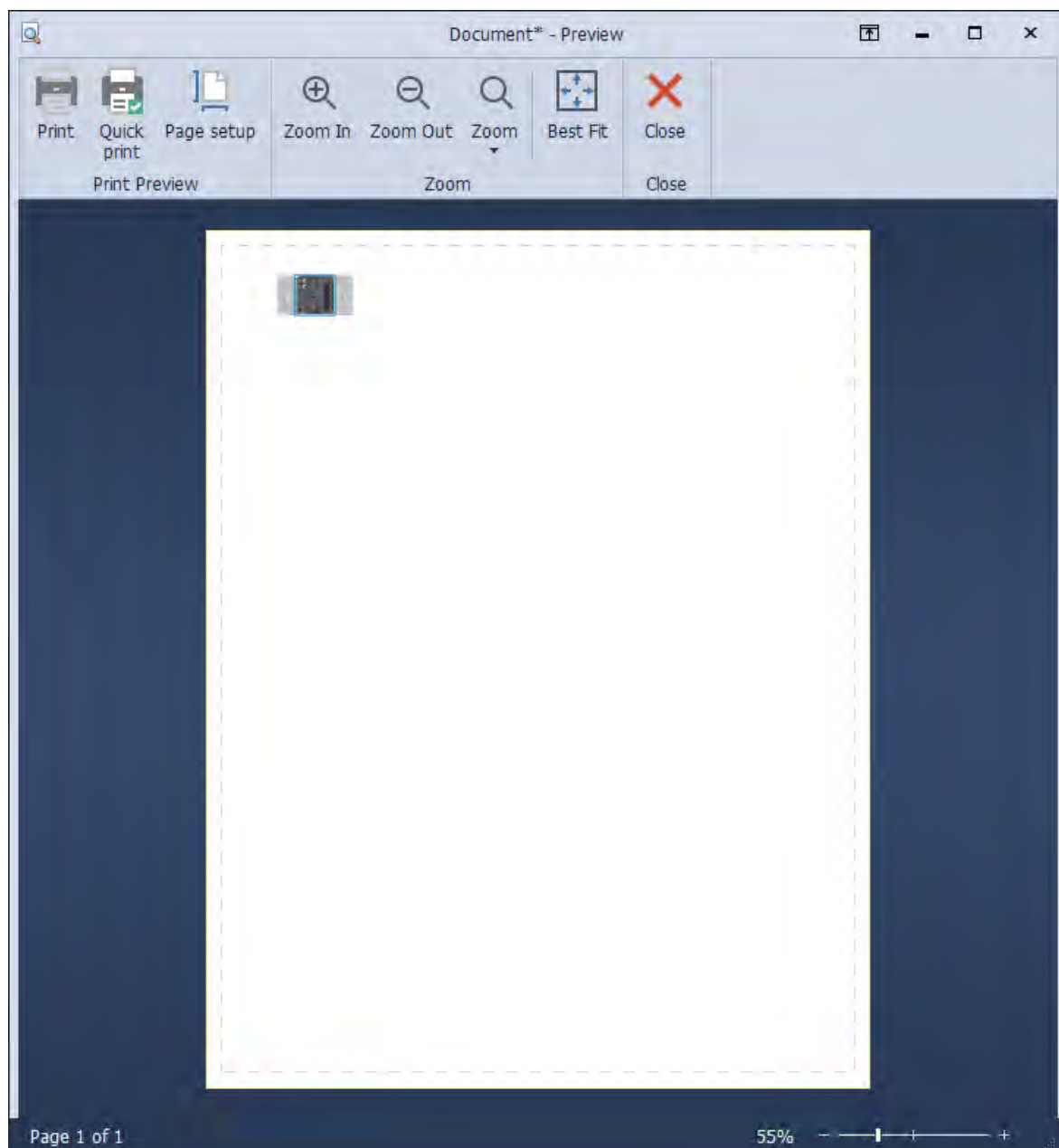


● 列印

列印專案硬體配置圖。

點擊「功能工具列」中的「檔案」的「列印」，或使用快捷鍵 **Ctrl+P**，即可進入列印設定視窗。





3

➤ **Print Preview (列印設定)**

Print : 設定列印詳細資訊

Quick print : 快速列印

Page setup : 列印版面配置

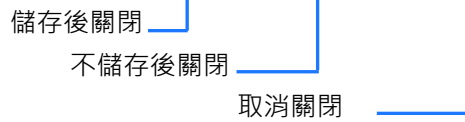
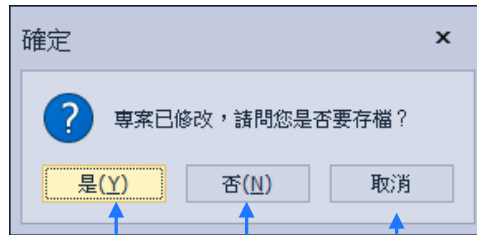
➤ **Zoom (檢視顯示大小)**

Zoom In : 放大顯示

Zoom Out : 縮小顯示

Zoom : 調整顯示比例

Best Fit : 根據視窗頁面調整顯示大小



3

3.1.2.3 功能工具列-編輯

編輯：主要提供編輯工作時所需的操作功能。



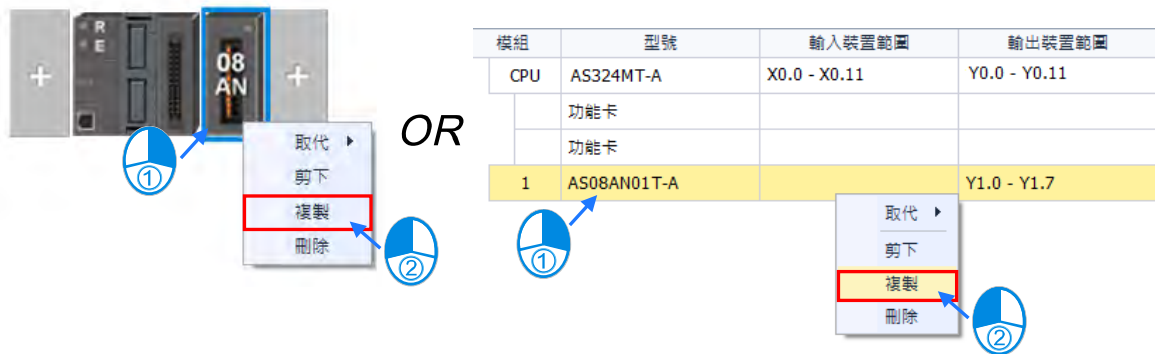
操作項	功能說明
貼上	貼上已複製或已剪下的模組與功能塊
剪下	剪下所選取的模組與功能塊
複製	複製所選取的模組與功能塊
刪除	刪除所選取的模組與功能塊

● 複製

複製模組的方法有下列兩種：

➤ 方法一

於硬體配置區的模組配置圖或模組配置訊息上選取要複製的模組後按下滑鼠右鍵，之後於快捷選單中點選「複製」。



3

➤ 方法二

(1) 先於硬體配置區的模組配置圖或模組配置訊息上點選要複製的模組。



(2) 於 HWCONFIG 的功能工具列點選「編輯」的「複製」，或使用快捷鍵 **Ctrl+C**。

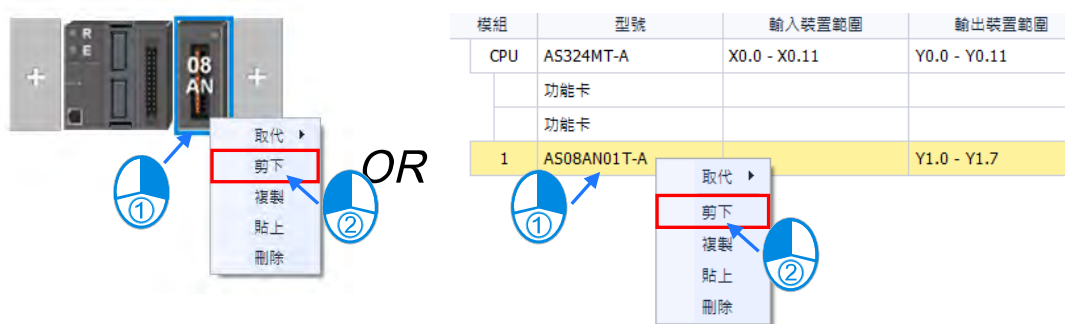


● 剪下

剪下模組的方法有下列兩種：

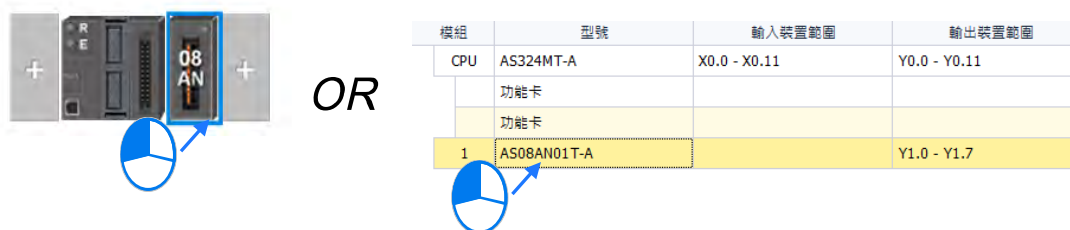
➤ 方法一

於硬體配置區的模組配置圖或模組配置訊息上點選要剪下的模組後按下滑鼠右鍵，之後於快捷選單中點選「剪下」。

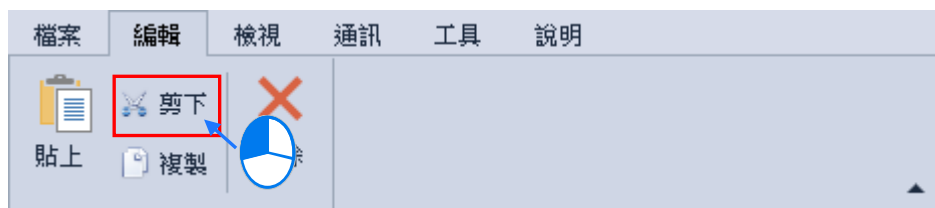


➤ 方法二

(1) 先於硬體配置區的模組配置圖或模組配置訊息上點選要剪下的模組。



(2) 於 HWCONFIG 的功能工具列點選「編輯」的「剪下」，或使用快捷鍵 **Ctrl+X**。



而被剪下的模組會呈現反灰的狀態，若要解除反灰狀態，則必須再該模組使用複製、貼上或刪除，選取其他模組使用剪下、複製或貼上也可以解除該模組狀態

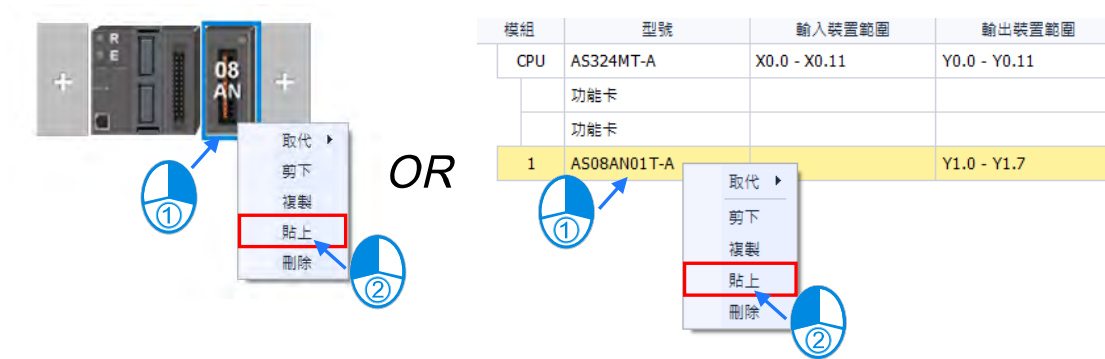


● 貼上

貼上模組的方法有下列兩種：

➤ 方法一

於硬體配置區的模組配置圖或模組配置訊息上點選要貼上的插槽或位置後按下滑鼠右鍵，再於快捷選單中點選「貼上」。



➤ 方法二

(1) 先於硬體配置區的模組配置圖或模組配置訊息上點選要貼上的插槽。

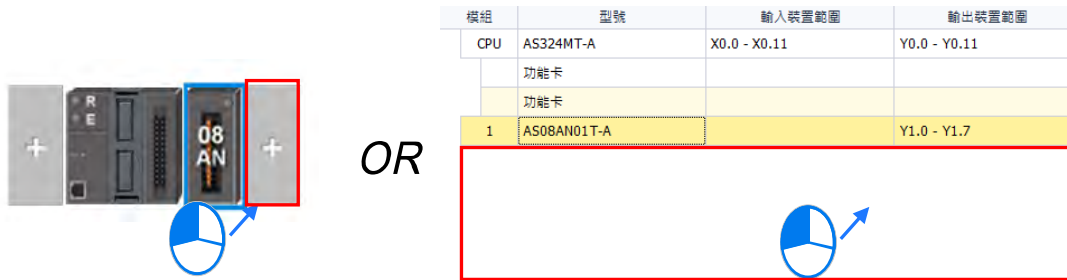


(2) 於 HWCONFIG 的功能工具列點選「編輯」的「貼上」或使用快捷鍵 **Ctrl+V**。



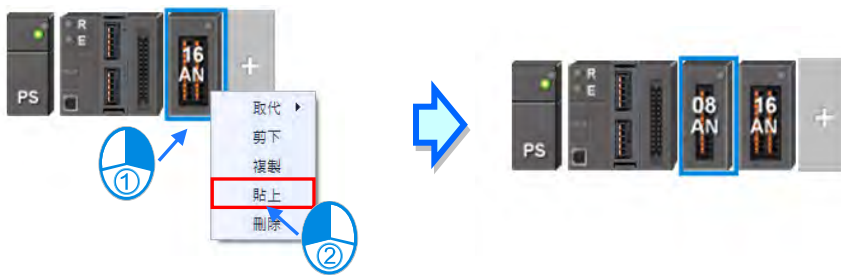
3

AS 系列機種為無背板設計，若不選定位置貼上模組只能貼在最右方的位置，與模組配置區的最下方的空白列。



3

若選取的位置已經有模組，則會貼上在選取位置的左側，其餘模組會往右側移動，如下圖，在 16AN 模組上按下快捷選單的「貼上」，將 08AN 模組貼上至選取的位置並將 16AN 模組往右側移動。

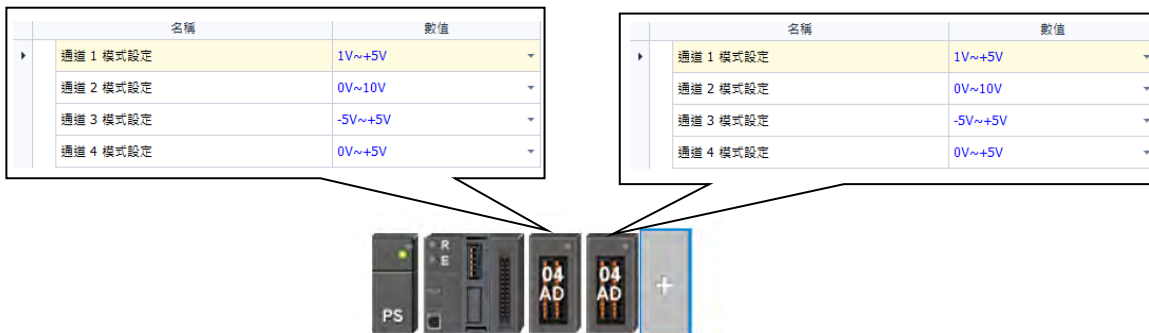


補充說明

當模組進行貼上的操作時，模組參數的處理規則如下。

➢ 輸入/輸出裝置範圍：模組的裝置範圍皆會由右側第一個模組開始排序，並不會重複，於被貼上的模組，會自動變更對應裝置至可用的位址。

模組參數設定也可以被複製、貼上如下圖範例，被貼上的模組參數會與被複製的模組相同。



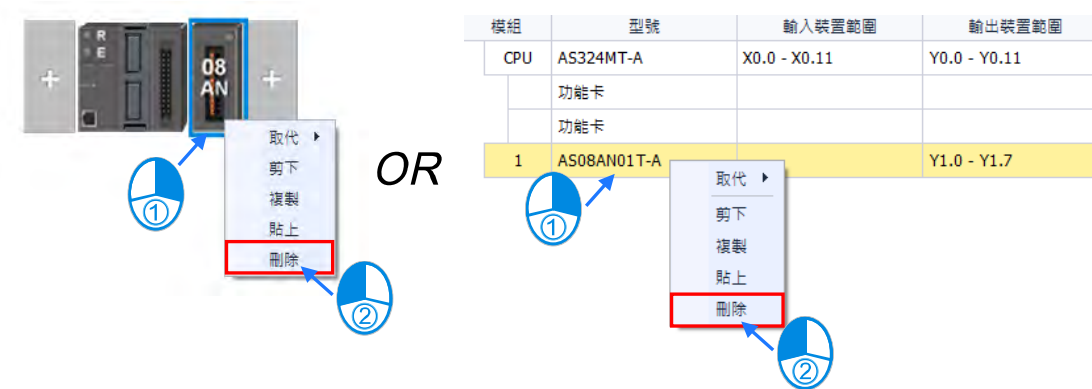
模組	型號	輸入裝置範圍	輸出裝置範圍
電源	AS-PS02A		
CPU	AS324MT-A	X0.0 - X0.11	Y0.0 - Y0.11
	AS-F422		
	功能卡		
1	AS04AD-A	D28000 - D28019	
2	AS04AD-A	D28020 - D28039	

● 刪除

刪除模組的方法有下列兩種：

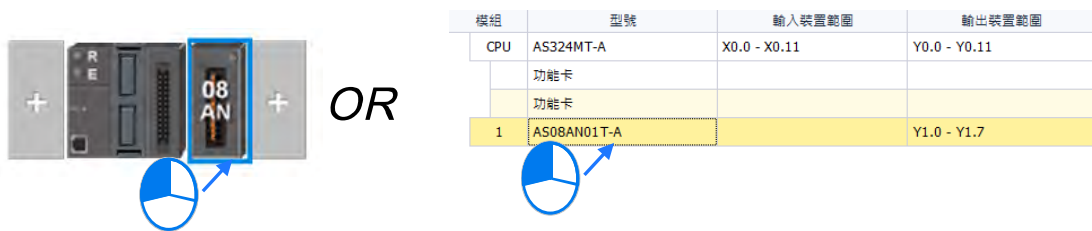
➤ 方法一

於硬體配置區的模組配置圖或模組配置訊息上點選要刪除的模組後按下滑鼠右鍵，之後於快捷選單中點選「刪除」。

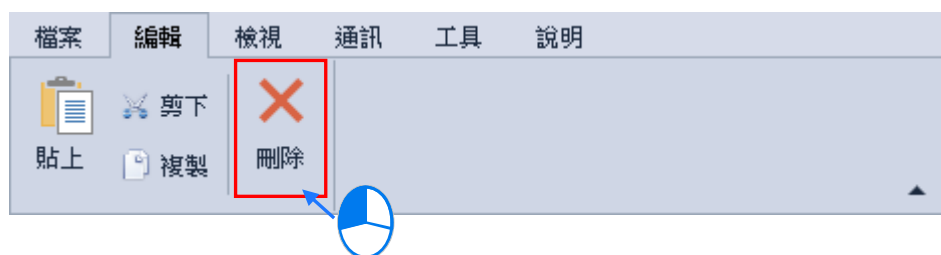


➤ 方法二

(1) 先於硬體配置區的模組配置圖或模組配置訊息上點選要刪除的模組。

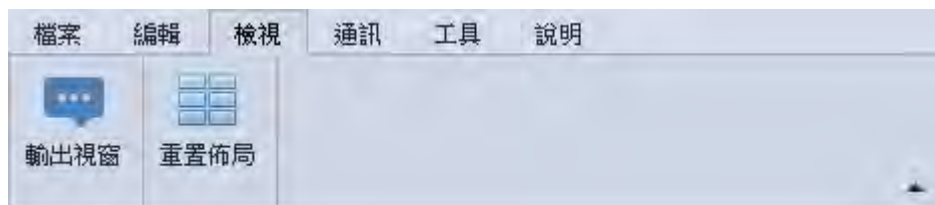


(2) 於 HWCONFIG 的功能工具列點選「編輯」的「刪除」，或使用 **Delete** 鍵。



3.1.2.4 功能工具列-檢視

檢視：主要功能為檢視專案中的資訊以及設定工作環境的顯示方式。



操作項	功能說明
輸出視窗	開啟或關閉輸出視窗
重置佈局	重置 HWCONFIG 版面配置

3.1.2.5 功能工具列-通訊

通訊：主要為透過 HWCONFIG 對 PLC 主機進行連線操作或設定的功能。



操作項	功能說明
上載	將 PLC 中的參數與模組設定傳輸至 HWCONFIG。
下載	將 HWCONFIG 專案中的參數與模組設定傳輸至 PLC。
掃描	掃描 PLC 目前的模組配置及設定。
線上模式	即時監看模組內的資訊。
通訊設定	設定通訊連接的方式。

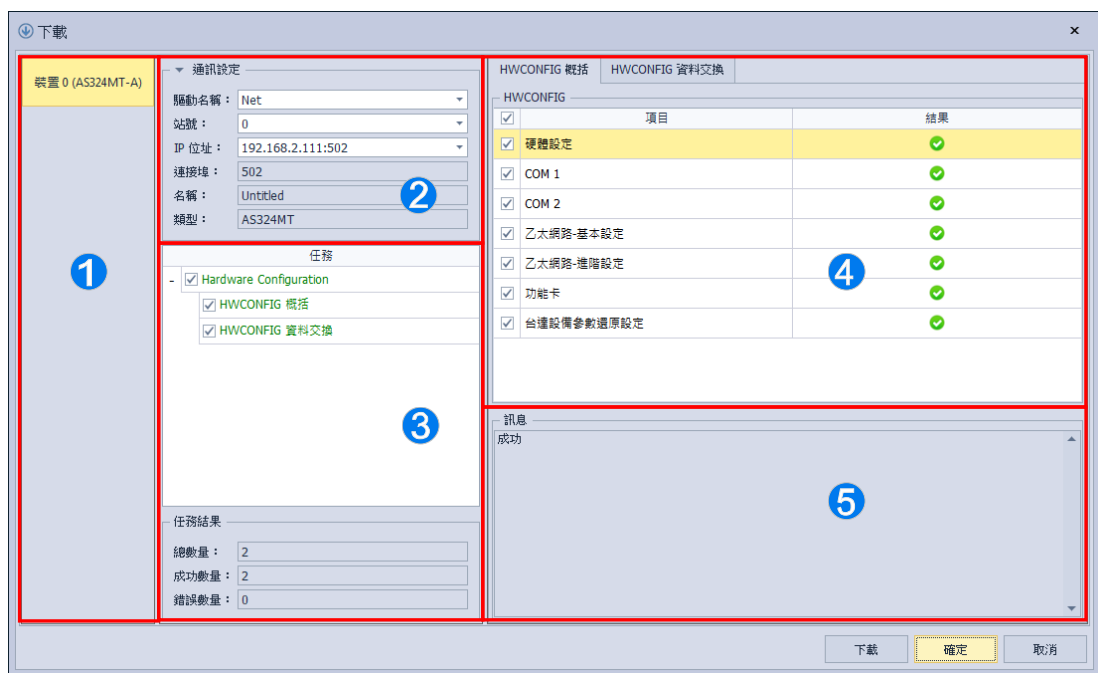
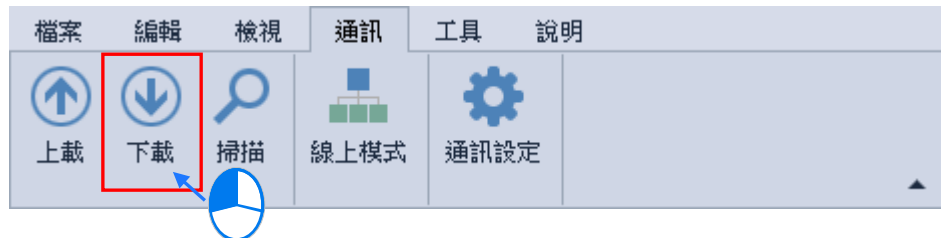
●上載/下載

於 HWCONFIG 當中所做的設定，必須被下載至主機中，才可正式生效。

- (1) 單機版 HWCONFIG 在進行參數上下載之前，請先確認已將通訊設定完成。
- (2) 若由 ISPSOft 開啟 HWCONFIG，HWCONFIG 會直接引用 ISPSOft 的通訊設定，因此進行參數上下載之前，請務必確認 ISPSOft 與主機正常連線。執行時，若主機目前的標籤名稱與參數設定的名稱不符時，畫面亦會出現提示訊息。

➤ 下載系統硬體參數

HWCONFIG 進行下載時，先於功能工具列中點選「通訊」的「下載」或使用快捷鍵 **Ctrl+F8**，之後便會出現下載視窗。



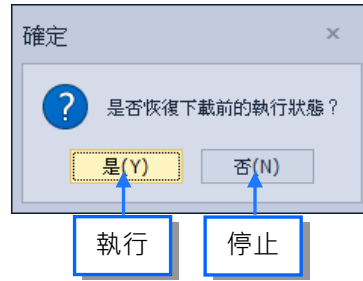
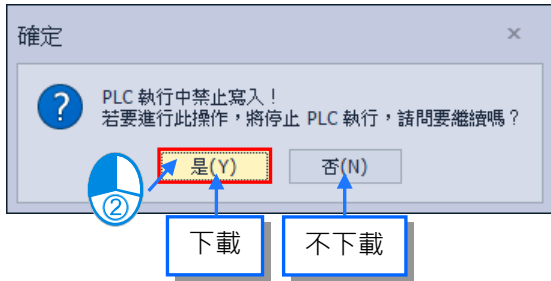
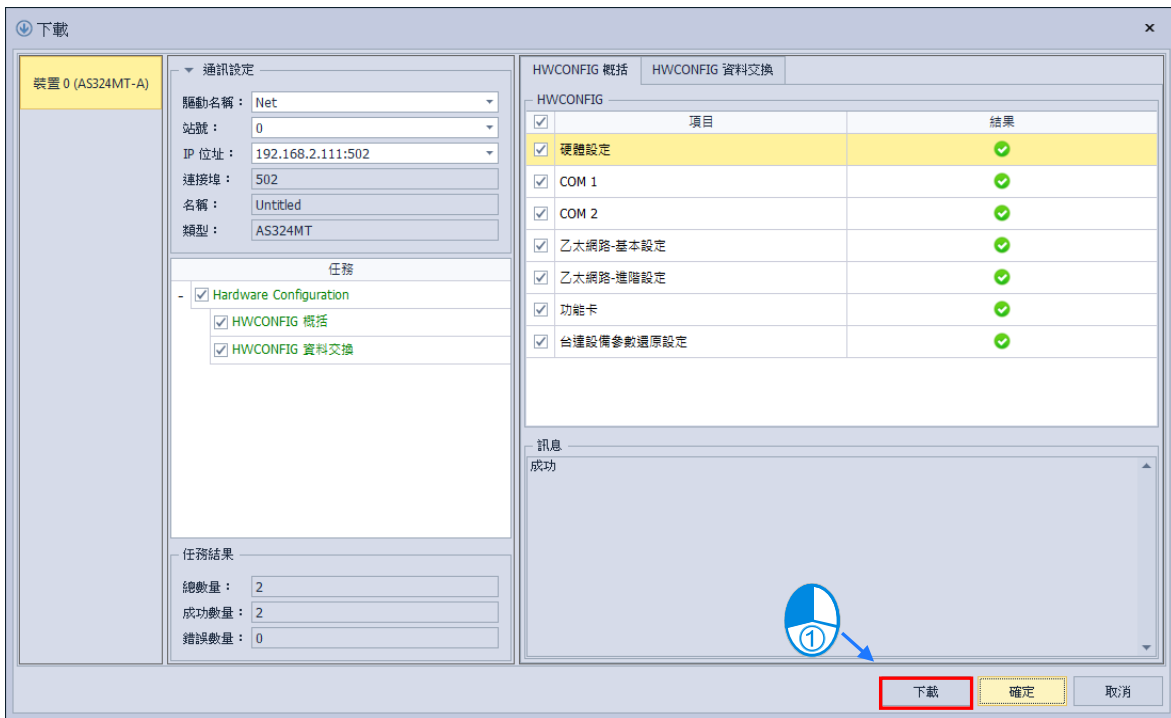
- ❶ 裝置：顯示目前所編輯的裝置名稱與裝置型號。
- ❷ 通訊設定：可以進行通訊設定，以及顯示目前所通訊的機型與 PLC 名稱。
- ❸ 下載任務：為選取項目；為部分選取項目；為不選取項目。
- ❹ 下載任務之細節：可勾選裝置系統設定以及資料交換的細節項目，右方會顯示下載結果。
- ❺ 訊息：顯示下載結果的訊息。

*1. 由 ISPSOFT 軟體所開啟的 HWCONFIG，會直接引用 ISPSOFT 的通訊設定，故通訊設定無法做更變更，如果要更改通訊需先存檔並關閉 HWCONFIG，再於 ISPSOFT 更改通訊設定後，再次打開 HWCONFIG 即可。

下載視窗開啟後，首先進行通訊設定，接著可視需求勾選要下載的項目，按下「下載」按鈕後待處理視窗結束後，便可將參數下載至主機中。

下圖為下載視窗，下載時的主機狀態必須為 STOP；若主機處於 RUN 的狀態時，系統便會在開始下載前先顯示提示訊息，按下確定後，接著會詢問是否恢復 PLC 的運行狀態。

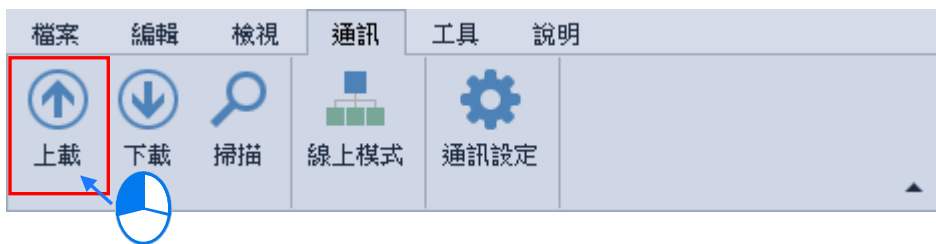
3

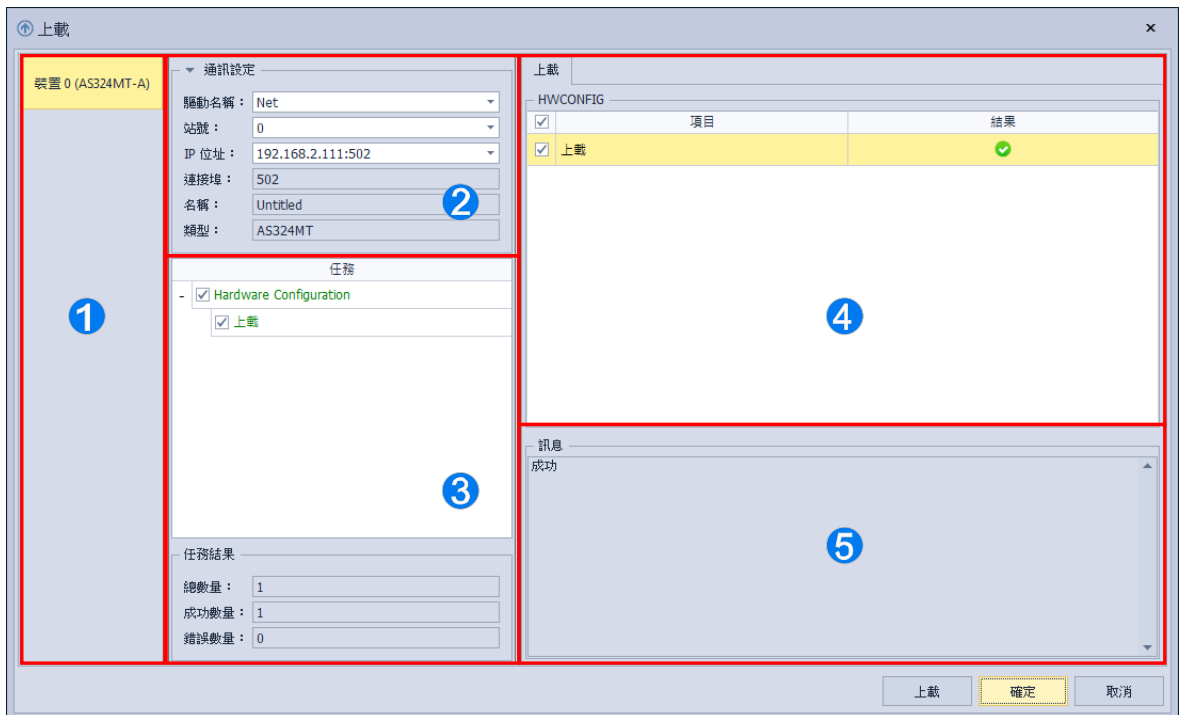


- *1. 當主機目前的標籤名稱與參數設定的名稱不符時，在開始執行上傳或下載之前會先顯示提示訊息。
- *2. HWCONFIG 會引用 ISPSOft 的通訊設定，因此進行上下載前請確認 ISPSOft 已可與主機正常連線。

➤ 上傳系統硬體參數

HWCONFIG 進行下載時，先於功能工具列中點選「通訊」的「上傳」或使用快捷鍵 **Ctrl+F9**，後便會出現上傳視窗。





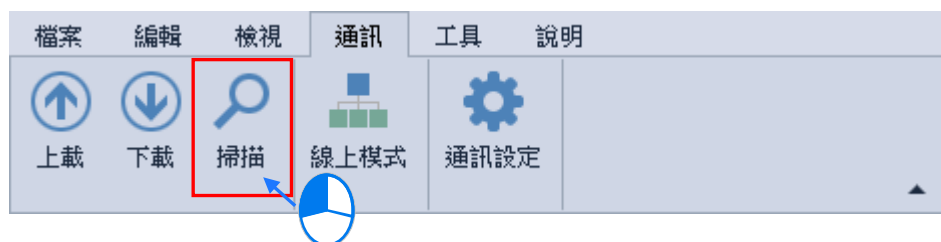
- ❶ 裝置：顯示目前所編輯的裝置名稱與裝置型號。
- ❷ 通訊設定：可以進行通訊設定，以及顯示目前所通訊的機型與 PLC 名稱。
- ❸ 上載任務：為選取項目；為不選取項目。
- ❹ 上載任務之細節：可勾選要上載的細節項目，右方會顯示上載結果。
- ❺ 訊息：顯示上載結果訊息。

*1. 由 ISPSOft 軟體所開啟的 HWCONFIG，會直接引用 ISPSOft 的通訊設定，故通訊設定無法做更變更，如果要更改通訊需先存檔關閉 HWCONFIG，於 ISPSOft 更改通訊設定後，再次打開 HWCONFIG 即可。

上載視窗開啟後，首先進行通訊設定，接著按下「上載」按鈕後待處理視窗結束後，便可將參數上載至主機中，其中上載項目包括模組配置、模組內部參數及 PLC 主機參數。

執行上傳操作時，所有的主機參數將會全部被上傳回來；而執行下載操作時，畫面則會先出現勾選視窗，使用者可於視窗中勾選想要下載的參數主類別，而無須將所有的設定全部下載至主機中。

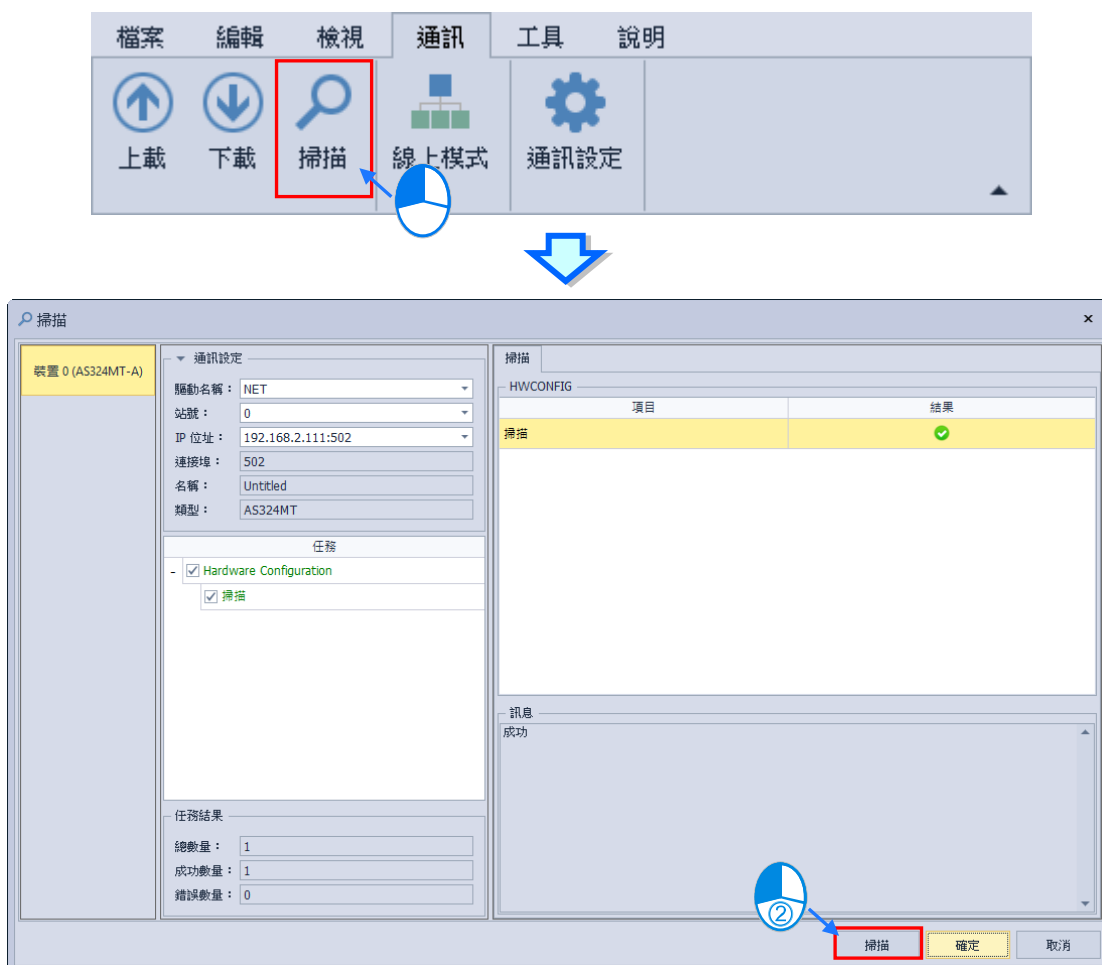
● 掃描



當使用者已經完成硬體設備配置並確認可以與主機正常通訊時，再利用 HWCONFIG 的「掃描」功能，將實際的模組配置狀態掃描上來，如此便可省去逐一增加模組的時間；而進行操作之前，請確認 ISPSOft 已經可以與主機正常連線。

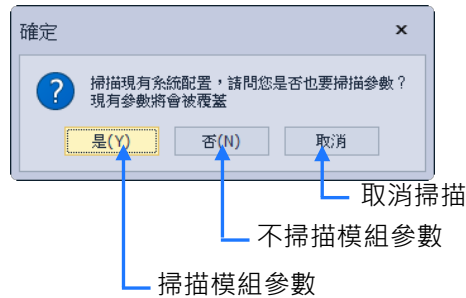
操作時請於 HWCONFIG 的功能工具列中點選「通訊」的「掃描」或使用快捷鍵 **Ctrl+F7**；而偵測的內容則包括模組型號、模組配置、模組韌體版本，但不包含主機參數；且在執行掃描後，原本的模組配置與模組參數設定都將會被覆蓋，而主機本身的參數設定，則不會受到影響。

3



部分機種可支援模組參數掃描，點選掃描後即會詢問是否要掃描參數（沒詢問為不支援此功能），若選擇「是」，則會將偵測主機右側模組（不包含主機內部參數*1），並偵測包括模組型號、模組配置、背板配置、模組韌體版本，且覆蓋目前軟體的配置，當偵測完成後，會出現完成視窗，點擊確定後，即完成偵測動作，此時實際的模組配置狀態將會顯示於硬體配置區，且會自動重新配置所有模組的輸入/輸出裝置範圍；若選擇「否」，則模組的參數會為系統預設值。

*1.主機內部參數包括功能卡參數。



掃描結束後訊息會出現成功，最後點擊「確定」才會更新專案內的模組，點擊「取消」則不會更改專案內的模組。

	上載	掃描
主機設定	V	
功能卡	O	V
功能卡設定	O	
模組配置	O	V
模組設定	O	掃描模組參數
模組韌體版本	O	V

*O：若主機內部模組配置與實際模組配置不同時，上載資料也會與實際模組配置不符。

接著我們將以一個實例來說明「掃描」與「上載」之間的差異。其中「掃描」會偵測目前模組與功能卡的配置狀況；而「上載」則會將存放於主機內部的主機參數與模組配置重新讀取回來，但讀取回來的硬體配置則可能已與實際連接的配置狀況不符。

範例：

首先將下圖的模組配置下載至主機中。



接下來先移除 16AN 模組接著再執行「掃描」的功能，而偵測後的結果則會與實際的模組配置相符。



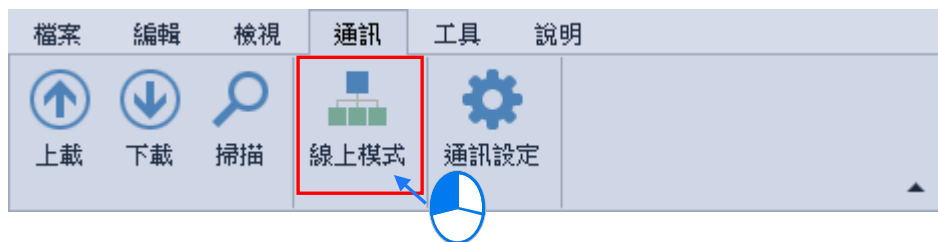
接著我們再執行「上載」的功能，而完成之後，我們會發現上載回來的模組配置與之前下載的相同。



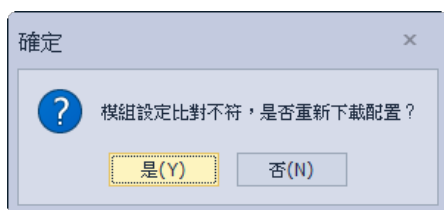
3

● 線上模式

- (1) 於功能工具列中點選「通訊」的「線上模式」或使用快捷鍵 **Ctrl+F4**，之後便可進入線上模式；而要離開線上模式時，只要再次點選「通訊」的「線上模式」或使用快捷鍵 **Ctrl+F4** 即可結束線上模式。



- (2) 在進入線上模式之前，軟體會先檢查存放於主機內部的硬體配置參數，並確認與模組配置區的模組設定是否一致，若兩者不一致時，系統便會要求重新下載配置參數。



(3) 進入線上模式後，畫面中的圖示便會根據，實際的模組狀態而有所不同的顯示；主機圖示的燈號由上至下分別是 Run 與 Error。

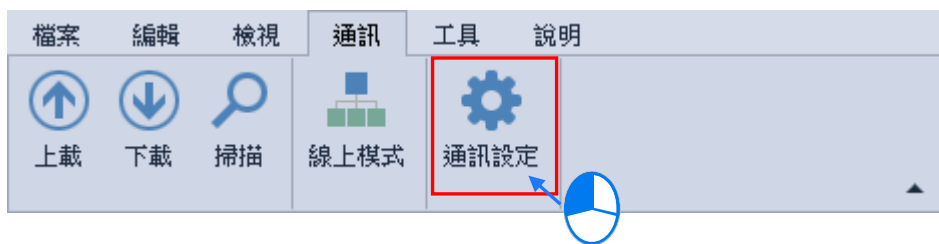


3

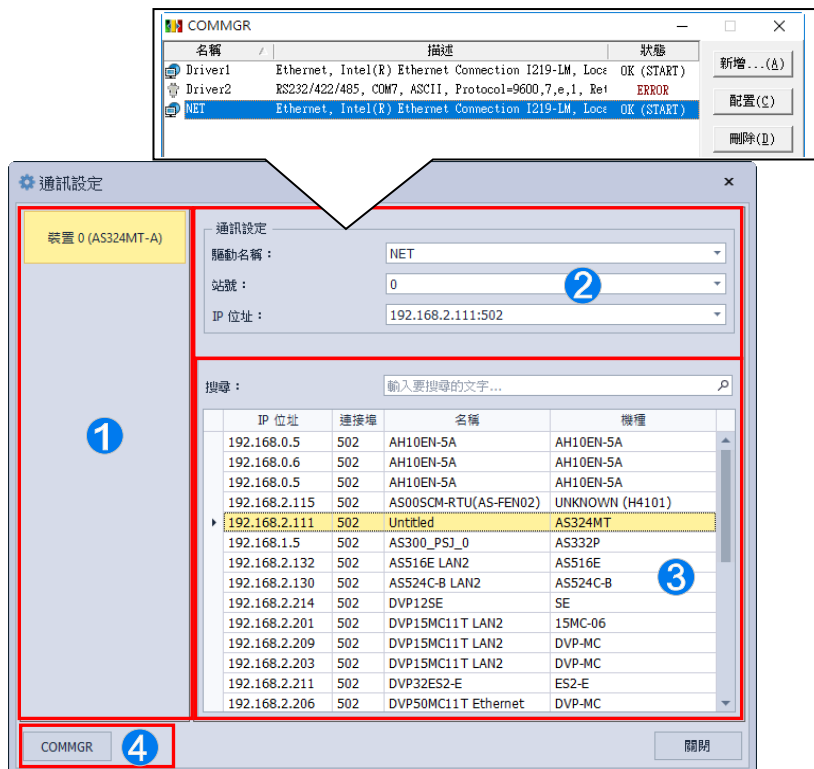
- 通訊設定

通訊設定只支援單機版。可與要連接的裝置連線。

於功能工具列中點選「通訊」的「通訊設定」，可開啟通訊設定頁面



3



- ❶ 裝置：顯示目前所編輯的裝置名稱與裝置型號。
- ❷ 通訊設定：可以選取 COMMGR 設定好的驅動。
- ❸ 選擇連線裝置：使用 Ethernet 時，可以選取要連接的主機。
- ❹ COMMGR：可開啟 COMMGR 進行通訊設定。

通訊視窗開啟後可先開啟 COMMGR 進行設定連線，接著在 ❷ 的通訊設定中選擇在 COMMGR 所設定的驅動，使用 Ethernet 時在 ❸ 會顯示該驅動包含的主機，最後設定結束後按下「完成」即可。

3.1.2.6 功能工具列-工具

工具：主要為 HWCONFIG 裝置描述檔以及環境設定的功能。



操作項	功能說明
裝置描述檔案管理工具	匯入 IODD、DDF 等裝置描述檔案。
環境設定	變更主題風格與顯示語言。
訂購清單	列出專案中配置所需的硬體及配件清單。

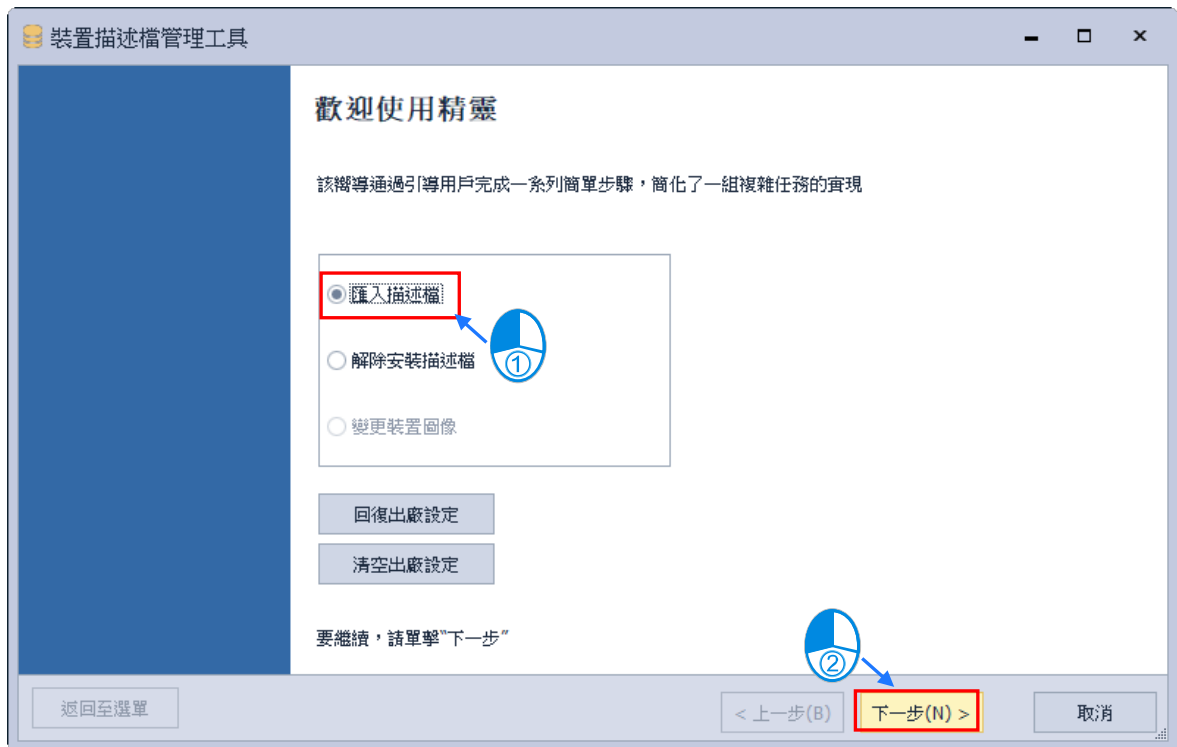
- 裝置描述檔案管理工具

裝置描述檔案管理工具，可以新增或刪除裝置描述檔。

於功能工具列中點選「工具」的「裝置描述檔案管理工具」，即可開啟裝置描述檔案管理頁面。



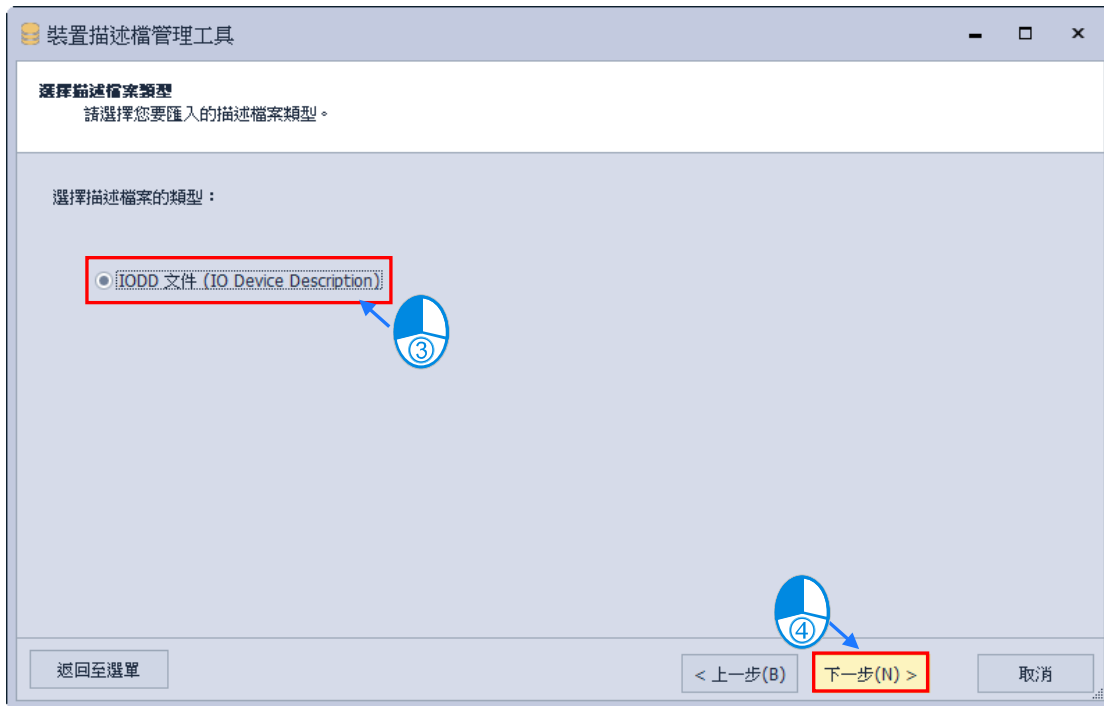
裝置描述檔案管理頁面，可以選擇「匯入描述檔」或「解除安裝描述檔」，選定好後點選「下一步」。



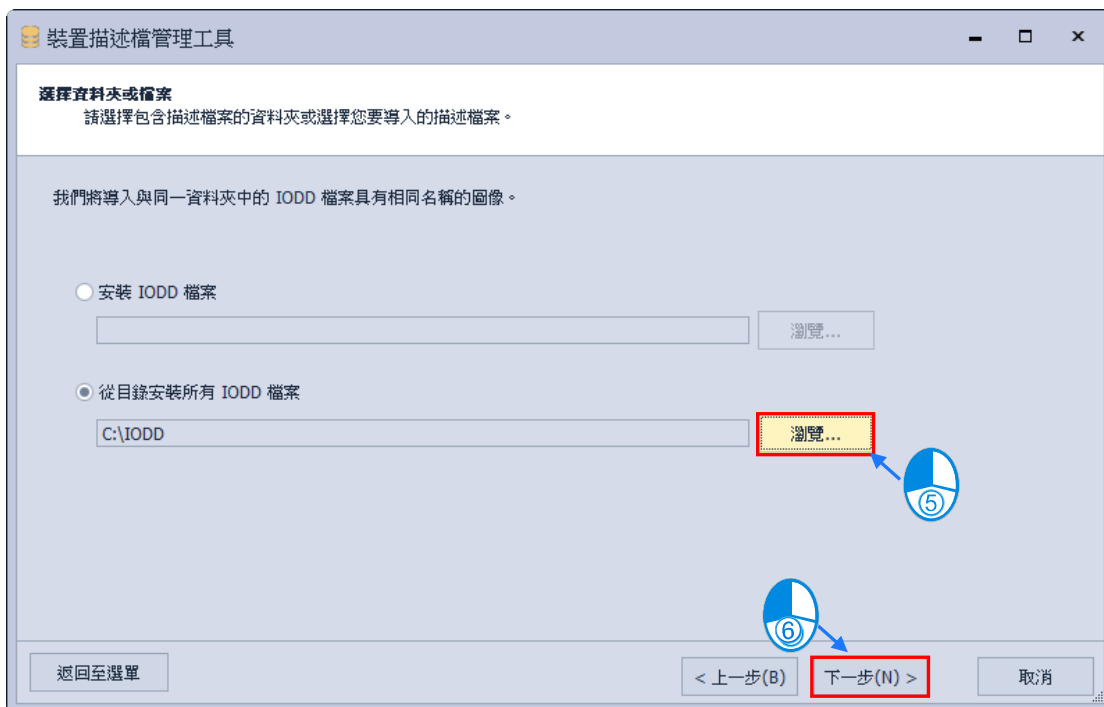
3

接著勾選描述檔的類型，選定好後點選「下一步」。

3



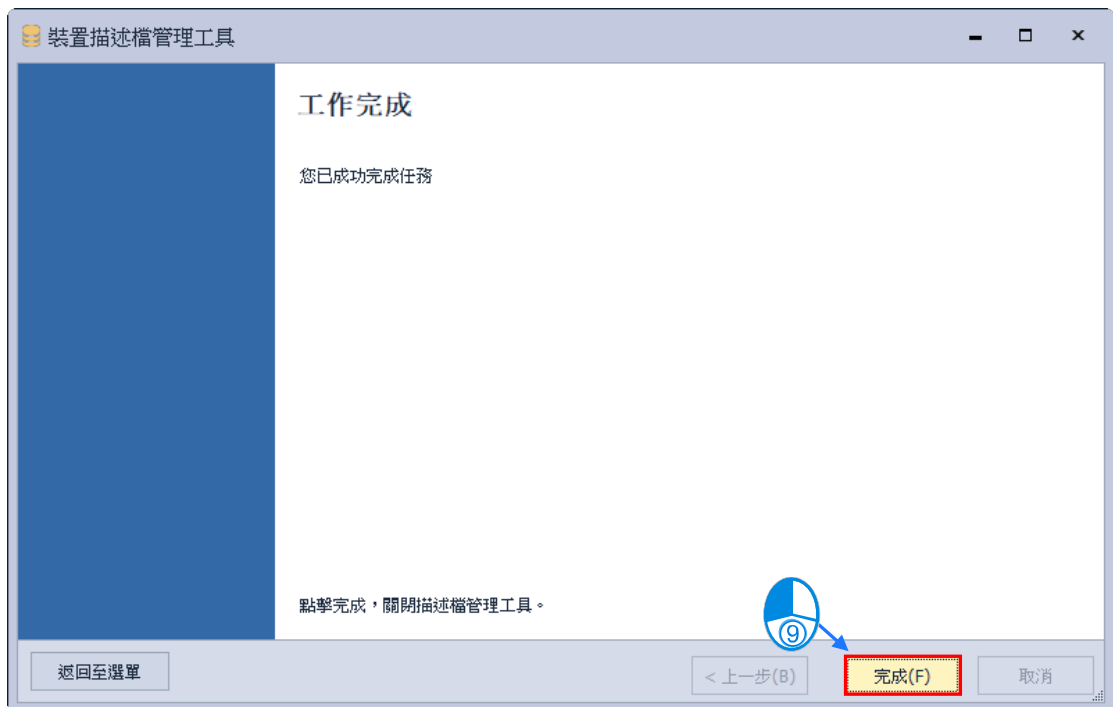
再來選擇描述檔的路徑，使用安裝文件一次只能新增單一文件，選擇從目錄安裝可以一次新增多個文件。



最後勾選所要加入的描述檔，點選「下一步」系統將會開始新增描述檔。



按下「完成」後則完成新增描述檔。



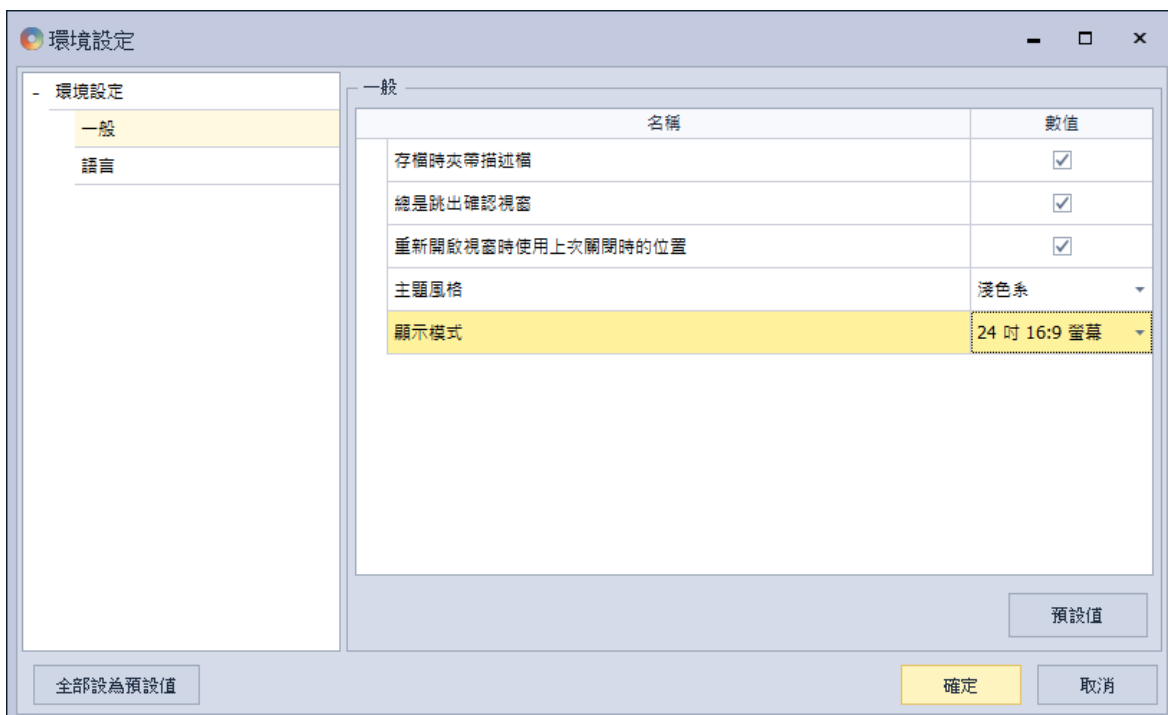
● 環境設定

於功能工具列中點選「工具」的「環境設定」，即可開啟環境設定頁面。



3

環境設定內有一般設定以及語言設定，下方會詳細說明設定內容。



一般頁面內的設定選項。

➤ 存檔時夾帶描述檔

設定在存檔時是否儲存描述檔，勾選為儲存描述檔；不勾選為不儲存描述檔。若有檔案儲存描述檔，在沒有該描述檔的 HWCONFIG 環境中，開啟檔案後描述檔會自動匯入；若沒有檔案儲存描述檔，則需要手動匯入描述檔。

➤ 總是跳出確認視窗

設定在操作變更時是否需跳出確認視窗，勾選為需跳出確認視窗；不勾選為不跳出確認視窗。

➤ 重新開啟視窗時使用上次關閉時的位置及大小

若勾選此選項，在關閉軟體會記錄視窗位置及大小配置；不勾選，則下次開啟軟體的視窗位置及大小為預設。

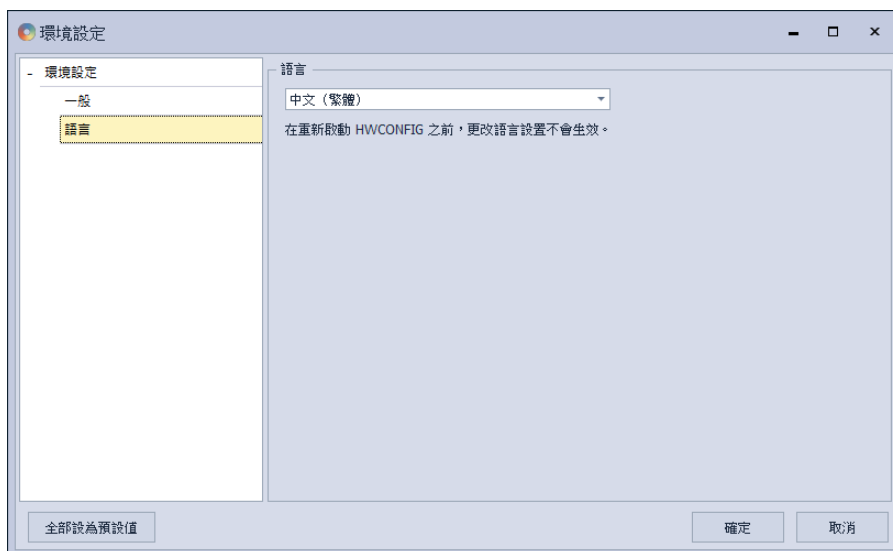
➤ 主題風格

預設風格為淺色系，可更改軟體風格淺色系或深色系兩種。

➤ 顯示模式

顯示模式依照螢幕尺寸有兩種選項可以做選擇，第一種為 13 吋 16 : 9 螢幕，參考 HWCONFIG 3 操作模式，在設定主機及模組參數時會跳出新的視窗提供使用者進行設定，建議筆電或螢幕較小時使用；第二種為 24 吋 16 : 9 螢幕，為 HWCONFIG 4 新操作模式，方便使用者在同一個畫面中顯示更多資訊。

3



語言頁面內的設定選項。

➤ 語言

單機版可進行顯示語言修改，修改完成後需重新啟動 HWCONFIG 才會生效；由 ISPSOFT 開啟的 HWCONFIG 會與 ISPSOFT 顯示語言相同，若需更改語言請先關閉 HWCONFIG 至 ISPSOFT 變更語言後再開啟 HWCONFIG。

● 訂購清單

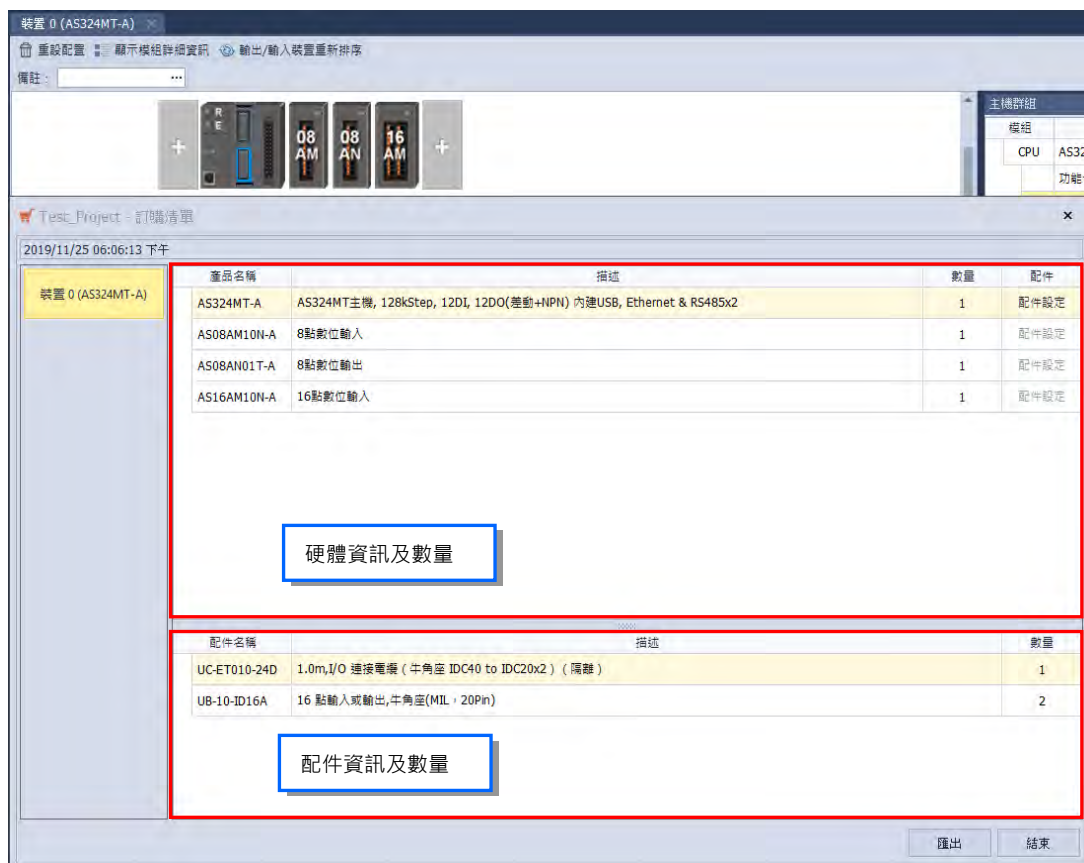
將會列出專案中所配置的硬體及配件清單。

點擊「功能工具列」中的「工具」的「訂購清單」，或使用快捷鍵 **Ctrl+F6**。



3

待訂購清單視窗出現後，上方會顯示目前專案內所配置 PLC 主機、功能卡、以及模組資訊及數量，下方會顯示配件資訊及數量。



配件可以於配件設定中調整配件規格及數量，點擊 PLC 主機右方的配件設定。



產品名稱	描述	數量	配件
AS324MT-A	AS324MT主機, 128kStep, 12DI, 12DO(差動+NPN) 內建USB, Ethernet & RS485x2	1	配件設定
AS08AM10N-A	8點數位輸入	1	配件設定
AS08AN01T-A	8點數位輸出	1	配件設定
AS16AM10N-A	16點數位輸入	1	配件設定

待配件設定視窗出現後，視窗會顯示專案需要的轉接線及轉接板數量，可以更改配件型號及數量，其數量必須等於專案所需的最小數量，按下「預設值」會恢復預設配件及數量，填寫完成後按下「確定」即可更改訂購清單內配件資料。

3

配件名稱	描述	數量
UC-ET010-24D	1.0m,I/O 連接電纜 (牛角座 IDC40 to IDC20x2) (隔離)	1
UB-10-ID16A	16 點輸入或輸出,牛角座(MIL, 20Pin)	2



AS324MT-A

模組數量: 1

繼電器

繼電器 數量:

轉接線總數: 1

UC-ET010-24D (1米, 隔離)

UC-ET020-24D (2米, 隔離)

UC-ET030-24D (3米, 隔離)

轉接板總數: 2

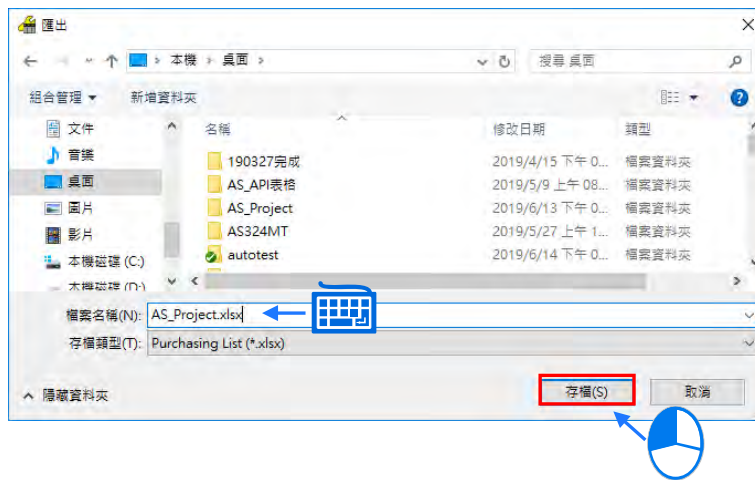
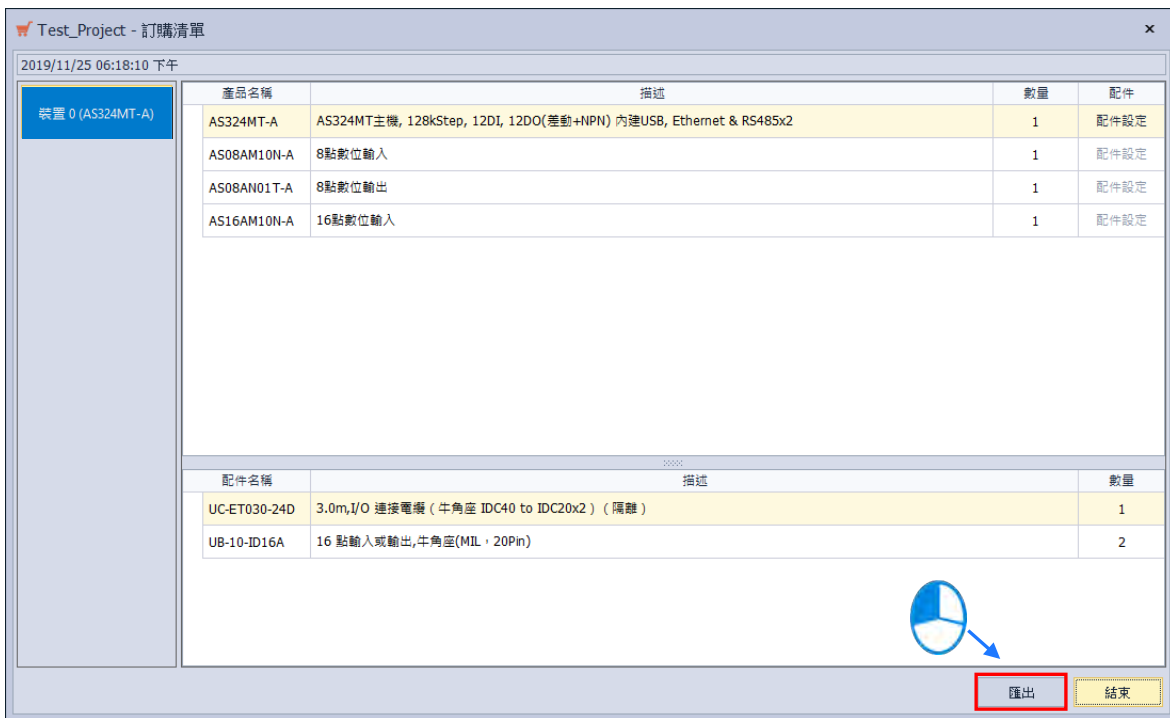
UB-10-ID16A



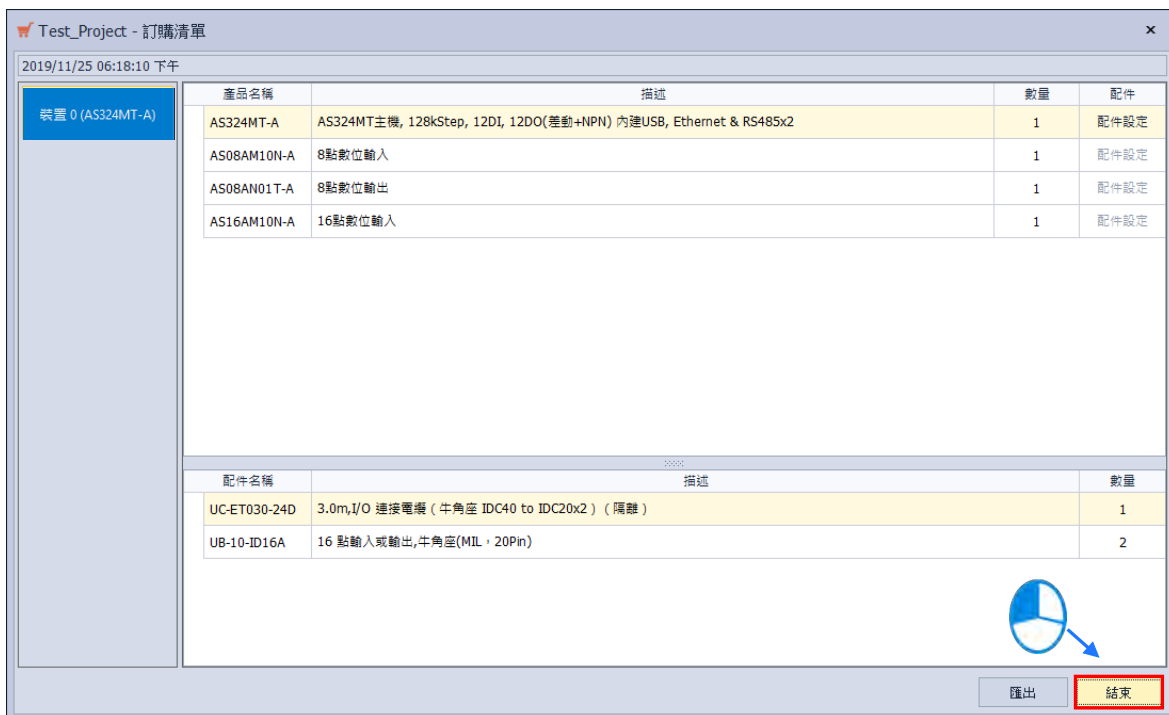
配件名稱	描述	數量
UC-ET030-24D	3.0m,I/O 連接電纜 (牛角座 IDC40 to IDC20x2) (隔離)	1
UB-10-ID16A	16 點輸入或輸出,牛角座(MIL, 20Pin)	2

設定好配件設定後，點擊「匯出」即可將訂購清單以 xlsx 檔的格式被匯出，而之後便可透過 EXCEL 對匯出的資料進行編輯或管理。

3

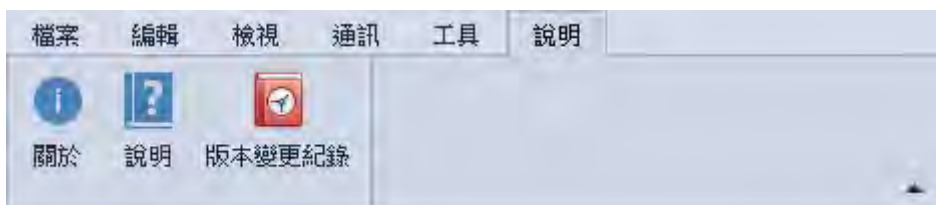


最後可點選「結束」來關閉訂購清單視窗。



3.1.2.7 功能工具列-說明

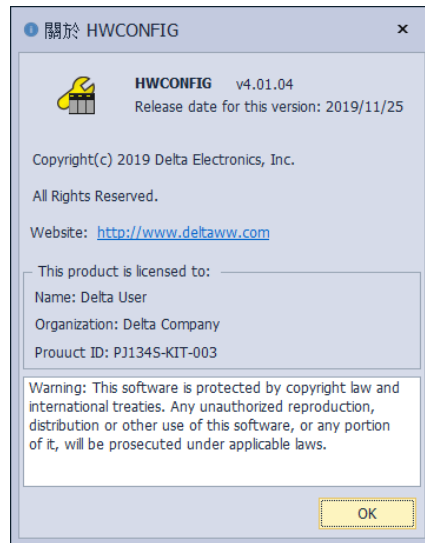
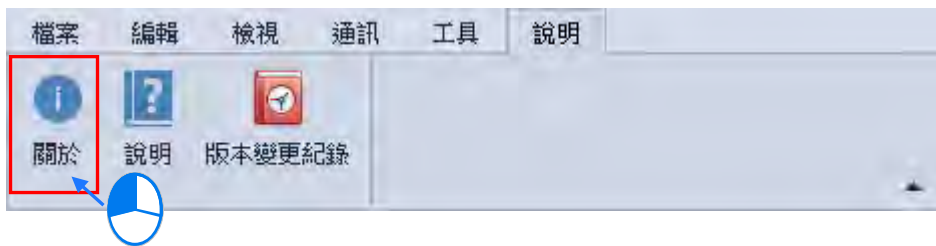
說明：主要為 HWCONFIG 所提供的一些輔助功能。



操作項	功能說明
關於	顯示軟體版本及發行日期等相關資訊。
說明	顯示 HWCONFIG 的使用說明。
版本變更紀錄	顯示此版本的新增與修改項目。

● 關於

於「說明」內點選「關於」會顯示 HWCONFIG 的軟體版本、發行日期、以及軟體資訊。



3.1.2.8 功能工具列-常用

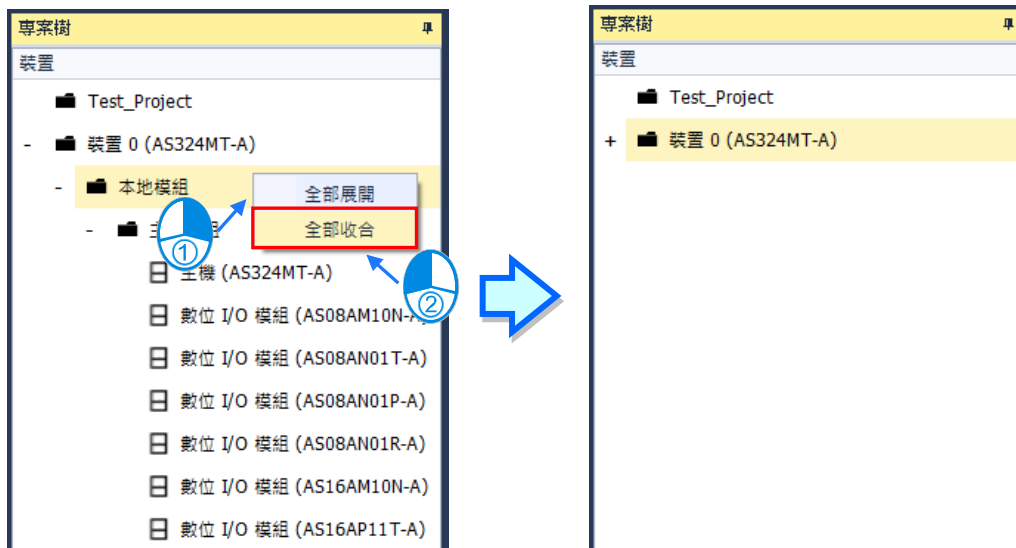
說明：主要為 HWCONFIG 常用選項。



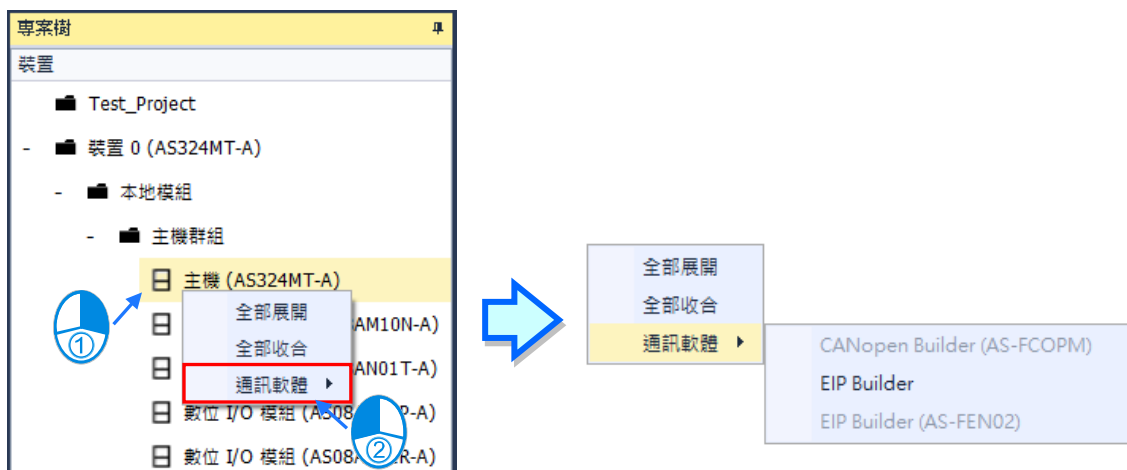
3.1.2.9 專案樹

專案樹的主要內容為目前專案所配置的裝置，並以階層樹狀結構做為其管理介面。

專案樹視窗預設會在隱藏在 HWCONFIG 視窗的左側，可以使用第 3.1.3.12 節的功能來固定或隱藏視窗，也可以從「檢視」中的專案樹選項來更改其顯示方式，在專案樹中某個物件上按下滑鼠右鍵時，會出現「全部展開」以及「全部收合」2 種選項，「全部展開」為展開所有樹狀圖的裝置、「全部收合」為收起所有樹狀圖的裝置。



在專案樹中的主機點擊右鍵，有個通訊軟體的項目，字體會呈現白色是指目前主機支援且可以打開的通訊軟體，字體會呈現灰色是指目前主機有支援但需配置特定功能卡才可以使用的通訊軟體。

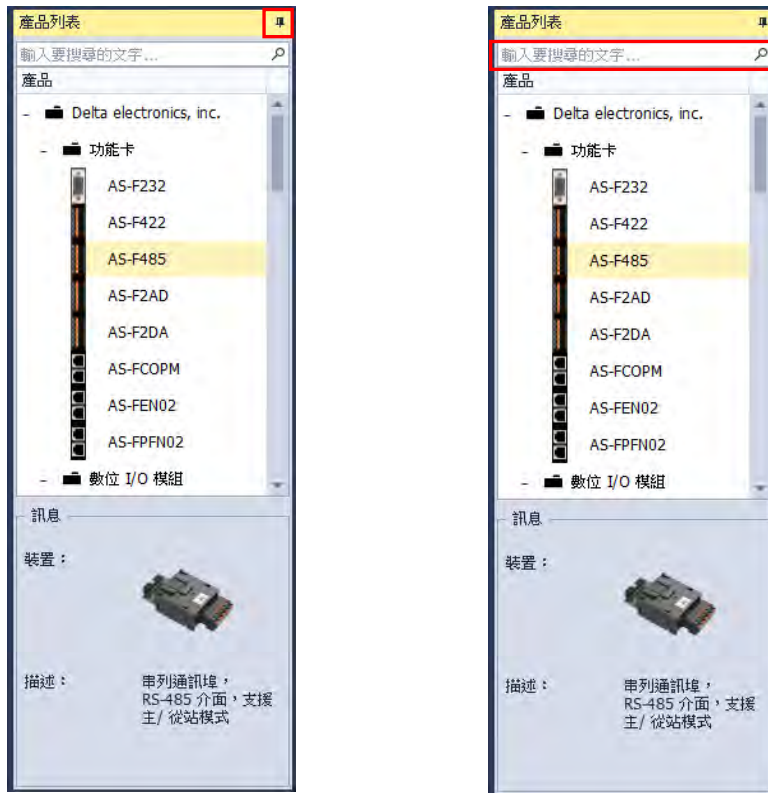


3.1.2.10 產品列表


產品列表主要顯示專案機種所支援的功能卡以及模組，並以階層樹狀結構做為其管理介面，點選任一裝置後，會於下方的訊息視窗顯示裝置圖片及描述資料。預設的**產品列表**會在固定在 HWCONFIG 視窗的右側，可以使用第 3.1.3.12 節的功能來顯示或隱藏**產品列表**。

產品列表上方有搜尋功能，輸入關鍵字可快速地找到裝置，如下方右側圖。

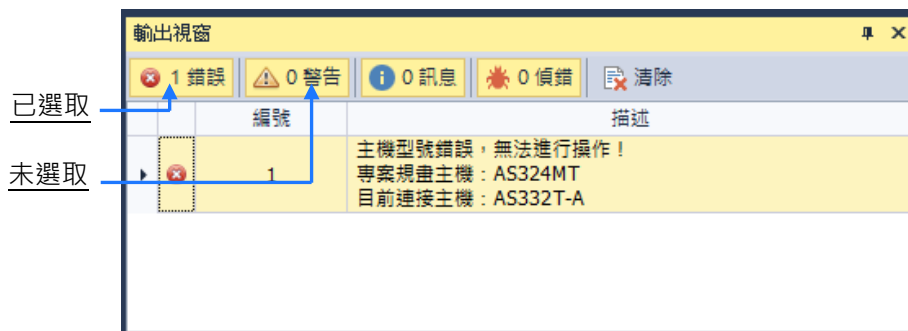
3



3.1.2.11 輸出視窗


輸出視窗主要是顯示進行硬體配置或連線模式時的提示訊息。預設的專案樹會在隱藏在 HWCONFIG 視窗的下方，可以使用第 3.1.3.12 節的功能來固定或隱藏輸出視窗，若點選右上角的  圖示可將其關閉，而透過功能工具列的「檢視」中的「輸出視窗」則可將其開啟或關閉。

輸出視窗下方有 5 個選項，選取錯誤、警告、訊息、與偵錯時會於描述內顯示訊息資訊，反之則不會顯示，若點選清除時則會清除所有資訊。



3.1.2.12 硬體配置區

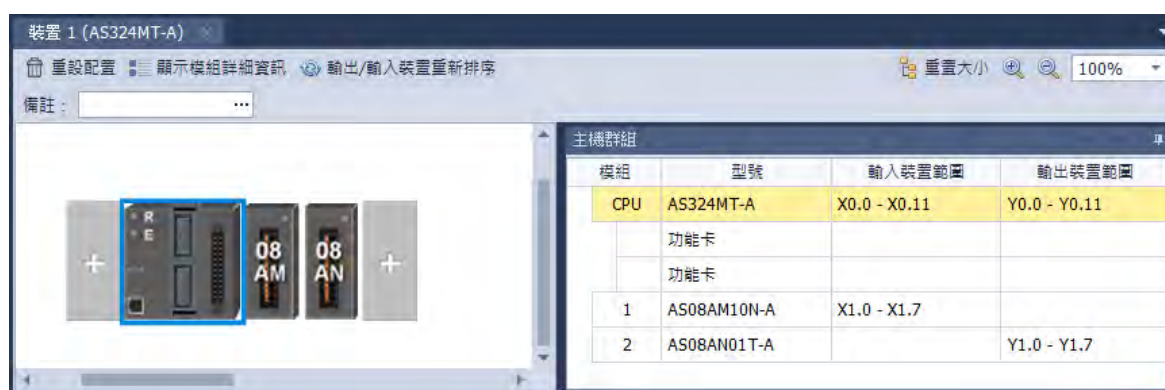
硬體配置區主要是設定專案的模組與功能卡配置。

單機版專案中若有多個 PLC 裝置，上方的分頁標籤可以讓使用者切換到不同 PLC 裝置，也可以使用右上角  圖示來切換分頁；而透過 ISPSOft 開啟的 HWCONFIG 專案無法新增 PLC 裝置，所以只會出現一個分頁。

硬體配置區上方有「清空模組」、「訊息」、「重新排列地址」、「重置大小」、「放大」、「縮小」、「百分比」以及「備註」。

操作項	功能說明
清空模組	清空所有配置的模組，以及主機參數設定。
訊息	顯示或關閉主機及模組名稱等資訊。
重新排列地址	重新排列右邊畫面輸入裝置範圍以及輸出裝置範圍的地址。
重置大小	將模組配置圖回到預設大小。
放大	點選以放大模組配置圖。
縮小	點選以縮小模組配置圖。
百分比	使用百分比來放大縮小模組配置圖。
備註	可填入備註訊息。

硬體配置區主要有兩個區塊，左邊為專案的模組配置圖；右邊為專案的模組配置相關訊息，包含模組、型號、輸入裝置範圍、以及輸出裝置範圍。

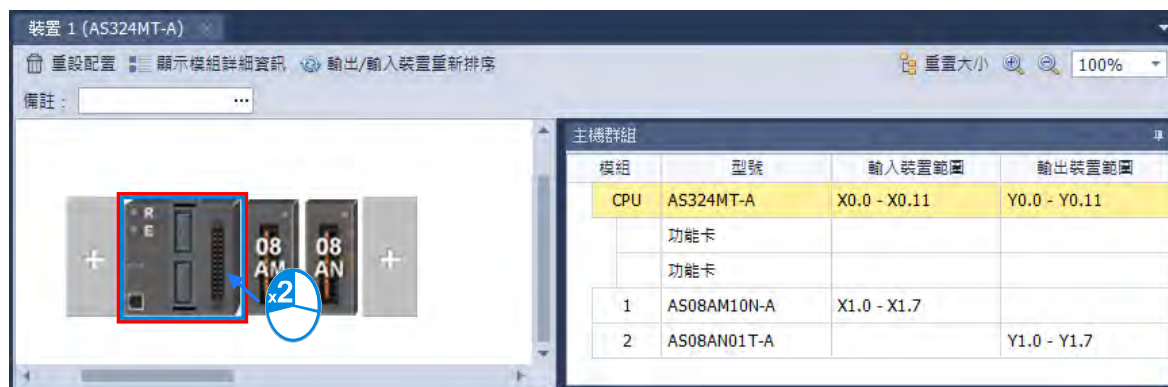




3.1.2.13 編輯區

編輯區主要是設定 PLC 與模組參數，PLC 包含系統設定、通訊埠設定、乙太網路設定、以及功能卡設定與模組的參數設定、資料交換的參數設定；模組參數設定請參照各模組操作手冊。

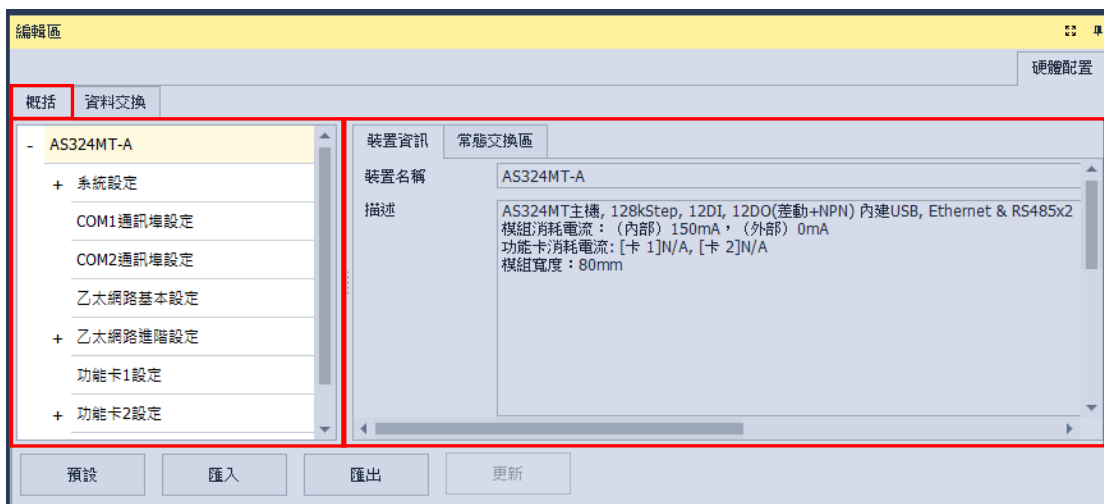
預設的**編輯區**不會在 HWCONFIG 中顯示，必須要雙擊主機開啟編輯區，如下圖。

3



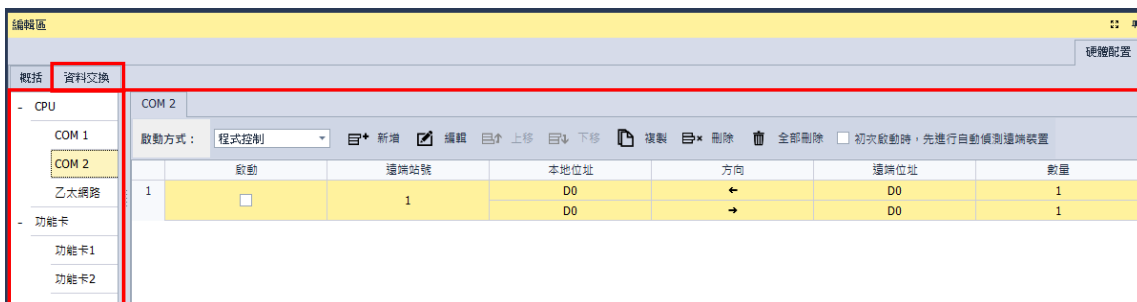
也可以使用參考第 3.1.2.6 節的環境設定將顯示模式改變後，可以使用下方第 3.1.3.12 節的功能來顯示或隱藏。點選**編輯區**右上角  圖示會將編輯區放大，但會覆蓋硬體配置區的視窗，若要顯示硬體配置區只要再次點擊  圖示，或從「檢視」選取「重置佈局」也可將硬體配置區還原。

編輯區的「**概括**」頁面左側會顯示主機、以及功能卡的參數設定選項；若在硬體配置區的模組配置圖上選取任一模組時，則此頁面會顯示模組的參數設定選項，右側則是顯示裝置資訊與常態交換區資訊。



3

「資料交換」頁面可以設定不同通訊方式的資料交換設定，右側為設定資料交換之內容。



3.1.3 進階功能操作

本章節將會介紹 HWCONFIG 軟體的進階功能。

3.1.3.1 新增模組

將模組加入至硬體配置區有多種方式，以下將會介紹所有新增模組的方式。

● 方法一

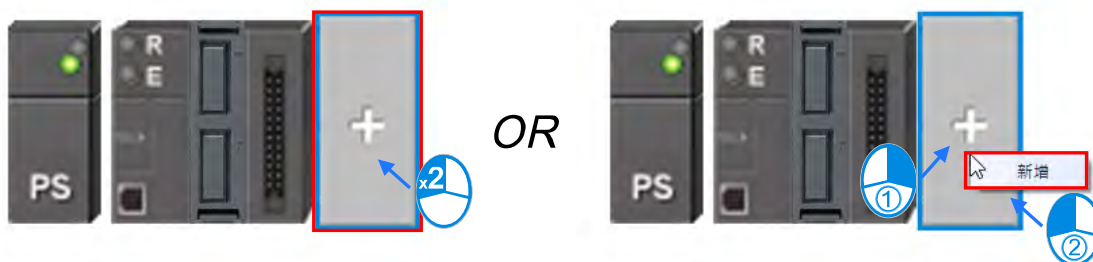
於右側**產品列表**中選取要加入的模組後，雙擊滑鼠左鍵，或直接將其拖曳至硬體配置區的空白處，會有虛線顯示拖曳要放置的位置，接著再放開滑鼠左鍵即可加入。

3

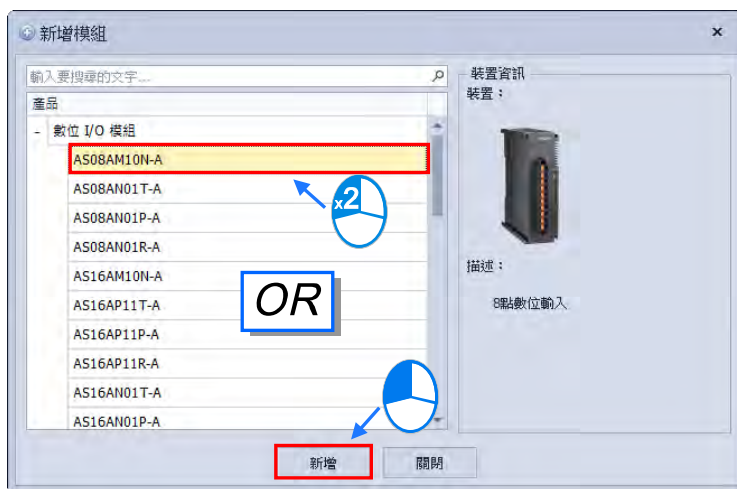


● 方法二

在模組配置圖的灰色加號圖示雙擊滑鼠左鍵或是點擊右鍵後選擇「新增」。



在要加入的模組上雙擊滑鼠左鍵或點擊「新增」就可以新增模組，可連續增加不同的模組，模組會依序向右側增加，完成操作後點選「關閉」即可關閉此視窗。

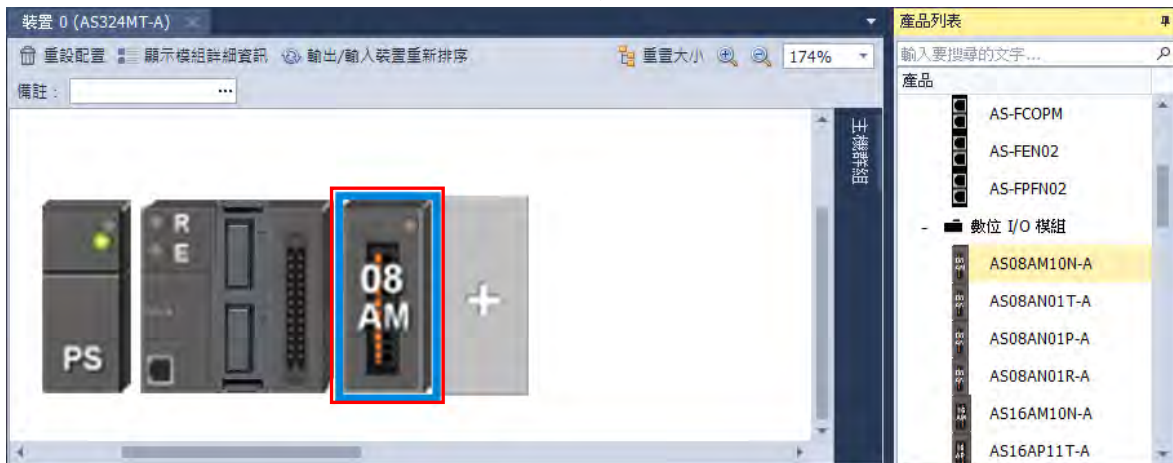


3

● 方法三

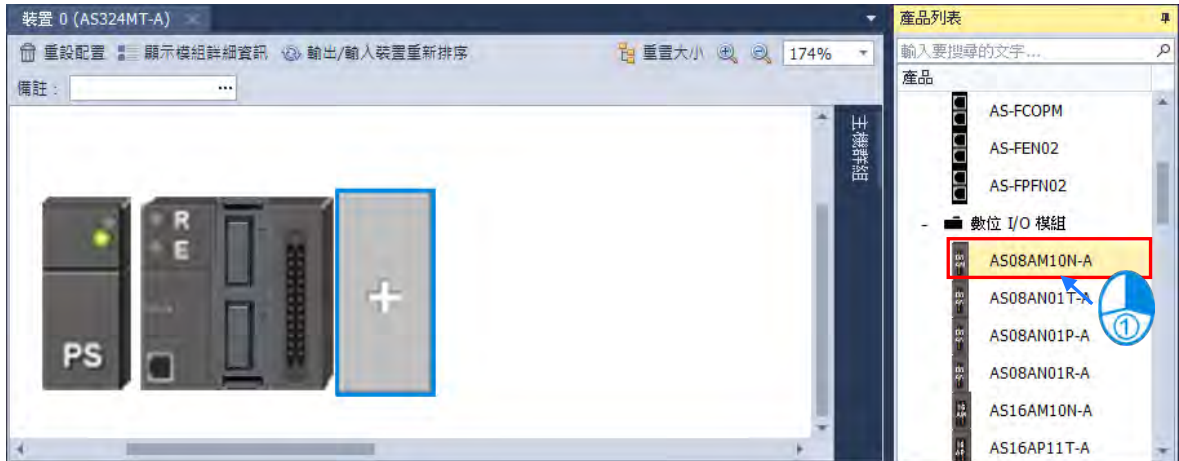
在產品列表中對要加入的裝置雙擊滑鼠左鍵或輸入 Enter 鍵，都會將裝置加入至硬體配置區。

3

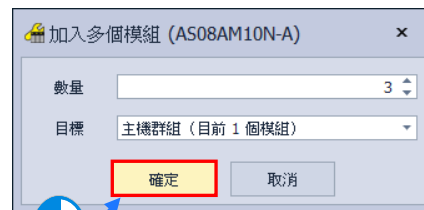
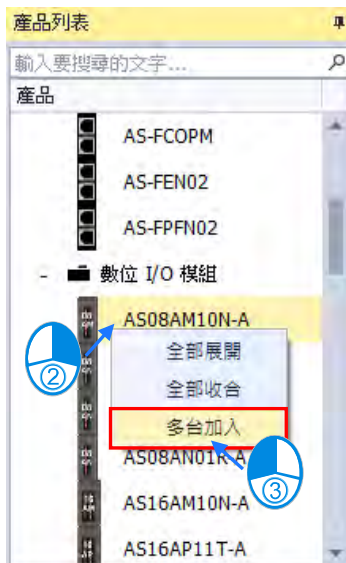


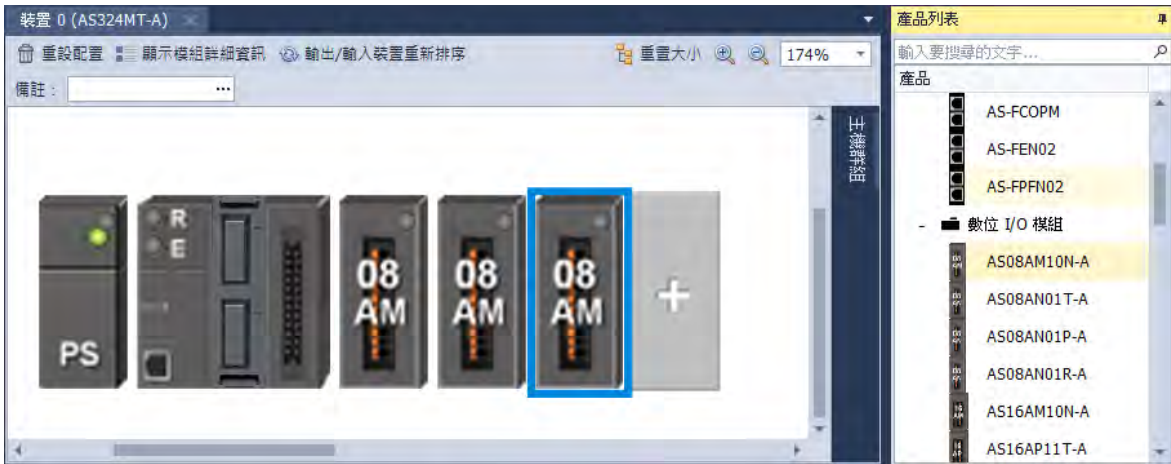
● 加入多台模組

按下滑鼠右鍵時，畫面便會出現「全部展開」、「全部收合」、以及「多台加入」等 3 種選項，「全部展開」為展開所有樹狀圖的物件；「全部收合」為收起所有樹狀圖的物件；「多台加入」為一次增加多台選定的模組至專案中，選擇多台加入後會出現加入多個模組視窗，可以在「數量」中填入想要加入的模組數量，按下「確定」後即會將模組加入。



3



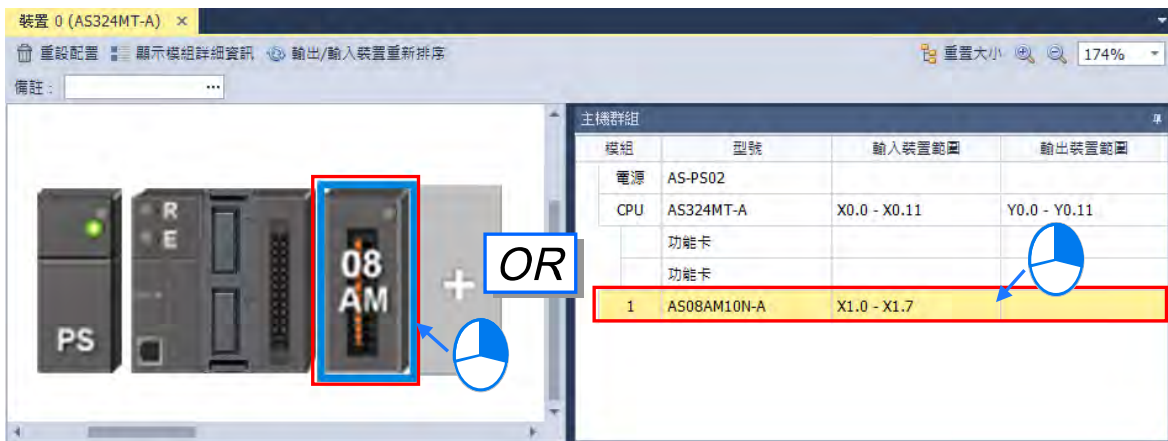


3

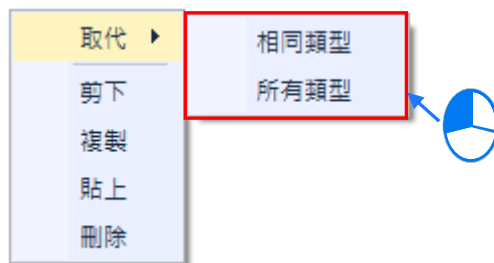
3.1.3.2 變更模組

若是想要對硬體配置區的模組進行取代變更，可以依循以下步驟：

- (1) 於硬體配置區的模組配置區或模組配置訊息上點選要變更的模組後按下滑鼠右鍵。



- (2) 於右鍵快捷選單中點選「取代」，之後再選擇取代的方式。關於兩種取代的方式說明如下。



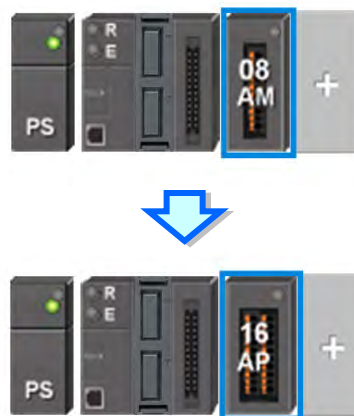
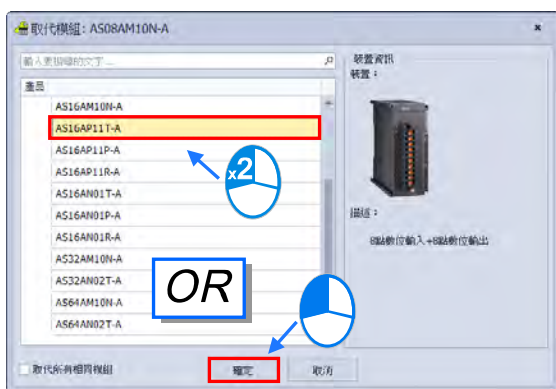
➤ 相同類型

僅可以取代為所選取模組的相同類型之其他模組，而取代之後的模組參數設定若無法完全對應時將會被恢復為系統預設值。

➤ 所有類型

可將所選模組取代為所有類型的模組，而執行之後，其效果等同將原本的模組刪除後，再重新插入另一模組，因此新模組的輸入/輸出裝置範圍將會重新配置，而模組的其他參數設定也將會重新恢復為系統預設值。

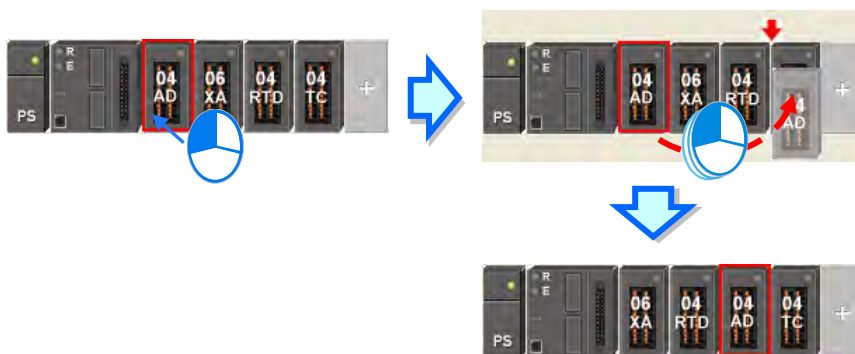
- (3) 點選要執行的取代方式後，畫面便會出現模組選擇視窗，視窗標題會顯示被取代的模組，而視窗中可選擇的模組項目，會依據所選的取代方式而有所不同；決定要替換的模組之後，於該模組項目上雙擊滑鼠左鍵或按下「確認」鍵即可取代模組。



3.1.3.3 模組的搬移

除了 CPU 之外，配置於硬體配置區的模組圖示皆可直接使用滑鼠拖曳來進行搬移。

AS 系列機種為無背板設計，故拖曳某一模組至其他模組的位置上時，將插入到目標模組之前的位置，並且會在插入的位置出現一個紅色的箭頭標示。



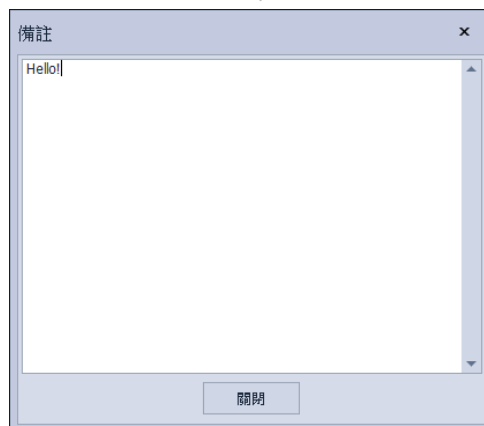
* 搬移時，模組註解、模組內部參數、D 對應裝置與智慧型模組的進階參數亦會跟隨模組移動，且輸入/輸出裝置範圍也會依據模組擺放位置而自動重新排。

3.1.3.4 備註編輯

- 專案備註

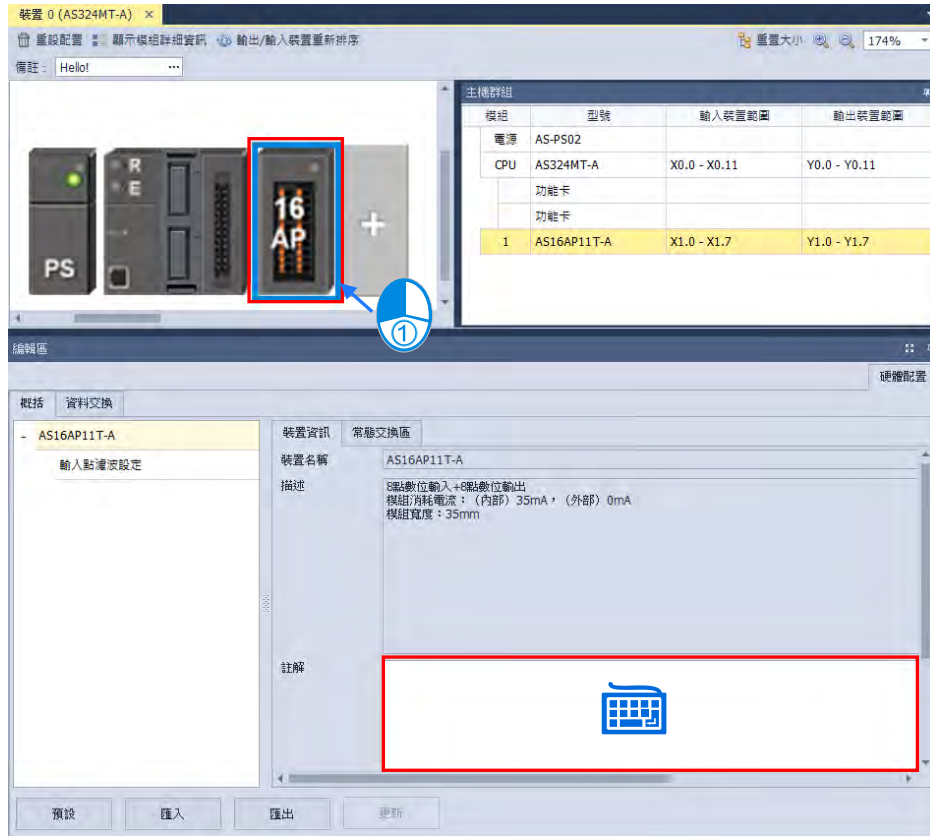
滑鼠左鍵點擊**硬體配置區**上方的備註欄位後，即會跳出備註視窗，便可輸入該專案的備註，而輸入完畢後按下鍵盤【Enter】鍵或點選「關閉」選項即可進行儲存。

3



● PLC 主機與模組註解

滑鼠左鍵點擊**硬體配置區**中的 PLC 主機或模組後，下方**編輯區**會出現其資訊，而在裝置資訊內有註解欄位，點擊滑鼠左鍵即可進行編輯。

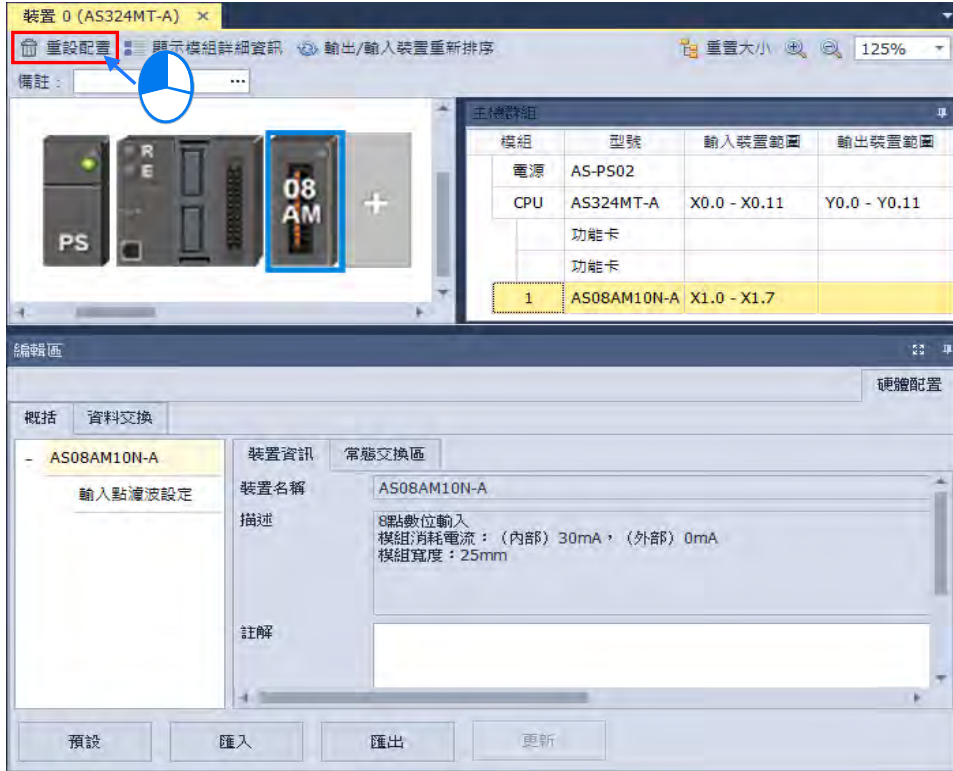


3

3.1.3.5 硬體配置區-重設配置

使用滑鼠左鍵點選**重設配置**，可清除所有模組以及 PLC 主機內部設定。

3



3.1.3.6 硬體配置區-顯示模組詳細訊息

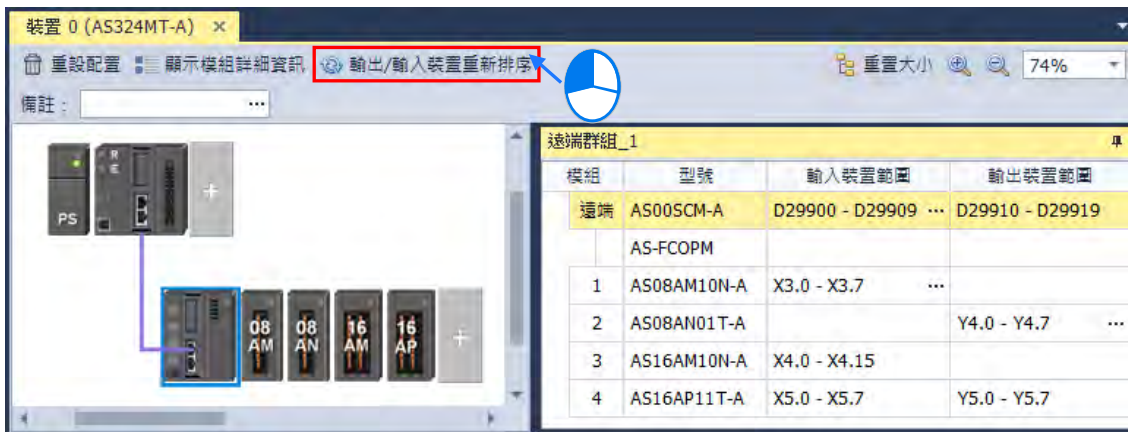
使用滑鼠左鍵點擊**硬體配置區**上方的**顯示模組詳細訊息**後，於硬體配置區將會顯示 PLC 及模組名稱，再次點擊就會隱藏顯示資訊。



3

3.1.3.7 硬體配置區-輸出/輸入裝置重新排序

在進行模組配置的過程中，可能會經過許多的搬移或是複製/剪下/貼上的操作，而此時各模組的**輸入/輸出裝置範圍**也可能已經被系統自動重新配置，而不是從可用的裝置範圍最前端分配，此時便可點選 HWCONFIG 硬體配置區上方的「**輸出/輸入裝置重新排序**」選項，之後所有模組的**輸入/輸出裝置範圍**便會由可用的裝置範圍最前端的裝置來分配。



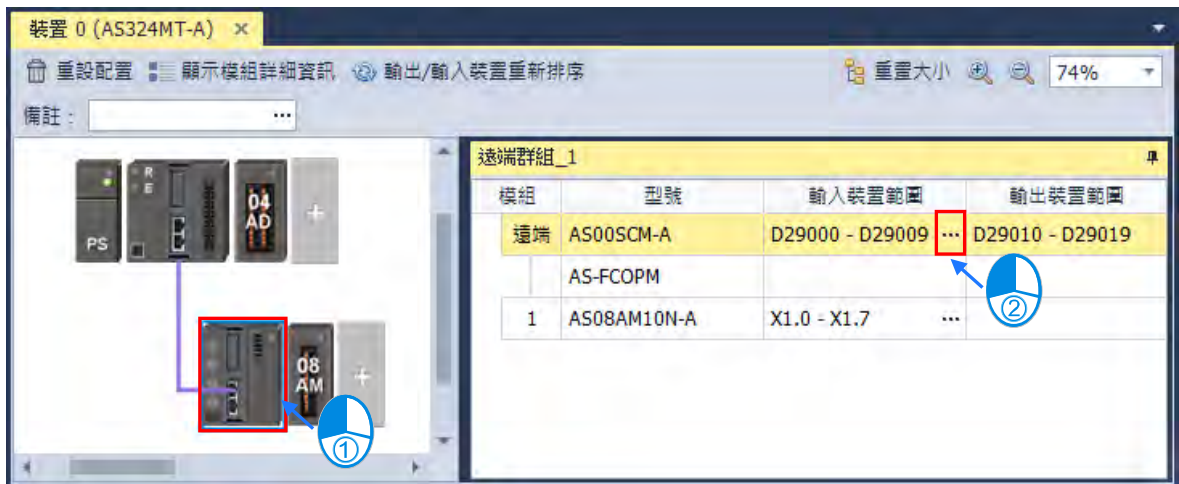
3



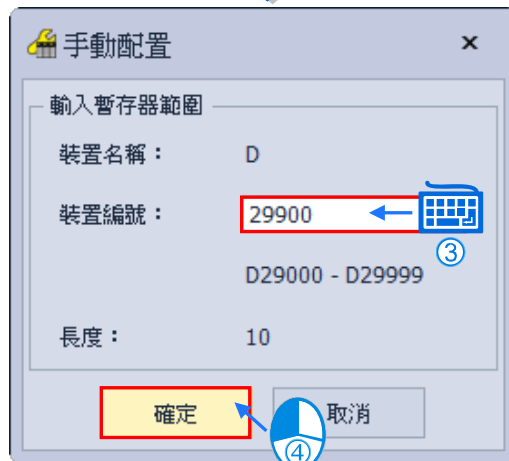
模組	型號	輸入裝置範圍	輸出裝置範圍
遠端	AS00SCM-A	D29000 - D29009 ...	D29010 - D29019
	AS-FCOPM		
1	AS08AM10N-A	X1.0 - X1.7 ...	
2	AS08AN01T-A		Y1.0 - Y1.7 ...
3	AS16AM10N-A	X2.0 - X2.15	
4	AS16AP11T-A	X3.0 - X3.7	Y2.0 - Y2.7

3.1.3.8 硬體配置區-改變遠端裝置位址

在進行模組配置的過程中，使用者可能希望能自行定義遠端裝置位址，而此時遠端群組的「**模組配置區**」的輸入/輸出裝置範圍右方會出現 **...** 的圖示，滑鼠左鍵點擊後即會出現位址設定視窗，輸入裝置編號後按下「**確定**」按鈕，便會更新此遠端模組的輸入與輸出裝置範圍，並且依序自動配置安裝的模組輸入與輸出裝置範圍。



3



模組	型號	輸入裝置範圍	輸出裝置範圍
遠端	AS00SCM-A	D29900 - D29909 ...	D29910 - D29919

3.1.3.9 硬體配置區-重置大小

在進行模組配置的過程中，可能會經過許多的搬移或是複製/剪下/貼上的操作，硬體配置區的畫面可能任何位置，圖像大小比例過大或過小，此時可以使用滑鼠左鍵點擊「重置大小」後，硬體配置區將會把 PLC 主機至於頁面中，並將硬體配置大小設置成 100%。



3

3.1.3.10 編輯區-模組內部參數的匯出與匯入

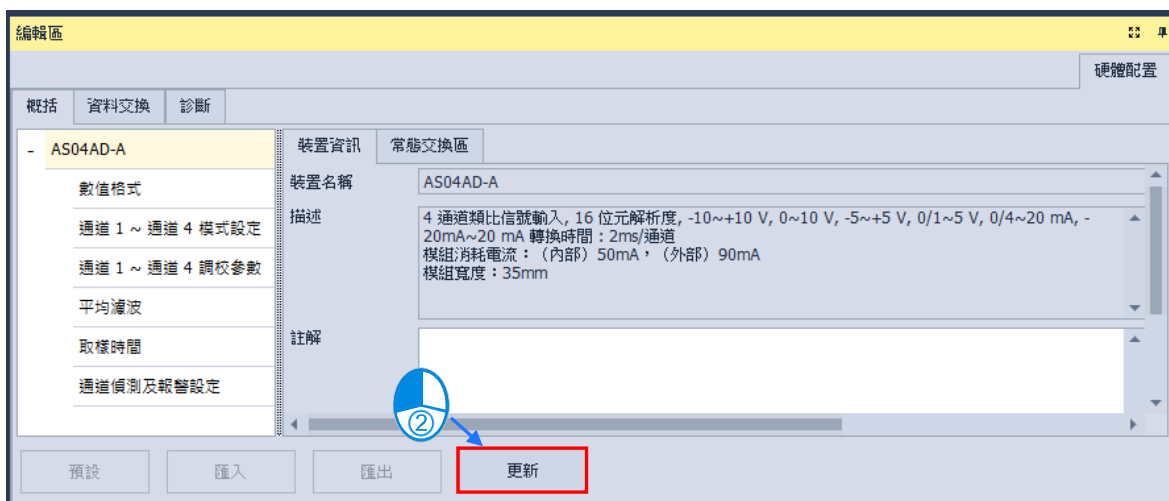
所有模組內部的參數值皆可匯出為 dep 檔另行備份，同時亦可將先前所匯出的檔案重新匯入。要匯出時，按下參數設定視窗的「匯出」按鈕，接著於檔案視窗中指定要匯出的路徑與檔名後儲存即可。要匯入時，按下參數設定視窗的「匯入」按鈕，接著於檔案視窗中指定要匯入的檔案後開啟即可。







匯入模組參數時系統會先檢查檔案格式及模組型號，當檔案格式或模組型號不符時，系統將會禁止匯入的操作，並顯示警示訊息。

3.1.3.11 編輯區-模組內部參數的更新

若要做模組內部參數的更新，首先，需進入線上模式，若要進入線上模式請參考第 3.1.2.5 節，進入線上模式後，點選想要更新參數的模組，接著點擊編輯區下方的**更新**，此時模組內部參數即完成更新，此功能僅限於數位輸入 I/O、類比 I/O 及網路模組。



3.1.3.12 顯示/隱藏功能

點選 HWCONFIG 的功能工具列  或  的圖示，可以將工具列功能隱藏。當工具列功能隱藏後，將滑鼠移動至標籤時，視窗會自動展開，只要鼠標離開該區域一段時間之後，該區域的視窗便會自動隱藏，若要切換視窗的顯示方式，點選  圖示，當圖示切換成  狀態時視窗會固定在 HWCONFIG 視窗中。

3.1.3.13 資訊檢視與模組診斷

- (1) 於線上模式點選 CPU 或模組圖示後按下滑鼠右鍵，並於選單中點選「診斷」項目，下方編輯區將會出現診斷分頁，顯示該模組的錯誤記錄。





編輯區

概括 資料交換 診斷 硬體配置

更新 清除

目前錯誤紀錄

群組編號	模組編號	模組名稱	錯誤代碼	日期及時間	描述
1	1	AS04AD-A	16#1801	2019/11/26 00:00:00	模組外部電源錯誤
2	1	AS04AD-A	16#1801	2019/11/26 00:00:00	模組外部電源錯誤

- (2) 在診斷的分頁中，「目前錯誤紀錄」欄位會顯示目前的錯誤狀態。
- (3) 當錯誤狀況已排除時，按下視窗中的清除圖示  便可清除畫面上與模組中儲存的錯誤記錄，並可解除模組的錯誤狀態；按下更新圖示  時，系統便會重新擷取資料並將其更新於視窗中。

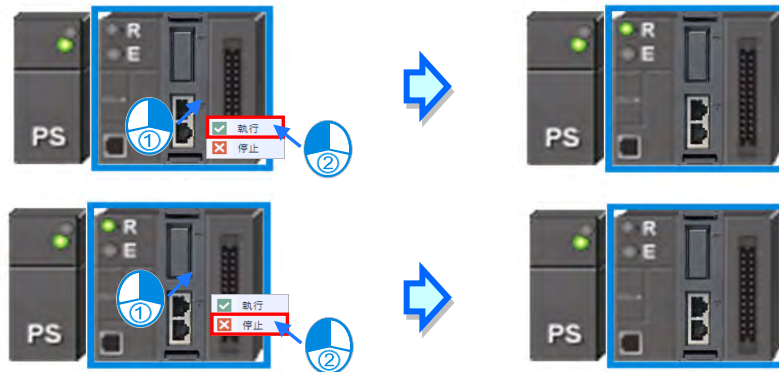
3.1.3.14 線上變更模組狀態

HWCONFIG 支援於線上模式時變更裝置本身的狀態，使用者可藉此進行簡單的測試。

 變更裝置狀態之前，請務必確認所做的操作不會造成系統或人員的傷害。

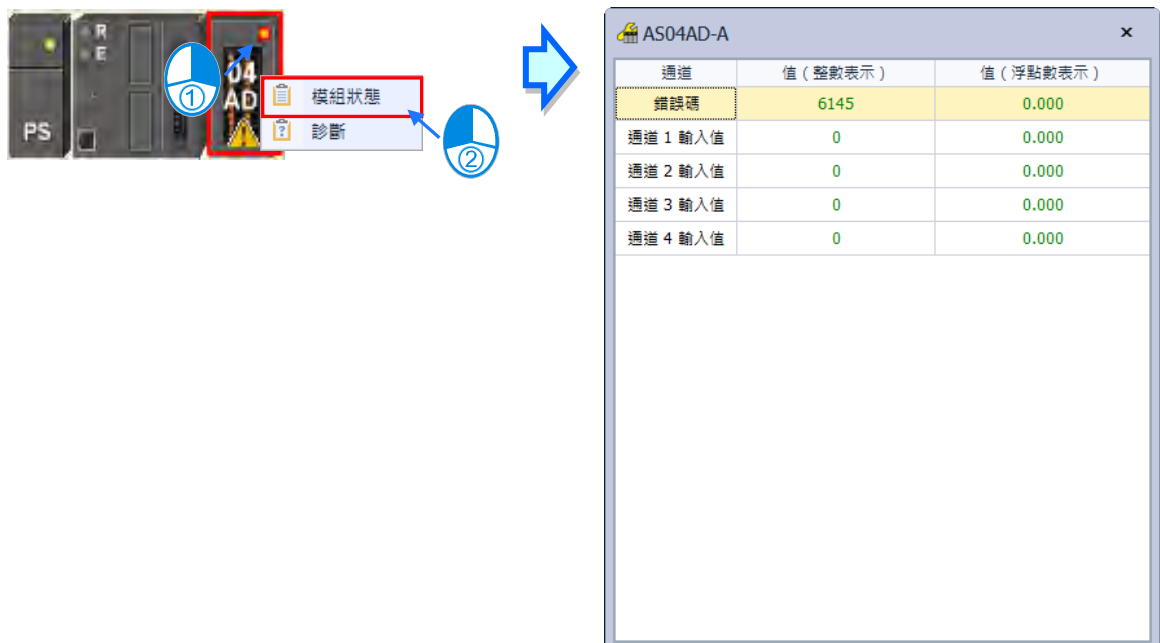
● 變更裝置的運轉狀態

於線上模式時點選 CPU 主機圖示後按下滑鼠右鍵，接著於快捷選單中點選「執行」項目，之後主機與模組便開始執行，其圖示亦會出現綠色的運轉顯示燈號；若點選「停止」項目，則此主機與模組便會恢復至停止狀態。



● 變更模組的輸入/輸出狀態

當硬體配置中包含數位 IO、類比 IO 或溫度模組時，於這些模組的右鍵快捷選單中點選「模組狀態」項目，接著便可於狀態視窗中監控這些 IO 模組的輸入或輸出狀態，其中數位 IO 模組還可透過右鍵選單來對輸入或輸出通道進行強制 ON/OFF 的操作。不過當執行此功能時，CPU 與該模組則必須處於執行 (RUN) 的狀態才會實際產生輸出。

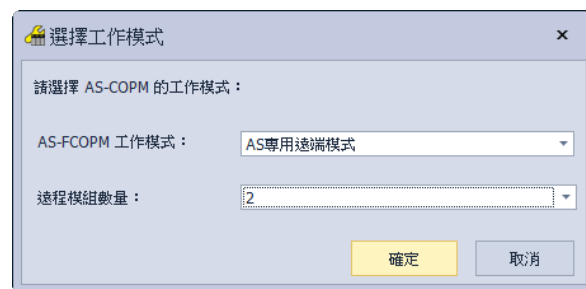
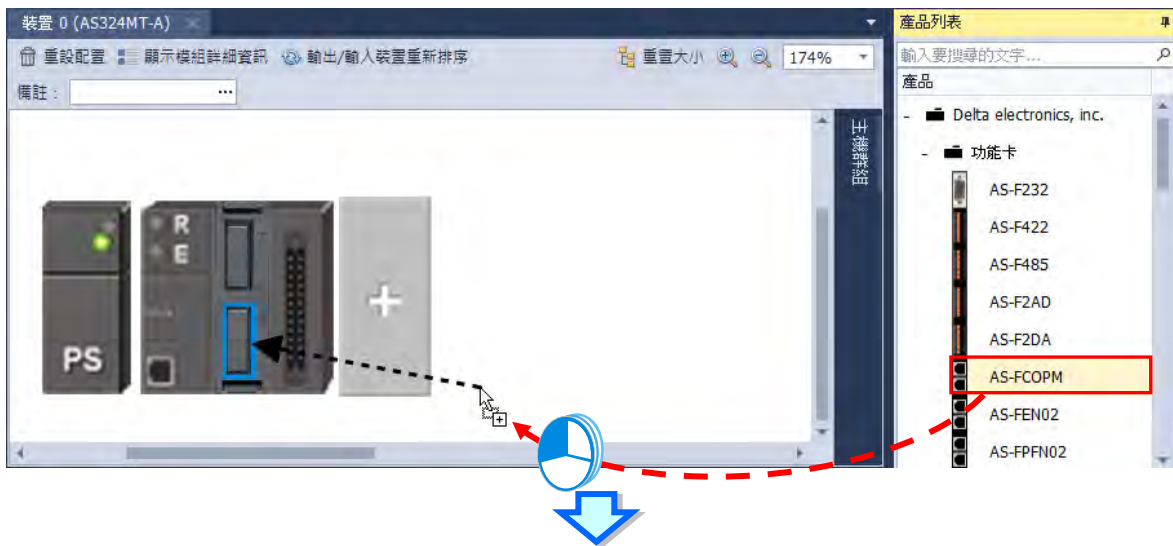


- 強制 ON：將點選的通道狀態強制設定為 ON。
- 強制 OFF：將點選的通道狀態強制設定為 OFF。
- 解除：解除點選通道的強制狀態。
- 全部解除：解除全部通道的強制狀態。

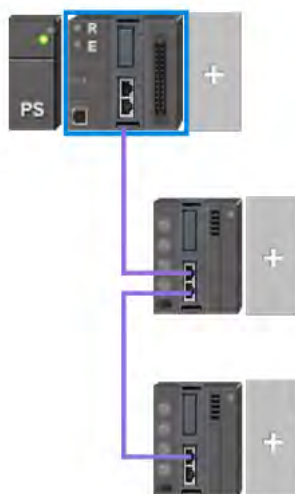
3.1.3.15 遠端模組配置

在產品列表清單中，將 **AS-FCOPM** 模組拖曳至主機的功能卡 2 插槽，便可選擇 **AS-FCOPM** 工作模式將其設定為 AS 專用遠端模式，再於**遠端模組數量**輸入連接的數量，如下圖。

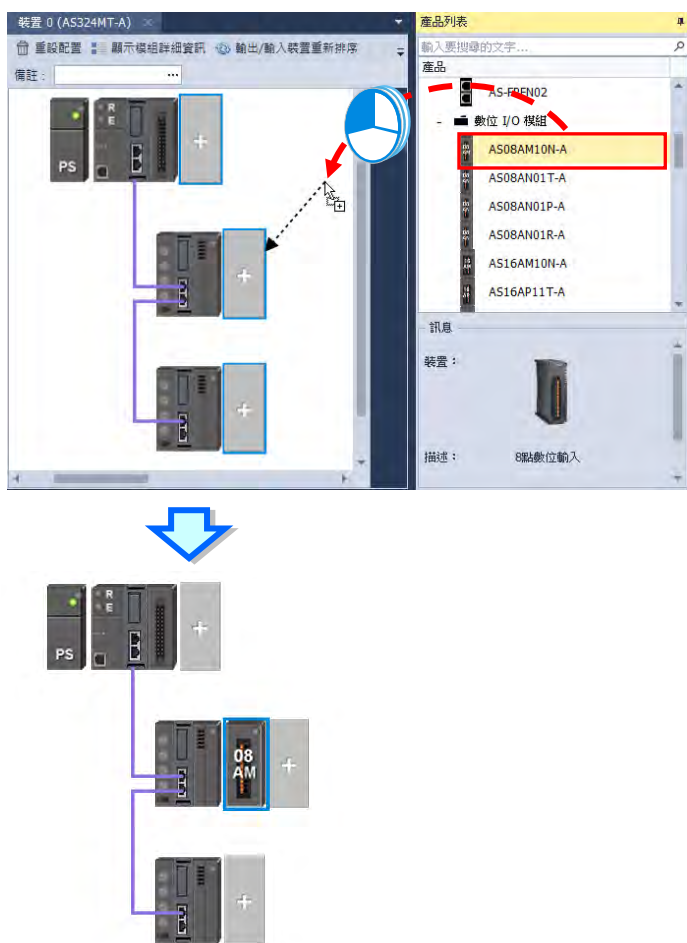
3



若**遠端模組數量**設定為 2 台，按下「**確定**」按鈕後，便會在系統配置區出現新增的 2 台遠端模組。



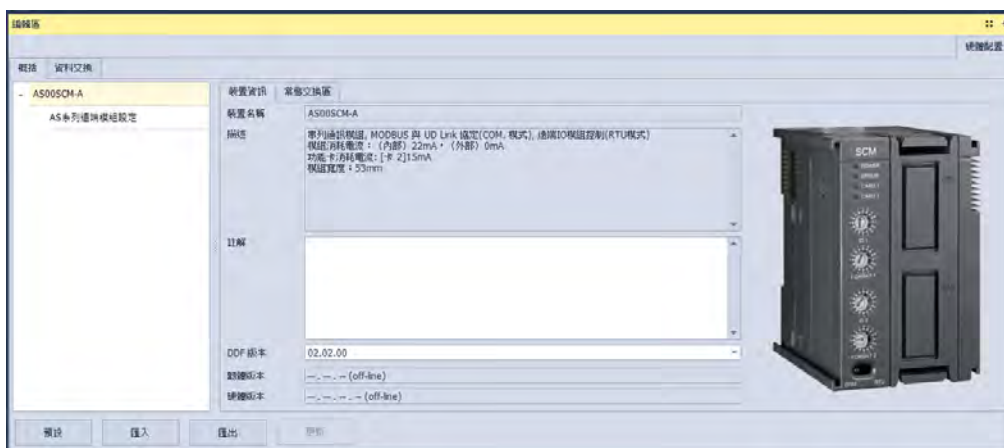
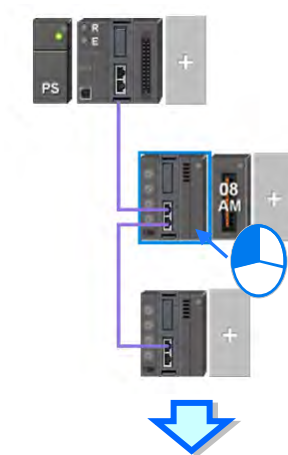
遠端模組只能配置數位 I/O 模組與類比 I/O 模組，其餘模組皆無法配置。



3

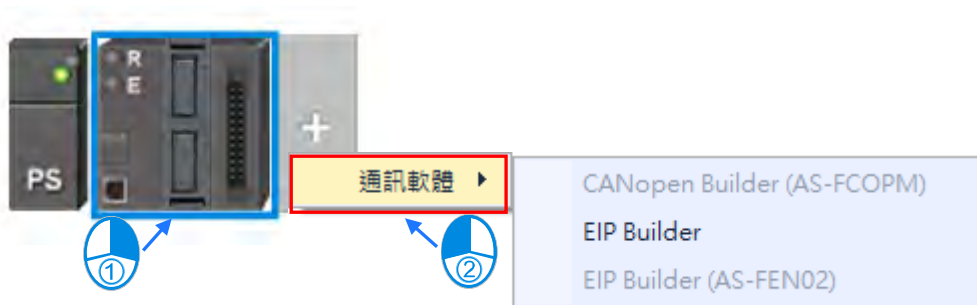
同樣的，可以用滑鼠左鍵點擊遠端模組及其附掛的擴充模組，在下方編輯區頁面進行設定。

3



3.1.3.16 開啟外部軟體

在硬體配置區的主機上點選滑鼠右鍵，接著點選「通訊軟體」，就會出現可以開啟的外部軟體，部分軟體需要搭配相關的功能卡，才可被使用。

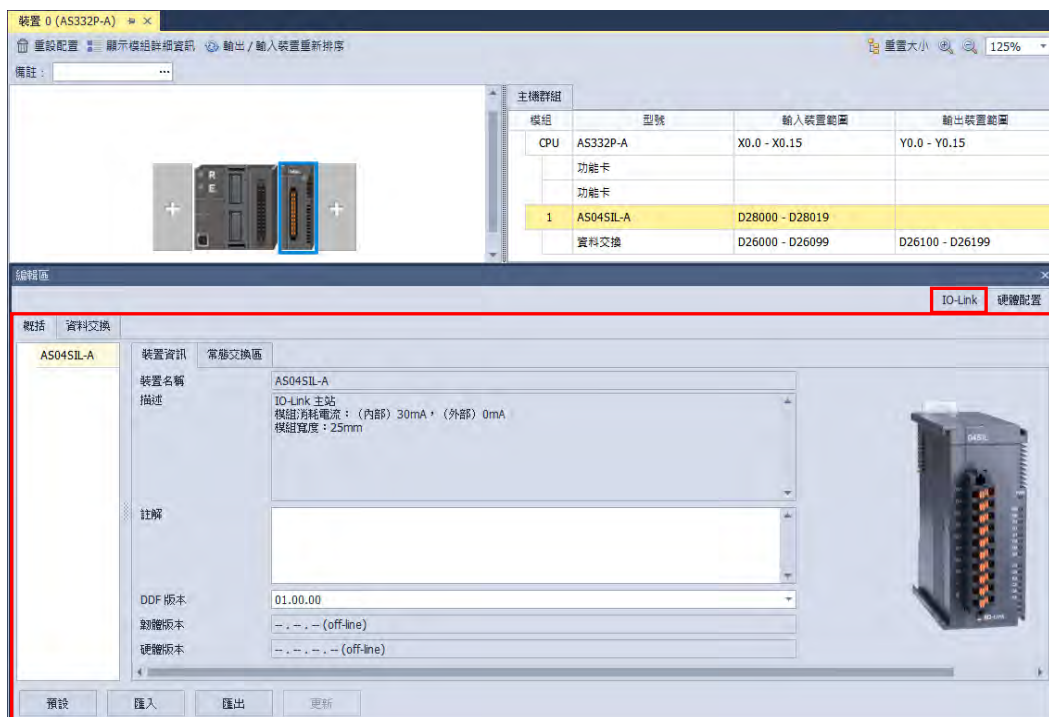
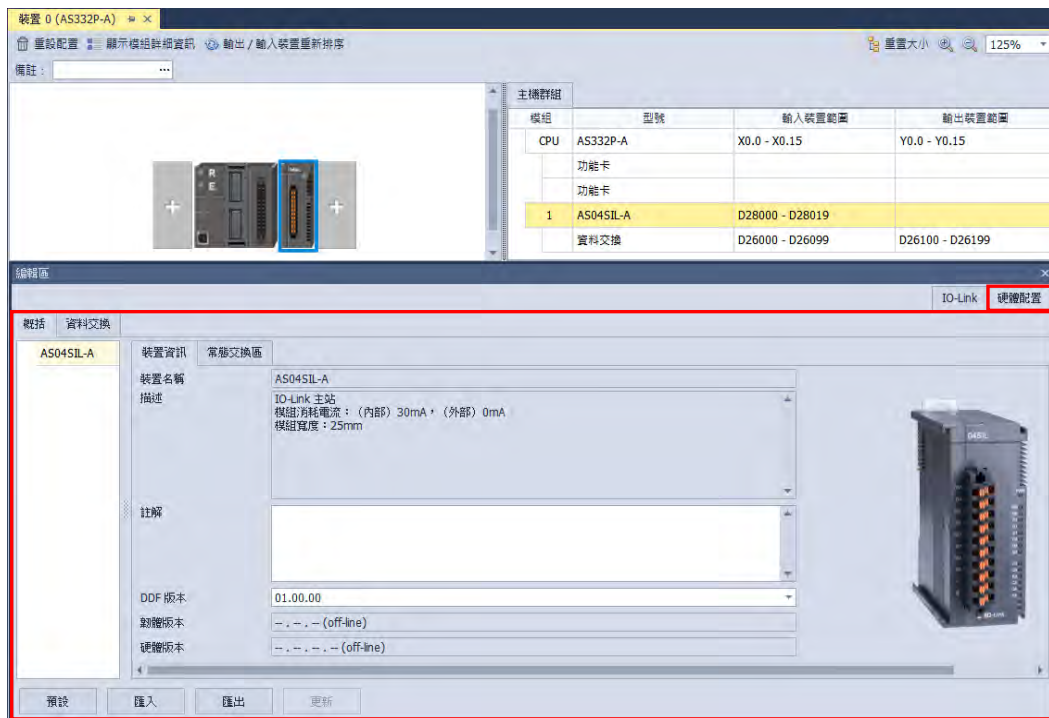


3.1.3.17 IO-Link

詳細設定操作請參考 AS04SIL 模組手冊。

AS04SIL-A 做為 IO-Link 主站支援 IO-Link 設備。主站與裝置之間採點對點連接，採用可靠的 3 線制技術，可通過非屏蔽標準電纜，連接智慧感測器/執行器作為 IO-Link 裝置。可向下相容傳統的數位感測器/執行器。電路狀態和數據通道設計均基於可靠的 24VDC 技術。

AS04SIL-A 編輯區介面有分為「硬體配置」分頁為模組資訊及模組相關設定、「IO-Link」分頁為 IO-Link 資訊及 IO-Link 相關設定，如下圖所示。



3

點選 IO-Link 後編輯區介面左右分別是「裝置列表」模組的裝置配置、「裝置資訊」裝置的詳細資訊。

3



「裝置資訊」的內容會因為「裝置列表」選擇「AS04SIL-A (模組)」或「裝置」而顯示不同資訊，下列做說明。

● AS04SIL-A (模組)

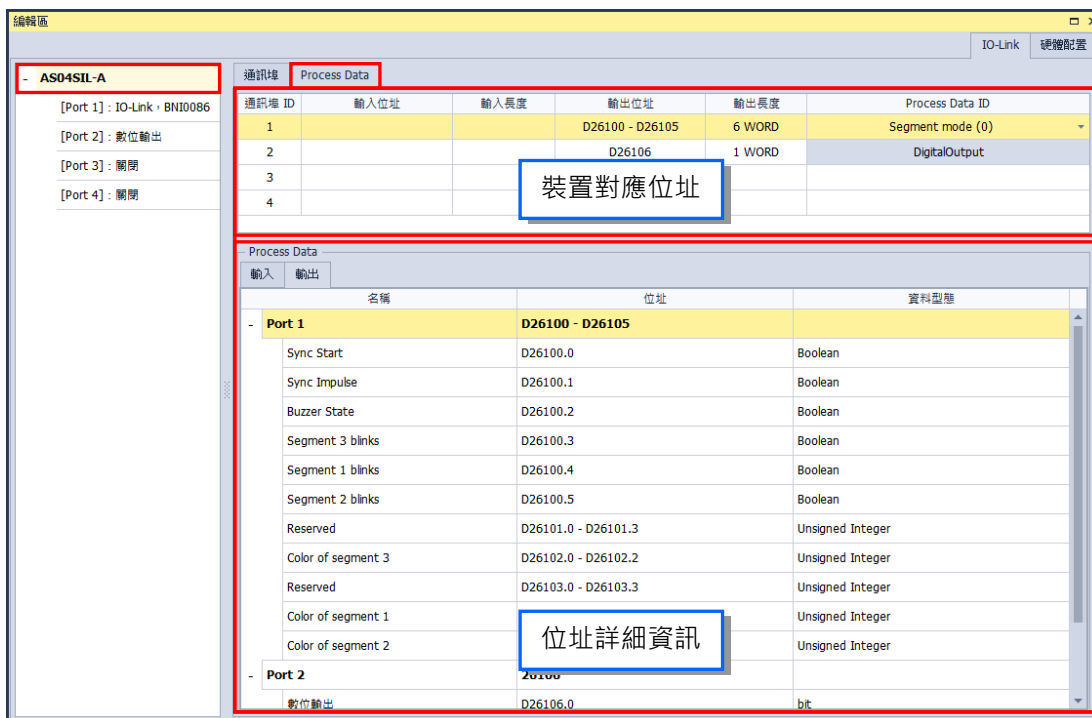
選擇「AS04SIL-A (模組)」時，右側「裝置資訊」內有「通訊埠」與「Process Data」兩個分頁。

「通訊埠」分頁上方的「裝置設定」主要功能為裝置配置以及裝置通訊設定；「通訊埠」分頁下方的「裝置訊息」主要功能為顯示裝置設定中所選取的裝置訊息。



3

「Process Data」分頁上方的「裝置對應位置」主要功能為切換裝置不同的工作模式 (參考 AS04SIL 模組手冊第 12.3.2.2 節) 以及顯示所有裝置對應的主機暫存器位址範圍；「Process Data」分頁下方的「位址詳細資訊」主要功能為顯示所有裝置位址所對應的詳細意義。

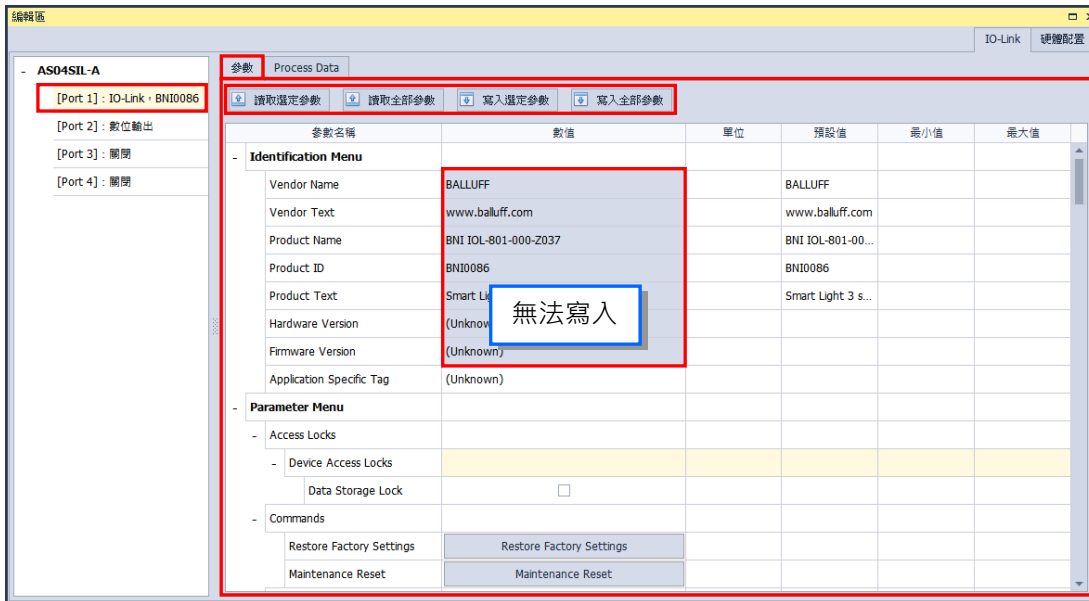


● 選擇裝置

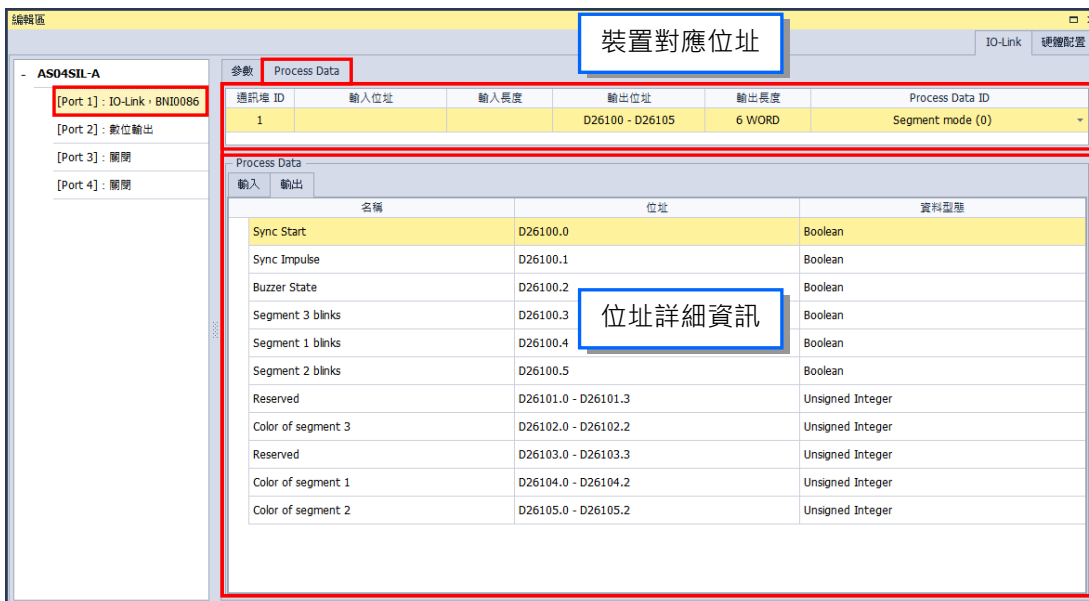
選擇「模組」時，以[Port 1]為例，右側「裝置資訊」內有「參數」與「Process Data」兩個分頁。

「參數」分頁主要功能為裝置參數設定，而頁面上方提供包含針對單一項目或子集進行讀取或寫入，以及對所有參數進行讀取或寫入的四個選項可以使用；頁面顯示參數設定資訊，包含參數名稱、數值、單位、預設值、最小值以及最大值，若是數值欄位為灰底表示該參數只能進行讀取，無法寫入，若為白底表示該參數可進行讀取或寫入。

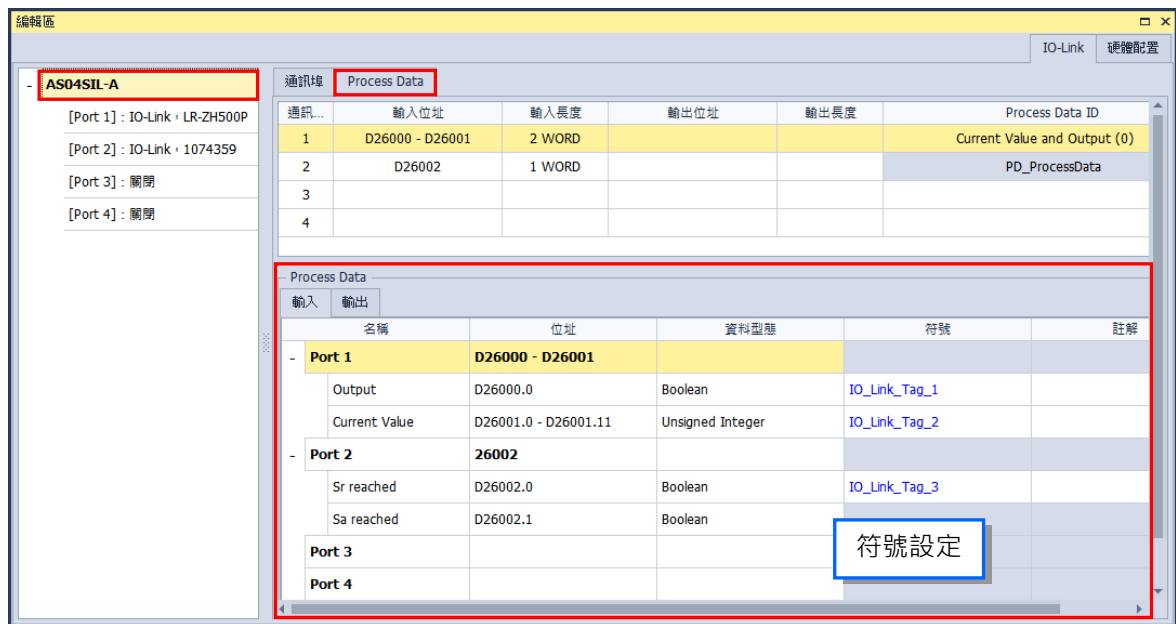
3



「Process Data」分頁上方的「裝置對應位置」主要功能為切換裝置不同的工作模式 (參考 AS04SIL 模組手冊第 12.3.2.2 節) 以及顯示[Port 1]裝置對應的位址範圍；「Process Data」分頁下方的「位置詳細資訊」主要功能為顯示[Port 1]裝置位址所對應的詳細意義。



從 ISPSOft 開啟 HWCONFIG 中「**Process Data**」會有符號設定功能(參考第 3.1.3.18 節)可在 ISPSOft 的全域符號中的 HWCONFIG 符號表中使用，即可在程式中存取對應的 IO-Link 設定參數。

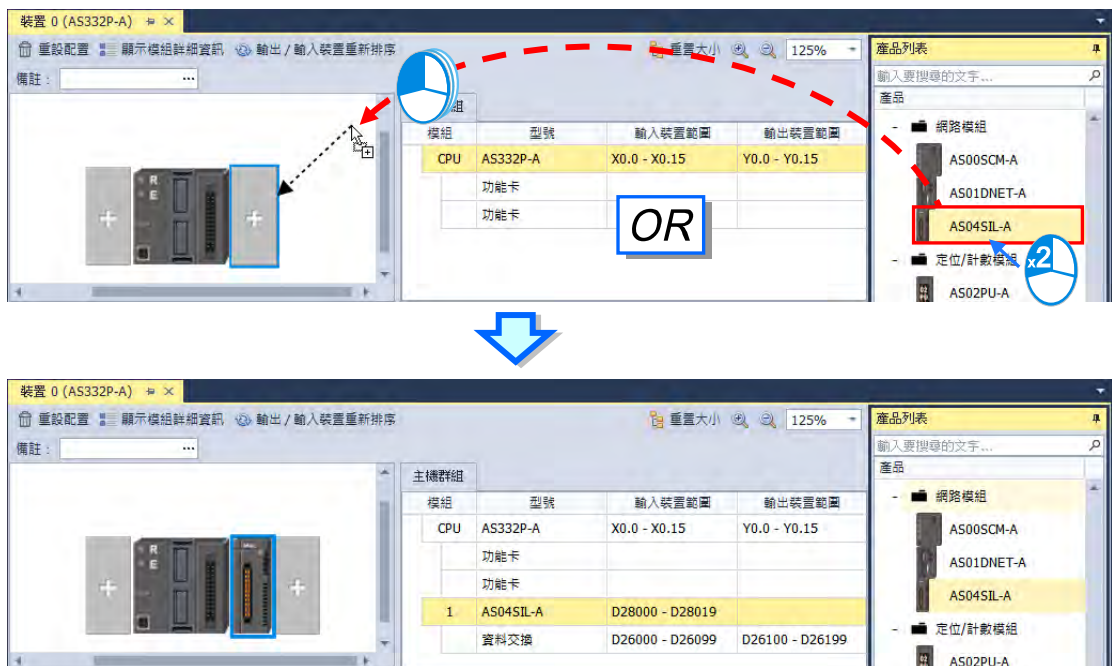


3

IO-Link 應用範例

- 步驟一_加入 AS04SIL-A 模組

在軟體右側「產品列表」->「網路模組」->「AS04SIL-A」將模組加入至硬體配置區，如下圖。



點選模組後，在下方編輯區右上角的「硬體配置」分頁為模組資訊及模組相關設定。

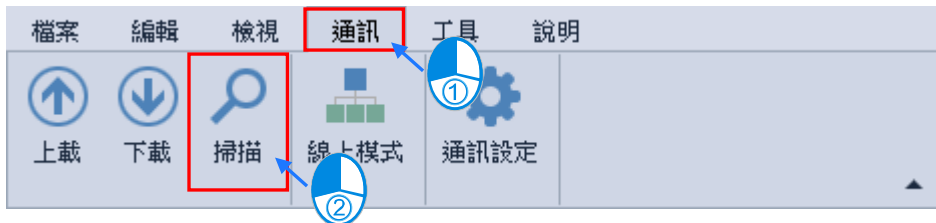
● 步驟二_加入裝置

加入裝置有三種方式

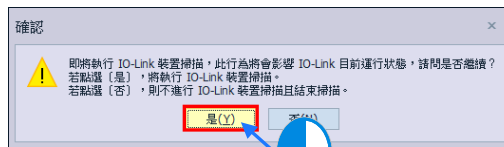
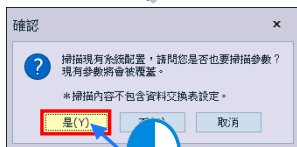
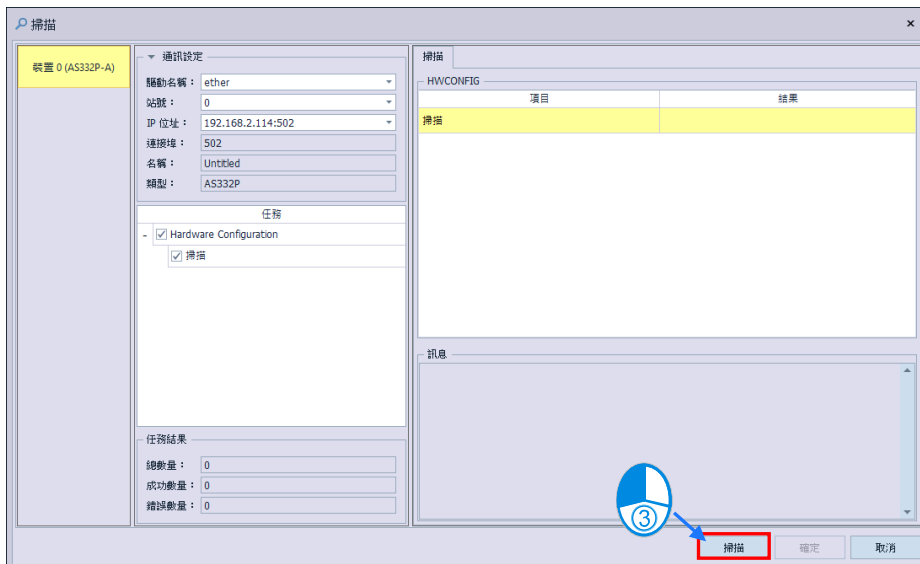
- 方法一_掃描所有模組的裝置並加入

確認硬體都配置好之後，點選功能工具列「通訊」->「掃描」

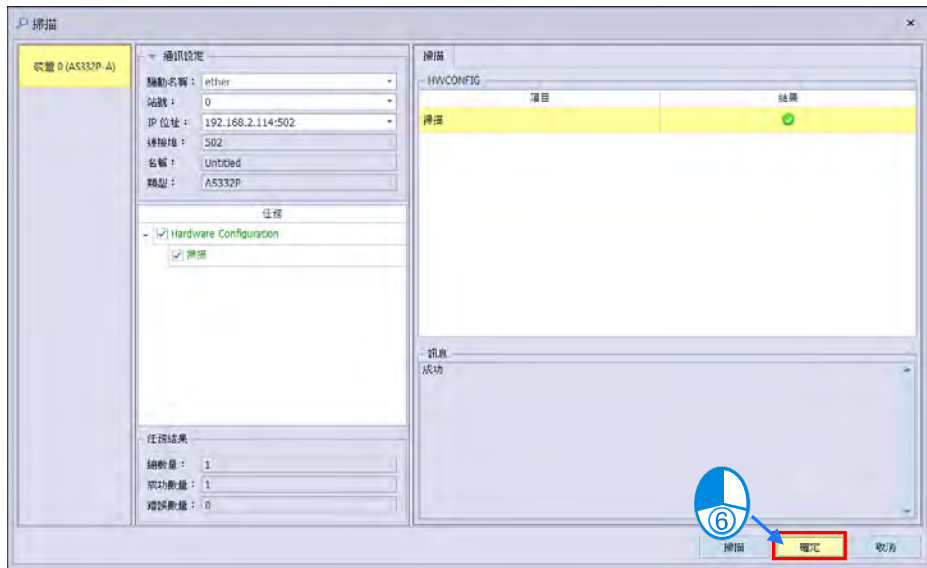
3



掃描視窗出現後點選「掃描」，接著點選「是」確認掃描，接下來點選「是」開始掃描 IO-Link 裝置。



掃描完成後點選「確定」，IO-Link 裝置即會自動加入至模組中。



3



➤ 方法二_掃描單一模組的裝置並加入

選擇欲設定的模組，後點選編輯區左方裝置列表的模組名稱後，接著點選「掃描所有裝置」，在確認訊息點選「是」開始掃描。

3

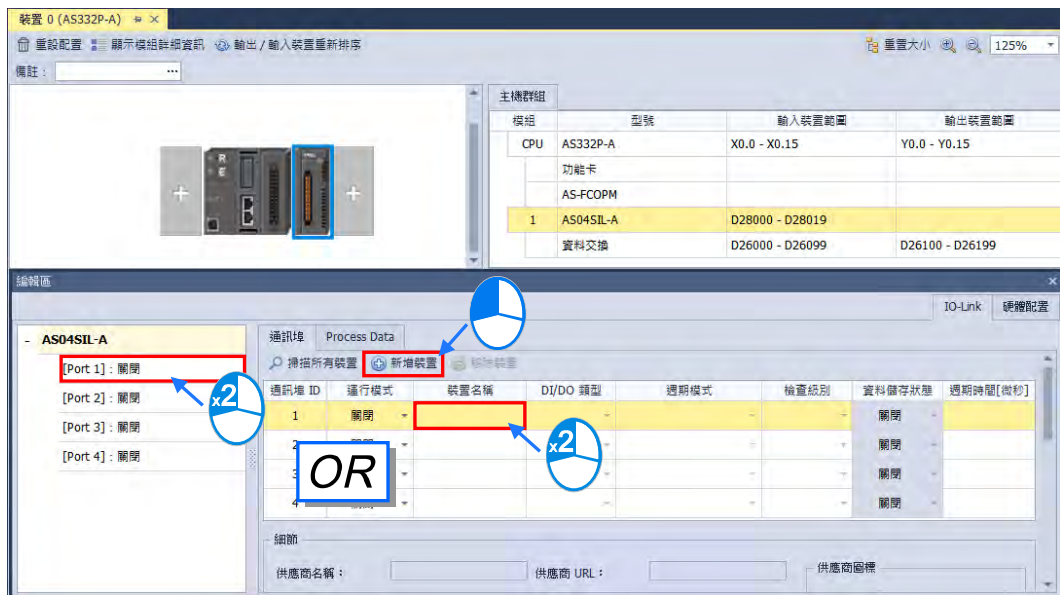


IO-Link 裝置即會自動加入至模組中。



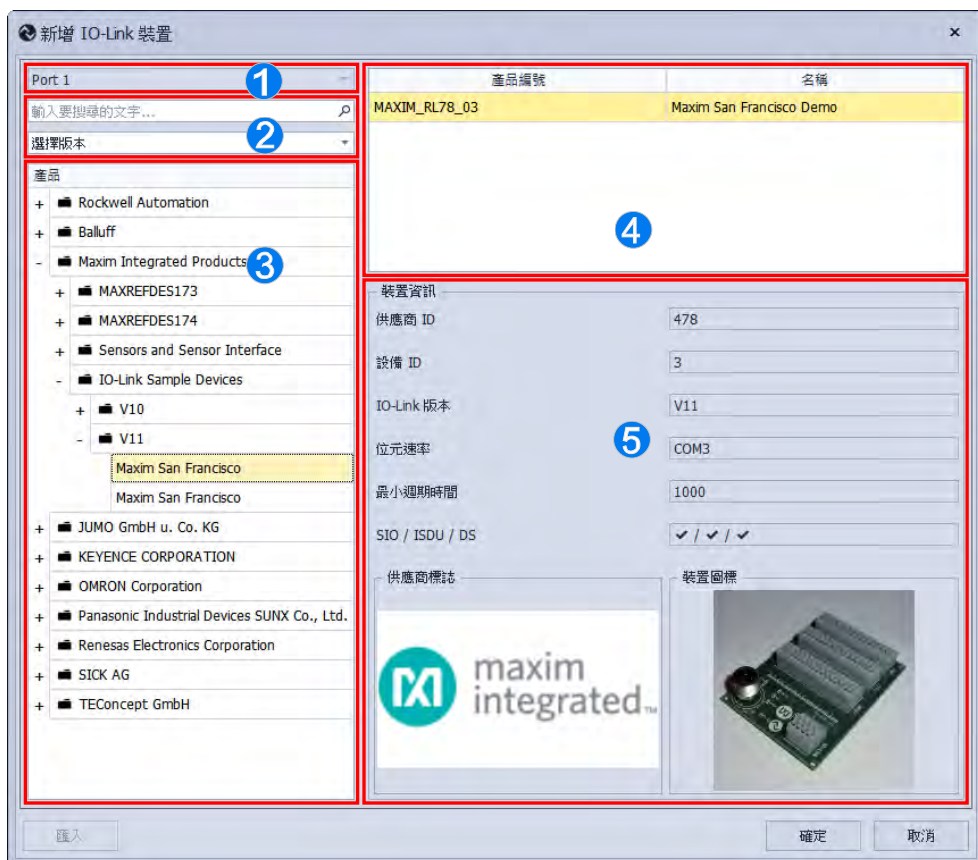
➤ 方法三_手動加入裝置

若要手動在 Port 1 加入裝置，先依照下方圖示點選。



3

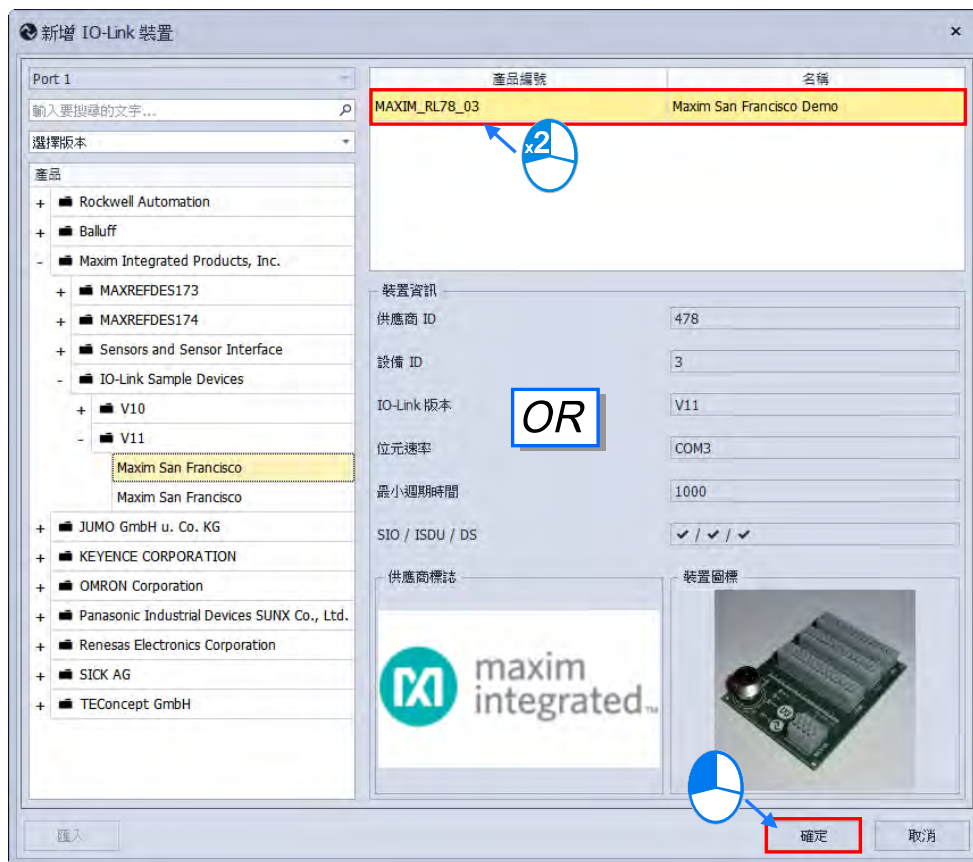
接著會開啟新增 IO-Link 裝置視窗，如下圖。



- ① 裝置新增的位置。
- ② 過濾器可以輸入關鍵字快速尋找裝置。
- ③ 裝置種類及版本。
- ④ 裝置型號。
- ⑤ 裝置詳細資訊。

3

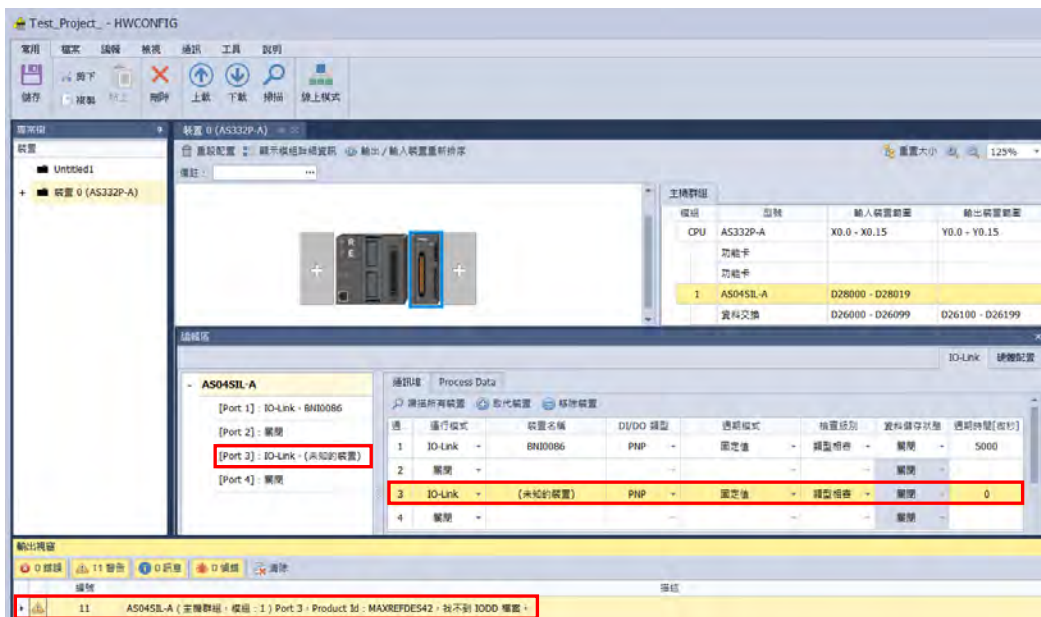
選擇好裝置型號後按下確定 IO-Link 裝置即會自動加入至模組中。





3

掃描後，若裝置沒有 IODD 描述檔，則會呈現未知的裝置，請先重新匯入描述檔後(參考第 3.1.2.6 節)，再重新掃描。

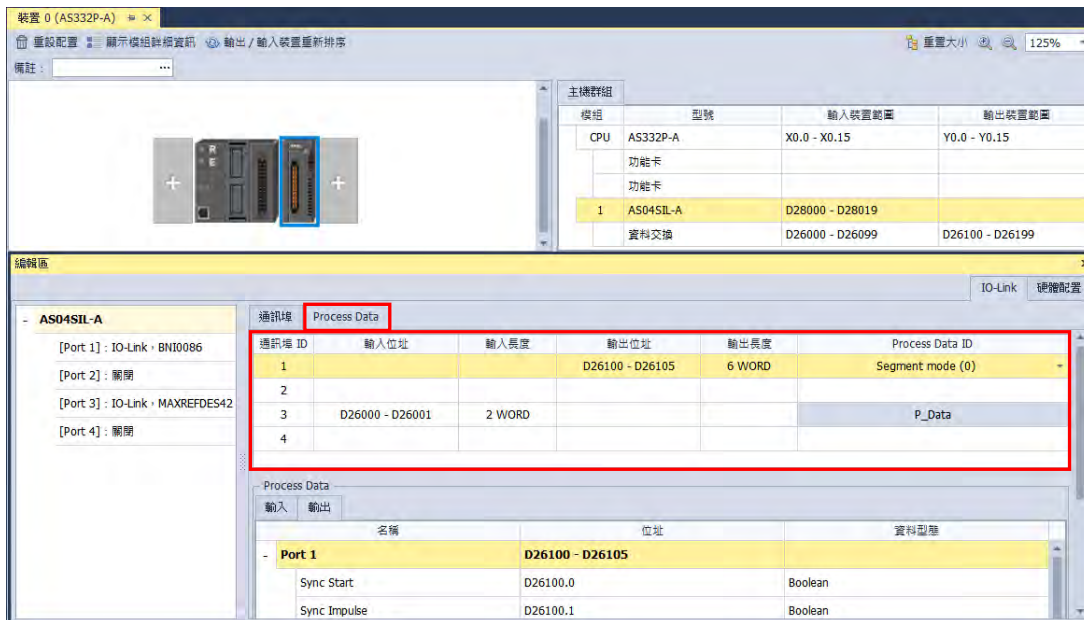
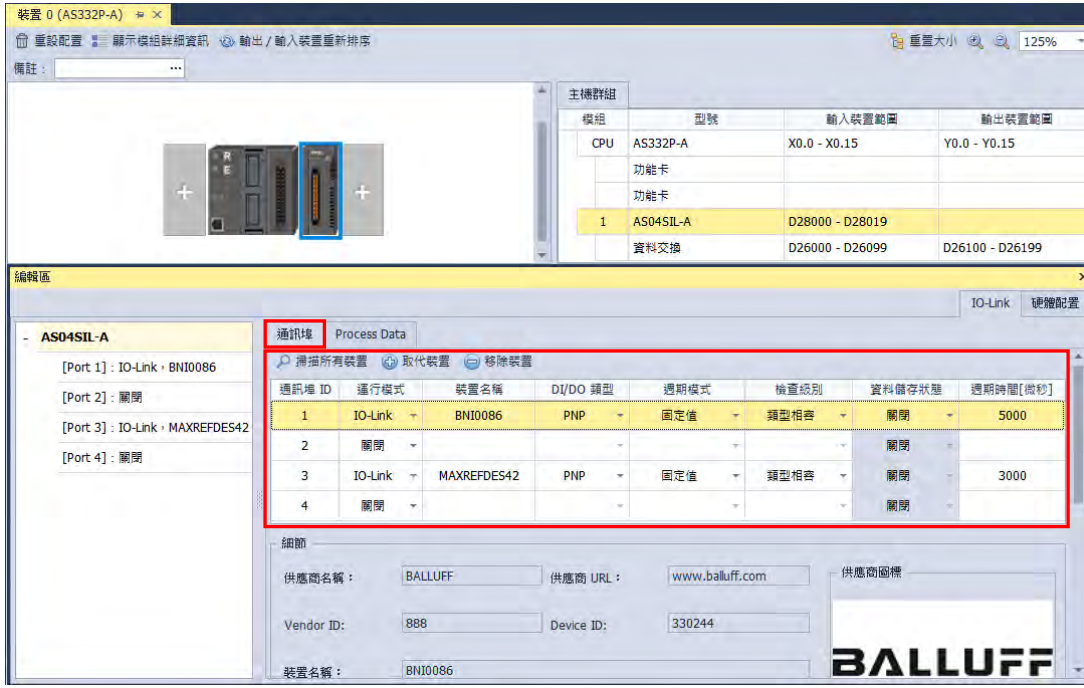


● 步驟三_設定裝置配置 (下載至 PLC)

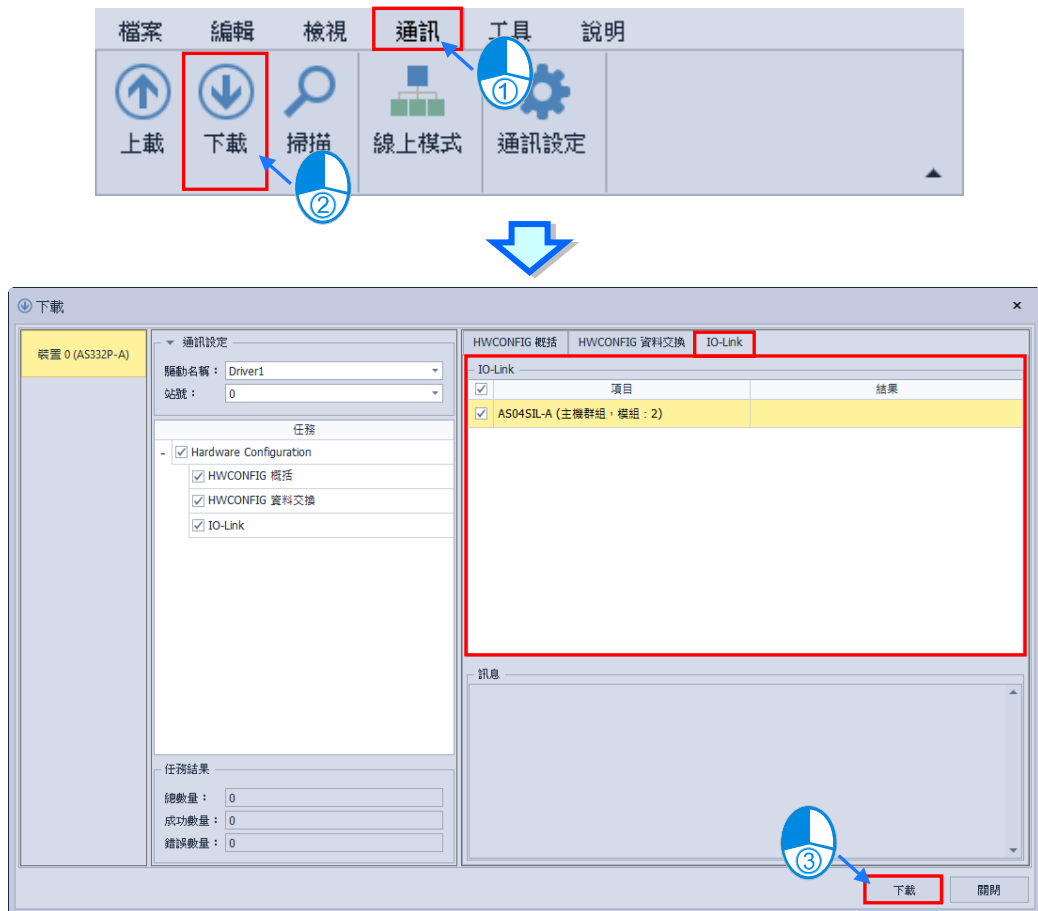
(*下載並不會將裝置參數下載至裝置內)

首先確認設定完所有模組的裝置「通訊埠」設定以及「Process Data」的工作模式後

3



點選功能工具列「通訊」->「下載」，下載視窗出現後，先確認 IO-Link 項目有被勾選，接著按「下載」，將 IO-Link 裝置配置下載至主機。



待成功畫面出現，即下載成功。

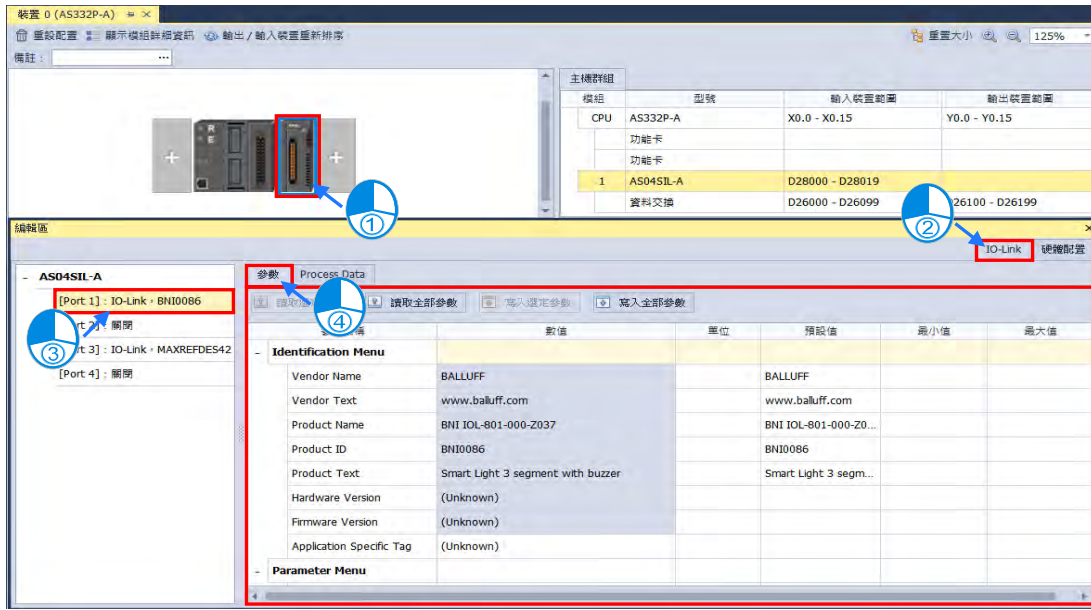
● 步驟四_裝置參數設定

裝置參數設定有兩種，分別為「一般參數設定」以及「Process Data 參數設定」。

➢ 一般參數設定

首先點選模組後，在下方編輯區 IO-Link 分頁中的裝置列表中選擇要設定的裝置，接著右側點選「參數」頁面，下方即可進行設定。

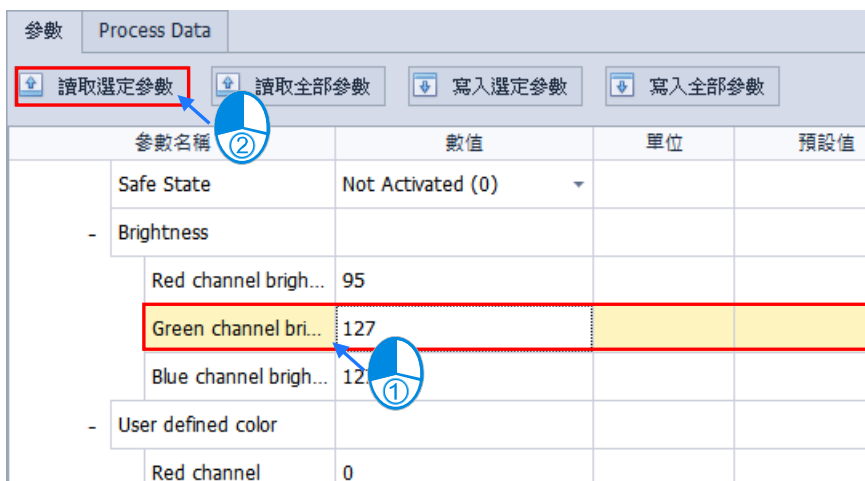
3



設定選項介紹 (若數值為灰底的參數只可以進行讀取)

(1) 讀取選定參數_針對單一項目或子集進行讀取

➢ 選定想要讀取的項目後，點選讀取選定參數



- 選定想要讀取的子集後，點選讀取選定參數 (有些單一選項無法直接讀取)

參數 Process Data			
<input type="button" value="讀取選定參數"/> <input type="button" value="讀取全部參數"/> <input type="button" value="寫入選定參數"/> <input type="button" value="寫入全部參數"/>			
參數名稱	數值	單位	預設值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing (true)		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

參數 Process Data			
<input type="button" value="讀取選定參數"/> <input type="button" value="讀取全部參數"/> <input type="button" value="寫入選定參數"/> <input type="button" value="寫入全部參數"/>			
參數名稱	數值	單位	預設值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

- (2) 讀取全部參數_針對所有項目進行讀取
直接點選讀取全部參數即可。

參數 Process Data			
<input type="button" value="讀取選定參數"/> <input type="button" value="讀取全部參數"/> <input type="button" value="寫入選定參數"/> <input type="button" value="寫入全部參數"/>			
參數名稱	數值	單位	預設值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing (true)		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

3

(3) 寫入選定參數_針對單一項目或子集進行寫入

- 選定想要寫入的項目後，點選寫入選定參數

3

The screenshot shows the 'Process Data' window with the following table:

參數名稱	數值	單位	預設值
Safe State	Not Activated (0)		
- Brightness			
Red channel bright...	95		
Green channel bri...	127		
Blue channel brig...	127		
- User defined color			
Red channel	0		

- 選定想要寫入的子集後，點選寫入選定參數 (有些單一選項無法直接讀取)

The screenshot shows the 'Process Data' window with the following table:

參數名稱	數值	單位	預設值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing (true)		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

The screenshot shows the 'Process Data' window with the following table:

參數名稱	數值	單位	預設值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

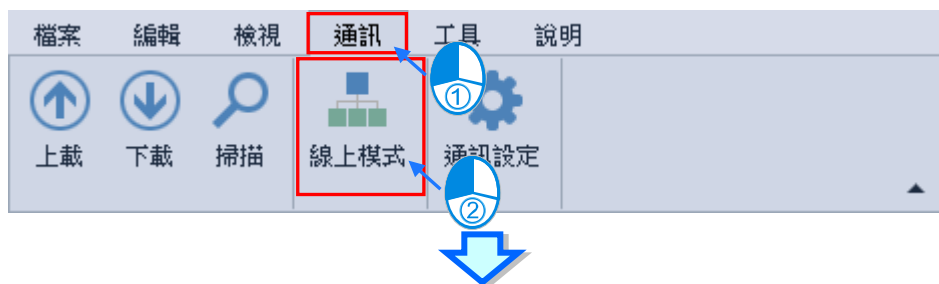
- (4) 寫入全部參數_針對所有項目進行寫入
直接點選寫入全部參數即可。

參數名稱	數值	單位	預設值
- Segment mode paramters			
- Blinking mode			
Segment 1 blinking mode	Flashing (true)		
Segment 2 blinking mode	Blinking with 50% duty cycle (false)		
Segment 3 blinking mode	Blinking with 50% duty cycle (false)		
Number of segments	Three segment (3)		Three segment (3)
- Level mode paramters			

3

➤ Process Data 參數設定

首先點選線上模式後，點選模組，在下方編輯區 IO-Link 分頁右上方的監控顯示裝置連線狀況，右下方顯示 Process Data 的參數數值。



狀態	通訊埠 ID	模式	類型	來源	實例	定義
✓	1	事件單一發生	通知	主站	應用	(0xFF21) 新連線裝置
	2	-	-	-	-	關閉
✓	3	事件單一發生	通知	主站	應用	(0xFF21) 新連線裝置
	4	-	-	-	-	關閉

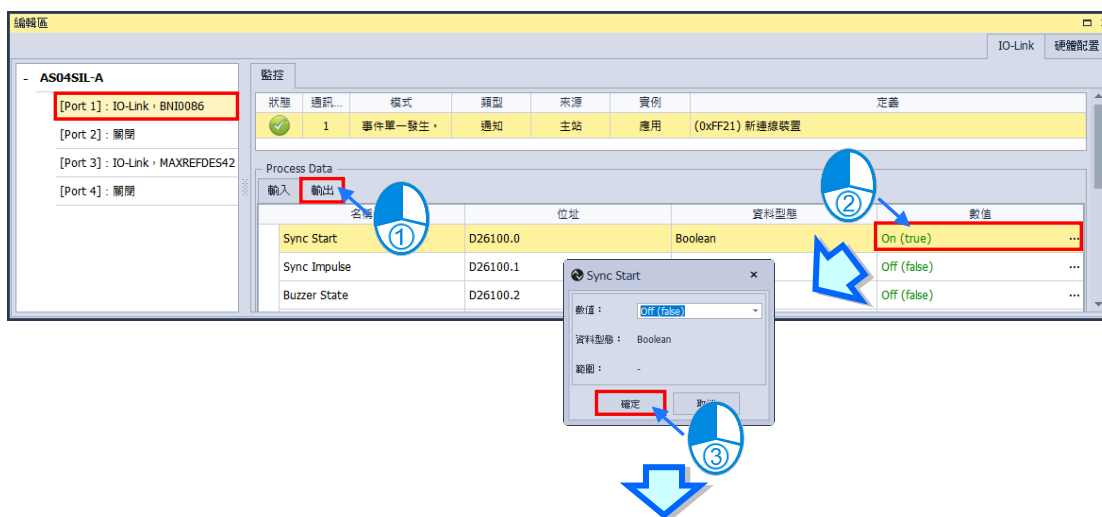
名稱	位址	資料型態	數值
Port 1			
Port 2			
Port 3	D26000 - D26001		
Ambient temperature switch...	D26000.0	Boolean	OFF (false)
Ambient temperature value	D26001.0 - D26001.15	Unsigned Integer	9068

可以從編輯區左側選擇單一裝置進行監控

3



在右下方的「Process Data」選擇輸出分頁，接著點選欲修改項目的數值欄位，會跳出設定視窗，設定完成後點選定即修改完成。



※ 在 HWConfig 的 IO-Link 中所顯示的參數值皆由電壓訊號經過轉換後具有物理意義，經由 ISPSOft 讀取裝置位址皆為電壓訊號，使用前請先自行確認。

3.1.3.18 符號設定

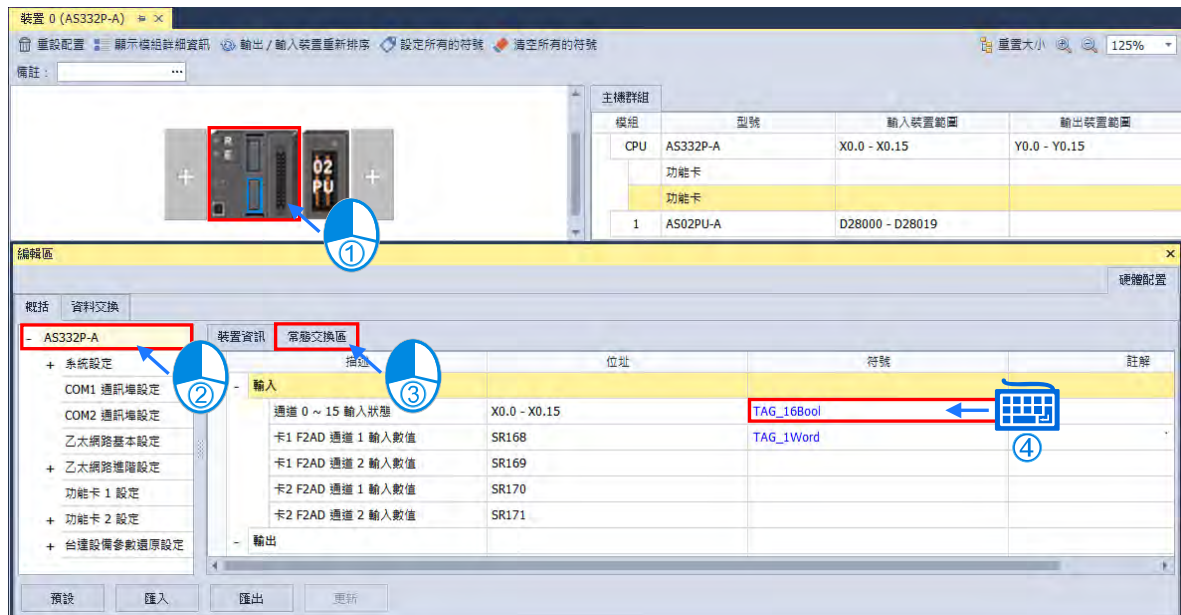
從 ISP 開啟 HWCONFIG 才支援符號設定功能，針對主機及模組的常態資料交換區 HWCONFIG 提供使用者設定變數符號，之後便可在 ISPSOFT 的全域符號中的 HWCONFIG 符號表中使用，即可在程式中存取對應的 HWCONFIG 主機及模組參數。

建立符號

首先點選主機或模組，在下方編輯區左側點選主機名稱，接著右側點選「常態交換區」，下方即可設定符號名稱及註解。

*符號命名規則：建立的符號名稱最大長度，不可超過 40 個字元，符號名稱不可使用特殊字元，例如 *、#、?、\、%、@ ... 等，且開頭不能為數字；符號名稱命名不可以重複。

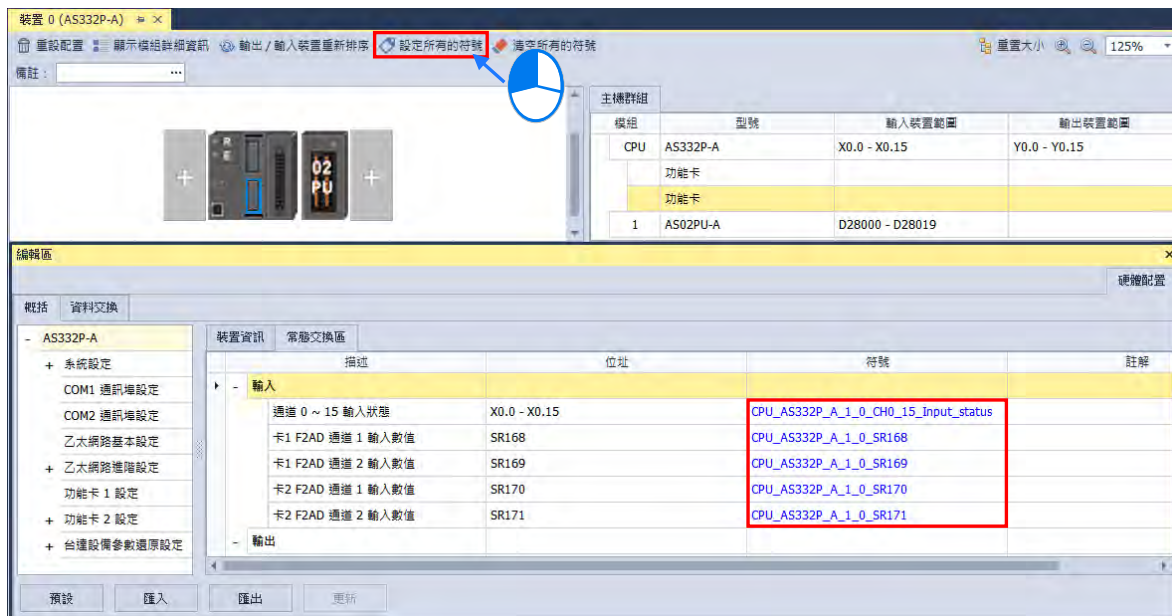
3



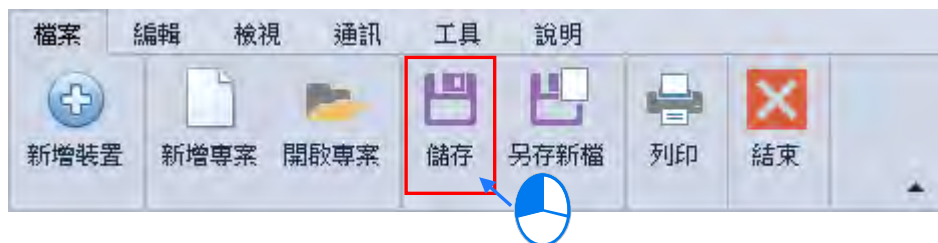
也可以點選硬體配置區上方的「設定所有的符號」，一次性的建立所有符號，也可以使用「清除所有的符號」來清空符號。

*設定所有符號命名規則：模組類型_模組型號_群組_模組順序_描述，範例：PIO_AS02PU_A_1_1_Error_code。

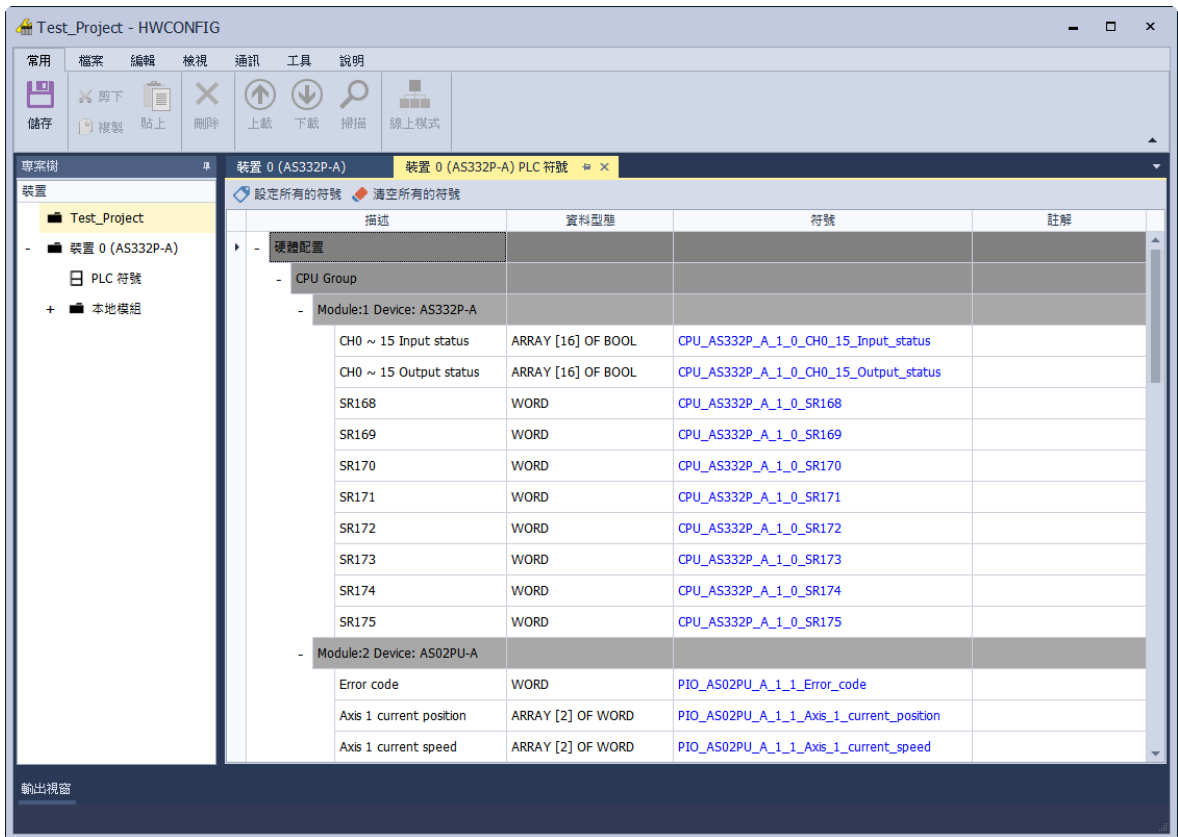
3



符號設定完成後按下功能工具列的「檔案」>「儲存」。



點選專案樹中的「PLC 符號」，會出現符號總攬，「PLC 符號」也提供編輯符號的功能，如下圖。

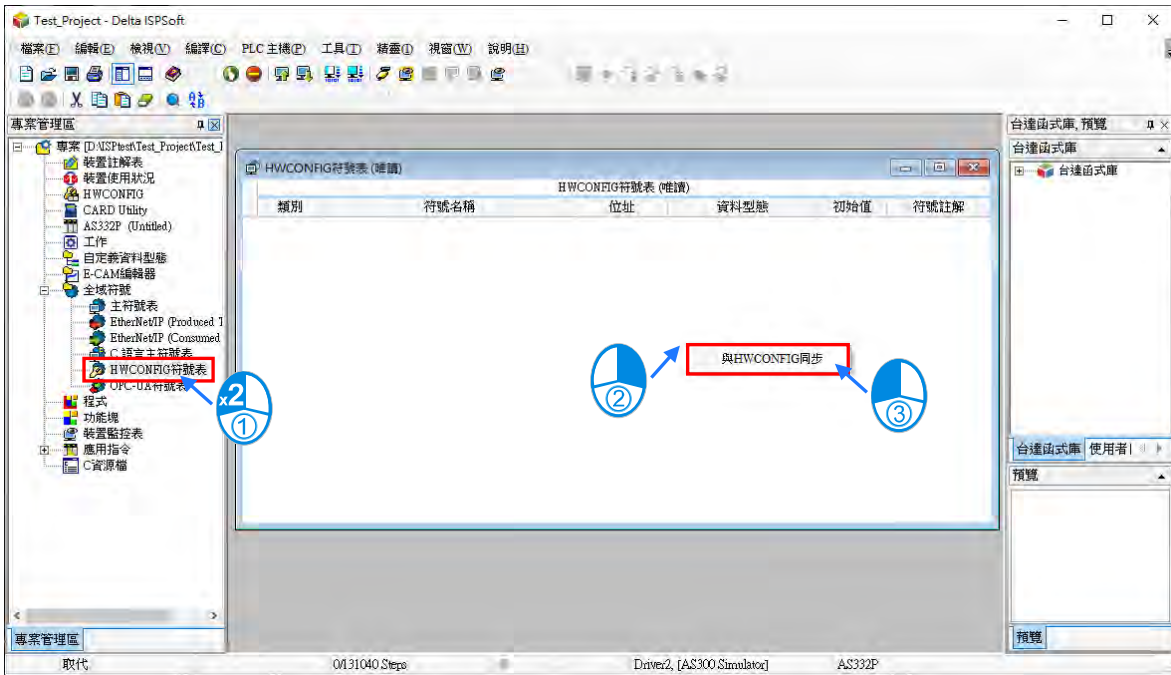


3

ISPSOft 載入 HWCONFIG 符號

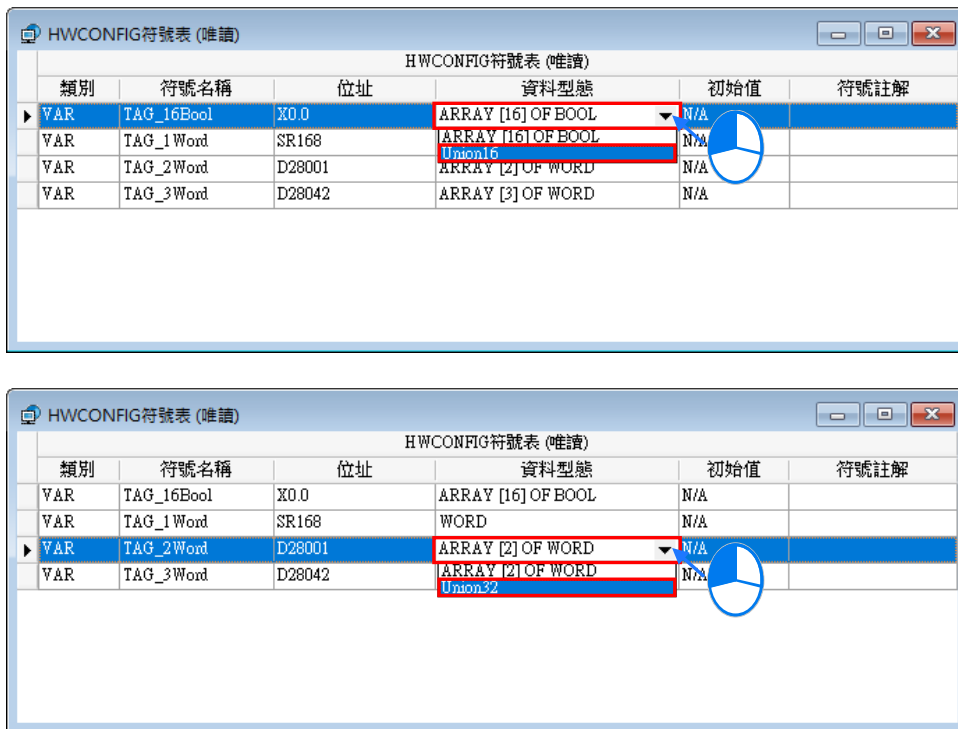
在 ISPSOft 專案樹點選「全域符號」 > 「HWCONFIG 符號表」，接著在 HWCONFIG 符號表頁面點選右鍵中的「與 HWCONFIG 同步」，就會更新 HWCONIG 符號表頁面，若關閉 HWCONFIG，也會自動完成 HWCONFIG 符號表更新，之後便可在 ISPSOft 的程式中使用此變數符號。

3



類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	TAG_1GBool	X0.0	ARRAY [16] OF BOOL	N/A	
VAR	TAG_1Word	SR168	WORD	N/A	
VAR	TAG_2Word	D28001	ARRAY [2] OF WORD	N/A	
VAR	TAG_3Word	D28042	ARRAY [3] OF WORD	N/A	

同步後的符號長度為 16 Bits (ARRAY [16] OF BOOL 或 WORD)，可以在 HWCONFIG 符號表的資料型態欄位使用下拉選單將資料型態選為 Union16 ; 32 Bits (ARRAY [2] OF WORD)，可以選為 Union32 ; 64Bits (ARRAY [4] OF WORD)，可以選為 Union64 以方便使用者以不同資料型態來使用，詳細 Union 使用方式請參考第 8.4 節。



3.1.4 PLC 主機參數設定

PLC 主機參設定頁面，裡面的項目會隨著所選取的 PLC 主機而有所不同，本章節只是初略簡介項目，詳細的參數設定及使用說明，還需參考各 PLC 主機的操作手冊。

3.1.4.1 系統設定-系統資訊

「系統資訊」頁面只有「PLC 名稱」與「註解」兩個欄位，其中「PLC 名稱」欄最多可輸入 16 字元；「註解」欄最多可輸入 32 個字元。所輸入的字元並沒有特別的制限，可輸入包括特殊字元及空白鍵，不過須注意一個中文字將會佔用兩個字元。



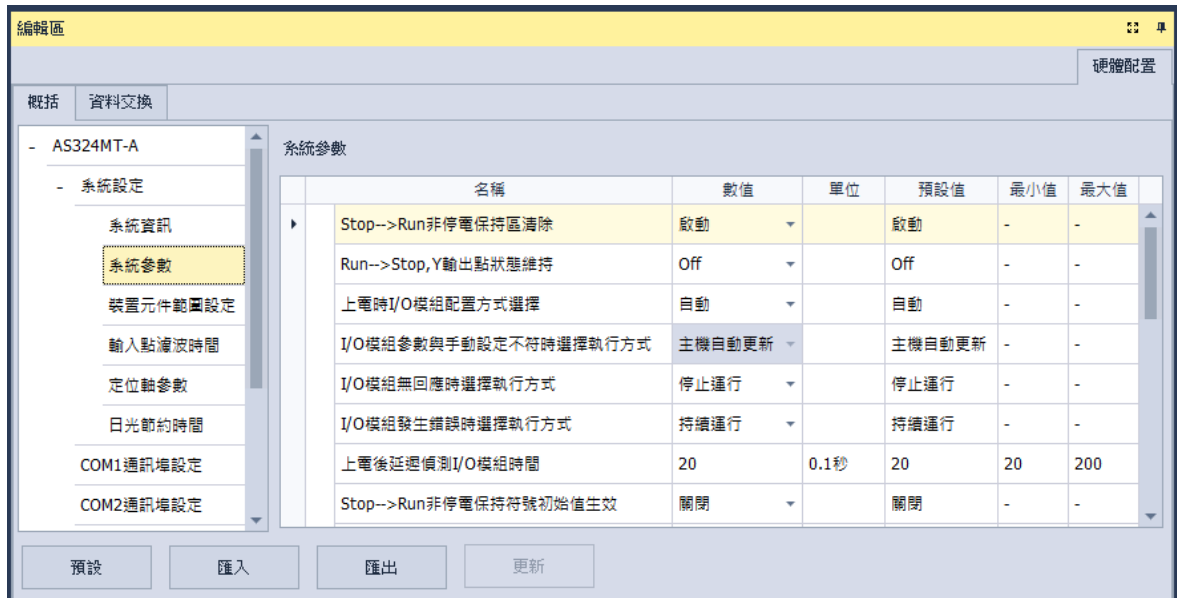
當新增一個 ISPSOft 專案時，系統便會以專案的名稱作為主機預設名稱，且會顯示於專案管理區的機種名稱後方，而後續則可於此處重新定義。



PLC 名稱的功能主要是用來做為裝置識別，尤其是在多台 PLC 進行連線操作時，藉由確認 PLC 名稱的方式便可防止使用者的誤操作；當使用者要對主機進行資料的上下載或其它的連線操作時，若發現操作對象的主機名稱與專案中的名稱不符時，系統便會提醒使用者做再次的確認，以避免對其他的主機造成非預期的影響。

3.1.4.2 系統設定-系統參數

「系統參數」頁面設定各項參數如下，可藉由下拉選單或在輸入欄位填入適當的數值。



- **Stop → Run 非停電保持區清除**

設定當 PLC 主機的狀態由 Stop 切換至 Run 時，非停電保持區的所有裝置狀態及數值是否被清除。

- 關閉：裝置狀態與數值保持現狀。
- 啟動：裝置狀態與數值清除為預設值。

- **Run → Stop 時 Y 輸出處理方式**

設定主機狀態在 RUN 與 STOP 狀態間切換時的 Y 裝置狀態。

- Off：將所有 Y 裝置的輸出都設為 Off。
- 保持：保持現狀輸出。

- **上電時 I/O 模組配置方式選擇**

設定主機上電時自動讀取模組實際配置，或由使用者在 HWCONFIG 選取手動設定的配置。

- 自動：使用者不須設定任何模組與參數，依據實際連接的模組運行。
- 手動：依據使用者由 HWCONFIG 定義。
- 手動+旗標：主機右側第 1~32 台模組啟動旗標 (V1.06.00 版 (含) 以後支援) 詳細操作方式需參考 AS 操作應用手冊。

3

● I/O 模組參數與手動設定不符時選擇執行方式

若前述上電時 I/O 模組配置方式選擇設為手動時，此項目才可設定當實際模組配置與手動設定不符時的處理

- 主機自動更新：主機依據實際模組配置更新存放在主機內部的配置資訊。
- 主機報警：主機回報錯誤。

● I/O 模組無回應時選擇執行方式

設定當連接的擴充模組其中有通訊斷線無回應的模組時，主機與其他正常的模組是否持續運行。

- 停止運行：主機停止且回報錯誤。
- 持續運行：主機與其他正常模組繼續運行。

● I/O 模組發生錯誤時選擇執行方式

設定當擴充模組出現輕微錯誤時的處理方式。

- 停止運行：主機停止且回報錯誤。
- 持續運行：主機繼續運行，但會記錄警告訊息。

● 上電後延遲偵測 I/O 模組時間

設定主機上電後，延遲多久的時間再對連線的 I/O 模組與遠端 I/O 偵測。

● Stop → Run 非停電保持符號初始值生效

選取啟動時，則 PLC 由停止狀態變為運行狀態時，非停電保持符號的值會恢復為初始值。

● Stop → Run 停電保持符號初始值生效

選取啟動時，則 PLC 由停止狀態變為運行狀態時，停電保持符號的值會恢復為初始值。

● 指定主機內建 X 輸入點控制 Run/Stop

設定是否由某個 X 輸入點控制主機的 Run/Stop 狀態。

- 關閉：功能關閉，狀態由主機上的指撥開關切換。
- 啟動：使用指定的 X 輸入點控制，但主機上的指撥開關操作仍有效。

● 選擇主機內建 X 輸入點

若前述指定主機內建 X 輸入點控制 Run/Stop 設為啟動時，由下拉選單選擇一個 X 輸入點作為 Run/Stop 切換用。



● 啟動固定掃描週期時間

設定主機在最短在固定的時間完成一次掃描週期。

- 關閉：功能關閉。
- 啟動：則當實際的程序掃描時間小於設定值時，主機便會等待時間到達設定值，才會開始下一次的掃描；而若實際的掃描時間大於此設定值，則主機便不理會此設定值，而根據實際的掃描時間進行運作。

● 輸入固定時間

若前述啟動固定掃描週期時間設為啟動時，在此欄位輸入設定的週期時間。須注意若設定的掃描時間大於設定掃描逾時時間的設定值時，一旦主機開始運行後，會發生逾時錯誤。

● 設定掃描逾時時間

設定的掃描逾時時間，若程式執行超過此時間，主機會回報錯誤。

● 顯示電池低電壓警報

當紀錄時間的鋰電池處於低電壓狀態或未安裝時，可以設定是否顯示警報。

- 關閉：功能關閉。
- 啟動：鋰電池處於低電壓狀態或未安裝時，顯示警報。

● 錯誤歷史資料存放位置

設定選擇錯誤記錄的存放位置。

- PLC：存放於 PLC。存滿二十筆資料後，舊的紀錄會被新的紀錄覆蓋。
- PLC & SD Card：存放於 PLC 與 SD 卡。PLC 存滿二十筆資料後，會被搬移至 SD 卡存放。

● COM 口通訊錯誤記錄

設定 COM 口發生錯誤時是否開啟錯誤記錄。

- 關閉：功能關閉。
- 啟動：啟動記錄功能。

● 24Vdc 電源不穩定處理機制

可選擇 24Vdc 電源發生跳電狀況的處理機制。

- 穩定後持續運行：當 24Vdc 供應電源發生不穩定狀況時，自動等待電源恢復穩定之後，再持續運行。
- 進入快閃錯誤燈狀態：當 24Vdc 供應電源發生不穩定狀況時，強制 PLC 進入快閃 ERROR 燈，並且 STOP 的狀態。

- **通訊模組刷新機制**

- 掃描速度優先：自動分批次更新資料異動的部分，以不增加太多掃描時間為優先。
- 資料同步優先：每一次掃描周期更新所有資料異動的部分，依資料內容同步為優先。

- **AS 遠端模組刷新機制**

- PLC 程式掃描優先：使用 PLC 程式掃描時間為優先。
- 單台遠端同步優先：使用單台遠端同步為優先。
- 全部遠端同步優先：使用全部遠端同步為優先。

3

- **I601 時間中斷設定**

設定 I601 時間中斷的時間，搭配工作管理中的定時中斷 0 使用。

- **I602 時間中斷設定**

設定 I602 時間中斷的時間，搭配工作管理中的定時中斷 1 使用。

- **I603 時間中斷設定**

設定 I603 時間中斷的時間，搭配工作管理中的定時中斷 2 使用。

- **I604 時間中斷設定**

設定 I604 時間中斷的時間，搭配工作管理中的定時中斷 3 使用。

3.1.4.3 系統設定-裝置元件範圍設定

「裝置元件範圍設定」頁面設定各項參數如下，供使用者設定各種裝置的停電保持區範圍及副程式元件範圍。



編輯區

硬體配置

AS324MT-A

系統設定

裝置元件範圍設定

名稱	數值	單位	預設值	最小值	最大值
停電保持D元件起始位址	20000	...	20000	0	29999
停電保持D元件結束位址	23999		23999	0	29999
停電保持16位元C元件起始位址	448	...	448	0	511
停電保持16位元C元件結束位址	511		511	0	511
停電保持32位元C元件起始位址	128	...	128	0	255
停電保持32位元C元件結束位址	255		255	0	255
停電保持M元件起始位址	6000	...	6000	0	8191
停電保持M元件結束位址	8191		8191	0	8191
停電保持S元件起始位址	512	...	512	0	2047
停電保持S元件結束位址	1023		1023	0	2047
副程式型T元件起始位址	412	...	412	0	511
副程式型T元件結束位址	511		511	0	511

預設 匯入 匯出 更新

3

按下 **...** 開啟設定視窗，可設定裝置的起始與結束位址。按下「預設值」按鈕恢復預設值，按下「清除」按鈕將欄位清除為 0，完成設定後按下「確定」以儲存設定；或不儲存設定按下「取消」離開頁面。



參數設置

起始位址: 20000

結束地址: 23999

預設值 清除

確定 取消

3.1.4.4 系統設定-輸入點濾波時間

「輸入點濾波時間」頁面用於設定主機上每個輸入點的濾波時間，若輸入點接收到的訊號狀態持續時間小於濾波時間，則會被視為雜訊將其過濾，使用者可視現場應用環境的干擾狀況設定適當的濾波時間。

3

名稱	數值	單位	預設值	最小值	最大值
主機X輸入點濾波方式選擇	預設時間		預設時間	-	-
X0.0輸入點濾波時間	1	微秒	1	0	20000
X0.1輸入點濾波時間	1	微秒	1	0	20000
X0.2輸入點濾波時間	1	微秒	1	0	20000
X0.3輸入點濾波時間	1	微秒	1	0	20000
X0.4輸入點濾波時間	1	微秒	1	0	20000
X0.5輸入點濾波時間	1	微秒	1	0	20000
X0.6輸入點濾波時間	1	微秒	1	0	20000

➤ 主機 X 輸入點濾波方式選擇

設定是否設定各輸入點的濾波時間或使用預設值

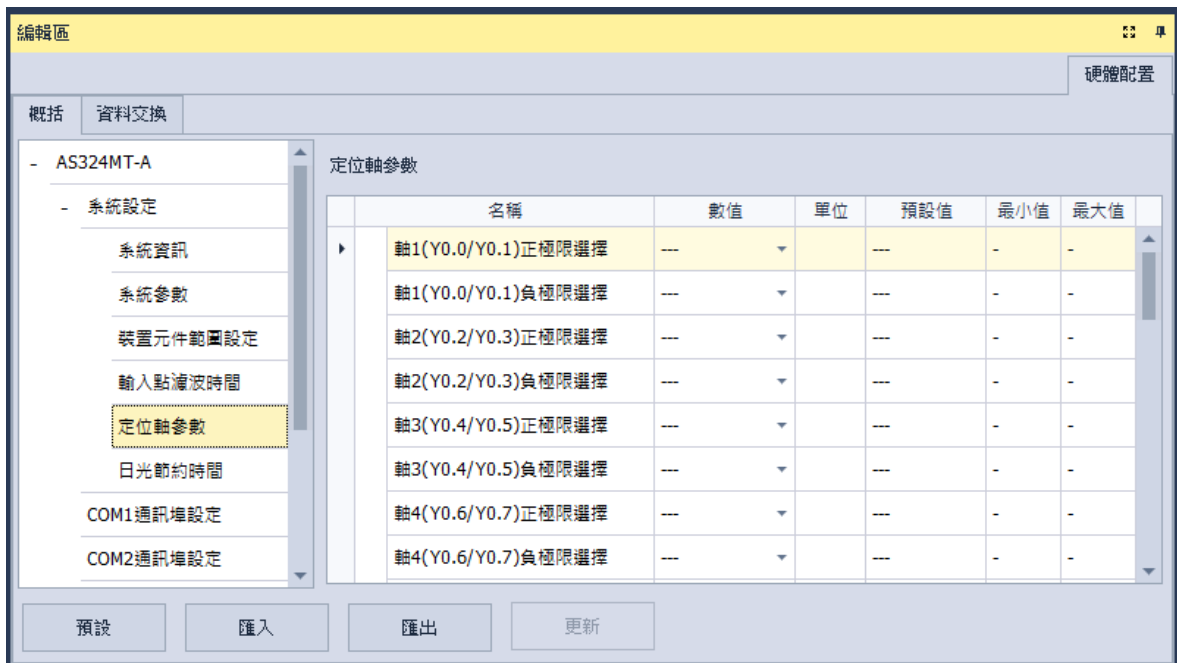
- 預設時間：使用預設的濾波時間。
- 自訂時間：由使用者自行設定每個 X 輸入點的濾波時間。

➤ X0.0~X0.15 輸入點濾波時間

若前述主機 X 輸入點濾波方式選擇設為自訂時間時，可於此輸入 X0.0~X0.15 各自的濾波時間。

3.1.4.5 系統設定-定位軸參數

「定位軸參數」頁面用於設定指定輸入點作為軸 1~軸 6 各通道的正極限與負極限，可設定共 12 個極限點；並可設定指定輸入點作為軸 1~軸 6 的 Z 相觸發點、原點復歸後清除輸出點選擇；以及對正、負向軟體極限位置做設定。



➤ **軸 1 (Y0.0/Y0.1) 正、負極限選擇 ~ 軸 6 (Y0.10/Y0.11) 正、負極限選擇**

由下拉選單選擇上緣或下緣觸發及輸入的 X 接點。

➤ **軸 1 (Y0.0/Y0.1) Z 相觸發選擇 ~ 軸 6 (Y0.10/Y0.11) Z 相觸發選擇**

由下拉選單選擇上緣或下緣觸發及輸入的 X 接點。

➤ **軸 1 (Y0.0/Y0.1) 原點復歸後清除輸出點選擇 ~ 軸 6 (Y0.10/Y0.11) 原點復歸後清除輸出點選擇**

由下拉選單選擇輸出接點。

➤ **軸 1 (Y0.0/Y0.1) 正、負向軟體極限位置設定 ~ 軸 6 (Y0.10/Y0.11) 正、負向軟體極限位置設定**

可於欄位設定使用者需求的脈波數當成軸 1 ~ 軸 6 的正、負向軟體極限位置，設定範圍為-2147483648 ~ 2147483647。

3

3.1.4.6 系統設定-日光節約時間

「日光節約時間」頁面用於開啟或關閉日光節約時間功能及設定進行日光節約時間的日期。開啟日光節約時間時，設定的時間區段內時鐘將自動提早設定的節約時間。



The screenshot shows the configuration page for '日光節約時間' (Daylight Saving Time) under the '系統設定' (System Settings) section for device 'AS324MT-A'. The table below represents the data shown in the interface:

名稱	數值	單位	預設值	最小值	最大值
日光節約時間關閉/開啟	關閉		關閉	-	-
開始日期-月	4		4	3	5
開始日期-日	1		1	1	31
結束日期-月	10		10	9	11
結束日期-日	1		1	1	31
節約時間	60	分鐘	60	1	180

➤ 日光節約時間關閉/開啟

設定關閉或開啟日光節約時間。

➤ 開始日期-月

設定開始進行日光節約時間的月份。

➤ 開始日期-日

設定開始進行日光節約時間的日期。

➤ 結束日期-月

設定結束進行日光節約時間的月份。

➤ 結束日期-日

設定結束進行日光節約時間的日期。

➤ 節約時間

設定調整提早的日光節時間，單位為分鐘。

3.1.4.7 COM 通訊埠設定

「COM 通訊埠設定」在各系列機種依據型號不同有具備 1 或 2 個 COM 埠，點選項目列表中的 **COM1 通訊埠設定** 或 **COM2 通訊埠設定**，可分別設定 COM1 與 COM2 參數，兩 COM 埠設定項目皆相同。

名稱	數值	單位	預設值	最小值	最大值
COM1 站號	1		1	1	254
通訊協定設定時機	Stop --> Run		Stop --> Run	-	-
通訊速率選擇	9600	bps	9600	-	-
自訂通訊速率	96	0.1kbps	96	1	5000
資料位元	7	bit	7	-	-
同位元	偶同位		偶同位	-	-
停止位元	1	bit	1	-	-
MODBUS 通訊模式	ASCII		ASCII	-	-
回應通訊前等待時間	0	毫秒	0	0	3000
接收等待逾時時間	200	毫秒	200	0	3000
COM1 燈號顯示	COM1		COM1	-	-

- ❶ 設定此通訊埠的通訊站號，做為通訊上的裝置識別，且連接在同一條線路上的站號不允許重複。
- ❷ 設定此通訊埠的設定生效的時機，選擇 **Stop → Run** 為主機狀態由 Stop 切換為 Run 時生效，選擇**上電時**則為主機上電時生效。
- ❸ 設定此通信埠所採用的通訊速率 (Baud Rate)。可由下拉選單選擇預先定義的通訊速率之一，或選擇**自定義**，輸入自訂通訊速率。
- ❹ 設定通訊相關設定。
- ❺ 設定**回應通訊前等待時間**，當 AS 主機接收到通訊命令後等待設定的時間再進行回覆，配合遠端設備模式切換的時間。設定**接收等待逾時時間**，當 AS 主機作為主站時送出通訊命令，等待設定的時間後未收到回覆則視為逾時。
- ❻ 設定 COM1 燈號顯示，若下拉選單選擇為 COM1，則通訊燈號顯示的為 COM1 的 RS485 通訊狀態，若選擇為 Card1，則通訊燈號顯示的為擴充卡 CARD1 的通訊狀態。
設定 COM2 燈號顯示，若下拉選單選擇為 COM2，則通訊燈號顯示的為 COM2 的 RS485 通訊狀態，若選擇為 Card2，則通訊燈號顯示的為擴充卡 CARD2 的通訊狀態。

3.1.4.8 乙太網路基本設定

於項目列表中點選「乙太網路基本設定」項目即可進入。此處主要用於設定主機乙太網路 (Ethernet) 埠的通訊參數，關於透過 Ethernet 來與主機進行連線。



編輯區

硬體配置

概括 資料交換

- AS324MT-A

- + 系統設定
- COM1通訊埠設定
- COM2通訊埠設定
- 乙太網路基本設定
- + 乙太網路進階設定
- 功能卡1設定
- + 功能卡2設定
- + 台達設備參數還原設定

乙太網路基本設定

名稱	數值	單位	預設值	最小值	最大值
IP位址	192.168.1.5		192.168.1.5	1.1.1.1	223.255.255.255
子網路遮罩	255.255.255.0		255.255.25...	0.0.0.0	255.255.255.255
網道位址	192.168.1.1		192.168.1.1	1.1.1.1	223.255.255.255
TCP連線保持時間	30	秒	30	1	65535
IP定址模式	靜態		靜態	-	-

預設 匯入 匯出 更新

設定主機 IP 的定址模式。若設為「靜態」，代表 IP 位址等參數將由使用者直接指定，但若設為「DHCP」或「BOOTP」，則代表 IP 位址將經由 DHCP/BOOTP 來動態分配。

3.1.4.9 乙太網進階設定-IP 過濾器

「IP 過濾器」頁面提供使用者設定網路裝置的過濾功能，透過此設定便可多加一層の確認機制，以確保通訊的對象為我們所允許的裝置。此功能可用以限定通訊對象。當此功能被啟用後，只有 IP 位址列於清單範圍中的裝置才允許跟主機進行通訊。此功能的設定步驟如下，而清單中最多可設定 8 組的位址範圍。

名稱	數值	單位	預設值	最小值	最大值
啟動 IP 位址過濾設定	<input type="checkbox"/>		<input type="checkbox"/>	-	-
1st 起始 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
1st 結束 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
2nd 起始 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
2nd 結束 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
3rd 起始 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
3rd 結束 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
4th 起始 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
4th 結束 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
5th 起始 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255
5th 結束 IP 位址	0.0.0.0		0.0.0.0	0.0.0.0	223.255.255.255

要啟動 IP 過濾器功能，請勾選啟動 IP 位址過濾設定，便可輸入以下 1~8 組的 IP 過濾範圍的起始與結束位址。

3.1.4.10 乙太網進階設定-網路時間校正

「網路時間校正」讓主機可以透過 NTP 伺服器來進行時間校正的功能，並可於此處進行相關設定。

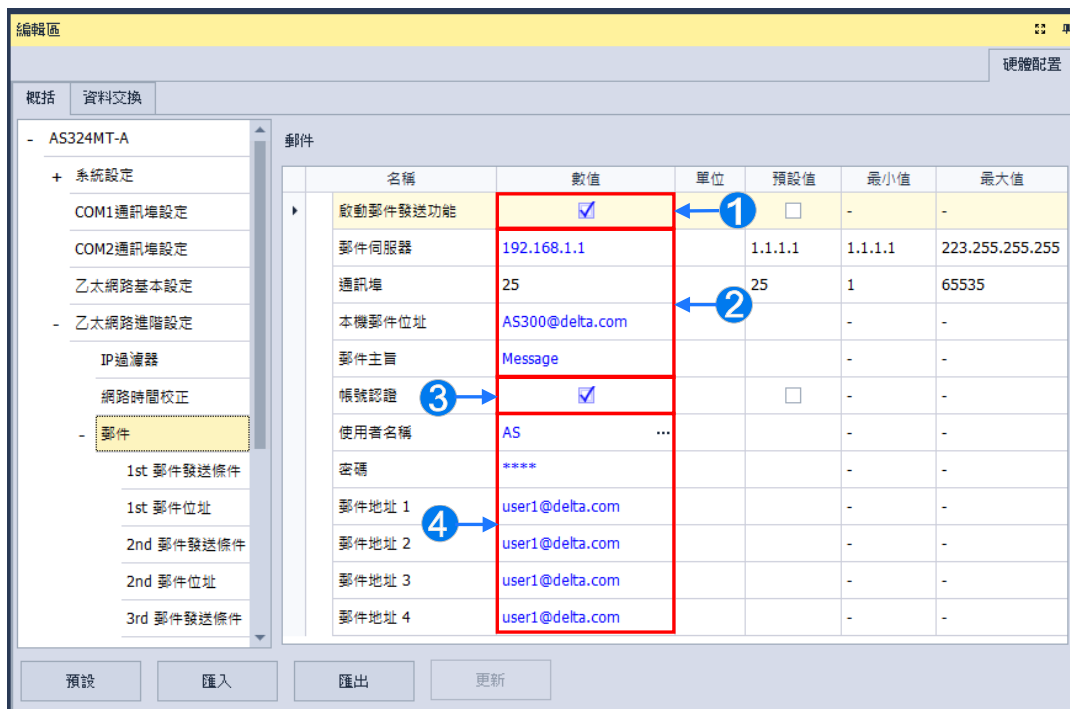
3

名稱	數值	單位	預設值	最小值	最大值
啟動 NTP 客戶端服務	<input checked="" type="checkbox"/>		<input type="checkbox"/>	-	-
NTP 伺服器	1.1.1.1		1.1.1.1	1.1.1.1	223.255.255.255
更新週期	30	分	30	1	1440
時區	(GMT-12:00) 國際換日線以西		(GMT-12:00...	-	-

- 1 勾選**啟動 NTP 客戶端服務**開啟功能。
- 2 設定 **NTP 伺服器**的位址與定期更新的**更新週期**。以上圖為例，主機將會每隔 30 分鐘便連線至 NTP 伺服器。
- 3 設定採用時間校正的**時區**。

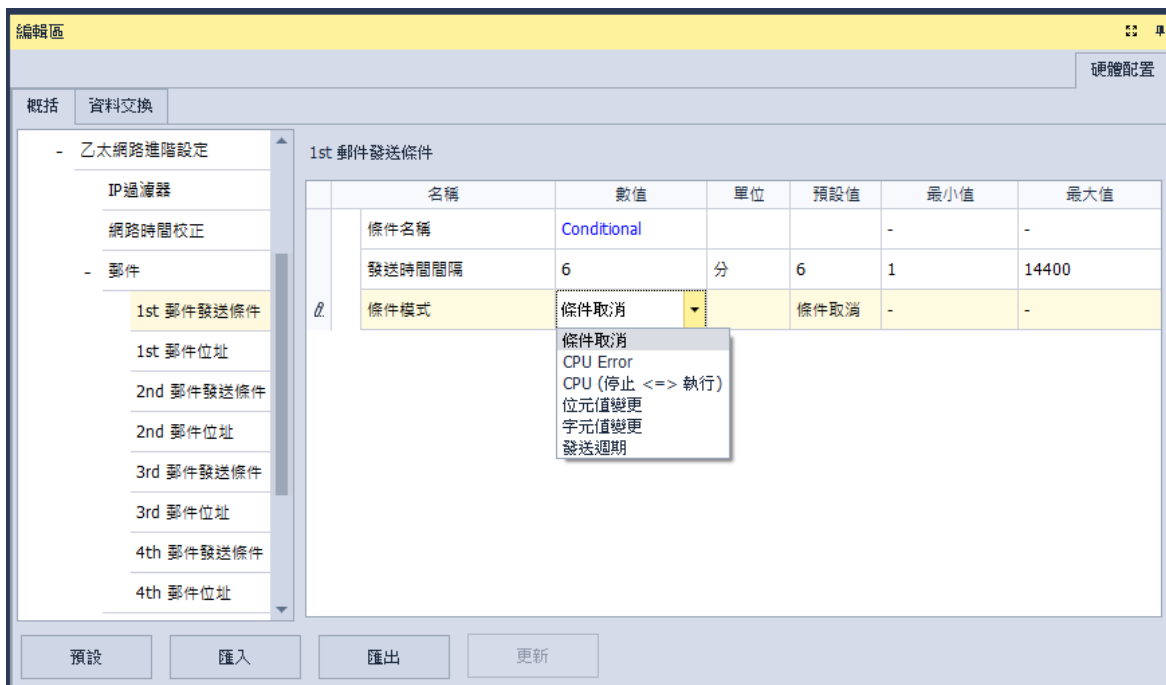
3.1.4.11 乙太網進階設定-郵件

「郵件」頁面提供使用者設定郵件相關功能。啟用當觸發設定的條件模式時，將寄送郵件至設定的郵件地址的相關設定。共可設定四組郵件發送條件與四組郵件位址。



- ❶ 勾選**啟動郵件發送功能**開啟功能，便可設定下方欄位。
- ❷ 由**郵件伺服器**設定**外送郵件伺服器**的 IP 位址，在**通訊埠**輸入**外送郵件伺服器**的通訊埠，於**本機郵件位址**設定發送電子郵件時所使用的寄件信箱，輸入**郵件主旨**作為每封郵件的主旨開頭。
- ❸ 勾選**帳號認證**開啟功能，依據外送郵件伺服器的狀況，於此處設定登入前是否須先驗證帳號及密碼。
- ❹ 輸入要傳送的目標**郵件地址**。

在「郵件發送條件」項目輸入「條件名稱」與「發送時間間隔」，再於「條件模式」的下拉選單選擇郵件發送的觸發條件。當發送條件持續成立時，系統便會每隔一段發送時間間隔就發出一封電子郵件；而在時間間隔內，針對同樣的發送條件，郵件則不會再重複發送。



「條件模式」可設為以下

➤ CPU Error


當主機發生錯誤時，發送條件便會成立，而關於主機錯誤狀態的相關說明，請參考各機種的操作手冊。當選擇此條件時，請再於右側的下拉選單中挑選錯誤的類別。

- (a) **致命錯誤**：當主機發生被定義為**致命錯誤**的狀況時才會觸發郵件發送。
- (b) **所有錯誤**：只要主機有錯誤產生時，郵件發送條件就會成立。

➤ CPU (停止 <=> 執行)

當主機的運行狀態由 STOP 切換至 RUN，或由 RUN 切換至 STOP 時，發送條件便會成立。


➤ 位元值變更

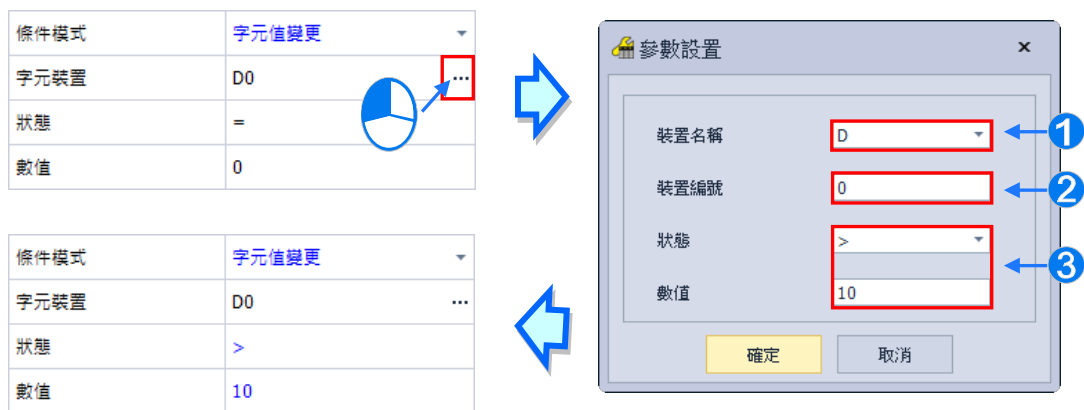
當指定裝置的位元狀態變化符合設定條件時便會觸發郵件發送。要設定條件時，按下表格右方  圖示即可開啟視窗進行設定。以下圖為例，當 X1.0 由 OFF 變為 ON 時，郵件發送條件就會成立。



- ❶ 裝置名稱：於此下拉選擇裝置的類別。
- ❷ 裝置編號與位元：於此輸入裝置的位址，且當所選裝置為 X/Y 時須一併指定位元編號。
- ❸ 狀態：於此下拉選擇觸發的條件為正緣觸發或負緣觸發。

➤ 字元值變更

當指定裝置的內容數值符合設定條件時便會觸發郵件發送。按下表格右方  圖示即可開啟視窗進行設定。以下圖為例，當 D0 大於 10 時，郵件發送條件就會成立。

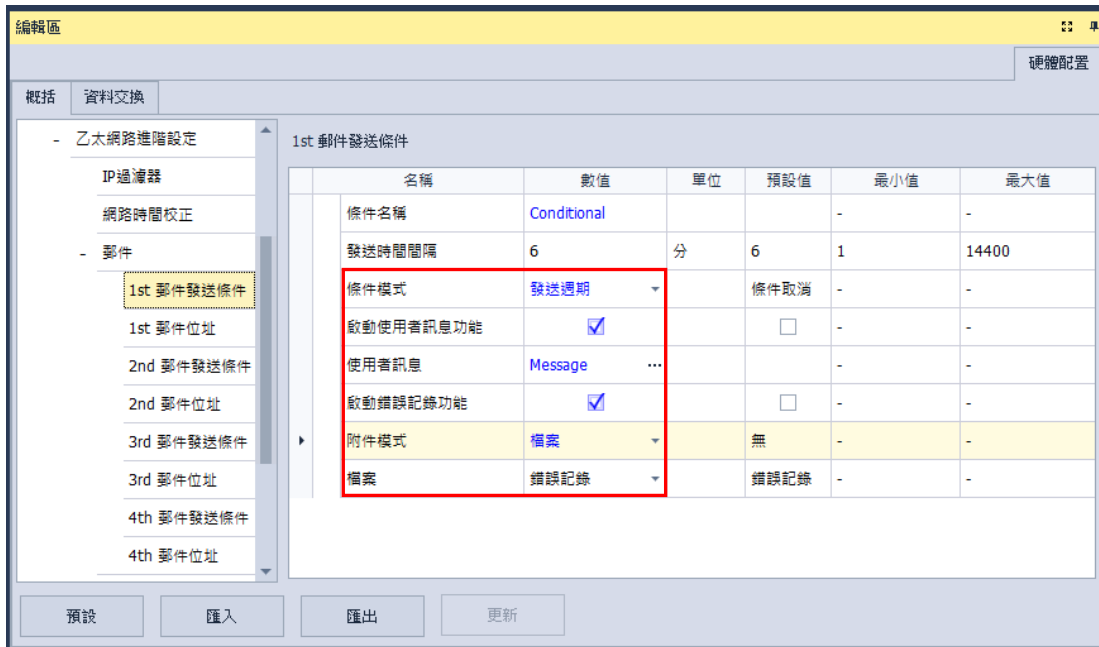


- ❶ 裝置名稱：於此下拉選擇裝置的類別。
- ❷ 裝置編號：於此輸入裝置的位址。
- ❸ 狀態與數值：於此設定觸發的條件。

➤ 發送週期

定期發出電子郵件。而發送時間則根據上方「發送時間間隔」來決定。

當有設定任何條件模式時，會出現使用者訊息與錯誤記錄的相關設定。



勾選**啟動使用者訊息功能**再按下**使用者訊息**右方 **...** 圖示，便可於彈出視窗中輸入要做為郵件本文的內容；若勾選**啟動錯誤記錄功能**，便可於郵件內容中自動添加錯誤記錄。

接著於「**附件模式**」下拉選單選擇是否要於郵件中加入附件。而要設定附件時，請務必確認各機種所允許的最大郵件檔案大小，相關資訊請參考各機種的操作手冊。




(1) 無

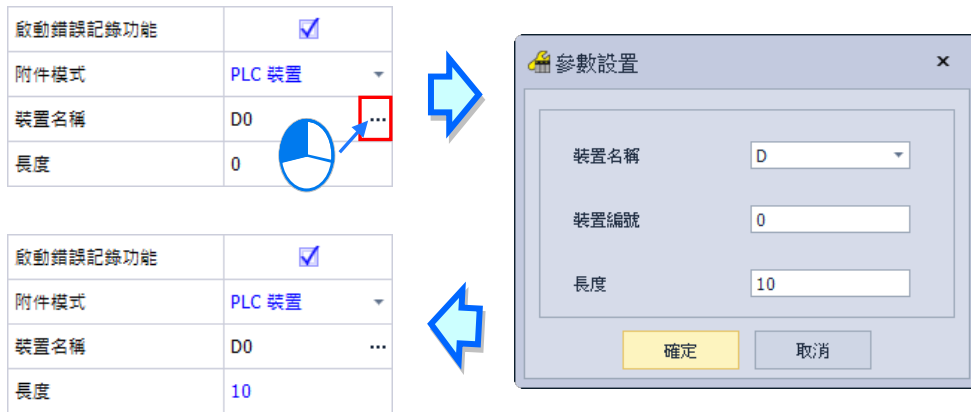
選擇此項代表無需附加檔案。

(2) 檔案

可選擇記憶卡中的**錯誤記錄**或**系統備份檔**做為郵件附件。

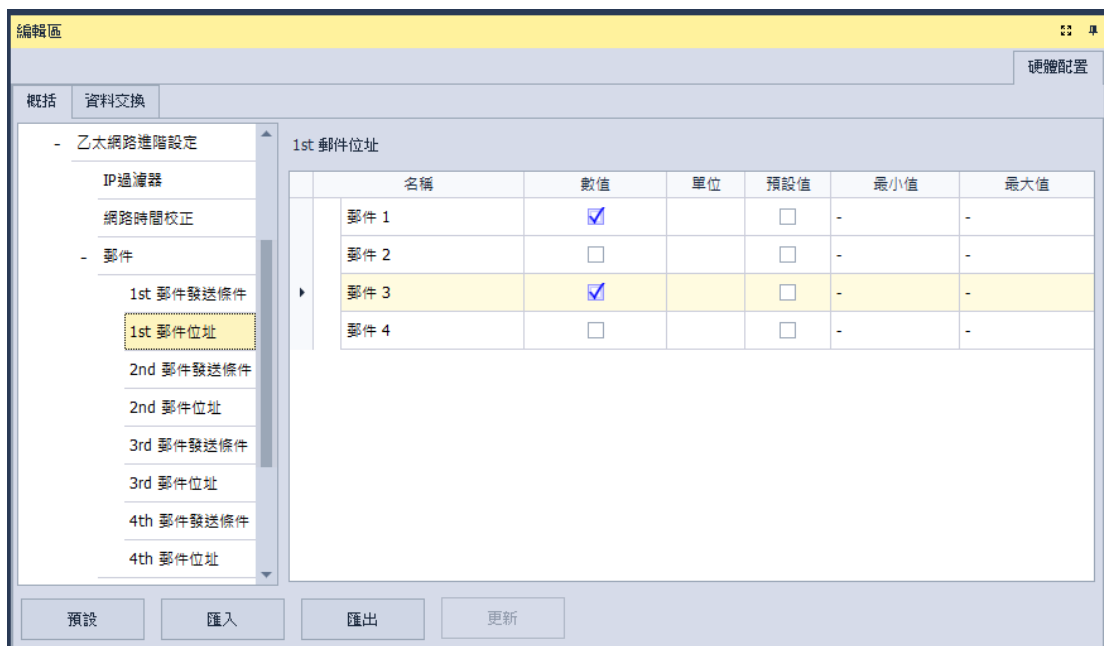
(3) PLC 裝置

若選擇此項，則當發送電子郵件時，系統會主動擷取下方設定範圍中的裝置狀態或數值來做為附加檔案；而在選擇此項之後，於下方的裝置名稱按下右方  圖示，便會出現設定視窗。以下圖為例，當此組條件成立時，裝置 D0~D9 的內容值便會被當作附件而一併發送出去。



3

在郵件位址項目，勾選當郵件發送條件達成時要發送的目標郵件地址。郵件地址在郵件項目中設定。



3.1.4.12 乙太網進階設定-通訊接口

「通訊接口」頁面中，主要在於設定透過乙太網路來進行資料傳遞的通訊接口參數；不過此功能仍需搭配特定的 API 應用指令才可進行操作。

主機與其他主機之間，經由通訊接口 (Socket) 的建立來進行資料傳遞，且同時支援 TCP 與 UDP 兩種傳輸協定。TCP 與 UDP 協定的參數幾乎相同，差別只在於 UDP 沒有「連線保持時間」的參數，參數說明如下。

3

名稱	數值	單位	預設值	最小值	最大值
遠端 IP 位址	192.168.1.1		0.0.0.0	0.0.0.0	223.255.255.255
遠端通訊埠	65500		0	0	65535
本機通訊埠	65501		0	0	65535
傳送資料位址	D0	...	D0	-	-
傳送資料長度	10		0	0	512
接收資料位址	D100	...	D0	-	-
接收資料長度	20		0	0	512
連線保持時間	60	秒	30	0	65535

名稱	數值	單位	預設值	最小值	最大值
遠端 IP 位址	192.168.1.1		0.0.0.0	0.0.0.0	223.255.255.255
遠端通訊埠	65500		0	0	65535
本機通訊埠	65501		0	0	65535
傳送資料位址	D0	...	D0	-	-
傳送資料長度	10		0	0	512
接收資料位址	D100	...	D0	-	-
接收資料長度	20		0	0	512

- **遠端 IP 位址**：設定遠端裝置的 IP 位址。
- **遠端通訊埠**：設定在這組連線中，遠端裝置所使用的通訊埠編號，輸入範圍為 0 ~ 65535。
- **本機通訊埠**：設定在這組連線中，本地主機所使用的通訊埠編號，輸入範圍為 0 ~ 65535。
- **傳送資料位址**：設定在本地主機中，存放傳送資料的啟始位址。
- **傳送資料長度**：設定本地主機要傳送的資料長度，輸入範圍為 0 ~ 512。
- **接收資料位址**：設定在本地主機中，存放接收資料的啟始位址。
- **接收資料長度**：設定本地主機要接收的資料長度，輸入範圍為 0 ~ 512。
- **連線保持時間**：設定連線的最大閒置時間。當建立的連線超過此時間無資料傳輸時，主機便會自動中斷連線。

設定時，本機與遠端的通訊埠編號不可相同，且傳送與接收的裝置位址範圍不可重疊。而以上圖為例，遠端裝置的 IP 位址為 192.168.1.1，且開啟一個編號為 65500 的通訊接口，而本地主機也開啟一個編號為 65501 的通訊接口，兩者之間便將透過這組連線通道來進行資料的傳遞。

當本地主機要傳送資料給遠端裝置時，本地主機會先將資料存放在 D0~D9 共 10 個 WORD 的空間暫存之後，再將資料傳送出去；而當接收到來自遠端裝置的資料時，主機則會將接收的資料存放在裝置 D100 ~ D119 共 20 個 WORD 的空間中。

若收到的資料量大於所設定的長度時，主機會根據設定的長度，將最前面的 20 個資料存放在 D100 ~ D119 中，超出的部份則會直接將其丟棄；同樣的，若收到的資料量小於設定長度時，主機則會從設定的啟始位置 D100 依序存放資料，而未被存放新資料的位置則會保持原本的內容值。

當 TCP 通道連線已超過 60 秒未有任何的資料傳輸時，主機便會主動將主機端的通訊接口 (Socket) 關閉，並結束通道連線。

3.1.4.13 乙太網進階設定- RTU 對應

「RTU」對應頁面用於設定 PLC 主機與台達 RTU-EN01 從站，可經由乙太網路操作遠端設備。關於 RTU-EN01 的操作與設定，請參考該產品手冊。

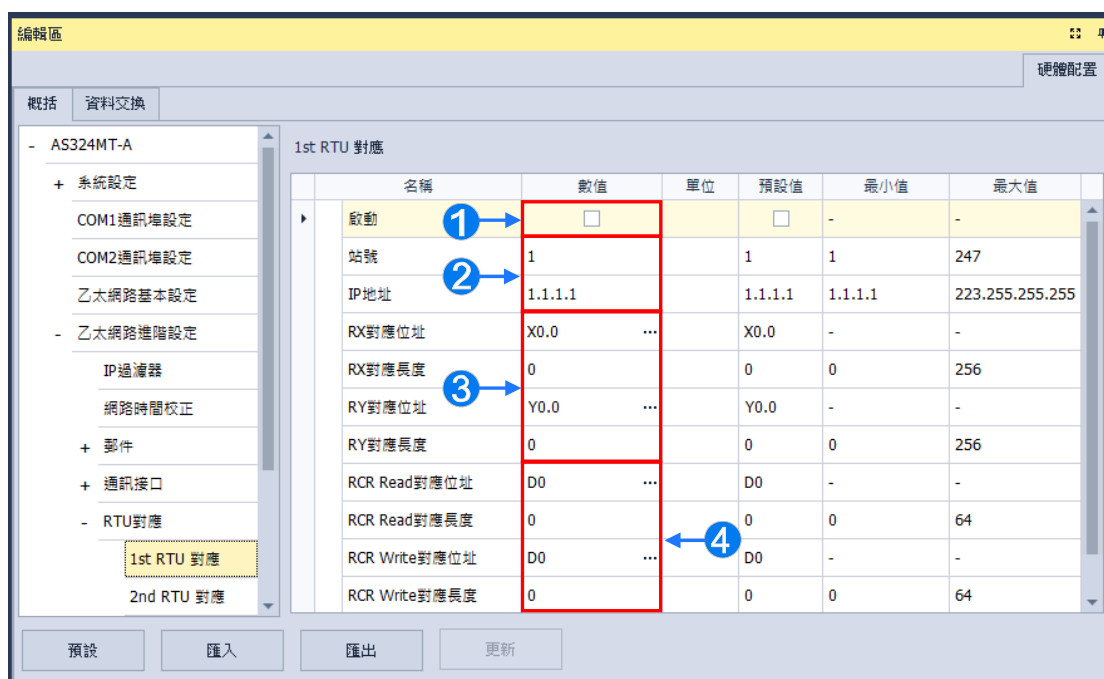
The screenshot shows a software configuration window titled '編輯區' (Edit Area) with a '硬體配置' (Hardware Configuration) tab. The '資料交換' (Data Exchange) sub-tab is active, showing '乙太網路進階設定' (Ethernet Advanced Settings) on the left and 'RTU 對應' (RTU Correspondence) on the right. The 'RTU 對應' section contains a table with the following data:

名稱	數值	單位	預設值	最小值	最大值
啟動Remote I/O對應	<input type="checkbox"/>		<input type="checkbox"/>	-	-
更新週期	10	毫秒	10	1	60000
通訊逾時	1000	毫秒	1000	1	60000

At the bottom of the window, there are buttons for '預設' (Default), '匯入' (Import), '匯出' (Export), and '更新' (Update).

勾選**啟動 Remote I/O 對應**，開啟 RTU 模組與 PLC 主機的連線功能，設定**更新週期**，對 RTU 依此時間週期性更新。設定**通訊逾時**，當 RTU 超過此時間未回應則為逾時。

RTU 對應頁面用於設定每組 RTU-EN01 與 PLC 主機的資料對應。而 RTU-EN01 與其安裝的 I/O 模組需透過 DCISoft 設定，關於 DCISoft 請參考其操作手冊。



- ❶ 勾選**啟動**開啟此站 RTU 功能，各站可單獨設定。
- ❷ 設定**站號**與 **IP 地址**。
- ❸ 設定 **RX 對應位址**與 **RX 對應長度**，用於將連接於 RTU 的 DI 模組上的數位輸入點，對應到 X/M 裝置與長度。而設定 **RY 對應位址**與 **RY 對應長度**，用於將連接於 RTU 的 DO 模組上的數位輸出點，對應到 Y/M 裝置與長度。
- ❹ 設定 **RCR Read 對應位址**與 **RCR Read 對應長度**，用於將連接於 RTU 的 AI 模組上的類比輸入點，對應到 D/SR 裝置與長度。而設定 **RCR Write 對應位址**與 **RCR Write 對應長度**，用於將連接於 RTU 的 AO 模組上的類比輸出點，對應到 D/SR 裝置與長度。

3.1.4.14 功能卡 1 設定

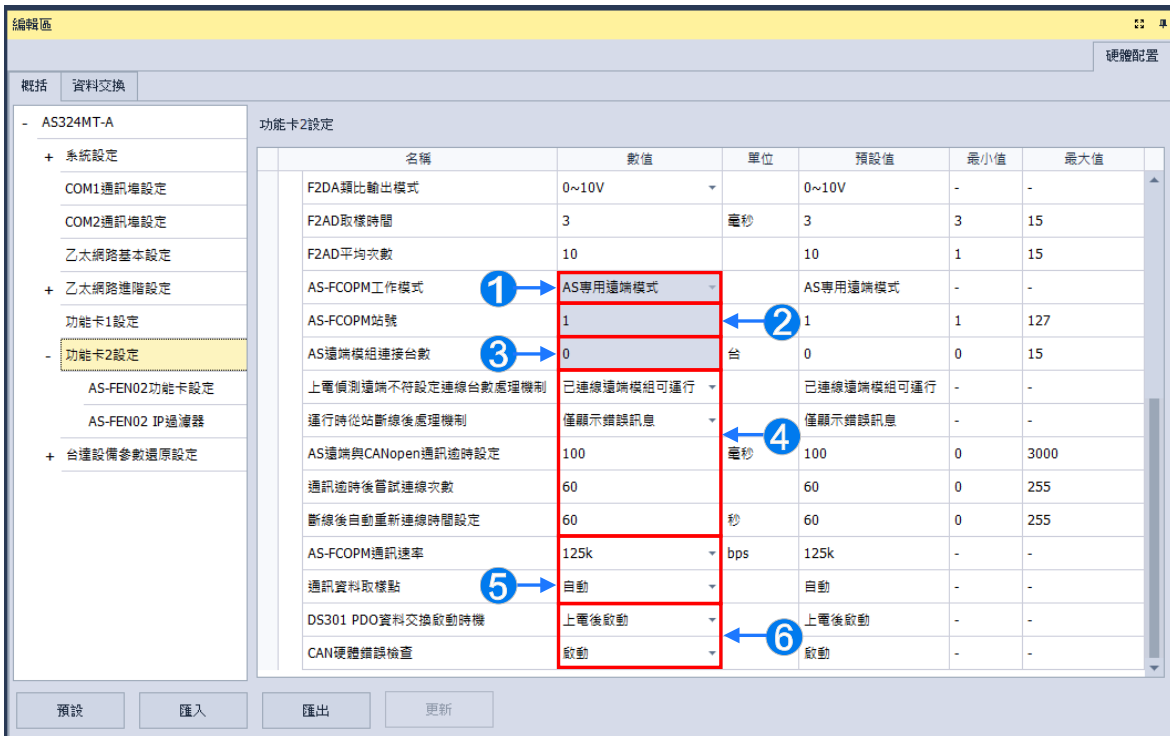
功能卡 1 設定頁面，用於設定可安裝功能卡的功能參數設定。

名稱	數值	單位	預設值	最小值	最大值
Card 1 配置模式選擇	自動偵測		自動偵測	-	-
手動選擇功能卡	無		無	-	-
Card 1 站號	1		1	1	254
通訊協定設定時機	Stop --> Run		Stop --> Run	-	-
通訊速率選擇	9600	bps	9600	-	-
資料位元	7	bit	7	-	-
同位元	偶同位		偶同位	-	-
停止位元	1	bit	1	-	-
MODBUS 通訊模式	ASCII		ASCII	-	-
回應通訊前等待時間	0	毫秒	0	0	3000
接收等待逾時時間	200	毫秒	200	0	3000
F2AD 類比輸入模式	0~10V		0~10V	-	-
F2DA 類比輸出模式	0~10V		0~10V	-	-
F2AD 取樣時間	3	毫秒	3	3	15
F2AD 平均次數	10		10	1	15

- ❶ Card1 配置模式選擇欄位，可選擇自動偵測偵測實際卡片型號與設置，或選擇手動選擇，便可於手動選擇功能卡欄位中選擇要配置的 AS-F232 通訊卡/F422 通訊卡/F485 通訊卡/F2AD 卡/F2DA 卡。
- ❷ 設定 AS-F232 通訊卡/F422 通訊卡/F485 通訊卡的通訊參數。
- ❸ 而 F2AD 類比輸入模式設定 F2AD 卡接受 0~10V 或 4~20mA 訊號；F2DA 類比輸出模式設定 F2DA 卡送出 0~10V 或 4~20mA 訊號。F2AD 取樣時間與 F2AD 平均次數分別用來設定 F2AD 卡取樣的間隔時間與平均次數。

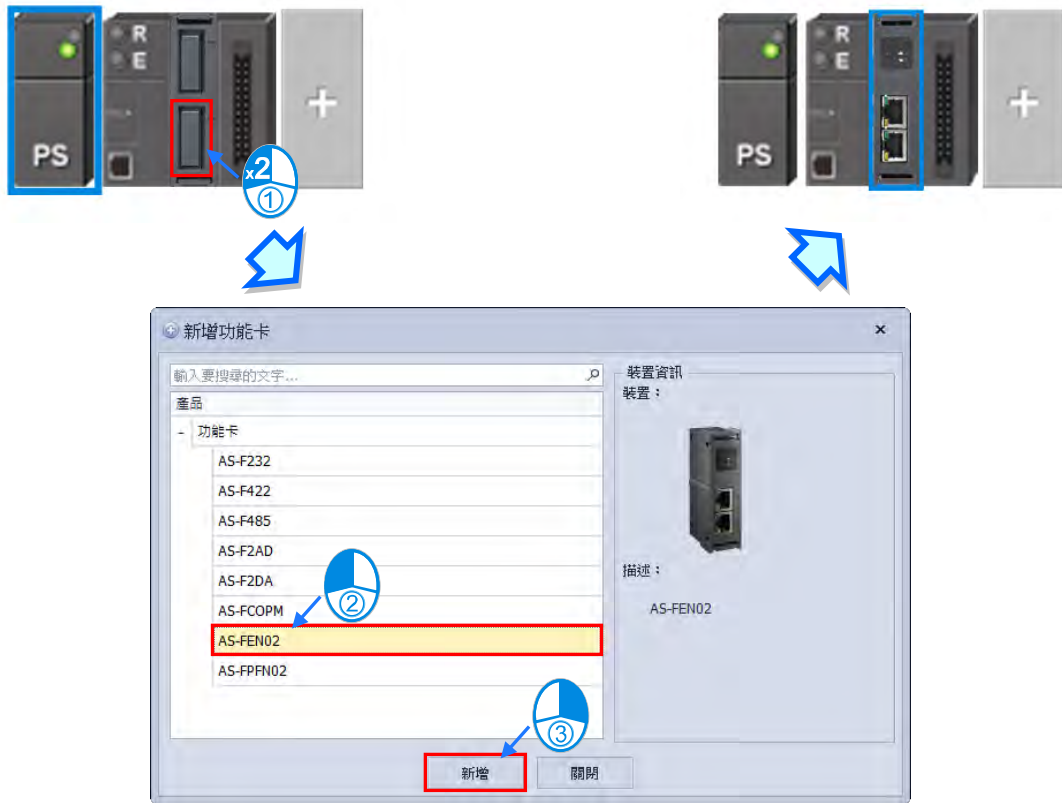
3.1.4.15 功能卡 2 設定

功能卡 2 設定頁面，除了本項目本身設定還可設定此項目內 **AS-FEN02 功能卡設定** 及 **IP 過濾器**。功能卡 2 與功能卡 1 差異為**手動選擇功能卡**欄位多了 AS-FCOPM 卡、AS-FEN02 卡選項，設定為此功能卡後，便可設定以下參數。



- ❶ AS-FCOPM 工作模式欄位，可選擇 AS 專用遠端模式/台達專屬驅動器及 AS 遠端模式/CANopen DS301 協議/台達專屬遠端與 CANopen DS301 混和模式共四種。AS 專用遠端模式為用於連接 AS 系列機種遠端 I/O，台達專屬驅動器及 AS 遠端模式用於對台達伺服及變頻器產品的專用通訊協定以及用於連接 AS 系列機種遠端 I/O，CANopen DS301 用於 DS301 標準通訊協議的場合。
- ❷ 當 AS-FCOPM 工作模式欄位設為 CANopen DS301 時，可設定此站的站號。
- ❸ 當 AS-FCOPM 工作模式欄位設為 AS 專用遠端模式或台達專屬驅動器及 AS 遠端模式時，使用者可以在 AS 遠端模組連接台數輸入連接的台數。
- ❹ 上電偵測遠端不符設定連線台數處理機制為當設定的連接台數與實際連線偵測到的數量不同時，已連線的遠端模組可運行或所有遠端模組皆停止。
運行時從站斷線後處理機制為當運行時遠端 I/O 斷線的處理方式為僅顯示錯誤訊息或停止運行並報警。
AS 遠端與 CANopen 通訊逾時設定為當遠端 I/O 超過設定時間未回應時，視為逾時。
通訊逾時後嘗試連線次數及斷線後自動重新連線時間設定為通訊重傳次數與發生逾時後重新建立連線的時間。
- ❺ AS-FCOPM 通訊速率可設定通訊速率設定。
通訊資料取樣點則可設定資料取樣點的百分率。
- ❻ DS301 PDO 資料交換啟動時機為 CANopen DS301 協議時，功能才可生效，決定 DS301 PDO 資料交換啟動時機，可分為上電後啟動及 PLC RUN 啟動。
CAN 硬體檢查錯誤可以設定啟動或關閉。

而 AS-FEN02 功能卡設定，則可以在硬體配置區的功能卡插槽，雙擊滑鼠左鍵接著選取 AS-FEN02 功能卡這樣就配置完成了，如下圖。



也可於下方編輯區的功能卡 2 設定分頁，在 **Card 2 配置模式** 選擇**手動選擇**並將**手動選擇功能卡**設定為 **AS-FEN02 卡**後，則**功能卡 1 設定**，將無法做設定，接著便可設定「**AS-FEN02 功能卡設定**」之乙太網路 (Ethernet) 埠的通訊參數，如下圖。

3

編輯區

硬體配置

AS324MT-A

系統設定

COM1通訊埠設定

COM2通訊埠設定

乙太網路基本設定

乙太網路進階設定

功能卡1設定

功能卡2設定

AS-FEN02功能卡設定

AS-FEN02 IP過濾器

台達設備參數還原設定

功能卡2設定

名稱	數值	單位	預設值	最小值	最大值
Card 2配置模式選擇	手動選擇		自動偵測	-	-
手動選擇功能卡	AS-FEN02卡		無	-	-
Card 2站號	1		1	1	254
通訊協定設定時機	Stop -> Run		Stop ->...	-	-
通訊速率選擇	9600	bps	9600	-	-
資料位元	7	bit	7	-	-
同位元	偶同位		偶同位	-	-
停止位元	1	bit	1	-	-
MODBUS通訊模式	ASCII		ASCII	-	-

預設 匯入 匯出 更新



編輯區

硬體配置

AS324MT-A

系統設定

COM1通訊埠設定

COM2通訊埠設定

乙太網路基本設定

乙太網路進階設定

功能卡1設定

功能卡2設定

AS-FEN02功能卡設定

AS-FEN02 IP過濾器

台達設備參數還原設定

AS-FEN02功能卡設定

名稱	數值	單位	預設值	最小值	最大值
IP位址	192.168.1.5		192.168.1.5	1.1.1.1	223.255.255.255
子網路遮罩	255.255.255.0		255.255.2...	0.0.0.0	255.255.255.255
閘道位址	192.168.1.1		192.168.1.1	1.1.1.1	223.255.255.255
TCP連線保持時間	30	秒	30	1	65535
IP定址模式	靜態		靜態	-	-

預設 匯入 匯出 更新

IP 過濾器是當 Card 2 使用 AS-FEN02 卡後，也可設定「IP 過濾器」頁面，此頁面為設定網路裝置的過濾功能，只有 IP 位址列於清單範圍中的裝置才允許跟 AS-FEN02 卡進行通訊，此功能最多可設定 8 組的位址範圍，如下圖。



要啟動 IP 過濾器功能，請勾選啟動 IP 位址過濾設定，便可輸入以下 1~8 組的 IP 過濾範圍的起始與結束位址。

3.1.4.16 台達設備參數還原功能

*操作軟體：必須使用 ISPSOft V3.10 (含) 以上，搭配 EIP Builder V1.70 (含) 以上。

*PLC 主機：AS 主機韌體必須更新到 V1.08.00 (含) 以上。

*驅動器：目前只支援還原 Delta C2000 變頻器參數版本：V2.06 (含) 以上。

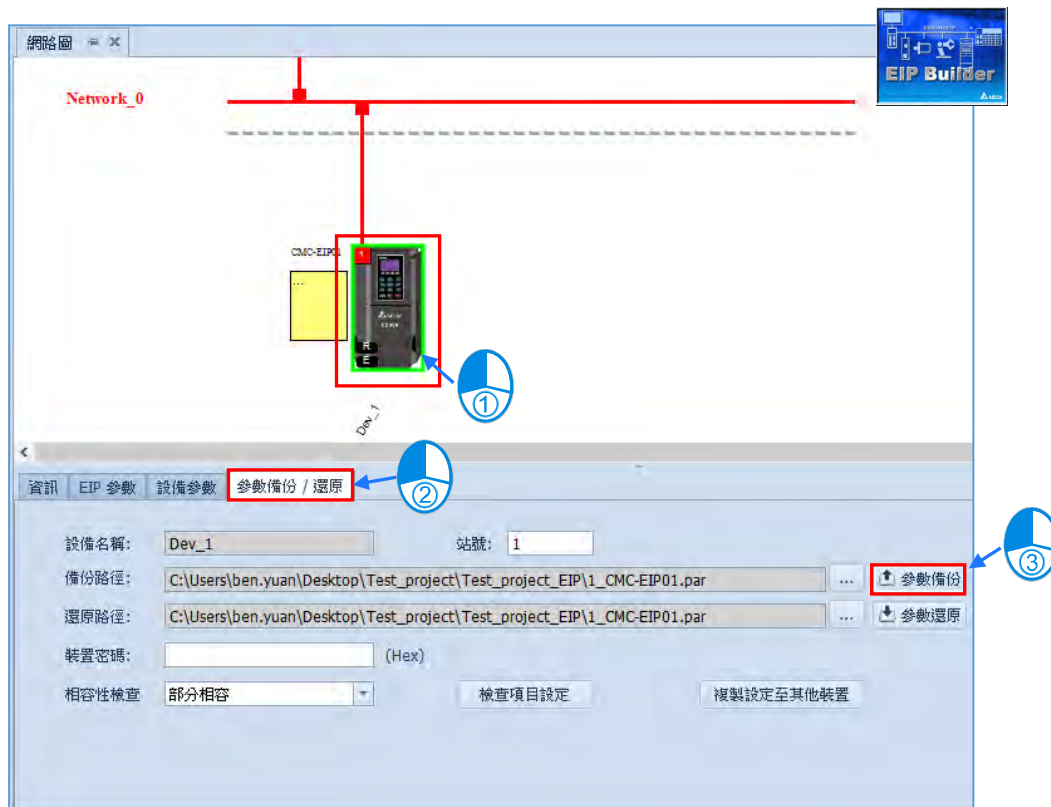
● 目的

可使用 EIP Builder 軟體進行備份參數動作並將備份結果下載到 AS 主機的 SD 卡上 (需插入 SD 卡)，如果已經做好備份的則可以略過「將設備參數備份到 PLC 的 SD 卡上」的步驟，直接跳到「主機還原的相關設定」步驟，接著設定「台達設備參數還原設定」並下載到 AS 主機，就可以透過設定 PLC 的 SR 裝置，讓 AS 主機依需求進行台達設備的參數還原動作，目前 AS 主機最高可以支援 32 台台達設備，關於詳細的使用方式請參考 AS 系列操作手冊。

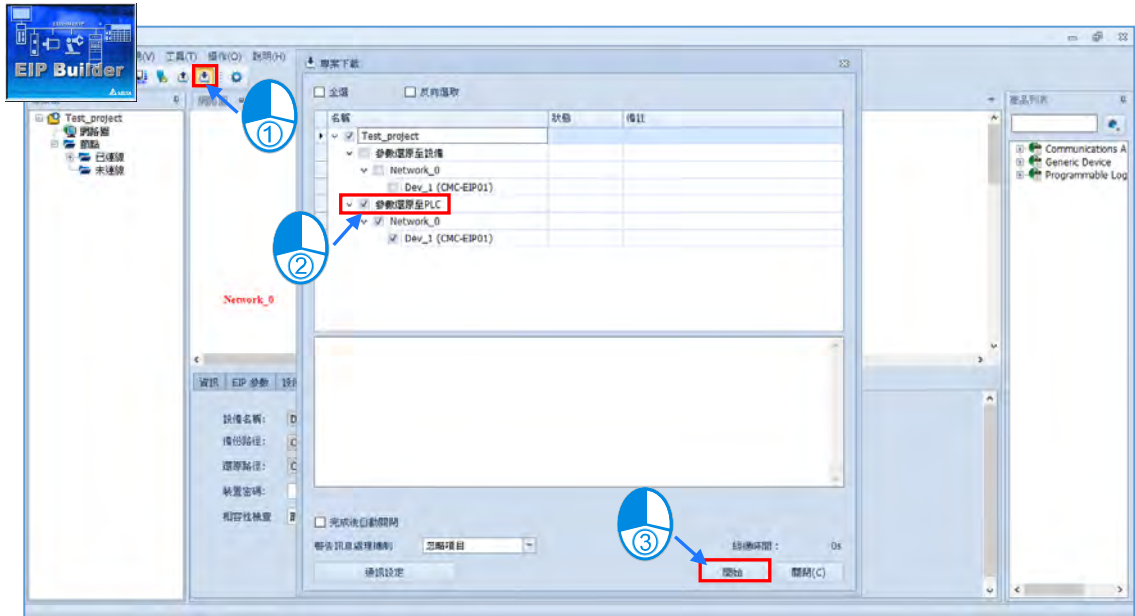
- 將設備參數備份到 PLC 的 SD 卡上

請參考第 3.1.3.16 節開啟 EIP Builder 並且完成專案的網路規畫後，先點選要備份的 C2000 變頻器「CMC-EIP01」設備，接著點選「參數備份 / 還原」分頁，再點選「參數備份」直接透過網路對台達驅動器進行參數備份（儲存在電腦 副檔名.par）如下圖，多台設備時可使用不同的站號，檔案命名會自動隨著站號而改變，以區分各站號的參數備份檔，亦可直接點選「參數還原」透過網路對台達驅動器進行參數還原。

3



接著先確認 SD Card 已經放進 AS 主機內，並且確認 PLC 處於 STOP 狀況，就可以執行下載動作，並且選擇參數還原至 PLC，再按下開始按鈕後就會進行選定設備的參數備份動作，完成後就表示資料已經存放到 PLC 的 SD 卡內部了。



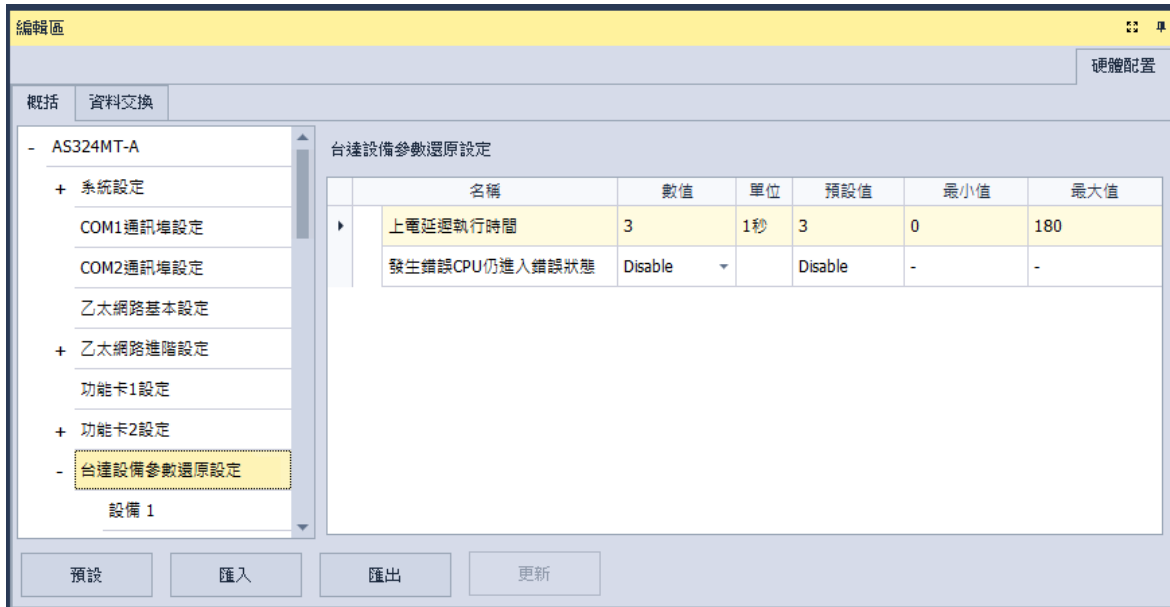
補充說明

- ◆ 預設儲存路徑
 - AS300 : \SDCard\PLC CARD\AS300\DevPara
 - AS200 : \SDCard\PLC CARD\AS200\DevPara
 - ◆ 檔案名稱格式 : mmmm_nn_tt_pppppppppppppppppppp.par
 - mmmm : 設備或模組代碼 (PLC 主機 : 0000)
 - nn : 設備通訊端口編號 (PLC Ethernet 通訊口編號 : 03)
 - tt : 通訊協議類型 (固定代碼 : 01)
 - pppppppppppppppppppp : 驅動器設備資訊 (IP + ID + 補 0)
- Ex. IP : 192.168.1.9, ID : 9 · 檔名為 : C0A80109 (IP) + 09 (ID) + 000000 (補 0) = C0A8010909000000.par

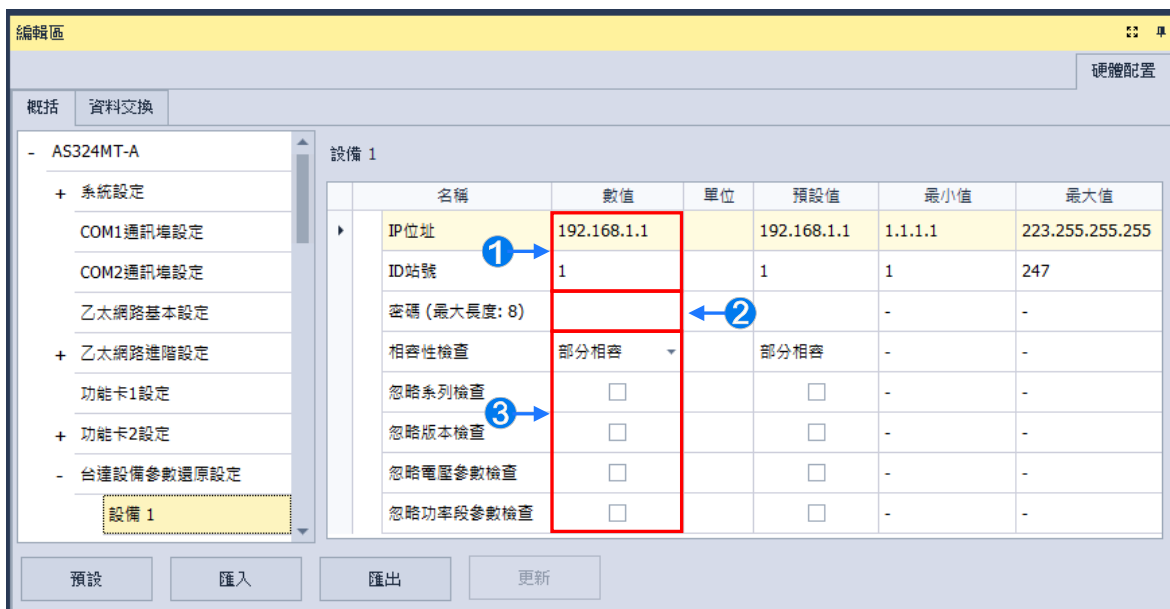
● 主機還原的相關設定

在主機的 HWCONFIG 的「台達設備參數還原設定」的頁面可以設定上電延遲執行的時間預設為 3 秒鐘，也可以設定發生錯誤時 CPU 是否要進入 STOP 狀態。

3



在「設備 1~32」的頁面可以設定每一台台達設備的連接資訊與還原條件，如下圖。



- ❶ 輸入目標設備的 IP 地址與 ID 站號。
- ❷ 輸入目標設備的密碼，最大長度為 8 個字元，格式可選擇。
- ❸ 相容性檢查欄位是設定由目標設備決定檢查條件是否可以忽略，設為部分相容的時候可以勾選下面四個還原條件。

嘗試忽略系列檢查：設備的系列不相同的時後依然可以**嘗試**還原。

嘗試忽略版本檢查：設備的版本不相同的時後依然可以**嘗試**還原。

嘗試忽略電壓參數檢查：設備的電壓參數不相同的時後依然可以**嘗試**還原。

嘗試忽略功率段參數檢查：設備的功率段參數不相同的時後依然可以**嘗試**還原。

當**相容性檢查**欄位設為**完全相容**的時候則必須所有條件相容時才執行還原。

當**相容性檢查**欄位設為**強制寫入**的時候則會**嘗試**直接強制執行還原。

若目標設備不支援**相容性檢查**的條件，則即使設定了忽略條件，還是有可能無法還原，

關於目標設備支援那些條件，則需要參考所連接的目標設備使用手冊。

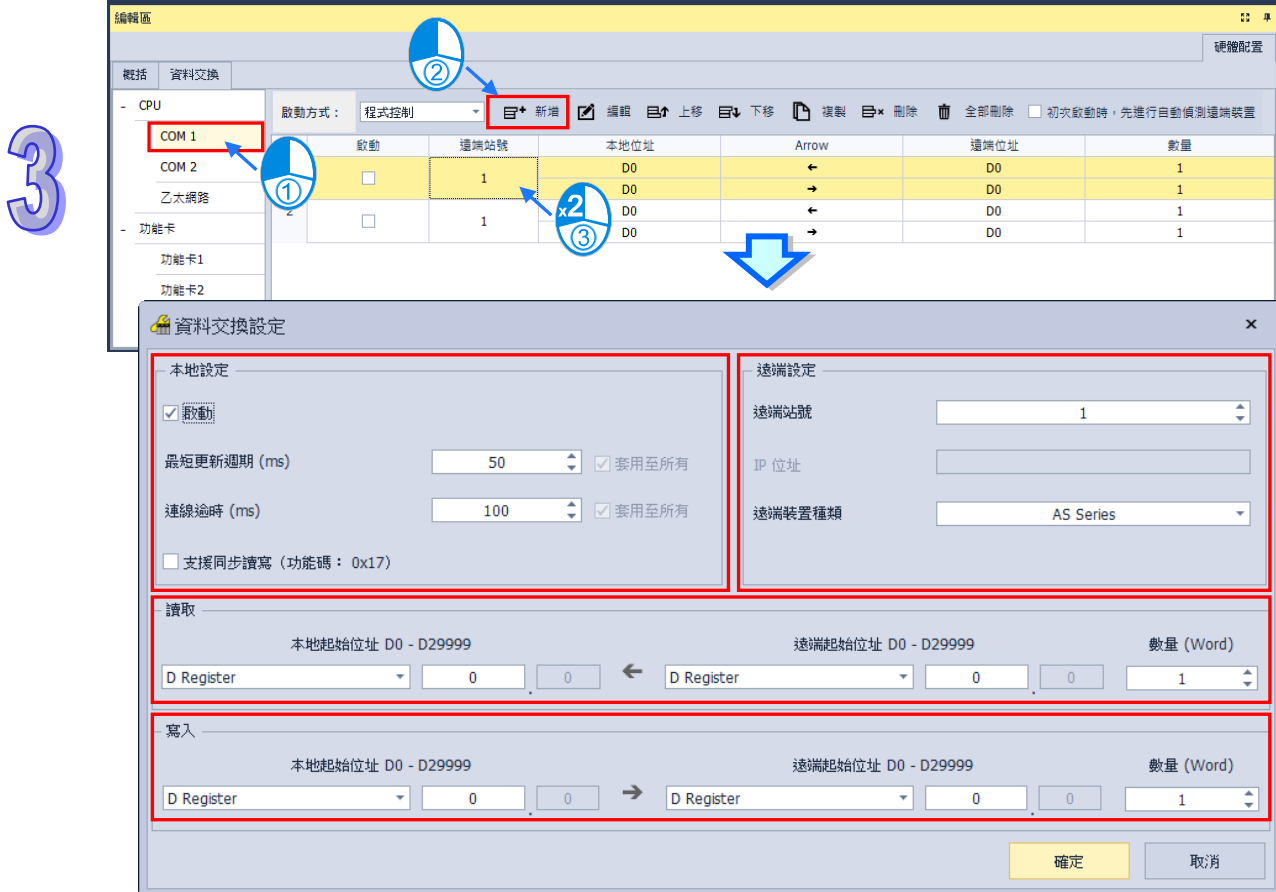
接著必須參照下表的 **SR 裝置說明**，依據需求去設定對應的數值到 **SR 裝置** (停電保持)，可以使用 **PLC 編輯軟體**、**HMI** 或是 **SCADA** 等方式來進行設定，完成後下次主機重新啟動時就會依設定進行對應的參數還原動作。

3

SM / SR 編號	功能名稱	說明
SR1600	參數還原執行代碼	還原動作代碼如下數值與說明： 0：不執行。(預設值) 1：當 PLC 上電或 Stop→Run 時，執行參數還原動作，直到執行完成時才清除為 0
SR1601.0~SR1601.15 ⋮ SR1602.0~SR1602.15	指定參數還原的目標驅動器編號	SR1601.0 對應到目標驅動器編號 1 SR1601.1 對應到目標驅動器編號 2 ⋮ SR1602.15 對應到目標驅動器編號 32 Off：不執行。(預設值) On：被指定執行參數還原之動作

3.1.4.17 資料交換

在資料交換分頁的 **COM**、**乙太網路**、或**功能卡**，點擊「新增」後，點選「編輯」或雙擊所要進行設定的資料交換表，開啟資料交換表的設定項目，如下圖。



- 「本地設定」的資料區塊

勾選**啟動**代表要啟動此組資料交換表，並且隨著前述的**啟動方式**啟動。**最短更新週期 (ms)**為設定此組資料交換表進行資料交換的時間週期。**連線逾時 (ms)**為設定若目標設備超過此設定時間未回應，則為逾時。勾選**支援同步讀寫 (功能碼 0x17)**將利用特定的 MODBUS 功能碼，主站便可在一次的命令當中完成**讀與寫**的動作，以提高資料交換的效率；不過設定時，請確認參與資料交換的所有裝置都有支援同步讀寫的 MODBUS 功能碼，否則便可能在接收到主站的命令後，會因無法識別該功能碼而造成讀寫失敗。

- 「遠端設定」的資料區塊

遠端站號用於進行資料交換的目標從站設備站號。

IP 位址為目標從站設備 IP 位址 (只支援乙太網路)。

遠端裝置種類為目標從站設備的機種，包括台達 PLC 與標準 MODBUS 設備。

- 「讀取」的資料區塊

設定當 PLC 主機自遠端設備讀取資料時，PLC 主機用以存放資料的**裝置類別**、**裝置起始位址**，接著指定要讀取的遠端設備**裝置類別**及**起始位置**。設定輸入的**資料長度**。

- 「寫入」的資料區塊

設定當 PLC 主機寫入資料至遠端設備時 PLC 主機來源資料的**裝置類別**、**起始位址**，接著指定被寫入資料的遠端設備**裝置類別**及**起始位置**。設定輸出的**資料長度**。

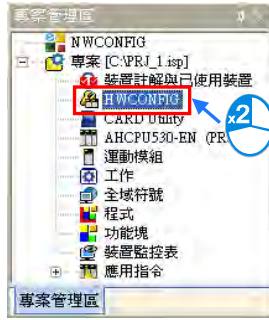
3.2 硬體規劃工具 - HWCONFIG 3.0

HWCONFIG 是 ISPSOFT 的硬體規劃工具之一，其功能包括模組背板的配置、CPU 主機參數的設定、模組參數設定、硬體參數上下載及簡易的線上偵測與診斷功能。本章節會先就各機種皆通用的功能先行敘述，再指出某機種及其特有的功能。

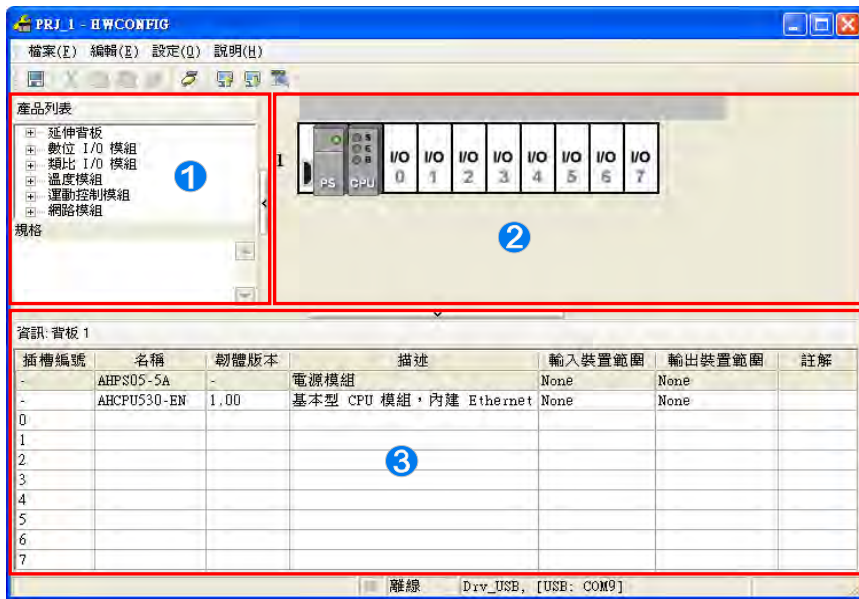
請注意！所有硬體規劃的參數於設定完畢後，必須被下載至主機中才可生效。

3.2.1 HWCONFIG 的環境介紹

於專案管理區的「**HWCONFIG**」項目上雙擊滑鼠左鍵即可開啟硬體規劃工具的工作視窗。



3



- ❶ 產品列表：此區以目錄管理的方式呈現所有可用的硬體資源。
- ❷ 系統配置區：為主要的工作區，使用者可於此處進行整個系統的配置與設定。
- ❸ 資訊列表：以列表方式管理目前的系統配置。

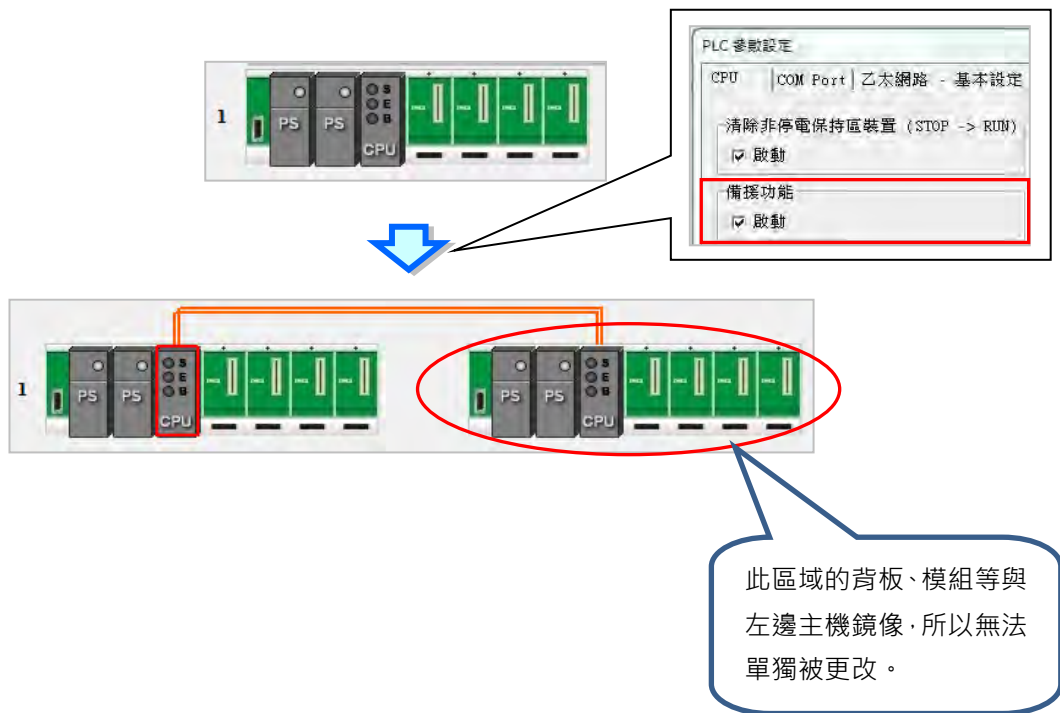
系統配置區會以圖示顯示目前的配置狀況。AH 系列機種的每個背板圖示的左側會顯示該背板的編號；操作時，以滑鼠點選某個模組圖示即可將該模組選取起來。



而要選取背板時，請先將滑鼠移至背板圖示左側的連接端口位置，接著點擊滑鼠左鍵即可。



若是使用 AH560 備援系列機種，滑鼠左鍵雙擊 CPU 之後，在 PLC 參數設定頁面，開啟「備援功能」後，會在右側產生備援主機，備援主機的設定是與左邊主機鏡像，所以無法單獨被更改。

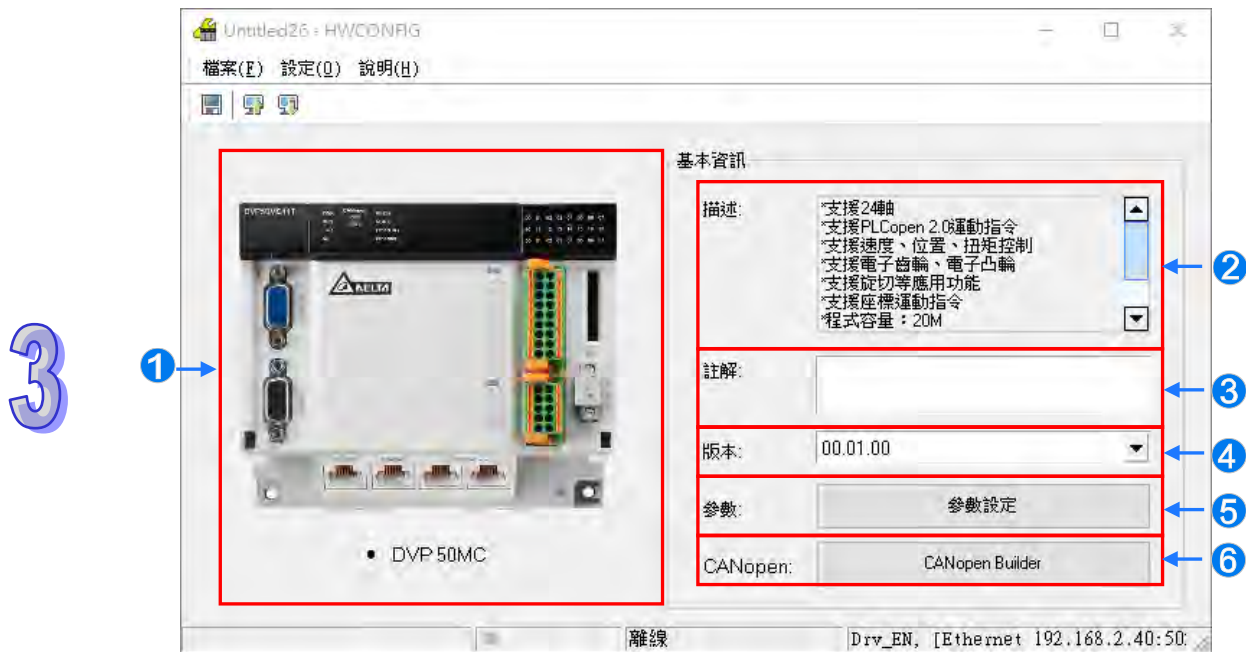


3

系統配置區與資訊列表會即時同步更新，當系統配置區與資訊列表某一個模組被選取的時候，兩方的模組都會被圈選顯示。



若是使用 DVPxxMC 機種，硬體規劃工具的工作視窗如下圖顯示。



- ❶ 機種主機圖樣。
- ❷ 描述：描述機種主機的功能。
- ❸ 註解：主機的註解。
- ❹ 版本：主機之 DDF 版本。
- ❺ 參數：點擊「參數設定」鈕，即可進入主機參數設定。
- ❻ CANopen：點擊「CANopen Builder」鈕，即可開啟「CANopen Builder」軟體。

3.2.2 模組配置

3.2.2.1 新增模組

➤ 方法一

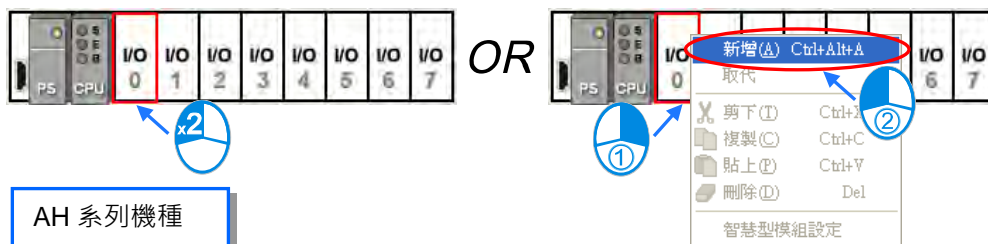
AH 系列機種於左側**產品列表**中選取要加入的模組後，直接將其拖曳至要放置的空白插槽即可。



*.由產品列表拖曳模組至系統配置區時，該模組只能置放於空白的插槽，若拖曳至配有模組的插槽上時，系統將禁止該操作。

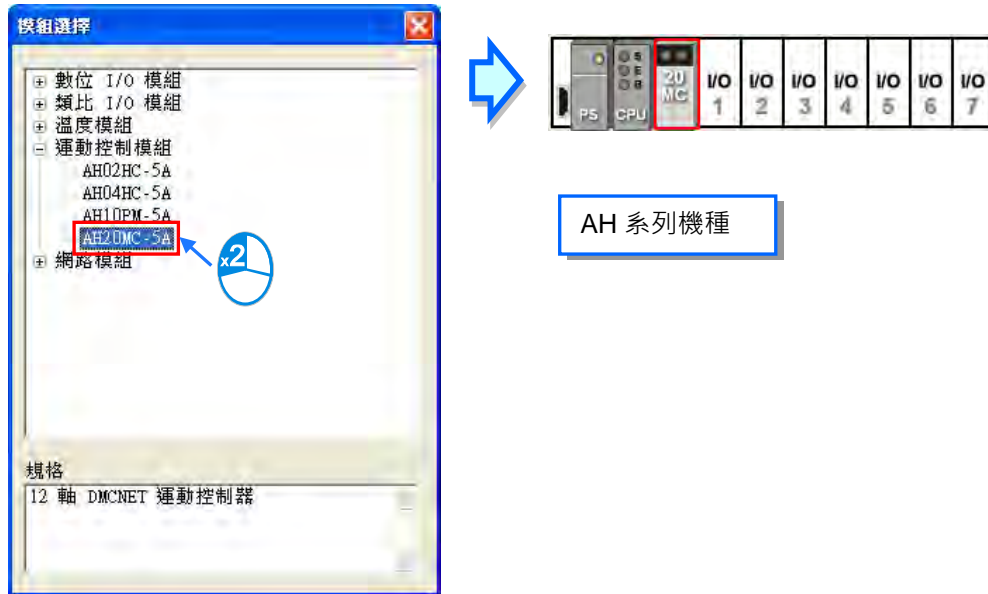
方法二

(1) AH 系列機種於空白插槽，雙擊滑鼠左鍵或是點擊右鍵後選擇「新增 (A)」，若是 AS 系列機種，則於 + 位置雙擊滑鼠左鍵或是點擊右鍵後選擇「開啟 (O)」。



- (2) 在要加入的模組上雙擊滑鼠左鍵即可。

3



➤ 方法三

- (1) 於資訊列表中，在對應插槽的空白行列上雙擊滑鼠左鍵，或點擊滑鼠右鍵後選擇「新增」，之後畫面便會出現模組選擇視窗。

資訊: 背板 1

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0						
1						
2						
3						
4						
5						
6						
7						

OR

資訊: 背板 1

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0						
1						
2						
3						
4						
5						
6						
7						

新增(A) Ctrl+Alt+A

取代

剪下(T) Ctrl

複製(C) Ctrl+C

貼上(P) Ctrl+V

刪除(D) Del

智慧型模組設定

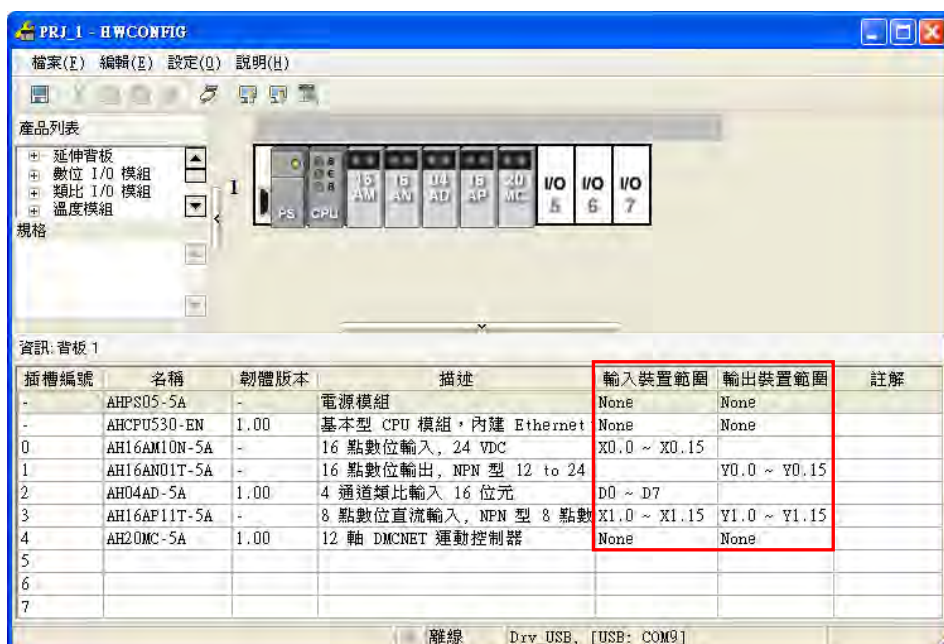
- (2) 當於選單中點選某個模組時，於下方的**規格**欄位便會顯示該模組的規格及描述資料，在確定欲加入的模組後，於該項目上雙擊滑鼠左鍵即可將其加入。



3

3.2.2.2 模組的位址配置

當 AH 系列機種模組有經常性的資料更新需求時，例如類比輸入模組會持續更新所接收到的類比信號，並將其轉換為主機可處理的資料，因此系統便會自動為其配置對應的裝置位址用以存取資料，而配置的裝置位址亦會顯示於**資訊列表**中的「**輸入裝置範圍**」與「**輸出裝置範圍**」兩個欄位。請參考下圖範例。AS 系列機種主機群組不支援手動設定位址配置，依據模組的安裝位置自動配置固定位址，而 RTU 遠端群組僅能設置第一台 SCM 模組的起始裝置位址，依序配置後續的模組，AS 系列機種遠端模組配置請參考第 3.1.3.8 節。



AH16AM10N-5A 模組 (插槽 0)

此為 16 點輸入的數位 IO 模組，系統為其配置了 X0.0 ~ X0.15 共 1 個 WORD 的裝置用以存放輸入該模組的各點狀態。

AH16AN01T-5A 模組 (插槽 1)

此為 16 點輸出的數位 IO 模組，系統為其配置了 Y0.0 ~ Y0.15 共 1 個 WORD 的裝置用以存放將由該模組輸出的各點狀態。

3

AH04AD-5A 模組 (插槽 2)

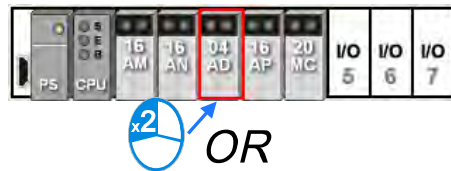
此為 4 通道的類比輸入模組，系統為其配置了 D0 ~ D7 共 8 個 WORD，而外部輸入的類比信號便會在經過轉換後被存放於 D0~D7。

AH16AP11T-5A (插槽 3)

此為 8 點輸入/8 點輸出的數位 IO 模組，且因配置裝置時至少須以 1 個 WORD 為單位，因此系統配置了 X1.0 ~ X1.15 (實際使用範圍為 X1.0 ~ X1.7) 的裝置用以存放輸入該模組的各點狀態；而另外也配置了 Y1.0 ~ Y1.15 (實際使用範圍為 Y1.0 ~ Y1.7) 的裝置用以存放將由該模組輸出的各點狀態。

AH20MC-5A (插槽 4)

此為運動控制模組，且因此模組並沒有經常性的資料更新需求，因此系統不會為其配置裝置位址。當要更進一步得知所配置的裝置位址對應到哪些輸出入功能時，請於該模組的圖示或資訊列表的行列上雙擊滑鼠左鍵，之後系統便會開啟該模組的內部參數設定視窗。



插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EM	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH16AM10N-5A	-	16 點數位輸入，24 VDC	X0.0 ~ X0.15		
1	AH16AN01T-5A	-	16 點數位輸出，NPN 型 12 to 24		Y0.0 ~ Y0.15	
2	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
3	AH16AP11T-5A	-	8 點數位直流輸入，NPN 型 8 點數	X1.0 ~ X1.15	Y1.0 ~ Y1.15	
4	AH20MC-5A	1.00	12 軸 DMCNET 運動控制器	None	None	
6						
7						

*.於資訊列表雙擊滑鼠左鍵時，建議點擊在「插槽編號」、「名稱」或「描述」等位置，以避開可編輯的欄位。

於參數設定視窗的主畫面點選「常態交換區」標籤便可切換至裝置位址與對應功能的資訊列表。

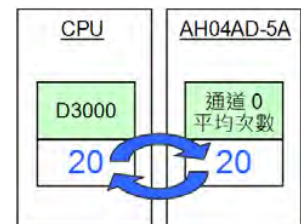
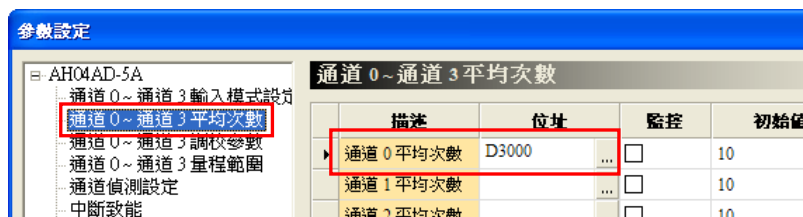


3


描述	位址
▶ 通道 0 輸入值	D0 ~ D1
通道 1 輸入值	D2 ~ D3
通道 2 輸入值	D4 ~ D5
通道 3 輸入值	D6 ~ D7

除了系統會為經常性更新的資料而自動配置用以存取該資料的裝置之外，對於模組內部非經常性更新的參數，使用者亦可自行定義對應該參數的 **D 對應裝置**，而之後當系統開始運作時，CPU 主機中的 **D 對應裝置** 便會與指定的模組內部參數保持同步，因此使用者可直接經由 **D 對應裝置** 而間接的存取模組內部的資料，透過此方法會較傳統採用 **FROM / TO** 指令來存取模組內部資料的方法來的更有效率。如下述範例，我們於 **AH04AD-5A** 的模組參數設定視窗中，將 D3000 定義為模組參數 - 「**通道 0 平均次數**」的 **D 對應裝置**，因此後續當我們將 HWCONFIG 的參數下載至主機之後，我們便可直接修改 CPU 主機的 D3000 內容值而間接變更 **AH04AD-5A** 模組的**通道 0 平均次數**。

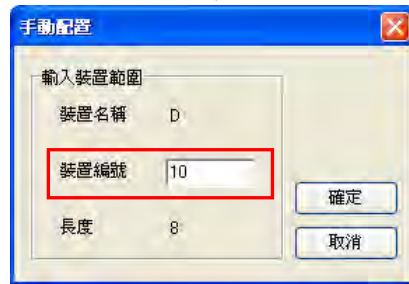
除此之外，運作期間若對 **D 對應裝置** 寫入模組參數所不允許的值時，系統會立即將兩者修正回原來的值。



*.關於模組內部參數的設定方法將於第 3.7 節當中另行介紹；而關於 **FROM / TO** 指令則請參照各機種的指令手冊。

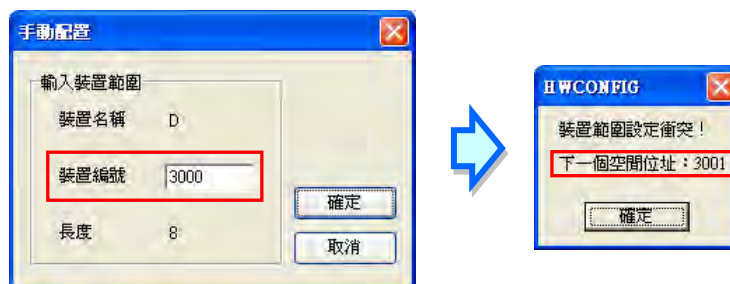
對於資訊列表中的輸入/輸出裝置範圍，除了可由系統自動配置之外，使用者亦可自行定義；設定時請先於資訊列表中點選欲設定的欄位後，於欄位右側的  按鈕點擊滑鼠左鍵或是直接輸入啟始位址，此時系統便會開啟設定視窗，而接著在輸入完啟始位址之後，按下「確定」按鈕即可。

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH16AM10N-5A	-	16 點數位輸入，24 VDC	X0.0 ~ X0.15		
1	AH16AN01T-5A	-	16 點數位輸出，NPN 型 12 to 24		Y0.0 ~ Y0.15	
2	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
3	AH16AP11T-5A	-	8 點數位直流輸入，NPN 型 8 點數	X1.0 ~ X1.15	Y1.0 ~ Y1.15	



插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH16AM10N-5A	-	16 點數位輸入，24 VDC	X0.0 ~ X0.15		
1	AH16AN01T-5A	-	16 點數位輸出，NPN 型 12 to 24		Y0.0 ~ Y0.15	
2	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D10 ~ D17		
3	AH16AP11T-5A	-	8 點數位直流輸入，NPN 型 8 點數	X1.0 ~ X1.15	Y1.0 ~ Y1.15	

無論是指定資訊列表中的輸入/輸出裝置範圍，或是定義模組內部參數的 D 對應裝置，所輸入的裝置位址範圍皆不可重疊，而當設定的位址發生衝突時，系統將會依據目前所輸入的位址而自動修正並提示下一個可用的空間位址；例如裝置 D3000 已被其他的模組佔用，則當要重複指定使用 D3000 時，系統便會由輸入的 D3000 開始自動搜尋下一個可用的空間位址。



當模組的輸入/輸出裝置範圍被變更之後，後續再加入模組時，其預設的裝置位址則會以最後變更的裝置位址為依據，且自動跳過已使用的裝置位址來為其進行配置；而即使最後變更的位址，其編號不是系統中最大的，新模組的位址配置還是會依據最後變更的裝置位址而定。

參考下述範例，首先我們將位於插槽 2 的模組輸入裝置範圍變更為 D50 ~ D57。

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EI	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
1	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D100 ~ D107		
2	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D200 ~ D207...		
3						


插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EI	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
1	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D100 ~ D107		
2	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D50 ~ D57		
3						

之後當加入新模組時，其輸入裝置範圍將會自動由最後變更的 D50 ~ D57 之後開始配置，而不是依據整個系統中最大的 D100 ~ D107，因此最後該模組的輸入裝置範圍便會被配置於 D58~D65。

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EI	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
1	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D100 ~ D107		
2	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D50 ~ D57		
3	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D58 ~ D65		
4						

3.2.2.3 註解編輯

於系統配置區上方的灰色區域點擊滑鼠左鍵後，於下拉的編輯欄位便可輸入該專案的硬體配置註解，而輸入時按下鍵盤的【Shift】+【Enter】鍵即可進行換行，輸入完畢後直接按下【Enter】鍵即可。

AH5x0、AH5x1 系列機種及 AH560 備援系列機種於資訊列表中欲加入模組註解的「註解」欄位上點擊滑鼠左鍵後，直接按下鍵盤或點選註解欄位右側的  按鈕便可於彈出的視窗中輸入該模組的註解。

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH08DA-5B	1.00	8 通道類比輸出 16 位元		D0 ~ D15	
1						



插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH08DA-5B	1.00	8 通道類比輸出 16 位元		D0 ~ D15	AO: CH1~CH8
1						

3

3.2.2.4 刪除模組

刪除已配置模組的方法有下列兩種 (CPU 不可刪除，AH 系列機種的電源模組不可刪除)：


● 方法一

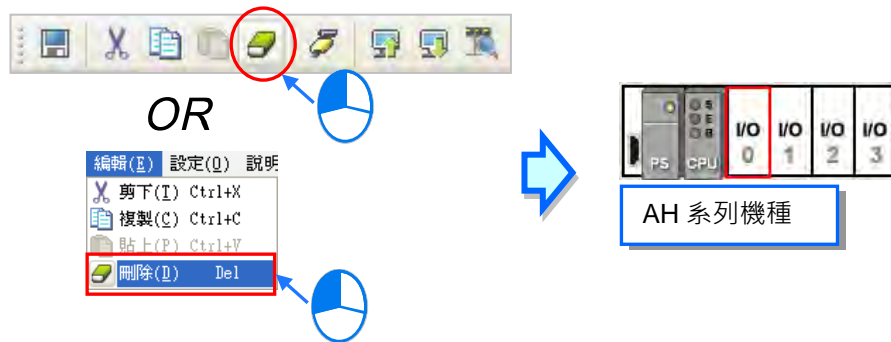
(1) 先於系統配置區或資訊列表上點選欲刪除的模組。



OR

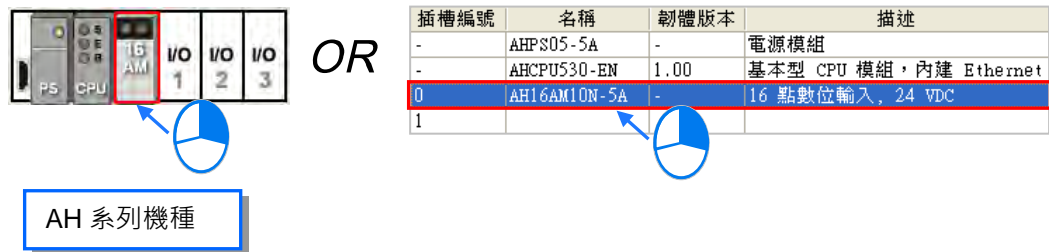
插槽編號	名稱	韌體版本	描述
-	AHPS05-5A	-	電源模組
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet
0	AH16AM10N-5A	-	16 點數位輸入，24 VDC
1			

- (2) 於 HWCONFIG 的功能工具列點選 **編輯 (E)** > **刪除 (D)** 或點擊圖示工具列的  圖示，亦會是直接按下鍵盤的【DEL】鍵。

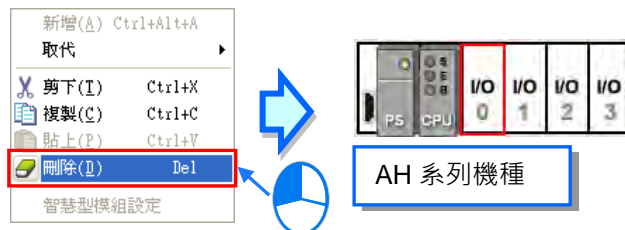


● 方法二

- (1) 先於**系統配置區**或**資訊列表**上點選欲刪除的模組後按下滑鼠右鍵。

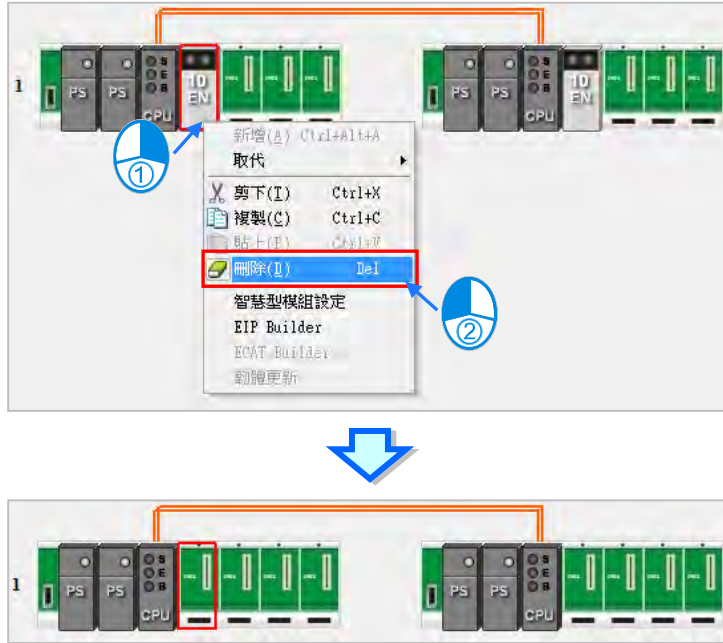


- (2) 之後於快捷選單中點選「**刪除 (D)**」。



補充說明

若使用者選擇的機種為 AH560 備援系列機種，在開啟「**備援功能**」後，如需刪除「**系統配置區**」備援主背板的模組，只能在左側備援主背板上刪除配置，而右側備援主背板上配置為左側備援主背板的鏡像映射，無法進行修改。



3

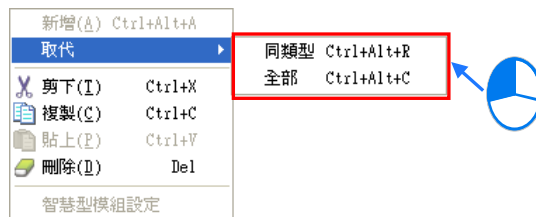
3.2.2.5 變更模組

若是 AH 系列機種想要對系統配置區的模組進行取代變更，可以依循以下步驟：

- (1) AH 系列機種於系統配置區或資訊列表上點選欲變更的模組後按下滑鼠右鍵(CPU 與電源模組不可變更)。

插槽編號	名稱	韌體版本	描述
-	AHPS05-5A	-	電源模組
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet
0	AH16AM10N-5A	-	16 點數位輸入，24 VDC
1			

- (2) 於快捷選單中點選「取代」，之後再選擇取代的方式。關於兩種取代的方式說明如下。



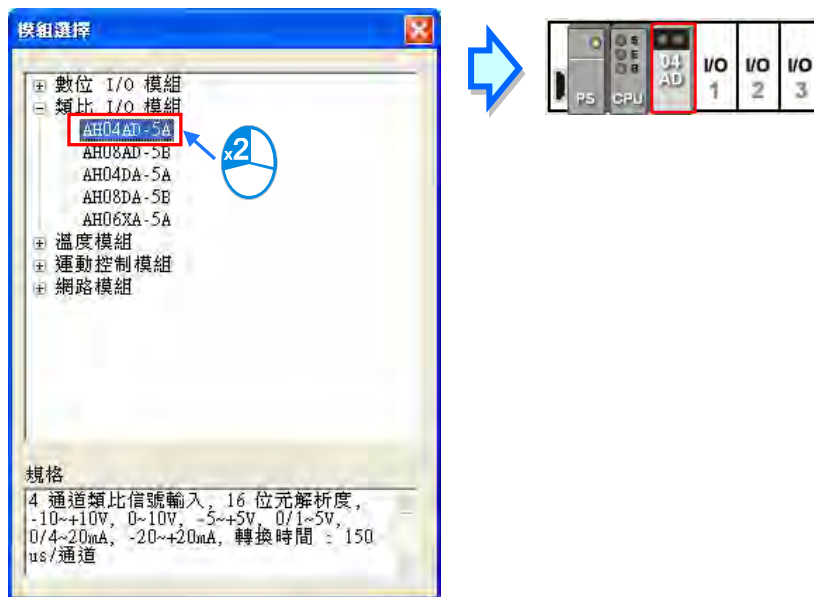
➤ 同類型

僅將所選取的模組取代為同類型的其他模組，而取代之後，新模組的輸入/輸出裝置範圍將會與原本相同，而其他的參數設定若無法完全對應時將會被恢復為系統預設值。

➤ 全部

可將所選的模組取代為任何類型的模組，而執行之後，其效果等同將原本的模組刪除後，再重新插入另一模組，因此新模組的輸入/輸出裝置範圍將會重新配置，而模組的其他參數設定也將會重新恢復為系統預設值。

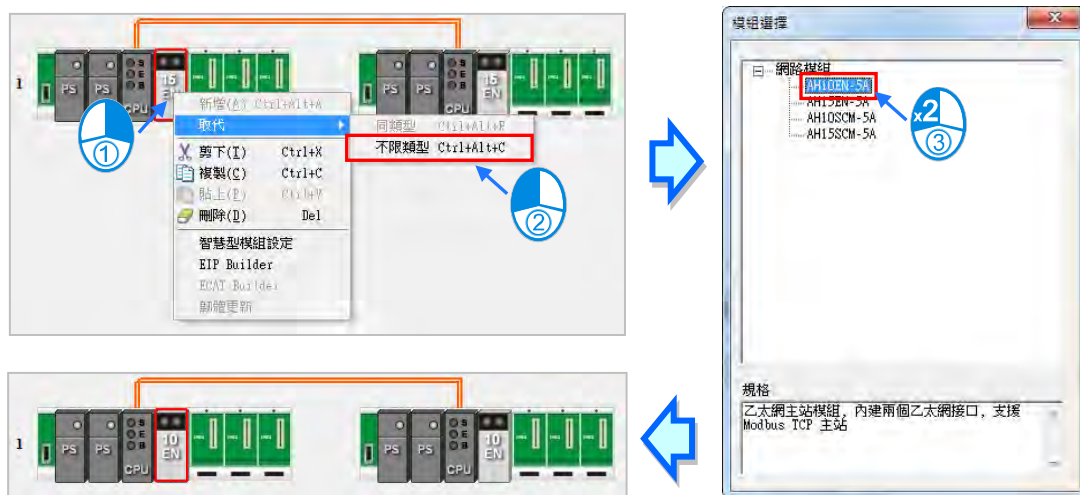
- (3) 點選欲執行的取代方式後，畫面便會出現模組選擇視窗，而視窗中只會依據所選的取代方式而顯示可選擇的模組項目；決定欲替換的模組之後，於該模組項目上雙擊滑鼠左鍵即可。



3

補充說明

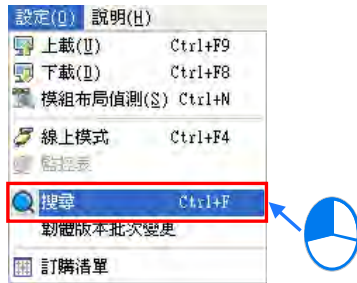
若使用者選擇的機種為 AH560 備援系列機種，在開啟「備援功能」後，如需變更「系統配置區」備援主背板上的模組，只能在左側備援主背板上變更配置，而右側備援主背板上配置為左側備援主背板的鏡像映射，無法進行修改。



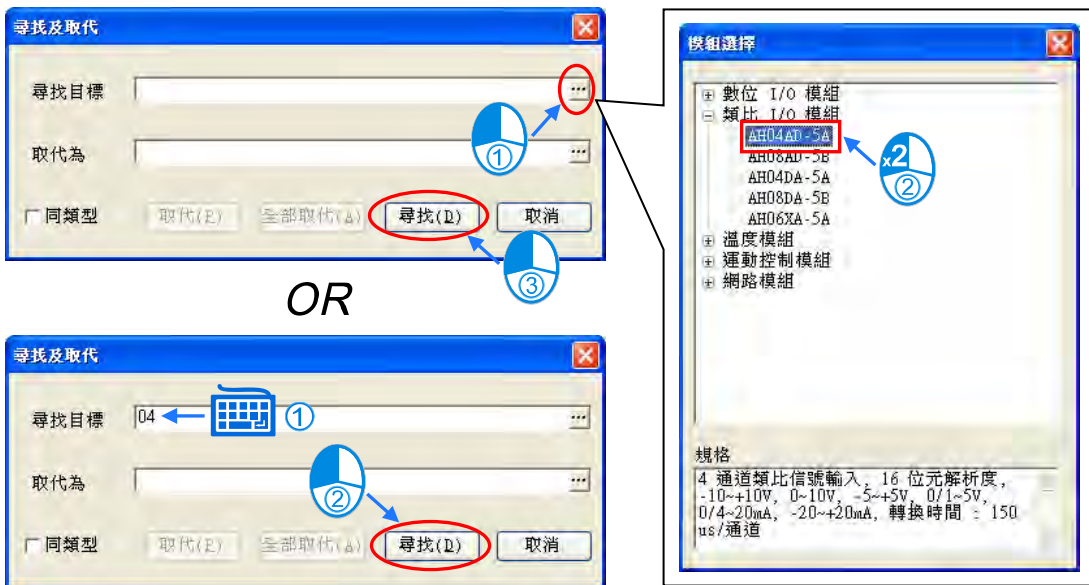
3.2.2.6 模組的搜尋/取代功能

若是 AH 系列機種想要對系統配置區的模組進行搜尋/取代，則可以依循以下步驟：

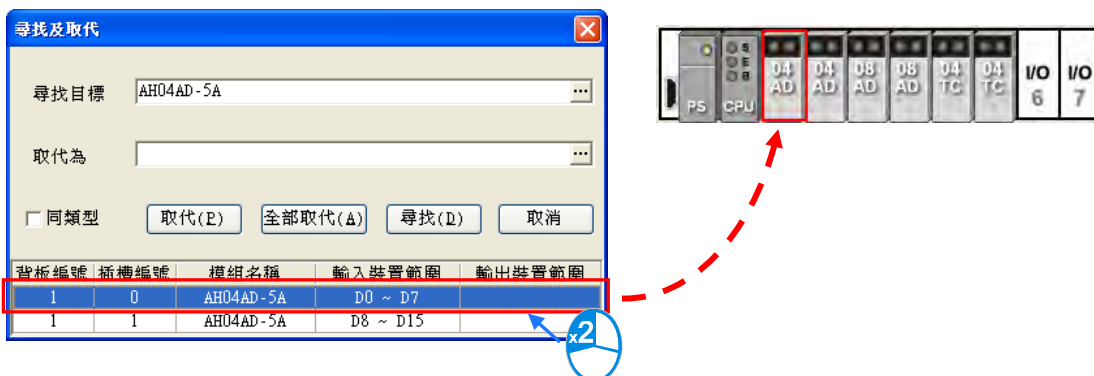
- (1) 於 HWCONFIG 的功能工具列中點選 **設定 (O) > 搜尋 (S)**，之後畫面便會出現操作視窗。



- (2) 點擊「**尋找目標**」欄位右側的按鈕，之後於出現的選單中選擇欲搜尋的模組型號後，於該項目上雙擊滑鼠左鍵，或是直接於該欄位中輸入模組型號的關鍵字串。完成後按下「**尋找**」按鈕。



- (3) 搜尋完畢後，符合搜尋條件的模組便會被列於下方的列表中；而於某一項目上雙擊滑鼠左鍵後，位於系統配置區的該模組便會被選取起來。



- (4) 當要取代某個模組時，請先於下方的搜尋列表中點選要被取代的項目，之後再勾選是否只取代為同類型的模組，其差異如下：



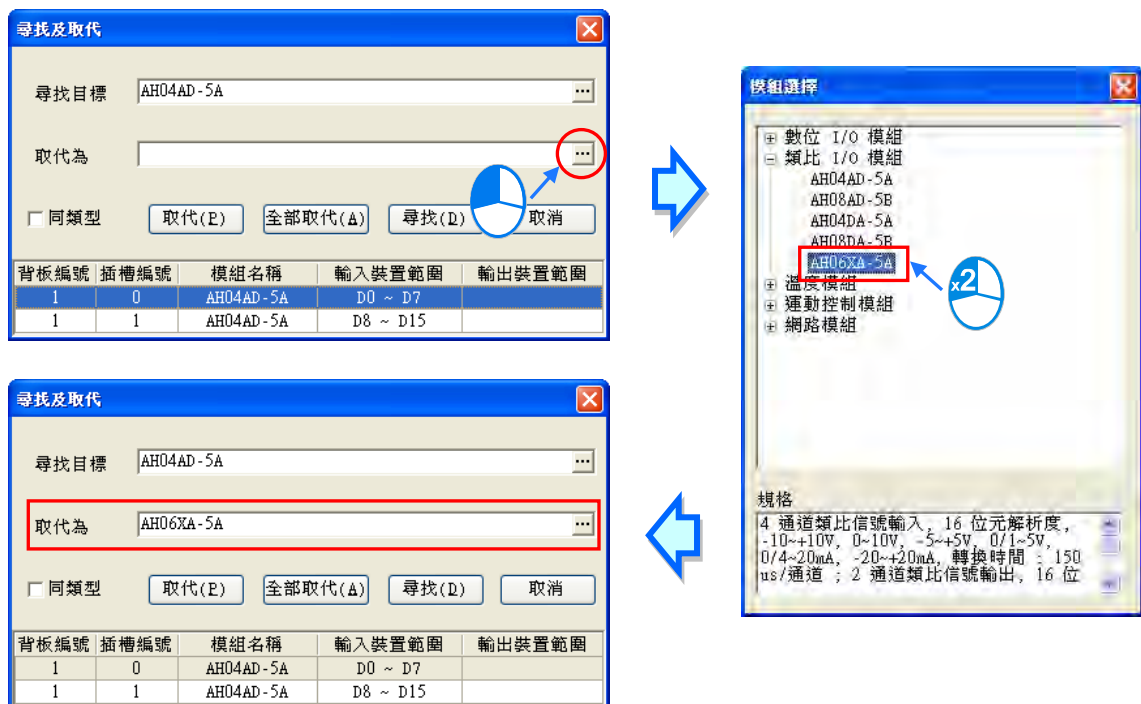
➤ 勾選同類型

僅將所選取的模組取代為同類型的其他模組，而取代之後，新模組的輸入/輸出裝置範圍將會與原本相同，而其他的參數設定若無法完全對應時將會被恢復為系統預設值。

➤ 不勾選同類型

可將所選的模組取代為任何類型的模組，而執行之後，其效果等同將原本的模組刪除後，再重新插入另一模組，因此新模組的輸入/輸出裝置範圍將會重新配置，而模組的其他參數設定也將會重新恢復為系統預設值。

- (5) 接著請點擊「取代為」欄位右側的按鈕，並於出現的選單中點選模組型號後，於該項目上雙擊滑鼠左鍵即可；且因取代的目標必須為一個確切的模組型號，所以此處不可使用輸入字串的方式。



*若「取代為」欄位保持空白時，則執行取代功能之後，所選的模組便會被刪除。

- (6) 設定完取代的條件後，按下「取代」按鈕便可將所選的項目取代為所設定的新模組；而完成取代之後，下方的搜尋列表便會立即刷新，且會自動選取下一個項目。

3

尋找及取代

尋找目標: AH04AD-5A

取代為: AH06XA-5A

同類型 **取代(E)** 全部取代(A) 尋找(F) 取消

背板編號	插槽編號	模組名稱	輸入裝置範圍	輸出裝置範圍
1	0	AH04AD-5A	D0 ~ D7	
1	1	AH04AD-5A	D8 ~ D15	

HWCONFIG 已經完成搜尋，並完成了 1 項的取代作業。

- (7) 若要執行**全部取代**時，請在設定完取代的條件後，按下「全部取代」按鈕，之後系統便會將列表中的所有項目取代為所設定的新模組。

尋找及取代

尋找目標: AH04AD-5A

取代為: AH06XA-5A

同類型 取代(E) **全部取代(A)** 尋找(F) 取消

背板編號	插槽編號	模組名稱	輸入裝置範圍	輸出裝置範圍
1	0	AH04AD-5A	D0 ~ D7	
1	1	AH04AD-5A	D8 ~ D15	

HWCONFIG 已經完成搜尋，並完成了 2 項的取代作業。

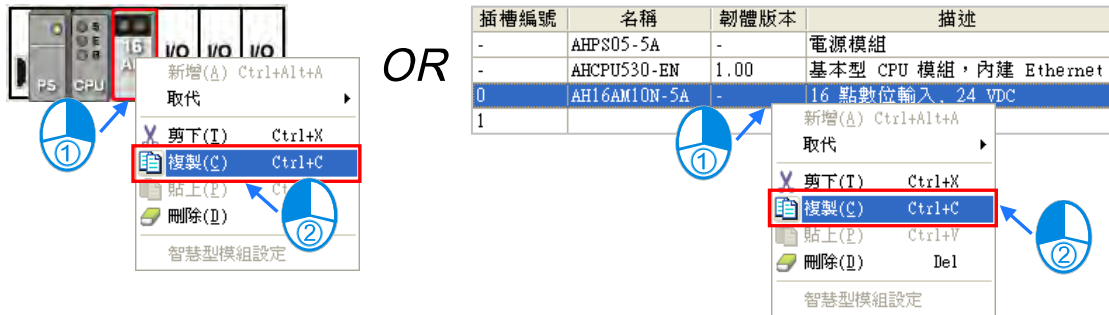
*.若是使用 AH560 備援系列機種，在開啟「備援功能」後，在執行搜尋/取代功能的取代功能，如需取代「系統配置區」備援主背板上的模組，模組只能在左側備援主背板上取代變更模組，而右側備援主背板上配置為左側備援主背板的鏡像映射，無法進行修改；而在執行搜尋時，右側備援主背板的配置不會列在搜尋列表中。

3.2.2.7 模組的複製/貼上

複製模組的方法有下列兩種 (CPU 與電源模組不可進行複製/貼上操作)：

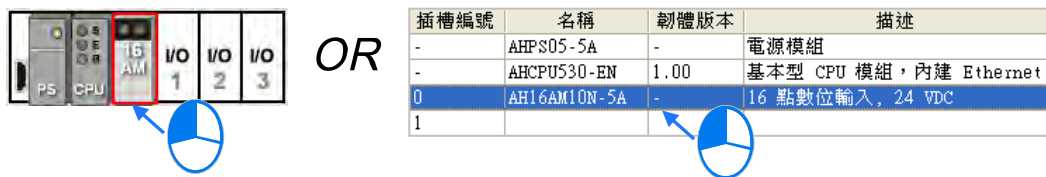
➤ 方法一


於系統配置區或資訊列表上點選欲複製的模組後按下滑鼠右鍵，之後於快捷選單中點選「複製(C)」。

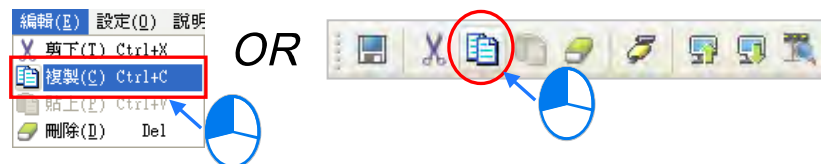


➤ 方法二

(1) 先於系統配置區或資訊列表上點選欲複製的模組。



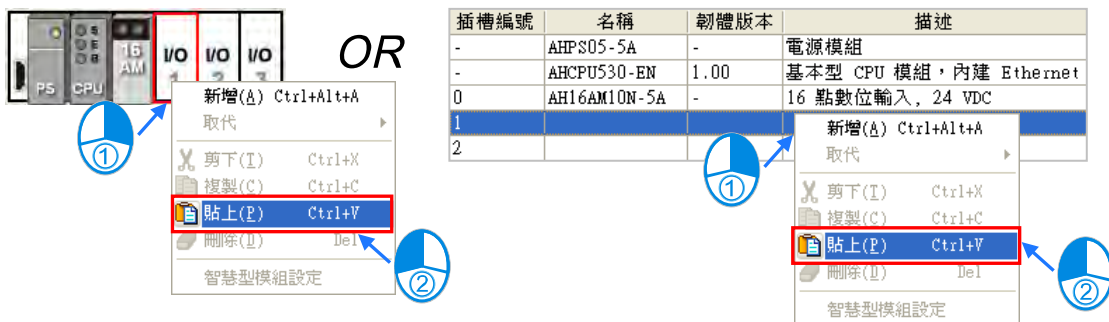
(2) 於 HWCONFIG 的功能工具列點選 編輯 (E) > 複製 (C) 或點擊圖示工具列的  圖示。



貼上模組的方法有下列兩種：

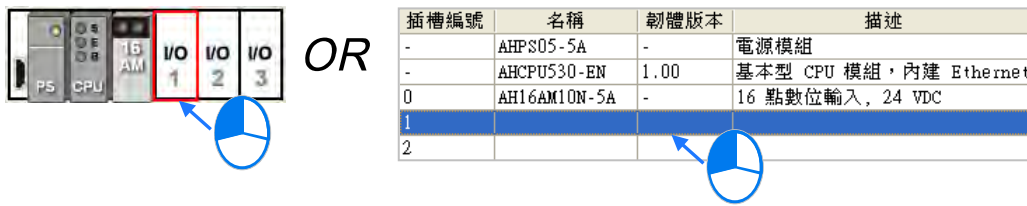
➤ 方法一

於系統配置區或資訊列表上點選欲貼上的插槽後按下滑鼠右鍵，之後於快捷選單中點選「貼上(P)」。

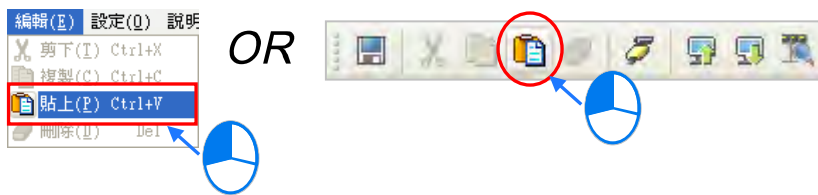


➤ 方法二

(1) 先於系統配置區或資訊列表上點選欲貼上的插槽。



(2) 於 HWCONFIG 的功能工具列點選 編輯 (E) > 貼上 (P) 或點擊圖示工具列的 圖示。



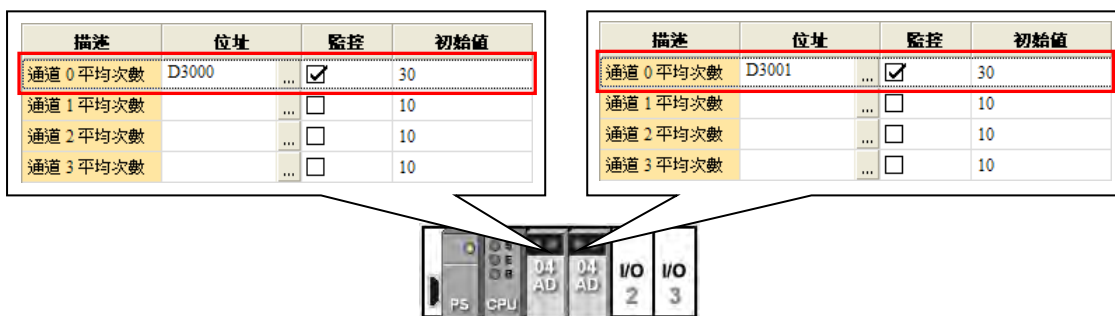
3

補充說明

當模組進行複製/貼上的操作時，模組參數的處理規則如下。

- 輸入/輸出裝置範圍：被貼上的模組會自動跳號至可用的裝置範圍。
- 模組註解：一併被複製至貼上的模組。
- 模組內部參數：一併被複製至貼上的模組。
- D 對應裝置：有指定的 D 對應裝置的欄位，於被貼上的模組內部，會自動跳號至可用的位址。
- 智慧型模組的進階參數：不會被複製而成為系統預設值，使用者須重新進行設定。

如下圖範例，其中插槽 0 為被複製的模組，而插槽 1 則為被貼上的模組。



插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHP505-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		AI
1	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D8 ~ D15		AI
2						

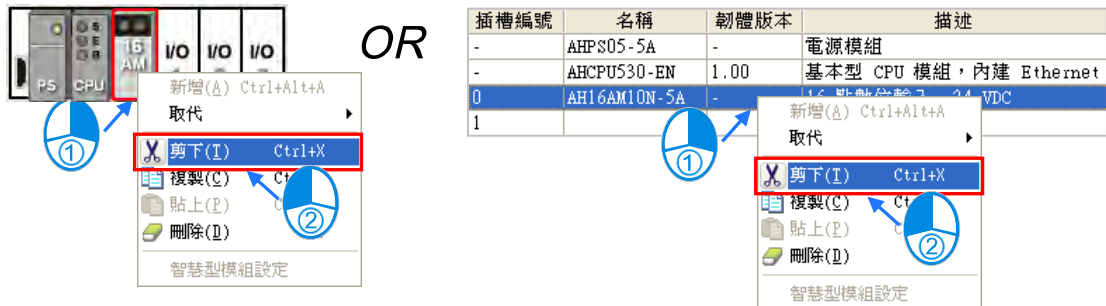
若使用者選擇的機種為 AH560 備援系列機種，在開啟「備援功能」後，如需複製/貼上「系統配置區」備援主背板上的模組，模組只能在左側備援主背板上做複製/貼上的動作，而右側備援主背板上配置為左側備援主背板的鏡像映射，無法進行修改。

3.2.2.8 模組的剪下/貼上

剪下模組的方法有下列兩種 (CPU 與電源模組不可進行剪下/貼上操作)：

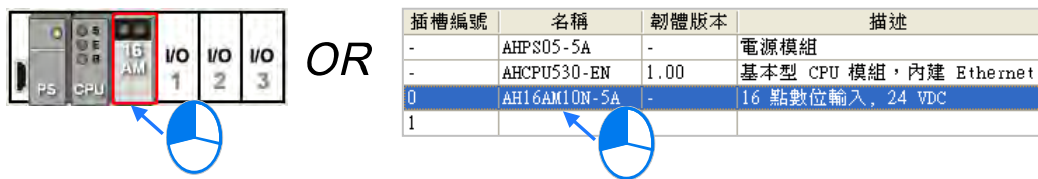
➤ 方法一

於系統配置區或資訊列表上點選欲剪下的模組後按下滑鼠右鍵，之後於快捷選單中點選「剪下(I)」。

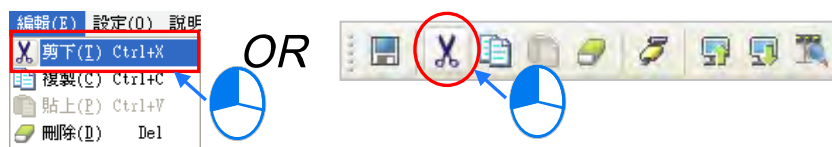


➤ 方法二

(1) 先於系統配置區或資訊列表上點選欲剪下的模組。



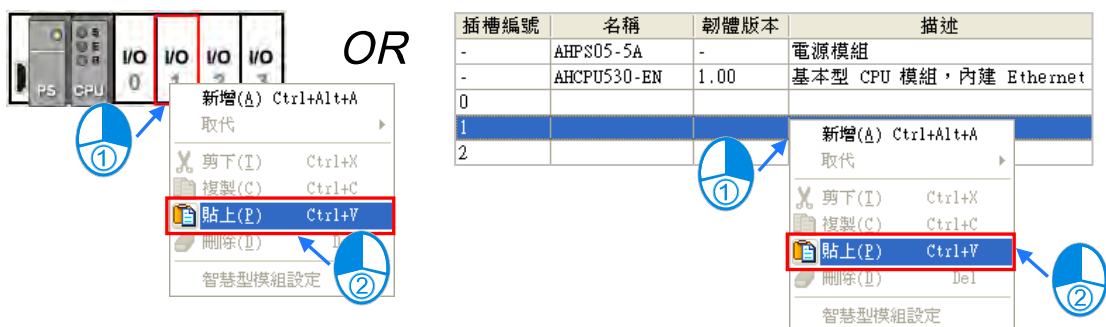
(2) 於 HWCONFIG 的功能工具列點選 編輯 (E) > 剪下 (I) 或點擊圖示工具列的 圖示。



貼上模組的方法有下列兩種：

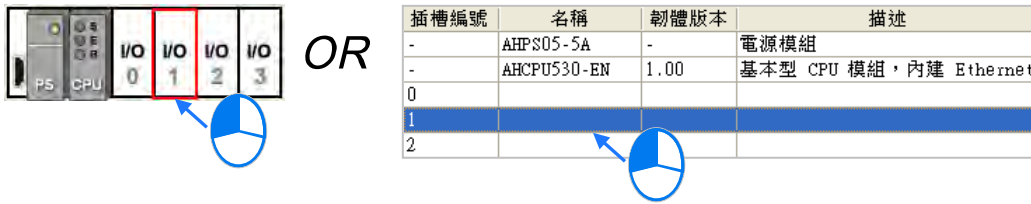
➤ 方法一

於系統配置區或資訊列表上點選欲貼上的插槽後按下滑鼠右鍵，之後於快捷選單中點選「貼上(P)」。



➤ 方法二

(1) 先於系統配置區或資訊列表上點選欲貼上的插槽。



(2) 於 HWCONFIG 的功能工具列點選 編輯 (E) > 貼上 (P) 或點擊圖示工具列的 圖示。



補充說明

模組進行剪下/貼上的操作時，每次剪下後僅可執行一次貼上的操作。而被剪下模組的輸入/輸出裝置範圍、模組註解、模組內部參數、D 對應裝置與智慧型模組的進階參數將會一併轉移至被貼上的模組。

若使用者選擇的機種為 AH560 備援系列機種，在開啟「備援功能」後，如需剪下/貼上「系統配置區」備援主背板上的模組，模組只能在左側備援主背板上做剪下/貼上的動作，而右側備援主背板上配置為左側備援主背板的鏡像映射，無法進行修改。

3.2.2.9 模組的搬移

除了 CPU 與電源模組之外，配置於系統配置區的模組圖示皆可直接使用滑鼠拖曳來進行搬移。

AH 系列機種的搬移有以下兩種狀況：

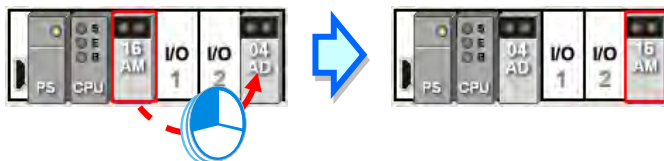
➤ 狀況一

選取模組後按住滑鼠左鍵，接著將其拖曳至空白插槽，之後該模組便會被搬移至新位置上。



➤ 狀況二

選取模組後按住滑鼠左鍵，接著將其拖曳至其他模組的位置，此時兩模組的位置便會互相交換。



補充說明

若使用者選擇的機種為 AH560 備援系列機種，在開啟「備援功能」後，如需搬移「系統配置區」備援主背板上的模組，模組只能在左側備援主背板上做搬移的動作，而右側備援主背板上配置為左側備援主背板的鏡像映射，無法進行修改。

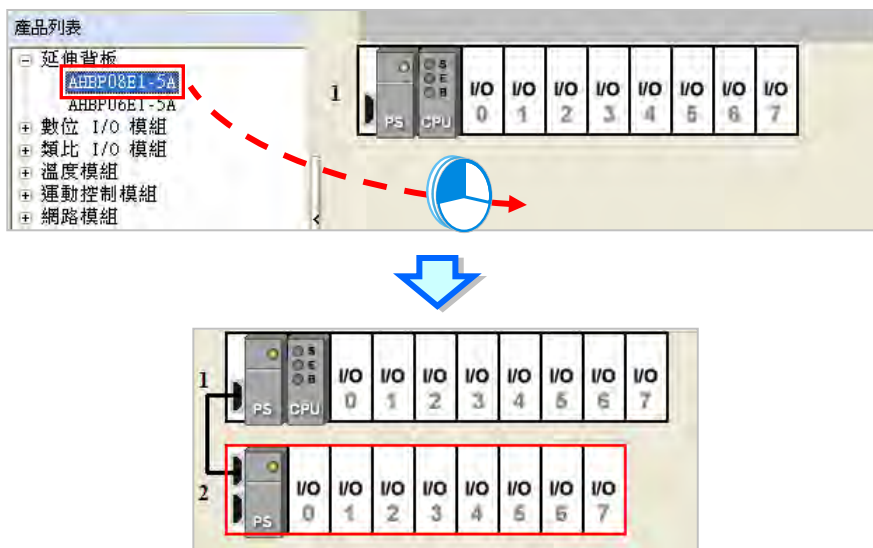
3

3.2.2.10 新增延伸背板及備援延伸背板

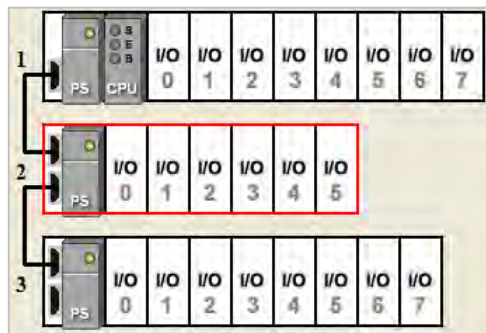
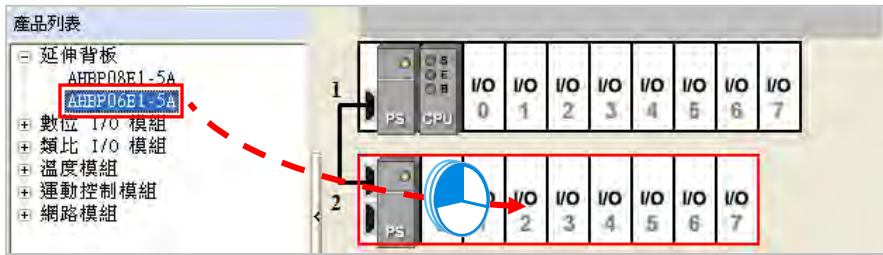
AH 系列機種新增延伸背板的方法有下列兩種：

➤ 方法一

於左側產品列表中選取欲加入的背板型號後，直接將其拖曳至系統配置區即可；而當將背板拖曳至空白區域放置時，新背板將會被放置在所有現有背板的下方。



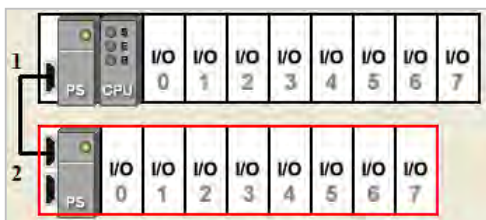
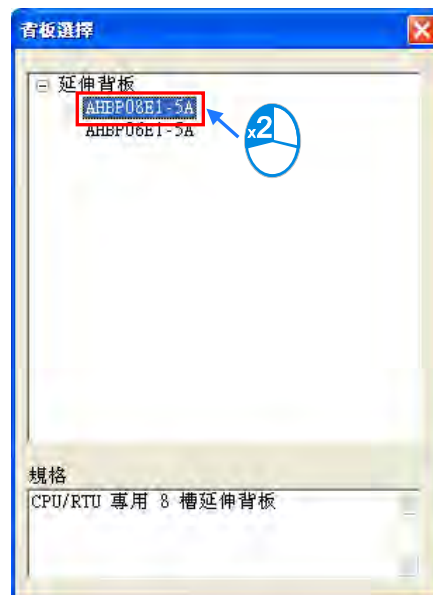
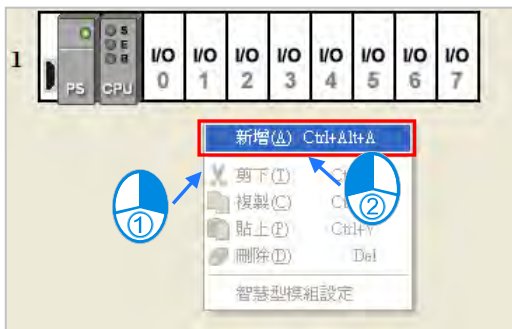
若是將背板拖曳至某個現有背板的位置上放置時，新背板則會被插於該背板的上方。



3

方法二

於系統配置區的空白處按下滑鼠右鍵，之後於快捷選單中選擇「新增 (A)」，接著於選擇視窗中選取欲加入的背板後，於該項目上雙擊滑鼠左鍵即可。此時新背板將會被放置在所有現有背板的下方。



點選某個背板圖示的左側後按下滑鼠右鍵，之後於快捷選單中選擇「新增 (A)」，接著於視窗中選取欲加入的背板後，於該項目上雙擊滑鼠左鍵即可。此時新背板將會被插於一開始所點選的背板下方。

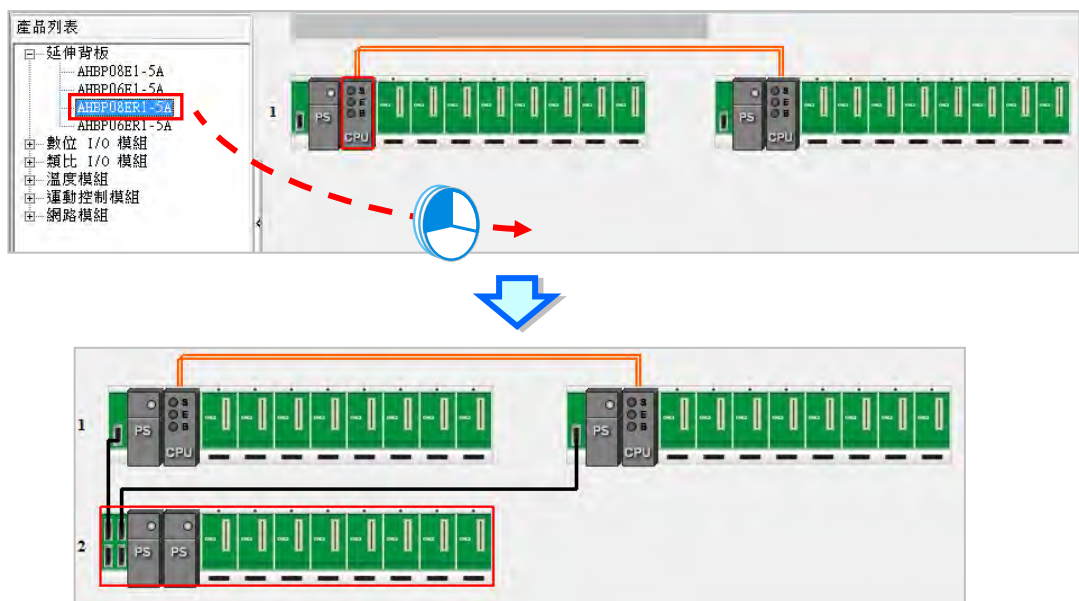


3

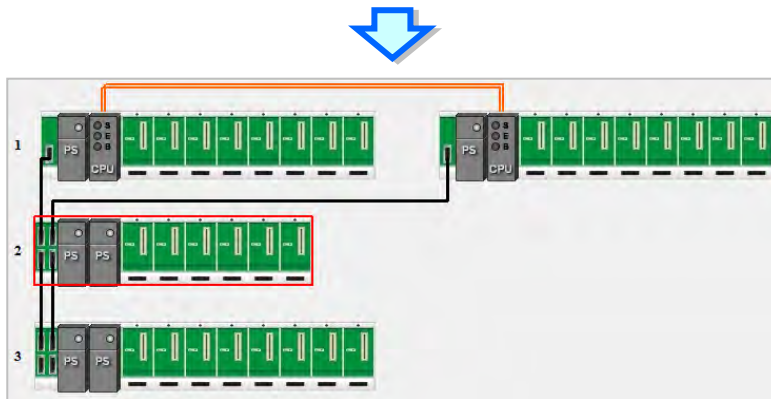
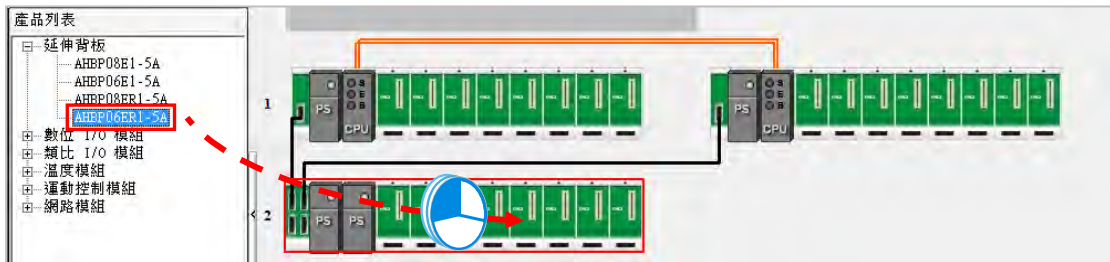
使用者若欲加入備援延伸背板，須注意 AH5x0 與 AH5x1 機種加入備援延伸背板時，備援延伸背板與延伸背板無法同時存在；若使用 AH560 備援系列機種，當使用者開啟「備援功能」，僅可新增備援延伸背板，關於備援延伸背板，詳細內容請參閱該產品相關手冊，這裡介紹新增延伸背板的方法，有下列兩種：

➤ 方法一

於左側產品列表中選取欲加入的備援延伸背板型號後，直接將其拖曳至系統配置區即可；而當將備援延伸背板拖曳至空白區域放置時，新備援延伸背板將會被放置在所有現有背板的下方。



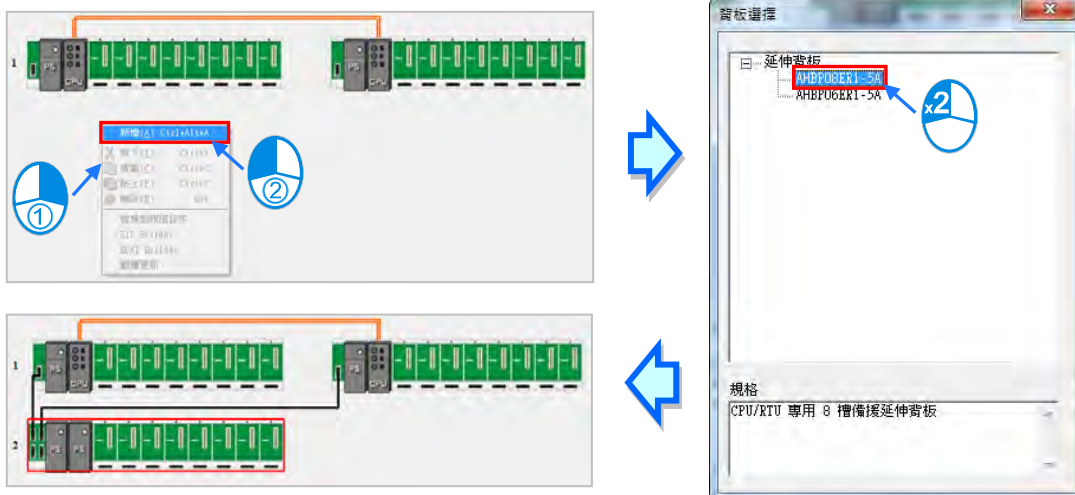
若是將備援延伸背板拖曳至某個現有備援延伸背板的位置上放置時，新備援延伸背板則會被插於該備援延伸背板的上方。



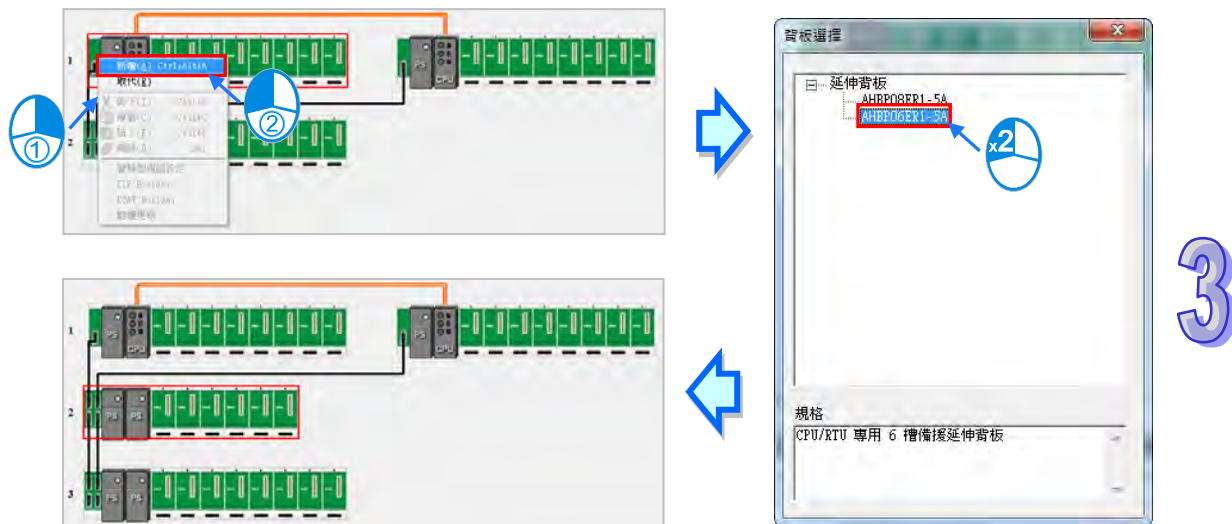
*備援延伸背板不得插於主背板及備援主背板的上方，且不能與延伸背板同時存在。

➤ 方法二

於系統配置區的空白處按下滑鼠右鍵，之後於快捷選單中選擇「新增 (A)」，接著於選擇視窗中選取欲加入的備援延伸背板後，於該項目上雙擊滑鼠左鍵即可。此時新備援延伸背板將會被放置在所有現有背板的下方。



