
Win-GRAF 使用手冊

泓格科技股份有限公司 版權所有。

泓格科技股份有限公司感謝您購買本公司的 Win-GRAF 系列控制系統，本系列產品結合方便整合的特性以及強大的發展性，適用於各類工業用控制系統，並期望對使用 Win-GRAF 作為開發軟體的系統整合人員、程式設計師以及系統維護人員能有所助益。

本公司出產的 Win-GRAF 可程式自動控制器 (PAC: Programming Automation Controller) 包括:

ViewPAC-1000:	VP-1238-CE7
ViewPAC-2000:	VP-2208-CE7, VP-6208-CE7, VP-2238-CE7, VP-6238-CE7
ViewPAC-4000:	VP-4208-CE7, VP-4238-CE7
WinPAC-5000:	WP-5238-CE7
WinPAC-8000:	WP-8148, WP-8448, WP-8848 WP-8128-CE7, WP-8428-CE7, WP-8828-CE7
XPAC-8000-CE6	XP-8048-CE6, XP-8348-CE6, XP-8748-CE6 XP-8038-CE6, XP-8138-CE6, XP-8338-CE6, XP-8738-CE6

免責聲明

泓格科技股份有限公司，對於因使用本系列產品所造成的任何損害並不負任何法律上的責任。本公司保留在任何時候修訂本書而不需通知的權利，並將確實地提供正確且可靠的資訊。然而，泓格科技股份有限公司無義務對任何因非法或不當使用導致的第三方侵權行為承擔任何法律責任。

商標 與 版權

版權所有 © 2014 泓格科技股份有限公司保留所有權利。本手冊中所提及之所有商標，均屬於其合法註冊公司所有。本書每一頁下方皆列有 版本號 與 發佈日期，若想取得最新文件可到 <http://ftp.icpdas.com/pub/cd/win-graf-workbench-cd/tutorials/> 來下載。

技術支援

Win-GRAF 網站:

http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/win-graf.html

新版 Win-GRAF Workbench, Lib 與 PAC Driver:

http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-driver.html

若您對產品有任何問題與建議，歡迎隨時與我們聯繫。E-mail: service@icpdas.com。

目 錄

WIN-GRAF 使用手冊	1-1
目 錄.....	1-2
第 1 章 軟體安裝與硬體設定	1-9
1.1 安裝 Win-GRAF 軟體	1-9
1.2 開啟 Win-GRAF 軟體	1-12
1.2.1 軟體操作模式.....	1-13
1.2.2 軟體介面說明.....	1-14
1.2.3 程式庫管理員 (Library Manager) 功能.....	1-16
1.3 設定 Win-GRAF PAC 的 IP 位址.....	1-17
第 2 章 編寫一個簡單的 WIN-GRAF 範例.....	2-1
2.1 建立 Win-GRAF 專案	2-1
2.1.1 建立樣版專案 (Demo01).....	2-1
2.1.2 重要專案設定.....	2-3
2.2 專案介紹.....	2-5
2.2.1 Demo01 - LD 程式.....	2-5
2.2.2 Demo01 - 變數說明.....	2-7
2.3 小試身手.....	2-8
2.3.1 宣告專案變數.....	2-8
2.3.2 宣告 I/O 變數.....	2-10
2.3.3 建立 LD 程式.....	2-12
2.3.4 編譯程式.....	2-18
2.3.5 下載程式到 PAC	2-19
2.3.6 測試程式.....	2-22
第 3 章 MODBUS SLAVE: 開放 WIN-GRAF PAC 與 圖控/HMI 軟體來相互溝通	3-1
3.1 啟用 Win-GRAF PAC 為 Modbus TCP Slave	3-1
3.2 啟用 Win-GRAF PAC 為 Modbus RTU Slave.....	3-7
第 4 章 使用 “I/O BOARD” 功能	4-1
4.1 DI/DO 卡.....	4-3
4.2 i_scale (轉換表功能)	4-4
4.3 i_8017HW (8/16 通道 AI).....	4-6
4.4 i_8024 (4 通道 AO).....	4-8
4.5 i_87018W (8 通道 AI).....	4-10
4.6 i_exist (測試 I/O 模組是否存在?)	4-13
4.7 i_8084 (頻率量測, UP/Down 計數器, UP 計數器)	4-14
4.7.1 i_8084_freq (頻率量測).....	4-14

4.7.2	i_8084_cnt_ch04 (4 通道 UP/Down 計數器)	4-17
4.7.3	i_8084_cnt_ch08 (8 通道 UP 計數器)	4-19
4.8	i_8093 (3 軸之高速 Encoder 模組)	4-21
4.9	I-8084W, I-8093W, I-87082W, I-87084W, I-7083 與 I-7080 模組的計數功能	4-23
4.9.1	COUNTER_START (開始計數)	4-23
4.9.2	COUNTER_STOP (停止計數)	4-25
4.9.3	COUNTER_GET (取得計數值)	4-26
4.9.4	COUNTER_STATE (取得計數狀態)	4-27
4.9.5	COUNTER_RESET (重置計數值)	4-28
4.10	Ping_ip (測試遠端的 Ethernet/Internet 設備連線)	4-29
4.11	I-8088W (8 通道 PWM 輸出模組)	4-31
第 5 章	MODBUS MASTER: 連接其它 MODBUS SLAVE 設備	5-1
5.1	啟用 Win-GRAF PAC 為 Modbus RTU/ASCII Master (I/O & XV-board)	5-1
5.1.1	讀取 DI 資料	5-3
5.1.2	寫出 DO 資料	5-6
5.1.3	讀取 AI 資料	5-8
5.1.4	寫出 AO 資料 (16-bit)	5-11
5.1.5	寫出 AO 資料 (32-bit)	5-13
5.1.6	XV Board 使用說明	5-15
5.1.7	連接 XV107, XV107A (8 DI, 8 DO)	5-21
5.1.8	連接 XV110 (16 DI)	5-23
5.1.9	連接 XV111, XV111A (16 DO)	5-24
5.1.10	連接 XV116 (5 DI, 6 Relay)	5-26
5.1.11	連接 XV308 (8 AI, 8 DIO)	5-28
5.1.12	連接 XV310 (4 AI, 2 AO, 4 DI, 4 DO)	5-34
5.1.13	停止/啟用 Modbus RTU/ASCII Master Port	5-37
5.2	啟用 Win-GRAF PAC 為 Modbus TCP/UDP Master (Ethernet I/O)	5-38
5.2.1	連接 ET-7000 系列 I/O 模組	5-43
5.2.2	連接 ET-7060 (6 DI, 6 Relay)	5-46
5.2.3	連接 ET-7018Z (10 AI)	5-48
5.2.4	停止/啟用 Modbus TCP/UDP Master Port	5-49
5.3	連接具有 2 個 IP 位址的 Modbus TCP Slave 設備	5-50
5.4	連接 tGW-700 來擴充 Modbus RTU Master 串口	5-55
5.4.1	使用 tGW-700 系列 (Modbus TCP 轉 Modbus RTU/ASCII 閘道器)	5-55
5.4.2	連接 tGW-700 系列與 LC-103 模組 (1 DI, 3 Relay)	5-58
5.4.3	測試範例程式 (demo_tgw725.zip)	5-62
第 6 章	可保存變數與資料儲存	6-1
6.1	可保存變數 (Retain Variable)	6-1
6.1.1	RETAIN_VAR (設定保存一個變數)	6-3

6.1.2	RETAIN_ARRAY (設定保存一個陣列變數)	6-4
6.1.3	RETAIN_FLAG_SET/GET/CLR (設定/取得/刪除 Flag 的狀態)	6-5
6.2	可保存變數 (使用檔案)	6-7
6.3	備份資料到 EEPROM	6-11
6.3.1	EEP_READ (讀取 EEPROM)	6-12
6.3.2	EEP_WRITE (寫入 EEPROM)	6-12
第 7 章	在 PAC 間互傳資料 (DATA BINDING)	7-1
第 8 章	連接 DCON I/O 模組	8-1
8.1	設定 "DCON" I/O 卡	8-2
8.2	使用 I/O 功能方塊	8-4
8.2.1	"D_7065" 功能方塊	8-5
8.2.2	"D_7018Z" 功能方塊	8-6
8.2.3	"D_7083" 功能方塊	8-8
8.2.4	"D_87084_FREQ" 功能方塊	8-9
8.2.5	"D_87084_CNT4" 功能方塊	8-10
8.2.6	"D_87084_CNT8" 功能方塊	8-11
8.2.7	"DL_100T485" 功能方塊	8-12
8.2.8	"D_GPS721" 功能方塊	8-13
第 9 章	即時線上更新 (ON LINE CHANGE)	9-1
9.1	"On Line Change" 功能的使用限制	9-1
9.2	使用 "On Line change" 功能	9-3
第 10 章	資料/型態轉換與使用 PAC 時間	10-1
10.1	AI 資料轉換	10-1
10.2	AO 資料轉換	10-2
10.3	資料型態轉換	10-4
10.4	BCD 轉換	10-5
10.5	組合/拆解 整數或布林值	10-6
10.6	組合/拆解 BYTE, WORD, DWORD	10-8
10.7	將變數拆成 Byte Array 或將 Byte Array 組成變數	10-11
10.8	取得/設定 PAC 時間	10-13
第 11 章	一般常用工具與有用的技巧	11-1
11.1	更新 Win-GRAF 程式庫 (Library)	11-1
11.2	更新 Win-GRAF 驅動程式 (Driver)	11-2
11.3	觀測清單 (Spy List)	11-3
11.4	備份/回存一個 Win-GRAF 專案 (Project)	11-5
11.5	以軟體重新啟動 PAC	11-7
11.6	在 LD 與 FBD 內使用 ST 語法	11-8
11.7	置換 PAC 內的配方表 (Recipe)	11-9

11.8	取得 PAC 所支援的函式 (Function) 與功能方塊 (Function Block)	11-11
11.9	上傳 Win-GRAF 專案原始碼	11-14
11.10	設定 PAC 的密碼	11-16
11.11	使用 ST 語法來操作功能方塊	11-18
11.12	如何保護您的 Win-GRAF 程式，讓盜用者無法使用?	11-19
第 12 章 範例程式說明		12-1
12.1	範例程式列表	12-2
12.2	計時器 (Timer) 操作	12-4
12.2.1	啟動、停止、重置計時	12-4
12.2.2	週期性的操作	12-5
12.2.3	偵測穩定的 ON 或 OFF 訊號	12-7
12.2.4	觸發後維持 "ON" 一段時間	12-8
12.3	序列埠的通訊操作	12-9
12.3.1	使用 COM Port 來傳送一個字串	12-10
12.3.2	使用 COM Port 對設備一問一答	12-11
12.3.3	COM Port 等待遠端設備傳來的資料	12-13
12.3.4	使用 COM Port 定期回報資料給遠端設備	12-15
12.4	讀/寫 PAC 內儲存裝置的檔案 (File)	12-16
12.4.1	寫入資料到 PAC 內的檔案	12-17
12.4.2	讀取 PAC 內的檔案資料	12-19
12.4.3	資料紀錄功能	12-22
第 13 章 使用 VB.NET 2008 程式來讀/寫 WIN-GRAF 變數		13-1
13.1	如何回存 Win-GRAF 專案?	13-1
13.2	如何開放 Win-GRAF 變數給 .NET 程式使用?	13-2
13.3	建立 VB.NET 新專案	13-4
13.3.1	加入專案參考 (Project Reference)	13-5
13.4	編譯應用程式	13-7
13.5	"UserShareNet.DLL" 內的函式說明	13-8
13.5.1	讀/寫 Boolean 的函式	13-8
13.5.2	讀/寫 整數 的函式	13-10
13.5.3	讀/寫 實數 的函式	13-12
13.5.4	讀/寫 字串 的函式	13-14
13.5.5	如何讓 VB.NET 程式讀取 Win-GRAF 字串變數?	13-16
第 14 章 使用 C# 程式來讀/寫 WIN-GRAF 變數		14-1
14.1	如何回存 Win-GRAF 專案?	14-1
14.2	如何開放 Win-GRAF 變數給 C# 程式使用?	14-1
14.3	建立 C# 新專案	14-1
14.3.1	加入 C# 專案參考	14-3

14.4	編譯應用程式.....	14-5
14.5	"UserShareNET.DLL" 內的函式說明.....	14-6
14.5.1	讀/寫 Boolean 的函式.....	14-6
14.5.2	讀/寫 整數 的函式.....	14-8
14.5.3	讀/寫 實數 的函式.....	14-10
14.5.4	讀/寫 字串 的函式.....	14-12
14.5.5	如何讓 C# 程式讀取 Win-GRAF 字串變數?.....	14-14
第 15 章	如何在 WIN-GRAF PAC 上使用 ELOGGER HMI	15-1
15.1	Win-GRAF 專案.....	15-1
15.2	eLogger 專案.....	15-1
第 16 章	備援 (冗余) 系統.....	16-1
16.1	特色與架構.....	16-1
16.2	重要通訊 Port 與安裝事項.....	16-4
16.3	範例程式介紹.....	16-6
16.3.1	"I/O Board" 設定.....	16-6
16.3.2	變數宣告 (demo_RDN_2).....	16-10
16.3.3	程式介紹 (demo_RDN_2).....	16-11
16.3.4	專案介紹 (demo_RDN_4).....	16-13
16.4	測試範例程式.....	16-17
16.4.1	測試 demo_RDN_2 · demo_RDN_3.....	16-17
16.4.2	測試 demo_RDN_4.....	16-23
16.5	哪些資料會自動備援到 Passive PAC?.....	16-29
第 17 章	排程控制 (SCHEDULE CONTROL).....	17-1
17.1	安裝 Schedule-Control Utility 與回存 Win-GRAF 範例程式.....	17-2
17.2	"demo_schedule" 範例程式說明.....	17-4
17.3	使用 Schedule-Control Utility 來設定 Target 1 ~ 10 的排程.....	17-7
17.4	測試範例程式.....	17-10
17.5	Schedule-Control Utility 設定說明.....	17-12
17.5.1	每個 Target 控制變數的 Address.....	17-12
17.5.2	控制標的 (Target) 設定說明.....	17-13
17.5.3	季節 (Season) 設定說明.....	17-15
17.5.4	一般日 · 假日 · 特別日 設定說明.....	17-17
17.5.5	排程 (Schedule) 設定說明.....	17-19
17.5.6	儲存與傳送 設定檔至 Win-GRAF PAC.....	17-21
17.5.7	時間同步 (Time Synchronization).....	17-22
17.5.8	控制器端的 Schedule-Control 工具程式.....	17-23
17.5.9	在 eLogger HMI 內應用 Schedule-Control.....	17-25
第 18 章	開發屬於自己的 FUNCTION 與 FUNCTION BLOCK	18-1

18.1	自定義的 Function/Function Block 開發流程	18-2
18.2	建立開發編譯環境.....	18-3
18.2.1	安裝 ViewPAC 或 WinPAC 的開發套件 (SDK).....	18-3
18.2.2	安裝 XPAC (XP-8xx8-CE6, XP-8xx8-Atom-CE6) 的開發套件 (SDK).....	18-6
18.3	定義 Function 或 Function Block.....	18-7
18.3.1	定義 Function Lib.....	18-7
18.3.2	定義 Function Block Lib	18-10
18.4	編寫 Function 或 Function Block 的邏輯程式.....	18-12
18.4.1	編寫 "T5BLOCKS.cpp"	18-14
18.4.2	編寫 Function 內的邏輯程式 (本例是 "bytes_to_long.c").....	18-15
18.4.3	編寫 Function Block 內的邏輯程式 (本例是 "long_to_bytes.c").....	18-16
18.4.4	編譯專案.....	18-18
18.5	測試自定義的 Function 與 Function Block.....	18-19
第 19 章	使用 3G 模組 – I-8212W-3GWA.....	19-1
19.1	硬體安裝與設定.....	19-2
19.2	軟體安裝.....	19-4
19.2.1	安裝 I-8212W-3GWA 的 Driver.....	19-4
19.2.2	設定 3G/2G 撥接參數	19-5
19.2.3	重要設定，不可忽略!!!.....	19-10
19.2.4	啟用 "Dial_up_utility_win_graf" 撥接軟體.....	19-12
19.3	控制 3G/2G 連線的函式說明	19-13
第 20 章	透過 ETHERNET 或 3G/2G 無線，傳送 PAC 內的檔案至遠方 PC.....	20-1
20.1	"WG-Communication-Server" 軟體說明.....	20-2
20.2	"Send_File_To"、"Send_File_State" 與 "Send_File_Abort" 函式.....	20-4
20.3	Win-GRAF 範例程式 (demo_send_file.zip).....	20-6
20.4	檔案傳送測試 (Ethernet).....	20-9
第 21 章	WIN-GRAF 簡訊功能.....	21-1
21.1	"GSM_Open"、"Send_SMS" 與 "Read_SMS" 函式.....	21-1
21.2	Win-GRAF 範例程式 (Demo_SMS.zip)	21-4
21.3	簡訊傳送測試.....	21-6
第 22 章	WIN-GRAF 3G 智能型解決方案.....	22-1
22.1	架設一台 PC/WG-Communication-Server	22-2
22.2	架設遠方的 PAC 來連上 WG-Communication-Server.....	22-5
22.3	在 User PC 與 SCADA PC 架設 WG-Communication-Client.....	22-7
22.3.1	由 PC 登入 Server 並編輯欲偵測的 PAC 連線	22-7
22.3.2	由 PC/Win-GRAF Workbench 更新遠端 PAC 的 Win-GRAF 專案	22-10
22.3.3	由 PC/SCADA 讀取/寫入遠端 PAC 的資料	22-12
第 23 章	HART MASTER.....	23-1

23.1	HART 模組 - I-87H17W 介紹	23-1
23.2	HART 通訊 Frame 格式介紹.....	23-3
23.3	範例程式介紹.....	23-5
23.3.1	I/O Board 設定 與 HART 函式.....	23-5
23.3.2	範例程式 - “Demo_HART_1”	23-10
23.3.3	範例程式 - “Demo_HART_2”	23-13
23.3.4	範例程式 - “Demo_HART_3”	23-19
23.4	測試範例程式.....	23-21
23.4.1	測試環境 與 “HC_Tool” 設定	23-21
23.4.2	測試 “Demo_HART_1” · “Demo_HART_2” 與 “Demo_HART_3”	23-23
附錄 A	資料型態與數值範圍	1
附錄 B	錯誤訊息排除	2
附錄 C	啟動 WINCE PAC 螢幕保護功能	4
附錄 D	使用 RS-232/485/422 擴充卡	5
附錄 E	啟用一個序列埠 讓 WIN-GRAF WORKBENCH 連進來	7
附錄 F	PAC 上序列埠的接腳圖	9
附錄 G	PAC 的 SNTP 網路自動校時功能	11
G.1	設定 SNTP Client 的網路自動校時	11
G.2	設定一台 Windows XP 電腦為 SNTP Server 來進行測試	13
G.3	設定 PAC 的 Gateway 與 DNS Server 位址	17

第 1 章 軟體安裝與硬體設定

1.1 安裝 Win-GRAF 軟體

安裝 Win-GRAF 軟體前，請先確認您電腦中的安裝環境。

系統需求:

作業系統: Windows 7, Windows 8 (32-bit 或 64 bit)

Microsoft .Net Framework 3.5 (可在微軟官方網站下載:

<http://www.microsoft.com/zh-tw/download/details.aspx?id=22>)

RAM: 至少 1 GB (建議 2 GB)

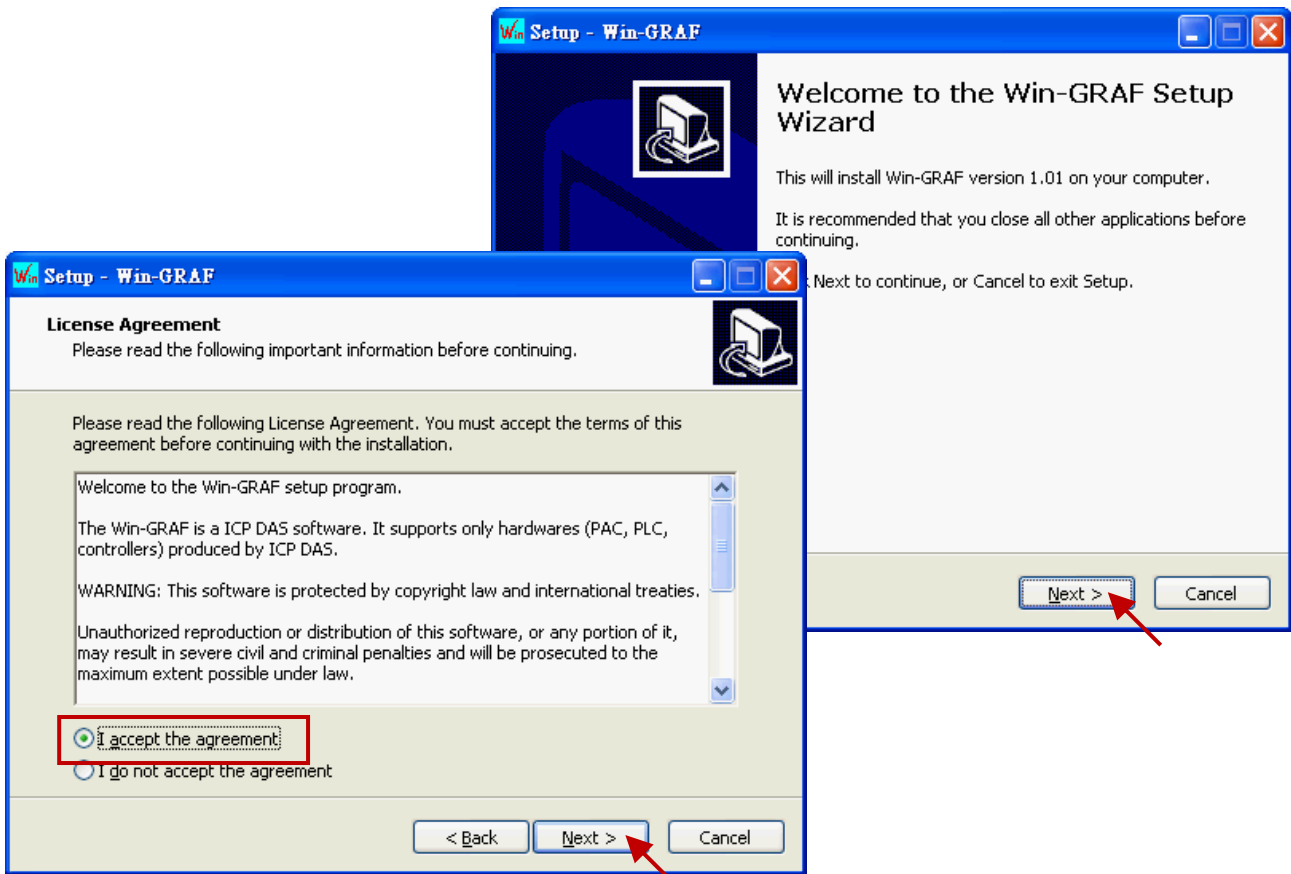
可用硬碟空間: 至少 200 MB

安裝步驟:

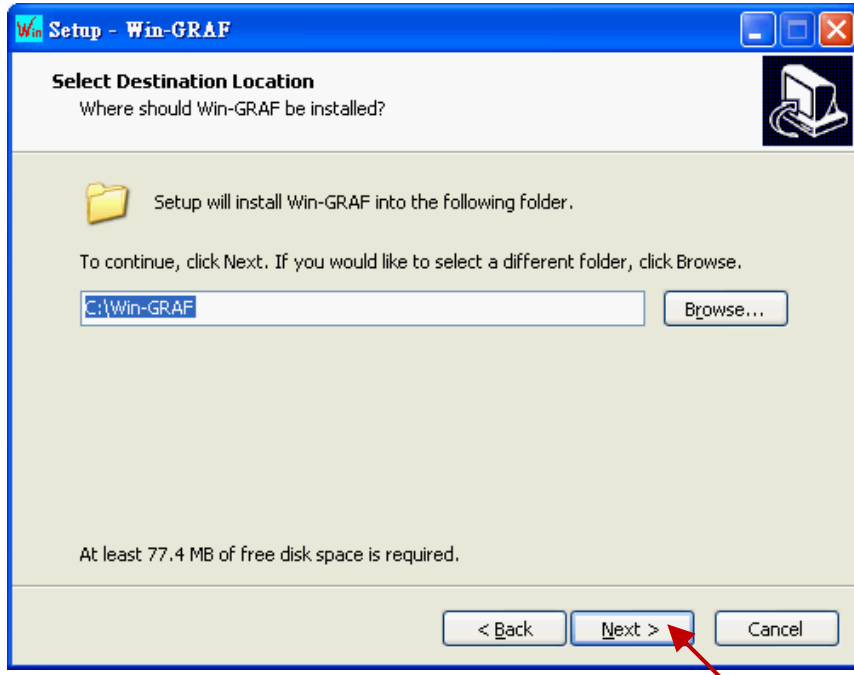
1. 於 Win-GRAF 安裝光碟中，滑鼠雙擊 “Win-GRAF-setup-ver-x.xx.exe” 檔開始安裝程序。
(或可上網下載最新版本的 Win-GRAF Workbench: http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-driver.html)



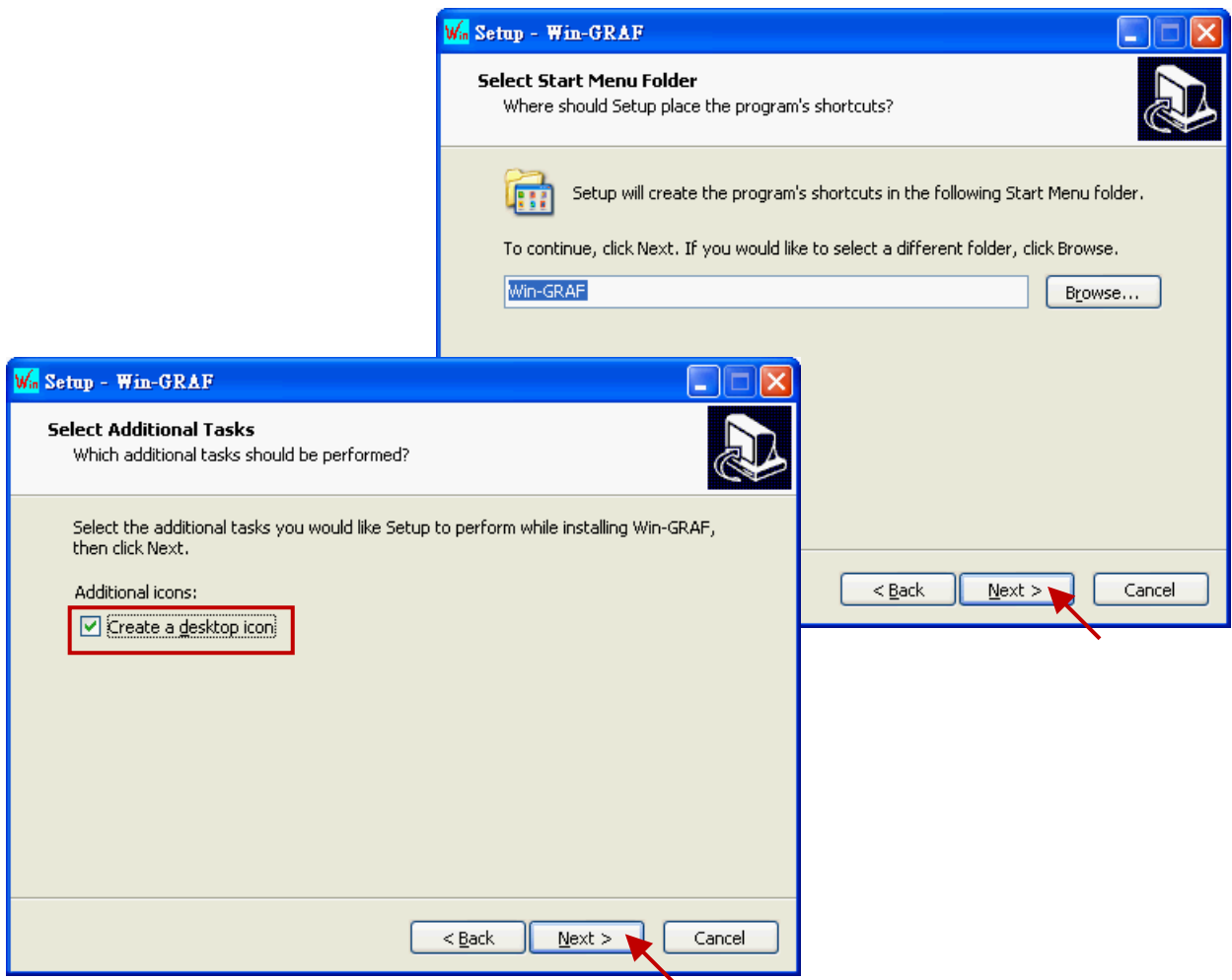
2. 點選 “Next” 進行下一步，接著選取 “I accept the agreement” 再點選 “Next” 繼續。



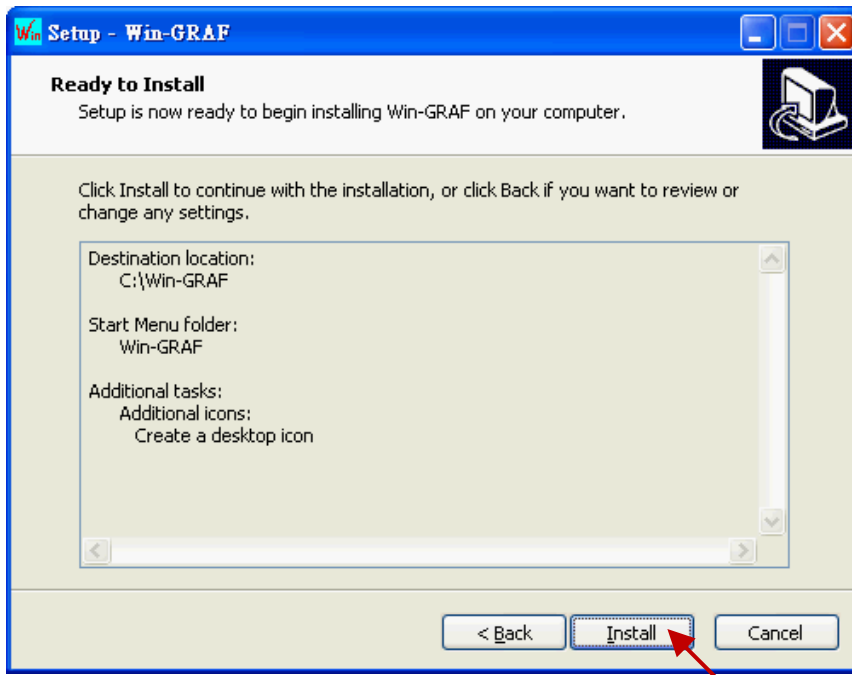
3. 建議使用預設的安裝目錄 (即 · “C:\Win-GRAF”) · 並點選 “Next” 進行下一步。



4. 點選 “Next” 於 “開始” 選單中建立 “Win-GRAF” 捷徑目錄，可勾選 “Create a desktop icon” 建立桌面捷徑，再點選 “Next” 進行下一步。

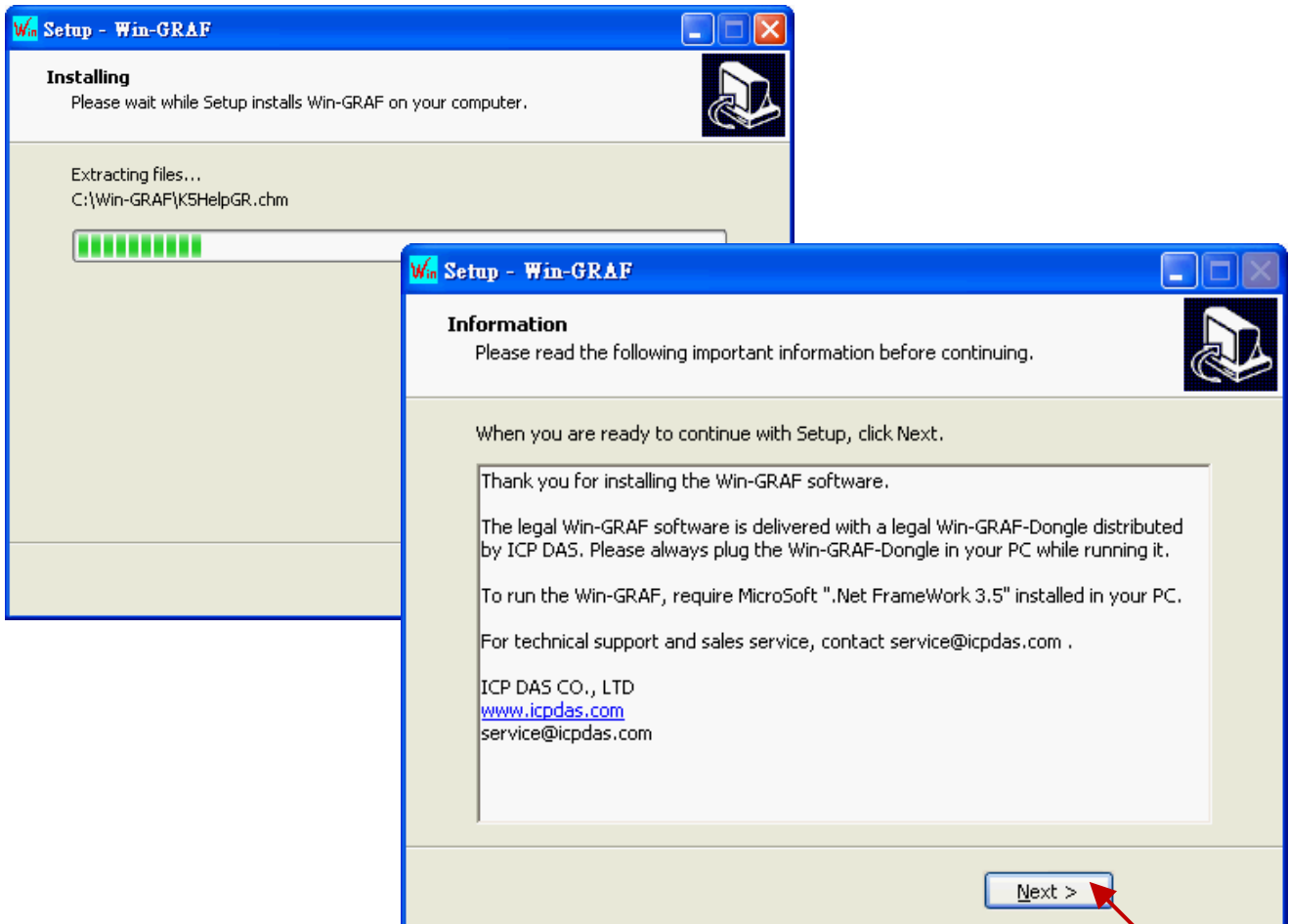


5. 點選 “Install” 開始安裝 Win-GRAF 軟體。



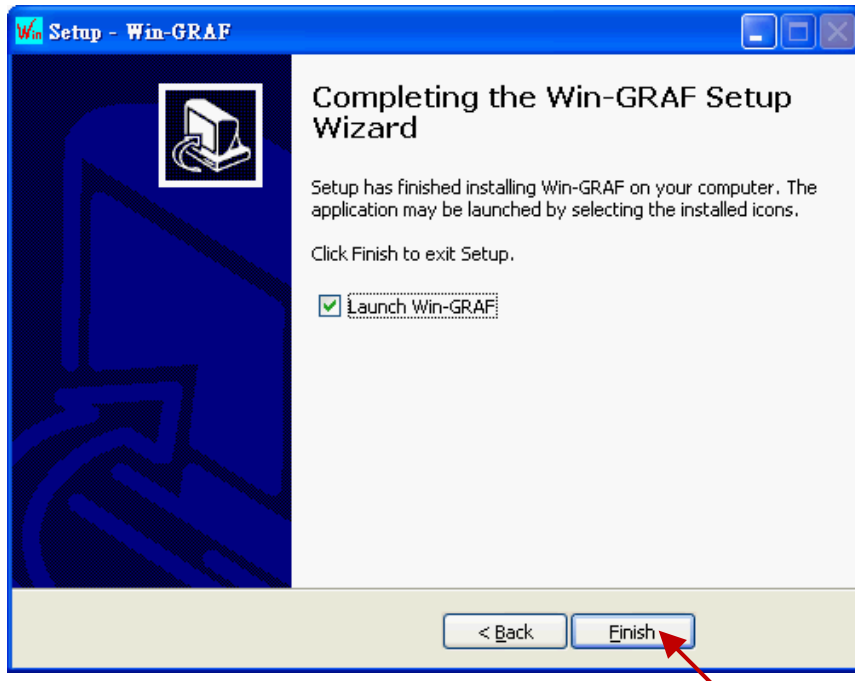
6. 等待安裝完成，您會見到以下訊息視窗，此處說明了：

- a. 泓格科技合法授權的 Win-GRAF 軟體配備有一個 USB 保護鎖 (Win-GRAF Dongle)，開啟該軟體前，請確認 USB 保護鎖已裝置在您的電腦中。
- b. 您的電腦必須安裝 “.NET FrameWork 3.5” 才能運行 Win-GRAF 軟體。



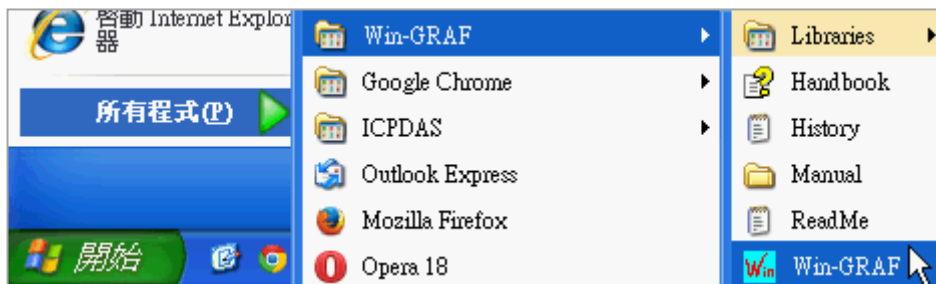
7. 您已完成 Win-GRAF 安裝，點選“Finish”離開此視窗。

(勾選“Launch Win-GRAF” 將會自動開啟此軟體，請確認您的 PC 已裝置了 Win-GRAF Dongle 。)



1.2 開啟 Win-GRAF 軟體

開啟軟體前，請確認您的 PC 已裝置了 USB 保護鎖 (Win-GRAF Dongle)。PC 若沒有插上 Win-GRAF Dongle，則 Win-GRAF Workbench 將以 Demo Mode 來運行。您可在“開始”選單中，點選“Win-GRAF”目錄再點選“Win-GRAF”來開啟軟體。



Win-GRAF 目錄說明:

Libraries: 可供使用者建立自訂的函式 或 修改內建的函式。

Handbook: 原廠提供關於軟體介面、編程環境、編程語言...等詳細說明。

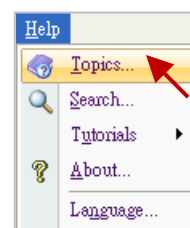
(或 Win-GRAF 的功能表 [Help] > [Topics])

History: 記錄 Win-GRAF 軟體更新內容的文字檔。

Manual: ICP DAS 提供的 Win-GRAF 相關手冊。

(或 Win-GRAF 的功能表 [Help] > [Tutorials]，存放於 C:\Win-GRAF\Tutorials)

ReadMe: 關於 Win-GRAF 版權相關資訊。



1.2.1 軟體操作模式

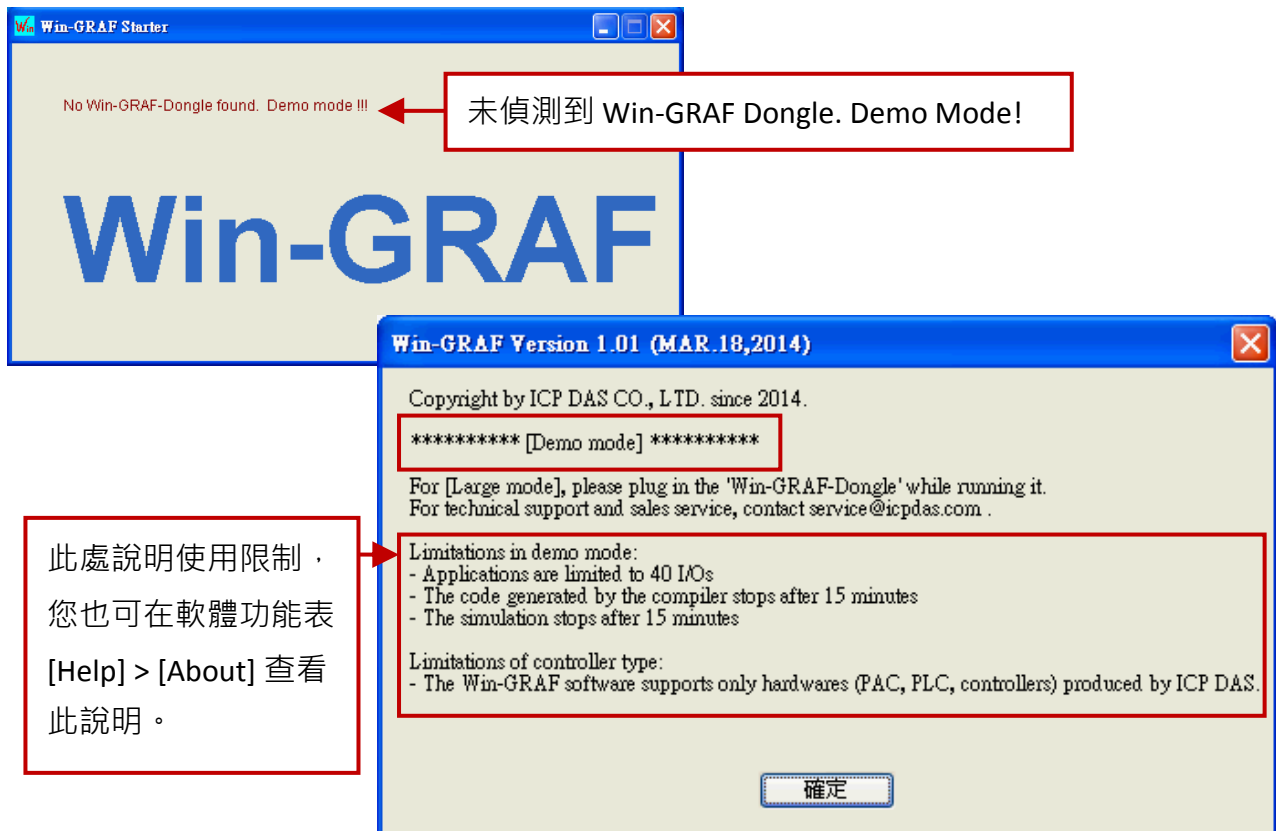
Win-GRAF 提供了兩種操作模式:

“Demo Mode”: PC 未裝置 USB 保護鎖，表示 Win-GRAF 編譯出來的專案 (Project) 只能在 PAC 內運行 15 分鐘，時間超過後需再重新下載專案到 PAC 中，且只能使用 40 個 I/O 點。

“Large Mode”: PC 有裝置 USB 保護鎖，表示 PAC 內的專案可一直 Run 沒有時間限制。

Demo Mode

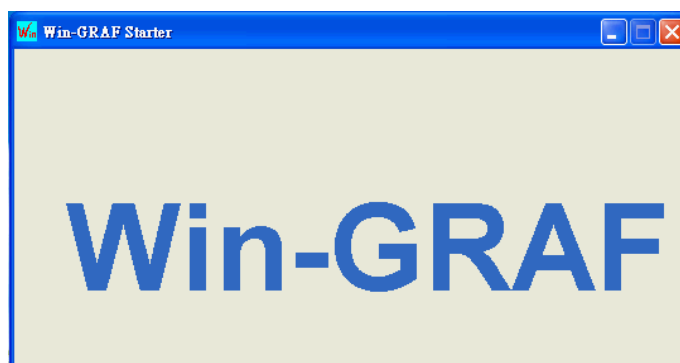
若您的 PC 未裝置 Win-GRAF Dongle，則軟體開啟時會先顯示以下畫面:



註: 若於 [Demo Mode] 下裝置 Win-GRAF Dongle，需重新開啟 Win-GRAF 軟體來進入 [Large Mode]。

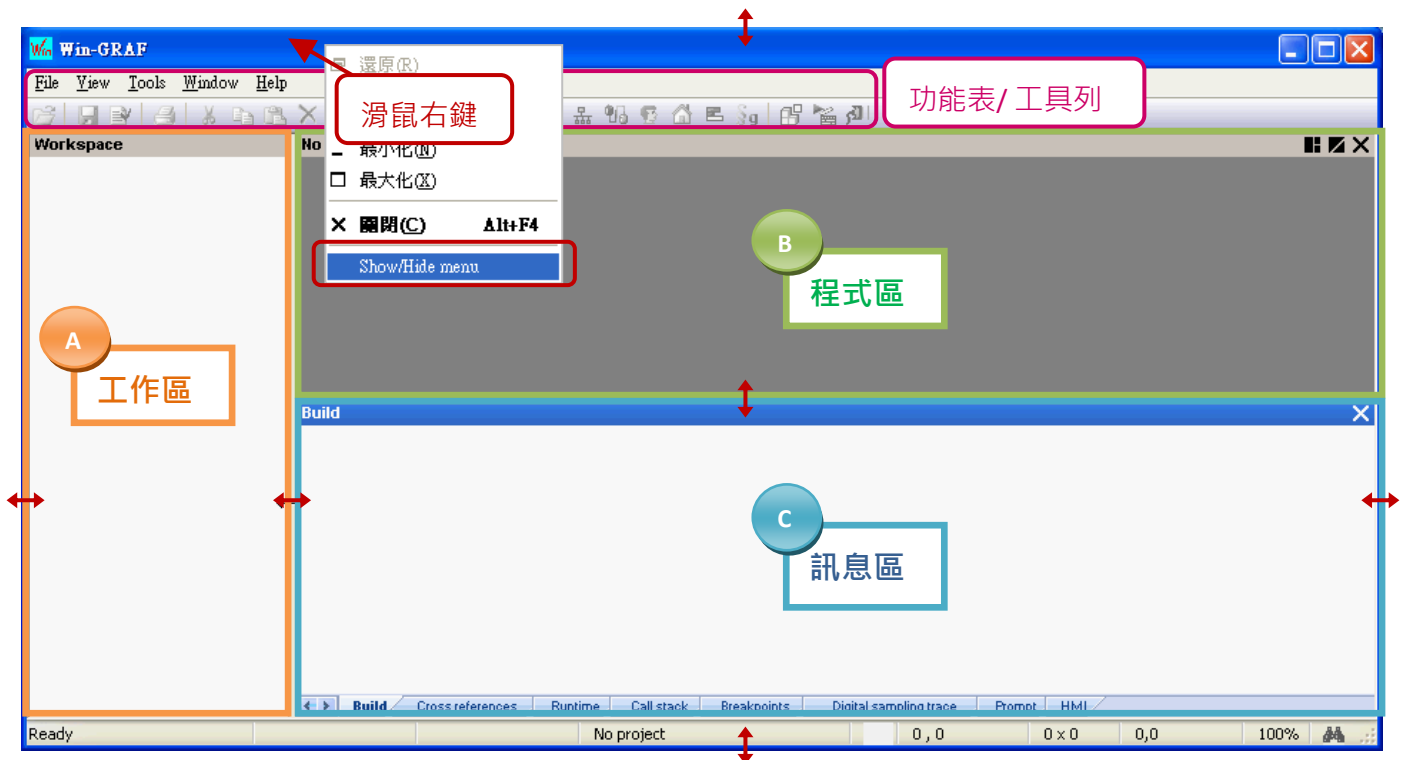
Large Mode

若您的 PC 已裝置 Win-GRAF Dongle，則軟體開啟時會先顯示以下畫面:



1.2.2 軟體介面說明

開啟 Win-GRAF 後，畫面如下：



註：滑鼠右鍵點選視窗最上方可顯示/隱藏功能表。

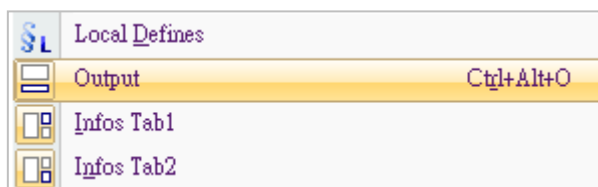
- A. 工作區: 用來建立/開啟 Win-GRAF 程式的清單。
- B. 程式區: 用來顯示/編輯程式，可再分割為多個功能區塊 (參考 [2.2.1 節](#))。
- C. 訊息區: 用來顯示編譯訊息，並提供多個診斷工具。

使用小技巧:

1. 可使用滑鼠拖曳的方式，任意調整視窗的大小。
2. 可在工作區上按“F1”鍵，開啟使用說明。

回復功能視窗

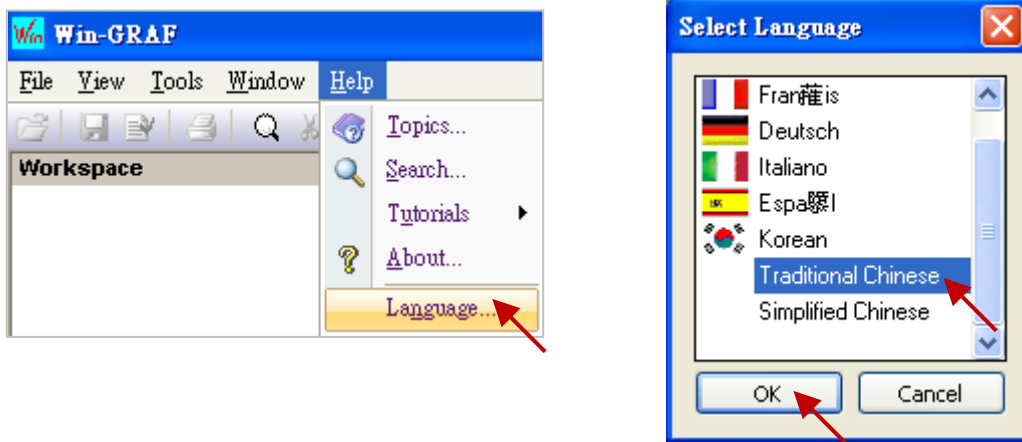
若操作中不慎關閉了程式區的變數視窗 或是 訊息區，可點選 **功能表 "View"** 再選取下列選項來回復視窗。



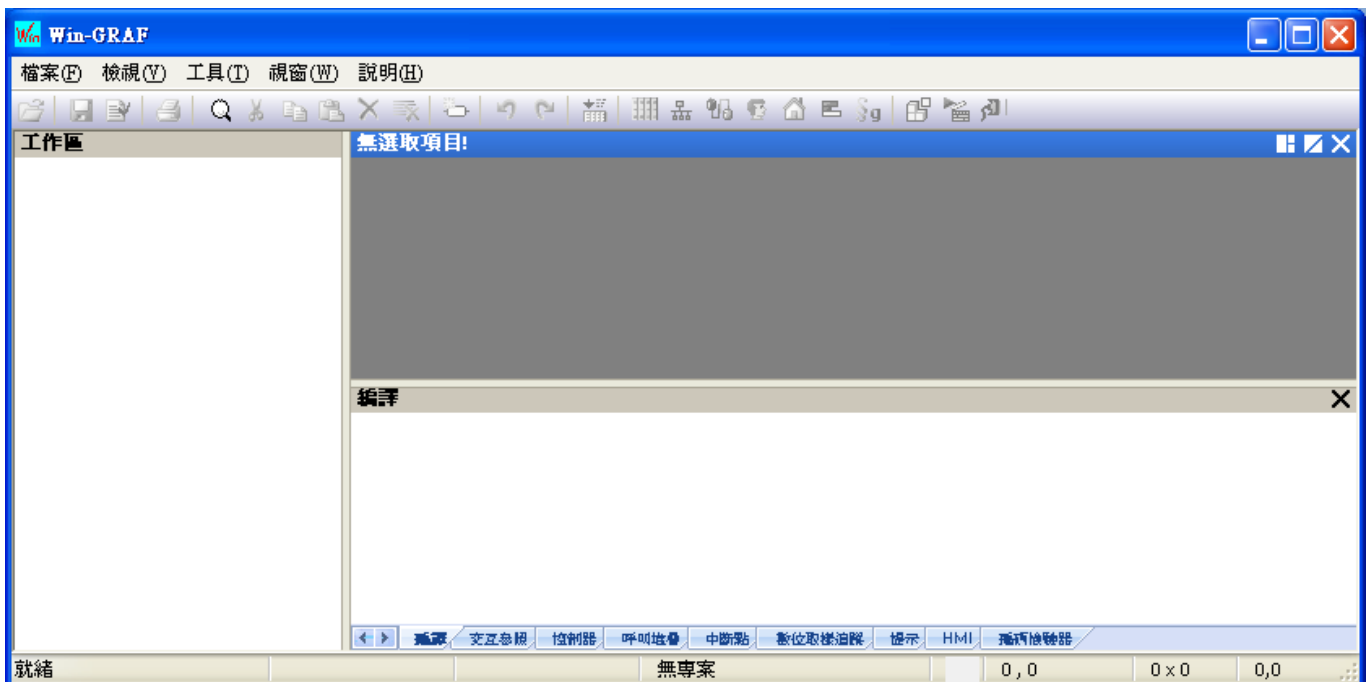
- Output: 表示訊息區的視窗。
- Infos Tab1: 表示程式區 - 變數視窗 (參考 [2.2.1 節](#))。
- Infos Tab2: 表示程式區 - 功能方塊視窗 (參考 [2.2.1 節](#))。

切換中文介面

1. 若想把 Win-GRAF Workbench 改成中文顯示，請點選工具列 “Help” 再點選 “Language”，接著在 “Select Language” 選擇中文語言 並按 “OK”。



2. Win-GRAF 會自動重新開啟，之後即可見到軟體介面已變成中文。



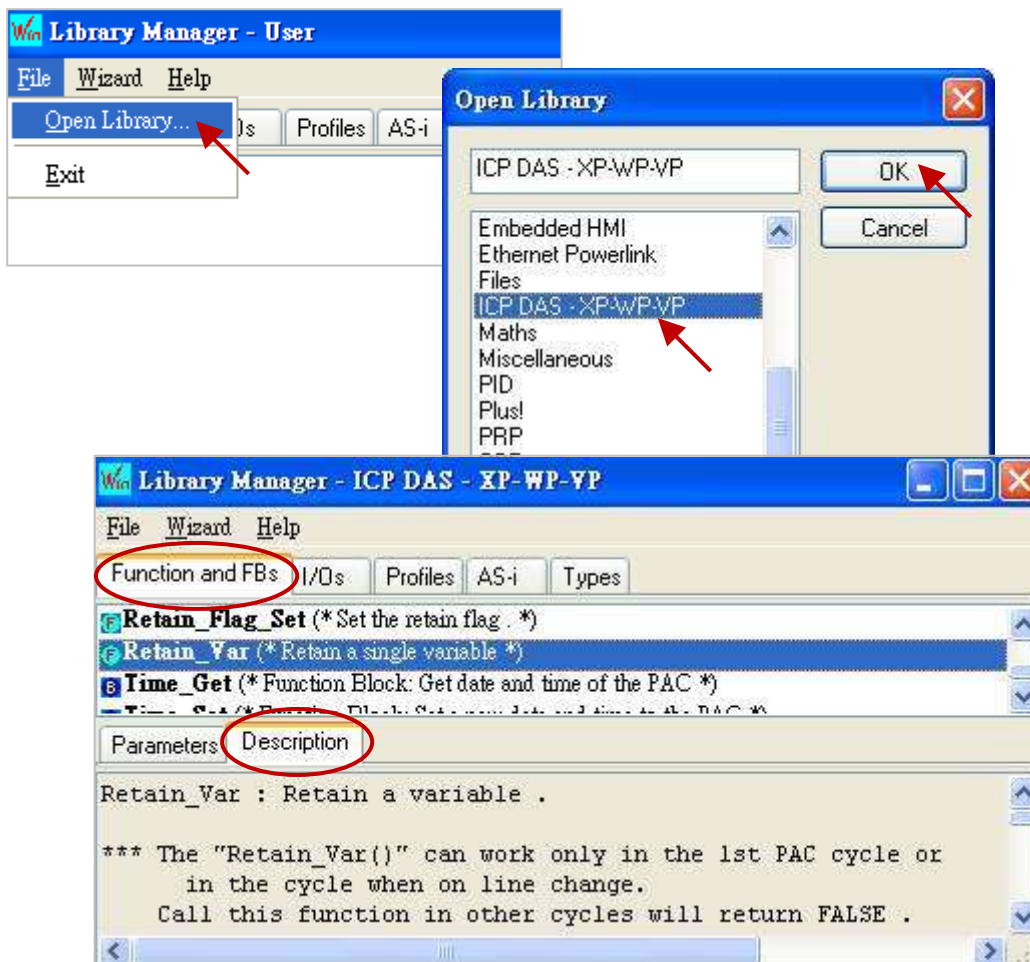
1.2.3 程式庫管理員 (Library Manager) 功能

Win-GRAF Workbench 提供了程式庫管理員 (Library Manager) 功能，可用來查詢各函式 (Function)、功能方塊 (Function Block) 與 I/O 卡 (I/O Board) 的功能說明，若想更新 Win-GRAF Lib 請參考 [11.1 節](#)。

1. 於 PC 的“開始”選單中，點選“Win-GRAF”目錄再點選“Libraries”>“OEM”。



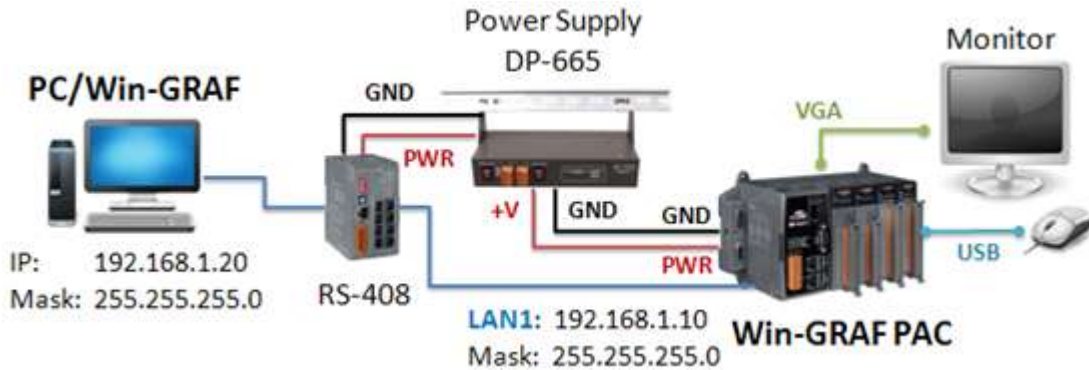
2. 於“Library Manager”視窗中，點選功能表“File”>“Open Library”並選擇“ICP DAS – XP-WP-VP”再點選“OK”。
3. 可在“Function and FBs”頁籤，點選下方的“Description”來查看各函式 (Function) 或功能方塊 (Function Block) 的說明；點選“I/Os”頁籤再點選下方的“Description”則可查看 I/O 卡 (I/O Board) 的功能說明。



1.3 設定 Win-GRAF PAC 的 IP 位址

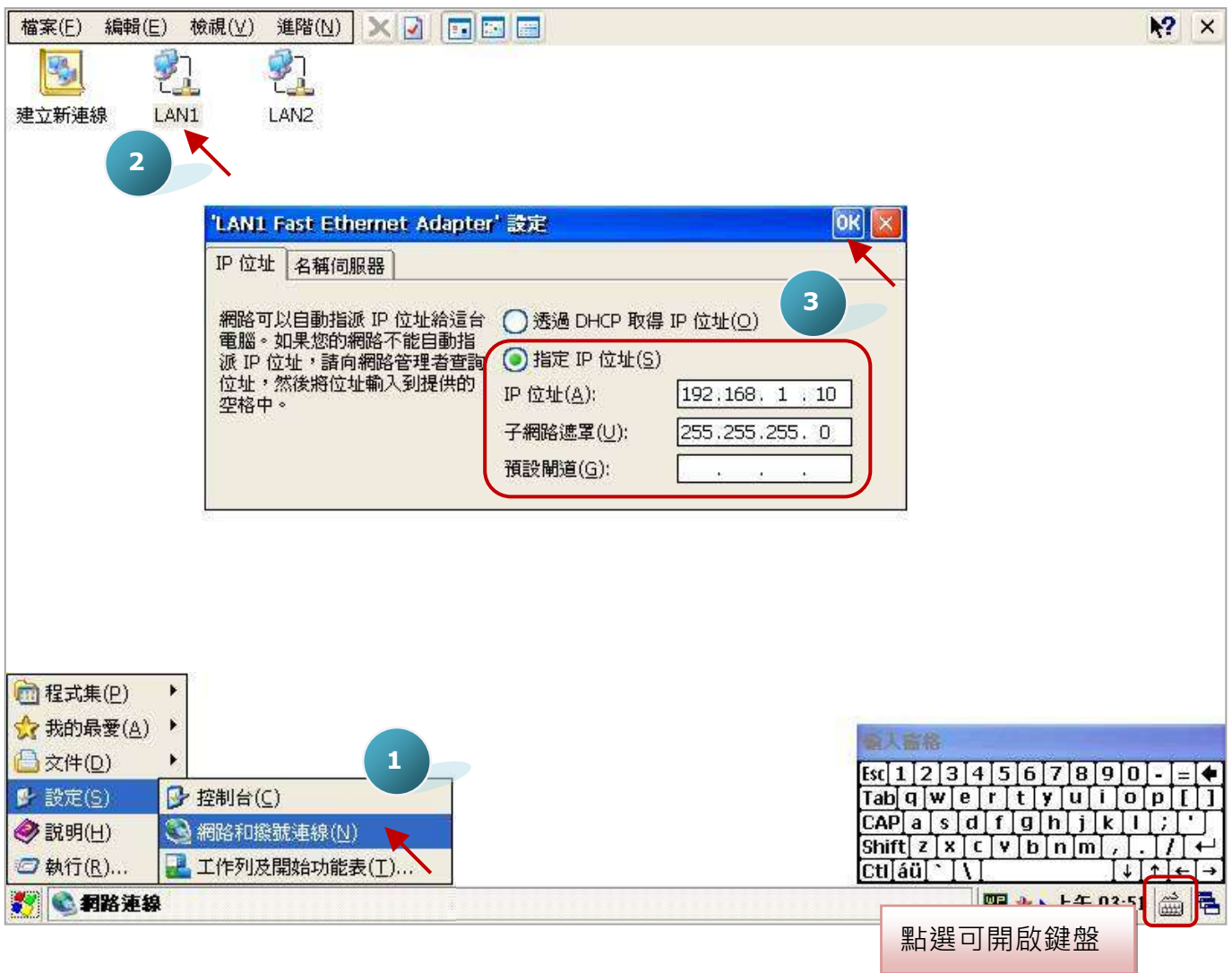
Win-GRAF Workbench 需知道 PAC 的 IP 位址才能與之連線，以下將說明 PAC 端的 IP 設定。
以 XPAC (XP-8xx8-CE6) 與 WinPAC (WP-8x48, WP-5xx8-CE7, WP-8x28-CE7) 為例：

硬體連接圖



PAC 端

使用連接在 PAC 上的 USB Mouse 點選螢幕左下角的“開始”>“設定”>“網路和撥號連線”，再雙擊 LAN1 (或 LAN2)，並輸入適當的 IP 位址。

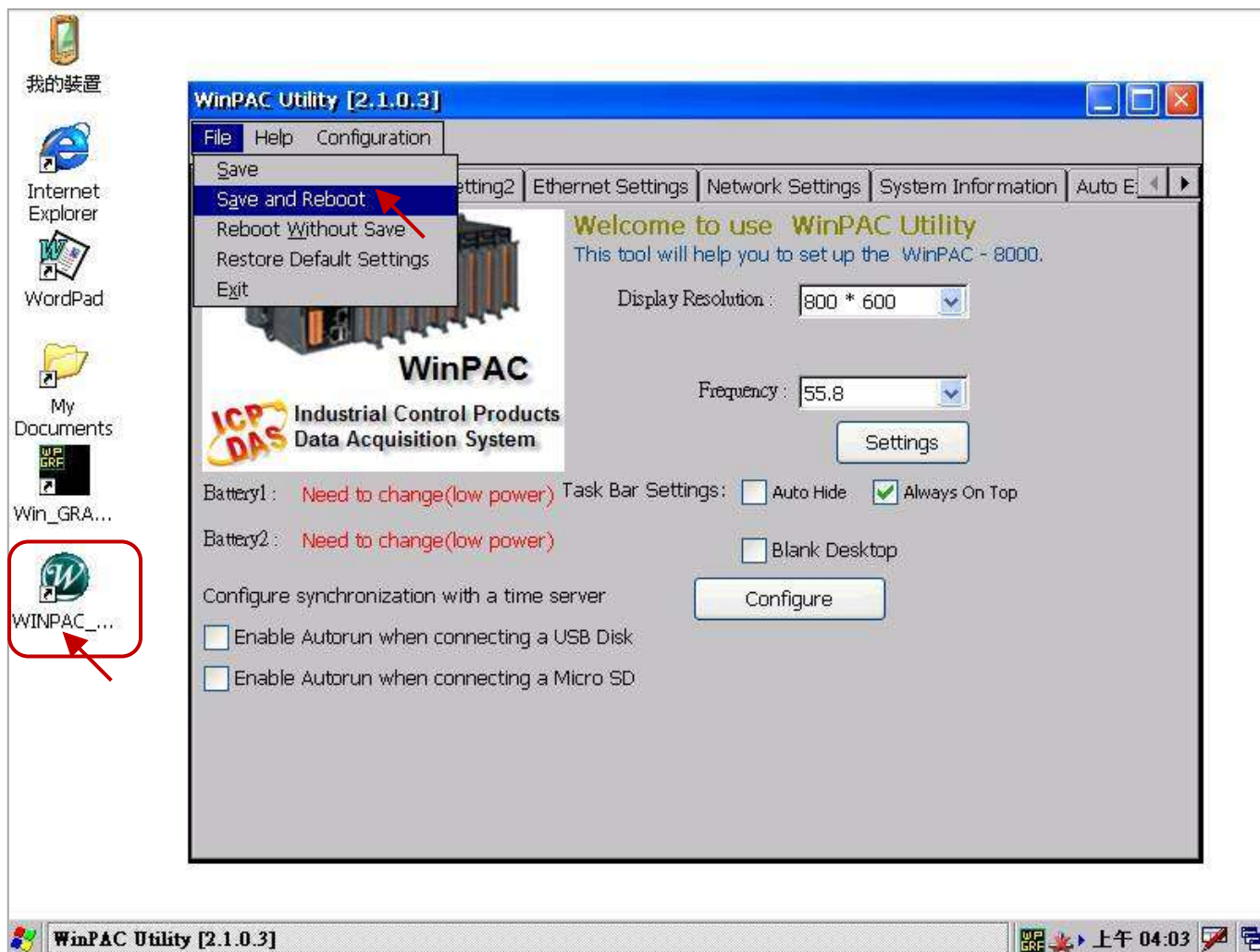


於 PAC 桌面上 (或 \System Disk\tools\ 路徑下) · 開啟 “WinPAC_Utility.exe” (或 “XPAC_Utility.exe”) · 再點選 “File” > “Save and Reboot” 重新啟動 PAC 。

注意:

PC/Win-GRAF 的 IP 與 PAC IP 需在同一個 IP 網段內 · 才連的上。

例如: PC 的 IP 設定為 192.168.1.20 (Mask: 255.255.255.0) 。



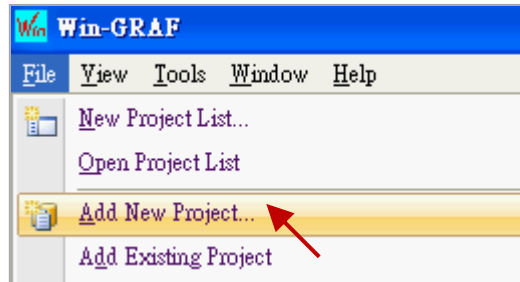
第 2 章 編寫一個簡單的 Win-GRAF 範例

本章將介紹一個簡單的樣版專案，可用來讀取或寫入 Win-GRAF PAC 中的系統時間。請依照以下步驟，一步步地實現此範例。

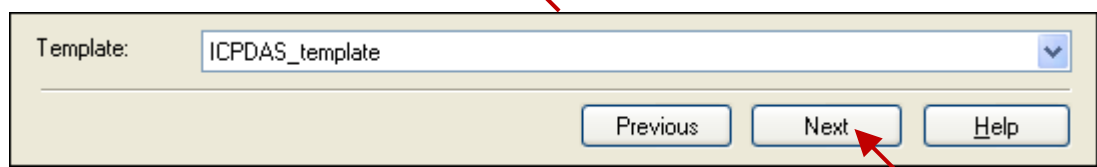
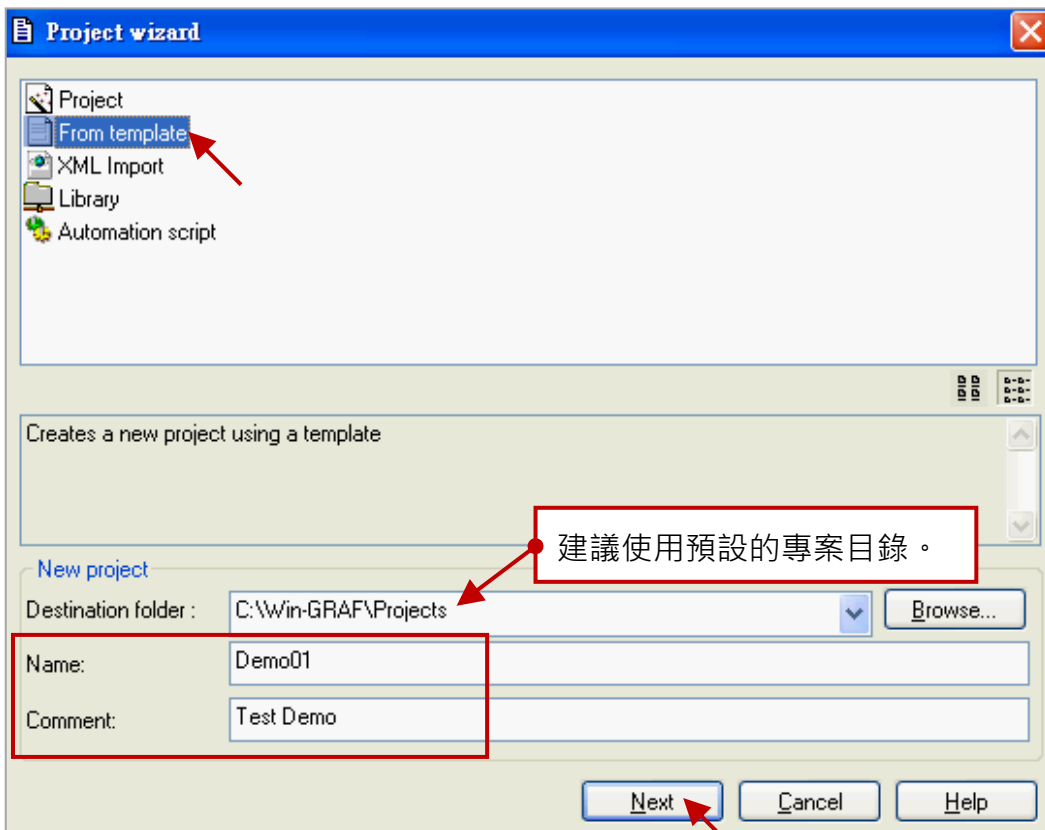
2.1 建立 Win-GRAF 專案

2.1.1 建立樣版專案 (Demo01)

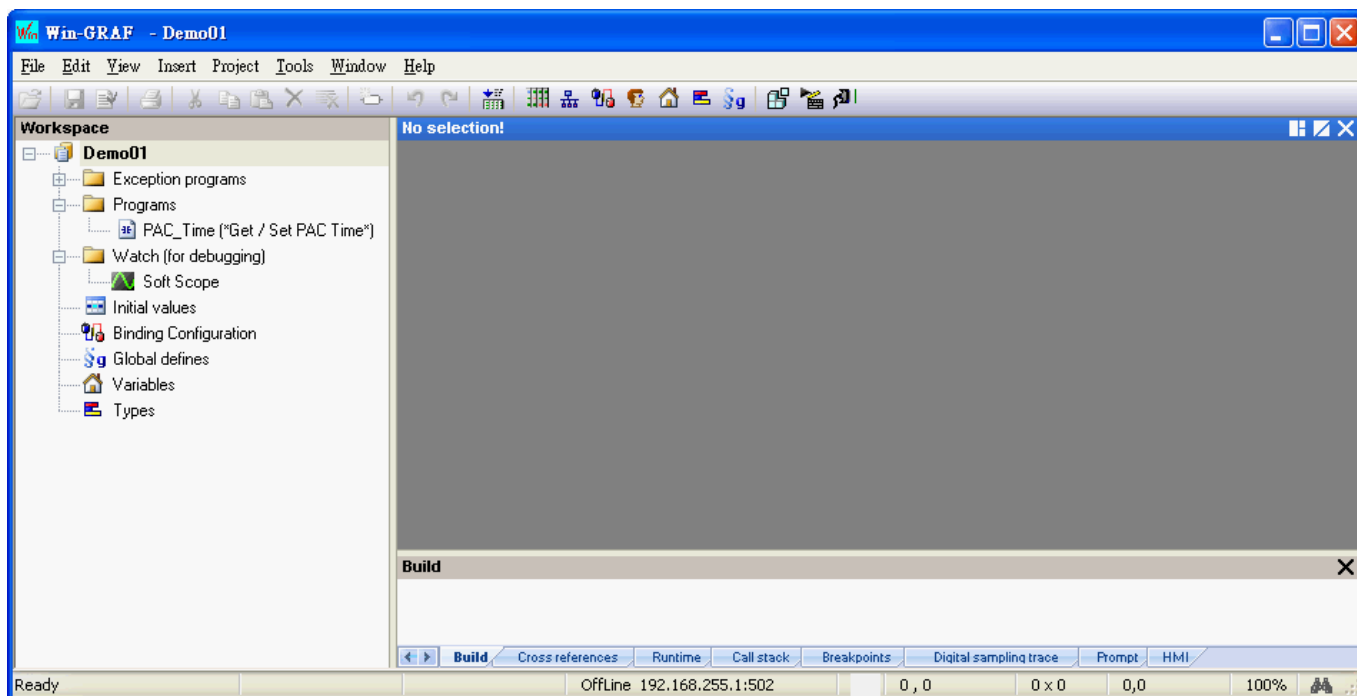
1. 開啟 Win-GRAF 軟體後 (可參考 [1.2 節](#))，點選 "File" 功能表再選擇 "Add New Project..." 選項。



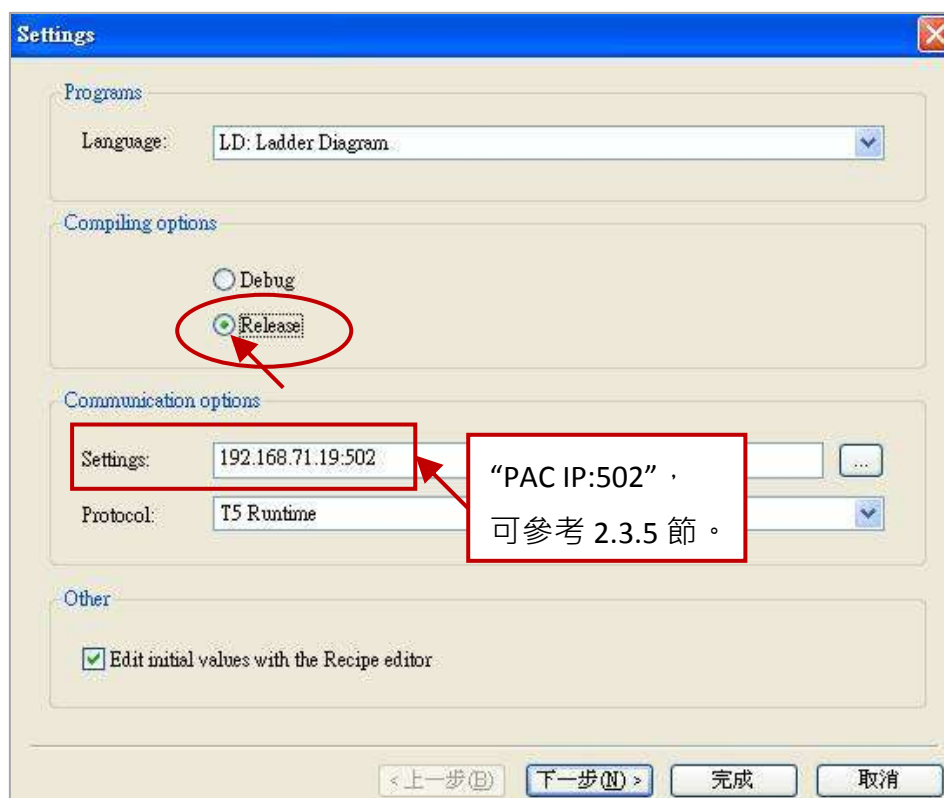
2. 點選 "From template" 來建立一個套用樣版的專案，於 "Name" 欄位中輸入專案名稱 (例如: Demo01)，並在 "Comment" 欄位中輸入此專案的註解，點選 "Next" 後，會出現 Win-GRAF 提供的預設樣版 "ICPDAS_template"，請直接點選 "Next" 進行下一步。



3. 您已建立一個 “Demo01” 樣版專案。



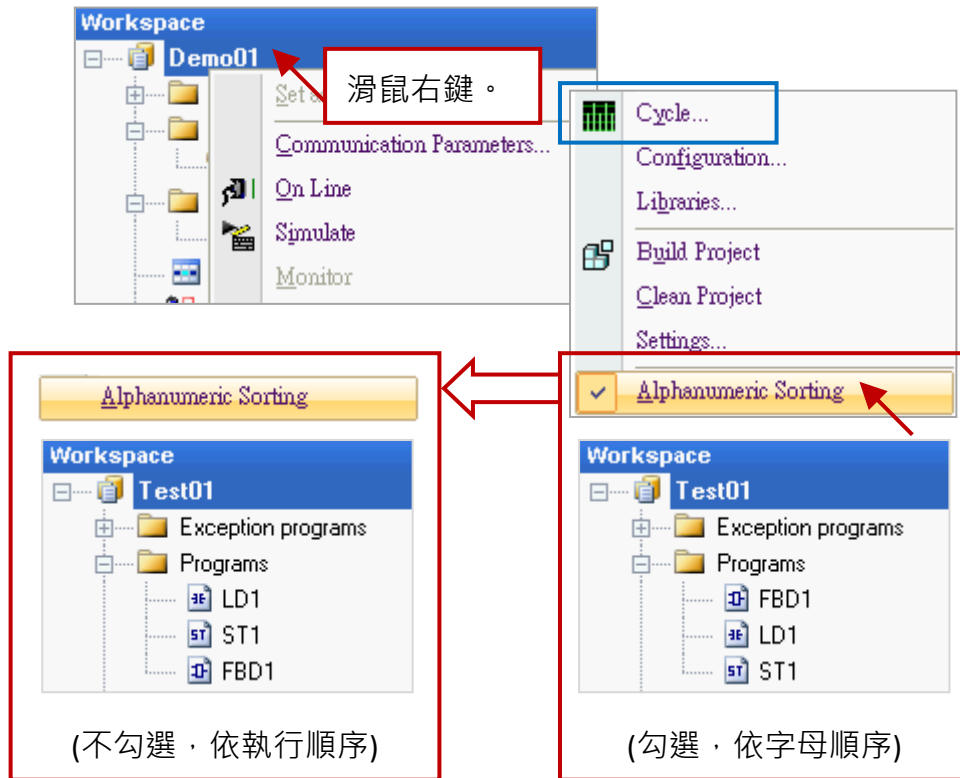
注意: 本範例選用 “From template” 方式來建立新專案，若您在步驟 2 選擇了 "Project" 方式，
“Compiling options” 請選擇 "Release"，其餘項目可在後續章節中進行設定，請點選 "下一步"
再點選 "完成"。



2.1.2 重要專案設定

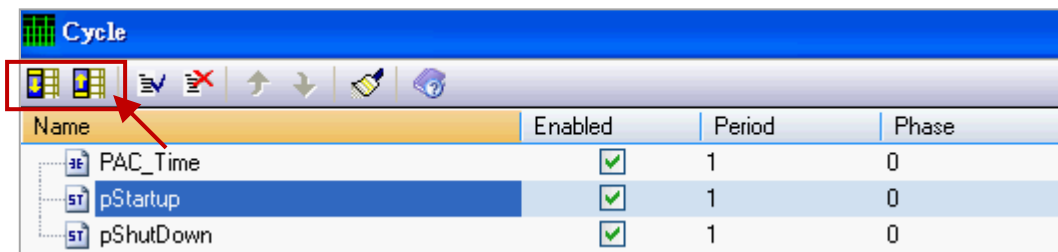
建立專案後，有兩個需先完成的重要設定。

1. 在 "工作區" 中，滑鼠右鍵點選專案名稱 (例如: Demo01)，並取消勾選 "Alphanumeric Sorting" (即最後一個選項)。若不勾選，表示依執行順序來排列程式。若勾選，表示依字母順序來排列程式 (例如: FBD1, LD1, ST1)。

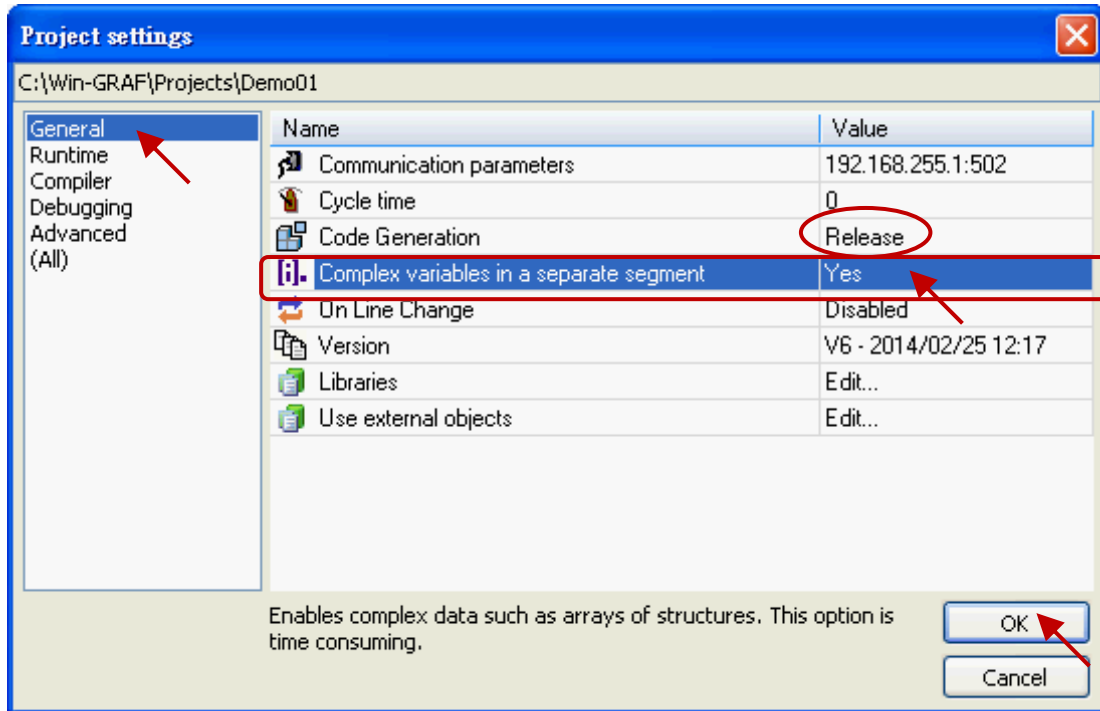
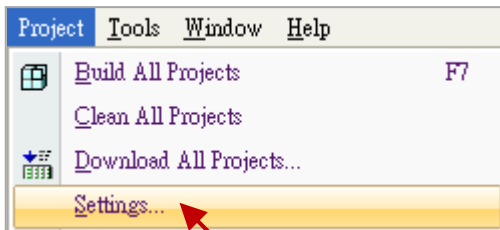


修改程式的執行順序:

注意: 若您需修改程式的執行順序，(如上圖) 滑鼠右鍵點選專案名稱 (例如: Demo01)，再點選 "Cycle" 選項來開啟設定視窗，再點選 "Move Up" 或 "Move Down" 按鈕來挪動順序。



2. 若選用 "Project" 方式 (見 [2.1.1 節](#) - 步驟 2，本例選用 "From template" 方式) 來建立新專案，請點選功能表 "Project" > "Settings..." 開啟專案設定視窗 (如下圖)，再點選 "General" 選項，並將 "Complex variables in a separate segment" 設定為 "Yes"，以允許使用像是陣列的資料型態，再點選 "OK" 離開設定視窗。

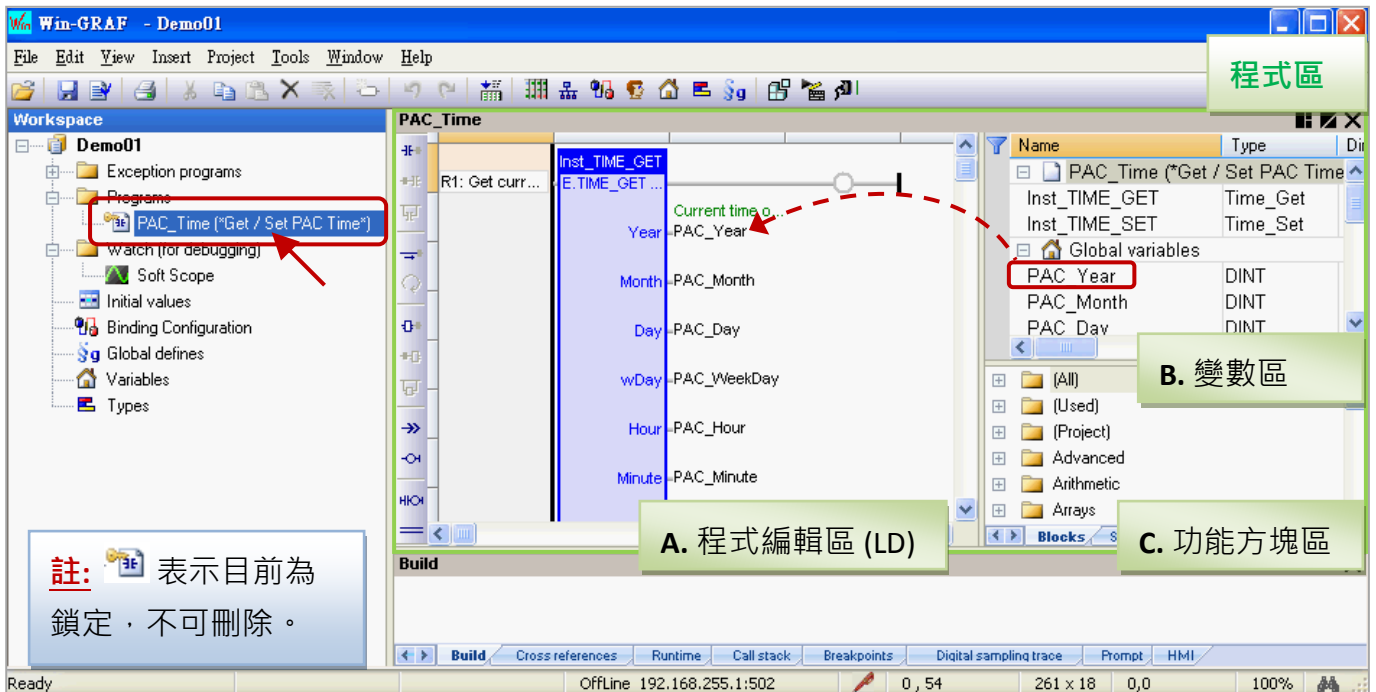


注意: "Code Generation" 需設定在 "Release" 。

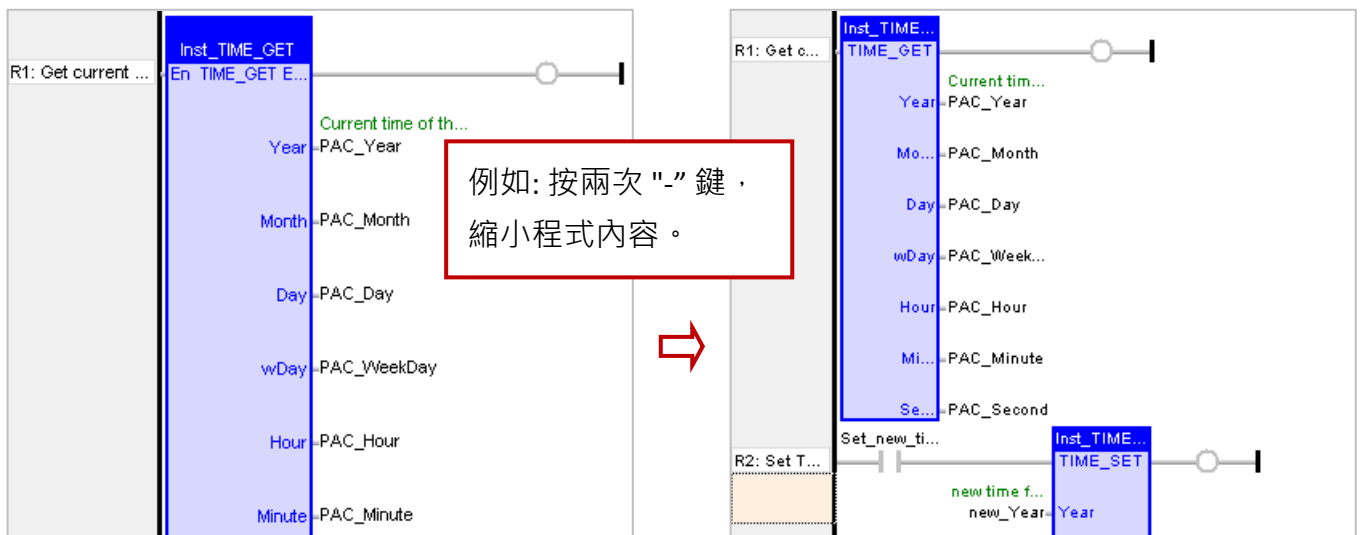
2.2 專案介紹

2.2.1 Demo01 - LD 程式

此程式用來讀取或寫入 Win-GRAF PAC 中的系統時間。於工作區中，滑鼠雙擊 LD 程式名稱 (即，"PAC_Time") 來開啟相關的視窗。如畫面中，程式區 劃分為 3 個區塊：



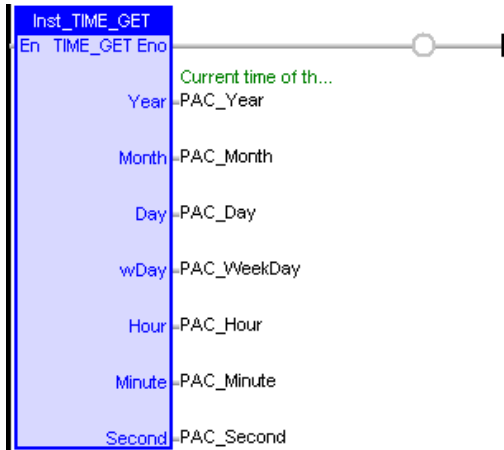
使用小技巧: 滑鼠點選一下程式編輯區，再按 "+" 或 "-" 鍵，可放大或縮小程式內容。



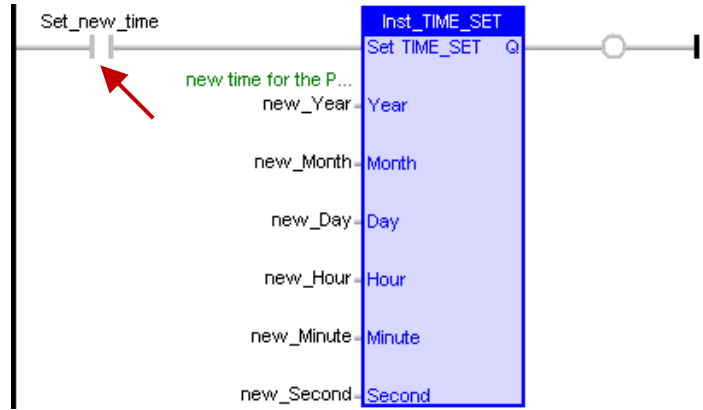
A. 程式編輯區 (LD):

此區塊用來編輯/顯示 LD 程式，您可點選左邊的元件按鈕來新增程式，也可用滑鼠拖曳方式將定義好的變數指定給功能方塊。

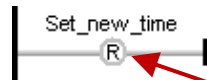
R1: 讀取 PAC 目前時間



R2: 將 "Set_new_time" 設為 "True"，可寫入新的時間



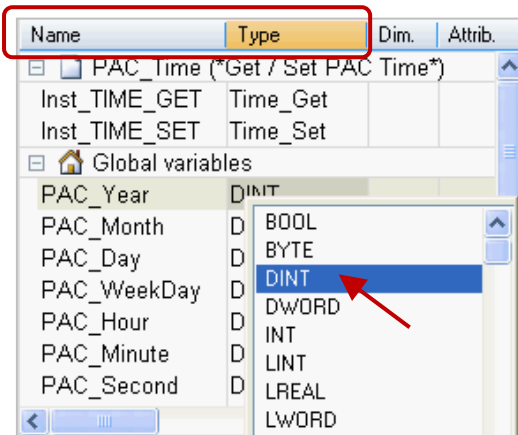
R3: 重置為 "False" 狀態



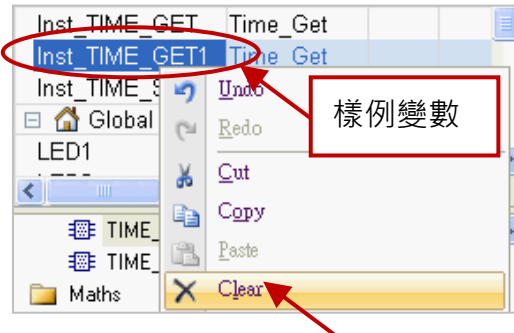
註: 您可點選空白鍵來變更其型態 (/, S, R)，按 F1 鍵可取得詳細說明。

B. 變數區:

此區塊顯示了程式中使用的功能方塊與變數。滑鼠雙擊任一 "Name" 或 "Type" 項目，可修改其名稱或資料型態，再按 "Enter" 鍵完成修改。關於變數的定義方式，請參考 [2.3.1 節](#)。

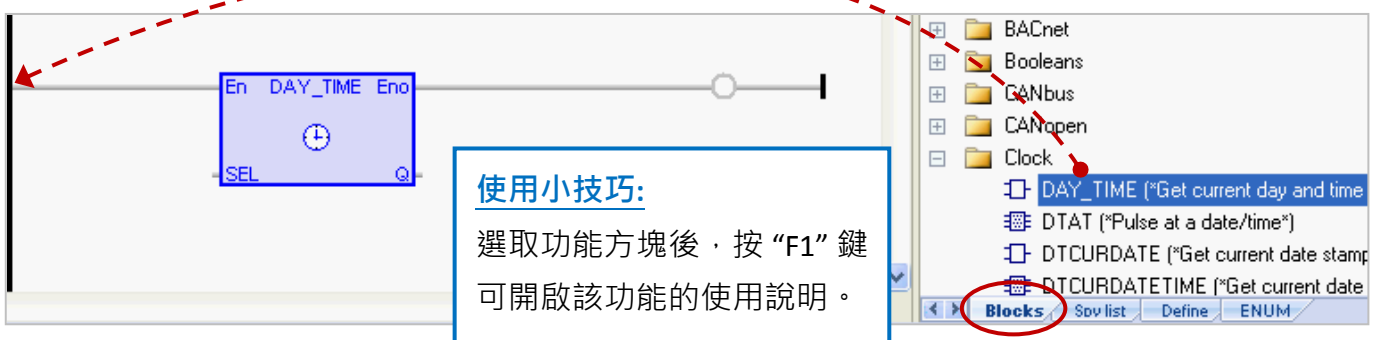


註: 程式內每增加一個功能方塊都會自動建立一個 "Inst_xxx.." 樣例變數，這樣才能正確使用功能方塊。但若刪除程式的功能方塊時，為了安全並不會自動刪除變數區的樣例變數，因此使用者可用滑鼠右鍵點選名稱再選擇 "Clear" 來刪除不需使用的樣例變數。



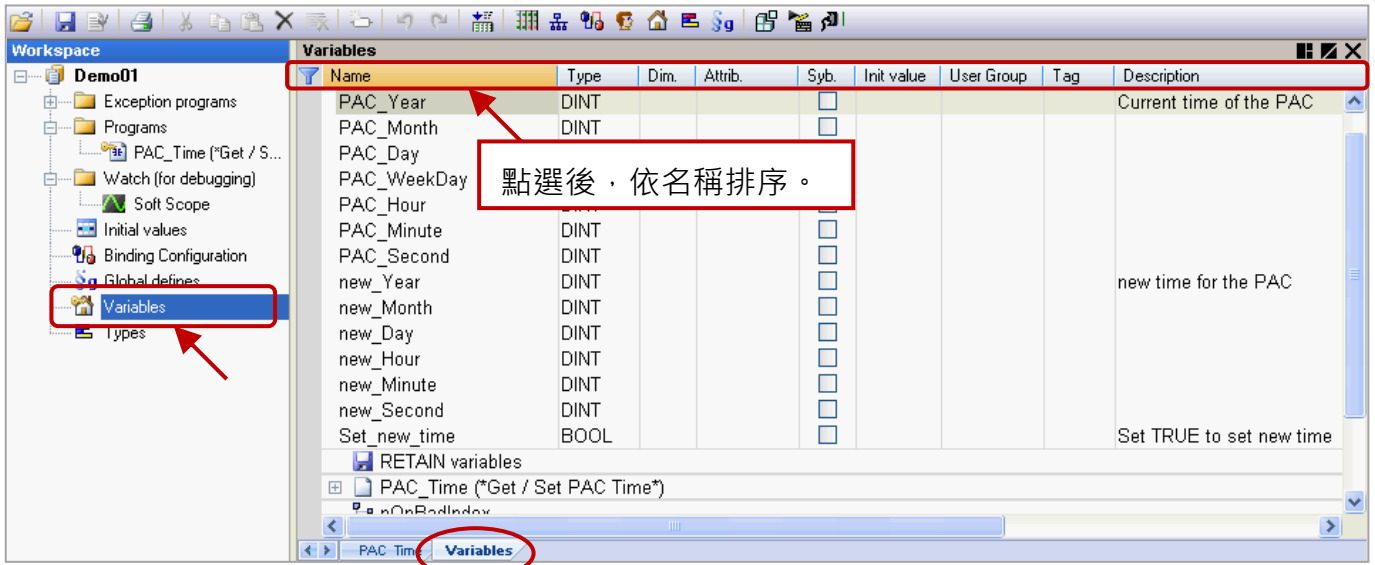
C. 功能方塊區:

此處 ("Blocks" 頁籤) 提供了許多類型的功能方塊，您可依需求將功能方塊拖曳到程式的編輯區。



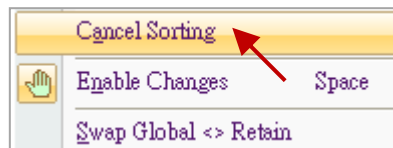
2.2.2 Demo01 - 變數說明

工作區中，滑鼠雙擊“Variables”項目來開啟變數視窗，此處列出了所有“Demo01”專案中需使用、已定義好的變數。



使用小技巧: 在變數視窗中，可點選任一標題欄位 (例如：“Name”) 來進行排序。

若要恢復原有排序，以滑鼠右鍵點選視窗內任一處並選擇“Cancel Sorting”。



欄位說明: (可按“F1”鍵，查看詳細說明)

可用滑鼠雙擊任一欄位項目，來進行設定/修改。

- Name:** 變數名稱，無大小寫的區別，都視為相同可使用的字元。
名稱只能是 A~Z (或 a~z)、0~9 與 “_” (底線字元)，而且第一個字必需是英文字元。
- Type:** 資料型態。(數值範圍，請參考 [附錄 A](#))
- Dim.:** 可供陣列使用 (例如: 輸入 “10” 來表示 Counter [0] ~ Counter [9])。
- Attrib.:** 若滑鼠雙擊此欄位，會設定為 “Read Only” 表示該變數只能讀取，不可變更值。
- Syb.:** 若勾選，會將變數名稱也下載至 PAC 中。
- Init value:** 可輸入變數的初始值。
- User Group:** 可將變數分成群組 (例如: 群組 1, 2)，以方便查看、搜尋變數。
- Tag:** 可輸入易於辨識的名稱 (Nick name)。
- Description:** 可輸入變數的用途說明。


2.3 小試身手

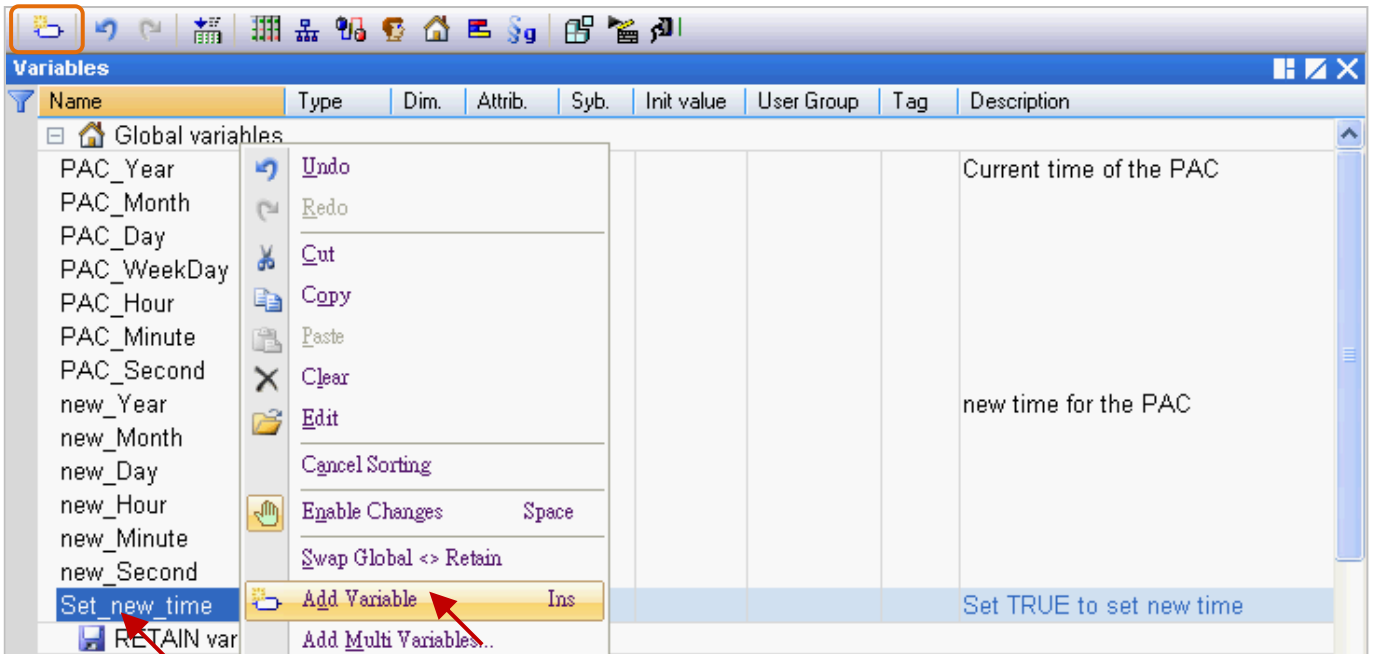
在前述章節中，分別介紹了“Demo01”專案中的 LD 程式 (2.2.1 節) 與 變數說明 (2.2.2 節)，本章節將在此專案中練習如何宣告變數與新增一個提供閃爍功能的 LD 程式。

注意：所有的 Win-GRAF PAC 皆不支援“ULINT”與“LWORD”資料型態。

2.3.1 宣告專案變數

首先，我們將宣告 2 個布林變數 (即，LED1 與 LED2) 可供程式使用。

1. 在“Variables”視窗中，滑鼠右鍵點選任一個“Name”項目，再選擇“Add Variable”或按鍵盤“Ins”或點選工具按鈕  來新增一個變數。



2. 滑鼠雙擊新增的“NewVar”項目將名稱修改為“LED1”，再按“Enter”完成設定。
此例，資料型態為“BOOL”。

Name	Type	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description
Set_new_time	BOOL				<input type="checkbox"/>			Set TRUE to set new time
NewVar	BOOL				<input type="checkbox"/>			
LED1	BOOL				<input type="checkbox"/>			

注意：所有的修改都必須按“Enter”才能完成設定。

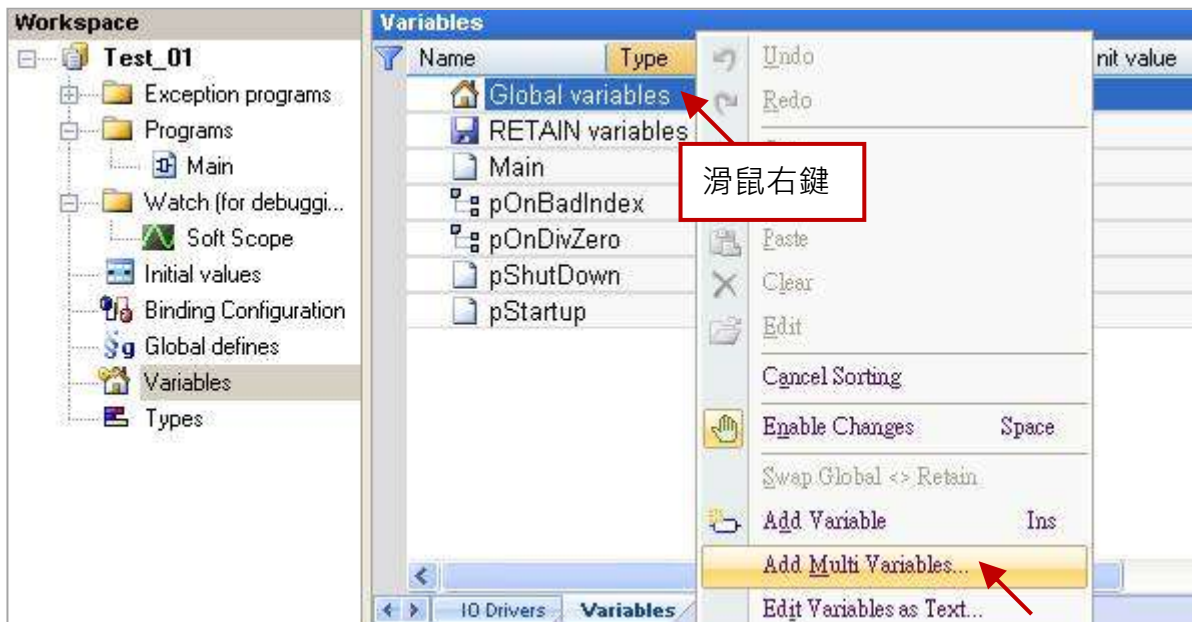
3. 如同上述步驟，請再新增“LED2”布林變數。

Name	Type	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description
Set new time	BOOL				<input type="checkbox"/>			Set TRUE to set new time
LED1	BOOL				<input type="checkbox"/>			
LED2	BOOL				<input type="checkbox"/>			

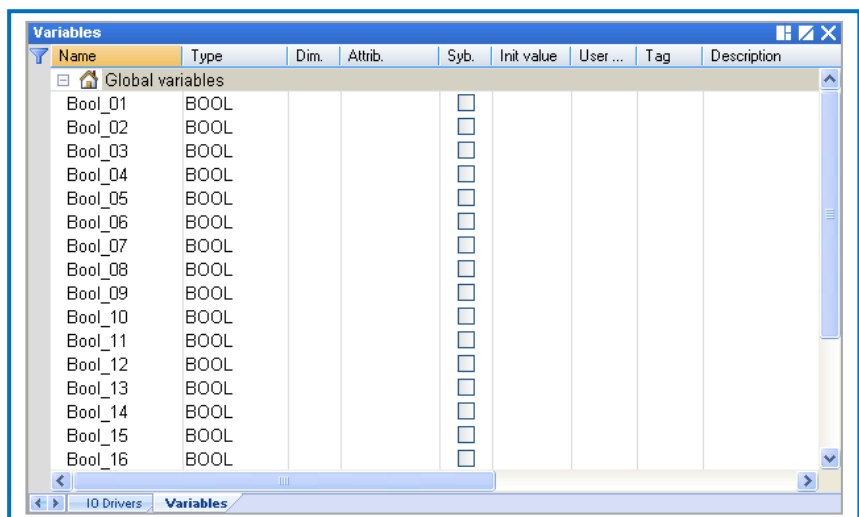
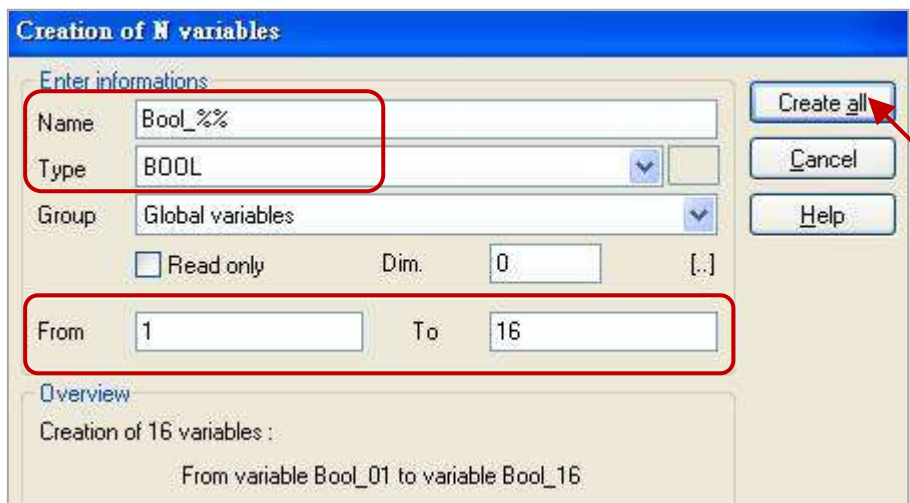
使用小技巧：對於有連續順序的變數，在步驟 2 可改以輸入“LED”再以複製 (Ctrl+C) 並貼上 (Ctrl+V) 的方式自動產生編號 (即，“LED1”、“LED2”)，最後再刪除“LED”即可。

使用小技巧 2:

1. 若需新增多個變數時 (例如: “Boo_01 ~ Boo_16”) , 滑鼠右鍵點選 “Global variables” 再選擇 “Add Multi Variables” 。



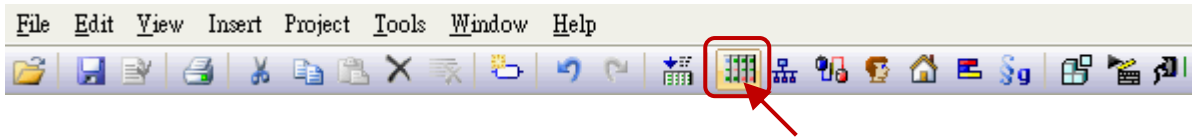
2. 如圖 , 輸入以下設定值 (Name: “Bool_%%” ; Type: “BOOL” ; From: 1 ; To: 16) 來建立名稱為 “Bool_01” ~ “Bool_16” 的布林變數 , 再點選 “Create all” 完成設定 。



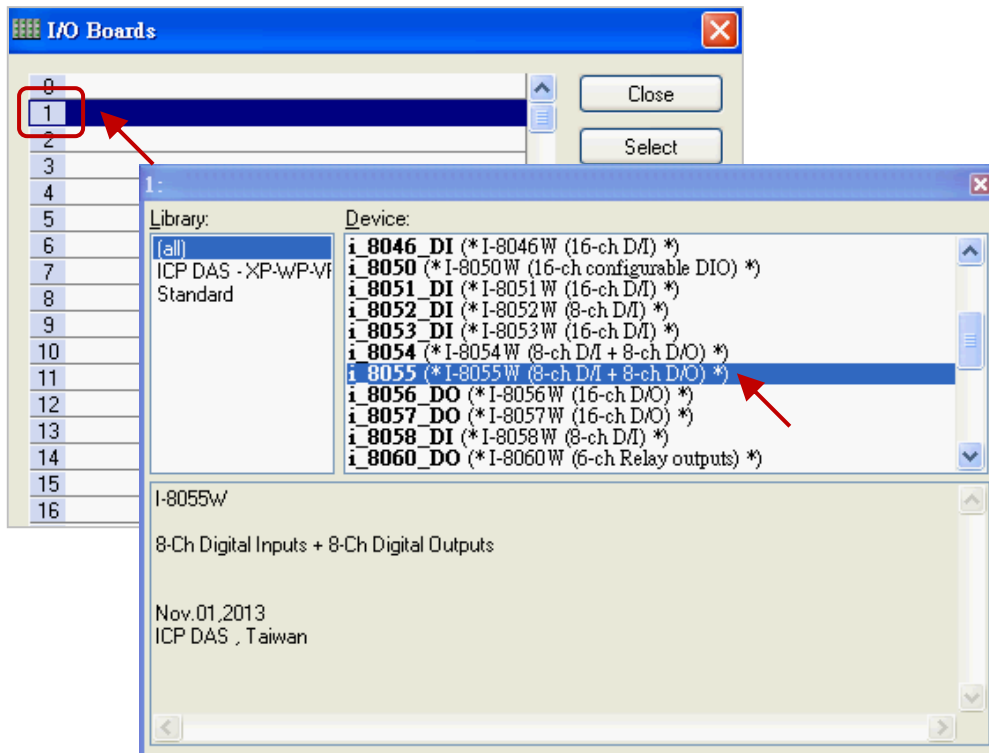
2.3.2 宣告 I/O 變數

此範例中，會在 PAC 的 Slot 1 插上一片 I-8055W 模組來顯示閃爍功能。因此，在“Demo01”專案中，需新增 I/O 連結來對應到 PAC 中的 I/O 模組。

1. 點選工具列的 "Open I/Os" 按鈕來新增 I/O 連結。

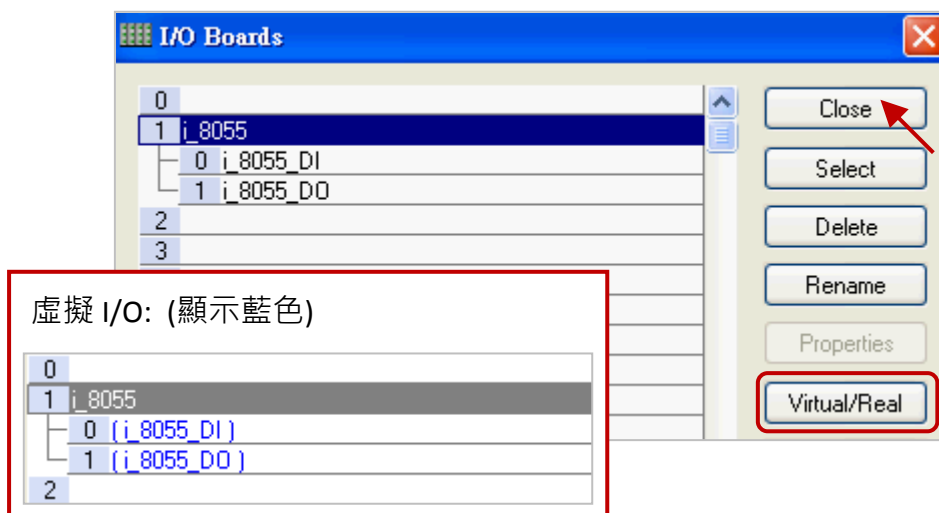


2. 滑鼠雙擊 "Slot 1" 再雙擊 "i_8055" 以完成 I/O 選取。

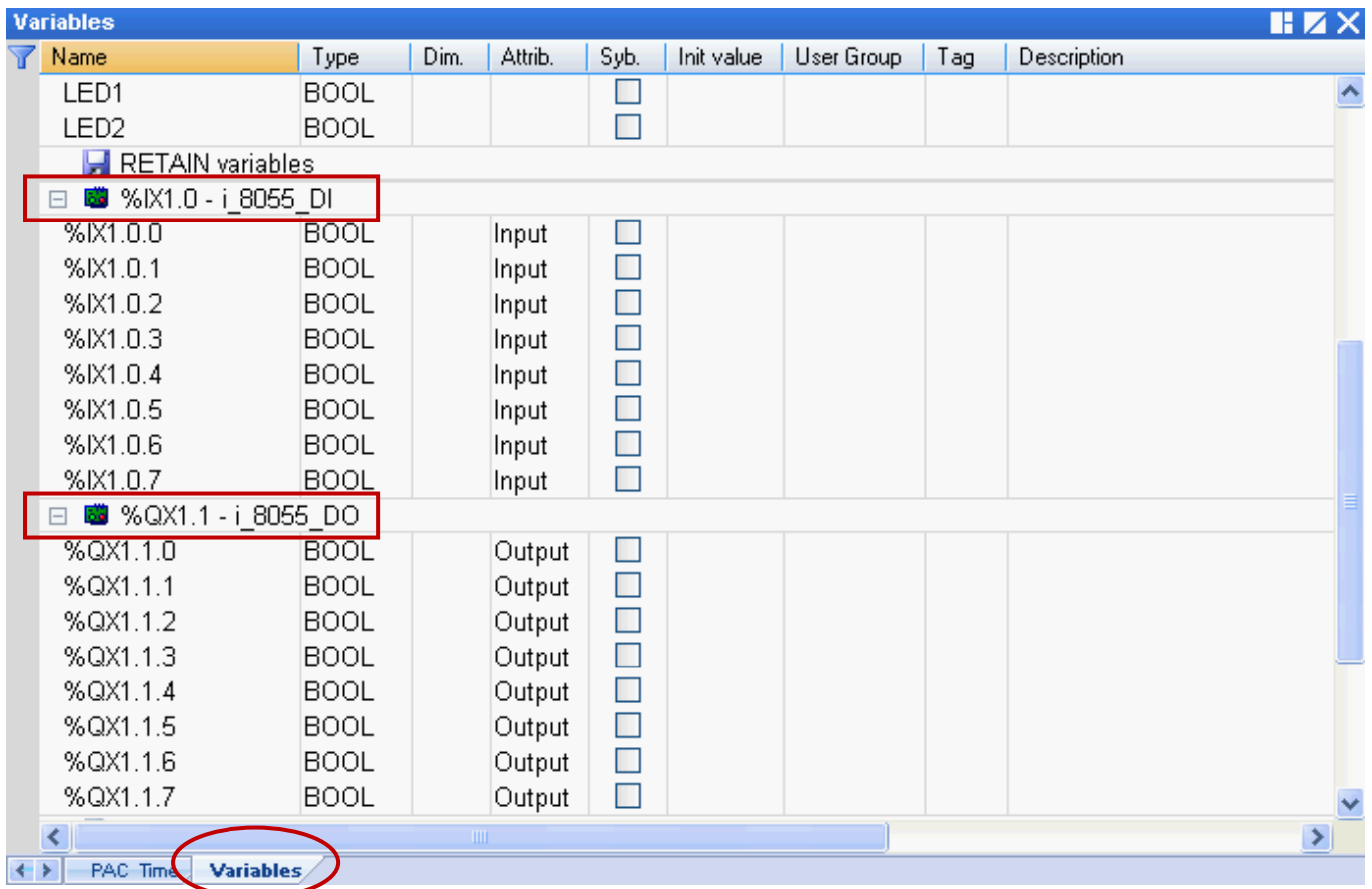


3. 點選“Close”離開“I/O Boards”視窗。

註:可點選“Virtual/Real”來切換為 虛擬 I/O (測試用) 或 真實 I/O，此處使用真實 I/O。



剛剛在 “I/O Boards” 視窗連上 “i_8055” 後，會自動在 “Variables” 視窗中新增 8 個 Input 與 Output 變數供程式使用。



%IX1.0 - i_8055_DI

- I: 表示 “Input”
- X: 表示 “Boolean”
- 1: 表示 “Slot 1”

%QX1.1 - i_8055_DO

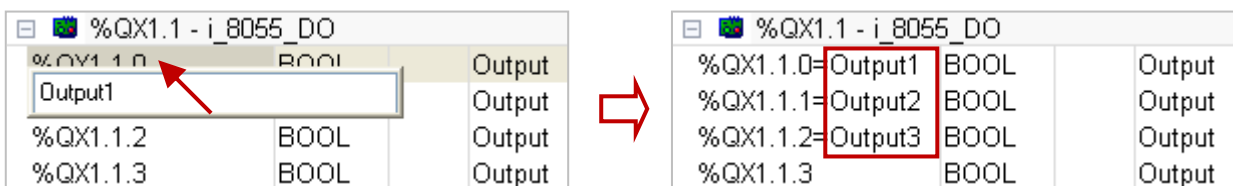
- Q: 表示 “Output”
- X: 表示 “Boolean”
- 1: 表示 “Slot 1”

%ID 或 %QD

- D: 表示 “Integer/Real”

此範例中需使用 3 個 Output 變數，您可修改變數名稱以方便識別。

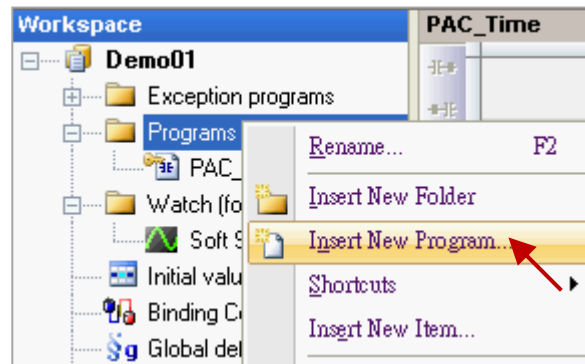
滑鼠雙擊欲使用的項目，輸入名稱後按 “Enter” 鍵完成設定。



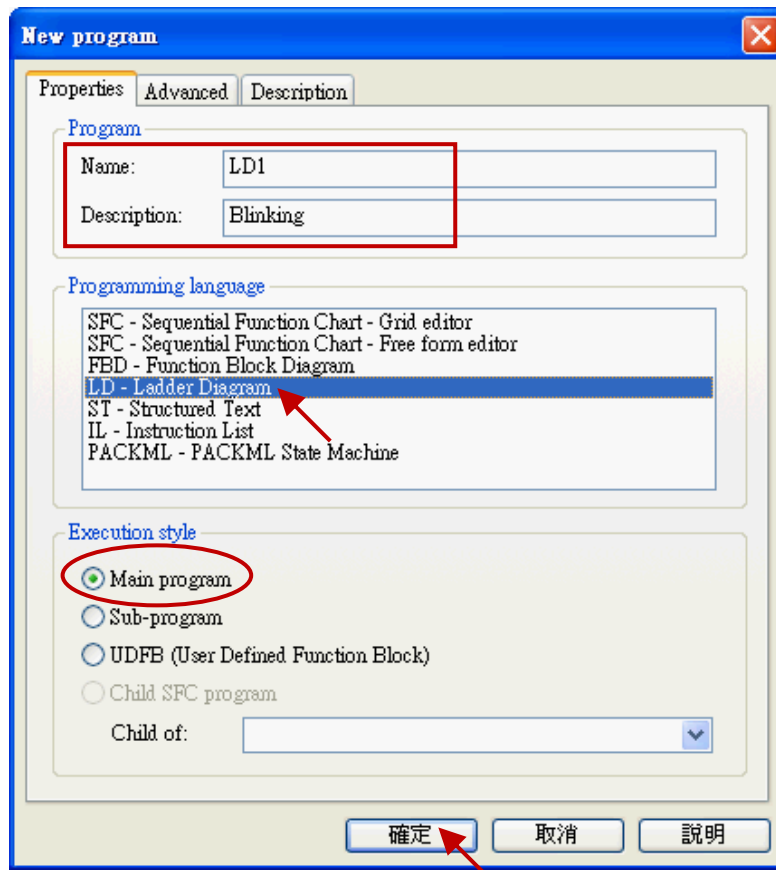
2.3.3 建立 LD 程式

在“Demo01”專案中，我們將新增一個名為“LD1”的程式，用來產生閃爍功能。

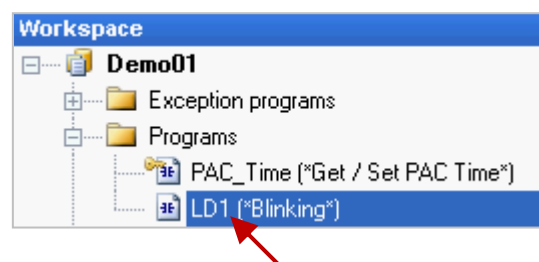
1. 工作區中，滑鼠右鍵點選“Programs”再選擇“Insert New Program...”。



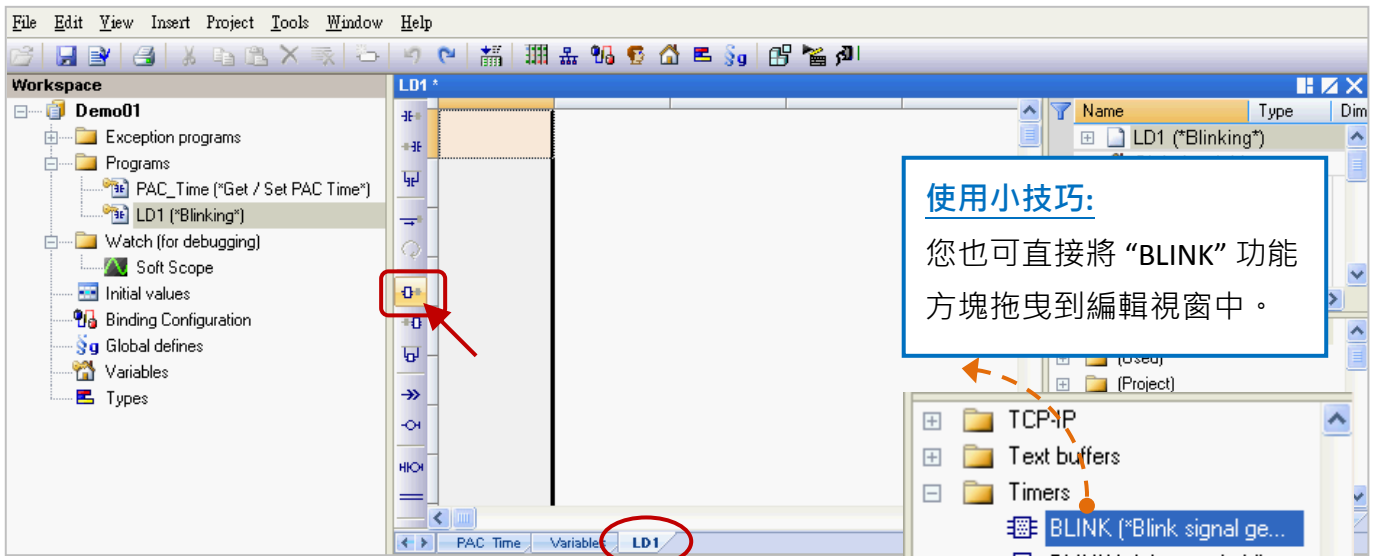
2. 於“Name”欄位中填入程式名稱，並在“Description”欄位加入簡單的描述。接著，選取“LD – Ladder Diagram”編程語言，再點選“確定”。



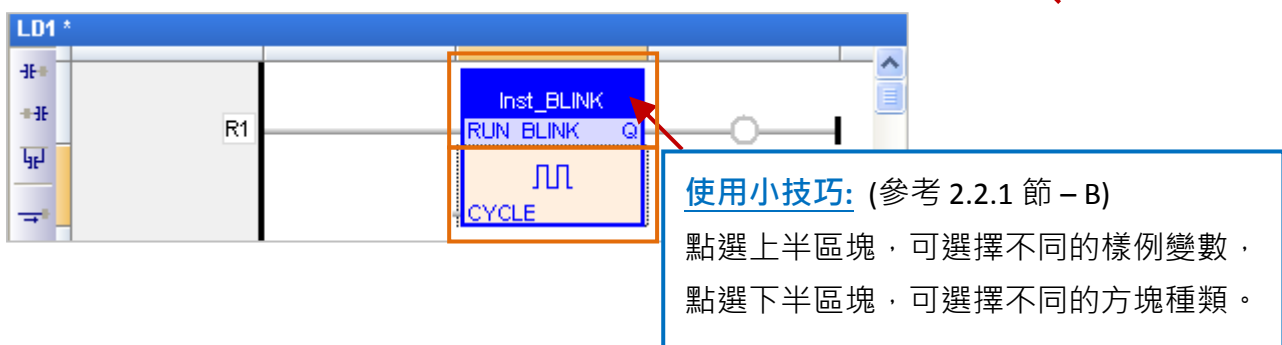
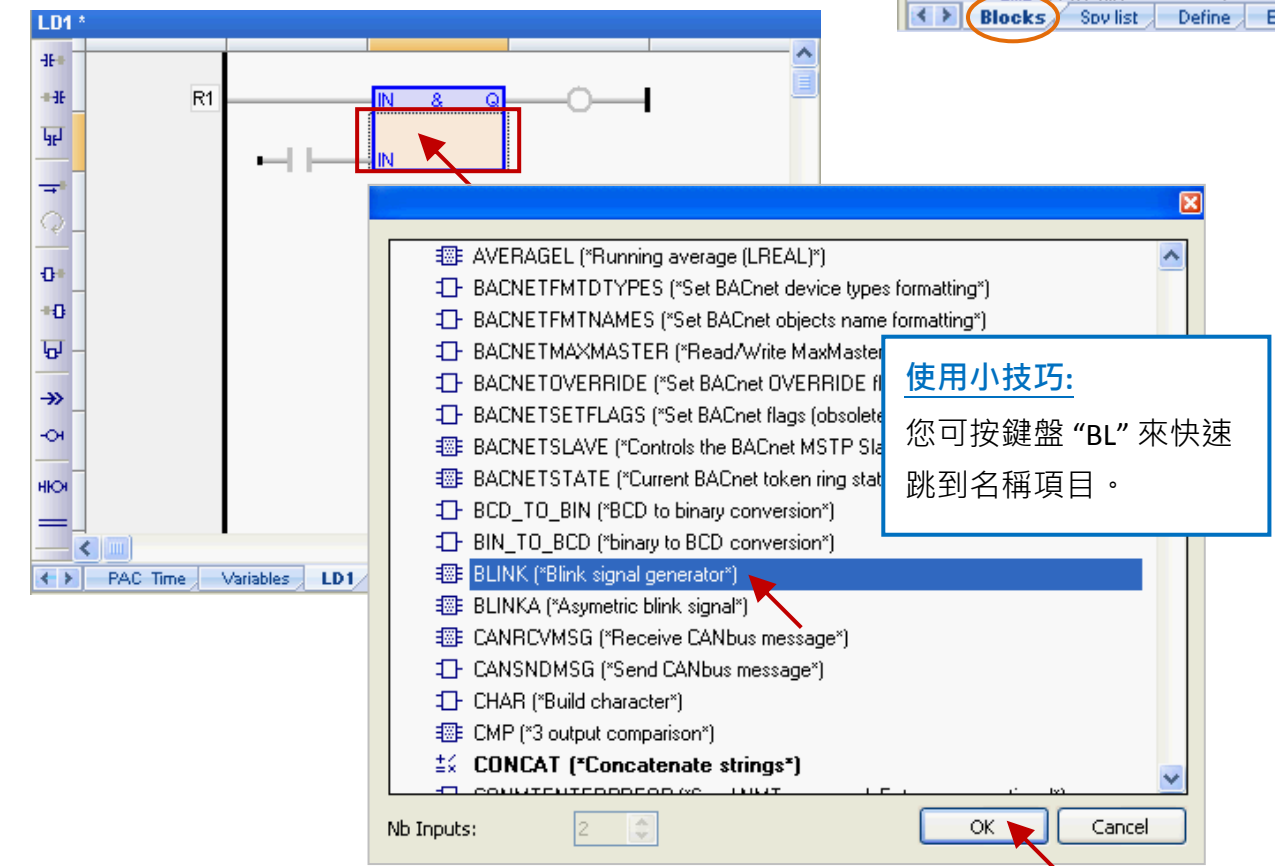
3. 滑鼠雙擊“LD1”程式來開啟編輯視窗。



4. 點選“LD1”視窗左側的“Insert FB..”按鈕來加入一個功能方塊 (Function Block)。



5. 滑鼠雙擊此功能方塊，並選取“BLINK”再按“OK”。

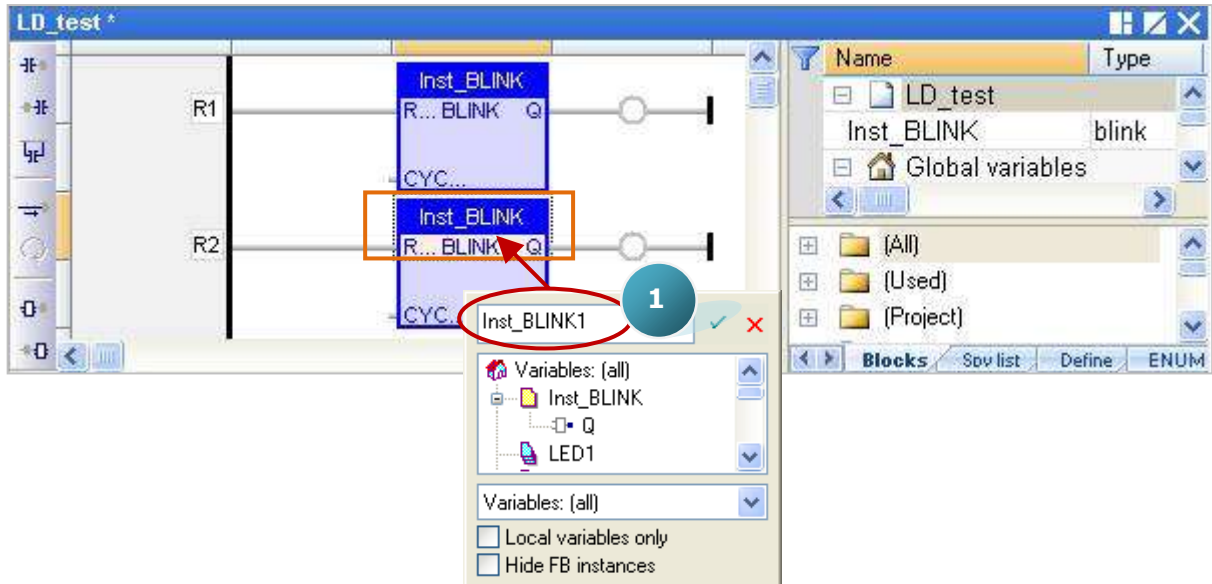




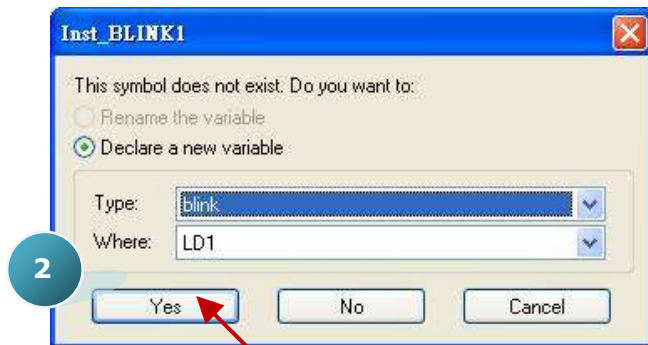
重要注意事項:

使用者在編輯程式時，可能會以複製並貼上的方式來新增功能方塊，但此方式會造成功能方塊的樣例變數名稱重複，而導致功能異常。因此，請務必再建立一個新的樣例變數名稱。

1. 滑鼠雙擊該功能方塊，並輸入新的名稱 (例如: "Inst_BLINK1")，再點選 完成設定。



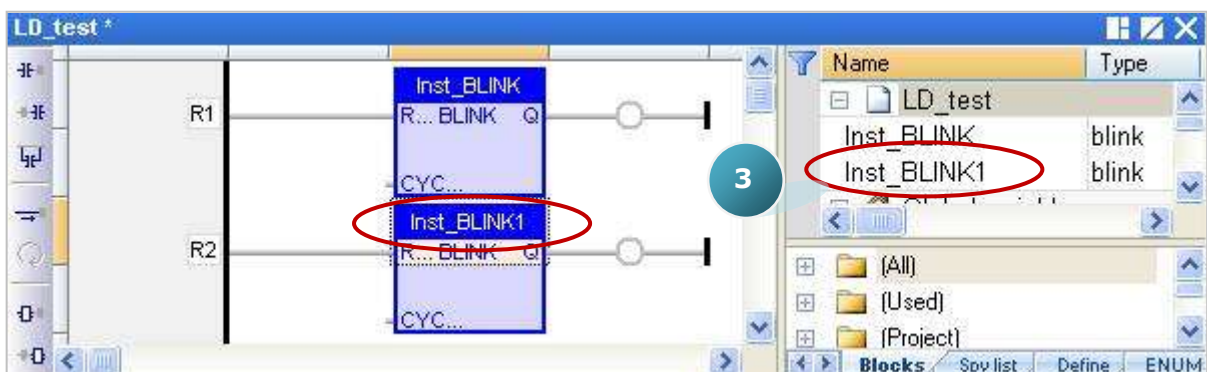
2. 於 "Inst_BLINK1" 變數視窗中，點選 "Yes" 來建立此樣例變數。



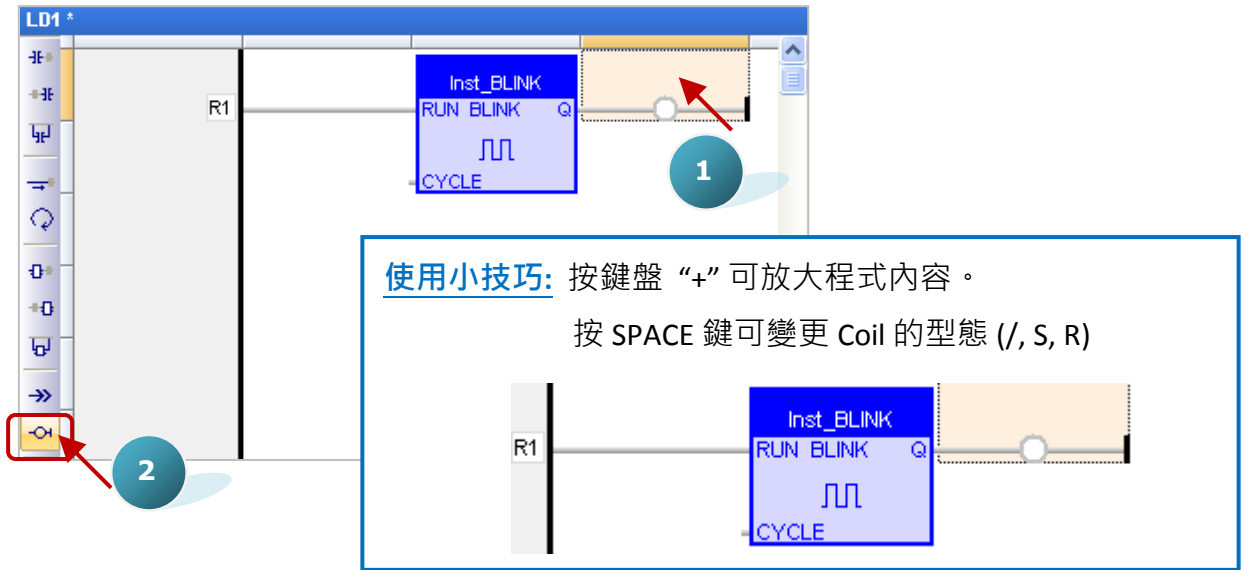
使用小技巧:

您也可依照相同的方式，滑鼠雙擊功能方塊旁的 "Coil" 來新建/指定一個變數 (參考下方步驟 7)。

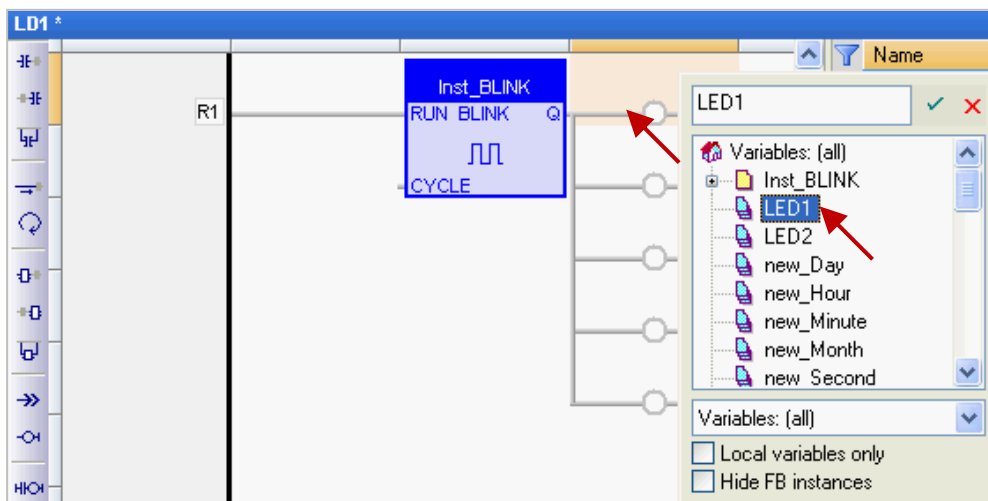
3. 變數區已建立了一個名為 "Inst_BLINK1" 的樣例變數。



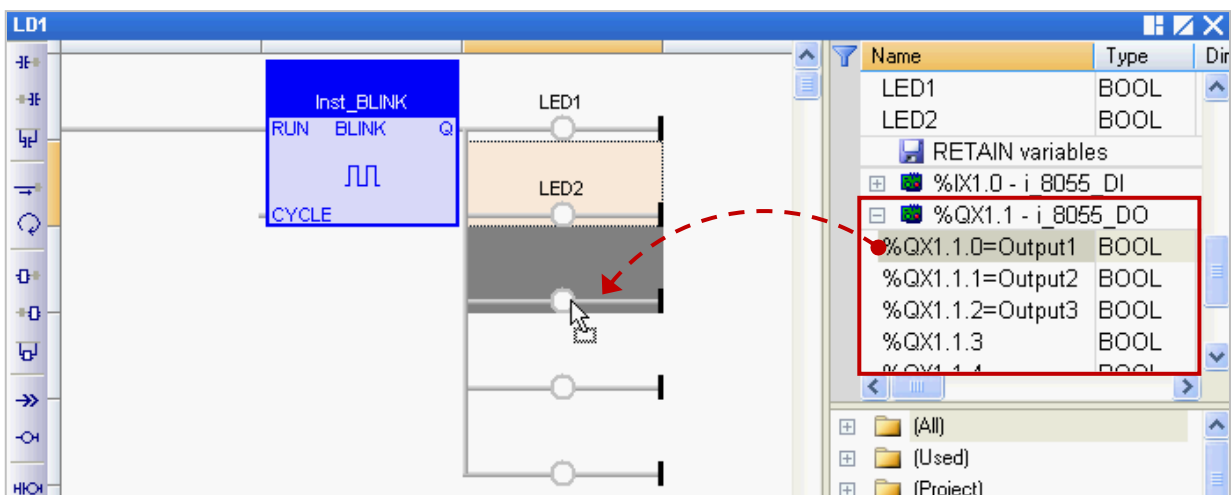
6. 點選“BLINK”功能方塊右方的“Coil”，再連續點選“Insert Coil”按鈕來新增4個“Coil”。




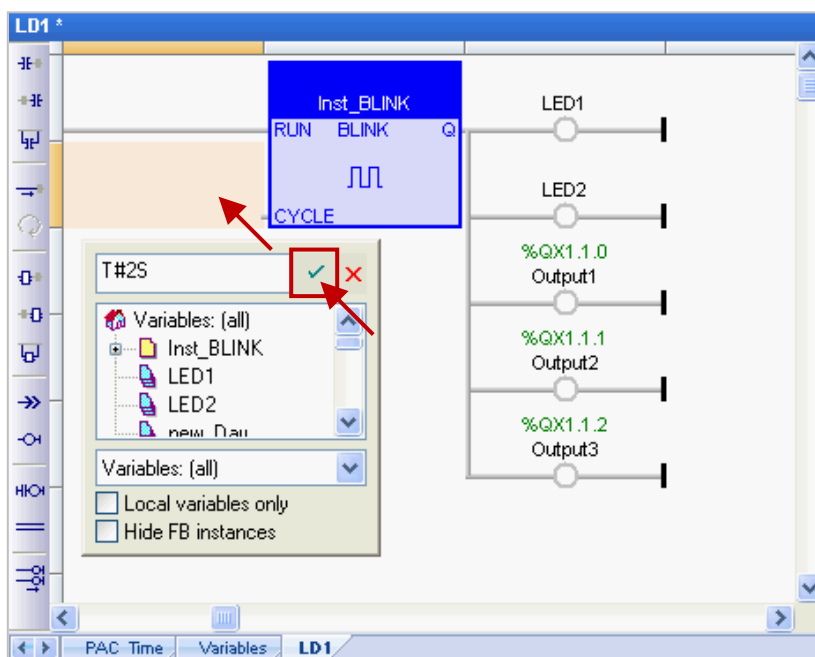
7. 滑鼠雙擊第一個“Coil”再雙擊“LED1”來指定變數。依照同樣的方式，將第二個“Coil”指定為“LED2”變數。



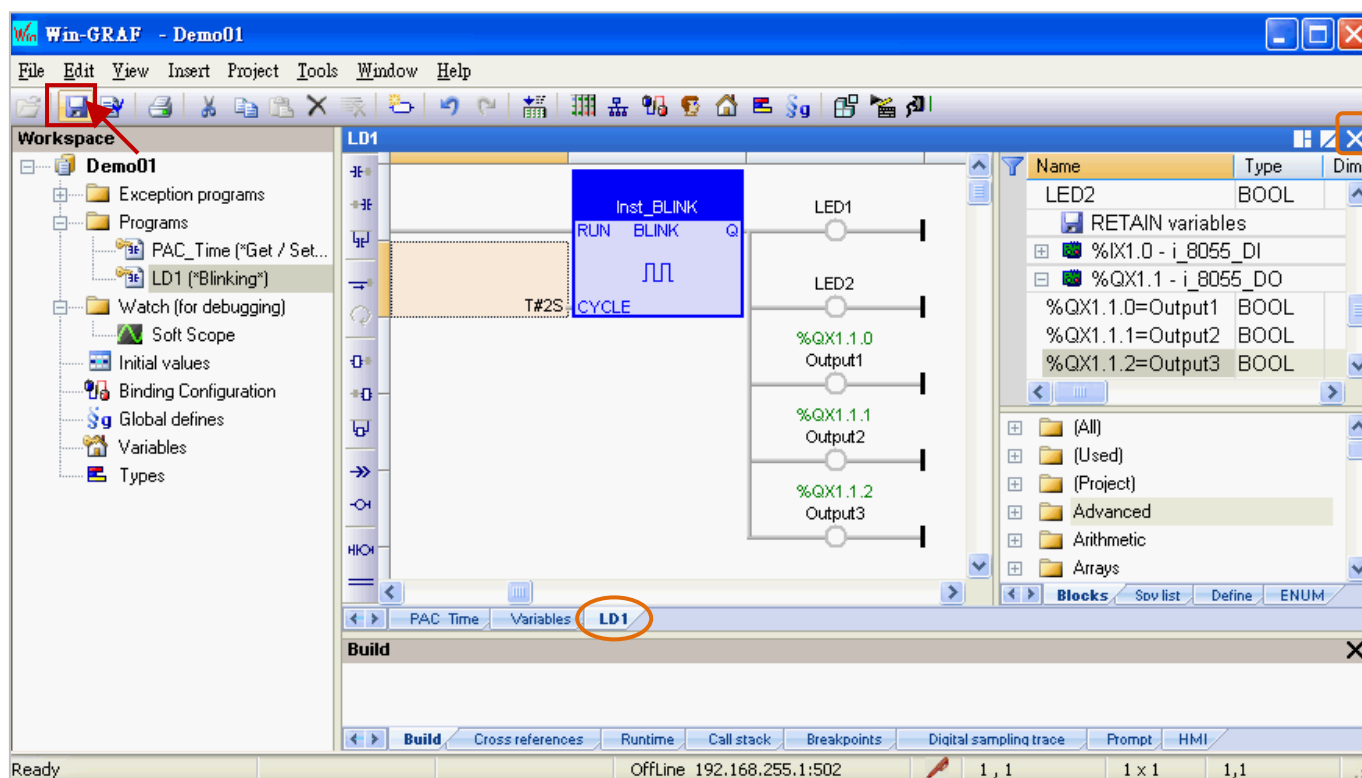
8. 接著，可用滑鼠拖曳方式來指定“Output1”、“Output2”、“Output3”變數。





9. 滑鼠雙擊“CYCLE”的左側，並輸入“T#2S”（表示每2秒閃爍一次），再點選  完成設定。



10. 最後，點選“Save”按鈕，儲存“LD1”程式。

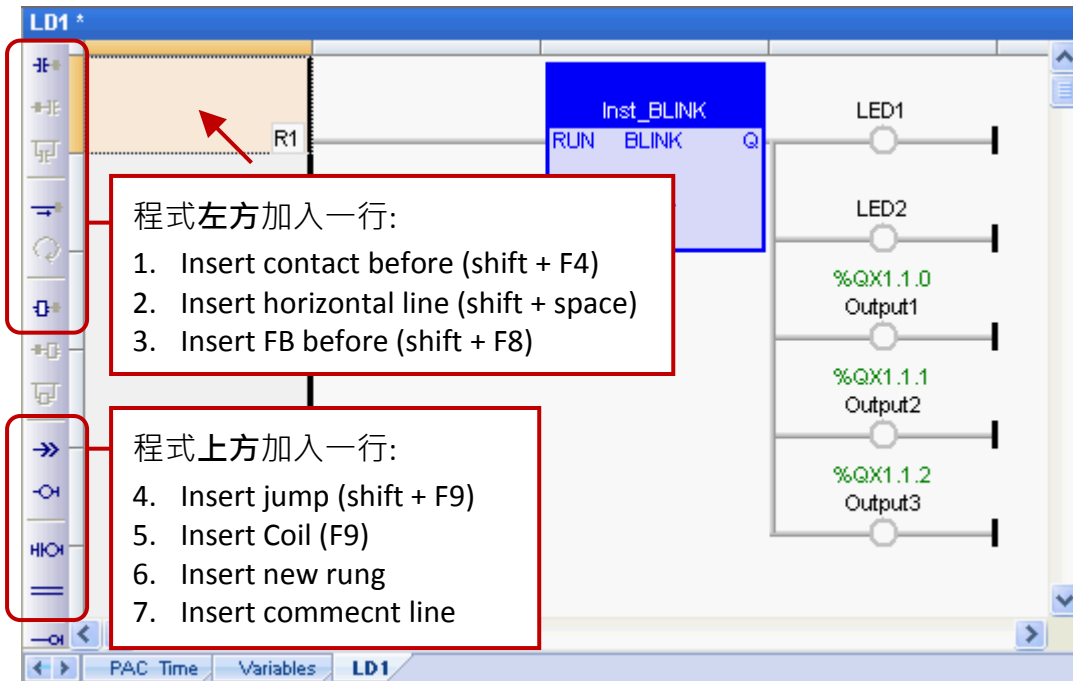


註:  表示該程式目前為開啟狀態（鎖定，不可刪除），可點選程式區右上角的“X”來關閉程式視窗（解除鎖定，）。

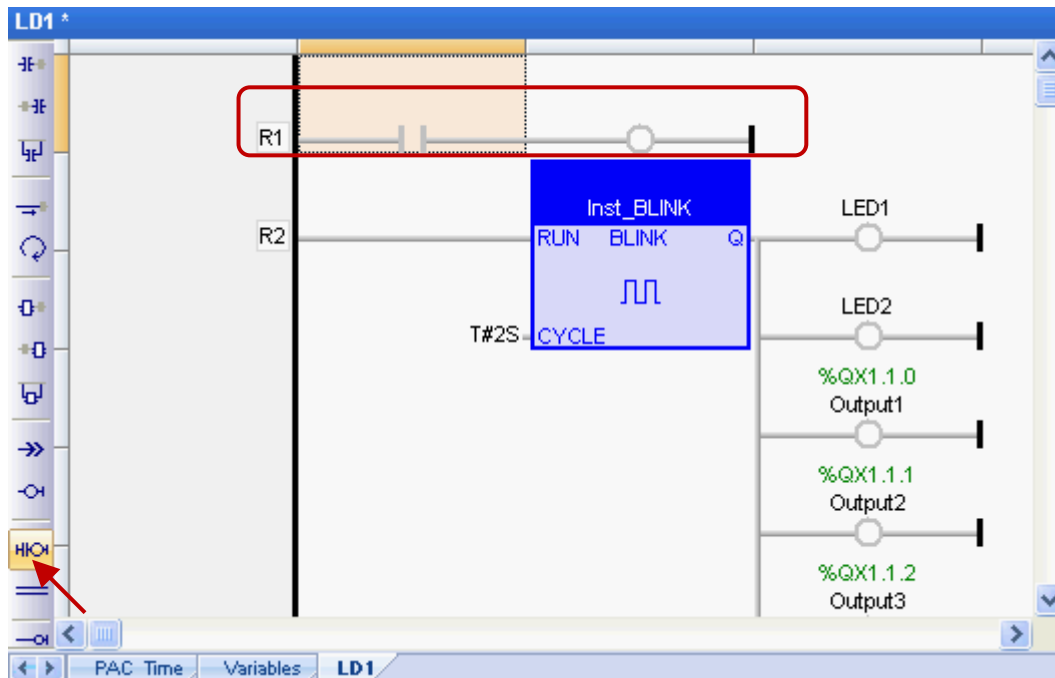
程式編輯完成後，若您想在程式的最上方新增一行程式：

1. 滑鼠點選程式編輯區的左上角，此時元件區會顯示可點選的項目，點選元件按鈕 (如圖 · 4 ~ 7) 可在第一行加入程式。

註：若點選元件按鈕 (如圖 · 1 ~ 3) 則會在程式的左方加入一行程式。



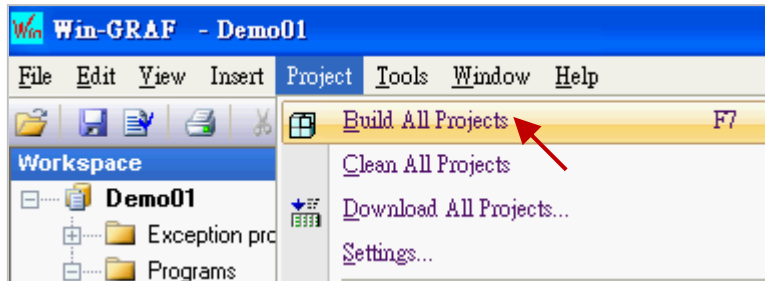
2. 此例 · 點選 “Insert new rung” 可在第一行加入程式。



2.3.4 編譯程式

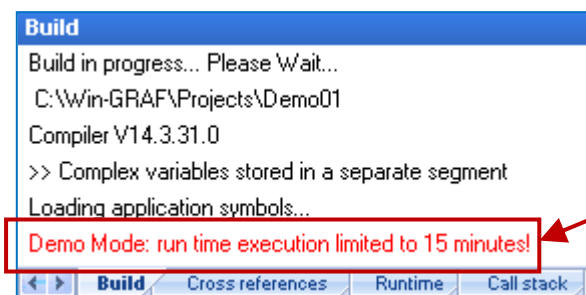
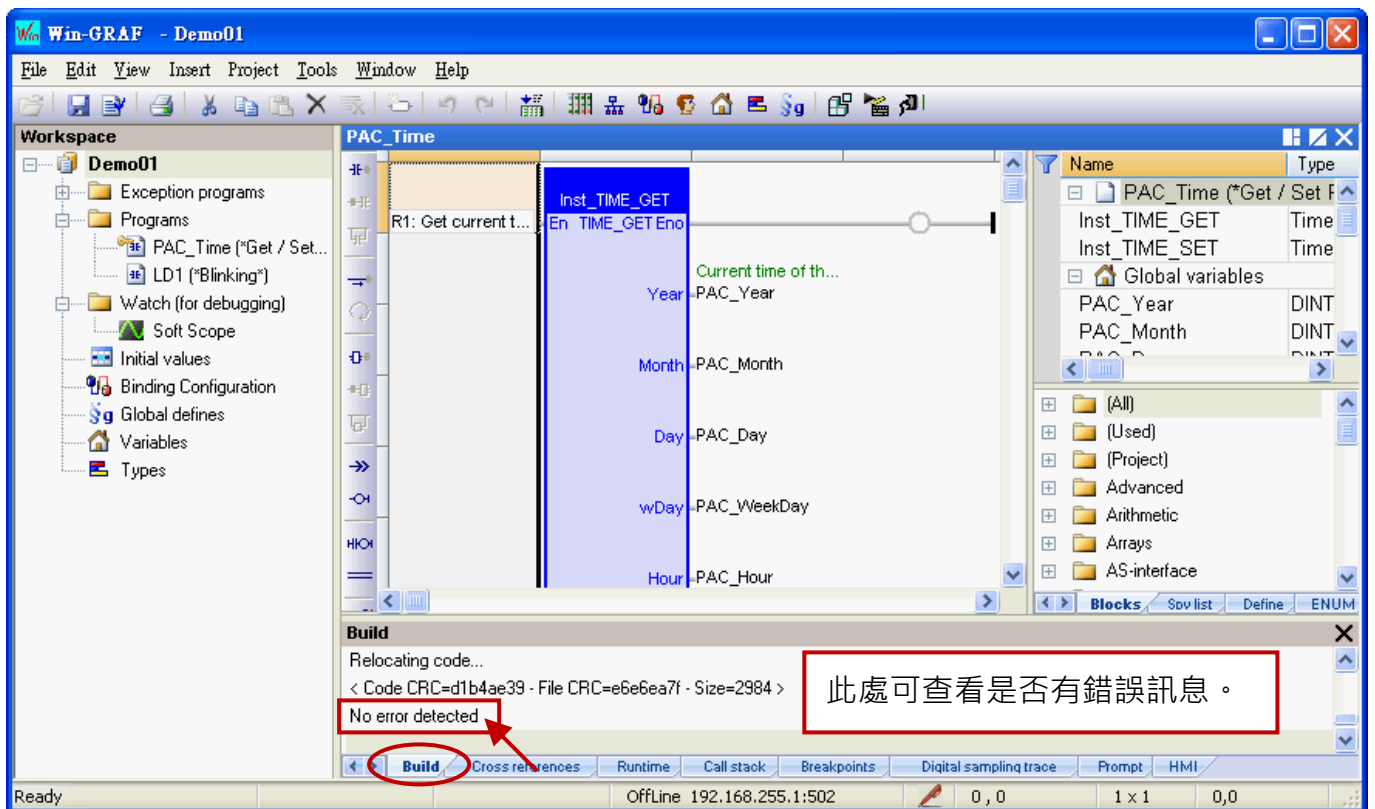
在上一節中，我們已新增並儲存了 LD 程式，為了讓 Win-GRAF 程式可在 PAC 上正常運作，接下來要進行程式編譯。

1. 點選功能表“Project”再選擇“Build All Projects”開始編譯程式。



2. 若訊息區中出現“**No error detected**”表示編譯成功。

注意：若編譯完成後，又再修改並儲存程式，請先點選(上圖)“Clean All Projects”清除先前的編譯結果並再執行編譯一次。

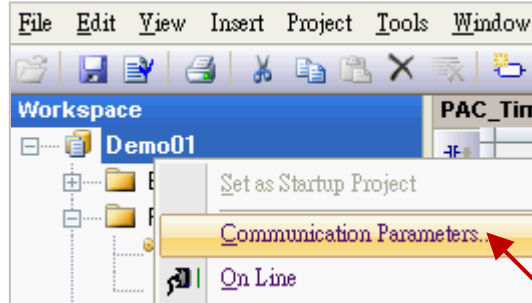


若使用 Demo Mode，"Build" 訊息區會提示 PAC 中的 Win-GRAF 專案只能運行 15 分鐘。


2.3.5 下載程式到 PAC

下載程式之前，請先設定好通訊參數 (僅支援 Ethernet 通訊埠的方式)。若使用者想使用序列埠 (例如: RS-232) 去連 PAC 的序列埠，請參考 [附錄 E](#)。

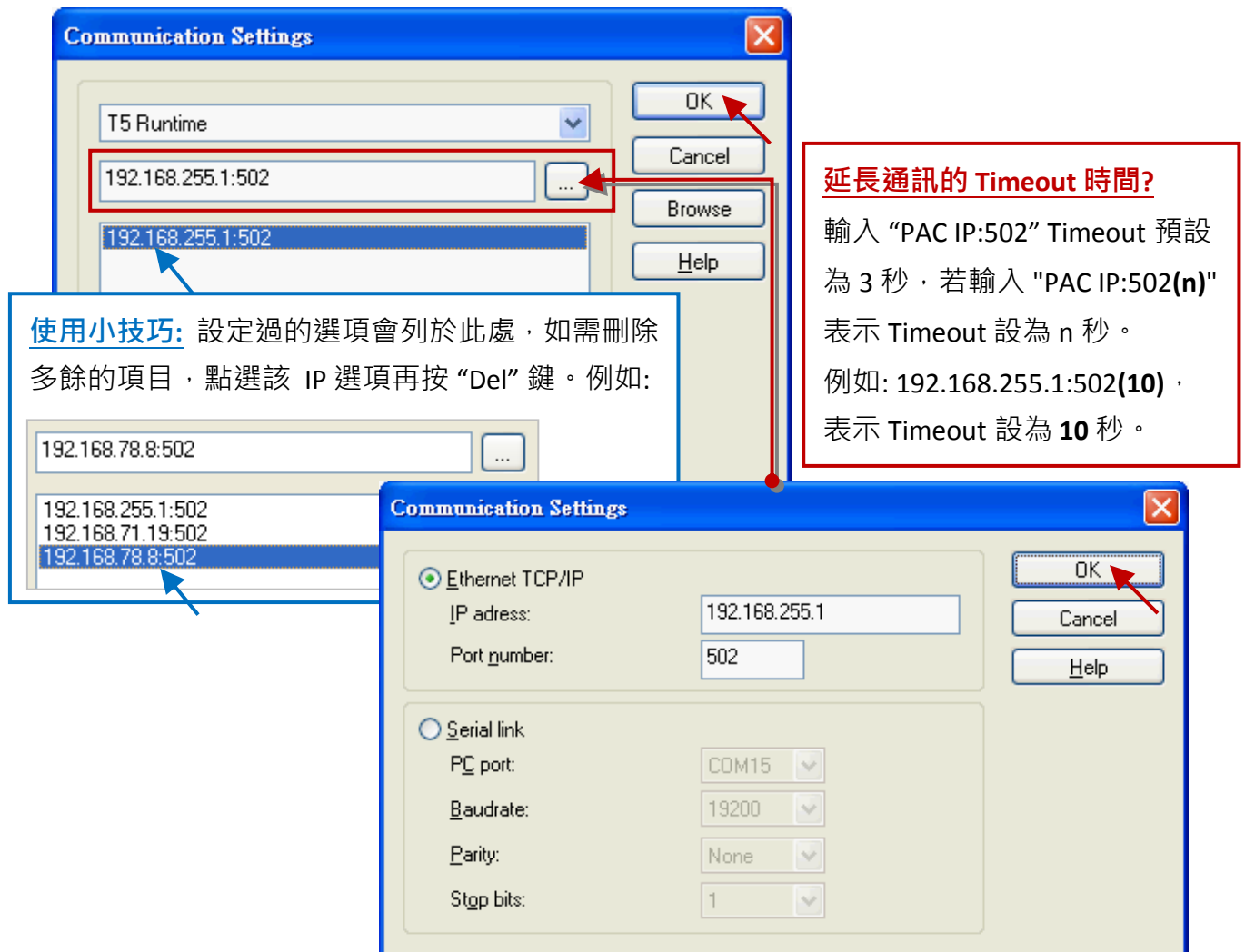
1. 滑鼠右鍵點選專案名稱 (即，Demo01)，再選擇 “Communication Parameters...” 開啟設定視窗。



2. 填入 “PAC IP:502” (例如: “192.168.255.1:502”) 來新增一個 IP 選項，並按 “OK”。

也可點選  按鈕，來新增 或 修改 IP 位址。


(註: PAC IP 出廠預設為 192.168.255.1，Port 號固定為 502)

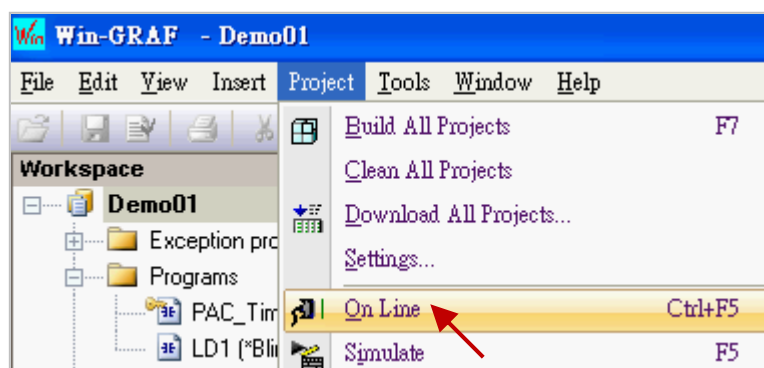


延長通訊的 Timeout 時間?
輸入 “PAC IP:502” Timeout 預設為 3 秒，若輸入 “PAC IP:502(n)” 表示 Timeout 設為 n 秒。
例如: 192.168.255.1:502(10)，表示 Timeout 設為 10 秒。

使用小技巧: 設定過的選項會列於此處，如需刪除多餘的項目，點選該 IP 選項再按 “Del” 鍵。例如:

3. 建立連線前，請確認您的 Win-GRAF PAC 已經開機且網路通訊正常。

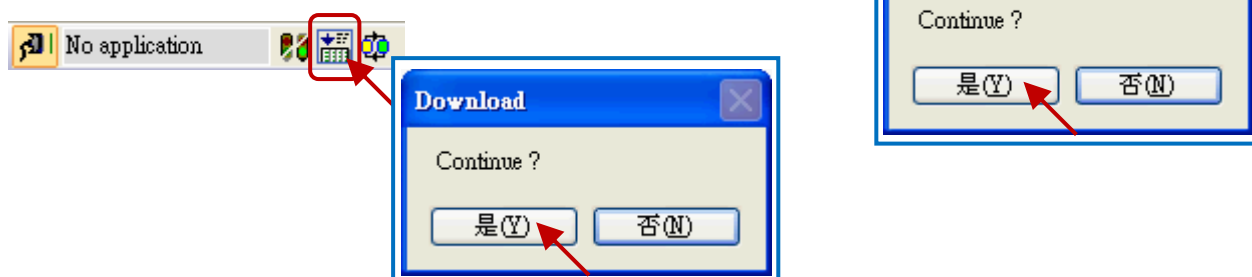
4. 點選功能表“Project”再選擇“On Line”，或點選工具按鈕  來與 PAC 建立連線。



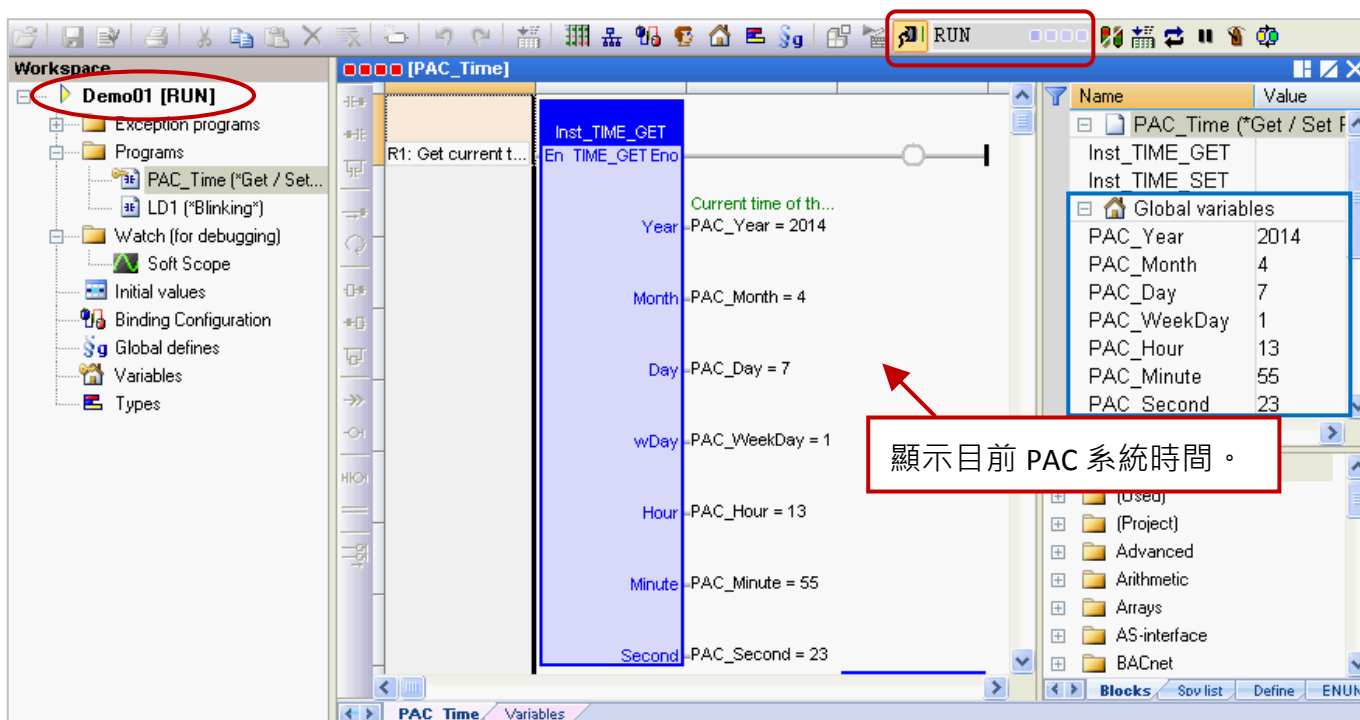
5. 如圖，若出現“App: TEST”字樣 (與目前專案名稱“Demo01”不同)，表示 PAC 內已有一個正在運行的專案 (名稱: “TEST”)，請點選工具按鈕“Stop application”停止該專案運行。



6. 接著，點選工具按鈕“Download”來下載“Demo01”專案。



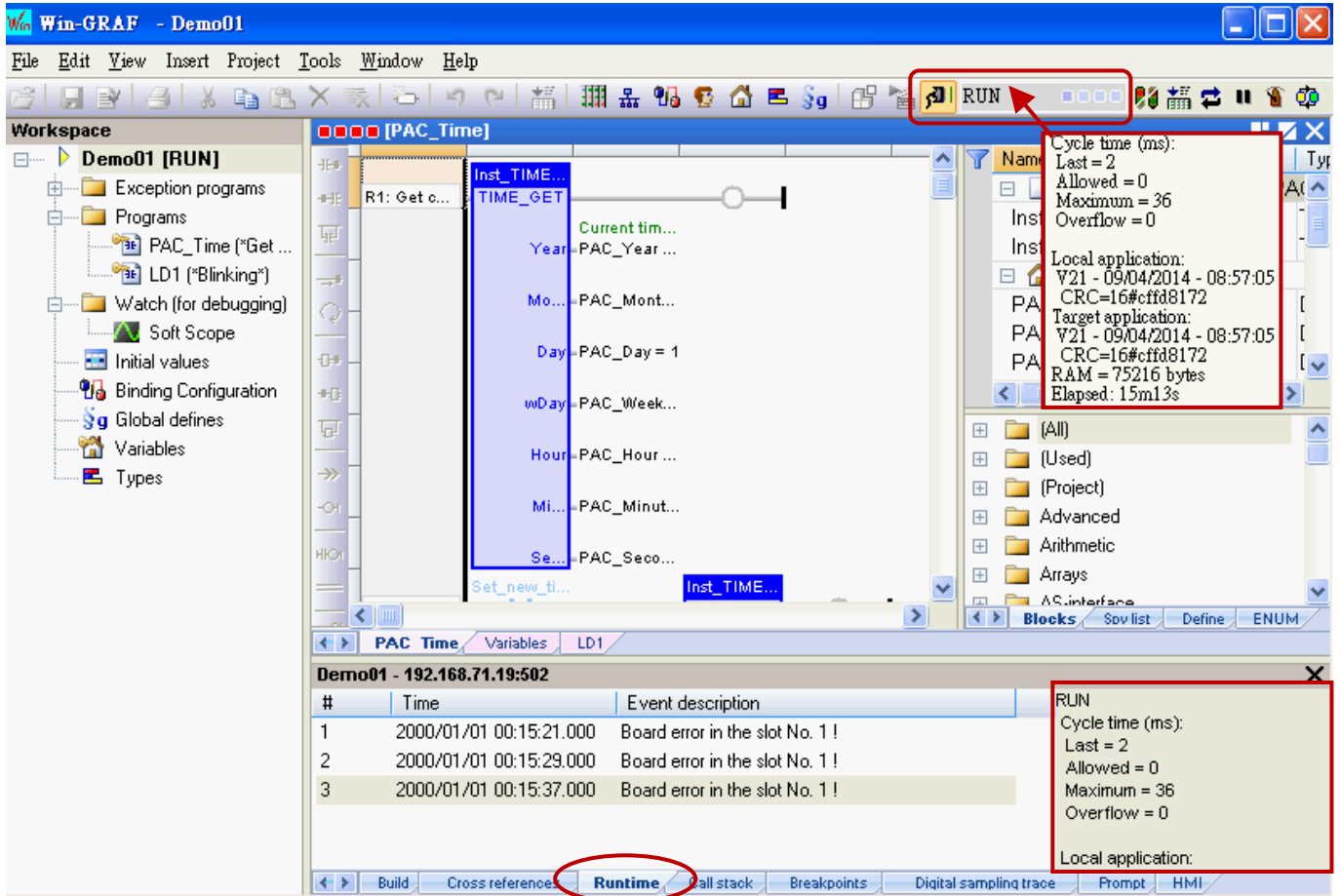
7. 若出現“RUN”字樣，表示連線成功且“Demo01”專案運行中。




註: 若下載過程中有出現錯誤訊息，請參考 [附錄 B](#) 來排除問題。

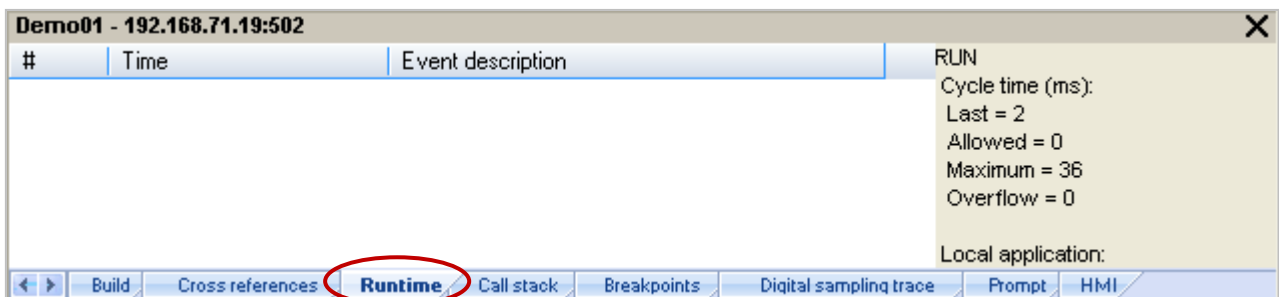
Cycle time

在連線的狀態下，滑鼠移到工具列“RUN”的位置可查看目前 PAC 內程式的 Cycle time。您也可在右下方的訊息區中查看此資訊。



執行連線時 ()，會自動切換到“Runtime”訊息區，您可在此查看下載到 PAC 中的程式是否出現錯誤訊息。(例如：“Board error in the slot No. 1 !”表示 PAC 的 Slot 1 上無插上 I/O 模組或是有異常，此例需在 PAC 的 Slot 1 插上 I-8055W 模組。)

請將 PAC 關機，插上 I-8055W 模組後再重新開機，並再執行連線時 () 一次。

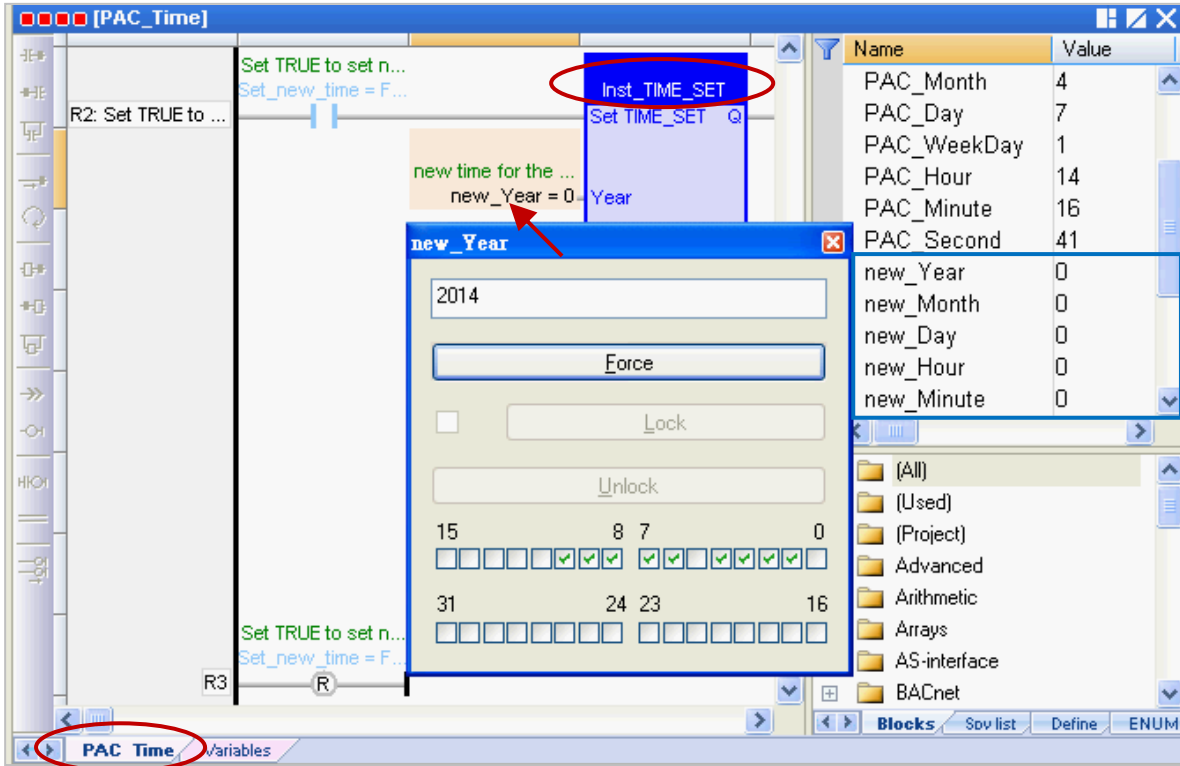


2.3.6 測試程式

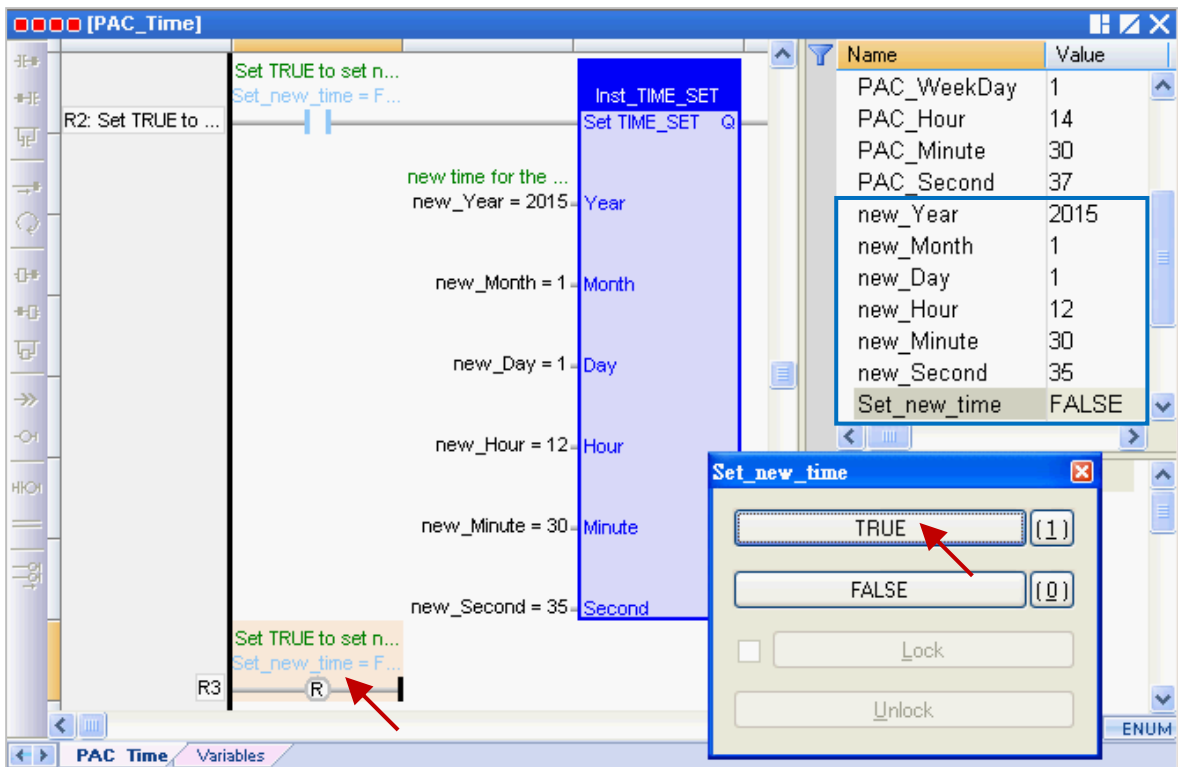
上一章節中，已成功下載“Demo01”專案，接下來要介紹如何測試程式。

“PAC_Time”程式：

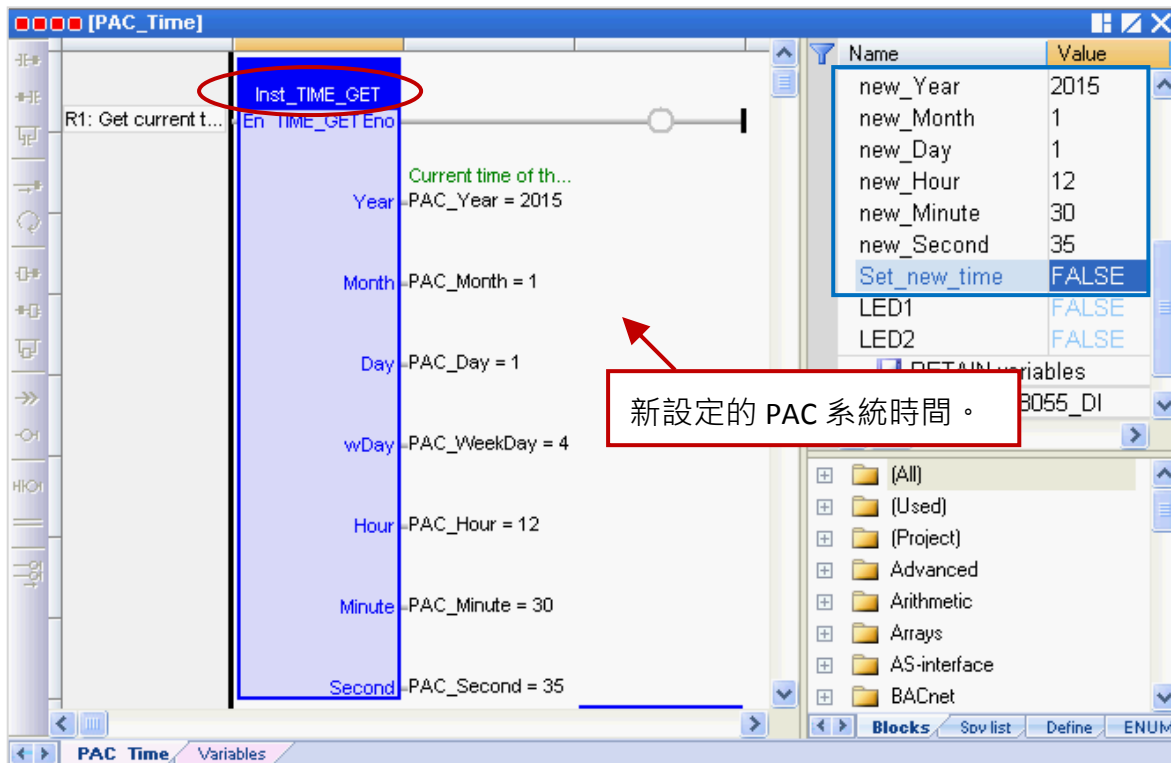
1. 在“TIME_SET”功能方塊中(或變數區中)，滑鼠雙擊各變數名稱(例如：“new_Year”)來依序修改系統時間。(假設：將資料改為2015年1月1日，12點30分35秒。)



2. 將“Set_new_time”變數設定為“TRUE”以寫入新的系統時間。

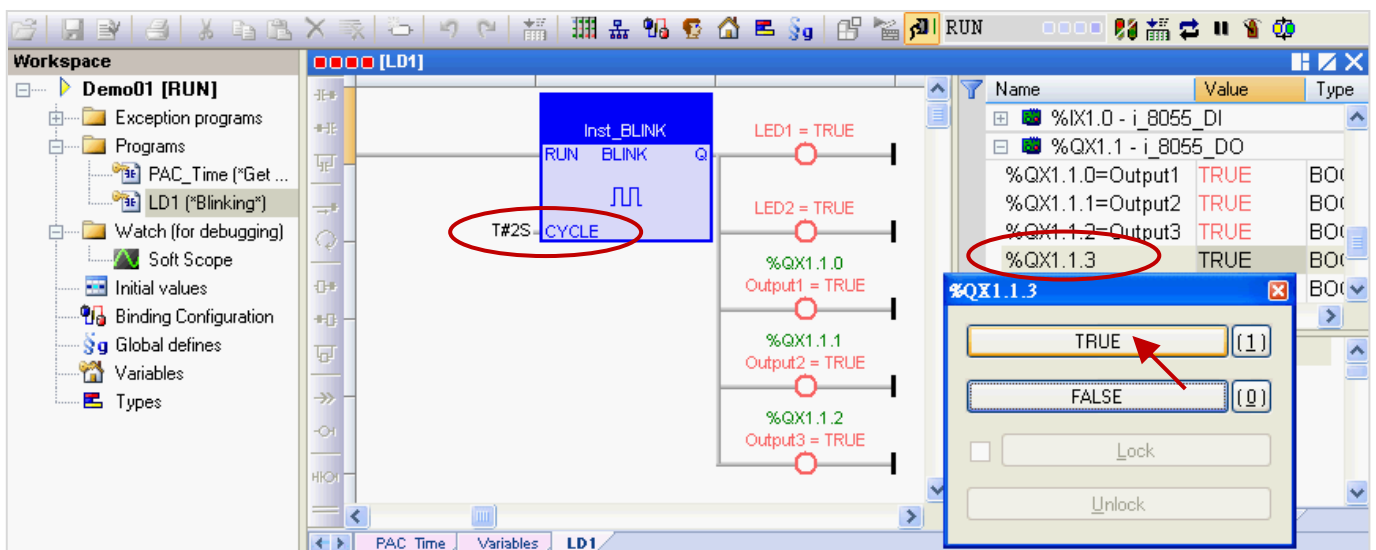



3. 在“TIME_GET”功能方塊中(或變數區中)，顯示了新設定的 PAC 系統時間，且“Set_new_time”變數會自動重置為“FALSE”。



“LD1” 程式:

4. 當“Demo01”專案運行時，可查看 PAC 上 (Slot 1) 的 I-8055W I/O 模組 - DO0、DO1、DO2 是否如同設定值(即，“T#2S”表示每 2 秒閃爍一次)。您也可改為在“CYCLE”左側指定一個資料型態為“TIME”的變數，設定方式參考 2.3.1 節，以便修改時間設定。)
5. 若將變數區的“%QX1.1.3”設定為“TURE”，可見到 I-8055W I/O 模組 – DO3 的燈號為亮。



6. 您可再次點選工具按鈕  來取消與 PAC 的連線。

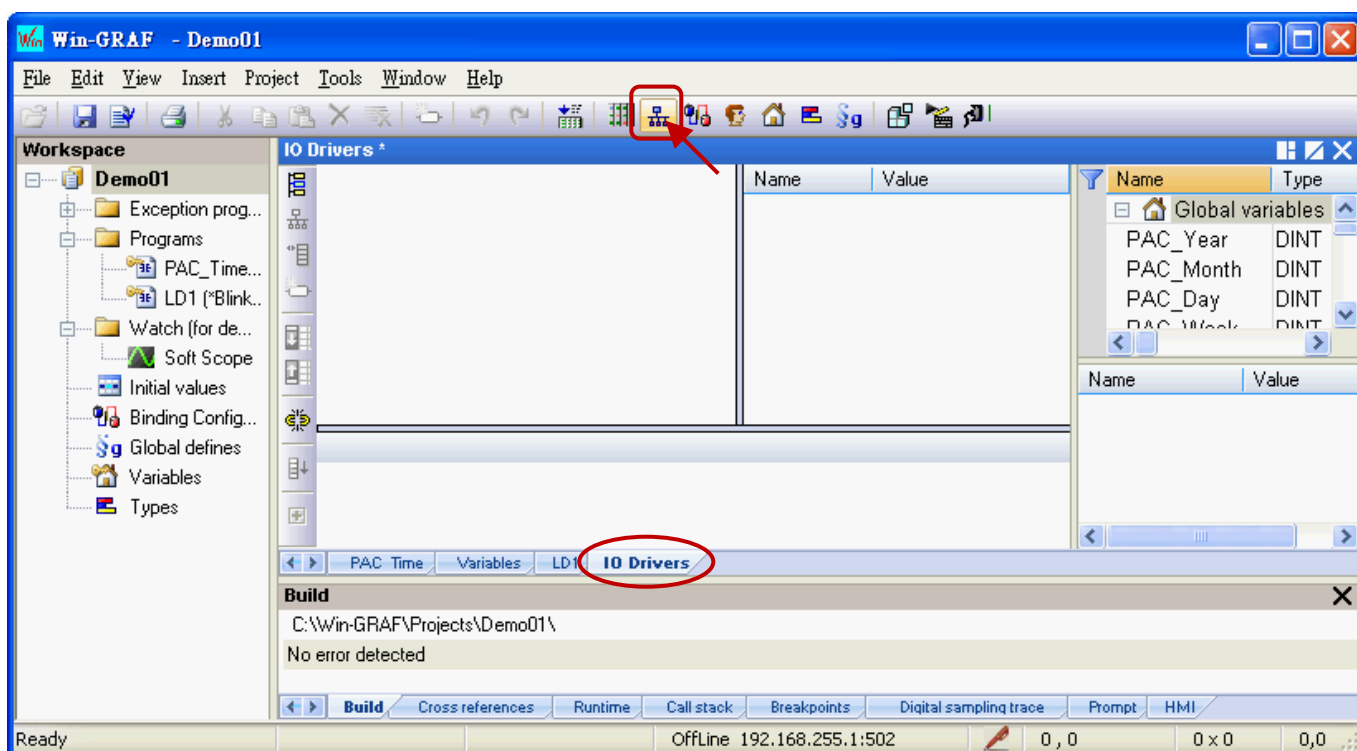
注意: 請勿點選“Stop Application”來停止連線，這會停止 PAC 中運行的程式。

第 3 章 Modbus Slave: 開放 Win-GRAF PAC 與 圖控/HMI 軟體來相互溝通

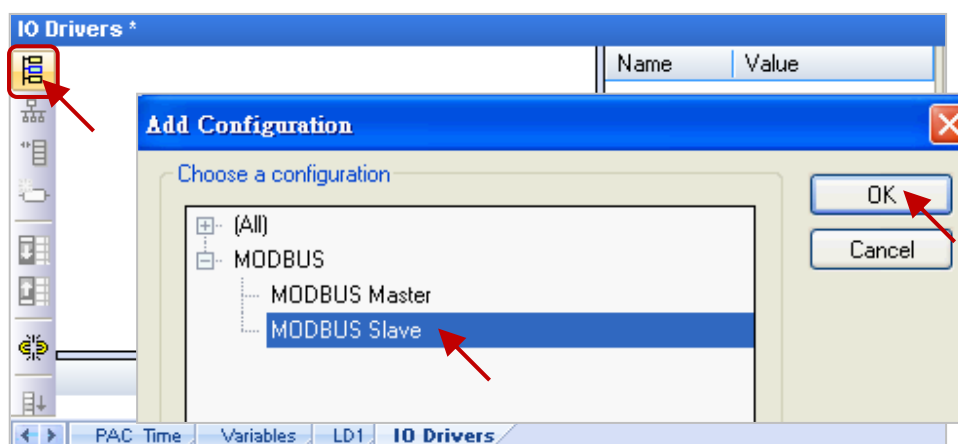
在第 2 章的“Demo01”專案中，說明了如何讀/寫 PAC 中的系統時間(即“PAC_Time”程式)與一個閃爍功能(即“LD1”程式)，此章節將提供“Demo01”專案中的變數，讓圖控軟體(例如:泓格科技的“InduSoft”)或 HMI 軟體來進行存取。Win-GRAF 軟體提供了兩種方式，可開放 PAC 中的資料，一種是啟用 Win-GRAF PAC 為 Modbus TCP Slave，另一種是啟用 Win-GRAF PAC 為 Modbus RTU Slave (見 3.2 節，需先完成 3.1 節內的 Modbus Slave 設定)，請依照以下內容來進行設定。

3.1 啟用 Win-GRAF PAC 為 Modbus TCP Slave

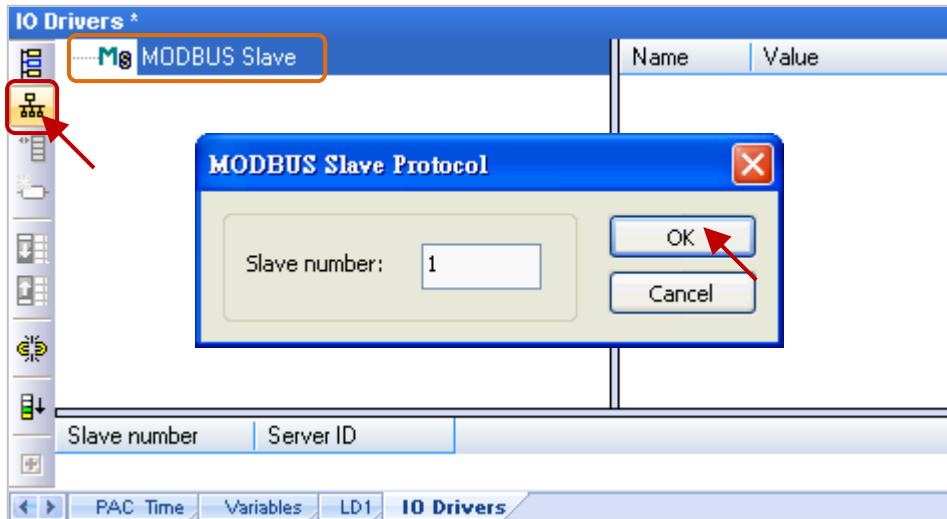
1. 滑鼠點選工具列上的“Open Fieldbus Configuration”按鈕來開啟“I/O Drivers”視窗。



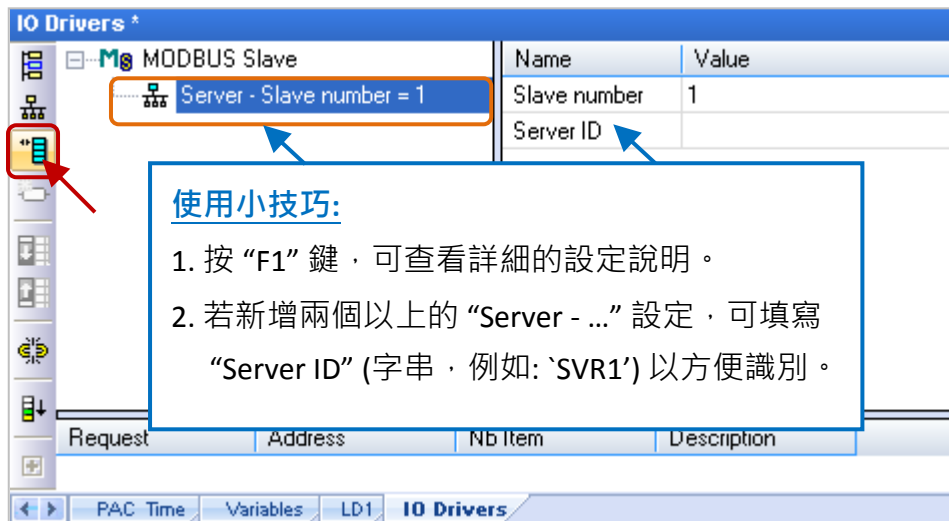
2. 點選“I/O Drivers”視窗左側的“Insert Configuration”按鈕，再點選“MODBUS Slave”並點選“OK”來啟用一個 Modbus TCP Slave。



3. 點選左側的 “Insert Master/Port” 按鈕，並設定 “Slave number” (此例為 “1”)，再點選 “OK”。



4. 點選左側的 “Insert Slave/Data Block” 按鈕，來開啟 “MODBUS Slave Request” 設定視窗。



5. 在 “Description” 填入識別的資訊並點選 “Input Registers” 選項。

可供 Modbus Master **讀取** 資料:

選項	資料型態
Input Bits	BOOL
Input Registers	BYTE, INT, DINT, REAL, ... 等。

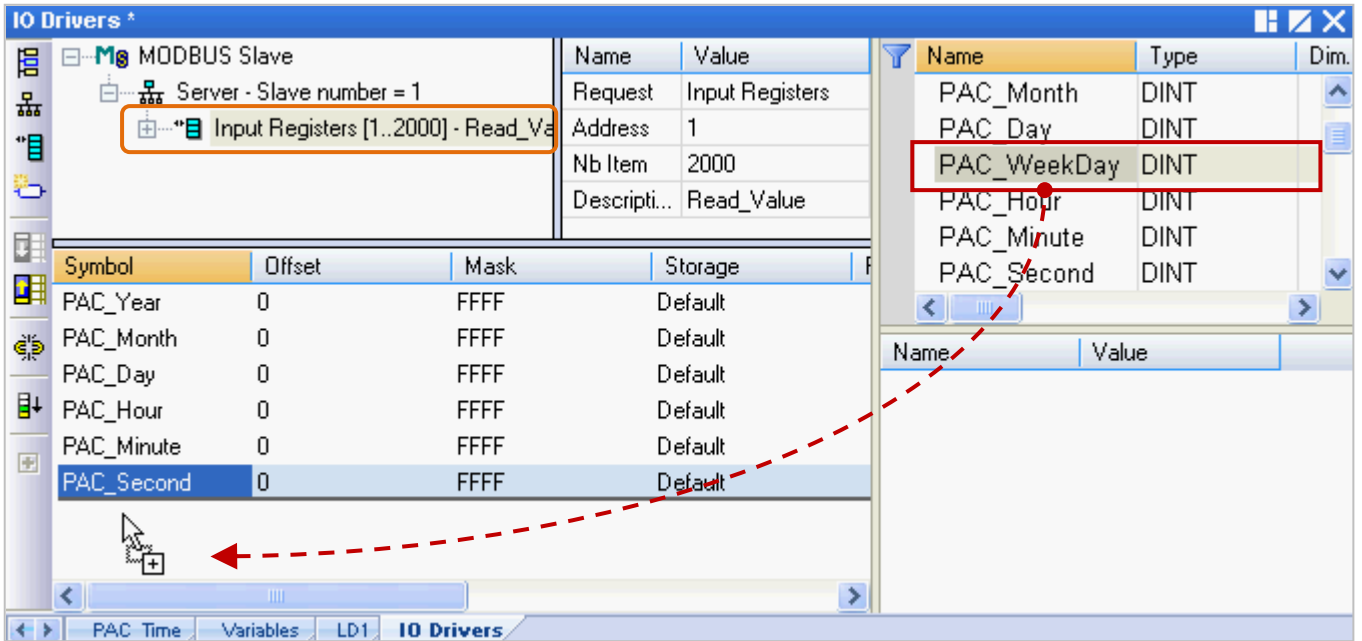
可供 Modbus Master **寫入** 資料:

選項	資料型態
Coil Bits	BOOL
Holding Registers	BYTE, INT, DINT, REAL, ... 等。

(資料型態，可參考[附錄 A](#))



- 如上圖，建議將“Base address”設定為“1”，而“Nb items”表示一個“Data block”最多可提供多少個變數資料，若 Modbus Master (例如，圖控軟體) 要求的資料位址大於此數值 (此例設定為“2000”)，則 Modbus Slave (即，Win-GRAF PAC) 將不回應。
- 將變數區的變數 (此例為“PAC_xxx”，資料型態: DINT) 以滑鼠拖曳的方式，拉到“Symbol” 區塊。



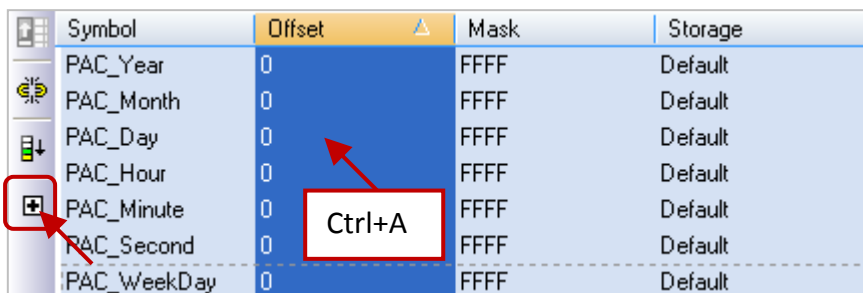
- 滑鼠雙擊“Offset” 欄位並填入數值後，按“Enter” 完成設定。

注意: (1) “Offset” 的值是由“0” 開始，而“Offset” 的值加 1 (Base address) 才是該變數的 Modbus 位址。
 (2) 若選用的是 32-bit 或以上的資料型態 (此例為 DINT)，則需佔用 2 個 Modbus 位址如下表，“0, 2, 4, 6, ...”。(更多關於資料型態，可參考[附錄 A](#)。)

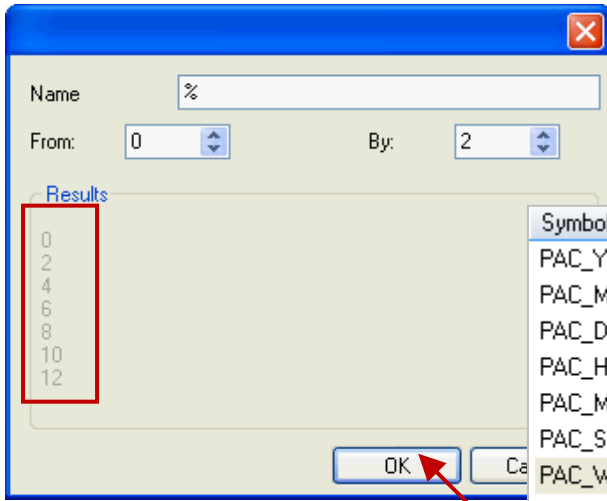
Symbol	Offset	Mask	Storage
PAC_Year	0	FFFF	Default
PAC_Month	2	FFFF	Default
PAC_Day	4	FFFF	Default
PAC_Hour	6	FFFF	Default
PAC_Minute	8	FFFF	Default
PAC_Second	0	FFFF	Default
PAC_WeekDay	0	FFFF	Default