點選某個背板圖示的左側後按下滑鼠右鍵，之後於快捷選單中選擇「新增 (A)」，接著於視窗中選取欲加入的備援延伸背板後，於該項目上雙擊滑鼠左鍵即可。此時新備援延伸背板將會被插於一開始所點選的背板下方。

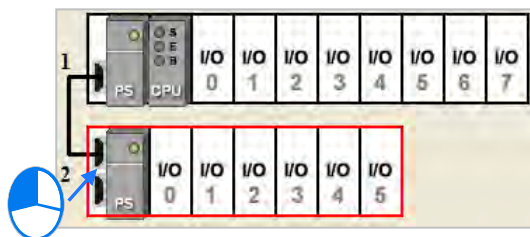



3.2.2.11 刪除延伸背板及備援延伸背板

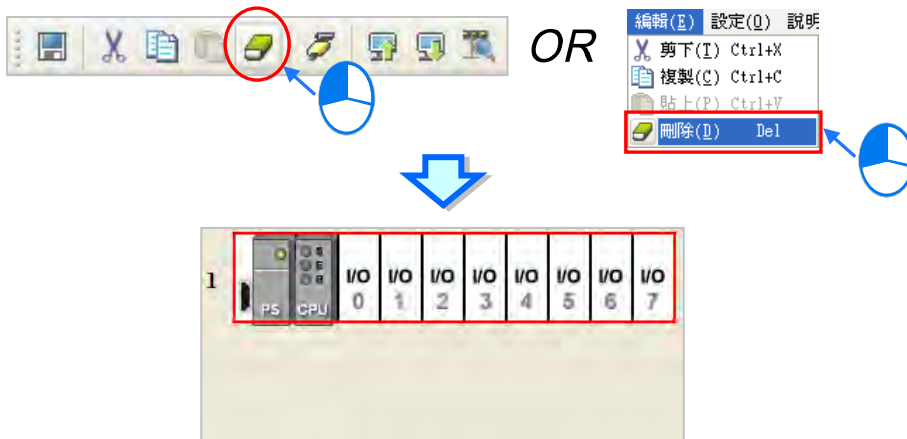
刪除延伸背板的方法有下列兩種 (主背板不可刪除)：

➤ 方法一

(1) 於欲刪除的背板圖示左側點擊滑鼠左鍵，以將該背板選取起來。



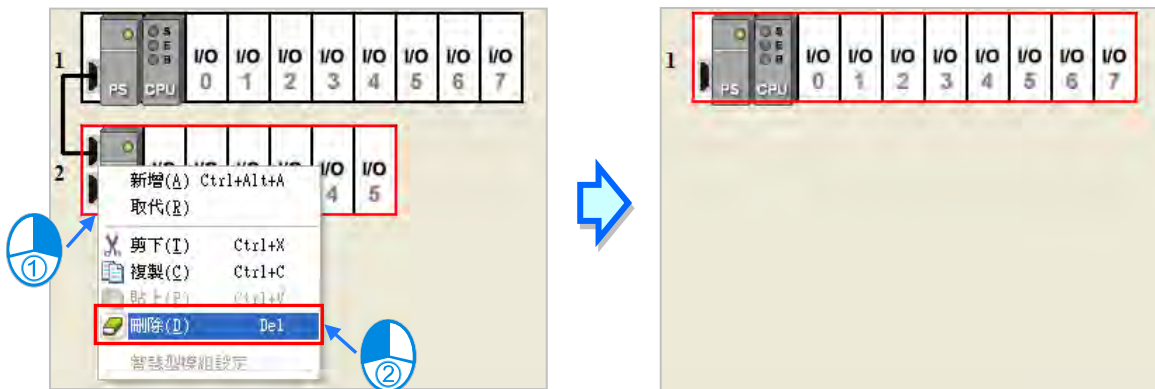
- (2) 於 HWCONFIG 的功能工具列點選 **編輯 (E)** > **刪除 (D)** 或點擊圖示工具列的  圖示，亦會是直接按下鍵盤的【DEL】鍵。



3

➤ 方法二

於欲刪除的背板圖示左側點擊滑鼠右鍵，之後於快捷選單中點選「刪除 (D)」。

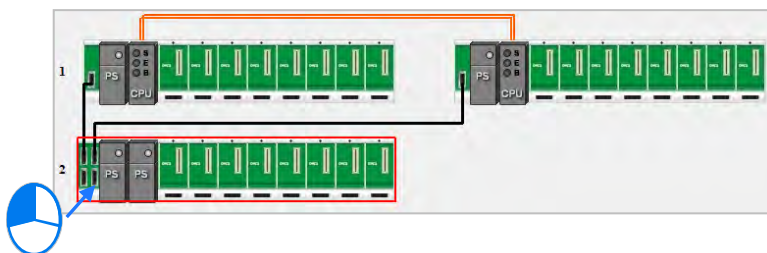



* AHxxEMC 系列機種不支援延伸背板相關功能。

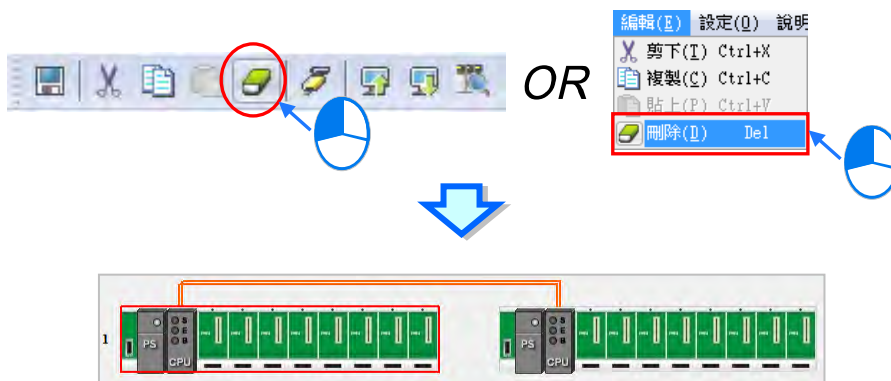
刪除備援延伸背板的方法有下列兩種（備援主背板不可刪除）：

➤ 方法一

- (1) 於欲刪除的備援延伸背板圖示左側點擊滑鼠左鍵，以將該備援延伸背板選取起來。

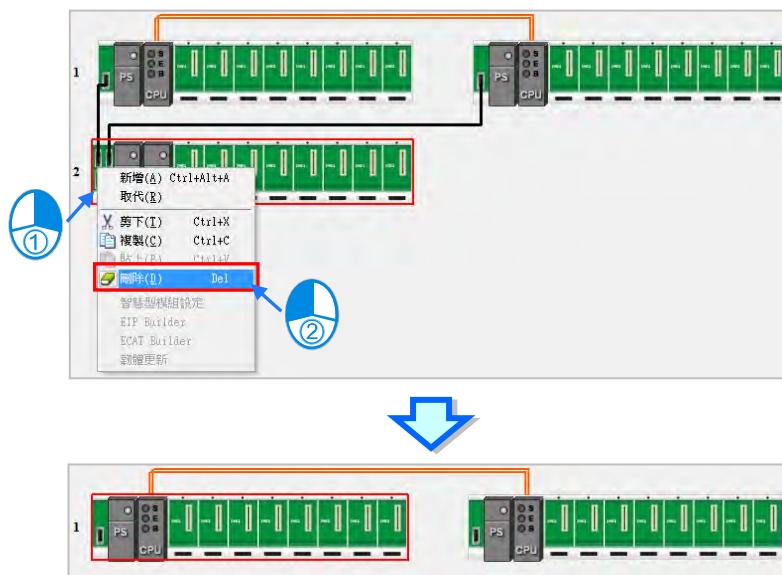


- (2) 於 HWCONFIG 的功能工具列點選 **編輯 (E)** > **刪除 (D)** 或點擊圖示工具列的  圖示，亦會是直接按下鍵盤的【DEL】鍵。



➤ 方法二

於欲刪除的備援延伸背板圖示左側點擊滑鼠右鍵，之後於快捷選單中點選「**刪除 (D)**」。

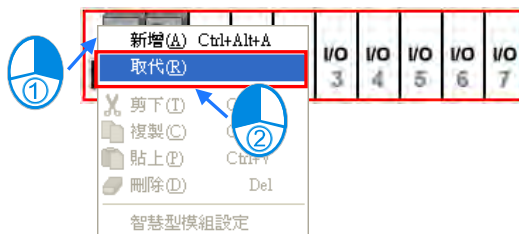


*AHxxEMC 系列機種不支援備援延伸背板相關功能。

3.2.2.12 變更背板

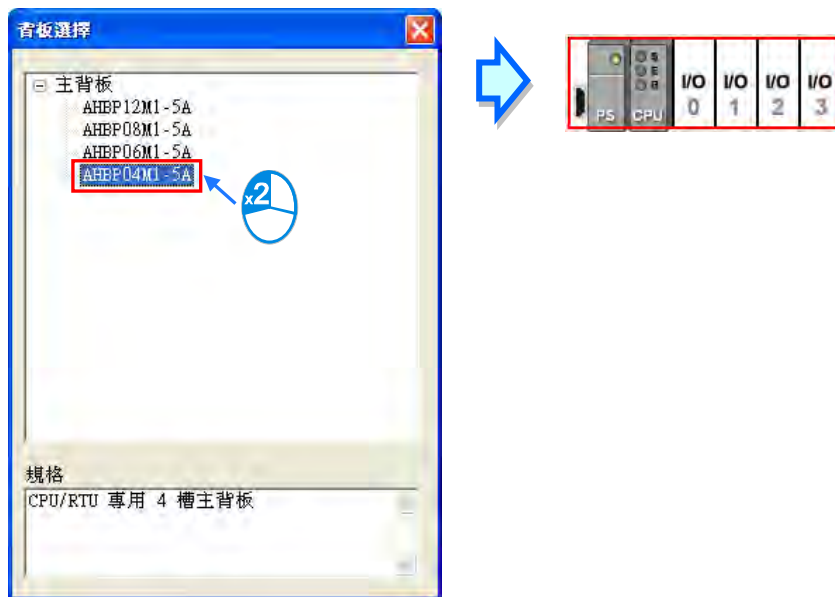
如欲變更背板可遵循以下步驟：

- (1) 於欲變更的背板圖示左側點擊滑鼠右鍵，之後於快捷選單中點選「**取代 (R)**」。



(2) 在選擇視窗中點選欲替換的背板型號後，於該項目上雙擊滑鼠左鍵。

3

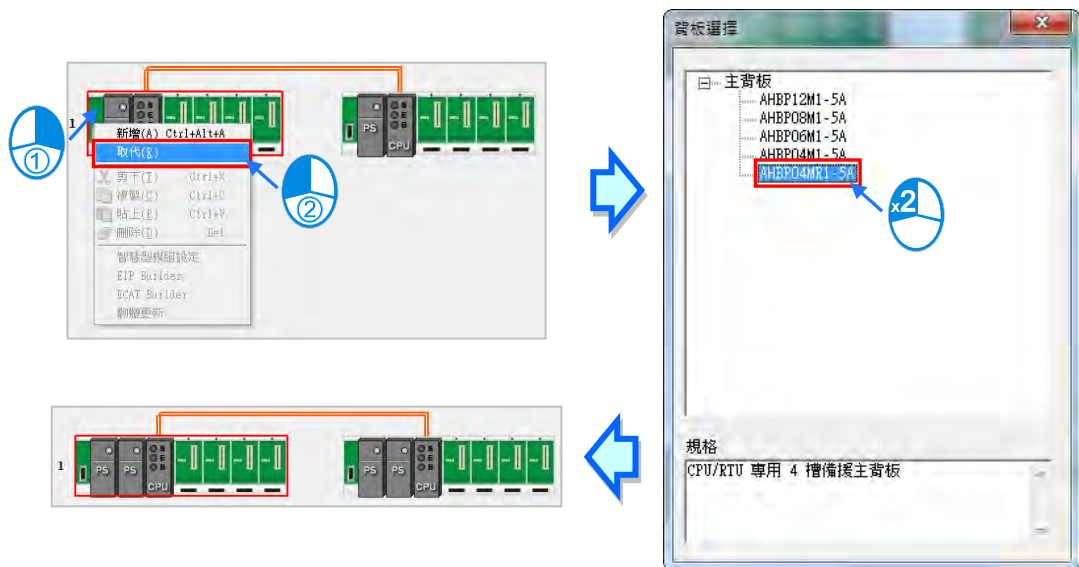


補充說明

當新背板的插槽數量小於原背板時，變更之後，原背板上超出新背板插槽編號的模組將會被自動刪除，因此變更前請務必小心確認。



若是選擇 AH560 備援系列機種，且開啟「備援功能」，欲需變更備援主背板，只能變更左側備援主背板，右側備援主背板為左側備援主背板的鏡像映射，無法進行修改。



3.2.2.13 延伸背板及備援延伸背板的剪下/複製/貼上

複製或剪下延伸背板的方法有下列兩種（主背板不可進行複製/剪下/貼上的操作）：

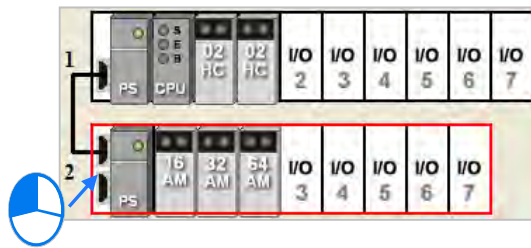
➤ 方法一

於欲複製或剪下的背板圖示左側點擊滑鼠右鍵，之後於快捷選單中點選「複製 (C)」或「剪下 (I)」。




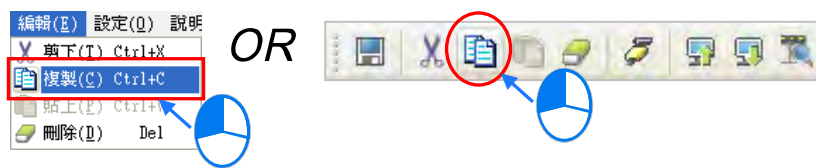
➤ 方法二


於欲複製或剪下的背板圖示左側點擊滑鼠左鍵，以將該背板選取起來。

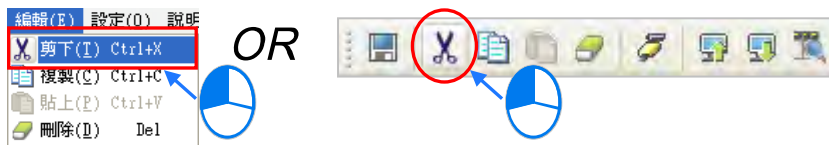


3

要執行複製操作時，請於功能工具列點選 編輯 (E) > 複製 (C) 或點擊圖示工具列的  圖示。



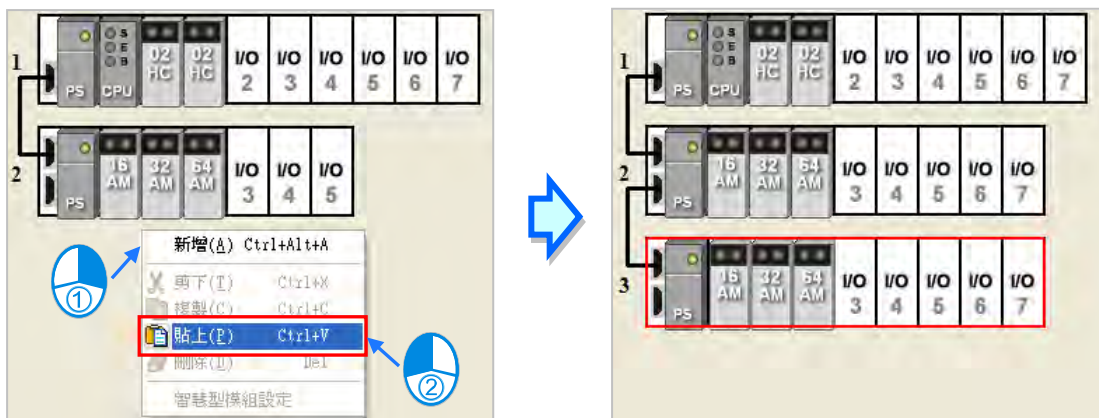
要執行剪下操作時，則請於功能工具列點選 編輯 (E) > 剪下 (I) 或點擊圖示工具列的  圖示。




貼上延伸背板的方法亦有下列兩種。而進行貼上操作時，若點選的貼上位置是系統配置區的空白處，則貼上的背板將會被放置在所有現有背板的下方；但若點選的貼上位置是某個現有的延伸背板，則貼上的背板將會被插於該現有背板的上方。

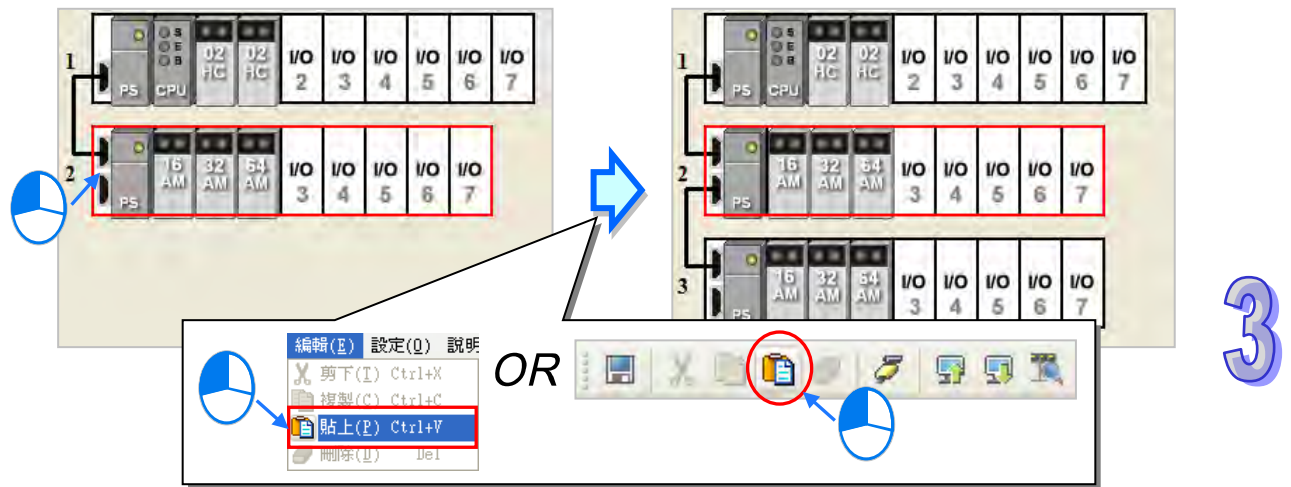
➤ 方法一

於系統配置區點選欲貼上的位置後按下滑鼠右鍵，之後於快捷選單中點選「貼上 (P)」。



➤ 方法二

先點選欲貼上的位置後，於功能工具列點選 **編輯 (E) > 貼上 (P)** 或點擊圖示工具列的  圖示。

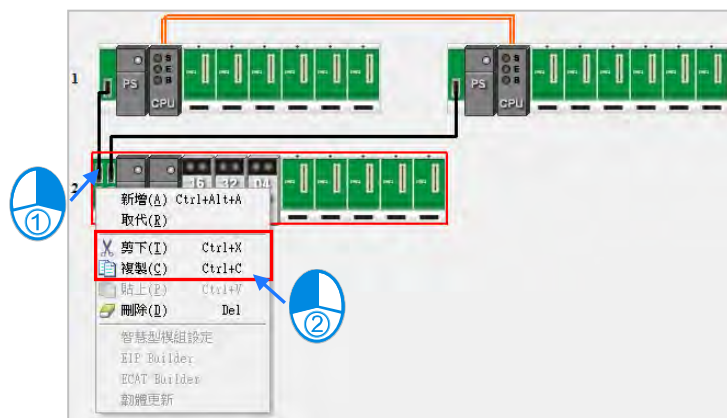


*AHxxEMC 系列機種不支援延伸背板相關功能。

複製或剪下備援延伸背板的方法有下列兩種（主背板不可進行複製/剪下/貼上的操作）：

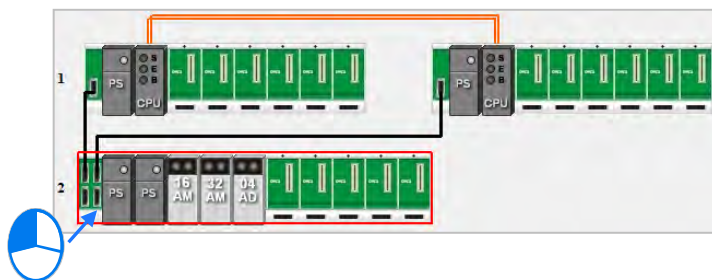
➤ 方法一


於欲複製或剪下的備援延伸背板圖示左側點擊滑鼠右鍵，之後於快捷選單中點選「複製 (C)」或「剪下 (I)」。

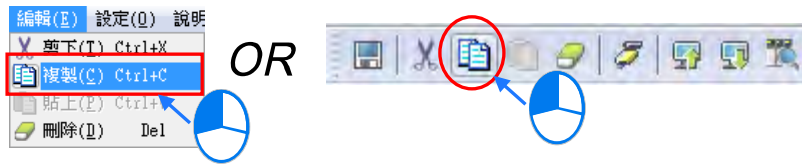



➤ 方法二

於欲複製或剪下的背板圖示左側點擊滑鼠左鍵，以將該備援延伸背板選取起來。



要執行複製操作時，請於功能工具列點選 **編輯 (E) > 複製 (C)** 或點擊圖示工具列的  圖示。



要執行剪下操作時，則請於功能工具列點選 **編輯 (E) > 剪下 (I)** 或點擊圖示工具列的  圖示。

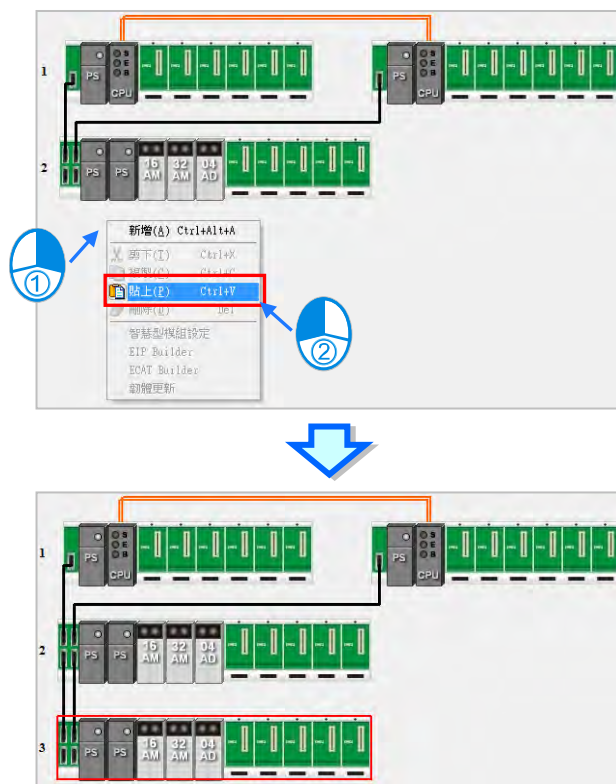


3


貼上備援延伸背板的方法亦有下列兩種。而進行貼上操作時，若點選的貼上位置是**系統配置區**的空白處，則貼上的備援延伸背板將會被放置在所有現有背板的下方；但若點選的貼上位置是某個現有的備援延伸背板，則貼上的備援延伸背板將會被插於該現有備援延伸背板的上方。

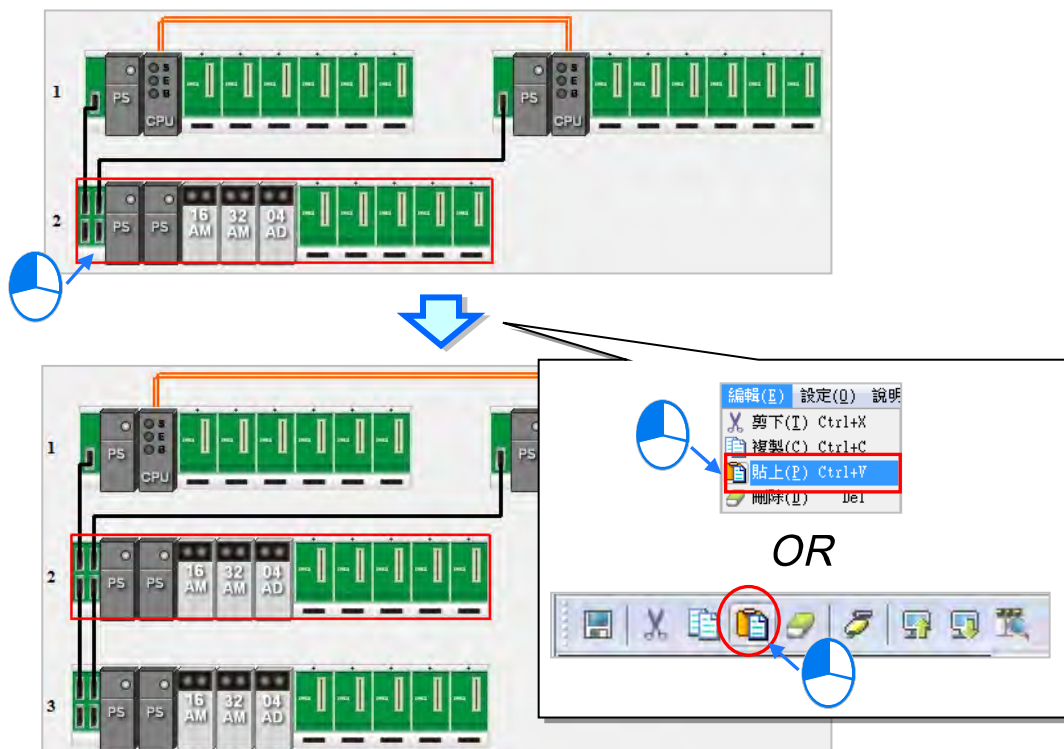
➤ 方法一

於**系統配置區**點選欲貼上的位置後按下滑鼠右鍵，之後於快捷選單中點選「**貼上 (P)**」。



方法二

先點選欲貼上的位置後，於功能工具列點選 **編輯 (E)** > **貼上 (P)** 或點擊圖示工具列的  圖示。



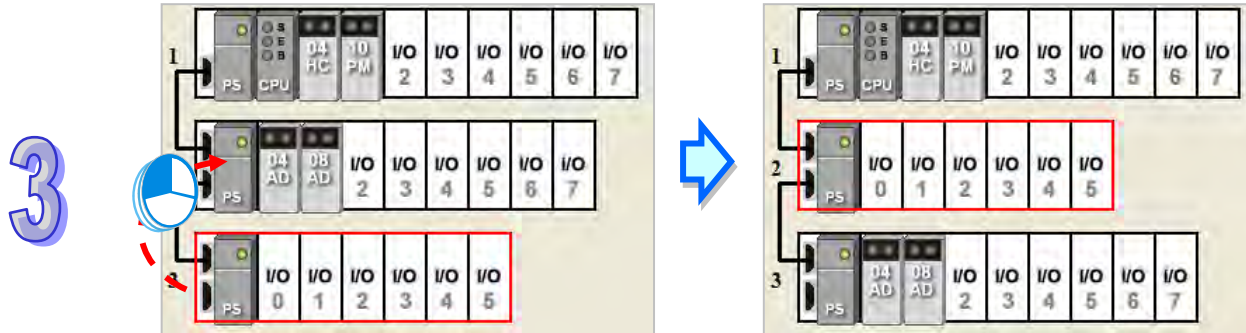
*AHxxEMC 系列機種不支援備援延伸背板相關功能。

補充說明

延伸背板、備援延伸背板在進行剪下/貼上時，被剪下背板上的所有模組的輸入/輸出裝置範圍、模組註解、模組內部參數、D 對應裝置與智慧型模組的進階參數將會一併轉移至被貼上的延伸背板、備援延伸背板；但進行的若是複製/貼上的操作時，被複製背板上的所有模組的模組註解與模組內部參數皆會一併被複製，而輸入/輸出裝置範圍與 D 對應裝置則會自動跳號，至於智慧型模組的進階參數則不會被複製而成為系統預設值。

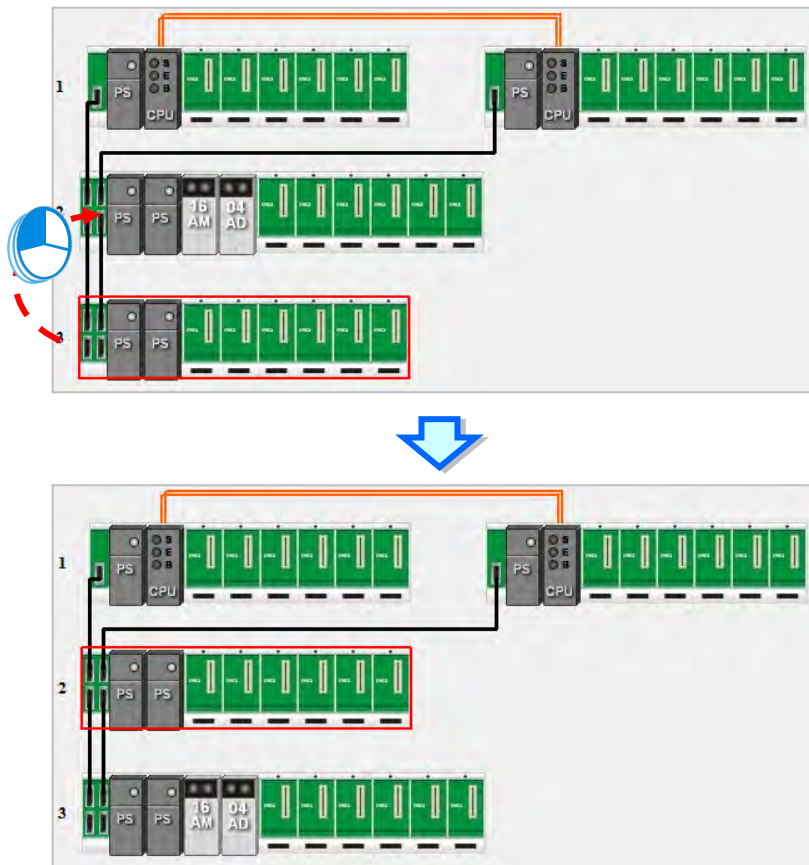
3.2.2.14 延伸背板圖示及備援延伸背板圖示的搬移

點選延伸背板的左側後按住滑鼠左鍵，接著將該背板拖曳至其他延伸背板的位置後放開，此時兩個背板便可交換位置，且延伸背板上所有模組的輸入/輸出裝置範圍、模組註解、模組內部參數、D 對應裝置與智慧型模組的進階參數亦將會跟隨模組而一併被搬移。不過操作時，主背板只能固定在最上方。



*AHxxEMC 系列機種不支援延伸背板相關功能。

點選備援延伸背板的左側後按住滑鼠左鍵，接著將該背板拖曳至其他延伸背板的位置後放開，此時兩個背板便可交換位置，且延伸背板上所有模組的輸入/輸出裝置範圍、模組註解、模組內部參數、D 對應裝置與智慧型模組的進階參數亦將會跟隨模組而一併被搬移。不過操作時，備援主背板只能固定在最上方。



*AHxxEMC 系列機種不支援備援延伸背板相關功能。

3.2.2.15 重新排序輸入/輸出裝置範圍

AH 系列機種在進行模組配置的過程中，可能會經過許多的搬移或是複製/剪下/貼上的操作，而此時各模組的輸入/輸出裝置範圍也可能已經經過系統的自動配置而不是按照連續的順序，此時便可於 HWCONFIG 的功能工具列中依序點擊 檔案 (F) > 輸入/輸出裝置重新排序，之後所有模組的輸入/輸出裝置範圍便會依據插槽及背板的編號順序而重新排序。

此操作並不會變動各模組內部所設定的 D 對應裝置，且重新排序時，若輸入/輸出裝置範圍的位址與 D 對應裝置衝突時，系統亦會自動跳過 D 對應裝置的位址，而以下一個可用的空間位址來進行編排。

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D64 ~ D71		
1	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D56 ~ D63		
2	AH04DA-5A	1.00	4 通道類比輸出 16 位元		D16 ~ D23	
3	AH04DA-5A	1.00	4 通道類比輸出 16 位元		D24 ~ D31	



插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
1	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D8 ~ D15		
2	AH04DA-5A	1.00	4 通道類比輸出 16 位元		D16 ~ D23	
3	AH04DA-5A	1.00	4 通道類比輸出 16 位元		D24 ~ D31	

3.2.3 主機與模組的版本管理

隨著主機與模組版本的更新，可使用的功能或參數可能也會有所不同，而在 HWCONFIG 中，提供了定義功能與參數的設備描述文件；當在安裝 ISPSOFT 時，最新版的設備描述文件便會一併被安裝至系統中。

因設備描述文件的版本會向下包含所有韌體版本的資訊，因此當我們在 HWCONFIG 當中進行配置時，便可依據實際設備的韌體版本而選擇適合的配置，且依據所配置的版本，可設定的參數也會有所不同；不過當發生於 HWCONFIG 中所設定的韌體版本與實際設備不符時，兩者之間的相容性與下載之後的狀態請參考下表的敘述。

版本差異	相容性說明
設定版本較實際韌體版本舊	設備仍可正常運作，而未於 HWCONFIG 中定義的模組功能或參數則會以系統預設的狀態參與運行。

版本差異	相容性說明
設定版本較實際韌體版本新	設備無法運作並進入錯誤狀態，此時使用者必須先於 HWCONFIG 中重新選擇符合實際模組的韌體版本，並在重新確認參數之後再次執行下載的程序。

*1.上述的相容性機制僅限 V1.0 版本以後的模組，V1.0 版以前的版本則無法與 V1.0 版或以上的版本相容。

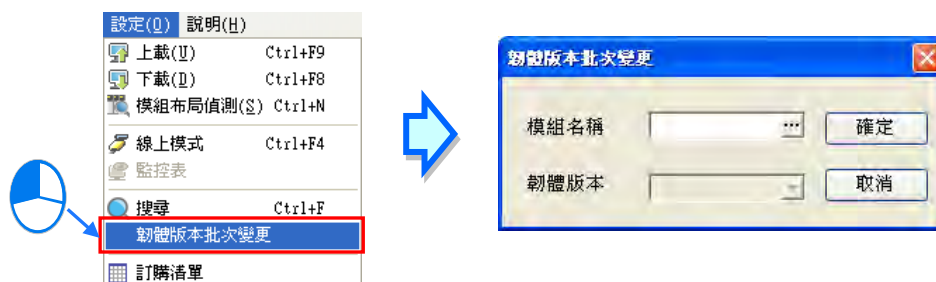
*2.依據機種的不同，設備描述檔案可能稱為「DDF」或「MDS」檔。

欲設定模組的韌體版本時，請直接於資訊列表的「韌體版本」欄位下拉選擇版本編號即可，而無韌體版本的模組於該欄位便不會有版本編號的顯示。

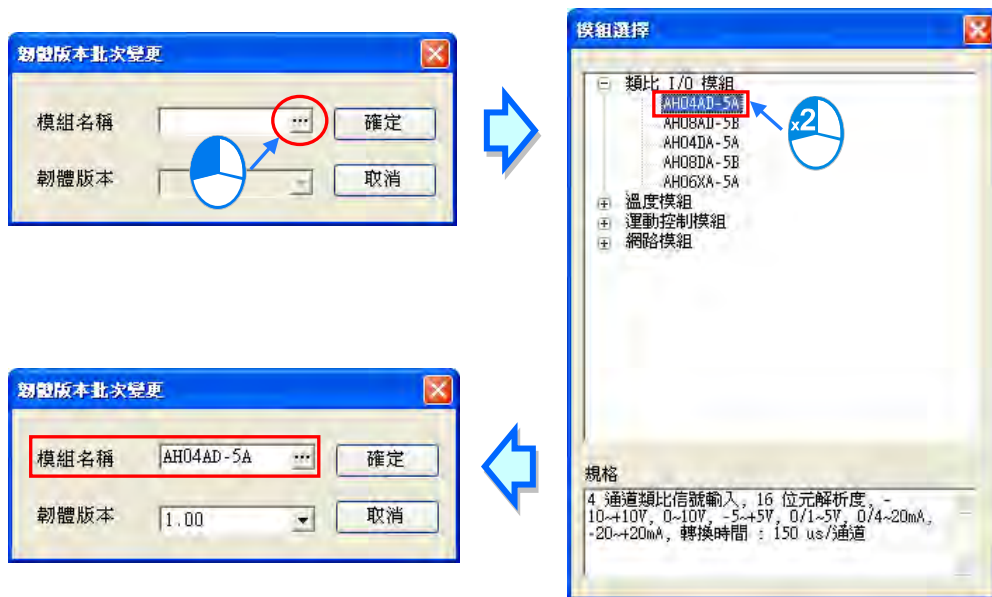
3

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道 16 位元	D0 ~ D7		
1	AH04AD-5A	0.38	4 通道 16 位元	D8 ~ D15		
2		0.41				
3		1.00				

當要批次變更某個型號的模組韌體版本時，請於功能工具列中點選 設定(O) > 韌體版本批次變更，接著畫面便會出現設定視窗。

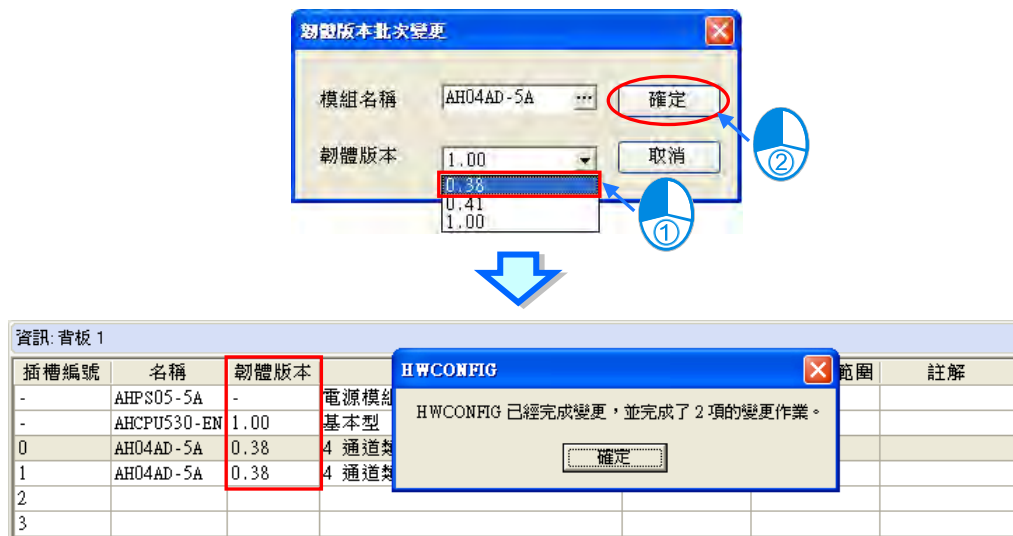


首先點擊「模組名稱」欄位右側的按鈕，並於隨後開啟的模組選擇視窗中選擇欲設定的模組型號，確認型號後於該項目上雙擊滑鼠左鍵即可。



3

接著請再於「韌體版本」的欄位下拉選擇欲替換的版本編號，而設定完成後，按下「確定」鍵便可將所有指定模組的韌體版本替換為所選的版本編號。



補充說明

欲獲知模組實際韌體版本的方法可透過模組布局偵測的功能，或是在連線的狀態下，藉由檢視模組資訊來得知；不過兩種方式皆必須在 ISPSOFT 可與主機正常連線，且實際模組也已完成安裝的狀態下。關於這兩個功能的詳細說明將分別於第 3.8.5 節與第 3.8.6 節的內容當中另行介紹。

3.3 AH500 機種的 PLC 主機參數設定

3.3.1 開啟 PLC 主機的參數設定頁面

啟動 HWCONFIG 之後，於系統配置區的 CPU 圖示上雙擊滑鼠左鍵，接著系統便會自動開啟 CPU 主機的參數設定頁面，且隨著主機型號的不同，其可設定的參數與顯示的畫面也會有所差異。

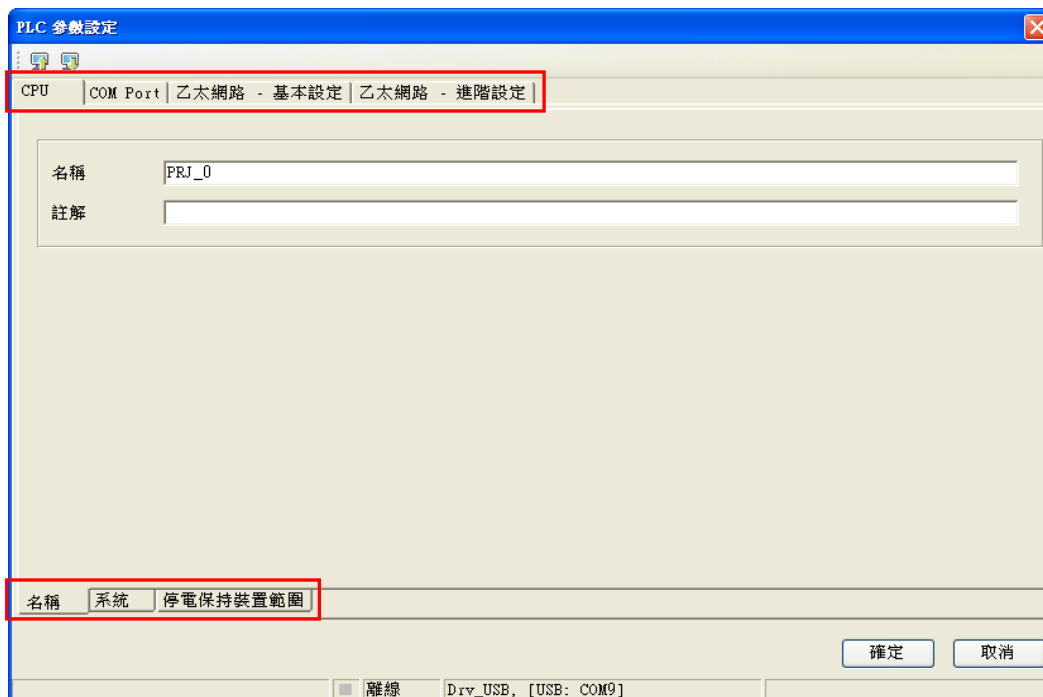
⚠ 設定主機參數前，請務必查閱各機種的操作手冊，並確認所設定的參數對主機本身以及整體系統的影響，以避免造成系統或人員的傷害。

3

*參數設定完畢之後，仍必須被下載至主機當中才可生效，下載的方式請參考第 3.8.4 節。



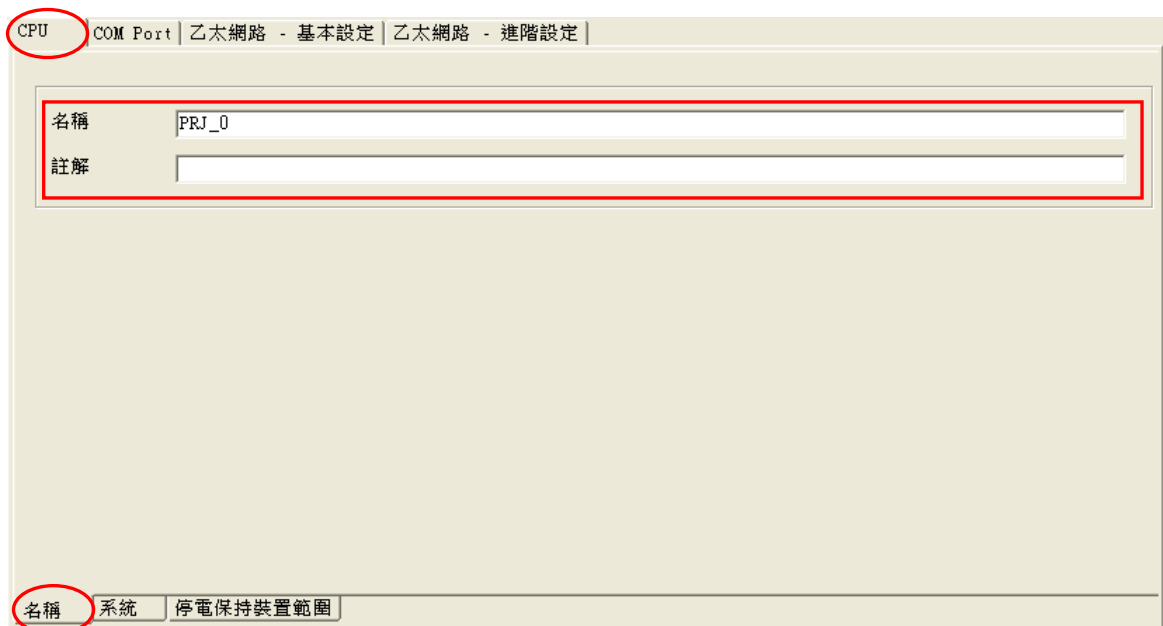
設定頁面中亦根據參數類別而區分為許多的分頁，操作時可配合視窗上方的主類別標籤以及位於每個主類別頁面下方的副類別標籤來切換各個參數設定的分頁。



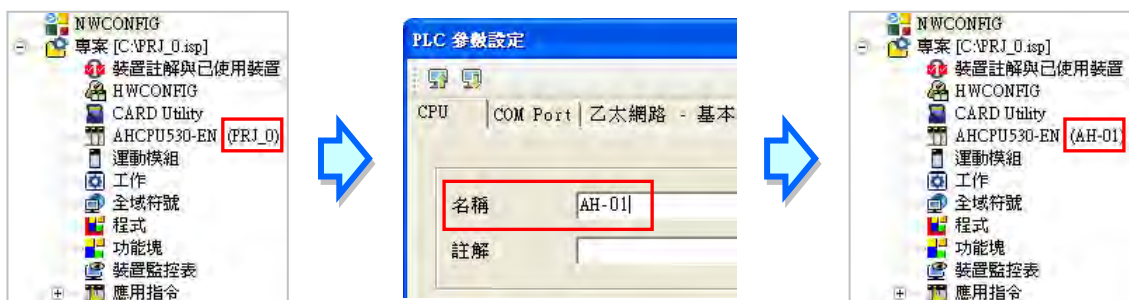
3.3.2 CPU 基本參數設定

3.3.2.1 CPU 基本參數 - 名稱

此類參數位於「**CPU**」主類別的「**名稱**」副類別頁面中，而此頁面只有「**名稱**」與「**註解**」兩個欄位，對於 AH5x0 系列機種，「**名稱**」欄最多可輸入 32 個字元，而「**註解**」欄最多可輸入 60 個字元；對於 AH5x1 系列機種或 AH560 備援系列機種，「**名稱**」欄最多可輸入 31 個字元，而「**註解**」欄最多可輸入 63 個字元，「**註解**」欄所輸入的字元並沒有特別的制限，包括特殊字元及空白鍵皆可輸入，不過須注意一個中文字將會佔用兩個字元。



當新增一個 ISPSOft 專案時，系統便會以專案的名稱作為主機的預設名稱，且會顯示於**專案管理區**的機種名稱後方，而後續則可於此處重新定義。



PLC 標籤名稱的功能主要是用來做為裝置識別，尤其是在聯網系統的應用或是在進行連線操作時，藉由確認 PLC 標籤名稱的方式便可防止使用者的誤操作；當使用者要對主機進行資料的上下載或其它的連線操作時，若發現操作對象的主機標籤名稱與專案中的標籤名稱不符時，系統便會提醒使用者做再次的確認，以避免對其他的主機造成非預期的影響。

3.3.2.2 CPU 基本參數 - 系統

此類參數位於「CPU」主類別的「系統」副類別頁面中，其功能主要是設定 CPU 本身的系統參數。

➤ 清除非停電保持區裝置 (STOP → RUN)

若此項設定被勾選，則當 PLC 主機的狀態由 STOP 切換至 RUN 時，非停電保持區的裝置狀態及數值將會被全部清除。

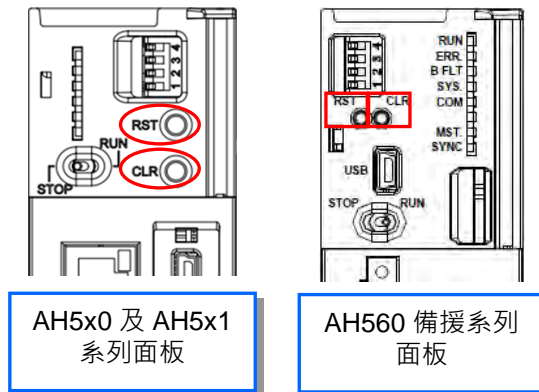
➤ Y 裝置狀態 (STOP → RUN)

設定主機狀態由 STOP 切換至 RUN 時的 Y 裝置狀態。

- 清除：將所有 Y 裝置的輸出都設為 OFF。
- 保持現在狀態：輸出將保持現狀。
- 回復停止前狀態：恢復上次切換為 STOP 前的狀態。

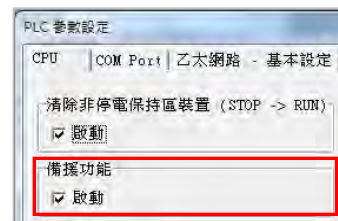
➤ 重置按鈕 & 清除按鈕

此參數主要用來設定 CPU 本體面板上的重置(RST)按鈕與清除(CLR)按鈕的功能是否被啟動。關於按鈕功能的說明與操作方式請參考各機種的操作手冊。

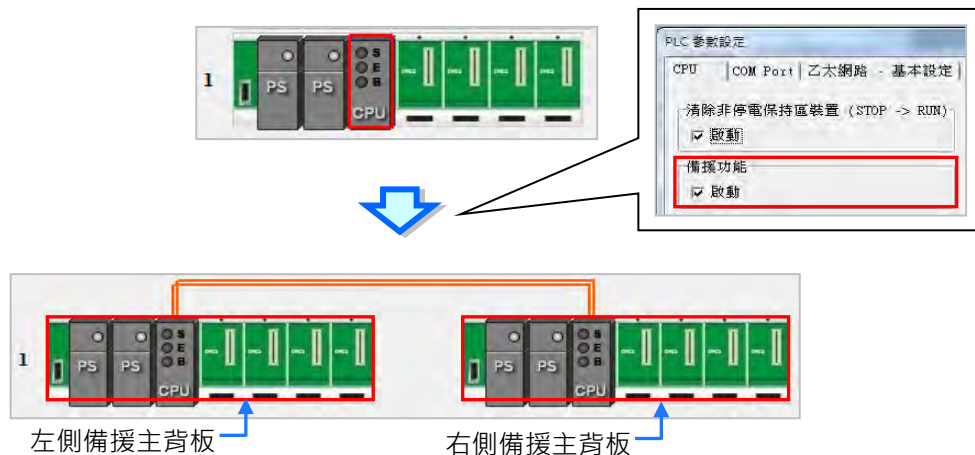


➤ 啟動備援功能

此功能啟動時，此時「系統配置區」會出現兩個備援主背板，如下圖，分別為「左側備援主背板」及「右側備援主背板」，而「右側備援主背板」為「左側備援主背板」的鏡像映射，是無法進行新增、刪除、變更模組等動作，若需進行此類動作，需於「左側備援主背板」進行。在啟動備援功能之後，「系統配置區」內所有延伸背板將自動改為備援延伸背板；備援主背板可配置的模組及備援功能詳細操作方式請參閱 **AH560 備援系列機種該產品之相關手冊**。(目前僅 AH560 備援系列機種支援啟動備援功能)

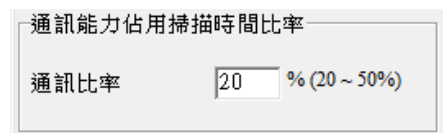


3



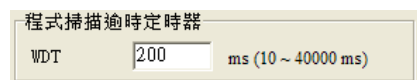
➤ 通訊能力佔用掃描時間比率

此功能啟動時，使用者可自行調控通訊能力佔用掃描時間的比率。(目前僅 AH5x1 及 AH560 備援系列機種支援此功能)



➤ 程式掃描逾時定時器 (WDT)

此參數用來設定程式掃描的逾時時間。一般情況下，主機系統會在每次掃描週期結束後自動重置 WDT，但若因某些原因而造成掃描程序無法在設定的時間內完成時，系統便會產生 WDT 逾時而讓主機進入錯誤狀態。關於錯誤處置的相關說明請參考各機種的操作手冊。



➤ 固定掃描時間

勾選是否固定最小的程序掃描時間，並於此處設定該時間的參數值。若此功能被啟動，則當實際的程序掃描時間小於設定值時，主機便會強制等待時間到達設定值後，才會開始下一次的掃描；而若實際的掃描時間大於此設定值，則主機便不理會此設定值而根據實際的掃描時間進行運作。除此之外，設定掃描時間時須特別注意，若設定的掃描時間大於**程式掃描逾時定時器 (WDT)** 的設定值時，一旦主機開始運行後，便必定會發生 WDT 逾時的錯誤。

☑ 啟動固定掃描時間
掃描時間 ms (1 ~ 32000 ms)

➤ 定時中斷時間設定

AH500 系列機種共提供 4 個定時中斷源，且可於此處設定每個中斷源的觸發時間間隔。請參考下圖，在 ISPSOft 中，定時中斷副程式的建立方式是採用 TASK 與 POU 的架構，相關說明將於第 5 章的內容當中有詳盡的介紹。

定時中斷時間設定

定時中斷 0	<input type="text" value="100"/>	ms (1 ~ 1000 ms)
定時中斷 1	<input type="text" value="40"/>	ms (1 ~ 1000 ms)
定時中斷 2	<input type="text" value="20"/>	ms (1 ~ 1000 ms)
定時中斷 3	<input type="text" value="10"/>	ms (1 ~ 1000 ms)

工作列表

- 外部中斷 (236)
- 外部中斷 (237)
- 外部中斷 (238)
- 外部中斷 (239)
- 外部中斷 (240)
- 外部中斷 (241)
- 外部中斷 (242)
- 外部中斷 (243)
- 外部中斷 (244)
- 外部中斷 (245)
- 外部中斷 (246)
- 外部中斷 (247)
- 外部中斷 (248)
- 外部中斷 (249)
- 外部中斷 (250)
- 外部中斷 (251)
- 定時中斷 0 (252)
- 定時中斷 1 (253)
- 定時中斷 2 (254)
- 定時中斷 3 (255)

工作屬性

中斷標籤 定時中斷 0 (252)

CPU 計時中斷
請至 HWCONFIG 的 CPU 系統頁面
設定

未指派 POU: []

已指派 POU: []

➤ 遠端執行功能

設定是否允許透過 IO 接點來切換主機的 RUN/STOP 狀態，並於此指定對應的 X 裝置位址。以右圖的設定為例，當輸入接點 X0.0 的狀態為 ON 時，主機將被切換至 RUN 狀態；而當 X0.0 為 OFF 時，主機則會被切換至 STOP。

☑ 啟動遠端執行功能
X , (X0.0 ~ X511.15)

➤ 固定通訊時間回應

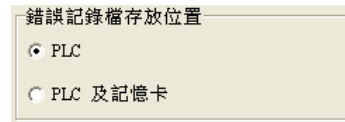
當未啟動此功能時，系統會在每次的掃描週期結束後，才會處理所有通訊埠的回應；而將此功能啟動後，主機便會每隔一段時間就會輪流處理各通訊埠的工作，因此可得到較即時的通訊回應；不過當系統處理通訊回應的工作時，會先將程序掃描的工作暫時中斷，因此啟用此功能將可能造成掃描時間的延長，請務必確認使用時不會造成系統運作的問題。

固定通訊時間回應
 啟動

➤ 錯誤記錄檔存放位置

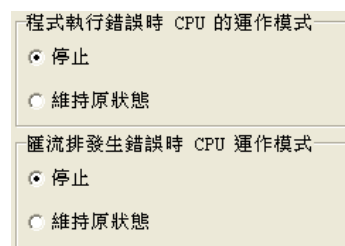
當設定為 **PLC** 時，主機的錯誤記錄將只被存放於主機內部的 **SR** 暫存器中，但若選擇 **PLC 及記憶卡** 時，記錄除了會存放於主機內部之外，亦會被備份於記憶卡之中。

AH500 系列機種的內部只能存放 20 筆錯誤記錄，當超過 20 筆時，最新的記錄將會直接覆蓋掉最舊的記錄；但若選擇的是 **PLC 及記憶卡**，則當主機內部的記錄超過 20 筆時，在覆蓋掉舊記錄之前，系統便會先將舊記錄備份於記憶卡中。



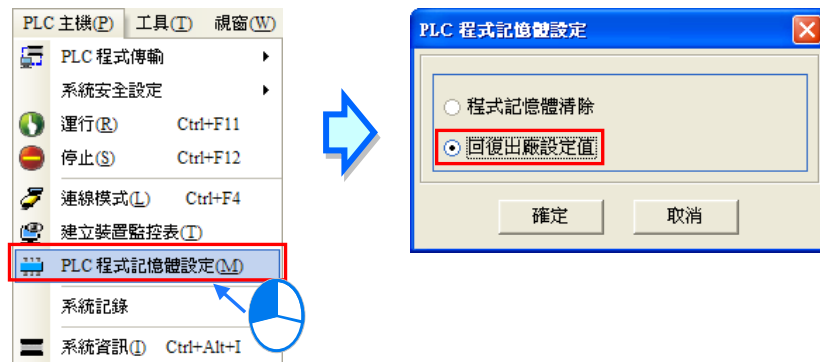
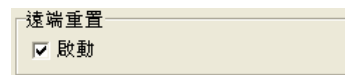
➤ 程式執行錯誤 & 匯流排發生錯誤的 CPU 運作模式

當主機在運行過程中發生錯誤時，根據所產生的錯誤碼，主機的運行狀態亦會有所差異，而其中有部分的錯誤則可由使用者自行定義主機的運作狀態；設定時，僅須於此處的欄位分別選擇發生錯誤時的狀態處理機制，並於後續確實將參數下載至主機當中即可。關於主機錯誤的處理機制請參考各機種的操作手冊或相關技術文件。



➤ 遠端重置

設定是否允許經由遠端系統來重置主機。以下圖 **ISPSOFT** 為例，必須先勾選此功能，並確實將參數下載至主機之後，於 **ISPSOFT** 中才可執行將主機恢復為出廠預設值的功能。

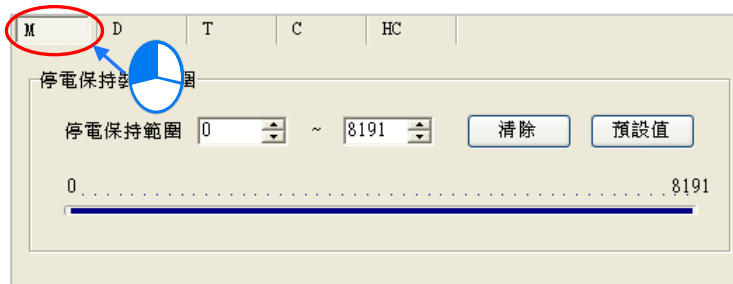


3.3.2.3 CPU 基本參數 - 停電保持裝置範圍

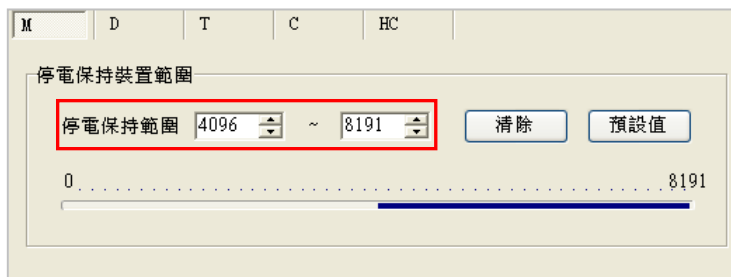
此類參數位於「CPU」主類別的「停電保持裝置範圍」的副類別頁面中，主要的功能在於規劃各裝置的停電保持區範圍。



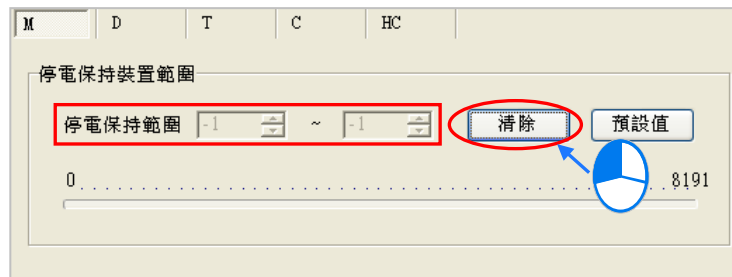
- (1) 用滑鼠點選欲設定的裝置類別便可切換至對應的頁面，而在設定區域的下方會顯示一個條狀圖，其兩側的數值則代表可設定的範圍極限，至於內部藍色的部份則代表已設為停電保持的範圍。



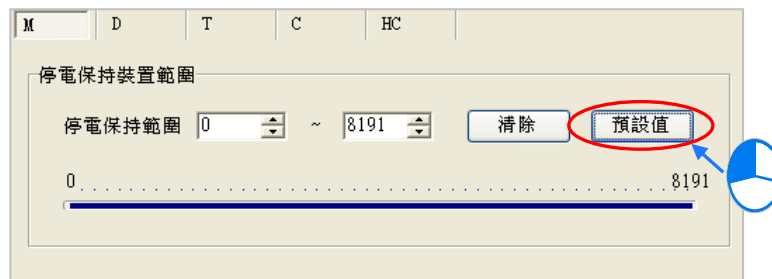
- (2) 設定時請直接於設定區域的兩個欄位中輸入範圍的起訖位址，其中左側為範圍起點，而右側則為範圍終點；輸入數值時，左側欄位的數值必須小於右側，且不可超出條狀圖所顯示的範圍極限。



- (3) 若想讓該裝置的所有範圍都不具備**停電保持**的功能時，只要按下設定區域中的「**清除**」鍵即可，而此時的範圍欄位便會變為灰階，並顯示為 $-1 \sim -1$ 。



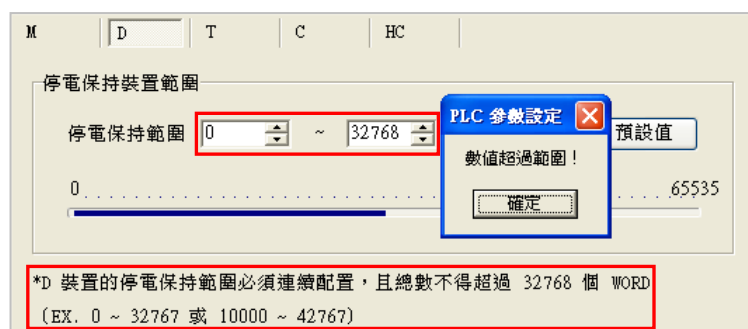
- (4) 若想讓目前所選裝置的停電保持範圍恢復為系統預設值時，請按下「**預設值**」按鈕即可。



補充說明

部分裝置的停電保持區域並非涵蓋整個範圍，當有最大數量的限制時，於設定區域的下方便會顯示相關的提示。下圖則為 AHCPU530-EN 機種中，D 裝置停電保持範圍的設定頁面；由提示可知，允許設為停電保持範圍的總數只有 32768 個 WORD，這代表使用者在整段可設範圍的 0 ~ 65535 之中，必須將停電保持區設為任何小於或等於 32768 的範圍，例如 0 ~ 32767，或 10000 ~ 42767。

以下圖為例，圖中所輸入的裝置範圍為 0 ~ 32768，而長度已經達到 32769 個 WORD，因此要離開該頁面時，系統便會出現**數值超過範圍**的警訊。



3.3.2.4 CPU 進階設定

此頁面可以設定停止 → 運行初始值設定。



- **非停電保持符號初始值生效**

選取啟動時，則 PLC 由停止狀態變為運行狀態時，非停電保持符號的值會恢復為初始值。

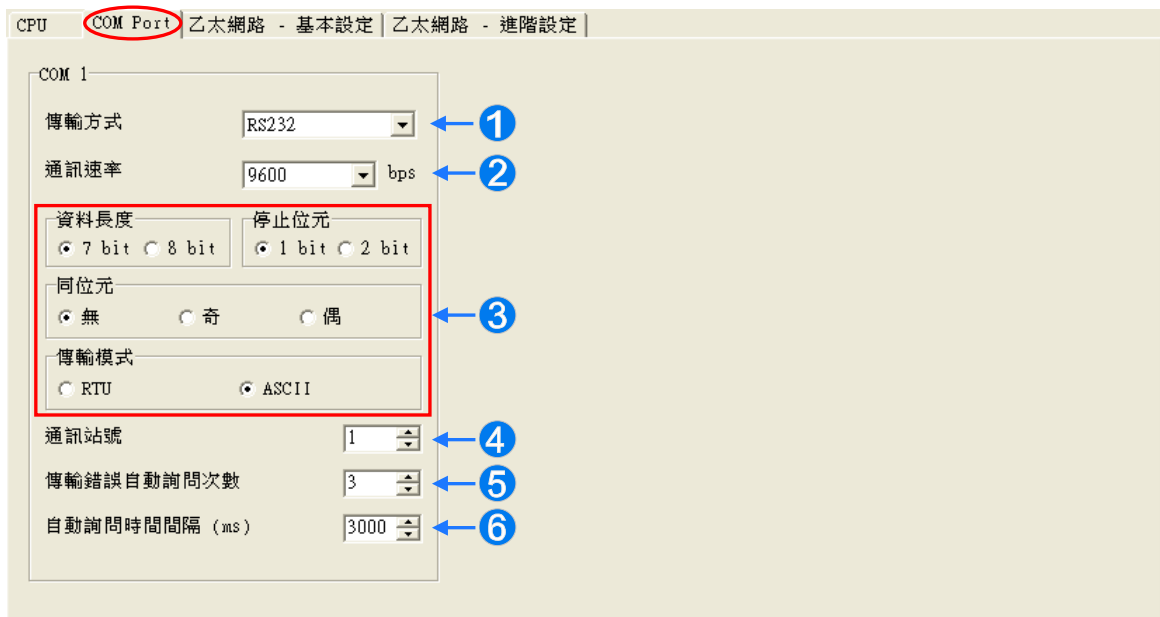
- **停電保持符號初始值生效**

選取啟動時，則 PLC 由停止狀態變為運行狀態時，停電保持符號的值會恢復為初始值。

3.3.3 序列埠 (COM Port) 參數設定

於主類別標籤中點選「**COM Port**」項目即可進入序列埠 (COM Port) 參數的設定畫面，而當所設定的機種本身具有兩個 COM Port 時，如 AHCPU5xx-RS2，此頁面便會有兩個相同的 COM Port 設定區域，且兩個 COM Port 之間的設定都是彼此獨立的。

關於透過序列埠 (COM Port) 來與主機進行連線的相關注意事項請參考第 2.4.8 節當中的說明。



- ❶ 設定此通信埠所採用的傳輸方式，選項有 RS232 / RS485 / RS422。
- ❷ 設定此通信埠所採用的通訊速率 (Baud Rate)。當選用的傳輸方式為 RS232 時，其通訊速率有 9600 / 19200 / 38400 / 57600 / 115200...等選項，而當所選的傳輸方式為 RS485 或 RS422 時，其通訊速率則會包括 9600 / 19200 / 38400 / 57600 / 115200 / 230400 / 460800 / 921600。
- ❸ 設定其他通訊協定參數，其中當**傳輸模式**為 RTU 時，**資料長度**只能強迫設定為 8 bit。
- ❹ 設定此通訊埠的通訊站號，以做為網路通訊上的裝置識別，而在同一條網路上的站號則不允許重複，其設定範圍為 0 ~ 247。

當該埠口扮演從站 (Slave) 的角色，且所連接的網路上亦有其它的從站時，此埠口的站號便不可設定為 0，因站號 0 在通訊協定中具有**廣播通訊**的使用意義。當主站 (Master) 在資料封包中指定將資料傳送給站號 0 的從站 (Slave) 時，這代表是要將資料傳送給所有的從站，而所有的從站，不論其本身的站號為何都會接受這筆資料封包。

- ❺ 當通訊失敗時，主機便會重試連線，而重試的次數便可於此欄位中進行設定，範圍則為 0 ~ 20。
- ❻ 於此可設定通訊失敗時的重試時間間隔。當 PLC 主機因通訊失敗而嘗試重新連線時，便會在間隔此設定時間後才會進行下一次的連線動作。可設定的範圍為 100~65535 ms。

3.3.4 乙太網路 (Ethernet) 基本參數設定

此參數頁面必須選用內建乙太網路埠口的機種才可進入，如 AHCPU5xx-EN，設定時於主類別標籤中點選「乙太網路 - 基本設定」項目即可進入。此處主要用於設定主機乙太網路 (Ethernet) 埠的通訊參數，關於透過 Ethernet 來與主機進行連線的相關注意事項請參考第 2.4.8 節當中的說明，而關於乙太網路的概念，請自行參考其他相關文件或書籍，此處將不再多做介紹。

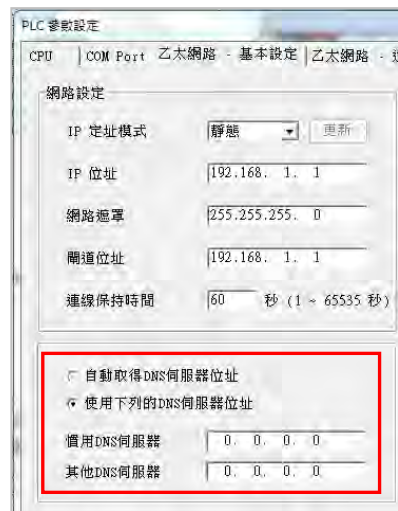
3

- ❶ 設定主機 IP 的定址模式。若設為「靜態」，代表 IP 位址等參數將由使用者直接指定；，但若設為「DHCP」或「BOOTP」，則代表 IP 位址將經由 DHCP/BOOTP 來動態分配。
- ❷ 當定址模式選擇「靜態」時，於此處便可設定主機的 IP 位址、網路遮罩及閘道位址。
- ❸ 當主機在網路上沒有進行任何資料傳遞，且已超過連線保持時間時，主機便會自動中斷連線。

當將定址模式設為「動態」時，依據下列的步驟進行操作便可讀回由 DHCP 所配置的位址參數。

- (1) 將主機連上乙太網路，並確認網路與主機皆可正常運作。
- (2) 定址模式設為「動態」，並將參數正確的下載至主機中。
- (3) 按下右側的「更新」鍵後，系統便會讀回主機內部的位址配置，並填入下方的欄位中。

若使用者是使用 AH5x1 或 AH560 備援系列機種，則可使用 DNS 伺服器位址功能，使用者可使用自動取得 DNS 伺服器位址，或使用自行指定的慣用 DNS 伺服器位址或其他 DNS 伺服器位址。



3

3.3.5 乙太網路 (Ethernet) 進階參數設定

若所設定的主機為內建乙太網路埠的機種，如 AHCPU5xx-EN，畫面中便會有乙太網路的頁面標籤；於主類別標籤中點選「乙太網路 - 進階設定」項目，之後便可進入網路進階功能的設定頁面。

3.3.5.1 乙太網路進階參數 - 過濾器

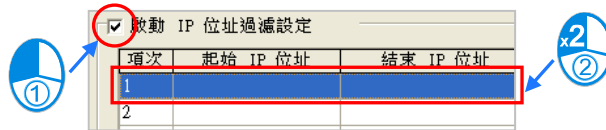
此類參數位於「乙太網路 - 進階設定」主類別的「過濾器」副類別頁面中。主要在於設定網路裝置的過濾功能，而透過此處的設定便可多加一層的確證機制，以確保通訊的對象為我們所允許的裝置。



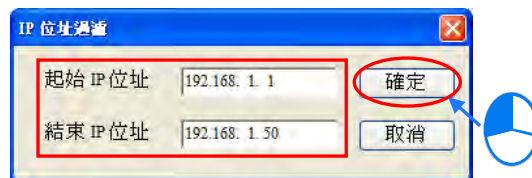
➤ IP 位址過濾

此功能可用以限定通訊對象。當此功能被啟用後，只有 IP 位址列於清單範圍中的裝置才允許跟主機進行通訊，至於其餘 IP 位址的裝置，其所傳送進來的資料封包，主機將會直接將其丟棄。此功能的設定步驟如下，而清單中最多可設定 16 組的位址範圍。

- (1) 勾選「**啟動 IP 位址過濾設定**」，之後於欲修改或新增過濾範圍的行列上雙擊滑鼠左鍵。



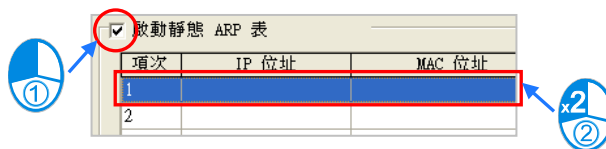
- (2) 於設定視窗中輸入位址範圍，而輸入時，起始位址必須小於結束位址，完成後按下「**確定**」即可。



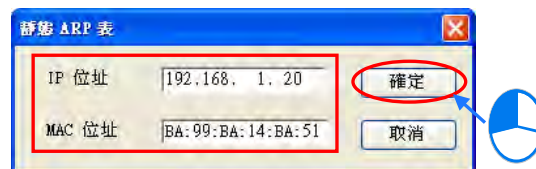
➤ 靜態 ARP 表

此功能用來綁定 IP 位址與對應的 MAC 位址。因每個裝置的 MAC 位址是唯一且絕對不會重複的，因此在知道通訊對象 A 的 MAC 位址時，便可將其 MAC 位址與裝置 A 被配置的 IP 位址加以綁定，之後主機便只會將該 IP 位址認定為是裝置 A 的專屬位址，而即使有另一裝置 B 被設定為同樣的 IP 位址，主機也不會對裝置 B 做出任何回應。此功能的設定步驟如下，而清單中最多可綁定 16 組裝置位址。

- (1) 勾選「**啟動靜態 ARP 表**」，之後於欲修改或新增的項目行列上雙擊滑鼠左鍵。



- (2) 於設定視窗中輸入指定裝置的 MAC 及 IP 位址，完成後按下「**確定**」即可。



以下圖的表格為例，當下列表格的設定被成功下載至主機之後，當主機開始運行時，只有 IP 位址位於 192.168.1.1 ~ 192.168.1.50 以及 192.168.1.100 ~ 192.168.1.150 這兩個區間範圍內的裝置才允許與主機進行通訊；當主機與 IP 位址為 192.168.1.20 的裝置進行通訊時，該裝置的 MAC 位址必須為 BA : 99 : BA : 14 : BA : 51；同樣的，當主機與 IP 位址為 192.168.1.21 的裝置進行通訊時，該裝置的 MAC 位址亦必須為 00 : 10 : 5A : 81 : CA : FD。

☑ 啟動 IP 位址過濾設定		
項次	起始 IP 位址	結束 IP 位址
1	192.168.1.1	192.168.1.50
2	192.168.1.100	192.168.1.150
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

☑ 啟動靜態 ARP 表		
項次	IP 位址	MAC 位址
1	192.168.1.20	BA:99:BA:14:BA:51
2	192.168.1.21	00:10:5A:81:CA:FD
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		

3

*.欲獲知主機本身的 MAC 位址時，可藉由 HWCONFIG 中的線上功能來檢視 CPU 主機的資訊(第 3.8.6 節)；又或者透過 ISPSOft 的系統資訊來得知目前連線主機的資訊與狀態(第 18.4.1 節)。

3.3.5.2 乙太網路進階參數 - 網路時間校正

此類參數位於「乙太網路 - 進階設定」主類別的「網路時間校正」副類別頁面中，主要在於啟動主機以 NTP 伺服器來進行時間校正的功能，並可於此處進行相關設定。關於 NTP 的概念請自行參考相關文件或書籍，此處將不再多加介紹。

CPU | COM Port | 乙太網路 - 基本設定 | **乙太網路 - 進階設定**

啟動 NTP 客戶端服務

NTP 伺服器: 192.168.1.1

更新週期: 30 分 (1~1440分)

日光節約時間

啟始日期: 1 / 2 (月 / 日)

結束日期: 3 / 4 (月 / 日)

時間: 0 點鐘

時區: (GMT-12:00) 國際換日線以西

過濾器: **網路時間校正** | 郵件 | 郵件發送條件 | 郵件位址與發送條件設定 | 通訊接口 | 網頁功能

欲設定網路時間校正時，請先勾選「**啟動 NTP 客戶端服務**」，之後於下方的區域設定相關參數。

- ❶ 設定 **NTP 伺服器** 的 IP 位址。主機將會定期連線至此伺服器來校正內部的時間。
- ❷ 設定定期更新的時間間隔。以上圖為例，主機將會每隔 30 分鐘便連線至 **NTP 伺服器**。
- ❸ 設定是否啟用**日光節約時間**，並於下方欄位設定採用**日光節約時間**的啟始與結束時間。
- ❹ 設定所採用的時區。

3.3.5.3 乙太網路進階參數 - 郵件

此類參數位於「**乙太網路 - 進階設定**」主類別的「**郵件**」副類別頁面中，主要在於設定郵件功能。

欲啟動郵件功能時，請先勾選「**啟動郵件發送功能**」，之後於下方的區域設定相關參數。

- ❶ 郵件伺服器：於此設定外送郵件伺服器的 IP 位址。
- ❷ 通訊埠：於此輸入外送郵件伺服器的通訊埠，或點選右側的 預設值(25) 使用預設值。
- ❸ 本機郵件位址：設定發送電子郵件時所使用的寄件信箱，最多可輸入 64 個字元。
- ❹ 郵件主旨：此處的字串會固定被加在每封郵件的主旨開頭，而輸入的內容只可為英數字或特殊符號，且最多只可輸入 16 個字元。
- ❺ 帳號認證：依據外送郵件伺服器的狀況，於此處設定登入前是否須先驗證帳號及密碼。
- ❻ 帳號認證資訊：輸入登入外送郵件伺服器時須驗證的**帳號**與**密碼**，最多可輸入 16 個字元。而當上述的各項參數皆已正確輸入，且電腦可與外送郵件伺服器正常連線時，按下「**認證**」鍵便可測試帳號密碼是否正確。

3.3.5.4 乙太網路進階參數 - 郵件發送條件

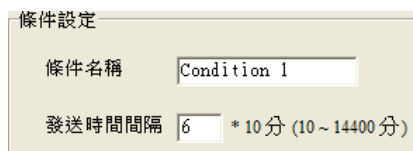
此類參數位於「乙太網路 - 進階設定」主類別的「郵件發送條件」副類別頁面中，主要在於設定發送郵件的觸發條件。欲設定此處的功能之前，必須先勾選「郵件」副類別頁面中的「**啟動郵件發送功能**」，請參考第 3.3.5.3 節的相關說明。

項次	條件名稱	條件模式	附件
1			
2			
3			
4			
5			
6			
7			
8			

在「發送條件設定」的表格中會列出所有的發送條件，最多可設定 8 組；當要設定或檢視某組條件的詳細內容時，只要在該列位置上點擊兩下滑鼠左鍵，之後便會開啟設定視窗。



(1) 首先於「條件設定」區域設定此組郵件發送的條件名稱與時間間隔。



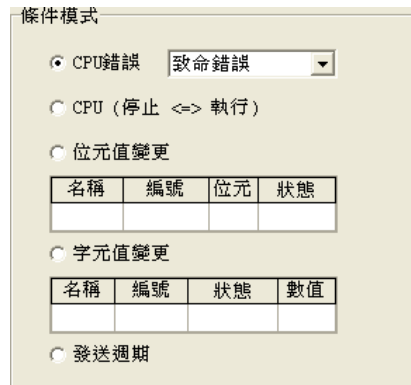
➤ 條件名稱

設定此組發送條件的名稱，而輸入的內容只可為英數字或特殊符號，且最多只可輸入 16 個字元。而當有電子郵件因為這組發送條件而被寄送出去時，這個名稱便會被附加在郵件的主旨中。

➤ 發送時間間隔

設定重複發送的時間間隔，以每 10 分鐘為一單位。當此組發送條件持續成立時，系統便會每隔一段時間就發出一封電子郵件；而在間隔時間內，針對同樣的發送條件，郵件則不會再重複發送。

(2) 接著於「條件模式」區域選擇發送電子郵件的觸發條件。



3

➤ CPU 錯誤

當主機發生錯誤時，發送條件便會成立，而關於主機錯誤狀態的相關說明，請參考各機種的操作手冊。當選擇此條件時，請再於右側的下拉選單中挑選錯誤的類別。

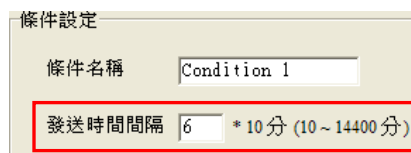
- (a) **致命錯誤**：當主機發生被定義為**致命錯誤**的狀況時才會觸發郵件發送。
- (b) **所有錯誤**：只要主機有錯誤產生時，郵件發送條件就會成立。

➤ CPU (停止 <=> 執行)

當主機的運行狀態由 STOP 切換至 RUN，或由 RUN 切換至 STOP 時，發送條件便會成立。

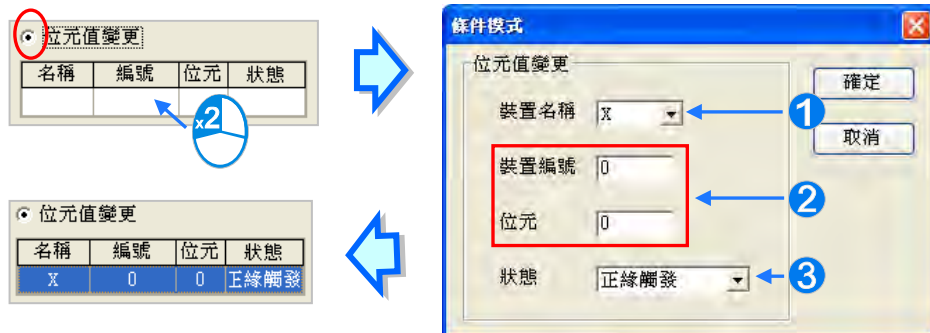
➤ 發送週期

定期發出電子郵件。而發送時間則根據上方「條件設定」中的「發送時間間隔」來決定。



➤ 位元值變更

當指定裝置的位元狀態變化符合設定條件時便會觸發郵件發送。以下圖為例，當 X0.0 由 OFF 變為 ON 時，郵件發送條件就會成立。欲設定條件時，請於選項下方的表格上雙擊左鍵即可開啟視窗。

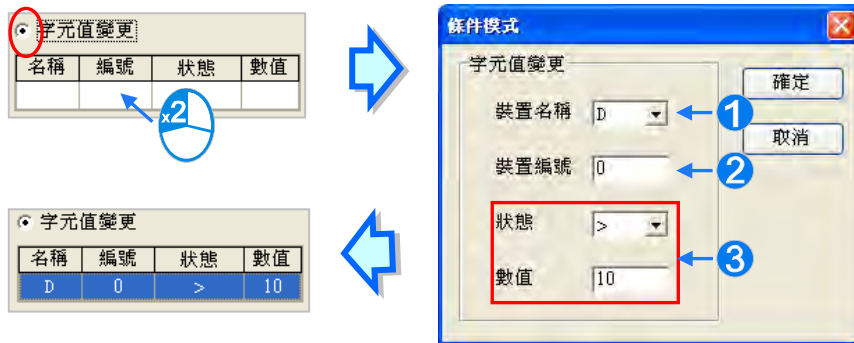


- ❶ 裝置名稱：於此下拉選擇裝置的類別。
- ❷ 裝置編號 & 位元：於此輸入裝置的位址，且當所選裝置為 X/Y/D/L 時須一併指定位元編號。
- ❸ 狀態：於此下拉選擇觸發的條件為正緣觸發或負緣觸發。

➤ 字元值變更

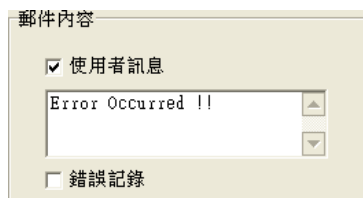
當指定裝置的內容數值符合設定條件時便會觸發郵件發送。以下圖為例，當 D0 大於 10 時，郵件發送條件就會成立。欲設定條件時，請於選項下方的表格上雙擊左鍵即可開啟視窗。

3

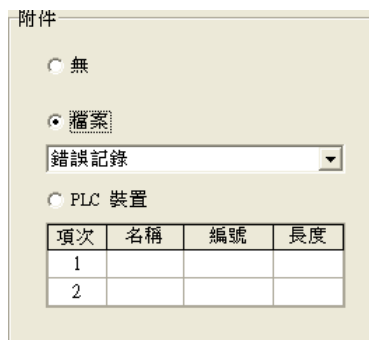


- ❶ 裝置名稱：於此下拉選擇裝置的類別。
- ❷ 裝置編號：於此輸入裝置的位址。
- ❸ 狀態 & 數值：於此設定觸發的條件。

(3) 接著於「郵件內容」區域勾選要做為郵件本文的內容，其項目包括自訂的訊息與存放於主機內部的錯誤記錄，而其中使用者自訂的訊息只可包含英數字及特殊符號。



(4) 接著於「附件」區域設定是否要於郵件中加入附件。而欲設定附件時，請務必確認各機種所允許的最大郵件檔案大小，相關資訊請參考各機種的操作手冊。



➤ 無

選擇此項代表無需附加檔案。

➤ 檔案

可選擇記憶卡中的**錯誤記錄**或**系統備份檔**做為郵件附件。

➤ PLC 裝置

若選擇此項，則當發送電子郵件時，系統會主動擷取下方設定範圍中的裝置狀態或數值來做為附加檔案；而在選擇此項之後，於下方的裝置欄位上雙擊左鍵便會出現設定視窗，至多可設定兩組。以下圖為例，當此組條件成立時，裝置 D0~D9 的內容值便會被當作附件而一併發送出去。



(5) 完成所有設定後按下「確定」即可建立一組條件。



3

3.3.5.5 乙太網路進階參數 - 郵件位址與發送條件設定

此類參數位於「乙太網路 - 進階設定」主類別的「郵件位址與發送條件設定」的副類別頁面中。主要在於設定郵件發送的觸發條件與郵件收件者之間的關係。而欲設定此處的功能之前，必須於「郵件」副類別頁面中勾選「啟動郵件發送功能」，並於「郵件發送條件」的副類別頁面中至少設定一組的郵件發送條件，相關說明請參考第 3.3.5.3 節與第 3.3.5.4 節。

3

CPU | COM Port | 乙太網路 - 基本設定 | **乙太網路 - 進階設定**

下方列表功能必需先勾選郵件分頁中"啟動郵件發送功能"方可使用。下方列表的功能在於設定郵件的傳送條件，一共可設定 8 組收件人。每組郵件傳送條件的設定步驟如下：

- (1) 於左側條件 1 ~ 條件 8 欄位中勾選此組郵件的傳送條件。
- (2) 於右側的郵件位址欄位中輸入收件人的郵件位址。

在完成設定以及下載的程序後，只要該組設定所勾選的條件中有任何一個成立，主機便會自動傳送郵件至指定的收件人信箱。

郵件位址與發送條件設定

	條件 1	條件 2	條件 3	條件 4	條件 5	條件 6	條件 7	條件 8	郵件位址
郵件 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

過濾器 | 網路時間校正 | 郵件 | 郵件發送條件 | **郵件位址與發送條件設定** | 通訊接口 | 網頁功能

設定時請先於右側的「郵件位址」欄位輸入收件者的電子郵件，而每個主機至多可設定 8 個位址，且每個位址最大可輸入 64 個字元；接著再於左側勾選對應此收件者的發送條件，而未設定郵件位址或是於「郵件發送條件」的副類別頁面中未設定該組條件時，對應的勾選位置將會呈現灰階的狀態。

以下圖為例，當條件 1 或條件 4 成立時，系統會自動發出郵件至 **user_1@delta.com.tw**；當條件 2 或條件 5 成立時，系統則會發出郵件至 **user_2@delta.com.tw**；而當條件 1 成立時，系統除了會發出郵件至 **user_1@delta.com.tw** 之外，同樣會發出郵件至 **user_4@delta.com.tw**。

郵件位址與發送條件設定

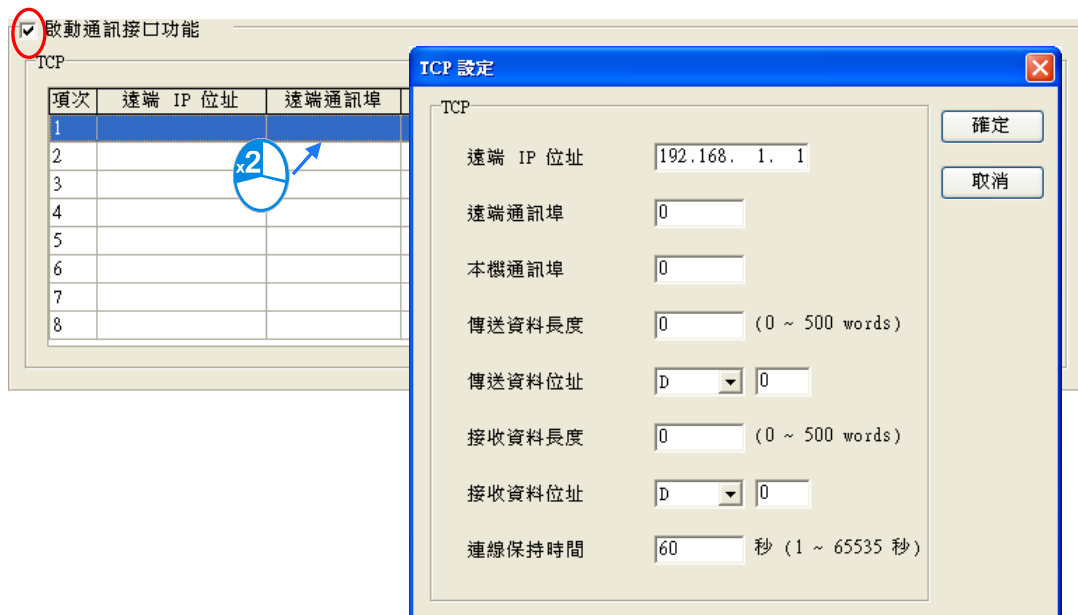
	條件 1	條件 2	條件 3	條件 4	條件 5	條件 6	條件 7	條件 8	郵件位址
郵件 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	user_1@delta.com.tw
郵件 2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	user_2@delta.com.tw
郵件 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	user_4@delta.com.tw
郵件 5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
郵件 8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

3.3.5.6 乙太網路進階參數 - 通訊接口 (Socket)

此類參數位於「乙太網路 - 進階設定」主類別的「通訊接口」副類別頁面中，主要在於設定透過乙太網路來進行資料傳遞的通訊接口參數；不過此功能仍需搭配特定的 API 應用指令才可進行操作，相關細節請參考 **AH500 程式手冊**。關於通訊接口 (Socket) 的概念請自行參考相關文件或書籍，此處將不再多加介紹。



AH500 系統支援主機本身與其他主機或裝置之間，經由通訊接口 (Socket) 的建立來進行資料傳遞，且同時支援 TCP 與 UDP 兩種傳輸協定，而兩種協定都可各自設定 8 組的連線通道；欲設定時請先勾選「啟動通訊接口功能」，之後於表格中欲設定的行列處雙擊滑鼠左鍵即可開啟設定視窗。



TCP 與 UDP 協定的參數幾乎相同，差別只在於 UDP 沒有「連線保持時間」的參數，因此下列僅以 TCP 的參數設定視窗來說明各參數本身所代表的意義。

編號	參數名稱	設定值	備註
1	遠端 IP 位址	192.168.1.100	
2	遠端通訊埠	65500	
3	本機通訊埠	65501	
4	傳送資料長度	10	(0 ~ 500 words)
5	傳送資料位址	D, 0	
6	接收資料長度	20	(0 ~ 500 words)
7	接收資料位址	D, 100	
8	連線保持時間	60 秒	(1 ~ 65535 秒)

- ❶ 設定遠端裝置的 IP 位址。
- ❷ 設定在這組連線中，遠端裝置所使用的通訊埠編號，輸入範圍為 0 ~ 65535。
- ❸ 設定在這組連線中，本地主機所使用的通訊埠編號，輸入範圍為 0 ~ 65535。
- ❹ 設定本地主機要傳送的資料長度，輸入範圍為 0 ~ 500 words。
- ❺ 設定在本地主機中，存放傳送資料的啟始位址。
- ❻ 設定本地主機要接收的資料長度，輸入範圍為 0 ~ 500 words。
- ❼ 設定在本地主機中，存放接收資料的啟始位址。
- ❽ 設定連線的最大閒置時間。當建立的連線超過此時間無資料傳輸時，主機便會自動中斷連線。

設定時，本機與遠端的通訊埠編號不可相同，且傳送與接收的裝置位址範圍不可重疊。而以上圖為例，遠端裝置的 IP 位址為 192.168.1.100，且開啟一個編號為 65500 的通訊接口，而本地主機也開啟一個編號為 65501 的通訊接口，兩者之間便將透過這組連線通道來進行資料的傳遞。

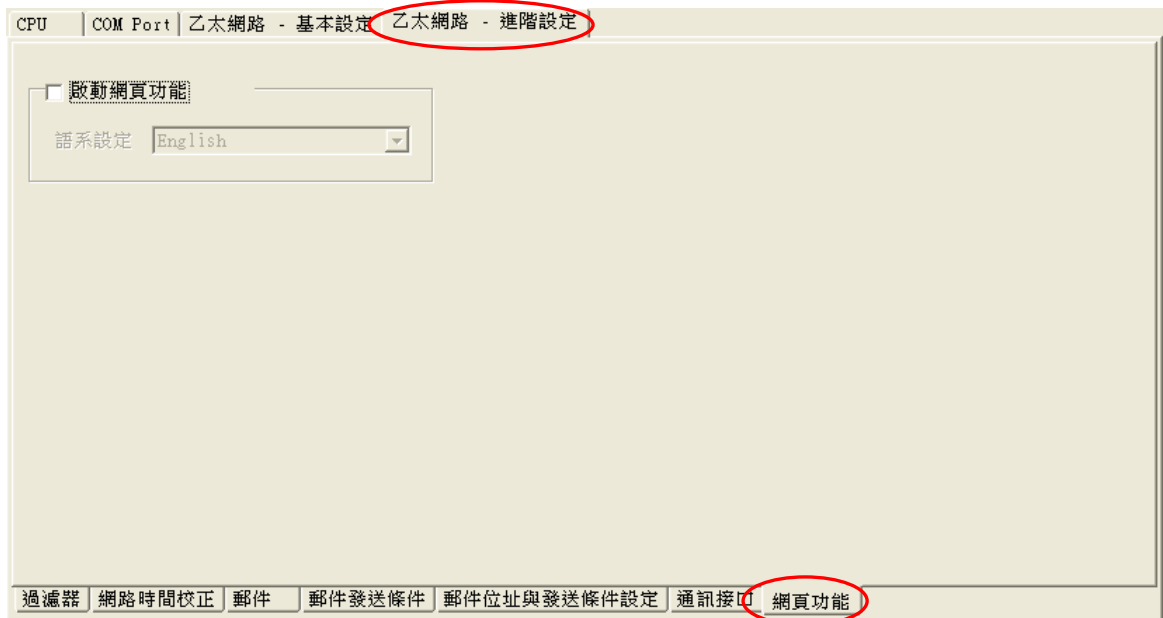
當本地主機要傳送資料給遠端裝置時，本地主機會先將資料存放在 D0~D9 共 10 個 WORD 的空間暫存之後，再將資料傳送出去；而當接收到來自遠端裝置的資料時，主機則會將接收的資料存放在裝置 D100 ~ D119 共 20 個 WORD 的空間中。

若收到的資料量大於所設定的長度時，主機會根據設定的長度，將最前面的 20 個資料存放在 D100 ~ D119 中，超出的部份則會直接將其丟棄；同樣的，若收到的資料量小於設定長度時，主機則會從設定的啟始位置 D100 依序存放資料，而未被存放新資料的位置則會保持原本的內容值。

當通道連線已超過 60 秒未有任何的資料傳輸時，主機便會主動將主機端的通訊接口 (Socket) 關閉，並結束通道連線。

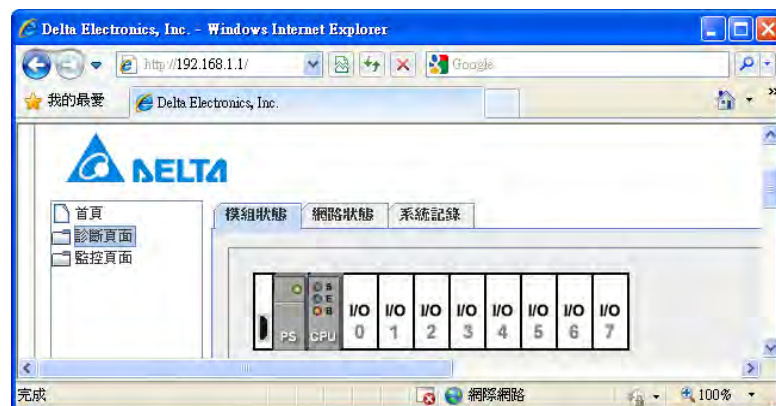
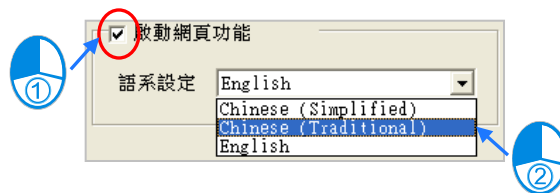
3.3.5.7 乙太網路進階參數 - 網頁功能

此參數位於「乙太網路 - 進階設定」主類別的「網頁功能」副類別頁面中，主要在於設定是否啟動內嵌於主機端的網頁監控功能。



3

首先勾選「啟動網頁功能」項目，之後於下拉選單中點選欲採用的網頁語系；當此參數被正確的下載至主機後，使用者即可透過網頁瀏覽器直接檢視內嵌於主機端的監控網頁。



*關於主機內嵌的網頁監控功能，相關細節請參考各機種的操作手冊。

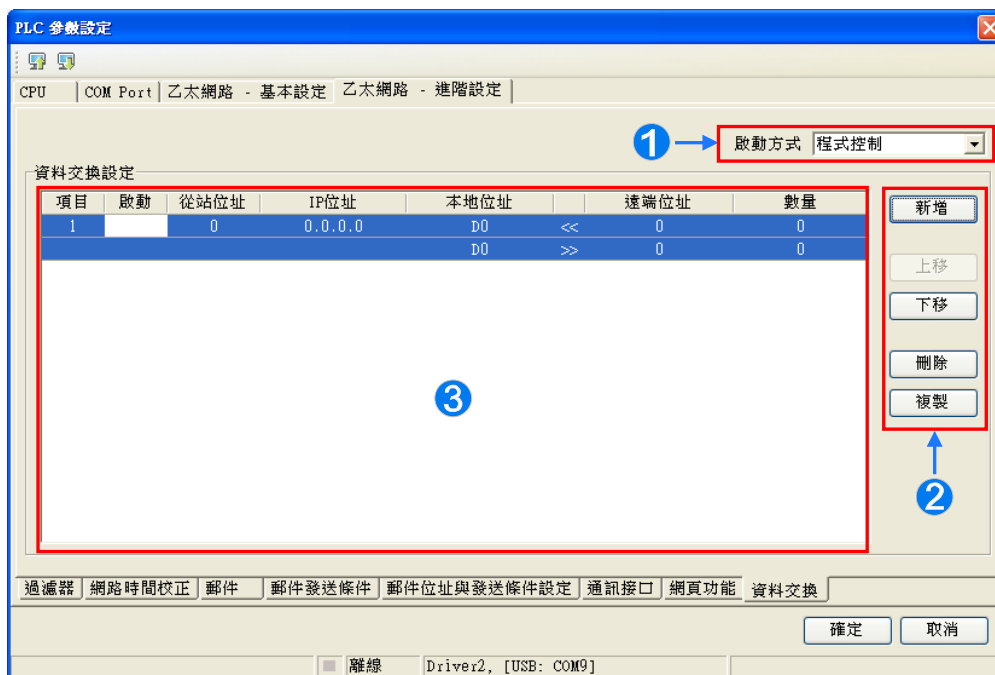
3.3.5.8 乙太網路進階參數 - 資料交換

AH5x0-EN 系列機種的 CPU 模組本身內建了基於標準 MODBUS TCP 的資料交換功能，此功能位於「乙太網路 - 進階設定」主類別的「資料交換」副類別頁面中，功能的設定方式如下。

3



第一次開啟此分頁時，因尚未設定任何資料，所以表格是空白的，點擊右側的「新增」按鈕便可插入一組新資料，在該資料上方雙擊左鍵便可開啟資料設定畫面。

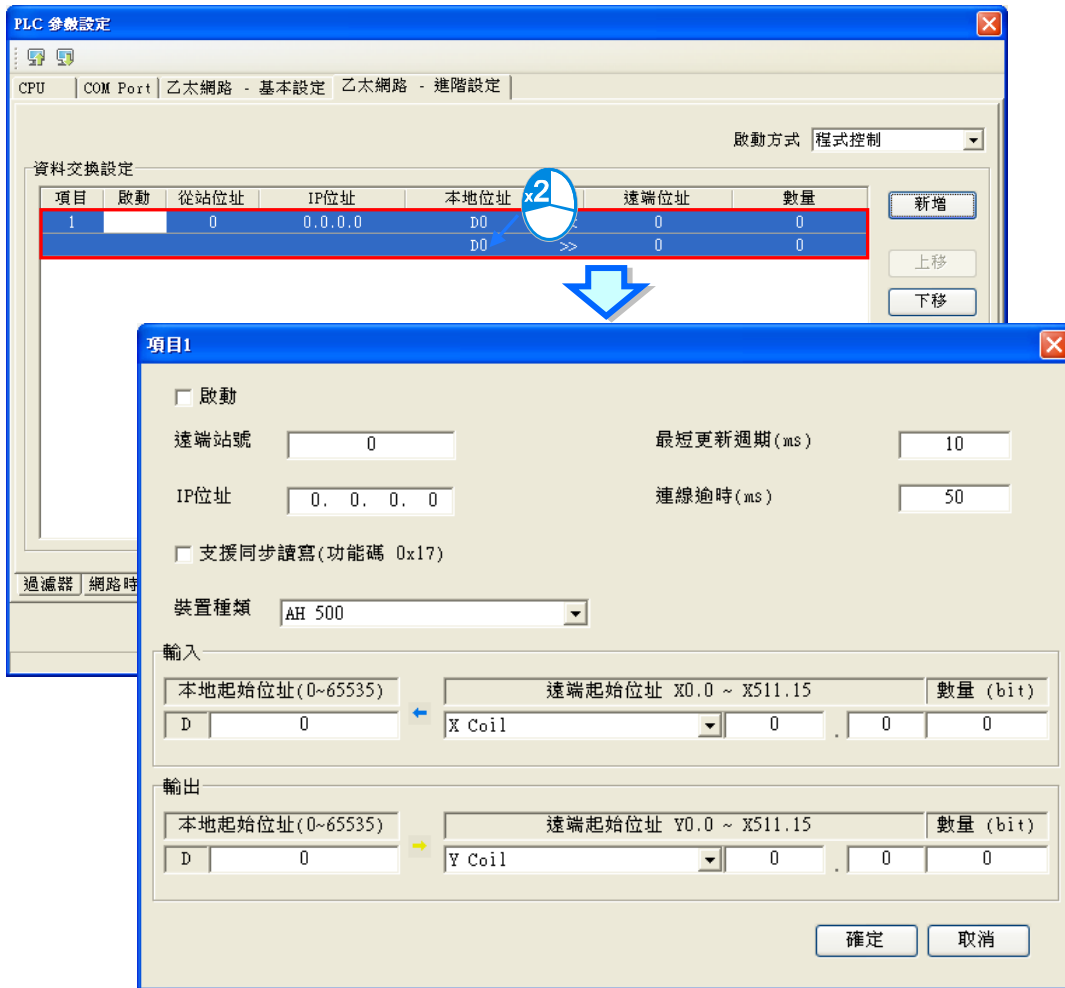


- ❶ 設定**啟動方式**為程式控制時為由 PLC 的程式控制是否執行設定的資料交換的旗標；PLC 執行為當 PLC 為 RUN 的狀態時會自動執行設定的資料交換；永遠執行則為當 PLC 上電時便會不斷執行資料交換。
- ❷ 按下**新增**按鈕便會在區域❸的資料交換表區域新增一列新的資料交換表。按下**上移**與**下移**按鈕可以對目前選擇的資料交換表往上一列或下一列搬移。按下**刪除**按鈕可刪除目前選擇的資料交換表。按下**複製**按鈕可以將目前選擇的資料交換表複製後貼上到最下方一列。
- ❸ 列出建立的資料交換表。說明如下表。

表格欄位	說明
項目	表示資料交換區塊的序號。
啟動	表示進行資料交換時，是否啟動此組資料交換表。
從站位址	註明此組資料交換表的目標站號位址。
IP 位址	註明此組資料交換表的目標 IP 位址，同一個位址可規劃為多組資料交換表。
本地位址	註明主站於此組資料交換表中所使用的裝置位址範圍。
<< / >>	「<<」：輸入，表示此列所顯示的是主站讀取從站的資料區塊。 「>>」：輸出，表示此列所顯示的是主站寫入從站的資料區塊。
遠端位址	註明從站於此組資料交換表中所使用的裝置位址範圍。而當此組資料交換表的從站為自行定義的「MODBUS Device」裝置時，此範圍會以 16 進制的位址編號來顯示。
數量	註明此組交換表的長度，基本上與裝置範圍所計算的結果一致。

以滑鼠左鍵雙擊欲設定的資料交換表，開啟資料交換表的設定項目，如下圖。勾選**啟動**代表欲啟動此組資料交換表，並且隨著前述的**啟動方式**啟動。**遠端站號**用於進行資料交換的目標從站設備站號。**最短更新週期 (ms)**為設定此組資料交換表進行資料交換的時間週期。**連線逾時 (ms)**為設定若目標設備超過此設定時間未回應，則為逾時。勾選**支援同步讀寫 (功能碼 0x17)**將利用特定的 MODBUS 功能碼，主站便可在一次的命令當中完成**讀與寫**的動作，以提高資料交換的效率；不過設定時，請確認參與資料交換的所有裝置都有支援同步讀寫的 MODBUS 功能碼，否則便可能在接收到主站的命令後，會因無法識別該功能碼而造成讀寫失敗。**裝置種類**為目標從站設備的機種，包括台達 PLC 與標準 MODBUS 設備。

3



- 「輸入」的資料區塊

設定當 AH5x0-EN 系列機種自遠端設備讀取資料時，AH5x0-EN 系列機種用以存放資料的 **D 裝置起始位址**，接著指定欲讀取的遠端設備**裝置類別**及**起始位置**。設定輸入的**資料長度**。

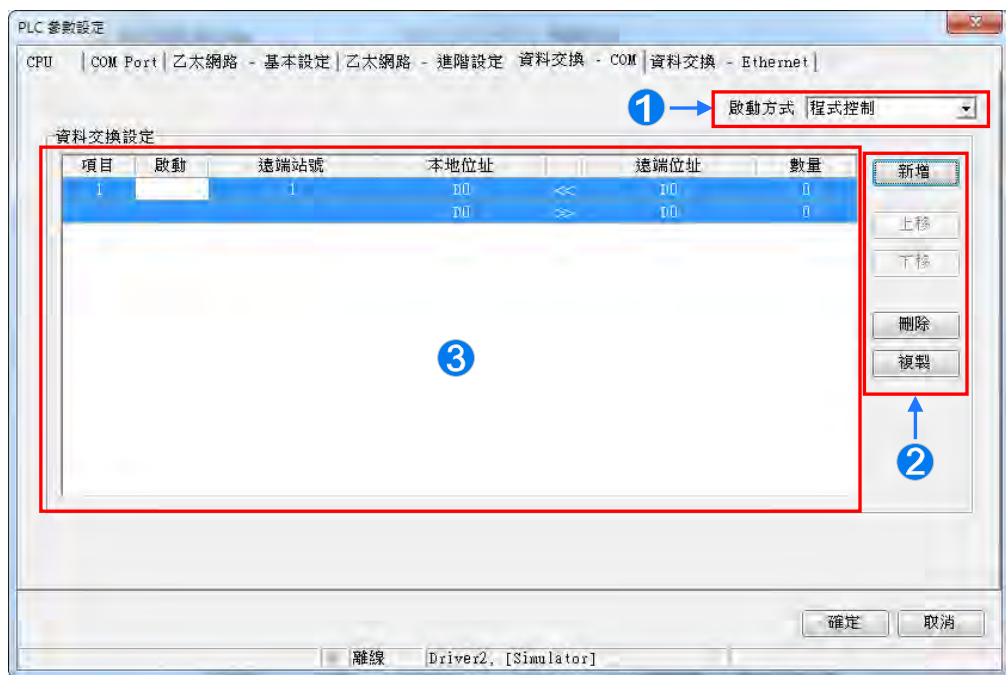
- 「輸出」的資料區塊

設定當 AH5x0-EN 系列機種寫入資料至遠端設備時，AH5x0-EN 系列機種來源資料的 **D 裝置類別起始位址**，接著指定被寫入資料的遠端設備**裝置類別**及**起始位置**。設定輸出的**資料長度**。

AH5x0-EN 系列機種在設定資料交換區塊時，本地端只能以 Word 為單位的 D 裝置，遠端則可以 Bit 為單位，且 Bit 不必由 0 開始，例如遠端的起始位址可設為 D0.3；當設定的資料長度不是 Word 的倍數時，本地端仍會以對應的 Bit 進行資料交換，未參予資料交換的 Bit 則保持原狀。如將遠端的 M0~M19 讀取至本地端的 D0~D1，則資料會依序存放在 D0.0~D1.3，而 D1.4~D1.15 則會保持原狀。

3.3.6 資料交換 - COM 與資料交換 - Ethernet

AH5x1 系列機種及 AH560 備援系列機種另外提供了資料交換的設定頁面，而依據機種型號，有提供兩個 COM 埠，或是一個 COM 埠與一個乙太網路接口的機種，這些參數分別位於「資料交換-COM1」與「資料交換-COM2」或「資料交換-COM」與「資料交換-Ethernet」分頁，用於建立 AH5x1 系列機種內建通訊埠的資料交換表。AH5x1 系列機種及 AH560 備援系列機種可作為主站，對於作為從站的遠端設備進行資料交換，COM1 與 COM2 埠口透過 MODBUS 協定；而乙太網路埠口則透過 MODBUS TCP 協定。設定欄位三分頁皆相同，如下圖。

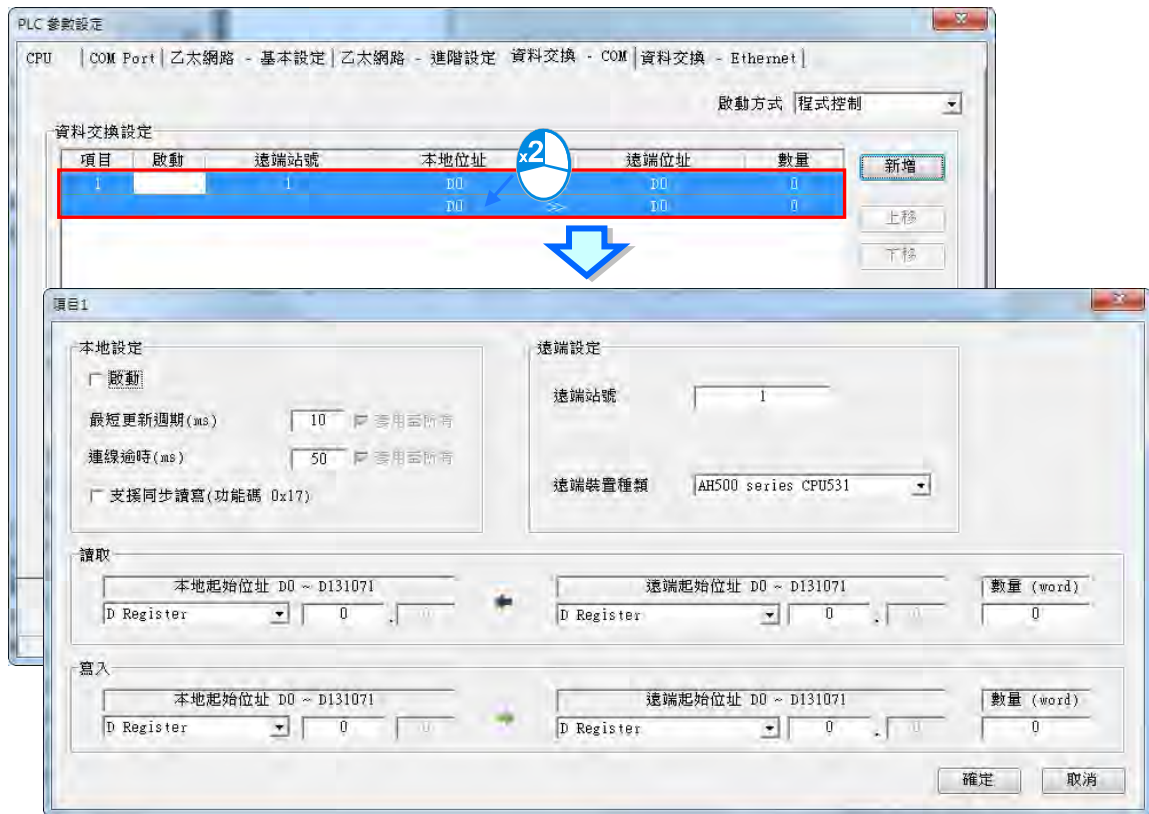


- ❶ 設定**啟動方式**為程式控制時為由 PLC 的程式控制是否執行設定的資料交換的旗標；PLC 執行為當 PLC 為 RUN 的狀態時會自動執行設定的資料交換；永遠執行則為當 PLC 上電時便會不斷執行資料交換。
- ❷ 按下**新增**按鈕便會在區域❸的資料交換表區域新增一列新的資料交換表。按下**上移**與**下移**按鈕可以對目前選擇的資料交換表往上一列或下一列搬移。按下**刪除**按鈕可刪除目前選擇的資料交換表。按下**複製**按鈕可以將目前選擇的資料交換表複製後貼上到最下方一列。
- ❸ 列出建立的資料交換表。說明如下表。

表格欄位	說明
項目	表示資料交換區塊的序號。
啟動	表示進行資料交換時，是否啟動此組資料交換表。
遠端站號	註明此組資料交換表所屬的從站位址，同一個位址可規劃為多組資料交換表。 COM1 與 COM2 分頁註明為站號，乙太網路分頁則註明為 IP。
本地位址	註明主站於此組資料交換表中所使用的裝置位址範圍。
<</>>	「<<」：輸入，表示此列所顯示的是主站讀取從站的資料區塊。 「>>」：輸出，表示此列所顯示的是主站寫入從站的資料區塊。
遠端位址	註明從站於此組資料交換表中所使用的裝置位址範圍。而當此組資料交換表的從站為自行定義的「MODBUS Device」裝置時，此範圍會以 16 進制的位址編號來顯示。
數量	註明此組交換表的長度，基本上與裝置範圍所計算的結果一致。

3.3.6.1 資料交換-COM1、資料交換-COM2 或資料交換-COM

在資料交換-COM1、資料交換-COM2 或資料交換-COM 分頁，以滑鼠左鍵雙擊欲設定的資料交換表，開啟資料交換表的設定項目，如下圖。勾選**啟動**代表欲啟動此組資料交換表，並且隨著前述的**啟動方式**啟動。**遠端站號**用於進行資料交換的目標從站設備站號。**最短更新週期 (ms)**為設定此組資料交換表進行資料交換的時間週期。**連線逾時 (ms)**為設定若目標設備超過此設定時間未回應，則為逾時。勾選**支援同步讀寫 (功能碼 0x17)**將利用特定的 MODBUS 功能碼，主站便可在一次的命令當中完成讀與寫的動作，以提高資料交換的效率；不過設定時，請確認參與資料交換的所有裝置都有支援同步讀寫的 MODBUS 功能碼，否則便可能在接收到主站的命令後，會因無法識別該功能碼而造成讀寫失敗。**遠端裝置種類**為目標從站設備的機種，包括台達 PLC 與標準 MODBUS 設備。



3

- 「輸入」的資料區塊

設定當 AH5x1 系列機種自遠端設備讀取資料時，AH5x1 系列機種用以存放資料的裝置類別、裝置起始位址，接著指定欲讀取的遠端設備裝置類別及起始位置。設定輸入的資料長度。

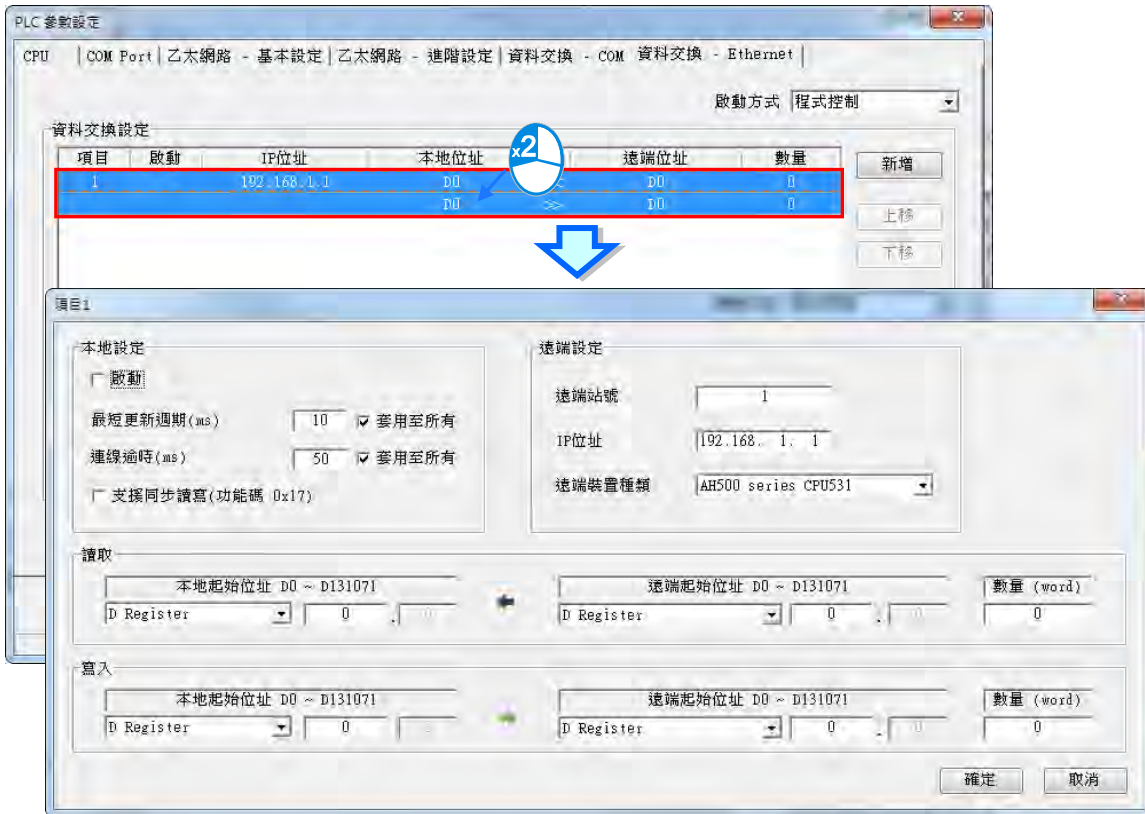
- 「輸出」的資料區塊

設定當 AH5x1 系列機種寫入資料至遠端設備時，AH5x1 系列機種來源資料的裝置類別、起始位址，接著指定被寫入資料的遠端設備裝置類別及起始位置。設定輸出的資料長度。

3.3.6.2 資料交換-Ethernet

在資料交換-Ethernet 分頁，以滑鼠左鍵雙擊欲設定的資料交換表，開啟資料交換表的設定項目，如下圖。設定內容基本上與資料交換-COM 分頁相同，只有裝置種類所支援的目標從站設備的機種，為包括台達 PLC 與標準 MODBUS TCP 設備，以及 IP 位址用於輸入遠端從站設備的 IP 位址，其餘設定內容請參閱資料交換-COM 分頁內容。

3



若在上圖的最短更新週期 (ms) 或連線逾時 (ms) 欄位右方勾選套用至所有資料，按下「確定」按鈕後，則此兩設定將會被寫入到資料交換-Ethernet 分頁其他的資料交換表中，也可不做勾選單獨設定此資料交換表。

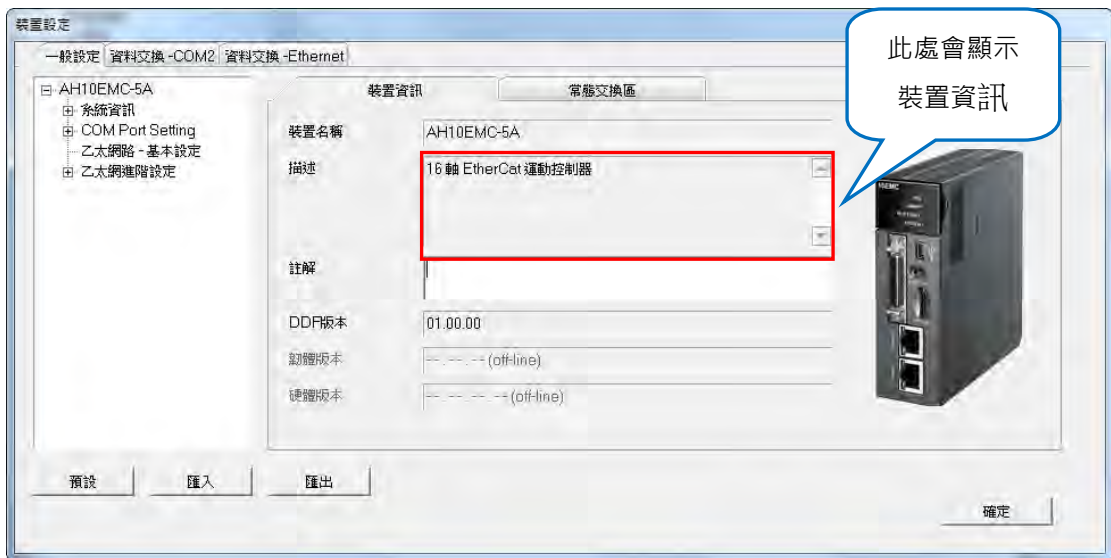
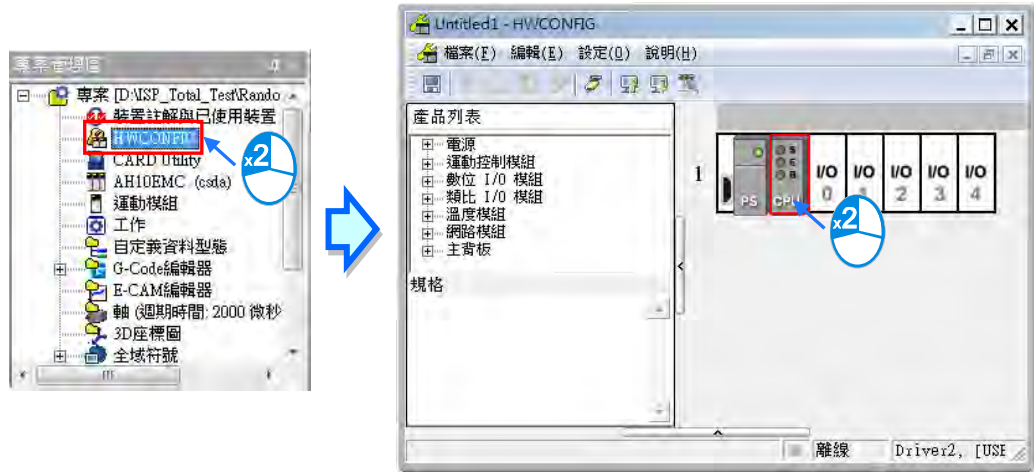
3.4 AHxxEMC 機種的 PLC 主機參數設定

3.4.1 開啟 PLC 主機的參數設定頁面

啟動 HWCONFIG 之後，於系統配置區的 CPU 圖示上雙擊滑鼠左鍵，接著系統便會自動開啟 CPU 主機的參數設定頁面，且隨著主機型號的不同，其可設定的參數與顯示的畫面也會有所差異。

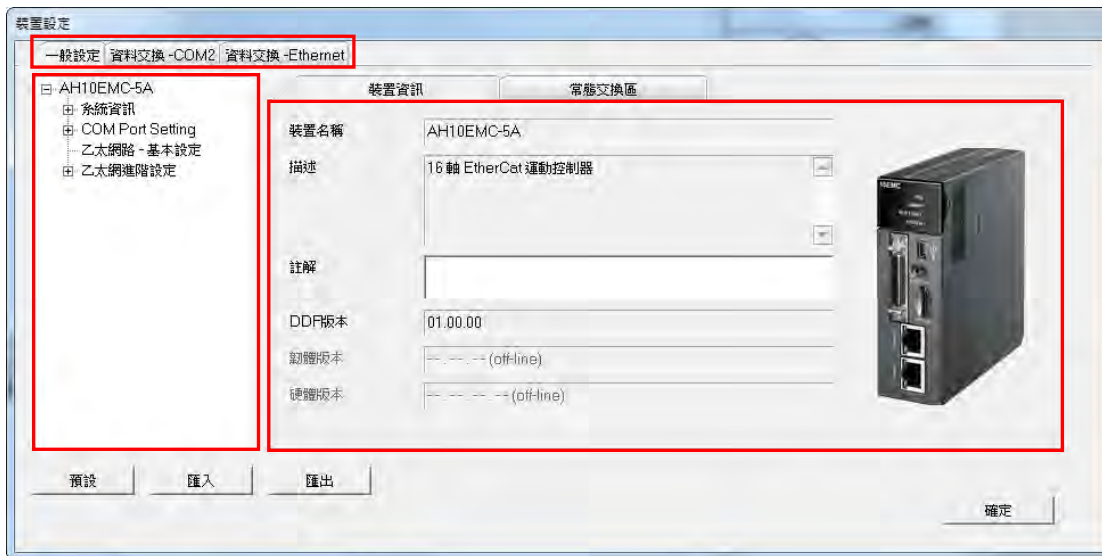
⚠ 設定主機參數前，請務必查閱各機種的操作手冊，並確認所設定的參數對主機本身以及整體系統的影響，以避免造成系統或人員的傷害。

*參數設定完畢之後，仍必須被下載至主機當中才可生效，下載的方式請參考第 3.8.4 節。



3

參數設定頁面中亦根據參數類別而區分為許多的分頁，操作時可配合視窗上方的主類別標籤以及位於每個主類別頁面中的項目階層切換不同的設定頁面。頁面右方標題列依序列出參數名稱，及供使用者填入的設定值欄位，設定值的單位及設定值的預設值、最大值與最小值。



3.4.2 一般設定

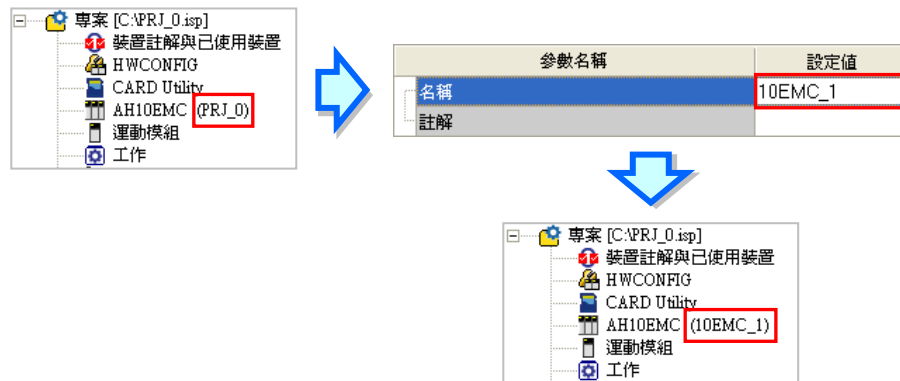
3.4.2.1 一般設定 - 系統資訊

此類參數位於「一般設定」分頁的「系統資訊」項目中，此項目內包括了系統資訊、系統參數、停電保持設定、本體 IO 中斷。

「系統資訊」頁面只有「名稱」與「註解」兩個欄位，其中「名稱」欄最多可輸入 31 個字元，而「註解」欄最多可輸入 63 個字元。且所輸入的字元並沒有特別的制限，包括特殊字元及空白鍵皆可輸入，不過須注意一個中文字將會佔用兩個字元。



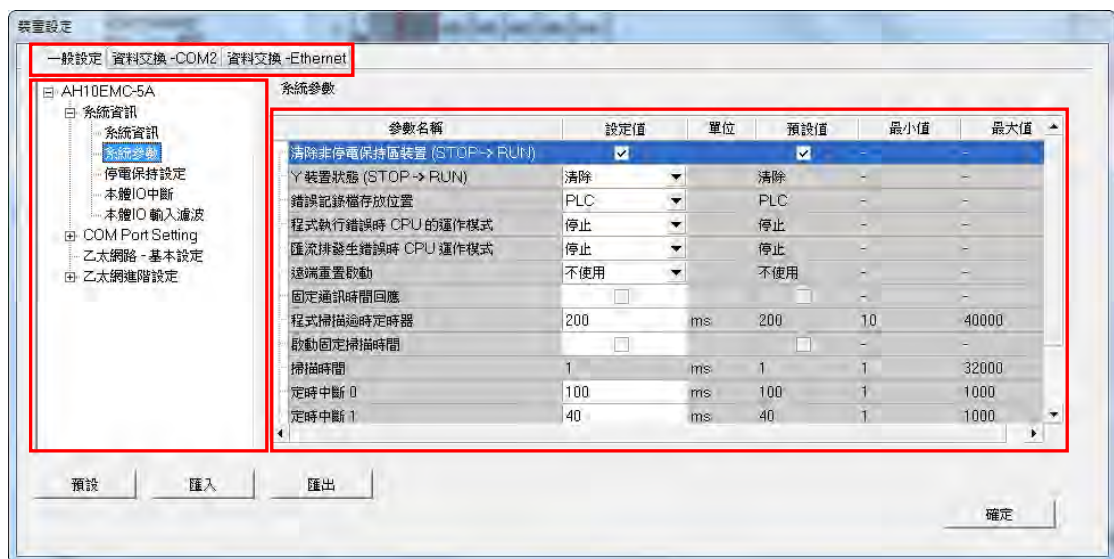
當新增一個 ISPSOft 專案時，系統便會以專案的名稱作為主機預設名稱，且會顯示於專案管理區的機種名稱後方，而後續則可於此處重新定義。



3

PLC 標籤名稱的功能主要是用來做為裝置識別，尤其是在聯網系統的應用或是在進行連線操作時，藉由確認 PLC 標籤名稱的方式便可防止使用者的誤操作；當使用者要對主機進行資料的上下載或其它的連線操作時，若發現操作對象的主機標籤名稱與專案中的標籤名稱不符時，系統便會提醒使用者做再次的確認，以避免對其他的主機造成非預期的影響。

「系統參數」頁面設定各項參數如下，可藉由下拉選單或輸入框設定適當的參數值。



➤ 清除非停電保持區裝置 (Stop → Run)

勾選設定當 PLC 主機的狀態由 STOP 切換至 RUN 時，非停電保持區的所有裝置狀態及數值是否被清除。

➤ Y 裝置狀態 (Stop → Run)

設定主機狀態在 STOP 切換到 RUN 時的 Y 裝置狀態。

- 清除：將所有 Y 裝置的輸出都設為 OFF。

- 保持現在狀態：輸出將保持現狀。
- 回復停止前狀態：Y 裝置的輸出將回復到主機 STOP 之前的狀態。

➤ 錯誤記錄檔存放位置

設定選擇錯誤記錄的存放位置。

- PLC：存放於 PLC。存滿二十筆資料後，舊的紀錄會被新的紀錄覆蓋。
- PLC 及記憶卡：存放於 PLC 與記憶卡。PLC 存滿二十筆資料後，會被搬移至記憶卡存放。

3

➤ 程式執行錯誤時 CPU 的運作模式

設定選擇程式執行錯誤時 CPU 的運作模式。

- 停止：CPU 將切換到 Stop。
- 維持原狀態：CPU 維持發生錯誤前的狀態。

➤ 匯流排發生錯誤時 CPU 運作模式

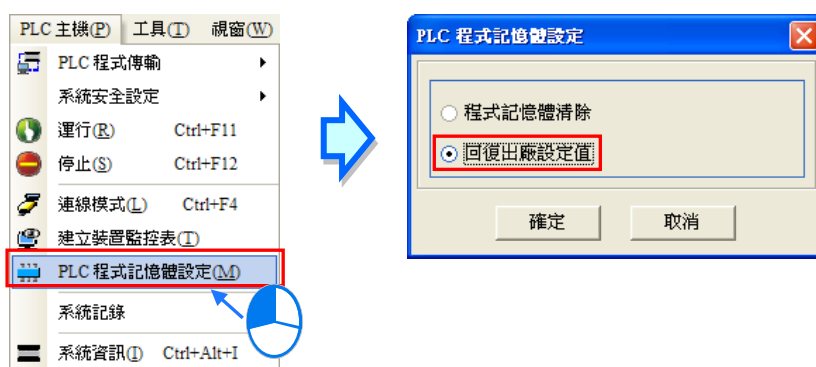
設定選擇匯流排發生錯誤時 CPU 的運作模式。

- 停止：CPU 將切換到 Stop。
- 維持原狀態：CPU 維持發生錯誤前的狀態。

➤ 遠端重置啟動

設定是否允許經由遠端系統來重置主機。以下圖 ISPSOft 為例，必須先勾選此功能，並確實將參數下載至主機之後，於 ISPSOft 中才可執行將主機恢復為出廠預設值的功能。

- 不使用：不使用遠端重置功能。
- 使用：使用遠端重置功能。



➤ 固定通訊時間回應

當未啟動此功能時，系統會在每次的掃描週期結束後，才會處理所有通訊埠的回應；而將此功能啟動後，主機便會每隔一段時間就會輪流處理各通訊埠的工作，因此可得到較即時的通訊回應；不過當系統處理通訊回應的工作時，會先將程序掃描的工作暫時中斷，因此啟用此功能將可能造成掃描時間的延長，請務必確認使用時不會造成系統運作的問題。

➤ 程式掃描逾時定時器

設定的掃描逾時時間，若程式執行超過此時間，主機會回報錯誤。

➤ 啟動固定掃描時間

勾選設定主機在最短在固定的時間完成一次掃描週期。啟動後當實際的程序掃描時間小於設定值時，主機便會強制等待時間到達設定值後，才會開始下一次的掃描；而若實際的掃描時間大於此設定值，則主機便不理會此設定值而根據實際的掃描時間進行運作。

➤ 掃描時間

若前述**啟動固定掃描時間**設為啟動時，在此欄位輸入設定的週期時間。須注意若設定的掃描時間大於**程式掃描逾時定時器**的設定值時，一旦主機開始運行後，會發生逾時錯誤。

➤ 定時中斷 0

設定定時中斷的時間，搭配工作管理中的定時中斷 0 使用。請參考第五章。

➤ 定時中斷 1

設定定時中斷的時間，搭配工作管理中的定時中斷 1 使用。請參考第五章。

➤ 定時中斷 2


設定定時中斷的時間，搭配工作管理中的定時中斷 2 使用。請參考第五章。

➤ 高速掃描

設定定時中斷的時間，搭配工作管理中的定時中斷 3 使用。請參考第五章。

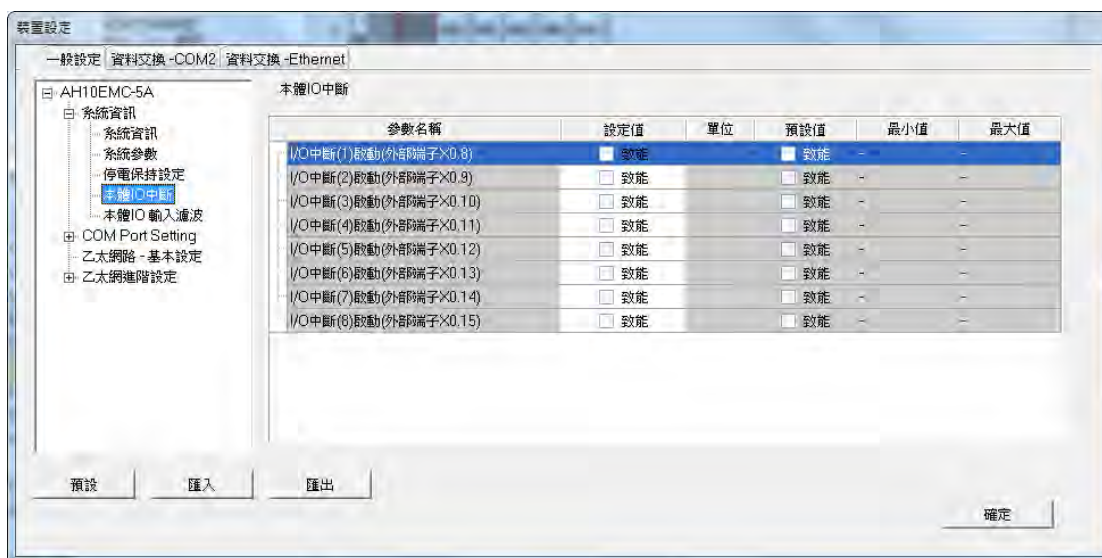
「**停電保持設定**」頁面設定各項參數如下，供使用者設定各種裝置的停電保持區範圍。



按下  開啟設定視窗，可設定此類裝置的起始與結束位址。若要使用停電保持功能，請在零到最大值範圍對起始與結束位址做設定。按下「預設值」按鈕恢復預設值，按下「清除」按鈕將欄位清除為-1，此時為不使用停電保持功能，完成設定後按下「確定」儲存設定或「取消」放棄設定後離開頁面。

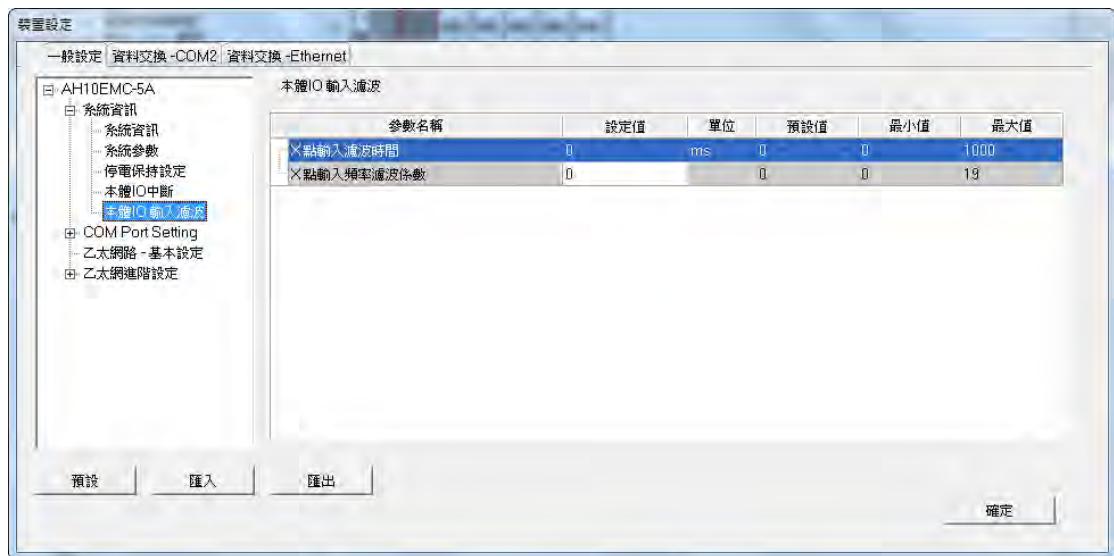


「本體 IO 中斷」頁面設定各項參數如下，供使用者勾選設定各 X 輸入點啟動中斷輸入功能。



X0.8~X0.15 將依序搭配工作管理中的 IO 中斷 (1) ~ IO 中斷 (8)，請參考第五章。

「本體 IO 輸入濾波設定」頁面設定各項參數如下，供使用者做主機輸入點濾波設定，用以過濾掉不需要的干擾訊號。



3

➤ X 點輸入濾波時間

用來設定輸入端接收脈波的反應時間，設定範圍 0~1000，單位 ms，小於此反應時間則會被過濾。

➤ X 點輸入硬體濾波時間

此設定值用來設定，小於 X 點輸入 Duty On 的時間，輸入範圍為 1~50000000，單位為 0.1us，此時間值 t 與濾波頻率之間的關係為濾波頻率 (f) * = $\frac{1}{2(\text{濾波時間}t)}$ (Hz)，輸入頻率高於此濾波頻率之訊號將會被濾除。(X 點輸入 Duty On 的時間之詳細解說，請參閱 **AH Motion Controller 操作手冊**)

*.關於「本體 IO 輸入濾波」，詳細設定內容請參閱 **AH Motion Controller 操作手冊**。

3.4.2.2 一般設定 - COM Port Setting

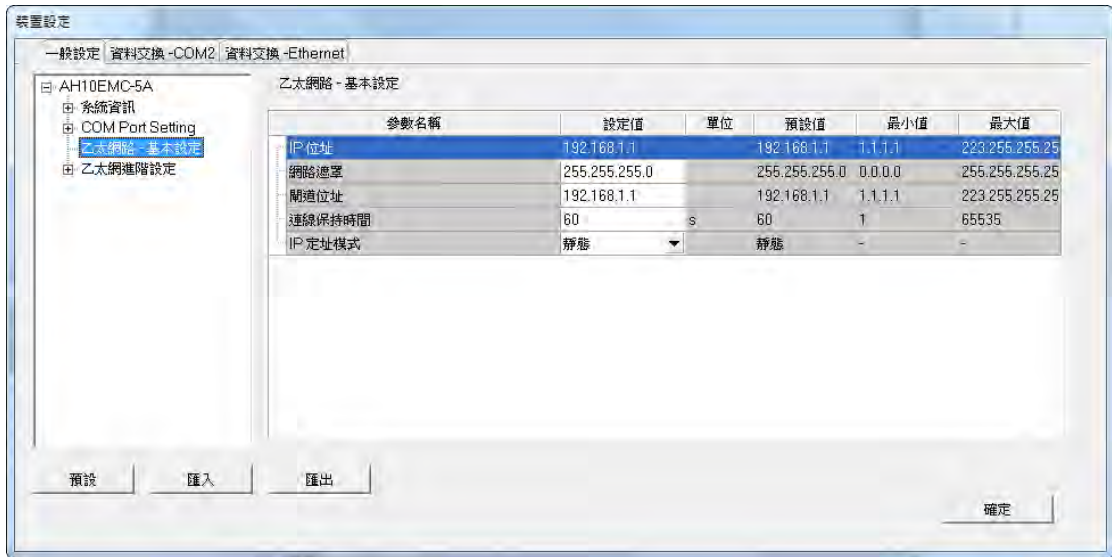
AHxxEMC 機種具備 2 個 COM 埠，點選項目列表中的 **COM Port 1 Setting** 或 **COM Port 2 Setting**，可分別設定 COM1 與 COM2 參數，COM1 專供使用者上下載，COM2 還可以進行資料交換，兩 COM 埠設定項目皆相同。



- ❶ 設定該埠口通訊協定。
- ❷ 設定此通訊埠的通訊站號，以做為網路通訊上的裝置識別，而在同一條網路上的站號則不允許重複。當該埠口扮演從站 (Slave) 的角色，且所連接的網路上亦有其它的從站時，此埠口的站號便不可設定為 0，因站號 0 在通訊協定中具有**廣播通訊**的使用意義。當主站 (Master) 在資料封包中指定將資料傳送給站號 0 的從站 (Slave) 時，這代表是要將資料傳送給所有的從站，而所有的從站，不論其本身的站號為何都會接受這筆資料封包。
- ❸ 當通訊失敗時，主機便會重試連線，而重試的次數便可於**傳輸錯誤自動詢問次數**欄位中進行設定。**自動詢問時間間隔**可設定通訊失敗時的重試時間間隔。當 PLC 主機因通訊失敗而嘗試重新連線時，便會在間隔此設定時間後才會進行下一次的連線動作。

3.4.2.3 一般設定 - 乙太網路 - 基本設定

於項目列表中點選「乙太網路 - 基本設定」項目即可進入。此處主要用於設定主機乙太網路 (Ethernet) 埠的通訊參數，關於透過 Ethernet 來與主機進行連線的相關注意事項請參考第 2.4.8 節當中的說明，而關於乙太網路的概念，請自行參考其他相關文件或書籍，此處將不再多做介紹。

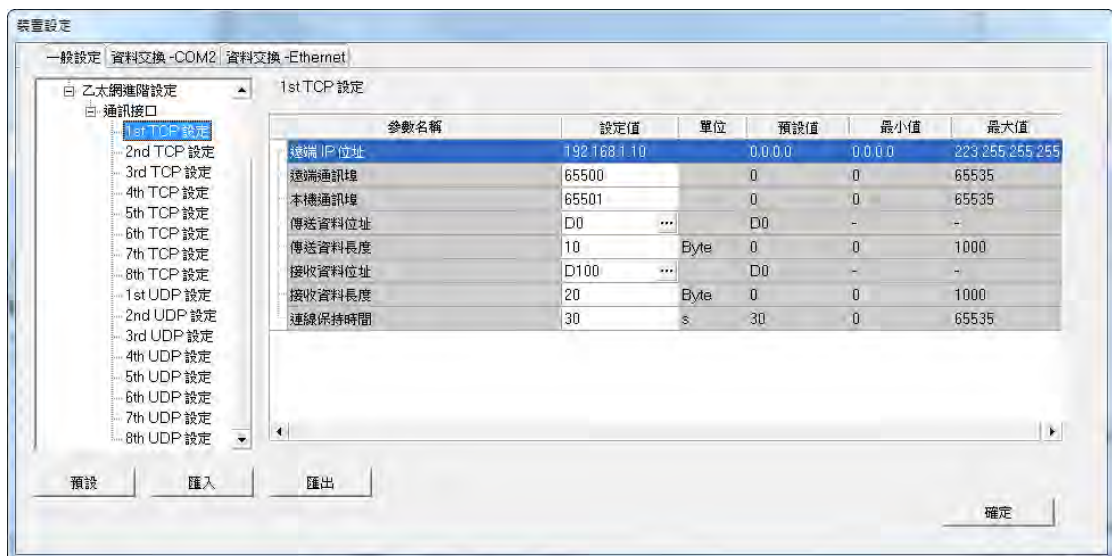


設定主機 IP 的定址模式。若設為「靜態」，代表 IP 位址等參數將由使用者直接指定，但若設為「動態」，則代表 IP 位址將由動態分配。

3.4.2.4 一般設定 - 以太网進階設定

以太网進階設定頁面中，主要在於設定透過以太网來進行資料傳遞的通訊接口參數，內含「通訊接口」頁面；此項目主要設定透過以太网來進行資料傳遞的通訊接口參數；不過此功能仍需搭配特定的 API 應用指令才可進行操作，相關細節請參考 **AHxxEMC 機種相關指令手冊**。關於通訊接口 (Socket) 的概念請自行參考相關文件或書籍，此處將不再多加介紹。

AHxxEMC 系列機種系統支援主機本身與其他主機或裝置之間，經由通訊接口 (Socket) 的建立來進行資料傳遞，且同時支援 TCP 與 UDP 兩種傳輸協定，而兩種協定都可各自設定 8 組的連線通道；欲設定時，請分別於 **TCP 設定** 頁面與 **UDP 設定** 頁面做設定。



TCP 與 UDP 協定的參數幾乎相同，差別只在於 UDP 沒有「**連線保持時間**」的參數，參數說明如下。

- **遠端 IP 位址**：設定遠端裝置的 IP 位址。
- **遠端通訊埠**：設定在這組連線中，遠端裝置所使用的通訊埠編號，輸入範圍為 0 ~ 65535。
- **本機通訊埠**：設定在這組連線中，本地主機所使用的通訊埠編號，輸入範圍為 0 ~ 65535。
- **傳送資料位址**：設定在本地主機中，存放傳送資料的啟始位址。
- **傳送資料長度**：設定本地主機要傳送的資料長度，輸入範圍為 0 ~ 1000 Bytes。
- **接收資料位址**：設定在本地主機中，存放接收資料的啟始位址。
- **接收資料長度**：設定本地主機要接收的資料長度，輸入範圍為 0 ~ 1000 Bytes。
- **連線保持時間**：設定連線的最大閒置時間。當建立的連線超過此時間無資料傳輸時，主機便會自動中斷連線。

3

設定時，本機與遠端的通訊埠編號不可相同，且傳送與接收的裝置位址範圍不可重疊。而以上圖為例，遠端裝置的 IP 位址為 192.168.1.100，且開啟一個編號為 65500 的通訊接口，而本地主機也開啟一個編號為 65501 的通訊接口，兩者之間便將透過這組連線通道來進行資料的傳遞。

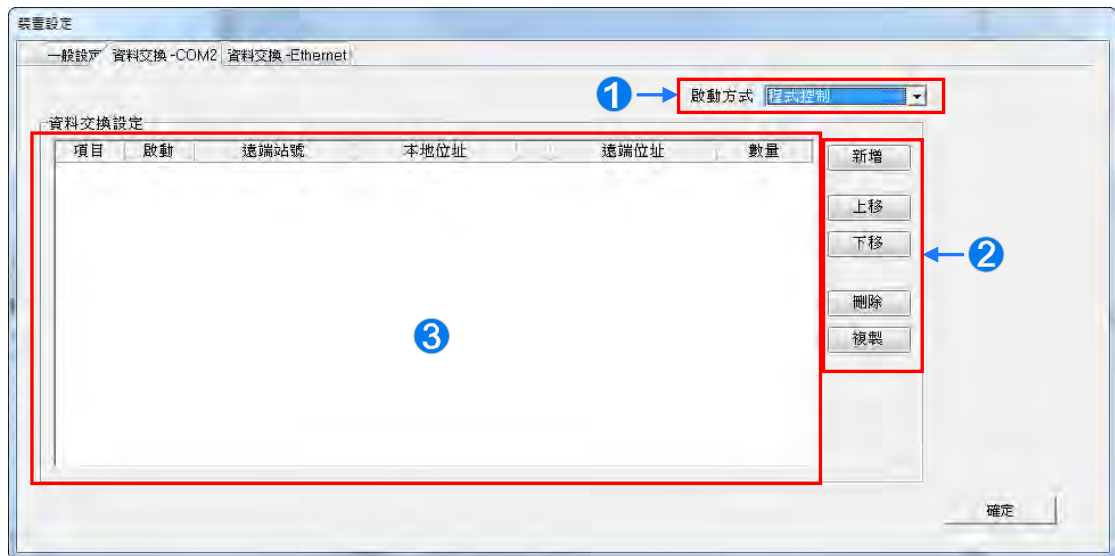
當本地主機要傳送資料給遠端裝置時，本地主機會先將資料存放在 D0~D9 共 10 個 WORD 的空間暫存之後，再將資料傳送出去；而當接收到來自遠端裝置的資料時，主機則會將接收的資料存放在裝置 D100 ~ D119 共 20 個 WORD 的空間中。

若收到的資料量大於所設定的長度時，主機會根據設定的長度，將最前面的 20 個資料存放在 D100 ~ D119 中，超出的部份則會直接將其丟棄；同樣的，若收到的資料量小於設定長度時，主機則會從設定的啟始位置 D100 依序存放資料，而未被存放新資料的位置則會保持原本的內容值。

當 TCP 通道連線已超過 30 秒未有任何的資料傳輸時，主機便會主動將主機端的通訊接口 (Socket) 關閉，並結束通道連線。

3.4.3 資料交換-COM2 與資料交換-Ethernet

此類參數分別位於「**資料交換-COM2**」與「**資料交換-Ethernet**」分頁，用於建立 AHxxEMC 系列機種內建通訊埠的資料交換表。AHxxEMC 系列機種可作為主站，對於作為從站的遠端設備進行資料交換，COM2 埠口透過 MODBUS 協定；而乙太網路埠口則透過 MODBUS TCP 協定。設定欄位兩分頁皆相同，如下圖。



3

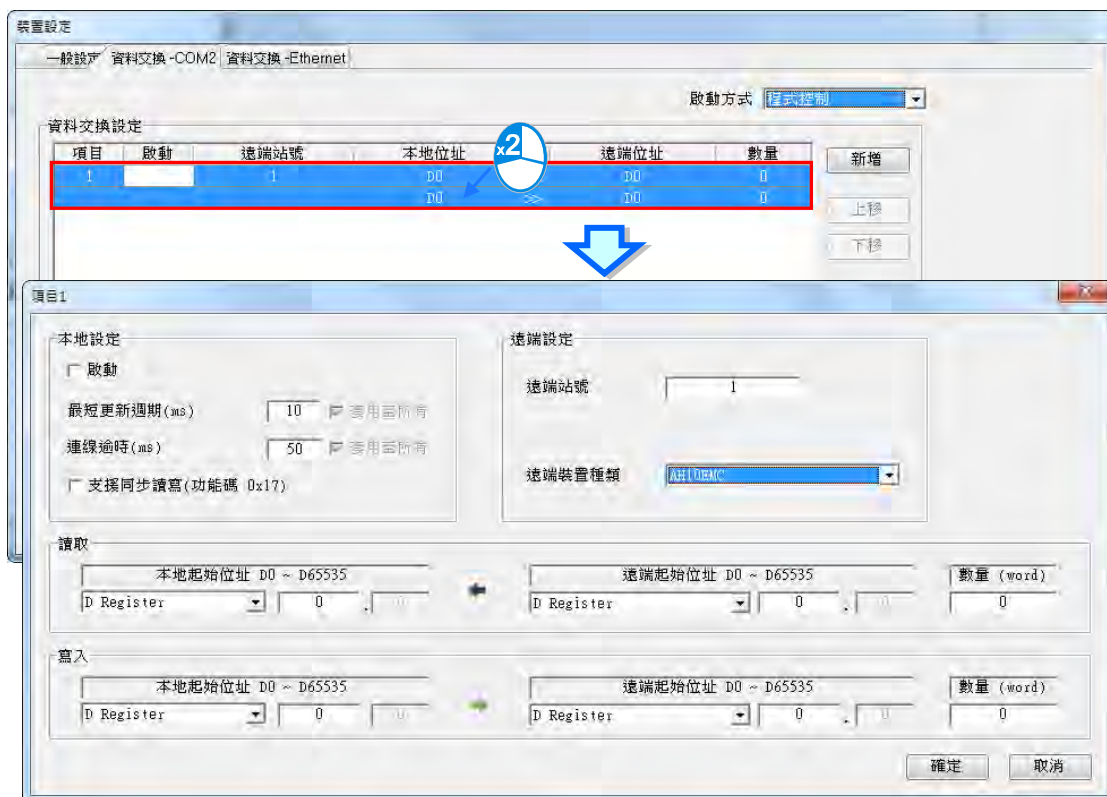
- ❶ 設定**啟動方式**為**程式控制**時為由 PLC 的程式控制是否執行設定的資料交換；**PLC 執行**為當 PLC 為 RUN 的狀態時會自動執行設定的資料交換；**永遠執行**則為當 PLC 上電時便會不斷執行資料交換。
- ❷ 按下「**新增**」按鈕便會在區域❸的資料交換表區域新增一列新的資料交換表。按下「**上移**」與「**下移**」按鈕可以對目前選擇的資料交換表往上一列或下一列搬移。按下「**刪除**」按鈕可刪除目前選擇的資料交換表。按下「**複製**」按鈕可以將目前選擇的資料交換表複製後貼上到最下方一列。
- ❸ 列出建立的資料交換表。說明如下表。

表格欄位	說明
項目	表示資料交換區塊的序號。
啟動	表示進行資料交換時，是否啟動此組資料交換表。
遠端站號/ IP 位址	註明此組資料交換表所屬的從站位址，同一個位址可規劃為多組資料交換表。 COM2 分頁註明為站號，乙太網路分頁則註明為 IP。
本地位址	註明主站於此組資料交換表中所使用的裝置位址範圍。
<< / >>	「<<」：輸入，表示此列所顯示的是主站讀取從站的資料區塊。 「>>」：輸出，表示此列所顯示的是主站寫入從站的資料區塊。
遠端位址	註明從站於此組資料交換表中所使用的裝置位址範圍。而當此組資料交換表的從站為自行定義的「MODBUS Device」裝置時，此範圍會以 16 進制的位址編號來顯示。
數量	註明此組交換表的長度，基本上與裝置範圍所計算的結果一致。

3.4.3.1 資料交換-COM2

在**資料交換-COM2** 分頁，以滑鼠左鍵雙擊欲設定的資料交換表，開啟資料交換表的設定項目，如下圖。勾選**啟動**代表欲啟動此組資料交換表，並且隨著前述的**啟動方式**啟動。**遠端站號**用於進行資料交換的目標從站設備站號。**最短更新週期 (ms)** 為設定此組資料交換表進行資料交換的時間週期。**連線逾時 (ms)** 為設定若目標設備超過此設定時間未回應，則為逾時。勾選**支援同步讀寫(功能碼 0x17)** 將利用特定的 MODBUS 功能碼，主站便可在一次的命令當中完成**讀與寫**的動作，以提高資料交換的效率；不過設定時，請確認參與資料交換的所有裝置都有支援同步讀寫的 MODBUS 功能碼，否則便可能在接收到主站的命令後，會因無法識別該功能碼而造成讀寫失敗。**裝置種類**為目標從站設備的機種，包括台達 PLC 與標準 MODBUS 設備。

3



- 「讀取」的資料區塊

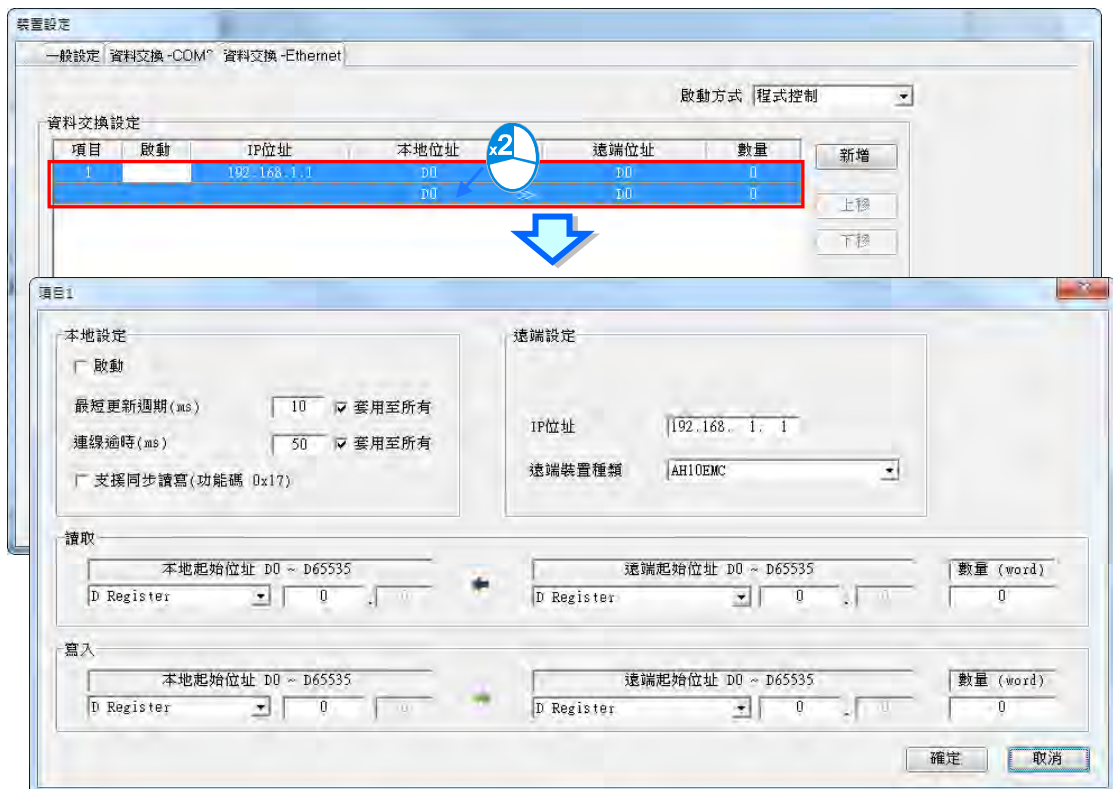
設定當 AHxxEMC 系列機種自遠端設備讀取資料時，AHxxEMC 系列機種用以存放資料的**裝置類別**、**裝置起始位址**，接著指定欲讀取的遠端設備**裝置類別**及**起始位置**。設定輸入的**資料長度**。

- 「寫入」的資料區塊

設定當 AHxxEMC 系列機種寫入資料至遠端設備時，AHxxEMC 系列機種來源資料的**裝置類別**、**起始位址**，接著指定被寫入資料的遠端設備**裝置類別**及**起始位置**。設定輸出的**資料長度**。

3.4.3.2 資料交換-Ethernet

在**資料交換-Ethernet**分頁，以滑鼠左鍵雙擊欲設定的資料交換表，開啟資料交換表的設定項目，如下圖。設定內容基本上與**資料交換-COM2**分頁相同，只有**裝置種類**所支援的目標從站設備的機種，為包括台達 PLC 與標準 MODBUS TCP 設備，以及 **IP 位址**用於輸入遠端從站設備的 IP 位址，其餘設定內容請參閱**資料交換-COM2**分頁內容。



若在上圖的**最短更新週期 (ms)**或**連線逾時 (ms)**欄位右方勾選**套用至所有資料**，按下「**確定**」按鈕後，則此兩設定將會被寫入到**資料交換-Ethernet**分頁其他的資料交換表中，也可不做勾選單獨設定此資料交換表。

3.5 DVP 系列機種的參數設定

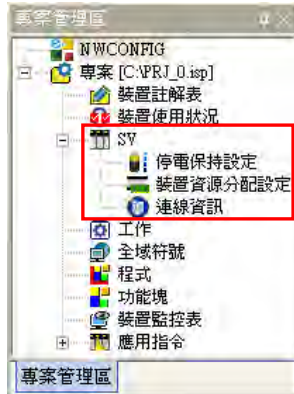
舊型的 DVP 系列機種沒有支援 HWCONFIG 工具，新一代的 DVP-ES3 機種有支援 HWCONFIG，DVP-ES3 機種的參數設定可參見第 3.5.4 節。

3.5.1 DVP 主機的系统管理工具

在 DVP 系列機種的專案中，將**專案管理區**的主機名稱展開後，於其下層便可發現與該主機相關的系統管理工具，而部分的功能則須與主機進行連線後方可操作。各項目的功能簡述如下：

- **停電保持設定**：設定主機停電保持裝置的起訖範圍。
- **裝置資源分配設定**：設定自動配置變數符號時所使用的裝置範圍。

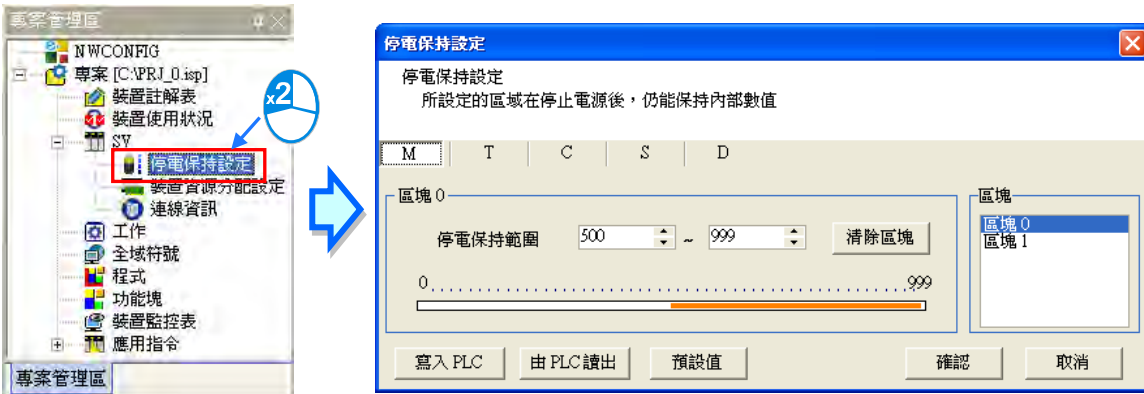
- **連線資訊**：檢視目前主機的系統設置與參數。



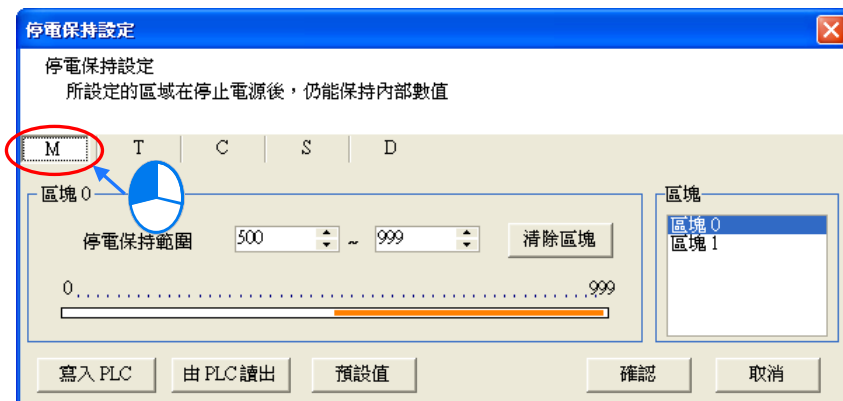
上述功能除了「裝置資源分配設定」將會於第 6.2.11 節的內容當中另行介紹之外，「停電保持設定」與「連線資訊」的功能將會於接下來的兩個小節當中逐一說明。

3.5.2 停電保持設定

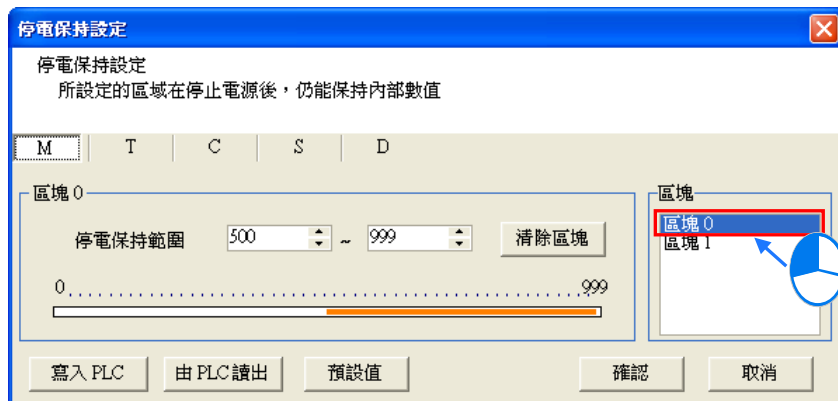
- (1) 於「停電保持設定」的項目上雙擊滑鼠左鍵之後，畫面便會出現該機種的停電保持設定視窗。



- (2) 首先用滑鼠點選欲設定的裝置類別。

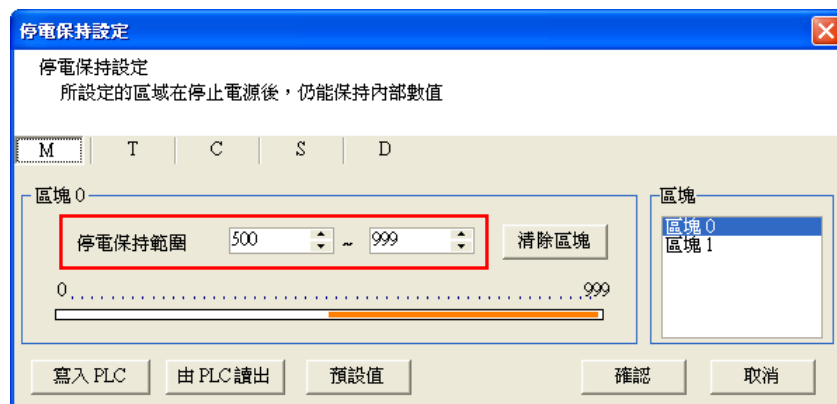


- (3) 因主機的停電保持區可能是一個不連續的範圍，因此在設定前請先選擇欲設定的範圍區塊。



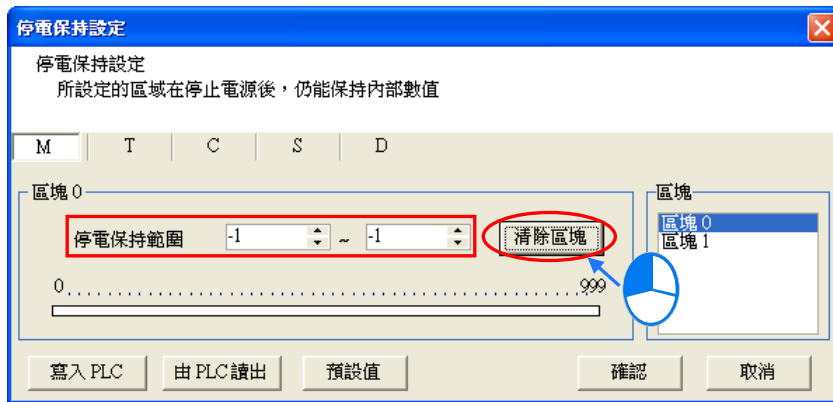
- (4) 在設定區域的下方會顯示一個條狀圖，而條狀圖兩側的數值則代表可設定的範圍極限，至於內部的橘色部份則代表已設為停電保持的範圍；但若下方的設定區域顯示為灰階時，便代表目前的區塊範圍於所選機種中無法設為停電保持。

設定時請直接於「**停電保持範圍**」的兩個欄位中分別輸入範圍的起訖位址，其中左側欄位為範圍起點，而右側則為範圍終點；輸入設定值時，左側起點欄位的數值必須小於右側的終點欄位，且不可超出條狀圖所顯示的範圍極限。

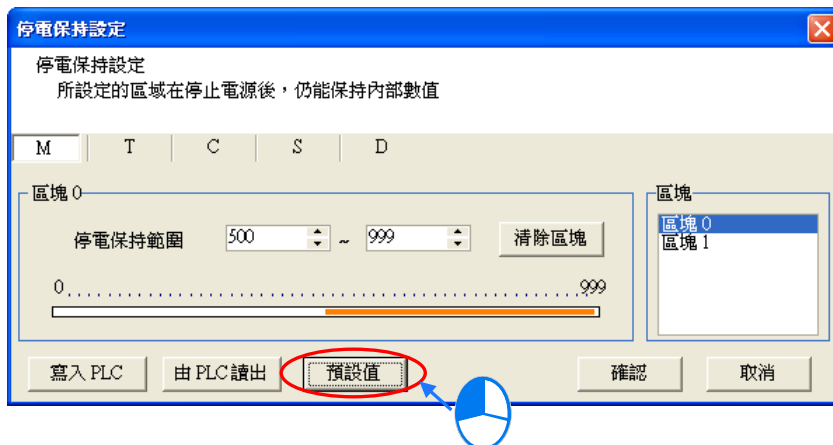


- (5) 若想讓該區塊的所有裝置都不具備停電保持的功能時，只要按下設定區域中的「清除區塊」按鈕即可，而此時的範圍欄位則會顯示-1 ~ -1。

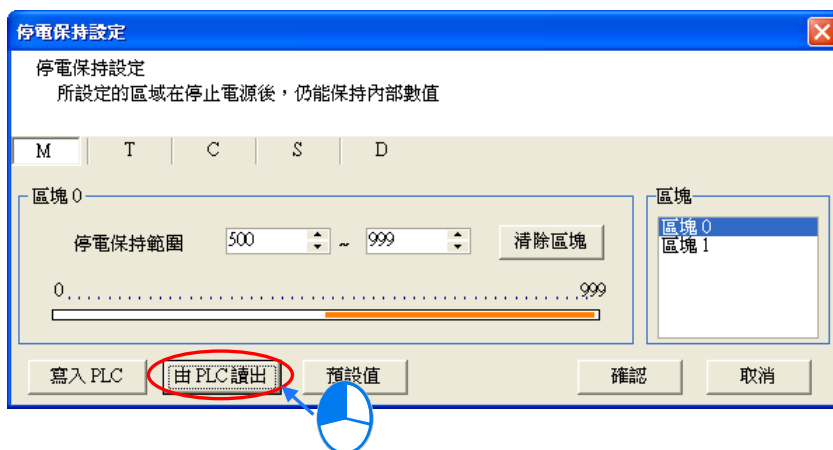
3



- (6) 若想讓設定恢復為系統預設值時，請按下「預設值」按鈕即可，而此時所有裝置的停電保持設定都會恢復為系統預設值。



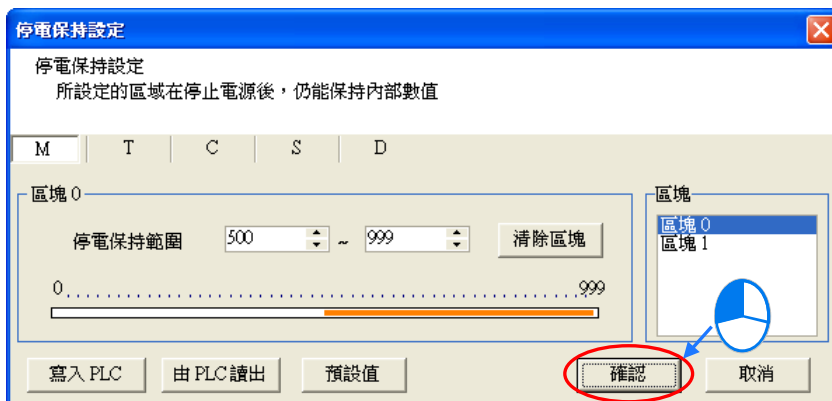
- (7) 在 ISPSOft 可與 PLC 主機正常連線的狀況下，按下「由 PLC 讀出」按鈕便可將主機中所有裝置的停電保持設定讀取回來。



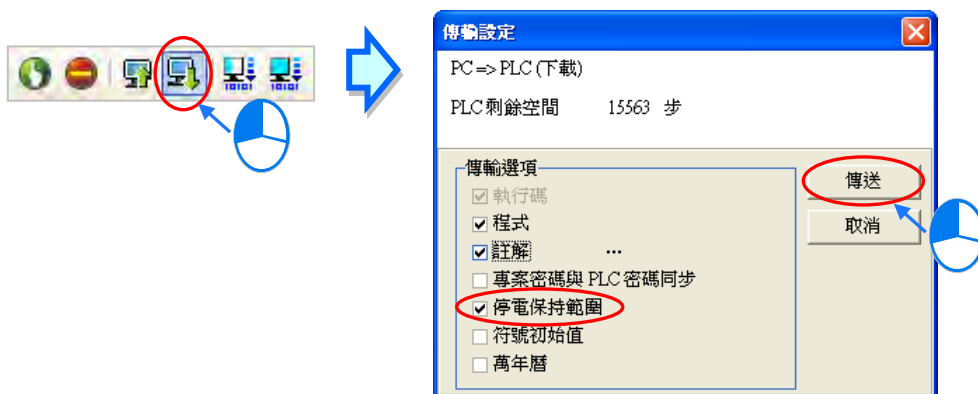
- (8) 完成所有的設定後，在 ISPSOft 可與 PLC 主機正常連線的狀況下，按下「寫入 PLC」按鈕便可將所有裝置的停電保持設定立即寫入主機當中，而此時系統便會自動將設定視窗關閉並記憶所有的設定值，之後當儲存 ISPSOft 的專案時，此處的設定值亦會一併被儲存。



- (9) 若暫時不須要將設定值寫入 PLC 主機時，按下「確認」鍵便可套用所有的設定，而之後當儲存 ISPSOft 的專案時，此處的設定值亦會一併被儲存。

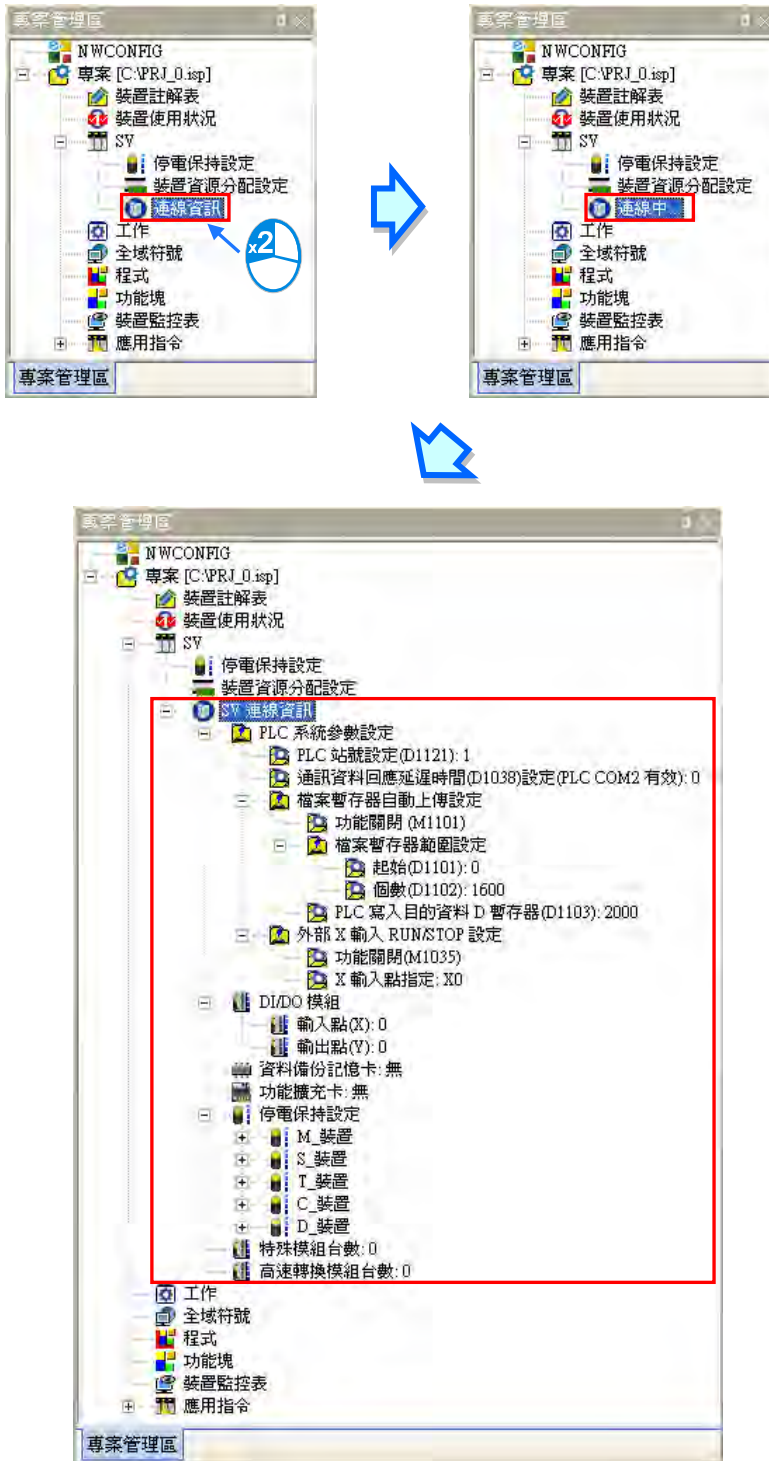


- (10) 除了可透過設定視窗的功能將設定值寫入主機之外，亦可於下載專案程式時，將該專案中所設定的停電保持範圍一併寫入主機當中，操作時僅須將傳輸選項中的「停電保持範圍」勾選起來，之後按下「傳送」便可將資料下載至主機中。關於下載的詳細操作將於第 17.1.3 節中另行介紹。



3.5.3 連線資訊

在 ISPSOft 可與 PLC 主機正常連線的狀況下，於專案管理區的「連線資訊」項目上雙擊滑鼠左鍵，之後會將 PLC 主機中的系統參數擷取回來，並顯示於「連線資訊」項目下，此處的資訊僅可檢視。



3.5.4 DVP-ES3 機種的主機參數設定

雙擊專案管理區的「HWCONFIG」之後，於「參數設定」點擊滑鼠左鍵，接著系統便會自動開啟 CPU 主機參數設定頁面。

⚠ 設定主機參數前，請務必查閱各機種的操作手冊，並確認所設定的參數對主機本身以及整體系統的影響，以避免造成系統或人員的傷害。

*參數設定完畢之後，仍必須被下載至主機當中才可生效，下載的方式請參考第 3.8.4 節。



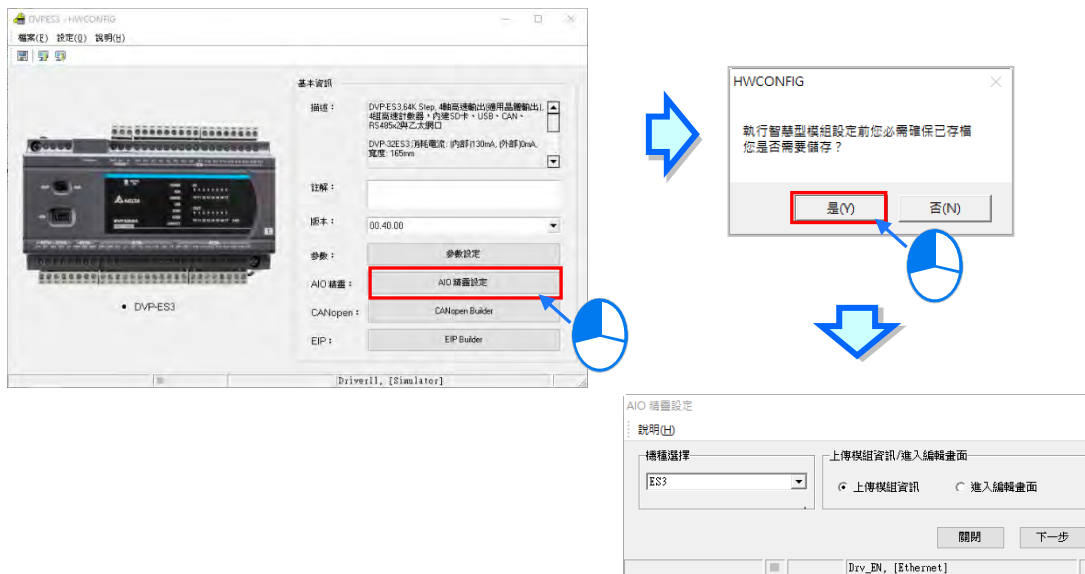
3

裝置設定頁面中亦根據參數類別而區分為許多的分頁，操作時可配合視窗上方的主類別標籤以及位於每個主類別頁面中，頁面左方的項目階層切換不同的設定頁面。頁面右方標題列依序列出參數名稱，及供使用者填入的設定值欄位，設定值的單位及設定值的預設值、最大值與最小值。

3



DVP-ES3 機種需透過 AIO 精靈設定，來配置右側模組，而這些參數亦將決定實際運作時，各個模組所表現出來的功能與特性。欲進行模組參數的設定時，請於「AIO 精靈設定」點擊滑鼠左鍵，之後系統便會開啟 AIO 精靈設定，接著可以點選「上傳模組資訊」將配置由 PLC 上載回來，或選取「進入編輯畫面」手動建立模組。如下圖。



3.6 DVPxxMC 機種的 PLC 主機參數設定

3.6.1 開啟 PLC 主機的參數設定頁面

啟動 HWCONFIG 之後，於「參數設定」點擊滑鼠左鍵，接著系統便會自動開啟 CPU 主機的參數設定頁面，且隨著主機型號的不同，其設定的參數與顯示的畫面也會有所差異。

! 設定主機參數前，請務必查閱各機種的操作手冊，並確認所設定的參數對主機本身以及整體系統的影響，以避免造成系統或人員的傷害。

*.參數設定完畢之後，仍必須被下載至主機當中才可生效，下載的方式請參考第 3.8.4 節。

3



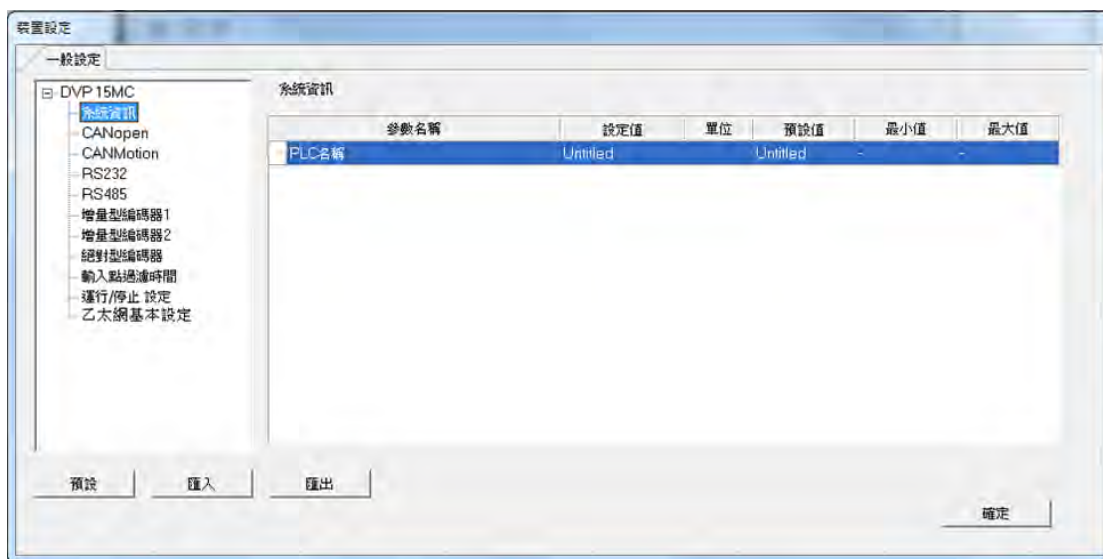
參數設定頁面中亦根據參數類別而區分為許多的分頁，操作時可配合視窗上方的**主類別標籤**以及位於每個主類別頁面中的項目階層切換不同的設定頁面。頁面右方標題列依序列出**參數名稱**，及供使用者填入的**設定值欄位**，設定值的**單位**及設定值的**預設值**、**最大值**與**最小值**。



3.6.2 一般設定

3.6.2.1 一般設定-系統資訊

此類參數位於「一般設定」分頁的「系統資訊」項目中，此頁面內只有「PLC 名稱」，而欄最多可輸入 15 字元。且所輸入的字元並沒有特別的制限，包括特殊字元及空白鍵皆可輸入，不過須注意一個中文字將會佔用兩個字元。



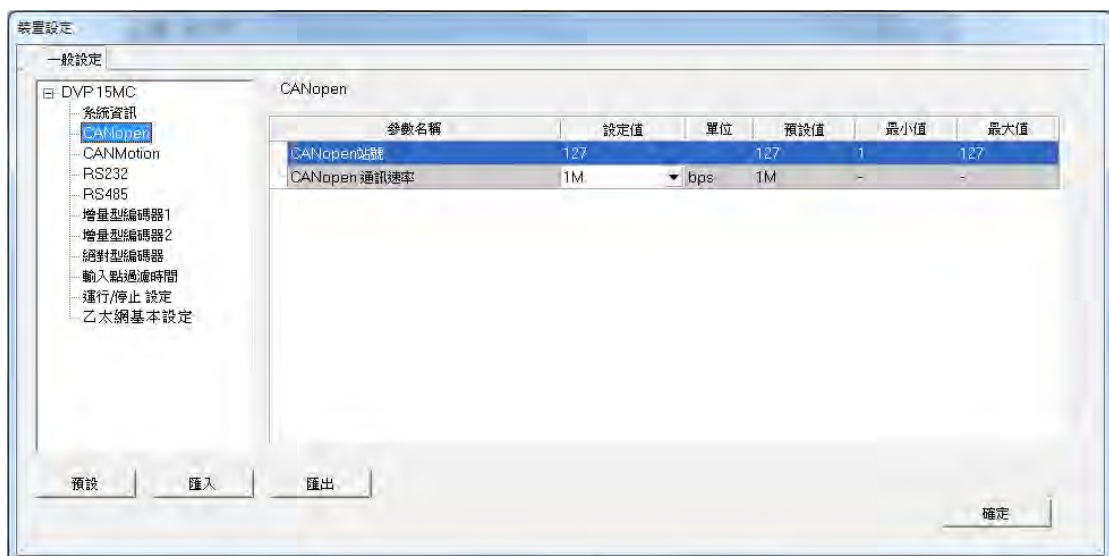
當新增一個 ISPSOft 專案時，系統便會以專案的名稱作為主機의 預設名稱，且會顯示於專案管理區的機種名稱後方，而後續則可於此處重新定義。



PLC 標籤名稱的功能主要是用來做為裝置識別，尤其是在聯網系統的應用或是在進行連線操作時，藉由確認 PLC 標籤名稱的方式便可防止使用者的誤操作；當使用者要對主機進行資料的上下載或其它的連線操作時，若發現操作對象的主機標籤名稱與專案中的標籤名稱不符時，系統便會提醒使用者做再次的確認，以避免對其他的主機造成非預期的影響。

3.6.2.2 一般設定-CANopen

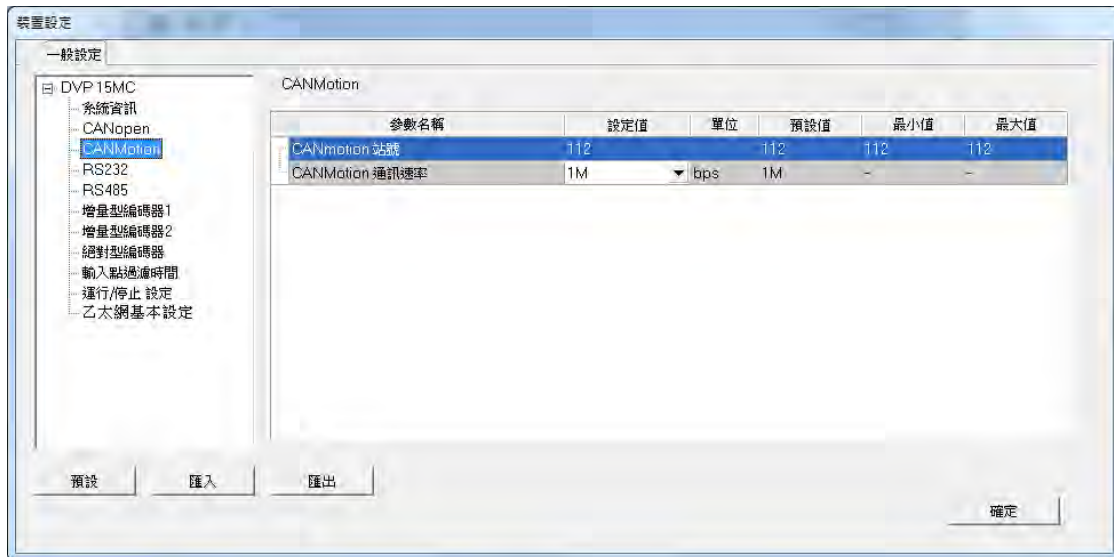
「CANopen」頁面設定各項參數如下，可藉由下拉選單或輸入框設定適當的參數值。



此處用作設定主機之 CANopen 通訊口站號和通訊口通訊波特率，通訊波特率可選擇 20k bps、50k bps、125k bps、250k bps、500k bps、800k bps 及 1Mbps。

3.6.2.3 一般設定-CANMotion

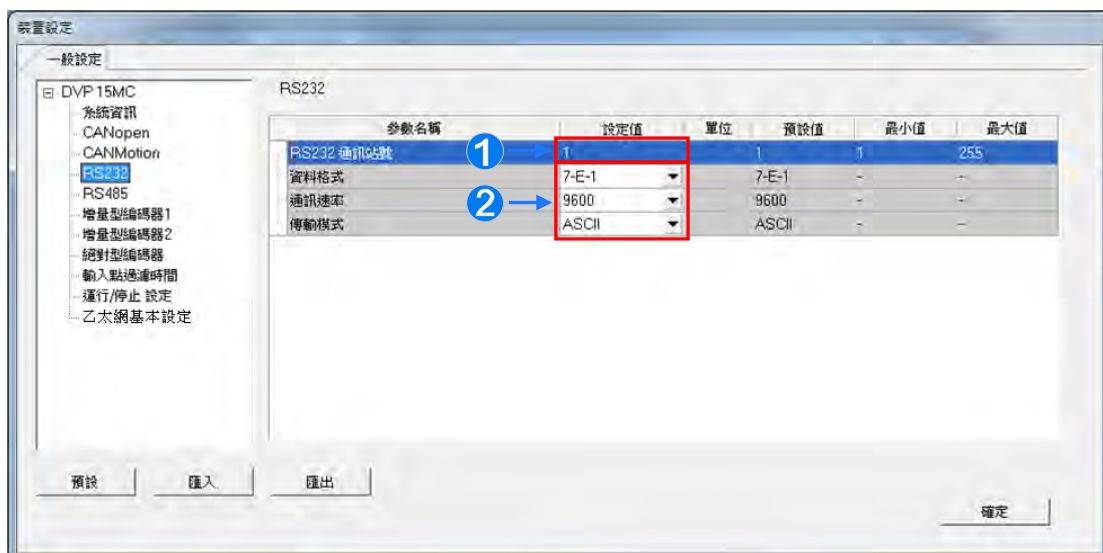
「CANMotion」頁面設定各項參數如下，可藉由下拉選單或輸入框設定適當的參數值。



此處用作主機之 Motion 通訊口站號，預設為 112，不可更改；但可做通訊口通訊速率設定，通訊速率可選擇 1Mbps 或 500kbps。

3.6.2.4 一般設定-RS232

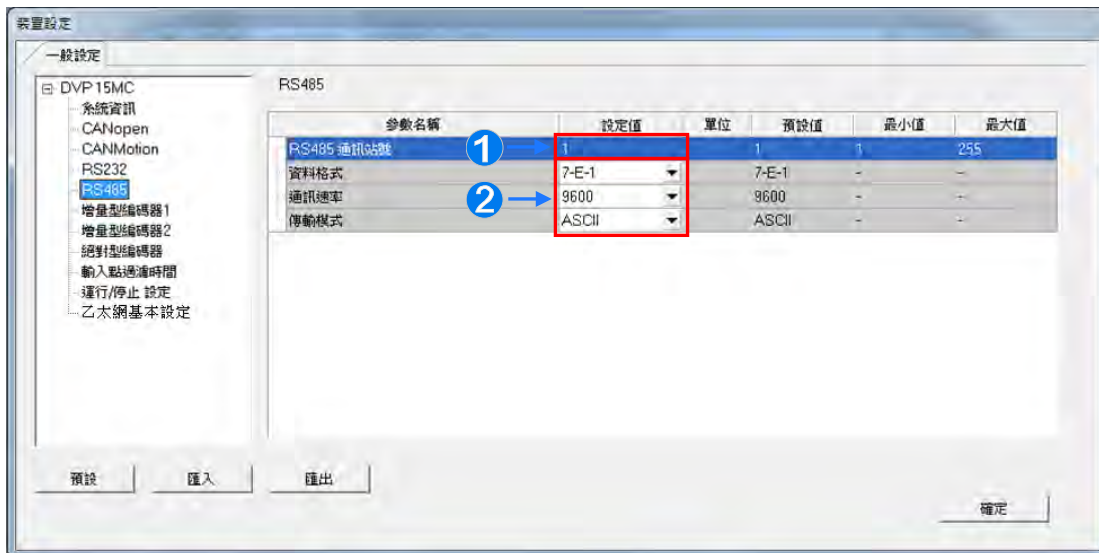
「RS232」此頁面用作主機 RS232 通訊埠設定。



- ❶ 此處用作設定 RS232 通訊站號。
- ❷ 此處用作設定 RS232 資料格式、通訊速率及傳輸模式，而通訊速率有 9600、19200、38400、57600 及 115200 可選擇。

3.6.2.5 一般設定-RS485

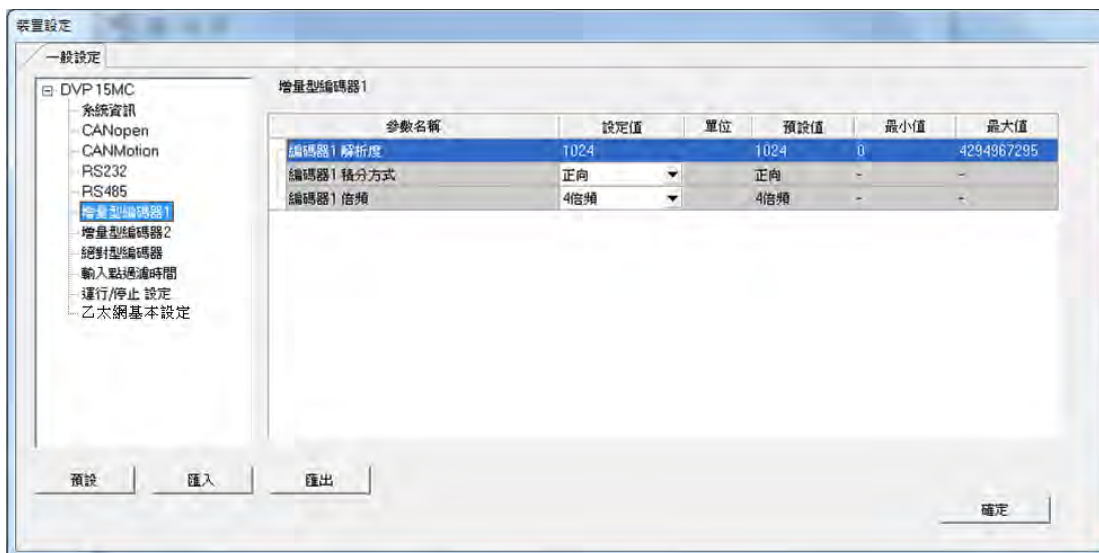
「RS485」此頁面用作主機 RS485 通訊埠設定。



- ① 此處用作設定 RS485 通訊站號。
- ② 此處用作設定 RS485 資料格式、通訊速率及傳輸模式，而通訊速率有 9600、19200、38400、57600 及 115200 可選擇。

3.6.2.6 一般設定-增量型編碼器 1

「增量型編碼器 1」此頁面用作連接第一組增量型編碼器的設定。



此處用作設定增量型編碼器 1 解析度、積分方式及倍頻。

➤ 增量型編碼器 1 解析度

此處用作「增量型編碼器 1 解析度」之設定，預設為 1024。

➤ 增量型編碼器 1 積分方式

此處提供「反向」及「正向」之選擇。

➤ 增量型編碼器 1 倍頻

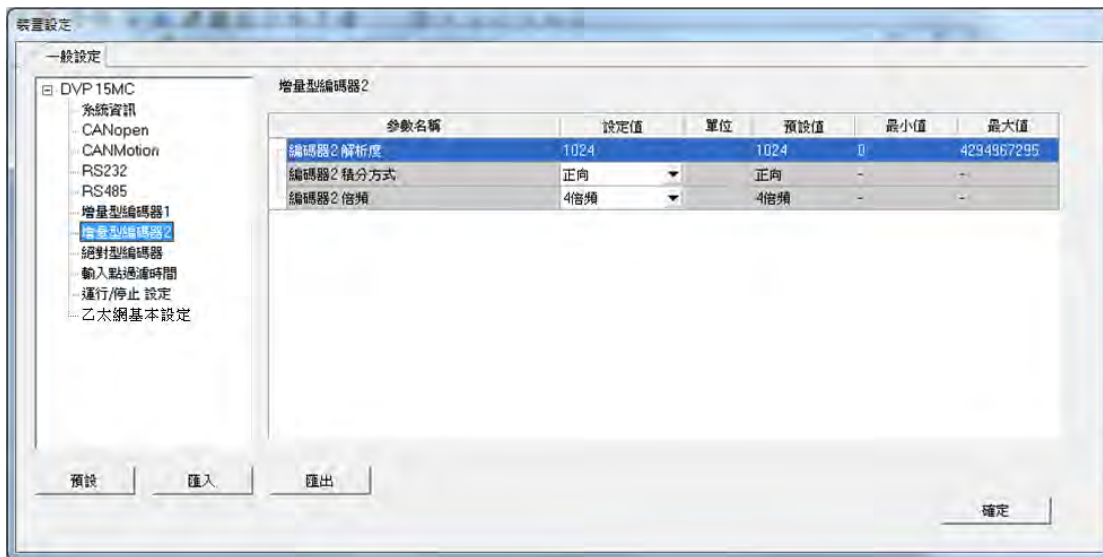
此處提供「1 倍頻」、「2 倍頻」及「4 倍頻」之選擇。

而增量型編碼器之積分方式及倍頻具體說明如下表。

積分方式	含義	計數方式
正向 1 倍頻	偵測到 B 相上緣時計數	增計數
正向 2 倍頻	偵測到 B 相上緣時和下緣時均計數	增計數
正向 4 倍頻	偵測到 A 相和 B 相上緣時和下緣時均計數	增計數
反向 1 倍頻	偵測到 B 相上緣時計數	減計數
反向 2 倍頻	偵測到 B 相上緣時和下緣時均計數	減計數
反向 4 倍頻	偵測到 A 相和 B 相上緣時和下緣時均計數	減計數

3.6.2.7 一般設定-增量型編碼器 2

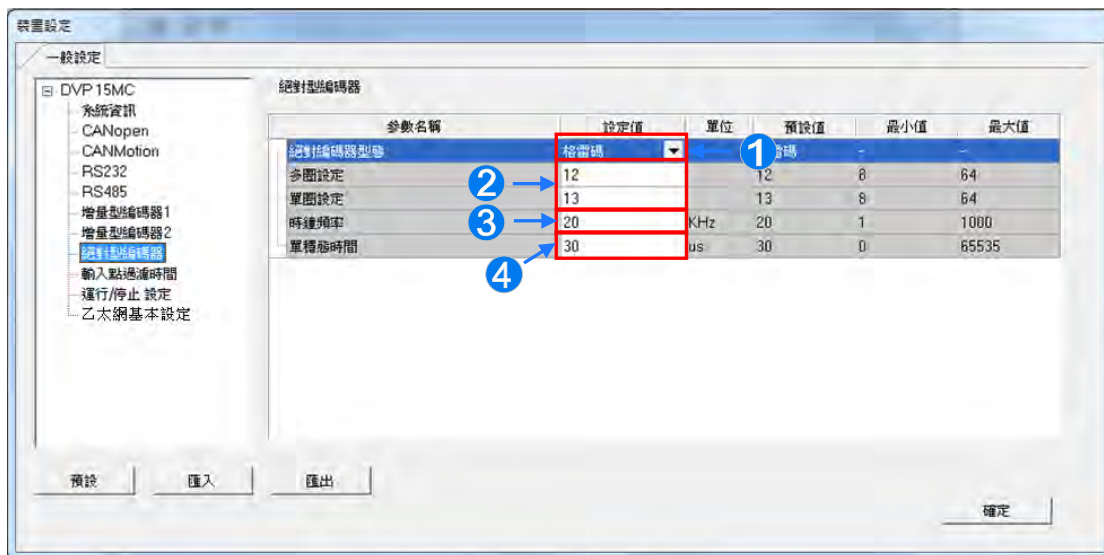
「增量型編碼器 2」此頁面用作連接第二組增量型編碼器的設定。



此處用作設定增量型編碼器 2 解析度、積分方式及倍頻，其設定方式同增量型編碼器 1，可參考第 3.6.2.6 節。

3.6.2.8 一般設定-絕對型編碼器

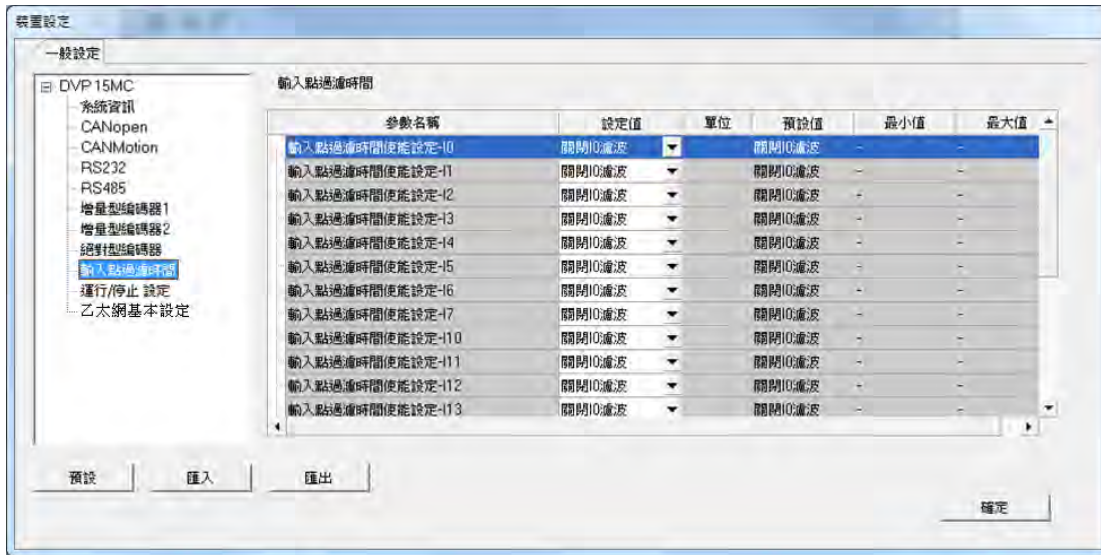
「絕對型編碼器」此頁面用作連接絕對型編碼器的設定。



- ❶ 此項目為選擇絕對型編碼器的輸出為「二進位」或「格雷碼」，此參數由編碼器決定。
- ❷ 此項目之「多圈設定」為用作設定多圈解析度，絕對型編碼器脈波數超過允許的最大圈後重新計數，若設定為 N 則表示可以轉的圈數為 2 的 N 次方，此參數由編碼器決定；而「單圈設定」項目用作設定單圈解析度，絕對型編碼器轉一圈產生的脈波數，若設定為 N 則表示單圈產生的脈波數為 2 的 N 次方，此參數由編碼器決定。
- ❸ 「時間頻率」項目為設定資料傳輸速率。
- ❹ 「單穩態時間」項目，單穩態時間由時鐘序列最後一個脈波下緣開始計時，它決定了最低傳輸頻率。

3.6.2.9 一般設定-輸入點過濾時間

「輸入點過濾時間」此頁面用於設定 DVPxxMC 輸入點過濾設定和 IO 刷新跟隨 Task 之機制。



➤ 輸入點過濾時間使能設定-I0~I7

此項濾波器為 I0~I7 這些點對短脈波訊號的過濾，從而降低輸入干擾訊號的影響，通過增大濾波值，可減少輸入訊號的抖動或外部干擾的影響。可由下拉選單，選擇開啟或關閉 IO 濾波。

➤ 輸入點過濾時間使能設定-I10~I17

此項濾波器為 I10~I17 這些點對短脈波訊號的過濾，從而降低輸入干擾訊號的影響，通過增大濾波值，可減少輸入訊號的抖動或外部干擾的影響。可由下拉選單，選擇開啟或關閉 IO 濾波。

➤ I0~I7 過濾時間因數

此處為設定 I0~I7 過濾時間因數 0~255，過濾時間： $t=31\mu s * (\text{過濾時間因數})$ 。因此過濾時間是 31us 的倍數，過濾時間因數預設值為 0。

➤ I10~I13 過濾時間因數

此處為設定 I10~I13 過濾時間因數 0~255，過濾時間： $t=31\mu s * (\text{過濾時間因數})$ 。因此過濾時間是 31us 的倍數，過濾時間因數預設值為 0。

➤ I14~I17 過濾時間因數

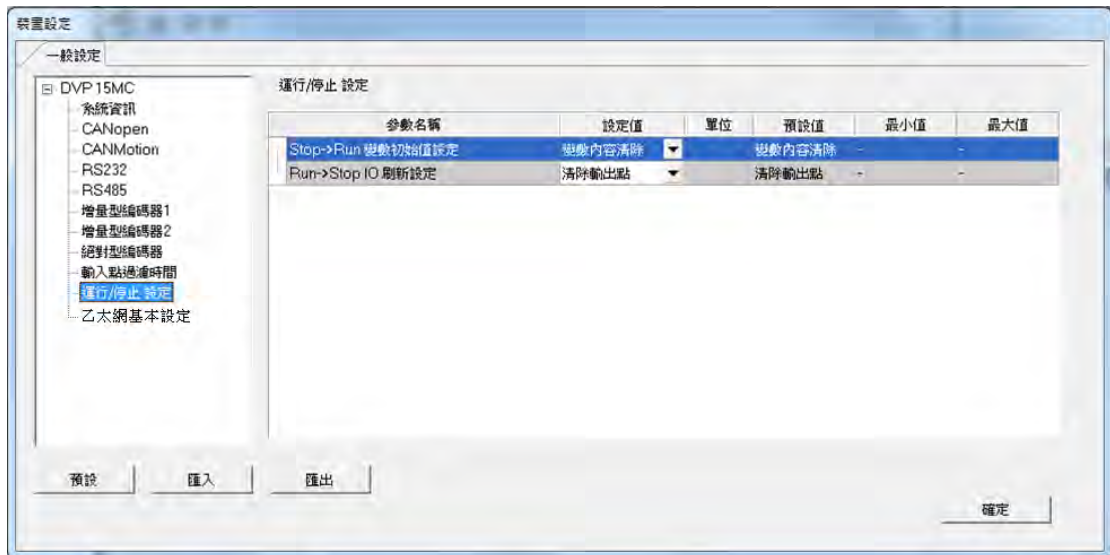
此處為設定 I14~I17 過濾時間因數 0~255，過濾時間： $t=31\mu s * (\text{過濾時間因數})$ 。因此過濾時間是 31us 的倍數，過濾時間因數預設值為 0。

➤ IO 刷新跟隨 Task

I/O 包括 DVPxxMC 本身 I/O 點和左右擴充模組資料、CANopen 資料。選擇「FF」則 IO 資料會在系統處理中刷新；選擇其他指定 Task 則 IO 資料在該指定 Task 執行前更新。

3.6.2.10 一般設定-運轉 / 停止設定

「運轉 / 停止」此頁面用於設定 IO 是否刷新以及變數裝置是否恢復初始值。



➤ Stop → Run 變數初始值設定

此項用可選擇「變數內容清除」或「變數內容保持」。

- (1) 當選擇「變數內容清除」，由停止到運行時，非停電保持型變數和非停電保持裝置的值會恢復為初始值或預設值，停電保持型的變數 (VAR_RETAIN 型) 和停電保持區的裝置並不會恢復初始值。

非停電保持型的變數綁定停電保持裝置時，若該變數設定初始值，則當由停止狀態變為運行狀態時或重上電後，變數值會恢復為初始值；若該變數未設定初始值，則由停止狀態變為運行狀態時或重上電後，該變數的值會變為綁定裝置中存儲的值。

- (2) 當選擇「變數內容保持」，由停止到運行時，變數和裝置會保持停止前的值。軟體重新下載時，由停止到運行時，非停電保持型變數和非停電保持區裝置的值會恢復為初始值或預設值；停電保持型變數 (VAR_RETAIN 型) 和保持區裝置不會初始化。

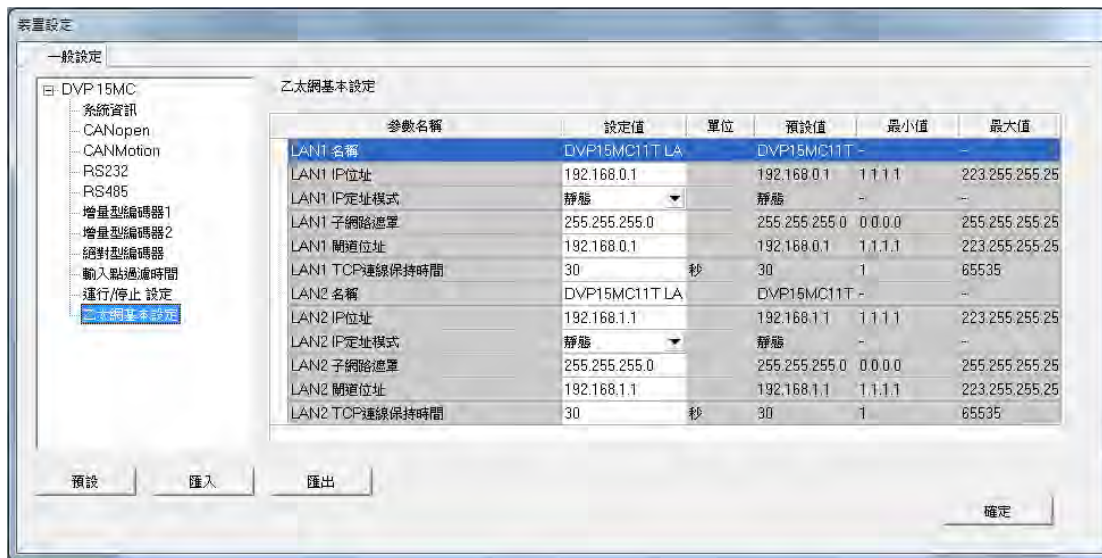
➤ Run → Stop IO 刷新設定

此項用作主機在從運行到停止時，IO 刷新的設定，可選擇「清除輸出點」或「保持輸出點」。

- (1) 當選擇「清除輸出點」，當處於停止狀態時，會刪除現在 IO 資料，並且不在刷新 IO 資料。
- (2) 當選擇「保持輸出點」，由運行到停止時，不會刪除現在 IO 資料，並且在處於停止狀態時仍然保持 IO 刷新。

3.6.2.11 一般設定-乙太網路基本設定

於項目列表中點選「乙太網基本設定」項目即可進入。主要用於設定主機的兩個乙太網路 (Ethernet) 埠之通訊參數，15MC 支持 LAN1/LAN2；50MC 只有 LAN1 另一個 Port 為 EtherCAT。



3

3.7 模組參數設定

3.7.1 設定 AH5x0 與 AH5x1 模組內部參數

在 HWCONFIG 中，使用者可各別為每個配置的模組設定其內部參數，而這些參數亦將決定實際運作時，各個模組所表現出來的功能與特性。(AH560 備援系列機種目前模組內部參數設定可比照 AH5x1 系列。) 欲進行模組參數的設定時，請於系統配置區的模組圖示上雙擊滑鼠左鍵，之後系統便會自動開啟該模組的參數設定視窗。各機種系列的編輯介面略有不同，如下圖。



⚠ 此處的參數設定主要還是依據模組本身所支援的功能，因此設定前請務必查閱該模組的操作手冊，並確認所設定的參數對模組本身以及整體系統的影響，以避免造成系統或人員的傷害。

AH5x0 與 AH5x1 機種在設定視窗的主畫面中包含「MDS 資訊」與「常態交換區」兩個頁面，並可藉由上方的標籤來進行切換。而其中「MDS 資訊」頁面主要會顯示目前的 MDS 文件版本，而 MDS 版本亦會決定該模組可設定的參數數量與參數值範圍。此外，若某個專案檔中的 HWCONFIG 參數是以較新的 MDS 文件所建立的，則在開啟該專案檔的 HWCONFIG 時，系統便會提示使用者更新 MDS 文件的檔案。



MDS 資訊 常態交換區

模組名稱	AH04AD-5A
MDS 版本	1.00.00
MDS 建立日期	2012/07/14

「常態交換區」頁面則是顯示在該模組的輸入/輸出裝置範圍中，各裝置位址所實際對應的功能。

資訊: 背板 1

插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH04AD-5A	1.00	4 通道類比輸入 16 位元	D0 ~ D7		
1						
2						
3						

MDS 資訊 常態交換區

描述	位址
▶ 通道 0 輸入值	D0 ~ D1
通道 1 輸入值	D2 ~ D3
通道 2 輸入值	D4 ~ D5
通道 3 輸入值	D6 ~ D7

欲設定參數時，請直接於左側的參數分類上點選取參數類別，之後於右側便會出現對應的參數設定表格；而不論所選的參數類別為何，表格本身的格式都是固定的，差別僅在於顯示的內容；此外，依據該模組所配置的韌體版本，對應的參數項目與可設的參數值可能也會有所不同。



以下則是表格中各個欄位的相關說明，而當某個欄位的底色呈現灰階時，便代表該欄位無法設定。

➤ 描述

此欄位所顯示的為各個參數的名稱或敘述。

描述	位址	監控	初始值	註解
通道 0 平均次數	...	<input type="checkbox"/>	10	通道 0 平均次數
通道 1 平均次數	...	<input type="checkbox"/>	10	通道 1 平均次數
通道 2 平均次數	...	<input type="checkbox"/>	10	通道 2 平均次數
通道 3 平均次數	...	<input type="checkbox"/>	10	通道 3 平均次數

➤ 位址

於此欄位設定 **D 對應裝置** 的位址。此功能主要用於對模組內部非經常性更新的參數進行存取，當為某個模組中的參數指定了 **D 對應裝置** 之後，若 HWCONFIG 的參數有被正確下載至主機中，則當系統開始運作時，CPU 主機中的 **D 對應裝置** 便會與指定的模組參數保持同步，因此使用者可直接透過 **D 對應裝置** 而間接的存取模組內部的資料，透過此方法會較傳統採用 **FROM / TO** 指令來存取模組內部資料的方法來的更有效率。此外，每個機種都有其 **D 對應裝置** 的指定數量上限，設定時請務必查閱各機種的相關使用說明。

如下述範例，我們於 **AH04AD-5A** 的模組參數設定視窗中，將 D3000 定義為模組參數 - 「**通道 0 平均次數**」的 **D 對應裝置**，因此後續當我們將 HWCONFIG 的參數下載至主機之後，我們便可直接修改 CPU 主機的 D3000 內容值而間接變更 **AH04AD-5A** 模組的**通道 0 平均次數**。

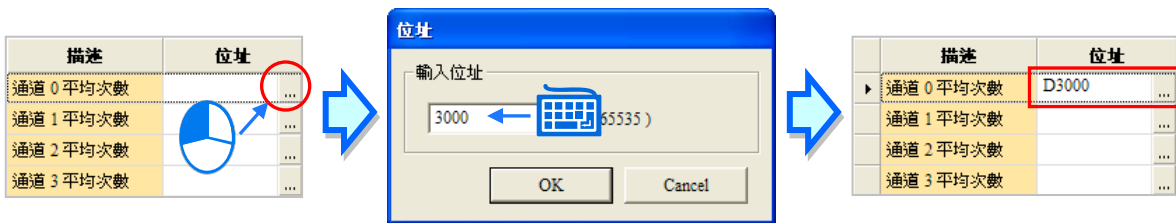
此外，運作期間若對 **D 對應裝置** 寫入模組參數所不允許的值時，系統會立即將兩者修正回原來的值。



*.關於 FROM / TO 指令的相關說明請參照各機種的指令手冊。

3

欲指定 **D** 對應裝置時，請先於該參數的「位址」欄位中，點擊右側的 **...** 按鈕，之後於位址視窗中輸入欲使用的 **D** 裝置位址即可，但須注意不可重複使用在其它 **D** 對應裝置或輸出輸入裝置範圍中使用過的位址；而要解除 **D** 對應裝置時，只要以同樣的方式開啟位址視窗後，再將位址的數值清除即可。



當「位址」欄位的底色為灰階時，代表該參數不可指定 **D** 對應裝置，但若是如下圖的例子，其顯示的樣式則代表這些參數的數值為位元格式，而第一個欄位中所指定的 **D** 對應裝置，其裝置內容值中的某些位元狀態便會對應至下列的各個參數，至於其對應關係則必須參考各模組的說明文件。

描述	位址	監控	初始值	註解
警示燈號：通道 0 輸入信號超出硬體範圍	...	<input type="checkbox"/>	<input type="checkbox"/> 閃爍	警示燈號：通道 0 輸入
警示燈號：通道 1 輸入信號超出硬體範圍		<input type="checkbox"/>	<input type="checkbox"/> 閃爍	警示燈號：通道 1 輸入
警示燈號：通道 2 輸入信號超出硬體範圍		<input type="checkbox"/>	<input type="checkbox"/> 閃爍	警示燈號：通道 2 輸入
警示燈號：通道 3 輸入信號超出硬體範圍		<input type="checkbox"/>	<input type="checkbox"/> 閃爍	警示燈號：通道 3 輸入

➤ 監控

在 HWCONFIG 中有提供**監控表**的功能，使用者可透過**監控表**中的 **D** 對應裝置而間接存取模組內部的資料；因此若模組的參數有設定 **D** 對應裝置並將該參數的「**監控**」欄打勾，則之後開啟**監控表**時，所指定的 **D** 對應裝置便會顯示在**監控表**中；對於未設定 **D** 對應裝置的參數，其「**監控**」欄則無法勾選。

以下圖為例，雖然每個參數都有設定 **D** 對應裝置，但只有前面兩個參數有勾選**監控**的功能，因此當開啟 HWCONFIG 的**監控表**之後，只有 D3000 與 D3001 會被列於**監控表**中。關於 HWCONFIG 的**監控表**功能將於第 3.8.6 節當中另行介紹。

描述	位址	監控	初始值	註解
通道 0 平均次數	D3000	<input checked="" type="checkbox"/>	20	通道 0 平均次數
通道 1 平均次數	D3001	<input checked="" type="checkbox"/>	10	通道 1 平均次數
通道 2 平均次數	D3002	<input type="checkbox"/>	10	通道 2 平均次數
通道 3 平均次數	D3003	<input type="checkbox"/>	10	通道 3 平均次數

➤ 初始值

此欄位用於設定所選參數的初始值，後續當下載 HWCONFIG 的參數時，所設定的初始值亦會一併被寫入模組當中；而依據所選參數的不同，此欄位有可能是直接輸入或是下拉選單的格式；關於允許輸入的數值範圍與初始值下載之後的處理機制，請參考各機種的使用說明。

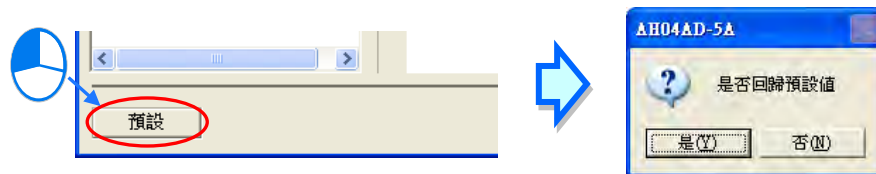
描述	位址	監控	初始值	註解
通道 0 平均次數	D3000	...	20	通道 0 平均次數
通道 1 平均次數		...	10	通道 1 平均次數
通道 2 平均次數		...	10	通道 2 平均次數
通道 3 平均次數		...	10	通道 3 平均次數

➤ 註解

點選此欄位之後，接著便可直接編輯該參數的註解說明，且最多可輸入 32 個字元，其中一個中文字將佔用兩個字元，而「註解」欄的預設值通常與「描述」欄的內容相同。

描述	位址	監控	初始值	註解
通道 0 平均次數	D3000	...	20	Comment...
通道 1 平均次數	D3001	...	10	通道 1 平均次數
通道 2 平均次數	D3002	...	10	通道 2 平均次數
通道 3 平均次數	D3003	...	10	通道 3 平均次數

當按下參數設定視窗左下角的「預設」按鈕後，該模組的所有參數值便會全部恢復為系統預設值。



3.7.2 設定 AHxxEMC 機種模組內部參數

在 HWCONFIG 中，使用者可各別為每個配置的模組設定其內部參數，而這些參數亦將決定實際運作時，各個模組所表現出來的功能與特性。欲進行模組參數的設定時，請於**系統配置區**的模組圖示上雙擊滑鼠左鍵，之後系統便會自動開啟該模組的參數設定視窗。各機種系列的編輯介面略有不同，如下圖。



! 此處的參數設定主要還是依據模組本身所支援的功能，因此設定前請務必查閱該模組的操作手冊，並確認所設定的參數對模組本身以及整體系統的影響，以避免造成系統或人員的傷害。

AHxxEMC 系列機種在設定視窗的主畫面中包含「裝置資訊」與「常態交換區」兩個頁面，並可藉由上方的標籤來進行切換。而其中「裝置資訊」頁面主要會顯示模組的相關資訊與外觀，與目前的 DDF 文件版本，而 DDF 版本亦會決定該模組可設定的參數數量與參數值範圍。另外，當連線時便會顯示主機的韌體版本與硬體版本。

「常態交換區」頁面則是顯示在該模組的輸入/輸出裝置範圍中，各裝置位址所實際對應的功能。

插槽編號	類別	名稱	輸入裝置範圍	輸出裝置範圍	註解
	電源	AS-PS02			
	CPU	AS332P	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
	模組資訊1	類比 I/O 模組	D28000 ~ D28019		

描述	位址
狀態碼	D28000 ~ D28001
通道 1 輸入值	D28002 ~ D28003
通道 2 輸入值	D28004 ~ D28005
通道 3 輸入值	D28006 ~ D28007
通道 4 輸入值	D28008 ~ D28009

3

欲設定參數時，請直接於左側的參數分類上點選取參數類別，之後於右側便會出現對應的參數設定表格；而不論所選的參數類別為何，表格本身的格式都是固定的，差別僅在於顯示的內容；此外，依據該模組所配置的韌體版本，對應的參數項目與可設的參數值可能也會有所不同。

AS04AD-A

- 數值格式
- 通道 1 ~ 通道 4 模式設定
- 通道 1 ~ 通道 4 調校參數
- 平均濾波**
- 取樣時間
- 通道值則設定

平均濾波

參數名稱	設定值	單位	預設值	最小值	最大值
通道1 平均次數	10		10	1	100
通道2 平均次數	10		10	1	100
通道3 平均次數	10		10	1	100
通道4 平均次數	10		10	1	100
通道 1 平均濾波比例	10%		10%	-	-
通道 2 平均濾波比例	10%		10%	-	-
通道 3 平均濾波比例	10%		10%	-	-
通道 4 平均濾波比例	10%		10%	-	-

以下則是表格中各個欄位的相關說明，而當某個欄位的底色呈現灰階時，便代表該欄位無法設定。

➤ 參數名稱

此欄位所顯示的為各個參數的名稱或敘述。

參數名稱	設定值	單位	預設值	最小值	最大值
通道1 平均次數	10		10	1	100
通道2 平均次數	10		10	1	100
通道3 平均次數	10		10	1	100
通道4 平均次數	10		10	1	100

➤ 設定值

此欄位用於設定所選參數的初始值，後續當下載 HWCONFIG 的參數時，所設定的初始值亦會一併被寫入模組當中；而依據所選參數的不同，此欄位有可能是直接輸入或是下拉選單的格式；關於允許輸入的數值範圍與初始值下載之後的處理機制，請參考各機種的使用說明。

參數名稱	設定值	單位	預設值	最小值	最大值
通道1 平均次數	10		10	1	100
通道2 平均次數	10		10	1	100
通道3 平均次數	10		10	1	100
通道4 平均次數	10		10	1	100

3

➤ 單位、預設值、最小值、最大值

用於分別列出參數的物理量單位，參數的預設值，及參數可設定的範圍的最小值與最大值，供使用者設定參考。

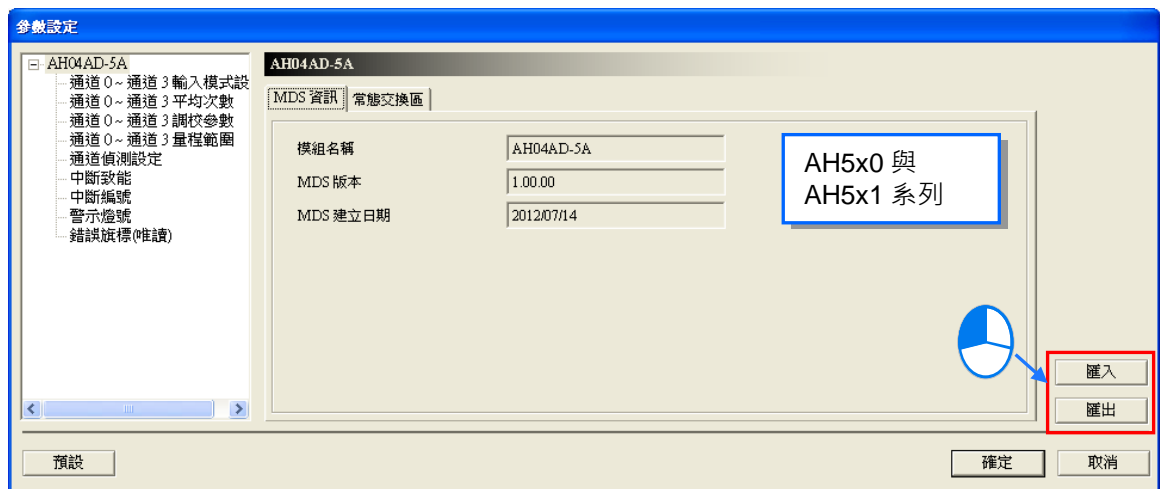
參數名稱	設定值	單位	預設值	最小值	最大值
通道1 平均次數	10		10	1	100
通道2 平均次數	10		10	1	100
通道3 平均次數	10		10	1	100
通道4 平均次數	10		10	1	100

當按下參數設定視窗左下角的「預設」按鈕後，該模組的所有參數值便會全部恢復為系統預設值。



3.7.3 模組內部參數的匯出與匯入

所有模組內部的參數值皆可匯出為 CSV 檔另行備份，同時亦可將先前所匯出的檔案重新匯入。欲匯出時，按下參數設定視窗的「匯出」按鈕，接著於檔案視窗中指定欲匯出的路徑與檔名後儲存即可。欲匯入時，按下參數設定視窗的「匯入」按鈕，接著於檔案視窗中指定欲匯入的檔案後開啟即可。匯出與匯入按鈕在各機種系列的編輯介面略有不同，如下圖。(AH560 備援系列機種目前模組內部參數的匯出與匯入可比照 AH5x1 系列。)



3

匯入模組參數時系統會先檢查檔案格式及模組型號，當檔案格式或模組型號不符時，系統將會禁止匯入的操作，並顯示警示訊息；而當原匯出檔案中所指定的 **D 對應裝置** 位址在目前的配置中已被佔用時，匯入之後該 **D 對應裝置** 的位址便會自動跳號。

3.7.4 模組內部參數的更新

若要做模組內部參數的更新，首先，需進入線上模式，若要進入線上模式請參考第 3.8 節，進入線上模式後，點選想要更新參數的模組，此時會跳出裝置設定視窗，接著點擊視窗中下方的更新按鍵，此時模組內部參數即完成更新，此功能僅限於數位輸入 I/O、類比 I/O 及網路模組。

3.7.5 智慧型模組參數設定

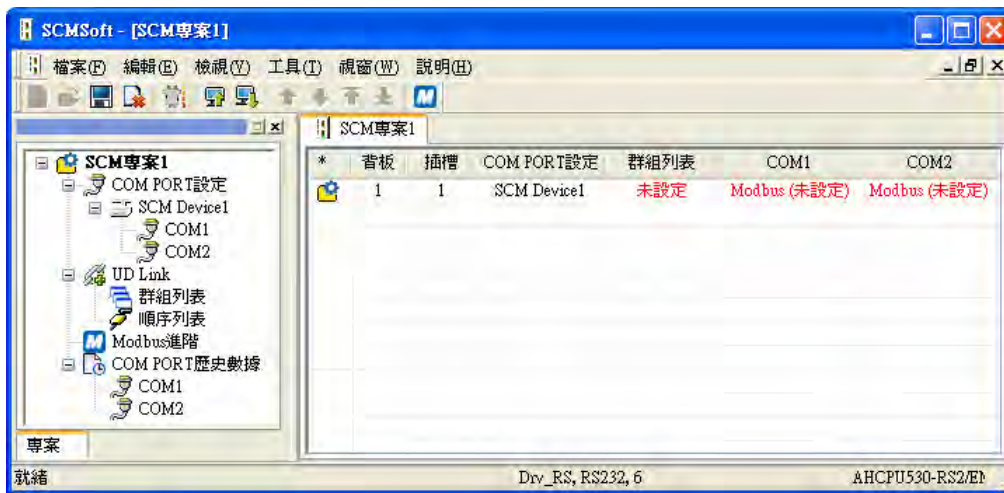
台達電子對於部分模組有另外提供專屬的組態工具，可透過 HWCONFIG 來直接開啟這些工具軟體對模組進行更進階的組態與參數設定，例如 CANopen 的網路規劃.....等等。

如下圖以 AH5x0 的系統為例，欲設定這些智慧型模組的進階參數時，請先於系統配置區中點選該模組的圖示後按下滑鼠右鍵，之後於快捷選單中點選「智慧型模組設定」，而若此時於 HWCONFIG 中所做的配置與參數設定尚未完成存檔時，系統會先要求進行存檔後才允許開啟對應的軟體；而開啟之後，在關閉智慧型模組的專屬工具軟體之前，使用者將無法在 HWCONFIG 中做任何的編輯。部份組態工具需另行安裝，未安裝時將出現訊息提示使用者安裝組態軟體。

關於開啟軟體之後的詳細操作方式請自行參考各軟體以及模組所提供的使用手冊。



3




3.8 HWCONFIG 硬體參數管理及線上診斷

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	DVPxxMC	AS5xx
HWCONFIG 列印功能	V	V	V	V	V			V
訂購清單	V	V	V	V	V			V

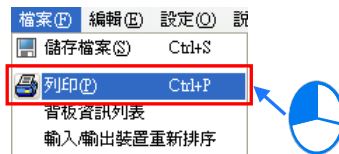
3.8.1 硬體規劃的儲存與列印

當完成所有的硬體規劃後，便可將 HWCONFIG 中所做的設定與整個專案一起儲存，而此處儲存的內容包括模組配置、模組內部參數及 PLC 主機參數，但不包括智慧型模組的進階參數；而所儲存的檔案除了 PLC 主機參數會另外以「專案名稱.para」的檔名儲存之外，模組配置與模組內部參數皆會儲存於「專案名稱.hw」的檔案中，且所有的檔案皆會與 ISPSoft 的專案檔 (*.isp) 存放於同一路徑下。

欲存檔時，請於功能工具列中點選 **檔案 (F) > 儲存檔案 (S)** 或點擊圖示工具列的  圖示即可。



而要進行列印時，請於功能工具列中點選 **檔案 (F) > 列印 (P)**，之後便會開啟列印管理工具，不過列印的內容僅會包含系統配置區的配置圖面。關於**列印管理工具**的說明請參考**附錄 C**。

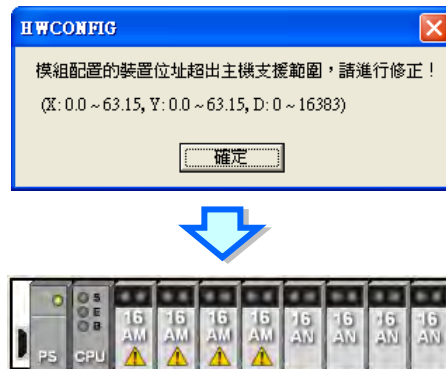


3

補充說明

設定完硬體參數後，當後續又於 ISPSOft 中變更機種類別，且若新機種無法完全支援原硬體設定中的裝置位址範圍，則重新開啟 HWCONFIG 時，畫面便會出現警示訊息。

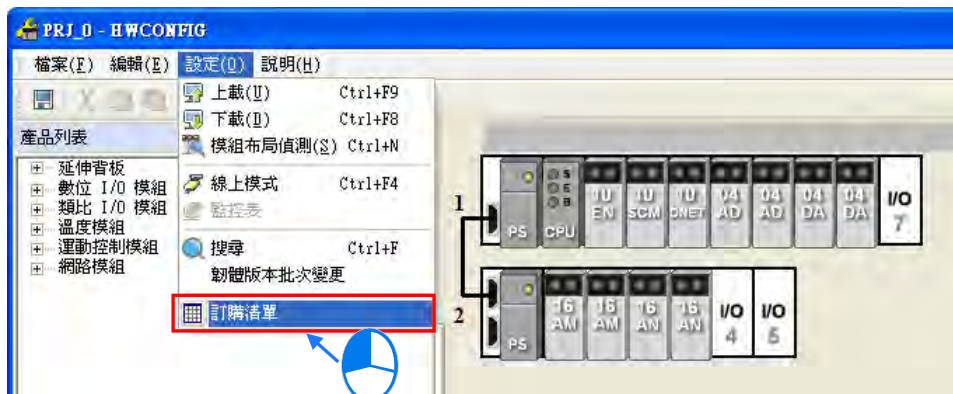
進入 HWCONFIG 之後，有問題的模組圖示將會出現三角形的錯誤標誌，代表該模組處於異常狀態；此時包括**下載**與**線上模式**的功能皆無法使用，而必須待使用者將有問題的位址修正之後，才可重新進行後續的操作；當錯誤位址被修正之後，三角形的錯誤標誌便會立即消失。



3.8.2 訂購清單

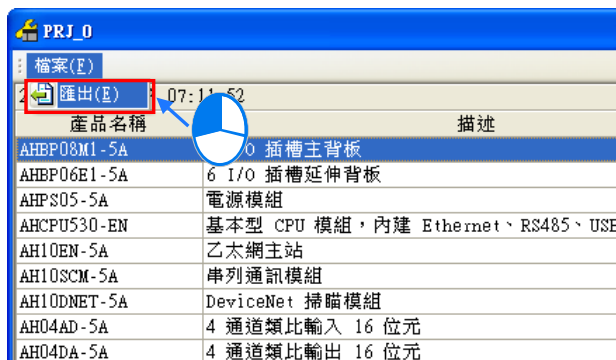
當完成所有的硬體配置後，於功能工具列中點選 **設定 (O) > 訂購清單**，之後系統便會自動統計所有配置的硬體，並將其以清單的方式呈現。

3



產品名稱	描述	數量
AHBP08M1-5A	8 I/O 插槽主背板	1
AHBP06E1-5A	6 I/O 插槽延伸背板	1
AHPS05-5A	電源模組	2
AHCPU530-EN	基本型 CPU 模組，內建 Ethernet、RS485、USB 與 SD 卡介面	1
AH10EN-5A	乙太網主站	1
AH10SCM-5A	串列通訊模組	1
AH10DNET-5A	DeviceNet 掃描模組	1
AH04AD-5A	4 通道類比輸入 16 位元	2
AH04DA-5A	4 通道類比輸出 16 位元	2
AH16AM10N-5A	16 點數位輸入，24 VDC	2
AH16AN01R-5A	16 點數位輸出，繼電器 240 VAC/24 VDC	2
延伸通訊線	延伸背板專用延伸線	1

當於清單視窗的左上角點選 **檔案 (E) > 匯出 (E)** 時，此清單便可以 CSV 檔的格式被匯出，而之後使用者便可透過 EXCEL 來對匯出的資料進行編輯或管理。



3.8.3 AH 系列機種的背板資訊列表

當完成所有的硬體配置後，於功能工具列中點選 **檔案 (F) > 背板資訊列表**，之後便可透過資訊列表來檢視所有的配置。而**背板資訊列表**的呈現則是以背板為單位，並採用階層管理的方式，使用者可自行收藏或展開各個背板的下層項目，以增加檢視時的便利性。



插槽編號	名稱	韌體版本	描述	輸入裝置範圍	輸出裝置範圍	註解
資訊: 背板 1						
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	1.00	基本型 CPU 模組，內建 Ethernet	None	None	
0	AH10EN-5A	0.40	乙太網主站	D0 ~ D19	D20 ~ D39	
1	AH10SCM-5A	0.40	串列通訊模組	D40 ~ D57		
2	AH10DNET-5A	0.40	DeviceNet 掃描模組	None	None	
3	AH04AD-5A	0.41	4 通道類比輸入 16 位元	D58 ~ D65		
4	AH04AD-5A	0.41	4 通道類比輸入 16 位元	D66 ~ D73		
5	AH04DA-5A	0.41	4 通道類比輸出 16 位元		D74 ~ D81	
6	AH04DA-5A	0.41	4 通道類比輸出 16 位元		D82 ~ D89	
7						
資訊: 背板 2						
-	AHPS05-5A	-	電源模組	None	None	
0	AH16AM10N-5A	-	16 點數位輸入, 24 VDC	X0.0 ~ X0.15		
1	AH16AM10N-5A	-	16 點數位輸入, 24 VDC	X1.0 ~ X1.15		
2	AH16AND1R-5A	-	16 點數位輸出, 繼電器 240 VAC/		Y0.0 ~ Y0.15	
3	AH16AND1R-5A	-	16 點數位輸出, 繼電器 240 VAC/		Y1.0 ~ Y1.15	
4						
5						


當於列表視窗的左上角點選 **檔案 (F) > 匯出 (E)** 時，此列表同樣可以 CSV 的格式被匯出，之後使用者便可透過 EXCEL 來對匯出的資料進行編輯或管理。



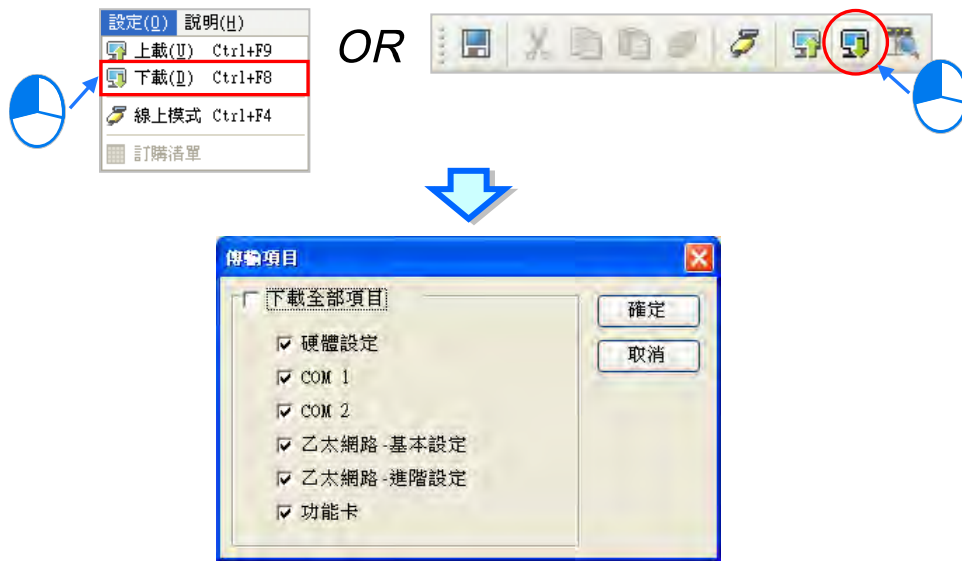
3.8.4 系統參數上下載

於 HWCONFIG 當中所做的設定，必須被下載至主機中才可正式生效，且因 HWCONFIG 會直接引用 ISPSOft 的通訊設定，因此進行參數上下載之前，請務必確認 ISPSOft 與主機正常連線。執行時，若主機目前的標籤名稱與參數設定的名稱不符時，畫面亦會出現提示訊息。

➤ 下載系統硬體參數


於功能工具列中點選 **設定 (O) > 下載 (D)**，或點選圖示工具列的  圖示，之後便會出現下載項目的確認視窗，可視需求勾選要下載的項目，但不包括智慧型模組的進階參數，按下「確定」按鈕後便可將參數下載至主機中。

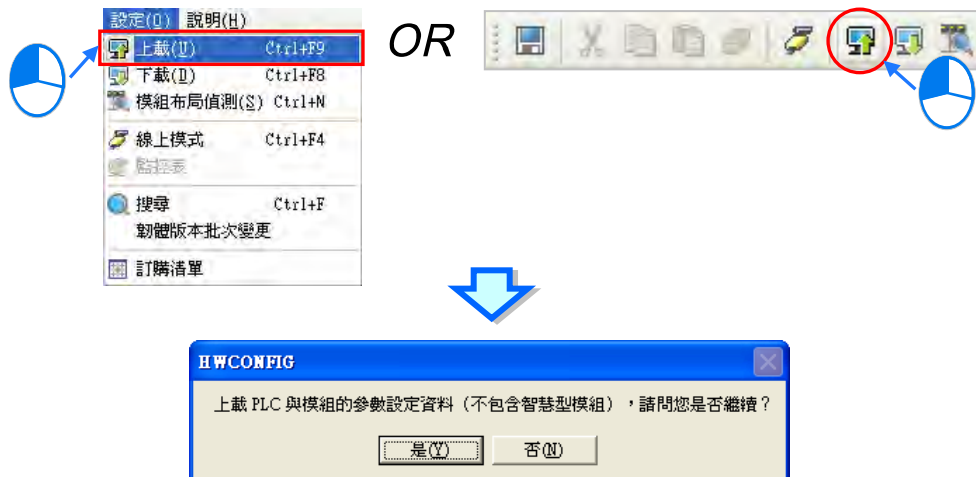
3





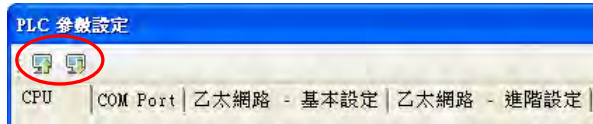
*.AH5x0 系列機種無法選擇要下載的項目，AH560 備援系列及 AH5x1 系列可以選擇要下載的項目。

➤ 上載系統硬體參數

於功能工具列中點選 **設定 (O) > 上載 (U)**，或點選圖示工具列的  圖示，之後即可將主機中的硬體組態參數上傳回 HWCONFIG 的設定畫面中，其內容包括模組配置、模組內部參數及 PLC 主機參數，但不包括智慧型模組的進階參數。

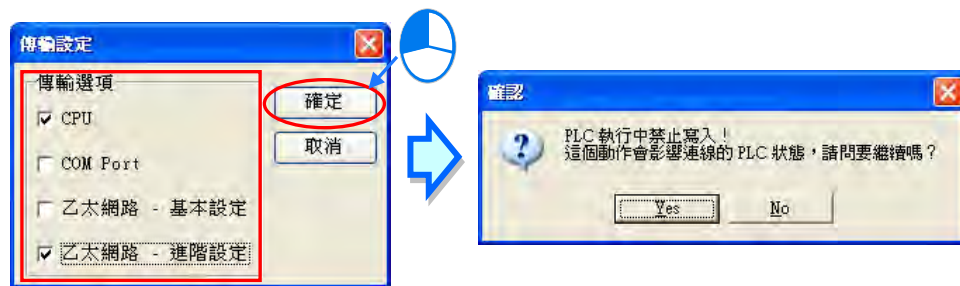


AH5x0 系列機種在 PLC 參數設定頁面另外提供了上下載功能，而此方法便僅會針對 PLC 主機參數的部份進行操作。當於 PLC 參數設定頁面的左上角點擊  圖示時便可進行下載，而點擊  圖示則可進行上傳。



執行上傳操作時，所有的主機參數將會全部被上傳回來；而執行下載操作時，畫面則會先出現勾選視窗，使用者可於視窗中勾選想要下載的參數主類別，而無須將所有的設定全部下載至主機中。

下圖即為下載項目的勾選視窗，而當下載項目包含「CPU」或「乙太網路 - 進階設定」時，下載時的主機狀態必須為 STOP，因此若主機處於 RUN 的狀態時，系統便會在開始下載前先顯示提示訊息。




*1. 當主機目前的標籤名稱與參數設定的名稱不符時，在開始執行上傳或下載之前會先顯示提示訊息。

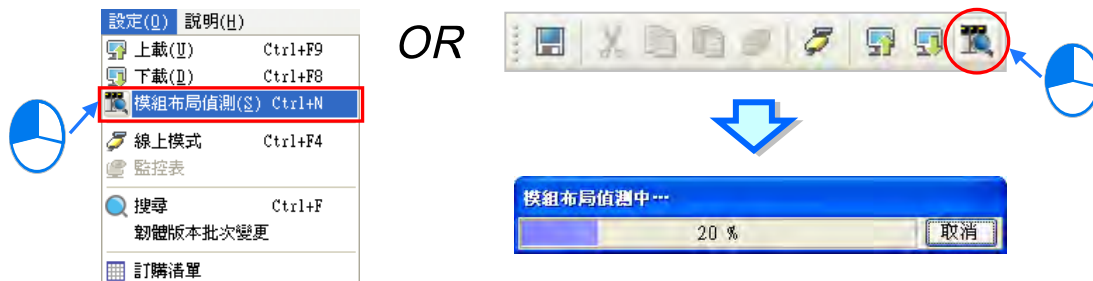
*2. HWCONFIG 會引用 ISPSOft 的通訊設定，因此進行上下載前請確認 ISPSOft 已可與主機正常連線，請參考第 2.4 節。

3.8.5 AH 系列機種的模組布局偵測

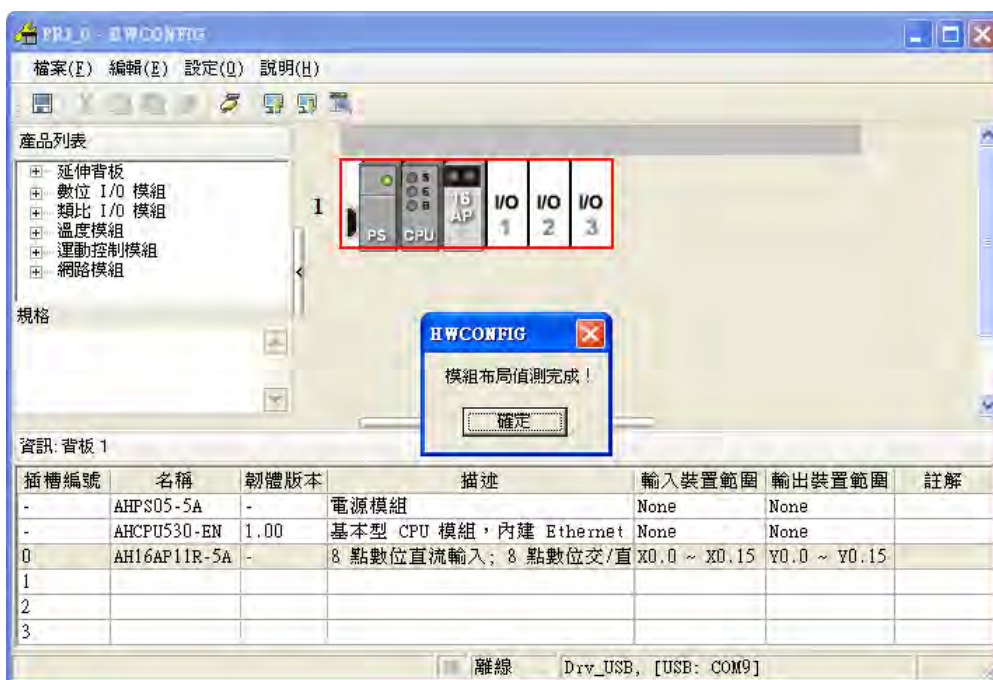
進行模組配置時，除了可透過前述的方式進行操作之外，當使用者已取得相關的硬體設備時，操作上便可先將硬體設備架設起來，之後再利用 HWCONFIG 的「模組布局偵測」功能，並透過通訊連線的方式將實際的模組配置狀態掃描上來，如此便可省去逐一配置模組的時間；而進行操作之前，請確認 ISPSOft 已可與主機正常連線，詳細的設定與操作方式請參考第 2.4 節當中的內容。

操作時請於 HWCONFIG 的功能工具列中點選 **設定 (O) > 模組布局偵測 (S)**，或點選圖示工具列中的  圖示即可；而偵測的內容則包括模組型號、背版與插槽配置、模組韌體版本，但並不會包含模組的參數設定與 CPU 主機參數；且在執行**模組布局偵測**之後，原本的模組配置與參數設定都將會完全清除，而 CPU 本身的參數設定則不會受到影響。

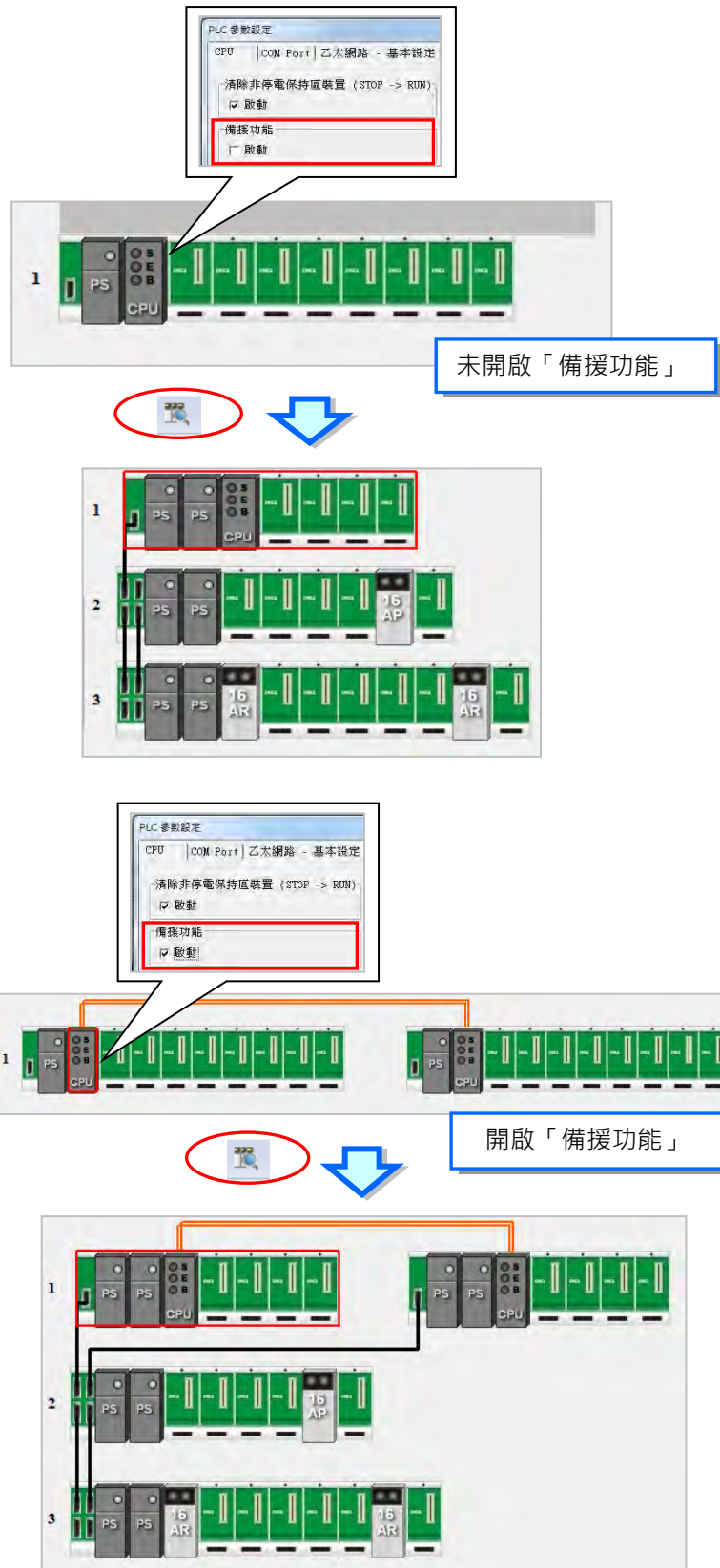
3



完成偵測之後，系統便會將實際背版與插槽上的模組配置狀態顯示於**系統配置區**，且會重新配置所有模組的**輸入/輸出裝置範圍**，而所有模組的內部參數也都會設為系統預設值。



若是使用 AH560備援系列機種，在開啟備援功能後（備援功能的開啟，請參考3.3.2.2節），做「模組布局偵測」會與未開啟備援功能不同，可參考下兩例圖說明（此為範例說明，實際偵測結果，請考量現場連線狀況）。



AH5x0、AH5x1系列機種及 AH560備援系列機種配置的韌體版本則會依據實際偵測到的模組韌體版本；但若偵測到的韌體版本較系統擁有的 MDS 文件還要高時，該欄位便會顯示可選的最高版本編號，但會於**註解**欄標註該模組實際的韌體版本。AHxxEMC 系列機種配置的韌體版本則會依據實際偵測到的模組韌體版本；但若偵測到的韌體版本較系統擁有的 DDF 版本不符時，該欄位便會顯示可選的最高版本編號，但會於**註解**欄標註該模組實際的韌體版本

*.關於 MDS 文件與模組韌體版本的相關說明請參考第 3.4.1 節當中的內容。

接著我們將以一個實例來說明「**模組布局偵測**」與「**上載**」之間的差異。其中「**模組布局偵測**」只會偵測模組於背板上的配置狀況，但不會得知各模組的參數設定；而「**上載**」則會將存放於主機內部的系統硬體參數重新讀取回來，但讀取回來的硬體配置資料則可能已與實際的背板配置狀況不符。

首先將下圖的硬體參數下載至主機中，而其中模組 AH16AN01R-5A (插槽1) 的內部參數如下。

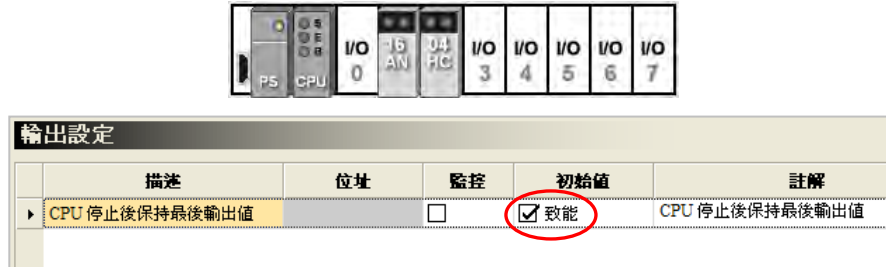


接著我們將置於插槽2的 AH04HC-5A 從實際的背板上移除，此時主機的 BUS FAULT 燈號便會亮起，代表主機內部的硬體參數已與實際的硬體配置不符。

接下來我們再執行「**模組布局偵測**」的功能，而偵測後的結果則會與實際的硬體配置狀況相符，此時再開啟模組 AH16AN01R-5A (插槽1) 的參數視窗，我們會發現模組內的參數皆是系統預設值，而與先前下載至主機的參數不同，這代表「**模組布局偵測**」只是單純偵測模組於背板上的配置狀況。



接著我們再執行「上載」的功能，而完成之後，我們會發現上傳回來的硬體配置與模組參數，其內容與之前下載至主機時的設定無異，這也代表「上載」的功能僅會忠實的將存放於主機內部的系統硬體參數讀取回來，而與實際的硬體狀態無關。





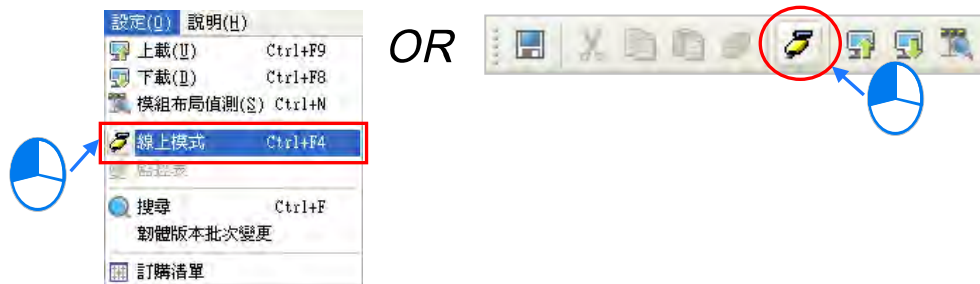
接著我們再次執行「模組布局偵測」的功能，並將偵測的結果下載至主機中，此時主機的 BUS FAULT 燈號便會熄滅，這代表主機內部的硬體參數已與實際的硬體配置一致。

3.8.6 AH 系列機種的線上診斷

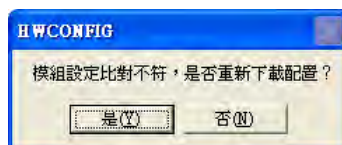
HWCONFIG 除了提供離線的裝置組態之外，更可在與 PLC 連線的狀況下，利用線上操作的功能，來進行簡易的系統操作與檢修。而因 HWCONFIG 會直接引用 ISPSOft 的通訊設定，因此進行線上操作之前，請務必確認 ISPSOft 與主機正常連線。此外，當要執行時，若發現主機目前的標籤名稱與參數設定的標籤名稱不符時，畫面亦會出現提示訊息。

3.8.6.1 啟動線上操作模式

- (1) 於功能工具列中點選 設定 (O) > 線上模式，或按下圖示工具列中的  圖示，之後便可進入 HWCONFIG 的線上模式；而要離開線上模式時，只要再次將  圖示重新按起即可。

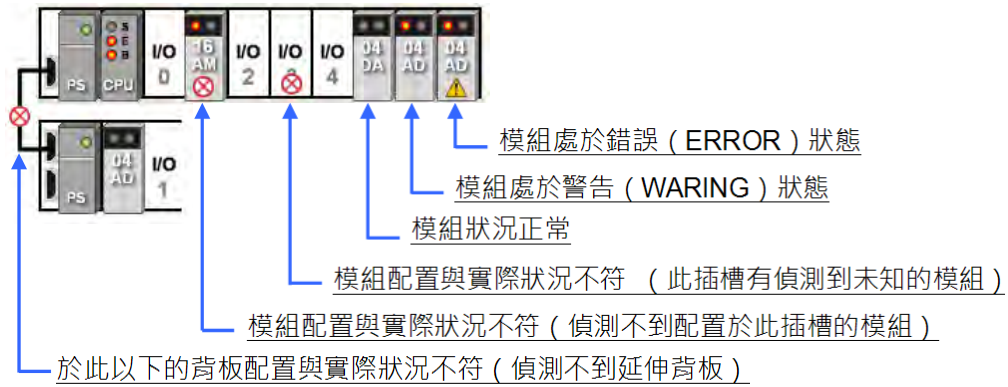


- (2) 在進入線上模式之前，系統會先檢查存放於主機內部的硬體配置參數，並確認與系統配置區的模組設定是否一致，若兩者不一致時，系統便會要求重新下載配置參數。

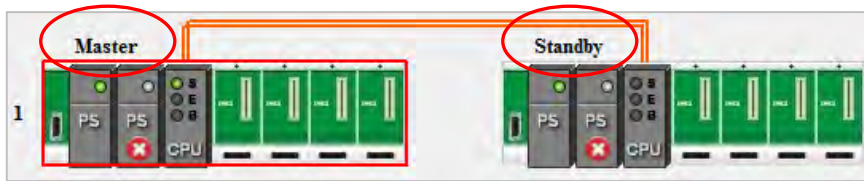


- (3) 進入線上模式後，畫面中的圖示便會根據實際的模組狀態而有不同的顯示；其中 CPU 圖示上的燈號將與主機實際的燈號狀態一致，圖示的燈號由上至下分別是 Run、Error 與 Bus Fault。

3



若是使用 AH560 備援系列機種，在開啟「備援功能」，並進入線上模式後，還會顯示 Master 和 Standby 狀態。



3.8.6.2 資訊檢視與模組診斷

於線上模式點選 CPU 或模組圖示後按下滑鼠右鍵，接著於快捷選單中點選「模組資訊」，之後便可透過連線取得 CPU 或模組的相關資訊，並顯示於隨之開啟的視窗當中。



若在模組圖示按下滑鼠右鍵，並於選單中點選「診斷」項目，則畫面將會出現該模組的錯誤記錄視窗。



在錯誤記錄視窗中，「**當前錯誤記錄**」欄位會顯示目前的錯誤狀態，而「**歷史錯誤記錄**」欄位則會列出既往的錯誤履歷，但不同模組可存放的記錄筆數皆不相同，若該模組不會儲存歷史記錄時，「**歷史錯誤記錄**」欄位便會維持空白。此外，數位 IO 模組沒有錯誤記錄，因此無法使用「**診斷**」功能。

當錯誤狀況已排除時，按下視窗中的「**清除**」鍵便會清除畫面上與模組中儲存的錯誤記錄，並可解除模組的錯誤狀態；按下「**更新**」鍵時，系統便會重新擷取資料並將其更新於視窗中。

3.8.6.3 線上變更模組狀態

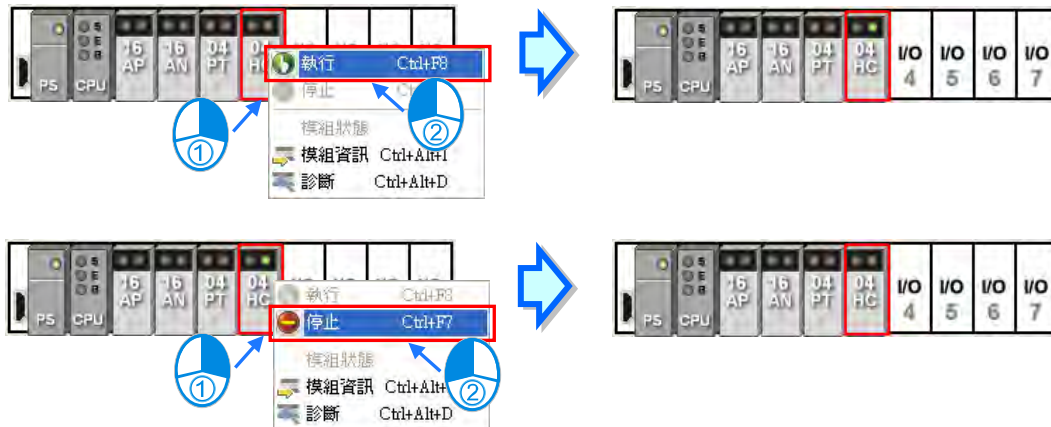
HWCONFIG 支援於線上模式時變更裝置本身的狀態，使用者可藉此進行簡單的測試。

⚠ 變更裝置狀態之前，請務必確認所做的操作不會造成系統或人員的傷害。

➤ 變更裝置的運轉狀態

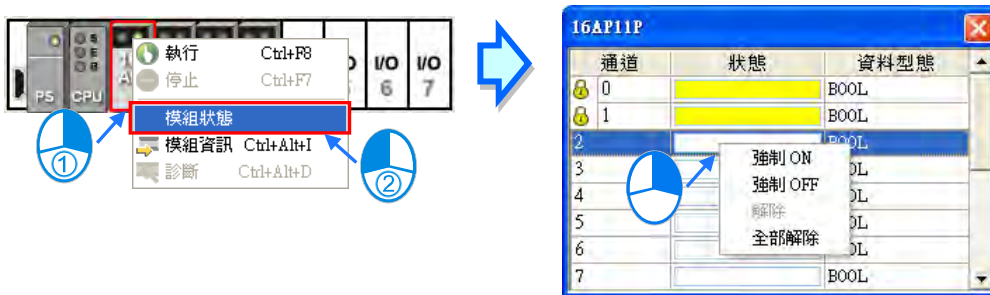
於線上模式點選 CPU 或模組圖示後按下滑鼠右鍵，接著於快捷選單中點選「執行」項目，之後所選的裝置便會進入執行的狀態，且當模組開始執行時，其圖示亦會出現綠色的運轉顯示燈號；而同樣的，於快捷選單中點選「停止」項目後，所選的裝置便會恢復至停止狀態。此外，當變更 CPU 的運轉狀態時，所有模組的狀態也會一併變更。

3



➤ 變更模組的輸入/輸出狀態

當硬體配置中包含數位 IO、類比 IO 或溫度模組時，於這些模組的右鍵快捷選單中點選「模組狀態」項目，接著便可於狀態視窗中監控這些 IO 模組的輸入或輸出狀態，其中數位 IO 模組還可透過右鍵選單來對輸入或輸出通道進行強制 ON/OFF 的操作。不過當執行此功能時，CPU 與該模組則必須處於執行 (RUN) 的狀態才會實際產生輸出。



- **強制 ON**：將點選的通道狀態強制設定為 ON。
- **強制 OFF**：將點選的通道狀態強制設定為 OFF。
- **解除**：解除點選通道的強制狀態。
- **全部解除**：解除全部通道的強制狀態。

➤ 變更模組參數值

當模組有設定 **D 對應裝置** 時，使用者便可在進入線上模式後直接變更這些參數的設定值。不過依據模組的設計，執行此功能時，CPU 與模組可能須處於執行 (RUN) 的狀態下才會實際產生作用。

- (1) 在線上模式下，於欲變更參數值的模組圖示上雙擊滑鼠左鍵，之後畫面便會開啟參數視窗，接著請於視窗左側點選欲設定的參數類別，而此時也將發現參數表格的呈現與離線時有很大的差異。



- (2) 表格中的「現在值」欄位代表目前模組的參數值，但不會保持即時更新，而當按下參數視窗右側的 **更新** 鍵時，系統才會重新擷取並將數值填入「現在值」的欄位中。



- (3) 表格中的「設定值」欄位則用以輸入欲變更的數值，且只有預先設定 **D 對應裝置** 的參數才可進行輸入，其他參數的「設定值」欄位則會呈現灰階。要變更參數值時，請直接於「設定值」欄位輸入欲變更的數值，並在完成所有設定值的輸入後，按下視窗右側的 **SV->PV** 圖示便可將所有的參數值寫入 **D 對應裝置** 中。



*變更參數設定值前請查閱各模組的使用說明，以確認允許輸入的範圍。

3.8.6.4 監控表

當 HWCONFIG 處於線上模式時，於功能工具列中點選 **設定 (O)** > **監控表**，之後便可於監控表中檢視 **D** 對應裝置的狀態值；但監控表中的項目無法自行添加，而必須於模組參數設定中，預先勾選該參數的監控功能之後，在開啟的監控表中才會自動帶出對應的項目，相關說明請參考第 3.7.2 節。

3

描述	位址	監控	初始值
通道 0 校正偏移量 (攝氏/華氏)	D3008-D3009 ...	<input checked="" type="checkbox"/>	0.000000
通道 1 校正偏移量 (攝氏/華氏)	D3010-D3011 ...	<input checked="" type="checkbox"/>	0.000000
通道 2 校正偏移量 (攝氏/華氏)	D3012-D3013 ...	<input checked="" type="checkbox"/>	0.000000
通道 3 校正偏移量 (攝氏/華氏)	D3014-D3015 ...	<input checked="" type="checkbox"/>	0.000000
通道 0 校正斜率	D3016-D3017 ...	<input type="checkbox"/>	1.000000
通道 1 校正斜率	D3018-D3019 ...	<input type="checkbox"/>	1.000000
通道 2 校正斜率	...	<input type="checkbox"/>	1.000000
通道 3 校正斜率	...	<input type="checkbox"/>	1.000000

背板編號	插槽編號	模組名稱	裝置名稱	現在值	數值型態	註解
1	2	AH04PT-5A	D3008	0	浮點數	通道 0 校正偏移量 (攝氏/華氏)
1	2	AH04PT-5A	D3010	0	浮點數	通道 1 校正偏移量 (攝氏/華氏)
1	2	AH04PT-5A	D3012	0	浮點數	通道 2 校正偏移量 (攝氏/華氏)
1	2	AH04PT-5A	D3014	0	浮點數	通道 3 校正偏移量 (攝氏/華氏)

*.若只變更模組參數設定視窗中的「監控」欄位，則設定完成後該參數設定無須下載至主機中即可生效。

監控表中的項目僅供參數值的檢視之用，使用者將無法於此變更各個參數值，而若要變更 **D** 對應裝置的內容值時，請參考第 3.7.1 節的說明來進行操作；此外，用滑鼠點擊「數值型態」欄位後便可於下拉選單中選擇監控數值的顯示格式。

背板編號	插槽編號	模組名稱	裝置名稱	現在值	數值型態	註解
1	2	AH04PT-5A	D3000	0	十進位	通道 0 輸入模式設定
1	2	AH04PT-5A	D3001	0	十進位	通道 1 輸入模式設定
1	2	AH04PT-5A	D3002	0	十進位	通道 2 輸入模式設定
1	2	AH04PT-5A	D3003	0	十進位	通道 3 輸入模式設定
1	2	AH04PT-5A	D3004	20	十進位	通道 0 平均次數
1	2	AH04PT-5A	D3005	20	十進位	通道 1 平均次數
1	2	AH04PT-5A	D3006	40	十進位	通道 2 平均次數
1	2	AH04PT-5A	D3007	40	十進位	通道 3 平均次數
1	2	AH04PT-5A	D3008	0	浮點數	通道 0 校正偏移量
1	2	AH04PT-5A	D3010	0	浮點數	通道 1 校正偏移量
1	2	AH04PT-5A	D3012	0	浮點數	通道 2 校正偏移量
1	2	AH04PT-5A	D3014	0	浮點數	通道 3 校正偏移量
1	2	AH04PT-5A	D3016	1	浮點數	通道 0 校正斜率
1	2	AH04PT-5A	D3018	1	浮點數	通道 1 校正斜率

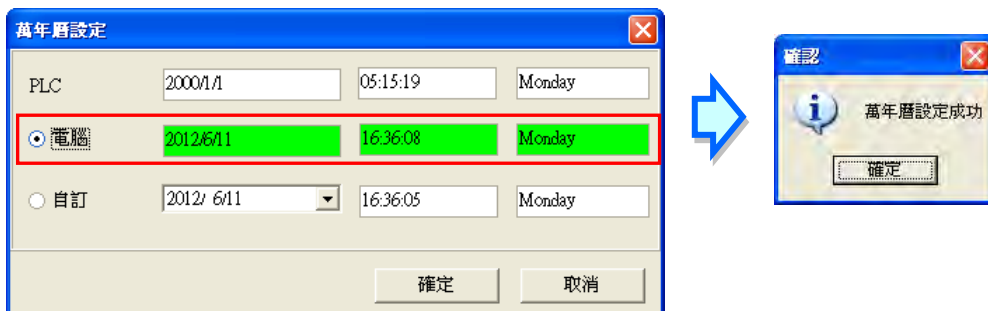
3.9 萬年曆設定

PLC 主機中的萬年曆資訊可透過 ISPSOft 所提供的工具來進行設定，而設定之前，同樣必須先確認 ISPSOft 已可與主機正常連線，詳細的設定與操作方式請參考第 2.4 節當中的內容。

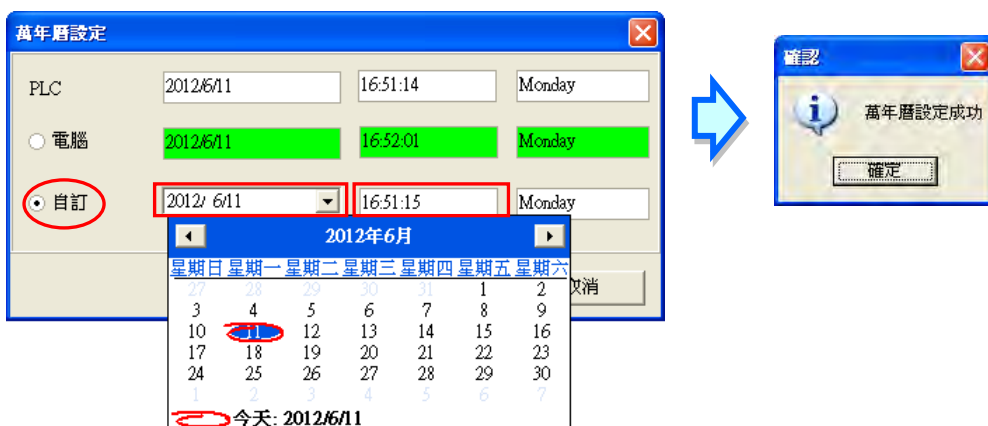
於功能工具列中點選 **工具 (I) > 萬年曆設定 (I)**，之後系統便會出現萬年曆的設定視窗，其中最上方的日期時間為視窗開啟時由 PLC 內部所擷取出來的資訊。



當要使用電腦目前的時間做為萬年曆的設定值時，選擇「電腦」選項後按下「確定」鍵即可。



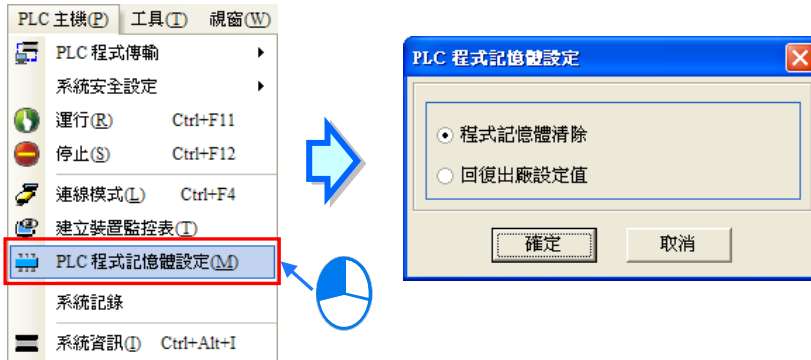
若要自訂萬年曆的時間，則請先選擇「自訂」選項後，分別於日期與時間欄位中輸入欲設定的值，而其中點擊日期欄位右側的按鈕便會出現輔助工具；完成設定後按下「確定」鍵即可。



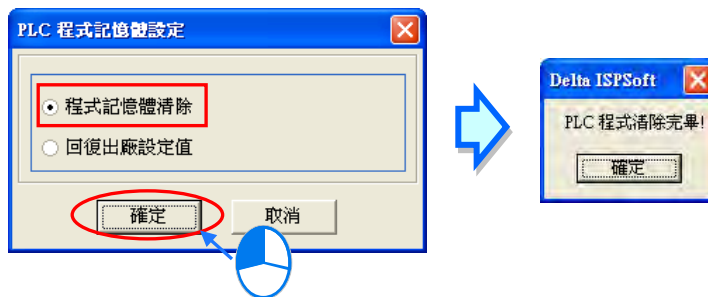
3.10 PLC 記憶體設定 (清除程式&回復出廠值)

透過 ISPSOft，使用者可將 PLC 主機中的程式記憶體清除，或是直接將 PLC 恢復為出廠預設值，而同樣的，操作前請先確認 ISPSOft 已可與主機正常連線，詳細的設定與操作方式請參考第 2.4 節。

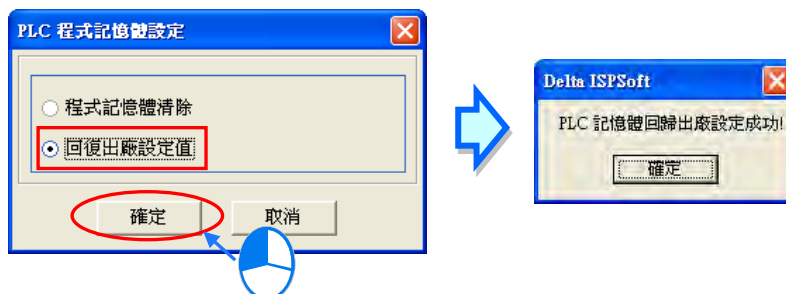
於功能工具列中點選 **PLC 主機 (P)** > **PLC 程式記憶體設定 (M)**，之後便會出現操作視窗。



選擇「程式記憶體清除」後按下「確定」，系統便會將主機內部的程式記憶體資料全部清除。



若選擇「回復出廠設定值」，則主機內部的所有資料與狀態都會在確認執行之後被恢復為出廠設定值。而依據機種的不同，完成後可能必須將 PLC 重新上電才可繼續之後的操作。



若連接的是 AH 系列機種，則必須先將 PLC 主機參數中的「遠端重置」功能啟動並完成參數的下載，之後才允許透過 ISPSOft 將主機恢復為出廠設定值。詳細說明請參考第 3.3 節與第 3.4 節當中的內容。

4

第4章 快速入門

目錄

4.1	快速入門	4-2
4.1.1	範例功能說明	4-2
4.1.2	硬體規劃.....	4-2
4.1.3	程式規劃.....	4-3
4.2	ISPSoft 的專案開發程序	4-3
4.3	新增專案	4-4
4.4	硬體組態	4-5
4.4.1	模組配置.....	4-6
4.4.2	主機與模組參數設定.....	4-8
4.5	建立程式	4-10
4.5.1	新增階梯圖程式	4-10
4.5.2	基本編輯 - 建立接點與線圈	4-12
4.5.3	基本編輯 - 階梯圖區段的新增/插入與指令編輯模式	4-15
4.5.4	基本編輯 - 階梯圖區段的選取與操作	4-17
4.5.5	基本編輯 - 並联接點裝置	4-20
4.5.6	基本編輯 - 編輯註解.....	4-21
4.5.7	基本編輯 - 插入 API 應用指令	4-22
4.5.8	基本編輯 - 比較接點的建立與常數的輸入格式	4-24
4.5.9	完成範例程式	4-25
4.5.10	程式的檢查與編譯.....	4-26
4.6	測試與除錯.....	4-27
4.6.1	建立連線.....	4-27
4.6.2	下載程式與組態參數.....	4-30
4.6.3	連線測試.....	4-32

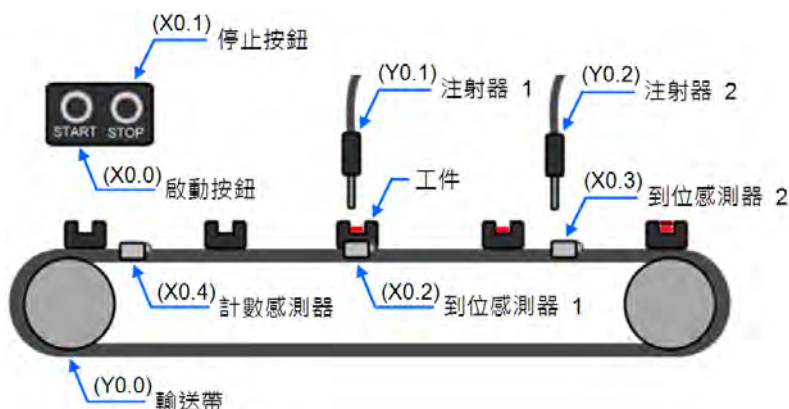
4.1 快速入門

本章將以一個簡單的範例，帶領使用者於 ISPSOft 的開發環境中，快速建立一個傳統的階梯圖程式；不過在本章中尚不會導入 IEC61131-3 的相關編程觀念，如 POU、功能塊 (FB)、變數符號...等，目的是為了讓尚不熟悉 IEC61131-3 的使用者同樣能快速地了解 ISPSOft 所提供的功能，並得以了解如何建立傳統的階梯圖程式。至於 IEC61131-3 的相關觀念亦將於後續的章節當中陸續介紹。

4.1.1 範例功能說明

設備開始運轉時，工件會在輸送帶上持續地由左至右傳送，當感測器檢知到工件位於注射器的下方時，PLC 便會送出一觸發信號給注射裝置，此時注射器便會開始注射膠體，而注射時間將會由外部進行設定，在 PLC 的程式中則無須加以控制，但設計上必須能讓觸發信號恢復至 OFF 狀態，以便進行下次的觸發。輸送帶上共有兩站的點膠動作，且兩站的程序完全相同。

另外，在輸送帶的左側會安置一感測器，當工件通過時，計數值便會加 1，當計數值到達 100 時，內部的完成旗標便會被設置為 ON，而該旗標的狀態則可於後續供其他程序運用，本範例將暫不提及。



4.1.2 硬體規劃

在本範例中，我們將選用 AH5x0 系列主機 AHCPU530-EN，數位 IO 模組則選用 8 點輸入/8 點輸出的 AH16AP11R-5A，而主背板則選用 4 槽的 AHBP04M1-5A。下表即為本範例的 IO 配置表。

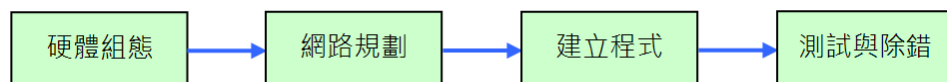
Type	ID	說明
DI	X0.0	啟動按鈕
DI	X0.1	停止按鈕
DI	X0.2	到位感測器 1
DI	X0.3	到位感測器 2
DI	X0.4	計數感測器
DO	Y0.0	輸送帶
DO	Y0.1	注射器 1 觸發信號
DO	Y0.2	注射器 2 觸發信號

4.1.3 程式規劃

- (1) 當**啟動按鈕** (X0.0) 由 OFF 變為 ON 時，將內部的運轉旗標設為 ON，並讓**輸送帶** (Y0.0) 開始持續運轉；而當**停止按鈕** (X0.1) 由 OFF 變為 ON 或偵測到錯誤發生 (錯誤旗標為 ON) 時，則將運轉旗標重置為 OFF，且讓所有的動作停止。
- (2) 當**到位感測器 1** (X0.2) 為 ON 時，便將**注射器 1 觸發信號** (Y0.1) 也設置為 ON；而當感測器變為 OFF 時，便將該信號重置回 OFF。
- (3) 當**到位感測器 2** (X0.3) 為 ON 時，便將**注射器 2 觸發信號** (Y0.2) 也設置為 ON；而當感測器變為 OFF 時，便將該信號重置回 OFF。
- (4) 每當**計數感測器** (X0.4) 由 OFF 變為 ON 時，便將內部的計數值加 1；而當計數值大於或等於 100 時，則將內部的完成旗標設置為 ON。

4.2 ISPSoft 的專案開發程序

下圖為一般在 ISPSoft 中進行專案開發時的建議程序，不過使用者仍可依據實際的應用情形與本身的使用習慣加以調整，相關說明如下。



● 硬體組態

設定 PLC 主機的相關參數，如停電保持區、通訊埠站號...等，而 AH500 系列的機種還須針對所搭載的模組進行組態配置與參數設定等工作。

● 網路規劃

當所應用的系統有運用網路架構或是裝置之間的資料交換時，透過 ISPSoft 本身所提供的網路規劃工具 - **NWCONFIG**，使用者便可輕易進行網路部署、PLC Link、Ether Link...等網路組態的規劃。

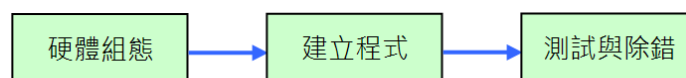
● 建立程式

於 ISPSoft 的程式編輯器上撰寫控制程序，並在撰寫完成之後進行編譯的工作；而當編譯產生錯誤時，利用**編譯訊息區**的引導功能，使用者便可快速移動至產生錯誤的位置以進程式碼的確認。


● 測試與除錯

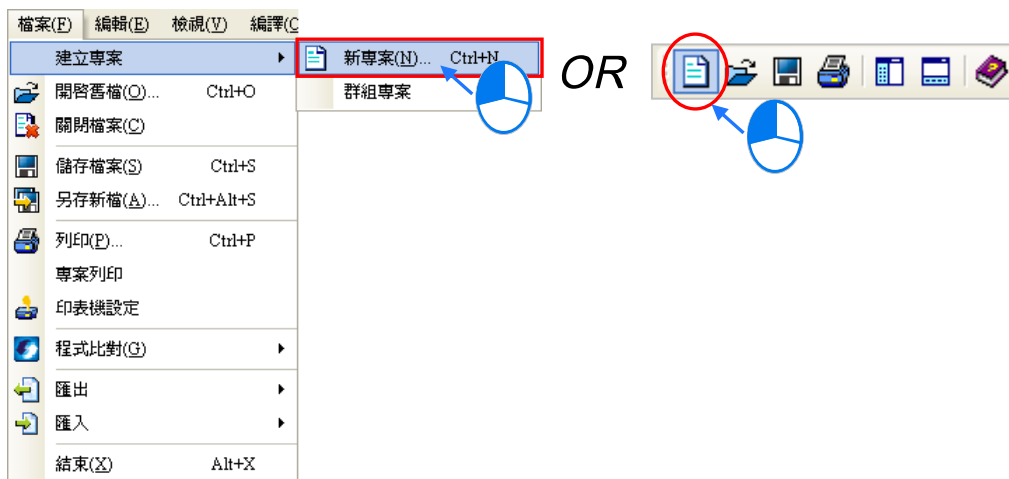
將編譯完成的程式、硬體與網路的組態參數下載至 PLC，並利用 ISPSoft 所提供的各種線上監控功能來進行測試與除錯的工作。

針對本章所介紹的範例，因不包含網路架構的部份，因此僅須進行如下的工作，而各工作的建構方式與操作步驟，我們亦將於後續的幾個小節當中逐一說明。




4.3 新增專案

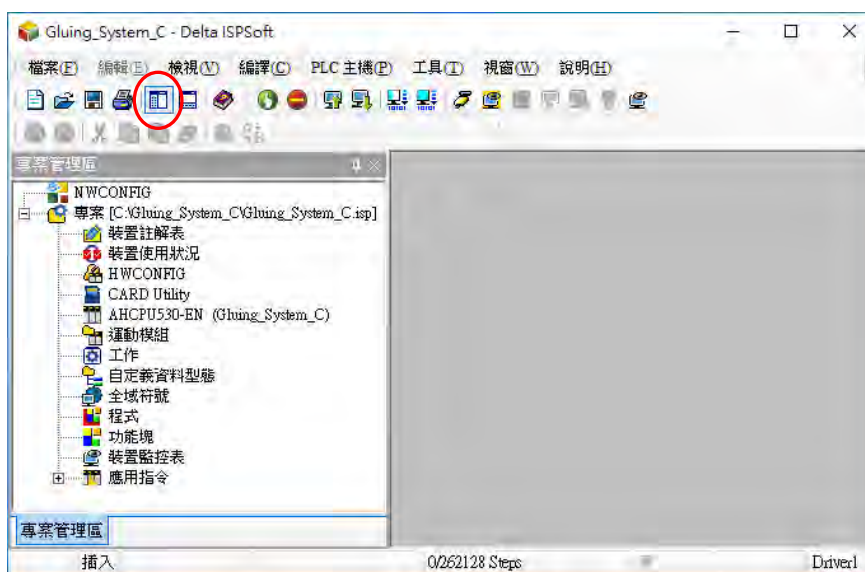
啟動 ISPSOft 後，於工具列中點選 **檔案 (F)** > **建立專案** > **新專案 (N)** 或直接點擊  圖示。



接著於專案建立視窗中依序輸入「專案名稱」及「檔案路徑」，並於「控制器種類」與「機種」的下拉選單中選擇正確的機種型式，完成後按下「確定」。(本範例中的機種型式為 AHCPU530-EN)



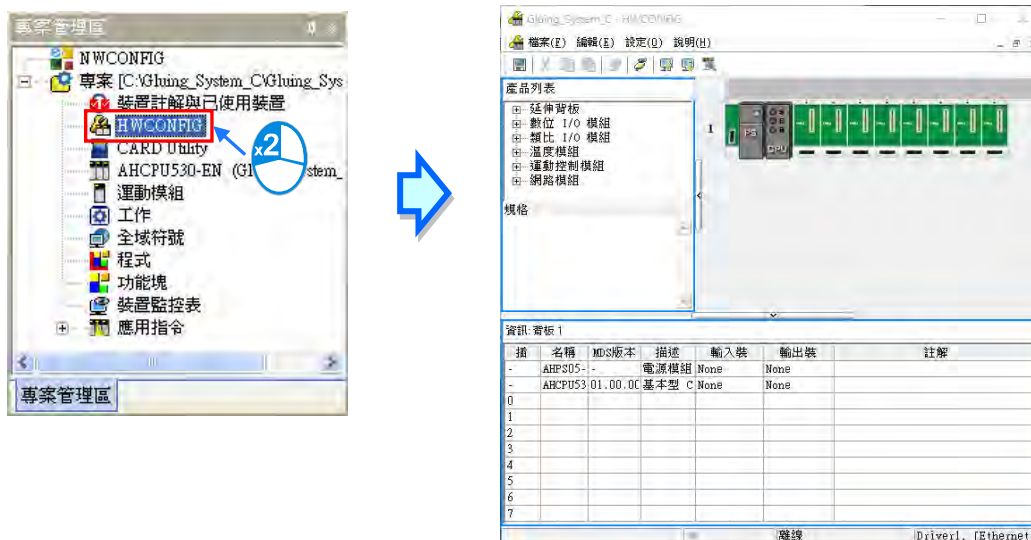
成功建立專案後，畫面的左側便會開啟一個「專案管理區」，並以階層樹狀圖的方式列出所有物件；若未出現該區域時，請點擊功能工具列中的 **檢視 (V)** > **工作區 (W)** 或點擊  圖示即可。



4

4.4 硬體組態

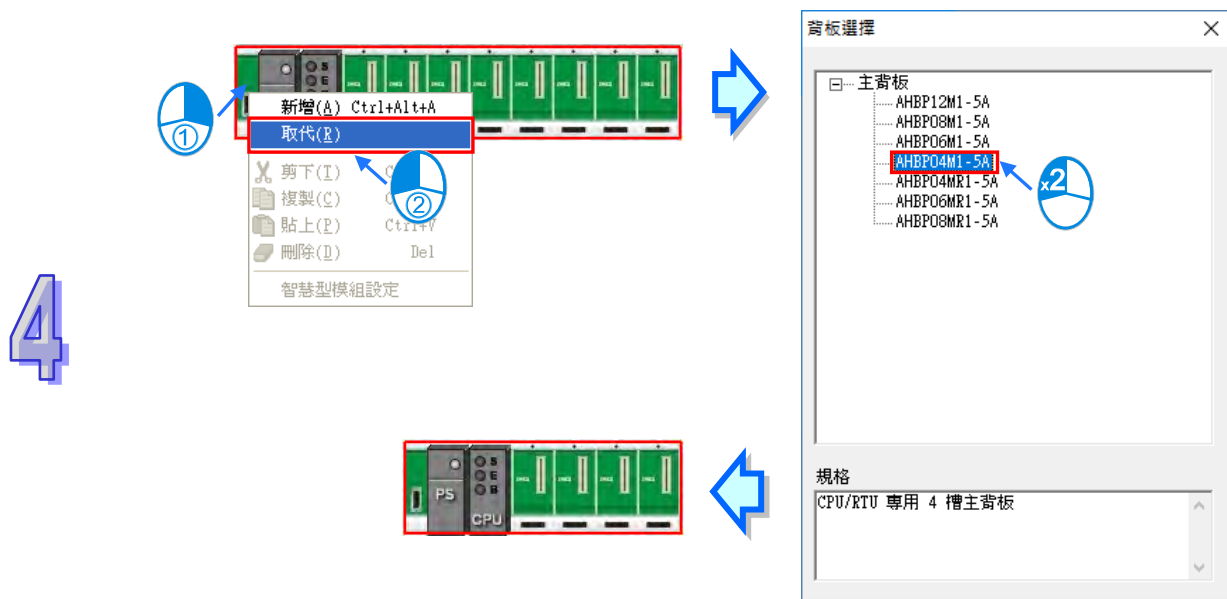
於專案管理區中的「HWCONFIG」項目上雙擊滑鼠左鍵即可啟動硬體規劃工具。



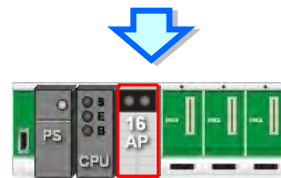
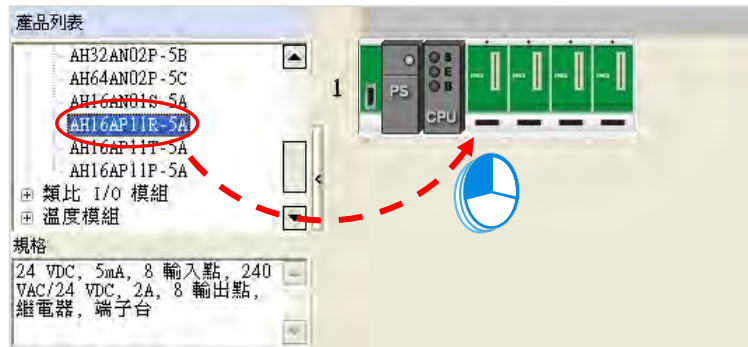
4.4.1 模組配置

初次進入 HWCONFIG 的操作畫面時，預設的配置只有搭載 CPU 與電源模組的一個 8 槽背板。但在本範例的規劃中，背板應為 4 槽的 **AHBP04M1-5A**，並須搭配 **AH16AP11R-5A** 的數位 IO 模組。

欲變更背板時，請先於系統配置區的背板圖示左側點擊滑鼠右鍵，並於快捷選單中點選「取代 (R)」，之後再於背板選擇視窗中的「**AHBP04M1-5A**」項目上雙擊滑鼠左鍵，如此便可完成背板的變更。



接著請展開左側「產品列表」的「數位 IO 模組」，並於該目錄下找到 **AH16AP11R-5A** 的項目，之後直接將該項目拖曳至系統配置區的背板插槽上即可。而成功新增一模組後，於下方表格的對應位置中便會出現該模組的項目，並列出相關的資訊與位址配置。



4

資訊: 背板 1

插槽編號	名稱	MDS版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	01.00.00	基本型 CPU 模組，內建 Ethernet、RS485、	None	None	
0	AH16AP11R-5A	-	8 點數位直流輸入；8 點數位交/直流輸出	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1						
2						
3						

HWCONFIG 會自動為每個加入的模組配置對應的裝置位址，而當配置的位址與規劃不符時，於表格中對應該模組的位址欄位上點擊滑鼠左鍵，之後再點擊欄位旁的 **...** 鍵即可自行指定位址。

資訊: 背板 1

插槽編號	名稱	MDS版本	描述	輸入裝置範圍	輸出裝置範圍	註解
-	AHPS05-5A	-	電源模組	None	None	
-	AHCPU530-EN	01.00.00	基本型 CPU 模組，內建 Ethernet、RS485、	None	None	
0	AH16AP11R-5A	-	8 點數位直流輸入；8 點數位交/直流輸出	X0.0 ~ X0.15	Y0.0 ~ Y0.15	
1						
2						
3						



手動配置

輸入裝置範圍

裝置名稱 X

裝置編號 0
(X0 ~ X511)

長度 1

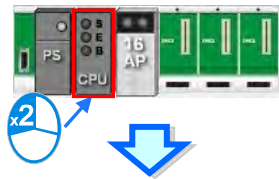
確定 取消

4.4.2 主機與模組參數設定

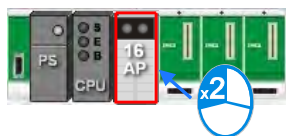
完成模組配置後，接著便可進一步針對 PLC 主機與擴充模組進行各別的參數設定，操作時僅須於欲設定參數的主機或模組上雙擊滑鼠左鍵即可開啟對應的設定視窗。

首先於 CPU 的圖示上雙擊滑鼠左鍵，之後即可開啟主機的參數設定視窗。於設定視窗中，我們可以透過上方的主類別標籤及下方的副類別標籤來切換各個設定頁，並可在各個頁面中設定 PLC 主機的相關參數；而在本範例中，我們僅須重新定義 PLC 主機的標籤名稱即可。

PLC 標籤名稱的參數位於「CPU」主類別的「名稱」副類別頁面中，且通常在第一次開啟 PLC 主機的參數設定視窗時便會進入該頁面；進入該頁面後，我們可發現在「名稱」的欄位中已存在一個與專案名稱相同的預設名稱，接著請直接將其修改為「Main_Controller」後，按下「確定」鍵離開即可。

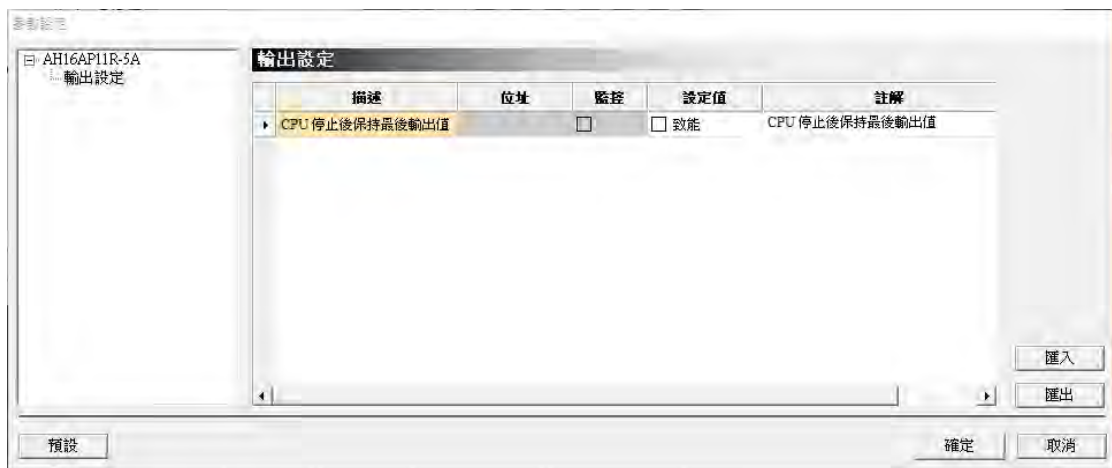



接下來請於配置的 IO 模組圖示上雙擊滑鼠左鍵，之後即可開啟設定視窗。



進入模組的參數設定視窗後，主畫面中會顯示該模組的相關資訊，而要開始設定參數時，請先於視窗左側的列表中點擊欲設定的參數類別，之後即可於參數表格中設定相關的參數。

在本範例中，我們只須延用預設值而無須做任何變更，所以此步驟可直接略過。



完成主機及模組的參數設定後，第一階段的硬體規劃工作便已完成，不過後續仍須將所做的配置與設定下載至主機後才可生效；在此我們可先將其進行儲存，而待之後下載專案程式時再一併下載即可。欲進行儲存時，請直接於功能工具列中點擊 **檔案 (F) > 儲存檔案 (S)**，又或者於圖示工具列中點擊  圖示即可。完成之後便可關閉 **HWCONFIG** 的視窗。



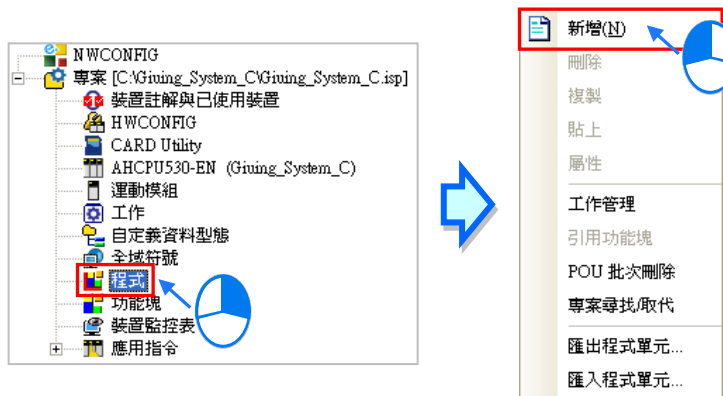
*更詳細的 **HWCONFIG** 操作方式，請參考第 3 章的相關說明。

4.5 建立程式

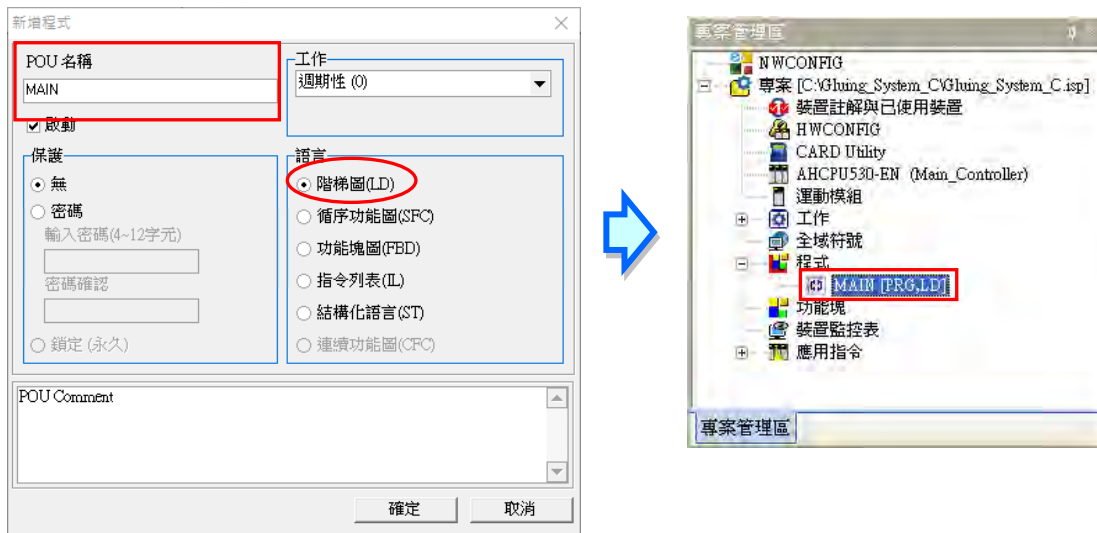
在接下來的內容中，我們將逐步示範如何在 ISPSOft 當中完成一個傳統的階梯圖程式，內容包括建立程式物件、階梯圖的基本編輯、程式編譯...等，並期待可讓使用者快速上手且具備基本的操作能力。

4.5.1 新增階梯圖程式

- (1) 於專案管理區的「程式」項目上點擊滑鼠右鍵，接著於選單中依次點選「新增 (N)」。



- (2) 於「POU 名稱」欄位中輸入程式的名稱，並在「語言」欄位中選擇階梯圖 (LD)，而其他的設定則請保持與下圖相同的預設值即可；設定完成之後，請按下「確定」鍵，此時於專案管理區的「程式」項目下便會多出一個物件，而該物件在 ISPSOft 當中便稱之為程式組織單元 - POU。



(3) 完成 POU 的新增後，在 ISPSOft 的編輯工作區中便會自動開啟一個程式編輯視窗。




而進入 LD 的編輯環境後，於 ISPSOft 的視窗中亦會出現對應的編程圖示工具列，其功能簡介如下。

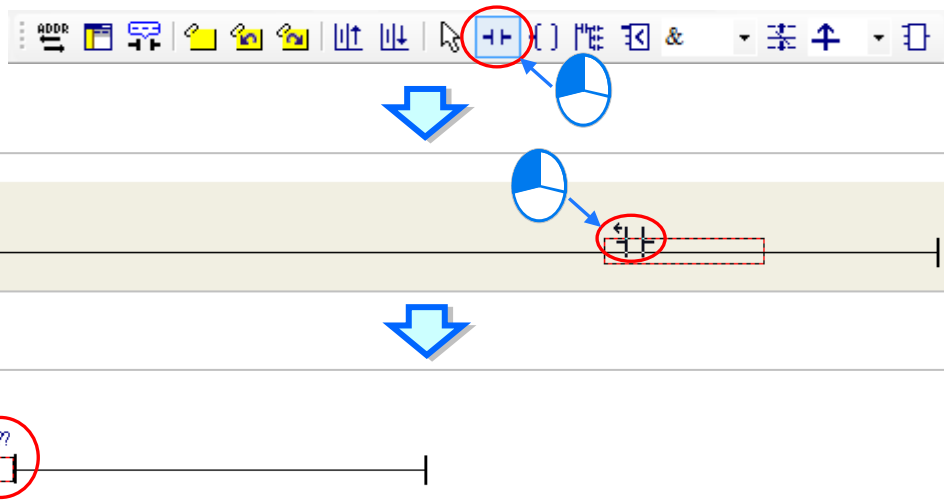



圖示	鍵盤 (快捷鍵)	功能說明
	無	切換為位址模式
	Shift + Ctrl + C	顯示/隱藏區段註解
	無	顯示/隱藏裝置註解與提示
	Shift + Ctrl + B	在目前選取的階梯圖區段上加入或取消書籤標記
	Shift + Ctrl + P	移至前一個書籤的標記位置
	Shift + Ctrl + N	移至下一個書籤的標記位置
	Ctrl + I	於目前選取的階梯圖區段之前插入一個新區段
	Shift + Ctrl + I	於目前選取的階梯圖區段之後新增一個新區段
	ESC	切換為選取工具
	指令編輯模式	插入接點裝置
	指令編輯模式	插入線圈裝置
	指令編輯模式	插入比較接點
	指令編輯模式	選擇比較接點的類型
	指令編輯模式	插入區塊邏輯指令 (NP/PN/INV/FB_NP/FB_PN)
	指令編輯模式	選擇區塊邏輯指令的類型 (NP/PN/INV/FB_NP/FB_PN)
	Shift + Ctrl + U	插入指令或功能塊

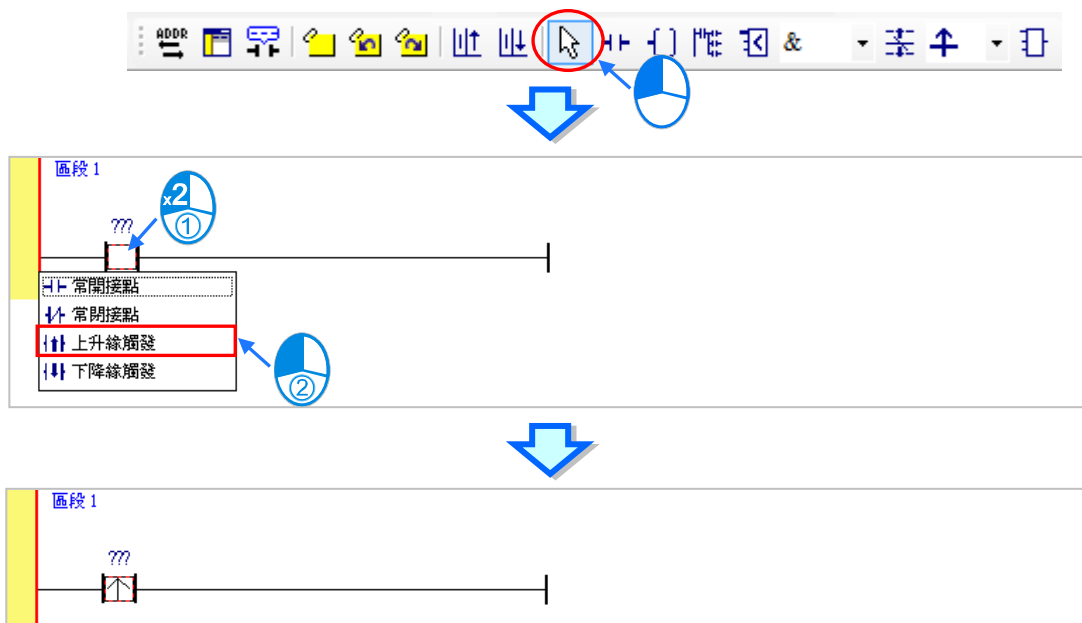
*.關於指令編輯模式請參考第 4.5.3 節的相關說明。


4.5.2 基本編輯 - 建立接點與線圈

- (1) 於圖示工具列中點選  圖示以切換為**接點工具**，接著將滑鼠移至階梯圖區段的紅色框線處，而此時滑鼠游標便會變為接點的圖示；當滑鼠往紅色框線的左右或下方偏移時，鼠標的圖示也會隨之變化，依此便可決定**接點**的插入方式；而編輯階梯圖程式時，必須將滑鼠靠近編輯位置才可進行操作，且插入的元件亦會由系統自動排列，使用者將無法任意搬移元件的位置。
- 此處尚無須考慮接點的插入方式，因此請直接將鼠標靠近紅色框線處按下滑鼠左鍵即可。

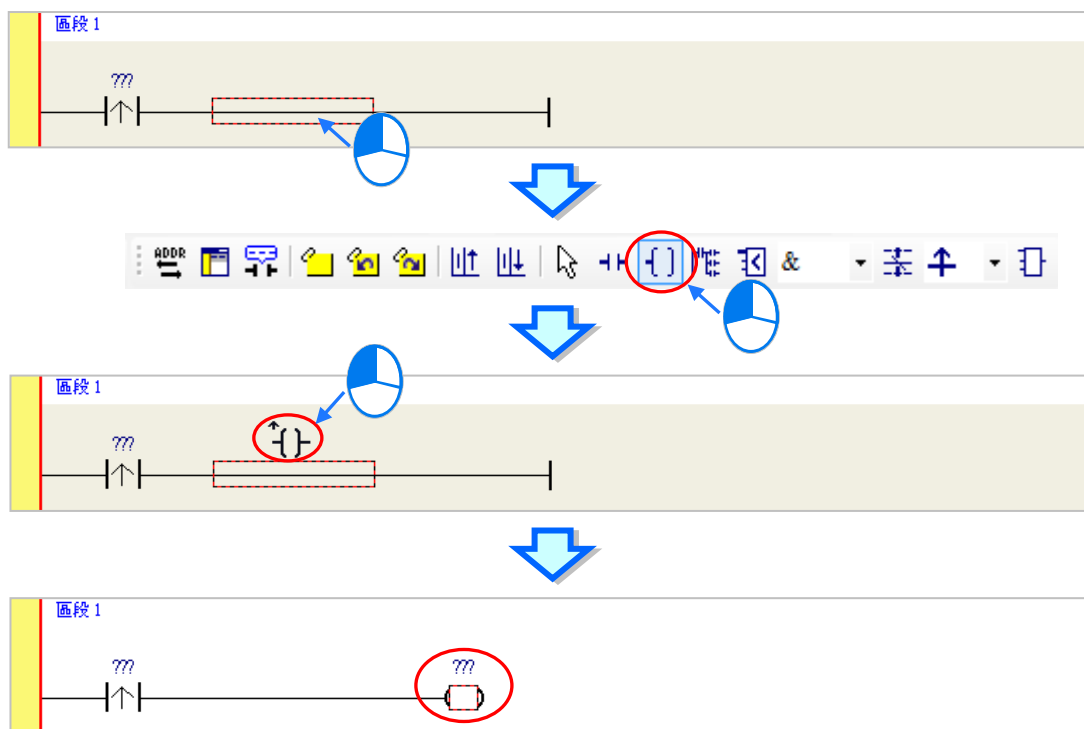



- (2) 接著請點選  圖示或按下鍵盤的【ESC】鍵將滑鼠游標切回**選取狀態**，並將滑鼠移至接點的圖示上雙擊滑鼠左鍵，此時便會出現接點型式的下拉選單，其中包括「常開接點」、「常閉接點」、「上升緣觸發」及「下降緣觸發」。在本範例中，此接點請選擇「上升緣觸發」的型式。

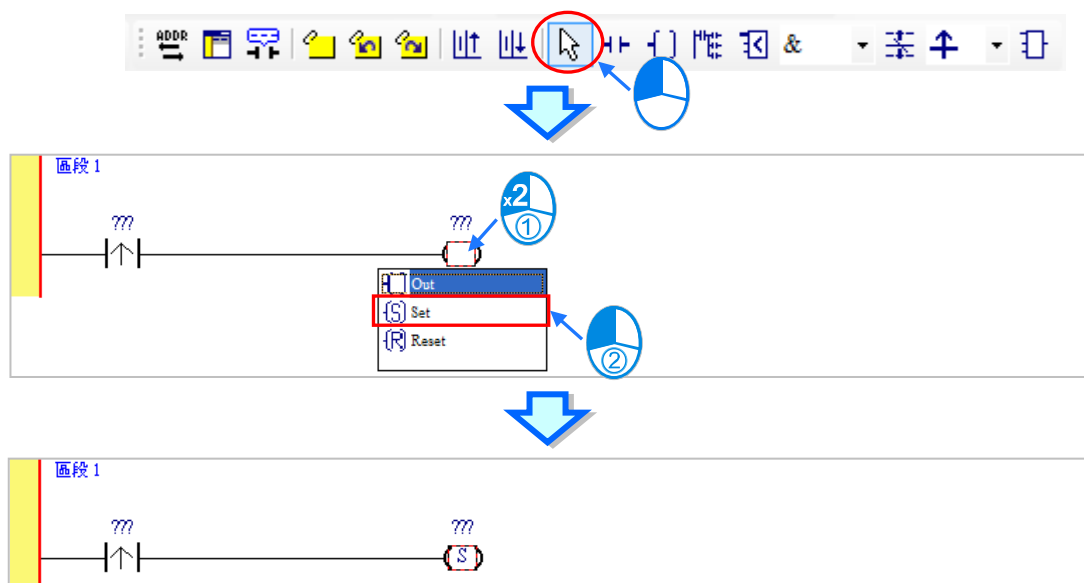


- (3) 接下來請將滑鼠移至右方的線路上點擊一下左鍵以改變編輯位置，之後再於圖示工具列中點選  圖示以切換為線圈工具，接著再將滑鼠移至紅色框線處；同樣的，當滑鼠往紅色框線的上方或下方偏移時，鼠標的圖示也會跟著變動，依此便可決定線圈的插入方式。

此處同樣無須考慮線圈的插入方式，因此請將鼠標靠近紅色框線處並按下滑鼠左鍵即可。

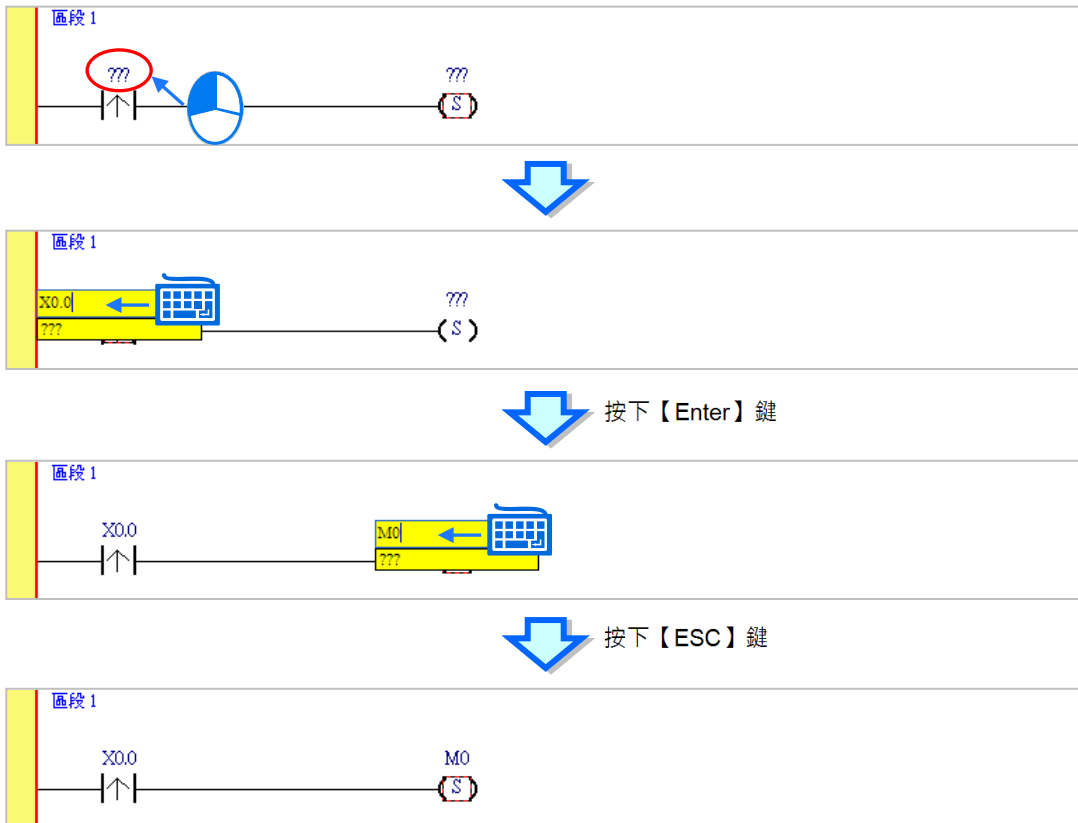


- (4) 接著請點選  圖示或按下鍵盤的【ESC】鍵將鼠標切回選取狀態，並將滑鼠移至線圈的圖示上雙擊滑鼠左鍵，此時便會出現線圈型式的下拉選單，其中包括「Out」、「Set」及「Reset」。在本範例中，此線圈請選擇「Set」的型式。



- (5) 最後請於接點或線圈的圖示上方點擊 ??? 欄位，此時該欄位便會進入編輯狀態，而之後便可在該欄位中輸入欲配置給此元件的裝置位址；輸入完成後，按下鍵盤的【Enter】鍵即可自動跳至該區段中的下一個編輯欄位，而完成所有欄位的編輯後按下【Esc】鍵即可結束編輯。
- 在本範例中，請於接點處輸入 X0.0，而線圈的部份請輸入 M0。

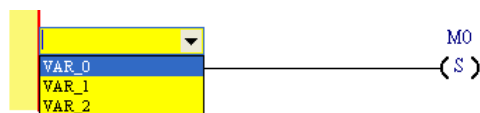
4





補充說明

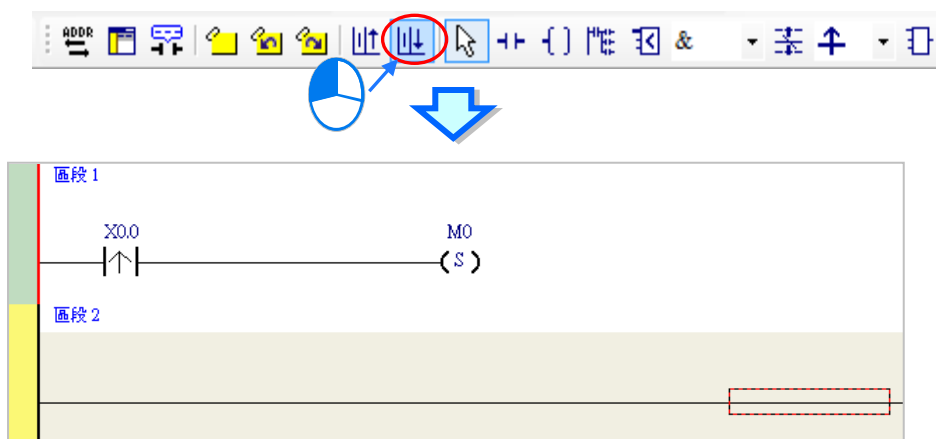
當用滑鼠點擊某個區段後按下【Enter】鍵，接著系統便會自動開啟編輯欄位，利用鍵盤的【Enter】鍵便可在同一個區段中反覆移動編輯欄位，而利用【Tab】鍵則可跨區段移動，但到了最後一個編輯欄位之後便會停止；編輯時，使用者可直接於編輯欄位中使用鍵盤輸入，而完成編輯後，按下【Enter】鍵便會自動再切換至該區段中的下一個編輯位置；當要結束編輯時，只要按下【Esc】鍵即可。

若使用者有宣告變數符號，則在裝置位址欄位呈現編輯狀態時，先清除裝置輸入欄位後便會自動出現符合該欄位型態變數符號的下拉選單供使用者挑選，此時直接使用滑鼠點選或利用鍵盤的上下鍵便可進行選擇。關於變數符號的相關說明將於第 6 章的內容當中有詳盡的介紹，此處將不再多加敘述。



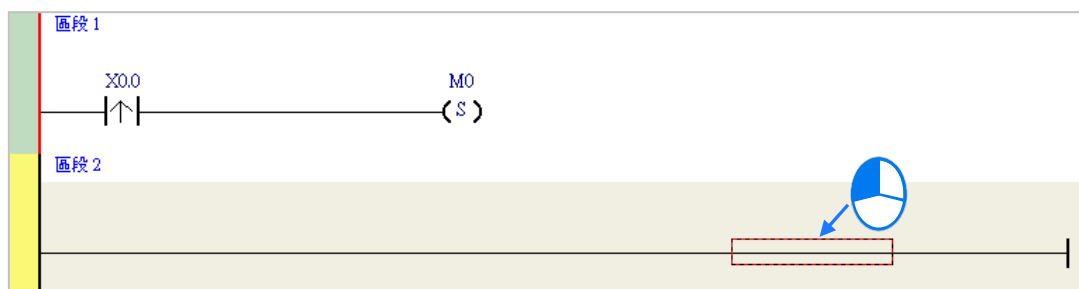
4.5.3 基本編輯 - 階梯圖區段的新增/插入與指令編輯模式

當於圖示工具列中點選  圖示便可於目前選取的區段下方新增一個空白區段；而點選  圖示則會於目前選取的區段上方插入一個空白區段。此處我們先在區段 1 的下方新增一個空白區段。

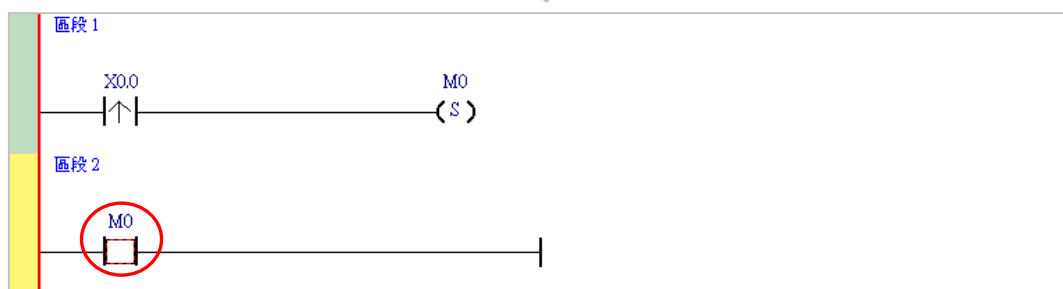
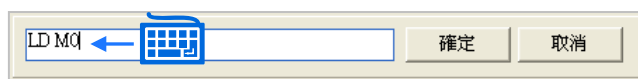


建立接點與線圈的方式除了利用前一節所介紹的圖示工具之外，尚可利用指令編輯模式來達成。

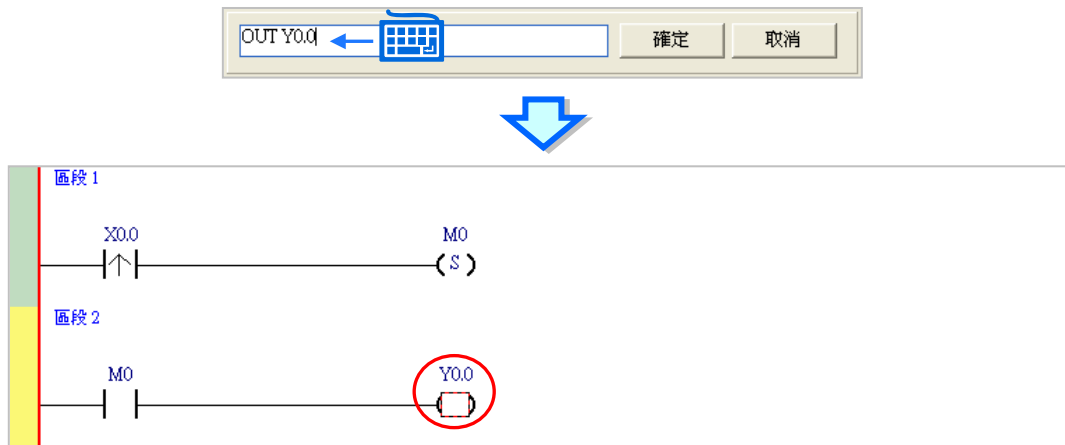
- (1) 首先請用滑鼠點擊區段 2 的線路以決定編輯位置。



- (2) 接著請直接用鍵盤輸入 IL 指令（無須分大小寫）- "LD M0"，且一按下鍵盤之後，畫面便會自動出現編輯視窗，待輸入完畢後按下鍵盤的【Enter】鍵或以滑鼠點擊視窗的「確定」鈕即可。



(3) 接著請再輸入 IL 指令 - ” **OUT Y0.0** ”，並完成如下的程式。



4

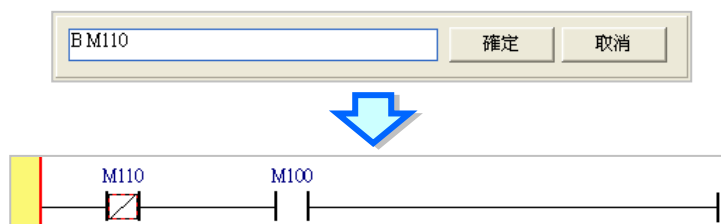
補充說明

針對接點與線圈，系統亦提供簡易輸入的功能，請參考下述。(輸入的指令無須分大小寫)

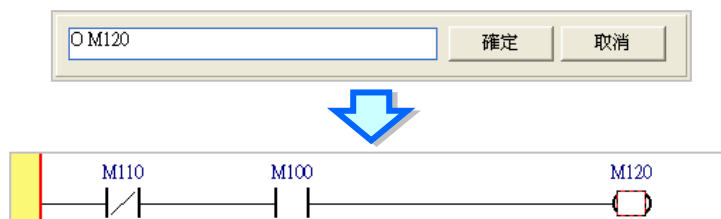
- 插入常開接點 (A 接點) 指令 : ” A 裝置位址 ”



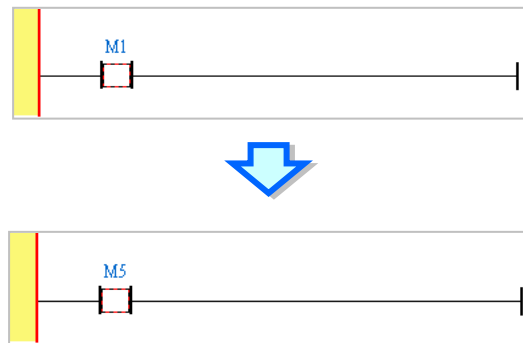
- 插入常閉接點 (B 接點) 指令 : ” B 裝置位址 ”



- 插入輸出線圈 (OUT) 指令 : ” O 裝置位址 ”

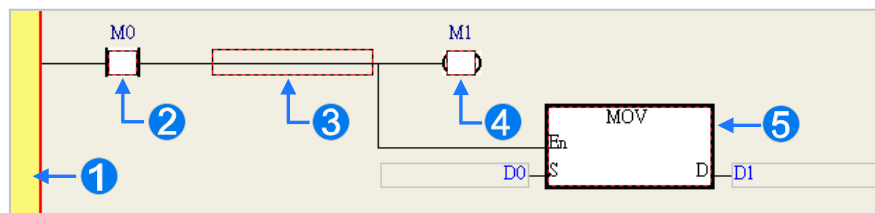


當加入裝置項目時，切換鍵盤的【Insert】鍵便可選擇使用插入或取代模式，切換鍵盤的【Insert】鍵並在 M1 位置輸入 LD M5，原本的 M1 就會被 M5 取代，如下圖。



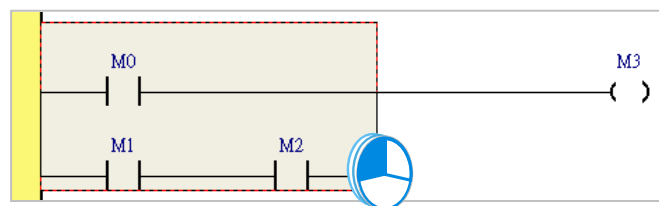
4.5.4 基本編輯 - 階梯圖區段的選取與操作

要於階梯圖區段中進行選取時，請先按下鍵盤的【ESC】鍵，或於圖示工具列中點擊  圖示，待鼠標切換為選取狀態後，直接於欲選取的物件上點擊滑鼠左鍵即可。下列即為基本的選取操作。

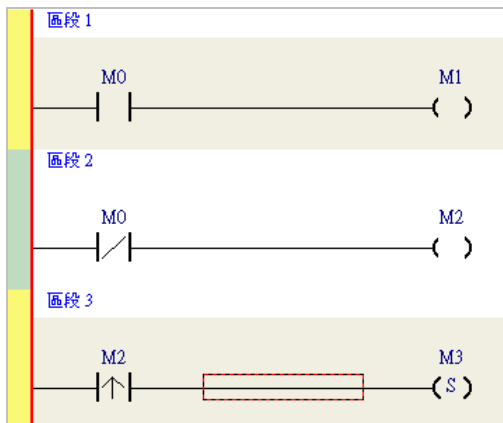


- ❶ 選取整個區段。
- ❷ 選取輸入接點。
- ❸ 選取整個區段。
- ❹ 選取輸出線圈。
- ❺ 選取方塊。

要選取區塊時，可先點選任一裝置後按住滑鼠左鍵，接著拖曳滑鼠便可將區塊框選起來；或是先點選區塊範圍的第一個物件後，按下鍵盤的【Ctrl】+【B】，接著再點選區塊範圍的最後一個物件，並再次按下【Ctrl】+【B】鍵便可將其框選；而框選時，該區塊必須位於同一個階梯圖區段中，且必須是連續位置的物件，而階梯圖區段中的輸入裝置與輸出裝置則無法被框選為同一個區塊。



當要同時選取多個階梯圖區段時，請先按住鍵盤的【Ctrl】鍵，接著便可用滑鼠分別點選多個不連續位置的區段；另外也可先按住【Shift】鍵後，再分別用滑鼠點選欲選取範圍的第一個與最後一個階梯圖區段，之後便可將整個範圍內的區段全部選取起來。



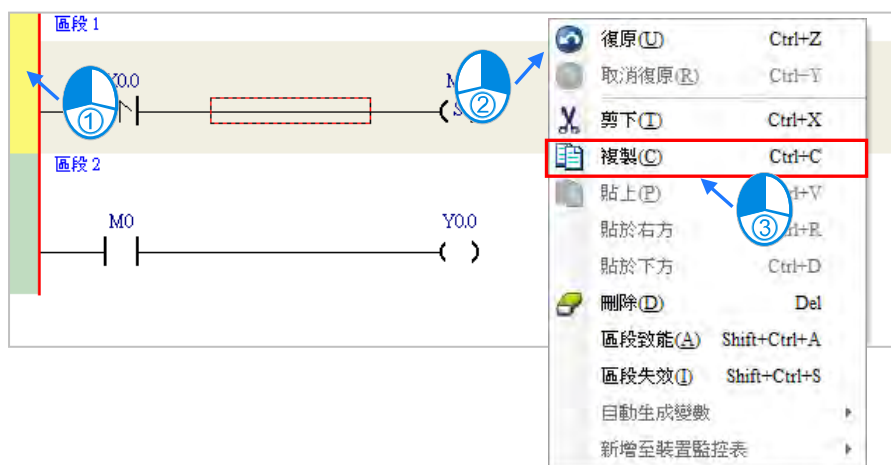
4

當選取某個物件後按下滑鼠右鍵，之後便可利用快捷選單執行一些基本的操作。

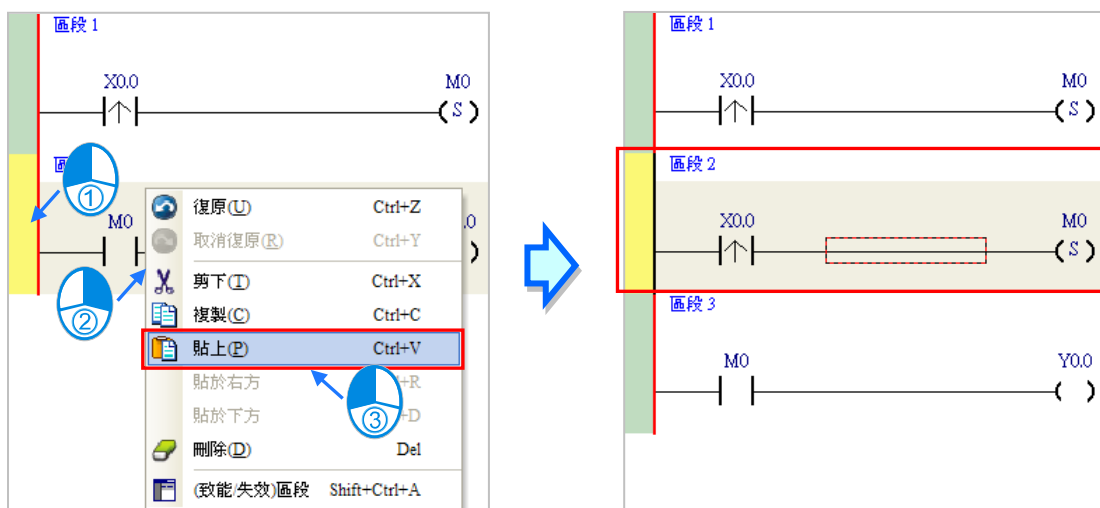
操作項目	功能說明
復原	復原方才的編輯動作 (最多可復原至 20 個步驟)
取消復原	取消復原而重複方才的動作
剪下	剪下選取的裝置、區塊或階梯圖區段
複製	複製選取的裝置、區塊或階梯圖區段
貼上	將方才複製或剪下的物件貼在目前的編輯位置
貼於右方	將物件貼在目前選取的編輯區右方 (代表與選取區塊串聯)
貼於下方	將物件貼在目前選取的編輯區下方 (代表與選取區塊並聯)
刪除	刪除選取的裝置、區塊或階梯圖區段
區段致能	將選取的區段狀態設為致能
區段失能	將選取的區段狀態設為失效 (失效區段則會於編譯時被略過)
自動生成變數	於功能塊的接腳上使用，可自動生成變數
新增至裝置監控表	於所選取的接腳上使用，將使裝置快速新增至監控表

接下來則繼續本章範例的操作。

- (1) 首先選取**區段 1** 的整個階梯圖區段後按下滑鼠右鍵，之後於快捷選單中點選「複製」。




- (2) 接著選取**區段 2**，並在按下右鍵後於快捷選單中點選「貼上」，此時方才所複製的階梯圖區段便會被插在目前選取的**區段 2** 上方，而原本的**區段 2** 則將自動遞增編號為**區段 3**。

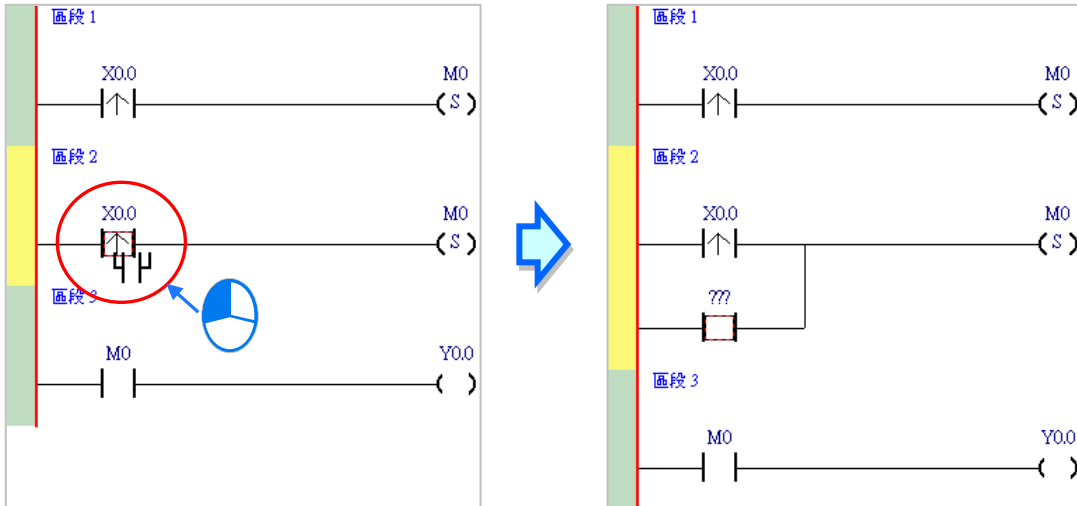


4

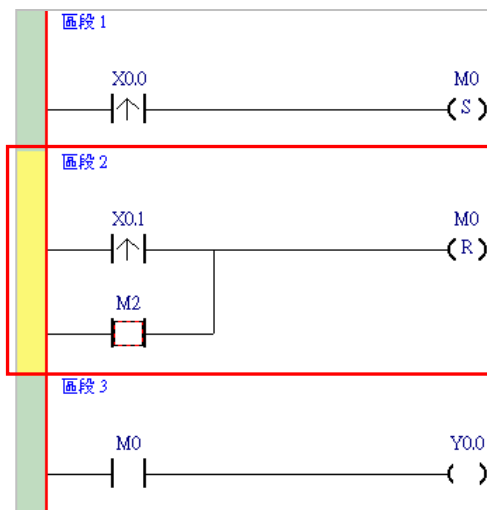
4.5.5 基本編輯 - 並聯接點裝置

- (1) 首先請於圖示工具列中點選  圖示以將鼠標切換為**接點工具**，接著再將滑鼠移至目前**區段 2** 當中的輸入接點，而此時鼠標亦會變為接點的形狀；刻意將鼠標稍微往下方偏移，待鼠標符號變為並聯的形狀後按下滑鼠左鍵，如此便可完成一個並聯型態的區塊。

4

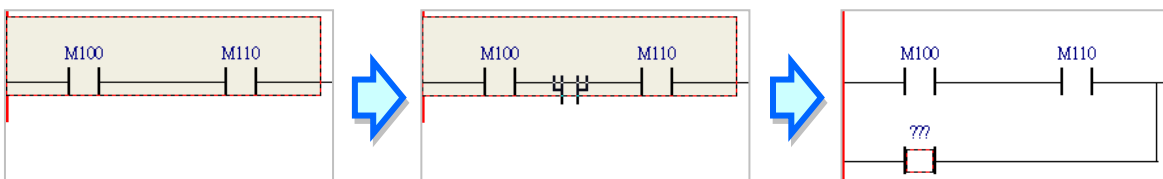


- (2) 接著請參考前述的方法將**區段 2** 修改成如下圖的程式。




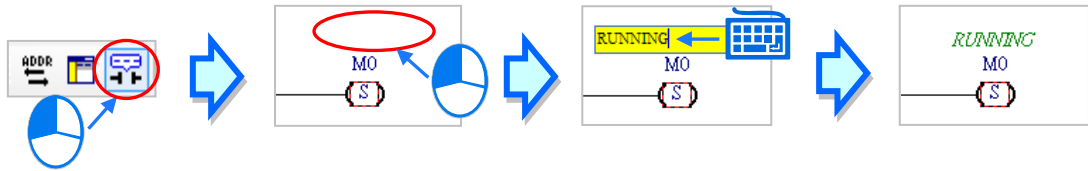
補充說明

當要與某個區塊進行並聯時，只要先做區塊選取之後，再比照上述的方式操作即可。

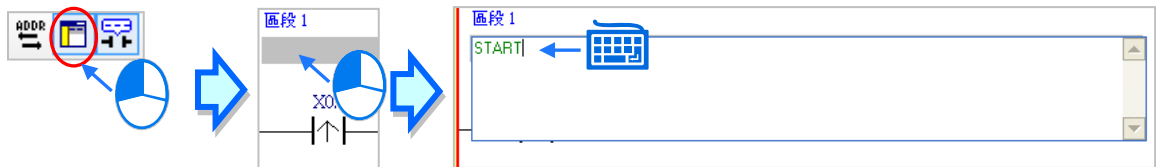


4.5.6 基本編輯 - 編輯註解

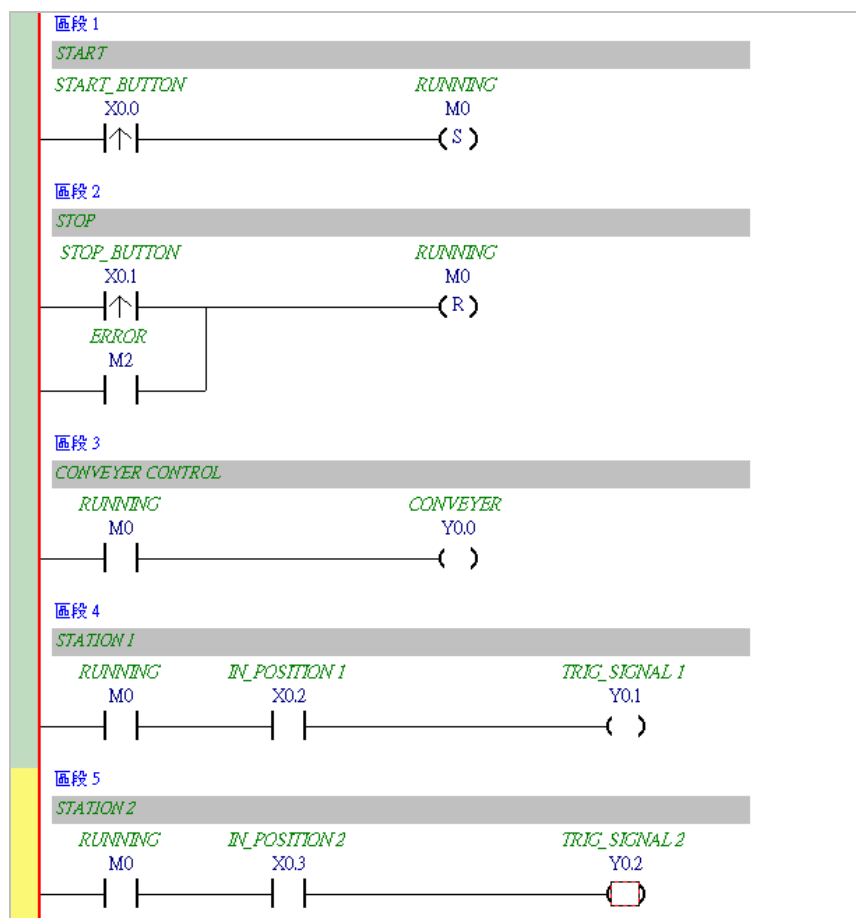
- (1) 首先請確認  圖示為按下的狀態後，直接於欲編輯註解的裝置名稱上方點擊滑鼠左鍵，接著便可於編輯欄位中輸入裝置註解，完成後按下鍵盤的【Enter】鍵即可。



- (2) 要為整個區段加入區段註解時，請先確認  圖示為按下的狀態，接著於區段編號下方的欄位上點擊滑鼠左鍵，之後即可於下拉的編輯區域中輸入註解；編輯時，按下【Shift】+【Enter】鍵便可換行，而編輯完成後，直接按下鍵盤的【Enter】鍵即可結束編輯。

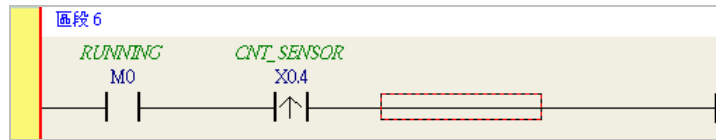


- (3) 接著請參考前述的說明完成如下的程式。



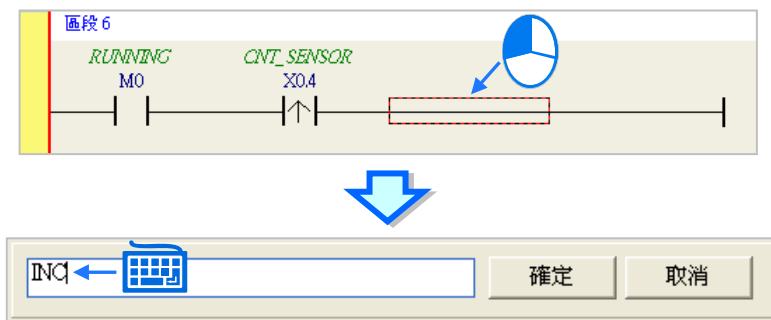
4.5.7 基本編輯 - 插入 API 應用指令

首先請於上一節完成的五個區段下方新增區段 6，並編寫如下的程式。之後由下列三種方式選擇其中一種來插入 API 應用指令。



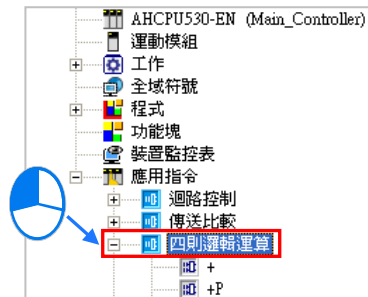
● 方法一

先點選欲插入指令的位置，接著直接以鍵盤輸入指令名稱（此範例為 INC）後按下【Enter】鍵即可。

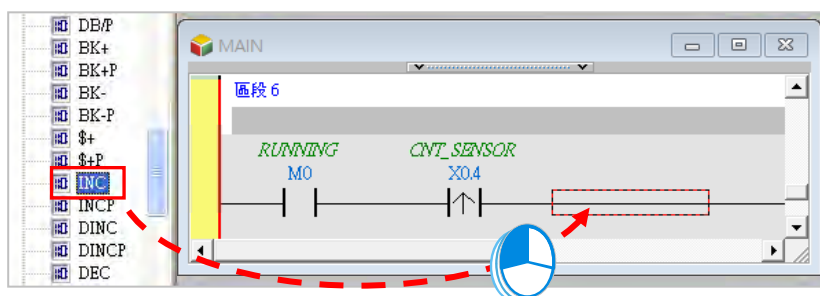


● 方法二


展開專案管理區的「應用指令」項目，並在找到欲插入指令的類別後將其展開。

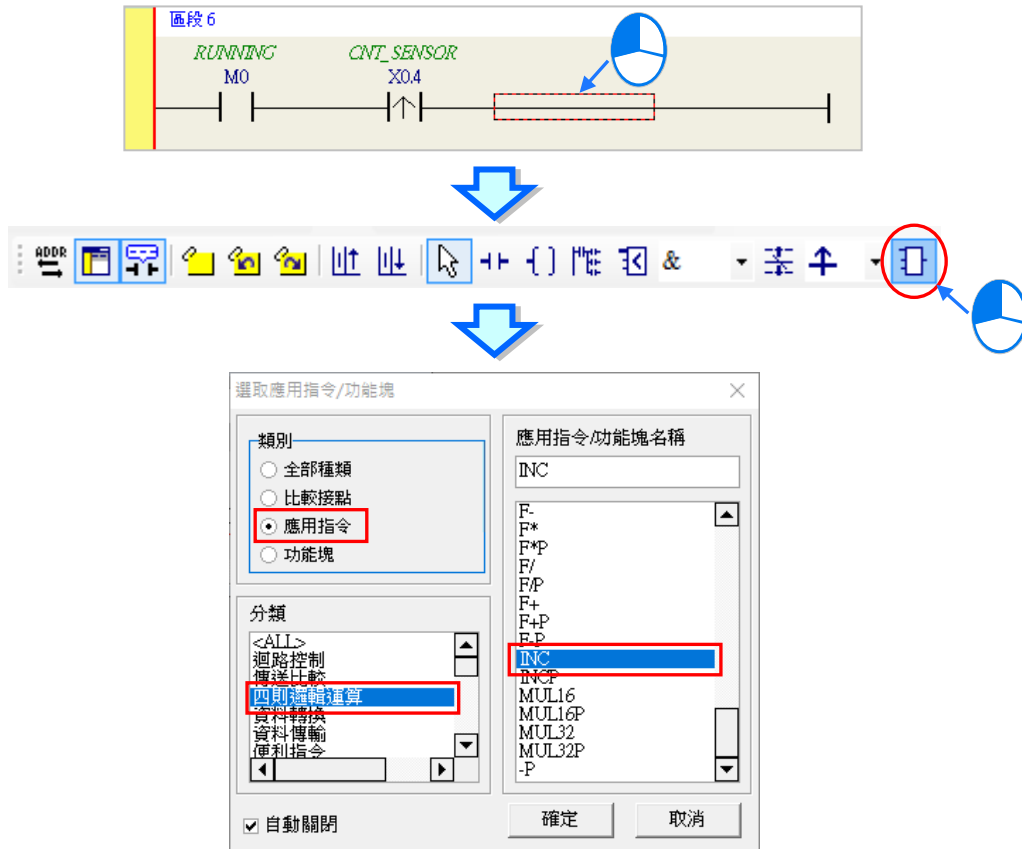


點選欲插入的指令項目（此範例為 INC）後按住滑鼠左鍵，接著將其拖曳至插入位置後放開即可。



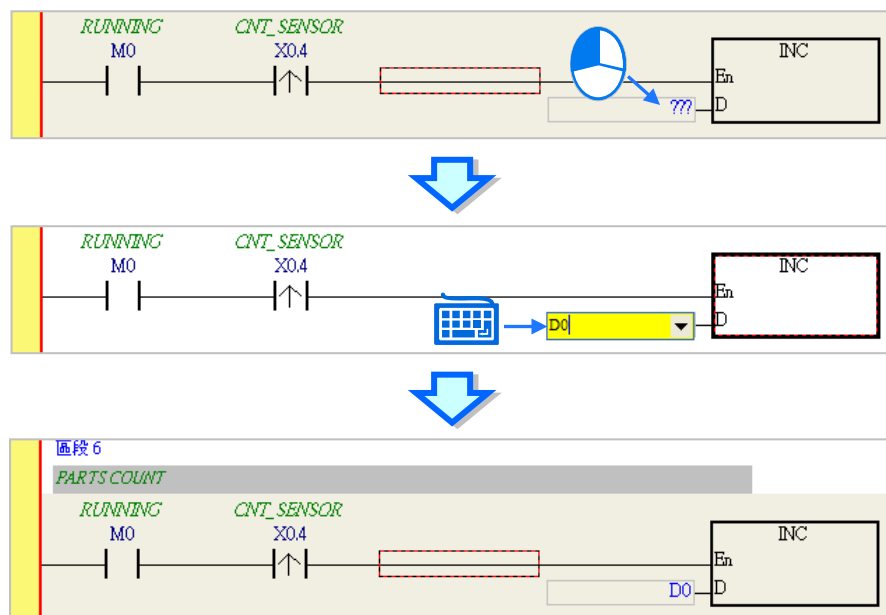
● 方法三

先於欲插入指令的位置上點擊滑鼠左鍵，之後點選  圖示，接著再於出現的選取視窗中點選要加入的 API 應用指令（此範例為 INC），完成後按下「確定」鍵即可。



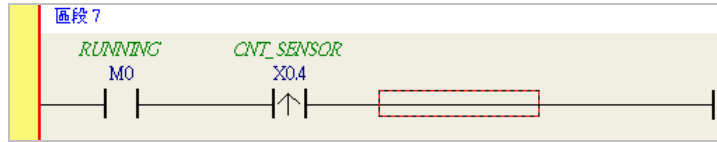
4


成功插入 API 應用指令後，請參考前述的方式修改指令的運算元，並完成如下圖的階梯圖程式。




4.5.8 基本編輯 - 比較接點的建立與常數的輸入格式

插入比較接點的方式，除了第 4.5.7 節所介紹的三種方法之外，還可以依據下列的步驟。在此之前，我們先在所有區段的下方新增一個區段 7，並編寫如下的程式。

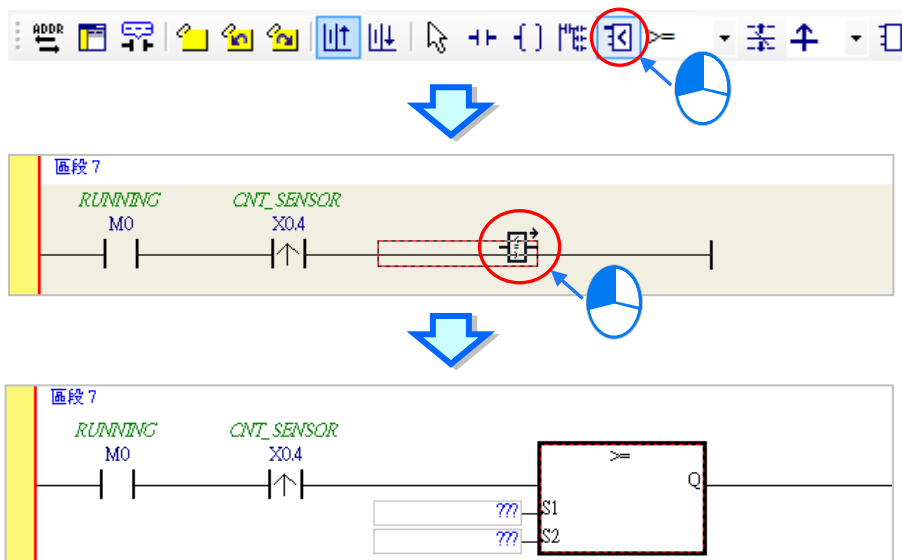


- (1) 先點選工具列圖示  並下拉選擇比較接點的類別。(此處請選擇 \geq)

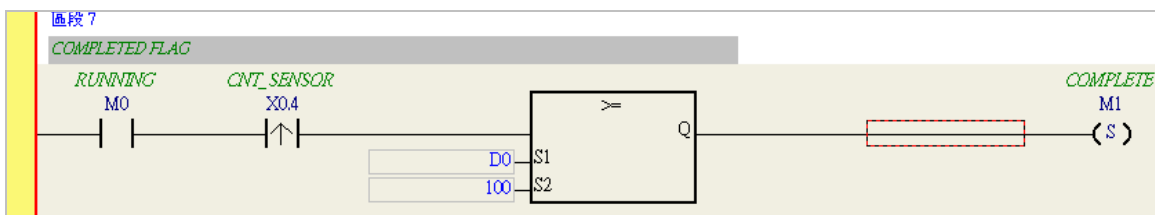


- (2) 點選工具列圖示 ，並將滑鼠移至欲插入比較接點的位置。此時滑鼠的偏移同樣會改變鼠標的圖示而決定插入的位置。在確定位置之後按下滑鼠左鍵即可插入比較接點。

4

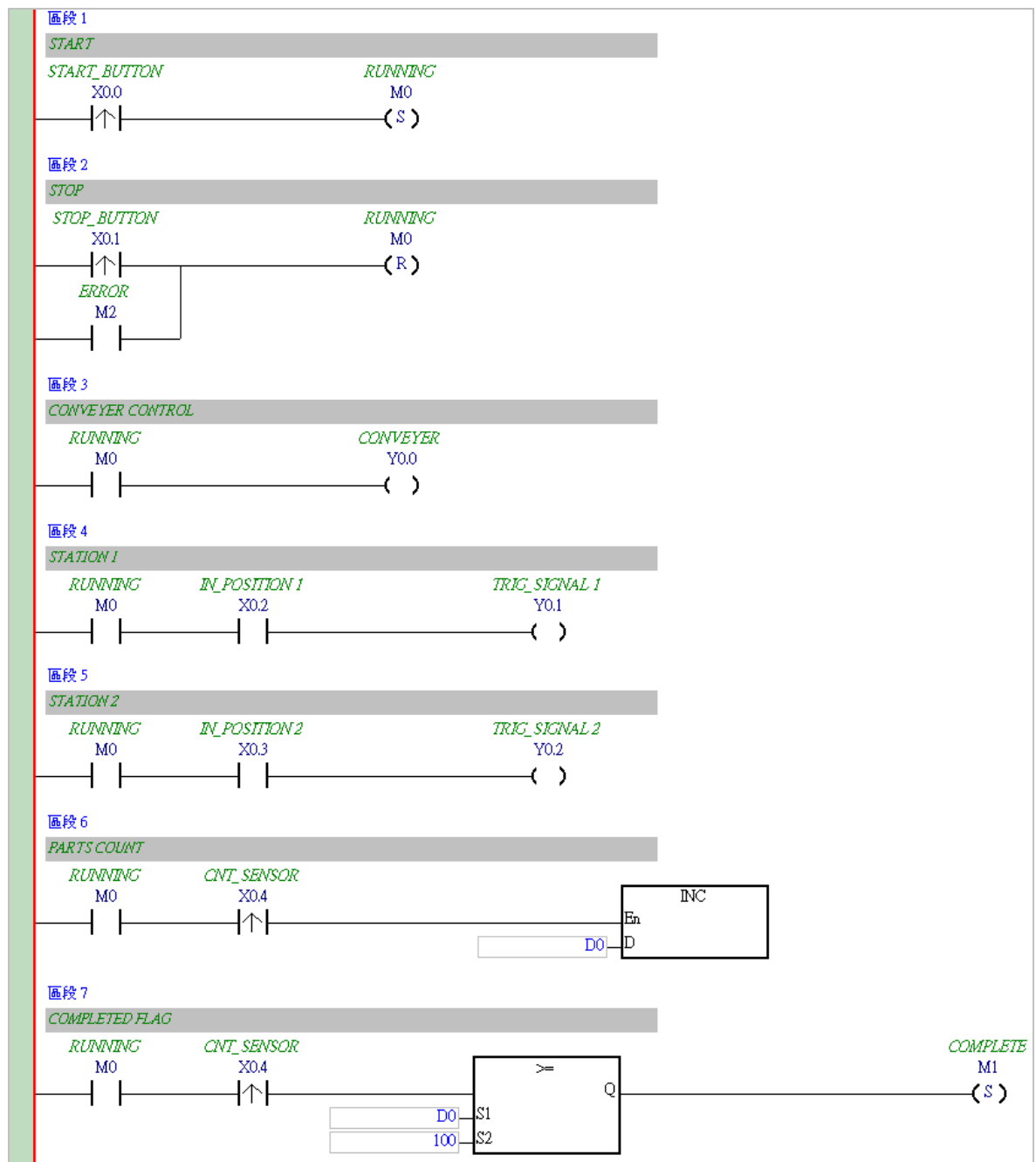


接著請參考前述的方式完成如下的階梯圖程式。而有別於早期在 WPLSoft 中使用「K」與「H」來表達常數的方式，當要在 ISPSOft 中輸入 10 進制常數時，僅需直接輸入數字即可，而 16 進制常數則必須於前方加上「16#」，如 16#7FFF；亦可使用「8#」與「2#」來表達 8 進制與 2 進制常數。