使用小技巧:

滑鼠點選任一欄位再按“Ctrl+A” 全選，點選左側的“Iterate Property” 按鈕，開啟設定視窗。



“Name” 維持設定 “%”、將 “From” 填入 0、“By” 填入 2，再點選 “OK”。



(若 Name 修改為 “%%”，此例會顯示為 00, 22, 44, 66, 88, 1010, 1212。您可依實際需求來修改，並查看 “Results” 的顯示結果)

Symbol	Offset	Mask	Storage
PAC_Year	0	FFFF	Default
PAC_Month	2	FFFF	Default
PAC_Day	4	FFFF	Default
PAC_Hour	6	FFFF	Default
PAC_Minute	8	FFFF	Default
PAC_Second	10	FFFF	Default
PAC_WeekDay	12	FFFF	Default

9. 滑鼠點選 “Storage” 來選取整欄，再按 “Enter” 鍵顯示下拉選單，接著選擇 “DWORD (Low - High)” 再按 “Enter” 鍵完成設定。(若資料為 16-bit 或以下，不需設定 “Storage” 項目)

Symbol	Offset	Mask	Storage	Range (Low)
PAC_Year	0	FFFF	Default	
PAC_Month	2	FFFF	Default	
PAC_Day	4	FFFF	Default	
PAC_Hour	6	FFFF	Default	
PAC_Minute	8	FFFF	Default	
PAC_Second	10	FFFF	Default	
PAC_WeekDay	12	FFFF	Default	

Storage dropdown menu options:

- Default
- DWORD (High - Low)
- DWORD (Low - High)**
- STRING(6)
- STRING(8)
- STRING(10)

您可展開這個 “Data Block”，如圖，“Offset” 的值加 1 (Base address) 為該變數的 Modbus 位址。

Symbol	Offset	Mask	Storage	Range (Lo)
PAC_Year	0	FFFF	DWORD (Low - High)	
PAC_Month	2	FFFF	DWORD (Low - High)	
PAC_Day	4	FFFF	DWORD (Low - High)	
PAC_Hour	6	FFFF	DWORD (Low - High)	
PAC_Minute	8	FFFF	DWORD (Low - High)	
PAC_Second	10	FFFF	DWORD (Low - High)	
PAC_WeekDay	12	FFFF	DWORD (Low - High)	

10. 接著，需再新增一個 “Data Block” 用來供 Modbus Master 讀取 Boolean 資料。

設定方式與步驟 4 ~ 8 類似：

- (1) 滑鼠點選 “Server - ...” 再點選左側的 “Insert Slave/Data Block” 按鈕，來開啟設定視窗。
- (2) 設定視窗中，填入識別資料後，選擇供讀取的選項 “Input Bits”、設定 “Base address” 為 “1”、設定 “Nb items” 為 “2000”。

可供 Modbus Master 讀取資料：

選項	資料型態
Input Bits	BOOL
Input Registers	BYTE, INT, DINT, REAL, ... 等。

(資料型態，可參考[附錄 A](#))

(3) 將變數區的 BOOL 變數 “LED1”、“LED2” 拖曳到 “Symbol” 區域並設定其 “Offset” 為 “0”、“1”。


Name	Type
LED1	BOOL
LED2	BOOL

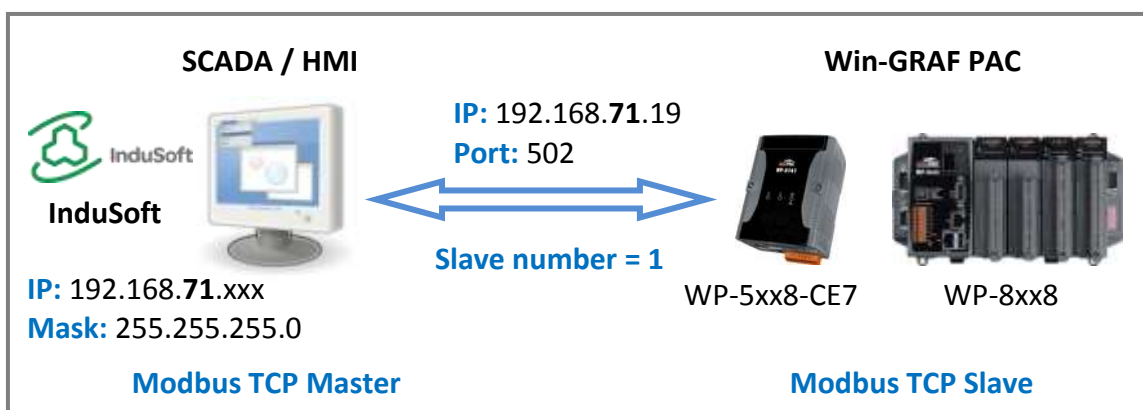
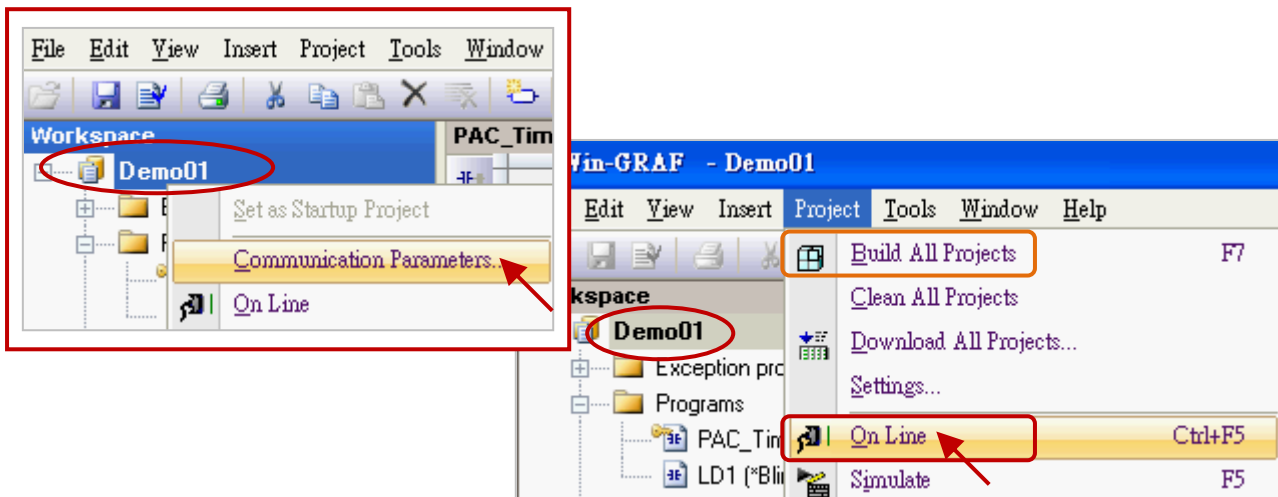
Symbol	Offset	Mask	Storage	Range
LED1	0	FFFF	Default	
LED2	1	FFFF	Default	

您已完成 Modbus Slave 設定，接下來請重新編譯並下載程式到 Win-GRAF PAC 中。

11. 點選功能表 “Project” > “Build All Projects” 重新編譯程式 (參考 [2.3.4 節](#)) 。
出現 “No error detected” 表示編譯成功。



12. 滑鼠右鍵點選專案名稱 (即，Demo01) ，再選擇 “Communication Parameters...” 設定 PAC IP (例如：192.168.71.19:502) ，點選功能表 “Project” > “On Line” (或 ) 來建立連線，並將專案下載到 Win-GRAF PAC 中 (參考 [2.3.5 節](#)) 。



(您可參考 [P1-1](#)，來查看所有支援的 PAC 型號)

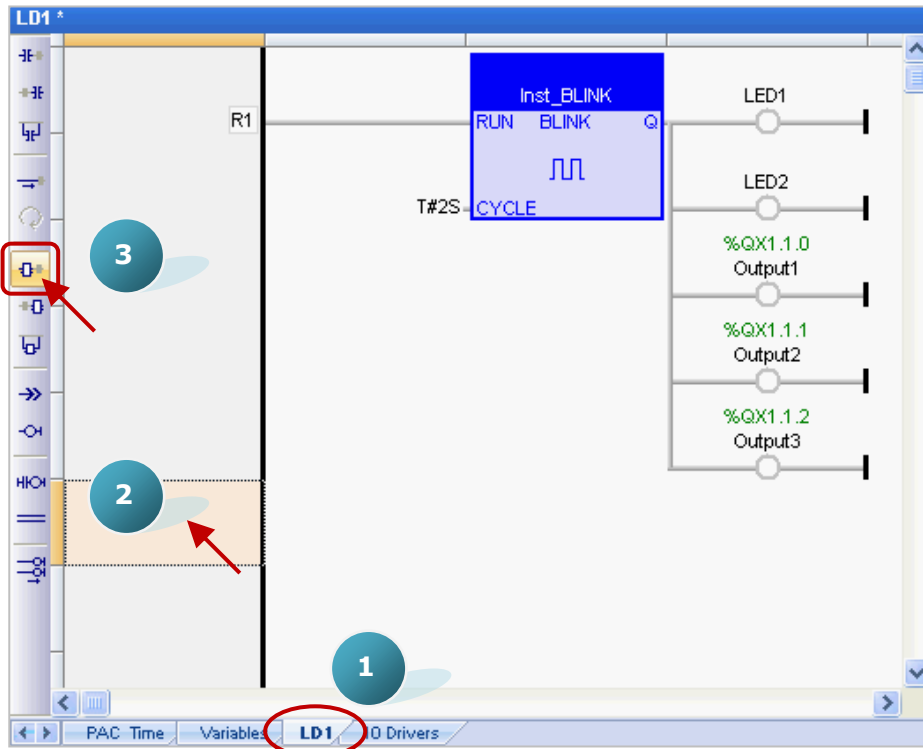
如此之後 HMI 或 SCADA 軟體便可使用 Modbus TCP protocol 來存取以上所列的、有設定 Modbus 位址的 Win-GRAF 變數。

3.2 啟用 Win-GRAF PAC 為 Modbus RTU Slave

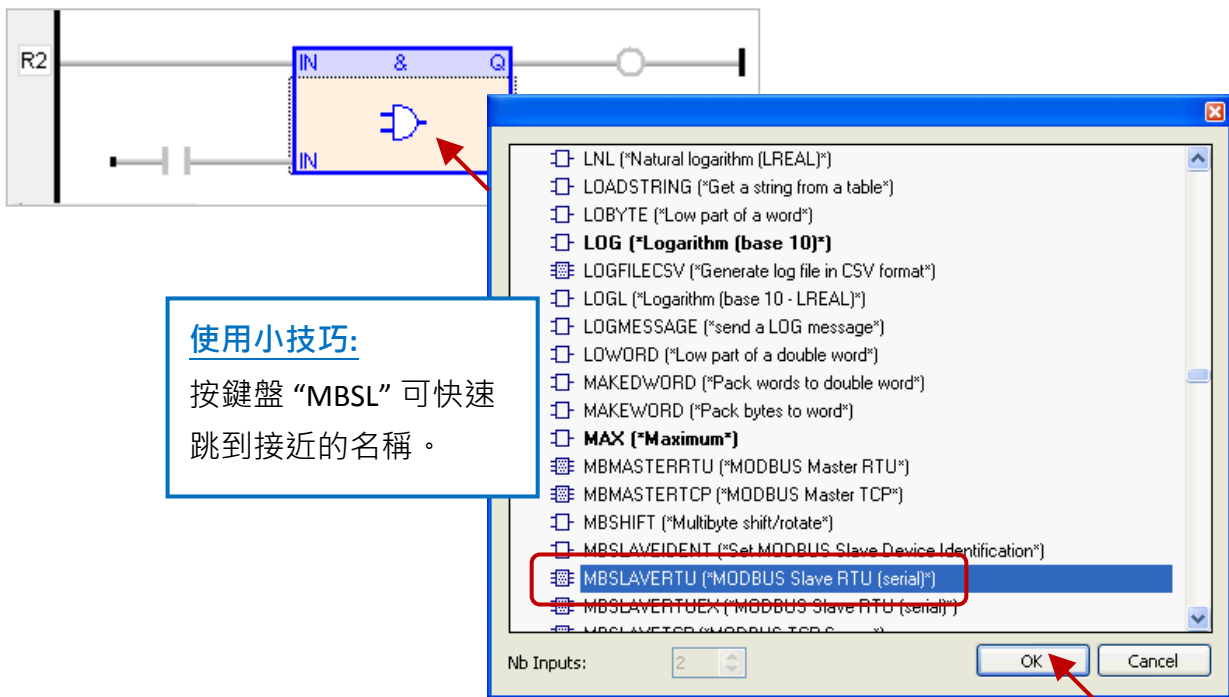
開始前，您必須先完成 [3.1 節](#) 中的內容、設定好需開放的 Modbus Slave 資料。啟用 Win-GRAF PAC 為 Modbus RTU Slave 的方式為，在程式中加入“MBSLAVERTU”或“MBSLAVERTUEX”功能方塊，請依照下列說明來完成設定。

加入“MBSLAVERTU”功能方塊

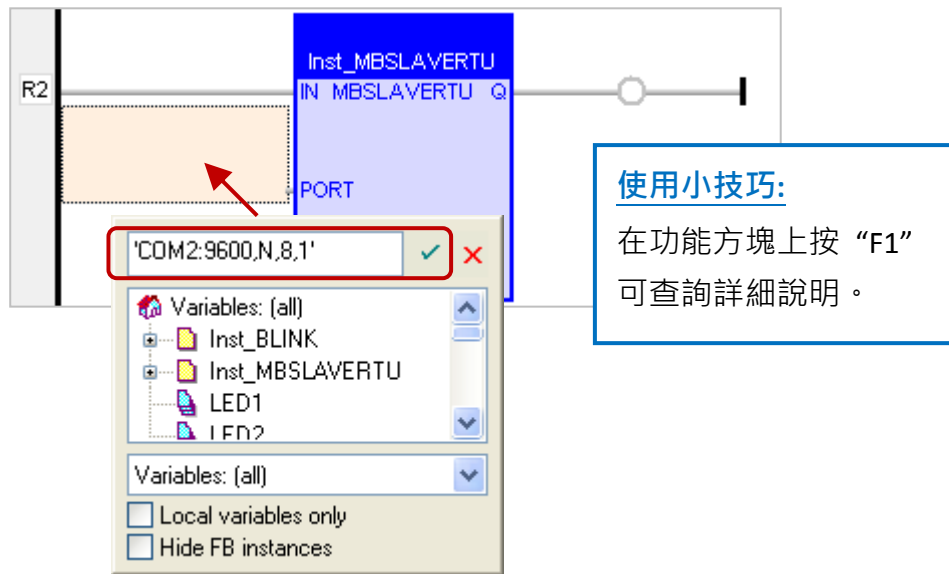
1. 在“LD1”視窗中，滑鼠點選需加入功能方塊的位置，再點選左側的“Insert FB..”按鈕。



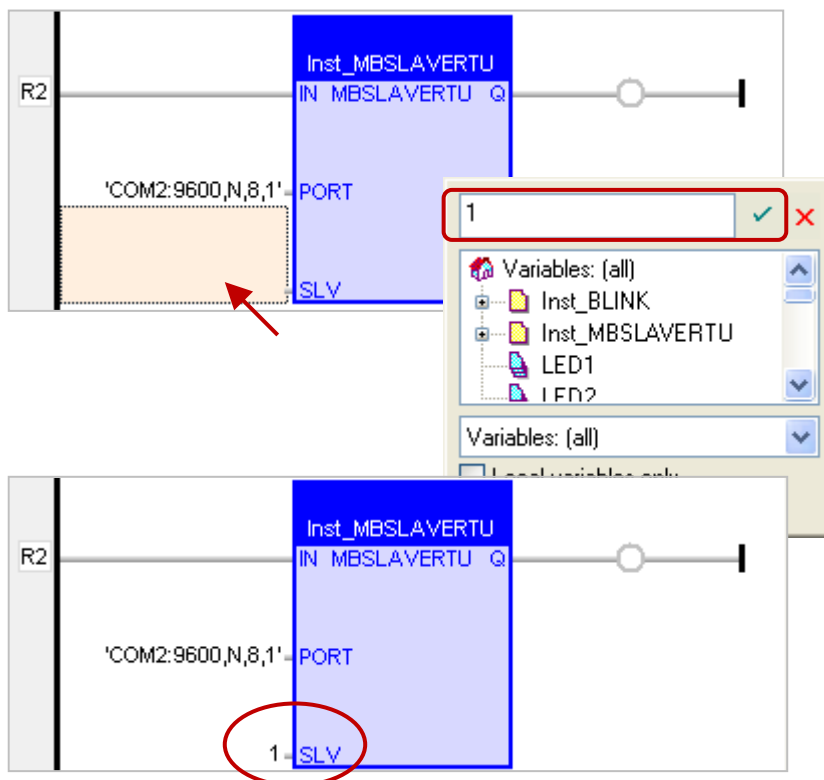
2. 滑鼠雙擊此功能方塊，並選取“MBSLAVERTU”再按“OK”。



3. 於 “MBSLAVERTU” 功能方塊，滑鼠雙擊 “PORT” 左側的區域並輸入字串 'COM2:9600,N,8,1' (表示使用 Win-GRAF PAC 的 COM2 與 Modbus Master 進行溝通)，再點選 完成設定。

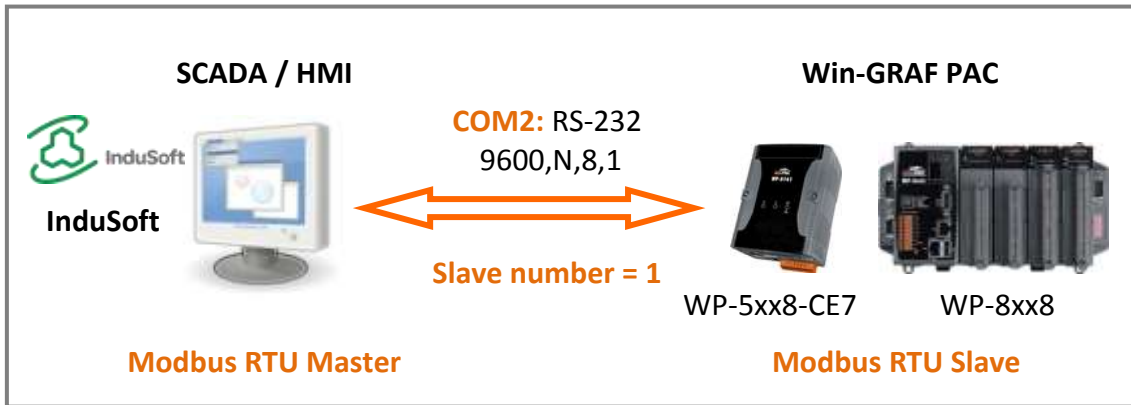


4. 滑鼠雙擊 “SLV” 左側的區域並輸入 “1” (即 3.1 節 - 步驟 3 設定的值)，再點選 完成設定。



您已完成 “MBSLAVERTU” 功能方塊的設定，接下來請重新編譯並下載程式到 Win-GRAF PAC 中。
(參考 [2.3.4 節](#) · [2.3.5 節](#))

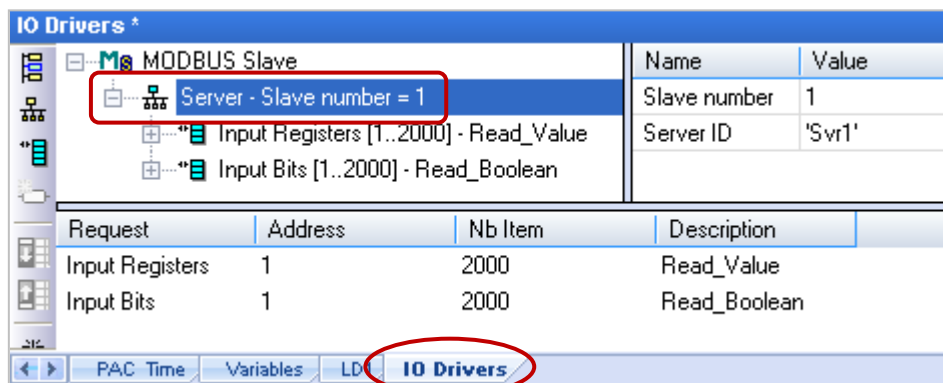
注意: 同一台 PAC 內可啟用多個 Modbus RTU Slave (建議不超過 16 Port)，方法為新增多個 “MBSLAVERTU” 功能方塊，並輸入不同的 Port 編號。



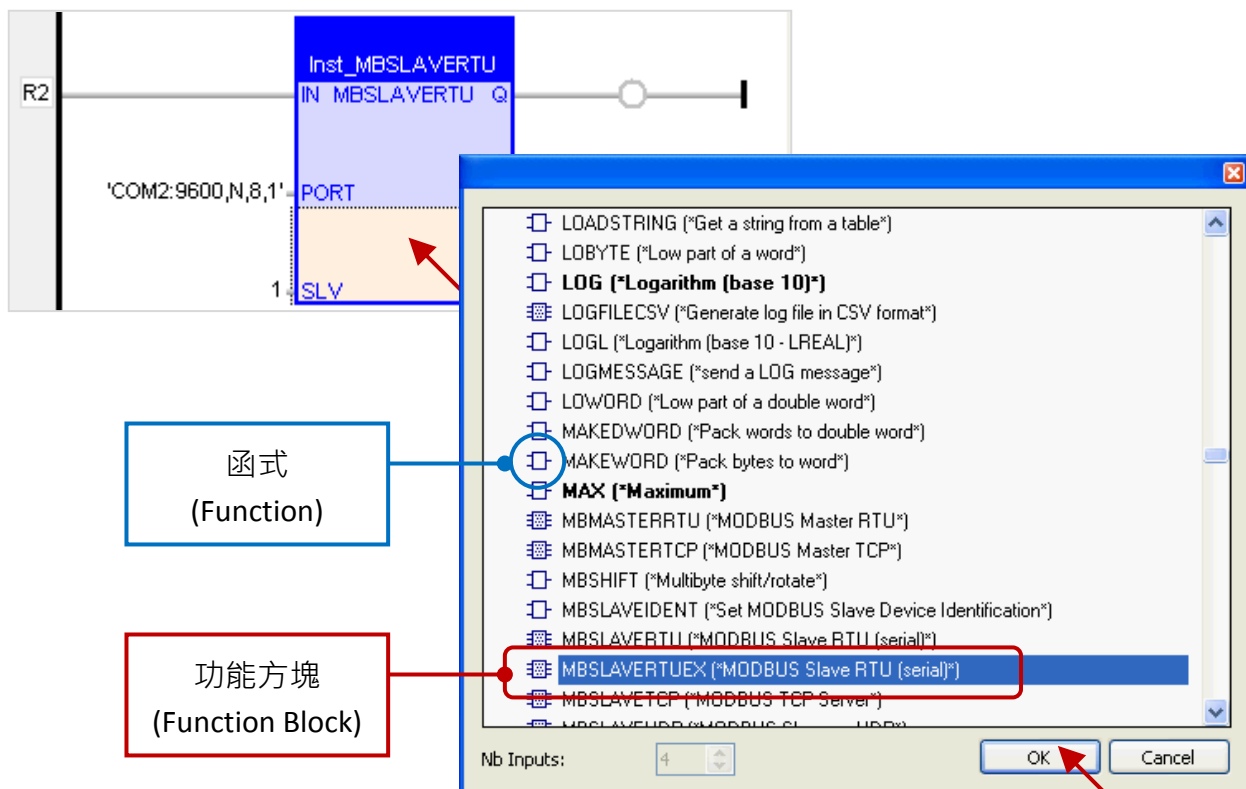
(您可參考 [P1-1](#)，來查看所有支援的 PAC 型號)

加入 “MBSLAVERTUEX” 功能方塊

若您在 “I/O Drivers” 中建立多個 “Server - ...” 設定 (建議只使用一個)，則需使用 “MBSLAVERTUEX” 功能方塊。



1. 您可依照上述步驟 1 ~ 4 來建立一個 “MBSLAVERTUEX” 功能方塊。若您想變更先前已建立的功能方塊，可用滑鼠雙擊 “MBSLAVERTU” 並重新選取為 “MBSLAVERTUEX”，再點選 “OK”。

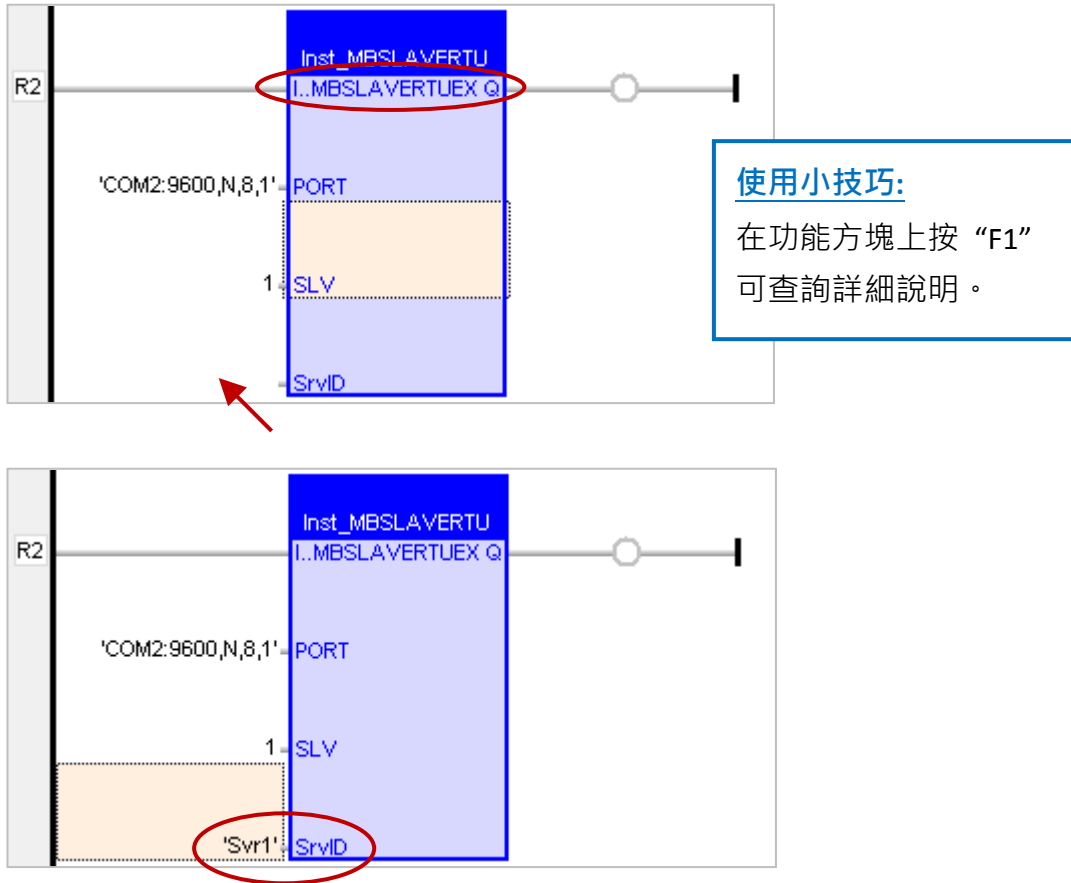


2. “MBSLAVERTUEX” 多了一個 “SrvID” 設定項目，滑鼠雙擊 “SrvID” 左側再填入所需的 “Server ID” (字串格式，例如 ‘Svr1’)。

注意:

若使用 “MBSLAVERTU” 功能方塊，只會參考第一個 “Server – Slave number..” 設定。

若使用 “MBSLAVERTUEX” 功能方塊，則會參考具有相同 “Server ID” 的 “Server – Slave...” 設定。



您已完成 “MBSLAVERTUEX” 功能方塊的設定，接下來請重新編譯程式並下載到 Win-GRAF PAC 中。(參考 [2.3.4 節](#)，[2.3.5 節](#))

第 4 章 使用 “I/O Board” 功能

此章節將介紹如何使用 Win-GRAF Workbench 中的 “I/O Boards” 功能來對應到實體的 I/O 模組 或 啟用其它功能。首先，請先了解每台 Win-GRAF PAC 的插槽編號 與 所支援的 PAC I/O 模組：

PAC 型號	插槽編號 (左至右)	插槽上支援的 I/O 模組
WP-8xx8	0 ~ 7	支援 I-8K 與 I-87K I/O 高卡模組。 (例如: I-8017HW 與 I-87055W) 但，不支援 I-8K 與 I-87K I/O 低卡模組。 (例如: I-8017H 與 I-87055)
WP-8xx8-CE7	0 ~ 7	
XP-8x38-CE6 (*)	0 ~ 7	
XP-8x48-CE6 (*)	1 ~ 7	
VP-x2x8-CE7 (*)	0 ~ 2	
WP-5238-CE7	-	掌上型 PAC，可支援一片 XV board (例如: XV107, XV116, XV308 等)。XV board 屬於 Modbus Slave I/O 板卡，請參考 5.1.6 節 ~ 5.1.12 節 來使用它。

(*) : XP-8038-CE6, XP-8048-CE6, VP-x208, WP-5xx8-CE7 為 0 槽的 PAC。 (可參考 [P1-1](#)，查詢 PAC 型號)

加入 “I/O board”

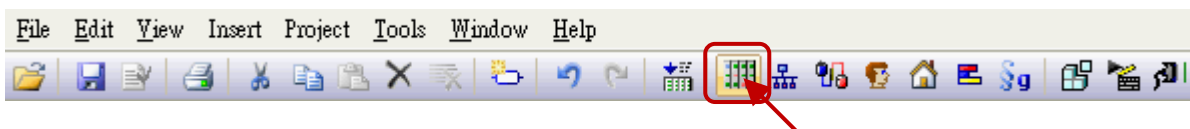
以下將 “I/O board” 稱為 “I/O 卡” 指的是軟體中的 I/O 功能 (例如: “i_8037_DO”)，而 “I/O 模組” 指的是硬體設備 (例如: “I-8037W”)。

注意:

除了在 Win-GRAF 軟體設定 I/O 卡外，有些 I/O 模組還需進行硬體上的 Jumper 設定 (例如: Single-ended 與 Differential Jumper)。因此，使用前請先至各產品網頁上查詢設定資訊 或 查看 I/O 模組的外殼說明 或 出貨附的文件。I-8K 與 I-87K 系列產品網頁：

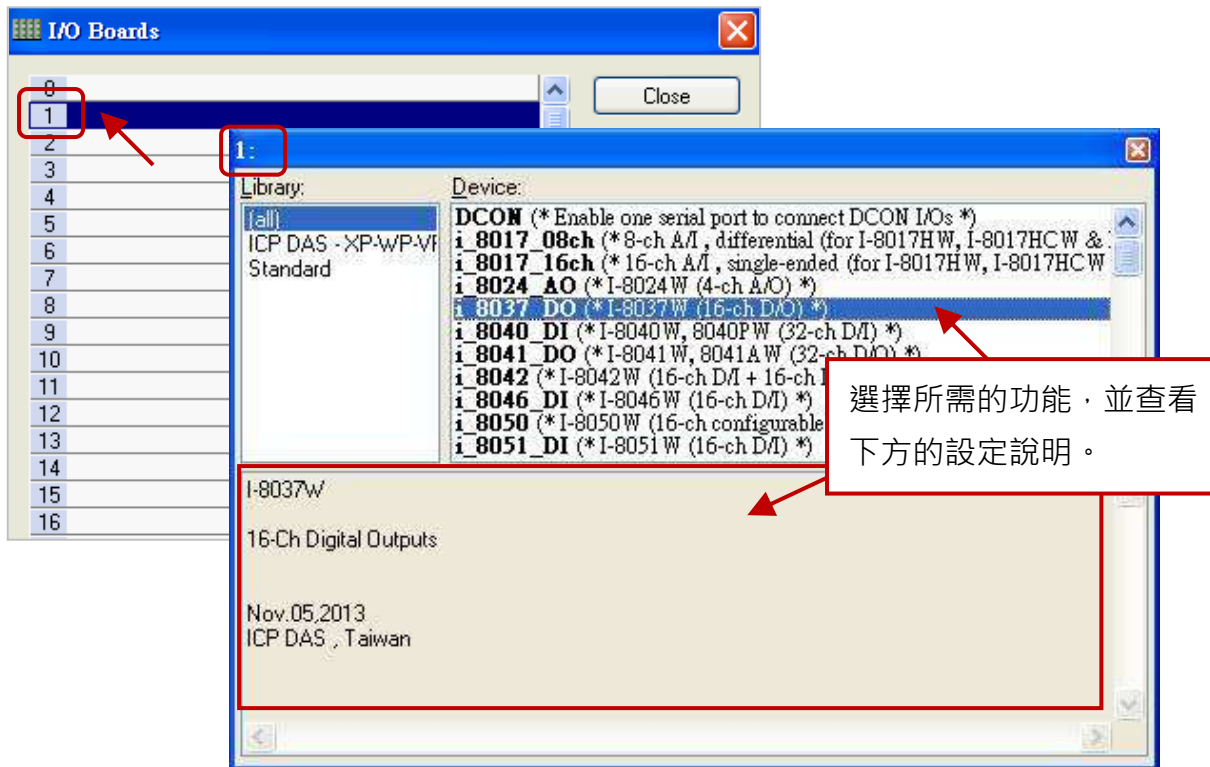
http://www.icpdas.com/root/product/solutions/remote_io/rs-485/i-8k_i-87k/i-8k_i-87k_selection.html

1. 點選 Win-GRAF 工具列的 "Open I/Os" 按鈕來開啟 “I/O Boards” 視窗。

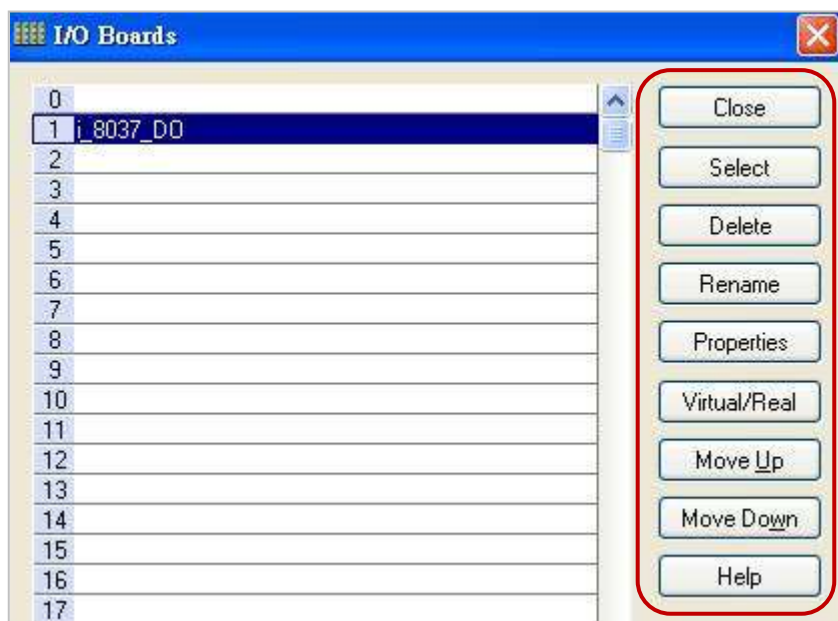


2. 如下圖，滑鼠雙擊 與 PAC I/O 模組相對應的 Slot 編號 (0 ~ 255)，再雙擊欲使用的 I/O 卡 (例如: i_8037_DO) 以完成選取。

注意: Slot0 ~ 7 是保留給 PAC I/O 模組，Slot8 (含) 以上供給其它用途使用。



按鈕說明: (點選以下按鈕可修改該 Slot 的設定)



“Close”:

關閉此視窗。

“Select”:

開啟 "Device" 功能選擇視窗。
(或按 “Enter” 開啟; “ESC” 離開)

“Delete”:

刪除此 I/O 卡。

“Rename”:

修改此 I/O 卡的名稱。

“Properties”:

查看此 I/O 卡的設定說明。

“Virtual/Real”:

將此 I/O 卡 切換為 虛擬 I/O (測試用) 或 真實 I/O。(快速鍵: Space)

“Move Up”:

將此 I/O 卡 往上移動。

“Move Down”:

將此 I/O 卡 往下移動。

“Help”:

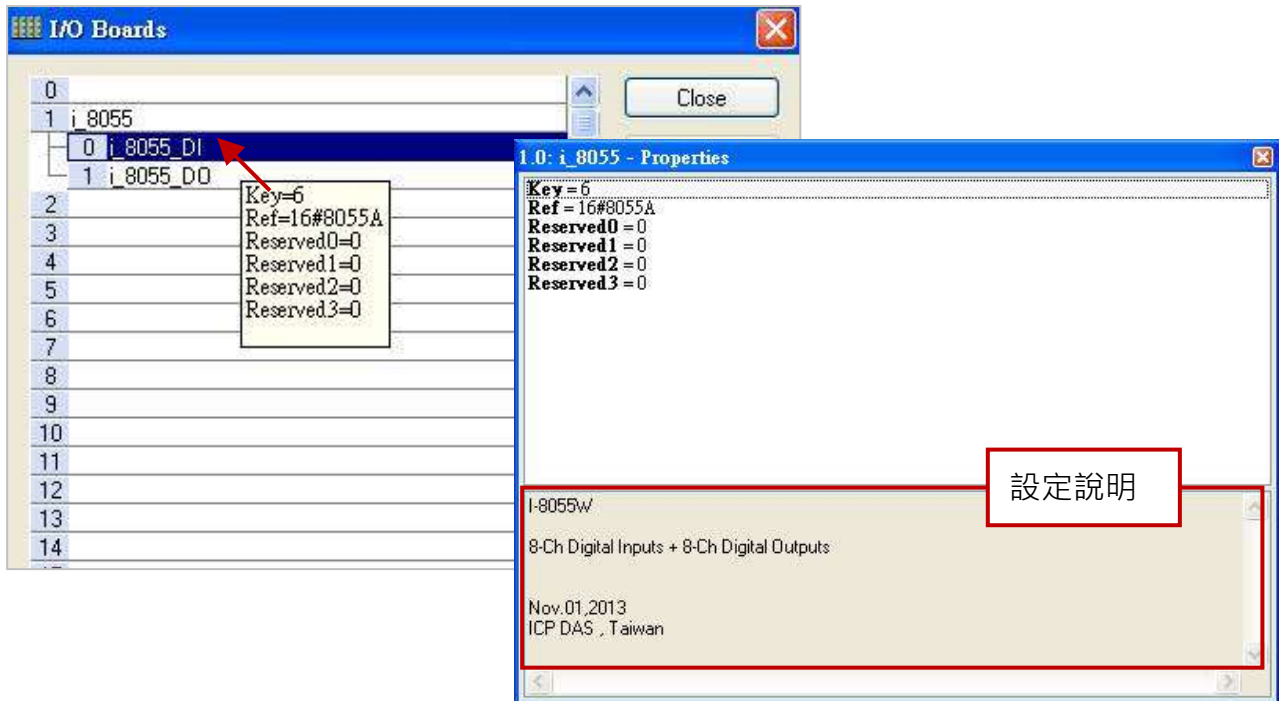
查看 I/O 卡使用說明。

4.1 DI/DO 卡

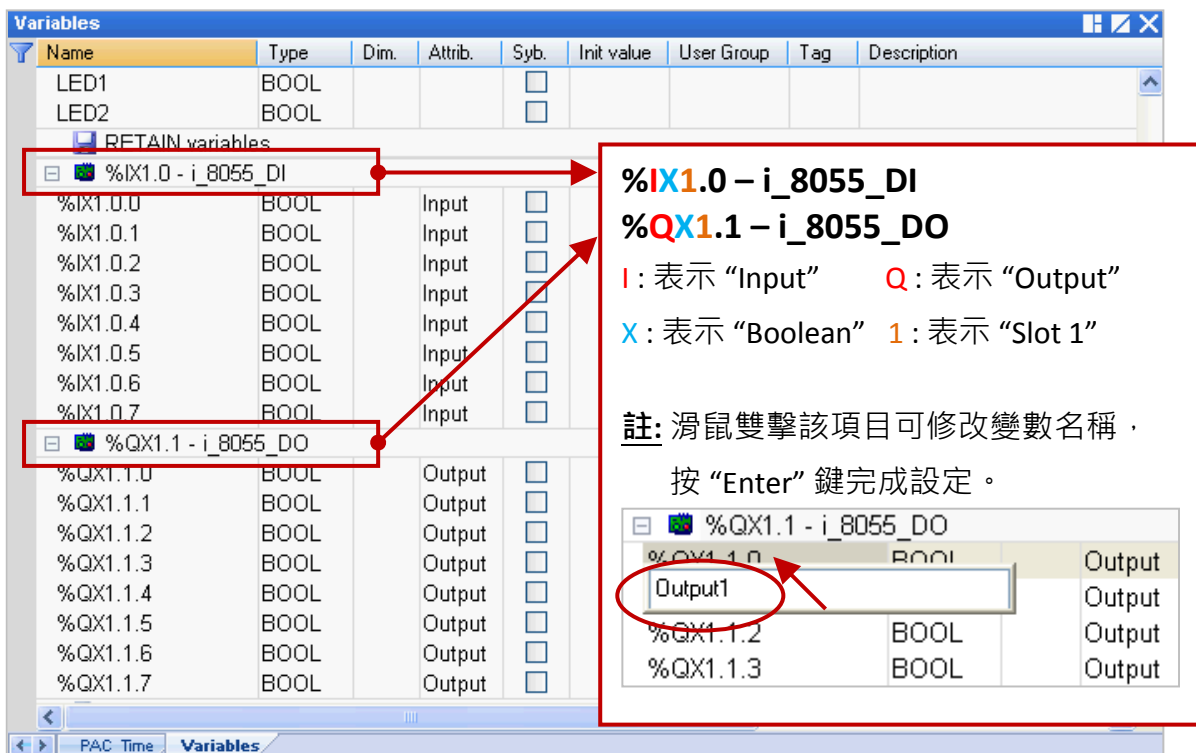
此章節以 I-8055W 為例，請參考 [第四章](#) (P4-1) 來開啟並加入 I/O 卡。

1. 滑鼠雙擊 “i_8055_DI” (或 “i_8055_DO”) 可開啟 “Properties” 視窗。

註: 滑鼠停留在 “i_8055_DI” (或 “i_8055_DO”) 也會顯示設定說明。



連上 “i_8055” I/O 卡後，會自動在 “Variables” 視窗中新增 8 個 Input 與 Output 變數供程式使用。

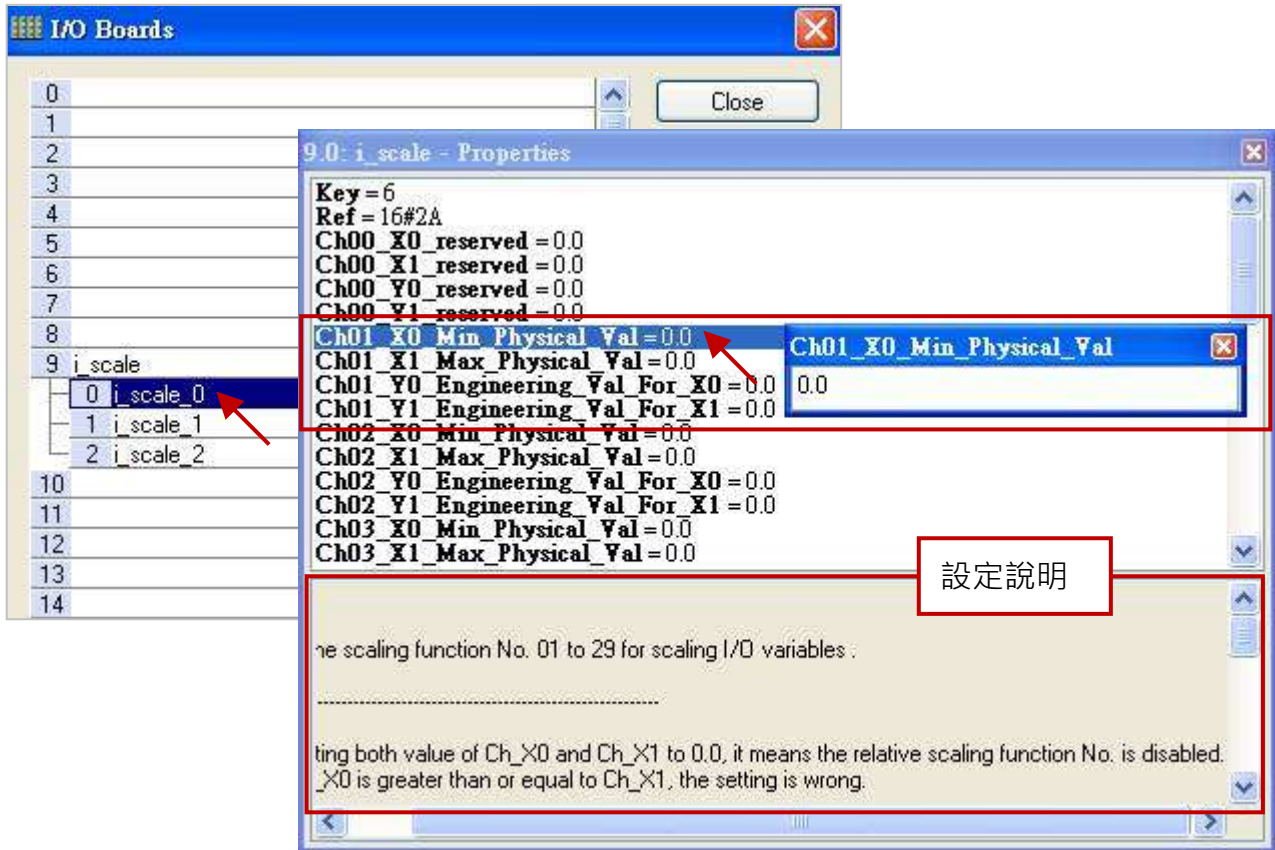


4.2 i_scale (轉換表功能)

“i_scale” 是用來設定轉換表功能，最多可設定 29 組，它適用於插在 Slot0 ~ 7 上的 AI/AO 模組，來進行數值轉換。請參考 [第四章 \(P4-1\)](#) 來開啟並加入此 I/O 卡。

1. 滑鼠雙擊 "i_scale_0" (或 "i_scale_1"、"i_scale_2") 開啟 “Properties” 視窗，來查看設定說明。

注意: Slot0 ~ 7 是保留給 PAC I/O 模組，請使用在 Slot8 (含) 以上的位置。



參數說明: (Ch 表示 Ch01 ~ Ch29，Ch00 為保留值，無需設定。)

Ch_X0_Min_Physical_Val: AI (或 AO) board 的最小值 (X0)。

Ch_X1_Max_Physical_Val: AI (或 AO) board 的最大值 (X1)。

Ch_Y0_Engineering_Val_For_X0: X0 轉換後的工程應用值。

Ch_Y1_Engineering_Val_For_X1: X1 轉換後的工程應用值。

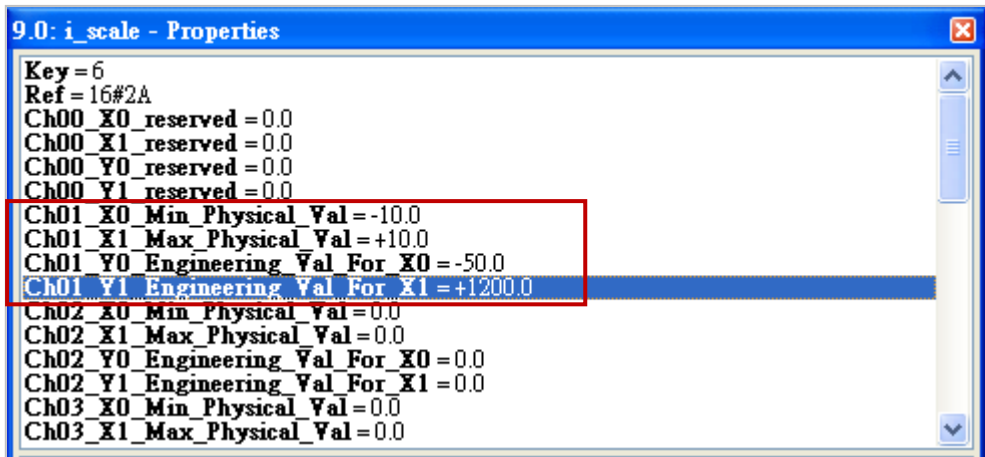
2. 滑鼠雙擊所需項目，來輸入數值並按 “Enter” 鍵完成設定。

注意:

1. 若 Ch_X0 與 Ch_X1 皆設定為 “0.0”，表示該對應編號不啟用轉換功能。
2. 若 Ch_X0 的設定值大於或等於 Ch_X1，表示設定錯誤。
3. 若 Ch_Y0 的設定值等於 Ch_Y1，表示設定錯誤。

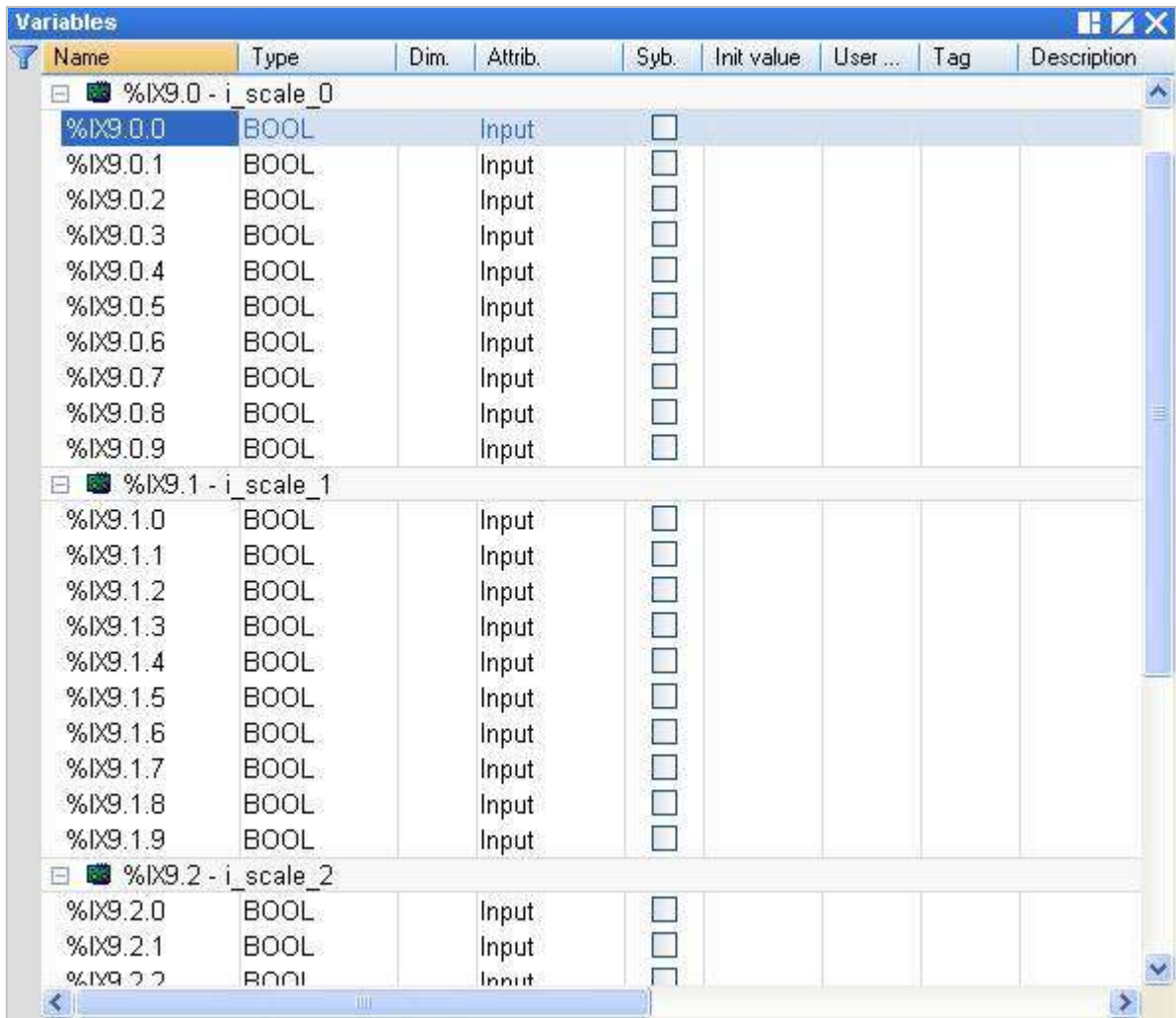
例如，若 AI board 的值為 “4 ~ 20 mA” 並想轉換為工程值 “0 ~ 10000”，請設定 Ch_X0 為 “4.0”、Ch_X1 為 “20.0”、Ch_Y0 為 “0.0” 與 Ch_Y1 為 “10000.0”。

例如，若 AO board 的值為 “-10 ~ +10 V” 並想轉換為工程值 “-50 ~ 1200”，請設定 Ch_X0 為 “-10.0”、Ch_X1 為 “+10.0”、Ch_Y0 為 “-50.0” 與 Ch_Y1 為 “+1200.0”。



3. 在 “I/O Boards” 視窗內連上 “i_scale” 後，會自動在 “Variables” 視窗中新增 30 個布林變數，當 Win-GRAF 有連上 PAC 時，會顯示出轉換功能的狀態。

- True: 表示轉換功能 OK。
- FALSE: 表示未啟用轉換功能 或 設定錯誤。

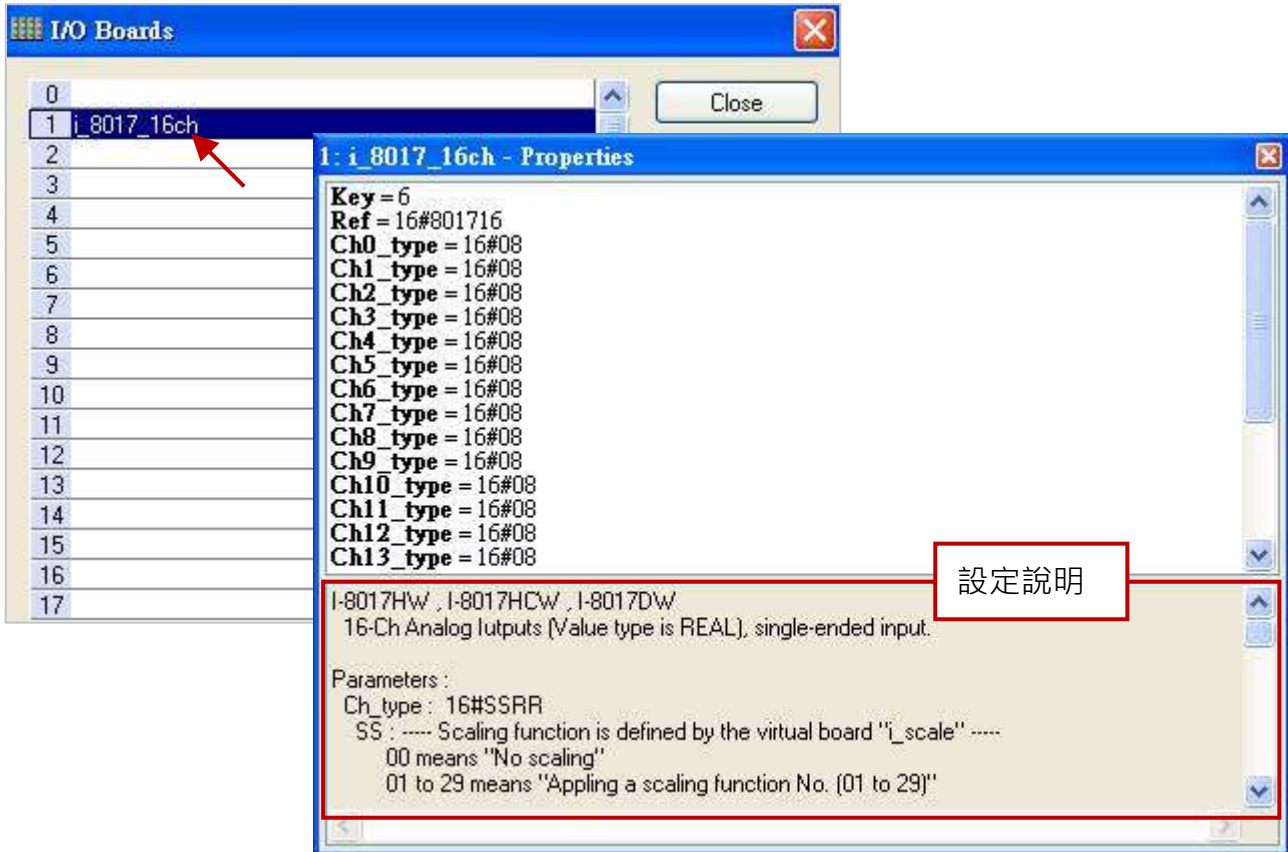


4.3 i_8017HW (8/16 通道 AI)

I-8017HW 可以是 8 通道差動式 (Differential) 或是 16 通道單端式 (Single-ended) 類比輸入模組 (資料型態為 "REAL")。此處選擇 16 通道模組來做說明，請參考 [第四章](#) (P4-1) 來開啟並加入 I/O 卡。

注意：請先設定好 I-8017HW I/O 模組上的 Jumper (Differential / Single-ended)。

1. 滑鼠雙擊 "i_8017_16ch" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

Ch_type: 16#SSRR

SS: 此值定義在 "i_scale" 轉換表功能中。(請參考 [4.2 節](#))

00 表示不使用轉換功能。

01 ~ 29 表示套用轉換表內編號 01 ~ 29 的設定。

若設定 SS 為其它值，將會採用預設值 "00"。

RR: 定義 AI 訊號型態/範圍。

05 表示 AI 模組的訊號型態/範圍為 "-2.5 ~ +2.5 V"。

06 表示 AI 模組的訊號型態/範圍為 "-20 ~ +20 mA"。

07 表示 AI 模組的訊號型態/範圍為 "-1.25 ~ +1.25 V"。

08 表示 AI 模組的訊號型態/範圍為 "-10 ~ +10 V"。

09 表示 AI 模組的訊號型態/範圍為 "-5 ~ +5 V"。

若設定 RR 為其它值，將會採用預設值 "08"。

2. 滑鼠雙擊所需項目，來輸入數值並按 "Enter" 鍵完成設定。

例如，16#08 表示 AI 模組的訊號型態/範圍為 “-10 ~ +10 V” 且不啟用轉換功能。

則通道值為 “5.67” 表示 AI 值為 “5.67 V”。

例如，16#209 表示 AI 模組的訊號型態/範圍為 “-5 ~ +5 V” 且啟用轉換功能 (即，“Ch02”)。

則 AI 值會依據 “Ch02” 的設定，來轉換為一個工程值。

例如，16#1709 表示 AI 模組的訊號型態/範圍為 “-5 ~ +5 V” 且啟用轉換功能 (即，“Ch17”)。

則 AI 值會依據 “Ch17” 的設定，來轉換為一個工程值。



Noise_Filter_Max: 設定允許最大的上限值，高於該值會視為雜訊。

Filter 會過濾資料值，預設值為 “9999.9”。

例如，設定為 “7.9”，當資料值大於 7.9 V (或 7.9 mA) 該值會被過濾掉。

Noise_Filter_Min: 設定允許最小的下限值，低於該值會視為雜訊。

Filter 會過濾資料值，預設值為 “-9999.9”。

例如，設定為 “1.5”，當資料值大於 1.5 V (或 1.5 mA) 該值會被過濾掉。

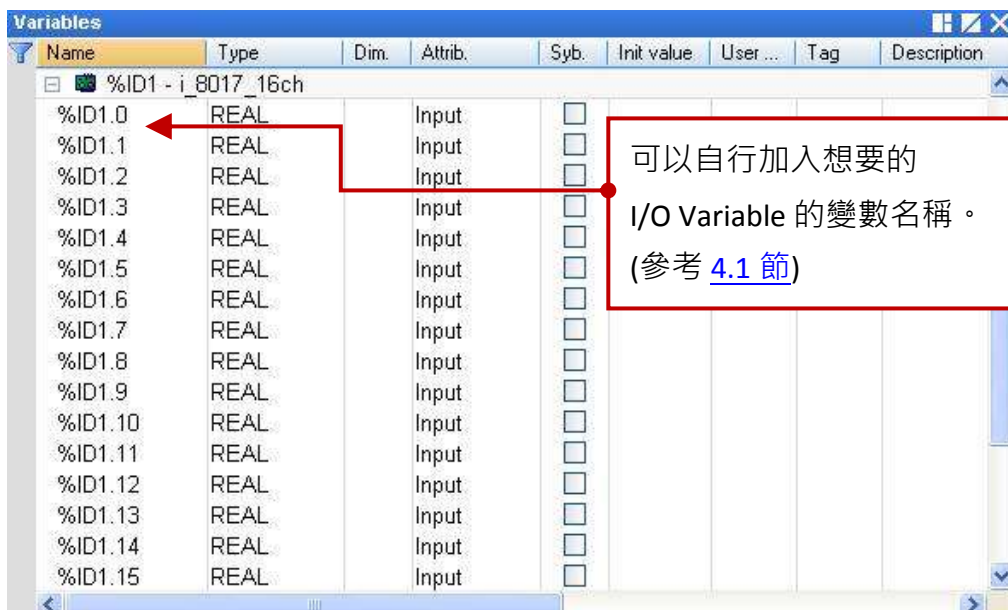
註: 若設定值 Noise_Filter_Min >= Noise_Filter_Max，Filter 將不啟用。

若設定值 Noise_Filter_Min < “-1000” 與 Noise_Filter_Max > “1000”，Filter 將不啟用。

Sample_Number: 設定取樣多少次以後，才取其平均值並更新 AI 模組的資料值，預設值為 “1”

(範圍: 1 ~ 500)。若設定為 “500” 表示最慢速的取樣率，取樣 500 次後，才會計算出平均值並更新 1 次。但此值越大，輸入波形越平滑。

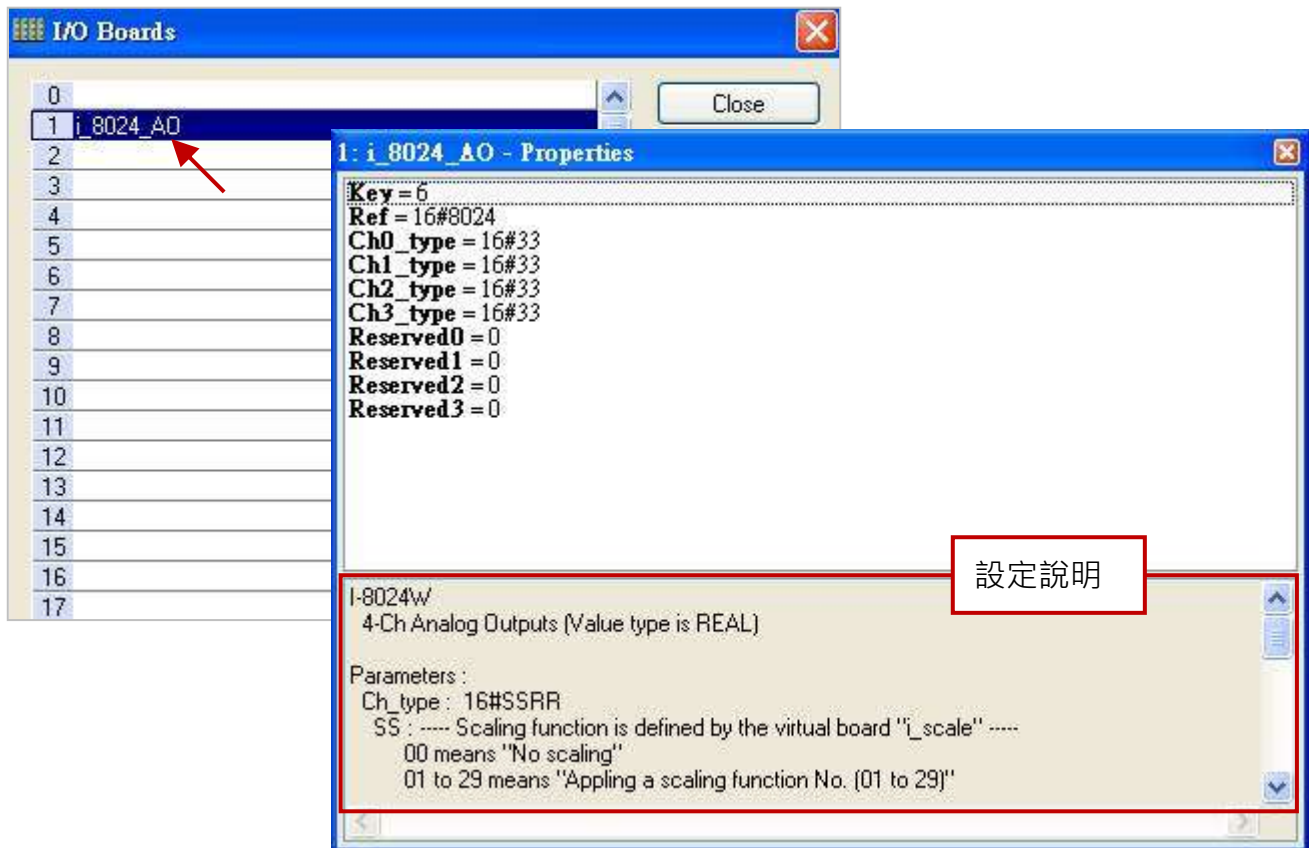
3. 在 “I/O Boards” 視窗內連上 “i_8017_16ch” 後，會自動在 “Variables” 視窗中新增 16 個 “REAL” 輸入變數，可供程式使用。



4.4 i_8024 (4 通道 AO)

I-8024W 是一款 4 通道類比輸出模組 (資料型態為 "REAL")，可輸出 +/- 10 V 或 0 ~ +20 mA 的訊號。請參考 [第四章](#) (P4-1) 來開啟並加入此 I/O 卡。

1. 滑鼠雙擊 "i_8024_AO" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

Ch_type: 16#SSRR

SS : 此值定義在 "i_scale" 轉換表功能中。(請參考 [4.2 節](#))

00 表示不使用轉換功能。

01 ~ 29 表示套用轉換表，編號 01 ~ 29 的設定。

若設定 SS 為其它值，將會採用預設值 "00"。

RR : 定義 AO 訊號型態/範圍。

30 表示 AO 模組的訊號型態/範圍為 "0 ~ +20 mA"。

33 表示 AO 模組的訊號型態/範圍為 "-10 ~ +10 V"。

若設定 RR 為其它值，將會採用預設值 "33"。

2. 滑鼠雙擊所需項目，來輸入數值並按 "Enter" 鍵完成設定。

例如，16#33 表示 AO 模組的訊號型態/範圍為 "-10 ~ +10 V" 且不啟用轉換功能。

若通道值為 "5.67" 表示 AO 值為 "5.67 V"。

若通道值為 "-3.752" 表示 AO 值為 "-3.752 V"。

- 例如，16#133 表示 AO 模組的訊號型態/範圍為 “-10 ~ +10 V” 且啟用轉換功能 (即，“Ch01”)。則 AO 值會依據 “Ch01” 的設定，來轉換為一個工程值。
- 例如，16#30 表示 AO 模組的訊號型態/範圍為 “0 ~ 20 mA” 且不啟用轉換功能。若通道值為 “12.5” 表示 AO 值為 “12.5 mA”；值為 “6.27” 表示 AO 值為 “6.27 mA”。
- 例如，16#1730 表示 AO 模組的訊號型態/範圍為 “0 ~ 20 mA” 且啟用轉換功能 (即，“Ch17”)。則 AO 值會依據 “Ch017” 的設定，來轉換為一個工程值。



- 在 “I/O Boards” 視窗內連上 “i_8024_AO” 後，會自動在 “Variables” 視窗中新增 4 個 “REAL” 輸出變數，可供程式使用。



4.5 i_87018W (8 通道 AI)

I-87018W 是一款 8 通道類比輸入模組 (資料型態為 "REAL")，可用來量測溫度 (Thermocouple)、電流 (-20 mA ~ +20 mA) 與 電壓輸入 (+/- 15 mV, +/- 50 mV, +/- 100 mV, +/- 500 mV, +/- 1 V, +/- 2.5 V)。請參考 [第四章 \(P4-1\)](#) 來開啟並加入 I/O 卡。

重要說明:

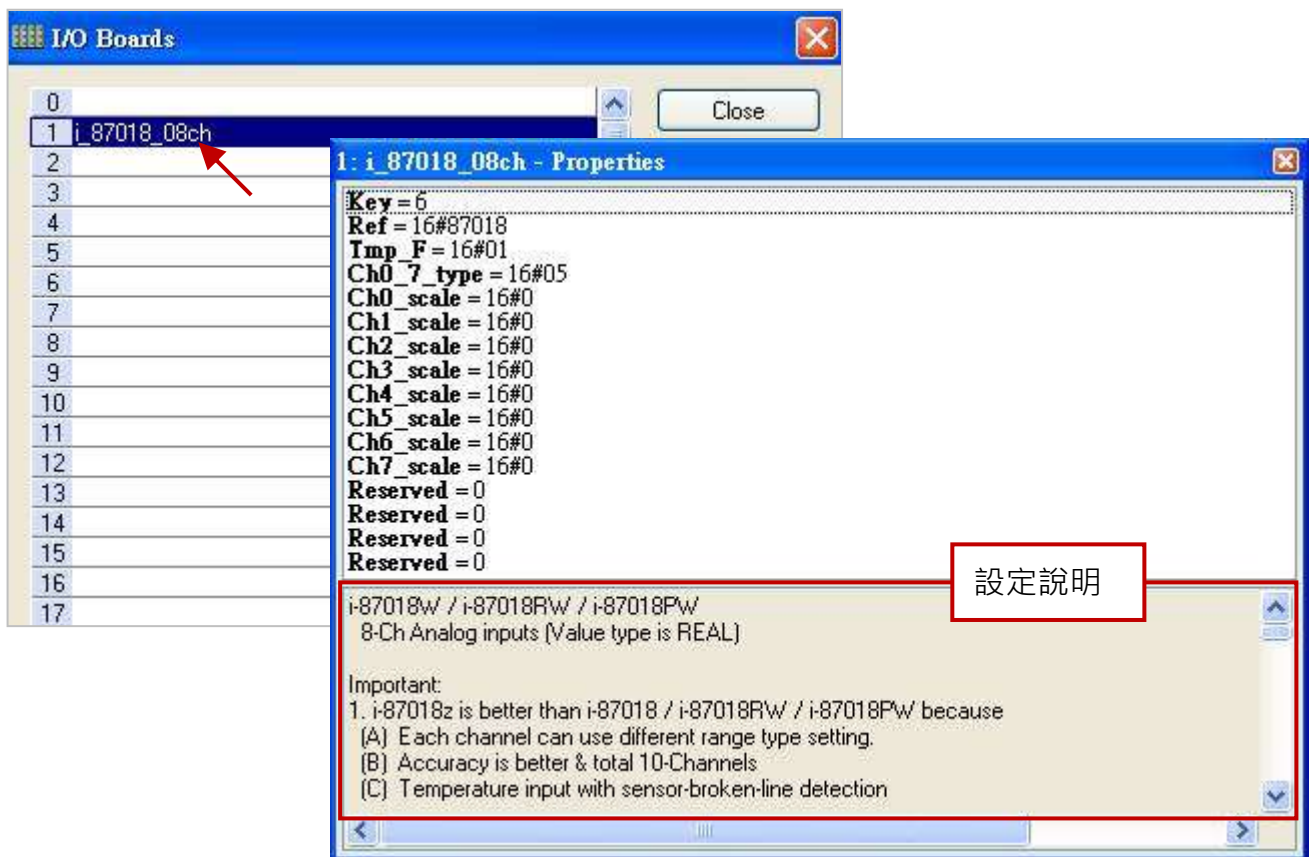
- I-87018ZW 優於 I-87018W / I-87018RW / I-87018PW，因為

- (A) 每個通道可設定為不同的輸入型態。
- (B) 量測準確度較高，且有 10 個通道。
- (C) 量測溫度輸入時，有斷線偵測功能。

請參考產品網頁: http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm

- I-87018W 未支援斷線偵測功能。

1. 滑鼠雙擊 "i_87018_08ch" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

Tmp_F: 16#FF

FF: 溫度格式，僅適用在 AI 模組為溫度 (Thermocouple) 類型時。

- 01 表示單位為攝氏溫度。
- 02 表示單位為華氏溫度。

Ch0_7_type: 16#RR

RR : 定義 AI 訊號型態/範圍。

一般範圍: (適用於 "mA" 或 "V")

設定值	AI 模組的訊號型態/範圍
00	-0.015 ~ +0.015 V
01	-0.05 ~ +0.05 V
02	-0.1 ~ +0.1 V
03	-0.5 ~ +0.5 V
04	-1 ~ +1 V
05	-2.5 ~ +2.5 V
06	20 ~ +20 mA

當 I-87018W 與 I-87018RW 用來量測電流時，需外接一個 125 Ohm 電阻。

Thermocouple 範圍: (適用於溫度量測)

設定值	類型	AI 模組的訊號型態/範圍
0E	J	-210 ~ +760 °C
0F	K	-270 ~ +1372 °C
10	T	-270 ~ +400 °C
11	E	-270 to +1000 °C
12	R	0 ~ +1768 °C
13	S	0 ~ +1768 °C
14	B	0 ~ +1820 °C
15	N	0 ~ +2320 °C
17	L	-200 ~ +800 °C
18	M	-200 ~ +100 °C
19	L _{DIN43710}	-200 ~ +900 °C

若設定 RR 為其它值，將會採用預設值 "05"。

2. 滑鼠雙擊所需項目，來輸入數值並按 "Enter" 鍵完成設定。

例如，16#05 表示 AI 模組的訊號型態/範圍為 "-2.5 ~ +2.5 V" 且不啟用轉換功能。

若通道值為 "1.28" 表示 AI 值為 1.28 V"。

若通道值為 "-0.752" 表示 AI 值為 "-0.752 V"。

例如，若 "Ch0_7_type" 設定為 16#0F 且 "Tmp_F" 設定為 16#01，表示

AI 模組的訊號型態/範圍為 "-270 ~ +1372 °C"，若通道值為 "25.75" 表示 "25.75 °C"。

例如，若 "Ch0_7_type" 設定為 16#10 且 "Tmp_F" 設定為 16#02，表示

AI 模組的訊號型態/範圍為 "-454 ~ +752 °F"，若通道值為 "25.75" 表示 "25.75 °F"。



注意: 若使用具有斷線偵測功能的溫度模組 (例如: I-87018ZW)，溫度值大於 "9000.0" 表示，

1. 溫度感測器可能斷線。
2. 溫度感測器可能損毀。
3. DCON 模組的設定與溫度感測器不符。
4. 感測器量到錯誤的電阻值。

Ch0_scale ~ Ch7_scale: 16#SS

SS : 此值定義在 "i_scale" 轉換表功能中。(請參考 [4.2 節](#))

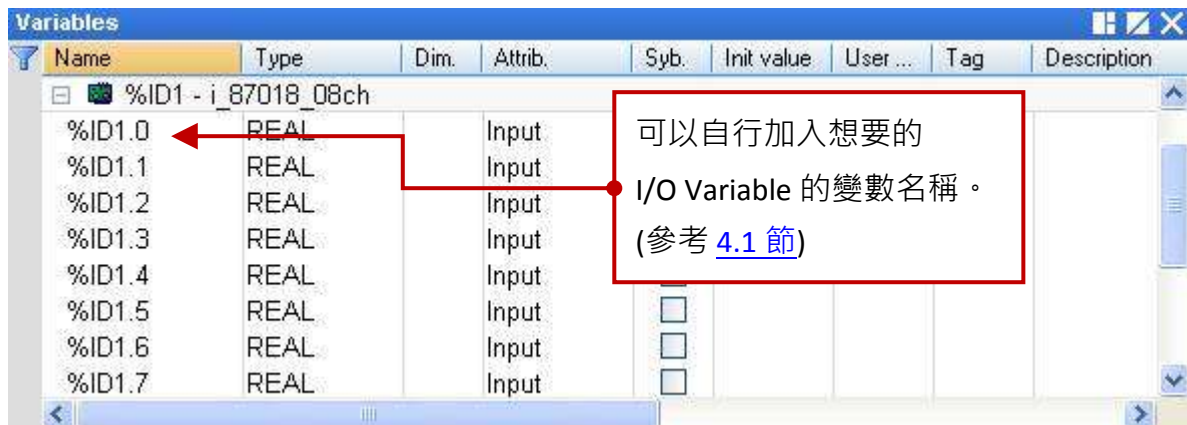
00 表示不使用轉換功能。

01 ~ 29 表示套用轉換表，編號 01 ~ 29 的設定。

若設定 SS 為其它值，將會採用預設值 "00"。

例如，16#17 表示 AI 值會依據 "Ch17" 的設定，來轉換為一個工程值。

3. 在 "I/O Boards" 視窗內連上 "i_87018_08ch" 後，會自動在 "Variables" 視窗中新增 8 個 "REAL" 輸入變數，可供程式使用。



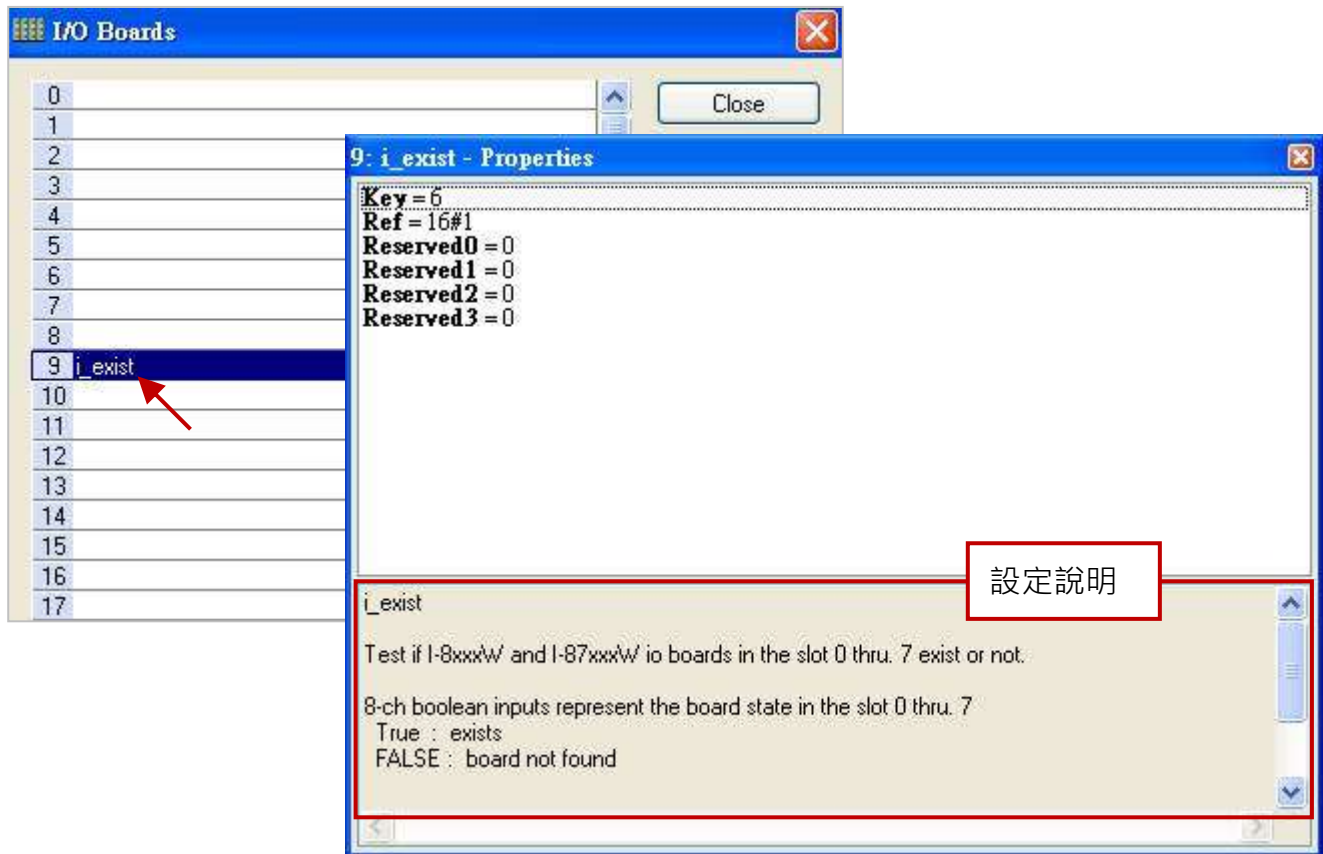
4.6 i_exist (測試 I/O 模組是否存在?)

“i_exist” 是用來測試 PAC 上 (Slot0 ~ 7) 的 I-8K 與 I-87K I/O 模組是否存在?

請參考 [第四章](#) (P4-1) 來開啟並加入此 I/O 卡。

1. 滑鼠雙擊 "i_exist" 開啟 “Properties” 視窗，來查看設定說明。

注意: Slot0 ~ 7 是保留給 PAC I/O 模組，請使用在 Slot8 (含) 以上的位置。



2. 在 “I/O Boards” 視窗內連上 “i_exist” 後，會自動在 “Variables” 視窗中新增 8 個 “BOOL” 輸入變數，當 Win-GRAF 有連上 PAC 時，會顯示出 Slot 0 ~ 7 上 I/O 模組的狀態。

TRUE: 表示 I/O 存在。

FALSE: 表示找不到 I/O。

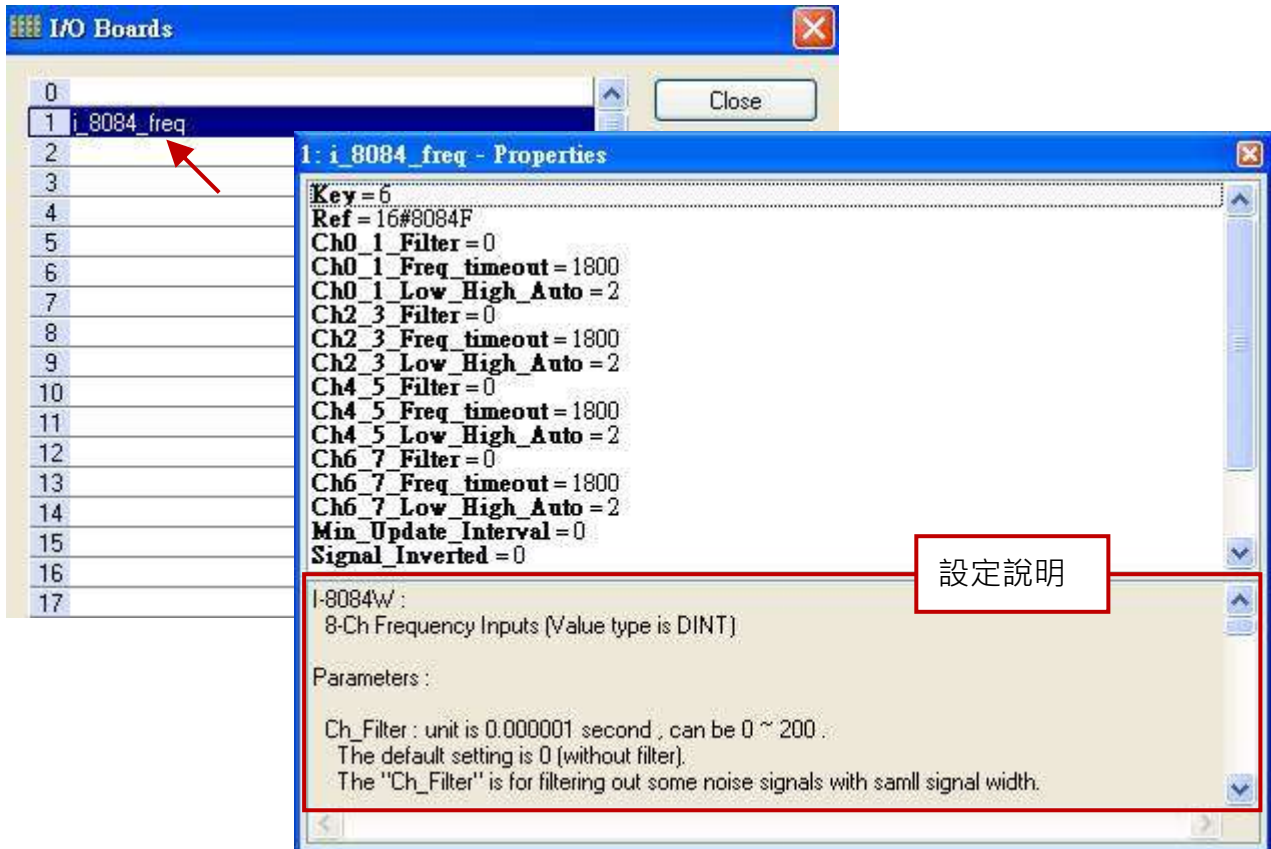


4.7 i_8084 (頻率量測, UP/Down 計數器, UP 計數器)

I-8084W 是一款高速的 4/8 通道頻率/計數器模組 (資料型態為 "DINT")，可用於頻率量測、UP/Down 計數器、UP 計數器，以下將介紹此三種模式。請參考 [第四章](#) (P4-1) 來開啟並加入此 I/O 卡。

4.7.1 i_8084_freq (頻率量測)

1. 滑鼠雙擊 "i_8084_freq" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

Ch_Filter: 單位為微秒 ($\mu\text{s} = 0.000001$ 秒)，可設為 "0" ~ "200"，預設值為 "0" (不過濾)。

"Ch_Filter" 是用來過濾掉一些很小的雜訊寬度。建議設為 "0"，若無雜訊考量或採用即時 (Real-time) 量測。下表為建議的設定:

最大輸入頻率 (Hz)	建議的設定值
1K	200
2K	100
5K	40
10K	20
20K	10
100K	2
450K	1
450K	0 (未過濾)

Ch_Freq_Timeout:

單位為毫秒 (ms = 0.001 秒) , 可設為 "20" ~ "1800" , 若設定為其它值 , 將會採用 "1800" 。

若在 "Ch_Freq_Timeout" 間隔時間中 , I-8084W 沒有任何訊號波形輸入 , 則相關通道的頻率值將設定為 "0" 。

例如 , 設定此值為 100 ms 且輸入頻率為 500 Hz (表示約 2 ms 產生一個訊號波形) 。

當輸入頻率下降到 9 Hz (表示約 111 ms 產生一個訊號波形) , 此 "111" ms 已經超過了設定值 "100" ms (即 "Freq_Timeout") 。

因為在 100 ms 的間隔時間內 , 沒有訊號波形產生 , 因此 , 頻率值將會設定為 "0" 。

設定為 20 ms 時 , 將無法檢測 50 Hz 以下的頻率值 (變為 "0") 。

設定為 100 ms 時 , 將無法檢測 10 Hz 以下的頻率值 (變為 "0") 。

設定為 1800 ms 時 , 可檢測 0 Hz 、 1 Hz ~ 450 KHz 的頻率值 。

Ch_Low_High_Auto: (建議設定為 "2 : 自動模式")

0 表示低頻模式 。

1 表示高頻模式 。

2 表示自動切換為 低頻 或 高頻模式 。

若設定為其它值 , 將會採用預設值 "2 : 自動模式" 。

模式 2 會自動切換頻率模式 , 當輸入頻率大於 3500 Hz 會切換到高頻模式 , 當輸入頻率小於 1000 Hz 會切換到低頻模式 。

若輸入頻率通常小於 1000 Hz 或 常不正確 , 請勿設定為 1 (高頻模式) 。

若輸入頻率通常大於 3500 Hz , 請勿設定為 0 (低頻模式) 。

Min_Update_Interval:

單位為毫秒 (ms = 0.001 秒) , 可設為 "0" 或 "20" ~ "1000" 。

預設值 "0" 表示 "每達到 PAC Cycle 時間 , 更新頻率一次" 。

其餘設定值表示 "每達到此間隔時間 , 更新頻率一次" 。

頻率的更新時間也取決於 Win-GRAF PAC 的 Cycle 時間 , 若 PAC Cycle 時間為 200 ms 而

"Min_Update_Interval" 設定小於 200 , 則實際的頻率更新時間將為 200 ms 。

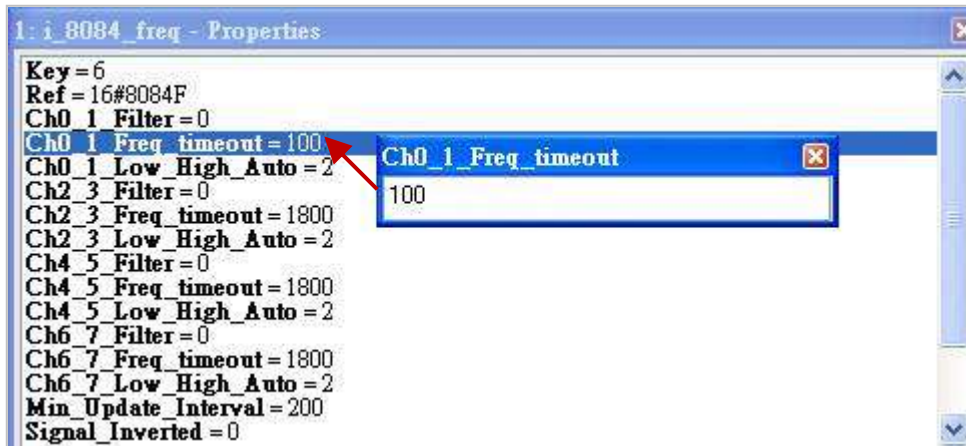
設定 "Min_Update_Interval" 為較大的值 , 將得到較平滑的頻率波形值 , 但頻率值更新緩慢 。

Signal_Inverted:

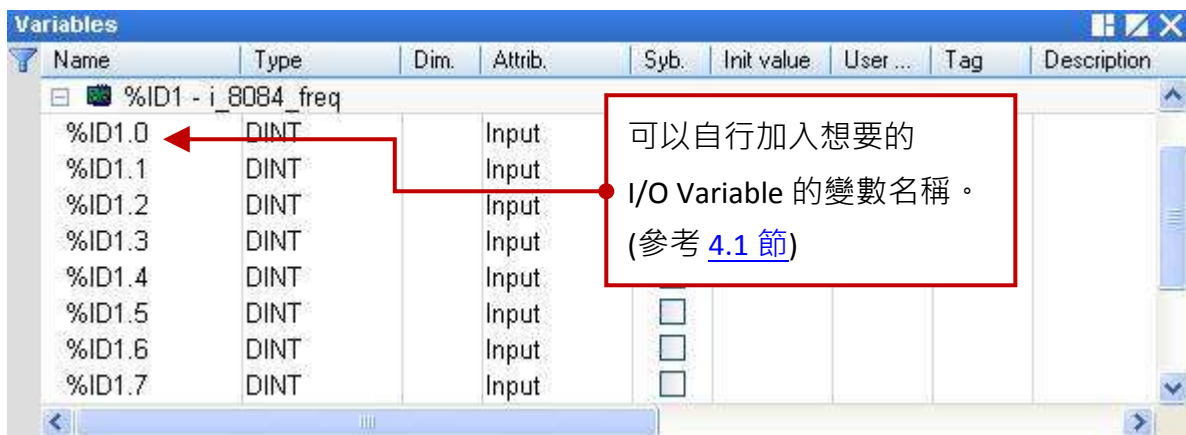
0: 輸入訊號正常 (無反相)

1: 輸入訊號為反相 (表示電壓 HIGH 會視為 LOW 處理 , 而電壓 LOW 會視為 HIGH 處理)

2. 滑鼠雙擊所需項目，來輸入數值並按“Enter”鍵完成設定。



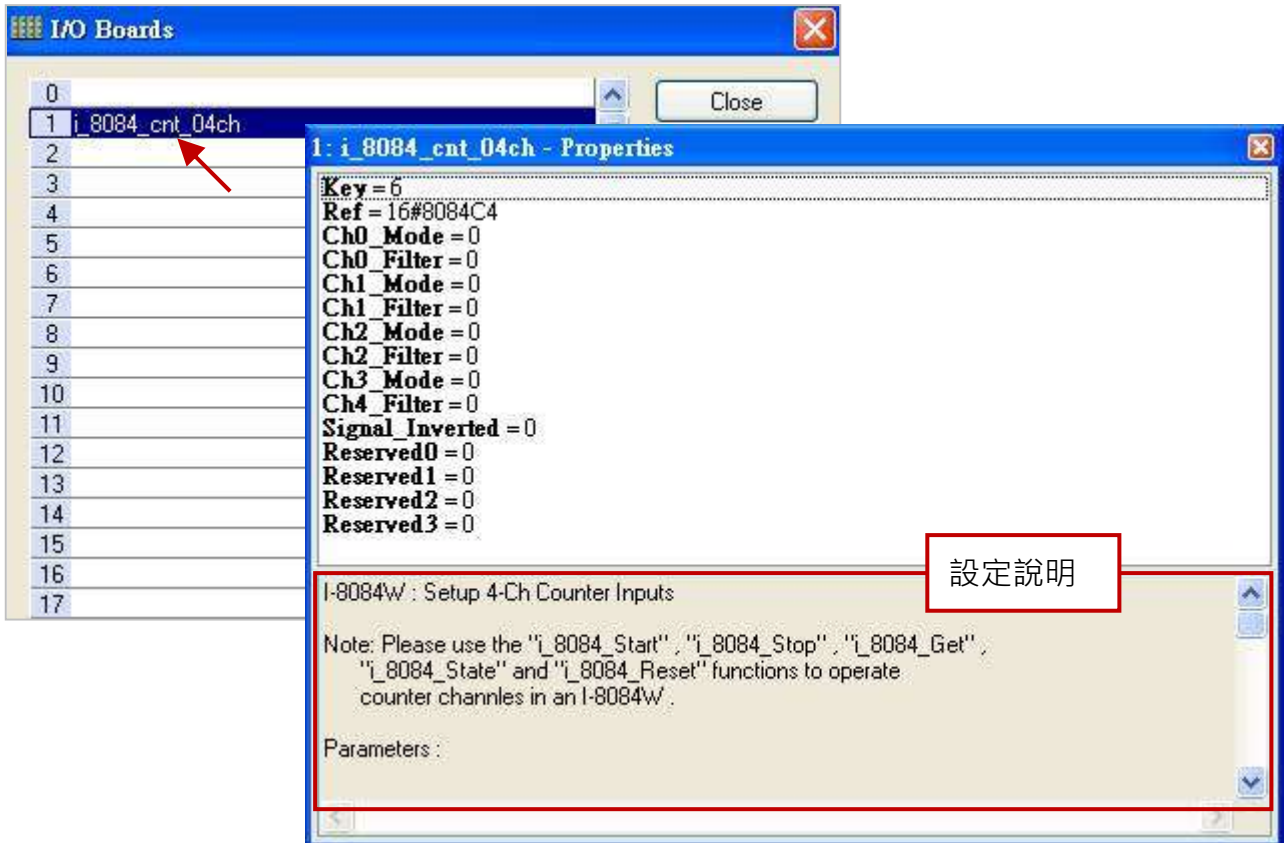
3. 在“I/O Boards”視窗內連上“i_8084_freq”後，會自動在“Variables”視窗中新增 8 個“DINT”輸入變數，可用來顯示各通道的頻率值。



4.7.2 i_8084_cnt_ch04 (4 通道 UP/Down 計數器)

註: 請使用 Win-GRAF Workbench 中的 "COUNTER_START"、"COUNTER_STOP"、"COUNTER_GET"、"COUNTER_STATE" 與 "COUNTER_RESET" 函式 (Function)，來操作 I-8084W 的計數通道。

1. 滑鼠雙擊 "i_8084_cnt_ch04" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

Ch_Mode: 輸入模式可設定為 "0"、"1" 與 "4"，若設定為其它值，將會採用 "0"。

- 0: Pulse/DIR 模式。
- 1: UP/DOWN 模式。
- 4: A/B phase (Quard.) 模式。

Ch_Filter: 單位為微秒 ($\mu s = 0.000001$ 秒)，可設為 "0" ~ "200"，預設值為 "0" (不過濾)。

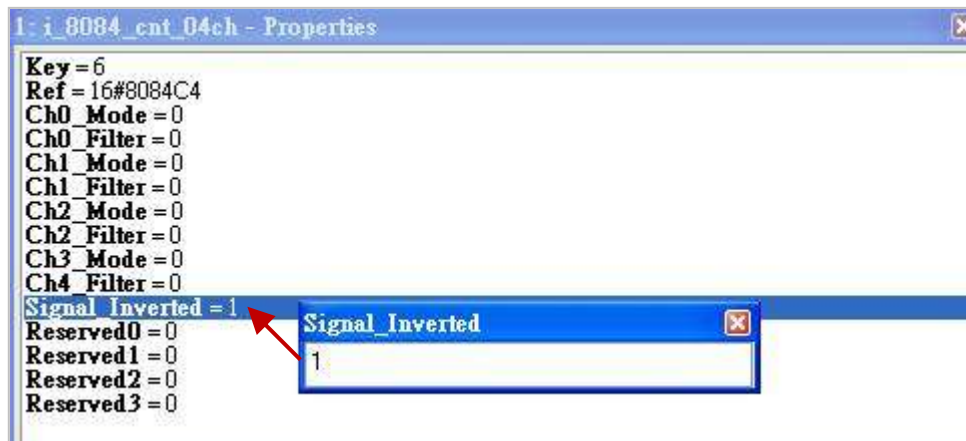
"Ch_Filter" 是用來過濾掉一些很小的雜訊寬度。建議設為 "0"，若無雜訊考量或採用即時 (Real-time) 量測。下表為建議的設定:

最大輸入頻率 (Hz)	建議的設定值
1K	200
2K	100
5K	40
10K	20
20K	10
100K	2
450K	1
450K	0 (未過濾)

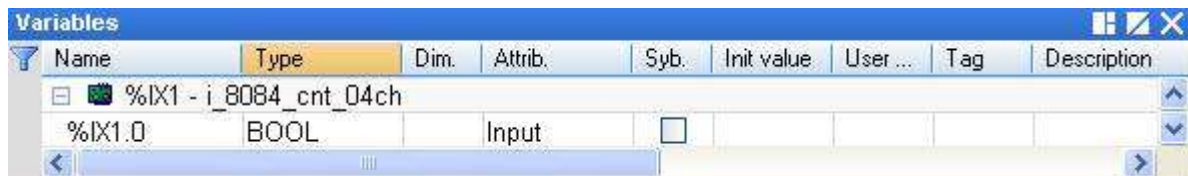
Signal_Inverted:

- 0: 輸入訊號正常 (無反相)。
 - 1: 輸入訊號為反相 (表示電壓 HIGH 會視為 LOW 處理，而電壓 LOW 會視為 HIGH 處理)。
- 例如，若 "Signal_Inverted" 設為 "0" (無反相) 且 "Ch_Mode" 設為 "0" (Pulse/DIR)，則當 "DIR" 訊號為 High 時，計數器將會開始計數。
- 若 "Signal_Inverted" 設為 "1" (反相) 且 "Ch_Mode" 設為 "0" (Pulse/DIR)，則當 "DIR" 訊號為 High 時，計數器將會倒數計數。

- 2. 滑鼠雙擊所需項目，來輸入數值並按 "Enter" 鍵完成設定。



- 3. 在 "I/O Boards" 視窗內連上 "i_8084_cnt_ch04" 後，會自動在 "Variables" 視窗中新增 1 個 "BOOL" 輸入變數 (無作用，固定為 "FALSE")。

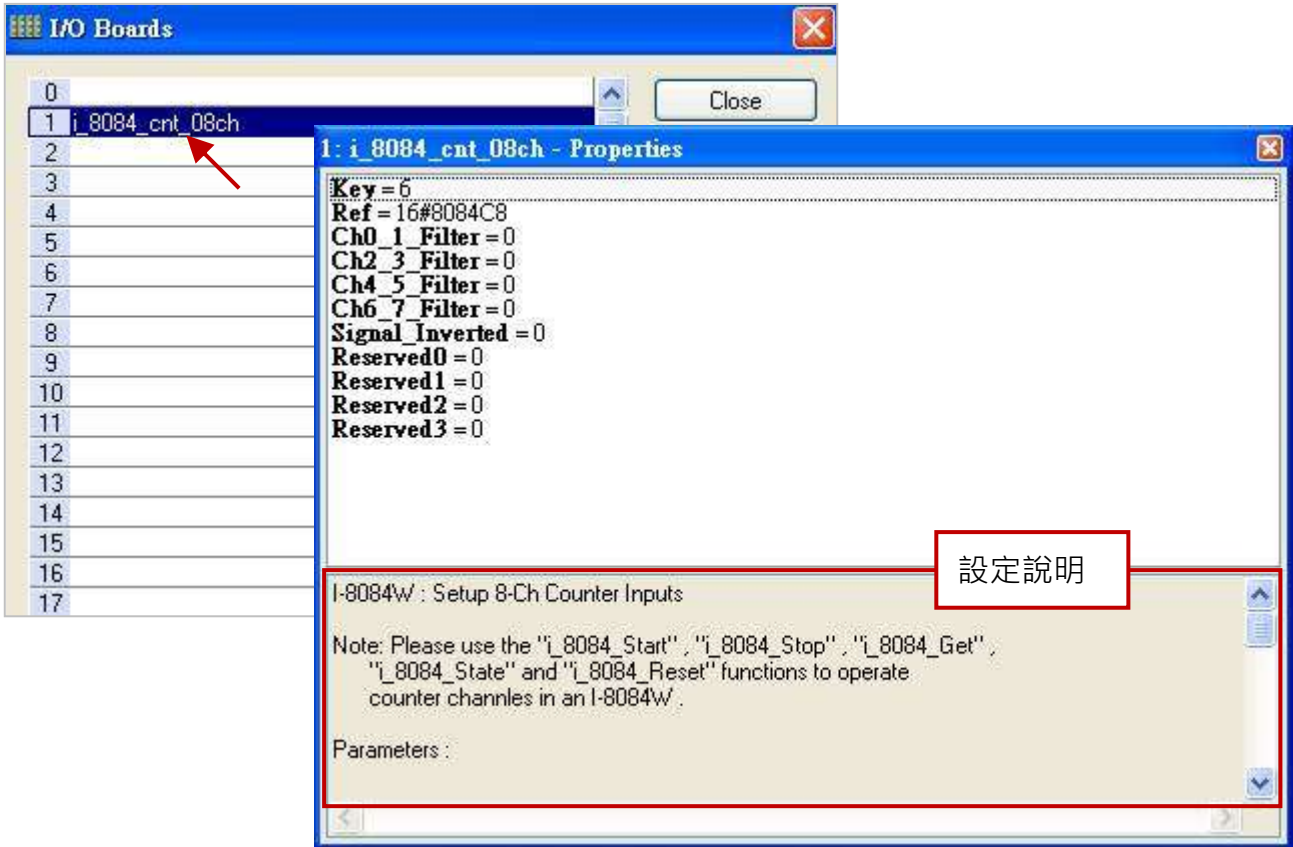


- 4. 完成 "I/O board" 連結後，請參考 [4.9 節](#) 在 "程式區" 使用 "COUNTER_START"、"COUNTER_STOP"、"COUNTER_GET"、"COUNTER_STATE" 與 "COUNTER_RESET" 函式或 ST 語法，來操作 I-8084W 的 Counter 功能。

4.7.3 i_8084_cnt_ch08 (8 通道 UP 計數器)

註: 請使用 Win-GRAF Workbench 中的 "COUNTER_START"、"COUNTER_STOP"、"COUNTER_GET"、"COUNTER_STATE" 與 "COUNTER_RESET" 功能方塊，來操作 I-8084W 的計數通道。

1. 滑鼠雙擊 "i_8084_cnt_ch08" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

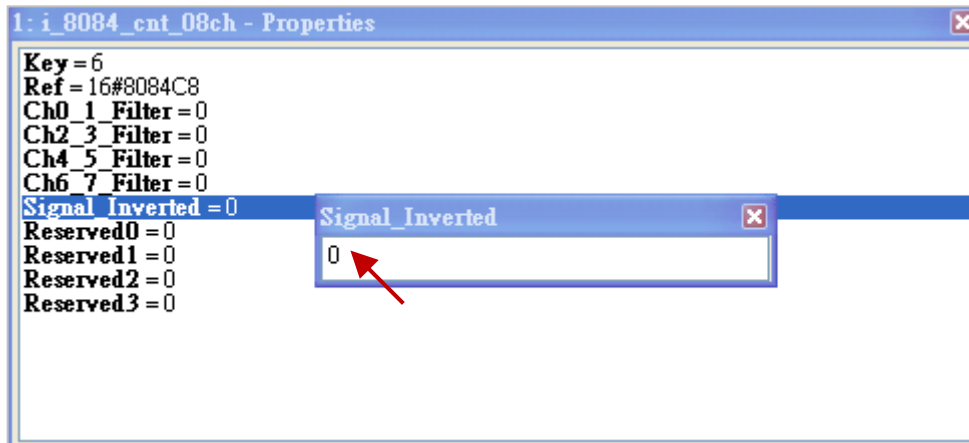
Ch_Filter: 單位為微秒 ($\mu s = 0.000001$ 秒)，可設為 "0" ~ "200"，預設值為 "0" (不過濾)。
"Ch_Filter" 是用來過濾掉一些很小的雜訊寬度。建議設為 "0"，若無雜訊考量或採用即時 (Real-time) 量測。下表為建議的設定:

最大輸入頻率 (Hz)	建議的設定值
1K	200
2K	100
5K	40
10K	20
20K	10
100K	2
450K	1
450K	0 (未過濾)

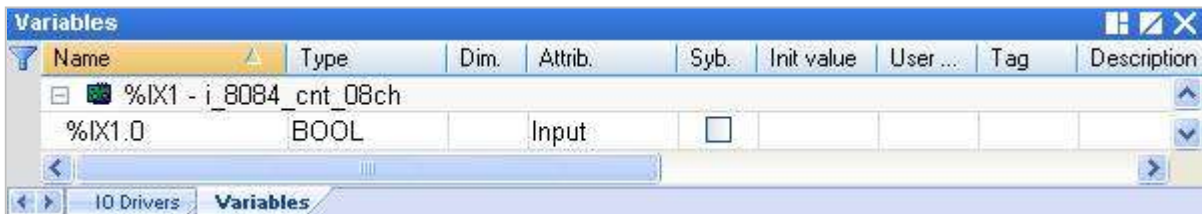
Signal_Inverted:

- 0: 輸入訊號正常 (無反相)。
- 1: 輸入訊號為反相 (表示電壓 HIGH 會視為 LOW 處理，而電壓 LOW 會視為 HIGH 處理)。

2. 滑鼠雙擊所需項目，來輸入數值並按“Enter”鍵完成設定。



3. 在“I/O Boards”視窗內連上“i_8084_cnt_ch08”後，會自動在“Variables”視窗中新增 1 個“BOOL”輸入變數(無作用，固定為“FALSE”)。

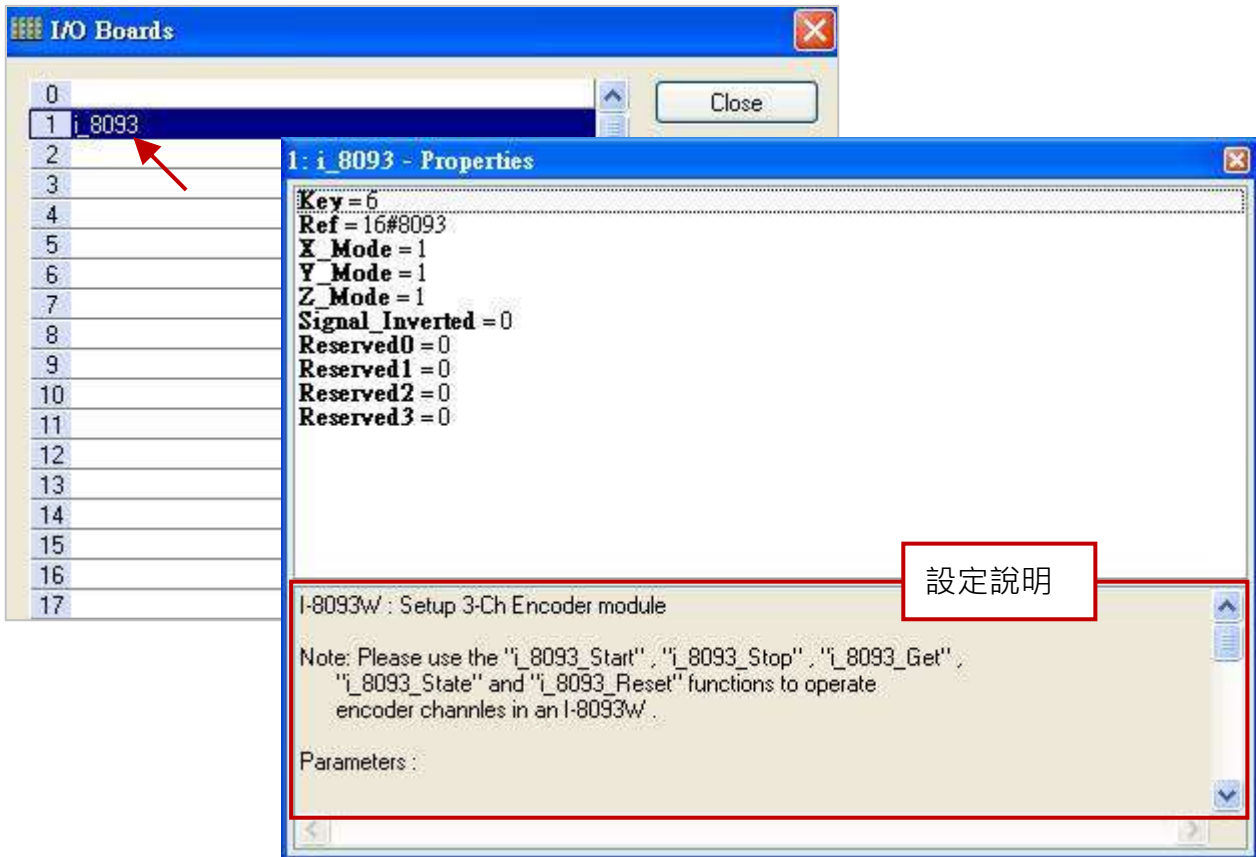


4. 完成“I/O board”連結後，請參考 [4.9 節](#) 在“程式區”使用 "COUNTER_START"、"COUNTER_STOP"、"COUNTER_GET"、"COUNTER_STATE" 與 "COUNTER_RESET" 函式或 ST 語法，來操作 I-8084W 的 Counter 功能。

4.8 i_8093 (3 軸之高速 Encoder 模組)

I-8093W 是一款 3 軸之高速 Encoder 模組，其每一軸都可獨立設定成 Quadrant、Pulse/Direction 或 CW/CCW 之輸入模式。若不熟悉 I/O 卡的開啟/加入方式，可參考 [第四章 \(P4-1\)](#)。

1. 滑鼠雙擊 "i_8093" 開啟 "Properties" 視窗，來查看設定說明。



參數說明:

X_Mode, Y_Mode, Z_Mode:

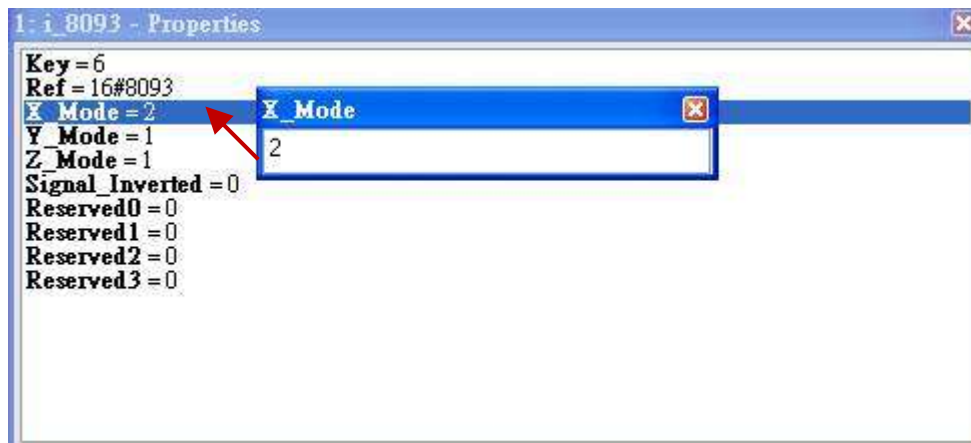
X、Y 與 Z 軸的輸入模式，可設定為 "1"、"2" 與 "3"，若設定為其它值，將會採用 "1"。

- 1: CW/CCW 計數模式。
- 2: Pulse/Direction 計數模式。
- 3: A/B phase (quadrant) 計數模式。

Signal_Inverted:

- 0: 輸入訊號正常 (無反相)。
 - 1: 輸入訊號為反相 (表示電壓 HIGH 會視為 LOW 處理，而電壓 LOW 會視為 HIGH 處理)。
- 例如，若 "Signal_Inverted" 設為 "0" (無反相) 且 "X_Mode" 設為 "2" (Pulse/Direction)，則當 "Dirextion" 訊號為 High 時，計數器將會開始計數。
- 若 "Signal_Inverted" 設為 "1" (反相) 且 "X_Mode" 設為 "2" (Pulse/Direction)，則當 "Dirextion" 訊號為 High 時，計數器將會倒數計數。

2. 滑鼠雙擊所需項目，來輸入數值並按“Enter”鍵完成設定。



3. 在“I/O Boards”視窗內連上“i_8093”後，會自動在“Variables”視窗中新增3個“BOOL”輸入變數可供使用。

Ch0: X 軸的 Z-index

Ch1: Y 軸的 Z-index

Ch2: Z 軸的 Z-index

Name	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
%IX1 - i_8093								
%IX1.0	BOOL		Input	<input type="checkbox"/>				
%IX1.1	BOOL		Input	<input type="checkbox"/>				
%IX1.2	BOOL		Input	<input type="checkbox"/>				

4. 完成“I/O board”連結後，請參考 [4.9 節](#) 在“程式區”使用“COUNTER_START”、“COUNTER_STOP”、“COUNTER_GET”、“COUNTER_STATE”與“COUNTER_RESET”函式或 ST 語法來操作 I-8093W 的 Encoder 功能。

4.9 I-8084W, I-8093W, I-87082W, I-87084W, I-7083 與 I-7080 模組的計數功能

此章節將介紹如何使用 LD 或 ST 語法 來使用 "COUNTER_START"、"COUNTER_STOP"、"COUNTER_GET"、"COUNTER_STATE" 與 "COUNTER_RESET" 函式 (Function)，來操作 Counter/Encoder 功能。若不熟悉程式 與 功能方塊的新增方式，可參考 [2.3.3 節](#)。

注意:

1. 此章節僅針對 PAC 上的 I-8084W 與 I-8093W 模組作說明。
2. 使用下列功能方塊前，請先參考 [4.7.2 節](#) (UP/Down 計數器)、[4.7.3 節](#) (UP 計數器) 與 [4.8 節](#) (Encoder) 來連結 "I/O Boards" 功能。

4.9.1 COUNTER_START (開始計數)

假設: 在 PAC 的 Slot 2 使用 I-8084W 模組，並啟動通道 5 的計數功能。

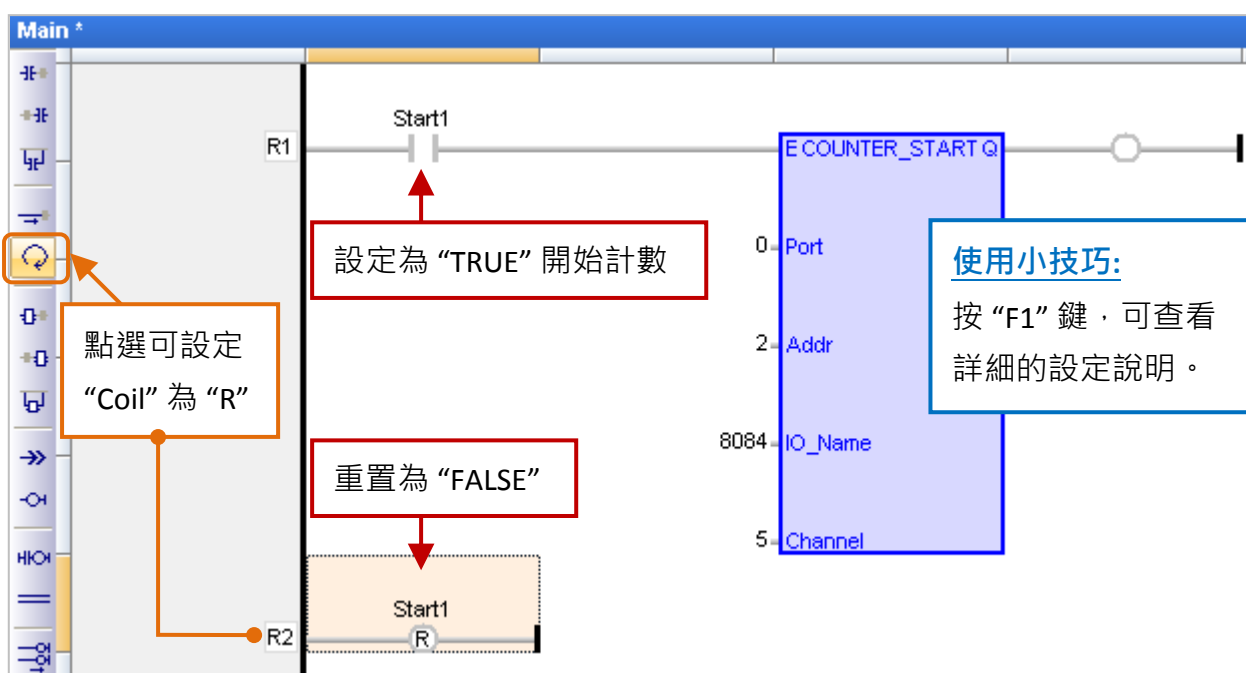
ST 語法:

```
IF Start1 = TRUE THEN
  Start1 := FALSE;
  TMP_BOOL := Counter_Start (0, 2, 8084, 5);
END_IF;
```

註: 可先在 "變數區" 建立一個 "Start1"、"TMP_BOOL", 布林變數。

LD 語法:

(“Start1”: 布林變數，設定為 “TRUE” 開始計數，執行後重置 “Start1” 為 “FALSE”。)



Port: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為 "0"；若是透過 COM Port 連接的 DCON 遠端 I/O 模組，可設定為 "1 ~ 37" (視 PAC 而定，表示 COM1 ~ COM37)。

Addr: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為插槽編號 (0 ~ 7)。

若是透過 COM Port 連接的 DCON 遠端 I/O 模組，請設定為該模組的 Net-ID 位址 (1 ~ 255)。

IO_Name: (資料型態: "DINT")

Counter/Encoder 模組的名稱，可設定為 "8084"、"8093"、"87084"、"87082"、"7083" 與 "7080"。

Channel: (資料型態: "DINT")

Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...，視模組而定。

例如: I-8093W 中 "0" 表示 X 軸; "1" 表示 Y 軸; "2" 表示 Z 軸。

Q: 資料型態為 "BOOL"，"TRUE": 表示 OK；"FALSE": 表示錯誤。

4.9.2 COUNTER_STOP (停止計數)

假設：在 PAC 的 Slot 1 使用 I-8093W 模組，且停用 X 軸的計數功能。

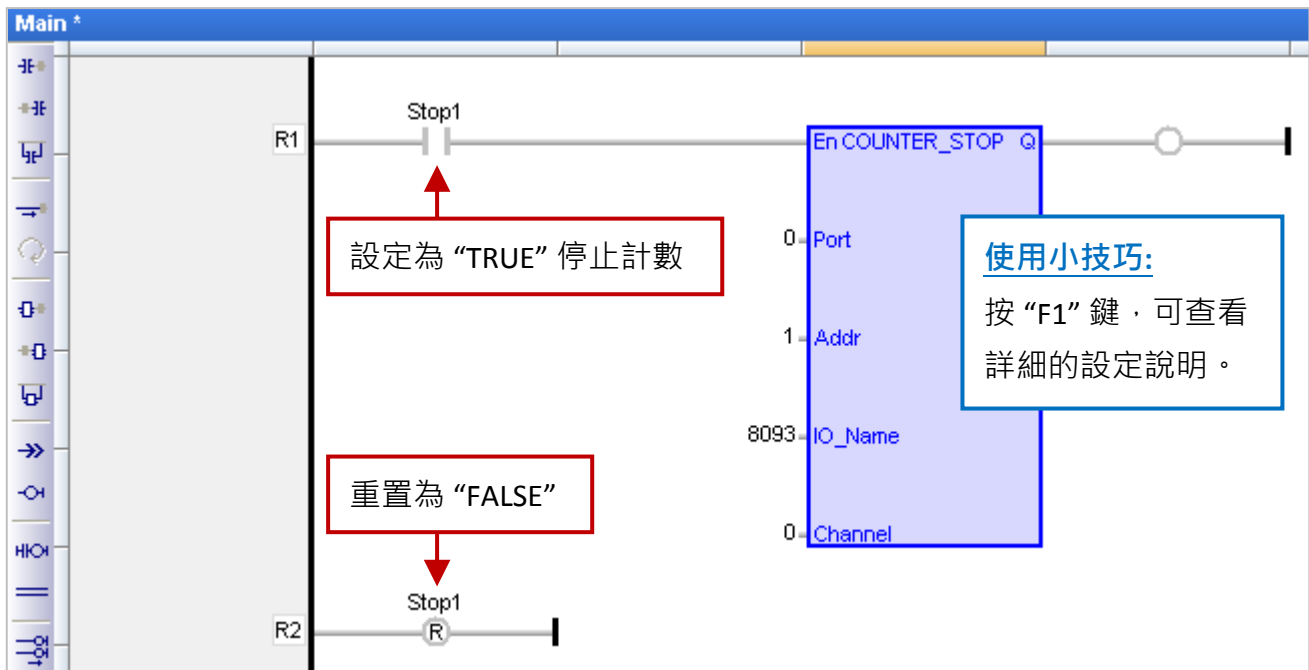
ST 語法:

```
IF Stop1 = TRUE THEN
  Stop1 := FALSE ;
  TMP_BOOL := Counter_Stop (0, 1, 8093, 0) ;
END_IF ;
```

註：可先在 "變數區" 建立一個 "Stop1"、"TMP_BOOL"，布林變數。

LD 語法:

(“Stop1”: 布林變數，設定為 “TRUE” 開始計數，執行後重置 “Stop1” 為 “FALSE”。)



Port: (資料型態: “DINT”)

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為 "0"；若是透過 COM Port 連接的 DCON 遠端 I/O 模組，可設定為 "1 ~ 37" (視 PAC 而定，表示 COM1 ~ COM37)。

Addr: (資料型態: “DINT”)

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為插槽編號 (0 ~ 7)。

若是透過 COM Port 連接的 DCON 遠端 I/O 模組，請設定為該模組的 Net-ID 位址 (1 ~ 255)。

IO_Name: (資料型態: “DINT”)

Counter/Encoder 模組的名稱，可設定為 "8084"、"8093"、"87084"、"87082"、"7083" 與 "7080"。

Channel: (資料型態: “DINT”)

Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...，視模組而定。

例如: I-8093W 中 "0" 表示 X 軸; "1" 表示 Y 軸; "2" 表示 Z 軸。

Q: 資料型態為 “BOOL”，“TRUE”: 表示 OK；“FALSE”: 表示錯誤。

4.9.3 COUNTER_GET (取得計數值)

假設：在 PAC 的 Slot 1 使用 I-8093W 模組，取得 Z 軸的 Encoder 值。

ST 語法:

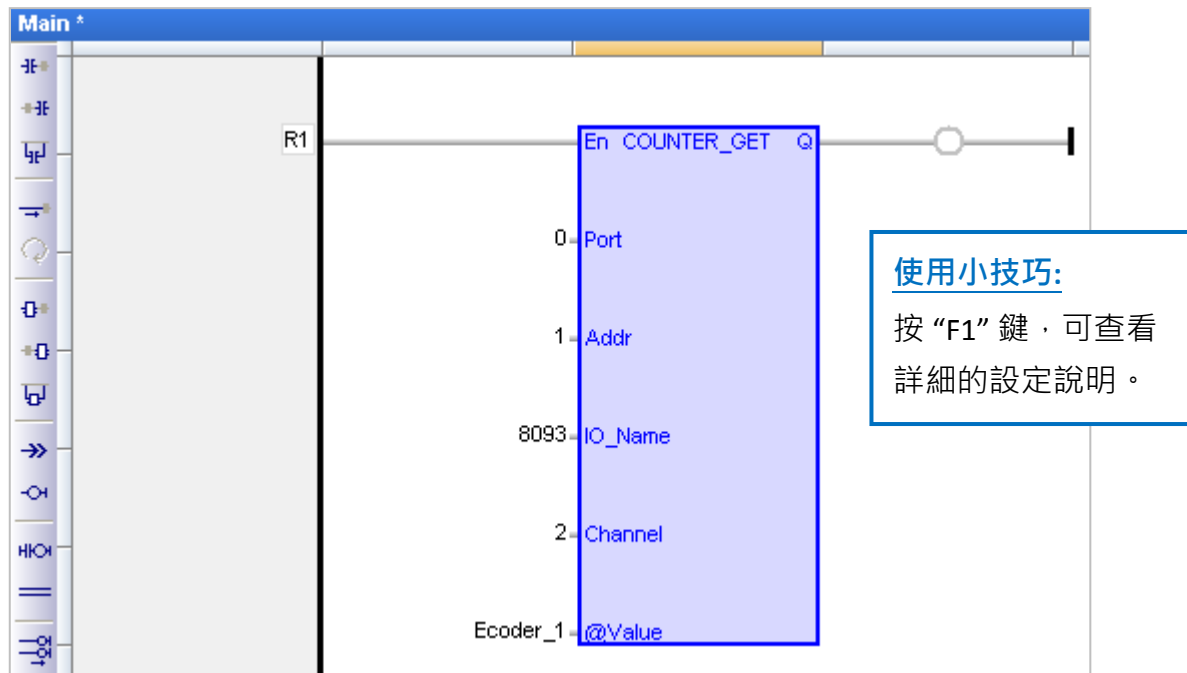
```
TMP_BOOL := Counter_Get (0, 1, 8093, 2, Encoder_1);
```

註：可先在 "變數區" 建立變數。

"TMP_BOOL" (BOOL)

"Encoder_1" (DINT)。

LD 語法:



Port: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為 "0"；若是透過 COM Port 連接的 DCON 遠端 I/O 模組，可設定為 "1 ~ 37" (視 PAC 而定，表示 COM1 ~ COM37)。

Addr: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為插槽編號 (0 ~ 7)。

若是透過 COM Port 連接的 DCON 遠端 I/O 模組，請設定為該模組的 Net-ID 位址 (1 ~ 255)。

IO_Name: (資料型態: "DINT")

Counter/Encoder 模組的名稱，可設定為 "8084"、"8093"、"87084"、"87082"、"7083" 與 "7080"。

Channel: (資料型態: "DINT")

Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...，視模組而定。

例如: I-8093W 中 "0" 表示 X 軸; "1" 表示 Y 軸; "2" 表示 Z 軸。

@Value: (資料型態 可為 "DINT", "UDINT", "DWORD" 與 "LINT")

用來顯示計數值。(關於數值範圍，可參考 [附錄 A](#))

Q: 資料型態為 "BOOL"，"TRUE": 表示 OK；"FALSE": 表示錯誤。

4.9.4 COUNTER_STATE (取得計數狀態)

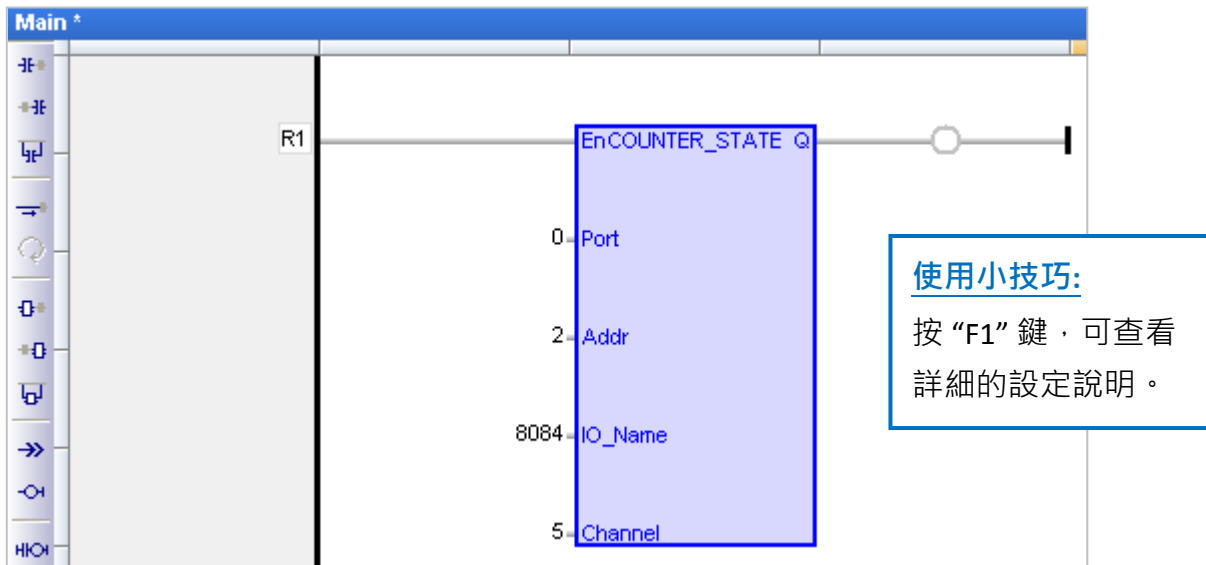
假設：在 PAC 的 Slot 2 使用 I-8084W 模組，並取得通道 5 的計數狀態。

ST 語法:

```
TMP_BOOL := Counter_State (0, 2, 8084, 5);
```

註：可先在 "變數區" 建立一個 "TMP_BOOL" 布林變數。

LD 語法:



Port: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為 "0"；若是透過 COM Port 連接的 DCON 遠端 I/O 模組，可設定為 "1 ~ 37" (視 PAC 而定，表示 COM1 ~ COM37)。

Addr: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為插槽編號 (0 ~ 7)。

若是透過 COM Port 連接的 DCON 遠端 I/O 模組，請設定為該模組的 Net-ID 位址 (1 ~ 255)。

IO_Name: (資料型態: "DINT")

Counter/Encoder 模組的名稱，可設定為 "8084"、"8093"、"87084"、"87082"、"7083" 與 "7080"。

Channel: (資料型態: "DINT")

Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...，視模組而定。

例如: I-8093W 中 "0" 表示 X 軸; "1" 表示 Y 軸; "2" 表示 Z 軸。

Q: 資料型態為 "BOOL"，"TRUE": 表示 OK；"FALSE": 表示錯誤。

4.9.5 COUNTER_RESET (重置計數值)

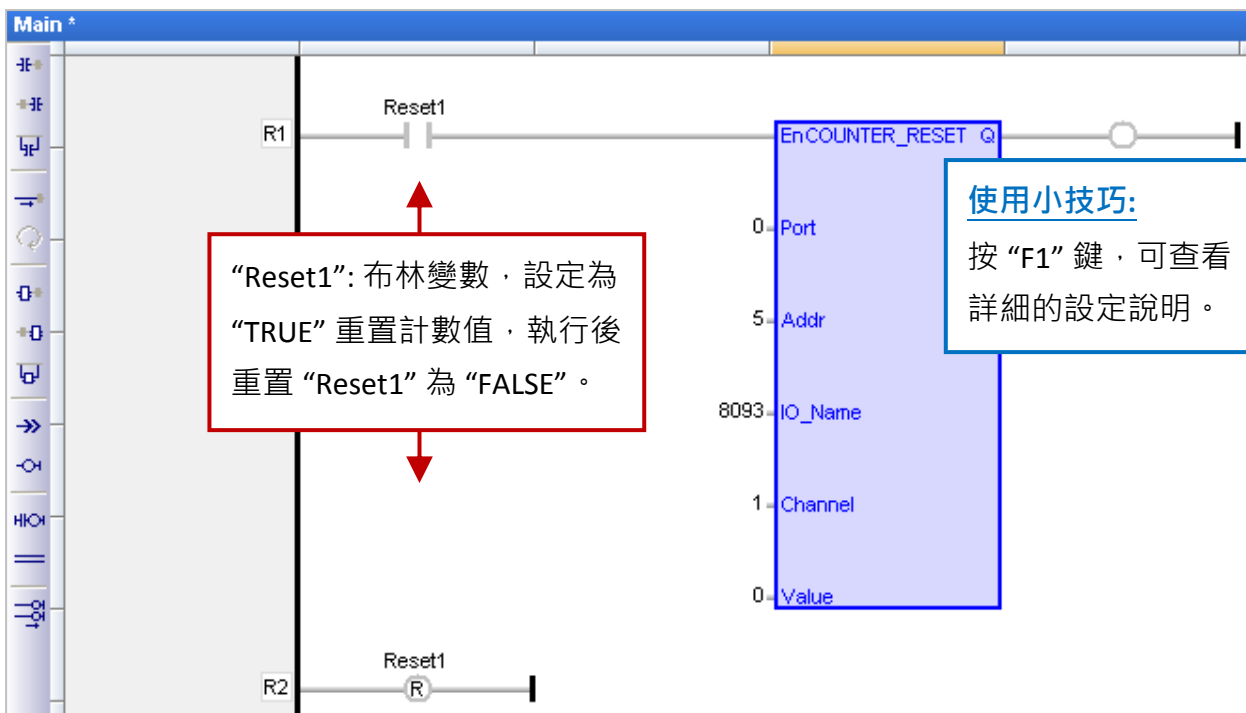
假設：在 PAC 的 Slot 5 使用 I-8093W 模組，並重置 Y 軸的 Encoder 值為 "0"。

ST 語法:

```
IF Reset1 = TRUE THEN  
    Reset1 := FALSE ;  
    TMP_BOOL := Counter_Reset (0, 5, 8093, 1, 0) ;  
END_IF ;
```

註：可先在 "變數區" 建立一個 "Reset1"、"TMP_BOOL", 布林變數。

LD 語法:



Port: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為 "0"；若是透過 COM Port 連接的 DCON 遠端 I/O 模組，可設定為 "1 ~ 37" (視 PAC 而定，表示 COM1 ~ COM37)。

Addr: (資料型態: "DINT")

若是在 PAC 的 Slot 0 ~ 7 使用的 I/O 模組，請設定為插槽編號 (0 ~ 7)。

若是透過 COM Port 連接的 DCON 遠端 I/O 模組，請設定為該模組的 Net-ID 位址 (1 ~ 255)。

IO_Name: (資料型態: "DINT")

Counter/Encoder 模組的名稱，可設定為 "8084"、"8093"、"87084"、"87082"、"7083" 與 "7080"。

Channel: (資料型態: "DINT")

Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...，視模組而定。

例如: I-8093W 中 "0" 表示 X 軸; "1" 表示 Y 軸; "2" 表示 Z 軸。

Value: (資料型態 可為 "DINT", "UDINT", "DWORD" 與 "LINT")