4.5.9 完成範例程式

階梯圖編程的基礎操作方式都已大致介紹完畢，而依據前面各節的說明，我們便可逐步完成如下圖的範例程式；在編輯的過程中，我們會發現左側的母線一直是呈現紅色的狀態，這代表此程式尚未完成編譯，而在接下來的章節中，我們將繼續介紹如何對完成的範例程式進行編譯，並將編譯後的程式下載至主機當中進行基本的測試。




*1. 上圖的範例程式亦會放置在 ISPSOft 安裝目錄下的 `..\Project\Example\Gluing_System_C\` 資料夾中。

*2. 更詳盡的階梯圖編程說明，請參考第 10 章的相關介紹。

4.5.10 程式的檢查與編譯


當完成一個程式的編輯之後，接著便可對其進行文法檢查或編譯的工作。其中「檢查」只會對目前工作視窗的內容進行語法與結構的確認，而「編譯」則會針對整個專案進行確認，並在檢查無誤後自動產生可讓 PLC 運行的執行碼，且編譯成功後，階梯圖左側的母線也會轉變為黑色的狀態。

● 檢查

於功能工具列中點選 **編譯 (C)** > **檢查 (C)** 或直接於圖示工具列中點擊  圖示即可。

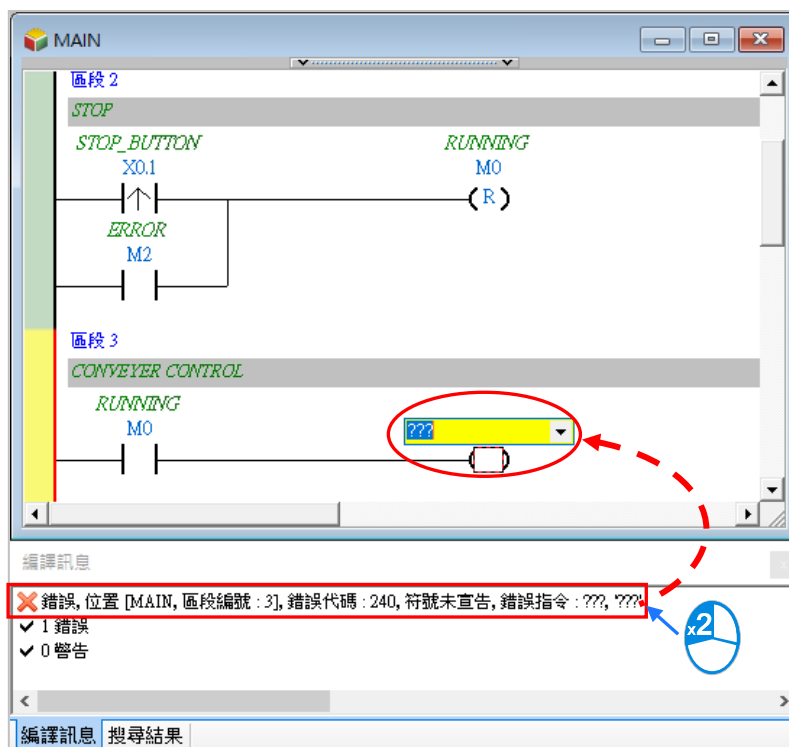


● 編譯

於功能工具列中點選 **編譯 (C)** > **編譯 (A)** 或直接於圖示工具列中點擊  圖示即可。



執行完畢後，於 ISPSOft 的「編譯訊息」視窗中將會顯示確認的結果，而當發現錯誤時，該視窗便會列出相關的訊息，直接於該訊息上點擊兩下滑鼠左鍵，系統便會自動引導使用者移動至產生該錯誤的位置，而接著便可進行修正的工作；當修正完畢後，只要再次執行「檢查」或「編譯」的動作即可。

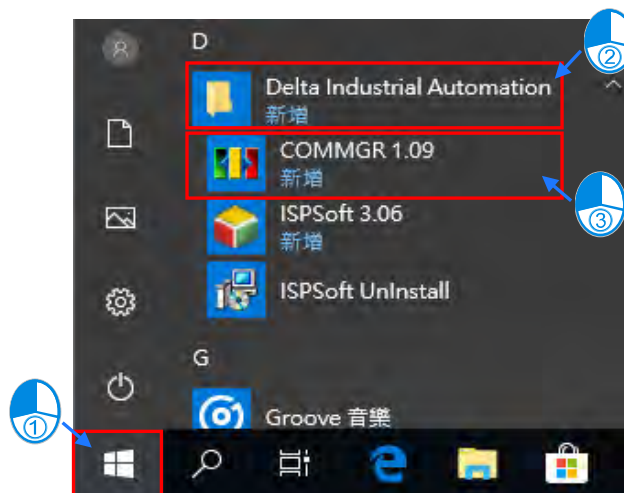


4.6 測試與除錯

4.6.1 建立連線

要將程式與參數下載至主機或是進行線上監控之前，必須先在 ISPSOft 與主機之間建立連線，以下針對本章範例所選用的主機 **AHCPU530-EN**，並透過 **USB** 的連線方式來稍做說明，若已經與主機建立連線者可直接跳至第 4.6.2 節下載程式與組態參數。

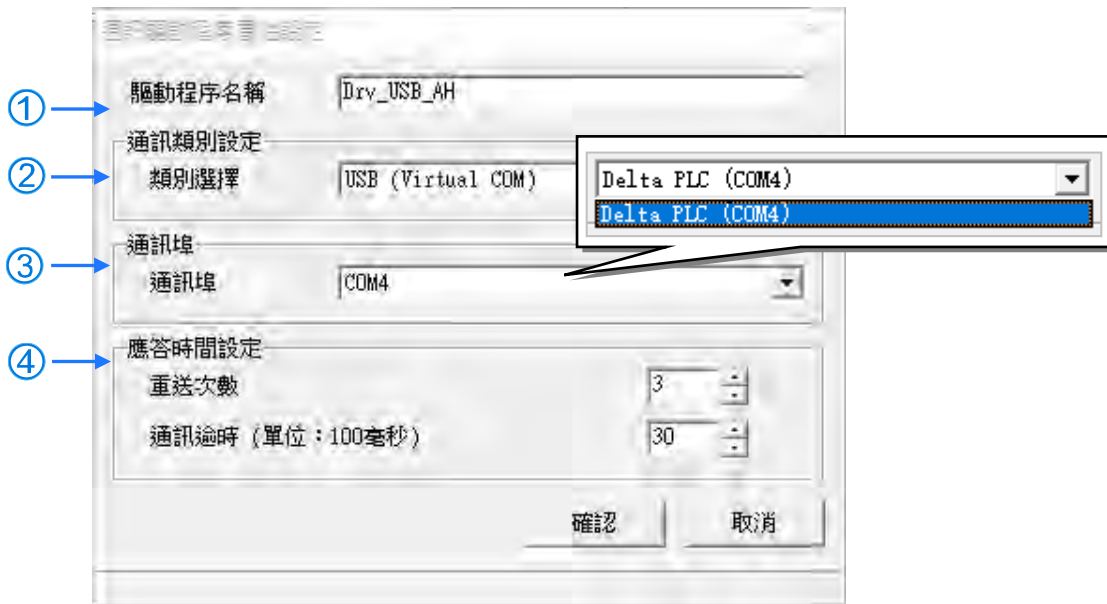
- (1) 將模組正確安裝至主背板上，且安裝的插槽編號與 HWCONFIG 中的配置必須一致；接著確認接線無誤後，將主機上電。
- (2) 將主機與電腦透過 USB 通訊電纜接續在一起，若電腦中已安裝 AH500 系列的 USB 驅動程式，此時於 Windows 的裝置管理員中便會出現「Delta PLC」的項目，並且會分配到一個 COM Port 的編號。關於安裝 USB 驅動程式的相關說明與注意事項請參考 附錄 A 當中的內容。
- (3) 確定**通訊管理員 (COMMGR)** 已啟動並常駐在 Windows 的系統列中，如果沒有啟動請至「程式集」中依下列路徑至「Delta Industrial Automation」→「COMMGR」目錄中點擊「COMMGR」的程式執行捷徑。



- (4) 於系統列的 COMMGR 圖示上雙擊滑鼠左鍵，之後便會開啟 COMMGR 的管理視窗；接著請於 COMMGR 視窗的右側點擊「新增... (A)」鍵以新增一組 Driver。

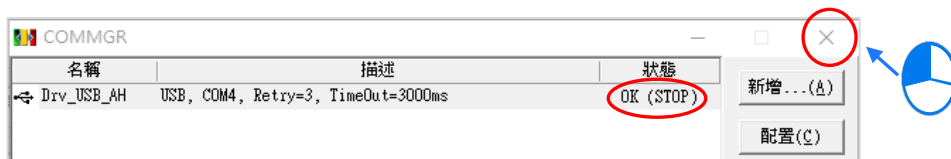


(5) 於 Driver 設定視窗中依序完成各欄位的參數設定，完成後按下「確認」按鈕即可。

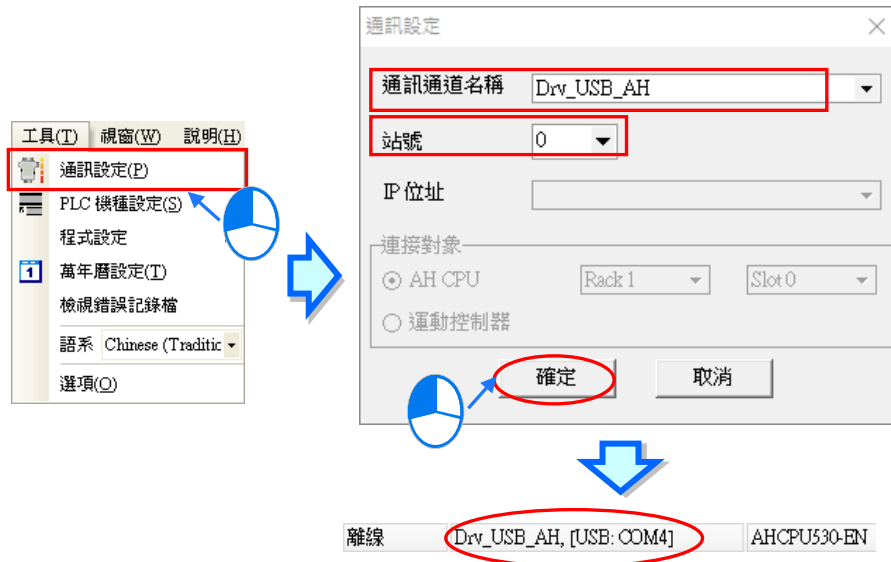


- ① 於「驅動程序名稱」欄位輸入此 Driver 的識別名稱。
- ② 於「通訊類別設定」的「類別選擇」欄位下拉選擇「USB (Virtual COM)」的連線型態。
- ③ 於「通訊埠」欄位下拉選擇欲建立連線通道的通訊埠口，若前述 (1) (2) 兩個步驟已正確完成操作，則此時於下拉選單中便會顯示已連線的主機裝置名稱與其 COM Port 編號。
- ④ 設定通訊回應的相關參數。其中「重送次數」參數是設定連線錯誤時的重試次數，而「通訊逾時」參數則是設定重試的間隔時間。

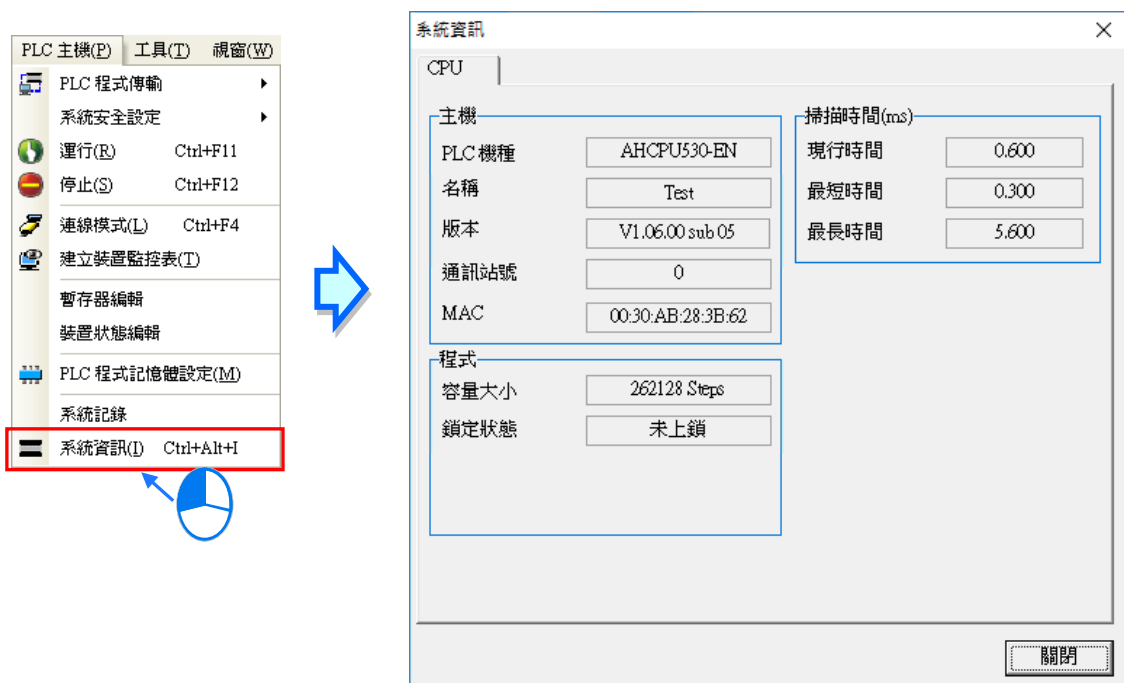
(6) 確認 Driver 的 Status 欄位為 OK 便可關閉視窗。而此時 COMMGR 仍會繼續常駐在 Windows 的系統列中。



啟動 ISPSOft 之後，於功能工具列中依序點擊 **工具 (I)** > **通訊設定 (P)**，接著請於通訊設定視窗的「**通訊通道名稱**」欄位處下拉選擇剛才所建立的 COMMGR Driver，而站號欄位則保持**站號 0**的設定即可。完成後按下「**確定**」鍵離開，此時 ISPSOft 的狀態列便會顯示目前的設定。



(7) 於功能工具列中點選 **PLC 主機 (P)** > **系統資訊 (!)**，若已可順利與主機進行通訊時，ISPSOft 便會從主機中擷取相關資訊，並將其顯示於畫面中。

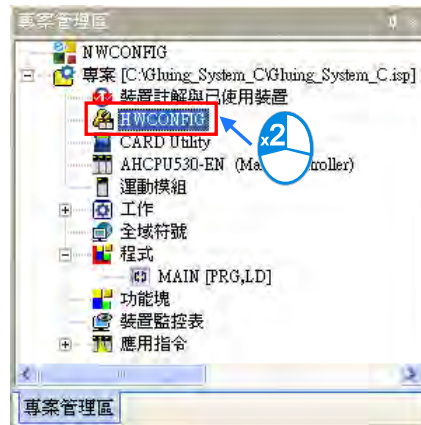


4.6.2 下載程式與組態參數

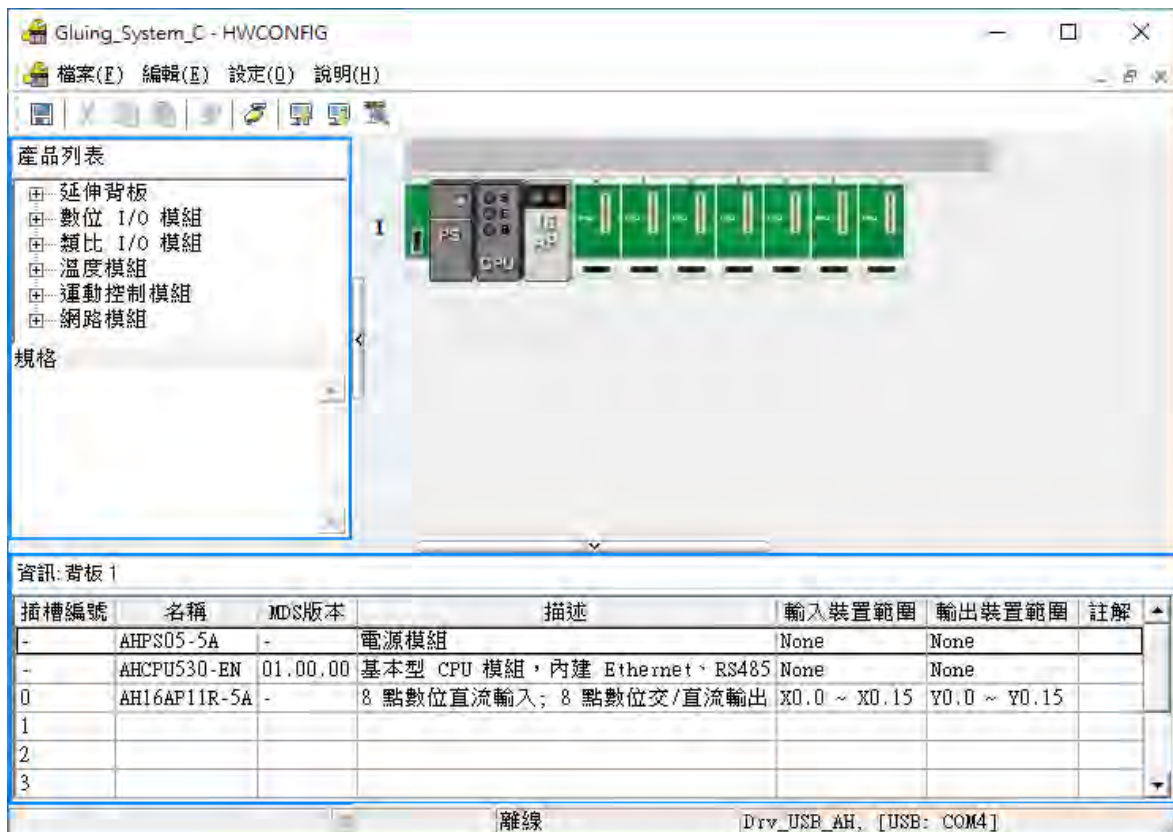
當已確定 ISPSOft 可與主機正常進行連線之後，接著便可將專案中的參數與程式下載至主機當中。首先請啟動 ISPSOft 並開啟於前幾節內容中所建立好的專案，而此範例中需要下載的參數共有 2 項，分別是硬體規劃參數與專案程式。


- 下載硬體規劃參數

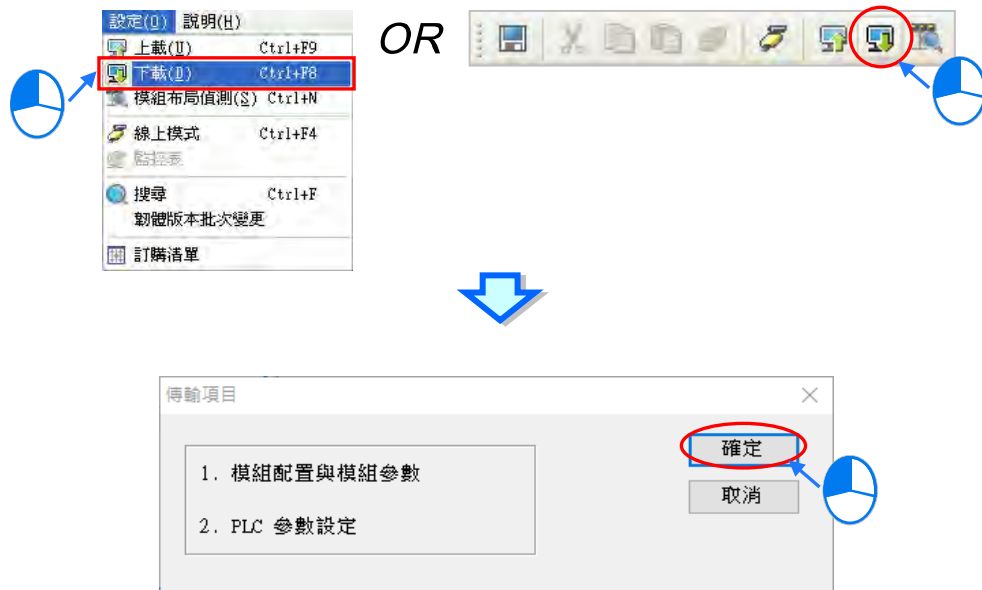
(1) 於專案管理區的「HWCONFIG」項目上雙擊滑鼠左鍵，以開啟硬體規劃工具。



(2) 進入 HWCONFIG 的操作環境後，畫面中便會顯示先前所規劃的硬體配置，而在下載之前請再次確認實際硬體的配置與此處的規劃是否相同。




- (3) 於 **HWCONFIG** 的功能工具列點擊 **設定 (O) > 下載 (D)** 或點擊圖示工具列的  圖示，接著畫面便會出現下載項目的提示視窗，按下「**確定**」按鈕之後即可開始進行下載。



- (4) 成功下載後，主機本體上的 **BUS FAULT** 燈號應要呈現熄滅的狀態，此時便可將 **HWCONFIG** 的視窗關閉。但若 **BUS FAULT** 燈號仍呈現恆亮或閃爍的狀態時，代表 **PLC** 本身處於異常狀態，此時請再次確認 **HWCONFIG** 中的規劃與實際的硬體配置是否一致，或參考該機種的操作手冊進行錯誤排除。更進一步的 **HWCONFIG** 操作說明請參考第 3 章的相關內容。



● 下載專案程式

確認程式編譯無誤後，於 **ISPSOFT** 的功能工具列點選 **PLC 主機 (P) > PLC 程式傳輸 > 下載 (D)** 或點擊圖示工具列的  圖示，接著便會出現下載項目的勾選視窗，此時可將項目中的「**程式**」與相關註解項目勾選起來，以便日後可由 **PLC** 中將程式重新上傳，完成後按下「**重新傳輸**」即可。




4.6.3 連線測試

當程式下載至 PLC 主機後，我們便可經由 ISPSOft 來對 PLC 的執行狀況進行監控。而於 ISPSOft 中，監控功能又可分為「裝置監控」與「程式監控」。

監控模式	說明
 裝置監控	可透過監控表來即時監控主機目前的裝置狀態，且因在此模式下，ISPSOft 僅需更新裝置狀態，因此 ISPSOft 目前開啟的程式與主機內部的程式無須一致。
 程式監控	在此模式下，系統會即時將程式的運作狀況顯示於程式畫面中，也因此系統會要求 ISPSOft 目前開啟的程式必須與主機內部的程式一致。

*.裝置監控模式可單獨啟動，而程式監控模式則必須伴隨裝置監控模式一起啟動。

直接於功能工具列中點選 **PLC 主機 (P)** > **連線模式 (L)**，或按下圖示工具列的  圖示便可進入線上監控，而此時系統將會同時啟動「裝置監控」與「程式監控」兩種模式：

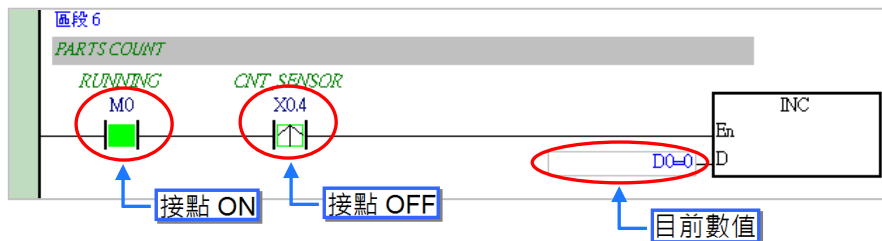
4





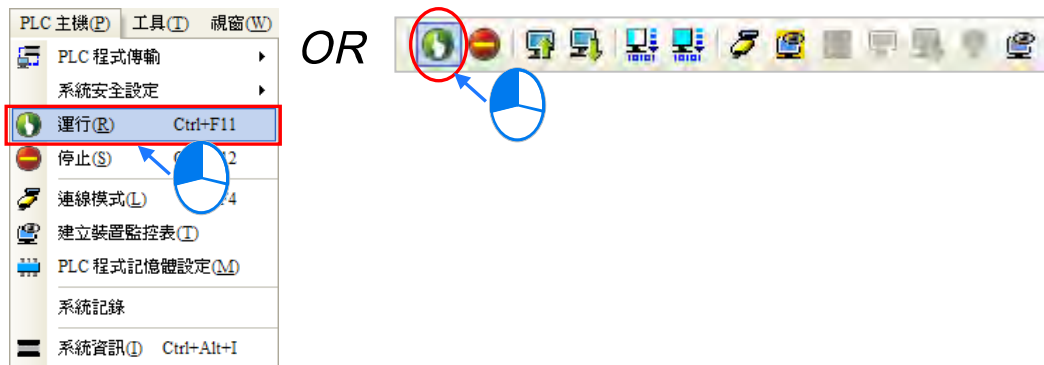
進入線上模式時，由 ISPSOft 的狀態列，便可檢視目前的掃描時間、通訊狀況與主機狀態。



且啟動**程式監控**的模式後，於原本編輯的程式畫面中便會即時顯示目前的裝置狀態。

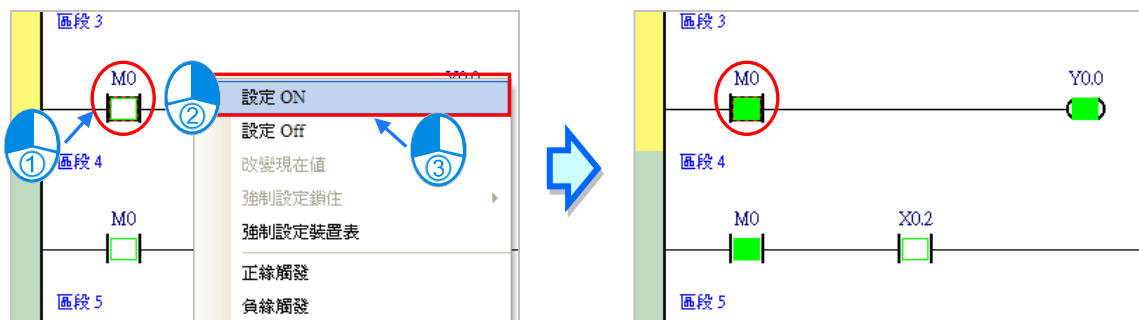


除了可藉由切換主機本體的 RUN/STOP 開關來改變主機的運轉狀態之外，於 ISPSOft 中亦提供相同的功能供使用者進行操作；當於 ISPSOft 的功能工具列中點選 **PLC 主機 (P) > 運行 (R)** 或按下圖示工具列的  圖示時，PLC 主機便會切換至執行 (RUN) 狀態；而當要將 PLC 主機切換回停止 (STOP) 狀態時，只要再次於功能工具列中點選 **PLC 主機 (P) > 停止 (S)** 或直接按下圖示工具列的  圖示即可。



於線上監控的模式下，先點選任一裝置後按下滑鼠右鍵，之後於快捷選單中便會有許多的控制選項可供使用，利用這些功能便可於測試過程中暫時改變某個裝置的狀態或內容值。

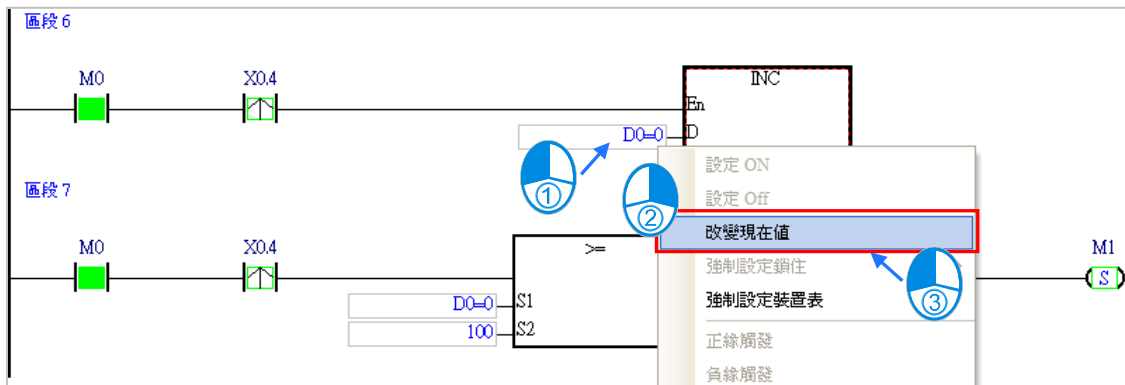
⚠ 變更裝置狀態前，請務必確認所做的操作不會影響系統運作，或造成系統與人員的傷害。



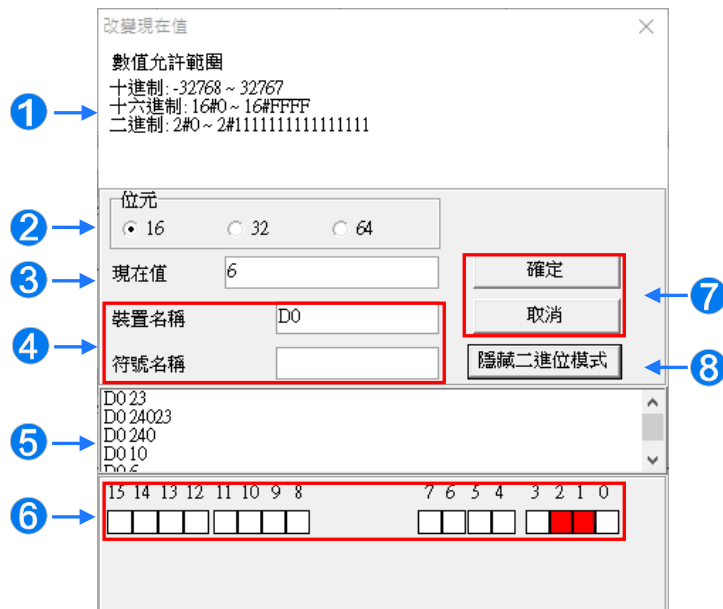
下表則為各操作項的說明簡述，而其中**強制設定鎖住**的功能只可用於 X 與 Y 接點。

操作項目	說明
設定 ON	將點選的接點狀態設置為 ON。
設定 OFF	將點選的接點狀態設置為 OFF。
正緣觸發	不管被點選的接點狀態為何，系統皆會將該接點先設為 OFF 之後，再設為 ON。
負緣觸發	不管被點選的接點狀態為何，系統皆會將該接點先設為 ON 之後，再設為 OFF。
強制設定鎖住	將 X 或 Y 接點強制鎖定在 ON 或 OFF 的狀態。
強制設定裝置表	利用表格的方式，批次對 X 或 Y 接點的強制鎖定狀態進行管理。

當要改變某個裝置的內容值時，請先點選該裝置後按下滑鼠右鍵，接著於快捷選單中點選「改變現在值」，之後便可由數值設定視窗中設定該運算元的內容值。



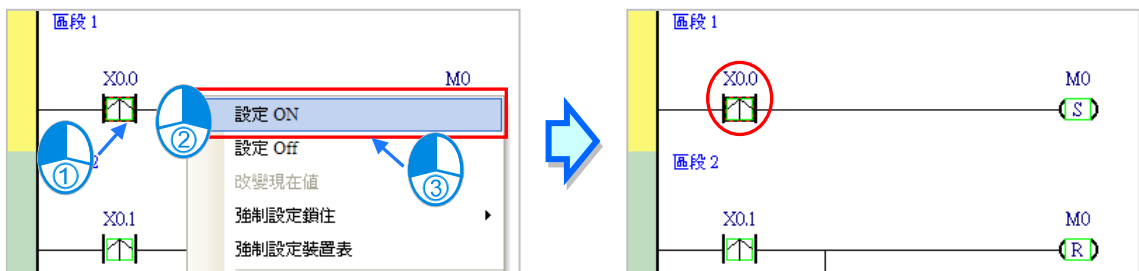
下列即為數值設定視窗的相關說明。



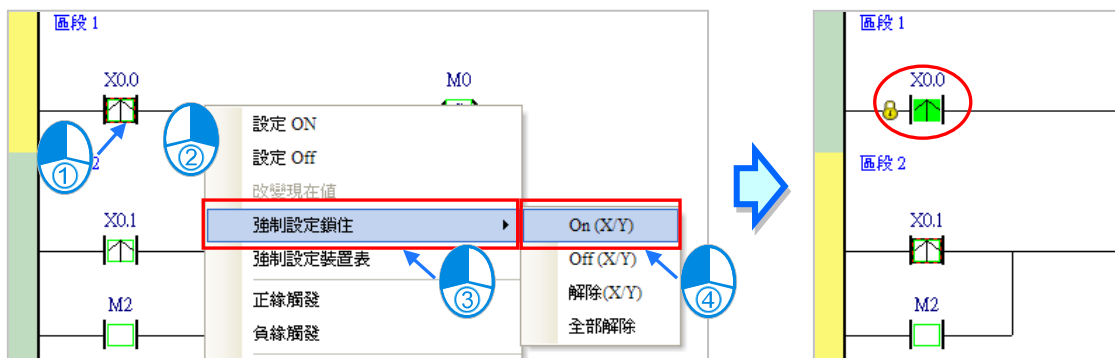
- ① 提示訊息。
- ② 於此選擇輸入的數值格式為 16 位元或 32 位元或 64 位元。
- ③ 於此輸入欲變更的數值。
- ④ 欲變更現在值的裝置名稱或變數符號名稱。
- ⑤ 數值變更記錄。(顯示格式：裝置名稱 變更數值)。
- ⑥ 當開啟二進位模式時，用滑鼠便可直接於此處設定各個 Bit 的 ON/OFF 狀態。
- ⑦ 按下「確定」鍵可套用設定值，按下「取消」鍵則可關閉此設定視窗。
- ⑧ 點擊此鍵可開啟或隱藏二進位模式。

在本範例中，X0.0~X0.15 與 Y0.0~Y0.15 為配置給數位 IO 模組 AH16AP11R-5A 的輸入/輸出裝置範圍，因此當硬體參數下載至 PLC 後，裝置 X0.0~X0.15 的狀態便會與實際模組的輸入狀態保持同步。

此時若嘗試於程式監控畫面中，將 X0.0~X0.15 的接點狀態直接設為 ON 或設為 OFF，該接點的狀態仍舊會依據外部實際輸入的訊號而被重複刷新。



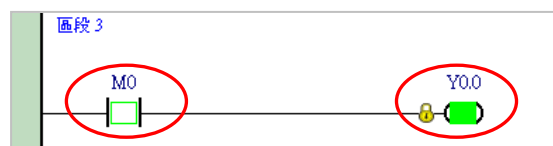
不過在進行測試的過程中，還是可藉由**強制設定**的功能來將 X 接點鎖定在固定的狀態。設定時，請先點選欲設定的 X 或 Y 接點，接著按下滑鼠右鍵並於快捷選單中點選「**強制設定鎖住**」，之後便可透過下階的項目來進行相關的操作。而處於鎖定狀態的 X 或 Y 接點，其圖示旁便會出現一個大鎖的符號。



4

強制設定鎖住	說明
On (X/Y)	將點選的 X 或 Y 接點鎖定在 ON 的狀態。
Off (X/Y)	將點選的 X 或 Y 接點鎖定在 OFF 的狀態。
解除 (X/Y)	解除點選接點的鎖定狀態。
全部解除	解除所有接點的鎖定狀態。


當程式中的 Y 接點處於**強制設定**狀態時，代表該接點的輸出狀態將不會受到程式執行結果的影響。



*. 離開線上監控時並不會自動解除主機內部的強制設定狀態，因此結束測試後，請確認是否須將強制設定的狀態全部解除。

下述為建立監控表的方式，且不論是離線或連線的狀態下皆可進行監控表的建立。

● 方法一

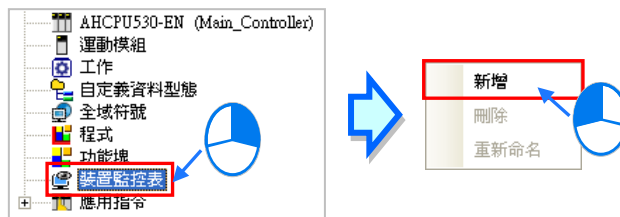
於功能工具列中點選 **PLC 主機 (P)** > **建立裝置監控表 (I)**，或於圖示工具列中點選  圖示。



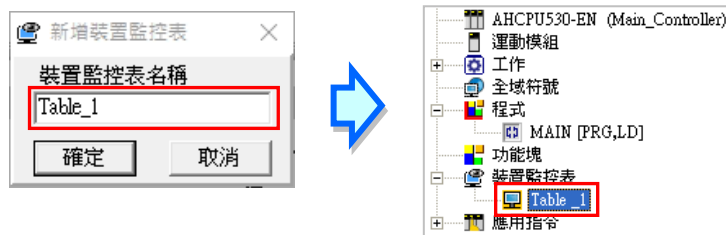
4

● 方法二

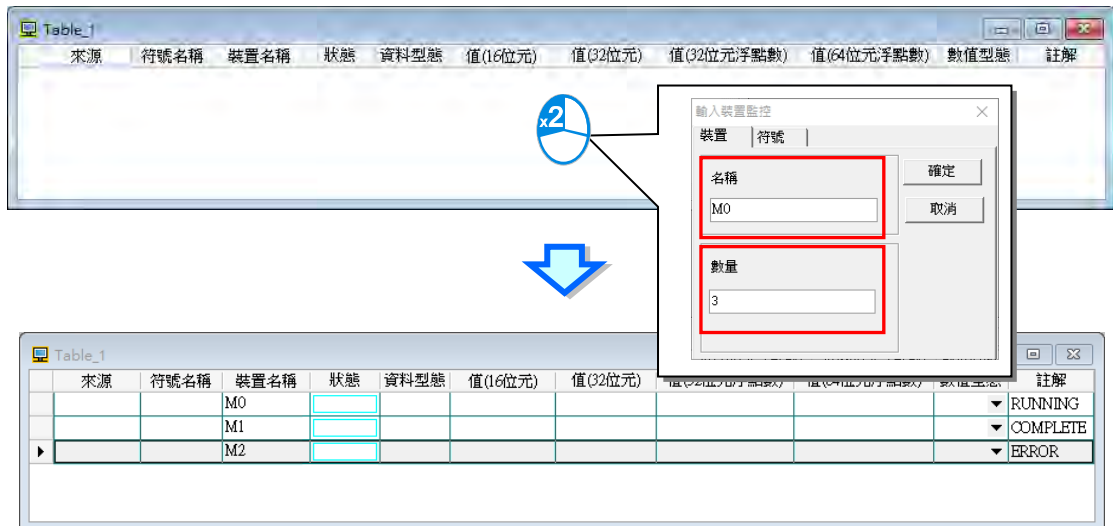
在專案管理區點選「裝置監控表」後按下滑鼠右鍵，快捷選單中點選「新增」。



依據上述方式操作之後，接著在彈出的視窗中輸入監控表的名稱，完成後按下「確定」，專案管理區的「裝置監控表」項目下便會產生一個物件，當後續要開啟監控表時，只要於該物件上雙擊滑鼠左鍵即可；一個專案中可建立多個監控表，而所建立的監控表亦將隨著專案一起被儲存。



當新增一個監控表時，畫面便會自動開啟該監控表的管理視窗，此時便可直接於該視窗中建立監控項目。欲建立項目時，請於**監控表**的空白處雙擊滑鼠左鍵，或直接輸入裝置名稱，接著可以在彈出的視窗中輸入啟始的裝置位址與欲加入的數量，不過須注意每個監控表至多只可建立 100 個監控項目。

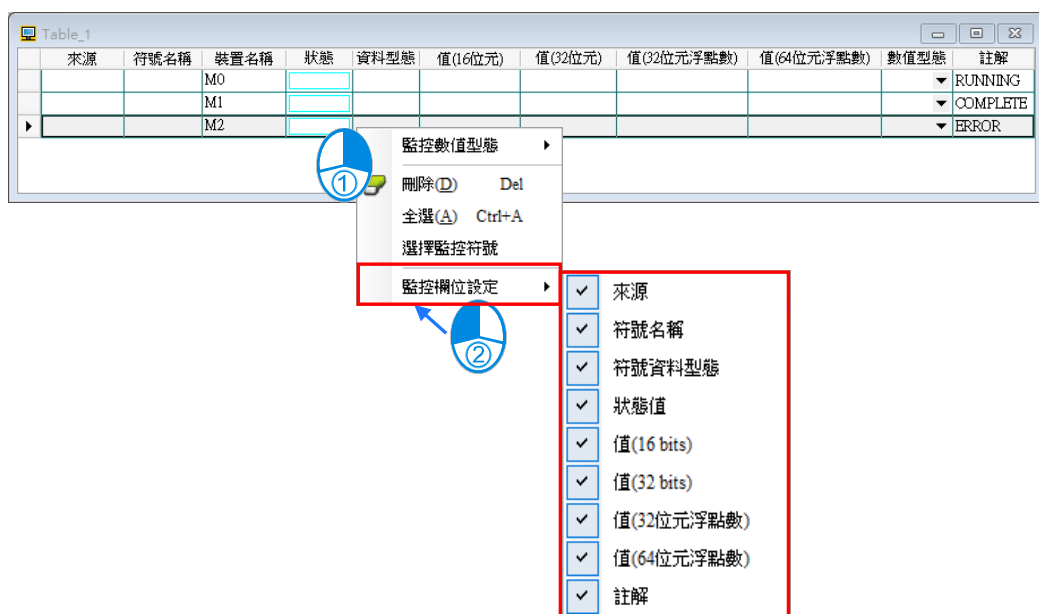


當加入裝置項目時，切換鍵盤的【Insert】鍵便可選擇使用**插入**或**取代**模式；ISPSOft 視窗的狀態列中便可得知目前所選擇的狀態。

當選擇**插入**模式時，新增的監控項目將會插於目前監控表中被選取的項目上方；而選擇**取代**模式時，新增的監控項目便會直接將目前所選取的監控項目取代掉。



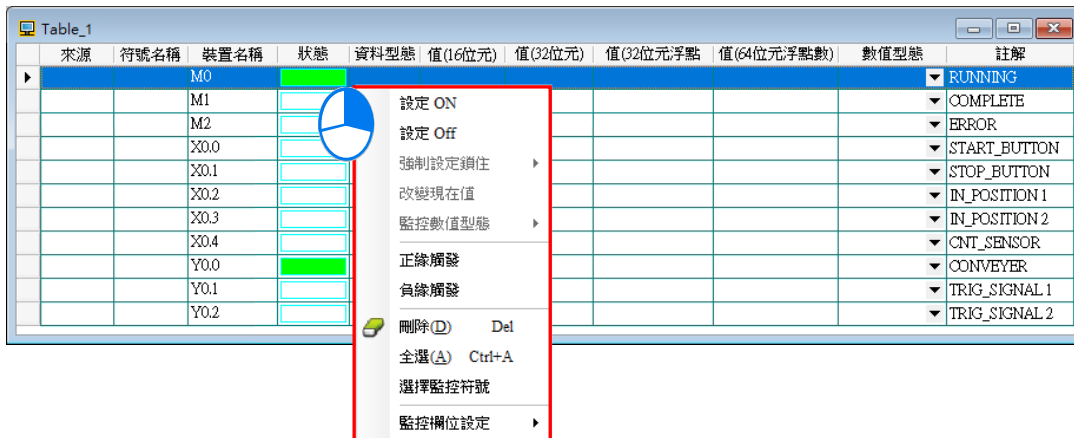
當要隱藏某些監控表的欄位時，請於監控表中按下滑鼠右鍵，接著於選單中點選「**監控欄位設定**」，之後便可透過其下階的項目來進行設定，而當取消勾選時，該欄位便會被隱藏。



下表則為監控表中各欄位的相關說明。

監控表欄位	說明
來源	顯示變數符號的來源。
符號名稱	顯示變數符號的名稱。
裝置名稱	顯示監控的裝置名稱。
狀態	當監控的對象為位元或接點時，此欄位會顯示其 ON/OFF 狀態。
資料型態	若監控的對象為變數符號，則此欄位會顯示該變數符號的資料型態。
值 (16 位元)	實際連線監控時，此欄位會以 16 位元的型態來顯示監控值。
值 (32 位元)	實際連線監控時，此欄位會以 32 位元的型態來顯示監控值。
值 (32 位元浮點數)	實際連線監控時，此欄位會以 32 位元的浮點數型態來顯示監控值。
值 (64 位元浮點數)	實際連線監控時，此欄位會以 64 位元的浮點數型態來顯示監控值。
數值型態	於此下拉可選擇連線監控時所顯示的數值格式。
註解	顯示監控裝置的 裝置註解 或監控符號的 符號註解 。

當建立好一個監控表後，可於線上模式下透過監控表來監控表格中的各個項目；而當處於線上模式時，於**監控表**中點選某個項目後下滑鼠右鍵，之後於快捷選單中便會出現與程式監控畫面相同的操作項，利用這些功能便可對監控項目的狀態或內容值進行變更。



依據本節所介紹的內容，對於我們所建立的範例程式便可進行簡易的測試與除錯，進一步的測試與偵錯功能，**第 18 章**的內容當中有更詳盡的介紹。

5

第5章 POU 與 TASK

目錄

5.1 關於程式組織單元 (POU) 與工作 (TASK)	5-2
5.2 認識程式組織單元 (POU)	5-3
5.2.1 POU 的編程架構與種類	5-3
5.2.2 ISPSOft 的 POU 架構	5-4
5.3 認識工作 (TASK)	5-5
5.3.1 ISPSOft 中的工作 (TASK) 管理	5-5
5.3.2 專案管理區的工作 (TASK) 項目	5-10
5.3.3 工作 (TASK) 中的 POU 執行順序	5-12
5.4 POU 的管理操作	5-14
5.4.1 POU 的建立與屬性設定	5-14
5.4.2 POU 的啟動狀態	5-18
5.4.3 POU 的刪除與複製	5-19
5.4.4 POU 密碼的設定與解除	5-21
5.4.5 匯出程式 POU	5-23
5.4.6 匯入程式 POU	5-24
5.4.7 匯出功能塊 POU	5-25
5.4.8 匯入功能塊 POU	5-26
5.5 工作 (TASK) 管理的操作	5-27
5.5.1 工作 (TASK) 屬性與中斷條件的設定	5-29
5.5.2 POU 的配置	5-34
5.5.3 POU 的排序	5-36
5.6 範例實作	5-37
5.6.1 POU 與 TASK 的編程範例	5-37
5.6.2 DVP 機種的中斷程式範例	5-41
5.6.3 AH5x0 機種的中斷程式範例	5-46

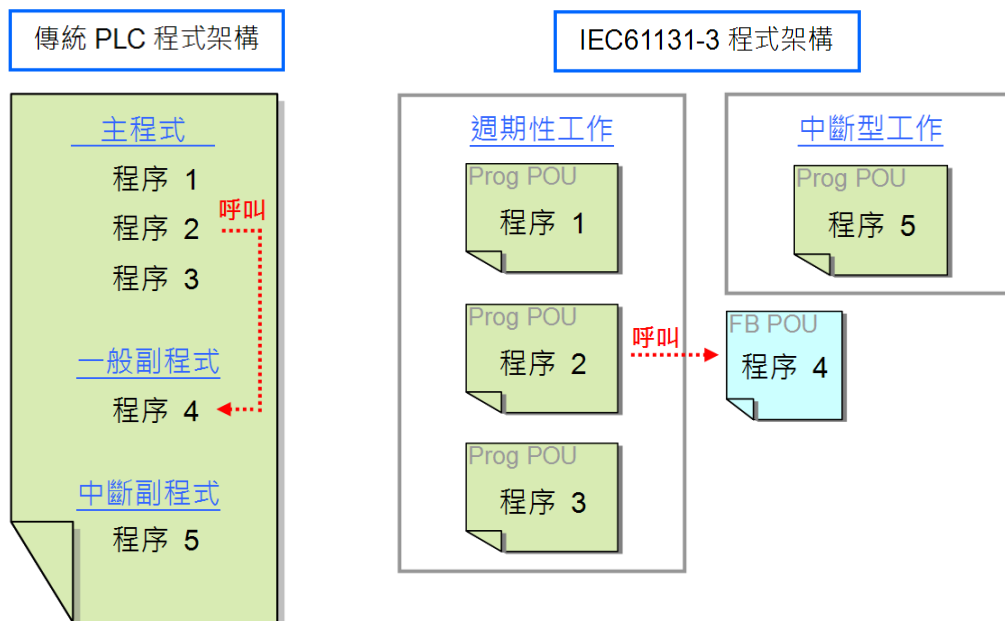
5.1 關於程式組織單元 (POU) 與工作 (TASK)

程式組織單元 (Program Organization Unit, POU) 與工作 (TASK) 是 IEC61131-3 中相當重要的編程觀念。此二者的最大意義在於讓 PLC 的編程由傳統的程式撰寫提升至專案管理的層次。下表即為傳統階梯圖與 IEC61131-3 在程式架構上的分別。

傳統 PLC 程式架構	IEC61131-3 程式架構
主程式	<ol style="list-style-type: none"> 1. 將單一的程式切割為許多獨立的程式單元 (Prog POU)。 2. 將建立的程式單元指定為週期性的工作 (TASK)。
一般副程式	<ol style="list-style-type: none"> 1. 將副程式建立為功能塊 (FB POU)。 2. 在其它的程式單元中對建立好的功能塊 (FB) 進行呼叫。
中斷副程式	<ol style="list-style-type: none"> 1. 為不同的中斷程序，各別建立獨立的程式單元 (Prog POU)。 2. 將建立好的程式單元指定至對應的中斷型工作 (TASK) 中。

下列即為兩種程式架構的示意圖，很明顯的可看出傳統的 PLC 編程必須從頭到尾的將程式撰寫在同一份原始碼中，而 IEC61131-3 的程式架構則是將程式切割為若干個程式單元，且每個程式單元皆可獨立開發，之後再透過工作管理的方式將各個程式單元分配到適當的執行位置。

5



當控制系統到達一定的規模時，其內部的控制程序必定也會有相當的複雜度，且程序與程序之間的互動也勢必更加緊密，此時若仍採用傳統的 PLC 程式架構，不論是開發過程或是日後的維護，必定會付出相當大的時間與人力；而在 IEC61131-3 的架構中，藉由 POU 的概念便可讓所開發的程式具有高重用性與高模組化的特性，且透過 TASK 的管理更可讓程式單元之間輕易的達到整合。而在接下來的幾個小節中，我們將陸續介紹 POU 與 TASK 的相關觀念，並說明如何在 ISPSOft 當中進行實作。

5.2 認識程式組織單元 (POU)

5.2.1 POU 的編程架構與種類

程式組織單元 (POU) 是建構 PLC 程式的基本元素，有別於傳統的 PLC 程式，由 IEC61131-3 所導入的程式架構，其特色在於將原本一大段的程式分割成若干個小單元，而這些小單元便稱之為 POU。如上一節的內容，傳統的 PLC 程式是將所有的程序，包括副程式都寫在同一段程式碼當中，當程式愈來愈龐大時，對於程式的維護與除錯都會是相當大的負擔；而 POU 的編程方式則是將程式依功能或特性，分割成若干個小單元，如此對於程式的開發與後續的維護都可以帶來相當大的便利；除此之外，因為其模組化的特性，不同的 POU 亦可交由不同的設計者來進行開發，對於人力的分配與專案的進行也都能帶來顯著的效益。

在 ISPSOft 中的 POU，依功能特性共可分為程式 (PROG)、功能塊 (FB) 與函式 (FC) 三種。

- 程式 (Program · PROG)

程式 POU 可依據其指定的工作 (TASK) 類型而決定其扮演的角色；若被指定至週期性的工作時，該程式 POU 扮演的便是主程式的角色；而若被指定至中斷型的工作時，該程式 POU 便扮演中斷副程式的角色；此外，在程式 POU 當中可以對功能塊 (FB) 進行呼叫。

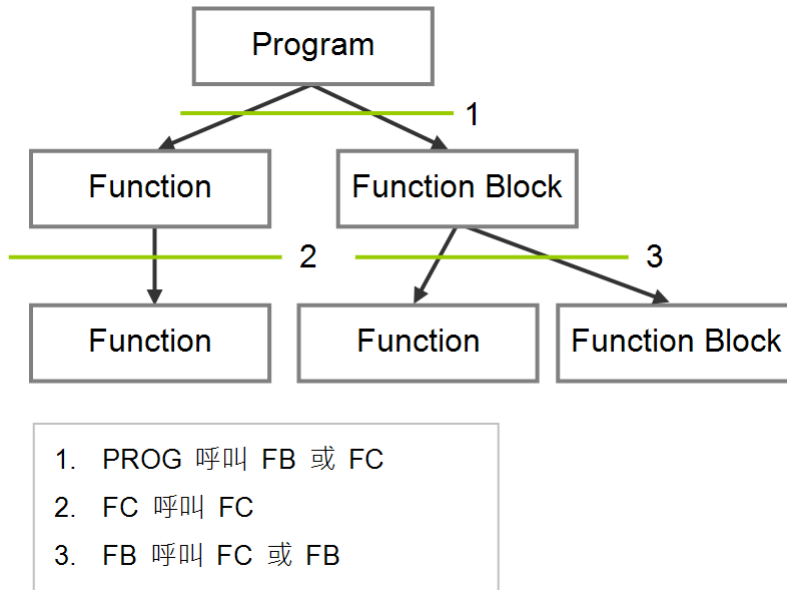
- 功能塊 (Function Block · FB)

功能塊 (FB) 內部可以宣告靜態型 (有記憶性的) 的變數符號，因此變數符號的值在運算之後可以被保持住，即使輸入相同的參數值，回應的數值仍會依據功能塊目前記憶的數值而與輸入的數值一起運算後再輸出，所以輸入相同的參數值，回應的數值不一定是一樣的。此外，功能塊的內部還可以再呼叫其他功能塊。

- 函式 (Function · FC)

每一個函式都是一個獨立封裝的程式單元，可以為其定義輸入參數，用於值操作。相較於程式和功能塊，函式沒有固定的記憶空間，所以不會儲存上一次的運算結果，也就使得同一函式在輸入參數相同的情況下，產生的結果相同，例如加法運算，目前只有 AS5xx 與 DVPxxMC 機種提供此功能。

下圖為各 POU 的呼叫對應關係。

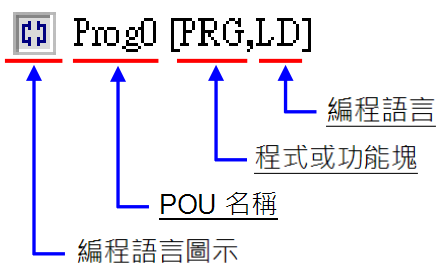
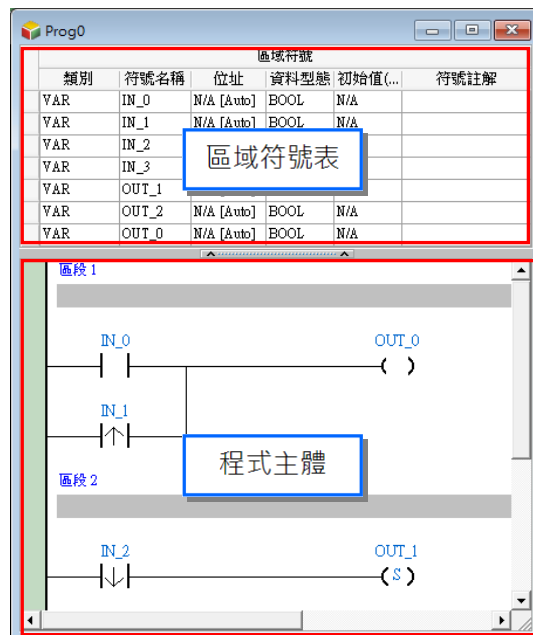
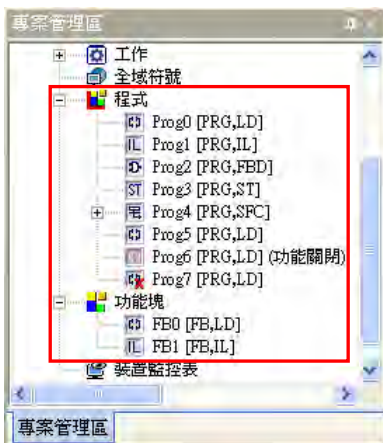


5.2.2 ISPSOft 的 POU 架構








5

所有使用者所建立的 POU 物件皆會在專案管理區中被列出，而其中程式類型與功能塊類型的 POU 會被分開管理；此外，根據該 POU 物件的編程語言及狀態，所顯示的圖示也會有所不同，而在每個 POU 名稱的後方亦會一併帶出相關的資訊。

於專案管理區的 POU 物件上雙擊滑鼠左鍵即可開啟該 POU 的編輯視窗，而視窗主要分為上下兩個部份，上半部為此 POU 的區域符號表，而下半部則為程式的主體，且隨著編程語言的不同，下半部的編輯環境也會隨之不同。關於變數符號表與程式編輯的相關說明，將於後續章節當中另行介紹。



下表則為各種 POU 圖示的相關說明。

圖示	說明
	代表此物件為使用 階梯圖 (LD) 所開發的 POU
	代表此物件為使用 指令列表 (IL) 所開發的 POU
	代表此物件為使用 功能塊圖 (FBD) 所開發的 POU
	代表此物件為使用 結構化語言 (ST) 所開發的 POU
	代表此物件為使用 循序功能圖 (SFC) 所開發的 POU
	代表此物件為使用 連續功能圖 (CFC) 所開發的 POU
	代表此物件為使用 C 語言 (C) 所開發的 POU
	當圖示呈現灰階代表此 POU 的狀態為「 關閉 」，而被設為關閉的 POU 將在編譯的過程中被略過，當然也不會被執行。
	當圖示出現紅色叉號代表此 POU 尚未被指定工作 (TASK)，而未被指定工作 (TASK) 的 POU 將在編譯的過程中被略過，當然也不會被執行。

*.關於 POU 的啟動狀態將於第 5.4.2 節中另行介紹，而工作 (TASK) 的指定則請參考下一節的說明。

5.3 認識工作 (TASK)

工作 (TASK) 的意義在於賦予各個程式 POU 一個明確的執行任務，並指定每個程式 POU 之間的執行順序或是啟動方式，而在同樣的應用場合，工作 (TASK) 也有被稱作**任務 (TASK)**。

換句話說，把各個**程式 POU** 比喻為球員的話，那麼**工作 (TASK)** 便類似於球員所擔任的位置或是球場上的陣型，而決定著球員之間的配合模式。

5.3.1 ISPSOft 中的工作 (TASK) 管理

在 ISPSOft 中，一個專案中的**程式 POU** 並非都會參與實際的執行，而是必須在指派 TASK 之後才可確定該 POU 是否執行以及如何執行；當 POU 未被指派 TASK 時，該 POU 僅會被當做一般的原始碼儲存在專案中，本身並不會被編譯為 PLC 的執行碼。只有**程式 POU** 需要被指派 TASK，至於**功能塊 (FB)** 的執行方式，則是依據呼叫它們的上位 POU 程式而定。

IEC61131-3 程式架構的最大特色便是將程式切割成若干個獨立開發的 POU，而最後編譯時則會將所有 POU 程式重新排列組合為一個可逐步掃描的執行碼，其排列組合的依據便是 TASK 的配置狀態。

在 ISPSOft 中，一個**程式 POU** 只能指派給一個 TASK，只有 AS5xx 與 DVPxxMC 機種可以一個**程式 POU** 指派給多個 TASK，但同一個 TASK 卻可配置一個以上的 POU，且所配置的 POU 也可在該 TASK 中指定執行的順序；當使用者新增一個**程式類型的 POU** 時，就必須為其指派一個預設的 TASK。

適用機種	AS/AH/DVP	AS5xx/DVPxxMC
工作管理	一個 程式 POU 只能指派給一個 TASK	一個 程式 POU 可以指派給多個 TASK
運行方式	週期性、定時中斷以及條件中斷	自由運行、週期以及事件

AS、AH 以及 DVP 機種的 TASK 運行方式大致可分為週期性、定時中斷以及條件中斷等 3 種。

● 週期性

被指派至週期性 TASK 的程式 POU，其執行方式便是單純的來回掃描。在 DVP 系列的機種中，因其系統設置只有一個週期性的 TASK，因此所有週期性掃描的 POU 皆須被分配至此 TASK；而 AH 系列的機種則有 32 個週期性的 TASK 可供選擇，其編號為 0~31，掃描順序則以編號小的為優先，且 AH 系列機種尚有支援控制 TASK 的 API 指令 (TKON 與 TKOFF)，使用者可利用該指令於 PLC 的運行過程中任意啟動或暫停某個 TASK 的執行。

以 SFC 語言編輯的 POU 只能被配置於週期性的 TASK 中，而不可配置於中斷型的 TASK。

● 定時中斷

被指派至定時中斷 TASK 的程式 POU，其功能便類似定時中斷副程式，當定時中斷的時間到達後，被分配至該 TASK 的所有 POU 便會依照排列順序執行一遍。定時中斷 TASK 的數量會依據選用的 PLC 機種而定，該機種支援幾組定時中斷源，系統便會有幾組定時中斷的 TASK。

● 條件中斷

條件中斷亦可區分為多種類型，如外部中斷、I/O 中斷、計數中斷...等，而不同機種所提供的中斷服務與數量皆會有所不同，因此規劃前必須先確認所選用的專案機種支援哪些中斷模式。當程式 POU 被分配至條件中斷 TASK 時，其功能便類似中斷副程式，一旦中斷的條件成立，如計數到達設定值，則被分配至該 TASK 的所有 POU 便會依照排列順序執行一遍。

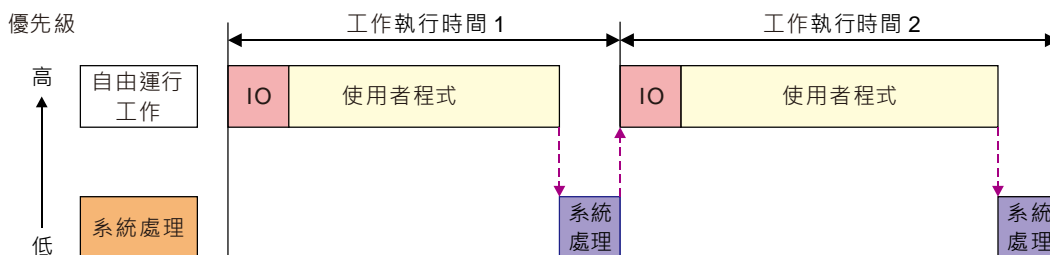
針對 AS5xx 與 DVPxxMC 機種，一個程式 POU 可以指派給多個 TASK，且同一個程式 POU 可以重複指派給同一個 Task，而一個 TASK 可配置一個以上的 POU，且所配置的 POU 也可在該 TASK 中指定執行的順序，而 AS5xx 與 DVPxxMC 機種 TASK 運行方式，分為自由運行、週期以及事件 3 種，接下來就 TASK 運行方式及 TASK 執行優先順序做介紹。

● TASK 運行方式

(1) 自由運行

程式一開始運行工作 (TASK) 就被處理，一個運行週期結束後，工作 (TASK) 將在下一個週期中被重新啟動。

➤自由運行工作 (TASK) 執行方式如下：

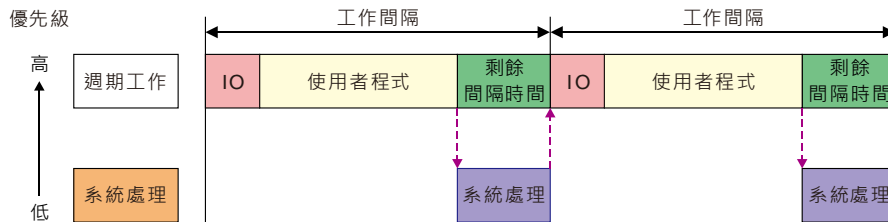


注意：自由運行工作沒有固定持續時間；所以上圖中的任務執行時間 1 和任務執行時間 2 可能不相同。

(2) 週期

按照工作 (TASK) 管理工具內之工作屬性的「間隔時間」所設定之時間執行。

➤週期工作 (TASK) 執行方式如下：



IO：表示 I/O 更新。I/O 包括本機 I/O 點和左右側擴充模組資料、CANopen 資料。此資料可以指定在設定的工作 (Task) 執行前更新；未指定時，此資料在系統處理中更新。

使用者程式：表示使用者程式的執行，按照工作 (Task) 中程式分配循序執行。

剩餘間隔時間：控制器執行系統處理；如果有低優先順序的工作 (Task)，則會先執行低優先順序的工作 (Task)，然後再執行系統處理。

系統處理：控制器執行完所有工作 (Task) 請求後執行系統處理；系統處理包括 Ethernet 通訊、RS232、RS485 通訊處理等。

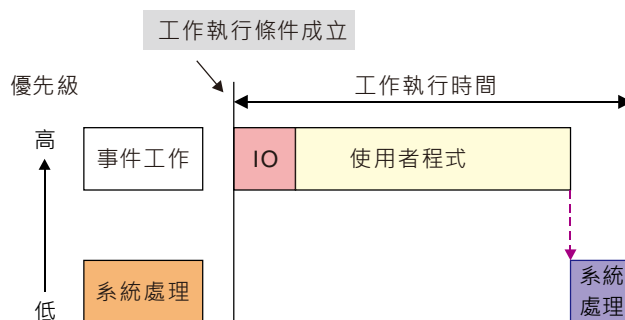
5

注意：如果為某個週期工作 (Task) 定義的週期過短，則該工作 (Task) 會在執行完使用者程式後立即重複執行，而不會執行其它低優先順序的工作 (Task) 或任何系統處理，這將會影響所有工作 (Task) 的執行。如果為該工作 (Task) 設定了程式掃描逾時定時器，則會導致程式掃描逾時定時器逾時，控制器會進入 Error 狀態，並且停止執行使用者程式；如果沒有為工作 (Task) 設定程式掃描逾時定時器，控制器會無法執行系統處理，出現通訊逾時等問題。

(3) 事件

事件工作 (Task) 是指僅在指定的事件發生時執行一次。事件工作 (Task) 的執行時機取決於事件發生的時機和事件工作 (Task) 的優先順序。

➤事件工作 (TASK) 執行方式如下：



➤可供選擇的事件包含以下幾種觸發方式：

- 運動事件 (運動控制工作 (Task));
- 本機輸入點 (I0 ~ I7 · I10 ~ I17) 的上緣或下緣；
- CANopen 網路的同步訊號；
- 增量型編碼器 1 或增量型編碼器 2 的 Z 脈波訊號的上緣；

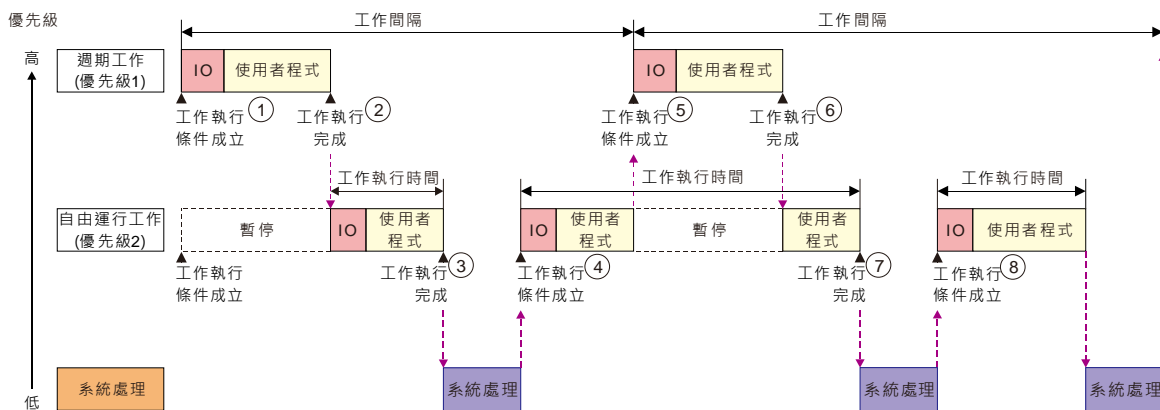
事件工作 (Task) 完成前，該事件工作 (Task) 的執行條件再次成立時，忽略第 2 次的執行條件成立。
事件工作 (Task) 完成前是指正在執行的事件工作 (Task) 或正在等待執行的事件工作 (Task)。

● TASK 執行優先順序

控制器無法同時執行多個工作 (Task)，每個工作 (Task) 須分配一個優先順序，各工作 (Task) 按照設定的優先順序執行。優先順序可以設定為 1 到 24 (1 表示最高優先順序，24 表示最低優先順序)，每個工作 (Task) 的優先順序必須唯一。優先執行優先順序較高的工作 (Task)，高優先順序的工作 (Task) 可以中斷低優先順序的工作 (Task)。

建議將高即時要求的工作 (Task) 設定為高優先順序，將低即時要求的工作 (Task) 設定為低優先順序。在 ISPSOft 中默認建立的運動控制工作 (Task) 預設被配置為優先順序 1。

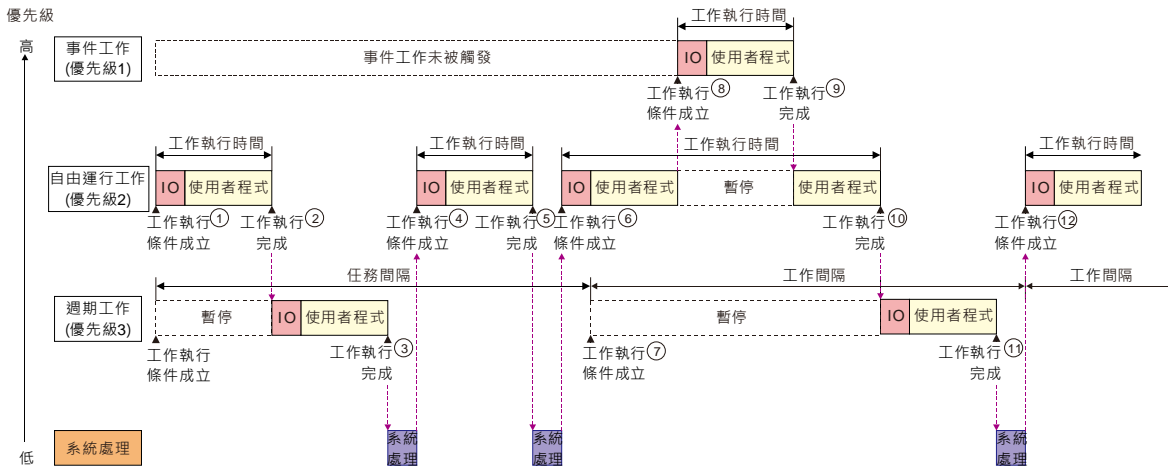
➤兩個工作 (Task) 同時執行時 (週期和自由運行)



- ① 週期工作 (Task) 和自由運行工作 (Task) 執行條件同時成立，因為週期工作 (Task) 的優先順序高，所以先執行週期工作 (Task)；
- ② 週期工作 (Task) 執行完成，自由運行工作 (Task) 開始執行；
- ③ 自由運行工作 (Task) 執行完成，由於沒有其它工作 (Task) 請求，控制器執行系統處理；
- ④ 因為高優先順序的週期工作 (Task) 請求還沒有到來，所以繼續執行自由運行工作 (Task)；
- ⑤ 自由運行工作 (Task) 執行過程中，高優先順序的週期工作 (Task) 請求到來，週期工作 (Task) 打斷自由運行工作 (Task)，控制器執行週期工作 (Task)；

- ⑥ 週期工作 (Task) 執行完成，控制器繼續執行低優先順序的自由運行工作 (Task) 未執行的部分；
- ⑦ 自由運行工作 (Task) 執行完成，由於沒有其它的工作 (Task) 請求，控制器執行系統處理；
- ⑧ 系統處理完成，因為沒有高優先順序的週期工作 (Task) 請求，所以繼續執行自由運行工作 (Task)。

➤三個工作 (Task) 混合執行時 (事件、自由運行、週期)



- ① 自由運行工作 (Task) 和週期工作 (Task) 執行條件同時成立，因為自由運行工作 (Task) 的優先順序高，所以先執行自由運行工作 (Task)；
- ② 自由運行工作 (Task) 執行完成，週期工作 (Task) 開始執行；
- ③ 週期工作 (Task) 執行完成，由於沒有其它的工作 (Task) 請求，控制器執行系統處理；
- ④ 系統處理完成，執行自由運行工作 (Task)；
- ⑤ 自由運行工作 (Task) 執行完成，由於沒有其它的工作 (Task) 請求，控制器執行系統處理；
- ⑥ 系統處理完成，執行自由運行工作 (Task)；
- ⑦ 週期工作 (Task) 執行條件成立，因為週期工作 (Task) 的優先順序低於自由運行工作 (Task)，所以自由運行工作 (Task) 繼續執行，週期工作 (Task) 等待；
- ⑧ 事件工作 (Task) 執行條件成立，因為事件工作 (Task) 的優先順序最高，所以事件工作 (Task) 中斷自由運行工作 (Task)；
- ⑨ 事件工作 (Task) 執行完畢，控制器繼續執行低優先順序的自由運行工作 (Task) 未執行的部分；
- ⑩ 自由運行工作 (Task) 執行完畢，⑦的週期工作 (Task) 請求還未回應，控制器執行週期工作 (Task)；
- ⑪ 週期工作 (Task) 執行完畢由於沒有其它的工作 (Task) 請求，控制器執行系統處理。

5.3.2 專案管理區的工作 (TASK) 項目

在專案管理區的「工作」項目下會列出所有 POU 的配置狀況，而未被分配任何 POU 的 TASK 項目則不會被列出；每個 TASK 項目下的 POU 排列順序則代表了各 POU 被執行的順序。

以下圖的 AH 機種專案為例，專案中總共建立了 9 個程式 POU，且其 TASK 的配置狀況如下。

工作 (TASK) 名稱	POU	執行方式
未分配	Prog2、Prog3	因未被指派 TASK，因此這兩個 POU 並不會被執行。
週期性 (0)	Prog6、Prog4	因掃描順序會以編號小的週期性 TASK 為優先，因此當程式執行時便會以 Prog6 → Prog4 → Prog5 → Prog7 的順序來回掃描。
週期性 (1)	Prog5、Prog7	
I/O 中斷 (0)	Prog1	當該中斷條件成立時，Prog1 就會被執行一次。
定時中斷 1 (253)	Prog8、Prog9	定時中斷 1 會每隔一段固定時間就被觸發一次，而分配至此 TASK 的 Prog8 → Prog9 也會依序執行一次。

*.關於中斷條件的設定將在第 5.5 節中稍做介紹，而更詳細的說明請參考各機種的使用手冊。

5

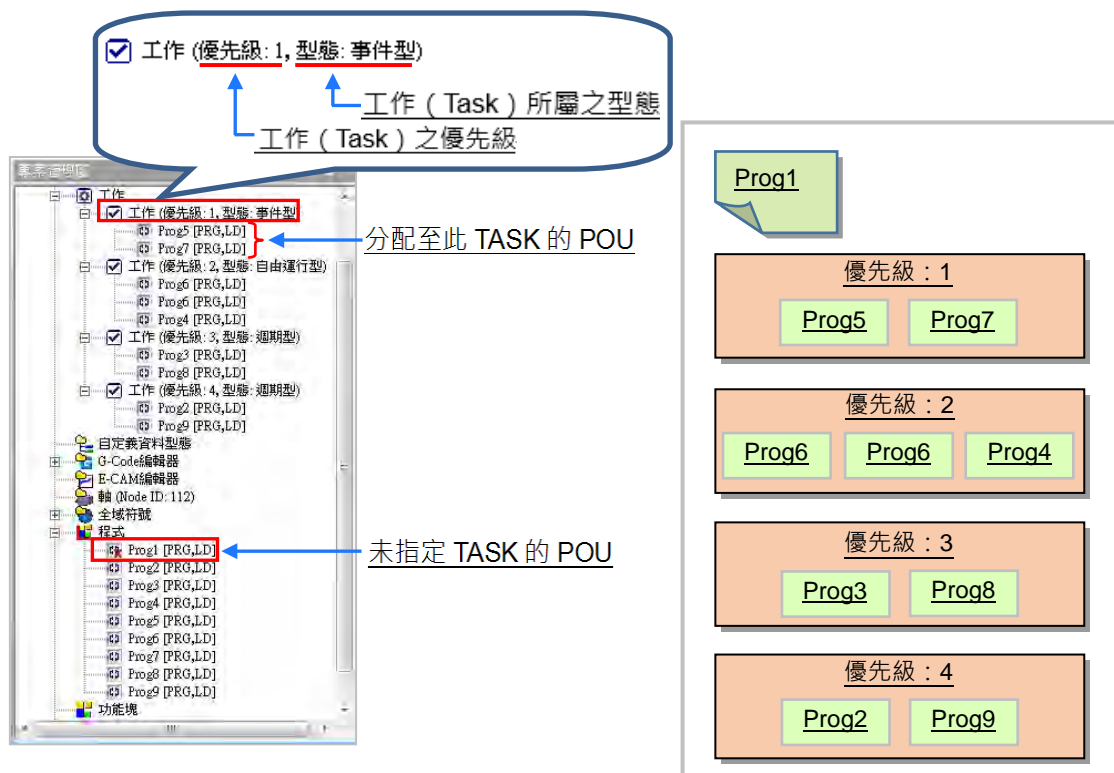
The screenshot shows the '專案管理區' (Project Management Area) interface. On the left, a tree view shows tasks and their assigned POU. Annotations point to: '週期性 (0)' task with Prog6 and Prog4; '週期性 (1)' task with Prog5 and Prog7; 'I/O 中斷 (0)' task with Prog1; '定時中斷 1 (253)' task with Prog8 and Prog9; and '程式' (Programs) section where Prog2 and Prog3 are not assigned to any task. On the right, a summary diagram shows four task boxes: '週期性 (0)' containing Prog6 and Prog4; '週期性 (1)' containing Prog5 and Prog7; 'I/O 中斷 (0)' containing Prog1; and '定時中斷 1 (253)' containing Prog8 and Prog9. Above these are Prog2 and Prog3, which are not part of any task.

欲啟動中斷功能時，仍必須於週期性的程式中插入 API 應用指令「EI」方可正常執行中斷程式。



而 AS5xx 與 DVPxxMC 機種在工作 (TASK) 的名稱顯示上稍有不同，增加了優先級與型態兩種，下面專案中總共建立了 9 個程式 POU，且其 TASK 的配置狀況如下。而工作 (TASK) 優先級順序及工作 (TASK) 型態綜合決定工作 (TASK) 執行的順序。

工作 (TASK) 優先級	工作 (TASK) 型態	POU	執行方式
未分配		Prog1	因未被指派 TASK，因此這個 POU 並不會被執行。
優先級：1	事件型	Prog5、Prog7	當程式執行時便會以 Prog5 → Prog7 的順序來回掃描。
優先級：2	自由運行型	Prog6、Prog6、Prog4	當程式執行時便會以 Prog6 → Prog6 → Prog4 的順序來回掃描。
優先級：3	週期型	Prog3、Prog8	當程式執行時便會以 Prog3 → Prog8 的順序來回掃描。
優先級：4	週期型	Prog2、Prog9	當程式執行時便會以 Prog2 → Prog9 的順序來回掃描。



5

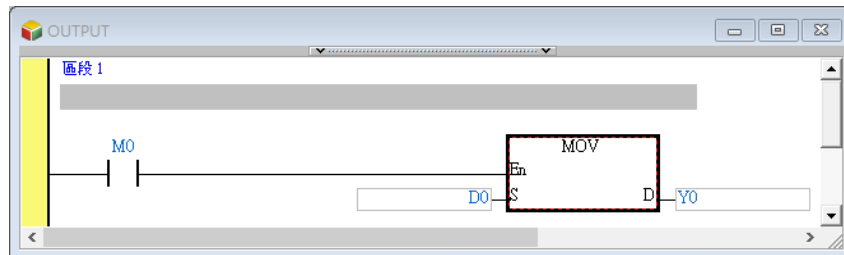
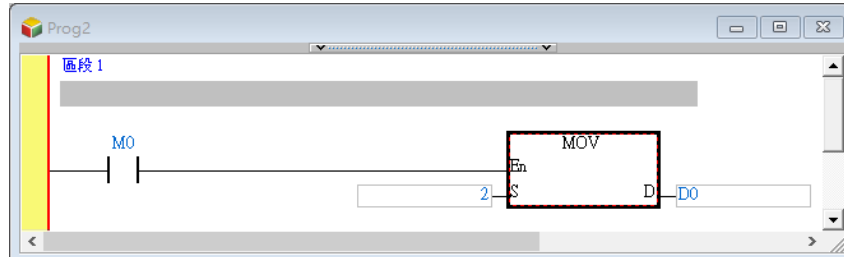
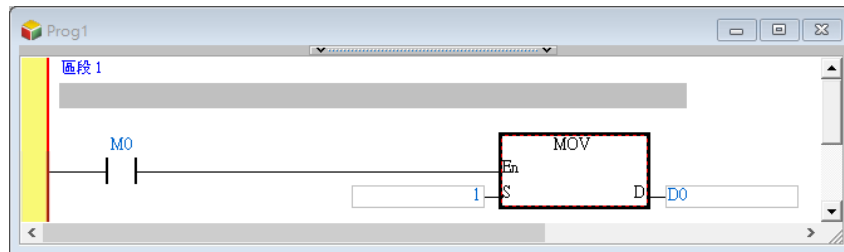
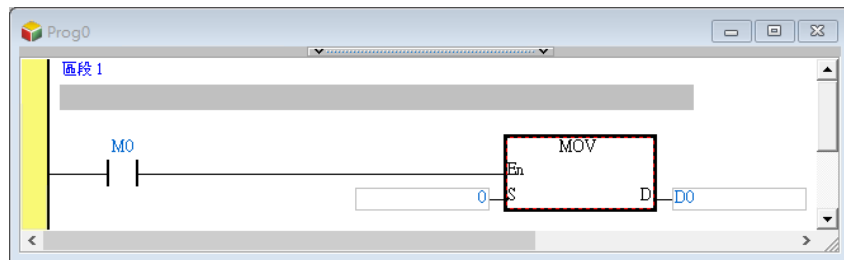
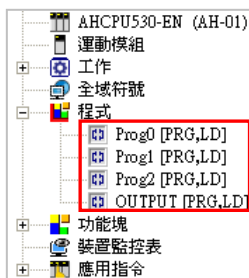
5.3.3 工作 (TASK) 中的 POU 執行順序

同一個工作 (TASK) 項目下可以指定一個以上的 POU，而這些 POU 的排列順序則會決定實際執行時的順序。請參考下列範例，該專案共有 4 個 POU，而各 POU 的功能則簡述於下表。

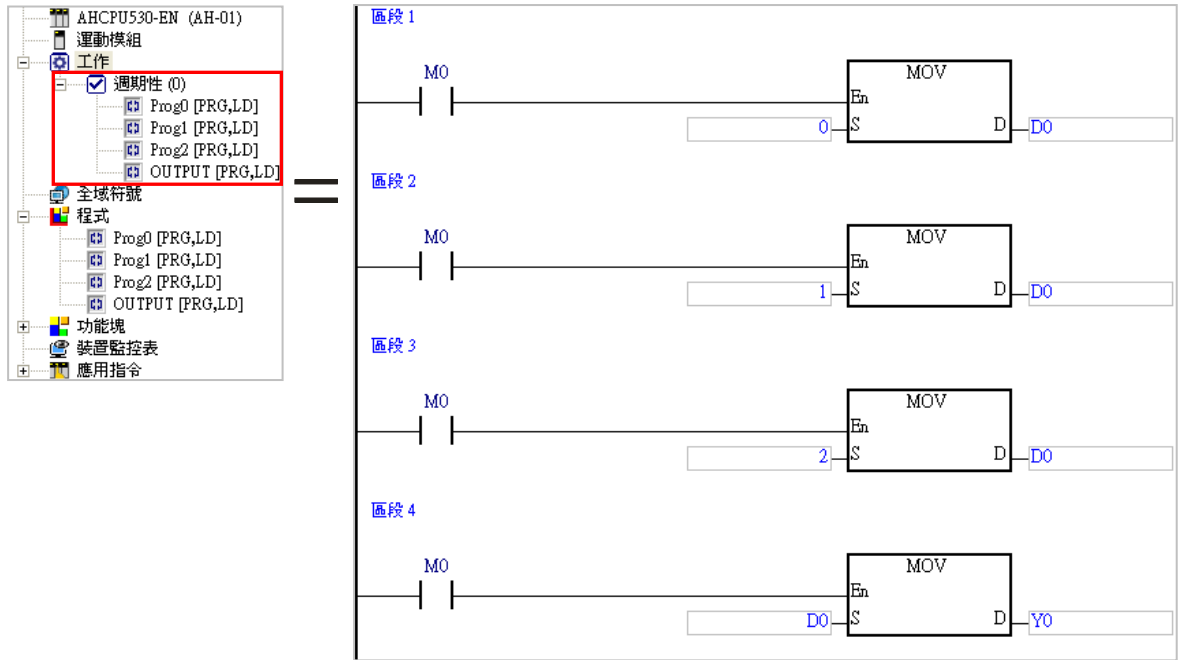
POU 名稱	功能說明
Prog0	當 M0 為 ON 時，將裝置 D0 的值設置為 0。
Prog1	當 M0 為 ON 時，將裝置 D0 的值設置為 1。
Prog2	當 M0 為 ON 時，將裝置 D0 的值設置為 2。
OUTPUT	當 M0 為 ON 時，依據目前 D0 的值決定 Y0 的輸出。

各個 POU 內的程式則請參考下圖。

5

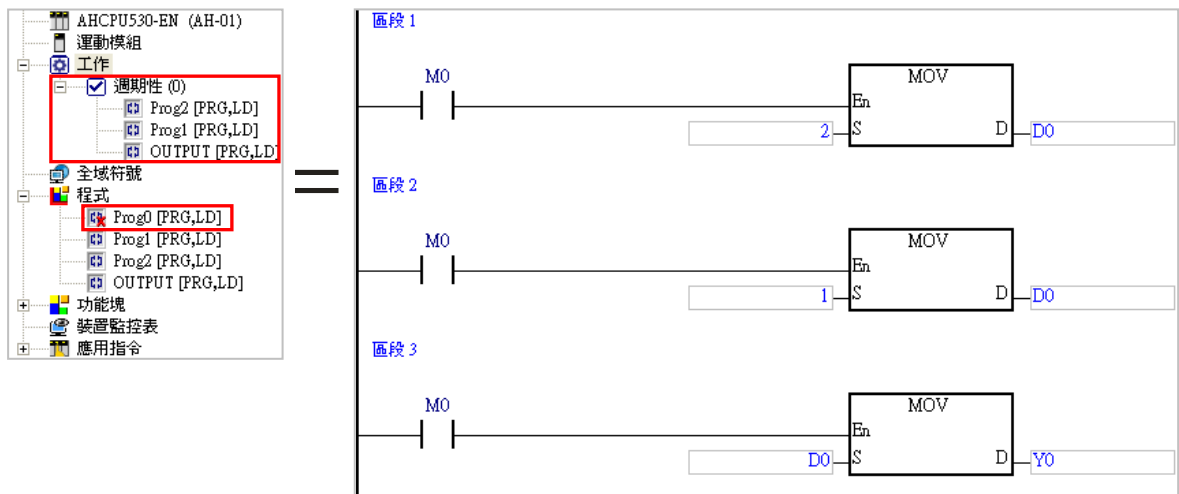


假設 4 個程式 POU 都被指派至週期性 (0) 的工作 (TASK) 中，且各 POU 的排序如下圖所示，則其執行順序便為 Prog0 → Prog1 → Prog2 → OUTPUT；而這也表示執行至「OUTPUT」時的 D0 值將決定於「Prog2」的程序，因此當 M0 為 ON 時，執行至「OUTPUT」時的 D0 值便會等於 2，而將 D0 的值搬移給 Y0 後，Y0 的值也會為 2，而這也代表在一個掃描週期結束後，實際輸出的 Y0 通道中只有 Y0.1 為 ON。下圖即為其等效的程式。



5

但若我們將「Prog2」移至最上方，且將「Prog0」由工作 (TASK) 清單中移除，則「Prog0」中的程式便不會被編譯為執行碼，而執行的順序則會變為 Prog2 → Prog1 → OUTPUT，因此程式執行至「OUTPUT」時的 D0 值將會決定於「Prog1」的程序，而當 M0 為 ON 時，Y0 的值便會等於 1，亦即最後輸出的 Y0 通道中只有 Y0.0 為 ON。



5.4 POU 的管理操作

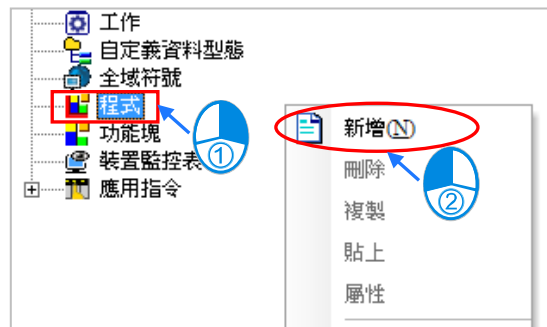
ISPSOft 所支援的 POU 種類分為程式、功能塊及函式，其中函式的用法大致上與功能塊相同可參考功能塊相關介紹。下列表格是各機種所支援的 POU 種類。

	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx/ DVPxxMC
程式	V	V	V	V	V	V	V
功能塊	V	V	V	V	V	V	V
函式							V

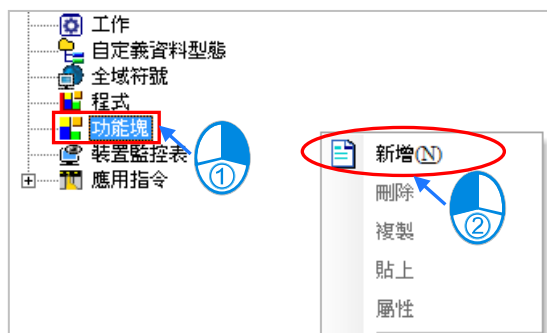
5.4.1 POU 的建立與屬性設定

(1) 於專案管理區中點選要建立的項目，若要建立程式類型的 POU 請點選「程式」，若要建立功能塊類型的 POU 則點選「功能塊」。點選後按下滑鼠右鍵並點選 新增 (N)。

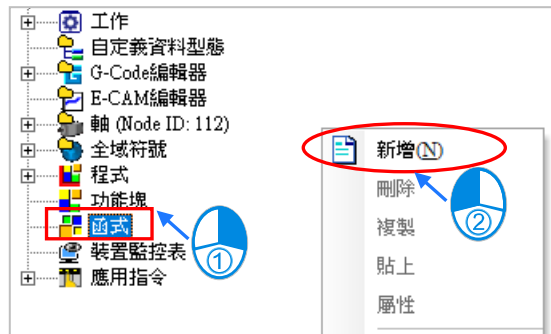
● 新增程式 POU



● 新增功能塊 POU



● 新增函式 POU

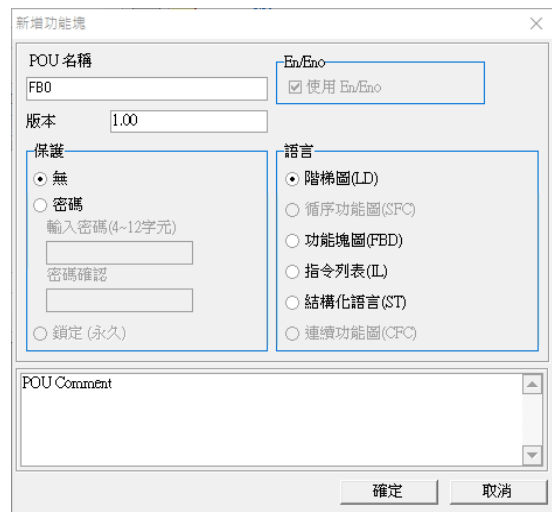


(2) 於隨後出現的視窗中設定該 POU 的屬性，而雖然程式類型與功能塊類型的 POU 屬性有些許差異，但大致上的設定是相同的，在程式 POU 的屬性內，工作 (Task) 欄位之下拉式選單內容隨選擇不同機種專案可能會有所不同。

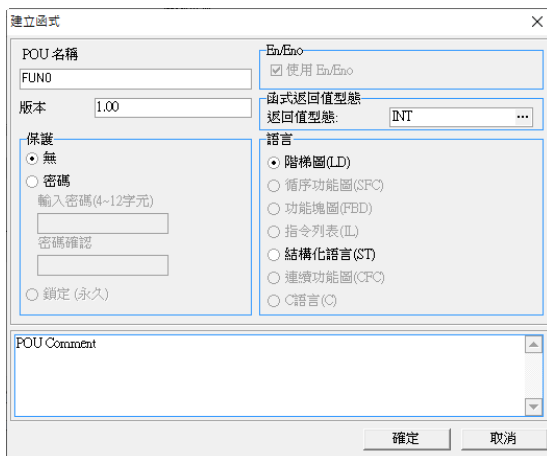
● 程式 POU



● 功能塊 POU



● 函式 POU




5

● **POU 名稱**：於此處輸入 POU 的名稱，而其命名的規則如下。

- (a) 不可使用特殊字元或空白，例如 *、#、?、\、%、@ ...等。
- (b) 至多只可輸入 30 個字元，且須注意一個中文字將會佔用兩個字元。
- (c) 名稱當中可使用底線，但不可連續使用，或是置於結尾，例如 "POU_1" 是合法的名稱，但 "POU__1" 與 "POU_1_" 則是不合法的名稱。

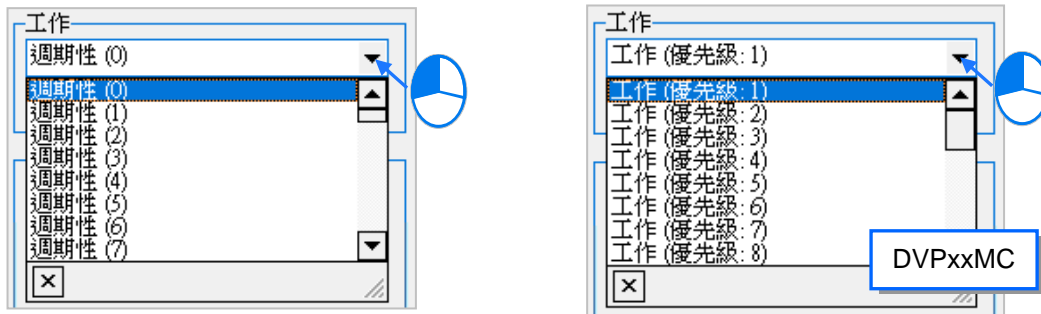


● **函式返回值型態**：建立的為函式類型的 POU，則需要設定函式定義返回值的資料型態，也可通過  按鈕進行選擇，若設為 VOID，表示該函數沒有返回值，即是沒有輸出參數。

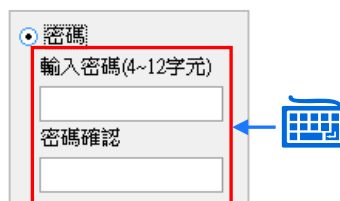
● **工作**：建立的為程式類型的 POU，則可於此處預先指定此 POU 的工作 (TASK) 類型。

下圖左側為一般機種，如果專案機種為 DVPxxMC 或 AS5xx 在顯示的內容會有所不同，如下圖右側。

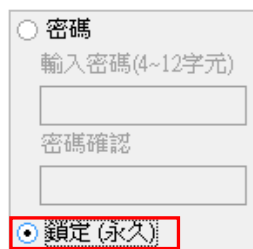
5



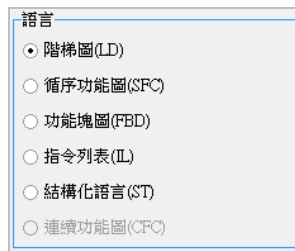
● **密碼**：要設定密碼時，請先後於此二欄位中輸入相同的密碼，輸入的字元可為英數字或符號。



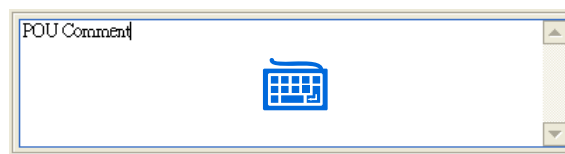
● **鎖定 (永久)**：要將功能塊永久鎖定時，可使用此選項，鎖定後將無法開啟功能塊，建議使用此選項前先將專案做備份；此選項要在新增完功能塊 POU 後，於該 POU 項目上點擊滑鼠右鍵並點選 屬性，才可選擇。



- **語言**：於此選擇開發此 POU 的編程語言（編程語言選擇內容，可能隨機種有所變化）。

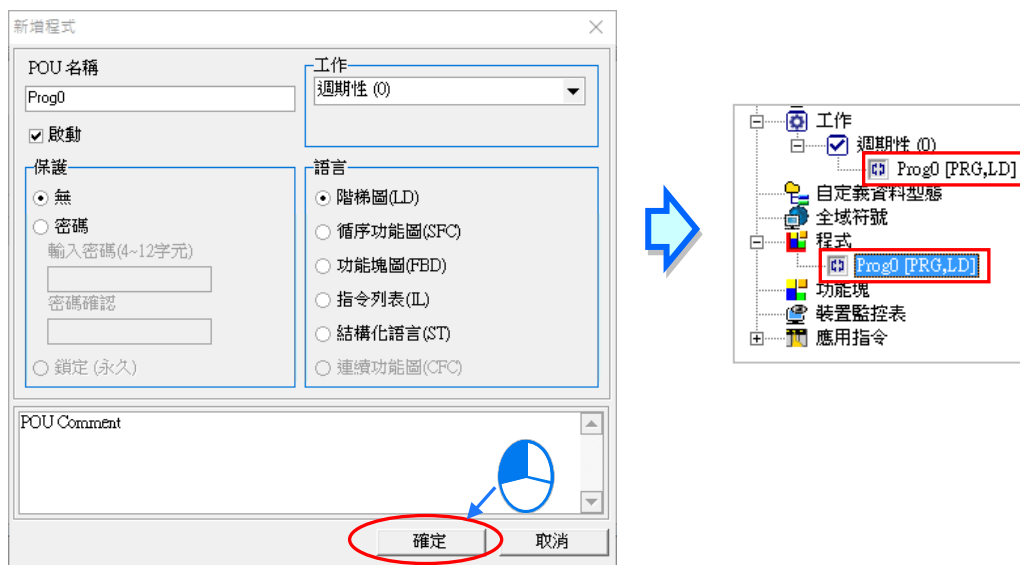


- **註解**：要加入註解時，請直接於下方空白處輸入註解即可。

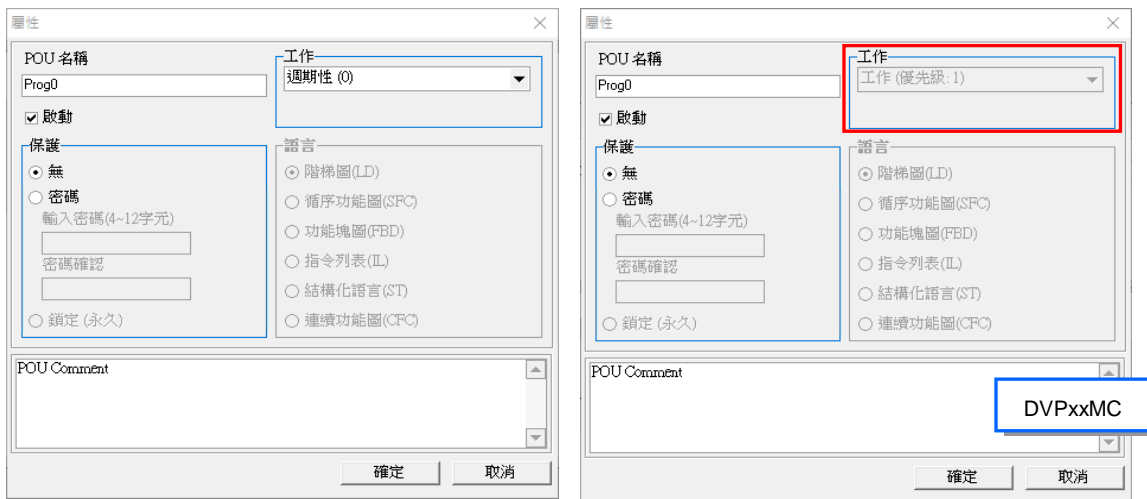
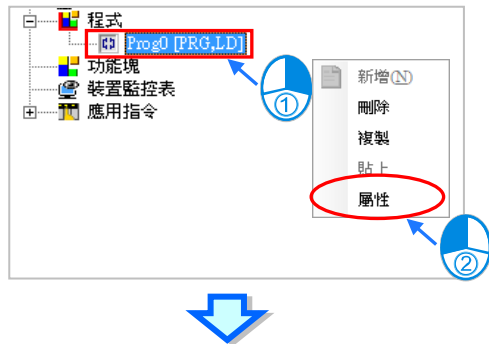


*.若建立的為程式類型的 POU，則在視窗中還會有一個「啟動」的選項，關於此功能將於第 5.4.2 節的內容當中另行介紹。

- (3) 完成設定並按下「確定」後，於專案管理區的「程式」或「功能塊」項目下便會列出該 POU 的項目；而若建立的是程式類型的 POU 時，因已為其指定預設的工作 (TASK) 類型，因此於「工作」項目下亦會將該 POU 的項目列出。



- (4) 欲修改已建立的 POU 屬性時，於該 POU 項目上點擊滑鼠右鍵並點選 **屬性**，之後便可參考前面的敘述進行設定，使用屬性編輯是無法變更編程的語言，如果專案機種為 AS5xx 或 DVPxxMC 機種，如下圖右側使用屬性編輯，是無法從此處變更工作 (Task)。



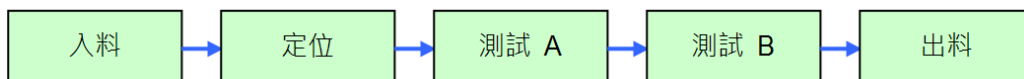
5

5.4.2 POU 的啟動狀態

ISPSOft 提供可將某個 POU 暫時關閉的功能，當 POU 被設為關閉狀態時，該 POU 便會在編譯的過程中被略過，且其內部的程序也不會被執行。不過此功能僅支援程式類型的 POU。

此功能可被用於進行調測或除錯的過程，當在過程中想先略過某些程序，則在操作上便可暫時將該 POU 設為失效狀態，待調測完畢後再將其恢復為啟動狀態即可。

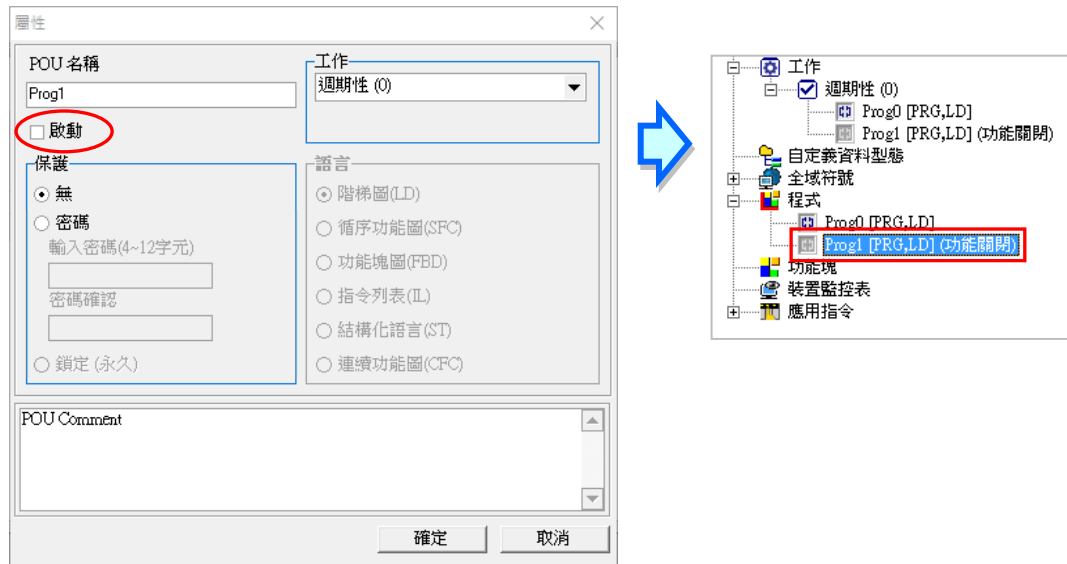
如下圖範例，該設備共有五個工作站，且其工作程序被分為 5 個程式 POU 來進行開發。



假設該設備已組立完畢，但測試儀器尚未完成架設，此時即可將「測試 A」與「測試 B」兩個 POU 設為「未啟動」狀態，如此便可略過此兩站的動作，而僅針對其他的部份進行調測。



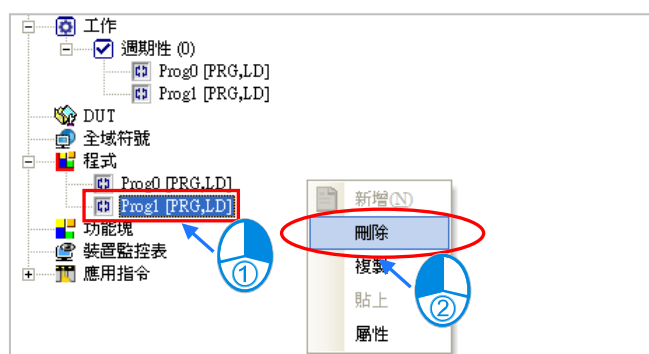
欲設定 POU 的啟動狀態時，可在 POU 屬性視窗中的「啟動」項目進行勾選，取消勾選便代表將該 POU 的功能設為關閉，而處於關閉狀態的 POU，在專案管理區中亦會以不同的圖示來顯示。



- *1. 雖然將 POU 由原本指定的 TASK 清單中移除也可達到關閉該 POU 功能的效果，但若之後要重新執行該 POU 的功能時，除了須再次將其分配至原本的 TASK 之外，還必須對該 TASK 清單中的 POU 重新進行排序，因此針對暫時性的 POU 功能關閉，利用此節介紹的功能來進行操作會相對的較為便利。
- *2. 除了可透過 POU 屬性視窗來設定啟動狀態之外，藉由工作管理工具同樣可進行設定，詳細操作請參考第 5.5 節的介紹。

5.4.3 POU 的刪除與複製

欲刪除 POU 時，請於欲刪除的項目上點擊滑鼠右鍵，接著於快捷選單中點選 刪除。



POU 一旦被刪除之後便無法再還原，確認之後按下「**確定**」鍵即可將所選的 POU 自專案中刪除；而若刪除的是程式類型的 POU，則系統亦會自動將其於工作 (TASK) 清單中剔除。



POU 也支援批次刪除，請於專案管理區的**程式**或**功能塊**項目上點擊滑鼠右鍵，接著於快捷選單中點選「**POU 批次刪除**」。於選擇視窗勾選欲刪除的 POU 後按下「**確定**」按鈕，也可使用「**全部選取**」、「**反向選取**」、「**全部取消**」按鈕。再於確認視窗按下「**確定**」按鈕後完成 POU 的批次刪除。

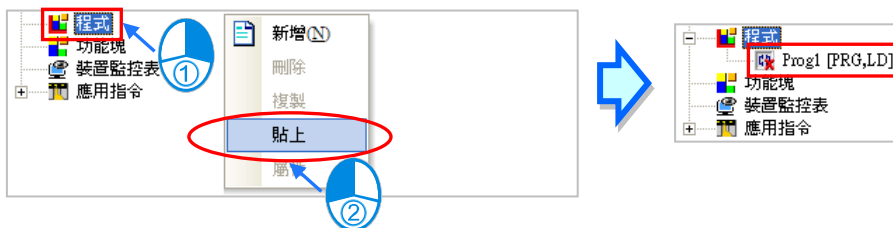
5



欲複製 POU 時，請於欲複製的項目上點擊滑鼠右鍵，並於快捷選單中依次點選 **複製**。



複製的目的位置可為同一個專案或其它專案，但不可將程式類型的 POU 複製至「功能塊」項目下，而功能塊類型的 POU 亦不可複製至「程式」項目下。欲貼上所複製的 POU 時，請於目的位置的「程式」或「功能塊」項目上點擊滑鼠右鍵，並於快捷選單中點選 貼上 即可；而若複製的 POU 為程式類型時，貼上的 POU 將預設為未指派工作 (TASK) 的狀態。

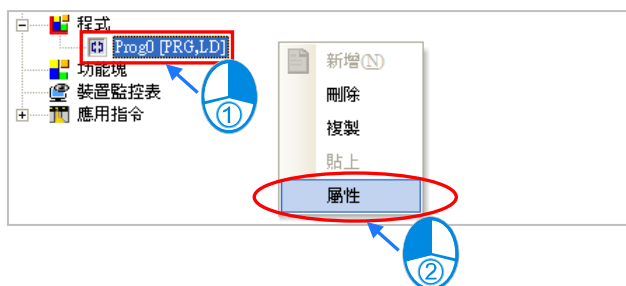


*若貼上的位置已存在相同名稱的 POU 時，系統會自動在貼上的 POU 名稱後面加上 CopyOf_n (n 為流水號) 做為新名稱。

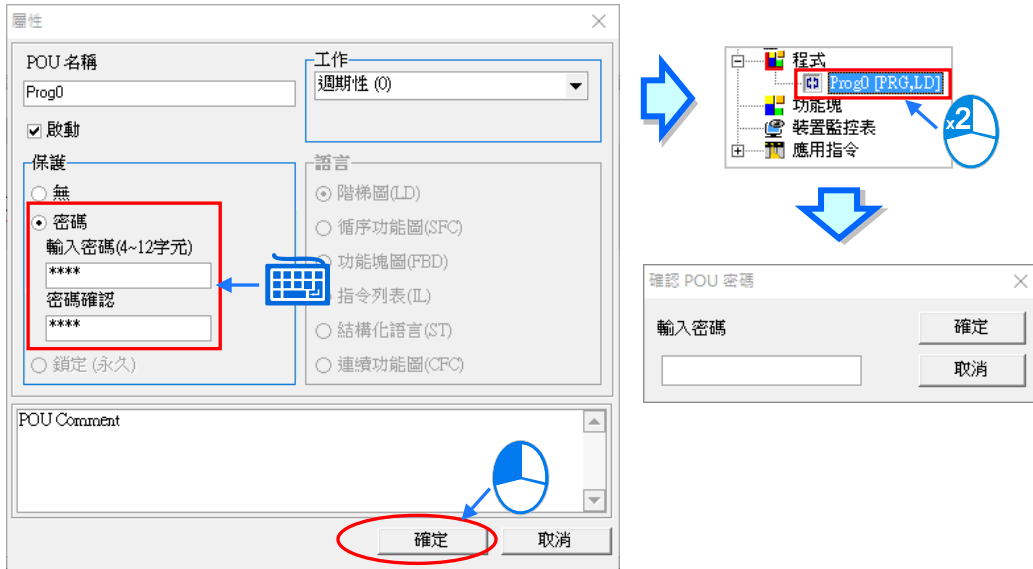
5.4.4 POU 密碼的設定與解除

在 ISPSOft 中支援單獨對各個 POU 設定密碼的功能，而設定與解除的方式如下。

- (1) POU 的密碼可在一開始新增該 POU 時便直接設定，但若一開始並未設定密碼，而是在後續才要加上密碼保護時，請先至專案管理區中，在欲設定密碼的 POU 項目上按下滑鼠右鍵，之後於快捷選單中點選 屬性 以開啟 POU 的屬性視窗。

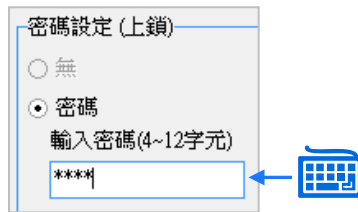


- (2) 在「密碼設定」區先後輸入兩次密碼，兩次輸入的密碼必須一致，而輸入的字元可為英數字或符號，完成之後按下「確定」鍵即可。一旦 POU 加上密碼保護之後，每次開啟該 POU 的視窗時，系統便會要求先輸入密碼以進行確認。



5

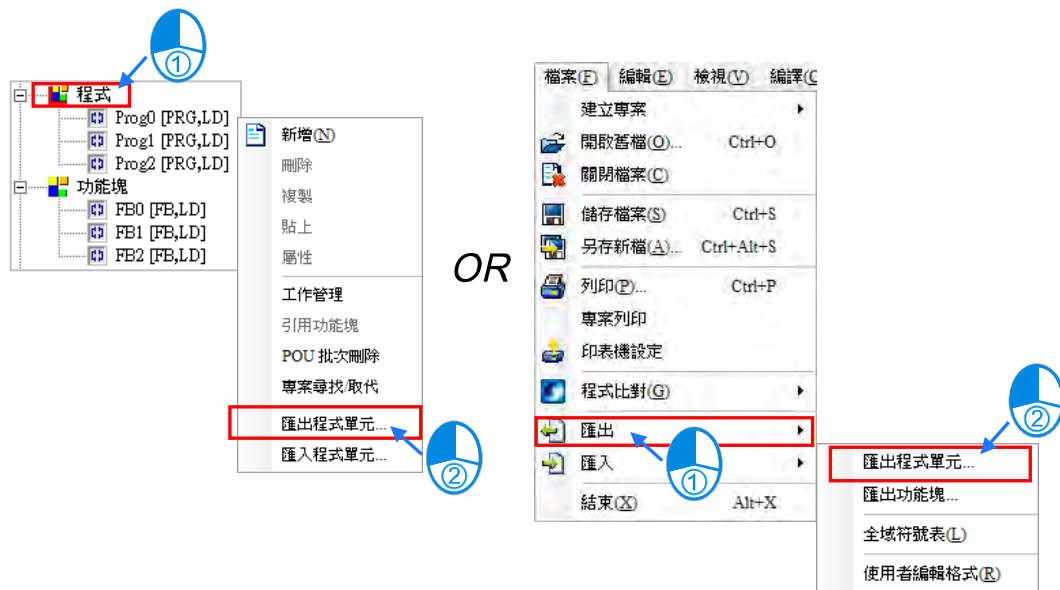
- (3) 欲解除密碼時，請再次開啟 POU 的屬性視窗，之後再於「密碼設定」區輸入正確的密碼，完成後按下「確定」鍵即可。



5.4.5 匯出程式 POU

ISPSOft 支援將 POU 進行「匯出」與「匯入」的功能，不過在群組專案的架構下，每次則只能針對目前正在編輯的專案進行操作。

於功能工具列點選 **檔案 (F) > 匯出 > 匯出程式單元**，之後便可直接對目前正在編輯的專案進行程式單元的匯出操作；另外也可於**專案管理區**中點選目前正在編輯的專案中的**程式**項目，之後按下滑鼠右鍵並於快捷選單中點選 **匯出程式單元**。



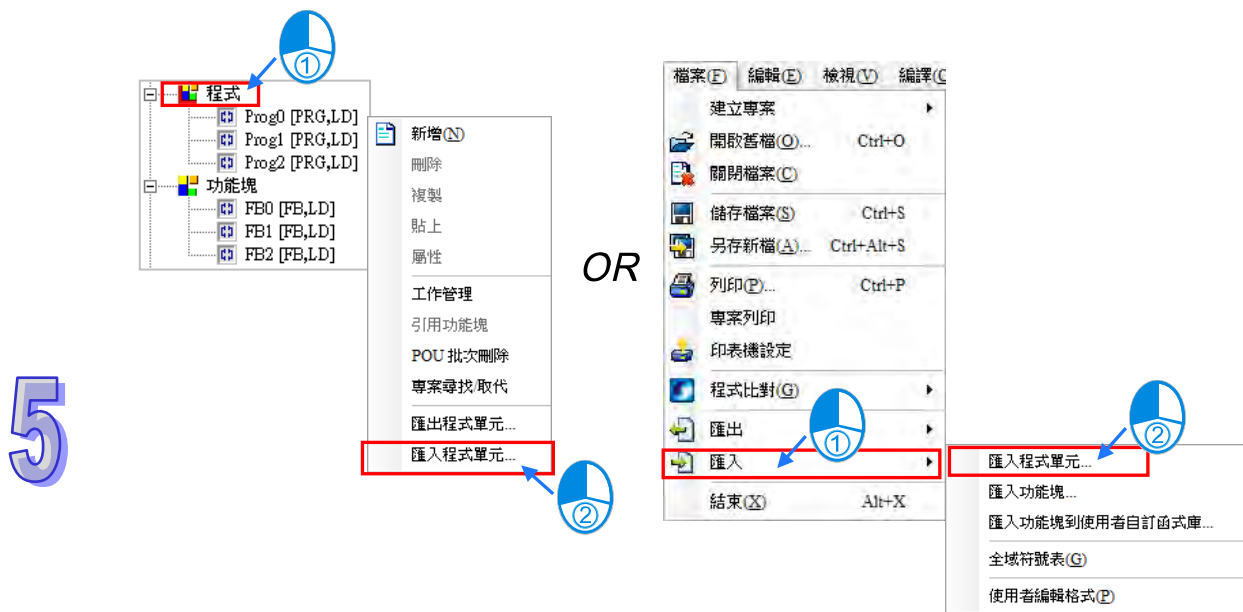
首先於設定視窗中設定匯出的檔名及路徑，並於「程式」清單中勾選欲匯出的程式 POU，而若要為匯出的檔案加上密碼保護時，請先勾選「密碼設定」後，於兩個密碼欄位中輸入密碼，且兩個密碼必須一致以確認輸入無誤，而輸入的字元可為英數字或空白，完成後按下「**確定**」鍵即可。



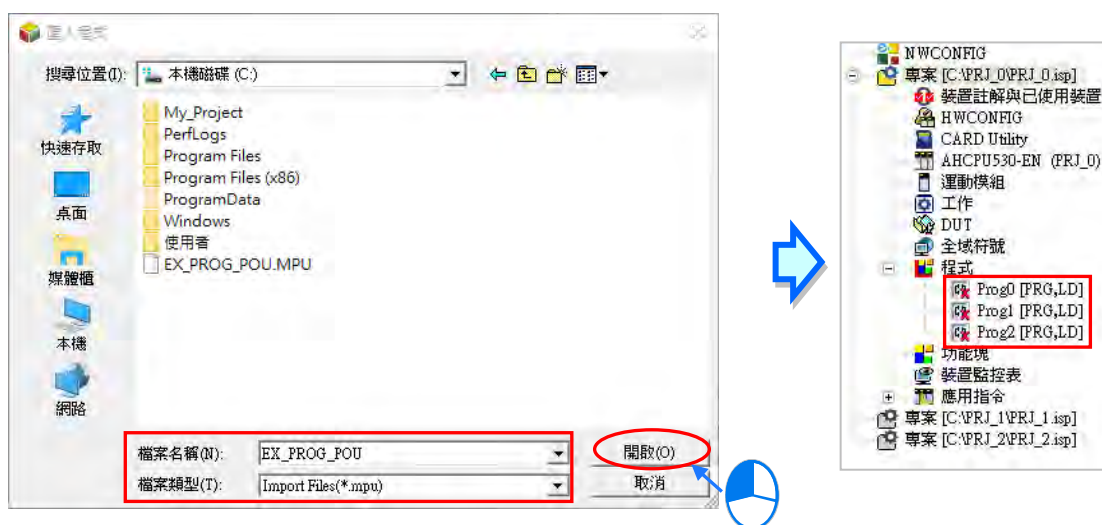
5.4.6 匯入程式 POU

與「匯出」的操作相同，進行「匯入」時亦只能針對目前正在編輯的專案進行操作。執行匯入時，不可將匯出的功能塊 POU，重新匯入至「程式」項目下。

於功能工具列點選 **檔案 (F) > 匯入 > 匯入程式單元**，之後便可直接對目前正在編輯的專案進行程式單元的匯入操作；另外也可於**專案管理區**中點選目前正在編輯專案中的**程式**項目，之後按下滑鼠右鍵並於快捷選單中點選 **匯入程式單元**。



選擇要匯入的檔案後按下「開啟」，而若該檔案有設定密碼保護時，請於出現的視窗中輸入正確的密碼後按下「確定」鍵即可。匯入的 POU 將不會被指派任何的工作 (TASK)，且專案中有相同名稱的 POU 時，系統將不允許重複名稱的 POU 被匯入，而其他的 POU 則可正常匯入。



5.4.7 匯出功能塊 POU

ISPSOft 支援將 POU 進行「匯出」與「匯入」的功能，不過在群組專案的架構下，每次則只能針對目前正在編輯的專案進行操作。

於功能工具列點選 **檔案 (F) > 匯出 > 匯出功能塊**，之後便可直接對目前正在編輯的專案進行功能塊的匯出操作；另外也可於**專案管理區**中點選目前正在編輯的專案中的**功能塊**項目，之後按下滑鼠右鍵並於快捷選單中點選 **匯出功能塊**。



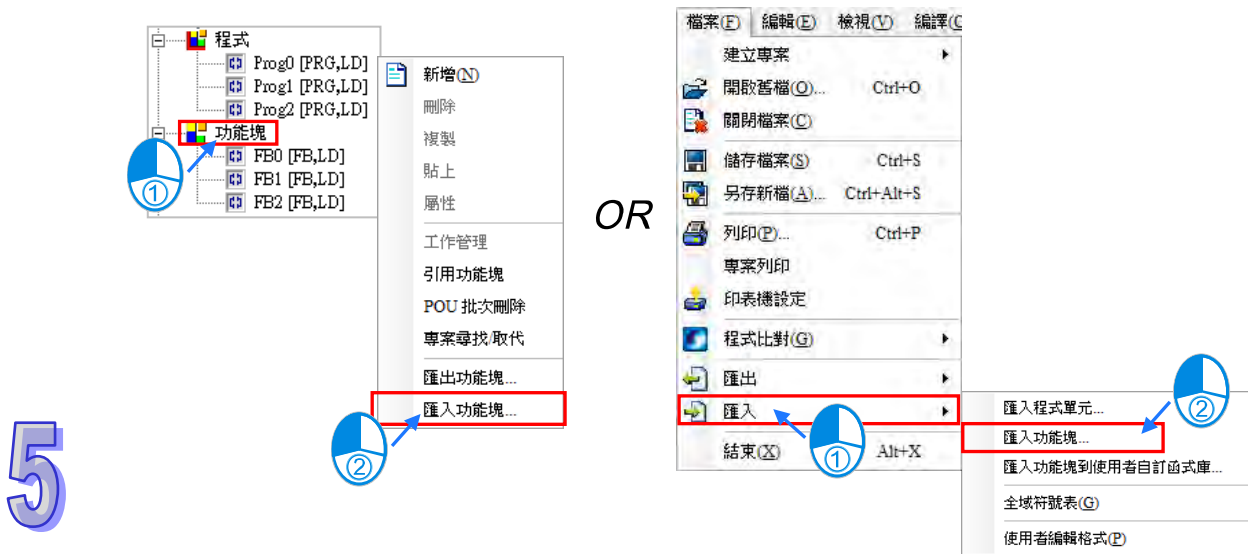
首先於設定視窗中設定匯出的檔名及路徑，並於「功能塊」清單中勾選欲匯出的功能塊 POU，而若需要為匯出的檔案加上密碼保護時，請先勾選「密碼設定」後，於兩個密碼欄位中輸入密碼，且兩個密碼必須一致以確認輸入無誤，而輸入的字元可為英數字或空白，完成後按下「確定」鍵即可。



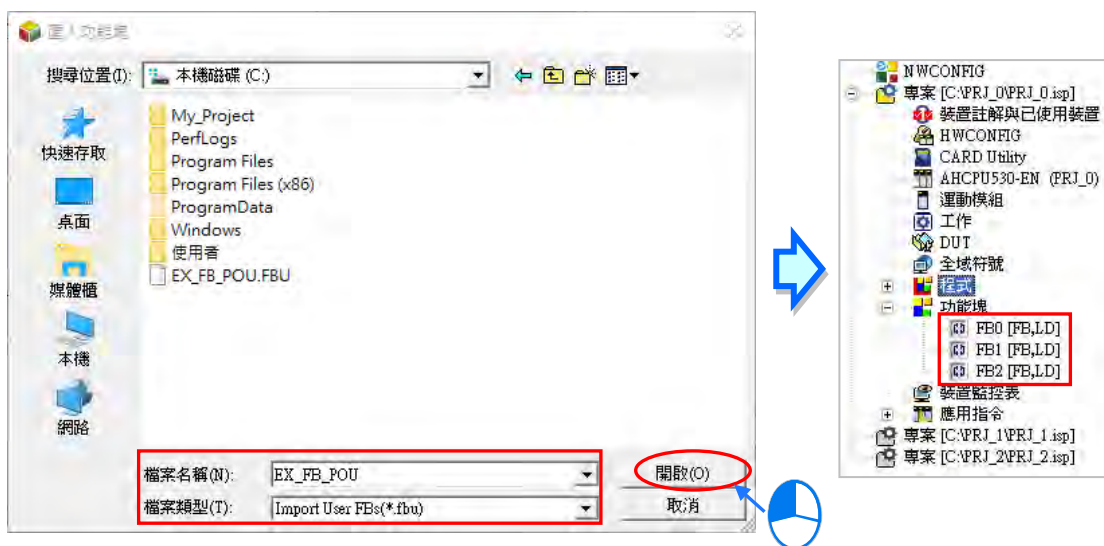
5.4.8 匯入功能塊 POU

與「匯出」操作相同，進行「匯入」時亦只能針對目前正在編輯的專案進行操作。亦不可將匯出的程式單元 POU，重新匯入至「功能塊」項目下。

於功能工具列點選 **檔案 (F) > 匯入 > 匯入功能塊**，之後便可直接對目前正在編輯的專案進行功能塊的匯入操作；另外也可於**專案管理區**中點選目前正在編輯的專案中的**功能塊**項目，之後按下滑鼠右鍵並於快捷選單中點選 **匯入功能塊**。

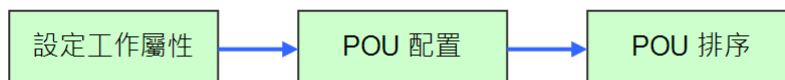


選擇要匯入的檔案後按下「開啟」，而若該檔案有設定密碼保護時，請於出現的視窗中輸入正確的密碼後按下「確定」鍵即可。當專案中有相同名稱的 POU 時，系統將不允許重複名稱的 POU 被匯入，而其他的 POU 則可正常匯入。



5.5 工作 (TASK) 管理的操作

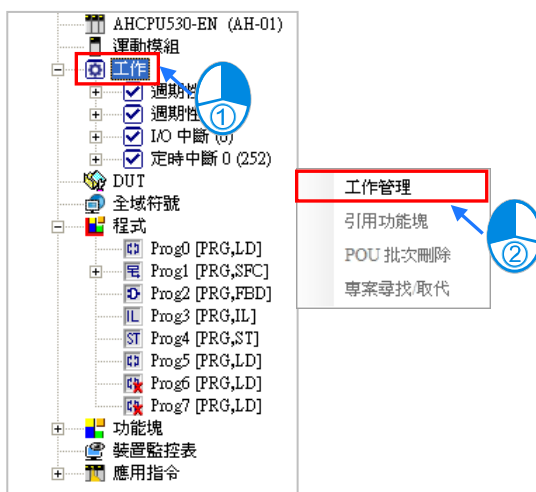
下圖為在 ISPSOft 中進行工作管理的操作步驟，詳細的操作方式將於後續的幾個小節當中逐一介紹。



進行工作管理之前必須先開啟 ISPSOft 中的**工作管理工具**，而其開啟的方式有以下三種。

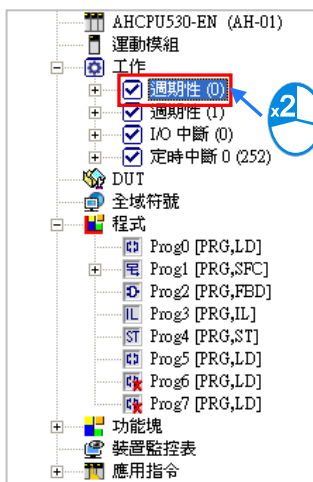
● 方法一

於**專案管理區**的「**工作**」項目上點擊滑鼠右鍵，接著於快捷選單中點選「**工作管理**」，而之後系統便會直接開啟管理工具。



● 方法二

展開**專案管理區**的「**工作**」項目，並於要設定的工作 (TASK) 項目上雙擊滑鼠左鍵，而之後系統便會開啟管理工具，並直接顯示該 TASK 的設定頁面。



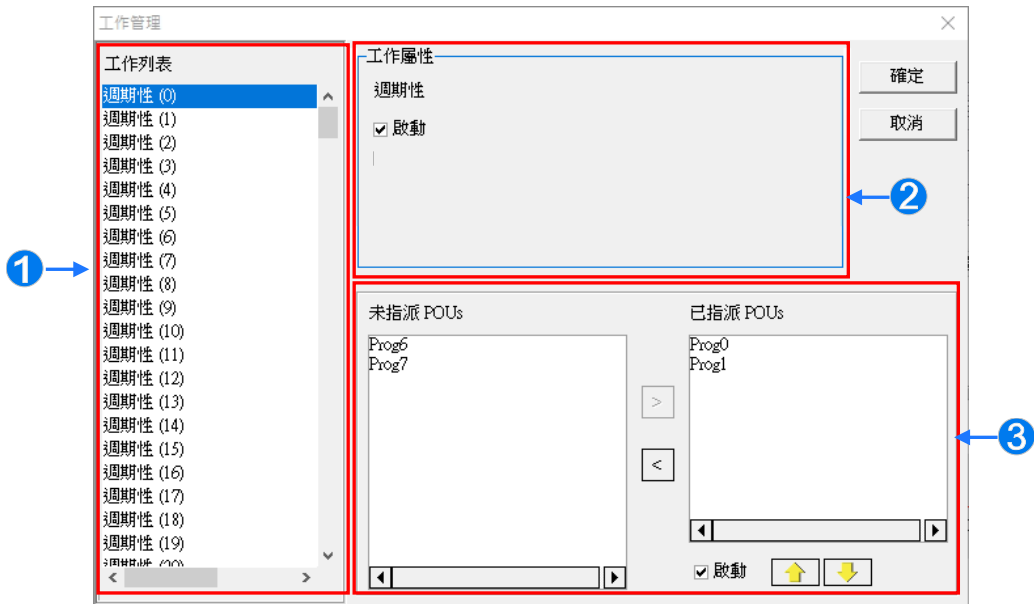
● 方法三

於專案管理區點選「程式」項目中，接著點擊滑鼠右鍵並於快捷選單中點選「工作管理」，而之後系統便會自動開啟管理工具。



5

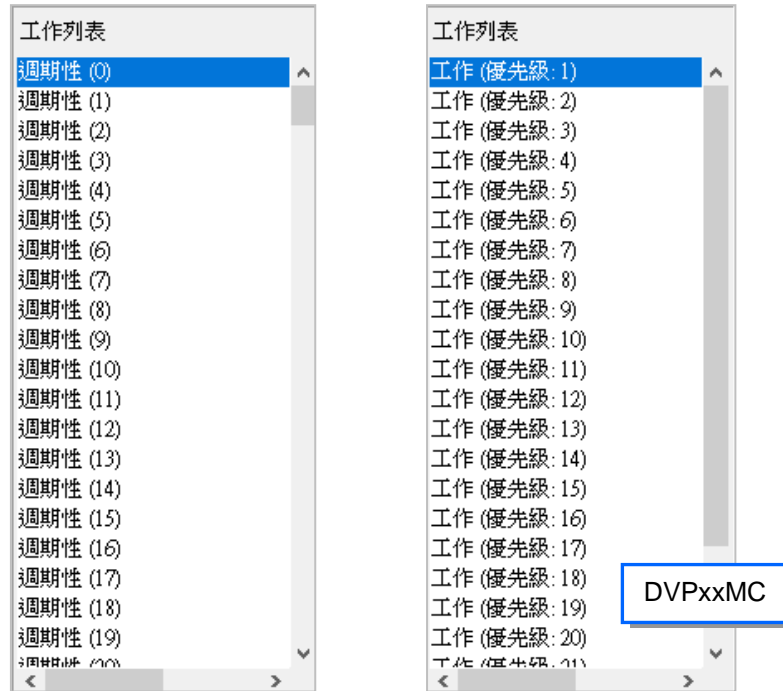
下圖便是工作 (TASK) 管理工具的操作介面。



- ① 工作 (TASK) 列表：此處會列出所有可規劃的工作 (TASK) 項目。
- ② 工作 (TASK) 屬性：此處會顯示所選 (TASK) 的屬性設定及說明。
- ③ 工作 (TASK) 管理區：此處為對 POU 進行配置與排序的操作區。

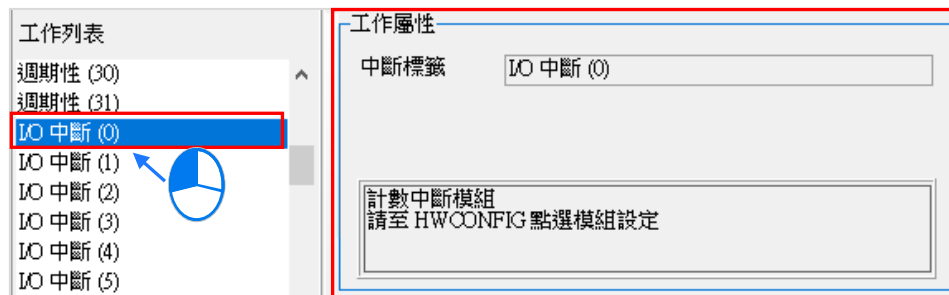
5.5.1 工作 (TASK) 屬性與中斷條件的設定

在「工作列表」中的 (TASK) 項目，會因為不同機種，所提供的 TASK 項目而有所不同，下圖左側為一般機種；若為 DVPxxMC 或 AS5xx 機種，則顯示為下圖右側。

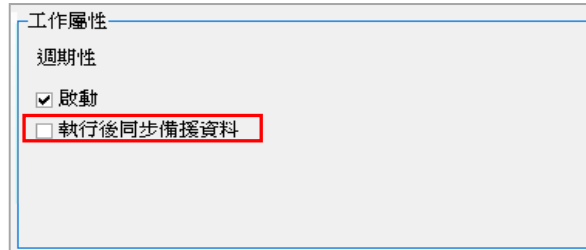


5

於工作列表中選定工作 (TASK) 項目後，「工作屬性」的區域便會顯示對應該 TASK 的設定與提示，而此處的功能主要在於設定該 TASK 的執行條件；專案所提供的 TASK 項目與所選機種的功能特性有關，因此每個 TASK 的屬性內容都會有所差異，不過基本的設定方式都是相同的。




而「工作屬性」的區域所顯示的項目，也會隨著專案機種而有所不同，如下圖，如果專案機種選取 AH560 備援機種，才會有「執行後同步備援資料」的選項，而這個功能的相關使用內容，可以參閱該產品相關手冊。



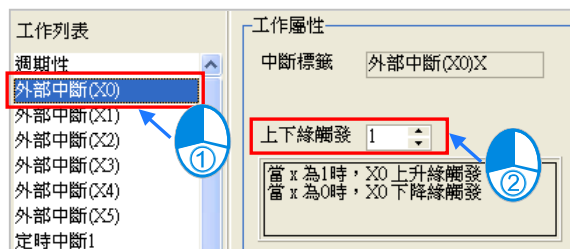
先以幾個簡單的例子來說明如何對工作屬性進行設定，而其他工作屬性的設定方式則請參考下述的範例，再配合對應機種的操作手冊或相關說明文件來進行設定。

EX1：假設要設定 SV 機種專案的中斷 TASK 屬性，首先必須先確認 SV 機種所提供的中斷指標，而下表即為 **DVP-PLC 應用技術手冊 (程式篇)** 中註明的 SV 機種中斷指標。

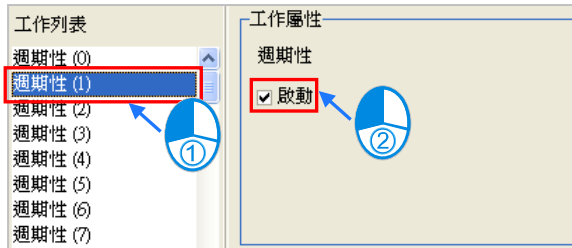
外部中斷插入	I00□(X0) , I10□(X1) , I20□(X2) , I30□(X3) I40□(X4) , I50□(X5) , 6 點 (□=1, 上升緣觸發  , □=0, 下降緣觸發 )
定時中斷插入	I6□□ , I7□□ , 2 點 (□□=1~99, 時基=1ms) I8□□ , 1 點 (□□=1~99, 時基=0.1ms)
高速計數到達中斷插入	I010、I020、I030、I040、I050、I060 , 6 點
脈波中斷插入	I110、I120、I130、I140 , 4 點
通訊中斷插入	I150、I160、I170 , 3 點

5

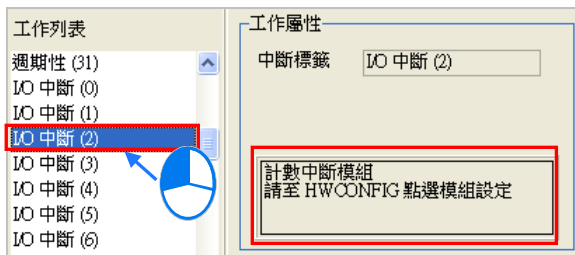
下圖則為 SV 專案的 TASK 列表。若要設定與中斷指標 I001 (即 X0 的上升緣觸發中斷) 對應的 TASK 時，請先於工作列表中選擇「外部中斷 (X0)」，之後於右側的屬性設定區域便會顯示與其相關的設定與提示，而於提示的內容可得知，當上下緣觸發的欄位設定為 1 時便代表為上升緣觸發，因此接著只要再將「上下緣觸發」的欄位設定為 1 即可。



EX2：下圖為 AHCPU530-EN 機種的 TASK 屬性設定畫面，且點選的 TASK 頁面為週期性 (1)；而因 AH 系列機種可透過 **TKON** 與 **TKOFF** 的 API 應用指令來啟動或關閉週期性 **TASK** 的功能，因此於此處可設定該 **TASK** 的初始狀態；當「啟動」項目未被勾選時，該 **TASK** 於程式執行時便會先處於關閉的狀態，而之後便必須等待其他 **TASK** 中的 **POU** 對其使用 **TKON** 指令後，該 **TASK** 才會開始執行。關於 **TKON** 與 **TKOFF** 指令的相關說明請參考相關程式手冊。



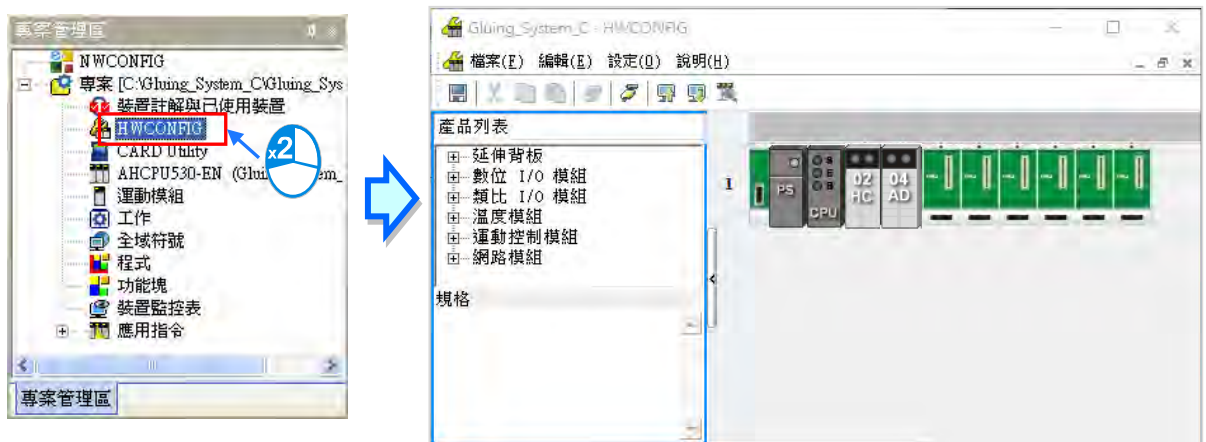
EX3：下圖為 AHCPU530-EN 機種的 TASK 屬性設定畫面，且點選的 TASK 頁面為 I/O 中斷 (2)；而因 AH 機種的 I/O 中斷條件可由使用者自行規劃，因此提示文字中便會註明至 **HWCONFIG** 中進行設定。下列我們將繼續示範如何於 **HWCONFIG** 中規劃中斷條件。



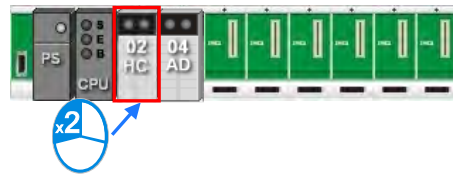
5

此處我們假設指定 **I/O 中斷 (2)** 的中斷條件是要來自計數器模組 AH02HC-5A 的通道 0，而下列則為透過 **HWCONFIG** 設定 IO 中斷條件的範例，更詳細的操作步驟請參考手冊第 3 章的說明內容，而關於 AH 系列機種的中斷功能，則請參考 AH 機種的操作手冊或相關的技術文件。

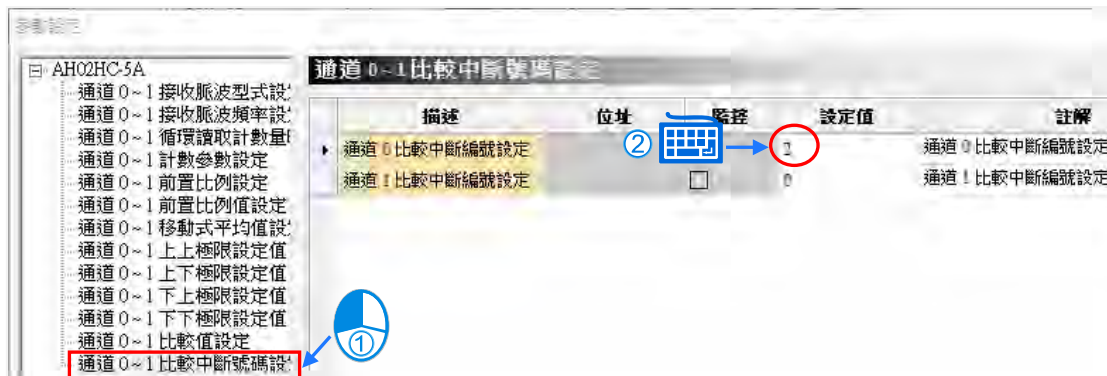
- (1) 開啟 ISPSOft 中的 **HWCONFIG**。



- (2) 於欲產生中斷信號的計數器模組上雙擊滑鼠左鍵，接著便會開啟模組參數的設定視窗。

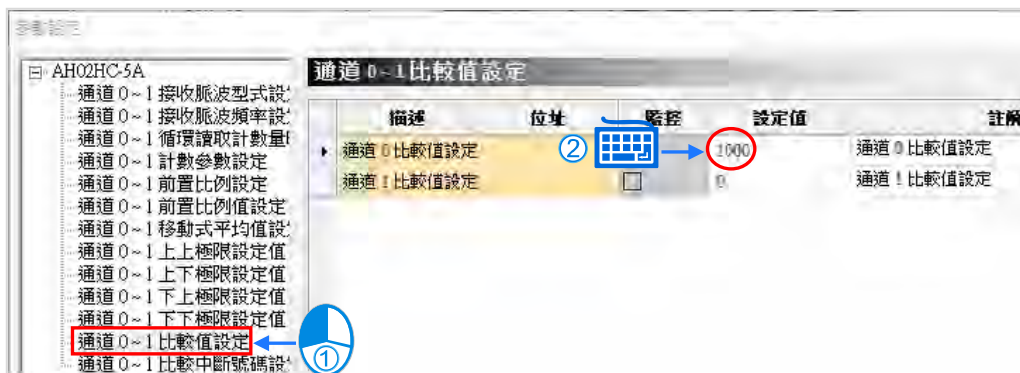


- (3) 開啟模組設定視窗後，請先切換到中斷編號的設定頁面，並將「通道 0 比較中斷編號設定」中的「初始值」欄位設定為預計的中斷編號 2。



*.通道 0 與通道 1 的中斷編號可相同，而這樣的設定代表當有任何一個通道的條件成立時都會觸發同一個 I/O 中斷。

- (4) 接著切換至比較值的設定頁面，並於「通道 0 比較值設定」的「初始值」欄位中設定觸發中斷條件的計數值。



- (5) 而完成設定後請務必記得存檔，並在開始執行前將所設定的參數下載至主機中才可生效。

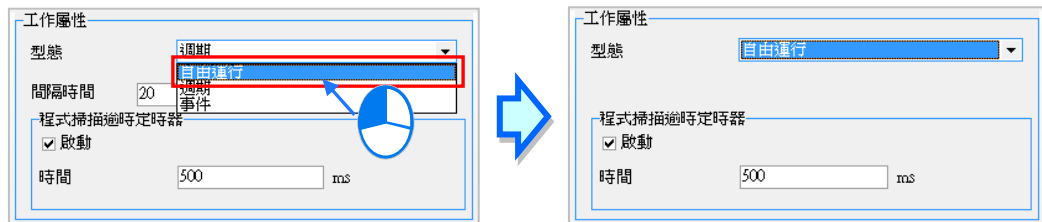


若專案使用 DVPxxMC 或 AS5xx 機種，那麼工作列表的選項會有，優先級 1~24(1 代表最高優先級，24 代表最低優先級)。選取完優先級之後，可以在工作屬性項目裡，選擇適當的「型態」及設定「程式掃描逾時定時器」。接下來就「型態」以及「程式掃描逾時定時器」來作介紹。

● 型態

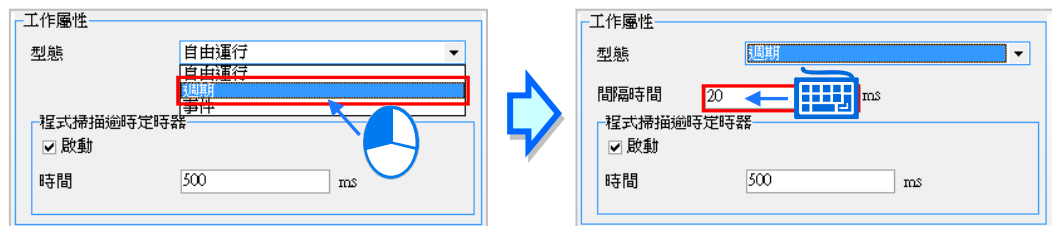
使用者可選擇的「型態」，共可分為三種型態，分別為「自由運行」、「週期」以及「事件」。

(1) 自由運行



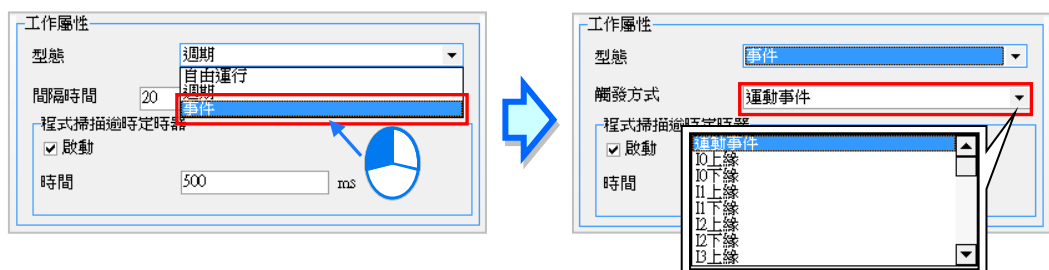
於「型態」欄位下拉式選單中，選擇「自由運行」，此項目為程式一開始運行工作 (TASK) 就被處理，一個運行週期結束後，工作 (TASK) 將在下一個週期中被重新啟動。

(2) 週期



於「型態」欄位下拉式選單中，選擇「週期」，並可於「間隔時間」欄位輸入欲設定之時間；週期工作 (Task) 運作將按照工作 (TASK) 管理工具內工作屬性的「間隔時間」設定之時間週期執行。

(3) 事件



於「型態」欄位下拉式選單中，選擇「事件」，並可於「觸發方式」欄位下拉式選單選擇欲觸發的方式。

➤可供選擇的事件包含以下幾種觸發方式：

- 運動事件 (運動控制工作 (Task));
- 本機輸入點 (I0 ~ I7 , I10 ~ I17) 的上緣或下緣 ;
- CANopen 網路的同步訊號 ;
- 增量型編碼器 1 或增量型編碼器 2 的 Z 脈波訊號的上緣 ;

事件工作 (Task) 是指僅在指定的事件發生時執行一次。事件工作 (Task) 的執行時機取決於事件發的時機和事件工作 (Task) 的優先順序。

● 程式掃描逾時定時器

屬性項目裡的「程式掃描逾時定時器」，用於監控工作 (TASK) 單次執行時間是否超出設定的超時時間，若超出，則會產生相應的報錯資訊。每個工作 (TASK) 均可設置其超時時間以及選擇是否開啟「程式掃描逾時定時器」功能。

使用者若欲設定「程式掃描逾時定時器」可勾選「啟動」項目後，再輸入欲設定之時間。

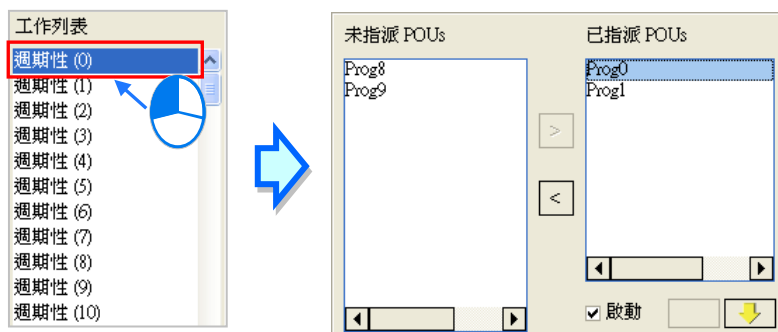
5




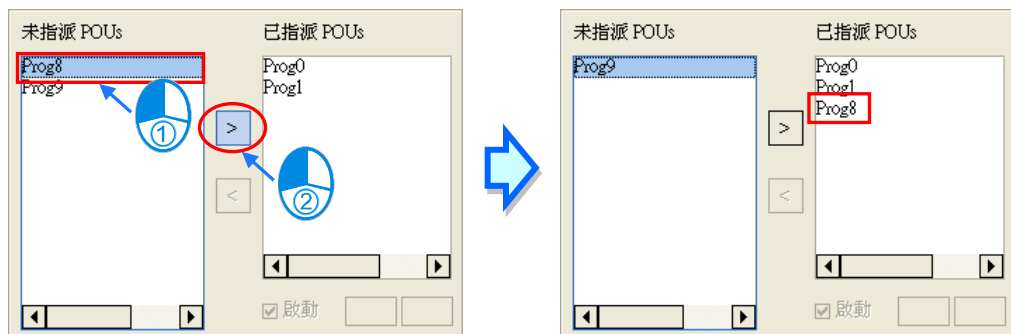
5.5.2 POU 的配置


完成 TASK 的屬性設定後，接下來便要將該 TASK 會執行的 POU 程序配置於該 TASK 的清單中，配置時請注意，一個程式 POU 只能指派給一個 TASK，而且 SFC 編輯的 POU 只能被配置於週期性的 TASK 中。

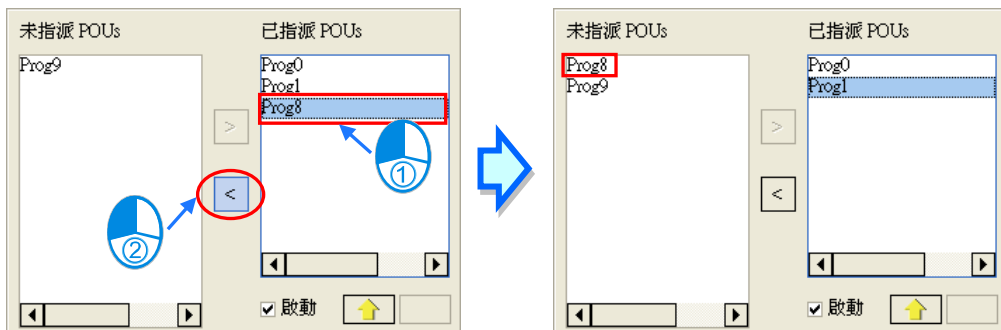
- (1) 先於左側工作列表中點選要設定的 TASK 項目，之後管理區中便會顯示該 TASK 的配置狀況，其中左側為尚未指派 TASK 的 POU，而右側則為已分配給此 TASK 的 POU 項目。



- (2) 於工作 (TASK) 管理區的左側列表中，點選欲分配至此 TASK 項目的 POU 後，按下  圖示即可將該 POU 分配給此 TASK 。

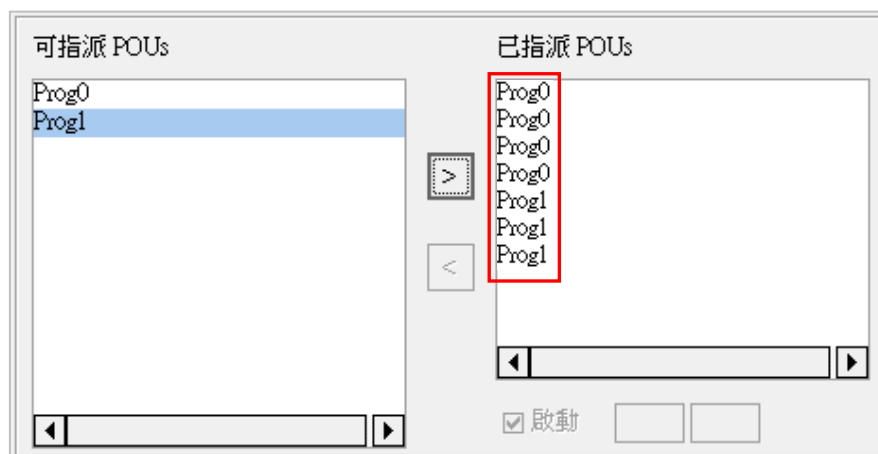


- (3) 於工作 (TASK) 管理區的右側列表中，點選欲取消指派的 POU 後，按下  圖示即可將該 POU 由此 TASK 清單中移除。




5

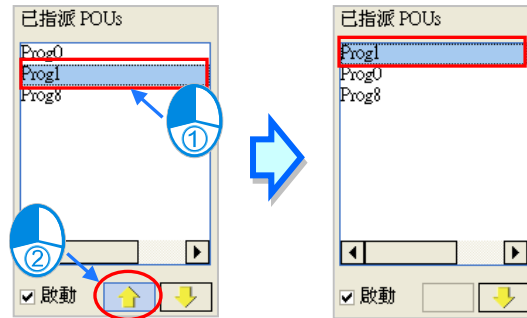
若專案使用 DVPxxMC 或 AS5xx 機種，則同一個 POU 可以重複指派給多個不同的 Task，也可以在同一個 Task 內重複指派多次，如下圖。




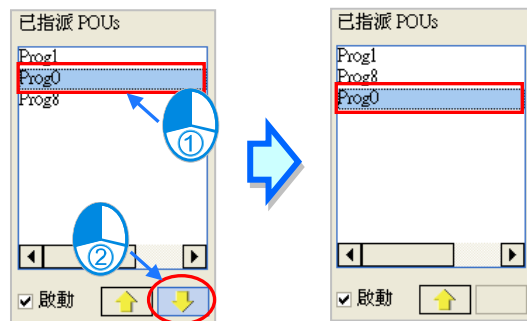
5.5.3 POU 的排序

完成 TASK 清單中的 POU 配置後，接著便要對這些 POU 進行排序，以決定實際執行時的順序。

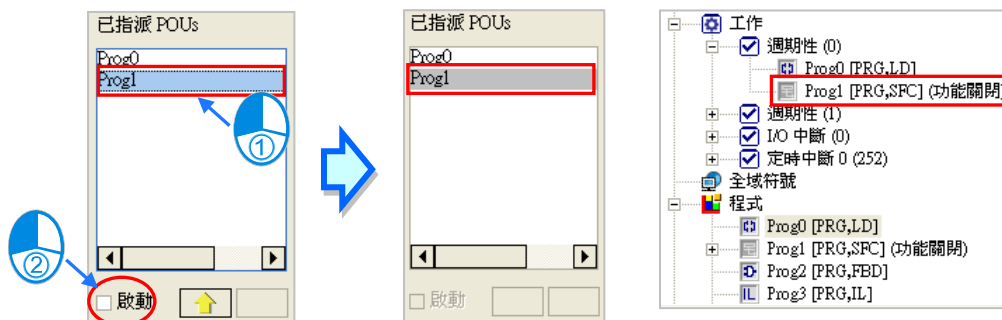
- (1) 於工作 (TASK) 管理區的「已指派 POUs」清單中，先點選欲變更執行順序的 POU 之後，按下  圖示即可將該 POU 的執行順序往上移動一位，代表其執行順序提前。



- (2) 於工作 (TASK) 管理區的「已指派 POUs」清單中，先點選欲變更執行順序的 POU 之後，按下  圖示即可將該 POU 的執行順序往下移動一位，代表其執行順序延後。



- (3) 於工作 (TASK) 管理區的「已指派 POUs」清單中點選某一 POU 後，勾選或取消「啟動」選項也可改變該 POU 的啟動狀態，而設為未啟動狀態的 POU 於列表中亦會以灰階顯示。



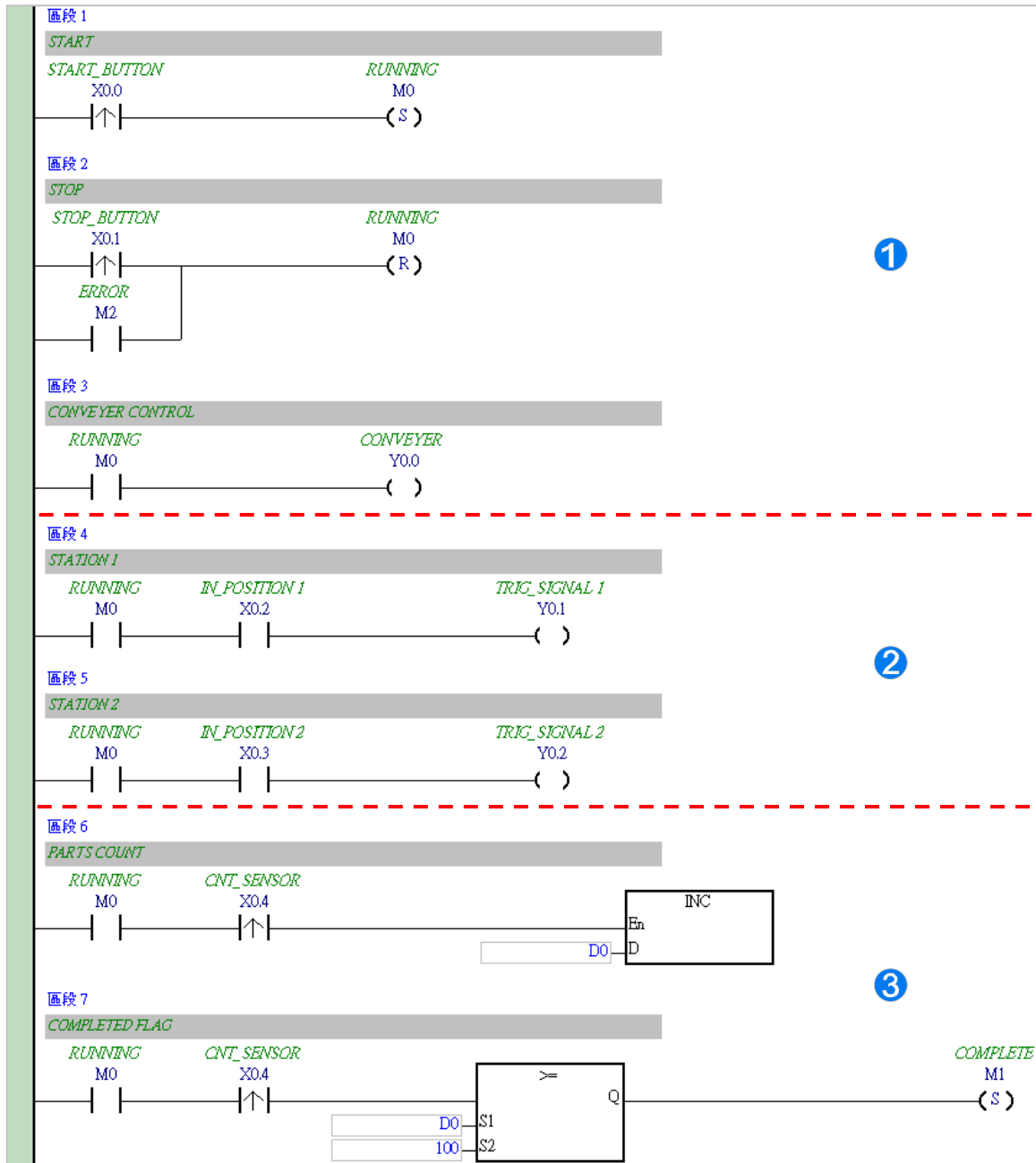
*.關於 POU 的啟動狀態請參考第 5.4.2 節的相關說明。

- (4) 完成所有工作 (TASK) 的設定後，按下工作管理工具的「確定」鍵即可。

5.6 範例實作

5.6.1 POU 與 TASK 的編程範例

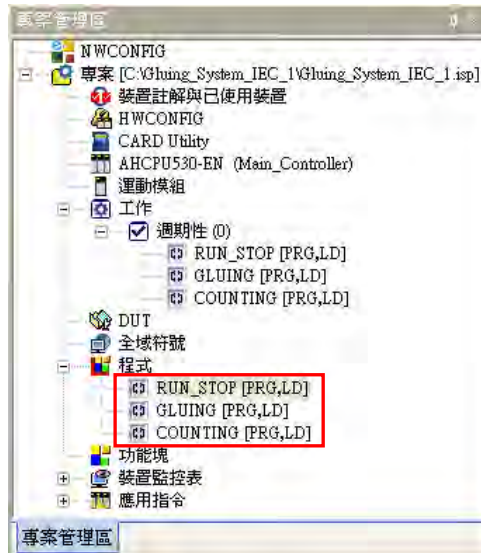
延續第 4 章內容中的範例，接著我們將在此節中以 POU 與 TASK 的編程架構重新進行規劃。下圖即為我們在第 4 章中所建立的程式，且在此我們將依據程序的功能與作用，而將其重新規劃為三個部份。



- ① 此部分的主要功能在於控制系統的啟動或停止。
- ② 此部分的程序主要在於點膠的操作。
- ③ 此部分的作用在於計數已送入的工件數量。

此節的內容將著重於程式架構的部份，而關於**硬體組態**、**參數設定**與**除錯測試**的操作則與第 4 章的說明內容相同，請試著自行完成相關的部分。

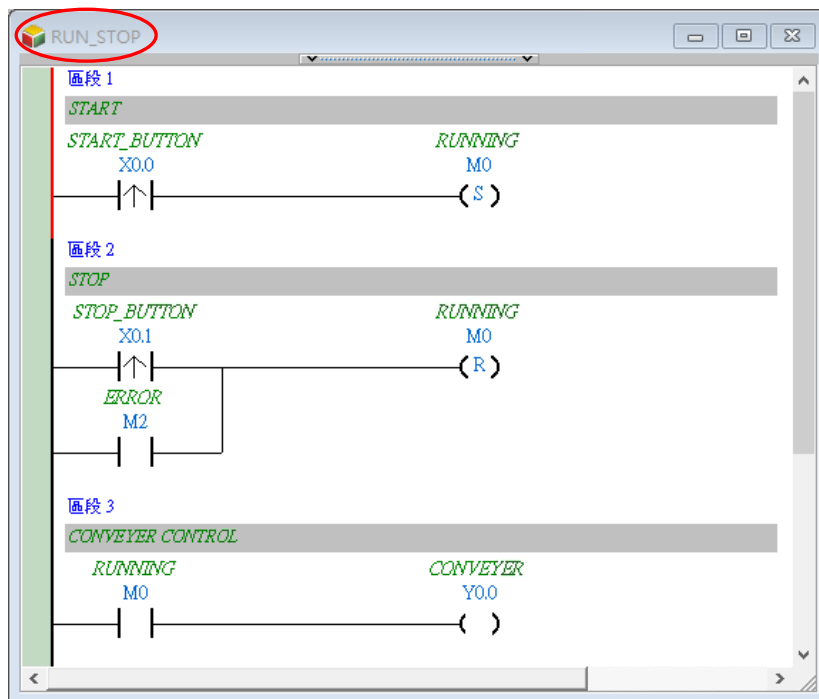
首先請新增一專案，並參考第 4 章的內容完成硬體組態的工作，接著依據上述的規劃於「程式」項目下分別建立如下的 3 個 POU，建立的方式請參考第 5.4.1 節的相關說明；而建立 POU 時，屬性視窗中的「工作」欄位請先保持「週期性 (0)」的預設值即可。



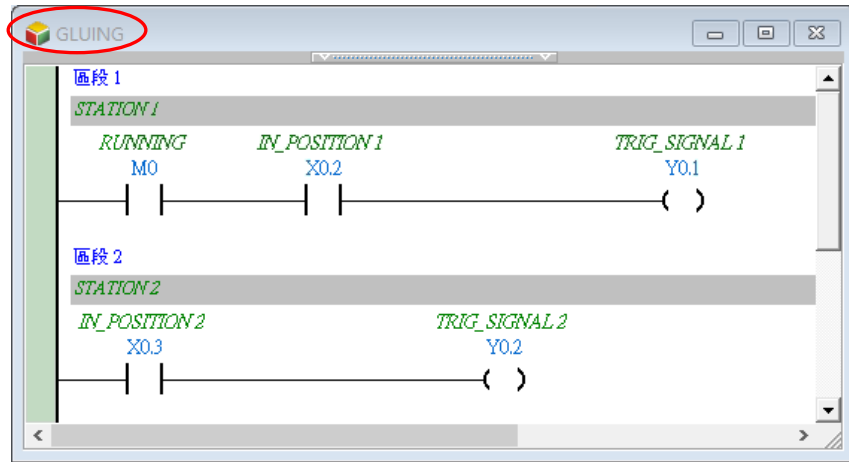
5

接著請參考第 4 章所介紹的階梯圖編輯方式，於各個 POU 視窗中完成以下的程式。

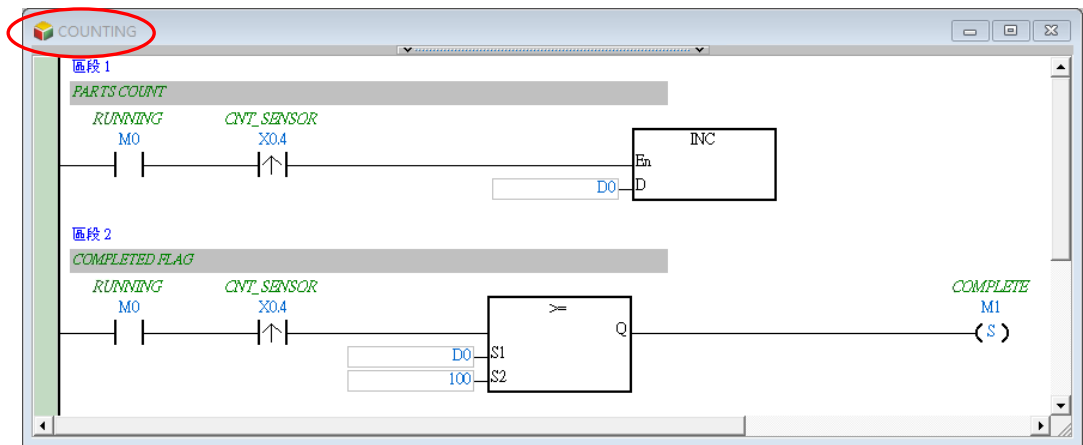
首先是控制系統啟動/停止的 POU – 「**RUN_STOP**」。



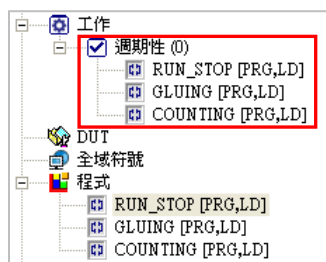
再來是控制點膠動作的 POU – 「GLUING」。



最後完成負責計數功能的 POU – 「COUNTING」。

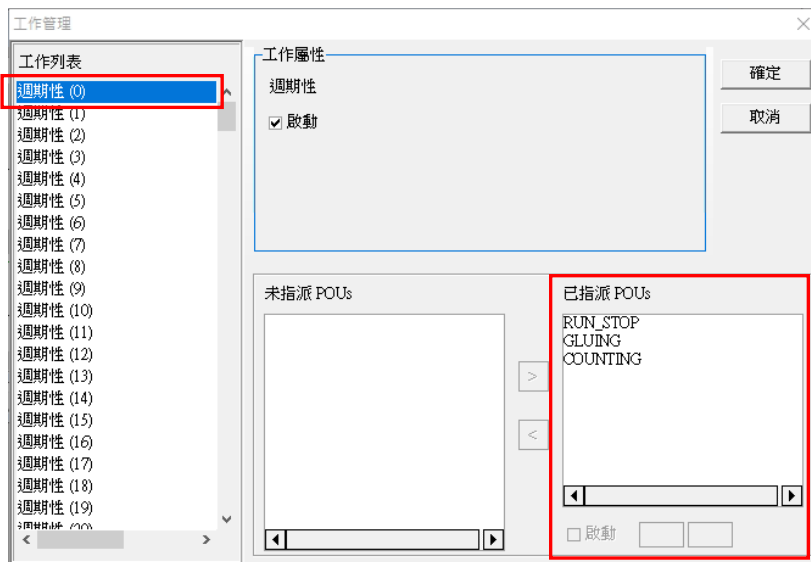
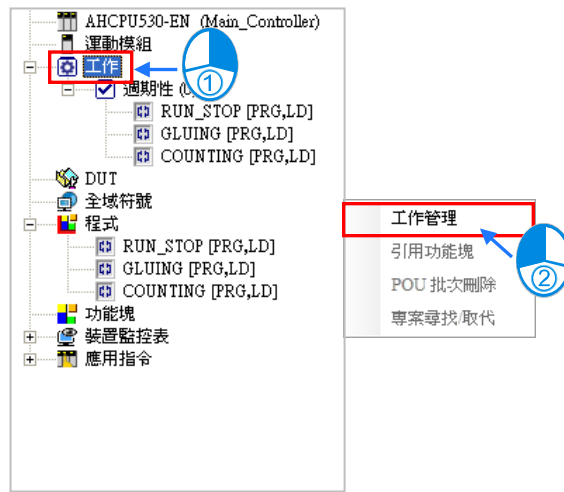


完成程式建立後，請確認「工作」項目下的 POU 順序與下圖一致。



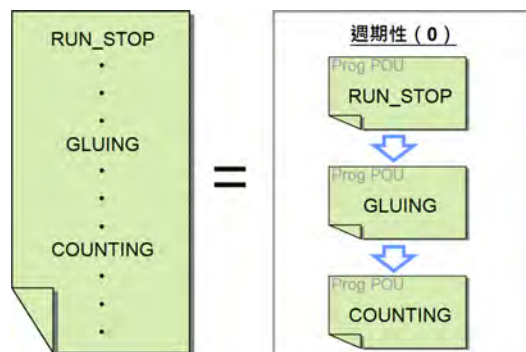
5

若要重新規劃各 POU 的工作配置時，請於專案管理區的「工作」項目上點擊滑鼠右鍵，之後於快捷選單中點選「工作管理」便可開啟工作管理工具，將 POU 加入到週期性(0)的已指派 POU 內。



5

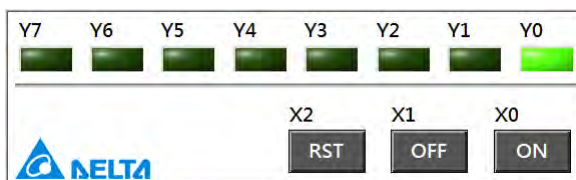
若已依據上述的順序完成 POU 的規劃，則之後此程式的執行結果便會與第 4 章的傳統階梯圖程式相同。接著請自行參考第 4 章的內容將程式完成編譯，並下載至機器中進行後續的測試。



5.6.2 DVP 機種的中斷程式範例

● 範例說明

當「ON」的按鈕 (X0) 被按下時，Y0~Y7 便會由 Y0 開始輪流掃瞄；當「OFF」的按鈕 (X1) 被按下時，輸出的動作便會停止，而燈號則會停留在目前的狀態；當「RST」的按鈕 (X2) 被按下時，輸出便會被重置，且讓掃描燈號重新由 Y0 開始。



● 程式規劃

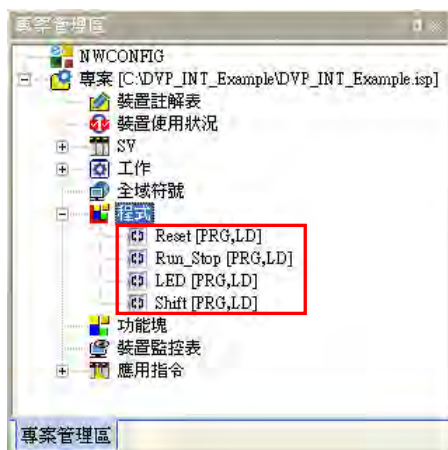
此範例所選用的機種為 SV，且在此我們將程式規劃為 4 個程式 POU，而各 POU 的功能請見下表。

POU 名稱	功能說明
Run_Stop	1. 初始化運轉旗標 (M0) 狀態與輸出資料暫存器 (D0) 的值，並致能中斷。 2. 當 X0 由 OFF 變為 ON 時，將運轉旗標 (M0) 設置為 ON。 3. 當 X1 由 OFF 變為 ON 時，將運轉旗標 (M0) 重置為 OFF。
LED	依據輸出資料暫存器 (D0) 的值更新 Y0~Y7 的輸出狀態。
Shift	每隔 0.5 秒，將輸出資料暫存器 (D0) 的內容偏移一個位元，並判斷當數值超過 128 (即偏移超過 8 個位元) 時，將輸出資料暫存器 (D0) 的值重設為 1 (代表重新由 Bit 0 開始進行偏移)。
Reset	將輸出資料暫存器 (D0) 的值重設為 1，亦即將輸出的位元重置為 Bit 0。且此 POU 預計會由外部中斷 I201 (X2 上升緣) 來觸發執行。

*.關於 SV 機種的中斷指標及相關的 API 應用指令請參考 DVP-PLC 應用技術手冊 (程式篇)。

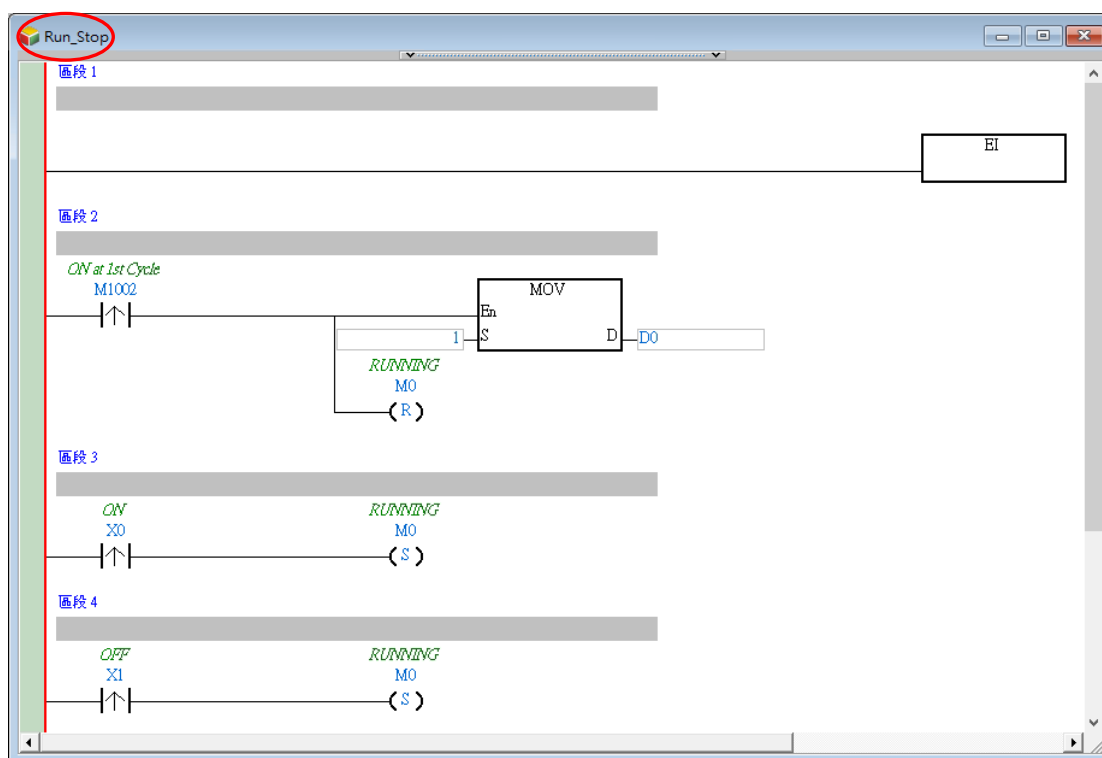
● 建立程式

首先請建立一個 SV 機種的專案，接著請於專案管理區的「程式」項目下分別建立如下的 4 個 POU，且 POU 屬性的「工作」欄位可先保持「週期性」的預設值即可。

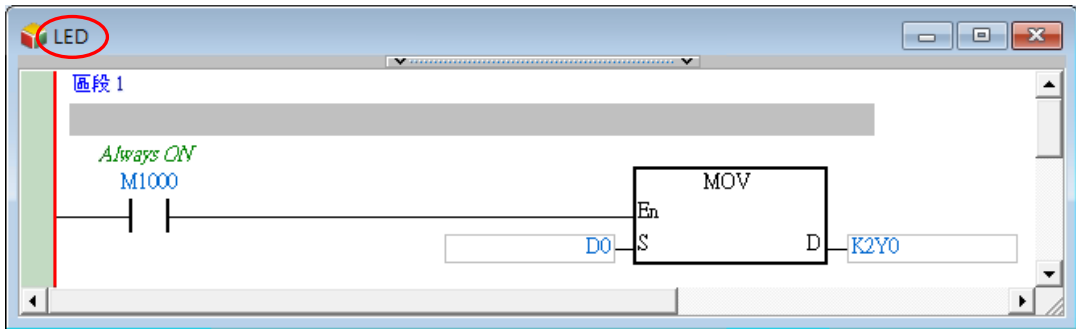


接著請於各 POU 的視窗中完成下列的程式。

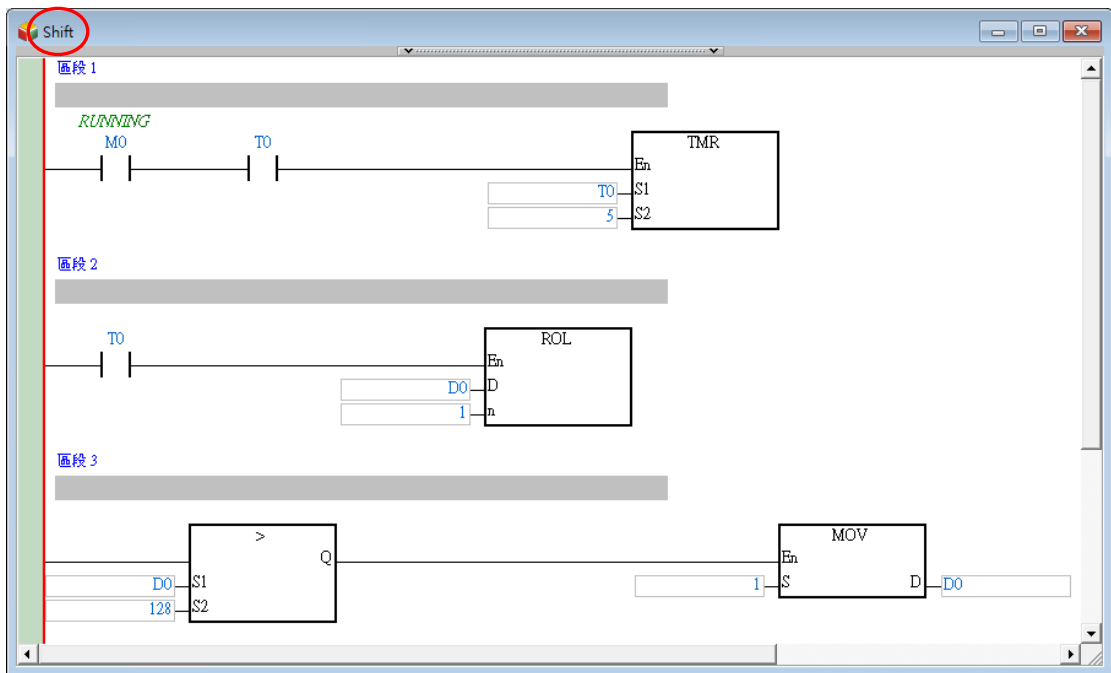
5



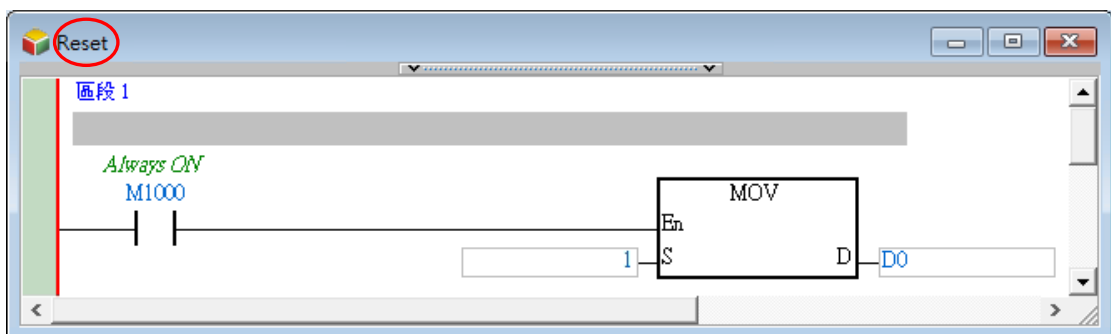
*.MOV 指令屬於 API 應用指令的「傳送比較」類別，而 EI 指令則屬於「迴路控制」類別。



*.MOV 指令屬於 API 應用指令的「傳送比較」類別。

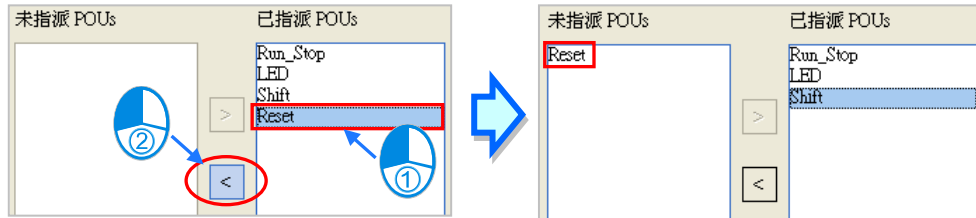


*.TMR 指令屬於 API 應用指令的「基本指令」類別，ROL 指令屬於「旋轉位移」類別，MOV 指令屬於「傳送比較」分類，而比較接點則可由專案管理區的「應用指令」下的「接點型態邏輯運算」分類中找到。



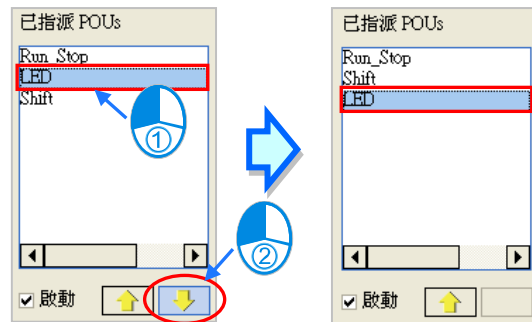
*.MOV 指令屬於 API 應用指令的「傳送比較」類別。

完成程式編輯後，請打開工作管理工具，而因建立 POU 時，其「工作」屬性都沿用「週期性」的預設值，因此開啟工作管理工具後，我們會發現所有的 POU 都配置在「週期性」的 TASK 清單中。但「Reset」的功能將由外部中斷來觸發，所以請將「Reset」由「週期性」的 TASK 清單中移除。

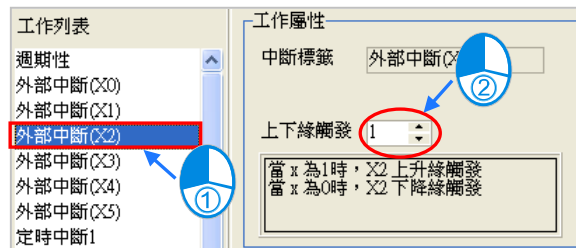


確定「週期性」的 TASK 清單後，接著便要對清單中的 POU 進行排序，而因為我們希望將初始化與輸入的功能放在掃描週期的最前面，而輸出的功能則放在掃描週期的最後，因此便必須將「週期性」的 TASK 清單依照「Run_Stop」→「Shift」→「LED」的順序重新排序。

5



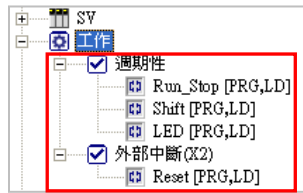
接著請於工作列表中選擇「外部中斷 (X2)」，並將屬性設定區域的「上下緣觸發」欄位設定為 1，代表此 TASK 的中斷條件為 X2 的上升緣觸發。



完成 TASK 的屬性設定後，接著請將「Reset」POU 配置至「外部中斷 (X2)」的 TASK 清單中，且因清單中只有一個 POU，因此無須對清單中的 POU 進行排序。



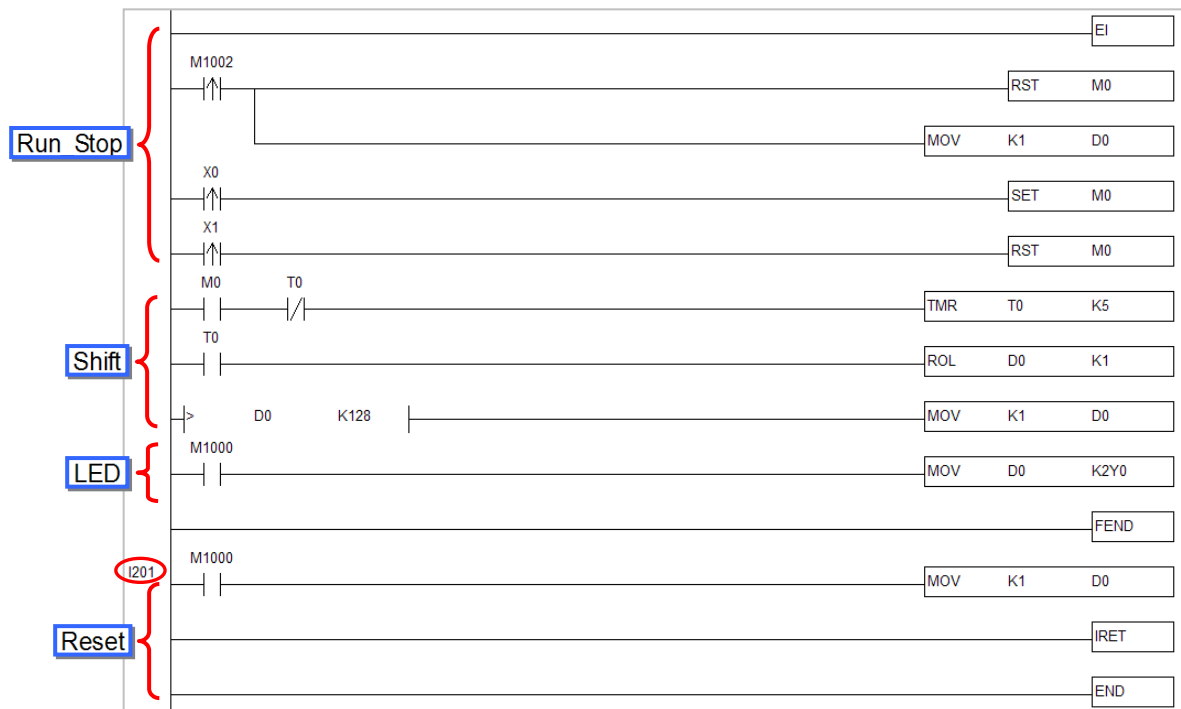
完成所有操作後請按下「確定」鍵套用於工作管理工具中所做的配置與設定，而其配置的結果如下。此範例程式到目前為止已編輯完畢，接著請自行編譯並試著下載至主機中進行後續的測試。



補充說明

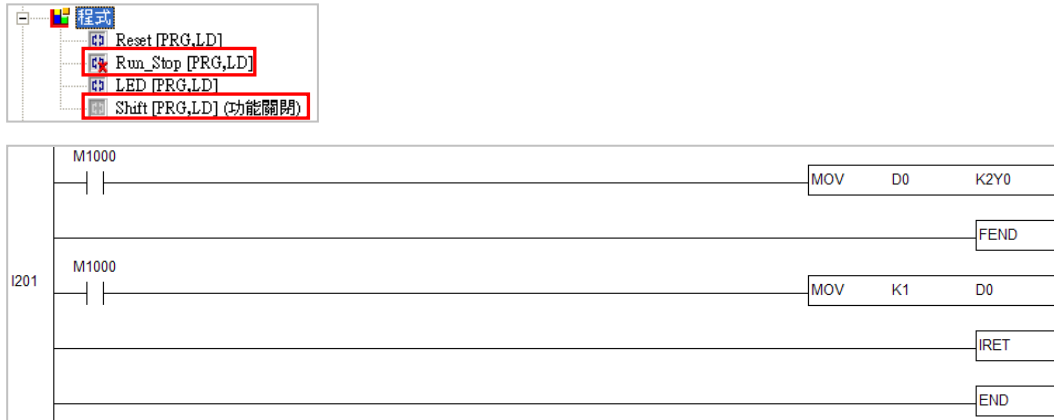
若上述的範例程式已成功編譯並下載至主機時，接著我們便可以試著透過 WPLSoft 軟體將 PLC 內部的程式碼再上傳回來進行比對，下列即是上傳回來的程式碼。

由下圖的程式可發現，主程式的部分即是由 3 個「週期性」TASK 的 POU 所組成，且組成的順序為「Run_Stop」→「Shift」→「LED」，而中斷副程式 I201 的內容則與「Reset」POU 相同。



5

接下來我們可以試著將「Run_Stop」由 TASK 清單中移除，並將「Shift」的 POU 狀態設為關閉，接著完成編譯並下載至主機，而之後再透過 WPLSoft 軟體將程式碼上傳，此時我們便可發現主程式的部份已不再包含這兩個 POU 的程序。

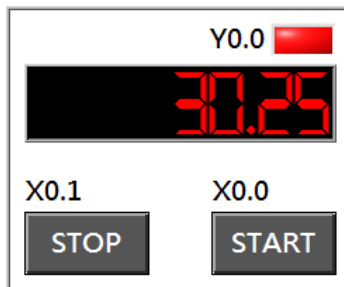


*.WPLSoft 亦為台達的 PLC 編程軟體，而其程式架構則屬於傳統的 PLC 編程架構，相關操作請參考 WPLSoft 的使用手冊。

5.6.3 AH5x0 機種的中斷程式範例

● 範例說明

按下「START」鍵 (X0.0) 時系統便會開始計時，之後必須在 30 秒內按下「STOP」鍵 (X0.1)，否則「警報器」(Y0.0) 便會啟動。

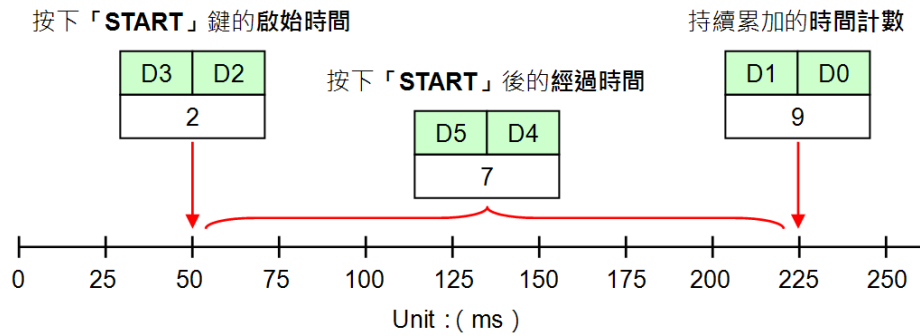


● 硬體規劃

本節的範例將選用 AHCPU530-EN 主機，數位 IO 模組則為 8 點輸入/8 點輸出的 AH16AP11R-5A，而主背板則選用 4 槽的 AHBP04M1-5A。(此規劃與第 4 章的範例相同)

● 程式規劃

此範例中，我們將利用定時中斷的特性來產生一個固定 25ms 時基的計數，並利用程式的規劃來達到計時的功能；如下圖，定時中斷 0 的 TASK 會每隔 25ms 就觸發一次，而每次觸發中斷時便會將 32Bits 的時間計數暫存器 (D0,D1) 的值加 1，且只要 PLC 維持 RUN 的狀態，此動作便會持續進行，藉此便可產生一個固定時基的計數；之後只要讀取時間計數暫存器 (D0,D1) 的值便可知 PLC 已經運轉的時間，例如時間計數暫存器 (D0,D1) 的值为 10，則 PLC 運轉的時間便是 250ms (10x25ms)。當「START」鍵按下時，程式會將時間計數暫存器 (D0,D1) 的現在值傳給啟始時間暫存器 (D2,D3)，之後時間計數暫存器 (D0,D1) 會繼續累加，而啟始時間暫存器 (D2,D3) 則會保持在「START」鍵被按下的瞬間；因此後續只要將時間計數暫存器 (D0,D1) 的值減去啟始時間暫存器 (D2,D3) 的值，便可知「START」鍵被按下之後的經過時間，我們會將這個值放到經過時間暫存器 (D4,D5) 中。



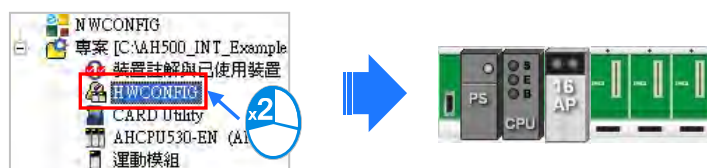
此節的範例除了會示範 AH5x0 機種的定時中斷之外，還將示範 **TKON** 與 **TKOFF** 這兩個 TASK 的控制指令。在此我們總共將程式規劃為 5 個程式 POU，其功能簡述如下。

POU 名稱	功能說明
INT_Timer	每 25ms 便會將 32Bits 的時間計數暫存器 (D0,D1) 的值加 1 。 (此 POU 將規劃於定時中斷 0 的 TASK 中)
Initialize	初始化旗標 (M0,M1) 及所有相關的暫存器 (D0~D7)，並致能中斷。 此 POU 會在第一個掃描週期執行時，將其它週期性 POU 所屬的週期性 (0) TASK 功能啟動，接著再將自己本身所屬的週期性 (1) TASK 功能關閉，所以此 POU 僅會在第一個掃描週期時被執行。
Control	當 X0.0 由 OFF 變為 ON 時，便啟動運轉旗標 (M0) 並將目前時間計數暫存器 (D0,D1) 的值傳送給啟始時間暫存器 (D2,D3)；當 X0.1 由 OFF 變為 ON 或計時產生溢位時，便關閉運轉旗標 (M0)。
Time_CHK	將時間計數暫存器 (D0,D1) 的值減去啟始時間暫存器 (D2,D3) 的值以求出「START」鍵被按下後的經過時間，並將其存放於經過時間暫存器(D4,D5) 中；且當計算出來的經過時間小於 0 時，便將溢位旗標 (M1) 設為 ON。
Signal	當經過時間超出 30 秒 (25ms x 1200) 或溢位旗標 (M1) 為 ON 時，將警報器 (Y0.0) 設置為 ON

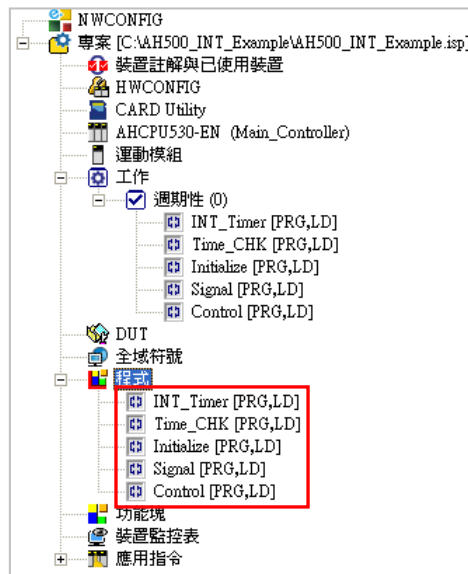
*.關於 AH5x0 機種的中斷指標及相關的 API 應用指令請參考 AH500 操作手冊及 AH500 程式手冊。

● 建立程式

首先請建立一 **AHCPU530-EN** 機種的專案，接著進到 HWCONFIG 增加一台 16AP 模組並下載硬體組態及參數設定到主機內。

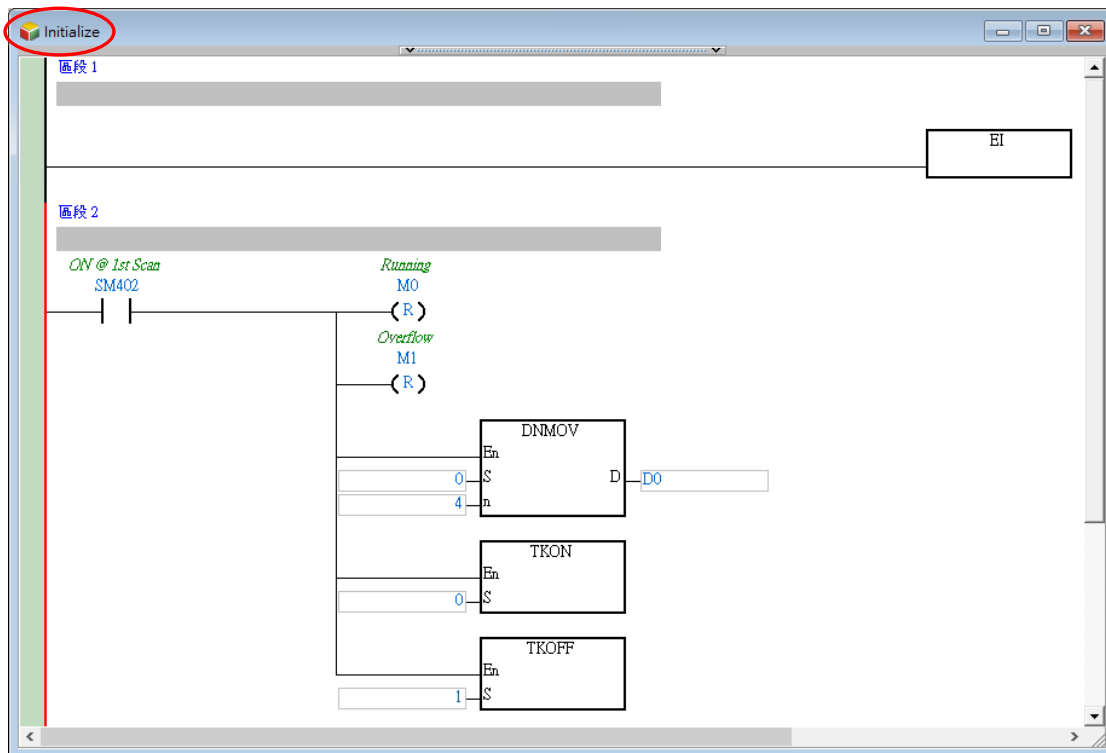


請於專案管理區的「程式」項目下分別建立如下的 5 個 POU，且 POU 屬性的「工作」欄位可先保持「週期性 (0)」的預設值即可。

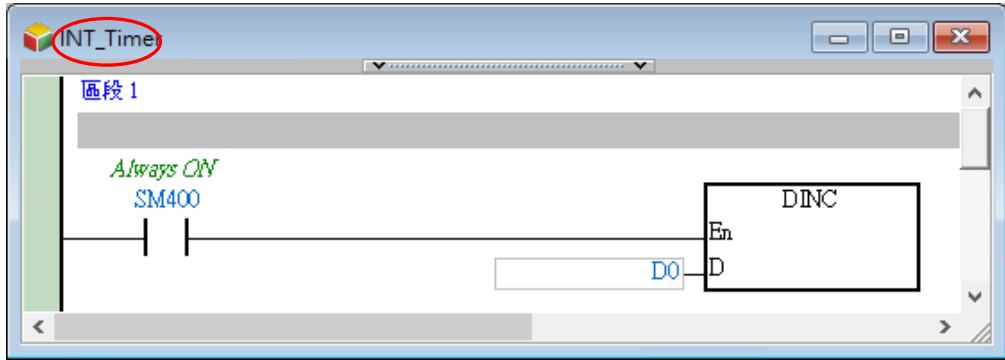


接著請於各 POU 的視窗中完成下列的程式。

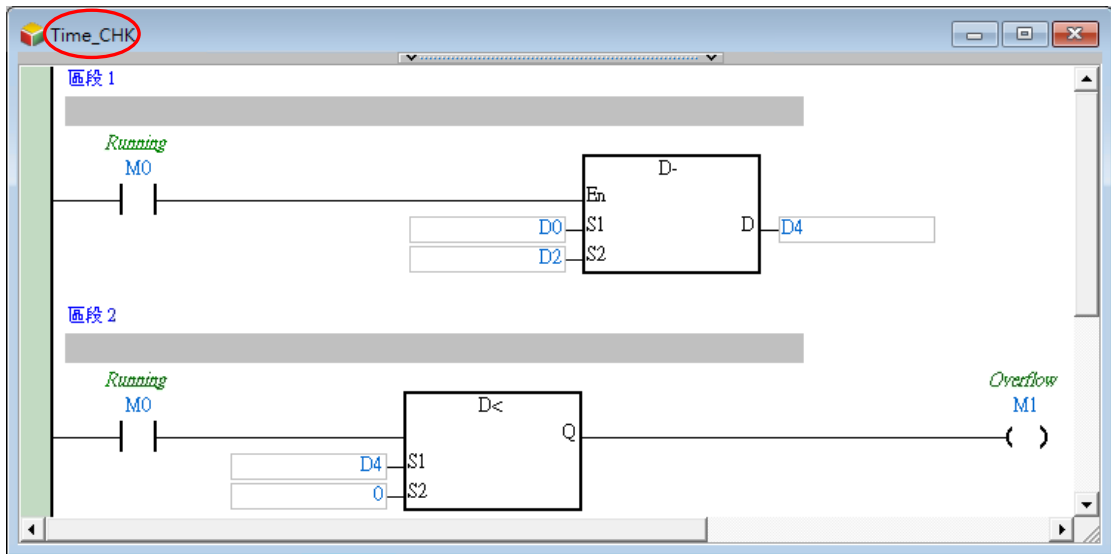
5



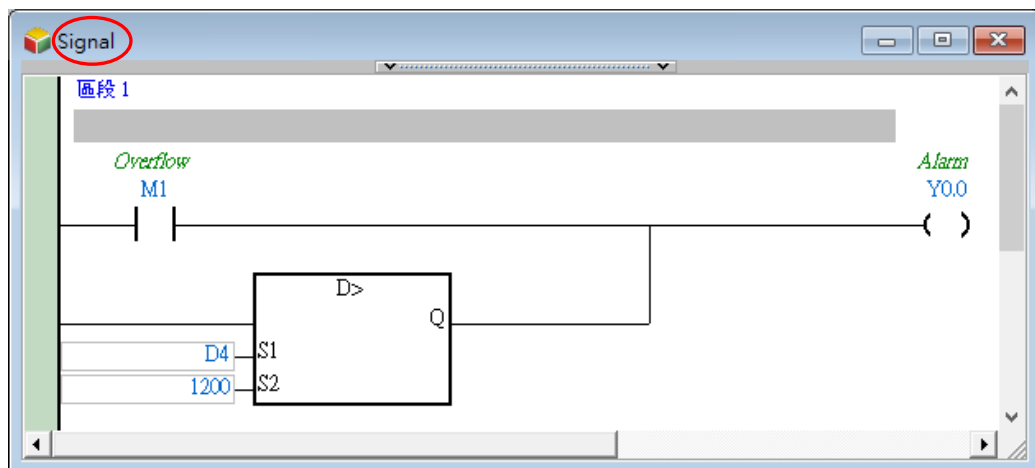
*.EI 指令屬於 API 應用指令下的「迴路控制」類別，DNMOV 指令則屬於「資料傳輸」類別，而 TKON 指令與 TKOFF 指令則屬於「工作管理」類別。



*.DINC 指令屬於 API 應用指令下的「四則邏輯運算」類別。

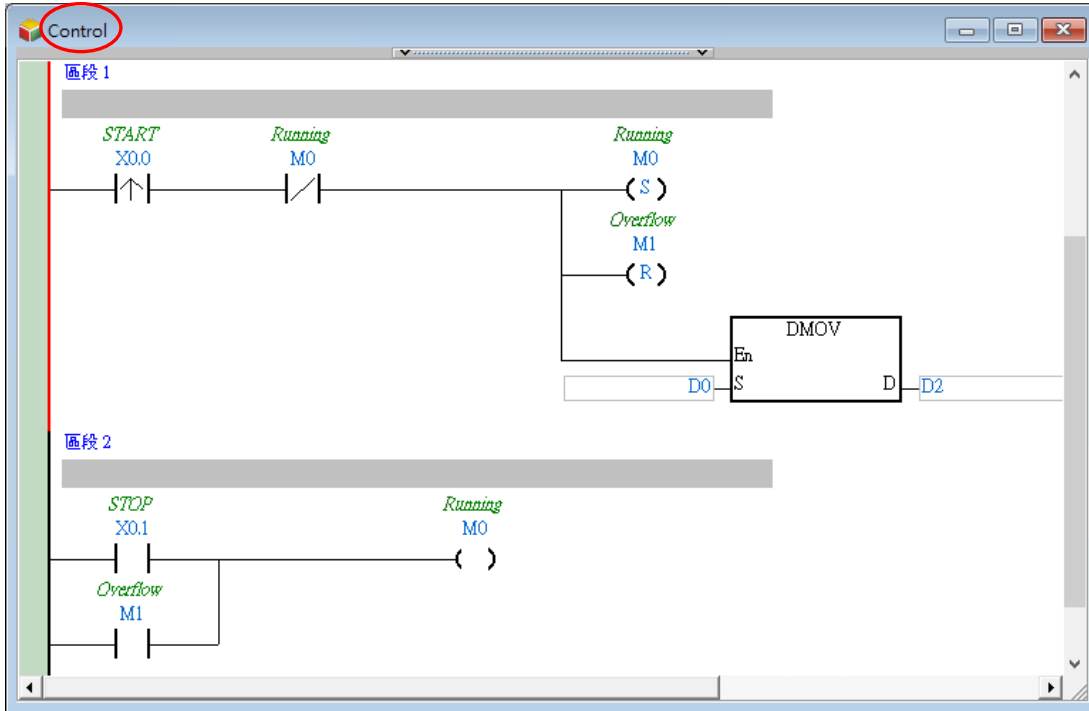


*.D- 指令屬於 API 應用指令下的「四則邏輯運算」類別，比較接點 D< 則位於「接點型態邏輯運算」的分類中。



*.比較接點 D> 可由專案管理區的「應用指令」下的「接點型態邏輯運算」分類中找到。

5



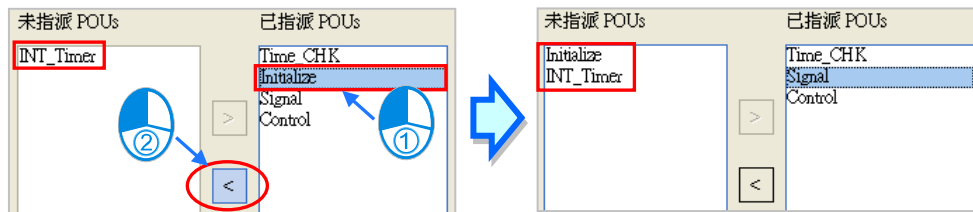
*.DMOV 指令屬於 API 應用指令下的「資料傳輸」類別。

5

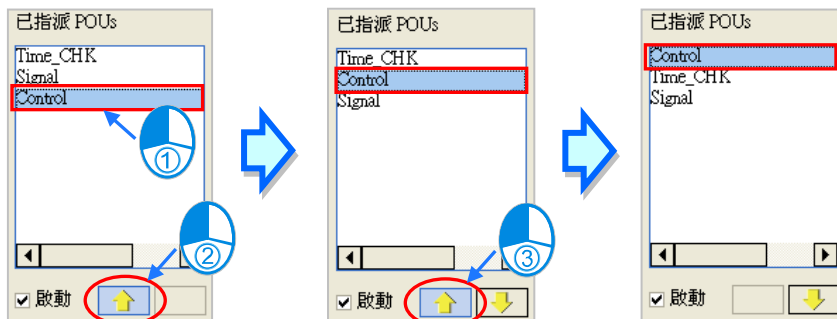
完成程式編輯後請打開工作管理工具，首先設定「週期性 (0)」的 TASK 屬性，而在程式的規劃中，「週期性 (0)」TASK 的功能將會由「Initialize」這個 POU 來啟動，因此我們必須先取消勾選「週期性 (0)」的「啟動」項目。



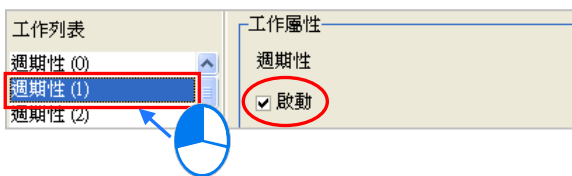
因「INT_Timer」須配置於「定時中斷 0」的 TASK，而「Initialize」則須配置在「週期性 (1)」的 TASK 中，所以在此先將這兩個 POU 由「週期性 (0)」的 TASK 清單中移除。



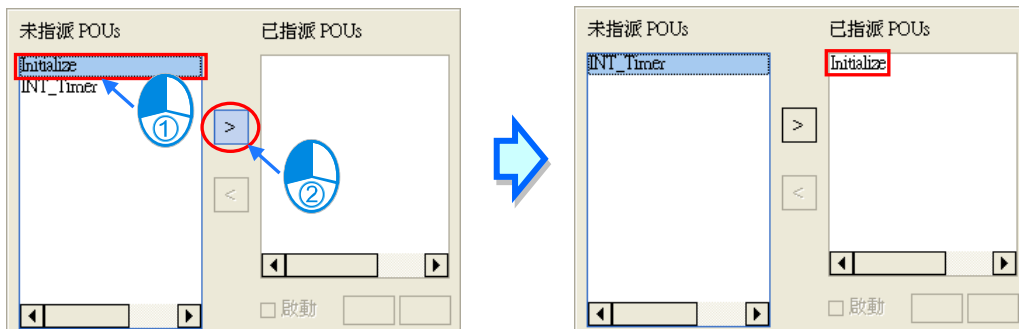
確定「週期性 (0)」的 TASK 清單後，接著便要對清單中的 POU 進行排序，而因為我們希望將與輸入相關的功能放在掃描週期的最前面，而與輸出相關的功能則放在掃描週期的最後，因此將 TASK 清單中的 POU 依照「Control」→「Time_CHK」→「Signal」的順序重新排序。



接著我們選擇「週期性 (1)」的 TASK，且因「週期性 (1)」的 TASK 必須先在第一個掃描週期之後才會關閉，因此請直接保留其「啟動」項目的勾選狀態。

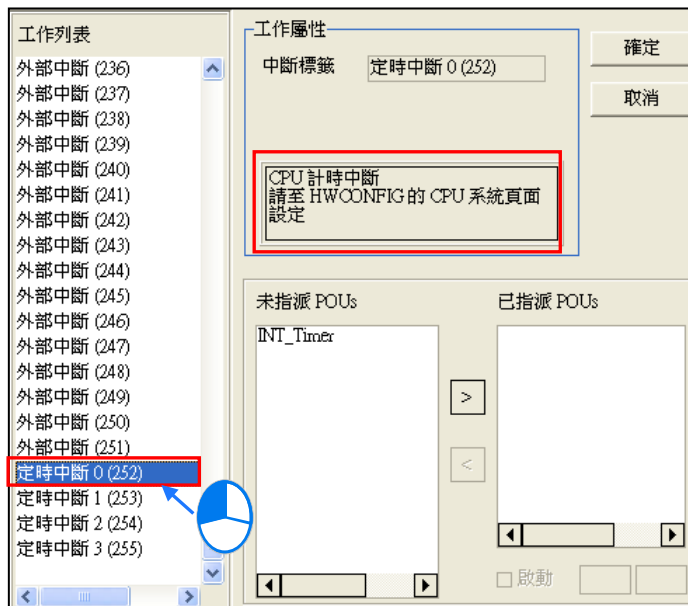


之後將「Initialize」這個 POU 配置到「週期性 (1)」的 TASK 中即可，且因 TASK 中只有一個 POU 所以無須進行排序。



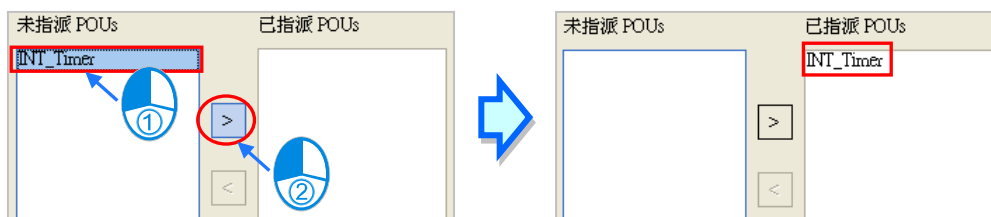
5

接著我們選擇「定時中斷 0 (252)」的 TASK，而由工作屬性區的提示得知，此 TASK 的屬性必須於 HWCONFIG 中進行設定，因此我們可在此先完成 TASK 的配置，後續再另外進行工作屬性的設定。

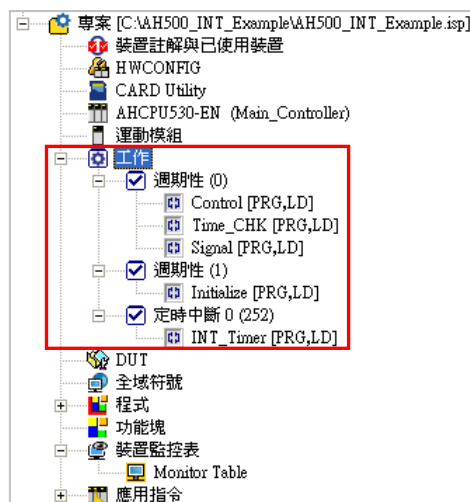


5

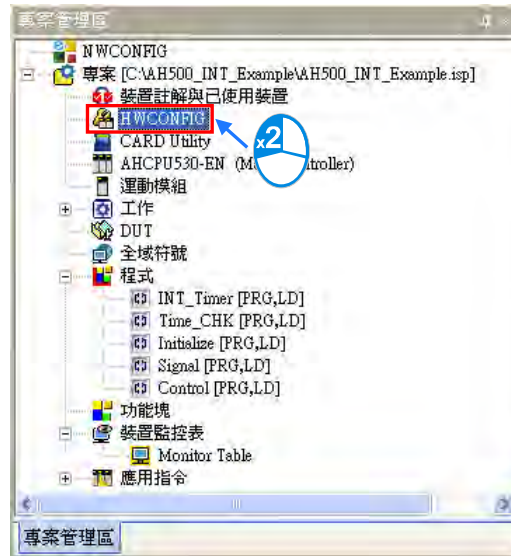
將「INT_Timer」這個 POU 配置到「定時中斷 0 (252)」的 TASK 中。



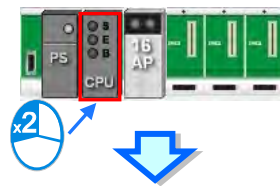
完成所有操作後請按下「確定」鍵套用於工作管理工具中所做的配置與設定，而配置的結果如下。



接著便要繼續設定「定時中斷 0」的工作屬性。請先開啟 **HWCONFIG** 工具。



開啟 **HWCONFIG** 工具後，請於 CPU 模組的圖示上雙擊滑鼠左鍵以開啟 CPU 的屬性設定視窗。接著切換至「CPU」主類別下的「系統」副類別。



5



於「系統」副類別的頁面中可見到「定時中斷時間設定」的區域；將「定時中斷 0」欄位的值設定為 25ms 後，按下視窗右下角的「確定」鍵即可完成設定。

定時中斷 0	<input type="text" value="25"/>	ms (1 ~ 1000 ms)
定時中斷 1	<input type="text" value="40"/>	ms (1 ~ 1000 ms)
定時中斷 2	<input type="text" value="20"/>	ms (1 ~ 1000 ms)
定時中斷 3	<input type="text" value="10"/>	ms (1 ~ 1000 ms)

完成後請將 **HWCONFIG** 中的設定存檔，並將此處的設定值一併下載至主機中才可正式生效。
接著請將程式編譯，並試著下載至主機進行測試。

5

6

第6章 變數符號

目錄

6.1 認識變數符號.....	6-2
6.1.1 變數符號的作用範圍與命名原則.....	6-2
6.1.2 變數符號的類別.....	6-3
6.1.3 變數符號的資料型態.....	6-5
6.1.4 變數符號的位址配置與初始值.....	6-10
6.1.5 變數符號的間接指定暫存器修飾.....	6-12
6.1.6 變數符號的位元操作 (AH/AS 機種限定).....	6-14
6.2 ISPSOft 中的變數符號管理.....	6-15
6.2.1 變數符號表.....	6-15
6.2.2 新增變數符號.....	6-18
6.2.3 陣列與字串型態符號的使用原則.....	6-23
6.2.4 變數符號的修改與符號表編輯.....	6-26
6.2.5 清除變數符號的配置位址.....	6-28
6.2.6 下載變數符號的初始值.....	6-29
6.2.7 符號表的匯出.....	6-31
6.2.8 符號表的匯入.....	6-32
6.2.9 符號表的排序.....	6-35
6.2.10 變數符號的篩選.....	6-36
6.2.11 裝置資源分配設定.....	6-37
6.3 範例實作.....	6-38
6.3.1 變數符號規劃.....	6-39
6.3.2 建立範例程式.....	6-40

6.1 認識變數符號

在傳統的 PLC 程式開發過程中，往往需要花費很多的時間在管理裝置位址，而在面對大型的專案時，不論在管理或除錯上都會是很大的負擔。有鑑於此，在 IEC 61131-3 中便導入了一般高階語言中的變數觀念，除了可自行定義變數符號名稱來取代 PLC 的裝置編號之外，更可利用自動配置位址的功能來節省配置裝置的時間，同時也提高了程式的可讀性及開發效率。

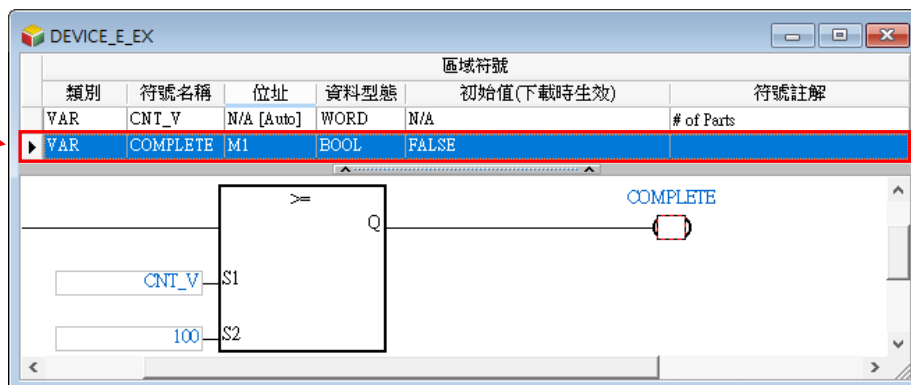
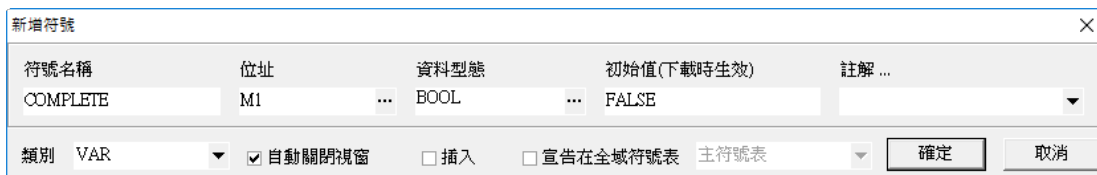
*變數在 ISPSOft 中被稱之為符號 (Symbol)，因此在手冊之中，變數與符號代表的將會是相同的意義。

6.1.1 變數符號的作用範圍與命名原則

變數符號在使用前必須經過宣告的程序 (如下圖)，且依據宣告位置的不同又可分為**全域符號**與**區域符號**；**全域符號**可讓專案中的所有 POU 共用，而**區域符號**則只能在宣告該符號的 POU 當中使用。此外，在不同 POU 當中的區域符號名稱是可以重複的，但若宣告的區域符號與某個全域符號的名稱重複時，則在宣告該區域符號的 POU 當中，系統會自動將此名稱的變數符號視為區域符號。

變數符號在命名時有其一定的規則與限制，說明如下。

- 變數符號的名稱不可超過 30 個字元，且須注意一個中文字會佔用兩個字元。
- 不可使用系統保留的關鍵字，如指令名稱、裝置名稱，或是其它在系統中已被賦予特殊意義的保留字，但若只是被包含在名稱當中則是合法的，如“M0”是不合法的名稱，但“_M0”則是合法的。
- 變數符號的名稱不可與列舉名稱重複，列舉的用法請參閱第 8 章。
- 變數符號的名稱當中不可有空白，例如“INPUT CH0”便是不合法的名稱。
- 名稱當中可使用底線，但不可連續使用，或是置於結尾，例如“INPUT_CH0”是合法的名稱，但“INPUT__CH0”與“INPUT_CH0_”則是不合法的名稱。
- 變數符號的名稱不可使用特殊字元，例如 *、#、?、\、%、@ ...等。



*變數符號請避免將第一個字命名為數字，以及 DFB_ (系統內定的函式庫開頭)；變數符號的位址建議使用自動配置以達分配的最佳化。

6.1.2 變數符號的類別

變數符號的類別依功能可分為下列 7 種，而在全域變數支援的符號類別與程式（區域變數）相同，程式（區域變數）、功能塊（區域變數）及函式（區域變數）各種類別的特性與使用說明如下表所示。

變數符號 類別	區域變數						
	程式			功能塊			函式
	AH/AS	DVP	AS5xx/ DVPxxMC	AH/AS	DVP	AS5xx/ DVPxxMC	AS5xx/ DVPxxMC
VAR	✓	✓	✓	不可指定 位置	只支援 ES3 且不可指 定位置	不可指定 位置	不可指定 位置
VAR_RETAIN	不可指定 位置 (*1)	只支援 ES3 且不可指 定位置 (*1)	✓			不可指定 位置	不可指定 位置
VAR_STATIC			✓			不可指定 位置	不可指定 位置
VAR_STATIC_RETAIN			✓			不可指定 位置	不可指定 位置
VAR_INPUT				不可指定 位置	不可指定 位置	不可指定 位置	不可指定 位置
VAR_OUTPUT				不可指定 位置	不可指定 位置	不可指定 位置	
VAR_IN_OUT				不可指定 位置	不可指定 位置		

*1. 其資料型態亦不可為 **TIMER**、**COUNTER** 或 **STEP**。

● VAR - 一般變數符號

當變數符號被宣告成此類別時，代表該變數符號僅供一般的運算之用，而其所代表的意義則必須視宣告時，該變數符號被指定的資料型態或是對應的裝置名稱。

● VAR_RETAIN - 斷電保持型的一般變數符號

當變數符號被宣告成此類別時，會自動被配置在斷電保持的區域，因此在 PLC 主機斷電後，該變數符號仍會保有最後的狀態值。

● VAR_STATIC - 靜態型的變數符號

具有靜態屬性，STATIC 型態的變數只能在定義該變數的 POU 中使用。當變數是 STATIC 型態時，系統會給該變數分配一個固定的位址，使得該變數的值在控制器運行過程於相同 POU 的不同 Instance 之間能夠共享變數。

● VAR_STATIC_RETAIN - 靜態型的斷電保持變數符號

當變數符號被宣告成此類別時，會變成具有靜態屬性且有斷電保持，因此在 PLC 主機斷電後，該變數符號仍會保有最後的狀態值。

● VAR_INPUT - 功能塊輸入端點的變數符號

此類別的變數符號主要用做功能塊的輸入端點，當功能塊接受呼叫時，便可利用此類別的變數符號來接收呼叫方的輸入值。在 LD/FBD 等編程環境中，此類別的變數符號會被排列在功能塊圖示的左側，並會配置一個端點用以接受呼叫方的輸入值。

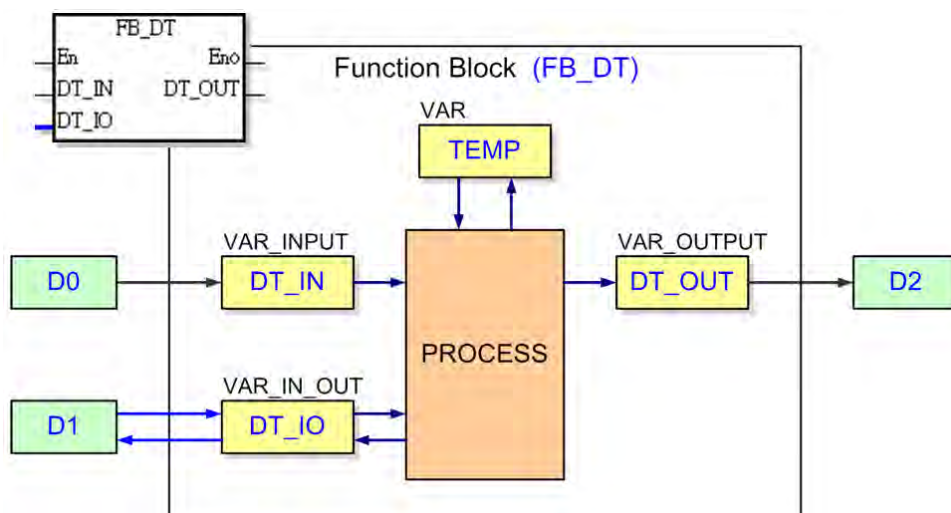
● VAR_OUTPUT - 功能塊輸出端點的變數符號

此類別的變數符號主要用做功能塊的輸出端點，當功能塊執行完畢之後，其運算的結果便可經由此類別的變數符號來回傳給呼叫方。在 LD/FBD 等編程環境中，此類別的變數符號會被排列在功能塊圖示的右側，並會配置一個端點用以輸出結果。

● VAR_IN_OUT - 功能塊反饋型端點的變數符號

此類的變數符號，主要用做功能塊的反饋型端點，其功能請參考下列。

當功能塊被呼叫時，呼叫方會先將 D1 的狀態值輸入 VAR_IN_OUT 類別的變數符號 DT_IO，而運算結束後，DT_IO 便會再將自己最後的狀態值反饋輸出至 D1。在 LD/FBD 等編程環境中，此類別的變數符號會被排列在功能塊圖示的左側，且該端點會以藍色的粗線來與呼叫方進行連結。



6.1.3 變數符號的資料型態

變數符號的資料型態會決定其內容值所代表的意義，例如有兩個符號 VAR_1 與 VAR_2，其中 VAR_1 的資料型態為 BOOL，而 VAR_2 則是 WORD 型態；當兩者在程式當中被使用時，VAR_1 便可用來代表一個接點裝置，而 VAR_2 則可用來存放 16 Bits 大小的數值，且可進行四則運算或資料搬移。

下表為 ISPSOft 所支援的資料型態，而除了不同機種所支援的資料型態會有所不同之外，在全域變數與程式（區域變數）大致相同，但與功能塊（區域變數）及函式（區域變數）可宣告的變數符號有所不同，請參考下表。

資料型態	說明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程式	功能塊	程式	功能塊	程式	功能塊	函式
BOOL	布林格式，使用 0、1 或 TRUE or FALSE 其內容值可代表一個接點裝置的狀態；AS5xx/DVPxxMC 則只支援 TRUE or FALSE，預設值為 FALSE。	✓	✓	✓	✓	✓	✓	✓
BYTE	8 Bits 數值格式，有效範圍為 16#00 ~ FF 可存放 8 個 Bit 大小的資料，預設值為 16#00。					✓	✓	✓
WORD	16 Bits 數值格式，有效範圍為 16#0000 ~ FFFF 可存放 16 個 Bit 大小的資料，預設值為 16#0000。	✓	✓	✓	✓	✓	✓	✓
DWORD	32 Bits 數值格式，有效範圍為 16#00000000 ~ FFFFFFFF 可存放 32 個 Bit 大小的資料，預設值為 16#00000000。	✓	✓	✓	✓	✓	✓	✓
LWORD	64 Bits 數值格式，有效範圍為 16#0000000000000000 ~ FFFFFFFFFFFFFFFF 可存放 64 個 Bit 大小的資料，預設值為 16#0000000000000000。	只支援 AH	只支援 AH			✓	✓	✓
SINT	8 Bits 的帶符號整數格式，有效範圍為 -128 ~ 127 其最高位元代表正負號，預設值為 0。					✓	✓	✓
INT	16 Bits 的帶符號整數格式，有效範圍為 -32768 ~ 32767 其最高位元代表正負號，預設值為 0。	✓	✓	只支援 ES3	只支援 ES3	✓	✓	✓

資料型態	說明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程式	功能塊	程式	功能塊	程式	功能塊	函式
DINT	32 Bits 的帶符號整數格式，有效範圍為-2147483648 ~ 2147483647 其最高位元代表正負號，預設值為 0。	✓	✓	只支援 ES3	只支援 ES3	✓	✓	✓
LINT	64 Bits 的帶符號整數格式，有效範圍為-9223372036854775808 ~ 9223372036854775807 其最高位元代表正負號，預設值為 0。	只支援 AH	只支援 AH			✓	✓	✓
USINT	8 Bits 的無符號整數格式，有效範圍為 0 ~ 255，預設值為 0。					✓	✓	✓
UINT	16 Bits 的無符號整數格式，有效範圍為 0 ~ 65535，預設值為 0。					✓	✓	✓
UDINT	32 Bits 的無符號整數格式，有效範圍為 0 ~ 4294967295，預設值為 0。					✓	✓	✓
ULINT	64 Bits 的無符號整數格式，有效範圍為 0 ~ 18446744073709551615，預設值為 0。					✓	✓	✓
REAL	32 Bits 的浮點數值格式，有效範圍為-3.402823e+38 ~ -1.175495e-38、0 及 1.175495e-38 ~ 3.402823e+38，其內容為利用 32 Bits 的資料來表達一個具有小數點的數值，預設值為 0.0。	✓	✓	✓	✓	✓	✓	✓

資料型態	說明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程式	功能塊	程式	功能塊	程式	功能塊	函式
LREAL	64 Bits 的浮點數值格式，有效範圍為-1.79769313486231e+308 ~ -2.22507385850721e-308、0 及 2.22507385850721e-308 ~ 1.79769313486231e+308，其內容為利用 64 Bits 的資料來表達一個具有小數點的數值，預設值為 0.0。	只支援 AH	只支援 AH			✓	✓	✓
ARRAY	陣列格式，而陣列長度與存放的資料型態則必須於宣告時一併指定（最大的陣列長度為 512）。	✓	✓	✓	✓	✓	✓	✓
STRING	字串格式，以 8 個 Bit 代表 1 個 ASCII 字元，AH/AS 進行宣告時須一併指定字串長度，最多 128 個字元；AS5xx/DVPxxMC 的字串格式在頭尾，必須加上單引號，如 'abcd'，初始值最大長度可為 32000 個字元，若不設定初始值則字元預設長度為 80 個字元。	✓	✓	只支援 ES3	只支援 ES3	✓	✓	✓
STEP	用來做為步進點狀態的識別旗標 (*1)(*2)。	✓		✓	不支援 ES3			
功能塊	代表一個功能塊實例的名稱(*2)。	✓	✓	✓	只支援 ES3	✓	✓	✓
COUNTER	計數器格式，用來代表計數器裝置。	✓		✓	不支援 ES3			
TIMER	計時器格式，用來代表計時器裝置。	✓		✓	不支援 ES3			
POINTER	類別選取 VAR_IN_OUT 才能選用 WORD 裝置的指標型態 (*2)。		✓		只支援 ES3			
T_POINTER	類別選取 VAR_IN_OUT 才能選用，計時器專用的指標型態(*2)。		✓		只支援 ES3			

資料型態	說明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程式	功能塊	程式	功能塊	程式	功能塊	函式
C_ POINTER	類別選取 VAR_IN_OUT 才能選用，計數器專用的指標型態變數 (*2)。		✓		只支援 ES3			
HC_ POINTER	類別選取 VAR_IN_OUT 才能選用，高速計數器專用的指標型態變數 (*2)。		✓		只支援 ES3			
TIME	時間格式、可以使用 TIME#XXXXXXdXXhXXmXXsX XX.XXXms，或 T#XXXXXXdXXhXXmXXsXXX. XXXms 輸入，以 ms 為單位。有效範圍為： T#0ms~T#213503d23h34m33s7 09.551ms 數值可以為小數但不能出現負數；數值可以超出時間的有效範圍，如 T#25h 與 T#1d1h； d、h、m、s、ms 單位可以減少，但至少保留一個，且不能任意排列；數值必須為 10 進制，預設值為 T#0ms。					✓	✓	✓
DATE	時間格式、可以使用 DATE#年- 月-日，或 D#年-月-日。有效範圍為： D#1970-01-01~D#2106-02-07。 以日為單位。年、月、日均不能缺少，且不能超出有效範圍；數值前可以新增多個 0；數值必須為 10 進制，預設值為 D#1970-01-01。					✓	✓	✓

6

資料型態	說明	AH/AS		DVP		AS5xx/ DVPxxMC		
		程式	功能塊	程式	功能塊	程式	功能塊	函式
TOD	時間格式、可以使用 TIME_OF_DAY#時：分：秒.毫秒 或 TOD#時：分：秒.毫秒。有效 範圍為： TOD#00:00:00~23:59:59.999。以 ms 為單位。數值為 0 時顯示 TOD#00:00:00，數值為 1 時顯示 TOD#00:00:00.001，數值為 86399999 時顯示 TOD#23:59:59.999，數值為 86400000 時顯示 TOD#00:00:00，數值為 4294967295 時顯示 TOD#17:2:47.295，預設值為 TOD#00:00:00。					✓	✓	✓
DT	時間格式、可以使用 DT#年-月- 日-時-分-秒輸入。有效範圍為： DT#1970-01-01-0:0:0~DT#2106 -02-07-6:28:15。以 s 為單位，預 設值為 DT#1970-01-01-0:0:0。					✓	✓	✓
Pointer	指標變數存儲的是一個記憶體的 位址，程式中使用函式 ADR 可得 到某個變數的位址；對符號使用 「^」可取得位址的內容數值。值 得注意的是，使用指標前必須先給 指標值，否則會造成運行異常！					✓	✓	✓
Reference	需要設定初始值並且設定為相同 資料型態的符號，表示為此符號的 別名 (Alias)。舉例，在符號表宣 告一個資料型態為 REFERENCE TO INT 的符號 (例如：myref)， 初始值欄位填入被參考的符號 (例 如：mydata)，使用 myref 等同於 使用 mydata。					✓	✓	✓

*1. 關於步進點的說明請參考 第 14 章 - 循序功能圖 SFC。

*2. 功能塊型態與指標型態的變數符號有其特殊意義，相關說明請參考 第 7 章-功能塊 FB 與函式庫。

6.1.4 變數符號的位址配置與初始值

每個變數符號都會依據其資料型態而配置一個對應的裝置位址，並且可為其設定一個**初始值**，當下載專案程式時便可選擇是否將**初始值**一併寫入該變數符號所對應的裝置位址中，**初始值**的相關操作請參考本章第 6.2.6 節的說明。

下表為變數符號的位址配置原則，而不同的機種，其配置原則也不相同。

機種	配置原則
AH/AS	<ul style="list-style-type: none"> ● 全域符號與程式 POU 的區域符號可自行指定對應的裝置位址或交由系統自動配置。 ● 功能塊的區域符號僅可交由系統自動配置，而無法自行指定。 ● 自動配置的位址會配置在主機預留的記憶體區塊，而不會佔用 D 或 M 裝置；但針對 STEP、TIMER 與 COUNTER 幾種型態，自動配置時仍舊會為其配置 S、T 與 C 裝置。
DVP	<ul style="list-style-type: none"> ● 所有的全域符號與區域符號皆可自行指定對應的裝置位址或交由系統自動配置。 ● 系統自動配置的位址皆會佔用可用的裝置位址。(使用者可設定自動配置的裝置範圍)
AS5xx/ DVPxx MC	<ul style="list-style-type: none"> ● 全域符號與程式 POU 的區域符號可自行指定對應的裝置位址或交由系統自動配置。 ● 功能塊的區域符號僅可交由系統自動配置，而無法自行指定。 ● 自動配置的位址會配置在主機預留的記憶體區塊，而不會佔用 I / Q / M 裝置。

下表則為各種資料型態可指定或配置的裝置種類。

6

資料型態	AH/AS		DVP	
	自行指定	自動配置	自行指定	自動配置
BOOL	M / SM 接點及位元操作的 D / L / X / Y 裝置	內部記憶體	M / X / Y 接點裝置	M 接點裝置
WORD	D / L / X / Y / E / SR 裝置	內部記憶體	D / E / F 裝置	D 裝置
DWORD	D / L / X / Y / E / SR 裝置	內部記憶體	D / E / F 裝置	D 裝置
LWORD	D / L / X / Y / E / SR 裝置 (只支援 AH)	內部記憶體	無此型態	
INT	D / L / X / Y 裝置	內部記憶體	無此型態	
DINT	D / L / X / Y 裝置	內部記憶體	無此型態	
LINT	D / L / X / Y 裝置 (只支援 AH)	內部記憶體	無此型態	
REAL	D / L / X / Y 裝置	內部記憶體	D 裝置	D 裝置
LREAL	D / L / X / Y 裝置 (只支援 AH)	內部記憶體	無此型態	
STRING	D / L / X / Y 裝置	內部記憶體	無此型態	
STEP	S 裝置 (只支援 AH5x0)	S 裝置	S 裝置	S 裝置
COUNTER	C / HC 裝置	C 裝置	C 裝置	C 裝置

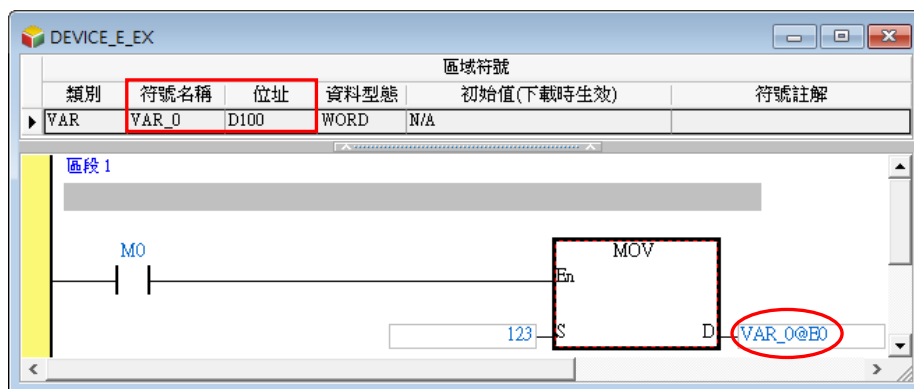
資料型態	AH/AS		DVP	
	自行指定	自動配置	自行指定	自動配置
TIMER	T 裝置	T 裝置	T 裝置	T 裝置
ARRAY	ARRAY 型態的變數符號會根據指定的陣列資料型態決定所配置的裝置類型，且會由自行指定或自動配置的啟始位址開始，連續配置符合陣列長度的裝置數量。 ARRAY 所指定的裝置不可為 SR / SM / E / F。			

資料型態	AS5xx/DVPxxMC	
	自行指定	自動配置
BOOL	I/Q/M 之 B 與 X	內部記憶體
BYTE	I/Q/M 之 B 裝置	內部記憶體
WORD	I/Q/M 之 W 裝置	內部記憶體
DWORD	I/Q/M 之 D 裝置	內部記憶體
LWORD	I/Q/M 之 L 裝置	內部記憶體
USINT	I/Q/M 之 B 裝置	內部記憶體
SINT	I/Q/M 之 B 裝置	內部記憶體
UINT	I/Q/M 之 W 裝置	內部記憶體
INT	I/Q/M 之 W 裝置	內部記憶體
UDINT	I/Q/M 之 D 裝置	內部記憶體
DINT	I/Q/M 之 D 裝置	內部記憶體
LINT	I/Q/M 之 L 裝置	內部記憶體
ULINT	I/Q/M 之 L 裝置	內部記憶體
REAL	I/Q/M 之 D 裝置	內部記憶體
LREAL	I/Q/M 之 L 裝置	內部記憶體
STRING	I/Q/M 之 B 裝置	內部記憶體
TIME	I/Q/M 之 L 裝置	內部記憶體
DATE	I/Q/M 之 D 裝置	內部記憶體
TOD	I/Q/M 之 D 裝置	內部記憶體
DT	I/Q/M 之 D 裝置	內部記憶體
ARRAY	I/Q/M 裝置，根據型態決定為何種裝置(B、W、D、L)。WORD Array 不支援 WORD[ArrayIndex].Bit 的指定位址寫法	內部記憶體

- *1. AH/AS 機種中，當變數符號的對應裝置位址被配置於內部記憶體時，使用者無法得知實際的位址值。
- *2. 關於 ARRAY 與 STRING 符號的相關注意事項，請參考本章 第 6.2.3 節 的相關說明。
- *3. 關於 DVP 機種自動配置範圍的設定方式與限制，請參考本章 第 6.2.11 節 的相關說明。
- *4. 功能塊型態的變數符號具有特殊意義，相關說明將於 第 7 章 - 功能塊 FB 與函式庫 當中另行介紹。
- *5. 位元操作指的是以 X0.0、Y0.1...等格式來表達字組中的某個位元，詳細說明請參考 附錄 B 的相關內容。

6.1.5 變數符號的間接指定暫存器修飾

ISPSOft 允許使用間接指定暫存器來修飾變數符號，其格式為 符號名稱@間接指定暫存器，其中間接指定暫存器可以是一個 E 或 F 裝置的編號，也可以是一個指定 E、F 裝置位址的變數符號。



間接指定暫存器內部儲存的資料，其意義代表被修飾對象的偏移值；以上圖為例，當間接指定暫存器 E0 的值為 2 時，則 VAR_0@E0 代表的便是 VAR_0 所指定的裝置 (D100) 再偏移 2 個位址，即裝置 D102，而當 M0 為 ON 時，被變更內容的對象便會是 D102。

6

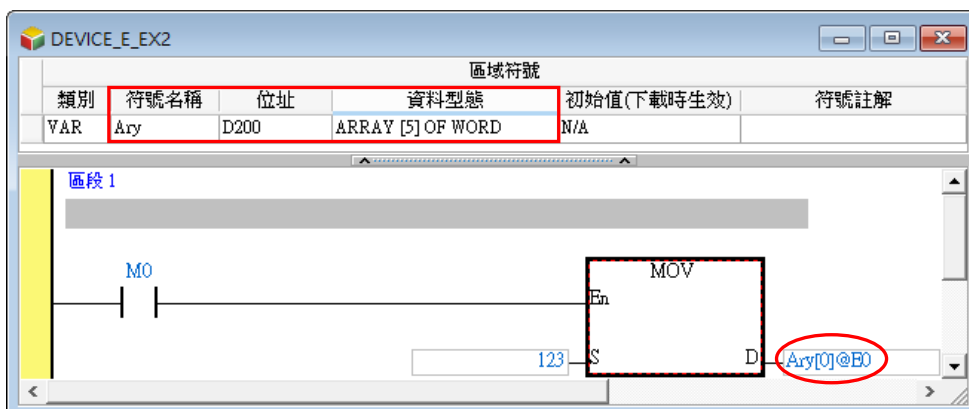
裝置名稱	狀態	值(16位元)	數值型態
E0	<input type="checkbox"/>	2	有號數十進制
M0	<input type="checkbox"/>		
D100	<input type="checkbox"/>	0	有號數十進制
D101	<input type="checkbox"/>	0	有號數十進制
D102	<input type="checkbox"/>	0	有號數十進制
D103	<input type="checkbox"/>	0	有號數十進制
D104	<input type="checkbox"/>	0	有號數十進制

➔

裝置名稱	狀態	值(16位元)	數值型態
E0	<input type="checkbox"/>	2	有號數十進制
M0	<input checked="" type="checkbox"/>		
D100	<input type="checkbox"/>	0	有號數十進制
D101	<input type="checkbox"/>	0	有號數十進制
D102	<input type="checkbox"/>	123	有號數十進制
D103	<input type="checkbox"/>	0	有號數十進制
D104	<input type="checkbox"/>	0	有號數十進制

此規則亦可套用於陣列型態的變數符號中，其格式為 陣列符號名稱[索引]@間接指定暫存器，其中索引值必須為常數格式，若索引值為變數符號的格式時，將不再允許使用間接指定暫存器來修飾。

如下圖，Ary 為一含 5 個元素的陣列，且其指定的啟始位址為 D200，而系統在編譯時便會自動為其配置 D200~D204 這個區間的裝置；因此若間接指定暫存器 E0 的值為 2，則 Ary[0]@E0 的操作對象便會是 Ary[0] 的位址 D200 再偏移 2 個位址，即裝置 D202，同時也是陣列元素 Ary[2] 的位址。但若再將間接指定暫存器 E0 的值為變更為 6，則 Ary[0]@E0 的操作對象便會是 D206，很明顯已經超出此陣列所涵蓋的範圍，不過這樣的使用仍是合法的，因為間接指定暫存器只負責忠實的進行位址偏移而已，所以若拿來作為陣列索引的用途時請特別小心。關於陣列的使用將於第 6.2.3 節另行介紹。



關於利用間接指定暫存器來修飾運算元的用法，在不同系列的機種上亦會有所差異，請參考下表。

應用場合	範例	AH/AS	DVP	AS5xx DVPxxMC
修飾指令運算元	MOV 100 VWD@E0	依指令規則而定	依指令規則而定	不支援
修飾接點或線圈	LD VBL_0@E0 OUT VBL_1@E1	支援	依機種而定	不支援
修飾陣列型態的變數符號	ARY[0]@E0	支援	支援	不支援

*1. AH/AS 系列的機種當中，間接指定暫存器只有 E 裝置，而 DVP 系列機種除了 E 裝置之外尚有 F 裝置可搭配使用，相關說明請參考各機種的操作手冊或技術文件。

*2. 間接指定暫存器不可再拿間接指定暫存器來修飾，例如 E0@E1，因此若拿間接指定暫存器來修飾變數符號時，請確認該符號的指定位址不可為 E 裝置或 F 裝置。

補充說明

- 間接指定暫存器的使用是針對某個裝置進行定址的偏移，但若所修飾的變數符號，其裝置位址是採用自動配置時，因使用者無法得知該變數符號會被隨機配置到哪個位址，因此貿然使用間接指定暫存器將可能造成程式執行的錯亂。
- 當經間接指定暫存器修飾後的裝置位址超過可用範圍時，PLC 的運作將可能造成錯誤，因此在使用上請務必小心；以 AHCPU530-EN 機種為例，使用 D65500@E0 為運算元，當 E0 的值超過 35 時，便會造成定址的範圍超過 D 裝置的可用範圍 D65535，而使得 PLC 進入錯誤狀態。
- 當間接指定暫存器的值被變更時，實際操作的裝置便會隨即變更，因此若原本操作的裝置沒有在程式的其他位置被使用到的話，該裝置的值便會保留在最後的狀態；以 AHCPU530-EN 機種為例，如下圖程式，當 E0 的值為 2 時，M0 的狀態會輸出給 M102，因此 M102 的狀態為 ON；但當 E0 的值由 2 變成 3 時，實際操作的裝置便會變更成 M103，此時 M103 便會為 ON，而因 M102 在程式中已經沒有其他程序對其進行操作，因此 M102 便會繼續保持在 ON 的狀態。

在 AH/AS 系列機種中，間接指定暫存器與位元操作可以混合使用，而當兩者混合使用時，「@」符號會有較高的優先權。請參考下列的範例，其中變數 VB 為 WORD 格式。

EX1：VB.1@E0 (VB 指定為 D100；E0=3)

解讀方式為先解讀 1@E0，因此 VB.1@E0 等同 VB.4，而實際的操作對象便會是 D100 的 Bit 4。

而 AH 系列機種另外還支援，以下的修飾方式。

EX1：VB@E0.1 (VB 指定為 D100；E0=3)

先解讀 VB@E0，因此內容會解讀為 D103.1，而實際的操作對象便會是 D103 的 Bit 1。

EX2：VB@E0.1@E1 (VB 指定為 D100；E0=3；E1=2)

解讀方式為將 VB@E0，與 1@E1 分開解讀。在此 VB@E0 等同於 D103，而 1@E1 則等於 3，因此整個敘述可解讀為 D103.3，所以實際的操作對象便會是 D103 的 Bit 3。

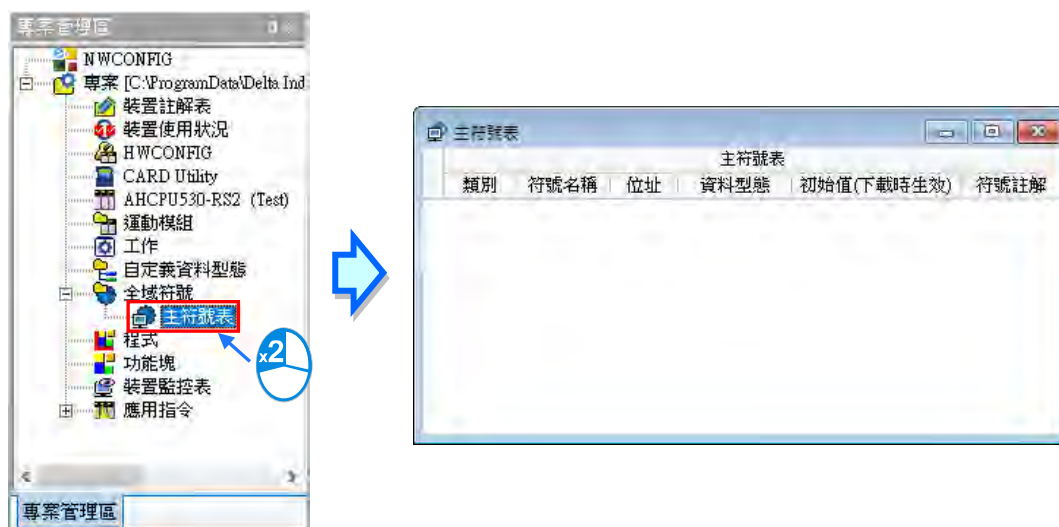
- *1. 使用位元操作時，系統僅會忠實的取得資料中的某個位元狀態，至於該位元狀態是否具有意義，則須由使用者自行判斷。
- *2. 在現階段的 ISPSOft 中，可使用的位元編號範圍只有 0~15，不過在未來新版的軟體中，系統將視變數符號的資料長度，而允許使用 0~31 或 0~63 的位元編號範圍。

6.2 ISPSOft 中的變數符號管理

6.2.1 變數符號表

● 全域符號表

於專案管理區的「全域符號」節點中「主符號表」項目上雙擊滑鼠左鍵，雙擊滑鼠左鍵主符號表便可開啟主符號表。

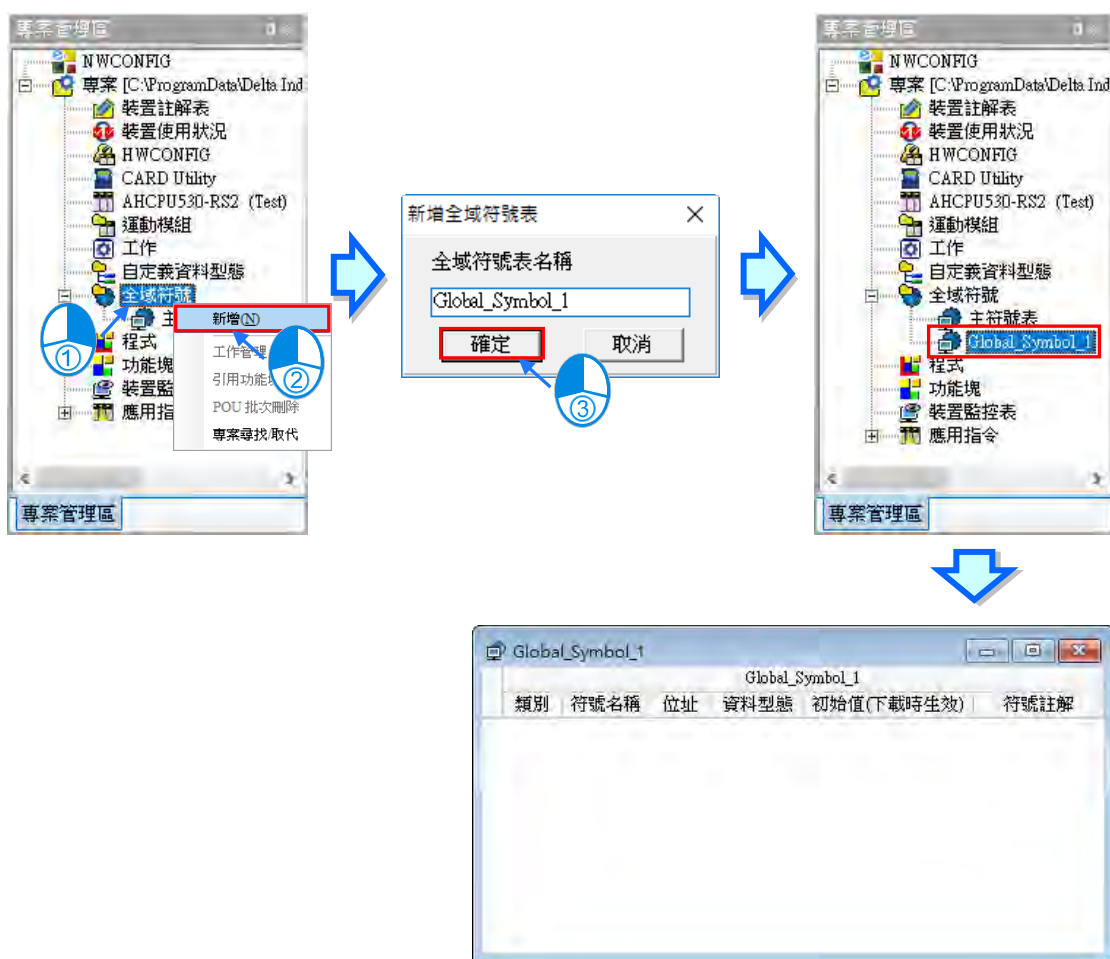


部分機種的全域符號表又依照功能的不同而分類了數個子項目，例如 AHxxEMC 系列機種的全域符號表可展開如下圖，主符號即表為一般的全域符號表，其他的符號表則為對應不同功能所產生的全域符號表，這些符號表中的符號在程式中也可如全域符號般被使用，但建立符號的方式請參閱各相關功能的說明章節內容。



於專案管理區的「全域符號」項目上雙擊滑鼠右鍵，接著點擊「新增(N)」及可新增自定義變數符號表，以便使用者方便管理全域變數，而新增完成後便會出現於「全域符號」下方節點。

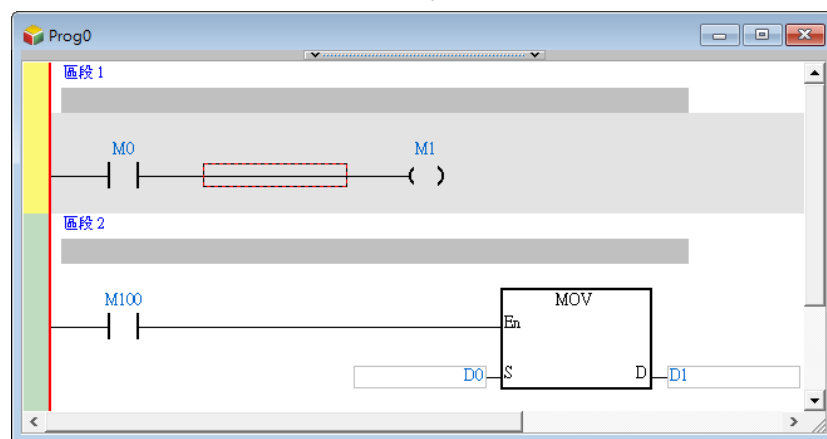
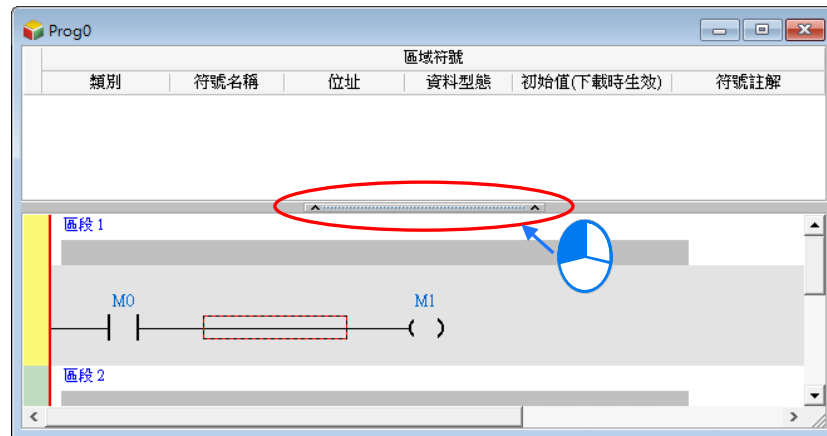
6



於專案管理區的「全域符號」下方節點中所新增的自定義變數符號表上點擊滑鼠右鍵，接著點擊「刪除」及可將其刪除，而刪除完成後便會消失於「全域符號」下方節點。

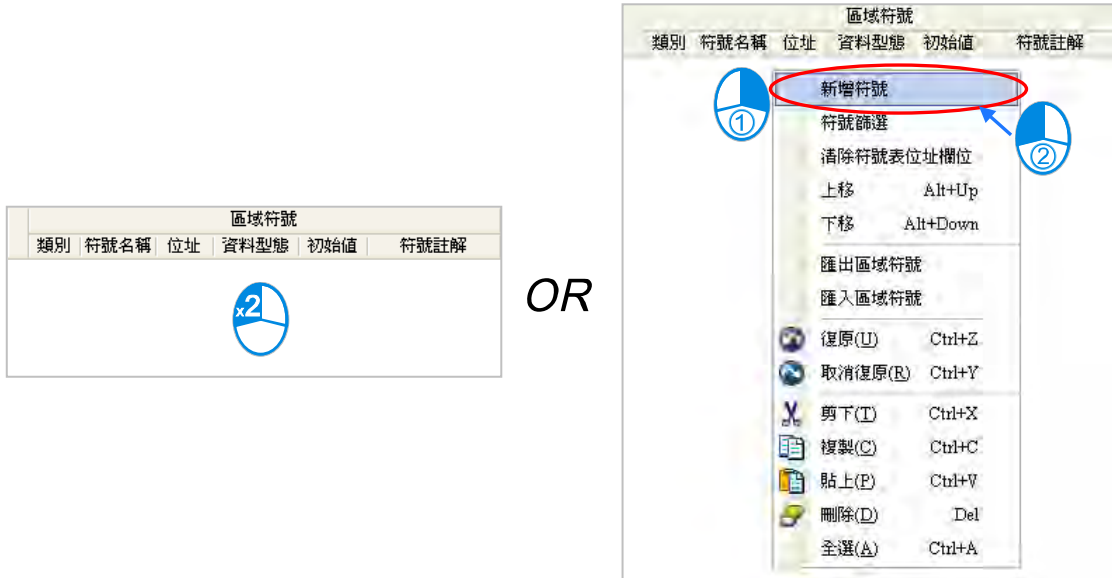
- 區域符號表

在 ISPSOft 中，區域符號表被安排在每個 POU 視窗的上方，而當用滑鼠點擊區域符號表下方的按鈕時，便可將區域符號表隱藏起來，再點擊一次便可重新開啟。

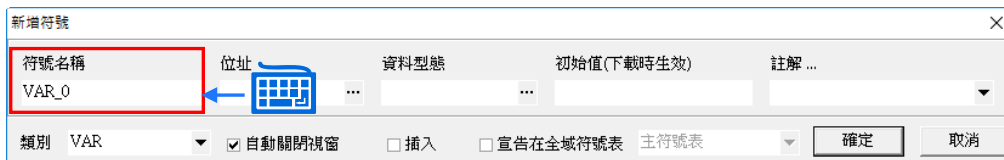


6.2.2 新增變數符號

- (1) 直接於符號表的空白處雙擊滑鼠左鍵，或於空白處點擊滑鼠右鍵後，於選單中點選「新增符號」。

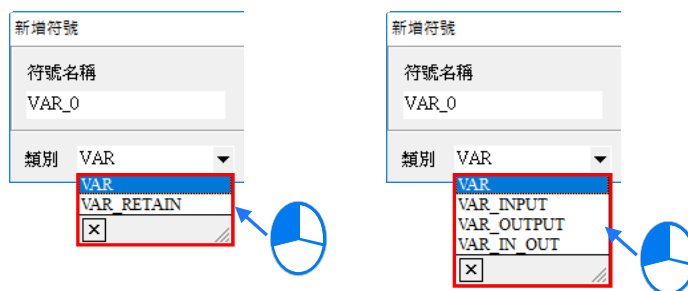


- (2) 執行新增符號後，畫面便會出現變數符號的宣告視窗。請先於「符號名稱」欄位輸入欲使用的變數符號名稱，而命名的規則請參考本章第 6.1.1 節的相關說明。

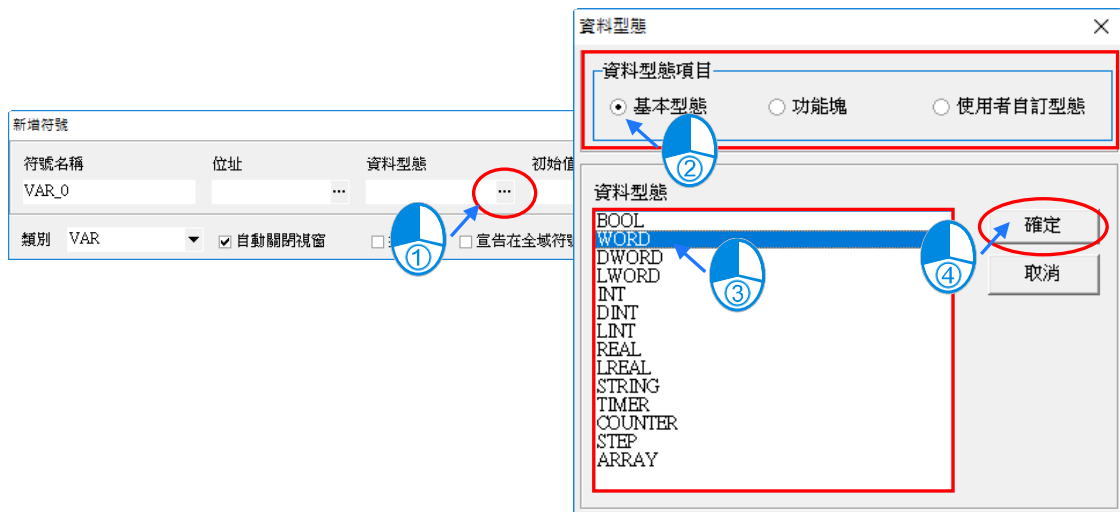


- *1. 隨著不同的宣告位置，如不同機種、全域/區域，出現的視窗亦會有所差異，不過基本的欄位與設定方式都是相同的。
*2. 在各種編程語言的環境中都有自動導出宣告視窗的方法，詳細說明請參考手冊第 8 章~第 12 章的相關內容。

- (3) 於「類別」欄位的下拉選單中選擇變數符號的類別，而隨著變數符號宣告的位置不同，該欄位可選擇的項目亦會有所不同。相關說明請參考本章第 6.1.2 節的內容。



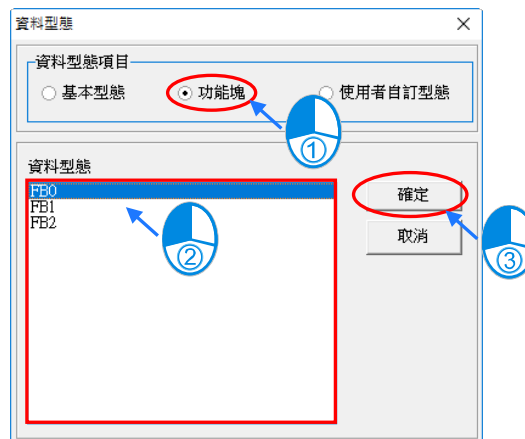
- (4) 按下「資料型態」欄位右側的按鈕，並於隨後彈出的視窗中設定變數符號的資料型態，完成後按下「確定」鍵即可。關於各種資料型態的說明請參考本章第 6.1.3 節的內容。



關於功能塊、使用者自訂型態、STRING 與 ARRAY 等幾種資料型態的設定，補充說明如下。

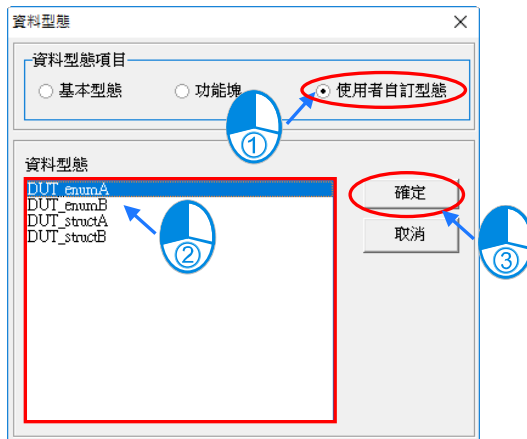
- 功能塊

於「資料型態項目」中選擇「功能塊」，之後再於清單中指定對應的功能塊定義。宣告為此資料型態的變數符號便代表宣告一個所選功能塊的實例，詳細說明請參考手冊第 7 章的相關內容。



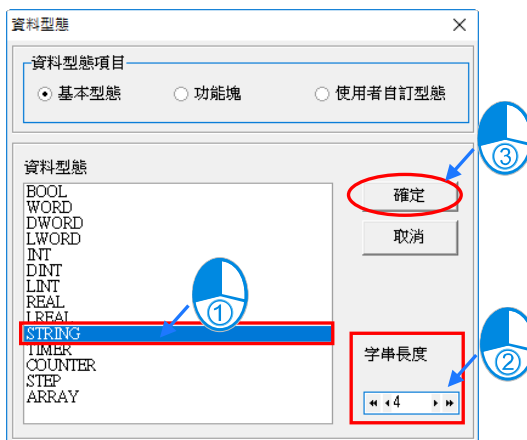
● 使用者自訂型態

於「資料型態項目」中選擇「使用者自訂型態」，之後再於清單中指定已宣告的 DUT。宣告為此資料型態的變數符號，詳細說明請參考手冊第 8 章的相關內容。



● STRING

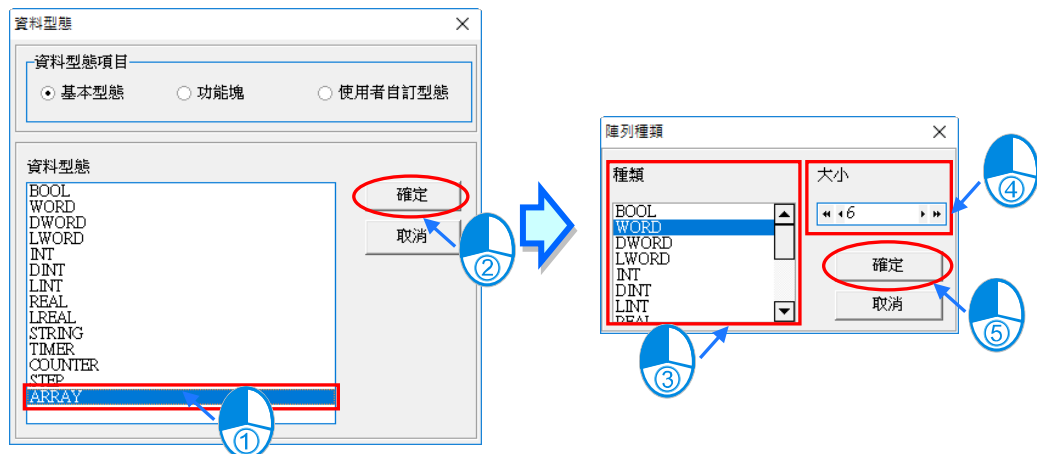
若選擇的資料型態為 STRING，則請一併於視窗的右下角設定字串的最大長度，其範圍為 1~128，而所設定的值則代表此字串符號最大可容納的字元數。設定時以滑鼠點擊 ◀ 或 ▶ 按鈕便可遞減或遞增設定值的個位數，而點擊 ◀◀ 或 ▶▶ 按鈕便可遞減或遞增設定值的十位數。



6

● ARRAY

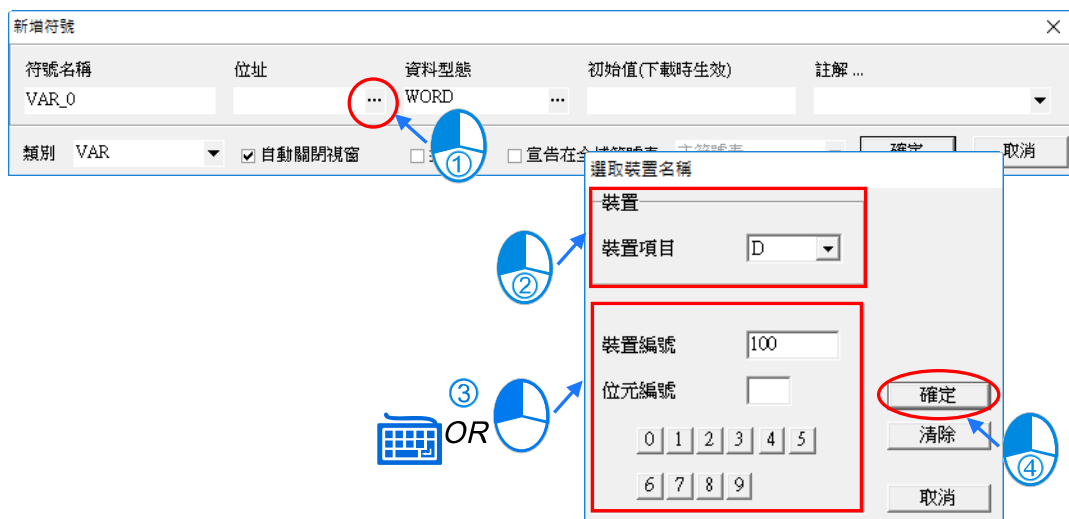
若選擇的資料型態為 ARRAY，則按下確定後便會出現 ARRAY 的設定視窗，請於此視窗中一併設定 ARRAY 的種類與大小；而其中陣列大小的範圍為 1~512，設定時以滑鼠點擊 ◀ 或 ▶ 按鈕便可遞減或遞增設定值的個位數，而點擊 ◀◀ 或 ▶▶ 按鈕便可遞減或遞增設定值的十位數。



- (5) 若「位址」欄位保持空白，則代表此變數符號的裝置位址將交由系統自動配置；而若要指定裝置位址時，可直接於「位址」欄位輸入裝置位址或點擊該欄位右側的按鈕，並於隨後彈出的視窗中指定使用裝置。相關說明請參考本章第 6.1.4 節的內容。



OR



*1. 當選擇的資料型態為功能塊時，「位址」欄位將無法進行設定。

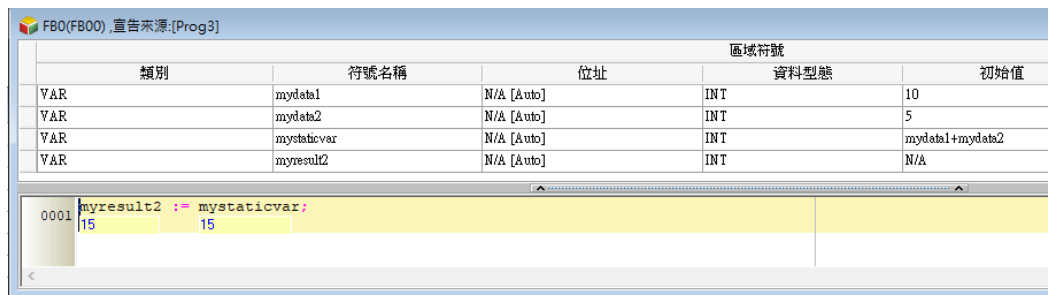
*2. 若是支援可使用位元操作機種，而此裝置設定視窗便會有「位元編號」的欄位，填入編號可以表示，如 X0.0、D0.1...

- (6) 若須為此變數符號設定初始值時，請於「初始值」欄位中直接輸入即可，而所設定的值必須符合變數符號的資料型態及大小；且當變數符號的資料型態為 **ARRAY** 時，點擊「初始值」欄位後會出現一個初始值的設定視窗，之後便可於該視窗中逐一設定各個陣列元素的初始值。

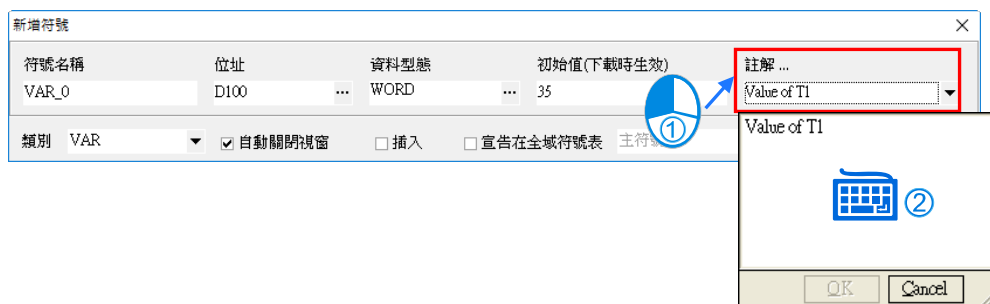
BOOL 符號的初始值為 **TRUE** 或 **FALSE**，且在輸入時亦可簡略的輸入 **T** 或 **F**，或者直接輸入 **1** 或 **0**，系統將會在輸入完畢之後自動進行轉換；而在輸入字串符號的初始值時，其字元數不得超過所設定的字串長度，且亦無須為該字串加上雙引號，否則加上的雙引號將會被視為字串內容的一部分，若是選擇 **DVPxxMC** 與 **AS5xx** 機種，在輸入字串符號的初始值時，需要對該字串的頭尾加上單引號的符號；若符號的資料型態為功能塊或使用者自訂形態時，「初始值」欄位將不可設定。



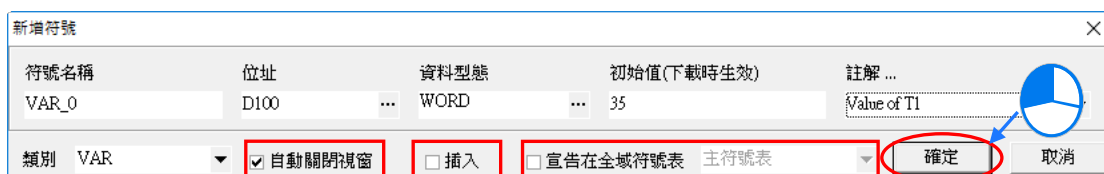
在 **DVPxxMC** 與 **AS5xx** 的初始值支援 **ST** 運算式的寫法，如下圖。



- (7) 欲加入註解時，請先於「註解」欄位點擊滑鼠左鍵，接著便可於出現的下拉頁面中輸入註解。



- (8) 宣告視窗中尚有三個可勾選的選項，其功能請參考下述。而確認所有的設定後，按下「確定」鍵即可完成一個變數符號的新增。



- 自動關閉視窗

若取消此項的勾選，則按下確定後，宣告視窗便不會自動關閉，如此即可方便使用者連續宣告變數符號。而其預設值的設定方式請參考第 2.3.1 節中的「系統基本設定 - 符號表」項目。

- 插入

當勾選此項時，此處宣告的變數符號便會新增在符號表被選擇的變數符號上方；未勾選時，則被新增到符號表最後一列。

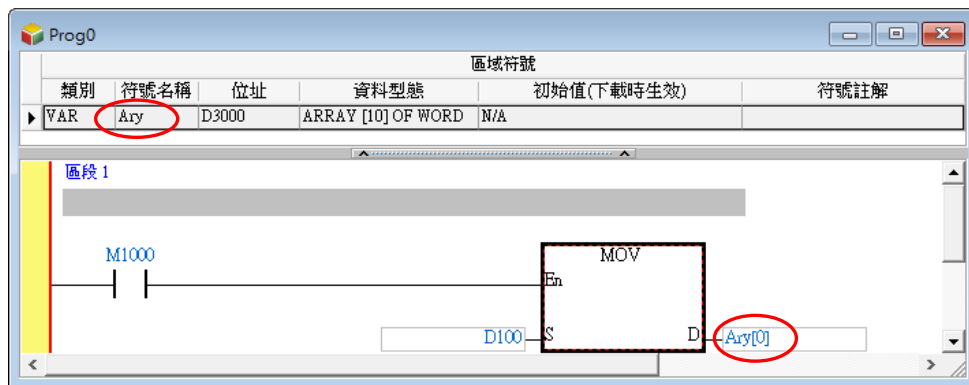
- 宣告在全域符號表

當勾選此項時，右方下拉選單會出現主符號表及自定義全域符號表，此處宣告的變數符號便會自動被宣告在所選擇的主符號表或自定義全域符號表中；而若一開始開啟宣告視窗的位置就是在主符號表或自定義全域符號表時，此選項便不會出現。

6.2.3 陣列與字串型態符號的使用原則

- 陣列型態的變數符號

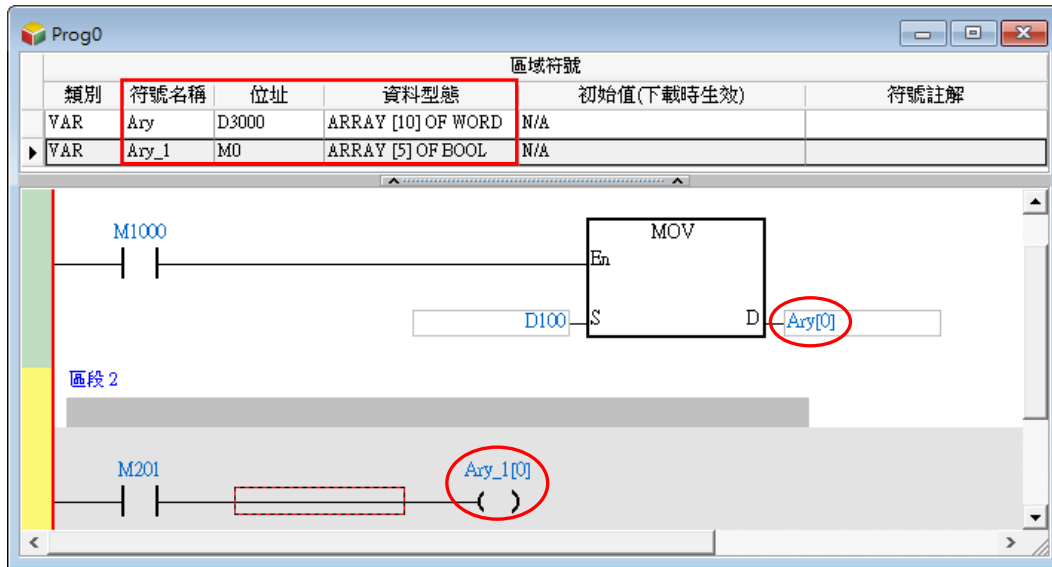
如下圖範例，當要在程式當中存取陣列型態的變數符號時，其表達的格式為 **符號名稱[索引]**，而在一般的狀況下，**索引值**只可為 10 進制常數，但若是 AH/AS 系列機種的專案，且選擇的為 ST 語言時，**索引值**的部份則可為變數符號；而不不論是常數或變數，**索引值**的內容必須由 0 開始，且不可大於或等於陣列的大小，例如大小為 10 的陣列符號，其**索引值**的範圍則必須為 0~9；此外，當使用變數做為**索引值**時，請小心避免變數的內容值超出**索引值**的合法範圍，否則將可能在運作時造成資料的錯亂。



宣告陣列型態的變數符號時，除了須指定陣列元素的資料型態之外，亦須一併指定陣列的大小，範圍則為 1~512；且在自行指定陣列符號的啟始位址時，須根據陣列的資料型態而指定對應的裝置，但須注意不可指定為 SR、SM、E、F...等裝置，關於各種資料型態所對應的裝置類別請參考第 6.1.4 節的相關說明；而在配置陣列符號所使用的位址區間時，不論是自行指定或自動配置，皆會由啟始的位址開始，連續配置符合陣列長度的裝置數量。

如下圖，其中 **Ary_0** 的「資料型態」欄位為 **ARRAY[3] OF DWORD**，代表該陣列的大小為 3，亦即包含 3 個元素，且其元素的資料型態為 DWORD，而由「位址」欄位可知所指定的啟始位址為 D0，因此該陣列將會佔用 D0~D5 共 6 個 WORD (3 個 DWORD) 的裝置位址。

接著再看 **Ary_1** 的陣列符號，其「資料型態」欄位為 **ARRAY[5] OF BOOL**，代表該陣列的大小為 5，且元素的資料型態為 **BOOL**，而由「位址」欄位可知所指定的啟始位址為 **M0**，因此該陣列將會佔用 **M0~M4** 共 5 個 M 裝置；若是使用 **DVPxxMC** 與 **AS5xx** 機種，「資料型態」欄位為 **ARRAY[10] OF BOOL**，位址指定 **%IB0**，則代表該陣列會占用到 **%IB0 ~ %IB9** 裝置，共 10 個 **%IB** 裝置。



在宣告陣列型態的變數符號時可各別為每個元素定義其初始值，請參考下列範例。

A_Ary 的「初始值」欄位顯示為 **[1,2,3,4,5]**，代表陣列元素 **A_Ary[0]** 的初始值為 1，**A_Ary[1]** 的初始值為 2，其他依此類推；而 **B_Ary** 的「初始值」欄位則顯示為 **[1,3(0),5]**，其中的第二個初始值為 **3(0)**，代表由第二個元素開始，有連續 3 個元素的初始值皆為 0。

6

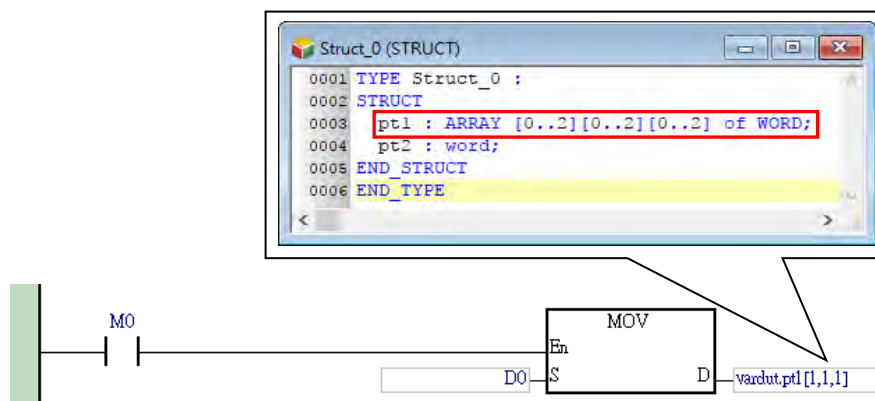
區域符號					
類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	A_Ary	D7000	ARRAY [5] OF WORD	[1,2,3,4,5]	
VAR	B_Ary	D7005	ARRAY [5] OF WORD	[1,3(0),5]	

A_Ary		B_Ary	
元素	初始值	元素	初始值
A_Ary[0]	1	B_Ary[0]	1
A_Ary[1]	2	B_Ary[1]	0
A_Ary[2]	3	B_Ary[2]	0
A_Ary[3]	4	B_Ary[3]	0
A_Ary[4]	5	B_Ary[4]	5

在 DVPxxMC 與 AS5xx 的機種支援多維陣列資料型態，可定義最多三維陣列，於宣告的時候可以在陣列種類視窗選取陣列大小輸入，如下圖。



另外，當自定義資料型態中具有二維或三維陣列的元素時，此類型的變數符號將可使用[n,n]或[n,n,n]加以指定。如下圖自定義資料型態中的元素 pt1 為三維陣列，則程式中的變數符號 vardut 可以 vardut.pt1[1,1,1]的方式指定陣列中的第[1,1,1]個陣列元素。



● 字串型態的變數符號

ISPSOft 中的字串是以 ASCII 格式來表達，而字串常數的表達格式則是加上雙引號，例如 "ABCD"，而 DVPxxMC 與 AS5xx 機種則是以單引號，例如 'ABCD'；且字串型態的變數符號通常會搭配 API 應用指令來進行操作，相關的使用規則請參考各機種的應用指令手冊或相關技術文件。

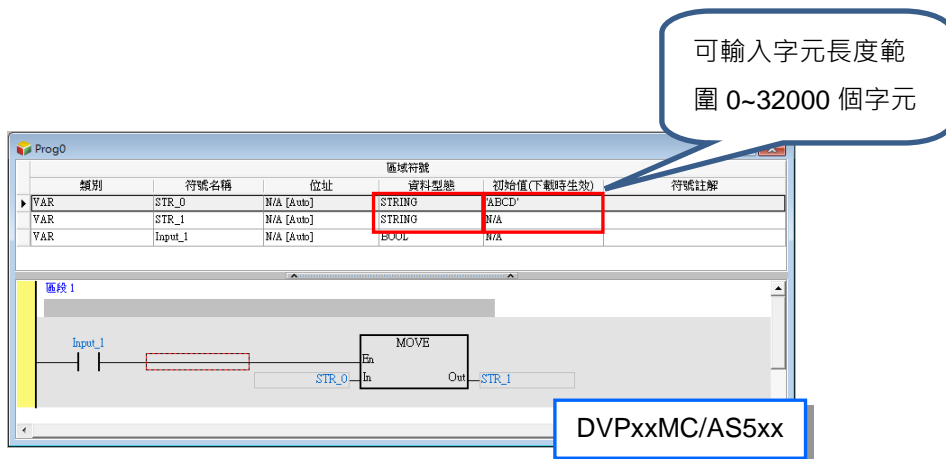
使用者在宣告字串型態的符號時便須一併指定字串的最大長度，指定的範圍為 1~128，其中 1 個字元就會佔用 1 個 Byte 的空間；而為其配置位址時，除了配置符合最大長度的空間之外，尚須多配置一個 Byte 的空間以儲存字串的結束字元，且若配置的最後一段空間未滿 1 個 WORD 時，便須補滿 1 個 WORD；例如 3 個字元的字串須配置 2 個 WORD 的空間 (4 Bytes)，而 4 或 5 個字元的字串則須配置 3 個 WORD (6 Bytes)。

請參考下圖所宣告的字串符號，其中在資料型態 **STRING** 後的括號中會顯示字串的最大長度；指定字串符號的初始值時無須加上雙引號，且初始值的字串長度必須小於或等於宣告的最大長度。



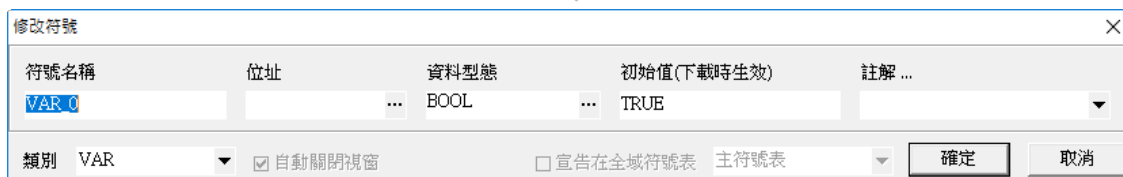
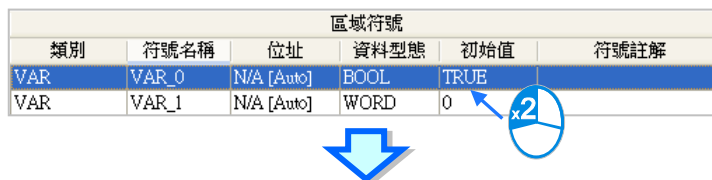
若使用者選擇的是 DVPxxMC 與 AS5xx 機種，宣告字串時長度由軟體配置使用者無需調整，只有在指定初始值時，初始值字元長度為 0~32000 個字元，在宣告完後，其資料型態 **STRING** 後的括號中不會顯示字串的最大長度。

6

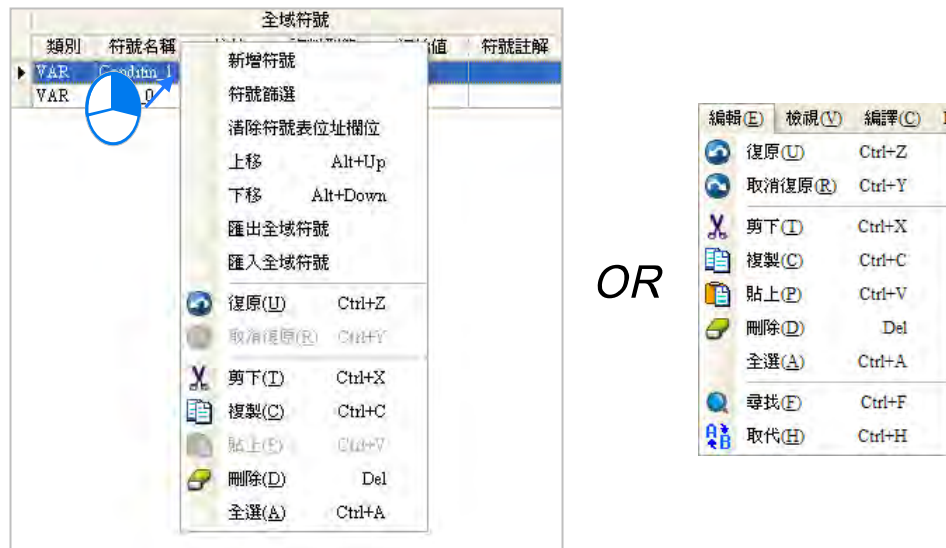


6.2.4 變數符號的修改與符號表編輯

若要修改變數符號本身的屬性時，請直接於欲修改的變數符號上雙擊滑鼠左鍵，之後便可再次開啟該變數符號的宣告視窗，接著參考本章第 6.2.2 節的說明修改各欄位的設定值即可。



當要對符號表中的項目進行編輯時，使用者可透過功能工具列中的「編輯」功能項或符號表的右鍵快捷選單來進行操作。而其中各種基本操作的相關說明請參考下表。



OR

操作項目	功能說明
復原	復原方才的編輯動作（*最多可復原 20 個步驟）
取消復原	取消復原而重複方才的動作
剪下	剪下選取的變數符號
複製	複製選取的變數符號
貼上	將方才複製或剪下的變數符號貼在目前編輯的符號表中
刪除	刪除選取的變數符號
全選	選取符號表中所有的變數符號

*1. 在不同機種或 POU 之間進行變數符號的 複製/剪下/貼上 操作時，請務必確認貼上後的變數符號屬性是否正確。

*2. AH/AS 系列機種與 DVP 系列機種與 DVPxxMC/AS5xx 機種，三者之間的變數符號皆不可互相進行複製/剪下/貼上的操作。

*3. 若貼上的符號表中已有相同名稱的符號時，系統會自動在貼上的符號名稱後方加上 CopyOf_n (n 為流水號) 做為新名稱。

6.2.5 清除變數符號的配置位址

不論是使用者自行指定或是系統自動配置的裝置位址，執行此功能便可將符號表中的所有裝置位址全部清除，而之後再次執行編譯時，系統便會自動為其重新配置。另外，此功能僅會清除目前點選的符號表中的裝置位址，而不會影響其他的符號表。

執行時請先於符號表中點擊滑鼠右鍵後，接著於快捷選單中點選「清除符號表位址欄位」即可。

主符號表

類別	符號名稱	位址	資料...	初始值(下載時生效)	符號註解
VAR	IN_0	X0.0	BOOL	N/A	
VAR	IN_1	X0.1			
VAR	IN_2	X0.2			
VAR	OUT_0	Y0.0			
VAR	OUT_1	Y0.1			

快捷選單:

- 新增符號
- 符號篩選
- 清除符號表位址欄位
- 上移 Alt+Up
- 下移 Alt+Down
- 匯出全域符號
- 匯入全域符號
- 復原(U) Ctrl+Z
- 取消復原(B) Ctrl+Y
- 剪下(T) Ctrl+X
- 複製(C) Ctrl+C
- 貼上(P) Ctrl+V
- 刪除(D) Del
- 全選(A) Ctrl+A

↓


主符號表

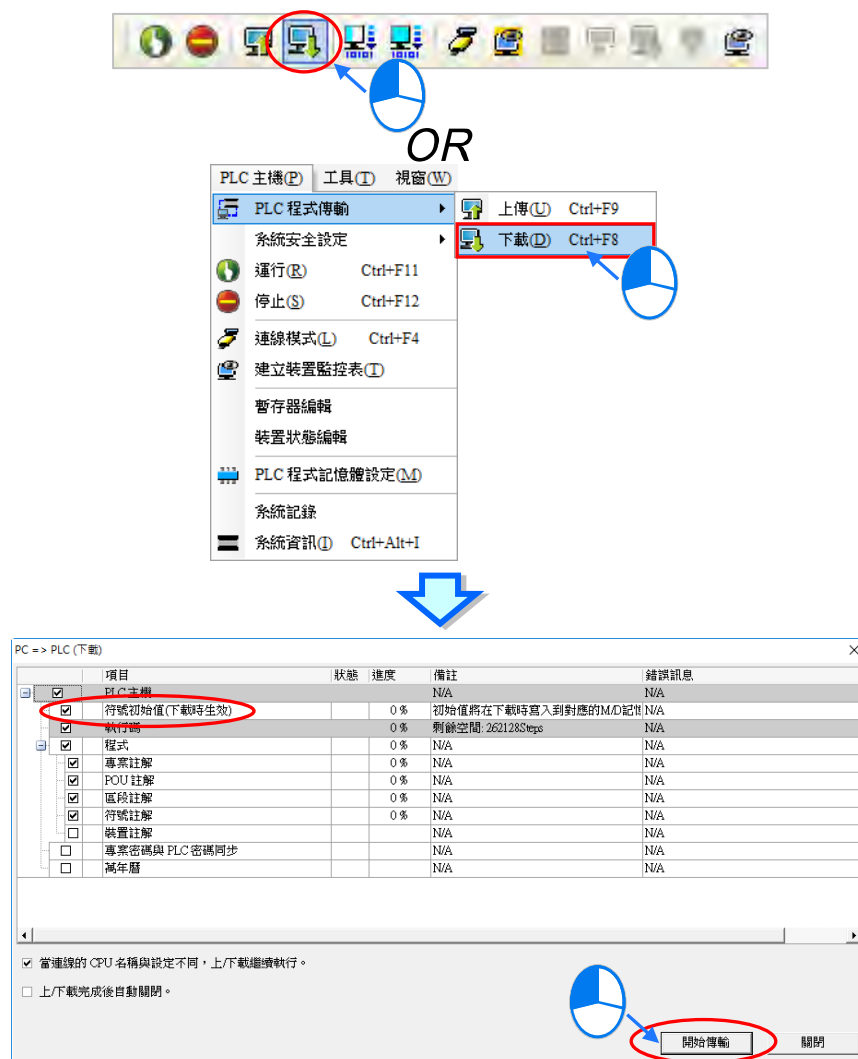
類別	符號名稱	位址	資料...	初始值(下載時生效)	符號註解
VAR	IN_0	N/A [Auto]	BOOL	N/A	
VAR	IN_1	N/A [Auto]	BOOL	N/A	
VAR	IN_2	N/A [Auto]	BOOL	N/A	
VAR	OUT_0	N/A [Auto]	BOOL	N/A	
VAR	OUT_1	N/A [Auto]	BOOL	N/A	

6

6.2.6 下載變數符號的初始值

每個變數符號在宣告時都可為其設定一個**初始值**，而在下載專案程式時便可選擇是否將**初始值**一併寫入該變數符號所對應的裝置位址中。不過初始值的寫入動作僅會在每次下載時才執行一次，而後續如主機斷電或是 RUN/STOP 切換後，再次進入 RUN 模式的這個階段，變數符號的初始值無法回復原設定；因此對於須確保每次進入執行狀態必須初始化的變數符號，建議還是於程式中加入初始值的初始化程序。執行線上更新的動作時，系統也不會重新下載變數符號的初始值。

欲執行時，請於功能工具列中點選 **PLC 程式傳輸 > 下載 (D)** 或點擊圖示工具列的  圖示，接著勾選「傳輸選項」的「符號初始值 (下載時生效)」項目後，按下「傳送」即可將符號初始值寫入 PLC 當中。



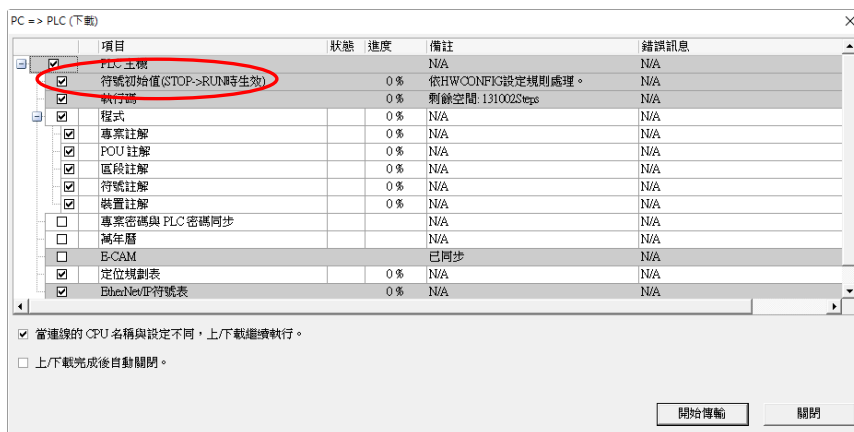
*.進行下載前請先確認 ISPSOft 已可與主機正常進行連線。

部分機種的新韌體支援新的初始值方式請參考下表，新的初始值方式提供 **Stop → Run 非停電保持符號初始值生效**、**Stop → Run 停電保持符號初始值生效**與**停電保持符號初始值生效**三種功能，使用 **Stop → Run 停電保持與非停電保持初始值生效**功能，請於下載專案程式時先參考第 3.1.4 節在 HWCONFIG 設定 **Stop → Run 停電保持與非停電保持初始值生效**的設定並下載至 PLC 主機。

機種	韌體版本	初始值表最大容量
AH5x1-RS2	1.05.00	80K Bytes
AH5x1-EN	2.03.00	80K Bytes
AH560-EN2	1.02.00	80K Bytes
AS	1.06.60	20K Bytes
ES3	0.40.10	20K Bytes
AHxxEMC	2.03	80K Bytes
AS 模擬器	1.06.60	20K Bytes

ISPSOft 也會自動偵測連接裝置是否支援新的初始值方式，若連接的為支援新的初始值機種，則在下載選單中「傳輸選項」的「符號初始值 (Stop → Run 時生效)」項目會變成無法取消勾選，如下圖。

6



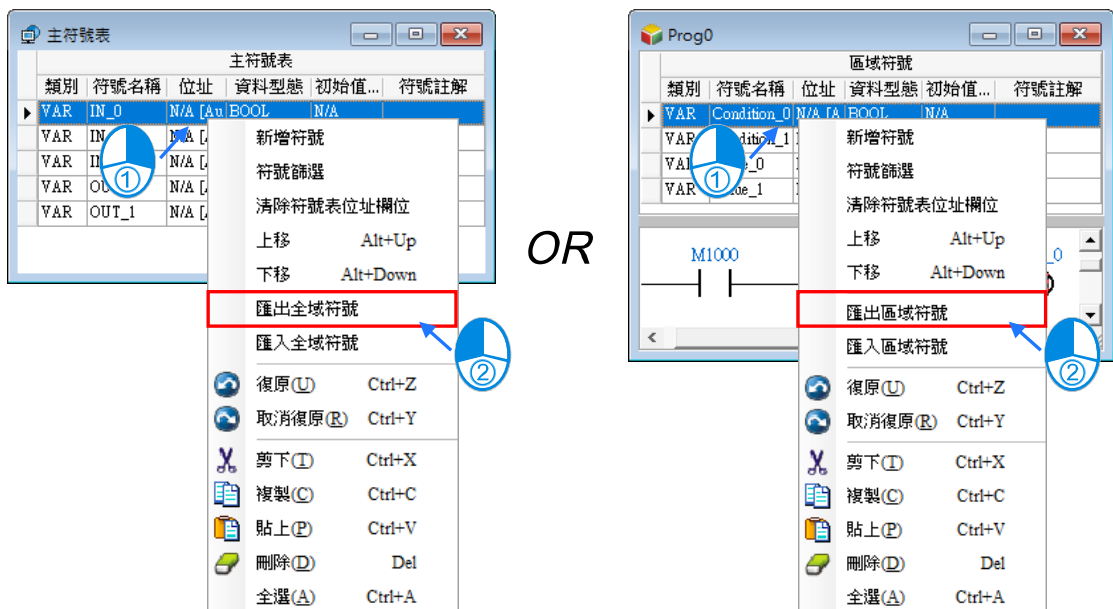
下載完成初始值表之後點選 **PLC 主機 (P)** > **初始值設定 (I)** > **停電保持符號初始值生效 (R)**，就可以使用單次停電保持符號初始值生效的動作，必須確認 ISPSoft 已可與主機正常進行連線，如下圖。



6.2.7 符號表的匯出

於專案中所建立的變數符號亦可被匯出為 CSV 檔，而之後使用者便可直接透過 EXCEL 來開啟，並對匯出的資料進行後續的編輯與管理。

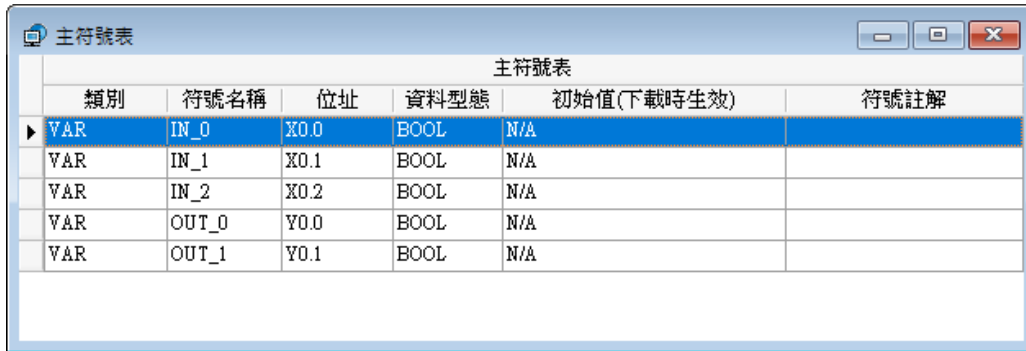
欲匯出變數符號時，請直接於符號表中按下滑鼠右鍵，接著於快捷選單中點選「匯出全域符號」或「匯出區域符號」的項目即可。



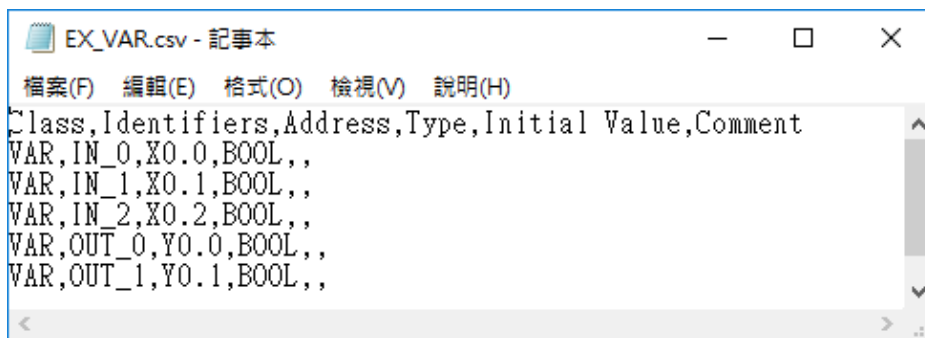
而除了透過右鍵的快捷選單之外，在匯出全域符號表時，亦可由功能工具列點選**檔案 (F)** > **匯出** > **全域符號表 (L)** 或在專案管理區選取要匯出的全域符號表點擊右鍵匯出，每次則只能針對目前選取的全域符號表進行操作。

6.2.8 符號表的匯入

建立變數符號時，除了可於 ISPSOft 的符號表當中逐一進行宣告之外，使用者亦可先將符號表匯出，接著再透過 EXCEL 直接開啟，並依據匯出的格式進行編輯，而編輯完畢之後，只要再將其重新匯入 ISPSOft 的符號表中即可，如此便可輕鬆地建立大量的變數符號。下列便是匯出後的符號表格式。



主符號表					
類別	符號名稱	位址	資料型態	初始值(下載時生效)	符號註解
VAR	IN_0	X0.0	BOOL	N/A	
VAR	IN_1	X0.1	BOOL	N/A	
VAR	IN_2	X0.2	BOOL	N/A	
VAR	OUT_0	Y0.0	BOOL	N/A	
VAR	OUT_1	Y0.1	BOOL	N/A	

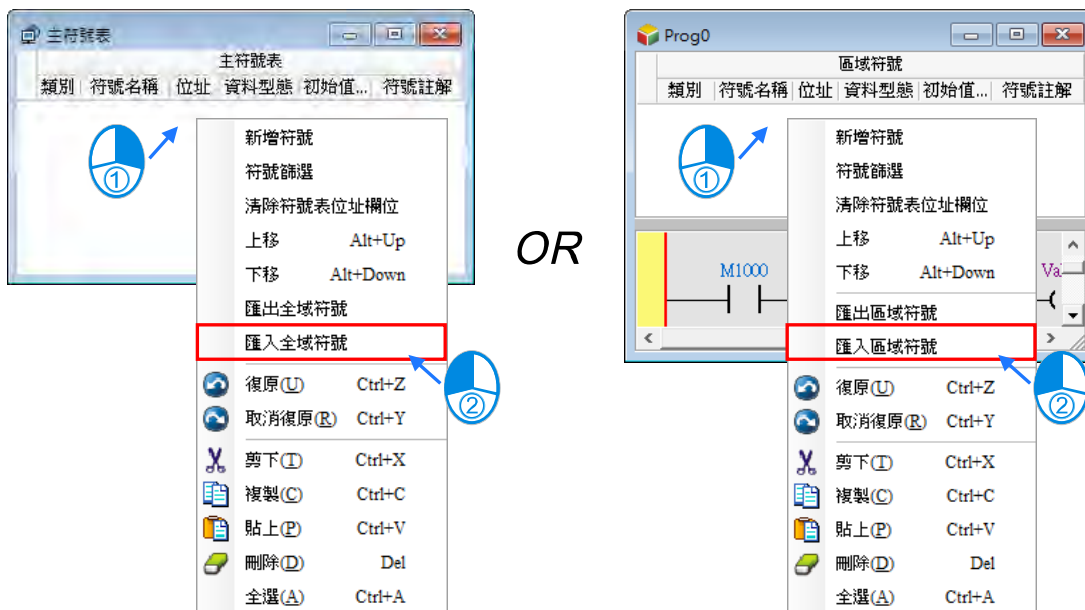
```

Class,Identifiers,Address,Type,Initial Value,Comment
VAR,IN_0,X0.0,BOOL,,
VAR,IN_1,X0.1,BOOL,,
VAR,IN_2,X0.2,BOOL,,
VAR,OUT_0,Y0.0,BOOL,,
VAR,OUT_1,Y0.1,BOOL,,

```

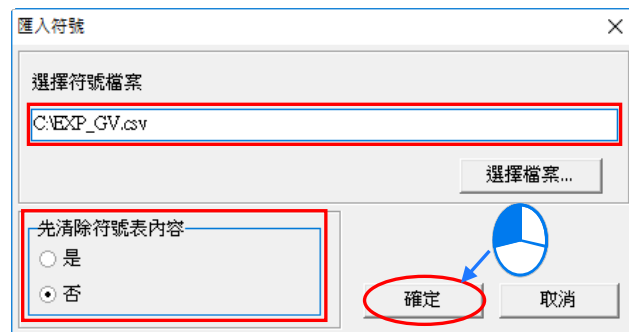
6

欲匯入變數符號時，請直接於符號表中按下滑鼠右鍵，接著於快捷選單中點選「匯入全域符號」或「匯入區域符號」的項目即可。



而除了透過右鍵的快捷選單之外，在匯入全域符號表時，亦可由功能工具列點選**檔案 (F) > 匯入 > 全域符號表 (L)** 或在專案管理區選取要匯入的全域符號表點擊右鍵匯入，每次則只能針對目前選取的全域符號表進行操作。

依據上述的方式執行之後，接著請於彈出的視窗中選擇欲匯入的檔案，並於左下角的欄位中設定在匯入之前，是否要先清除該符號表中所有已存在的符號項目，完成後按下「**確定**」鍵即可。

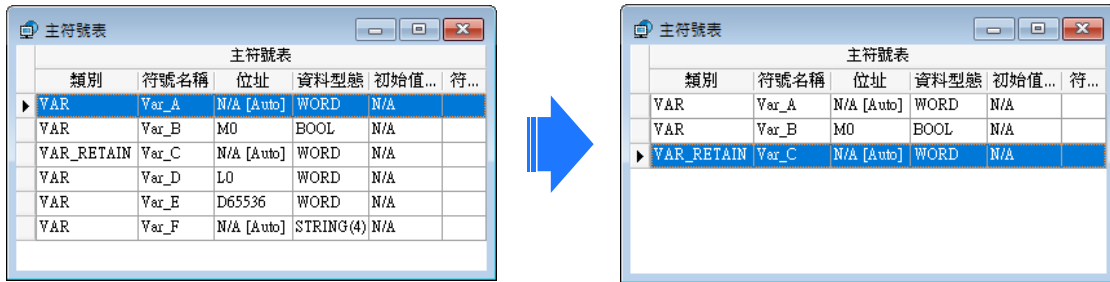


補充說明

變數符號可在不同的機種或 POU 之間進行匯出與匯入，但目前 DVPxxMC 與 AS5xx 機種不支援與其他機種的變數符號做匯出與匯入動作，而若準備匯入的檔案中有存在以下狀態的變數符號時，該變數符號將被自動排除或轉換，而其他的變數符號則可正常被匯入；當匯入的過程中有出現異常時，於專案的路徑下便會產生一個記錄檔，其檔名為「**專案名稱_ErrorLog**」，檢視該檔便可知匯入過程中所發生的問題。

- (a) 當接受匯入的符號表中已存在相同名稱的變數符號時，檔案中重複名稱的變數符號將自動被排除，但若選擇在匯入前先清除符號表內容則不會有此問題。
- (b) 當檔案中的變數符號，其資料型態為被匯入機種所不支援的型態時，該變數符號將被排除。
- (c) 檔案中的變數符號有指定裝置位址，但該裝置為被匯入機種不支援的裝置種類，或裝置位址超過被匯入機種的裝置定址範圍時，該變數符號將被排除。
- (d) 在不同 POU 或全域符號之間進行匯出匯入的操作時，若欲匯入的符號類別，如 VAR_INPUT、VAR_OUTPUT...，與被匯入的符號表不相容時，該項目將被排除。
- (e) 將 AH/AS 系列機種的全域符號表匯入 DVP 系列機種時，若變數符號的類別為 VAR_RETAIN，則匯入後，所有 VAR_RETAIN 的類別將會被視為 VAR 類別。

下列為將 AHCPU530-EN 專案的全域符號表匯入至 SV 機種專案的範例，其中符號 Var_A、Var_B、Var_C 皆可被匯入，但符號 Var_C 的 VAR_RETAIN 類別將被自動視為 VAR 類別；而變數符號 Var_D 則因為 SV 機種不支援 L 裝置而被排除；另外，變數符號 Var_E 則是因為 D65536 已超過 SV 機種的定址範圍，而變數符號 Var_F 則是因為 SV 機種不支援 STRING 型態的變數符號，因此也都會被自動排除。



除了須注意符號表之間的資料格式之外，在符號註解的部份亦須注意下列事項。

下圖是分別以 EXCEL 與記事本來開啟的同一個檔案，而由當中可發現，實際 CSV 檔的格式是以逗點來區隔每個欄位，因此當我們在符號表的註解欄位中使用逗點時，系統便會自動把逗點之後的內容視為其他欄位，如下圖紅框中的 C2 與 C3，而重新匯入時，逗點之後的內容便會因此而遺失。

6

	A	B	C	D	E	F	G	H
1	Class	Identifiers	Address	Type	Initial Value	Comment		
2	VAR	IN_0	X0.0	BOOL	FALSE	C1	C2	C3
3	VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3		

```

class,Identifiers,Address,Type,Initial Value,Comment
VAR,IN_0,X0.0,BOOL,FALSE,C1, C2, C3
VAR,IN_1,X0.1,BOOL,FALSE,C1 / C2 / C3
    
```

● 匯出前

全域符號						
類別	符號名稱	位址	資料型態	初始值	符號註解	
VAR	IN_0	X0.0	BOOL	FALSE	C1, C2, C3	
VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3	



● 重新匯入後

全域符號						
類別	符號名稱	位址	資料型態	初始值	符號註解	
VAR	IN_0	X0.0	BOOL	FALSE	C1	
VAR	IN_1	X0.1	BOOL	FALSE	C1 / C2 / C3	

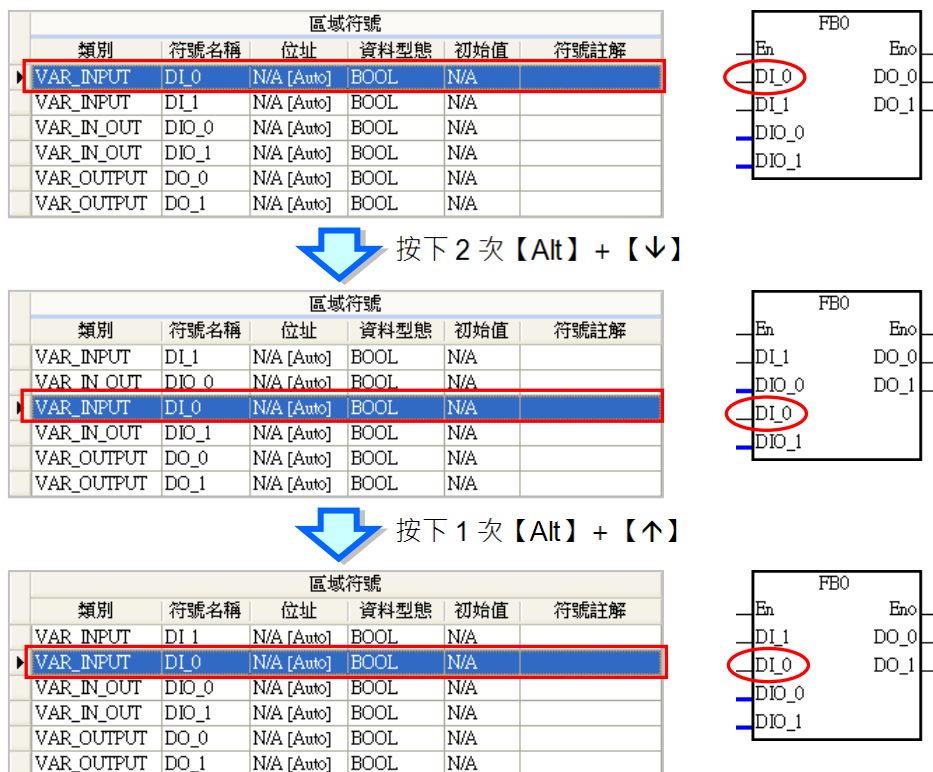
6.2.9 符號表的排序

不論是全域或區域符號表，當要調整符號的排序時，可以直接在符號表上用右鍵點選要移動的符號，再於快捷選單上點選上移或下移，移動指定的符號。



也可於符號表中點選欲調整的變數符號，之後利用鍵盤的【Alt】 + 【↑】或【Alt】 + 【↓】鍵調整該變數符號的上下位置。

在功能塊的區域符號表中，VAR_INPUT、VAR_IN_OUT 及 VAR_OUTPUT 等類別的變數符號，其排列順序將會影響功能塊端點的運算元順序。



在全域符號表及程式 POU 的區域符號表中，於「符號名稱」或「位址」的欄位名稱上點擊滑鼠左鍵後，符號表中的變數符號便會以該欄位為參考值而重新進行排序；而此時該欄位名稱上也會出現排序方向的符號，再次點擊欄位名稱則可改變排序的方向。而如同上述，功能塊區域符號的排列將會影響功能塊端點的運算元次序，因此功能塊的區域符號表將無法使用此種方式進行排序。

類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	OUT_1	N/A [Auto]	BOOL	FALSE	
VAR	OUT_0	N/A [Auto]	BOOL	FALSE	
VAR	IN_2	N/A [Auto]	BOOL	FALSE	
VAR	IN_1	N/A [Auto]	BOOL	FALSE	
VAR	IN_0	N/A [Auto]	BOOL	FALSE	

類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	IN_0	N/A [Auto]	BOOL	FALSE	
VAR	IN_1	N/A [Auto]	BOOL	FALSE	
VAR	IN_2	N/A [Auto]	BOOL	FALSE	
VAR	OUT_0	N/A [Auto]	BOOL	FALSE	
VAR	OUT_1	N/A [Auto]	BOOL	FALSE	

6.2.10 變數符號的篩選

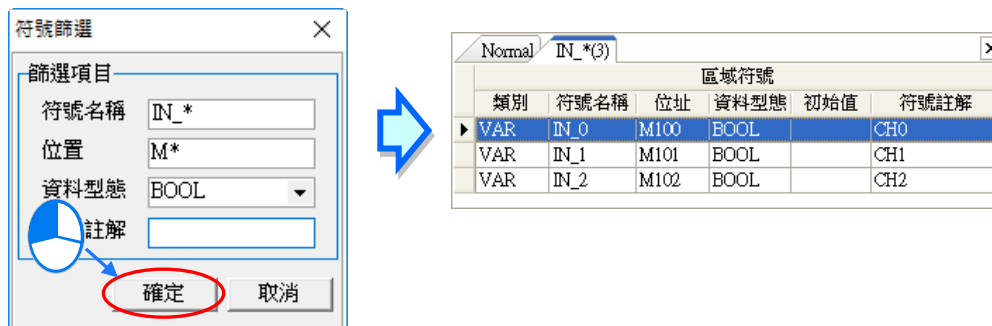
此功能可用於檢視具有共同條件的變數符號。

- (1) 於符號表處按下滑鼠右鍵，之後點選「符號篩選」。

6

類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	IN_0	M100	BOOL	FALSE	CH0
VAR	IN_1	M101	BOOL	FALSE	CH1
VAR	IN_2	M102	BOOL	FALSE	CH2
VAR	OUT_0	M200	BOOL	FALSE	Value_0
VAR	OUT_1	M201	BOOL	FALSE	Value_1
VAR	OUT_2	M202	BOOL	FALSE	Value_2
VAR	MFC_0	D200	BOOL	FALSE	PV_MFC_0
VAR	MFC_1	D201	BOOL	FALSE	PV_MFC_1
VAR	TEMP	N/A [Auto]	BOOL	FALSE	

- (2) 於出現的視窗中設定篩選條件，而輸入的內容亦可使用萬用字元「*」來代表任何字串，且當系統進行篩選時，符號表中的變數符號必須同時符合每個設定的條件才會被篩選出來，而若希望略過某個條件時，只要讓該條件的欄位保持空白即可。設定完成後按下「確定」鍵即可。



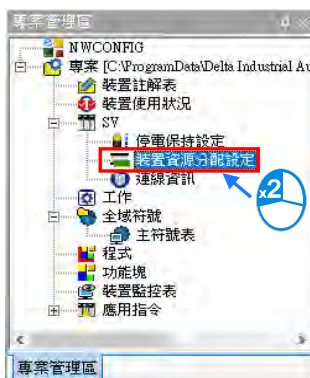
- (3) 被篩選出來的變數符號會顯示於符號表中另外開啟的顯示分頁中，點擊符號表上方的標籤即可進行切換，當篩選的組別過多時，利用頁面右上方的 ◀ 與 ▶ 鍵亦可對標籤進行切換，而點擊頁面右上方的 ✕ 鍵即可關閉該分頁的顯示。此外，這些另外開啟的顯示分頁僅供檢視之用，本身無法進行任何編輯，若要對符號表進行編輯時，請先切回原符號表，即標籤名為 **Normal** 的顯示分頁，接著再進行後續的操作。



6.2.11 裝置資源分配設定

部分的 DVP 機種中，自動配置的位址都會使用原本的裝置資源，而為了便於管理，使用者可事先於專案中設定用來進行自動配置的裝置範圍，且不同的專案皆可有其專屬的設定；當進行專案編譯時，系統只會使用此範圍內的裝置來進行配置，但若變數符號的數量超過所設定的範圍時，編譯的結果便會產生裝置資源不足的錯誤訊息。關於裝置位址配置的相關說明請參考本章第 6.1.4 節的內容。

設定時先展開專案管理區的機種項目，之後再於其下層的「裝置資源分配設定」項目上雙擊滑鼠左鍵。



接著便可於出現的視窗中設定各種裝置的配置範圍，且各種裝置都有其各自的範圍限制，例如 SV 機種的 D 裝置只能從 D2000 開始，M 裝置只能從 M2000 開始，C 裝置只能配置一般用的 16 Bits 計數器，T 裝置則只可設定時基為 100ms 的計時器，而當輸入不允許的位址時，系統便會自動進行修正。完成後按下「確定」鍵即可，而此處的設定亦會隨著專案一起被儲存。



6.3 範例實作

6

在第 4 章的內容中，我們曾以一個簡單的範例來示範如何建立一個階梯圖程式，並在第 5.6.1 節中以 POU 與 TASK 的程式架構重新規劃；而在接下來的內容中，我們將延續於 5.6.1 節中所建立的範例，並在程式中以變數符號來取代原本的裝置名稱。

6.3.1 變數符號規劃

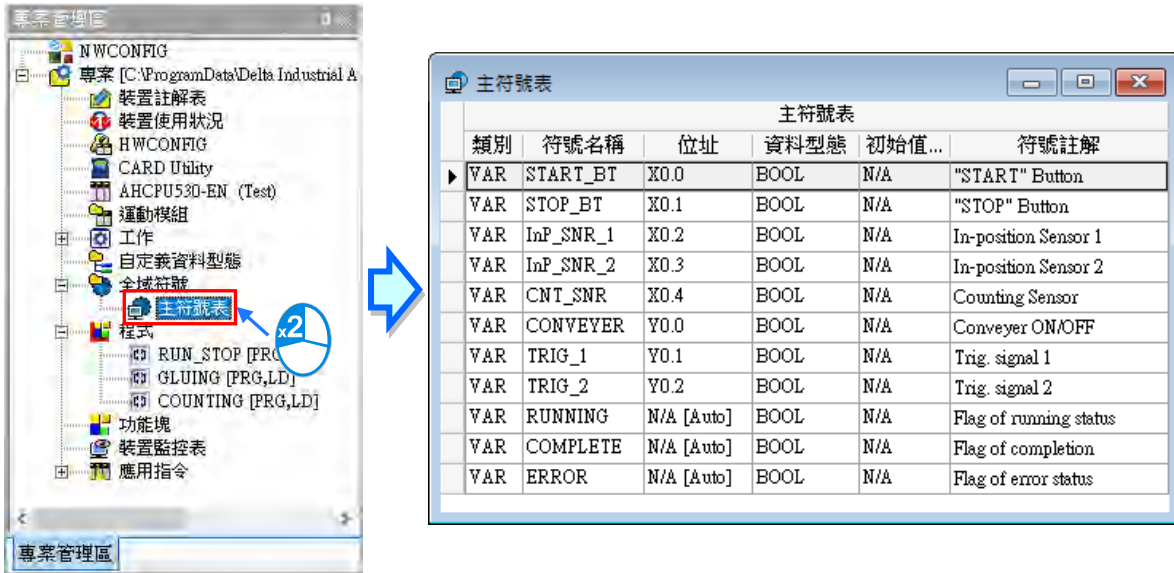
下表為重新規劃的裝置符號對照表，而其規劃的原則與建議請參考下述。

原裝置名稱	符號名稱	指定位址	宣告位置	功能說明
X0.0	START_BT	X0.0	全域符號表	啟動按鈕
X0.1	STOP_BT	X0.1	全域符號表	停止按鈕
X0.2	InP_SNR_1	X0.2	全域符號表	到位感測器 1
X0.3	InP_SNR_2	X0.3	全域符號表	到位感測器 2
X0.4	CNT_SNR	X0.4	全域符號表	計數感測器
Y0.0	CONVEYER	Y0.0	全域符號表	輸送帶
Y0.1	TRIG_1	Y0.1	全域符號表	注射器 1 觸發信號
Y0.2	TRIG_2	Y0.2	全域符號表	注射器 2 觸發信號
M0	RUNNING	自動配置	全域符號表	運轉旗標
M1	COMPLETE	自動配置	全域符號表	完成旗標
M2	ERROR	自動配置	全域符號表	錯誤旗標
D0	CNT_DT	自動配置	區域符號表	入料計數值

- X 與 Y 裝置為與 PLC 外部連接的接點，且其裝置位址與設備的接線有關，所以必須自行指定正確的裝置位址；此外，雖然部分的 X 與 Y 裝置僅會於某個 POU 當中被使用，但仍建議將其建立於全域符號表中，這對於後續的維護會較為便利，例如變更 IO 接線，或是要於後續新增的程式中存取這些接點時；不過這並非必然的，使用者仍可依據自己的習慣來進行規劃。
- 原本的 M0、M1 與 M2 這 3 個旗標，因為會被不同的 POU 進行存取，所以將這三個變數符號宣告於全域符號表中，且在操作上無須刻意指定裝置位址，因此將其規劃為自動配置。
- 原本的 D0 暫存器，其功能僅做為計數值的暫存之用，且只會用於「**COUNTING**」的 POU 當中，因此在規劃上，其對應的變數符號僅須直接宣告在「**COUNTING**」POU 的區域符號表當中，且其裝置位址亦只要規劃為自動配置即可。

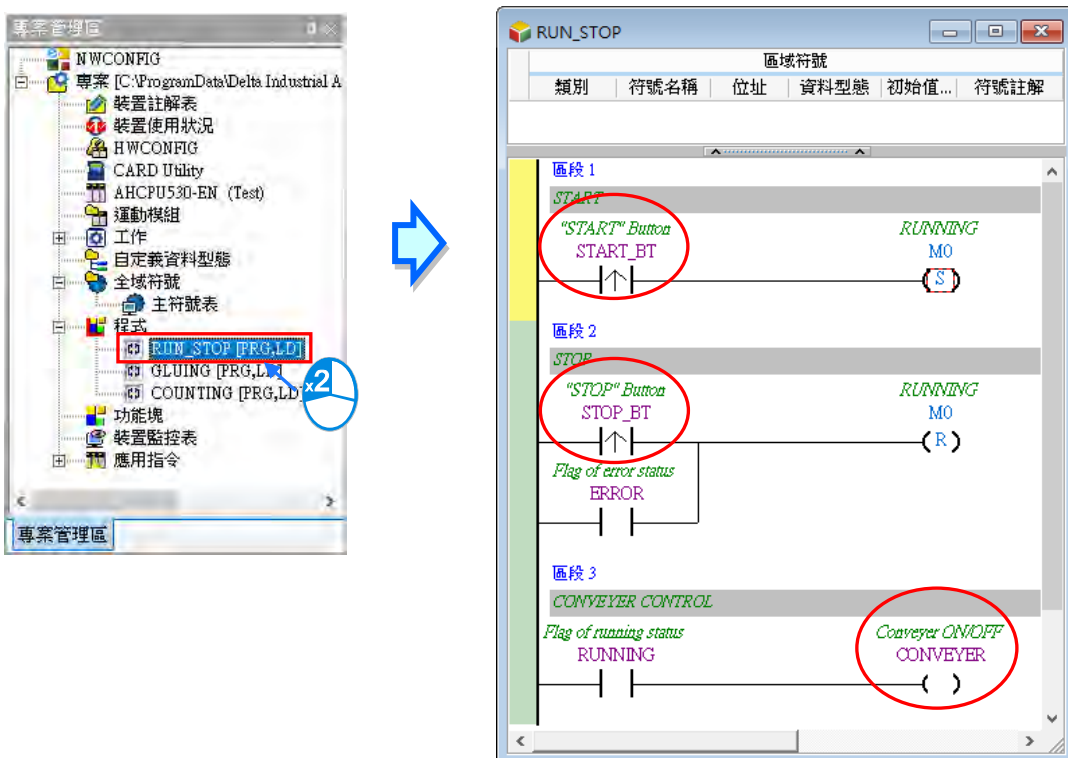
6.3.2 建立範例程式



接下來可直接開啟第 5.6.1 節當中的範例來進行修改或是重新建立程式。若要重新建立程式時，請先新增一個專案，之後參考第 4 章與第 5.6.1 節的相關介紹，並配合接下來的說明逐一完成此節的範例。首先請開啟全域符號表，並參考本章第 6.2 節的內容於全域符號表中建立如下的變數符號。

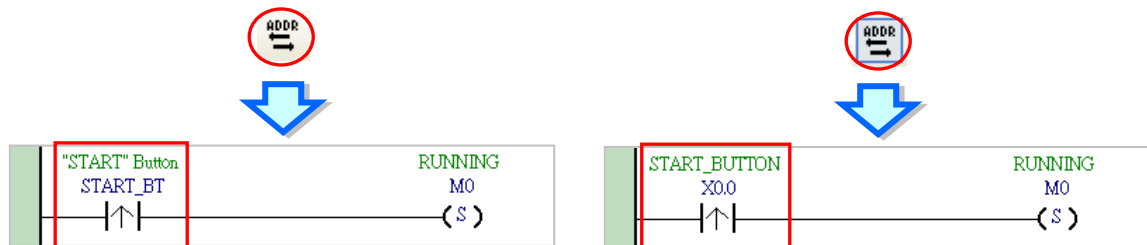


完成全域符號表的建立後，請開啟「RUN_STOP」這個程式 POU，而若是直接開啟第 5.6.1 節中的範例來進行修改的使用者將可發現，自行指定位址的裝置都已變更為對應的符號名稱，且原本的裝置註解也已顯示為符號註解；若未有如下圖的顯示時，請參考後續的內容進行顯示的切換。

6

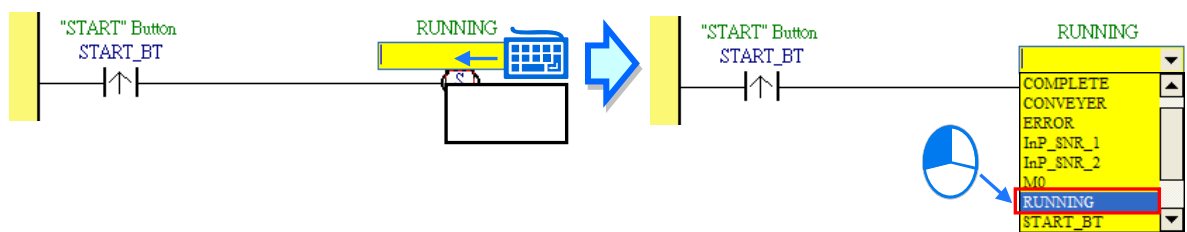


對於有指定裝置位址的變數符號，於 LD 的環境中，使用者可自行選擇該變數符號的呈現方式；當圖示工具列的  圖示為按下的狀態時，程式畫面便會以指定的裝置位址與裝置註解來顯示這些變數符號；而若  圖示為未按下的狀態時，程式畫面便會以符號名稱與符號註解來顯示。

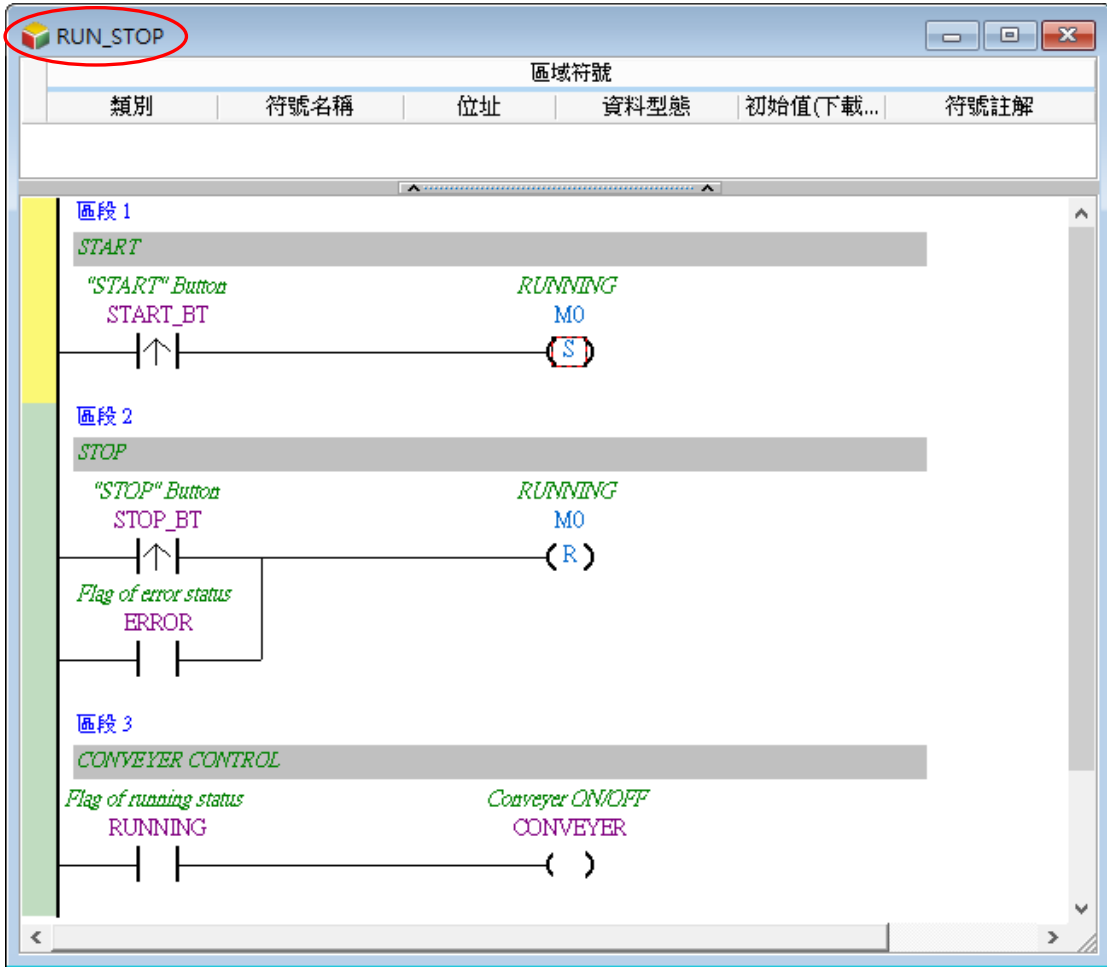


*.上圖的 **MO** 因並未配置給任何的變數符號，所以切換時仍會一直保持原狀；而同樣的，若變數符號未配置裝置位址，或是由系統自動配置至內部記憶體時，該變數符號的顯示也會保持原狀。

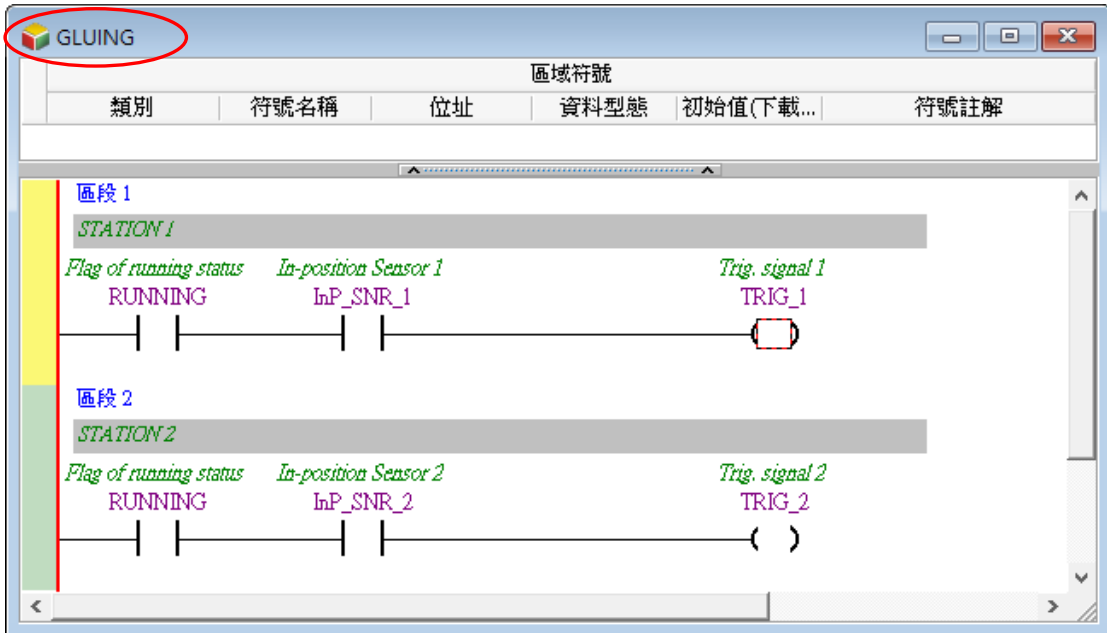
接著可將程式中的**運轉旗標 MO** 變更為變數符號「**RUNNING**」，而變更的方式除了依據第 4 章的說明直接輸入之外，先清除裝置輸入欄位後點擊滑鼠左鍵，之後自動出現符合該欄位型態變數符號的下拉選單供使用者挑選。而若是使用鍵盤時，亦可在該欄位進入編輯狀態時，配合【↑】與【↓】鍵便可進行選取，完成後按下【Enter】鍵即可。

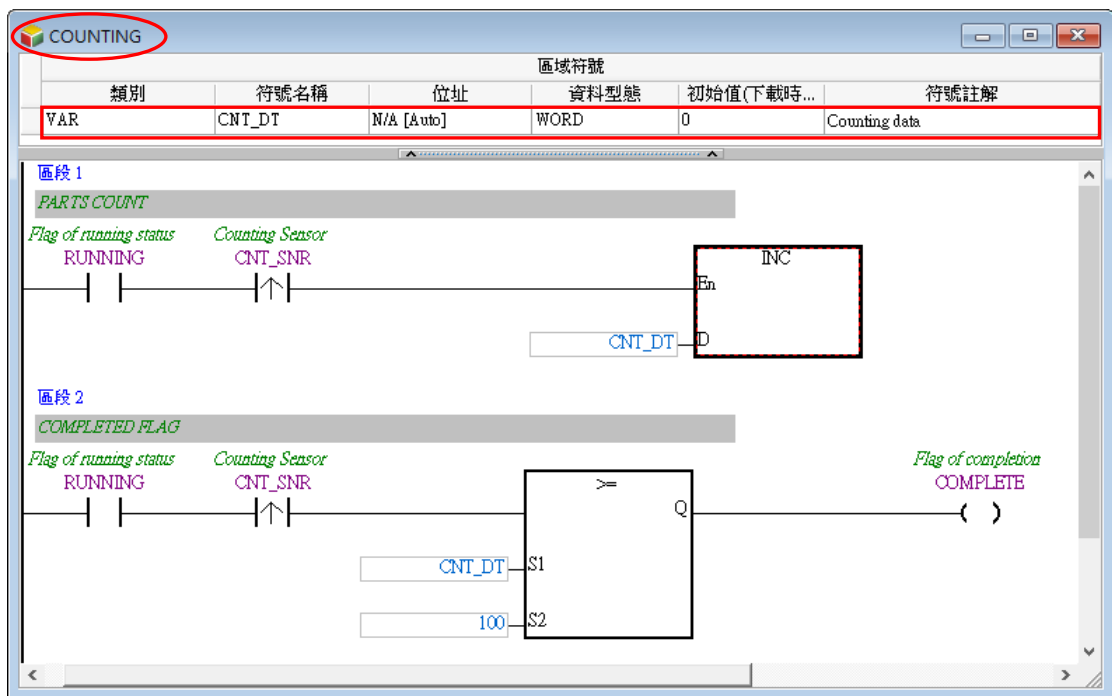


接下來請參考之前的說明逐一完成如下的程式 POU。



6





若是重新建立程式的使用者，請依據第 4 章與第 5.6.1 節的相關說明完成硬體組態、硬體參數設定、工作 (TASK) 管理... 等操作；而完成範例程式後，請自行編譯專案程式，並將硬體參數與程式下載至主機當中進行測試；而下載程式時，請一併勾選「符號初始值」以便將「COUNTING」POU 中的區域符號「CNT_DT」的初始值一併寫入主機中。



MEMO

6

7

第7章 功能塊與函式庫

目錄

7.1 認識功能塊 (Function Block)	7-2
7.1.1 功能塊簡介	7-2
7.1.2 功能塊的特性與優點	7-3
7.2 ISPSOft 功能塊 (Function Block) 架構	7-4
7.2.1 功能塊的致能 (En) 端點	7-4
7.2.2 功能塊的變數符號	7-5
7.2.3 功能塊的輸出與輸入	7-6
7.2.4 指標型態的變數符號	7-12
7.2.5 功能塊的定義與實例	7-16
7.2.6 功能塊呼叫功能塊的巢式結構	7-19
7.2.7 功能塊的記憶體配置	7-21
7.3 功能塊的使用	7-26
7.3.1 功能塊的基本規格	7-26
7.3.2 功能塊專用的脈波型指令 (AH/AS 機種限定)	7-27
7.3.3 功能塊的程式監控	7-29
7.3.4 功能塊的修改	7-30
7.4 範例實作	7-31
7.4.1 程式規劃	7-31
7.4.2 建立範例程式	7-31
7.4.3 函式 (Function) 範例程式	7-41
7.5 認識函式庫 (Library)	7-42
7.5.1 新增使用者自訂函式庫	7-42
7.5.2 引用使用者自訂函式庫	7-44
7.5.3 引用台達函式庫	7-46

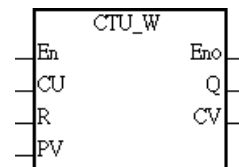
7.1 認識功能塊 (Function Block)

功能塊 (**Function Block**，簡稱 **FB**) 與函式 (**Function**，簡稱 **FC**) 是在 PLC 編程中相當重要的一個環節，同時也是 IEC61131-3 所規範的重點項目之一，而因其本身所具有的特性及優點，所以在 ISPSOft 中同時也針對功能塊的特性提供了相當多的支援與功能。在 ISPSOft 中，提供了使用者自行定義的 FB POU 與系統定義的 FB POU；目前只有 DVPxxMC 與 AS5xx 機種支援使用者自行定義的 FC POU，且提供了 API 應用指令做為系統的 FC 供使用者調用。

7.1.1 功能塊簡介

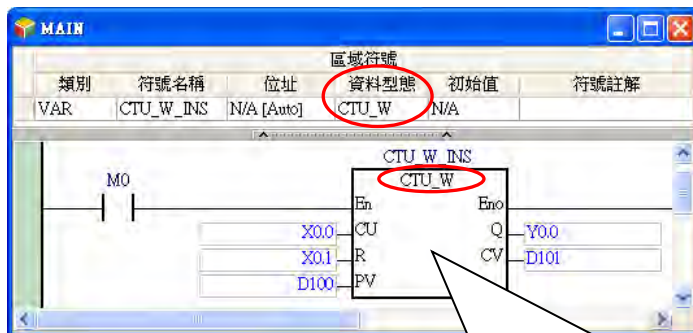
功能塊是一種具有運算功能的程式元件，本身也屬於 POU 的一種，但它無法自行運作，而是必須透過過程 POU 對它進行呼叫，並傳遞相關參數之後，才能執行功能塊所定義的功能；而功能塊在執行完畢後，也同樣可將內部的運算結果再回傳給上層 POU (呼叫者) 所指定的裝置或變數符號中。

右圖即為功能塊的圖示，外觀與 API 應用指令相當類似，於左右兩側皆有輸入與輸出的端點，而使用的方法也十分相近，較大的差異在於上層 POU (呼叫者) 在呼叫功能塊時，使用者必須為其宣告一個功能塊型態的變數符號，而且在功能塊內部也允許再對其他的功能塊進行呼叫。



功能塊是含記憶功能的單位，每一個功能塊都會被配置一組實體的記憶區塊，用來存放內部變數符號的狀態值，且因功能塊內部的變數符號狀態會在執行結束後被保留下來，所以每次功能塊的執行結果都將受到上次執行狀態的影響，因此實務上有可能在同樣的輸入條件下，而得到不同的輸出結果。

下圖便是個一功能塊的應用範例，其中可看到上層 POU (呼叫方) 的區域符號表中有一個功能塊型態的變數符號，且該資料型態與功能塊的名稱相同。關於功能塊的使用將於接下來的章節中陸續介紹。



7.1.2 功能塊的特性與優點

功能塊之所以被 IEC 61131-3 納入標準，並且受到眾多使用者的喜愛，主要是因為它具有許多傳統 PLC 編程方式所無法達到的特性與優點。

- **模組化設計**

將一個龐大的程式分成許多不同的副程式，並將各個副程式以功能塊的型式加以建立，使用時只要在程式 POU 中加以呼叫並排列組合即可。

- **高獨立性**

不同功能塊可依據及其本身的功能特性或開發人員的習慣而使用不同的編程語言來建立，如 LD、IL、ST、FBD...等；而在不同編程語言的 POU 中呼叫功能塊時，也完全無需考慮該功能塊是以何種語言所建立的；例如在 LD 的程式中，若遇到一段須進行複雜運算的程序時，使用者便可將這段程式以 ST 語言建立成一個功能塊，之後再於原來的 LD 程式中進行呼叫即可。

- **可重用性**

功能塊一旦建立之後，只要符合使用規則及限制，原則上可一再的被重複使用。

- **高可攜性**

功能塊開發完成後，不只是在原來的專案中才可使用，利用匯出/匯入的功能，在別的專案程式中亦可使用這些開發好的功能塊，因此使用者可逐漸地建立屬於自己的功能塊元件庫。

- **維護便利**

因功能塊已將其所屬的程式封裝成一個獨立的模組，因此當發現功能塊內部的程式有錯誤或是所定義的功能與實際需求不符時，使用者僅須集中焦點在修改功能塊內部所定義的內容即可，而無須在整個龐大的程式當中進行除錯或修改。

- **增加程式的可讀性**

使用者可將較為複雜或重複性高的程式封裝成一個功能塊，接著再於原本的程式當中呼叫使用，如此便可精簡原本程式的架構，並提高其可讀性。

- **高保密性**

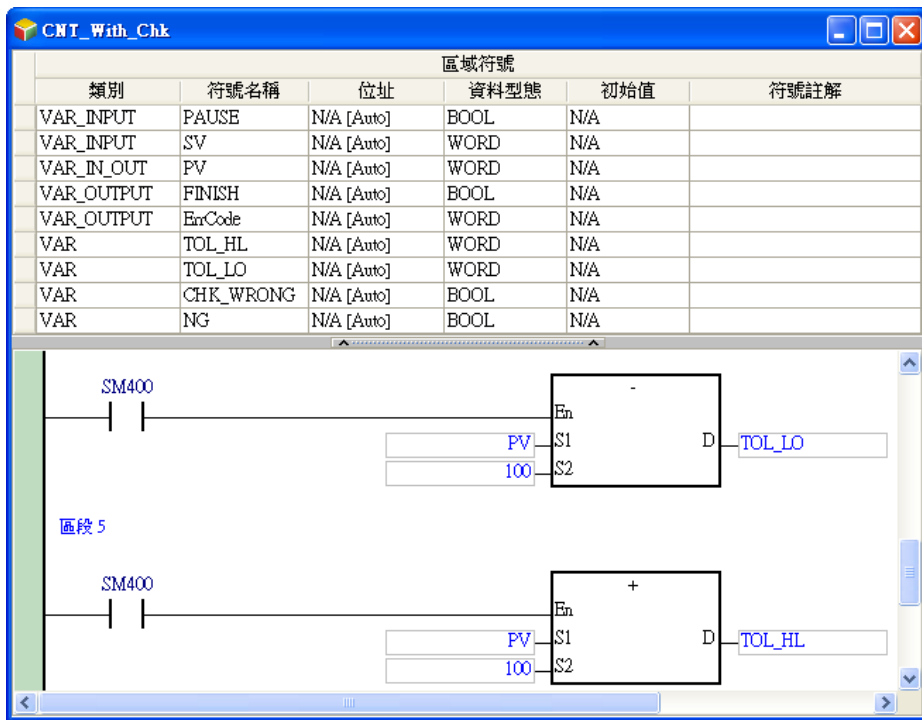
所建立的功能塊可單獨設定密碼，使用者可放心的將自己的核心技術建立在功能塊中並提供給他人使用；除此之外，系統商或獨立軟體開發者亦可在不公開原始碼的情況下，只提供特定功能的功能塊給有需求的客戶。

- **高開發效率**

依據上述所提及的種種特點，以專案開發的角度來看，功能塊的應用可以提供一個高效率的開發環境，且因功能塊的模組化特性，更可讓不同的開發人員甚至外包廠商共同參與整個專案的開發，對於人力資源的利用，亦可提供相當大的彈性。

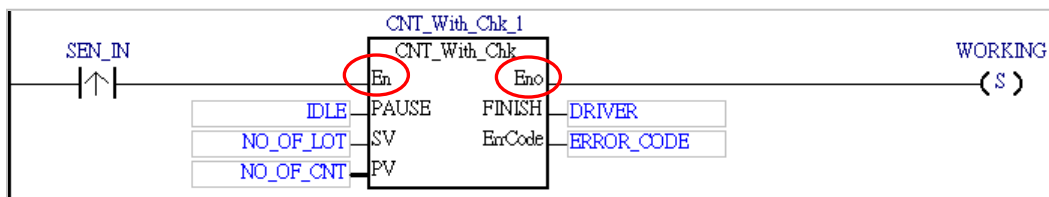
7.2 ISPSOft 功能塊 (Function Block) 架構

請參考下圖，功能塊及函式內部的結構與一般的程式 POU 相似，分為區域符號表與程式主體，且在編輯的方式上也與一般的程式 POU 幾乎相同。在 ISPSOft 中可用來建立功能塊的編程語言分別有 LD、IL、ST 及 FBD，但不包含 SFC，而在這些語言的程式當中也同樣可以呼叫功能塊，其中包括在 SFC 的 Action 中同樣可以對功能塊進行呼叫。此外，功能塊內部的程式也允許再對其他的功能塊進行呼叫，而呼叫功能塊時，完全無須考慮該功能塊的程式是以何種語言所編寫的。



7.2.1 功能塊的致能 (En) 端點

於上層 POU (呼叫方) 當中呼叫功能塊的方式與執行 API 應用指令的方式類似，同樣以輸入 En 端點的邏輯狀態來決定該功能塊是否被執行；當連結至 En 端點的狀態為 ON 時，功能塊才會被執行，反之當連結至 En 端點的狀態為 OFF 時，該功能塊的動作就會被略過；至於功能塊的 Eno 端點，其功能則僅是把 En 端點的邏輯狀態再串接輸出給後方的區塊。此外，當於 ST 語言中使用功能塊時則不會帶入 En 接點的狀態，詳細說明將於第 13 章的內容當中另行介紹。



7.2.2 功能塊的變數符號

與一般的程式 POU 相同，功能塊的內部可以建立屬於自己的區域符號表，而且也可以在功能塊當中使用全域符號，只是在功能塊內部使用全域符號時，便可能失去其可攜性，因為當我們利用匯出與匯入的方式將功能塊複製到另一個專案當中使用時，該專案不見得會有相同的全域符號。

下列為與功能塊相關的變數符號說明摘要，各機種在功能塊使用變數符號類別的支援狀態，如下表。

- 功能塊的變數符號類別

類別	說明	AH/AS	DVP	AS5xx DVPxxMC
VAR	僅供功能塊內部運算之用，且在運算結束後，變數符號的狀態會被保留下來。	✓	✓	✓
VAR_INPUT	用做功能塊的輸入端點。可接收上層 POU (呼叫方) 所指定的運算元，並在執行功能塊時將該運算元的內容值複製進功能塊。	✓	✓	✓
VAR_OUTPUT	用做功能塊的輸出端點。可在功能塊執行完畢後，將運算的結果透過此類變數符號輸出至上層 POU (呼叫方) 所指定的運算元。	✓	✓	✓
VAR_IN_OUT	功能塊的反饋型端點。執行功能塊時，先透過此端點將上層 POU (呼叫方) 所指定的運算元內容傳進功能塊，待功能塊執行完畢後，再將運算的結果反饋輸出至上層 POU (呼叫方) 所指定的運算元。	✓	✓	
VAR_RETAIN	RETAIN 型態的變數具有斷電保持的屬性，斷電保持的變數最多可以定義 15999 個。			✓
VAR_STATIC	系統會給該變數分配一個固定的位址，使得該變數的值在控制器運行過程於相同功能塊的不同 Instance 之間能夠共享變數。			✓
VAR_STATIC_RETAIN	變數為 STATIC 型態且具有斷電保持屬性。			✓

- 功能塊內部區域符號所支援的資料型態

機種	說明
AH/AS	BOOL、WORD、DWORD、LWORD (*註 2)、INT、DINT、LINT (*註 2)、REAL、LREAL (*註 2)、ARRAY、STRING、POINTER (*註 1)、T_POINTER (*註 1)、C_POINTER (*註 1)、HC_POINTER (*註 1)、功能塊型態、使用者自訂型態
DVP	BOOL、WORD、DWORD、REAL、ARRAY、STEP、TIMER、COUNTER
AS5xx/ DVPxxMC	BOOL、BYTE、WORD、DWORD、LWORD、SINT、INT、DINT、LINT、USINT、UINT、UDINT、ULINT、REAL、LREAL、ARRAY、STRING、TIME、DATE、TOD、DT、功能塊型態、使用者自訂型態

*1. POINTER、T_POINTER、C_POINTER 與 HC_POINTER 等指標型態的相關說明將於第 7.2.4 節當中另行介紹。

*2. AS 系列機種在使用功能塊內部區域符號不支援 LWORD、LINT 及 LREAL。

- 功能塊的記憶體配置

機種	說明
AH/AS	宣告為功能塊型態的變數符號 (功能塊實例) 與建立於功能塊內部的區域符號，宣告時皆無法自行指定裝置位址，系統將會自動為其配置一個主機內部預留的記憶體區塊。
DVP	每個宣告為功能塊型態的變數符號 (功能塊實例) 都會佔用一個 P 裝置；而建立於功能塊內部的區域符號，配置時則會視其資料型態，自主機目前可用的裝置資源中，為每個功能塊實例的區域符號配置一個對應的裝置位址。
AS5xx/ DVPxxMC	宣告為功能塊型態的變數符號 (功能塊實例) 與建立於功能塊內部的區域符號，宣告時皆無法自行指定裝置位址，系統將會自動為其配置一個主機內部預留的記憶體區塊。

*.關於功能塊實例與記憶體配置的詳細說明將於第 7.2.5 節~第 7.2.7 節的內容當中另行介紹。

7

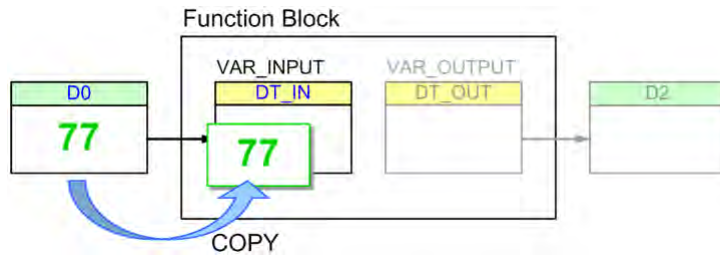
7.2.3 功能塊的輸出與輸入

一般功能塊的輸出與輸入都是採用傳值的方式，即不論是呼叫方將某個裝置指定給功能塊的輸入端，或是功能塊的運算結果經由輸出端點回傳給呼叫方的裝置，彼此只是把本身的值複製給對方而已，單個功能塊支持多個輸入與輸出，而函式僅有一個輸出參數也就是其返回值。

在此以下列的圖例來加以說明。

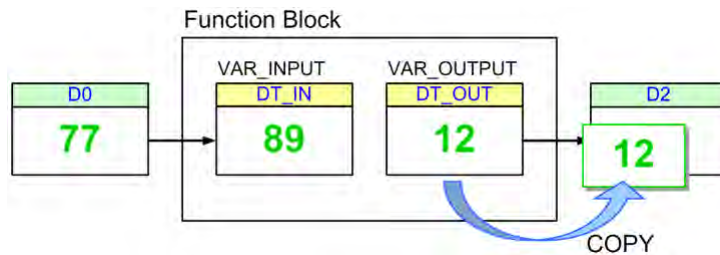
● 功能塊執行前

呼叫方將裝置 D0 指定給功能塊的輸入端點 DT_IN (VAR_INPUT 類別)，因此當功能塊被呼叫時，系統便會將裝置 D0 的現在值複製給功能塊的變數符號 DT_IN。

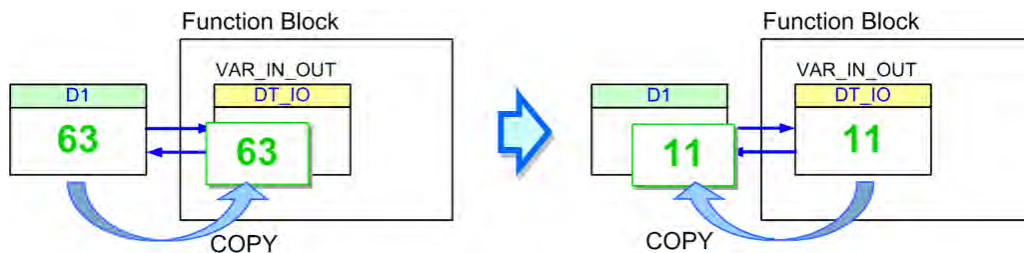


● 功能塊執行後

運算結束後，變數符號 DT_IN 的內容值變為 89，不過此改變並不會影響呼叫方的裝置 D0；此外，功能塊的 DT_OUT (VAR_OUTPUT 類別) 會將最後的運算結果複製給呼叫方指定的裝置 D2，而之後即使 D2 的內容在程式執行的過程中被改寫了，這個結果依舊不會影響功能塊內部的 DT_OUT。

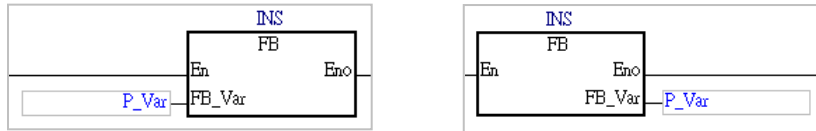


至於變數符號類別為 VAR_IN_OUT 的情況，雖然此類別的變數符號會在功能塊執行結束後，立即的更新呼叫方指定給該端點的裝置內容，但本質上還是先將呼叫方裝置的現在值複製進來後，再將最後的結果複製回去給呼叫方裝置。



*.在 AH/AS 系列的機種中，功能塊的 VAR_IN_OUT 類別端點可宣告為指標型態的變數符號，而有別於上述的傳值方式，透過指標型態的變數符號，呼叫方可以把外部裝置的位址指標傳進功能塊，詳細說明請參考下一節的內容。

接著請參考下列兩個功能塊的圖例，其中 FB_Var 為功能塊的端點變數符號，而變數符號 P_Var 則為上層 POU (呼叫方) 指定給功能塊端點的運算元。基本上 P_VAR 與 FB_VAR 兩者之間的資料型態必須符合下述的**基本原則**，但為了編程上的便利，ISPSOft 尚提供幾個**特例原則**，只要使用上仍符合這幾個**特例原則**中的其中一個，即使不符合**基本原則**，系統仍會將其視為合法的使用方式。DVPxxMC 與 AS5xx 機種則遵守其資料型態的原則。



● 一般資料型態的基本原則

呼叫方的變數符號 (P_VAR) 與功能塊端點 (FB_VAR) 的資料型態須完全一致。

● 一般資料型態的特例原則 (參考下表)

1. 當有任何一方為 WORD 系列的資料型態時，另一方便允許是 WORD、INT、REAL 任一系列的資料型態，而滿足此條件時，P_VAR 的資料長度便允許大於或等於 FB_VAR。
2. 若 P_VAR (呼叫方) 的資料型態為 STEP，則 FB_VAR (功能塊) 的資料型態可為 BOOL，此操作代表呼叫方的步進點狀態將會在功能塊的內部使用。

適用機種		AS/AH/DVP								
資料型態		FB_Var (功能塊)								
		WORD	DWORD	LWORD	INT	DINT	LINT	REAL	LREAL	BOOL
P_VAR (呼叫方)	WORD	✓			✓					
	DWORD	✓	✓		✓	✓		✓		
	LWORD	✓	✓	✓	✓	✓	✓	✓	✓	
	INT	✓			✓					
	DINT	✓	✓			✓				
	LINT	✓	✓	✓			✓			
	REAL	✓	✓					✓		
	LREAL	✓	✓	✓					✓	
	BOOL									✓
	STEP									✓

除了上述單純資料型態的原則之外，針對 ARRAY 型式的變數符號，其使用原則說明如下。

● ARRAY 操作的基本原則

1. 兩方 ARRAY 元素的資料型態必須相同。
2. 資料型態相同時，P_VAR (呼叫方) 的 ARRAY 長度允許大於 FB_VAR (功能塊) 的長度。

● ARRAY 操作的特例原則

1. 比較 ARRAY 的資料型態時，前述一般資料型態的特例原則亦可套用。
2. 當一方為 ARRAY 型式，而另一方為非 ARRAY 型式的變數符號時，系統會將非 ARRAY 型式的變數符號視為陣列長度為 1 的 ARRAY。

7

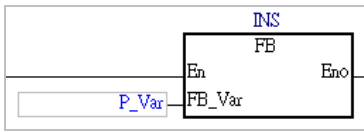
3. 當有一方的 ARRAY 型態為 WORD 系列時，另一方可為 WORD、INT、REAL 任一系列，且兩方在比較資料長度時，可以採用加總的結果來進行判斷。例如 DWORD 相容於長度為 2 的 INT 陣列，而長度為 4 的 WORD 陣列亦相容於一個 LINT 的變數符號；不過長度為 4 的 INT 陣列與一個 LINT 的變數符號並不相容，因兩方並沒有 WORD 型態的資料格式，因此不符合此原則。

下列則為上述幾項使用原則的說明範例。

- ✓ EX1：P_VAR (呼叫方) 為 INT，FB_VAR (功能塊) 為 INT。
 - ➔ 符合一般資料型態的基本原則，毫無疑問的是合法使用。
- ✓ EX2：P_VAR (呼叫方) 為 WORD，FB_VAR (功能塊) 為 INT。
 - ➔ 符合一般資料型態的特例原則 1，因此是合法使用。
- ✓ EX3：P_VAR (呼叫方) 為 LWORD，FB_VAR (功能塊) 為 DINT。
 - ➔ 符合一般資料型態的特例原則 1，因此是合法使用。
- ✗ EX4：P_VAR (呼叫方) 為 WORD，FB_VAR (功能塊) 為 LINT。
 - ➔ 雖然有一方為 WORD 型態，但 P_VAR 的資料長度小於 FB_VAR，因此是不合法的。
- ✗ EX5：P_VAR (呼叫方) 為 LINT，FB_VAR (功能塊) 為 DINT。
 - ➔ 兩方資料型態不一致，因此是不合法的。
- ✓ EX6：P_VAR (呼叫方) 為長度 4 的 INT 陣列，FB_VAR (功能塊) 為長度 3 的 INT 陣列。
 - ➔ 符合 ARRAY 操作的基本原則，因此是合法使用。
- ✓ EX7：P_VAR 為長度 2 的 WORD 陣列，FB_VAR 則為一般的 WORD 型態 (非陣列格式)。
 - ➔ 符合 ARRAY 操作的特例原則 2，因此是合法使用。
- ✓ EX8：P_VAR 為長度 4 的 WORD 陣列，FB_VAR 則為一般的 DWORD 型態 (非陣列格式)。
 - ➔ 符合 ARRAY 操作的特例原則 3，因此是合法使用。
- ✗ EX9：P_VAR 為長度 4 的 DINT 陣列，FB_VAR 為長度 2 的 INT 陣列。
 - ➔ 兩方資料型態不一致，因此是不合法的。
- ✓ EX10：P_VAR 為一般的 DWORD 型態 (非陣列格式)，FB_VAR 則為長度 2 的 WORD 陣列。
 - ➔ 符合 ARRAY 操作的特例原則 3，因此是合法使用。

● DVPxxMC 與 AS5xx 資料型態的原則

以下圖為例



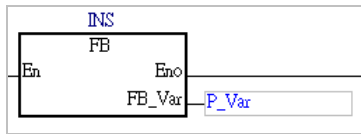
1. 此處一般情況呼叫方的變數符號 (P_VAR) 與功能塊端點 (FB_VAR) 的資料型態須完全一致，但在 ISPSOft 也允許部分情況 P_VAR (呼叫方) 的資料型態長度允許小於或等於 FB_VAR (功能塊) 的資料型態長度，此詳細說明請參考下表。
2. 若 FB_VAR(功能塊) 的變數資料型態為 ARRAY，則 P_VAR(呼叫方) 資料型態必須為 ARRAY，資料型態內容及長度大小需與 FB_VAR (功能塊) 相同。

7

適用機種		DVPxxMC/AS5xx														
資料型態		FB_Var (功能塊)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	LINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (呼叫方)	BOOL	✓														
	BYTE		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	WORD			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
	DWORD				✓	✓			✓	✓			✓	✓	✓	✓
	LWORD					✓				✓			✓	✓	✓	✓
	USINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	UINT			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
	UDINT				✓	✓			✓	✓			✓	✓	✓	✓
	ULINT					✓				✓				✓	✓	✓
	SINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INT			✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓
	DINT				✓	✓			✓	✓			✓	✓	✓	✓
	LINT					✓				✓				✓	✓	✓
	REAL														✓	✓
	LREAL														✓	✓

適用機種		DVPxxMC/AS5xx				
資料型態		FB_Var (功能塊)				
		TIME	DATE	TOD	DT	STRING
P_VAR (呼叫方)	TIME	✓				
	DATE		✓			
	TOD			✓		
	DT				✓	
	STRING					✓

再以下圖為例



1. 此處一般情況呼叫方的變數符號 (P_VAR) 與功能塊端點 (FB_VAR) 的資料型態須完全一致，但在 ISPSOft 也允許部分情況 P_VAR (呼叫方) 的資料型態長度允許大於或等於 FB_VAR (功能塊) 的資料型態長度，此詳細說明請參考下表。
2. 若 FB_VAR(功能塊) 的變數資料型態為 ARRAY，則 P_VAR(呼叫方) 資料型態必須為 ARRAY，資料型態內容及長度大小需與 FB_VAR (功能塊) 相同。

適用機種		DVPxxMC/AS5xx														
資料型態		FB_Var (功能塊)														
		BOOL	BYTE	WORD	DWORD	LWORD	USINT	UINT	UDINT	ULINT	SINT	INT	DINT	LINT	REAL	LREAL
P_VAR (呼叫方)	BOOL	✓														
	BYTE		✓				✓				✓					
	WORD		✓	✓			✓	✓			✓	✓				
	DWORD		✓	✓	✓		✓	✓	✓		✓	✓	✓			
	LWORD		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	USINT		✓				✓				✓					
	UINT		✓	✓			✓	✓			✓	✓				
	UDINT		✓	✓	✓		✓	✓	✓		✓	✓	✓			
	ULINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	SINT		✓				✓				✓					
	INT		✓	✓			✓	✓			✓	✓				
	DINT		✓	✓	✓		✓	✓	✓		✓	✓	✓			
	LINT		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
	REAL		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	LREAL		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

7

適用機種		DVPxxMC/AS5xx				
資料型態		FB_Var (功能塊)				
		TIME	DATE	TOD	DT	STRING
P_VAR (呼叫方)	TIME	✓				
	DATE		✓			
	TOD			✓		
	DT				✓	
	STRING					✓

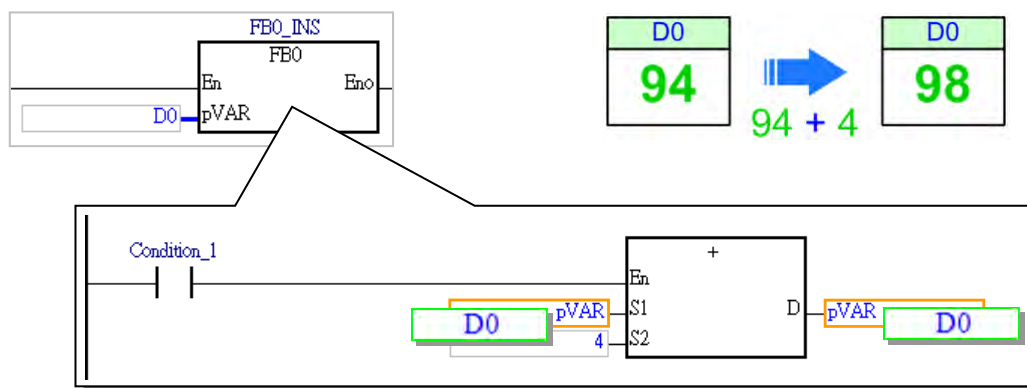
補充說明

- (a) 在符合上述原則的情況下，當 P_VAR (呼叫方) 的資料長度大於 FB_VAR (功能塊) 時，若是將 P_VAR 輸入 FB_VAR 的場合，系統便會由 P_VAR 的低位元開始，只取符合 FB_VAR 長度的資料；若是由 FB_VAR 輸出給 P_VAR 的場合，系統則會由低位元開始，只改寫 P_VAR 中符合 FB_VAR 長度的資料，超過長度的部份則會保持原狀。使用時請小心資料的正確性。
- (b) 在 AH 的功能塊變數符號中尚有指標型態的特殊用法，詳細說明請參考第 7.2.4 節的內容。
- (c) 在 AS5xx/DVPxxMC 機種，當 P_VAR (呼叫方) 的資料長度小於 FB_VAR (功能塊) 時，若是將 P_VAR 輸入 FB_VAR 的場合，系統便會由 P_VAR 的低位元開始，只取符合 P_VAR 長度的資料輸入給 FB_VAR；若是由 FB_VAR 輸出給 P_VAR 的場合，系統則會由低位元開始，只改寫 P_VAR 中符合 FB_VAR 長度的資料，超過長度的高位元部份則會被寫成 0。使用時請小心資料的正確性。

7.2.4 指標型態的變數符號

指標型態的變數符號主要用於將呼叫方指定的變數符號或裝置指標傳進功能塊內部，其使用方式上與一般的資料型態有相當大的不同。一般資料型態的變數符號會使用複製的方式來傳遞兩方目前的內容值；但若功能塊端點被宣告為指標型態時，傳進功能塊的資料則會是指定給該端點的變數符號或裝置本身的指標，因此在功能塊內部進行處理的便會是被指定的對象本身。

如下圖範例，呼叫方將裝置 D0 指定給功能塊端點「pVAR」，而「pVAR」在功能塊內部則被宣告為指標型態的變數符號，因此當功能塊被呼叫並開始執行時，功能塊內部便會把「pVAR」視為裝置 D0 的本身，而不是單純的將 D0 的內容值複製進來而已。



關於指標型態的變數符號，其使用原則如下。

1. 只適用於 **AH/AS** 以及 **DVP-ES3** 系列的機種。
2. 只可在功能塊內部宣告，且必須宣告為 **VAR_IN_OUT** 類別的功能塊端點。
3. 各種指標型態的變數符號及使用限制請參考下表。

指標型態	使用限制	
一般指標 POINTER	數量限制	每個功能塊定義中，最多可宣告 16 個
	可傳指標	WORD/DWORD/LWORD/INT/DINT/LINT 等型態的變數符號或 D 裝置、L 裝置或 X、Y 等字組裝置 (如 X0、Y0)。
計時器專用指標 T_POINTER	數量限制	每個功能塊定義中，最多可宣告 8 個
	可傳指標	T 裝置或 TIMER 型態的變數符號
計數器專用指標 C_POINTER	數量限制	每個功能塊定義中，最多可宣告 8 個
	可傳指標	C 裝置或 COUNTER 型態的變數符號
高速計數器專用指標 HC_POINTER	數量限制	每個功能塊定義中，最多可宣告 8 個
	可傳指標	HC 裝置或指定配置 HC 裝置的 COUNTER 型態符號

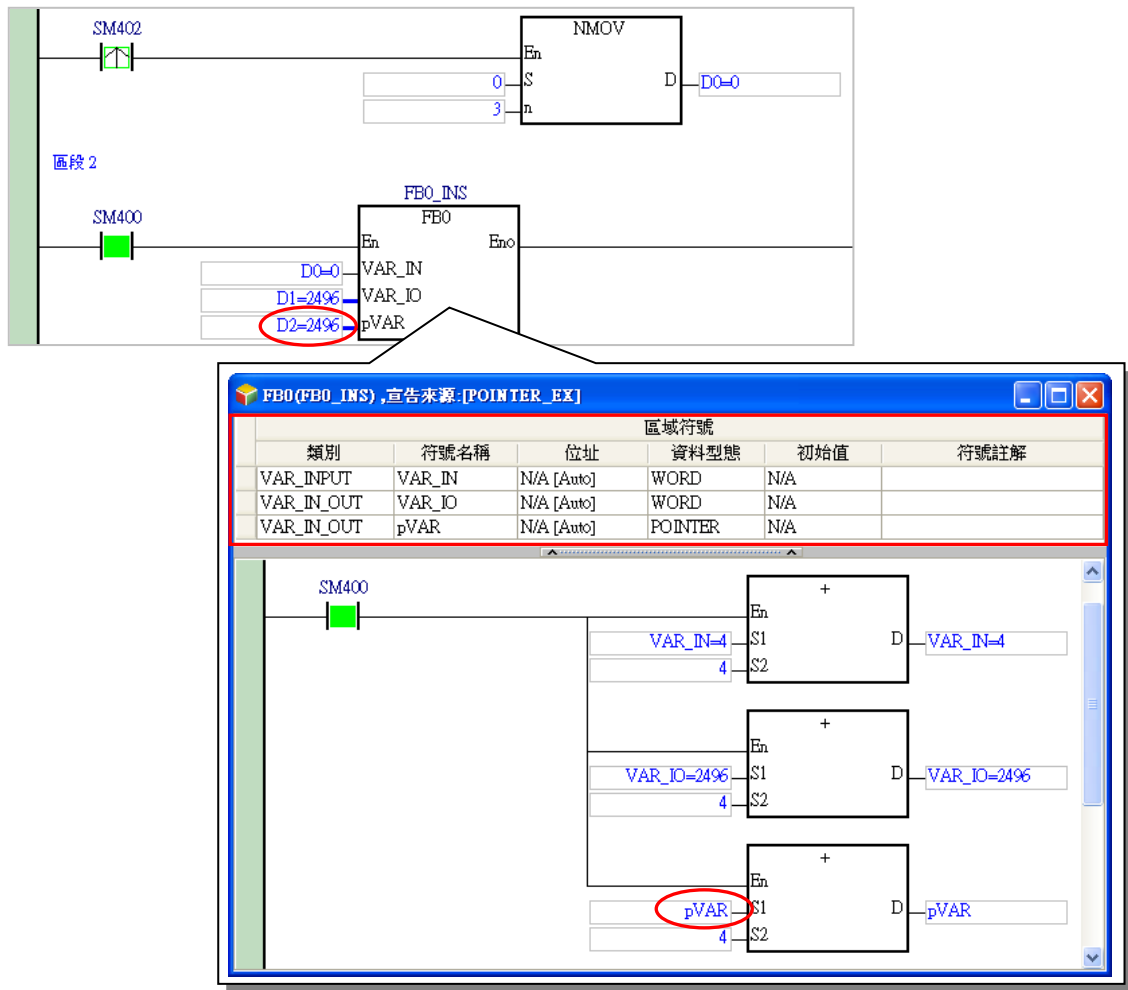
● POINTER 型態符號的使用範例

功能塊 **FB0** 分別有三個輸入端點，其資料型態與類別請參考下圖紅框部份的區域符號表，而呼叫方的 **POU** 則會分別把 **D0**、**D1** 與 **D2** (初始值皆為 0) 指定給這三個端點。下述則為程式執行的狀況。

- **D0** 指定給功能塊端點 **VAR_IN**，且因該符號為 **VAR_INPUT** 類別的 **WORD** 型態，因此每次呼叫功能塊時都會先將裝置 **D0** 的現在值複製給 **VAR_IN**，而經過功能塊內部的運算之後，**VAR_IN** 的值會被加上 4，但這個結果並不會影響呼叫方指定的裝置 **D0**，因此 **D0** 都會固定保持為初始值 0。
- **D1** 指定給功能塊端點 **VAR_IO**，且因該符號為 **VAR_IN_OUT** 類別的 **WORD** 型態，因此每次呼叫功能塊時都會先將裝置 **D1** 的現在值複製給 **VAR_IO**，而經運算之後，**VAR_IO** 的值會被加上 4，且基於 **VAR_IN_OUT** 類別的特性，功能塊執行完畢後會將 **VAR_IO** 的值再度複製回去給裝置 **D1**，因此每次 **D1** 傳進 **VAR_IO** 的值都是累加後的結果，所以 **D1** 的內容會在執行的過程中持續累加。
- **D2** 指定給指標型態的功能塊端點 **pVAR**，而基於指標型符號的特性，當功能塊在執行時會直接將現在 **D2** 的值加上 4 之後，再直接把結果寫回 **D2**，因此在執行過程中，**D2** 的內容會持續累加。

7

雖然 D1 與 D2 的執行結果相同，但實際執行的過程卻有所差異。此外，因 pVAR 的資料內容是指定裝置 D2 的指標，因此在進行線上監控時並不會如同變數符號 VAR_IO 一般顯示目前的數值。



7

*.SM400 為 AH/AS 機種的特殊繼電器，其特性為主機上電後會保持為 ON，而 SM402 則是啟始正向 (RUN 的瞬間為 "ON")

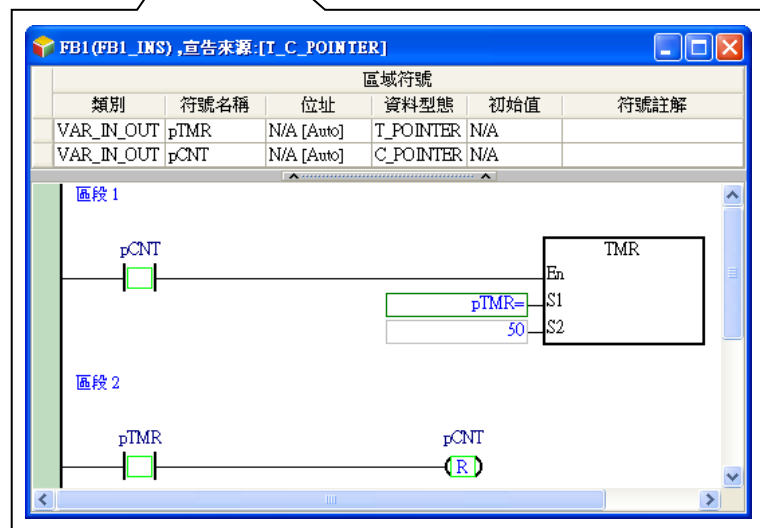
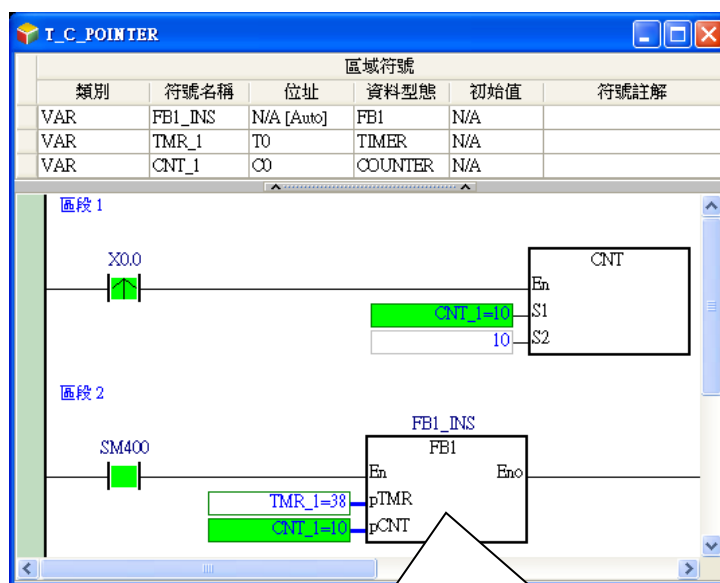
● T_POINTER 與 C_POINTER 型態符號的使用範例

因 AH/AS 系列主機的功能塊內部無法宣告 TIMER 與 COUNTER 型態的變數符號，因此我們可利用 T_POINTER 與 C_POINTER 型態的變數符號將計時/計數器的裝置指標傳進功能塊內部進行運作。

功能塊 FB1 共有 2 個輸入端點，分別是 T_POINTER 型態的 pTMR 及 C_POINTER 型態的 pCNT；而呼叫方則分別把變數符號 TMR_1 (TIMER 型態) 與 CNT_1 (COUNTER 型態) 指定給兩端點。

在呼叫方的程式當中，當 X0.0 由 OFF→ON 時 CNT_1 的計數值便會加 1，一直到 CNT_1 的計數值到達 10 之後，CNT_1 的接點狀態便會為 ON；而功能塊內部因接收了 TMR_1 與 CNT_1 的指標，因此執行時會直接判斷 CNT_1 (pCNT) 的接點狀態，當 CNT_1 (pCNT) 為 ON 時便會啟動 TMR_1 (pTMR) 的計時；當計時到達 5 秒後，CNT_1 (pCNT) 的狀態便會被重置。

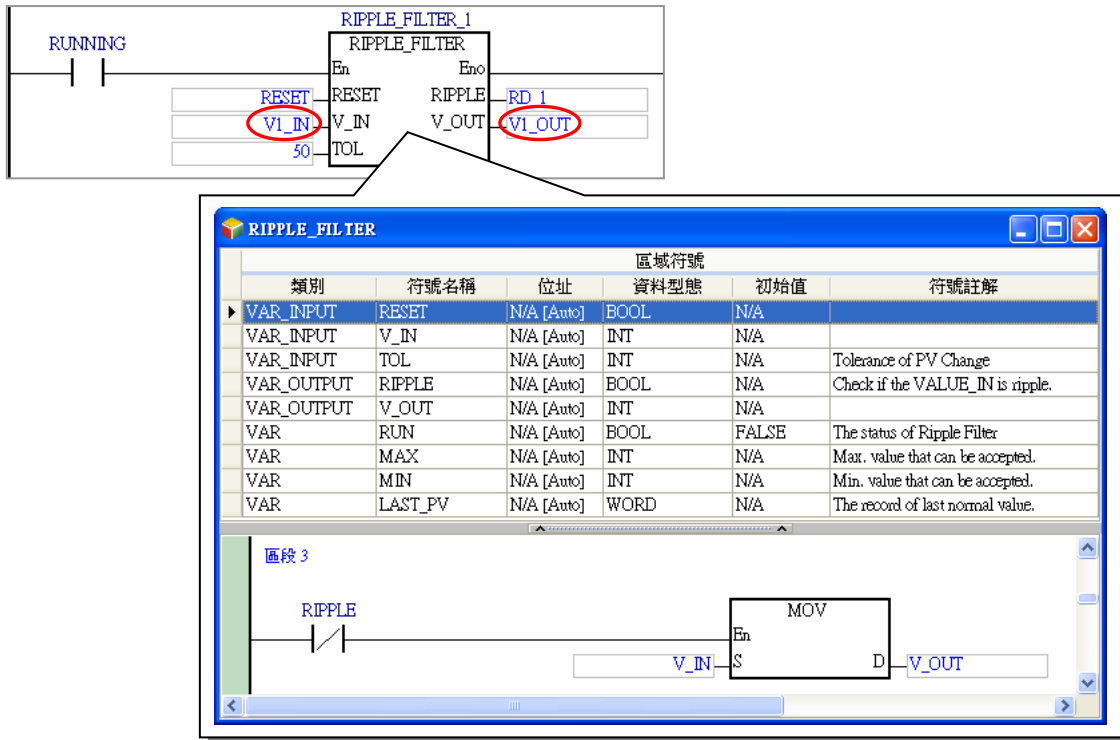
同樣的，指標型符號於線上監控時不會顯示對應裝置的現在值。



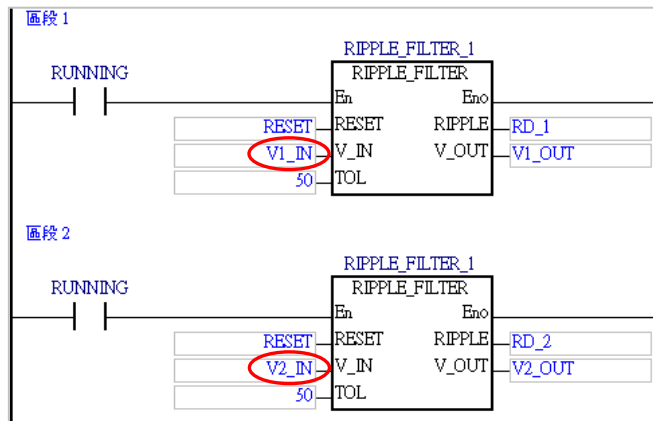
7

7.2.5 功能塊的定義與實例

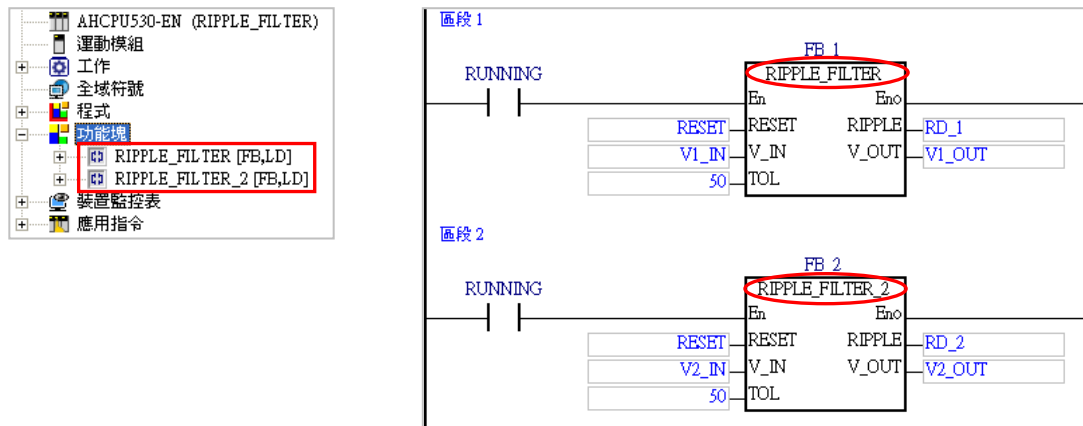
功能塊的**定義與實例**是一個相當重要的觀念，請參考下列的範例。上層 POU (呼叫方) 會在每次執行時將變數符號 **V1_IN** 的值輸入給功能塊 **RIPPLE_FILTER**，接著功能塊內部便會將此次輸入的值與上次記錄的值進行比較後，再將最後的運算結果輸出給上層 POU 的 **V1_OUT**；而為了記錄每次運算的結果，功能塊內部便必須宣告可以儲存運算結果的區域符號，因此進行編譯時，系統便必須為這些區域符號配置一個實體的記憶區塊。



7 接下來考慮一個應用狀況，若上層 POU 有好幾個資料都須要進行相同的處理程序，而各個資料之間亦必須彼此獨立，則若是如下圖的程式，其執行的結果便會是錯誤的，因每次 **V2_IN** 輸入功能塊時，在功能塊內部進行比較的基準卻是上次運算 **V1_IN** 的結果，所以兩者的資料將會彼此干擾。



而若是重新建立一個內容完全相同的**功能塊 POU**，雖然編譯時會為兩個**功能塊 POU** 的區域符號配置不同的**記憶區塊**，而執行的結果也會是正確的，然而在編程的程序上便會產生許多重複的工作。



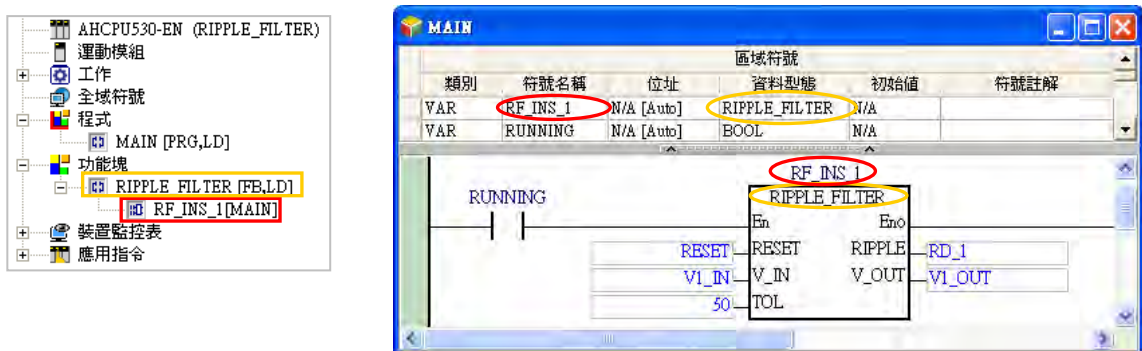
有鑑於此，在 IEC61131-3 中便導入了**功能塊實例**的觀念。

當我們在**專案管理區**中新增了一個**功能塊 POU**，並在此 POU 當中進行區域符號的宣告以及程式的編輯，而完成之後所產生的成品稱便之為「**功能塊定義**」，但其意義僅如同一份文件而已，本身並不會參與實際的運算，同時也不會佔用 PLC 運作的任何資源。



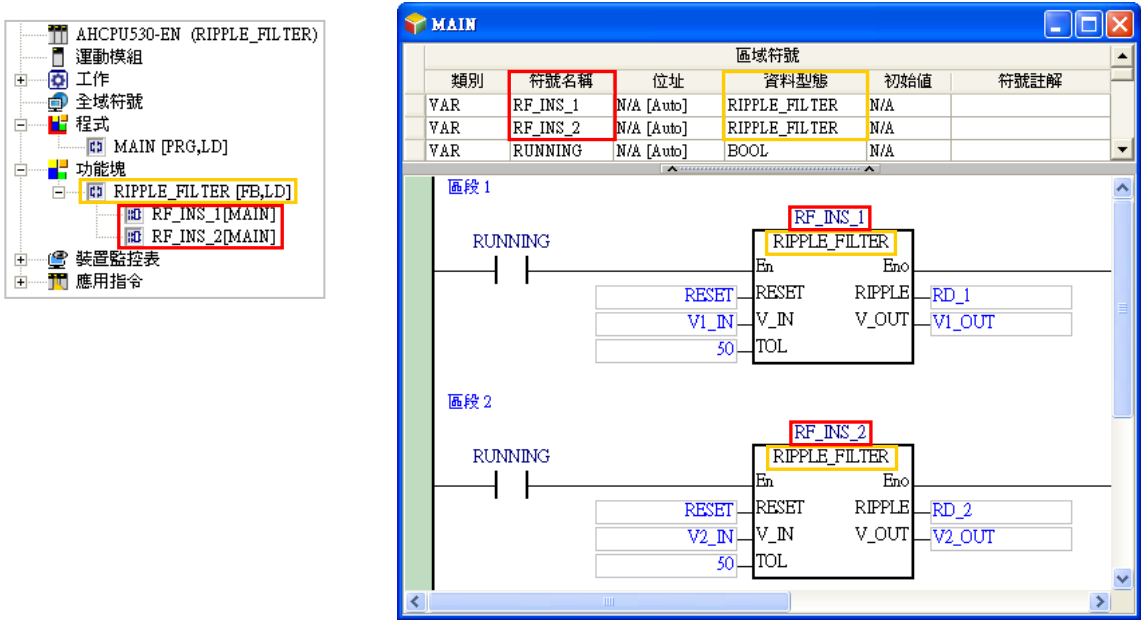
當要在某個程式 POU 當中呼叫該**功能塊**時，使用者便必須先宣告該**功能塊**型態的變數符號，此時便代表要依據「**功能塊定義**」來製造一個會參與實際運作的物件，其所產出的成品便稱之為「**功能塊實例**」，而其中**功能塊**型態的符號名稱即是「**功能塊實例**」的名稱。

當進行編譯時，系統便會依據定義而為**功能塊實例**與其內部的區域符號配置一個實體的**記憶區塊**。

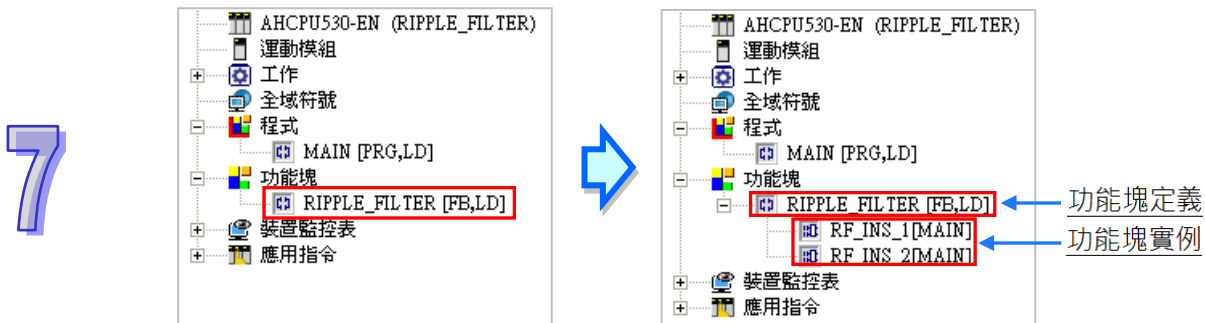


7

再重新考慮之前的狀況，當運作上必須要有相同的運算程序，但又需要彼此獨立的記憶區塊，操作上便可利用功能塊實例的特性，只要在呼叫功能塊時，為需要不同記憶區塊的功能塊，另外宣告一個不同名稱的功能塊符號即可，系統便會將不同的功能塊符號名稱視為不同的功能塊實例，而在編譯時便會為其配置獨立的記憶區塊。



如下圖，專案程式在編譯前僅會有**功能塊定義**的項目，而編譯後，於該定義的下層便會產生實際參與程式運作的**功能塊實例**，其顯示格式為 **實例名稱[呼叫方 POU 名稱]**。



與一般的變數符號相同，只要在符號表中宣告了一個功能塊型態的變數符號，之後即使在編輯的程式當中沒有使用到，編譯時仍會為其產生對應的功能塊實例；此外，在 AH5X0 系列機種中，被程式 POU 所調用的「**功能塊實例**」數量合計不得超過 1024 個。(未產生實例的**功能塊定義**則可不計)，但其他機種系列則無此限制，直到記憶體被配置完畢為止。

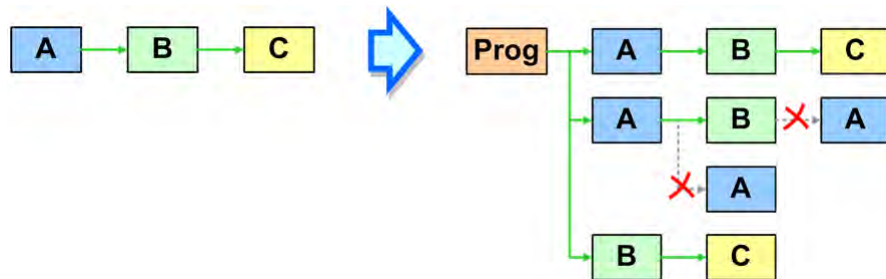
綜合上述，功能塊的操作包含了「**定義**」與「**實例**」兩種觀念。「**定義**」屬於規劃的層面，而「**實例**」則偏向執行的層面；以產品為例，「**功能塊定義**」相當於工程人員針對開發產品所訂定的產品規格，而「**功能塊實例**」則相當於依據此規格所生產出來的產品。

7.2.6 功能塊呼叫功能塊的巢式結構

在 ISPSOft 的功能塊規格中，功能塊可以呼叫功能塊，在 AH 系列機種最多可允許呼叫至 32 層，在 AS/DVP 系列機種最多可允許呼叫至 16 層，而 DVPxxMC 與 AS5xx 機種則沒有此限制，而在認定上，由程式 POU 所呼叫的功能塊實例會被定義為第一層。

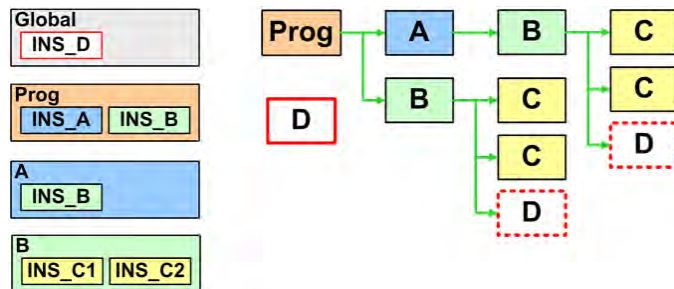


如下圖的結構，在功能塊 A 的程式中呼叫了功能塊 B，而在功能塊 B 的程式中也呼叫了功能塊 C，此時功能塊 A 便是功能塊 B 與 C 的上層功能塊，而功能塊 B 本身也是 C 的上層功能塊；且因 ISPSOft 不支援自己呼叫自己的結構，所以一旦功能塊 A 是功能塊 B 的上層，則在功能塊 B 以下的所有下層就不允許再呼叫功能塊 A，但所有下層功能塊都可以在最上層的程式 POU 當中被呼叫。



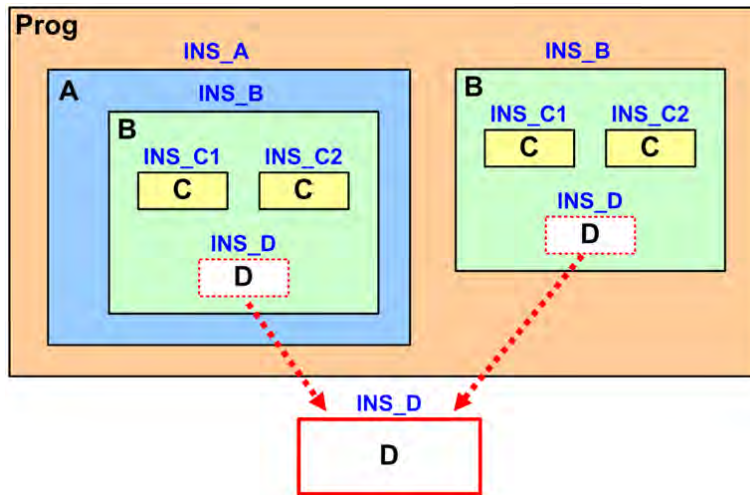
當功能塊 A、B、C 皆未被任何程式 POU 所呼叫時，即使功能塊 B 被 A 所呼叫，而 C 也被 B 所呼叫，但因三者皆不會參與程式的實際運作，所以編譯時都不會產生**功能塊實例**；直到在某個程式 POU 中或在全域符號表中宣告了功能塊 A 的實例後，系統才會在編譯時依據三者的呼叫關係而產生對應的**功能塊實例**。下列將以一範例說明在功能塊呼叫功能塊的結構下，系統的功能塊實例將如何產生。

請參考下列功能塊實例的宣告狀況及各功能塊之間的呼叫關係。其中除了功能塊 D 的實例被宣告於全域符號表之外，其餘的功能塊實例皆被宣告於上層 POU（呼叫方）的區域符號表中。



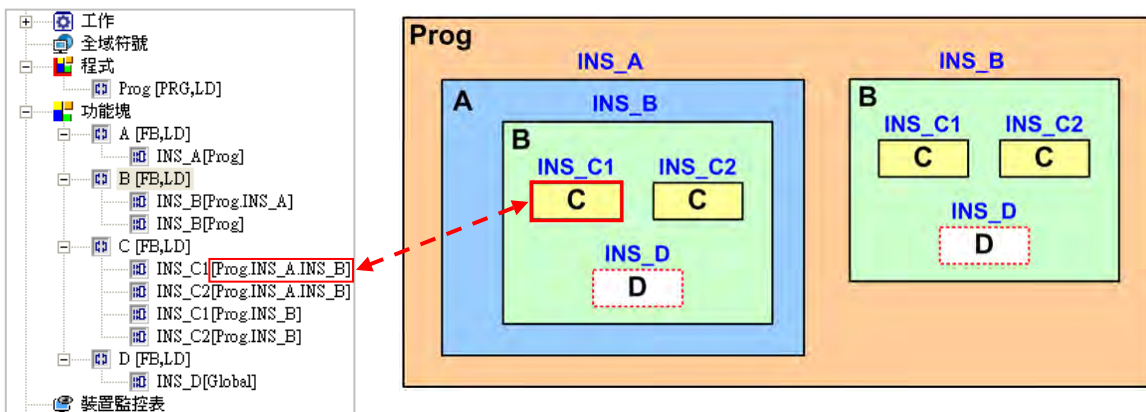
宣告位置		宣告項目
全域符號		1 個功能塊 D 實例：INS_D
區域符號	程式 Prog	1 個功能塊 A 實例：INS_A 1 個功能塊 B 實例：INS_B
	功能塊 A	1 個功能塊 B 實例：INS_B
	功能塊 B	2 個功能塊 C 實例：INS_C1 及 INS_C2

下列示意圖為系統依據上述的呼叫關係而在編譯時所產生的功能塊實例。且在下列的圖示中可發現，因功能塊 C 的兩個實例被宣告於功能塊 B 的區域符號表中，因此當有不同的功能塊 B 實例被產生時，就如同細胞分裂一般，宣告於其區域符號表的功能塊 C 實例也會等比的被產生出來。因為功能塊 D 的實例被宣告於全域符號表內，所以該實例的功能與其內部儲存的資料狀態便會讓所有的 POU 一起共用，因此系統只會產生一個功能塊 D 的實例，且雖然功能塊 D 的實例會在兩個地方被呼叫，但實際在兩處程式當中運作的都會是同一個功能塊 D 的實例。



下圖則為實際進行編譯後，在專案管理區中所顯示的項目，其中在每個功能塊實例後方的中括號內，其文字的內容會由最上層的程式 POU 開始，依序往下列出各層呼叫方的 POU 名稱。

7



在 DVP 機種中，功能塊實例不可宣告在功能塊的區域符號表中，而當要在功能塊內部呼叫其他功能塊時，下層功能塊實例便必須宣告在全域符號表，因此操作時請小心避免運算資料的錯亂，若需要獨立的記憶區塊時，請於全域符號表中宣告不同的功能塊實例。

在 AH/AS 的機種中，功能塊實例允許宣告在功能塊內部的區域符號表中，因此若非所呼叫的下層功能塊實例也必須與其他的 POU 共用，否則可直接將該下層功能塊的實例宣告在區域符號表中即可。且在 AH/AS 系列的機種中，功能塊實例的宣告位置將影響系統配置記憶體區塊時所佔用的資源。

7.2.7 功能塊的記憶體配置

當功能塊內部有宣告為自動配置位址的區域符號時，在編譯及產生實例的過程中，系統便會為這些區域符號配置一個獨立的記憶體區塊；若功能塊內部宣告的區域符號有自行指定裝置位址，則該變數符號便會被視為絕對位址，而在編譯時便不會另外為其配置記憶體區塊，之後即使在不同的功能塊實例當中，只要與該變數符號有關的程序，系統都會以使用者自行指定的裝置內容來參與運算。

當系統在配置資源時，DVP 系列、DVPxxMC 系列、AS5xx 系列、AH5X0 系列、AH5X1 系列、AH560 備援系列、AHxxEMC 系列與 AS 系列機種所採用的機制並不相同，請參考下述說明。

● DVP 系列機種

DVP 機種在配置記憶體區塊時，每個實例都會被自動配置一個 P 裝置，但若該 POU 的名稱是以“P0_”或“P1_”做為開頭，如“P0_FB”，則編譯後，該實例將被強制配置 P0 或 P1 裝置，其他依此類推；而後續若該功能塊被宣告了一個以上的實例，則編譯時便會因為重複使用相同的 P 裝置而造成錯誤。針對功能塊內部所宣告的區域符號，系統將會依據每個變數符號的資料型態而為其配置對應的裝置位址，例如 WORD 型態會配置一個 D 裝置，而 BOOL 型態則會配置一個 M 裝置，因此 DVP 機種的實例大小會依其所宣告的區域符號而定，且不同的實例會佔用不同的裝置位址。下圖便是一個 DVP 機種所產生的兩個實例，而在開啟功能塊實例的視窗後，於其各自的區域符號表中便可看到系統會為不同的實例配置不同的裝置位址。

The image shows a software interface for configuring function blocks. On the left is a project tree with 'INS_0[Prog0]' and 'INS_1[Prog0]' highlighted. In the center is a ladder logic diagram with two function blocks, 'INS_0' and 'INS_1', each receiving a trigger (TRIG_1, TRIG_2) and outputting data (DT0_OUT, DTI_OUT). Below are two windows showing the variable address tables for 'INS_0' and 'INS_1'.