設定欲重置的數值。

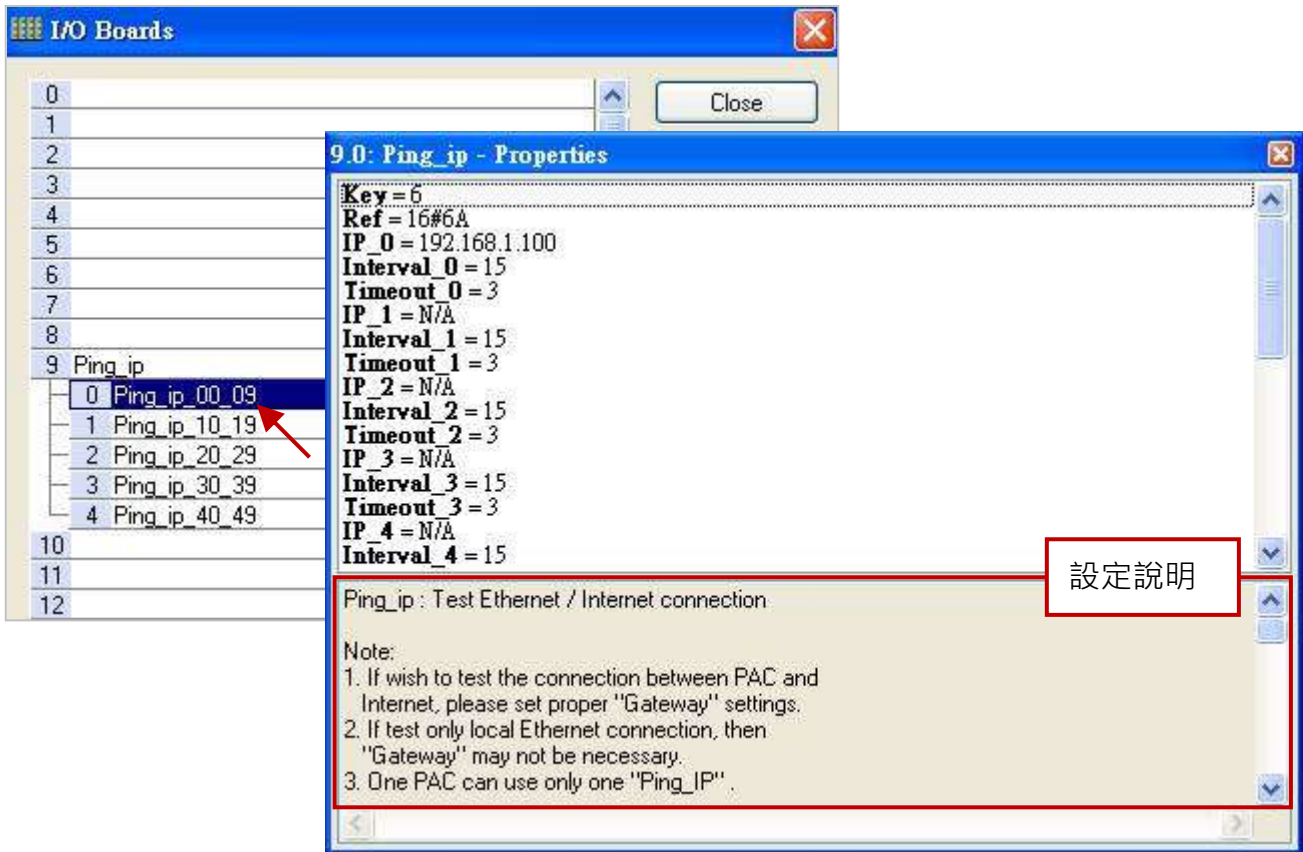
Q: 資料型態為 "BOOL"，"TRUE": 表示 OK；"FALSE": 表示錯誤。

4.10 Ping_ip (測試遠端的 Ethernet/Internet 設備連線)

"Ping_ip" 功能可用來測試遠端的 Ethernet/ Internet 設備連線是否正常? 最多可設定 50 個 IP 位址。若不熟悉 "I/O board" 的開啟/加入方式, 可參考 [第四章 \(P4-1\)](#)。

1. 滑鼠雙擊 "Ping_ip" 開啟 "Properties" 視窗, 來查看設定說明。

注意: Slot0 ~ 7 是保留給 PAC I/O 模組, 請使用在 Slot8 (含) 以上的位置。



註:

1. 如欲測試 PAC 與 Internet 設備之間的連線, 請在 PAC 內設置正確的 "Gateway" 設定。
2. 若僅測試區域網路 (Ethernet) 的連線, 則無需設置 "Gateway"。
3. 一台 PAC 只允許使用 1 個 "ping_ip" 功能 (勿使用 2 個或多個)。
4. 當連線 (Ping) 成功後, 會回傳一個布林值 "TRUE"。
5. 當連線 (Ping) 失敗後, 會再傳一次。若仍失敗, 會回傳一個布林值 "FALSE"。

參數說明:

IP_01 ~ IP_49: (資料型態: "STRING"。)

遠端設備的 IP 位址, 設定為 'N/A' 表示不啟用

例如: 設定為 "192.168.1.100" 或 "52.19.125.242" 或 "N/A"。

Interval_01 ~ Interval_49: (資料型態: "DINT"。)

傳送 "ping" 命令的間隔時間。單位為秒, 預設值為 15 秒, 可設定為 6 ~ 86,400 秒, 若設定值小於 6 會設定為 "6", 若設定值大於 86,400 (24 小時) 則會設定為 "86,400"。

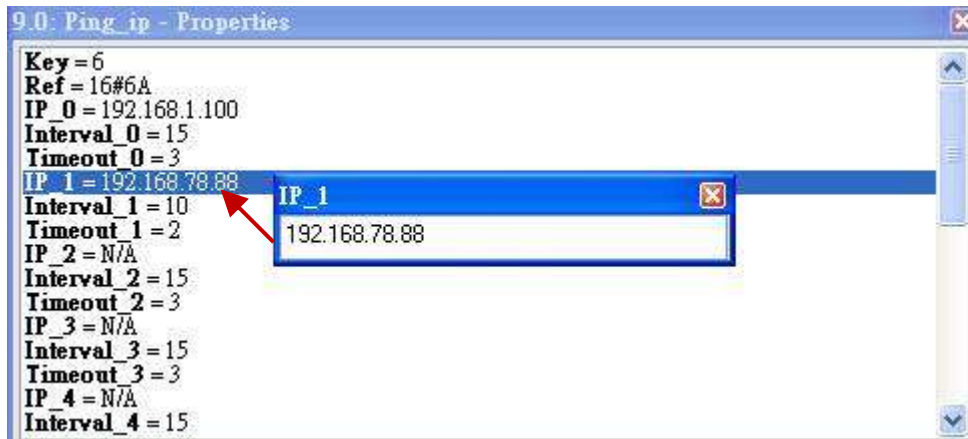
Timeout_01 ~ Timeout_49: (資料型態: "DINT"。)

“ping” 命令的逾時時間。單位為秒，預設值為 3 秒，可設定為 2 ~ 30 秒，若設定值小於 2 會設定為 “2”，若設定值大於 30 則會設定為 “30”。

注意: “Interval_x” 的值至少要設定為 “Timeout_x” 的三倍，否則 PAC 會採用三倍的值。

例如: “Timeout_00” 設定為 “10”，“Interval_00” 設定為 “20”，則 PAC 會採用的 “Interval_00” 值為 “30” (即， $10 \times 3 = 30$)。

2. 滑鼠雙擊所需項目，來輸入數值並按 “Enter” 鍵完成設定。



3. 在 “I/O Boards” 視窗內連上 “Ping_ip” 後，會自動在 “Variables” 視窗中新增 50 個布林變數，當 Win-GRAF 有連上 PAC 時，會顯示出連線的狀態。

True: 表示連線 OK。

FALSE: 表示連線失敗 或 接線問題。



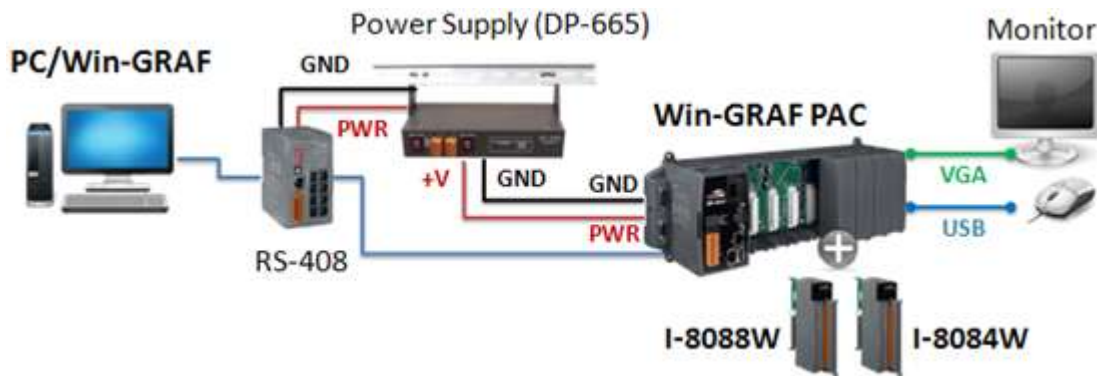
4.11 I-8088W (8 通道 PWM 輸出模組)

I-8088W 是一款 8 通道脈寬調變 (PWM) 輸出模組。它的訊號輸出功率 ($Duty = High / (High + Low)$) 可以是 0.1% ~ 99.9%，它的訊號輸出頻率 (Frequency) 可以設成 1 Hz ~ 500 KHz。另外，I-8088W 支援 2 種 PWM 輸出模式，一種是連續輸出模式 (Continuous mode)，另一種是突發模式 (Burst mode)。設定為突發模式時，它只會輸出所指定的訊號波數量，然後就休息不輸出。設定為連續輸出模式，就會連續一直輸出。更多關於 I-8088W 的規格說明，請參訪網址：

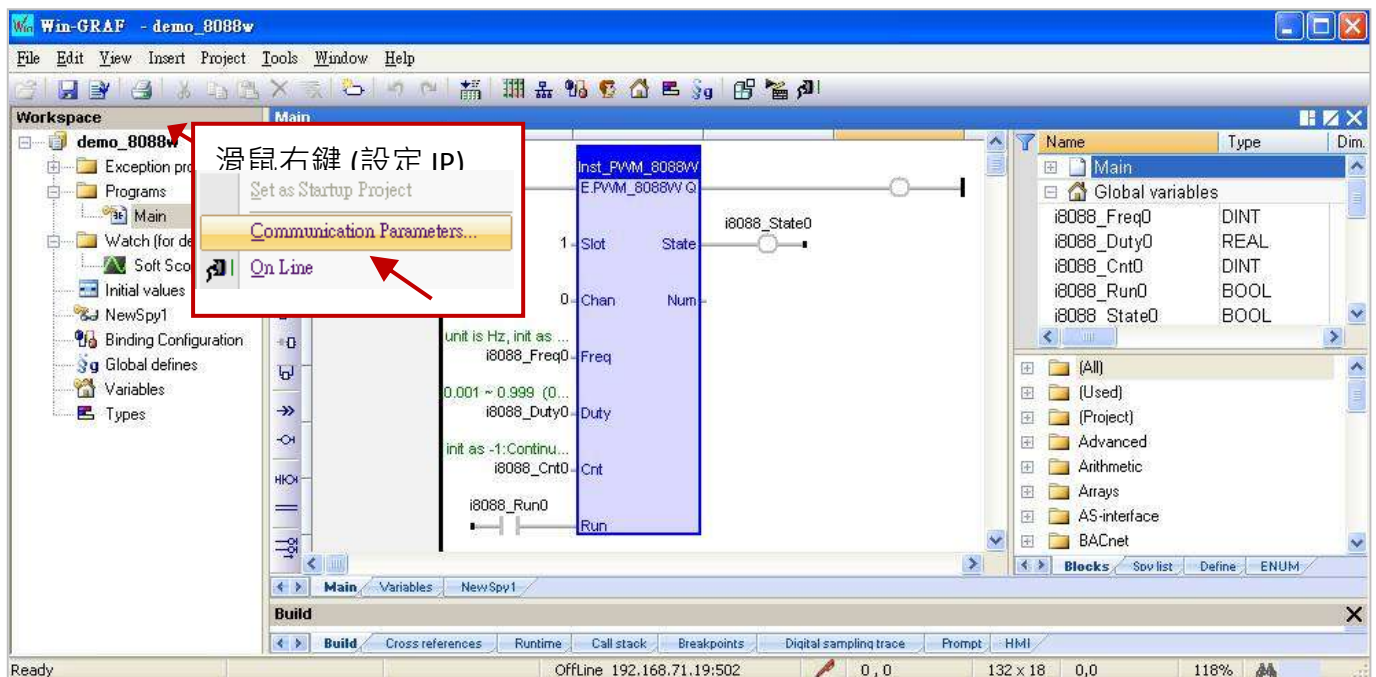
http://www.icpdas.com/products/Remote IO/i-8ke/i-8088w_c.htm。

硬體連接圖：

本範例是使用 I-8084W (Slot 2) 來量測 I-8088W (Slot 1) 的 PWM 訊號的頻率 (實際的 PWM 應用應該不需使用 I-8084W)，請將 I-8088W 的 PWM 輸出 (通道 0，PW0) 接到 I-8084W 的頻率輸入 (通道 0，COA+)。

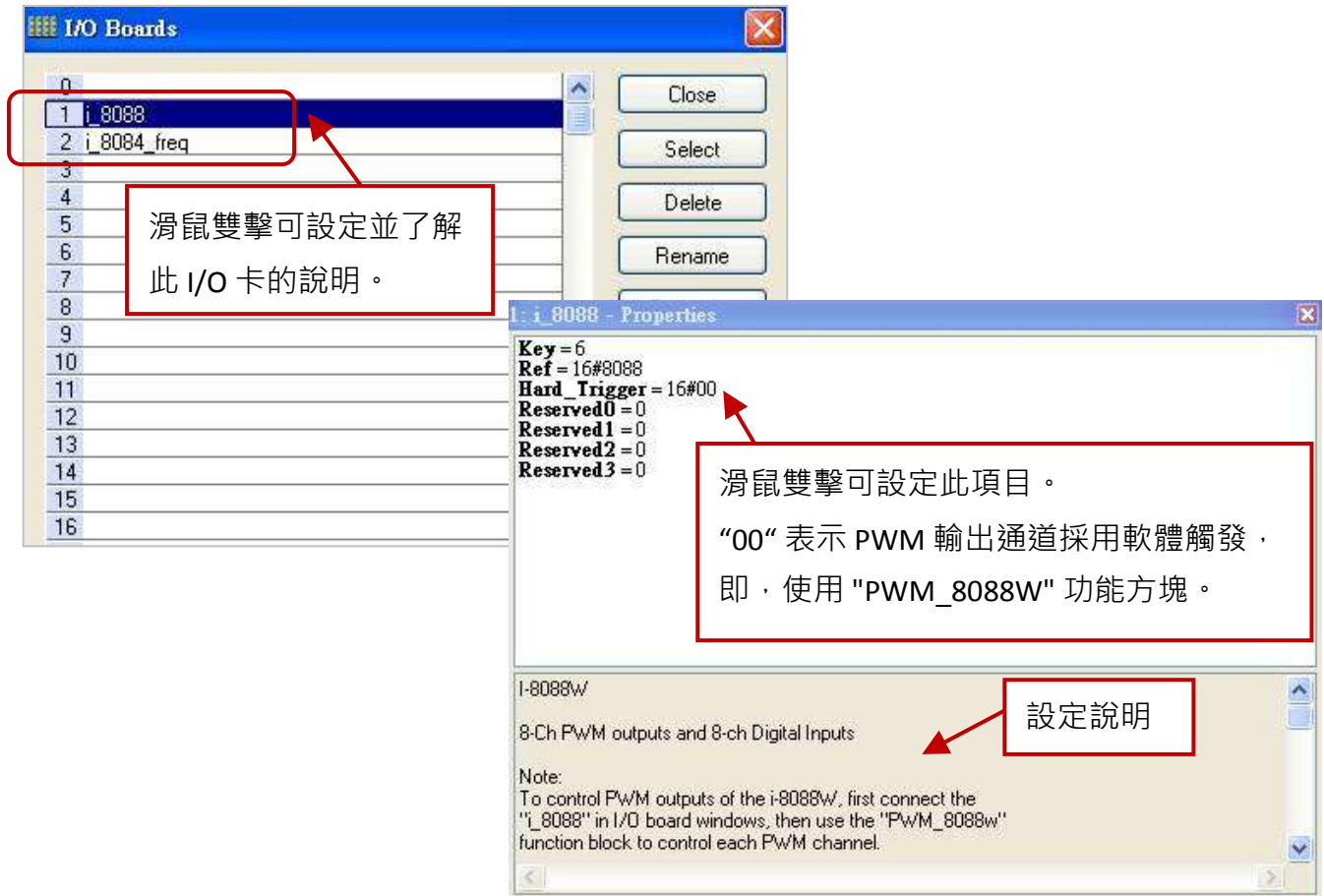


在出貨光碟中 (\\Napdos\Win-GRAF\demo-project) 提供了以下將說明的範例程式，請參考 [第 12 章](#) 來回存/開啟此專案 (demo_8088w.zip) 並設定好 PAC 目前的 IP 位址。



I/O Boards:

本範例在 "I/O Boards" 視窗內，加入 "i_8088" 與 "i_8084_freq" 於對應的 I/O 插槽編號上 (可參考 [第四章](#))。滑鼠雙擊 Slot 編號可開啟 "Properties" 視窗，您可參考下方的使用說明來設定此 I/O 卡。

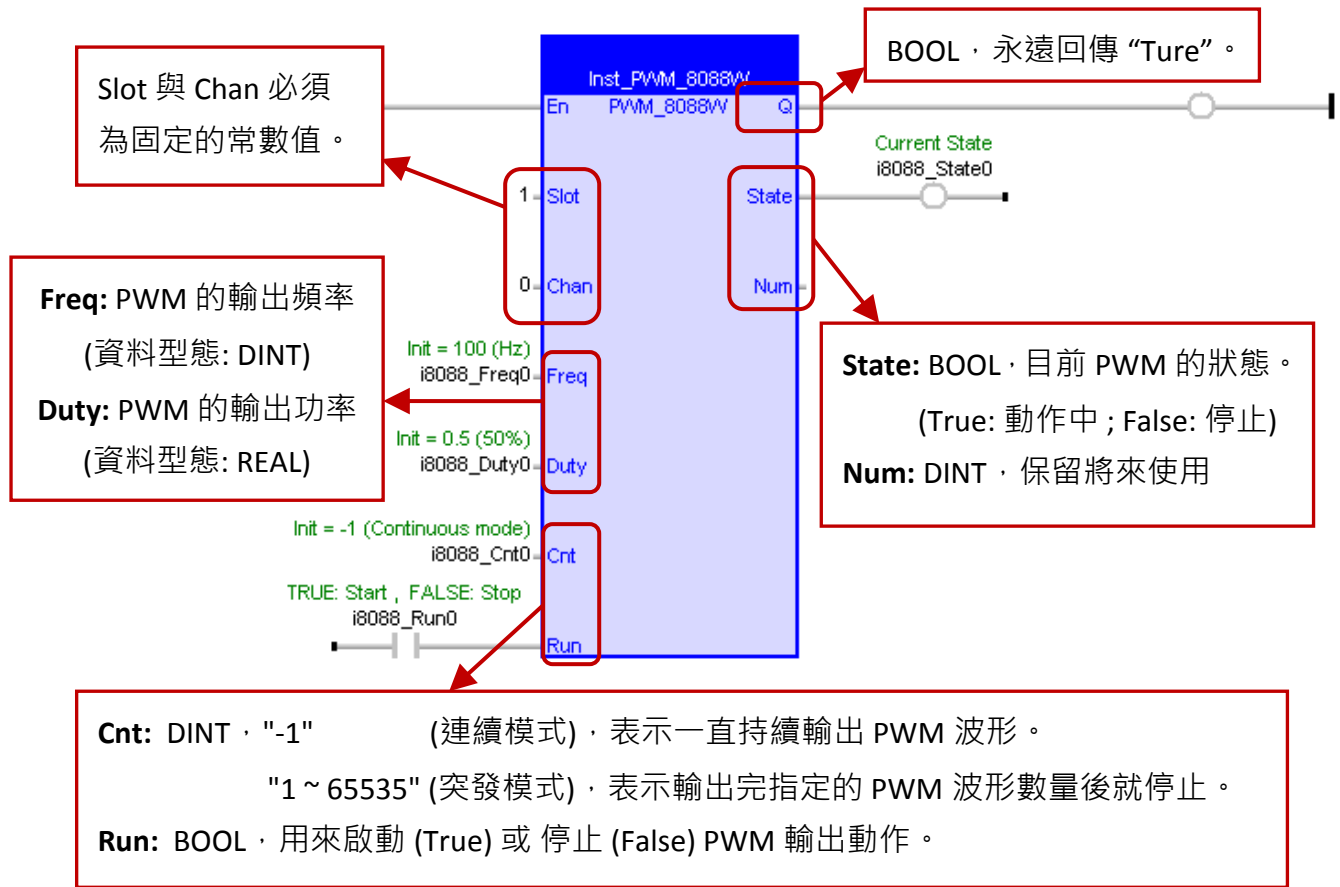


連上 "i_8088" 與 "i_8084_freq" I/O 卡後，會自動在 "Variables" 視窗 (或變數區) 新增相關的變數。另外，也可在此處新增程式中需使用的變數 (可參考 [2.3.1 節](#))。



"PWM_8088W" 功能方塊:

接著，編寫類似下方的 階梯圖 (LD) 程式來控制 I-8088W 個別通道的 PWM 輸出。



參數說明:

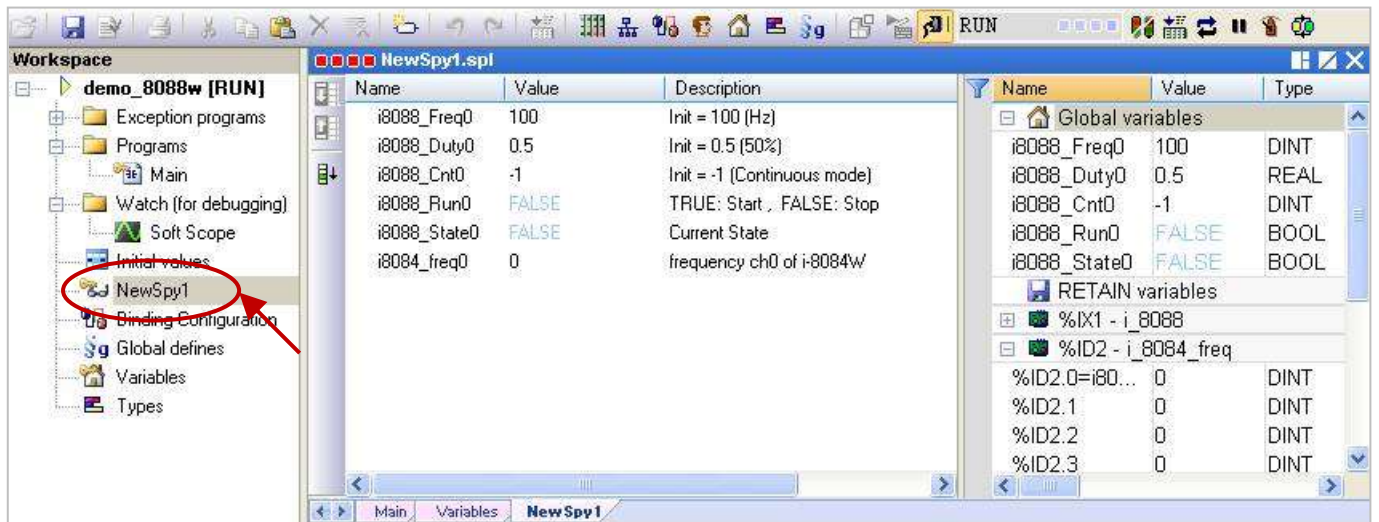
- Slot:** 使用的 I/O 插槽編號，必須為固定的常數值 (資料型態: DINT)，此例為 "1"。
- Chan:** 使用的 I/O 通道編號，必須為固定的常數值 (資料型態: DINT)，此例為 "0"。
- Freq:** PWM 的輸出頻率 (資料型態: DINT，單位: Hz)，值可以是 1 Hz ~ 500 KHz。此例設定初值 (Init value) 為 "100" Hz。
- Duty:** PWM 的輸出功率 (資料型態: REAL)，值可以是 0.001 ~ 0.999 (即 0.1 % ~ 99.9 %)。此例設定初值 (Init value) 為 0.5 (即 50 %)。
- Cnt:** 輸出模式 (資料型態: DINT)
 - 連續模式: 設定為 "-1" (此例的初值)，一直持續輸出 PWM 波形。
 - 突發模式: 可設定 "1 ~ 65535"，輸出完指定的 PWM 波形數量後就停止。
- Run:** 使用一個 BOOL 變數來控制啟動 PWM 輸出。(True: 啟動 ; False: 停止)
- State:** 目前 PWM 的狀態 (資料型態: BOOL)。(True: 動作中 ; False: 停止)

測試程式:

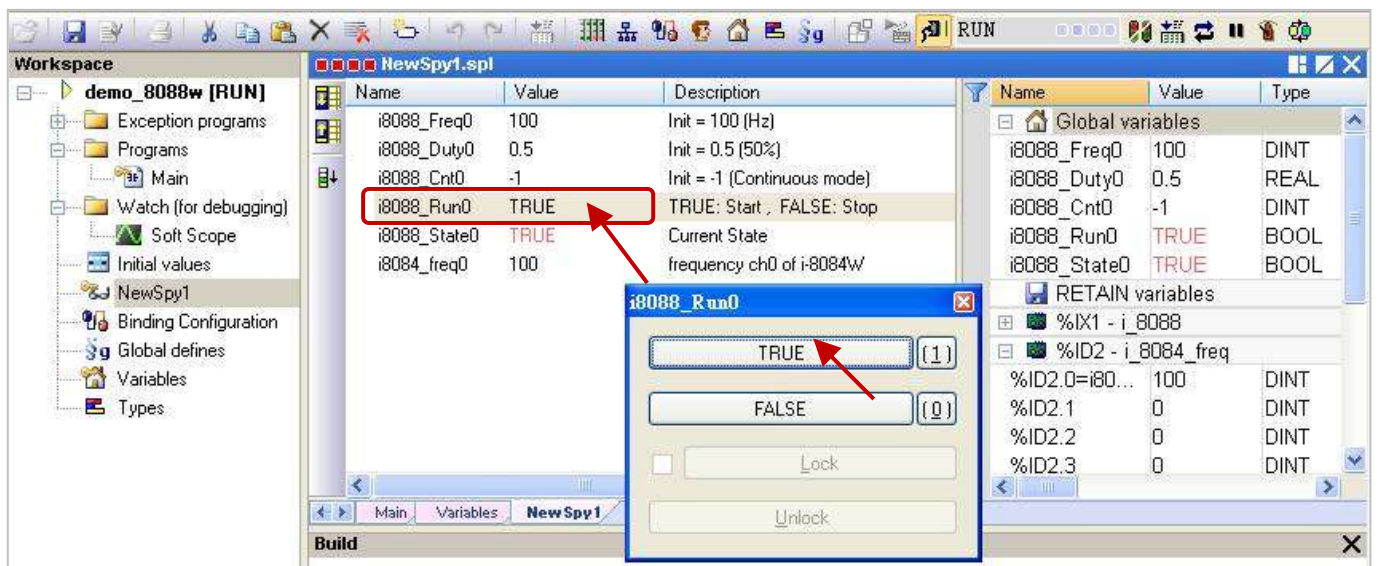
測試之前，請確認已設定好 PAC IP，再編譯/下載程式到 PAC 中。

(若不熟悉操作，可參考 [2.3.4 節](#) 與 [2.3.5 節](#))

與 PAC 連線時，在觀測清單中 (Spy list，設定方式可參考 [11.3 節](#)) 可見到 I-8088W 的 PWM 輸出頻率 ("i8088_Freq0") 為 100 Hz，輸出功率 ("i8088_Duty0") 為 0.5，採用連續輸出模式 ("i8088_Cnt0" = -1)，且目前 I-8084W 量測到的頻率 ("i8084_Freq0") 為 0 Hz。



請將 "i8088_Run0" 設定為 "TRUE" 來啟動 PWM 輸出，此時 "i8088_State" 也會由 "FALSE" 變為 "TRUE" 並將 PWM 訊號輸出給 I-8084W，"i8084_Freq0" 的量測值將由 0 Hz 變為 100 Hz。



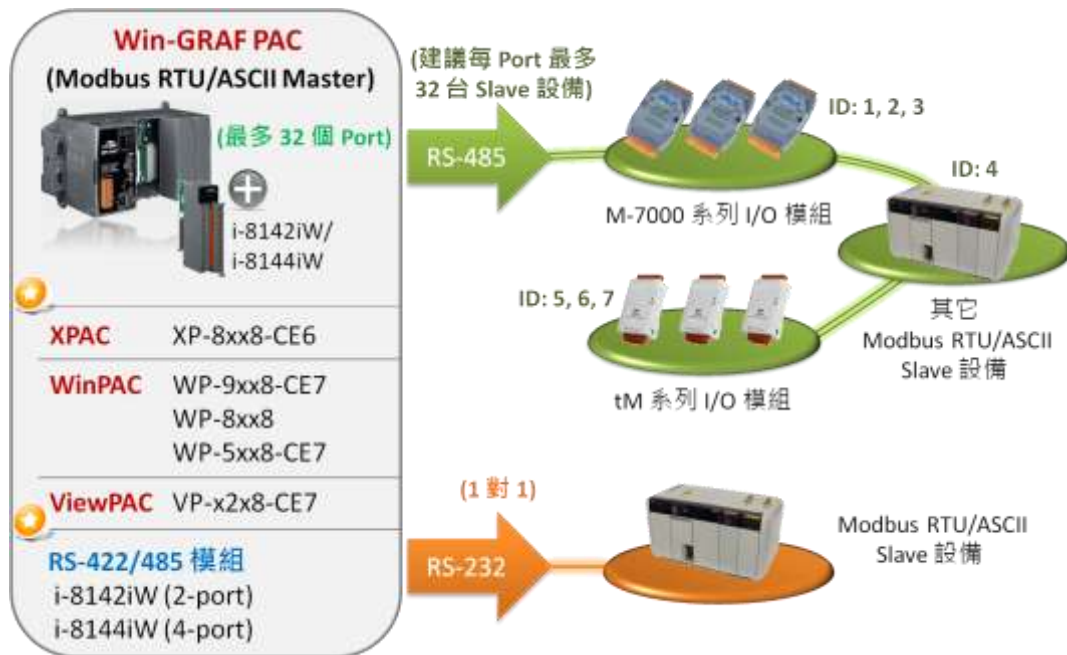
若將 "i8088_Cnt0" 設定為 "500" (Burst 模式)，則輸出完 500 個 PWM 波形後 "i8084_Freq0" 的量測值將變為 "0"，您可試著修改不同的設定值，再將 "i8088_Run0" 設定為 "TRUE" 來觀察輸出的變化。

第 5 章 Modbus Master: 連接其它 Modbus Slave 設備

此章節將介紹如何啟用 Win-GRAF PAC 為 Modbus Master 來連接 Modbus RTU/ASCII Slave 或 Modbus TCP/UDP Slave 設備。如需在 WP-5xx8-CE7 使用 XV Board，請參考 [5.1.6 節](#) ~ [5.1.12 節](#)。

5.1 啟用 Win-GRAF PAC 為 Modbus RTU/ASCII Master (I/O & XV-board)

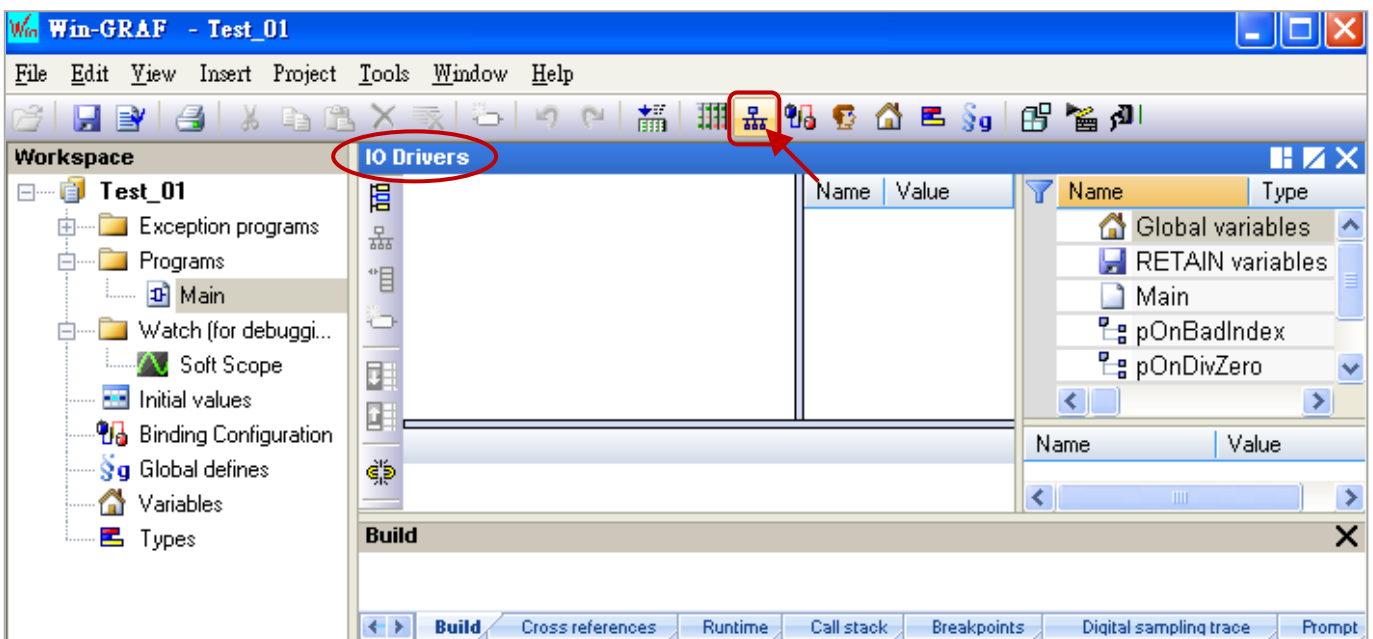
應用示意圖:



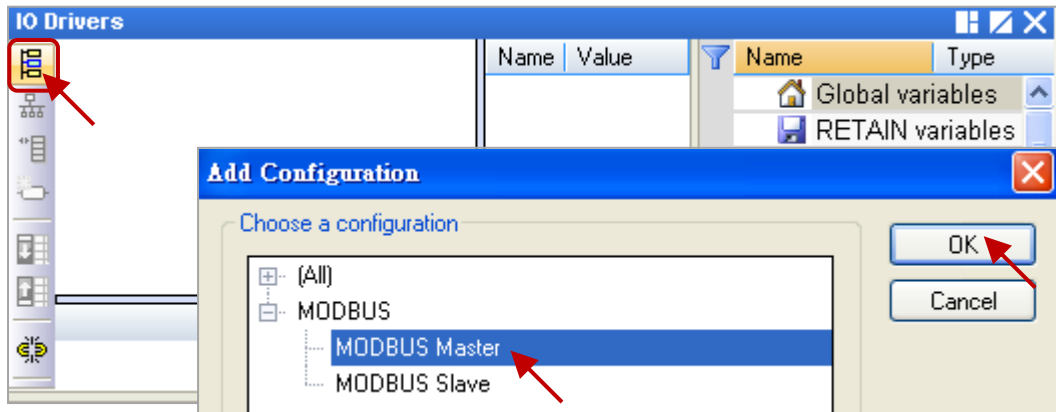
(您可參考 [P1-1](#)，來查詢詳細的 PAC 型號)

請參考以下操作步驟:

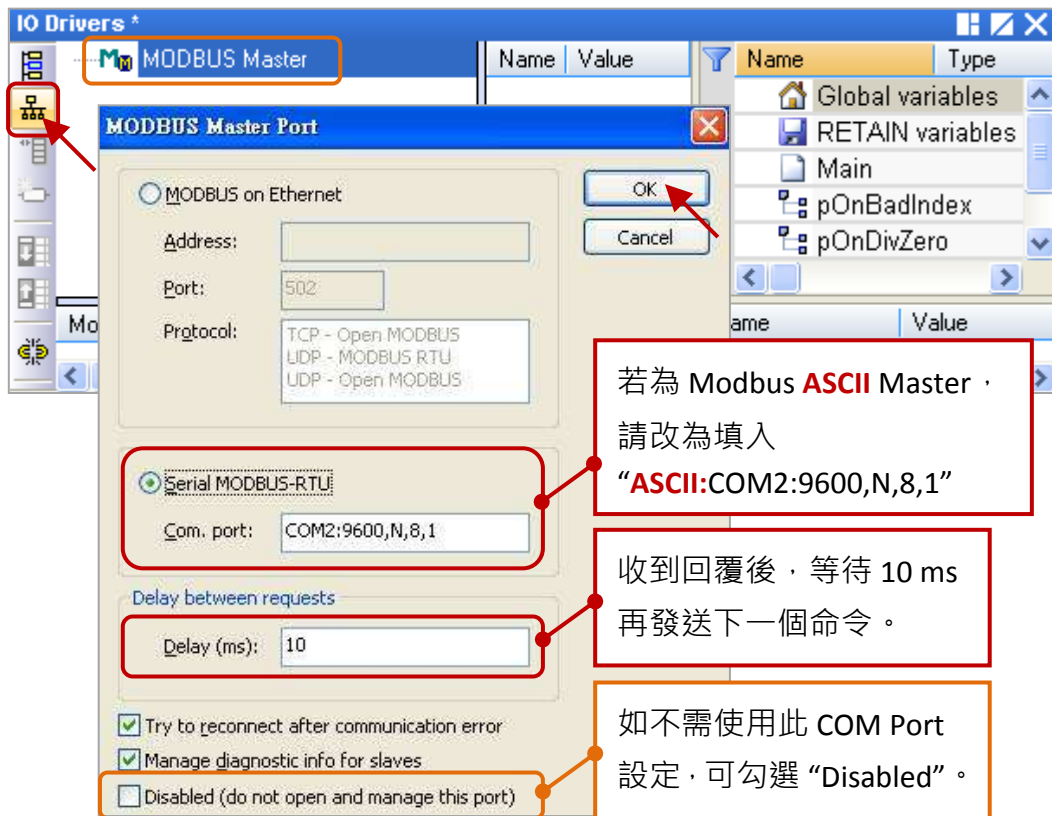
1. 滑鼠點選工具列上的“Open Fieldbus Configuration” 按鈕來開啟 “I/O Drivers” 視窗。



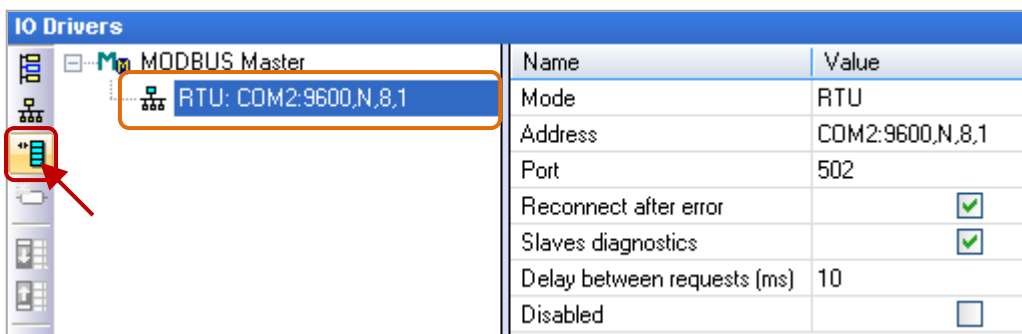
2. 點選 “I/O Drivers” 視窗左側的 “Insert Configuration” 按鈕，再點選 “MODBUS Master” 並點選 “OK” 來啟用一個 Modbus Master 設定。



3. 點選左側的 “Insert Master/Port” 按鈕，開啟設定視窗。點選 “Serial MODBUS-RTU” 並設定 COM Port (例如: “COM2:9600,N,8,1”) 與 Delay (建議值: 10 ms，可設為 0 ~ 10000)，再點選 “OK”。



4. 點選左側的 “Insert Slave/Data Block” 按鈕，來建立一個 Data Block。



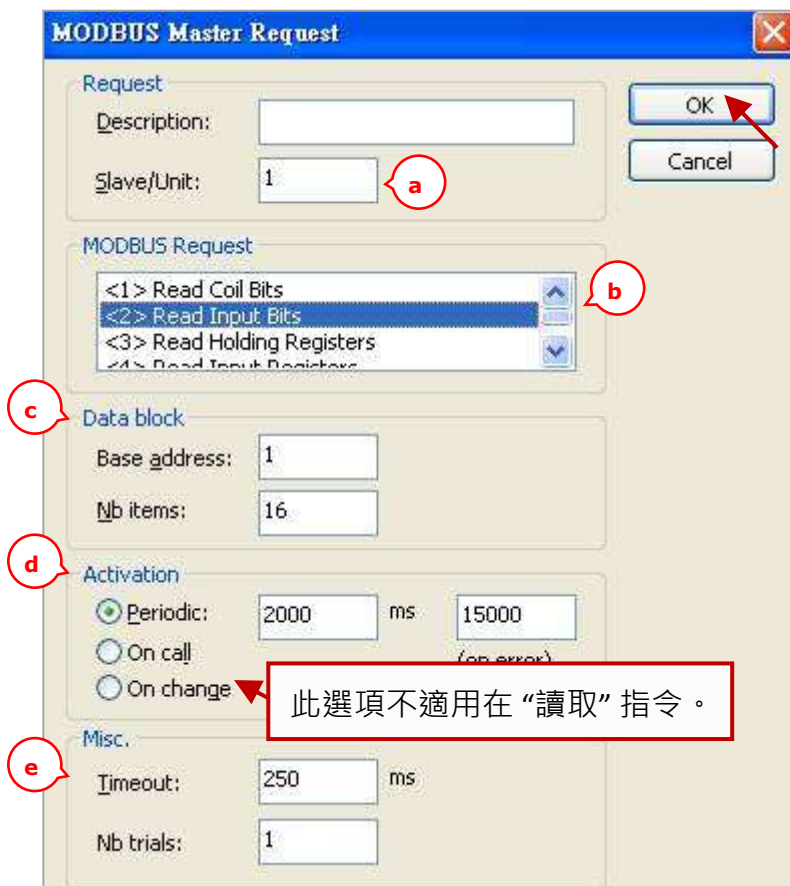
我們將介紹 5 個 Data Block，而每一個 Data Block 都代表一個 Modbus Master Request。

項目	Function Code	Modbus Request	說明
1	2	Read Input Bits	讀取 DI 資料
2	5	Write single coil bit	寫出 DO 資料
3	4	Read Input Registers	讀取 AI 資料
4	6	Write single holding register	寫出 AO 資料 (16-bit)
5	16	Write Holding Registers	寫出 AO 資料 (16/32 bit)

註: 若想在程式運行過程中停用 (Disable) Modbus RTU/ASCII Master Port，請參考 [第 5.1.13 節](#) 來使用 "MBRTU_M_disable" 函式。

5.1.1 讀取 DI 資料

1. 於 "MODBUS Master Request" 設定視窗中，設定以下項目並於完成後按 "OK"。



a. Slave/Unit:

填入 Slave 設備的站號 (Net-ID，此例為 "1")。

b. MODBUS Request: 此例，

選擇 "<2> Read Input Bits" 選項。

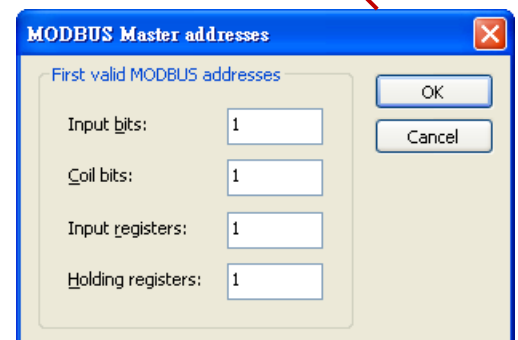
c. Base address:

預設值由 1 開始。

Nb items:

讀取 DI 的數量 (此例為 16)。

註: 如需修改 "Base address"，可使用滑鼠右鍵點選 "MODBUS Master" 再選擇 "MODBUS Master Addresses" 修改其值。



d. Activation: 表示 Modbus Request 發送的方式。

Periodic: 表示週期性的發送，此例為每 2 秒發送一次。"on error" 表示每當發生異常時，下一次的發送時間 (此例為 15 秒)。

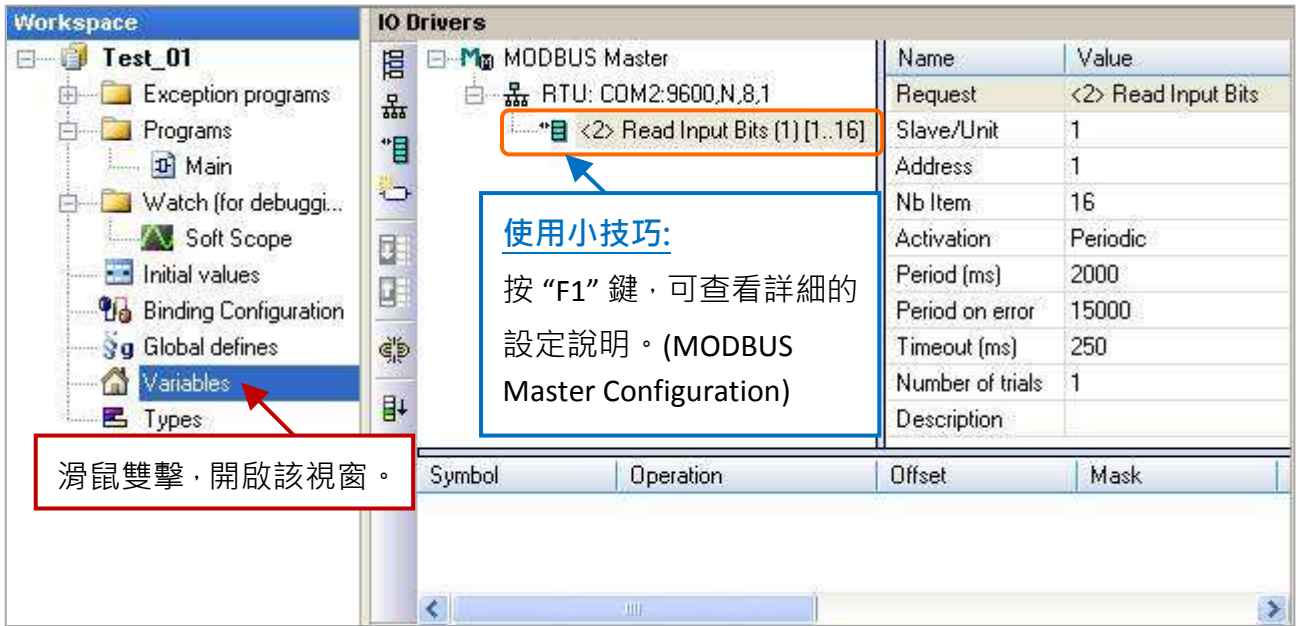
On call: 表示程式有呼叫時，才進行發送一次。

On change: 表示寫出的資料有改變時，才進行發送一次。

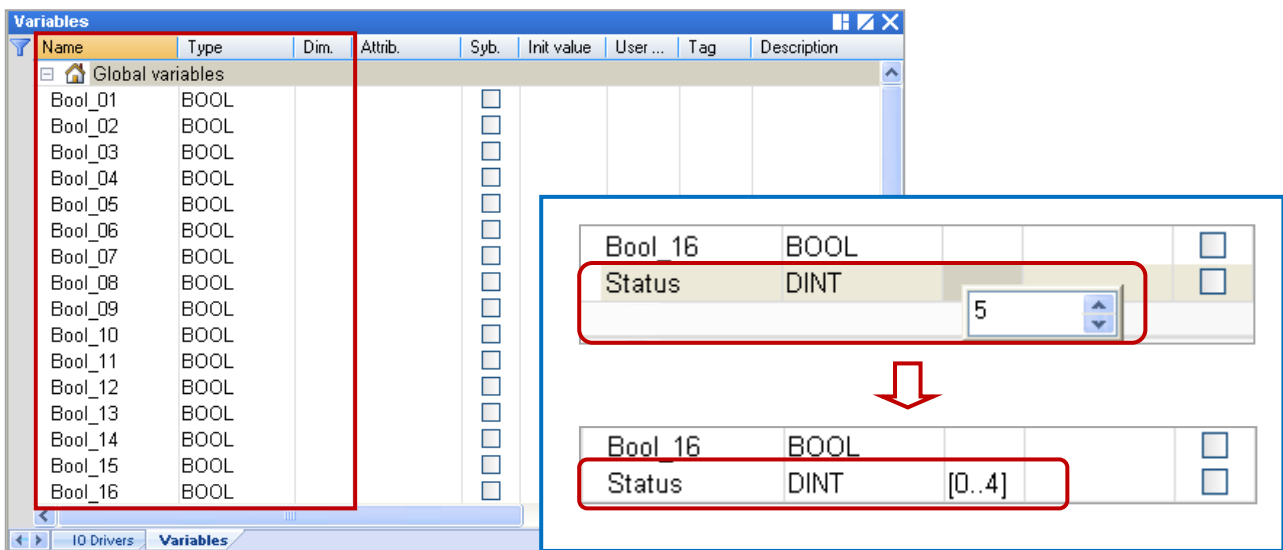
e. Timeout: 設定多久未回應，即表示異常。

(對於 Modbus RTU/ASCII 建議值: 200 ~ 1000 ms；此例為 250 ms)

2. 接著，請開啟 “Variables” 視窗，設定需使用的變數。

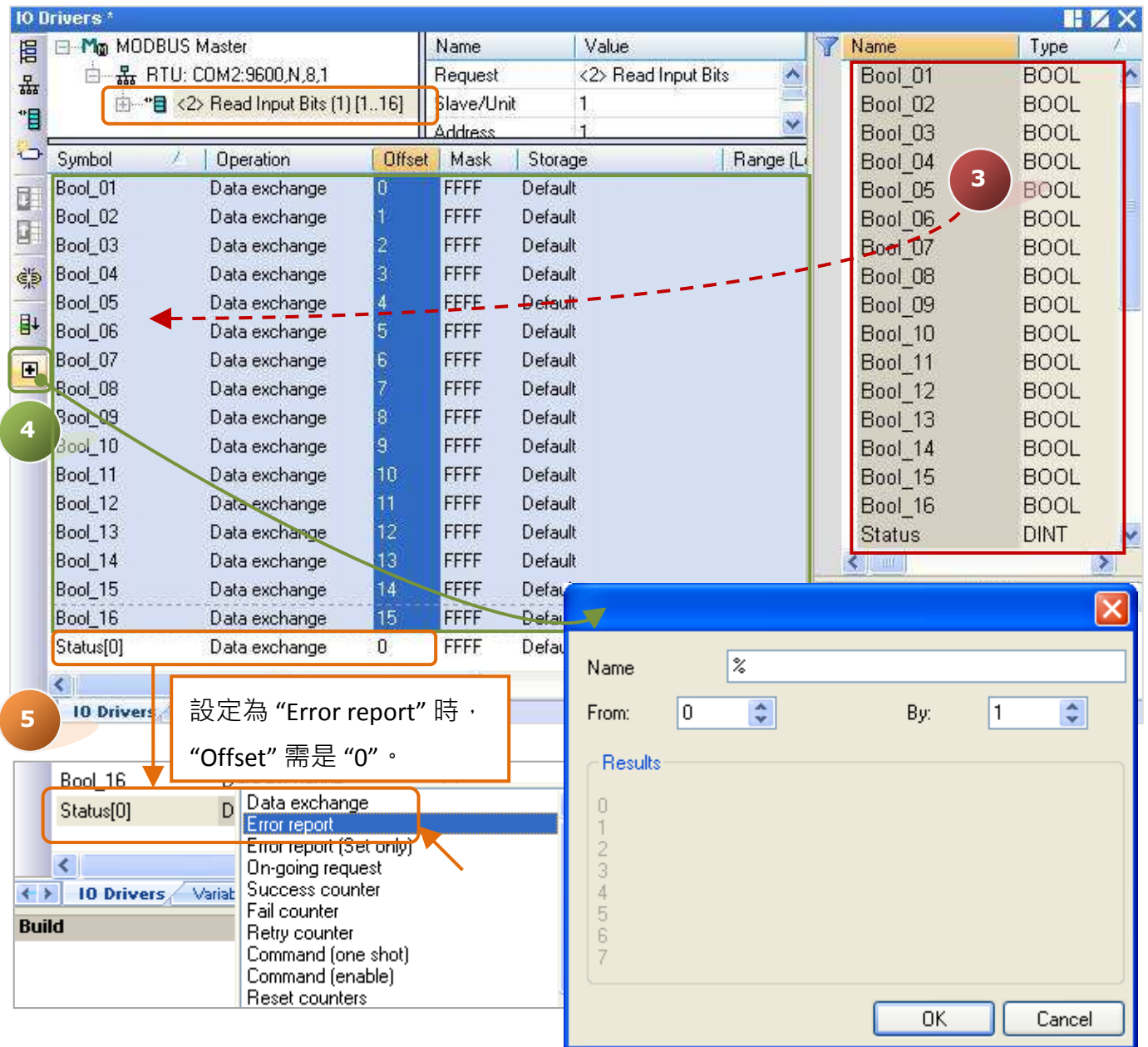


設定 16 個變數 (Name: “Boo_01 ~ Boo_16”; Type: BOOL) 用來讀取資料，與 1 個陣列變數 (Name: “Status”; Dim.: 5; Type: DINT) 用來記錄資料的存取狀況。設定方式可參考 [2.3.1 節](#)，設定完成後，畫面如下。



3. 如下圖，於 “I/O Drivers” 視窗，請將變數區中的變數 (“Boo_01” ~ “Boo_16” 與 “Status”) 拖曳到第 1 個 Data Block 的 “Symbol” 區域。**注意:** “Status” 是一個陣列變數，拖曳到 “Symbol” 區域會是 “Status[0] ~ [4]”，請按 “Delete” 鍵來刪除 “Status[1] ~ [4]”。
4. 接著，選取 “Boo_01 ~ Boo_16” 的 “Offset” 欄位，並點選左側的 “Iterate Property” 按鈕，再設定 “Offset” 值 (From: 0 ; By: 1，可參考 [3.1 節](#) – 步驟 8)。

- 設定 “Status[0]” 的 “Operation” 為 “Error report” (表示讀取失敗時，該變數值為一個 “Error Code”，讀取成功時則會重置為 “0”)。

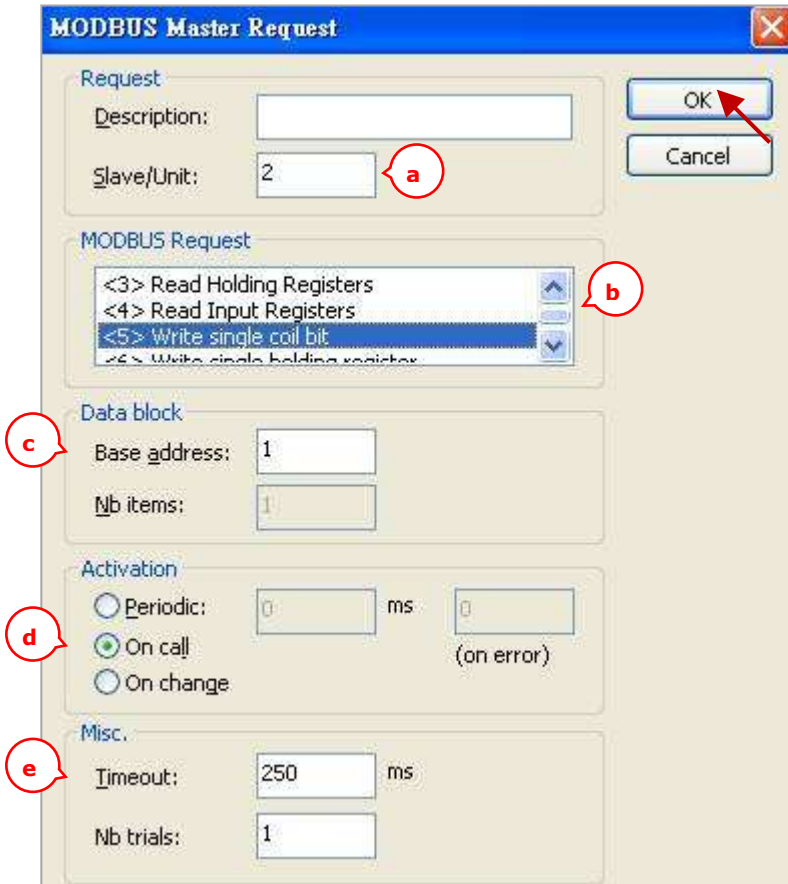


您可在此 “I/O Drivers” 視窗中，按 “F1” 鍵並查看詳細的 Modbus Master 設定說明。

Error Code	說明	Error Code	說明
0	OK (通訊正常)。	8	資料同位元檢查錯誤 (Data Parity Error)。
1	不支援 MODBUS 功能。	10	無效的閘道器 (Gateway) 路徑。
2	無效的 MODBUS 位址。	11	閘道器 (Gateway) 目標異常。
3	無效的 MODBUS 數值。	128	通訊異常 (Timeout)。
4	MODBUS Server 異常。	129	Bad CRC16。
6	Server 忙碌中。	130	RS232 通訊錯誤。

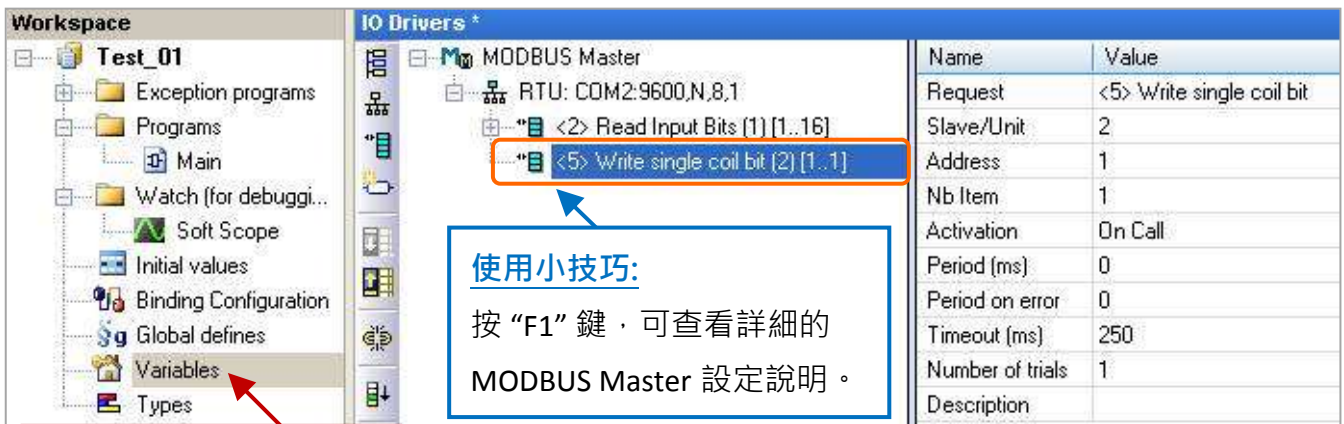
5.1.2 寫出 DO 資料

1. 參考 [5.1 節](#) - 步驟 4，建立第 2 個 Data Block，於“MODBUS Master Request”設定視窗中，設定以下項目並於完成後按“OK”。



- a. **Slave/Unit:**
填入 Slave 設備的站號 (Net-ID)，此例為“2”。
- b. **MODBUS Request:** 此例，
選擇“<5> Write single coil bit”。
- c. **Base address:**
預設值由 1 開始。
(如需修改其值，可參考 [5.1.1 節](#)。)
- d. **On call:** 表示程式有呼叫時，才進行發送一次。
(其它項目說明，可參考 [5.1.1 節](#)。)
- e. **Timeout:**
設定多久未回應，即表示異常。
(對於 Modbus RTU/ASCII 建議值：200 ~ 1000 ms；此例為 250 ms)

2. 接著，請開啟“Variables”視窗，設定需使用的變數。



滑鼠雙擊，開啟該視窗。

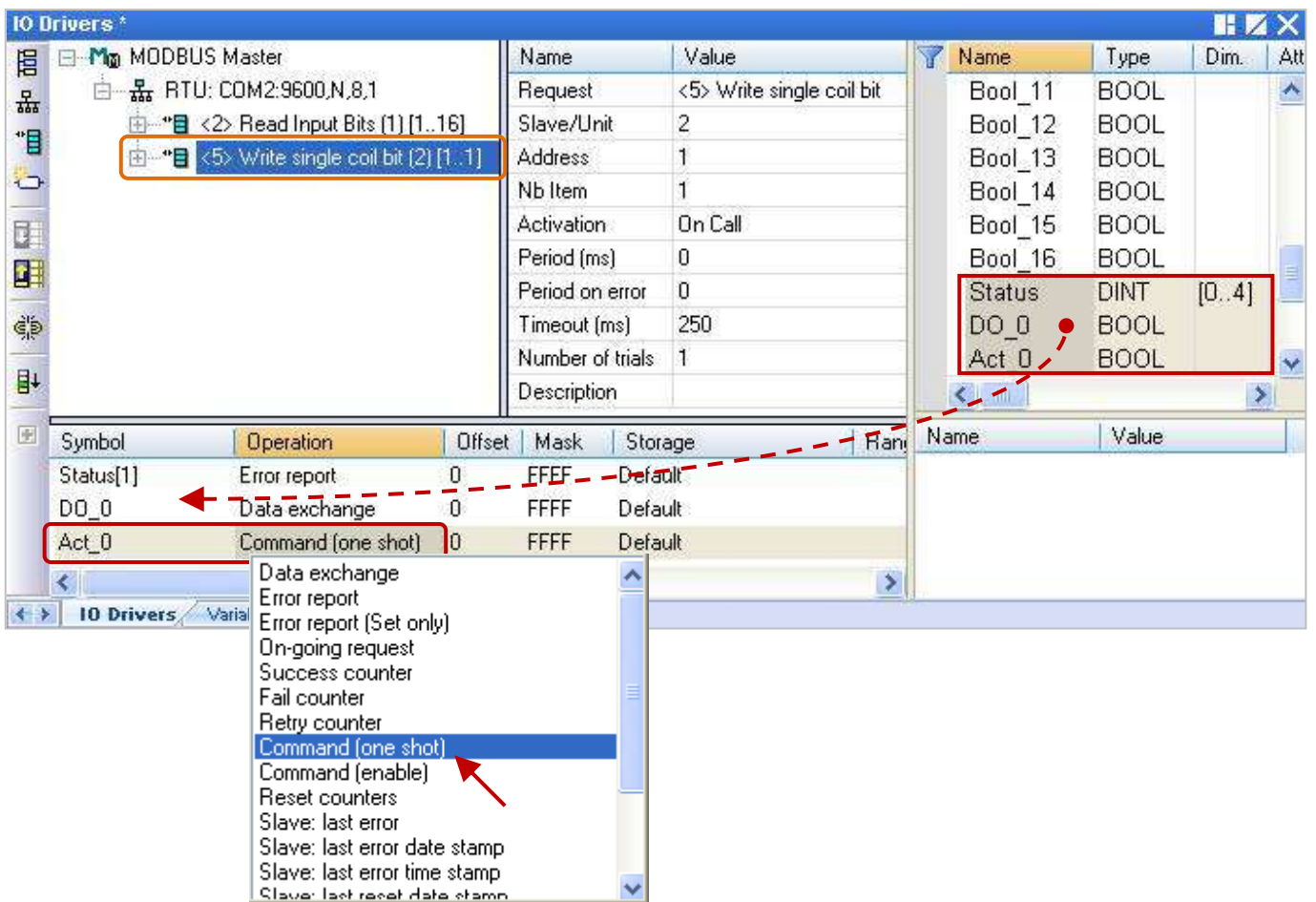
設定 2 個布林變數 (設定方式可參考 [2.3.1 節](#))。

變數名稱	資料型態	說明
DO_0	BOOL	用來寫出 DO 資料。
Act_0	BOOL	此例選擇了“On call”的寫出方式，因此需設定此變數來啟動它。

設定完成後，畫面如下。



3. 於 "I/O Drivers" 視窗，請將變數區中的變數 ("DO_0"、"Act_0" 與 5.1.1 節中建立的 "Status") 拖曳到第 2 個 Data Block 的 "Symbol" 區域。**注意:** "Status" 是一個陣列變數，拖曳到 "Symbol" 區域會是 "Status[0] ~ Status[4]"，請按 "Delete" 鍵來刪除 "Status[0]" 與 "Status[2] ~ [4]"。
4. 設定 "Status[1]" 的 "Operation" 為 "Error report" (表示讀取失敗時，該變數值為一個 "Error Code"，讀取成功時則會重置為 "0")，按 "F1" 鍵則可查看 Modbus Master 設定說明，於標題 "Status and command variables" 中有詳細的命令、"Error Code" 說明。
5. 設定 "Act_0" 的 "Operation" 為 "Command (one shot)"，表示當 "Act_0" 被設定為 "TRUE" 時，會發送指令一次，並自動重置為 "FALSE"；若選用 "Command (Enable)"，表示當 "Act_0" 被設定為 "TRUE" 時，會輪流發送連續指令直到 "Act_0" 被設定為 "FALSE" 時，才會停止發送指令。



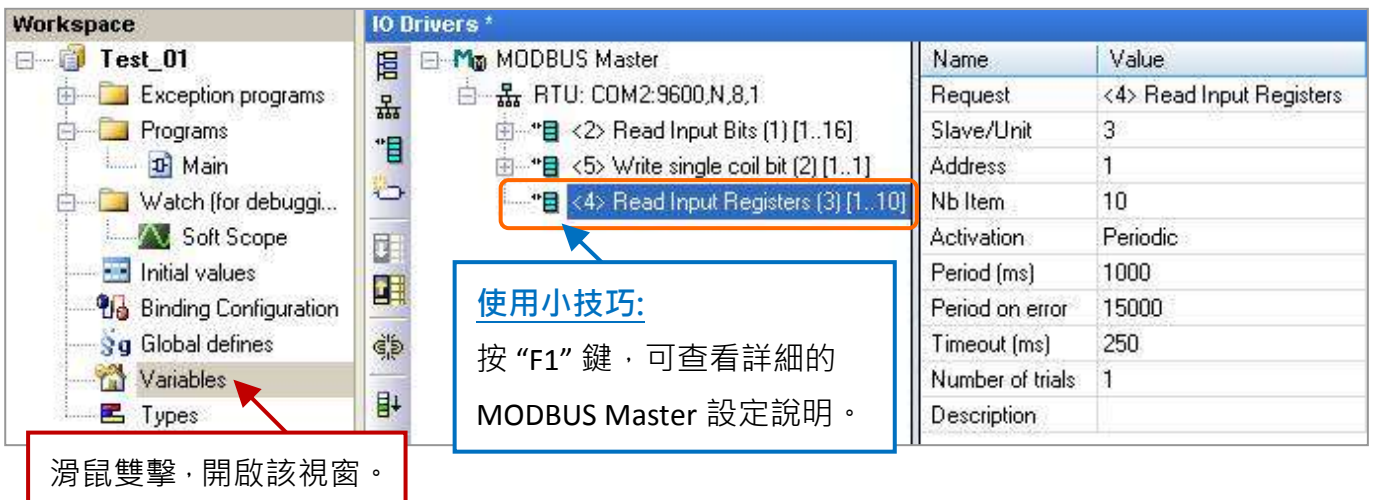
5.1.3 讀取 AI 資料

1. 參考 [5.1 節](#) - 步驟 4，建立第 3 個 Data Block，於“MODBUS Master Request”設定視窗中，設定以下項目並於完成後按“OK”。



- a. **Slave/Unit:**
填入 Slave 設備的站號 (Net-ID，此例為“3”)。
- b. **MODBUS Request:** 此例，
選擇“<4> Read Input Registers”。
- c. **Base address:**
預設值由 1 開始。
(如需修改其值，可參考 [5.1.1 節](#)。)
- Nb items:**
讀取 AI 的數量 (此例為“10”)。
- d. **Periodic:** (可參考 [5.1.1 節](#)。)
表示週期性的發送請求，此例為每 1 秒發送一次。“on error”表示每當發生異常時，下一次的發送時間 (此例為 15 秒)。
- e. **Timeout:**
設定多久未回應，即表示異常。
(對於 Modbus RTU/ASCII 建議值：200 ~ 1000 ms；此例為 250 ms)

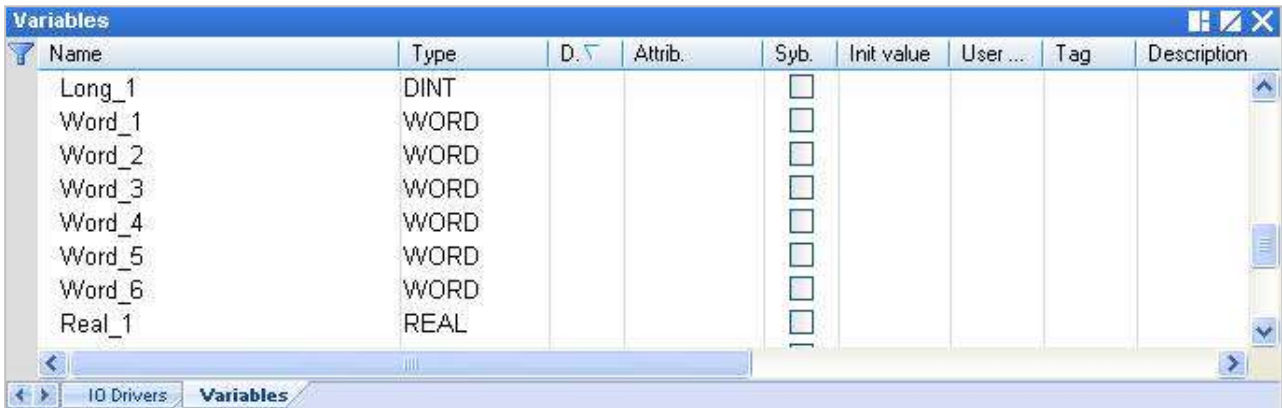
2. 接著，請開啟“Variables”視窗，設定需使用的變數。



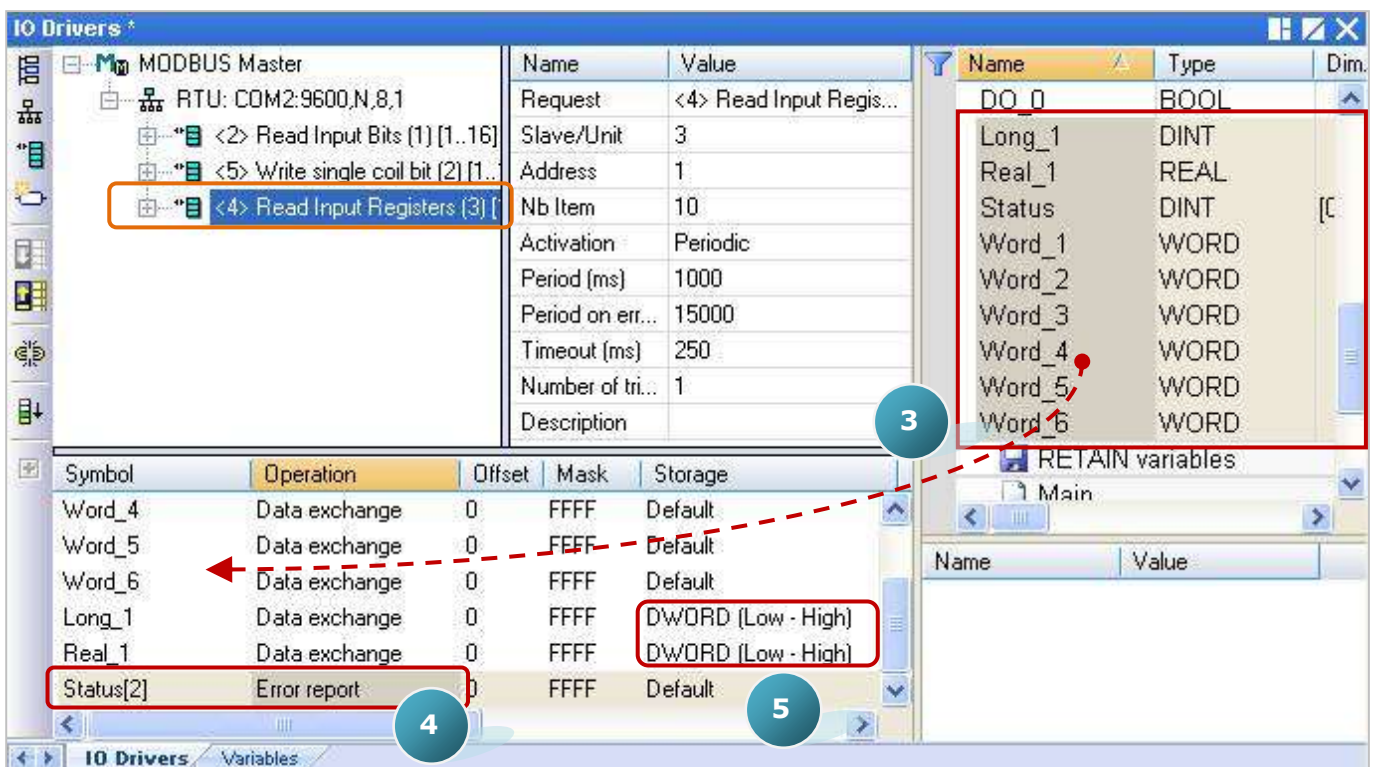
此處需設定 6 個 Word (16 bit)、1 個 Double Word (32 bit) 與 1 個 Real (32 bit) 變數。
(設定方式可參考 [2.3.1 節](#))，請依照下表來設定。

變數名稱	資料型態	說明
Word_1 ~ Word_6	WORD	用來讀取 AI 資料 (16 bit)。
Long_1	DINT	用來讀取 AI 資料 (32 bit)。
Real_1	REAL	用來讀取 AI 資料 (32 bit)。

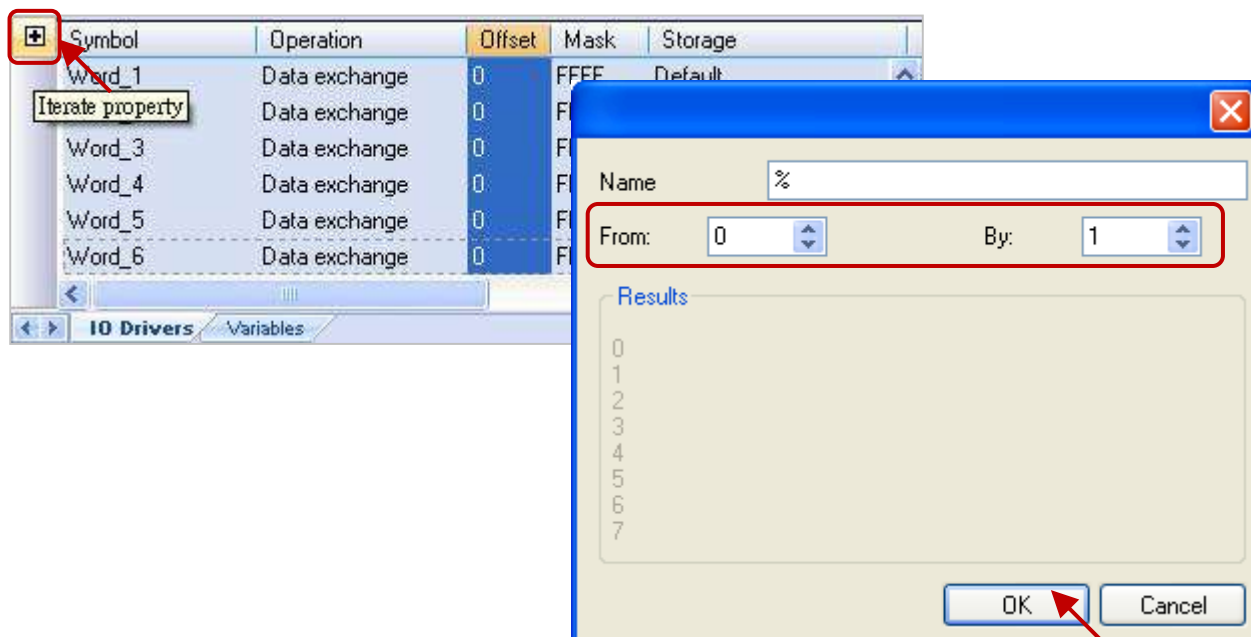
您可參考 [附錄 A](#) 來了解變數的資料形態與範圍，設定完成後，畫面如下。



- 於 “I/O Drivers” 視窗，請將變數區中的變數 (“Word_1 ~ Word_6”、“Long_1”、“Real_1” 與 [5.1.1 節](#) 中建立的 “Status”) 拖曳到第 3 個 Data Block 的 “Symbol” 區域。**注意:** “Status” 是一個陣列變數，拖曳到 “Symbol” 區域會是 “Status[0] ~ Status[4]”，請刪除 “Status[0] ~ [1]” 與 “Status[3] ~ [4]”。
- 設定 “Status[2]” 的 “Operation” 為 “Error report” (表示讀取失敗時，該變數值為一個 “Error Code”，讀取成功時則會重置為 “0”)，按 “F1” 鍵則可查看 Modbus Master 設定說明，於標題 “Status and command variables” 中有詳細的命令、“Error Code” 說明。
- “Long_1”、“Real_1” 為 32-bit 資料 (一個資料需占用 2 個 Modbus 位址)，設定其 “Storage” 為 “DWORD (Low – High)”。

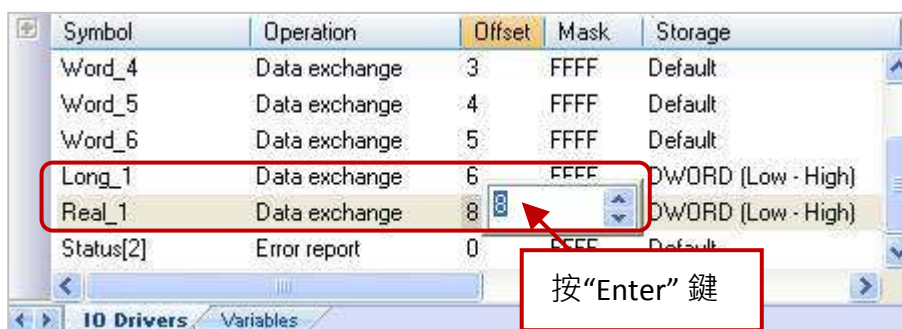


6. 如圖，選取 "Word_1 ~ Word_6" 並點選 "Iterate property" 設定 Offset 值 (From: 0 ; By: 1)。



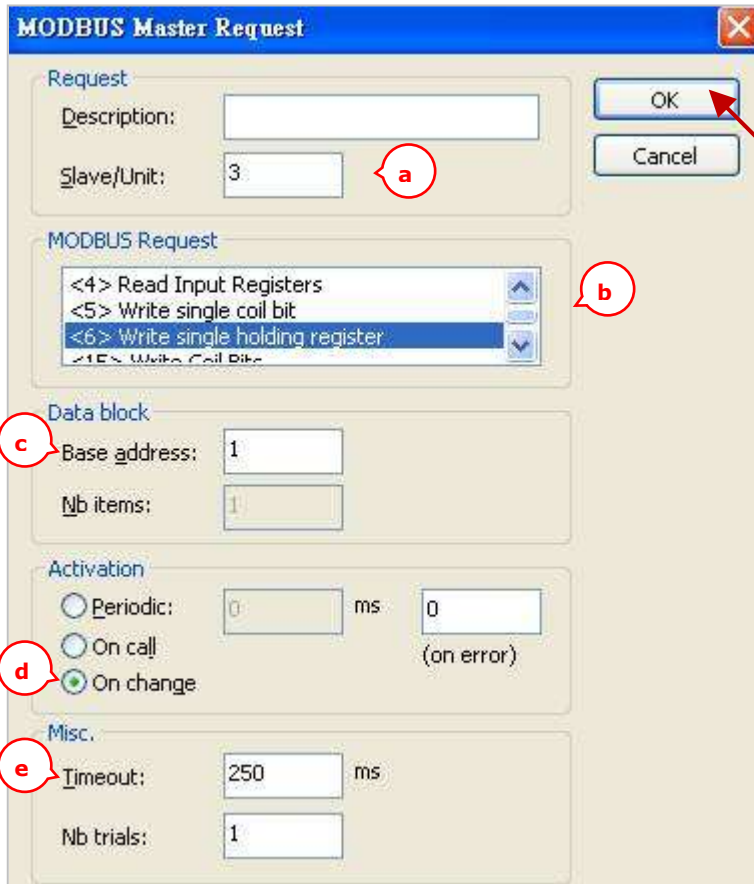
7. 接著，滑鼠雙擊 "Long_1"、"Real_1" 的 Offset 輸入其值為 "6"、"8" 並按 "Enter" 鍵完成設定。

註： 32-bit 資料需占用 2 個 Modbus 位址，例如下方 "Long_1" 的 Offset 為 "6"，則下一個 "Real_1" 的 Offset 值需設定為 "8"。



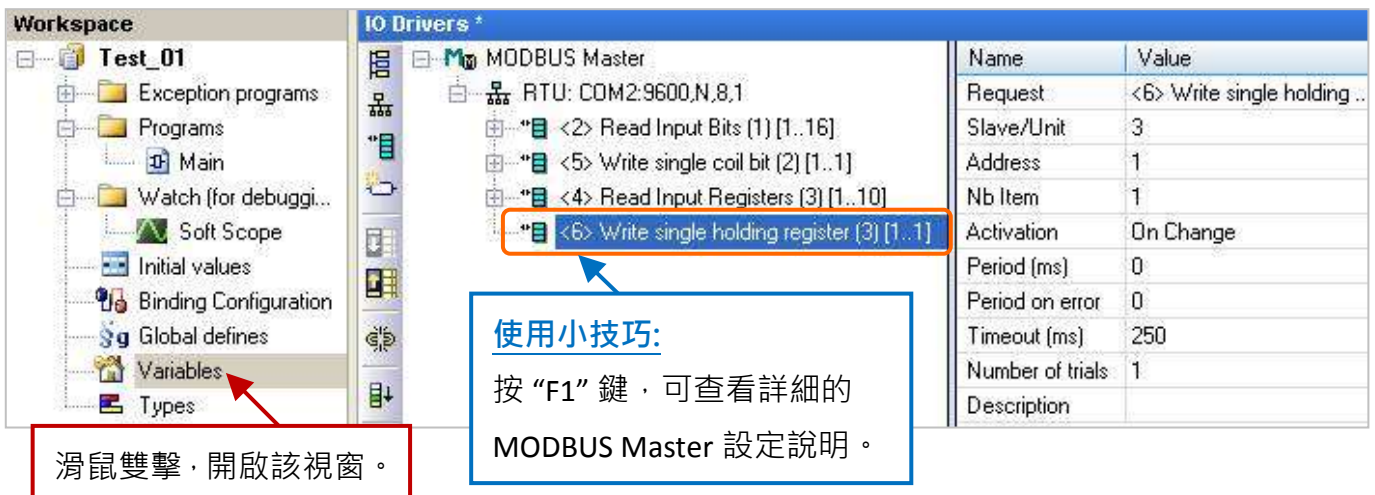
5.1.4 寫出 AO 資料 (16-bit)

1. 參考 [5.1 節](#) - 步驟 4，建立第 4 個 Data Block，於“MODBUS Master Request”設定視窗中，設定以下項目並於完成後按“OK”。



- a. **Slave/Unit:**
填入 Slave 設備的站號 (Net-ID，此例為“3”)。
- b. **MODBUS Request:** 此例，選擇“<6> Write single holding register”。
- c. **Base address:**
預設值由 1 開始。
(如需修改其值，可參考 [5.1.1 節](#)。)
- d. **On change:** 表示寫出的資料有改變時，才進行發送一次。
(其它項目說明，可參考 [5.1.1 節](#)。)
- e. **Timeout:**
設定多久未回應，即表示異常。
(對於 Modbus RTU/ASCII 建議值：200 ~ 1000 ms；此例為 250 ms)

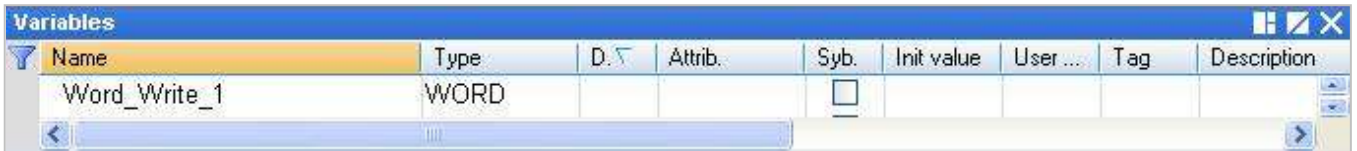
2. 接著，請開啟“Variables”視窗，設定需使用的變數。



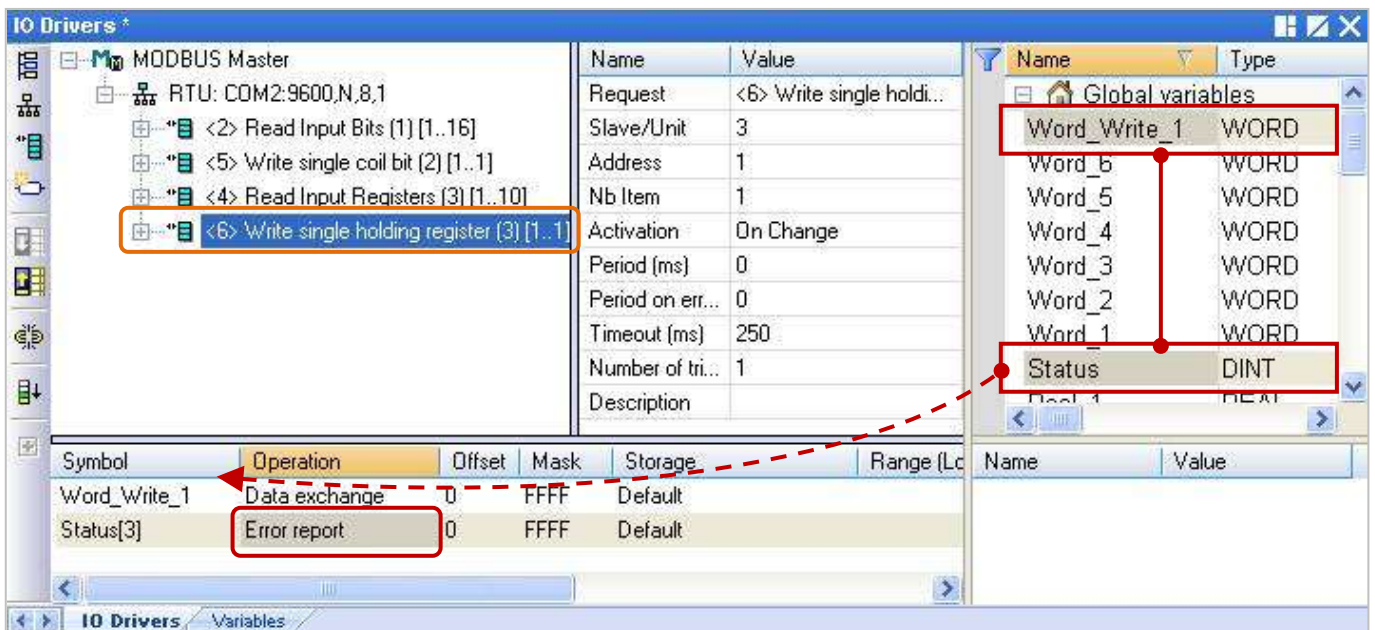
設定 1 個數值變數，您可參考 [附錄 A](#) 來了解變數的資料形態與範圍 (設定方式可參考 [2.3.1 節](#))。

變數名稱	資料型態	說明
Word_Write_1	WORD	用來寫出 AO 資料 (16-bit)。

設定完成後，畫面如下。

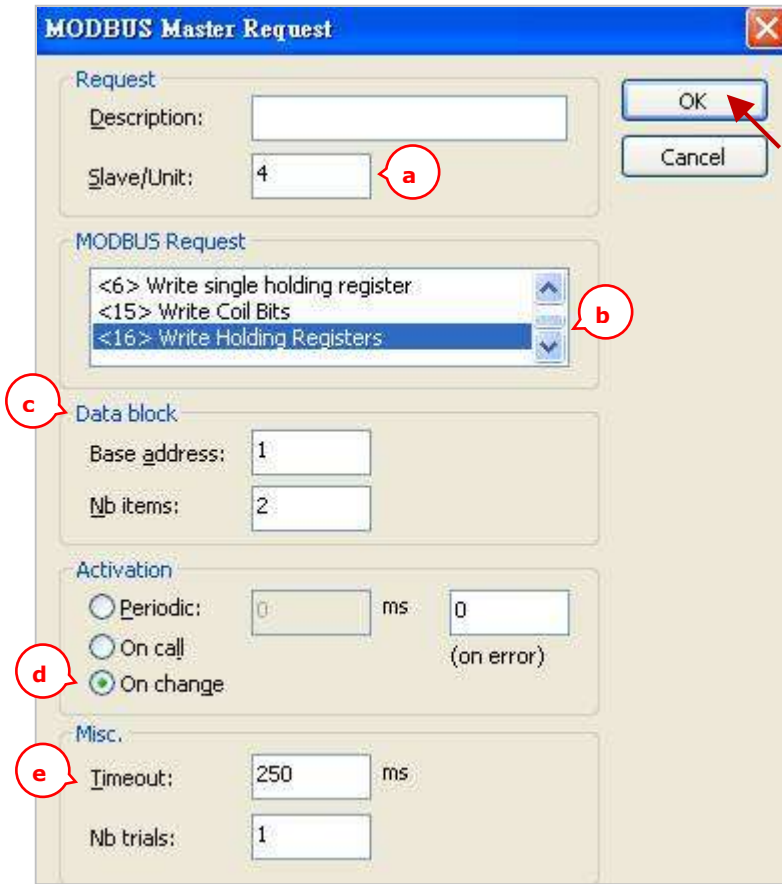


- 於 “I/O Drivers” 視窗，請將變數區中的變數 (“Word_Write_1” 與 5.1.1 節中建立的 “Status”) 拖曳到第 4 個 Data Block 的 “Symbol” 區域。**注意:** “Status” 是一個陣列變數，拖曳到 “Symbol” 區域會是 “Status[0] ~ Status[4]”，請按 “Delete” 鍵來刪除 “Status[0] ~ [2]” 與 “Status[4]”。
- 設定 “Status[3]” 的 “Operation” 為 “Error report” (表示讀取失敗時，該變數值為一個 “Error Code”，讀取成功時則會重置為 “0”)，按 “F1” 鍵則可查看 Modbus Master 設定說明，於標題 “Status and command variables” 中有詳細的命令、“Error Code” 說明。



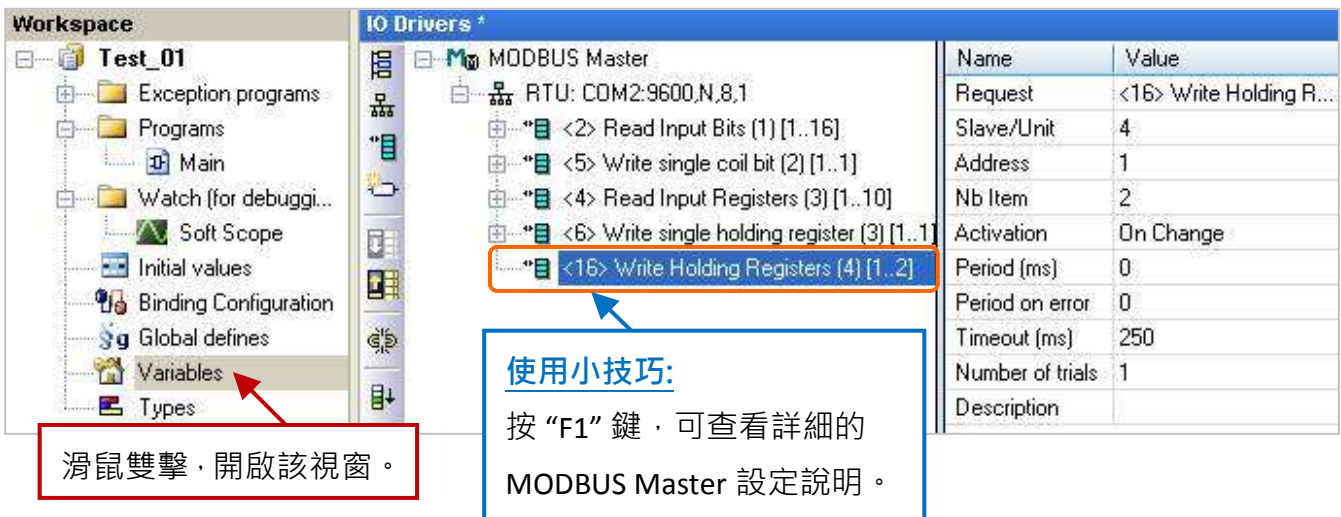
5.1.5 寫出 AO 資料 (32-bit)

1. 參考 [5.1 節](#) - 步驟 4，建立第 5 個 Data Block，於 “MODBUS Master Request” 設定視窗中，設定以下項目並於完成後按 “OK”。



- a. **Slave/Unit:**
填入 Slave 設備的站號 (Net-ID，此例為 “4”)。
- b. **MODBUS Request:** 此例，選擇 “<16> Write Holding Registers”。
- c. **Base address:**
預設值由 1 開始。
(如需修改其值，可參考 [5.1.1 節](#)。)
- Nb items:**
寫入 AO 的數量 (此例為 “2”，占用 2 個 Modbus 位址)。
- d. **On change:** 表示寫出的資料有改變時，才進行發送一次。
(其它項目說明，可參考 [5.1.1 節](#)。)
- e. **Timeout:**
設定多久未回應，即表示異常。
(對於 Modbus RTU/ASCII 建議值: 200 ~ 1000 ms；此例為 250 ms)

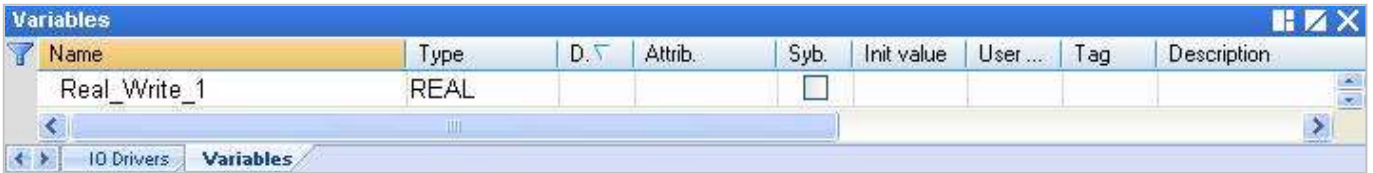
2. 接著，請開啟 “Variables” 視窗，設定需使用的變數



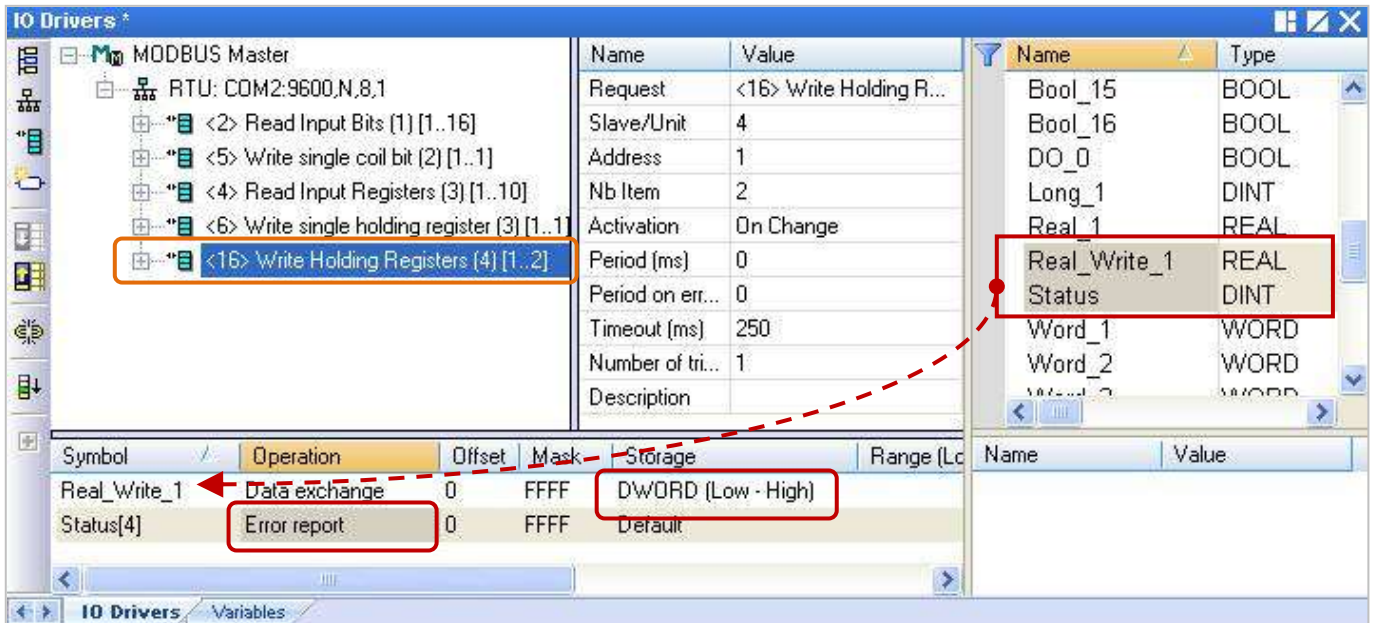
設定 1 個實數變數，您可參考 [附錄 A](#) 來了解變數的資料形態與範圍 (設定方式可參考 [2.3.1 節](#))。

變數名稱	資料型態	說明
Real_Write_1	REAL	用來寫出 AO 資料 (32-bit)。

設定完成後，畫面如下。



3. 於“IO Drivers”視窗，請將變數區中的變數 (“Real_Write_1” 與 5.1.1 節中建立的 “Status”) 拖曳到第 5 個 Data Block 的 “Symbol” 區域。**注意:** “Status” 是一個陣列變數，拖曳到 “Symbol” 區域會是 “Status[0] ~ Status[4]”，請刪除 “Status[0] ~ [3]”。
4. 設定 “Status[4]” 的 “Operation” 為 “Error report” (表示讀取失敗時，該變數值為一個 “Error Code”，讀取成功時則會重置為 “0”)，按 “F1” 鍵則可查看 Modbus Master 設定說明，於標題 “Status and command variables” 中有詳細的命令、“Error Code” 說明。
5. “Real_Write_1” 為 32-bit 資料 (一個資料需占用 2 個 Modbus 位址)，設定其 “Storage” 為 “DWORD (Low – High)”。



5.1.6 XV Board 使用說明

XV board 是屬於 Modbus Slave I/O 板卡，使用前請先將此板卡插在 WP-5xx8-CE7 內，並在 Win-GRAF 中啟用 WP-5xx8-CE7 為 Modbus Master (參考 5.1 節)。請參訪 XV board 選型頁面，查詢詳細的規格資訊：

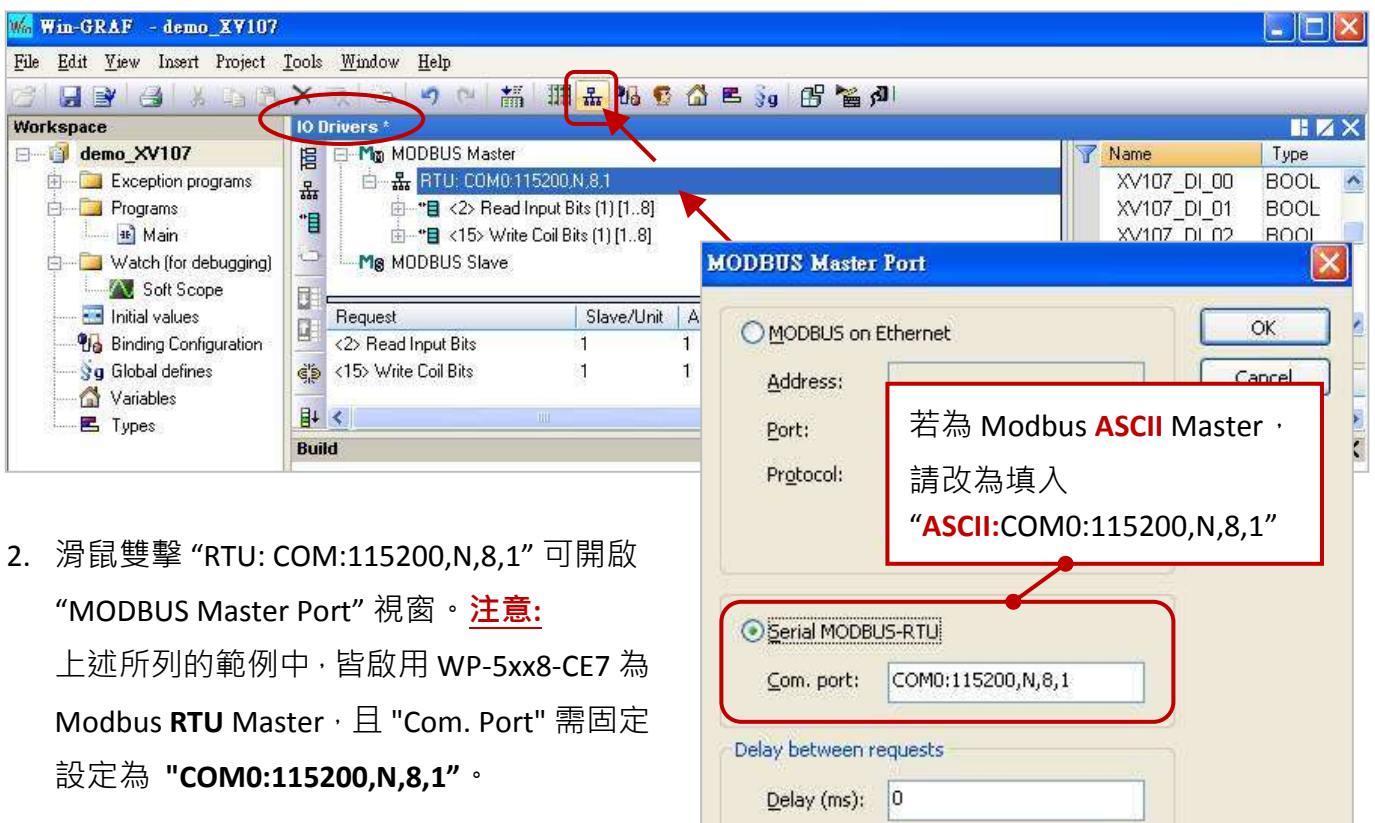
www.icpdas.com/root/product/solutions/hmi_touch_monitor/touchpad/xv-board_selection.html

以下章節所提供的 Win-GRAF 範例程式，可在出貨光碟中找到，請參考 第 12 章，點選功能表 "File" > "Add Existing Project" > "From Zip"，來回存範例程式並查看詳細的程式內容。(CD-ROM:\Napdos\Win-GRAF\demo-project\)

範例	檔名	說明
XV107, XV107A	demo_XV107.zip	讀取 8 DI，寫出 8 DO
XV110	demo_XV110.zip	讀取 16 DI
XV111, XV111A	demo_XV111.zip	寫出 16 DO，讀取 1 DO
XV116	demo_XV116.zip	讀取 5 DI，寫出 6 DO
XV308_1 XV308_2 XV308_3	demo_XV308_1.zip demo_XV308_2.zip demo_XV308_3.zip	1. 讀取 8 AI，讀取 8 DI 2. 讀取 8 AI，寫出 8 DO 3. 讀取 8 AI，寫出 4 DO，讀取 4 DI
XV310	demo_XV310.zip	讀取 4 AI，寫出 4 DO，讀取 4 DI，寫出 4 AO

通用設定說明:

1. 滑鼠點選工具列上的“Open Fieldbus Configuration” 按鈕來開啟 “I/O Drivers” 視窗。



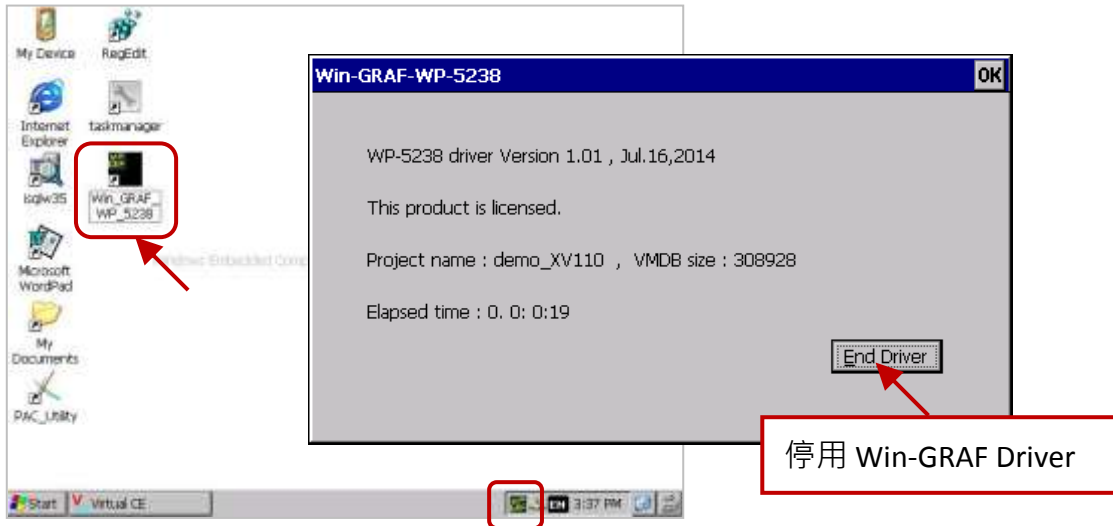
2. 滑鼠雙擊 “RTU: COM:115200,N,8,1” 可開啟 “MODBUS Master Port” 視窗。注意：上述所列的範例中，皆啟用 WP-5xx8-CE7 為 Modbus RTU Master，且 "Com. Port" 需固定設定為 "COM0:115200,N,8,1"。

規劃 AI/AO 通道

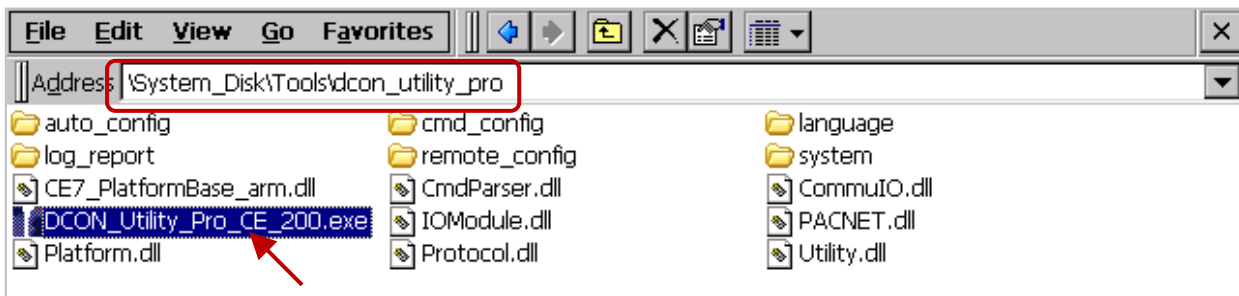
若要在 WP-5xx8-CE7 內使用 XV Board (例如: XV308, XV310) 的 AI/AO 通道，使用前需在 PAC 內先暫停 Win-GRAF Driver，再使用 "DCON_Utility_Pro_CE_200.exe" 來規劃每個 AI/AO 通道。

以 WP-5238-CE7 為例來說明:

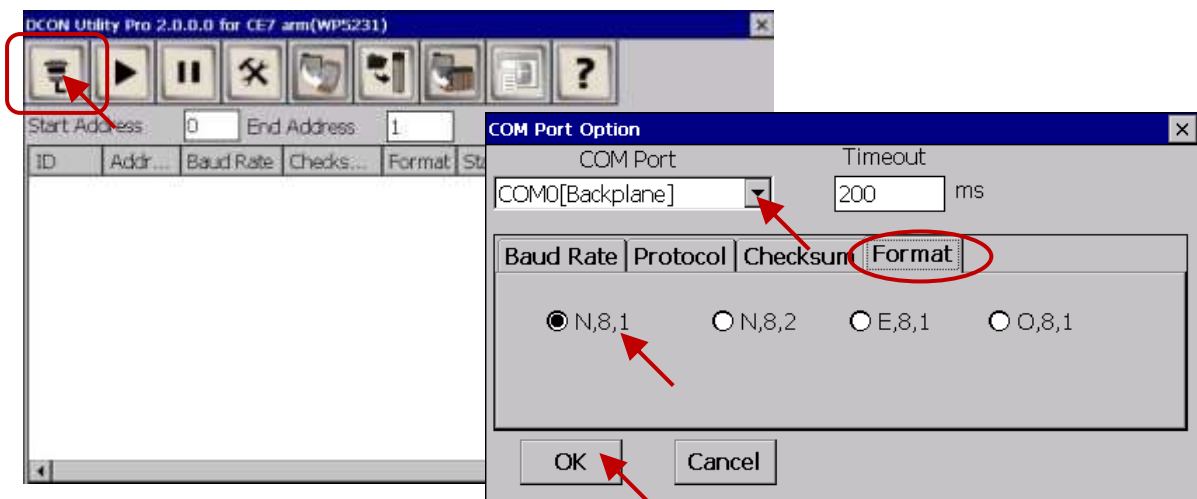
1. 點選 "Win_GRAF_WP_5238" (或下方的小圖示) 開啟 Win-GRAF Driver 視窗，並點選 "End Driver"。



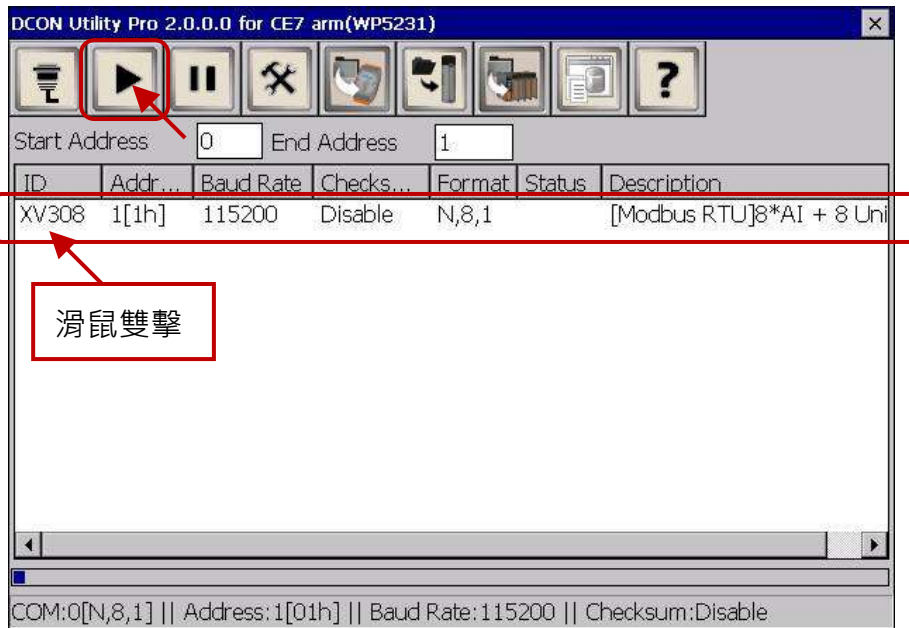
2. 點選桌面上的 "My Device" 並在 \System_Disk\Tools\dcon_utility_pro 路徑下，點選 "DCON_Utility_Pro_CE_200.exe" 開啟設定視窗。



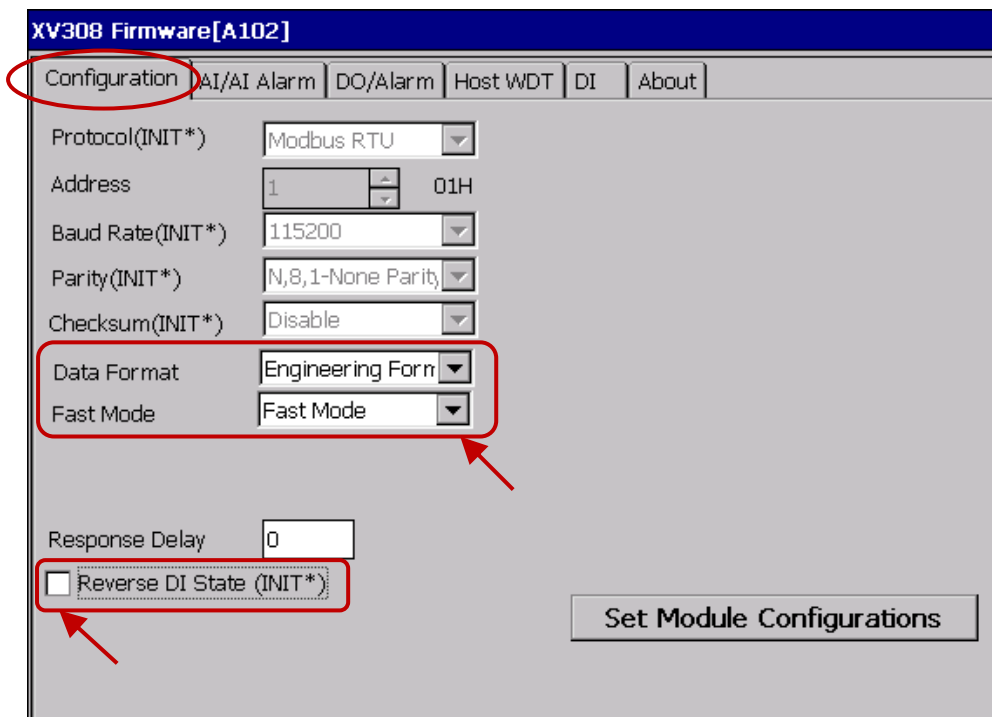
3. 點選 COM Port 按鈕開啟設定視窗，接著 "COM Port" 固定設為 "COM0"，"Baud Rate" 固定設為 115200, "Format" 固定為 "N,8,1"，再按 "OK"。



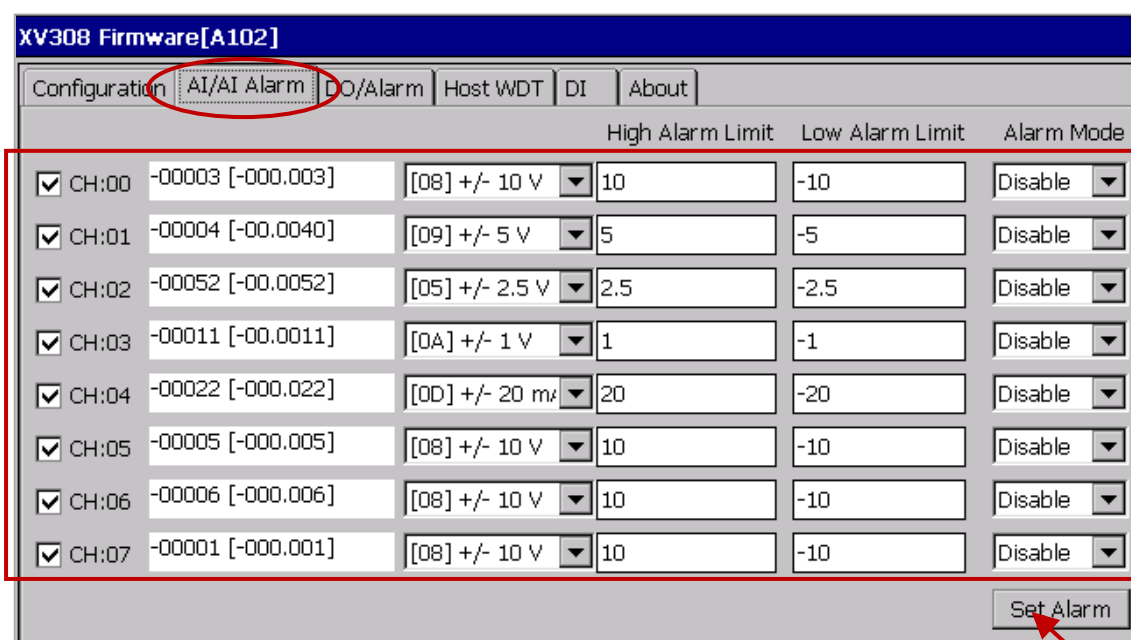
4. 點選搜尋按鈕後，將列出 PAC 中的 XV Board (例如: XV308)，滑鼠雙擊此項目來開啟設定視窗。



5. 於設定畫面中，將 "Data Format" 設定為 "Engineering Format" (建議設定)，將 "Fast Mode" 設定為 "Fast Mode" 並取消勾選 "Reverse DI State (INIT*)"。



6. 於設定畫面中，依需求為每個 AI 通道規劃適當的設定並記得勾選欲使用的 AI 通道，完成後再按 "Set Alarm" 按鈕。



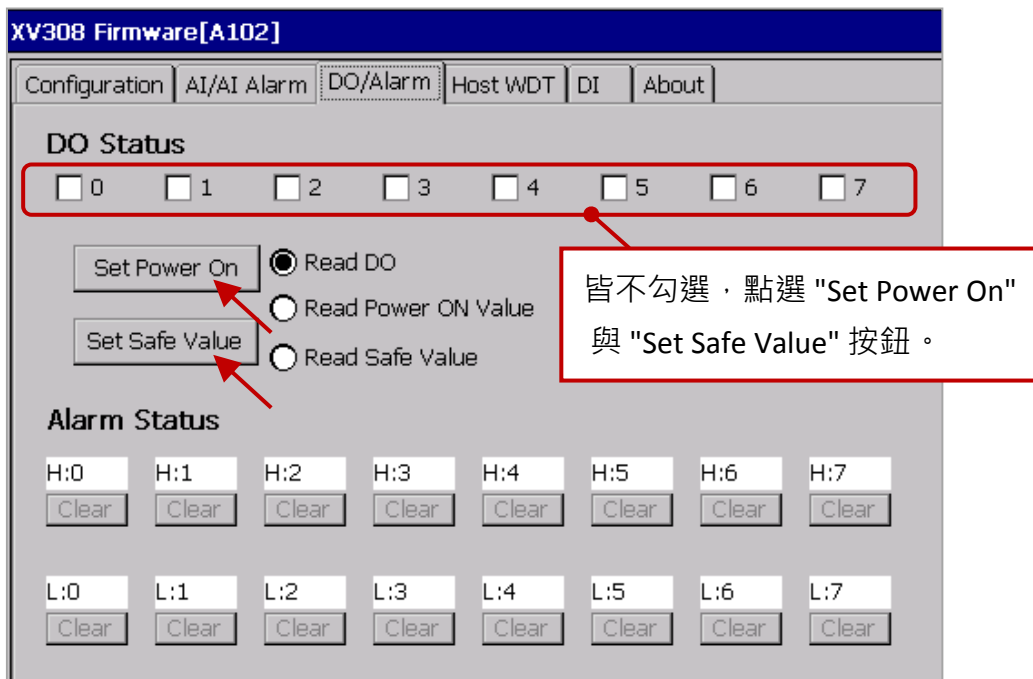
XV308:

Type Code	Range	Data Format	Minimum	Maximum
05	+/-2.5 V	Engineering	-25000	+25000
		Hexadecimal	8000h	7FFFh
06	+/-20 mA	Engineering	-20000	+20000
		Hexadecimal	8000h	7FFFh
07	+4 mA ~ +20 mA	Engineering	+4000	+20000
		Hexadecimal	0000h	FFFFh
08	+/-10 V	Engineering	-10000	+10000
		Hexadecimal	8000h	7FFFh
09	+/-5 V	Engineering	-5000	+5000
		Hexadecimal	8000h	7FFFh
0A	+/-1 V	Engineering	-10000	+10000
		Hexadecimal	8000h	7FFFh
0D	+/-20 mA	Engineering	-20000	+20000
		Hexadecimal	8000h	7FFFh
1A	0 mA ~ +20 mA	Engineering	0	+20000
		Hexadecimal	0000h	FFFFh

註:

- 資料格式建議使用 "Engineering"，較方便辨識數值。
(例如: +/-2.5 V 顯示為 -25000 ~ +25000，+4 mA ~ +20 mA 顯示為 +4000 ~ +20000)
- 若使用的 "Type Code" 為 06, 07, 0D, 1A 需注意 XV Board 上 8 個 Jumper 的位置是否正確!
www.icpdas.com/root/product/solutions/datasheet/hmi_touch_monitor/XV308.pdf

注意: 若使用 XV308 需在 "DO/Alarm" 頁籤中，不勾選任何 DO 的情況下，點選 "Set Power On" 與 "Set Safe Value" 按鈕。



- 最後，切換到 "Configuration" 頁籤並點選 "Set Module Configuration" 按鈕 (步驟 5)，即完成了 AI/AO 配置，請關閉 "DCON_Utility_Pro_CE_200.exe" 並點選桌面上的 "Win_GRAF_WP_5238" 重新啟動 Win-GRAF Driver (步驟 1)。

可依照上述類似的步驟來規劃 XV310 的 AI/AO 通道。

XV310 - Analog Input:

Type Code	Range	Data Format	Minimum	Maximum
05	+/-2.5 V	Engineering	-25000	+25000
		Hexadecimal	8000h	7FFFh
06	+/-20 mA	Engineering	-20000	+20000
		Hexadecimal	8000h	7FFFh
07	+4 mA ~ +20 mA	Engineering	+4000	+20000
		Hexadecimal	0000h	FFFFh
08	+/-10 V	Engineering	-10000	+10000
		Hexadecimal	8000h	7FFFh
09	+/-5 V	Engineering	-5000	+5000
		Hexadecimal	8000h	7FFFh
0A	+/-1 V	Engineering	-10000	+10000
		Hexadecimal	8000h	7FFFh
0D	+/-20 mA	Engineering	-20000	+20000
		Hexadecimal	8000h	7FFFh
1A	0 mA ~ +20 mA	Engineering	0	+20000
		Hexadecimal	0000h	FFFFh

註:

1. 資料格式建議使用 "Engineering"，較方便辨識數值。
(例如: +/-2.5 V 顯示為 -25000 ~ +25000，+4 mA ~ +20 mA 顯示為 +4000 ~ +20000)
2. 若使用的 "Type Code" 為 0, 1, 06, 07, 0D, 1A 請注意板卡上 4 個 Jumper 的位置是否正確!
www.icpdas.com/root/product/solutions/datasheet/hmi_touch_monitor/XV310.pdf

XV310 - Analog Output:

Type Code	Range	Data Format	Minimum	Maximum
0	0 mA ~ +20 mA	Engineering	0	+20000
		Hexadecimal	0000h	FFFFh
1	+4 mA ~+20 mA	Engineering	+4000	+20000
		Hexadecimal	0000h	FFFFh
2	0V ~ +10 V	Engineering	0	+10000
		Hexadecimal	0000h	FFFFh
3	+/-10 V	Engineering	-10000	+10000
		Hexadecimal	8000h	7FFFh
4	0 V ~ +5 V	Engineering	0	+5000
		Hexadecimal	0000h	FFFFh
5	+/-5 V	Engineering	-5000	+5000
		Hexadecimal	8000h	7FFFh

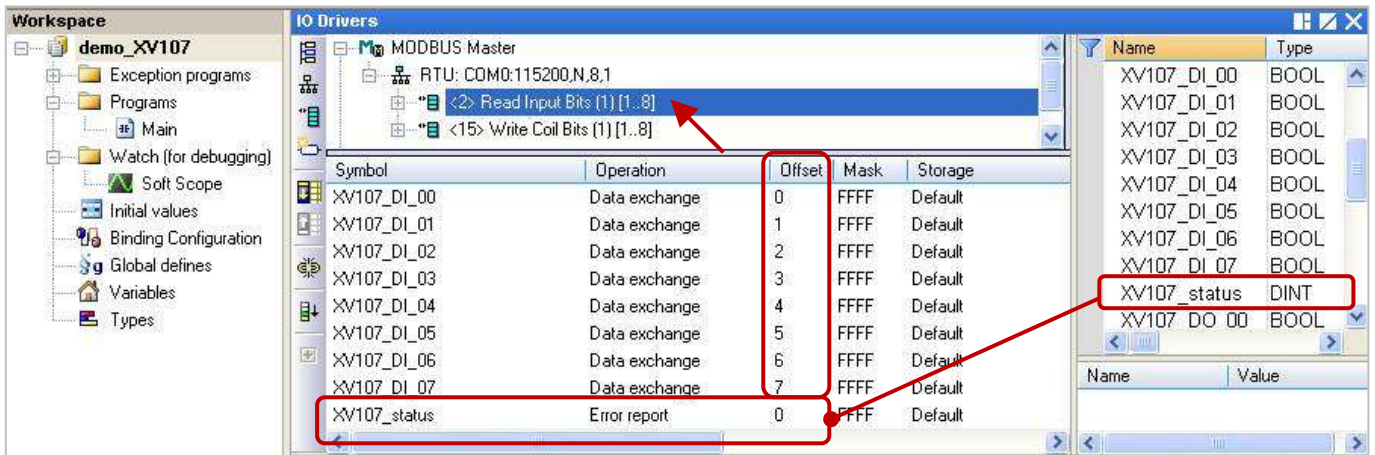
5.1.7 連接 XV107, XV107A (8 DI, 8 DO)

XV107, XV107A 是一款具有 8 DI 與 8 DO 的擴充卡，本章節說明的 Win-GRAF 範例程式為 "demo_XV107.zip"，使用前請先參考 5.1.6 節的 XV Board 使用說明。

範例說明:

此範例建立了 2 個 Data Block，一個用來讀取 8 個 DI 資料，另一個用來寫出 8 個 DO 資料。

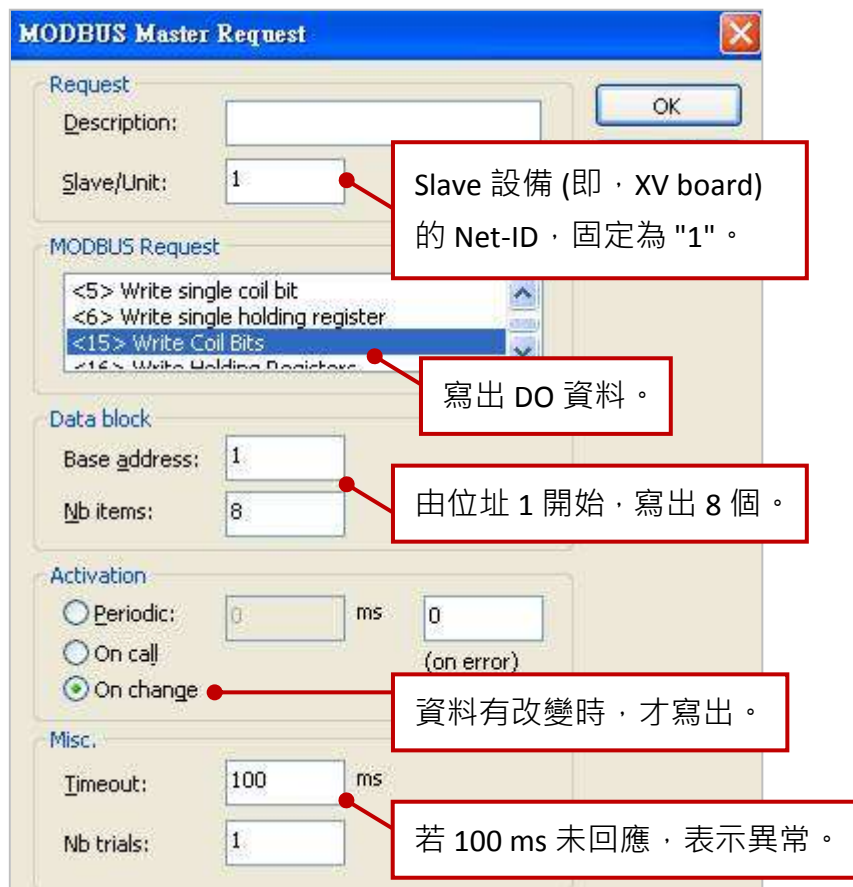
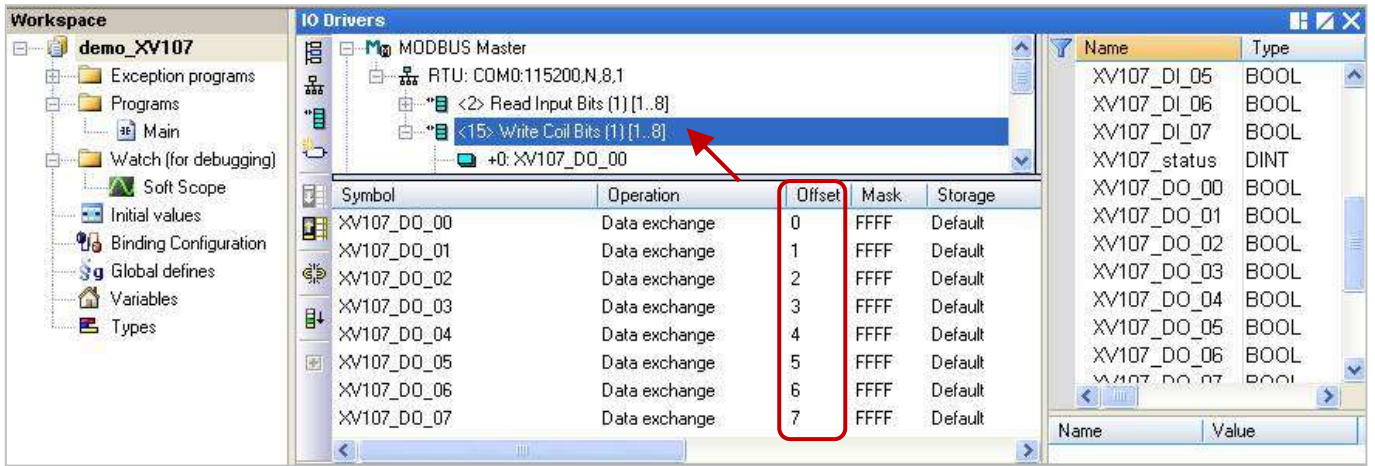
1. 滑鼠雙擊第 1 個 Data Block (即，<2> Read Input Bits) 來開啟設定視窗。



注意: "Offset" 的值是由 "0" 開始，而 "Offset" 值加 1 (Base address) 才是該變數的 Modbus 位址。若 "Operation" 設定為 "Error report"，該變數 (資料型態: DINT) 的 "Offset" 值需設定為 "0"。



2. 滑鼠雙擊第 2 個 Data Block (即 · <15> Write Coil Bits) 來開啟設定視窗。

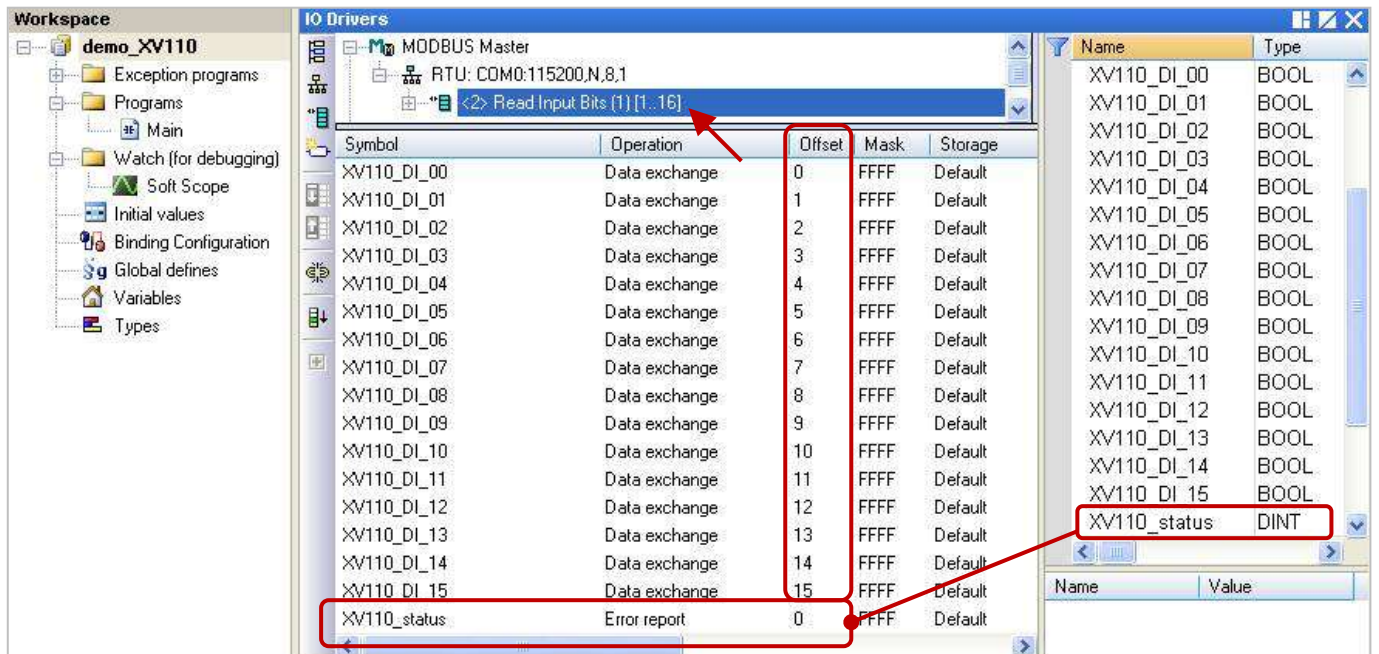


5.1.8 連接 XV110 (16 DI)

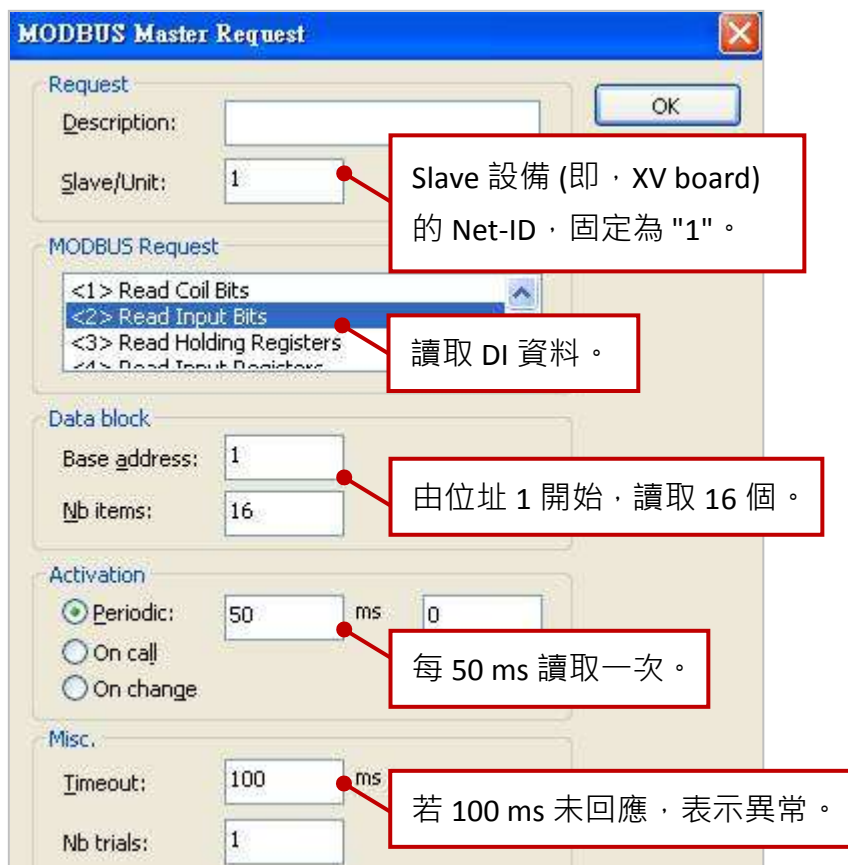
XV110 是一款具有 16 DI 的擴充卡，本章節的 Win-GRAF 範例程式為 "demo_XV110.zip"，使用前請先參考 5.1.6 節的 XV Board 使用說明。