INS_0 Variable Address Table:

類別	符號名稱	位址	資料型態	初始值	符號註解
VAR_INPUT	V_IN	D7000	WORD	0	
VAR_OUTPUT	V_OUT	D7001	WORD	0	
VAR	TEMP	M3003	BOOL	FALSE	

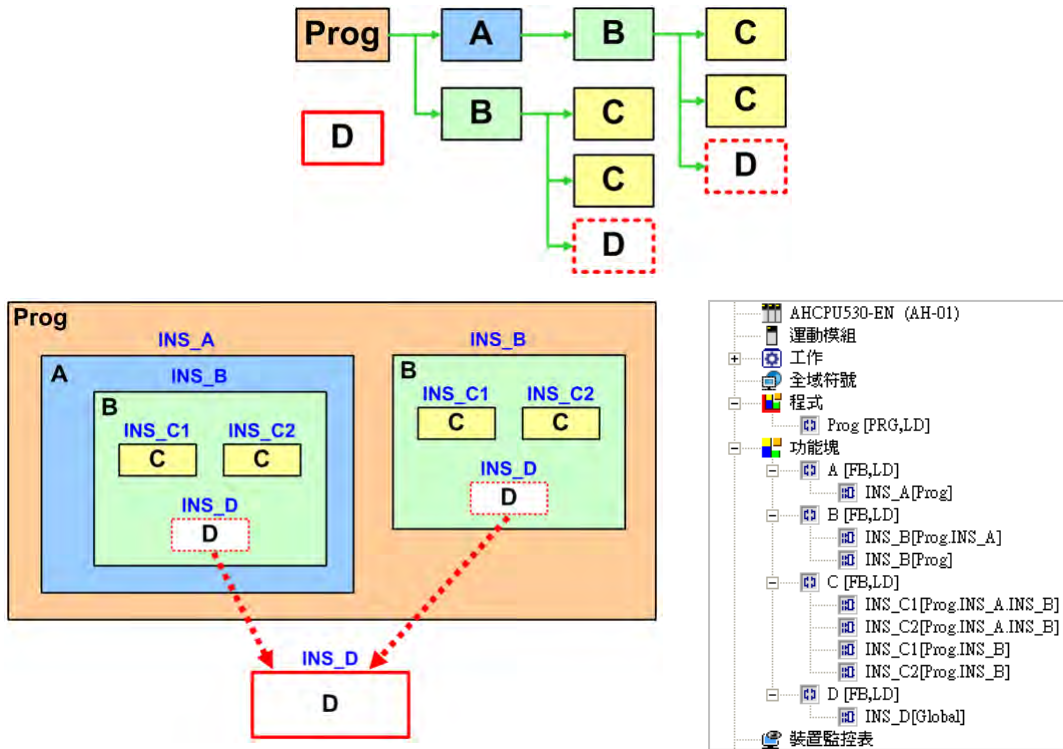
INS_1 Variable Address Table:

類別	符號名稱	位址	資料型態	初始值	符號註解
VAR_INPUT	V_IN	D7006	WORD	0	
VAR_OUTPUT	V_OUT	D7007	WORD	0	
VAR	TEMP	M3007	BOOL	FALSE	

7

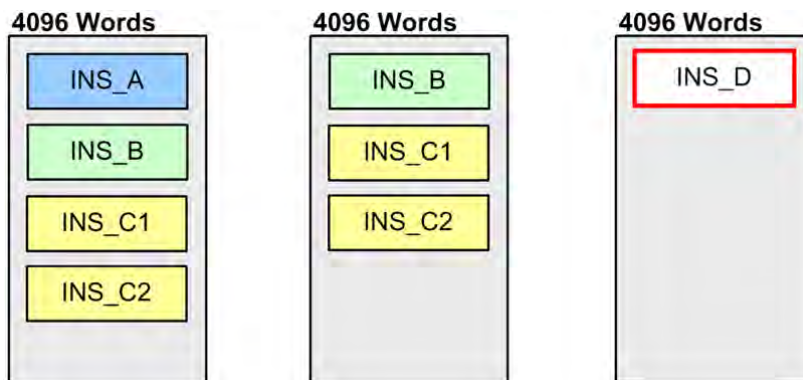
● AH5X0 系列機種

在 AH5X0 系列機種中，系統將會在內部的記憶體中預留固定數量的「實例記憶區塊」供功能塊實例使用，而數量的大小將依機種而定，且個有產生實例的功能塊定義會佔用一個 P 裝置；其中每個區塊的容量大小都會固定為 4096 Words，且因配置的位置是主機內部預留的記憶體，因此使用者無法藉由開啟功能塊實例的視窗而得知系統配置的結果。下列為第 7.2.6 節範例中的功能塊呼叫結構。



以 AH5X0 機種來說，對於在程式 POU 的區域符號表中所宣告的第一層功能塊實例，系統會單獨配置一個 4096 Words 大小的「實例記憶區塊」，之後包括第一層的功能塊實例與其下層呼叫的所有功能塊實例，其所需的所有記憶體都會被分配在同一個 4096 Words 的區塊中；另外，針對宣告於全域符號表的功能塊實例，系統亦會單獨為其配置一個相同大小的「實例記憶區塊」。

而根據上述的範例，系統總共會配置如下 3 個 4096 Words 大小的「實例記憶區塊」。



AH5X0 機種為功能塊實例所配置的「實例記憶區塊」會存放在主機另外預留的內部記憶體中，而配置時並不會佔用使用者可用的裝置資源，且不論功能塊實例的大小，所配置的「實例記憶區塊」都會固定為 4096 Words，因此在編程時請小心避免超過可用的區塊大小。計算的方式如下。

- (1) 每個功能塊實例都會固定佔用最基本的 2 Words 大小。
- (2) 在每個實例的區域符號表中，單獨針對每個變數符號，依據其資料大小計算每個變數符號所佔用的 Word 數，未滿 1 個 Word 以 1 個 Word 計算。至於功能塊型態的變數符號（即下層的功能塊實例）則可於上述（1）中一起統計，此處可先略過不計。

請參考下列的範例，其中各個功能塊定義所宣告的區域符號亦統計於下列表格中。



下列則為此範例的計算過程。

- (1) 每個功能塊的實例會固定佔用 2 Words 的大小，而本範例總共產生 4 個功能塊實例，因此共計佔用 8 Words。
- (2) 接著計算各個功能塊實例的區域符號表中，所有非功能塊型態的變數符號所佔用的記憶體大小。

➤ 功能塊實例 **INS_A**

➔ 2 Words (DWORD x 1 個) + 4 Words (LREAL x 1 個)，總計為 $2 + 4 = 6$ (Words)。

➤ 功能塊實例 **INS_B**

➔ 1 Word (WORD x 1 個) + 4 Words (REAL x 2 個)，總計為 $1 + 4 = 5$ (Words)。

➤ 功能塊實例 **INS_C1 / INS_C2**

因每個變數符號必須單獨計算，且未滿 1 個 Word 必須計為 1 個 Word，所以計算每個 BOOL 型態的變數符號時都必須以 1 個 Word 來計算；而長度為 22 的 BOOL 陣列，因長度超過 1 個 Word，但未滿 2 個 Word，因此必須計為 2 個 Word。計算如下。

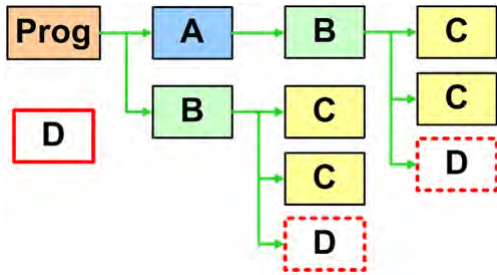
➔ 2 Words (BOOL x 2 個) + 2 Words (長度為 22 的 BOOL 陣列 x 1 個)，總計為 $2 + 2 = 4$ ；

而功能塊 C 總共產生了 2 個實例，所以兩個功能塊實例合計為 $2 \times 4 = 8$ (Words)。

統計總和之後，可得知此區塊對應的功能塊實例所佔用的記憶體為 $8 + 6 + 5 + 8 = 27$ (Words)。

● AH5X1 系列、AH560 備援系列、AHxxEMC 系列與 AS 系列機種

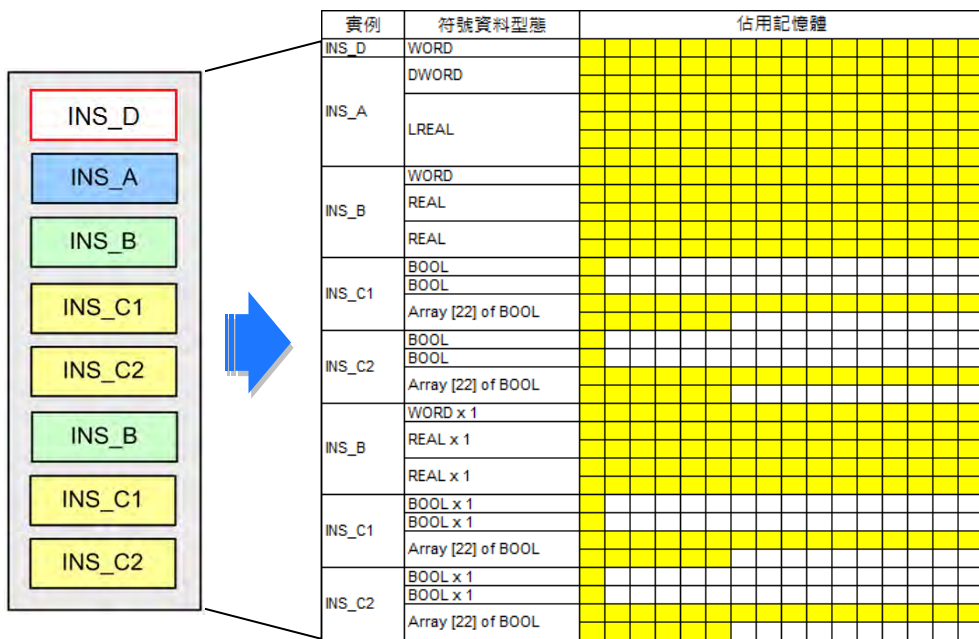
AH5X1 系列、AH560 備援系列、AHxxEMC 系列與 AS 系列機種功能塊的記憶體，則會存放在主機另外預留的內部記憶體中，而配置時並不會佔用使用者可用的裝置資源，與 AH5X0 系列的不同之處為 AH5X1 系列、AHxxEMC 系列與 AS 系列機種不會限制實例的大小或個數，而是將此內部記憶體配置完為止，且每個有產生實例的功能塊定義會佔用一個 P 裝置。下列使用第 7.2.6 節範例，其中各個功能塊定義所宣告的區域與全域符號資料型態數量亦統計於下列表格中。



功能塊定義	區域/全域符號數量
功能塊 A	DWORD x 1 個、LREAL x 1 個、下層功能塊實例 x 1 個
功能塊 B	WORD x 1 個、REAL x 2 個、下層功能塊實例 x 2 個
功能塊 C	BOOL x 2 個、長度為 22 的 BOOL 陣列 x 1 個
功能塊 D	WORD x 1 個

因此內部記憶體佔用的方式將如下圖所示

7



下列則為此範例的計算過程。

(1) 所有的實例依宣告的順序佔用內部記憶體，首先加入宣告在全域符號表中的功能塊實例 INS_D。

➢ 功能塊實例 INS_D

➔ 1 Words，總計為 1 (Words)。

(2) 接著依照呼叫的順序計算各個區域符號表中，所有非功能塊型態的變數符號所佔用的記憶體大小。

➤ 功能塊實例 INS_A

→ 2 Words (DWORD x 1 個) + 4 Words (LREAL x 1 個)，總計為 2 + 4 = 6 (Words)。

➤ 功能塊實例 INS_B

→ 1 Word (WORD x 1 個) + 4 Words (REAL x 2 個)，總計為 1 + 4 = 5 (Words)。

➤ 功能塊實例 INS_C1 / INS_C2

因每個變數符號必須單獨計算，且未滿 1 個 Word 必須計為 1 個 Word，所以計算每個 BOOL 型態的變數符號時都必須以 1 個 Word 來計算；而長度為 22 的 BOOL 陣列，因長度超過 1 個 Word，但未滿 2 個 Word，因此必須計為 2 個 Word。計算如下。

→ 2 Words (BOOL x 2 個) + 2 Words (長度為 22 的 BOOL 陣列 x 1 個)，總計為 2 + 2 = 4；而功能塊 C 總共產生了 2 個實例，所以兩個功能塊實例合計為 2 x 4 = 8 (Words)。

➤ 功能塊實例 INS_B / INS_C1 / INS_C2

→ 如同前述 INS_B 佔用 5 (Words)；INS_C1 / INS_C2 佔用 8 (Words)。所以三個功能塊實例合計為 5 + 8 = 13 (Words)。

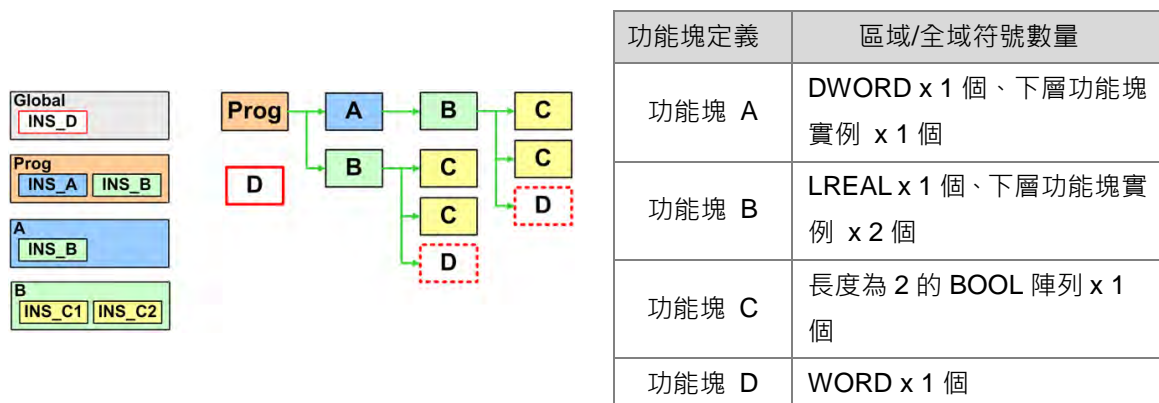
統計總和之後，可得知此區塊對應的功能塊實例所佔用的記憶體為 1 + 6 + 5 + 8 + 13 = 33 (Words)。

● DVPxxMC 系列與 AS5xx 系列機種

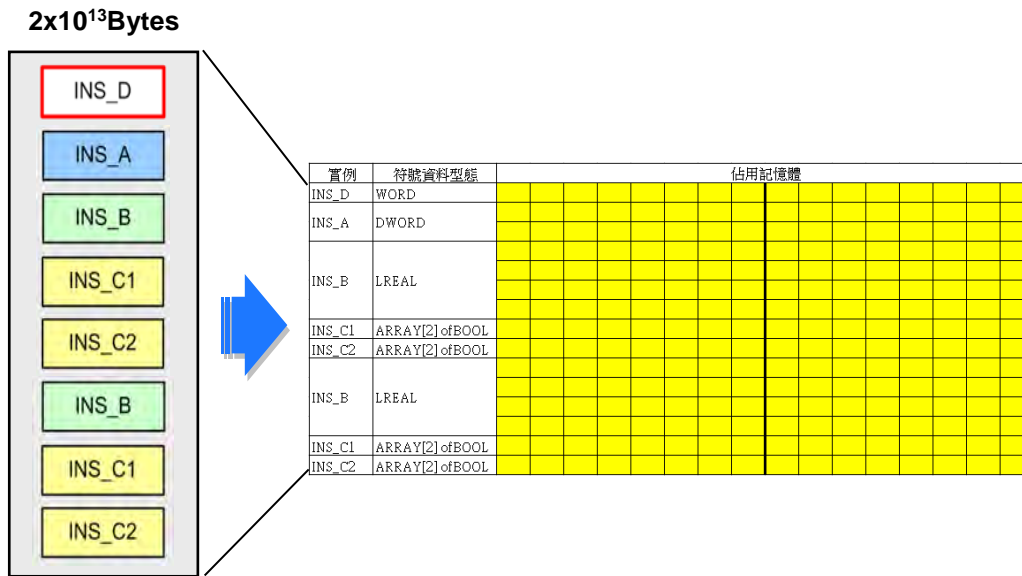
DVPxxMC 與 AS5xx 機種在配置記憶區塊時，實例會分配在記憶體之某一區塊，直到此區塊佔滿 2×10^{13} Byte 的記憶體空間為止，但使用者須注意，記憶區塊仍會被 POU 之數量所影響，當使用者定義 FB 之數量越多，相對地，在引用每一定義之 FB 所宣告之實例數，將會有所限制。

在每個實例的區域符號表中，單獨針對每個變數符號，依據其資料大小計算每個變數符號所佔用的 Byte 數，在未滿 1 個 Byte 的情況，若沒有綁定裝置以 1 個 Byte 計算；若變數符號資料型態為 BOOL 且有綁定裝置，裝置型態若為 Bit 則以 Bit 計算，裝置型態若為 Byte 則以 Byte 計算。至於功能塊型態的變數符號 (即下層的功能塊實例)。

下列為第 7.2.6 節範例中的功能塊呼叫結構，詳細內容請參考該節的相關說明。其中各個功能塊定義所宣告的區域與全域符號資料型態數量亦統計於下列表格中。



因此內部記憶體佔用的方式將如下圖所示。



下列則為此範例的計算過程。

➤ 功能塊實例 **INS_A**

→ 4 Bytes (DWORDx 1 個) · 總計為 5 (Bytes)。

➤ 功能塊實例 **INS_B**

→ 8 Bytes (LREAL x 1 個) · 總計為 8 (Bytes)。

➤ 功能塊實例 **INS_C1 / INS_C2**

BOOL 陣列，每個 ARRAY 的 BOOL 元素將固定占掉 1 個 Byte。計算如下。

→ 2 Bytes (長度為 2 的 BOOL 陣列 x 1 個) · 總計為 2；而功能塊 C 總共產生了 2 個實例，所以兩個功能塊實例合計為 $2 \times 2 = 4$ (Bytes)。

統計總和之後，可得知此區塊對應的功能塊實例所佔用的記憶體為 $5 + 8 + 4 = 17$ (Bytes)。

7.3 功能塊的使用

7.3.1 功能塊的基本規格

下列表格為 ISPSOft 的功能塊規格，使用者在使用功能塊時，請務必確認所有的操作都確實符合下表所規範的內容，否則便有可能造成編譯或執行上的錯誤。

● 功能塊規格

功能塊定義數	全機種：無限制。
功能塊實例數	AH5X1/AH560 備援系列/AHxxEMC/AS：決定於可用控制器裝置資源。 AH5X0：最大的「實例記憶區塊」數量會依機種而定 (請參考附錄 B)；決定於可用控制器裝置資源。 DVP：決定於可用控制器裝置資源。

功能塊容量	AH5X1/ AH560 備援系列/AS：無限制（直到內部記憶體被配置完畢為止） AH5X0：每一個「實例記憶區塊」所提供的記憶體容量固定為 4096 Words DVP：決定於可配置的裝置資源。
巢狀結構層數	AH 系列機種：功能塊呼叫功能塊的巢狀結構，最大允許至 32 層、 AS/DVP 系列機種：最大允許至 16 層、 DVPxxMC 與 AS5xx 機種：則沒有限制。

● 注意事項

- 所有的功能塊端點都必須被指定一個運算元，該運算元可為裝置位址或變數符號，而當端點的符號類別為 VAR_INPUT 時，所指定的運算元亦可為常數。
- 不可在下層功能塊中呼叫上層功能塊，或是自己呼叫自己。
- 匯出功能塊時僅包含該功能塊的定義，下層功能塊的定義與使用的全域符號則不會一併匯出。
- 除了 ST 語言之外，使用功能塊時的 En 接點前方必須連接至有邏輯狀態的接點或區塊。
- 支援建立功能塊的語言有 LD、IL、ST、FBD 及 CFC，但不包含 SFC。
- 可呼叫使用功能塊的語言有 LD、IL、ST、FBD 及 CFC，而在 SFC 的 Action 程式中也可使用功能塊。
- 使用者只能修改功能塊定義而無法對功能塊實例進行修改。(請參考第 7.3.4 節)
- 對功能塊定義進行線上編輯時，若修改的結果造成上層 POU 的功能塊圖示異常，此時必須先離線並將該圖示刪除後再重新加入，接著才可重新編譯並下載程式。(請參考第 7.3.4 節)
- 使用跳躍指令如 CJ、JMP...時，其目的地僅限同層功能塊的內部，而不可跳出功能塊外部或跳至下層功能塊的內部。
- 在 AH/AS 系列機種的功能塊內部，使用者不可宣告 TIMER 與 COUNTER 型態的變數符號，但可以使用指標型態的變數符號。
- AH/AS 機種的功能塊內部不可使用脈波型的指令(如 MOVP)，而其他不可使用的應用指令尚有 LDP/ANDP/ORP、LDF/ANDF/ORF、PLS、PLF、NP、PN、MC、MCR、GOEND。
- AH/AS 機種本身有提供在功能塊內部專用的脈波型指令，詳細內容請參考下節的相關說明。

*1. 新增功能塊 POU 的方法請參考第 5.4.1 節的介紹；而宣告功能塊實例的方法則請參考第 6.2.2 節的說明。

*2. 呼叫功能塊的方法請參考第 10 章~第 15 章中各編程語言的相關介紹。

7.3.2 功能塊專用的脈波型指令 (AH/AS 機種限定)

因 AH/AS 系列機種的功能塊內部無法使用脈波型的指令，如 LDP、LDF、NP、PN 等，因此在應用上可考慮搭配下列的指令來達到相同的效果。關於各指令的使用說明及限制請參考各機種程式手冊。

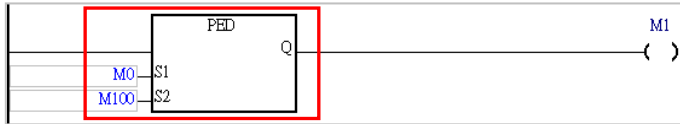
● NED 及 PED

其功能類似 LDP 及 LDF，唯 NED 與 PED 需另外指定一個 BOOL 格式的暫存裝置，且被用來做為暫存功能的裝置在整個專案中都不可再被使用於其他地方，否則便可能產生運算錯亂。

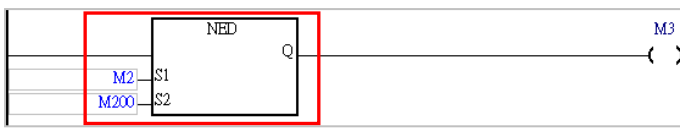
- 於程式 POU 中可使用 LDP 指令 (上升緣觸發) 與 LDF 指令 (下降緣觸發) 。



- 於功能塊內部可使用 PED 來取代 LDP，但必須多指定一個暫存用的裝置，如下圖的 M100。



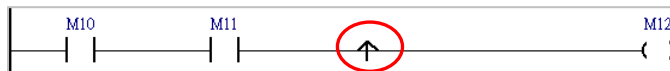
- 於功能塊內部可使用 NED 來取代 LDF，但必須多指定一個暫存用的裝置，如下圖的 M200。



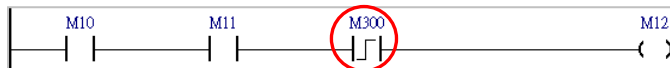
● FB_NP 及 FB_PN

其功能類似 NP 及 PN，唯 FB_NP 與 FB_PN 需另外指定一個 BOOL 格式的暫存裝置，且被用來做為暫存功能的裝置在整個專案中都不可再被使用於其他地方，否則便可能產生運算錯亂。

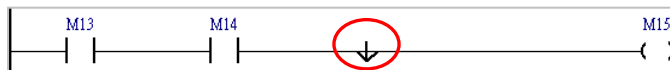
- 於程式 POU 中可使用 NP 指令。



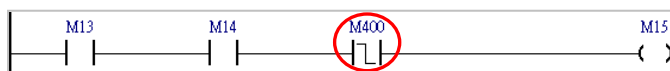
- 於功能塊內部可使用 FB_NP 來取代 NP，但必須多指定一個暫存用的裝置，如下圖的 M300。



- 於程式 POU 中可使用 PN 指令。



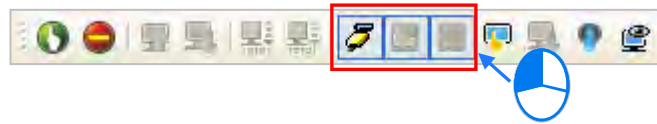
- 於功能塊內部可使用 FB_PN 來取代 PN，但必須多指定一個暫存用的裝置，如下圖的 M400。



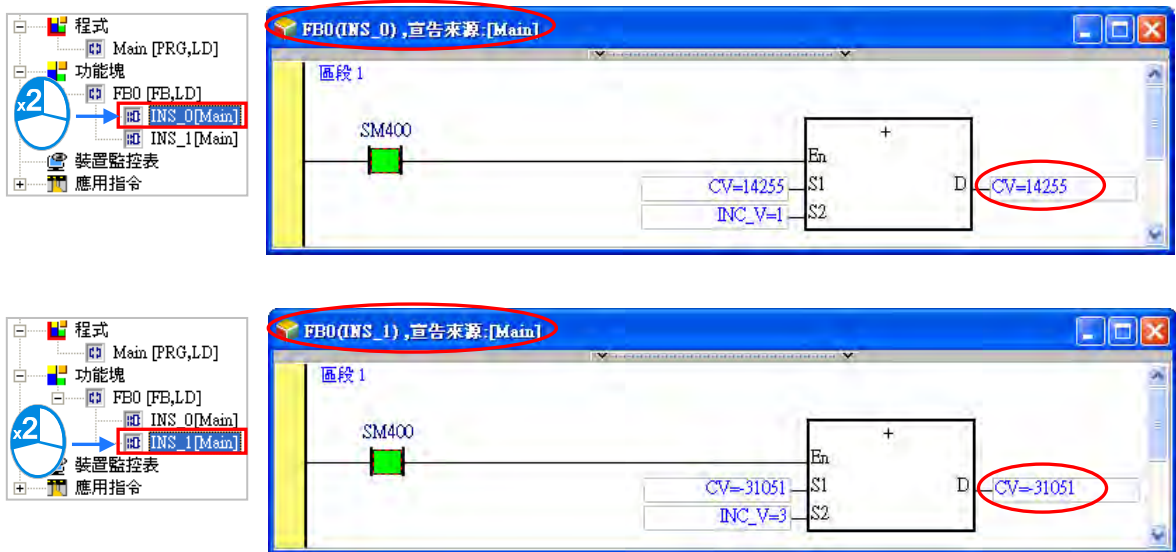
7

7.3.3 功能塊的程式監控

因為在程式執行時，實際參與運作的是「**功能塊實例**」，所以當進行線上的程式監控時，必須開啟「**功能塊實例**」的視窗才可正常進行監控。



如下圖，當進行程式監控時，必須開啟「**功能塊實例**」的視窗才可進行監控；且由下圖可發現，不同的功能塊實例，其運作過程是彼此獨立的。



若開啟「**功能塊定義**」的視窗時，該視窗中將不會有任何監控值的顯示，因此無法進行程式監控。



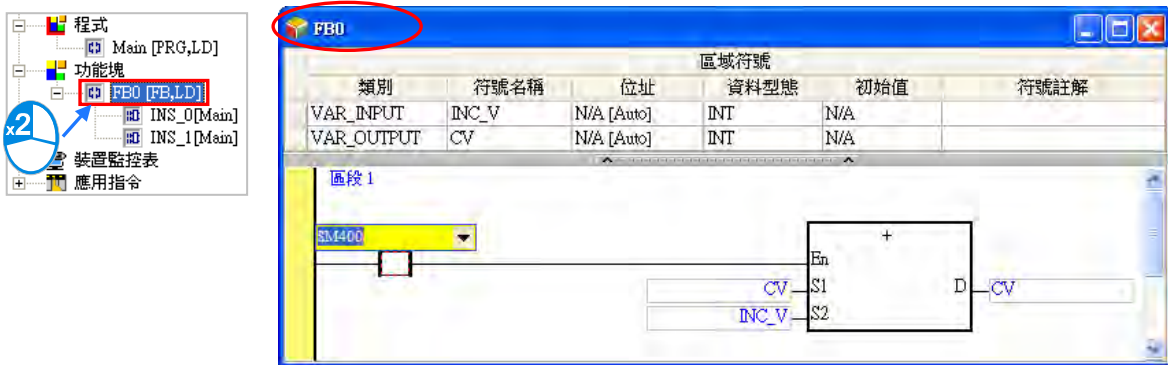
*.進行線上編輯時，則必須於「**功能塊定義**」的視窗中才可以修改功能塊的內容，相關細節請參考第 7.3.4 節的說明。

7

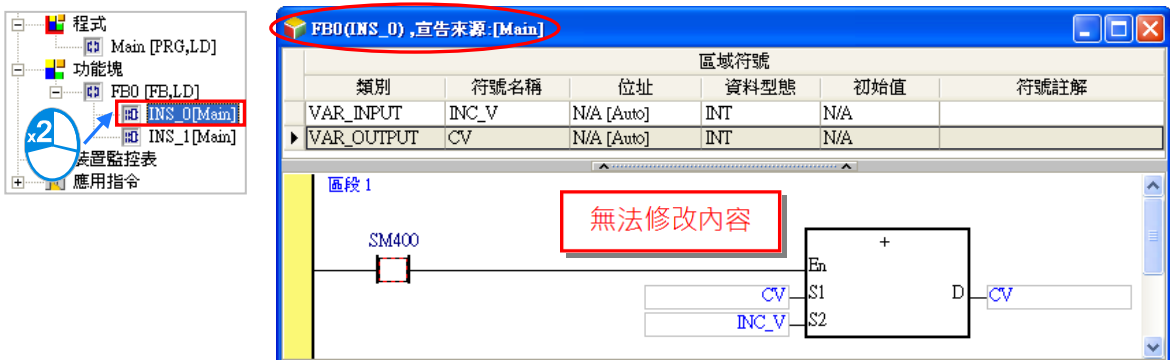
7.3.4 功能塊的修改

因「功能塊實例」是依據「功能塊定義」所產生的物件，因此「功能塊實例」中的內容將無法直接修改，而必須在「功能塊定義」的視窗中才可以修改內容；當然，所修改的內容將會套用在所有由該功能塊定義所產生的實例中，且在修改之後，仍必須經過編譯方可產生新的「功能塊實例」。同樣的，進行線上編輯時，亦必須開啟「功能塊定義」的視窗才可修改內容。

如下圖，於「功能塊定義」處雙擊滑鼠左鍵，之後便可於「功能塊定義」的視窗中修改程式內容。

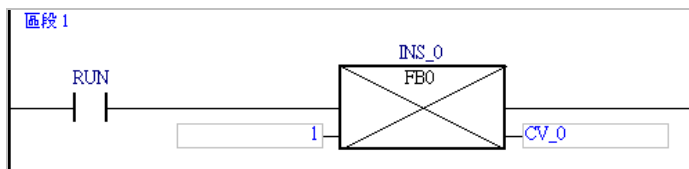


若開啟的是「功能塊實例」視窗，則使用者將無法於此處修改程式內容。



7

當某個功能塊已經被其他的 POU 所呼叫時，則刪除該功能塊便可能造成上層 POU 中的功能塊圖示產生異常（如下圖），此時只要將該圖示刪除即可。



7.4 範例實作

在第 4 章的內容中，我們曾以一個簡單的範例來示範如何建立傳統的階梯圖程式，並在第 5.6.1 節中以 POU 與 TASK 的程式架構重新規劃，之後又在第 6.3 節中使用變數符號來取代原本的裝置定址；而在接下來的內容中，我們將延續第 6.3 節中所建立的範例，並在範例中示範如何規劃並使用功能塊。

7.4.1 程式規劃

下表為第 6.3 節的範例程式中，各 POU 的功能簡介。

POU 名稱	功能說明
RUN_STOP	當按下 啟動按鈕 時讓設備開始運轉，而按下 停止按鈕 或偵測到錯誤發生時，設備便會停止運轉。
GLUING	共包含兩個相同的工作站程序，內容為當 到位感測器 為 ON 時，便將 注射器的觸發信號 設置為 ON；而當 到位感測器 變為 OFF 時，便將該信號重置回 OFF。
COUNTING	計數工件的入料數量，當入料數量到達 100 時，便將 完成旗標 設置為 ON。

根據各 POU 的程序特性，我們預計規劃如下的兩個功能塊。

- 在「**GLUING**」POU 中共包含了兩段相同的程序，因此我們可以將該程序建立為一個功能塊，之後便可重複使用；而所完成的功能塊將會於「**GLUING**」POU 當中被呼叫，且因兩個工作站之間的運作應該彼此獨立，因此我們會為其宣告兩個獨立的功能塊實例。
- 在「**COUNTING**」POU 中的程序，除了**運轉旗標**與**完成旗標**之外，本身的程序並不會與其它程序有太多的聯結，因此獨立性相當的高，所以我們可以將其建立為功能塊以增加程式的可讀性。

7.4.2 建立範例程式

接下來可直接開啟第 6.3 節當中的範例來進行修改或是重新建立程式。若要重新建立程式時，請先新增一個專案，之後參考第 4 章至第 6 章的範例介紹，並配合接下來的說明逐一完成此節的範例。

首先請參考第 5.4.1 節的說明，於專案管理區的「功能塊」項目下新增一個 POU - 「**EQUIP_TRIG**」。



7

功能塊「**EQUIP_TRIG**」主要是負責點膠的功能。接著請參考第 6.2.2 節的說明，於區域符號表中建立如下的 2 個端點符號，並參考第 4 章至第 6 章的範例介紹，完成下圖的程式。

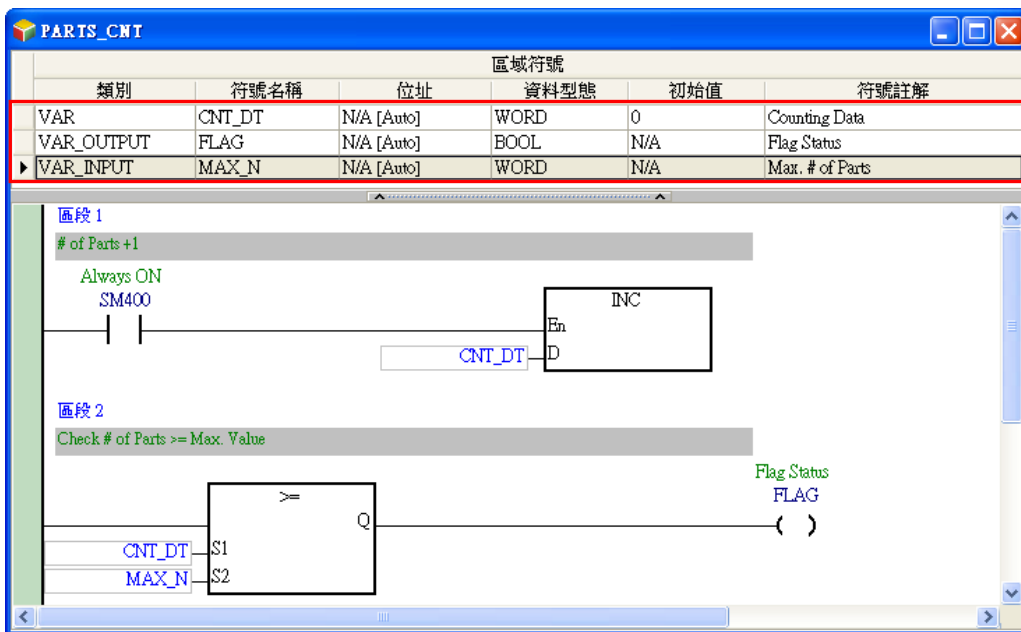


再來請於專案管理區的「功能塊」項目下新增另一 POU - 「**PARTS_CNT**」。

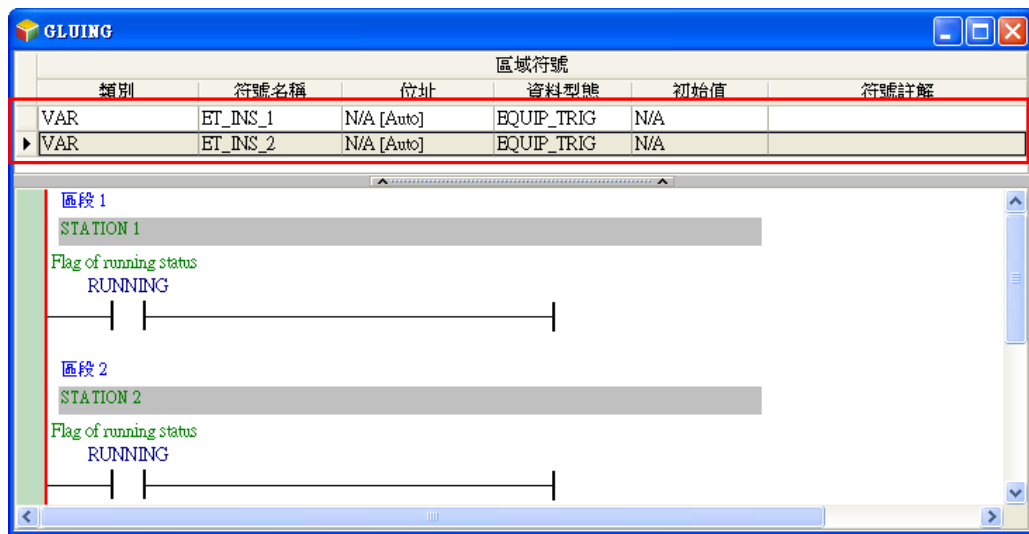


功能塊「**PARTS_CNT**」的功能則是負責計數入料工件的數量。接著請參考第 6.2.2 節的說明，於區域符號表中建立如下的 3 個符號，而其中 CNT_DT 的初始值請將其設定為 0；之後請參考第 4 章至第 6 章的範例介紹，完成下圖的程式。

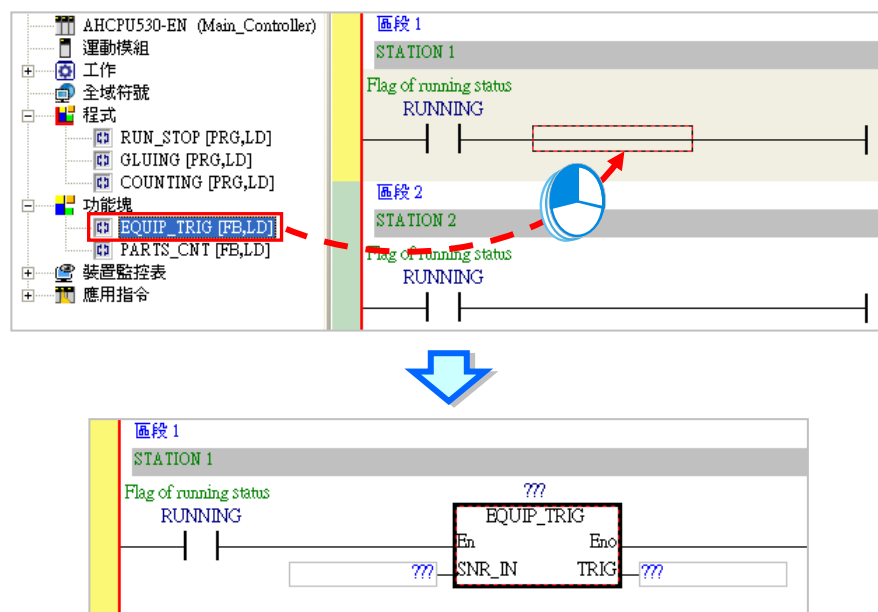
7



再來我們便要在「GLUING」POU 當中呼叫功能塊「EQUIP_TRIG」。首先請參考第 6.2.2 節的說明於「GLUING」POU 的區域符號表中宣告如下圖的 2 個功能塊實例，並將程式修改如下。

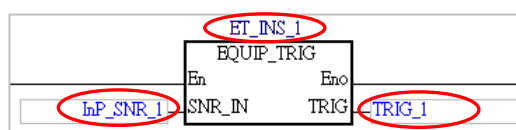


要在程式當中插入功能塊的方式與插入應用指令 API 的方式類似，直接由專案管理區中，將準備插入的功能塊「EQUIP_TRIG」拖曳至欲插入的位置即可。而若該功能塊的下層已產生實例時，拖曳時請注意必須拖曳功能塊的定義，而不可拖曳其下層的功能塊實例項目。

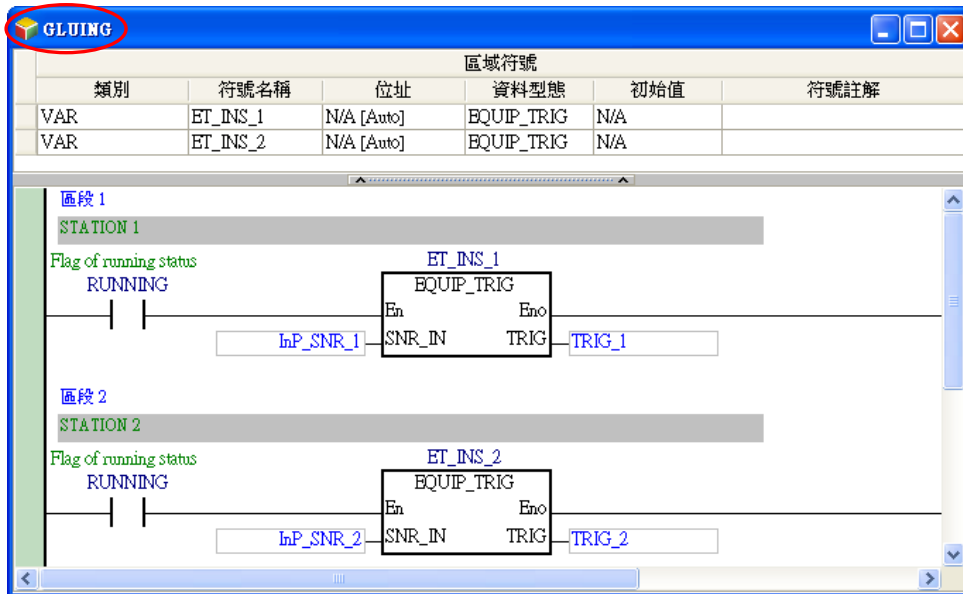


7

插入功能塊後，緊接著請逐一指定功能塊實例與運算元。

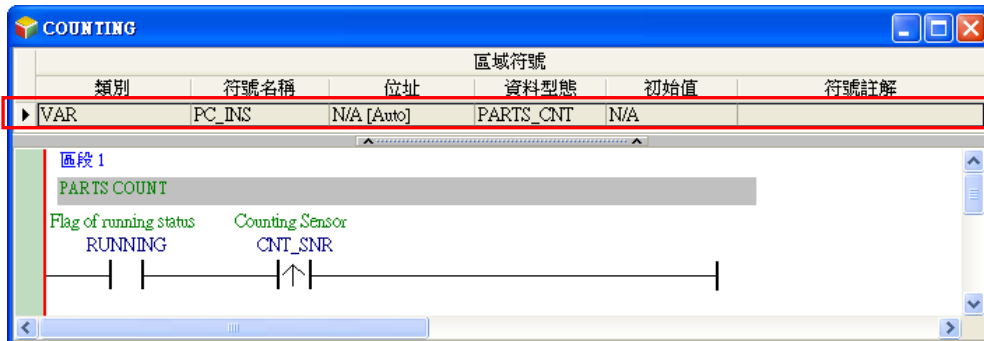


接下來請依照上述的說明完成「GLUING」POU 的修改，而完成的程式如下。

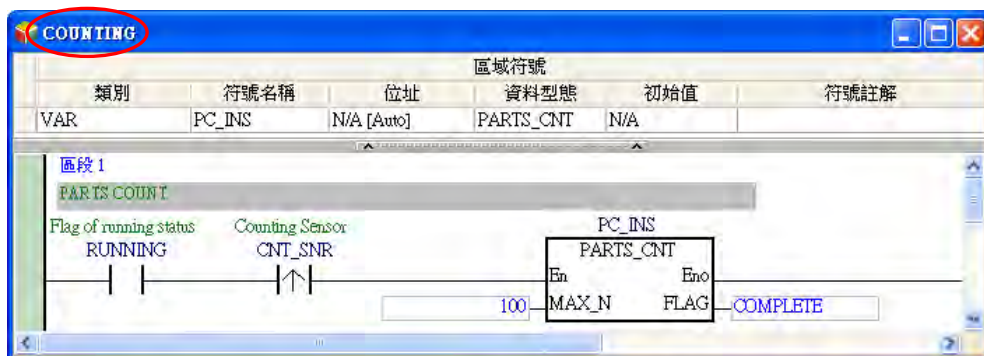


我們接著繼續在「COUNTING」POU 當中呼叫功能塊「PARTS_CNT」。首先請於「COUNTING」POU 的區域符號表中宣告如下的功能塊實例，而原本存在的區域符號「CNT_DT」則已沒有作用，所以請將其刪除，之後請將「COUNTING」POU 的程式修改如下。

7



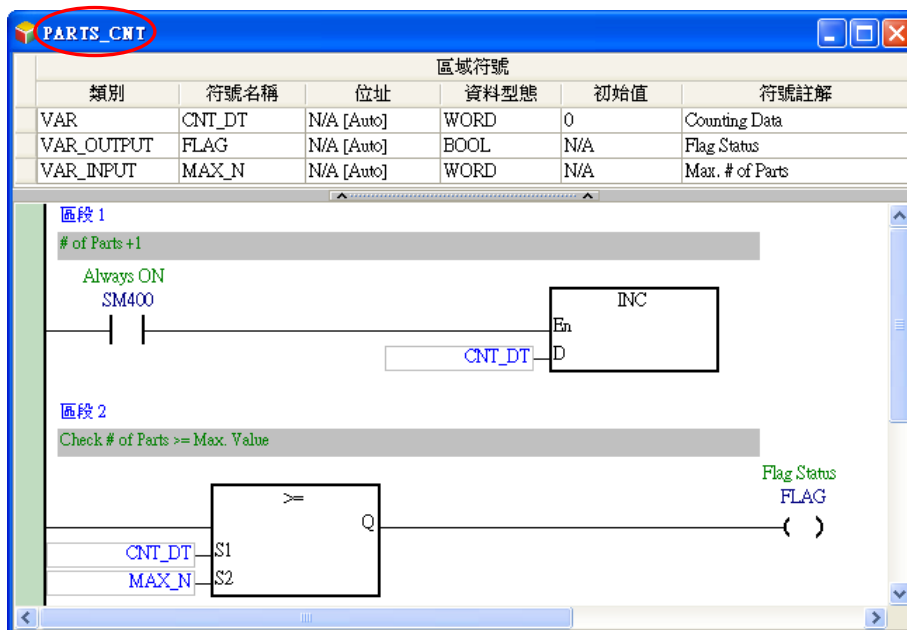
接著請參考前述的說明插入功能塊「PARTS_CNT」，並完成「COUNTING」POU 的修改。



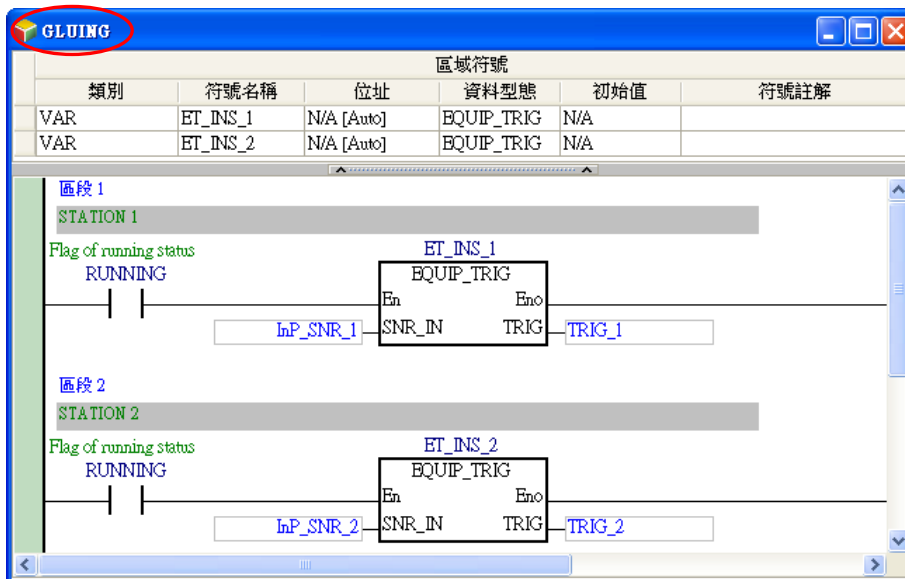
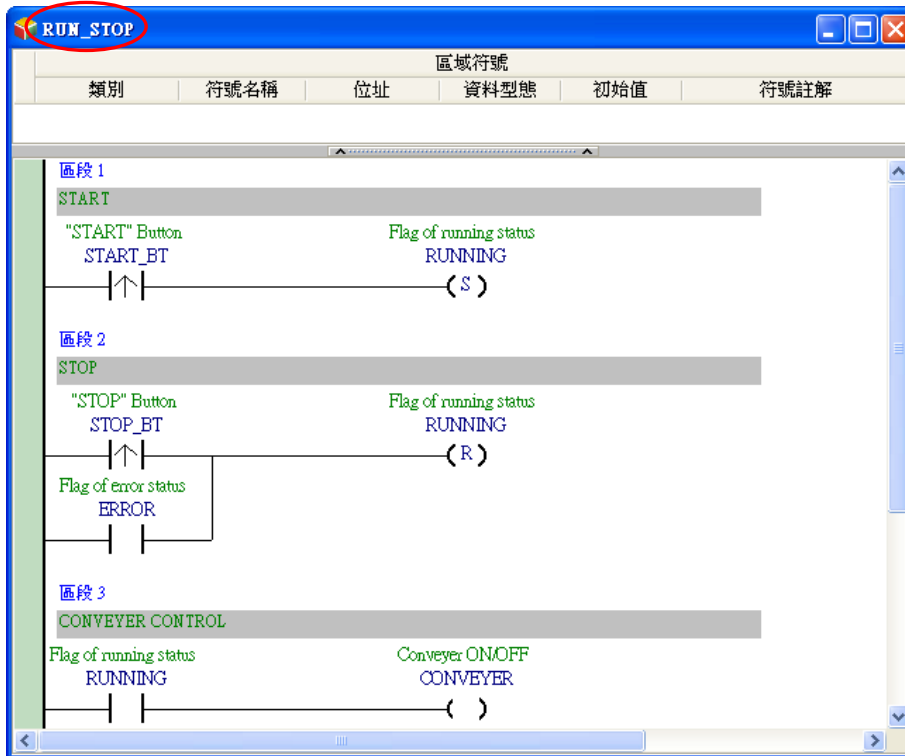
到目前為止，此節的範例程式已編輯完成，接著請再次比對各位所編輯的程式與下列各項是否相同。



類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	START_BT	X0.0	BOOL	N/A	"START" Button
VAR	STOP_BT	X0.1	BOOL	N/A	"STOP" Button
VAR	InP_SNR_1	X0.2	BOOL	N/A	In-position Sensor 1
VAR	InP_SNR_2	X0.3	BOOL	N/A	In-position Sensor 2
VAR	CNT_SNR	X0.4	BOOL	N/A	Counting Sensor
VAR	CONVEYER	Y0.0	BOOL	N/A	Conveyer ON/OFF
VAR	TRIG_1	Y0.1	BOOL	N/A	Trig. signal 1
VAR	TRIG_2	Y0.2	BOOL	N/A	Trig. signal 2
VAR	RUNNING	N/A [Auto]	BOOL	N/A	Flag of running status
VAR	COMPLETE	N/A [Auto]	BOOL	N/A	Flag of completion
VAR	ERROR	N/A [Auto]	BOOL	N/A	Flag of error status



7

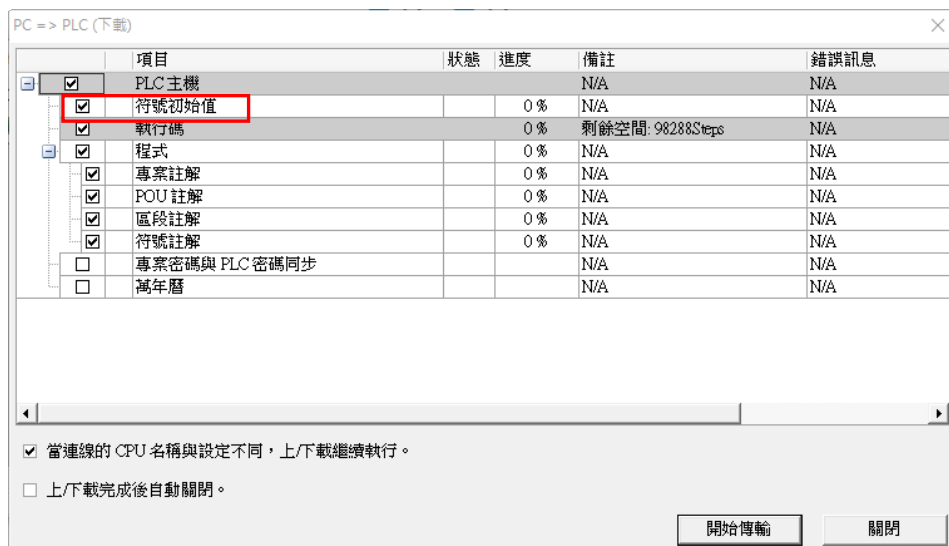


7

在確認編輯的程式與本節所列的範例相同後，請試著先執行編譯，而編譯完成後便可發現於專案管理區中會多出功能塊實例的物件。



接著請確認已完成硬體組態、硬體參數設定、工作 (TASK) 管理...等工作，若否，則請依據第 4 章至第 6 章的範例，逐一完成編輯與設定；完成後便可重新執行編譯，並將硬體參數與程式下載至主機當中進行測試。下載程式時，請一併勾選「符號初始值」，以便將功能塊「PARTS_CNT」中的區域符號「CNT_DT」的初始值一併寫入主機中。

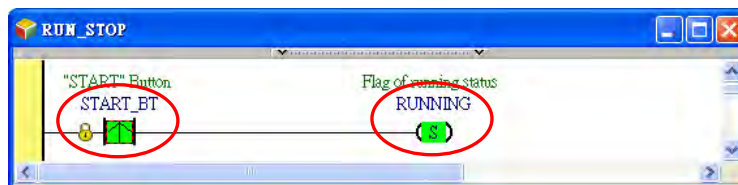
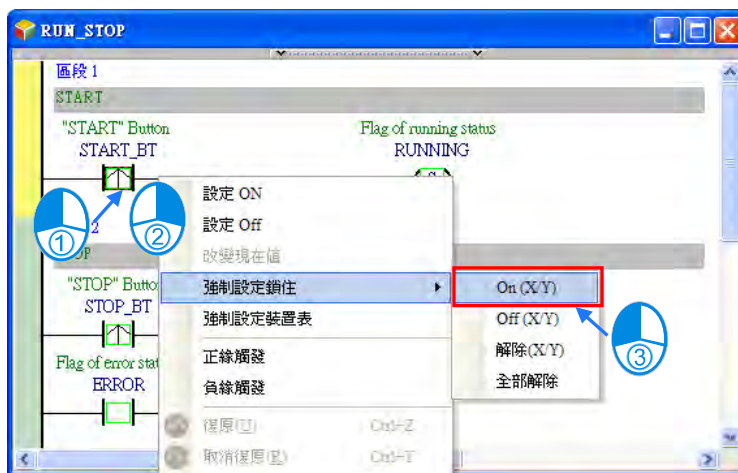


7

完成下載之後，我們試著進入程式監控的狀態，並將主機狀態設為「**RUN**」，接著請依據下列的步驟來進行功能塊特性的驗證。

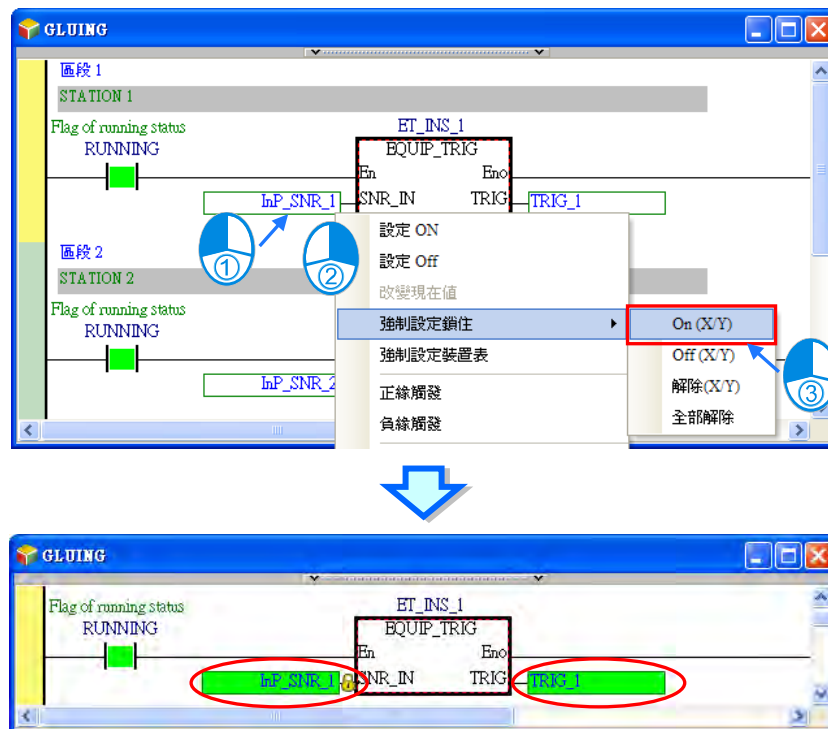


進入程式監控畫面並啟動 PLC 後，請先開啟「**RUN_STOP**」POU 的視窗，並將「**START_BT**」的狀態強制設定為 ON，此時「**RUNNING**」旗標便會為 ON，代表設備已經開始運轉。



7

接著請再開啟「GLUING」POU 的視窗，並將「InP_SNR_1」的狀態也強制設定為 ON，而此時的「TRIG_1」狀態也會跟著變為 ON，代表感測器已感測到工件，並將觸發信號送出。

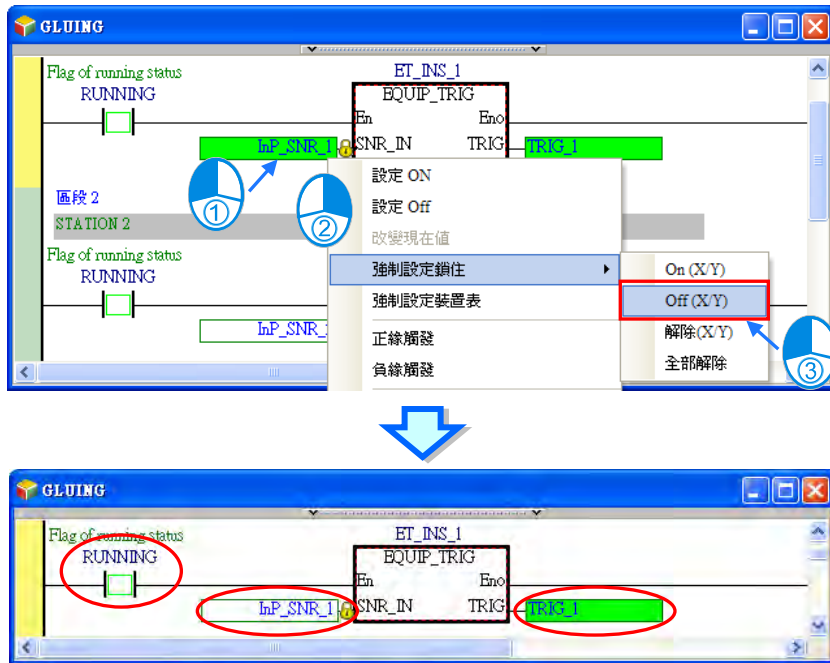


再來請至「RUN_STOP」POU 的視窗中，並將「STOP_BT」的狀態強制設定為 ON，而此時旗標「RUNNING」的狀態便會恢復為 OFF，代表設備已停止運轉。



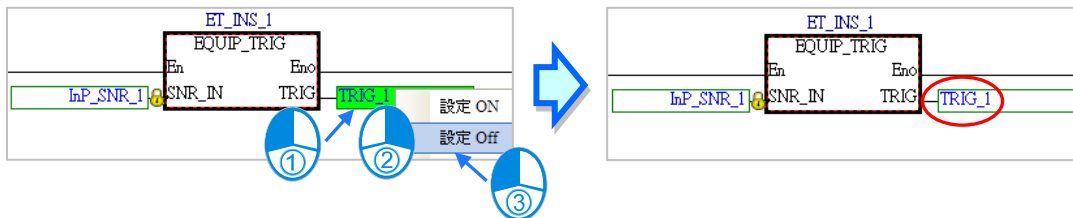
7

接著請再回至「GLUING」POU 的視窗，並將「InP_SNR_1」的狀態強制設定為 OFF，但此時輸出「TRIG_1」的狀態卻仍會維持為 ON；而此狀況則是正確的，主要是因為連接至功能塊 EN 接點的旗標「RUNNING」已是 OFF 狀態，而這也代表功能塊將不會被執行，因此功能塊並不會將其「TRIG」端點的值輸出給變數符號「TRIG_1」，所以「TRIG_1」的狀態會維持原狀。

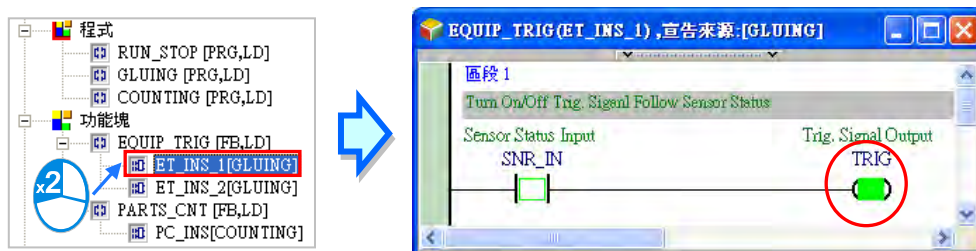


接著我們直接將「TRIG_1」的狀態設置為 OFF。

7



然後開啟功能塊實例「ET_INS_1」的視窗。而由功能塊實例的監控畫面可發現，「TRIG」端點的狀態仍會維持 ON，這代表當功能塊實例不被執行時，其內部的變數符號仍會保持在最後執行的狀態。



接著我們再開啟功能塊實例「ET_INS_2」的視窗並與「ET_INS_1」進行比對，可發現兩者的狀態並不相同，而這也代表不同功能塊實例之間的運作是彼此獨立的。



依據上述的操作，我們可驗證下述三件事。

1. 當 EN 接點的狀態為 OFF 時，功能塊內部的程序將不會被執行。
2. 功能塊內部的變數符號狀態會在執行結束後被保留下來。
3. 不同功能塊實例之間的運作是彼此獨立的。

不過很明顯的，目前程式的執行狀況已與原本不同，在第 6.3 節的範例中，當「RUNNING」旗標為 OFF 時，「TRIG_1」的狀態會被設置為 OFF；而在本節的範例中，當「RUNNING」旗標為 OFF 時，「TRIG_1」會保持在最後的狀態；不過這也是事先就已預期且可接受的狀況，主要是因為在設備的規劃中，將「TRIG_1」設置為 OFF 的目的是為了能進行下次的觸發，因此即使在設備不運轉的狀況下，「TRIG_1」的狀態保持原狀也不會影響設備原本的功能性。關於設備功能的規劃，請參考第 4.1 節的相關說明。

7.4.3 函式 (Function) 範例程式

宣告一個 FUN0 的函式，返回值為 INT 型態。其中 FUN0 定義兩個輸入 INT 型態的符號相加後再加 1，並將返回值填入 FUN0。用法如下圖：

FUN0			
			區域符號
類別	符號名稱	位址	資料型態
VAR_INPUT	input1	N/A [Auto]	INT
VAR_INPUT	input2	N/A [Auto]	INT

```

0001 FUN0 := input1 + input2 + 1;

```

Prog0			
			區域符號
類別	符號名稱	位址	資料型態
VAR	mydata1	N/A [Auto]	INT
VAR	mydata2	N/A [Auto]	INT
VAR	myresult	N/A [Auto]	INT

```

0001 mydata1 := 2;
2
0002 mydata2 := 3;
3
0003 myresult := FUN0(mydata1, mydata2);
6          2      3

```

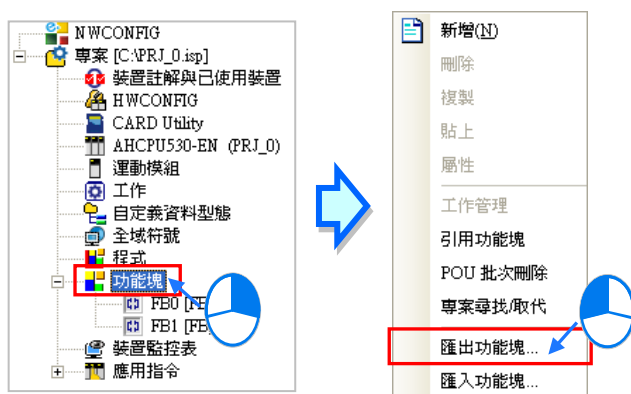
7

7.5 認識函式庫 (Library)

函式庫(Library)包括了使用者自訂函式庫(User Defined Library)與台達函式庫(Delta Library)。有了前述章節功能塊的概念之後，對於常用的功能塊，使用者可以將其匯出成為使用者自訂函式庫(User Defined Library)，以便在不同的專案中重複使用這些功能塊。而台達也對於各種應用場合提供一系列的功能塊，存放在台達函式庫(Delta Library)中，供使用者直接運用。

7.5.1 新增使用者自訂函式庫

當於專案中建立了功能塊，希望在不同的專案中重複使用時，便可利用匯出到使用者自訂函式庫的功能。選擇專案管理區的功能塊項目，點擊滑鼠右鍵開啟快捷選單，點選「匯出功能塊」。

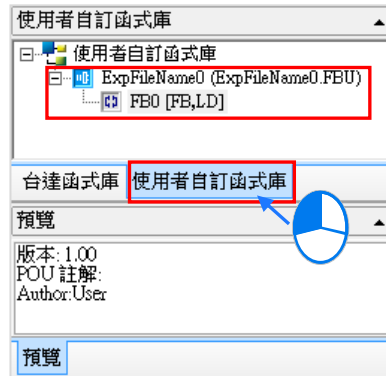


將出現匯出功能塊視窗，依序輸入檔案名稱、勾選匯出至使用者自訂函式庫及要匯出的功能塊；檔案名稱為用於函式庫的分類，同一檔案名稱可匯出多個功能塊函式庫。完成設定後按下「確定」按鈕，再於匯出成功視窗按下「確定」按鈕完成。

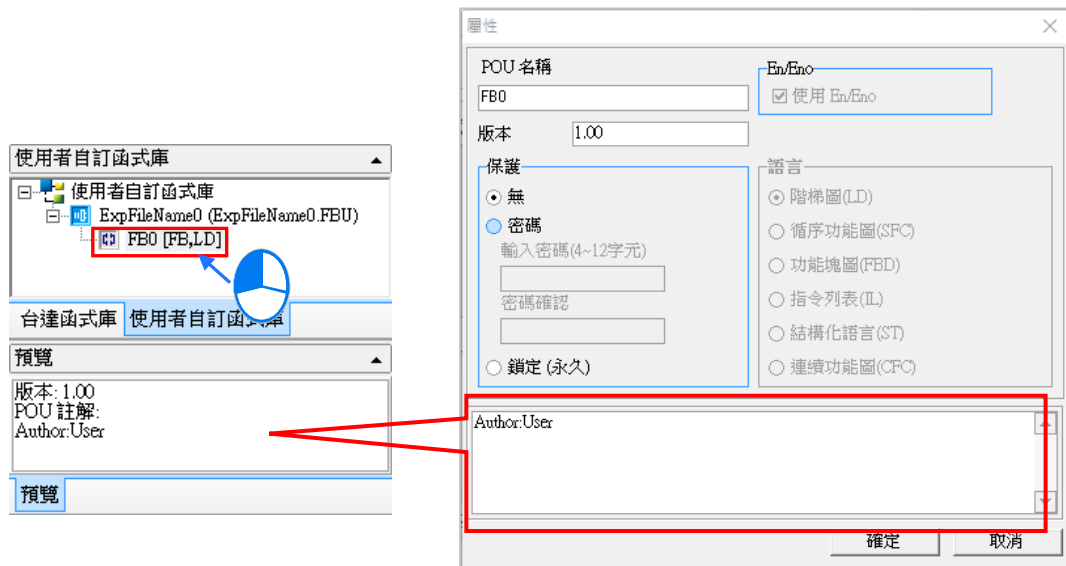
7



在函式庫區點選使用者自訂函式庫分頁，便會在使用者自訂函式庫區中找到剛才匯出的函式。



此時點選此函式，在預覽區將會顯示此函式功能塊屬性設定的註解內容。

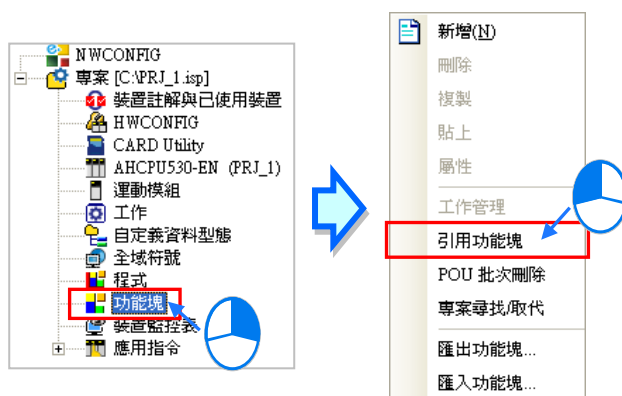


7.5.2 引用使用者自訂函式庫

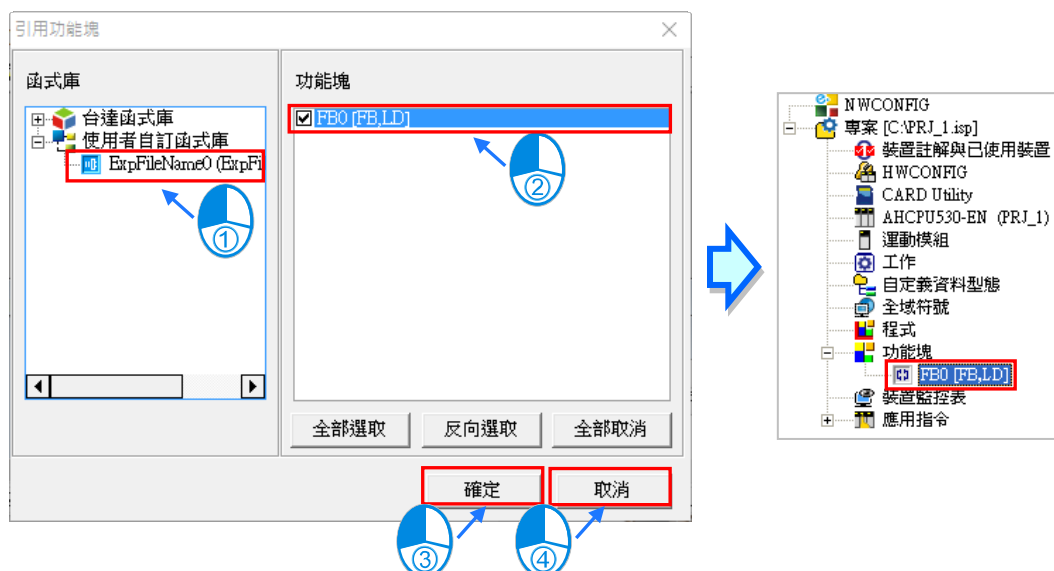
如同前述章節，一旦匯出功能塊到使用者自訂函式庫後，同一台電腦即便開啟新的專案，在使用者自訂函式庫中，也會出現已匯出功能塊的函式，此時便可以下列方式將函式引用到功能塊區中。

- 方法一

點選功能塊項目，按下滑鼠右鍵開啟快捷選單，點選引用功能塊。

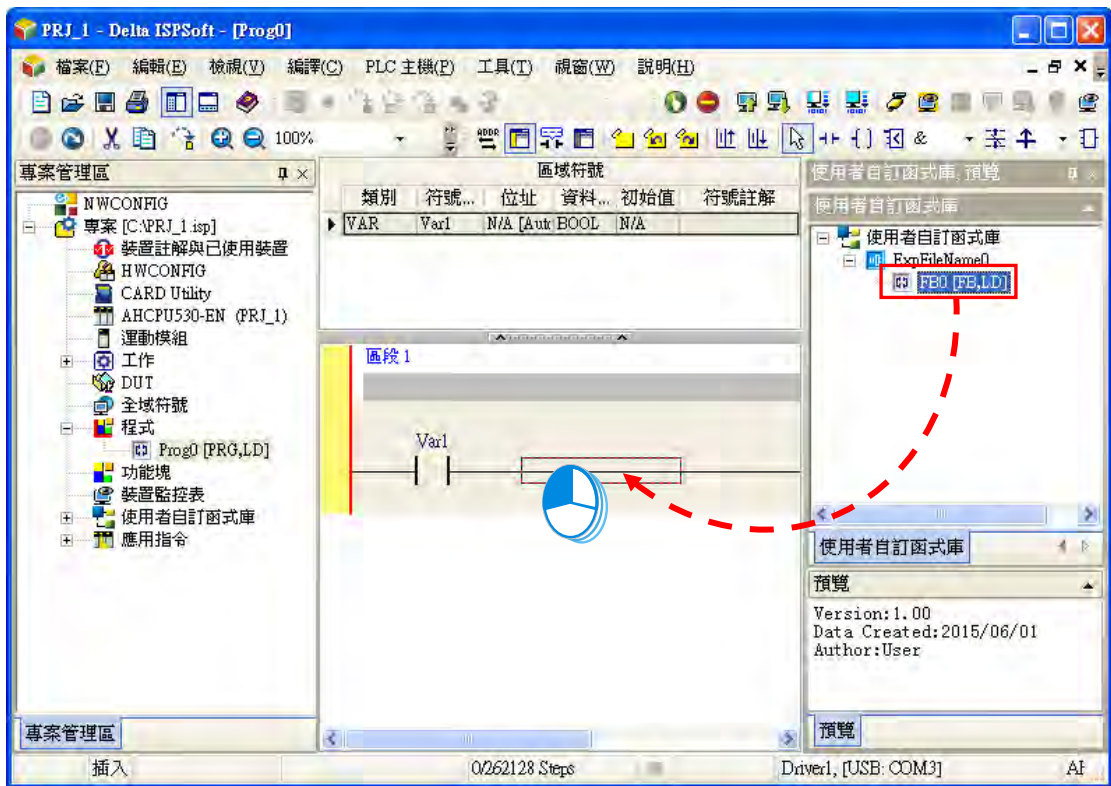


此時會出現引用功能塊視窗，點選函式庫的類別，從中選擇要引用的函式，點選「確定」按鈕。最後完成所有函式的引用後，按下「取消」按鈕，關閉視窗，在專案管理區的功能塊項目下，便會新增引用的函式庫功能塊，功能塊的操作則與一般的功能塊相同。

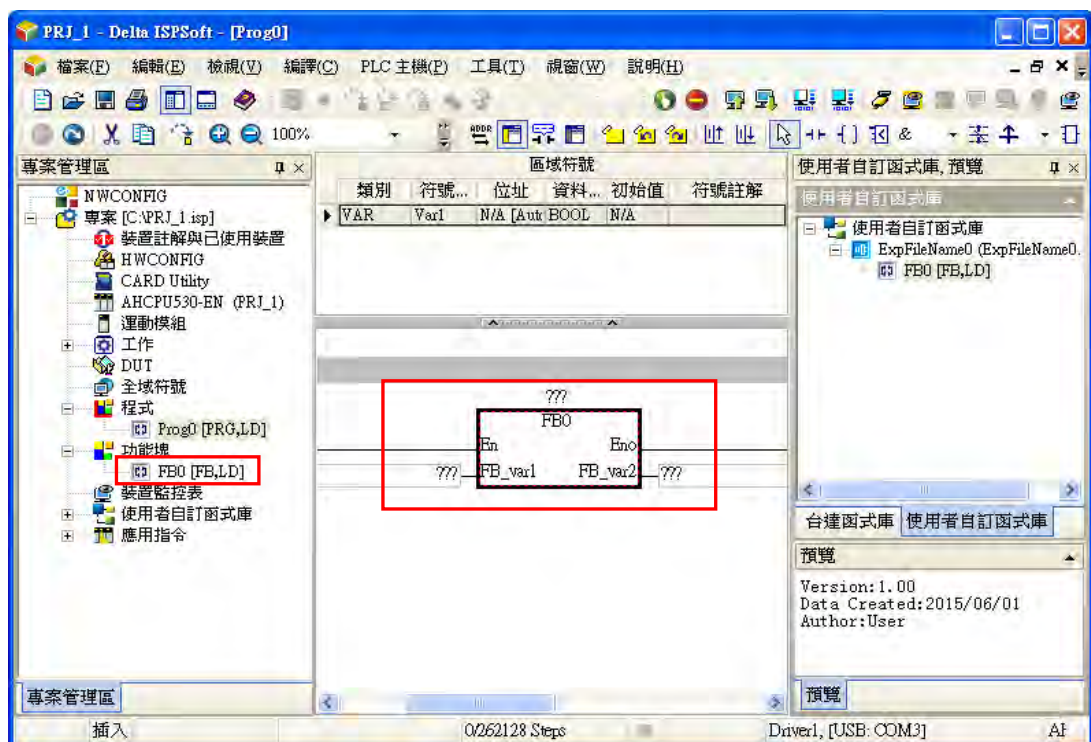


● 方法二

直接在使用者自訂函式庫中點選要使用的函式，按住滑鼠後拖曳到 POU 中要呼叫的位置，便會加入到 POU 中。

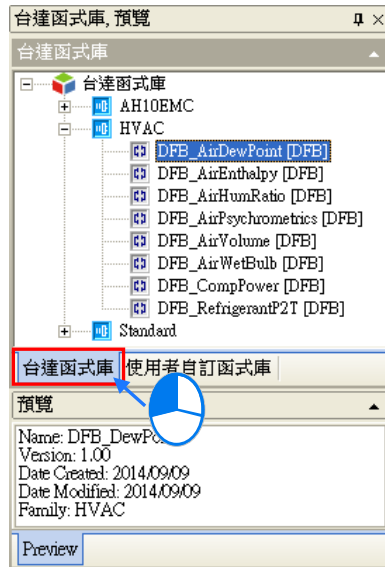


加入到 POU 中後，在專案管理區的功能塊項目下，也會自動新增出此函式的功能塊節點，功能塊的操作則與一般的功能塊相同。



7.5.3 引用台達函式庫

對於各種產業的應用場合，台達提供了一系列的函式庫供使用者引用，點選函式庫區的**台達函式庫**分頁，便會分類列出目前版本所支援的台達函式庫。

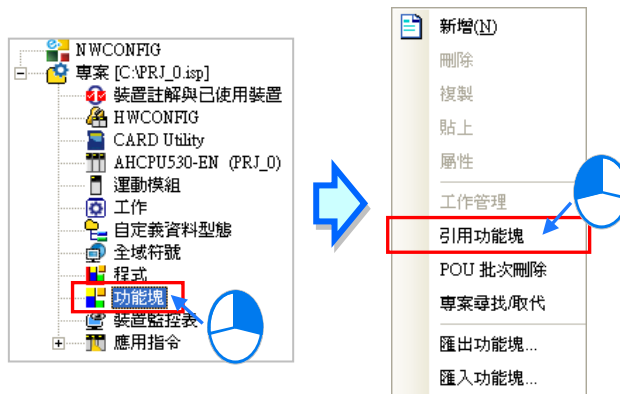


同樣的，對於台達函式庫有以下兩種方法引用到功能塊區中。

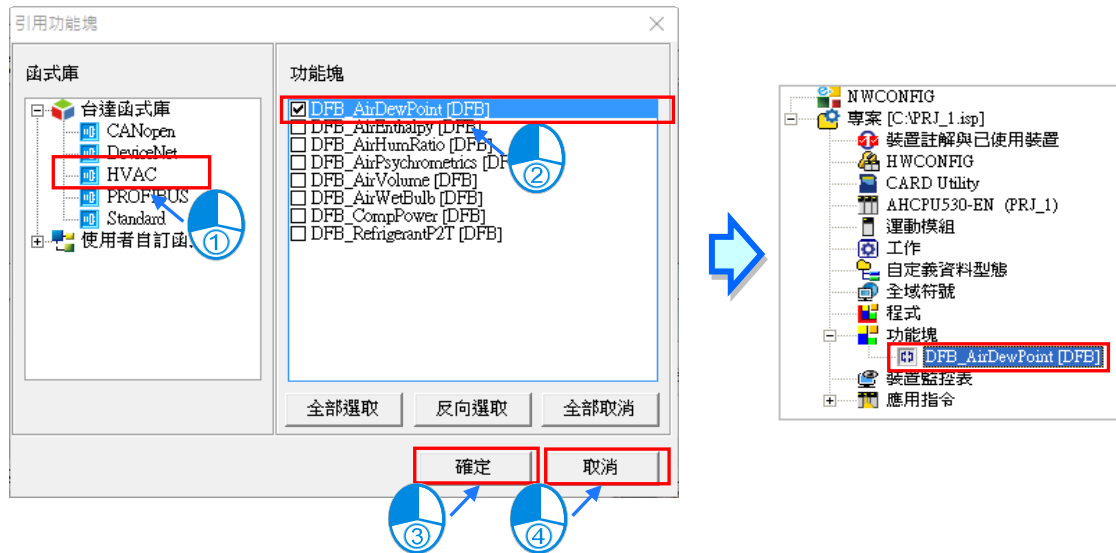
- 方法一

點選功能塊項目，按下滑鼠右鍵開啟快捷選單，點選「引用功能塊」。

7

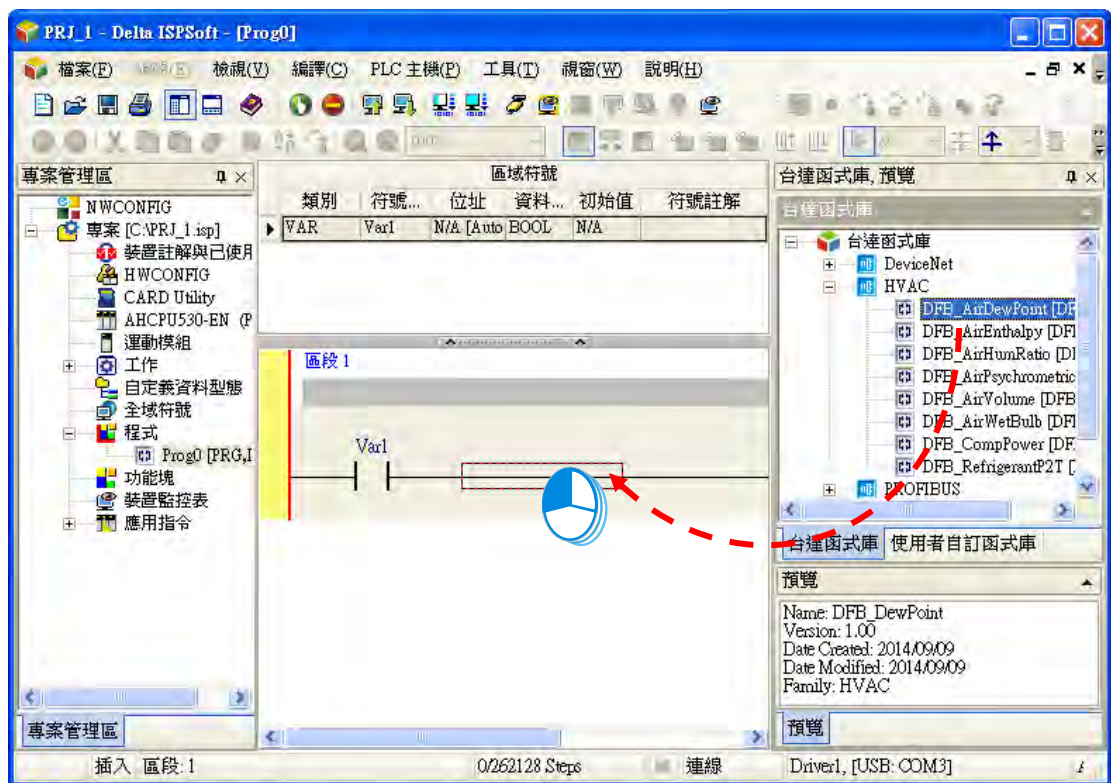


此時會出現引用功能塊視窗，點選函式庫的類別，從中選擇要引用的函式，點選「確定」按鈕。最後完成所有函式的引用後，按下「取消」按鈕，關閉視窗，在專案管理區的功能塊項目下，便會新增引用的函式庫功能塊，功能塊的操作則與一般的功能塊相同。

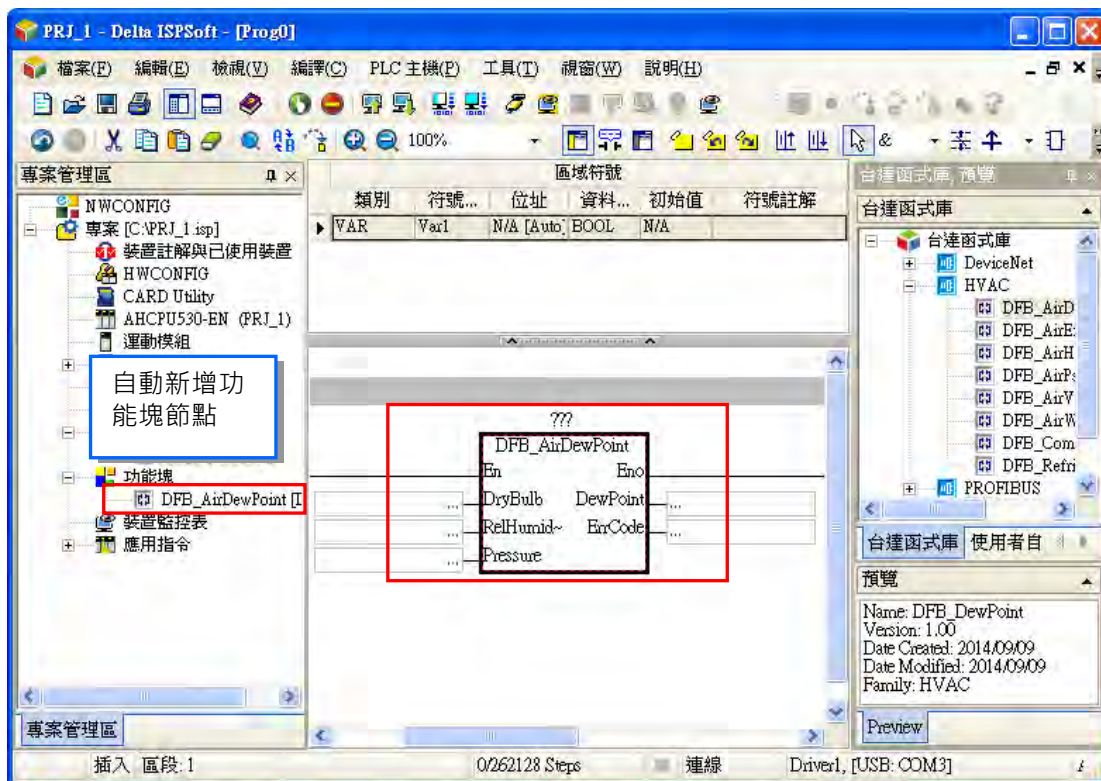


● 方法二

直接在台達函式庫中點選要使用的函式，按住滑鼠後拖曳到 POU 中要呼叫的位置，便會加入到 POU 中。



加入到 POU 中後，在專案管理區的功能塊項目下，也會自動新增出此函式的功能塊節點，功能塊的操作則與一般的功能塊相同。



*1. 台達函式庫的功能塊定義的程式內容，使用者無法開啟檢視或編輯。

8

第8章 自定義資料型態

目錄

8.1 自定義資料型態	8-2
8.2 結構 (STRUCTURE)	8-2
8.2.1 結構 (STRUCTURE) 的意義	8-2
8.2.2 建立結構 (STRUCTURE)	8-3
8.2.3 使用結構 (STRUCTURE) 型態變數符號	8-5
8.2.4 結構 (STRUCTURE) 的常見用法	8-7
8.2.5 結構 (STRUCTURE) 改變為列舉 (ENUMERATION)	8-10
8.3 列舉 (ENUMERATION)	8-11
8.3.1 列舉 (ENUMERATION) 的意義	8-11
8.3.2 建立列舉 (ENUMERATION)	8-11
8.3.3 使用列舉 (ENUMERATION) 型態變數符號	8-13
8.3.4 列舉 (ENUMERATION) 的常見用法	8-15
8.3.5 列舉 (ENUMERATION) 改變為結構 (STRUCTURE)	8-16
8.4 聯合 (UNION)	8-17
8.4.1 聯合 (UNION) 的意義	8-17
8.4.2 建立聯合 (UNION)	8-17
8.4.3 使用聯合 (UNION) 型態變數符號	8-20
8.4.4 聯合 (UNION) 的常見用法	8-21
8.4.5 內建聯合的支援與名稱	8-23

8.1 自定義資料型態

自定義資料型態 (又稱 DUT (Data Type Unit) 或 UDT (User-defined Data Type)) · 為供使用者使用現有基本資料型態組合而成的資料型態。在 PLC 程式開發過程中，有時基本的資料型態仍不能滿足使用者程式中變數符號的使用需要。在 IEC 61131-3 中便導入了一般高階語言中的自定義資料型態觀念，ISPSOft 支援下列三種自定義資料型態，同時也提高了程式的可讀性及開發效率。

類型	說明
結構	結構由多個成員構成，這些成員可以有不同的資料型態。
列舉	列舉型態由多個列舉值構成，每一個列舉值都對應一個 INT 型態的常數。
聯合	聯合與結構相似，不同之處為聯合的所有成員共用一塊記憶體，改變一個成員的值後，所有成員的值都會改變。

8.2 結構 (STRUCTURE)

8.2.1 結構 (STRUCTURE) 的意義

結構 (STRUCTURE) 是由多個各種資料型態的成員組成，如下圖，是由四個成員組成的結構所佔用的記憶體空間大小共為 7 個 WORD。而結構可以做為另一結構的成員。結構可用於定義分組資料以及作為一個資料單元來傳送數值。

STRUCTURE		數值內容																																
成員名稱	資料型態	WORD 1																WORD 0																
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Str1	ARRAY [0..31] OF BOOL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	1	1
Str2	WORD																	12,039 (16#2F07)																
Str3	ARRAY [0..1] OF WORD	32,769(16#8001)																12,039 (16#2F07)																
Str4	DWORD	2,147,561,223(16#80012F07)																																



8.2.2 建立結構 (STRUCTURE)

於專案管理區的「自定義資料型態」項目點擊滑鼠右鍵，點選「新增」，出現新增自定義資料型態視窗，輸入「名稱」，選擇類型為「結構」後，按下「確定」按鈕，即會於自定義資料型態項目下新增此項目。



*.結構的名稱命名限制請參閱下方結構名稱限制。

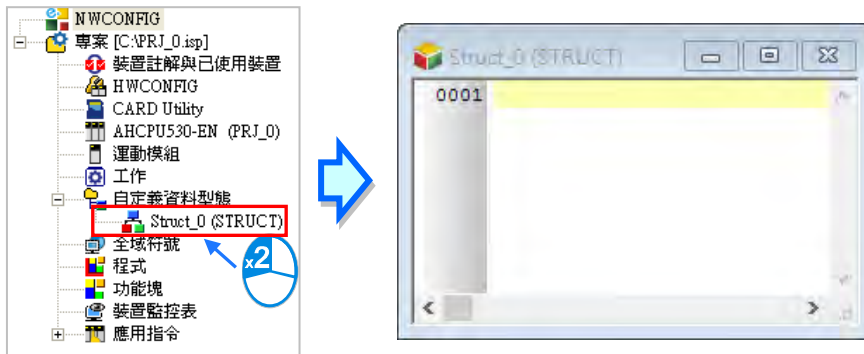
一個結構項目，只能宣告出一個結構 (STRUCTURE)，新增完成後會隨即在編輯工作區開啟結構編輯視窗，並自動帶出註釋與模板供使用者參考修改。

```

Structure_0 (STRUCT)
0001 {
0002 ** Example1:
0003 **
0004 ** TYPE DUT_StructA :
0005 ** STRUCT
0006 **   Element1 : BOOL;
0007 **   Element2 : REAL;
0008 **
0009 **   (Array notation: ARRAY [lowerbound_index..upperbound_index] OF DataType)
0010 **   (Hint: the max dimension of ARRAY is 3)
0011 **   Element3 : ARRAY [0..1][0..3][0..5] OF INT;
0012 ** END_STRUCT
0013 ** END_TYPE
0014 **
0015 ** Example2:
0016 **
0017 ** TYPE DUT_StructB :
0018 ** STRUCT
0019 **   Element1 : DUT_EnumA; (Hint: using Enum in Struct declaration)
0020 **   Element2 : ARRAY [0..7] OF DUT_StructA;
0021 ** END_STRUCT
0022 ** END_TYPE
0023 **
0024 **
0025 TYPE Structure_0 :
0026 STRUCT
0027   Element1 : BOOL;
0028 END_STRUCT
0029 END_TYPE

```

之後也可於專案管理區的此結構項目上雙擊滑鼠左鍵再次開啟編輯視窗。



結構的格式如下，藍色字體為系統關鍵字，黑色字體為使用者自定義的名稱與成員名稱。

TYPE <自定義名稱>:

STRUCT

<成員 1 名稱>: <資料型態>;

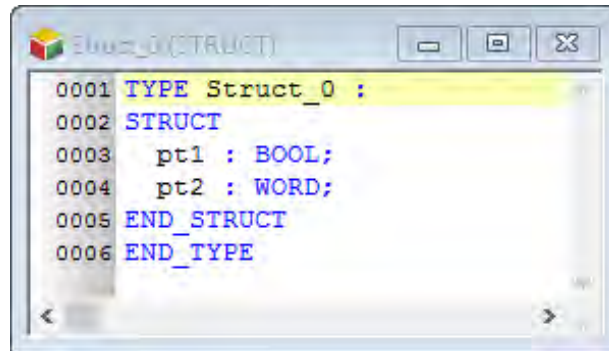
<成員 2 名稱>: <資料型態>;

...

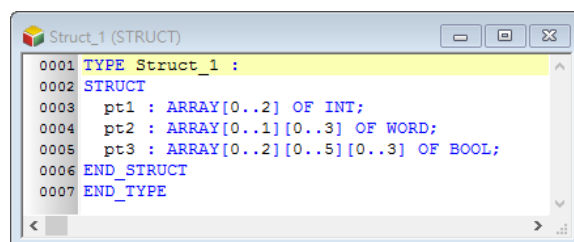
<成員 n 名稱>: <資料型態>;

END_STRUCT

END_TYPE



- **自定義名稱**：預設帶入的自定義名稱，不能修改為其他名稱否則編譯會報錯。
- **成員名稱**：定義成員的名稱。
- **資料型態**：定義成員的資料型態，可為 **BOOL**、**WORD**、**DWORD**、**LWORD**、**INT**、**DINT**、**LINT**、**REAL**、**LREAL**、**ARRAY** 或另一 **STRUCTURE** 或 **ENUMERATION**。若選擇 **DVPxxMC** 或 **AS5xx** 機種，則可支援更多資料類型如 **BYTE**、**SINT**、**USINT**、**UINT**、**UDINT**、**ULINT**、**TIME**、**DATE**、**TOD**、**DT**、**STRING**。當成員資料型態為 **ARRAY** 時格式為 **ARRAY [0..n] OF <資料型態>;**，n 最大為 65535，若選擇 **DVPxxMC** 或 **AS5xx** 機種，則最大為 2147483647。若成員資料型態為 **STRUCTURE** 時，直接寫入另一 **STRUCTURE** 的名稱。**ARRAY** 中的資料型態無法再輸入 **ARRAY**，若要定義多維陣列，則最多支援三維陣列，三維陣列格式為 **ARRAY [0..n] [0..n] [0..n] OF <資料型態>;**，二維陣列格式為 **ARRAY [0..n] [0..n] OF <資料型態>;**，如下圖。



自定義名稱與成員名稱使用時有其一定的規則與限制，說明如下。

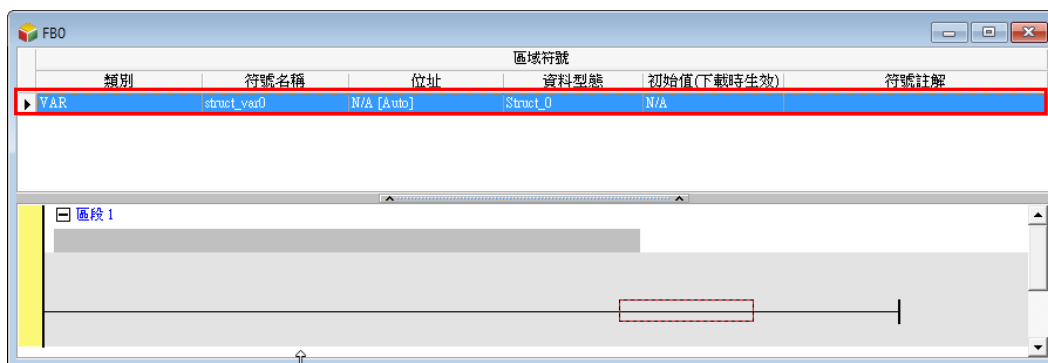
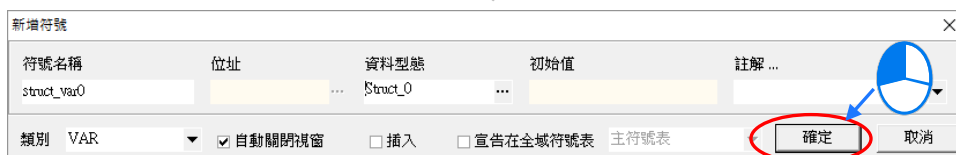
- 自定義名稱不可與其他自定義資料型態名稱重複。
- 同一個自定義中不允許重複的成員名稱。
- 自定義名稱與成員名稱不可超過 30 個字元，且須注意一個中文字會佔用兩個字元。
- 不可使用系統保留的關鍵字，如指令名稱、裝置名稱，或是其它在系統中已被賦予特殊意義的保留字，但若只是被包含在名稱當中則是合法的，如 “M0” 是不合法的名稱，但 “_M0” 則是合法的。
- 自定義名稱與成員名稱當中不可有空白，例如 “INPUT CH0” 便是不合法的名稱。
- 自定義名稱與成員名稱當中可使用底線，但不可連續使用，或是置於結尾，例如 ”INPUT_CH0” 是合法的名稱，但 “INPUT__CH0” 與 “INPUT_CH0_” 則是不合法的名稱。
- 自定義名稱與成員名稱不可使用特殊字元，例如 *、#、?、\、%、@ ...等。

8.2.3 使用結構 (STRUCTURE) 型態變數符號

完成結構型態的建立，再執行**檢查**或**編譯**後，在符號表中便可宣告此結構型態的變數，於全域或區域符號表開啟新增符號視窗，輸入**符號名稱**後點選資料型態右側... 按鈕，如下圖。

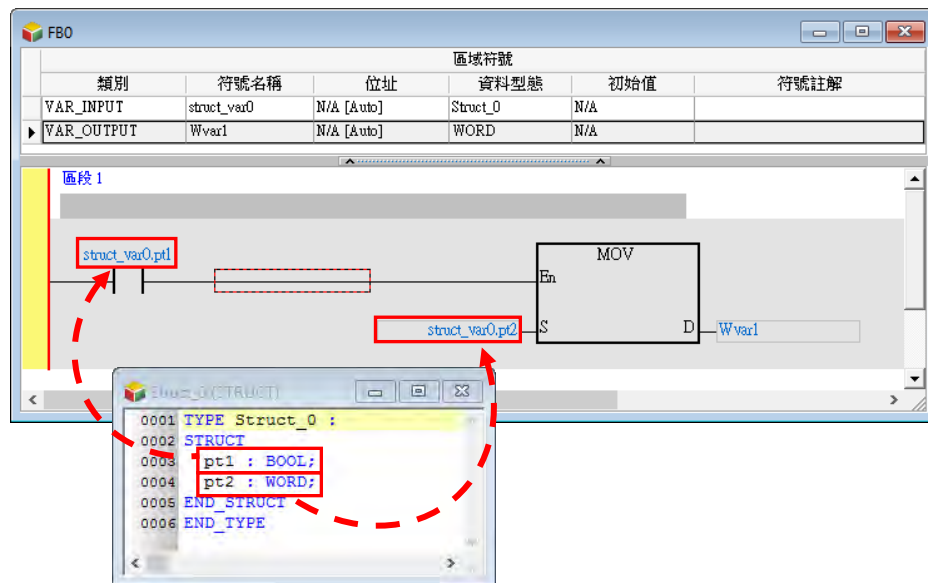


於資料型態視窗點選「使用者自訂型態」，在於下方欄位點選已宣告的結構資料型態。按下「確定」後完成新建符號。

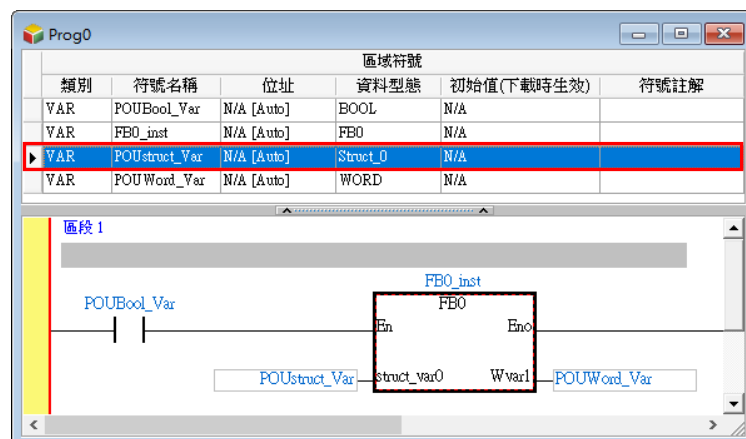


8.2.4 結構 (STRUCTURE) 的常見用法

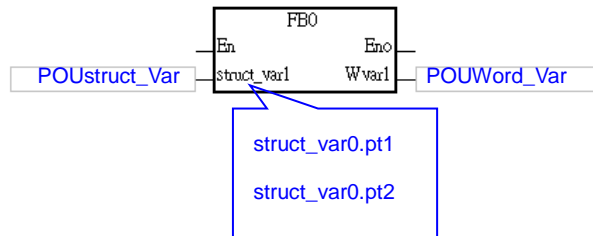
在完成宣告結構與變數符號後，便可於 POU 中使用。如下圖，功能塊中變數符號 **struct_var0** 的資料型態為 **Struct_0**。在程式中使用此結構變數符號的成員，可用”.”的方式，例如下圖中 **struct_var0.pt1** 為使用此結構變數符號的第一個成員 **pt1** 作為此接點的 **BOOL** 資料型態運算元；而 **struct_var0.pt2** 為使用此結構變數符號的第二個成員 **pt2** 作為此指令接腳的 **WORD** 資料型態運算元。



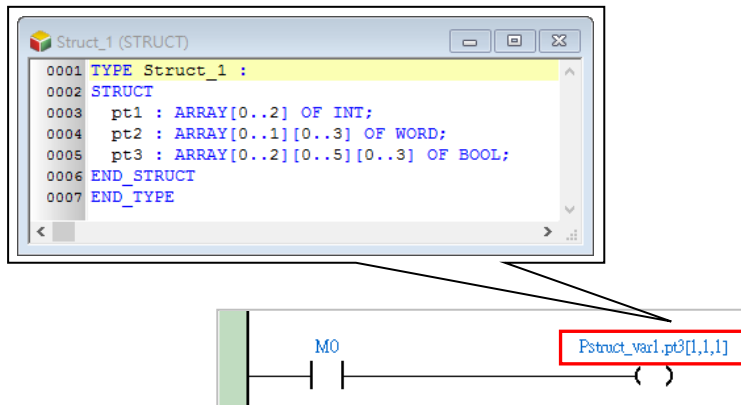
若在 POU 中使用上述功能塊，為了對應功能塊中 **Struct_0** 資料型態的輸入與輸出接腳，也必須在此 POU 中宣告 **Struct_0** 資料型態的變數符號作為運算元，如下圖 **POUstruct_var**，同樣的結構的成員也可用”.”的方式使用。



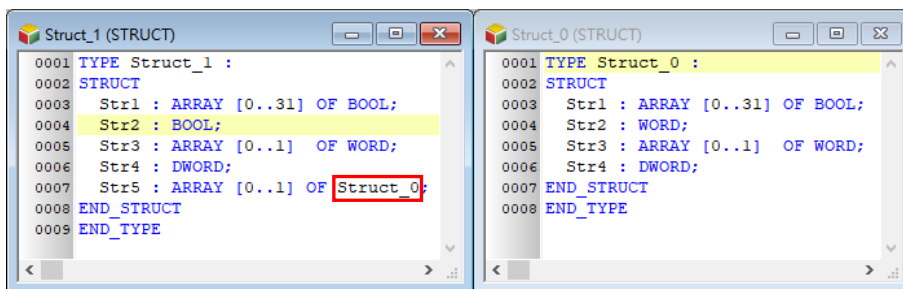
由上述說明可知，當結構變數符號 **POUstruct_Var** 輸入到 **FBO** 時，便同時傳遞了此變數符號內所有的成員，供 **FBO** 中的 **struct_var0.pt1** 與 **struct_var0.pt2** 運算。因此使用者在開發功能塊時，可將結構變數定義為功能塊輸入或輸出接腳，便可同時傳遞此變數內的所有成員，藉此節省了一一定義接腳的動作，同時也讓程式更為簡潔。



當結構的內容包含陣列成員時，使用此類結構的變數符號可以下形式書寫，如下圖中將 **M0** 的狀態寫入自定義資料型態變數符號 **Pstruct_var1** 中指定成員 **pt3** 中三維陣列的第[1,1,1]個。



另一個例子為結構可以當成另一個結構的成員，如下圖，結構 **Struct_1** 內包含 **Struct_0** 的成員。



8

其中 Struct_1 的所佔用的記憶體大小如下表所示

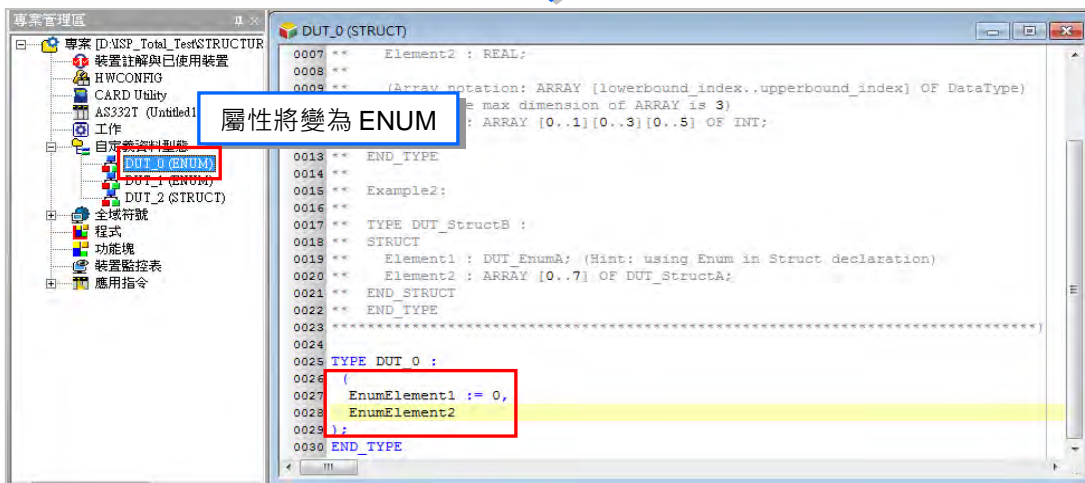
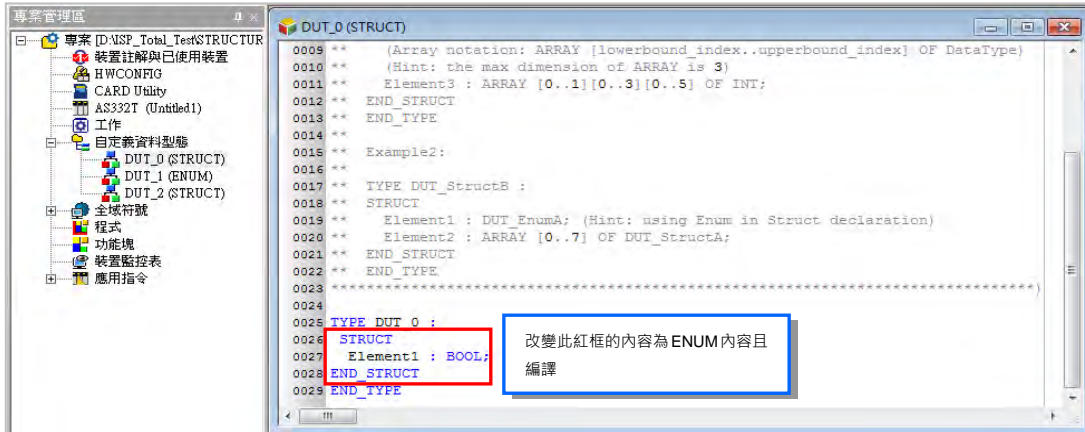
自定義名稱	成員名稱	資料型態	佔用記憶體(Bit)
Struct_1	Str1	ARRAY [0..31] OF BOOL	
	Str2	BOOL	
	Str3	ARRAY [0..1] OF WORD	
	Str4	DWORD	
	Str5	ARRAY [0..1] OF Struct_0	

而上圖紅色框框內為一個 Struct_0 成員所佔的記憶體大小為 7 個 WORD，如下圖。

		Struct_0																															
成員名稱	資料型態	數值內容																															
		WORD 1														WORD 0																	
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Str1	ARRAY [0..31] OF BOOL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	1	1
Str2	WORD															12,039 (16#2F07)																	
Str3	ARRAY [0..1] OF WORD	32,769(16#8001)														12,039 (16#2F07)																	
Str4	DWORD	2,147,561,223(16#80012F07)																															

8.2.5 結構 (STRUCTURE) 改變為列舉 (ENUMERATION)

當使用者想將自定義資料型態「結構」改成「列舉」，可以將 STRUCT 與 END_STRUCT 改成“(”以及“);” ，此時再將成員內容改成 ENUM 型態，在執行編譯，則自定義資料型態的屬性將由 (STRUCT) 變為 (ENUM)。ENUM 的設定內容請參考第 8.3 節。



8.3 列舉 (ENUMERATION)

8.3.1 列舉 (ENUMERATION) 的意義

列舉 (ENUMERATION) 用於定義一組字串別名以及對應的數值，便於幫助使用者記憶。

8.3.2 建立列舉 (ENUMERATION)

於專案管理區的「自定義資料型態」項目點擊滑鼠右鍵，點選「新增」，出現新增自定義資料型態視窗，輸入「名稱」，選擇類型為「列舉」後，按下「確定」按鈕，即會於自定義資料型態項目下新增此項目。



*.列舉的名稱命名限制請參閱下方列舉名稱限制。

一個列舉項目，只能宣告一個列舉 (ENUMERATION) 新增完成後會隨即在編輯工作區開啟列舉編輯視窗，並自動帶出註釋與模板供使用者參考修改。

```

Enum_0 (ENUM)
0001 |*****
0002 |** Example1:
0003 |**
0004 |** TYPE DUT_EnumA :
0005 |** (
0006 |**   Red := 0,
0007 |**   Blue := 2,
0008 |**   Green := 3
0009 |** );
0010 |** END_TYPE
0011 |**
0012 |** Example2:
0013 |**
0014 |** TYPE DUT_EnumB :
0015 |** (
0016 |**   Red := 0, (Hint: allowable to set or not to set the Enum value)
0017 |**   Blue,
0018 |**   Green
0019 |** );
0020 |** END_TYPE
0021 |*****
0022 |
0023 |TYPE Enum_0 :
0024 | (
0025 |   EnumElement1 := 0,
0026 |   EnumElement2
0027 | );
0028 |END_TYPE

```

之後也可於專案管理區的此列舉項目上雙擊滑鼠左鍵再次開啟編輯視窗。

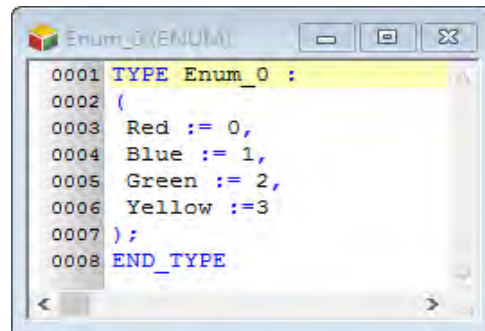


列舉的格式如下，藍色字體為系統關鍵字，黑色字體為使用者自定義名稱、成員名稱及數值。

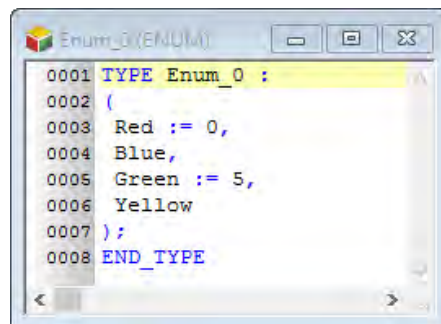
TYPE <自定義名稱>:

```
(
  <成員 1 名稱> := <數值 1>,
  <成員 2 名稱> := <數值 2>,
  ...
  <成員 n 名稱> := <數值 n>
);
```

END_TYPE



- **自定義名稱**：預設帶入的自定義名稱，不能修改為其他名稱否則編譯會報錯。
- **成員名稱**：定義成員的名稱。
- **數值**：定義每個成員的數值，範圍可為-32768~32767，也可空白不輸入，系統將會根據前一成員的值依序自行帶入預設值，且避開已使用的值。如下圖，系統將會自動設定成員 Blue 的值为 1，而 Yellow 的值为 6。



自定義名稱與成員名稱使用時有其一定的規則與限制，說明如下。

- 自定義名稱不可與其他自定義資料型態名稱重複。
- 同一個自定義中不允許重複的成員名稱。
- 自定義名稱與成員名稱不可超過 30 個字元，且須注意一個中文字會佔用兩個字元。
- 不可使用系統保留的關鍵字，如指令名稱、裝置名稱，或是其它在系統中已被賦予特殊意義的保留字，但若只是被包含在名稱當中則是合法的，如 “M0” 是不合法的名稱，但 “_M0” 則是合法的。
- 自定義名稱與成員名稱當中不可有空白，例如 “INPUT CH0” 便是不合法的名稱。
- 自定義名稱與成員名稱當中可使用底線，但不可連續使用，或是置於結尾，例如 ”INPUT_CH0” 是合法的名稱，但 “INPUT__CH0” 與 “INPUT_CH0_” 則是不合法的名稱。
- 自定義名稱與成員名稱不可使用特殊字元，例如 *、#、?、\、%、@ ... 等。
- 列舉數值為 INT 資料型態。

8.3.3 使用列舉 (ENUMERATION) 型態變數符號

完成列舉型態的建立，再執行**檢查**或**編譯**後，在符號表中便可宣告此列舉型態的變數，如下圖於全域或區域符號表開啟新建符號視窗，輸入**符號名稱**後點選資料型態右側... 按鈕。



於資料型態視窗點選**使用者自訂型態**，再於下方欄位點選已宣告的列舉資料型態。按下「**確定**」後完成新建符號。

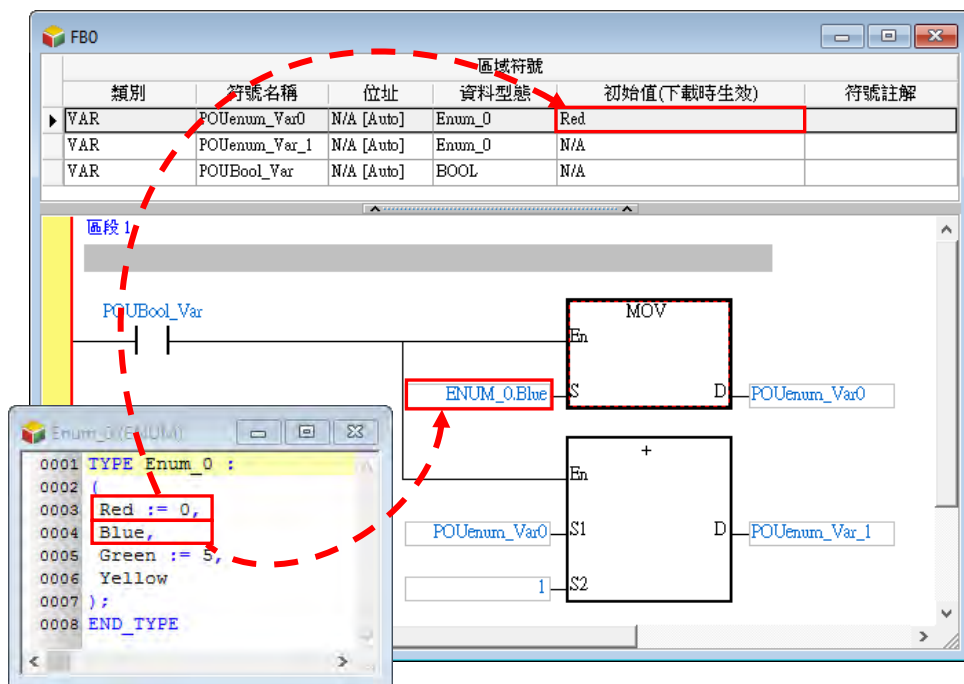


8

8.3.4 列舉 (ENUMERATION) 的常見用法

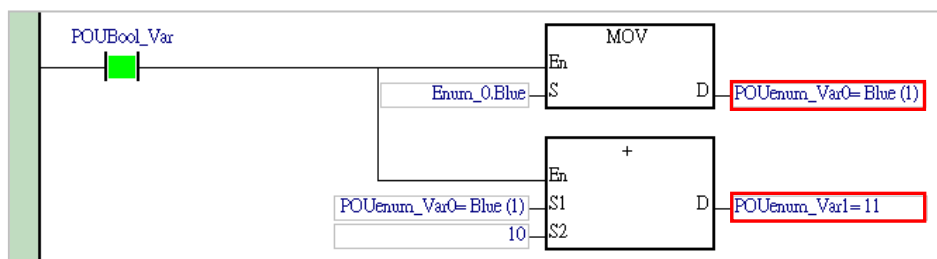
在完成宣告列舉與變數符號後，便可於 POU 中使用。如下圖，變數符號 **POUenum_Var0** 與 **POUenum_Var1** 的資料型態為 Enum_0，在符號表中可以成員名稱或數值定義初始值，如下圖的 **Red**。

程式中可用“.”的方式指定已定義的列舉中某一成員，如下圖的 **Enum_0.Blue**，並將其指定到一個 Enum_0 型態變數 **POUenum_Var0** 中(以 MOV 指令)，相當於將定義的成員數值寫入列舉變數中，因此 MOV 指令的結果便會使 **POUenum_Var0** 的值为 1。



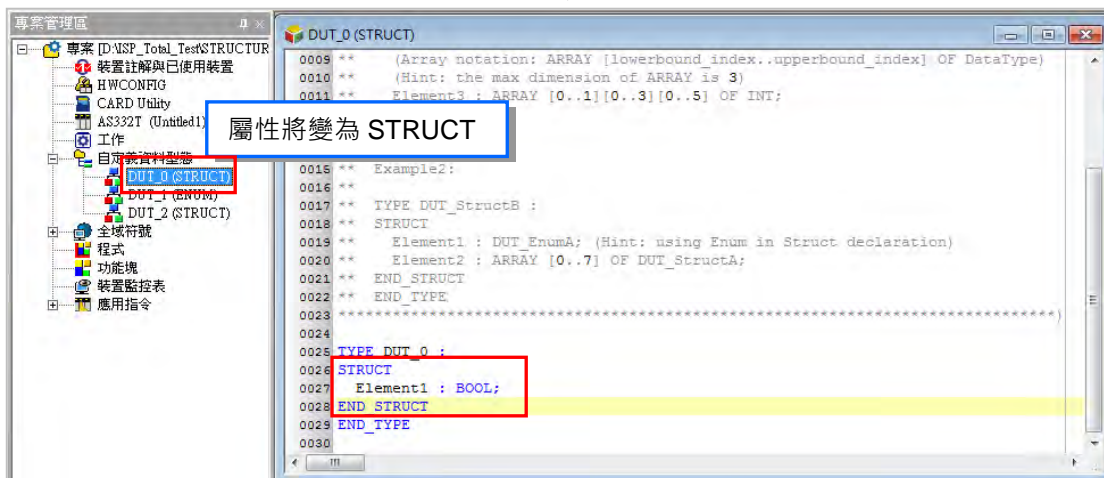
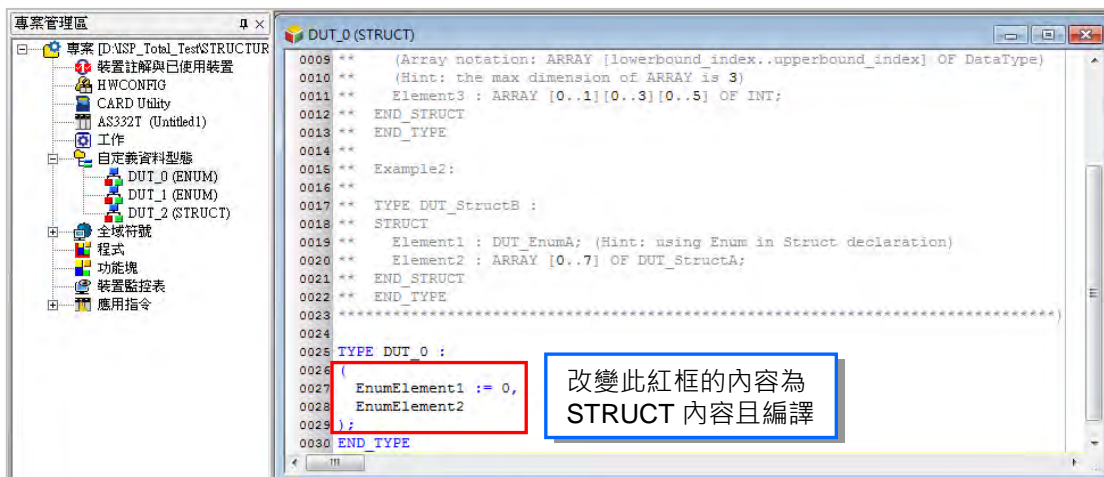
另外也可以對列舉變數符號進行數值運算，如上圖以+指令對 **POUenum_Var0** 的數值進行+1，並將結果寫入到 **POUenum_Var1**，因為 **POUenum_Var0** 的數值為 1，故 **POUenum_Var1** 的執行結果將為 2，即為定義的成員 **Green**。

因此在程式實際運行並線上監控時，所有的列舉變數符號便會顯示當前數值所定義的成員名稱，如下圖的 **POUenum_Var0=Blue(1)**。需注意的是，若列舉變數符號當前的數值未在定義的成員範圍內，則會直接顯示此數值，如下圖的 **POUenum_Var1=11** 因為數值 11 未在列舉中定義。



8.3.5 列舉 (ENUMERATION) 改變為結構 (STRUCTURE)

當使用者想將自定義資料型態「列舉」改成「結構」，可以將“ (”與“); ”改成 **STRUCT** 以及 **END_STRUCT**，此時刻再將成員內容改成 **STRUCT** 型態，再執行編譯，則自定義資料型態的屬性將由 (**ENUM**) 變為 (**STRUCT**)，**STRUCT** 的設定內容請參考第 8.2 節。



8.4 聯合 (UNION)

8.4.1 聯合 (UNION) 的意義

聯合與結構相似，不同之處為聯合的所有成員共用一塊記憶體且一個自定義聯合的最大值，取決於聯合中的最大成員，如下圖聯合是由四個成員組成其所占用的記憶體為 32 個 Bit (2 個 Word)，改變一個成員的數值後，所有成員的數值都會跟著改變。

		UNION																															
成員名稱	資料型態	數值內容																															
		WORD 1																WORD 0															
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
uni1	ARRAY [0..31] OF BOOL	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	0	0	0	1	1	1
uni2	WORD																	12,039 (16#2F07)															
uni3	ARRAY [0..1] OF WORD	32,769(16#8001)																12,039 (16#2F07)															
uni4	DWORD	2,147,561,223(16#80012F07)																															

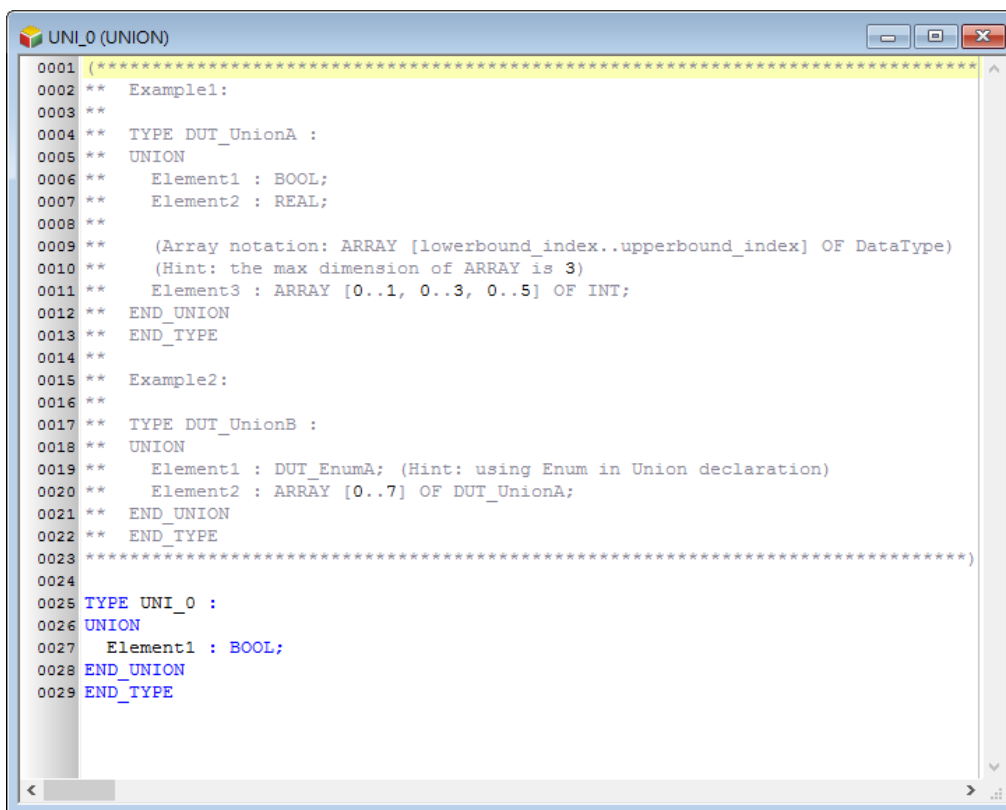
8.4.2 建立聯合 (UNION)

於專案管理區的「自定義資料型態」項目點擊滑鼠右鍵，點選「新增」，出現新增自定義資料型態視窗，輸入「名稱」，選擇類型為「聯合」後，按下「確定」按鈕，即會於自定義資料型態項目下新增此項目。



*.聯合的名稱命名限制請參閱下方聯合名稱限制。

一個聯合項目，只能宣告出一個聯合 (UNION)，新增完成後會隨即在編輯工作區開啟聯合編輯視窗，並自動帶出註釋與模板供使用者參考修改。



```
0001 (*****  
0002 ** Example1:  
0003 **  
0004 ** TYPE DUT_UnionA :  
0005 ** UNION  
0006 **   Element1 : BOOL;  
0007 **   Element2 : REAL;  
0008 **  
0009 **   (Array notation: ARRAY [lowerbound_index..upperbound_index] OF DataType)  
0010 **   (Hint: the max dimension of ARRAY is 3)  
0011 **   Element3 : ARRAY [0..1, 0..3, 0..5] OF INT;  
0012 ** END_UNION  
0013 ** END_TYPE  
0014 **  
0015 ** Example2:  
0016 **  
0017 ** TYPE DUT_UnionB :  
0018 ** UNION  
0019 **   Element1 : DUT_EnumA; (Hint: using Enum in Union declaration)  
0020 **   Element2 : ARRAY [0..7] OF DUT_UnionA;  
0021 ** END_UNION  
0022 ** END_TYPE  
0023 *****)  
0024  
0025 TYPE UNI_0 :  
0026 UNION  
0027   Element1 : BOOL;  
0028 END_UNION  
0029 END_TYPE
```

之後也可於專案管理區的此聯合項目上雙擊滑鼠左鍵再次開啟編輯視窗。



8

聯合的格式如下，藍色字體為系統關鍵字，黑色字體為使用者自定義的名稱、成員名稱及數值。

TYPE <自定義名稱>:

UNION

<成員 1 名稱> := <資料型態>;

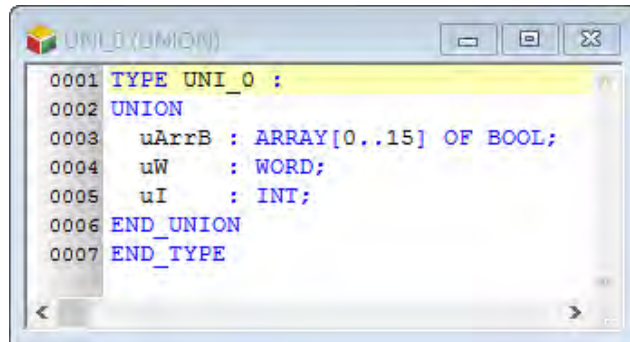
<成員 2 名稱> := <資料型態>;

...

<成員 n 名稱> := <資料型態>;

END_UNION

END_TYPE

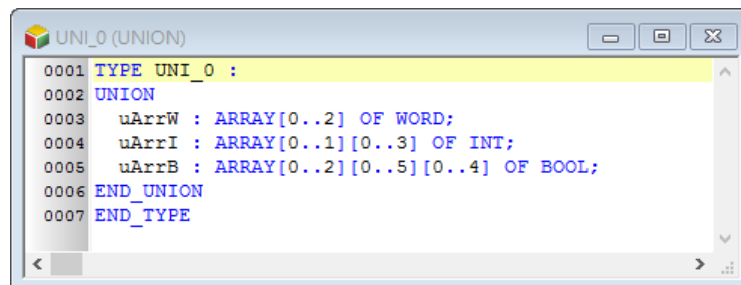


```

0001 TYPE UNI_0 :
0002 UNION
0003   uArrB : ARRAY[0..15] OF BOOL;
0004   uW    : WORD;
0005   uI    : INT;
0006 END_UNION
0007 END_TYPE

```

- **自定義名稱**：預設帶入的自定義名稱，不能修改為其他名稱否則編譯會報錯。
- **成員名稱**：定義成員的名稱。
- **資料型態**：定義成員的資料型態，可為 **BOOL**、**WORD**、**DWORD**、**LWORD**、**INT**、**DINT**、**LINT**、**REAL**、**LREAL**、**ARRAY** 或另一 **STRUCTURE** 或 **ENUMERATION**。若選擇 **DVPxxMC** 或 **AS5xx** 機種，則可支援更多資料類型如 **BYTE**、**SINT**、**USINT**、**UINT**、**UDINT**、**ULINT**、**TIME**、**DATE**、**TOD**、**DT**、**STRING**。當成員資料型態為 **ARRAY** 時格式為 **ARRAY [0..n] OF <資料型態>**；**n** 最大為 65535，若選擇 **DVPxxMC** 或 **AS5xx** 機種，則最大為 2147483647。若成員資料型態為 **STRUCTURE** 時，直接寫入另一 **STRUCTURE** 的名稱。**ARRAY** 中的資料型態無法再輸入 **ARRAY**，若要定義多維陣列，則最多支援三維陣列，三維陣列格式為 **ARRAY [0..n] [0..n] [0..n] OF <資料型態>**；二維陣列格式為 **ARRAY [0..n] [0..n] OF <資料型態>**；如下圖。



```

0001 TYPE UNI_0 :
0002 UNION
0003   uArrW : ARRAY[0..2] OF WORD;
0004   uArrI : ARRAY[0..1][0..3] OF INT;
0005   uArrB : ARRAY[0..2][0..5][0..4] OF BOOL;
0006 END_UNION
0007 END_TYPE

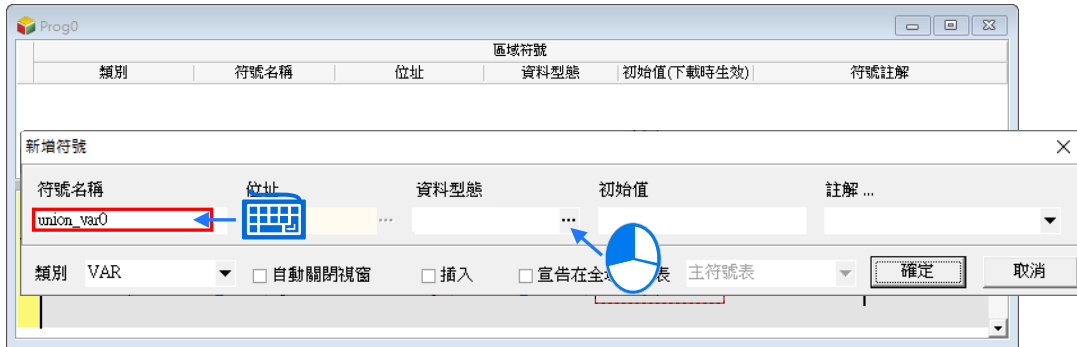
```

自定義名稱與成員名稱使用時有其一定的規則與限制，說明如下。

- 自定義名稱不可與其他自定義資料型態名稱重複。
- 同一個自定義中不允許重複的成員名稱。
- 自定義名稱與成員名稱不可超過 30 個字元，且須注意一個中文字會佔用兩個字元。
- 不可使用系統保留的關鍵字，如指令名稱、裝置名稱，或是其它在系統中已被賦予特殊意義的保留字，但若只是被包含在名稱當中則是合法的，如 **“MO”** 是不合法的名稱，但 **“_MO”** 則是合法的。
- 自定義名稱與成員名稱當中不可有空白，例如 **“INPUT CH0”** 便是不合法的名稱。
- 自定義名稱與成員名稱當中可使用底線，但不可連續使用，或是置於結尾，例如 **“INPUT_CH0”** 是合法的名稱，但 **“INPUT__CH0”** 與 **“INPUT_CH0_”** 則是不合法的名稱。
- 自定義名稱與成員名稱不可使用特殊字元，例如 *****、**#**、**?**、****、**%**、**@** ...等。

8.4.3 使用聯合 (UNION) 型態變數符號

完成聯合型態的建立，再執行**檢查**或**編譯**後，在符號表中便可宣告此聯合型態的變數，於全域或區域符號表開啟新增符號視窗，輸入**符號名稱**後點選資料型態右側... 按鈕，如下圖。

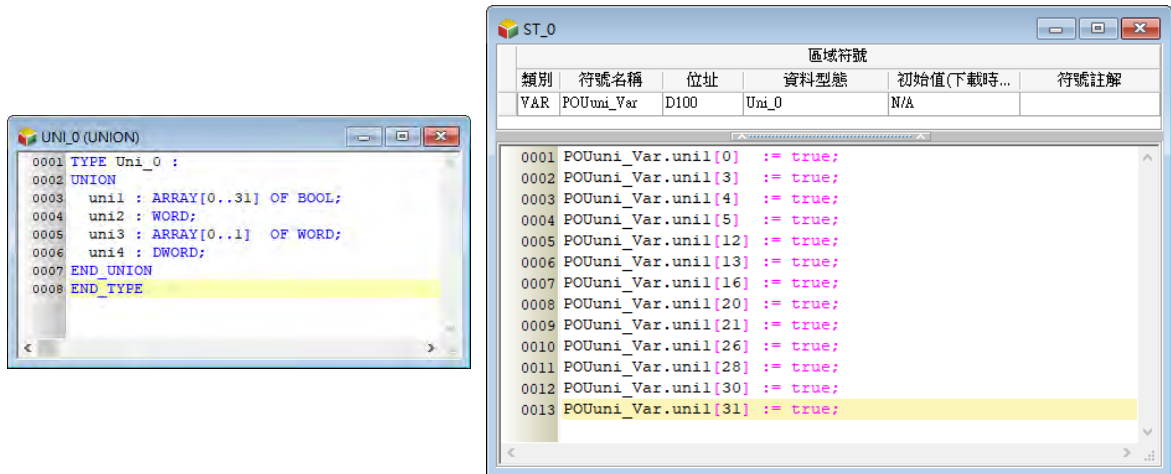


於資料型態視窗點選**使用者自訂型態**，再於下方欄位點選已宣告的列舉資料型態。按下「**確定**」後完成新建符號。



8.4.4 聯合 (UNION) 的常見用法

下列範例建立一個聯合裡面包含 uni1 到 uni4 四個成員，接著將變數符號 **POUni_Var** 資料型態宣告為 Uni_0，然後使用程式改變 **POUni_Var.uni1** 的數值，如下圖。



再透過監控表查看 **POUni_Var.uni2** 與其他成員的數值是否改變，如下圖。

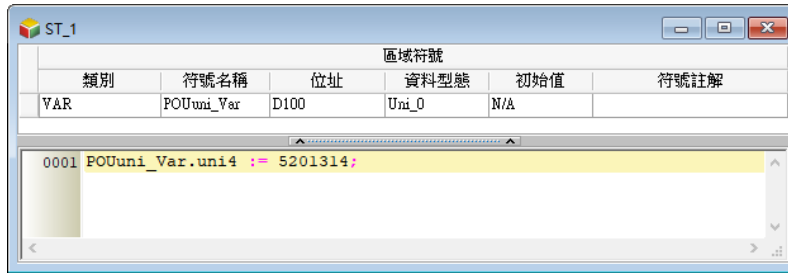
來源	符號名稱	裝置名稱	狀態	資料型態	值(16位元)	值(32位元)	數值型態
ST_0	POUni_Var.uni2	D100		WORD	12345	3559993401	自動偵測
ST_0	POUni_Var.uni3[0]	D100		WORD	12345	3559993401	自動偵測
ST_0	POUni_Var.uni3[1]	D101		WORD	54321	54321	自動偵測
ST_0	POUni_Var.uni4	D100		DWORD	12345	3559993401	自動偵測

由下表數值可以看出成員 uni1 到 uni4 所對應的數值。

		POUni_Var																																
成員名稱	資料型態	數值內容																																
		WORD 1																WORD 0																
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
uni1	ARRAY [0..31] OF BOOL	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1
uni2	WORD																	12,345																
uni3	ARRAY [0..1] OF WORD	54,321																12,345																
uni4	DWORD	3,559,993,401																																



第二個範例改變 **POUuni_Var.uni4** 的數值，如下圖。



再從監控表觀察其他成員的數值，所以改變一個成員的數值內容後，所有成員的數值都會隨之改變。

來源	符號名稱	裝置名稱	狀態	資料型態	值(16位元)	值(32位元)	數值型態
ST_1	POUuni_Var.uni1[0]	D100.0		BOOL			
ST_1	POUuni_Var.uni1[1]	D100.1		BOOL			
ST_1	POUuni_Var.uni1[2]	D100.2		BOOL			
ST_1	POUuni_Var.uni1[3]	D100.3		BOOL			
ST_1	POUuni_Var.uni1[4]	D100.4		BOOL			
ST_1	POUuni_Var.uni1[5]	D100.5		BOOL			
ST_1	POUuni_Var.uni1[6]	D100.6		BOOL			
ST_1	POUuni_Var.uni1[7]	D100.7		BOOL			
ST_1	POUuni_Var.uni1[8]	D100.8		BOOL			
ST_1	POUuni_Var.uni1[9]	D100.9		BOOL			
ST_1	POUuni_Var.uni1[10]	D100.10		BOOL			
ST_1	POUuni_Var.uni1[11]	D100.11		BOOL			
ST_1	POUuni_Var.uni1[12]	D100.12		BOOL			
ST_1	POUuni_Var.uni1[13]	D100.13		BOOL			
ST_1	POUuni_Var.uni1[14]	D100.14		BOOL			
ST_1	POUuni_Var.uni1[15]	D100.15		BOOL			
ST_1	POUuni_Var.uni1[16]	D101.0		BOOL			
ST_1	POUuni_Var.uni1[17]	D101.1		BOOL			
ST_1	POUuni_Var.uni1[18]	D101.2		BOOL			
ST_1	POUuni_Var.uni1[19]	D101.3		BOOL			
ST_1	POUuni_Var.uni1[20]	D101.4		BOOL			
ST_1	POUuni_Var.uni1[21]	D101.5		BOOL			
ST_1	POUuni_Var.uni1[22]	D101.6		BOOL			
ST_1	POUuni_Var.uni1[23]	D101.7		BOOL			
ST_1	POUuni_Var.uni1[24]	D101.8		BOOL			
ST_1	POUuni_Var.uni1[25]	D101.9		BOOL			
ST_1	POUuni_Var.uni1[26]	D101.10		BOOL			
ST_1	POUuni_Var.uni1[27]	D101.11		BOOL			
ST_1	POUuni_Var.uni1[28]	D101.12		BOOL			
ST_1	POUuni_Var.uni1[29]	D101.13		BOOL			
ST_1	POUuni_Var.uni1[30]	D101.14		BOOL			
ST_1	POUuni_Var.uni1[31]	D101.15		BOOL			
ST_1	POUuni_Var.uni2	D100		WORD	23970	5201314	有號數十進制
ST_1	POUuni_Var.uni3[0]	D100		WORD	23970	5201314	有號數十進制
ST_1	POUuni_Var.uni3[1]	D101		WORD	79	79	有號數十進制

8

由下表數值可以看出成員 uni1 到 uni4 所對應的數值。

POUni_Var																																	
成員名稱	資料型態	數值內容																															
		WORD 1													WORD 0																		
		Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
uni1	ARRAY [0..31] OF BOOL	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1	1	0	1	1	0	1	0	0	0	1	0
uni2	WORD														23,970																		
uni3	ARRAY [0..1] OF WORD	79													23,970																		
uni4	DWORD	5,201,314																															

8.4.5 內建聯合的支援與名稱

部分機種在 ISPSOft 內部有支援內建的聯合供使用者使用，請參考下表。

機種 聯合	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
Union16	V	V	V	V	V		
Union32	V	V	V	V	V		
Union64		V	V	V	V		

下表為內建的聯合所包含的成員及資料型態，使用者可於宣告符號時，在使用者自訂資料型態選取。

聯合 成員	Union16	Union32	Union64
uArrB	ARRAY [0..15] OF BOOL	ARRAY [0..31] OF BOOL	ARRAY [0..63] OF BOOL
uW	WORD		
uI	INT		
uArrW		ARRAY [0..1] OF WORD	ARRAY [0..3] OF WORD
uArrI		ARRAY [0..1] OF INT	ARRAY [0..3] OF INT
uDW		DWORD	
uDI		DINT	
uR		REAL	
uArrDW			ARRAY [0..1] OF DWORD
uArrDI			ARRAY [0..1] OF DINT
uArrR			ARRAY [0..1] OF REAL
uLW			LWORD
uLI			LINT
uLR			LREAL

MEMO



第9章 軸

目錄

9.1 軸	9-2
9.1.1 軸的意義	9-2
9.1.2 建立軸	9-3
9.1.2.1 AHxxEMC 機種的建立軸	9-3
9.1.2.2 DVPxxMC / AS5xx 機種的建立軸與軸基本設定	9-5
9.1.3 全域符號表的主符號表與軸符號表	9-9
9.2 軸的參數設定、監控與調測	9-10
9.2.1 軸參數設定	9-10
9.2.1.1 AHxxEMC 機種的軸參數設定	9-10
9.2.1.2 DVPxxMC / AS5xx 機種的軸參數設定	9-14
9.2.2 監控與調測	9-21
9.2.2.1 AHxxEMC 機種的監控與調測	9-21
9.2.2.2 DVPxxMC/AS5xx 機種的監控與調測	9-26
9.2.3 伺服參數備份還原	9-33
9.2.4 自動增益調整	9-36

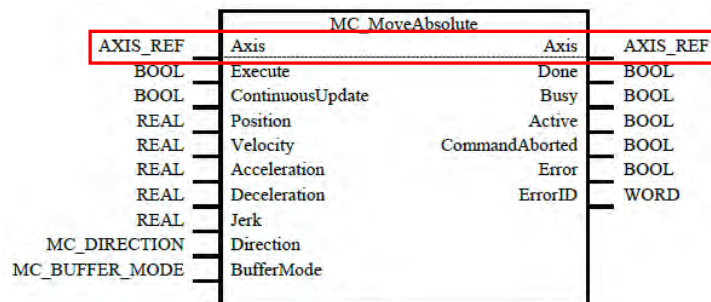
9.1 軸

支援軸相關功能的機種請參考下列表格。

功能 \ 機種	AHxxEMC	DVPxxMC/AS5xx
軸參數設定	V	V
監控與調測	V	V
伺服參數備份還原	V	
自動增益調整	V	只支援 DVP50MC/DVP50MC-06/AS516E-B

9.1.1 軸的意義

軸在軟體上的定義，依據 PLCopen MC 規範定義，「軸」為用於運動控制功能塊中的 STRUCTURE 資料型態 VAR_IN_OUT 參數。軸提供了程式中的運動控制到實際輸出設備中的一個介面，軸中紀錄了各項運動控制相關資訊，例如位置、速度、...，等等。如下圖為 PLCopen MC 規範定義的功能塊，而 AXIS_REF 便為規範定義的軸。



在實際應用上，便是利用各種運動功能塊控制軸，再輸出到目標的伺服馬達驅動器上，達到運動控制。

*.DVPxxMC/AS5xx 機種，目前因開發延續性內容的關係，並不遵守「軸」用於運動控制功能塊中的 STRUCTURE 資料型態 VAR_IN_OUT 原則，而採用 STRUCTURE 資料型態 VAR_INPUT。

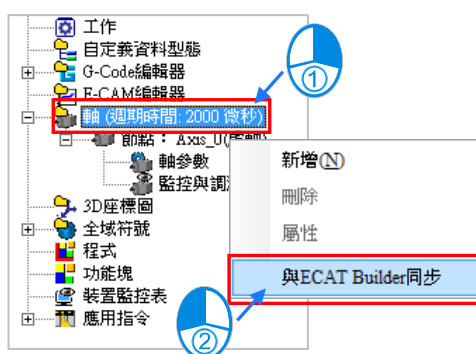
9.1.2 建立軸

9.1.2.1 AHxxEMC 機種的建立軸

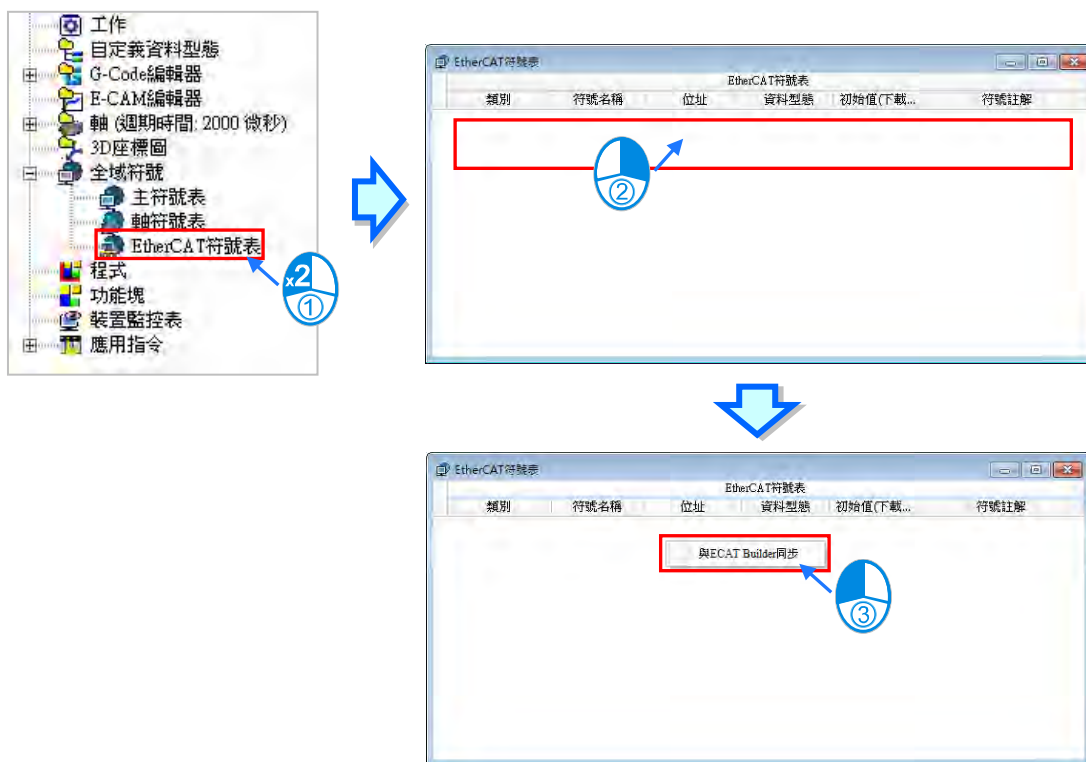
在專案管理區的軸項目點擊滑鼠右鍵，於快捷選單點選「新增」，出現新增軸視窗，輸入軸符號名稱，按下「確定」按鈕，即會於軸項目下新增此節點項目。須注意軸符號名稱不得與其他全域符號或區域符號重複；至於軸項目右側的「週期時間」，為 ECAT Builder 內主站的「週期時間」，「週期時間」的詳細內容請參閱 ECAT Builder 使用手冊第 3.2 節「主站設定」。



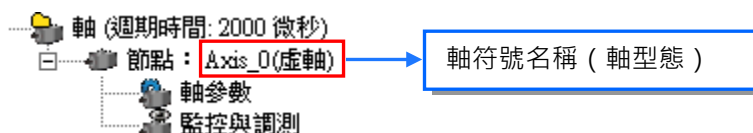
若使用者欲將 ECAT Builder 的資訊與「軸」的資訊（「軸」的資訊包含週期時間、節點資訊）同步，可先在 ECAT Builder 內做存檔的動作，接著於專案管理區的軸項目點擊滑鼠右鍵，於快捷選單點選「與 ECAT Builder 同步」，此時刻 ECAT Builder 的資訊將與「軸」的資訊同步，且與 EtherCAT 符號表的内容同步。



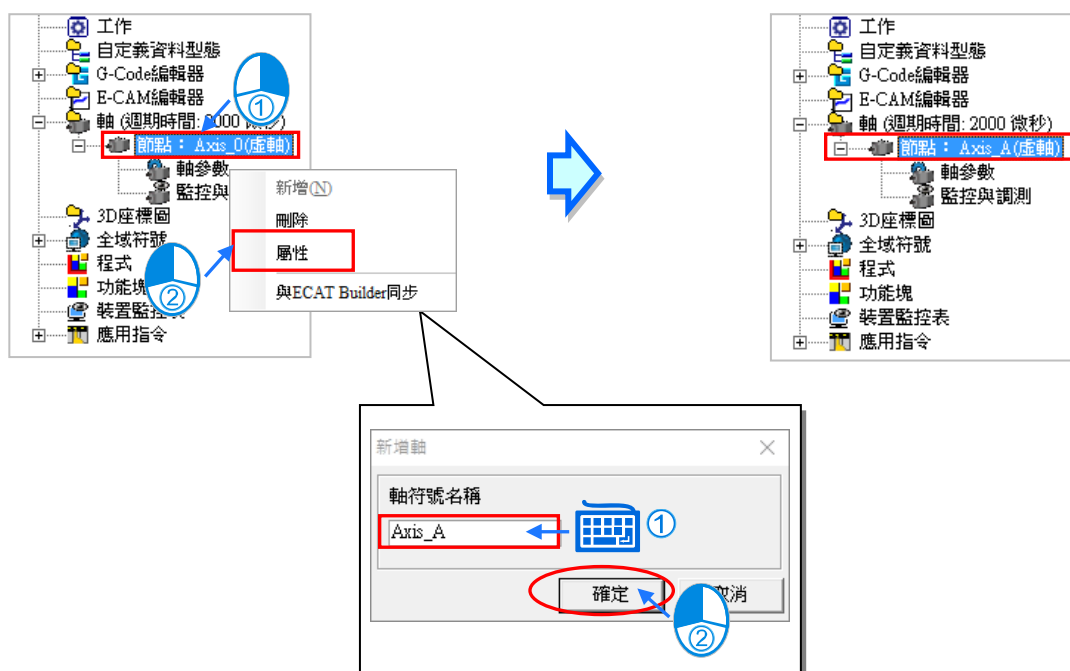
除了利用軸項目做同步更新，使用者也可利用在 EtherCAT 符號表點擊滑鼠右鍵，於快捷選單點選「與 ECAT Builder 同步」，將 ECAT Builder 的資訊與「軸」的資訊、EtherCAT 符號表的內容做同步。



軸節點項目名稱列出前述的軸符號名稱與軸型態，軸型態為依據參數設定中的軸型態設定，可為虛軸或實軸的 EtherCAT 從站名稱與站號。每個軸項目下有軸參數和監控與調測兩個子項目，請參閱後續章節。

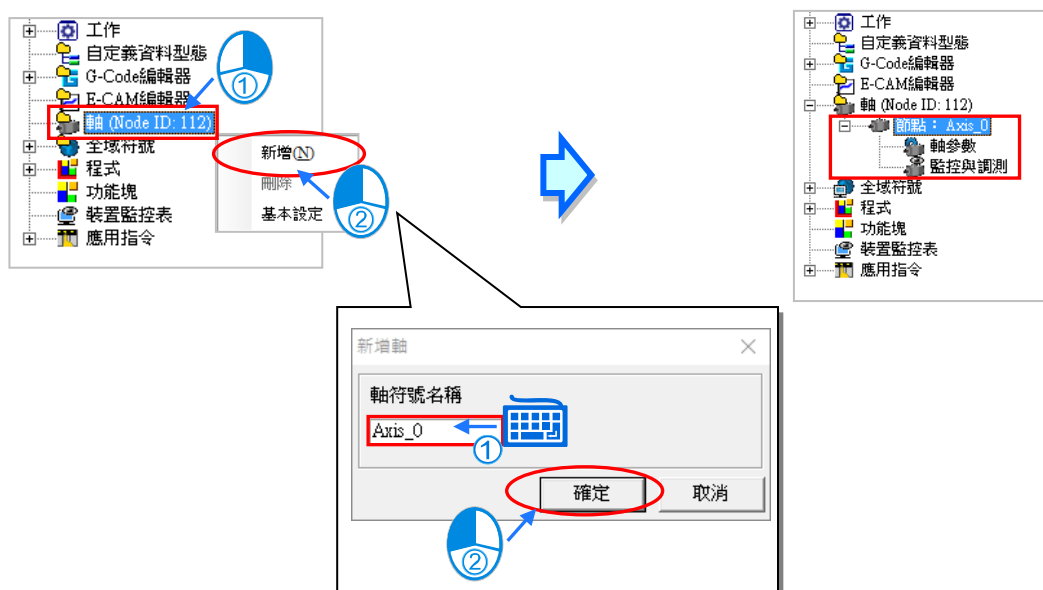


在已建立的軸上點擊滑鼠右鍵，可於快捷選單點選「刪除」，刪除此軸；或是點選「屬性」，重新設定軸符號名稱。



9.1.2.2 DVPxxMC / AS5xx 機種的建立軸與軸基本設定

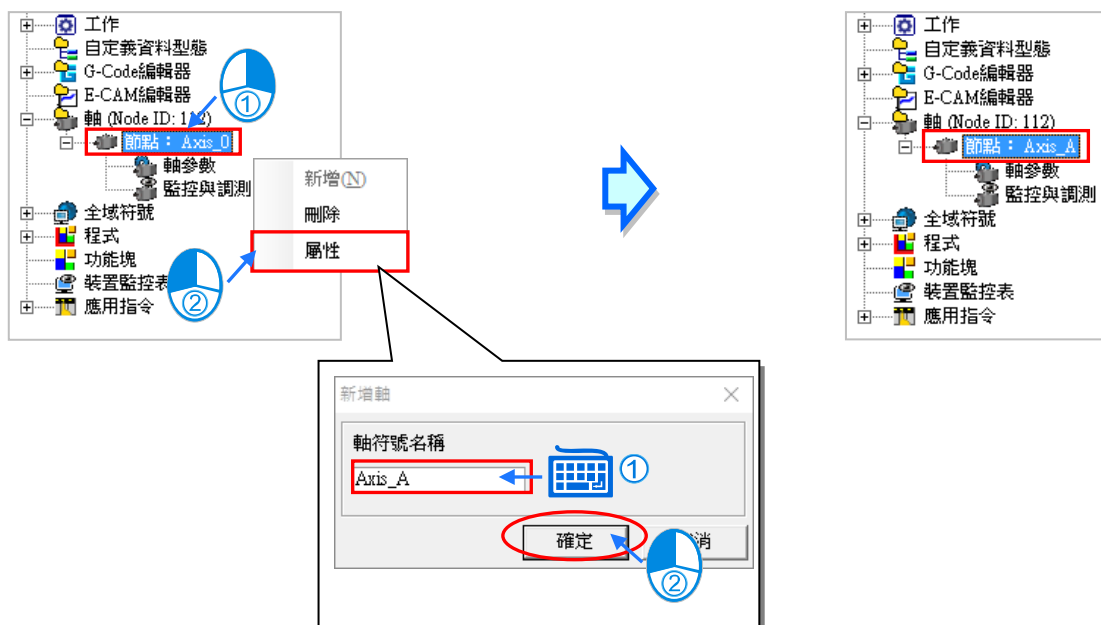
在專案管理區的軸項目點擊滑鼠右鍵，於快捷選單點選「新增」，出現新增軸視窗，輸入軸符號名稱，按下「確定」按鈕，即會於軸項目下新增此節點項目，目前 DVPxxMC 與 AS5xx 機種建立軸之最大數目為 32 軸，但實軸最大僅支援 24 軸，而關於實軸、虛軸等之軸類型相關設定，可參考第 9.2.1.2 節。須注意軸符號名稱不得與其他全域符號或區域符號重複；至於軸項目右側的「NodeID:112」，為控制器 Motion 站號，預設是 112，不可更改。



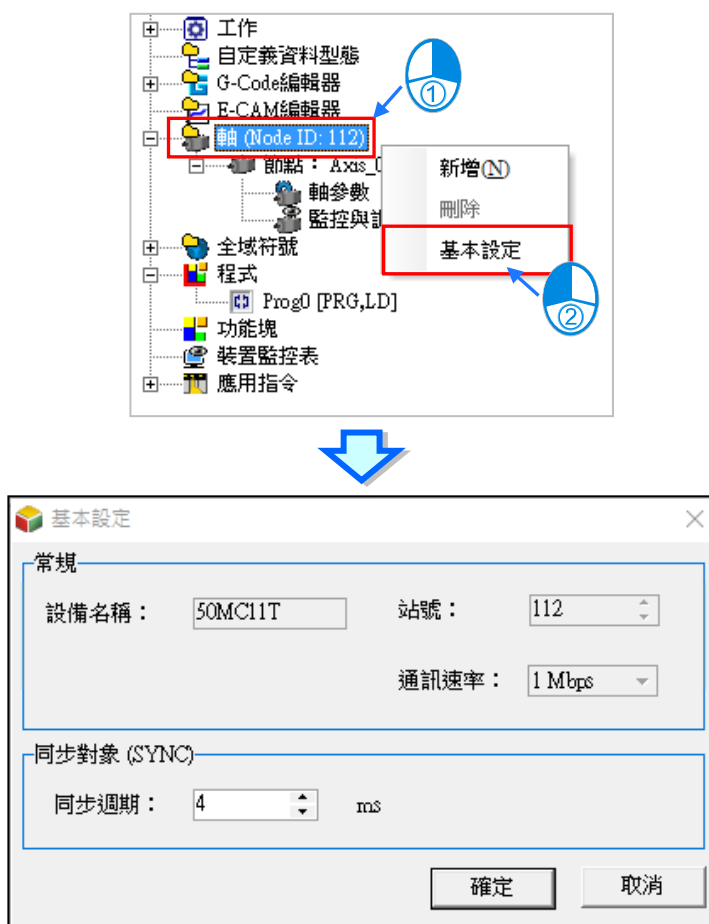
軸節點項目會列出前述的軸符號名稱。每個軸項目下有軸參數和監控與調測兩個子項目，請參閱後續章節。



在已建立的軸上點擊滑鼠右鍵，可於快捷選單點選「刪除」，刪除此軸；或是點選「屬性」，重新設定軸符號名稱。



若使用者需要調整軸的基本設定，於專案管理區的軸項目點擊滑鼠右鍵，於快捷選單點選「基本設定」，即可調整軸的基本設定。



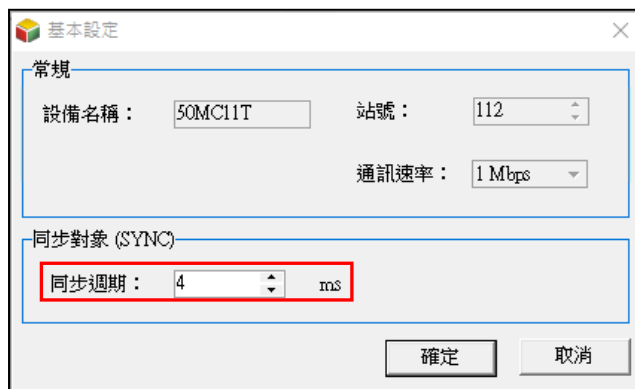
而軸的基本設定可分成「常規」與「同步對象 (SYNC)」兩個項目，接下來就這兩個項目作介紹。

- 常規



- ❶ 設備名稱：預設為專案使用的設備名稱，不可修改。
- ❷ 站號：預設為 112，不可修改。
- ❸ 通訊速率：此處會顯示 HWCONFIG 設置的 CANMotion 通訊速率，此處 CANMotion 通訊速率之設定可參考第 3.6.2 節。

● 同步對象 (SYNC)



同步週期：此處設定同步週期，主站按照同步週期設定時間週期性發送同步封包。在總線運動控制方式中，同步週期是一個非常重要的參數。因此如果同步週期的設定不當，可能會導致伺服在通訊中出現 AL303/AL302/ AL301 報警，或者一些伺服動作異常現象。

首先我們說明一下同步週期的構成。每個同步週期開始，會先掃描運動控制程式，掃描結束後將運算所得的控制資訊發送給所有軸。因此同步週期可以理解為運動控制程式執行時間 + DVPxxMC 和所有伺服的通訊時間。

運動控制程式執行時間可以參考系統變數中最大程式執行時間的數值，單位為微秒。1000 微秒為 1 毫秒。在實際應用中，建議向上取整數。例如最大程式執行時間數值為 2567 微秒=2.5 毫秒，我們可以取 3 毫秒作為程式執行時間。

DVPxxMC 和伺服通訊的時間可以大概預估為每軸 0.5 毫秒。在時間應用中，建議向上取整數，例如 應用中配置 5 個伺服。通訊時間為 5×0.5 毫秒 = 2.5 毫秒。我們可以取 3 毫秒作為通訊時間。

因此可以得到計算公式：同步週期時間 (毫秒) = 最大程式執行時間數值向上取整數 (毫秒) + DVPxxMC 和所有伺服的通訊時間 (毫秒) + 1 (程式變更預留時間) (毫秒)。

為了預防程式變更後，程式的執行時間增大較多，導致之前設定的同步時間不適用，因此在設定的時候儘量做 1 到 2 毫秒的預留時間。

例如：最大程式執行時間為 1634 微秒，應用中共有 5 個伺服。程式變更預留時間 1 毫秒。

同步週期 = 2 毫秒(最大程式執行時間 1634 微秒向上取整數) + 3 毫秒(5×0.5 向上取整數) + 1 毫秒 (程式預留時間) = 6 毫秒。

DVPxxMC 與 AS5xx 機種，目前有提供軸的相關系統結構，可參考下表。

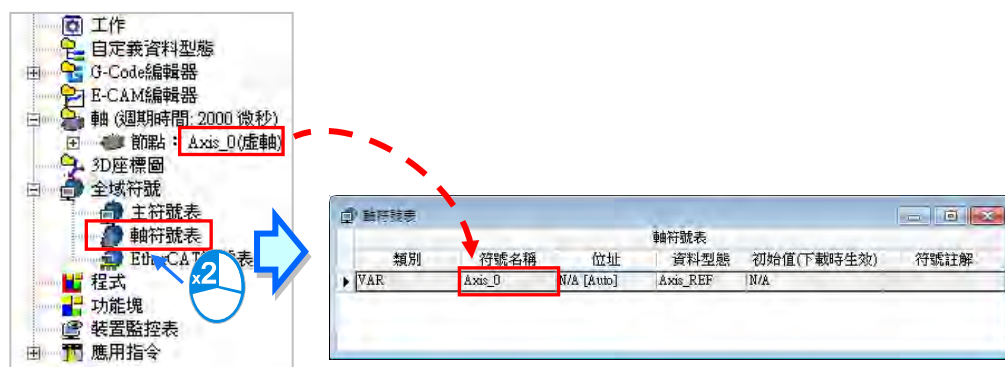
成員變數	資料類型	功能描述
CmdPos	LREAL	命令位置 (單位: 單位)
CmdVel	LREAL	命令速度 (單位: 單位/秒)
CmdAcc	LREAL	命令加速度 (單位: 單位/秒 ²)
CmdTrq	INT	命令扭矩 (額定扭矩的千分比)
ActPos	LREAL	實際位置 (單位: 單位)
ActVel	LREAL	實際速度 (單位: 單位/秒)
ActTrq	INT	實際扭矩 (額定扭矩的千分比)
ActCur	INT	實際電流

9.1.3 全域符號表的主符號表與軸符號表

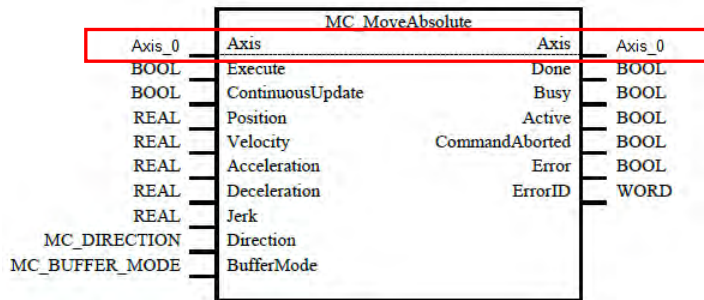
支援軸項目的機種，為了便於使用上的分類管理，全域符號表將包含多個子項目，其中包括**主符號表**、**軸符號表**，與其他**網路符號表**。主符號表相當於其他機種的全域符號表，用於編輯全域符號。而軸符號表則列出所有已建立的軸符號名稱。其他網路符號表為此機種所支援的總線網路，由其他網路配置軟體 (如 ECAT Builder) 所帶入的符號資料，關於其他網路配置軟體，請參閱各軟體操作手冊。



完成軸的建立後，在**全域符號表**中的**軸符號表**便會自動產生此軸符號名稱的變數符號，所以軸符號表內容無法編輯，如下圖以滑鼠左鍵雙擊軸符號表項目，開啟軸符號表。



軸變數符號可用於 PLCopen MC 功能塊與台達功能塊的接腳。如下圖。

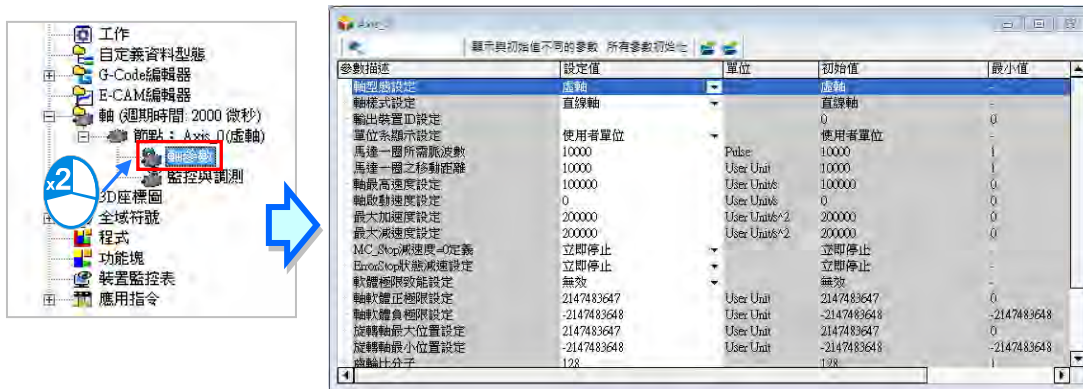


9.2 軸的參數設定、監控與調測

9.2.1 軸參數設定

9.2.1.1 AHxxEMC 機種的軸參數設定

在專案管理區的軸建立一個節點項目後，便會自動開啟此軸的軸參數視窗。關閉後使用者也可自行雙擊此軸節點項目下的軸參數項目再次開啟參數設定視窗，此視窗列出此軸所有關於運動控制的設定，設定項目說明如下，詳細意義可參照各機種手冊。完成設定後便可與專案一同下載至主機。



軸型態設定

用於設定有實際輸出到驅動元件的**實軸**，或僅在控制器中運算的**虛軸**。

軸樣式設定

對於不同的應用場合可設定軸為**直線軸**或**旋轉軸**，直線軸為朝正方向（或負方向）旋轉時位置會不斷遞增（遞減），而旋轉軸為朝正方向（或負方向）旋轉到達極限值後；位置值會從最小值（最大值）開始重新計算。

9

輸出裝置 ID 設定

其中當軸型態設定為實軸時，輸出裝置 ID 設定必須由 ECAT Builder 中帶入的伺服名稱與站號選擇，代表此軸為實際輸出到設定的 EtherCAT 伺服驅動器。如下圖，輸出裝置 ID 設定選擇為名稱 ASDA-A2-E CoE Drive 站號 5 的 EtherCAT 從站。使用者須注意輸出裝置 ID 設定的下拉選單內容，是在 ECAT Builder 存檔後，關閉 ECAT Builder 且關閉 HWCONFIG 時，會更新；或在 ECAT Builder 存檔後，有將「軸」的資訊與 ECAT Builder 同步時，會更新，「軸」的資訊與 ECAT Builder 同步方式可參閱第 9.1.2 節。



單位係顯示設定

選擇系統單位；以使用者單位為單位。

馬達一圈所需脈波數

設定控制器送出多少脈波將使伺服馬達旋轉一圈。

馬達一圈之移動距離

設定伺服馬達旋轉一圈將使實際機構移動多少距離。

軸最高速度設定

設定軸允許的速度上限。

軸啟動速度設定

軸開始運動時的啟動速度。

最大加速度設定

設定軸允許的加速度上限。

最大減速度設定

設定軸允許的減速度上限。

MC_Stop 減速度=0 定義

當以指令或是發生錯誤使軸停止時，設定軸停止的方式，可設定立即停止或以目前的減速度減速停止。

ErrorStop 狀態減速設定

當軸運動超越軟極限設定而使軸停止時，設定軸停止的方式，可設定立即停止或以目前的減速度減速停止。

軟體極限致能設定

設定是否開啟軟體極限，極限值由下兩參數設定。

軸軟體正極限設定

設定軸軟體正極限值，只適用於直線軸。

軸軟體負極限設定

設定軸軟體負極限值，只適用於直線軸。

旋轉軸最大位置設定

設定旋轉軸朝正方向旋轉的最大位置，旋轉到達極限值後；位置值會從最小值開始重新計算。

旋轉軸最小位置設定

設定旋轉軸朝負方向旋轉的最小位置，旋轉到達極限值後；位置值會從最大值開始重新計算。

齒輪比分子

設定電子齒輪比的分子值。

齒輪比分母

設定電子齒輪比的分母值。

原點復歸加速度

原點復歸加速度，執行原點復歸時的加速度。

原點復歸速度

原點復歸速度，執行原點復歸時的速度。

原點復歸接近速度

執行原點復歸時，馬達先以原點復歸速度運轉，當觸發近原點信號時，馬達開始減速到原點復歸接近速度運轉。

位置偏差容許值

此值為實際位置與命令位置誤差容許值。

單軸運動立即啟動

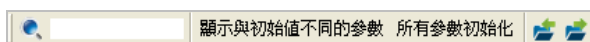
此處可以選擇「關閉」或「啟動」，若選擇「啟動」，當單軸運動啟動後，不會受到 PLC 主機週期時間影響而馬上執行運動，詳細內容請見該機種相關手冊。



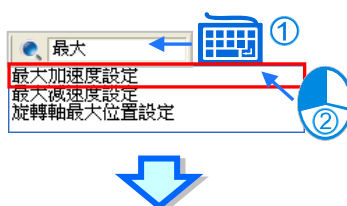
MC_CamIn 主軸方向

此處可以選擇「正向」或「不指定」，當選擇「正向」時，當主軸反向運動時，從軸不會動作；當選擇「不指定」時，從軸不受主軸正反向限制。

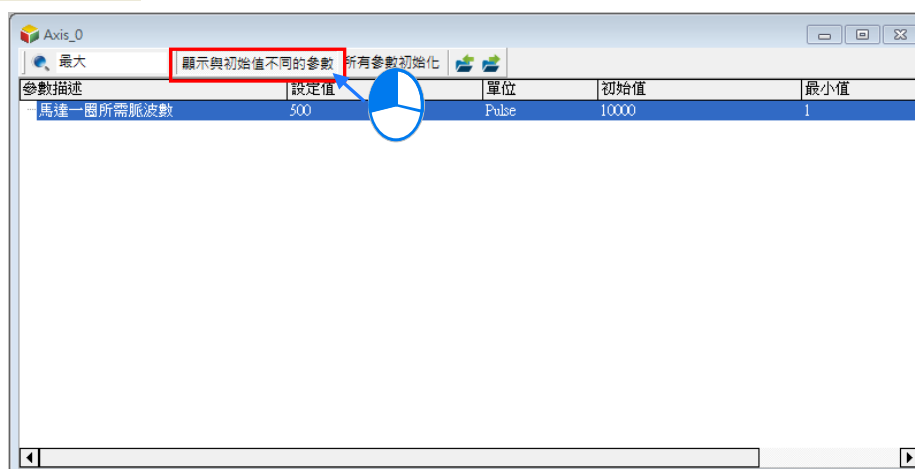
軸參數設定視窗提供了如下圖的工具列，便於編輯此軸參數表格。





搜尋工具，輸入字串後會列出相關的字串，點選後表格會自動鎖定到此欄位。



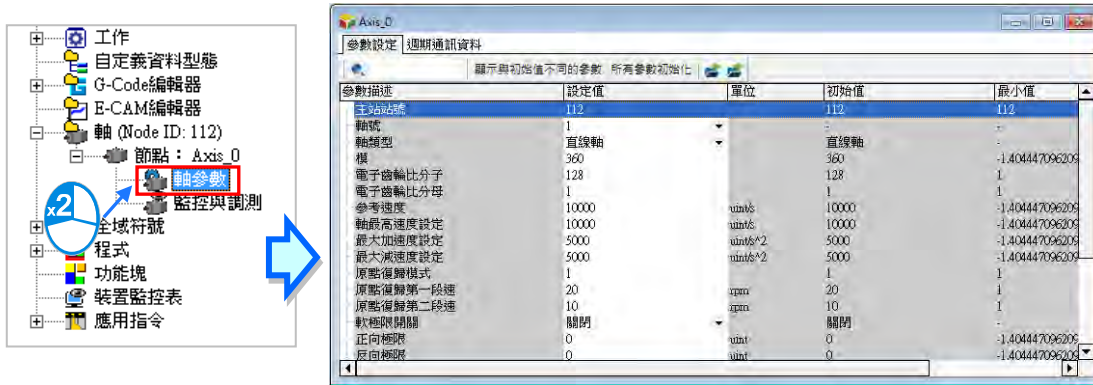
點選「顯示與初始值不同的參數」按鈕後，表格將過濾出與初始設定不同的參數欄位。



點選「所有參數初始化」按鈕，將會把所有設定欄位恢復為初始值。點選匯出  按鈕，可將目前的參數以.CSV格式匯出到指定路徑；點選匯入  按鈕可將指定路徑的設定檔案匯入。

9.2.1.2 DVPxxMC / AS5xx 機種的軸參數設定

若使用者選擇的是 DVPxxMC 或 AS5xx 機種，一旦建立了一個軸節點項目後，便會自動開啟此軸的軸參數視窗。關閉後使用者也可自行雙擊此軸節點項目下的**軸參數**項目再次開啟參數設定視窗。



而軸參數視窗列出此軸所有關於運動控制的設定，設定項目可分為兩種，「參數設定」和「週期性通訊資料」，「參數設定」的參數內容詳細意義可參照各機種手冊。完成設定後便可與專案一同下載至主機。

- 參數設定



參數設定的內容如下：

主站站號

此項目為主站站號，目前固定為站號 112。

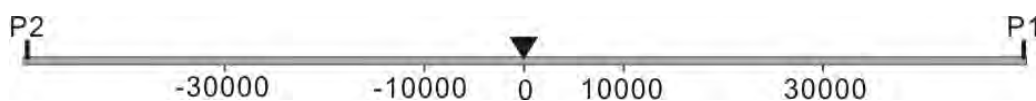
軸號

此處設定軸號，取值範圍 1 ~ 32。

軸類型

用以選擇軸的類型，分為「直線軸」和「旋轉軸」。

➤ 直線軸

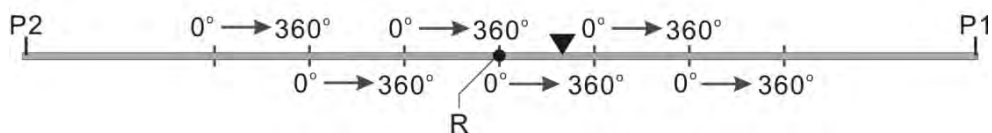
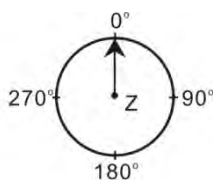


直線軸模型

直線軸模型註解：

P1	正極限
P2	反極限
▼	伺服所在的位置

➤ 旋轉軸



旋轉軸模型（「模」設為 360）

旋轉軸模型註解：

P1	正極限
P2	反極限
▼	伺服所在的位置
R	原點位置
Z	伺服馬達的軸心

模

用以平分終端執行機構實際位置的週期。

電子齒輪比分子

和電子齒輪比分母一起設定馬達轉一圈需要的脈波數。

電子齒輪比分母

和電子齒輪比分子一起設定馬達轉一圈需要的脈波數。

參考速度

此參數功能暫時保留，對控制器功能無影響。

軸最高速度設定

設定軸連接機構的最大速度。

最大加速度設定

此參數功能暫時保留，對控制器功能無影響。

最大減速度設定

此參數功能暫時保留，對控制器功能無影響。

原點復歸模式

設定伺服驅動器原點復歸模式，取值範圍 1~35。詳細內容請參考該機種相關手冊。

原點復歸第一段速

為原點復歸開始找到原點開關的速度。

原點復歸第二段速

為找到原點開關後到機械原點的速度。



軟極限開關

設定是否開啟軟體極限。當選項開啟後，軸終端執行機構位置到達設定的「正向極限」或「反向極限」時，運動指令報錯並且停止。

正向極限

允許軸的正向極限。

反向極限

允許軸的反向極限。

偏差錯誤開關

設定控制器是否偵測軸的命令位置和回饋位置的差值。該參數設定為開啟時，控制器偵測命令位置和實際位置誤差的值；該參數設定為關閉時，控制器不偵測命令位置和實際位置誤差的值。

位置偏差允許值

允許軸的命令位置和回饋位置之間差值的最大值。

位置偏差持續時間

允許軸的最大位置誤差值的持續時間。

齒輪箱輸入圈數

與齒輪箱輸出圈數一起確定機構齒輪比。

齒輪箱輸出圈數

與齒輪箱輸入圈數一起確定機構齒輪比。

機構導程

齒輪箱輸出端轉一圈對應終端執行機構移動的單元數目。

軸模式

用於設定有實際輸出到驅動元件的**實軸**、僅在控制器中運算的**虛軸**以及配合編碼器應用的**編碼器軸**。


編碼器軸模式

可選擇的模式分為增量型編碼器 1、增量型編碼器 2、絕對型編碼器及資料來源，此項目需在上一項目「軸模式」，選擇在「編碼器軸」時，才可使用。

編碼器軸模式選項	用途
增量型編碼器 1	編碼器軸的資料來源為增量型編碼器 1
增量型編碼器 2	編碼器軸的資料來源為增量型編碼器 2

編碼器軸模式選項	用途
絕對型編碼器	編碼器軸的資料來源為絕對型編碼器
資料來源	編碼器軸的資料來源為指定裝置中的資料

編碼器軸資料來源

此項目需在上一項目「編碼器軸模式」，選擇在「資料來源」時，才可使用。當使用者選擇「資料來源」後，可對「編碼器軸資料來源」的設定值欄位此按鈕點擊滑鼠左鍵，此時將出現「編碼器軸資料來源」視窗，如下圖。



- ❶ 此處下拉選單可以選擇裝置名稱，可選擇%IL、%QL及%ML。
- ❷ 此處可設定裝置的編號。

而選擇在「資料來源」時，編碼器軸的位置變化量為指定裝置值每個掃描週期的變化量。

EtherCAT 節點號

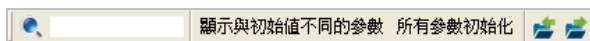
DVP50MC、DVP50MC-06、以及 AS516E-B 的機種支援此項目，節點編號是由 1001 開始一直到 1024。

- 週期性通訊資料

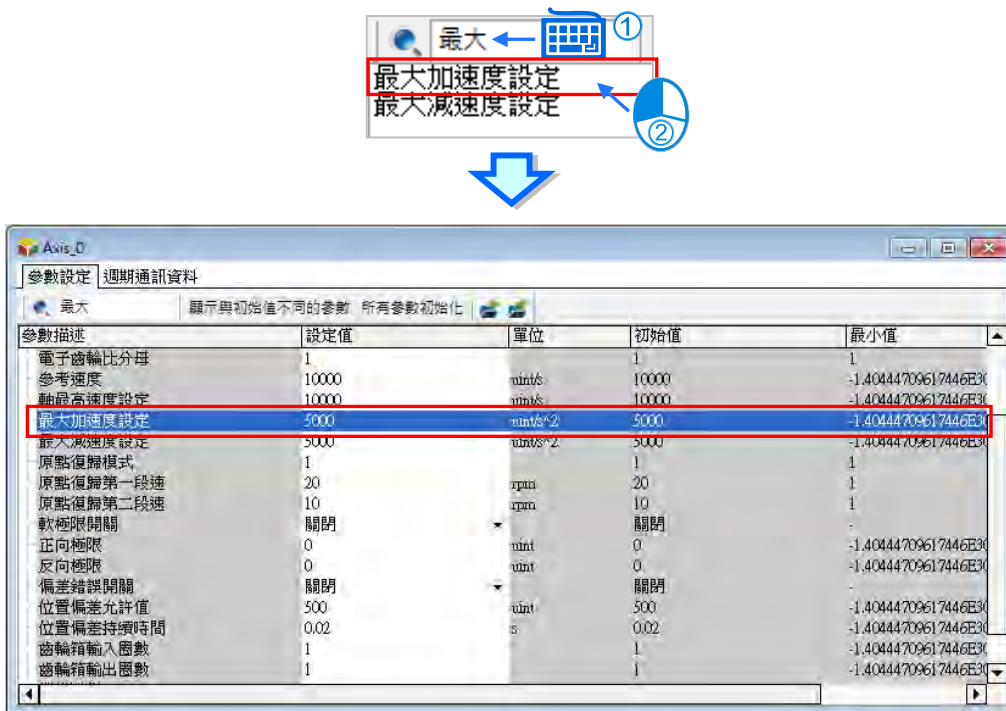


勾選「速度」、「扭矩」、「電流」或者「使用者定義參數」後，則伺服驅動器在每個同步週期後往總線上會發出對資料資訊，週期通訊資料最多選擇 8 個位元組的資料。週期通訊資料對應的資料勾選後，才可以在軸結構體中監控到對應的參數。

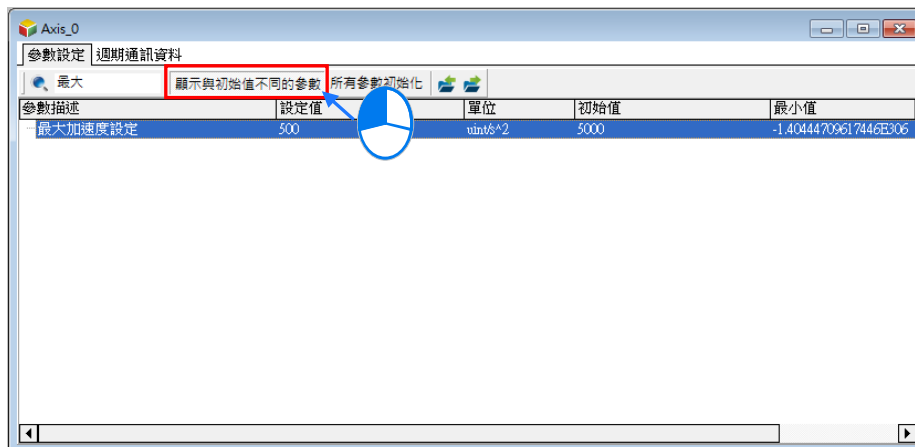
軸參數設定視窗提供了如下圖的工具列，便於編輯此軸參數表格。



搜尋工具，輸入字串後會列出相關的字串，點選後表格會自動鎖定到此欄位。



點選 顯示與初始值不同的參數 按鈕後，表格將過濾出與初始設定不同的參數欄位。

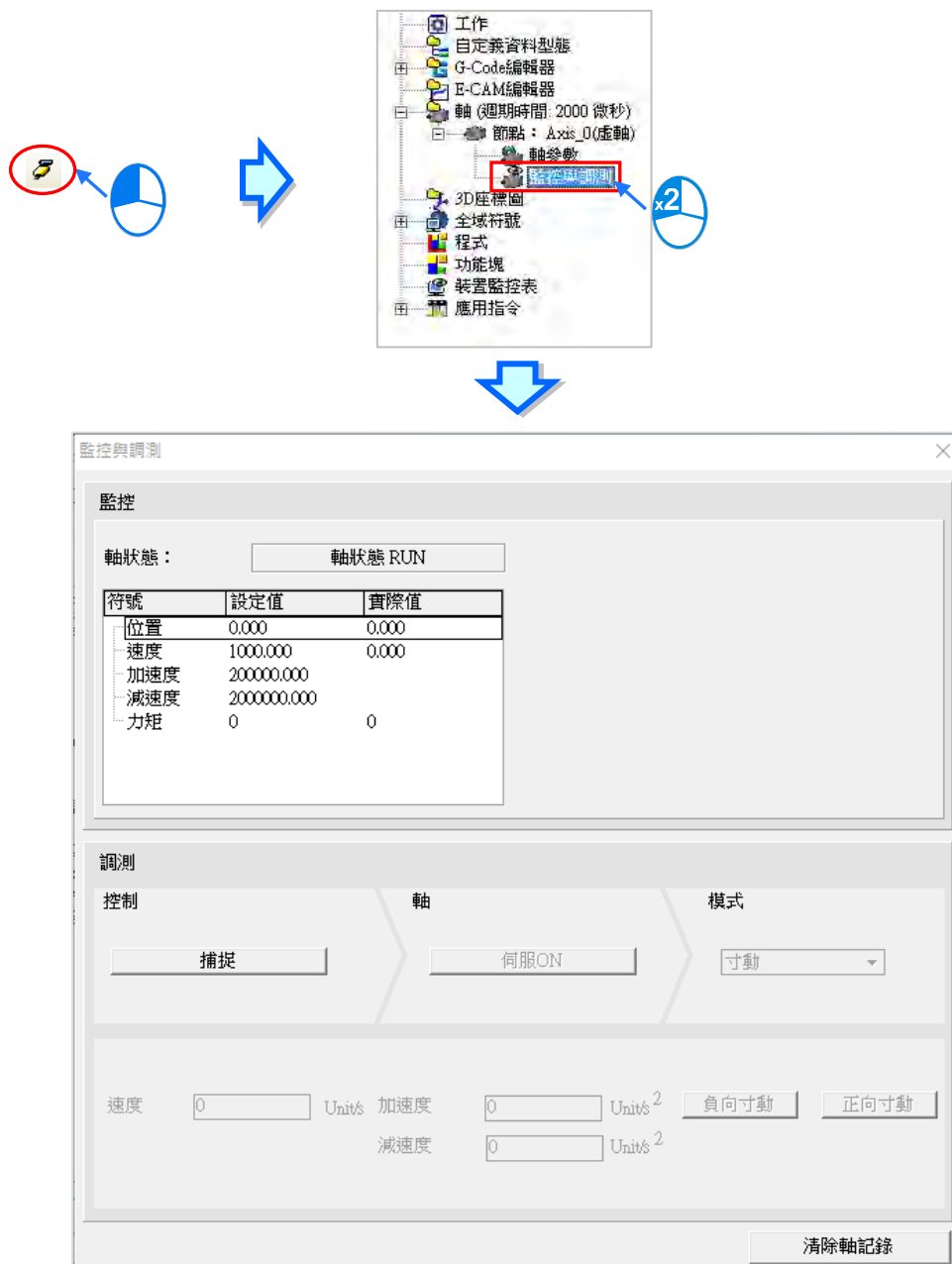


點選 所有參數初始化 按鈕，將會把所有設定欄位恢復為初始值。點選匯出 按鈕，可將目前的參數以.CSV 格式匯出到指定路徑；點選匯入 按鈕可將指定路徑的設定檔案匯入。

9.2.2 監控與調測

9.2.2.1 AHxxEMC 機種的監控與調測

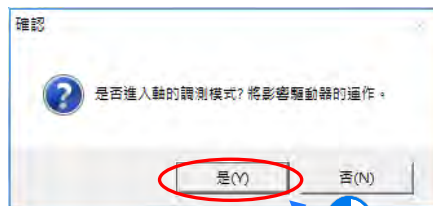
當 ISPSOft 進入線上模式後，以滑鼠左鍵雙擊軸項目下的**監控與調測**項目，開啟監控與調測視窗。如下圖。



視窗的上半部為監控視窗，用於即時顯示軸的狀態與軸的物理量設定值與實際值，例如位置、速度等等。



視窗的下半部為調測視窗，用於在測試階段連接硬體後，對此軸即時下達調測的命令，如下圖，使用時按下「捕捉」的按鈕，將會出現詢問視窗，確認取得對此軸的控制。請注意，對軸進行調測前，須確認現場的配置與人員的位置，在馬達運作時不會造成人員的傷害或物體的損壞。



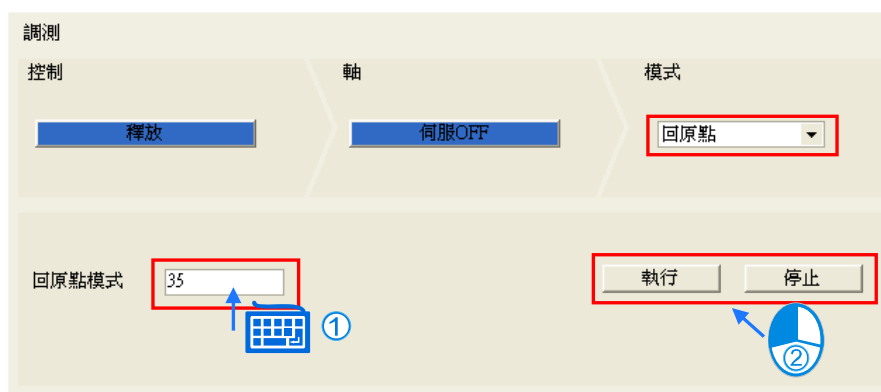
欲關閉調測模式時，點選「釋放」按鈕，便釋放此軸的控制權。

取得軸控制後，接著可按下「伺服 ON」或「伺服 OFF」開關伺服驅動器。



當於**伺服 ON**時，於**模式**的欄位便可選擇各種對軸調測的模式，不同的模式下方也會出現不同的執行功能項目供使用者輸入。對應的功能如下說明。

回原點：依據軸參數的設定執行原點復歸，依據 CANopen 規範的 35 種回原點模式中，輸入**回原點模式**欄位，按下「執行」按鈕，便會開始進行原點復歸動作；原點復歸時按下「停止」按鈕，便會中止動作。



寸動：依據軸參數設定的寸動速度執行寸動，輸入寸動**速度**、寸動**加速度**、寸動**減速度**後，按下按鈕執行「**正向寸動**」或「**負向寸動**」。



速度：輸入**速度**、**加速度時間**、**減速度時間**值。當使用者按下「**正向**」或「**負向**」按鈕時，驅動器以設定的**加速度時間**加速到設定速度維持正向或負向運轉，按下「**停止**」後以設定**減速度時間**減速停止。



相對位置：輸入**相對位置**（正或負表示）、**速度**、**加速度時間**、**減速度時間**值。當使用者按下「**啟動**」按鈕時，驅動器以設定的**加速度時間**加速到設定速度維持運轉，依**減速度時間**減速停止後到達**相對位置**。運轉時按下「**停止**」按鈕，便會中止動作。

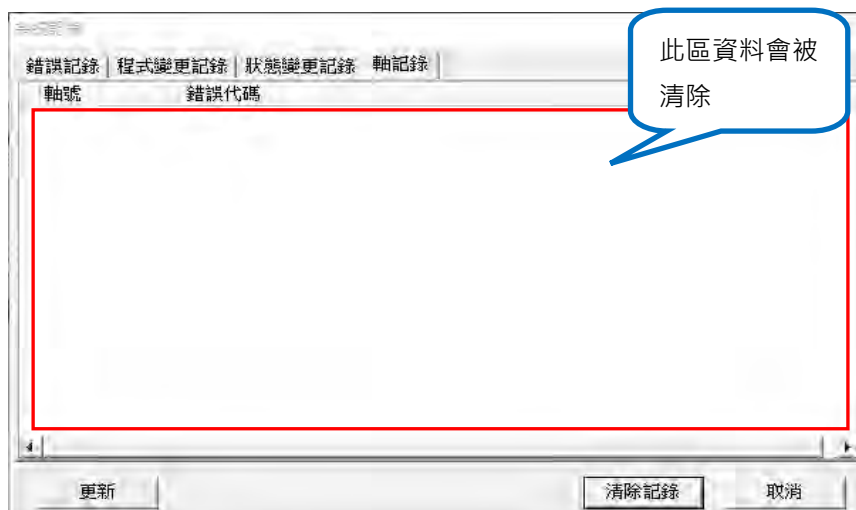
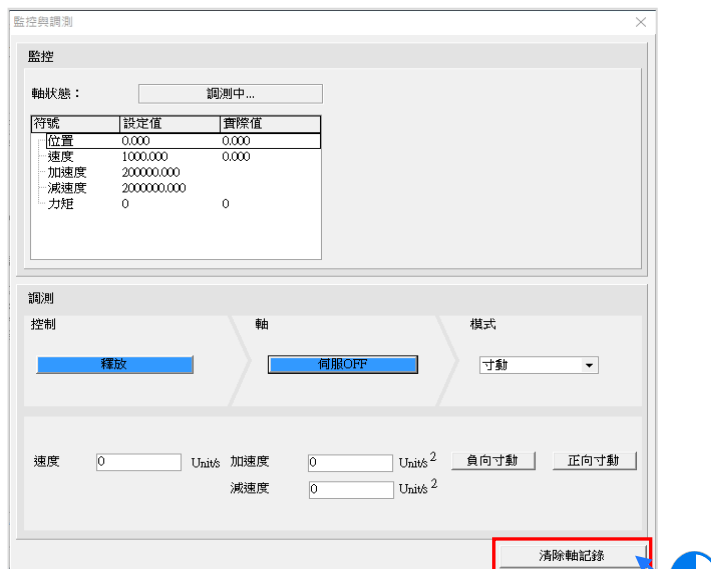


9

絕對位置：輸入位置、速度、加速度時間、減速度時間值。當使用者按下「啟動」按鈕時，驅動器以設定的加速度時間加速到設定速度維持運轉，依減速度時間減速停止後到達絕對位置。運轉時按下「停止」按鈕，便會中止動作。

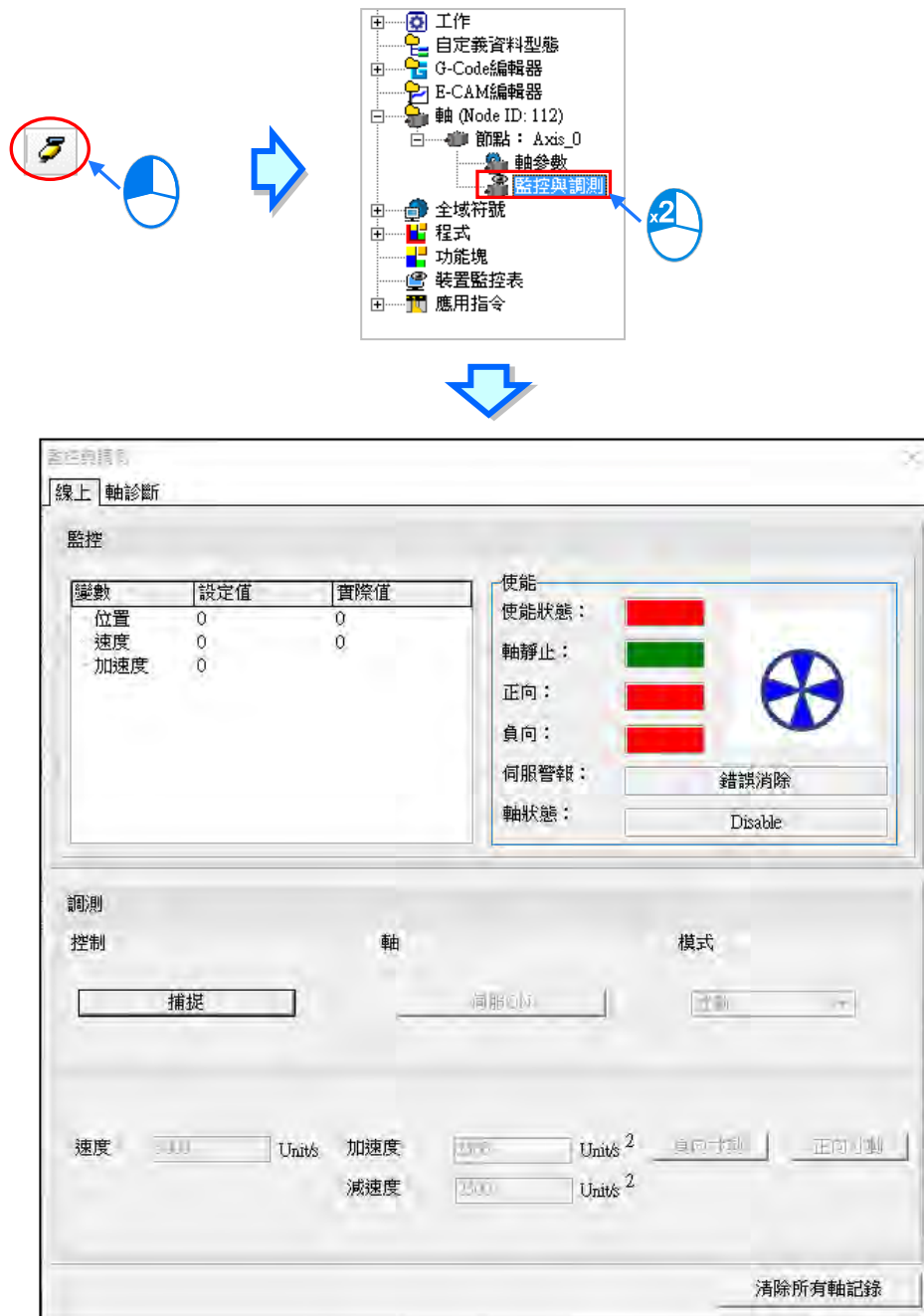


點擊「清除軸記錄」按鈕，會清除系統紀錄內的軸記錄。

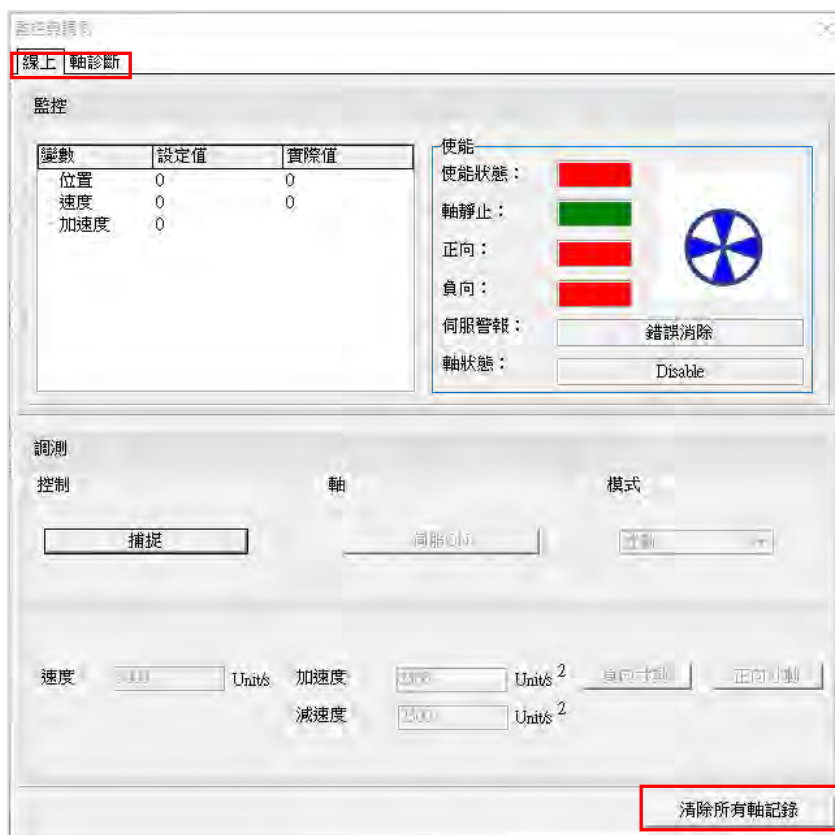


9.2.2.2 DVPxxMC/AS5xx 機種的監控與調測

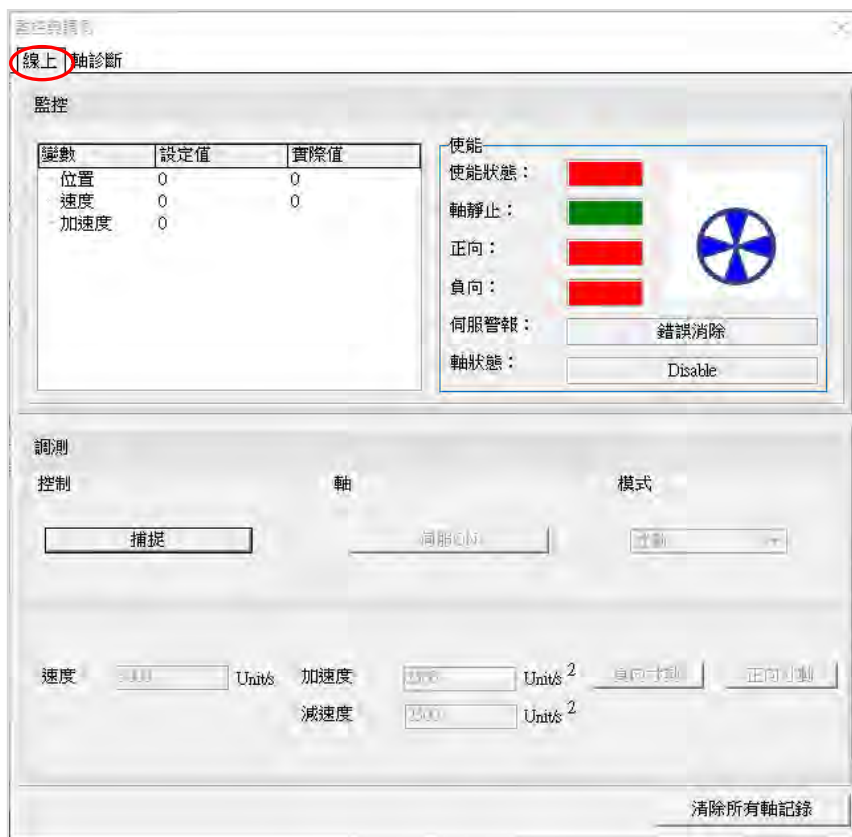
若使用者選擇的是 DVPxxMC 或 AS5xx 機種，當 ISPSOft 進入線上模式後，以滑鼠左鍵雙擊軸項目下的**監控與調測**項目，開啟監控與調測視窗。如下圖。



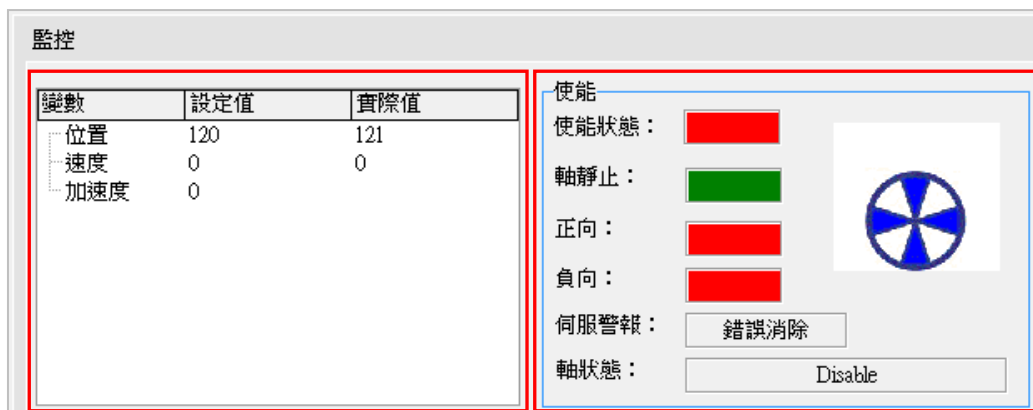
監控與調測視窗，將可見兩個分頁「線上」及「軸診斷」，來做進一步監控調測使用，而底部的「清除所有軸記錄」按鈕，可用來清除專案管理區的所有軸之記錄。



● 線上



「線上」分頁的上半部為監控視窗，左側用於即時顯示軸的狀態與軸的物理量設定值與實際值，例如位置、速度等等，右側「使能」區則會顯示軸之相關狀態及警報，此區域可配合「線上」分頁的下半部之調測視窗做使用。



「線上」分頁的視窗下半部為調測視窗，用於在測試階段連接硬體後，對此軸即時下達調測的命令，如下圖，使用時按下「捕捉」的按鈕，將會出現詢問視窗，確認取得對此軸的控制。請注意，對軸進行調測前，須確認現場的配置與人員的位置，在馬達運作時不會造成人員的傷害或物體的損壞。



欲關閉調測模式時，點選「釋放」按鈕，便釋放此軸的控制權。

取得軸控制後，接著可按下「伺服 ON」或「伺服 OFF」開關伺服驅動器。

*.若使用者選擇之軸模式為「虛軸」，則取得軸控制後，將會自動將此軸維持在伺服 ON 狀態，軸模式設定可參閱第 9.2.1.2 節。



控制器 Motion 通訊口在連接伺服的前提下，點選「**伺服 ON**」，軟體會提示寸動時需注意以下事項。

- (a) 確認週期資料交換中已勾選速度。
- (b) 程式中請勿對該軸使用 MC_Power 指令。
- (c) 如果再寸動過程中，伺服通訊斷線，伺服軸會在 1 秒後自動停止。

而當伺服由 **OFF** 變為 **ON** 時，「使能」區之「使能狀態」項目將由紅燈變為綠燈，且「軸狀態」項目將由「Disable」變為顯示「standstill」。



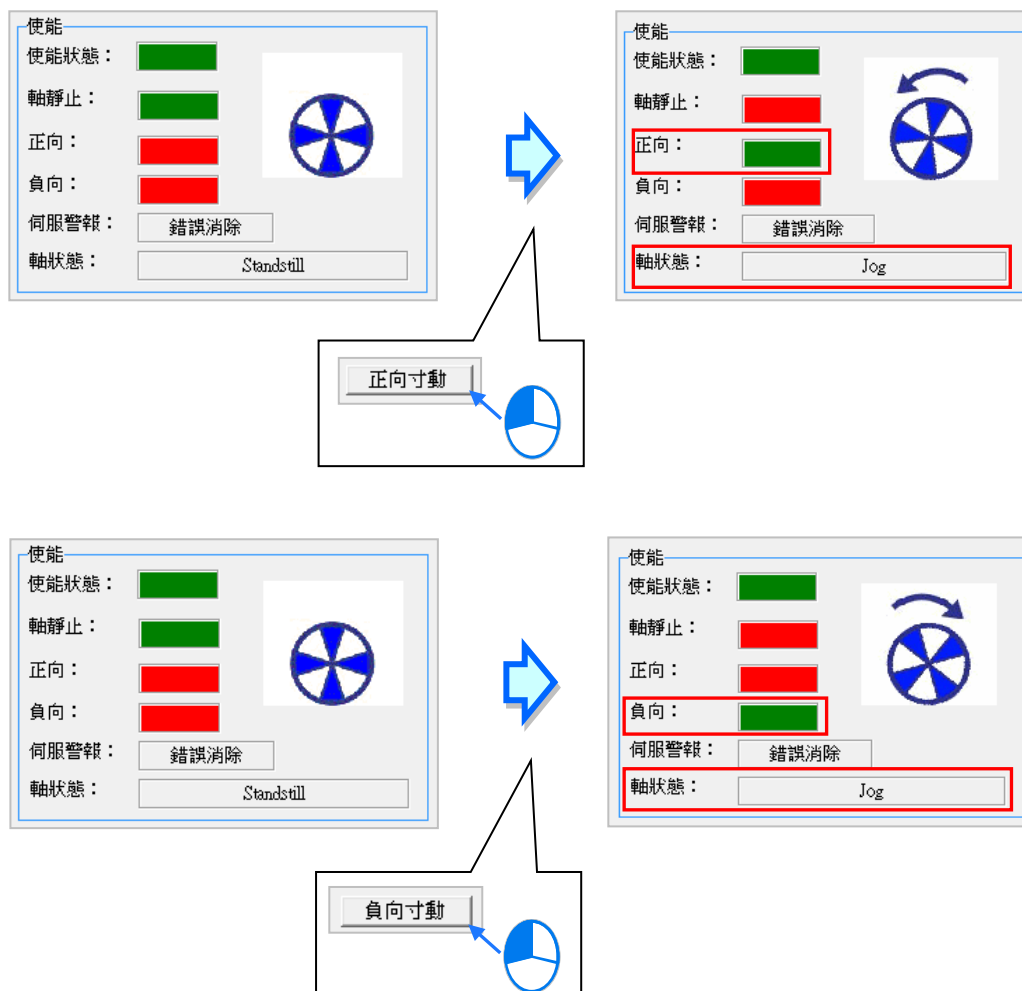
當於**伺服 ON**時，於「**模式**」的欄位便可選擇各種對軸調測的模式，目前僅提供「寸動」供使用者選擇。而其功能如下說明。

寸動：依據軸參數設定的寸動速度執行寸動，輸入寸動**速度**、寸動**加速度**、寸動**減速度**後，按下按鈕執行「**正向寸動**」或「**負向寸動**」。



9

當於按下按鈕執行「正向寸動」/「負向寸動」時，「使能」區之「正向」/「負向」項目將由紅燈變為綠燈，且「使能」區之「軸狀態」項目將顯示「Jog」。(軸之寸動功能需配合控制器為運行時，才可做使用)

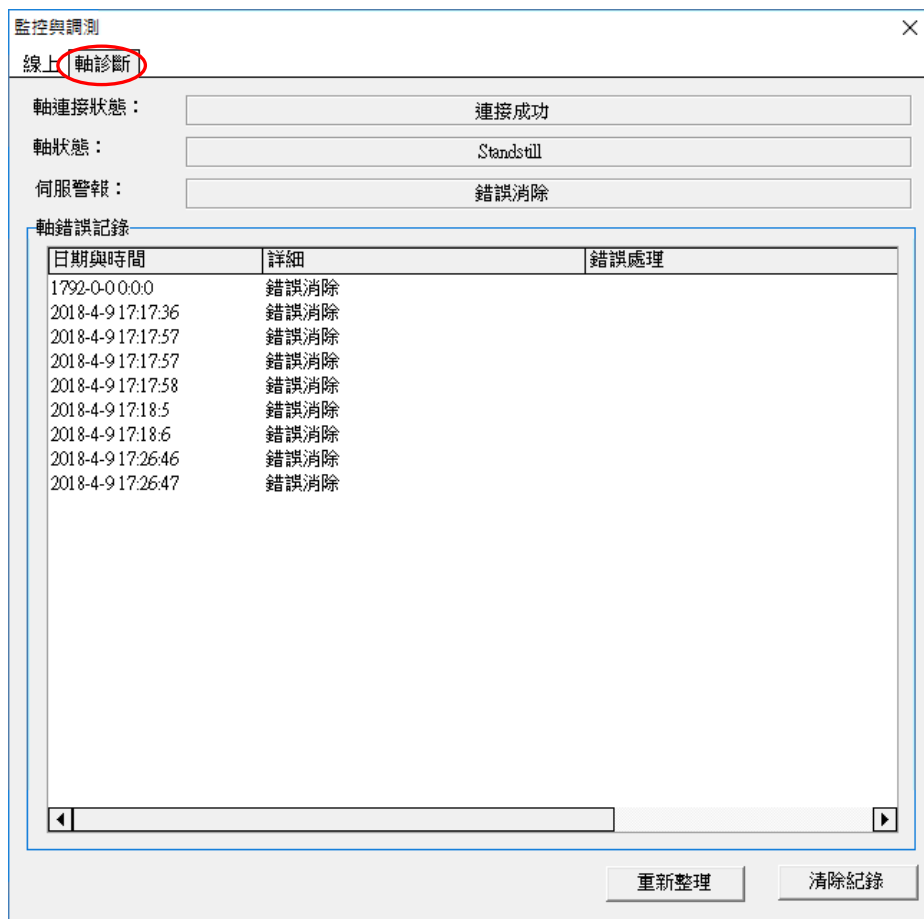


下表為「使能」區各項參數的含義。

參數名稱	說明
使能狀態	綠色表示使能，紅色表示未使能
軸靜止	綠色表示靜止，紅色表示軸寸動
正向	綠色表示軸正轉，紅色表示軸反轉或靜止
負向	綠色表示軸反轉，紅色表示軸正轉或靜止
伺服警報	顯示伺服對應錯誤訊息
軸狀態	分為 Disable、Standstill、ErrorStop、Stopping、Homing、Discrete Motion、Continuous Motion、Synchronized Motion 及 Jog

● 軸診斷

此處用於診斷控制器和軸之間連接是否正常及軸是否警報，會記錄軸連接狀態、軸狀態及伺服警報，而「軸錯誤記錄」區會記錄錯誤原因及錯誤處理。



上圖各項參數的含義如下表所示。

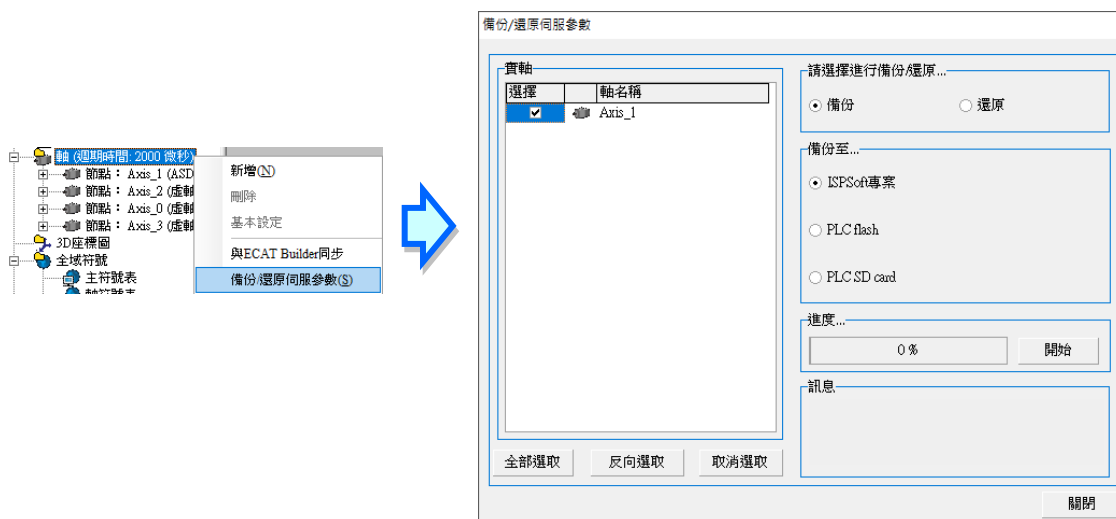
參數名稱	說明
軸連接狀態	控制器和伺服驅動器是否成功連接
軸狀態	軸對應狀態機 (狀態機之介紹可參閱該機種相關手冊)
伺服警報	顯示伺服警報內容
軸錯誤記錄	記錄伺服報警資訊，包括警報發生的時間、可能的原因和錯誤處理方法

點擊「重新整理」，可以更新軸診斷頁面的內容，點擊「清除紀錄」，可以清除目前軸錯誤紀錄的資訊。



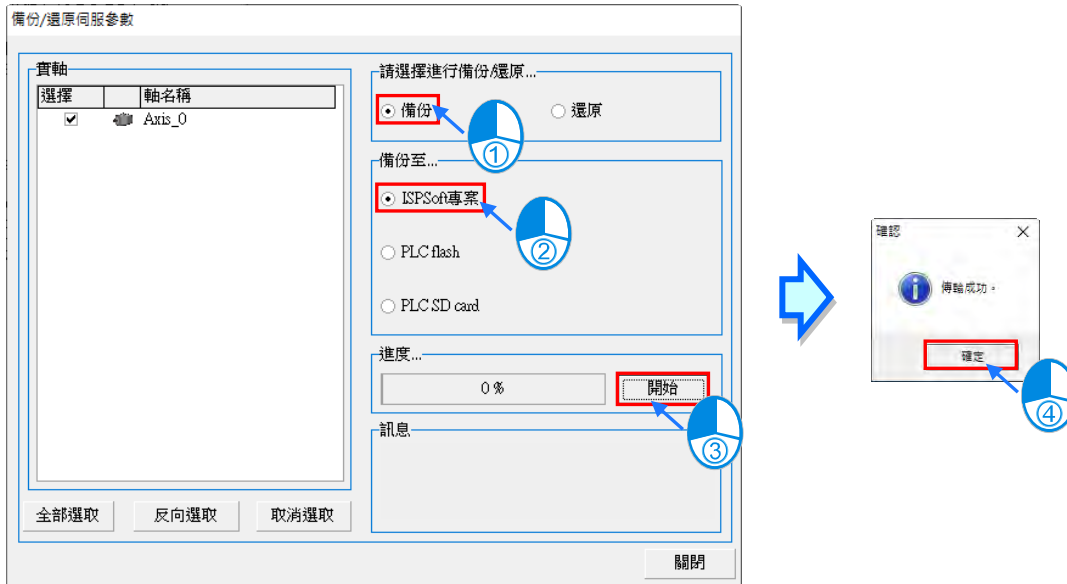
9.2.3 伺服參數備份還原

在「軸」節點點擊滑鼠右鍵，接著點擊「備份/還原伺服參數」，透過備份/還原伺服參數頁面的內容，可以將所連接的實軸設定，備份至 ISPSOft 專案、PLC Flash、PLC SD Card 這三個地方，亦可將備份好的配置還原至 ISPSOft 專案、PLC Flash、PLC SD Card 這三個地方，選好要操作的動作點擊開始即可。



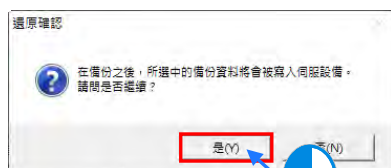
範例將實軸設定備份至 ISPSoft 專案，再從 ISPSoft 專案還原至新專案中。

在「備份/還原伺服參數」頁面中點選「備份」接著再點選備份至...中的「ISPSoft 專案」，最後按下「開始」等待進度條完成後跳出完成視窗，即完成備份。



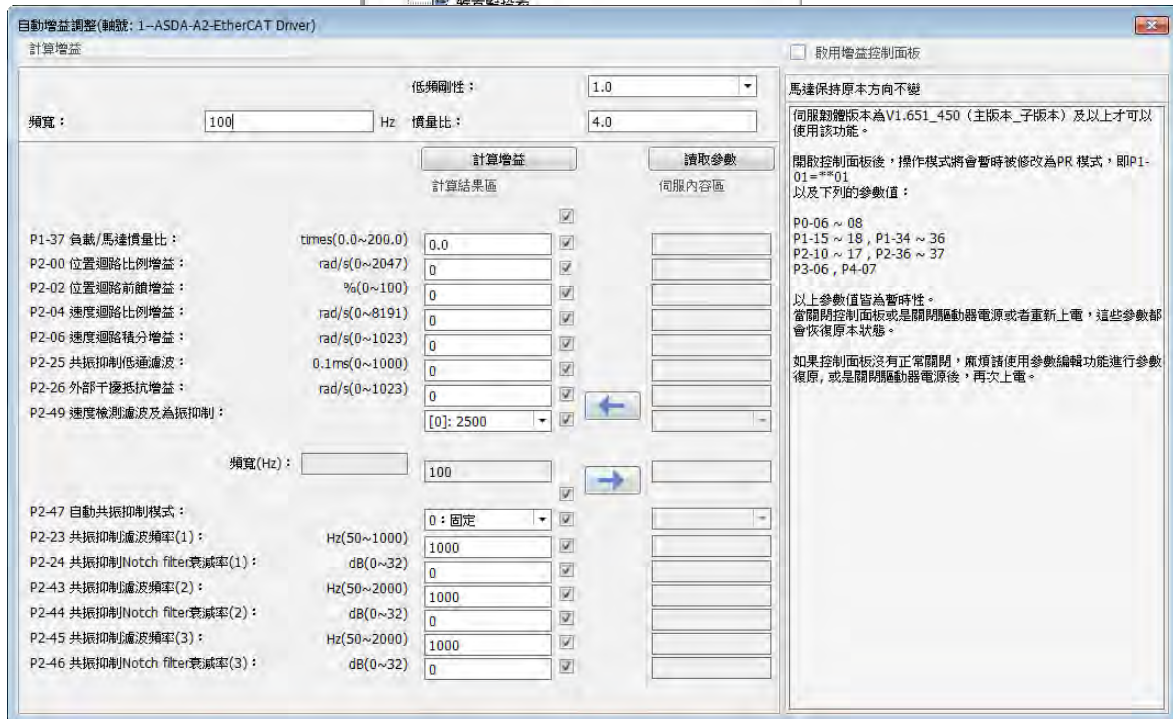
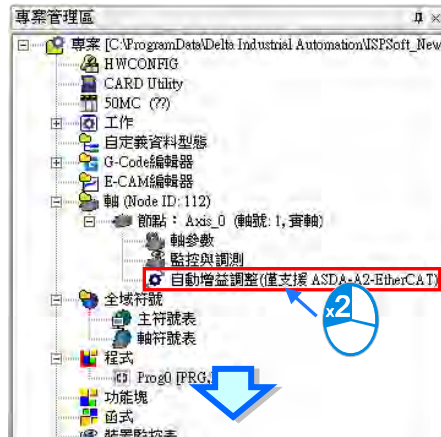
範例將實軸設定從 ISP 專案還原至新專案中。

在「**備份/還原伺服參數**」頁面中點選「**還原**」接著再點選備份至...中的「**ISPSoft 專案**」，最後按下「**開始**」，跳出還原確認視窗，確定將實軸設定寫入伺服設備後點選「**是**」，等待進度條完成後跳出確認視窗即完成還原。

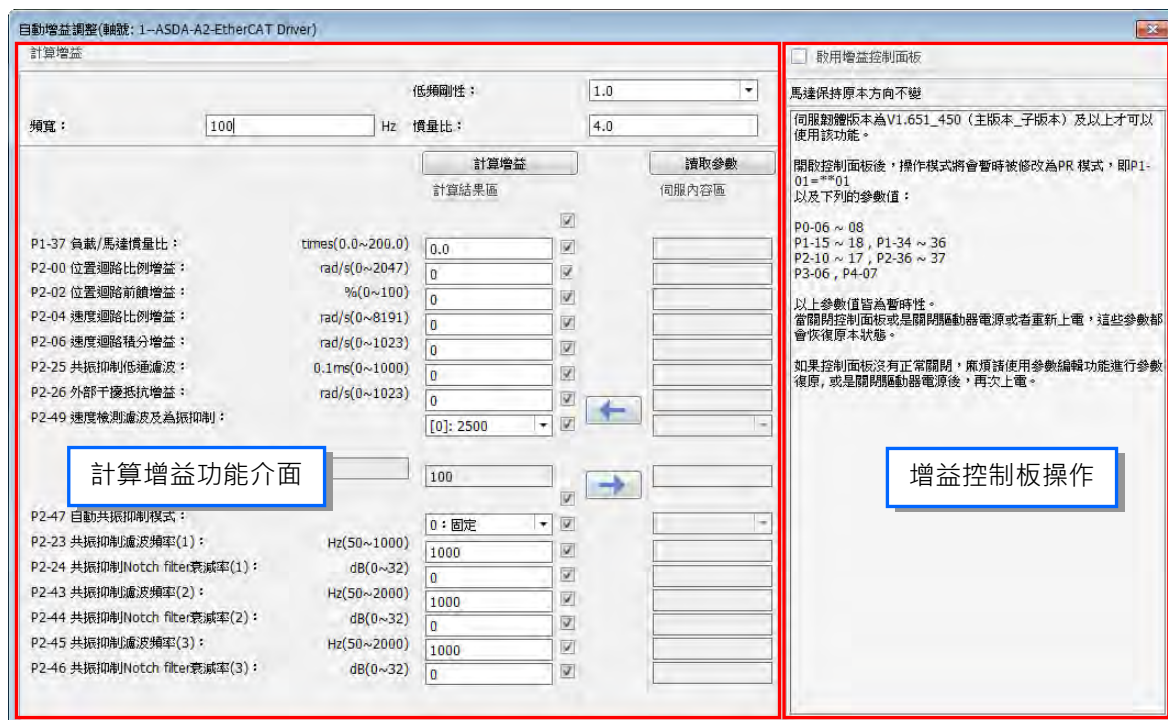


9.2.4 自動增益調整


在專案樹中的「軸」節點下拉選單中，點擊「自動增益調整」打開自動增益調整介面。

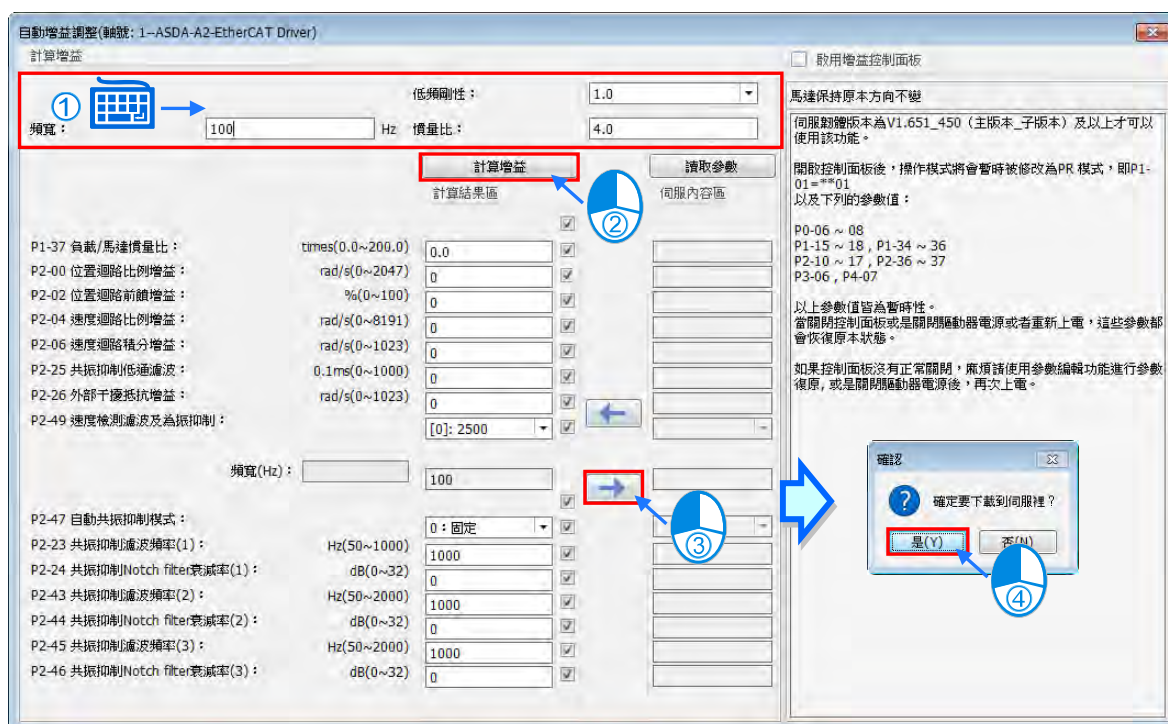


自動增益調整介面中，左半邊為計算增益功能介面，右半邊為增益控制板操作介面。

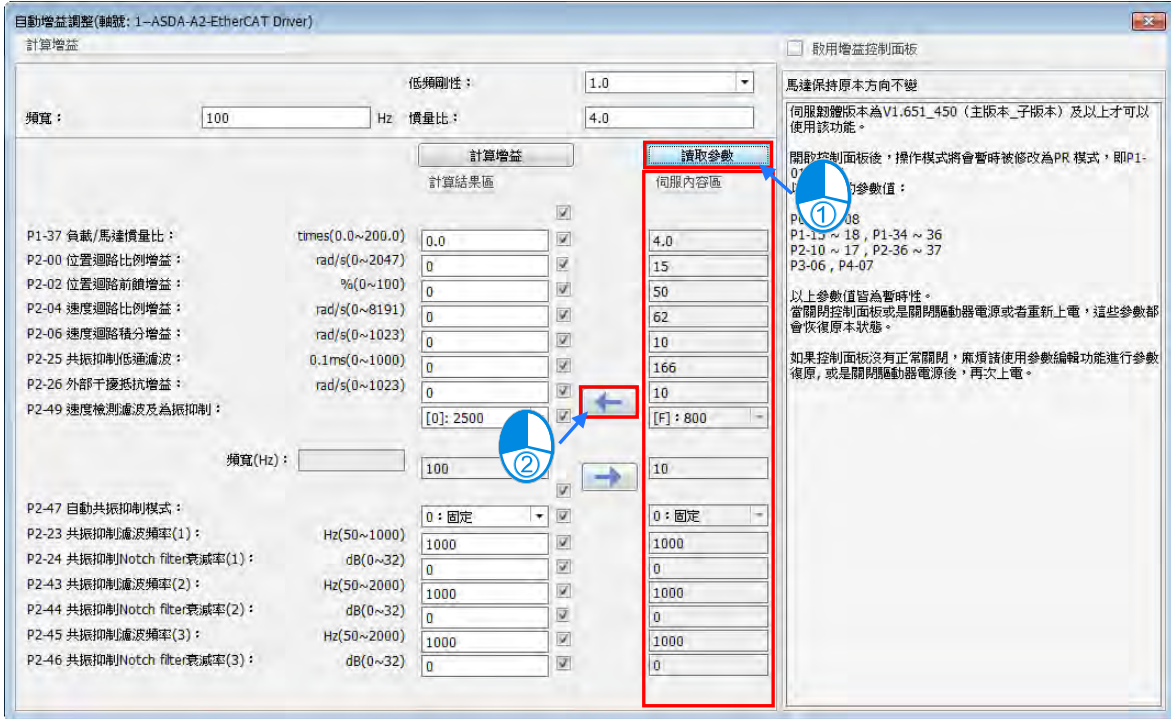


● 計算增益功能介面

計算增益：使用者可在下圖紅色方框處輸入適當的低頻剛性、頻寬和慣量比後，按下「計算增益」按鈕，即可算出相關的增益參數值。再按下“”按鈕，可以將計算得到的結果寫入到伺服中。寫入到伺服時，可勾選是否下載該參數，有勾選表示下載該參數。

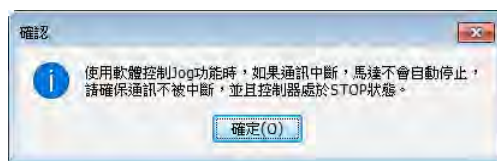


讀取參數：按下「讀取參數」按鈕後，會讀取伺服中參數的現在值並顯示在介面的“伺服內容區”中，以供使用者檢查是否合適。參數讀取成功後，使用者可以使用“←”按鈕，將“伺服內容區”中的值寫入“計算結果區”，然後修改相關參數後再寫入到伺服。

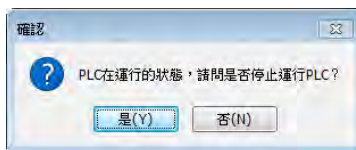


● 增益控制板操作

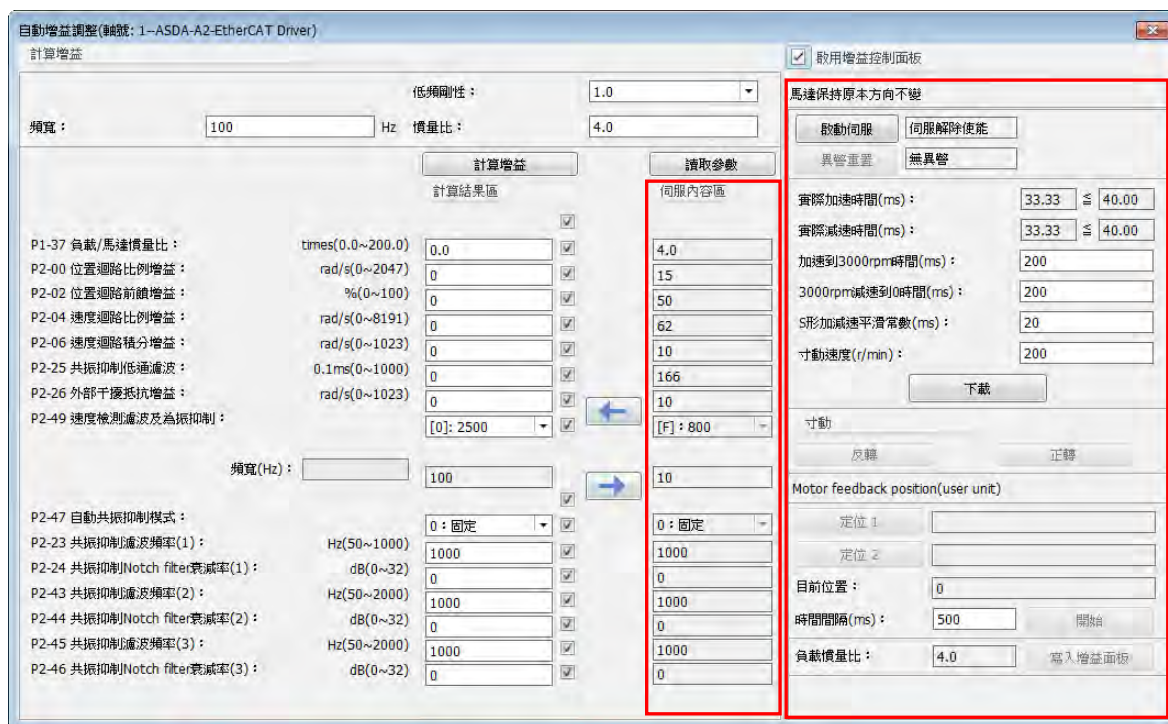
使用增益控制板前，需要使用者勾選右側「增益控制板操作」中的「啟用增益控制面板」，勾選之後軟體會提示相關資訊如下圖所示，提醒使用者在使用控制伺服點動過程中，保持通訊正常。



點擊「確定」後，若控制器處於運行(RUN)狀態，則軟體會快顯視窗讓控制器由運行(RUN)狀態切換到停止 (STOP) 狀態，如下圖所示。




點擊上圖中「是」按鈕後，軟體會打開增益控制板，同時自動讀取參數並填入「伺服內容區」。啟動增益控制板時，軟體會控制伺服進入 Pr 模式，伺服參數 P1-01、P0-10~12、P0-18~20、P1-15~18、P5-20~21、P1-36、P2-10~17、P5-60、P5-61、P3-06、P4-07 會暫時被修改，其中 P1-01 的輸出方向設定不會被修改。當關閉增益控制板時，這些參數會恢復啟動增益控制板前的設定值。



該介面中相關功能介紹如下（詳細設定請參考產品手冊）：

1. 啟動伺服：按下「啟動伺服」以啟動伺服使能，若此時伺服有異警發生，則無法繼續操作，必須先排除異警後，才可以繼續操作。
2. 設定點動相關參數：設定加減速時間，S 形加減速平滑常數與點動速度。點動速度代表調整時電機轉動速度。設定完成後，點擊「下載」按鈕，將設定寫入伺服，或者在按下點動按鈕時，這些參數也會寫入伺服。
3. Jog：依方向不同選擇正向或者反向點動按鈕。按住按鈕時，馬達運轉，鬆開按鈕時，馬達停止。馬達回授位置會顯示在“目前位置”一欄中。使用者可以在正向點動完成後，點擊「定位 1」按鈕，可以將馬達回授位置顯示到「定位 1」一欄中，在反向點動完成後，點擊「定位 2」按鈕，將馬達回授位置顯示到「定位 2」一欄中。
4. 定位：點擊「定位 1」或「定位 2」按鈕，軟體會讀取當前的馬達回授位置值，並顯示到「定位 1」或「定位 2」一欄中。
5. 時間間隔：定位 1 和定位 2 執行完後所需等待的時間，如定位 1 執行完後等待“時間間隔”設定的時間後再執行定位 2，定位 2 執行完後等待“時間間隔”設定的時間後再執行定位。

6. 開始調試：點擊「**開始**」按鈕，開始執行調試。開始調試後，「**開始**」按鈕會變為「**停止**」按鈕，調試過程中，定位 1 和定位 2 交替執行，觀察負載慣量比的變化，當趨近於穩定狀態（變動幅度不大），使用者可以按下“停止”按鈕停止馬達動作。
7. 負載慣量比：參考第 6 點說明部分。
8. 寫入增益面板：按下「**寫入增益面板**」，可以將估測好的慣量數值寫入計算增益功能介面的「**計算結果區**」中。當使用者設定好相關參數後，可以點擊“”按鈕，將參數寫入伺服。
9. 關閉增益控制板：結束動態計算後，請使用者先按「**啟動伺服**」按鈕，將伺服設為“**Servo Off**”狀態，在取消勾選“**啟動增益控制板**”。

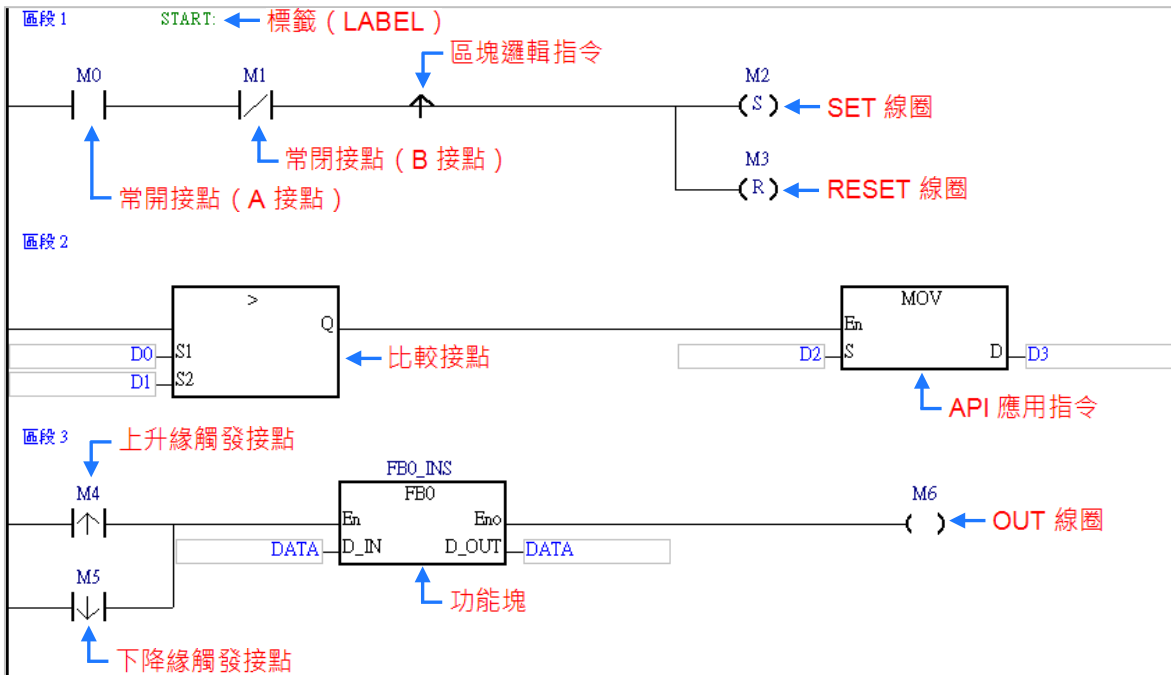
第10章 階梯圖 LD

目錄

10.1 階梯圖 (LD) 簡介	10-2
10.1.1 ISPSOft 的 LD 編輯環境	10-2
10.1.2 階梯圖區段 (NETWORK)	10-4
10.1.3 選取物件	10-5
10.2 於 ISPSOft 中建立 LD 程式	10-6
10.2.1 建立接點 (Contact)、線圈 (Coil) 及 MPS	10-6
10.2.1.1 插入接點 (Contact) 與變更接點型態	10-6
10.2.1.2 插入線圈 (Coil) 與變更線圈型態	10-8
10.2.1.3 新增 MPS	10-9
10.2.2 於 LD 中使用裝置、變數符號及常數	10-10
10.2.3 指令編輯模式	10-11
10.2.4 插入 API 應用指令與功能塊	10-12
10.2.5 編輯比較接點	10-16
10.2.6 編輯區塊邏輯指令	10-19
10.2.7 編輯多重輸出	10-21
10.2.8 設置標籤 (Label)	10-22
10.2.9 編輯註解	10-23
10.2.10 符號模式與位址模式的顯示切換	10-25
10.2.11 書籤功能	10-26
10.2.12 階梯圖區段的致能與失效	10-28

10.1 階梯圖 (LD) 簡介

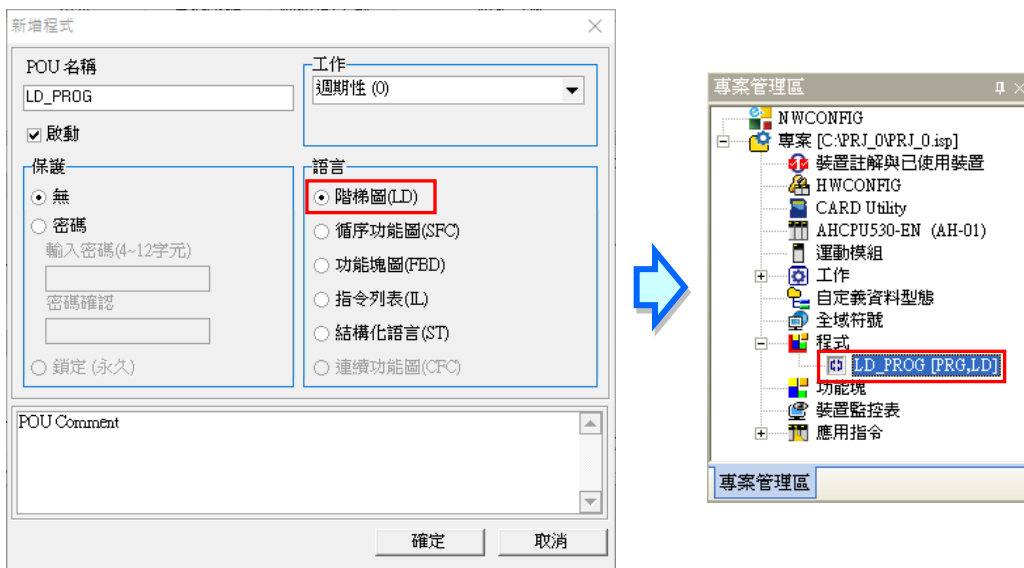
階梯圖 (Ladder Diagram · LD) 為 IEC61131-3 所規範的 PLC 編程語言之一，也是最被廣泛使用的 PLC 開發工具，其特色在於以控制迴路圖來表達程序的邏輯。下圖即為 ISPSOft 中的階梯圖程式。



*.關於階梯圖的編程概念請參考台達 DVP 機種的應用技術手冊 (程式篇)，本章將不再多加說明。

10.1.1 ISPSOft 的 LD 編輯環境

在建立 POU 時，於語言欄位選擇「階梯圖 (LD)」便可建立階梯圖程式，請參考第 5.4.1 節的說明。



下圖即為 ISPSOft 當中的階梯圖 (LD) 編輯環境，視窗的上方為區域符號表，而下方則是編輯工作區。



進入 LD 的編輯環境後，ISPSOft 的視窗中便會出現對應的編程圖示工具列，其功能簡介如下。



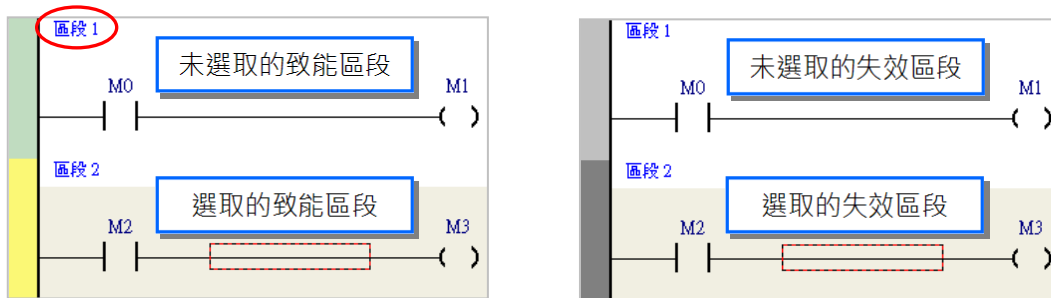
圖示	鍵盤 (快捷鍵)	功能說明
	無	切換顯示符號或位址
	Shift + Ctrl + C	顯示/隱藏區段註解
	無	顯示/隱藏裝置註解與提示
	Shift + Ctrl + B	在目前選取的階梯圖區段上加入或取消書籤標記
	Shift + Ctrl + P	移至前一個書籤標記的位置
	Shift + Ctrl + N	移至下一個書籤標記的位置
	Ctrl + I	於目前選取的階梯圖區段之前插入一個新區段
	Shift + Ctrl + I	於目前選取的階梯圖區段之後新增一個新區段
	ESC	切換為選取工具
	指令編輯模式	插入接點裝置
	指令編輯模式	插入線圈裝置
	指令編輯模式	新增 MPS
	指令編輯模式	插入比較接點
	指令編輯模式	選擇比較接點的類型
	指令編輯模式	插入區塊邏輯指令 (NP/PN/INV/FB_NP/FB_PN)
	指令編輯模式	選擇區塊邏輯指令的類型 (NP/PN/INV/FB_NP/FB_PN)
	Shift + Ctrl + U	插入指令或功能塊


*.關於指令編輯模式請參考本章第 10.2.3 節的相關說明。

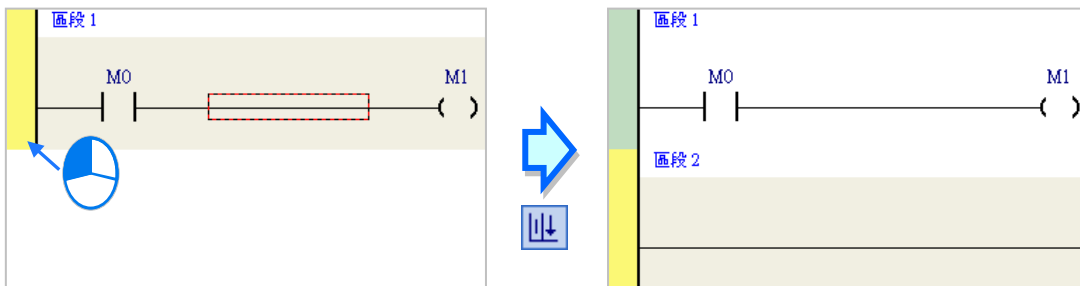
10.1.2 階梯圖區段 (NETWORK)


階梯圖 (LD) 程式是以區段 (NETWORK) 為編輯單位，每個階梯圖區段都是一個獨立的運算程序，而一個階梯圖程式亦是由許多的區段所建構而成。除此之外，在 ISPSOft 中，每個階梯圖區段可插入的元件數量並沒有特別的限制，因此不會出現跳行的連結符號。

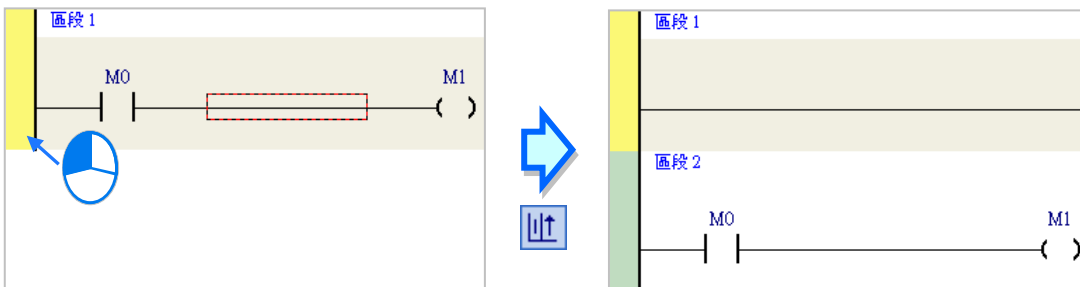
在每個階梯圖區段的左上角都會標示該區段的區段編號，且由區段左側的顏色便可得知該區段目前的狀態，其狀態的顯示依功能可分為**致能與失效**，依編輯狀態則可分為**選取與未選取**，而所顯示的顏色則可於 ISPSOft 的「**選項**」設定中由使用者自行定義，詳細操作請參考第 2.3.1 節。




當新增一個階梯圖程式時，系統會預先建立一個空白的階梯圖區段，而要新增階梯圖區段時，請先選取某個現有區段後，於圖示工具列中點擊  圖示即可於選取的階梯圖區段下方新增一空白區段。

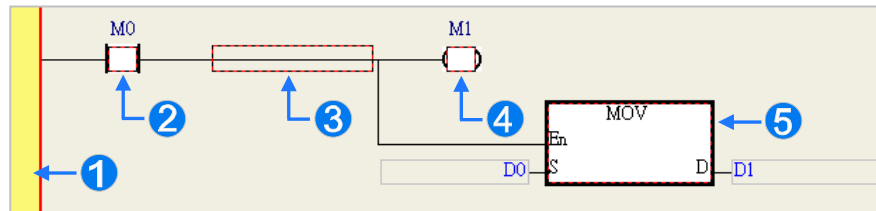


而於圖示工具列中點擊  圖示則可於選取的階梯圖區段上方插入一空白區段。



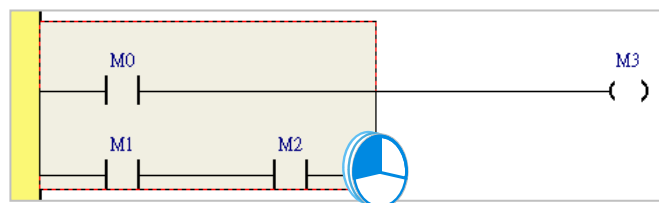
10.1.3 選取物件

要於階梯圖區段中進行選取時，請先按下鍵盤的【ESC】鍵，或於圖示工具列中點擊  圖示，待鼠標切換為選取狀態後，直接於欲選取的物件上點擊滑鼠左鍵即可。下列即為基本的選取操作。

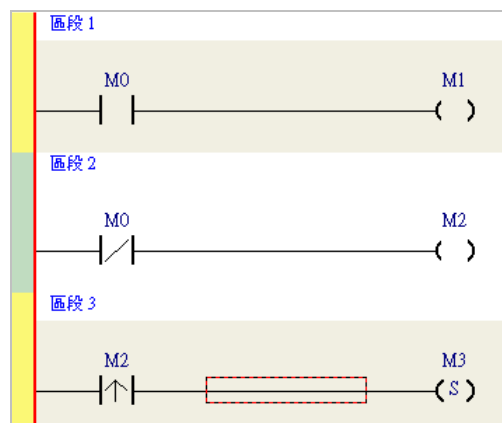


- ❶ 選取整個區段。
- ❷ 選取輸入接點。
- ❸ 選取整個區段。
- ❹ 選取輸出線圈。
- ❺ 選取方塊。

要選取區塊時，可先點選任一裝置後按住滑鼠左鍵，接著拖曳滑鼠便可將區塊框選起來；或是先點選區塊範圍的第一個物件後，按下鍵盤的【Ctrl】+【B】，接著再點選區塊範圍的最後一個物件，並再次按下【Ctrl】+【B】鍵便可將其框選；而框選時，該區塊必須位於同一個階梯圖區段中，且必須是連續位置的物件，而階梯圖區段中的輸入裝置與輸出裝置則無法被框選為同一個區塊。



當要同時選取多個階梯圖區段時，請先按住鍵盤的【Ctrl】鍵，接著便可用滑鼠分別點選多個不連續位置的區段；另外也可先按住【Shift】鍵後，再分別用滑鼠點選欲選取範圍的第一個與最後一個階梯圖區段，之後便可將整個範圍內的區段全部選取起來。



10.2 於 ISPSOft 中建立 LD 程式

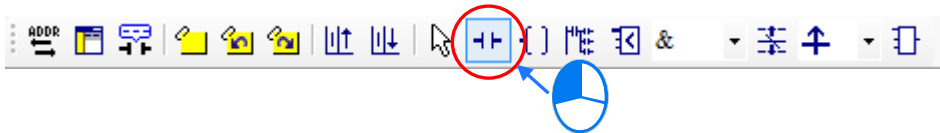
10.2.1 建立接點 (Contact)、線圈 (Coil) 及 MPS

下表即為 ISPSOft 中各種接點與線圈型態的圖示。

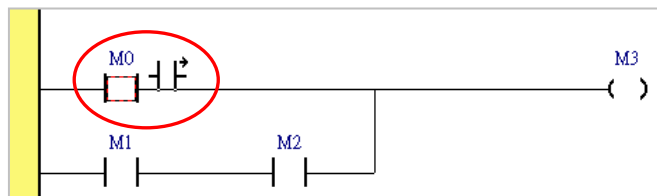
接點				線圈		
常開接點	常閉接點	上升緣觸發	下降緣觸發	OUT	SET	RESET

10.2.1.1 插入接點 (Contact) 與變更接點型態

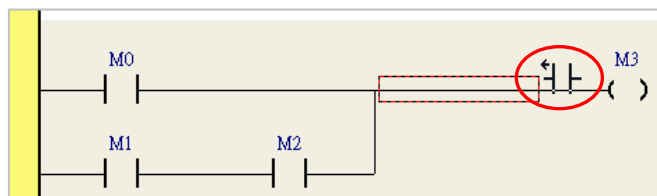
- 首先請選擇欲插入接點 (Contact) 的位置，該位置可為線路、接點或區塊，而所選的位置也決定了之後的串並聯結構。當要與某個區塊做串並聯時，在執行前請先將該區塊框選起來。確定插入位置之後，請於圖示工具列中點擊 圖示以將鼠標切換為接點工具。



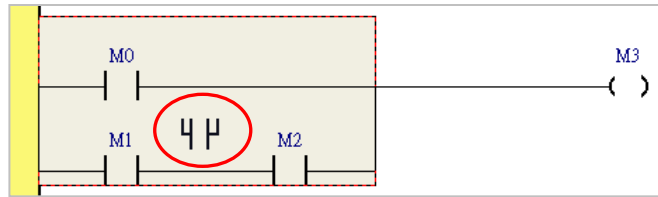
- 插入接點時，必須先將鼠標靠近欲插入的位置，而此時滑鼠游標便會變為接點的圖示。
 - 當鼠標靠近某個接點時，便代表要與該接點進行串並聯的動作。



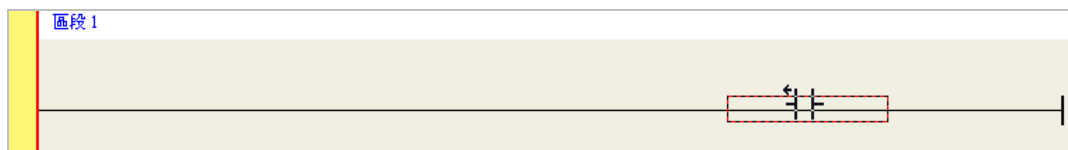
- 當鼠標靠近線路時，便代表要在該線路上插入接點。



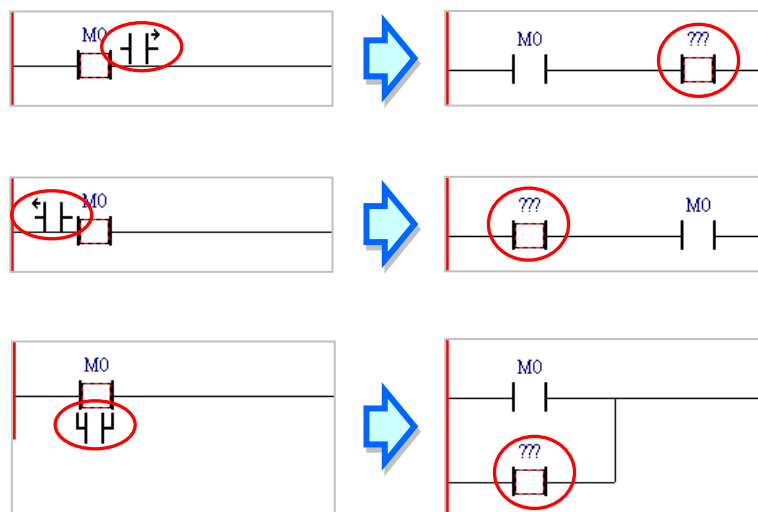
- 當鼠標靠近區塊時，便代表要與該區塊進行串並聯。




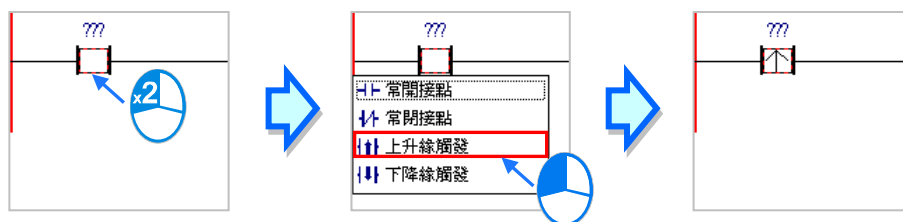
- 當要在新區段中插入接點時，請將切換為接點工具的鼠標靠近如下圖線路的紅框處即可。




- (3) 在欲插入的位置上刻意將滑鼠往左右或下方偏移時，鼠標的圖示也會跟著變化，而依據鼠標的圖示便可決定新接點的插入方式。在確定插入的方式之後按下滑鼠左鍵，接著系統便會插入一個接點，並將其自動排列在適當的位置上。



- (4) 插入接點後，點選  圖示或按下鍵盤的【ESC】鍵將滑鼠游標切回選取狀態，接著於該接點的圖示上雙擊滑鼠左鍵，之後便可於下拉選單中選擇接點的類型。



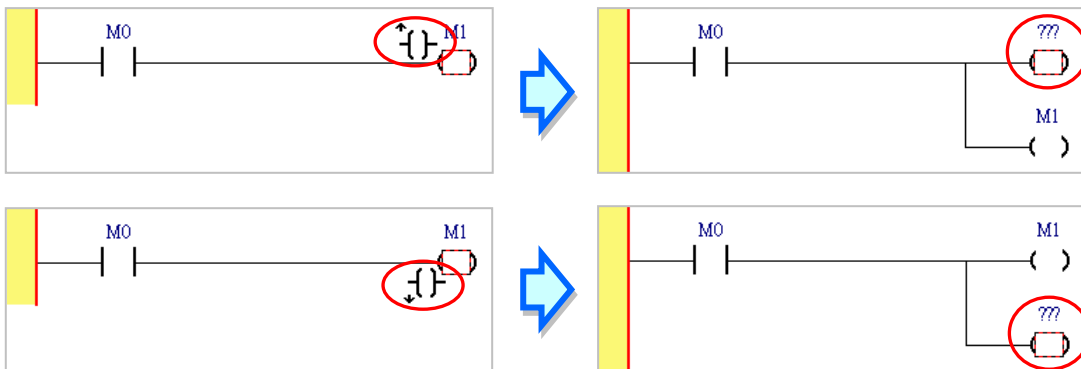
10.2.1.2 插入線圈 (Coil) 與變更線圈型態

- (1) 首先請選擇欲插入線圈 (Coil) 的位置，該位置可為線路、線圈或 API 應用指令。確認插入位置後，請於圖示工具列中點擊  圖示以將鼠標切換為線圈工具。

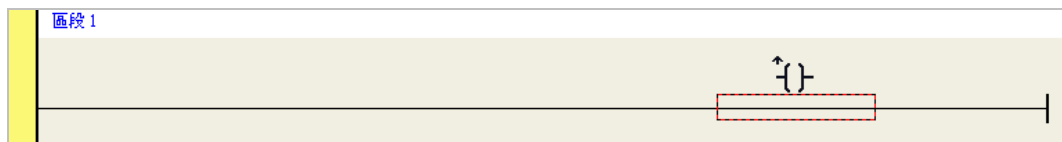



- (2) 插入線圈時，必須先將鼠標靠近欲插入的位置，而此時滑鼠游標便會變為線圈的圖示；當鼠標靠近某個線圈或 API 應用指令，並將鼠標往該物件的上方或下方偏移時，便代表線圈將插於該物件的上方或下方；而當鼠標靠近的位置是區段的線路上時，插入線圈之後，新線圈只會被排列在所有輸出物件的最上方。

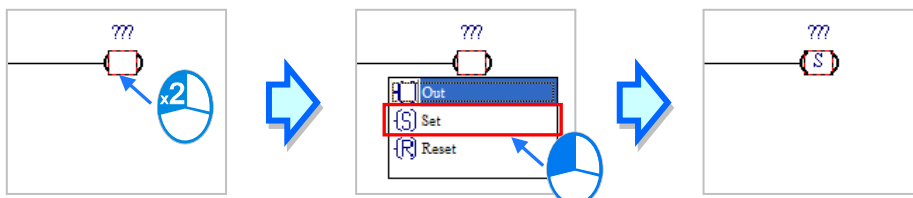
確定插入位置後按下滑鼠左鍵，接著系統便會插入一個線圈，並將其自動排列在適當的位置上。




當區段中沒有任何輸出物件時，則必須將鼠標靠近選取線路的位置才可插入線圈。

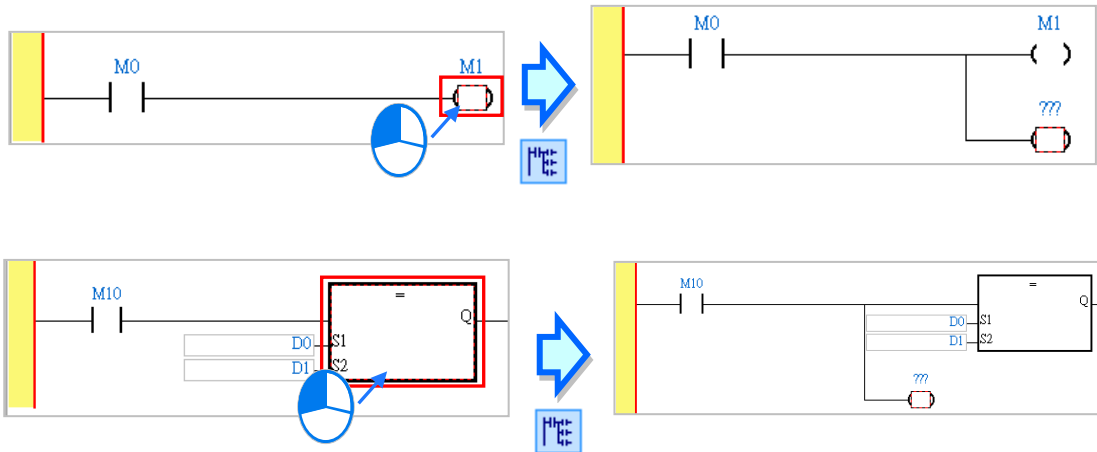


- (3) 插入線圈之後，點選  圖示或按下鍵盤的【ESC】鍵將滑鼠游標切回選取狀態，接著於該線圈的圖示上雙擊滑鼠左鍵，之後便可於下拉選單中選擇線圈的類型。

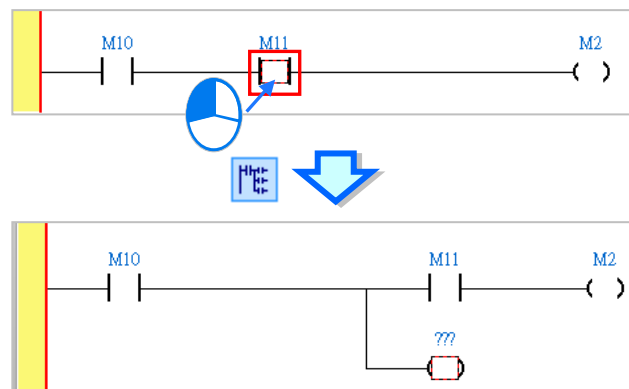


10.2.1.3 新增 MPS

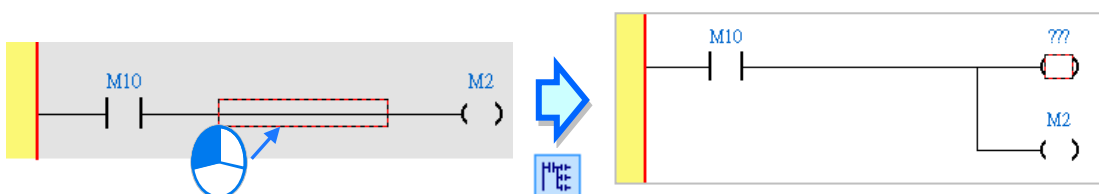
- (1) 首先請選擇欲新增 MPS 的位置，該位置可為線路、線圈、比較或 API 應用指令。確認新增位置後，請於圖示工具列中點擊  圖示，即可新增 MPS，新增 MPS 功能為新增一個分歧點。
- (2) 使用者可點選線圈或比較指令，接著點擊新增 MPS 鈕，即可新增一個分歧點。



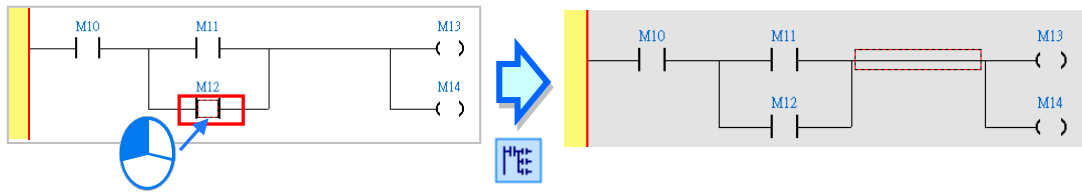
- (3) 使用者可點選接點，接著點擊新增 MPS 鈕，即可新增一個分歧點。



- (4) 使用者也可點選線路，接著點擊新增 MPS 鈕，即可新增一個分歧點。

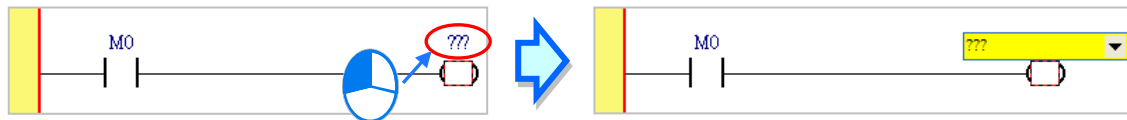


(5) 某些位置點無法新增 MPS，若點擊新增 MPS 鈕，則會自動跳至一個可新增 MPS 的位置點。

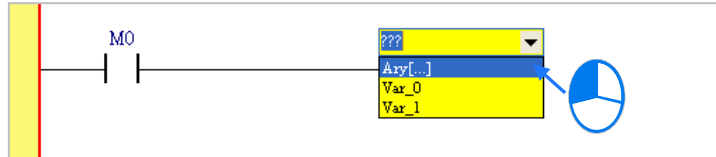


10.2.2 於 LD 中使用裝置、變數符號及常數

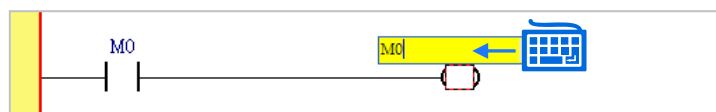
欲於插入的物件中編輯運算元時，請直接於運算元的位置上點擊滑鼠左鍵，之後該運算元便會呈現編輯狀態，接著再依據下述方式進行操作即可。



直接點擊編輯欄位右側的箭頭按鈕，接著便可於下拉選單中選擇已宣告好的變數符號；而當有宣告陣列符號時，於下拉選單中可挑選該陣列，如下圖的 `Ary[...]`，再於 `[]` 中輸入陣列的元素，多維陣列的輸入請見下述說明。



另外亦可直接於編輯欄位以鍵盤輸入裝置名稱或是變數符號，而當輸入未經宣告的變數符號時，輸入完畢後按下鍵盤【Enter】鍵，系統預設會自動導出符號宣告視窗。(設定方式請參考第 2.3.1 節)



當用滑鼠點擊某個區段後按下【Enter】鍵，接著系統便會自動開啟編輯欄位，而利用鍵盤的【Enter】鍵便可在同一個區段中反覆移動編輯欄位，利用【Tab】鍵則可跨區段移動，但到了最後一個編輯欄位後便會停止；編輯時，使用者可直接於編輯欄位中使用鍵盤輸入，該欄位的下拉選單便會隨之展開並過濾出符合的變數符號，利用鍵盤的上下鍵便可進行選擇；完成編輯後，按下【Enter】鍵便會自動再切換至該區段中的下一個編輯位置，而要結束編輯時，只要按下【Esc】鍵即可。

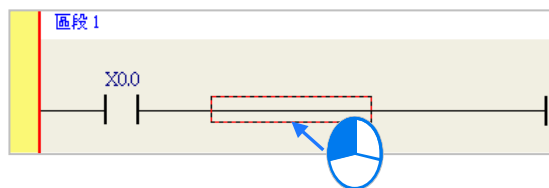
補充說明

- (a) 要在 ISPSOft 的階梯圖 (LD) 中使用常數時，其格式的範例如下。
- 10 進制：23456 (未加任何標註的數字會被視為 10 進制)
 - 16 進制：16#5BA0 (數字前方加上 16#)
 - 8 進制：8#55640 (數字前方加上 8#)
 - 2 進制：2#101101110100000 (數字前方加上 2#)
 - 字串：“XYZ” (以雙引號包住字串)；DVPxxMC/AS5xx 機種是用單引號包住字串如，'XYZ'。
 - BOOL：須使用常開/常閉接點 SM400/SM401 (AH/AS) 或 M1000/M1001 (DVP) 來表達。
- (b) 使用陣列時，其輸入格式為 **符號名稱[索引]**，如上圖下拉選單中的 Ary[...]，索引值可使用小於陣列大小的 10 進制常數，且必須由 0 開始；陣列最高支援三維陣列，以 “,” 分隔，例如 Ary[1,2,3]，可用於搭配自定義資料型態使用，請參考第 8 章。

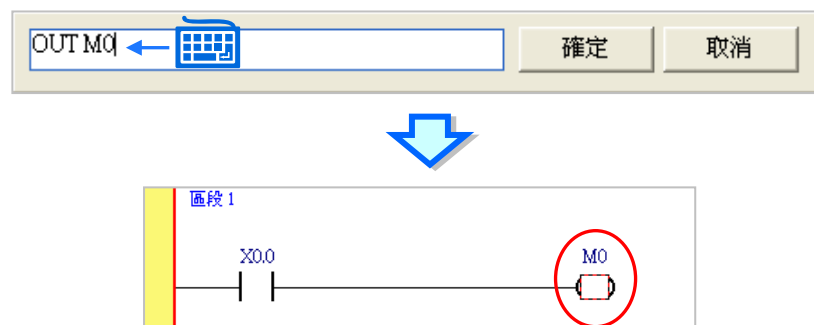
10.2.3 指令編輯模式

在 ISPSOft 的階梯圖中，亦提供以直接輸入指令的方式來新增裝置、API 指令或功能塊。

- (1) 首先於編輯工作區中點選欲插入物件的位置。



- (2) 接著直接用鍵盤輸入指令，而一按下鍵盤之後，畫面便會自動出現編輯視窗，待輸入完畢後按下鍵盤的【Enter】鍵或以滑鼠點擊視窗的「確定」鈕即可。(輸入的指令無須分大小寫)



針對接點與線圈，系統亦提供簡易輸入的功能，請參考下圖。(輸入的指令無須分大小寫)

- 插入常開接點 (A 接點) 指令 : " A 裝置位址 "

- 插入常閉接點 (B 接點) 指令 : " B 裝置位址 "

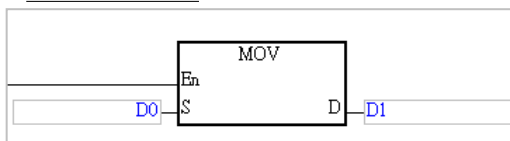
- 插入輸出線圈 (OUT) 指令 : " O 裝置位址 "

10.2.4 插入 API 應用指令與功能塊

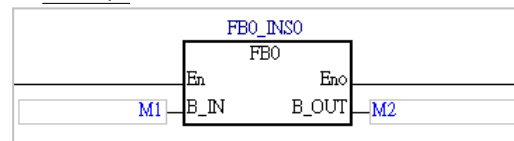
在 ISPSoft 的階梯圖中，API 應用指令與功能塊是以方塊的圖示來呈現，而在方塊中除了對應指令或功能塊的運算元端點之外，尚包含一一致能端點 **En**，該端點的功能在於連結前方的邏輯狀態，當前方的邏輯狀態為 **ON** 時，該 API 指令或功能塊才會被執行。

不過兩者之間不同的是，功能塊的輸出會包含一個 **Eno** 端點，而 API 指令則沒有，因此 API 指令在階梯圖中只能被加在階梯圖區段的末端，而無法如同功能塊一般於後方串接其他元件；此外，在功能塊圖示的上方還會有一個**功能塊實例**的欄位，而關於功能塊的詳細介紹請參考第 7 章的內容。

● API 應用指令



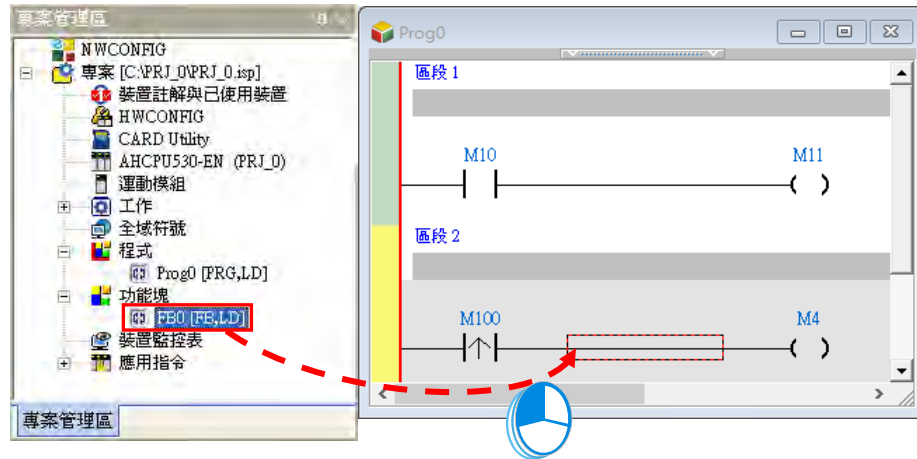
● 功能塊



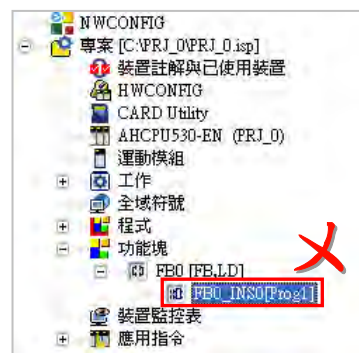
於 ISPSoft 的 LD 編輯環境中，針對插入 API 應用指令與功能塊的操作提供了多種方式，說明如下。

- 方法一


於專案管理區中展開應用指令或功能塊的目錄，在其中找到欲插入的項目後，於該項目上按住滑鼠左鍵，之後再將其拖曳至欲插入的位置後放開左鍵即可。

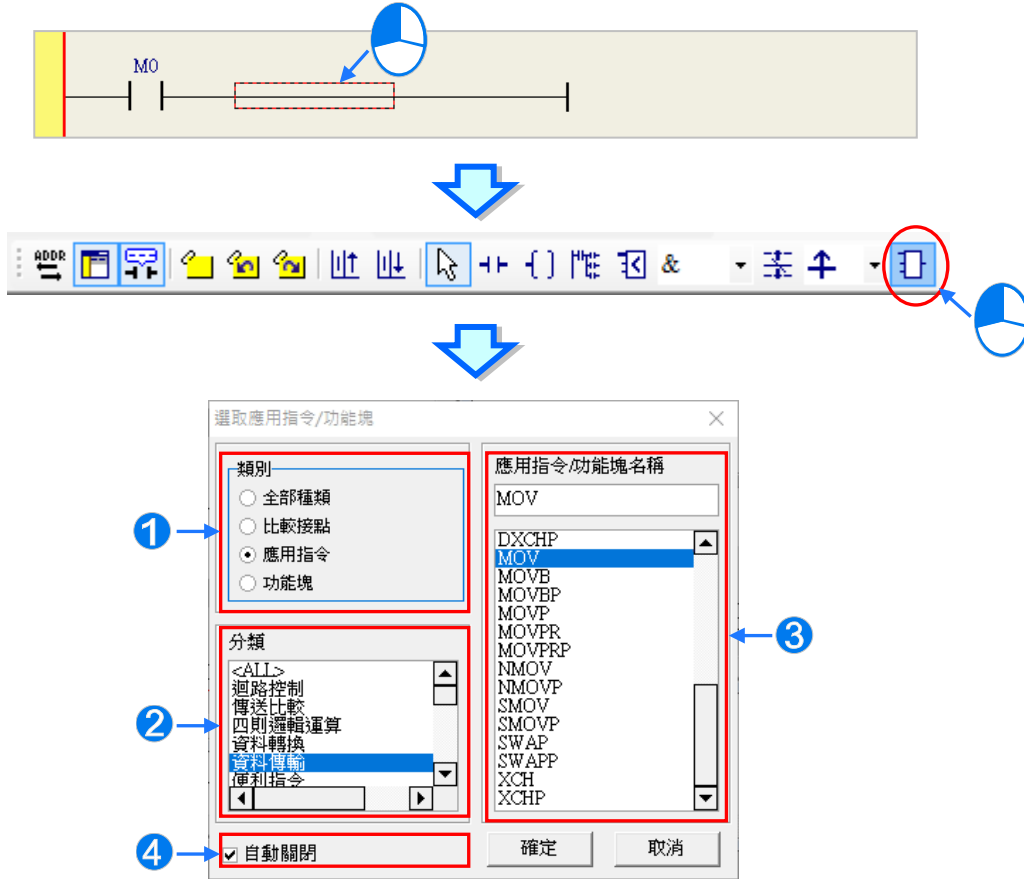


拖曳功能塊時須注意只能拖曳功能塊的定義，而不能拖曳其下層的功能塊實例項目。



● 方法二

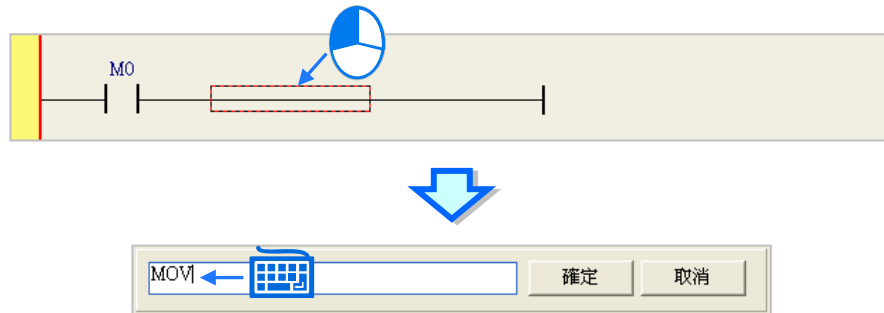
先於欲插入 API 應用指令或功能塊的位置上點擊滑鼠左鍵，之後於圖示工具列中點選  圖示，接著再於出現的選取視窗中點選要加入的項目，完成後按下「確定」鍵即可將其插入指定位置。



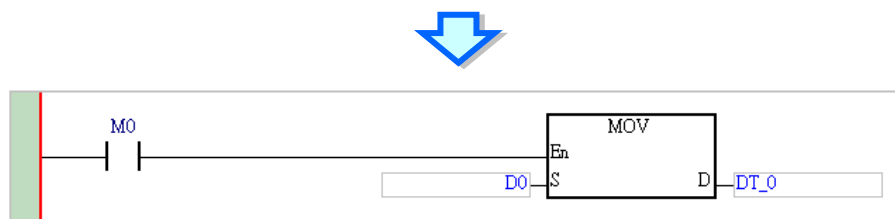
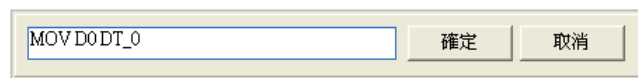
- ❶ 先選取欲加入的類別。
- ❷ 若欲加入的是 API 應用指令時，請再於此處選擇指令分類。
- ❸ 於此選取 API 應用指令、比較接點或功能塊。
- ❹ 若取消此處的勾選，之後便可利用此視窗連續加入指令或功能塊。

- 方法三

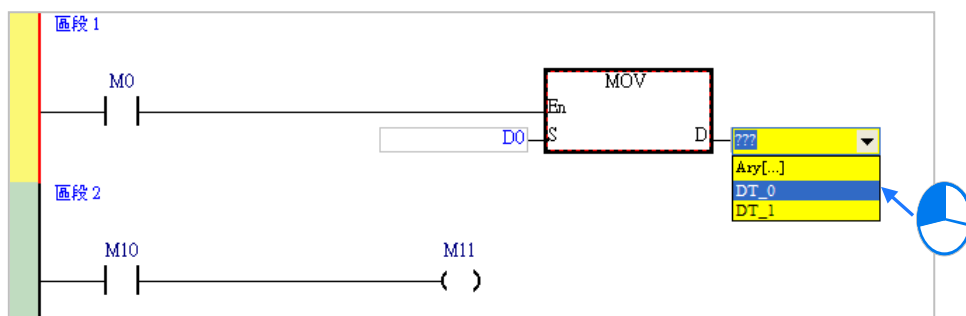
先於欲插入 API 應用指令或功能塊的位置上點擊滑鼠左鍵，之後直接以鍵盤輸入功能塊定義或 API 應用指令的名稱，此時畫面便會出現指令編輯模式的輸入欄位，輸入完畢後按下鍵盤的【Enter】鍵或以滑鼠點擊視窗的「確定」鈕即可。(輸入的指令或功能塊的名稱無須分大小寫)



直接以指令編輯模式輸入時，若輸入的是功能塊，則輸入時只可輸入功能塊定義的名稱，而不可加上功能塊端點的運算元；但若輸入的是 API 應用指令，則輸入時可依照該指令的 IL 語法一併輸入指令的運算元，且輸入的運算元數量無須完整，系統將會依序將其插入對應的運算元欄位，並讓缺少的運算元欄位維持「???'的顯示；若輸入的運算元包含未經宣告的變數符號時，確認輸入之後，系統亦會自動導出符號宣告視窗。詳細的指令語法則請參考各機種的指令手冊。

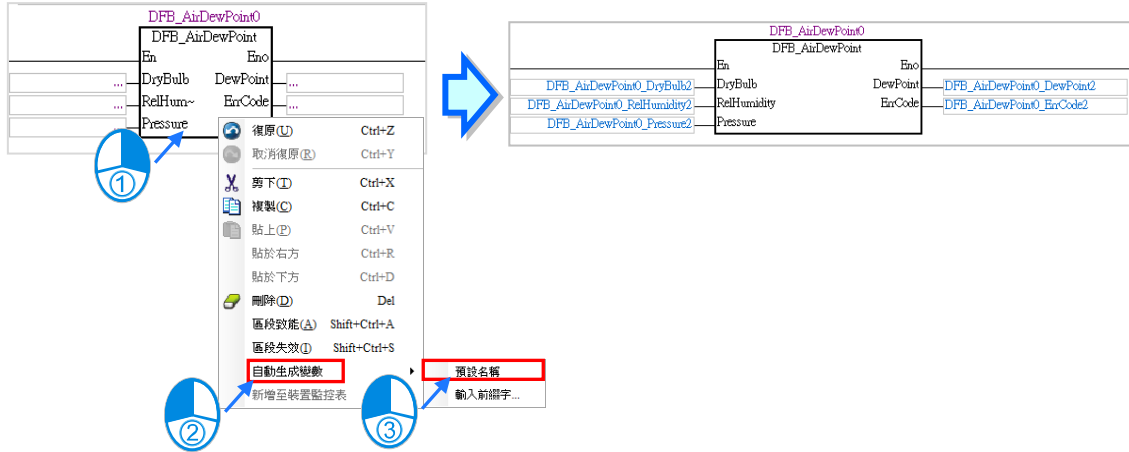


成功插入 API 應用指令或功能塊之後，請再參考第 10.2.2 節的說明，逐一完成每個運算元的編輯。

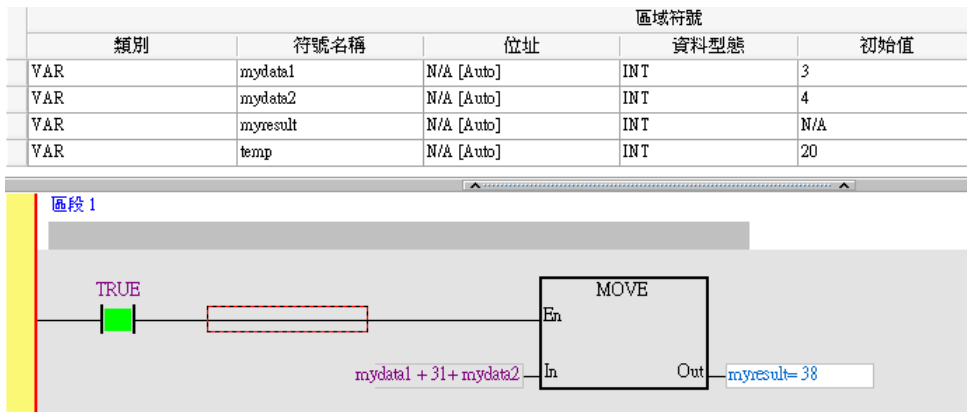


*.自動導出符號宣告視窗的功能可於「選項」當中進行設定，設定方式請參考第 2.3.1 節的說明。

可以在功能塊上方點擊滑鼠右鍵，選取自動生成變數的功能，選擇預設名稱，會以功能塊的名稱+引腳名稱，產生變數並填入引腳位置，如下圖；而選擇輸入前綴字，則是會以輸入的文字，產生變數並填入引腳位置。

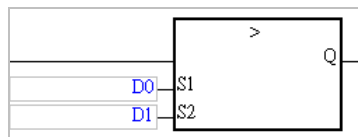


在 DVPxxMC 與 AS5xx 機種的功能塊引腳支援 ST 運算式的寫法，如下圖。

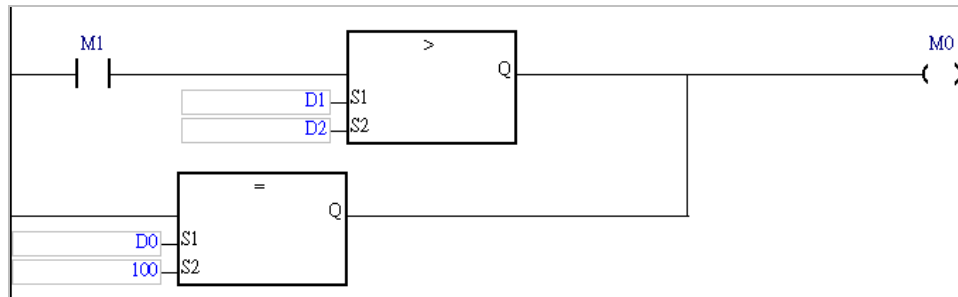


10.2.5 編輯比較接點


下圖是 ISPSOft 中的比較接點圖示，方塊圖示上方的「>」則是比較的類型，「S1」與「S2」端點則為比較的對象，其接受的資料格式決定於比較的類型，而輸出「Q」則可直接趨動線圈或是串聯接點。

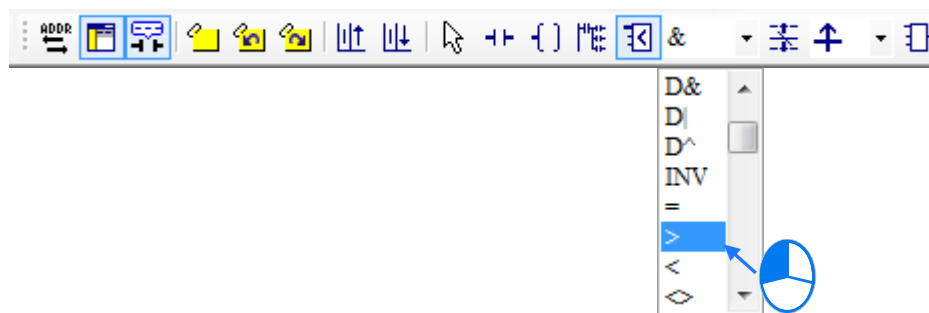



在 LD 的程式中，比較接點可如同一般的接點而與其它接點進行串並聯的邏輯運算。

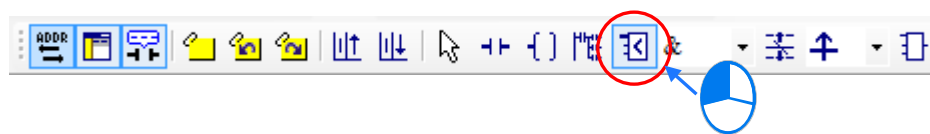


於 LD 程式插入比較接點的方式與插入 API 應用指令相同，操作方式請參考第 10.2.4 節的相關說明。而除此之外，亦可使用如同插入一般接點 (Contact) 的方式來插入比較接點，說明如下。

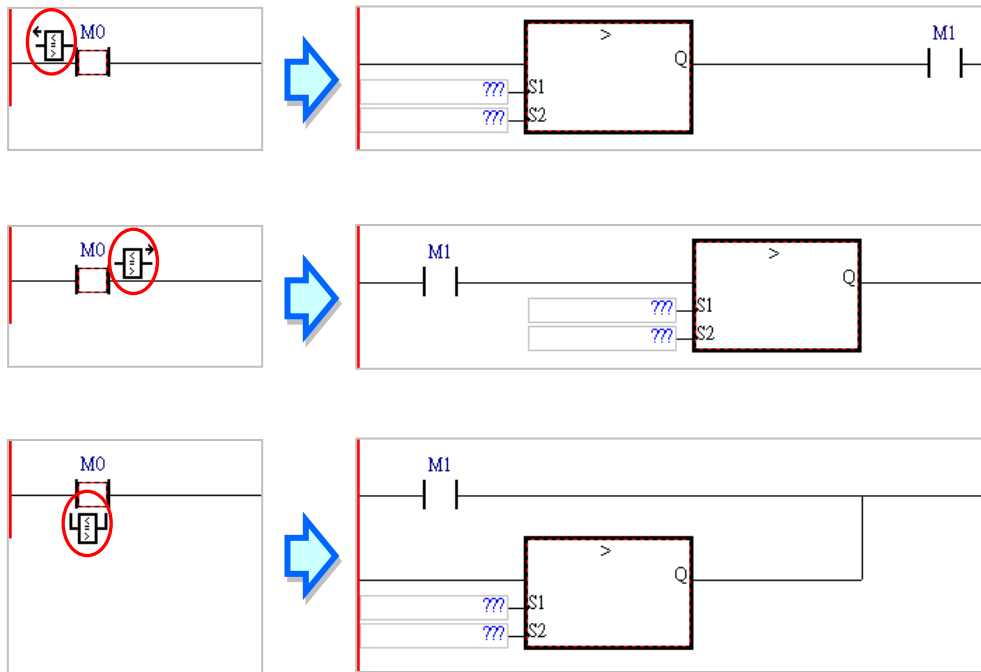
- (1) 首先請於圖示工具列的  欄位下拉選擇比較類型。



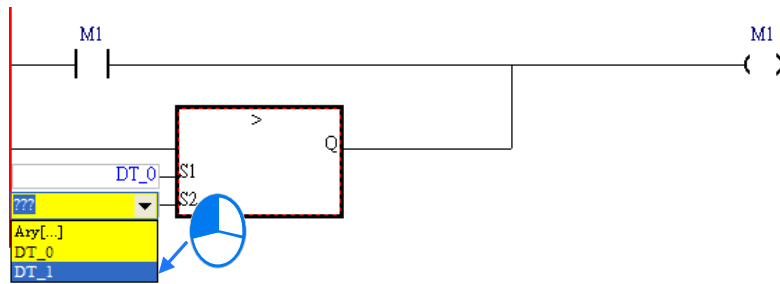
- (2) 接著請選擇欲插入比較接點的位置，該位置可為線路、接點或區塊，而所選的位置也決定了之後的串並聯結構。當要與某個區塊做串並聯時，在執行前請先將該區塊框選起來。確定插入位置之後，請於圖示工具列中點擊  圖示以將鼠標切換為比較接點工具。



- (3) 插入接點時，必須先將鼠標靠近欲插入的位置，而此時滑鼠游標便會變為比較接點的圖示；而在該位置上刻意將滑鼠往左右或下方偏移時，鼠標的圖示也會跟著變化，而依據鼠標的圖示便可決定新接點的插入方式。在確定插入的方式之後按下滑鼠左鍵，接著系統便會插入一個接點，並將其自動排列在適當的位置上。

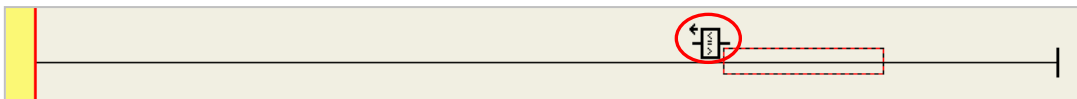


- (4) 成功插入比較指令之後，請再參考第 10.2.2 節的說明，逐一完成每個運算元的編輯。



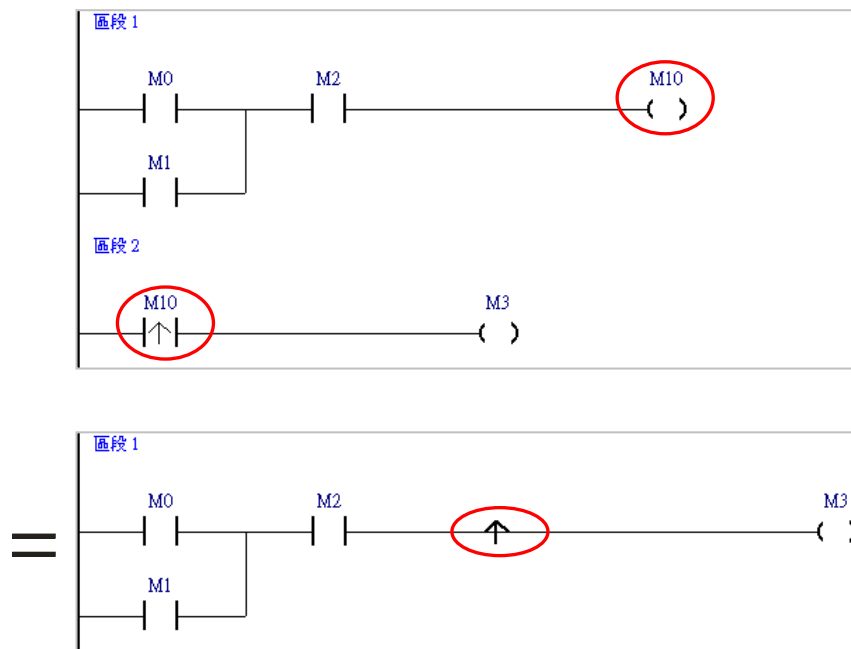
補充說明

當要在新區段中插入比較接點時，必須將切換為比較接點工具的鼠標靠近線路才可將其插入。



10.2.6 編輯區塊邏輯指令

如下圖，當要將某個區塊的邏輯運算結果拿來做上升緣觸發操作時，傳統的編輯方式必須將其分為兩個階梯圖區段，而使用區塊邏輯指令便可將其合併為同一個階梯圖區段。



下表即為 ISPSOft 中所提供的區塊邏輯指令及各機種可以使用指令，在 POU 程式或 FB 功能塊的支援列表。

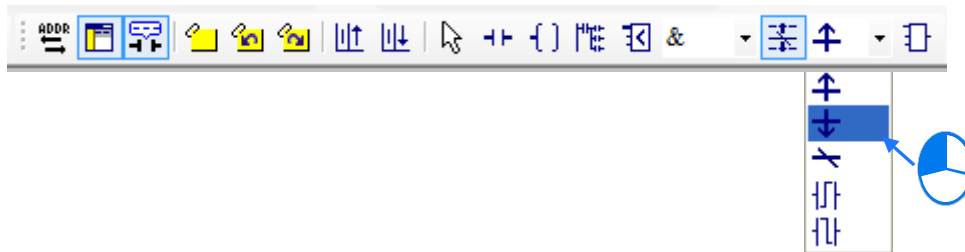
階梯圖 圖示	指令 名稱	功能說明	AH/AS		DVP		AS5xx DVPxxMC	
			POU	FB	POU	FB	POU	FB
	NP	當前方區塊的邏輯狀態由 OFF 變為 ON 時才會輸出 ON。	✓	✓	✓	✓		
	PN	當前方區塊的邏輯狀態由 ON 變為 OFF 時才會輸出 ON。	✓	✓	✓	✓		
	INV	將前方區塊的邏輯狀態反相後輸出。	✓	✓	✓	✓	✓	✓
	FB_NP	功能與 NP 類似，唯須額外指定一暫存裝置或變數。		✓			只支援 ES3	
	FB_PN	功能與 PN 類似，唯須額外指定一暫存裝置或變數。		✓			只支援 ES3	


* 關於 FB_NP 與 FB_PN 指令請查閱各機種程式手冊，或參考第 7.3.2 節當中的相關說明。

下列則為插入**區塊邏輯指令**的方式。

● 方法一

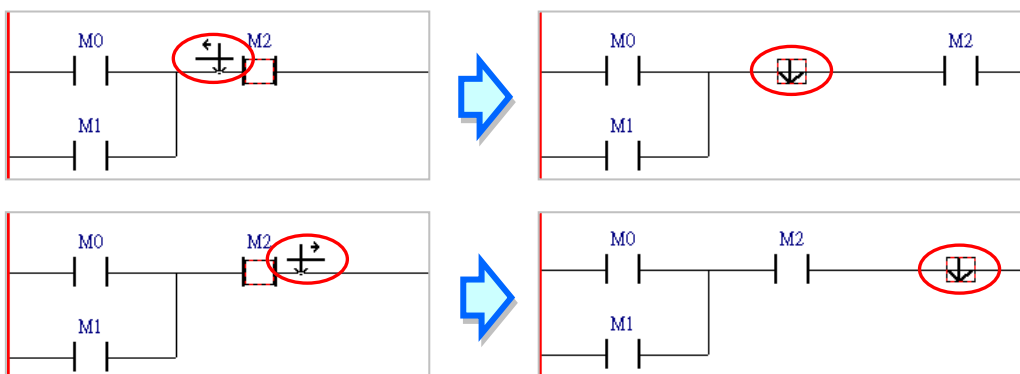
- (1) 請先於圖示工具列的  欄位下拉選擇欲插入的指令類型。



- (2) 點擊工具列圖示  將滑鼠游標切換為區塊邏輯指令的狀態。

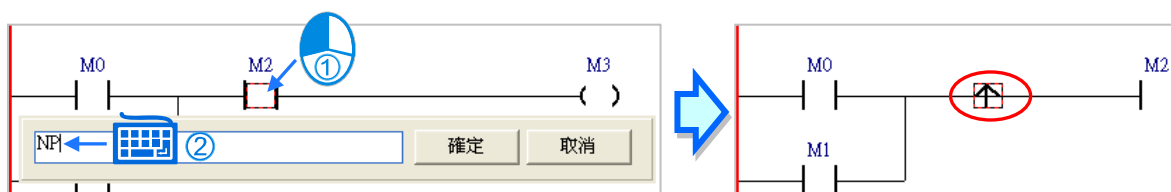


- (3) 當將滑鼠游標靠近欲插入的位置時，滑鼠游標便會變為區塊邏輯指令的圖示，而刻意將鼠標往該位置的左方或右方偏移時，鼠標便會隨之變化以代表插入的方式。在確定插入的方式之後按下滑鼠左鍵，接著系統便會插入一個區塊邏輯指令，並將其自動排列在適當的位置上。



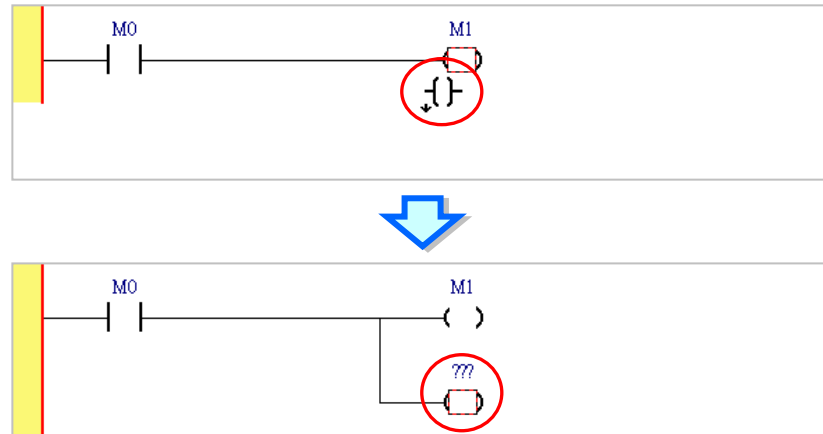
● 方法二

先點選欲插入的位置之後，直接以鍵盤輸入**區塊邏輯指令**的名稱，此時畫面便會出現**指令編輯模式**的輸入欄位，輸入完畢後按下鍵盤的【Enter】鍵或以滑鼠點擊視窗的「確定」鈕，此時系統便會於所選位置的前方插入輸入的指令圖示。(輸入的指令無須分大小寫)

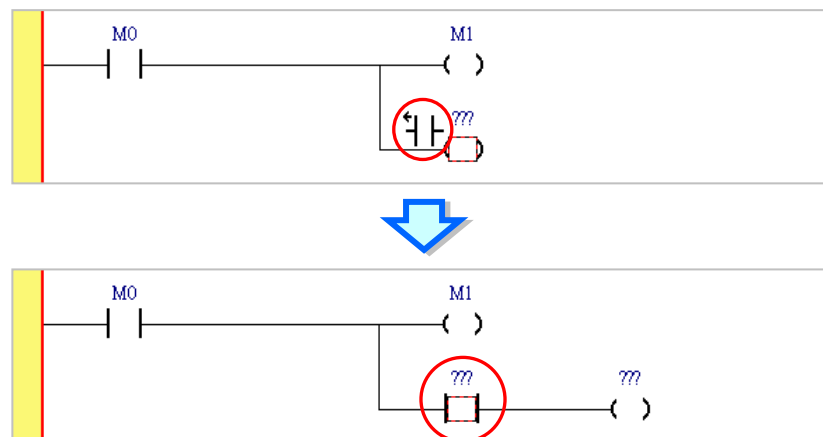


10.2.7 編輯多重輸出

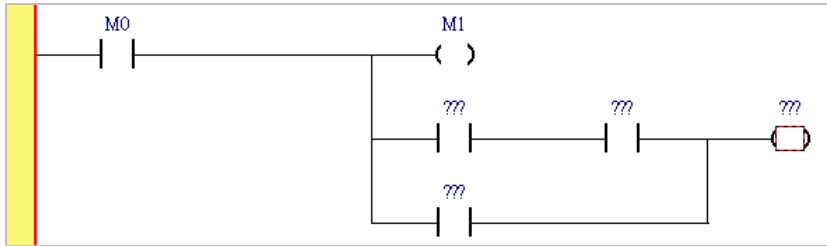
(1) 欲編輯多重輸出時，請先於線路上完成輸出的線圈或 API 應用程式。



(2) 接著請將滑鼠切換為接點或比較接點工具，之後再將鼠標靠近輸出物件後按下滑鼠左鍵。



(3) 接下來便可依據前述的說明逐一完成多重輸出的結構。

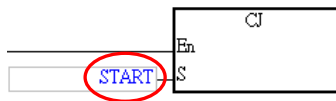


10.2.8 設置標籤 (Label)

當要使用跳躍指令，如 CJ、JMP...時，便必須為跳躍的目標區段設置一個標籤 (Label)，而標籤的位置則位於每個區段編號的右側。



要在指令中使用標籤 (Label) 時，標籤符號無須再加上冒號『 : 』。

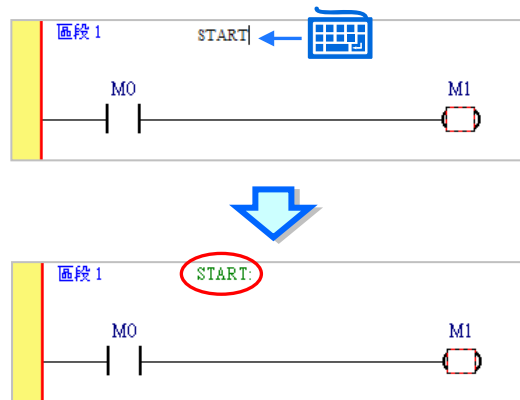


下列則為設置標籤 (Label) 的操作方式。

(1) 在欲設置標籤的區段編號右側點擊滑鼠左鍵。

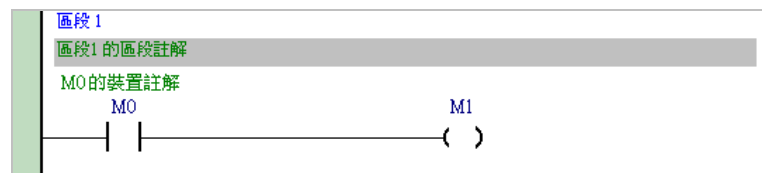


- (2) 請輸入標籤符號名稱；而輸入時，標籤後的冒號『：』可直接省略，系統之後會自行將其補上。完成之後按下【Enter】鍵即可。




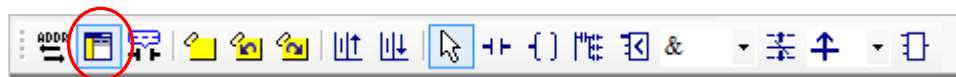
10.2.9 編輯註解

在 ISPSOft 的階梯圖畫面中，除了可直接在程式畫面編輯裝置註解外，使用者亦可替每個階梯圖區段加上區段註解，而區段註解則位於每個階梯圖區段的編號下方。

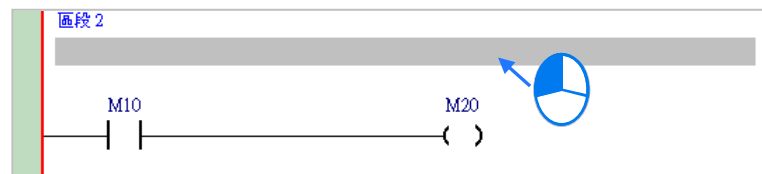


● 編輯區段註解

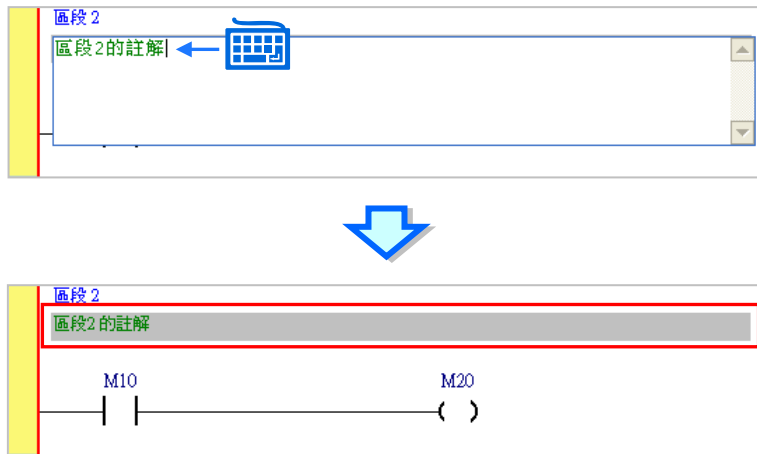
- (1) 當階梯圖工具列中的  圖示為按下的狀態時，於每個區段編號的下方會顯示該區段的區段註解；而同樣的，當要隱藏區段註解時，請直接將該圖示恢復為未按下的狀態即可。




- (2) 於區段註解的欄位上點擊滑鼠左鍵，之後便會出現區段註解的輸入欄位。



- (3) 接著請直接在出現的編輯欄位中編輯註解，其中按下鍵盤的【Shift】+【Enter】即可換行輸入；輸入完畢後，直接按下鍵盤的【Enter】鍵或用滑鼠點擊他處即可。

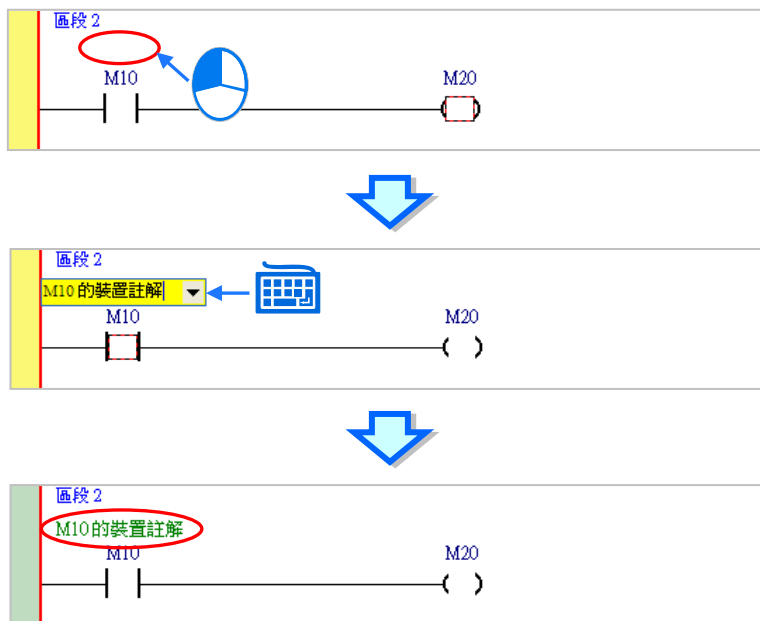


● 編輯裝置註解


- (1) 當階梯圖工具列中的  圖示為按下的狀態時，在每個裝置編號的上方便會顯示裝置註解；而同樣的，當要隱藏裝置註解時，請直接將該圖示恢復為未按下的狀態即可。




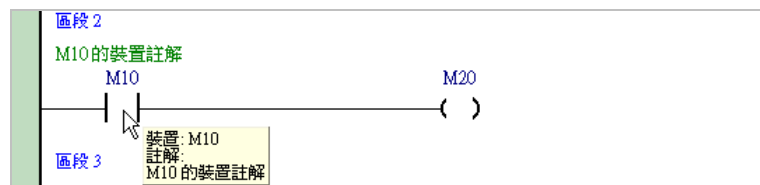
- (2) 於裝置註解的欄位上點擊滑鼠左鍵，接著便可直接於該欄位中編輯裝置註解，也可以對 API 的 PIN 腳編輯註解；輸入完畢後，直接按下鍵盤的【Enter】鍵或用滑鼠點擊他處即可。




補充說明


若階梯圖工具列的  圖示為按下的狀態，當程式中的接點或線圈是使用符號名稱，且該變數符號有編輯符號註解時，畫面中亦會顯示該符號的符號註解，但因裝置註解與符號註解在 ISPSOFT 中是屬於不同的資料，因此使用者將無法直接於程式畫面中編輯符號註解；若要編輯符號註解時，使用者仍必須至符號表中才可進行編輯，相關操作請參考第 6.2.4 節。

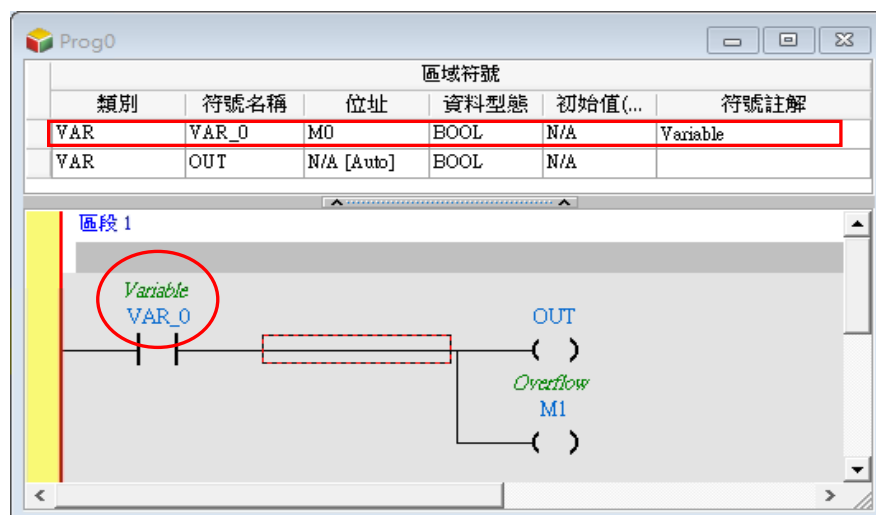
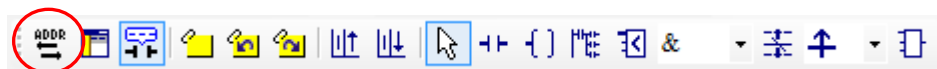
當階梯圖工具列的  圖示為按下的狀態時，若將滑鼠移至某個裝置的上方稍作停留，則此時畫面便會出現該裝置的相關提示，而提示的內容則可於 ISPSOFT 的「選項」功能中進行設定，請參考手冊中第 2.3.1 節的相關說明。




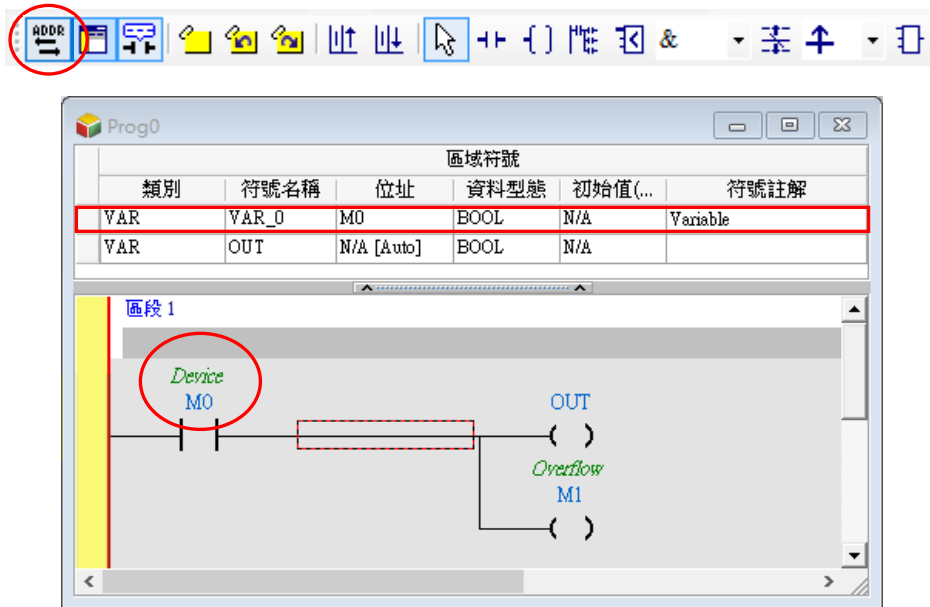
10.2.10 符號模式與位址模式的顯示切換

在 LD 與 FBD 的編輯環境中，當某個變數符號在宣告時有一併指定裝置位址時，藉由切換圖示工具列的  圖示便可變更該變數符號的顯示模式；但未指定裝置位址的變數符號，或未被變數符號所指定的裝置位址則會保持原狀。

如下圖，當  圖示呈現未按下的狀態時，代表目前的顯示方式為符號模式，此時有指定裝置位址的變數符號便會以符號名稱來顯示，而元件上方的註解亦會顯示為符號註解。





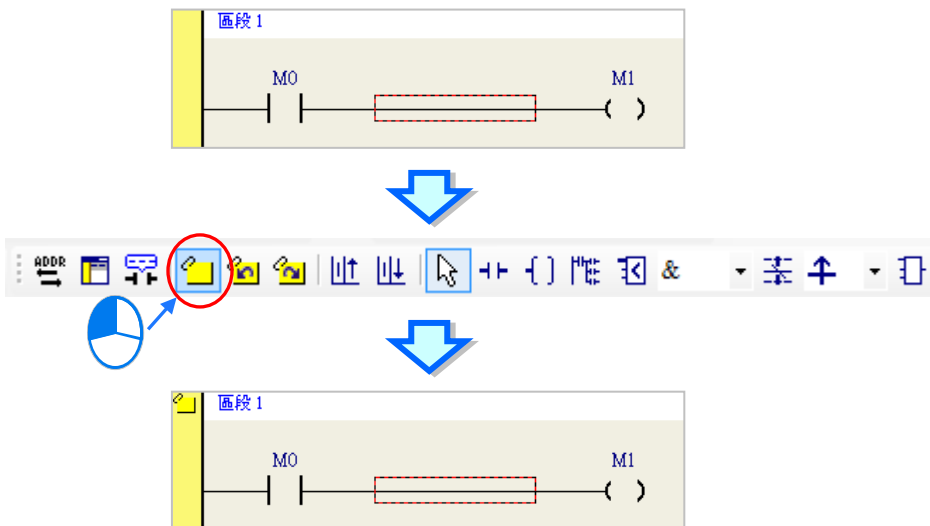
如下圖，當  圖示被按下之後，便代表目前的顯示方式為位址模式，此時有指定裝置位址的變數符號便會以裝置名稱來顯示，而元件上方的註解亦會顯示為裝置註解。





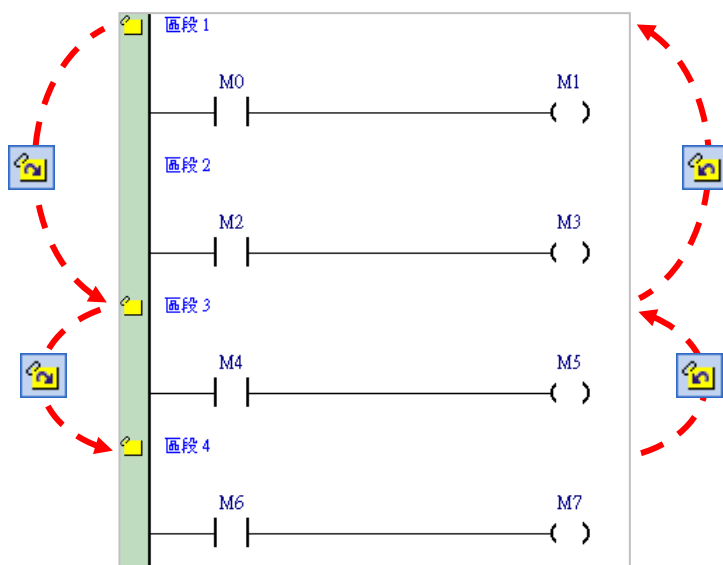
10.2.11 書籤功能



書籤的功能主要在於協助使用者迅速搜尋或移動至加入書籤標註的階梯圖區段。

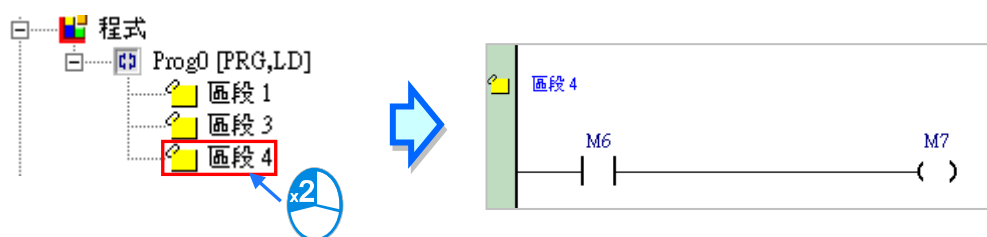
- (1) 點擊圖示工具列的  圖示便可於目前選取的區段中加入一書籤；若要取消書籤時，只要選取該區段後，再次按下  圖示即可。



- (2) 當程式中有插入書籤時，按下工具列圖示  即可移動至下一個書籤的位置，而按下  圖示則可移動至前一個書籤的位置。



- (3) 於程式中的區段插入書籤，會同時長出  圖案於專案管理區相對應的程式下方，滑鼠雙擊  區段編號，即可移動至書籤所在的位置。



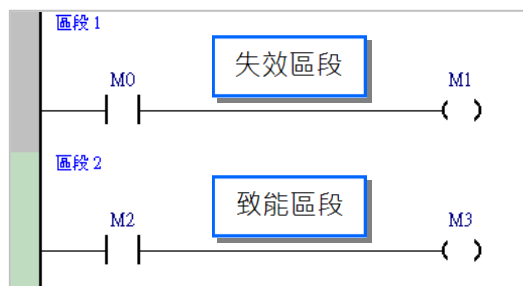
補充說明

於功能工具列的 **編輯 > 書籤** 項目中亦有相同的操作項可供使用，其中點選「**移除所有書籤**」項目則可同時將所有的書籤移除，不過此操作僅會針對目前正在編輯的視窗產生作用。

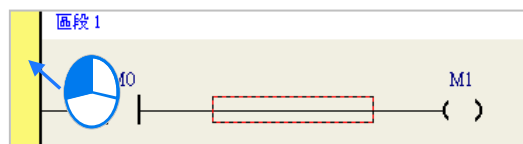
10.2.12 階梯圖區段的致能與失效

若某個階梯圖區段被設為失效狀態，則程式的編譯動作便會主動略過該區段，因此當使用者在進行測試或除錯的工作時，利用此功能便可暫時關閉程式中的某些程序。

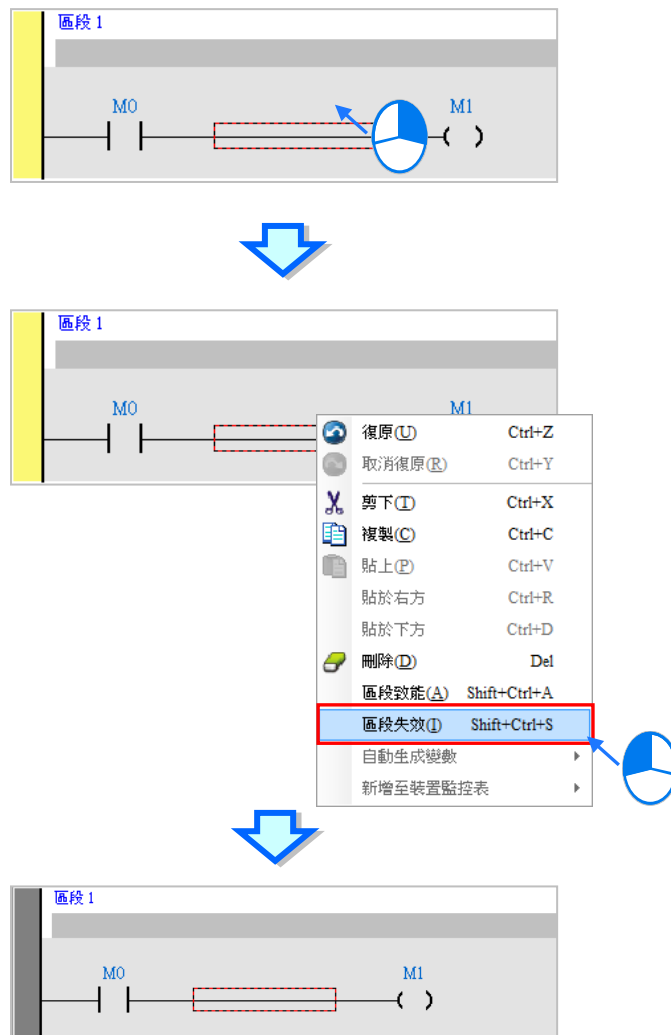
設為**致能**或**失效**的區段，在該區段的左側會使用不同的顏色來表示，而顯示的顏色則可於 ISPSOft 的「選項」功能中進行設定，請參考手冊中第 2.3.1 節的相關說明。



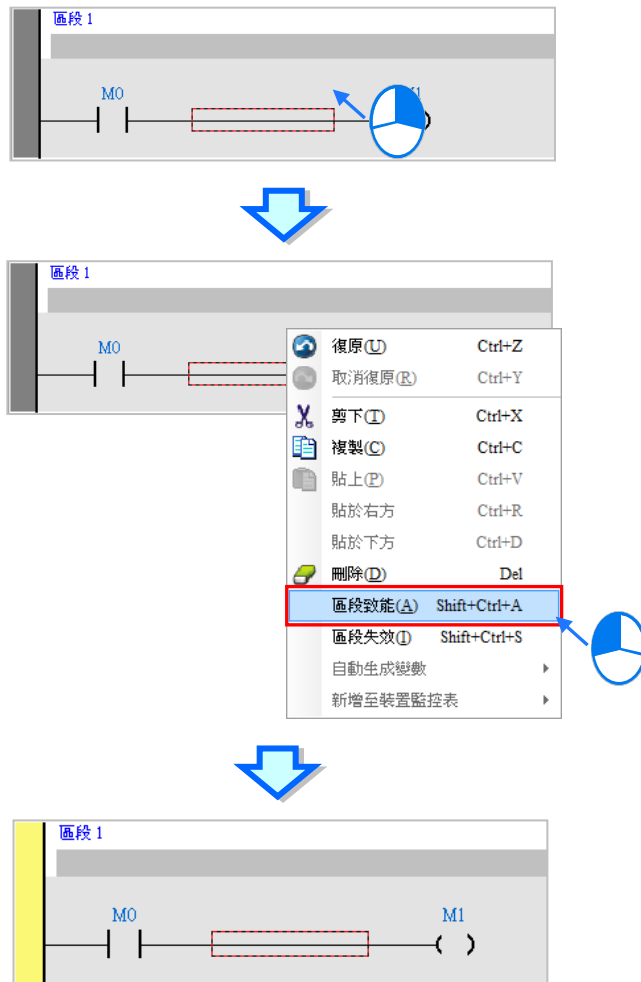
要變更區段的狀態時，請先將該區段選取起來，而選取時亦可同時選取多個區段。



於功能工具列中點選 **編輯 (E)** > **區段失效 (I)** 或是在選取的區段上按下滑鼠右鍵，接著於快捷選單中選取「**區段失效 (I)**」，則此時所選區段則會為失效狀態。



若再點選編輯 (E) > 區段致能 (A) 或是在選取的區段上按下滑鼠右鍵，接著於快捷選單中選取「區段致能 (A)」，則會恢復致能狀態。



第11章 功能塊圖 FBD

目錄

11.1 功能塊圖 FBD 簡介	11-2
11.1.1 關於功能塊圖 (FBD)	11-2
11.1.2 ISPSOft 的 FBD 編輯環境	11-2
11.1.3 功能塊圖區段 (NETWORK)	11-4
11.1.4 選取物件	11-5
11.2 於 ISPSOft 中建立 FBD 程式	11-6
11.2.1 輸入節點與輸出節點	11-7
11.2.2 於 FBD 中使用裝置、變數符號及常數	11-10
11.2.3 AND 與 OR 方塊.....	11-11
11.2.4 反相邏輯	11-14
11.2.5 上緣觸發輸入與下緣觸發輸入	11-16
11.2.6 設置 (SET) 輸出與重置 (RESET) 輸出.....	11-18
11.2.7 API 應用指令、比較接點與功能塊.....	11-20
11.2.8 設置標籤 (Label)	11-24
11.2.9 註解與提示	11-25
11.2.10 符號模式與位址模式的顯示切換.....	11-27
11.2.11 書籤功能	11-28
11.2.12 功能塊圖區段的致能與失效	11-29

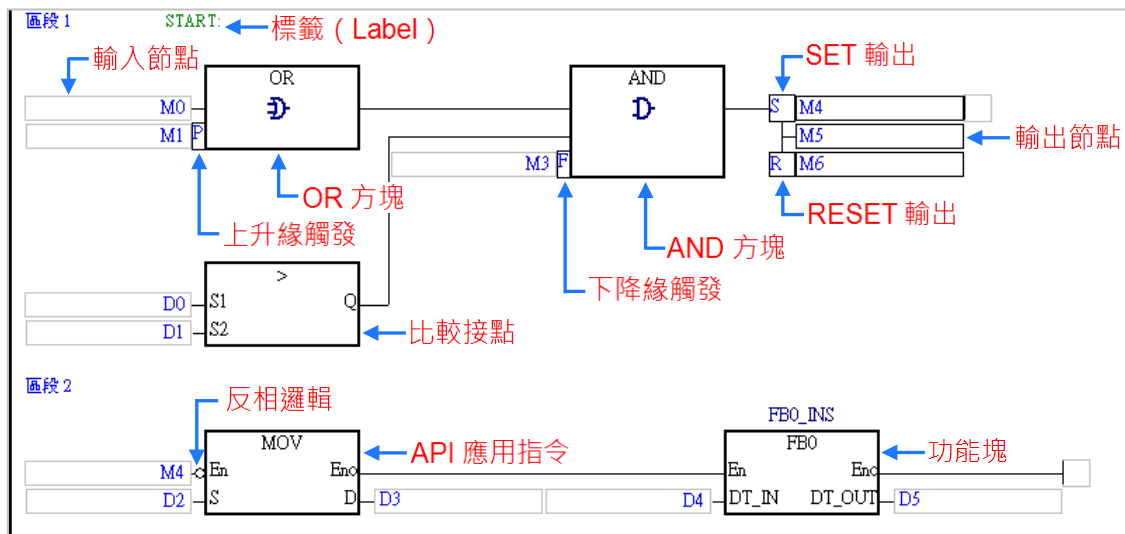
11.1 功能塊圖 FBD 簡介



11.1.1 關於功能塊圖 (FBD)

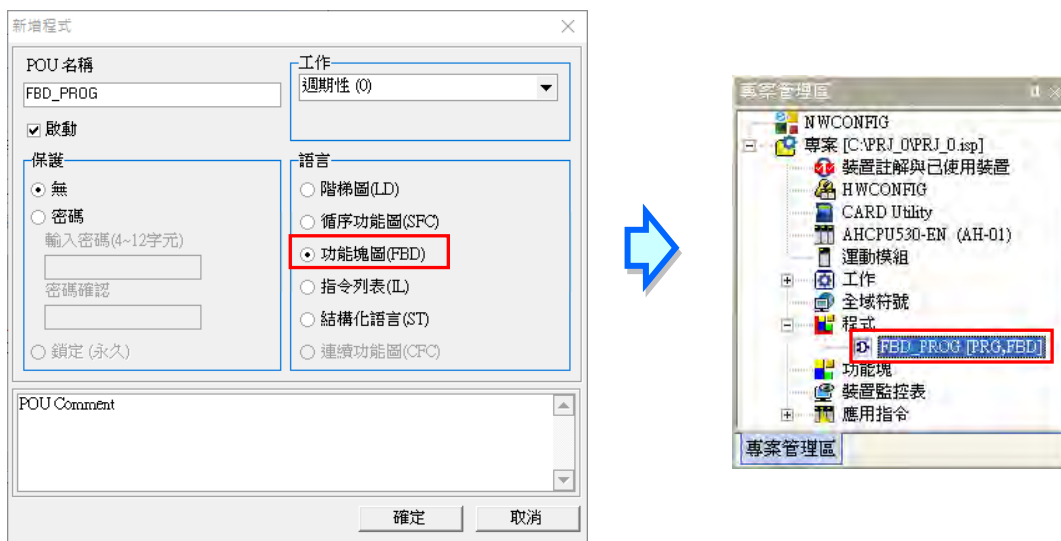
功能塊圖 (**Function Block Diagram, FBD**) 也是符合 IEC 61131-3 的編程語言，與階梯圖一樣屬於圖形化的編程方式，不同的是階梯圖是以接點為主，類似傳統的控制回路圖，而 FBD 則是以功能塊為主，呈現方式較接近於系統方塊圖。FBD 的特點在於著重功能塊的模組化優勢，編程時可先將整個程序有系統的切割成許多功能塊之後，接著再於 FBD 程式中將各個功能塊重新連結部署即可。

下圖即是 ISPSOft 當中的 FBD 程式。

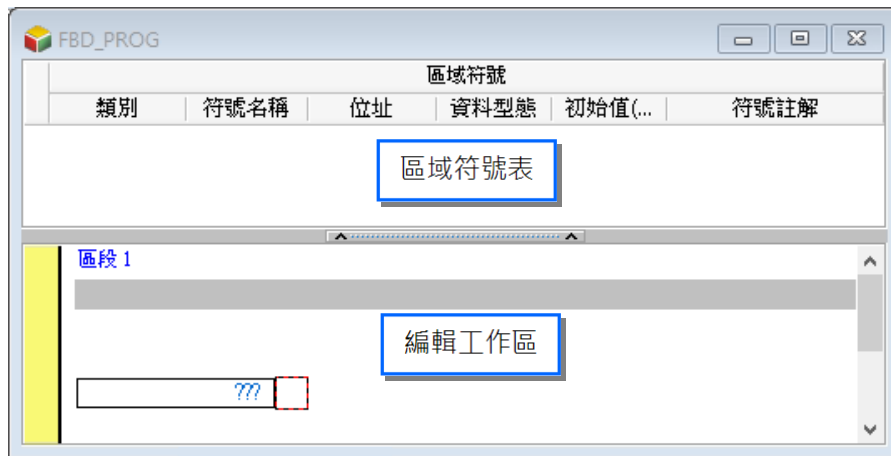


11.1.2 ISPSOft 的 FBD 編輯環境

在建立 POU 時，於語言欄位選擇「功能塊圖 (FBD)」便可建立功能塊圖程式，請參考第 5.4.1 節。



下圖即為 ISPSOft 中的功能塊圖 (FBD) 編輯環境，視窗的上方為區域符號表，下方則是編輯工作區。



進入 FBD 的編輯環境後，ISPSOft 的視窗中便會出現對應的編程圖示工具列，其功能簡介如下。



圖示	鍵盤 (快捷鍵)	功能說明
	無	切換顯示符號或位址
	Shift + Ctrl + C	顯示/隱藏區段註解
	無	開啟/關閉提示功能
	Shift + Ctrl + A	將目前選取的功能塊圖區段設為致能或失效狀態
	Shift + Ctrl + B	在目前選取的功能塊圖區段上加入或取消書籤標記
	Shift + Ctrl + P	移至前一個書籤的標記位置
	Shift + Ctrl + N	移至下一個書籤的標記位置
	Ctrl + I	於目前選取的功能塊圖區段之前插入一個新區段
	Shift + Ctrl + I	於目前選取的功能塊圖區段之後新增一個新區段
	Shift + F1	插入 AND 方塊
	Shift + F2	插入 OR 方塊
	Shift + F3	插入輸出節點
	Shift + F4	增加 AND/OR 方塊的輸入端點
	Shift + F5	刪除 AND/OR 方塊的輸入端點
	Shift + F6	插入反相邏輯
	Shift + F7	插入設置 (SET) /重置 (RESET) 輸出

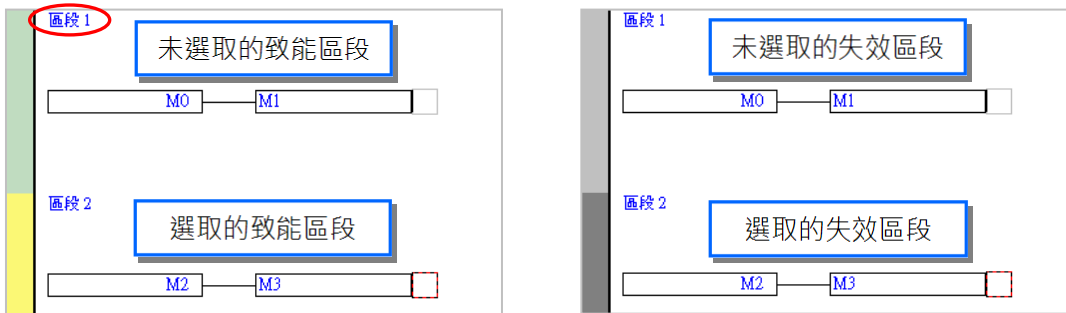
圖示	鍵盤 (快捷鍵)	功能說明
	Shift + F8	插入上緣/下緣觸發
	Shift + Ctrl + U	插入指令或功能塊

*.插入指令與功能塊亦可使用鍵盤直接輸入的方式，請參考本章第 11.2.7 節的相關說明。

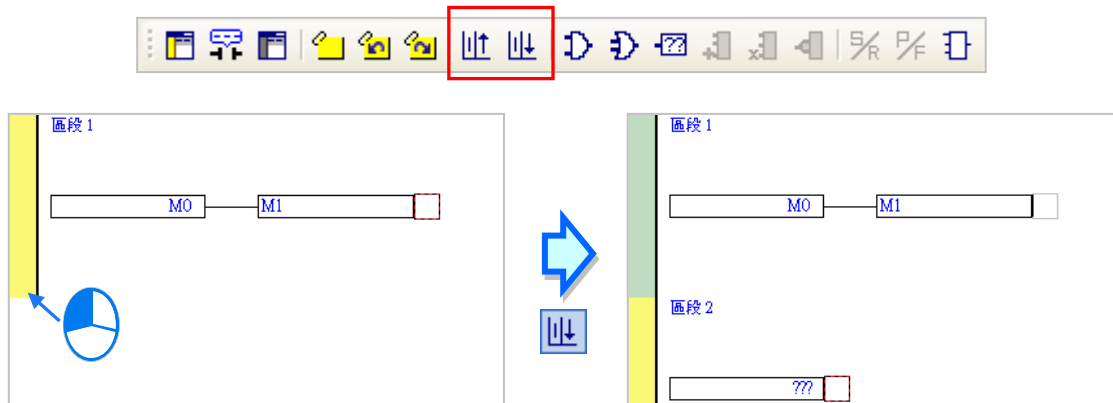
11.1.3 功能塊圖區段 (NETWORK)


功能塊圖 (FBD) 程式與階梯圖 (LD) 程式一樣是以區段 (NETWORK) 為編輯單位，每個功能塊圖區段都是一個獨立的運算程序，而一個功能塊圖程式亦是由許多的區段所建構而成。在 ISPSOft 中，每個功能塊圖區段可插入的元件數量並沒有特別的限制。

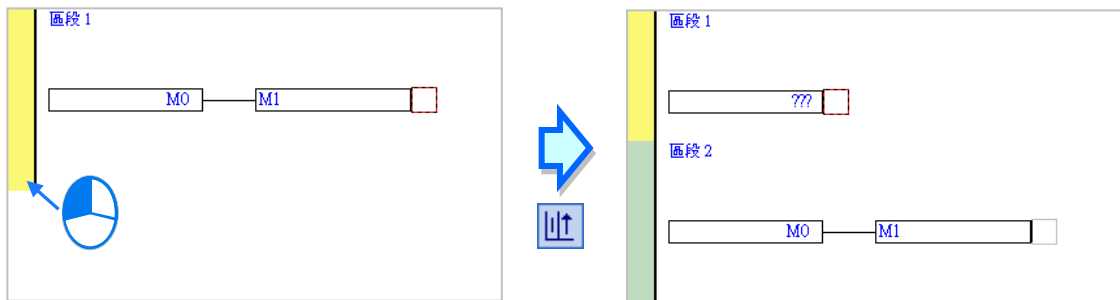
在每個功能塊圖區段的左上角都會標示該區段的區段編號，且由區段左側的顏色便可知該區段目前的狀態，其狀態的顯示依功能可分為**致能與失效**，依編輯狀態則可分為**選取與未選取**，而所顯示的顏色則可於 ISPSOft 的「選項」設定中由使用者自行定義，詳細操作請參考第 2.3.1 節。



當新增一個 FBD 程式時，系統會預先建立一個空白的功能塊圖區段，而要新增區段時，請先選取某個現有區段後，於圖示工具列中點擊 圖示即可於選取的功能塊圖區段下方新增一空白區段。

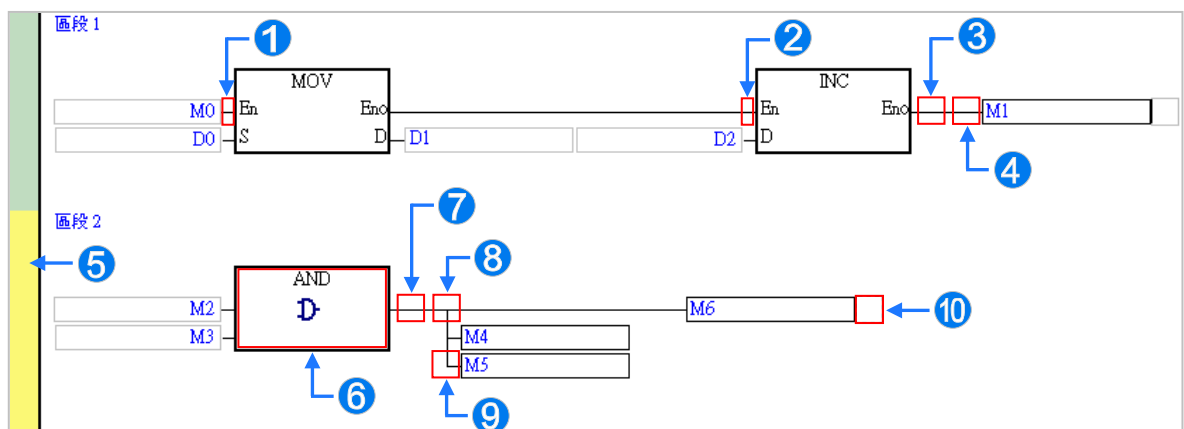


於圖示工具列中點擊  圖示則可於選取的功能塊圖區段上方插入一空白區段。



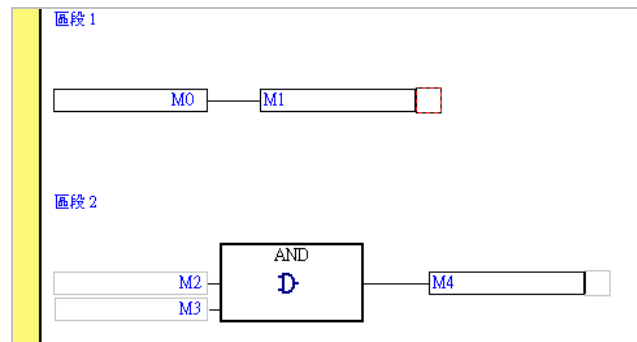
11.1.4 選取物件

在編輯工作區中的紅色方框便代表目前的編輯位置，也代表目前所選取的物件，而編輯時亦必須將滑鼠移至紅框的位置點擊後才可進行選取；此外，根據選取位置的不同，編輯時可使用的功能也會有所差異，而當進行一般的複製/剪下/貼上/刪除的操作時，不同的選取位置，其所包含的操作對象也會有很大的不同；下列即為不同選取位置所代表的意義。



- ① 選取此點左側的輸入節點，如上圖的 M0。
- ② 選取此點之前的所有區塊。
- ③ 選取此點之前的所有區塊。
- ④ 選取右側的輸出節點，如上圖的 M1。
- ⑤ 選取整個區段。
- ⑥ 選取方塊。
- ⑦ 選取此點之前的所有區塊。
- ⑧ 選取此點下方的所有輸出節點。
- ⑨ 選取此點右側的輸出節點。
- ⑩ 選取此點之前的所有區塊。

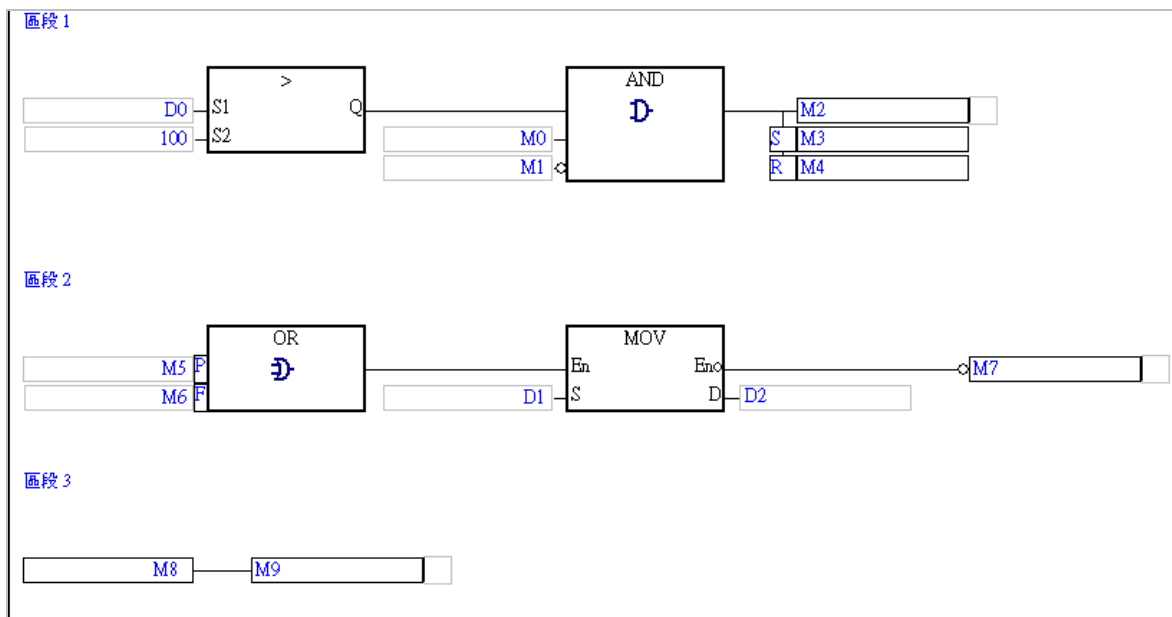
當要同時選取多個功能塊圖區段時，請先按住鍵盤的【Ctrl】鍵，接著便可用滑鼠分別點選多個不連續位置的區段；另外也可先按住【Shift】鍵後，再分別用滑鼠點選欲選取範圍的第一個與最後一個功能塊圖區段，之後便可將整個範圍內的區段全部選取起來。



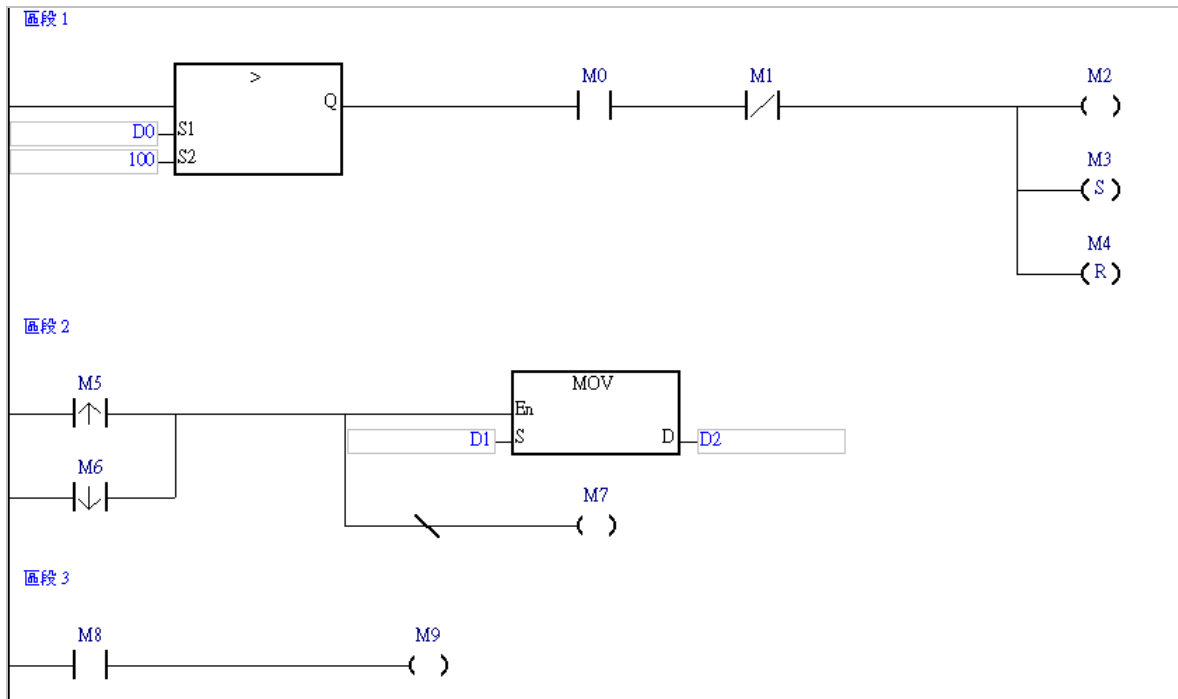
11.2 於 ISPSOft 中建立 FBD 程式

下列為基本的功能塊圖 (FBD) 程式範例與其等效的階梯圖 (LD) 程式，而關於各種 FBD 圖示所代表的意義與其建立的方式，我們也將在本節後續的內容當中逐一介紹。

- 功能塊圖 (FBD) 程式



● 等效階梯圖 (LD) 程式



11.2.1 輸入節點與輸出節點

在 FBD 的圖形中，裝置與變數符號皆會以節點的圖示來表達，而雖然所代表的節點圖示並不會因資料型態的不同而有所區分，但實際運作時，彼此仍必須符合下述原則才可互相傳遞資料。

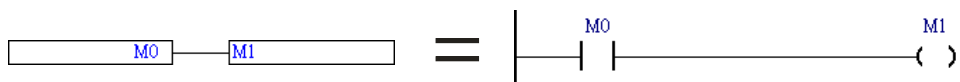
- (a) M、S、T、C、HC 裝置與 BOOL、STEP、TIMER、COUNTER 等具備位元型態的符號可互傳。
- (b) D、L 裝置可與所有 BOOL、STEP、REAL、STRING 以外所有型態的符號互傳資料。
- (c) T、C 裝置可與 D、L 裝置及 WORD、INT、TIMER、COUNTER 等型態的符號互傳資料。
- (d) D、L、HC 裝置與指定 HC 裝置的 COUNTER 符號可與 DWORD 與 DINT 型態的符號互傳資料。
- (e) WORD 系列與 INT 系列可互傳資料，但輸出節點的資料長度必須大於或等於輸入節點。
- (f) REAL 系列與 STRING 只可與相同型態且長度相同的符號互相傳遞資料。
- (g) 多重輸出節點的結構中，每個輸出節點的型態皆須符合上述規定，而每組多重輸出至多只可增加至 32 個，但只要往右擴增輸出節點之後，便可再建立一組多重輸出。

● 圖示

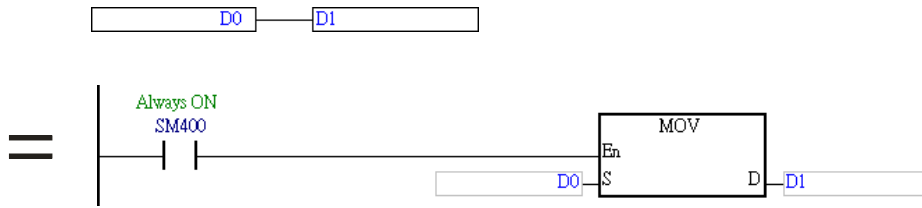


● 範例

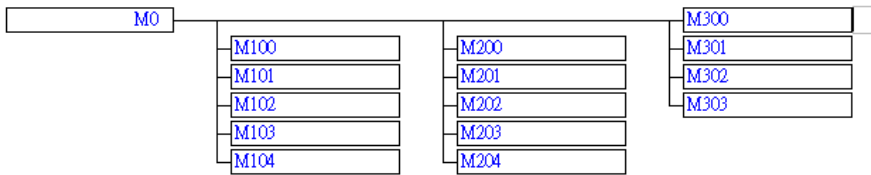
EX1：將 M0 接點的狀態輸出至 M1。



EX2：將裝置 D0 的資料傳遞給裝置 D1。



EX3：以 M100、M200 與 M300 為首，下圖總共有三組多重輸出，其中每組多重輸出至多可再向下增加到 32 個輸出節點，而向右則可繼續擴增，且數量上並沒有特別限制。

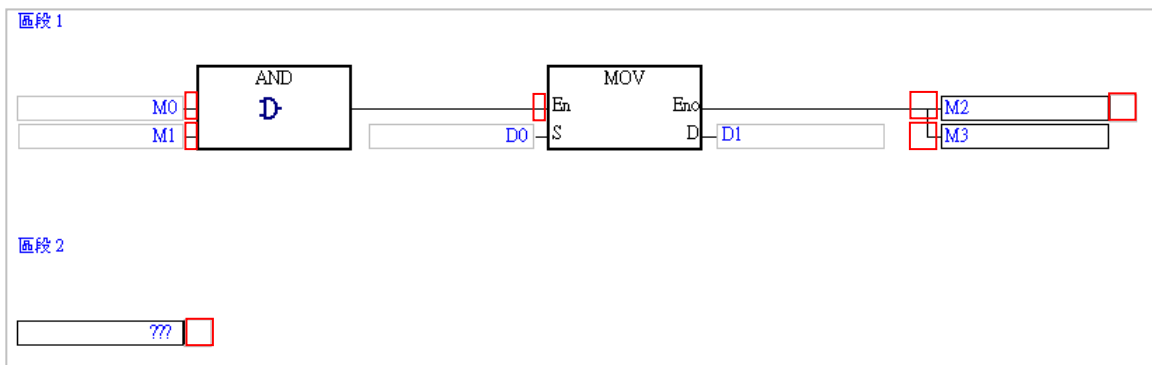



EX4：下表則為幾個錯誤的範例。

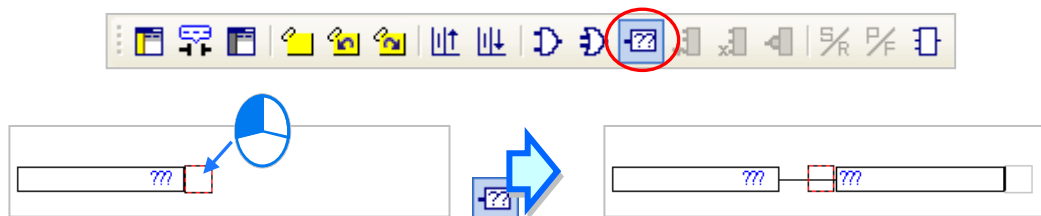
錯誤範例	說明
	輸入節點與輸出節點的資料型態不符。
	輸出節點的資料長度不得小於輸入節點。
	REAL 符號不可直接與 D 裝置互傳資料。
	REAL 系列的符號必須長度相同才可互傳資料。
	STRING 符號必須長度相同才可互傳資料。
	TIMER 符號不可輸出給 DWORD 符號，但可傳給 WORD。
	HC 裝置不可輸出給 WORD 符號，但可傳給 DWORD。
	多重輸出的每個輸出節點皆需符合規則，而左例的錯誤則是其中一個輸出節點的資料長度小於輸入節點。

● 插入輸出節點

(1) 於編輯工作區中點選欲插入輸出節點的位置，而下圖紅框的位置則都允許插入輸出節點。

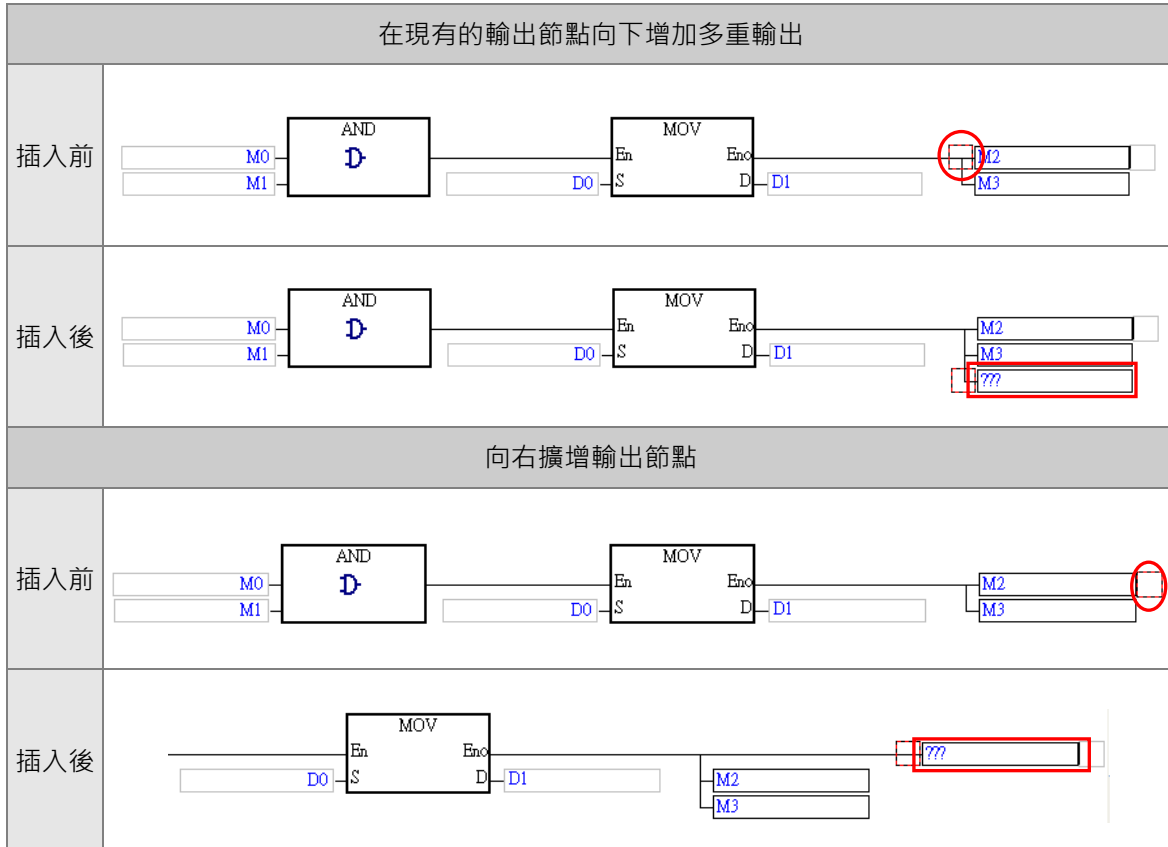


(2) 除了上圖的例子之外，其他的結構也都有可能允許插入輸出節點，若所點選的位置允許加入輸出節點時，圖示工具列中的  圖示便會呈現可點選的狀態，此時點擊該圖示便可插入輸出節點。



下表中則列舉幾個輸出節點的結構。

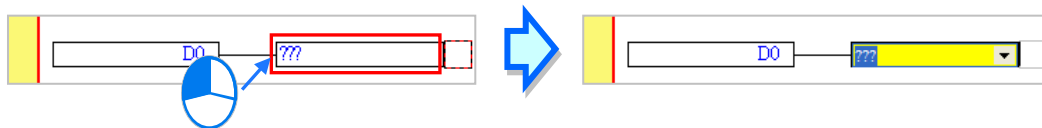
於 AND/OR 方塊的輸入接腳插入輸出節點	
插入前	
插入後	
於 AND/OR 方塊的輸出接腳插入輸出節點	
插入前	
插入後	



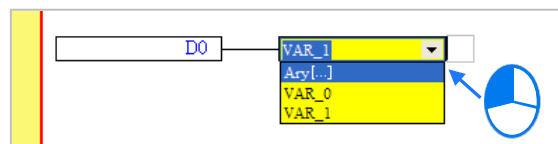
*.功能塊或 API 應用指令本身的運算元端點不允許建立多重輸出。

11.2.2 於 FBD 中使用裝置、變數符號及常數

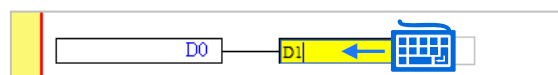
欲編輯節點或指令的運算元時，請直接於該元件的位置上點擊滑鼠左鍵，之後該元件便會呈現編輯狀態，接著再依據下述方式進行操作即可。



直接點擊編輯欄位右側的箭頭按鈕，接著便可於下拉選單中選擇已宣告好的變數符號；而當有宣告陣列符號時，於下拉選單中可便可挑選該陣列的各個元素，如下圖的 Ary[...]。



另外亦可直接於編輯欄位以鍵盤輸入裝置名稱或是變數符號，而當輸入未經宣告的變數符號時，輸入完畢後按下鍵盤【Enter】鍵，系統預設會自動導出符號宣告視窗。(設定方式請參考第 2.3.1 節)



當用滑鼠點擊某個區段後按下【Enter】鍵，接著系統便會自動開啟編輯欄位，而利用鍵盤的【Enter】鍵便可在同一個區段中反覆移動編輯欄位，利用【Tab】鍵則可跨區段移動，但到了最後一個編輯欄位後便會停止；編輯時，使用者可直接於編輯欄位中使用鍵盤輸入，而若按下【Page Down】鍵時，該欄位的下拉選單便會隨之展開，利用鍵盤的上下鍵便可進行選擇；完成編輯後，按下【Enter】鍵便會自動再切換至該區段中的下一個編輯位置，而要結束編輯時，只要按下【Esc】鍵即可。

補充說明

(a) 要在 ISPSOft 的功能塊圖 (FBD) 中使用常數時，其格式的範例如下。

- 10 進制：23456 (未加任何標註的數字會被視為 10 進制)
- 16 進制：16#5BA0 (數字前方加上 16#)
- 8 進制：8#55640 (數字前方加上 8#)
- 2 進制：2#101101110100000 (數字前方加上 2#)
- 字串：“XYZ” (以雙引號包住字串)
- BOOL：須使用常開/常閉接點，可以使用 SM400/SM401 (AH/AS 系列機種) 或 M1000/M1001 (DVP 系列機種) 來表達。

(b) 使用陣列時，其輸入格式為 符號名稱[索引]，如上圖下拉選單中的 Ary[1]。

11.2.3 AND 與 OR 方塊

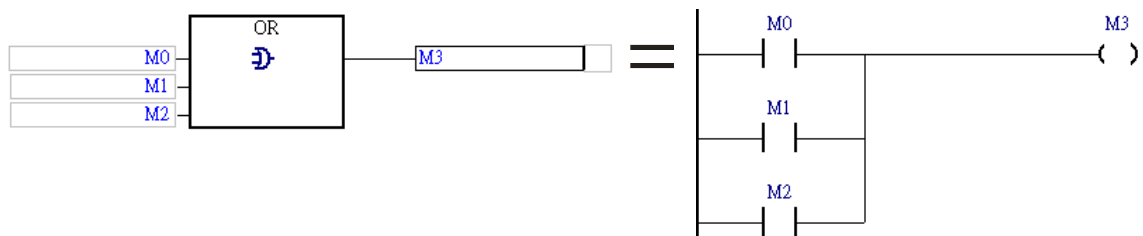
FBD 中的接點串並聯皆是透過 AND 與 OR 的方塊來表達，而其輸入與輸出節點則必須為布林格式，且數量至多可增加至 32 個；此外，AND/OR 的輸出至少須有一個輸出節點或是再串接至下一元件。

● 圖示

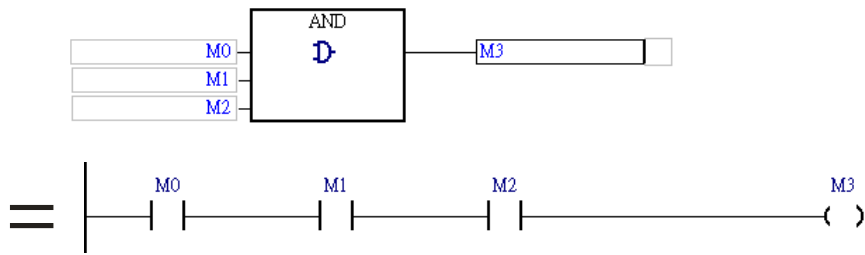
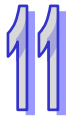


● 範例

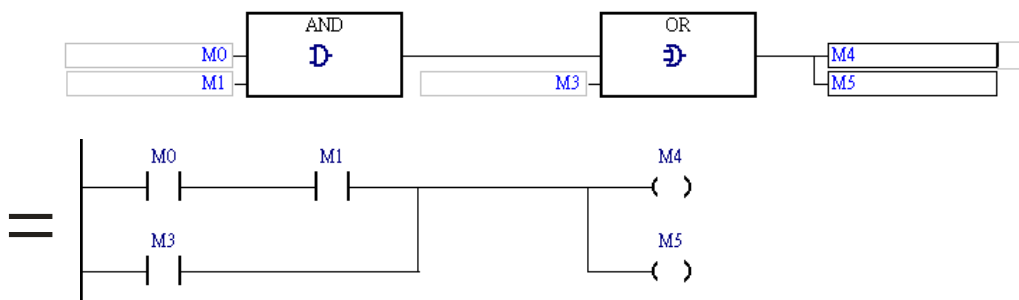
EX1：將 M0、M1、M2 三個接點輸入 OR 方塊進行運算，並將結果輸出給 M3；其運算則相當於將 M0、M1、M2 三個接點並聯之後，再將並聯後的導通結果拿來驅動 M3 線圈。



EX2：將 M0、M1、M2 三個接點輸入 AND 方塊進行運算，並將結果輸出給 M3；其運算則相當於將 M0、M1、M2 三個接點串聯之後，再將串聯後的導通結果拿來驅動 M3 線圈。

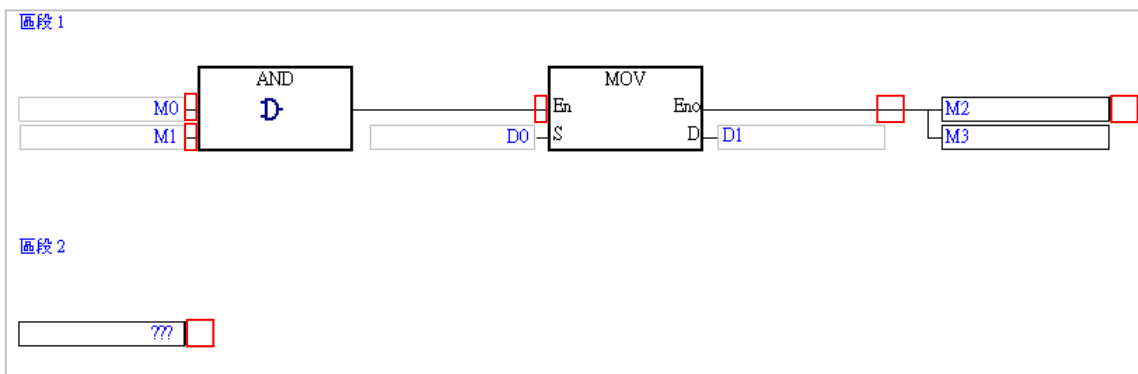




EX3：下圖則為混合 AND/OR 方塊所建立的邏輯運算，亦相當於階梯圖 (LD) 中的串並聯結構。



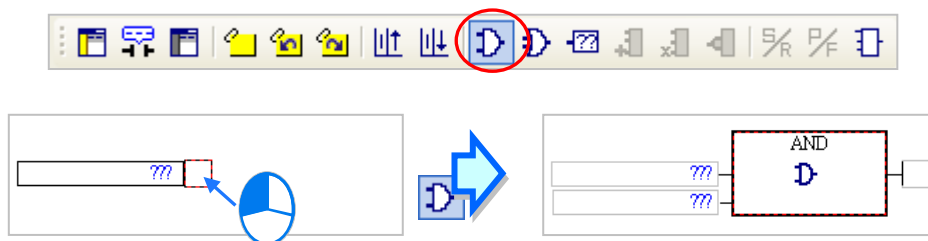
● 插入 AND/OR 方塊


(1) 於編輯工作區中點選欲插入 AND/OR 方塊的位置，而下圖紅框的位置則都允許插入方塊。

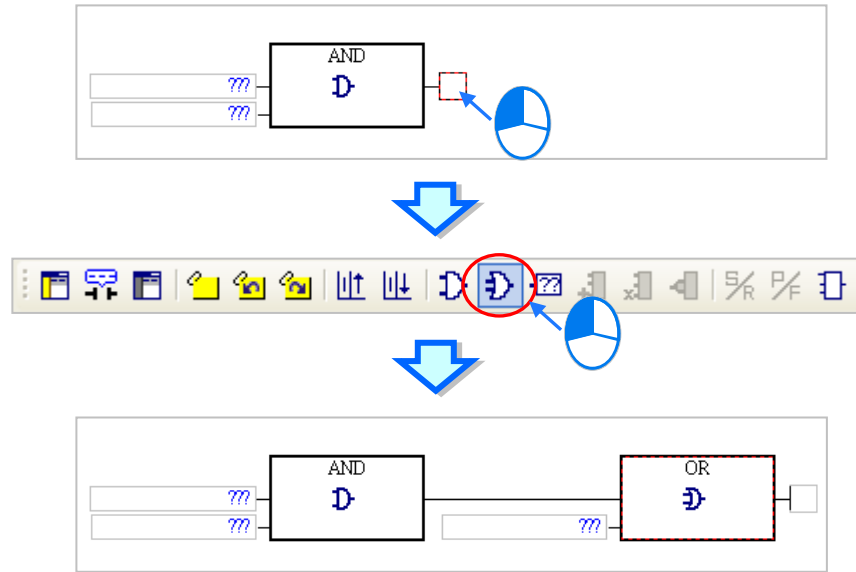


(2) 除了上圖的例子之外，其他的結構也都可能允許插入方塊，而若所點選的位置允許加入方塊時，圖示工具列中的  與  圖示便會呈現可點選的狀態。


點擊  圖示時便可將 AND 方塊插入所選的位置。

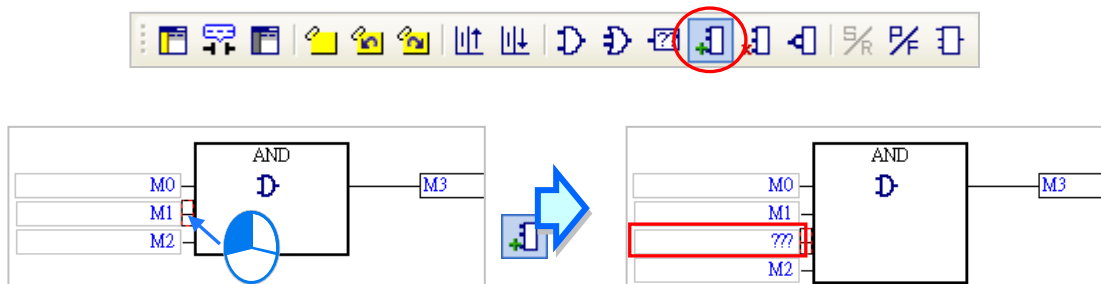


點擊  圖示則可於所選位置插入 OR 方塊。

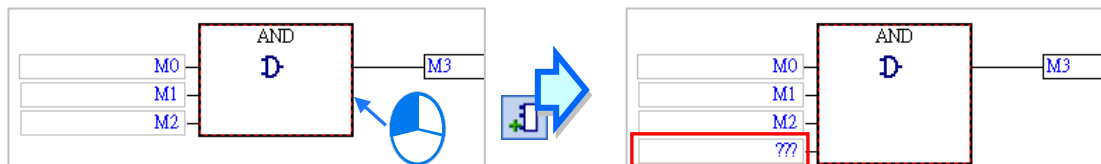


● 增加 AND/OR 方塊的輸入節點


首先點選某個 AND/OR 方塊的輸入節點，接著點擊圖示工具列的  圖示便會於所選節點的下方新增一節點；但須注意若點選的位置是位於最上方的輸入節點，如下圖的 M0，便無法新增輸入節點。



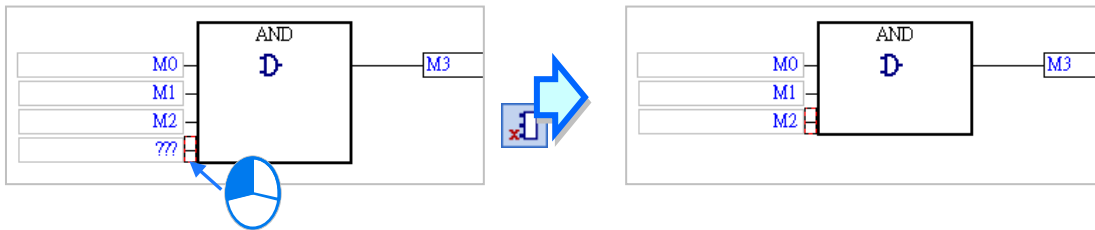
而若一開始選取的位置是在方塊上，則輸入節點會直接新增於方塊的最下方。



● 刪除 AND/OR 方塊的輸入節點

首先於 AND/OR 方塊中點選欲刪除的輸入節點後，點擊圖示工具列的  圖示便可將其刪除，但刪除節點時須注意下列兩點限制。

- (a) 位於最上方的輸入節點，如下圖的 M0，不可刪除。
- (b) 只剩下兩個輸入節點時便不允許再刪除輸入節點。



11.2.4 反相邏輯

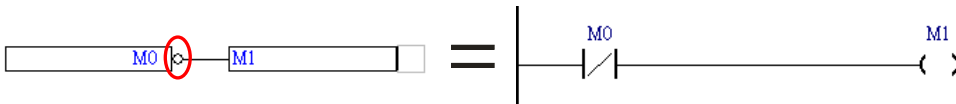
在 FBD 的圖形中，對於具有布林邏輯的位置，如 BOOL 型態的輸出與輸入節點、功能塊或 API 應用指令的 En 接點，我們便可在該位置上插入反相邏輯的運算；不過須注意，功能塊或 API 應用指令本身的運算元端點則不允許加上反相邏輯。

● 圖示

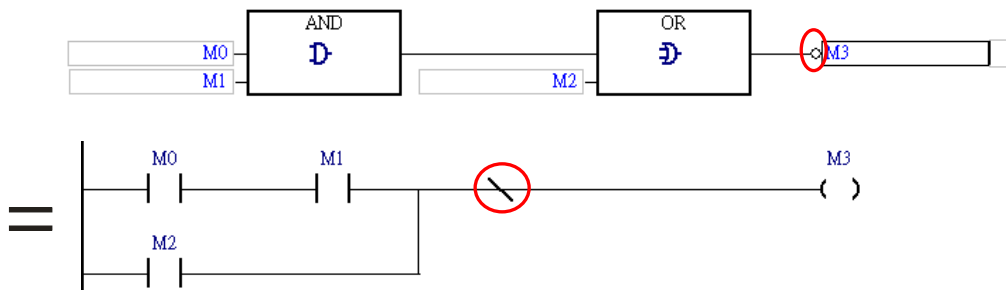


● 範例

EX1：將 M0 的狀態反相後輸出至 M1，其操作等同讓 M0 的常閉接點直接驅動 M1 線圈。



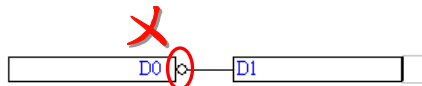
EX2：在輸出節點加上反相邏輯便代表將前方區塊的邏輯運算結果經過反相後再輸出。



EX3：下圖是一個二選一的程序執行架構。在同一個功能塊圖區段中，當 M0 為 ON 時，系統將會執行功能塊 BLACK；而當 M0 為 OFF 時，系統則會執行功能塊 WHITE。

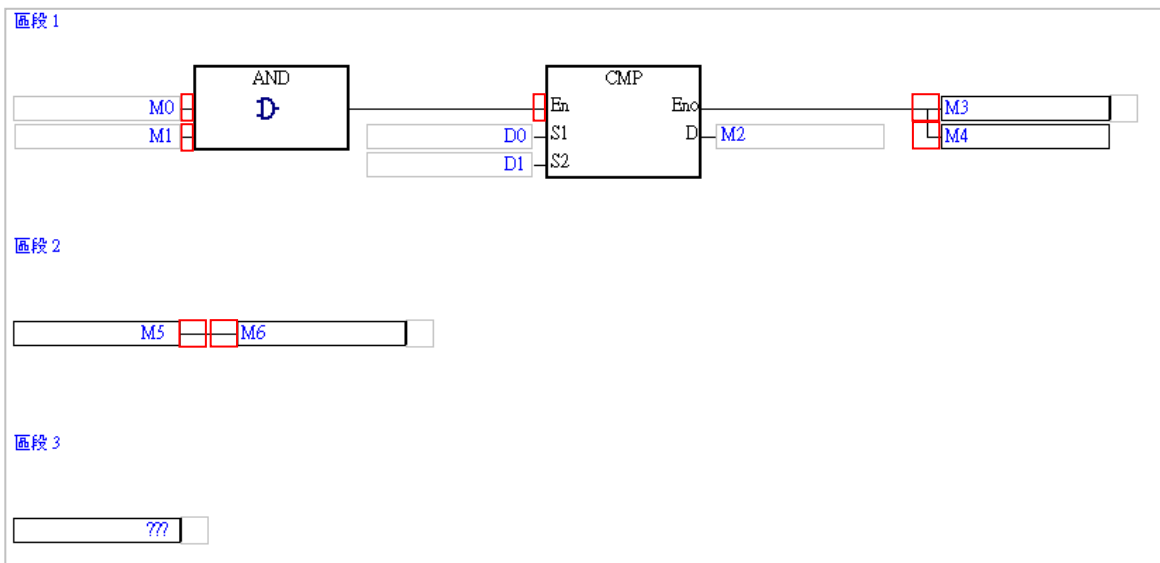



EX4：下圖為錯誤的範例，反相邏輯不得放置於非布林型態的節點上。

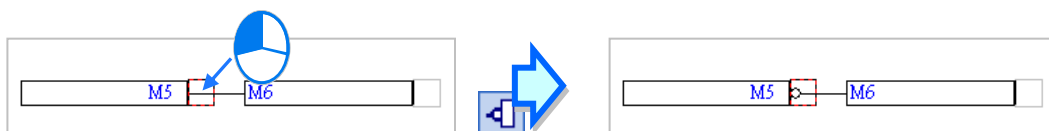
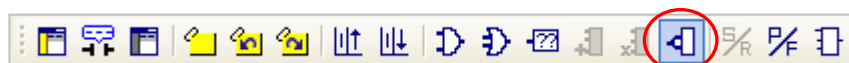



● 插入反相邏輯

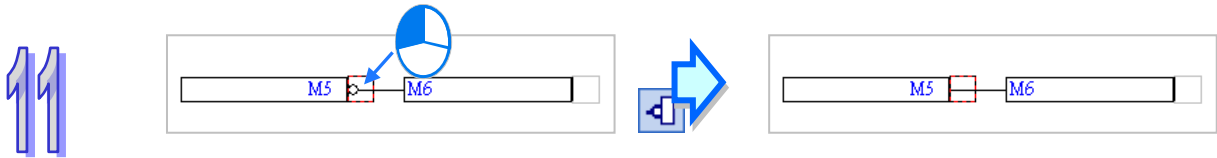
(1) 於編輯工作區中點選欲插入反相邏輯的位置，而下圖紅框的位置則都允許插入反相邏輯。其中 CMP 指令的輸出雖然也是 BOOL 型態的運算元，不過在 FBD 的程式中，功能塊或 API 應用指令本身的運算元端點並不允許進行附加的狀態編輯，如增加多重輸出或加上反相邏輯；此外，反相邏輯亦不允許加在沒有輸出節點或未繼續串接其他元件的位置上，如區段 3 的輸入節點。





(2) 除了上圖的例子之外，其他的結構也都有可能允許插入反相邏輯，若所點選的位置允許加入反相邏輯時，圖示工具列中的  圖示便會呈現可點選的狀態，點擊該圖示便可插入反相邏輯。



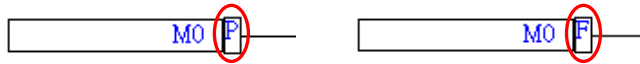
欲取消反相邏輯時，只要於相同位置再次點擊圖示工具列中的  圖示即可。



11.2.5 上緣觸發輸入與下緣觸發輸入

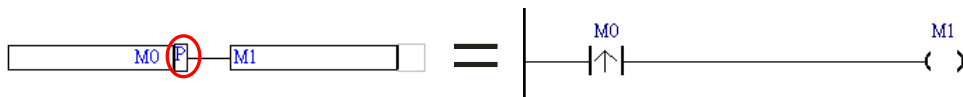
在 ISPSOft 的 FBD 圖形中，輸入節點上的  符號代表上緣觸發，而  符號則代表下緣觸發；在進行編輯時，邊緣觸發只能加在布林型態的輸入節點，或是 AND/OR 方塊的輸入端點，而功能塊或 API 應用指令本身的運算元及 En 端點則不允許加上邊緣觸發。

● 圖示

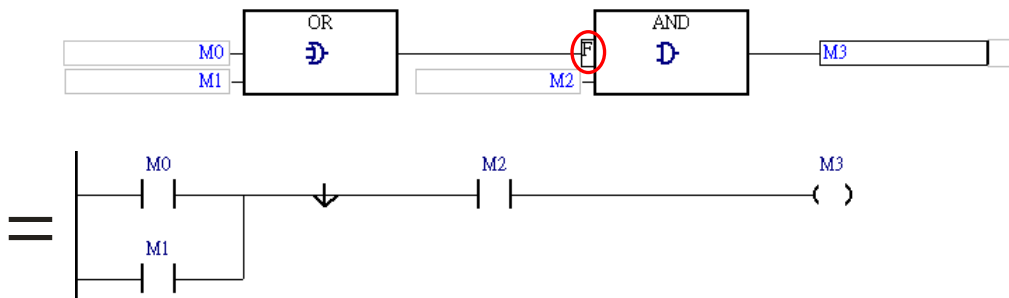


● 範例

EX1：當接點 M0 的狀態由 OFF 變為 ON 的週期，M1 的輸出便會為 ON，而在下個週期時，M1 則會再度恢復為 OFF。



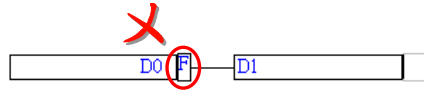
EX2：在 AND/OR 方塊的輸入端點處，雖然沒有明確的輸入節點，但仍允許加上邊緣觸發。



EX3：如下圖，第一個 API 應用指令的 En 端點因直接連接到輸入節點 M0，因此便可在 M0 的位置加上邊緣觸發，不過第二個 API 應用指令的 En 端點便不允許加上邊緣觸發的運算。

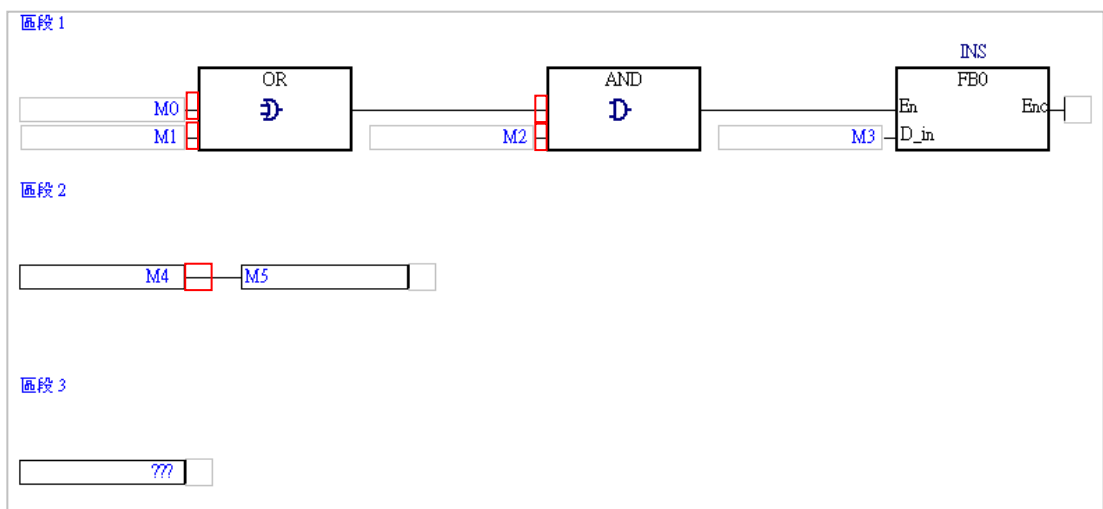



EX4：下圖為錯誤的範例，邊緣觸發不能加在非布林型態的輸入節點上。

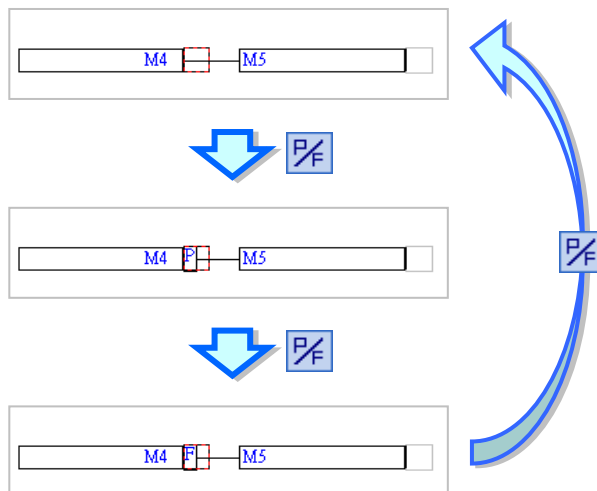
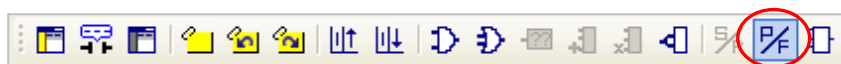


● 插入邊緣觸發

- (1) 於編輯工作區中點選欲插入邊緣觸發的位置，而下圖紅框的位置則都允許插入邊緣觸發。其中功能塊 FB0 的輸入端點雖然也是 BOOL 型態的運算元，不過在 FBD 的程式中，功能塊或 API 應用指令本身的運算元端點並不允許進行附加的狀態編輯，如加上反相邏輯或邊緣觸發；此外，邊緣觸發亦不允許加在沒有輸出節點或未繼續串接其他元件的位置上，如區段 3 的輸入節點。



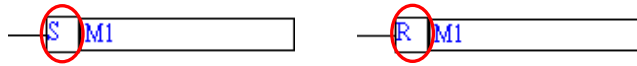
- (2) 除了上圖的例子之外，其他結構也可能允許插入邊緣觸發，若所點選的位置允許加入邊緣觸發時，工具列中的  圖示便會呈現可點選的狀態，反覆點擊該圖示便可切換邊緣觸發的狀態。



11.2.6 設置 (SET) 輸出與重置 (RESET) 輸出

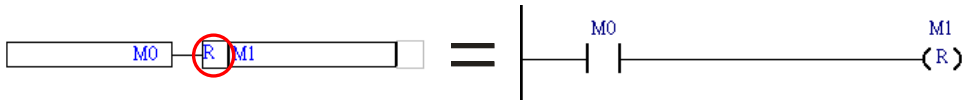
在 ISPSOft 的 FBD 圖形中，若輸出節點加上 **S** 符號便代表設置 (SET) 輸出，而加上 **R** 符號則代表重置 (RESET) 輸出；在進行編輯時，設置 (SET) 與重置 (RESET) 只能加在布林型態的輸出節點，而功能塊或 API 應用指令本身的運算元則不允許加上設置 (SET) 或重置 (RESET) 的運算。

● 圖示

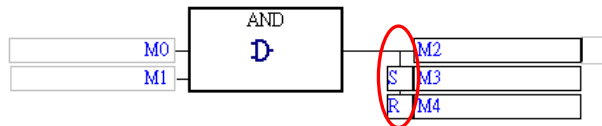


● 範例

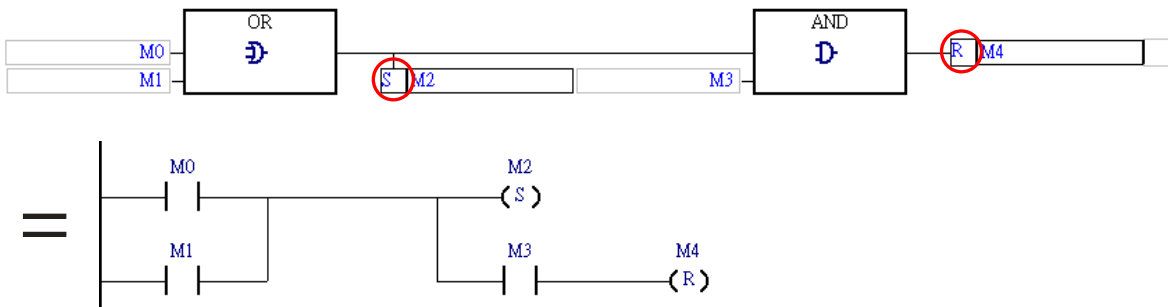
EX1：當 M0 為 ON 時，裝置 M1 便會被重置 (RESET) 為 OFF。



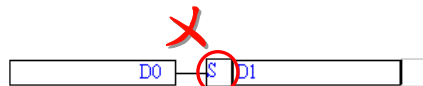
EX2：如下圖，在多重輸出的結構中，每個輸出節點皆可獨立編輯其輸出型態。



EX3：當 M0 與 M1 任一個接點為 ON 時，M2 便會被設置 (SET) 為 ON；而當 M0 與 M1 任一個接點為 ON 且 M3 也為 ON 時，M4 便會被重置 (RESET) 為 OFF。

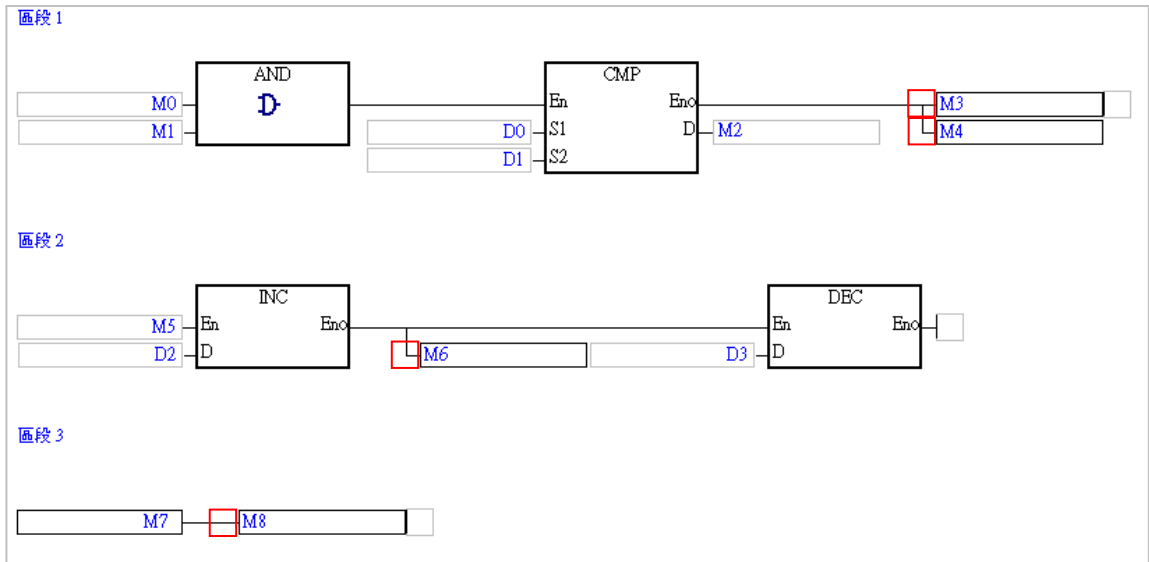



EX4：下圖為錯誤的範例，設置 (SET) 或重置 (RESET) 輸出不能加在非布林型態的輸出節點。

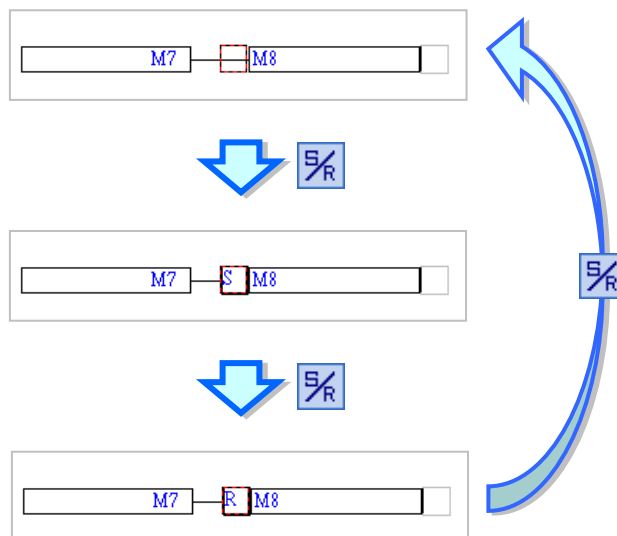


● 插入設置 (SET) 或重置 (RESET) 輸出

- (1) 於編輯工作區中點選欲插入設置 (SET) 或重置 (RESET) 輸出的位置，而下圖紅框的位置則都允許插入設置 (SET) 或重置 (RESET) 輸出。其中 CMP 指令的輸出端點雖然也是 BOOL 型態的運算元，不過在 FBD 的程式中，功能塊或 API 應用指令本身的運算元端點並不允許進行附加的狀態編輯，如加上反相邏輯或設置 (SET) /重置 (RESET) 輸出。



- (2) 除了上圖的例子之外，其他結構也可能允許插入設置 (SET) 或重置 (RESET) 輸出，若所點選的位置允許加入設置 (SET) 或重置 (RESET) 輸出時，工具列中的  圖示便會呈現可點選的狀態，而反覆點擊該圖示便可切換所選位置的輸出狀態。

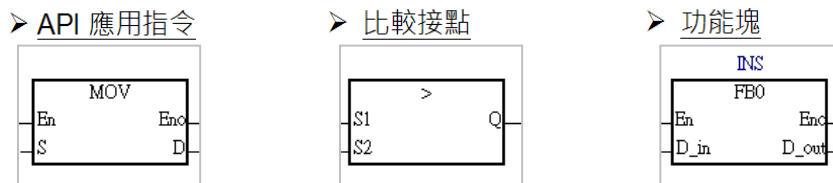


11.2.7 API 應用指令、比較接點與功能塊

在 ISPSOft 的 FBD 圖形中，API 應用指令、比較接點與功能塊的圖示與階梯圖 (LD) 相同，皆是以方塊的圖示來呈現，且在功能塊圖示的上方同樣會有一個**功能塊實例**的欄位。

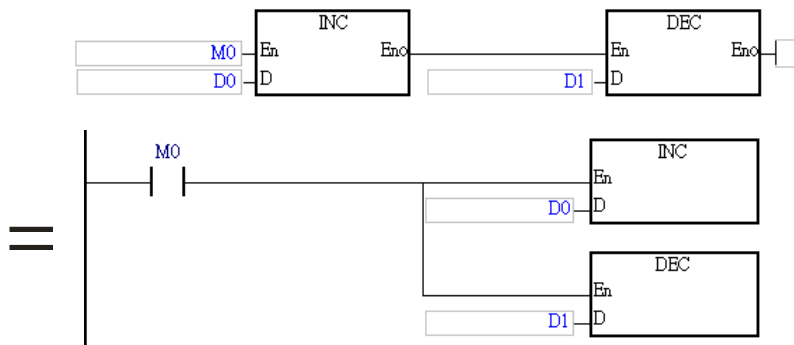
FBD 的 API 應用指令及功能塊用法與 LD 相同，當連接至致能端點 En 的邏輯狀態為 ON 時，該功能塊或 API 應用指令才會被執行；不過基於 FBD 的特性，於同一區段中的功能塊或 API 應用指令之間皆可再插入其它的邏輯運算，如反向邏輯或 AND/OR 方塊，因此使用起來具有相當高的靈活性。

● 圖示

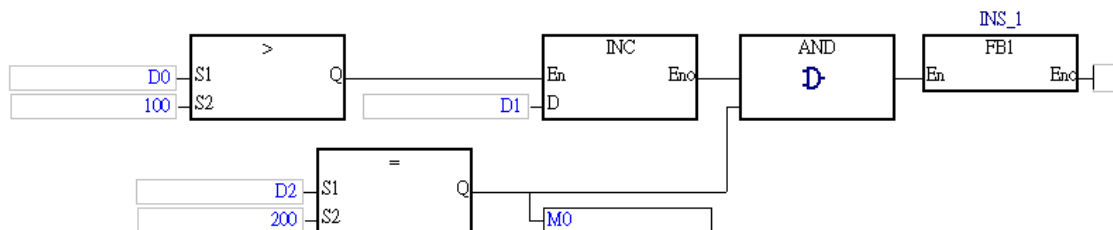


● 範例

EX1：FBD 中的 API 應用指令有包含 Eno 端點，因此在 FBD 的程式中，API 應用指令可於其後方繼續串接其他指令或元件。



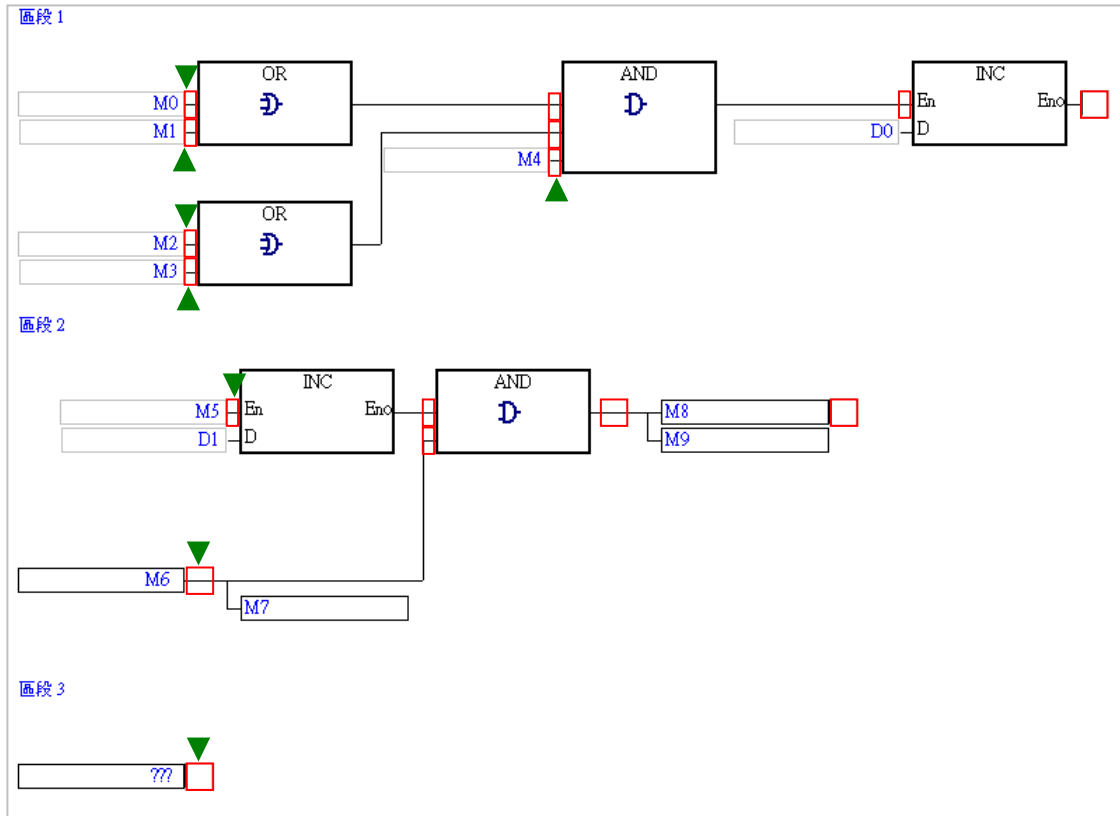
EX2：在 FBD 的程式中，比較接點的前方無法再串接其他元件，而其輸出則可串接至 AND/OR 方塊的輸入端點或直接串接輸出節點，亦或是連接至功能塊或 API 應用指令的 En 端點。



*.關於功能塊的詳細介紹請參考第 7 章的內容。

● 插入 API 應用指令、比較接點與功能塊

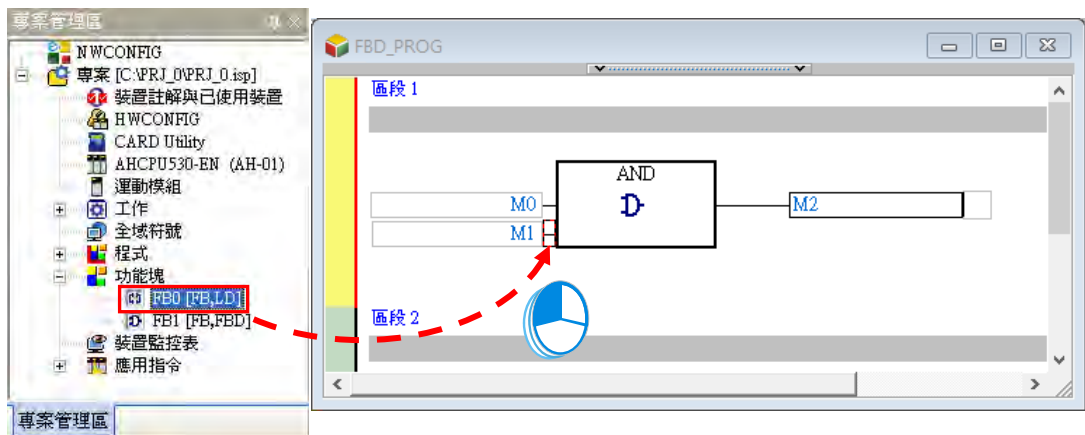
- (1) 首先確認欲插入 API 應用指令、比較接點與功能塊的位置，而下圖紅框的位置則都允許插入 API 應用指令與功能塊，但比較接點則只可插在下圖中有綠色三角形標誌的位置。



- (2) 在 FBD 的編輯環境中，針對插入 API 應用指令、比較接點與功能塊的操作提供了多種方式。

➢ 方法一

於專案管理區中展開應用指令或功能塊的目錄，在其中找到欲插入的項目後，於該項目上按住滑鼠左鍵，之後再將其拖曳至欲插入的位置後放開左鍵即可。




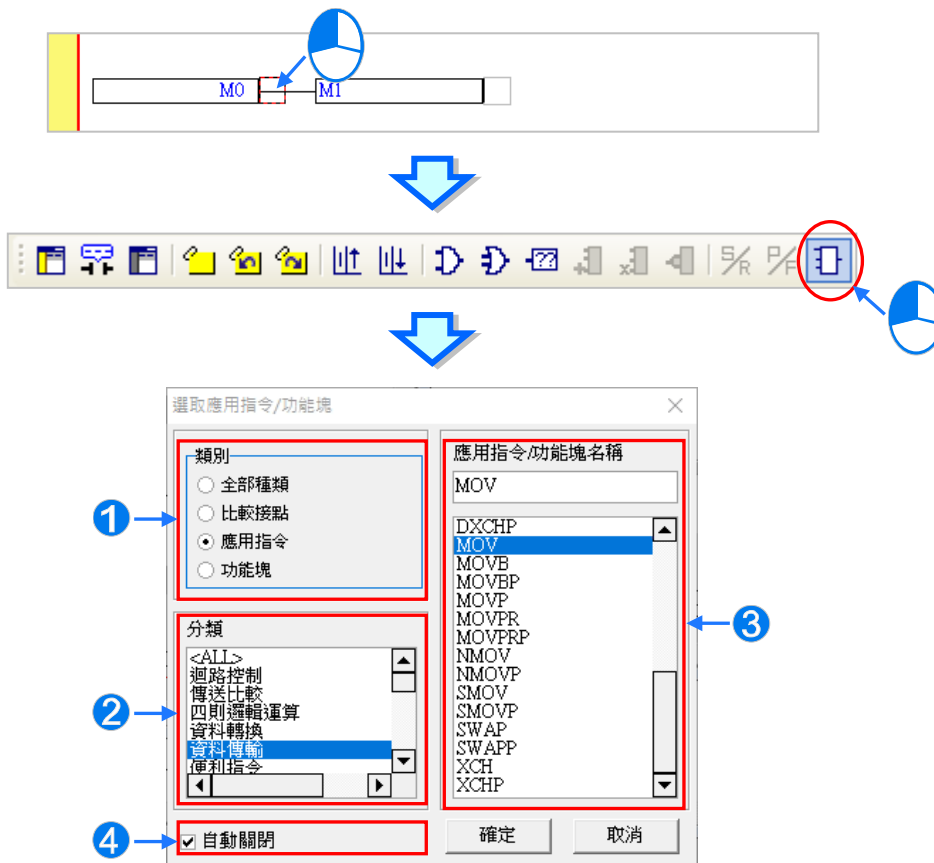
拖曳功能塊時須注意只能拖曳功能塊的定義，而不能拖曳其下層的功能塊實例項目。

11



➤ 方法二

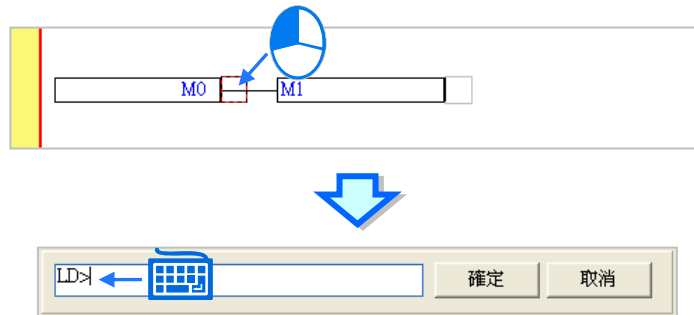
先於欲插入 API 應用指令、比較接點或功能塊的位置上點擊滑鼠左鍵，之後於工具列中點選  圖示，接著再於出現的選取視窗中點選要加入的項目，完成後按下「確定」鍵即可。



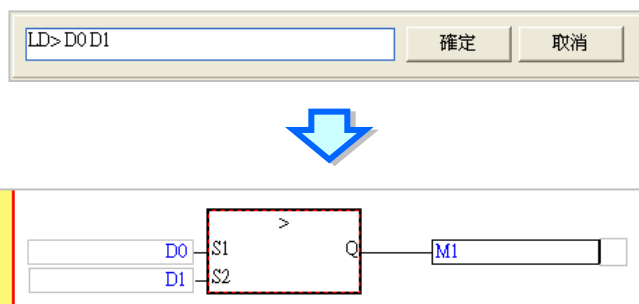
- ① 先選取欲加入的類別。
- ② 若欲加入的是 API 應用指令（函式庫）時，請再於此處選擇指令分類。
- ③ 於此選取 API 應用指令、比較接點或功能塊。
- ④ 若取消此處的勾選，之後便可利用此視窗連續加入指令或功能塊。

方法三

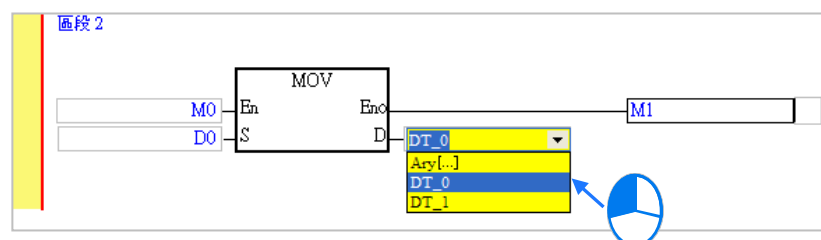
先於欲插入 API 應用指令、比較接點或功能塊的位置上點擊滑鼠左鍵，之後直接以鍵盤輸入功能塊定義、比較接點指令或 API 應用指令的名稱（無須分大小寫），此時畫面便會出現指令編輯模式的輸入欄位，輸入完畢後按下鍵盤的【Enter】鍵或以滑鼠點擊視窗的「確定」鈕即可。



若欲插入的是功能塊，則輸入時只可輸入功能塊定義的名稱，而不可加上功能塊端點的運算元；但若欲插入的是比較接點或 API 應用指令，則輸入時可依照該指令的 IL 語法一併輸入指令的運算元，且輸入的運算元數量無須完整，系統將會依序將其插入對應的運算元欄位，並讓缺少的運算元欄位維持「???'的顯示；若輸入的運算元包含未經宣告的變數符號時，確認輸入之後，系統亦會自動導出符號宣告視窗。關於 IL 指令的語法請參考各機種的指令手冊。



(3) 成功插入 API 應用指令、比較接點或功能塊之後，請再逐一完成每個運算元的編輯。

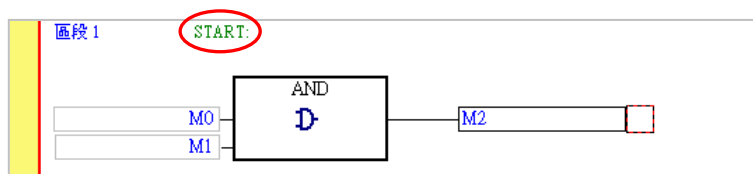


*自動導出符號宣告視窗的功能可於「選項」當中進行設定，設定方式請參考第 2.3.1 節的說明。

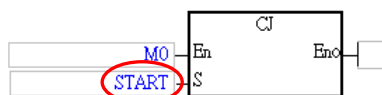
11.2.8 設置標籤 (Label)



當要使用跳躍指令，如 CJ、JMP...時，便必須為跳躍的目標區段設置一個標籤 (Label)，而標籤的位置則位於每個區段編號的右側。若選擇的專案機種為 DVP 系列機種或 AH/AS 系列機種時，因其 P 裝置是由系統自行配置，因此使用者不可直接使用 P 裝置，而必須輸入一個不重複的標籤符號名稱，且標籤符號不需於符號表中宣告。關於 P 裝置的用法與限制請參考各機種的使用手冊。

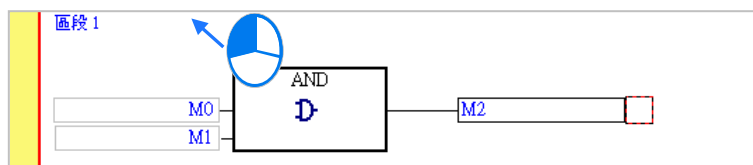


要在指令中使用標籤 (Label) 時，使用標籤符號無須再加上冒號『：』。

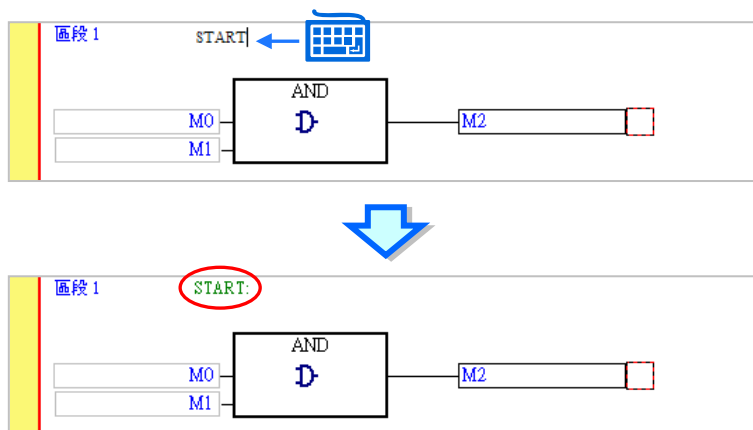


下列則為設置標籤 (Label) 的操作方式。

- (1) 在欲設置標籤的區段編號右側點擊滑鼠左鍵。

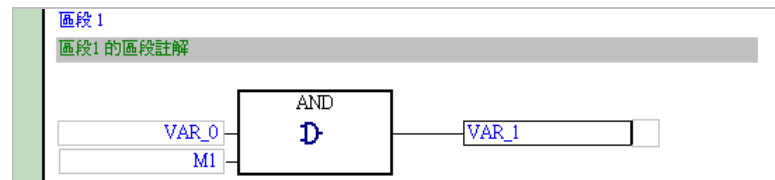


- (2) 若為 DVP 或 AH/AS 機種則請輸入標籤符號名稱；而輸入時，標籤後的冒號『：』可直接省略，系統之後會自行將其補上。完成之後按下【Enter】鍵即可。




11.2.9 註解與提示

如同階梯圖 (LD) 的編輯環境，於 FBD 中同樣可替每個功能塊圖區段加上區段註解，而區段註解則位於每個功能塊圖區段的編號下方。

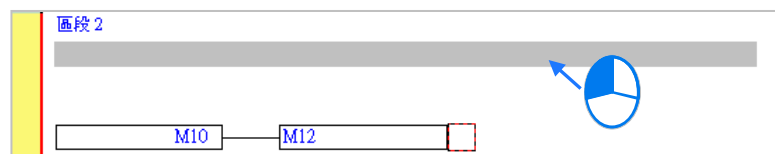


● 編輯區段註解

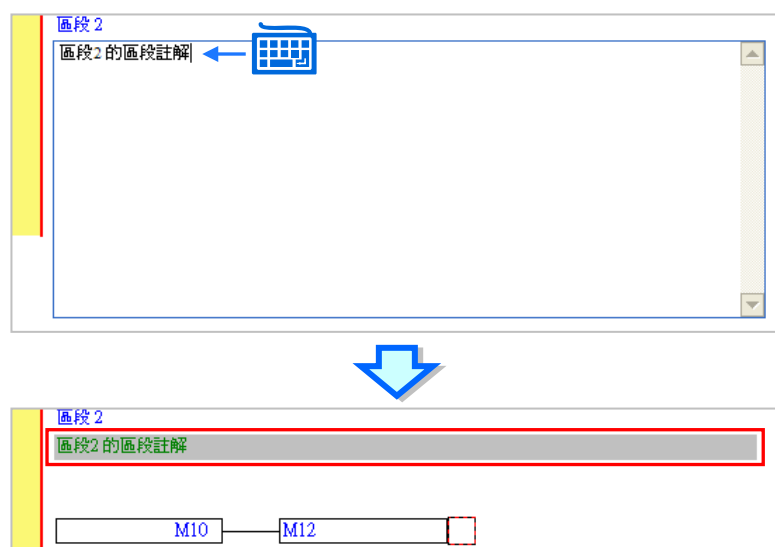
- (1) 當功能塊圖工具列中的  圖示為按下的狀態時，於每個區段編號的下方會顯示該區段的區段註解；而同樣的，當要隱藏區段註解時，請直接將該圖示恢復為未按下的狀態即可。




- (2) 於區段註解的欄位上點擊滑鼠左鍵，之後便會出現區段註解的輸入欄位。

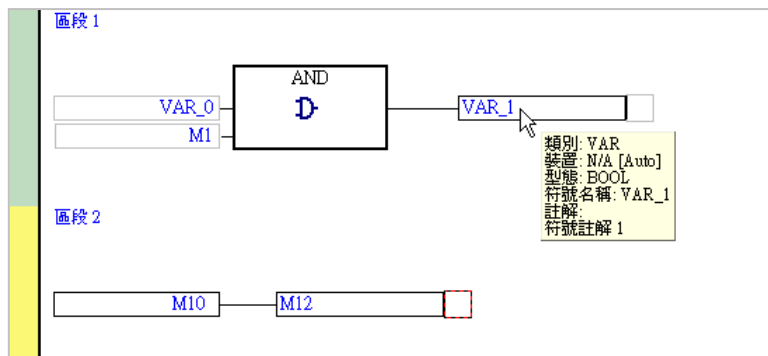


- (3) 接著請直接在出現的編輯欄位中編輯註解，其中按下鍵盤的【Shift】+【Enter】即可換行輸入；輸入完畢後，直接按下鍵盤的【Enter】鍵或用滑鼠點擊他處即可。

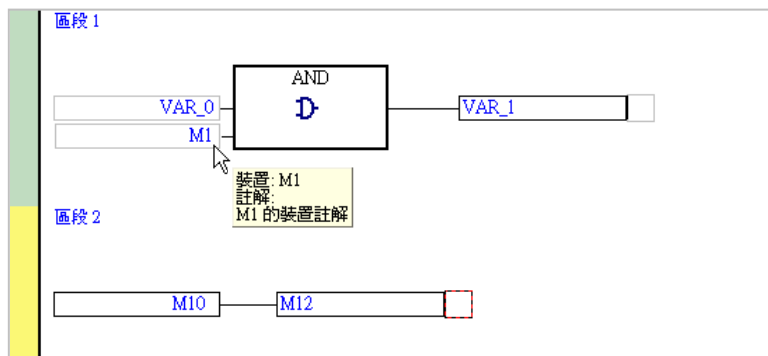


● FBD 的裝置與符號提示

在 FBD 的圖形中，裝置註解與符號註解並不會直接顯示於程式畫面中，而要檢視註解時，請確認圖示工具列中的  圖示為按下的狀態，之後再將滑鼠移至欲檢視註解的位置稍作停留，此時畫面便會出現該裝置或變數符號的相關提示，而提示的內容則可於 ISPSOft 的「選項」功能中進行設定，請參考第 2.3.1 節的相關說明；同樣的，當要關閉提示時，請直接將該圖示恢復為未按下的狀態即可。




除此之外，因裝置與變數符號在 ISPSOft 中是屬於不同的物件，因此將滑鼠指向變數符號與指向裝置時所顯示的提示與註解內容亦會有所不同。




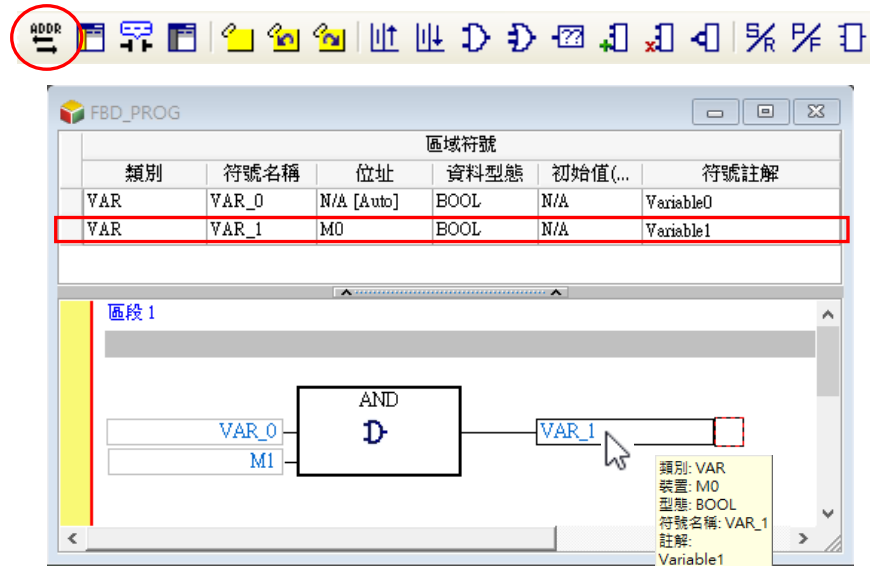
補充說明


當要編輯裝置註解時，亦可利用 ISPSOft 所提供的裝置註解管理工具；於該工具中，使用者可透過集中管理的方式對每個裝置的註解進行編輯，詳細的操作方式則將於第 17.2 節當中另行介紹。

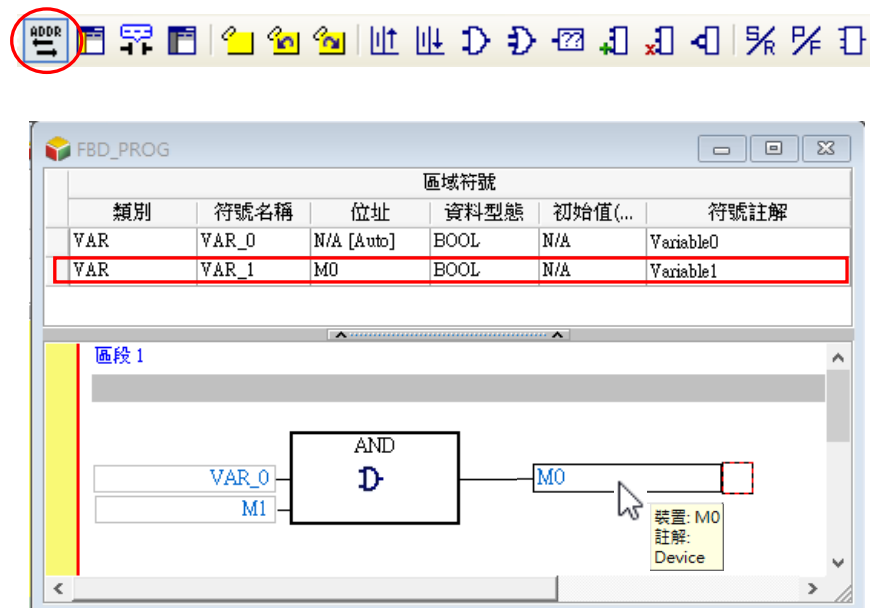
11.2.10 符號模式與位址模式的顯示切換

在 LD 與 FBD 的編輯環境中，當某個變數符號在宣告時有一併指定裝置位址時，藉由切換圖示工具列的  圖示便可變更該變數符號的顯示模式；但未指定裝置位址的變數符號，或未被變數符號所指定的裝置位址則會保持原狀。

如下圖，當  圖示呈現未按下的狀態時，代表目前的顯示方式為**符號模式**，此時有指定裝置位址的變數符號便會以**符號名稱**來顯示，而提示的內容亦會顯示為變數符號的提示。





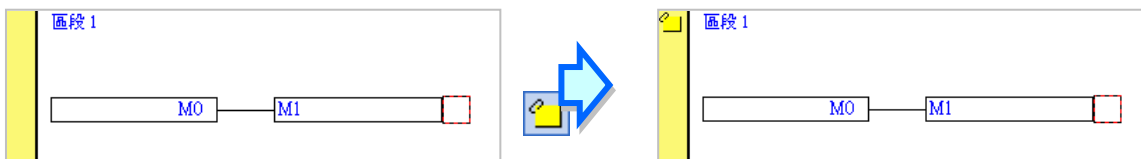
如下圖，當  圖示被按下之後，便代表目前的顯示方式為**位址模式**，此時有指定裝置位址的變數符號便會以**裝置名稱**來顯示，而提示的內容亦會顯示為裝置的提示。





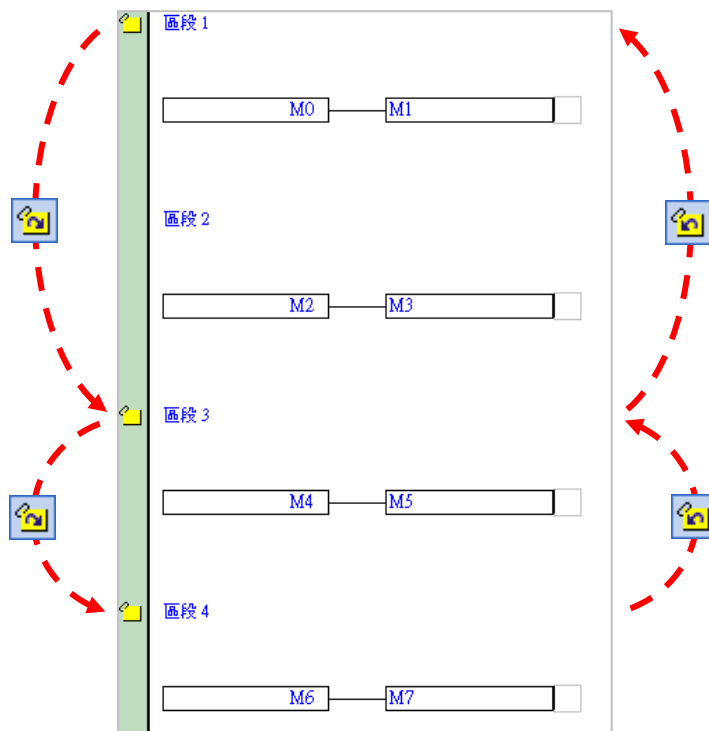
11.2.11 書籤功能

書籤的功能主要在於協助使用者迅速搜尋或移動至加入書籤標註的功能塊圖區段。

- (1) 點擊圖示工具列的  圖示便可於目前選取的區段中加入一書籤；若要取消書籤時，只要選取該區段後，再次按下  圖示即可。



- (2) 當程式中有插入書籤時，按下工具列圖示  即可移動至下一個書籤的位置，而按下  圖示則可移動至前一個書籤的位置。



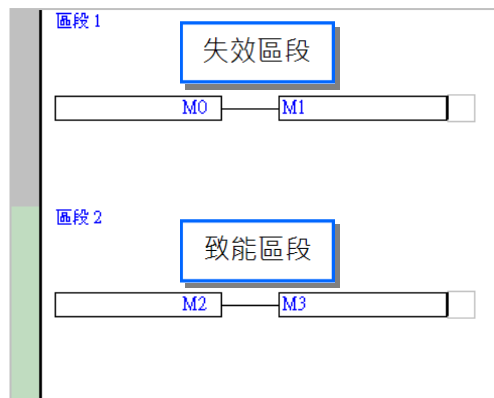
補充說明

於功能工具列的 **編輯 > 書籤** 項目中亦有相同的操作項可供使用，其中點選「**移除所有書籤**」項目則可同時將所有的書籤移除，不過此操作僅會針對目前正在編輯的視窗產生作用。

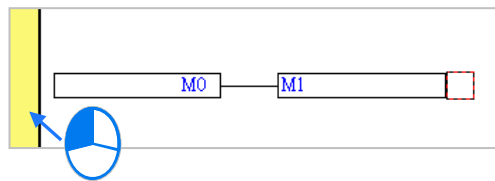
11.2.12 功能塊圖區段的致能與失效


若某個功能塊圖區段被設為失效狀態，則程式的編譯動作便會主動略過該區段，因此當使用者在進行測試或除錯的工作時，利用此功能便可暫時關閉程式中的某些程序。

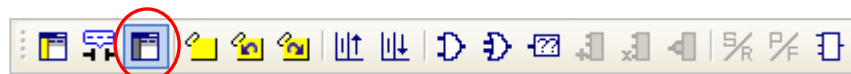
設為**致能**或**失效**的區段，在該區段的左側會使用不同的顏色來表示，而顯示的顏色則可於 ISPSoft 的「選項」功能中進行設定，請參考手冊中第 2.3.1 節的相關說明。



要變更區段的狀態時，請先將該區段選取起來，而選取時亦可同時選取多個區段。



於功能工具列中依序點選 **編輯 (E)** > **(致能/失效) 區段**，或是在選取的區段上按下滑鼠右鍵，接著於快捷選單中選取「**(致能/失效) 區段**」，另外也可直接於圖示工具列中點擊  圖示，之後便可切換被選取區段的致能或失效狀態。



當要將所有的區段都恢復為致能狀態時，請於功能工具列中點選 **編輯 (E)** > **失效區段全部取消**，不過此操作僅會針對目前正在編輯的視窗產生作用。

MEMO



第12章 指令列表 IL

目錄

12.1 指令列表 (IL) 簡介.....	12-2
12.1.1 IL 程式的結構.....	12-2
12.1.2 IL 的功能塊 (FB) 呼叫指令.....	12-4
12.1.3 IL 編程的注意事項.....	12-5
12.2 於 ISPSOft 中建立 IL 程式.....	12-6
12.2.1 ISPSOft 的 IL 編輯環境.....	12-6
12.2.2 編輯 IL 指令.....	12-7
12.2.3 插入 API 應用指令與功能塊.....	12-8
12.2.4 書籤功能.....	12-10
12.2.4.1 加入書籤.....	12-10
12.2.4.2 取消書籤.....	12-11
12.2.4.3 移至書籤位置.....	12-12

12.1 指令列表 (IL) 簡介

指令列表 (**Instruction List · IL**) 是一種類似組合語言的編程方式，在早期 PC 尚未十分普及的年代，PLC 的編程方式便是利用一種稱為**書寫器**的裝置將控制指令逐一的輸入 PLC 當中，而當時所使用的語言便是**指令列表 (IL)**。

12.1.1 IL 程式的結構

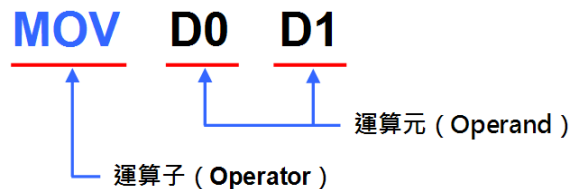
IL 程式是由一行一行的指令敘述建構而成，而其中的每一行指令便代表一個動作，因此要建構一個完整階梯圖區段所能達成的任務時，以 IL 而言便可能須要編輯多行的程式指令才可達到。

下列即為一個簡單的 IL 程式範例，內容中共有 5 行的程式碼，而右側的 LD 示意圖則用來表達每行 IL 指令對應於階梯圖程式所代表的意義。

LD M0：載入 M0 的狀態值	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
OR M1：加入 M1 的值做 OR 運算	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
AND M2：再加入 M2 的值做 AND 運算	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
OUT M3：將目前的運算結果輸出至 M3	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	
(*Note*)：此行為程式註解，因此編譯時會自動略過。	
<pre>0001 LD M0 0002 OR M1 0003 AND M2 0004 OUT M3 0005 (*Note*)</pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">無動作</div>

IL 的指令結構可拆解為**運算子 (Operator)**與**運算元 (Operand)**的組合，其中運算子是代表所要執行的動作，而運算元則是參與執行動作的成員或對象，且其型態依指令格式可以是裝置、變數符號或常數；此外，在 IL 語法中，各個運算元之間必須使用空白來區隔。

如下列範例，其中 **MOV** 代表要執行的是資料搬移的動作 (運算子)，而 **D0** 與 **D1** 則分別是參與搬移動作的來源與目的裝置 (運算元)。



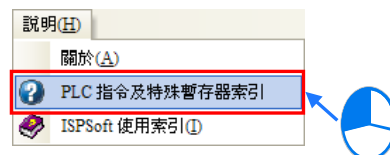
12

在 IL 語法中，被框在 (* 與 *) 符號之間的文字都會被視為註解，而其內部不論包含任何指令或關鍵字，當進行編譯時，系統都會自動略過這段內容；此外，只要在不破壞指令格式的前提下，註解可被加在程式中的任何位置，但須留意程式的可讀性。如下圖，在下列範例中的所有註解都是合法的。

```

0001 LD M0 (*Note*)
0002 (*Note*) OR M1
0003 AND (*Note*) M2
0004 OUT M3
0005 (*Note 1
0006 Note 2
0007 Note 3*)
  
```

IL 的指令可分為基本指令與 API 應用指令，相關指令的格式與用法在各機種的指令手冊中已有詳盡的說明，本章將不再多加敘述；而在編輯過程中要確認指令的用法時，可於 ISPSOft 的功能工具列中點選 **說明 (H)** > **PLC 指令及特殊暫存器索引**，之後系統便會自動開啟說明文件。



此外，若點選某個指令後再按下鍵盤的【F1】鍵，則系統便會在開啟文件後直顯示該指令的頁面。

```

0001 LD M0
0002 OR M1
0003 AND M2
0004
0005 MOV D0 D1
0006
  
```

➔

【F1】

PLC 指令及特殊暫存器索引

API	指令碼	運算元	功能
0300	D MOV P	S · D	資料移動

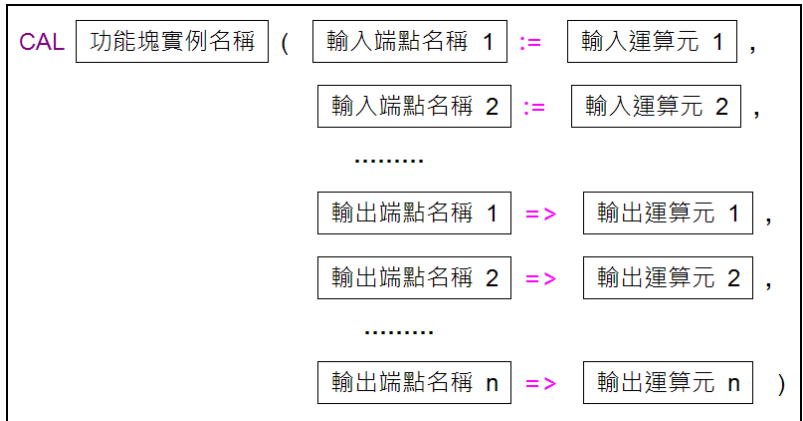
裝置	X	Y	M	S	T	C	H	D	L	SM	SR	E	PR	K	16#	"\$"	F
S	●	●			●	●	●	●	●	●	○	○	○	○	○	○	○
D	●	●			●	●	●	●	●	○	○	○	○	○	○	○	○

脈波軌	16 位元指令 (5 steps)	32 位元指令 (5 steps)
行型	AH500	AH500

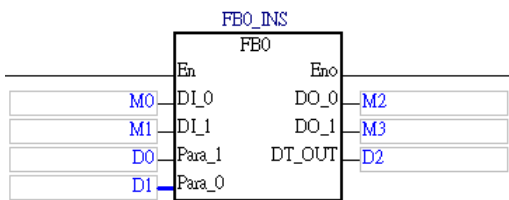
12.1.2 IL 的功能塊 (FB) 呼叫指令

下列則為在 IL 程式中呼叫功能塊的指令格式與注意事項。

12



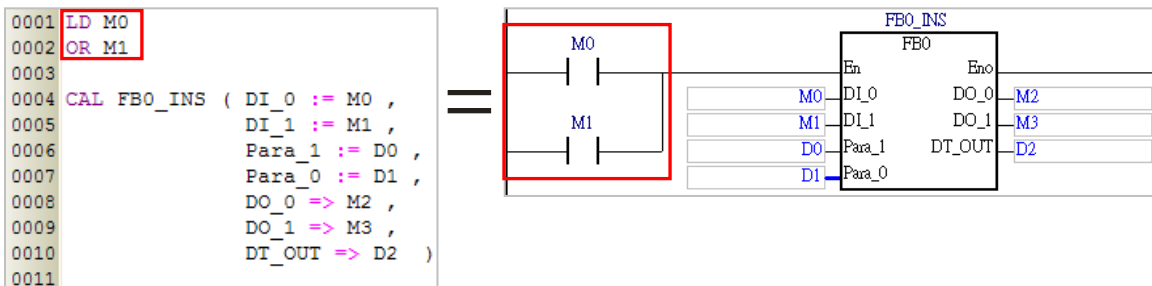
- 除了 **CAL** 指令、**功能塊實例名稱**與**左括號**須位於同一行之外，其餘部分皆可任意換行。
- 功能塊的每個輸入輸出端點與其對應的運算元都必須被包含在括號之中，且每組端點與運算元之間須以逗號隔開，而可用的運算元型態則必須依據各功能塊的定義來加以指定。
- **CAL** 指令之後的名稱應為**功能塊實例**而非**功能塊定義**，亦即應輸入被宣告成**功能塊型態**的變數符號名稱。關於功能塊定義與與功能塊實例的詳細說明請參考**第 7 章**的相關內容。
- 輸入側端點所使用的指定符號為「:=」，而輸出側端點所使用的指定符號則是「=>」。
- 結構內的端點與運算元順序必須依照各功能塊的定義，而不可任意變動，其順序則為排除 **En/Eno** 端點後，先由上而下列出所有輸入側的端點，其中 **VAR_IN_OUT** 類別的端點則被歸類為輸入側；接著再由上而下列出所有輸出側的端點。請參考下列功能塊的階梯圖圖示與 **IL** 指令。



```

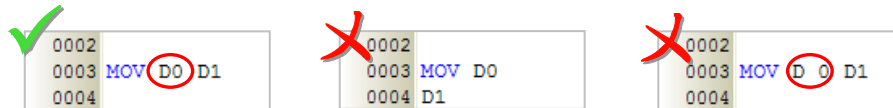
0003 CAL FBO_INS ( DI_0 := M0 ,
0004                DI_1 := M1 ,
0005                Para_1 := D0 ,
0006                Para_0 := D1 ,
0007                DO_0 => M2 ,
0008                DO_1 => M3 ,
0009                DT_OUT => D2 )
    
```

- 在功能塊指令結構的內部並沒有任何 **En** 端點的指定動作，因此當執行至該指令時，系統便會以執行至該指令之前的邏輯運算狀態做為功能塊，致能與否的判斷條件。

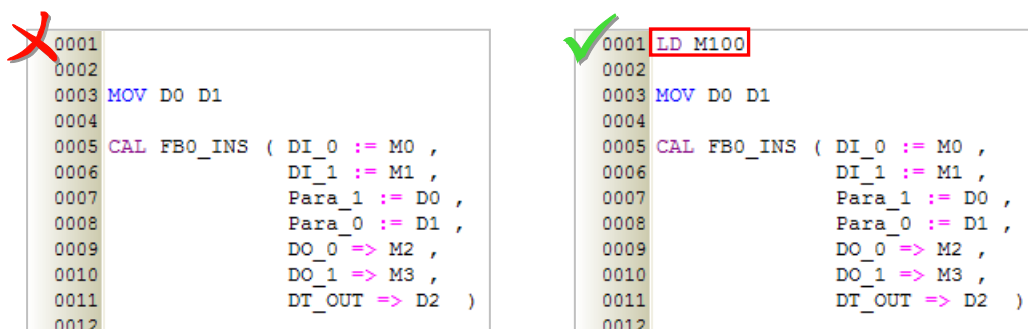


12.1.3 IL 編程的注意事項

- 除功能塊指令之外，在 IL 程式中的每行文字都會被視為一個指令，因此運算元與運算子之間不可有換行，否則便會被視為兩個不完整的指令。此外，於**指令名稱**、**功能塊名稱**或**運算元名稱**的字串之間不可插入空白，否則進行編譯時將無法識別。



- 在不破壞指令格式與完整性的前提下，任何的空白或換行都是被允許的。
- 全形與半形文字在 ISPSOft 中會被視為不同的字元，因此請盡量避免使用全形文字以免造成識別及解讀上的誤判。
- 在 IL 的語法中，英文無大小寫的分別，亦即 OUT、Out 與 out 的意義是相同的。
- 要在 IL 程式當中使用常數時，其表達的格式如下。
 - 10 進制：23456 (未加任何標註的數字會被視為 10 進制)
 - 16 進制：16#5BA0 (數字前方加上 16#)
 - 8 進制：8#55640 (數字前方加上 8#)
 - 2 進制：2#101101110100000 (數字前方加上 2#)
 - 字串："XYZ" (以雙引號包住字串)
 - BOOL：須使用常開/常閉接點 SM400/SM401 (AH/AS) 或 M1000/M1001 (DVP) 來表達。
- 使用陣列時，其輸入格式為 **符號名稱[索引]**，且因 IL 尚不支援變數型態的索引值，因此索引值只可使用小於陣列大小的 10 進制常數，且必須由 0 開始。
- 當使用功能塊或 API 應用指令時，若在下達指令之前沒有任何 LD 指令或邏輯運算的區塊，則執行編譯時便會產生錯誤。

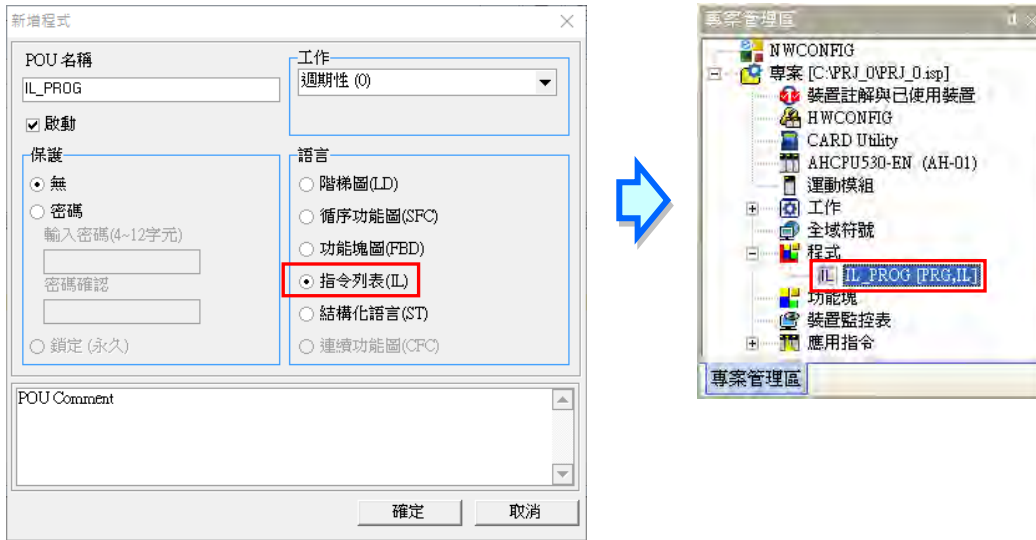


- IL 語言可編輯的行數並沒有限制，但仍需考慮編譯後的程式大小是否會超出主機的記憶體容量。
- ISPSOft 的 IL 編輯環境支援複製/剪下/貼上的功能，因此使用者可從其他的文字編輯器中將編輯好的內容複製到 ISPSOft 的 IL 編輯工作區中。

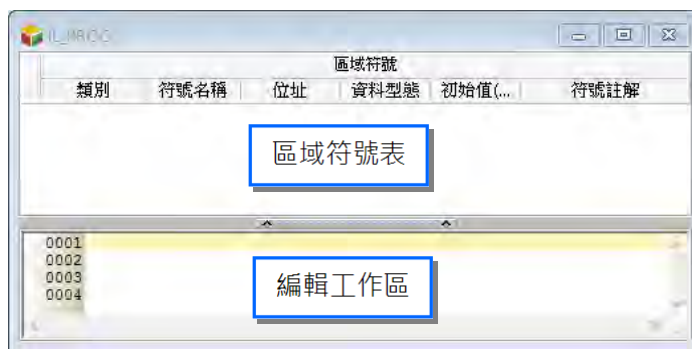
12.2 於 ISPSOft 中建立 IL 程式

12.2.1 ISPSOft 的 IL 編輯環境

在建立 POU 時，於語言欄位選擇「指令列表 (IL)」便可建立 IL 程式，請參考第 5.4.1 節的說明。



下圖即為 ISPSOft 中的 IL 編輯環境，其視窗配置的上方為區域符號表，而下方則是編輯工作區，且在編輯工作區的左側預設會有行號顯示，使用者可自行設定其顯示的格式，詳細操作請參考第 2.3.1 節的相關說明。在進入 IL 的編輯環境後，視窗中亦會出現對應的圖示工具列，其功能如下。



圖示	鍵盤 (快捷鍵)	功能說明
	Shift + Ctrl + B	在目前選取的行列上加入或取消書籤標記
	Shift + Ctrl + P	移至前一個書籤的標記位置
	Shift + Ctrl + N	移至下一個書籤的標記位置
	Shift + Ctrl + U	插入指令或功能塊

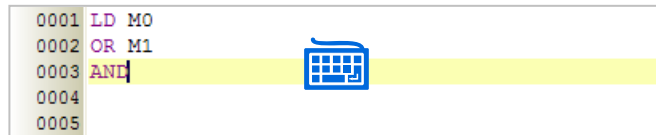
12.2.2 編輯 IL 指令

在 IL 的編輯工作區中，其操作方式與一般的文字編輯器類似，編輯時只要依據指令格式直接使用鍵盤在工作區上面輸入或修改文字即可，且當按下鍵盤的【Enter】鍵時便可執行換行的動作。

```

0001 LD M0
0002 OR M1
0003 AND
0004
0005

```

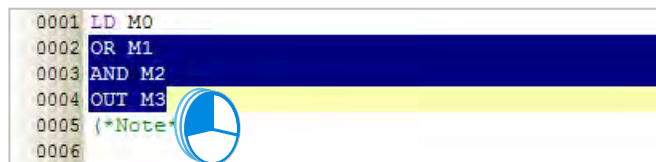


選取文字的方法也與一般的文字編輯器相同。先將滑鼠點在選取區塊的起點並按住滑鼠左鍵，之後往選取的方向移動便可框選文字；又或者是先用滑鼠點選在某個區塊的起點，接著按住鍵盤的【Shift】鍵，之後再用滑鼠點選該區塊的終點便可將整段區塊選取起來。

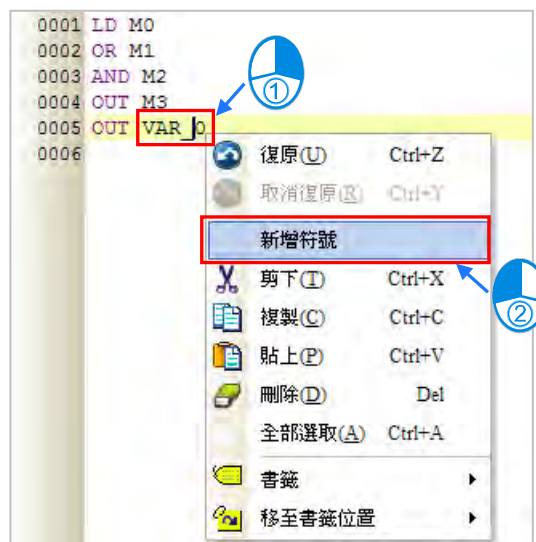
```

0001 LD M0
0002 OR M1
0003 AND M2
0004 OUT M3
0005 (*Note
0006

```



當在編輯的過程中輸入了未經宣告的變數符號時，輸入完畢後用滑鼠點擊該文字並按下右鍵，之後再於快捷選單中點選「新增符號」便可自動導出變數符號的宣告視窗。



符號名稱	位址	資料型態	初始值(下載時生效)	註解...
VAR_0		WORD		

類別: VAR 自動關閉視窗 插入 宣告在全域符號表

12.2.3 插入 API 應用指令與功能塊

插入 API 應用指令與功能塊的方法有下述三種。

● 方法一

依據應用指令與功能塊的格式，直接於編輯工作區中利用鍵盤輸入即可。

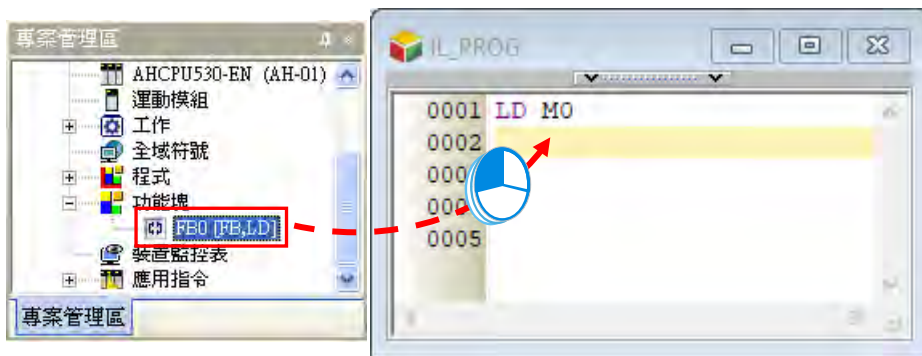
```

0001 LD MO
0002 CAL FBO_INS (
0003     TRIG := MO ,
0004     DT_IN := DO ,
0005     DT_OUT => D1
0006 )
0007
0008

```

● 方法二

於專案管理區中展開應用指令或功能塊的目錄，在其中找到欲插入的項目後，於該項目上按住滑鼠左鍵，之後再將其拖曳至欲插入的位置後放開左鍵即可。



插入應用指令或功能塊之後，請再依據其定義的格式，直接於插入的樣板中編輯對應的運算元。

```

0001 LD MO
0002 CAL ????? (
0003     TRIG := ?????,
0004     DT_IN := ?????,
0005     DT_OUT => ?????
0006 )
0007

```

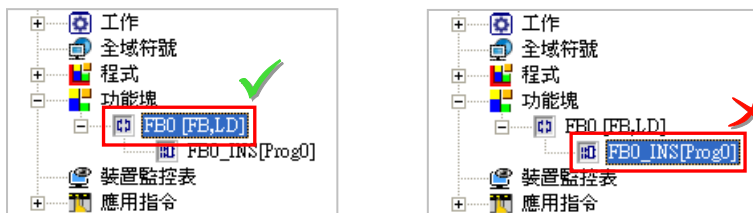
➡

```


0001 LD MO
0002 CAL FBO_INS (
0003     TRIG := MO ,
0004     DT_IN := DO ,
0005     DT_OUT => D1
0006 )
0007

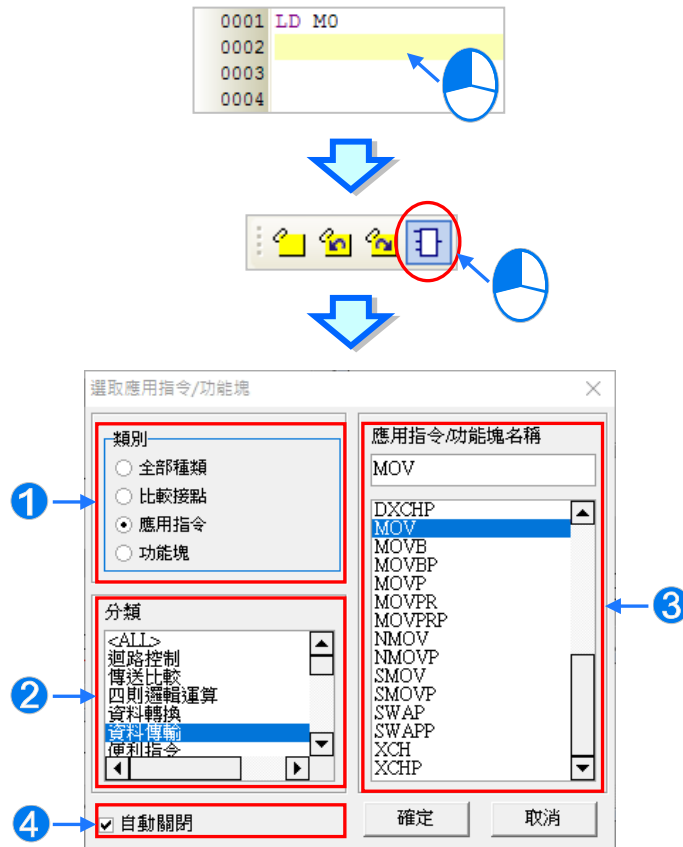
```

除此之外，拖曳功能塊時須注意只能拖曳功能塊的定義，而不能拖曳其下層的功能塊實例項目。



● 方法三

先於欲插入 API 應用指令或功能塊的位置上點擊滑鼠左鍵，之後於圖示工具列中點選  圖示，接著再於出現的選取視窗中點選要加入的項目，完成後按下「確定」鍵即可將其插入指定位置。



- ① 先選取欲加入的類別。
- ② 若欲加入的是 API 應用指令 (函式庫) 時，請再於此處選擇指令分類。
- ③ 於此選取 API 應用指令、比較接點或功能塊。
- ④ 若取消此處的勾選，之後便可利用此視窗連續加入指令或功能塊。

接著請再依據應用指令與功能塊的格式，直接於插入的樣板中編輯對應的運算元。



12.2.4 書籤功能

書籤的主要功能在於協助使用者迅速移動至加入書籤的行列，而每個 IL 程式至多可加入 10 個書籤。


12.2.4.1 加入書籤

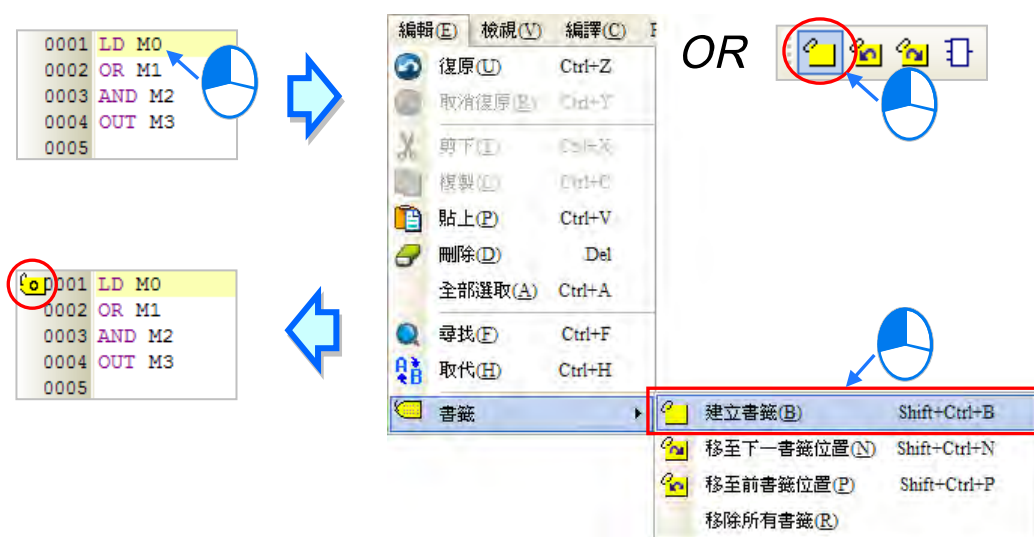
● 方法一

先用滑鼠點選欲加入書籤的行列後按下滑鼠右鍵，接著於快捷選單中點選「書籤」項目，之後再於該項目中選擇欲加入的書籤編號即可。不過若選擇的編號已使用於其他行列，則該編號的書籤仍會被搬移至目前選取的行列上。



● 方法二

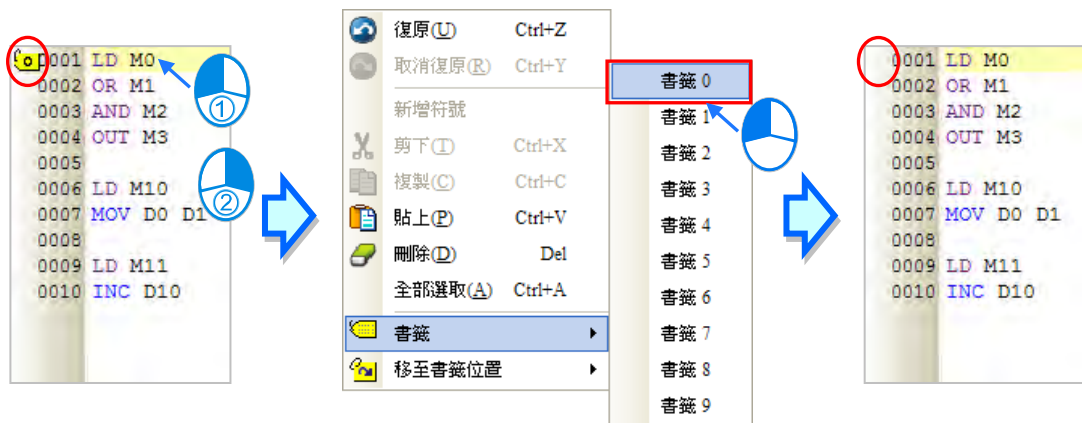
首先要在要加入書籤的行列上點擊一下滑鼠左鍵，接著於圖示工具列中點擊  圖示，或於功能工具列中依序點擊 **編輯 (E)** > **書籤** > **建立書籤 (B)** 即可。不過使用此方法建立的書籤編號只能按照順序遞增，使用者無法自行指定。




12.2.4.2 取消書籤

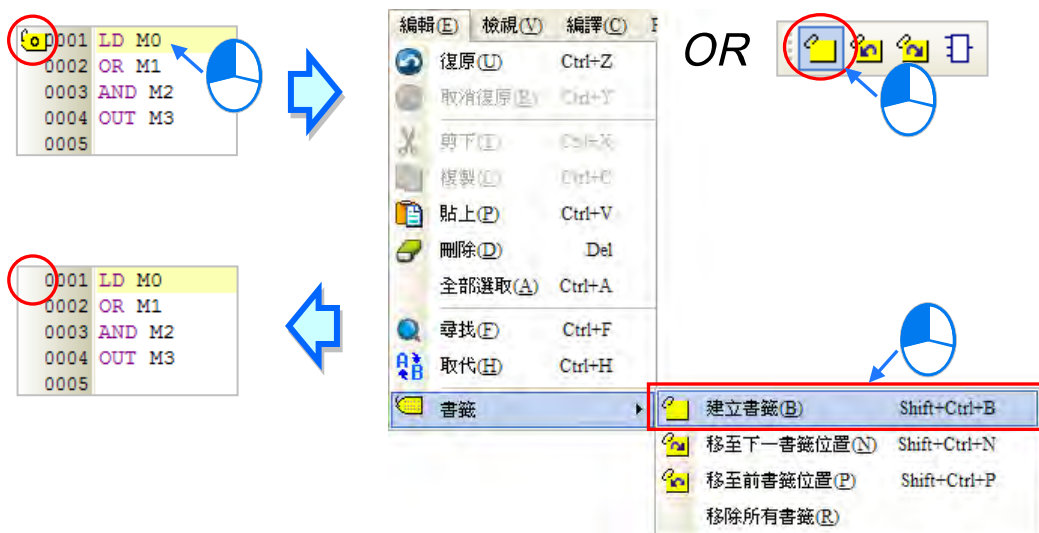
● 方法一

先用滑鼠點選欲取消書籤的行列後按下右鍵，接著於快捷選單中點選「書籤」項目，之後再於該項目中點選與所選行列相同編號的書籤即可。不過若選擇的是其它編號時，所選行列的書籤便會被變更為該編號，且若此編號的書籤已使用於其他行列，則該書籤仍會被搬移至目前所選取的行列上。



● 方法二

首先在要取消書籤的行列上點擊一下滑鼠左鍵，接著於圖示工具列中點擊  圖示，或於功能工具列中再次點擊 編輯 (E) > 書籤 > 建立書籤 (B) 即可。但若於功能工具列中點擊的是 編輯 (E) > 書籤 > 移除所有書籤 (R)，則目前編輯視窗中的所有書籤便會被全部取消。





12.2.4.3 移至書籤位置

● 方法一

於編輯工作區中按下滑鼠右鍵，接著於快捷選單中點選「移至書籤位置」項目，之後在於目錄中選擇移動目標的書籤編號即可。



● 方法二

當程式中有插入書籤時，於功能工具列的 **編輯 (E)** > **書籤** 項目下，或於圖示工具列中按下  圖示即可移動至下一個書籤的位置，而按下  圖示則可移動至前一個書籤的位置。



第13章 結構化語言 ST

目錄

13.1 認識結構化語言 ST.....	13-3
13.1.1 ST 程式的基本結構	13-3
13.1.2 敘述句 (Statement)	13-4
13.1.3 表達式 (Expression)	13-5
13.1.4 運算元 (Operand) 與運算子 (Operator)	13-6
13.1.5 關鍵字 (Keyword) 與註解 (Comment)	13-7
13.1.6 於 ST 程式中使用陣列符號	13-8
13.1.7 ST 編程的注意事項	13-9
13.2 ISPSOft 中的 ST 敘述句結構	13-11
13.2.1 指定結構 :=	13-11
13.2.2 條件結構 IF	13-13
13.2.3 條件結構 CASE.....	13-15
13.2.4 迴圈結構 REPEAT.....	13-17
13.2.5 迴圈結構 WHILE	13-18
13.2.6 迴圈結構 FOR.....	13-19
13.2.7 API 應用指令結構	13-20
13.2.8 功能塊 (FB) 呼叫結構.....	13-21
13.2.9 空白敘述	13-23
13.2.10 RETURN 敘述	13-24
13.2.11 EXIT 敘述	13-24
13.3 於 ISPSOft 中建立 ST 程式	13-26
13.3.1 ISPSOft 的 ST 編輯環境	13-26
13.3.2 編輯 ST 敘述.....	13-27
13.3.3 插入 API 應用指令與功能塊	13-28
13.3.4 書籤功能	13-29
13.4 ST 的程式範例	13-30
13.4.1 範例說明	13-30
13.4.2 硬體規劃.....	13-31
13.4.3 程式規劃.....	13-31

13.4.4 建立範例程式 13-32

13.1 認識結構化語言 ST

結構化語言 (**Structured Text** , **ST**) 是一種類似 C 語言或 PASCAL 等高階語言的編程方式，其最大的特色在於整個程式是由許多的文字敘述所建構而成，而相較於指令列表 (**IL**)，**ST** 則具有較高的可讀性，且可透過類似數學方程式的文法來表達一個複雜的運算式，因此對於一個需要進行繁複運算的程序來說，**ST** 是較適合此種應用場合的開發語言。除此之外，對於習慣編寫 C 語言等高階語言的程式設計者而言，**ST** 語言亦提供了一個較為熟悉的編程環境。

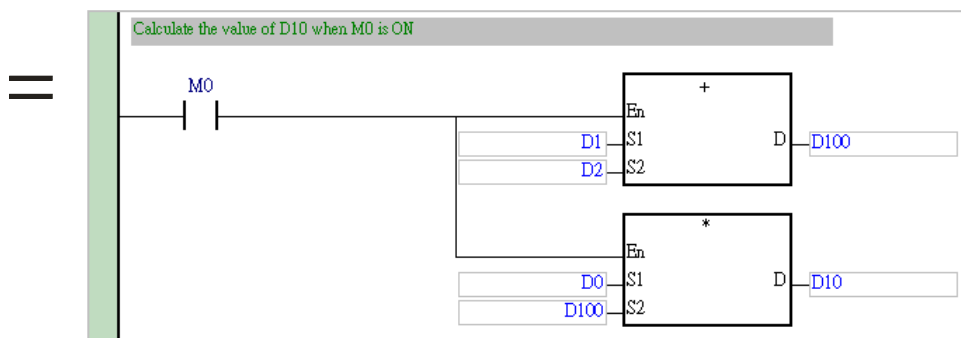
13.1.1 ST 程式的基本結構

下列便是一個很基本的 **ST** 程式範例，它與下方的階梯圖具有同樣的功能，但卻能用更精簡的方式來加以表達，且所展現的文字結構已經近乎人類所使用的語言文法；而在 **ST** 程式當中，這樣具有執行意義的文字結構便稱之為敘述句 (**Statement**)。

```

0001 (*Calculate the value of D10 when M0 is ON*)
0002 IF M0 THEN
0003     D10 := D0*(D1+D2) ;
0004 END_IF ;
0005

```



上述的 **ST** 程式範例可被解析為如下的結構，而一個 **ST** 程式便是由許多這樣的基本結構所組成。

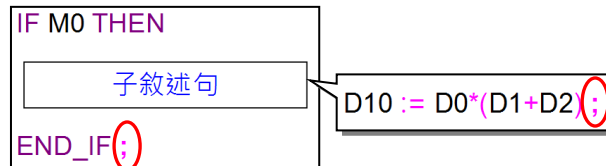
敘述句 (**Statement**)



* 在 **ST** 的語法中，『:=』符號代表指定的意義，而『=』符號則代表等於的意義，此點與 **C** 語言的表達方式不同。

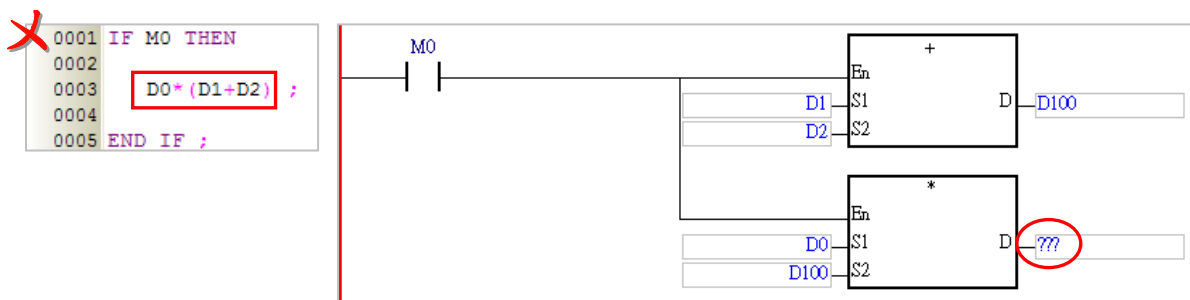
13.1.2 敘述句 (Statement)

敘述句是 ST 語言中最基本的執行單位，它代表的是一個可被完整執行的工作。而如同前述的範例，一個完整敘述句的呈現可能並不侷限在同一行文字當中，但它必定是以分號『；』來做為該敘述句的結尾。另外，一個敘述句中也允許再包含多個或多層的子敘述句，且不論其所處的位置為何，只要是一個完整敘述句的結尾，最後必定會跟隨著『；』符號。



13

一個完整的敘述句相當於一個具備完整功能的階梯圖區段 (NETWORK)，它必須能夠明確的表達一件工作。以上述程式中的子敘述句為例，它所執行的工作便是將各裝置的內容值依照數學式表達的順序 $D0 * (D1 + D2)$ 來進行運算，並將運算的結果指定給裝置 D10；但如下圖紅色框線中的內容便不是一個合法的敘述句，它只是一個表達式 (Expression)，代表的只是一個數學運算的數值，卻不是一件具體的工作，就如同右側缺少輸出運算元的階梯圖程式一樣，並不是一個可被完整執行的程序。



由前述可知，一個 ST 程式是由許多的敘述句所組成，而一個敘述句的結構也同時包含了許多元素；下圖便是一個 ST 的程式範例，而其中每個紅框的位置都代表一個完整的敘述句。

```

0001
0002 IF M0 THEN
0003   M2 := TRUE ;
0004 ELSIF M1 THEN
0005   M2 := FALSE ;
0006 END_IF ;
0007
0008 M20 := M10 ;
0009
0010 M21 := M11 ;
0011
0012 IF M2 THEN
0013   ROR (D0, 2) ;
0014 END_IF ;
0015

```

* 在 ST 的語法中，『:=』符號代表指定的意義，而『=』符號則代表等於的意義，此點與 C 語言的表達方式不同。

13.1.3 表達式 (Expression)

表達式在敘述句結構當中是一個相當重要的組成元素，它代表的是一個「值」，例如 TRUE 或 FALSE 的布林值，亦或是 16 或 -3 的整數值；而表達式視使用的場合，可以是一個運算式或是一個常數，當然也可以是一個變數符號或裝置。下列便是一些表達式的例子。

- M0 & M1 (布林值的表達式)

此表達式所代表的是 M0 與 M1 進行 & 運算後的布林值。

- M0 = FALSE (布林值的表達式)

此表達式所代表的是「M0=FALSE」這個條件是否成立。當 M0 的值為 ON 時，此表達式所代表的布林值便是 FALSE；但若 M0 為 OFF 時，因條件的成立，表達式所代表的布林值便會為 TRUE。

- M0 (布林值的表達式)

此表達式則直接以 M0 的值當作其代表的布林值。當 M0 的值為 ON 時，表達式所代表的布林值便會為 TRUE，而當 M0 為 OFF 時，表達式所代表的布林值則為 FALSE。

- D1 + D2 (數值的表達式)

此表達式代表的數值為 D1 與 D2 相加後的運算結果。

- D0 (數值的表達式)

此表達式則直接以 D0 的現在值做為其代表的數值。

- D2 = D0 + D1 (布林值的表達式)

這是一個相當容易被混淆的布林值表達式，它代表的是「D2=D0+D1」這個條件是否成立。也就是說當 D0 與 D1 相加的結果等於 D2 的現在值時，表達式代表的便是布林值 TRUE；相反的，如果 D0 與 D1 相加的結果不等於 D2 的現在值時，表達式所代表的即是布林值 FALSE。

- D2 := D0 + D1 ; (敘述句，非表達式。)

此為一個完整的敘述句而非表達式，其代表的意義是「將 D0 與 D1 的相加結果指定給 D2」的一個工作；不過這個敘述句也是由「D2」與「D0+D1」這兩個表達式所組成的。

請參考下列範例，此程式主要由兩個 IF 結構的敘述句與其子敘述句所組成。當程式執行至第一個 IF 敘述句時，系統會先判斷表達式「M0」的值是否為 TRUE，若此時裝置 M0 的狀態為 ON 時，系統便會執行「將 D21+D22 的值指定給 D20」的敘述句工作；而執行至第二個 IF 敘述句時，系統會先判斷 $D0 * (D1 + D2)$ 的運算結果是否等於 D20 的值，若兩方相等，則代表表達式「 $D20 = D0 * (D1 + D2)$ 」的布林值為 TRUE，此時系統便會執行「將 M1 的值設為 TRUE」的敘述句工作；但若兩方不相等，系統則會執行「將 M1 的值設為 FALSE」的敘述句工作。

```

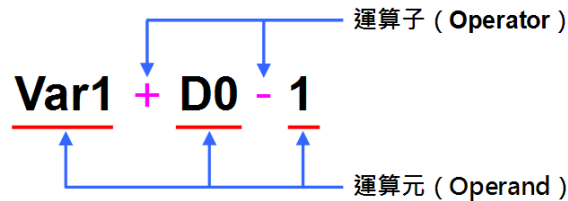
0001 IF M0 THEN
0002     D20 := D21 + D22 ;
0003 END_IF;
0004
0005 IF D20 = D0*(D1+D2) THEN
0006     M1 := TRUE ;
0007 ELSE
0008     M1 := FALSE ;
0009 END_IF;

```


13.1.4 運算元 (Operand) 與運算子 (Operator)

運算元與運算子為組成一個表達式的基本元素。運算元指的是參與運算的對象，而運算子則代表所執行的運算動作，例如表達式 $D0 + D1$ 中， $D0$ 與 $D1$ 都是運算元，而 $+$ 號則是運算子。

由上一節的各個例子可知，表達式可為一群運算元與運算子的組合，但也可單獨的以一個運算元來呈現一個表達式，而其中做為運算元的可為裝置、變數符號或是常數。



如同數學式一般，運算子本身也有執行運算的優先順序，而當優先的等級相同時，進行運算的順序便會是由左至右。下表即為 ISPSOft 中 ST 語法的運算子一覽表。

符號	功能	資料格式		表達式範例		優先等級
		運算元	運算結果 (表達式的值)	表達式	值	
()	區塊優先	不限	不限	$(D0 + 6) * 3$	33	↑ 最高
**	次方	浮點數	浮點數	$2.0 ** 5.0$	3.2E+1	
-	負號	帶號數值	帶號數值	$-(D0 + 3)$	-8	
NOT	邏輯反向	布林	布林	$NOT M0$	TRUE	
*	乘法	任何數值	任何數值	$D0 * 3$	15	
/	除法	任何數值	任何數值	$15 / D0$	3	
MOD	除法取餘數	整數	整數	$D0 MOD 3$	2	
+, -	加法、減法	任何數值	任何數值	$D0 + 3$	8	
<, >, <=, >=	數值比較	任何數值	布林	$D0 > 2$	TRUE	
=, <>	等於、不等於	任何數值	布林	$D0 <> 2$	TRUE	
		布林		$M0 = TRUE$	FALSE	
AND, &	"及"運算	布林	布林	$M0 \& M1$	FALSE	
XOR	"互斥或"運算	布林	布林	$M0 XOR M1$	TRUE	
OR	"或"運算	布林	布林	$M0 OR M1$	TRUE	最低

*1. 在上述的表達式範例中，各裝置的預設值為 $M0=FALSE$ · $M1=TRUE$ · $D0=5$ 。


*2. 使用次方運算子「**」時，其搭配的運算元必須使用常數或宣告為浮點格式的變數符號，而不可直接使用裝置名稱。

13.1.5 關鍵字 (Keyword) 與註解 (Comment)

在程式中具有特殊意義的文字便稱之為關鍵字，例如「**TRUE**」與「**FALSE**」在程式中代表的是一個布林值，而前述範例中的「**IF**」則代表敘述句，所要處理的工作；除此之外，包括 **API 應用指令** 的名稱也會被視為關鍵字，且為了避免程式的解讀錯誤，在程式中被視為關鍵字的文字將無法被宣告為變數符號的名稱，但若只是被包含在符號名稱當中則是允許的，例如「**FIFO**」及「**_IF**」。

在 ST 語法中，被框在 (* 與 *) 和 /* 與 */ 符號之間的文字都會被視為註解，也可以使用 // 雙斜線做單行註解而其內部不論是否包含敘述句、表達式，亦或是任何的關鍵字，當進行編譯時，系統都會自動略過這段內容；除此之外，只要在不破壞文體結構的前提下，註解可被加在程式中的任何位置，但須留意程式的可讀性。


請參考下列範例，其中編輯於左側範例中的所有註解都是合法的，而右側的註解則因為將關鍵字「**END_IF**」切開而破壞其完整性，與單行註解將必要的結尾符號；切斷，因此編譯時便會產生錯誤。



```

0001 (*Note*)
0002
0003 IF (*Note*) M100 = TRUE THEN
0004
0005     D10 (*Note*) := D10 * (D1+D2);
0006
0007 END_IF (*Note*); (*Note*)
0008
0009 (*Note*)
0010
0011 //Note
0012
0013 IF /*Note*/ M100 = TRUE THEN
0014
0015     D10 /*Note*/ := D10 * (D1+D2);
0016
0017 END_IF/*Note*/; /*Note*/
0018
0019 /*Note*/

```

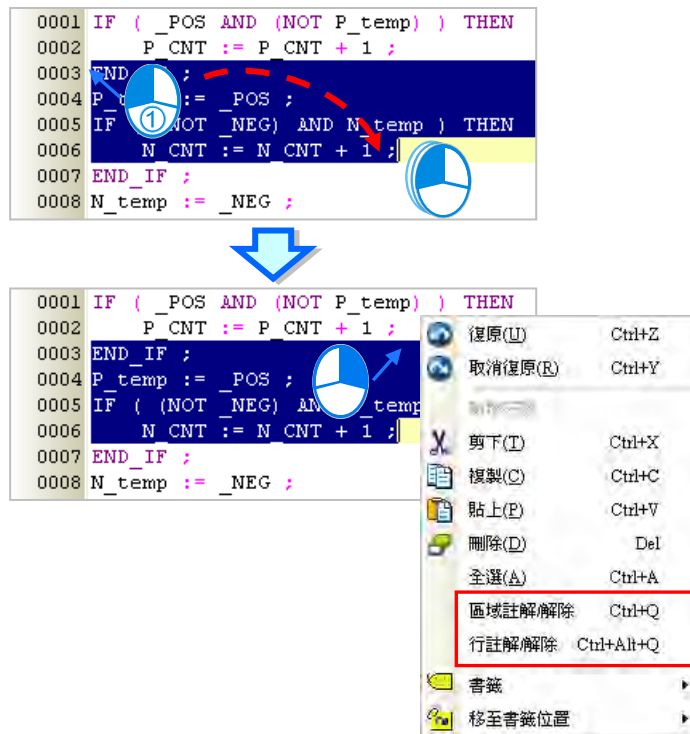


```

0001
0002 IF M100 = TRUE THEN
0003
0004     D10 := D10 * (D1+D2);
0005
0006 END (*Note*)_IF;
0007
0008
0009 IF M100 = TRUE THEN
0010
0011     D10 := //D10 * (D1+D2);
0012
0013 END_/*Note*/IF;
0014

```

ISPSOft 中也提供了將程式轉換為區域註解或行註解的快捷功能。先以滑鼠左鍵，選取一段 ST 語言區塊後，按下滑鼠右鍵開啟快捷選單。



若點選 **區域註解/解除** 後，所選區塊前後便會自動加上註解符號，如下圖左；若點選 **行註解/解除** 後，所選區塊的每一行前後便加上註解符號，如下圖右。

```
0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 (*P temp := POS ;
0005 IF ( (NOT NEG) AND N temp ) THEN
0006     N_CNT := N_CNT + 1 ;
0007 END_IF ;*)
0008 N_temp := _NEG ;
```

區域註解

```
0001 IF ( _POS AND (NOT P_temp) ) THEN
0002     P_CNT := P_CNT + 1 ;
0003 END_IF ;
0004 (*P temp := POS ;*)
0005 (*IF ( (NOT NEG) AND N temp ) THEN*)
0006 (* N_CNT := N_CNT + 1 ; *)
0007 (*END_IF ;*)
0008 N_temp := _NEG ;
```

行註解

13.1.6 於 ST 程式中使用陣列符號

在 ISPSOft 的 ST 程式中使用陣列符號時，其表達格式為 **符號名稱[索引]**，其中索引的部份可為常數或變數符號，不過只有在 AH/AS 系列機種的專案中才可使用變數符號做為索引值；而不論是變數符號或常數，索引值的內容都必須由 0 開始，同時也不可大於或等於陣列的大小，例如大小為 10 的陣列符號，其索引值的範圍則必須為 0~9。關於陣列符號的使用說明請參考第 6.2.3 節的相關內容。

```
0001 IF M0 THEN
0002
0003 Ary_A[0] := Ary_B[IDX] ;
0004
0005 END_IF ;
```

除此之外，使用陣列時還須注意下列幾點事項。

- 使用陣列符號時只可對單一的陣列元素進行操作，而無法將代表整個陣列的變數符號當作運算元。
- 目前 ISPSOft 的陣列型態的變數符號只支援至一維陣列；但自定義資料形態中的結構 (STRUCTURE) 內所定義的元素支援最多三維陣列，此類變數符號可以下形式書寫，如下圖中將 D0 的值寫入自定義資料型態變數符號 vardut 中指定元素 pt1 中三維陣列的第[1,1,1]個。自定義資料型態請參考第 8 章。

```

0001 IF M0 THEN
0002
0003 vardut.pt1[1,1,1]:=D0;
0004
0005 END_IF;
0006

```

- 當陣列索引為變數符號時，若在操作的過程中不慎造成索引值超出合法範圍，此時 PLC 仍會繼續運作且不會有錯誤碼產生，但實際參與運算的資料便可能已產生錯亂。
- 使用常數索引值的陣列符號允許使用間接指定暫存器來進行修飾，但若使用變數符號當作陣列索引時，系統便不允許再加上間接指定暫存器。關於間接指定暫存器的說明請參考第 6.1.5 節。

13

13.1.7 ST 編程的注意事項

- 在不破壞關鍵字或敘述句結構的前提下，任何的換行或空白都是被允許的。
- 原則上全形與半形文字在 ISPSOft 中會被視為不同的字元，因此請盡量避免使用全形文字以免造成識別及解讀上的誤判。
- 在 ST 的語法中，英文無大小寫的分別，亦即 IF、If 與 if 的意義是相同的。(此點與 C 語言不同)
- 在 ST 語言中使用常數時，其表達的格式如下。
 - 10 進制：23456 (未加任何標註的數字會被視為 10 進制)
 - 16 進制：16#5BA0 (數字前方加上 16#)
 - 8 進制：8#55640 (數字前方加上 8#)
 - 2 進制：2#101101110100000 (數字前方加上 2#)
 - 字串：“XYZ” (以雙引號包住字串)
 - BOOL：TRUE 或 FALSE。

- 在 ISPSOft 的 ST 語法中，原則上一個表達式裡的運算元必須為相同的資料型態才可以進行運算，但若運算元為 INT 與 WORD 型態的數值時，兩者之間便可進行混合運算，且 DINT 與 DWORD 之間的混合運算也同樣是合法的，不過若是不同資料長度的運算元則不允許進行混合運算。

```

0001 Var_0 := Var_INT + Var_WORD ; ✓
0002
0003 Var_1 := Var_DINT + Var_DWORD ; ✓
0004
0005 Var_2 := Var_INT + Var_DINT ; ✗

```

- 進行運算時，不論是 INT 系列或 WORD 系列的資料，系統都會將其視為帶符號的整數來處理。
- ST 語言可編輯的行數並沒有限制，但仍必需考慮主機的記憶體容量。
- ISPSOft 的 ST 編輯環境支援 **複製/剪下/貼上** 的功能，因此使用者可從其他的文字編輯器中將編輯好的內容複製到 ISPSOft 的 ST 編輯工作區中。
- 下表則為 ST 語言不支援的指令：

AH/AS 系列機種

所有 IL 的專用指令皆不可使用，例如 LD、LDI、OUT、SET、RST、LD&、LDP...等。

所有脈衝型的 API 指令皆不支援，例如 ROLP、RORP、BCDP、TRDP...等。

所有**四則邏輯運算**、**邏輯控制**、**接點型態邏輯運算與結構指令**類別的指令皆不支援。另外也不支援 PLS、PLF、MOV、DMOV、DFMOV、\$MOV、MC、MCR、CJ、JMP、GOEND...等指令。

一般計數器編號搭配的 CNT、DCNT 指令，以及副程式型計時器編號搭配 TMR 指令

- 下表則為 ST 語言有支援的指令，但須有所限制使用的規範事項：

AS 系列機種

所有高速輸出指令 (API 27xx)，如 DPLSY、DDRVI 等，使用前請參考 DDRVI 指令說明。

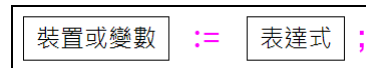
高速計數器搭配 DCNT 指令時，判斷式需永遠被執行，以利刷新現在計數值。

通訊類的指令，而且需等待設定完成旗標的指令，須注意程式流程可讓指令被掃描執行。

13.2 ISPSOft 中的 ST 敘述句結構

13.2.1 指定結構 :=

- 格式



- 功能說明

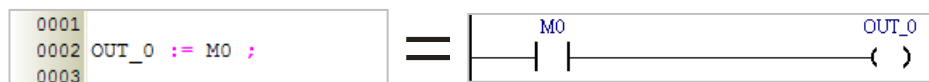
將右側表達式所代表的值指定（傳送）給左側的裝置或變數符號。

- 使用規則與限制

- 結構右側的表達式可為常數、裝置、變數符號或任何運算式，但左側只能是裝置或變數符號。
- 指定結構兩側的運算元必須符合下列規則才可傳遞資料。
 - (a) M、S、T、C、HC 裝置與 BOOL、STEP、TIMER、COUNTER 等位元型態的符號可互傳。
 - (b) D、L 裝置可與所有 BOOL、STEP、REAL、STRING 以外所有型態的符號互傳資料。
 - (c) T、C 裝置可與 D、L 裝置及 WORD、INT、TIMER、COUNTER 等型態的符號互傳資料。
 - (d) D、L、HC 裝置與指定 HC 裝置的 COUNTER 符號可與 DWORD 與 DINT 的符號互傳資料。
 - (e) WORD 系列與 INT 系列可互傳資料。
 - (f) REAL 系列與 STRING 只可與相同型態且長度相同的符號互相傳遞資料。
 - (g) WORD 與 DWORD 可以互傳資料，但 DWORD 傳到 WORD 資料可能會遺失。
 - (h) INT 與 DINT 可以互傳資料，但 DINT 傳到 INT 資料可能會遺失。

- 範例

EX1：將 M0 的值傳遞給變數 OUT_0。



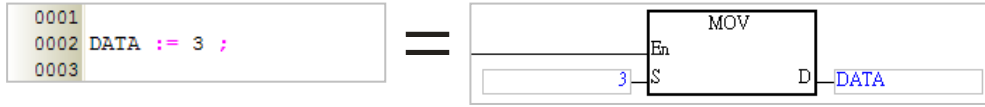
EX2：將裝置 M0 設置為 ON。



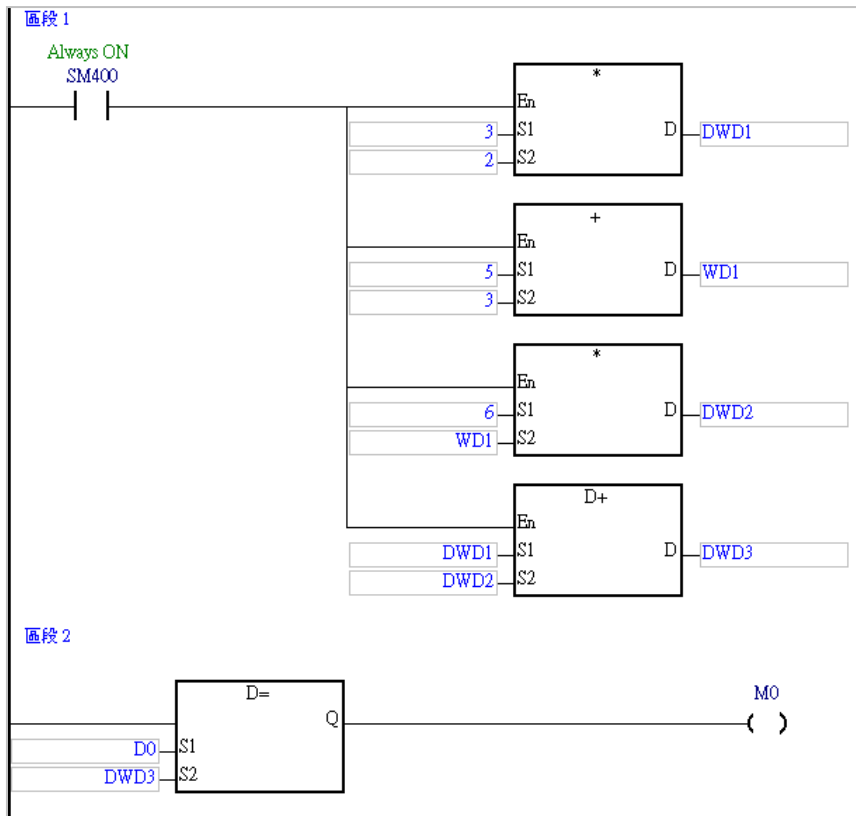
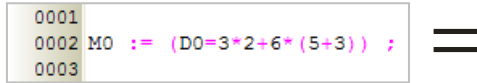
EX3：將 D1 與 D2 的內容值相加後傳遞給 D0。



EX4：將變數 DATA 的內容值指定為 3。



EX5：確認 D0 的內容值是否等於 $3*2+6*(5+3)$ ，並把判斷結果傳送給裝置 M0。

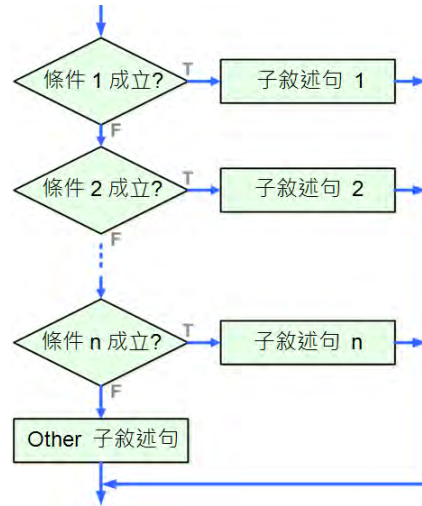


EX6：下表則為幾個錯誤的範例。

ST 敘述句	說明
D0 := M0 ;	兩側的資料型態不符。
D0 := V_REAL ;	REAL 符號不可直接與 D 裝置互傳資料。
V_LREAL := V_REAL ;	REAL 系列的符號必須長度相同才可互傳資料。
V_DWORD := V_TIMER ;	TIMER 符號不可傳值給 DWORD 符號，但可傳給 WORD。
V_WORD := HC0 ;	HC 裝置不可傳值給 WORD 符號，但可傳給 DWORD。

13.2.2 條件結構 IF

● 格式



13

● 功能說明

若「布林值表達式 1」的值為真 (True) 時，程式便會執行「子敘述句 1」後離開；若為否 (False) 則往下判斷，並依此類推...；當所有的判斷值都為否時，最後便會執行「Other 子敘述句」後離開。

● 使用規則與限制

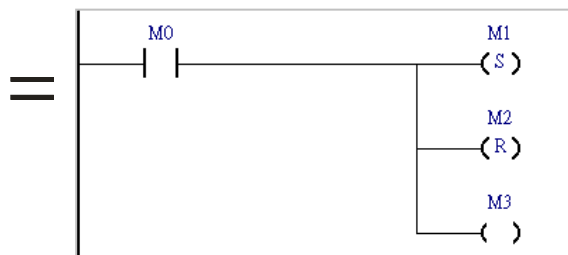
- 布林值表達式可為裝置、變數符號或是任何運算式，但不可為常數。
- 子敘述句的內容可以是任何合法的完整敘述句，當然也可以是另一個 IF 結構。
- 跟在關鍵字 THEN 或 ELSE 後面的子敘述句數量並沒有限制。
- 結構中的 ELSIF 與 ELSE 部份均可省略，而當 ELSE 被省略時，若當中所有的布林值表達式都為否 (FALSE) 時，該結構最後將不會執行任何子敘述句而直接離開。
- 結構中的 ELSIF 可隨意增加，但 ELSE 的部份只能有一個，且必須加在最後一段。

● 範例

EX1：如果 M0 為 ON，則把 M1 設為 ON，M2 設為 OFF，否則就直接結束此步驟；且不論 M0 的值為何，其值將會永遠傳遞給 M3。

```

0001 IF M0 THEN
0002     M1 := TRUE ;
0003     M2 := FALSE ;
0004 END_IF ;
0005
0006 M3 := M0 ;
  
```



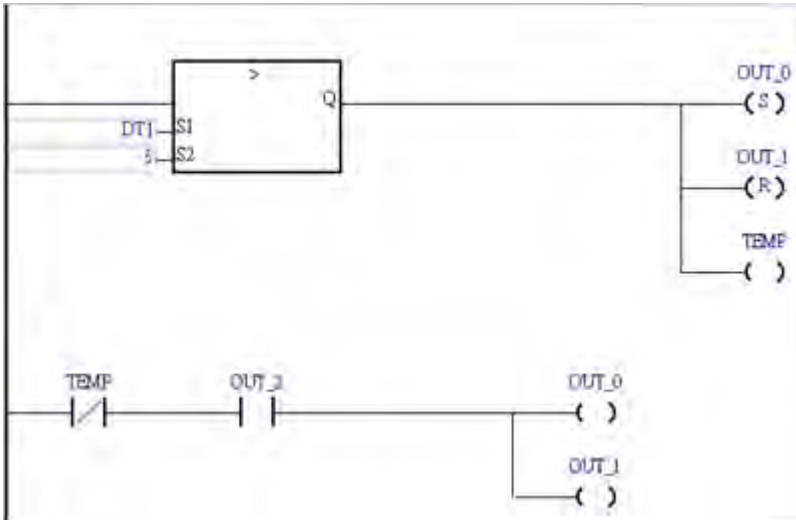
13

EX2：若 DT1 的值大於 5，則將 OUT_0 設置為 ON，OUT_1 重置為 OFF，否則 OUT_0 與 OUT_1 的狀態都將依循 OUT_2。

```

0001 IF DT1>5 THEN
0002     OUT_0 := TRUE ;
0003     OUT_1 := FALSE ;
0004 ELSE
0005     OUT_0 := OUT_2 ;
0006     OUT_1 := OUT_2 ;
0007 END_IF;
    
```

=

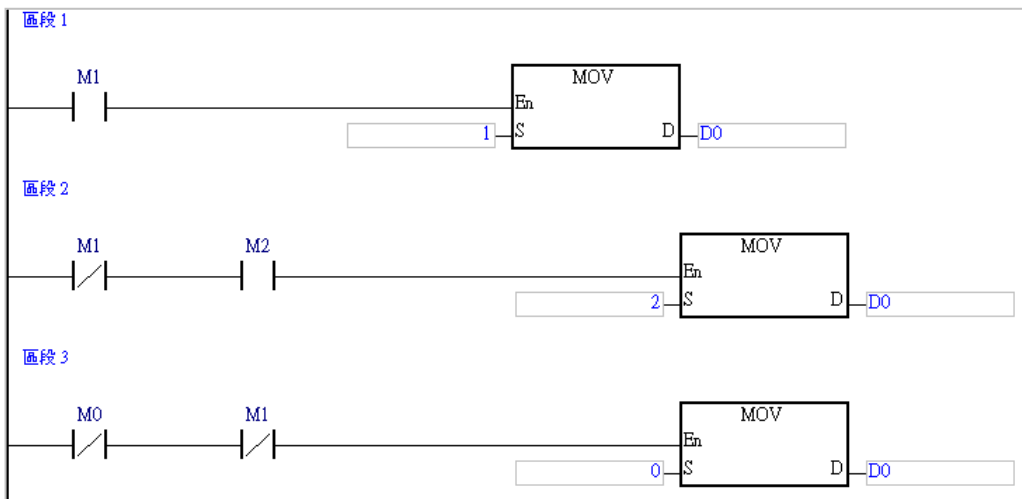


EX3：若 M1 為 ON，則將 D0 的值設為 1；但若 M1 不為 ON，則繼續判斷 M2 是否為 ON，若是，則將 D0 的值設為 2；但若 M2 也不為 ON，則 D0 的值將被設為 0。

```

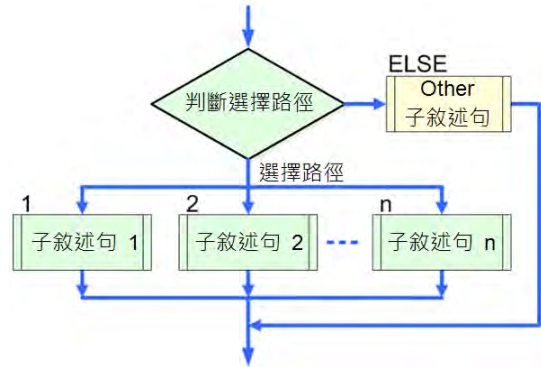
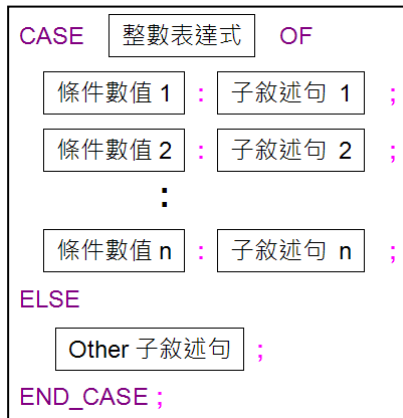
0001 IF M1 THEN
0002     D0 := 1 ;
0003 ELSIF M2 THEN
0004     D0 := 2 ;
0005 ELSE
0006     D0 := 0 ;
0007 END_IF;
    
```

=



13.2.3 條件結構 CASE

● 格式



● 功能說明

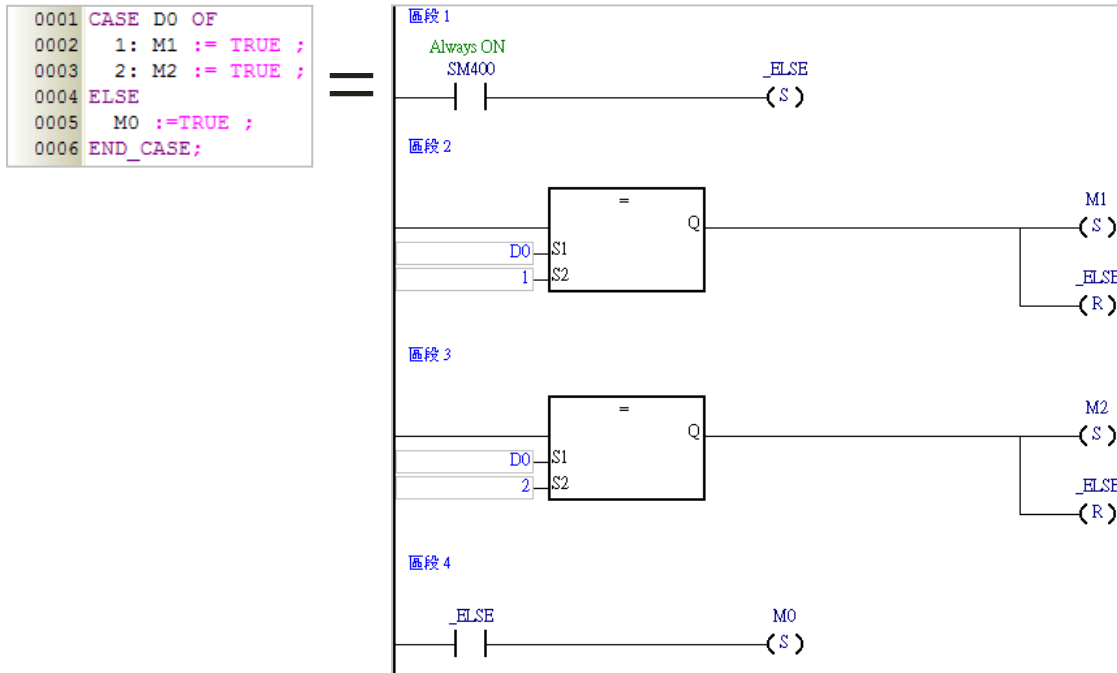
判斷「**整數表達式**」的值是否在「**條件數值**」的列表中，若是，則執行該條件所對應的「**子敘述句**」後離開此結構；若否，則執行關鍵字「**ELSE**」之後的「**Other 子敘述句**」後離開。

● 使用規則與限制

- 上圖結構中的**整數表達式**可為裝置、變數符號或是任何運算式，但不可為常數，且其代表數值的資料型態必須為 INT (-32768 ~ 32767) 或 DINT (-2147483648 ~ 2147483647) 型態的整數，而若直接使用裝置編號，如 D0，系統則會自動將該裝置的內容值視為 INT 型態的資料；除此之外，若使用運算式且其結果不為整數時，程式將會自動略去小數而只取整數的部份。
- 列表中的條件數值不可有重複且必須為整數，其範圍則須視 CASE 關鍵字後的整數表達式型態，即 INT 型態為 -32768 ~ 32767，而 DINT 則是 -2147483648 ~ 2147483647。
- 當有不同的條件數值卻執行相同的敘述句時，編輯時便可將其合併為一列。
 - (a) 若各條件為不連續數值，則各數值間須以逗號隔開，例如 **1, 3, 5 : 子敘述句**，而此例則代表當**整數表達式**的數值滿足 1、3、5 中的任何一個時，系統便會執行同一個子敘述句。
 - (b) 若各條件數值是某個連續的區塊，則編輯時可使用『 .. 』符號來表示條件數值的範圍，例如 **3..6 : 子敘述句** 便代表當**整數表達式**的數值落在 3 到 6 的範圍內 (包含 3 與 6) 時，系統便會執行同一個子敘述句。且當用『 .. 』符號表示條件數值時，左側數值必須小於右側，如 **6..3 :** 的範圍敘述便是不合法的。
- 子敘述句的內容可以是任何合法的完整敘述句，當然也可以是其他的結構。
- 在 **CASE** 結構中的每個條件數值都可對應多個子敘述句。
- 關鍵字 **ELSE** 與其對應的子敘述句可被省略，但實際操作時，若**整數表達式**的值沒有在條件數值的列表當中，則該結構將會不執行任何子敘述句而直接離開。
- 在執行完對應條件的子敘述句後，程式便會直接跳離該 **CASE** 結構，而無需其他的跳脫指令。

● 範例

EX1：若 D0=1，則 M1 為 ON；若 D0=2，則 M2 為 ON；若 D0 為其他數值則 M0 為 ON。

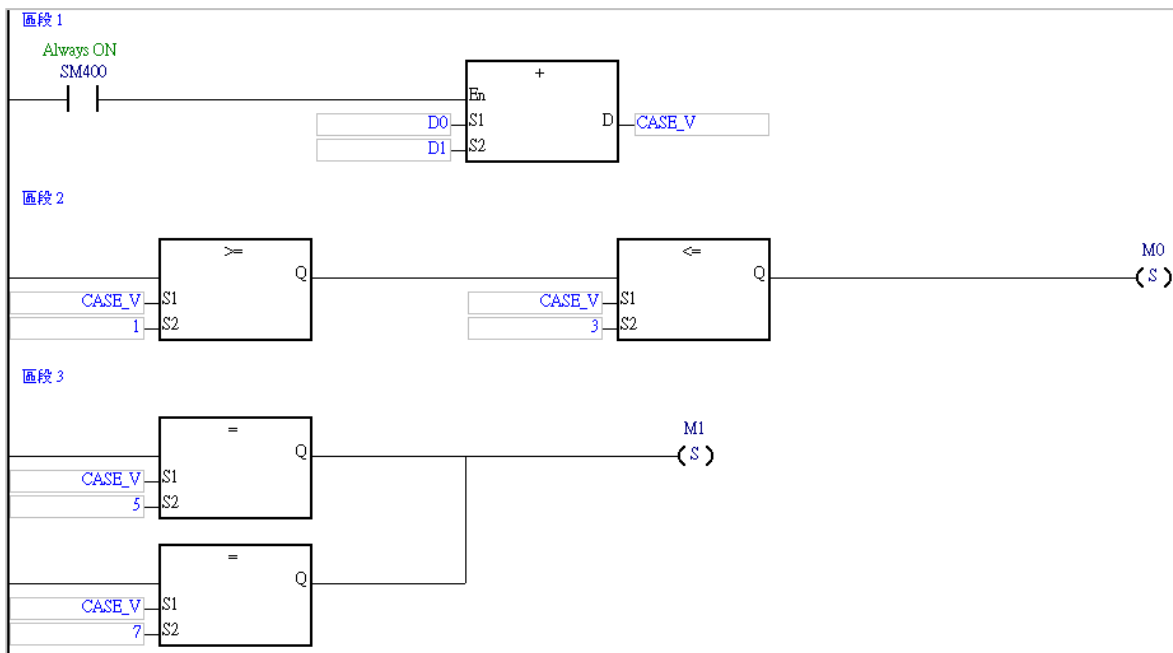


13

EX2：先判斷 D0+D1 的運算值，若該值的範圍為 1~3，則將 M0 設置為 ON；但若該值為 5 或 7，則將 M1 設置為 ON，而當該值不在上述的範圍時，程式將不做任何動作並直接離開此結構。

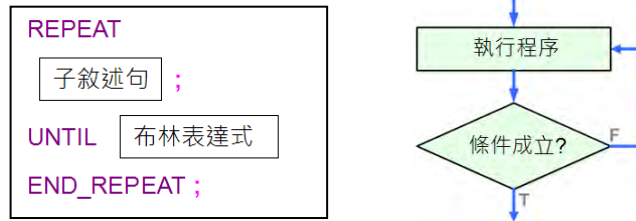
```

0001 CASE D0+D1 OF
0002 1..3: M0 := TRUE;
0003 5,7 : M1 := TRUE;
0004 END_CASE;
    
```



13.2.4 迴圈結構 REPEAT

- 格式



- 功能說明

先執行子敘述句後再判斷「布林表達式」的值是否為真 (True)，若其值為真，則程式會直接離開此結構；若為否 (False)，則結構內部的程序便會重複執行，直到該值為真 (True) 之後才會離開。

- 使用規則與限制

- 布林表達式可為裝置、變數或是任何運算式，但不可為常數。
- REPEAT 結構為先執行子敘述句後再判斷是否離開，因此結構中的子敘述句至少會執行一次。
- 子敘述句的內容可以是任何合法的完整敘述句，當然也可以是其他的結構，且 REPEAT 結構中的子敘述句數量並沒有限制，但若是迴圈內部還有迴圈的巢狀結構時，最多則只能有 64 層。
- UNTIL 後的布林表達式為離開迴圈的判斷條件，因此請勿使用無法更新的裝置或變數來當作該表達式的運算元，以避免造成無限迴圈。
- X 裝置的值必須在程式的掃描周期完成後才會被更新，因此若要以 X 裝置的狀態做為離開迴圈的條件時，請務必在結構內部加上 IO 更新指令的敘述句，或是在中斷程式中更新 X 裝置的值。

- 範例

EX1：DT1 的初始值為 0，進入迴圈後，DT1 便會持續加 2 遞增，直到 DT1 大於 100 時才會結束迴圈。

```

0001 DT1 := 0 ;
0002 REPEAT
0003     DT1 := DT1 + 2 ;
0004 UNTIL DT1 > 100
0005 END_REPEAT ;
  
```

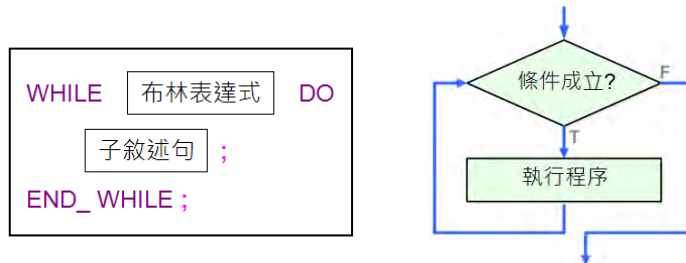
EX2：下列為一個階層運算的範例。進入迴圈之前，D0 的值會先初始化為 1，而迴圈結束後，D0 的值便會等於 D10 內容值的階層運算；若 D10 的值為 5，則最後的 D0 值便會等於 $5 \times 4 \times 3 \times 2 \times 1$ 。

```

0001 D10 := 5 ;
0002
0003 D0 := 1 ;
0004 REPEAT
0005     D0 := D0 * D10 ;
0006     D10 := D10 - 1 ;
0007 UNTIL D10 = 0
0008 END_REPEAT ;
  
```

13.2.5 迴圈結構 WHILE

● 格式



● 功能說明

先判斷「布林表達式」的值是否為真 (True)，若為真 (True)，則程式便會繼續執行子敘述句，且只要該值持續為真 (True) 時，結構內部的程序便會重複執行，直到該值為否 (False) 之後才會離開。

● 使用規則與限制

- 布林表達式可為裝置、變數或是任何運算式，但不可為常數。
- WHILE 結構為先判斷條件再決定是否執行迴圈中的敘述，因此內部的敘述句可能完全不執行。
- 子敘述句的內容可以是任何合法的完整敘述句，當然也可以是其他的結構，且 WHILE 結構中的子敘述句數量並沒有限制，但若是迴圈內部還有迴圈的巢狀結構時，最多則只能有 64 層。
- 跟在關鍵字 WHILE 之後的布林表達式為離開迴圈的判斷條件，因此請勿使用無法更新的裝置或變數來當作該表達式的運算元，以避免造成無限迴圈。
- X 裝置的值必須在程式的掃描周期完成後才會被更新，因此若要以 X 裝置的狀態做為離開迴圈的條件時，請務必在結構內部加上 IO 更新指令的敘述句，或是在中斷程式中更新 X 裝置的值。

● 範例

EX1：DT1 的初始值為 0，進入迴圈後，DT1 便會持續加 2 遞增，直到 DT1 大於 100 時才會結束迴圈。

```

0001 DT1 := 0 ;
0002
0003 WHILE DT1 <= 100 DO
0004   DT1 := DT1 + 2 ;
0005 END_WHILE;
  
```

EX2：右側為 WHILE 結構與 REPEAT 結構的比較，而在此例中，我們會先讓離開迴圈的條件在進入之前就成立。因 REPEAT 結構是先執行再判斷，因此其結構內部的子敘述句會被執行一次，所以離開迴圈後的 D0 值便會等於 1；而 WHILE 結構則是先判斷再執行，且因一開始所判斷的離開條件就已成立，所以其結構內部的子敘述句完全沒被執行過，因此結束後的 D1 值還是會保持為 0。

```

0001 MO := TRUE ;
0002 D0 := 0 ;
0003 D1 := 0 ;
0004
0005 REPEAT
0006   DO := DO + 1 ;
0007 UNTIL MO
0008 END_REPEAT ;
0009
0010 WHILE (NOT MO) DO
0011   D1 := D1 + 1 ;
0012 END_WHILE ;

```

13

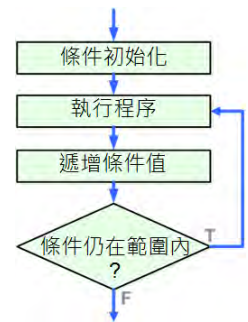
13.2.6 迴圈結構 FOR

● 格式

```

FOR 條件變數或裝置 := 起點值表達式 TO 終點值表達式 BY 遞增量表達式 DO
    子敘述句 ;
END_FOR ;

```



● 功能說明

先將「條件變數或裝置」的內容設為「起點值」，之後每執行完一次迴圈內的程序就將「條件變數或裝置」的內容值加上一次的「遞增量」，接著再判斷其值是否還在「起點值」至「終點值」的範圍內，若是，則繼續執行迴圈；若否，則程式便會直接離開 FOR 結構。

● 使用規則與限制

- 上述的起點值、終點值與遞增量表達式可以是裝置、變數符號、運算式或常數，且其代表數值的資料型態必須為 INT (-32768 ~ 32767) 或 DINT (-2147483648 ~ 2147483647) 型態的整數，而若直接使用裝置編號，如 D0，系統則會自動將該裝置的內容值視為 INT 型態的資料；除此之外，若使用運算式且其結果不為整數時，程式將會自動略去小數而只取整數的部份。
- 當起點值小於終點值時，所設定的遞增量便必須為負數，否則將會造成無限迴圈。
- 條件變數的資料型態亦必須是 INT 或 DINT，且當條件變數或裝置在迴圈的程序當中有參與運算時，請小心避免因變更到它的值而造成無限迴圈。
- 子敘述句的內容可以是任何合法的完整敘述句，當然也可以是其他的結構，且 FOR 結構中的子敘述句數量並沒有限制，但若是迴圈內部還有迴圈的巢狀結構時，最多則只能有 64 層。

- 實際運作時，系統會先將條件變數或裝置的內容值加上遞增量後，再判斷是否可以離開迴圈，但若加上遞增量之後的值會產生溢位時，程式便會因為離開的條件無法成立而造成無限迴圈。

● 範例

在迴圈的執行過程中，符號 INDEX 會由 1 開始每次遞增 2，一直到第 5 次的循環結束後，INDEX 的值會再被加上 2 而等於 11，而此時系統便會判斷已超過條件範圍 (1~9) 而離開迴圈；且因符號 SU 的初始值為 0，因此離開迴圈後，SU 的值便會是 1~9 的奇數平方和，即 $1^2 + 3^2 + 5^2 + 7^2 + 9^2$ 。

```
0001 SU := 0 ;
0002
0003 FOR INDEX := 1 TO 9 BY 2 DO
0004     SU := INDEX * INDEX + SU ;
0005 END_FOR ;
0006
```

13

13.2.7 API 應用指令結構

● 格式

```
API 指令名稱 ( 運算元 1 , 運算元 2 , ... , 運算元 n );
```

● 功能說明

執行 API 應用指令的功能。

● 使用規則與限制

- 於 ST 程式中執行 API 指令時，本身並不須要指定 En 端點的狀態，只要程式執行到該敘述句時，對應的 API 指令就會被執行。不過應用上還是可以搭配 IF 結構來達到具有 En 端點的效果。
- 只要不破壞運算元與關鍵字之完整性，任何的換行都是被允許的。
- 在 ST 語法中的 API 應用指令也是一個敘述句，因此最後的結尾一樣要加上分號『;』。
- 指令中的每個運算元都必須被加在括號中，而可用的運算元則請參考各機種的指令手冊。
- 內部運算元的排列順序不可變動，其順序請參考各機種的指令手冊。
- 在 ISPSOft 中的 API 應用指令並不支援回傳值的功能，因此 `M0 := CMP (D0, D10, M0);` 的敘述句便是不合法的。
- 只要指令的結構不變，其文字的換行是允許的。

✓

```
0001 CMP (
0002     D0,
0003     D10,
0004     M0
0005 );
```

✓

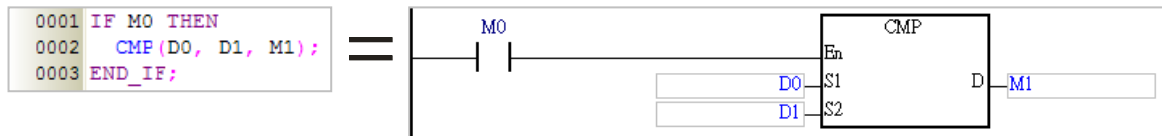
```
0001 CMP
0002 ( D0, D10, M0 );
0003
0004
0005
```

✗

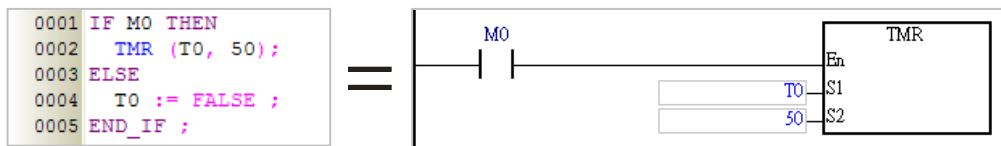
```
0001 CM
0002 P ( D0, D10, M0 );
0003
0004
0005
```

● 範例

EX1：搭配 IF 結構使用 CMP 指令。

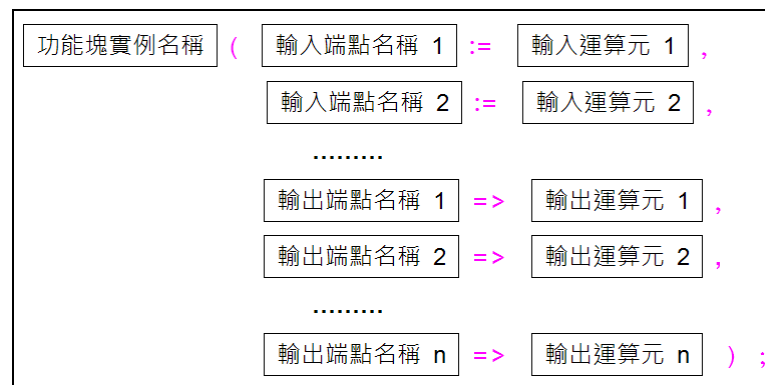


EX2：因在 ST 程式中的 API 應用指令無法操作 En 端點，因此若使用了計時器指令 (TMR) 時，請務必於程式中加入重置計時器的敘述句，如下例的行號 4，否則該計時器將無法歸零。



13.2.8 功能塊 (FB) 呼叫結構

● 格式



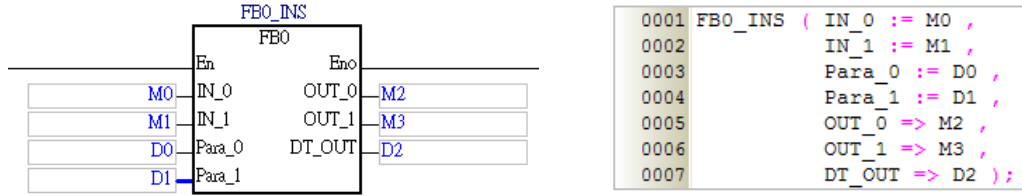
● 功能說明

執行功能塊的功能。

● 使用規則與限制

- 於 ST 程式中呼叫功能塊時，本身並不須要指定 En 端點的狀態，只要程式執行到該敘述句，被呼叫的功能塊就會執行。不過應用上還是可以搭配 IF 結構來達到具有 En 端點的效果。
- 只要不破壞運算元與關鍵字的完整性，任何的換行都是被允許的。
- 功能塊的結構也是一個敘述句，因此最後的結尾一樣要加上分號「;」。
- 功能塊的每個輸出入端點與其對應的運算元都必須被包含在括號之中，且每組端點與運算元之間須以逗號隔開，而可用的運算元型態則必須依據各功能塊的定義來加以指定。
- 括號之前的名稱應為**功能塊實例**而非**功能塊定義**，亦即應輸入被宣告成功能塊型態的變數符號名稱。關於功能塊定義與與功能塊實例的詳細說明請參考第 7 章的相關內容。
- 輸入側端點所使用的指定符號為「:=」，而輸出側端點所使用的指定符號則是「=>」。

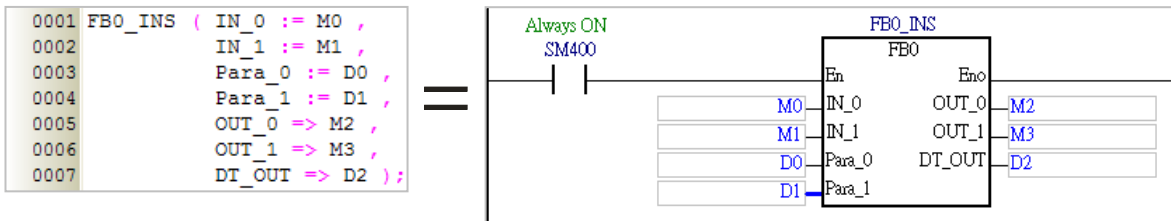
- 結構內的端點與運算元順序必須依照功能塊的定義，而不可任意變動，其順序則為排除 En/Eno 端點後，先由上而下列出所有輸入側端點，其中 VAR_IN_OUT 類別的端點則被歸類為輸入側；接著再由上而下列出所有輸出側的端點。請參考下列功能塊的階梯圖圖示與 ST 結構。



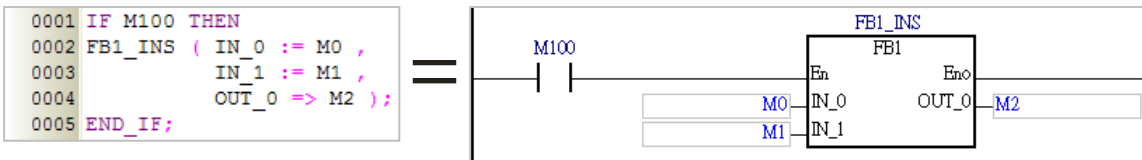
● 範例

EX1：在 ST 的程式中，只要執行到呼叫功能塊的敘述句時，被呼叫的功能塊就會執行。

13



EX2：搭配 IF 結構來執行功能塊的功能。

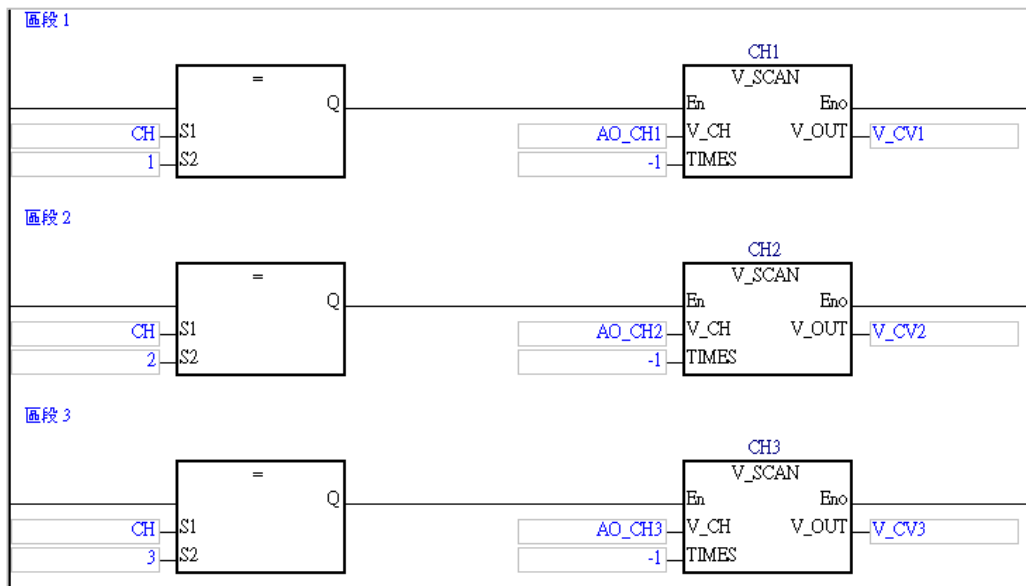


EX3：根據變數符號 CH 的值來決定執行哪一個功能塊實例。

```

0001 CASE CH OF
0002   1: CH1 ( V_CH := AO_CH1,
0003           TIMES := -1,
0004           V_OUT => V_CV1 );
0005
0006   2: CH2 ( V_CH := AO_CH2,
0007           TIMES := -1,
0008           V_OUT => V_CV2 );
0009
0010   3: CH3 ( V_CH := AO_CH3,
0011           TIMES := -1,
0012           V_OUT => V_CV3 );
0013 END_CASE;
    
```

=



13.2.9 空白敘述

- 格式

```

;
    
```

- 功能說明

用於必須加入敘述句，但卻又無須執行任何動作的應用場合。

- 範例

下圖為一個不執行任何動作的 FOR 迴圈，此迴圈的功能在於將程式停滯在迴圈的處理中以達到延遲 (DELAY) 執行的效果，而調整迴圈次數便可調整延遲的時間。

```

0001 FOR D100:=1 TO 3000 BY 1 DO
0002   ;
0003 END_FOR;
    
```

13.2.10 RETURN 敘述

- 格式

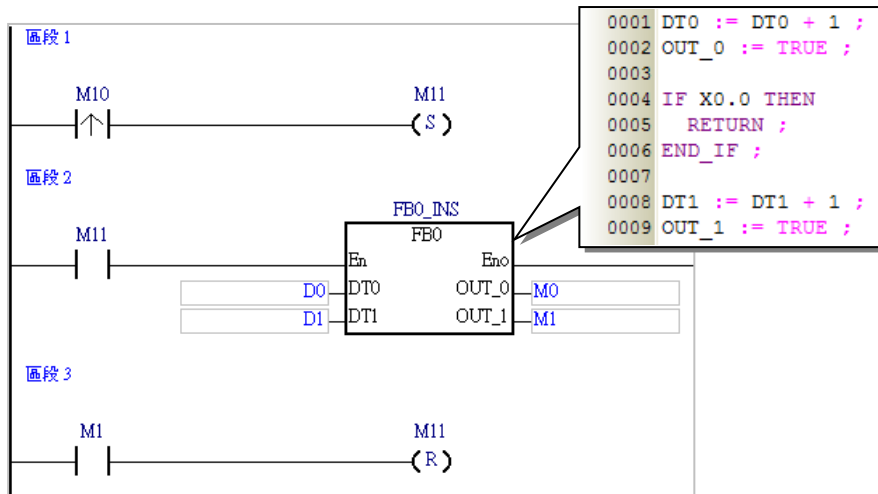
```
RETURN ;
```

- 功能說明

此敘述句，只可加在用 ST 語言撰寫的功能塊程式當中，其功能為強制離開目前正在執行的功能塊。

- 範例

下列中的功能塊 FB0 為使用 ST 語言所撰寫的程式。當程式實際運作時，在功能塊執行完上半段的程序後便會先判斷 X0.0 的狀態，若 X0.0 為 ON，則此功能塊的動作便會立即結束；但若 X0.0 為 OFF，則該功能塊，便會執行完整段的程序後才會離開。



13.2.11 EXIT 敘述

- 格式

```
EXIT ;
```

- 功能說明

此敘述句，只可用於迴圈結構的內部，包括 **REPEAT**、**WHILE** 及 **FOR**，其功能則為強制脫離迴圈。

● 範例

EX1：下例為一個 REPEAT 的迴圈結構，而離開的條件除了待 DT1 持續遞減至 0 之外，迴圈中又加入了一個 IF 結構搭配 EXIT 的敘述句，且讓 D0 由 0 開始每次遞增 1，而當 D0 遞增至 100 時，程序便會直接離開迴圈，所以該迴圈最多只能執行 100 次；在下列的範例中，因 DT1 的設定值為 110，因此當 DT1 的值遞減至 10 的時候，迴圈便會因為已執行了 100 次而直接結束，所以最後 DT1 的值便會等於 10，而不會遞減至 0。

```
0001 DT1 := 110 ;
0002 D0 := 0 ;
0003
0004 REPEAT
0005
0006     DT1 := DT1 - 1 ;
0007     D0 := D0 + 1 ;
0008
0009     IF D0=100 THEN
0010         EXIT ;
0011     END_IF ;
0012
0013 UNTIL DT1 = 0
0014 END_REPEAT ;
```

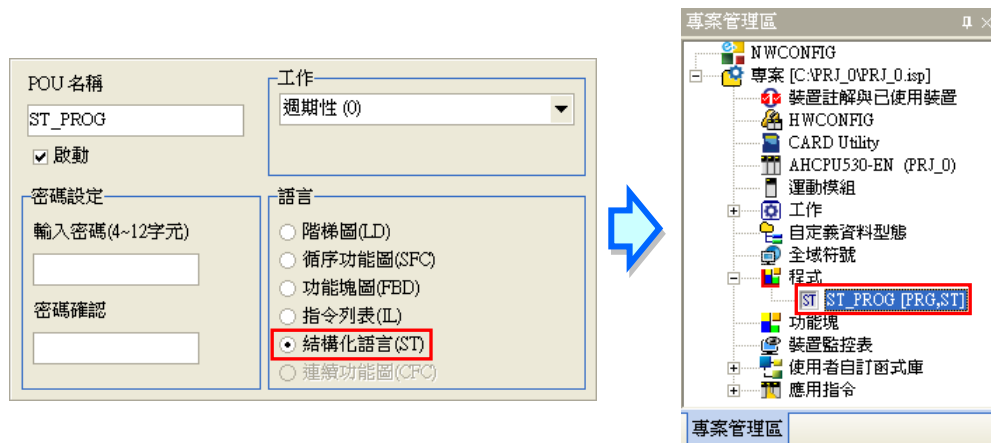
EX2：下例為一個 FOR 迴圈，原本是進行 DT1 內容值（初始值為 10）的階層運算，並將最後的運算結果放置在裝置 D0 當中，但其中加入了一個 IF 結構搭配 EXIT 敘述句後，只要 D0 大於 100 時，程式便會立即停止計算並強制離開迴圈；因此在下列的範例中，當 D10 遞增至 5 時，D0 的值便會是 $1*2*3*4*5=120$ ，而迴圈則也會因此提前結束。

```
0001 DT1 := 10 ;
0002 D0 := 1 ;
0003
0004 FOR D10 := 1 TO DT BY 1 DO
0005
0006     D0 := D0 * D10 ;
0007
0008     IF D0>100 THEN
0009         EXIT ;
0010     END_IF ;
0011
0012 END_FOR ;
```

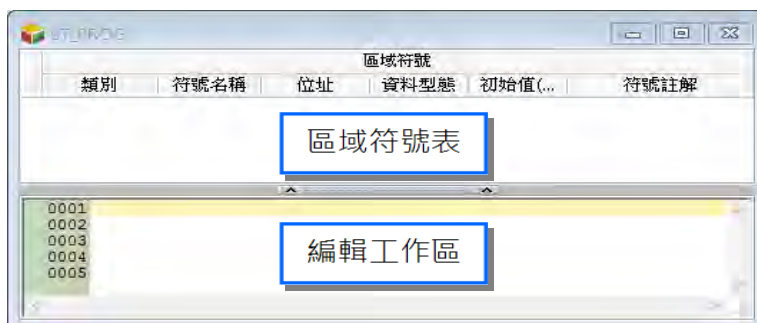
13.3 於 ISPSoft 中建立 ST 程式

13.3.1 ISPSoft 的 ST 編輯環境

建立 POU 時，於語言欄位選擇「結構化語言 (ST)」便可建立 ST 程式，請參考第 5.4.1 節的說明。



下圖即為 ISPSoft 中的 ST 編輯環境，其視窗配置的上方為區域符號表，而下方則是編輯工作區，且在編輯工作區的左側預設會有行號顯示，使用者可自行設定其顯示的格式，詳細操作請參考第 2.3.1 節的相關說明。在進入 ST 的編輯環境後，視窗中亦會出現對應的圖示工具列，其功能如下。



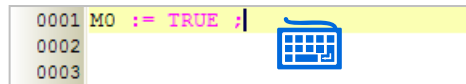
:= IF REPEAT WHILE FOR CASE

圖示	功能說明
:=	插入指定結構 := 的範本
IF	插入條件結構 IF 的範本
REPEAT	插入迴圈結構 REPEAT 的範本
WHILE	插入迴圈結構 WHILE 的範本
FOR	插入迴圈結構 FOR 的範本
CASE	插入條件結構 CASE 的範本

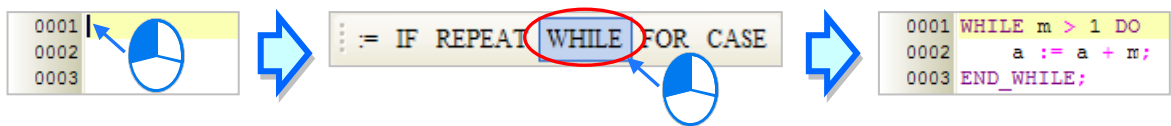
13.3.2 編輯 ST 敘述

在 ST 的編輯工作區中，其操作方式與一般的文字編輯器類似，編輯時只要依據 ST 語法的格式直接使用鍵盤在工作區上面輸入或修改文字即可，且當按下鍵盤的【Enter】鍵時便可執行換行的動作。

```
0001 MO := TRUE ;
0002
0003
```



除此之外，亦可先點選欲編輯的位置後，直接透過圖示工具列來插入結構的範本並進行修改即可。

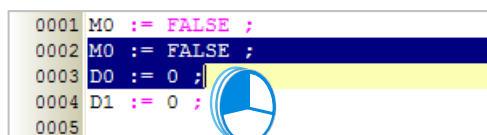


或者直接輸入結構的第一個英文字母，接著點選自動帶出的項目來插入結構，鍵盤的游標也會直接移動到輸入條件位置，使用者可更快速的編輯 ST。

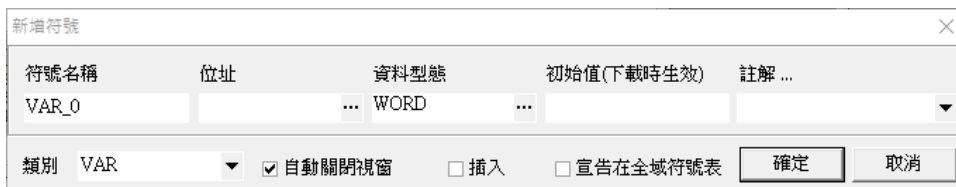
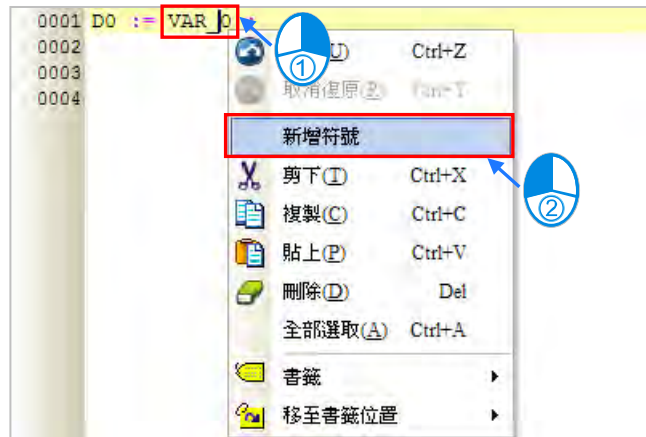


選取文字的方法也與一般的文字編輯器相同。先將滑鼠點在選取區塊的起點並按住滑鼠左鍵，之後往選取的方向移動便可框選文字；又或者是先用滑鼠點選在某個區塊的起點，接著按住鍵盤的【Shift】鍵，之後再用滑鼠點選該區塊的終點便可將整段區塊選取起來。

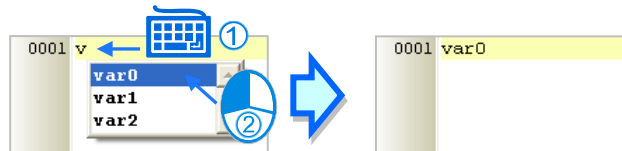
```
0001 MO := FALSE ;
0002 MO := FALSE ;
0003 DO := 0 ;
0004 D1 := 0 ;
0005
```



當在編輯的過程中輸入了未經宣告的變數符號時，輸入完畢後用滑鼠點擊該文字並按下右鍵，之後再於快捷選單中點選「新增符號」便可自動導出變數符號的宣告視窗。



程式輸入的過程中，會自動從已建立的變數符號中篩選出符合的變數符號供使用者挑選，可利用鍵盤的「↑」、「↓」鍵，或直接在選單中點擊滑鼠左鍵完成輸入。

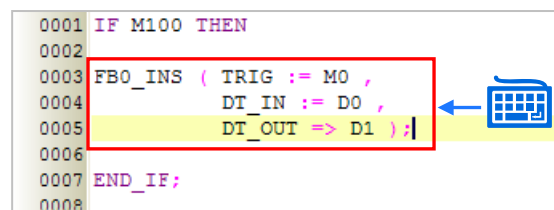


13.3.3 插入 API 應用指令與功能塊

插入 API 應用指令與功能塊的方法有下述兩種。

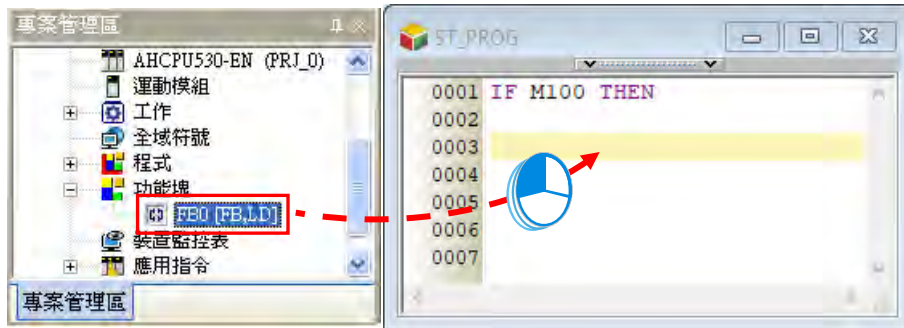
● 方法一

依據應用指令與功能塊的格式，直接於編輯工作區中利用鍵盤輸入即可。

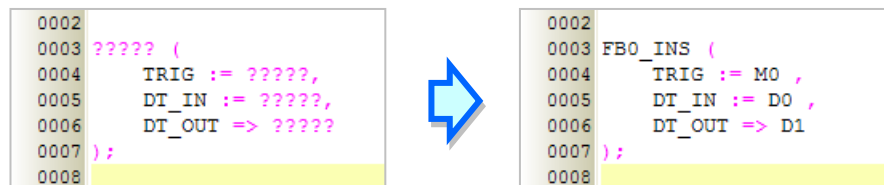


- 方法二

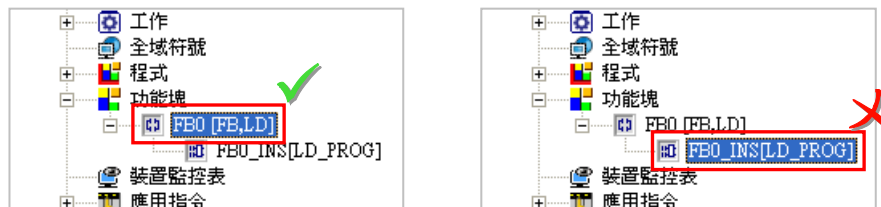
於專案管理區中展開應用指令或功能塊的目錄，在其中找到欲插入的項目後，於該項目上按住滑鼠左鍵，之後再將其拖曳至欲插入的位置後放開左鍵即可。



插入應用指令或功能塊之後，請再依據其定義的格式，直接於插入的範本中編輯對應的運算元。



拖曳功能塊時須注意只能拖曳功能塊的定義，而不能拖曳其下層的功能塊實例項目。



13.3.4 書籤功能

書籤功能為方便搜尋或移動至特定編輯位置的便利工具。

- 加入/移除書籤

先用滑鼠點選欲加入書籤的行列後按下滑鼠右鍵，接著於快捷選單中點選「書籤」項目，之後再於該項目中選擇欲加入的書籤編號即可。而要移除書籤時，同樣的請先用滑鼠點選欲取消書籤的行列後按下滑鼠右鍵，接著於快捷選單中點選「書籤」項目，之後再於該項目中點選與所選行列相同編號的書籤即可；不過若選擇的是其它編號時，所選行列的書籤便會被變更為該編號。若於功能工具列中點擊編輯 (E) > 書籤 > 移除所有書籤 (R)，則目前編輯視窗的中的所有書籤便會被全部取消。

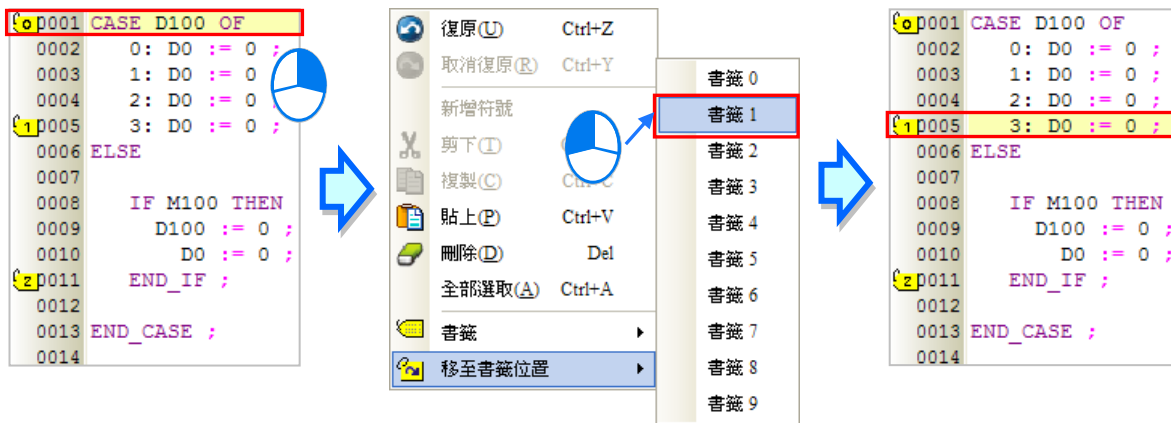
操作時若點選了一個已使用於其他行列的編號，則該編號的書籤便會被搬移至目前選取的行列上。



● 移至書籤位置

於編輯工作區中按下滑鼠右鍵，接著於快捷選單中點選「移至書籤位置」項目，之後在於目錄中選擇移動目標的書籤編號即可。

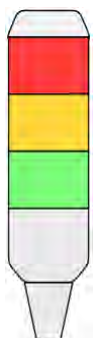
13



13.4 ST 的程式範例

13.4.1 範例說明

此範例為自動控制應用中經常會使用到的塔燈控制，而此處選用的則是三色燈號加上蜂鳴器的配置。下表即為各種狀態下的控制機制，其中當有一個以上的狀態成立時，系統將以優先權較高的為主。



狀態碼	設備狀態	燈號顯示				優先
		紅燈	黃燈	綠燈	蜂鳴器	
4	設備異常	ON	OFF	OFF	Flash	最高
3	設備異常 (關閉蜂鳴器)	Flash	OFF	OFF	OFF	↑
2	運轉中	OFF	OFF	ON	OFF	
1	工作完成	OFF	Flash	OFF	OFF	
0	閒置	OFF	ON	OFF	OFF	最低

13.4.2 硬體規劃

在本範例中，我們將選用 AH 系列主機 **AHCPU530-EN**，數位 IO 模組則選用 8 點輸入/8 點輸出的 **AH16AP11R-5A**，而主背板則選用 4 槽的 **AHBP04M1-5A**。下表即為本範例的 IO 配置表。

Type	ID	說明
DO	Y0.0	紅燈輸出
DO	Y0.1	黃燈輸出
DO	Y0.2	綠燈輸出
DO	Y0.3	蜂鳴器輸出

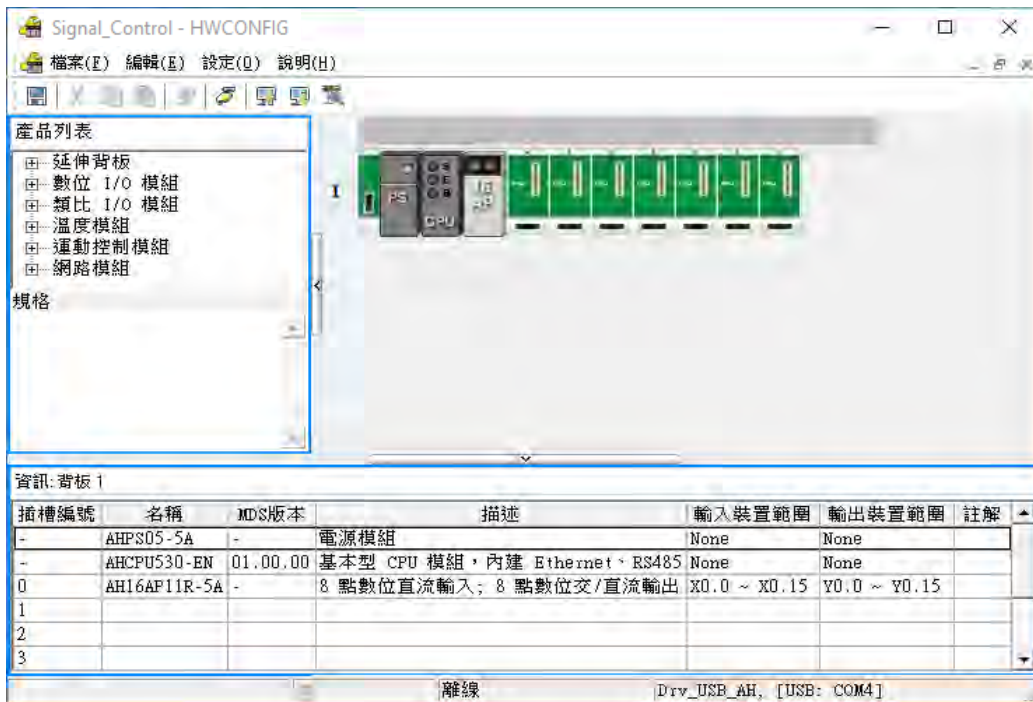
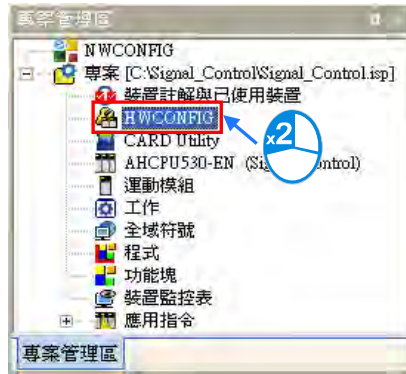
13.4.3 程式規劃

- (1) 當旗標 ERROR 為 ON，且旗標 BUZZER_OFF 為 OFF 時，將狀態碼 SC 設為 4。
- (2) 當旗標 ERROR 為 ON，而旗標 BUZZER_OFF 為 ON 時，將狀態碼 SC 設為 3。
- (3) 當旗標 ERROR 為 OFF，且旗標 RUNNING 為 ON 時，將狀態碼 SC 設為 2。
- (4) 當旗標 ERROR 與 RUNNING 都為 OFF，但旗標 COMPLETE 為 ON 時，將狀態碼 SC 設為 1。
- (5) 當旗標 ERROR、RUNNING 與 COMPLETE 都為 OFF 時，將狀態碼 SC 設為 0。
- (6) 依據狀態碼 SC 切換對應的燈號顯示狀態。而因燈號控制在許多的應用中都會使用，且燈號的顯示可以有許多種變化，因此考量程式的擴充性與可再用性，我們會將此段程式規劃為功能塊。

13.4.4 建立範例程式

首先請開啟 ISPSOft 並建立一新專案，本範例的硬體規劃與第 4 章相同，因此請參考第 4 章的內容，並自行完成硬體規劃的工作。

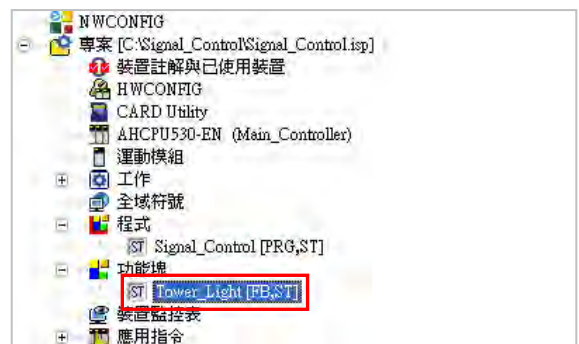
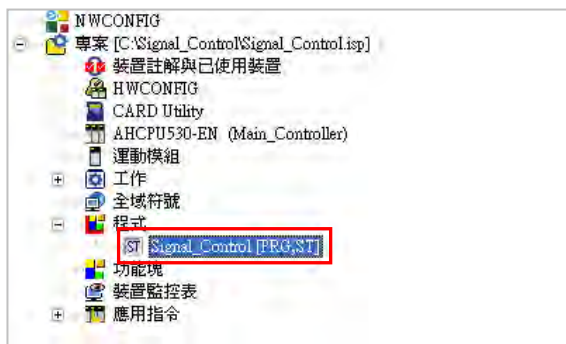
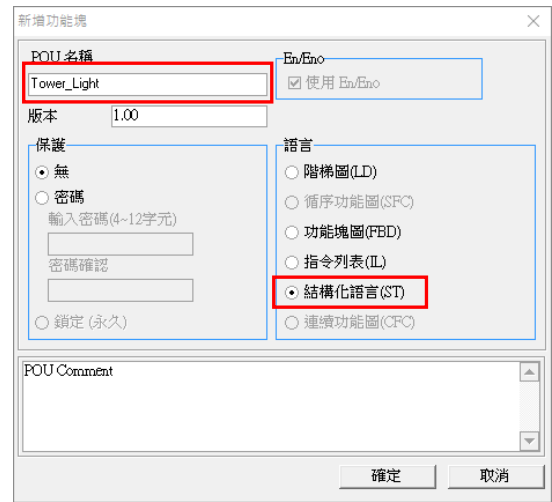
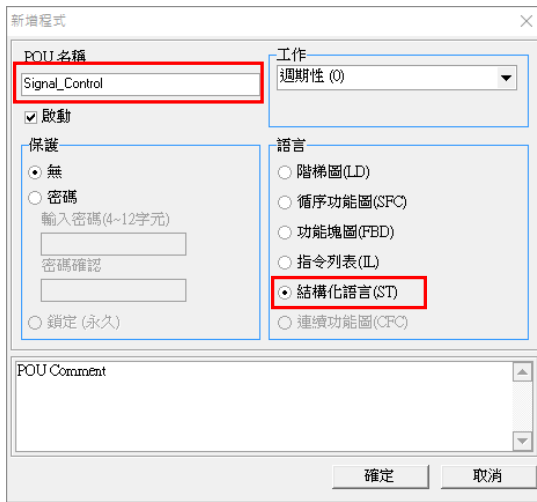
13



接下來請直接於專案管理區中分別建立主程式與功能塊的 POU，且在建立時請都選擇使用 ST 語言。



13



再來請將已知的系統旗標及 IO 配置建立在全域符號表中。

類別	符號名稱	位址	資料型態	初始值(下載時生效)	符號註解
VAR	ERROR	N/A [Auto]	BOOL	FALSE	
VAR	RUNNING	N/A [Auto]	BOOL	FALSE	
VAR	COMPLETE	N/A [Auto]	BOOL	FALSE	
VAR	BUZZER_OFF	N/A [Auto]	BOOL	FALSE	
VAR	RED	Y0.0	BOOL	FALSE	
VAR	YELLOW	Y0.1	BOOL	FALSE	
VAR	GREEN	Y0.2	BOOL	FALSE	
VAR	BUZZER	Y0.3	BOOL	FALSE	

接著再編輯功能塊 POU 的程式，而在規劃上，我們須要建立一個輸入端點用以接收主程式所指定的狀態碼，且針對燈號與蜂鳴器的部份，我們亦將其狀態的輸出結果建立為功能塊的輸出端點。

因功能塊的功用在於接收主程式的狀態碼，並在經過判斷之後將對應的燈號狀態輸出，所以相當適合使用 CASE 的結構來設計；而當後續有不同的燈號顯示模式時，我們只須要新增一個狀態碼，接著再於此功能塊的程式中插入一個 CASE 的條件即可，因此對於後續的擴充將會有相當大的彈性。

編輯時可於編輯工作區中插入 CASE 結構的範本後再加以修改，或是直接輸入下列的 ST 程式。

類別	符號名稱	位址	資料型態	初始值(下...	符號註解
VAR_INPUT	MODE	N/A [Auto]	INT	N/A	
VAR_OUTPUT	RL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	YL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	GL	N/A [Auto]	BOOL	FALSE	
VAR_OUTPUT	BR	N/A [Auto]	BOOL	FALSE	

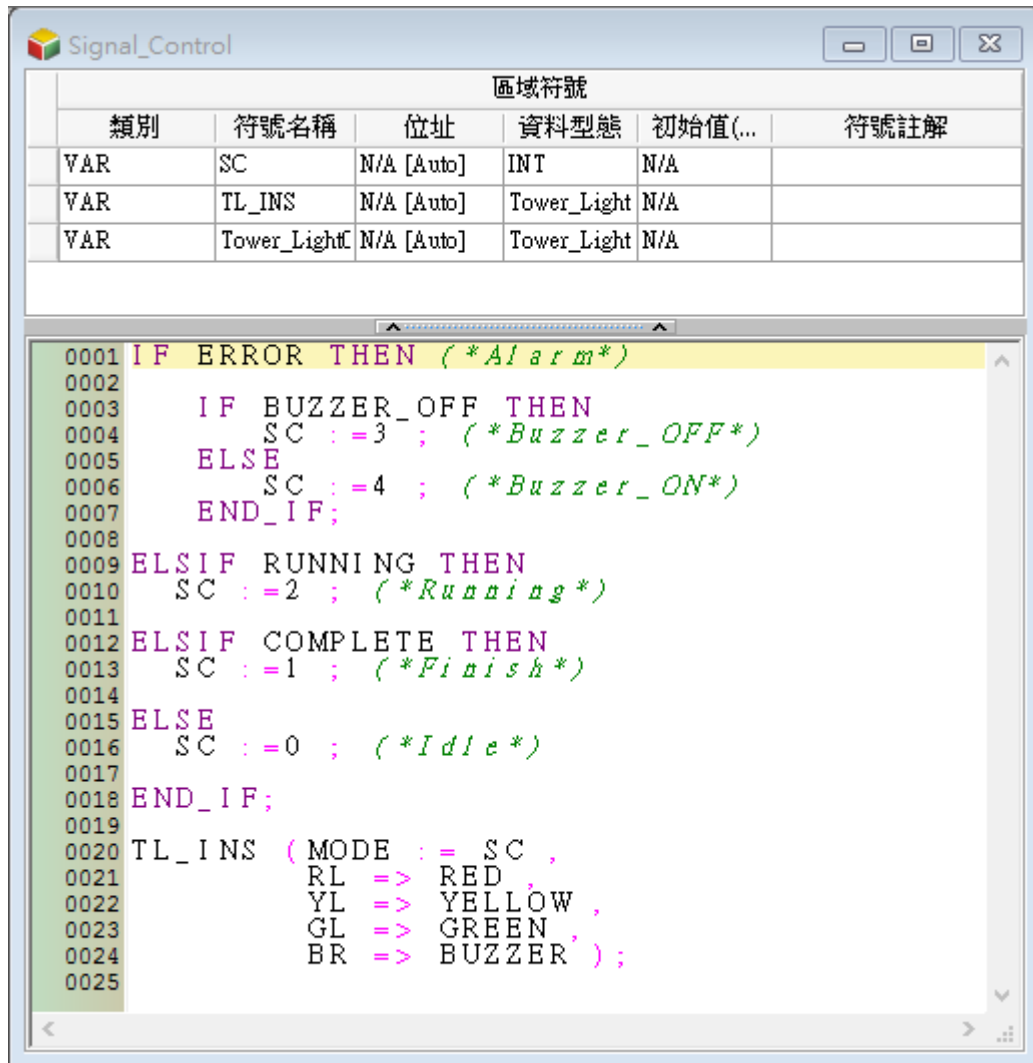
```

0001 CASE MODE OF
0002
0003 (* SM407: 0.5S on / 0.5S off *)
0004
0005     0:  RL: =FALSE ;
0006         YL: =TRUE  ;
0007         GL: =FALSE ;
0008         BR: =FALSE ;
0009
0010     1:  RL: =FALSE ;
0011         YL: =SM407 ;
0012         GL: =FALSE ;
0013         BR: =FALSE ;
0014
0015     2:  RL: =FALSE ;
0016         YL: =FALSE ;
0017         GL: =TRUE  ;
0018         BR: =FALSE ;
0019
0020     3:  RL: =SM407 ;
0021         YL: =FALSE ;
0022         GL: =FALSE ;
0023         BR: =FALSE ;
0024
0025     4:  RL: =TRUE  ;
0026         YL: =FALSE ;
0027         GL: =FALSE ;
0028         BR: =SM407 ;
0029
0030 END_CASE;

```

接下來便可繼續編輯主程式的部份，其中我們必須先在主程式的區域符號表中宣告代表狀態碼的變數符號 **SC** 以及功能塊實例 **TL_INS**；而在程式的規劃上，因為在燈號的控制中有優先權的機制，因此此程式較適合使用 **IF** 的結構來進行設計；在確定狀態碼之後，我們還須呼叫方才所建立的功能塊，並將狀態碼提供給功能塊進行運算，而最後所得到的結果便可再指定給實際輸出的 **IO** 接點。

最後完成的主程式如下。



The screenshot shows the 'Signal_Control' software interface. At the top, there is a table titled '區域符號' (Area Symbols) with the following content:

類別	符號名稱	位址	資料型態	初始值(...)	符號註解
VAR	SC	N/A [Auto]	INT	N/A	
VAR	TL_INS	N/A [Auto]	Tower_Light	N/A	
VAR	Tower_Light	N/A [Auto]	Tower_Light	N/A	

Below the table is a code editor window containing the following ST program code:

```

0001 IF ERROR THEN (*Alarm*)
0002
0003     IF BUZZER_OFF THEN
0004         SC := 3 ; (*Buzzer_OFF*)
0005     ELSE
0006         SC := 4 ; (*Buzzer_ON*)
0007     END_IF;
0008
0009 ELSIF RUNNING THEN
0010     SC := 2 ; (*Running*)
0011
0012 ELSIF COMPLETE THEN
0013     SC := 1 ; (*Finish*)
0014
0015 ELSE
0016     SC := 0 ; (*Idle*)
0017
0018 END_IF;
0019
0020 TL_INS ( MODE := SC ,
0021         RL => RED ,
0022         YL => YELLOW ,
0023         GL => GREEN ,
0024         BR => BUZZER );
0025
  
```

完成以上的編輯後，接著請自行完成編譯，並將硬體參數與程式下載至主機當中進行後續的測試。

MEMO

13

第14章 循序功能圖 SFC

目錄

14.1 認識循序功能圖 SFC	14-3
14.1.1 SFC 程式架構	14-3
14.1.2 SFC 動作原理	14-4
14.2 ISPSOft 中的 SFC 架構.....	14-5
14.2.1 步進點與 Action.....	14-5
14.2.2 轉換點 (Transition)	14-9
14.2.3 共進分歧與選擇分歧	14-11
14.2.4 共進合流與選擇合流	14-12
14.2.5 跳躍點	14-13
14.2.6 Actions 的修飾條件.....	14-15
14.2.6.1 條件定義.....	14-15
14.2.6.2 設定修飾條件的注意事項	14-19
14.2.7 初始步進點	14-20
14.2.8 內部屬性	14-21
14.2.8.1 Step 屬性參數.....	14-21
14.2.8.2 Action 與 Transition 屬性參數	14-25
14.3 於 ISPSOft 中建立 SFC 程式	14-33
14.3.1 ISPSOft 的 SFC 編輯環境	14-33
14.3.2 Actions 與 Transitions 的建立與管理	14-35
14.3.3 新增步進點結構.....	14-37
14.3.4 插入並聯轉換點.....	14-38
14.3.5 插入並聯步進點.....	14-39
14.3.6 插入步進點結構 - 共進分歧&選擇合流.....	14-40
14.3.7 插入步進點結構 - 選擇分歧&共進合流.....	14-41
14.3.8 插入跳躍點	14-42
14.3.9 指定步進點與轉換點符號.....	14-43
14.3.10 指定初始步進點.....	14-43
14.3.11 配置 Actions/摺疊 Action 列表	14-44

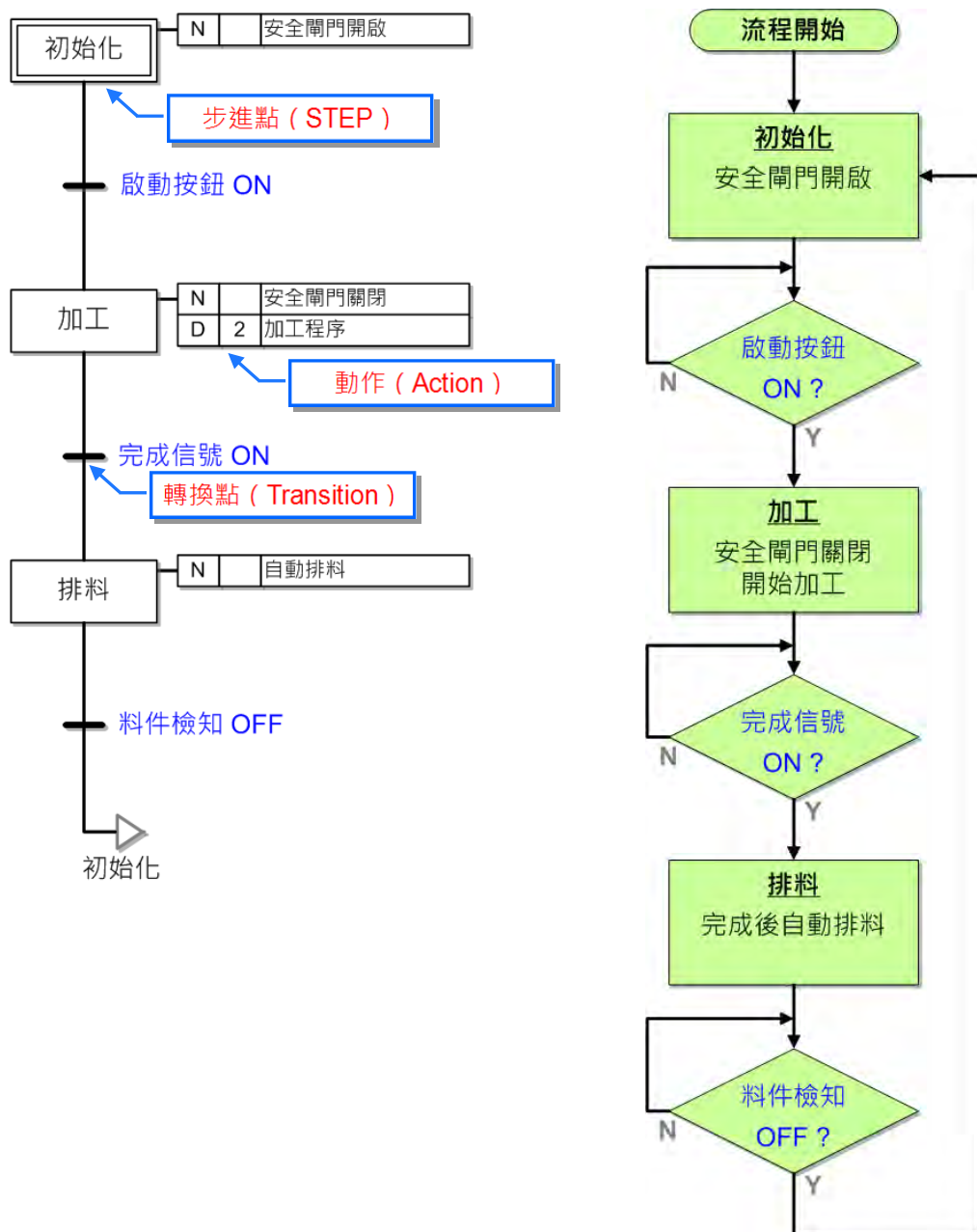
14.4 SFC 的程式範例	14-47
14.4.1 範例說明	14-47
14.4.2 硬體規劃	14-48
14.4.3 程式規劃	14-48
14.4.4 建立範例程式	14-49

14.1 認識循序功能圖 SFC

14.1.1 SFC 程式架構

循序功能圖 (Sequential Function Chart, SFC) 是 IEC 61131-3 所規範的 PLC 編程語言之一，其最大的特色在於以類似流程圖的觀念來規劃程式，適用於著重流程控制及狀態轉移的控制程序。

在下圖中，左側為基本的 SFC 圖形，而右側則為其對應的程式流程圖。由左側的結構可看出 SFC 程式是由許多的步進點 (STEP)、轉換點 (Transition) 以及每一個步進點當中的細部動作 (Action) 所構築而成，其中步進點的功能如同流程圖的執行程序，而轉換點則如同流程圖中的條件判斷。

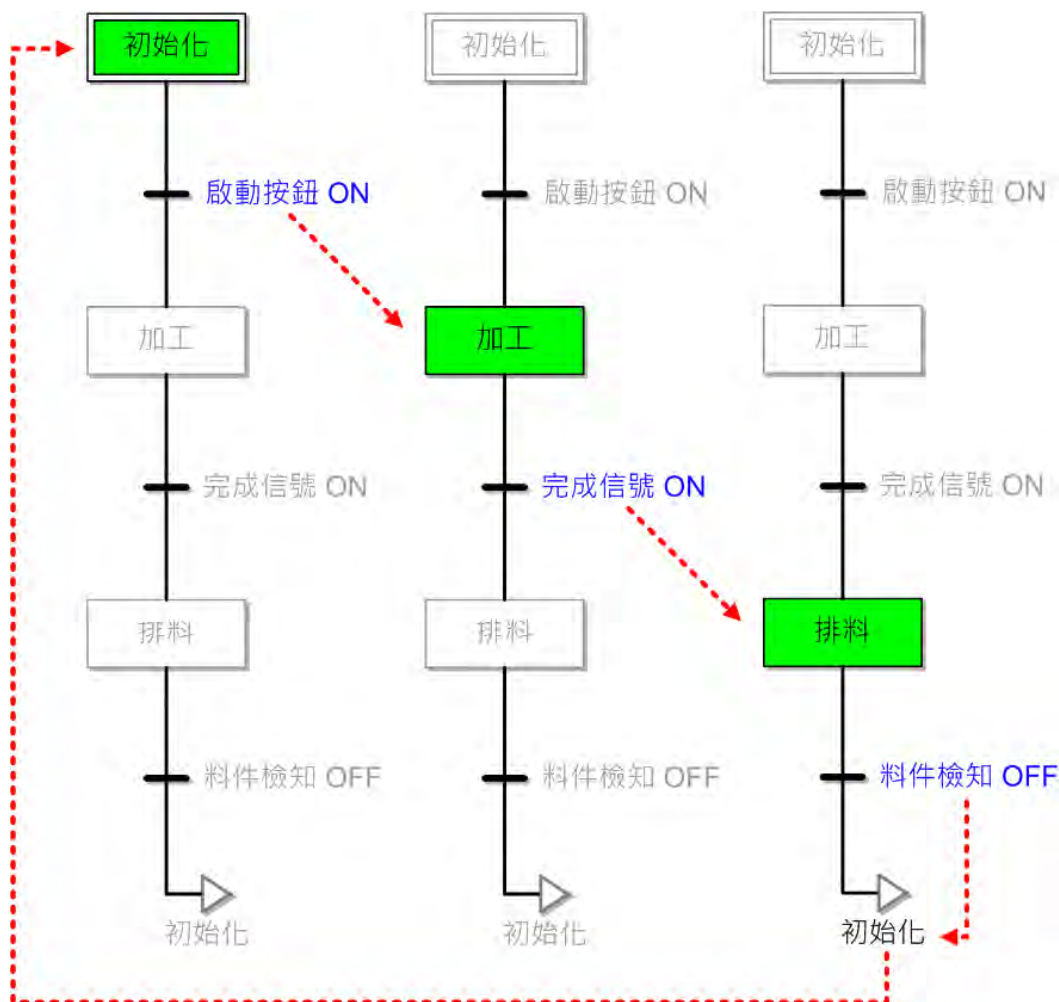


14.1.2 SFC 動作原理

SFC 程式的執行方式就如同接力賽一般，一個接著一個地傳遞執行權，當有某個步進點成為啟動狀態時，代表該步進點取得執行權，而系統便會執行該步進點所定義的程序，直到該步進點下方的轉換點條件成立後，程式才會將執行權移交給下一個步進點，並將目前執行的步進點關閉，像這樣執行權轉移的程序也叫做**步進點轉移**；而當依照這樣的規則重複執行，直到最下方的轉換點條件成立時，系統便會再將執行權轉移至最後指定跳躍的步進點來繼續整個程序的循環。

在執行過程中，系統只會關注擁有執行權的步進點，並執行其內部定義的動作程序，且也只會確認該步進點下方的轉換條件是否成立；換句話說，對於未獲得執行權的步進點而言，其下方的轉換條件並不會影響程式的執行步序，而系統也不會主動執行這些步進點所指定的動作程序。

在 ISPSOft 的 SFC 架構中，步進點與轉換點必須是交錯排列的，因此在編繪 SFC 的圖形時，連續兩個步進點或連續兩個轉換點的結構將無法通過編譯。



上例是單純應用的 SFC 架構，而在 ISPSOft 中尚提供許多豐富且靈活的編程功能，包括分歧與合流的結構、模組化的動作 (Action) 與轉換點 (Transition) 程序、可定義動作 (Action) 執行方式的修飾條件...等；而這些功能亦將在後續的章節中逐一介紹，並說明如何在 ISPSOft 當中進行實作。

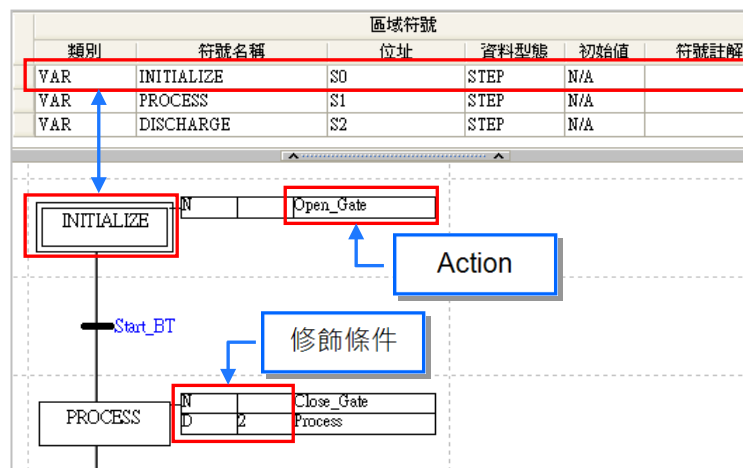
14.2 ISPSOft 中的 SFC 架構

若使用者使用 SFC 編程，SFC 編程架構內容可能會隨機種而不同，接下來本章節將對 SFC 架構做介紹。

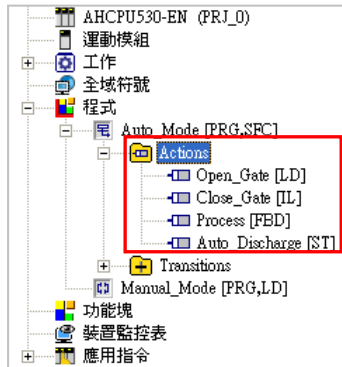
14.2.1 步進點與 Action

在 ISPSOft 中，每個步進點都必須配置一個 **STEP** 型態的變數符號來做為其啟動與否的狀態旗標，而當對應的旗標為 ON 時，便代表該步進點已進入啟動狀態，亦即取得執行權。AH5x0 機種及 DVP 機種每個 **STEP** 符號都將佔用一個 PLC 內部的 S 裝置，至於 S 裝置的使用限制與相關說明，詳細內容請參考各機種的操作手冊。

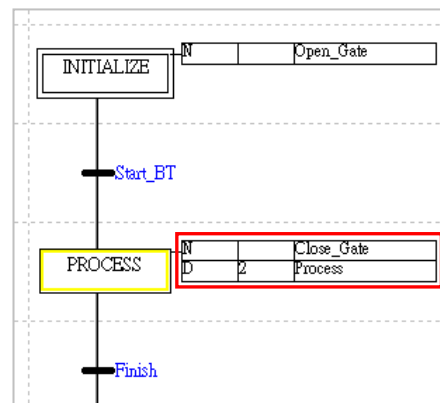
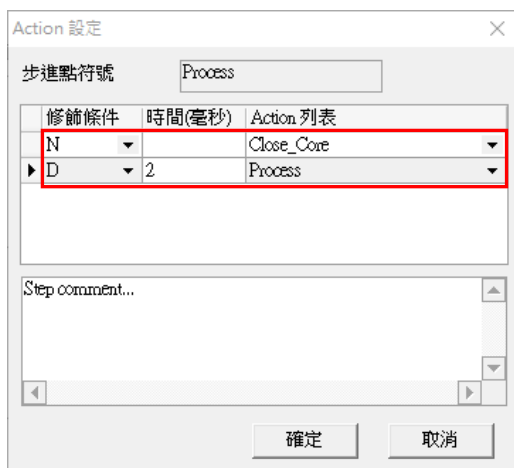
當步進點被啟動後，其內部所指定的動作即會依據所定義的方式開始執行，直到該步進點將執行權移交出去之後，所有對應的動作便會視其執行方式的定義來決定停止與否；而在 ISPSOft 中，這些動作程序亦可如同副程式一般的被模組化，並可重複在不同的步進點當中被指定，且同一個步進點亦可指定一個以上的動作模組，而這些動作模組在 ISPSOft 當中便稱之為 **Action**；此外，每個 Action 皆可單獨被定義其執行方式，至於決定 Action 執行方式的定義，在 ISPSOft 當中便稱之為**修飾條件**。



在 ISPSOft 中，Action 本身也是一段程式碼，且該程式碼亦可用階梯圖 (LD)、指令列表 (IL)、功能塊圖 (FBD) 或是結構化語言 (ST) 來進行開發，而所建立的 Action 程式亦會被列於專案管理區中；此外，Action 的編程視窗與各個語言的編程環境相同，唯一差異只有 Action 的編程視窗並沒有區域符號表，這是因為在 ISPSOft 中，同一個 SFC 程式的 Action 與 Transition 會共用同一份區域符號表。

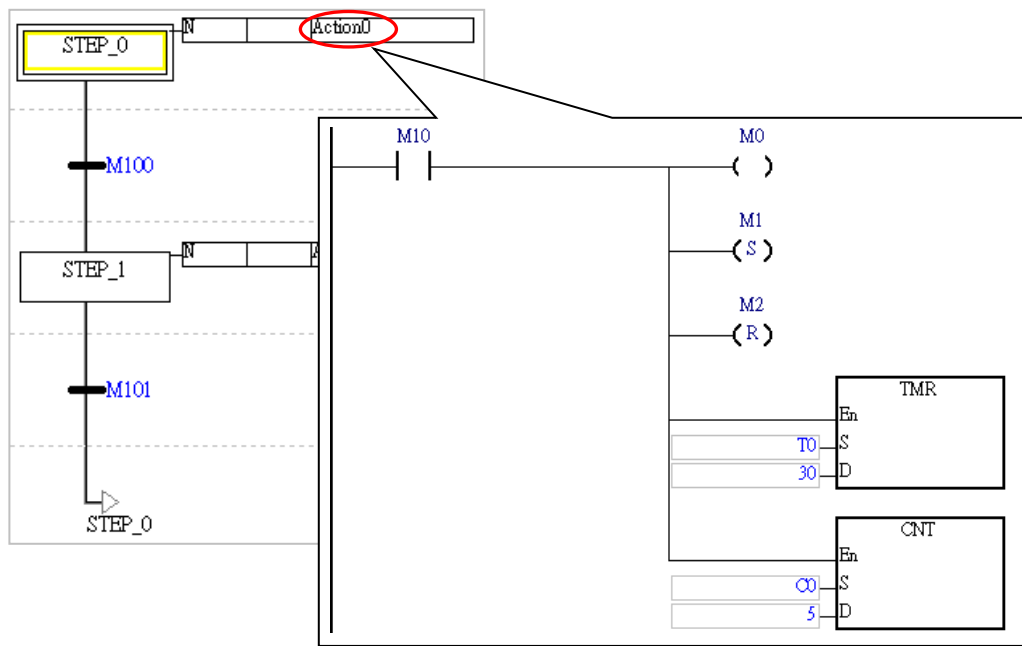


在編輯 SFC 的程式時，一般會先建立好 Action 的程式之後，再將其配置給各個步進點，並逐一定義每個 Action 的執行方式 (修飾條件)；下圖即為 Action 的設定畫面與設定完成後的 Action 列表。



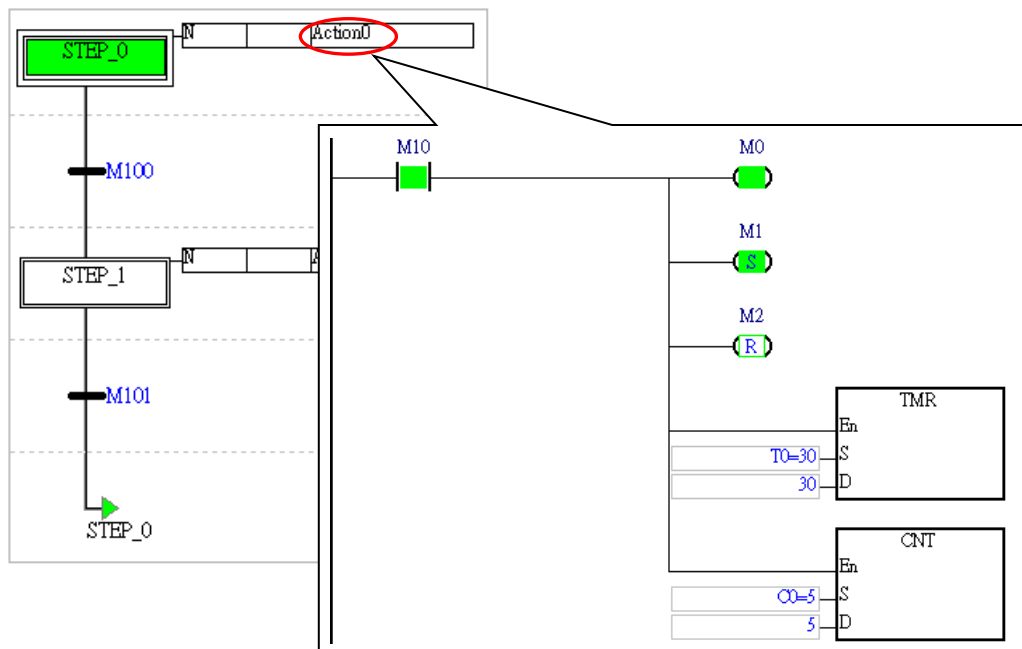
在 SFC 的執行過程中有一個相當重要的程序，稱之為「Final Scan」，該程序一般是發生在 Action 停止執行時，系統會再自動執行一個程序以關閉該 Action 當中的所有輸出動作；當執行「Final Scan」時，原本 Action 當中透過 OUT 指令驅動的線圈全部會關閉而呈現 OFF 狀態，而所有於該 Action 當中執行的應用指令與功能塊也不再執行，包括計時器也將被重置；不過若是透過 SET 或 RESET 指令而驅動的線圈，則因該指令的特性，所以仍會保持原狀，且計數器也只會停止計數，而其接點的狀態與計數值則同樣會維持不變，而不會如同計時器一般被重置。

以下圖為例，其中 Action0 為配置於 STEP_0 的 Action，而其修飾條件則為 N (Normal)，即一般的執行條件，因此當執行權由 STEP_0 轉移至 STEP_1 時，系統便會再執行一個 Final Scan 的程序。

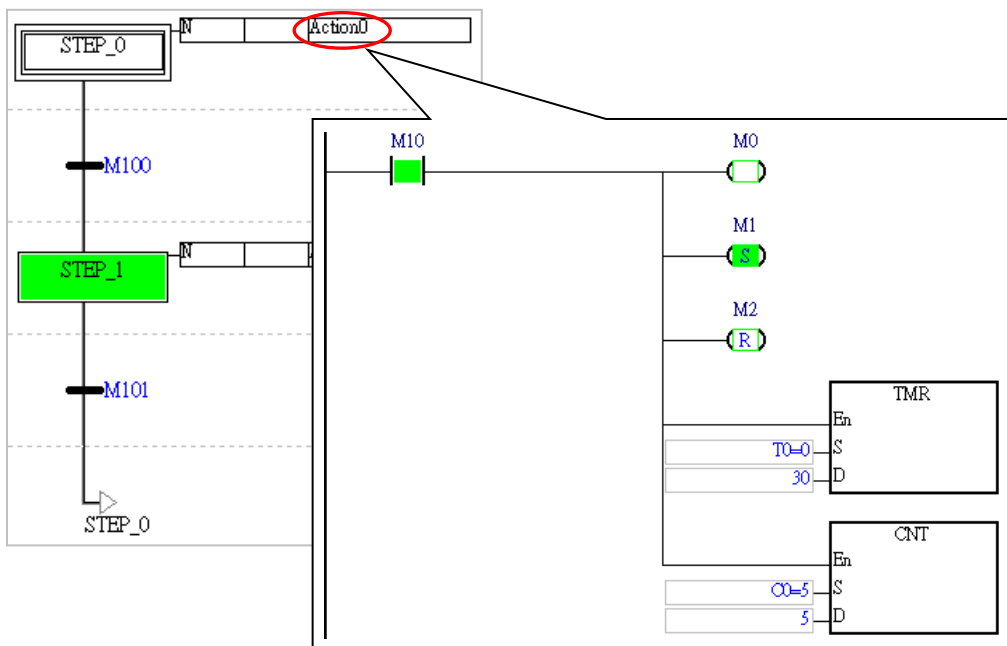


14

下圖為步進點 STEP_0 為啟動狀態的監控畫面，其中 Action0 的 M10 處於 ON 的狀態，因此輸出 M0 的狀態也會為 ON，而 M1 則被設置 (SET) 為 ON，M2 也被重置 (RST) 為 OFF，至於計時器 T0 與計數器 C0 則都處於計時/計數到達的狀態。

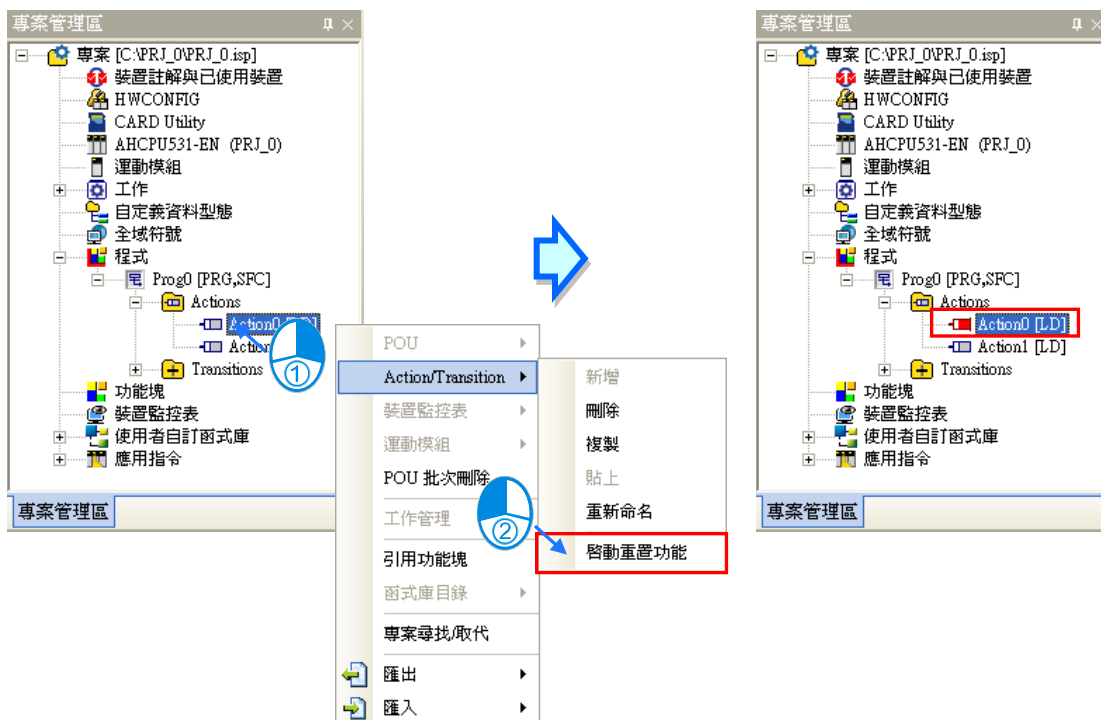


下圖則為步進點轉移至 STEP_1 後的監控畫面，其中可發現，即使 M10 仍保持為 ON，輸出的 M0 同樣會被關閉，且計時器 T0 也會被重置；不過 M1、M2 與計數器 C0 則還是會保持在原本的狀態。



14

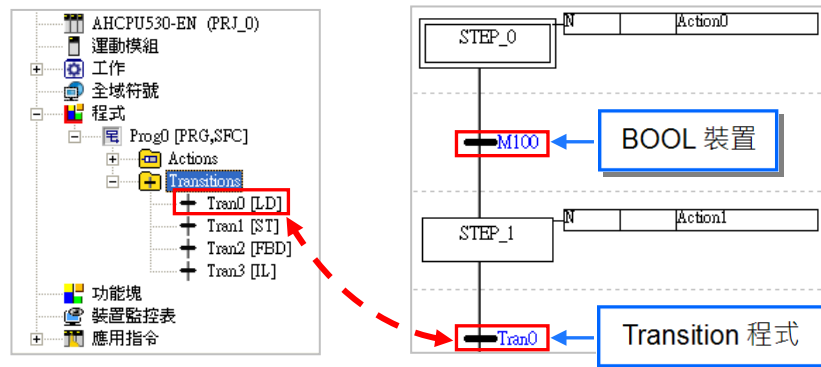
AH5x1、AH560 備援系列機種、AHxxEMC 與 AS 系列機種可設定每一個 Action 是否執行「Final Scan」，ISPSoft 為預設執行上述的「Final Scan」，欲取消時，在不執行「Final Scan」的 Action 上按下滑鼠右鍵，將 **Action/Transition > 啟動重置功能** 取消選擇，則此 Action 上的圖示將以紅色表示不執行「Final Scan」，因此，前述範例中的 STEP_0 在步進點轉移後，Action0 中的狀態仍將為轉移前的狀態。



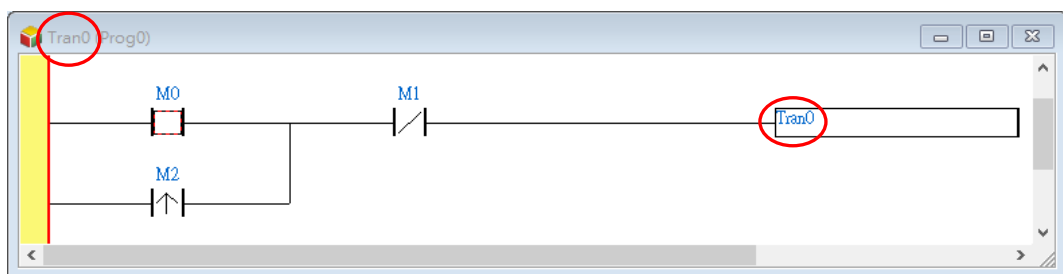
14.2.2 轉換點 (Transition)

轉換點是決定 SFC 執行時序的重點指標，它會決定目前啟動中的步進點是否該將執行權移轉到下一個步進點，且如同前述，只有位於目前啟動中的步進點，其下方的轉換點才能決定執行權的移轉。

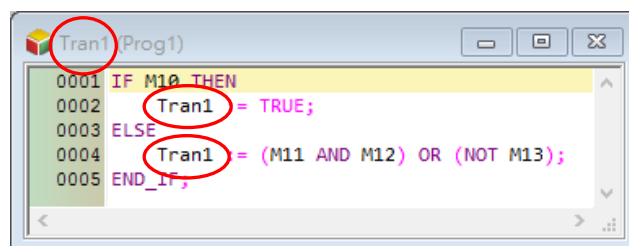
在 ISPSOft 中，轉換點可以是某個布林格式的裝置或變數，同時也可以是具備邏輯運算的程式碼；若轉換點是一段邏輯運算的輸出，則在編輯時便可先將這段運算程序建立為 Transition 程式，之後再將其指定於對應的轉換點即可，且所建立好的 Transition 程式也同樣會被列於專案管理區中。



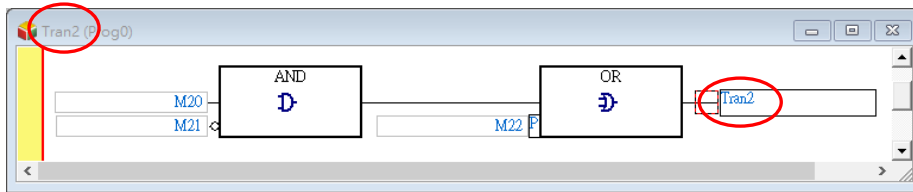
支援 Transition 程式的語言包括 LD、IL、FBD 與 ST，且雖然編輯的規則不同，其共通點則都是必須將某個布林狀態輸出至與該 Transition 程式同名的符號，而該符號亦無須另外宣告；不過若在符號表中有宣告與某個 Transition 程式同名的變數符號時，其編譯結果便會產生錯誤。除此之外，在 Transition 的程式中無法使用 API 應用指令與功能塊，但可以使用比較接點與區塊邏輯指令 (NP、PN、INV)。當使用 LD 來開發 Transition 程式時，其程式碼只能有一個階梯圖區段，而最後輸出的接點也只能是與該 Transition 程式同名的符號；除此之外，在 LD 的 Transition 程式當中不得有多重輸出。



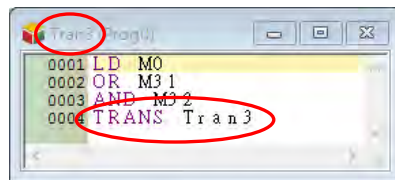
當使用 ST 來開發 Transition 程式時，其程式碼並沒有行數的限制，只要在程式碼中有將某個布林狀態指定給與該 Transition 程式同名的符號即可。



當使用 FBD 來開發 Transition 程式時，其程式碼也只可有一個功能塊圖區段，而最後的輸出節點也必須是與該 Transition 程式同名的符號，且不得有多重輸出。



當使用 IL 來開發 Transition 程式時，其程式碼並沒有行數的限制，但無法使用 OUT、SET、RST 與 API 應用指令，亦即程式中只可建立邏輯運算的部份，且最後輸出的指令亦必須固定為「TRANS」，並搭配與該 Transition 程式同名的符號做為其運算元。



當建立一個 Transition 程式時，系統便會預先將與該 Transition 程式同名的符號加入程式碼中，使用者僅需完成條件運算的部份即可；而因為同一個 SFC 程式的 Action 與 Transition 會共用同一份區域符號表，因此在 Transition 程式的編輯視窗中同樣沒有區域符號表。

由上述的規則可知，於 Transition 程式當中並無法建立過於複雜的程序，因此當轉換條件必須經過複雜的運算時，建議可將該程序建立於步進點的 Action 程式當中，並於該 Action 程式中產生一個決定步進點轉移的條件旗標，而之後再將該旗標指定於步進點下方的轉換點位置即可。

雖然使用者無須於符號表中宣告 Transition 程式的符號，但系統仍會為其配置一個記憶體的位置做為轉換條件是否成立的判斷旗標，因此編輯 Transition 的程式時，必須確保該符號可以被指定一個確切的邏輯狀態，否則便可能會產生執行上的錯誤。

以下圖 ST 的 Transition 程式為例，在第一次執行的過程中，一旦 (M0 AND M1) 的結果為 ON 時，Tran4 也將被設為 ON，而此時條件便會成立並將執行權轉移至下一個步進點；但當 SFC 的程序再度將執行權轉移回此步進點時，下圖的 Transition 程式便會再次運作，而此時若 (M0 AND M1) 的運算結果尚為 OFF，則下圖的程式便不會執行「THEN」後面的敘述句，因此 Tran4 將無法被更新而保持在上次為 ON 的狀態，而此時系統便會判斷條件已成立，並立即將執行權轉移至下一個步進點。

若以下圖的程式而言，較佳的方式則是加上「ELSE」的描述，以確保 Transition 符號可以被指定一個確切的邏輯狀態；而除了 IF 結構之外，使用 CASE 結構時亦必須留意相同的問題。

```
0001 IF (M0 AND M1) THEN
0002     Tran4 := TRUE;
0003 END_IF;
```

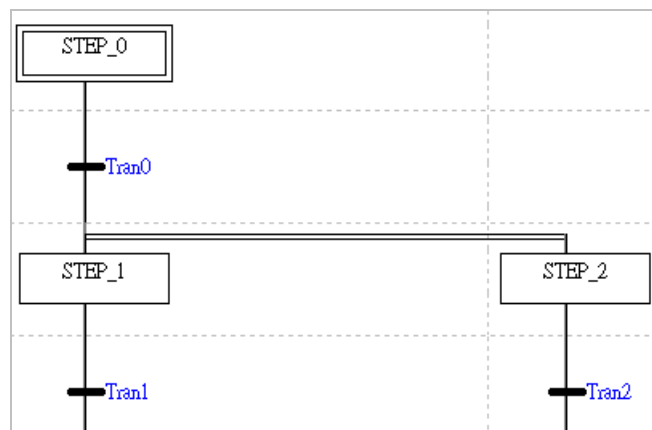
14.2.3 共進分歧與選擇分歧

在 SFC 程式中可以有並行處理的程序，這樣的結構我們便稱之為**分歧**，而分歧結構依執行的機制可分為**共進分歧**與**選擇分歧**。

● 共進分歧

下圖為共進分歧的圖例，其中分歧的步進點會刻意用雙線來進行連結；而共進分歧的特徵便是所有分歧的步進點會共用同一個轉換點，當該轉換點的條件成立之後，下方的所有步進點就會同時被啟動，亦即所有的平行路徑都會執行。

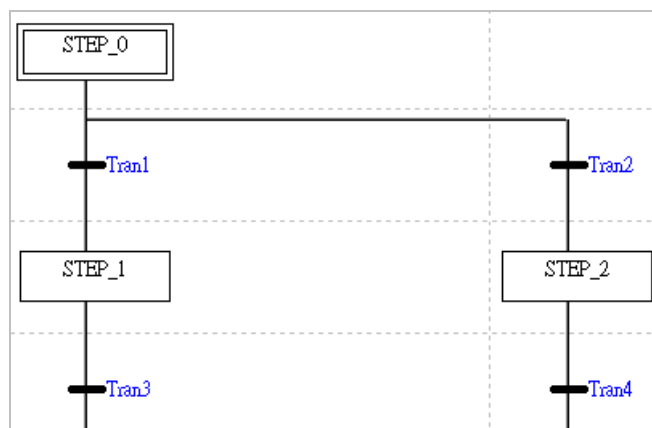
如下圖，當轉換點 Tran0 成立後，STEP_0 便會關閉，而 STEP_1 及 STEP_2 則會一起啟動。



● 選擇分歧

下圖為選擇分歧的圖例，其特徵為所有分歧的步進點都會有各自的轉換點，當其中某一個轉換點的條件成立之後，執行權就會移轉至該轉換點下方的步進點，而其他的轉換點則不再繼續判斷，因此在選擇分歧的平行路徑中，同時間只有一條路徑的步進點程序會被執行。因為轉換點的判斷順序是由左至右，因此如果有多個轉換點條件同時成立時，系統仍只會認定位於最左邊的轉換點。

以下圖為例，當轉換點 Tran2 的條件成立後，STEP_0 便會關閉並將 STEP_2 啟動，而此時的 Tran1 則會停止判斷並讓 STEP_1 繼續保持在不啟動的狀態。

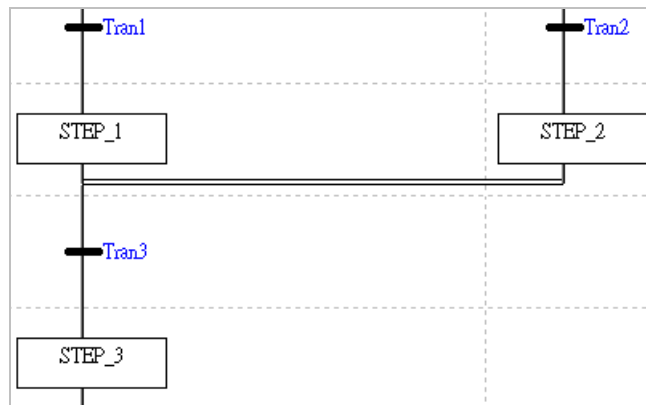


14.2.4 共進合流與選擇合流

在 SFC 的分歧結構中，最後仍須將所有分歧的路徑再重新匯整在一起，而這樣的匯整程序便稱之為合流；如同分歧結構一般，合流依執行的機制亦可分為共進合流與選擇合流。

● 共進合流

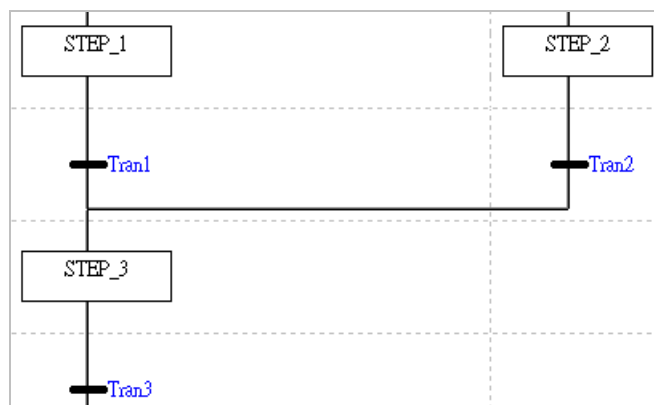
在圖示中，共進合流的步進點之間亦刻意用雙線來連結，而其特徵則為所有分歧的步進點都會聯結至同一個轉換點，因此必須等待連接至該轉換點上方的所有步進點都被啟動後，轉換點才可進行判斷。如下圖的範例，須先等待 STEP_1 及 STEP_2 都被啟動後，轉換點 Tran3 才會開始判斷，而當 Tran3 的條件成立時，STEP_1 及 STEP_2 便會一起被關閉，並啟動 STEP_3。



● 選擇合流

下圖為選擇合流的圖例，其特徵為所有合流的步進點都會有各自的轉換點，而只要其中某一個轉換點的條件成立之後，執行權就會先行移轉至下一個的步進點；但若平行路徑中仍有其他被啟動的步進點時，則所有被啟動的步進點仍會繼續執行，一直到其合流的轉換點成立之後，便會再次啟動合流下方的步進點，因此選擇合流有可能造成合流點下方的循序流程同時有多個步進點被啟動，使用上請務必確認是否會造成程序上的錯亂。

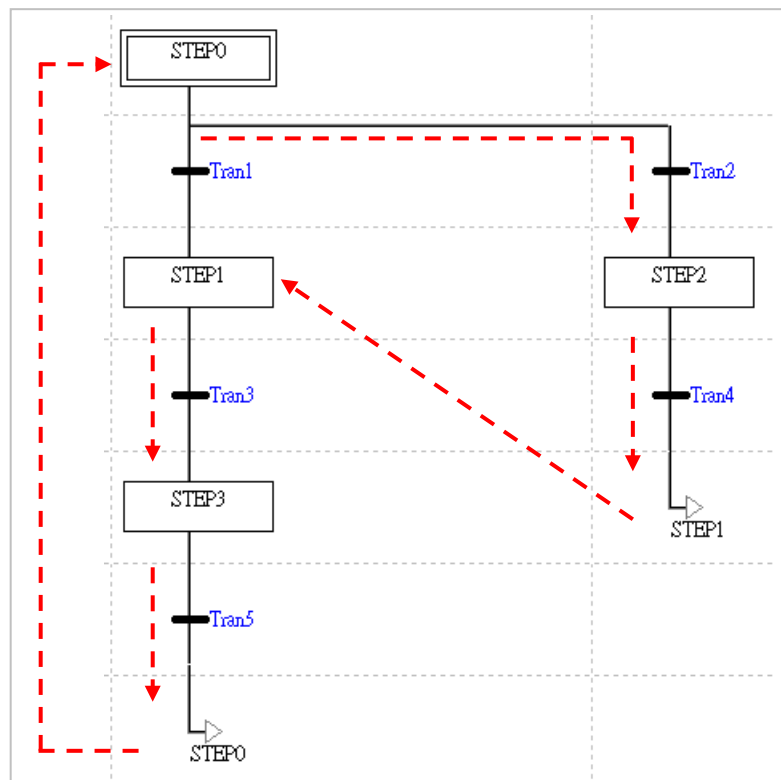
如下圖，當轉換點 Tran1 成立時，STEP_1 便會關閉並直接將 STEP_3 啟動，而後續當 Tran2 的條件也成立時，STEP_2 便一樣會被關閉，且不論 STEP_3 的狀態為何，系統仍會再次啟動 STEP_3。



14.2.5 跳躍點

SFC 允許使用跳躍的結構來將程式中的任一個步進點指定為執行權轉移的對象，且在 SFC 的圖形中至少會有一個跳躍點，也就是在圖形最下方決定程序循環起點的位置。因為使用跳躍功能的自由度相當高，所以使用時請務必確認不會造成程序上的錯亂。

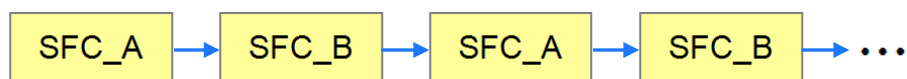
下圖為一個包含選擇分歧結構的 SFC 程式，此處假設轉換點 Tran2 的條件先行成立，因此執行的路徑便會先轉移至右側，而當 Tran4 成立之後，程序便會經由跳躍功能轉移至左側的 STEP1；之後待程序執行到最下方且當 Tran5 的條件成立時，程序便會重新由 STEP0 開始繼續執行。



使用步進點跳躍時須注意所指定的目標必須為同一 SFC 圖形中有使用的步進點，若該步進點未於同一 SFC 程式中被使用時，即使該步進點已宣告於區域或全域符號表，編譯時仍會產生錯誤。

不過若在系統的規劃上需要從某個 SFC 跳躍至另一個 SFC 的程式當中繼續執行時，在設計上可善用步進點與轉換點的特性來達成，例如可將跳躍目標的步進點或轉換點符號宣告於全域符號表，並在達成跳躍條件時將該符號設置為 ON，而此段程序則可寫於某個 Action 的程式當中。

如下列範例，我們希望程式運作時會先執行 **SFC_A** 的程式，而當 **SFC_A** 的所有步序都完成後便跳躍至 **SFC_B** 的程式當中繼續執行；而同樣的，當 **SFC_B** 的所有步序都完成後便重新跳回 **SFC_A** 以繼續下個循環；而在設計上我們則可透過轉換點的特性來達成。

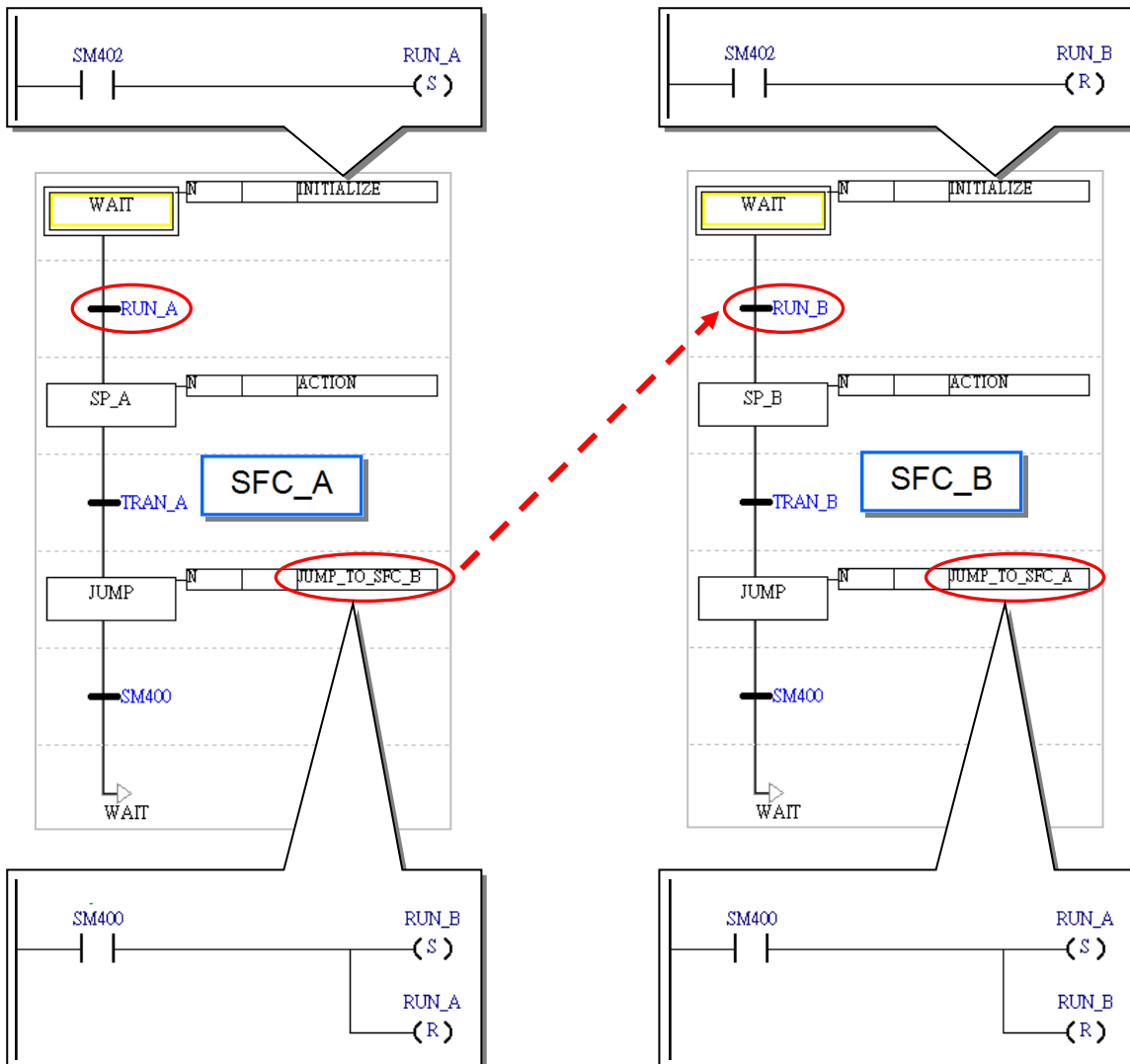


為了便於說明，下列的範例僅會用步進點「SP_A」與「SP_B」來表達原本要執行的 SFC 程式，且在其前後也都加上了控制用的步進點，並將轉換點「RUN_A」與「RUN_B」宣告於全域符號表中。當程式執行時，兩個 SFC 都會進入步進點「WAIT」，並執行「INITIALIZE」的 Action；且因 SFC_A 中的「INITIALIZE」會在第一個掃描週期中將轉換點「RUN_A」設置為 ON，因此 SFC_A 的程式便會在隨後轉移至步進點「SP_A」中，並執行預定的程序；而 SFC_B 中的「INITIALIZE」則會在第一個掃描週期中將轉換點「RUN_B」設置為 OFF，因此 SFC_B 便會停留在「WAIT」當中繼續等待。

當 SFC_A 執行至步進點「JUMP」時，其 Action 會將轉換點「RUN_B」設置為 ON，並將「RUN_A」重置為 OFF，而當轉換點「RUN_B」設置為 ON 後，SFC_B 的程式便可轉移至步進點「SP_B」中開始執行其預定的程序；除此之外，因為「JUMP」下方的轉換點是保持為 ON 的特殊繼電器，因此 SFC_A 的步序會立即跳回「WAIT」步進點，且因其轉換點「RUN_A」已被重置為 OFF，因此 SFC_A 便會停留在「WAIT」步進點當中，並等待 SFC_B 的程序再次將「RUN_A」設置為 ON。

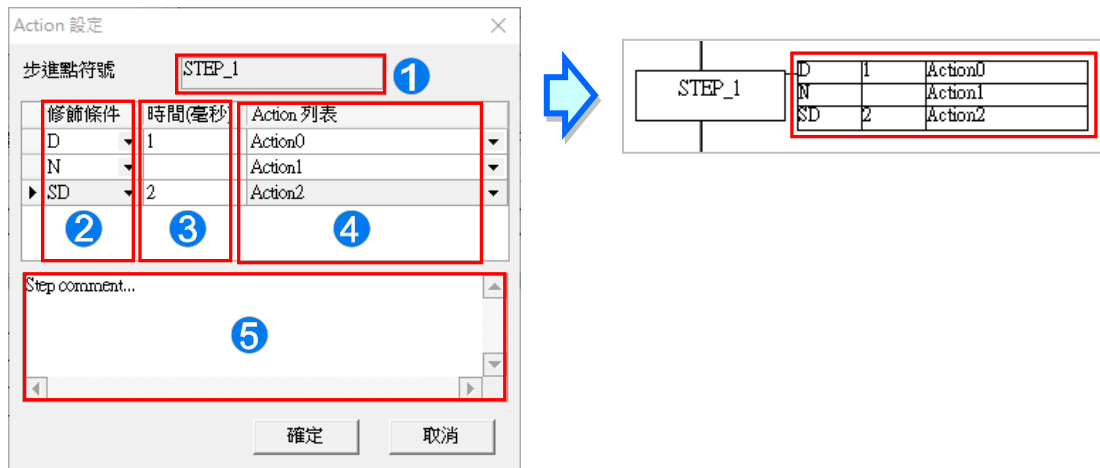
在這個架構中，步進點「WAIT」與「JUMP」的功能只用於切換彼此的轉換點，其餘的時間在實際意義上都是處於空轉的狀態；而當程式所應用的機種為 AH/AS 系列時，在規劃上還可將兩個步進點當中的 Action 指定為「P」或「P1」的修飾條件，相關細節將於第 14.2.6 節當中另行介紹。

14



14.2.6 Actions 的修飾條件

在 ISPSOft 中，使用者可透過對 Action 的修飾，來對其執行方式加以定義，不過目前的 DVP 系列機種無法設定修飾條件，因此都將被視為條件 N，即一般的執行方式。



- ① 顯示此組設定所屬的步進點名稱。
- ② 於此選擇欲指定的修飾條件。
- ③ 當選擇與時間相關的修飾條件時，於此可設定時間屬性。
- ④ 於此選擇被修飾的 Action。
- ⑤ 此欄位可編輯此組設定的注解說明。

14.2.6.1 條件定義

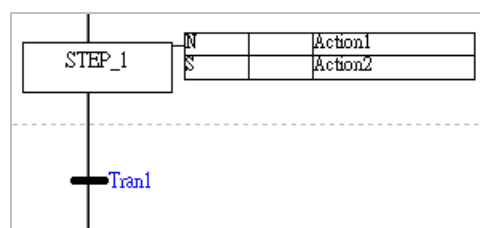
● N (Normal)

此為一般的執行方式，當步進點啟動時，動作便會立即執行，而當步進點因轉移而失效後，系統便會再執行一次 **Final Scan**，並將所有的動作與輸出停止，不過對於使用類似 SET 指令的輸出則會保持原狀。目前 DVP 系列機種的所有 Action 都將指定為此種條件。

● S (Set)

當步進點啟動時，動作便會立即執行，且一旦 Action 開始執行之後，即使步進點已因執行權轉移而失效，該 Action 仍會保持執行狀態。

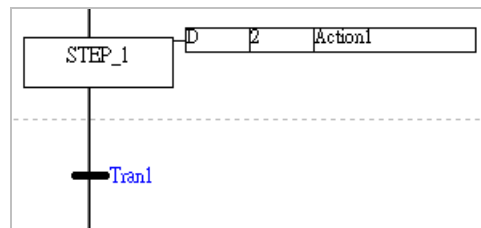
如下圖，當 STEP_1 啟動時，Action1 與 Action2 便會立即執行，但當 Tran1 條件成立而讓 STEP_1 失效時，Action1 便會停止動作，並執行 **Final Scan**，而 Action2 則會繼續維持執行狀態。



● D (Delay)

當步進點啟動時，須延遲一段時間後 Action 才會開始執行；但若轉換條件在延遲時間到達之前即已成立，則此步進點將會直接關閉並轉移至下一步進點，而 Action 也不會執行，且到了下次這個步進點再次啟動時，延遲的時間將會重新開始計時。

如下圖，當 STEP_1 啟動後，Action1 並不會馬上執行，而必須在延遲 2 秒之後，Action1 才會開始執行動作，但若 STEP_1 在延遲時間到達前即已失效，Action1 便不會執行動作。



● SD (Set Delay)

當步進點啟動時，需延遲一段時間後 Action 才會開始執行，而一旦 Action 開始執行之後，不論步進點是否已經轉移，該 Action 仍會保持執行狀態。除此之外，即使轉換條件在延遲時間到達之前即已成立，且完成了步進點的轉移，然而延遲計時的動作仍會繼續，並在計時到達之後開始執行 Action 的動作。

● DS (Delay Set)

此條件與 SD 類似，差別在於當轉換條件在延遲時間到達之前就已成立時，設定為 DS 條件的延遲計時動作便會停止，且 Action 也不會執行，而到了下次這個步進點再次啟動時，延遲的時間便會重新開始計時。不過一旦 Action 開始執行之後，不論步進點是否已經轉移，該 Action 仍會保持執行狀態。

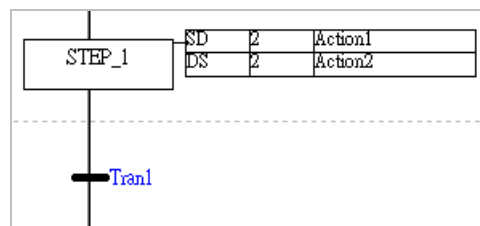
如下圖，當 STEP_1 啟動後，Action1 與 Action2 並不會馬上執行，而必須在延遲 2 秒之後才會開始動作，不過依據 STEP_1 維持啟動狀態的時間，可分別考慮下列兩種狀況。

(a) STEP_1 維持啟動狀態的時間大於 2 秒

Action1 與 Action2 在時間到達 2 秒時便會開始執行，且之後即使步進點 STEP_1 因步進點轉移而失效後，Action1 與 Action2 仍會維持執行狀態。

(b) 步進點 STEP_1 維持啟動狀態的時間小於 2 秒

Action1 在步進點轉移之後仍會繼續計時，而在時間到達 2 秒時便會開始執行並維持執行狀態；而 Action2 則會在步進點轉移後停止計時，並且也不會執行 Action2 的動作。



● L (Limit)

Action 於步進點啟動時便會立即執行，但當執行時間到達設定值時，即使步進點尚未轉移，Action 還是會自動停止，並執行 **Final Scan**；而若步進點在時間到達之前即已轉移，Action 便會隨之停止。

● SL (Set Limit)

當步進點啟動時，動作便會立即執行，但會固定執行時間。當 Action 的執行時間已到達設定值時，即使步進點尚未轉移，Action 仍會自動停止，並執行一次 **Final Scan**；而當步進點在時間到達之前即已轉移時，Action 還是會等待執行時間到達之後才會停止。

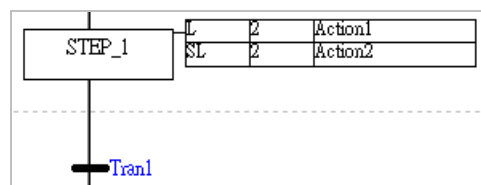
如下圖，當 STEP_1 啟動時，Action1 與 Action2 便會開始執行，此處再分別考慮下列兩種狀況。

(a) 步進點 STEP_1 維持啟動狀態的時間大於 2 秒

Action1 與 Action2 在時間到達 2 秒時便會自動停止動作，並執行 **Final Scan**。

(b) 步進點 STEP_1 維持啟動狀態的時間小於 2 秒

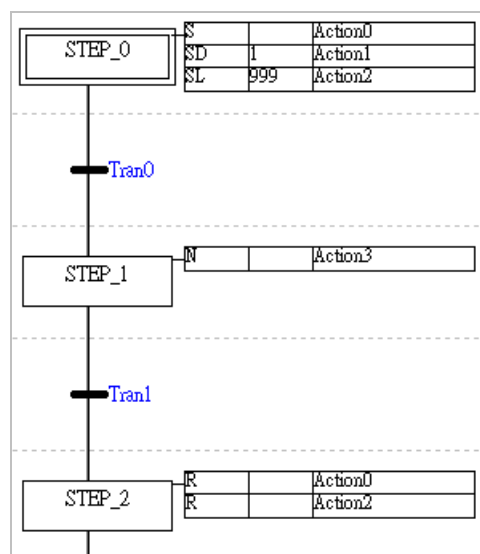
Action1 在步進點轉移之後便會停止動作並執行 **Final Scan**，但 Action2 仍會繼續等待時間到達 2 秒之後才會停止。



● R (Reset)

停止 Action 並執行 **Final Scan**，且即使是由 S、SD、DS 或 SL 條件來啟動的 Action 都會被關閉。

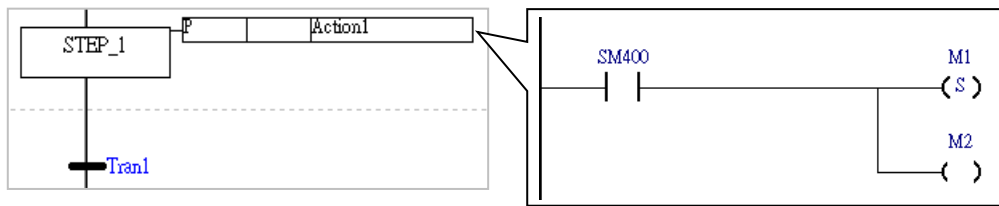
如下圖，假設 Action0、Action1 與 Action2 在 STEP_0 為啟動狀態時即已順利執行動作，並且在步進點轉移至 STEP_2 之前都會維持執行狀態；當步進點轉移至 STEP_2 時，Action0 與 Action2 的動作都將被停止，並執行 **Final Scan**，而 Action1 則因未被重置，所以仍會保持執行狀態。



● P (Pulse)

在步進點啟動後的第一個掃描週期，動作便會立即執行，但到了第二個週期時，即使步進點仍在啟動狀態，所有的動作還是會被停止，並執行 **Final Scan**。

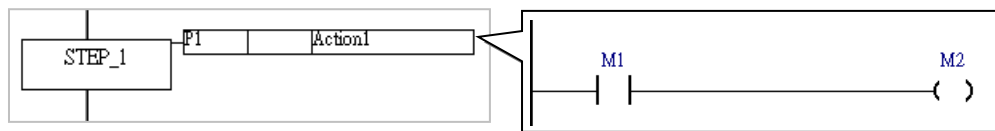
如下圖，在 STEP_1 啟動後的第一個掃描週期 Action1 便會開始執行，而因 SM400 的狀態為 ON，因此 M1 與 M2 也都會為 ON；而到了第二個掃描週期時，系統便會將 Action1 的動作關閉，此時即使 SM400 還是為 ON，M2 都會變成 OFF，不過因為 M1 使用 SET 輸出，所以仍會保持在 ON 的狀態。



● P1 (Raising Pulse)

設定此條件的 Action 只會在步進點啟動後的第一個掃描週期執行一次，且也不會執行 **Final Scan** 的程序；因此當 Action 的內部有使用 OUT (如下圖) 或 TMR 等需要將動作關閉的指令時，若將其賦予 P1 的修飾條件，在使用上仍是允許的，但編譯時會出現警告。

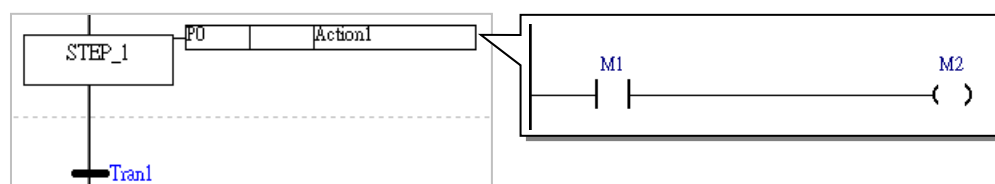
下圖的 Action1 在 STEP_1 啟動後的第一個掃描週期便會執行動作，此時若 M1 為 ON，則 M2 也會為 ON，但因使用 P1 修飾的 Action 只會在該步進點啟動後的第一個掃描週期中執行一次，並且在之後也不會執行 **Final Scan** 的程序，因此後續不論 M1 的狀態為何，M2 都將不再隨之改變狀態。



● P0 (Falling Pulse)

設定此條件的 Action 只會在步進點剛失效的掃描週期中執行一次，且之後也不會執行 **Final Scan** 的程序；因此當 Action 的內部有使用 OUT (如下圖) 或 TMR 等需要將動作關閉的指令時，若將其賦予 P0 的修飾條件，在使用上仍是允許的，但編譯時會出現警告。

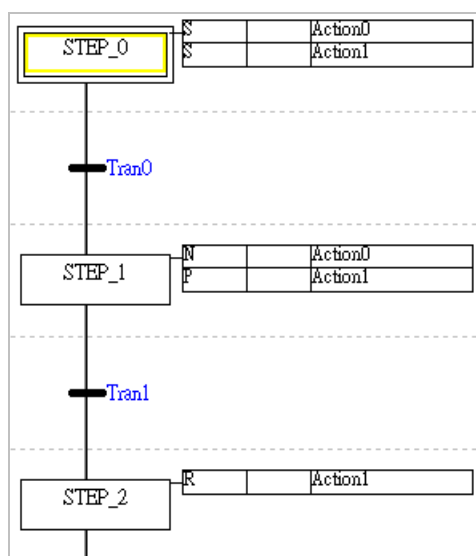
下圖的 Action1 在 STEP_1 啟動後仍不會執行動作，而必須等待 Tran1 的條件成立而使得 STEP_1 失效時才會執行一次，此時若 M1 為 ON，則 M2 也會為 ON，且因使用 P0 修飾的 Action 只會在該步進點失效的掃描週期中執行一次，並且在之後也不會執行 **Final Scan** 的程序，因此之後不論 M1 的狀態為何，M2 都將不再隨之改變狀態。



14.2.6.2 設定修飾條件的注意事項

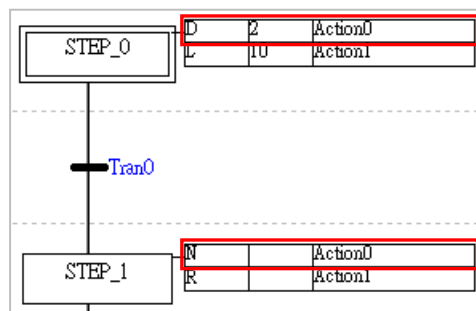
- 當 Action 一旦由 S 條件來啟動之後，即使於他處將該 Action 設定為 N、P 或其它會將 Action 停止的條件時，該 Action 仍會維持在執行的狀態，而唯有在被賦予條件 R 並順利執行之後，該 Action 才會被重置。因此當程式中的 Action 曾被賦予 S 的條件屬性時，請務必確認是否須配置條件 R 來將該 Action 重置。

如下圖，Action0 與 Action1 在 STEP_0 啟動時便會開始執行並維持其執行狀態，而當步進點轉移至 STEP_1 時，即使兩者分別被賦予 N 與 P 的修飾條件，Action0 與 Action1 仍會繼續執行；直到步進點轉移至 STEP_2 時，Action1 會因修飾條件 R 而被重置並停止動作，但 Action0 則因沒有任何 R 條件的修飾，以致在整個程式的執行過程中會一直保持在執行狀態。



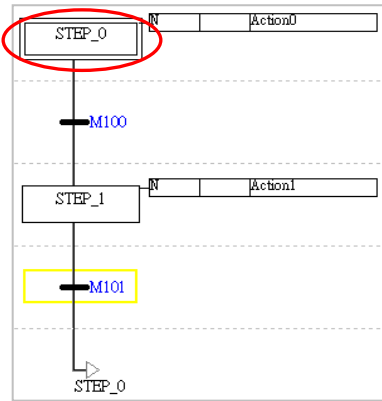
- 設定與時間相關的修飾條件的 Action，除了設定為 R 的修飾條件之外，只能在 SFC 程式中指定一次。也就是當某個 Action 被設定為 D、SD、DS、L 或 SL 的其中一個條件時，除了設定為 R 之外，該 Action 就不可再被使用。

如下圖範例，Action0 與 Action1 在 STEP_0 的步序中都分別被賦予與時間相關的修飾條件，而在 STEP_1 的步序中，Action1 則被賦予條件 R，因此其用法是正確的；但 Action0 卻又再度被賦予非 R 的修飾條件，因此 Action0 的部份在程式進行編譯時便會產生錯誤。



14.2.7 初始步進點

在 SFC 的執行過程中，因每個步進點都會輪流地被啟動且持續循環，因此必須定義一個唯一的步進點做為程式第一次執行的啟始位置，而這個步進點便稱之為**初始步進點**。在 ISPSOft 中，初始步進點會以雙線的方框來表示，且每個 SFC 程式只能有一個初始步進點。

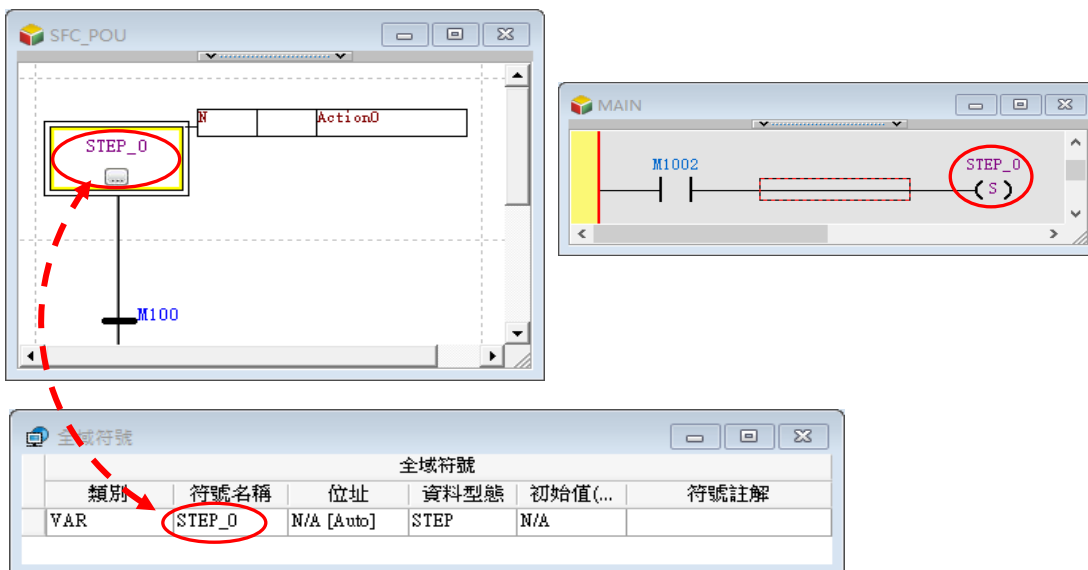


14

AH/AS 系列機種可指定 SFC 程式中的任一個步進點做為初始步進點，且在 PLC 啟動運轉後，當第一次執行至該 POU 時便會自動由指定的初始步進點開始執行。

而在 DVP 系列的機種中，其 SFC 的 POU 並無法自動執行，而必須藉由其他的 POU 對 SFC 中的某個步進點執行 SET 指令之後，該 SFC 才會開始運轉；因此在 DVP 機種的 SFC 圖形中無法指定初始步進點，且會固定在圖形中的第一個位置；而實際運作時，真正的初始步進點則決定於在其他 POU 中被下達 SET 指令的步進點裝置或符號。下列則為 DVP 機種的 SFC 啟動範例。

首先，請決定該 SFC 的實際初始步進點，並將初始步進點的變數符號宣告於全域符號表，接著再於其它的 POU 當中對該初始步進點使用 SET 指令；而在對步進點符號下達 SET 指令的程式碼中，須注意其前方的條件接點或邏輯輸出，要避免在程式執行的過程中還會變為 ON 的狀態，否則便可能再次將該步進點設置為 ON 而造成 SFC 的執行程序產生錯亂；建議可以使用特殊繼電器 M1002，而該繼電器的特性則是只有在主機啟動後的第一個掃描週期才會為 ON。



14.2.8 內部屬性

為了讓使用者在編程時，能更容易使用 SFC 某些屬性參數，內部屬性供使用者在各 POU 之間調用。這些屬性參數可分為 Step、Action 及 Transition 三種類別，本章節將會對此三種類別內容逐一介紹。

14.2.8.1 Step 屬性參數

若使用者需要調用 SFC 的 POU 中 Step 的屬性參數，目前屬性參數的架構，是以結構(STRUCTURE)方式來呈現。

此結構由「Step 所在 SFC 之 POU 名稱」、「Step 的名稱」及「Step 的成員」組成，使用者可以選擇適當的 POU 及 Step 名稱後，再決定所需使用 Step 的成員。



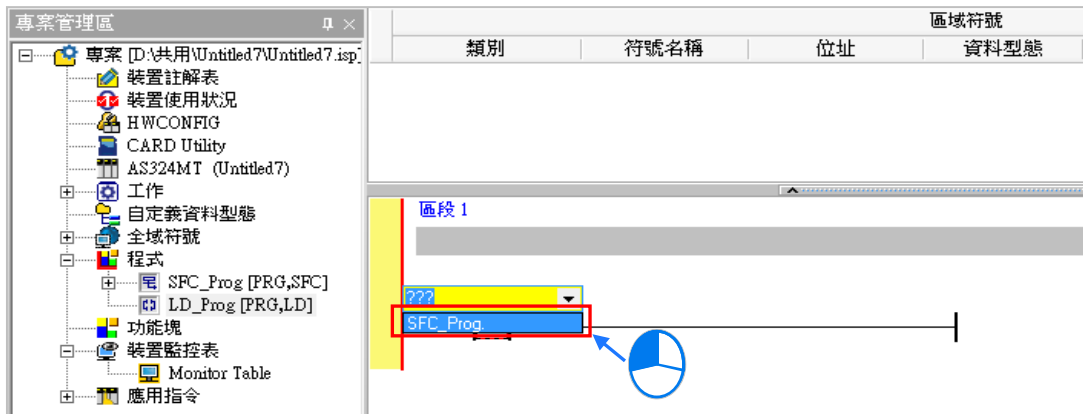
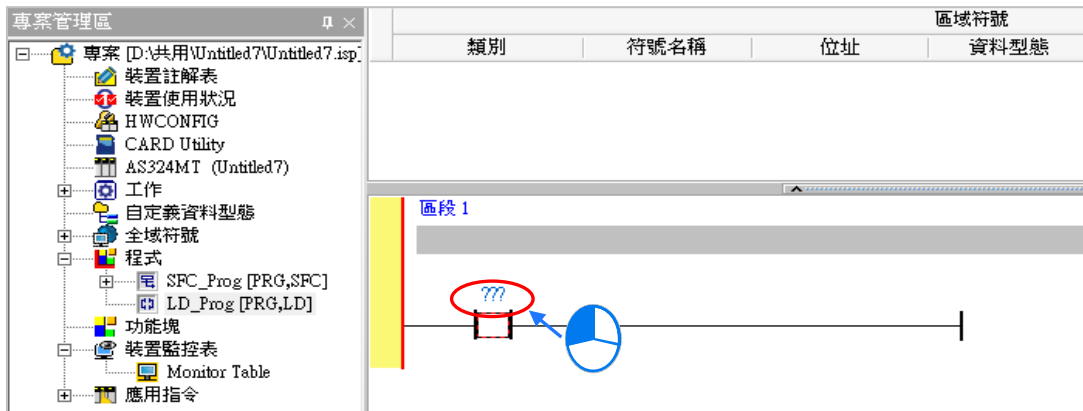
14

下表為 Step 的成員分類及作用一覽表。

類型	成員	說明	讀	寫
Step	X	當 Step 正在執行，此 SFC 屬性參數將為 ON；此 SFC 屬性參數用法將取代過去 S 裝置	V	X
	done	當 Step 之 T 屬性（目前 Step 已執行時間） \geq Step 之 timePreset 屬性，此值將為 ON	V	X
	counter	當 Step 之 X 屬性由 OFF 變為 ON，此屬性參數計數加 1	V	X
	timePreset	決定 Step 之 done 屬性由 OFF 到 ON 所需經過的時間(單位: ms，資料長度: DWORD)	V	X
	maxTime	記錄 Step 之 T 屬性，曾經計時最大時間(單位:ms，資料長度: DWORD)	V	X
	T	目前 Step 已執行多少時間(單位:ms，資料長度: DWORD)	V	X

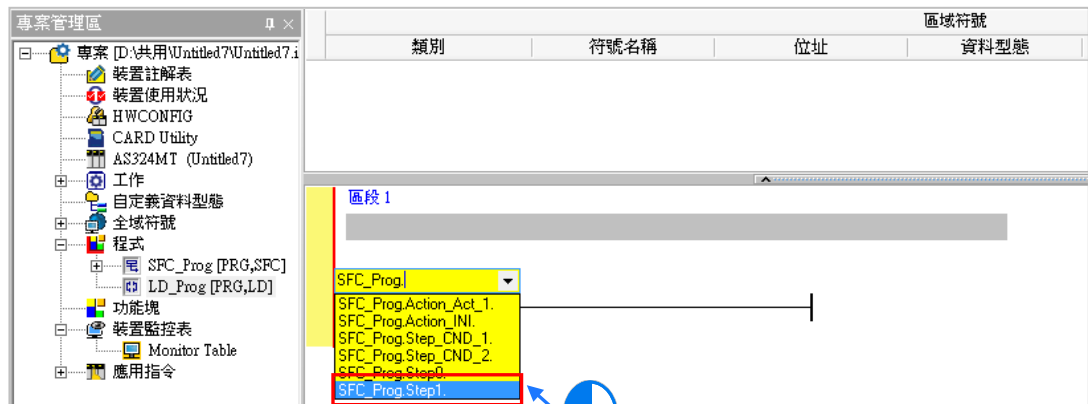
使用 SFC 之 Step 的成員 X 的操作使用為例，若希望在撰寫之 SFC 程式 POU 的 Step 開始執行時，觸發 Ladder 程式 POU 中的常開接點為 ON，程式撰寫的方式，說明如下。

可以在 Ladder 程式 POU 中使用一常開接點，並於常開接點位置上方，點擊滑鼠左鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 的名稱，接著點擊 SFC 程式 POU 的名稱。

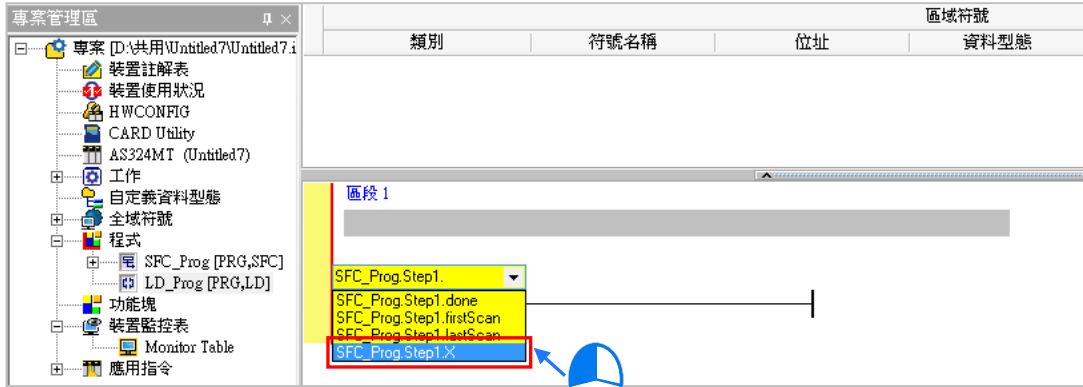


14

點擊 SFC 程式 POU 的名稱後，再點擊鍵盤右鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 含 Step 的名稱，接著點擊欲選擇的 SFC 程式 POU 含 Step 的名稱。

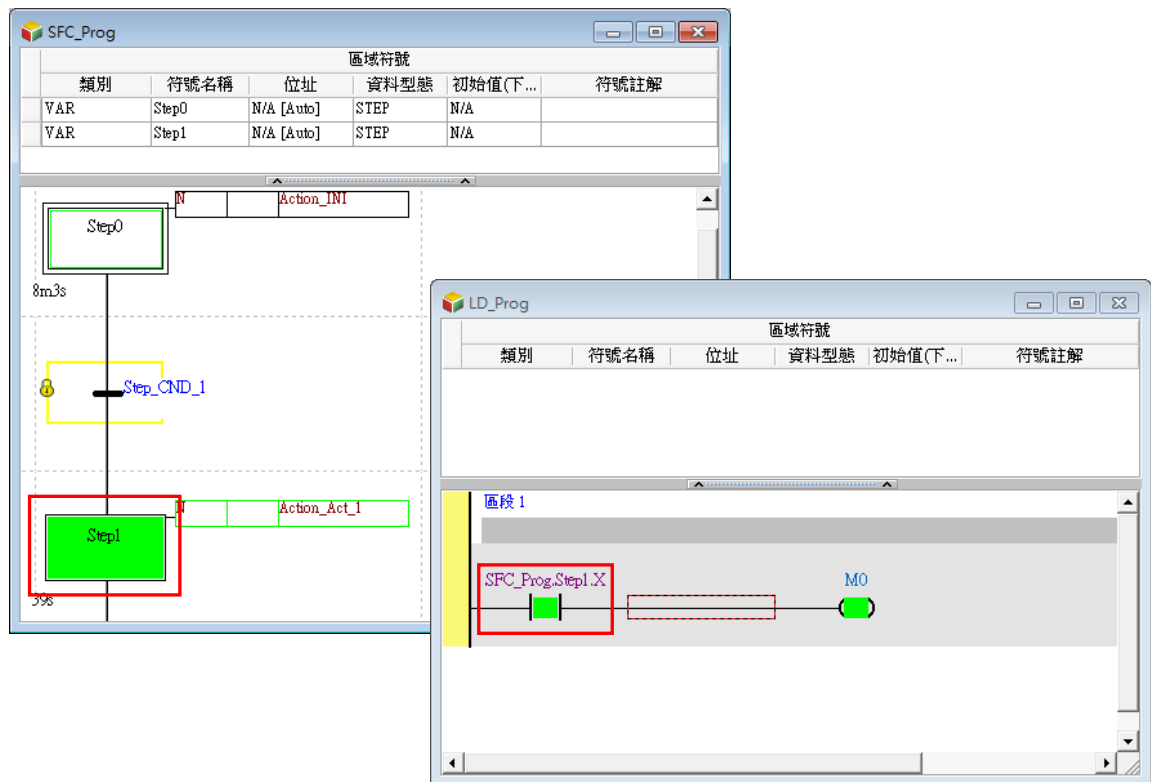


點擊 SFC 程式 POU 含 Step 的名稱後，再點擊鍵盤右鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 含 Step 含成員的名稱，接著點擊欲選擇的 SFC 程式 POU 含 Step 含成員的名稱，即完成選取。



14

當在執行 SFC 程式 POU 的 Step 時，SFC_Prog.Step1.X 此常開接點將會 ON。

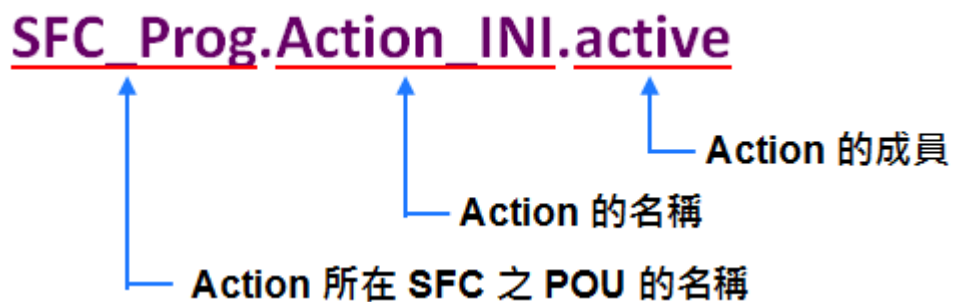


14

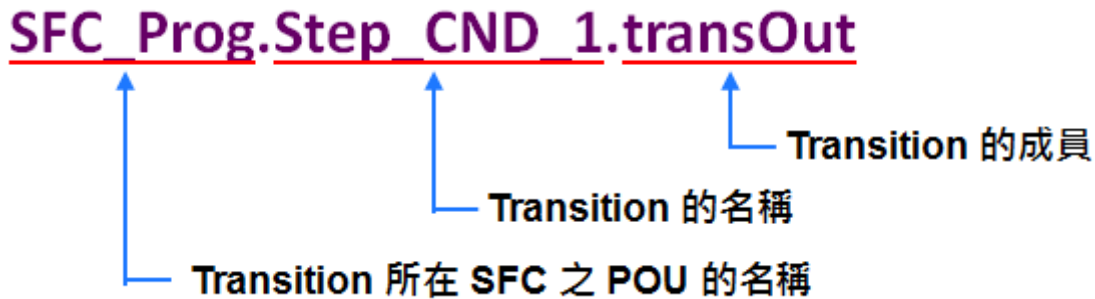
14.2.8.2 Action 與 Transition 屬性參數

若使用者需要調用 SFC 的 POU 中，Action 或 Transition 的屬性參數，目前屬性參數的架構，是以結構 (STRUCTURE) 方式來呈現。

Action 屬性參數結構由「Action 所在 SFC 之 POU 名稱」、「Action 的名稱」及「Action 的成員」組成，使用者可以選擇適當的 POU 及 Action 名稱後，再決定所需使用 Action 的成員。



Transition 屬性參數結構由「Transition 所在 SFC 之 POU 名稱」、「Transition 的名稱」及「Transition 的成員」組成，使用者可以選擇適當的 POU 及 Transition 名稱後，再決定所需使用 Transition 的成員。

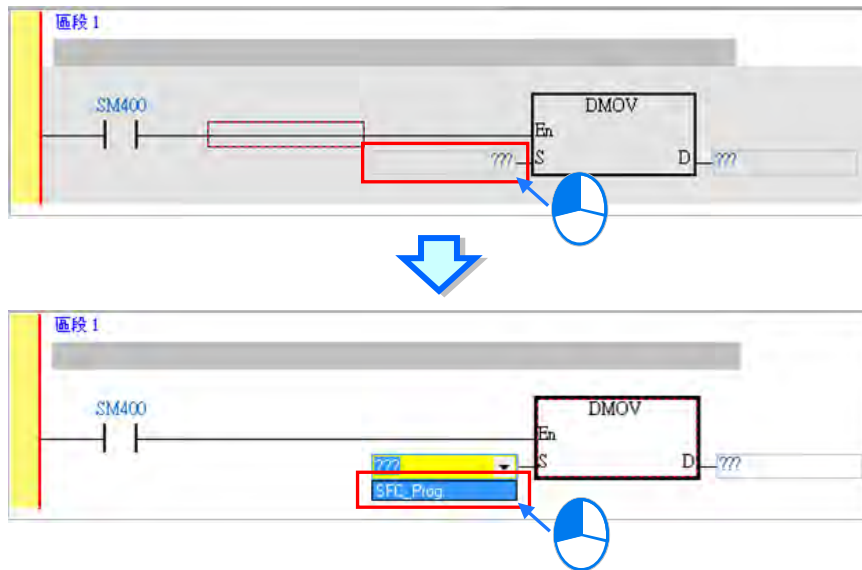


下表為 Action 及 Transition 的成員分類及作用一覽表。

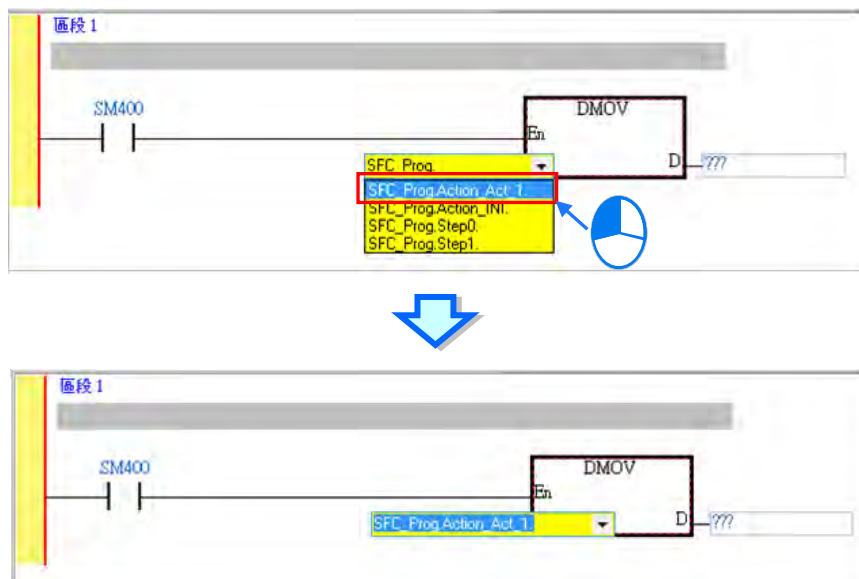
類型	成員	說明	讀	寫
Action	enabled	當 Action 的 qualifier 條件成立時，此 SFC 屬性參數為 ON，可看出 Action 是否啟用	V	X
	active	當完成第一次執行 Action 程式，此 SFC 屬性參數將為 ON	V	X
	rstDisable	當 Final scan 的功能啟用時，此值為 OFF 當 Final scan 的功能不啟用時，此值為 ON	V	X
	counter	當 Action 之 active 屬性由 OFF 變為 ON，此屬性參數計數加 1	V	X
Action	setTimer	跟時間相關修飾 (qualifier) 的時間顯示 (單位:ms，資料長度: DWORD)	V	X
Transition	transOut	當 Transition 移轉的條件成立為 ON，或 Transition 被強制啟動時為 ON	V	X

使用 SFC 之 Action 的成員 counter 的操作使用為例，若希望監看 Action 的 active 屬性參數，由 0 變為 1 的次數，並將此 active 屬性參數值傳至 D100 暫存器，程式撰寫的方式，說明如下。

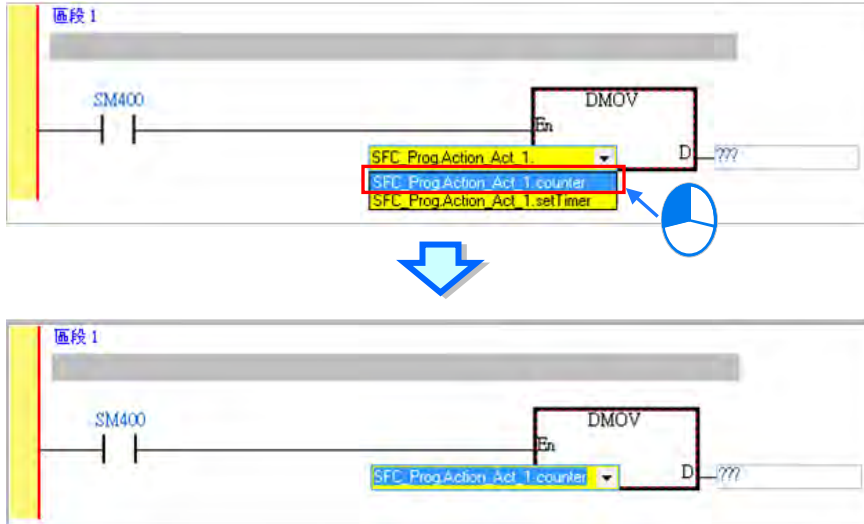
可以在 Ladder 程式 POU 中使用 SM400 常開接點及 DMOV 指令。對 DMOV 指令的 S 接腳，點擊滑鼠左鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 的名稱，接著點擊 SFC 程式 POU 的名稱。



點擊 SFC 程式 POU 的名稱後，再點擊鍵盤右鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 含 Action 的名稱，接著點擊欲選擇的 SFC 程式 POU 含 Action 的名稱。

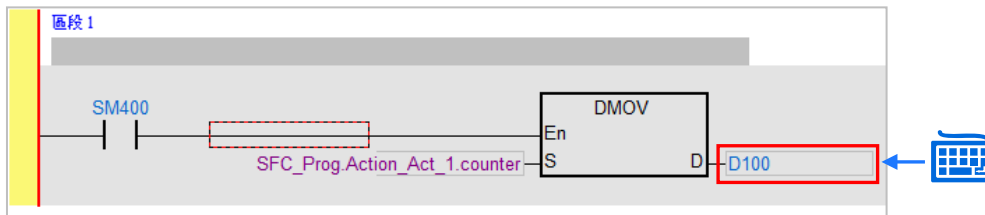


點擊 SFC 程式 POU 含 Action 的名稱後，再點擊鍵盤右鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 含 Action 含成員的名稱，接著點擊欲選擇的 SFC 程式 POU 含 Action 含成員的名稱，即完成選取。

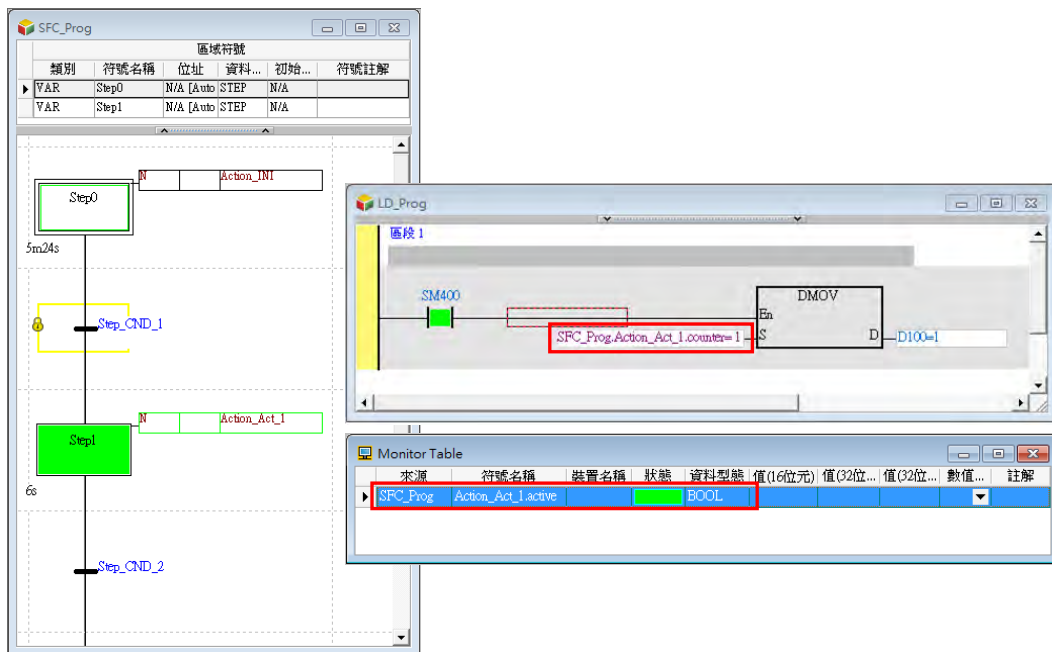
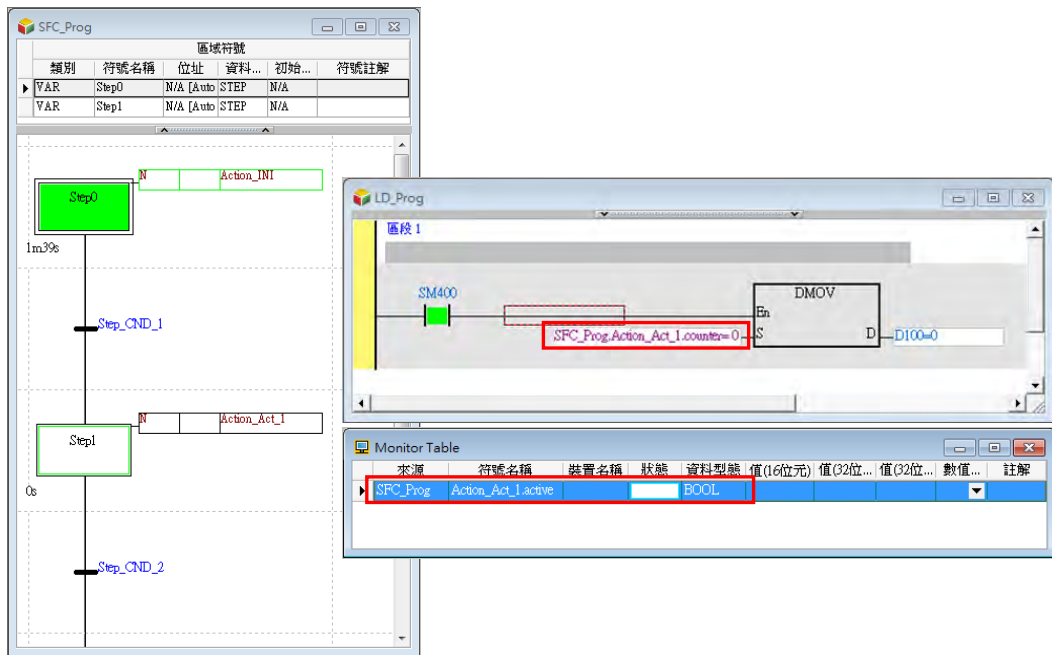


14

對 DMOV 指令的 D 接腳，輸入 D100，完成撰寫。



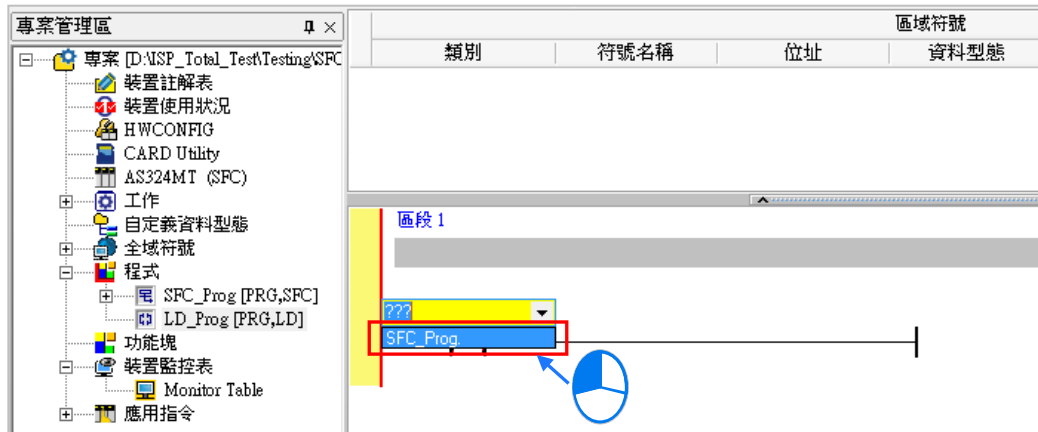
在 SFC_Prog.Action.Act_1.Active · 由 0 變為 1 時 · SFC_Prog.Action.Act_1.counter 計數將會加 1。



14

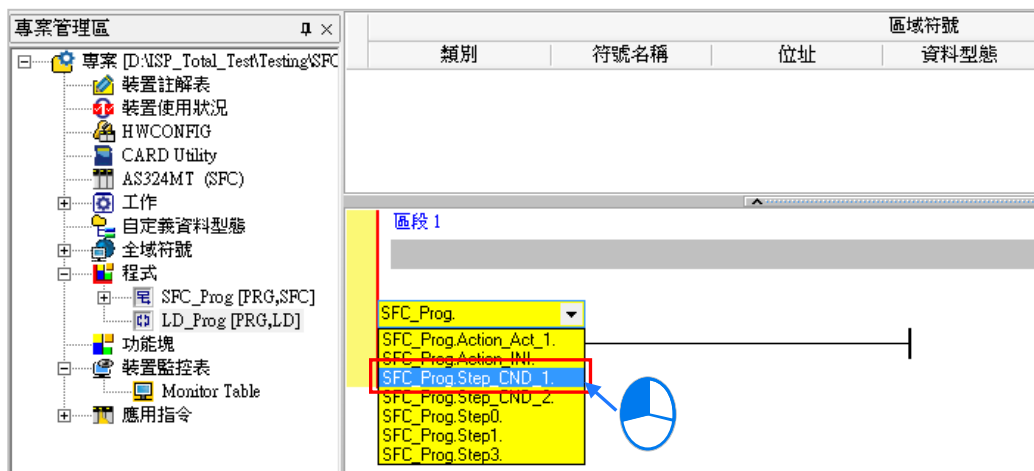
使用 SFC 之 Transition 的成員 transOut 的操作使用為例，若希望在撰寫之 SFC 程式 POU 的 Transition 移轉條件成立時，觸發 Ladder 程式 POU 中的常開接點為 ON，程式撰寫的方式，說明如下。

可以在 Ladder 程式 POU 中使用一常開接點，並於常開接點位置上方，點擊滑鼠左鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 的名稱，接著點擊 SFC 程式 POU 的名稱。



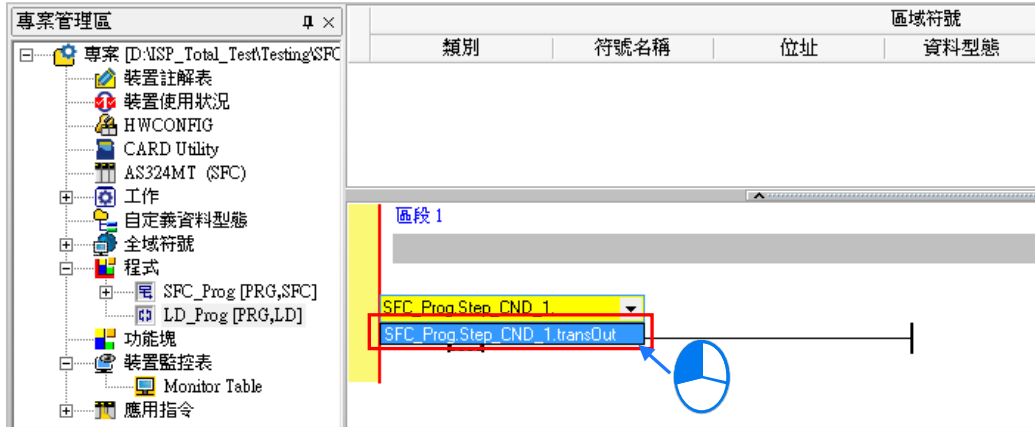
14

點擊 SFC 程式 POU 的名稱後，再點擊鍵盤右鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 含 Transition 的名稱，接著點擊欲選擇的 SFC 程式 POU 含 Transition 的名稱。



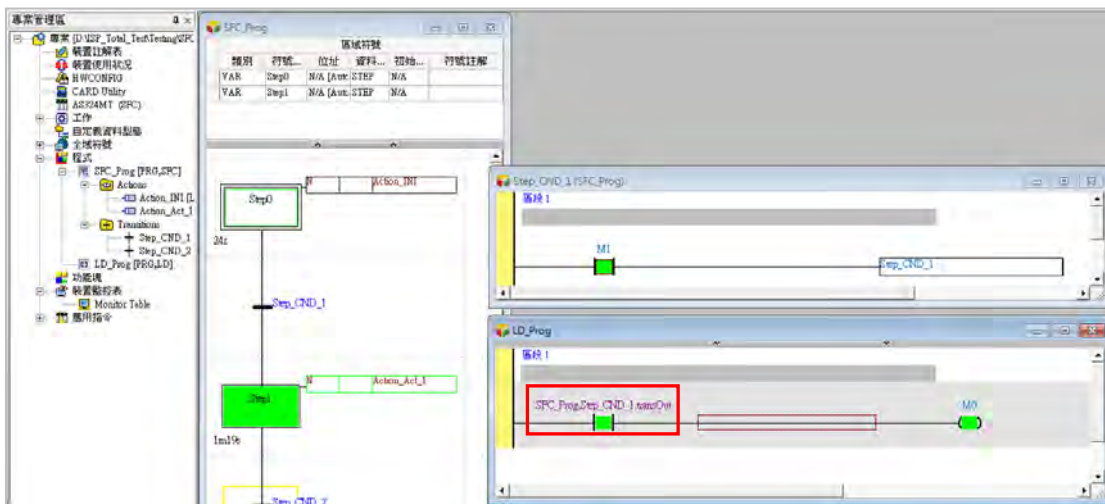
14

點擊 SFC 程式 POU 含 Transition 的名稱後，再點擊鍵盤右鍵，此時下拉選單將自動帶出可選擇的 SFC 程式 POU 含 Transition 含成員的名稱，接著點擊欲選擇的 SFC 程式 POU 含 Transition 含成員的名稱，即完成選取。



14

當在 SFC 程式 POU 的此 Step_CND_1 此 Transition 條件成立時，SFC_Prog.Step_CND_1.transOut 此常開接點將會 ON。

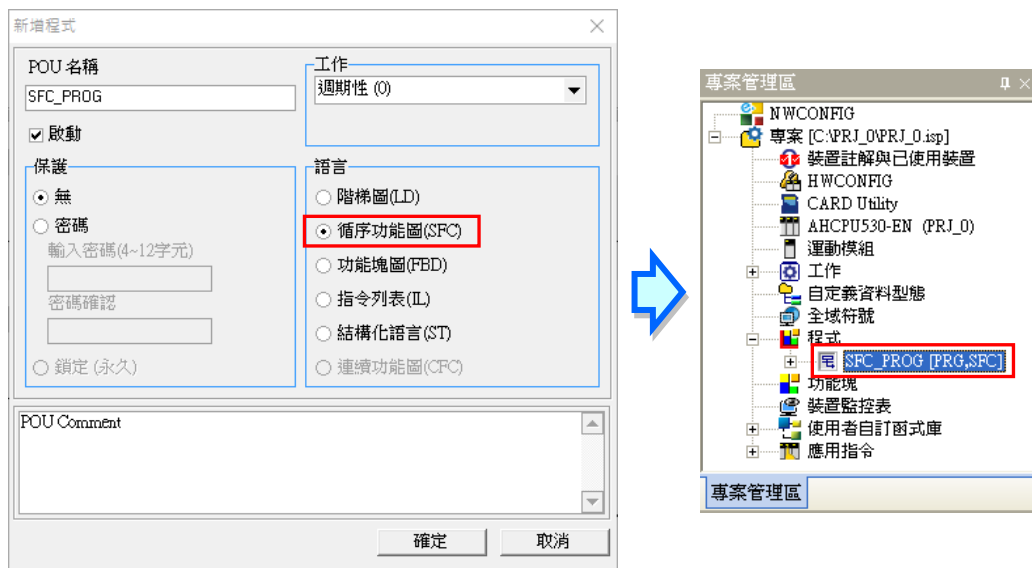


14.3 於 ISPSOft 中建立 SFC 程式

14.3.1 ISPSOft 的 SFC 編輯環境

在建立 POU 時，於語言欄位選擇「循序功能圖 (SFC)」便可建立 SFC 程式，更進一步的操作細節請參考第 5.4.1 節的說明。而除此之外，建立 SFC 程式時須再留意下列兩點事項。

- (a) 建立功能塊 POU 時不可選擇 SFC 語言。
- (b) SFC 語言的程式 POU 不得指派週期性以外的工作 (TASK)。



下圖為 ISPSOft 中的 SFC 編輯環境，基本的視窗配置與其他的語言相同，上方為區域符號表，下方則是 SFC 圖形的編輯工作區；而 SFC 的程式圖面則是以棋盤的方式來呈現，每個步進點或轉換點都會依序被放在圖面的方格中，當用滑鼠點擊程式圖面上的物件或是該物件所處的方格時，該物件便會被選取，並以黃色的外框來表示已被選取的狀態。



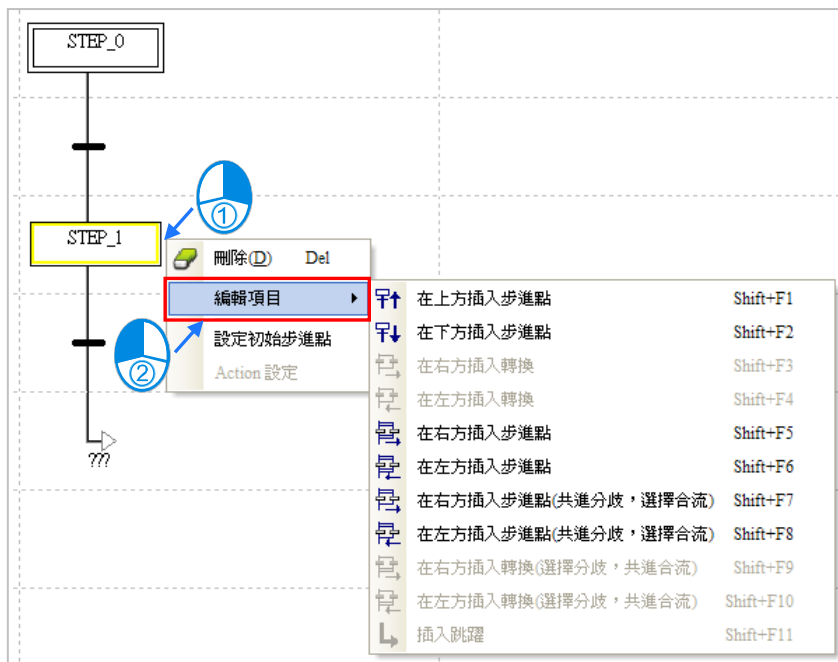
進入 SFC 的編輯環境後，ISPSOft 的視窗中便會出現對應的編程圖示工具列，其功能簡介如下。



圖示	鍵盤 (快捷鍵)	功能說明
	Shift + F1	新增步進結構於選取位置上方
	Shift + F2	新增步進結構於選取位置下方
	Shift + F3	於選取的轉換點右側並聯一新轉換點
	Shift + F4	於選取的轉換點左側並聯一新轉換點
	Shift + F5	於選取的步進點右側並聯一新步進點
	Shift + F6	於選取的步進點左側並聯一新步進點
	Shift + F7	於選取的步進點右側建立一 共進分歧&選擇合流的結構
	Shift + F8	於選取的步進點左側建立一 共進分歧&選擇合流的結構
	Shift + F9	於選取的轉換點右側建立一 選擇分歧&共進合流的結構
	Shift + F10	於選取的轉換點左側建立一 選擇分歧&共進合流的結構
	Shift + F11	插入一跳躍點

14

除了可透過圖示工具列來進行編輯之外，當於編輯工作區中點選某個物件後按下滑鼠右鍵，於快捷選單的「編輯項目」下一樣會有相同的操作項可供使用，而為了確保圖形的完整性與合法性，隨著點選位置的不同，可使用的圖形編輯功能也會跟著不同。



14.3.2 Actions 與 Transitions 的建立與管理

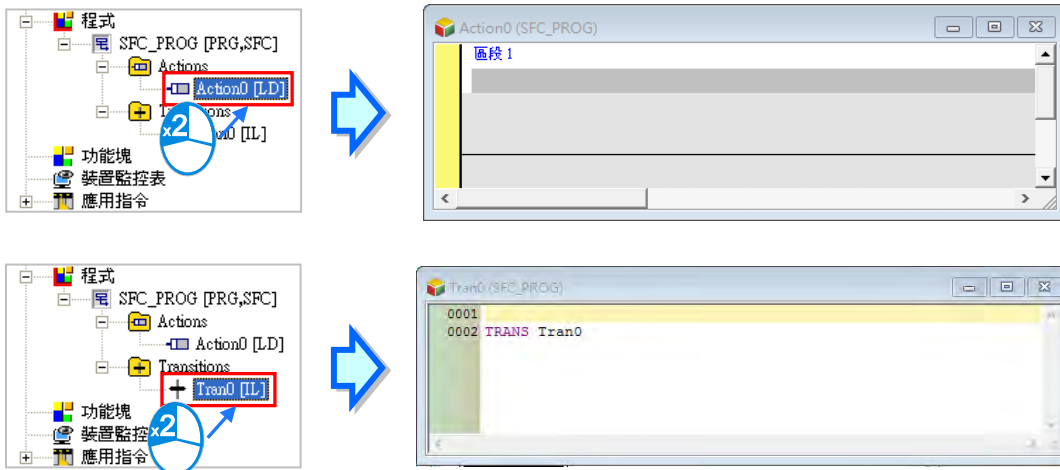
當要新增 Action 的項目時，請先於專案管理區中展開 SFC 的程式 POU，並點選其下方的「Actions」項目後按下滑鼠右鍵，接著再於快捷選單中點選「新增」，最後再於彈出的設定視窗中設定 Action 的名稱以及開發此 Action 時所使用的編程語言即可。



而要新增 Transition 的項目時，請同樣先於專案管理區中展開 SFC 的程式 POU 後，於其下方點選「Transitions」項目後按下滑鼠右鍵，接著再於快捷選單中點選「新增」，最後再於彈出的設定視窗中設定 Transition 的名稱以及開發此 Transition 時所使用的編程語言即可。



於專案管理區中建立 Action 與 Transition 的項目後，直接於建立的項目上雙擊滑鼠左鍵，之後畫面會出現對應所選語言的程式編輯視窗，而其中在 Transition 的編輯視窗中還會自動帶出 Transition 符號的部份。Action 與 Transition 的編輯方式與一般的 POU 相同，唯一不同的是在編輯視窗中沒有區域符號表，因此要宣告區域符號時，請於 SFC 圖形的編輯視窗中進行宣告，而同一個 SFC 程式的所有 Action 與 Transition 都將共用同一個區域符號表。



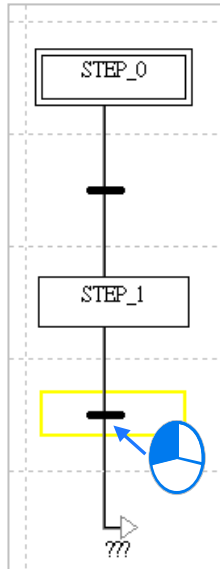
*. 若建立的 Transition 已指定給 SFC 圖面中的某個轉換點時，於該轉換點上雙擊滑鼠左鍵亦可開啟 Transition 的編輯視窗。



當在建立的 Action 或 Transition 項目上按下滑鼠右鍵後，於快捷選單中便有提供一些操作項可供使用，透過這些操作功能便可對所建立的 Action 或 Transition 項目進行管理。其中當使用複製/貼上功能時，須注意不可將 Actions 的項目複製至 Transitions 的項目下，當然也不可將 Transitions 的項目複製至的 Actions 項目下。



14.3.3 新增步進點結構












- (1) 先於編輯工作區中點選欲插入步進點結構的位置。

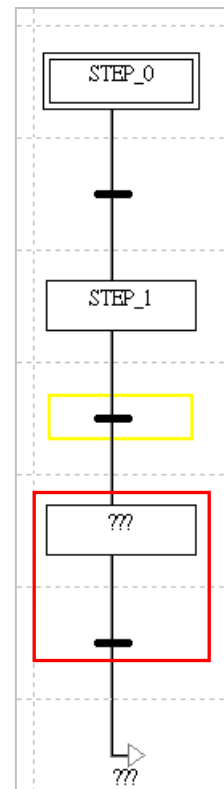


- (2) 點擊圖示工具列或右鍵快捷選單 > 編輯項目中的  圖示便可將新的步進點結構插於選取位置的上方，而按下  圖示則可將新結構插至選取位置的下方。



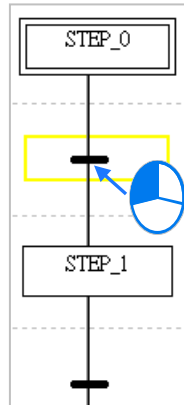
OR



	在上方插入步進點	Shift+F1
	在下方插入步進點	Shift+F2
	在右方插入轉換	Shift+F3
	在左方插入轉換	Shift+F4
	在右方插入步進點	Shift+F5
	在左方插入步進點	Shift+F6
	在右方插入步進點(共進分歧，選擇合流)	Shift+F7
	在左方插入步進點(共進分歧，選擇合流)	Shift+F8
	在右方插入轉換(選擇分歧，共進合流)	Shift+F9
	在左方插入轉換(選擇分歧，共進合流)	Shift+F10
	插入跳躍	Shift+F11



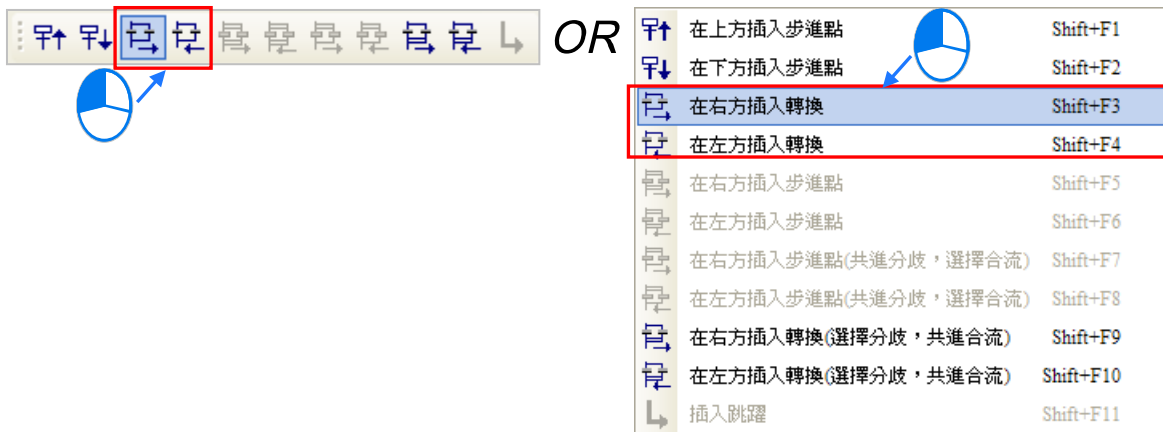
14.3.4 插入並聯轉換點












(1) 先於編輯工作區中點選欲並聯轉換點的位置。

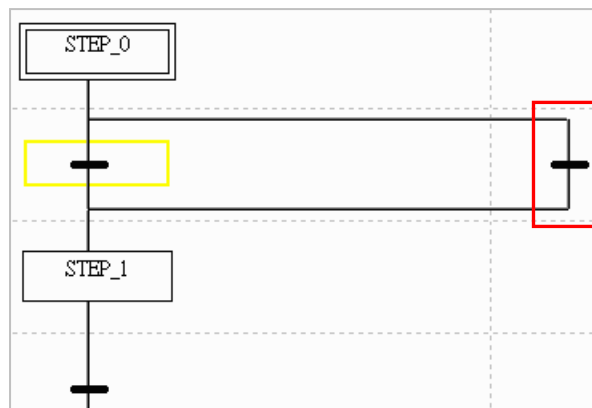


(2) 點擊圖示工具列或右鍵快捷選單 > 編輯項目中的  圖示便可將轉換點並聯至選取位置的右方，而按下  圖示則可將轉換點並聯至選取位置的左方。

14

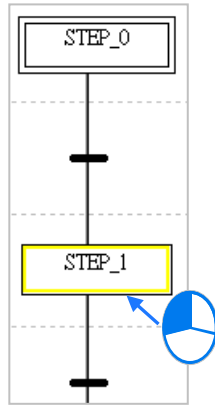




	在上方插入步進點	Shift+F1
	在下方插入步進點	Shift+F2
	在右方插入轉換	Shift+F3
	在左方插入轉換	Shift+F4
	在右方插入步進點	Shift+F5
	在左方插入步進點	Shift+F6
	在右方插入步進點(共進分歧，選擇合流)	Shift+F7
	在左方插入步進點(共進分歧，選擇合流)	Shift+F8
	在右方插入轉換(選擇分歧，共進合流)	Shift+F9
	在左方插入轉換(選擇分歧，共進合流)	Shift+F10
	插入跳躍	Shift+F11

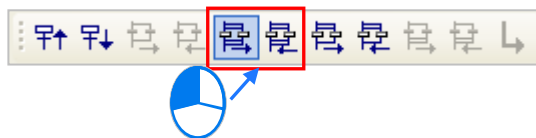


14.3.5 插入並聯步進點












(1) 先於編輯工作區中點選欲並聯步進點的位置 (圖面最上方的步進點無法與其他的步進點並聯)。



(2) 點擊工具列或右鍵快捷選單 > 編輯項目中的  圖示便可將步進點並聯至選取位置的右方，而按下圖示  則可將步進點並聯至選取位置的左方。



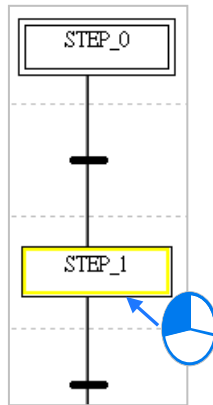
OR



	在上方插入步進點	Shift+F1
	在下方插入步進點	Shift+F2
	在右方插入轉換	Shift+F3
	在左方插入轉換	Shift+F4
	在右方插入步進點	Shift+F5
	在左方插入步進點	Shift+F6
	在右方插入步進點(共進分歧, 選擇合流)	Shift+F7
	在左方插入步進點(共進分歧, 選擇合流)	Shift+F8
	在右方插入轉換(選擇分歧, 共進合流)	Shift+F9
	在左方插入轉換(選擇分歧, 共進合流)	Shift+F10
	插入跳躍	Shift+F11















14.3.6 插入步進點結構 - 共進分歧&選擇合流

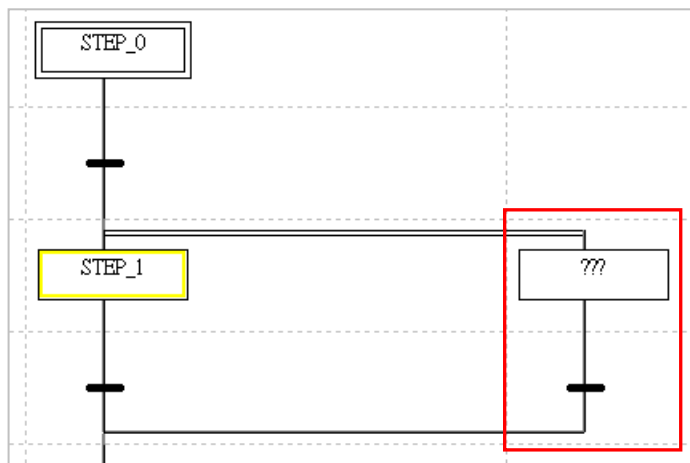
- (1) 先於編輯工作區中點選欲插入結構的步進點位置，而點選時必須點在步進點上，且不能是 SFC 圖面中最上方的步進點。



- (2) 點擊工具列或右鍵快捷選單 > 編輯項目中的  圖示便可將結構並聯至選取位置的右方，而按下  圖示則可將結構並聯至選取位置的左方。

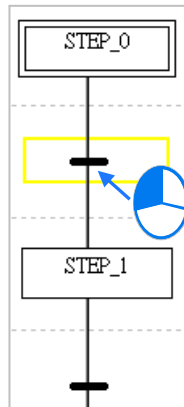
14



	OR		在上方插入步進點	Shift+F1
			在下方插入步進點	Shift+F2
			在右方插入轉換	Shift+F3
			在左方插入轉換	Shift+F4
			在右方插入步進點	Shift+F5
			在左方插入步進點	Shift+F6
			在右方插入步進點(共進分歧，選擇合流)	Shift+F7
			在左方插入步進點(共進分歧，選擇合流)	Shift+F8
			在右方插入轉換(選擇分歧，共進合流)	Shift+F9
			在左方插入轉換(選擇分歧，共進合流)	Shift+F10
			插入跳躍	Shift+F11














14.3.7 插入步進點結構 - 選擇分歧&共進合流

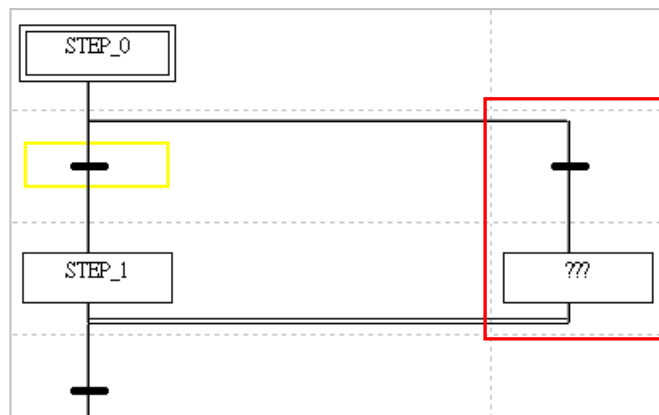
- (1) 先於編輯工作區中點選欲插入結構的轉換點位置，而點選時必須點在轉換點上，且不能是 SFC 圖面中最下方的轉換點。



- (2) 點擊工具列或右鍵快捷選單 > 編輯項目中的  圖示便可將結構並聯至選取位置的右方，而按下  圖示則可將結構並聯至選取位置的左方。

	在上方插入步進點	Shift+F1
	在下方插入步進點	Shift+F2
	在右方插入轉換	Shift+F3
	在左方插入轉換	Shift+F4
	在右方插入步進點	Shift+F5
	在左方插入步進點	Shift+F6
	在右方插入步進點(共進分歧, 共進合流)	Shift+F7
	在左方插入步進點(共進分歧, 共進合流)	Shift+F8
	在右方插入轉換(選擇分歧, 共進合流)	Shift+F9
	在左方插入轉換(選擇分歧, 共進合流)	Shift+F10
	插入跳躍	Shift+F11


14



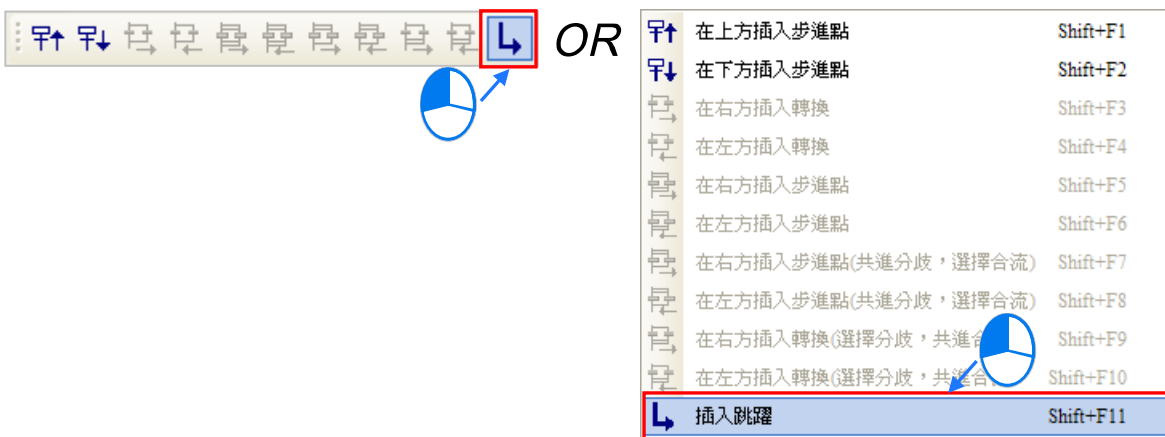
14.3.8 插入跳躍點

- (1) 先於編輯工作區中點選欲插入跳躍點的位置，而視合流的結構，點選時必須點在分歧路徑中最下方的步進點或轉換點。

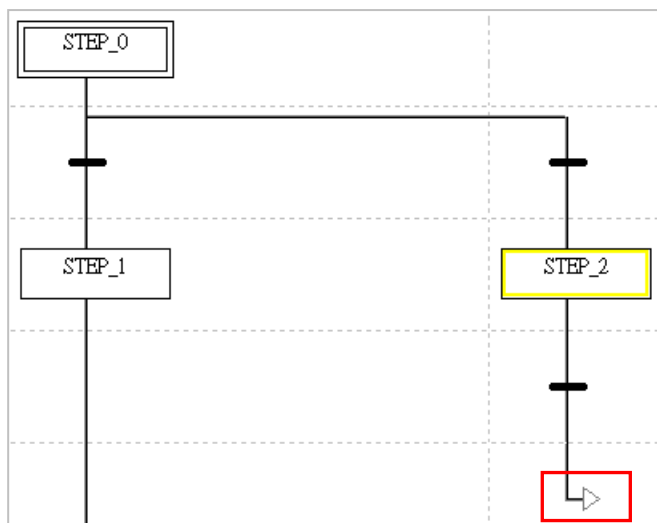


- (2) 點擊工具列或右鍵快捷選單 > 編輯項目中的  圖示即可插入跳躍點。

14

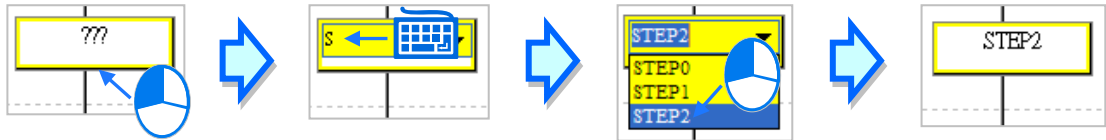


	在上方插入步進點	Shift+F1
	在下方插入步進點	Shift+F2
	在右方插入轉換	Shift+F3
	在左方插入轉換	Shift+F4
	在右方插入步進點	Shift+F5
	在左方插入步進點	Shift+F6
	在右方插入步進點(共進分歧，選擇合流)	Shift+F7
	在左方插入步進點(共進分歧，選擇合流)	Shift+F8
	在右方插入轉換(選擇分歧，共進合流)	Shift+F9
	在左方插入轉換(選擇分歧，共進合流)	Shift+F10
	插入跳躍	Shift+F11



14.3.9 指定步進點與轉換點符號

當完成圖形的部署後，接著便必須於圖形中逐一指定每個位置所對應的步進點或轉換點符號；編輯時請先點選欲編輯的步進點或轉換點，待黃色方框出現後，於該位置再點擊一次滑鼠左鍵便會出現該步進點或轉換點的編輯欄位，輸入部分文字後，便可從過濾出的下拉選單中選擇已建立好的項目即可。



除了利用滑鼠之外，使用者亦可直接使用鍵盤來進行操作。先利用鍵盤的上下鍵移動圖面中的黃色方框，在選定編輯位置後按下【Enter】鍵即會出現編輯欄位，此時即可直接使用鍵盤來輸入字串，便會出現過濾出的下拉選單，利用鍵盤的上下鍵便可進行選擇，完成編輯後，按下【Enter】鍵便會自動再切換至下一個編輯位置，當要結束編輯時，則只要按下【Esc】鍵即可。此外，當用鍵盤輸入的步進點名稱是尚未宣告的變數符號時，按下【Enter】鍵後，系統預設將會自動導出符號宣告視窗，但若輸入的位置是轉換點時，畫面便不會出現符號宣告視窗。



轉換點可以指定為結構變數符號中的 BOOL 型態元素，關於結構的說明請參閱第 8 章。

*自動導出符號宣告視窗的功能可由使用者自行設定是否開啟，設定方式則請參考第 2.3.1 節。

14.3.10 指定初始步進點

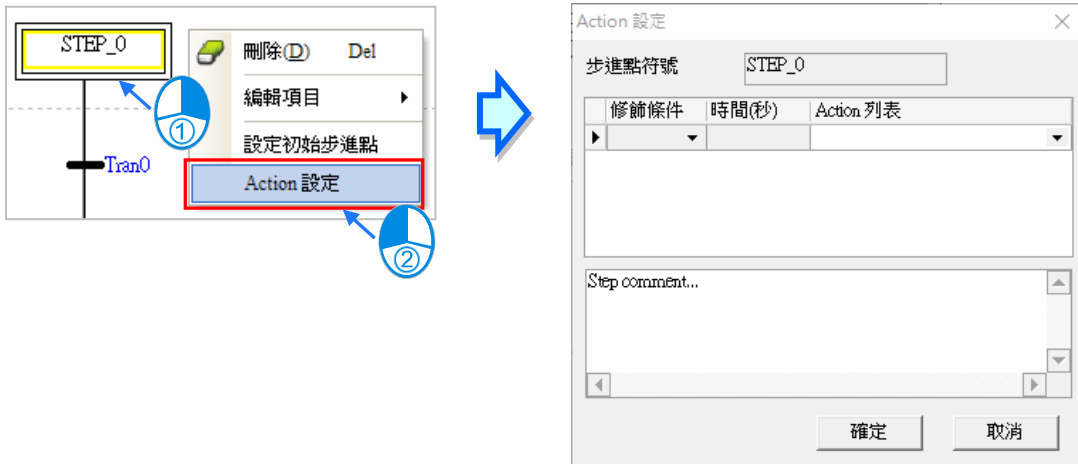
此功能僅限 AH/AS 機種，DVP 機種無法於圖面中指定初始步進點。設定時請先於編輯工作區中點選要設為初始步進點的位置，之後按下滑鼠右鍵，並於快捷選單中點選「設定初始步進點」即可。



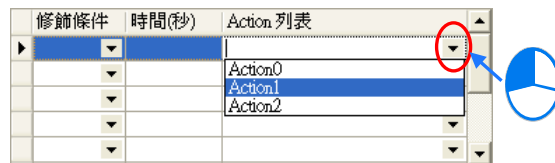
*關於 DVP 機種的步進點啟動請參考本章第 14.2.7 節的相關說明。

14.3.11 配置 Actions/摺疊 Action 列表

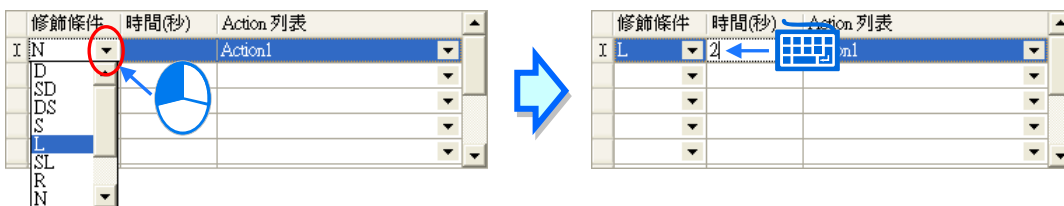
- (1) 首先點選欲配置 Action 的步進點後按下滑鼠右鍵，接著於快捷選單中點選「**Action 設定**」項目便可開啟設定視窗；但若該步進點的變數符號尚未宣告時，系統將不允許進行 Action 的配置。



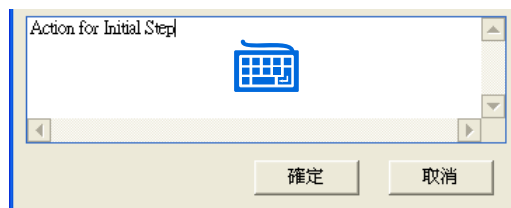
- (2) 開啟設定視窗後，請先於「**Action 列表**」中下拉選擇欲配置的 Action 項目。



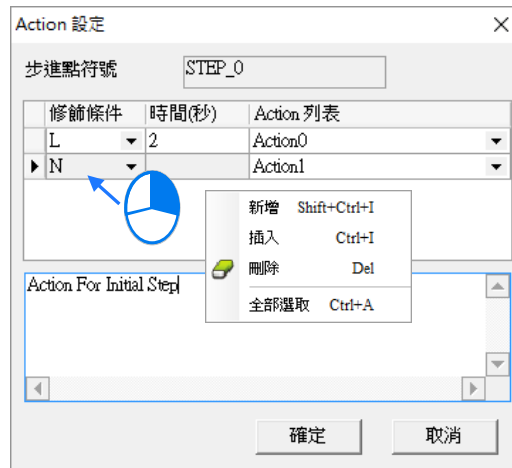
- (3) 選擇好 Action 項目後，請再於「**修飾條件**」欄位下拉選擇對應的條件，且當設定的條件與時間相關時，必須再於「**時間 (秒)**」欄位中設定時間參數，而輸入時必須以秒為單位，且只可輸入整數（DVP 機種無法設定修飾條件）。



- (4) 若要輸入註解時，請直接於最下方的空白欄位中進行編輯即可。

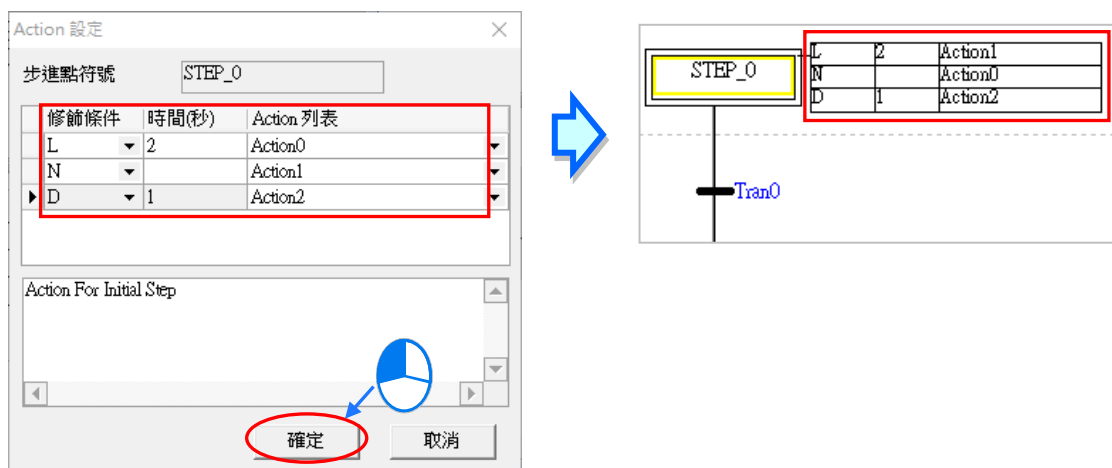


- (5) 當於列表中點選某一項目後按下滑鼠右鍵，之後便可透過快捷選單的操作項進行各種編輯；至於各操作項的功能則請參考下表。當要於 Action 設定視窗中同時選取多個行列時，可先按住鍵盤的【Ctrl】鍵，接著再用滑鼠分別點選不連續位置的行列；另外也可先按住【Shift】鍵後，再先後用滑鼠點選欲選取範圍的第一個與最後一個行列，之後便可將整個範圍內的行列全部選取起來。

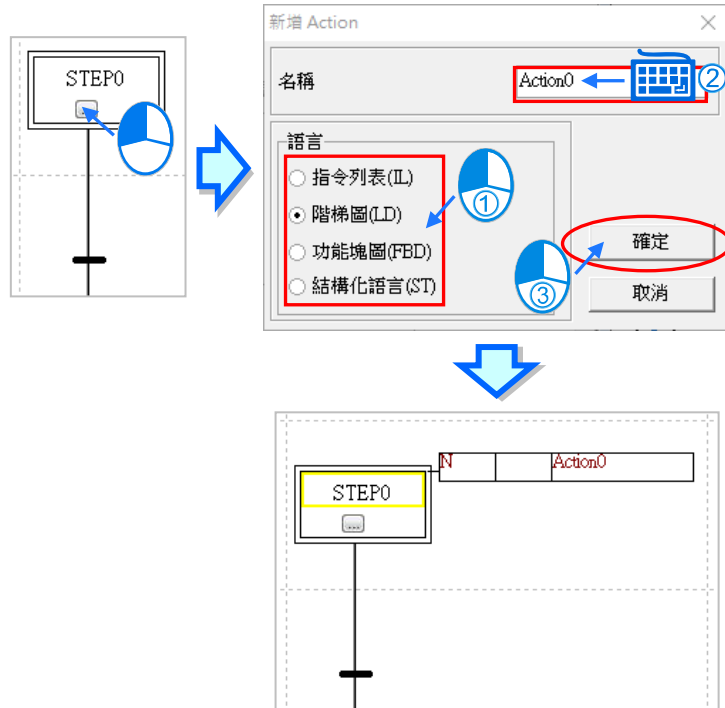


操作項	功能說明
新增	於所選項目的下方新增一新行列。
插入	於所選項目的上方插入一新行列。
刪除	刪除目前所選的項目。
全部選取	同時將所有項目選取起來。

- (6) 完成所有的設定後按下「確定」按鈕即可，而此時於該步進點的右側便會出現配置的列表。

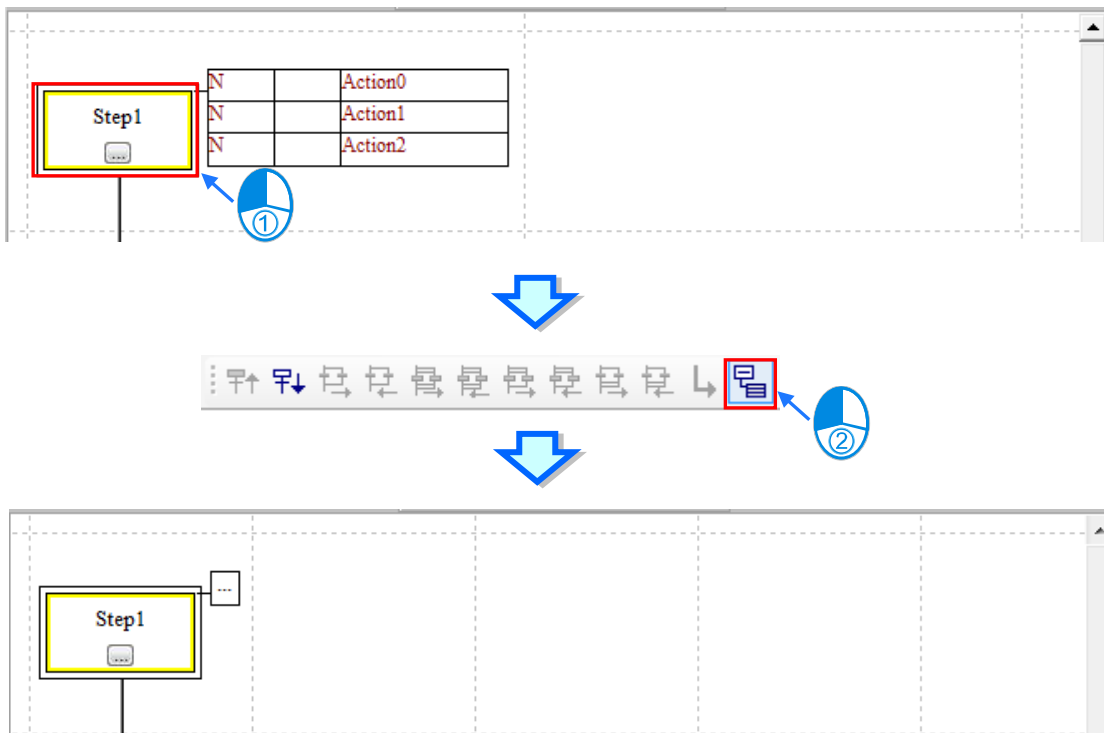


為便於使用者快速的新增與配置 Action，ISPSOft 在 STEP 上提供了一個快捷按鈕，點選此按鈕後便會出現新增 Action 視窗，輸入完成後按下「確定」按鈕，便會同時建立 Action 與配置此 Action 到 STEP 中。



14

若使用者覺得配置的 Action 會占用畫面空間，可點擊「折疊 Action 列表」鈕，來做折疊縮放的動作。



14.4 SFC 的程式範例

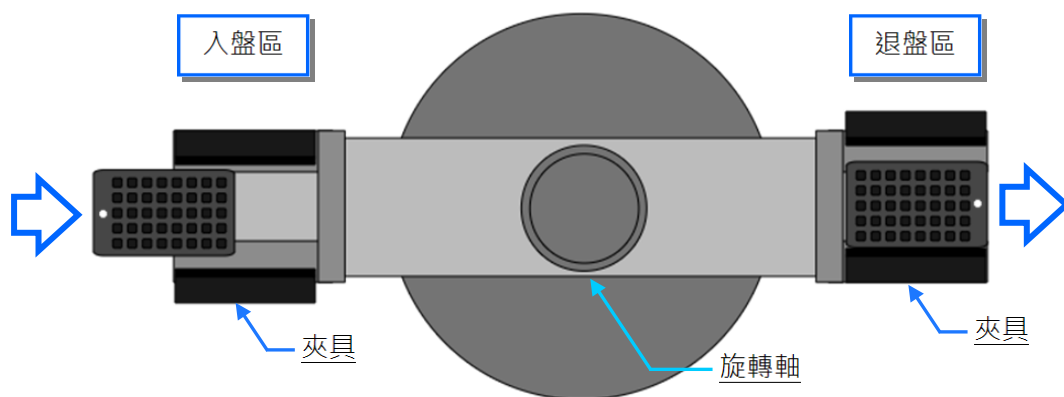
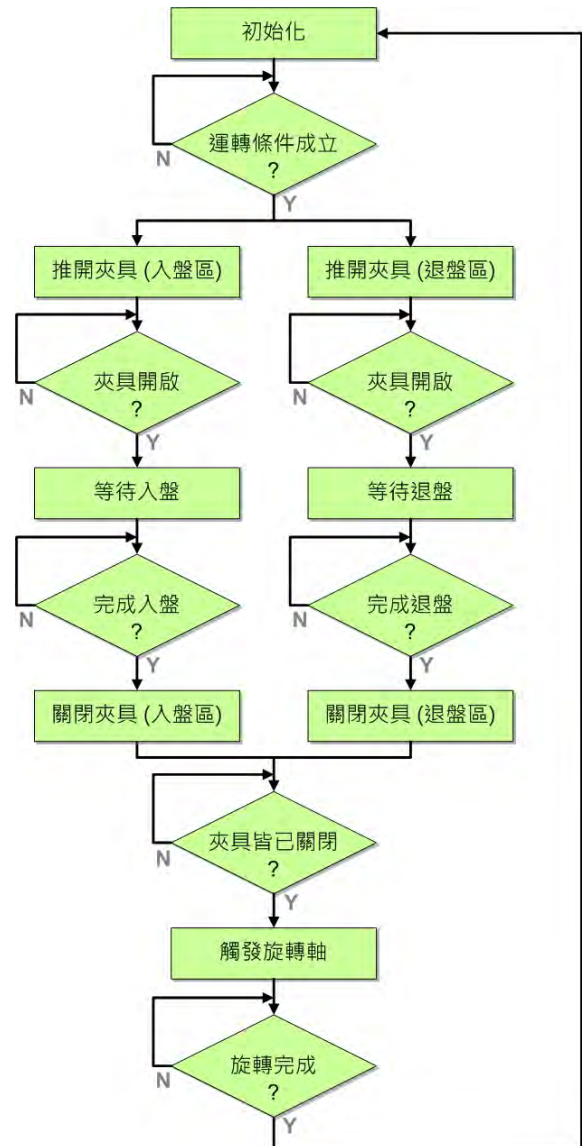
14.4.1 範例說明

此範例主要為兩個工作站之間的載盤轉送功能，右側為其工作流程圖，而下方則為實際設備的示意圖。

如下圖，平台的左側為入盤區，而右側則為退盤區，當入盤與退盤的工作完成後，連結平台中心的旋轉軸便會旋轉 180 度，之後於入盤區接收的載盤便可轉送至退盤區，且完成退盤的空夾具亦可再次回到入盤區進行下次的入盤動作。而在整個過程當中，除了旋轉軸的動作之外，其餘期間的入盤與退盤動作皆是彼此獨立的。

當系統開始運轉時，位於夾具下方的氣缸會先把兩側的夾具推開，當夾具開啟後，程式便可將入盤與退盤的 Ready 信號設置為 ON，並等待與此系統銜接的前後工作站各自完成其入盤與退盤的動作。

當完成入盤或退盤的動作後，系統便會收到回送的完成信號，此時便可將其對應的 Ready 信號設為 OFF，並將推開夾具的氣缸退回以讓夾具關閉；而當系統確認兩側的夾具皆已關閉後，系統便會送出旋轉軸的觸發信號，並在旋轉完成且收到回送的完成信號後開始繼續下一個循環。



14.4.2 硬體規劃

在本範例中，我們將選用 AH5x0 系列主機 **AHCPU530-EN**，數位 IO 模組則選用 8 點輸入/8 點輸出的 **AH16AP11R-5A**，主背板則選用 4 槽的 **AHBP04M1-5A**。下表即為本範例的 IO 配置表。

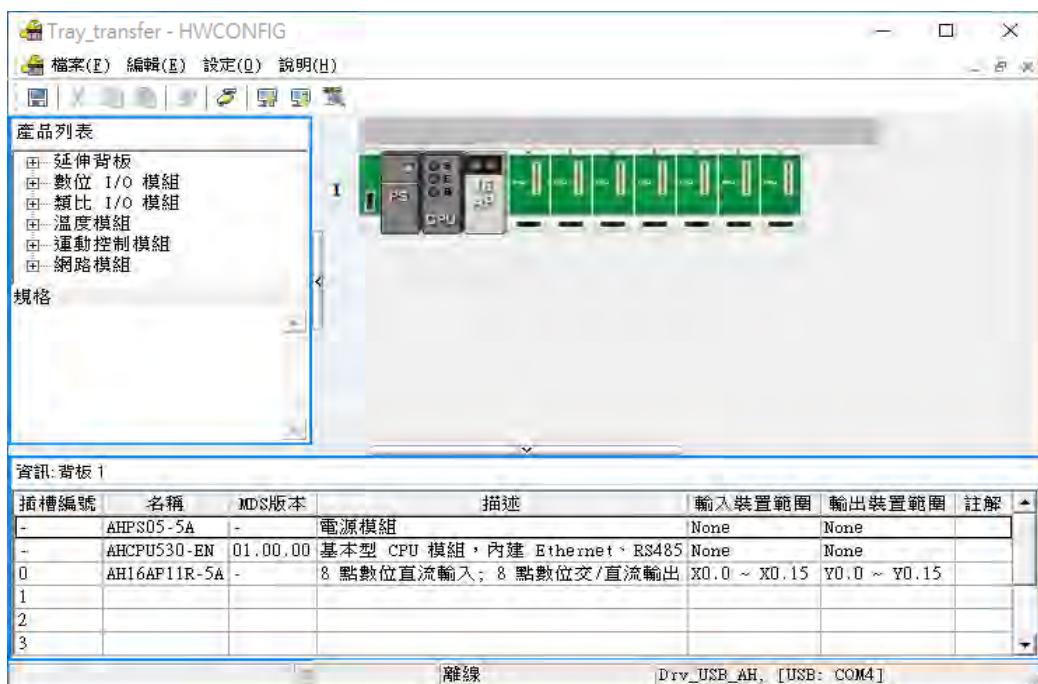
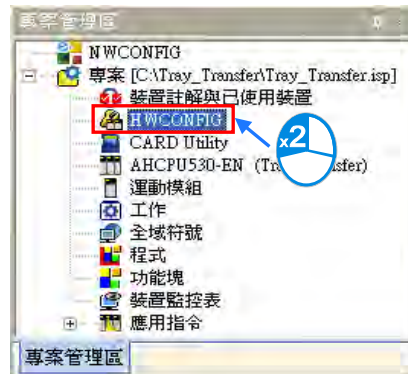
Type	ID	說明
DI	X0.0	入盤完成信號
DI	X0.1	退盤完成信號
DI	X0.2	旋轉完成信號
DI	X0.3	(入盤區) 夾具氣缸 - 開啟 Sensor
DI	X0.4	(入盤區) 夾具氣缸 - 關閉 Sensor
DI	X0.5	(退盤區) 夾具氣缸 - 開啟 Sensor
DI	X0.6	(退盤區) 夾具氣缸 - 關閉 Sensor
DO	Y0.0	入盤 Ready 信號
DO	Y0.1	退盤 Ready 信號
DO	Y0.2	旋轉軸觸發信號
DO	Y0.3	(入盤區) 夾具氣缸
DO	Y0.4	(退盤區) 夾具氣缸

14.4.3 程式規劃

- (1) 系統的**初始程序**會先將所有的觸發與 Ready 信號重置為 OFF，並等待系統的運轉條件成立；而在本範例的規劃中，當**運轉旗標**為 ON 且**錯誤旗標**為 OFF 時，運轉條件便會成立。
- (2) 一旦確定為運轉狀態後，系統會先將兩側的**夾具 (Y0.3 & Y0.4)**推開，並確認兩個夾具的**開啟 Sensor (X0.3 & X0.5)**為 ON。
- (3) 確認夾具已經開啟後，系統會先延遲 1 秒，待狀態穩定後便會分別送出入盤與退盤的 **Ready 信號 (Y0.0 & Y0.1)**，以通知與系統銜接的前後工作站。
- (4) 當入盤側的動作完成後，系統便會收到**入盤完成信號 (X0.0)**，此時系統會先延遲 1 秒，待狀態穩定後再將入盤區的**夾具 (Y0.3)**關閉。
- (5) 當退盤側的動作完成後，系統同樣會收到**退盤完成信號 (X0.1)**，此時系統亦會先延遲 1 秒，待狀態穩定後再將退盤區的**夾具 (Y0.4)**關閉。
- (6) 當確認兩側夾具的**關閉 Sensor (X0.4 & X0.6)**已全部為 ON 後，系統便會發出旋轉軸的觸發信號 (**Y0.2**) 以開始轉送載盤；待旋轉完畢後，系統便會收到旋轉完成信號 (**X0.2**)，接著步序便會重新回到**初始程序**，而此時若運轉條件仍然成立，則程序便會繼續進行下一次的循環。

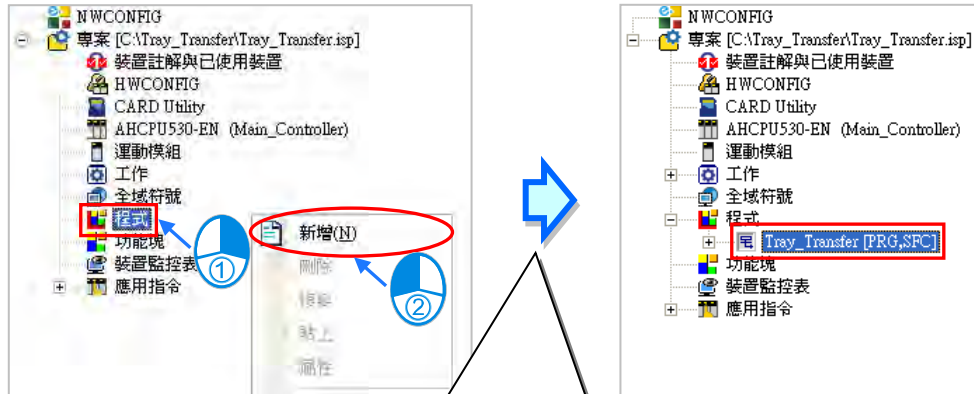
14.4.4 建立範例程式

首先請開啟 ISPSOft 並建立一新專案，本範例的硬體規劃與第 4 章相同，因此請參考第 4 章的內容，並自行完成硬體規劃的工作。

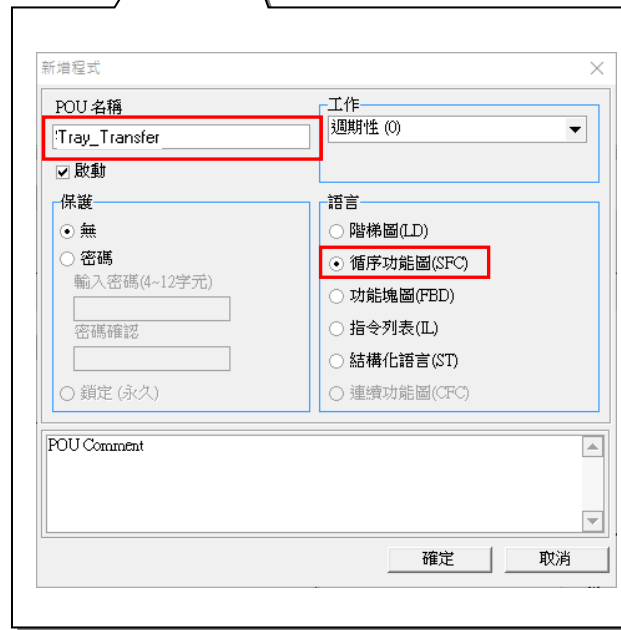


14

接下來請直接於專案管理區中新增一 SFC 的程式 POU。



14



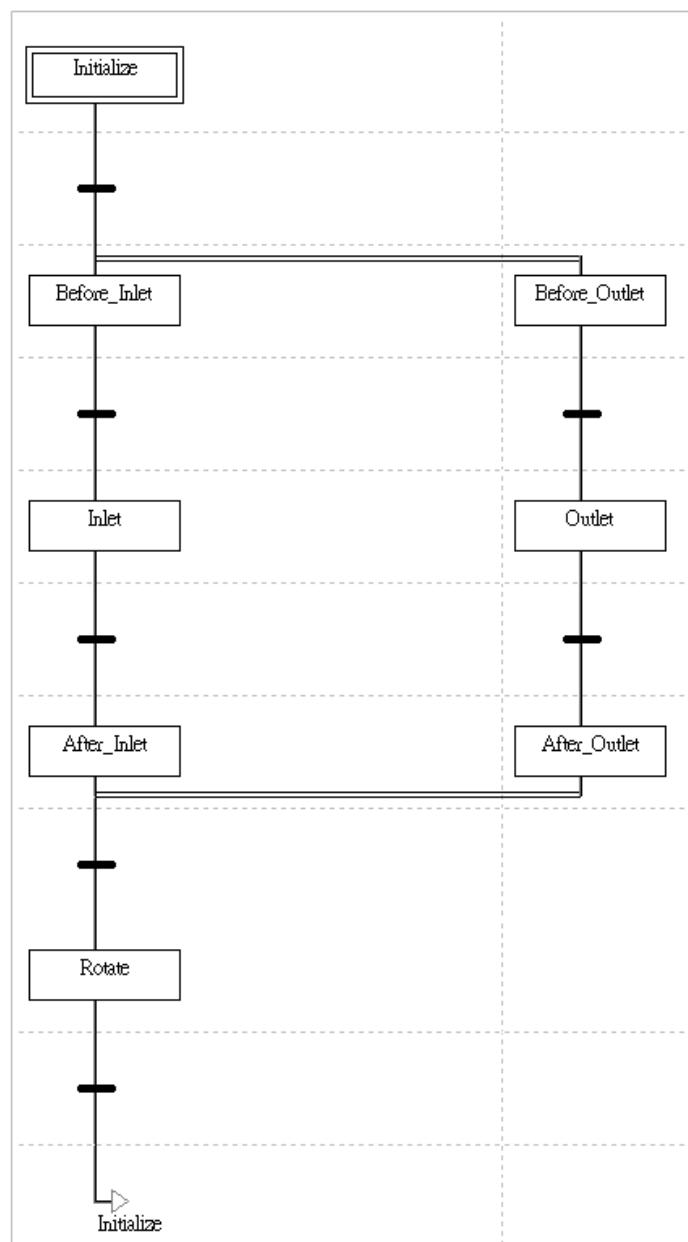
再來請將已知的系統旗標及 IO 配置建立在全域符號表中。

全域符號					
類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	IN_Done	X0.0	BOOL	FALSE	Signal for Inlet finish
VAR	OUT_Done	X0.1	BOOL	FALSE	Signal for Outlet finish
VAR	R_Done	X0.2	BOOL	FALSE	Signal for Rotation finish
VAR	Clip_1_OS	X0.3	BOOL	FALSE	Sensor for Inlet Clip open
VAR	Clip_1_CS	X0.4	BOOL	FALSE	Sensor for Inlet Clip close
VAR	Clip_2_OS	X0.5	BOOL	FALSE	Sensor for Outlet Clip open
VAR	Clip_2_CS	X0.6	BOOL	FALSE	Sensor for Outlet Clip close
VAR	Inlet_Rdy	Y0.0	BOOL	FALSE	Status signal for Inlet workstation
VAR	Outlet_Rdy	Y0.1	BOOL	FALSE	Status signal for Outlet workstation
VAR	Rotator_Trig	Y0.2	BOOL	FALSE	Trig signal for Rotator
VAR	Clip_1	Y0.3	BOOL	FALSE	DO Signal for Inlet Clip
VAR	Clip_2	Y0.4	BOOL	FALSE	DO Signal for Outlet Clip
VAR	RUN	N/A [Auto]	BOOL	FALSE	Flag of System running
VAR	ERROR	N/A [Auto]	BOOL	FALSE	Flag of Error

接著請在 SFC 程式的區域符號表中宣告各個步進點的變數符號。

區域符號					
類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	Initialize	N/A [Auto]	STEP	FALSE	
VAR	Before_Inlet	N/A [Auto]	STEP	FALSE	
VAR	After_Inlet	N/A [Auto]	STEP	FALSE	
VAR	Before_Outlet	N/A [Auto]	STEP	FALSE	
VAR	After_Outlet	N/A [Auto]	STEP	FALSE	
VAR	Inlet	N/A [Auto]	STEP	FALSE	
VAR	Outlet	N/A [Auto]	STEP	FALSE	
VAR	Rotate	N/A [Auto]	STEP	FALSE	

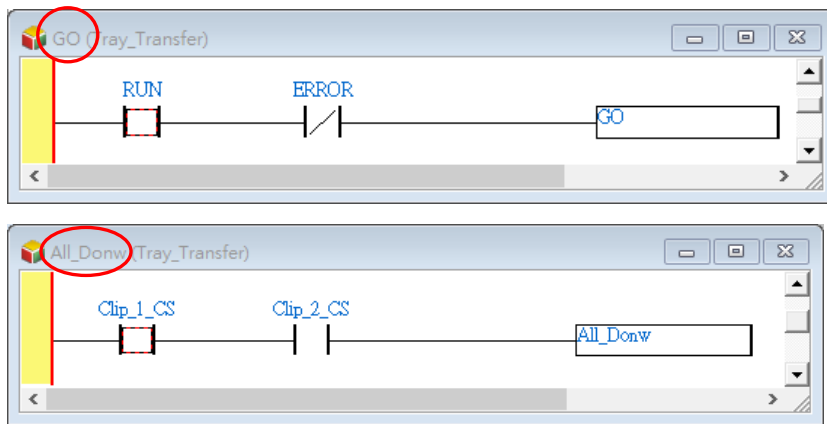
接著參照下圖建立 SFC 流程，而**初始步進點**的部份則請直接延用預設的第一個步進點即可。



再來請至專案管理區中，於「Transitions」項目下建立「GO」與「All_Done」兩個 Transition。

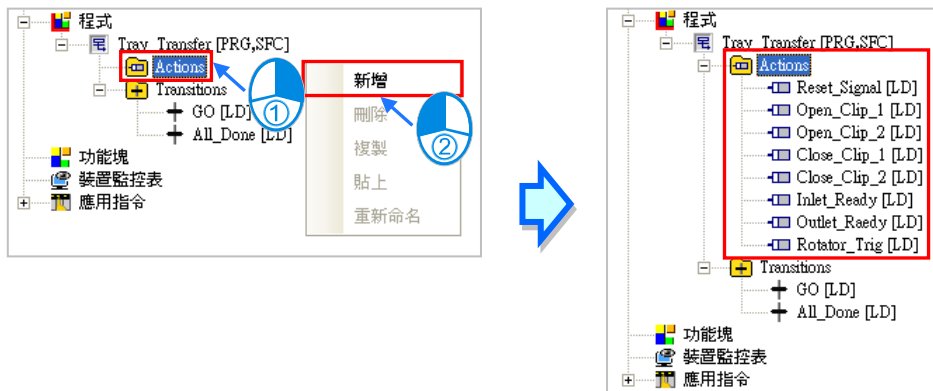


完成 Transition 項目的建立後，請逐一完成如下的 Transition 程式內容。

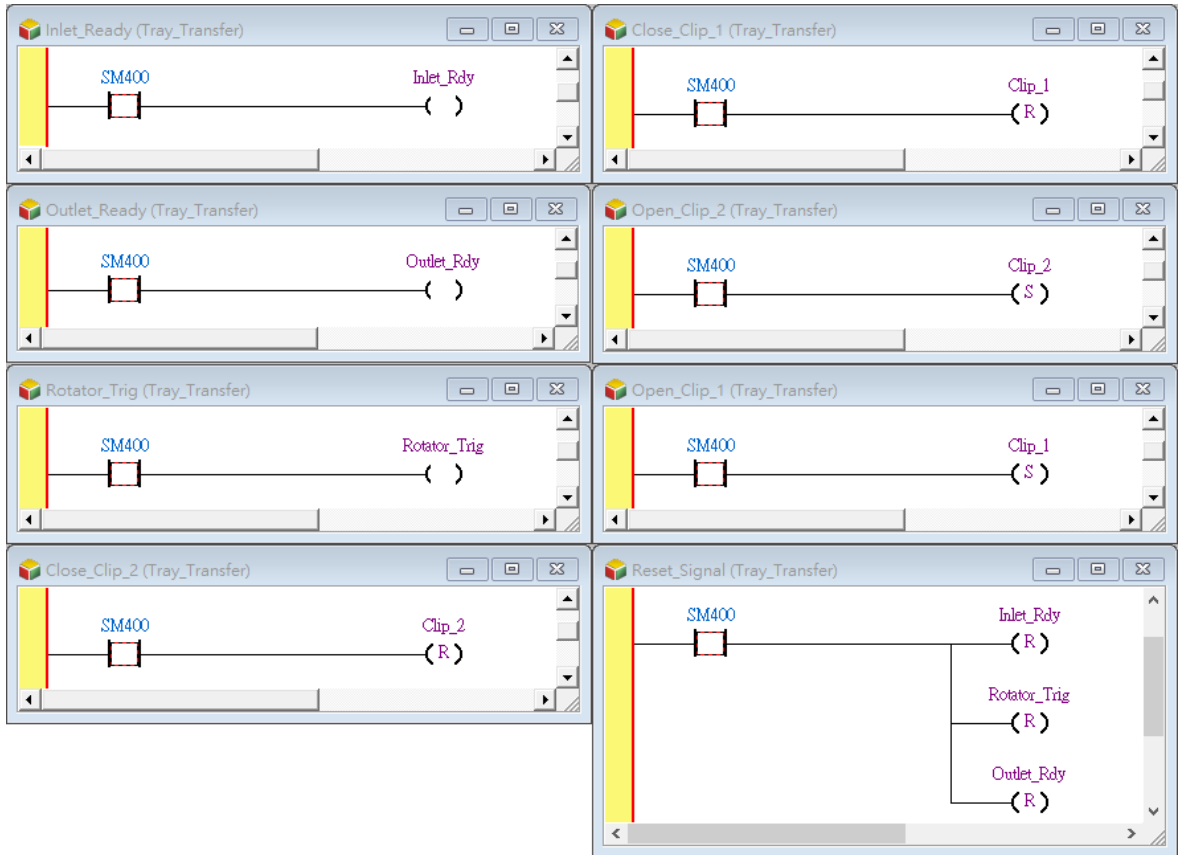


14

接著請再於 POU 的「Actions」項目下建立各個程序的 Action。



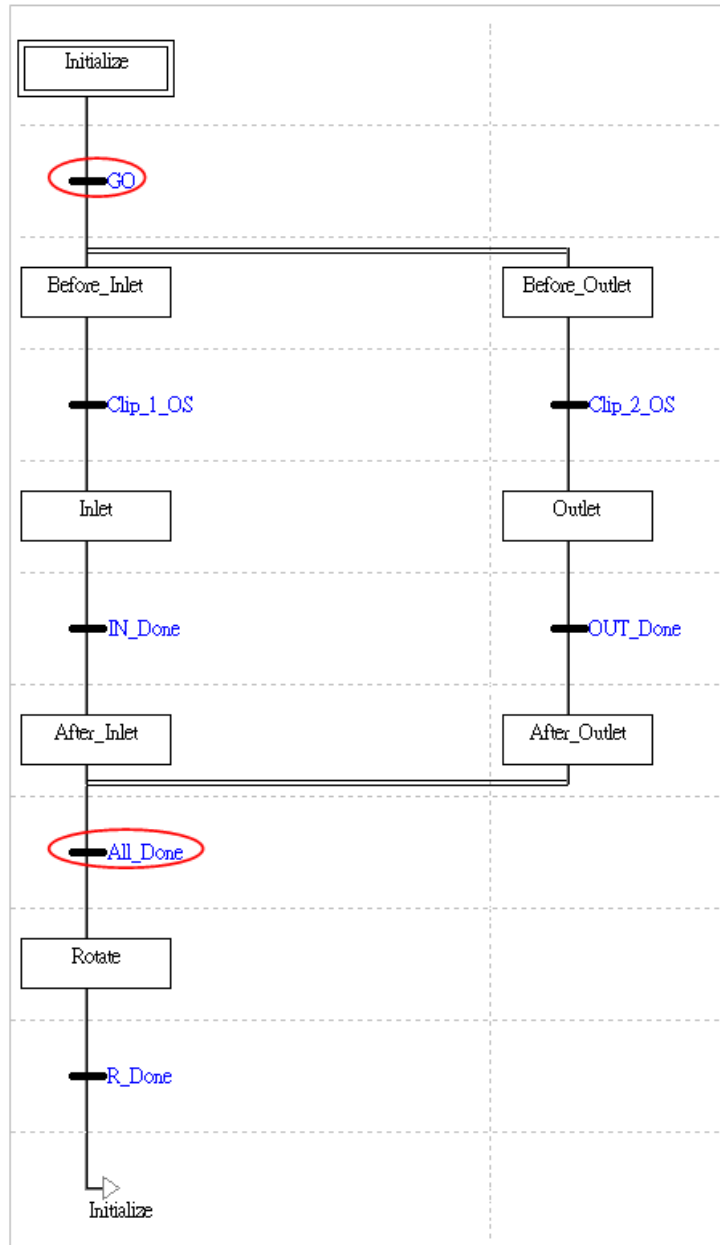
完成 **Action** 項目的建立後，請再逐一完成如下的 Action 程式內容。



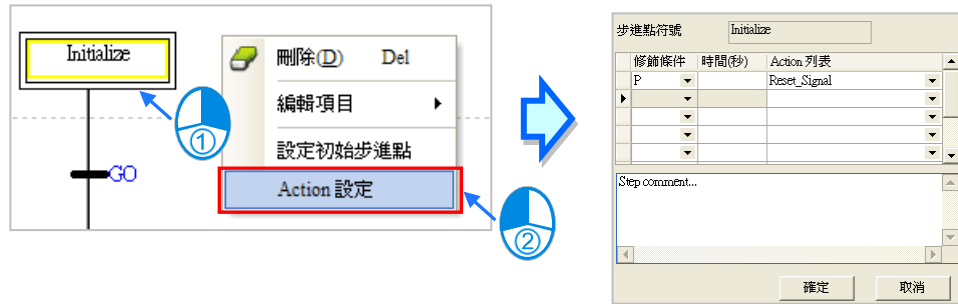
14

完成所有 Action 與 Transition 程式建立後，請於 SFC 的圖形中逐一指定各個轉換點，而其中除了「GO」與「All_Done」兩個 Transition 之外，其餘皆以之前建立的全域符號做為轉換點的條件。其完成圖形如下所示：

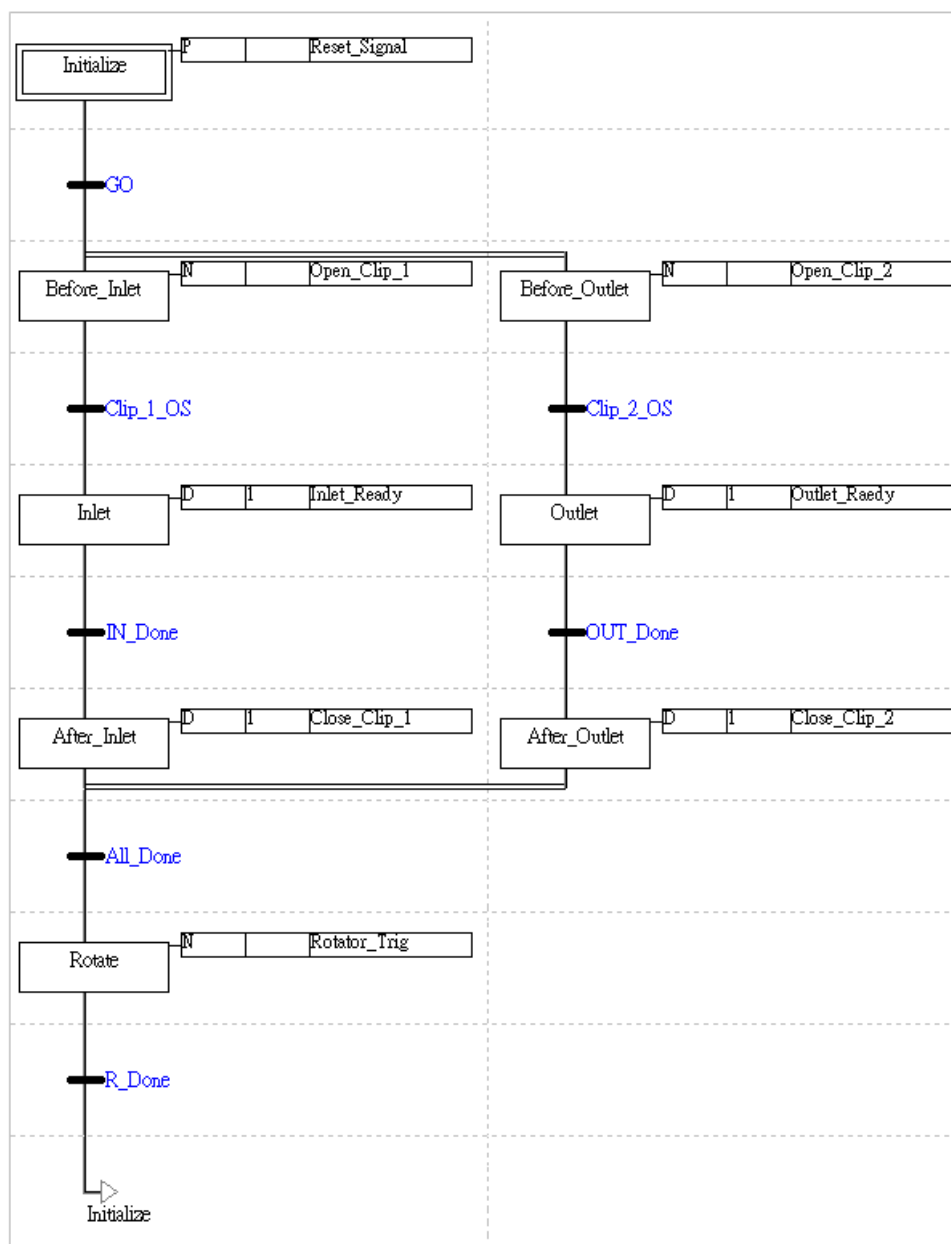
14



接下來請逐一於各個步進點中配置其對應的 Action，並指定各個 Action 的修飾條件。



最後完成的程式如下，接著請自行完成編譯，並將硬體參數與程式下載至主機當中進行後續的測試。



MEMO

14

第15章 連續功能圖 CFC

目錄

15.1 連續功能圖 (CFC) 簡介	15-2
15.1.1 關於連續功能圖	15-2
15.1.2 CFC 編程的注意事項	15-2
15.1.3 ISPSOft 的 CFC 編輯環境	15-3
15.1.4 CFC 的圖示工具列	15-3
15.1.5 CFC 編輯工作區的快捷選單	15-4
15.2 於 ISPSOft 中建立 CFC 程式	15-5
15.2.1 選取物件	15-5
15.2.2 輸入節點、輸出節點與邏輯閘	15-6
15.2.2.1 插入節點或閘	15-7
15.2.2.2 新增與刪除接腳	15-9
15.2.3 改變接腳的形式	15-10
15.2.4 物件的連線與取消	15-11
15.2.5 指令與功能塊	15-11
15.2.5.1 插入指令	15-13
15.2.5.2 插入功能塊	15-15
15.2.6 刪除物件	15-17
15.2.7 編輯裝置或符號	15-18
15.2.8 物件的致能與失效	15-19
15.2.9 插入註解	15-19
15.2.10 變更程式的執行順序	15-22
15.2.11 顯示裝置註解	15-22

15.1 連續功能圖 (CFC) 簡介

15.1.1 關於連續功能圖

連續功能圖 (Continuous Function Chart, CFC) 為基於 IEC61131-3 所進一步延伸發展出來的 PLC 編程語言，連續功能圖經常被使用在運動控制領域中，其最大的特色就是以類似邏輯設計圖來表達程序的控制，相較於其他的 PLC 編程語言，連續功能圖提供更明確的輸入與輸出的邏輯關係，及程式的執行順序。

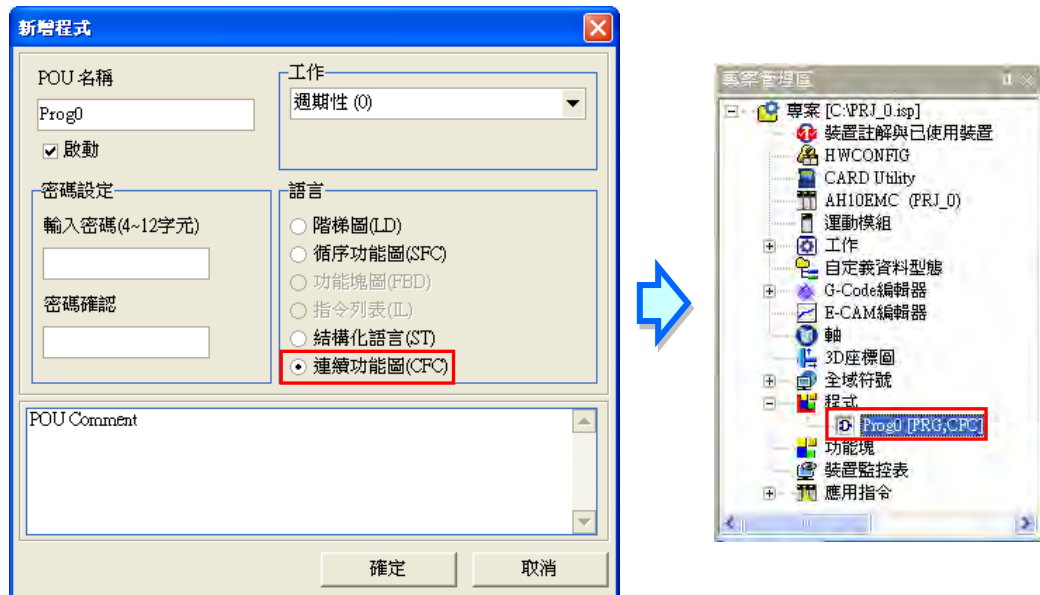
本章後續僅針對 ISPSOft 中的連續功能圖操作與建立方式加以介紹，至於連續功能圖的動作原理在此將不再多加敘述。

15.1.2 CFC 編程的注意事項

- 支援鍵盤輸入，但不支援 LD、LDI、LDP、LDF 指令，可以直接輸入暫存器或變數的名稱，自動貼上一個輸入節點。
 - 鍵盤輸入時，英文無大小寫的分別，亦即 OUT、Out 與 out 的意義是相同的。
 - 常數的進制表示如下。
 - 10 進制：2345
 - 16 進制：16#5BA0
 - 浮點數：4.123
- *. 仍支援 K、H 表示法。
- CFC 語言可編輯的迴路數並沒有限制，但仍需考慮編譯後的程式大小是否會超出主機的記憶體容量。
 - 節點、邏輯閘、指令或功能塊輸入接腳只能來自一個來源，但輸出接腳的目標可到多個。

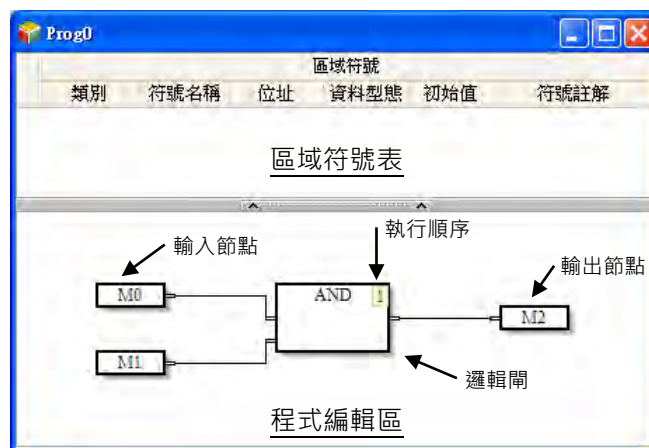
15.1.3 ISPSOft 的 CFC 編輯環境

在建立 POU 時，於語言欄位選擇「連續功能圖 (CFC)」便可建立連續功能圖程式，請參考第 5.4.1 節的說明。



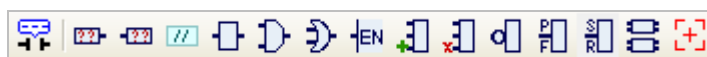
下圖為 ISPSOft 中的 CFC 編輯工作區環境。其視窗的上方配置為區域符號表，用於宣告區域符號，而下方則是程式編輯區，用於編寫 CFC 程式。連續功能圖程式的物件包括了節點與邏輯閘等物件，物件右上角的編號代表程式的執行順序。

15



15.1.4 CFC 的圖示工具列

進入 CFC 的編輯環境後，ISPSOft 的視窗中便會出現對應的編程圖示工具列，如下圖，其功能簡介如下表。



圖示	鍵盤 (快捷鍵)	功能說明
	無	顯示裝置註解
	Shift + F9	輸入節點
	Shift + F3	輸出節點
	Shift + F10	註解
	Shift + F11	OUT 閘
	Shift + F1	AND 閘
	Shift + F2	OR 閘
	Shift + F12	開啟/關閉功能塊或指令的 EN 接腳
	Shift + F4	新增輸入接腳
	Shift + F5	刪除輸入接腳
	Shift + F6	反相
	Shift + F8	正/負緣
	Shift + F7	設定/重置
	無	自動排列，將所有物件依據資料流排序執行順序，再由上至下，由左至右依執行順序排列。
	無	自動縮放，會將畫面調整至看到所有物件的大小

15

15.1.5 CFC 編輯工作區的快捷選單

在程式編輯區中按下滑鼠右鍵便會出現快捷選單如下圖，相關功能如下表。

	復原(U)	Ctrl+Z
	取消復原(R)	Ctrl+Y
	剪下(D)	Ctrl+X
	複製(C)	Ctrl+C
	貼上(P)	Ctrl+V
	刪除(D)	Del
	全選(A)	Ctrl+A
	註解(M)	Shift+F10
	執行順序(X)	
	致能/失效(I)	
	En	Shift+F12
	反向(N)	Shift+F6
	正/負緣(L)	Shift+F8
	設定/重置(S)	Shift+F7

操作項目	功能說明
復原	復原剛才的編輯動作
取消復原	取消復原而重複剛才的動作
剪下	剪下選取的物件。
複製	複製選取的物件。
貼上	將方才複製或剪下的物件貼在目前的編輯位置。
刪除	刪除選取的物件。
全選	全選程式編輯區所有物件。
註解	在滑鼠游標位置插入註解
執行順序	設定邏輯閘指令或功能塊的執行順序；設為資料流，會依照物件之間的先後關係重新安排執行順序；設為拓樸，會依照物件擺放的相對位置，由上至下，由左至右安排執行順序。
致能/失效	切換物件是否被執行
EN	開啟/關閉功能塊或指令的 EN 接腳
反向	切換接腳是否為反相
正/負緣	切換接腳的正/負緣狀態
設定/重置	切換接腳的設定/重置狀態

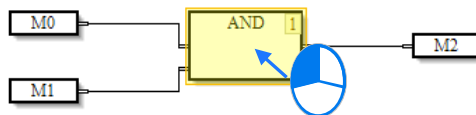
15.2 於 ISPSoft 中建立 CFC 程式

15.2.1 選取物件

CFC 語言的物件只要使用滑鼠便可進行選取，選取後的物件便可進行如剪下、複製等各種編輯動作。

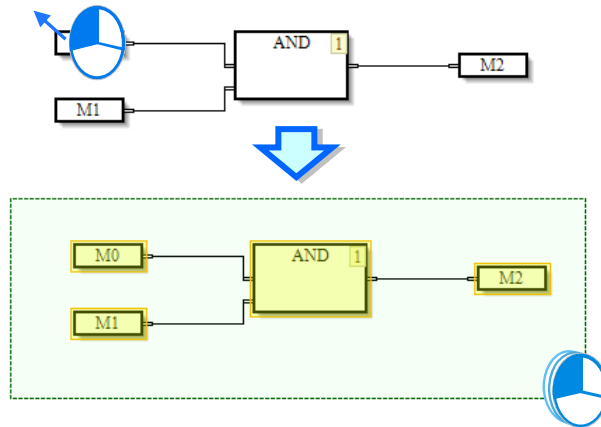
- 選取物件

將滑鼠移至欲選取的物件上方後，點擊滑鼠左鍵即可。被選取物件的物件將會以橘色表示，未選取的物件則以白色表示。如下圖。也可使用鍵盤的方向鍵切換選取的物件。



- 多重選取

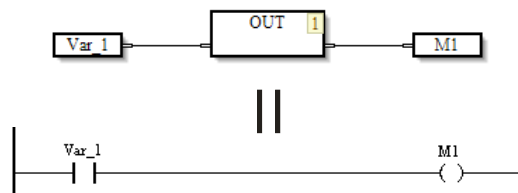
在空白處按住滑鼠左鍵後，拖曳欲選取的範圍，範圍內的物件都將被選取，如下圖。



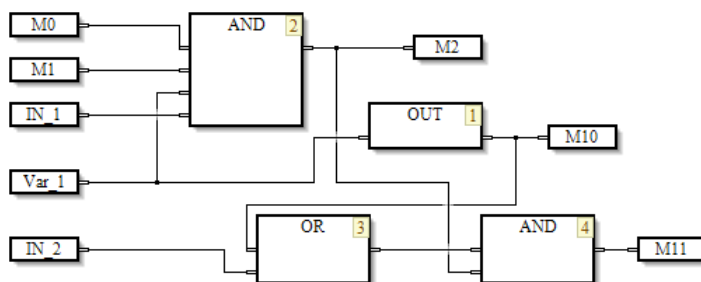
15.2.2 輸入節點、輸出節點與邏輯閘

CFC 圖的原理是藉由輸入節點輸入來源的變數或暫存器的狀態，藉由邏輯閘的運算後將狀態輸出到輸出節點。但輸入與輸出節點不可直接相連，故若需要將狀態進行搬移，必須使用 OUT 閘，如下圖，為變數 Var_1 的狀態搬移到暫存器 M1，及對應階梯圖相同意義的程式。

15



其他的邏輯閘包括了 AND 閘與 OR 閘，藉由連接這些物件，便可設計各種複雜的迴路，如下圖。

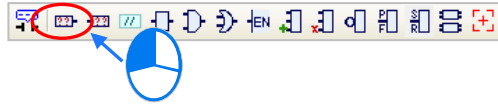


15.2.2.1 插入節點或閘

插入節點可用下列二種方式，請依使用習慣擇一即可。

- 方法一：圖示工具列

(1) 點擊圖示工具列上要插入的按鈕。如下圖。

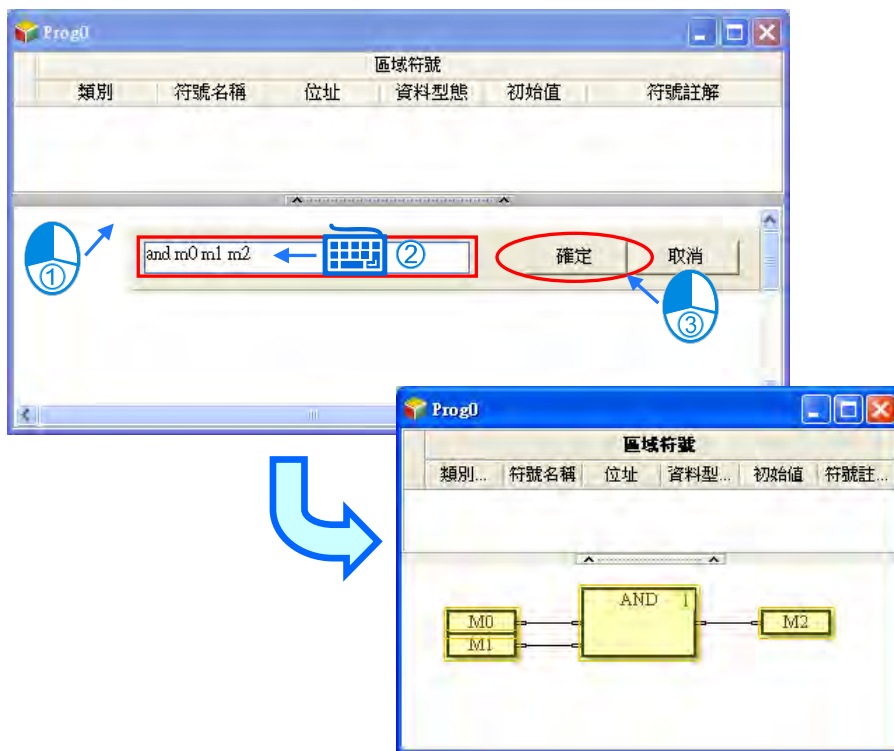


(2) 在程式編輯區要放置的位置點擊滑鼠左鍵，便完成物件的插入。如下圖。



- 方法二：鍵盤輸入

(1) 先用滑鼠左鍵點選程式編輯區空白處，再直接用鍵盤輸入指令後，畫面便會自動出現指令編輯視窗，待輸入完畢後按下鍵盤的「Enter」鍵，或以滑鼠點擊視窗的「確定」鈕，此時移動滑鼠游標，在程式編輯區需要插入指令的地方按下滑鼠左鍵，便在此處完成指令的建立（所輸入的指令無須分大小寫，且若輸入不正確的應用指令或對應裝置，按確定鍵時會有錯誤訊息）。如下圖。



15

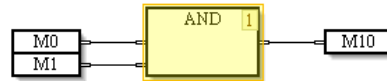
鍵盤輸入支援的指令如下表。


鍵盤輸入指令	功能說明
OUT	產生一個 OUT 閘
AND	產生一個 AND 閘
OR	產生一個 OR 閘
暫存器名稱	產生一個此暫存器的輸入節點
指令名稱	產生一個此指令的方塊
功能塊名稱	產生一個此功能塊的方塊

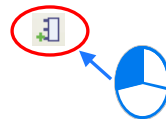
15.2.2.2 新增與刪除接腳

AND 與 OR 閘可使用以下步驟插入及刪除輸入接腳，便於多輸入條件時的設計。

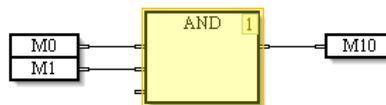
(1) 滑鼠點選邏輯閘，使之成為被選取的狀態。如下圖。



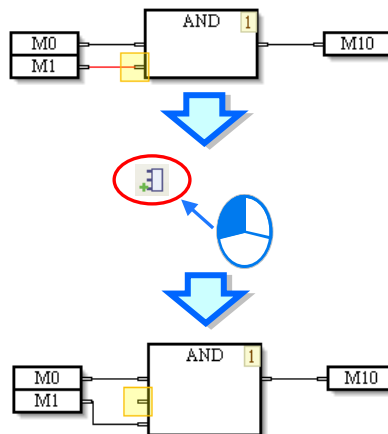
(2) 此時圖示工具列的「新增輸入接腳」按鈕便會亮起，使用滑鼠左鍵點擊。如下圖。





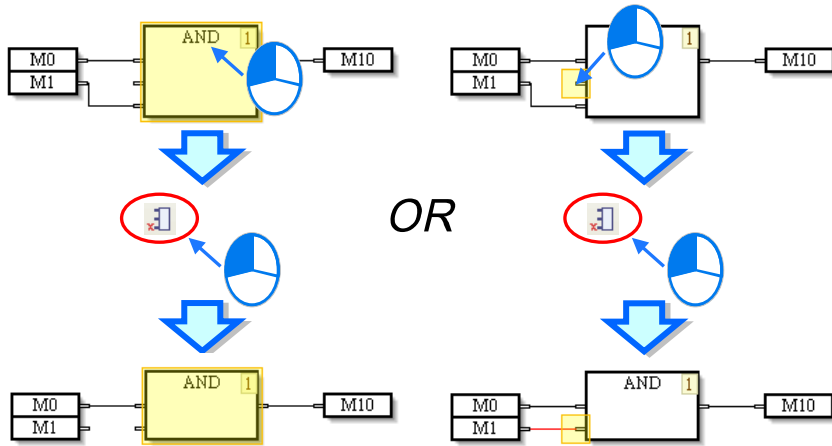
(3) 此邏輯閘便會在最下方新增一個輸入接腳，如下圖。



(4) 另一方面，若先選擇某個輸入接腳，再按下「新增輸入接腳」按鈕後，則會在選擇的輸入接腳上方插入一個輸入接腳，如下圖。




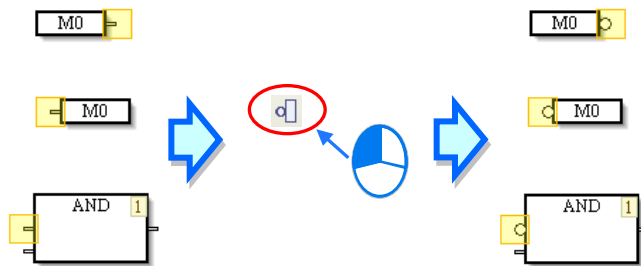
- (5) 當需要刪除接腳時，可以選擇邏輯閘後按下圖示工具列的「刪除輸入接腳」按鈕 ，此時會刪除邏輯閘最下方的輸入接腳；或著也可直接點選某個輸入接腳後按下「刪除輸出接腳」按鈕 ，刪除該接腳。




15.2.3 改變接腳的形式

可藉由以下的方式改變接腳的形式，以完成複雜的 CFC 程式設計。


- (1) 若需要將輸入或輸出的狀態取反相時，使用滑鼠點選節點、邏輯閘、功能塊、指令等等物件的接腳，使之成為被選取的狀態，再按下圖示工具列或是右鍵快捷選單的「反相」按鈕 ，便可使之進行狀態的反相。如下圖。若選取反相的接腳再次按下反相按鈕後，便取消反相的設定。

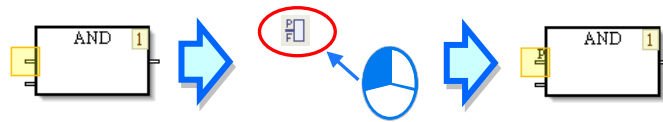


- (2) 若需要對輸出接腳的狀態進行設定或重置的設計時，選取輸出節點的接腳，再按下圖示工具列或是右鍵快捷選單的「設定/重置」按鈕 ，便可將此接腳進行設定或重置。如下圖。若在同一接腳多次按下此按鈕，則會將此接腳依序在設定、重置、取消設定之間變換。



15

- (3) 若需要對接腳的狀態進行正緣觸發或負緣觸發的設計時，選取邏輯閘、功能塊、指令等等物件的接腳，再按下圖示工具列或是右鍵快捷選單的「正/負緣」按鈕，便可將此接腳進行正緣或負緣觸發的設定。如下圖。若在同一接腳多次按下此按鈕，則會將此接腳依序在正緣、負緣、取消設定之間變換。



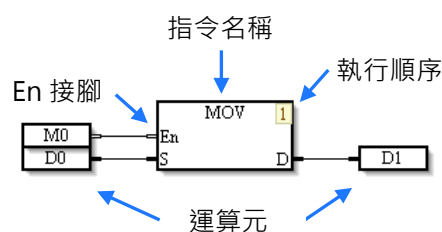
15.2.4 物件的連線與取消

欲連接已插入的物件，在其中一個物件的接腳按住滑鼠左鍵不放，此時可以連線的目標接腳將會以灰色顯示，滑鼠移動到另一個元件的最接近的接腳會出現箭頭符號，放開滑鼠左鍵將連接此二接腳。如下圖。反之若欲變更或取消物件的連線時，將線段拖曳到新的接腳或空白處即可。

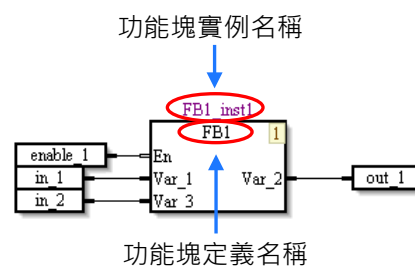


15.2.5 指令與功能塊

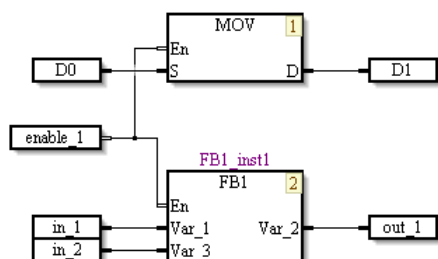
在 ISPSOft 的連續功能圖中，各種指令以方塊圖示呈現，方塊中列出指令名稱及運算元，右上角為 CFC 程式執行順序，另外透過 En 接腳連結前方的邏輯狀態，當 En 接腳的邏輯狀態為 ON 時，指令才會執行。如下圖。




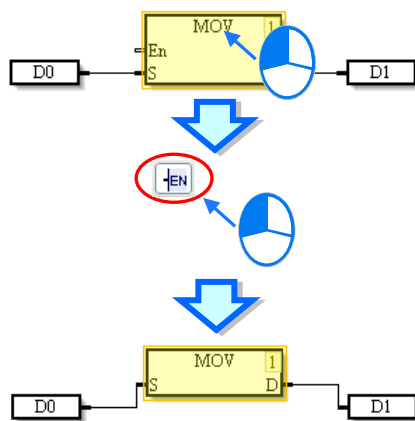
下圖則是 ISPSOft 連續功能圖中的功能塊符號，同樣透過 En 接腳連結前方的邏輯狀態，當前方的邏輯狀態為 ON 時，功能塊才會執行。功能塊符號上方的文字為該功能塊的符號名稱，也就是**功能塊實例**的名稱，關於功能塊的詳細介紹請參考手冊中**第 7 章**的相關說明。如下圖。



當需要透過同一個條件啟動多個指令或功能塊時，可以將此條件的輸入節點連接到各個指令與功能塊的 En 接腳，當此輸入為 ON 所連接的物件都將依照執行順序執行，如下圖的 enable_1 輸入，同時控制了圖中的指令與功能塊。



在不需要條件控制的情況，可以選取指令或功能塊後按下圖示工具列或右鍵選單的「En」按鈕 ，將 En 接腳關閉，此時 CFC 程式執行到此執行順序時皆會執行此指令或功能塊，如下圖。若再次按下此 En 按鈕，則會將此物件的 En 接腳再次開啟。

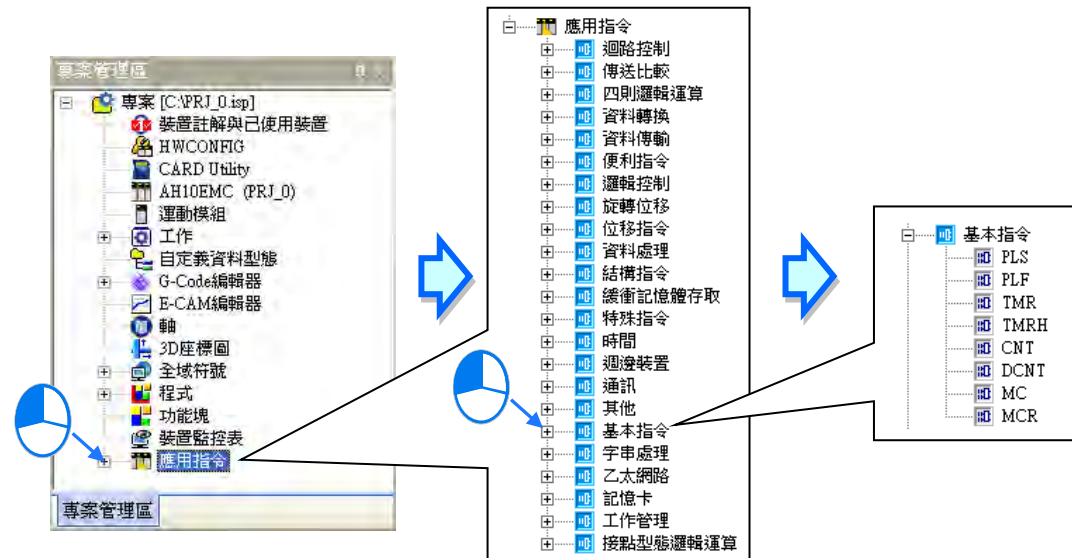


15.2.5.1 插入指令

插入指令可用下列兩種方式，請依使用習慣擇一即可。

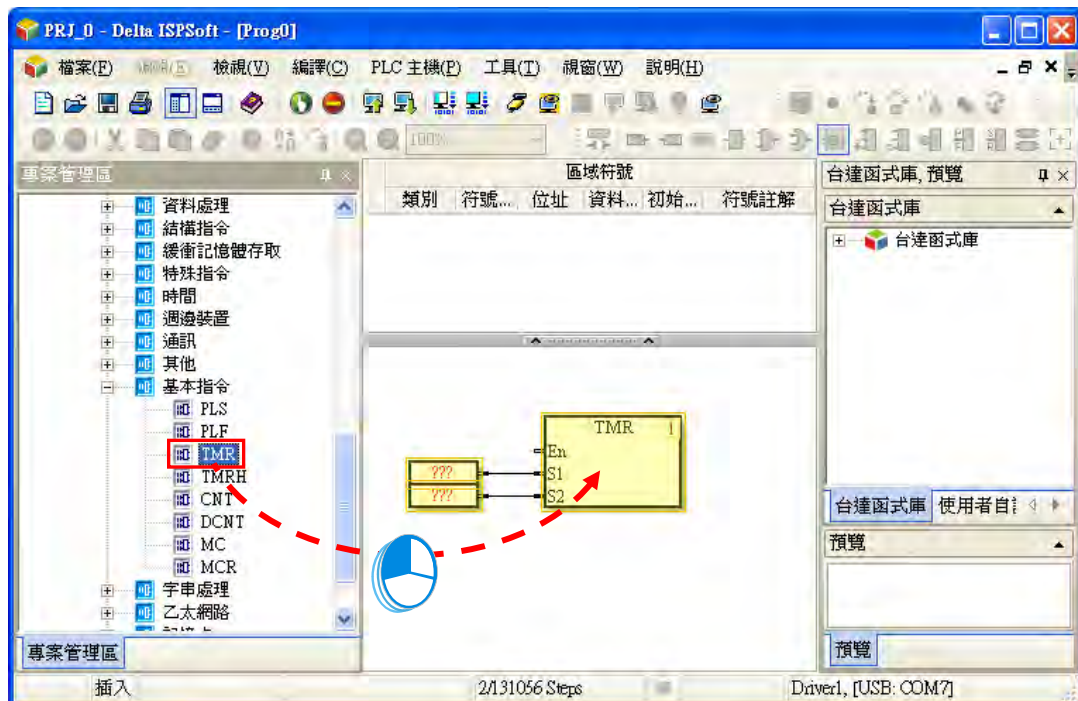
- 方法一：拖曳指令

(1) 展開應用指令項目，找到欲加入的指令。如下圖。



(2) 選擇要加入的指令按住滑鼠左鍵，拖曳到程式編輯區，在程式編輯區要插入指令的地方放開滑鼠左鍵，便在此處完成指令的建立。如下圖。

15



• 方法三：鍵盤輸入

先用滑鼠左鍵點選程式編輯區空白處，再直接用鍵盤輸入該指令後，畫面便會自動出現指令編輯視窗。待輸入完畢後按下鍵盤的「Enter」鍵，或以滑鼠點擊視窗的「確定」鈕，此時移動滑鼠游標，在程式編輯區需要插入指令的地方按下滑鼠左鍵，便在此處完成指令的建立（所輸入的指令無須分大小寫，且若輸入不正確的應用指令或對應裝置，按確定鍵時會有錯誤訊息）。如下圖。



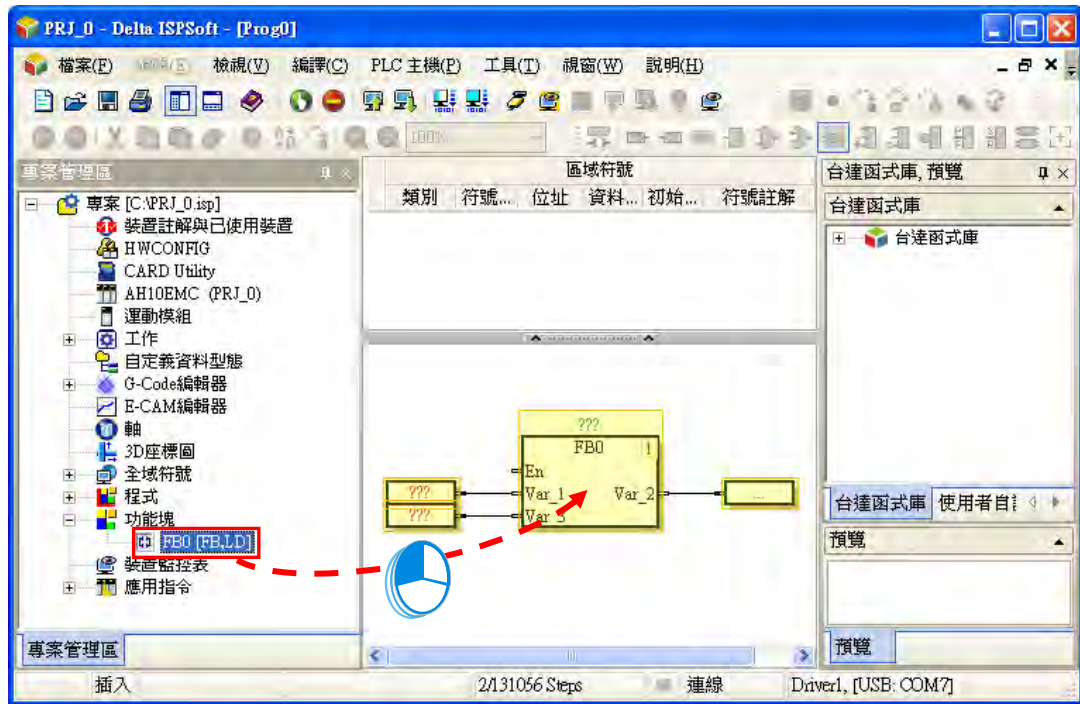
15

15.2.5.2 插入功能塊

插入功能塊的方法有二種，請依使用習慣擇一即可。

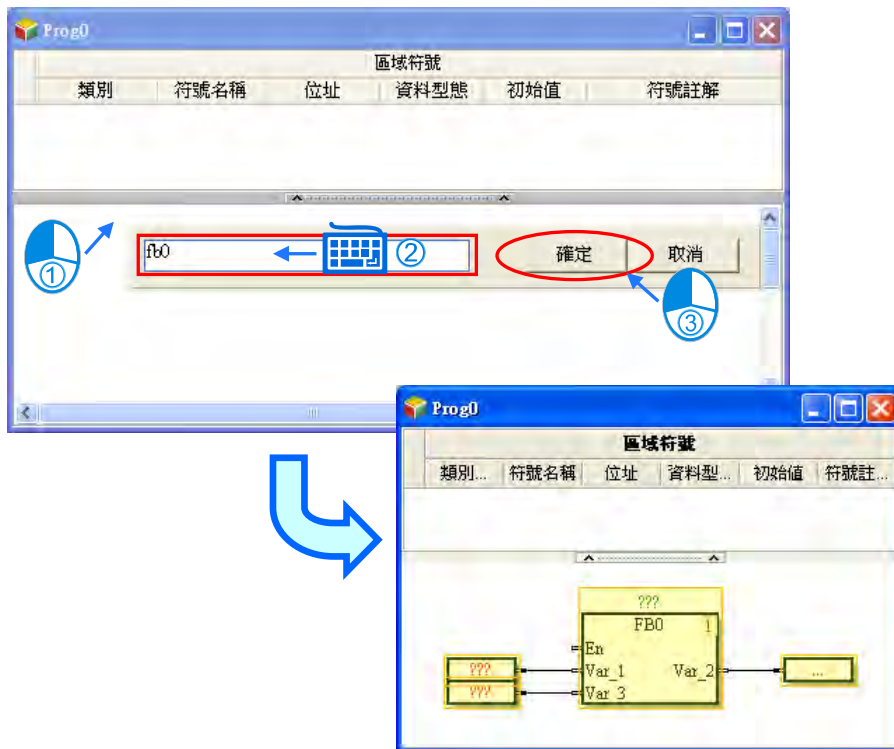
- 方法一：拖曳新增

點選欲插入的功能塊定義項目，按住滑鼠左鍵並將它拖曳至程式編輯區中欲插入的位置後放開，便在此處完成功能塊的建立，如下圖。



● 方法二：手動輸入


先用滑鼠左鍵點選程式編輯區空白處，再直接用鍵盤輸入該**功能塊定義**的名稱後，畫面便會自動出現指令編輯視窗，待輸入完畢後按下鍵盤的「**Enter**」鍵，或以滑鼠點擊視窗的「**確定**」鈕，此時移動滑鼠游標，在程式編輯區需要插入功能塊的地方按下滑鼠左鍵，便在此處完成功能塊的建立（所輸入的功能塊定義名稱無須分大小寫，且若輸入不正確的功能塊定義名稱或對應裝置，按確定鍵時會有錯誤訊息）。如下圖。

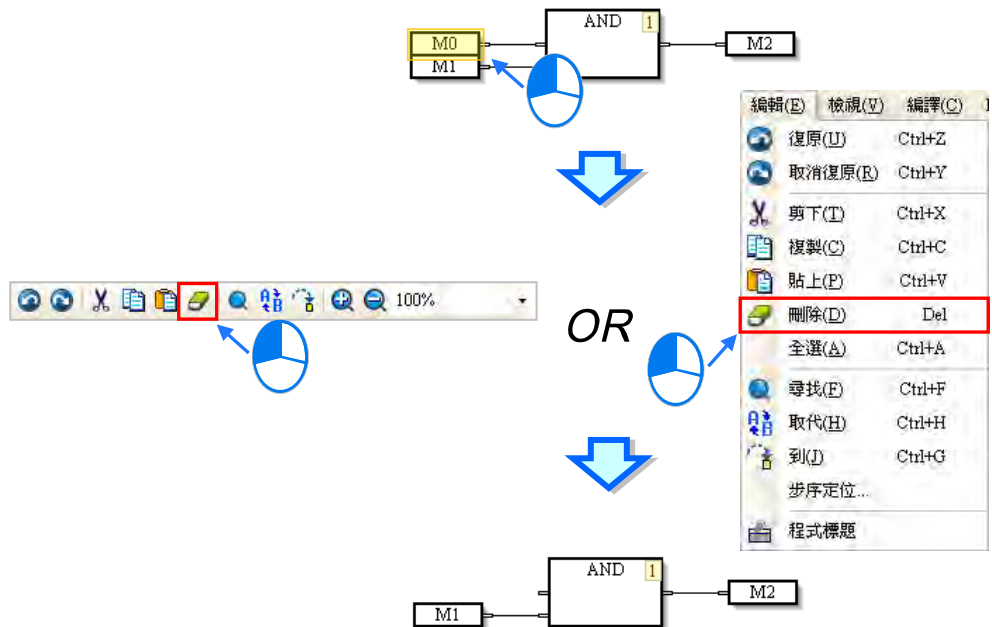


15

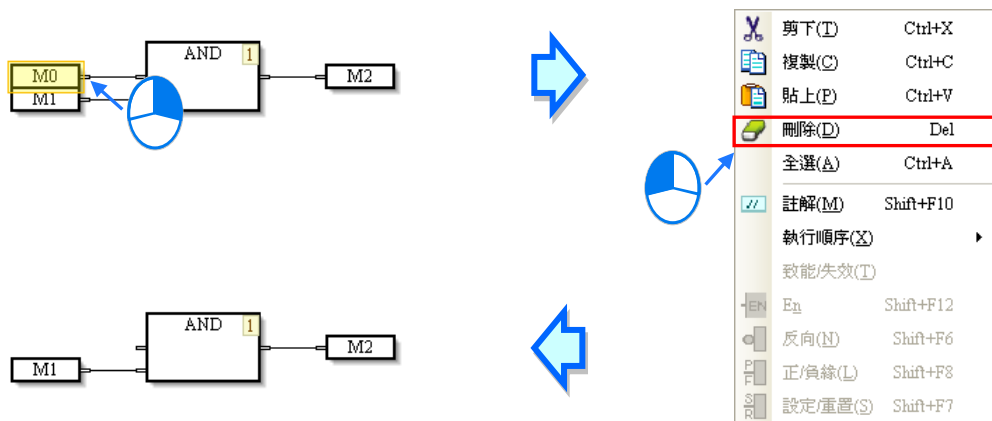
15.2.6 刪除物件

當需要刪除連續功能圖上的單一或多個物件時，可依下列方式刪除：

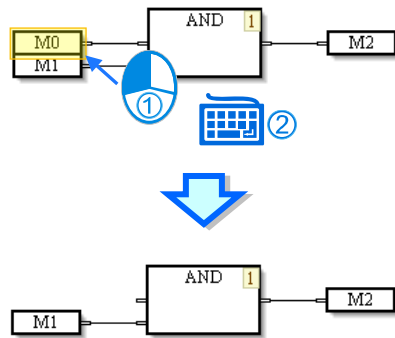
(1) 選取該物件，於功能工具列中點選 **編輯 (E) > 刪除 (D)** 或點擊圖示工具列  按鈕。如下圖。



(2) 選取該物件，按下滑鼠右鍵後，於快捷選單點選「**刪除 (D)**」。如下圖。



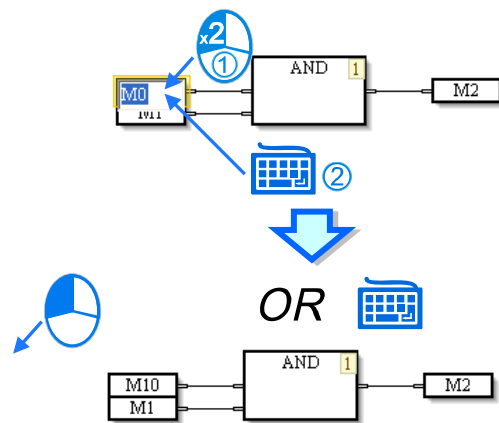
(3) 選取該物件，按下鍵盤「Delete」鍵，即可刪除。如下圖。



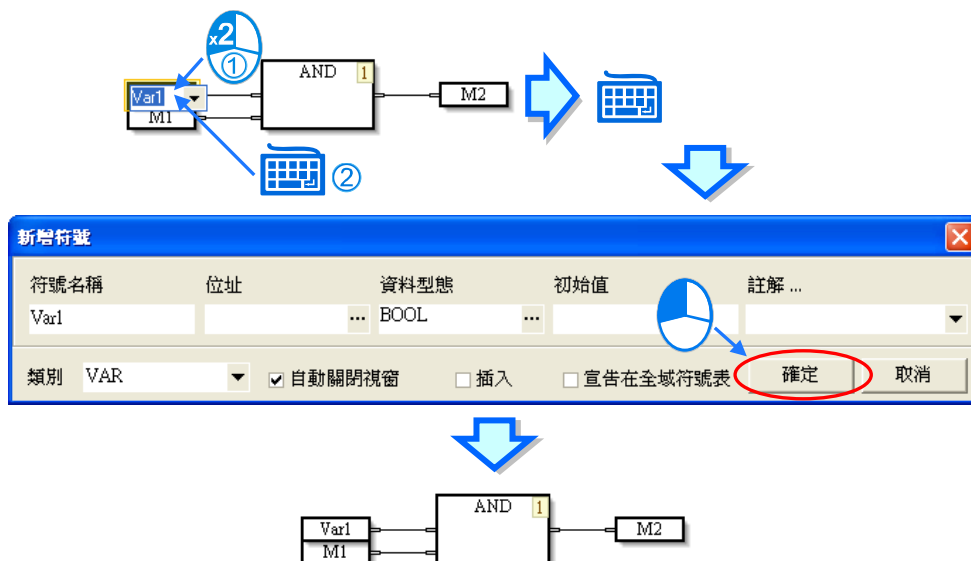
15.2.7 編輯裝置或符號

以滑鼠左鍵雙擊欲編輯的節點，直接輸入欲修改的裝置名稱，再點選程式編輯區空白處或按下鍵盤「Enter」鍵即可，如下圖。

15



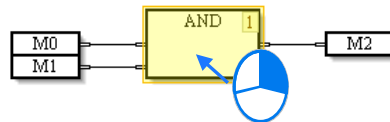
若於節點輸入變數符號，按下鍵盤的「Enter」鍵後，將出現新增符號視窗，完成設定後按下「確定」，便完成變數符號的宣告，如下圖，變數符號相關規定請參閱第 6 章。



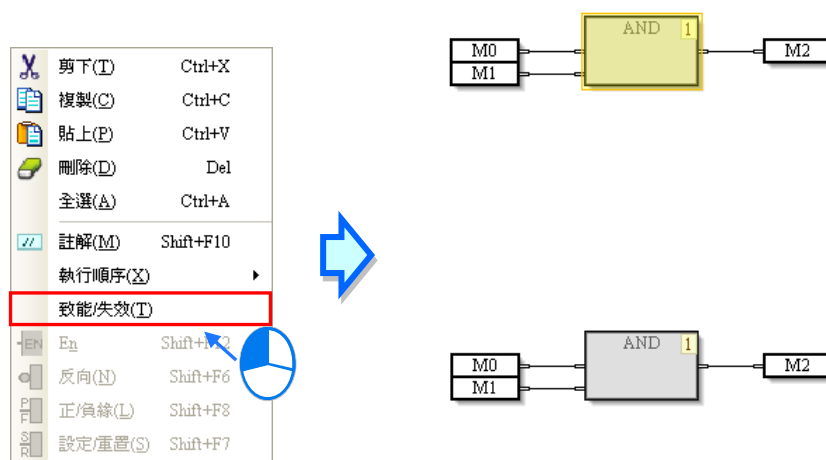
15.2.8 物件的致能與失效

若某個物件被設為失效狀態，則當程式進行編譯時，便會主動略過此失效物件的執行順序，因而此功能可被用來暫時關閉程式中的某些程序。

(1) 在欲失效的物件上點選滑鼠右鍵。如下圖。



(2) 於跳出的快捷選單選擇「致能/失效 (I)」，該物件會以暗色表示。如下圖。



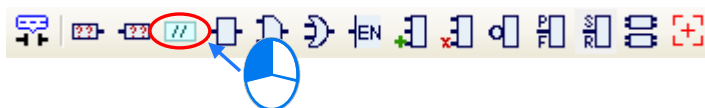
(3) 若須回復為致能物件，則重複上述步驟，於跳出的快捷選單選擇「致能/失效 (I)」即可。

15.2.9 插入註解

CFC 編輯器提供註解功能供使用者使用。共有兩種使用註解的方式說明如下。

• 方法一：在任意位置插入註解

(1) 點選圖示工具列的「註解」按鈕。如下圖。



(2) 在程式編輯區要放置的位置點擊滑鼠左鍵，便完成物件的插入。如下圖。



(3) 雙擊插入的註解物件，便開啟編輯視窗，於此輸入註解的說明內容。如下圖。若欲換行時請按下鍵盤的「Shift+Enter」鍵。

15



(4) 再點選畫面空白處或按下鍵盤的「Enter」鍵，完成註解的輸入。如下圖。此類註解為獨立於其他物件，不因其他物件的新增刪除或移動而改變。

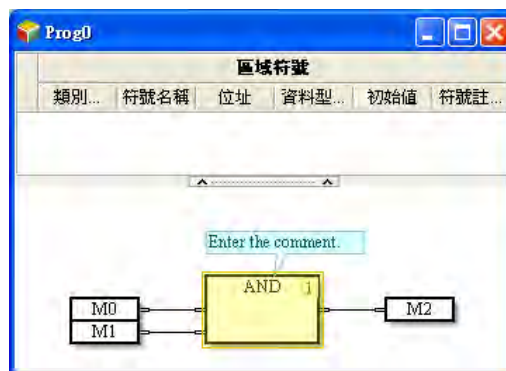


- 方法二：插入物件的註解

(1) 先選取要插入註解的物件。如下圖。



(2) 再點選圖示工具列的「註解」按鈕，會出現一個連接到此物件的註解，如下圖。同樣的可以於此註解物件輸入註解的說明內容。

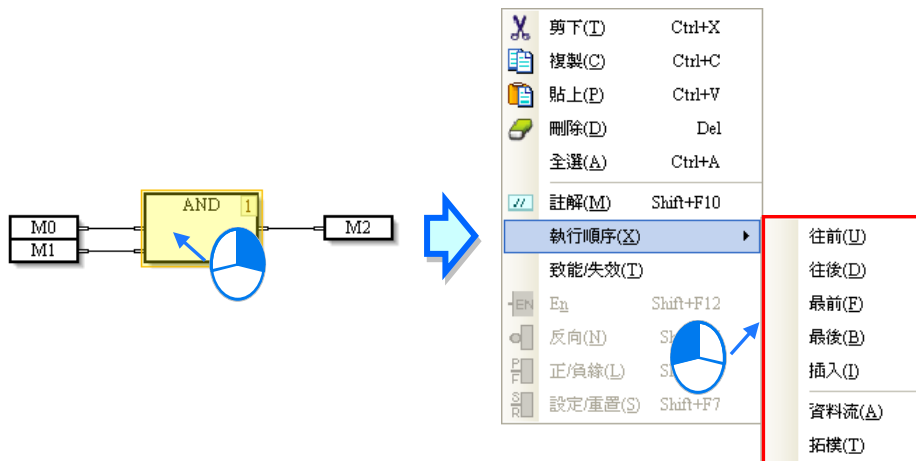



(3) 此類的註解會隨被選取物件的搬移而改變位置，當使用者搬移了此物件，連接其上的註解也將隨之移動。如下圖。



15.2.10 變更程式的執行順序


CFC 程式依據物件右上角的編號順序執行，當需要變更某物件的執行順序時，在此物件點擊右鍵，開啟快捷選單，展開「執行順序 (X)」選單，設定物件的執行順序，如下圖；設為「資料流」時，會依照物件之間的先後關係重新安排執行順序；設為「拓樸」時，會依照物件擺放的相對位置，由上至下，由左至右安排執行順序。

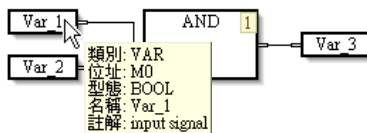


⚠ 變更執行順序會導致輸出結果改變，如不了解順序建議可使用工具列上的自動排列  按鈕。

15

15.2.11 顯示裝置註解

點選「顯示裝置註解」按鈕 ，成為按下的狀態後，將鼠標移至裝置或符號的文字上方會顯示該物件的相關資訊。資訊的內容包含位址與註解等。如下圖。



第16章 C 語言

目錄

16.1 C 語言編輯 (C Language)	16-2
16.1.1 C 語言之架構說明	16-2
16.1.2 C 語言編輯環境說明	16-3
16.1.3 C 語言編譯與除錯	16-9
16.1.4 內建函式庫指令說明	16-9
16.1.5 書籤與註解	16-10
16.1.6 範例操作說明.....	16-12

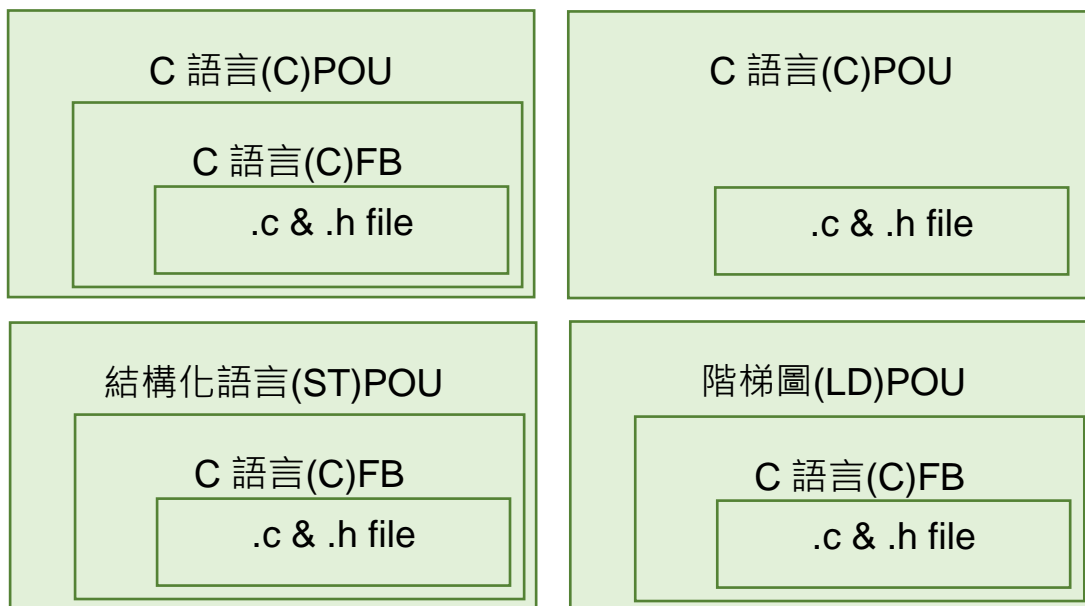
16.1 C 語言編輯 (C Language)

C 語言的特色在於整個程式是由許多的文字敘述所建構而成，而相較於指令列表 (IL)，C 語言則具有較高的可讀性，且可透過類似數學方程式的文法來表達一個複雜的運算式，因此對於一個需要進行繁複運算的程序來說，C 語言是較適合此種應用場合的開發語言。

*下載 C 語言的程式到主機之前，必須確認 AS 主機韌體是否為 V1.06.00 (含) 以上。

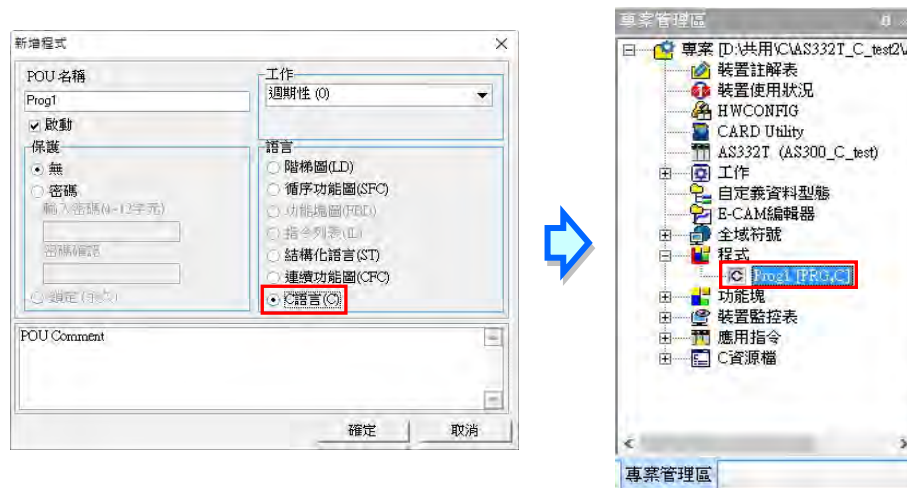
16.1.1 C 語言之架構說明

此編輯語言適合已熟悉 C 語言編輯之程式設計者，如欲開始採用 C 語言編輯前，請參考下圖 C 語言所支持的編輯語言架構；使用者可先使用市面上常見之 DEV-C、TurboC 以及 Visual Studio 等編輯軟體進行編輯、編譯與測試，之後在使用 ISPSOft 提供之匯入功能，匯入已經完成之 C 語言 ".C" 與 ".h" 檔案並使用在 C 語言 (C) FB 或 C 語言 (C) POU，如下圖，也可以在結構化語言 (ST) POU 或階梯圖 (LD) POU 引用 C 語言 (C) FB。



16.1.2 C 語言編輯環境說明

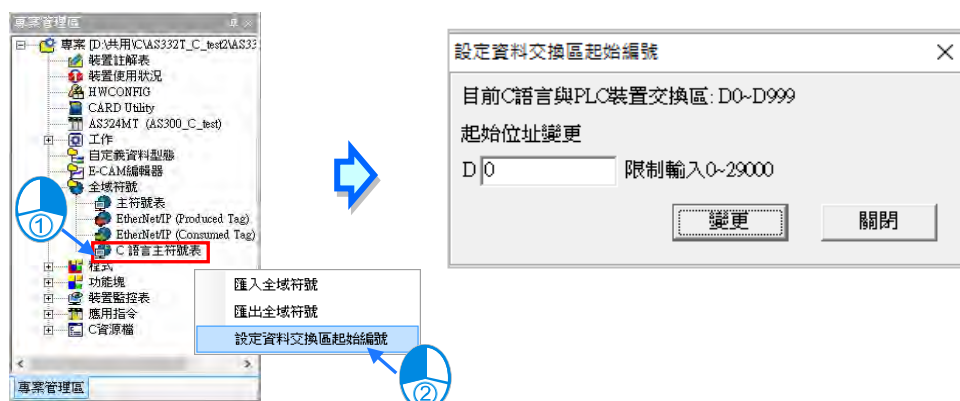
AS 系列主機提供的 C Code 執行碼儲存空間 (ROM) 為 64K Bytes，以及變數宣告可用空間 (RAM) 為 8K Bytes。以上兩個儲存空間與現有的 PLC 程式使用空間皆各自獨立，每次編譯完成之後，建議從編譯訊息框內關注剩餘空間。運用 ISPSOft 提供的 C 編輯畫面，進行 C 語言程式撰寫，建立 POU 時，於語言欄位選擇「C 語言 (C)」便可建立 C 程式。



在 C 的編輯工作區中，只支持標準 C 語言寫法，其他如：PLC 指令、C++ 或 Visual C... 等高階語法不支持。其操作方式與一般的文字編輯器類似，編輯時只要依據 C 語法的格式，直接使用鍵盤在工作區上面輸入或修改文字即可，且當按下鍵盤的【Enter】鍵時便可執行換行的動作

```
0019 C_Var_0 = 123;
0020
0021
```

進行變數宣告時，可使用矩陣 (Array)、結構 (Struct) 與指標 (Point) 等方式，但須注意變數可宣告之記憶體有限，因此不可超出範圍使用，否則嚴重可能會造成 PLC 當機之狀況。必須在全域符號的 C 語言主符號表，新增變數符號，在 C 語言符號可使用的裝置位址有 1000 個 D 裝置，可以自行設定起始位址範圍，在「C 語言主符號表」，單擊滑鼠右鍵之後再於快捷選單中點選「設定資料交換區起始編號」便可設定資料交換區的起始位置，起始位址可從 D0~D29000 填一個，系統會自動往後面設定連續 1000 個 D 裝置，設定完成會顯示在目前資料交換區範圍。

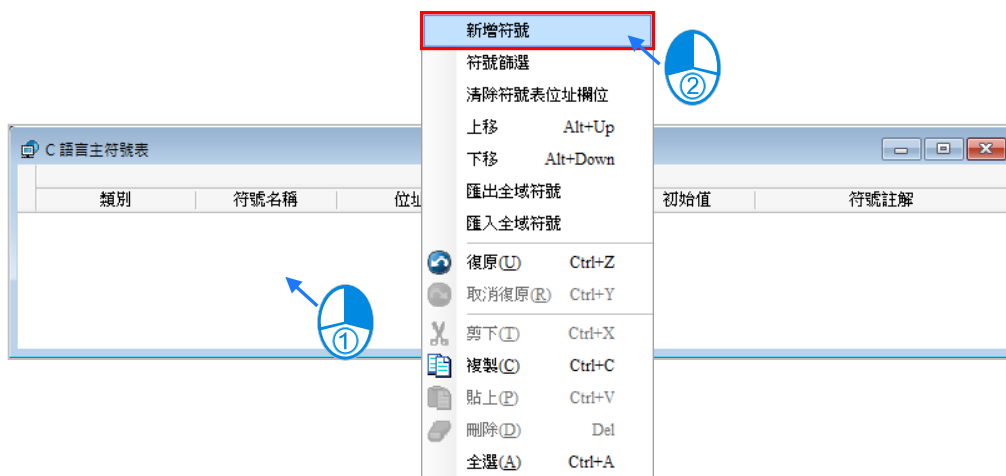


接著雙擊「C 語言主符號表」會彈出 C 語言主符號表視窗。

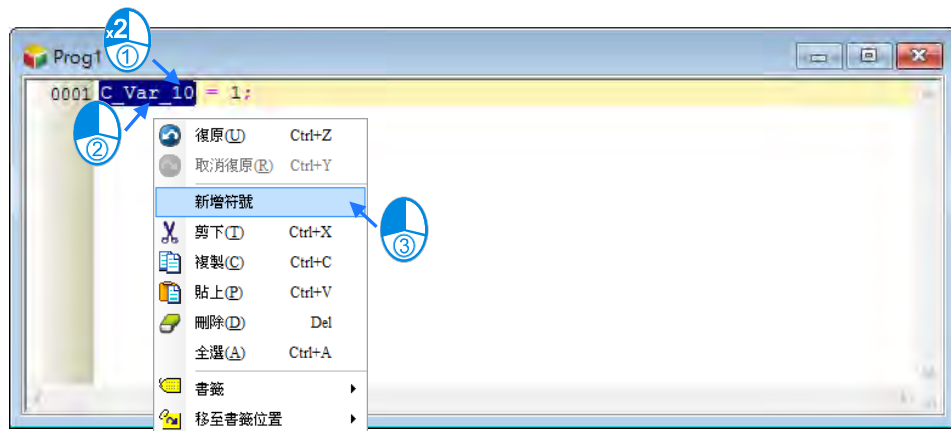


16

在 C 語言主符號表的空白處，使用滑鼠點擊右鍵，之後再於快捷選單中點選「新增符號」便可自動導出變數符號的宣告視窗。C 語言變數符號只支援 BOOL、WORD、DWORD、REAL 四種型態，位址起始位址為 D0~D999，往後面的連續 1000 個 D 裝置，宣告完之後的變數符號可以直接在 LD 語言與 C 語言的程式內使用。



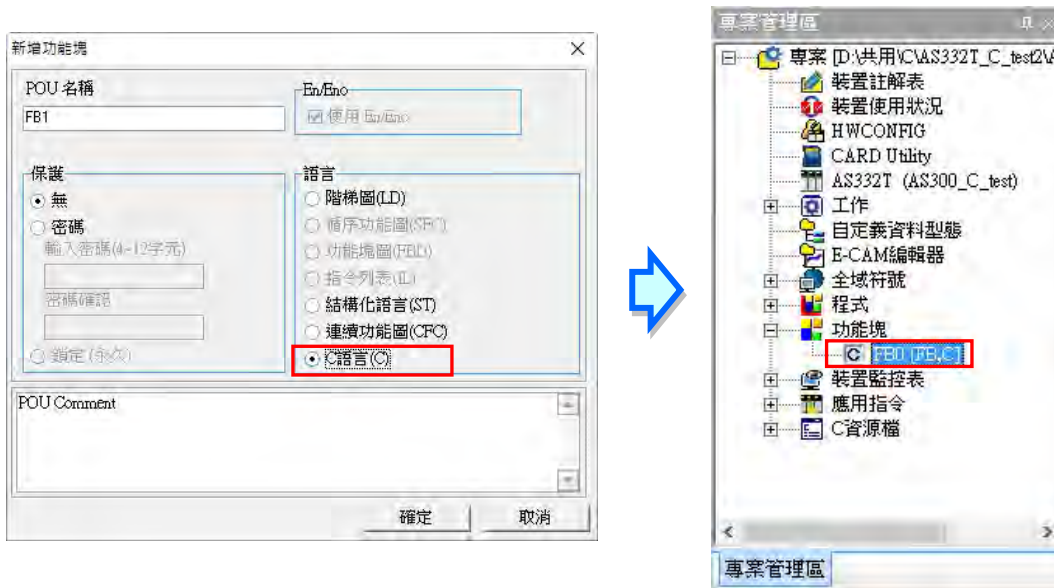
在編輯的過程中輸入了未經宣告的變數符號時，輸入完畢後用滑鼠雙擊該文字並按下右鍵，之後再於快捷選單中點選「新增符號」便可自動導出變數符號的宣告視窗。



程式輸入的過程中，會自動從以建立的變數符號中篩選出符合的變數符號供使用者挑選，可利用鍵盤的「↑」、「↓」鍵，或直接在選單中點擊滑鼠左鍵完成輸入。

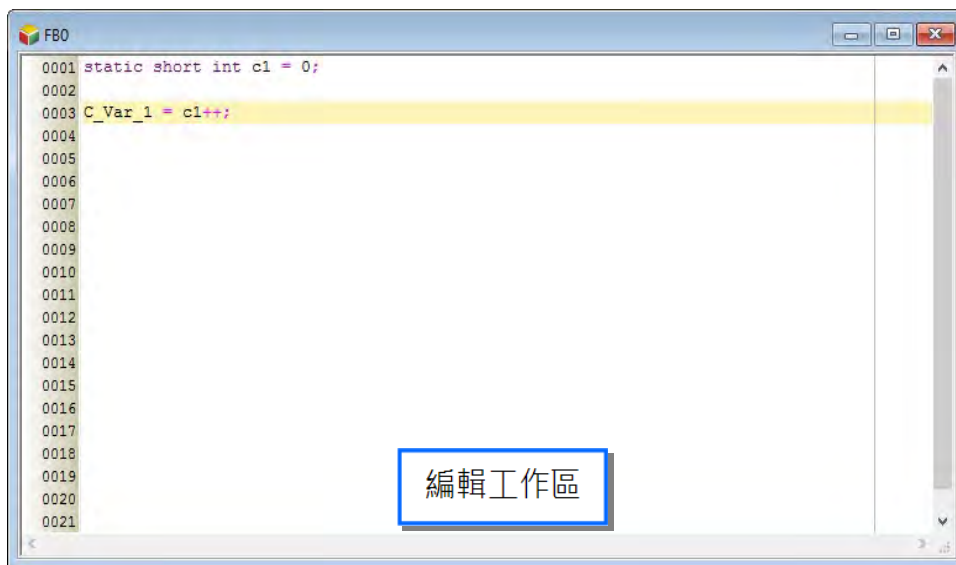


建立功能塊時，於語言欄位選擇「C 語言 (C)」便可建立 C 功能塊。



16

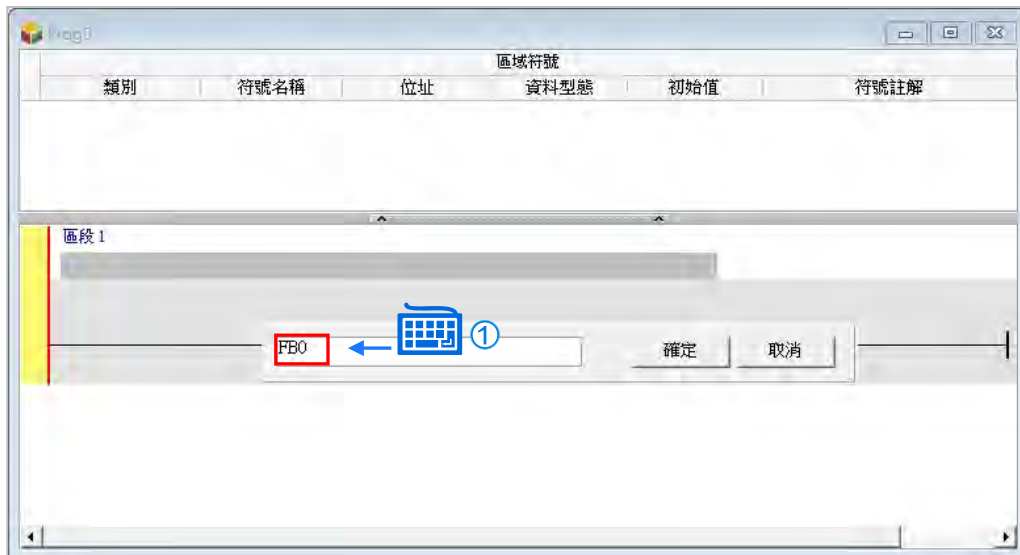
C 功能塊的編輯方式與 C 程式相同，依據 C 語法的格式，直接使用鍵盤在工作區上面輸入或修改文字即可。



插入 C 語言功能塊到 LD 語言程式的方法有下述兩種。

方法一

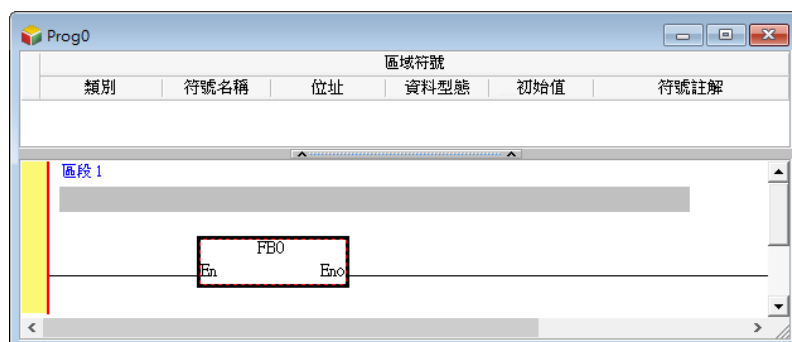
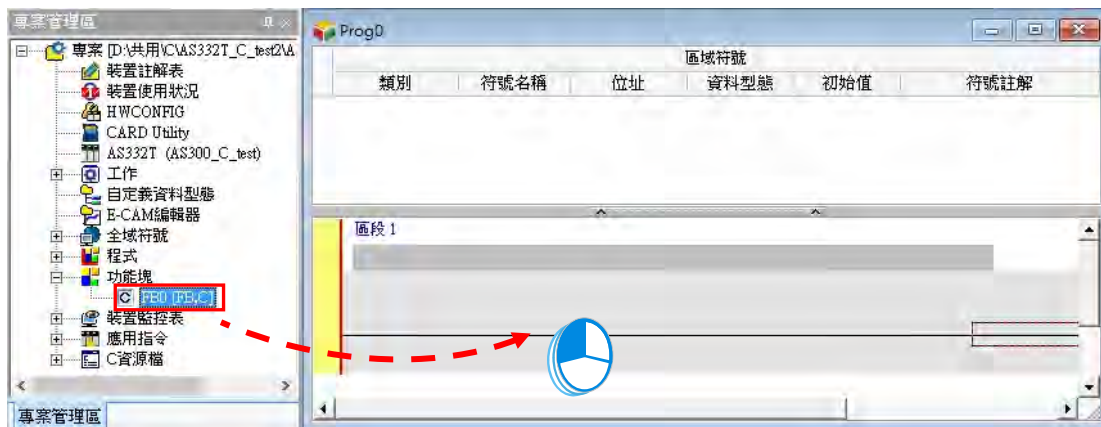
依據功能塊名稱，直接於 LD 語言的編輯工作區中利用鍵盤輸入即可。



16

方法二

於專案管理區中展開功能塊的目錄，在其中找到欲插入的項目後，於該項目上按住滑鼠左鍵，之後再將其拖曳至欲插入的位置後放開左鍵即可。



匯入已經完成之 C 語言".c"與".h"檔案，先於「C 資源檔」按下滑鼠右鍵，接著於快捷選單中點選「新增」項目，之後在選擇要匯入的 C 檔視窗加，選取要匯入的 C 檔案，如下圖。若要匯入.H 檔案則需要在「檔案類型」項目中選取 C headers (*.h)。

16



16.1.3 C 語言編譯與除錯

ISPSoft 提供標準的 GCC 編譯器，以及 PLC 已內建的三角函數與指數類函式庫，使用者不用宣告就可以使用，但不提供市面上其他外掛的標準 C 函式庫（如：`math.h`），以節省 PLC 內部提供的 C code 儲存空間。

```

C:\Windows\system32\cmd.exe
C:\ProgramData\Oracle\Java\javapath;C:\Windows\
stem32\WinSxS;C:\Windows\System32\WindowsPowerShe
penCL SDK\2.0\bin\x86;C:\Program Files\Lenovo\Ar
es\Common Files\lenovo\easyplusdk\bin;C:\Progra
rogram Files\Intel\WiFi\bin\;C:\Program Files\C
;C:\Program Files\Skype\Phone\;E:\PLC\Software\
compilation: ...
compilation: finished.
generate test.bin ...
generate bin file: finished
generate dis file ...
generate dis file: finished
請按任意鍵繼續...
  
```



ISPSoft 只顯示 GCC 編譯後的錯誤訊息，但無法支持點選錯誤描述之後，直接跳到錯誤位置修改的功能，使用者可在現有 PLC 程式與功能塊進行線上監看，但無法對 C 語言編輯的程式進行一行一行的 C Code 進行監看。

16.1.4 內建函式庫指令說明

一般常使用之整數型態或浮點數型態之四則運算“加、減、乘、除”，ISPSoft 都已內建，不需額外宣告即可使用。

已內建之三角函式名稱：

C 函式名稱	ISPSoft 編輯名稱	指令格式	功能說明
<code>sinf</code>	<code>DELTA_SINF</code>	Float DELTA_SINF (float radian) ;	Sin 運算
<code>cosf</code>	<code>DELTA_COSF</code>	Float DELTA_COSF(float radian) ;	Cos 運算
<code>tanf</code>	<code>DELTA_TANF</code>	Float DELTA_TANF (float radian) ;	Tan 運算
<code>asinf</code>	<code>DELTA_ASINF</code>	Float DELTA_ASINF (float arg) ;	Sin^{-1} 運算
<code>acosf</code>	<code>DELTA_ACOSF</code>	Float DELTA_ACOSF (float arg) ;	Cos^{-1} 運算
<code>atanf</code>	<code>DELTA_ATANF</code>	Float DELTA_ATANF (float arg) ;	Tan^{-1} 運算

C 函式名稱	ISPSOft 編輯名稱	指令格式	功能說明
sinhf	DELTA_SINHF	Float DELTA_SINHF (float arg) ;	Sinh 運算
coshf	DELTA_COSHF	Float DELTA_COSHF (float arg) ;	Cosh 運算
tanhf	DELTA_TANHF	Float DELTA_TANHF (float arg) ;	Tanh 運算

已內建之指數型函式名稱

C 函式名稱	ISPSOft 編輯名稱	指令格式	功能說明
sqrtf	DELTA_SQRF	Float DELTA_SQRF (float value) ;	平方根運算
logf	DELTA_LOGF	Float DELTA_LOGF (float base · float value) ;	Log 運算
lnf	DELTA_LNF	Float DELTA_LNF (float arg) ;	Ln 運算
powf	DELTA_POWF	Float DELTA_POWF (float base · float power) ;	乘幕運算

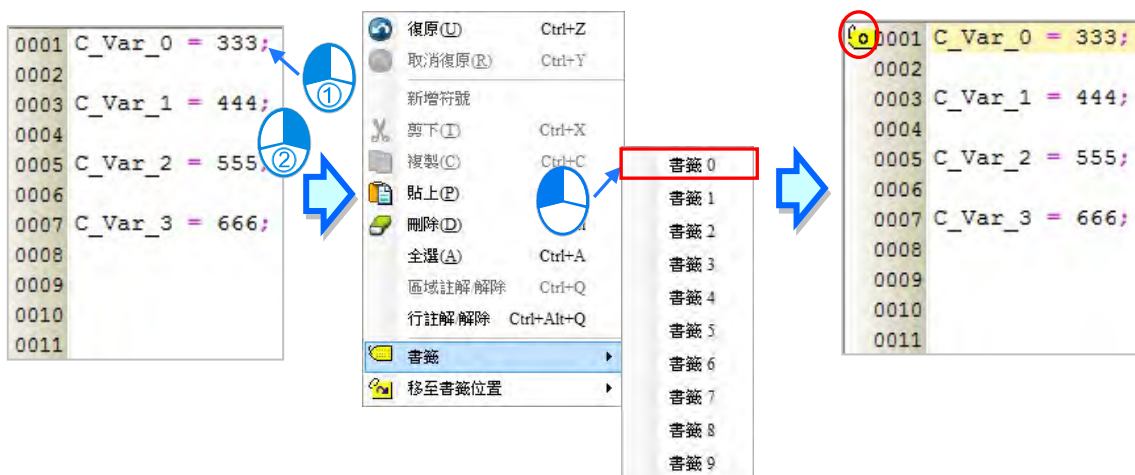
16

16.1.5 書籤與註解

書籤功能為方便搜尋或移動至特定編輯位置的便利工具。

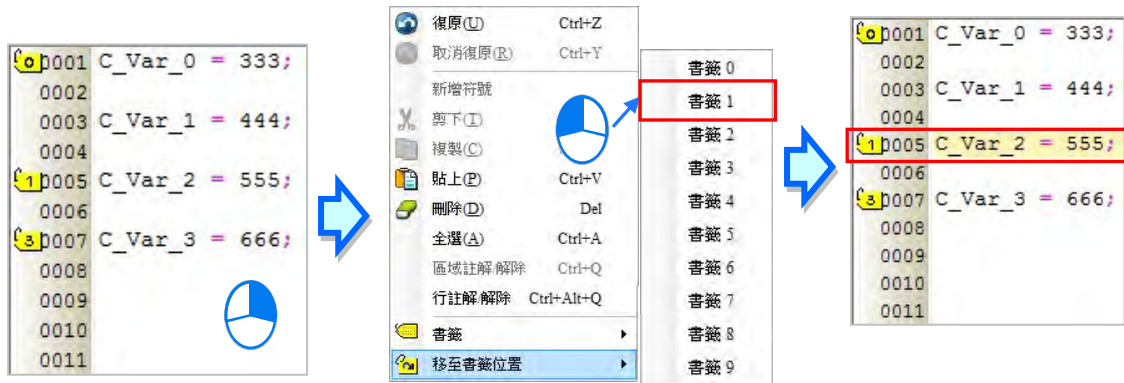
●加入/移除書籤

先用滑鼠點選欲加入書籤的行列後按下滑鼠右鍵，接著於快捷選單中點選「書籤」項目，之後再於該項目中選擇欲加入的書籤編號即可。而要移除書籤時，同樣的請先用滑鼠點選欲取消書籤的行列後按下滑鼠右鍵，接著於快捷選單中點選「書籤」項目，之後再於該項目中點選與所選行列相同編號的書籤即可；不過若選擇的是其它編號時，所選行列的書籤便會被變更為該編號。此外，若於功能工具列中點擊 **編輯 (E) > 書籤 > 移除所有書籤 (R)**，則目前編輯視窗中的中的所有書籤便會被全部取消。操作時若點選了一個已使用於其他行列的編號，則該編號的書籤便會被搬移至目前選取的行列上。



●移至書籤位置

於編輯工作區中按下滑鼠右鍵，接著於快捷選單中點選「移至書籤位置」項目，之後在於目錄中選擇移動目標的書籤編號即可。



註解功能為方便使用者，對選取區域或選取的行數，進行註解的工具。

● 行註解/解除

先用滑鼠點選欲註解的行列後按下滑鼠右鍵，接著於快捷選單中點選「行註解/解除」項目，之後在該行列的前面會自動加上 // 的註解。而要移除註解時，同樣的請先用滑鼠點選欲取消註解的行列後按下滑鼠右鍵，接著於快捷選單中點選「行註解/解除」項目，之後在該行列的前面 // 的註解會自動移除。



● 區域註解/解除

先用滑鼠左鍵選取欲註解的區域後按下滑鼠右鍵，接著於快捷選單中點選「區域註解/解除」項目，之後在該選取區域的前後會自動加上 / * 與 * / 的註解。而要移除註解時，同樣的請先用滑鼠左鍵選取欲取消註解的區域後按下滑鼠右鍵，接著於快捷選單中點選「行註解/解除」項目，之後在該區域的前後 / * 與 * / 的註解會自動移除。



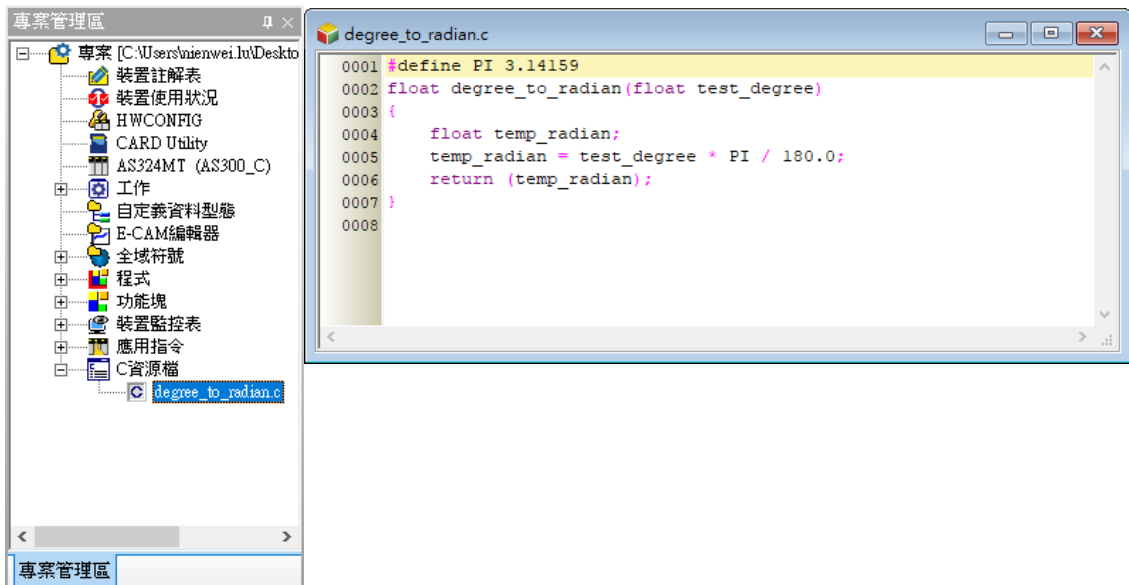
16.1.6 範例操作說明

範例：每 10ms 以每次 5 度變換輸出一筆 Sin 函數數值，並且用資料示波器功能，記錄下 0~ 355 度的輸出數值。

步驟 1：先建立 PLC 與 C 語言程式所需共用之變數符號名稱參考第 16.1.2 節，如下圖所示。

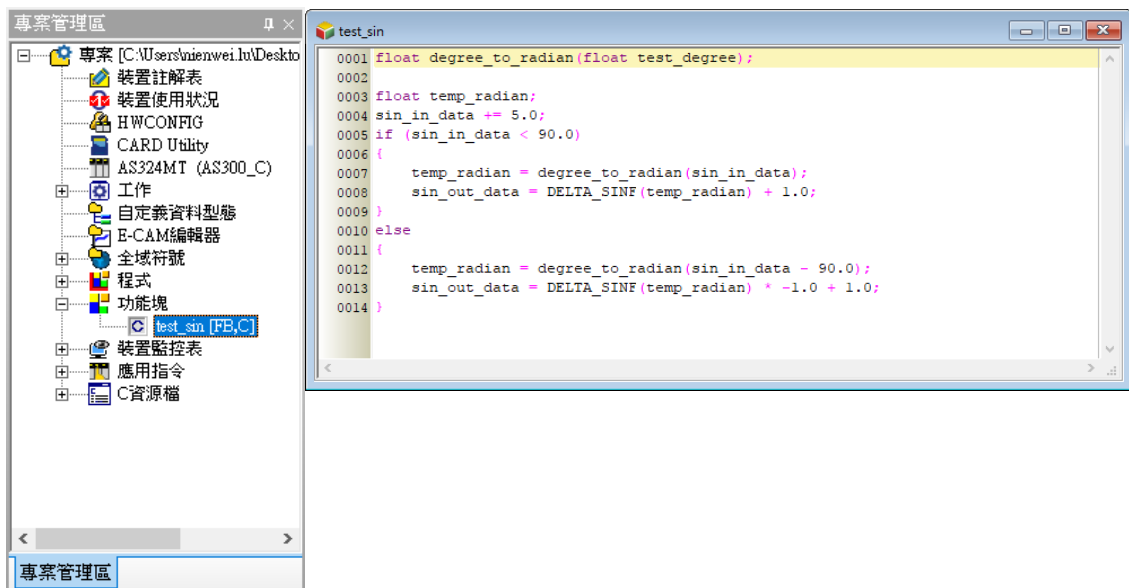


步驟 2：匯入一個自己已經編輯且完成測試的函數 `degree_to_radian.c` 參考第 16.1.2 節。此步驟使用者可依狀況自行刪減。

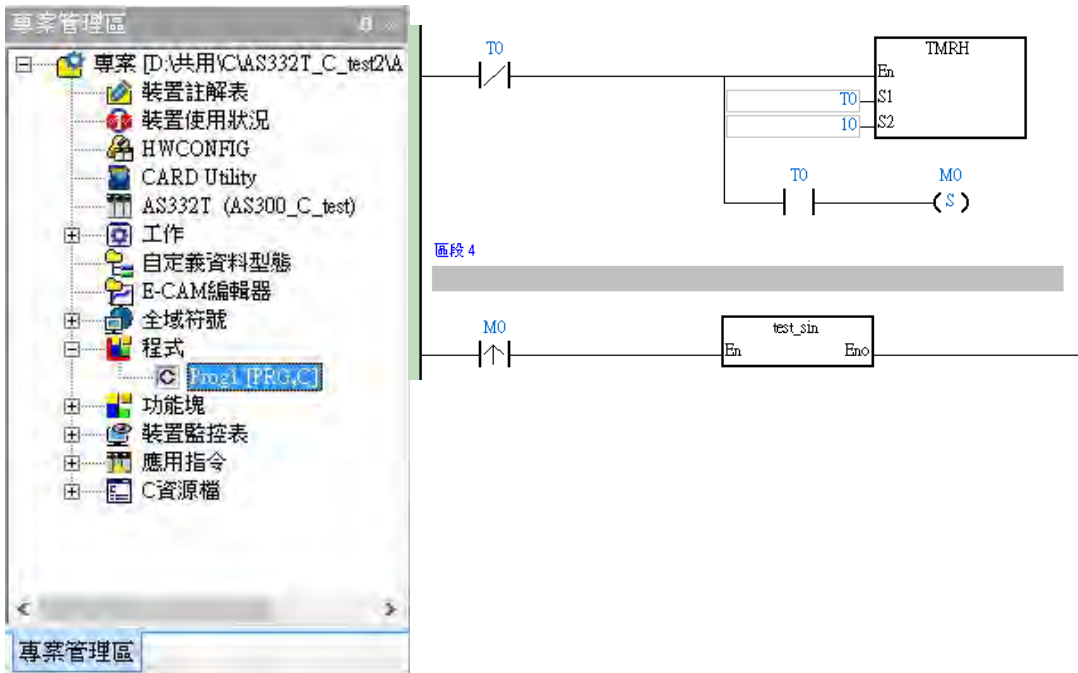


16

步驟 3：編輯一個新功能塊 `test_sin`，並寫入計算過程，如下圖所示。其中 `degree_to_radian` 函數庫。

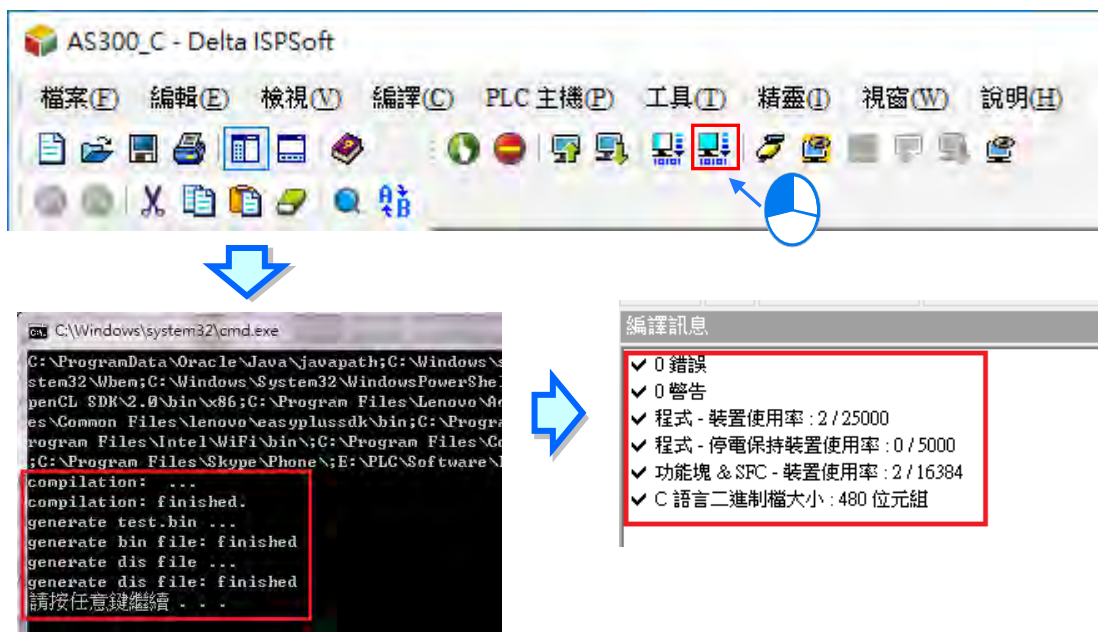


步驟 4：建立一個程式 (POU)，並寫一段每 10ms 呼叫一次 test_sin 的功能塊 (FB)，如下圖所示。

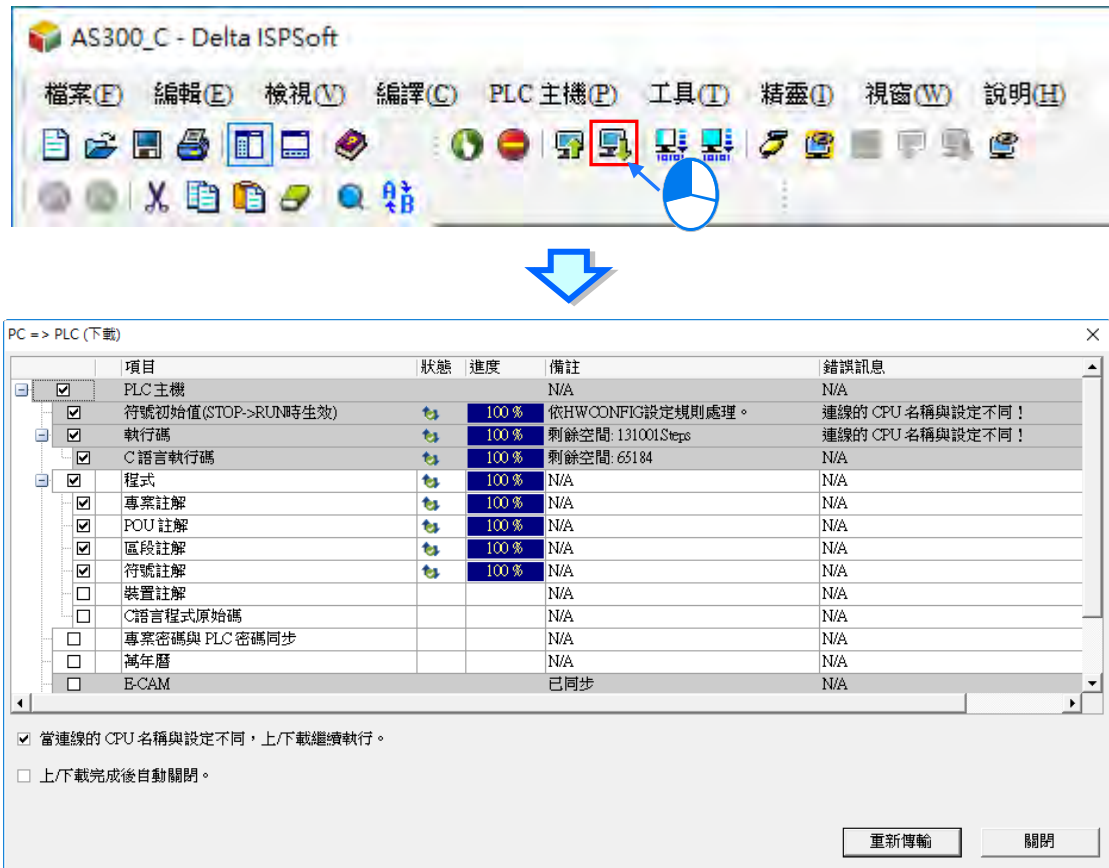


16

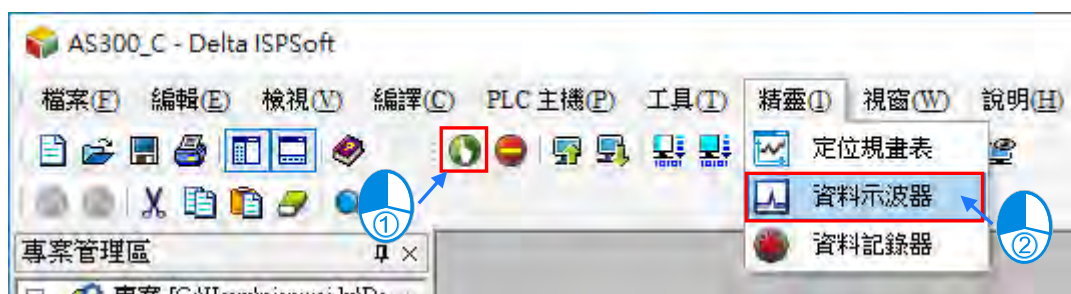
步驟 5：按下編譯 (Compiler) 與下載 PLC 程式，假設都無錯誤發生，則過程如下圖所示。



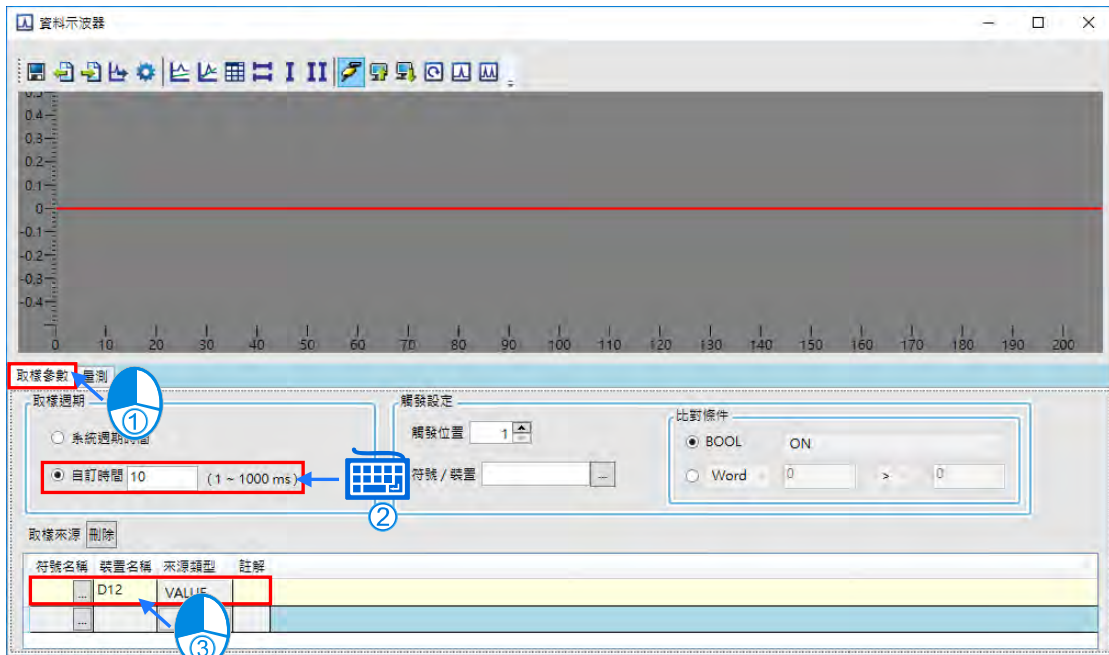
步驟 6：編譯 (Compiler) 成功後下載 PLC 程式。



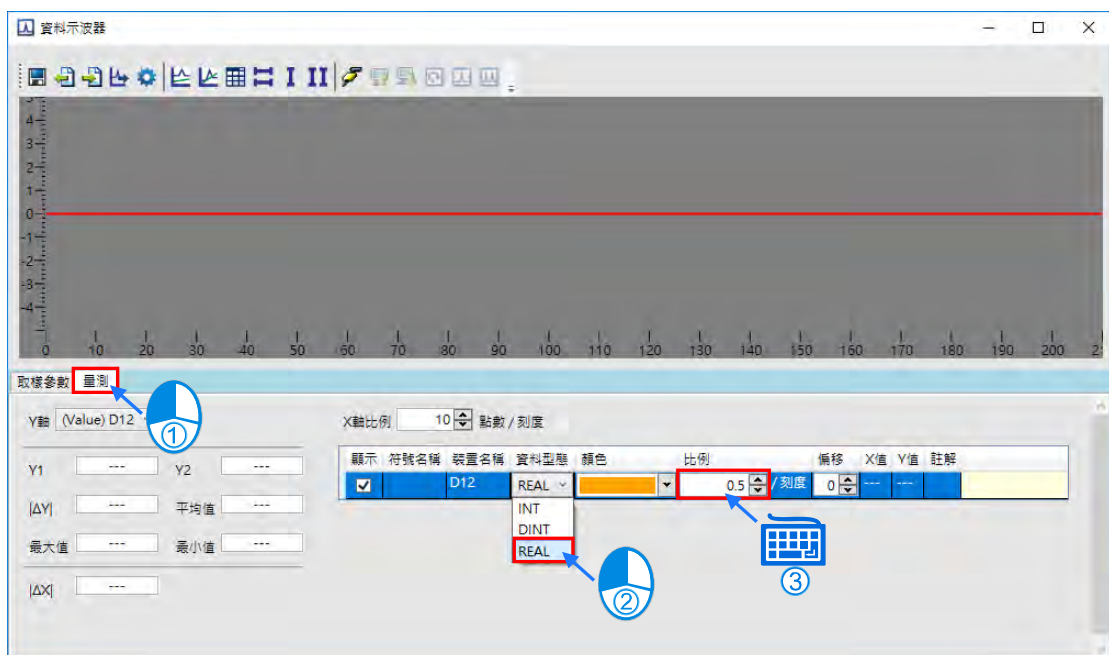
步驟 7：啟動 PLC Run 之後，開啟資料示波器，其過程如下圖所示。



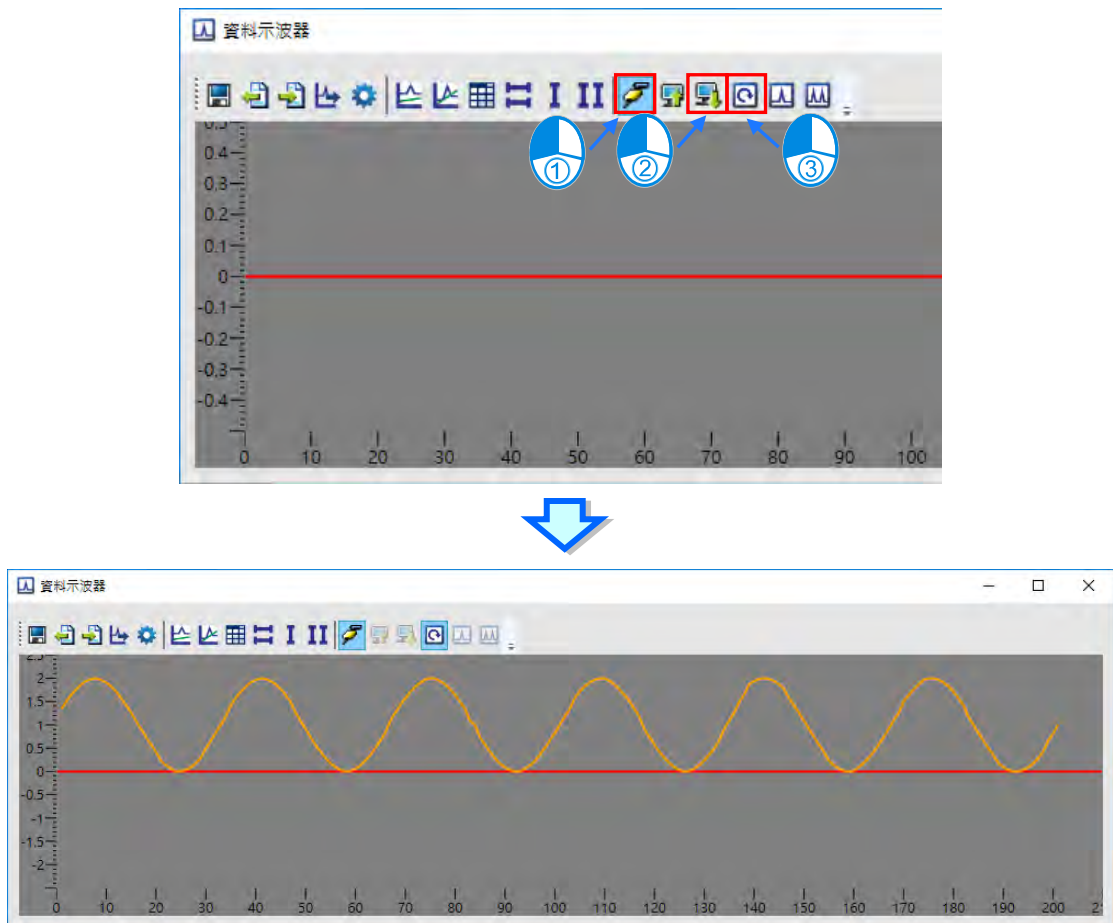
步驟 8：資料示波器設定取樣參數及量測，其過程如下圖所示。



16



步驟 9：資料示波器連線、下載、即時監看，其過程如下圖所示。



MEMO

16

第17章 編輯輔助工具

目錄

17.1 ISPSOft 的編輯工具與輔助功能.....	17-2
17.1.1 變更 PLC 機種.....	17-2
17.1.2 將 DVP 機種轉換為 AS 機種.....	17-3
17.1.3 專案的下載與上傳.....	17-15
17.1.4 LD、FBD 與 CFC 編輯環境的尋找取代與到功能.....	17-17
17.1.5 IL 與 ST 編輯環境的尋找取代功能.....	17-19
17.1.6 SFC 編輯環境的尋找取代功能.....	17-23
17.1.7 符號表的尋找功能.....	17-25
17.1.8 專案尋找與取代.....	17-26
17.1.9 列印功能.....	17-28
17.2 裝置與暫存器管理.....	17-30
17.2.1 DVP 機種的裝置註解表.....	17-30
17.2.2 AH/AS 系列機種的裝置註解表.....	17-34
17.2.3 裝置使用狀況.....	17-40
17.2.4 暫存器編輯.....	17-43
17.2.5 裝置狀態編輯.....	17-50
17.2.6 DVP 機種的檔案暫存器編輯.....	17-54
17.3 步序定位.....	17-61
17.3.1 使用步序定位.....	17-61
17.4 程式比對.....	17-62
17.4.1 程式比對簡介.....	17-62
17.4.2 與檔案比對.....	17-63
17.4.3 與 PLC 比對.....	17-64

17.1 ISPSOft 的編輯工具與輔助功能

17.1.1 變更 PLC 機種

ISPSOft 允許使用者變更專案機種，大部分的系列機種提供雙向互轉而部分機種只提供單向轉換，如 DVP 機種可以轉換到 AS 機種，但是 AS 機種不能轉換到 DVP 機種；AH5x0 機種可以轉換到 AH560 機種，但 AH560 機種不能轉換到 AH5x0 機種，詳細可以參考下表。

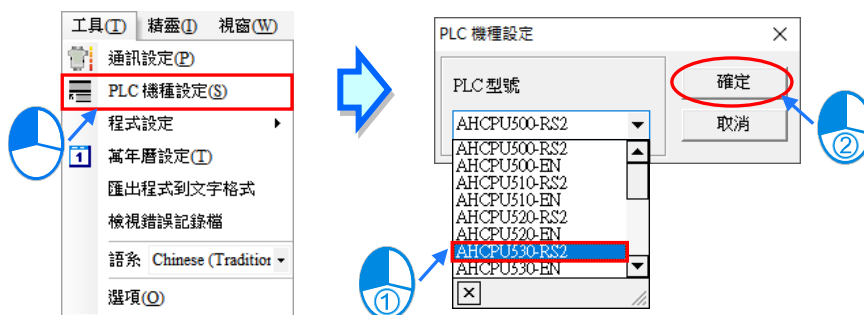
目的 來源	AS	AH5x0	AH5x1	AH560	AHxxEMC	DVP	AS5xx DVPxxMC
AS	V	V	V	V	V	只支援 ES3	
AH5x0		V	V	V	V		
AH5x1			V	V	V		
AH560			V	V	V		
AHxxEMC					V		
DVP	V					V(*1)	
AS5xx DVPxxMC							V

*1. DVP-ES3 機種不能轉換到其他 DVP 機種，但其他 DVP 機種可以轉換到 DVP-ES3 機種。

變更機種會將目前專案的內容改變，所以在變更機種之前建議將專案另外儲存一份，且變更前後的機種功能、裝置範圍或支援的指令可能都會有所差異，因此變更後請務必再次確認原本的程式、參數設定、硬體規劃與網路規劃...等部分，是否可讓系統正常運作。

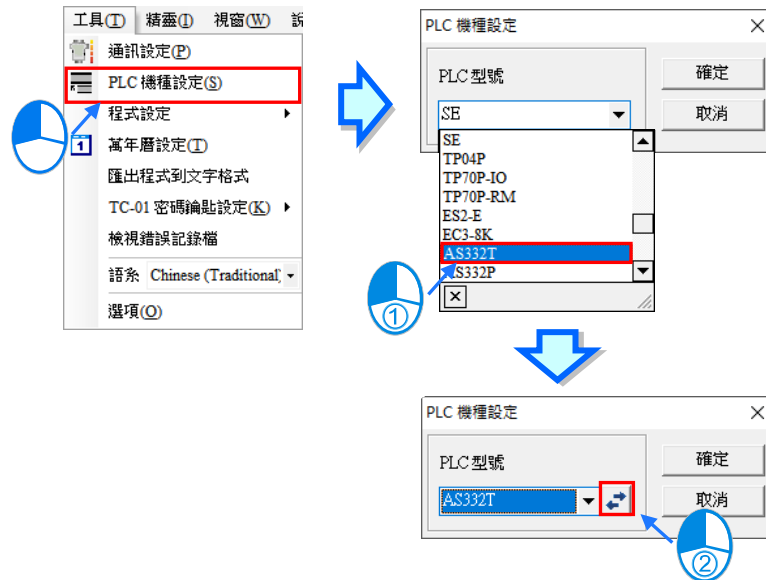
17

當要變更專案機種時，請於功能工具列中點選 **工具 (I)** > **PLC 機種設定 (S)**，之後於設定視窗中選擇欲變更的 PLC 型號後按下「確定」鍵即可。

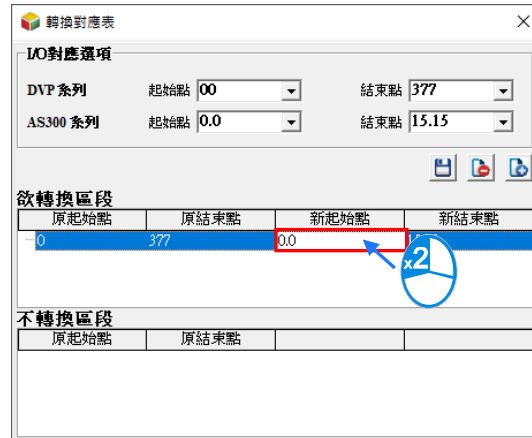


17.1.2 將 DVP 機種轉換為 AS 機種

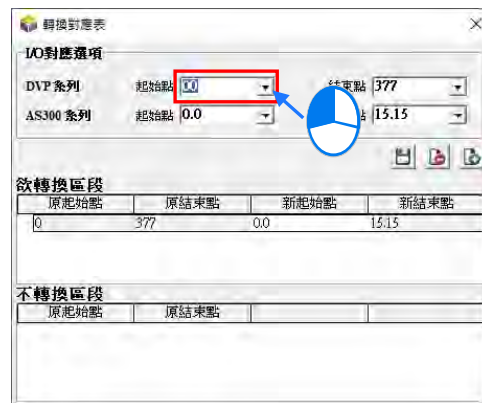
若要進行 DVP 機種轉換 AS 機種，請於功能工具列中點選 **工具 (I) > PLC 機種設定 (S)**，之後於設定視窗中選擇欲變更的 AS 機種後，可點擊選擇欄位旁邊的雙箭頭按鈕，進入「轉換對應表」，做 I/O 轉換的配置。



對應表中原起始點及原結束點，代表 DVP 機種之輸入 X、Y 的起始及結束點；新起始點及新結束點，則代表 AS 機種之輸入 X、Y 的起始及結束點，使用者需注意，DVP 系列 I/O 對 AS 系列 I/O 轉換範圍不一致時，會出現「警告：範圍不相符」；對欲設定的表格列，雙擊滑鼠左鍵，此時儲存/刪除按鈕將出現，即可用上下下拉式選單進行選擇或是直接選取欲變更欄位直接進行修改，如下圖。

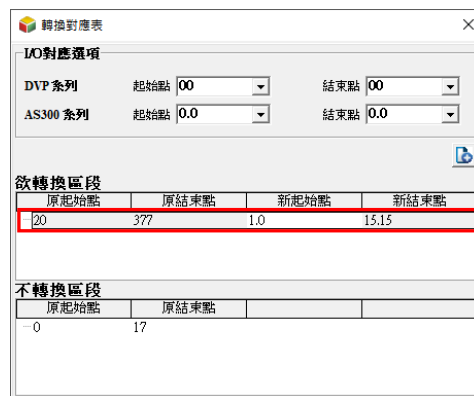
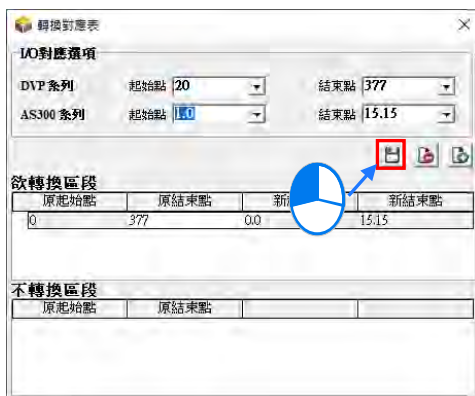


OR

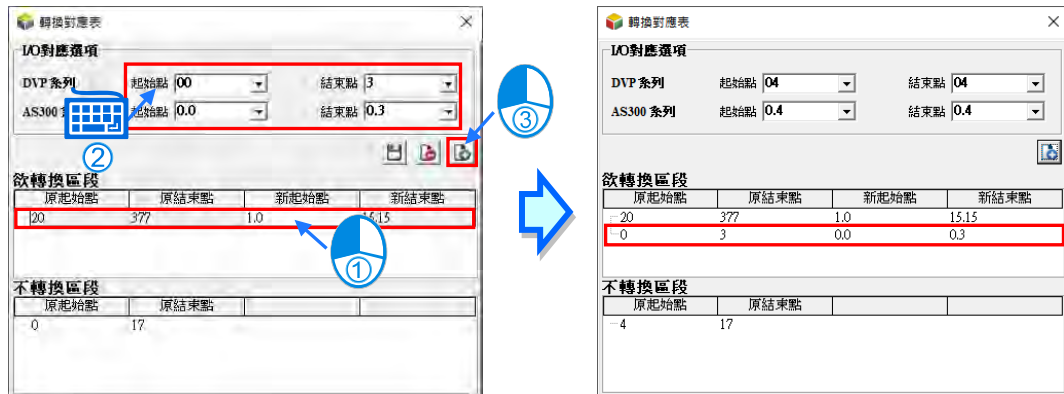


17

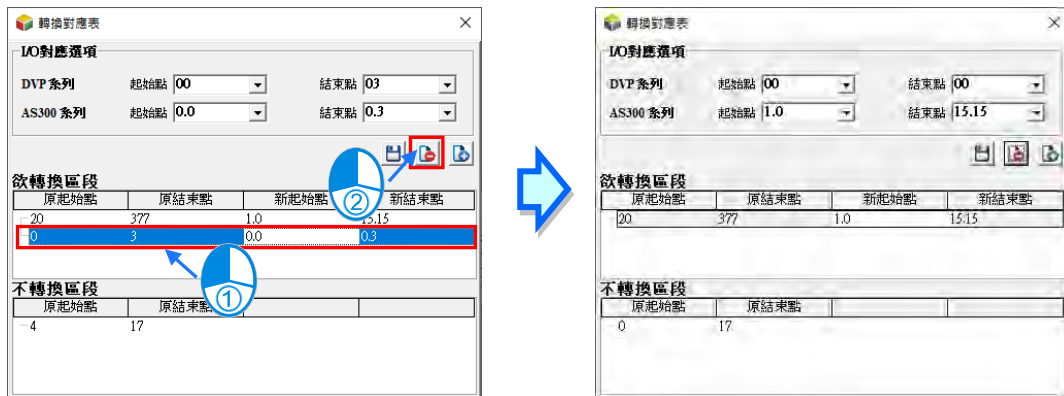
當設定完要變更的點位，點擊儲存按鈕，便可完成設定，此時會列出「欲轉換區段」以及「不轉換區段」。



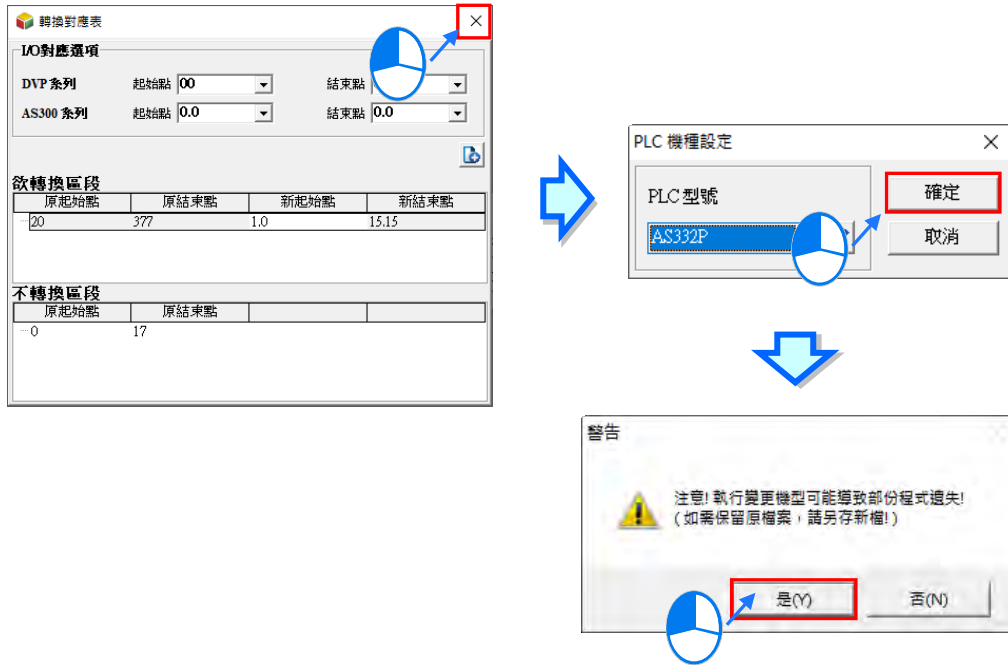
要新增一片段 I/O 轉換，可對表格列點擊滑鼠左鍵，輸入 I/O 欄位，再點擊新增按鈕，即可新增片段。



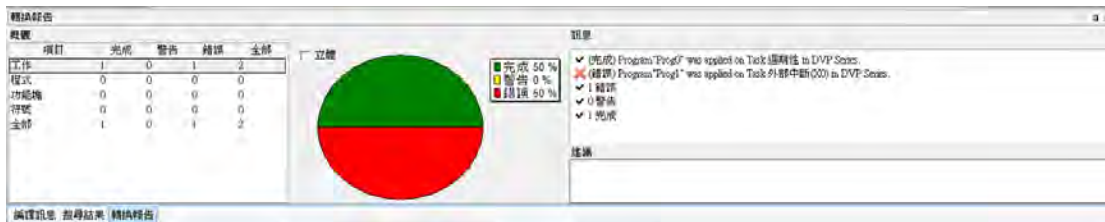
要刪除已設定的對應表格列，可對表格列雙擊滑鼠左鍵，再點擊刪除按鈕，便可完成刪除。



設定完轉換對應表，即可退出轉換對應表視窗，再對視窗，點擊「確定」，會出現詢問警告視窗，告知執行轉換變更機型可能導致部份程式遺失（此時若使用者欲保留原檔案，可先點擊「否」，取消轉換後，另存完檔後再進行轉換），是否要進行轉換，若選擇「是」，則會開始進行轉換工作。

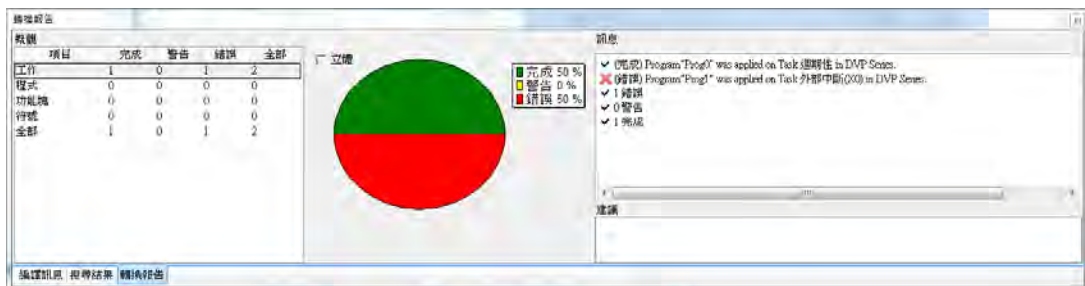
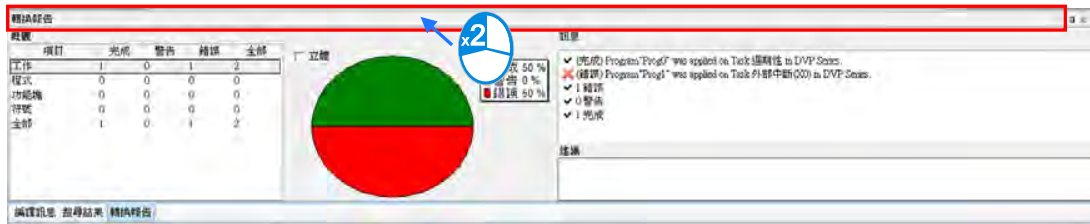


當完成轉換工作後，會出現「轉換報告」，如下圖。

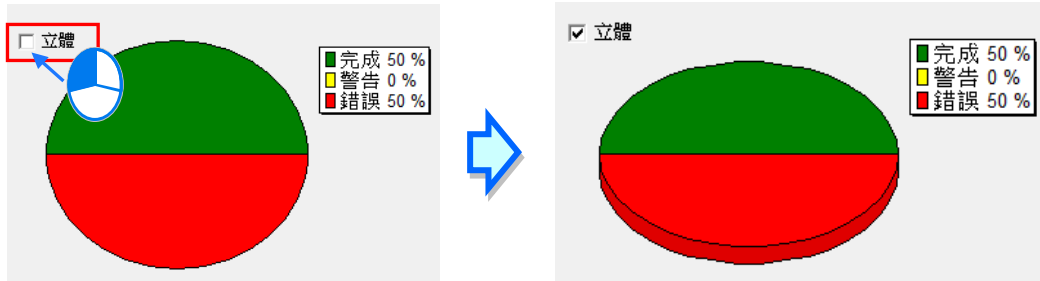


17

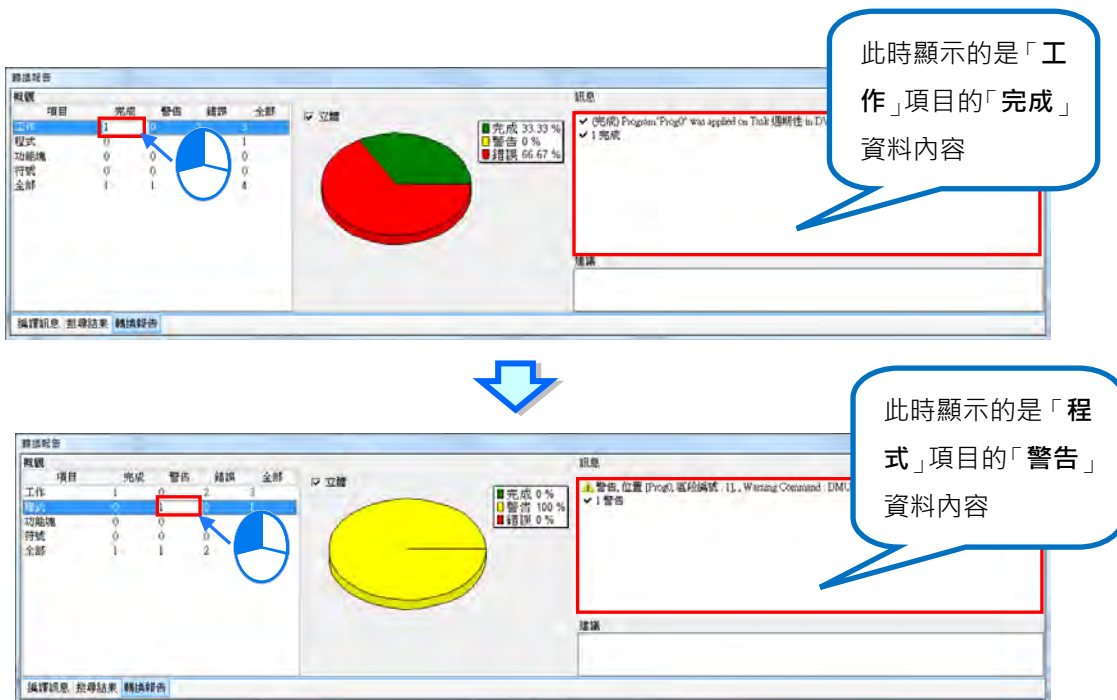
使用者可在「轉換報告」上方雙擊滑鼠左鍵，將視窗轉換為動態模式，再將視窗向右拉開並往上移，以方便觀察。



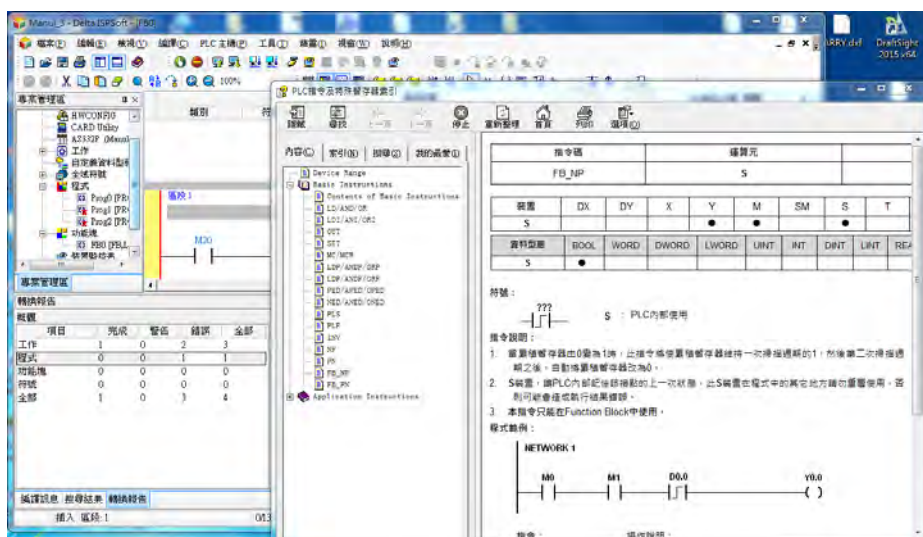
轉換報告左側「概觀」，會列表出轉換後的警告、錯誤及完成項目，內容分為工作、程式、功能塊、符號及全部，左下方的「全部」，計算每一行之加總，而右上方的「全部」，計算每一列之加總；中間「圓餅圖」會根據所選項目顯示錯誤、警告及完成的比例圖，綠色部分為「完成」、黃色部分為「警告」、紅色部分為「錯誤」，若勾選圓餅圖左上方的「立體」，則圓餅圖會呈現立體型式。



轉換報告右側「訊息」，會在可完全轉換的資料前端顯示（完成），若轉換後可能發生調適問題的資料前端則會顯示（警告），若轉換後發生無法轉換的資料前端則會顯示（錯誤），使用者也可在「概觀」表格欄點擊想要單獨檢視完成、警告或錯誤資料內容，則「訊息」框就只會顯示相對應內容，如下圖。

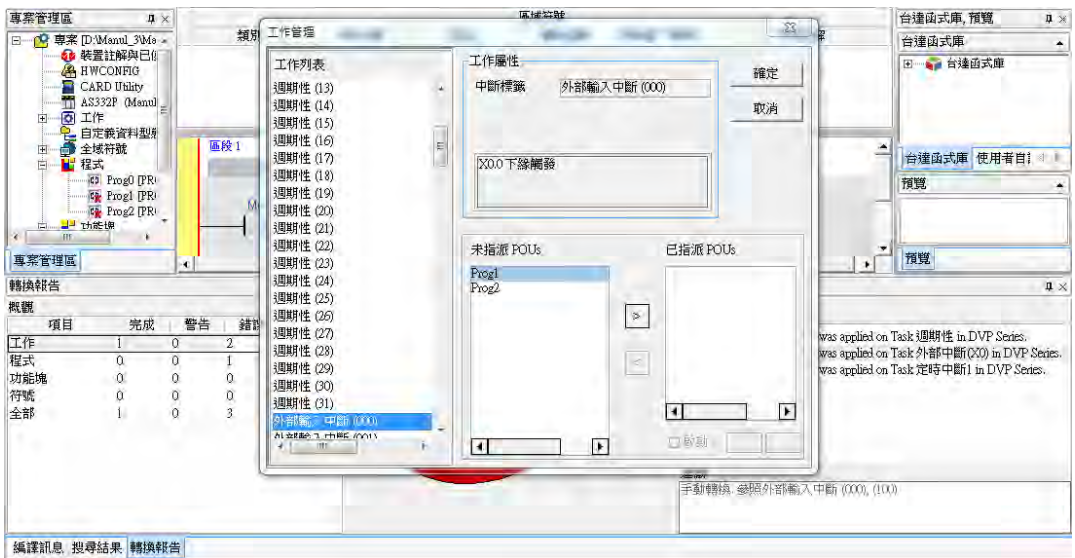
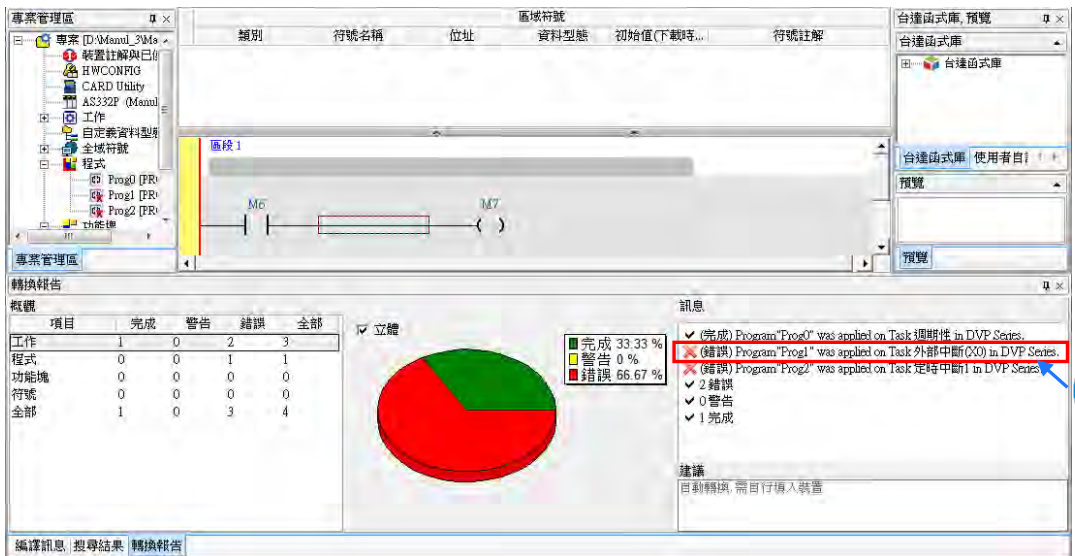


轉換報告右側下方「建議」，則會根據轉換後有出現「警告」或「錯誤」的訊息，給予調適修正的建議。針對轉換後，某些指令需做重新調適修改，會有相對應索引文件顯示，使用者可根據索引文件提示的內容，做適當的修飾。



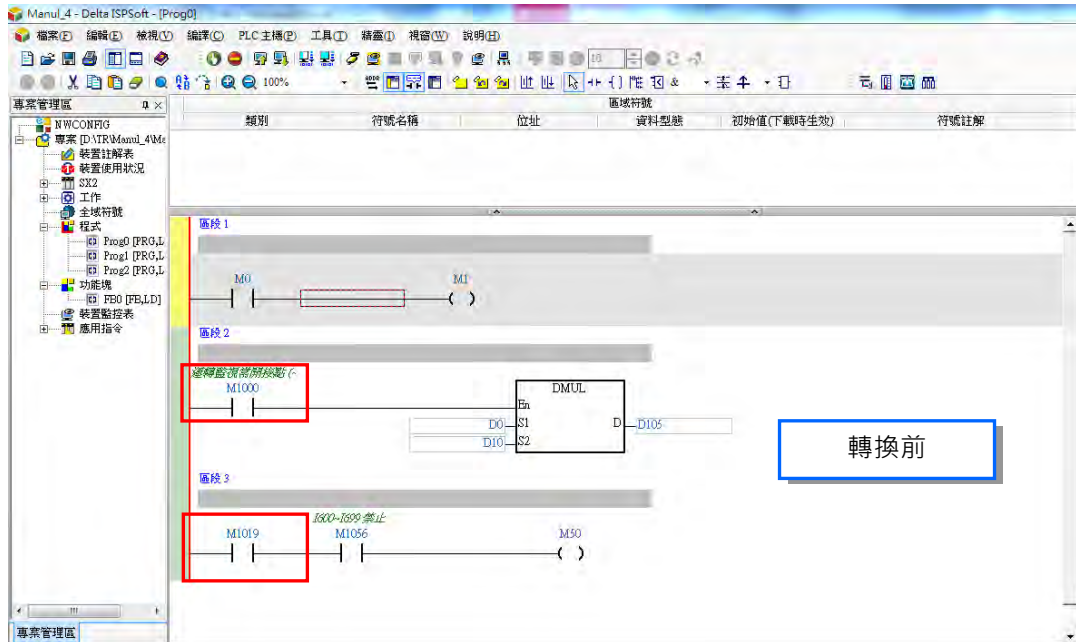
17

若轉換後，在「工作管理」部份有所變動，「建議」框中會顯示建議的解決方式，使用者可對「訊息」框中相對應的訊息雙擊滑鼠左鍵，此時會跳出「工作管理」視窗，以供使用者調整。



17

在做程式轉換時，DVP 系列機種之特殊輔助繼電器（特 M）或特殊資料暫存器（特 D），若 AS 系列機種有相對應的特殊內部電驛或特殊內部暫存器可轉換，則會自動生成；若沒有可轉換，則「建議」框中，會顯示調適的方法。



轉換報告

項目	完成	警告	錯誤	全部
工作	1	0	2	3
程式	1	1	3	5
功能塊	0	0	0	0
符號	0	0	0	0
全部	2	1	5	8

圖例：完成 20% (綠色), 警告 20% (黃色), 錯誤 60% (紅色)

訊息：

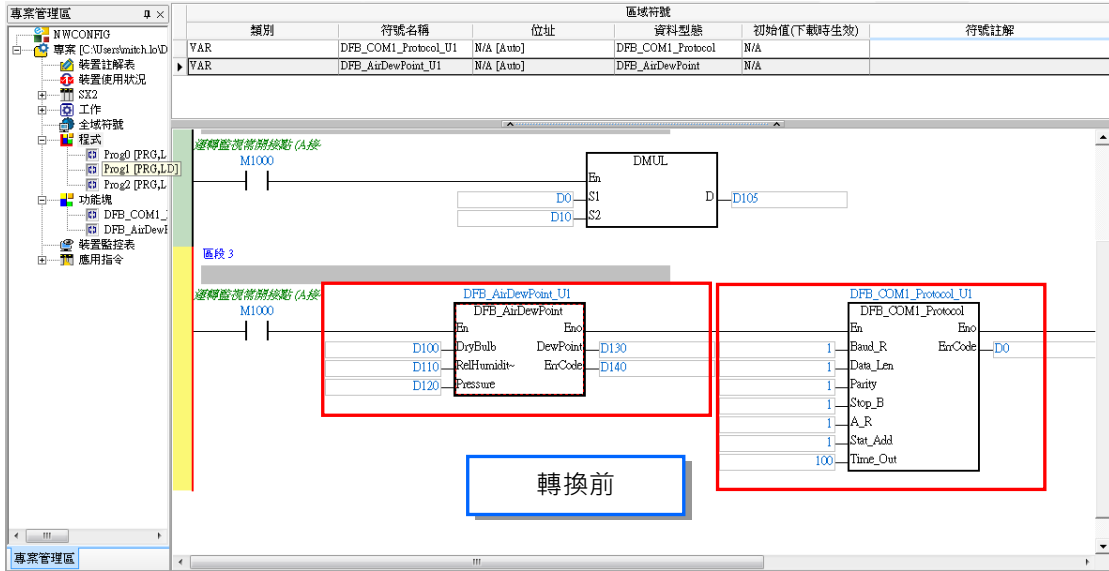
- 錯誤, 位置 (Pos0) 區段編號: 31, 錯誤指令: M1056
- 錯誤, 位置 (Pos0) 區段編號: 31, 錯誤指令: M1019
- 錯誤, 位置 (FD0) 區段編號: 11, 錯誤指令: ???> FB_NP
- 3 錯誤

建議：

- 手動轉換, 參照指令: USPD, PWD

17

在做程式轉換時，若 DVP 機種與 AS 系列機種都有支援之「台達功能塊」，則在轉換之後，會自動生成；若轉換之後，AS 系列機種沒有可對應的「台達功能塊」，功能塊則會出現叉號的錯誤提示，如下圖所示。



在做程式轉換時，若 DVP 機種使用的「符號名稱」與轉換後的 AS 系列機種關鍵字，例如：應用指令、裝置名稱，有所衝突時，轉換後會出現錯誤訊息。

類別	符號名稱	位址	資料型態	初始值(下載時生效)	符號註解
VAR	JOG	N/A [Auto]	WORD	N/A	
VAR	w1	N/A [Auto]	BOOL	N/A	



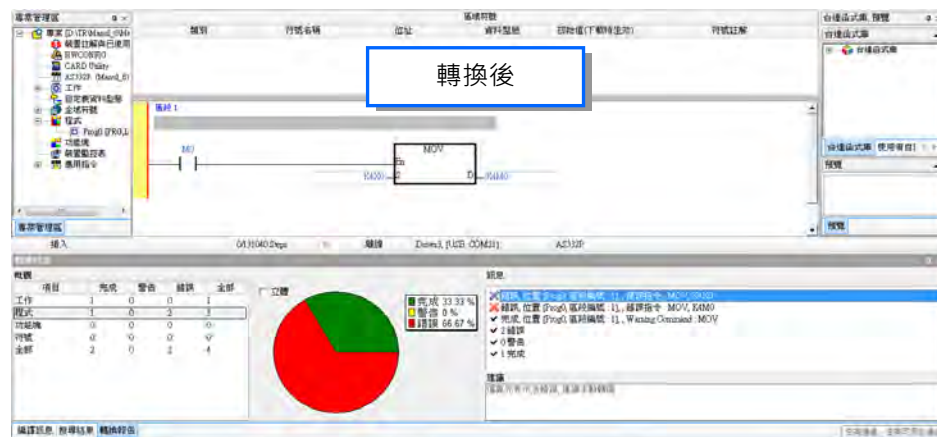
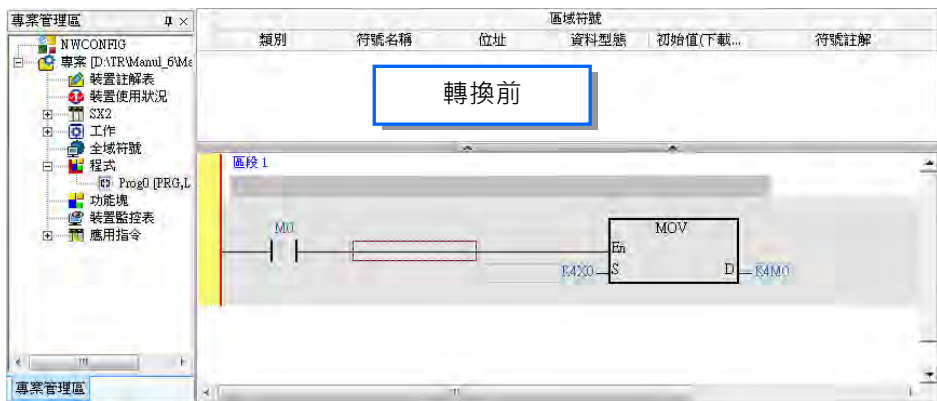
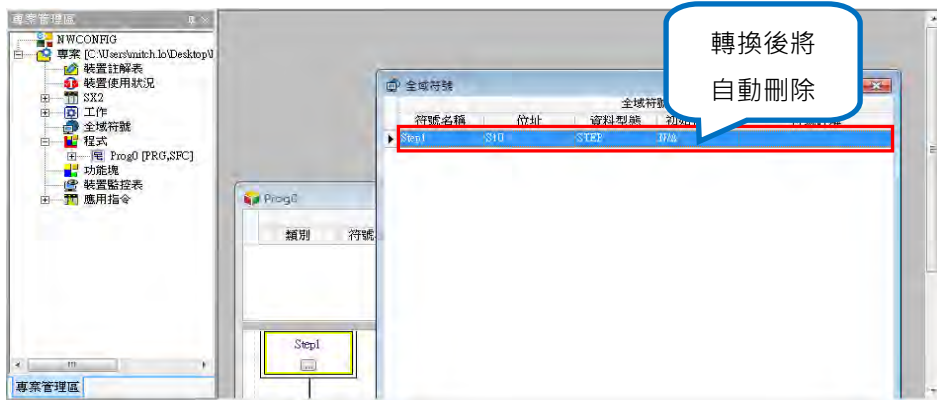
項目	完成	警告	錯誤	全部
工作	1	0	2	3
程式	0	1	0	1
功能塊	0	0	0	0
符號	0	0	2	2
全部	1	1	4	6

訊息

- ✖ 錯誤, 位置 [Prog0], 區域符號, 錯誤行號 (1), 符號名稱與系統保留字重覆, 錯誤指令: JOG
- ✖ 錯誤, 位置 [Prog0], 區域符號, 錯誤行號 (2), 符號名稱與系統保留字重覆, 錯誤指令: w1


建議

在做程式轉換時，DVP 系列機種轉 AS 系列機種轉換，只支援階梯圖 (LD) 及循序功能圖 (SFC) 語法轉換，若在 DVP 系列機種使用的符號型態是 AS 系列不支援的，轉換後將會被自動刪除，例如在符號表宣告一個資料型態為 STEP 的符號，在轉換後此符號將會被自動刪除；若在 DVP 系列機種有使用到 Kn 裝置表示法，在轉換為 AS 系列機種時會發生錯誤。




17

17.1.3 專案的下載與上傳

ISPSOft 的上下載功能，除了針對專案中的程式進行操作之外，同時亦可包含其他的功能或參數。欲進行下載時，請於功能工具列中點擊 **PLC 主機 (P)** > **PLC 程式傳輸** > **下載 (D)**，或直接於圖示工具列中點擊  圖示，接著再於出現的視窗中勾選欲下載的項目後按下「開始傳輸」即可。



而欲進行上傳時，於功能工具列中點選 **PLC 主機 (P)** > **PLC 程式傳輸** > **上傳 (U)**，或直接點擊圖示工具列的  圖示，之後再於出現的視窗中勾選欲上傳的項目後按下「開始傳輸」即可。



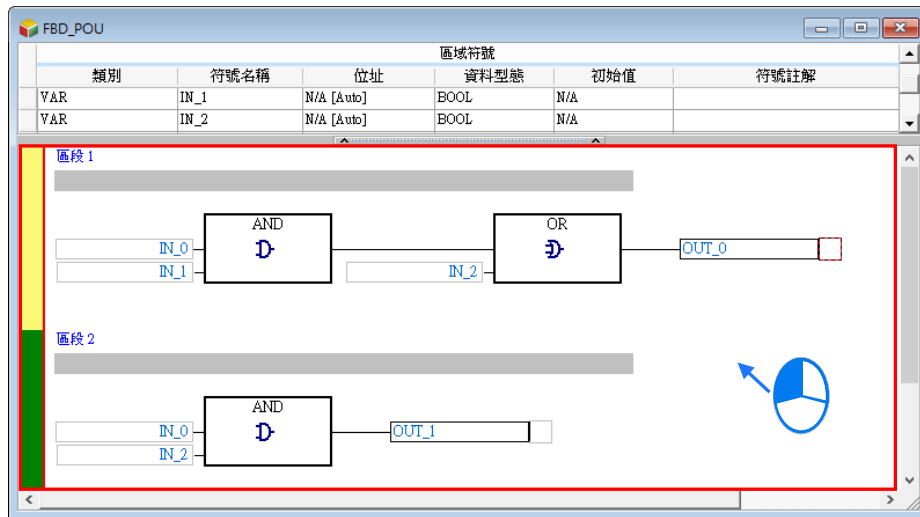
上下載視窗中，支援不同功能的各系列機種會有不同的上下載項目。關於各選項的說明請參考下表。

傳輸選項	說明	AH/AS		DVP		AS5xx DVPxxMC	
		上傳	下載	上傳	下載	上傳	下載
執行碼	程式編譯後的 PLC 執行碼	V	V	V	V	V	V
程式	使用者建立的專案程式	V	V	V	V	V	V
專案註解	專案程式的註解	V	V	V	V	V	V
POU 註解	POU 與 FB 的註解	V	V	V	V	V	V
區段註解	POU 與 FB 程式的區段註解	V	V	V	V	V	V
符號註解	全域符號及區域符號的符號註解	V	V	V	V	V	V
專案密碼與 PLC 密碼同步	將專案密碼同步到主機的 PLC 密碼		V		V		V
工作參數	工作參數設定					V	V
停電保持範圍	停電保持範圍的設定值				不支援 ES3		
停電保持符號表	符號表內的停電保持符號					V	V
CANopen 參數	CANopen Builder 參數設定值					V	V
符號初始值	將符號初始值寫入主機對應的記憶體位置		V		V		
萬年曆	將萬年曆設定的日期/時間寫入 PLC		V		V		V
定位規劃表	定位規劃表內容	只支援 AS	只支援 AS	只支援 ES3	只支援 ES3		
軸參數	軸參數內容	只支援 AHxx EMC	只支援 AHxx EMC			V	V
Ethernet/IP 符號表	Ethernet/IP 符號表內容		只支援 AS		只支援 ES3		
E-CAM	E-CAM 內容	只支援 AS 與 AHxx EMC	只支援 AS 與 AHxx EMC	只支援 ES3		V	V

17.1.4 LD、FBD 與 CFC 編輯環境的尋找取代與到功能

針對不同的編輯視窗，ISPSoft 所提供的尋找、取代與到功能皆會有所不同，而本節則先針對 LD、FBD 與 CFC 的編輯環境之功能操作加以說明。

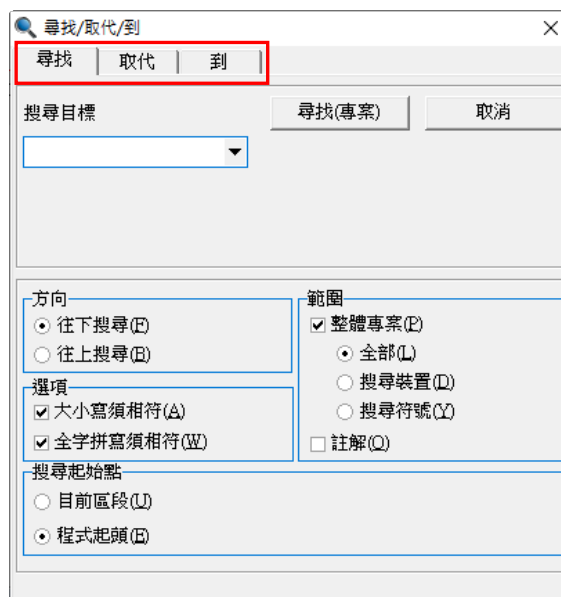
(1) 請開啟工作視窗，接著用滑鼠左鍵點選程式編輯區。



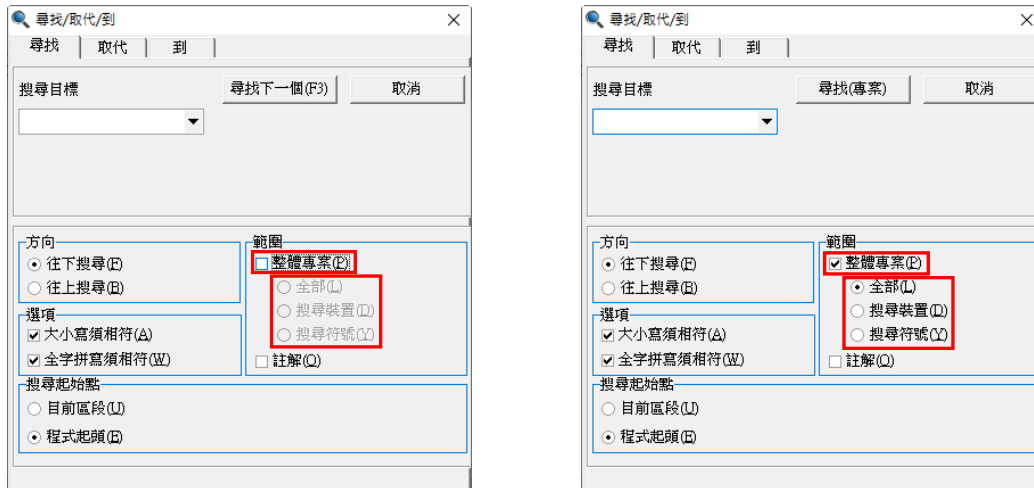
(2) 在功能工具列的「編輯 (E)」項目下選取尋找、取代與到，或選取圖示工具列中對應的圖示，皆可打開尋找/取代/到視窗，也可藉由切換視窗上方的標籤來更換操作的功能。



OR

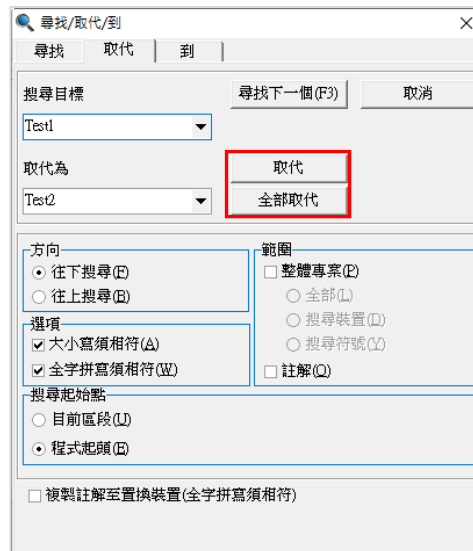


下圖為**尋找**功能的操作頁面，必須設定搜尋目標，並於下列欄位中選擇搜尋條件，而設定完畢後按下「**尋找下一個**」按鈕或鍵盤的【F3】鍵便可逐一進行搜尋的工作，且即使關閉搜尋視窗後，按下鍵盤的【F3】鍵仍可繼續搜尋最後一次搜尋的目標物。當範圍不勾選**整體專案**時，**尋找**的範圍僅限於目前正在編輯的視窗；當範圍勾選**整體專案**時，可勾選**全部**或**搜尋裝置**或**搜尋符號**，來進行分類搜尋，結果將列於訊息顯示區中。

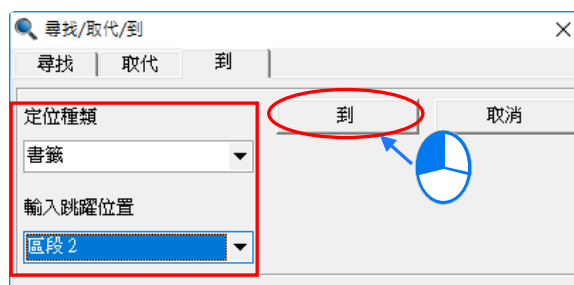


項目	說明
搜尋目標	直接輸入搜尋目標，或是點擊欄位右側的箭頭以下拉挑選搜尋過的字串。
方向	於此處選擇搜尋的方向。
選項	<p>大小寫須相符：比對字串須與搜尋目標的大小寫完全一致才符合搜尋條件。</p> <p>全字拼寫須相符：比對字串須與搜尋目標的內容完全一致才符合搜尋條件。若勾選全字拼寫須相符時可再勾選複製註解至置換裝置 (全字拼寫須相符)，將來源裝置註解複製到目標的裝置註解中。</p>
搜尋起始點	<p>目前區段：由目前點選的編輯位置開始搜尋。</p> <p>程式起頭：依據搜尋的方向，由程式的最前面或最後面開始搜尋整段程式。在 CFC 語言固定為此模式。</p>
範圍	<p>勾選搜尋時所包含的範圍。</p> <p>(勾選「整體專案」時，「全部」、「搜尋裝置」、「搜尋符號」才會致能。)</p> <p>全部：搜尋符合目標的所有裝置與符號。</p> <p>搜尋裝置：搜尋符合目標的所有裝置。</p> <p>搜尋符號：搜尋符合目標的所有符號。</p> <p>註解：搜尋符合目標的所有註解。</p>

下圖為**取代**功能的操作頁面，輸入搜尋目標，並於「**取代為**」欄位直接輸入欲替換的字串，按下「**取代**」按鈕便可將目前所找到的目標物字串變更為輸入的替換字串，而按下「**全部取代**」按鈕則會一次取代所有符合搜尋條件的目標物。



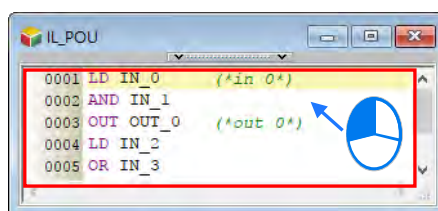
在 LD 與 FBD 語言中的「**到**」頁面可以執行快速移動的功能，於「**定位種類**」欄位下拉選擇欲採用的定位種類，共有**區段標籤**與**書籤**兩種，接著再於「**輸入跳躍位置**」欄位下拉選擇該類別的定位目標，完成後按下「**到**」按鈕便可引導使用者快速移動至程式中的指定位置。




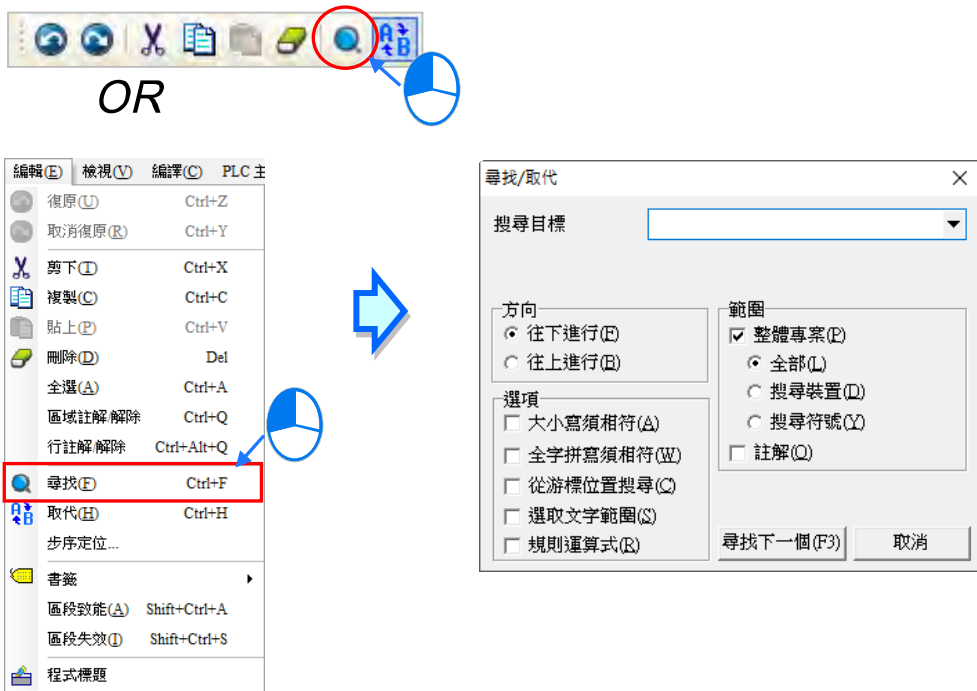
17.1.5 IL 與 ST 編輯環境的尋找取代功能


在 IL 與 ST 的環境中使用**尋找**與**取代**的功能。

(1) 請開啟工作視窗，接著用滑鼠左鍵點選程式編輯區。

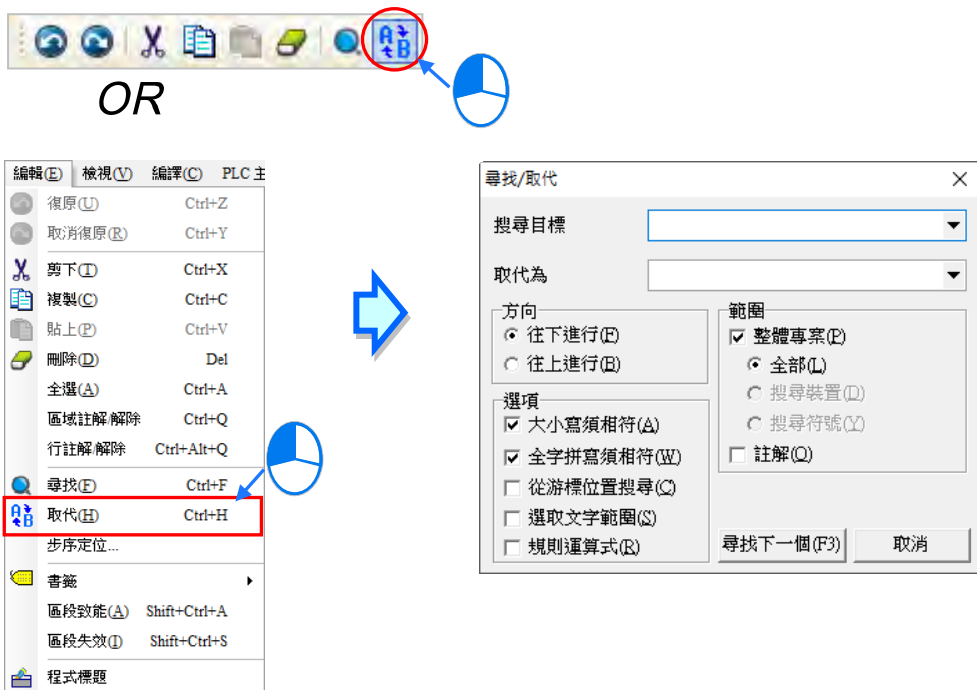


(2) 於功能工具列點選 **編輯 (E) > 尋找 (F)** 或點擊圖示工具列的  圖示即可開啟**尋找**操作視窗。

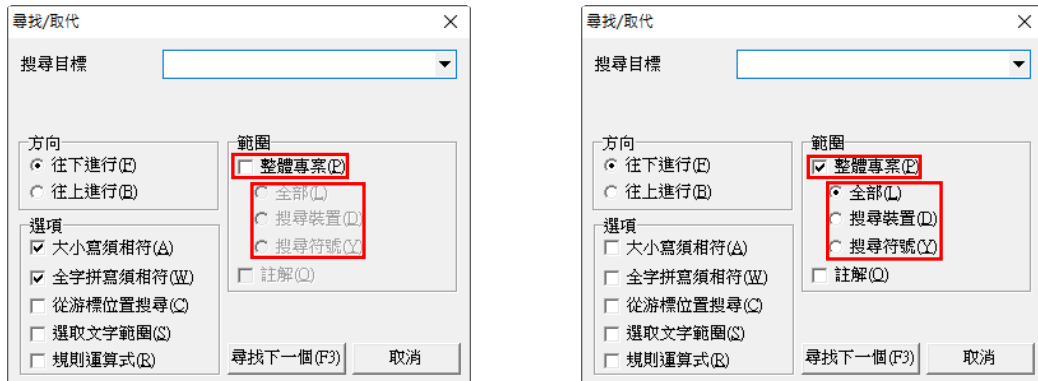


(3) 於功能工具列點選 **編輯 (E) > 取代 (H)** 或點擊圖示工具列的  圖示即可開啟**取代**操作視窗。

17



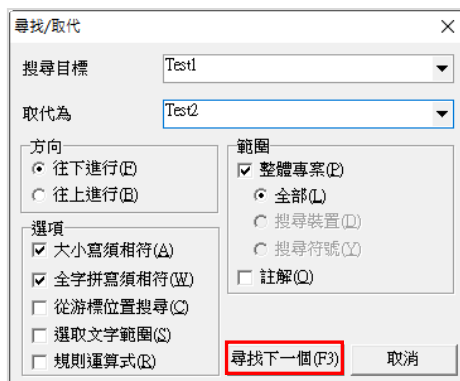
下圖為**尋找**功能的操作頁面，必須設定搜尋目標，並於下列欄位中選擇搜尋條件，而設定完畢後按下「**尋找下一個**」按鈕或鍵盤的【F3】鍵便可逐一進行搜尋的工作，且即使關閉搜尋視窗後，按下鍵盤的【F3】鍵仍可繼續搜尋最後一次搜尋的目標物。當範圍不勾選**整體專案**時，**尋找**的範圍僅限於目前正在編輯的視窗；當範圍勾選「**整體專案**」時，可勾選「**全部**」或「**搜尋裝置**」或「**搜尋符號**」，來進行分類搜尋，結果將列於訊息顯示區中。



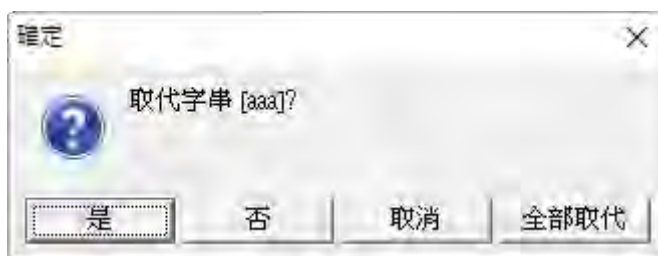
搜尋條件	說明
搜尋目標	直接輸入搜尋目標，或是點擊欄位右側的箭頭以下拉挑選搜尋過的字串。
方向	於此處選擇搜尋的方向。
選項	<p>大小寫須相符：比對字串須與搜尋目標的大小寫完全一致才符合搜尋條件。</p> <p>全字拼寫須相符：比對字串須與搜尋目標的內容完全一致才符合搜尋條件。若勾選全字拼寫須相符時可再勾選複製註解至置換裝置(全字拼寫須相符)，將來源裝置註解複製到目標的裝置註解中。</p> <p>從游標位置搜尋：只由目前游標的位置開始搜尋(未勾選此項時，系統會搜尋整個程式)。</p> <p>選取文字範圍：只由框選的文字範圍開始搜尋(未勾選此項時，系統會搜尋整個程式)。</p> <p>規則運算式：勾選此項後，「搜尋目標」欄位便可套用正規表示式的語法。</p>
範圍	<p>搜尋時所包含的範圍。</p> <p>(搜尋分頁勾選「整體專案」時，「全部」、「搜尋裝置」、「搜尋符號」才會致能。)</p> <p>全部：搜尋符合目標的所有字串。</p> <p>搜尋裝置：搜尋符合目標的所有裝置。</p> <p>搜尋符號：搜尋符合目標的所有符號。</p> <p>註解：搜尋符合目標的所有註解。</p>

* 正規表示式(Regular Expression)為一種標準語法，例如使用「*」號代表萬用字元，相關細節請自行查詢相關技術文件。

下圖為**取代**功能的操作頁面，輸入搜尋目標，並於下列欄位中選擇搜尋條件；而執行取代操作時輸入搜尋目標，並於「**取代為**」欄位直接輸入欲替換的字串，設定完畢後按下「**尋找下一個 (F3)**」按鈕便可取代操作，在取代頁面時勾選「**整體專案**」只有「**全部**」會致能。



當執行取代操作且找到符合條件的目標物時，畫面將會出現詢問視窗，而視窗中的按鈕功能如下。

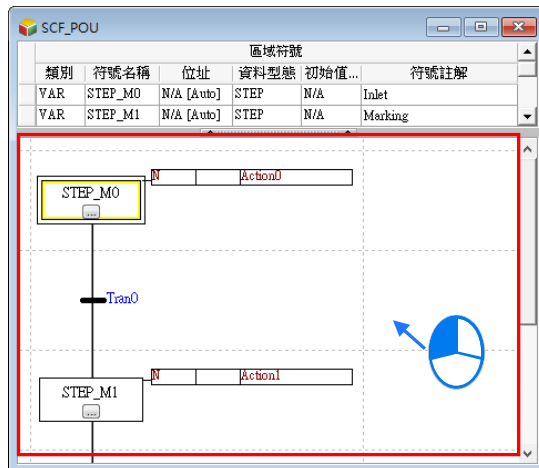


按鈕	功能說明
是	將目前搜尋到的字串變更為取代字串，並繼續搜尋下一個目標物。
否	不變更目前搜尋到的字串，並繼續搜尋下一個目標物。
取消	停止搜尋的操作並關閉此視窗
全部取代	將所有符合搜尋條件的目標物字串全部變更為取代字串

17.1.6 SFC 編輯環境的尋找取代功能

在 SFC 的編輯環境中使用尋找取代功能。

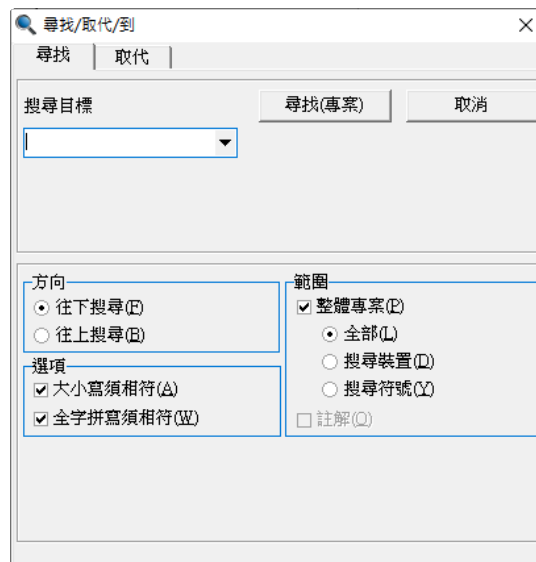
- (1) 請開啟工作視窗，接著用滑鼠左鍵點選程式編輯區。



- (2) 在功能工具列的「編輯 (E)」項目下選取尋找、取代，或選取圖示工具列中對應的圖示，皆可打開尋找/取代/到視窗，也可藉由切換視窗上方的標籤來更換操作的功能。



OR



17

下圖為**取代**功能的操作頁面，輸入搜尋目標，並於「**取代為**」欄位直接輸入欲替換的字串，按下「**取代**」按鈕便可將目前所找到的目標物字串變更為輸入的替換字串，而按下「**全部取代**」按鈕則會一次取代所有符合搜尋條件的目標物；若勾選「**整體專案**」並點選「**取代 (專案)**」將會把專案內所有字串進行取代。

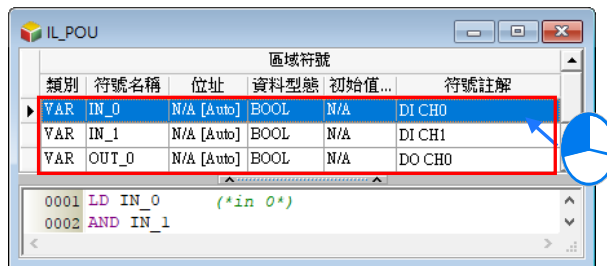



搜尋條件	說明
搜尋目標	直接輸入搜尋目標，或是點擊欄位右側的箭頭以下拉挑選搜尋過的字串。
方向	於此處選擇搜尋的方向。
選項	<p>大小寫須相符：比對字串須與搜尋目標的大小寫完全一致才符合搜尋條件。</p> <p>全字拼寫須相符：比對字串須與搜尋目標的內容完全一致才符合搜尋條件。若勾選全字拼寫須相符時可再勾選複製註解至置換裝置 (全字拼寫須相符)，將來源裝置註解複製到目標的裝置註解中。</p>
範圍	<p>搜尋時所包含的範圍。</p> <p>(搜尋分頁勾選「整體專案」時，「全部」、「搜尋裝置」、「搜尋符號」才會致能。)</p> <p>全部：搜尋符合目標的所有字串。</p> <p>搜尋裝置：搜尋符合目標的所有裝置。</p> <p>搜尋符號：搜尋符合目標的所有符號。</p> <p>註解：搜尋符合目標的所有註解。</p>

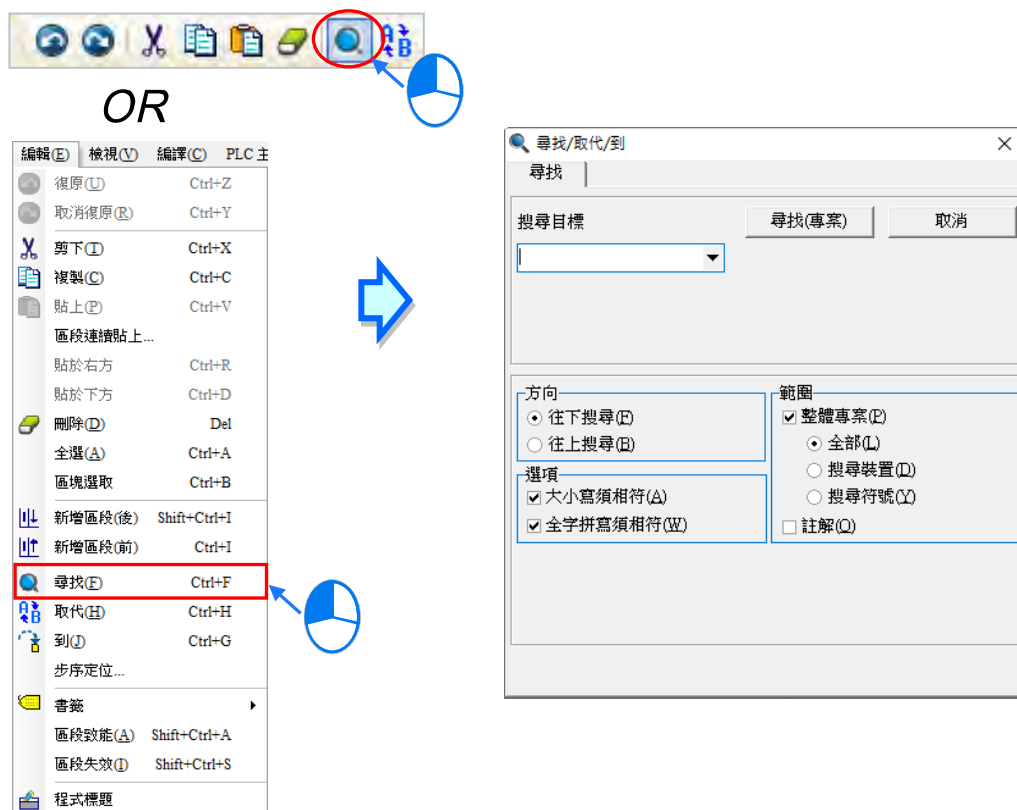
17.1.7 符號表的尋找功能

在區域符號表與全域符號表使用尋找的功能，只能對目前正在編輯的視窗進行操作。

(1) 開啟欲進行操作的編輯視窗後，在符號表上點擊一下滑鼠左鍵。



(2) 於功能工具列點選 編輯 (E) > 尋找 (F) 或點擊圖示工具列中的  圖示即可開啟操作視窗。

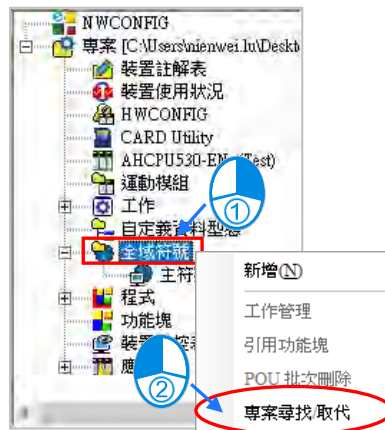


在範圍選取整體專案後按下「尋找(專案)」按鈕即可快速在整體專案搜尋到所需要的內容，也可以不選取整體專案後按下「尋找下一個(F3)」按鈕或鍵盤的【F3】鍵便可逐一進行搜尋的工作，且即使關閉搜尋視窗後，按下鍵盤的【F3】鍵仍可繼續搜尋最後一次搜尋的目標物。

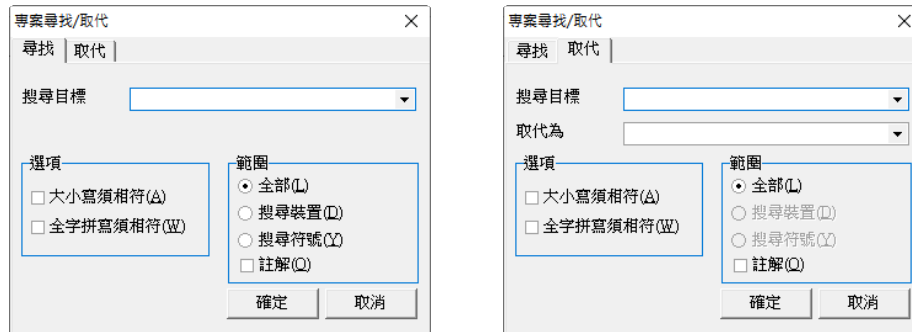
搜尋條件	說明
搜尋目標	直接輸入搜尋目標，或是點擊欄位右側的箭頭以下拉挑選搜尋過的字串。
方向	於此處選擇搜尋的方向。
選項	<p>大小寫須相符：比對字串須與搜尋目標的大小寫完全一致才符合搜尋條件。</p> <p>全字拼寫須相符：比對字串須與搜尋目標的內容完全一致才符合搜尋條件。若勾選全字拼寫須相符時可再勾選複製註解至置換裝置(全字拼寫須相符)，將來源裝置註解複製到目標的裝置註解中。</p>
範圍	<p>搜尋時所包含的範圍。</p> <p>(搜尋分頁勾選「整體專案」時，「全部」、「搜尋裝置」、「搜尋符號」才會致能。)</p> <p>全部：搜尋符合目標的所有字串。</p> <p>搜尋裝置：搜尋符合目標的所有裝置。</p> <p>搜尋符號：搜尋符合目標的所有符號。</p> <p>註解：搜尋符合目標的所有註解。</p>

17.1.8 專案尋找與取代

在專案管理區的**全域符號表**、**程式**或**功能塊**項目上按下滑鼠右鍵，接著於快捷選單中點選「**專案尋找/取代**」，可打開此功能，用於尋找/取代專案中的某個裝置或符號。

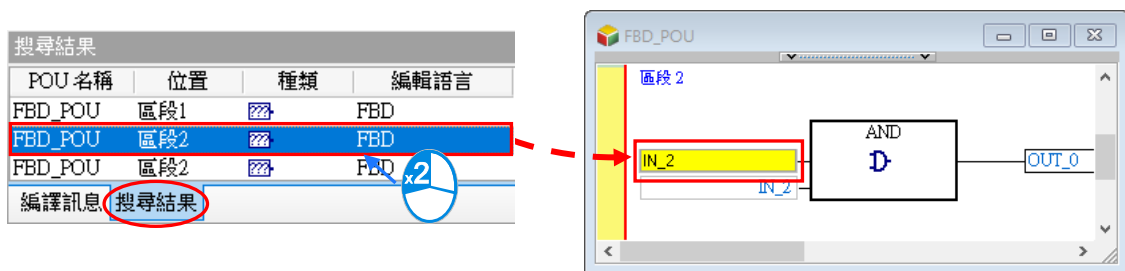


下圖為**尋找**與**取代**功能的操作頁面，可藉由視窗上方的標籤來進行切換，必須設定搜尋目標，並於下列欄位中選擇搜尋條件，且只有**尋找**可以使用「**搜尋裝置**」與「**搜尋符號**」選項，設定完畢後，若在「**尋找**」的頁面中按下「**確定**」鍵，接著系統便會開始進行搜尋的工作；而若在「**取代**」頁面中按下「**確定**」鍵，則系統便會在完成搜尋後，自動將符合條件的目標物字串全部變更為「**取代為**」欄位中的字串，且此動作將無法還原。



搜尋條件	說明
範圍	搜尋裝置 ：搜尋符合目標的所有裝置。 搜尋符號 ：搜尋符合目標的所有符號。 註解 ：搜尋符合目標的所有註解。
選項	大小寫須相符 ：比對字串須與搜尋目標的大小寫完全一致才符合搜尋條件。 全字拼寫須相符 ：比對字串須與搜尋目標的內容完全一致才符合搜尋條件。

無論是**尋找**或**取代**的操作，所有搜尋的結果將會顯示於 ISPSOft 視窗下方的**訊息顯示區**，且於某個搜尋結果的物件上雙擊滑鼠左鍵後，系統便會自動引導使用者移動至該目標物的所在位置。

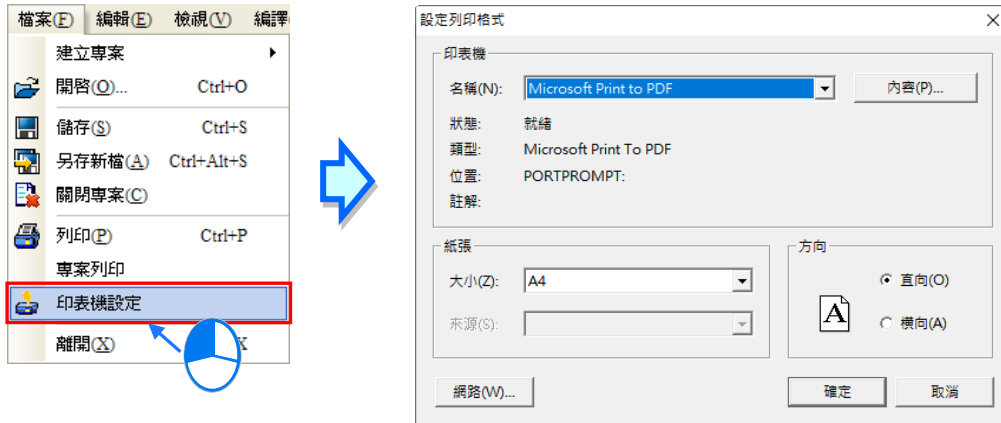


17.1.9 列印功能

ISPSOft 對於編輯工作區中的內容亦提供了列印的功能，相關說明如下。

● 印表機設定

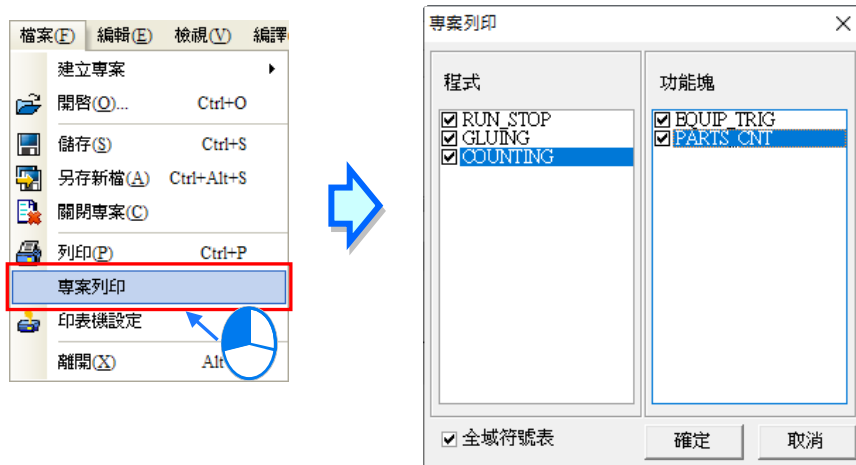
於功能工具列中點選 **檔案 (F) > 印表機設定**，之後便可開啟印表機的設定視窗，在列印文件之前可先於此處選擇要使用的列表機與列印格式。



● 專案列印

於功能工具列中點選 **檔案 (F) > 專案列印**，之後便可勾選欲列印的項目，可選的項目包含**程式**、**功能塊**與**全域符號表**，而完成後按下「確定」鍵即可依據目前的印表機設定將所選的項目列印出來。

17




下圖為列印的結果，列印內容會包含區域符號表，而註解部分則會依據當時視窗的註解顯示狀態。

宣告類別	符號	位址	型態	初始值	符號註解
VAR	ET_INS_1	N/A [Auto]	EQUIP_TRIG	N/A	
VAR	ET_INS_2	N/A [Auto]	EQUIP_TRIG	N/A	

區段 1									
RUNNING	<table border="1"> <tr> <td>ET_INS_1</td> <td></td> </tr> <tr> <td>EQUIP_TRIG</td> <td></td> </tr> <tr> <td>En</td> <td>Eno</td> </tr> <tr> <td>InP_SNR_1</td> <td>SNR_IN TRIG TRIG_1</td> </tr> </table>	ET_INS_1		EQUIP_TRIG		En	Eno	InP_SNR_1	SNR_IN TRIG TRIG_1
ET_INS_1									
EQUIP_TRIG									
En	Eno								
InP_SNR_1	SNR_IN TRIG TRIG_1								

● 視窗列印

若只要列印目前正在編輯的視窗內容時，請於功能工具列中點選 **檔案 (F) > 列印 (P)**，或直接點擊圖示工具列中的  圖示，之後便會開啟列印範圍的選擇視窗，而根據編輯視窗內容的不同，選擇之後所列印的結果也會不同，請參考下表。



編輯視窗	選擇全部範圍	選擇部分範圍
全域符號表	列印完整的全域符號表	全域符號表無法選擇列印部分範圍
LD 與 FBD 程式	列印完整的程式與區域符號表	列印完整區域符號表及選擇範圍內的程式 (所設定的 起始 與 結束 編號代表區段編號)
IL 與 ST 程式	列印完整的程式與區域符號表	列印完整區域符號表及選擇範圍內的程式 (所設定的 起始 與 結束 編號代表程式行號)
SFC 程式	列印完整的區域符號與 SFC 圖面 (不包含 Actions 與 Transitions)	SFC 程式無法選擇列印部分範圍
CFC 程式	列印完整的程式與區域符號表	CFC 程式無法選擇列印部分範圍

*.Actions 與 Transitions 的內容仍必須在開啟其視窗之後才可進行列印。

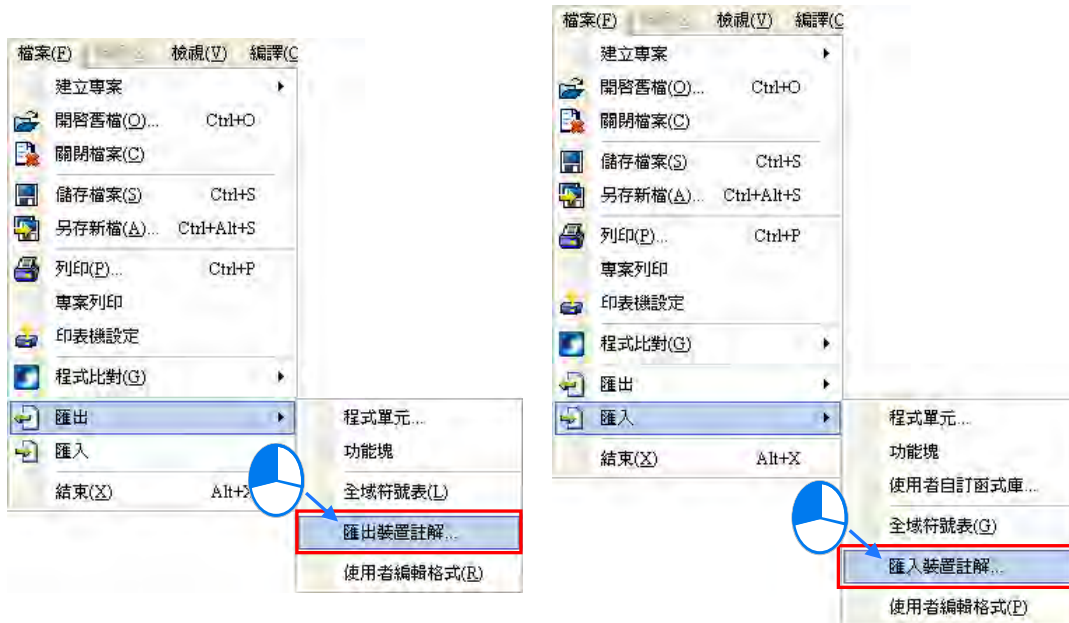
設定完成後按下「預覽列印」即可開啟列印管理工具，而關於**列印管理工具**的說明則請參考 附錄 C。



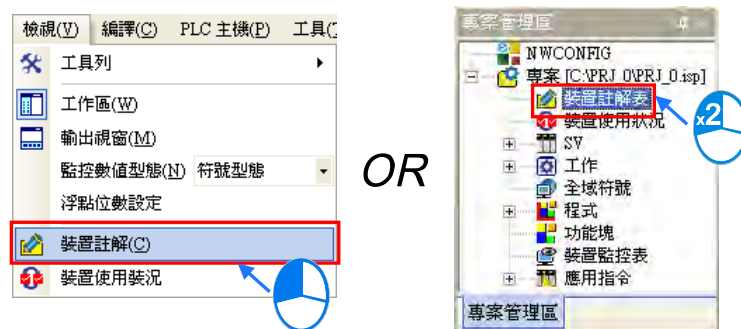
17.2 裝置與暫存器管理

17.2.1 DVP 機種的裝置註解表

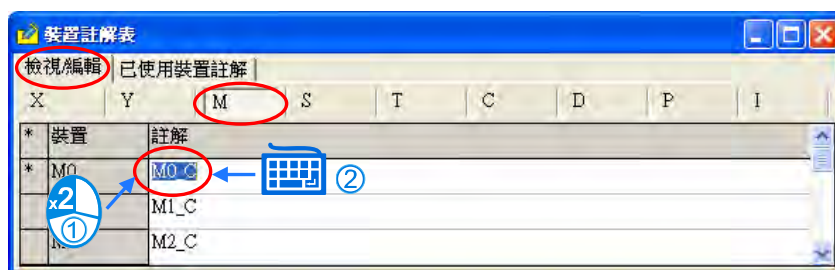
進行 DVP 機種的專案開發時，於功能工具列點擊 **檔案 (F) > 匯出 > 匯出裝置註解...**，之後便可將此專案中的裝置註解匯出；而要重新匯入時，只要點選 **檔案 (F) > 匯入 > 匯入裝置註解...** 即可。



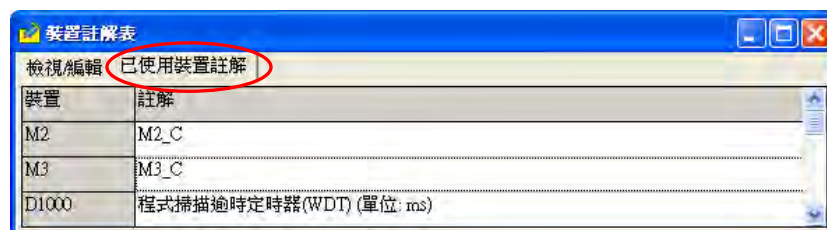
當要批次對裝置註解進行管理時，請於功能工具列中點選 **檢視 (V) > 裝置註解 (C)**，或直接於專案管理區的「裝置註解表」項目上雙擊滑鼠左鍵，之後系統便會自動開啟裝置註解的管理視窗。



「檢視/編輯」頁面會列出所有裝置的註解。先於頁面上方選擇裝置類別後，頁面中便會列出該裝置範圍的所有註解狀況，而當裝置名稱的左側出現「*」符號時，代表該裝置已在程式當中被使用。當要編輯註解時，只要在欲編輯的欄位上雙擊滑鼠左鍵，之後再以鍵盤輸入註解內容即可。

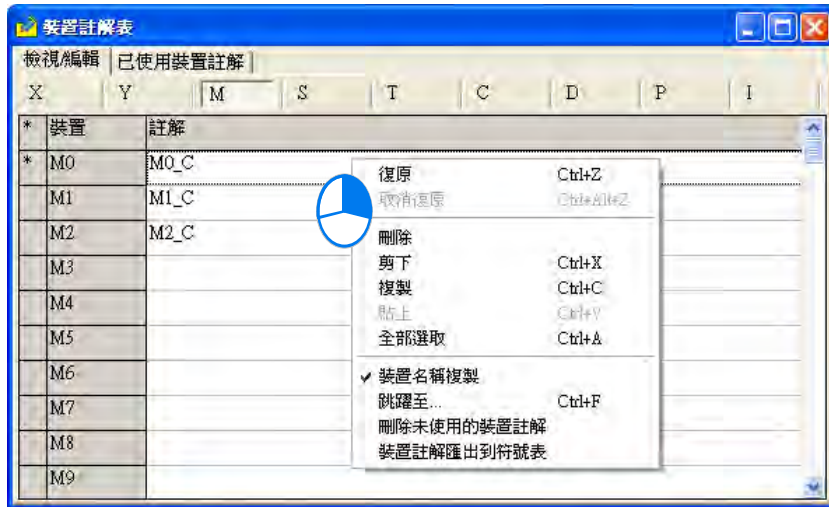


「已使用裝置註解」頁面則不分裝置類別，全部有設定裝置註解的裝置與其註解都會全部列出，且於此頁面中同樣可對各裝置的註解進行編輯，其方法則與前述「檢視/編輯」頁面中的編輯方式相同。



- *1. 裝置註解與符號註解是彼此獨立的，而此表顯示的僅會是裝置註解的部份，變數符號的註解並不會顯示在清單當中。
- *2. 特殊繼電器與特殊暫存器本身已具有預設的註解，但仍可重新編輯，不過重新編輯的註解只限在這個專案中才有效。

於列表中按下右鍵後，彈出的快捷選單中除了基本的 Windows 操作之外，亦包含許多專屬的功能。



● 跳躍至

點擊此功能後畫面便會出現一視窗，於「裝置名稱」欄位輸入欲檢視的裝置名稱後按下「確定」鍵，之後列表的顯示便會立即調整至可檢視該裝置的範圍。

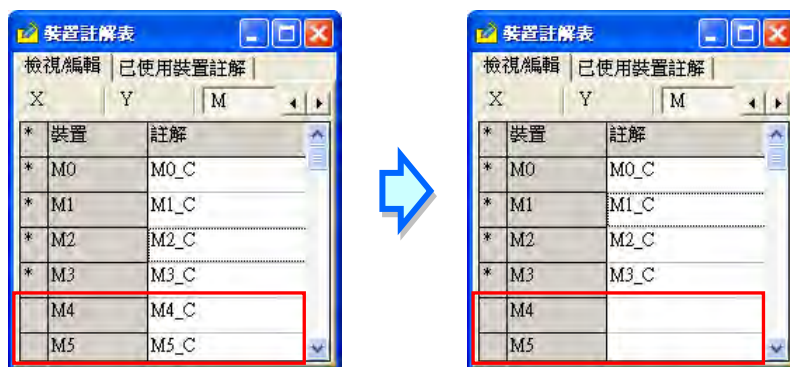
輸入裝置名稱時，裝置的類別必須與目前的頁面一致，否則便會找不到該裝置；而於「已使用裝置註解」頁面中使用此功能時，輸入的裝置名稱也必須確實位於該清單之中才可進行操作。



17

● 刪除未使用的裝置註解

點擊此項後，對於未使用於程式當中的裝置，系統會自動清除這些裝置的註解。



● 裝置名稱複製

當選取註解表中的項目，並執行複製或剪下的功能後，該項目便會暫存於 Windows 的剪貼簿中，並且可於其他的軟體中被貼上，如 EXCEL、記事本...；而在操作前，若「裝置名稱複製」的項目有被勾選時，複製或剪下的內容便會自動包含裝置的名稱，請參考下列兩個 EXCEL 表的比較。



裝置	註解
* M0	M0_C
M1	M1_C
M2	M2_C
M3	M3_C
M4	

	A	B	C
1	M0	M0_C	
2	M1	M1_C	
3	M2	M2_C	
4	M3	M3_C	
5			

	A	B
1	M0_C	
2	M1_C	
3	M2_C	
4	M3_C	
5		

● 裝置註解匯出到符號表

當已為變數符號配置裝置位址時，點擊此項後，裝置註解便可匯出至對應該裝置位址的符號註解。



裝置	註解
* M0	M0_C
M1	M1_C
M2	M2_C

類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	VAR_0	M0	BOOL	FALSE	VAR_0C
VAR	VAR_1	N/A [Auto]	BOOL	FALSE	VAR_1C

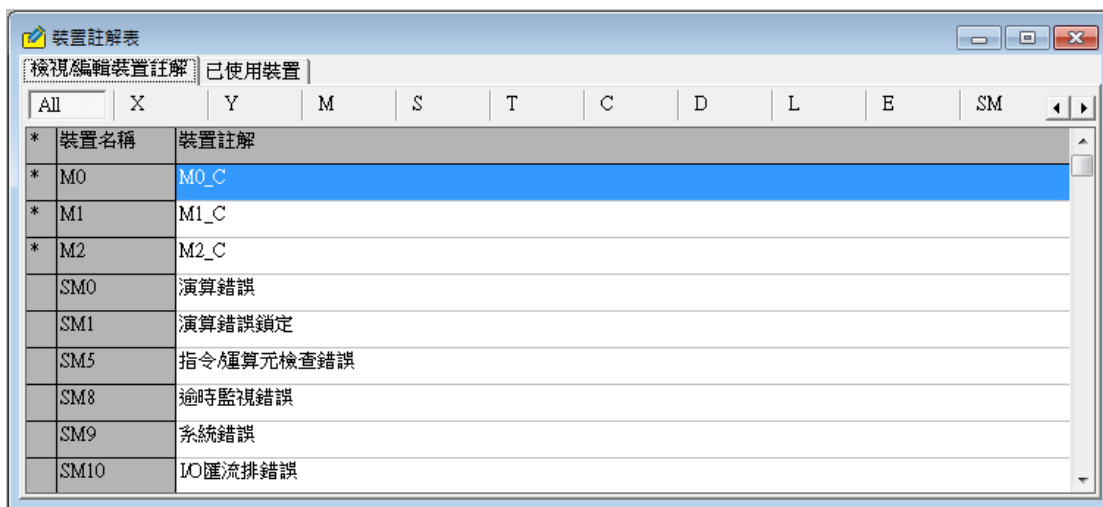
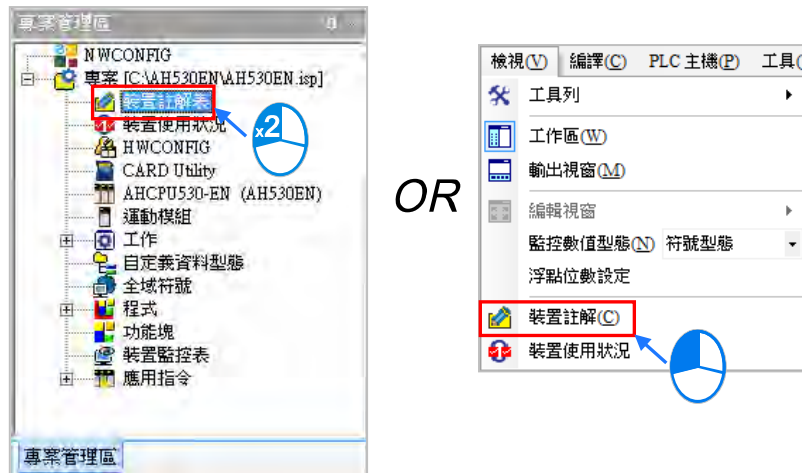


類別	符號名稱	位址	資料型態	初始值	符號註解
VAR	VAR_0	M0	BOOL	FALSE	M0_C
VAR	VAR_1	N/A [Auto]	BOOL	FALSE	VAR_1C

17.2.2 AH/AS 系列機種的裝置注解表

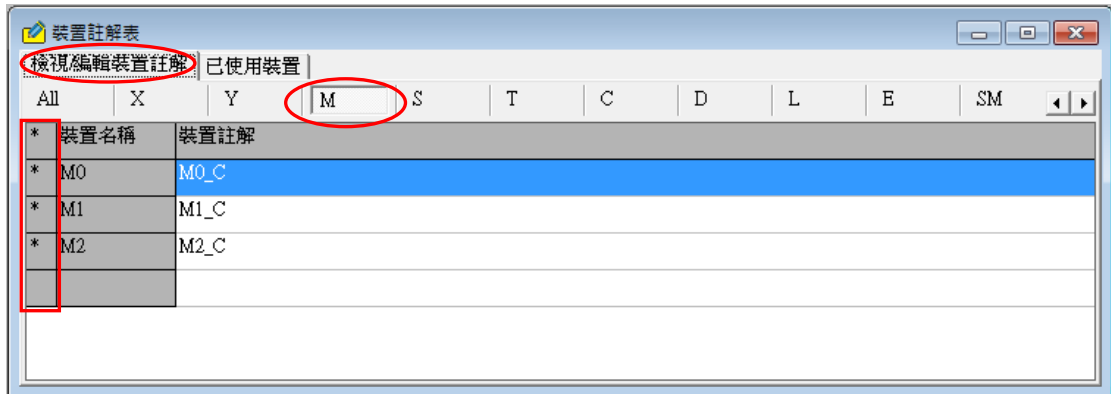
AH/AS 系列機種本身亦提供裝置管理的功能，於功能工具列中點選 **檢視 (V)** > **裝置注解 (C)**，或於專案管理區的「裝置注解表」項目上雙擊滑鼠左鍵，之後便可開啟管理視窗。

管理視窗總共包含兩個頁面，且藉由視窗上方的標籤便可進行切換。但若開啟此視窗後曾經變更程式內容或修改注解時，請先關閉視窗並重新編譯後再開啟視窗，如此系統才可取得最新的資訊。

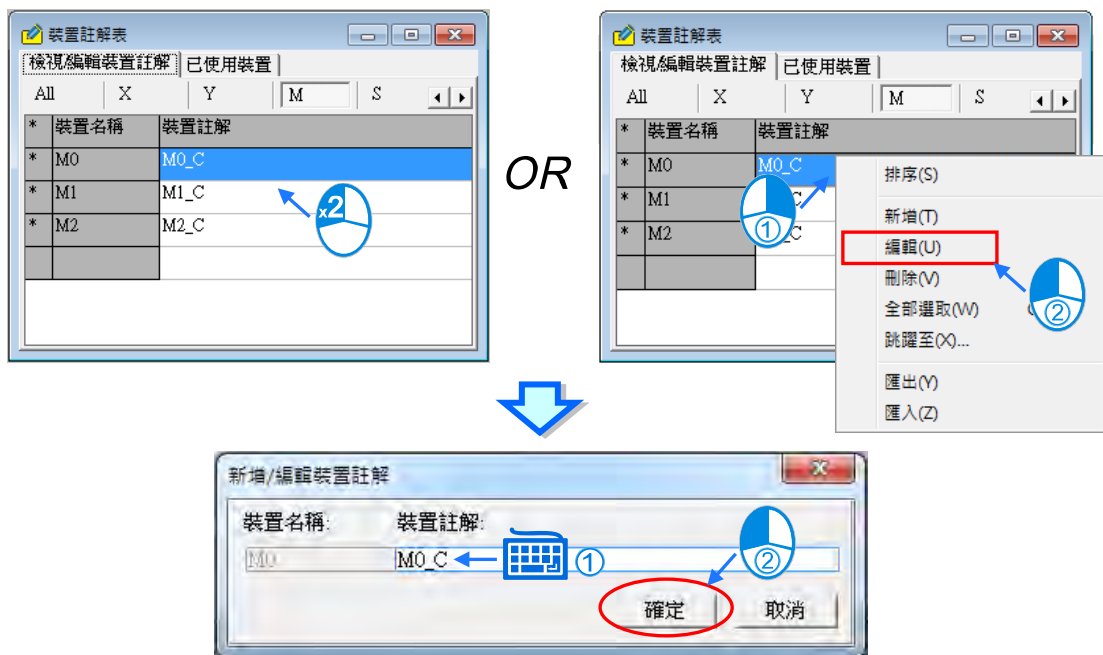


17

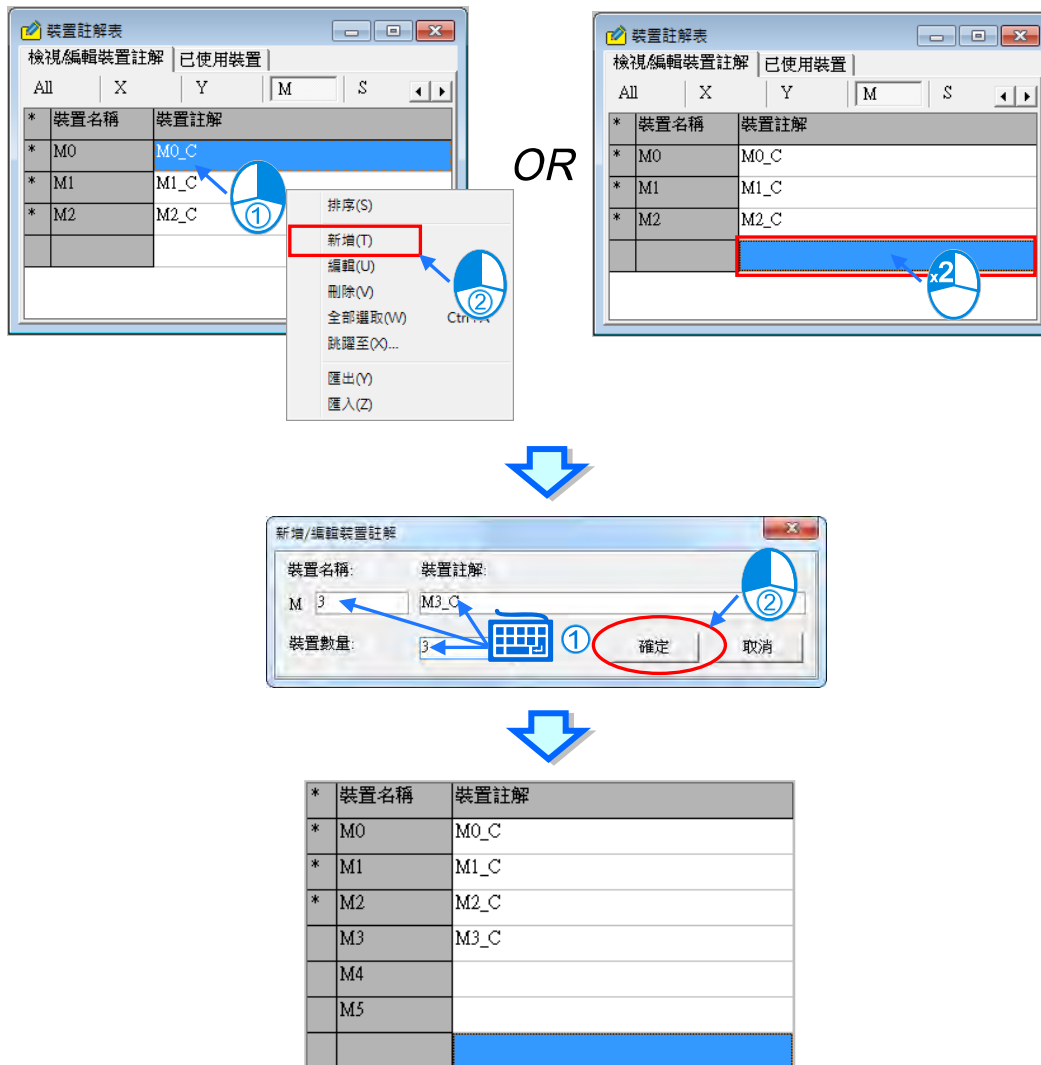
「檢視/編輯裝置註解」頁面會列出所有已加上裝置註解的裝置清單及其註解內容，而切換此頁面上方的標籤則可選擇欲檢視的裝置類別，其中選擇「All」代表不分類別，該頁面會列出所有已加上裝置註解的裝置。此外，當裝置名稱的左側出現「*」符號時，代表該裝置已在程式當中被使用。



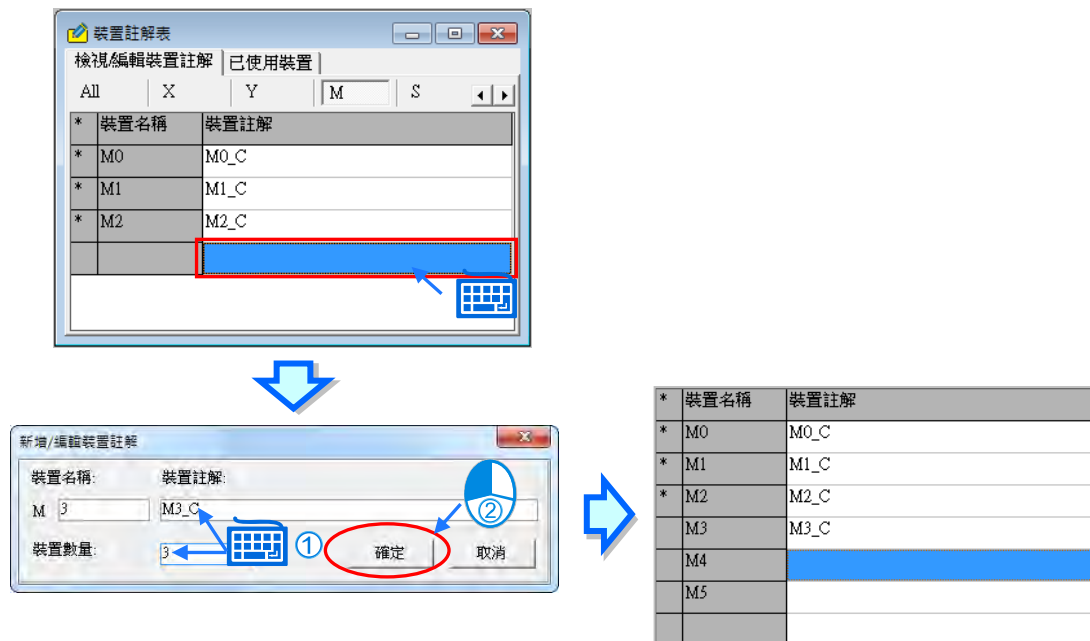
使用者亦可直接於「檢視/編輯裝置註解」頁面修改或新增裝置註解。要修改裝置註解時，請於要修改的項目上雙擊滑鼠左鍵，或是按下右鍵後於選單中點選「編輯」，之後便可於編輯視窗中進行修改。而其中 SR 與 SM 本身便會有系統預設的裝置註解，不過使用者還是可以重新編輯，但重新編輯後的註解則只限在這個專案中才有效。



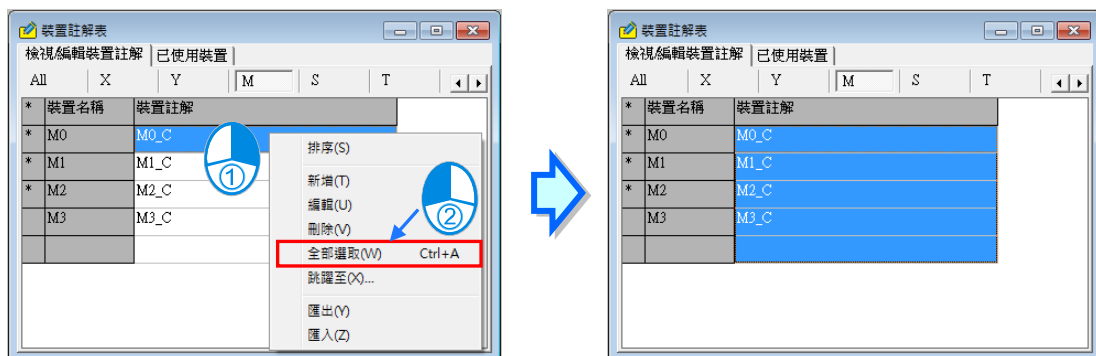
要新增裝置註解時，請按下滑鼠右鍵後於選單中點選「新增」，或直接於選單中雙擊滑鼠左鍵，之後畫面便會自動開啟編輯視窗，接著直接於編輯視窗中輸入裝置位置及註解即可，視窗底部的「裝置數量」，是讓使用者一次產生多筆裝置註解欄位來用，如下圖。



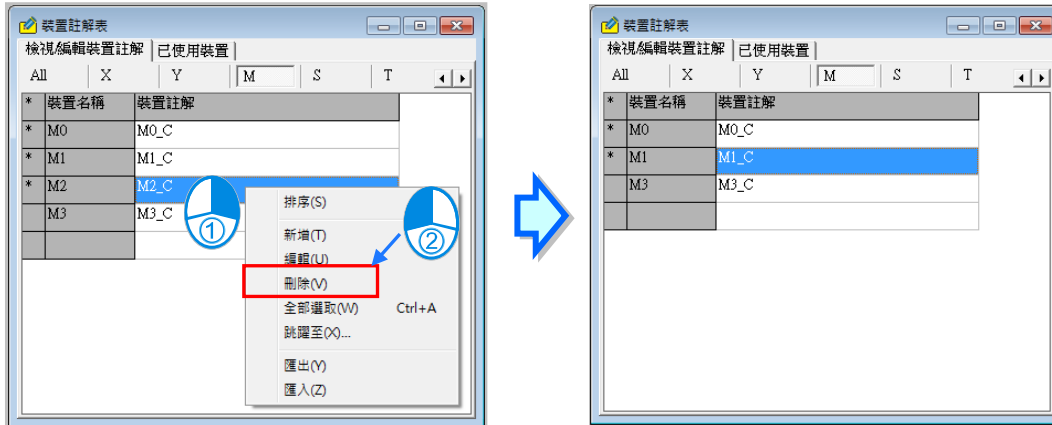
使用者也可以直接在此頁面以鍵盤輸入裝置位置，之後畫面便會自動開啟編輯視窗，接著直接於編輯視窗中輸入注解以及「裝置數量」即可，如下圖。



要移除裝置注解時，請先點選欲移除注解的項目；若要刪除大量注解時，可先點選第一個項目後按下鍵盤的【Shift】鍵，接著再點選最後一個項目，之後該範圍的項目便會被選取起來；而要選擇整個頁面的裝置注解時，請直接在該頁面中按下滑鼠右鍵，之後於快捷選單中點選「全部選取」即可。



選取完要刪除的項目後，接著請直接按下鍵盤的【DEL】鍵，或是在按下滑鼠右鍵後於快捷選單中點選「刪除」，之後便可將所選取的裝置註解全部清除。當刪除項目包含 SM 或 SR 的註解時，系統的處理動作則是將被選取的 SM 或 SR 的裝置註解恢復為系統預設值，而非將該項目刪除。

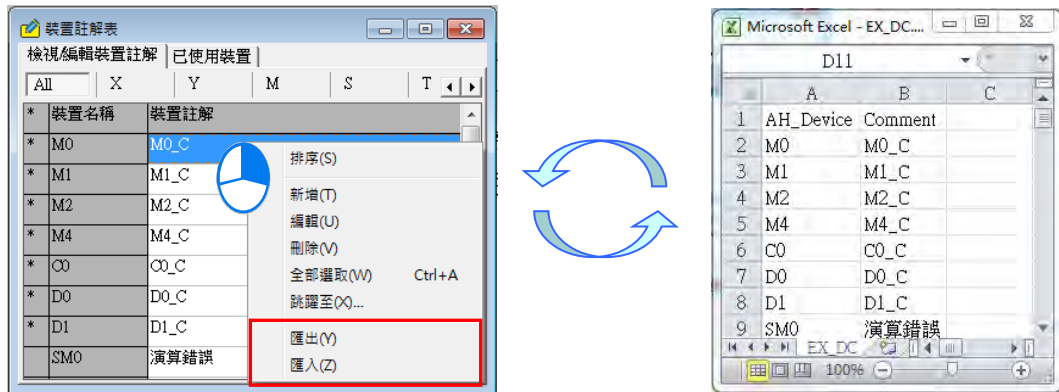


若要將裝置註解表中的項目重新排序時，請直接按下滑鼠右鍵，接著於快捷選單中點選「排序」，之後所有項目便會依據裝置位址而重新進行排序。



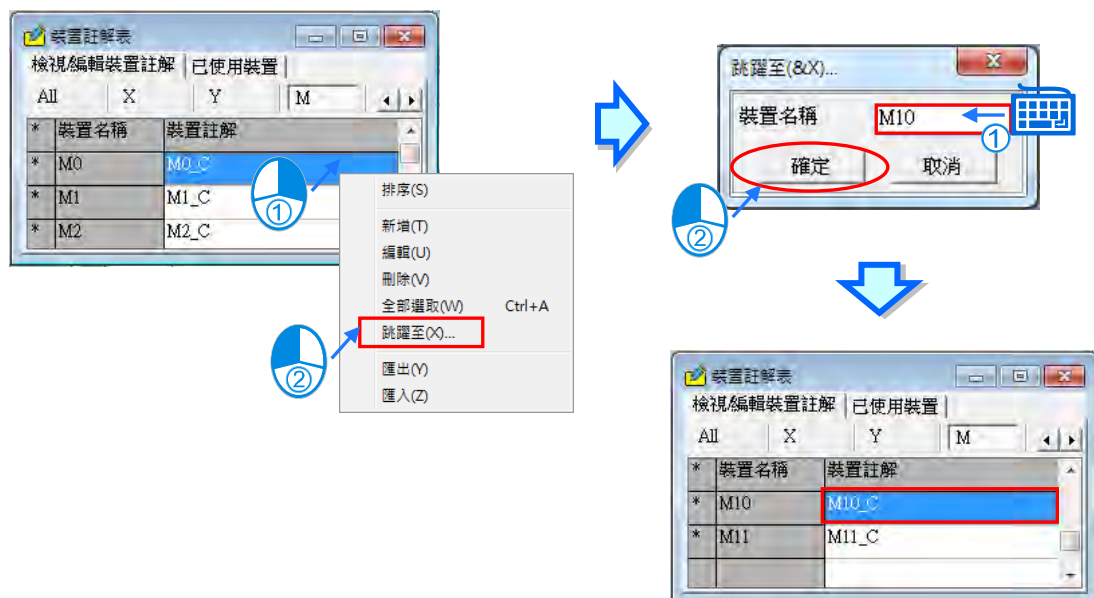
裝置註解表同樣可以 CSV 檔的格式進行匯出與匯入的操作，使用者可自行透過 EXCEL 來進行編輯與管理；而進行操作時，不分目前開啟的頁面，系統將會針對所有的裝置註解進行操作，不過針對 SM 與 SR 的部份，匯出時僅會包含使用者重新定義的項目，而保持系統預設的部份將不會被匯出；同樣的，當匯入的檔案中有包含 SM 或 SR 的項目時，匯入後亦只有該項目所對應的 SM 與 SR 註解會被變更，其餘部分則仍會保留為系統預設值。

欲進行「匯出」的操作時，請直接按下滑鼠右鍵，接著於快捷選單中點選「匯出」，之後再於檔案視窗設定欲匯出的檔名與路徑即可；同樣的，要進行「匯入」的操作時，請直接按下滑鼠右鍵，接著於快捷選單中點選「匯入」，之後再於檔案視窗選擇欲匯入的檔案即可。



欲快速定位到某個裝置時，按下滑鼠右鍵，接著於快捷選單中點選「跳躍至」後畫面便會出現一視窗，於「裝置名稱」欄位輸入欲檢視的裝置名稱後按下「確定」鍵，之後列表的顯示便會立即調整至可檢視該裝置的範圍。

輸入裝置名稱時，裝置的類別必須與目前的頁面一致，否則便會找不到該裝置；而於「已使用裝置」頁面中使用此功能時，輸入的裝置名稱也必須確實位於該清單之中才可進行操作。



17

「已使用裝置」頁面會列出所有已在程式中被使用的裝置清單與該裝置的相關資訊，而同樣也可藉由切換此頁面上方的標籤來選擇欲檢視的裝置類別。當在此頁面的某個項目上雙擊滑鼠左鍵時，系統便會自動引導使用者移至對應該項目的所在位置。



17.2.3 裝置使用狀況

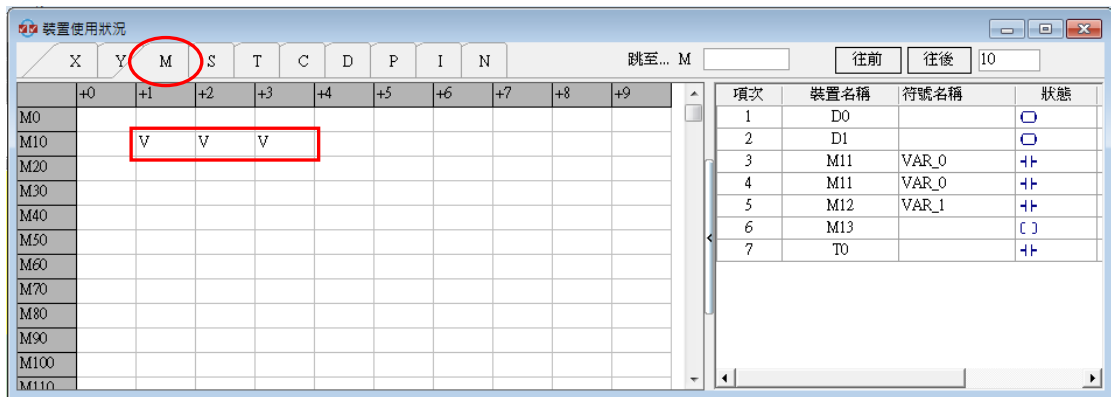
在編輯專案時，若利用此功能便可查詢專案中的裝置使用情形；不過若使用 DVP 系列機種，執行此功能之前必須先完成編譯的程序，此外，DVP 系列機種專案在開啟此表後曾經變更程式內容時，請先關閉表格並重新編譯後再開啟此表，如此表格中才可取得最新的資訊。

17

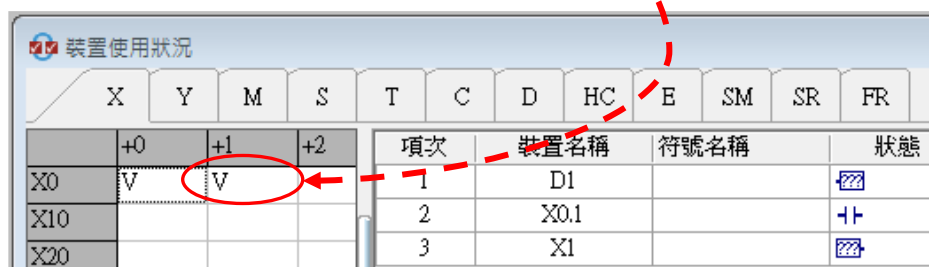
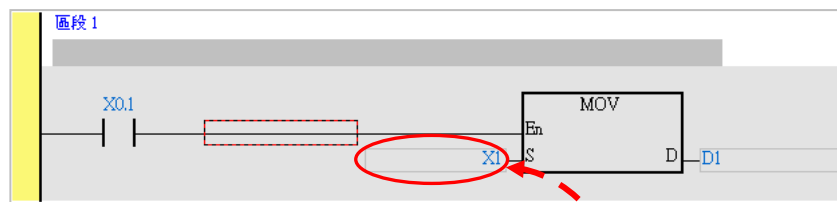
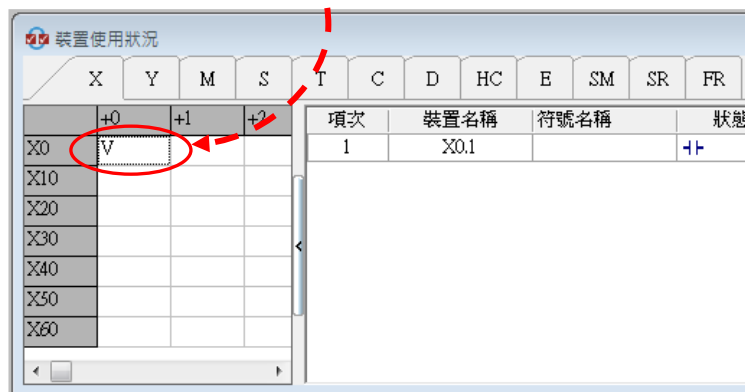
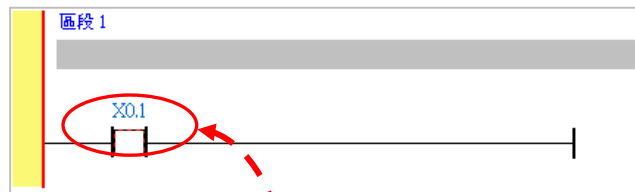
於功能工具列中點選 **檢視 (V)** > **裝置使用狀況**，或直接於專案管理區的「裝置使用狀況」項目上雙擊滑鼠左鍵，之後便可開啟管理視窗。



管理視窗主要分為兩個部份，其中左側為裝置的總表，點選視窗上方的標籤便可切換欲檢視的裝置類別，其中有打勾的欄位便代表該裝置已被使用於程式當中，如下圖的 M11、M12 與 M13。



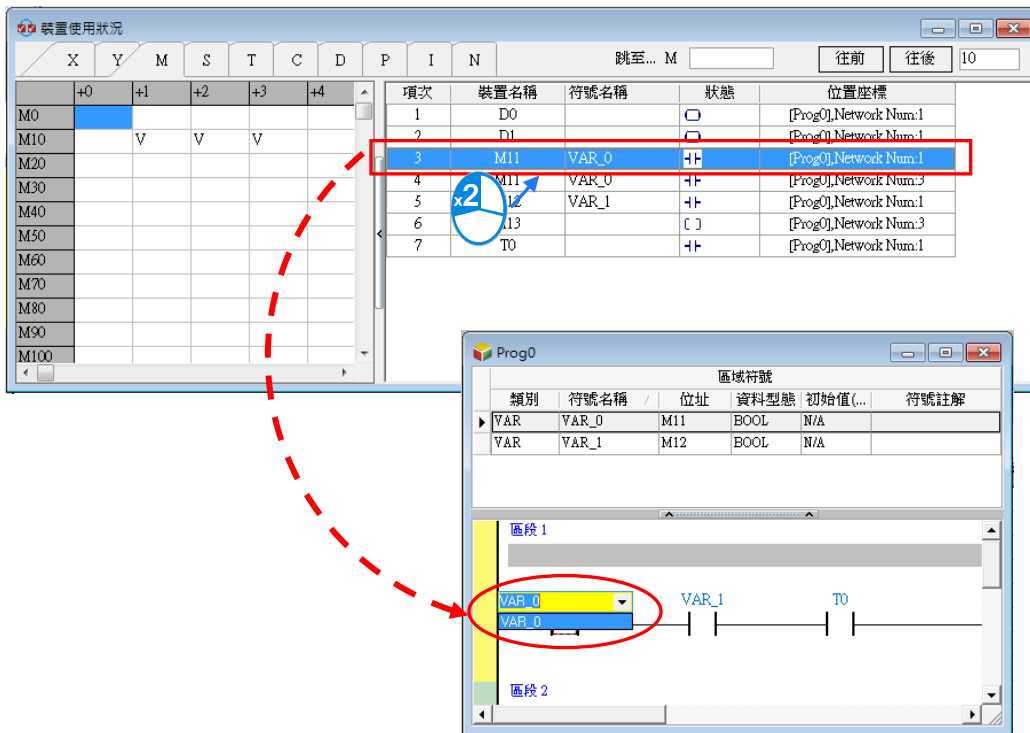
而若是選擇 AH/AS 系列機種則當有使用到此裝置之任一 Bit 或該裝置本身，該裝置欄位將會打勾。



在管理視窗的右側則會將所有已使用的裝置列於清單當中，且不會區分裝置類別；而當某個裝置在程式中被使用在不同位置時，每個位置的裝置資訊都會被列於清單之中，如下圖的 M11。



當於右側清單的項目上雙擊滑鼠左鍵時，系統便會自動引導使用者移至對應該項目的所在位置。




17

於快速移動欄位輸入裝置編號後按下【Enter】鍵，左側表格的藍色游標便會移動至該編號的位置。




若使用者欲前後移動藍色游標至指定編號，可先填寫欲移動數量再點擊**往前/往後**按鍵，來做前後移動。

	+0	+1	+2	+3
M20				
M30				
M40				




	+0	+1	+2	+3
M20				
M30				
M40				

跳至... M **往前** **往後**




	+0	+1	+2	+3
M20				
M30				
M40				



	+0	+1	+2	+3
M20				
M30				
M40				

跳至... M **往前** **往後**



17.2.4 暫存器編輯

ISPSOft 提供工具用以批次編輯 T、C、D、L 裝置的內容值，並可將其做為配方而進行儲存與下載的操作。開啟專案後，於功能工具列點選 **PLC 主機 (P) > 暫存器編輯**，之後畫面便會開啟編輯視窗。

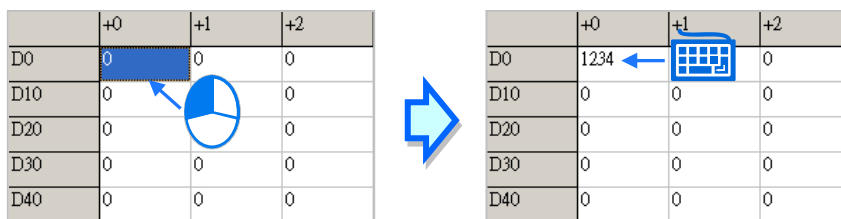
17



開啟視窗後，表格中所帶入的裝置數值並非由主機內部所上傳回來的目前值，而是上次開啟此視窗時所儲存的編輯內容，但若是在此專案中第一次開啟視窗時，表格的內容便會全部以 0 做為預設值。操作時，藉由切換視窗上方的標籤便可選擇欲檢視的裝置類別。

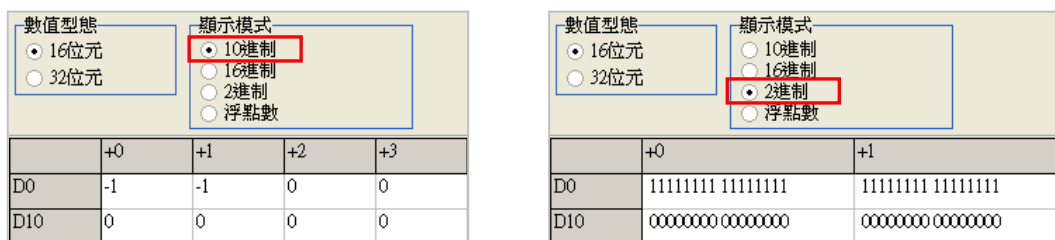


要編輯裝置內容時，請先點選欲編輯的裝置欄位後，直接以鍵盤輸入數值即可。



於「顯示模式」區域中可選擇表格中的數值顯示格式。

17



於「數值型態」區域中則可選擇數值的表達格式為 16 位元或 32 位元；當切換為 32 位元時，每個裝置中的數值將自動包含下一個裝置位址的內容而以 32 位元來呈現。

以下圖為例，D0 的 16 進制值為 AAAA，D1 的 16 進制值為 BBBB，D2 的 16 進制值為 CCCC；則當切換為 32 位元顯示時，D0 會自動包含 D1 的值而以 32 位元顯示為 BBBBAAAA，D1 則包含 D2 的值而顯示為 CCCCBBBB，而 D2 的顯示則因為 D3 的值為 0，所以在包含 D3 的值後會顯示為 CCCC。

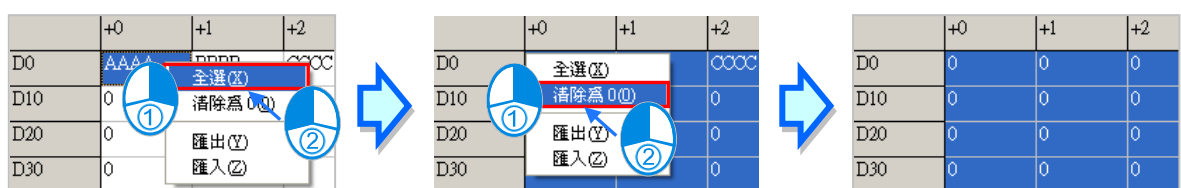
數值型態		顯示模式			
<input checked="" type="radio"/> 16位元 <input type="radio"/> 32位元		<input type="radio"/> 10進制 <input checked="" type="radio"/> 16進制 <input type="radio"/> 2進制 <input type="radio"/> 浮點數			
	+0	+1	+2	+3	
D0	AAAA	BBBB	CCCC	0	
D10	0	0	0	0	

數值型態		顯示模式		
<input type="radio"/> 16位元 <input checked="" type="radio"/> 32位元		<input type="radio"/> 10進制 <input checked="" type="radio"/> 16進制 <input type="radio"/> 2進制 <input type="radio"/> 浮點數		
	+0	+1	+2	
D0	BBBBAAAA	CCCCBBBB	CCCC	
D10	0	0	0	

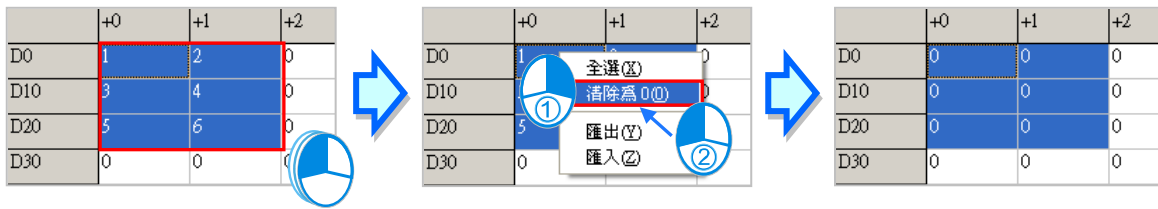
當按下視窗中的「全部清除」按鈕時，此頁面中的所有裝置內容便會被清除為 0；不過此操作僅會清除目前頁面中的裝置數值，其他頁面中的裝置數值則會保持原狀。



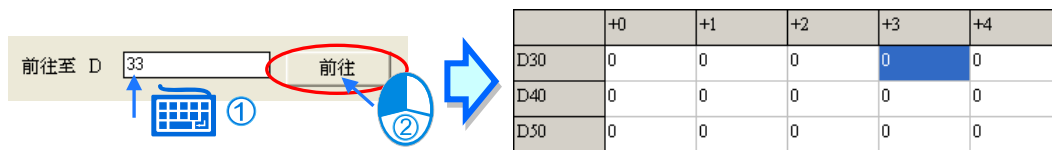
除此之外，亦可於頁面中按下滑鼠右鍵，接著於快捷選單中點選「全選」，之後再次按下滑鼠右鍵，並於選單中點選「清除為 0」便可將該頁面中的所有裝置內容清除。



若只想清除部份裝置時，可先使用滑鼠框選欲清除的範圍後，按下滑鼠右鍵並於快捷選單中點擊「清除為 0」，之後框選範圍內的裝置數值便會被清除。

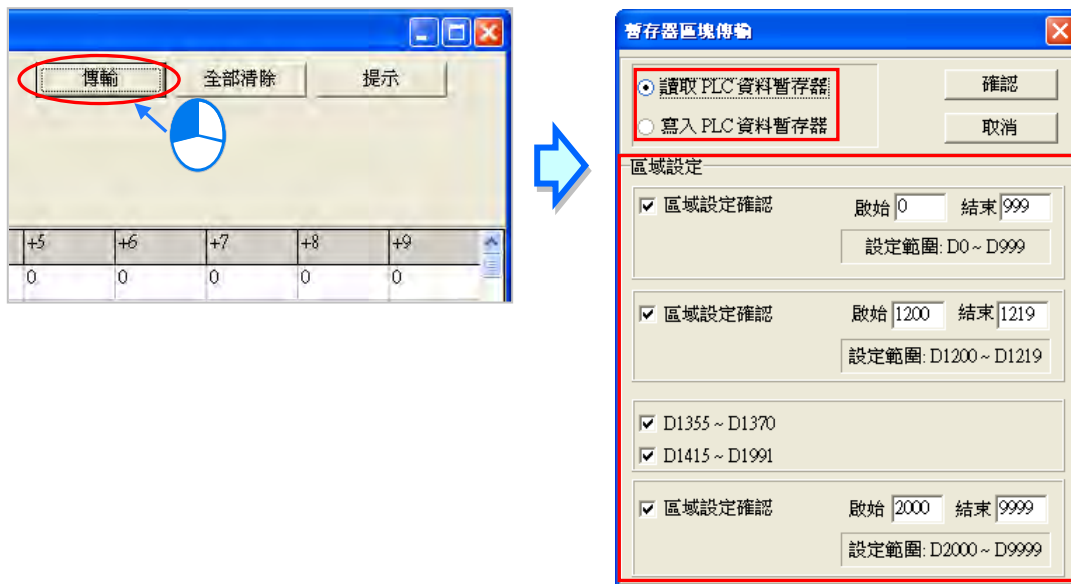


於前往至欄位輸入裝置編號後按下前往按鈕，下方表格的藍色游標便會移動至該編號的位置。



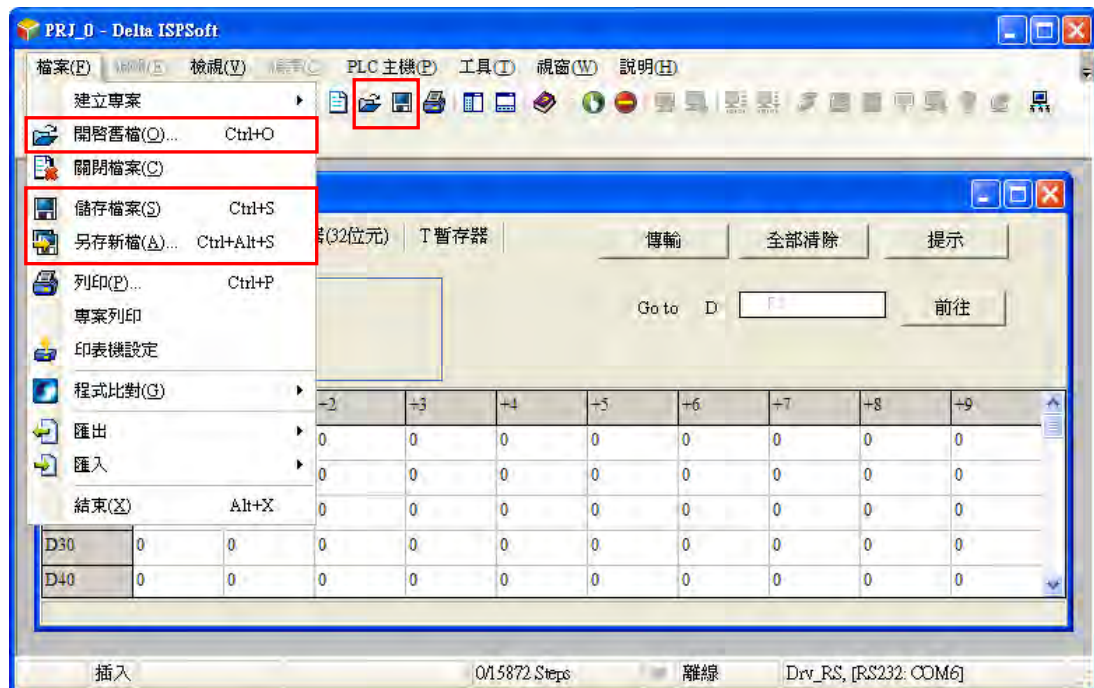
當按下視窗中的「傳輸」按鈕後，畫面中便會出現傳輸設定視窗，透過此視窗便可將表格中的數值下載至主機中，或是由主機中將資料讀取回來，但操作前請確認目前的 ISPSOft 已可與主機正常連線，詳細說明請參考第 2.4 節中的相關內容。

首先於視窗中選擇要執行的操作是要由 PLC 中讀取資料並填入編輯區中，或是將目前編輯的值寫入 PLC；而針對 D 裝置的部分，使用者還可於此處設定傳輸的範圍，未在勾選範圍內的數值則會保持原狀；當設定完成後按下「確認」按鈕即可開始傳輸，且傳輸的內容將不分裝置類別，包括 T、C 以及設定範圍內的 D 裝置都會一併被傳輸。

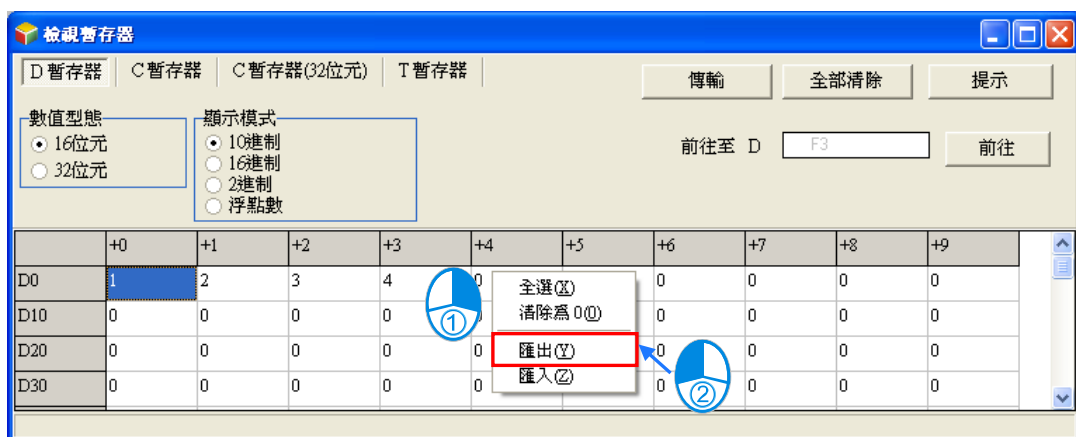


⚠ 將數值寫入暫存器前，請務必確認所做的操作不會影響系統運作，或造成系統與人員的傷害。

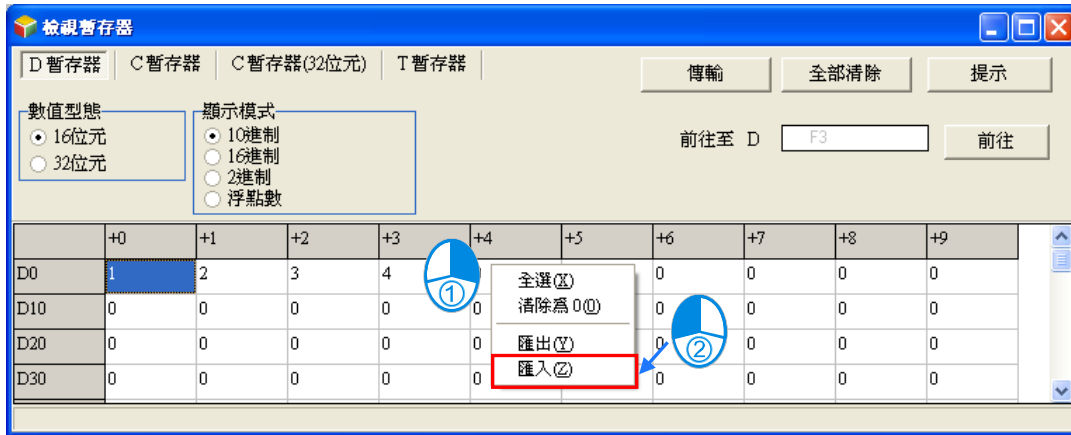
此處所編輯的內容亦可與專案一起儲存。當 ISPSOft 目前的編輯視窗為暫存器編輯工具時，於功能工具列中點選 **檔案 (F)** > **儲存檔案 (S)**，之後便可將此處的編輯內容儲存為 **專案名稱.dvl** 的檔案，並會與 ISPSOft 的專案檔 (*.isp) 存放於同一路徑；當下次於該專案中開啟此視窗時，系統便會自動於專案路徑下開啟此檔，但若專案路徑下沒有預定的檔案時，系統則會將表格的所有裝置值預設為 0。使用者亦可於功能工具列的 **檔案 (F)** 項目下點選 **另存新檔 (A)** 將其另外存放於自訂的路徑中，而利用 **檔案 (F)** 項目下的 **開啟舊檔 (O)** 則可將之前儲存的檔案再讀取回來。當然，所有的檔案操作也都可以透過圖示工具列中對應的圖示來進行操作。



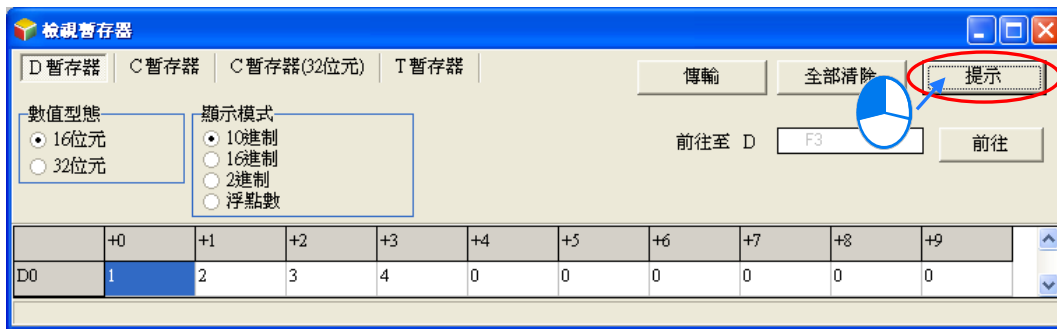
除了可將編輯的內容以檔案的型式儲存之外，使用者亦可將其匯出為標準的 CSV 檔，而之後便可直接於 EXCEL 當中進行編輯；操作時請於編輯頁面中點擊滑鼠右鍵，接著於快捷選單中點選「匯出」項目，在出現的設定範圍視窗選擇匯出的範圍，之後再指定欲匯出的路徑與檔名即可，而匯出的內容將僅包含當時所選取的裝置類別頁面。



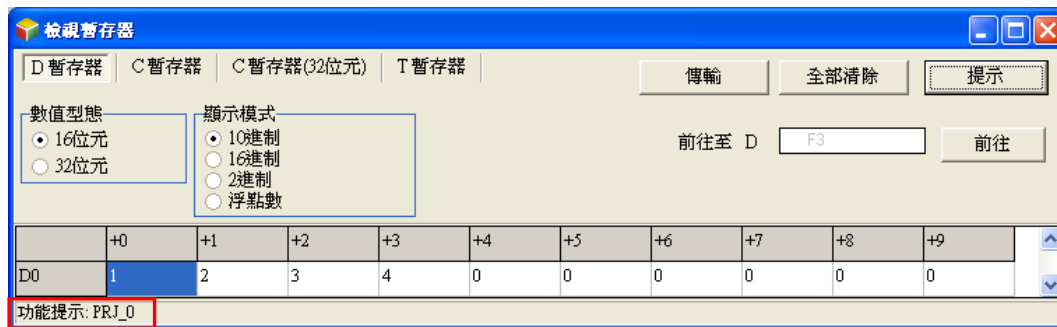
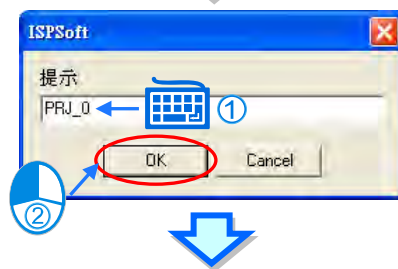
當要重新匯入檔案時，請於編輯頁面中點擊滑鼠右鍵並於選單中點選「匯入」，之後再指定欲匯入的檔案即可，不過匯入的檔案內容必須與目前頁面的裝置類別一致，不同裝置類別的內容將無法匯入。




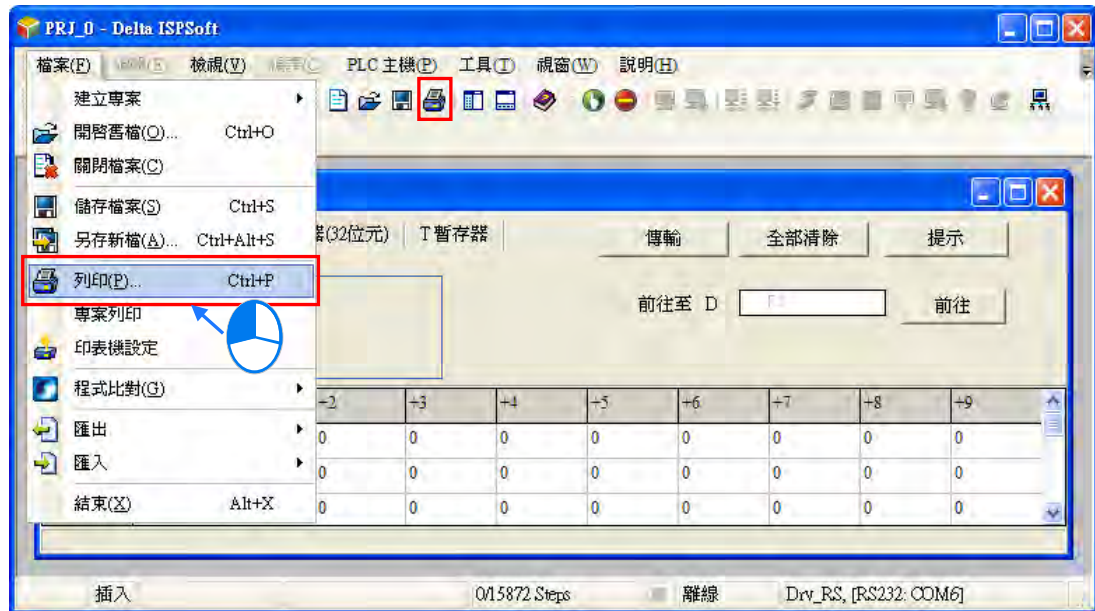
按下視窗中的「提示」按鈕後，畫面便會出現提示內容的輸入視窗，而完成輸入後，工具視窗的下方便會出現方才所設定的提示內容，且提示的內容將會隨著存檔時一併被儲存，當之後使用者開啟檔案時，由提示的內容便可清楚的辨別所開啟的表格，藉此即可便於使用者進行配方管理。



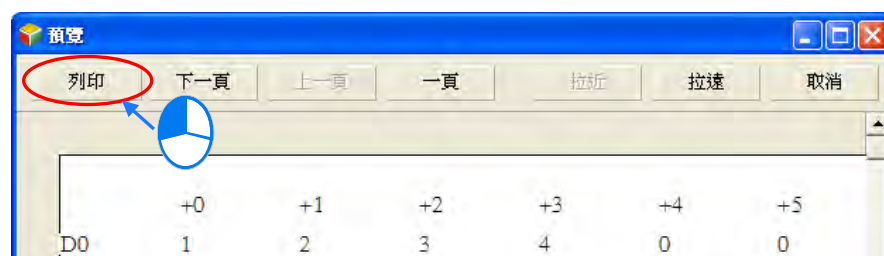
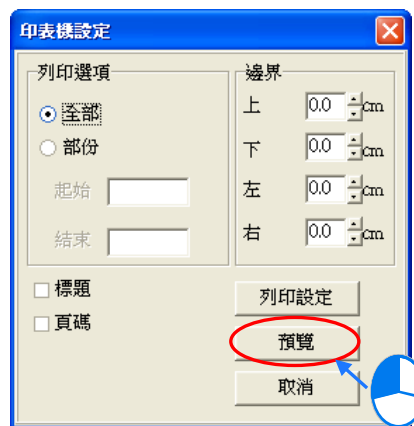
17



在開啟某個裝置類別的編輯頁面時，於 ISPSOft 的功能工具列點選 **檔案 (F) > 列印 (P)**，或者直接點擊圖示工具列中的  圖示，之後畫面便會出現列印管理視窗，而使用者便可藉此將目前所選的裝置類別頁面列印出來。

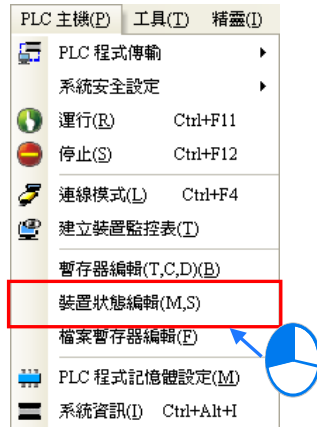


先於「**列印選項**」中設定欲列印的裝置範圍後，接著可再設定列印的**邊界**與欲附加於頁面中的項目；而按下「**列印設定**」則可開啟印表機的設定視窗，於此可設定使用的列表機與紙張大小等參數；完成後按下「**預覽**」便可開啟預覽畫面，確認無誤後按下預覽畫面的「**列印**」鍵即可將其印出。



17.2.5 裝置狀態編輯

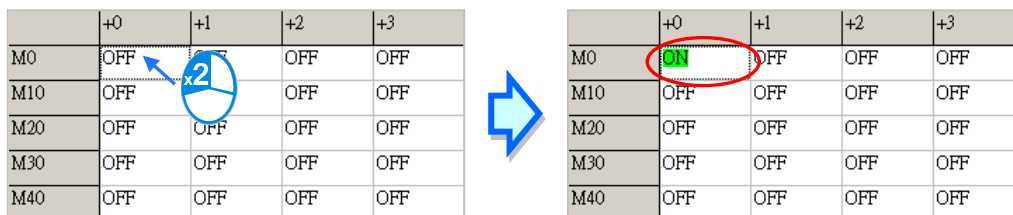
使用者亦可透過 ISPSOft 所提供的工具來批次編輯 M 裝置與 S 裝置的內容值，並可將其做為配方而進行儲存與下載的操作。開啟專案後，於功能工具列點選 **PLC 主機 (P)** > **裝置狀態編輯**，之後畫面便會開啟編輯視窗。



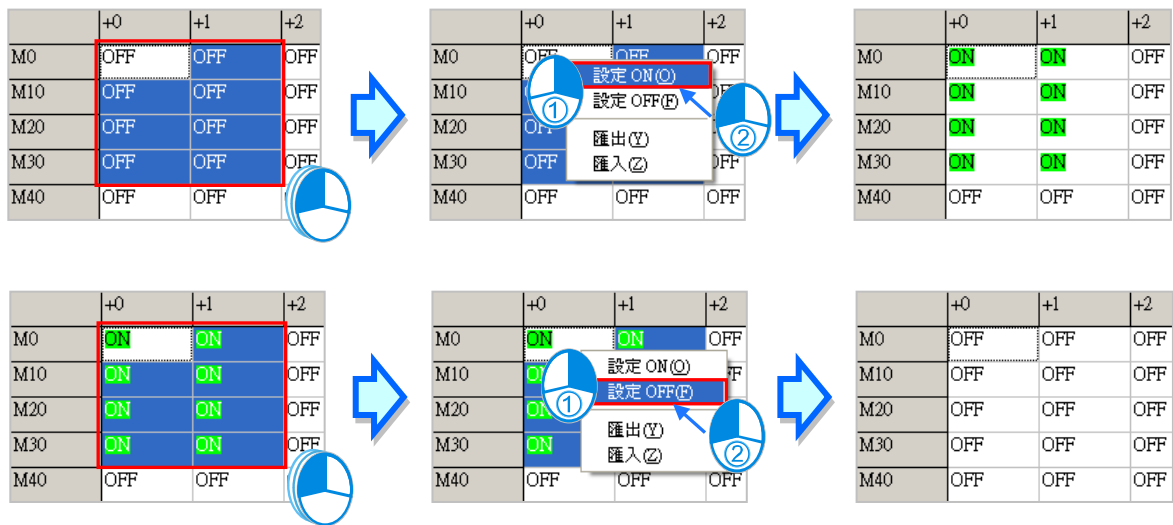
開啟視窗後，表格中所帶入的裝置數值並非由主機內部所上傳回來的目前值，而是上次開啟此視窗時所儲存的編輯內容，但若是在此專案中第一次開啟視窗時，所有的裝置便會以 OFF 做為預設值。操作時，藉由切換視窗上方的標籤便可選擇欲檢視的裝置類別。



要切換表格中的裝置狀態時，請直接於要編輯的欄位上雙擊滑鼠左鍵即可。



當要同時設定某段範圍的裝置狀態時，可先使用滑鼠框選欲設定的範圍後，按下滑鼠右鍵並於快捷選單中點選「設定 ON」或「設定 OFF」，之後框選範圍內的裝置便會全部被設為所選擇的狀態。

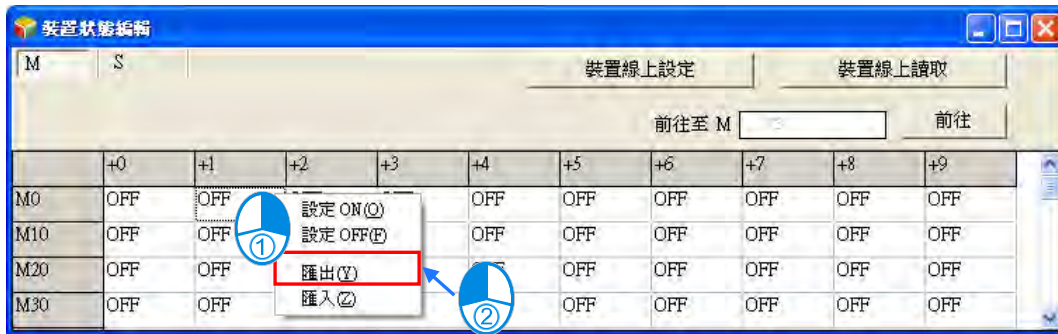


當按下視窗中的「裝置線上設定」按鈕後，系統便會將此處所做的編輯寫入主機當中，而當按下視窗中的「裝置線上讀取」按鈕時，系統則會將主機當中的狀態讀取回來，且傳輸時將不分裝置類別，包括 M 裝置與 S 裝置的內容將會一併傳輸；不過操作前請確認目前的 ISPSoft 已可與主機正常連線，詳細說明請參考第 2.4 節中的相關內容。

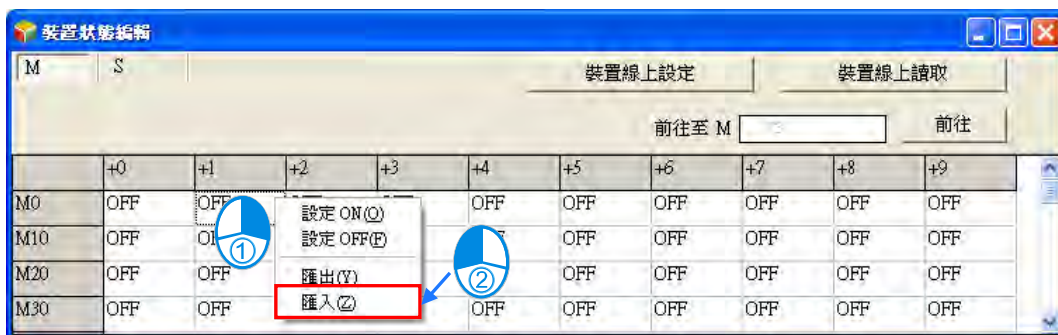


⚠ 將數值寫入主機前，請務必確認所做的操作不會影響系統運作，或造成系統與人員的傷害。

針對此處所編輯的內容，使用者可將其匯出為標準的 CSV 檔，而之後便可直接於 EXCEL 當中進行編輯；操作時請於編輯頁面中點擊滑鼠右鍵，接著於快捷選單中點選「匯出」項目，在出現的設定範圍視窗選擇匯出的範圍，之後再指定欲匯出的路徑與檔名即可，而匯出的內容將僅包含當時所選取的裝置類別頁面。



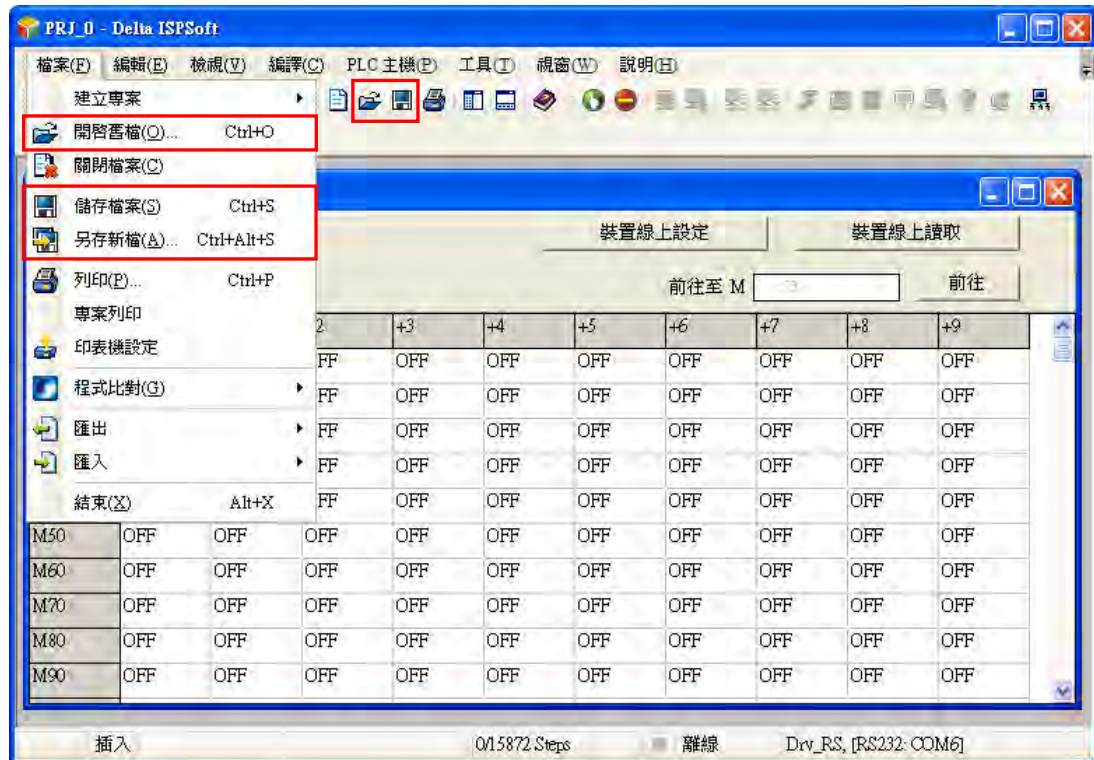
當要重新匯入檔案時，請於編輯頁面中點擊滑鼠右鍵並於選單中點選「匯入」，之後再指定欲匯入的檔案即可，不過匯入的檔案內容必須與目前頁面的裝置類別一致，不同裝置類別的內容將無法匯入。




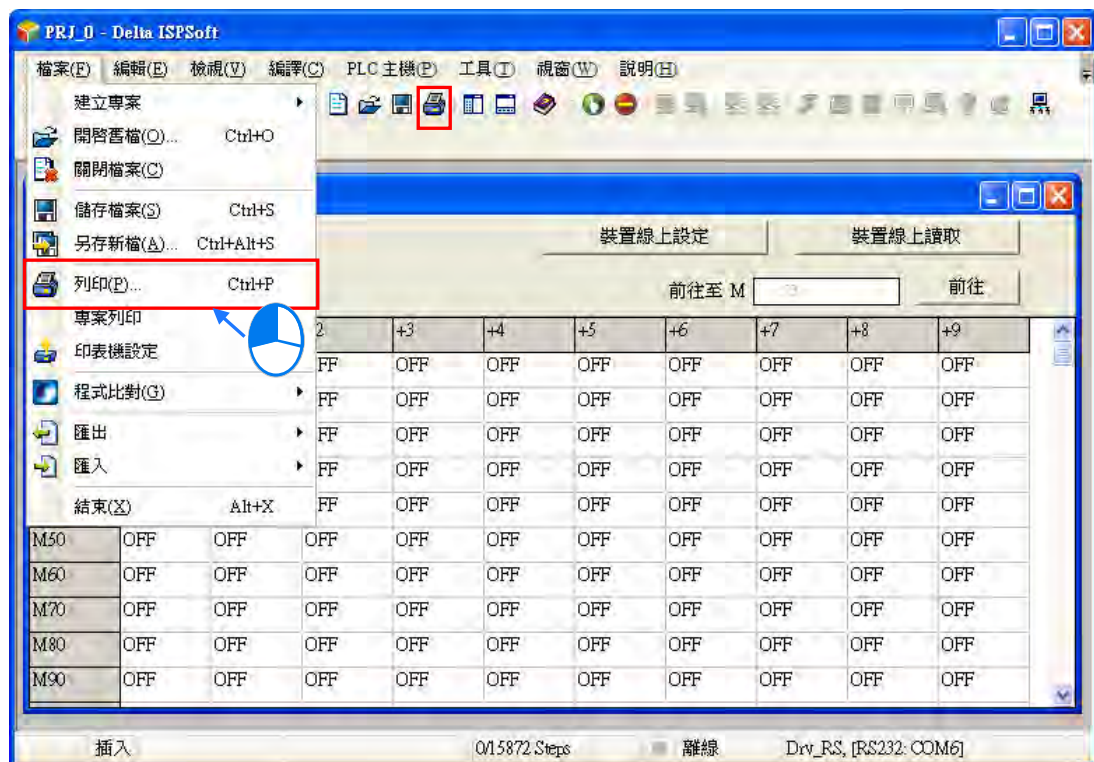
17

此處的編輯內容亦可與專案一起儲存，當 ISPSOft 目前的編輯視窗為裝置狀態編輯工具時，於功能工具列中點選 **檔案 (F)** > **儲存檔案 (S)**，之後便可將此處的內容儲存為 **專案名稱.dvb** 的檔案，並會與 ISPSOft 的專案檔 (*.isp) 存放於同一路徑；當下次於該專案中開啟此視窗時，系統便會自動於專案路徑下開啟此檔，但若專案路徑下沒有預定的檔案時，系統則會將所有的裝置值預設為 OFF。

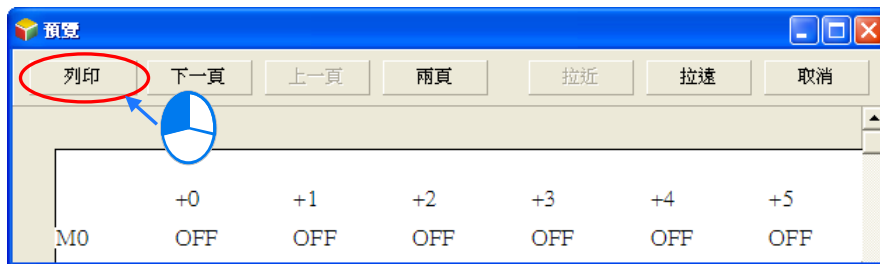
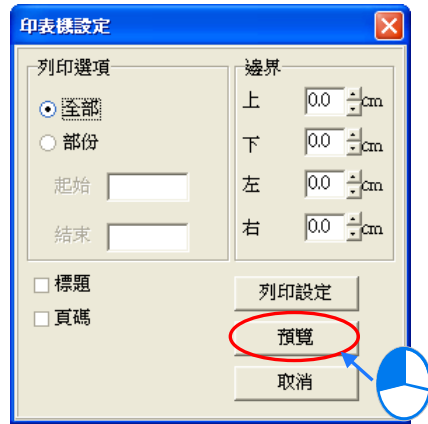
使用者亦可於功能工具列的 **檔案 (F)** 項目下點選 **另存新檔 (A)** 將其另外存放於自訂的路徑中，而利用 **檔案 (F)** 項目下的 **開啟舊檔 (O)** 則可將之前儲存的檔案再讀取回來。當然，所有的檔案操作也都可以透過圖示工具列中對應的圖示來進行操作。



在開啟某個裝置類別的編輯頁面時，於 ISPSOft 的功能工具列點選 **檔案 (F)** > **列印 (P)**，或者直接點擊圖示工具列中的  圖示，之後畫面便會出現列印管理視窗，而使用者便可藉此將目前所選的裝置類別頁面列印出來。



先於「列印選項」中設定欲列印的裝置範圍後，接著可再設定列印的邊界與欲附加於頁面中的項目；而按下「列印設定」則可開啟印表機的設定視窗，於此可設定使用的列表機與紙張大小等參數；完成後按下「預覽」便可開啟預覽畫面，確認無誤後按下預覽畫面的「列印」鍵即可將其印出。



17.2.6 DVP 機種的檔案暫存器編輯

針對 DVP 系列機種，使用者亦可透過 ISPSOft 所提供的工具來批次編輯檔案暫存器的內容值，並可將其做為配方而進行儲存與下載的操作。

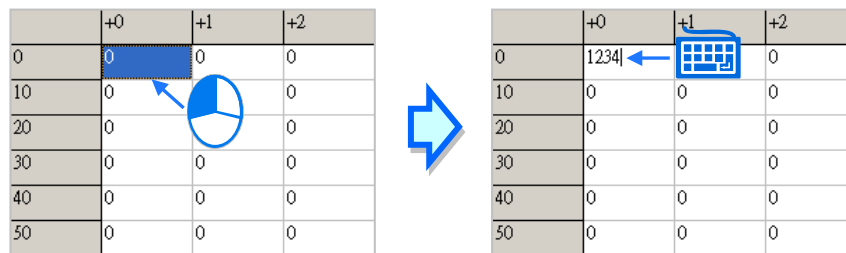
開啟 DVP 系列機種的專案後，於功能工具列點選 **PLC 主機 (P)** > **檔案暫存器編輯 (E)**，之後畫面便會開啟編輯視窗。



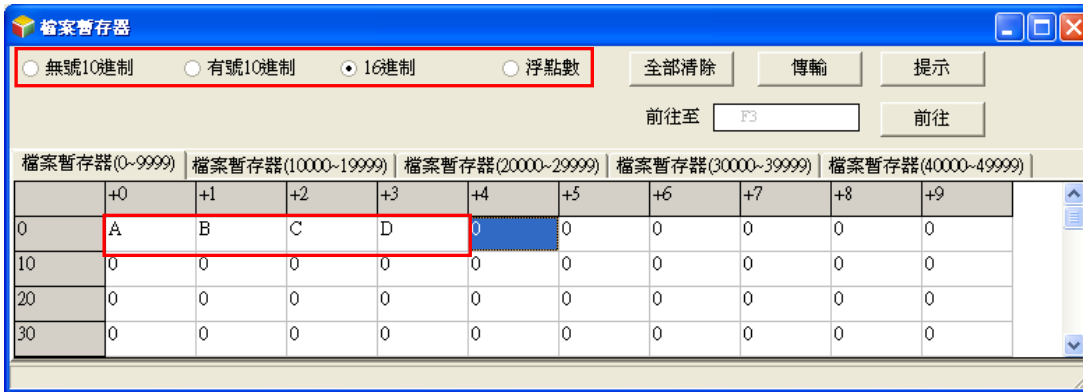
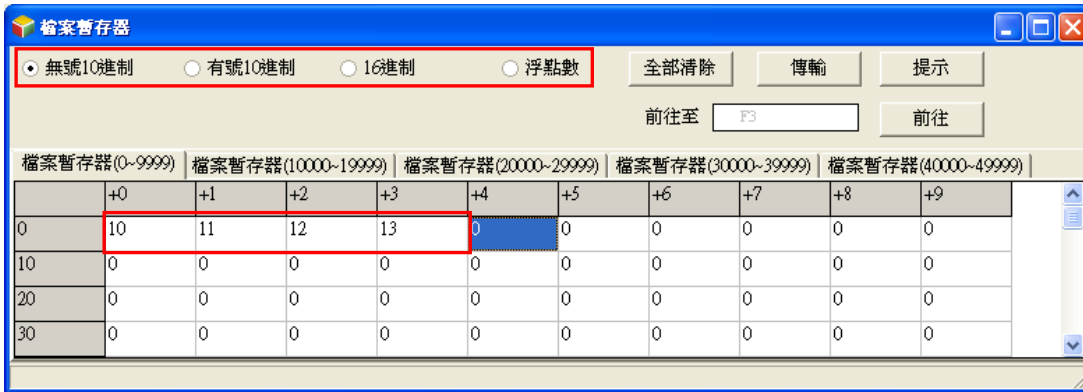
開啟視窗後，表格中所帶入的暫存器數值並非由主機內部所上傳回來的目前值，而是上次開啟此視窗時所儲存的編輯內容，但若是在此專案中第一次開啟視窗時，所有暫存器內容便會以 0 做為預設值。使用者點選標籤切換不同的檔案暫存器範圍。



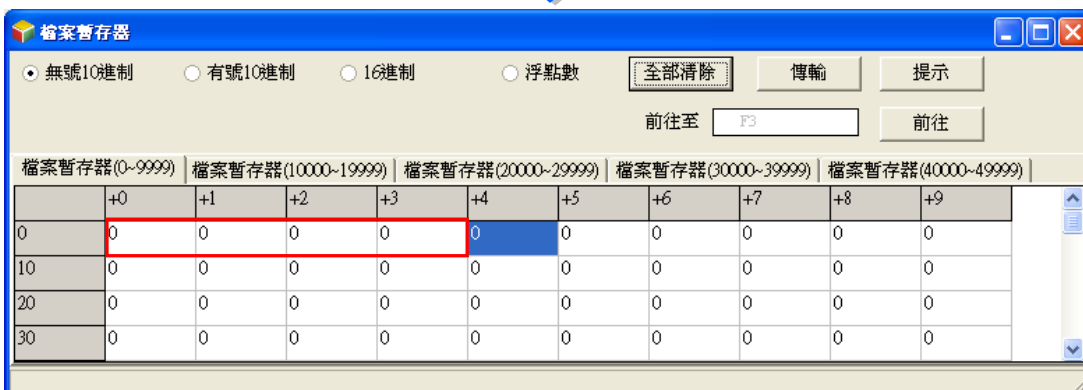
要編輯暫存器內容時，請先點選欲編輯的暫存器欄位後，直接以鍵盤輸入數值即可。



使用者可於視窗上方切換數值的顯示格式。



當按下視窗中的「全部清除」按鈕時，此頁面中的所有暫存器內容便會被清除為 0。



除此之外，亦可於頁面中按下滑鼠右鍵，接著於快捷選單中點選「全選」，之後再次按下滑鼠右鍵，並於選單中點選「清除為 0」便可將該頁面中的所有的暫存器內容清除。



只想清除部份暫存器時，可先使用滑鼠框選欲清除的範圍後，按下滑鼠右鍵並於快捷選單中點擊「清除為 0」，之後框選範圍內的暫存器數值便會被清除。



當按下視窗中的「傳輸」按鈕後，畫面中便會出現傳輸設定視窗，透過此視窗便可將表格中的數值下載至主機中，或是由主機中將資料讀取回來，但操作前請確認目前的 ISPSOft 已可與主機正常連線，詳細說明請參考第 2.4 節中的相關內容。

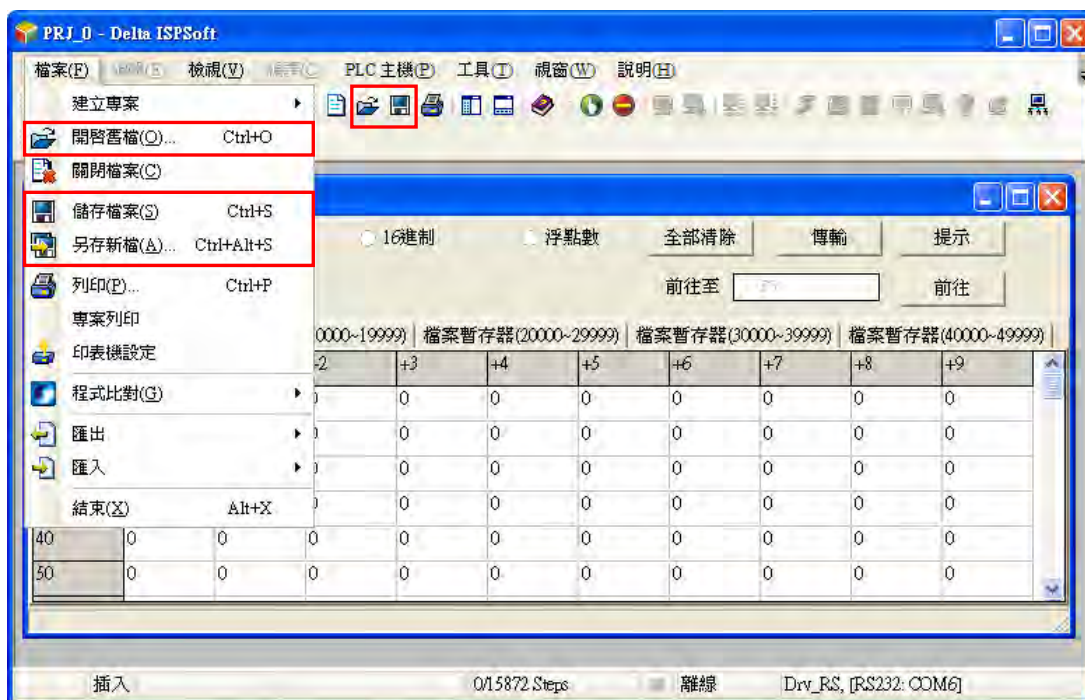
首先於視窗中的「傳輸型態」欄位下拉選擇要執行的操作是要由 PLC 中讀取資料並填入編輯區中，或是將目前編輯的值寫入 PLC，之後再於「傳輸設定」欄位中設定傳輸的範圍，完成後按下「確定」按鈕便可開始傳輸資料。



! 將數值寫入暫存器前，請務必確認所做的操作不會影響系統運作，或造成系統與人員的傷害。

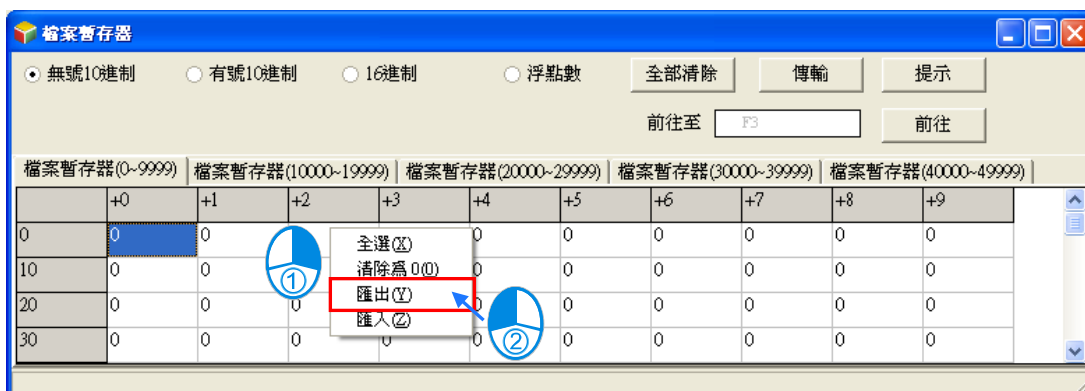
此處所編輯的內容可與專案一起儲存，當 ISPSOft 目前的編輯視窗為暫存器編輯工具時，於功能工具列中點選 **檔案 (F)** > **儲存檔案 (S)**，之後便可將此處的編輯內容儲存為 **專案名稱.wft** 的檔案，並會與 ISPSOft 的專案檔 (*.isp) 存放於同一路徑；當下次於該專案中開啟此視窗時，系統便會自動於專案路徑下開啟此檔，但若專案路徑下沒有預定的檔案時，系統則會將所有的暫存器內容值預設為 0。

使用者亦可於功能工具列的 **檔案 (F)** 項目下點選 **另存新檔 (A)** 將其另外存放於自訂的路徑中，而利用 **檔案 (F)** 項目下的 **開啟舊檔 (O)** 則可將之前儲存的檔案再讀取回來。當然，所有的檔案操作也都可以透過圖示工具列中對應的圖示來進行操作。

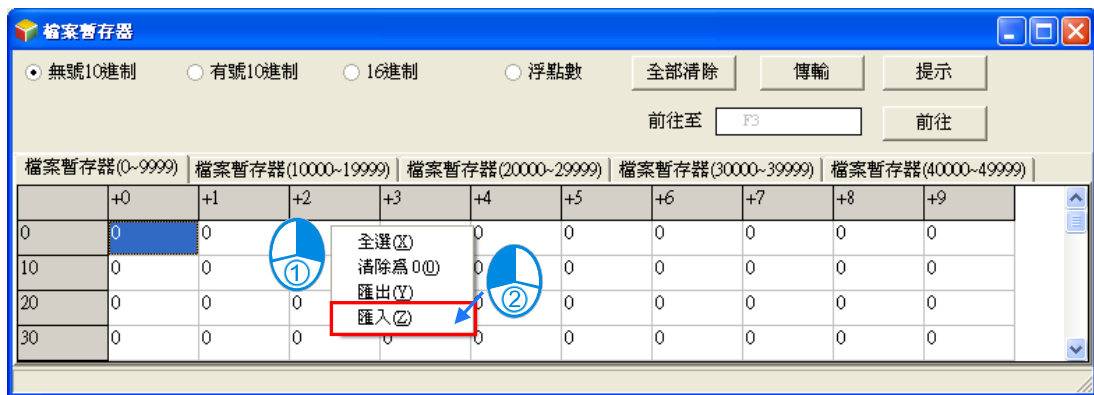


17

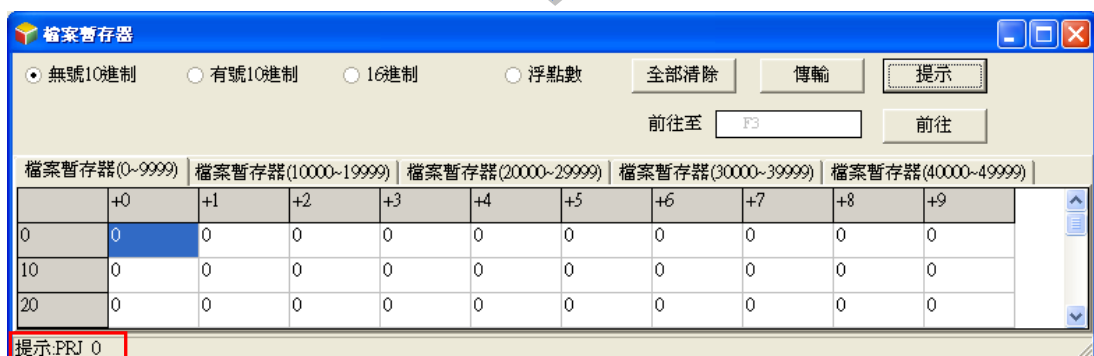
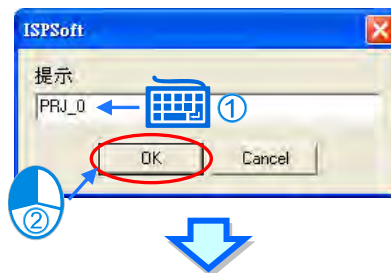
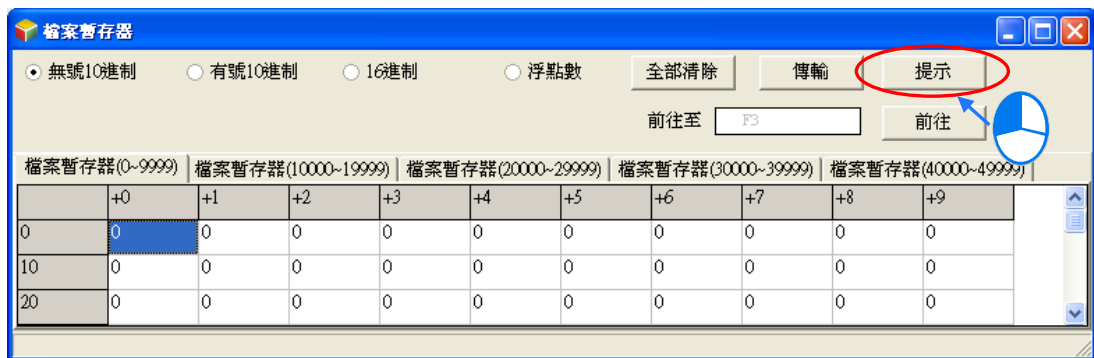
除了可將編輯的內容以檔案的型式儲存之外，使用者亦可將其匯出為標準的 CSV 檔，而之後便可直接於 EXCEL 當中進行編輯；操作時請於編輯頁面中點擊滑鼠右鍵，接著於快捷選單中點選「匯出」項目，之後再指定欲匯出的路徑與檔名即可。




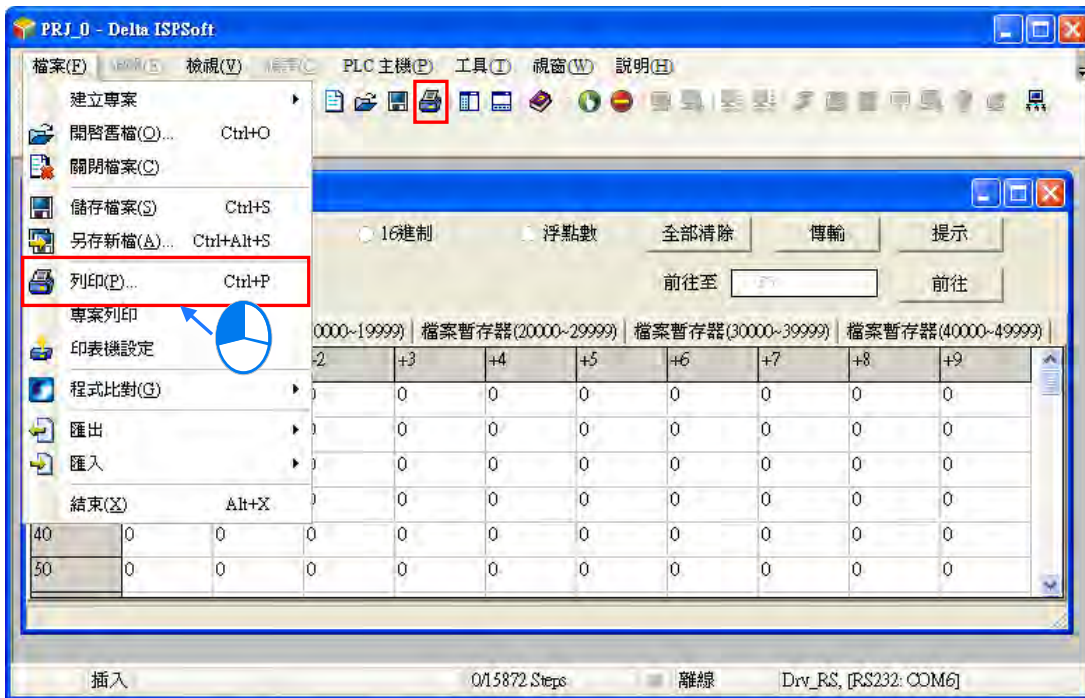
當要重新匯入檔案時，請於編輯頁面中點擊滑鼠右鍵，接著再於快捷選單中點選「匯入」，之後再指定欲匯入的檔案即可。



按下視窗中的「提示」按鈕後，畫面便會出現提示內容的輸入視窗，而完成輸入後，工具視窗的下方便會出現方才所設定的提示內容，且提示的內容將會隨著存檔時一併被儲存，當之後使用者開啟檔案時，由提示的內容便可清楚的辨別所開啟的表格，藉此即可便於使用者進行配方管理。

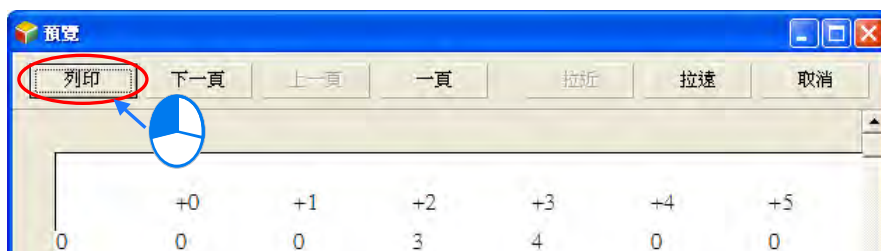
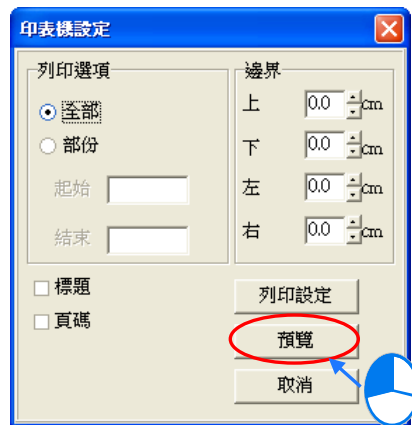


在開啟暫存器編輯工具時，於 ISPSOft 的功能工具列點選 **檔案 (F) > 列印 (P)**，或者直接點擊圖示工具列中的  圖示，之後畫面便會出現列印管理視窗，便可將目前的頁面列印出來。



先於「**列印選項**」中設定欲列印的裝置範圍後，接著可再設定列印的**邊界**與欲附加於頁面中的項目；而按下「**列印設定**」則可開啟印表機的設定視窗，於此可設定使用的列表機與紙張大小等參數；完成後按下「**預覽**」便可開啟預覽畫面，確認無誤後按下預覽畫面的「**列印**」鍵即可印出。

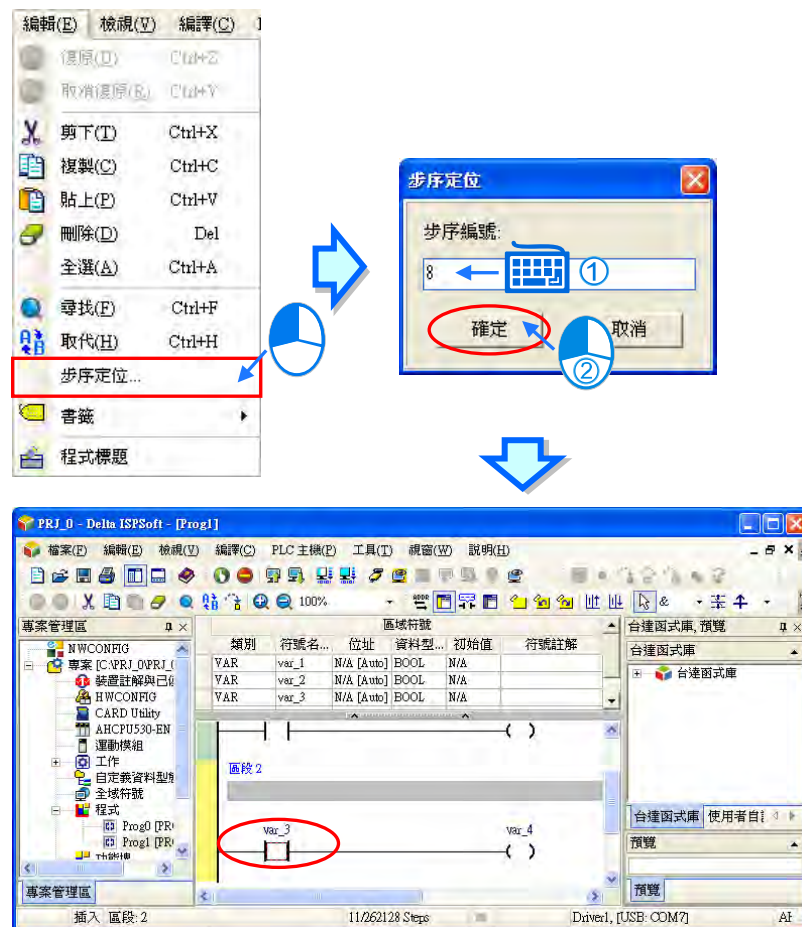
17



17.3 步序定位

17.3.1 使用步序定位

當主機運行中發生錯誤時，可藉由特殊暫存器（例如 AH 系列機種的 SR1 與 SR2 的 32 位元演算錯誤位址鎖定暫存器）得知錯誤的步（step）數，回到專案程式中尋找此步數所對應的程式，藉以了解發生錯誤的原因。於功能工具列點擊 **編輯 (E) > 步序定位...** 開啟步序定位視窗，於步序編號欄位輸入要查詢的步數後，按下確定按鈕，ISPSOft 便會自動開啟對應此步數的 POU 並定位到程式中。如下圖。使用步序定位時，專案需先行編譯。



17.4 程式比對

17.4.1 程式比對簡介

ISPSOft 提供了一個程式比對工具，供使用者比對電腦中的兩個 ISPSOft 專案，或是 ISPSOft 專案與連線的 PLC 主機內的程式。比對的規則如下。

- 比對的目標包括機種、型號、工作、全域符號、POU、區域符號、程式。
- POU 會先比對 POU 名稱，不同的 POU 名稱即便內容相同也將判別為不同。
- 變數符號表即便兩者有相同的內容，變數符號的排列順序不同也將判別為不同。如下圖，右側的符號表 var_a 與 var_b 與左側符號表的順序不同，故前兩行判別為不同。

類別	符號名稱	位址	資料型態	初始值
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A

類別	符號名稱	位址	資料型態	初始值
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A

- 變數符號表會比對出程式的跳行，如下圖。比對的結果將只判別為左側符號表第 3 行不存在於右側的符號表，其他相同。

類別	符號名稱	位址	資料型態	初始值
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_c	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A
VAR	var_e	N/A [Auto]	BOOL	N/A

類別	符號名稱	位址	資料型態	初始值
VAR	var_a	N/A [Auto]	BOOL	N/A
VAR	var_b	N/A [Auto]	BOOL	N/A
VAR	var_d	N/A [Auto]	BOOL	N/A
VAR	var_e	N/A [Auto]	BOOL	N/A

- LD、FBD、IL、ST 語言 POU 會比對出程式的跳行，如下圖。比對的結果將只判別為左側程式 POU 第 2 行不存在於右側的程式 POU，其他相同。

0001 ld var1	0001 ld var1
0002 and var3	0002 out var2
0003 out var2	0003 ld m0
0004 ld m0	0004 out var3
0005 out var3	

- SFC 語言 POU 的比對只要 SFC 有差異即判別為整個 POU 不同；但 SFC 中的 Action 或 Transition 不同時會辨別出 Action 或 Transition 中不同的區段或行。
- 專案需先存檔後，才可使用者程式比對功能。
- 主機在下載時必須勾選程式項目，之後才可比對此 PLC 主機的內容。

17.4.2 與檔案比對

用於比對電腦中的不同專案。請點選 **檔案 (F) > 程式比對 (G) > 與檔案比對 (E)**，在出現的開啟專案視窗中選擇要比對的目標專案，按下**開啟**按鈕。

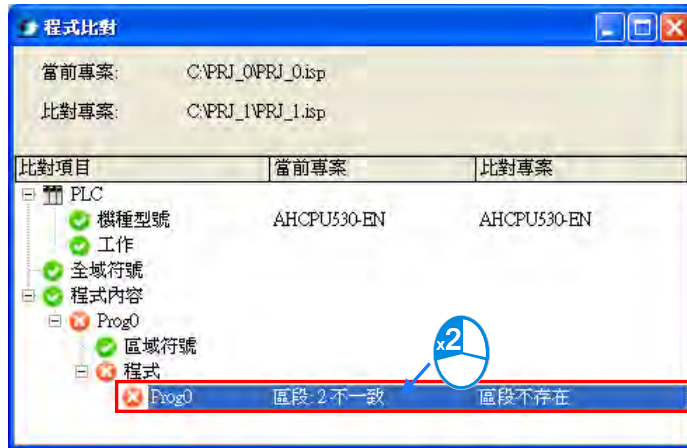


之後便會出現程式比對視窗，列出兩者的異同。說明如下。



- ❶ 目前開啟的專案位置。
- ❷ 比對的目標專案位置。
- ❸ 比對的結果，列出機種型號、工作、全域符號、程式內容的異同。

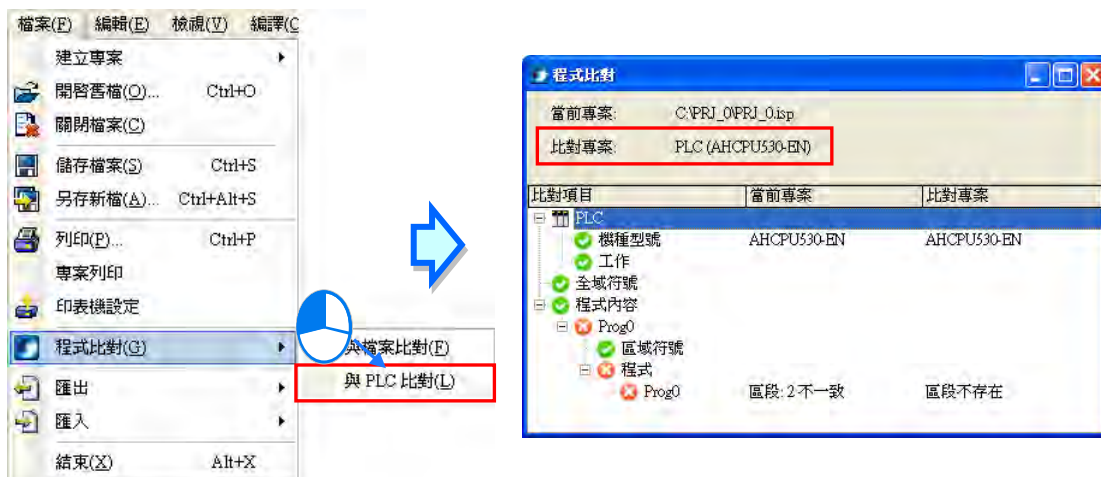
在不一致的項目上雙擊滑鼠左鍵，將自動開啟此項目的內容，如下圖，開啟 Prog0 定位並在區段 2。



17

17.4.3 與 PLC 比對

用於比對主機內容與電腦中的專案。請先確認 ISPSOft 與主機已建立連線，再點選 **檔案 (E) > 程式比對 (G) > 與 PLC 比對 (L)**，ISPSOft 將讀取目前連線主機的專案，開啟程式比對視窗，顯示比對項目內容。如下圖，可以看到比對專案為目前連線的 PLC 主機機種。



比對目標的 PLC 主機在下載時必須勾選**程式**項目，若與未下載程式的主機進行比對，則會出現訊息告知使用者無法比對。下載功能請參考第 17.1.3 節。



MEMO

17

第18章 測試與除錯工具



目錄

18.1 線上監控	18-2
18.1.1 切換主機的運轉/停止狀態	18-2
18.1.2 線上監控的功能與環境介紹	18-2
18.1.3 線上模式下變更 X 裝置的狀態	18-7
18.1.4 程式監控的線上操作	18-9
18.1.5 裝置監控表	18-16
18.1.6 線上編輯與線上更新	18-22
18.2 DVP 機種的除錯模式	18-25
18.2.1 啟動 DVP 的除錯模式	18-25
18.2.2 設定與清除中斷點	18-26
18.2.3 除錯模式下的程式執行	18-26
18.3 AH/AS 機種的除錯模式	18-28
18.3.1 進入除錯模式	18-28
18.3.2 中斷點操作	18-28
18.3.3 連續執行	18-29
18.3.4 單步執行	18-30
18.4 確認 PLC 狀態	18-31
18.4.1 PLC 系統資訊	18-31
18.4.2 PLC 系統記錄	18-33
18.5 3D 座標圖	18-40
18.5.1 功能說明	18-40
18.5.2 建立 3D 座標圖	18-41
18.5.3 加入曲線	18-42
18.5.4 顯示 3D 座標圖	18-44

18.1 線上監控

18.1.1 切換主機的運轉/停止狀態

在進行測試與除錯的過程中，須要經常性地改變主機的運行狀態，而透過 ISPSOft，我們可輕易的進行切換。在操作前，請確認目前 ISPSOft 已可與主機正常連線，詳細說明請參考第 2.4 節。

於功能工具列中點選 **PLC 主機 (P)** > **運行 (R)**，或按下圖示工具列的  圖示可將主機切換至 **RUN** 的狀態；而於功能工具列中點選 **PLC 主機 (P)** > **停止 (S)**，或按下圖示工具列的  圖示則可重新將主機切換至 **STOP** 狀態。





透過 ISPSOft 來切換主機的狀態時，並不用考慮主機本體的 **RUN/STOP** 開關位置；且由 ISPSOft 到達 **RUN/STOP** 命令後，若再次切換主機本體的 **RUN/STOP** 開關時還是可以變更主機的運行狀態。


18.1.2 線上監控的功能與環境介紹

當 ISPSOft 已可與主機正常連線後，我們便可經由**線上監控**模式來對 PLC 的執行狀況進行監控。關於主機與 ISPSOft 之間的連線設定方式請參考第 2.4 節的內容。




在 ISPSOft 中，**線上監控**的模式又可分為「**裝置監控**」與「**程式監控**」。

監控模式	說明
 裝置監控	可透過監控表來即時監控主機目前的裝置狀態，且在此模式下，ISPSOft 僅需更新裝置狀態，因此目前 ISPSOft 所開啟的程式與主機內部的程式無須一致。
 程式監控	在此模式下，系統會即時將程式的運作狀況顯示於程式畫面中，也因此系統會要求目前 ISPSOft 所開啟的程式必須與主機內部的程式一致。


*.裝置監控模式可單獨啟動，而程式監控模式則必須伴隨裝置監控模式一起啟動。

直接於功能工具列中點選 **PLC 主機 (P)** > **連線模式 (L)**，或按下圖示工具列的  圖示便可進入 **線上監控**，而此時系統將會同時啟動「裝置監控」與「程式監控」兩種模式。



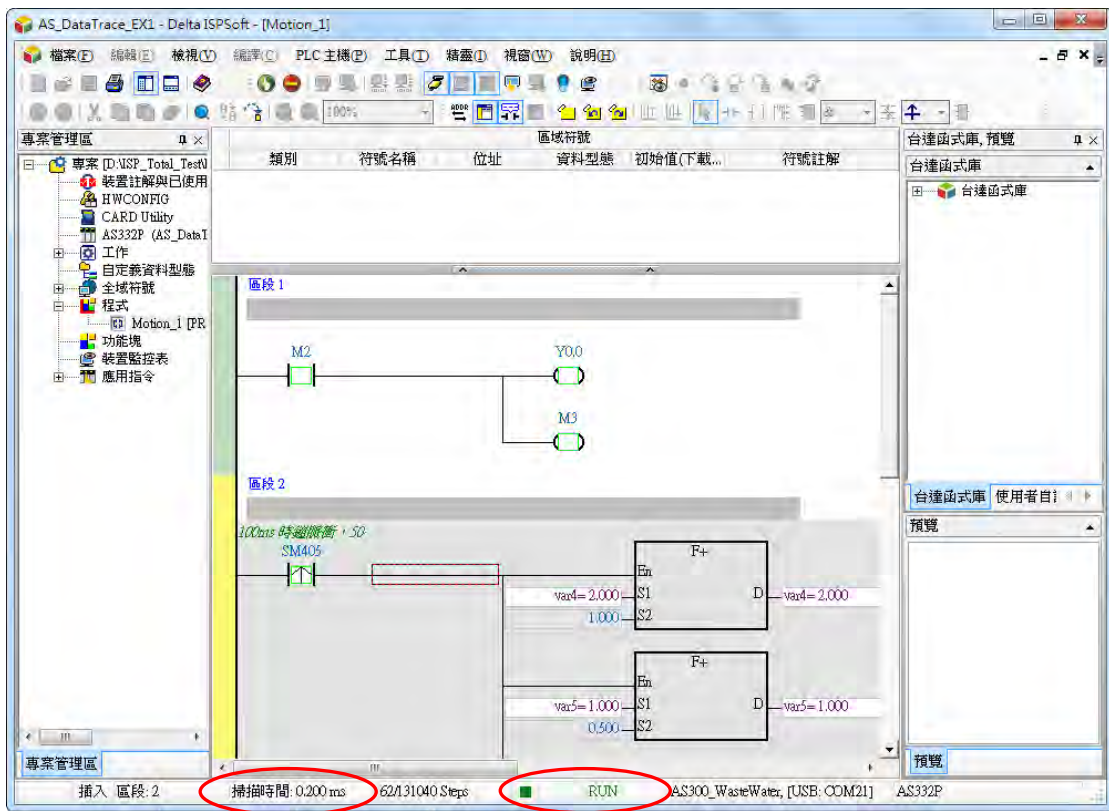
若一開始只要進行**裝置監控**時，按下圖示工具列的  圖示便可進入**裝置監控**模式，但此時**線上監控**的圖示  同樣會呈現被按下的狀態，代表已進入**線上監控**；而後續若要再進行**程式監控**時，只要再按下圖示工具列的  圖示便可進入**程式監控**的狀態。



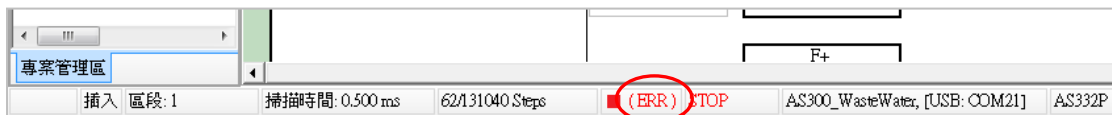
當要結束**線上監控**時，只要再次於功能工具列中點選 **PLC 主機 (P)** > **連線模式 (L)**，或按下圖示工具列的  圖示並將它恢復為未按下的狀態即可。



進入線上模式後，於視窗下方的狀態列便可知程式掃描時間與主機目前的狀態，而位於主機狀態左側的燈號則會即時顯示通訊狀況，當持續進行通訊時，該燈號便會維持綠燈的狀態。



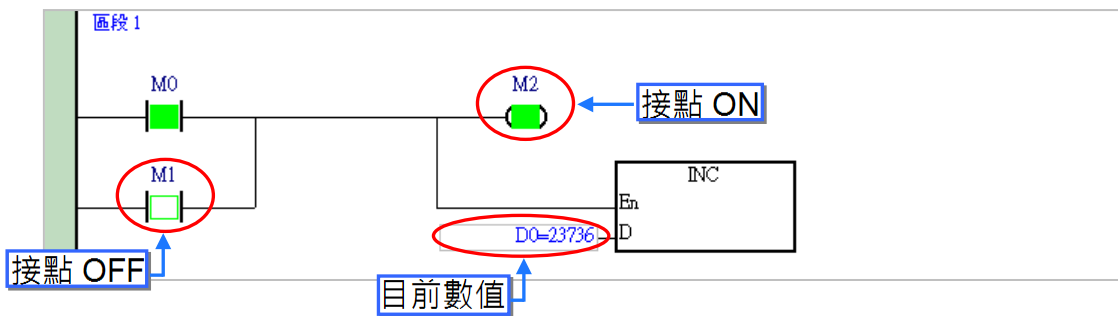
當主機處於錯誤狀態時，部分的操作可能無法進行，例如可能無法將主機切換至 RUN 狀態，此時請先參考各機種的使用手冊進行錯誤排除。



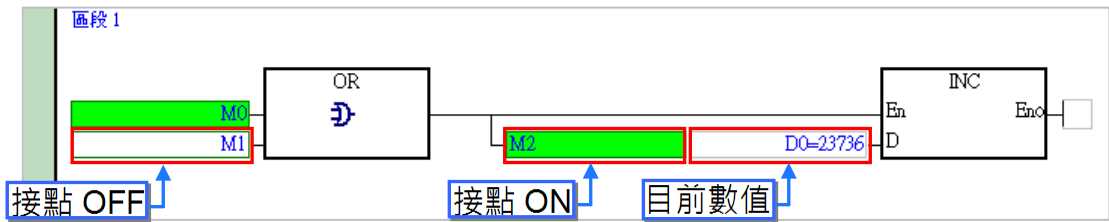
18

當進入「程式監控」的狀態後，程式的畫面會即時顯示各個裝置的狀態與現在值，藉此可進行相關的除錯與測試的工作；不過針對不同的編程語言，其顯示的畫面還是會有些許的差異。

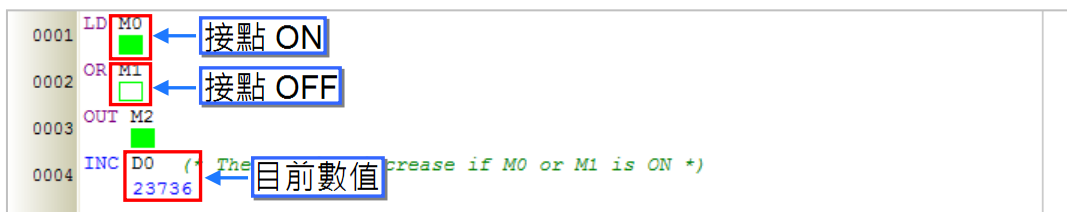
下圖為**階梯圖 (LD)** 的程式監控畫面。



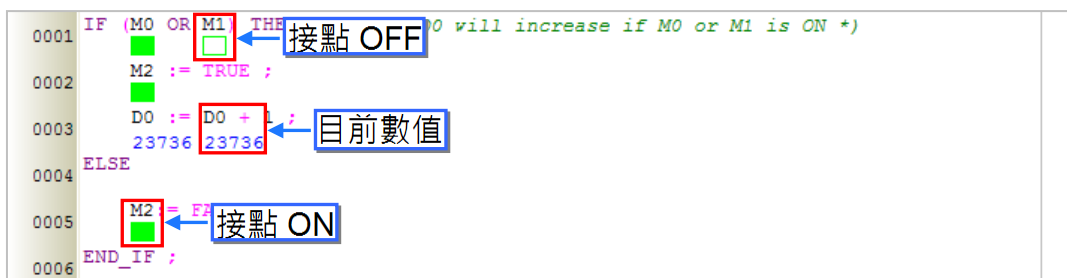
下圖為功能塊圖 (FBD) 的程式監控畫面。



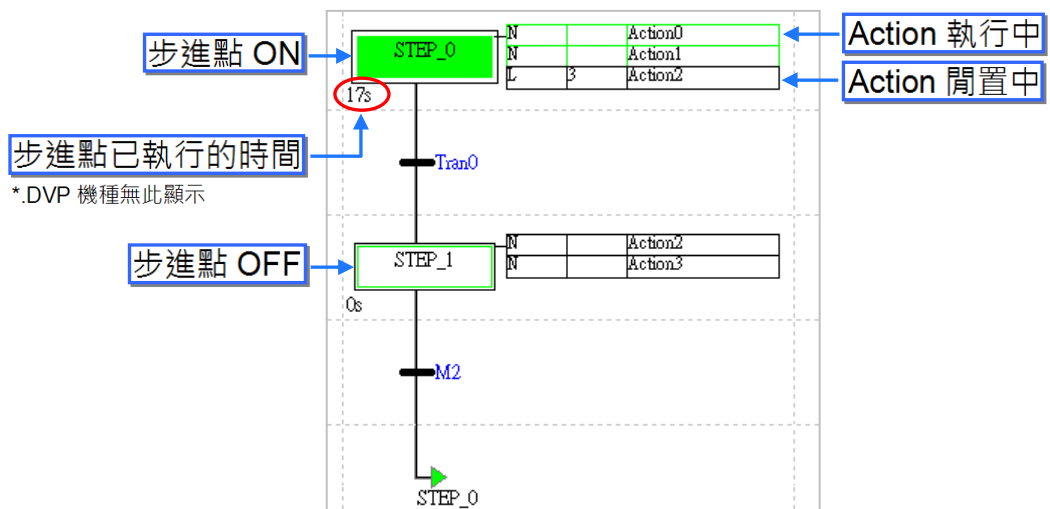
下圖為指令列表 (IL) 的程式監控畫面。



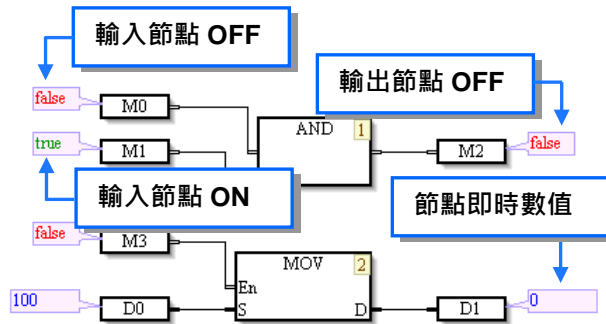
下圖為結構化語言 (ST) 的程式監控畫面。



下圖為循序功能圖 (SFC) 的程式監控畫面。除了 SFC 本身的圖面可進行監控之外，該 SFC 所屬的 Actions 與 Transitions 也都可進行監控。

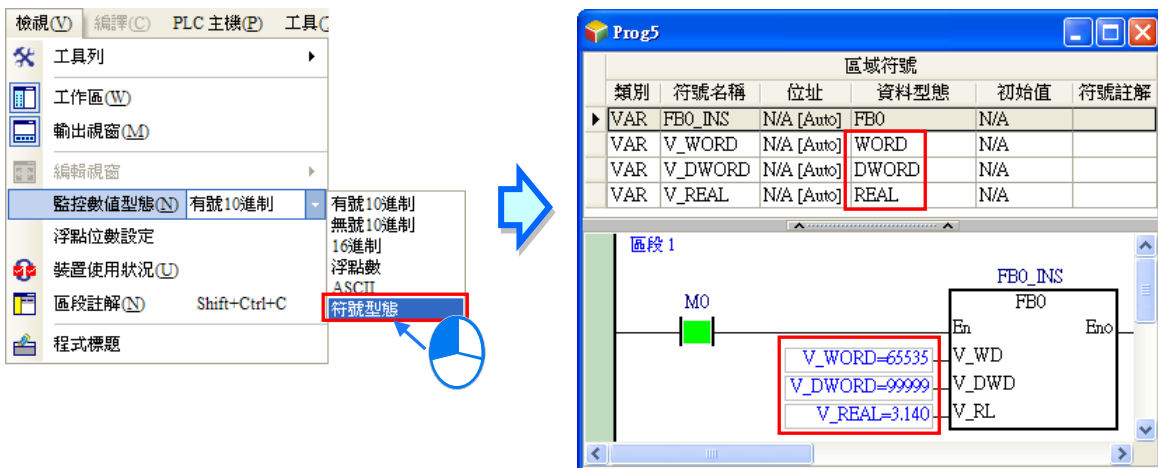


下圖為連續功能圖 (CFC) 的程式監控畫面。

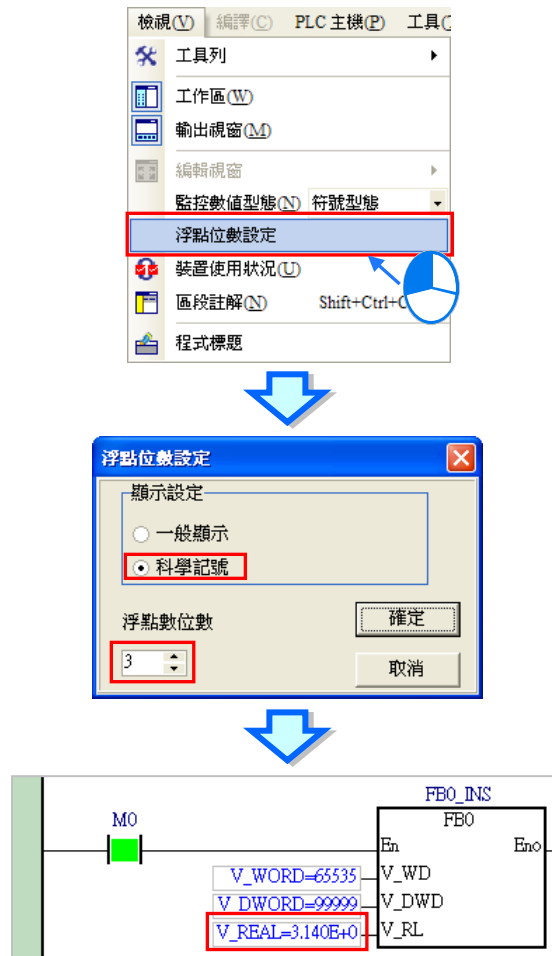


如上述各種語言的程式監控畫面，除了可觀察接點狀態之外，同時還可監控暫存器的即時值，而為了便於檢視，我們還可設定監控數值的顯示格式。

設定時，請於功能工具列中點選 **檢視 (V) > 監控數值型態 (N)**，之後於此下拉選擇數值格式，而當選擇「符號型態」時，監控數值便會依據變數符號所宣告的資料型態而自動變更顯示格式。




於功能工具列中點選 **檢視 (V)** > **浮點位數設定**，之後可設定浮點數的顯示格式。

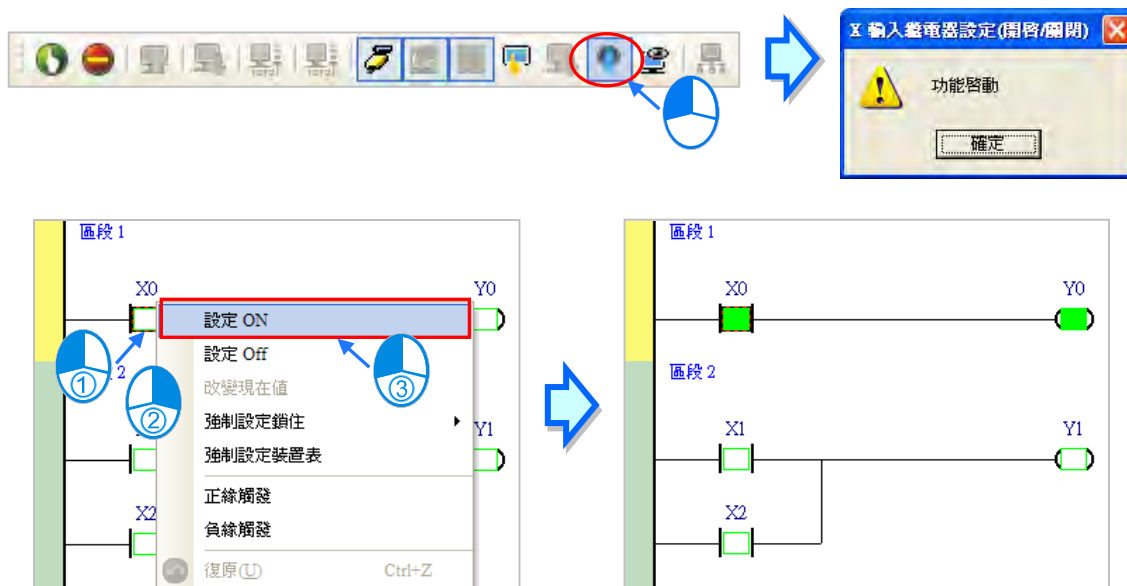


18.1.3 線上模式下變更 X 裝置的狀態

在「程式監控」的模式下，使用者可直接於程式畫面上變更接點的狀態，不過根據主機的設計，部份的機種可能不允許使用者於線上模式下直接變更 **X 裝置** 的狀態，當企圖變更 **X 裝置** 的狀態時，畫面將會顯示通訊錯誤的訊息。相關細節請參考各機種的使用說明。



若測試上有須要對 **X 裝置** 直接進行操作時，請於圖示工具列中點下  圖示，之後可開啟 **X 裝置** 的設定的功能；不過當此功能啟動後，外部的實際輸入將無法再更新 **X 接點** 的狀態。



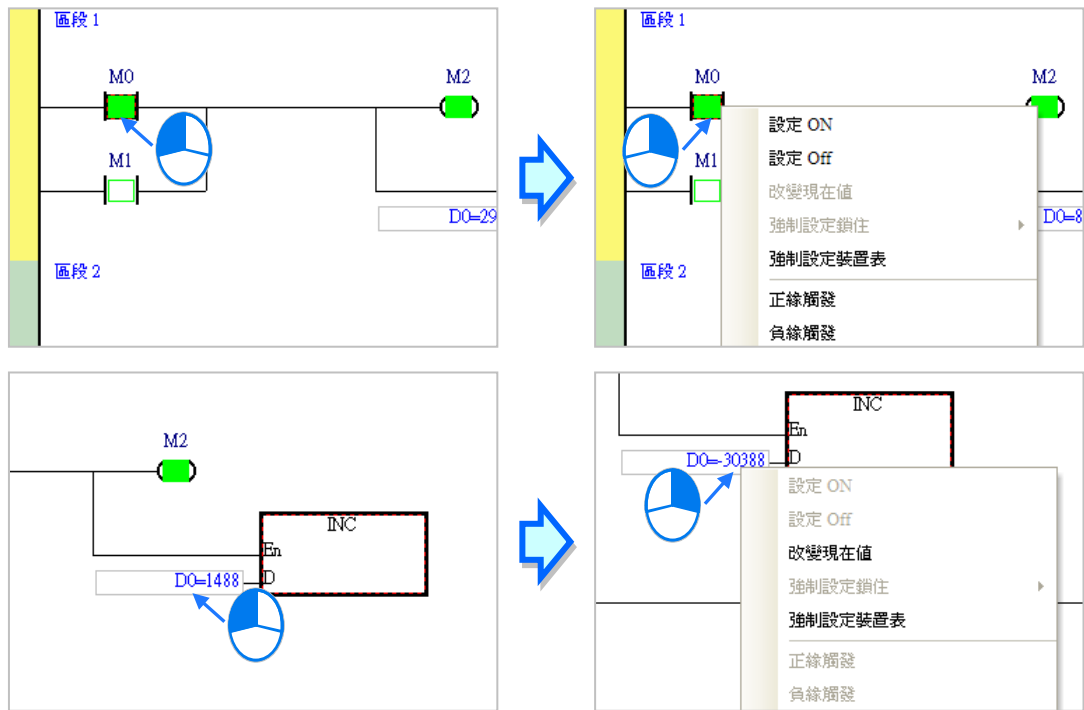
在離開線上模式後，設定 **X 裝置** 的功能並不會自動解除，因此在離開線上模式之前請確認是否須先將其解除，以免造成實際運作時，主機無法依據外部的輸入來更新 **X 接點** 的狀態。



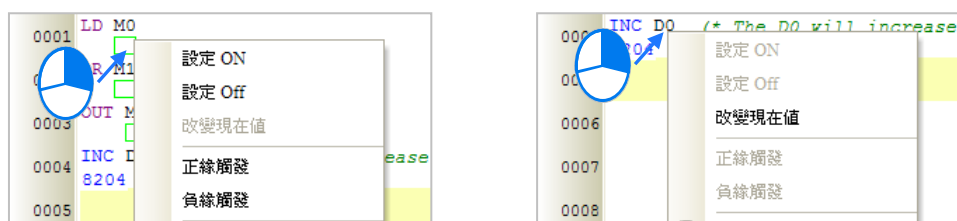
⚠ 將變更接點狀態前，請務必確認所做的操作不會影響系統運作，或造成系統與人員的傷害。

18.1.4 程式監控的線上操作

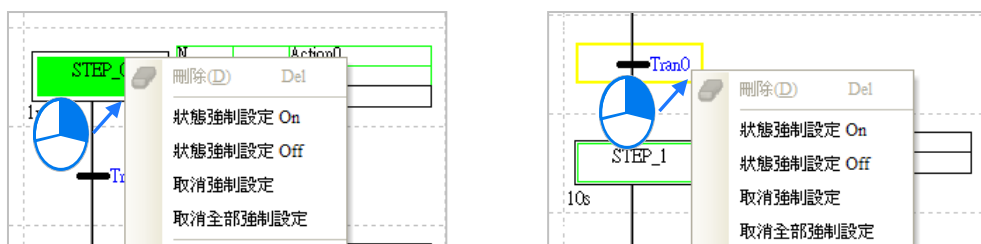
在「程式監控」的模式下，使用者可直接於程式畫面上變更接點的狀態或暫存器的內容值，藉此可進行除錯或測試的工作；而操作時，若是在 LD 或 FBD 的程式畫面中，請將欲先操作的裝置或指令運算元上點擊滑鼠左鍵，接著按下滑鼠右鍵，之後便可藉由快捷選單中來進行相關的操作；不過依據編程語言的不同，於該環境下可操作的項目可能會有些許差異。



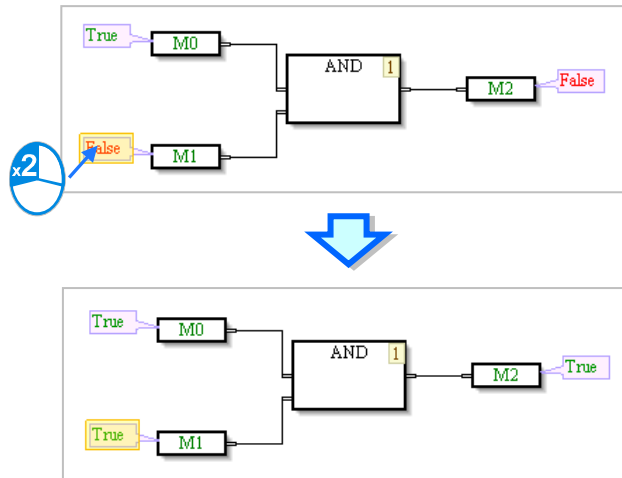
若在 IL 或 ST 的程式畫面中，請直接在欲操作的運算元上點擊右鍵，之後便可由快捷選單進行操作。



在 SFC 的圖面中，請直接於步進點或轉換點上點擊滑鼠右鍵，之後可由快捷選單來對其狀態進行 ON/OFF 的設定操作。



在 CFC 的圖面中，請直接於顯示的狀態上雙擊滑鼠左鍵，便會在 ON/OFF 的狀態中切換。



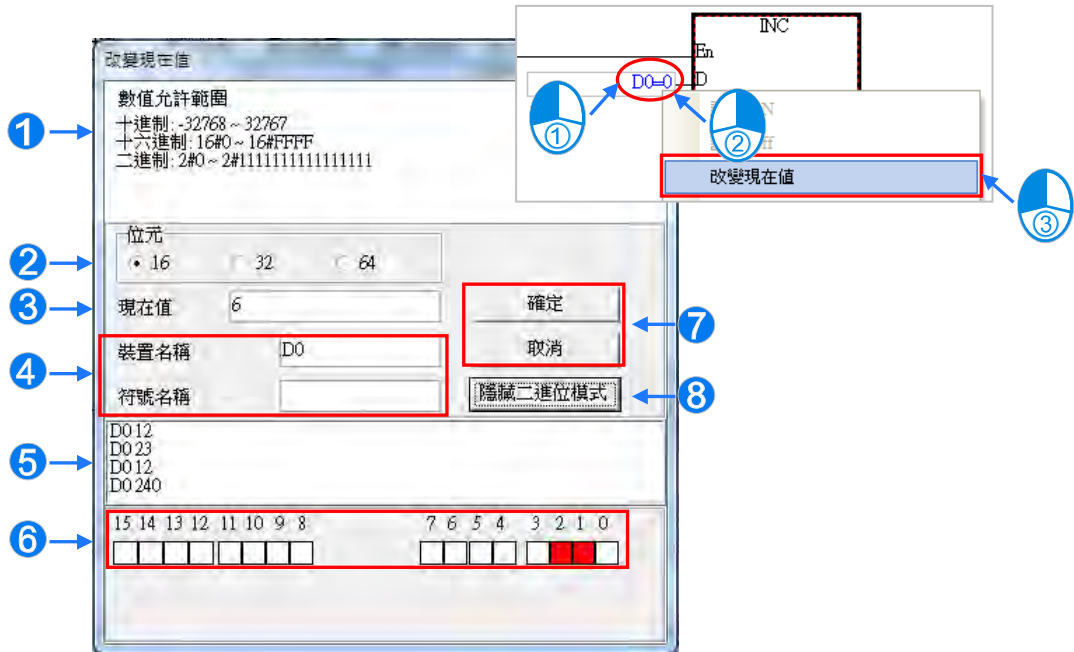
關於各操作項的說明簡述，請參考下表。

操作項目	說明
設定 ON	將點選的接點狀態設置為 ON。
設定 OFF	將點選的接點狀態設置為 OFF。
正緣觸發	不管被點選的接點狀態為何，系統皆會將該接點先設為 OFF 之後，再設為 ON。
負緣觸發	不管被點選的接點狀態為何，系統皆會將該接點先設為 ON 之後，再設為 OFF。
改變現在值	執行此功能後，使用者可於輸入視窗中設定點選運算元的內容值。
強制設定鎖住	將 X 或 Y 接點強制鎖定在 ON 或 OFF 的狀態。
強制設定裝置表	利用表格的方式，批次對 X 或 Y 接點的強制鎖定 ON/OFF 狀態進行管理。

! 將變更裝置狀態前，請務必確認所做的操作不會影響系統運作，或造成系統與人員的傷害。

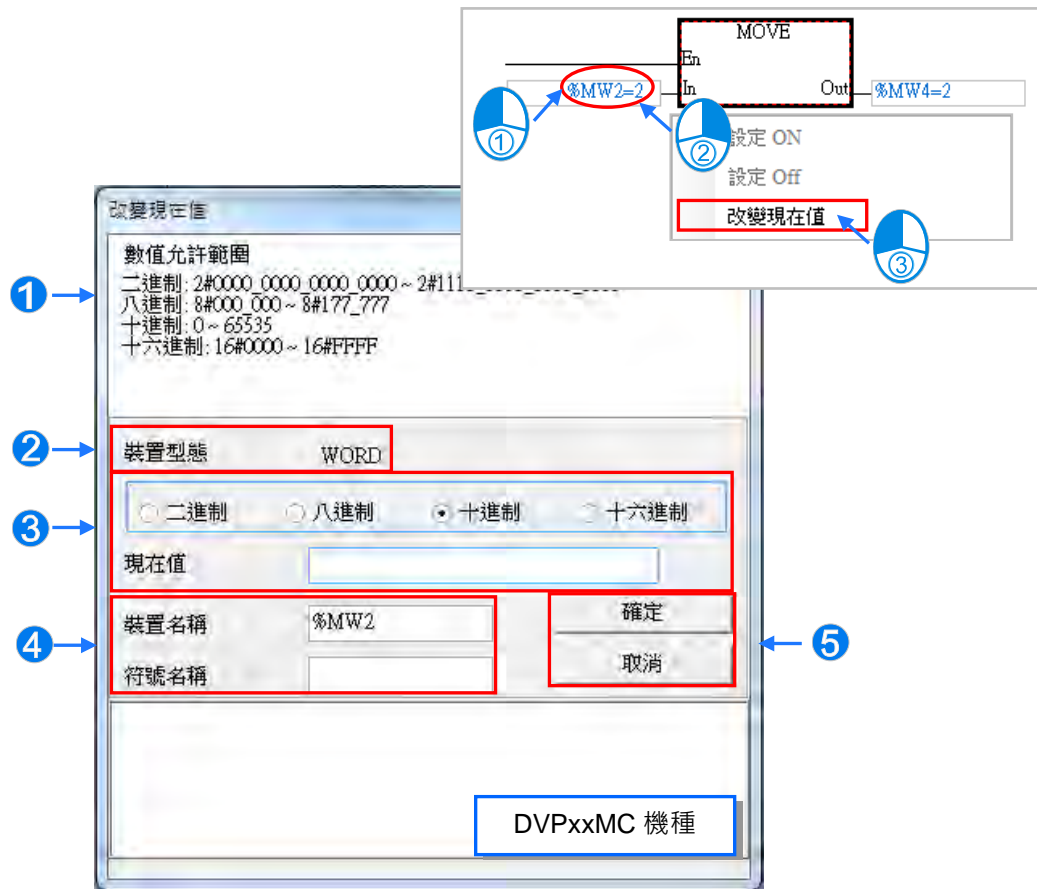
● 改變現在值

點選某一運算元後按下滑鼠右鍵，接著於快捷選單中點選「改變現在值」，之後便可即時由下圖的視窗中設定該運算元的內容值。



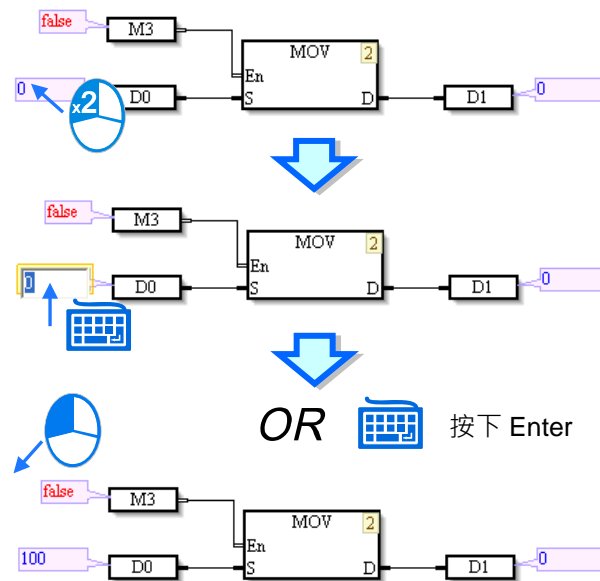
- ① 提示訊息。
- ② 於此選擇輸入的數值格式為 16 位元或 32 位元或 64 位元。(不同機種，支援選擇內容不同)
- ③ 於此輸入欲變更的數值。
- ④ 欲變更現在值的裝置名稱或變數符號名稱。
- ⑤ 數值變更記錄。(顯示格式：裝置名稱 變更數值)。
- ⑥ 當開啟二進位模式時，用滑鼠便可直接於此處設定各個 Bit 的 ON/OFF 狀態。
- ⑦ 按下「確定」鍵可套用設定值，按下「取消」鍵則可關閉此設定視窗。
- ⑧ 點擊此鍵可開啟或隱藏二進位模式。

若是使用者選擇是 DVPxxMC 機種，則會出現以下圖示。



- ① 提示訊息。
- ② 此處顯示目前裝置的型態。
- ③ 於此輸入欲變更的數值，而輸入格式可為二進制、八進制、十進制或十六進制。
- ④ 欲變更現在值的裝置名稱或變數符號名稱。
- ⑤ 按下「確定」鍵可套用設定值，按下「取消」鍵則可關閉此設定視窗。

關於 CFC 語言，改變現在值只要在顯示目前數值的位置上雙擊滑鼠左鍵，數值便會呈現編輯狀態，輸入欲改變的數值後，以滑鼠點擊空白處或按下鍵盤的 Enter 鍵，便完成數值的輸入。



● 強制設定鎖住

強制鎖定 ON/OFF 的功能僅可使用於 X 或 Y 接點，其作用在於將 X 或 Y 接點鎖定在某個狀態上；當某個 X 接點被鎖定後，實際外部輸入的狀況將無法改變該接點的狀態；而同樣的，當某個 Y 接點被鎖定後，無論程式內部的運行結果為何，最後 Y 接點的實際輸出仍會依據該接點被鎖定的狀態。

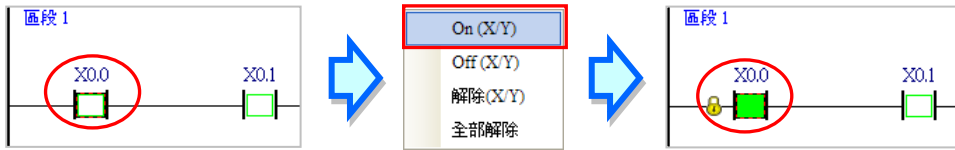
如下圖，因接點 Y0.0 被強制鎖定為 ON，因此即使 M0 為 OFF，Y0.0 還是會固定在 ON 的狀態，且處於強制鎖定 ON/OFF 狀態的接點，在該接點圖示的旁邊會出現一個大鎖的符號。



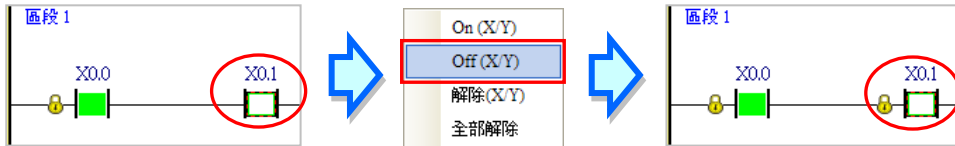
設定時，請先於欲操作的 X 或 Y 接點上點擊滑鼠左鍵，接著按下滑鼠右鍵並於快捷選單中點選「強制設定鎖住」，之後便可透過其下階的項目來進行相關的操作。



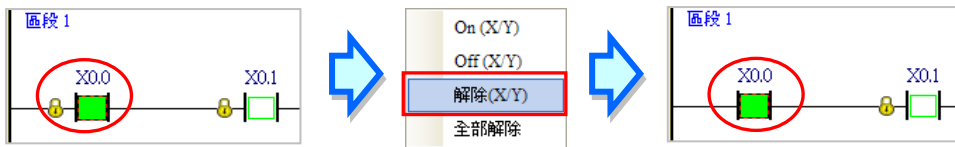
➤ **On (X/Y)** : 將點選的 X 或 Y 接點鎖定在 ON 的狀態。



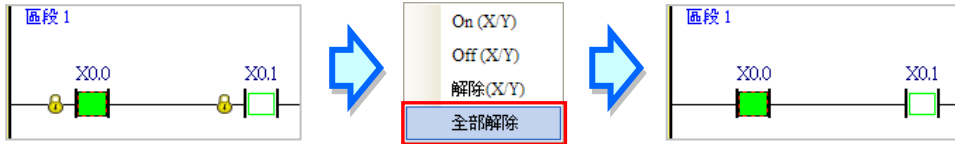
➤ **Off (X/Y)** : 將點選的 X 或 Y 接點鎖定在 OFF 的狀態。



➤ **解除 (X/Y)** : 解除點選接點的鎖定狀態。



➤ **全部解除** : 解除所有接點的鎖定狀態。



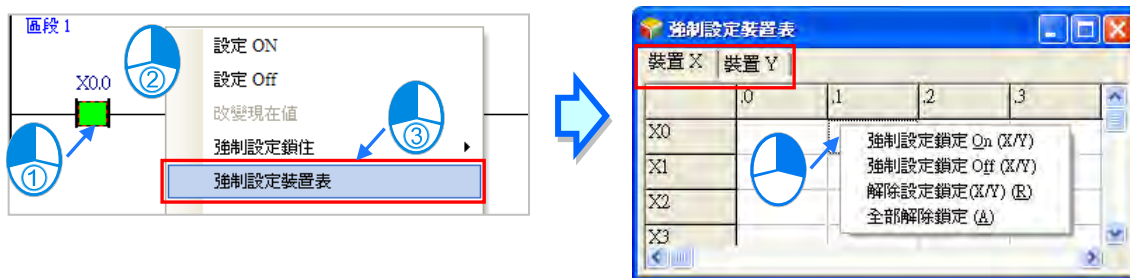
*1.不同機種對於強制鎖定 ON/OFF 的處理機制及點數限制可能不同，設定前請先參考各機種的相關使用說明。

*2.在離開線上模式後，各接點的強制鎖定狀態將不會自動解除，請務必確認目前的狀態是否會造成影响。

● **強制設定裝置表**

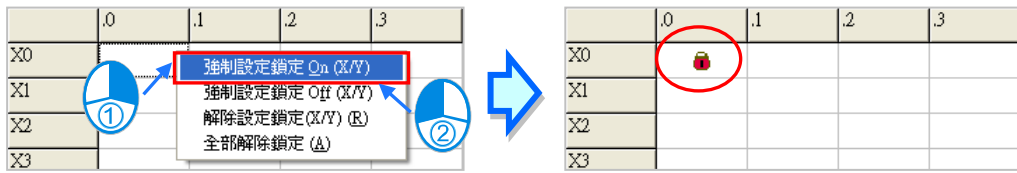
此功能可利用表格的方式，批次對 X 或 Y 接點的**強制鎖定 ON/OFF** 狀態進行管理。

設定時，請於程式畫面中按下滑鼠右鍵，之後於快捷選單中點選「**強制設定裝置表**」；而開啟表格後，請先切換上方的標籤選擇要設定的裝置種類，接著於表格中點選欲設定的接點並按下滑鼠右鍵，之後便可透過快捷選單中的項目來進行相關操作。

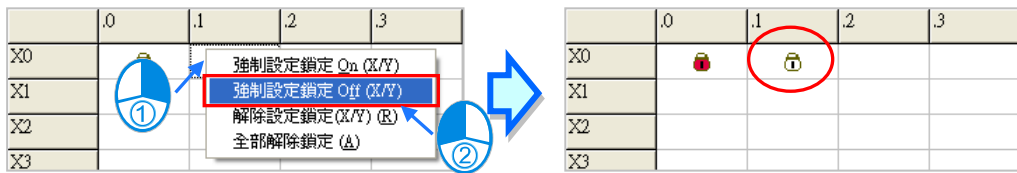


18

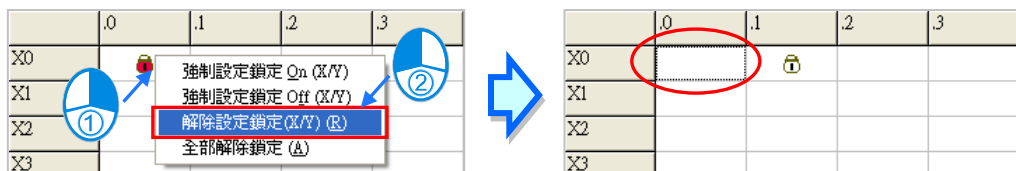
- 強制設定鎖定 On (X/Y) : 將點選的接點強制設定為 ON 。



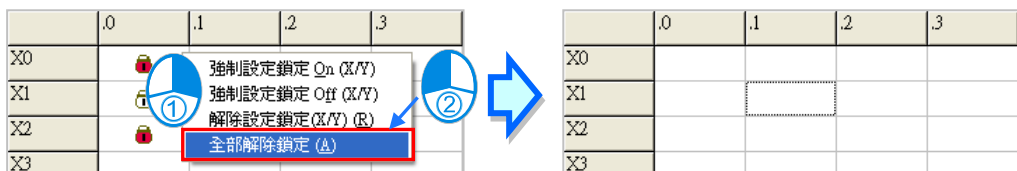
- 強制設定鎖定 Off (X/Y) : 將點選的接點強制設定為 OFF 。



- 解除設定鎖定 (X/Y) : 解除點選接點的強制狀態 。



- 全部解除鎖定 : 不分 X 或 Y 裝置 , 所有接點的強制狀態將會全部解除 。



*1.不同機種對於強制鎖定 ON/OFF 的處理機制及點數限制可能不同，設定前請先參考各機種的相關使用說明。

*2.在離開線上模式後，各接點的強制鎖定狀態將不會自動解除，請務必確認目前的狀態是否會造成影響。

18.1.5 裝置監控表

當進入線上模式後，不論是「程式監控」或是「裝置監控」模式，使用者皆可透過**監控表**的方式來對程式中的裝置或是變數符號進行監控；而在 ISPSOft 中，每個專案都允許建立一個以上的監控表，而每個監控表則至多可建立 100 個監控項目，使用者可自行對所要監控的項目進行歸類管理，且監控表的建立與管理皆可在離線的狀態下完成，而所建立的監控表亦將會隨著專案一起儲存，下次再開啟專案時便無須重新建立。

開始新增一個監控表的方法有如下三種。

- 方法一

於功能工具列中點選 **PLC 主機 (P)** > **建立裝置監控表 (I)**。




- 方法二

在專案管理區點選「裝置監控表」項目後按下滑鼠右鍵，之後於選單中點選「新增」。

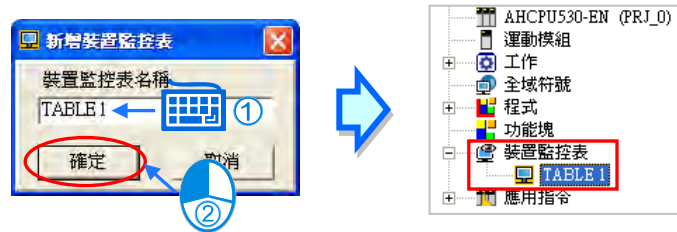


- 方法三

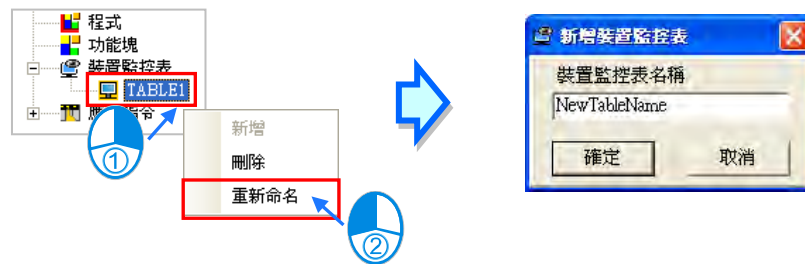
直接於圖示工具列中點選  圖示。



依據上述的方式執行之後，畫面便會出現一個彈出視窗，於該視窗輸入監控表的名稱後按下「**確定**」，之後於專案管理區的「**裝置監控表**」項目下便會產生一個物件，並自動於編輯工作區開啟該監控表的管理視窗，而後續便可於此視窗中進行監控項目的管理。



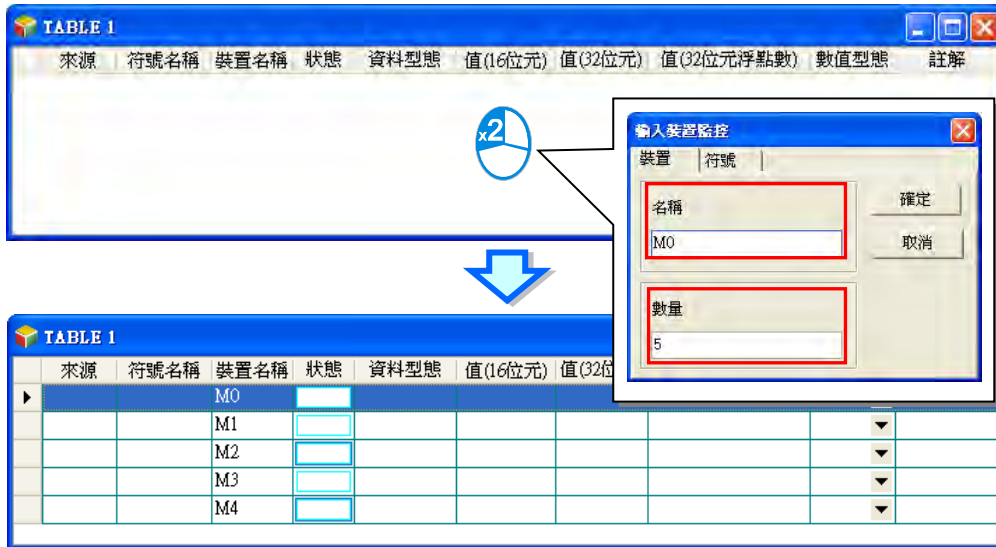
欲刪除監控表時，請於專案管理區中點選欲刪除的監控表項目後按下滑鼠右鍵，之後於快捷選單中選擇「**刪除**」即可。當要重新命名已建立好的監控表時，請於專案管理區中點選欲重新命名的監控表項目後按下滑鼠右鍵，接著於快捷選單中選擇「**重新命名**」，之後便可變更該監控表的名稱。



在 ISPSoft 中允許同時開啟多個監控表視窗，而要開啟時請移至專案管理區中，直接在欲開啟的監控表項目上雙擊滑鼠左鍵即可。



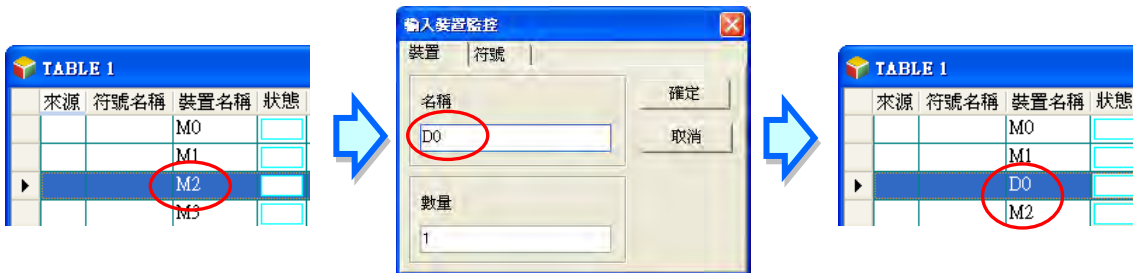
當要在監控表中加入監控項目時，請於**監控表**的空白處雙擊滑鼠左鍵，或直接以鍵盤輸入裝置名稱，之後畫面便會出現設定視窗，直接於該視窗中輸入啟始的裝置位址與欲加入的裝置數量即可，不過須注意每個監控表至多只可建立 100 個監控項目。



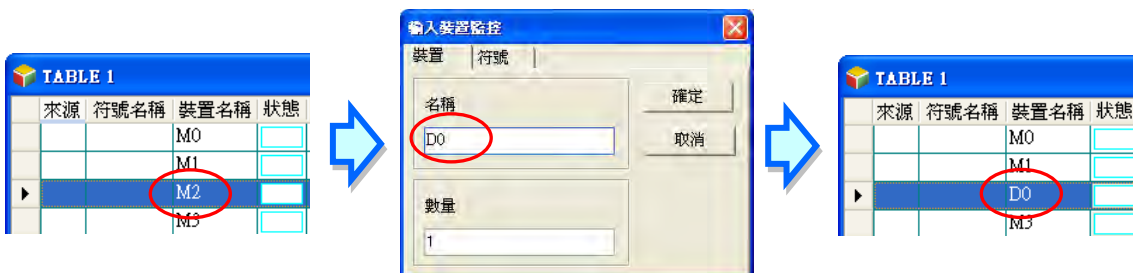
切換鍵盤的【Insert】鍵可選擇**插入**或**取代**兩種模式，且會顯示於 ISPSOft 的狀態列中。



當處於**插入**模式時，新增的監控項目會插入於目前在監控表中被選取的項目上方。

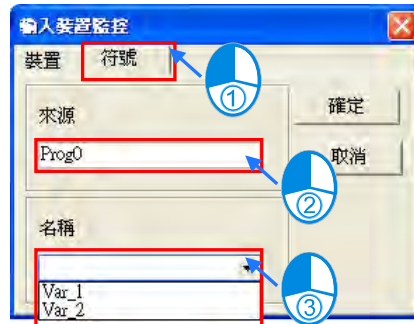


但若處於**取代**模式時，新增的監控項目則會取代目前被選取的項目。

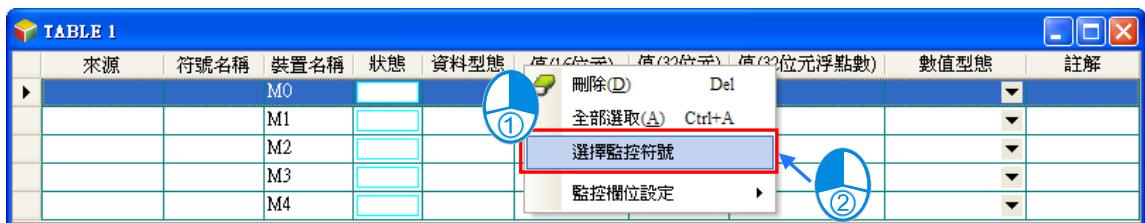


18

在監控表中除了可加入裝置之外，亦可直接於監控表中加入變數符號。

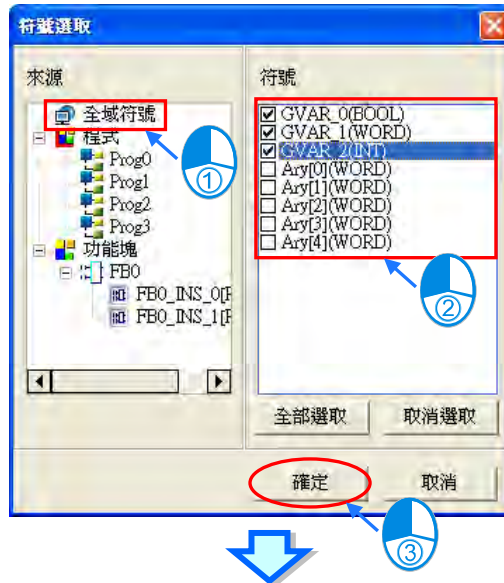


操作時只需於監控表中按下滑鼠右鍵，之後於快捷選單中點擊「選擇監控符號」項目即可。



接著請於**符號選取視窗**的左側選擇符號表的來源，其來源可以是全域符號表或 POU 的區域符號表，其中若要選擇功能塊的區域符號時，必須先執行編譯並產生功能塊實例後，功能塊內部的變數符號才會被配置一個實體的記憶體位置，而此時才可對其進行監控，且因不同的功能塊實例會配置不同的記憶體位置，因此挑選功能塊內部的區域符號時必須點選功能塊實例的項目後才可進行選取。

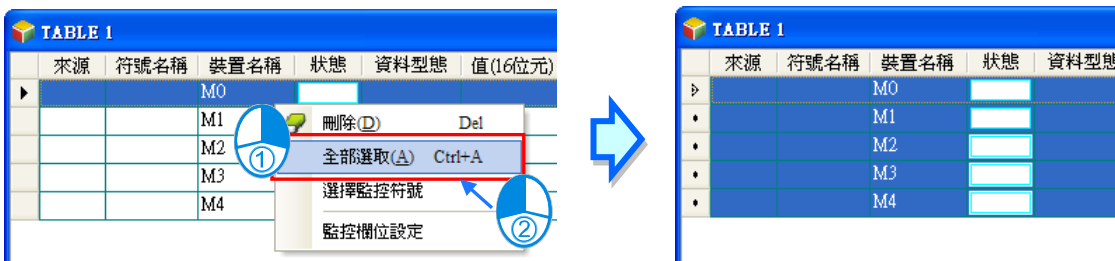
選取符號表的來源後，請再於符號選取視窗的右側勾選欲加入監控的項目，其中若有陣列符號時，清單中便會展開所有的陣列元素；勾選完畢後按下「確定」按鈕即可將所選項目加入監控表中，而之後可再繼續選擇其它的變數符號，當完成所有符號的加入之後，按下「取消」按鈕即可關閉視窗。



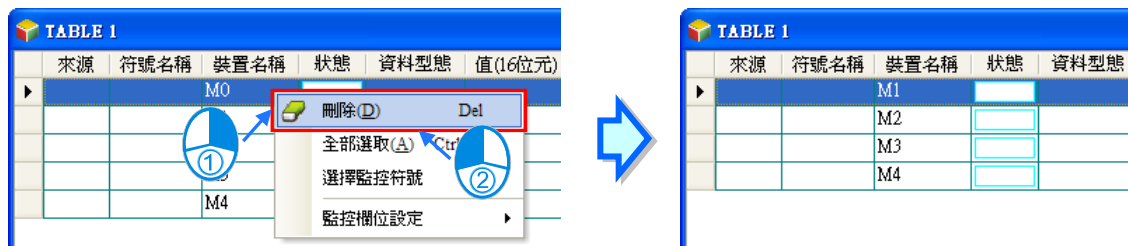
來源	符號名稱	裝置名稱	狀態	資料型態	值(16位元)	值(32位元)	值(32位元浮點數)	數值型態	註解
		M0	<input type="checkbox"/>						
		M1	<input type="checkbox"/>						
		M2	<input type="checkbox"/>						
		M3	<input type="checkbox"/>						
		M4	<input type="checkbox"/>						
Global Symbols	GVAR_0		<input type="checkbox"/>	BOOL					ERROR
Global Symbols	GVAR_1		<input type="checkbox"/>	WORD				有號數十進制	DATA
Global Symbols	GVAR_2		<input type="checkbox"/>	INT				有號數十進制	OFFSET

*.指標型態的變數符號無法進行監控。

當要刪除監控項目時，請先於監控表中點選欲刪除的項目，而選取時若先按住鍵盤的【Ctrl】鍵則可連續選取多個項目；若要選取連續範圍時，請先點選第一個項目後按住鍵盤的【Shift】鍵，之後再點選該範圍的最後一個項目便可將整個範圍選取起來；而若要選取全部的項目時，可直接於監控表中按下滑鼠右鍵，接著於快捷選單中點擊「全部選取 (A)」項目即可。



選取完欲刪除的項目後，請於該項目上點擊滑鼠右鍵，之後於快捷選單中點擊「刪除 (D)」項目，亦或是直接按下鍵盤的【DEL】鍵即可。



為了便於檢視，在使用時可將監控表中的部份欄位設為隱藏，而設定時請直接於監控表中按下滑鼠右鍵，接著於快捷選單中點擊「監控欄位設定」項目，之後可於其下階的選項中設定欲顯示的欄位，取消勾選代表該欄位將被隱藏。關於各欄位的相關說明請參考下述。

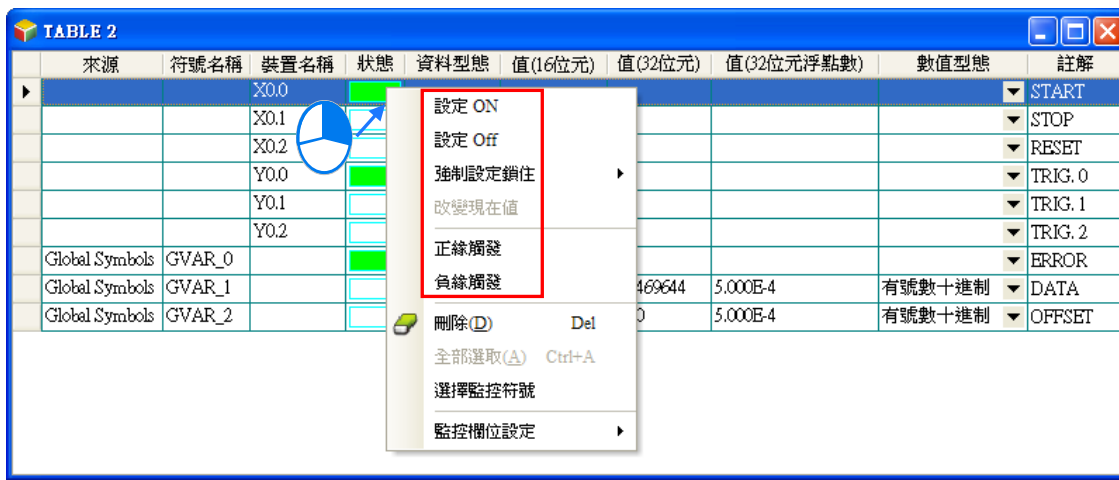


下表則為監控表中各欄位的相關說明。

監控表欄位	說明
來源	顯示變數符號的來源。
符號名稱	顯示變數符號的名稱。
裝置名稱	顯示監控的裝置名稱。
狀態	若監控的對象為位元或接點時，此欄位會顯示其 ON/OFF 狀態。
資料型態	若監控的對象為變數符號，則此欄位會顯示該變數符號的資料型態。
值 (8 位元)	實際連線監控時，此欄位會以 8 位元的型態來顯示監控值。
值 (16 位元)	實際連線監控時，此欄位會以 16 位元的型態來顯示監控值。
值 (32 位元)	實際連線監控時，此欄位會以 32 位元的型態來顯示監控值。
值 (64 位元)	實際連線監控時，此欄位會以 64 位元的型態來顯示監控值。
值 (32 位元浮點數)	實際連線監控時，此欄位會以 32 位元的浮點數型態來顯示監控值。

監控表欄位	說明
值 (64 位元浮點數)	實際連線監控時，此欄位會以 64 位元的浮點數型態來顯示監控值。
數值型態	於此下拉可選擇連線監控時所顯示的數值格式。
註解	顯示監控裝置的 裝置註解 或監控符號的 符號註解 。

當建立好一個監控表後，接著可於線上模式下透過監控表來監控表格中的各個項目，而如同「程式監控」一般，在線上模式下，於**監控表**中點選某個項目後下滑鼠右鍵，之後可利用快捷選單中的操作項來對監控項目的狀態或內容值進行變更。關於各種操作項目的相關說明請參考第 18.1.4 節。



補充說明

監控項目一旦加入監控表之後，後續若變更監控裝置的裝置註解時，請先關閉監控表視窗後再重新開啟以便取得更新；而若變更了變數符號的相關屬性或將其由符號表中移除時，該監控項目將可能無法自動更新，且後續亦將無法在線上監控時產生作用，此時請將其刪除後再重新加入監控表中。


18.1.6 線上編輯與線上更新

*.目前 DVPxxMC 機種不支援「線上編輯與線上更新」功能。

當系統處於「程式監控」的模式，且主機處於 RUN 的狀態下，使用者便可透過**線上編輯與線上更新**的功能來即時修改監控中的程式內容，而各種語言的編輯限制則請參考下表。

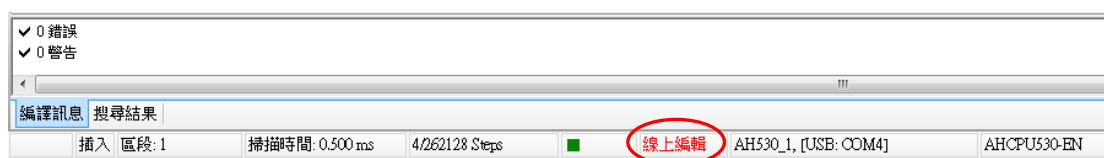
編程語言	說明
LD 語言	每次只可編輯一個程式區段，但允許新增區段，而一旦新增了一個區段後，可編輯的區域便會轉移至該新區段中，亦即無法再回頭編輯方才所編輯的區段。
FBD 語言	每次只可編輯一個程式區段，但允許新增區段，而一旦新增了一個區段後，可編輯的區域便會轉移至該新區段中，亦即無法再回頭編輯方才所編輯的區段。
IL 語言	沒有特別的行數限制。


編程語言	說明
ST 語言	沒有特別的行數限制。
SFC 語言	SFC 的圖形無法於線上編輯時進行修改，但可以修改 Actions 與 Transitions 內部的程式，其限制則依據所使用的編程語言而與上述相同。
CFC 語言	沒有特別的編輯限制。
共通限制	每次更新後只能編輯一個 POU。 線上編輯時無法修改或新增變數符號。

欲進行線上編輯時，請於「程式監控」的模式下，並確認主機處於 RUN 的狀態後，直接於圖示工具列中按下  圖示即可。




進入線上編輯之後，在 ISPSOft 的狀態列中便會顯示為「線上編輯」的狀態。




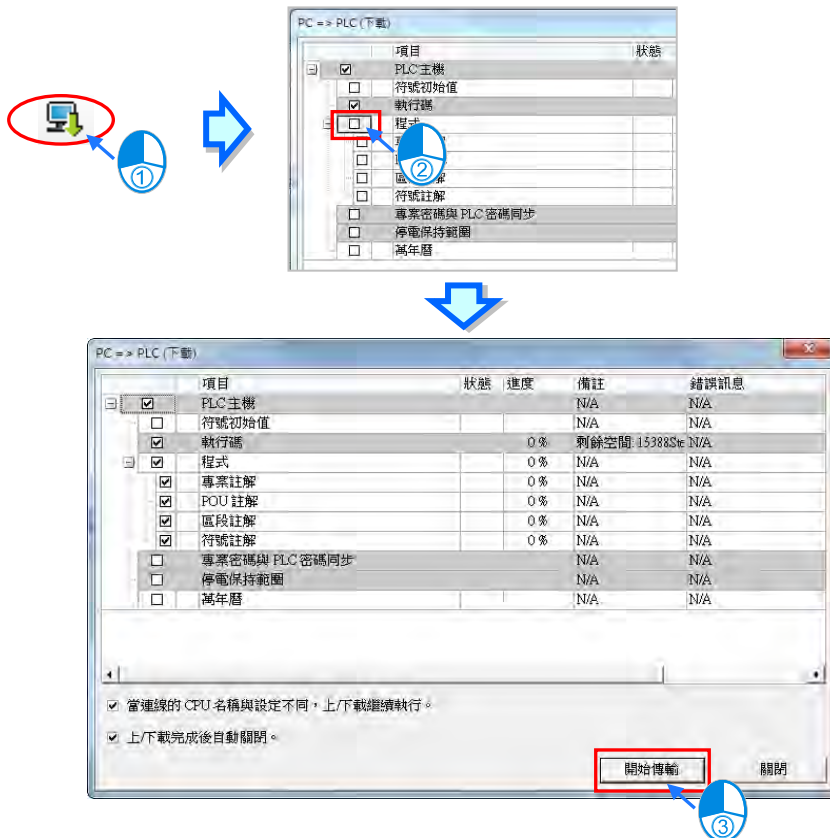
當完成程式的編輯並成功編譯之後，按下圖示工具列的  圖示便可將修改的程式下載至主機中進行更新，而若程式尚未完成編譯時，系統便會主動要求使用者進行編譯的工作。



若是使用 DVP 機種，在點擊圖示後，做下載時，出現「建議在此操作後重新下載程式」提示視窗，接著點選「OK」，接著點選「是(Y)」，此時如果有設 PLC 密碼就會彈出詢問密碼視窗，可以勾選**暫時儲存密碼**，這樣直到專案重啟之前使用線上更新就不會再次詢問密碼，接著點選「確定」，就會將目前編譯後的程式執行碼下載至 PLC。



若要将 PLC 原始碼下載至 PLC，則需再點擊，勾選「程式」選項，將 PLC 原始碼下載進 PLC，才可同步 PLC 原始碼。





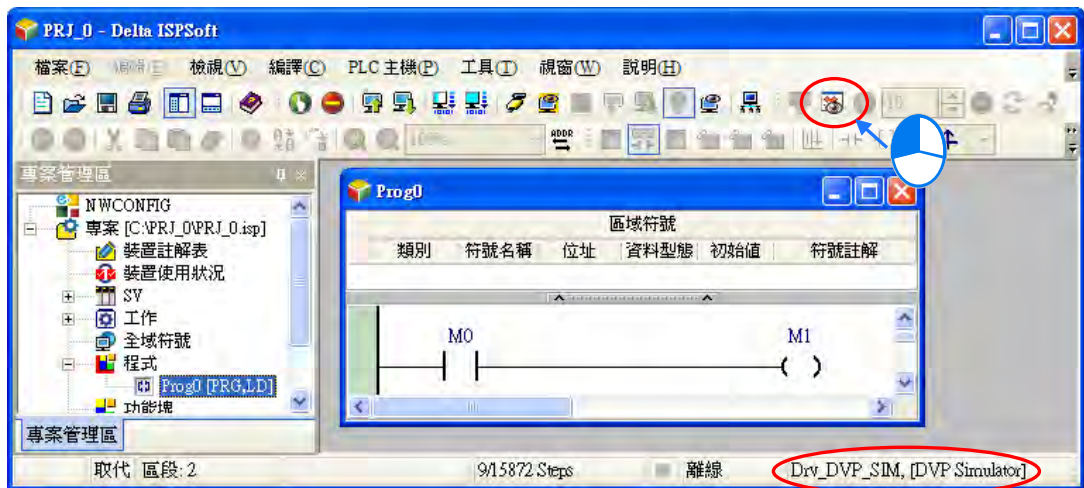
18.2 DVP 機種的除錯模式

ISPSOft 針對 DVP 機種有提供**除錯模式**的功能，而其內容包括了中斷點操作、單步執行、限定次數執行...等基本的除錯工具；藉由這些工具，使用者可方便的針對程序的部份進行偵錯與測試。而在開始進入 DVP 的除錯模式之前，請先留意下列兩點的注意事項。

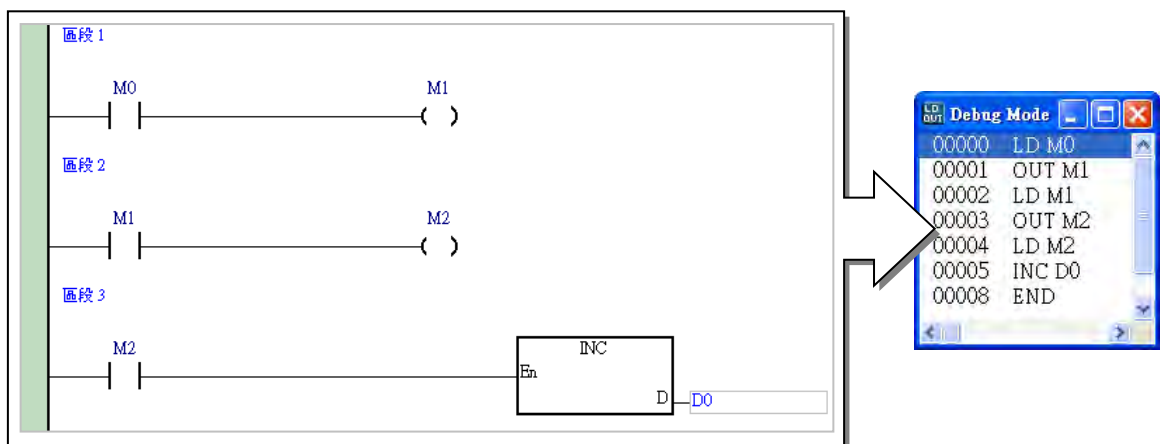
- (a) DVP 的除錯模式僅能在模擬器的連線模式下執行。
- (b) 線上模式下無法啟動**除錯模式**，但仍可在進入**除錯模式**之後啟動**裝置監控**的功能。

18.2.1 啟動 DVP 的除錯模式

首先請將系統設為 **DVP 模擬器**的連線模式，操作細節請參考第 2.4 節的相關說明，之後於圖示工具列中按下  圖示即可進入除錯模式，而此時畫面便會出現一個除錯視窗，所有的除錯工作都將在該視窗中進行。後續若要離開除錯模式時，只要再次按下  圖示並恢復為未按下的狀態即可。

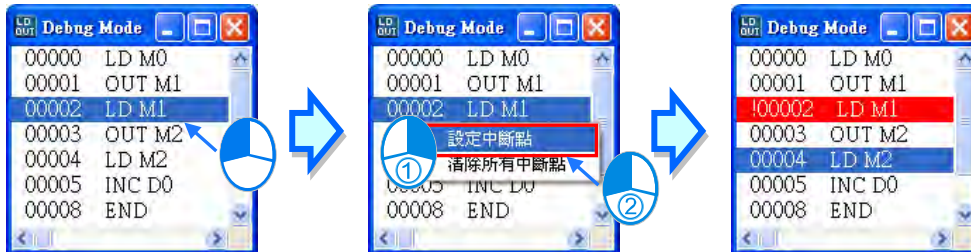


DVP 的除錯視窗是將程式編譯後而重新以 IL 語言來所呈現，其中藍色背景的行為程式目前暫停的位置，而暫停的位置則是尚未執行的程序。

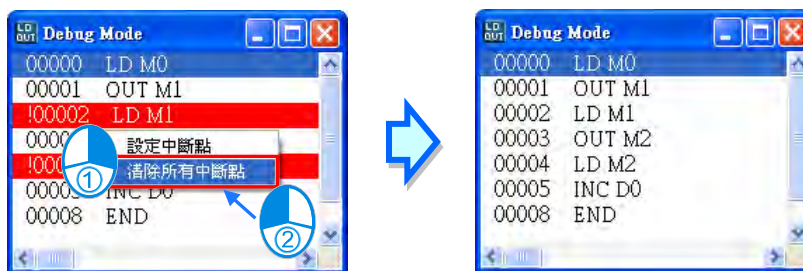


18.2.2 設定與清除中斷點

於除錯視窗中欲設定中斷點的行列上按下滑鼠右鍵，並於快捷選單中點選「設定中斷點」項目即可加入一個中斷點，而在同一行號上再重覆一次相同操作則可取消中斷點的設置。在除錯視窗中，加入中斷點的行號前方會出現一個「！」的符號，且該行列的背景會以紅色來顯示。





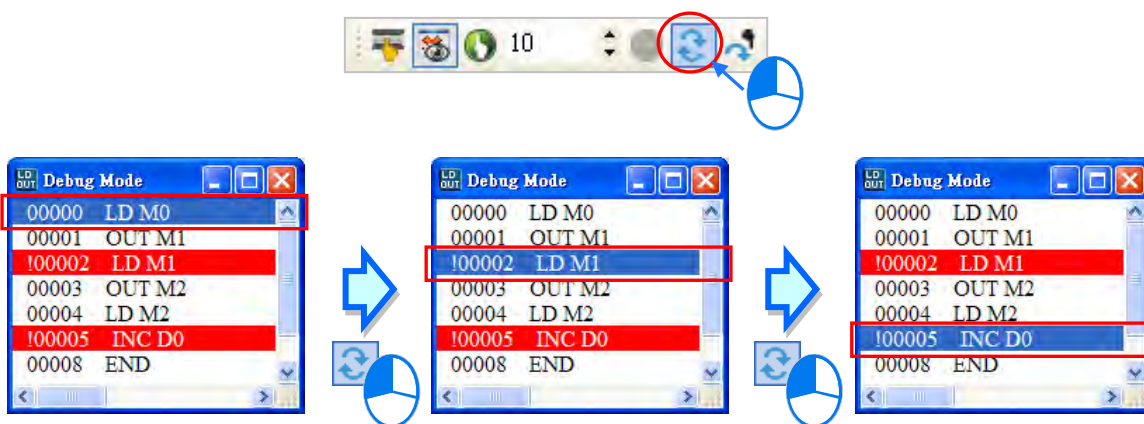
當要清除所有的中斷點設置時，請於除錯視窗中按下滑鼠右鍵，之後於快捷選單中點選「清除所有中斷點」項目即可。



18.2.3 除錯模式下的程式執行

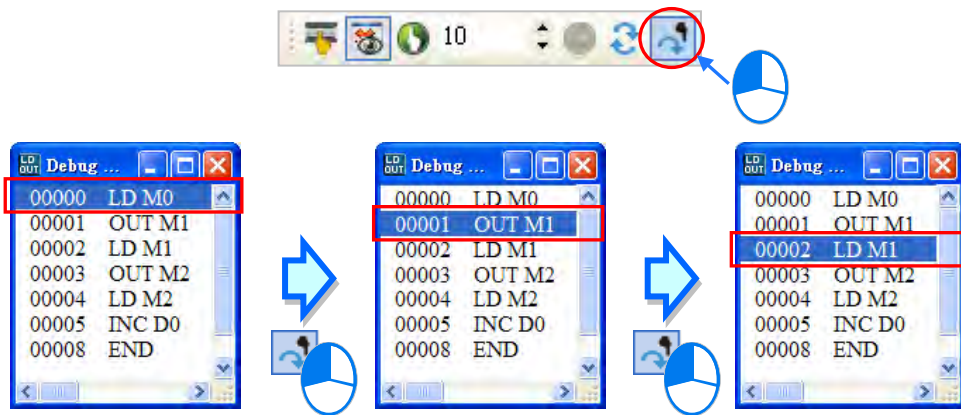
● 連續執行

按下圖示工具列的  圖示便會開始連續執行到設定中斷點的程式為止，此時再次按下  圖示便會繼續執行到下一個中斷點，而若未設定任何中斷點時，程式便會不中斷的來回掃描執行。




● 單步執行

在程式執行動作暫停的狀況下，每按下一次圖示工具列的  圖示可執行一步的程序。




● 限定次數執行


利用此功能可讓程式在掃描週期的次數到達限定次數時自動停止，且過程中完全不受中斷點的影響。執行前請先於圖示工具列的 欄位上設定執行次數，至多可設定至 32767，完成後按下該欄位左側的  圖示即可開始執行。

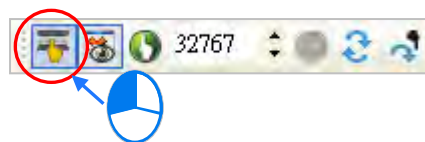


● 停止與重置

在執行過程中，只要按下同一工具列的  圖示後便會停止執行動作。



而按下圖示工具列的  圖示可重置所有除錯模式下的執行步序及狀態值，而下次再執行時便會由程式的開頭重新執行。




*.進入 DVP 的除錯模式後，透過監控表仍可於除錯模式的執行過程中監控各裝置的變化狀況。

18.3 AH/AS 機種的除錯模式

18.3.1 進入除錯模式

AH/AS 系列機種的除錯模式可支援各種編程語言，(AH560 備援系列機種，在達成同步後，不支援除錯模式)，且雖然不同語言在畫面的呈現上會有所差異，但大致上的操作是相同的；不過在 SFC 中，除錯模式的操作僅可在 Actions 當中進行，且該 Actions 所屬的步進點必須處於執行狀態；此外，在除錯模式下，若 Transitions 的動作無法產生步進點的轉移時，使用者必須自行於 SFC 的圖面中切換轉移點的 ON/OFF 狀態來轉移步進點。



先將主機切換至 RUN 並啟動「程式監控」，之後再按下圖示工具列的  圖示即可進入除錯模式。

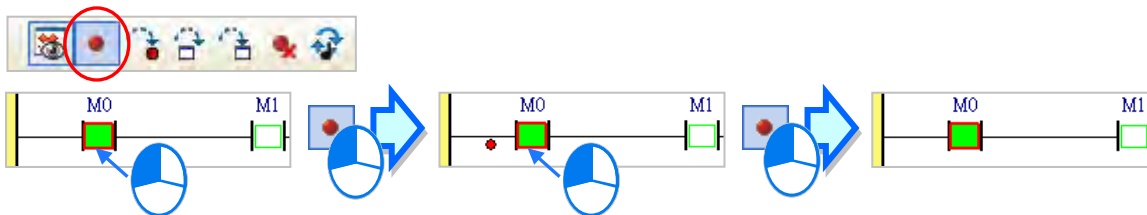


若在 ISPSOft 啟動除錯模式之前，主機本身就已經在除錯模式的狀態中 (可由狀態列得知)，此時便無法再讓目前的 ISPSOft 畫面切換至除錯模式；其原因有可能是因為上次未正常退出除錯模式或者是網路上的其他使用者正在對該主機進行除錯的工作；若要重置主機的除錯模式時，可先將該主機的狀態切換至 STOP，接著再切換至 RUN，然後便可依據上述的說明重新啟動除錯模式。

取代 區段:1 掃描時間: 1.0ms 195/262128 Steps  除錯 Drv_USB, [USB: COM9] AHCPU530-EN


18.3.2 中斷點操作

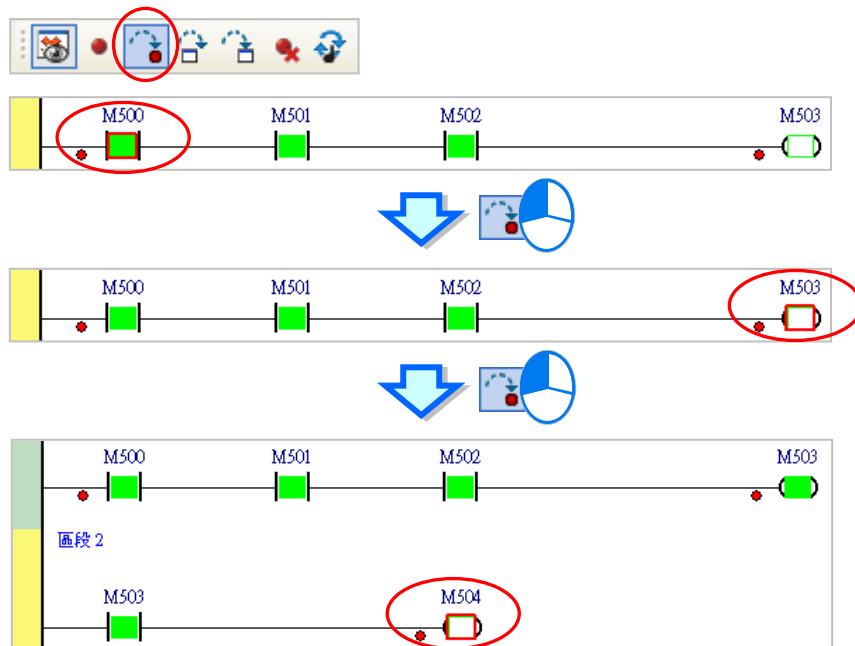
在 AH500 系列機種中，一個專案程式 (包含所有 POU) 最多可設定至 10 個中斷點。欲設置中斷點時，請先於程式畫面中點選欲設置的位置，之後按下圖示工具列的  圖示便可加入中斷點；同樣的，在已設定中斷點的位置上再次按下  圖示則可將該中斷點取消。







當要清除所有的中斷點時，請直接於圖示工具列中按下  圖示即可。

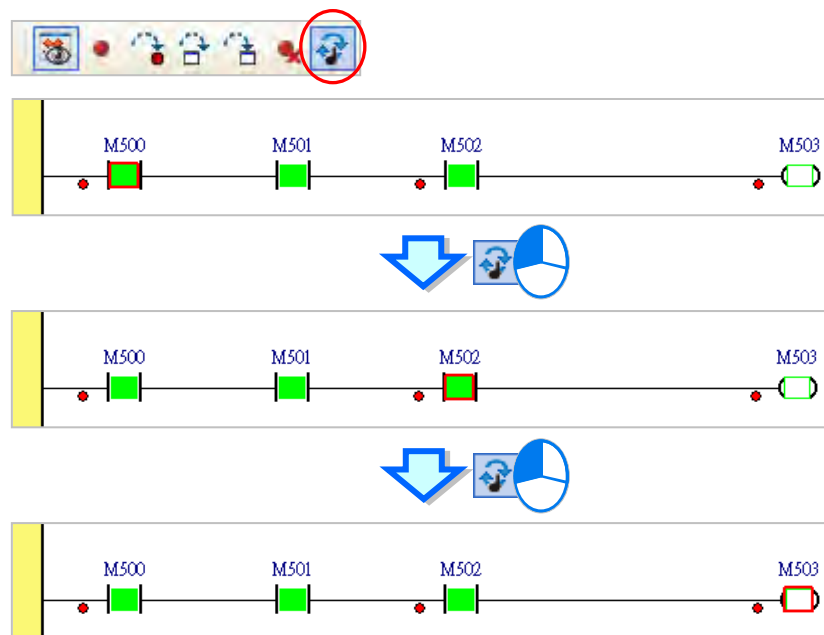


當程式中有設置中斷點時，每一次按下圖示工具列的  圖示，系統便會自動執行程式至下一個中斷點後暫停，而暫停的位置則是尚未執行的程序。







18.3.3 連續執行

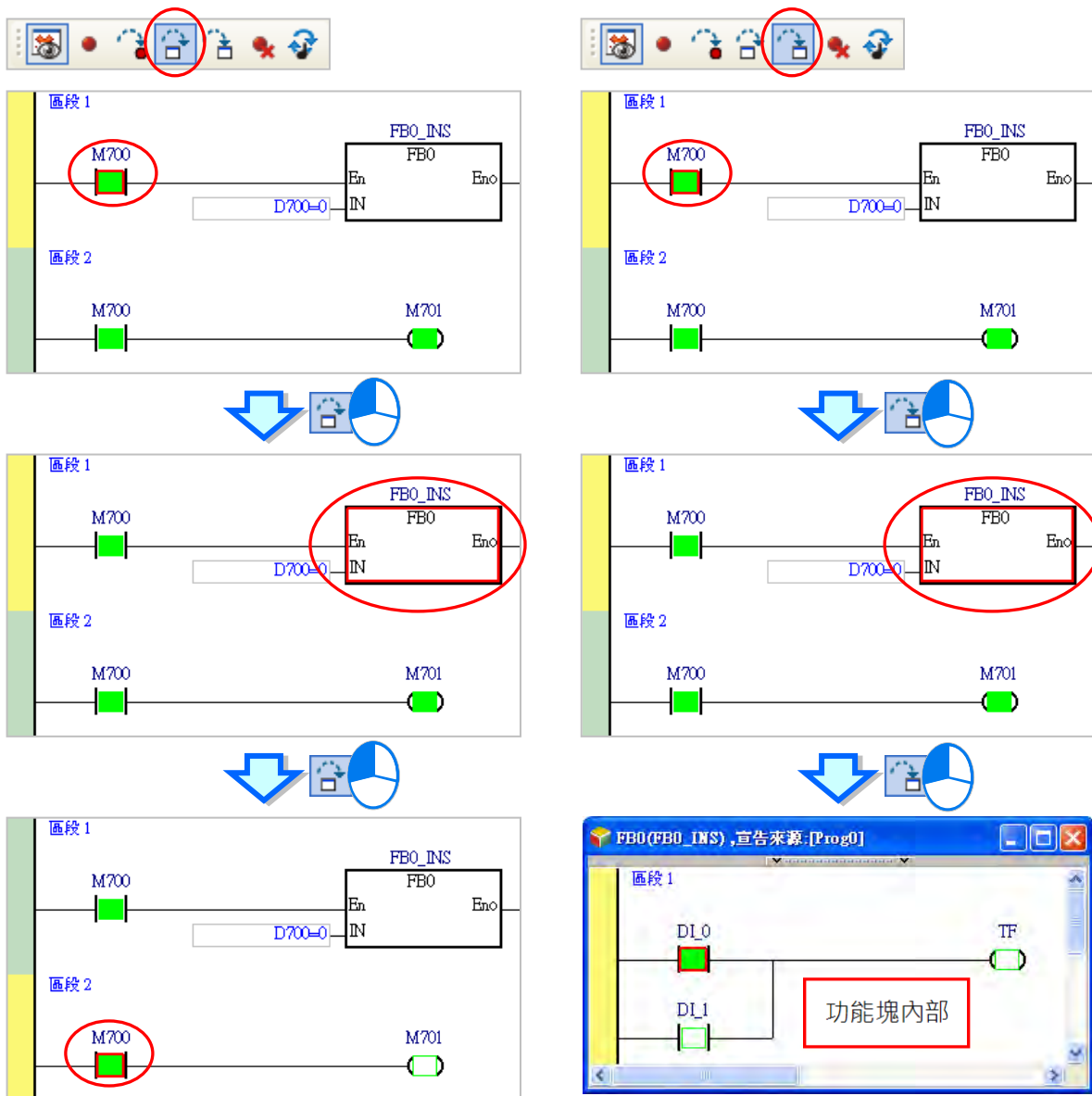
當按下圖示工具列的  圖示後，系統便會開始連續執行到設定中斷點的程序暫停，此時再次按下  圖示便會繼續執行到下一個中斷點；但若未設置任何中斷點時，按下  圖示後程式便會不中斷的來回掃描執行，而在執行過程中，再把  圖示按起便可停止程式的執行。



18.3.4 單步執行

每一次於圖示工具列中按下  圖示或  圖示時，系統便會在執行完目前位置的程序之後，在下一個程序之前暫停；至於兩個圖示的功能差別在於遇上**功能塊**時，若按下  圖示便會直接執行完功能塊的功能，並於下一個程序前暫停；而按下  圖示則會進入功能塊當中，並於功能塊內部繼續進行單步執行。不過倘若功能塊有設定 POU 密碼保護或者功能塊 EN 接點的狀態為 OFF 時，不論按下哪一個按鈕，單步執行皆不會進入功能塊的內部。此外，於 ST 語言中進行單步執行時，必須配合中斷點的設置才可正常操作，而 SFC 的 Action 與執行順序在 SFC POU 之後的其他 POU 亦必須設定中斷點才可正常操作。

下列圖示則分別以  與  圖示來比較進行單步執行時的差異。

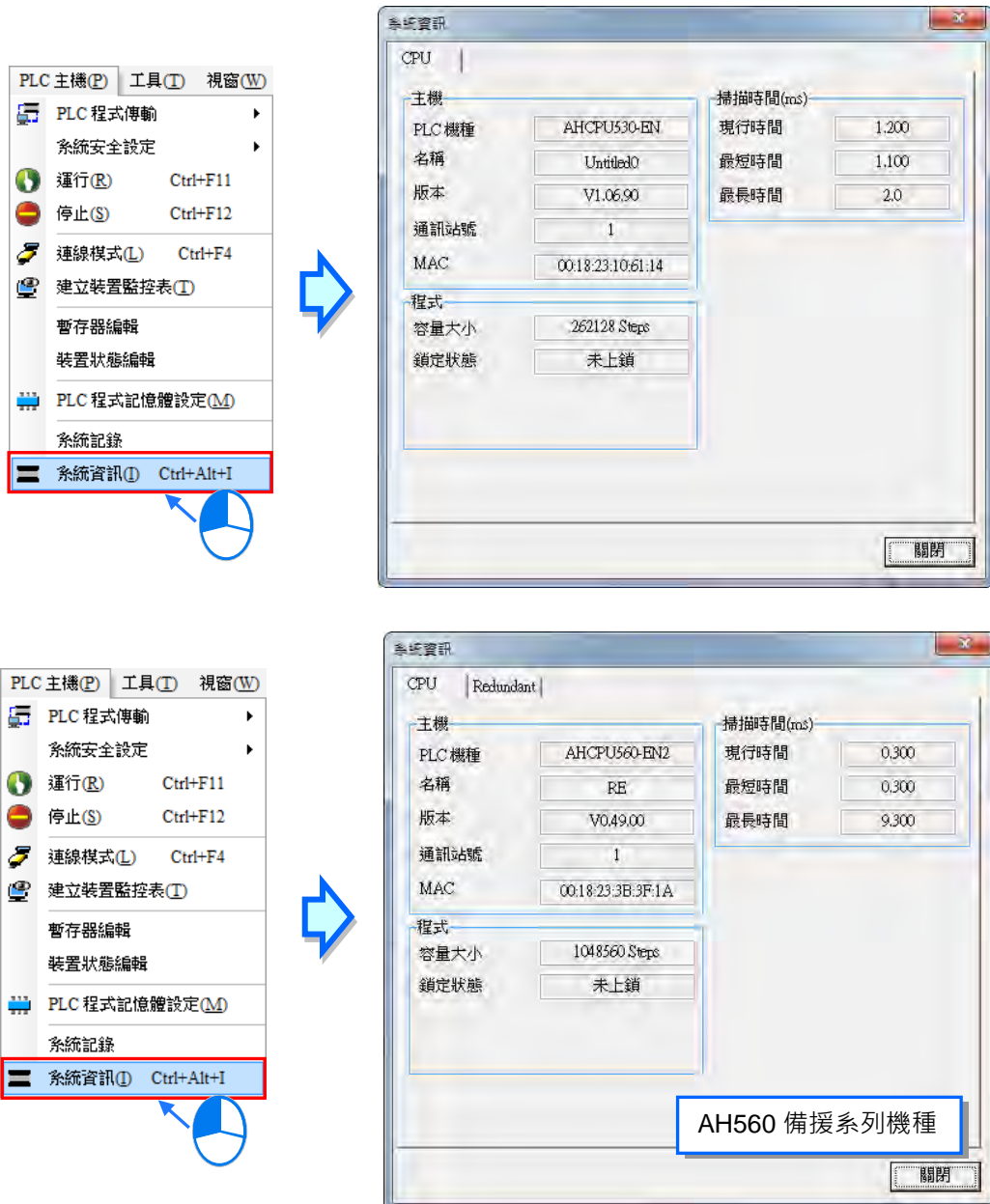


18

18.4 確認 PLC 狀態

18.4.1 PLC 系統資訊

在 ISPSOft 可與主機正常連線的情況下，於功能工具列點選 **PLC 主機 (P)** > **PLC 系統資訊 (I)**，系統便會主動抓取主機內部的相關資訊，並將其顯示於畫面當中；若是 AH560 備援系列機種，則系統資訊內會多一個「Redundant」分頁。



Redundant 分頁的資訊，為當前備援系統狀態或執行相關設定，此頁面僅在當前系統狀態為「備援系統就緒」時方可進行操作；在「單機模式」時只會顯示備援系統狀態資訊。如下圖。



Redundant 分頁內的系統資訊介紹如下：



- ❶ CPU ID : A 與 B 為控制器的識別代碼，可以檢查識別代碼來判斷備援系統是否發生控制器切換，也可以利用指令 API2901 「RCS」來進行設定或讀取。

同步設定：備援系統的資格檢查與同步動作的設定，可設定的選項為「Always」及「Condition」；相關內容與檢查條件，請參閱該產品相關手冊。

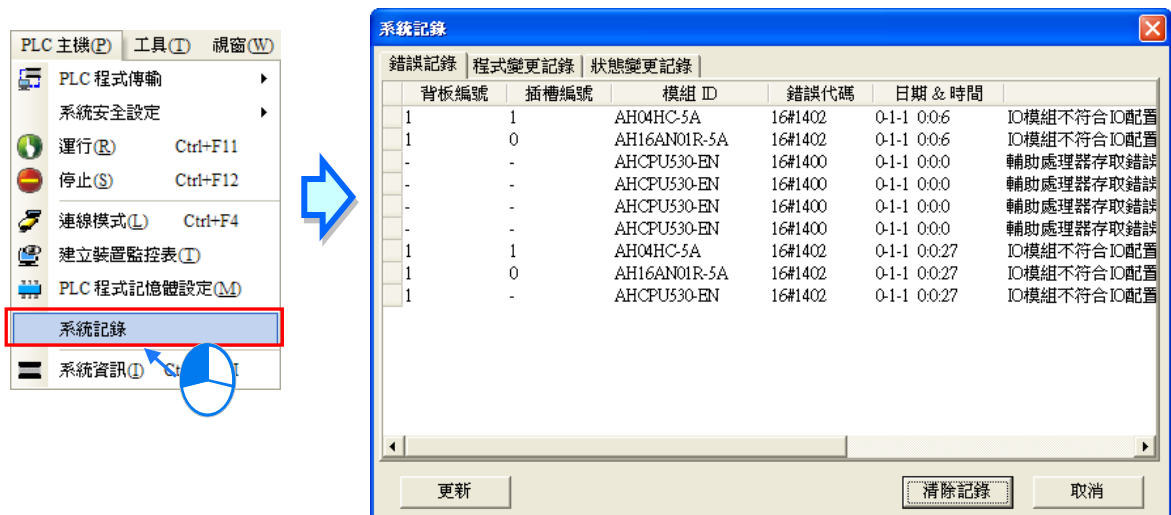
下載設定：設定完 CPU ID 與同步設定項目之後，必須點選此處下載到 PLC 主機。

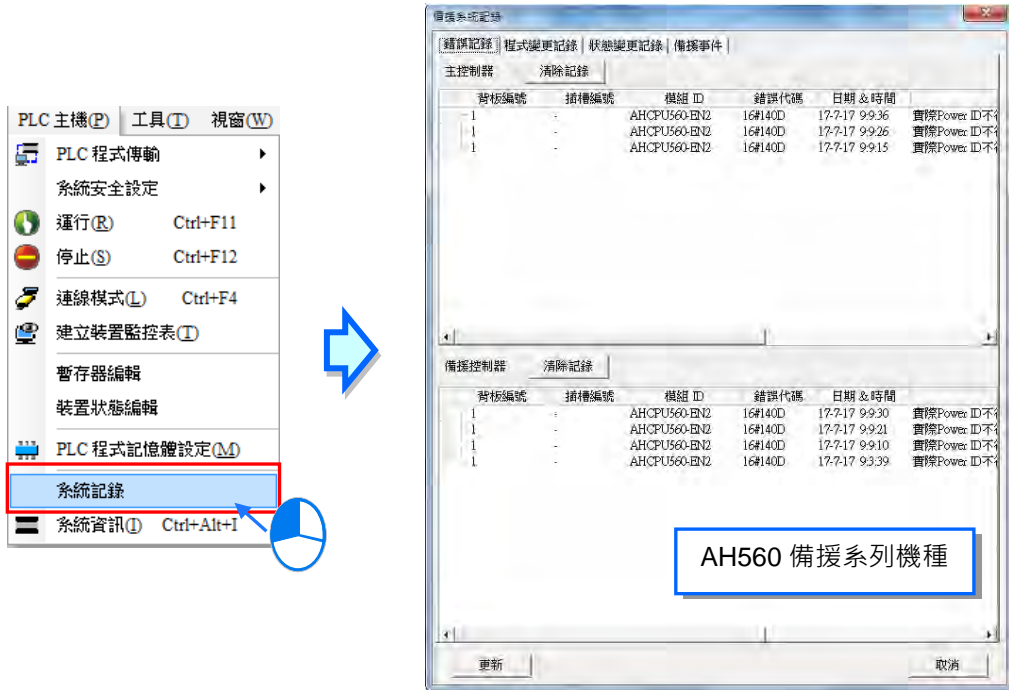
- ② **啟動同步**：主控制器對備援控制器發起資格檢查動作，如通過資格檢查則進行同步
取消同步：取消備援控制器資格
控制系統切換：執行控制器切換，備援控制器必須符合資格才允許控制器切換
取得控制權：在沒有主控制器存在時，使用此功能將備援控制器轉換成主控制器。
- ③ **顯示當前控制器當前同步資料量與歷史最大同步資料量單位為 (KBytes)**，也可以利用特殊暫存器「SR18」~「SR21」進行確認。
- ④ **當前連接控制器**：顯示當前所連結的為主控制器或備援控制器
當前系統狀態：顯示的狀態為「備援系統驗證中」、「備援系統就緒」、「單機模式」、「備援系統失敗」及「光纖同步電纜未連接」。
- ⑤ **更新**：更新 Redundant 分頁內的所有資訊

*.Redundant 分頁內配合 AH560 備援系列機種詳細操作內容，請參閱該產品相關手冊。

18.4.2 PLC 系統記錄

與主機正常連線的情況下，於功能工具列中點選 **PLC 主機 (P)** > **系統記錄**，便可檢視程式的變更及錯誤記錄；若是 AH560 備援系列機種，則可以檢視主控制器及備援控制器的記錄，如果沒有偵測到主控制器/備援控制器，則主控制器/備援控制器檢視區將呈現反灰。「系統記錄」視窗中總共包含數個頁面，並可透過上方的標籤來進行切換，所顯示的資料不會即時更新，但可點擊「更新」按鈕來更新資料。



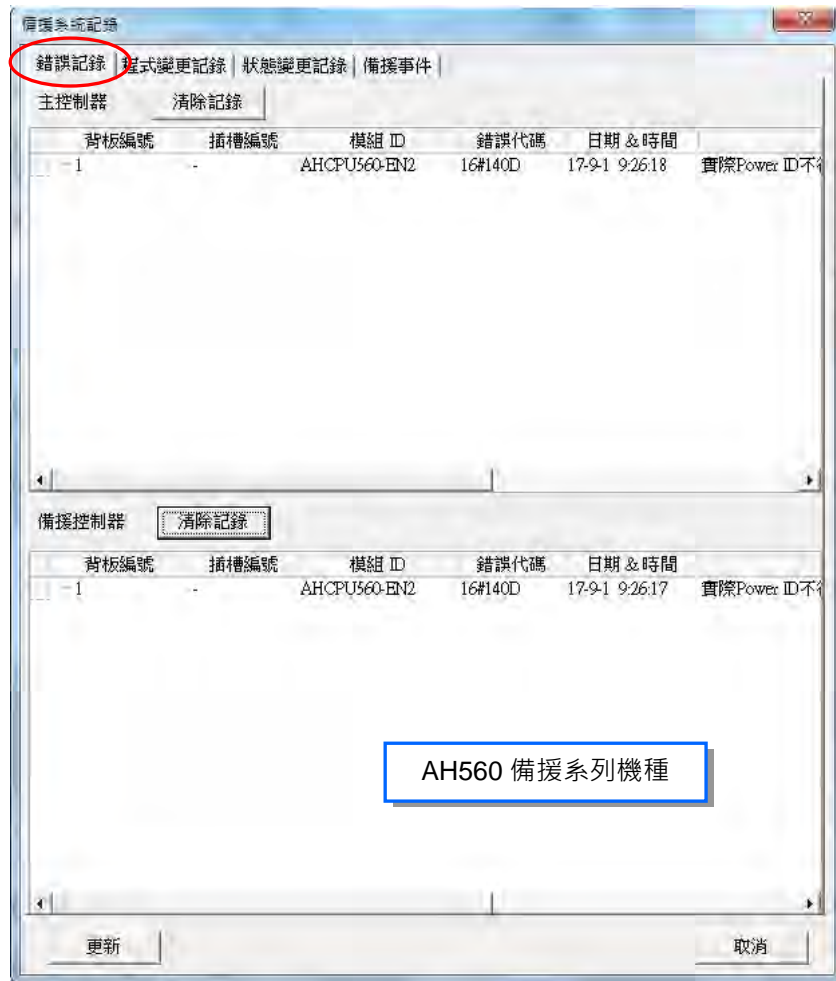



● 錯誤記錄

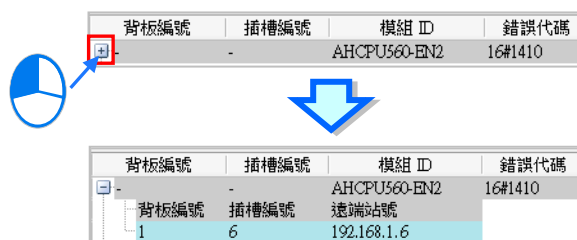
於此頁面可以檢視主機及所有模組的錯誤記錄，且在表格中會明確標示每筆記錄的錯誤代碼、發生時間及錯誤描述，另外還會包含產生該筆錯誤的模組 ID，以及該模組所在的背板與插槽編號。當按下「更新」按鈕時，系統便會再次向主機擷取資料，並將其更新於表格中；而按下「清除記錄」時，除了會將表格與主機內部的錯誤記錄清除之外，還會一併重置主機的錯誤狀態，但如果發生錯誤的原因依舊存在時，主機便仍會停留在錯誤狀態中，並會重新產生新的錯誤記錄。



18



在主機有連接到遠端模組時，錯誤紀錄的檢視會提供遠端模組所在背板編號、插槽編號及 RTU 模組遠端站號，需檢視此類資料可點擊  來做檢視。

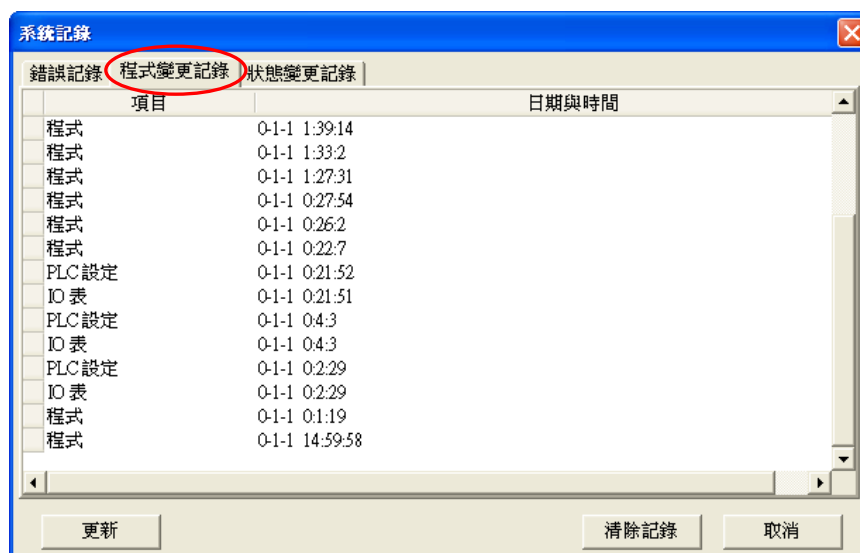


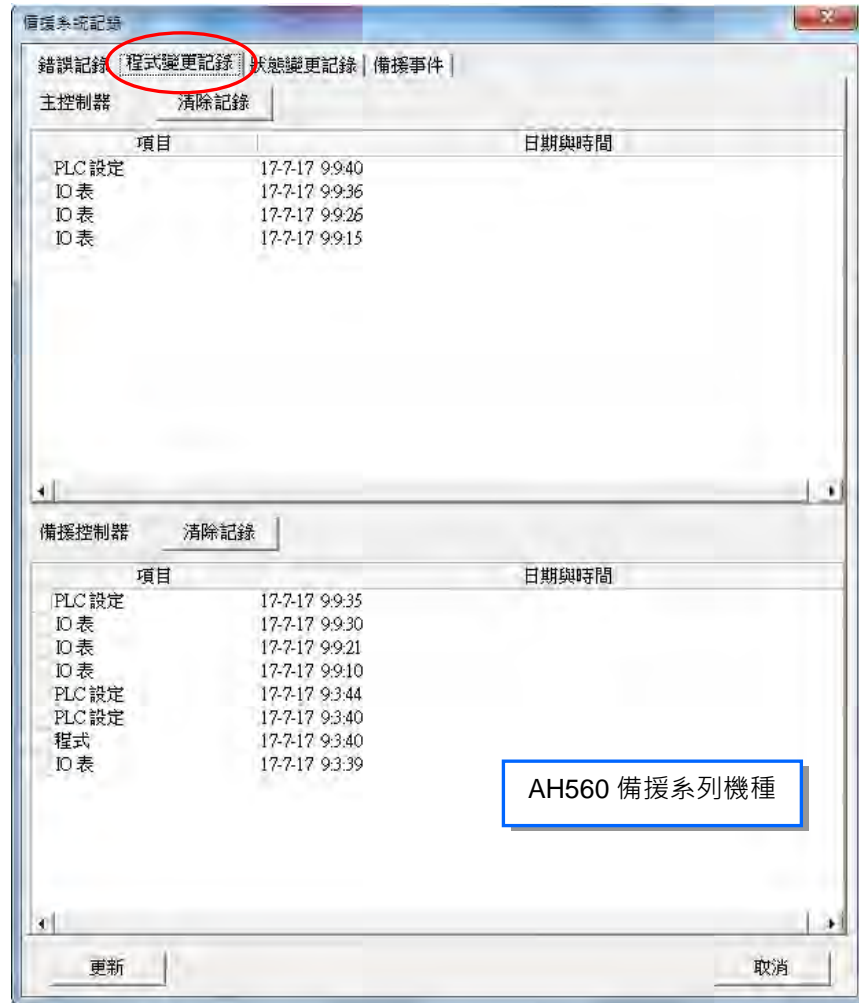
另外，錯誤記錄可設定存放至 SD 卡後，再將記錄檔案搬移到電腦管理，設定方式請參考第 3 章。此時可以點選 **工具 (I) > 檢視錯誤記錄檔**，按下「開啟記錄檔」按鈕，選擇錯誤紀錄存放的路徑，便可檢視歷史錯誤記錄內容。



● 程式變更記錄

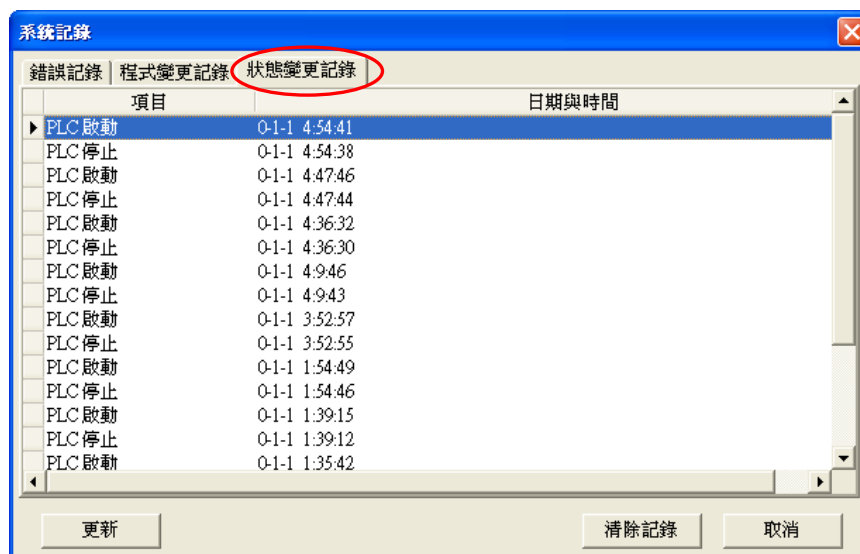
此頁面會顯示對該主機進行上下載的執行記錄，其中包括專案程式與規劃參數的上下載；同樣的，當按下「更新」按鈕時，系統便會再次向主機擷取資料，並將其更新於表格中；而按下「清除記錄」按鈕時，系統便會將表格與主機內部的程式變更記錄一併清除。

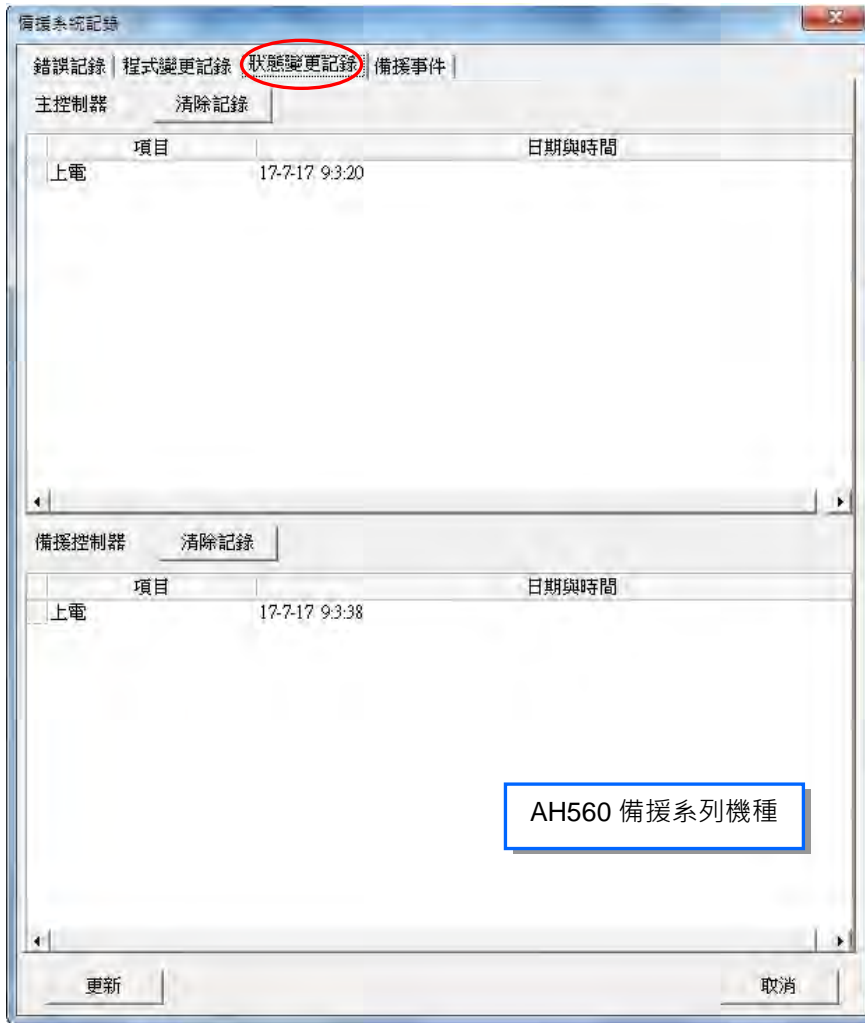




● 狀態變更記錄

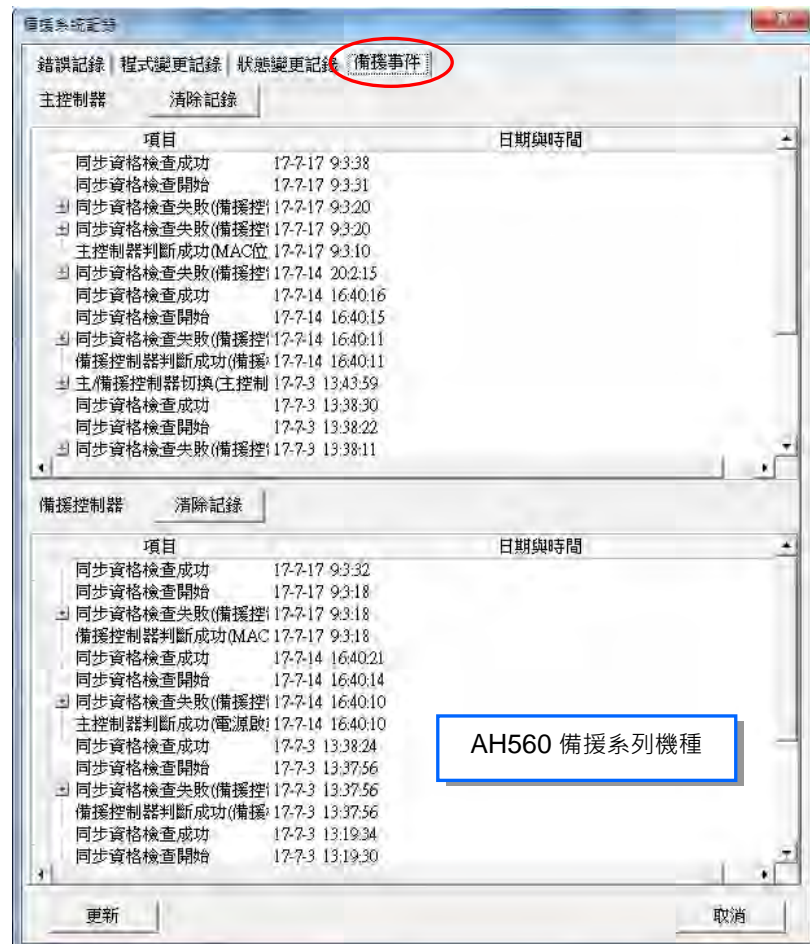
此頁面會顯示主機狀態的變更記錄。當按下「更新」按鈕時，系統便會再次向主機擷取資料，並將其更新於表格中；按下「清除記錄」按鈕時，系統便會將表格與主機內部的狀態變更記錄一併清除。





● 備援事件

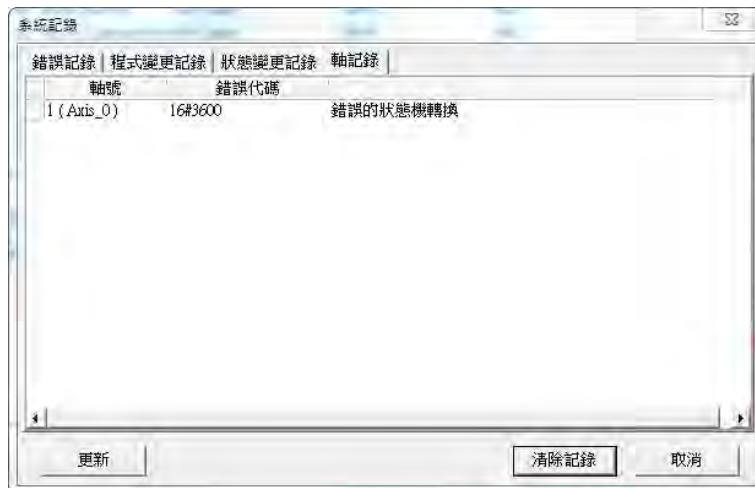
此頁面會顯示主控制器及備援控制器的事件記錄。當按下「更新」按鈕時，系統便會再次向主控制器及備援控制器擷取資料，並將其更新於表格中；而按下主控制器/備援控制器「清除記錄」按鈕時，系統便會將主控制器/備援控制器表格與主控制器/備援控制器內部的備援事件一併清除。



*.備援事件目前僅支援 AH560 備援系列機種。

● 軸記錄

頁面會顯示關於軸的錯誤記錄 (軸號欄位會顯示軸號，而軸號括弧內會顯示軸符號名稱)。當按下「更新」按鈕時，系統便會再次向主機擷取資料，並將其更新於表格中；而按下「清除記錄」按鈕時，系統便會將表格與主機內部的軸紀錄一併清除。



*.軸記錄目前僅支援 AHxxEMC 機種。

18.5 3D 座標圖

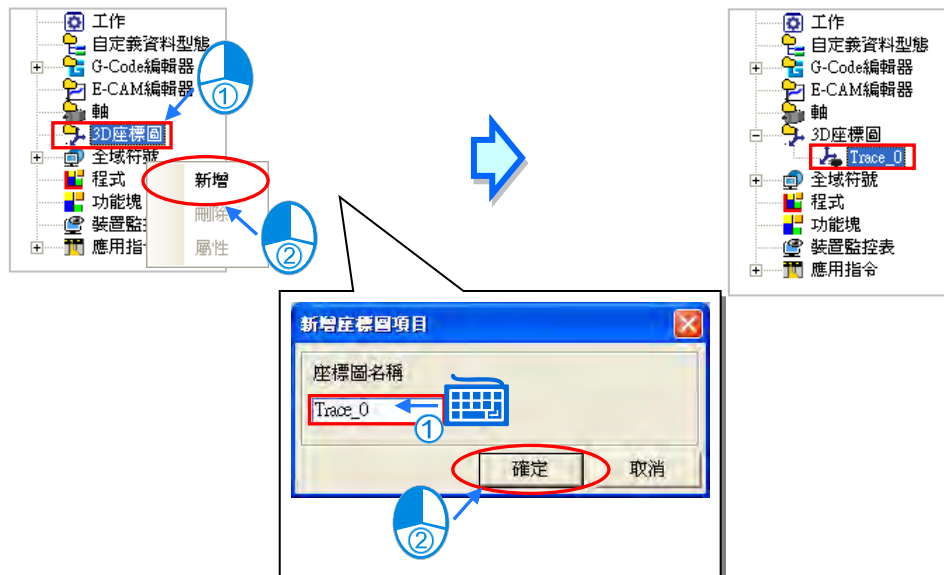
18.5.1 功能說明

監控模式中，當運動程式控制多軸運動時，可搭配 3D 座標圖功能畫出多軸間的運動軌跡，提供使用者判斷運動的正確性；也可監控任何變數符號或裝置以分析數值趨勢。

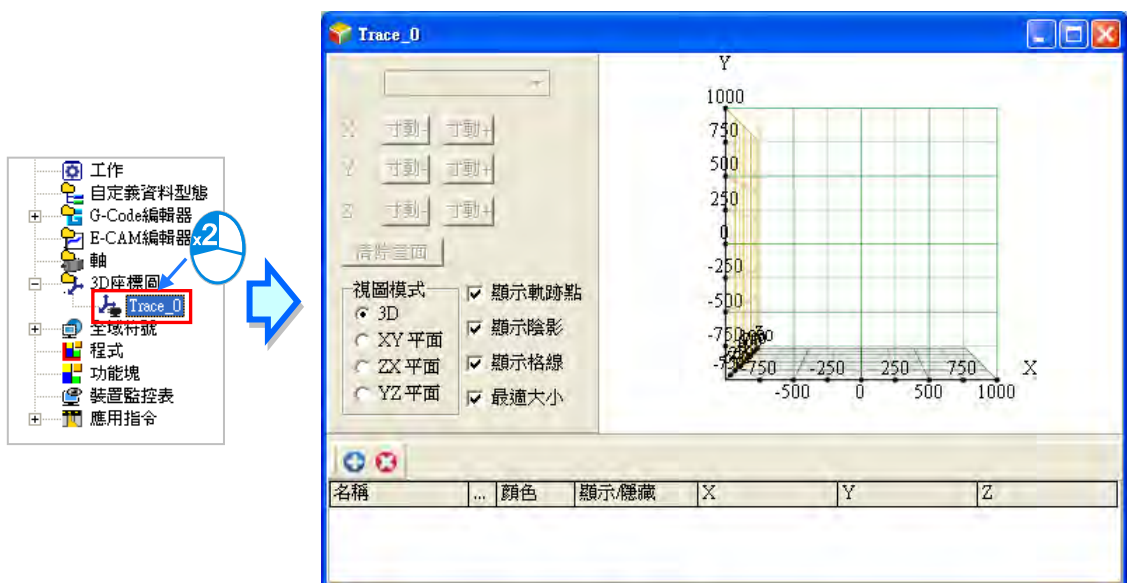
*.本功能目前僅支援 AHxxEMC 機種。

18.5.2 建立 3D 座標圖

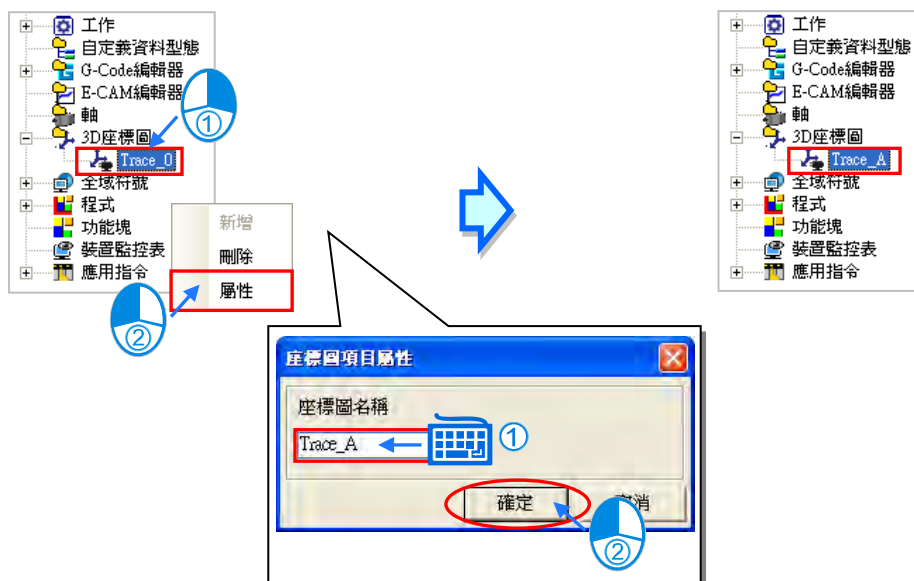
於專案管理區的 3D 座標圖項目按下右鍵，點選「新增」，輸入座標圖名稱，建立 3D 座標圖項目。專案最多可加入 10 組 3D 座標圖項目。



一旦建立了一個 3D 座標圖項目後，便會自動開啟此 3D 座標圖頁面。關閉視窗後使用者也可自行雙擊此 3D 座標圖項目再次開啟視窗，如下圖。

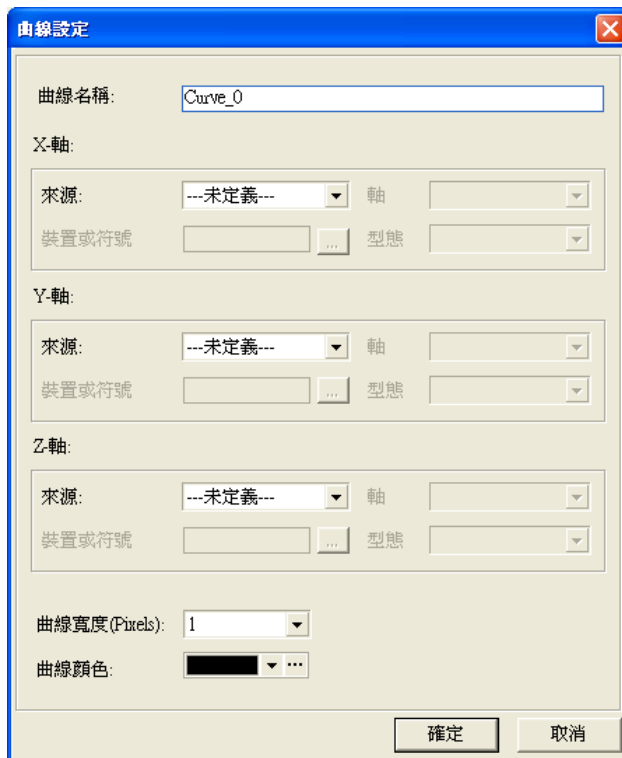


在已建立的 3D 座標圖項目上點擊滑鼠右鍵，可於快捷選單點選「刪除」，刪除此項目；或是點選「屬性」，重新設定項目名稱。



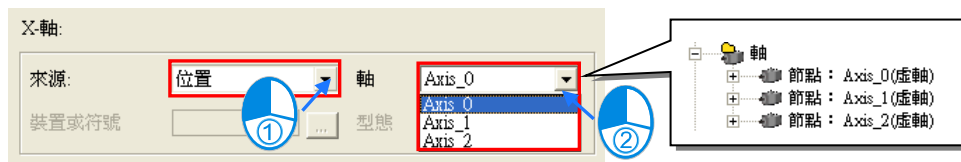
18.5.3 加入曲線

點選 3D 座標圖頁面下方的 按鈕新增一條曲線，出現曲線設定視窗如下圖。



輸入**曲線名稱**，分別選擇座標軸的 X-軸、Y-軸、Z-軸的來源，不同的來源需搭配不同的設定欄位，說明如下。

當來源選擇為**位置**或**速度**時，代表此座標軸資料來源為任一軸項目的位置或速度，因此可於**軸**的下拉選單選擇已建立的軸項目。當主機執行運動程式時，便可即時監控此軸的位置或速度。



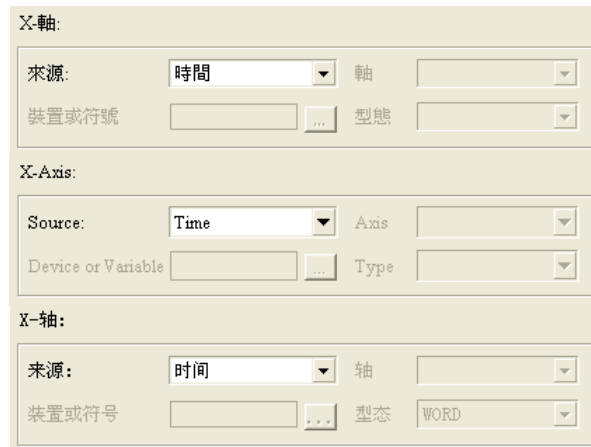
當來源選擇為**裝置**時，可直接於**裝置或符號**欄位輸入裝置位址，例如 D0，於**型態**欄位下拉選單選擇監控此裝置的型態，如下圖。



當來源選擇為**符號**時，可點選**...**按鈕開啟符號選取視窗，由已建立的全域符號或區域符號選擇要監控的變數符號，並且會根據此變數符號的設定，自動帶入型態



當來源選擇為時間時，此坐標軸數值將隨時間每秒遞增。

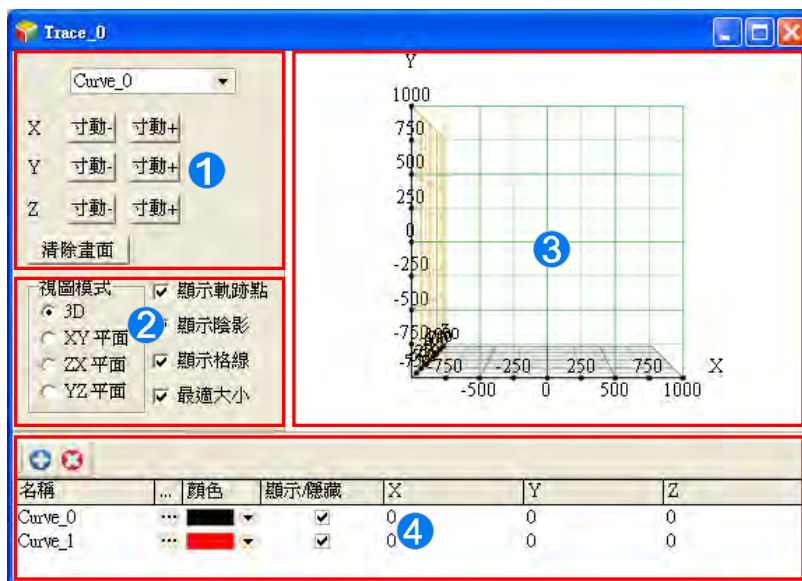


再於曲線設定視窗選擇曲線寬度與曲線顏色，按下「確定」按鈕完成曲線設定，便會將此條曲線加入 3D 座標圖下方列表。可選擇某條曲線後按下 按鈕刪除該條曲線；按下 按鈕可再次開啟曲線設定視窗；由顏色下拉選單改變曲線顏色，勾選顯示/隱藏欄位，切換是否顯示此條曲線在 3D 座標圖中。一組 3D 座標圖最多可加入 10 條曲線。

名稱	...	顏色	顯示/隱藏	X	Y	Z
Curve_0	...		<input checked="" type="checkbox"/>	0	0	0

18.5.4 顯示 3D 座標圖

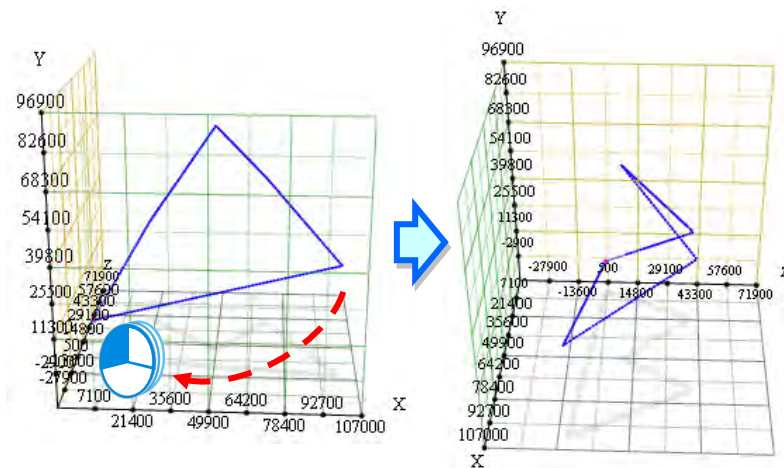
完成 3D 座標圖的建立之後，確認運動主機與電腦已建立連線，再依照前述章節完成程式下載與執行監控功能後，便可顯示 3D 座標圖。說明如下。



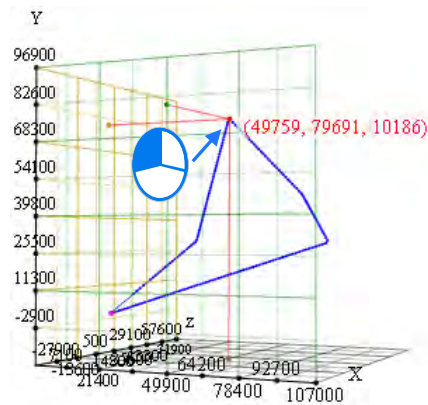
- 由下拉選單選擇區域 已建立的曲線，再分別對各個座標軸點選 或 進行正負方向的寸動，或按下 按鈕，清除已繪製的 3D 座標圖。

- ② 在視圖模式可切換 3D 的視角，或是由 X/Y/Z 軸所構成的平面。其他的選項可以切換是否顯示軌跡點、陰影效果與座標格線；當勾選**最適大小**時，右方的顯示區域將會根據目前軌跡的大小自動調整座標軸範圍。
- ③ 3D 座標圖顯示區域。
- ④ 建立 3D 座標圖中要顯示的曲線。

3D 座標圖軌跡圖形可使用滑鼠左鍵拖曳圖形後，旋轉圖形由不同角度觀察，如下圖，也可以滑鼠滾輪調整座標的刻度大小。



若滑鼠游標指到軌跡上時，會出現座標訊息，如下圖。



MEMO

18

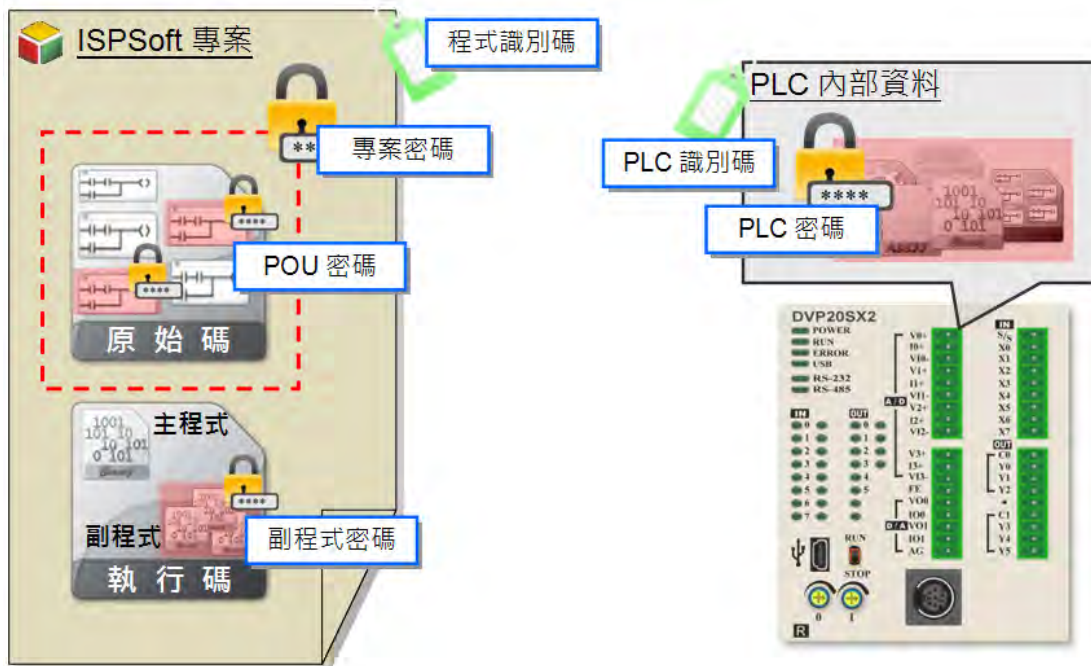
第19章 密碼管理與資料保護機制

目錄

19.1 ISPSOft 的密碼架構.....	19-2
19.2 程式識別碼與 PLC 識別碼	19-3
19.2.1 程式識別碼的設定與解除.....	19-3
19.2.2 PLC 識別碼的設定與解除	19-4
19.3 專案密碼與 PLC 密碼.....	19-5
19.3.1 專案密碼的設定與解除.....	19-5
19.3.2 PLC 密碼的設定與解除	19-7
19.3.3 專案密碼與 PLC 密碼同步.....	19-8
19.4 POU 密碼	19-9
19.5 副程式密碼.....	19-10
19.5.1 副程式密碼簡介	19-10
19.5.2 副程式密碼的設定與解除.....	19-11
19.6 其它的密碼與資料保護功能	19-12
19.6.1 程式上載失效.....	19-12
19.6.2 唯讀裝置區域設定	19-12
19.6.3 TC-01 密碼鑰匙設定	19-13

19.1 ISPSOft 的密碼架構

針對使用者所開發的專案，ISPSOft 亦提供了多重的保護機制，其目的在於確保使用者所開發的程式或核心技術都可以得到完善的保護。下列即為 ISPSOft 所提供的密碼保護架構，其中每種密碼都可選擇性的設定或解除，使用者可依據實際的需求而採用不同的資料保護機制。



密碼種類		說明
ISPSOft 專案	程式識別碼	做為第一道把關的認證資訊，且因進行專案下載時，設定於 ISPSOft 專案中的「 程式識別碼 」與設定於主機中的「 PLC 識別碼 」必須完全一致才可進行操作，因此可用來限制特定專案只可下載至特定主機。
	專案密碼	用來保護整體專案程式的密碼。若某個專案啟動了「 專案密碼 」的保護功能，則當使用者要開啟該專案中的 POU 時，系統便會要求輸入正確的密碼之後才允許開啟。
	POU 密碼	保護 POU 程式的密碼功能，且每個 POU 皆可獨立設定專屬密碼，使用者可用來保護特定的程式碼或核心技術。
	副程式密碼	用來保護執行碼中的副程式，可防堵透過上傳執行碼的方式來解析功能塊內部的程序。
	設計工具密碼	為了保護各項設計資料，不同機種對於支援的設計工具也提供密碼保護，包括 E-CAM 密碼與定位規劃表密碼等，將於各功能的章節介紹。

密碼種類		說明
PLC	PLC 識別碼	做為第一道把關的認證資訊，且因進行專案下載時，設定於 ISPSOft 專案中的「 程式識別碼 」與設定於主機中的「 PLC 識別碼 」必須完全一致才可進行操作，因此可用來限制特定專案只可下載至特定主機。
	PLC 密碼	用來保護 PLC 內部資料的密碼功能，使用者必須輸入正確的密碼之後才可對受「 PLC 密碼 」保護的資料進行上下載。

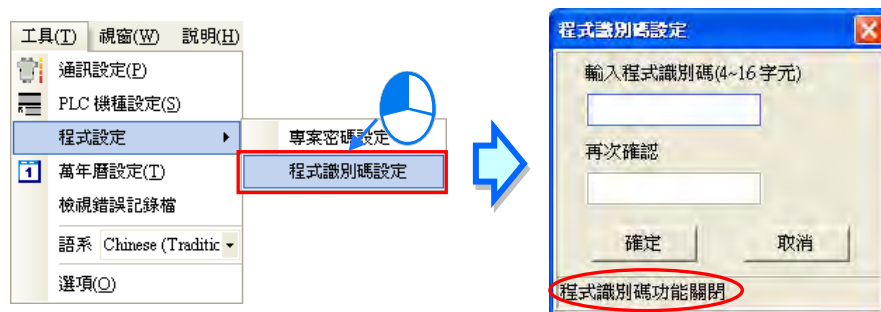
19.2 程式識別碼與 PLC 識別碼

識別碼是做為第一道把關的認證資訊，當進行下載時，設定於 ISPSOft 專案中的「**程式識別碼**」與設定於主機中的「**PLC 識別碼**」必須完全一致才可進行操作，因此識別碼的保護機制可用來限制有設定「**程式識別碼**」的專案只能下載至特定的主機中，而該主機必須設定相同的「**PLC 識別碼**」。

進行專案程式的上傳時，若該主機有設定「**PLC 識別碼**」時，系統會在開始上傳資料前要求使用者輸入正確的識別碼以進行驗證；而當開啟一個有設定「**程式識別碼**」的專案時，系統同樣會要求輸入正確的識別碼後才允許開啟，因此識別碼可用來做為使用者存取資料的權限認證。

19.2.1 程式識別碼的設定與解除

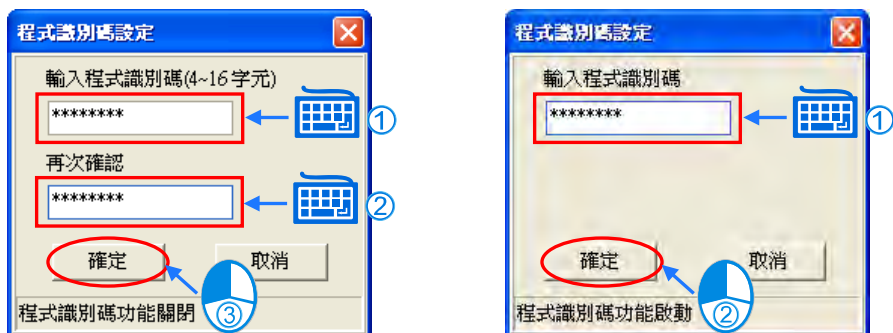
於功能工具列中點選 **工具 (T) > 程式設定 > 程式識別碼設定**，之後便會出現識別碼的設定視窗，而由該視窗的下方便可得知目前開啟的專案是否有啟動「**程式識別碼**」的保護功能。



若專案目前的「**程式識別碼**」功能為關閉狀態時，視窗中便會有兩個設定識別碼的欄位，而要設定識別碼時，請先後於兩個欄位中輸入相同的識別碼後按下「**確定**」鍵即可。

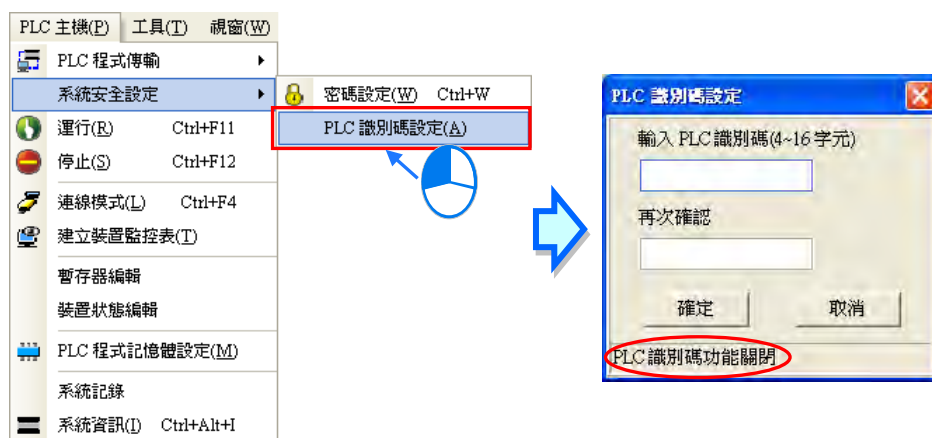
若目前「**程式識別碼**」的功能為啟動狀態，則視窗中只會有一個用來解除識別碼的欄位，而要解除識別碼時，請直接於此欄位輸入正確的識別碼後按下「**確定**」鍵即可。

設定的識別碼中僅能有英文字母、數字及空白，其中英文字母必須區分大小寫；不同機種的識別碼字數限制可能不同，請依照視窗欄位的提示進行設定。當設定完成後，視窗的狀態便會自動切換，確認無誤後按下「**取消**」鍵便可關閉視窗。



19.2.2 PLC 識別碼的設定與解除

要設定「**PLC 識別碼**」之前，請先確認 ISPSOft 已可與主機正常連線，於功能工具列中點選 **PLC 主機 (P)** > **系統安全設定** > **PLC 識別碼設定 (A)**，之後 ISPSOft 便會主動連線至 PLC 並將主機的狀態讀取回來；當成功讀取主機的資訊之後，ISPSOft 便會自動開啟識別碼的設定視窗，而由該視窗的下方便可知主機目前的「**PLC 識別碼**」功能是否啟動。



若主機目前的「**PLC 識別碼**」功能為關閉狀態時，視窗中便會有兩個設定識別碼的欄位，而要設定識別碼時，請先後於兩個欄位中輸入相同的識別碼後按下「**確定**」鍵即可。

若目前「**PLC 識別碼**」的功能為啟動狀態，則視窗中只會有一個用來解除識別碼的欄位，而要解除識別碼時，請直接於此欄位輸入正確的識別碼後按下「**確定**」鍵即可。

設定的識別碼中僅能有英文字母、數字及空白，其中英文字母必須區分大小寫；不同機種的識別碼字數限制可能不同，請依照視窗欄位的提示進行設定。當設定完成後，視窗的狀態便會自動切換，確認無誤後按下「**取消**」鍵便可關閉視窗。



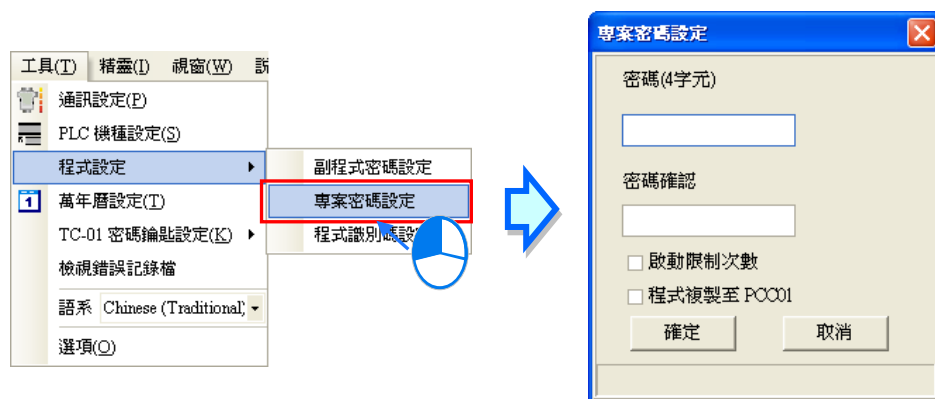
19.3 專案密碼與 PLC 密碼

「專案密碼」的主要功能在於保護整體的專案程式，若使用者開啟某個設有「專案密碼」的專案後，當使用者欲開啟該專案中的 POU 時，系統便會要求輸入正確的密碼之後才允許開啟，且「專案密碼」可設定輸入的限制次數，當使用者輸入密碼的錯誤次數超過最大限制時，該專案便會自動關閉。

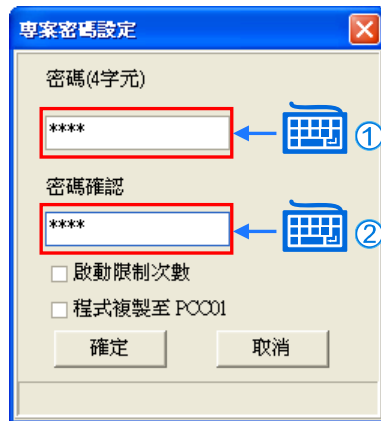
「PLC 密碼」則是設定於主機當中而用來保護主機內部資料的一組密碼，當要進行專案的上傳或下載時，系統都會要求輸入正確的密碼之後才允許後續的操作。設定「PLC 密碼」時可一併設定密碼驗證的限制次數，當使用者輸入密碼的錯誤次數超過最大限制時，該主機便會被鎖定而不再允許進行上傳或下載的操作，此時唯有將 PLC 回復至出廠設定值之後，鎖定的狀態才可解除。

19.3.1 專案密碼的設定與解除

於功能工具列中點選 **工具 (I)** > **程式設定** > **專案密碼設定**，之後便會出現「專案密碼」的設定視窗，且由該視窗的狀態便可知目前開啟的專案是否有啟動「專案密碼」的保護功能。當開啟的是 DVP 機種的專案時，於「專案密碼」的視窗中還會出現「程式複製至 PCC01」的功能項目。

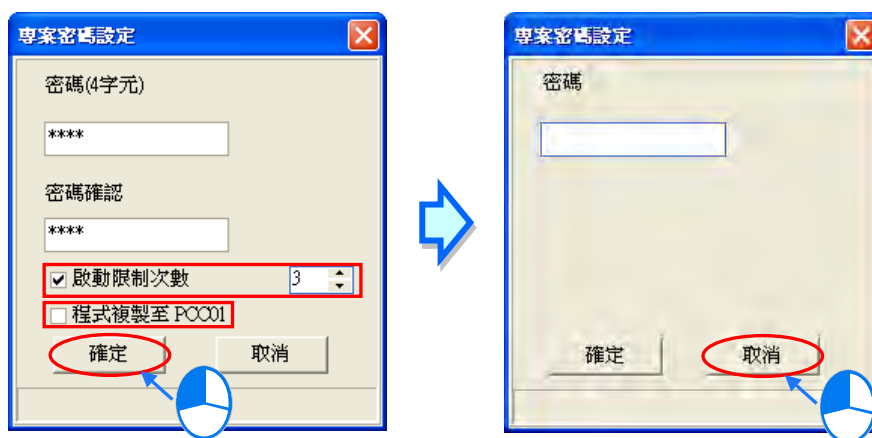


若目前的「專案密碼」為關閉狀態時，視窗中便會有兩個設定密碼的欄位，而要啟動「專案密碼」的功能時，請先後於兩個欄位當中輸入相同的密碼，而輸入的密碼只可為英文字母、數字及空白，其中英文字母必須區分大小寫；不同機種的密碼字數限制可能不同，請依照視窗的提示進行設定。



若要設定密碼輸入的限制次數時，請先勾選「啟動限制次數」之後，於其右側的欄位設定次數；而要啟動「程式複製至 PCC01」的功能（詳見下述）時，請直接勾選該項目即可。

完成設定後請按下「確定」鍵，若設定無誤時，視窗便會自動切換為啟動「專案密碼」的狀態，此時按下「取消」鍵便可關閉視窗。



當目前「專案密碼」的功能為啟動狀態，則視窗中只會有一個用來解除密碼的欄位，而要解除密碼時，請直接於此欄位輸入正確的密碼後按下「確定」鍵即可。同樣的，若成功解除密碼後，視窗便會自動切換為關閉「專案密碼」的狀態，此時按下「取消」鍵便可關閉視窗。

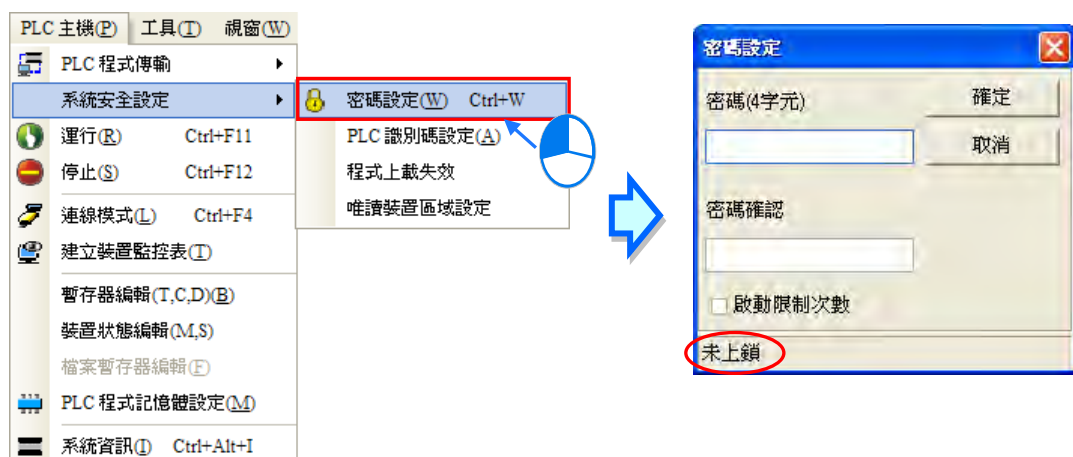


補充說明

PCC01 (產品全名為 **DVPPCC01**) 為 DVP 系列機種專用的資料備份卡，而在此節內容當中所提及的「程式複製至 PCC01」功能，其主要的作用在於下載專案至 PLC 主機後，系統將會一併將主機設置於某個狀態，而在此狀態下將 **PCC01** 連結至主機來進行備份時，系統便會將所下載的專案程式也一併備份至 **PCC01** 當中；不過在下載完畢後至 **PCC01** 連結至主機之前的期間，若主機有再接收到其它的通訊命令，如透過 ISPSOft 檢視 PLC 資訊或切換運行狀態，此時所設定的「程式複製至 PCC01」的功能狀態便會失效。此功能目前僅支援 DVP 系列的 ES2 機種，關於支援此功能的韌體版本與 **DVPPCC01** 的使用方式請參考相關使用手冊或技術文件。

19.3.2 PLC 密碼的設定與解除

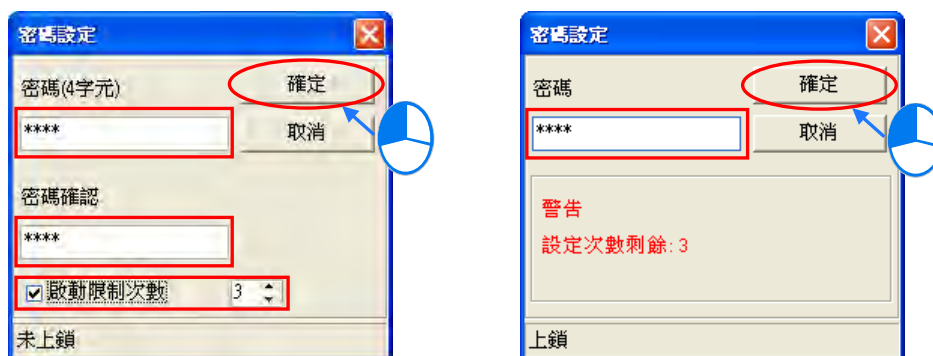
欲設定「PLC 密碼」之前，請先確認 ISPSOft 已可與主機正常連線，於功能工具列中點選 **PLC 主機 (P)** > **系統安全設定** > **密碼設定 (W)**，之後 ISPSOft 便會主動連線至 PLC 並將主機的狀態讀取回來；當成功讀取主機的資訊之後，ISPSOft 便會自動開啟密碼的設定視窗，而由該視窗的下方可得知主機目前的「PLC 密碼」功能是否啟動。



若主機目前的「**PLC 密碼**」功能為關閉狀態時，視窗中便會有兩個設定密碼的欄位；當要設定密碼時，請先後於兩個欄位中輸入相同的密碼；而若要設定密碼輸入的限制次數時，請先勾選「**啟動限制次數**」之後，再於其右側的欄位設定次數，完成後按下「**確定**」鍵即可。

若目前「**PLC 密碼**」的功能為啟動狀態，則視窗中只會有一個用來解除密碼的欄位，而要解除密碼時，請直接於此欄位輸入正確的密碼後按下「**確定**」鍵即可。

設定的密碼中僅能有英文字母、數字及空白，其中英文字母必須區分大小寫；不同機種的密碼字數限制可能不同，請依照視窗欄位的提示進行設定。當設定完成後，視窗的狀態便會自動切換，確認無誤後按下「**取消**」鍵便可關閉視窗。



19.3.3 專案密碼與 PLC 密碼同步

ISPSOft 有提供一個便利的功能，可讓使用者在下載的過程中同步設定「**專案密碼**」與「**PLC 密碼**」。執行下載操作時，若在傳輸選項當中勾選「**專案密碼與 PLC 密碼同步**」的項目時，則下載完成之後，主機內部的「**PLC 密碼**」將被改寫為與「**專案密碼**」一致。

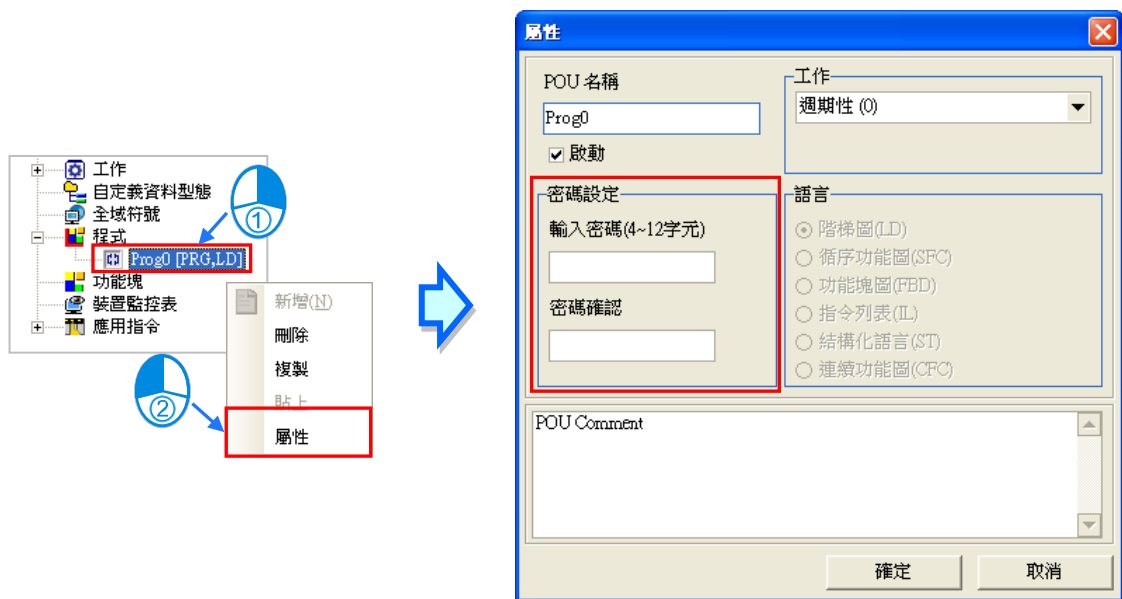
若在下載時勾選「**專案密碼與 PLC 密碼同步**」的項目，但該專案本身未設定「**專案密碼**」時，系統將會要求使用者完成「**專案密碼**」的設定；當使用者以開啟舊檔的方式開啟一個專案，且該專案本身有設定「**專案密碼**」時，下載選項中的「**專案密碼與 PLC 密碼同步**」項目便會強迫勾選。



19.4 POU 密碼

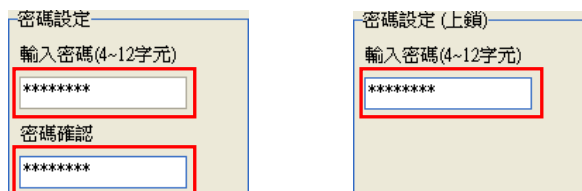
ISPSOft 允許使用者各別為不同的程式或功能塊 POU 設定一組專屬的密碼，當使用者要開啟有設定密碼保護的 POU 時，系統便會要求輸入正確的密碼之後才允許開啟，利用此功能便可對於包含特定程式碼或核心技術的資料多加一道保護機制。關於「**POU 密碼**」的設定與解除方式如下。

請於**專案管理區**中，在欲設定密碼的 POU 項目上按下滑鼠右鍵，之後於選單中點選 **屬性** 以開啟 POU 的屬性視窗，而於屬性視窗的「**密碼設定**」區域中便可得知目前「**POU 密碼**」的狀態。



若目前的「**POU 密碼**」功能為關閉狀態時，區域中便會有兩個設定密碼的欄位；當要設定密碼時，請先後於兩個欄位中輸入相同的密碼，而輸入的字元可為英文字母、數字或符號，其中英文字母必須區分大小寫，完成後按下「**確定**」鍵即可。

若目前的「**POU 密碼**」功能為啟動狀態，則視窗中只會有一個用來解除密碼的欄位，而要解除密碼時，請直接於此欄位輸入正確的密碼後按下「**確定**」鍵即可。

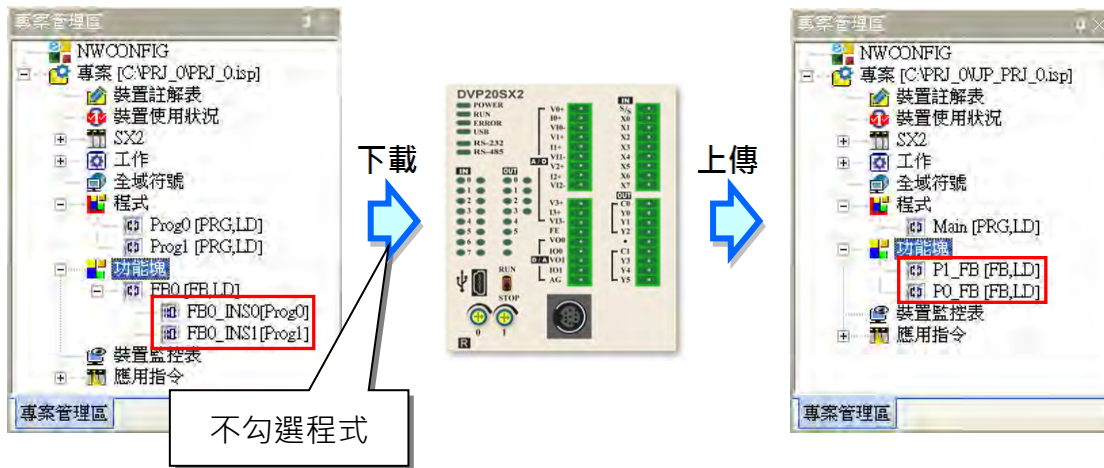


19.5 副程式密碼

* 目前 DVPxxMC 機種不支援副程式密碼。

19.5.1 副程式密碼簡介

當一個 DVP 機種的專案程式在編譯完成之後，系統所產生的執行碼可能會包含主程式、一般副程式與中斷副程式幾個部份，其中一般副程式的內容則是由原始碼中的功能塊 POU 編譯而來；而若後續在進行下載時不勾選「程式」而僅下載「執行碼」的部份，則上傳完成後，執行碼中的一般副程式便會再度被轉換為 ISPSOft 中的功能塊 POU，且因「POU 密碼」只會保護原始碼的 POU 程式，因此由執行碼重新轉換的 POU 便不受其保護，而使用者便可隨意開啟並對功能塊內部的內容進行解析。



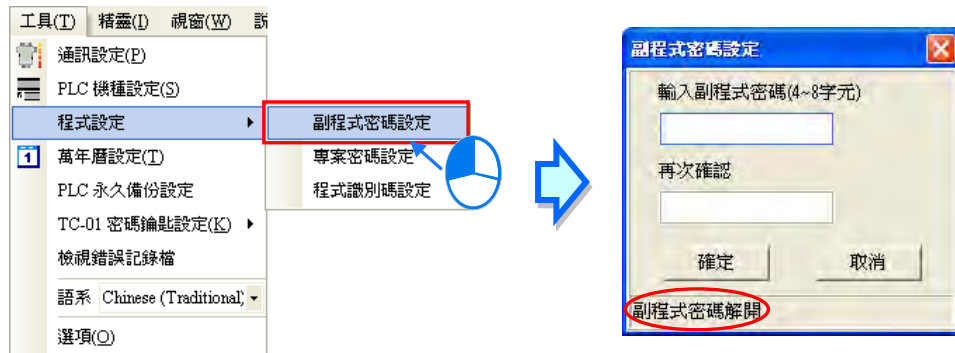
一般而言，開發者會將核心技術或較底層的程式包裝成功能塊，但若透過上述的方式取得執行碼時，所有的技術便有可能會被解析出來，因此為了避免此種狀況，使用者可在專案下載之前另外加上「副程式密碼」的保護機制；若下載的程式有設定「副程式密碼」且功能塊 POU 有設定密碼時，當執行碼中的副程式被上傳回來並轉換為功能塊之後，先前所設定「副程式密碼」便會被轉換為所有功能塊 POU 的「POU 密碼」，因此當使用者要開啟功能塊時，系統便會要求使用者輸入正確的密碼之後才允許開啟。

19



19.5.2 副程式密碼的設定與解除

於功能工具列中點選 **工具 (T) > 程式設定 > 副程式密碼設定**，之後便會開啟「副程式密碼」的設定視窗，而由該視窗的下方即可得知目前的「副程式密碼」功能是否啟動。



若目前「副程式密碼」的功能為關閉狀態時，視窗中便會有兩個設定密碼的欄位；當要設定密碼時，請先後於兩個欄位中輸入相同的密碼，完成後按下「**確定**」鍵即可。

若目前「副程式密碼」的功能為啟動狀態，則視窗中只會有一個用來解除密碼的欄位，而要解除密碼時，請直接於此欄位輸入正確的密碼後按下「**確定**」鍵即可。

設定的密碼中僅能有英文字母、數字及空白，其中英文字母必須區分大小寫；當設定完成後，視窗的狀態便會自動切換，確認無誤後按下「**取消**」鍵便可關閉視窗。

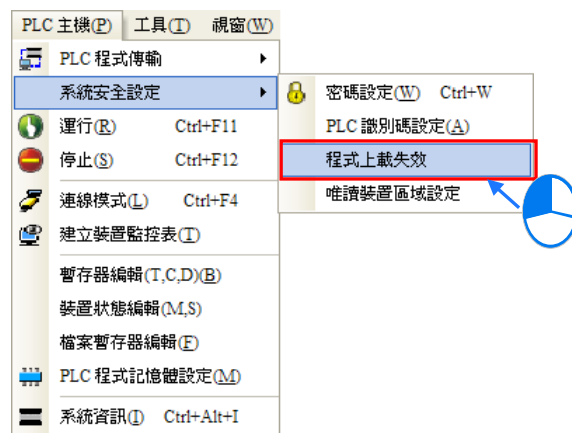


19.6 其它的密碼與資料保護功能

19.6.1 程式上載失效

此功能主要在於限制 PLC 的上載操作。當啟動此功能後，主機的上傳操作便會被禁止，而後續唯有將主機恢復至出廠預設值才可解除。若主機有設定「**PLC 密碼**」時，設定前必須先將其解除，且設定完成後，因已不再需要「**PLC 密碼**」的功能，因此使用者將無法再設定「**PLC 密碼**」。

欲設定此功能時，請於功能工具列中點選 **PLC 主機 (P)** > **系統安全設定** > **程式上載失效** 即可。且因設定之後，主機內部的資料將無法上傳，因此請務必確認是否須將主機的資料事先進行備份。

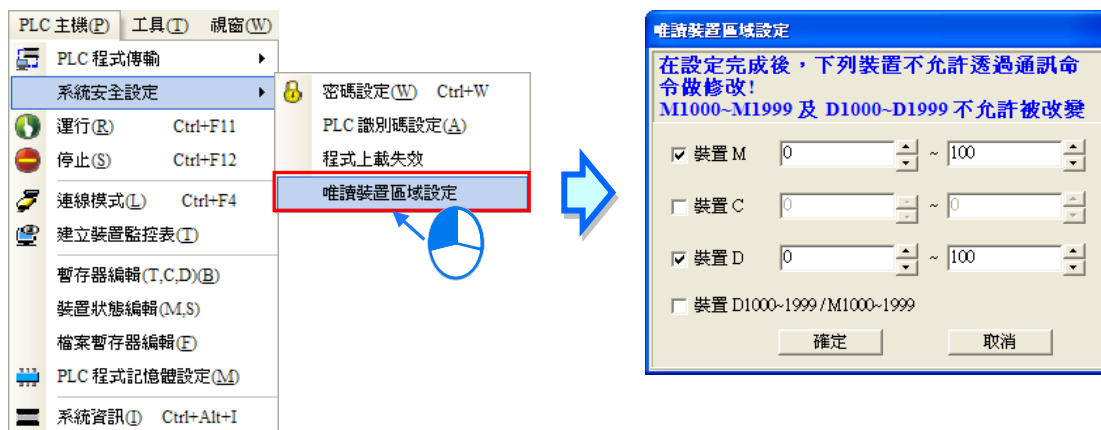


*.使用此功能前，請確認欲設定的主機機種與其韌體版本是否支援，相關細節請自行參考各機種的操作手冊或技術文件。

19.6.2 唯讀裝置區域設定

此功能在於避免 PLC 內部的裝置內容被隨意變更。當啟動此功能後，設定範圍內的裝置將無法再透過通訊連線的操作來變更內容值，而唯有將主機恢復至出廠預設值才可解除此狀態。

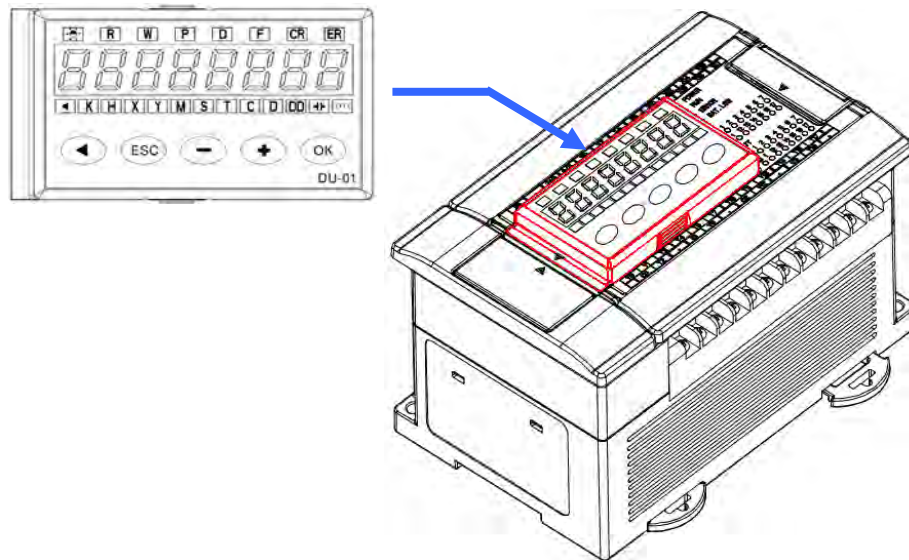
欲設定此功能時，請於功能工具列中點選 **PLC 主機 (P)** > **系統安全設定** > **唯讀裝置區域設定**，之後於設定視窗中勾選欲設置此功能的裝置類別，並設定裝置範圍，完成後按下「**確定**」鍵即可。



*.設定前請確認欲設定的主機機種與其韌體版本是否支援此功能，相關細節請自行參考各機種的操作手冊或技術文件。

19.6.3 TC-01 密碼鑰匙設定

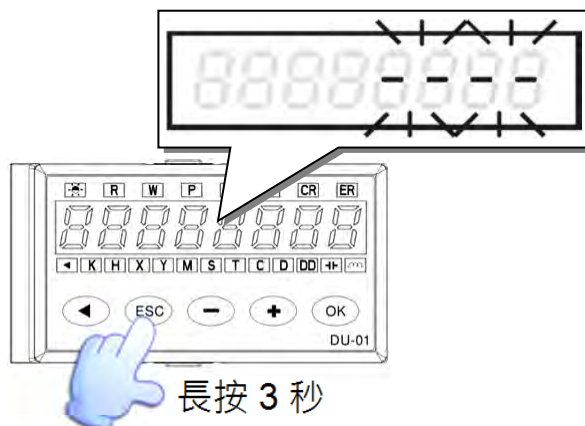
此功能主要是針對數位設定顯示器 **DVPDU-01** 所提供的輔助工具。下列即為 **DVPDU-01** 的圖片，其主要的應用是提供 DVP 系列主機一個簡易的操作介面。



DVPDU-01 的功能可分為裝置監控用的 **TS-01** 模式與資料讀寫用的 **TC-01** 模式；而在 **TC-01** 模式的應用中，為了可對有設定「**PLC 密碼**」的主機進行資料讀寫，事先便可在 **DVPDU-01** 當中設定一組與「**PLC 密碼**」相同的「**TC-01 密碼鑰匙**」，之後當進行資料讀寫時，系統便會在確認 **DVPDU-01** 中的「**TC-01 密碼鑰匙**」與主機的「**PLC 密碼**」一致後才會開始進行讀寫；當主機未設定「**PLC 密碼**」時，在資料讀寫的同時系統亦會自動將「**TC-01 密碼鑰匙**」設為該主機的「**PLC 密碼**」。

下列則為設定與清除「**TC-01 密碼鑰匙**」的操作描述，而設定前請先確認 **DVPDU-01** 處於等待的狀態，若目前的狀態為錯誤或是處於其它的操作過程中，請先完成操作或將該狀態解除。更進一步的說明請參考 **DVPDU-01** 的使用手冊或相關技術文件。

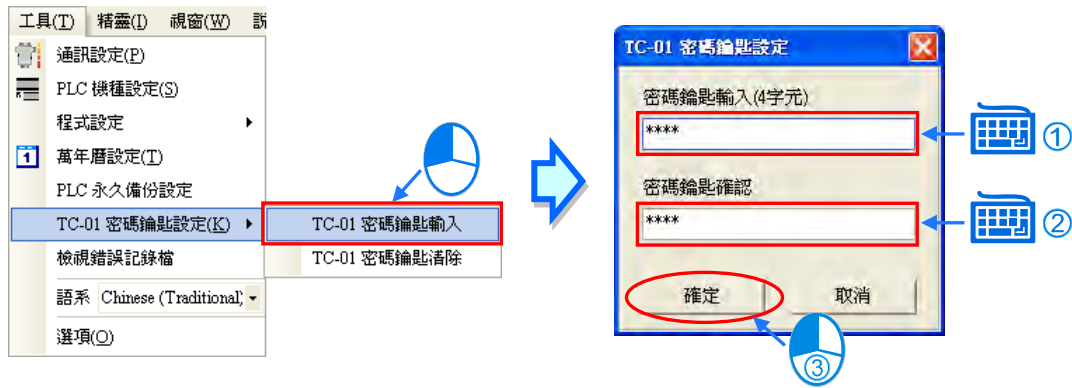
(1) 長按 **DVPDU-01** 面板的【**ESC**】鍵 3 秒，之後顯示器的部份便會進入密碼設定畫面。



(2) 於 ISPSOft 中設定或清除「TC-01 密碼鑰匙」。

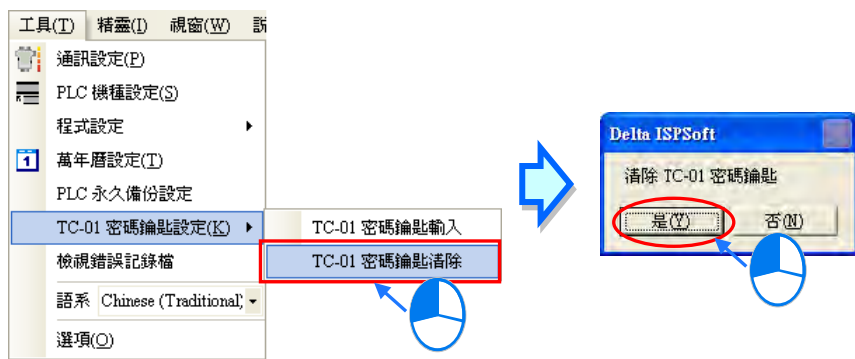
● 設定 TC-01 密碼鑰匙

於功能工具列點選 **工具 (I)** > **TC-01 密碼鑰匙設定 (K)** > **TC-01 密碼鑰匙輸入**，之後於設定視窗中先後輸入兩次相同的密碼，完成後按下「**確定**」鍵即可。

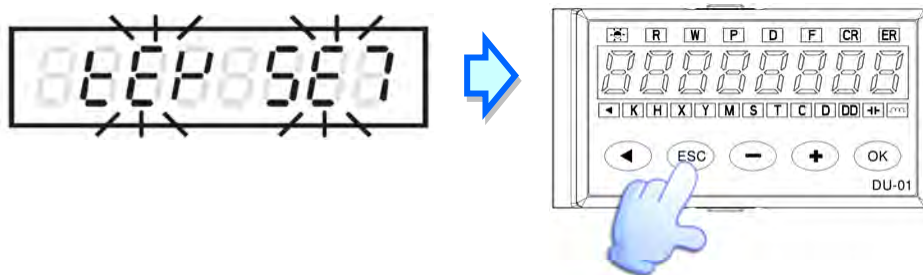


● 清除 TC-01 密碼鑰匙

於功能工具列中點選 **工具 (I)** > **TC-01 密碼鑰匙設定 (K)** > **TC-01 密碼鑰匙清除** 即可。



(3) 設定完成後，顯示器的畫面會顯示「KEY SET」，接著請再按下面板的【ESC】鍵離開即可。



第20章 網路規劃與資料交換

目錄

20.1 網路規劃工具 - NWCONFIG	20-3
20.1.1 NWCONFIG 簡介.....	20-3
20.1.2 相關基礎介紹	20-4
20.1.3 NWCONFIG 的通訊設定	20-6
20.1.3.1 NWCONFIG 中的連線機制	20-6
20.1.3.2 設定通訊參數	20-8
20.1.4 工作流程.....	20-9
20.2 網路架構的部署.....	20-13
20.2.1 部署節點.....	20-13
20.2.2 連結網路.....	20-17
20.2.3 調整或刪除已部署的裝置及網路	20-21
20.2.4 節點與網路屬性的設定	20-24
20.2.5 裝置或網路的隱藏/顯示.....	20-28
20.2.6 合法的網路架構	20-31
20.2.7 下載繞送路徑表 (Routing Table)	20-34
20.2.8 繞送路徑測試	20-36
20.3 PLC Link 的建構	20-38
20.3.1 開啟 PLC Link 的設置視窗	20-38
20.3.2 選擇主站裝置 (步驟 1)	20-40
20.3.3 設定通訊參數 (步驟 2)	20-41
20.3.4 設置資料交換表 (步驟 3)	20-43
20.3.4.1 資料交換表簡介.....	20-43
20.3.4.2 設定區塊.....	20-46
20.3.4.3 資料交換表的機種同步性.....	20-48
20.3.4.4 資料交換表的管理操作	20-49
20.3.5 PLC Link 的監控功能	20-50
20.3.6 設置 PLC Link 的注意事項	20-55
20.4 Ether Link 的建構	20-56
20.4.1 Ether Link 簡介	20-56

20.4.2 開啟 Ether Link 的設置視窗	20-57
20.4.3 資料交換表的建立與管理	20-59
20.4.4 裝置清單與圖示區的操作	20-62
20.4.5 Ether Link 的運作模式	20-64
20.4.6 下載 Ether Link 設置	20-66
20.4.7 上傳 Ether Link 設置	20-68
20.4.8 刪除非同步裝置	20-70
20.4.9 啟動/關閉線上監控模式	20-71
20.4.9.1 啟動監控	20-71
20.4.9.2 監控狀態顯示	20-73
20.4.9.3 關閉監控	20-73
20.4.10 線上啟動/停止 Ether Link (旗標模式)	20-75
20.4.10.1 啟動 Ether Link	20-75
20.4.10.2 停止 Ether Link	20-76
20.4.11 監控表與錯誤記錄	20-78
20.5 NWCONFIG 的管理與應用	20-79
20.5.1 存檔與列印	20-79
20.5.2 整體下載	20-80
20.5.2.1 下載參數說明	20-80
20.5.2.2 下載操作說明	20-80
20.5.3 ISPSOFT 中的繞送應用	20-82

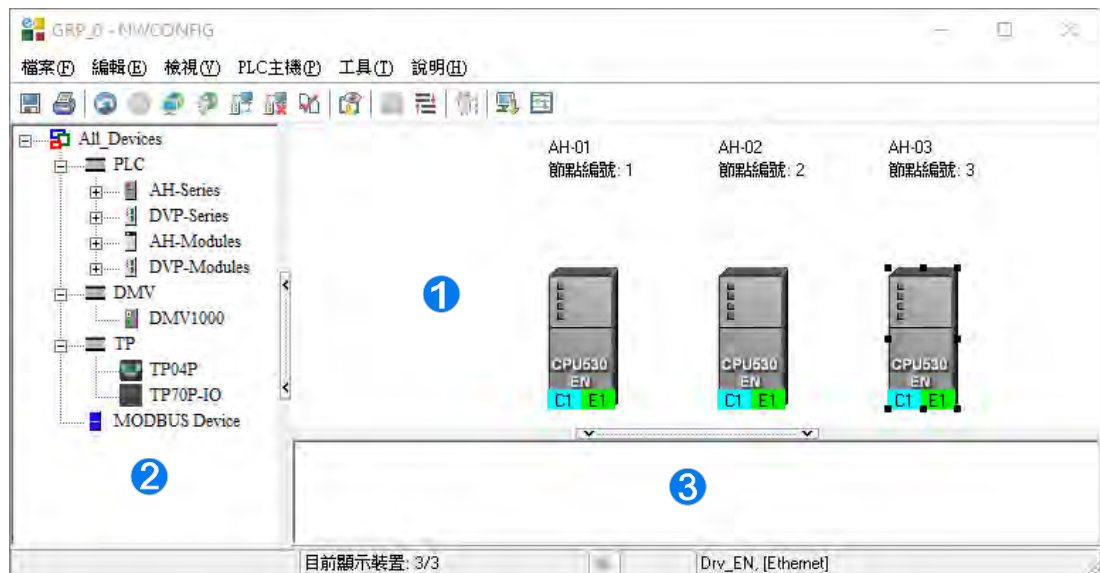
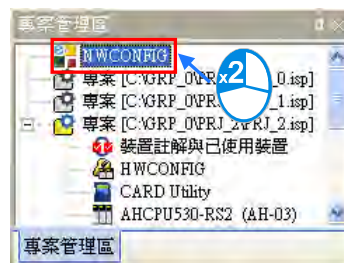
20.1 網路規劃工具 - NWCONFIG

20.1.1 NWCONFIG 簡介

NWCONFIG 為 ISPSOFT 所提供的網路規劃工具，功能在於規劃整個專案的網路架構，並藉此建立常態性的資料交換機制，而其主要負責的工作如下，詳細說明將於本章後續的內容逐一介紹。

- (a) 規劃整個專案的網路部署，並建立資料的傳送路徑。
- (b) 規劃 RS485 的資料交換機制 - **PLC Link**。
- (c) 規劃 Ethernet 的資料交換機制 - **Ether Link**。

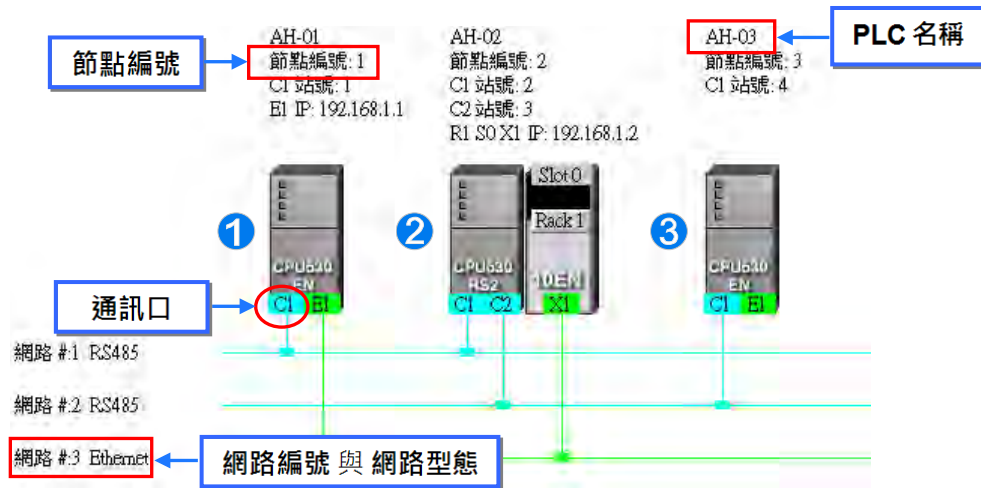
NWCONFIG 的定位在於規劃整個專案的網路架構，因此其所在的位置會在整個專案目錄的最上層，而欲開啟 **NWCONFIG** 時，請直接於專案管理區的「**NWCONFIG**」項目上雙擊滑鼠左鍵即可。



- ❶ 圖示工作區：為主要的工作區，使用者可於此規劃整個網路架構。
- ❷ 裝置列表：此區以目錄管理的方式呈現所有可使用的裝置資源。
- ❸ 訊息視窗：此處會顯示操作過程中的相關訊息。

20.1.2 相關基礎介紹

在開始進行網路規劃之前，針對操作過程所需具備的一些基礎認知，我們將在本節當中先加以介紹。



● 裝置與網路

裝置為 **NWCONFIG** 中最基本的元件，其所指的可以是 PLC 主機、擴充模組，或是使用者自行定義的設備；而此處的網路 (**NETWORK**) 則是指連接各個裝置的連線，且每個網路都必須指定一個唯一的網路編號，而其型態則有 RS-485 及 Ethernet 兩種。另外，因各裝置用以連接至網路的端點即是該裝置的通訊口，因此當一個裝置具有兩個以上的通訊口時，該裝置便可同時連接至不同編號的網路。關於 **NWCONFIG** 中的通訊口標示將於第 20.2.2 節當中另行介紹。

● PLC 名稱

如上圖的「AH-01」、「AH-02」與「AH-03」。當 PLC 的機種為 AH5x0 系列時，**PLC 名稱**則必須依據 **HWCONFIG** 中的設定，而主要的功能是在通訊時做為主機的識別，詳細說明請參考第 3.3.2.1 節；但以其它非 AH5x0 系列的機種而言，**PLC 名稱**則如同該主機的註解一般，本身並沒有任何操作上的意義。

● 節點 (Node) 與節點編號

節點 (**Node**) 指的是在一個網路中可以獨立運作的基本單位，如上圖的 ① ~ ③，其中節點 ② 雖包含主機與網路模組兩個裝置，但因模組本身無法獨立運作，因此仍須將兩者合併而視為一個節點。AH5x0 系列機種有支援「繞送功能」，即跨主機的連線操作，例如可透過上圖的節點 ① 來對節點 ③ 進行監控；而使用此功能的前提則是須先規劃資料的傳送路徑，並對路徑中的每個資料轉送點都賦予一個節點編號；當在進行規劃時，僅有 AH5x0 系列的機種可設定節點編號，且同一編號的網路上亦不可存在相同節點編號的節點；而最後完成路徑規劃並下載至各個主機之後，每個節點的主機便會擁有屬於自己的繞送表 (**Routing Table**) 以作為轉送資料的依據。

● 站號

用來做為 RS485 網路上的工作站識別，站號是依據通訊口來進行配置，基本上一個通訊口便代表一個工作站，且同一個網路編號上的工作站號不可重複。因此當一個節點本身包含數個序列埠 (**COM Port**) 時，只要有連上網路的序列埠 (**COM Port**) 都必須為其指定一個站號。

● IP 位址與 DHCP 模式

IP 位址是通訊口在網路中的位址，基本上一個通訊口便可代表一個網路端點，因此當一個網路中包含數個通訊口時，只要有連上網路的通訊口都必須指定一個 IP 位址，且 IP 位址不能重複。

DHCP 則為動態分配 IP 位址的機制，網路中必須存在著 DHCP 分配的主機，當某個通訊口使用 DHCP 模式時，該通訊口便可自動從 DHCP 主機取得一個 IP 位址。

● 網路遮罩

用來劃分子網域的參數值。

● PLC Link

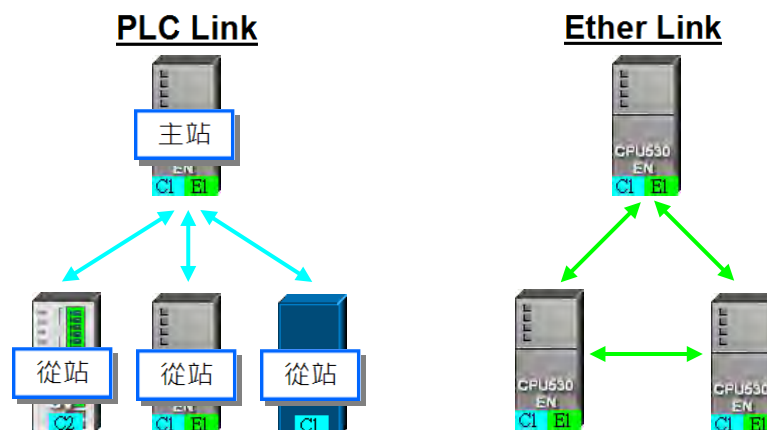
透過 RS485 的連線來進行資料交換的網路機制。當某個 RS485 的網路上存在多個節點時，我們便可於該網路中規劃一個資料交換的機制；當設置參數被下載至設為主站的主機且開始運行後，透過特殊繼電器與特殊暫存器的操作，系統便可進行資料交換的動作。

PLC Link 是一種主從架構，一個 RS485 的網路上只能有一個主站，而其它從站則必須被動的接收主站輪流發出的讀寫命令，因此從站與從站之間無法直接交換資料，而必須透過主站的轉發。

● Ether Link

透過 Ethernet 的連線來進行資料交換的網路機制。當某個 Ethernet 的網路上存在多個節點時，我們便可於該網路中規劃一個資料交換的機制，並指定其運作的方式；當 Ether Link 的設置參數被下載至各個主機且開始運行後，系統便會依據設定的啟動模式而自動地進行資料交換的動作。

有別於 PLC Link 的主從架構，Ether Link 的操作概念則是允許所有節點對其他的節點發出索取資料的需求，但只能被動的在接收到需求之後才可將資料傳送給提出需求的節點；沒有強迫寫入的動作，因此相對的有較高的安全性，且透過 TCP/IP 的協定，系統會自動管理封包傳送的機制，相較於 PLC Link 的輪流讀寫則會有較高的執行效率。



*.關於 RS485 與乙太網路 (Ethernet) 的相關概念，請自行參考其他書籍或技術文件。

20.1.3 NWCONFIG 的通訊設定

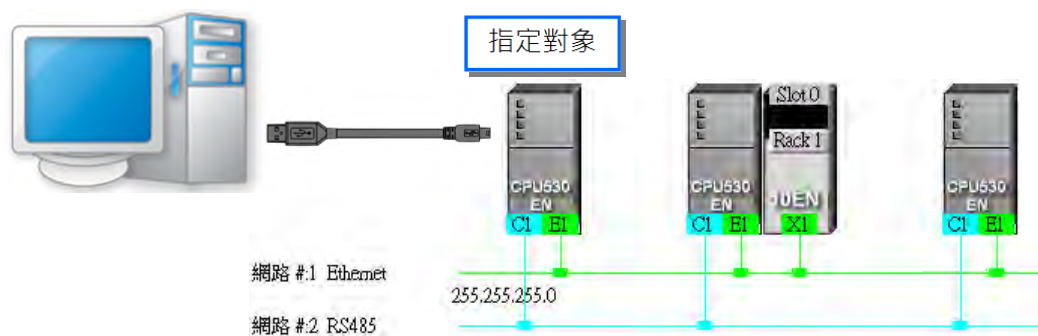
NWCONFIG 的主要工作是進行網路功能的組態與規劃，而在過程中無可避免的須要對所規劃的網路節點進行參數上下載或是監控的動作，且此處所規劃的網路節點，其中可能包含了 ISPSOft 專案機種以外的裝置，因此使用者必須另外在 NWCONFIG 當中設定通訊參數，而無法直接沿用 ISPSOft 當中的設定。在開始說明如何設定通訊參數之前，以下會先針對 NWCONFIG 中的通訊機制做一簡單的介紹，以便在進行設定時可選擇適當的通訊參數。

20.1.3.1 NWCONFIG 中的連線機制

當在 NWCONFIG 中進行上下載或線上監控時，其操作的對象將可能是網路規劃中的任何一個節點，且這些操作在 NWCONFIG 中也可能會提供**單點**與**多點**兩種操作模式，因此進行操作前請務必依據操作的模式來考量連線的方式，並依此選擇適當的通訊參數。

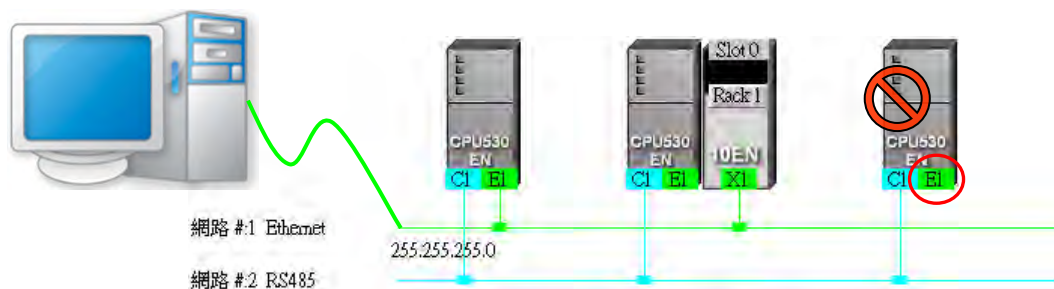
● 單點模式

僅針對指定的裝置進行上下載或線上監控，因此操作前請務必確認所指定的裝置與實際連線至電腦的裝置是否一致，否則便可能發生誤操作。

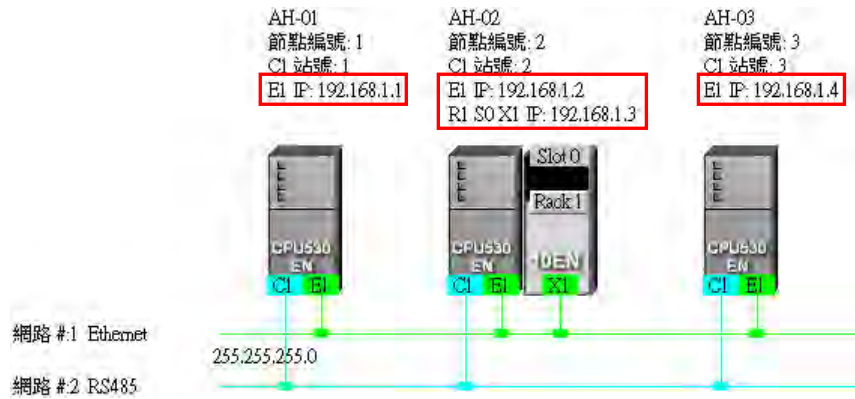


● 多點模式

同時對整個網路上的節點進行操作，但進行**多點模式**的前提是欲進行操作的所有裝置都必須連接至 Ethernet 的網路上，且每個裝置都必須設定好正確的 IP 位址，而 COMMGR Driver 也必須選擇為 Ethernet 的連線型式。

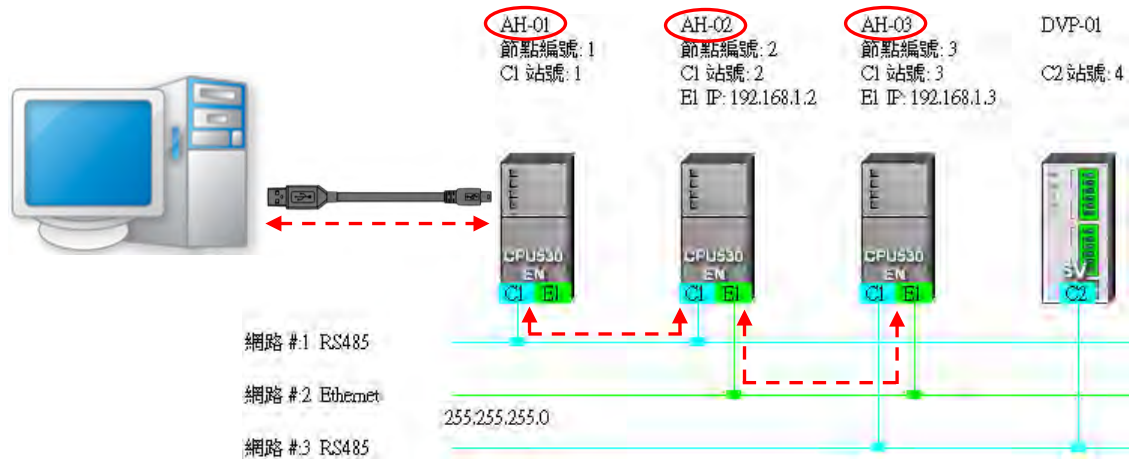


不論是單點或多點模式，若選擇的 COMMGR Driver 為 Ethernet 的型式，而之後在進行上下載或監控操作時，系統便會依據使用者在 NWCONFIG 中所規劃的 IP 位址來進行通訊，因此進行連線時務必確認各裝置內部的實際 IP 位址與 NWCONFIG 中的設置一致，且實際架設的網路也必須與規劃的配置一致或相容，否則便可能在操作過程中發生誤動作，或收發到非預期的資料。



除了上述所介紹的方式之外，NWCONFIG 亦可透過封包繞送模式的機制來進行連線操作。

封包繞送模式是 AH5x0 系列機種所提供的功能，主要的概念是透過轉送的機制來對非直接接續的裝置進行連線；如下圖，當實際與電腦連線的裝置為 AH-01，但在操作上卻須要與 AH-03 進行連線時，此時便可啟動封包繞送模式，並指定 AH-01 為初始站台，而之後當電腦送出通訊命令時，該命令便可經由 AH-01 → AH-02 → AH-03 的路徑送達欲連線的裝置 AH-03。




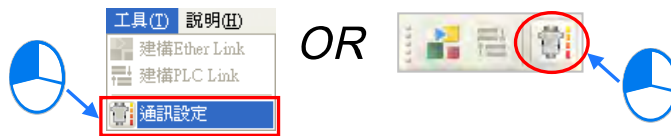
下列則為使用封包繞送模式的相關注意事項。

- 必須先於 NWCONFIG 完成所有網路的規劃，並將產生的繞送路徑表 (Routing Table) 下載至路徑中的每個節點。詳細內容將於第 20.2 節當中另行介紹。
- 只有 AH5x0 系列的機種可提供繞送的功能，而 DVP 系列機種或其他裝置則不可做為轉發封包的中繼站，不過還是可以當作繞送路徑的終點站，亦即欲進行通訊的對象。

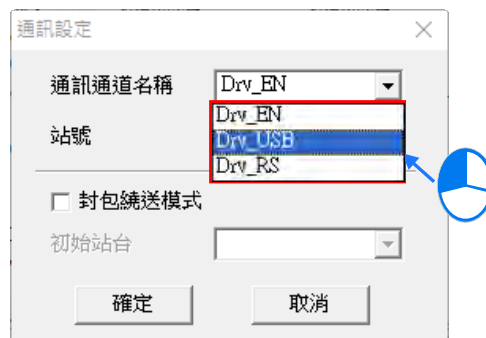
20.1.3.2 設定通訊參數

下列為在 NWCONFIG 中設定通訊參數的步驟，為確保連線正常請按照步驟操作。

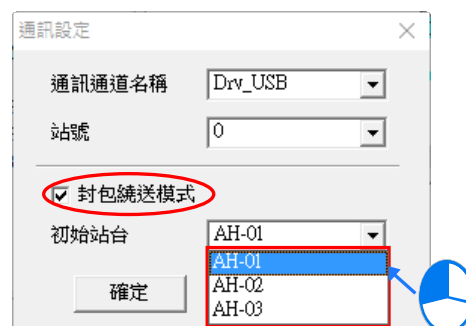
- (1) 正確啟動**通訊管理員 - COMMGR**，且於 COMMGR 中事先建立好搭配連線所需的 Driver 並確認狀態為「OK」。
- (2) 當使用**單點模式**進行操作時，請先確認要進行操作的對象與實際連線的裝置是否一致；而若要進行**多點模式**的操作時，請確認所有參與操作的裝置都必須連接至 Ethernet 的網路上，且須與 NWCONFIG 中的規劃一致。
- (3) 於 NWCONFIG 的功能工具列中點選 **工具(I) > 通訊設定**，或點選圖示工具列的  圖示，之後在畫面中便會出現通訊設定的視窗；請依下述說明進行設定，完成後按下「**確定**」即可。



- (4) 於設定視窗中的「**通訊通道名稱**」欄位下拉選擇欲使用的 COMMGR Driver



- (5) 勾選「**封包繞送模式**」，接著於下方的「**初始站台**」處下拉選擇與電腦實際連接的裝置。



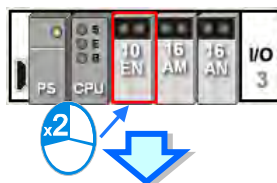
20.1.4 工作流程

因網路的規劃會牽涉到一個系統的整體運作，因此它的工作流程必須是循序漸進的，而在本節的內容中，針對完成一個網路規劃所須的工作流程會先做大略的介紹，至於細節操作的部份則將於後續的節次當中逐一說明；當然，此處所介紹的方式，僅是在一般情況下可有效率完成工作的使用建議，其方式不盡然可套用所有的應用，使用者仍可依據實際情況或個人習慣來加以調整。

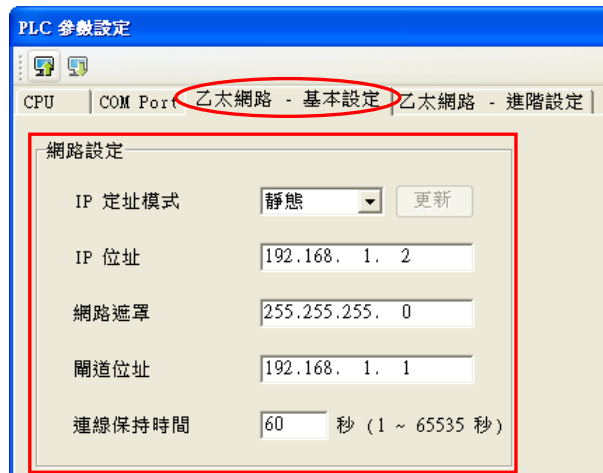
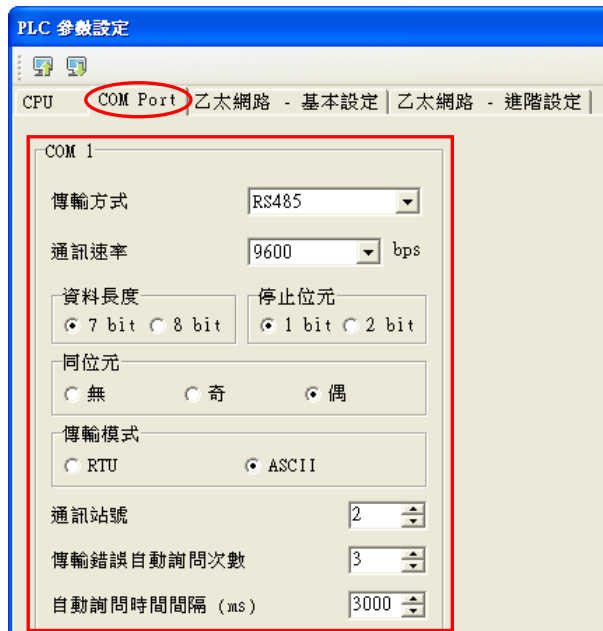
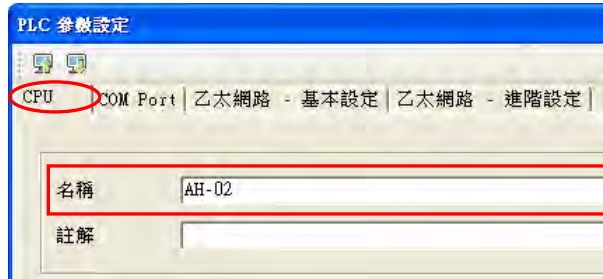
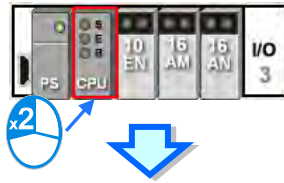
- (1) 在透過 ISPSOFT 架構系統之前，請事先完成整個網路系統的規劃設計，內容必須包括網路中各節點所採用的 PLC 機種或裝置種類、PLC 主機是否需搭配通訊模組、每個節點之間如何接續、進行接續的通訊埠口所規劃的 IP 位址或 RS485 站號、RS485 的通訊參數...等；除此之外，可能還須事先規劃預期進行資料交換的設備與資料交換的區塊，當然這個部分可能會牽涉到 PLC 程式的設計；在完成這些規劃之後，接著便可在 ISPSOFT 中逐一建構起整個網路的架構。
- (2) 於 ISPSOFT 中建立專案，如果在系統中有兩部以上的台達 PLC 時，我們會建議直接將其建立為群組專案，詳細操作方式請參考第 2.2 節的相關說明。



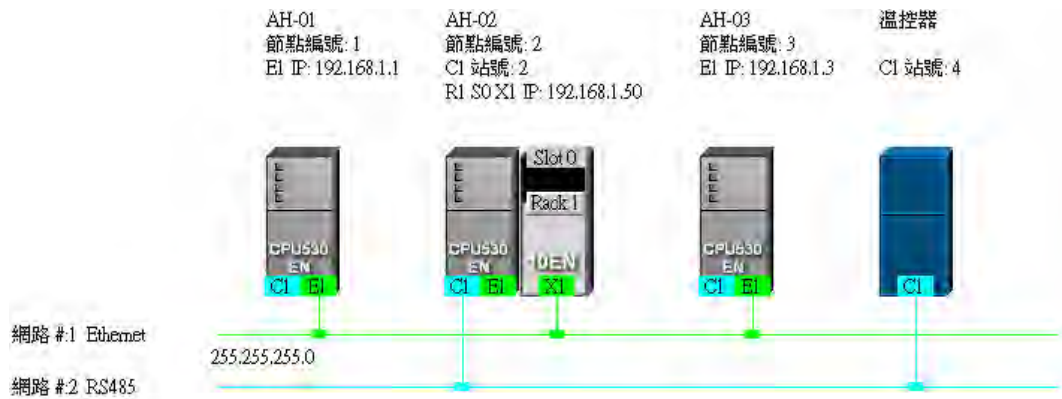
- (3) 針對 AH5x0 機種，請逐一開啟群組中每個單機專案的 HWCONFIG 工具，並完成每個機種的硬體規劃；而規劃的內容包括模組配置、通訊模組參數設定以及 CPU 參數的 PLC 標籤名稱、COM Port 設定、乙太網路基本設定...等；詳細操作方式請參考第 3 章的相關說明。



參數設定		網路參數			
AH10EN-5A		描述	位址	監控	初始值
網路參數		操作模式		<input type="checkbox"/>	單一 IP 位址 (Host 1 and Switching)
功能列表		Host 1(X1) 模式		<input type="checkbox"/>	靜態 IP
IP 過濾參數		Host 1(X1) IP 位址		<input type="checkbox"/>	192.168.1.50
IO 對映參數		Host 1(X1) 子網路遮罩		<input type="checkbox"/>	255.255.255.0
		Host 1(X1) 開道器		<input type="checkbox"/>	192.168.1.1

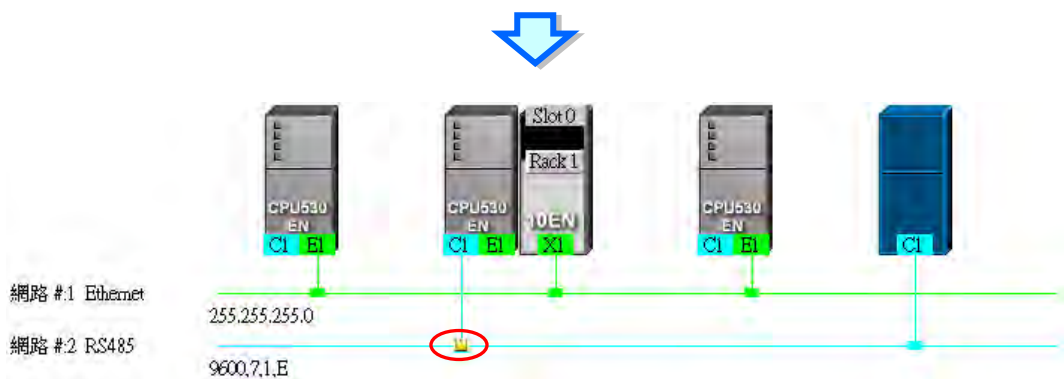


(4) 於 NWCONFIG 中完成網路部署。

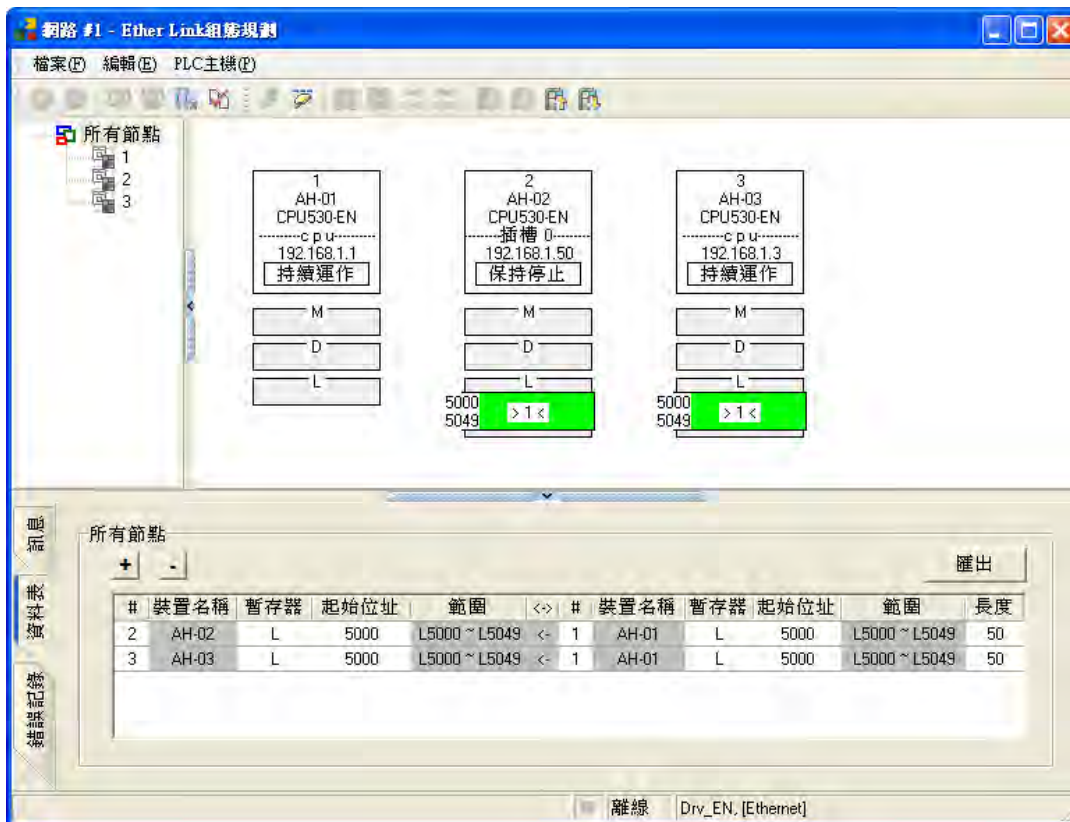


(5) 建立 PLC Link 或 Ether Link 的資料交換機制，而此兩者之間是獨立運作的，因此在規劃時並沒有先後順序的差別；規劃時須注意所有的資料讀寫區段要避免發生位址重疊，否則在實際運作時便可能產生錯亂，因此建議在規劃之前能夠依據程式的設計，事先定義資料交換的位址區段。下圖為 PLC Link 的資料交換表，且在完成設定之後，網路規劃的圖面上會加上主站的標示。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	4	讀	D3000~D3000	<=	16#1000~16#1000	1	啟動	User Defined
		寫	D3001~D3001	=>	16#1001~16#1001	1		
2	0	讀	D100	<=	D4096	0	關閉	Unknown
		寫	D100	=>	D4096	0		
3	0	讀	D200	<=	D4096	0	關閉	Unknown
		寫	D200	=>	D4096	0		
4	0	讀	D300	<=	D4096	0	關閉	Unknown
		寫	D300	=>	D4096	0		
5	0	讀	D400	<=	D4096	0	關閉	Unknown
		寫	D400	=>	D4096	0		



下圖則為 Ether Link 的資料交換表。



- (6) 依序將專案程式、HWCONFIG 的硬體規劃參數與 NWCONFIG 的網路規劃參數一一下載至每個主機當中；其中若網路規劃的內容包含 DVP 機種或 AH5x0 機種以外的裝置，請自行參照各裝置的使用說明逐一完成其通訊參數或相關功能的設定。

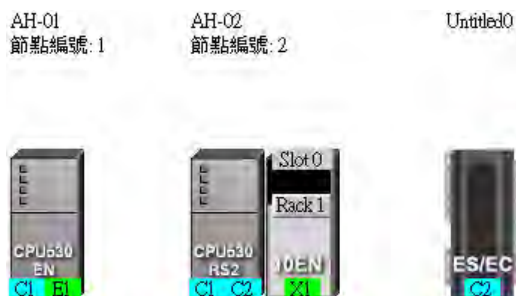
下圖則為 NWCONFIG 的下載選項視窗，其中依據各節點的規劃不同，可下載的項目也會不同。



20.2 網路架構的部署

20.2.1 部署節點

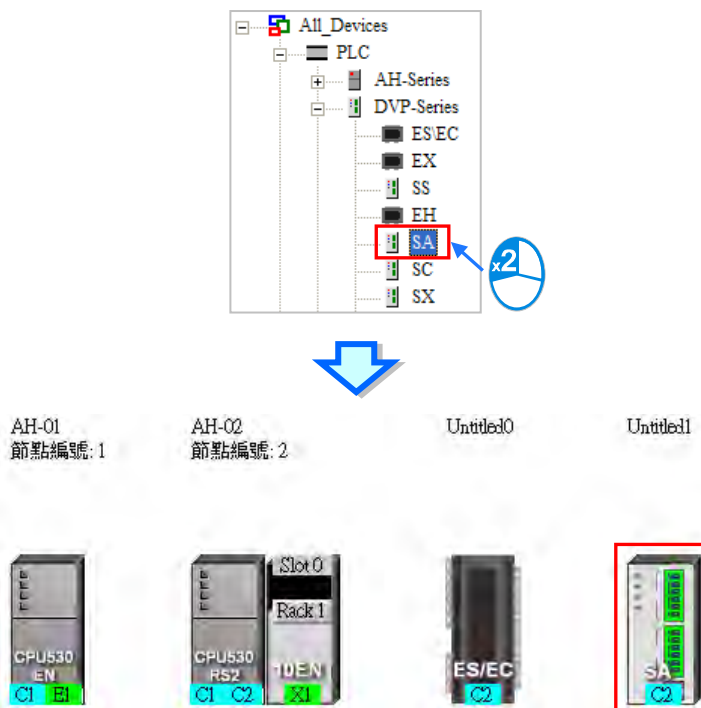
當首次開啟某個專案中的 NWCONFIG 時，系統會依據專案中所建立的機種，不分 AH5x0 或 DVP 系列，所有建立的專案機種都會自動被部署於工作區中，且 AH5x0 機種還會依據 HWCONFIG 的設定而將對應的模組與節點參數一併帶出來，而這些節點於後續的操作中也將不允許任意變更或刪除。



除了依據專案而自動帶出的節點之外，使用者仍可另外加入其它的 PLC 主機或裝置，而加入新裝置的方式共有下列兩種。

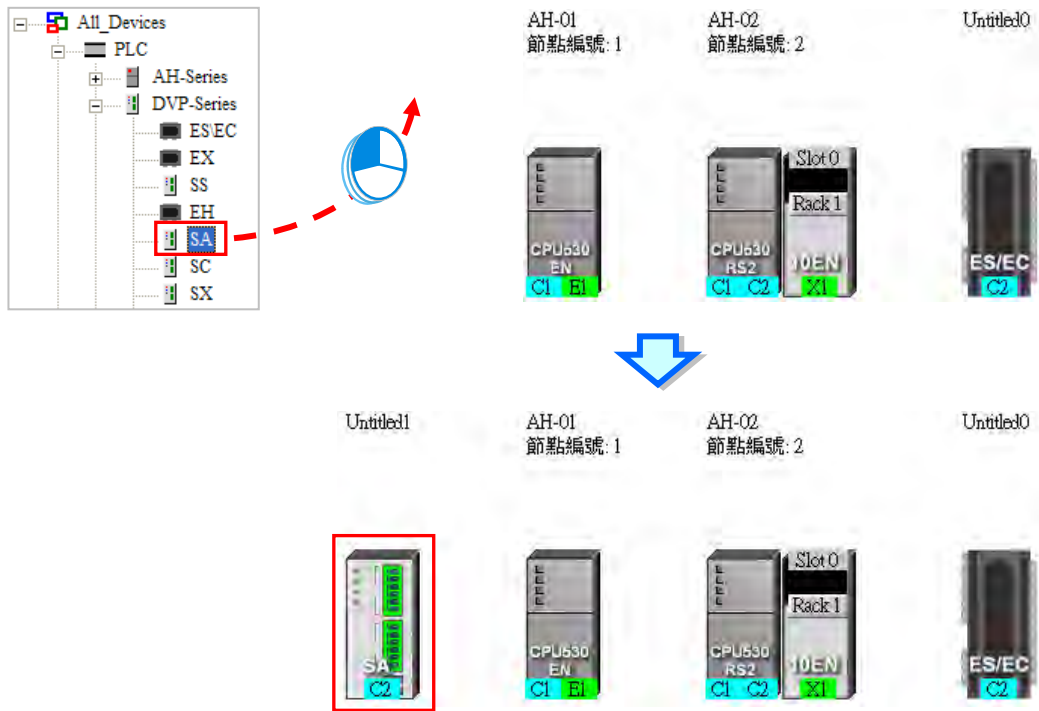
- 方法一

於左側的**裝置列表**中選擇欲新增的主機或裝置型號後，於該項目上雙擊滑鼠左鍵，之後該主機的圖示便會被排列在所有裝置的右側。

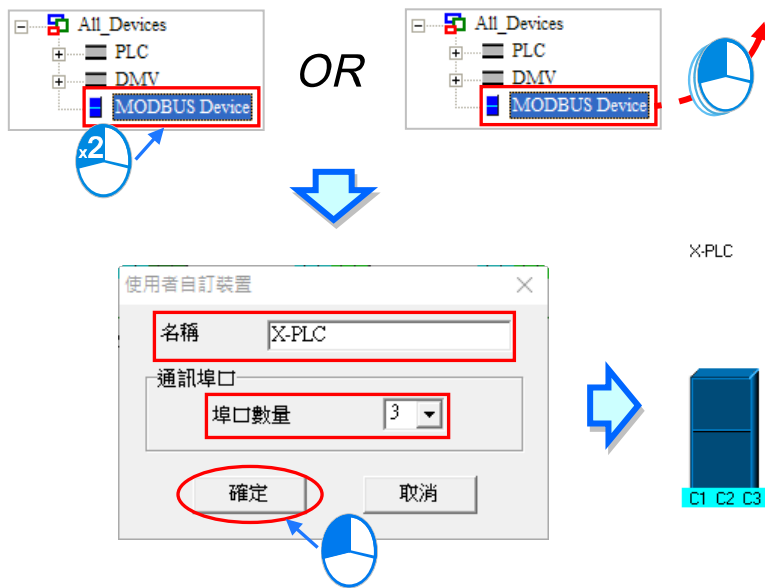


● 方法二

於左側的**裝置列表**中點選欲新增的主機或裝置型號後，直接將其拖曳至欲放置的位置即可。



在 NWCONFIG 中，使用者亦可藉由自行定義的方式加入標準 MODBUS 的裝置；操作時請先於**裝置列表**中點選「**MODBUS Device**」的圖示，接著依據下圖兩種方式將其加入工作區中；彈出使用者自定裝置視窗時於「**名稱**」欄位輸入該裝置的識別名稱，接著於「**埠口數量**」下拉選擇此裝置的埠口數量，至多可設定 3 個，完成後按下「**確定**」即可。

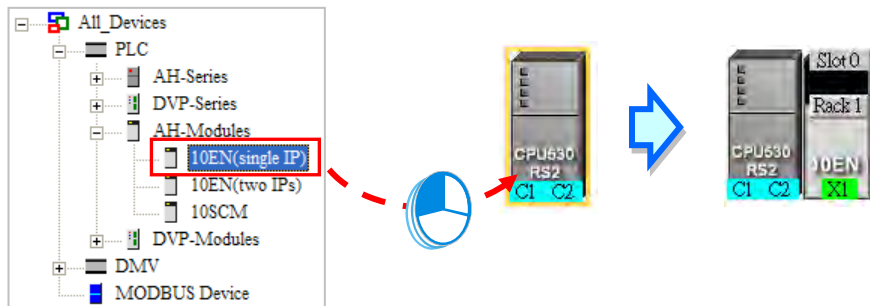



依據專案內容而自動帶出來的第一台 AH5x0 系列主機，其模組配置是與 HWCONFIG 同步，所以將無法於 NWCONFIG 當中為其加上模組，若要改變其配置必須到 HWCONFIG 設定。

欲加入通訊模組時亦可透過下列兩種方式。

● 方法一

於左側的**裝置列表**中點選模組型號後，直接將其拖曳至欲附掛該模組的 PLC 主機即可。

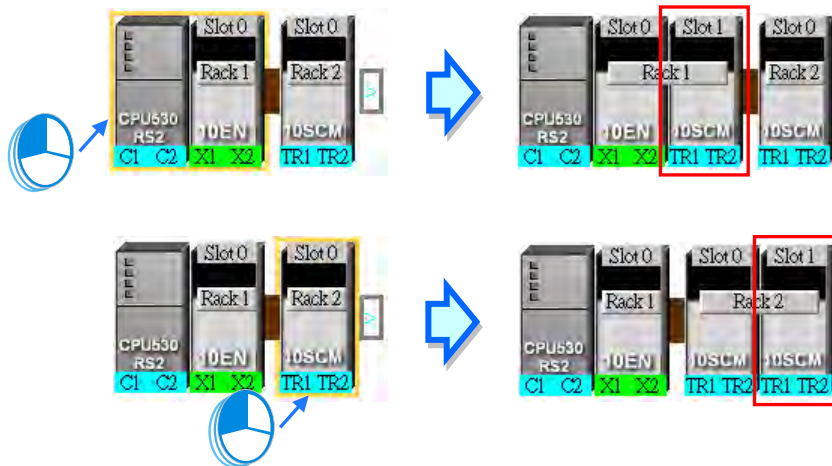



但若該主機本身不支援所選擇的模組時，鼠標便會變為  的圖示。

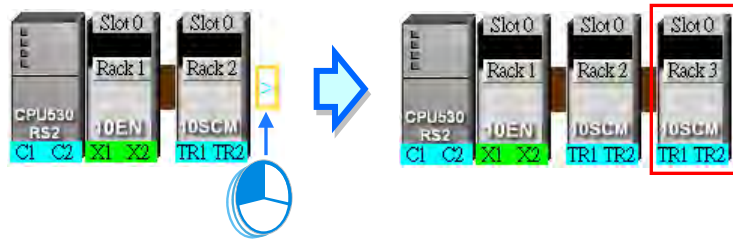


在 AH5x0 系列的機種中，10EN 模組僅可放置於主背板上，且至多可放置 8 個，而 SCM 模組則可以附掛在延伸背板上，且數量的限制會依據每個背板的插槽數量而定。

當透過拖曳的方式來加入 SCM 模組時，使用者還可決定將其拖曳至哪個背板上放置。

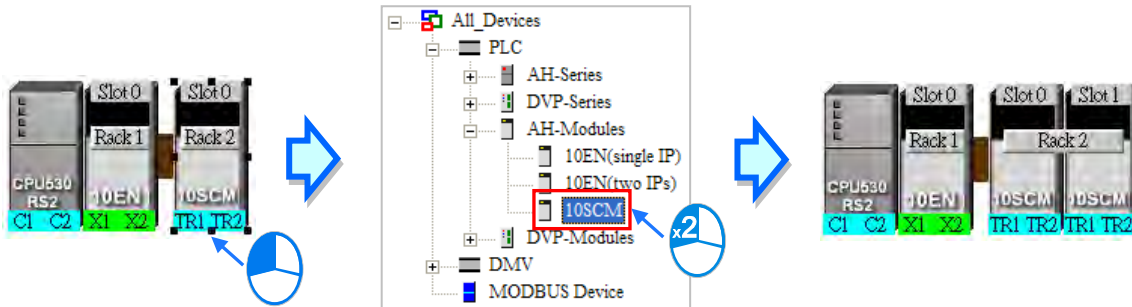


當要將 SCM 模組放置在其他的背板上時，請將其拖曳至該節點右側的  圖示處放開，此時於該節點的圖示上便會自動新增一個延伸背板。



● 方法二

先於圖示工作區點選欲加上模組的主機或背板，之後再於左側裝置列表的模組型號上雙擊滑鼠左鍵。



不論透過哪種方式，新增模組的背板與插槽編號都可能與實際上的配置不同，因此加上模組之後可能仍必須重新調整節點的屬性。



20.2.2 連結網路

完成節點的部署之後，接下來便是要將所有的節點依據事先的規劃連上所屬的網路。

於工作區中新增網路的作法共有下列幾種方式，而所支援的網路型態則有 Ethernet 與 RS485 兩種。

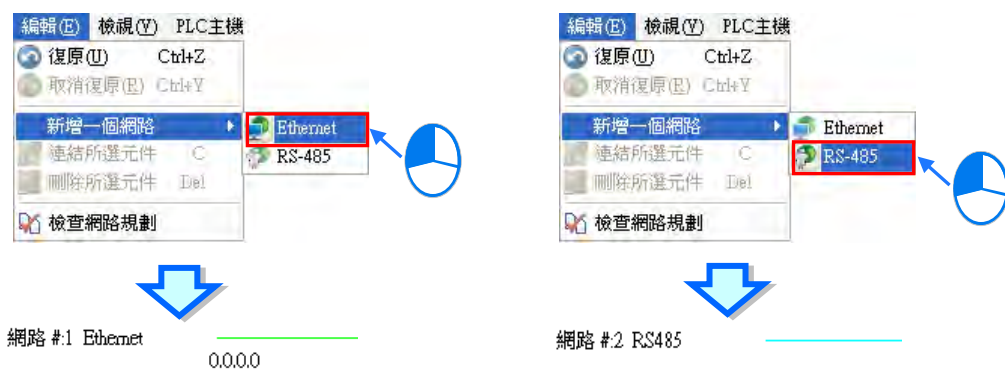
● 方法一

點擊圖示工具列的  圖示便可新增一 Ethernet 網路，而點擊  圖示則可新增一 RS485 網路。



● 方法二

於功能工具列中點選 **編輯 (E) > 新增一個網路**，之後再於其下層選項中選擇網路型態。



● 方法三

於工作區的空白處點擊滑鼠右鍵，之後於快捷選單中點選「新增一個網路」後於其下層選擇型態。



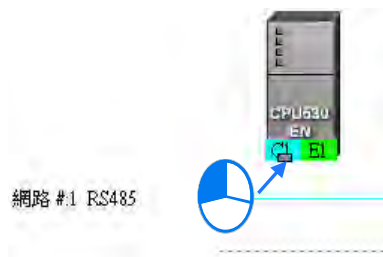
在每個節點圖示的下方都會顯示該節點的通訊口，其中藍色代表 RS485，而綠色則代表 Ethernet，至於通訊埠口上的編號則會與該裝置所定義的埠口編號一致，例如 E1 便代表 Ethernet 的第一個端口，C1 代表 COM 1，C2 則代表 COM 2；不過在 SCM 的模組圖示上，端口編號則會顯示 TR1 與 TR2，而 10EN 的端口則會顯示為 X1 或 X2，其編號的方式主要是要與實際模組上的埠口標示一致；當某個 Ethernet 通訊埠的 IP 位址被設為動態配置(DHCP)或是 SCM 模組的某個端口模式被設為 MODBUS 以外的型式時，該端口的顏色將會呈現灰色且無法連接至任何網路。



要將工作區中的節點連上網路的方式也有多種，說明如下。

● 使用滑鼠拖曳來將裝置連結至網路

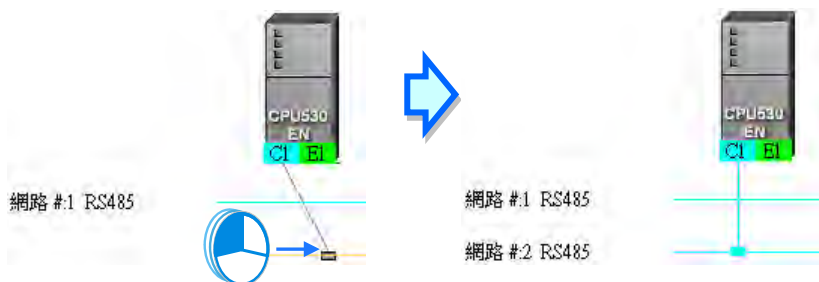
首先要在要連線的裝置通訊口上按住滑鼠左鍵不放，此時於現有網路的最下方會出現一條虛線。



將滑鼠拖曳至已建立的網路上時，若該網路的型態與通訊口一致，則放開滑鼠後便可建立連線。




若將滑鼠拖曳至最下方的虛線上放開，則系統便會先新增一條相同型態的網路後，再將其接上。

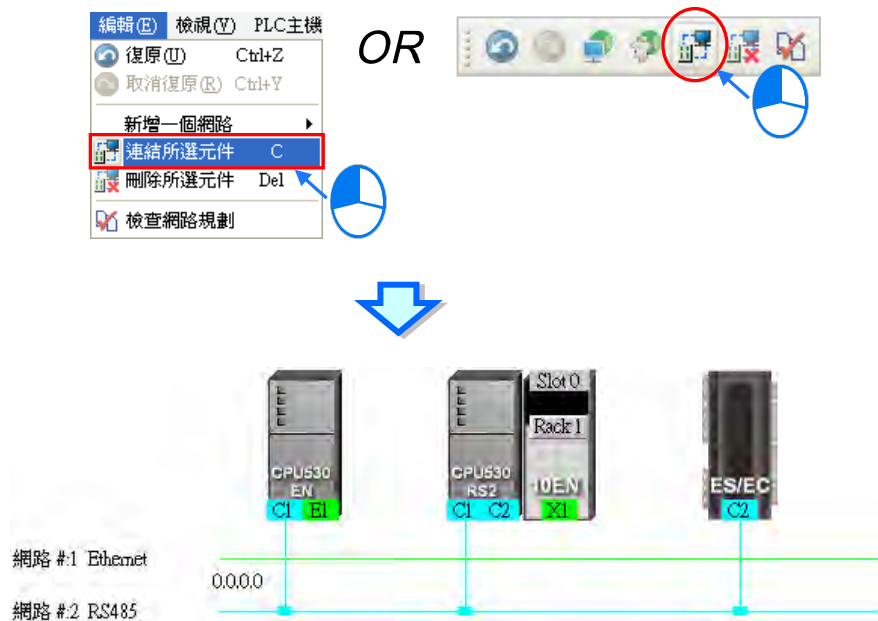


● 連接單個或多個裝置至現有網路

- (1) 按住鍵盤的【Shift】鍵後，用滑鼠依序點選欲連線的裝置與網路，而選取時須遵守下述兩個原則。
- 選取時，主機與模組須視為獨立的個體，而所選取的裝置至少都須保留一個未連線，且與選取網路相同型態的通訊口。
 - 同時間可選取多個裝置，但只能選取一條網路。



- (2) 於功能工具列中點選 **編輯 (E) > 連結所選元件**，或點擊圖示工具列的  圖示，之後系統便會自動將所選取的裝置連接至選取的網路上。

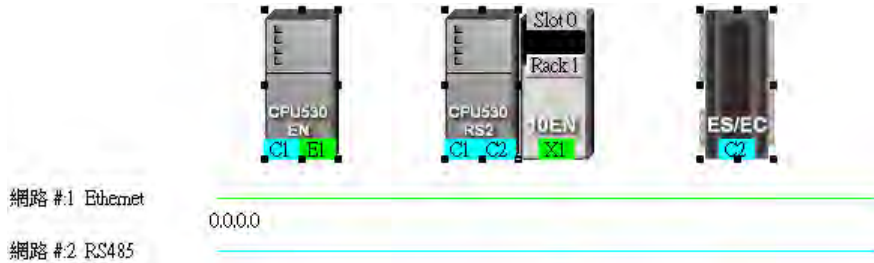



補充說明

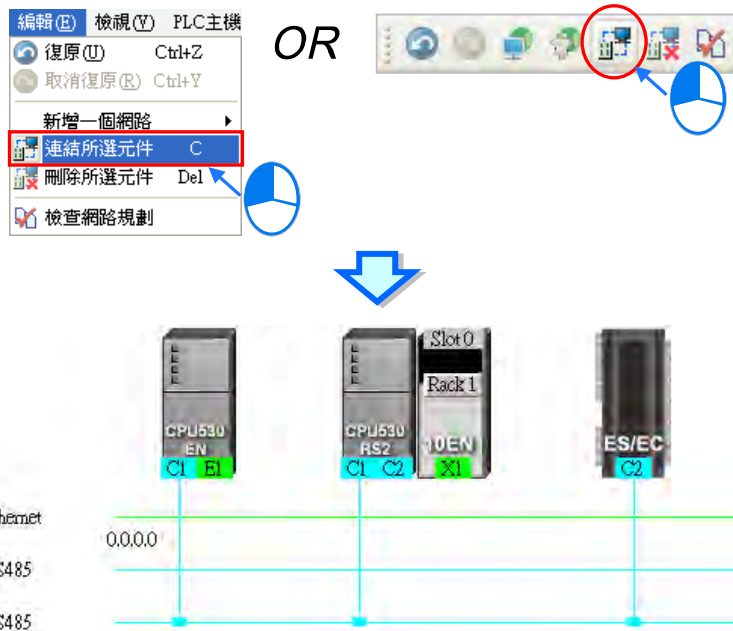
- 當選取的物件不符合上述 (1) 的兩個原則時，系統將不會建立連線，並且會出現警示訊息。
- 當選取的裝置包含一個以上符合連線條件的通訊口時，系統會優先連結編號較小的通訊口。
- 可使用滑鼠框選連續位置的裝置，而使用鍵盤的【Ctrl】+【A】則可選取所有的裝置與網路。
- 選取的物件是以裝置為單位，因此若同時選取某個節點的主機與模組，且兩者都符合上述的原則時，兩者的通訊口都會同時被連結至所選取的網路上。

● 自動連結單個或多個裝置至一新增網路

- (1) 按住鍵盤的【Shift】鍵後，用滑鼠依序點選欲連線的裝置，但不點選網路；選取時，主機與模組須視為獨立的個體，而所選取的裝置至少都須保留一個未連線且型態相同的通訊口。



- (2) 於功能工具列中點選 **編輯 (E) > 連結所選元件**，或點擊圖示工具列的  圖示，之後系統便會自動新增一條網路，並將所選取的裝置連接至該新增網路上。



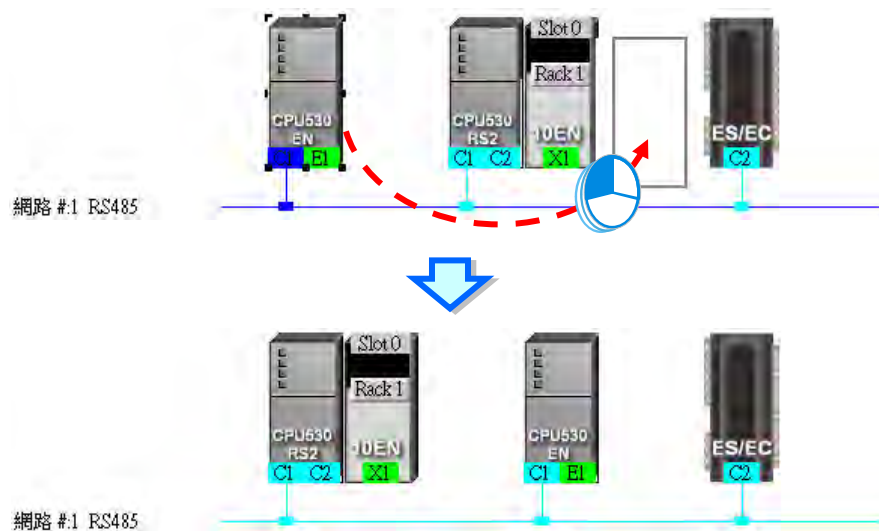
補充說明

- (a) 當選取的裝置不符合上述 (1) 的選取原則時，系統將不會建立連線，並且會出現警示訊息。
- (b) 當選取的裝置包含一個以上符合連線條件的通訊口時，系統會優先連結編號較小的通訊口。
- (c) 可使用滑鼠框選連續位置的裝置；而使用鍵盤的【Ctrl】+【A】則可選取所有的裝置與網路。
- (d) 選取的物件是以裝置為單位，因此若同時選取某個節點的主機與模組，且兩者都符合上述的選取原則時，兩者的通訊口都會同時被連結至網路上。
- (e) 在進行上述操作時，系統將會自動偵測選取裝置中空閒且型態一致的通訊端口，但若同時符合 RS-485 與 Ethernet 兩種型態時，系統將會優先建立 Ethernet 的連線。

20.2.3 調整或刪除已部署的裝置及網路

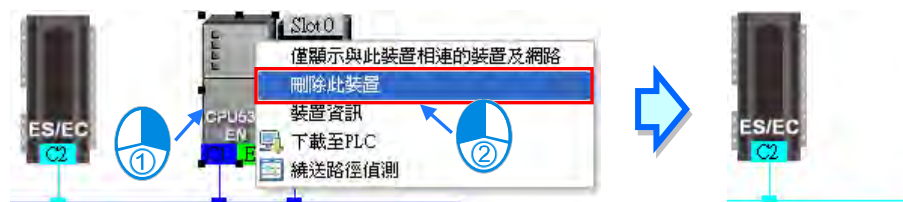
● 調整節點的排列順序

於工作區中的某個節點圖示上按住滑鼠左鍵，之後便可將其拖曳至其它的位置上擺放，如此便可改變節點圖示的排列順序；且因 NWCONFIG 中的節點圖示只能放在同一高度並持續向右方擴展，因此無法將節點圖示拖曳至上方或下方的位置擺放。

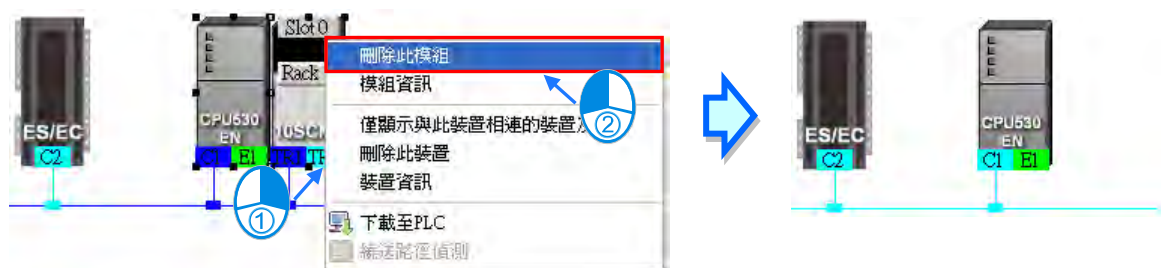


● 透過右鍵選單刪除單一裝置

要刪除整個節點時，請在主機的圖示上點擊滑鼠右鍵，接著於快捷選單中點選「刪除此裝置」，之後連同主機與附掛的模組便會一併被刪除；但依據專案設定而自動帶出來的主機或模組則無法刪除。

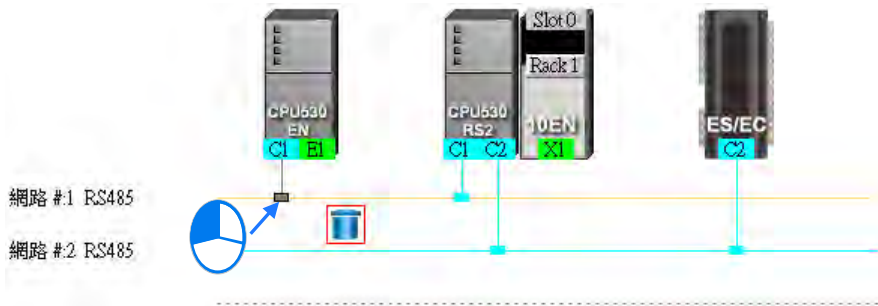


若只要刪除模組時，請點選欲刪除的模組之後按下滑鼠右鍵，接著於快捷選單中點選「刪除此模組」，之後該模組便會被刪除。



● 調整網路連結

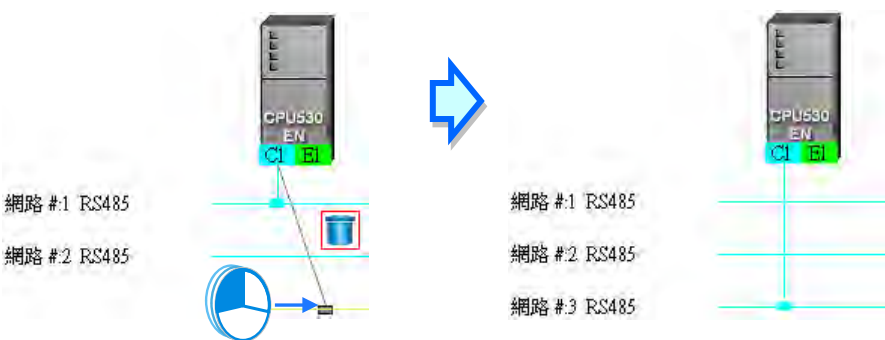
於網路上欲調整連結的接點上按住滑鼠左鍵，此時於該節點的右下方會出現一個垃圾桶的圖示，而在網路的最下方也會出現一條虛線。



若將滑鼠拖曳至垃圾桶圖示放開時，該端口的連線便會被取消。



若將滑鼠拖曳至下方的虛線上時，系統便會先新增一條網路後，再將通訊口連接至該網路上。

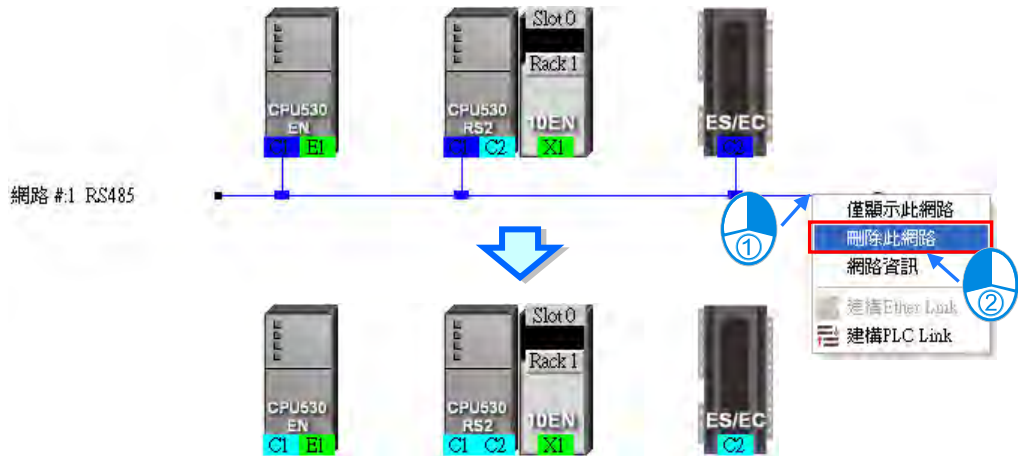


當將滑鼠拖曳至另一相同型態的網路上放開後，便可將該通訊口的連線變更至該網路。



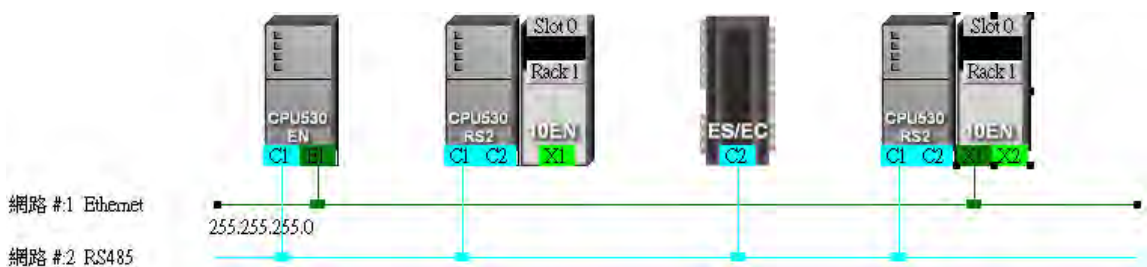
● 透過右鍵選單刪除單一網路


在欲刪除的網路上點擊滑鼠右鍵，接著於快捷選單中點選「刪除此網路」，之後該網路便會被刪除，而原本連接至該網路的接線也會一併消失。

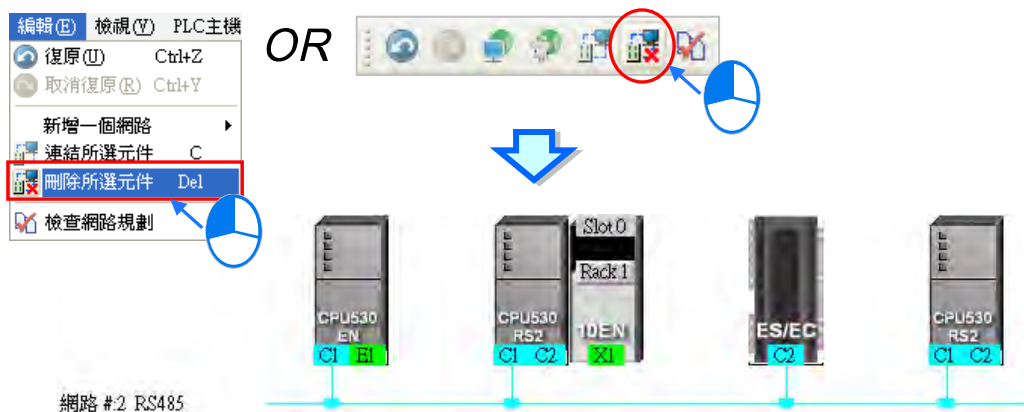


● 同時刪除多個裝置或網路

首先請於工作區中點選欲刪除的物件，而按住鍵盤的【Shift】鍵時，便可用滑鼠同時點選多個元件；也可用滑鼠框選連續位置的元件，或使用鍵盤的【Ctrl】+【A】將所有的物件同時選取起來。

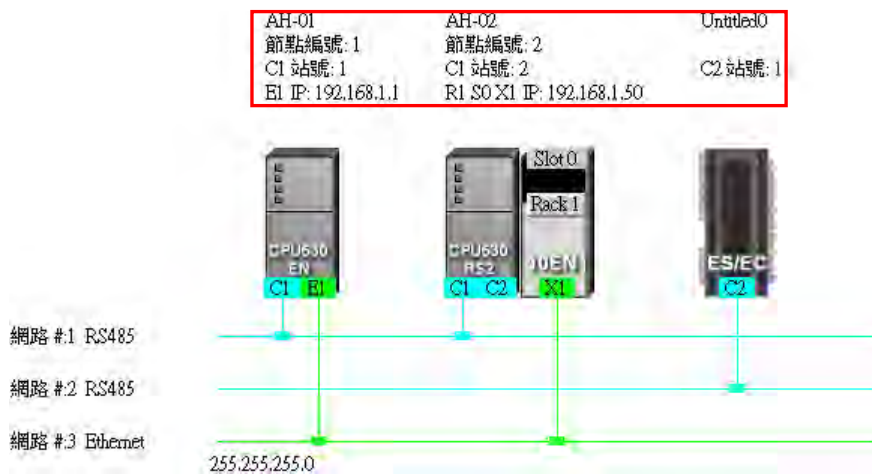


選取完要刪除的元件後，於功能工具列中依序點選 編輯 (E) > 刪除所選元件，或點擊圖示工具列的  圖示，亦或是直接按下鍵盤的【DEL】鍵即可刪除所選的元件，不過若是依據專案設定而自動帶出來的節點則無法將其刪除；此外，刪除主機時，附掛於該主機的模組亦會一併被刪除。

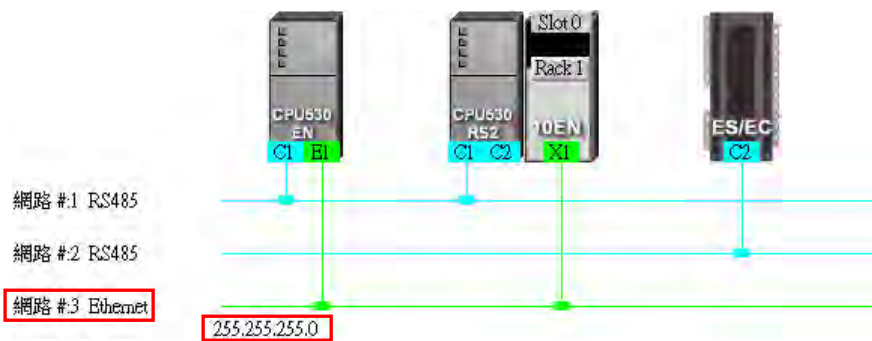


20.2.4 節點與網路屬性的設定

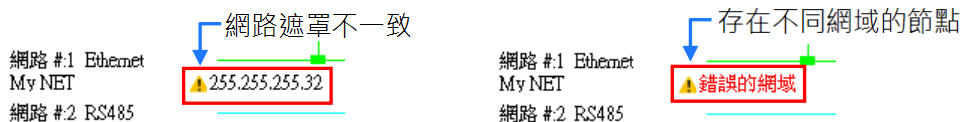
當完成網路的部署之後，於各個節點的上方會顯示該節點的相關資訊，而顯示的內容除了 PLC 名稱與節點編號之外，同時還會顯示有連結至網路的通訊埠屬性，若是 RS485 端口則會顯示站號，Ethernet 端口則會顯示 IP 位址；而其中只有連接網路的通訊埠才會有資訊顯示，未連接任何網路的端口將不會顯示資訊；屬於通訊模組的端口資訊還會顯示背板與插槽編號以便於識別。



除了節點資訊之外，於各個網路的左側亦會顯示該網路的編號與型態，而在 Ethernet 的網路上還會顯示該網路的網路遮罩設定。



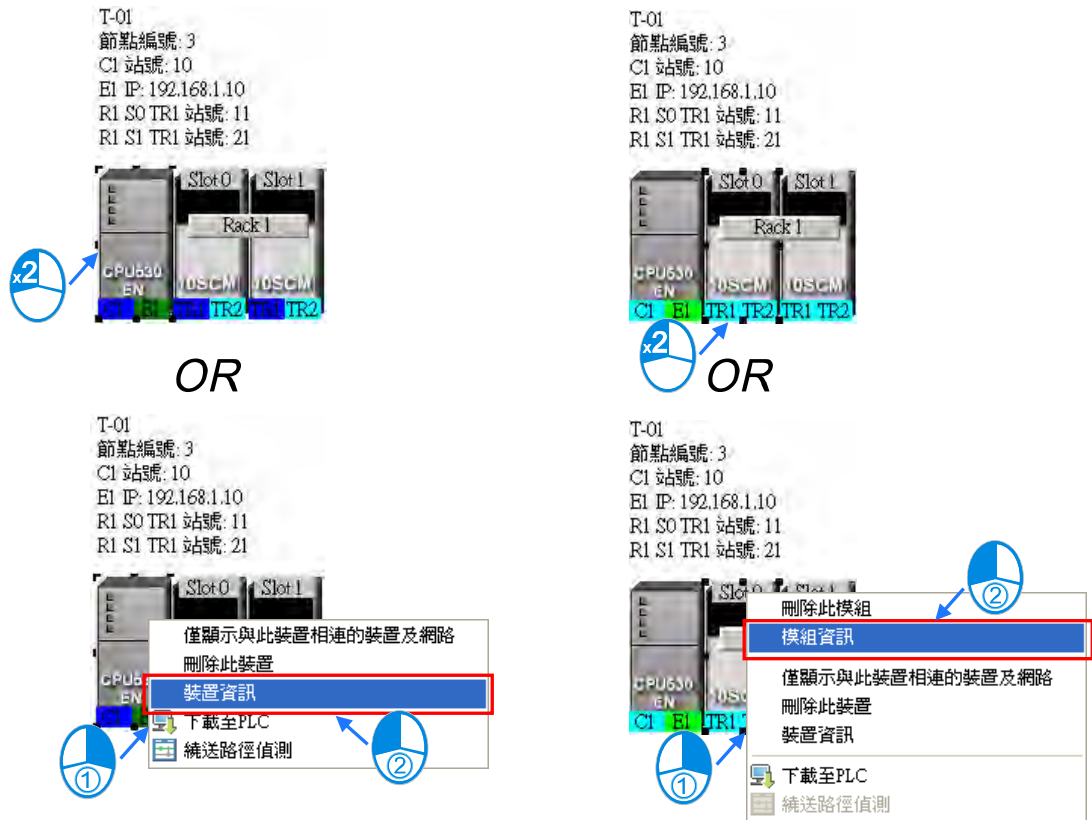
當連結至同一個網路編號的 Ethernet 端點中，若其中有某個端點所設定的網域不同，亦或是網路遮罩設定不一致時，在該網路上便會出現警示的圖示；而當網路遮罩設定不一致時，該網路上的網路遮罩值將會顯示最嚴謹的那一組。



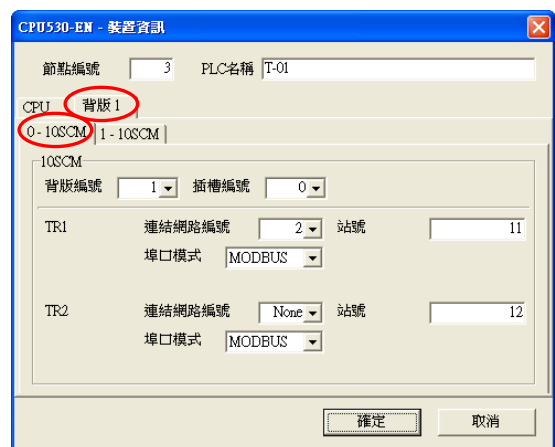
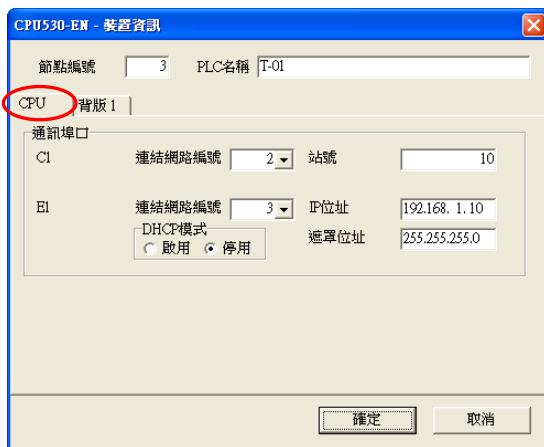
● 設定節點屬性

要開啟節點屬性視窗時，可透過下列兩種方式。

- (a) 直接於要設定屬性的主機圖示上雙擊滑鼠左鍵，或點選主機圖示後按下滑鼠右鍵，接著於快捷選單中點選「裝置資訊」。
- (b) 直接於要設定屬性的模組圖示上雙擊滑鼠左鍵，或點選模組圖示後按下滑鼠右鍵，接著於快捷選單中點選「模組資訊」。

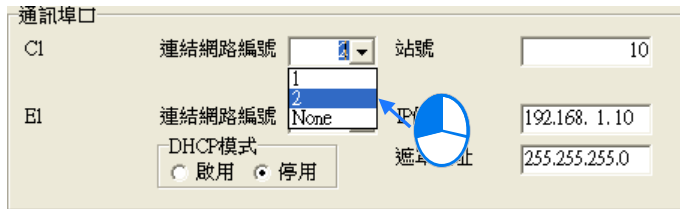


在節點屬性的設定視窗中，主機與模組的屬性會被放在不同的頁面，而使用者則可透過上方的標籤來進行切換，且開啟視窗時，系統將會依據使用者所點選的裝置而自動切換到對應的頁面。

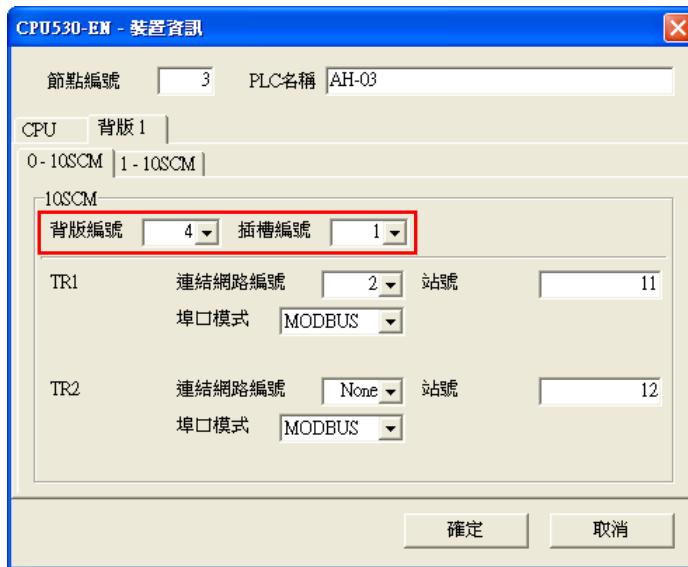


若所點選的裝置是依據專案設定而自動帶出來的第一台 AH5x0 系列機種，則開啟的屬性視窗中，其屬性欄位大部份是灰階的，其屬性必須透過 HWCONFIG 才可進行設定。

在設定頁面中，每個通訊口都會有一個對應的「**連結網路編號**」欄位，於下拉選單中便會出現可連結的網路編號選項；若選擇 **None** 時，則該端口的網路連線便會被刪除。



在模組的設定頁面中還可重新定義模組的**背板與插槽編號**。

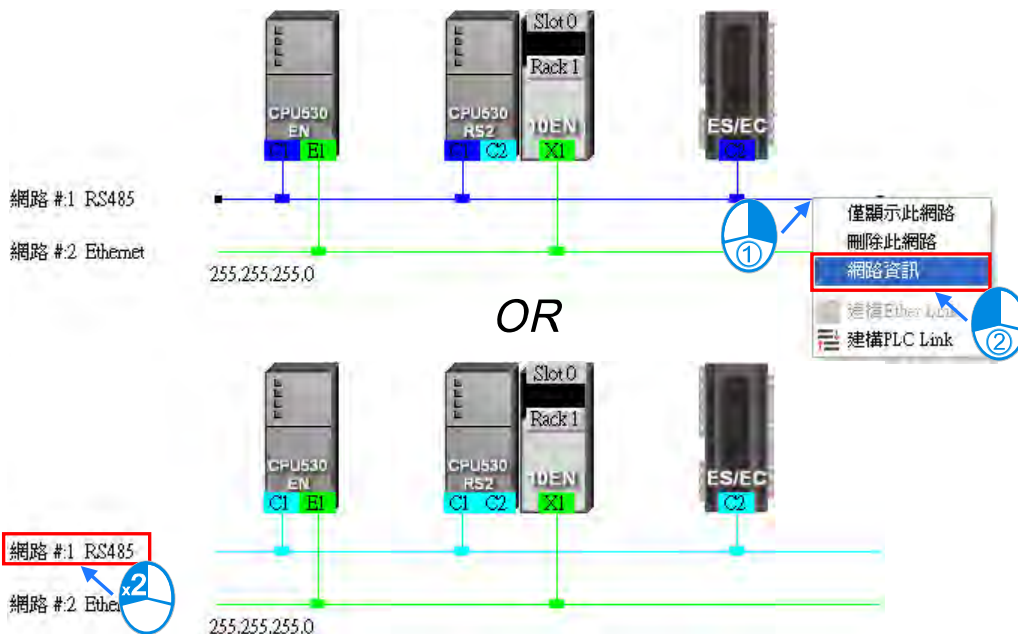


完成設定後請按下「**確定**」鍵，而此時該節點的狀態與上方的資訊便會立即更新。

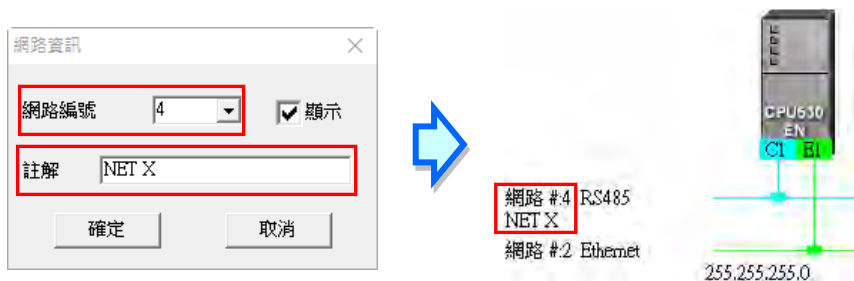


● 設定網路屬性

點選欲設定屬性的網路後按下滑鼠右鍵，之後於快捷選單中選擇「網路資訊」，亦或直接於該網路左側的文字上雙擊滑鼠左鍵，之後便會開啟網路屬性的設定視窗。



於網路屬性的設定視窗中，我們可重新定義該網路的編號，但只可選擇尚未被使用的網路編號，而除此之外，我們尚可為該網路加上註解；完成設定之後按下「確定」鍵便可立即套用所做的變更。

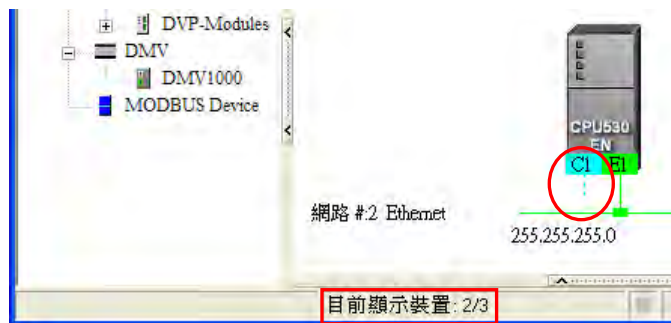


網路資訊視窗中的「顯示」項目取消勾選後，此網路會隱藏且有連的通訊口會以虛線來表示。



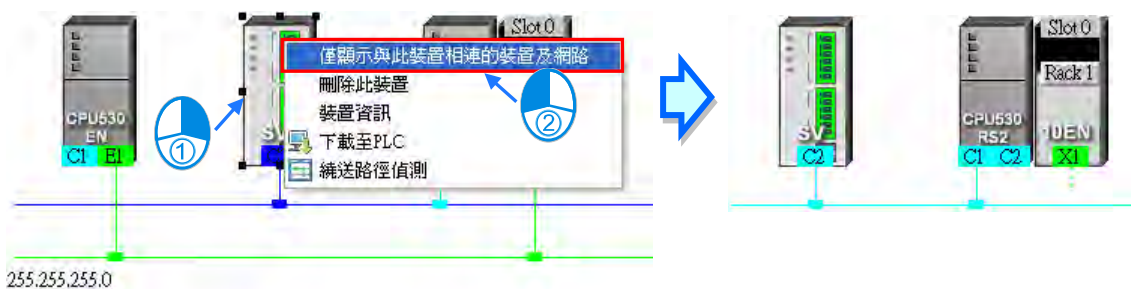
20.2.5 裝置或網路的隱藏/顯示

為了便於編輯與檢視，圖示區的裝置與網路皆可由使用者自行定義其顯示與隱藏的狀態，且由下方的狀態列中便可知目前顯示的裝置與總共的裝置數量。當某個裝置的端口有虛線時，便代表該端口有連結至某個被隱藏的網路。



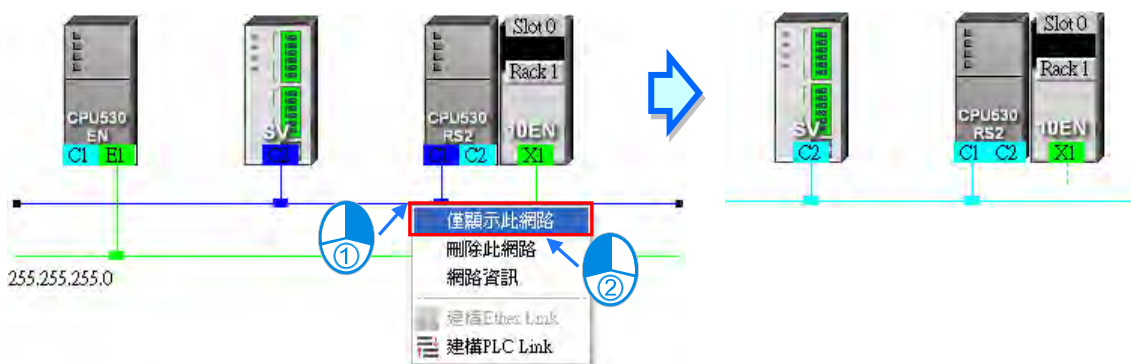
- 只顯示與某一裝置相關的物件

於某一節點的圖示上點擊滑鼠右鍵後，在快捷選單中選取「僅顯示與此裝置相連的裝置及網路」，之後圖示區便只會顯示與此節點直接相連的網路與裝置。



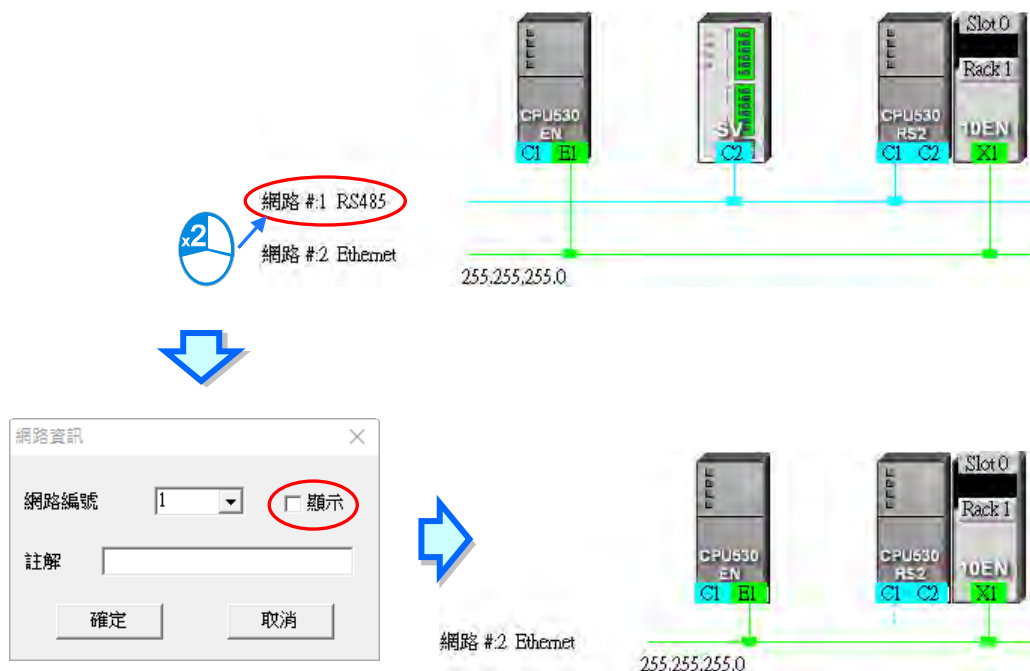
- 只顯示某一網路上的裝置

在某一網路上點擊滑鼠右鍵後，接著再於快捷選單中選取「僅顯示此網路」，之後圖示區便只會顯示與此網路直接相連的裝置。



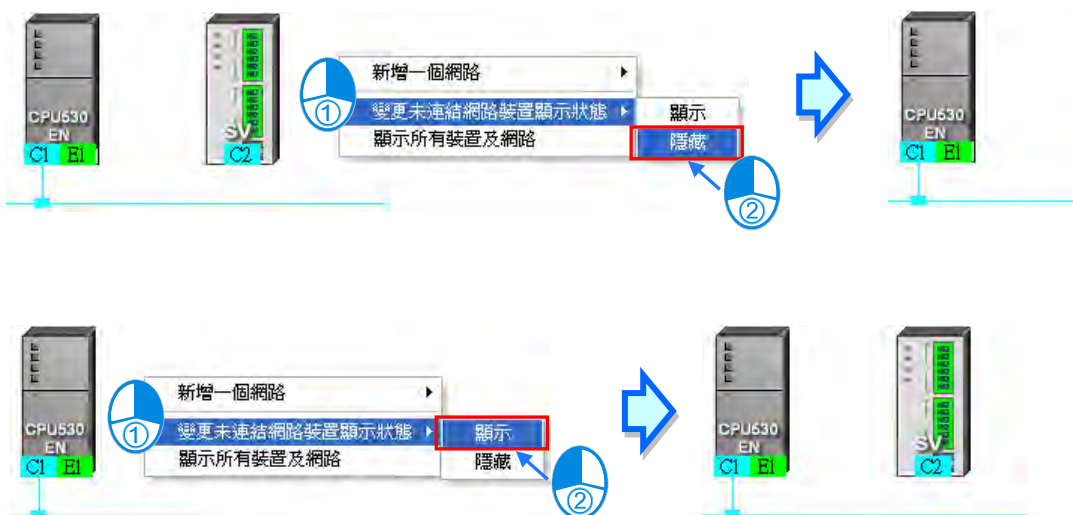
● 隱藏某一網路與連接至該網路的裝置

開啟網路屬性視窗後，將「顯示」項目的勾選取消，此網路會隱藏且有連的通訊口會以虛線來表示。




● 隱藏/顯示未連接的裝置

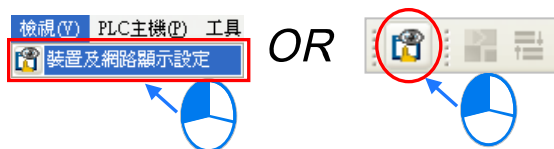
要隱藏或顯示圖示工作區中未連接至任何網路的裝置時，請先於圖示工作區的空白處點擊滑鼠右鍵，接著在快捷選單中選取「變更未連結網路裝置顯示狀態」，之後便可於下階的選單中設定裝置的顯示狀態；不過每次操作時僅會對目前工作區中的裝置進行設定，而後續加入的裝置則不受其影響。



● 批次設定所有元件的顯示狀態

利用此功能便可以網路為單位，而同時設定所有裝置的顯示狀態。

首先請於功能工具列中點選 **檢視(V) > 裝置及網路顯示設定**，亦或直接點選圖示工具列的  圖示便可開啟設定視窗。



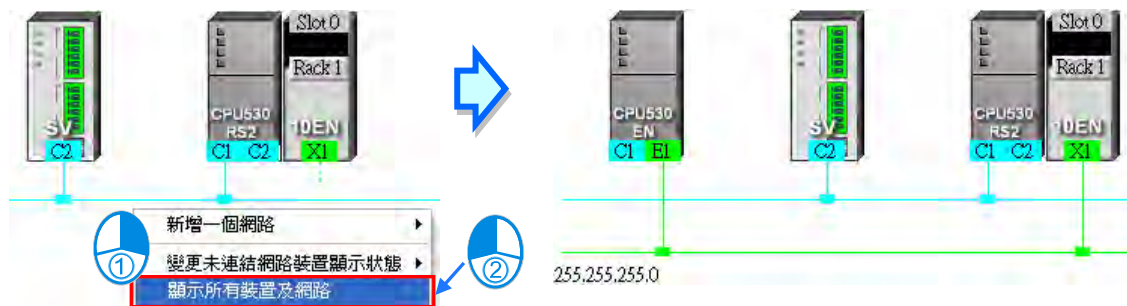
於設定視窗的「網路」區域中勾選欲顯示的網路編號，未勾選的選項即代表設為隱藏，而透過「全選」的項目則可同時勾選或取消所有項目的勾選狀態。

在「無連結網路裝置」的區域中，使用者可設定目前圖示區中未連接至任何網路的裝置顯示狀態；而同樣的，每次的操作僅會對目前工作區中的裝置進行設定，而後續加入的裝置則不受其影響。




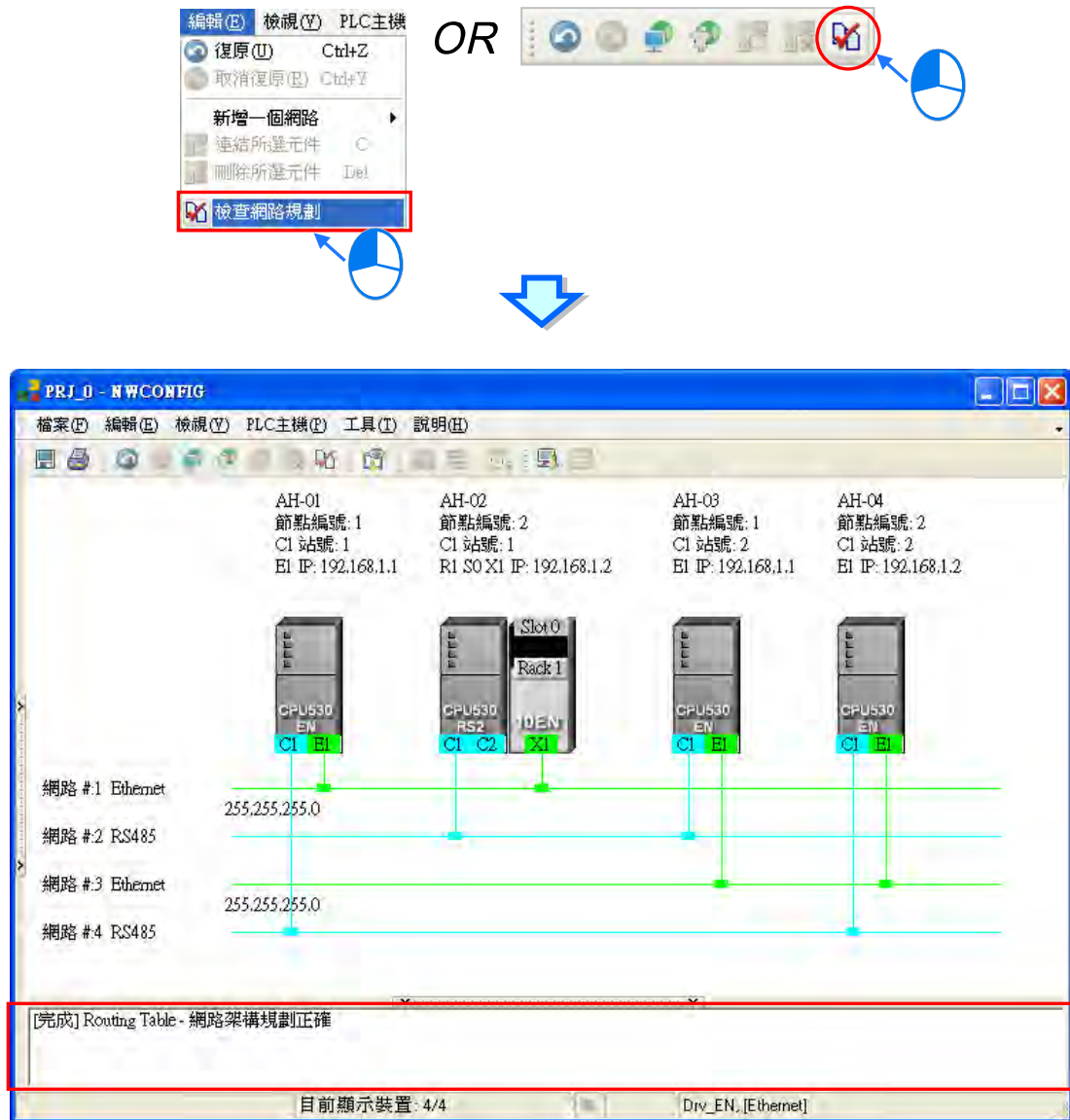
● 顯示所有元件

於圖示工作區的空白處點擊滑鼠右鍵後，在快捷選單中選取「顯示所有裝置及網路」，之後所有被隱藏的裝置及網路便會重新恢復顯示。



20.2.6 合法的網路架構

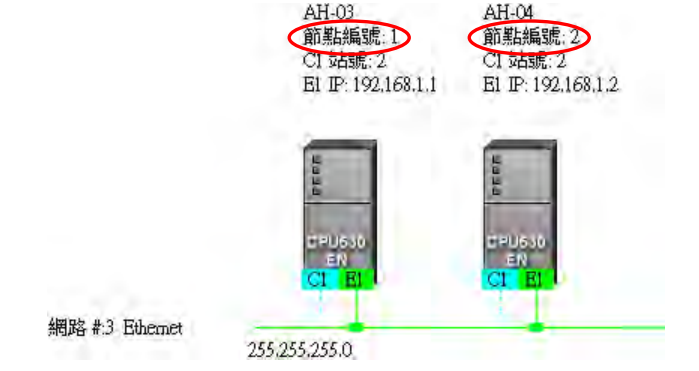
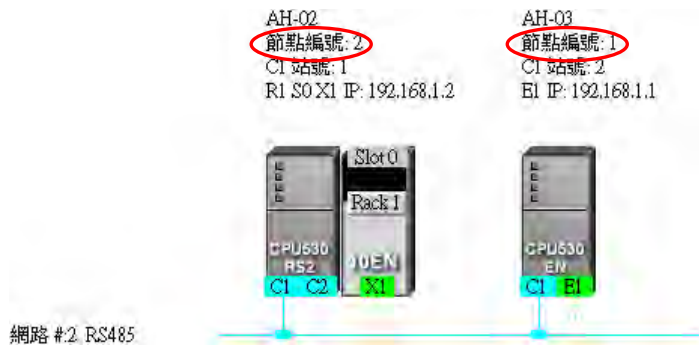
當完成網路架構的部署之後，於功能工具列中依序點選 **編輯 (E)** > **檢查網路規劃**，亦或於圖示工具列中點選  圖示，之後便可對目前所部署的網路進行合法性的檢查，而檢查的結果則會顯示於下方的訊息視窗中。



在進行檢查時，系統僅會針對有連接網路的端口進行確認，也可透過顯示於圖示工作區的節點或網路資訊來確認規劃是否正確，而主要的檢查項目則包括節點編號、RS485 站號與 IP 位址；若依據上圖的內容來說，整體看來會覺得存在著節點編號、RS485 站號與 IP 位址重複的問題，因此我們可以每次只檢視一條網路，便可清楚的確認到連接至同一個網路上的節點，其節點編號、RS485 站號與 IP 位址有沒有發生重複的現象。

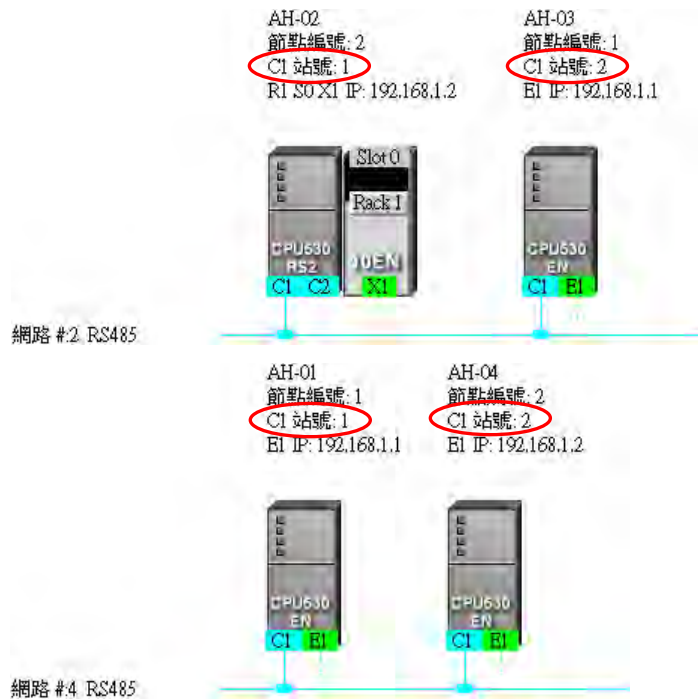
● 節點編號

原則為同一個網路上的節點編號不得重複，而透過每次只檢視一條網路的方式，我們便可清楚的確認到連接至同一個網路上的節點，其編號並沒有發生重複的現象。



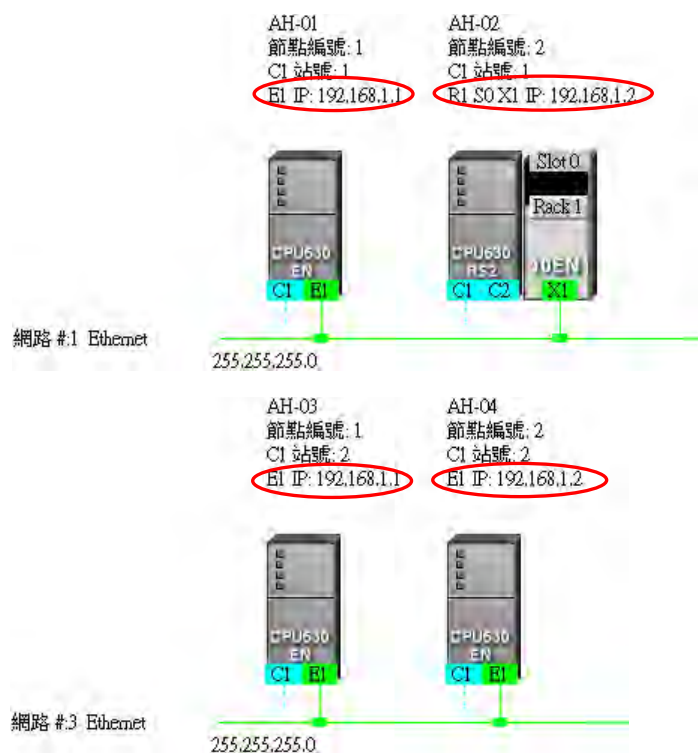
● RS485 站號

原則為同一個網路上的站號不得重複，而透過每次只檢視一條 RS485 網路的方式，我們便可清楚的確認到連接至同一個網路上的 RS485 端口，其站號亦沒有發生重複的現象。



● IP 位址

原則為同一個網路上的 IP 位址不得重複，因此我們可逐一檢視每一條 Ethernet 網路上的端口資訊，且在檢查之後便可確認同一個網路上的 IP 位址並沒有重複。



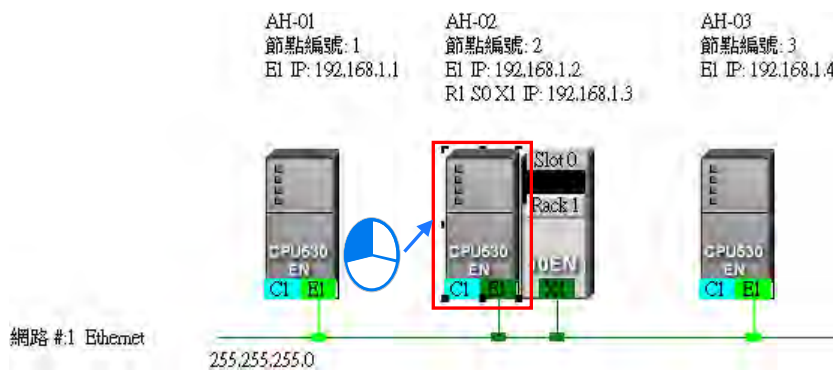
20.2.7 下載繞送路徑表 (Routing Table)


當完成網路架構的部署並檢查無誤後，接著便可將繞送路徑表 (Routing Table) 下載至各節點的主機當中，每個主機的內部只會儲存與其本身相關的路徑資料，所以不同節點會被下載的繞送路徑表 (Routing Table) 資料也會不同，而下載時也必須逐一完成每個節點的資料下載。

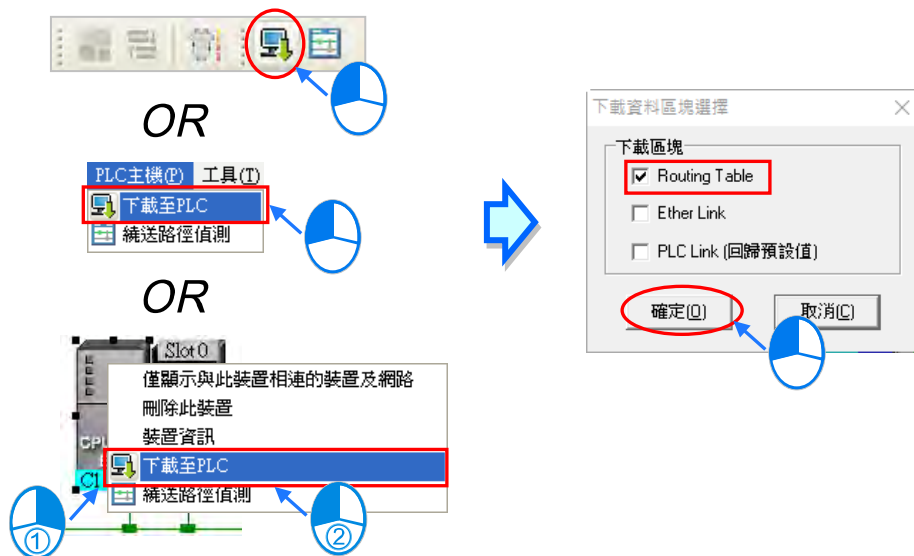
繞送路徑表 (Routing Table) 的下載方式可分為單點與多點兩種模式，且下載時的通訊設定不可使用封包繞送模式。

● 單點模式

首先選擇要下載的節點，並於圖示工作區中點選該節點的主機部分，且因繞送功能只支援 AH5x0 系列的機種，因此選擇其他機種或裝置時將無法下載繞送路徑表 (Routing Table)。

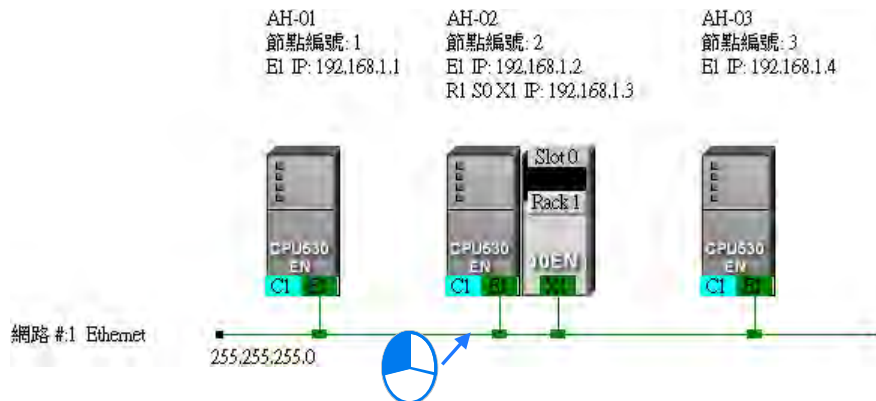



點選完要下載的主機之後，請於功能工具列中點選 **PLC 主機 (P) > 下載至 PLC**，或直接於圖示工具列中點選  圖示，同時也可於主機圖示上點擊滑鼠右鍵，之後再於右鍵快捷選單中點選「下載至 PLC」項目即可；而接下來請直接於下載選項當中勾選「Routing Table」，完成之後按下「確定」便可將繞送路徑表 (Routing Table) 下載至指定的主機當中。

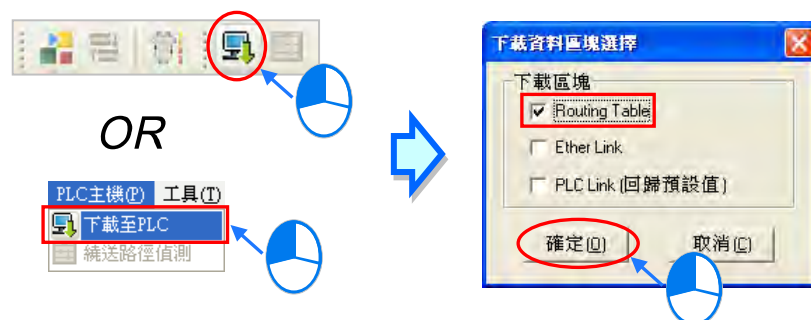


● 多點模式

通訊設定中的 COMMGR Driver 必須為 Ethernet 的型式，接著於圖示區中點選其中一條的 Ethernet 網路，之後便可將參數下載至與該網路連接的所有節點。



點選完網路之後，請於功能工具列中點選 **PLC 主機 (P)** > **下載至 PLC**，或直接於圖示工具列中點選  圖示即可；接著勾選「**Routing Table**」，完成之後按下「**確定**」鍵便可將繞送路徑表 (**Routing Table**) 下載至與該網路連接的所有節點。



補充說明

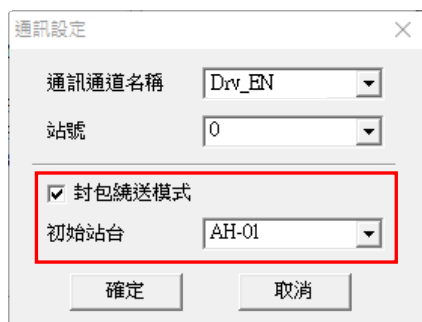
當點選的下載對象有包含 **Ether Link** 或 **PLC Link** 的參數可下載時，於下載選項的視窗中亦可將該項目一併勾選，但當勾選項目為灰階時，便代表該項目無法下載；關於 **Ether Link** 與 **PLC Link** 的相關內容將於後續的章節當中另行介紹。


因為每個主機內部只會儲存與其本身相關的路徑資料，因此無法將繞送路徑表 (**Routing Table**) 自主機內部上傳並於 NWCONFIG 的工作區中重建，因此系統並未提供繞送路徑表 (**Routing Table**) 的上傳功能。

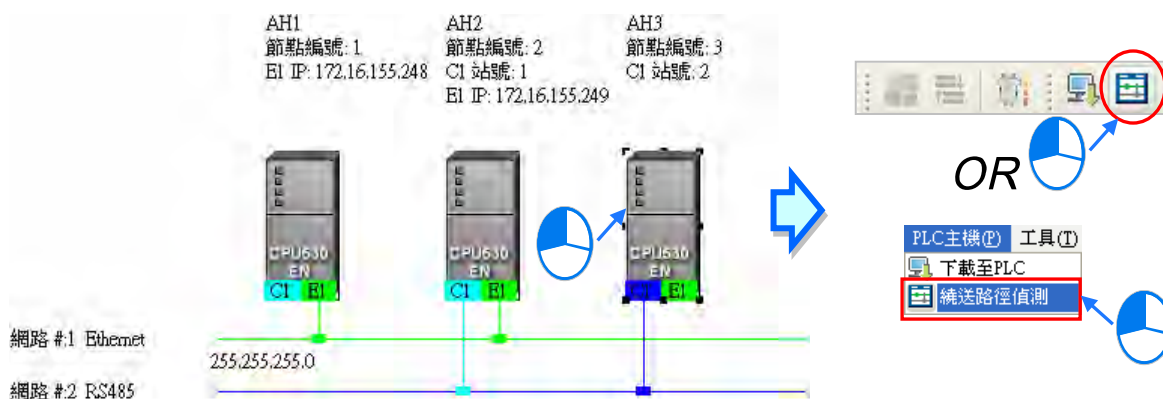
20.2.8 繞送路徑測試

當完成繞送路徑表 (**Routing Table**) 的下載之後，接下來便可透過 NWCONFIG 所提供的功能進行繞送路徑的測試。操作的步驟如下。

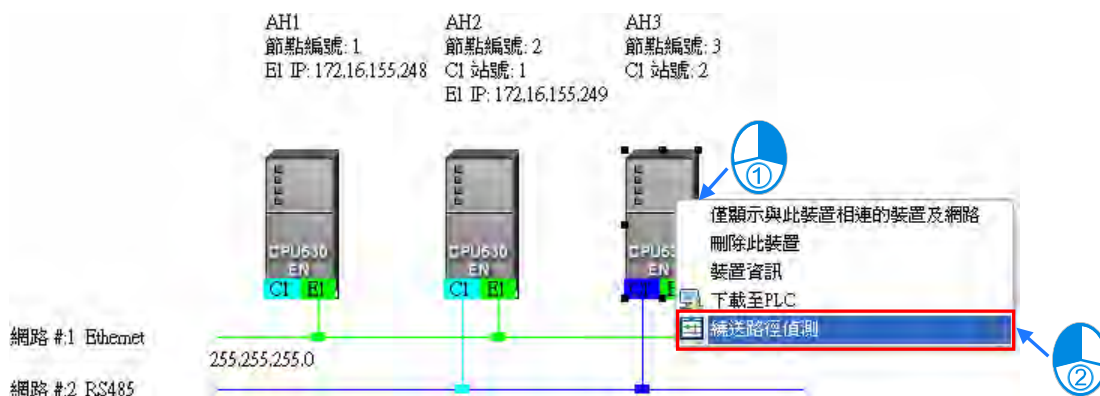
- (1) 請先確認所有節點都已依照 NWCONFIG 中的配置完成接線，且已處於正常運作的狀態，而每個裝置內部的設定也都與 NWCONFIG 中的設定一致。接著請於通訊設定視窗中啟用「**封包繞送模式**」，並選擇「**初始站台**」。



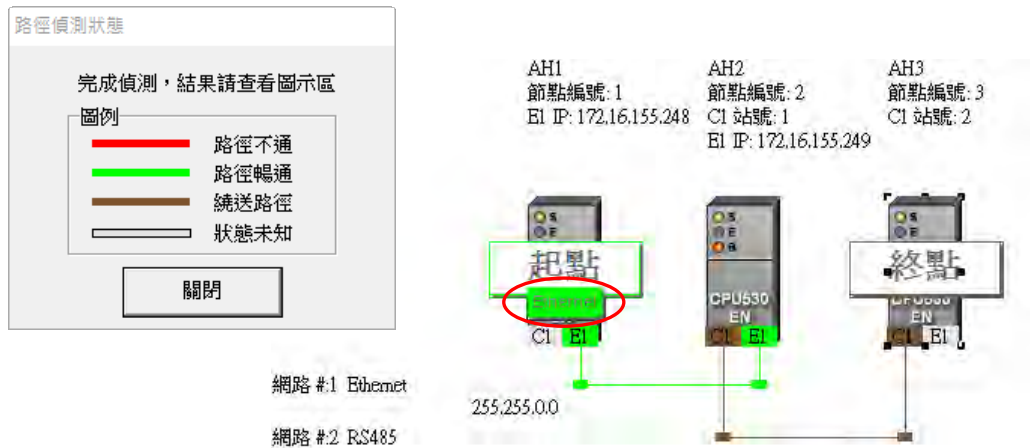
- (2) 於圖示工作區中點選測試路徑的**終點裝置**之後，直接點選圖示工具列中的  圖示，或是於功能工具列中依序點選 **PLC 主機 (P)** > **繞送路徑偵測**。



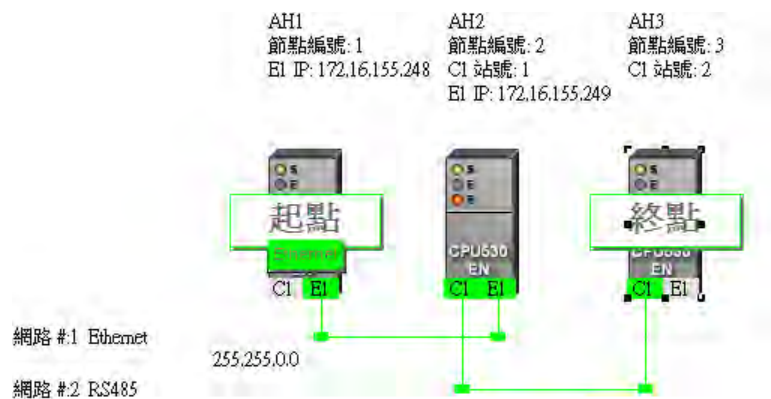
同時也可點選路徑的**終點裝置**後按下滑鼠右鍵，之後於快捷選單中點選「**繞送路徑偵測**」。



- (3) 執行「繞送路徑偵測」的功能之後，於圖示工作區中便會有對應的表達圖示，而此時畫面也會出現狀態視窗。其中於指定為**初始站台**的節點圖示中尚會顯示電腦與**初始站台**所採用的連線方式，請參考下圖中的紅框。



- (4) 完成偵測後，畫面上便會顯示最後偵測的結果。



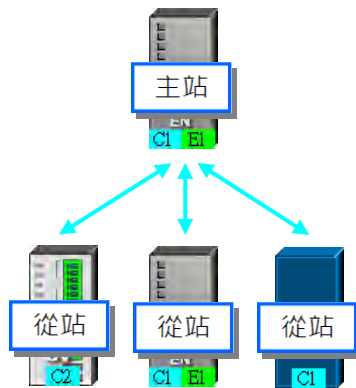
補充說明

若偵測的結果失敗時，請再次確認硬體配置都與 NWCONFIG 中的配置一致、且每個節點都已正確的連上網路並處於正常運作的狀態。

20.3 PLC Link 的建構

PLC Link 為透過 RS485 的連線來進行資料交換的網路機制。當某個 RS485 的網路上存在多個節點時，我們便可於該網路中規劃一個資料交換的機制；而當設置參數被下載至設為主站的主機且開始運行之後，透過特殊繼電器與特殊暫存器的操作，系統便可自動進行資料交換的動作。


PLC Link 是一種主從架構，一個 RS485 的網路上只能有一個主站，主站會輪流對其它從站發出讀寫的命令，而其它從站則必須被動的在接收到主站的命令後才可做出回應，因此從站與從站之間無法直接交換資料，但若在應用上需要從站之間的資料交換時，規劃時便必須透過主站來建立轉發的機制。

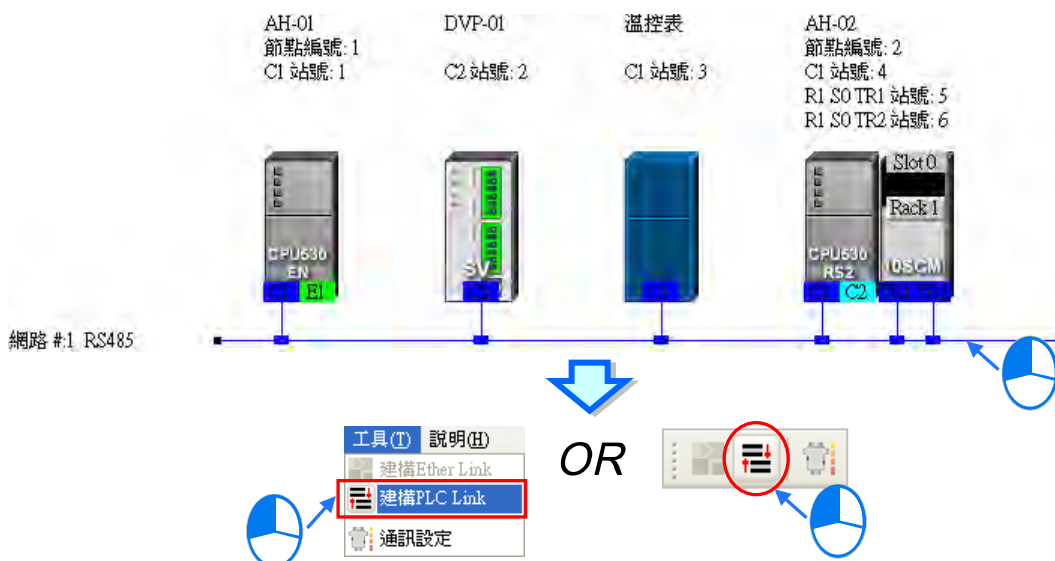


20.3.1 開啟 PLC Link 的設置視窗

開始建構 PLC Link 之前請務必確認所有的網路設置正確無誤後，依照下述方法進行操作。

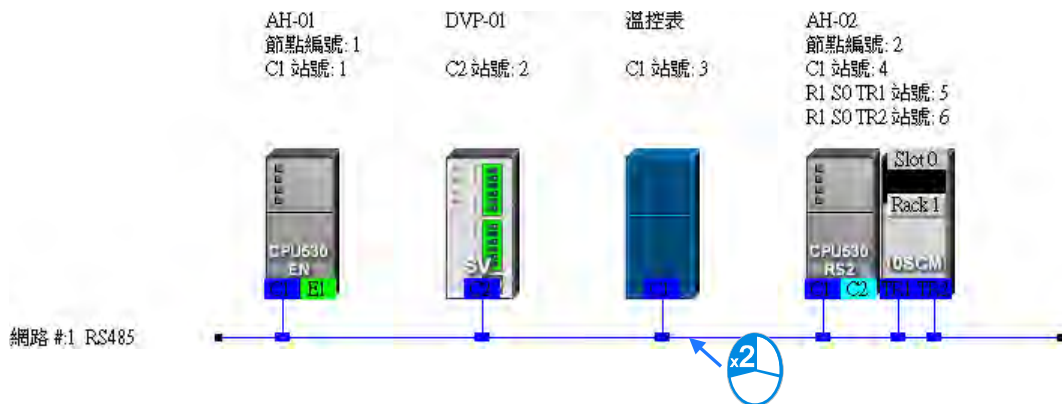
- 方法一

首先點選欲建構 PLC Link 的網路。接著於功能工具列中點選 **工具 (T) > 建構 PLC Link**，或直接點擊圖示工具列中的  圖示。



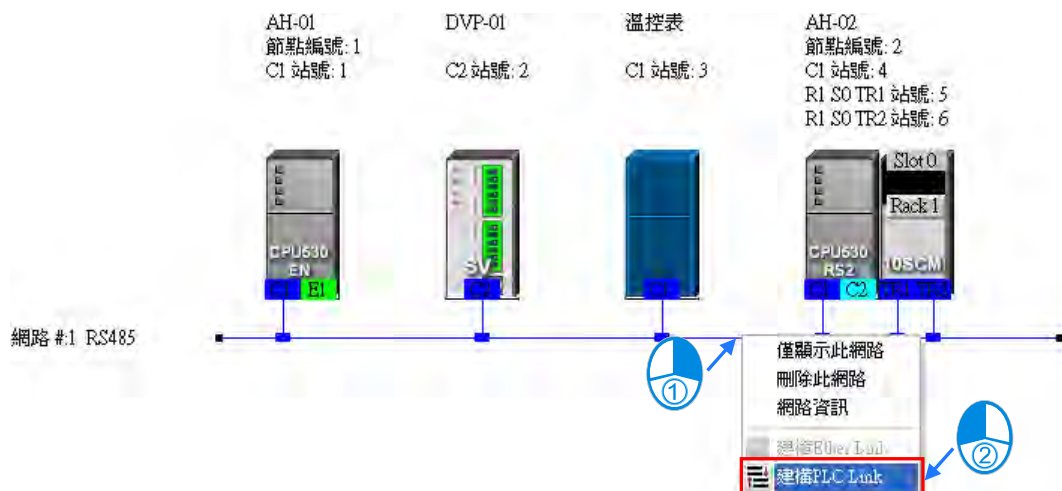
- 方法二

在要建構 PLC Link 的網路上雙擊滑鼠左鍵。



- 方法三

在要建構 PLC Link 的網路上點擊滑鼠右鍵，之後於快捷選單中點選「建構 PLC Link」。

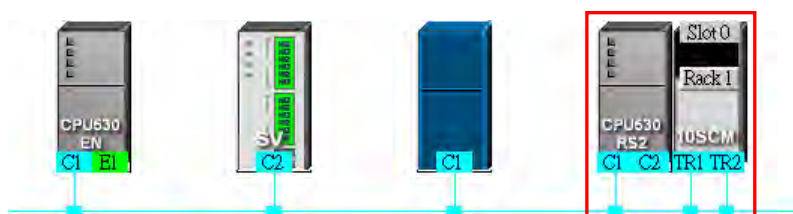


依據上述的其中一種方式操作完畢之後，畫面便會開啟 PLC Link 的規劃工具，而 PLC Link 的規劃工具則是以引導的方式，逐步帶領使用者完成 PLC Link 的建構，其主要的步驟分別是 **選擇主站裝置** → **設定通訊參數** → **設置資料交換表**，而當開啟規劃工具之後，系統將會直接進入上次的操作畫面，因此若該網路為第一次進行規劃時，開啟的畫面便會進入第一個步驟。

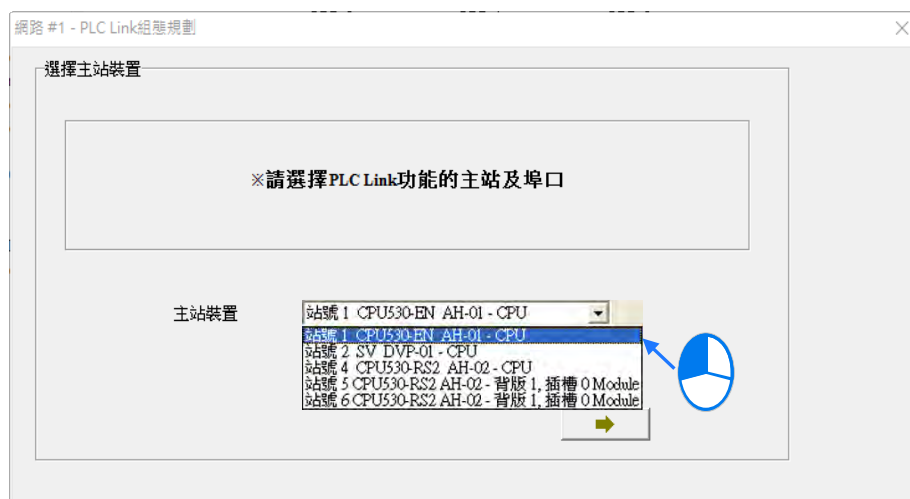
PLC Link 的實際執行是透過特殊繼電器與特殊暫存器的操作來進行，而此處所介紹的規劃工具則只是提供較為親切的使用者介面，以協助使用者將 PLC Link 的相關參數下載至對應的**特殊繼電器**與**特殊暫存器**當中，因此實際執行的情況仍會依據當時 PLC 內部的**特殊繼電器**與**特殊暫存器**而定，所以進行架構時，請務必在正式執行之前確認各主機在 PLC Link 這部份的相關功能與限制，以確保所規劃的 PLC Link 功能可正常運作。


20.3.2 選擇主站裝置 (步驟 1)

指定主站時只可選擇 AH5x0 系列或 DVP 系列的 PLC 機種，且因不是所有主機或模組的通訊口都可被指定為主站，因此設定之前請詳閱各機種與模組的使用說明。若一個節點中有多個通訊埠口被連接至同一網路，且若這些通訊口皆可成為主站時，則最後不論指定哪一個通訊口為主站，實際進行 PLC Link 操作的都將會是該節點的主機，因此執行的結果也都會相同的。

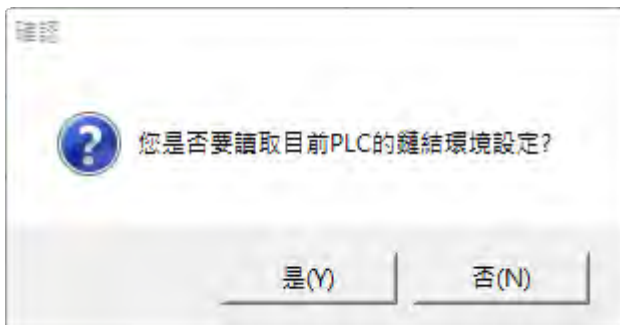


(1) 在「主站裝置」的下拉選單中指定主站，而此處的下拉選單則僅會列出可設為主站的通訊口。



(2) 選擇完成後，按下  圖示便可進入下一個步驟，而此時系統會先詢問使用者是否要直接上傳主站裝置內部的 PLC Link 設置，若選擇「否」，畫面便會直接進入**第二步驟**；若選擇「是」，系統便會自指定為主站的裝置中，將該節點主機內部的 PLC Link 設置上傳回來，而之後便會直接將上傳的資料顯示在第三步驟的畫面中；不過在執行上傳操作前，請先確認下列事項。

- (a) 請確認電腦與指定的主站裝置之間已正確用通訊電纜完成接續。
- (b) 已建立正確的 COMMGR Driver，且該 Driver 為「OK」狀態。
- (c) 已正確完成 NWCONFIG 中的通訊設定。



20.3.3 設定通訊參數 (步驟 2)

此步驟主要在設定進行 PLC Link 時的通訊參數，其中在畫面左側的區域則會顯示由主站上傳回來的通訊埠口參數，而在同一網路上的所有 PLC Link 從站，其通訊參數皆須與主站一致才可正常運作；不過若未執行上傳的動作時，此區域的各個欄位便會顯示「未知」，且當上傳回來的站號與網路配置的站號不一致時，該欄位便會呈現紅色以提醒使用者。



● 最大區塊大小

PLC Link 的資料交換是以區塊為單位，且一個 PLC Link 的規劃中，依據機種的不同，至多可設定 16 組或 32 組的資料交換區塊，而此處便是設定每個交換區塊的最大資料長度；根據指定為主站機種的不同，其可設定的最大資料長度也會有所差異，設定時請先查閱各機種的相關說明。



相關設定	說明	DVP	AH5x1
16 Words	單筆資料交換區塊最大長度 16 Words。只能使用 D 裝置。	V	V
100 Words	單筆資料交換區塊最大長度 100 Words。只能使用 D 裝置。	V	V
450 Words	單筆資料交換區塊最大長度 450 Words。且可以使用 M、D 或 L 裝置，主站與從站的裝置類別無須相同。		V

● RS485 通訊保持

DVP 系列的機種在斷電重開後，所有的通訊參數將可能會被恢復為預設值，勾選「RS485 通訊保持」的選項後，主機便會在由 STOP 切換至 RUN 時，重新將儲存的通訊參數載回；關於 DVP 系列機種的序列埠通訊參數，請參考各機種的操作手冊。

● 同步讀寫

一般的狀況下，主站對從站發出的寫入要求與讀取要求命令會分開傳送，但若勾選「同步讀寫」的項目時，利用特定的 MODBUS 功能碼 (16 進制碼為 17)，主站便可在一次的命令當中完成讀與寫的動作，如此便可提高資料交換的效率；不過設定時，請確認參與資料交換的所有裝置都有支援同步讀寫的 MODBUS 功能碼，否則便可能在接收到主站的命令後，會因無法識別該功能碼而造成讀寫失敗。除非選擇「450 Words」的模式，否則啟動同步讀寫時僅可選擇 D 裝置。



● 下載後啟動 PLC Link

若有勾選此項目，當完成 PLC Link 的規劃，並將其下載至主機之後，便會同時啟動 PLC Link 的動作。

● 命令間隔時間

此處用於設定主站每次發出命令的時間間隔。



完成此頁的設定後，按下  即可進入下一步驟；而按下  則可回至上一個步驟。

20.3.4 設置資料交換表 (步驟 3)

20.3.4.1 資料交換表簡介

下列即為 PLC Link 的資料交換表，當 PLC Link 開始執行後，主站裝置便會依據資料交換表的設定，輪流向各個從站進行資料讀寫的動作。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	2	讀	D0~D99	<=	D100~D199	100	啟動	CPU530-RS2
		寫	D0~D99	=>	D200~D299	100		
2	3	讀	D100~D199	<=	D100~D199	100	啟動	CPU530-RS2
		寫	D100~D199	=>	D200~D299	100		
3	4	讀	D200~D299	<=	D100~D199	100	關閉	CPU530-RS2
		寫	D200~D299	=>	D200~D299	100		
4	0	讀	D300	<=	16#1000	0	關閉	Unknown
		寫	D300	=>	16#1000	0		
5	0	讀	D400	<=	16#1000	0	關閉	Unknown
		寫	D400	=>	16#1000	0		

匯入 匯出 重置 檢查設定 上傳 下載 下載並監控  完成

表格欄位	說明
序號 (#)	資料交換區塊的序號，且依據機種的不同，至多可設定 16 組或 32 組的區塊。
從站站號	資料交換區塊屬於的站號，同一個站號可規劃為多組不同區塊的從站。若此處顯示為 0 則是代表未定義的從站，而非廣播站號。
讀/寫	「讀」：表示此列所顯示的是主站讀取從站的資料區塊。 「寫」：表示此列所顯示的是主站寫入從站的資料區塊。
主站裝置位址	主站於此組區塊中所使用的裝置位址範圍。
<=>	「<=」：表示此列所顯示的是主站讀取從站的資料區塊。 「=>」：表示此列所顯示的是主站寫入從站的資料區塊。
從站裝置位址	從站於此組區塊中所使用的裝置位址範圍。當此組區塊的從站為自行定義的「MODBUS Device」裝置時，此範圍會以 16 進制的位址編號來顯示。
長度	資料交換區塊的長度。
狀態	設定是否啟動此區塊的資料交換功能，且當此區塊有任何的錯誤設定時，狀態便會強制切為「關閉」，而當狀態為「關閉」時，該筆資料的表格底色便會呈現灰色。
裝置型態	顯示資料交換區塊的硬體裝置名稱。

下圖便是一個資料交換表的範例，其中啟動 PLC Link 功能的共有#1、#2、#4 與#5 共 4 組區塊，而其資料交換的規劃則請參考下表。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	2	讀	D3000~D3099	<=	D2500~D2599	100	啟動	SV
		寫	D3100~D3149	=>	D2600~D2649	50		
2	2	讀	D3000~D3009	<=	D3000~D3009	10	啟動	SV
		寫	D3200~D3209	=>	D3200~D3209	10		
3	0	讀	D0~D99	<=	16#0000~16#0063	100	關閉	Unknown
		寫	D0~D99	=>	16#0000~16#0063	100		
4	3	讀	D3400~D3424	<=	16#1000~16#1018	25	啟動	MODBUS Device
		寫	D3500~D3524	=>	16#1025~16#103D	25		
5	6	讀	D3600~D3699	<=	D3000~D3099	100	啟動	CPU530-RS2
		寫	D3700~D3799	=>	D3100~D3199	100		

匯入 匯出 重置 檢查設定 上傳 下載 下載並監控 完成

區塊序號	說明
#1	主站會讀取 2 號從站 (SV 機種) 的 D2500~D2599 共 100 Words 的資料，並將其存放於主站的 D3000~D3099，同時也會將主站本身的 D3100~D3149 共 50 Words 的資料寫入從站的 D2600~D2649 當中。
#2	主站會讀取 2 號從站 (SV 機種) 的 D3000~D3009 共 10 Words 的資料，並將其存放於主站的 D3000~D3009，同時也會將主站本身的 D3200~D3209 共 10 Words 的資料寫入從站的 D3200~D3209 當中。
#4	主站會自 3 號從站 (MODBUS Device) 的暫存器位址 16#1000~16#1018 當中讀取共 25 Words 的資料，並將其存放於主站的 D3400~D3424 當中，同時也會將主站本身的 D3500~D3524 共 25 Words 的資料寫入從站的 16#1025~16#103D 暫存器當中。
#5	主站會讀取 6 號從站 (CPU530-RS2 機種) 的 D3000~D3099 共 100 Words 的資料，並將其存放於主站的 D3600~D3699，同時也會將主站本身的 D3700~D3799 共 100 Words 的資料寫入從站的 D3100~D3199 當中。

20.3.4.2 設定區塊

在要設定資料交換區塊的行列上雙擊滑鼠左鍵即可開啟設定視窗。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	2	讀	D3000~D3099	<=>	D2500~D2599	100	啟動	SV
		寫	D3100~D3149	=>	D2600~D2649	50		
2	2	讀	D3000~D3009	<=>	D3000~D3009	10	啟動	SV
		寫	D3200~D3209	=>	D3200~D3209	10		
3	0	讀	D0~D99	<=>	16#0000~16#0063	100	關閉	Unknown
		寫	D0~D99	=>	16#0000~16#0063	100		
4	3	讀	D3400~D3424	<=>	16#1000~16#1018	25	啟動	MODBUS Device
		寫	D3500~D3524	=>	16#1025~16#103D	25		
5	6	讀	D3600~D3699	<=>	D3000~D3099	100	啟動	CPU530-RS2
		寫	D3700~D3799	=>	D3100~D3199	100		



參數設定

鍊結裝置

站號: 機種:

鍊結狀態: 關閉 啟動

讀取

主站參數設定: 起始位置 資料長度

從站參數設定: 起始位置

寫入

主站參數設定: 起始位置 資料長度

從站參數設定: 起始位置

● 鍊結裝置

「站號」欄位是指定此組資料交換區塊所屬的從站，不可選擇主站站號但可以選擇在別的資料交換區塊用過的從站站號，如果選擇 0 則右側的「機種」欄位會顯示「Unknown」表示沒有定義，否則會顯示從站站號對應的硬體裝置名稱。

鍊結裝置

站號: 機種:

鍊結狀態: 關閉 啟動

- 鏈結狀態

設定是否啟動此組資料交換區塊的資料交換功能。

鍊結裝置		鍊結狀態
站號	機種	<input type="radio"/> 關閉
4	CPU530-RS2	<input checked="" type="radio"/> 啟動

- 「讀取」

兩邊的裝置類型必須相同，資料長度為要交換的筆數，將「從站參數設定」的裝置起始位置算起的筆數，讀取到「主站參數設定」的裝置起始位置算起的筆數。

讀取			從站參數設定	
主站參數設定			起始位置	
D	3000	100 Words	D	2500

- 「寫入」

兩邊的裝置類型必須相同，資料長度為要交換的筆數，將「主站參數設定」的裝置起始位置算起的筆數，寫入到「從站參數設定」的裝置起始位置算起的筆數。

寫入			從站參數設定	
主站參數設定			起始位置	
D	3100	50 Words	D	2600

規劃區段時，同一區段的資料可被寫至多個不同的位置中，但卻不允許不同區段的資料存放於同一位置；以上圖為例，主站 D3100~D3149 的內容可被重複寫至不同的從站，但主站的 D3000~D3099 這個區段卻不能再接收其他從站的資料。

20.3.4.3 資料交換表的機種同步性

在資料交換表中的「裝置型態」欄位，其內容會依據該站號所對應的主機型號來顯示，而對應的依據則是目前 **NWCONFIG** 圖示工作區中的配置；同一個主機有多個站號的時候，其「裝置型態」顯示的都會是該主機型號。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	2	讀	D0~D99	<=	D100~D199	100	啟動	SV
		寫	D0~D99	=>	D200~D299	100		
2	3	讀	D100~D199	<=	16#1000~16#1063	100	啟動	MODBUS Device
		寫	D100~D199	=>	16#1000~16#1063	100		
3	4	讀	D200~D299	<=	D100~D199	100	啟動	CPU530-RS2
		寫	D200~D299	=>	D200~D299	100		
4	5	讀	D300~D399	<=	D100~D199	100	啟動	CPU530-RS2
		寫	D300~D399	=>	D200~D299	100		
5	6	讀	D400~D499	<=	D100~D199	100	啟動	CPU530-RS2
		寫	D400~D499	=>	D200~D299	100		



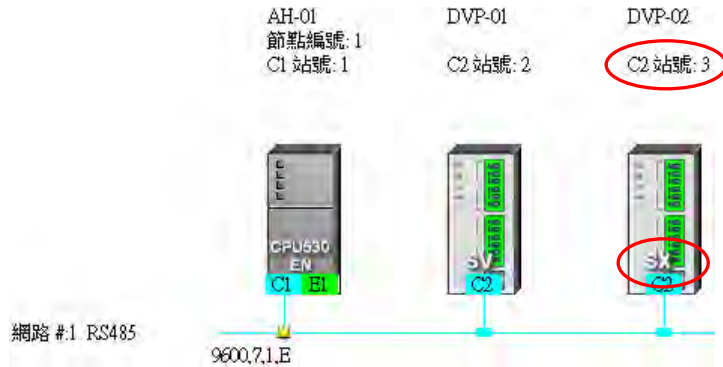
當已完成規劃並將所設定的資料儲存後，有變更機種的部分會以紅色的字體來提醒使用者，且狀態則會自動切換為「關閉」。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	2	讀	D0~D99	<=	D100~D199	100	關閉	DMV1000
		寫	D0~D99	=>	D200~D299	100		




當主站裝置為 AH5x0 系列的機種時，執行上傳後若配置與 NWCONFIG 目前的配置不符時，有變更機種的部分會以紅色的字體來提醒使用者，且狀態則會自動切換為「關閉」。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
2	3	讀	D400~D415	<=	16#0100~16#010F	16	關閉	5X
		寫	D400~D415	=>	16#0200~16#020F	16		



而當主站裝置為 DVP 系列的機種時，因 DVP 系列的主機內部不會儲存資料交換表中的裝置型態，因此上傳回來的資料將無法比對機種是否一致，所以「裝置型態」的欄位仍會依據圖示工作區中的機種名稱來顯示，但如果存在不合理的設定時，其狀態則會自動被切換為「關閉」。

20.3.4.4 資料交換表的管理操作

在資料交換表的下方有提供一些操作按鈕，而關於各個按鈕的功能則請參考下表的說明，其中若按下畫面中的  便可回至上一個步驟。



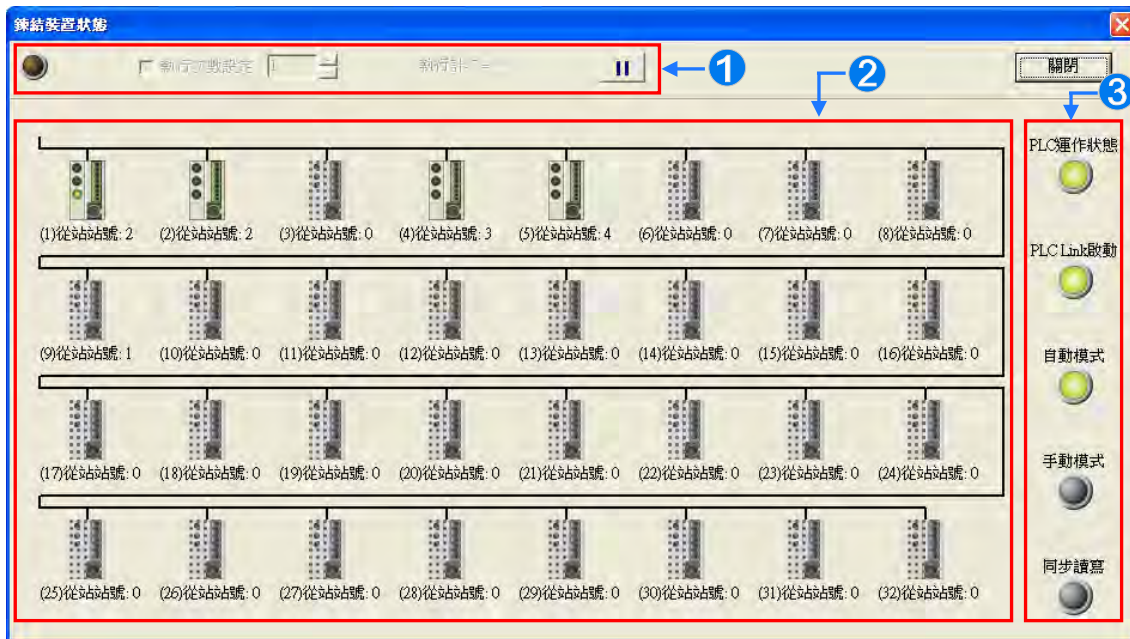
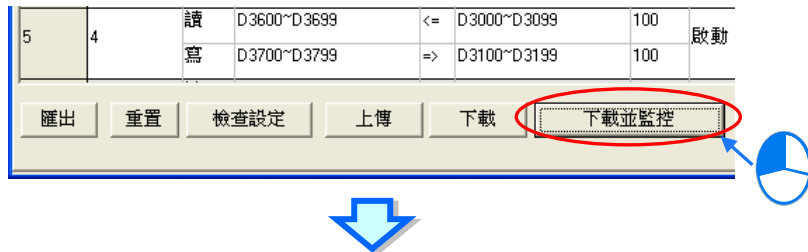
操作項	功能說明
匯出	將資料交換表匯出為 CSV 檔。
重置	將資料交換表的內容重置為初始狀態。

操作項	功能說明
檢查設定	檢查建立的資料交換表是否正確。
上傳	上傳主站的資料交換表。
下載	下載資料交換表至主站。
下載並監控	下載資料交換表至主站，並在下載完畢後自動開啟監控視窗。
完成	關閉視窗前系統會主動詢問使否要儲存所做的變更。

在 PLC Link 執行上下載或監控操作前，請務必確認實際連線的裝置是否正常。

20.3.5 PLC Link 的監控功能

進行監控前，請先確認系統與主機正常連線，接著按「下載並監控」便可開啟 PLC Link 的監控視窗；主站主機必須處於「RUN」的狀態下，PLC Link 的操作才可正常執行。

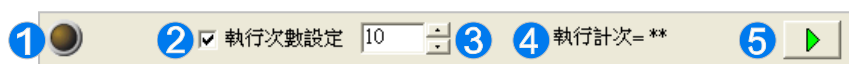


- ❶ 功能操作區：對 PLC Link 的資料交換功能進行操作。
- ❷ 圖示監控區：顯示主站對每個從站的讀寫狀態，並且可於此處進行線上的資料讀寫。
- ❸ 狀態燈號區：燈號是顯示目前的 PLC Link 狀態。各燈號的意義請見下表。

燈號	說明
PLC 運作狀態	亮起表示 PLC 主機處於 RUN。
PLC Link 啟動	亮起表示 PLC Link 啟動。
自動模式	亮起表示 PLC Link 的執行模式為「自動模式」。
手動模式	亮起表示 PLC Link 的執行模式為「手動模式」，而當執行次數到達「輪詢次數」之後，燈號便會熄滅。
同步讀寫	亮起表示是啟動同步讀寫的功能。

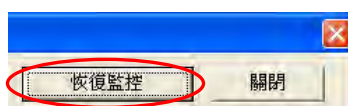
● 功能操作區

對 PLC Link 的資料交換功能進行設定或操作，此處所做的操作都會直接改寫主站主機內部所對應的特殊暫存器與特殊繼電器，所以離開 PLC Link 的監控功能之後，PLC Link 的運作仍會保持在最後操作的狀態，因此在離開監控畫面前請務必確認目前的 PLC Link 狀態是否正確。



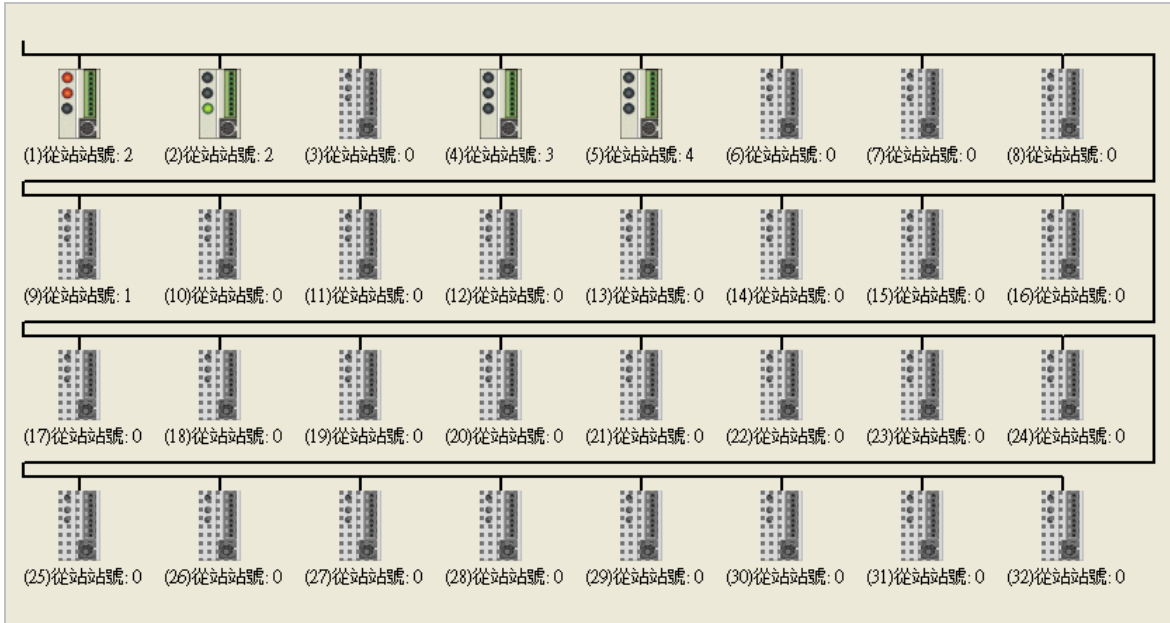
- ❶ 當燈號閃爍時就是指電腦與 PLC 主機保持連線。
- ❷ 勾選此項時，會依照❸所設定的「輪詢次數」執行，最大為 65535。
- ❸ 若有勾選❷，開始執行之後，會顯示目前已執行的次數；而計算的方式則是對所有從站完成一遍的資料讀寫。
- ❹ 啟動或停止監控的運作。

當在監控的過程中發生連線中斷時，視窗的右上角便會出現「恢復監控」的按鈕；在排除問題之後，按下此按鈕便可重新進行連線。

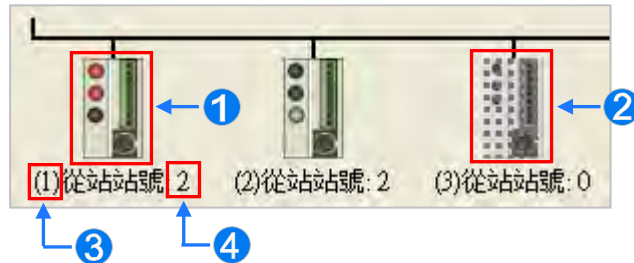


● 圖示監控區

PLC Link 在進行運作時，主站主機會依據規劃而輪流對各區塊所指定的從站進行讀寫，且透過**監控區**中的圖示，使用者便可得知主站主機對每個區塊進行讀寫時的狀態。



圖示**監控區**中的圖示代表的是主站的讀寫情況，而非從站裝置本身的狀態。圖示的表達說明如下。

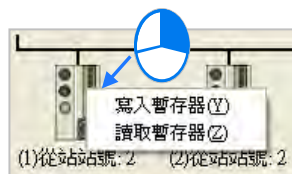


- ❶ 區塊讀寫狀態的圖示。透過圖示上的燈號便可得知此區塊的讀寫狀態。
- ❷ 圖示代表該區塊未定義 PLC Link 的功能，或功能狀態為關閉。
- ❸ 號碼對應的是資料交換表中的「**序號**」欄位，代表區塊的序號。
- ❹ 號碼對應的是資料交換表中的「**從站站號**」欄位，代表該區塊所指定的從站站號。

AH5x0 系列與 DVP 系列機種的 PLC Link 錯誤旗標有些許差異，因此在顯示也會有所不同，請參考下表的說明。

主站機種	區塊圖示	說明
AH5x0 系列		<p>❶ 亮起表示讀取從站資料發生錯誤時。</p> <p>❷ 亮起表示資料寫入從站發生錯誤時。</p> <p>❸ 亮起表示正在進行此區塊的資料交換。</p>
DVP 系列		<p>❶ 亮起表示從站進行資料讀寫發生錯誤。</p> <p>❷ 亮起表示正在進行此區塊的資料交換。</p>

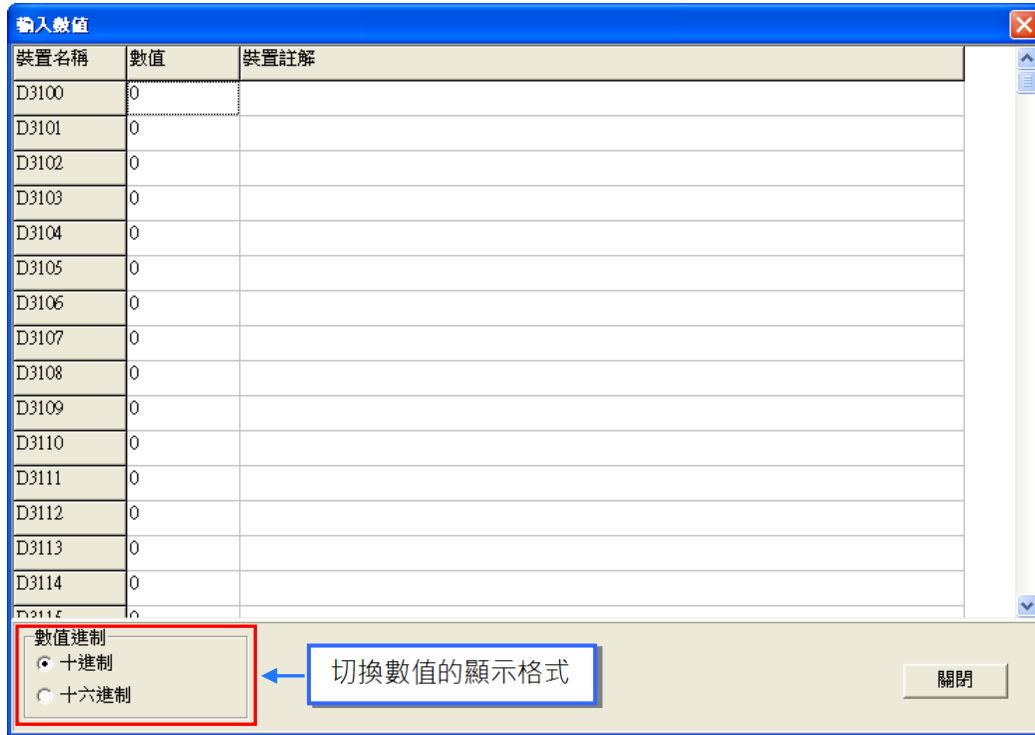
點選某一區塊的圖示後按下滑鼠右鍵，之後便可於快捷選單中選取要進行線上編輯的暫存器類別。



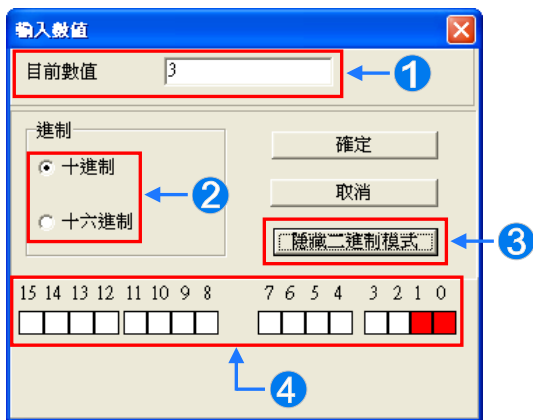
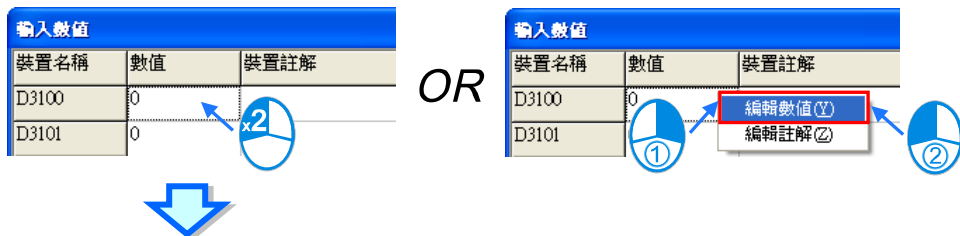
- **寫入暫存器**：指主站中要寫入從站資料的裝置位置，如下圖的 D3100~D3149。
- **讀取暫存器**：指主站中要讀回從站資料的裝置位置，如下圖的 D3000~D3099。

#	從站站號	讀/寫	主站裝置位址	<=>	從站裝置位址	長度	狀態	裝置型態
1	2	讀	D3000~D3099	<=	D2500~D2599	100	啟動	SV
		寫	D3100~D3149	=>	D2600~D2649	50		

當選取完暫存器類別後，畫面便會開啟對應該區段的暫存器監控表。



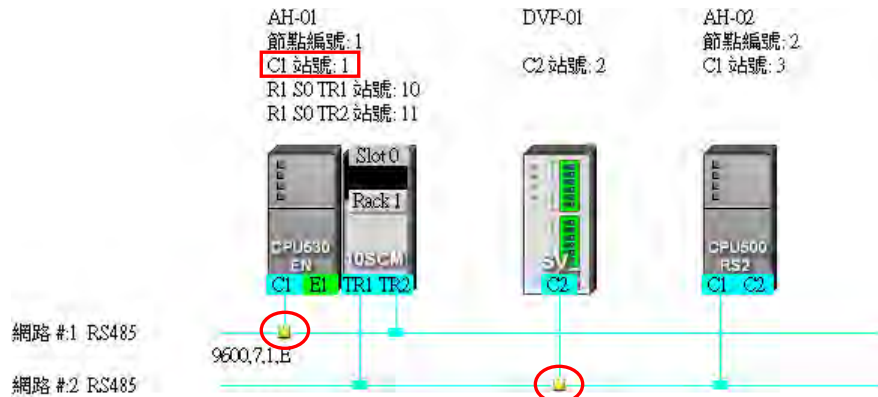
在某個裝置的「數值」欄位上雙擊滑鼠左鍵，或是按下右鍵後於快捷選單中點擊「編輯數值 (Y)」，之後便可於數值輸入視窗中設定該暫存器的內容值；而於「裝置註解」欄位上雙擊滑鼠左鍵，或是於右鍵的快捷選單中點選「編輯註解 (Z)」時便為該裝置加上註解，不過此處的註解為 PLC Link 專用的資訊，與 ISPSOft 編程中的裝置註解無關。



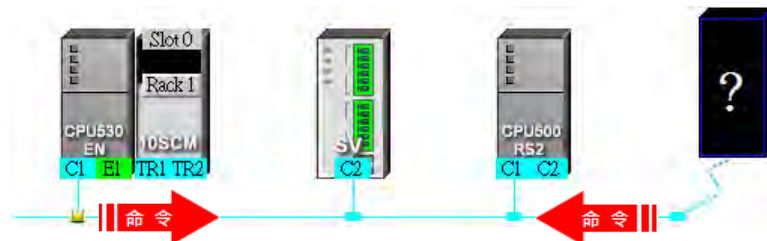
- ① 於此欄位輸入欲設定的數值。
- ② 此欄位可切換輸入數值時的格式。
- ③ 開啟或隱藏二進制的編輯功能。
- ④ 開啟二進制的編輯功能後，於此處便可以二進制的方式設定數值。

20.3.6 設置 PLC Link 的注意事項

規劃 PLC Link 時，同一個 RS485 的網路上只能有一個主站，而當設為主站的節點有多個序列埠時，該節點就不允許再成為其他 RS485 網路上的主站，只能當從站；如下圖的 **AH-01**，該節點的通訊端口 **C1** 為**網路#1** 上的主站，而透過模組的 **TR1** 端口就不能被指定為**網路#2** 的主站，只能當從站；**AH-01** 當中的 C1 與 TR2 都被連接至**網路#1**，且因兩個埠口的站號是彼此獨立的，因此在**網路#1** 的 PLC Link 規劃中，TR2 被指定為從站的狀況將是被允許的。



一個網路同時間只可以有一台主站發出命令，因此在網路規劃完畢之後，請務必確認實際的接線與 NWCONFIG 中的規劃一致或相容，避免產生類似下圖的錯誤。



當重新定義主站時，若原主站的 PLC Link 參數未重新設定，便可能形成同時有兩個主站發出讀寫命令而產生錯誤；因此當要變更主站時，請務必確認原主站的 PLC Link 動作已停止。



* 除了上述狀況之外，任何同時間有兩個或以上裝置發出讀寫命令的狀況都應該要避免。

20.4 Ether Link 的建構

20.4.1 Ether Link 簡介

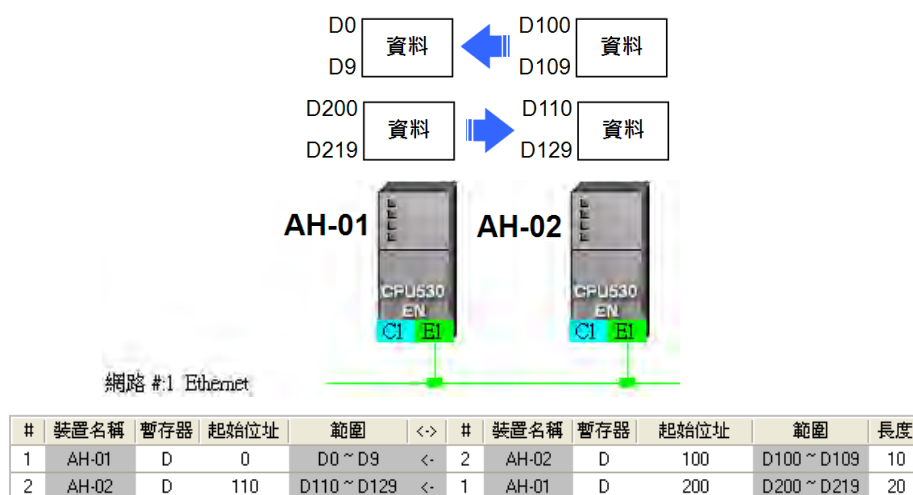
Ether Link 為透過乙太網路 (Ethernet) 的連線來進行資料交換的網路機制。當某個 Ethernet 網路上存在多個節點時，我們便可於該網路中規劃一個資料交換的機制；之後當設置參數被下載至各個節點的主機之後，根據設定的運作方式，系統便可自動進行資料交換的動作。

於 PLC Link 的架構中，資料交換的操作是著重在主站的讀取與寫入，而相較於 PLC Link 的主從架構，Ether Link 則是採用較安全的**資料索取**機制，其操作則是以網路上的**資料需求節點**為主。

在 Ether Link 的架構下，網路上的每個節點都可以成為**資料需求節點**，並主動的發出需求向其他節點索取資料，但卻必須被動的在接收到其他節點的需求之後，才可以將資料回傳給提出需求的節點，因此在這樣的架構下，並不會有節點主動或強制的將資料寫入其他裝置的情況。基於 TCP/IP 的通訊協定，所有的**資料需求節點**都可隨時提出索取資料的需求，系統本身則會自動安排資料封包的傳送機制，因此在進行資料交換時，這樣的機制便可提高執行的效率。

請參考下圖的範例，當要建構如下的資料交換關係時，規劃上便必須針對兩個**資料需求節點**，分別為其建立各自的需求項目於**資料交換表**中。

- (a) **AH-01** 向 **AH-02** 索取 D100~D109 的資料，並將取得的資料存放於 **AH-01** 的 D0~D9。
- (b) **AH-02** 向 **AH-01** 索取 D200~D219 的資料，並將取得資料存放於 **AH-02** 的 D110~D129。




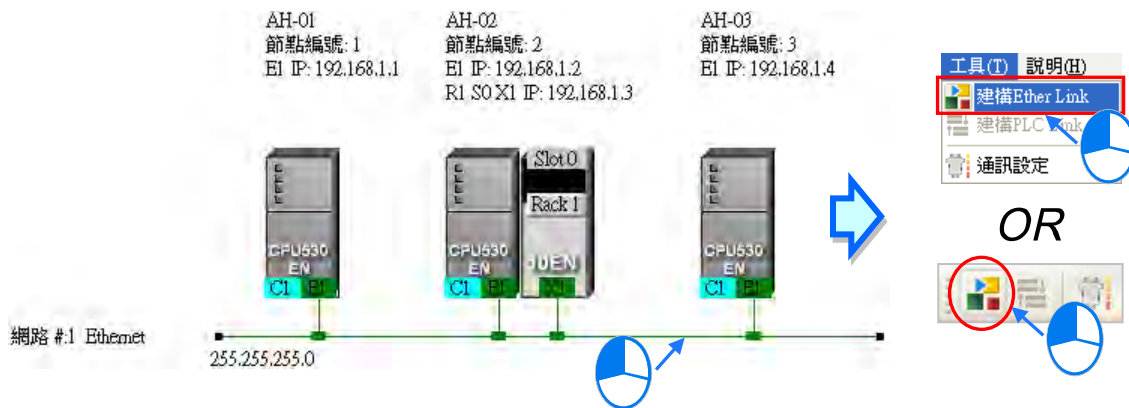
Ether Link 的設置是以網路為單位，而當某個節點被連接至不同編號的網路時，設置上便可為每個網路各自設定一組 Ether Link 的參數，因此同一個主機內部可能會儲存多組的 Ether Link 的參數；且因 Ether Link 並非如同 PLC Link 一般的主從架構，因此每個節點在不同編號的網路上都可以同時成為**資料需求節點**與**資料提供節點**；例如以上圖的第一筆資料來說，**AH-01** 是**資料需求節點**，而 **AH-02** 則是**資料提供節點**；而以第二筆資料來說，**AH-01** 是**資料提供節點**，而 **AH-02** 則是**資料需求節點**。

20.4.2 開啟 Ether Link 的設置視窗

建構 Ether Link 之前請先確認所有的網路設置正確無誤後，依照下述方法進行操作。

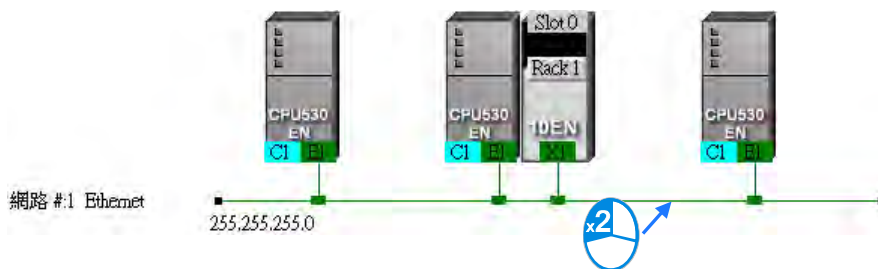
● 方法一

點選要建構 Ether Link 的網路後，於功能工具列中點選 **工具 (I)** > **建構 Ether Link**，或直接點擊圖示工具列中的  圖示。



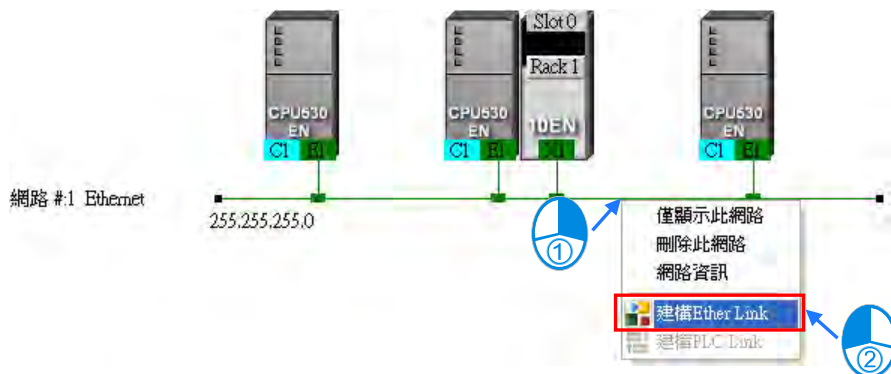
● 方法二

於要建構 Ether Link 的網路上雙擊滑鼠左鍵。

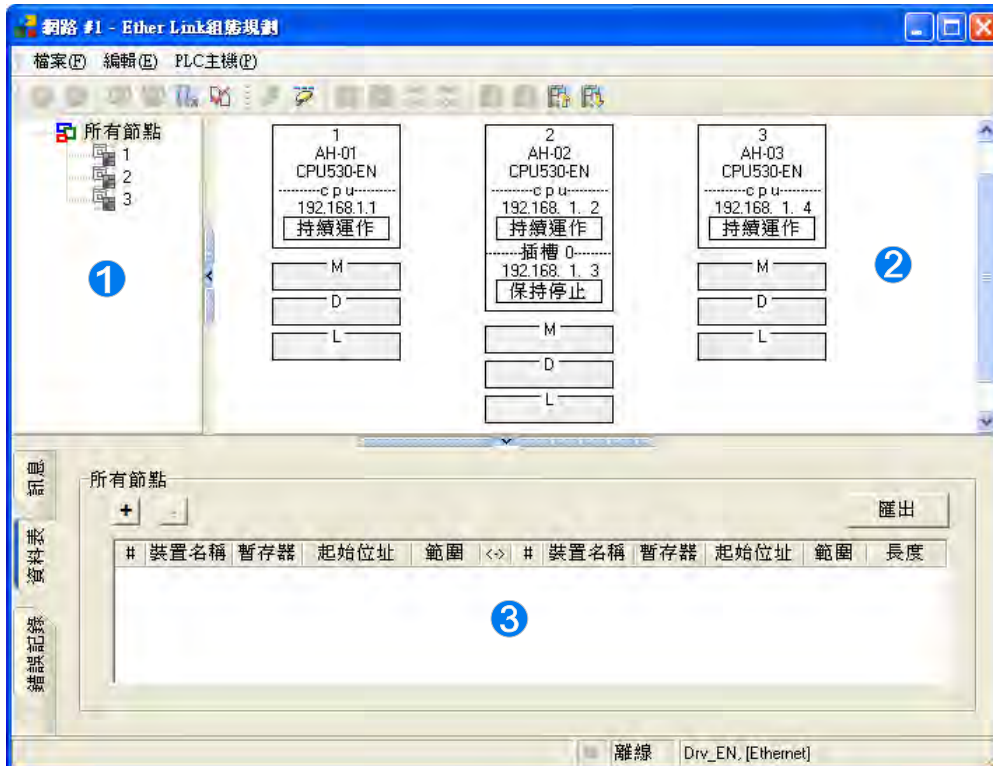


● 方法三

於要建構 Ether Link 的網路上點擊滑鼠右鍵，之後於快捷選單中點選「建構 Ether Link」。

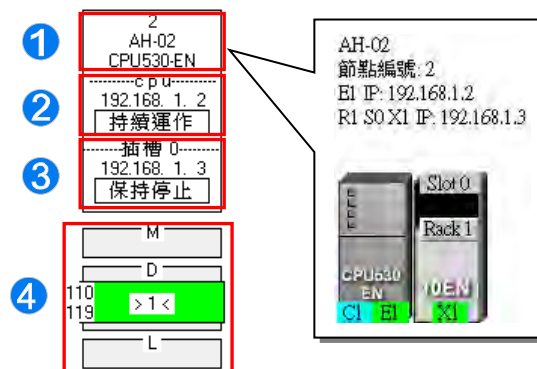


下圖便是 Ether Link 的設置畫面。



- ❶ 裝置清單：顯示所有的節點清單。
- ❷ 圖示區：以圖示表達節點的相關設定與資訊。
- ❸ 資訊視窗：顯示資料交換表、操作訊息以及裝置的錯誤記錄...等 3 個頁面。

下列則為圖示區中的節點圖示與其標示的相關說明



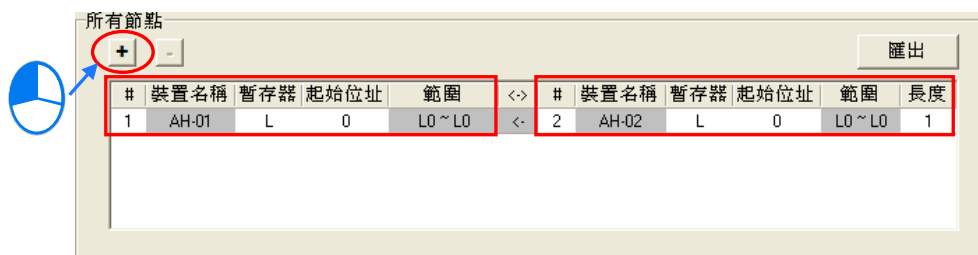
- ❶ 由上至下依序為節點編號、PLC 名稱及機種型號。
- ❷ 分隔線的 **cpu** 字樣，代表此處顯示的是主機的 IP 位址與其 Ether Link 的啟動模式。
- ❸ 分隔線的 **插槽 0** 字樣，代表此處顯示的是插槽 0 的模組 IP 位址與其 Ether Link 的啟動模式。
- ❹ 此處顯示的為資料交換區塊的設定。上圖 D 區塊的 **>1<** 代表向節點 1 索取資料，而左側的數字則代表取自節點 1 的資料將會存放於節點 2 的 D110~D119，至於區塊的顏色則是依據節點編號隨機配置；另外，M 與 L 區塊無任何顯示，代表該區塊並無向其他裝置索取資料。

20.4.3 資料交換表的建立與管理

點選「資料表」的標籤即可切換至資料交換表的操作頁面；其中在表格的左上方會顯示目前裝置清單中所選取的節點，而表格也只會顯示以該節點為資料需求節點時的資料。

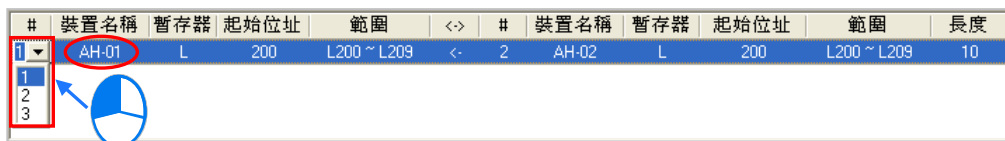


於表格上方點擊 **+** 鍵即可新增一筆資料，且資料的內容可分為兩個部份，左側代表資料需求節點在取得資料後的存放位置；而右側則代表欲索取資料的對象（資料提供節點）及索取的資料範圍。



設定資料交換區塊的步驟如下。

- (1) 於資料需求節點的「#」欄位下拉選擇節點編號，選擇完畢後，「裝置名稱」欄位便會顯示對應該節點編號的 PLC 名稱。



不過當目前裝置清單中所選取的不是「所有節點」，而是特定的節點編號時，因表格僅會列出以該節點為資料需求節點時的資料，所以此欄位將會固定為裝置清單中所選取的節點編號。



- (2) 於資料需求節點的「暫存器」欄位下拉選擇裝置類別，並於「起始位址」欄位輸入位址編號，以決定取得資料後的存放位置。

#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	L	500	L500 ~ L509	<-	2	AH-02	L	200	L200 ~ L209	10

- (3) 於資料提供節點的「#」欄位下拉選擇節點編號，且選擇時，需求節點與提供節點的編號不可相同，選擇完畢後，「裝置名稱」欄位也會顯示對應該節點編號的 PLC 名稱。

#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	L	200	L200 ~ L209	10

- (4) 再來則要設定欲索取的資料區塊。首先請於資料提供節點的「暫存器」欄位下拉選擇裝置類別，而所選取的類別則無須與資料需求節點一致，接著再於「起始位址」欄位輸入位址編號即可。


#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10

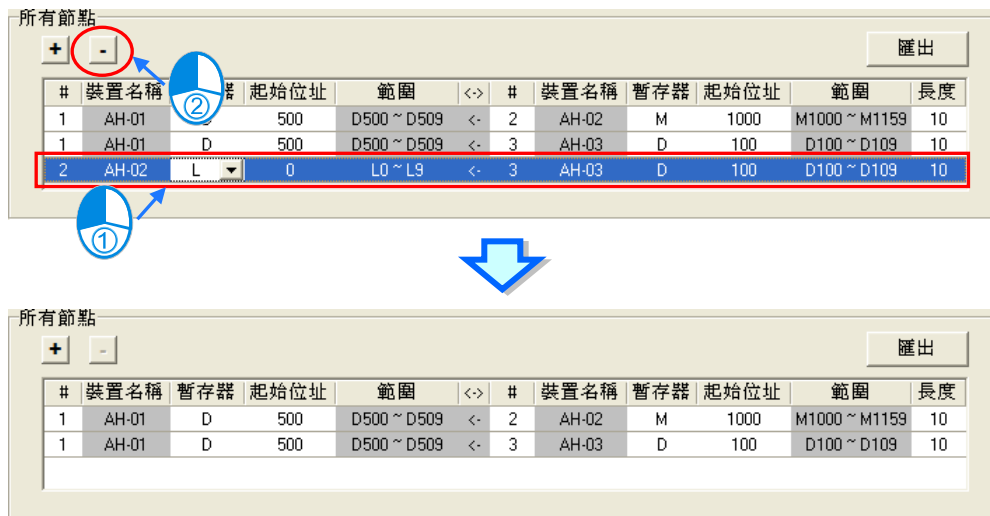
- (5) 最後再於「長度」欄位設定欲索取的資料長度 (單位為 Word)，且最大不得超過 1900；而設定完成後，於兩側的「範圍」欄位便會自動換算資料交換區塊的範圍。

#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10

接下來便可依照上述的說明逐一建立每筆資料，但須注意所設定的資料區段中，資料需求節點的存放區段不得重疊，而資料提供節點則無特別限制；不同的需求方可要求同一區塊的資料，但不可將不同的資料存放在同一區塊。


#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
1	AH-01	D	500	D500 ~ D509	<-	3	AH-03	D	100	D100 ~ D109	10
2	AH-02	L	0	L0 ~ L9	<-	3	AH-03	D	100	D100 ~ D109	10

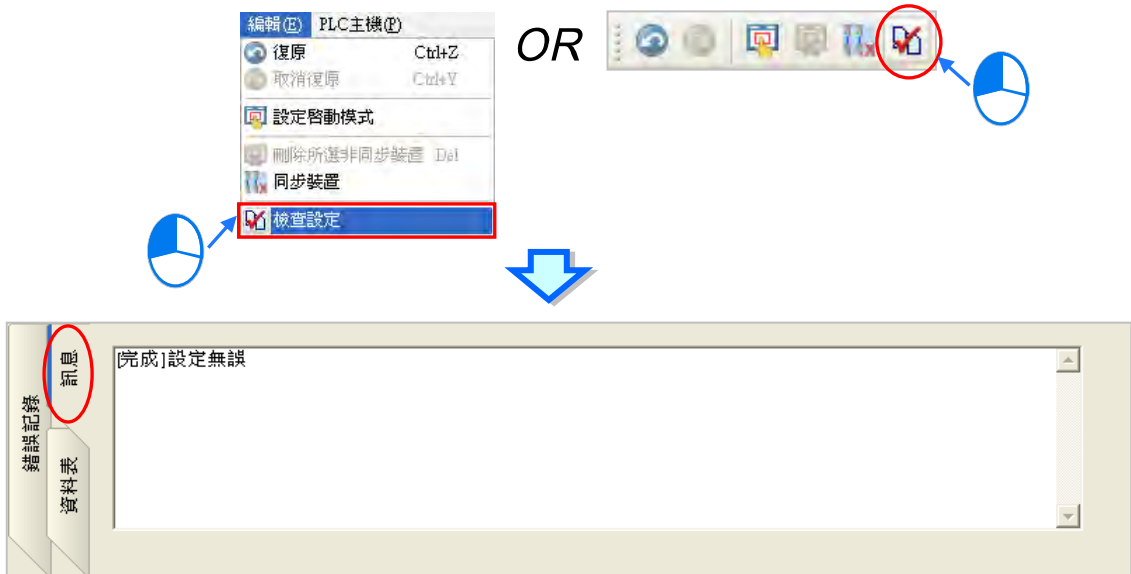
當在過程中在要刪除某筆資料時，請先選取該筆資料後按下  鍵即可。



按下表格右上角的「匯出」鍵則可將資料交換表的內容匯出為 **CSV** 檔。



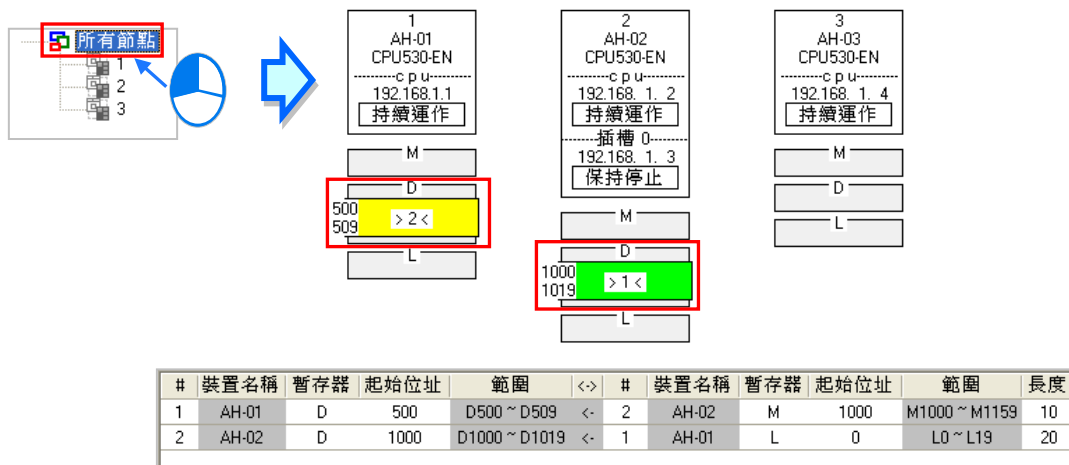
規劃完資料交換表後，於功能工具列中點選 **編輯 (E) > 檢查設定**，或按下圖示工具列中的  圖示即可對所設定的資料進行檢查，而檢查的結果將顯示於下方資訊視窗中的「訊息」頁面。



20.4.4 裝置清單與圖示區的操作

於裝置清單點選「所有節點」時，圖示區便會列出所有節點的圖示，下方的資料交換表也會列出所有資料交換區塊的設置，在各個節點圖示的位址區塊中，也會顯示各節點用來存放索取資料的位址範圍以及索取資料的對象（資料提供節點）。

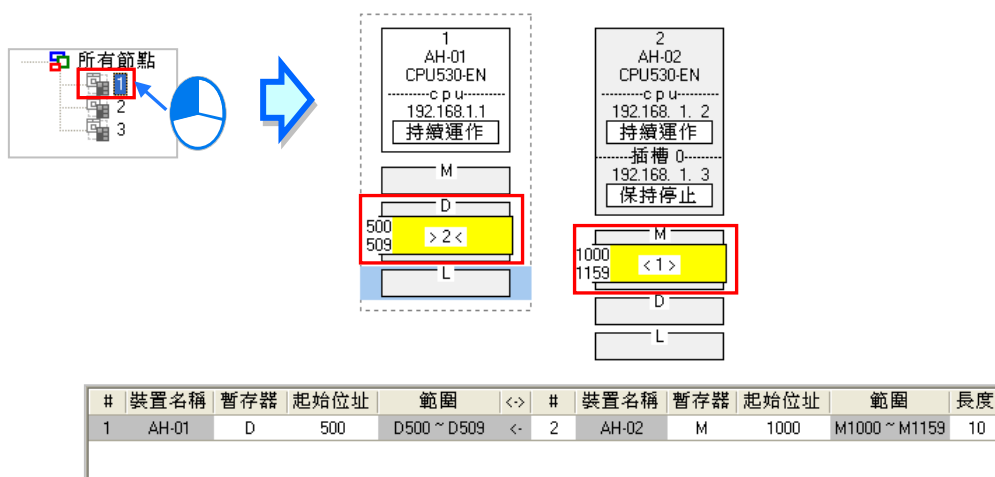
如下圖節點 1 的圖示，可以得知節點 1 的 D500~D509 將會用來存放取得的資料，且由區塊中的 >2< 可得知該資料來是來自節點 2；同樣的，我們也可由節點 2 的圖示得知其 D1000~D1019 的區塊將會用來存放取自節點 1 的資料。



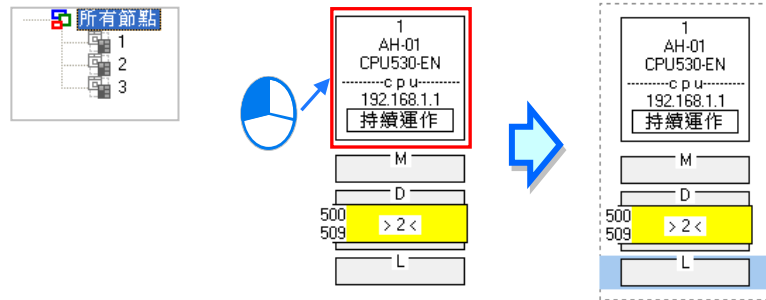
而當於裝置清單點選某個特定編號的節點時，圖示區便只會列出該節點與被該節點索取資料的其他節點圖示，而下方的資料交換表亦僅會列出以該節點為資料需求節點時的資料。

如下圖，其中節點 1 圖示的外圍有一個虛線的方框，代表目前節點 1 處於選取狀態，而節點 2 的圖示則以灰色底色來呈現，代表節點 2 目前為資料提供節點；在這個狀態中，節點 2 的圖示將無法選取，且節點 2 圖示下方的資料區塊也只會顯示提供資料給節點 1 的部份。除此之外，因節點 3 並未提供資料給節點 1，所以在這個狀態下，節點 3 的圖示便不會顯示。

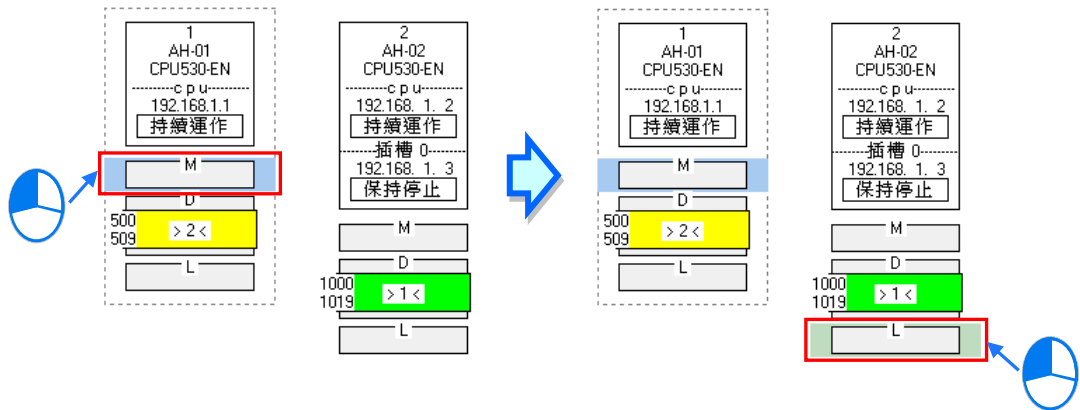
由節點 1 的圖示中可得知，節點 1 的 D500~D509 會存放取自節點 2 的資料；而由節點 2 的圖示與圖示區塊中的 <1> 則可得知，節點 2 會將本身 M1000~M1159 的資料提供給節點 1。



當在**裝置清單**中選取某個特定的節點編號時，圖示區中的該節點便會自動處於選取狀態；但當**裝置清單**中選取的是「**所有節點**」，且要在圖示區中選取某個節點時，只要在該節點的節點資訊處點擊滑鼠左鍵，之後便可將該節點選取起來；選取時須注意滑鼠必須點擊在節點資訊的位置，若點擊在下方的資料區塊則無法將該節點選取起來，而被選取起來的節點圖示外圍便會有一個虛線的方框。



當某個節點被選取起來之後，表示要將該節點指定為**資料需求節點**，此時可先於該節點的下方點選裝置類別，點選後該區塊便會出現淺藍色游標；之後再點選其它節點（**資料提供節點**）的裝置類別，而點選後該區塊則會出現淺綠色游標；須注意點選**資料提供節點**的裝置類別時，只須直接點選該節點下方的區塊，而不可將該節點整個選取起來，否則便等於重新指定**資料需求節點**。



完成之後，按下資料交換表 **+** 便可新增一筆資料，並會直接帶入**資料需求節點**與**資料提供節點**的節點編號與暫存器類別。

+ **-** 匯出

#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	<-	1	AH-01	L	0	L0 ~ L19	20

#	裝置名稱	暫存器	起始位址	範圍	<>	#	裝置名稱	暫存器	起始位址	範圍	長度
1	AH-01	D	500	D500 ~ D509	<-	2	AH-02	M	1000	M1000 ~ M1159	10
2	AH-02	D	1000	D1000 ~ D1019	<-	1	AH-01	L	0	L0 ~ L19	20
1	AH-01	M	0	M0 ~ M15	<-	2	AH-02	L	0	L0 ~ L0	1

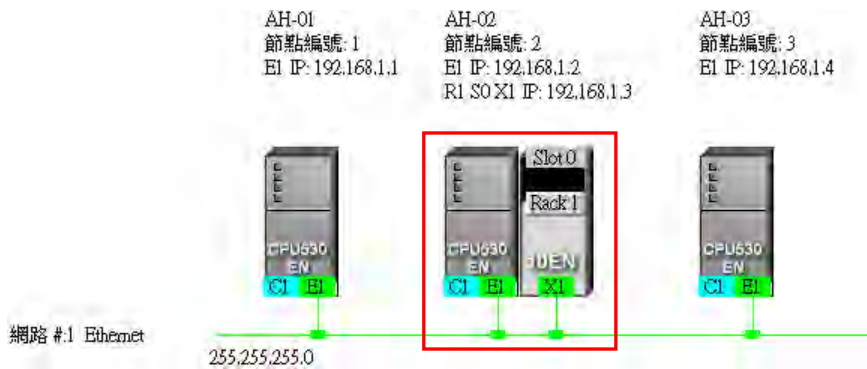
20.4.5 Ether Link 的運作模式

Ether Link 的**啟動模式**共有三種，相關說明請參考下表。

運作模式	說明
保持停止	主機在運作的期間都不會進行資料交換的動作。
持續運作	一旦主機開始運行後，資料交換的動作便會持續進行。
旗標模式	資料交換的動作將依據主機內部的特殊繼電器狀態來運行，而選擇旗標模式後，針對啟動資料交換的特殊繼電器，使用者還可設定其初始狀態。

*關於 Ether Link 的相關特殊繼電器，請參考各機種的使用手冊或相關技術文件。

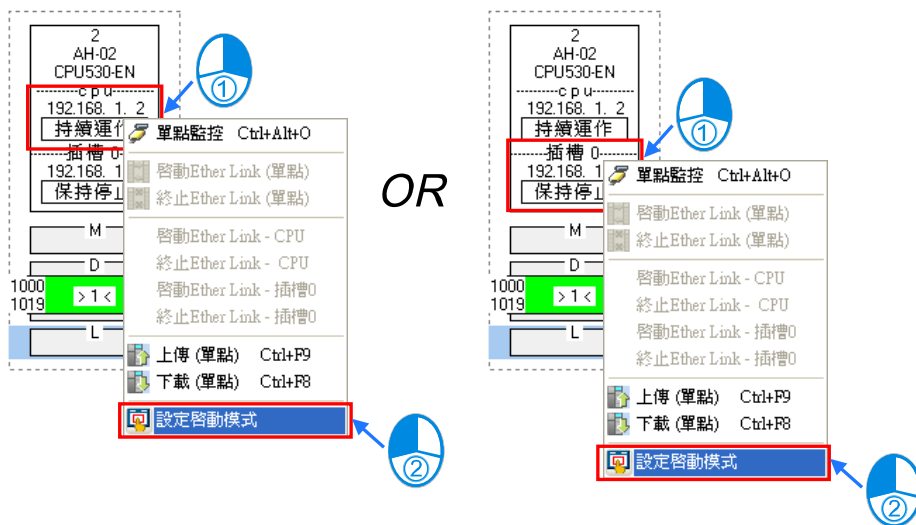
每組網路的 Ether Link 運作是以節點為單位，而當該節點有多個通訊埠被連接到同一個編號的網路上時，每個通訊埠的**啟動模式**皆可分開設定；實際運作時，系統便會根據設定自動分配每個通訊埠的傳送與接收，而當有某個通訊埠斷線時，系統也會自動將工作轉移至該節點的其他通訊埠。



要設定某個節點的**啟動模式**時，請參考下列的幾種方式。

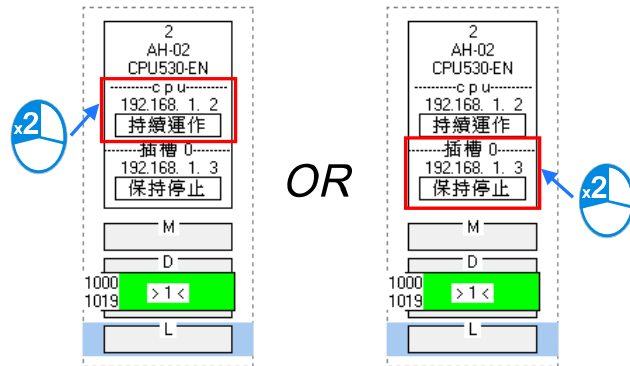
● 方法一

於 CPU 或模組資訊區塊上點擊滑鼠右鍵，之後於快捷選單中選取「設定啟動模式」。




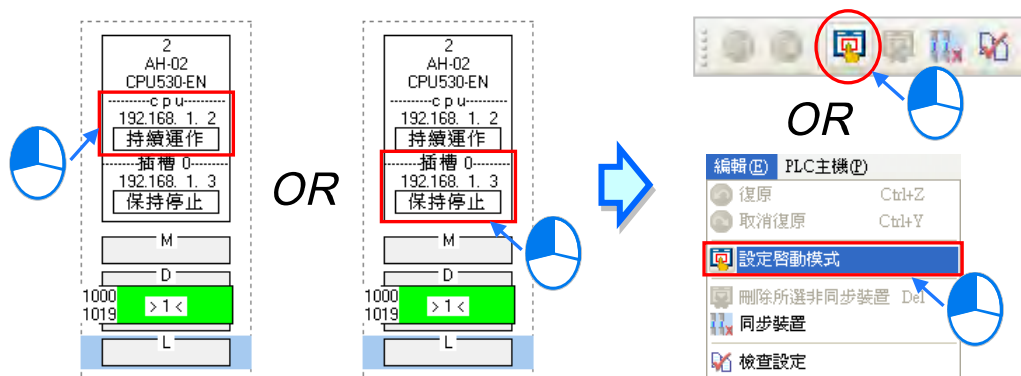
● 方法二

在要設定**啟動模式**的 CPU 或模組資訊區塊上雙擊滑鼠左鍵。



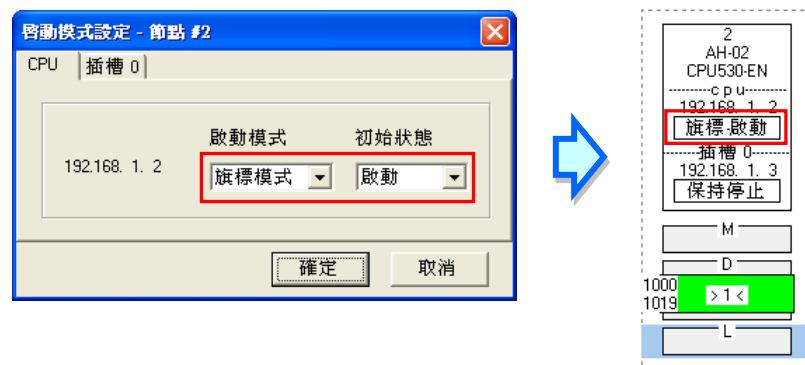
● 方法三

在要設定**啟動模式**的 CPU 或模組資訊區塊上點擊一下滑鼠左鍵後，於圖示工具列中點選  圖示，亦或於功能工具列中點選 **編輯 (E) > 設定啟動模式**。



依據上述的方式操作之後，畫面便會出現啟動模式的設定視窗，而依據在節點圖示上所點選的位置，開啟的視窗亦會自動切換至對應的頁面，不過使用者仍可透過上方的標籤自行切換。

接著請直接在「**啟動模式**」的欄位下拉選擇欲設定的模式，而當選擇的是**旗標模式**時，請接著於「**初始狀態**」的欄位下拉選擇對應特殊繼電器的初始狀態。完成後按下「**確定**」鍵即可。




20.4.6 下載 Ether Link 設置

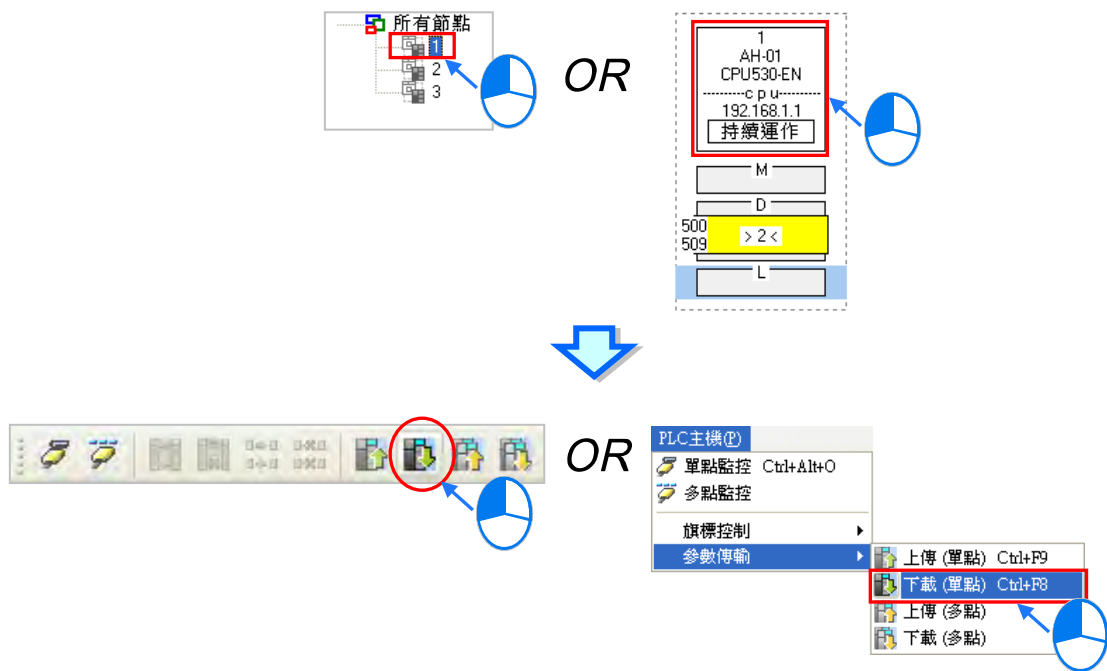
當完成所有的設置後，必須將 Ether Link 的參數下載至 PLC 主機當中，主機才可依據設置的參數進行資料交換的動作。

● 單點下載

僅下載所選主機的資料需求設置，並設定該主機與模組的啟動模式；下載之前，請確認系統與主機或模組連線都正常。

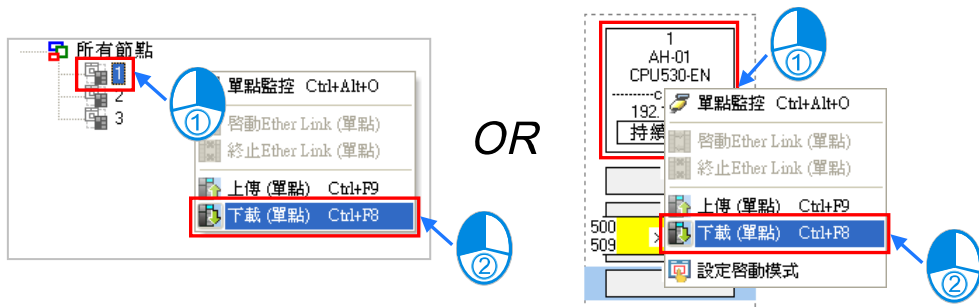
➢ 方法一

選取要下載參數的資料需求節點，之後於圖示工具列中點選  圖示，或是於功能工具列中依序點選 **PLC 主機 (P)** > **參數傳輸** > **下載 (單點)**。



➢ 方法二

於裝置清單或圖示區中，點選欲下載參數的資料需求節點後按下滑鼠右鍵，之後於快捷選單中點選「下載 (單點)」項目。

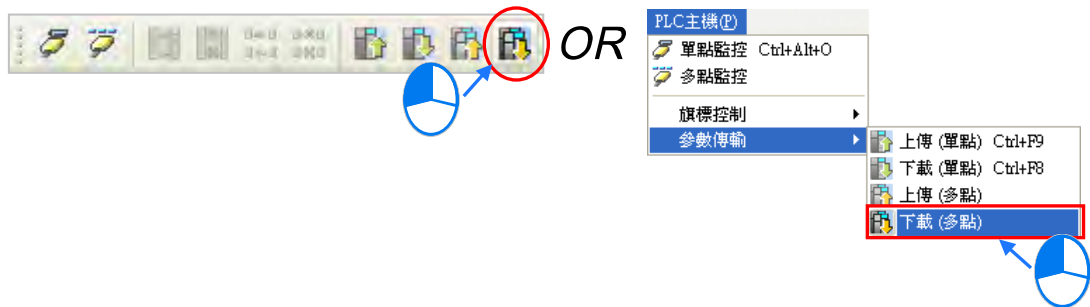


● 多點下載

下載所有節點的資料需求設置與啟動模式；下載之前，請確認系統與主機或模組連線都正常。

➢ 方法一

於功能工具列中點選 **PLC 主機 (P)** > **參數傳輸** > **下載 (多點)**，或於直接點擊  圖示。



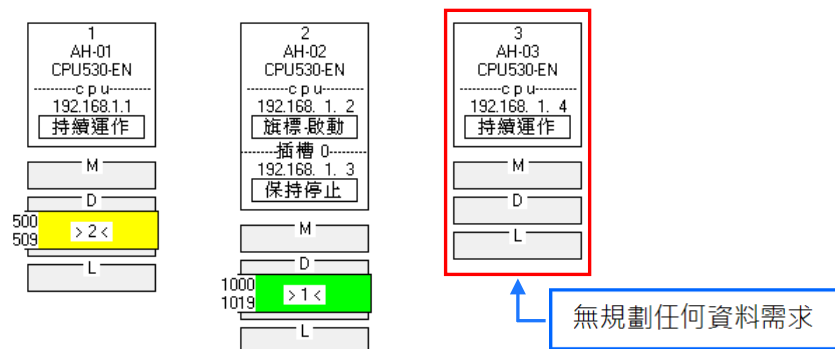
➢ 方法二

於裝置清單中點選「**所有節點**」後按下滑鼠右鍵，之後於快捷選單中點選「**下載 (多點)**」項目。



補充說明

當設置的參數存在一個未規劃任何資料需求的節點時，在進行多點下載後，該節點對應此組網路編號的 Ether Link 參數會被設定為無資料需求。




20.4.7 上傳 Ether Link 設置

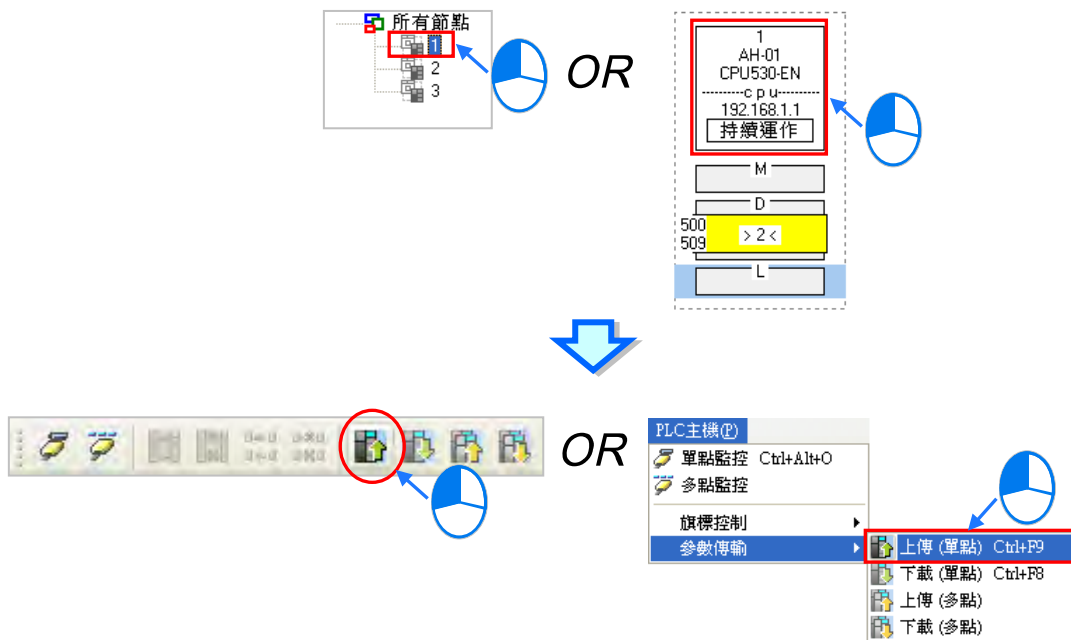
利用上傳的功能便可重新取得該主機內部的 Ether Link 設置參數。

● 單點上傳

只上傳單個節點的 Ether Link 設置參數；上載之前，請確認系統與主機或模組連線都正常。

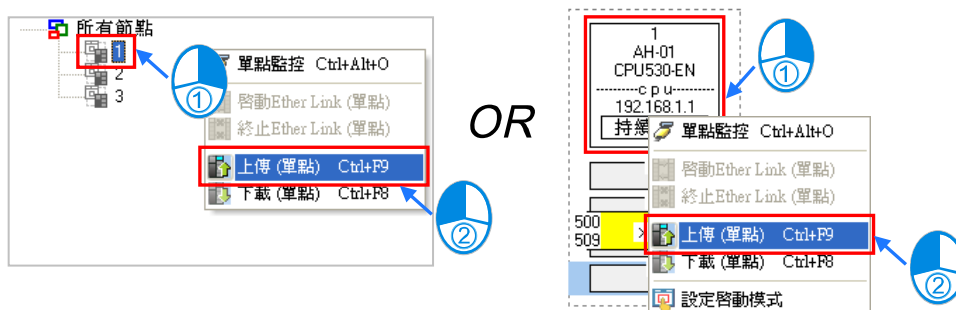
➤ 方法一

先選取要上傳參數的資料需求節點，之後於圖示工具列中點選  圖示，或是於功能工具列中依序點選 **PLC 主機 (P)** > **參數傳輸** > **上傳 (單點)**。



➤ 方法二


於裝置清單或圖示區中，點選要上傳參數的資料需求節點後按下滑鼠右鍵，之後於快捷選單中點選「上傳 (單點)」項目。

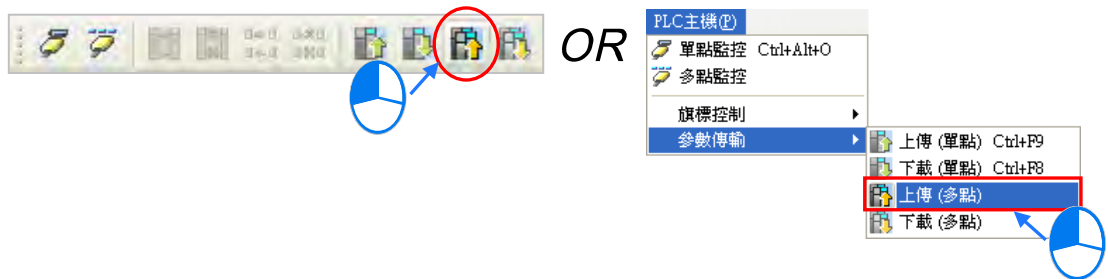


● 多點上傳

上傳所有節點的 Ether Link 設置參數；上載之前，請確認系統與主機或模組連線都正常。

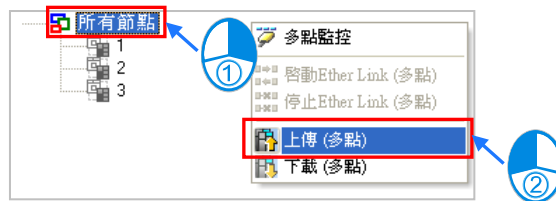
➢ 方法一

於功能工具列中點選 **PLC 主機 (P)** > **參數傳輸** > **上傳 (多點)**，或於直接點選  圖示。



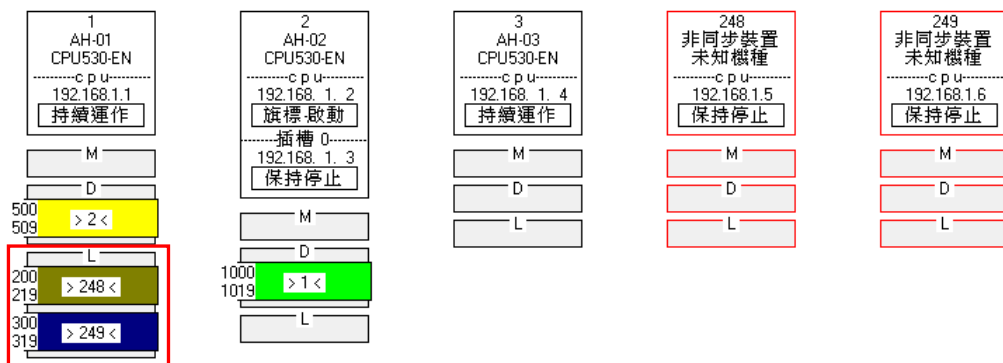
➢ 方法二

於裝置清單中點選「所有節點」後按下滑鼠右鍵，之後於快捷選單中點選「上傳 (多點)」項目。




補充說明

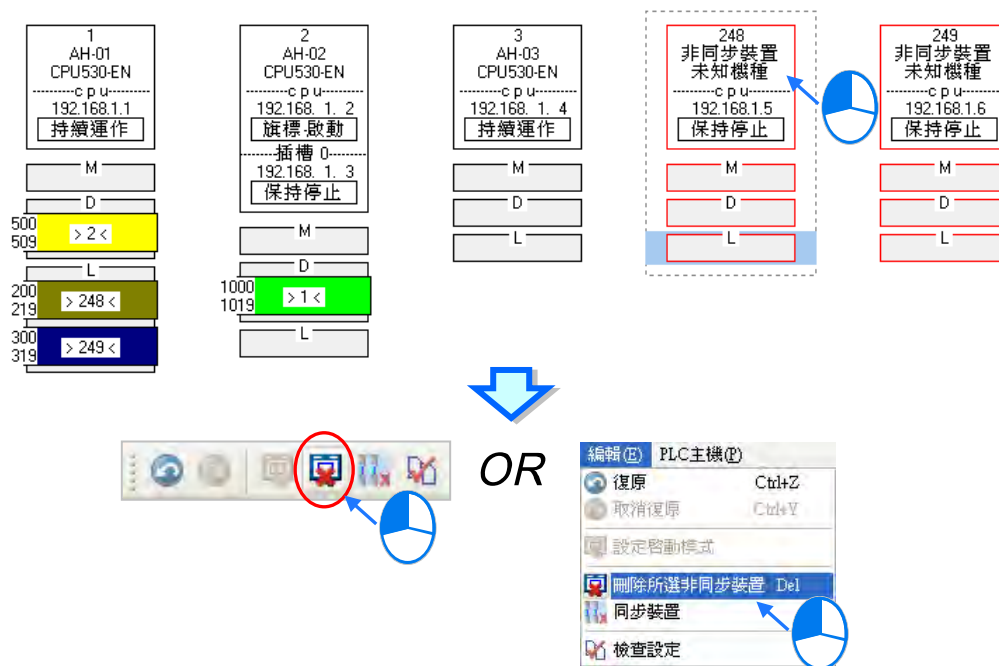
重新上傳 Ether Link 的參數後，如果有未配置於 NWCONFIG 網路中的節點時，該節點會以紅色框線顯示「非同步裝置」，而當 Ether Link 的設置存在此種非同步裝置時，就不能進行監控或下載的動作。




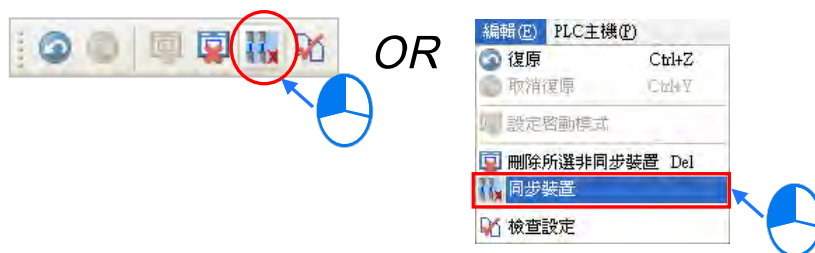
20.4.8 刪除非同步裝置

當存在非同步裝置時，將不能進行監控或下載的動作，此時先確認產生非同步裝置的原因，若是因為 NWCONFIG 中的網路配置有誤，請先修正網路配置後再重新上傳 Ether Link 的參數；但若是上傳的參數已不適用目前的網路配置時，請將非同步裝置刪除後再進行後續的操作。

於圖示區中點選欲刪除的非同步裝置後，於功能工具列點選 **編輯 (E)** > **刪除所選非同步裝置**，也可以按圖示工具列中的  圖示刪除所選的非同步裝置。



要同時刪除所有的非同步裝置時，請直接於功能工具列中點選 **編輯 (E)** > **同步裝置**，或按下圖示工具列中的  圖示即可。



20.4.9 啟動/關閉線上監控模式

在 Ether Link 的設置視窗中，可透過線上監控來操作或測試 Ether Link 的功能，而線上監控的啟動/關閉方式又可分為單點與多點。

啟動/關閉方式	說明	
單點	功能	每次僅啟動或關閉所選節點的線上監控功能。
	條件	ISPSOft 與所選的主機正常連線，且已完成 NWCONFIG 的通訊設定。
多點	功能	同時啟動或關閉所有節點的線上監控功能。
	條件	所有節點都已接上網路並且 ISPSOft 透過 Ethernet 進行連線，而在 NWCONFIG 的通訊設定中，也須選擇 Ethernet 型式的 Driver。



在進行監控前，請再次確認所有的節點，都已依照 NWCONFIG 中的規劃，完成架設並已正常運作。

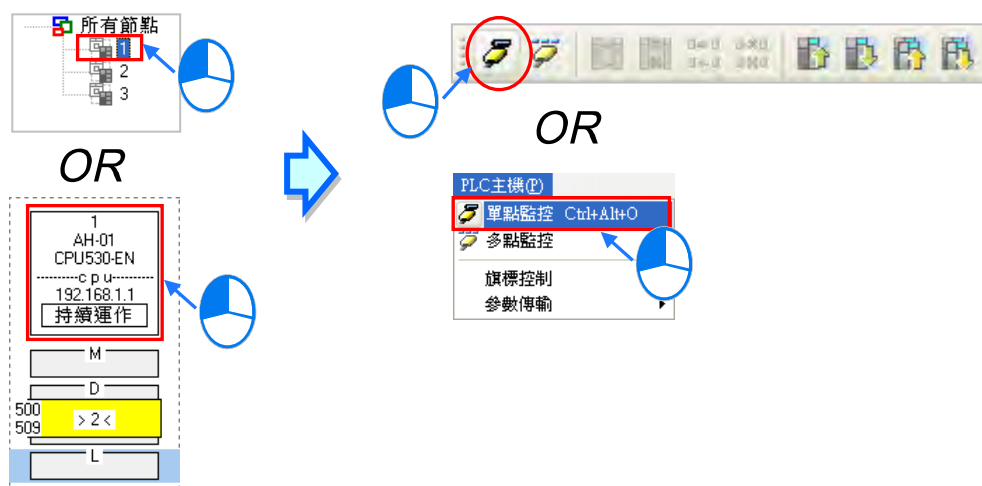
- (a) 每個節點，都已依照 NWCONFIG 中的規劃接上網路。
- (b) 每個節點，都已透過 HWCONFIG 設定各個通訊埠口的乙太網路參數，並下載至各個主機與模組當中，而設定的參數則必須與 NWCONFIG 中的規劃一致。
- (c) 已正確的將 Ether Link 的設置參數，下載至主機當中。
- (d) 各個節點都已上電並處於正常運作的狀態。

20.4.9.1 啟動監控


● 啟動單一節點的監控

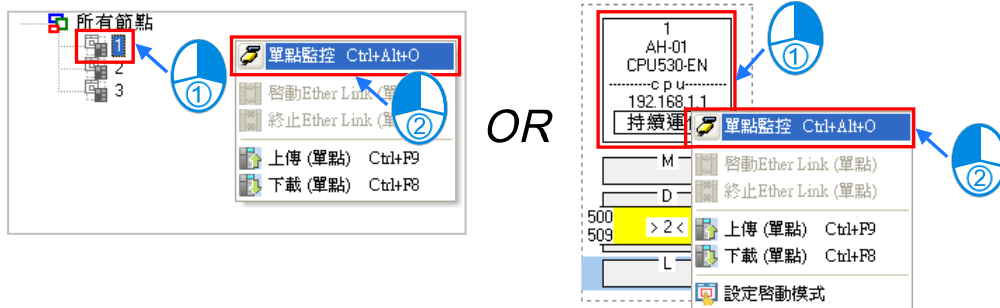
➤ 方法一

在要進行監控的資料需求節點，接著於圖示工具列中將  圖示按下，或是於功能工具列中依序點選 **PLC 主機 (P)** > **單點監控**。而進入單點監控後， 圖示也會呈現按下的狀態。




➤ 方法二

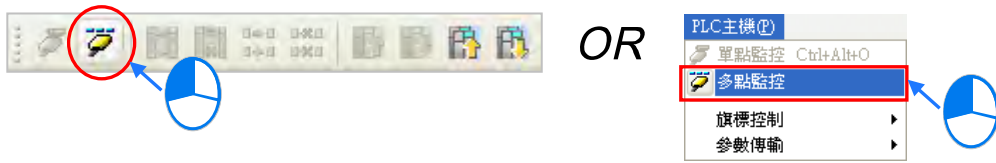
於裝置清單或圖示區中，點選要進行監控的資料需求節點後按下滑鼠右鍵，之後於快捷選單中點選「單點監控」項目。而進入單點監控後， 圖示也會呈現按下的狀態。



● 啟動所有節點的監控

➤ 方法一

於功能工具列點選 **PLC 主機 (P)** > **多點監控**，或直接將圖示工具列中的  圖示按下。



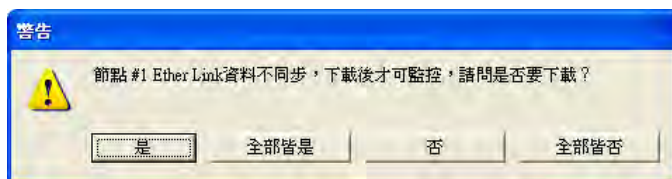
➤ 方法二

於裝置清單中點選「所有節點」後按下滑鼠右鍵，之後於快捷選單中點選「多點監控」項目。



補充說明

進入監控畫面前，系統會先比對視窗中的 Ether Link 設置與主機當中的設置是否相同，若不相同時，系統便會要求重新下載設置參數。

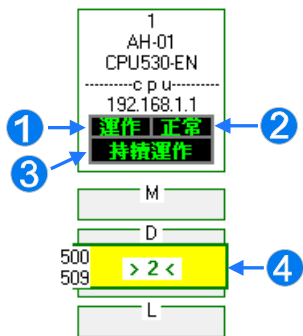


20.4.9.2 監控狀態顯示

當進入監控模式之後，監控中的節點在**裝置清單**中的圖示便會呈現彩色。



而此時在節點的圖示中也會以不同的文字或圖示來顯示目前 Ether Link 的運作狀態。



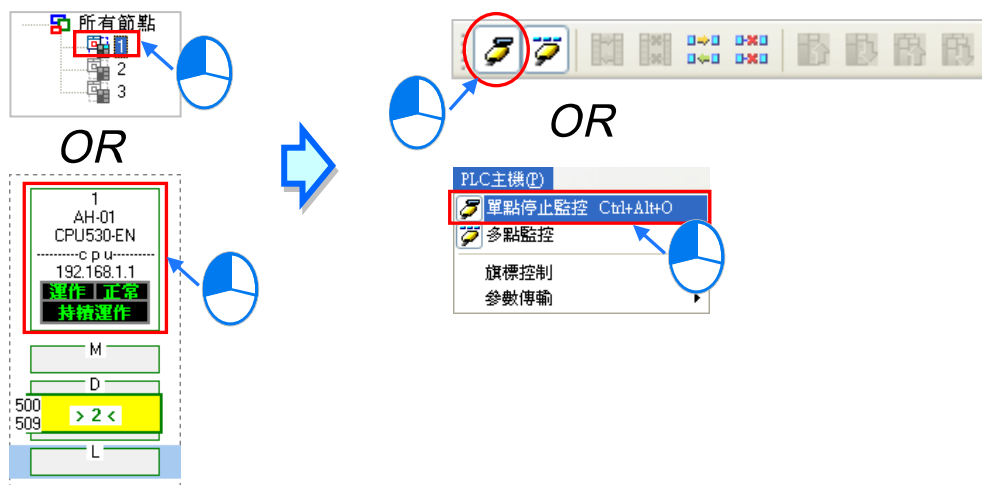
狀態	說明
❶ 運作/停止	運作 ：Ether Link 功能啟動
	停止 ：Ether Link 功能關閉
❷ 運作狀態	正常 ：運作正常
	錯誤 ：運作異常
❸ 啟動模式	保持停止 ：啟動模式為保持停止
	持續運作 ：啟動模式為持續運作
	旗標模式 ：啟動模式為旗標模式
❹ 資料區塊	500 > 2 < 509：資料交換中
	500 ≠ 2 ≠ 509：資料交換動作停止

20.4.9.3 關閉監控

- 關閉單一節點的監控

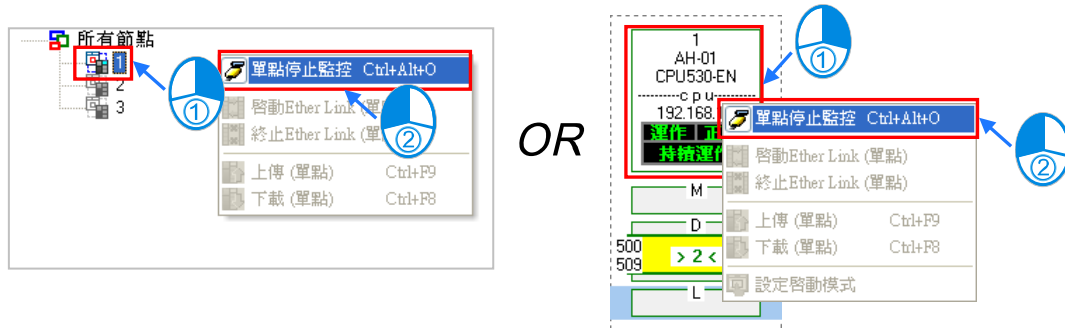
- 方法一

先選取欲關閉監控的**資料需求節點**，接著於圖示工具列中將 圖示重新按起，或是於功能工具列中依序點選 **PLC 主機 (P)** > **單點停止監控**。

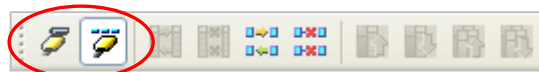


➤ 方法二

於裝置清單或圖示區中，點選要關閉監控的資料需求節點後按下滑鼠右鍵，接著於快捷選單中點選「單點停止監控」項目。



當所選節點的監控功能被關閉之後，點選該節點時，圖示工具列的 圖示會呈現未按下的狀態，但若還有其它的節點處於監控狀態時，圖示工具列的 圖示則仍會呈現按下的狀態。



● 關閉全部節點的監控

➤ 方法一

於功能工具列中點起 **PLC 主機 (P)** > 多點監控，或重新將圖示工具列中的 圖示按起。



➤ 方法二

於裝置清單中點選「所有節點」後按下滑鼠右鍵，接著於快捷選單中將「多點監控」點起。



20.4.10 線上啟動/停止 Ether Link (旗標模式)

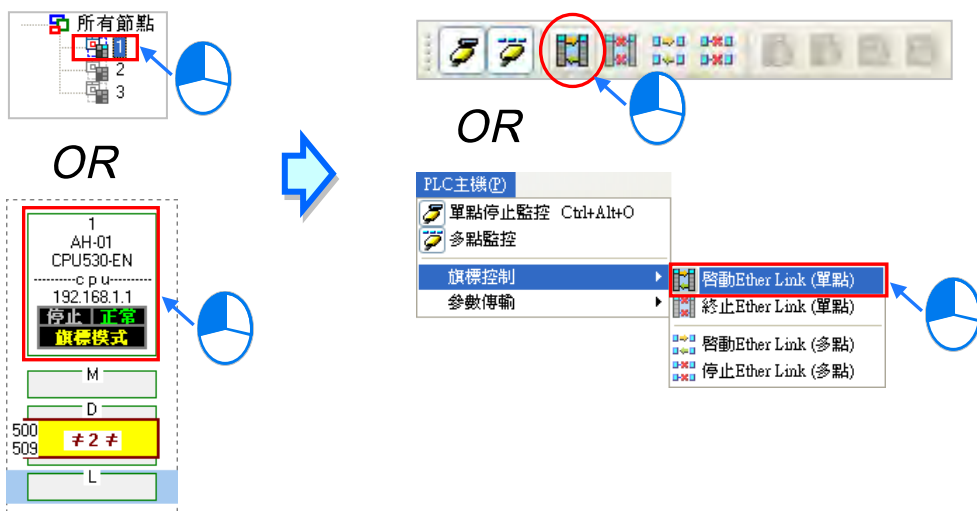
當節點的啟動模式為旗標模式時，於線上監控狀態下便可直接操作該旗標，並藉此啟動或停止某個監控節點的 Ether Link 運作；不過此操作僅限監控中的節點。

20.4.10.1 啟動 Ether Link

- 啟動單一節點的 Ether Link 功能

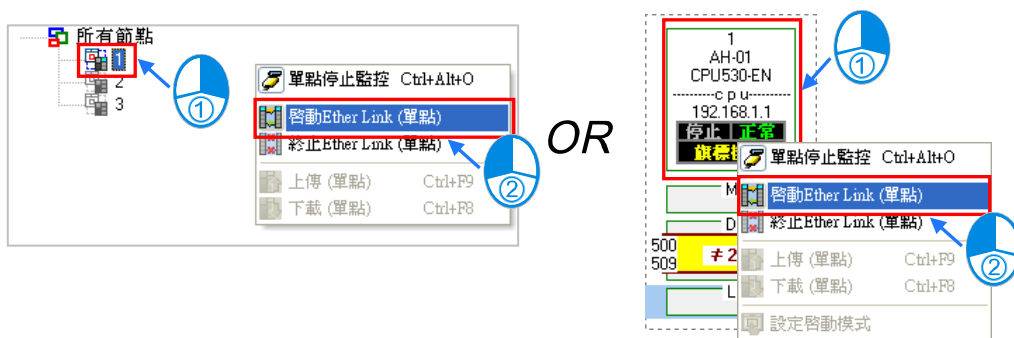
- 方法一

在要啟動 Ether Link 的節點，接著於圖示工具列中點擊  圖示，或是於功能工具列中依序點選 PLC 主機 (P) > 旗標控制 > 啟動 Ether Link (單點)。




- 方法二

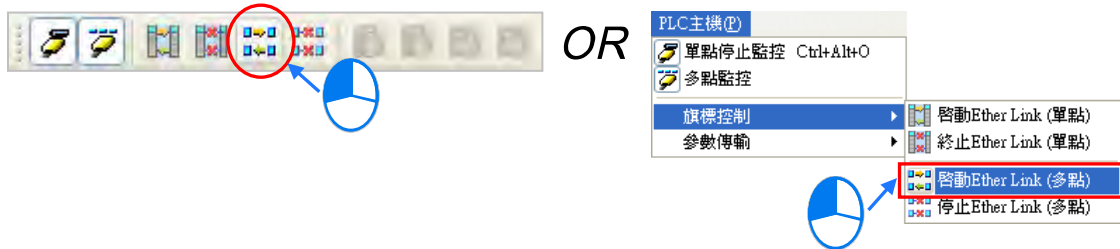
於裝置清單或圖示區中，點選取要啟動 Ether Link 的節點後按下滑鼠右鍵，接著於快捷選單中點選「啟動 Ether Link (單點)」項目。



● 啟動所有節點的 Ether Link 功能

➢ 方法一

於功能工具列中依序點選 **PLC 主機 (P)** > **旗標控制** > **啟動 Ether Link (多點)**，或直接於圖示工具列中點擊  圖示。



➢ 方法二

於裝置清單中點選「**所有節點**」後按下右鍵，接著於快捷選單中點選「**啟動 Ether Link (多點)**」。

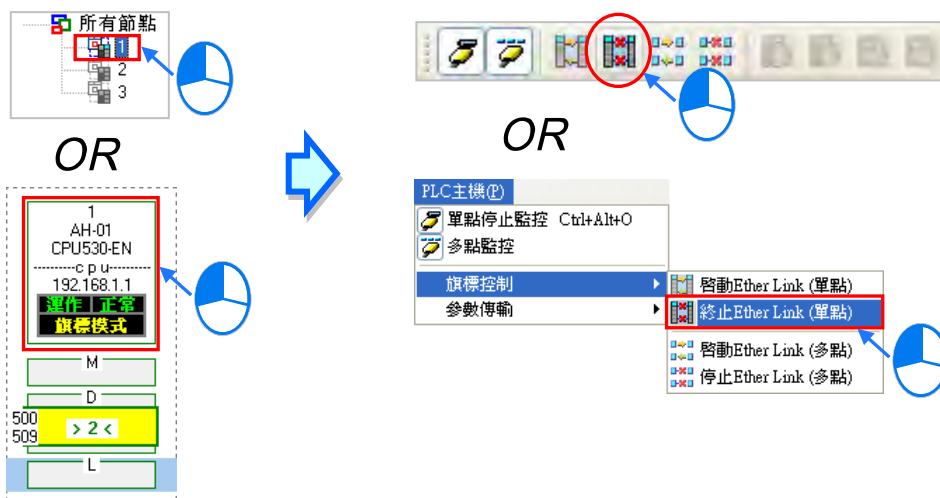


20.4.10.2 停止 Ether Link

● 停止單一節點的 Ether Link 功能

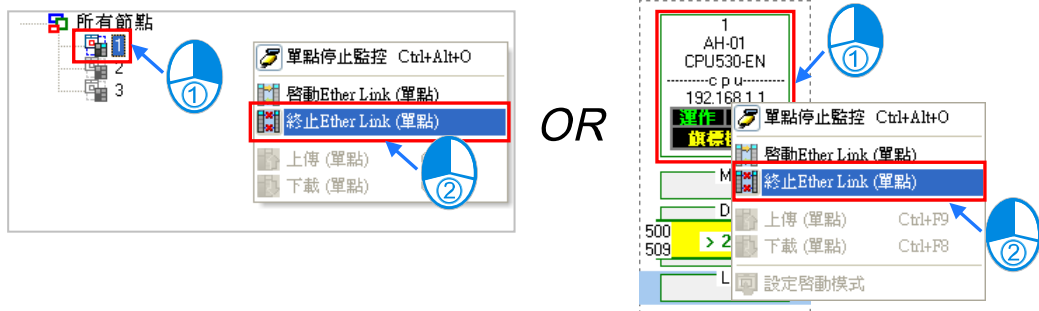
➢ 方法一

在要停止 Ether Link 的節點，接著於圖示工具列中點擊  圖示，或是於功能工具列中依序點選 **PLC 主機 (P)** > **旗標控制** > **終止 Ether Link (單點)**。



➤ 方法二

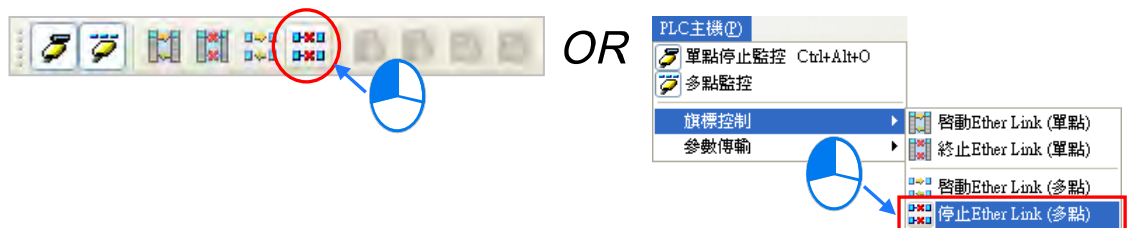
於裝置清單或圖示區中，點選取要停止 Ether Link 的節點後按下滑鼠右鍵，接著於快捷選單中點選「終止 Ether Link (單點)」項目。



● 停止所有節點的的 Ether Link 功能

➤ 方法一

於功能工具列中依序點選 **PLC 主機 (P)** > **旗標控制** > **停止 Ether Link (多點)**，或直接於圖示工具列中點擊 圖示。



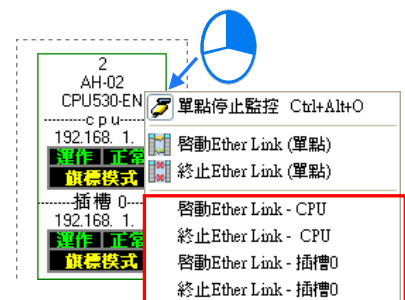
➤ 方法二

於裝置清單中點選「所有節點」後按下右鍵，接著於快捷選單中點選「停止 Ether Link(多點)」。



補充說明

當操作的節點本身包含多個通訊埠口時，於該節點圖示上按下右鍵後，接著便可透過快捷選單來單獨啟動或停止某個埠口的 Ether Link 功能。不過因 Ether Link 的資料交換是以節點為單位，因此即使某個埠口的 Ether Link 功能被關閉了，資料交換的動作仍可能經由該節點的其他埠口持續進行。



20.4.11 監控表與錯誤記錄

於監控節點的資料區塊上雙擊滑鼠左鍵，之後便可開啟該區塊的數值監控視窗，不過此監控視窗的功能僅供數值檢視，使用者無法於此對表格中的數值進行變更。



- 於此欄位可下拉選擇數值的顯示格式。
- 此處顯示的是監控內容所對應的的區塊資訊。
- 於此欄位輸入裝置位址後按下鍵盤的【Enter】鍵，之後便可將該位址移至可視範圍。


當選取某個節點時，下方訊息視窗中的「錯誤記錄」頁面便會顯示該節點主機內部的錯誤記錄；而當勾選「Ether Link 錯誤過濾項目」時，表格中便只會列出與 Ether Link 相關的錯誤記錄；按下「更新」便可重新擷取主機內部的錯誤記錄並將其顯示於表格中，而按下「清除」鍵則可將此頁面與主機內部的錯誤狀態一併清除。




錯誤記錄的內容會包含主機與模組所產生的錯誤記錄，而當某筆錯誤記錄是產生自模組時，除了從「裝置型態」的欄位得知該主機型號之外，也可由於「背板編號」與「插槽編號」兩個欄位得知該模組的所在位置；但若是 CPU 所產生的錯誤，「背板編號」與「插槽編號」欄位則不會有數字顯示。

20.5 NWCONFIG 的管理與應用

20.5.1 存檔與列印

要進行存檔時，請於功能工具列中點選 **檔案 (F) > 儲存**，或直接於圖示工具列中點選  圖示，便可將 NWCONFIG 中的參數進行儲存，且儲存的內容將包括網路配置、PLC Link 參數與 Ether Link 參數；而儲存之後，於專案的路徑下將產生與專案同名，但附檔名為 *.nw 與 *.nwsd 的兩個檔案。



當要進行列印時，請於功能工具列中點選 **檔案 (F) > 預覽列印**，或直接於圖示工具列中點選  圖示，之後系統便會自動開啟**列印管理工具**，但列印的內容只會包含網路部署的圖面。關於**列印管理工具**的說明請參考 **附錄 C**。



若在應用上須要列印 **PLC Link** 或 **Ether Link** 的資料時，請於各自的設定視窗中利用「匯出」的功能將**資料交換表**匯出為 CSV 檔，之後再透過 EXCEL 開啟並將其列印出來即可。

20.5.2 整體下載

於 NWCONFIG 的圖示工作區中，對網路的繞送路徑表 (Routing Table)、PLC Link 的參數與 Ether Link 的參數做整體的同步下載，而下載時也必須逐一完成每個節點資料的下載。

20.5.2.1 下載參數說明

- 繞送路徑表 (Routing Table)

每個主機內部只會儲存與其本身相關的路徑資料，因此不同節點會被下載的繞送路徑表 (Routing Table) 資料也會不同。

- PLC Link 參數

僅有指定為主站的主機須要下載 PLC Link 參數，但對於其他的從站而言，下載 PLC Link 便會將其內部對應 PLC Link 的特殊繼電器與特殊暫存器恢復為預設值；因此若不確定該裝置是否保有先前被設為主站的 PLC Link 參數時，請一併對其下載 PLC Link 參數，以免發生同一網路有多個主站的情形。

- Ether Link 參數

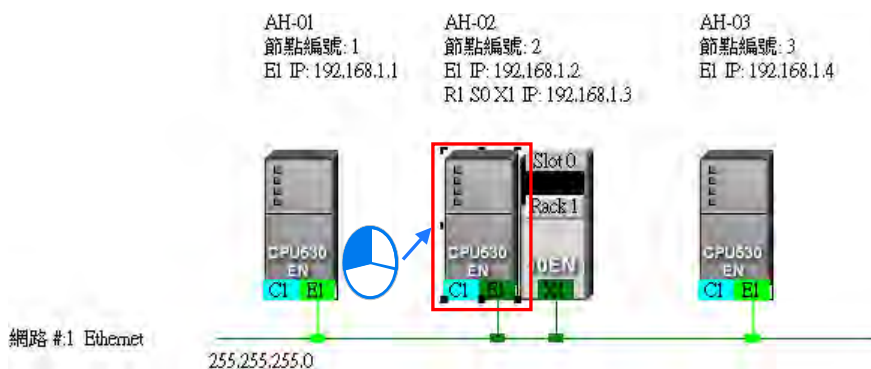
在資料交換表中有提出資料需求的節點都必須進行下載，不過在各主機的內部則僅會儲存與本身相關的資料需求表；若某裝置在 Ether Link 的規劃中沒有設定任何資料需求時，對其進行下載之後便會清除該裝置內部的資料，而同時也將依據 Ether Link 中的規劃設定該裝置的啟動模式。


20.5.2.2 下載操作說明

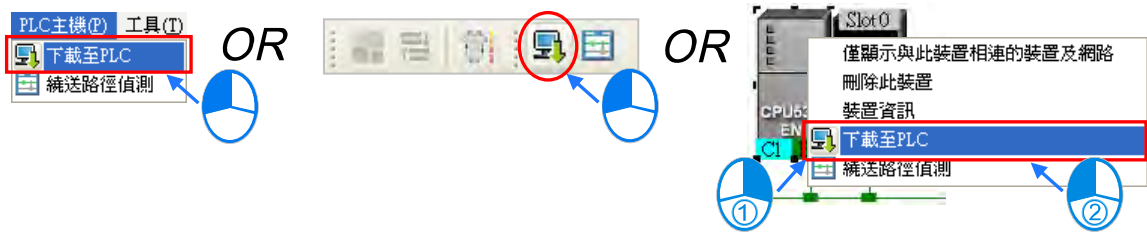
整體下載的方式可分為單點與多點兩種模式，且下載時的通訊設定不可使用封包繞送模式。

- 單點模式


首先請選擇欲下載的節點，並於圖示工作區中點選該節點的主機圖示。



接著於功能工具列中點選 **PLC 主機 (P)** > **下載至 PLC**，或於圖示工具列中點選  圖示，同時也可於主機圖示上點擊滑鼠右鍵，接著再於右鍵快捷選單中點選「**下載至 PLC**」項目即可。

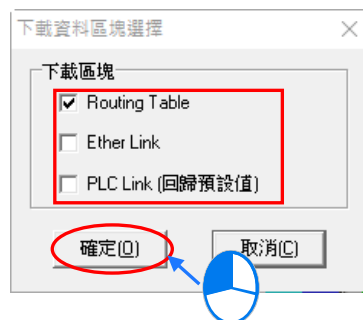


● **多點模式**

於圖示區中點選其中的 **Ethernet 網路** 後，於功能工具列中點選 **PLC 主機 (P)** > **下載至 PLC**，或直接於圖示工具列中點選  圖示即可。而若實際的接線與 NWCONFIG 中的配置一致時，之後便可將參數下載至與該網路連接的所有節點。



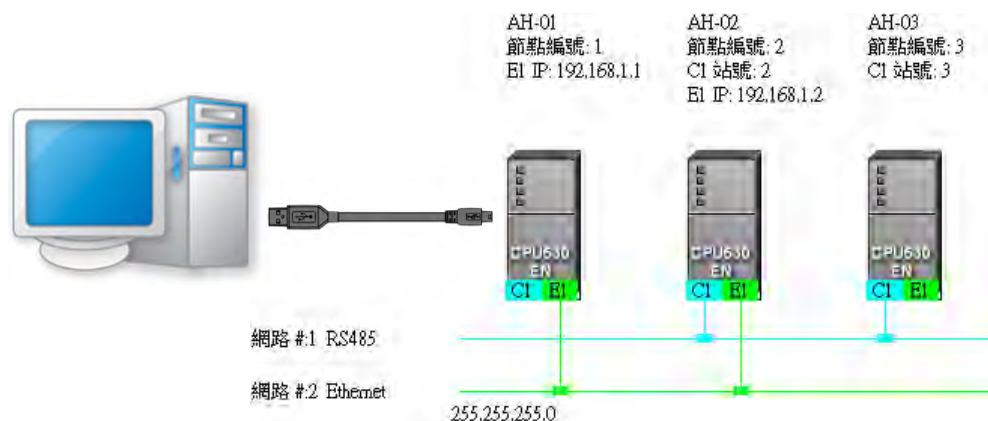
依據上述的步驟操作之後，畫面便會出現下載選項的設定視窗，而其中若有呈現灰階的選項時，表示不支援此項功能；勾選完要下載的項目後按下「**確定**」鍵即可。



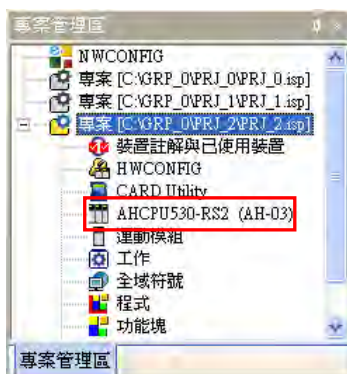
20.5.3 ISPSOft 中的繞送應用

當於 NWCONFIG 中完成網路架構的規劃，且已成功下載繞送路徑表 (**Routing Table**) 至各個節點之後，於 ISPSOft 中便可透過繞送功能來對非直接連接的主機進行上下載或監控；不過當要在 ISPSOft 中使用繞送功能時，其操作的對象則必須都是 AH5x0 系列的機種。

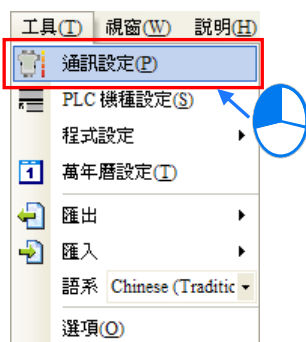
如下圖，當實際連接的主機是 AH-01，但要對 AH-03 進行監控時，透過繞送功能便可達成。



(1) 因要進行監控的是 AH-03，因此請於群組專案中啟動 AH-03 所屬的專案。



(2) 接著於功能工具列中依序點選 **工具 (I)** > **通訊設定 (P)**。



- (3) 於通訊設定視窗中勾選「封包繞送模式」，接著於「初始站台」欄位下拉選擇封包繞送的起點站。一般而言，「初始站台」會是與電腦實際連線的主機，此處的範例為 AH-01，但若電腦同時連接其他主機，或是透過 Ethernet 與多個主機進行連線時，請依據 NWCONFIG 的網路規劃來選擇第一個接收封包的裝置，亦即電腦直接與其通訊的對象。此外，當選擇使用封包繞送模式時，在「通訊通道名稱」欄位所選擇的項目便必須是可與於「初始站台」進行連線的 COMMGR Driver。完成後按下「確定」鍵即可結束設定，而接下來便可直接透過 AH-01 來對 AH-03 進行監控。

通訊設定

通訊通道名稱 Drv_USB

站號 0

封包繞送模式

初始站台 AH-01

確定 取消

MEMO

第21章 資料的備份與還原

目錄

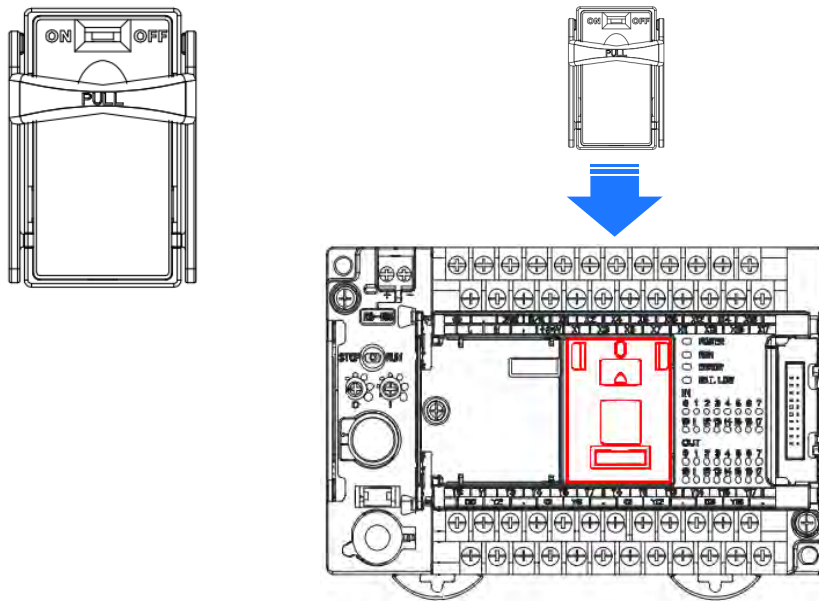
21.1 資料備份記憶卡輔助工具	21-2
21.1.1 資料備份記憶卡簡介	21-2
21.1.2 資料備份記憶卡的操作	21-3
21.2 PLC 永久備份設定	21-3
21.3 記憶卡工具 - CARD Utility	21-4
21.3.1 軟體介紹	21-5
21.3.2 備份操作	21-6
21.3.3 還原操作	21-12
21.3.4 命令列模式指令操作說明	21-15

21.1 資料備份記憶卡輔助工具

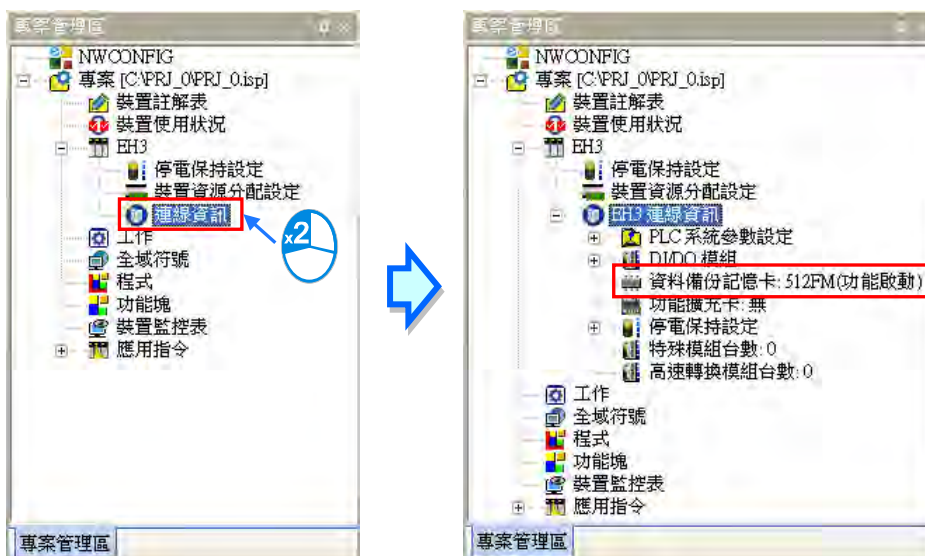
21.1.1 資料備份記憶卡簡介

21

DVP 系列的 EH2 與 EH3 機種可搭配其專屬的資料備份記憶卡，其產品型號分別是 **DVP-256FM** (限 EH2 使用，乳白色外殼) 與 **DVP-512FM** (限 EH3 使用，黑色外殼)，產品外觀如下圖。

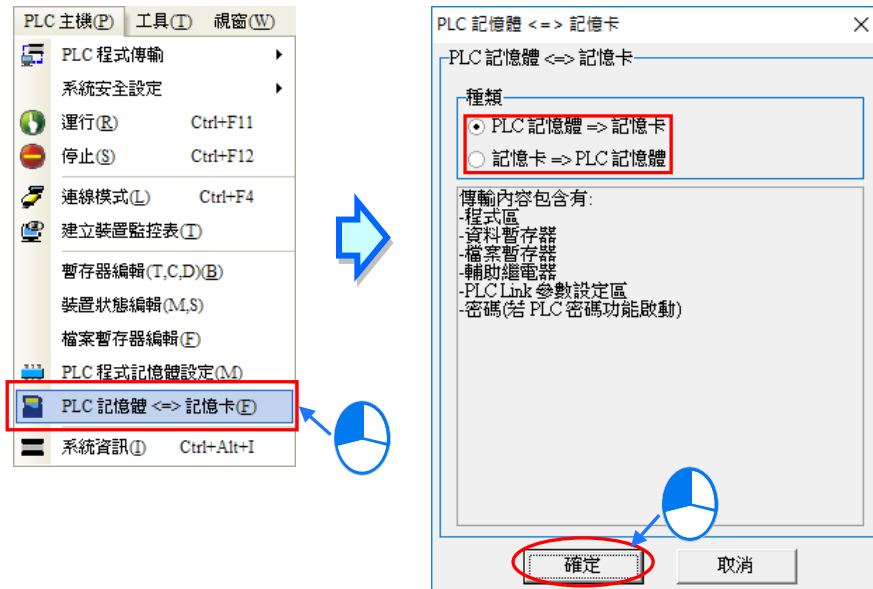


當記憶卡安裝完畢並將 PLC 上電後，於 ISPSOft 的專案管理區中展開機種目錄，接著於「連線資訊」上雙擊滑鼠左鍵，此時若 ISPSOft 可與主機正常連線，則主機的系統資訊便會被顯示在「連線資訊」的下層項目中，而透過其中的「資料備份記憶卡」項目便可得知目前記憶卡的狀態。



21.1.2 資料備份記憶卡的操作

要進行備份記憶卡的操作之前，請先確認 ISPSOFT 與主機正常連線，之後於功能工具列中點擊 **PLC 主機 (P)** > **PLC 記憶體<=>記憶卡 (E)**，接著選擇資料傳輸的種類後按下「**確定**」鍵即可。



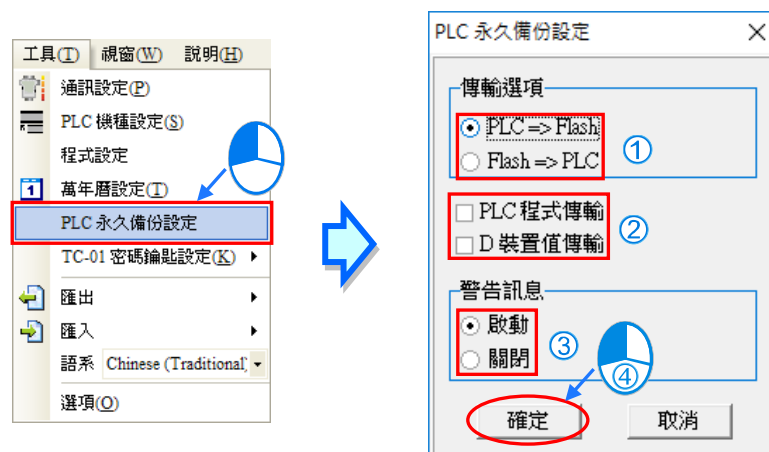
21

*.關於資料備份記憶卡的詳細說明與支援的機種、韌體版本等資訊請參考相關的使用手冊或技術文件。

21.2 PLC 永久備份設定

在 EH2 / EH2-L / EH3 / EH3-L / SV / SV2 幾個機種，其機器本體即有內建 Flash 的記憶空間可用來備份主機的資料。相關功能及使用限制請參考各機種的手冊或說明文件。

要透過 ISPSOFT 進行資料的備份或還原之前，請先確認 ISPSOFT 與主機正常連線，接著於功能工具列中依序點擊 **工具 (I)** > **PLC 永久備份設定**，待出現操作視窗後，請先選擇資料的傳輸選項，接著勾選欲傳輸的內容項目，之後再選擇是否啟動警告訊息 (此功能是設定當資料因電池電壓不足遺失，PLC 自動重新還原資料時，PLC 是否要彈出警告訊息)，完成後按下「**確定**」鍵即可。



21.3 記憶卡工具 - CARD Utility

開始之前請先參考 CARD Utility 備份還原支援列表、確認你的機種是否支援你要備份的項目。

21

	AS DVP-ES3	AH5x0	AH5x1RS	AH5x1EN AH560	AHxxEMC	DVPxxMC
原始程式碼 (註 1)	V	V	V	V	V	V
執行碼	V	V	V	V	V	V
專案密碼 程式識別碼	V	V	V	V	V	V
HWCONFIG 參數設定	V	V	V	V	V	V
主機與模組 配置表、各模 組參數設定	V	V	V	V	V	
裝置內容 (註 2)	M · D · C · HC · FR	X · Y · M · SM · SR · D · B · W · L · S · TB · T · CB · C · HCB · HC · E	X · Y · M · SM · SR · D · W · L · S · TB · T · CB · C · HCB · HC · E	X · Y · M · SM · SR · D · W · L · S · TB · T · CB · C · HCB · HC · E	X · Y · M · SM · SR · D · W · L · S · TB · T · CB · C · HCB · HC · E	(註 3)
通訊資料 交換表	<ul style="list-style-type: none"> • COM1 • COM2 • Ethernet • FEN02 • Function Card1 • Function Card2 	<ul style="list-style-type: none"> • RS485 • Ethernet 	<ul style="list-style-type: none"> • COM1 • COM2 	<ul style="list-style-type: none"> • COM1 • Ethernet • RTU IO table • EIP I/O connection table 	<ul style="list-style-type: none"> • COM2 • Ethernet • EtherCAT 	<ul style="list-style-type: none"> • CANopen
定位規劃表	V					
初始值表			V	V	V	
ECAM					V	V
Gcode						V
軸參數					V	V

註 1：將 ISPSOft 的專案資料匯出為備份檔 (DUP) 時，可選擇是否備份原始程式碼。從主機備份至電腦端或記憶卡時，若主機內原本就沒有下載原始程式碼則產生的備份檔也不包含原始程式碼。不包含原始程式碼的備份檔無法被還原成 ISPSOft 專案。

註 2：裝置備份還原機制

	備份			還原		
	ISP 專案 至電腦	主機 至電腦	主機 至記憶卡	電腦 至 ISP 專案	電腦 至主機	記憶卡 至主機
AS DVP-ES3	有裝置檔 (dvl · dvb · wft) 存在時 會備份裝置	可選擇要 不要備份 裝置	可選擇要 不要備份 裝置	備份檔中有裝 置資料時會還 原裝置檔 (dvl · dvb · wft)	備份檔中有裝 置資料時就會 還原	備份檔中有 裝置資料時 就會還原
AH5x0	不備份裝置			不還原裝置		
AH5x1RS						
AH5x1EN						
AH560						
AHxxEMC						

註 3：DVPxxMC 機種可使用 DMD 檔備份還原停電保持變量當前值。

21.3.1 軟體介紹

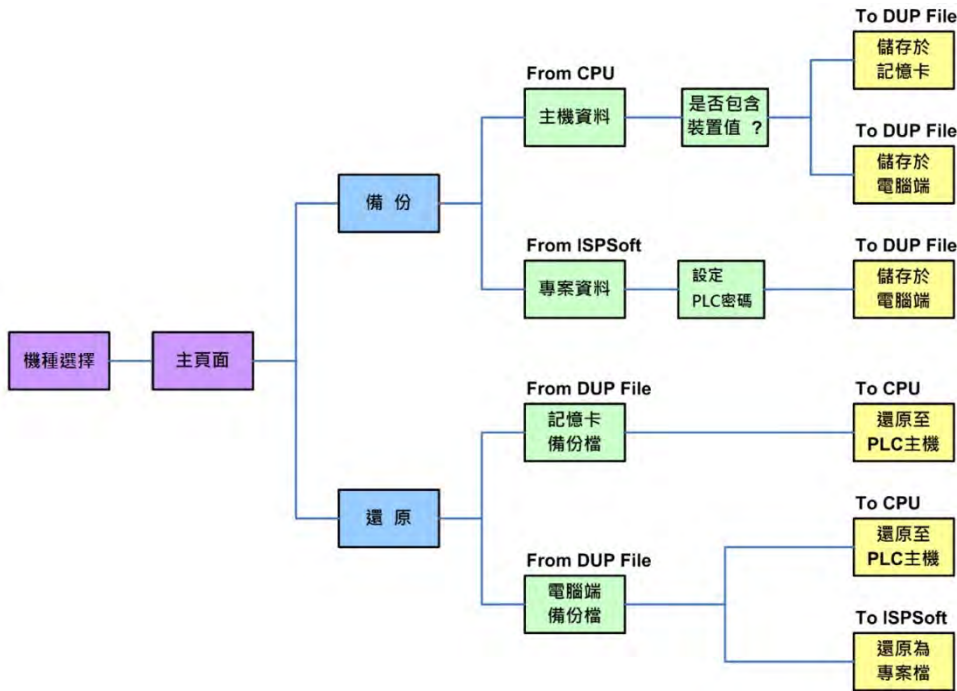
AH/AS 系列機種本身內建 SD 卡的插槽，可以透過記憶卡對主機進行資料的備份與還原；而在 ISPSOFT 中，針對 AH/AS 系列機種也提供便於操作的 **記憶卡工具 - CARD Utility**，其主要的功能是協助使用者在記憶卡、主機及 ISPSOFT 專案中的資料進行備份與還原，而其備份的資料則包含程式碼、參數設定、硬體與網路規劃以及主機內部的裝置內容值，而 HWCONFIG 中所儲存的智慧型模組設定相關資料，除了 ECAT Builder 支援備份與還原外，其他軟體工具尚不支援備份與還原。關於各系列機種的 SD 卡規格與使用限制請參考各機種的操作手冊或相關技術文件。

AH5x0 系列機種關於網路規劃的部份，因每個主機的內部只會儲存與自身相關的網路規劃資料，因此備份時也只會備份與所選 PLC 或 ISPSOFT 專案主機相關的部份，其內容則包括 Routing Table 與 Ether Link 的資料；而同樣的，在進行還原時，即使選擇將備份的資料還原回 ISPSOFT 的專案檔，而還原之後，該專案檔中的 NWCONFIG 資料仍將無法重建。關於網路規劃參數的細節請參考第 20 章當中的相關說明。

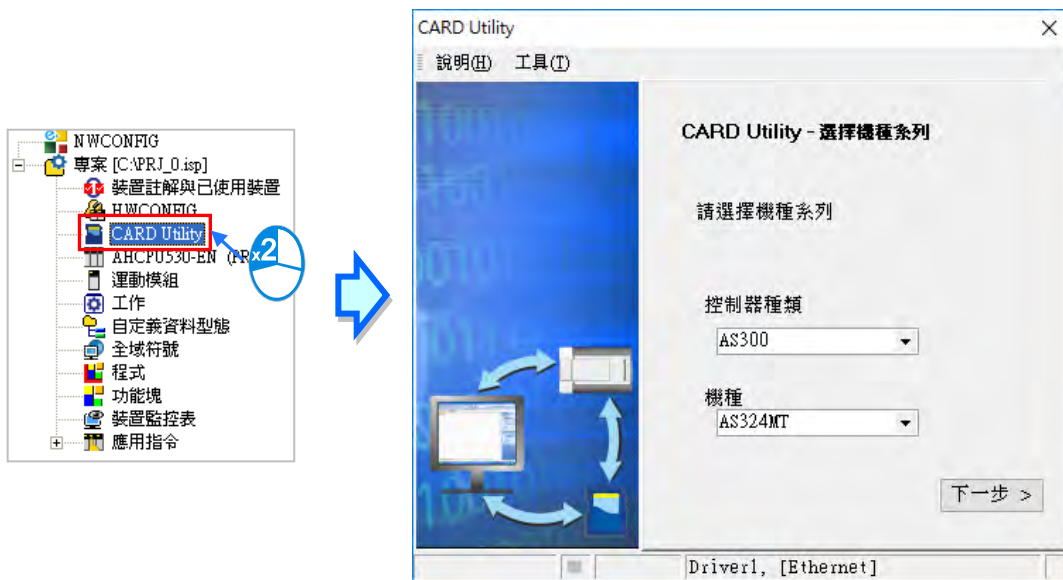
下列為 **CARD Utility** 所支援的功能簡述與流程示意圖。

- 將主機內部存放的資料匯出為備份檔 (*.dup)，並可選擇將其存放在 PLC 端的記憶卡，或是電腦端的自訂路徑。而在操作時，也可選擇是否要將主機內部裝置的目前數值一併進行備份。
- 將 ISPSOFT 的專案資料匯出為備份檔 (*.dup)，但只能存放在電腦端的自訂路徑，AS 系列機種可以將暫存器編輯 (.dvl)、裝置狀態編輯 (*.dvb)、檔案暫存器編輯 (*.wft) 所儲存的檔案視為裝置值進行備份。
- 將記憶卡 (PLC 端) 中的備份檔重新載回 CPU 主機當中。
- 將儲存在電腦端的備份檔 (*.dup) 重新載回主機當中，或是選擇重新還原為 ISPSOFT 的專案檔；而 AH 系列機種當選擇還原為 ISPSOFT 專案時，系統將自動略過裝置內容值以及網路規劃的部份。

AS 系列機種將電腦端的備份檔還原為 ISPSOft 專案時，暫存器編輯 (.dvl)、裝置狀態編輯 (*.dvv)、檔案暫存器編輯 (*.wft) 的內容也會被還原。



要啟動軟體時，請於專案管理區的「CARD Utility」項目上雙擊滑鼠左鍵，系統便會自動開啟操作精靈頁面，先選擇要進行操作的系列機種，按下「下一步」按鈕進入主頁面。




21.3.2 備份操作

當要進行備份的來源或目的位置有任何一個為 CPU 主機或記憶卡時，請在執行前確認 ISPSOft 與主機正常連線。

- (1) 於主頁面選擇「備份 (至 DUP 檔)」選項，之後按「下一步」進行後續的操作。



- (2) 選擇備份的來源項目，確定後按「下一步」以進行後續的操作。

當選擇「ISP 專案」時，請接著點擊下方欄位的  按鈕以選擇欲備份的 ISP 專案檔 (*.isp)，但若所選檔案中的程式尚未完成編譯，則於執行時系統將會顯示提示訊息；此時請重新編譯並儲存，完成之後便可再次對該檔進行備份的操作。



- (3) 當備份的來源為「CPU」時，便會提示備份是否包含主機裝置的內容值。



- (4) 選擇備份檔 (*.dup) 的存放位置，若選擇的備份來源為 ISPSoft 專案時，此處則僅可選擇存放於電腦端。

- a. 選擇存放於 PLC 端的記憶卡時，而 AH 系列機種備份檔的檔名與路徑會固定如下。

記憶卡根目錄\PLC CARD\AH500\SysDup\AUTOEXEC.dup。

AS 系列機種備份檔的檔名與路徑則會固定如下。

- **AS300 系列機種**

記憶卡根目錄\SDCard\PLC CARD\AS300\SysDup\AUTOEXEC.dup。

- **AS200 系列機種**

記憶卡根目錄\SDCard\PLC CARD\AS200\SysDup\AUTOEXEC.dup。

- b. 選擇存放於電腦端時，請點擊下方欄位右側的  按鈕以設定檔名與路徑。



- (5) 若選擇的備份來源為 ISPSOft 專案時，可另行設定 PLC 密碼，選擇 **PLC 密碼與專案密碼相同** 或是 **設定新的 PLC 密碼**，若選擇設定新的 PLC 密碼，將會出現密碼設定視窗供使用者設定密碼與限制次數。

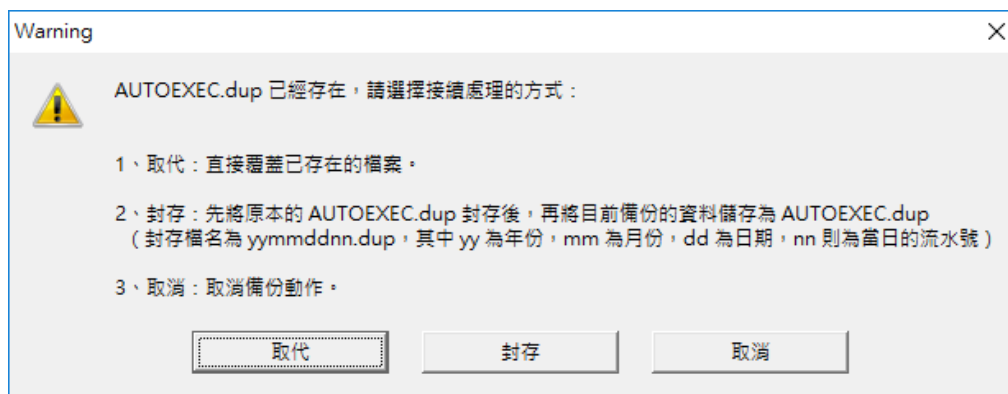


(6) 確認摘要的內容與欲進行的操作一致後，按下「執行」即可開始進行備份的動作。



將主機的資料備份至記憶卡的過程中，使用者可按下「取消」鍵來中斷軟體端的動作，但主機端的系統仍可能會繼續處於備份的狀態，此時則可將主機斷電以強制中斷，但所產生的備份檔便可能不完整，因此事後請務必將記憶卡中的備份檔刪除。

當選擇將資料備份至記憶卡時，因其檔名與路徑皆是固定的，因此若記憶卡中已存在舊有的備份檔時，系統將會出現詢問視窗，此時請依據提示的訊息以選擇最適合的做法。



當備份的來源有開啟密碼保護的功能時，其處理的機制如下。

操作項目	說明
CPU → 記憶卡	CPU 中的「PLC 識別碼」與「PLC 密碼」會一併包含於備份檔。

操作項目	說明
CPU → 電腦端	系統會先要求輸入主機的「PLC 識別碼」與「PLC 密碼」，並在確認無誤後才會開始進行備份；而完成備份之後，「PLC 識別碼」與「PLC 密碼」的資料將會一併包含於備份檔中。
ISPSOft 專案 → 電腦端	專案中的「程式識別碼」、「專案密碼」與備份時輸入的「PLC 密碼」會一併包含於備份檔。

21

- (7) 完成操作後，接著便可選擇回到主頁面繼續操作，或是直接關閉。



21.3.3 還原操作

當要進行還原的來源或目的位置有任何一個為 CPU 主機或記憶卡時，請在執行前確認 ISPSOft 與主機正常連線。

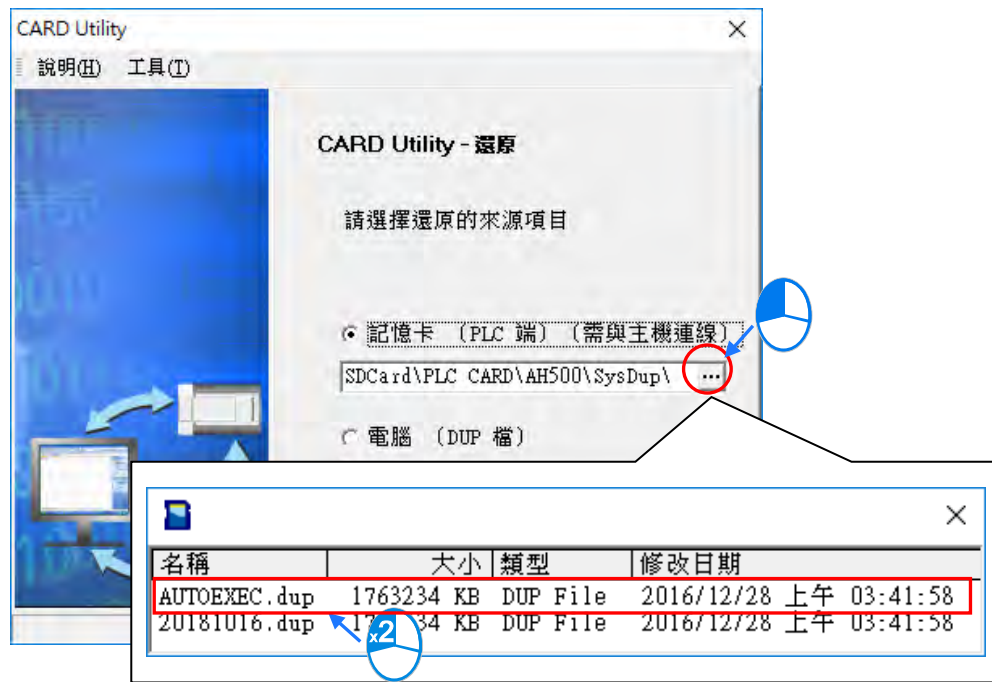
(1) 於主頁面選擇「還原 (從 DUP 檔)」選項，之後按「下一步」以進行後續的操作。



(2) 點擊路徑欄位右側的  按鈕選擇欲還原的備份檔。



若指定的備份檔是存放於 PLC 端的記憶卡，則選擇檔案時，系統會自動連線至 PLC 並將記憶卡指定路徑中的所有備份檔顯示於視窗中，接著在要還原的備份檔上雙擊滑鼠左鍵即可。



- (3) 選擇還原目的位置，完成設定後請按「下一步」。
- 若要將備份檔的資料重新載回主機時，請選擇「CPU」項目。
 - 若選擇「ISP 專案」時，請接著點擊路徑欄位的 ... 按鈕以設定還原後的檔名與路徑，而當設定的路徑為已經存在相同檔案名稱時，執行之後該檔案的內容將被還原的資料所覆蓋。



(4) 確認摘要的內容與要進行的操作一致後，按下「執行」即可開始進行還原的動作。



當執行的操作是將資料還原至主機時，若使用者按下等待視窗中的「取消」而中斷了系統的動作，此時所還原的資料將可能不完整，若不再重新執行還原操作，請務必將 CPU 主機恢復為出廠預設值，以免造成運作上的錯亂。若備份檔的來源位置是記憶卡時，即使在軟體端取消了還原動作，主機系統仍可能會繼續執行還原的程序，此時可將主機斷電以強制中斷。

當還原的來源或目的位置有包含密碼的資訊或開啟密碼保護的功能時，其處理的機制如下。

操作項目	說明
記憶卡 → CPU	<ul style="list-style-type: none"> a. 備份檔與 CPU 中的識別碼必須一致，否則便中斷程序。 b. 當 CPU 中有設定「PLC 密碼」時，備份檔與 CPU 中的密碼必須一致，否則便中斷程序。 c. 當 CPU 沒有設定「PLC 密碼」，而備份檔有包含密碼時，系統會執行還原並將備份檔的密碼設為 CPU 的「PLC 密碼」。
電腦端 → CPU	<ul style="list-style-type: none"> a. 備份檔與 CPU 中的識別碼必須一致，否則便中斷程序。 b. 當 CPU 中有設定「PLC 密碼」時，備份檔與 CPU 中的密碼必須一致，否則便中斷程序並顯示提示訊息。 c. 當 CPU 沒有設定「PLC 密碼」，而備份檔有包含密碼時，系統會執行還原並將備份檔的密碼設為 CPU 的「PLC 密碼」。
電腦端 → ISPSOft 專案	<p>還原時，若備份檔中有設定「PLC 密碼」，則會出現提示視窗詢問使用者輸入，正確才可進行還原。還原後，備份檔中的專案密碼與程式識別碼將被還原到 ISPSOft 專案；而備份檔中的「PLC 密碼」、「PLC 識別碼」則無法保存至 ISPSOft 專案。</p>

(5) 完成操作後，接著便可選擇回到主頁面繼續操作，或是直接關閉。



21.3.4 命令列模式指令操作說明

CARD Utility 支援透過命令列模式的指令操作，如需要使用可以到 ISPSOFT 的安裝目錄下找到 PLC CARD.exe 的執行檔，預設安裝路徑為 C:\Program Files (x86)\Delta Industrial Automation\ISPSOFT 3.10\PLC_CARD，接著按照下列說明操作。

- 指令與指令之間使用空白做為區隔。
- /Lock · /UnLock 不能同時使用。/C 與 /IP 不能同時使用。
- 指令及參數不區分大小寫，只有密碼設定有區分。
- 不需要輸入參數的指令

指令	功能
/RUN	啟動 PLC
/Stop	停止 PLC
/H	隱藏所有訊息視窗，但程式結束會回傳訊息代碼
/SaveOFF	關閉回傳訊息存檔功能
/?	顯示幫助訊息。使用此指令時，其他指令無效。

- 需要輸入參數的指令 (Pn 代表輸入的參數，參數之間以","隔開)

指令	參數	功能
/PLC=P1	P1=PLC 型號	指定要通訊的 PLC 型號，例如：/PLC=AS332T。 支援的型號如下： AS300N · AS320T · AS320P · AS324MT · AS332T · AS332P。 AS218TX · AS218PX · AS218RX · AS228T · AS228P · AS228R · ES3。
/F=P1	P1=檔案名稱	下載 DUP 檔案到 PLC，例如：/F=test.dup
/C=P1	P1=通訊埠編號	設定 PC 的 COM Port 通訊埠編號，例如：/C=1
/Lock=P1, P2	P1=要設定的密碼， P2=限制次數 (限制次數設 0 表示不限制)	PLC 無密碼狀態，且 DUP 檔中亦無密碼，當下載完成之後，可以此指令設定 PLC 密碼，例如：/Lock=1234, 0。(註：若 DUP 檔中有密碼則下載完成後會自動將該密碼設定成 PLC 密碼)
/UnLock=P1	P1=驗證的密碼	PLC 已經被鎖定密碼，若 DUP 檔中無與 PLC 相同的密碼，下載前須解除 PLC 密碼，例如：/Unlock=1234。(註：解除 PLC 密碼後，PLC 處於無密碼狀態，若要再次鎖定密碼需使用/Lock 指令或下載包含密碼的 DUP 檔)
/P=P1, P2, P3, P4	P1=Baud Rate P2=Data Bits P3=Parity P4=Stop Bits	設定通訊協定，預設值為 9600, 7, e, 1 不更改的值不用輸入，但是","一定要輸入。 ex1：全部都要更改 /p=115200, 8, n, 2 ex2：只更改 Baud Rate /p=115200, ,, " ,"一定要輸入
/Save=P1	P1=檔案名稱	指定回傳訊息代碼儲存路徑及檔名。 指定路徑目錄不存在時，會自動產生該目錄。未使用此指令或指定參數為空時 (/save=)以預設檔名"Result.txt"存在當前路徑。 Ex：/Save=C:\test\result1.txt 儲存內容請參考下面所列之回傳代碼表。
/IP=P1	P1=IP address	有指定/IP 指令時，表示以 Ethernet 方式通訊，P1 設定通訊之 IP address。此指令不能與/C 參數同時使用。 Ex: /IP=192.168.1.5

範例：

Ex1：經由 COM port 1 將 test.dup 下載到 PLC AS332T 並執行 PLC，通訊協定使用預設值 9600, 7, e, 1

- 指定位於當前路徑下的檔案："PLC CARD.exe" /PLC=AS332T /F=test.dup /RUN /C=1
- 指定位於特定路徑下的檔案："d:\temp\PLC CARD.exe" /PLC=AS332T /F=d:\temp\test.dup /RUN /C=1

Ex2：經由 Ethernet 停止 PLC AS324MT

"PLC CARD.exe" /PLC=AS324MT /Stop /IP=192.168.1.1

回傳代碼表

代碼	訊息
1	正常結束!
2	通訊逾時!
3	PLC 的型號不符!
4	DUP 檔內容錯誤!
5	通訊埠已被使用或無此通訊埠!
6	無法開啟檔案!
7	命令錯誤!
8	命令 H 錯誤!
9	命令 P 錯誤!
10	保留
11	通訊埠號碼錯誤!
12	命令 R 錯誤!
13	鮑率錯誤!
14	資料長度錯誤!
15	同位元錯誤!
16	停止位元錯誤!
17	命令 STOP 錯誤!
18	保留
19	密碼已上鎖!
20	保留
21	保留
22	保留
23	命令 C 錯誤!
24	保留
25	命令 Lock 錯誤!
26	命令 UnLock 錯誤!
27	密碼驗證失敗!
28	保留
29	PLC 已上鎖，請先移除密碼!
30	命令 IP 錯誤!
31	保留
32	無法取得正確的 PLC ID 狀態!
33	PLC ID 比對失敗!
34	通訊錯誤!

21

代碼	訊息
35	保留
36	不支援該型號!
37	保留
38	寫入 PLC 失敗!
39	命令失敗!
40	命令 PLC 錯誤!

第22章 G-Code 編輯器與 E-CAM 編輯器

目錄

22.1 G-Code 編輯器	22-2
22.1.1 關於 G-Code	22-2
22.1.2 使用 G-Code 編輯器	22-2
22.1.3 G-Code 的指令結構.....	22-6
22.1.4 G-Code 功能詳細介紹.....	22-8
22.1.4.1 G0 (快速定位)	22-8
22.1.4.2 G1 (直線補間)	22-10
22.1.4.3 G2 (順時針圓弧補間/螺旋補間)	22-13
22.1.4.4 G3 (逆時針圓弧補間/螺旋補間)	22-15
22.1.4.5 G4 (時間延遲)	22-18
22.1.4.6 G17/G18/G19 (平面選擇)	22-19
22.1.4.7 G50 (準確停止)	22-20
22.1.4.8 G51 (圓弧交接)	22-21
22.1.4.9 G52 (圓滑交接)	22-22
22.1.4.10 G90 (絕對模式)	22-24
22.1.4.11 G91 (相對模式)	22-25
22.1.4.12 M 碼.....	22-26
22.1.5 G-Code 範例-三軸同動繪出 Delta LOGO	22-28
22.2 E-CAM 編輯器	22-29
22.2.1 關於 E-CAM (電子凸輪)	22-29
22.2.2 電子凸輪意義說明.....	22-29
22.2.3 使用 E-CAM 編輯器	22-30

22.1 G-Code 編輯器

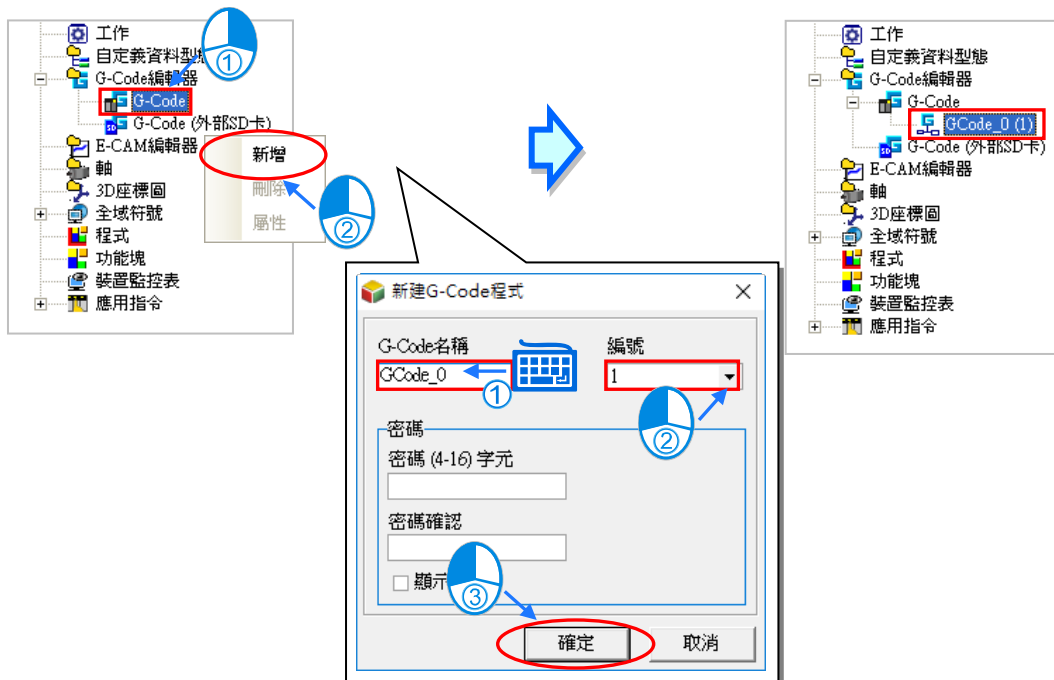
22.1.1 關於 G-Code

進行運動控制時，為了在兩軸或兩軸以上進行複雜的加工軌跡，使用者可以透過計算機輔助製造軟體 (Computer Aided Manufacture) 將所需加工路徑生成 G-Code，再將 G-Code 輸入控制器，來驅動各軸加工。由此可以節省運動程式開發的時間。G-Code 是一種廣泛用於自動化設備的 CNC (Computer Numerical Control) 程式語言。

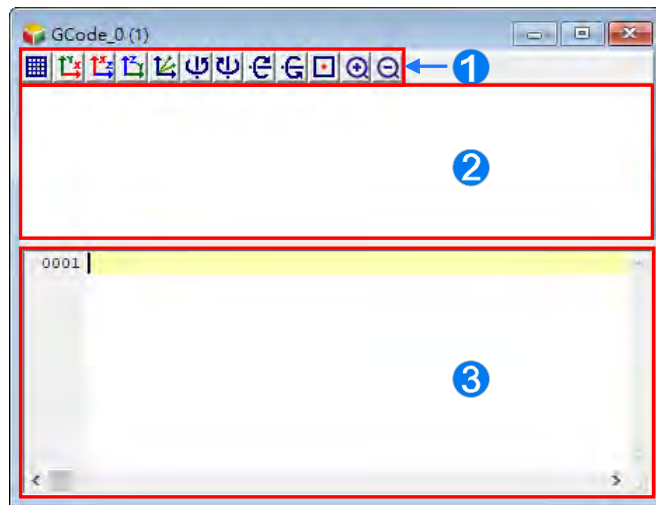
22

22.1.2 使用 G-Code 編輯器

專案管理區的 G-code 編輯器項目下會有兩個子項目，下載時分別會配置到不同的位置，G-Code 項目將存放於主機；而 G-code (外部 SD 卡) 則會存放至主機上安裝的 SD 記憶卡中 (目前 ISPSOft 針對 DVPxxMC 機種不支援此功能)。於選定的項目點擊 **右鍵 > 新增**，出現新增視窗如下圖，於 G-Code 名稱輸入此項目名稱，再選擇此項目的編號，若需要可以設定密碼保護 G-Code 設計資料，下次開啟編輯視窗時將會詢問密碼。主機上最多可建立 64 組 G-code 項目，每組最大 250KB。



完成新建後將會在編輯工作區開啟 G-Code 編輯視窗，功能說明如下：



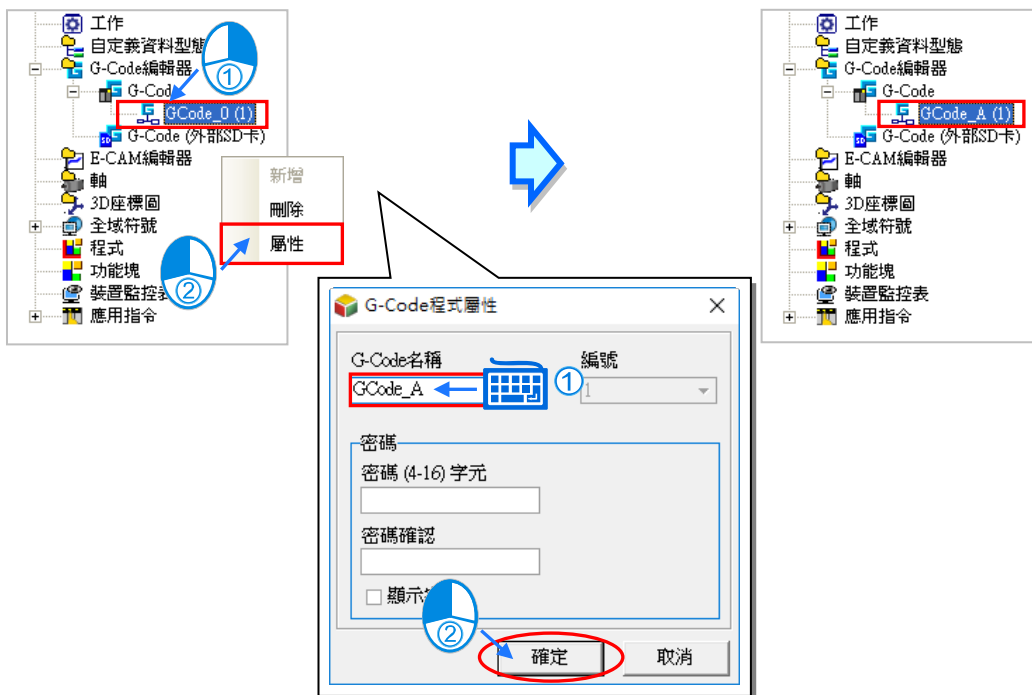
22

- ❶ 圖示工具列：提供各項功能按鈕，對 G-Code 軌跡示意區畫面進行調整，按鈕功能說明如下。
- ❷ 軌跡示意區：依據 G-Code 程式編寫區中的程式，即時顯示對應的路徑軌跡。
- ❸ G-Code 程式編寫區：在此處輸入 G-Code。

圖示工具列按鈕功能說明

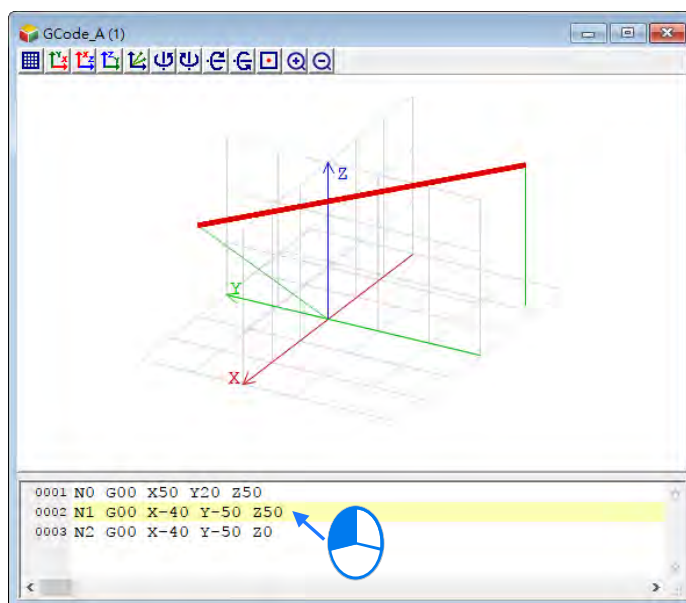
圖示	名稱	功能說明
	開啟/關閉格線	顯示軌跡示意區格線，分別開啟/關閉 XY、YZ、XY 平面的格線。
	切換到 XY 平面	切換軌跡示意區到 XY 平面
	切換到 XZ 平面	切換軌跡示意區到 XZ 平面
	切換到 YZ 平面	切換軌跡示意區到 YZ 平面
	顯示 G0 軌跡	顯示 G0 多軸高速定位的軌跡
	垂直逆時針旋轉	對目前軌跡示意區垂直軸逆時針旋轉
	垂直順時針旋轉	對目前軌跡示意區垂直軸順時針旋轉
	水平往上旋轉	對目前軌跡示意區水平軸往上旋轉
	水平往下旋轉	對目前軌跡示意區水平軸往下旋轉
	最適大小	調整軌跡示意區到最適大小
	放大	放大軌跡示意區圖形
	縮小	縮小軌跡示意區圖形

在已建立的 G-Code 項目上點擊滑鼠右鍵，可於快捷選單點選「刪除」，刪除此項目；或是點選「屬性」，重新設定項目名稱或密碼，但不可變更編號。

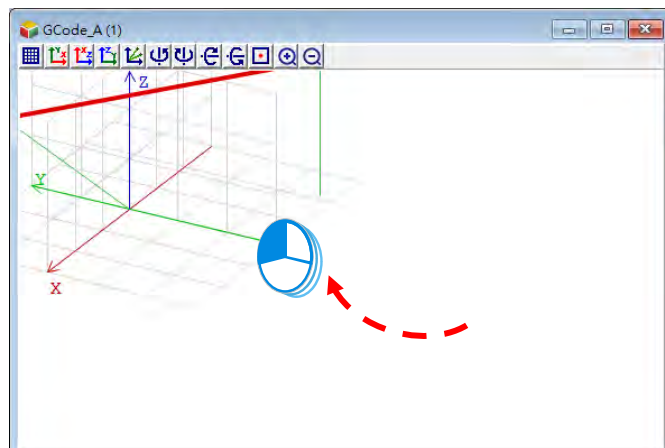
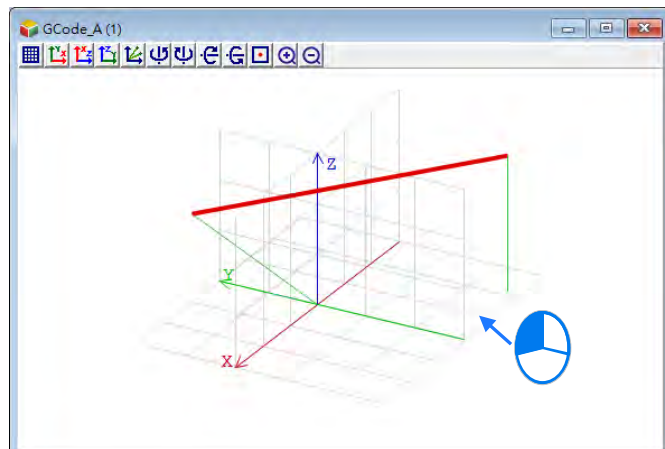


22

當 G-Code 程式編寫區點選某一行程式時，軌跡示意區對應的 G-Code 軌跡將以紅色線段顯示。



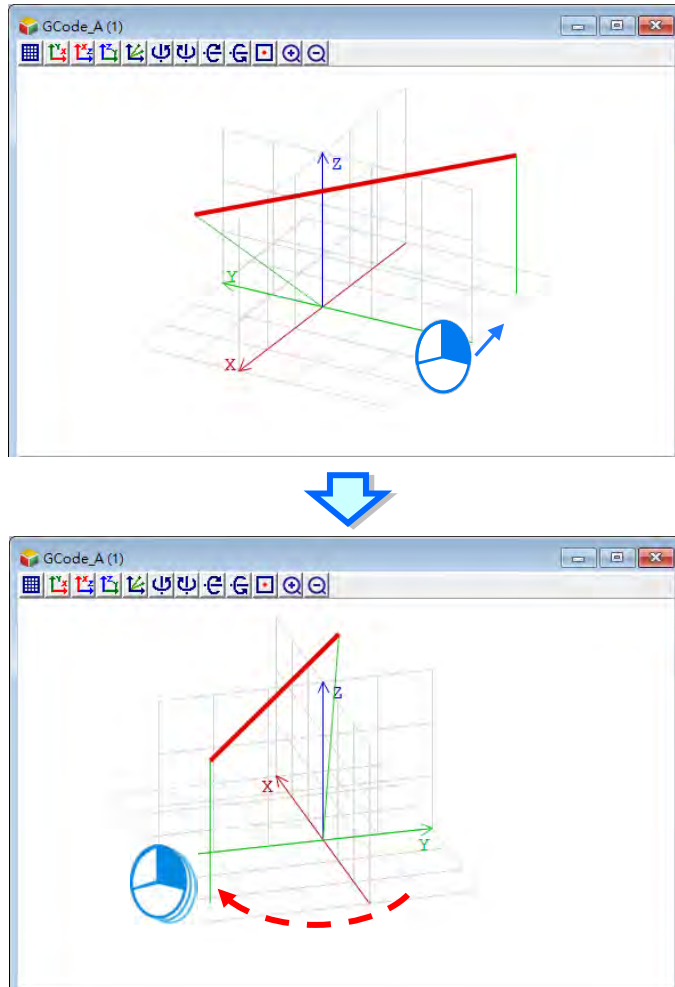
使用者可以在軌跡示意區按住滑鼠左鍵，拖曳滑鼠移動畫面顯示的區域。



22

使用者可以在軌跡示意區按住滑鼠右鍵，拖曳滑鼠旋轉畫面顯示的區域。

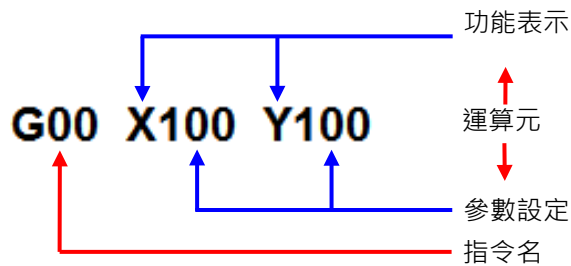
22



完成設定後便可與專案一同下載至主機，專案的上下載請參考第 17.1.3 節。

22.1.3 G-Code 的指令結構

G-Code 指令的結構可分為指令名及運算元；運算元又包括了功能表示與參數設定兩部分，如下所示。指令名表示指令執行的功能；功能表示則為運算元執行此指令的目標，例如目標軸位置或速度；參數設定則為運算元參數的數值。以下 G00 指令，為以最快速度移動 X 軸與 Y 軸到指定的目標位置(100, 100)。



所支援的 G-Code 指令的格式與用法以下列出幾點基本事項。

- 每一列 G-Code 開頭都必須有一行號，例如 N1、N2、... 等等
- 在不破壞指令格式下，是否有空白皆是被允許的。例如 G01X50 Y15F20 合法且相同於 G01 X50 Y15 F20。
- G00、G01、G02、G03 指令有繼承功能（即下一行省略 G 代碼可比照之前的 G 代碼）其他種類的 G 代碼無繼承功能；G01、G02、G03 速度參數 F 有延續功能。如下所示：

```

N0 G00 X50 Y15
N1 X-40 Y-50 ← G00 X-40 Y-50

N2 G01 X10 Y25 F20
N3 G03 X-40 Y-50 R100
      ↑
      G03 X-40 Y-50 R100 F20

```

22

- 運算元參數常數可輸入型式：阿拉伯數字及浮點數。
例：N0 G0 X100.0 Y100.0
- 高速定位指令（G0）不需設定移動速度。
例：N0 G0 X100.2 Y500.0
- G00、G01 指令中，軸參數 X、Y、Z 至少存在一軸（參數），其他省略的軸參數值參照前面最接近的正確行中相同參數的值。
- 註解支援方式有兩種：
 - 以（和）包覆，且不允許巢狀註解，例：（這是註解）
 - 在 % 後的文字，例：%這是註解
- 編寫指令方式為更符合一般 G 代碼，運算元參數標記的順序可不用按照 X、Y、Z、I、J、K、R 等順序編寫，可任意編寫順序。

例如：N0 G1 X100 Z300 Y200.45 F400 => N0 G1 X100 Y200.45 Z300 F400

DVPxxMC 系列運動控制器支援的 G 碼及輸入格式如下表所示：

G 代碼	功能說明	格式
G0	快速定位	N_G0 X_Y_Z_A_B_C_
G1	直線補間	N_G1 X_Y_Z_A_B_C_F_
G2	順時針圓弧補間/螺旋補間	N_G2 X_Y_Z_I_J_F_ N_G2 X_Y_Z_I_K_F_ N_G2 X_Y_Z_J_K_F_ N_G2 X_Y_Z_R_F_

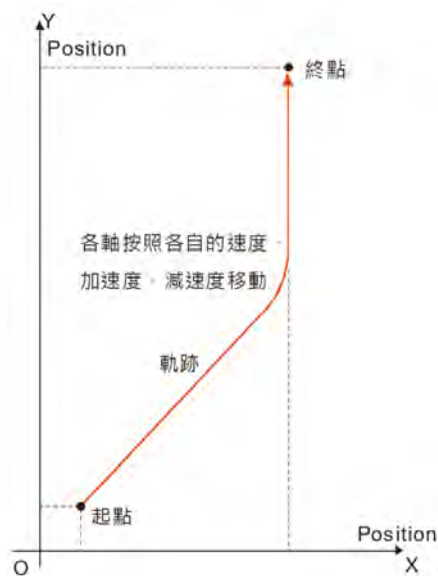
G 代碼	功能說明	格式
G3	逆時針圓弧補間/螺旋補間	N_G3 X_Y_Z_I_J_F_ N_G3 X_Y_Z_I_K_F_ N_G3 X_Y_Z_J_K_F_ N_G3 X_Y_Z_R_F_
G4	時間延遲	N_G4 X_ N_G4 P_
G17	XY 平面選擇	N_G17
G18	ZX 平面選擇	N_G18
G19	YZ 平面選擇	N_G19
G50	準確停止	N_G50
G51	圓弧交接	N_G51 D_
G52	圓滑交接	N_G52
G90	絕對模式	N_G90
G91	相對模式	N_G91
M0~M99	M 碼	N_M_D_

22.1.4 G-Code 功能詳細介紹

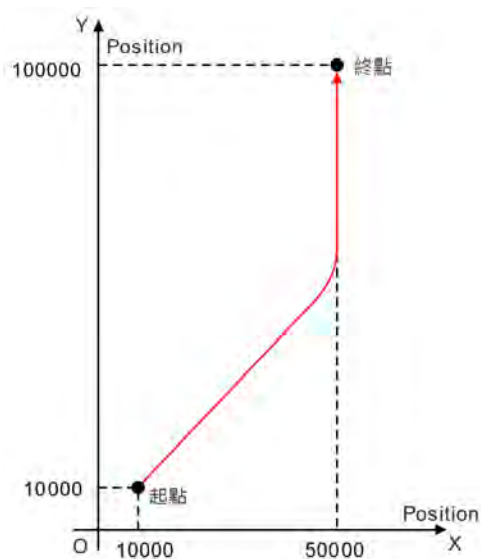
22.1.4.1 G0 (快速定位)

- 功能：

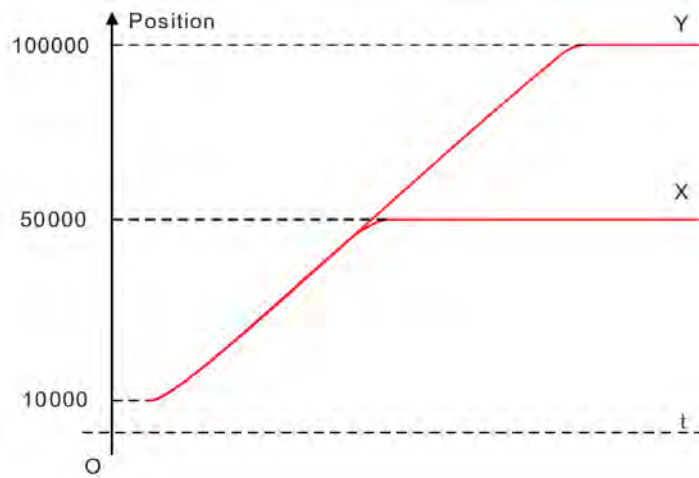
各軸以指定速度從現在位置運動到終點位置。最多可控制八個軸，運動過程中各軸是相互獨立的，運動軌跡如下所示。



- 格式：
N_G0 X_Y_Z_A_B_C_
- 參數說明：
N_：G 碼在程式中所在行的編號。
X_：指定 X 軸終點位置，單位：單元，資料型態：REAL。
Y_：指定 Y 軸終點位置，單位：單元，資料型態：REAL。
Z_：指定 Z 軸終點位置，單位：單元，資料型態：REAL。
A_：指定 A 軸終點位置，單位：單元，資料型態：REAL。
B_：指定 B 軸終點位置，單位：單元，資料型態：REAL。
C_：指定 C 軸終點位置，單位：單元，資料型態：REAL。
- 指令說明：
 1. G0 可以控制其中一個軸或多個軸，其他軸都為預設值。
 2. 各軸運動時的速度、加速度、減速度和加速度的變化率可以通過 DMC_SetG0Para 指令來設定這些參數。
- 範例：
X、Y 軸的初始位置都為 10000 單元，軸參數都為預設值。
將要執行的 G 碼如下：
N00 G90
N01 G0 X50000 Y100000
G 碼執行後，整個過程的 Y/X 曲線如下：



G 碼執行後，整個過程的位置/時間曲線如下：

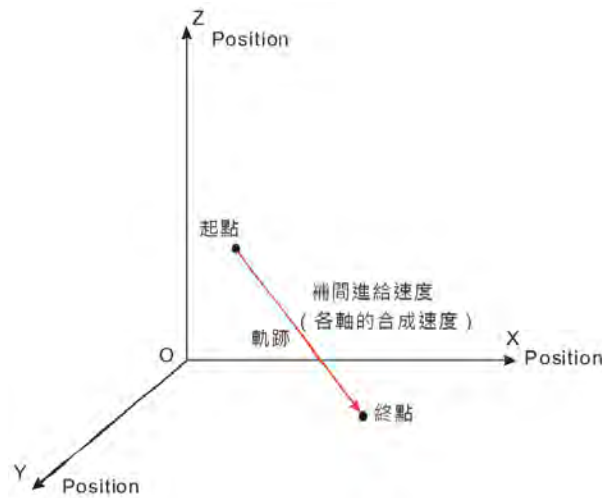


22

22.1.4.2 G1 (直線補間)

- 功能：

刀具以指定速度從某一點出發，直線移動到目標位置。此指令最多可控制八個軸，且各軸同起同停。如下圖所示，三個軸一起控制刀具的位置。



- 格式：
N_G1 X_Y_Z_A_B_C_F_
- 參數說明：
N_：G 碼在程式中所在行的編號。
X_：指定 X 軸終點位置，單位：單元，資料型態：REAL。
Y_：指定 Y 軸終點位置，單位：單元，資料型態：REAL。
Z_：指定 Z 軸終點位置，單位：單元，資料型態：REAL。
A_：指定 A 軸終點位置，單位：單元，資料型態：REAL。
B_：指定 B 軸終點位置，單位：單元，資料型態：REAL。
C_：指定 C 軸終點位置，單位：單元，資料型態：REAL。
F_：指定刀頭的進給速度，單位：單元/秒，資料型態：REAL。

刀頭勻速時，G 碼中所有軸的合成速度與 F 值相等。計算方法如下：

若存在兩個軸時，

$$F = \sqrt{V_1^2 + V_2^2}$$

若存在三個軸時，

$$F = \sqrt{V_1^2 + V_2^2 + V_3^2}$$

若存在更多軸時，按照上述方法類推。

- 指令說明：
 1. G0 可以控制其中一個軸或多個軸，其他軸都為預設值。
 2. 刀頭的速度、加速度、減速度和加速度的變化率可以通過 DMC_SetG1Para 指令來設定這些參數。

● 範例：

X、Y、Z 軸的初始位置都為 20000 單元，軸參數都為預設值。

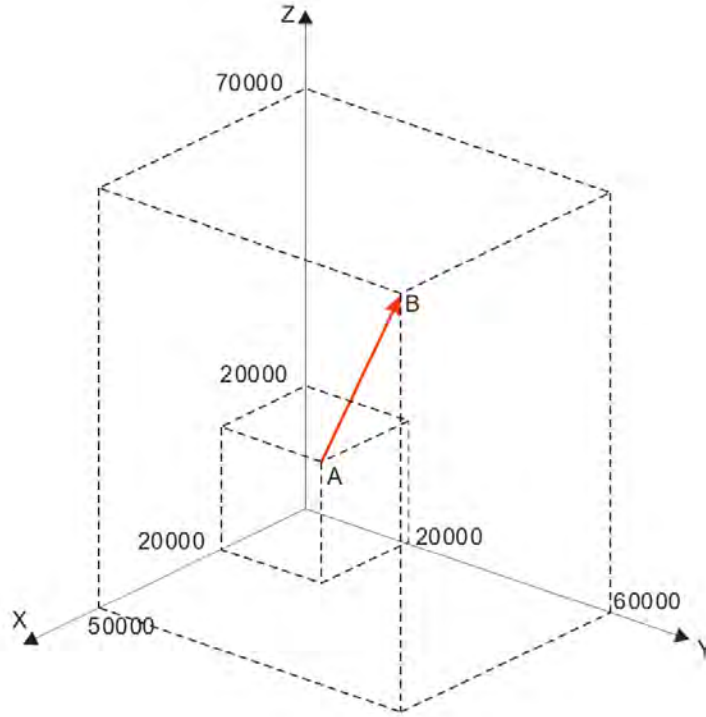
將要執行的 G 碼如下：

N00 G90

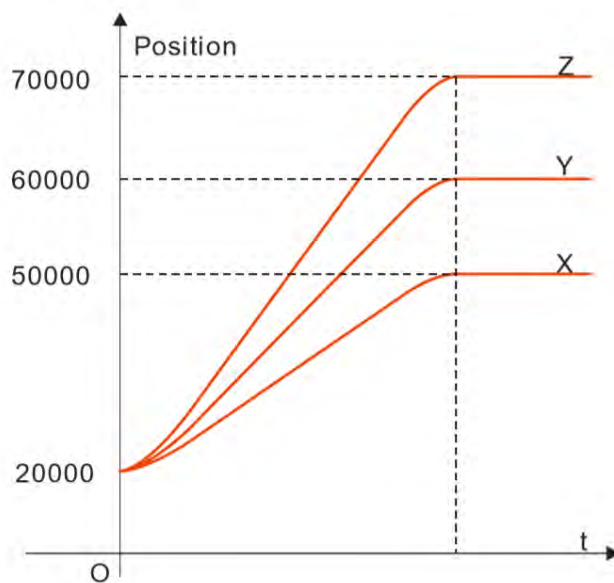
N01 G1 X50000 Y60000 Z70000

G 碼執行後，整個過程的曲線如下：

22



G 碼執行後，整個過程的位置/時間曲線如下：

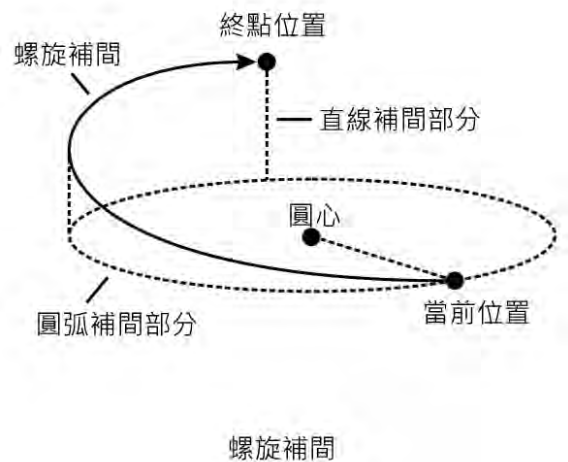
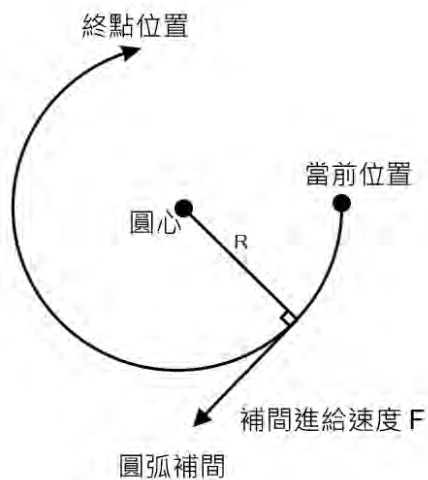


22.1.4.3 G2 (順時針圓弧補間/螺旋補間)

● 功能：

圓弧補間：刀具在指定平面及由圓心或半徑確定的圓弧上，以參數 F 設定的進給速度，對加工物件進行順時針方向圓弧切削。

螺旋補間：刀具在指定的平面及圓弧上順時針方向移動（圓弧補間）時，同時在指定平面的垂直方向上進行直線移動（直線補間），以參數 F 設定的進給速度。



22

● 格式：

格式 1：N_G2 X_Y_Z_I_J_F_

格式 2：N_G2 X_Y_Z_I_K_F_

格式 3：N_G2 X_Y_Z_J_K_F_

格式 4：N_G2 X_Y_Z_R_F_

● 參數說明：

N_：G 碼在程式中所在行的編號。

X_：指定 X 軸終點位置，單位：單元，資料型態：REAL。

Y_：指定 Y 軸終點位置，單位：單元，資料型態：REAL。

Z_：指定 Z 軸終點位置，單位：單元，資料型態：REAL。

I_J_：指定 XY 平面的圓心座標位置，單位：單元，資料型態：REAL。

I_K_：指定 XZ 平面的圓心座標位置，單位：單元，資料型態：REAL。

J_K_：指定 YZ 平面的圓心座標位置，單位：單元，資料型態：REAL。

R_：指定弧形半徑，資料型態：REAL。

F_：指定刀頭的進給速度，單位：單元/秒，資料型態：REAL。

刀頭勻速時，G 碼中所有軸的合成速度與 F 值相等。計算方法如下：

若存在兩個軸時，

$$F = \sqrt{V_1^2 + V_2^2}$$

若存在三個軸時，

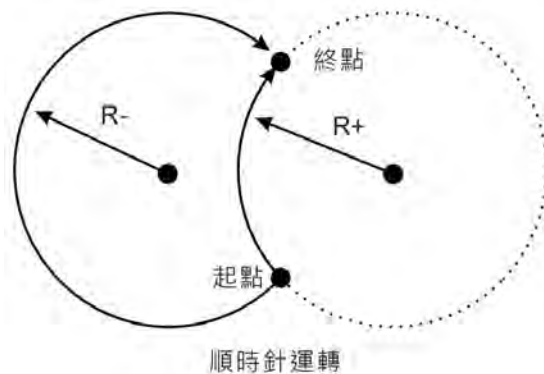
$$F = \sqrt{V_1^2 + V_2^2 + V_3^2}$$

若存在更多軸時，按照上述方法類推。

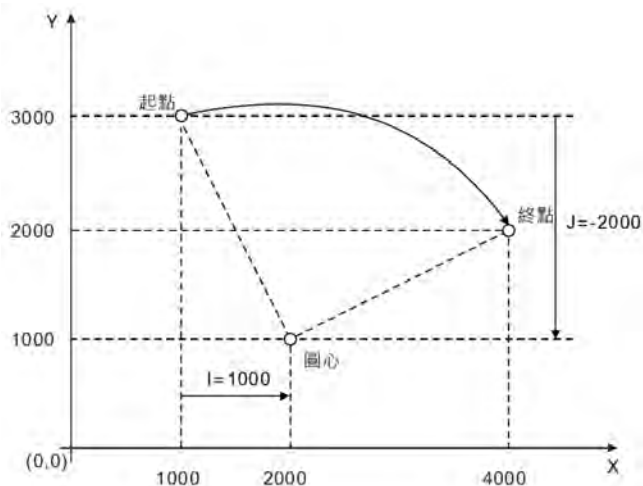
22

- 指令說明：

1. X、Y、Z 中的兩個軸在指定平面（平面由 G17/G18/G19 指令指定平面）上做圓弧補間運動，第 3 個軸垂直指定平面做直線補間運動。
2. F₁ 可以省略。在 G-Code 程式編寫區內若僅有一行指令，省略 F 參數後，刀頭的速度、加速度、減速度由軸組參數決定，可以通過 DMC_SetG1Para 指令來設定這些參數；若 G-Code 程式編寫區有多行指令且 G2 省略 F 參數，則刀頭的速度、加速度、減速度以刀頭之前所設定的 F 參數為準，若都沒有指定 F 參數，則以軸組參數中的速度、最大加速度、最大減速度為準。
3. 在 G90 絕對模式時，圓弧終點是以各自方向上的 0 單元為參考的絕對座標值。在 G91 相對模式時，圓弧終點是相對圓弧起點的增量值。
4. 不管是絕對還是相對模式，圓心座標 I₁J₁ (I₂K₁/J₂K₁) 始終是以起點為參考的相對座標。
5. 座標平面與 I、J、K 的關係，一條圓弧指令中最多出現 I、J、K 中的兩個，至於出現哪兩個則由相應的平面決定，如 XY 平面只能出現 I、J。
6. 格式 4 與格式 1、2、3 的不同在於，它通過起點、終點和半徑來確定一段圓弧。R 參數後面輸入正數時 (R+) 代表選擇劣弧（小於 180 度），輸入負數時 (R-) 代表選擇優弧（大於 180 度）。下圖所示實線部分為 G2 指令選擇 R+ 和 R- 時的運行軌跡，圓弧上的箭頭表示運轉方向：



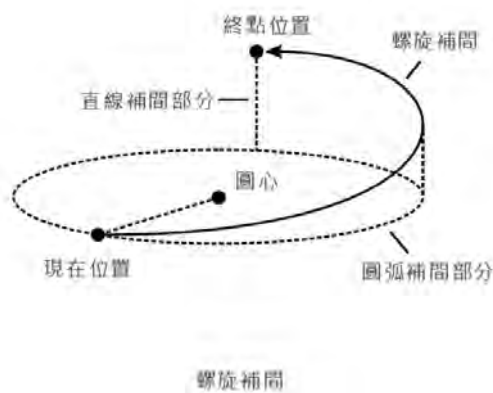
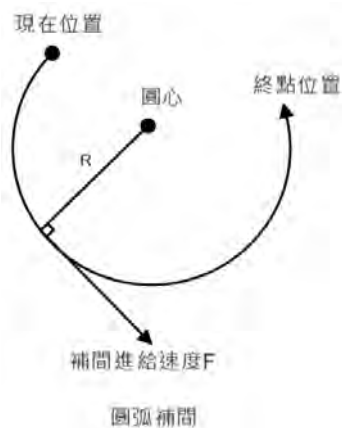
- 範例：
現在位置為 (1000 · 3000) · 軸參數都為預設值 ·
將要執行的 G 碼如下：
N00 G90
N01 G17
N02 G2 X4000 Y2000 I1000 J-2000 F5000
G 碼執行後 · 整個過程的 Y/X 曲線如下：



22

22.1.4.4 G3 (逆時針圓弧補間/螺旋補間)

- 功能：
圓弧補間：刀具在指定平面及由圓心或半徑確定的圓弧上，以參數 F 設定的進給速度，對加工物件進行逆時針方向圓弧切削。
螺旋補間：刀具在指定的平面及圓弧上逆時針方向移動（圓弧補間）時，同時在指定平面的垂直方向上進行直線移動（直線補間），以參數 F 設定的進給速度。



- 格式：

格式 1：N_G2 X_Y_Z_I_J_F_

格式 2：N_G2 X_Y_Z_I_K_F_

格式 3：N_G2 X_Y_Z_J_K_F_

格式 4：N_G2 X_Y_Z_R_F_

- 參數說明：

N_：G 碼在程式中所在行的編號。

X_：指定 X 軸終點位置，單位：單元，資料型態：REAL。

Y_：指定 Y 軸終點位置，單位：單元，資料型態：REAL。

Z_：指定 Z 軸終點位置，單位：單元，資料型態：REAL。

I_J_：指定 XY 平面的圓心座標位置，單位：單元，資料型態：REAL。

I_K_：指定 XZ 平面的圓心座標位置，單位：單元，資料型態：REAL。

J_K_：指定 YZ 平面的圓心座標位置，單位：單元，資料型態：REAL。

R_：指定弧形半徑，資料型態：REAL。

F_：指定刀頭的進給速度，單位：單元/秒，資料型態：REAL。

刀頭勻速時，G 碼中所有軸的合成速度與 F 值相等。計算方法如下：

若存在兩個軸時，

$$F = \sqrt{V_1^2 + V_2^2}$$

若存在三個軸時，

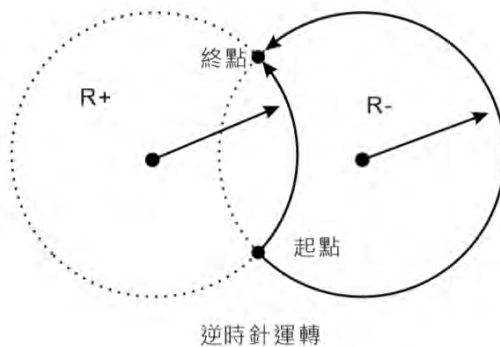
$$F = \sqrt{V_1^2 + V_2^2 + V_3^2}$$

若存在更多軸時，按照上述方法類推。

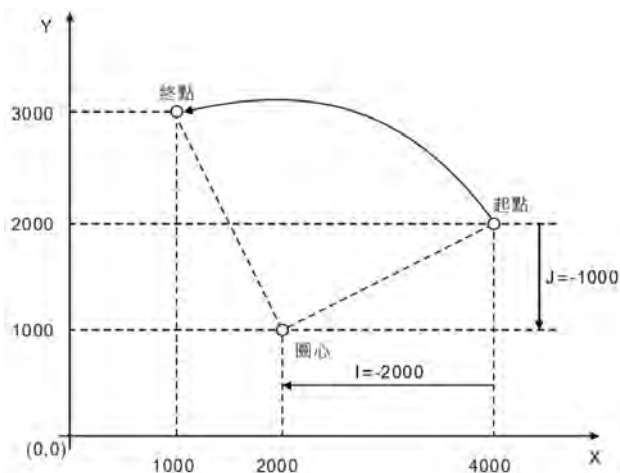
- 指令說明：

1. X、Y、Z 中的兩個軸在指定平面（平面由 G17/G18/G19 指令指定平面）上做圓弧補間運動，第 3 個軸垂直指定平面做直線補間運動。
2. F_可以省略。在 G-Code 程式編寫區內若僅有一行指令，省略 F 參數後，刀頭的速度、加速度、減速度由軸組參數決定，可以通過 DMC_SetG1Para 指令來設定這些參數；若 G-Code 程式編寫區有多行指令且 G3 省略 F 參數，則刀頭的速度、加速度、減速度以刀頭之前所設定的 F 參數為準，若都沒有指定 F 參數，則以軸組參數中的速度、最大加速度、最大減速度為準。
3. 在 G90 絕對模式時，圓弧終點是以各自方向上的 0 單元為參考的絕對座標值。在 G91 相對模式時，圓弧終點是相對圓弧起點的增量值。
4. 不管是絕對還是相對模式，圓心座標 I_J_ (I_K_/J_K_) 始終是以起點為參考的相對座標。

5. 座標平面與 I、J、K 的關係，一條圓弧指令中最多出現 I、J、K 中的兩個，至於出現哪兩個則由相應的平面決定，如 XY 平面只能出現 I、J。
6. 格式 4 與格式 1、2、3 的不同在於，它通過起點、終點和半徑來確定一段圓弧。R 參數後面輸入正數時 (R+) 代表選擇劣弧 (小於 180 度)，輸入負數時 (R-) 代表選擇優弧 (大於 180 度)。下圖所示實線部分為 G2 指令選擇 R+和 R-時的運行軌跡，圓弧上的箭頭表示運轉方向：



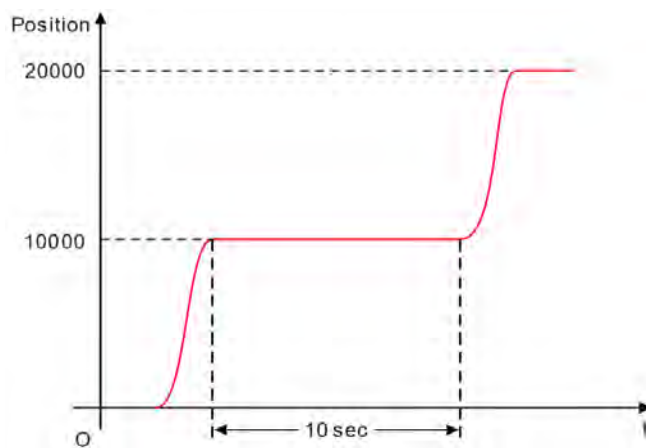
- 範例：
 現在位置為 (4000, 2000)，軸參數都為預設值。
 將要執行的 G 碼如下：
 N00 G90
 N01 G17 N02 G3 X1000 Y3000 I-2000 J-1000
 G 碼執行後，整個過程的 Y/X 曲線如下：



22.1.4.5 G4 (時間延遲)

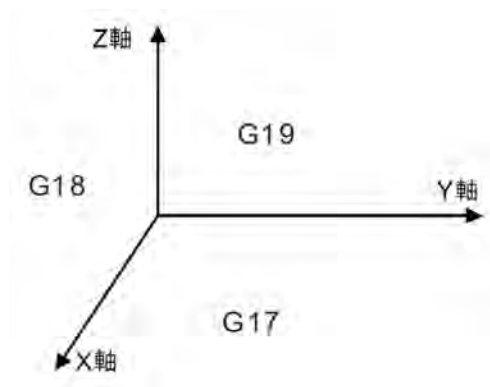
- 功能：
延遲指令。
- 格式：
格式 1：N_G4 X_
格式 2：N_G4 P_

● 參數說明：
N_：G 碼在程式中所在行的編號。
X_：指定延遲時間，單位：秒，資料型態：REAL。
P_：指定延遲時間，單位：毫秒，資料型態：REAL。
- 範例：
N00 G1 X10000
N01 G4 X10 (或 N01 G4 P10000)
N02 G1 X20000
G 碼執行後，整個過程的位置/時間曲線如下：



22.1.4.6 G17/G18/G19 (平面選擇)

- 功能：
此三個指令用於決定圓弧/螺旋補間平面選擇，對於直線補間無影響。
程式執行時，此三個工作平面可相互切換。程式中若無設定任何平面選擇，系統初始狀態為 XY 平面 (G17) 狀態。
- 格式：
N_G17
N_G18
N_G19
- 參數說明：
N_：G 碼在程式中所在行的編號。
- 平面示意圖如下圖所示：



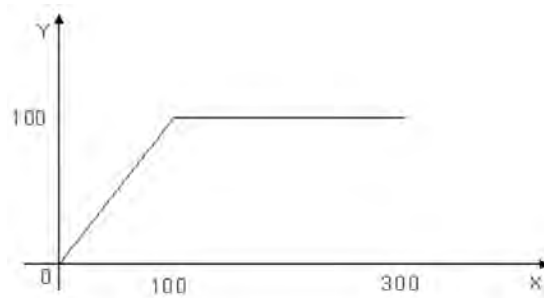
22.1.4.7 G50 (準確停止)

- 功能：
改變交接模式為準確停止，之後的交接模式都是準確停止，執行過程中可以使用 G51/G52 進行交接模式切換。每條 G 碼之間，終端機構的速度都會減為 0。

- 格式：
N_G50

- 參數說明：
N_：G 碼在程式中所在行的編號。

- 範例：
N00 G50
N01 G1 X100 Y100
N02 G1 X300 Y100
G 碼執行後，整個過程的 Y/X 曲線如下：



22

22.1.4.8 G51 (圓弧交接)

- 功能：

改變交接模式為圓弧交接，之後的交接模式都是圓弧交接，執行過程中可以使用 G50/G52 進行交接模式切換。每條 G 碼之間交接，終端機構的不減速，交接曲線為圓弧。在該模式下，整體運行速度以第一條 G 碼的速度為準，無法在運行中通過 F 分量變更速度。如果需要變更速度，可以通過 DMC_CartesianCoordinate 的 VelOverride 腳位變更速度超調量來控制終端機構的運行速度。

- 格式：

N_G51 D_

- 參數說明：

N_：G 碼在程式中所在行的編號。

D_：圓弧的半徑。

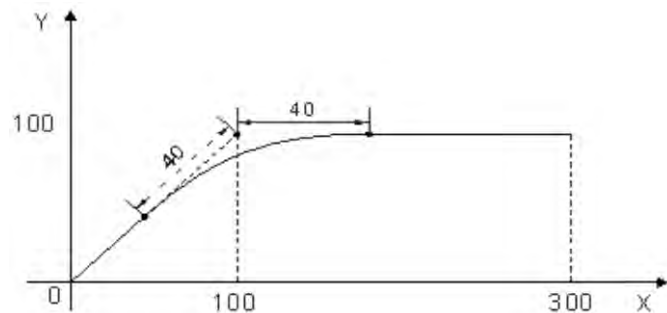
- 範例：

N00 G51 D40

N01 G1 X100 Y100

N02 G1 X300 Y100

G 碼執行後，整個過程的 Y/X 曲線如下：



22.1.4.9 G52 (圓滑交接)

- 功能：

改變交接模式為圓滑交接，之後的交接模式都是圓滑交接，執行過程中可以使用 G50/G51 進行交接模式切換。每條 G 碼之間交接，終端機構的不減速，適用於小線段連續補間。在該模式下，整體運行速度以第一條 G 碼的速度為準，無法在運行中通過 F 分量變更速度。如果需要變更速度，可以通過 DMC_CartesianCoordinate 的 VelOverride 腳位變更速度超調量來控制終端機構的運行速度。

- 格式：

N_G52

- 參數說明：

N_：G 碼在程式中所在行的編號。

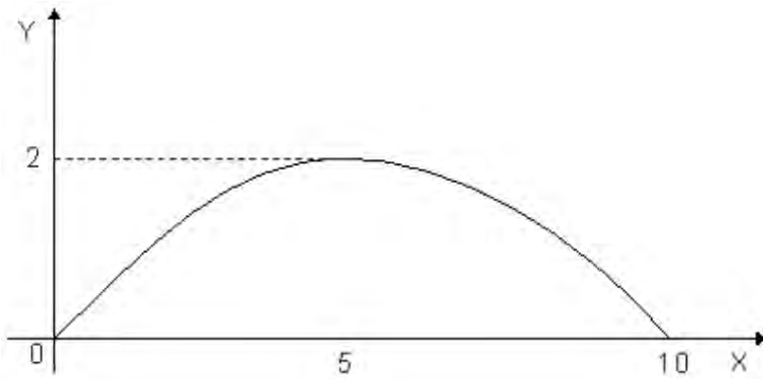
- 範例：

N00 G52	N01 G1 X0 Y0 E5 E-5 F5
N02 G1 X0.1 Y0.06282151816	N03 G1 X0.2 Y0.1255810391
N04 G1 X0.3 Y0.1882166266	N05 G1 X0.4 Y0.2506664671
N06 G1 X0.5 Y0.3128689301	N07 G1 X0.6 Y0.3747626292
N08 G1 X0.7 Y0.4362864828	N09 G1 X0.8 Y0.4973797743
N10 G1 X0.9 Y0.5579822121	N11 G1 X1 Y0.6180339887
N12 G1 X1.1 Y0.6774758405	N13 G1 X1.2 Y0.7362491054
N14 G1 X1.3 Y0.7942957813	N15 G1 X1.4 Y0.8515585831
N16 G1 X1.5 Y0.9079809995	N17 G1 X1.6 Y0.9635073482
N18 G1 X1.7 Y1.018082832	N19 G1 X1.8 Y1.07165359
N20 G1 X1.9 Y1.124166756	N21 G1 X2 Y1.175570505
N22 G1 X2.1 Y1.225814107	N23 G1 X2.2 Y1.274847979
N24 G1 X2.3 Y1.322623731	N25 G1 X2.4 Y1.369094212
N26 G1 X2.5 Y1.414213562	N27 G1 X2.6 Y1.457937255
N28 G1 X2.7 Y1.500222139	N29 G1 X2.8 Y1.541026486
N30 G1 X2.9 Y1.580310025	N31 G1 X3 Y1.618033989
N32 G1 X3.1 Y1.654161149	N33 G1 X3.2 Y1.688655851

N34 G1 X3.3 Y1.721484054	N35 G1 X3.4 Y1.75261336
N36 G1 X3.5 Y1.782013048	N37 G1 X3.6 Y1.809654105
N38 G1 X3.7 Y1.835509251	N39 G1 X3.8 Y1.859552972
N40 G1 X3.9 Y1.881761538	N41 G1 X4 Y1.902113033
N42 G1 X4.1 Y1.920587371	N43 G1 X4.2 Y1.937166322
N44 G1 X4.3 Y1.951833524	N45 G1 X4.4 Y1.964574501
N46 G1 X4.5 Y1.975376681	N47 G1 X4.6 Y1.984229403
N48 G1 X4.7 Y1.991123929	N49 G1 X4.8 Y1.996053457
N50 G1 X4.9 Y1.999013121	N51 G1 X5 Y2
N52 G1 X5.1 Y1.999013121	N53 G1 X5.2 Y1.996053457
N54 G1 X5.3 Y1.991123929	N55 G1 X5.4 Y1.984229403
N56 G1 X5.5 Y1.975376681	N57 G1 X5.6 Y1.964574501
N58 G1 X5.7 Y1.951833524	N59 G1 X5.8 Y1.937166322
N60 G1 X5.9 Y1.920587371	N61 G1 X6 Y1.902113033
N62 G1 X6.1 Y1.881761538	N63 G1 X6.2 Y1.859552972
N64 G1 X6.3 Y1.835509251	N65 G1 X6.4 Y1.809654105
N66 G1 X6.5 Y1.782013048	N67 G1 X6.6 Y1.75261336
N68 G1 X6.7 Y1.721484054	N69 G1 X6.8 Y1.688655851
N70 G1 X6.9 Y1.654161149	N71 G1 X7 Y1.618033989
N72 G1 X7.1 Y1.580310025	N73 G1 X7.2 Y1.541026486
N74 G1 X7.3 Y1.500222139	N75 G1 X7.4 Y1.457937255
N76 G1 X7.5 Y1.414213562	N77 G1 X7.6 Y1.369094212
N78 G1 X7.7 Y1.322623731	N79 G1 X7.8 Y1.274847979
N80 G1 X7.9 Y1.225814107	N81 G1 X8 Y1.175570505
N82 G1 X8.1 Y1.124166756	N83 G1 X8.2 Y1.07165359
N84 G1 X8.3 Y1.018082832	N85 G1 X8.4 Y0.9635073482
N86 G1 X8.5 Y0.9079809995	N87 G1 X8.6 Y0.8515585831
N88 G1 X8.7 Y0.7942957813	N89 G1 X8.8 Y0.7362491054
N90 G1 X8.9 Y0.6774758405	N91 G1 X9 Y0.6180339887
N92 G1 X9.1 Y0.5579822121	N93 G1 X9.2 Y0.4973797743
N94 G1 X9.3 Y0.4362864828	N95 G1 X9.4 Y0.3747626292

N96 G1 X9.5 Y0.3128689301	N97 G1 X9.6 Y0.2506664671
N98 G1 X9.7 Y0.1882166266	N99 G1 X9.8 Y0.1255810391
N100 G1 X9.9 Y0.06282151816	N101 G1 X10 Y0

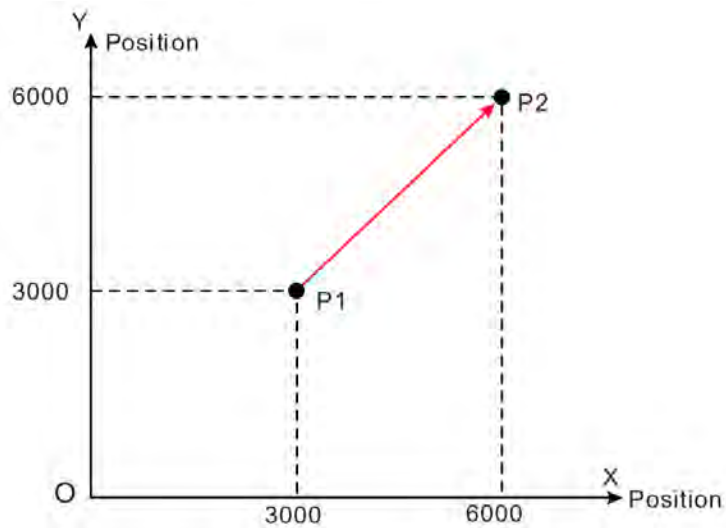
G 碼執行後，整個過程的 Y/X 曲線如下：



22.1.4.10 G90 (絕對模式)

- 功能：
G90 執行後，其後 G 碼中各軸的終點位置值均以 0 單元為基準，中途可以用 G91 指令切換為相對模式。程式預設為絕對模式。
- 格式：
N_G90
- 參數說明：
N_：G 碼在程式中所在行的編號。

- 範例：
X、Y 軸的初始位置都為 3000 單元，軸參數都為預設值。
將要執行的 G 碼如下：
N00 G90
N01 G0 X6000 Y6000
G 碼執行後，整個過程的 Y/X 曲線如下：



22

22.1.4.11 G91 (相對模式)

- 功能：
G91 執行後，其後 G 碼中各軸的終點位置均以從現在位置開始的增量值計算，中途可以用 G90 指令切換為相對模式。程式預設為絕對模式。
- 格式：
N_G91
- 參數說明：
N_：G 碼在程式中所在行的編號。

- 範例：

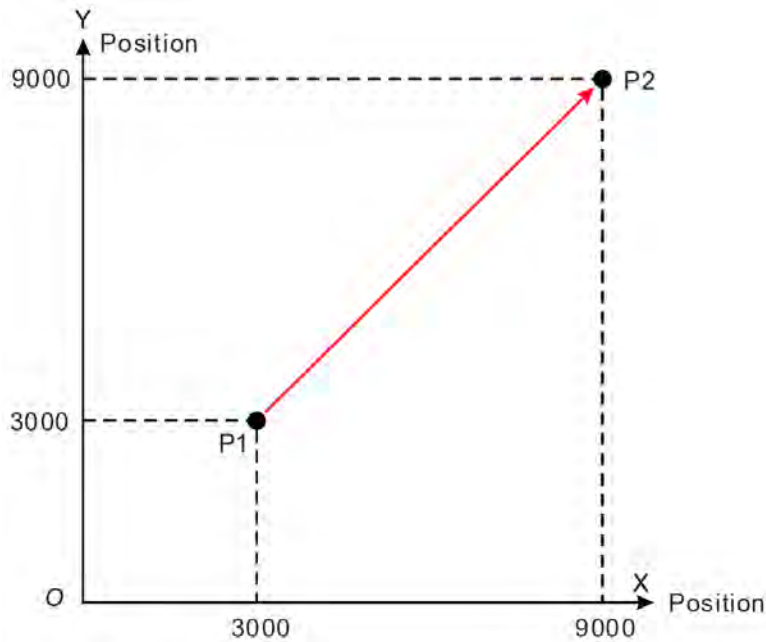
X、Y 軸的初始位置都為 3000 單元，軸參數都為預設值。

將要執行的 G 碼如下：

```
N00 G91
```

```
N01 G0 X6000 Y6000
```

G 碼執行後，整個過程的 Y/X 曲線如下：



22.1.4.12 M 碼

- 功能：

與一般程式進行交互使用。

- 格式：

```
N_M_D_
```

- 指令說明：

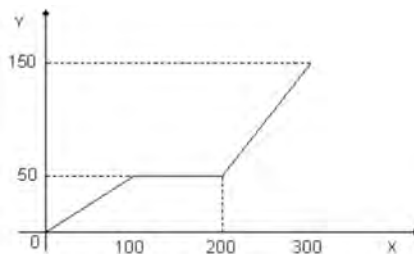
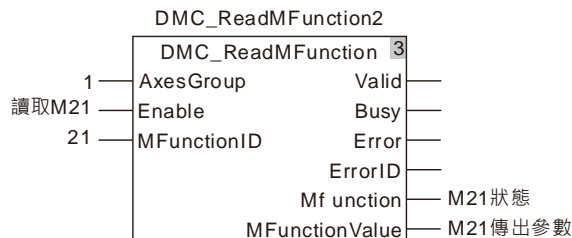
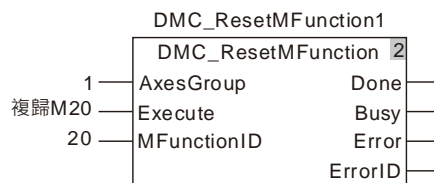
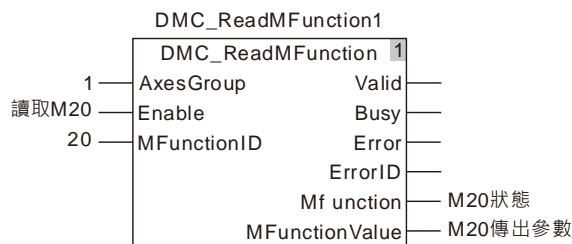
1. D 分量可以被省略，執行 M 碼時就不會傳出參數。
2. M 碼有兩種使用方式：一種是和 G 碼不寫在同一行，一種是和 G 碼寫在同一行。
3. 當 M 碼和 G 碼不在同一行，例如：N0 M10 D10.02；N1 G1 X100 Y100。當執行到 N0 這一行時，G 碼執行會停下，與此同時，使用 DMC_ReadMFunction 指令讀取 M 碼狀態，輸出位元 MFunction 置為 TRUE，並且 MFunctionValue 的值為 10.02。當使用 DMC_ResetMFunction 指令重定 M 碼後，G 碼才會繼續向下執行 N1。

4. 當 M 碼和 G 碼在同一行，並且只能放在 G 碼之後。例如：N0 G1 X100 M10 D10.02。當執行到這一行時，會執行 G1，同時使用 DMC_ReadMFunction 指令讀取 M 碼狀態，輸出位元 MFuntion 置為 TRUE，並且 MFunctionValue 的值为 10.02。G1 執行完成後則會繼續向下執行，不需要使用指令重定 M 碼。

● 範例：

```
N0 G1 X100 Y50
N1 M20
N2 G1 X200 Y50 M21 D3.14
N3 G1 X300 Y150
```

G 碼執行後，整個過程的 Y/X 曲線如下：

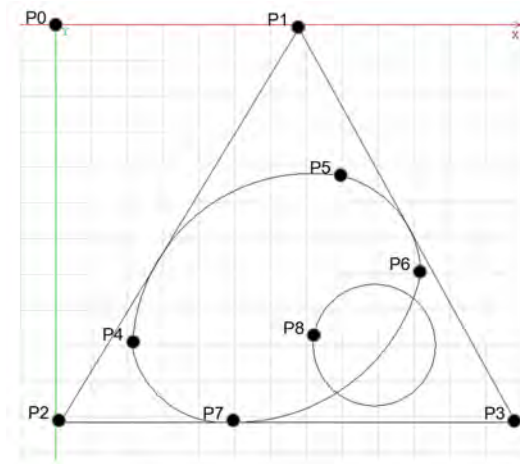


當兩個變數“讀取 M20”和“讀取 M21”為 TRUE 時，開始執行 G 碼，終端機構運行到 (100 · 50) 的位置停下，此時變數“M20 狀態”為 TRUE，等待其它動作執行完成後，再將變數“重定 M20”為 TRUE，復位 M 代碼。此時 G 碼繼續運行，終端機構開始運行到 (200 · 50)，同時變數“M21 狀態”為 TRUE，變數“M21 傳出參數”。

22.1.5 G-Code 範例-三軸同動繪出 Delta LOGO

使用 G-Code 與絕對定位來編寫繪製軌跡，繪出 Delta Logo。

【欲繪製圖形】



【G-Code 程式碼】

```

0001 N0 G90                                (P0)
0002 N1 G0 Z5000
0003 N2 G0 X32500 Y-500                    (P1)
0004 N3 G0 Z0
0005 N4 G1 X600 Y-53400 F10000            (P2)
0006 N5 G1 X61500 Y-53400 F10000         (P3)
0007 N6 G1 X32500 Y-500 f10000           (P4)
0008 N7 G0 Z5000
0009 N8 G0 X23100 Y-53400                 (P5)
0010 N9 G0 Z0
0011 N10 G2 X10300 Y-43600 R12600         (P6)
0012 N11 G2 X38440 Y-20500 R23900        (P7)
0013 N12 G2 X48880 Y-33300 R13900       (P8)
0014 N13 G2 X23100 Y-53400 R25600       (P5)
0015 N14 G0 Z5000
0016 N15 G0 X34500 Y-43000                (P9)
0017 N16 G0 z0
0018 N17 G2 X34500 Y-43000 I8150 J0 F5000 (P9)
0019 N18 G0 z5000
0020 N19 G0 x0 y0                          (P0)
0021

```

【路徑說明】

程式運行步驟：

步驟一：Z 軸提筆上移後，從原點 P0 移動到達 P1。

步驟二：P1 處 Z 軸下筆，從 P1 移動到達 P2，P2 移動到達 P3，P3 移動到達 P1，第三軸提筆，完成三角形。

步驟三：從 P1 移動到達 P4，P4 處 Z 軸下筆，從 P4 移動到達 P5，P5 移動到達 P6，P6 移動到達 P7，P7 移動到達 P4，Z 軸提筆，完成橢圓形。

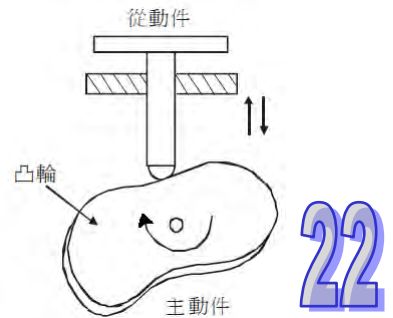
步驟四：從 P4 移動到達 P8，P8 處 Z 軸下筆，從 P8 直接畫一個圓至 P8，Z 軸提筆，完成圓形。

步驟五：Z 軸移動至原點 P0 後結束，DELTA LOGO 完成。

22.2 E-CAM 編輯器

22.2.1 關於 E-CAM (電子凸輪)

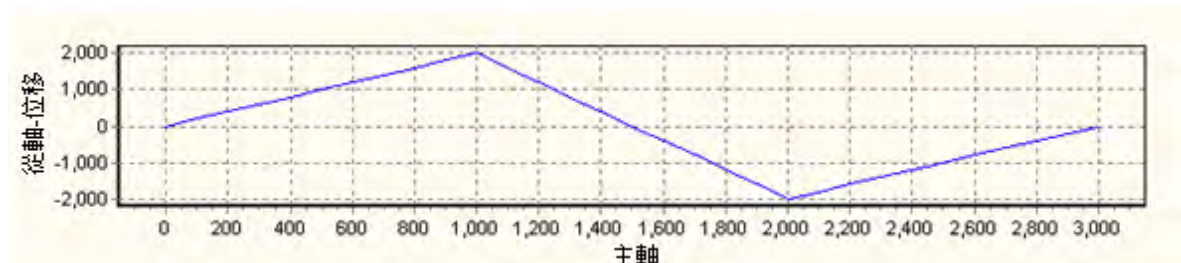
凸輪是一種不規則形狀的輸入件，可以經由直接接觸傳遞運動到從動件，使從動件依設定的規律運動。機械凸輪由凸輪、從動件以及機架等三個部份組合而成。如圖，當凸輪旋轉時，從動件會依照凸輪的外型，作上下的運動。當進行運動控制時，軸與軸之間常需要進行互相關連性的運動，此種運動即需要凸輪來達成。



電子凸輪 (E-CAM) 以建立電子凸輪表 (CAM Chart) 的方式，定義主動件與從動件的關係後，由控制器模擬凸輪的運動，使從軸依此關係對應主軸運動。電子凸輪的優點是，只要修改電子凸輪表，而不需要變更機構，即可完成凸輪的修改，且無機構磨耗的問題。

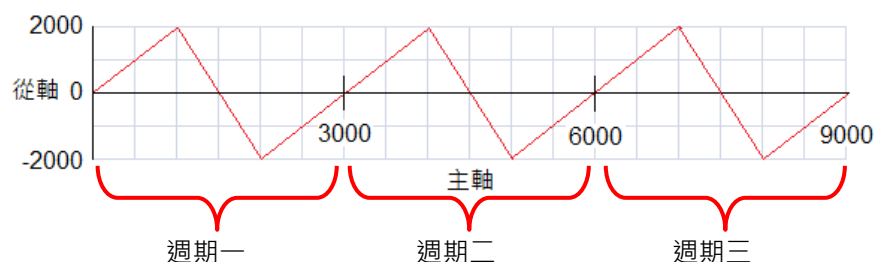
22.2.2 電子凸輪意義說明

一個完成建立的 CAM 圖表如下圖，CAM 圖表的橫軸代表主軸，縱軸代表從軸，CAM 圖表上的數值皆為單位數 (例如脈波數)，橫軸的 0~3000 個單位為一個電子凸輪週期，縱軸則為相對應的從軸輸出單位數。



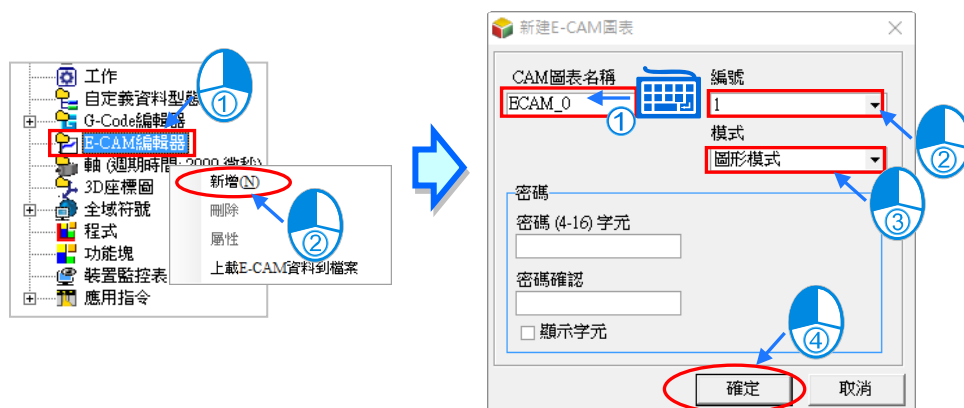
當這個 CAM 圖表被作為週期性的使用時，主軸與從軸的輸出關係如下圖。可以知道電子凸輪將會被運動主機不斷重複的使用，而當主軸完成一個電子凸輪週期時 (主軸送出 3000 個單位)，主軸的單位數會繼續累加，而非回到初始值；從軸在完成一個電子凸輪週期後，則會依據電子凸輪設定的行程初始值再度重複行程的變化。

若以連接伺服馬達的觀點而言，主軸伺服可視為不斷往同方向旋轉的傳動軸，從軸伺服則可視為依照電子凸輪關係，不斷做週期性往復運動的加工軸，這個特性可用於追剪與飛剪等應用。



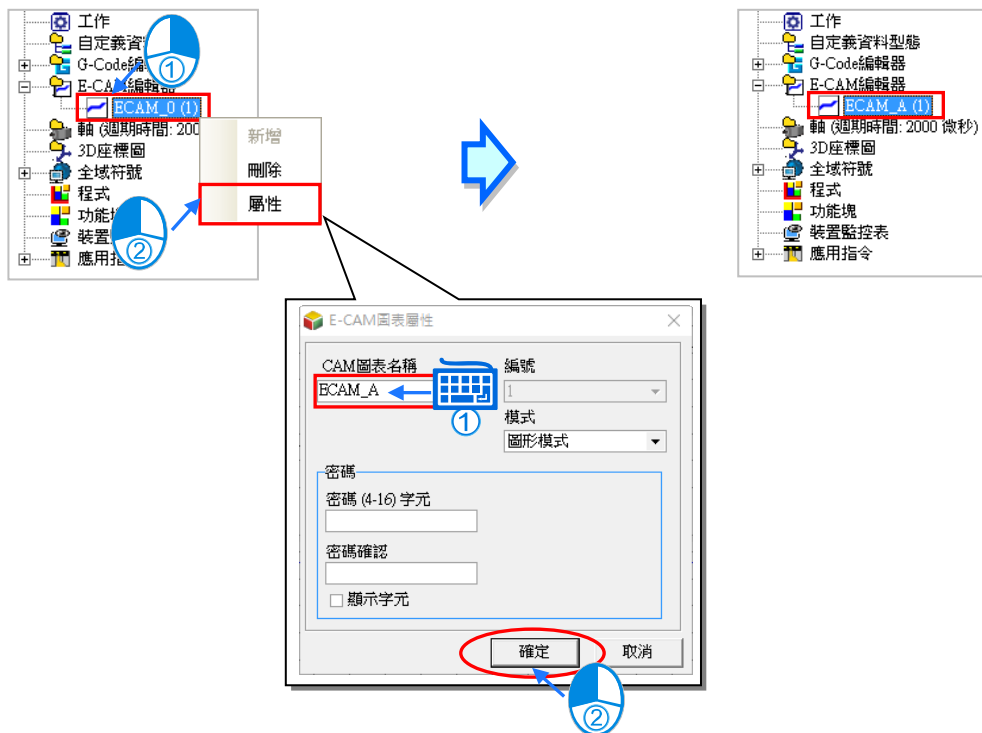
22.2.3 使用 E-CAM 編輯器

於專案管理區的 E-CAM 編輯器項目點擊 **右鍵** > **新增**，出現新增視窗如下圖，於 E-CAM 圖表名稱的欄位輸入名稱，再於編號點擊滑鼠左鍵，於下拉選單選擇編號，接著選擇所需的模式。也可以設定密碼來保護 E-CAM 設計資料，設定好密碼後，若為「**圖形模式**」，則下次開啟編輯視窗時將會詢問密碼；若為「**檔案模式**」，則在匯入匯出時會詢問密碼。

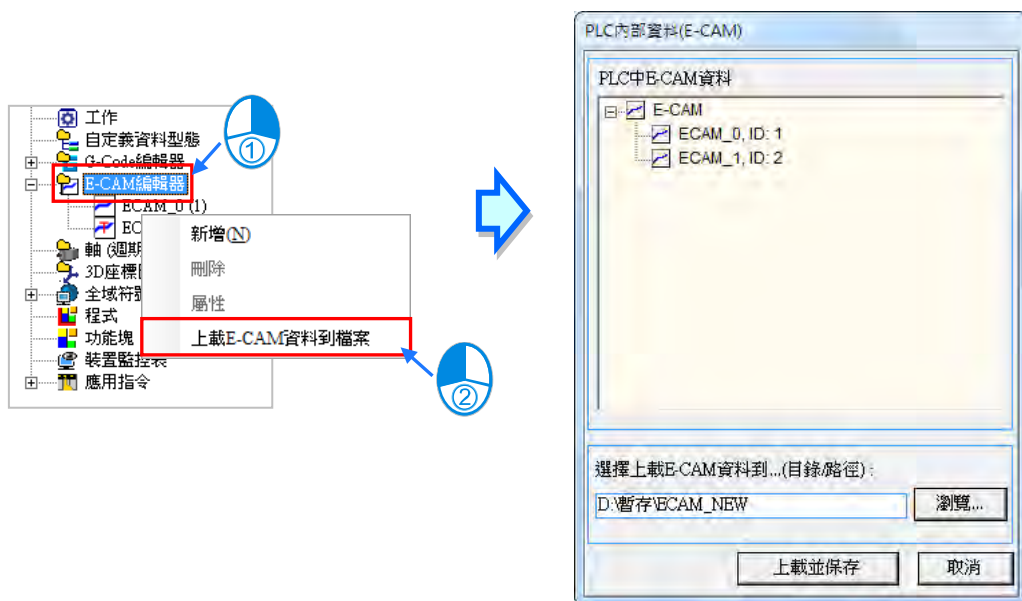


名稱	功能說明	AS	AHxxEMC	DVPxxMC
編號	E-CAM 表的編號	最大 16	最大 32	最大 32
ASDA 檔案模式	ASDA 檔案模式的 E-CAM 表	V		
檔案模式	檔案模式的 E-CAM 表		V	
圖形模式	圖形模式的 E-CAM 表		V	V
週期旗標	當拖拉第一個與最後一個控制點，兩點的從軸資料會即時同步			V
密碼	E-CAM 表的密碼		V	V
上載 E-CAM 到檔案	從 PLC 上載主軸、從軸位置		V	

在已建立的 E-CAM 項目上點擊滑鼠右鍵，可於快捷選單點選「刪除」，刪除此項目；或是點選「屬性」，重新設定項目名稱或密碼，但不可變更編號。



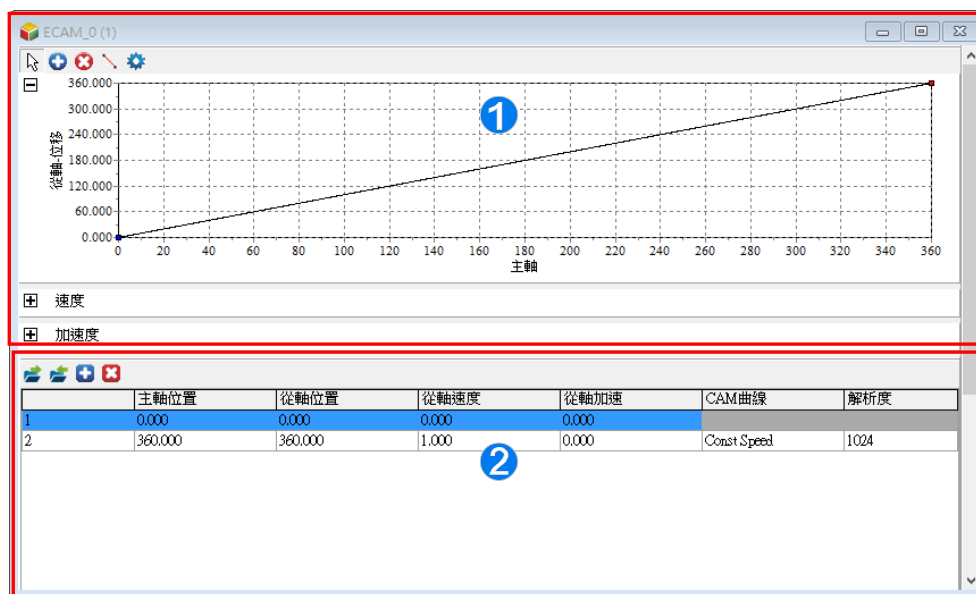
使用台達函式庫內 AH-Motion 的功能塊，例如：DFB_CamWire，線上修改過 E-CAM 圖表的點位，接著可以利用「上載 E-CAM 資料到檔案」的功能，將 E-CAM 表的主軸位置及從軸位置另存成檔案。使用此功能，需先與 AHxxEMC 主機連線，再於專案管理區的 E-CAM 編輯器項目點擊 右鍵 > 上載 E-CAM 資料到檔案，此時，會出現 PLC 內部資料 (E-CAM) 視窗。



在選擇完存檔路徑後，點擊「**上載並保存**」，這時儲存路徑目錄底下會出現未被功能塊修改過的檔案 xxxx_srcpdata.txt 及被功能塊修改過的檔案 xxxx_uploadpdata.txt 兩個檔案，xxxx 為 E-CAM 的項目名稱，如下圖。



若選擇「**圖形模式**」，完成新建 E-CAM 圖表後將會開啟 E-CAM 編輯視窗，功能說明如下：






- 關係圖區：依據資料設定區設計資料，即時顯示對應的 E-CAM 位置、速度、加速度關係圖。可以使用滑鼠左鍵拖曳移動圖中的關鍵點改變設計資料。也可使用功能按鈕新增關鍵點或線段。
- 規劃表區：使用功能按鈕新增或刪除表格，在表格中輸入 E-CAM 設計資料。資料將與關係圖同步。

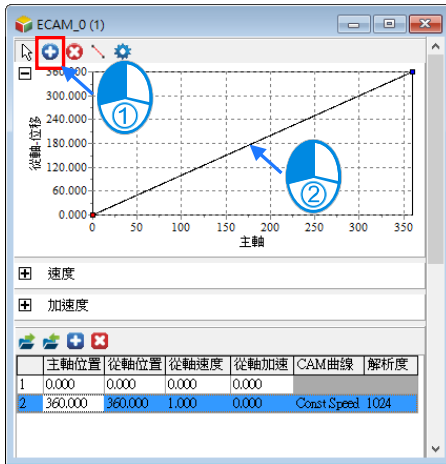
E-CAM 編輯器的功能按鈕說明如下

圖示	名稱	功能說明	AS	AHxxEMC	DVPxxMC
	選取工具	切換為選取工具	V	V	V
	新增關鍵點	在關係圖中新增一個關鍵點	V	V	V
	刪除關鍵點	在關係圖中刪除一個關鍵點	V	V	V
	新增線段	在關係圖中新增包含兩關鍵點的線段		V	V
	設定	開啟設定視窗	V	V	V
	匯出	匯出對應格式的 E-CAM	.csv	.dst	.dst
	匯入	匯入對應格式的 E-CAM	.csv	.dst	.dst
	上載從軸資料	從檔案暫存器/伺服上載資料回來	V		
	下載從軸資料	下載資料至檔案暫存器/伺服	V		
	新增一行	在規劃表區新增一筆關鍵點資料	V	V	V
	刪除一行	在規劃表區刪除一筆關鍵點資料	V	V	V
	新增挺桿點	在曲線上新增挺桿點的圖示 			V
	顯示 CAM 表	顯示 CAM 表			V
	顯示挺桿點表	顯示挺桿點表			V

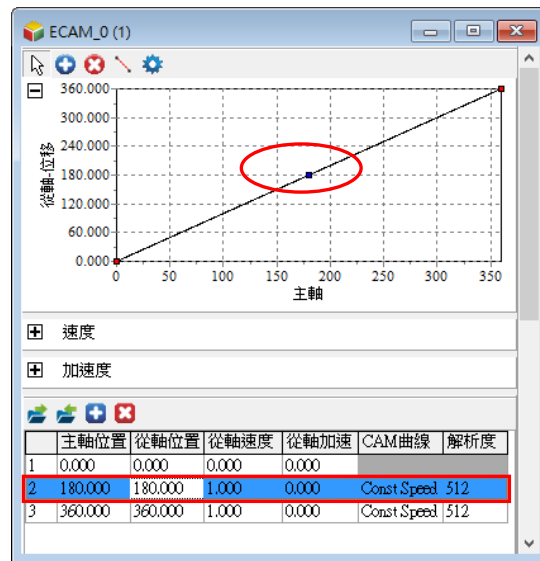
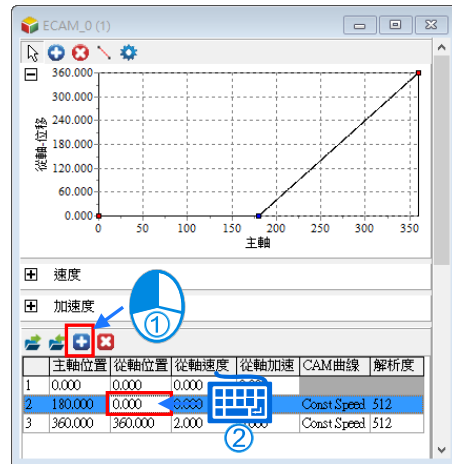
22




要新增點，先點選「新增關鍵點」 的按鈕後，在關係圖中要新增的位置，點擊滑鼠左鍵，便會建立一個關鍵點，此時下方的規劃表區也會自動增加此關鍵點的資料，或是在規劃表區點選，在於此關鍵點資料填入從軸位置 180 如下圖。若要連續新增多個關鍵點，在點選 按鈕後可以在關係圖中多次點擊滑鼠左鍵。

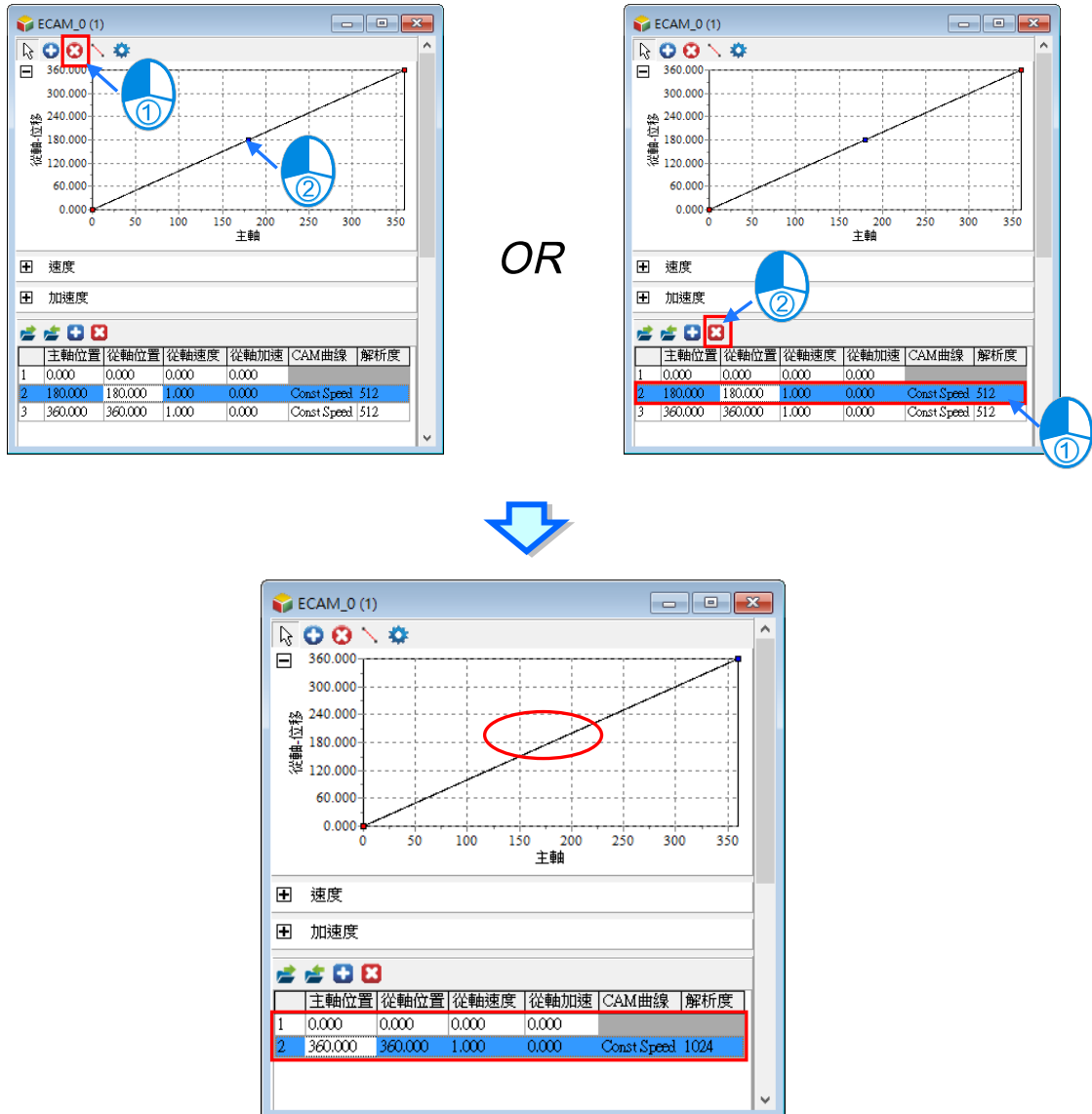
22



OR



要刪除點，在點選「刪除關鍵點」 按鈕後，在關係圖中點選要刪除的關鍵點，此時規劃表區也會自動刪除此關鍵點的資料，或於規劃表區選取要刪除的關鍵點資料，接著點選 即可刪除，如下圖。若要連續點刪除多個關鍵點，點選 按鈕後直接連續點選要刪除的點即可。




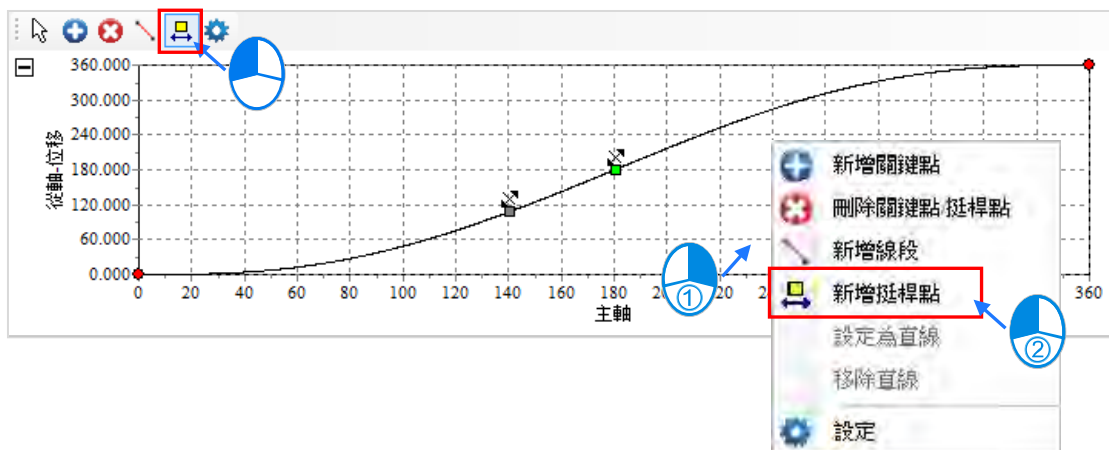
在電子凸輪運用過程中，可能需要在凸輪運行到某些特定位置時觸發相應的動作，這時就需要在凸輪的指定位置設定一個狀態開關，當凸輪運行到該位置時，狀態開關的狀態發生變化，在程式中可通過狀態開關的狀態控制相應的動作，這個狀態開關稱之為凸輪挺桿。

凸輪挺桿可以設定在凸輪的任何位置，且一個凸輪最多可設定 128 個凸輪挺桿。凸輪正向或反向經過時，挺桿的狀態可分別進行配置（關閉、置位、復位、取反）。在控制器程式中可通過凸輪挺桿相應指令獲取挺桿的狀態。凸輪挺桿狀態說明如下：

關閉	凸輪經過該挺桿對應的位置時，該挺桿狀態不變化。
置位	凸輪經過該挺桿對應的位置時，該挺桿狀態被設定為 TRUE。
復位	凸輪經過該挺桿對應的位置時，該挺桿狀態被設定為 FALSE。
取反	凸輪經過該挺桿對應的位置時，該挺桿狀態在 TRUE 和 FALSE 之間切換一次（如果為 TRUE 則變為 FALSE，如果為 FALSE 則變為 TRUE）。

注意：凸輪挺桿的狀態在下一個凸輪週期開始時會被復位。


新增凸輪挺桿可以透過工具列上的新增挺桿點圖示，或是在關係圖中點擊右鍵，接著於選單上選取新增挺桿點，如下圖。



在規劃表中每行關鍵點資料提供使用者設定一段凸輪曲線，再由數行關鍵點資料組成整個凸輪曲，所有已建立的關鍵點資料即構成一個電子凸輪週期，使用者於每筆關鍵點資料輸入各的主軸與從軸關係，說明如下。

名稱	功能說明	AS	AHxxEMC	DVPxxMC
關鍵點	紀錄 E-CAM 運動的座標	最大 721	最大 40	最大 40
CAM 曲線 Const Speed	等速度		V	
CAM 曲線 Const Acc.	等加速度		V	


名稱	功能說明	AS	AHxxEMC	DVPxxMC
CAM 曲線 Single Hypot.	單精度三角斜邊		V	
CAM 曲線 Cycloid	擺線		V	
CAM 曲線 Polynomial	多項式	V	V	V
CAM 曲線 B-Spline	B-樣條		V	
解析度	目前區段所使用的資料點數	自動配置	最大 2047	自動配置

點選「設定」 按鈕，開啟其他設定視窗，說明如下。

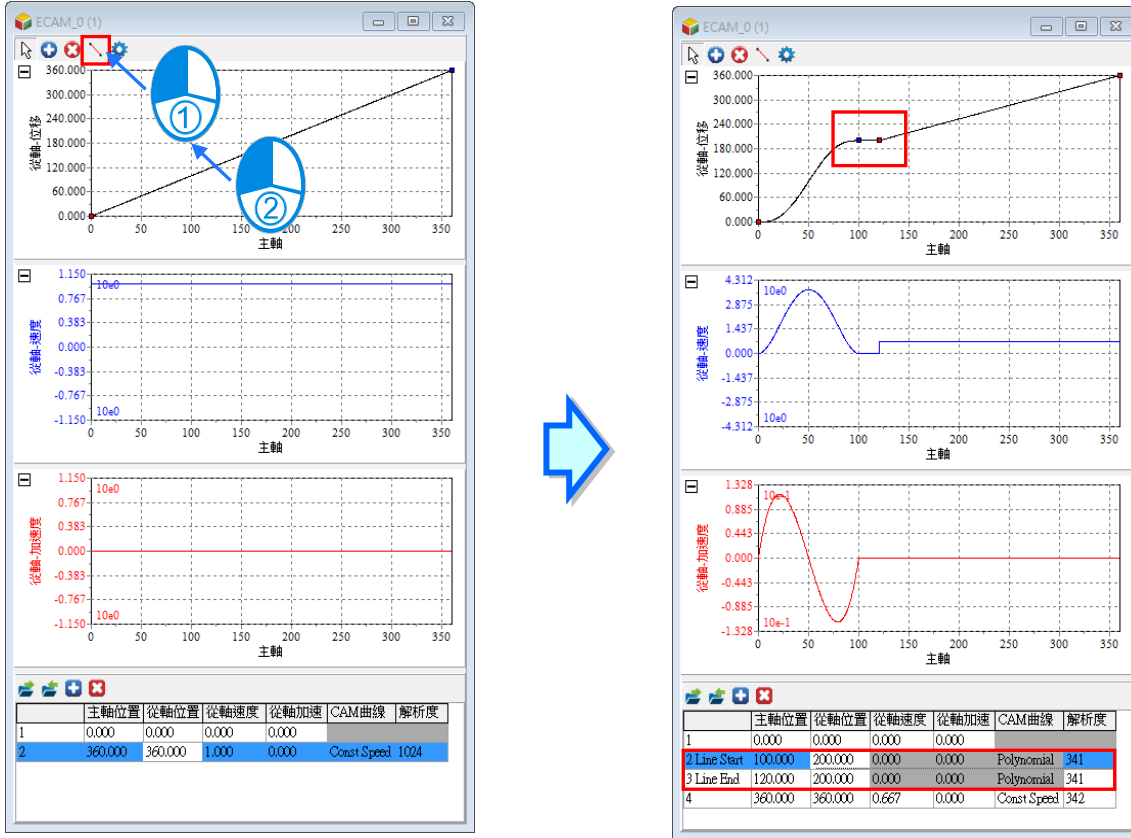
22



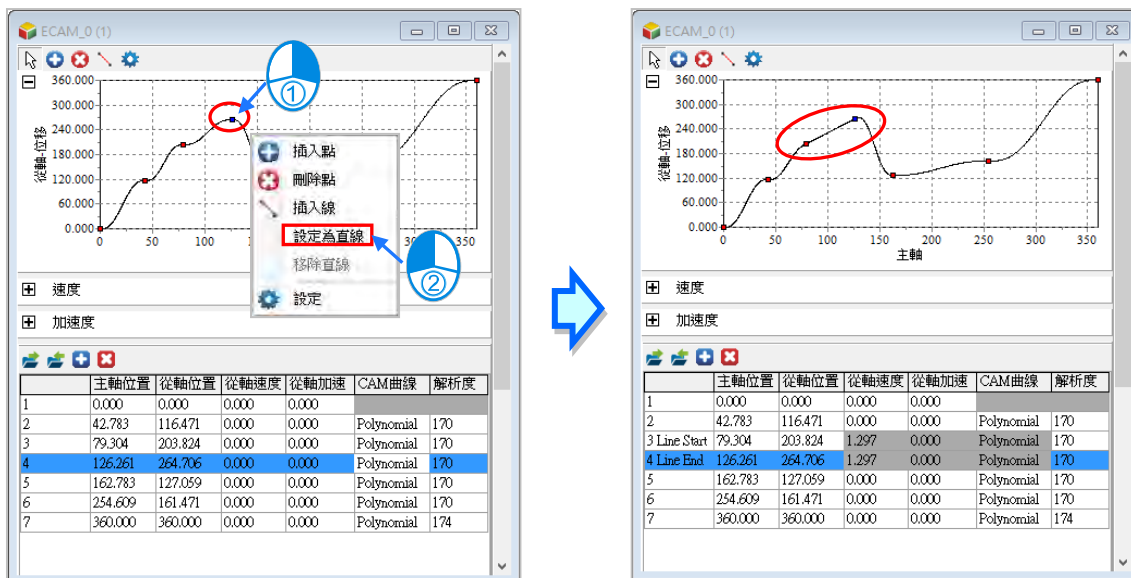
- ❶ 主軸位置：設定關係圖中主軸顯示範圍。
- ❷ 從軸位置：設定關係圖中從軸顯示範圍。
- ❸ 解析度：設定此 E-CAM 表，曲線區間的解析度總和。

要新增直線屬性的線段，於點選「新增線段」的按鈕後，在關係圖中要新增的位置點選滑鼠左鍵，此時下方的規劃表區也會自動增加此線段的兩個關鍵點資料。當插入線段後，插入線段的 CAM 曲線函數會被固定在「Polynomial」的模式，左方線段編號右側則會顯示「Line Start」及「Line End」。

22



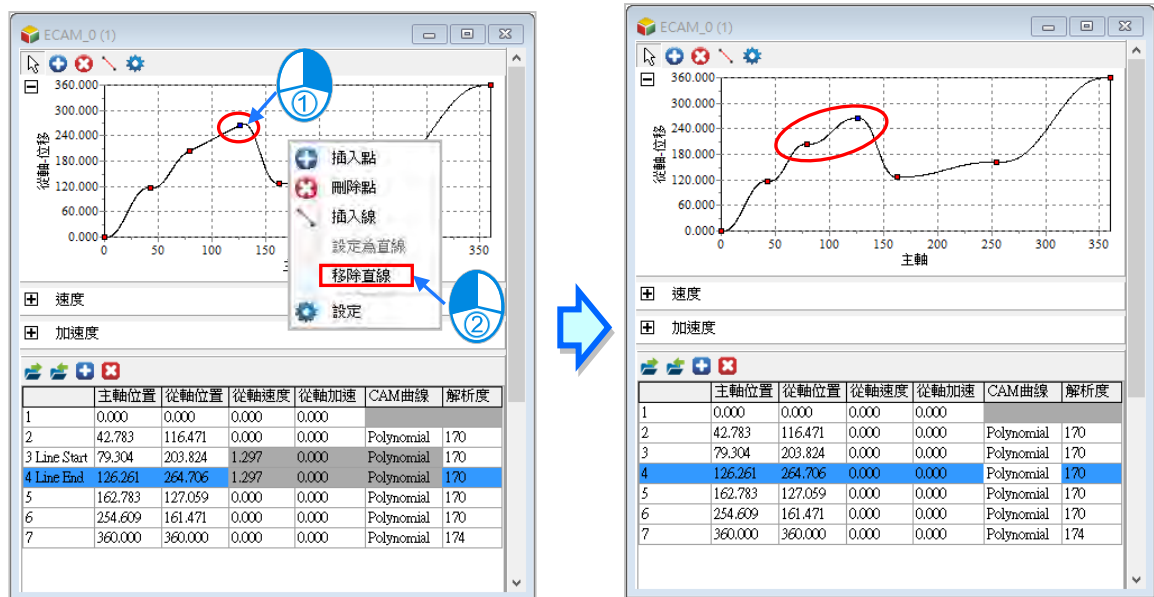
若要將 Polynomial 區間線段設定為直線，可先對關係圖中的關鍵點擊滑鼠右鍵，接著用滑鼠左鍵點選「設定為直線」，Polynomial 區間線段將變為直線。



使用「設定為直線」功能，需注意下列事項：

- 前一個與目前區間必須是 Polynomial 區間才可以使用此功能。
- Line Start 與 Line End 的關鍵點不能與其他直線區間共用直線屬性。

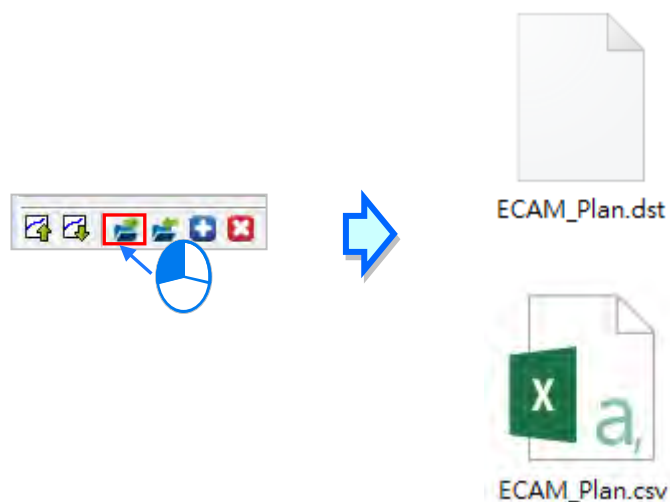
若要移除已設定的直線屬性，需用滑鼠右鍵點選關係圖中 Polynomial 區間線段的 Line Start 或 Line End 關鍵點，接著用滑鼠左鍵點選「移除直線」。



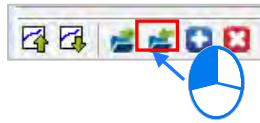
使用「移除直線」功能，需注意下列事項：

- 只有屬性為 Line Start 或 Line End 關鍵點才可以使用此功能。

匯出規劃表的資料，點選規劃表上方功能列的匯出  圖示，依照專案機型匯出的檔案格式會有所不同，且不同的機種的檔案格式不能通用。



要匯入規劃表的檔案時，直接點選匯入  圖示，接著選取要匯入的檔案就可以了。



下載從軸資料必須先將 ISPSOft 與 AS 機器建立連線，下面使用一個範例來操作，首先開啟一張 E-CAM 表並增加至 8 個關鍵點，接著點選「下載從軸資料[PC=>檔案暫存器/伺服]」，之後會彈出一個視窗，選擇要下載到檔案暫存器 (FR) 或是伺服，E-CAM 區域數目為要存取的曲線區域，這裡我們下載 8 個關鍵點所以等於 7 個曲線區域，當變更起始位置或 E-CAM 區域數目時，系統會自動計算結束位置，點擊「開始傳輸」，傳輸完畢後會彈出下載完成的視窗。

22




若要直接下載到伺服，則可以選擇伺服，首先需要選取相對應的伺服編號，接著可以設定起始位置，若勾選「同步資料至 PLC」選項，會把當前的 E-CAM 資料同步至 PLC 中，但專案中的 E-CAM 數量與編號需與 PLC 中的一致，才能夠同步資料。

PC => 檔案暫存器/伺服 (下載)

從軸資料傳輸

檔案暫存器(FR)

起始位置:

結束位置:

伺服

伺服編號:

起始位置:

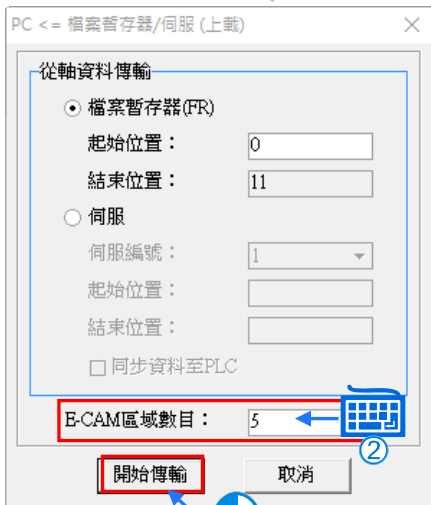
結束位置:

同步資料至 PLC

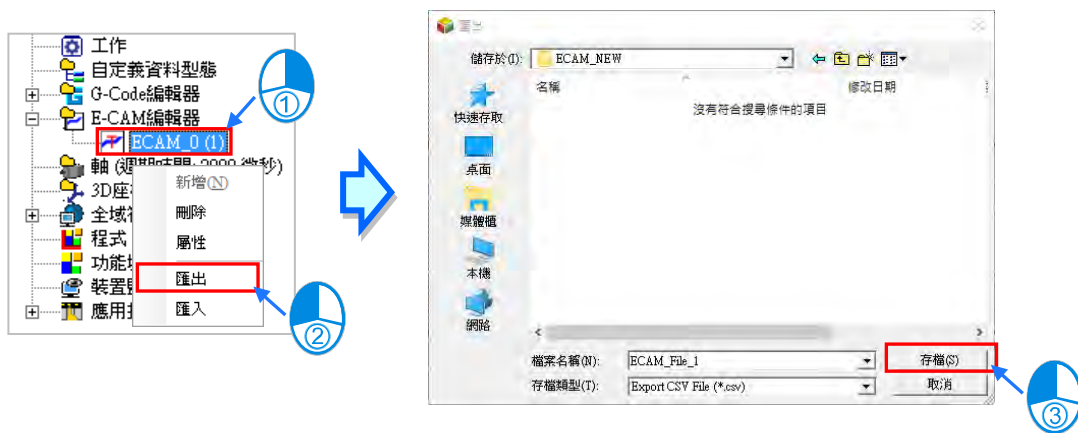
E-CAM 區域數目:

開始傳輸 取消

接著點選「上載從軸資料[PC<=檔案暫存器/伺服]」，之後會彈出一個視窗，此時我們只要上載 6 個關鍵點，所以這裡我們將 E-CAM 區域數目修改為 5，並點擊「開始傳輸」上載完成之後，E-CAM 表會被重新繪製為 6 個關鍵點，如下圖。



若模式為「檔案模式」，則需先將此 E-CAM 表做匯出的動作，才可做編輯。首先，於要匯出的 E-CAM 項目，點擊滑鼠右鍵，再選擇「匯出」，選擇路徑及檔名後，點擊「存檔」，即可完成匯出動作。



22

於匯出的 CSV 檔，輸入 E-CAM 動作的主軸位置 (X) 及從軸位置 (Y)，輸入格式為 0、正整數或正浮點數；輸入位置點數，最多 2048 組 XY 位置點，最少 2 組 XY 位置點，輸入完畢後，做存檔的動作，再回到 ISPSOft，做匯入的動作。

	A	B	C	D
1	:Delta	:E-Cam	Version	V1.0
2	X	Y		
3	0	0		
4	0.280561	0.804357		
5	0.561122	1.608714		
6	0.841682	2.413071		
7	1.122243	3.217427		
8	1.402804	4.021784		
9	1.683365	4.826141		
10	1.963925	5.630497		
11	2.244486	6.434855		
12	2.525047	7.239211		
13	2.805608	8.043568		

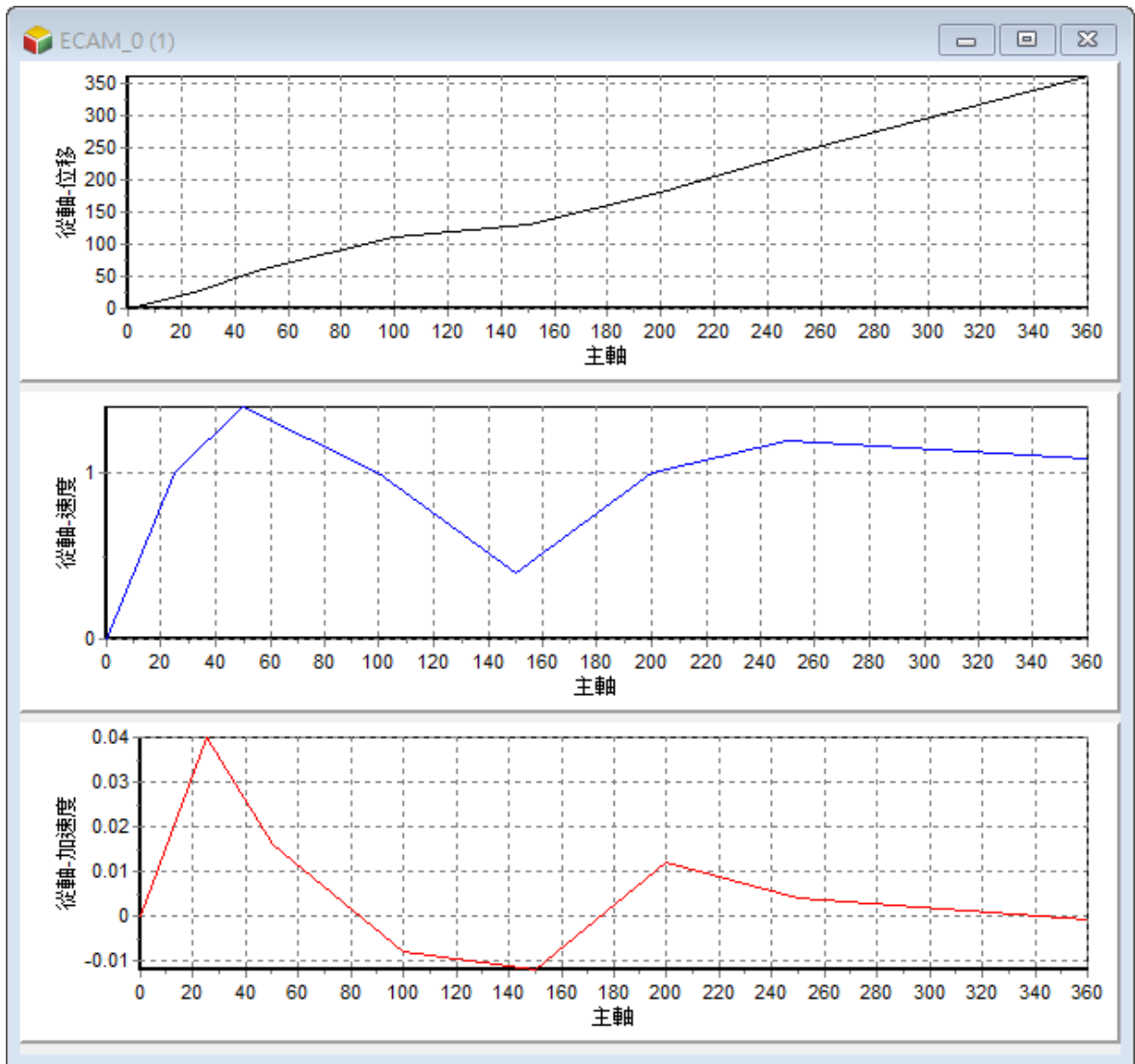
主軸位置

從軸位置

回到 ISPSOft，在要匯入的 E-CAM 表項目，點擊滑鼠右鍵，選擇「匯入」，並選取要開啟的檔案，再點擊「開啟舊檔」，此時會跳出「匯入成功」的資訊視窗，點擊「確定」，即完成匯入動作。



匯入成功後，可以看見相對應的 E-CAM 圖表，此圖表僅能檢視匯入檔案的數據圖示。



22

完成 E-CAM 編輯器的編寫之後，便可在 POU 中搭配運動控制指令或功能塊使用，例如以 MC_Camin 指令依 E-CAM 編輯器項目編號啟動指定的 E-CAM 項目。關於 MC_Camin 的使用請參閱各機種相關指令手冊。

22



切換 E-CAM 圖表的模式時，在點擊「確定」後，將出現「資訊」視窗，詢問「若模式被改變，則所有資料將被初始化，是否繼續？」，此時選擇「是」，ECAM 表資料將全被初始化。



第23章 精靈工具

目錄

23.1 定位規劃表.....	23-2
23.1.1 關於定位規劃表.....	23-2
23.1.2 設定定位規劃表.....	23-2
23.1.3 定位規劃表的模擬.....	23-6
23.1.4 下載與上載定位規劃表	23-8
23.1.5 軸控模式-單軸點對點模式	23-9
23.1.6 軸控模式-單軸多段輸出模式	23-10
23.1.7 軸控模式-雙軸直線補間模式	23-11
23.1.8 軸控模式-雙軸圓弧補間模式	23-11
23.2 資料示波器.....	23-13
23.2.1 關於資料示波器.....	23-13
23.2.2 開啟資料示波器.....	23-13
23.2.3 設定取樣參數	23-14
23.2.4 取樣模式.....	23-17
23.2.5 量測	23-19
23.3 資料記錄器.....	23-23
23.3.1 關於資料記錄器.....	23-23
23.3.2 開啟資料記錄器.....	23-24
23.3.3 設定取樣參數	23-26
23.3.4 監看與錄製模式.....	23-27
23.3.5 量測	23-31
23.4 高速計數器精靈	23-34
23.4.1 關於高速計數器精靈	23-34
23.4.2 使用高速計數器精靈	23-35
23.5 NTC 精靈.....	23-36
23.5.1 關於 NTC 精靈.....	23-36
23.5.2 使用 NTC 精靈.....	23-37

23.1 定位規劃表

23.1.1 關於定位規劃表

ISPSOft 提供使用者一種簡易的表格設定，來完成單軸或兩軸的運動控制，使用者無須了解複雜的運動控制規範，只要完成表格的建立並下載，再搭配應用指令啟動，主機會會依據設定的編號依序執行各軸輸出。

23.1.2 設定定位規劃表

點選工具列的 **精靈 (!)** > **定位規劃表** 啟動定位規劃表設定視窗。定位規劃表分成三個區域說明如下圖。



23



- ❶ 圖示工具列：提供各項操作定位規劃表的按鈕，功能說明如下表。
- ❷ 參數設定區：軸設定用於設定運動軸的控制方式。上載密碼保護用於設定密碼。定位表格用於設定軸控模式與運動路線。
- ❸ 目標模擬區：使用**相對模式 (符號/位址)** 或**絕對模式 (符號/位址)**，可對目標位置做模擬。

圖示工具列說明：

圖示	名稱	功能說明
	匯入	將.pnt 匯入到目前定位規畫表。
	匯出	將定位規劃表設定匯出成.pnt。

圖示	名稱	功能說明
	上傳定位規劃表	從主機讀回定位規劃表設定。
	下載定位規劃表	將定位規劃表設定寫入主機。

在軸設定分頁，可以設定每個軸的控制輸出方式，顯示的軸數會隨著機種支援而數量有所不同，如果要執行雙軸補間運動，則「輸出方式」不可選擇 Pulse，需選擇 Pulse/Dir 或 AB phase。在軸輸出方式後面，勾選「啟動」後可以對此軸輸入自訂的「數值設定」與「單位名稱」，及輸入 PLC 實際輸出脈波數區中的「數值設定」，如下圖，則代表軸 1 在此定位規劃表中的 100 個 mm 將輸出 1000 個脈波。按下「恢復初始值」按鈕回復單位設定頁面的初始設定。

軸設定 | 上傳密碼保護 | 定位表格

設定	使用者自訂基本單位			PLC 實際輸出脈波數
	輸出方式	啟動	數值設定	單位名稱
軸 1	Pulse	<input checked="" type="checkbox"/>	100	mm
軸 2	Pulse/Dir	<input checked="" type="checkbox"/>	10	mm
軸 3	AB phase	<input checked="" type="checkbox"/>	1	mm
軸 4	Pulse	<input type="checkbox"/>	1	mm
軸 5	Pulse	<input type="checkbox"/>	1	mm
軸 6	Pulse	<input type="checkbox"/>	1	mm

PLC 實際輸出脈波數
數值設定
1000
100
10
1
1
1

恢復初始值

小數點後位數 1

備註：
1. 設定數值為 1，以及輸出脈波數為 1，即表示不啟動單位轉換功能。
2. 設定數值與實際輸出脈波數，僅能輸入 32 位元的整數值，但定位表格內目標位置則可輸入浮點數。
3. 當單位轉換過程中，實際輸出脈波數若有小數部分，將自動被捨去。
4. 針對沒有使用到的軸，其該軸輸出類型的設定並不會下載到機器中。

23

在上載密碼保護分頁，可以設定密碼，用於保護下載到主機的定位規劃表在上載時必須輸入密碼，設定方式於密碼與確認密碼欄位輸入一致的密碼後，按下「確定」成功加密的話，下方會提示密碼保護啟動。如果要解除密碼設定時，只要再次於密碼欄位輸入密碼，按下「確定」成功解鎖的話，下方會提示未啟動密碼保護。

⚠ 請注意「上載密碼保護」僅提供上載定位規劃表時需要密碼，若將別的規劃表下載到有「上載在密碼保護」的主機，定位規劃表還是會被覆蓋。

軸設定 | 上傳密碼保護 | 定位表格

密碼 (4~16個字元)

確認密碼

顯示字元

確定 取消

未啟動密碼保護

➔

軸設定 | 上傳密碼保護 | 定位表格

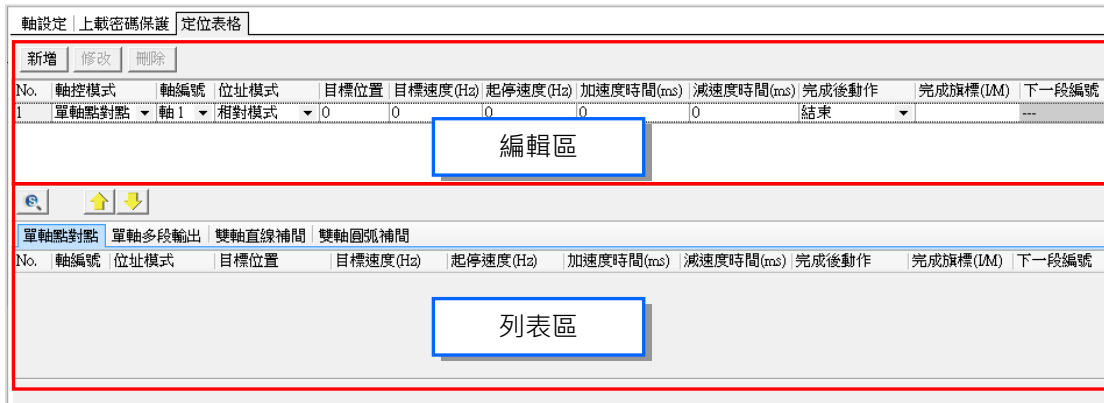
密碼 (4~16個字元)

顯示字元

確定

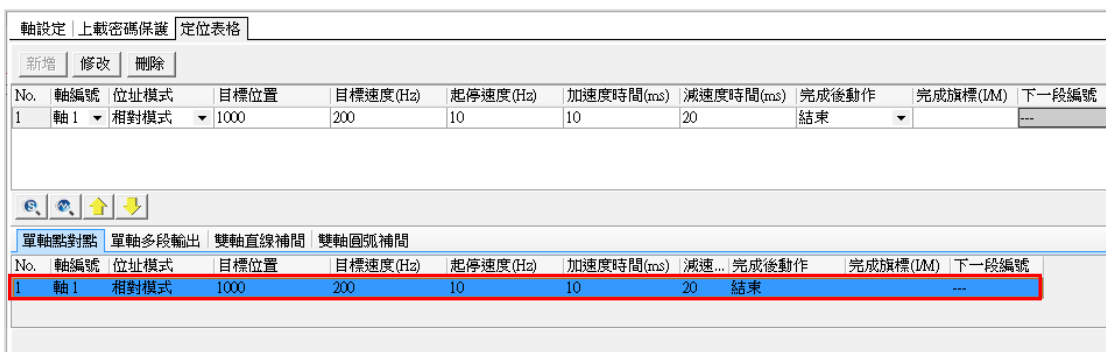
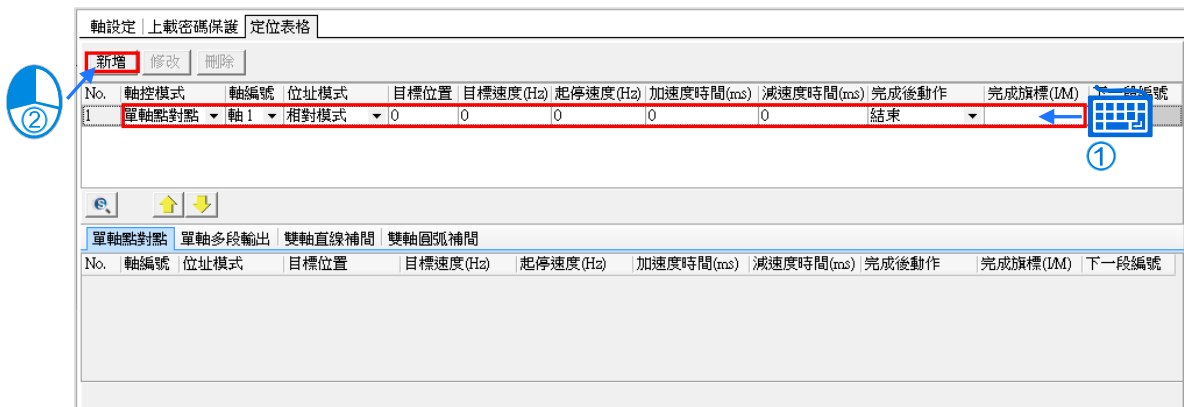
密碼保護啟動

在定位表格分頁，編輯區供新增、修改或刪除用，新增完的資料會出現在列表區。

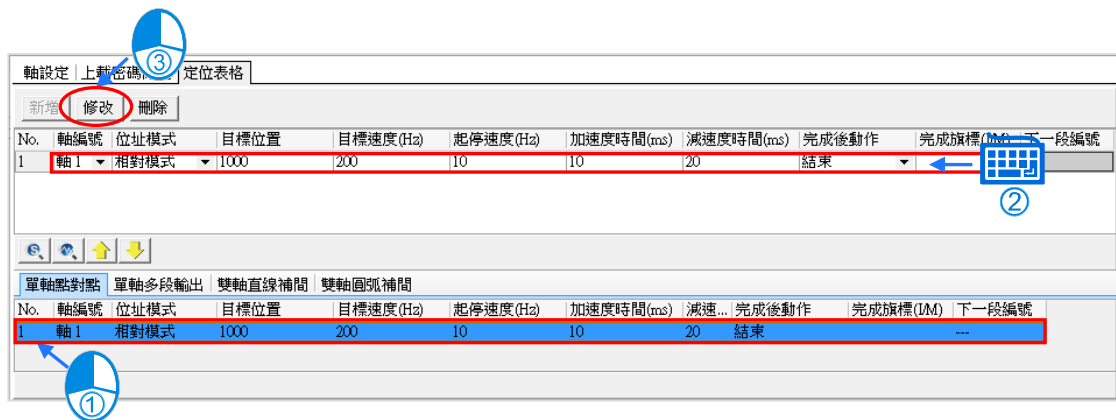


要新增一筆資料，先在編輯填入各項設定欄位後，按下「新增」，便會依據選擇的軸控模式，加入到下方相對的軸控模式分頁中。

23

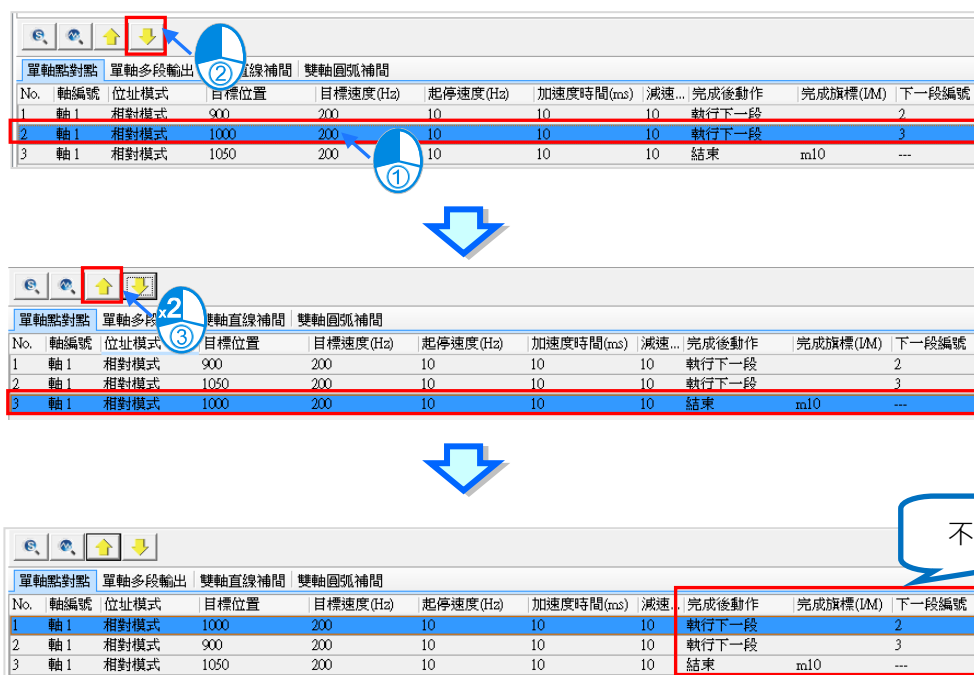


要修改則是在列表區選取該筆資料，之後再於編輯區編輯此筆資料內容，最後按下「修改」，便會更新到列表區中，如下圖。

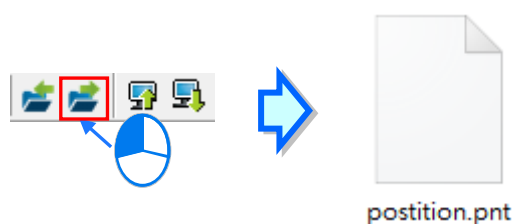


若要刪除則在列表區選取該筆資料，之後再於編輯區，按下「刪除」，此筆資料便會從列表區中刪除。在列表區的每一列編號，可藉由上移 / 下移 來做編號增減的動作，但需注意欄位「完成後動作」、「下一段編號」及「完成旗標 (I/M)」是不會隨著移動，如下圖。


23

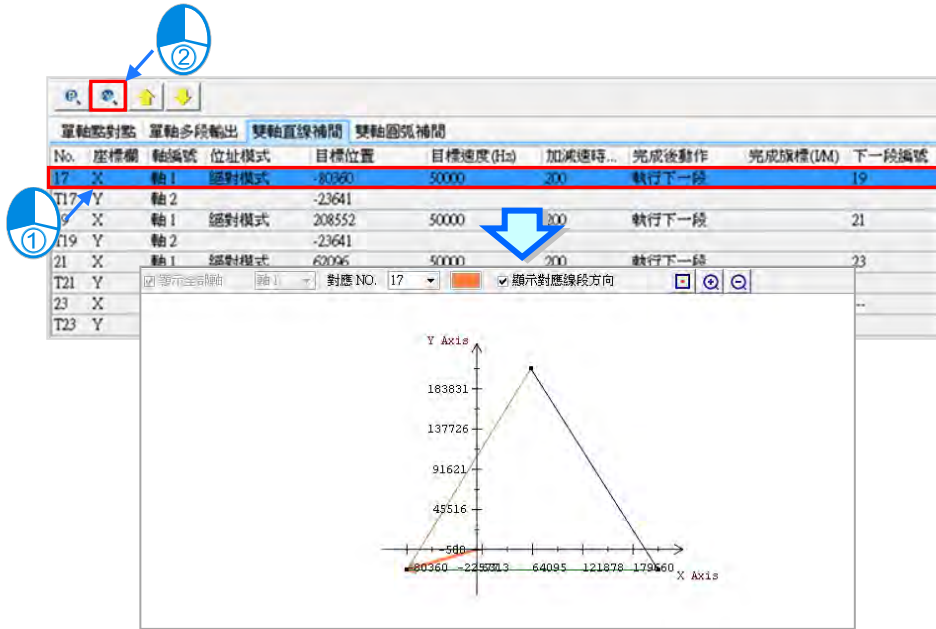


完成定位表格資料，可以點選「匯出」，將檔案匯出成.pnt，如下圖，之後就可以點選「匯入」，選取.pnt 檔案，定位表格資料就會被匯入。



23.1.3 定位規劃表的模擬


在列表區點選要模擬的任一行後，再按「指定編號繪圖」，則會彈出模擬圖視窗，透過圖形化的方式顯示規劃表的路徑，如下圖，此時會繪製從 No.17 開始到結束的路徑。

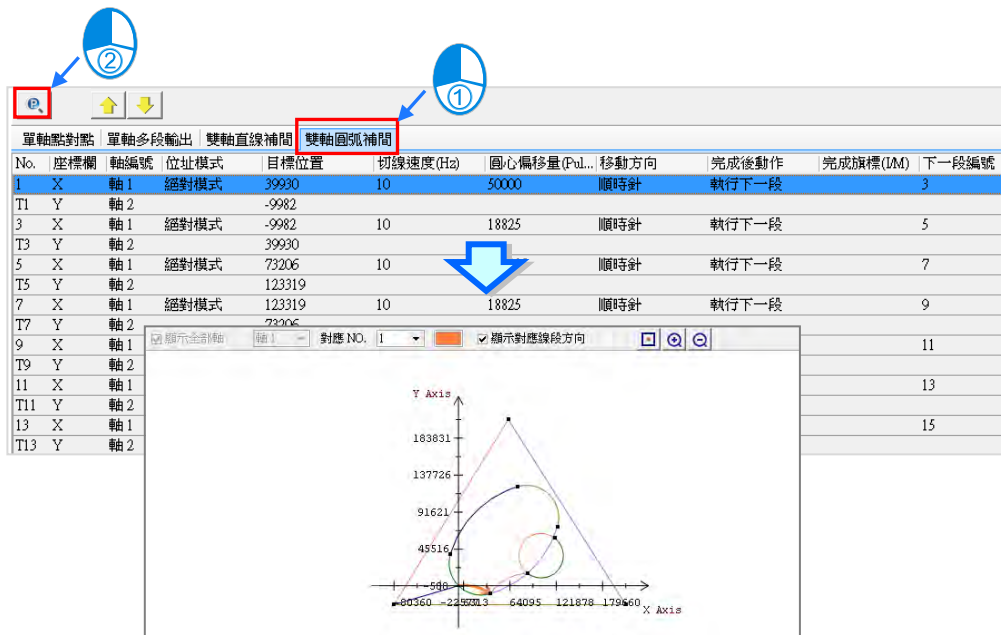


23

模擬圖工具列說明如下。


圖示	功能說明
<input type="checkbox"/> 顯示全部軸 軸 1	勾選顯示全部軸或是只顯示某一軸。
對應 NO. 1 	選擇某一編號在圖形中加粗標記，也可點選右側的色塊變更顏色。
<input checked="" type="checkbox"/> 顯示對應線段方向	顯示箭頭代表運動方向。
	調整圖形為最適大小。
	依 X 方向放大圖形。
	依 X 方向縮小圖形。
	依 Y 方向放大圖形。
	依 Y 方向縮小圖形。
	放大圖形。
	縮小圖形。

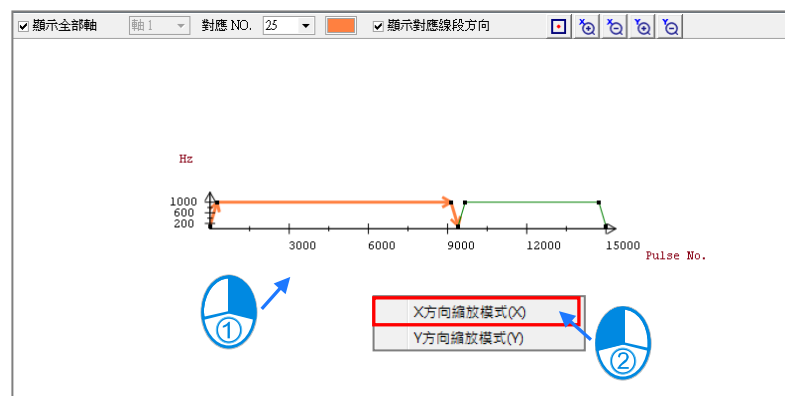
雙軸直線補間模式與雙軸圓弧補間模式，可在同一模擬圖中顯示，混合兩模式連續的軌跡，先點擊列表區的「雙軸圓弧補間」分頁，再點選「雙軸模擬模式」，如下圖。



No.	座標欄	軸編號	位址模式	目標位置	切線速度(Hz)	圓心偏移量(Pul.)	移動方向	完成後動作	完成旗標(IM)	下一段編號
1	X	軸 1	絕對模式	39930	10	50000	順時針	執行下一段		3
T1	Y	軸 2		-9982						
3	X	軸 1	絕對模式	-9982	10	18825	順時針	執行下一段		5
T3	Y	軸 2		39930						
5	X	軸 1	絕對模式	73206	10		順時針	執行下一段		7
T5	Y	軸 2		123319						
7	X	軸 1	絕對模式	123319	10	18825	順時針	執行下一段		9
T7	Y	軸 2		73206						
9	X	軸 1								11
T9	Y	軸 2								13
11	X	軸 1								15
T11	Y	軸 2								
13	X	軸 1								
T13	Y	軸 2								

23

單軸點對點模式與單軸多段輸出模式的模擬圖則只能顯示同一模式的軌跡，先點擊列表區的「單軸點對點」分頁，再點選「單軸模擬模式」，且於單軸模擬圖視窗空白處，點擊滑鼠右鍵，之後會彈出選單，可以切換滾輪要控制的方向，如下圖。



目標模擬區的功能，則需先在全域變數宣告符號，必須是 DWORD 或 DINT 以上的型態，接著改變編輯區的位址模式為「絕對模式 (符號/位址)」或「相對位置 (符號/位址)」，再於「目標位置」填入符號或裝置，如下圖。

! 定位規劃表的「目標位置」最小單位為 DWORD，所以若在填入，D100 則會佔掉 D100 及 D101 長度，請小心使用!

全域符號					
類別	符號名稱	位址	資料型態	初始值(下載時...	符號註解
VAR	Data_1	D100	DWORD	N/A	

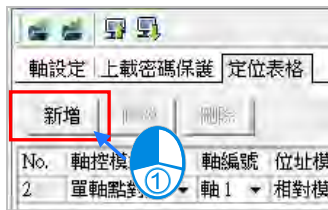


No.	軸控模式	軸編號	位址模式	目標位置	目標	起停速度(Hz)	加速度時間(ms)	減速度時間(ms)	完成後動作	完成旗標(JM)	下一段編號
2	單軸點對點	軸 1	相對模式(符號位址)	Data_1	200	10	10	10	結束		---



填完後，點擊「新增」之後在目標模擬區，將自動增加此筆資料，此時可以在此筆資料的目標位置(模擬)的項目，輸入欲模擬的目標位置值，之後可以透過「單軸模擬模式」或「雙軸模擬模式」，來檢視模擬路徑。

23



NO.	符號裝置	目標位置(模擬)
1	D200	500
2	Data_1	1000

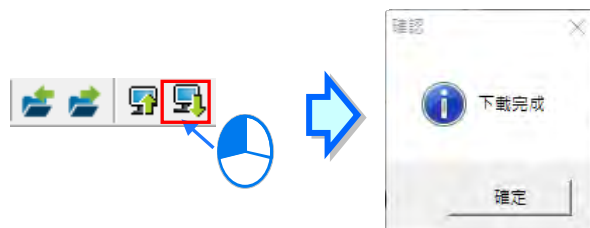


23.1.4 下載與上載定位規劃表

使用上載與下載定位規劃表功能，需先將 ISPSOft 與主機正常連線，便可以點選「上載」將主機儲存的定位規劃表上載，當主機中的定位規劃表有上載密碼保護時，將出現密碼詢問視窗，輸入正確的密碼才可將定位規劃表上載，如下圖。



點選「下載」按鈕，則可以將定位規劃表下載至主機中。



23.1.5 軸控模式-單軸點對點模式

設定單軸點對點模式，說明如下。

No.	軸控模式	軸編號	位址模式	目標位置	目標速度(Hz)	起停速度(Hz)	加速度時間(ms)	減速度時間(ms)	完成後動作	完成旗標(I/M)	下一段編號
1	單軸點對點	軸 1	絕對模式	0	0	0	0	0	結束		---

- **No.**：編號由系統自動產生，最大為 200。
- **軸控模式**：選擇運動的類型。此處選擇為**單軸點對點**。
- **軸編號**：此筆資料控制的軸。
- **位址模式**：「**相對模式**」依據目前的座標位址而增減。
 「**絕對模式**」依據絕對座標做為目標位址而增減。
 「**相對模式 (符號/位址)**」依據目前所代表的座標位址而增減，可以使用符號/位址。
 「**絕對模式 (符號/位址)**」依據絕對座標做為目標位址而增減，可以使用符號/位址。
- **目標位置**：目標座標。**位址模式**若選取包含 (符號/位置)，則這邊可以使用符號/位置。
- **目標速度**：目標的最高速度。
- **起停速度**：開始與結束的速度。
- **加速度時間**：到達目標速度的時間。
- **減速度時間**：到達停止的時間。
- **完成後動作**：「**執行下一段**」完成此筆資料後接續到下一筆。
 「**結束**」完成此筆資料後結束動作。
- **完成旗標 (I/M)**：讓此筆資料執行完畢時，將暫存器設為 ON，但只有**完成後動作**設為**結束**時才可設定暫存器範圍為 M0~M8191 或 I510~I519。
- **下一段編號**：接續下一筆資料的編號，不限制指定同一個軸編號，但是須注意同一軸編號不可有兩個以上表格編號或定位輸出指令同時輸出。若**完成後動作**設為**結束**後則自動帶入 ---。

23

表格輸入注意事項：

1. 目標速度不可輸入為 0；在相對模式下，目標位置不可輸入為 0。
2. I 中斷編號可重複使用，但使用者須自行判斷此中斷為哪一個列編號輸出完成。
3. 啟動表格輸出的指令為 API2718 TPO 指令；當 PLC 正在運行中，可運用 API2719 TPWS 修改輸出參數。

23.1.6 軸控模式-單軸多段輸出模式

設定單軸多段輸出模式，說明如下。

No.	軸控模式	軸編號	位址模式	目標位置	目標速度(Hz)	加減速時間(ms)	中斷訊號(I/M)	完成後動作	完成旗標(I/M)	下一段編號
1	單軸多段輸出 ▾	軸 1 ▾	相對模式 ▾	0	0	0		執行下一段 ▾		2

- **No.**：編號由系統自動產生，最大為 200。
- **軸控模式**：選擇運動的類型。此處選擇為**單軸多段輸出**。
- **軸編號**：此筆資料控制的軸。
- **位址模式**：「**相對模式**」依據目前的座標位址而增減。
 「**絕對模式**」依據絕對座標做為目標位址而增減。
 「**相對模式 (符號/位址)**」依據目前所代表的座標位址而增減，可以使用符號/位址。
 「**絕對模式 (符號/位址)**」依據絕對座標做為目標位址而增減，可以使用符號/位址。
- **目標位置**：目標座標。**位址模式**若選取包含 (符號/位置)，則這邊可以使用符號/位置。
- **目標速度**：目標的最高速度。
- **加減速時間**：到達目標速度及到達停止的時間。
- **中斷訊號 (I/M)**：目前表格進行尚未完成時，若中斷訊號設定的暫存器為 ON，則進行下一段定位表格。可輸入 I 中斷編號或 M 暫存器名稱。支援的中斷編號請參考產品相關程式手冊。
- **完成後動作**：「**執行下一段**」完成此筆資料後接續到下一筆。
 「**結束**」完成此筆資料後結束動作。
- **完成旗標 (I/M)**：讓此筆資料執行完畢時，將暫存器設為 ON，但只有**完成後動作**設為**結束**時才可設定暫存器範圍為 M0~M8191 或 I510~I519。
- **下一段編號**：接續下一筆資料的編號，不限制指定同一個軸編號，但是須注意同一軸編號不可有兩個以上表格編號或定位輸出指令同時輸出。若**完成後動作**設為**結束**後則自動帶入 ---。

表格輸入注意事項：

1. 在相對模式下，目標位置不可輸入為 0；在絕對模式下，相鄰的兩個列編號的目標位置不可輸入相同位置。
2. 若加減速時間不為 0，加速度間隔時間固定為 1ms；若加減速時間為 0，代表不作加減速。
3. I 中斷編號可重複使用，但使用者須自行判斷此中斷為哪一個列編號輸出完成。
4. 啟動表格輸出的指令為 API2718 TPO 指令；當 PLC 正在運行中，可運用 API2719 TPWS 修改輸出參數。

23.1.7 軸控模式-雙軸直線補間模式

設定雙軸直線補間模式，在 XY 平面上的直線補間運動，使用此模式每次會占用兩筆資料。說明如下。

No.	軸控模式	座標欄	軸編號	位址模式	目標位置	目標速度(Hz)	加減速時間(...)	完成後動作	完成旗標(IM)	下一段編號
I	雙軸直線補間	X	軸 1	相對模式	0	0	0	結束		---
T1		Y	軸 2		0					

- **No.**：編號由系統自動產生，雙軸模式會占用兩筆資料，第二筆資料的編號前會加註 T，因此全部使用雙軸模式時，編號最大為 100 與 T100 共 100 組。
- **軸控模式**：選擇運動的類型。此處選擇為**雙軸直線補間**。
- **座標欄**：此筆資料在模擬圖中的 X 或 Y。
- **軸編號**：此筆資料控制的軸。
- **位址模式**：「**相對模式**」依據目前的座標位址而增減。
 - 「**絕對模式**」依據絕對座標做為目標位址而增減。
 - 「**相對模式 (符號/位址)**」依據目前所代表的座標位址而增減，可以使用符號/位址。
 - 「**絕對模式 (符號/位址)**」依據絕對座標做為目標位址而增減，可以使用符號/位址。
- **目標位置**：目標座標。**位址模式**若選取包含 (符號/位置)，則這邊可以使用符號/位置。
- **目標速度**：目標的最高速度。
- **加減速時間**：到達目標速度及到達停止的時間。
- **完成後動作**：「**執行下一段**」完成此筆資料後接續到下一筆。
 - 「**結束**」完成此筆資料後結束動作。
- **完成旗標 (IM)**：讓此筆資料執行完畢時，將暫存器設為 ON，但只有**完成後動作**設為**結束**時才可設定暫存器範圍為 M0~M8191 或 I510~I519。
- **下一段編號**：接續下一筆資料的編號，不限制指定同一個軸編號，但是須注意同一軸編號不可有兩個以上表格編號或定位輸出指令同時輸出。若**完成後動作**設為**結束**後則自動帶入 ---。

23

表格輸入注意事項：

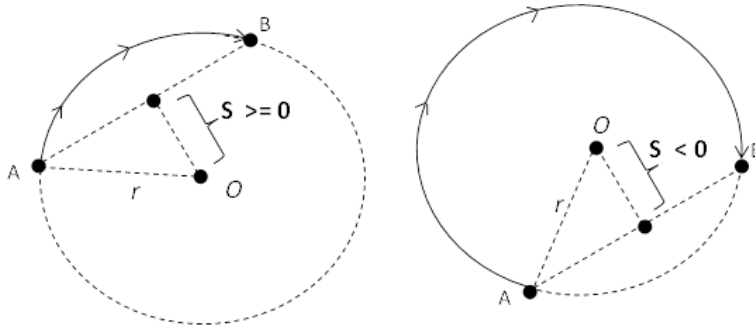
1. 當目標速度或加減速時間輸入為 0 時，PLC 會自動以預設之速度；在相對模式下，目標位置不可輸入為 0；在絕對模式下，相鄰的兩個列編號的目標位置不可輸入相同位置。
2. I 中斷編號可重複使用，但使用者須自行判斷此中斷為哪一個列編號輸出完成。
3. 啟動表格輸出的指令為 API2718 TPO 指令；當 PLC 正在運行中，可運用 API2720 TPWL 修改輸出參數。

23.1.8 軸控模式-雙軸圓弧補間模式

設定雙軸圓弧補間模式，在 XY 平面上的圓弧補間運動，使用此模式每次會占用兩筆資料。說明如下。

No.	軸控模式	座標欄	軸編號	位址模式	目標位置	切線速度(Hz)	圓心偏移量	移動方向	完成後動作	完成旗標(IM)	下一段編號
I	雙軸圓弧補間	X	軸 1	相對模式	0	0	0	順時針	結束		---
T1		Y	軸 2		0						

- **No.**：編號由系統自動產生，雙軸模式會占用兩筆資料，第二筆資料的編號前會加註 T，因此全部使用雙軸模式時，編號最大為 100 與 T100 共 100 組。
- **軸控模式**：選擇運動的類型。此處選擇為**雙軸圓弧補間**。
- **座標欄**：此筆資料在模擬圖中的 X 或 Y。
- **軸編號**：此筆資料控制的軸。
- **位址模式**：「**相對模式**」依據目前的座標位址而增減。
 「**絕對模式**」依據絕對座標做為目標位址而增減。
 「**相對模式 (符號/位址)**」依據目前所代表的座標位址而增減，可以使用符號/位址。
 「**絕對模式 (符號/位址)**」依據絕對座標做為目標位址而增減，可以使用符號/位址。
- **目標位置**：目標座標。**位址模式**若選取包含 (符號/位置)，則這邊可以使用符號/位置。
- **切線速度**：沿圓弧切線的方向運動的向量速度。
- **圓心偏移量**：圓弧目標點與起始位置的連線中心到圓心的距離。如下圖的 S，由 A 點到 B 點，順時針， $S > 0$ 代表要畫小於 180° 的弧； $S < 0$ 代表要畫大於 180° 的弧。



- **移動方向**：選擇**順時針**或**逆時針**方向。
- **完成後動作**：「**執行下一段**」完成此筆資料後接續到下一筆。
 「**結束**」完成此筆資料後結束動作。
- **完成旗標 (I/M)**：讓此筆資料執行完畢時，將暫存器設為 ON，但只有**完成後動作**設為**結束**時才可設定暫存器範圍為 M0~M8191 或 I510~I519。
- **下一段編號**：接續下一筆資料的編號，不限制指定同一個軸編號，但是須注意同一軸編號不可有兩個以上表格編號或定位輸出指令同時輸出。若**完成後動作**設為**結束**後則自動帶入 ---。

表格輸入注意事項：

1. PLC 會依切線速度，自動計算各段速度；在相對模式下，目標位置不可輸入為 0；在絕對模式下，相鄰的兩個列編號的目標位置不可輸入相同位置。
2. I 中斷編號可重複使用，但使用者須自行判斷此中斷為哪一個列編號輸出完成。
3. 啟動表格輸出的指令為 API2718 TPO 指令；當 PLC 正在運行中，可運用 API2721 TPWC 修改輸出參數。

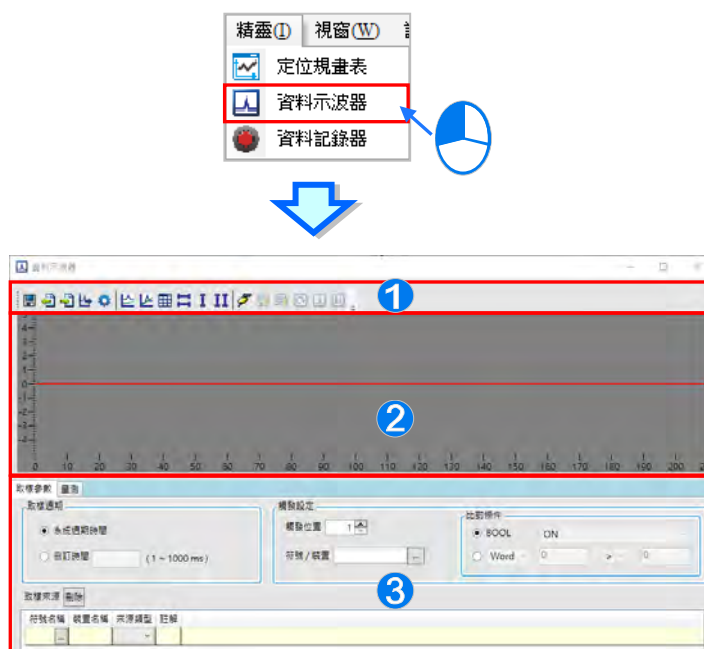
23.2 資料示波器

23.2.1 關於資料示波器

資料示波器用於在某個觸發條件達成後，即時收集變數符號或裝置的數值與狀態，以波形圖顯示，方便分析數值趨勢。

23.2.2 開啟資料示波器

要使用此功能時，請先編譯目前專案後，再點選 **精靈 (!)** > **資料示波器**，開啟資料示波器視窗，如下圖。



- ❶ 圖示工具列：提供各項操作資料示波器的按鈕，功能說明如下。
- ❷ 波形顯示區：顯示即時變數符號或裝置的波形圖。
- ❸ 參數設定區：取樣參數分頁用於設定取樣的來源與觸發條件。量測分頁用於顯示量測工具所量測到的數值及調整波形的顏色與比例等。

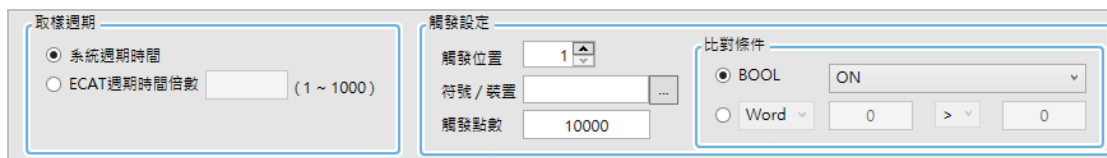
圖示工具列按鈕功能說明

圖示	名稱	功能說明
	儲存全部	將取樣設定資料與取樣波形資料結果儲存到指定路徑中。
	匯出取樣設定	將取樣設定內容匯出到指定路徑。
	匯入取樣設定	由指定路徑將取樣設定內容匯入。
	匯出波形資料	將取樣波形資料以.bmp 或.csv 格式匯出到指定路徑。

圖示	名稱	功能說明
	設定	設定資料示波器波形與環境的顏色。
	分離波形	將多條波形自動以間隔分離。
	重疊波形	將多條波形以中心線重疊。
	顯示格線	切換是否顯示格線。
	水平量測線	波形顯示區將出現兩條水平線，供使用者量測波形。
	垂直量測線	波形顯示區將出現一條垂直線，供使用者量測波形。
	雙垂直量測線	波形顯示區將出現兩條垂直線，供使用者量測波形。
	連線/離線	切換電腦與主機的連線或離線模式。
	上載 PLC 取樣設定	從主機讀回資料示波器的取樣設定。
	下載取樣設定到 PLC	將資料示波器的取樣設定寫入主機。
	即時監看	電腦即時從主機讀取來源的取樣波形。
	單次觸發	觸發條件滿足後，從主機讀取觸發前後共 200 點波形資料一次。
	連續觸發	每當觸發條件滿足後，便從主機讀取觸發前後共 200 點波形資料。

23.2.3 設定取樣參數

在取樣參數分頁，依照開啟的專案機種不同，項目會有所不同請參考下表。**取樣週期**是要設定取樣的時間。**觸發條件**則是要設定觸發位置與條件，當觸發條件達成時，主機將保存共 200 筆的取樣資料，並顯示到波形顯示區。





名稱	說明	AS	AHxxEMC	AH5x1
系統週期時間	主機掃描週期	V	V	V
ECAT 週期時間倍數	ECAT 掃描週期		最大 1000	
自訂時間	自訂的時間	最大 1000		最大 1000
觸發位置	在圖上要觸發的位置	最大 200	最大 10000	最大 200
符號/裝置	觸發的裝置/符號	只支援全域符號	支援全域符號、區域符號、軸符號、ECAT 符號、STRUCT 符號	只支援全域符號

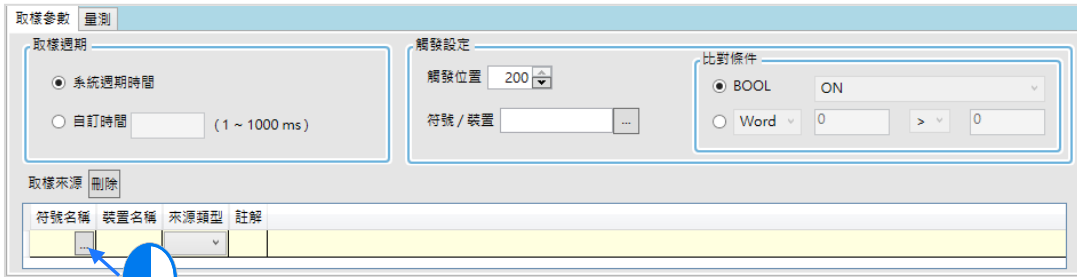
名稱	說明	AS	AHxxEMC	AH5x1
觸發點數	設定觸發位置的最大值		最大 10000	

當符號/裝置欄位選擇為 **BOOL** 的變數符號或裝置時，比對條件會自動選擇到 **BOOL**，如下圖，也可在下拉選單選擇 **ON**、**OFF**、**上緣觸發**、**下緣觸發**、**上或下緣觸發**。當設定的符號或裝置滿足此設定條件時，則觸發顯示在資料示波器上。

當符號/裝置欄位選擇為 **Word** 或 **DWord** 或 **Real** 的變數符號或裝置時，比對條件會自動選擇到 **Word**，如下圖，也可在下拉選單選擇 **DWord** 或 **Real** 資料型態，並使用條件式 **>**、**<**、**=**、**!=**、**~**、**!~**，搭配前後的數字欄位來做設定。當設定的符號或裝置滿足此條件時，則觸發顯示在資料示波器上。如下圖，為當 **D100** 的數值為 **0 > 0** 時，滿足觸發條件；若設定為 **10 !~ 100**，則代表 **D100** 為未滿 10 或超過 100，才滿足觸發條件。

23

要新增取樣符號，先點選符號名稱的  按鈕，接著在彈出的視窗點選符  按鈕，選擇要加入的變數符號，或直接在 **裝置名稱** 欄位輸入裝置名稱，按下「**新增**」後，就會加入取樣來源列表中，如下圖。



23

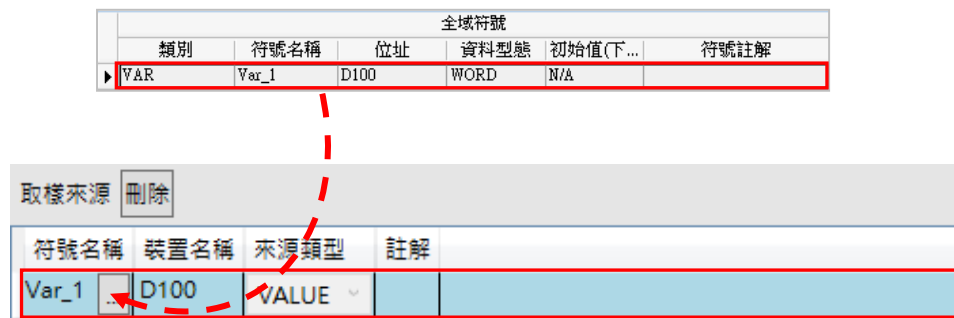


OR



符號名稱	裝置名稱	來源類型	註解
Var_1	D100	VALUE	



只有宣告在全域符號表的變數，並且編譯過專案，才可以這邊被選取，若有手動設定裝置位址或符號註解，也會一併帶入。




要刪除已加入的取樣來源時，點選要刪除的項目後，按下「刪除」按鈕，即可把該項目刪除。

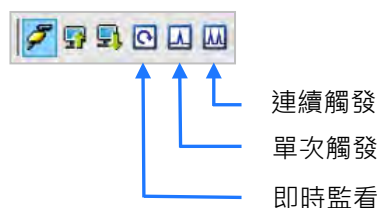


23

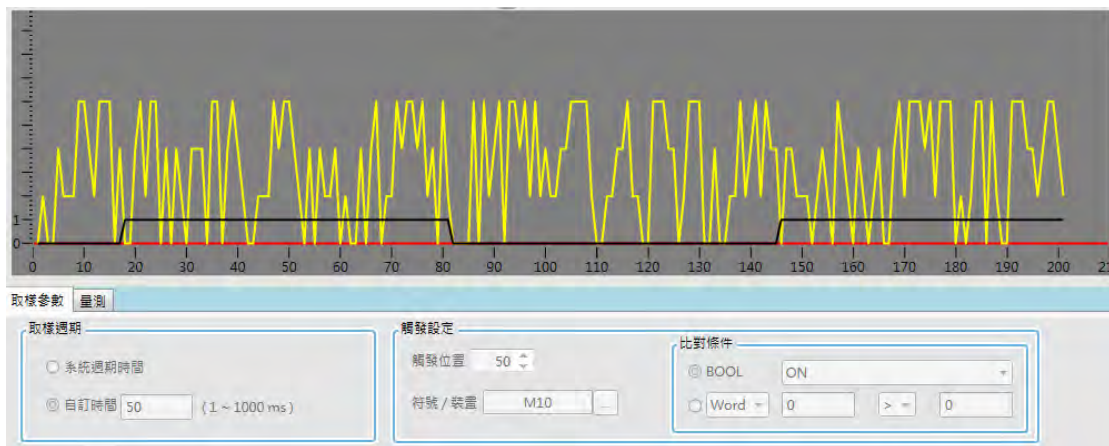
完成取樣參數的設定後，可按下圖示工具列的「匯出」，將取樣參數設定保存成.dtr 檔，之後可以在新專案使用「匯入」將取樣參數設定還原。

23.2.4 取樣模式

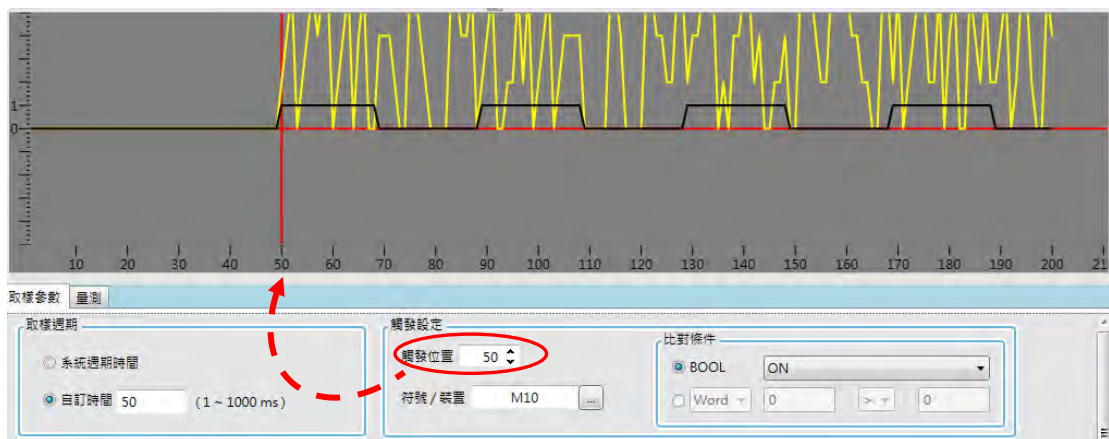
完成資料示波器的取樣參數設定後，在連線模式時，先下載參數設定，便可以使用圖示工具列的按鈕切換下述三種模式監看波形。



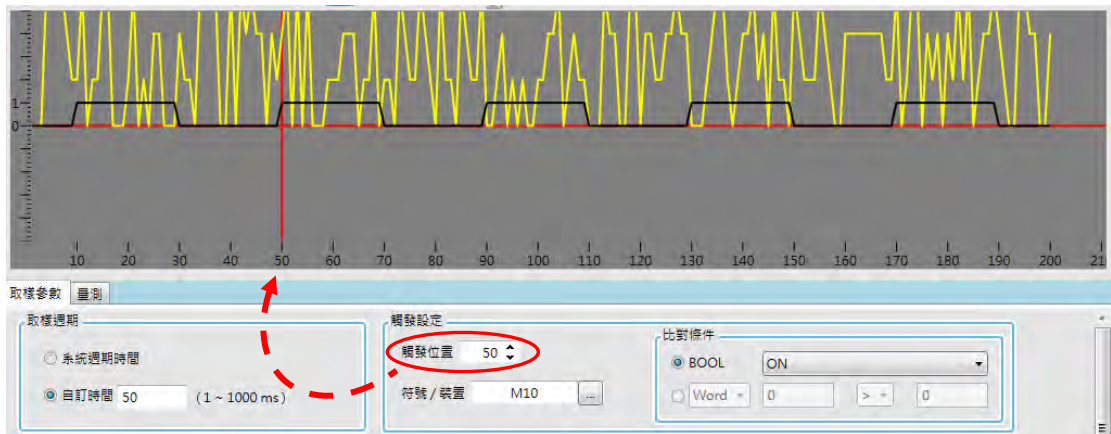
- 即時監看：依據設定的取樣週期，將取樣來源的變數符號或裝置的數值或狀態不斷繪製到資料示波器上。畫面會隨之向左捲動，只保留最新的 200 筆資料。



- 23 ● 單次觸發：依據前述的觸發位置設定，當觸發條件滿足時，將觸發位置之後的取樣資料，顯示在波形區，只觸發一次。如下圖，因觸發位置設為 50，故當觸發條件 M10=ON 時，會將取樣來源依據取樣週期取樣繪製到 50~200 的範圍中。波形中的紅色垂直線即代表觸發位置。



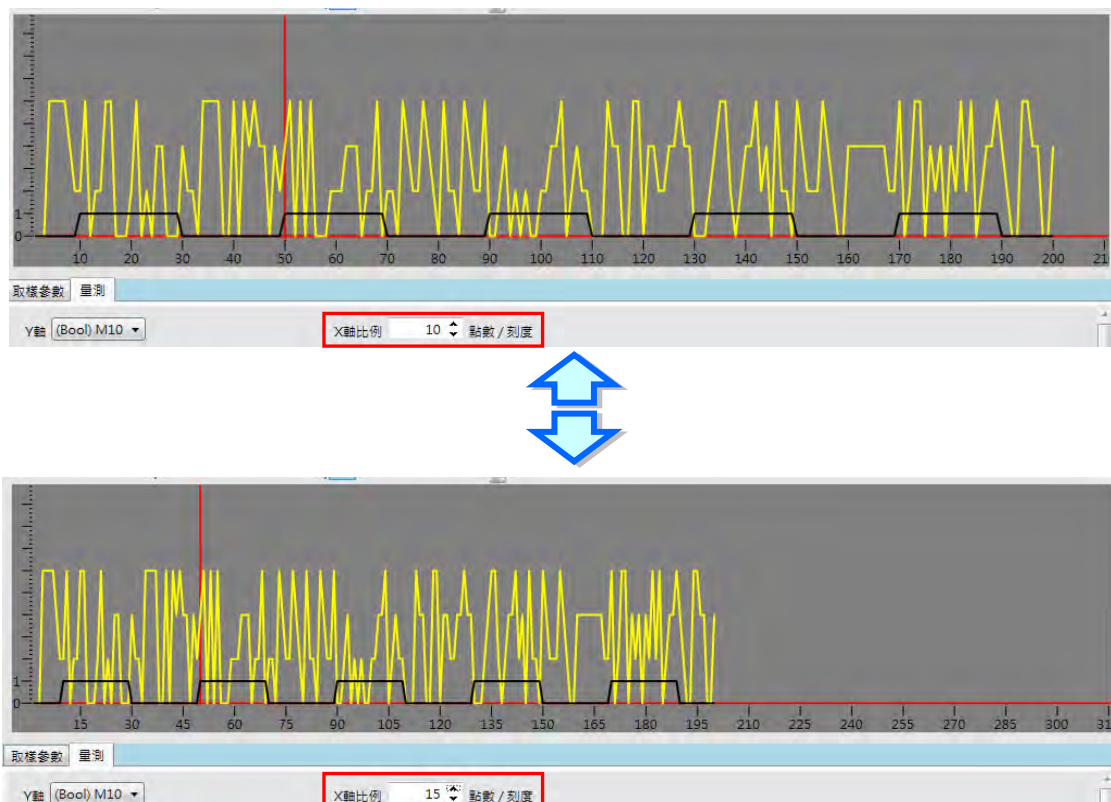
- 連續觸發：依據前述的觸發位置設定，資料示波器將保持不斷讀取的狀態，而每次當觸發條件滿足時，將觸發位置前後的取樣資料，自動顯示在資料示波器上，欲觀看圖形時，須將連續觸發的按鈕關閉恢復為未按下的狀態。如下圖，因觸發位置設為 50，故當觸發條件 M10=ON 時，會將取樣來源依據取樣週期取樣繪製 200 筆，包括觸發位置前 49 筆與之後的 150 筆。波形中的紅色垂直線即代表觸發位置。



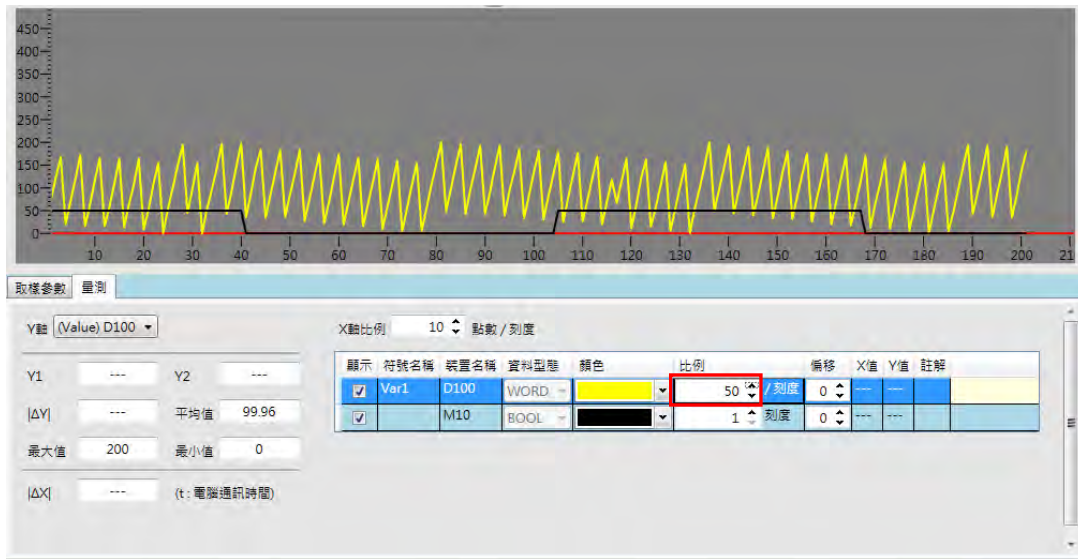
23

23.2.5 量測

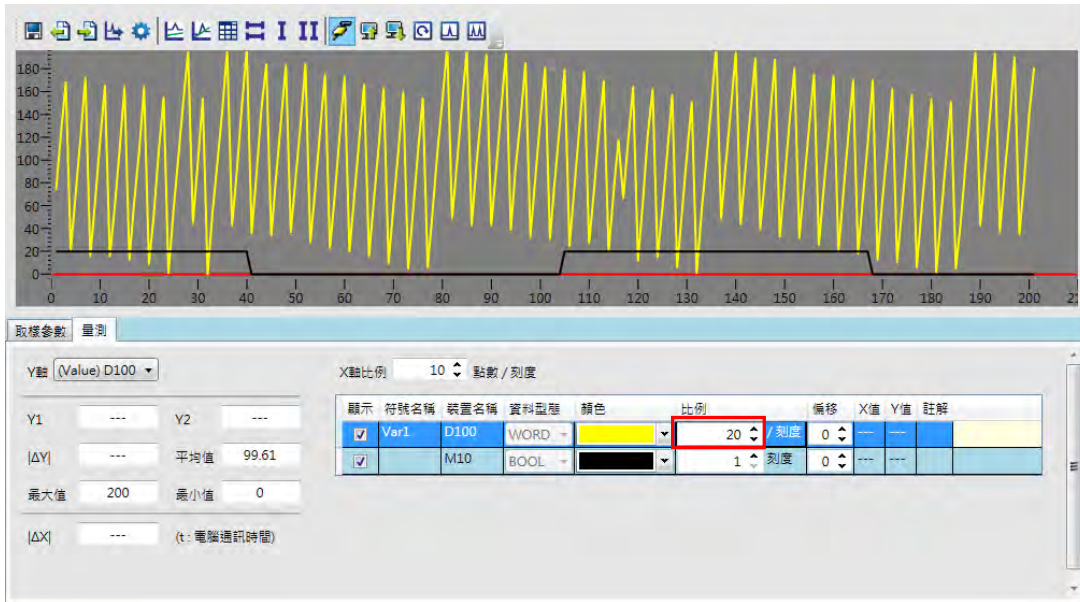
在量測頁面，可以調整資料示波器的波形外觀，以達更好的呈現效果。如下圖，變更 X 軸比例用以調整橫坐標每刻度的點數。



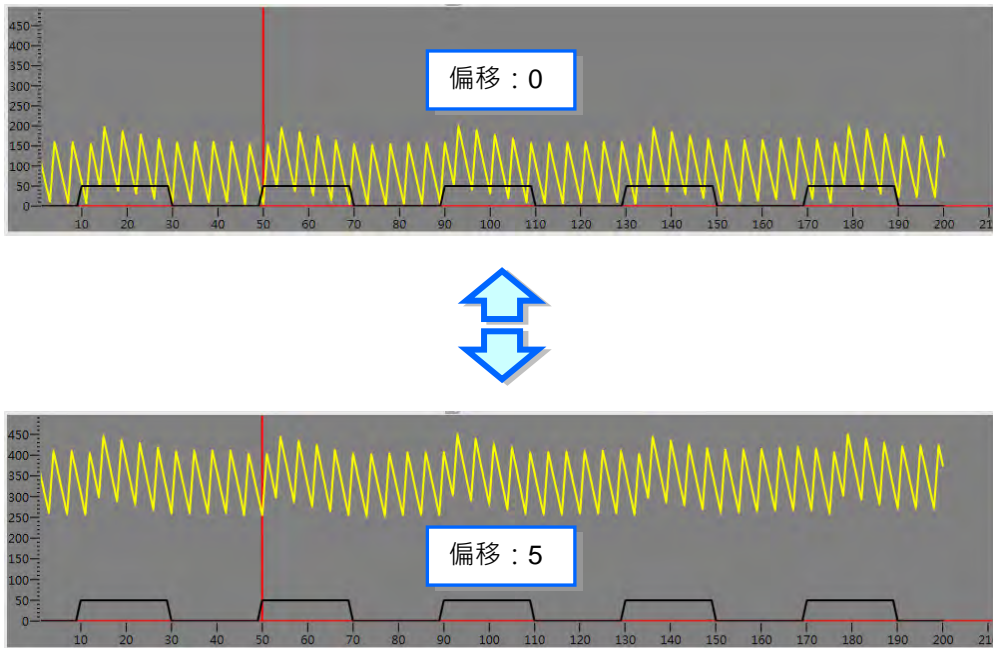
量測頁面的平均值、最大值、最小值為 Y 軸所選擇裝置波形的資訊。每個波形也可以設定不同的顏色、比例與偏移。比例為縱座標的比例，下圖為比例設為每刻度的值為 50 與 20 的比較。



23

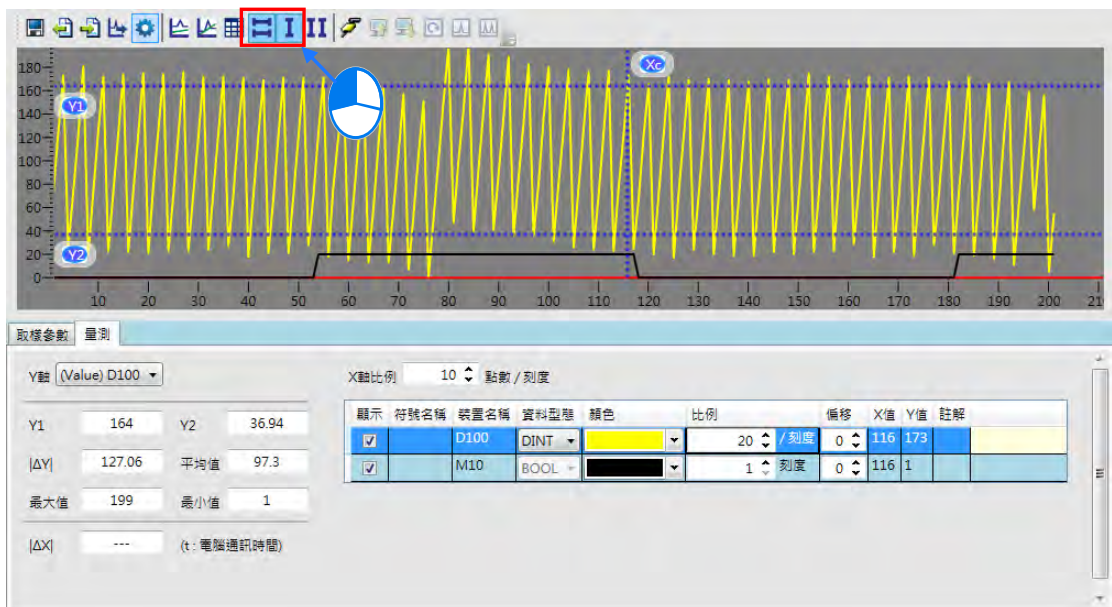


設定**偏移**，可調整波形在畫面中的位置，如下圖。

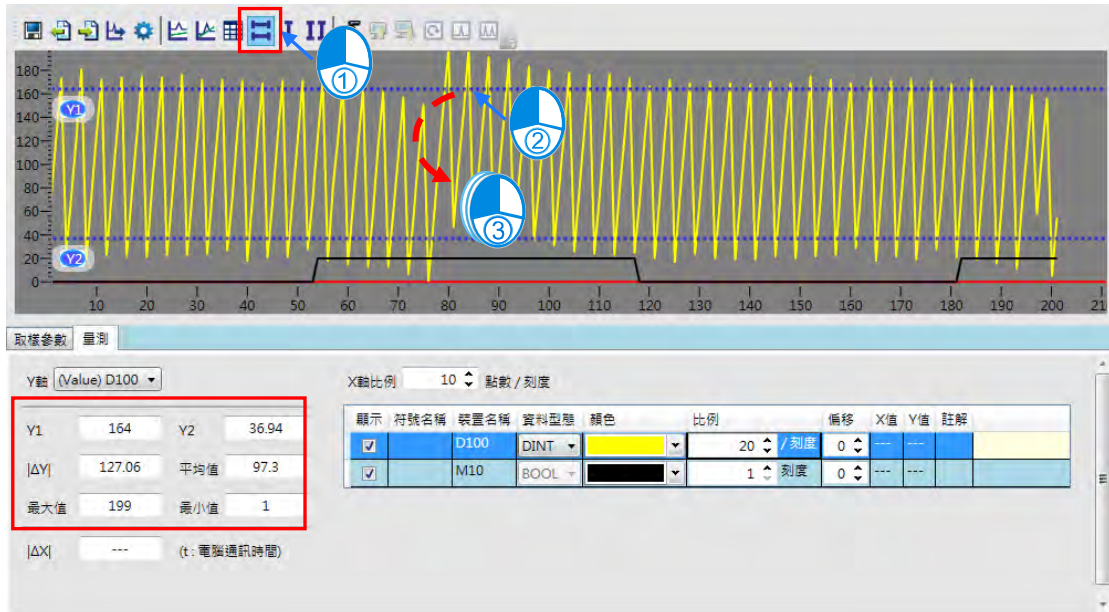


23

搭配水平量測線與垂直量測線，可以量測資料示波器的波形數值。如下圖，點選 按鈕開啟水平量測線，點選 按鈕開啟垂直量測線。

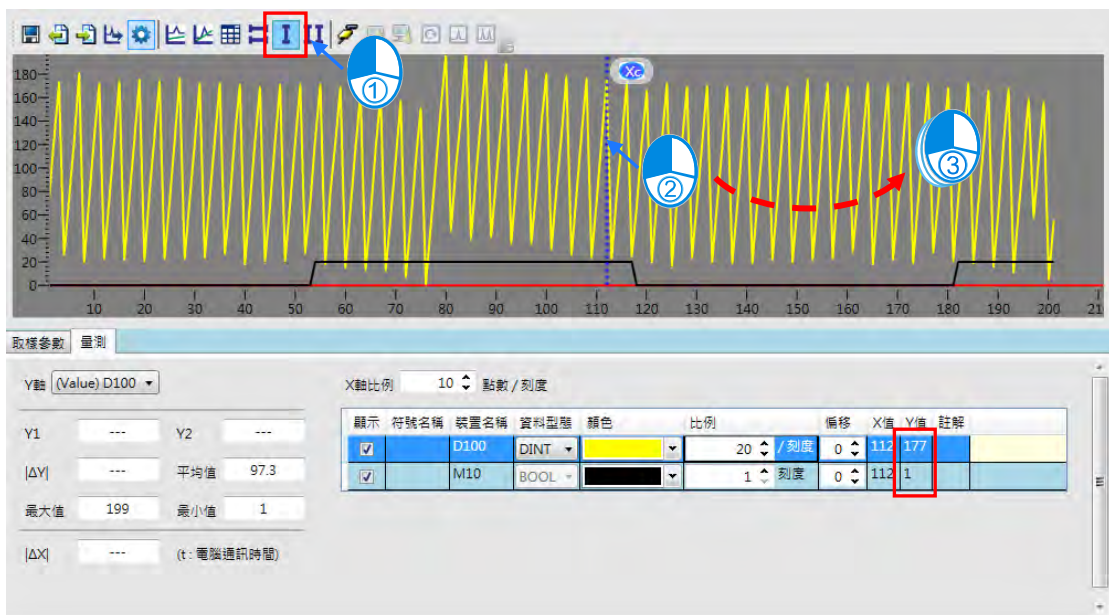


搭配水平量測線，可以拖曳兩條水平量測線 Y1 與 Y2，調整到要測量的位置。於參數設定區便會顯示水平量測線的各项數值，Y1 欄位代 Y1 的數值，Y2 欄位代表 Y2 的數值， $|\Delta Y|$ 欄位代表 Y1 與 Y2 的差值。

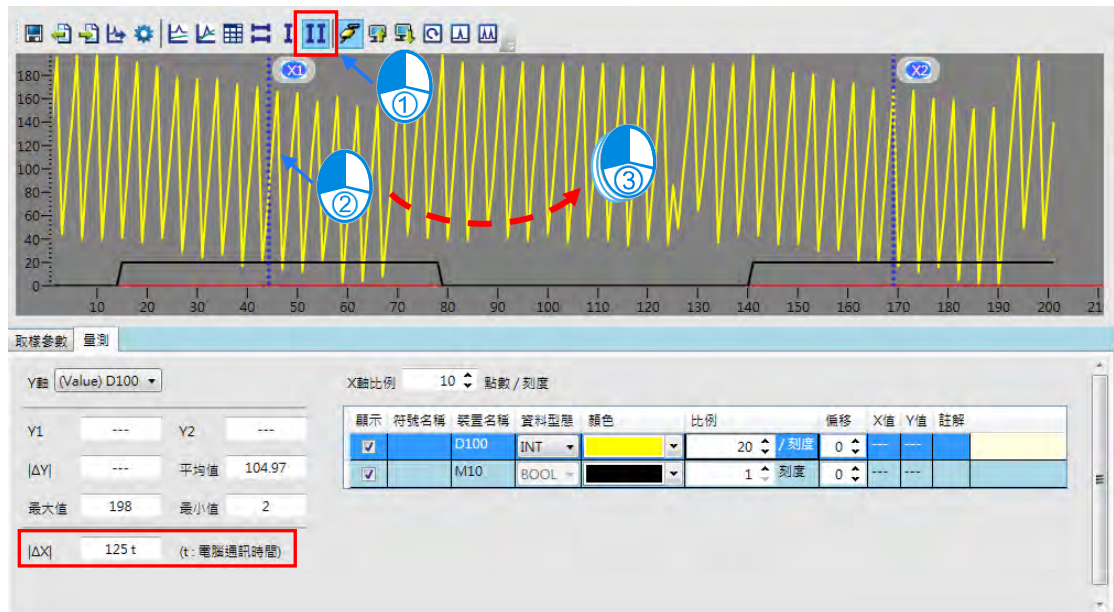


23

搭配垂直量測線，可以拖曳垂直量測線 Xc，則下方的 Y 值欄位，則會顯示各取樣來源與 Xc 交叉所對應的 Y 值。如下圖。



搭配雙垂直量測線，可以量測資料示波器的時間值。如下圖，點選 **II** 按鈕開啟雙垂直量測線，以滑鼠點選後拖曳兩條垂直量測線 X1 與 X2，調整到要測量的位置，於下方的區域便會顯示 $|\Delta X|$ 的時間值（t 值為電腦與 PLC 兩者一次通訊的時間）。



23

23.3 資料記錄器

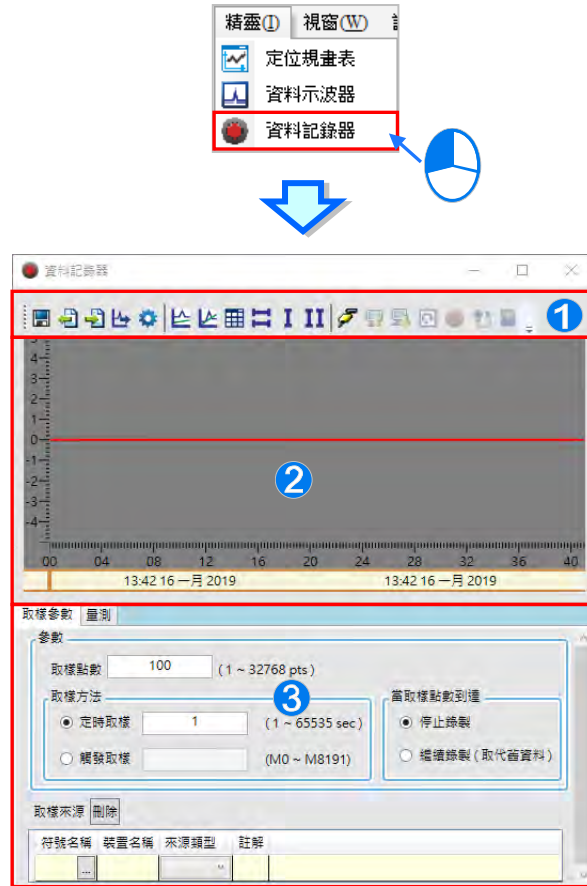
23.3.1 關於資料記錄器

資料記錄器用於長期記錄，變數符號或裝置的數值與狀態，以波形圖顯示，並且可儲存於主機內部與記憶卡中，方便分析數值趨勢。

23.3.2 開啟資料記錄器

要使用此功能時，請先編譯目前專案後，再點選 **精靈 (!)** > **資料記錄器**，開啟資料記錄器視窗，如下圖。

23





- ❶ 圖示工具列：提供各項操作資料記錄器的按鈕，功能說明如下表。
- ❷ 波形顯示區：顯示即時變數符號或裝置的波形圖。
- ❸ 參數設定區：取樣參數分頁用於設定取樣的來源與參數。量測分頁會顯示量測工具所量測到的數值及調整波形的顏色與比例等。

圖示工具列按鈕功能說明

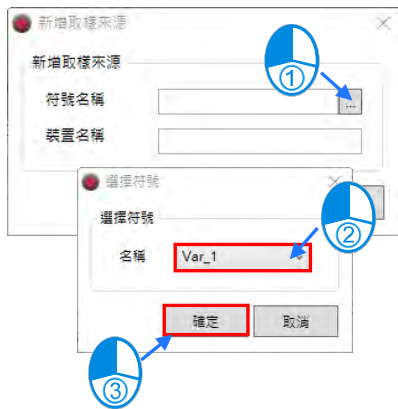
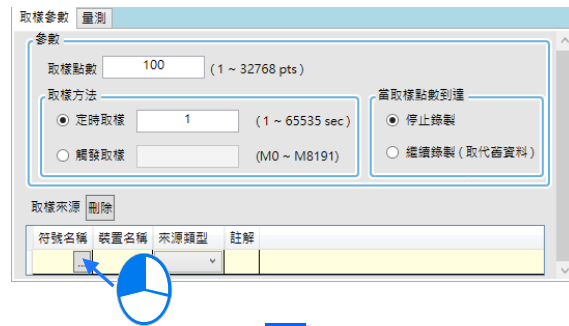
圖示	名稱	功能說明
	儲存全部	將取樣設定資料與取樣波形資料結果儲存到指定路徑中。
	匯出取樣設定	將取樣設定內容匯出到指定路徑。
	匯入取樣設定	由指定路徑將取樣設定內容匯入。
	匯出波形資料	將取樣波形資料以.bmp 或.csv 格式匯出到指定路徑。
	設定	設定資料記錄器波形與環境的顏色。
	分離波形	將多條波形自動以間隔分離。

圖示	名稱	功能說明
	重疊波形	將多條波形以中心線重疊。
	顯示格線	切換是否顯示格線。
	水平量測線	波形顯示區將出現兩條水平線，供使用者量測波形。
	垂直量測線	波形顯示區將出現一條垂直線，供使用者量測波形。
	雙垂直量測線	波形顯示區將出現兩條垂直線，供使用者量測波形。
	連線/離線	切換電腦與主機的連線或離線模式。
	上載 PLC 取樣設定	從主機讀回資料記錄器的取樣設定。
	下載取樣設定到 PLC	將資料記錄器的取樣設定寫入主機。
	即時監看	電腦即時從主機讀取來源的取樣波形。
	錄製資料	按下後開始錄製，再次按下後停止錄製。
	上載 PLC 記錄	由主機讀回資料記錄器的記錄值
	寫入記憶卡	將主機的記錄數值寫入到記憶卡中

23.3.3 設定取樣參數

在取樣參數分頁中，**取樣點數**用於設定取樣次數；**取樣方法**可以分成**定時取樣**與**觸發取樣**兩種方式，當使用**定時取樣**時，須設定取樣週期，則每隔一週期取樣一次；當使用**觸發取樣**時，須設定觸發裝置，當觸發裝置 ON 時，取樣一次，並且在取樣完成後，自動將觸發裝置 OFF，當取樣點數次數到達時，可以設定**停止錄製**或**繼續錄製 (取代舊資料)**。要新增取樣符號，先點選符號名稱的  按鈕，接著在彈出的視窗點選  按鈕，選擇要加入的變數符號，或直接在**裝置名稱**欄位輸入裝置名稱，按下「**新增**」後，就會加入取樣來源列表中，如下圖。

23

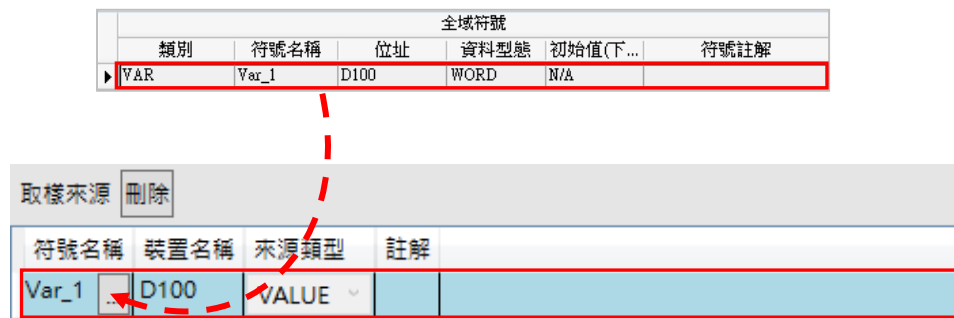


OR





符號名稱	裝置名稱	來源類型	註解
Var_1	D100	VALUE	

只有宣告在全域符號表的變數，並且編譯過專案，才可以這邊被選取，若有手動設定裝置位址或符號註解，也會一併帶入。




要刪除已加入的取樣來源時，點選要刪除的項目後，按下「刪除」按鈕，即可把該項目刪除。



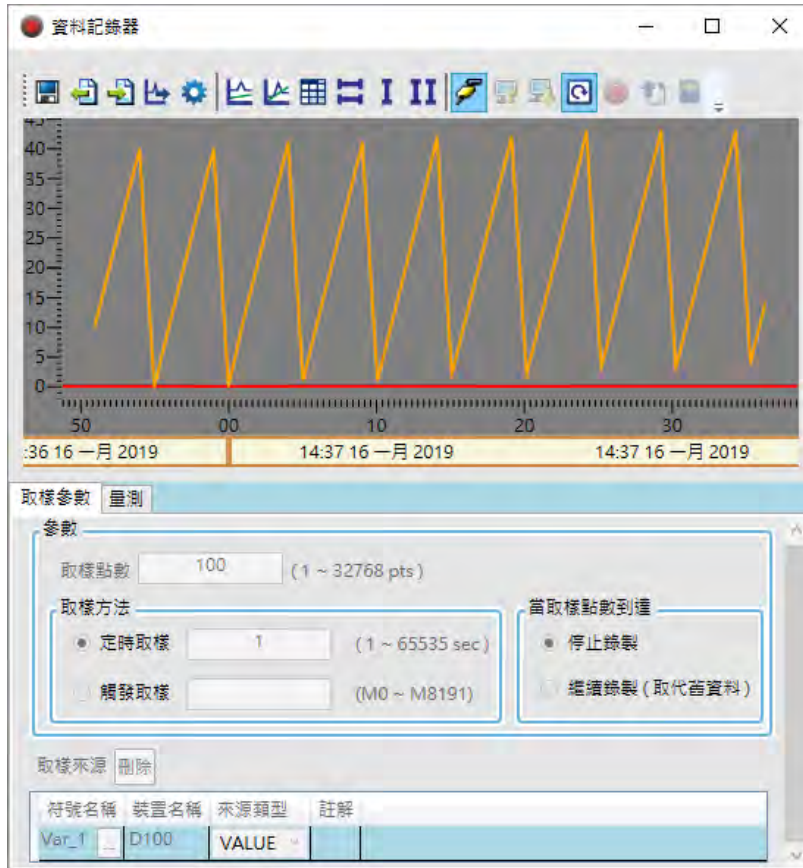
完成取樣參數的設定後，可按下圖示工具列的「匯出」，將取樣參數設定保存成.dlg 檔，之後可以在新專案使用「匯入」將取樣參數設定還原。

23.3.4 監看與錄製模式

完成資料記錄器的取樣參數設定後，在連線模式時，先下載參數設定，便可以使用圖示工具列的按鈕切換即時監看或錄製資料。



- 即時監看：依據設定的取樣週期，將取樣來源的變數符號或裝置的數值或狀態不斷繪製到資料記錄器上。畫面會隨波型自動調整到最適範圍，只保留最新的取樣點數個的資料。主機在此模式不會記錄取樣來源。



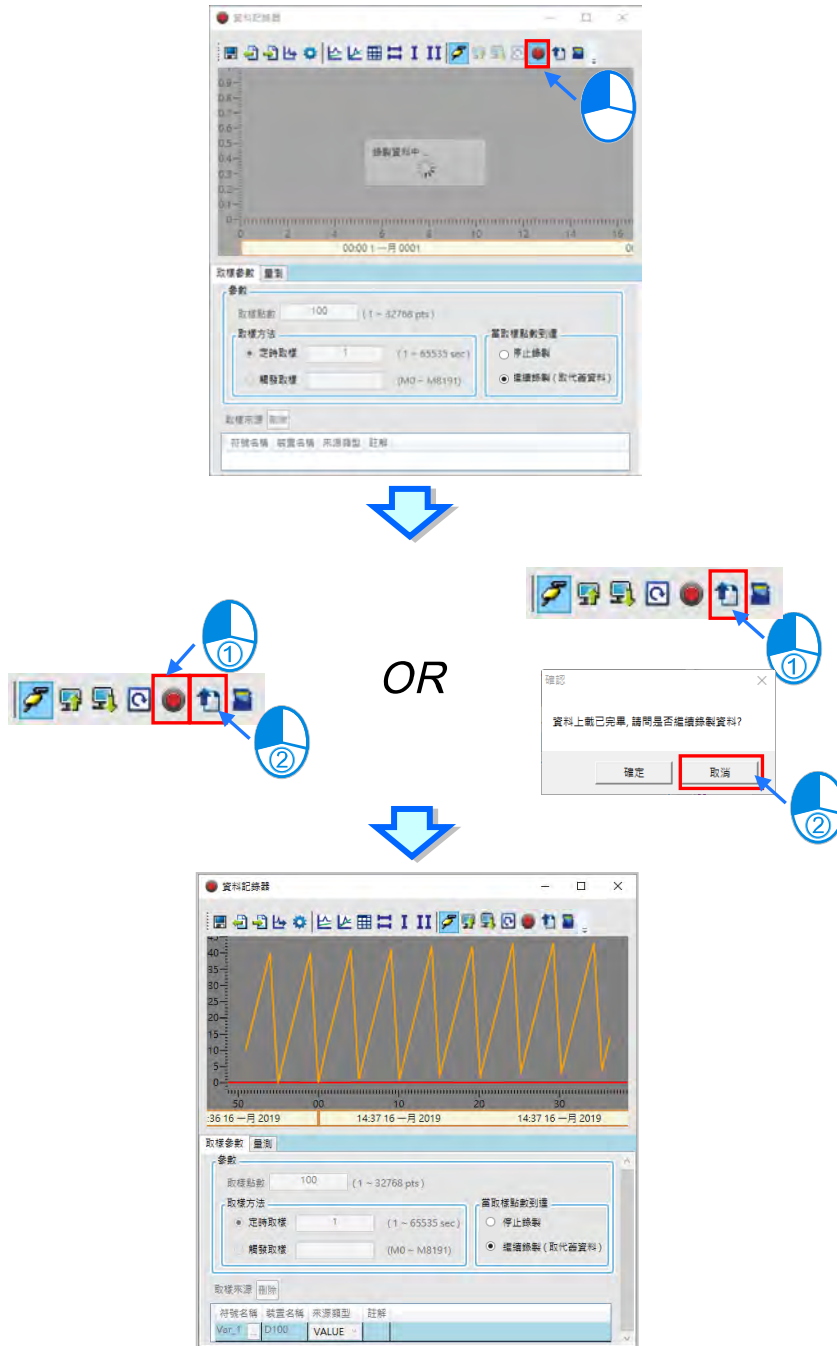
23


- 錄製資料：錄製取樣來源的變數符號或裝置數值與狀態，並儲存在主機的內部記憶體中。按下「錄製資料」開始錄製。使用取樣點數到達停止錄製，當設定的取樣點數到達後，便會彈出確認視窗，詢問是否上載錄製記錄，此時點選確定，接著會彈出確認視窗，確認是否再次錄製，點選取消後便會將上載的資料顯示在波形顯示區中。



若使用取樣點數到達**繼續錄製 (取代舊資料)**時，主機便會不斷錄製取樣來源資料，並保留最新的取樣點數範圍的資料，可以按下「**錄製資料**」，待停止錄製後再按下「**上載 PLC 記錄**」，或是在不停止錄製時直接按下「**上載 PLC 記錄**」接著會彈出確認視窗，確認是否再次錄製，點選取消後，便會將主機錄製的資料顯示在波形顯示區中。

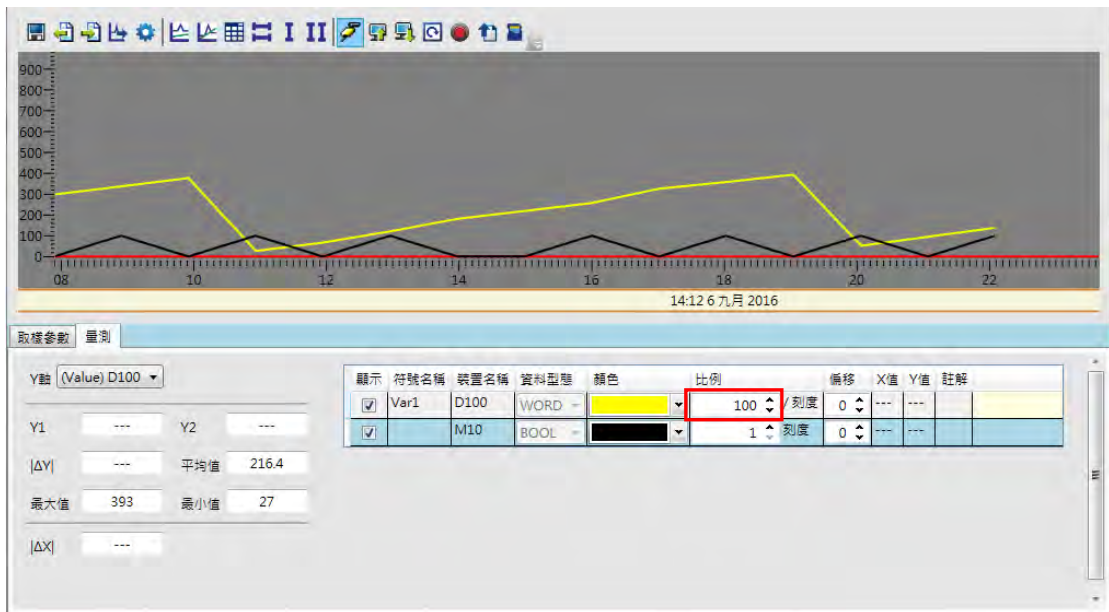
23



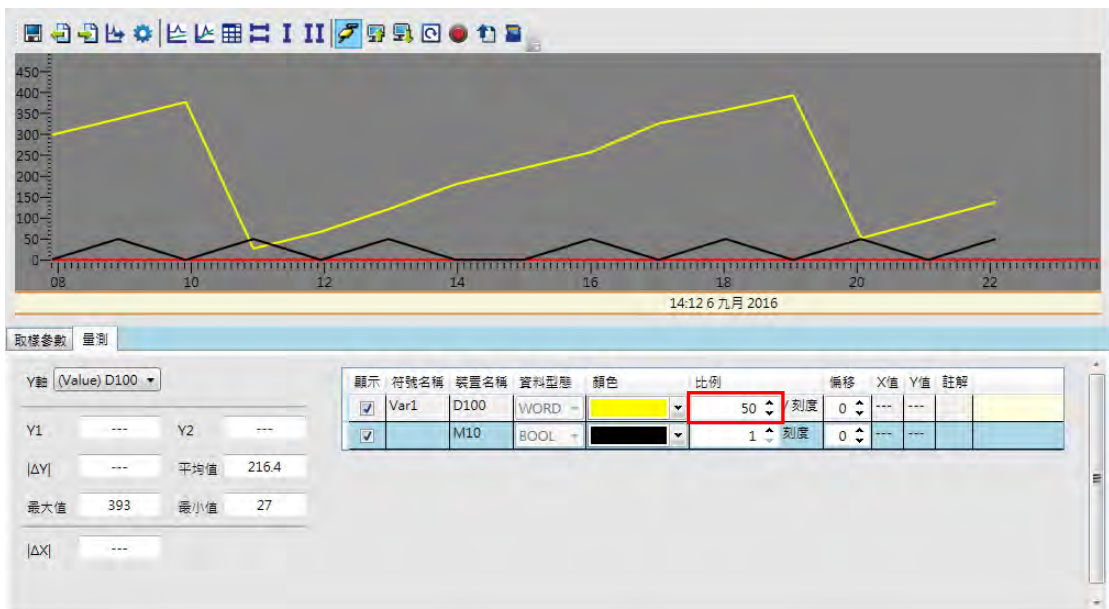
不論是否正在錄製資料，使用者可以按下「**寫入記憶卡**」，將主機目前的錄製資料寫入到**記憶卡根目錄\PLC CARD\機種\Log\DATA_LOGGER_年月日_時分秒**。

23.3.5 量測

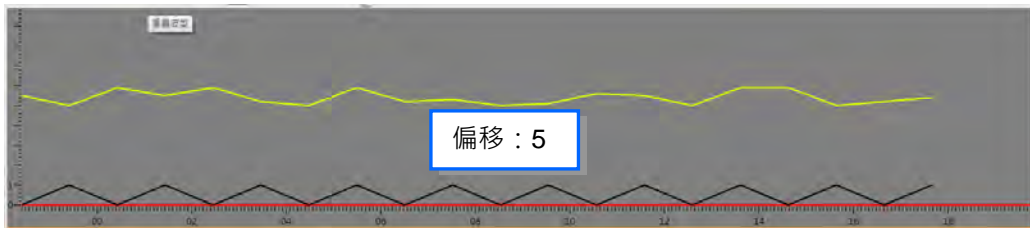
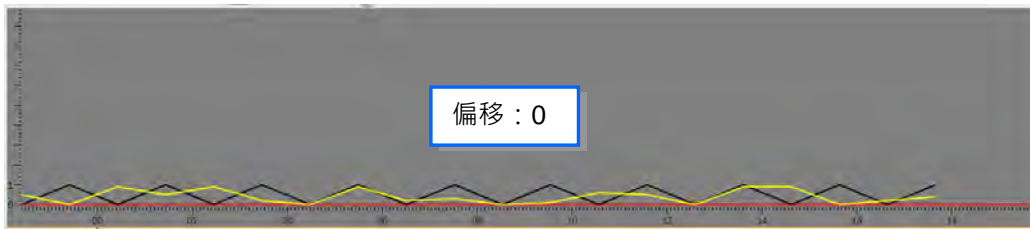
在量測頁面的平均值、最大值、最小值為 Y 軸所選擇裝置波形的資訊。每個波形也可以設定不同的顏色、比例與偏移。比例為縱座標的比例，下圖為比例設為每刻度 100 與 50 的比較。





23

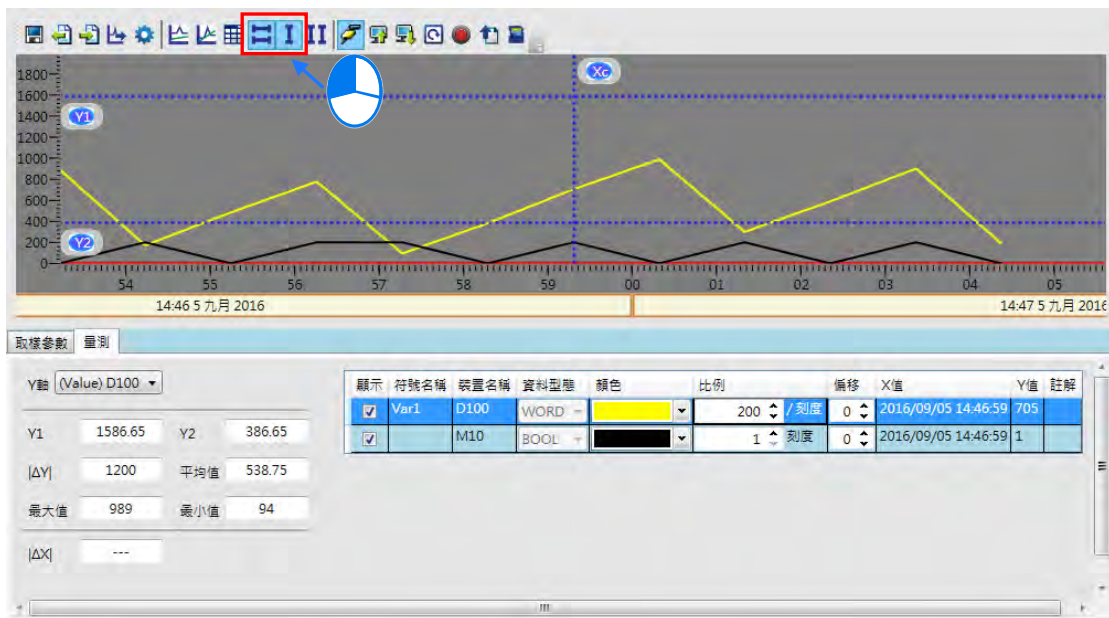


設定**偏移**，則可調整波形在畫面中的位置，如下圖。

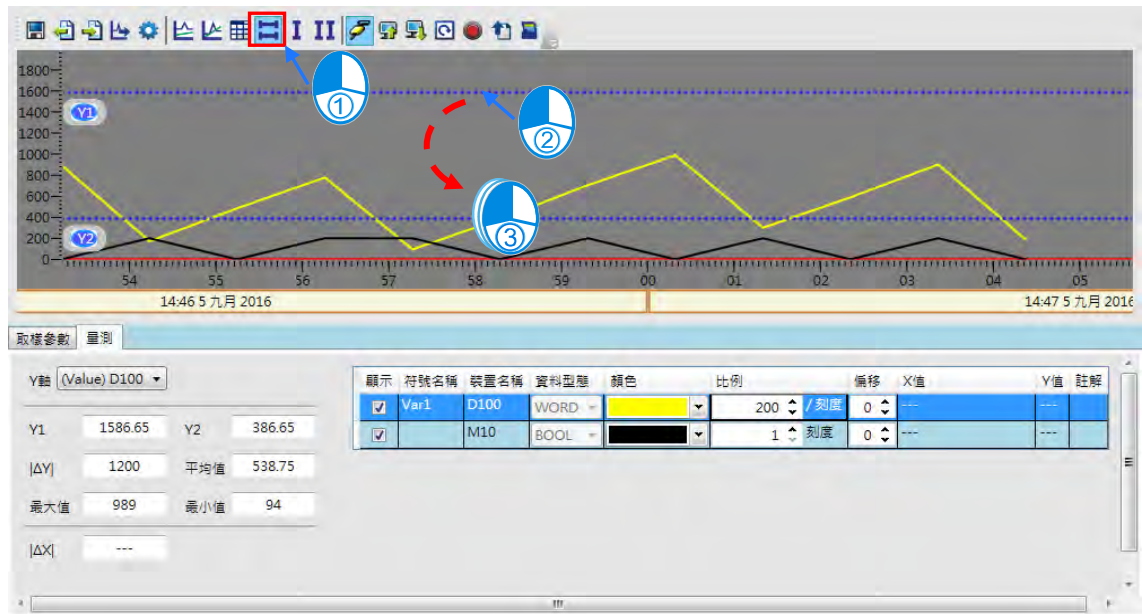


23

切換到**量測**頁面，搭配水平量測線與垂直量測線，可以量測資料記錄器的波形數值。如下圖，點選  按鈕開啟水平量測線，點選  按鈕開啟垂直量測線。



使用水平量測線，可以拖曳兩條水平量測線 Y1 與 Y2，調整到要測量的位置。於參數設定區便會顯示水平量測線的各項數值，Y1 欄位代 Y1 的數值，Y2 欄位代表 Y2 的數值，|ΔY| 欄位代表 Y1 與 Y2 的差值。

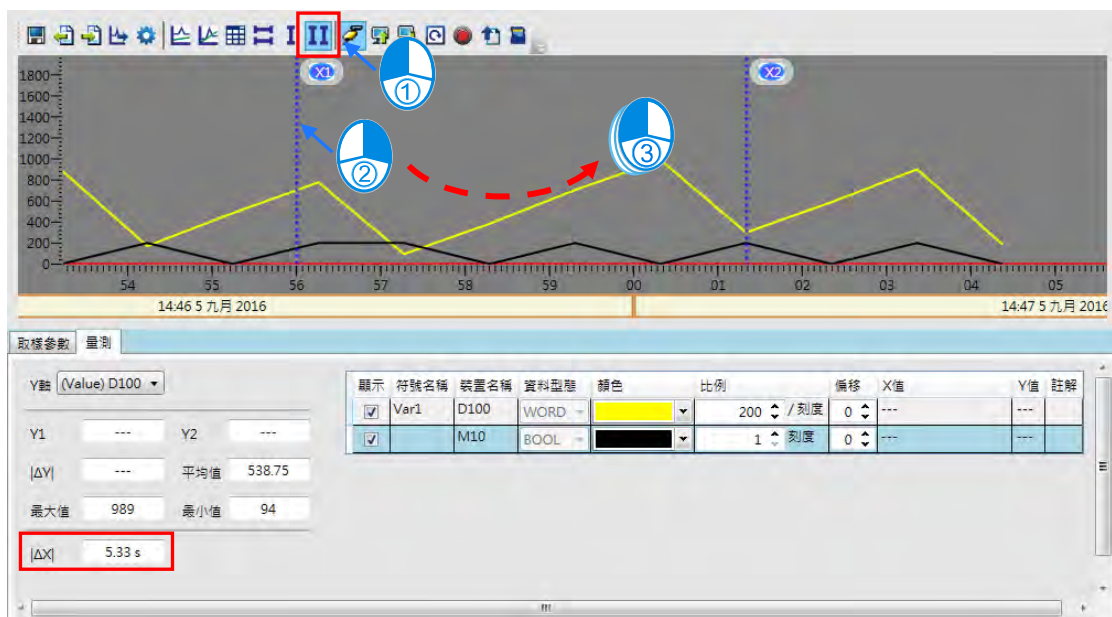


23

使用垂直量測線，可以拖曳垂直量測線 Xc，則下方的 Y 值欄位，則會顯示各取樣來源與 Xc 交叉所對應的 Y 值。如下圖。



使用雙垂直量測線，可以量測資料示波器的時間值。如下圖，點選 **II** 按鈕開啟雙垂直量測線，以滑鼠點選後拖曳兩條垂直量測線 X1 與 X2，調整到要測量的位置，於下方的區域便會顯示 ΔX 的時間值。



23

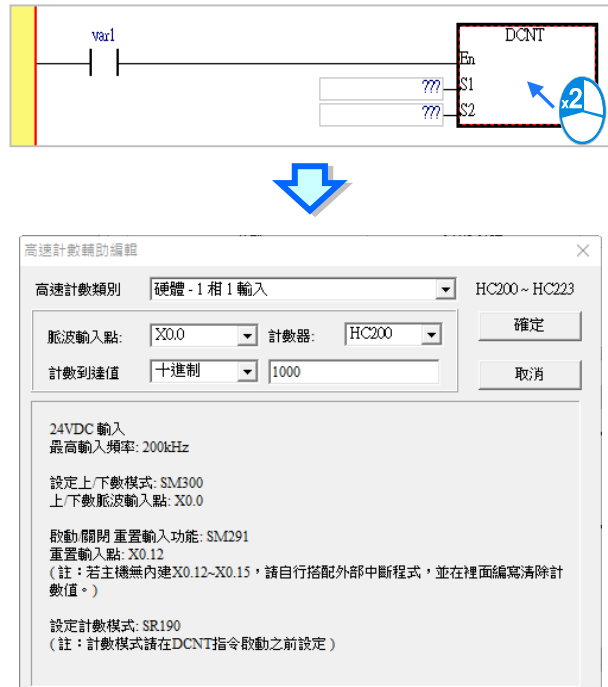
23.4 高速計數器精靈

23.4.1 關於高速計數器精靈

高速計數器精靈提供使用者依據高速計數器的類型、計數的形式、輸入點編號選擇對應的高速計數器裝置編號，便於使用者快速的設定 DCNT 指令。

23.4.2 使用高速計數器精靈

於階梯圖中的 DCNT 指令上，雙擊滑鼠左鍵，開啟高速計數器精靈設定視窗。如下圖。



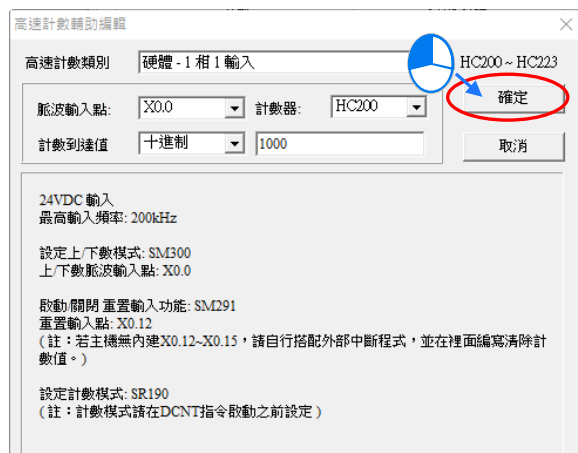
先選擇高速計數器的類別，包括**硬體 1 相 1 輸入**、**硬體 1 相 2 輸入**...等等，接著選取脈波輸入點的位置跟計數器的裝置。如下圖。



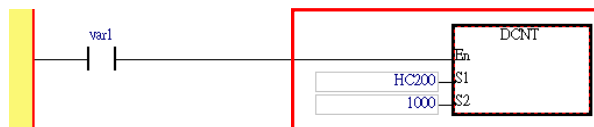
再來要輸入計數的數值。可以選取十進制、十六進制或使用 D 裝置，接著於右方欄位輸入數值或 D 裝置編號。



最後按下「確定」，高速計數器編號與計數器設定值將會被填入 DCNT 指令中。



23



23.5 NTC 精靈

23.5.1 關於 NTC 精靈

NTC 精靈為 DVP08NTC-S 模組的專用精靈軟體，提供使用者編輯自定義分度表、csv 檔匯入/匯出、一次上載/下載多張表、一次下載多台模組等功能，詳細操作方式可參考該模組的安裝手冊。

23.5.2 使用 NTC 精靈

要使用此功能時，請點選 **精靈 (!)** > **NTC 精靈**，開啟 NTC 精靈視窗，如下圖。



- ❶ 選項包含
 - 檔案：檔案儲存、讀取，分度表匯入、匯出功能。
 - 編輯：載入預設值、清除分度表。
 - 設定：通訊設定，RS-485 直連模組或是透過主機連接模組。
 - 說明：操作說明、軟體資訊說明。
- ❷ 輸入起始溫度、溫度間距、溫度個數，可以產生一張空白的表格。
- ❸ 分度表表格，可以填寫對應的電阻值，共八張分度表可供填寫，使用下方的頁籤切換。
- ❹ 分度表折線圖，依照左邊的表格產生出來的折線圖。
- ❺ 上下載分度表選擇，選擇欲上載/下載的分度表。

MEMO

23



附錄A USB 連線

目錄

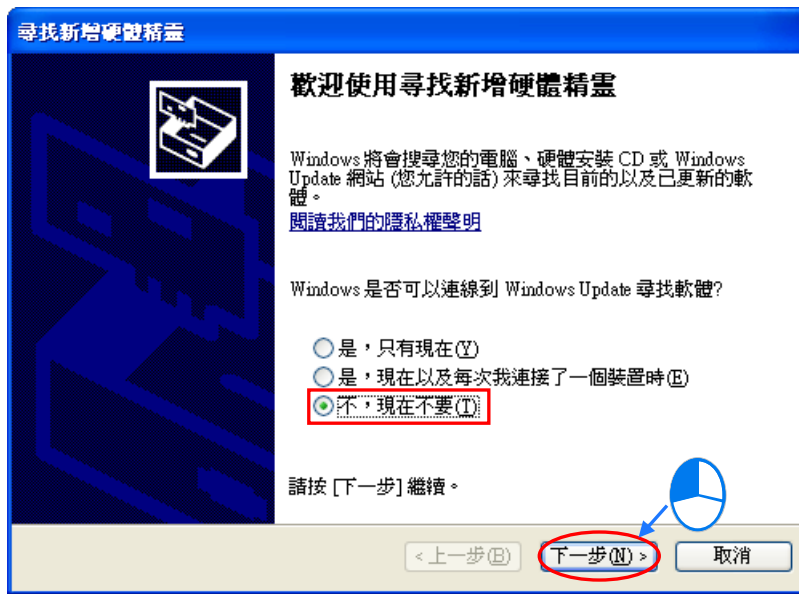
A.1 安裝 PLC 的 USB 驅動程式	A-2
A.1.1 在 Windows XP SP3 環境下安裝 USB 驅動程式	A-2
A.1.2 在 Windows 7 環境下安裝 USB 驅動程式.....	A-6
A.1.3 在 Windows 8.1 環境下安裝 USB 驅動程式	A-11
A.1.4 在 Windows 10 環境下安裝 USB 驅動程式	A-14
A.2 於 COMMGR 中建立 USB 型態的 Driver	A-17
A.3 DVP-SX2 系列的 USB 設定	A-19

A.1 安裝 PLC 的 USB 驅動程式

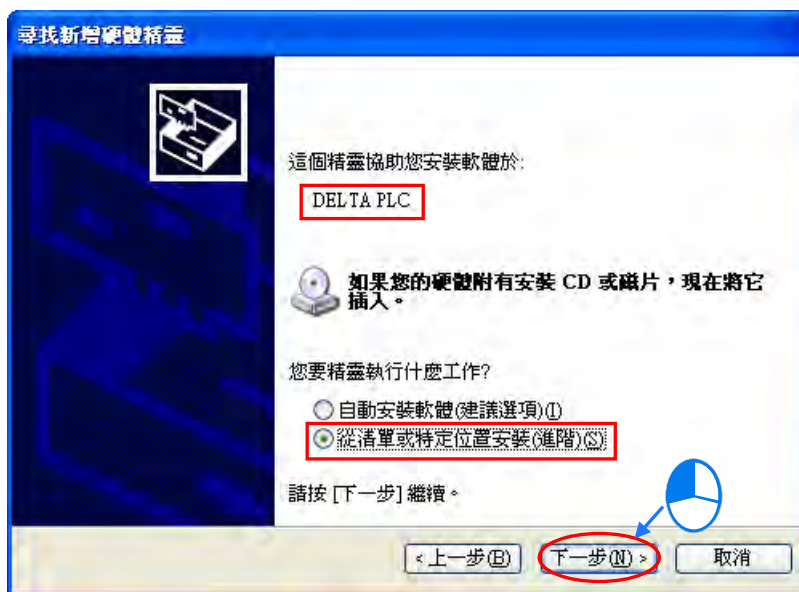
A.1.1 在 Windows XP SP3 環境下安裝 USB 驅動程式

下列將以 Windows XP 為例，逐步示範如何安裝 PLC 的 USB 驅動程式，至於其他作業系統則請自行參考該作業系統中有關新硬體安裝的相關說明。

- (1) 確認 PLC 主機已正常上電，並透過 USB 電纜將 PLC 主機連接至個人電腦的 USB 埠，此時畫面會出現新增硬體的提示視窗，選擇「不，現在不要 (I)」的選項之後點擊「下一步 (N)」。



- (2) 接著視窗中將會顯示偵測到的 USB 裝置名稱，下圖為 AH500 機種的裝置名稱，而其它機種所顯示的名稱可能會有不同。此步驟請選擇「從清單或特定位置安裝 (進階) (S)」。



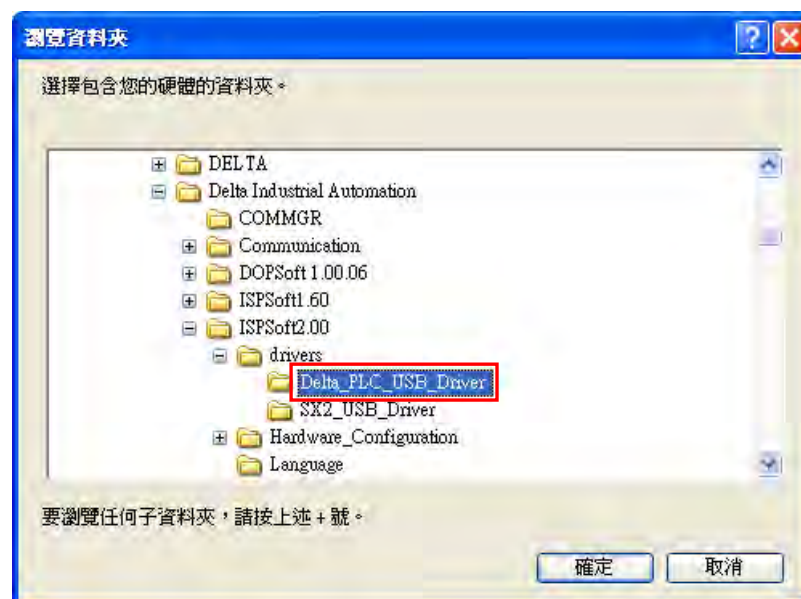
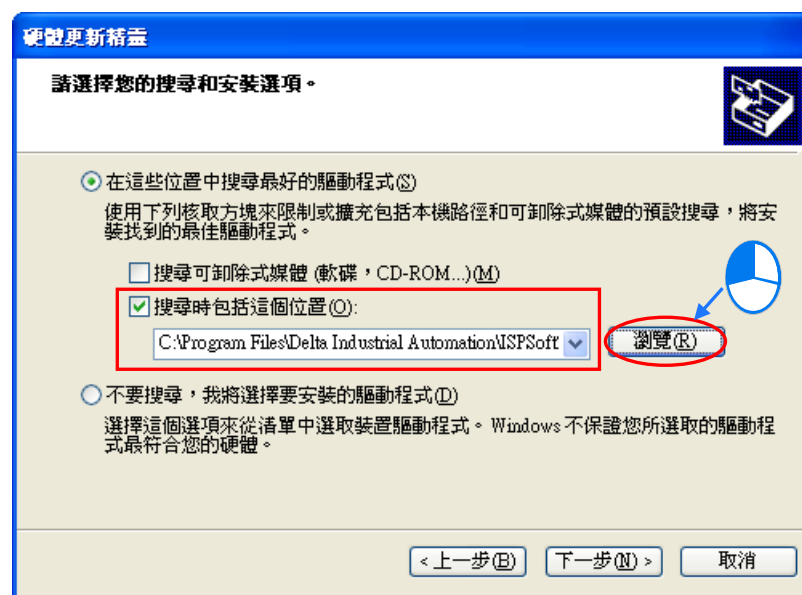
A

(3) 下表為台達目前內建 USB 通訊介面的 PLC 機種，而在安裝完 V2.00 版本以上的 ISPSOft 之後，各機種的驅動程式亦會被放置在下表的路徑中。

機種	驅動程式路徑
SX2、AH560EN2	ISPSOft 安裝路徑\drivers\SX2_USB_Driver\
SE	ISPSOft 安裝路徑\drivers\Delta_PLC_USB_Driver\
AH500 系列	ISPSOft 安裝路徑\drivers\Delta_PLC_USB_Driver\

*.ISPSOft 的預設安裝路徑為 C:\Program Files(x86)\Delta Industrial Automation\ISPSOftx.xx。

接著請根據所連接的 PLC 機種指定正確的驅動程式路徑，但若透過其他方式所取得的驅動程式，則請自行指定相對的路徑。完成後點擊「下一步 (N)」以繼續進行安裝步驟。



A

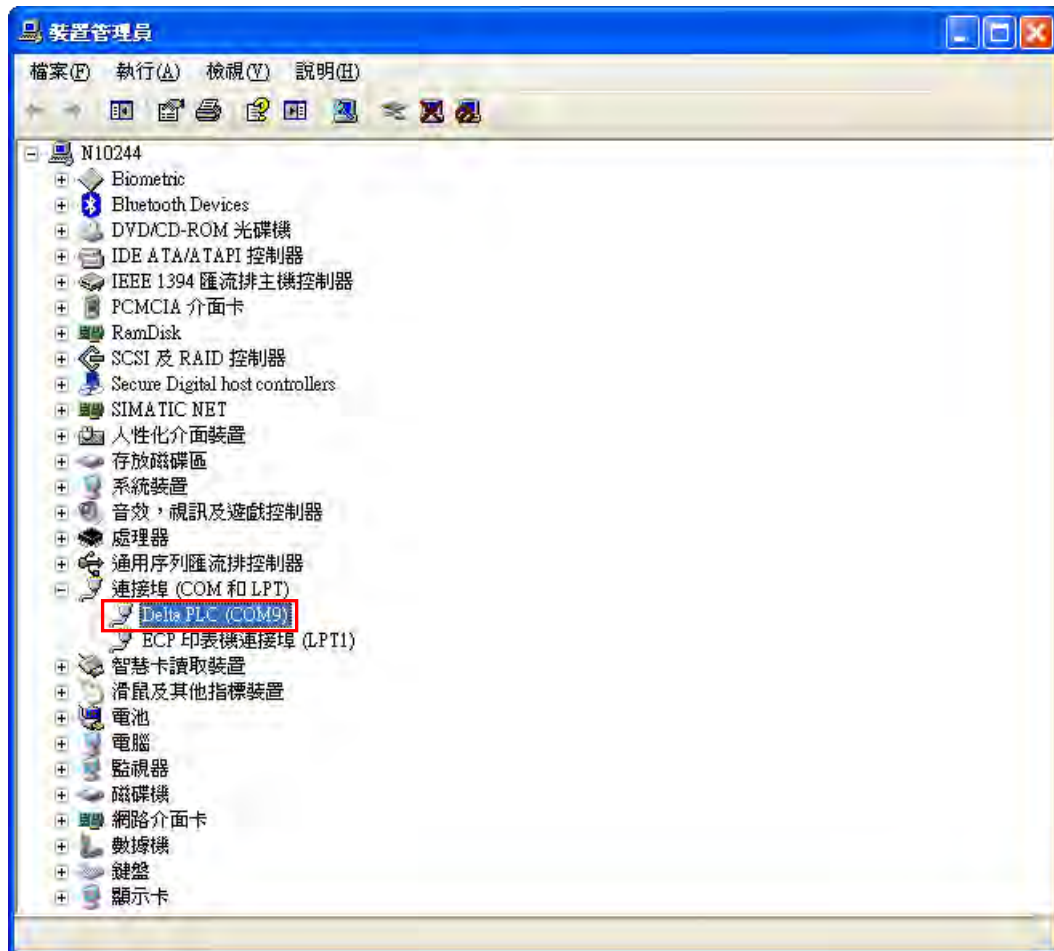
- (4) 當於指定的路徑中搜尋到正確的驅動程式之後，系統便會開始進行安裝，而安裝途中若出現未通過驗證測試的警告視窗時，請直接按下「繼續安裝 (C)」即可。



- (5) 安裝結束後，請按下「完成」鍵離開。



- (6) 完成安裝後，請先開啟 Windows 的裝置管理員，當連接埠 (COM 與 LPT) 的項目中有列出方才所安裝的裝置名稱時，便代表驅動程式已安裝成功，而此時 Windows 系統便會自動為其配置一個 COM Port 的編號。



*.上圖為 AH500 機種的裝置名稱，而其它機種所顯示的裝置名稱可能會有不同。

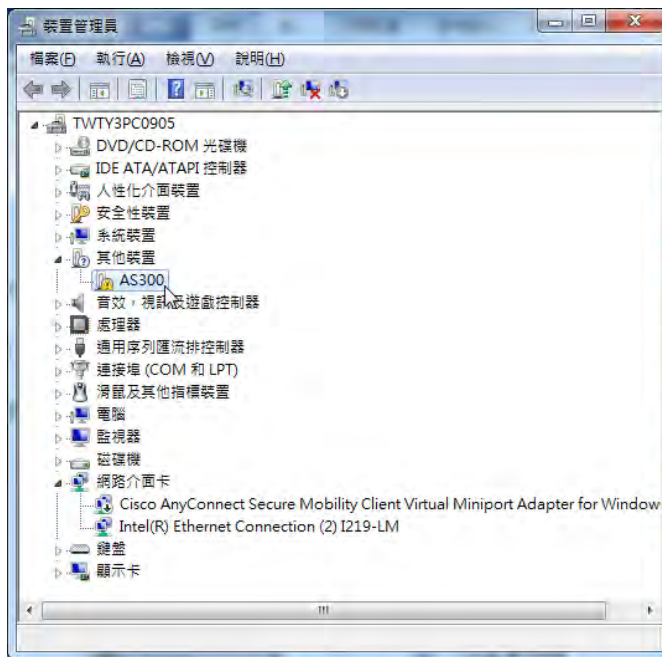
補充說明

- 當更換電腦端的 USB 埠口時，系統可能會要求重新安裝驅動程式，此時僅需依據前述步驟再次安裝驅動程式即可。而安裝完畢後，系統為其配置的 COM Port 編號亦可能會有不同。
- 若您的 Windows XP 尚未更新至 SP3 的版本時，在安裝過程中可能會出現缺少必要檔案的警示訊息，此時可採取如下的兩個處置方式。
 - (a) 取消安裝並更新 Windows XP 至 SP3 版，之後再依據前述的步驟重新安裝驅動程式。
 - (b) 自行取得所需的檔案後，於出現的警示視窗中指定該檔案的路徑。

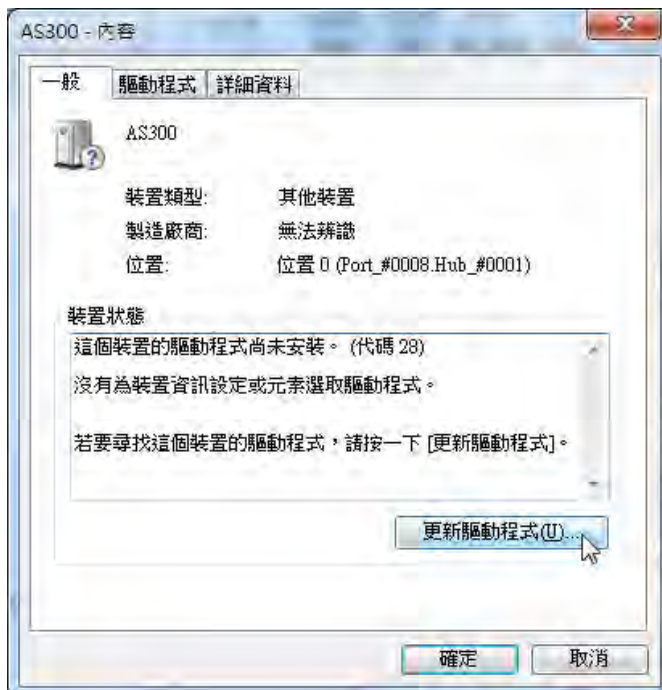
A.1.2 在 Windows 7 環境下安裝 USB 驅動程式

若使用者使用 Windows 7 作業系統，接下來將舉 AS300 系列逐步示範如何安裝 PLC 的 USB 驅動程式。

- (1) 確認 AS 主機已正常上電，並透過 USB 電纜將 AS 主機連接至個人電腦的 USB 埠。
- (2) 點選 控制台 > 裝置管理員 後，雙擊「其他裝置」選項下的「AS300」。



- (3) 在以下視窗中，點選「更新驅動程式」按鈕，如下所示。

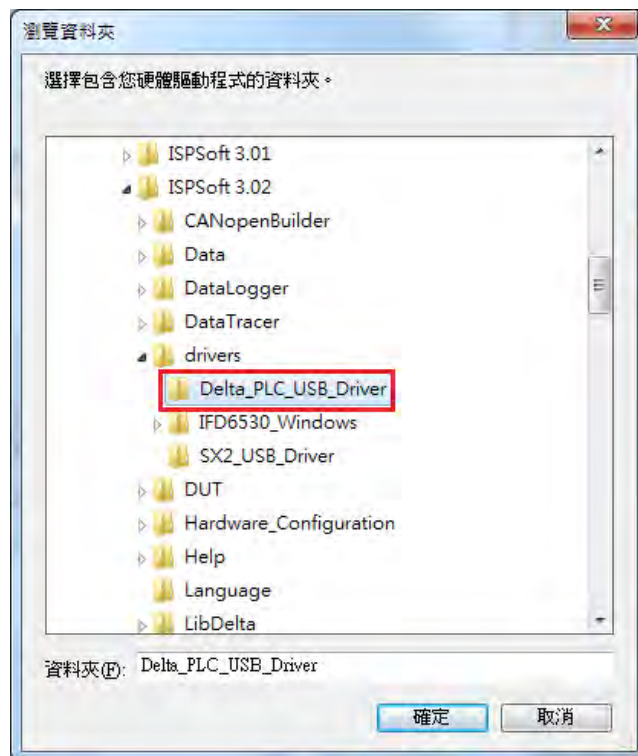
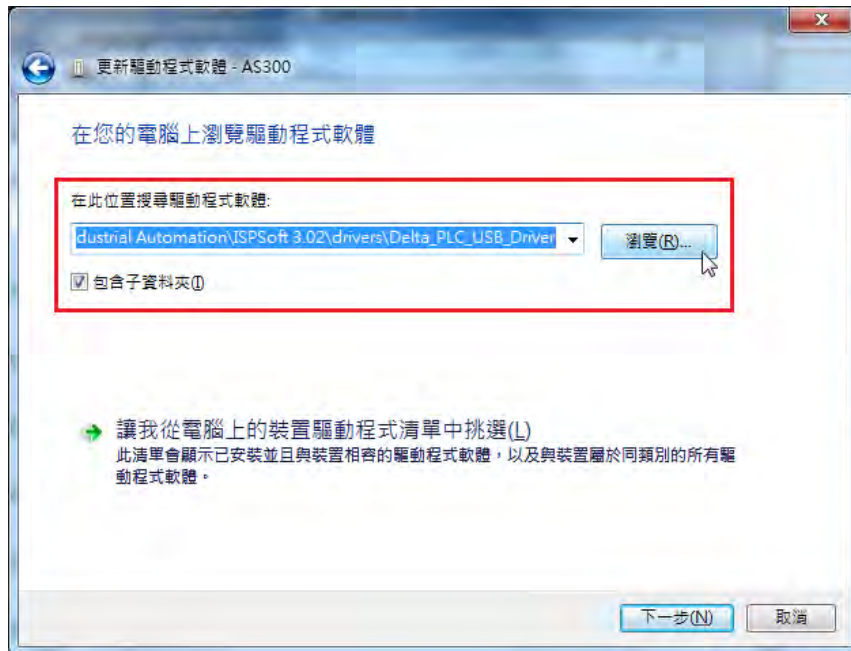


(4) 接著視窗中，請選擇「瀏覽電腦上的驅動程式軟體 (R)」。

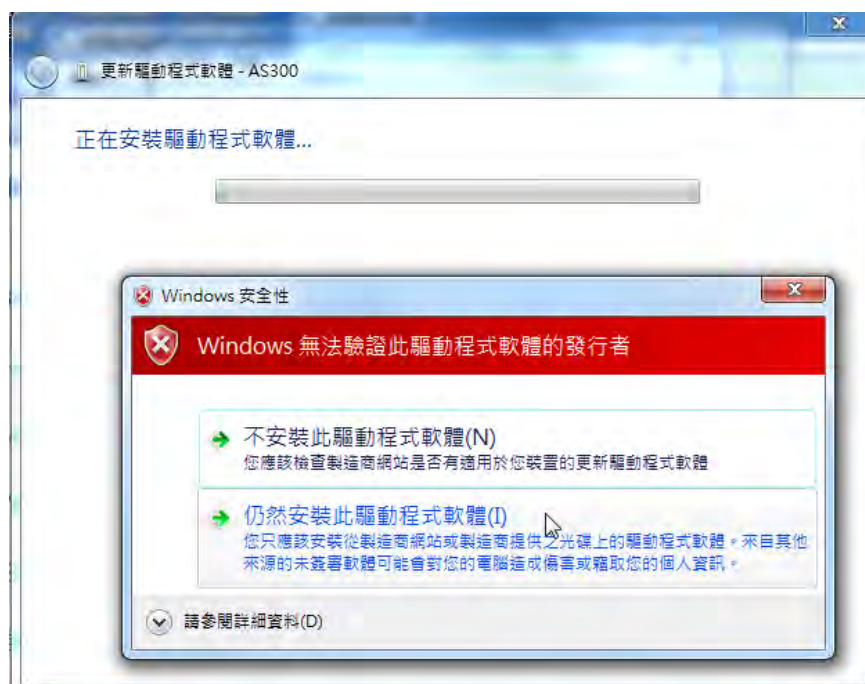


A

- (5) 在安裝完 V3.00 版本以上的 ISPSOft 之後，AS 主機的 USB 驅動程式會被放置在 ISPSOft 安裝路徑\drivers\Delta_PLC_USB_Driver\，若透過其他方式取得的驅動程式，請自行指定相關路徑。接著請指定正確的驅動程式路徑，但若透過其他方式所取得的驅動程式，則請自行指定相對的路徑。完成後點擊「下一步 (N)」以繼續進行安裝步驟。



- (6) 當於指定的路徑中搜尋到正確的驅動程式之後，系統便會開始進行安裝，而安裝途中若出現要求驗證測試的警告視窗時，直接按下「**仍然安裝此驅動程式軟體 (I)**」即可。

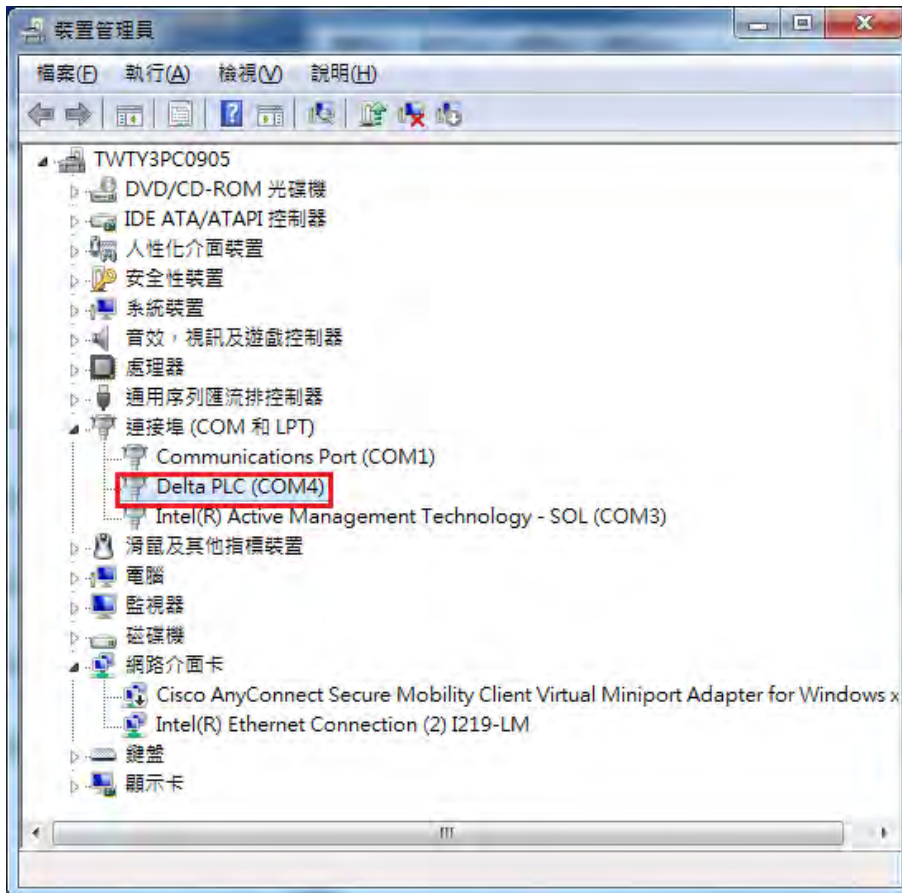


- (7) 安裝結束後，請按下「**關閉**」鍵離開。



A

- (8) 完成安裝後，在 Windows 的裝置管理員中的連接埠 (COM 與 LPT) 項目中，有列出方才所安裝的裝置名稱時，便代表驅動程式已安裝成功，而此時 Windows 系統便會自動為其配置一個 COM Port 的編號。



A

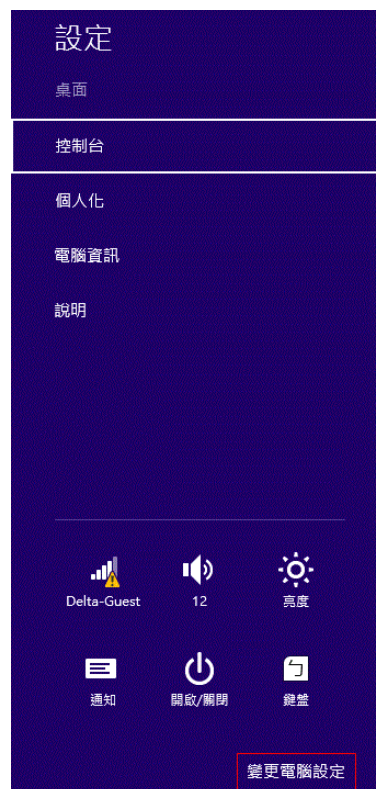
補充說明

- 當更換電腦端的 USB 埠口時，系統可能會要求重新安裝驅動程式，此時僅需依據前述步驟再次安裝驅動程式即可。而安裝完畢後，系統為其配置的 COM Port 編號亦可能會有所不同。

A.1.3 在 Windows 8.1 環境下安裝 USB 驅動程式

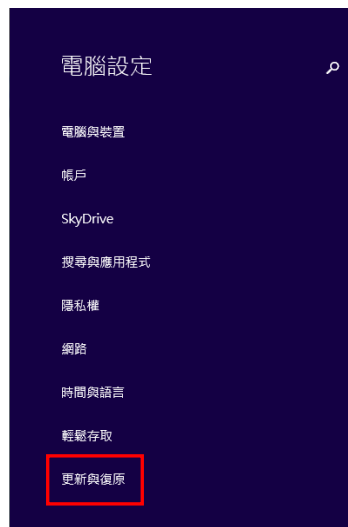
若使用者使用 Windows 8.1 作業系統，需先了解數位簽章的內容，數位簽章是一種可加入檔案的電子安全性標記。它可以讓您確認檔案的發行者，而且可以協助確認檔案自從數位簽章之後並未變更。因目前 DELTA PLC USB 驅動程式無數位簽章，接下來將逐步介紹如何關閉 Windows 8.1 作業系統的數位簽章功能，來達成電腦端可順利安裝 DELTA PLC USB 驅動程式，此設定僅限單次使用，當電腦重新開關機後，會回復原來的強制簽章模式。

(1) 按下鍵盤中  【WIN】+ 【I】，會顯示設定介面，點擊【變更電腦設定】。



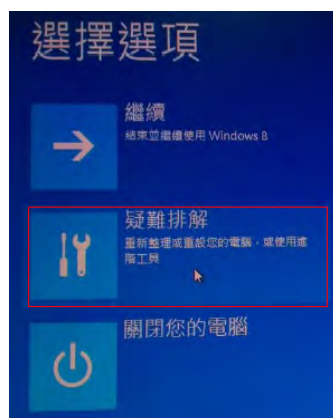
A

- (2) 在「電腦設定」畫面左半部底下選擇【更新和恢復】，在「更新和恢復」畫面選擇【恢復】，選完後選擇【立即重新啟動】。



A

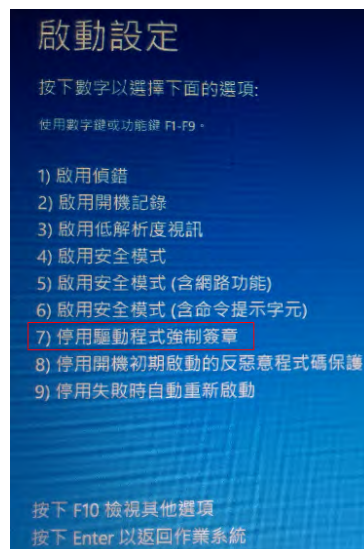
- (3) 待重新開機後於選擇選項畫面中點選【疑難排解】；在疑難排解畫面中點選【進階選項】。



(4) 進階選項畫面中點選【啟動設定】；在啟動設定畫面中點選【重新啟動】。



(5) 系統會進入另一啟動設定畫面，按下數字鍵 7 或 F7 執行停用驅動程式強制簽章，完成後系統會重新回到 Windows 8 作業畫面，此時即可安裝 DELTA USB 驅動程式。



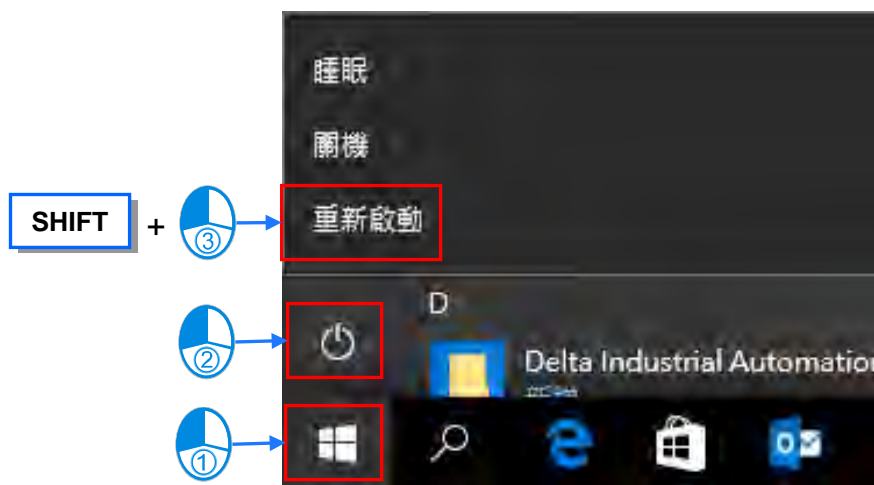
(6) 安裝 DELTA USB 驅動程式可參考 Windows 7 的安裝說明。

A

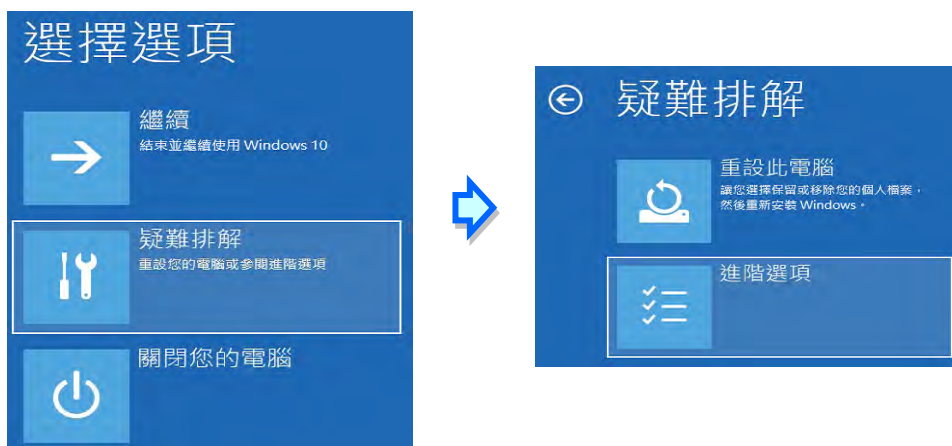
A.1.4 在 Windows 10 環境下安裝 USB 驅動程式

若使用者使用 Windows 10 作業系統，需先了解數位簽章的內容，數位簽章是一種可加入檔案的電子安全性標記。它可以讓您確認檔案的發行者，而且可以協助確認檔案自從數位簽章之後並未變更。因目前 DELTA PLC USB 驅動程式無數位簽章，所以接下來將逐步介紹如何關閉 Windows 10 作業系統的數位簽章功能，來達成電腦端可順利安裝 DELTA PLC USB 驅動程式，此設定僅限單次使用，當電腦重新開關機後，會回復原來的強制簽章模式。

(1) 點擊「開始」，再選擇「設定」，然後按住「SHIFT」鍵並點擊「重新啟動」。



(2) 待重新開機後於選擇選項畫面中點選【疑難排解】，接著點選疑難排解畫面中的【進階選項】。

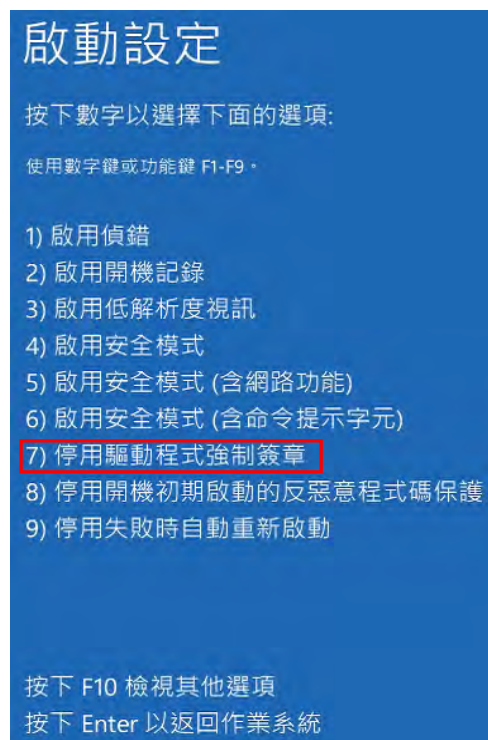


(3) 進階選項畫面中點選【啟動設定】，接著在啟動設定畫面中點選【重新啟動】。

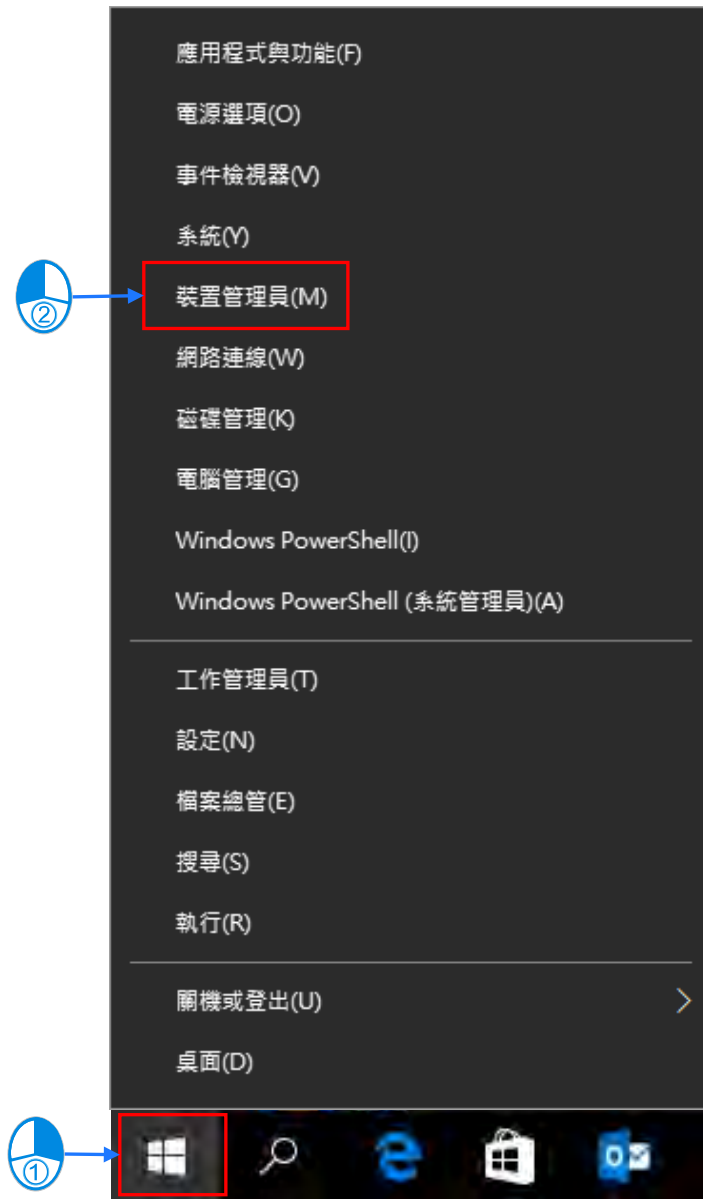


A

(4) 在啟動設定畫面，按下數字鍵 7 或 F7 執行停用驅動程式強制簽章，完成後系統會重新啟動。



(5) 安裝 DELTA USB 驅動程式，使用滑鼠右鍵點擊「開始」，接著點選「裝置管理員」。

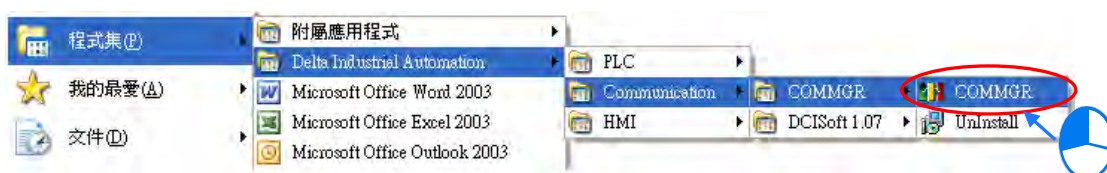


(6) 接著請參考第 A.1.2 節在 Windows 7 環境下安裝 USB 驅動程式，安裝 Delta_PLC_USB_Driver。

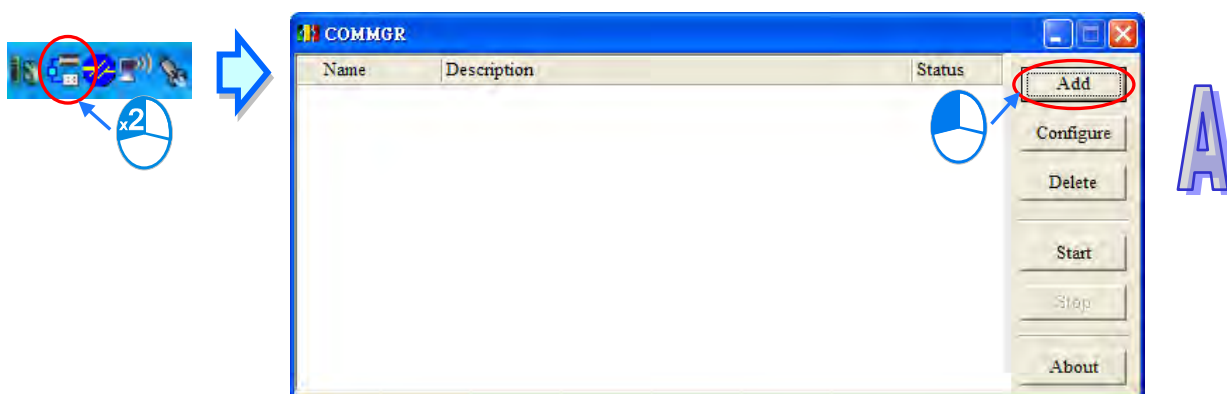
A.2 於 COMMGR 中建立 USB 型態的 Driver

於 COMMGR 中建立 USB 型態 Driver 的步驟如下。

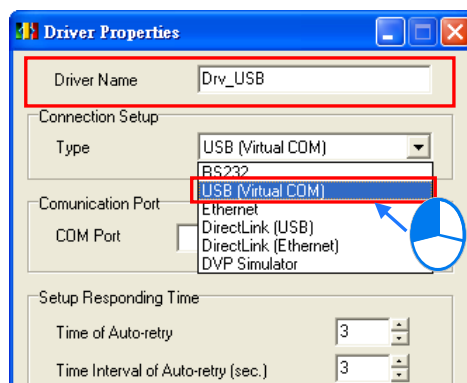
- (1) 確認 PLC 主機已正常上電，並透過 USB 電纜將 PLC 主機連接至個人電腦的 USB 埠。
- (2) 確定通訊管理員 **COMMGR** 已啟動並常駐在 Windows 的系統列中；若否，則請至「程式集」中依下列路徑至「**Delta Industrial Automation**」→「**Communication**」→「**COMMGR**」目錄下點擊「**COMMGR**」的程式執行捷徑。



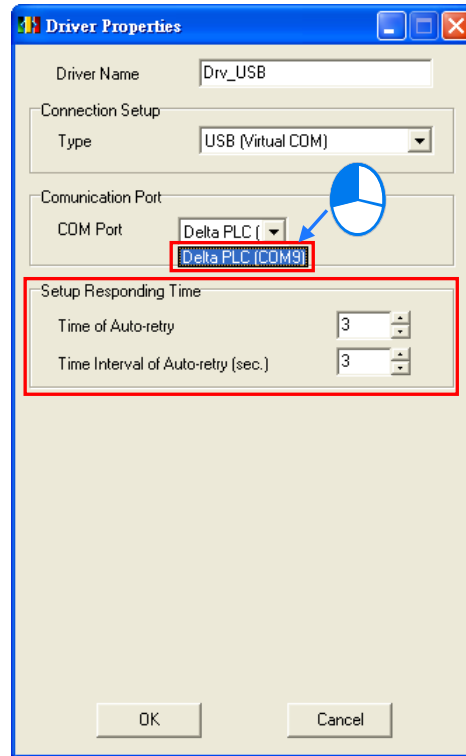
- (3) 於系統列的 COMMGR 圖示上雙擊滑鼠左鍵，之後於 COMMGR 視窗的右側點擊「**Add**」鍵。



- (4) 首先在「**Driver Name**」中輸入 Driver 的名稱，之後於「**Connection Setup**」的「**Type**」欄位中下拉選擇 **USB (Virtual COM)**；不過當 PLC 機種為 SX2 時，連線的型態則請選擇 **RS232**，相關細節將會於下一節當中另行說明。

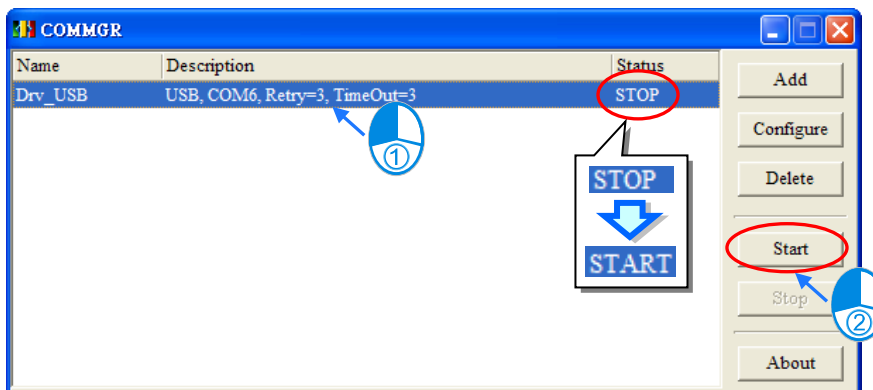


- (5) 接著請於「**Communication Port**」的「**COM Port**」欄位下拉選擇對應 PLC 主機的裝置名稱，之後於「**Time of Auto-retry**」設定連線錯誤時的重試次數，而「**Time Interval of Auto-retry**」參數則是設定重試的時間間隔。完成所有設定後按下「**OK**」鍵即可。



若於「**COM Port**」欄位的下拉選單中沒有對應的 PLC 選項時，請先確認 Windows 裝置管理員的**連接埠 (COM 與 LPT)** 項目中，是否有對應該主機的裝置項目，若否，則請確認 PLC 主機與個人電腦是否已透過 USB 電纜正確連接，並確認該主機的驅動程式是否已正確安裝。確認無誤後，請先關閉 COMMGR 的 Driver 設定視窗後再重新開啟。

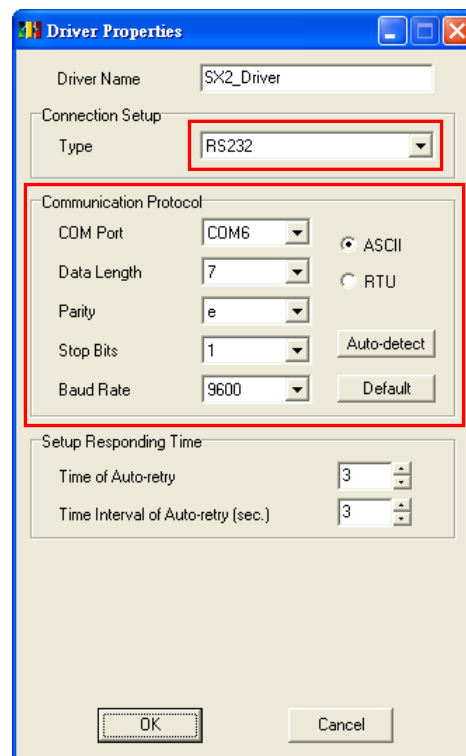
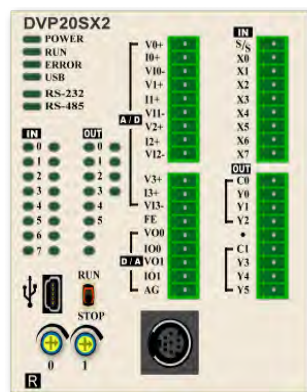
- (6) 於 COMMGR 的管理列表中點選方才建立好的 Driver，接著於視窗的右側點擊「**Start**」鍵，待此 Driver 的 Status 欄位變為「**START**」之後便已成功啟動。



*.後續的測試以及更詳細的 COMMGR 操作方式則請參考第 2.4 節的相關說明。

A.3 DVP-SX2 系列的 USB 設定

DVP-SX2 系列與其他機種的 USB 操作有些許不同，其架構是將 USB 轉 RS232 的電路內建於主機中，因此雖然主機外部的接口是 USB 介面，但內部的運作仍是採用 RS232，所以在 COMMGR 中設定連線型態時也必須選擇 RS232，且其設定的方式，包括「Auto-detect」功能，皆與一般的 RS232 相同。DVP-SX2 系列的 USB 埠口可透過特殊資料暫存器 D1109 來設定其通訊參數，詳細說明請參考該機種的編程手冊。而於 COMMGR 中設定 Driver 時，請確認所設定的 RS232 參數是否與主機中的設定一致。



A

MEMO





附錄B 各系列機種的相關注意事項

目錄

B.1	各系列機種的裝置定址	B-2
B.1.1	AH/AS 系列機種的裝置種類.....	B-2
B.1.2	AH/AS 系列機種 X、Y、D、L 裝置的定址格式	B-3
B.1.3	AH/AS 系列機種 X/Y 接點狀態的即時存取	B-4
B.1.4	DVPxxMC/AS5xx 機種的裝置種類	B-5
B.1.5	DVPxxMC/AS5xx 機種之裝置的定址格式	B-5
B.1.6	DVPxxMC/AS5xx 機種之停電保持裝置	B-6
B.2	各系列機種的裝置資源	B-6
B.2.1	AHCPU500-EN/AHCPU500-RS2 機種	B-6
B.2.2	AHCPU510-EN/AHCPU510-RS2 機種	B-7
B.2.3	AHCPU520-EN/AHCPU520-RS2 機種	B-7
B.2.4	AHCPU530-EN/AHCPU530-RS2 機種	B-8
B.2.5	AHCPU501-EN/AHCPU501-RS2 機種	B-8
B.2.6	AHCPU511-EN/AHCPU511-RS2 機種	B-9
B.2.7	AHCPU521-EN/AHCPU521-RS2 機種	B-9
B.2.8	AHCPU531-EN/AHCPU531-RS2 機種	B-10
B.2.9	AHCPU560-EN2 機種	B-10
B.2.10	AHxxEMC 系列機種	B-11
B.2.11	AS200/AS300 系列機種	B-11
B.2.12	DVPxxMC/AS5xx 系列機種	B-12
B.3	編譯與上下載時間	B-12

AH/AS 系列機種的系統架構與 **DVP 系列機種**及 **DVPxxMC/AS5xx 系列機種**之間有存在部份差異，因此在進行專案開發之前請務必詳閱各機種的使用手冊；附錄中的內容則是針對在 ISPSOft 中進行 **AH/AS 系列機種**、**DVP 系列機種**及 **DVPxxMC/AS5xx 系列機種**的專案開發時，使用者須特別注意的事項來加以補充說明。

B.1 各系列機種的裝置定址

B.1.1 AH/AS 系列機種的裝置種類

下表為 AH/AS 系列機種的裝置種類。

裝置種類	存取屬性	功能說明
X	位元/字元組	輸入繼電器，代表數位輸入 (DI) 點的狀態。(*1)
Y	位元/字元組	輸出繼電器，代表數位輸出 (DO) 點的狀態。(*1)
M	位元	內部輔助繼電器，用於儲存運算過程中的位元狀態。
S	位元	步進點繼電器，用來做為 SFC 步進點的狀態旗標。
D	位元/字元組	內部資料暫存器，用來做為運算資料的儲存暫存器。(*1)
L	位元/字元組	連結暫存器，主要用於資料交換的暫存之用。(*1)(*2)
T	位元	計時器裝置，用做位元裝置時則代表計時到達與否的狀態旗標。
	字元組	計時器裝置，用做字元組裝置時則代表計時器的現在值。
C	位元	計數器裝置，用做位元裝置時則代表計數到達與否的狀態旗標。
	字元組	計數器裝置，用做字元組裝置時則代表計數器的現在值。
HC	位元	32 位元計數器裝置，用做位元裝置時則代表計數到達與否的狀態旗
	字元組	32 位元計數器裝置，用做字元組裝置時則代表計數器的現在值。
E	字元組	間接指定暫存器，用做偏移 (Offset) 運算之用。(*3)
SM	位元	特殊輔助繼電器 (特 M)，做為系統特定功能的狀態旗標。(*4)
SR	字元組	特殊資料暫存器 (特 D)，做為系統特定功能的資料暫存器。(*4)

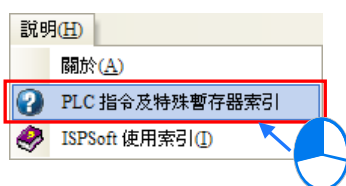
*1. 依據表達方式的不同，X、Y、D 與 L 裝置可分別進行字元組或位元操作，相關說明將於本節後續的內容中另行補充。

*2. 連結暫存器 (L 裝置) 主要用於資料交換的暫存之用，但亦可當做一般的 D 裝置來使用。AS 系列機種無 L 裝置。

*3. AH/AS 機種的間接指定暫存器只有 E 裝置，而沒有 F 裝置。

*4. 有別於 DVP 機種，AH/AS 機種將特殊輔助繼電器與特殊資料暫存器獨立規劃為 SM 與 SR 兩種類型的裝置。

關於各裝置的詳細使用方式與限制，請參考各機種的程式手冊，或在各機種專案中透過 ISPSOft 的輔助說明功能來進行查詢。

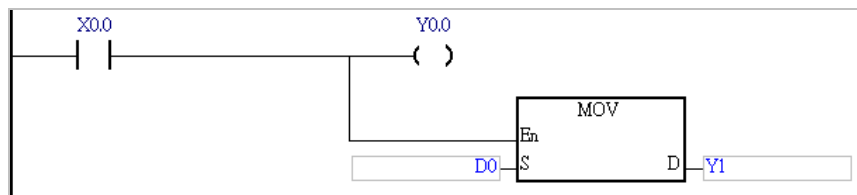


B.1.2 AH/AS 系列機種 X、Y、D、L 裝置的定址格式

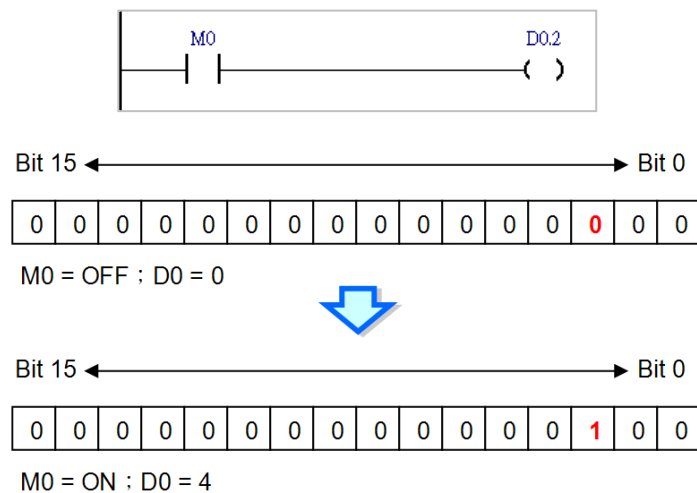
X/Y 位元裝置在 DVP 機種的程式是直接使用 X0、Y1...等表達方式，而 AH/AS 機種則須使用 X0.0、Y0.1 的格式來表達；AH/AS 機種的 X 與 Y 裝置可直接進行字元組的操作，因此在裝置表達的定義上，X0 或 Y0 都將被視為一個 WORD 的字元組裝置，其中 X0.0 代表 X0 這個字元組裝置的第一個位元（最低位元），而 X0.15 則代表最後一個位元（最高位元）。



如下圖範例，利用把 D0 的內容搬移給 Y1 的動作，便可批次改變 Y1.0 ~ Y1.15 的位元狀態。



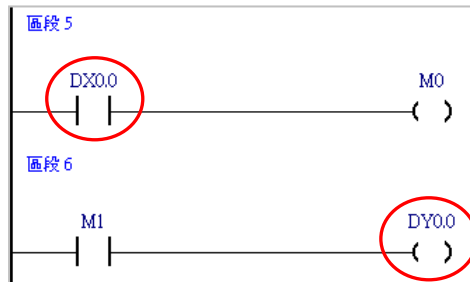
除了 X 與 Y 裝置可同時使用位元與字元組操作之外，在 AH/AS 系列機種中，D 與 L 等字元組裝置亦可使用相同的表達方式來直接操作其中的指定位元。參考下圖範例，接點 M0 的狀態會輸出給 D0.2，因此若裝置 D0 的初始值為 0，則當 M0 為 ON 時，D0 的值就會變為 4。



當 X、Y、D、L 裝置在進行位元或字元組操作時，基本上都是在同一個記憶體區塊，差別僅在於存取整個字元組或是指定位元；而 T、C、HC 裝置，其位元與字元組裝置則是完全獨立的兩個區塊；當該裝置名稱用於位元的指令或運算元時，其所代表的即是計數/計時到達與否的位元狀態旗標；而若使用於字元組的指令或運算元時，該裝置名稱便代表計數/計時器的現在值。

B.1.3 AH/AS 系列機種 X/Y 接點狀態的即時存取

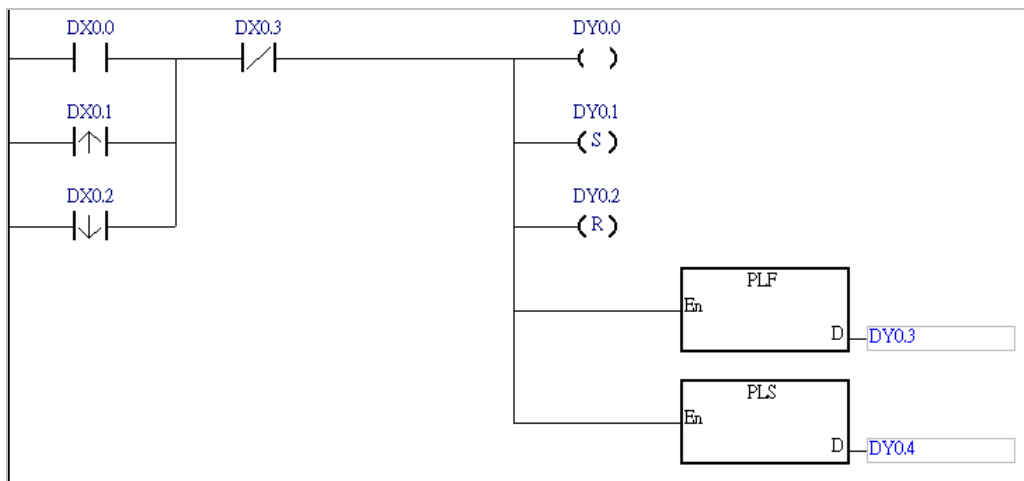
AH/AS 機種的 IO 更新時機與 DVP 機種類似，皆是在每個掃描週期的前後進行批次刷新的動作，但 AH/AS 機種尚支援特別的修飾方式，當在原本的 X 或 Y 裝置前加上修飾字元「D」時，系統便會在程式執行至該指令時立即更新裝置的狀態；不過此用法只可修飾 X 或 Y 裝置，而不可修飾變數符號。如下圖，當程式執行至區段 5 時，系統便會即時讀取外部輸入 X0.0 端點的信號狀態；同樣的，當程式執行至區段 6 時，系統亦會立即將 M1 的狀態經由 Y0.0 輸出，而無須等待掃描週期結束。



下表即為支援「D」字元修飾的指令，若在列表以外的指令中使用「D」字元修飾的 X 或 Y 裝置時，系統便會將其視為錯誤的語法，並在編譯時產生錯誤訊息；由下表中可得知，使用「D」字元修飾的 X 或 Y 裝置只可用於位元操作，且亦無法做為其他 API 應用指令的運算元。

裝置種類	支援指令
X 裝置	LD / LD1 / LDP / LDF / OR / ORI / ORP / ORF / AND / ANI / ANDP / ANDF
Y 裝置	OUT / SET / RST / PLS / PLF

下圖便是一些合法的應用範例。



下圖則為不合法的應用範例，使用「D」字元修飾的 Y 裝置不可做為輸入接點。



B.1.4 DVPxxMC/AS5xx 機種的裝置種類

- 裝置名稱說明

有別於 AH/AS 機種與 DVP 機種裝置，DVPxxMC/AS5xx 機種的裝置，使用的是 IEC 規範，如下圖。



- DVPxxMC/AS5xx 在軟體中使用到的相關裝置清單如下：

序號	項目	內容				
1	首碼 1 旗標	I	Q	M	-	-
2	首碼 1 名稱	輸入裝置	輸出裝置	中間裝置	-	-
3	首碼 2 旗標	X	B	W	D	L
4	首碼 2 資料型態	位元 裝置	位元組 裝置	字元組 裝置	雙字元組 裝置	四字元組 裝置
5	裝置範例	%IX0.0	%IB0	%IW0	%ID0	%ILO
6		%QX0.0	%QB0	%QW0	%QD0	%QL0
7		%MX0.0	%MB0	%MW0	%MD0	%ML0



B.1.5 DVPxxMC/AS5xx 機種之裝置的定址格式

- 裝置對應關係如下表所示：

如下表所示，%ML0 包括%MB0~%MB7，%MD0 包括%MB0~%MB3，%MW0 包括%MB0~%MB1。

裝置 名稱	裝置對應關係																							
	第 1 個 WORD					第 2 個 WORD					第 3 個 WORD					第 4 個 WORD								
	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15
%MX	%MX0.0~0.7			%MX1.0~1.7			%MX2.0~2.7			%MX3.0~3.7			%MX4.0~4.7			%MX5.0~5.7			%MX6.0~6.7			%MX7.0~7.7		
%MB	%MB0			%MB1			%MB2			%MB3			%MB4			%MB5			%MB6			%MB7		
%MW	%MW0					%MW1					%MW2					%MW3								
%MD	%MD0										%MD1													
%ML	%ML0																							

如下表所示，%ML1 包括%MB8~%MB15，%MD2 包括%MB8~%MB11，%MW4 包括%MB8~%MB9，%MB8 包括%MX8.0~8.7。

裝置名稱	裝置對應關係																							
	第 5 個 WORD						第 6 個 WORD						第 7 個 WORD						第 8 個 WORD					
	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15	Bit 0	...	Bit 7	Bit 8	...	Bit 15
%MX	%MX8.0 ~8.7		%MX9.0 ~9.7		%MX10.0 ~10.7		%MX11.0 ~11.7		%MX12.0 ~12.7		%MX13.0 ~13.7		%MX14.0 ~14.7		%MX15.0 ~15.7									
%MB	%MB8		%MB9		%MB10		%MB11		%MB12		%MB13		%MB14		%MB15									
%MW	%MW4						%MW5						%MW6						%MW7					
%MD	%MD2												%MD3											
%ML	%ML1																							

B.1.6 DVPxxMC/AS5xx 機種之停電保持裝置

%MW0~%MW999 為停電保持裝置。另外，軟體中定義的變數可以選擇是否為停電保持。停電保持的容量為 100K 位元組。

B.2 各系列機種的裝置資源

B.2.1 AHCPU500-EN/AHCPU500-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X63	字元：X0.0~X63.15
Y	字元組：Y0~Y63	字元：Y0.0~Y63.15
D	字元組：D0~D16383	字元：D0.0~D16383.15
L	字元組：L0~L16383	字元：L0.0~L16383.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能塊的「實例記憶區塊」數量	64 個	

B.2.2 AHCPU510-EN/AHCPU510-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X127	字元：X0.0~X127.15
Y	字元組：Y0~Y127	字元：Y0.0~Y127.15
D	字元組：D0~D32767	字元：D0.0~D32767.15
L	字元組：L0~L32767	字元：L0.0~L32767.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能塊的「實例記憶區塊」數量	256 個	

B.2.3 AHCPU520-EN/AHCPU520-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X255	字元：X0.0~X255.15
Y	字元組：Y0~Y255	字元：Y0.0~Y255.15
D	字元組：D0~D65535	字元：D0.0~D65535.15
L	字元組：L0~L65535	字元：L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能塊的「實例記憶區塊」數量	512 個	

B

B.2.4 AHCPU530-EN/AHCPU530-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X511	字元：X0.0~X511.15
Y	字元組：Y0~Y511	字元：Y0.0~Y511.15
D	字元組：D0~D65535	字元：D0.0~D65535.15
L	字元組：L0~L65535	字元：L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	
功能塊的「實例記憶區塊」數量	1024 個	

B.2.5 AHCPU501-EN/AHCPU501-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X127	字元：X0.0~X127.15
Y	字元組：Y0~Y127	字元：Y0.0~Y127.15
D	字元組：D0~D24575	字元：D0.0~D24575.15
L	字元組：L0~L24575	字元：L0.0~L24575.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	

B.2.6 AHCPU511-EN/AHCPU511-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X255	字元：X0.0~X255.15
Y	字元組：Y0~Y255	字元：Y0.0~Y255.15
D	字元組：D0~D49151	字元：D0.0~D49151.15
L	字元組：L0~L49151	字元：L0.0~L49151.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4096	
SM	SM0~SM4096	

B.2.7 AHCPU521-EN/AHCPU521-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X511	字元：X0.0~X511.15
Y	字元組：Y0~Y511	字元：Y0.0~Y511.15
D	字元組：D0~D98303	字元：D0.0~D98303.15
L	字元組：L0~L98303	字元：L0.0~L98303.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	

B

B.2.8 AHCPU531-EN/AHCPU531-RS2 機種

裝置類別	定址範圍	
X	字元組：X0~X1023	字元：X0.0~X1023.15
Y	字元組：Y0~Y1023	字元：Y0.0~Y1023.15
D	字元組：D0~D131071	字元：D0.0~D131071.15
L	字元組：L0~L131071	字元：L0.0~L131071.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4095	
SM	SM0~SM4095	

B.2.9 AHCPU560-EN2 機種

裝置類別	定址範圍	
X	字元組：X0~X4095	字元：X0.0~X4095.15
Y	字元組：Y0~Y4095	字元：Y0.0~Y4095.15
D	字元組：D0~D262143	字元：D0.0~D262143.15
L	字元組：L0~L262143	字元：L0.0~L262143.15
M	M0~M8191	
S	S0~S4095	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR4095	
SM	SM0~SM4095	

B

B.2.10 AHxxEMC 系列機種

裝置類別	定址範圍	
X	字元組：X0~X511	字元：X0.0~X511.15
Y	字元組：Y0~Y511	字元：Y0.0~Y511.15
D	字元組：D0~D65535	字元：D0.0~D65535.15
L	字元組：L0~L65535	字元：L0.0~L65535.15
M	M0~M8191	
S	S0~S2047	
C	C0~C2047	
HC	HC0~HC63	
T	T0~T2047	
E	E0~E31	
SR	SR0~SR2047	
SM	SM0~SM2047	

B.2.11 AS200/AS300 系列機種

裝置類別	定址範圍	
X	字元組：X0~X63	字元：X0.0~X63.15
Y	字元組：Y0~Y63	字元：Y0.0~Y63.15
D	字元組：D0~D29999	字元：D0.0~D29999.15
M	M0~M8191	
S	S0~S2047	
C	C0~C511	
HC	HC0~HC255	
T	T0~T511	
E	E0~E14	
SR	SR0~SR2047	
SM	SM0~SM4095	

B

B.2.12 DVPxxMC/AS5xx 系列機種

裝置名稱	裝置範圍
%IX	%IX0.0~%IX127.7
%QX	%QX0.0~%QX127.7
%MX	%MX0.0~%MX131071.7
%IB	%IB0~%IB127
%QB	%QB0~%QB127
%MB	%MB0~%MB131071
%IW	%IW0~%IW63
%QW	%QW0~%QW63
%MW	%MW0~%MW65535
%ID	%ID0~%ID31
%QD	%QD0~%QD31
%MD	%MD0~%MD32767
%IL	%IL0~%IL15
%QL	%QL0~%QL15
%ML	%ML0~%ML16383

B**B.3 編譯與上下載時間**

ISPSOft 的編譯時間與程式的大小或變數符號的多寡，及使用的電腦等級有關；而上下載時間則依據專案大小，與連線的種類及設定的通訊速度有關。在此列出以下數據供使用者參考。

電腦等級：CPU Intel(R) I7 · RAM 4GB · Windows 7 32 位元。

專案內容：5000 行一個接點輸出一個線圈的基本指令的階梯圖，2 個變數符號。

編譯時間：約 2 秒

專案內容：容量約 10K STEP

下載時間：透過 USB 約 4.9 秒

上載時間：透過 USB 約 4.6 秒



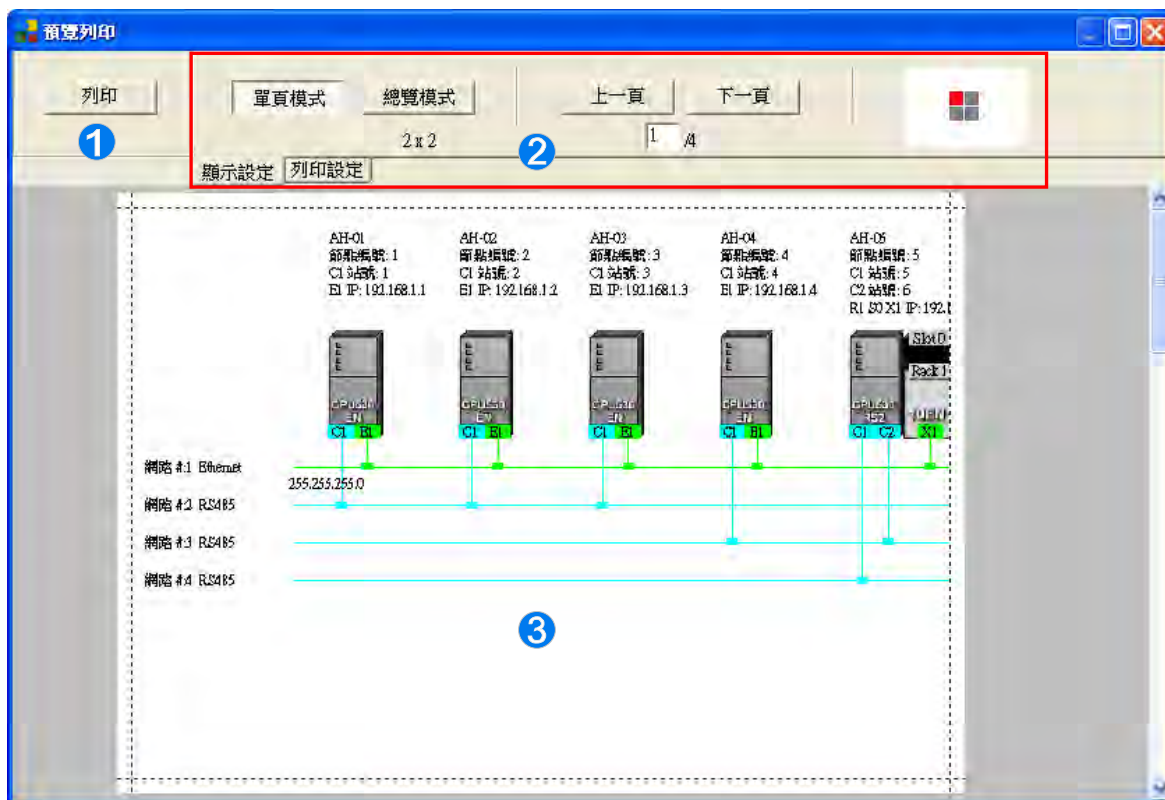
附錄 C 列印管理工具

目錄

C.1 環境簡介	C-2
C.2 操作功能說明	C-3

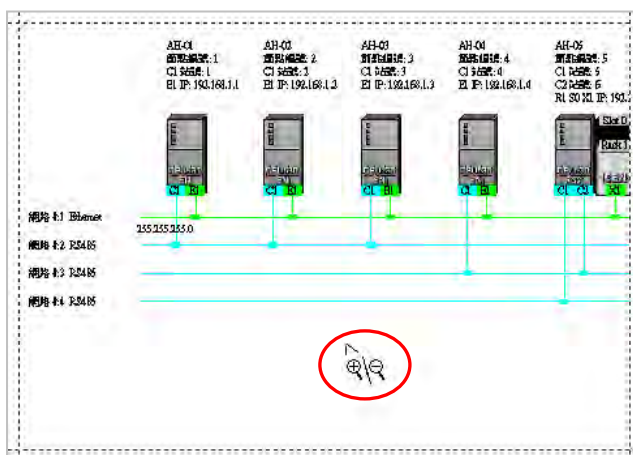
C.1 環境簡介

列印管理工具主要是提供一個便利的操作介面，使用者可在視窗中進行列印預覽與列印格式的設定。



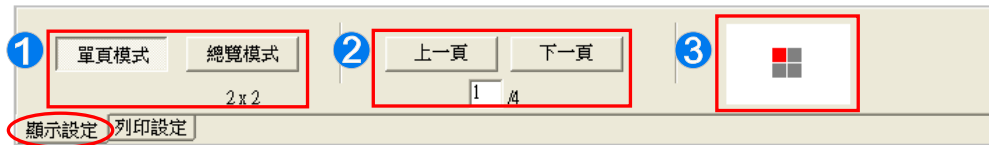
- ① 列印按鈕：完成設定並確認無誤後，按下此鍵便可開始列印。
- ② 設定操作區：共分為「顯示設定」與「列印設定」兩個頁面，而藉由下方的標籤便可進行切換。
- ③ 列印預覽區：此區將顯示列印結果的預覽畫面。

當將滑鼠移至列印預覽區並點擊左鍵時，鼠標的圖示便會變為放大鏡的狀態；此時若按下滑鼠左鍵，預覽畫面便會隨之放大，而按下右鍵時，畫面便會再度縮小。



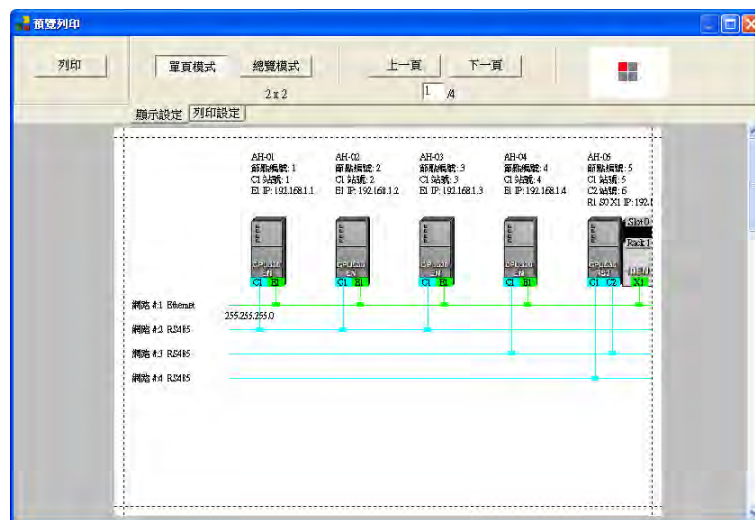
C.2 操作功能說明

下圖即為設定操作區的「顯示設定」頁面，其主要的功能在於調整下方列印預覽區的顯示狀態。

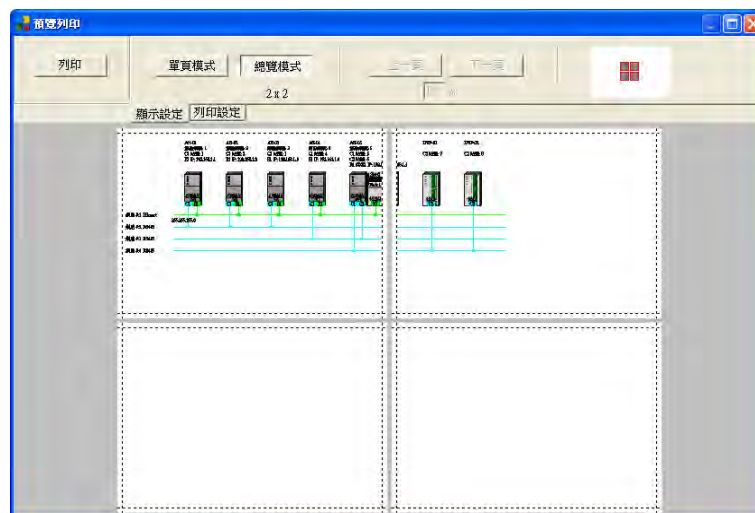


- ❶ 當列印的範圍超過紙張大小時，系統將會自動將其分割為多個頁面，而於此處便可切換預覽畫面的顯示方式；其中在按鈕下方的數字則代表頁面切割的數量。
- ❷ 透過此處的按鈕便可切換顯示頁面，而於其下方則會顯示目前的頁碼與總頁數。
- ❸ 此處的圖示代表目前頁面與整體文件的相對位置，其中紅色區塊即為目前頁面。

● 單頁模式



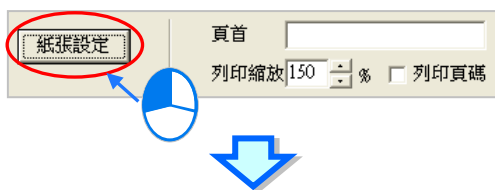
● 總覽模式



下圖則為設定操作區的「列印設定」頁面，其主要的功能在於設定列印的格式。



- 1 按下「紙張設定」按鈕便可開啟印表機的設定視窗。
- 2 於此可設定文件的列印比例，同時亦可設定列印時是否添加頁首文字與頁碼。
- 3 此處可設定列印文件的邊界。



當完成所有的設定並確認無誤之後，按下「列印」鍵便可開始列印。