範例說明：此範例建立了 1 個 Data Block，用來讀取 16 個 DI 資料。

1. 滑鼠雙擊 "<2> Read Input Bits" 來開啟設定視窗。



注意：“Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。若 “Operation” 設定為 “Error report”，該變數 (資料型態: DINT) 的 “Offset” 值需設定為 “0”。

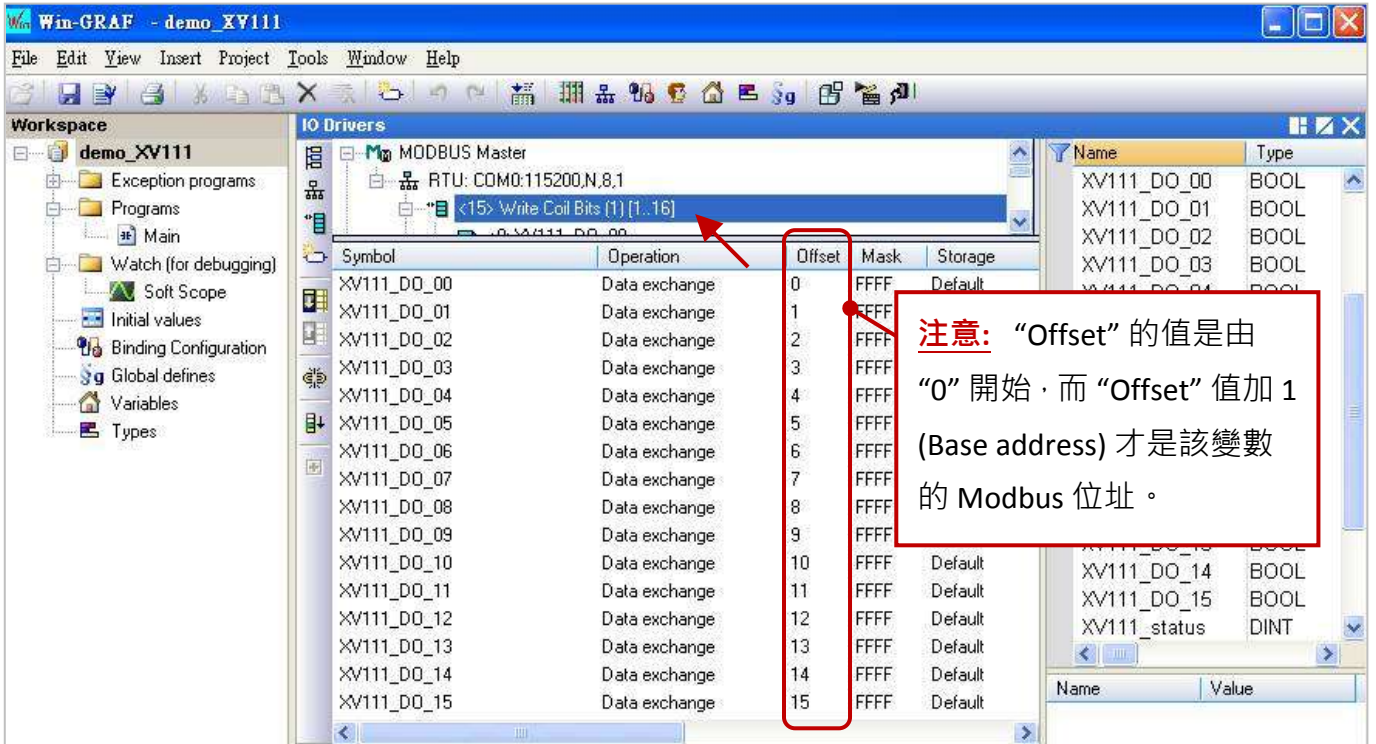


5.1.9 連接 XV111, XV111A (16 DO)

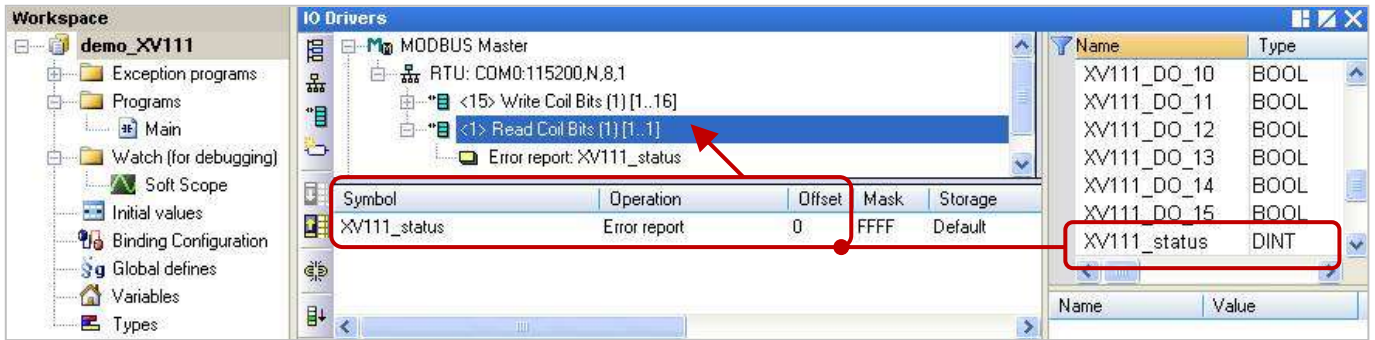
XV111, XV111A 是一款具有 16 DO 的擴充卡，本章節的 Win-GRAF 範例程式為 "demo_XV111.zip"，使用前請先參考 5.1.6 節的 XV Board 使用說明。

範例說明：此範例建立了 2 個 Data Block，一個用來寫出 16 個 DO 資料，另一個用來讀取 DO 狀態。

1. 滑鼠雙擊第 1 個 Data Block (即，<15> Write Coil Bits) 來開啟設定視窗。



2. 滑鼠雙擊第 2 個 Data Block (即 · <1> Read Coil Bits) 來開啟設定視窗。



注意: 若 “Operation” 設定為 "Error report"，該變數 (資料型態: DINT) 的 “Offset” 值需設定為 “0”。



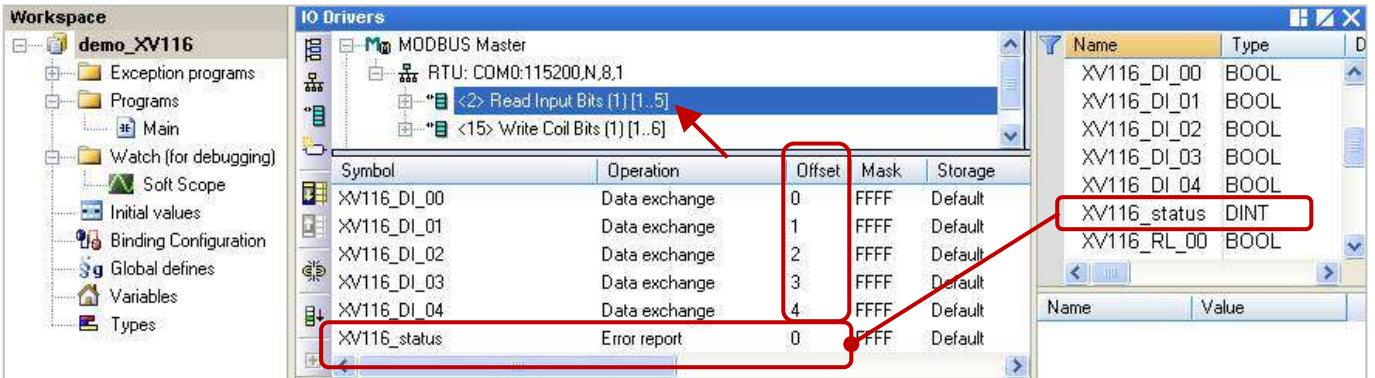
5.1.10 連接 XV116 (5 DI, 6 Relay)

XV116 是一款具有 5 DI 與 6 Relay 的擴充卡，本章節的 Win-GRAF 範例程式為 "demo_XV116.zip"，使用前請先參考 [5.1.6 節](#) 的 XV Board 使用說明。

範例說明:

此範例建立了 2 個 Data Block，一個用來讀取 5 個 DI 資料，另一個用來寫出 6 個 DO 資料。

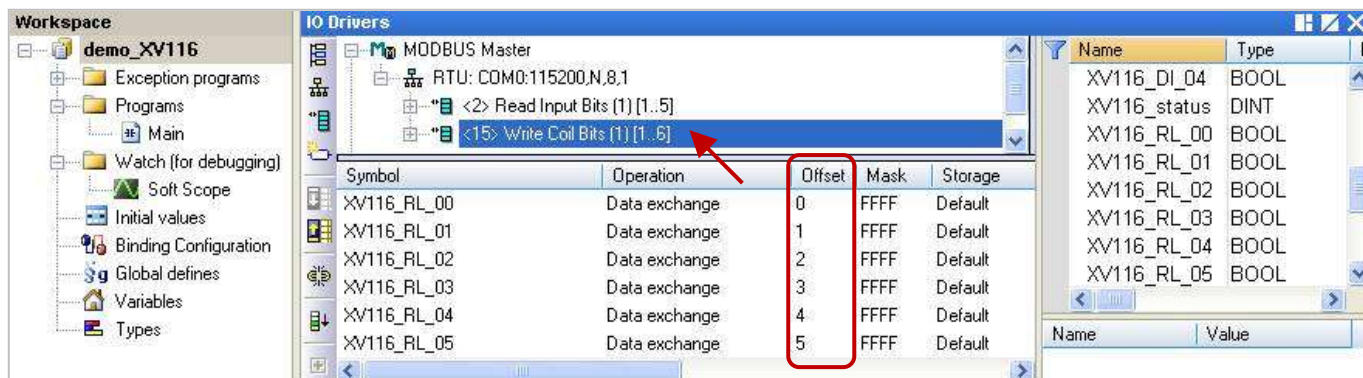
1. 滑鼠雙擊第 1 個 Data Block (即，<2> Read Input Bits) 來開啟設定視窗。



注意: "Offset" 的值是由 "0" 開始，而 "Offset" 值加 1 (Base address) 才是該變數的 Modbus 位址。若 "Operation" 設定為 "Error report"，該變數 (資料型態: DINT) 的 "Offset" 值需設定為 "0"。



2. 滑鼠雙擊第 2 個 Data Block (即 · <15> Write Coil Bits) 來開啟設定視窗。



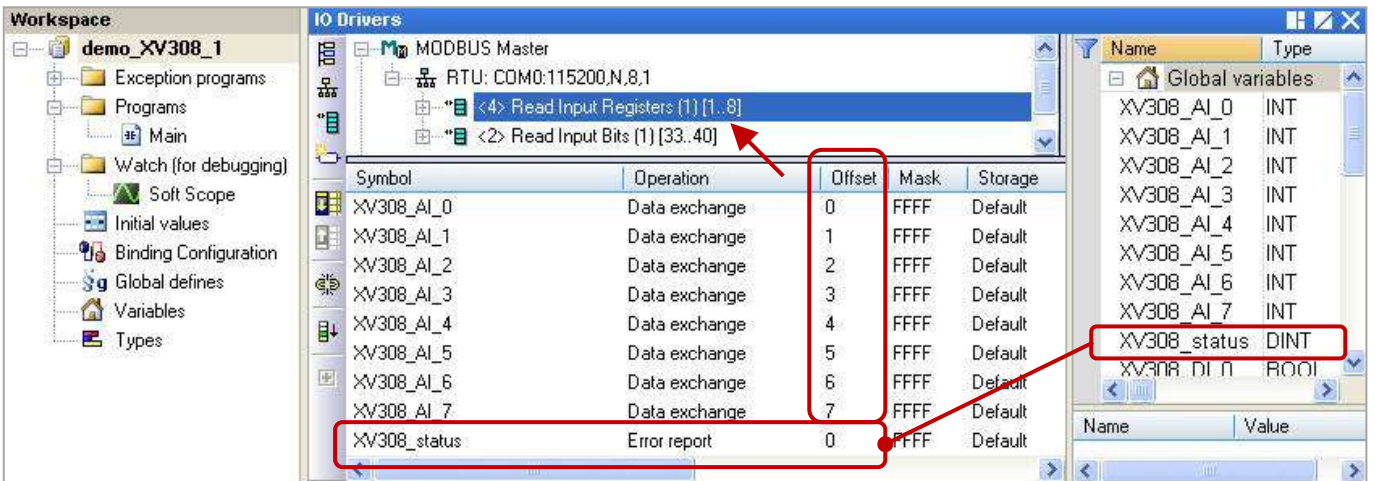
5.1.11 連接 XV308 (8 AI, 8 DIO)

XV308 是一款具有 8 AI 與 8 DIO (即 · DI + DO = 8) 的擴充卡 · 本章節說明的 Win-GRAF 範例程式為 "demo_XV308_1.zip", "demo_XV308_2.zip", "demo_XV308_3.zip" · 使用前請先參考 5.1.6 節的 XV Board 使用說明 · 並預先在 PAC 內使用 "DCON_UTILITY_Pro_CE_200.exe" 規劃 AI 通道。

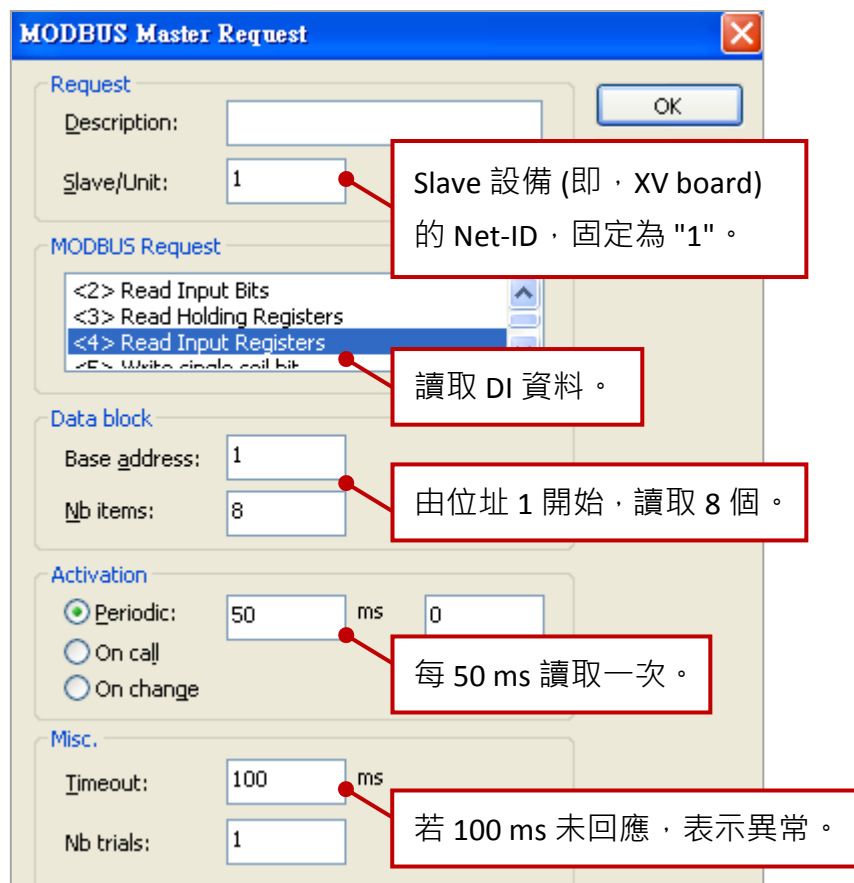
範例說明: (demo_XV308_1)

此範例建立了 2 個 Data Block · 一個用來讀取 8 個 AI 資料 · 另一個用來讀取 8 個 DI 資料。

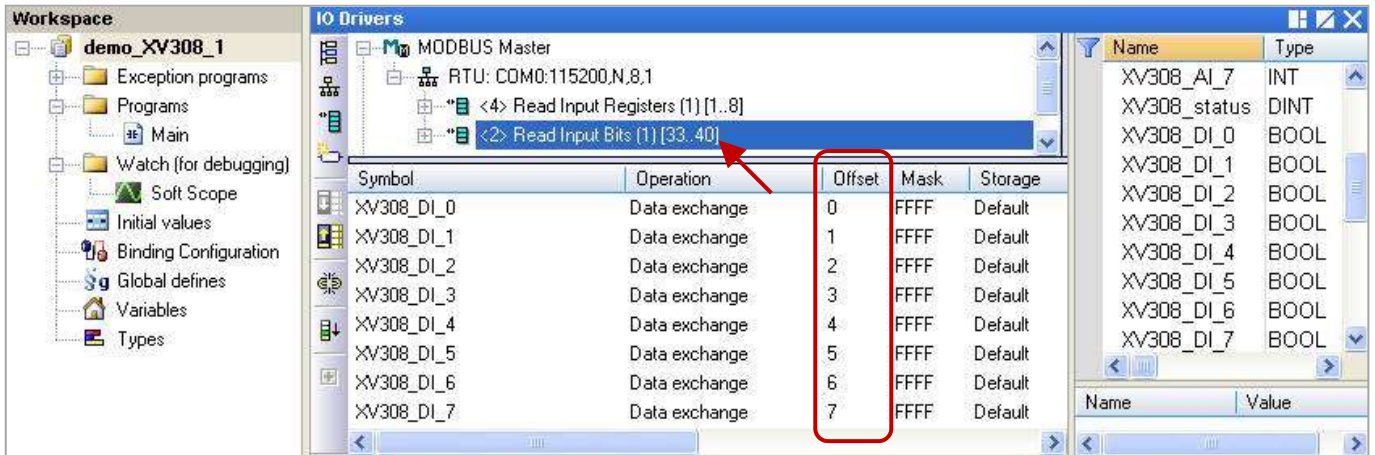
1. 滑鼠雙擊第 1 個 Data Block (即 · <4> Read Input Registers) 來開啟設定視窗。



注意: "Offset" 的值是由 "0" 開始 · 而 "Offset" 值加 1 (Base address) 才是該變數的 Modbus 位址 · 若 "Operation" 設定為 "Error report" · 該變數 (資料型態: DINT) 的 "Offset" 值需設定為 "0"。



2. 滑鼠雙擊第 2 個 Data Block (即 · <2> Read Input Bits) 來開啟設定視窗。



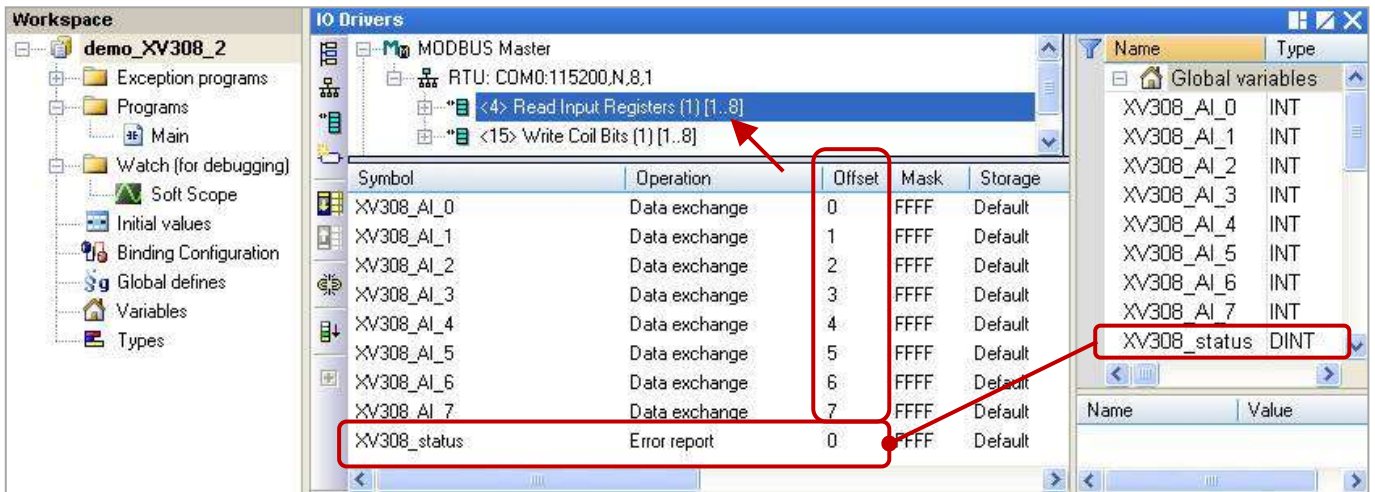
注意: 讀取 XV308 的 DI 資料時，位址必需由 "33" 開始。



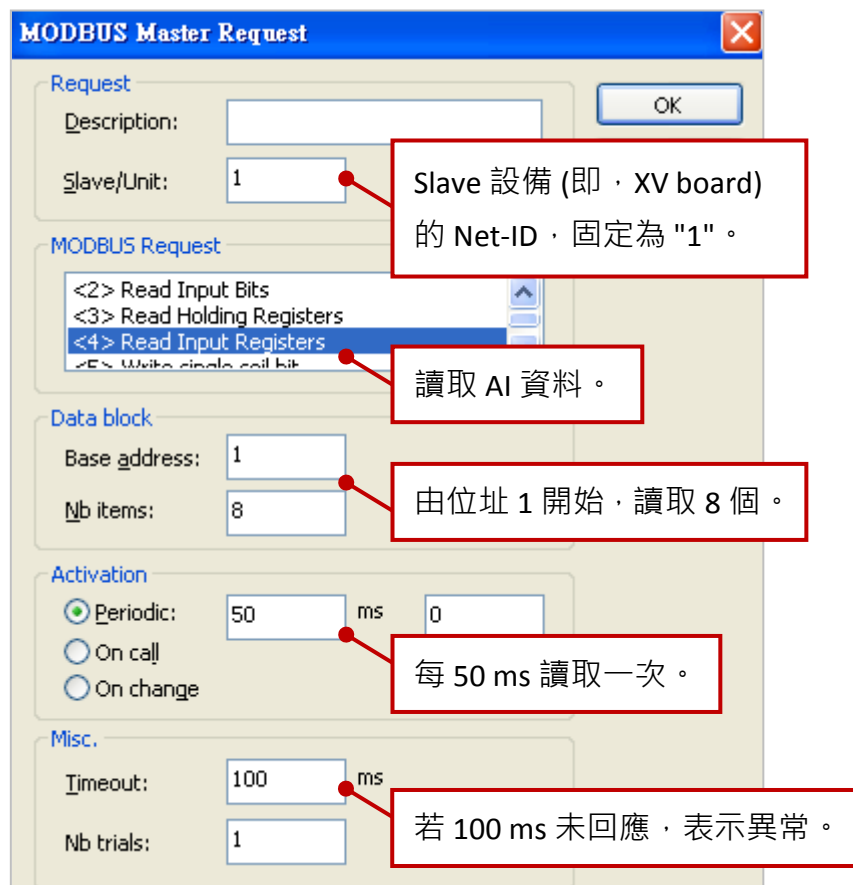
範例說明 : (demo_XV308_2)

此範例建立了 2 個 Data Block，一個用來讀取 8 個 AI 資料，另一個用來寫出 8 個 DO 資料。

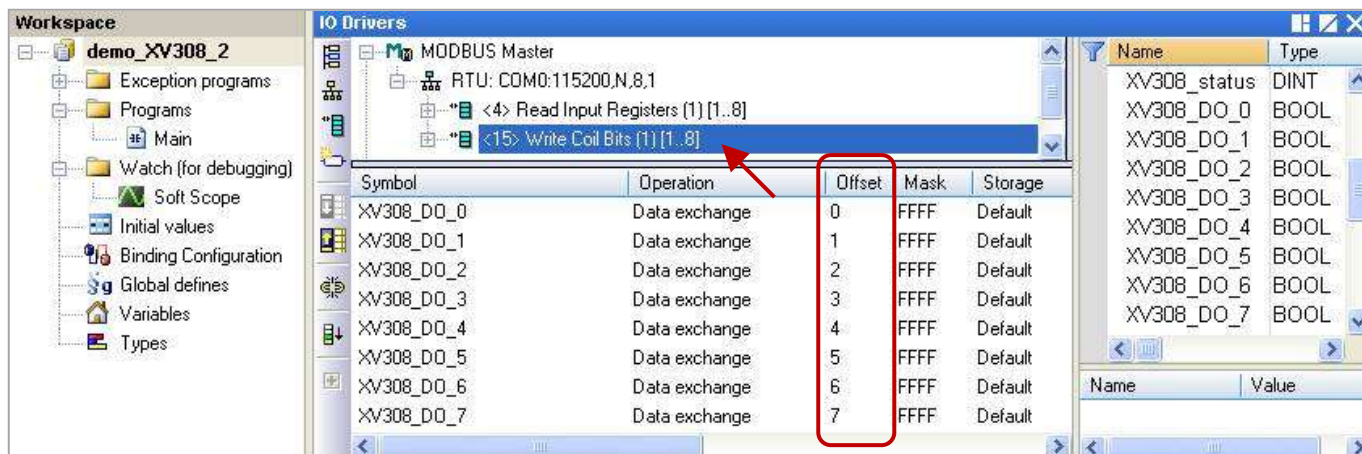
1. 滑鼠雙擊第 1 個 Data Block (即，<4> Read Input Registers) 來開啟設定視窗。



注意: “Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。若 “Operation” 設定為 “Error report”，該變數 (資料型態: DINT) 的 “Offset” 值需設定為 “0”。



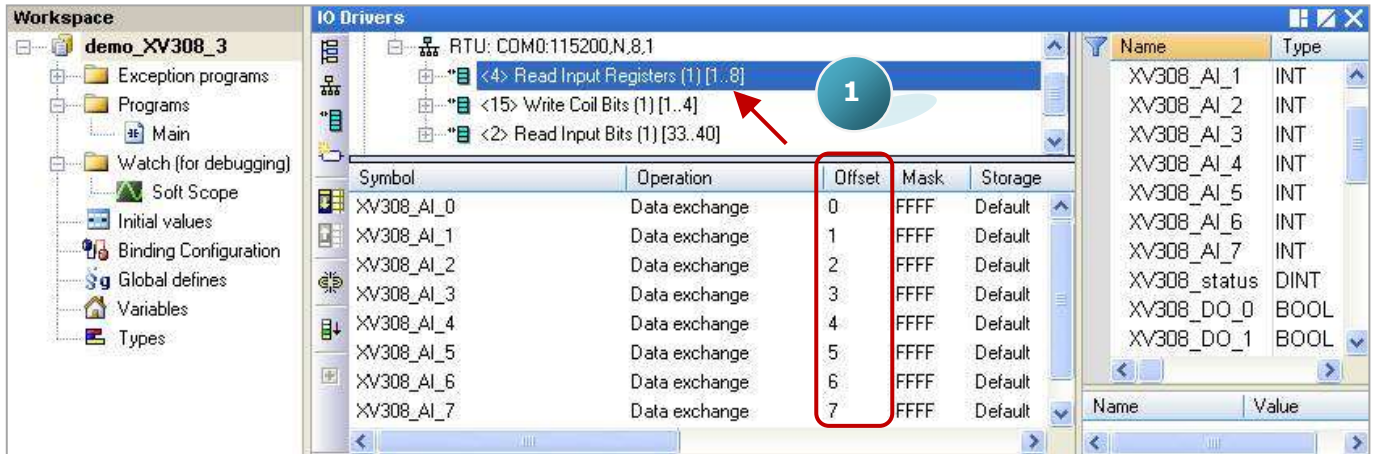
2. 滑鼠雙擊第 2 個 Data Block (即 · <15> Write Coil Bits) 來開啟設定視窗。



範例說明 : (demo_XV308_3)

此範例建立了 3 個 Data Block，第 1 個用來讀取 8 個 AI 資料，第 2 個用來寫出 4 個 DO 資料，第 3 個僅用來讀取 4 個 DI 資料。

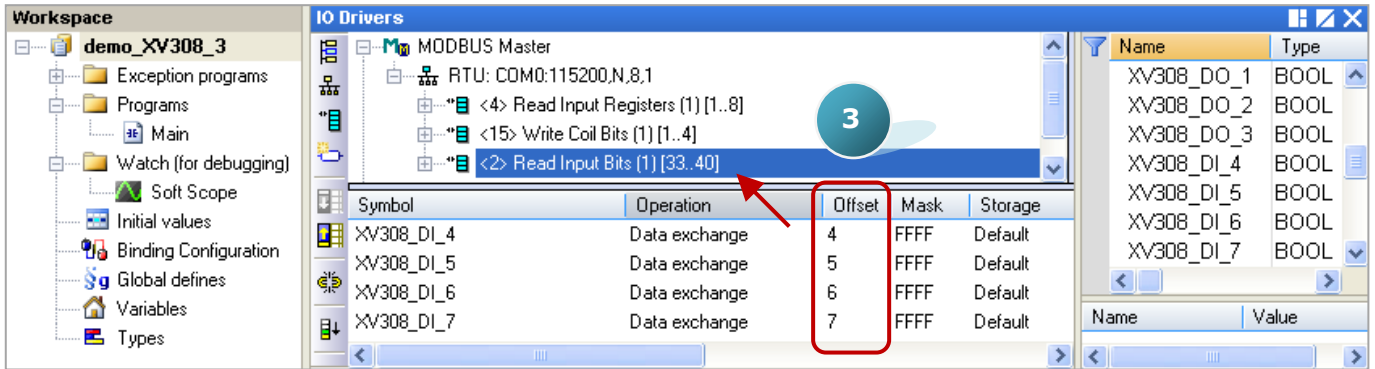
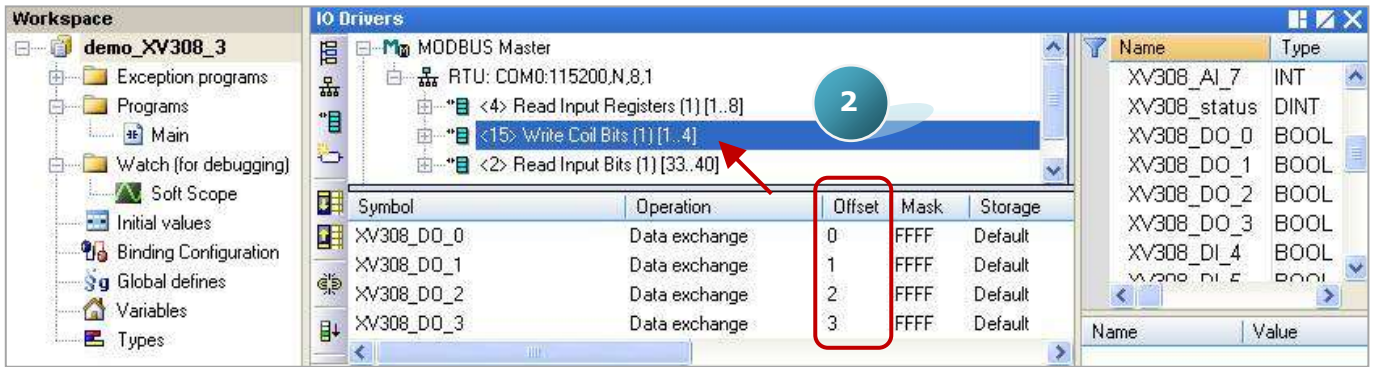
1. 滑鼠雙擊第 1 個 Data Block (即，<4> Read Input Registers) 來開啟設定視窗。



注意：“Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。



2. 如下圖，滑鼠雙擊第 2 個 Data Block (即，<15> Write Coil Bits) 來查看設定視窗。
3. 如下圖，滑鼠雙擊第 3 個 Data Block (即，<2> Read Input Bits) 來查看設定視窗。



2. <15> Write Coil Bits
寫出 4 個 DO。

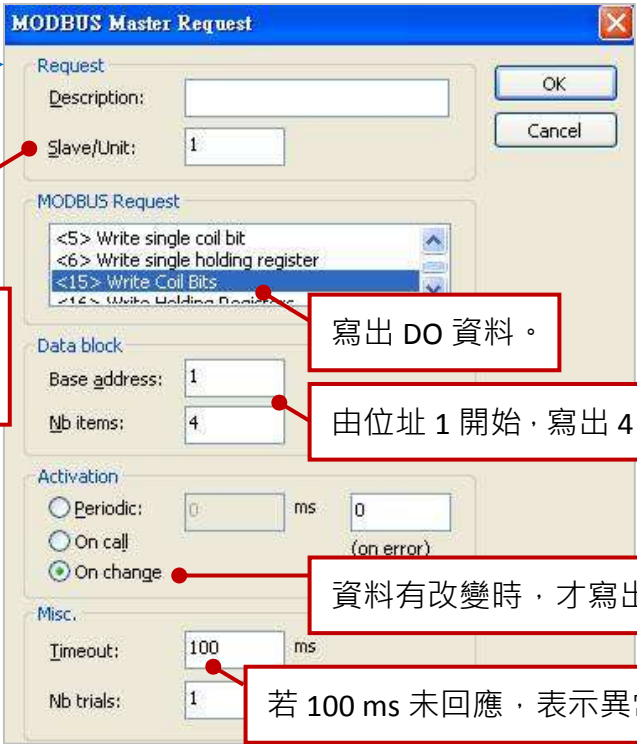
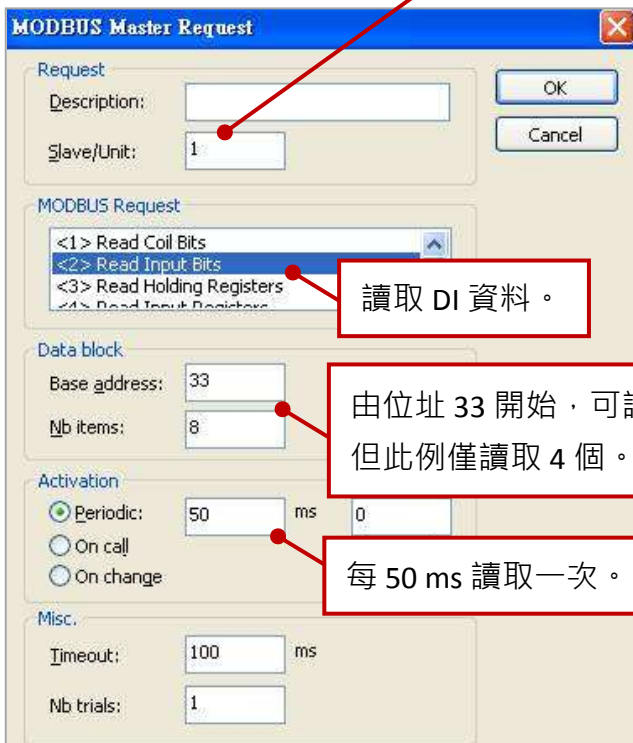
Slave 設備 (即 · XV board)
的 Net-ID · 固定為 "1"。

寫出 DO 資料。

由位址 1 開始 · 寫出 4 個。

資料有改變時 · 才寫出。

若 100 ms 未回應 · 表示異常。



3. <2> Read Input Bits · 讀取 4 個 DI。
注意: 讀取 XV308 的 DI 資料時 · 位址必需由 "33" 開始。

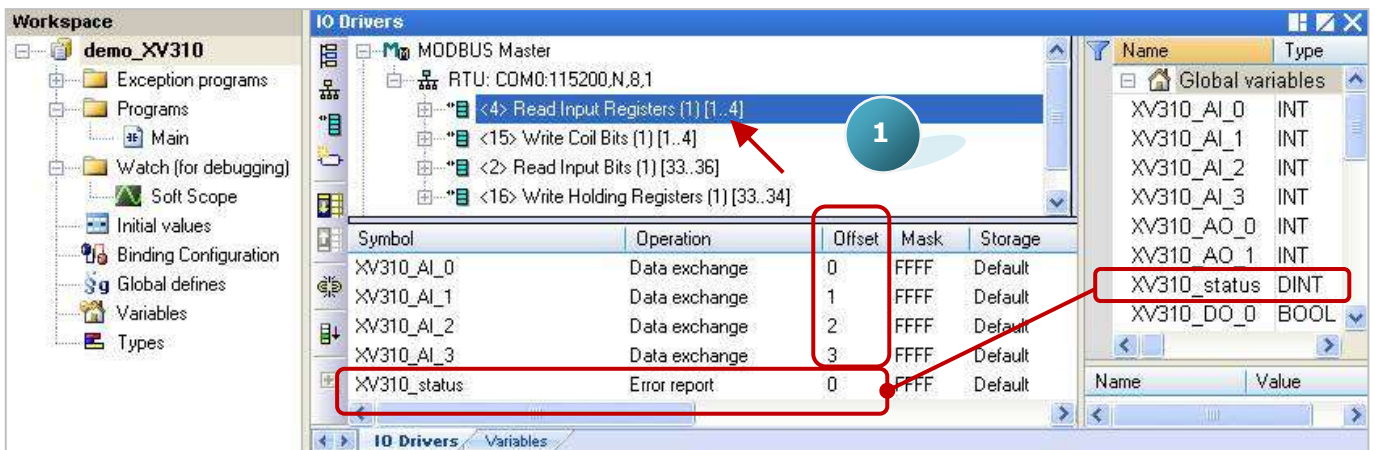
5.1.12 連接 XV310 (4 AI, 2 AO, 4 DI, 4 DO)

XV310 是一款具有 4 AI、2 AO、4 DI 與 4 DO 的擴充卡，本章節說明的 Win-GRAF 範例程式為 "demo_XV310.zip"，使用前請先參考 5.1.6 節的 XV Board 使用說明，並預先在 PAC 內使用 "DCON_Utility_Pro_CE_200.exe" 規劃 AI/AO 通道。

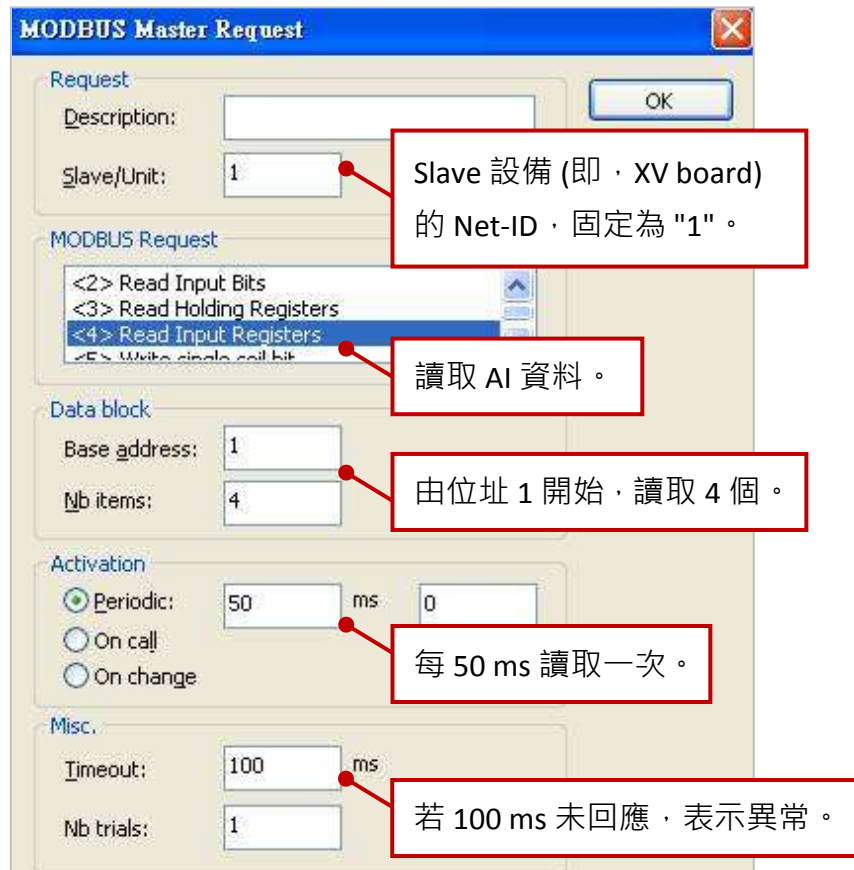
範例說明：

此範例建立了 4 個 Data Block，第 1 個用來讀取 4 個 AI 資料，第 2 個用來寫出 4 個 DO 資料，第 3 個用來讀取 4 個 DI 資料，第 4 個用來寫出 2 個 AO 資料。

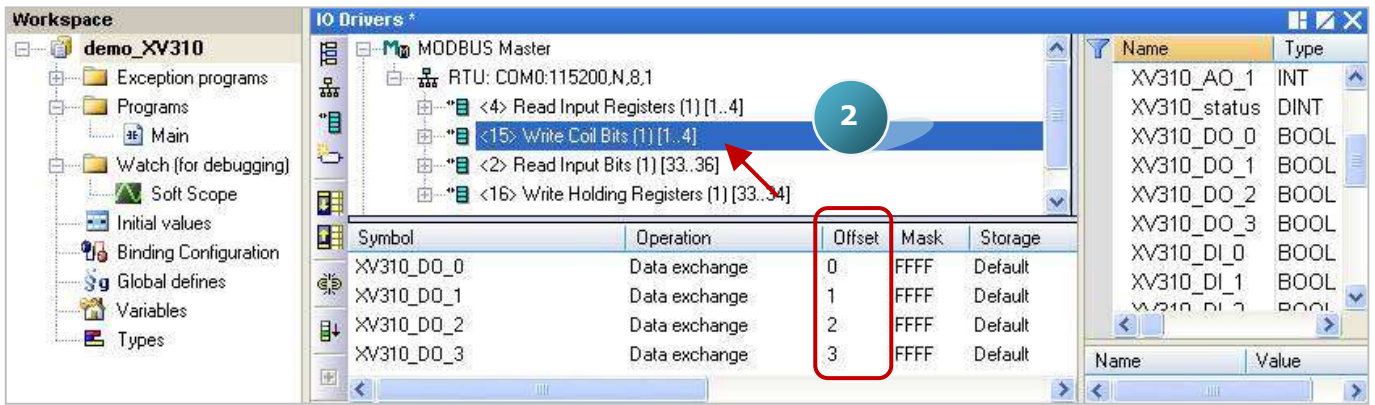
1. 滑鼠雙擊第 1 個 Data Block (即，<4> Read Input Registers) 來開啟設定視窗。



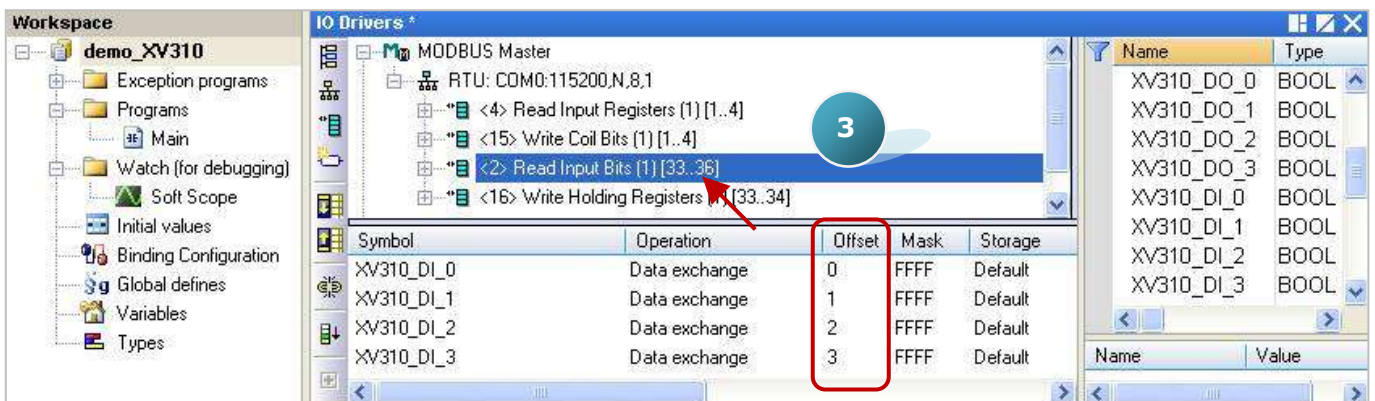
注意：“Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。若 “Operation” 設定為 “Error report”，該變數 (資料型態: DINT) 的 “Offset” 值需設定為 “0”。



2. 滑鼠雙擊第 2 個 Data Block (即 · <15> Write Coil Bits) 來查看設定視窗。



3. 滑鼠雙擊第 3 個 Data Block (即 · <2> Read Input Bits) 來查看設定視窗。





Slave 設備 (即 · XV board) 的 Net-ID · 固定為 "1" 。

讀取 DI 資料 。

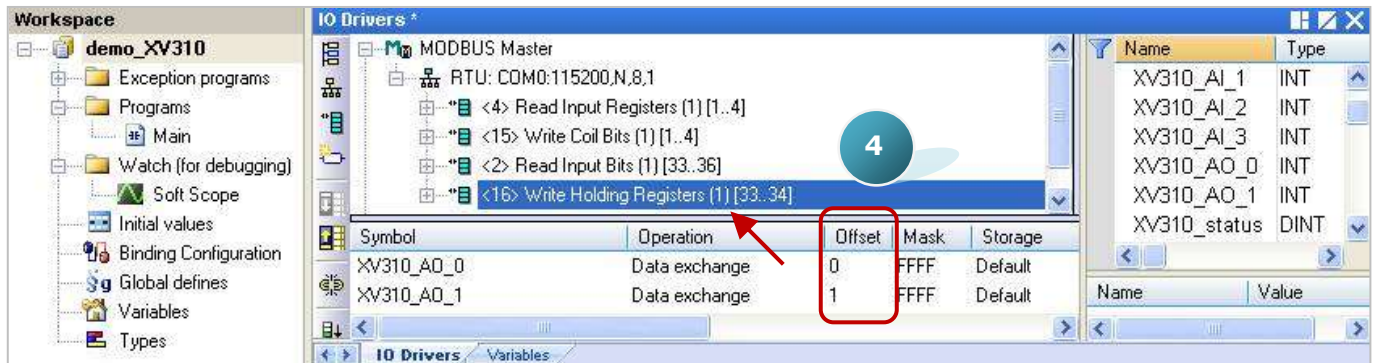
注意: 讀取 XV310 的 DI 資料時 · 位址必需由 "33" 開始 。

由位址 33 開始 · 讀取 4 個 。

每 50 ms 讀取一次 。

若 100 ms 未回應 · 表示異常 。

4. 滑鼠雙擊第 4 個 Data Block (即 · <16> Write Holding Registers) 。



Slave 設備 (即 · XV board) 的 Net-ID · 固定為 "1" 。

寫出 AO 資料 。

注意: 寫入 XV310 的 AO 資料時 · 位址必需由 "33" 開始 。

由位址 33 開始 · 寫出 2 個 。

資料有改變時 · 才寫出 。

若 100 ms 未回應 · 表示異常 。

5.1.13 停止/啟用 Modbus RTU/ASCII Master Port

在 Win-GRAF - "Fieldbus Configuration" 的 "IO Drivers" 畫面內啟用的 Modbus RTU/ASCII Master Port，都會在 PAC 開機後自動啟用，若想在程式運行中停用其中的某一個 Master Port 可以使用 "MBRTU_M_disable" 這個函式，如下：

```
(* To_disable 宣告為 BOOL *)
If To_disable then
  To_disable := FALSE ;
  MBRTU_M_disable (3) ;
End_if;
```

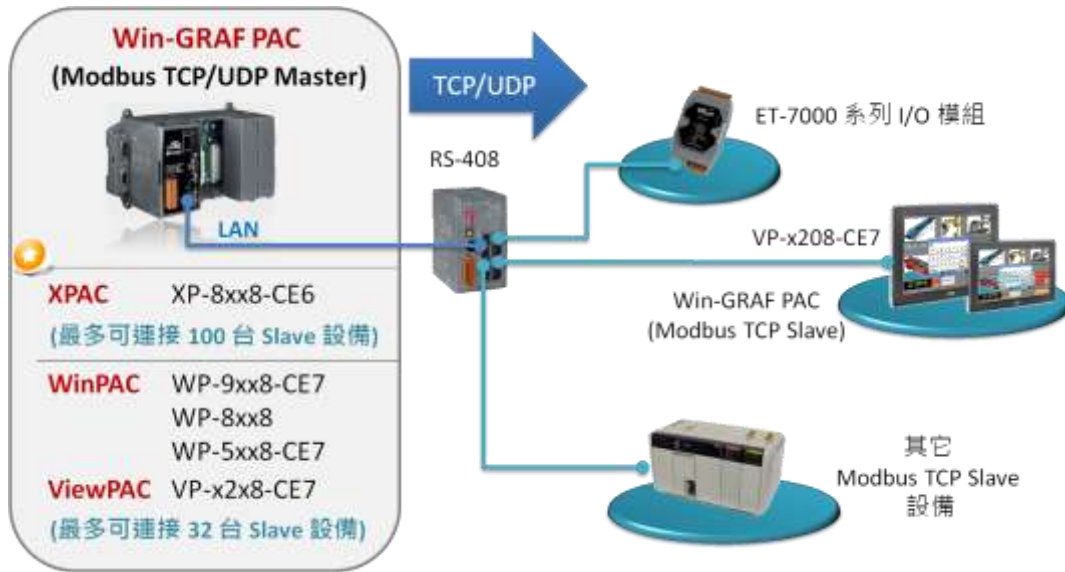
上方的程式碼中，當把 To_disable 設為 TRUE 時，會停用 COM3 這個 Modbus RTU/ASCII Master Port。之後若想再啟用它可以使用 "MBRTU_M_enable"，如下：

```
(* To_enable 宣告為 BOOL
   Status_com3 宣告為 BOOL *)
If To_enable then
  To_enable := FALSE ;
  MBRTU_M_enable (3) ;
End_if;
Status_com3 := MBRTU_M_status (3) ;
```

上方的 "MBRTU_M_status" 函式是用來取得 Modbus RTU/ASCII Master Port 的啟用 (True) 或 停用 (False) 狀態。

5.2 啟用 Win-GRAF PAC 為 Modbus TCP/UDP Master (Ethernet I/O)

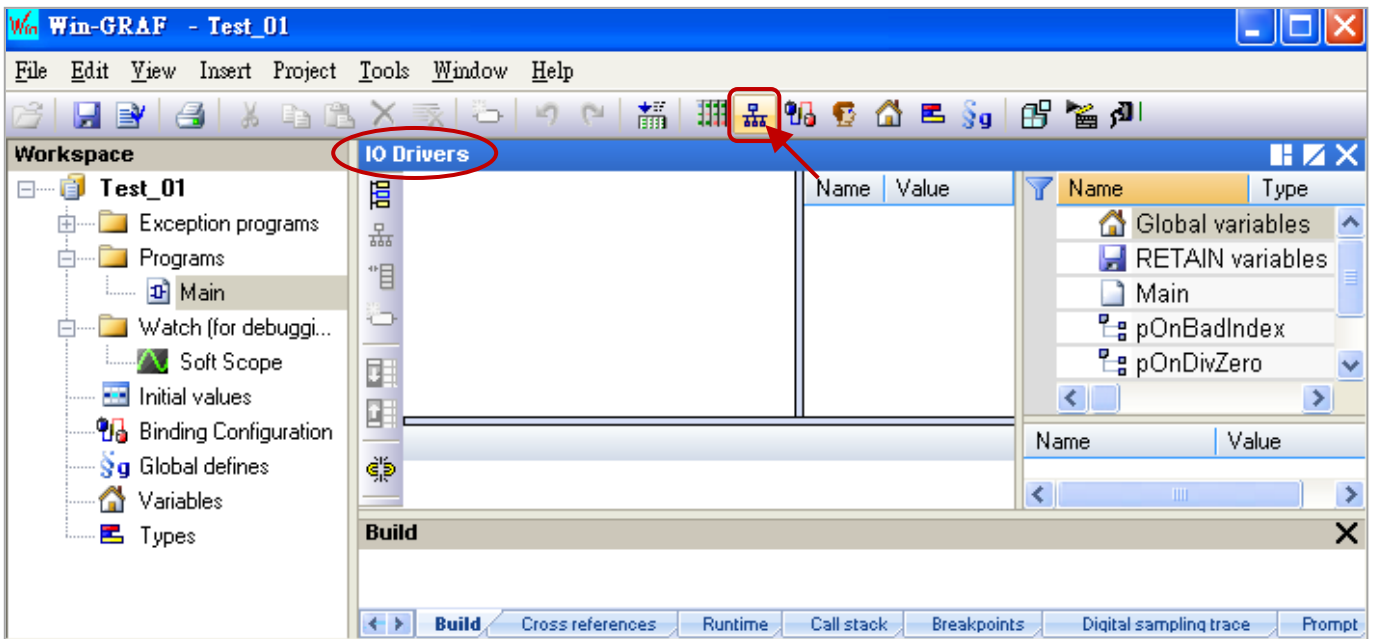
應用示意圖：



(您可參考 [P1-1](#)，來查詢詳細的 PAC 型號)

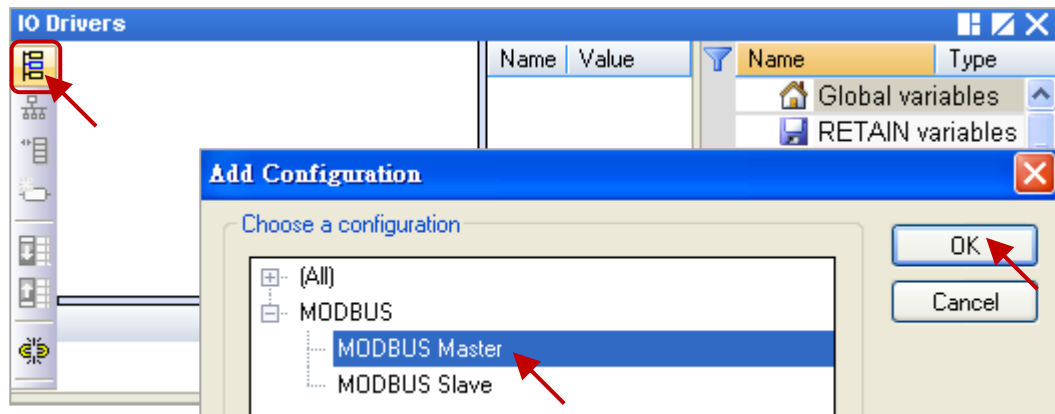
請參考以下操作步驟：

1. 滑鼠點選工具列上的“Open Fieldbus Configuration”按鈕來開啟“I/O Drivers”視窗。



2. 點選“I/O Drivers”視窗左側的“Insert Configuration”按鈕，再點選“MODBUS Master”並點選“OK”來啟用一個 Modbus Master 設定。

註：一個“Modbus Master”可有多個 Port 設定 (參考下一步驟)，可設定為 Modbus Master RTU/ASCII Port (參考 [5.1 節](#)) 或是 Modbus Master TCP/UDP Port，也可設定是否啟用該設定。



3. 點選左側的“Insert Master/Port” 按鈕，開啟設定視窗並選擇“MODBUS on Ethernet” 選項。
設定以下項目後，再點選“OK”。

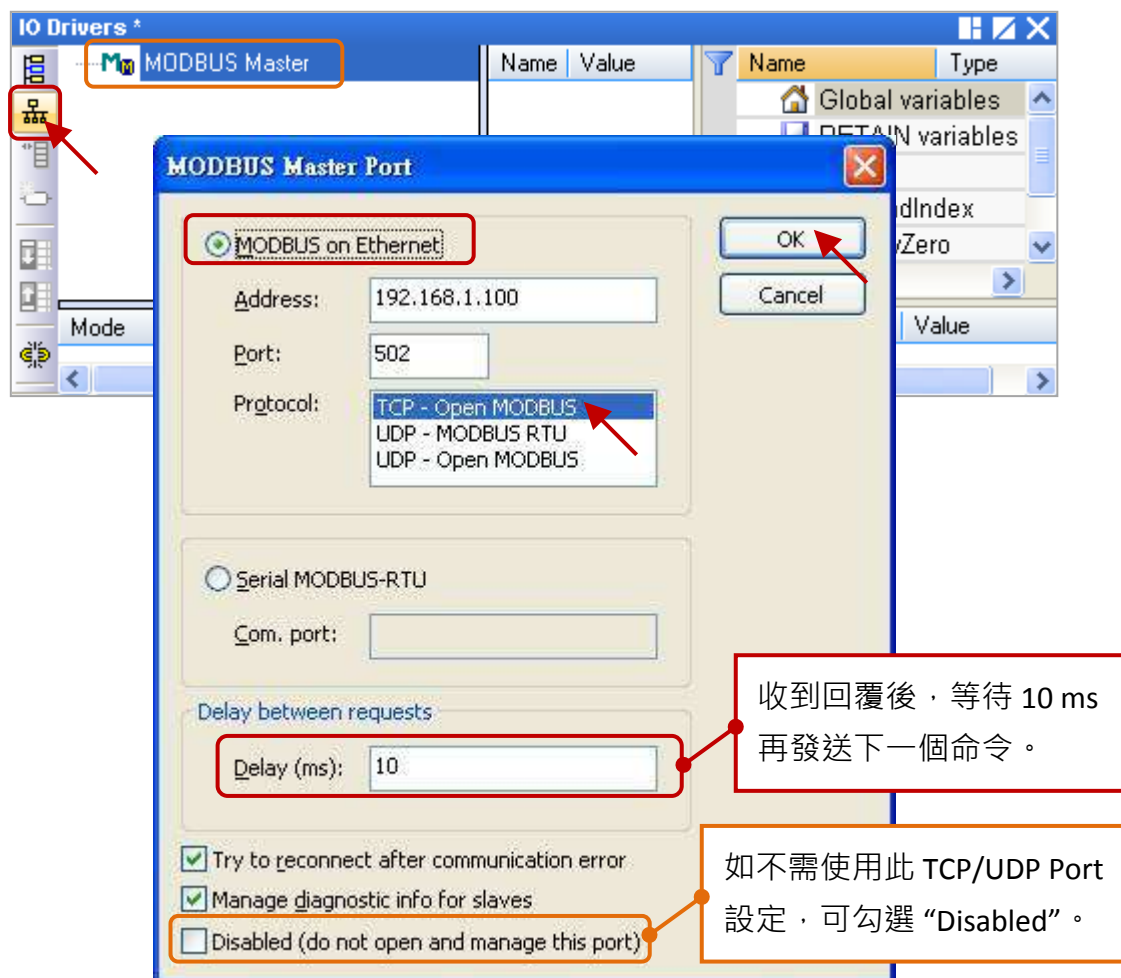
Address: 填入 Modbus TCP Slave 設備的 IP 位址。(例如: “192.168.1.100”)

Port: 填入 Modbus TCP Slave 設備的 Port 編號，一般是“502”。

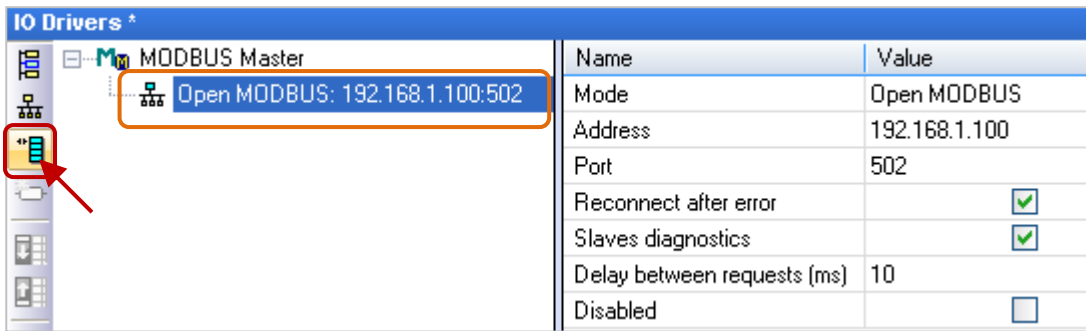
Protocol: 若為 Modbus TCP Master，請選擇“TCP – Open MODBUS”。

若為 Modbus UDP Master，請選擇“UDP – Open MODBUS”。

Delay: 填入命令的間隔時間(例如: 10 ms，可設為 0 ~ 10000)。



4. 點選左側的 “Insert Slave/Data Block” 按鈕，來建立一個 Data Block。



讀取 AI 資料

5. 於 “MODBUS Master Request” 設定視窗中，設定以下項目並於完成後按 “OK”。



a. Slave/Unit:

填入 Slave 設備的站號 (Net-ID，通常為 “1”)。

b. MODBUS Request: 此例，

選擇 “<4> Read Input Registers”。

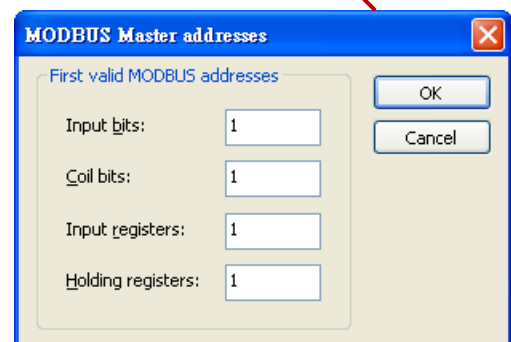
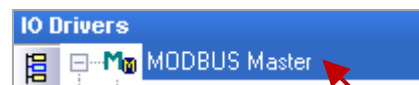
c. Base address:

預設值由 1 開始。

Nb items:

讀取 AI 的數量 (此例為 4)。

註: 如需修改 “Base address”，可使用滑鼠右鍵點選 “MODBUS Master” 再選擇 “MODBUS Master Addresses” 修改其值。



d. Activation: 表示 Modbus Request 發送的方式。

Periodic: 表示週期性的發送，此例為每 1 秒發送一次。“on error” 表示每當發生異常時，下一次的發送時間 (此例為 15 秒)。

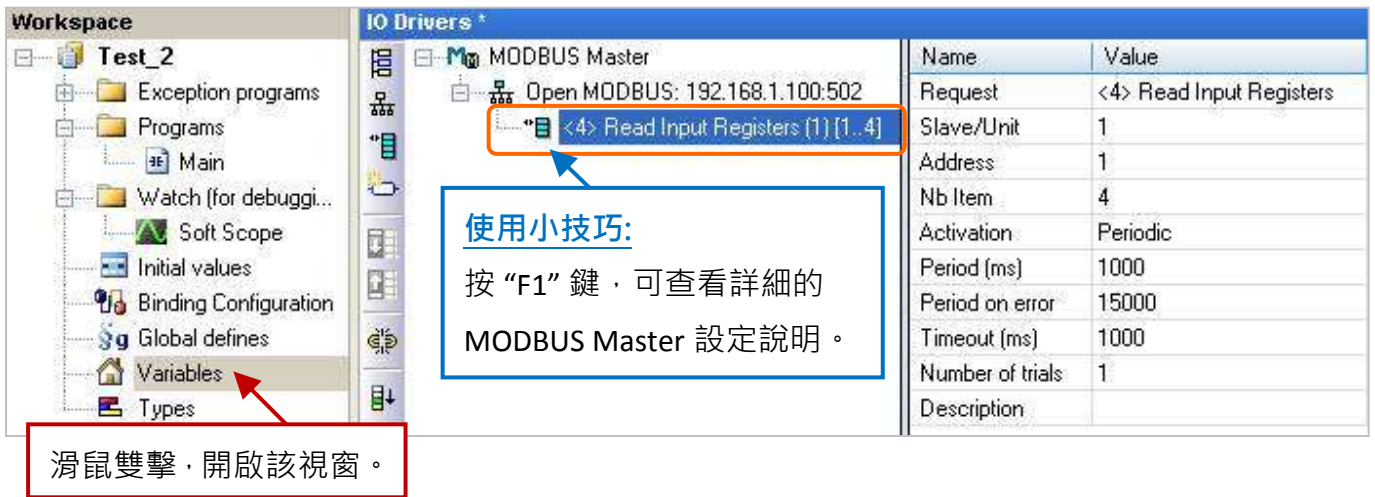
On call: 表示程式有呼叫時，才進行發送一次。

On change: 表示寫出的資料有改變時，才進行發送一次。

e. Timeout: 設定多久未回應，即表示異常。

(對於 Modbus TCP/UDP 建議值: 1000 ~ 3000 ms ; 此例為 1000 ms)

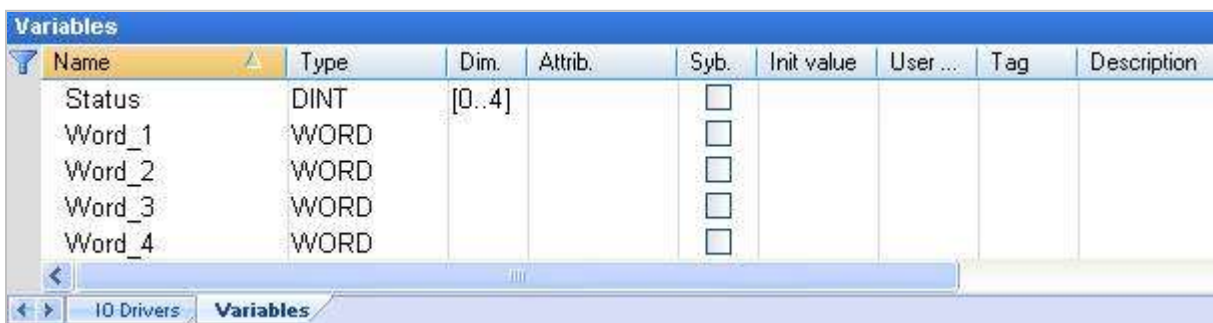
6. 接著，請開啟 “Variables” 視窗，設定需使用的變數。



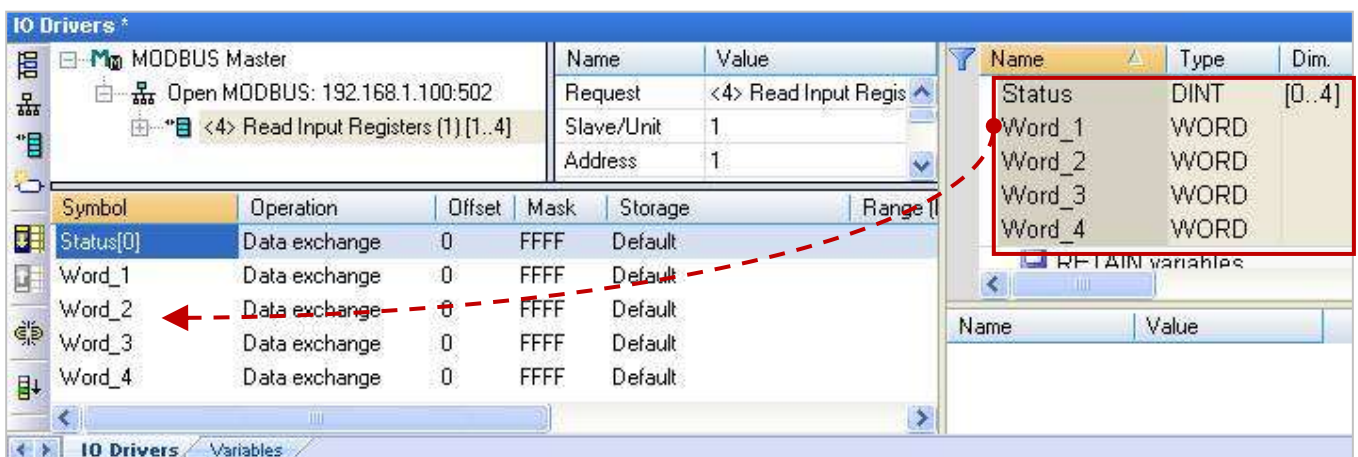
此處需設定 4 個 Word (16 bit) 變數 (設定方式可參考 [2.3.1 節](#))，請依照下表來設定。

變數名稱	資料型態	Dim.	說明
Word_1 ~ Word_4	WORD	---	用來讀取 AI 資料 (16 bit)。
Status	DINT	5	用來記錄資料的存取狀況。

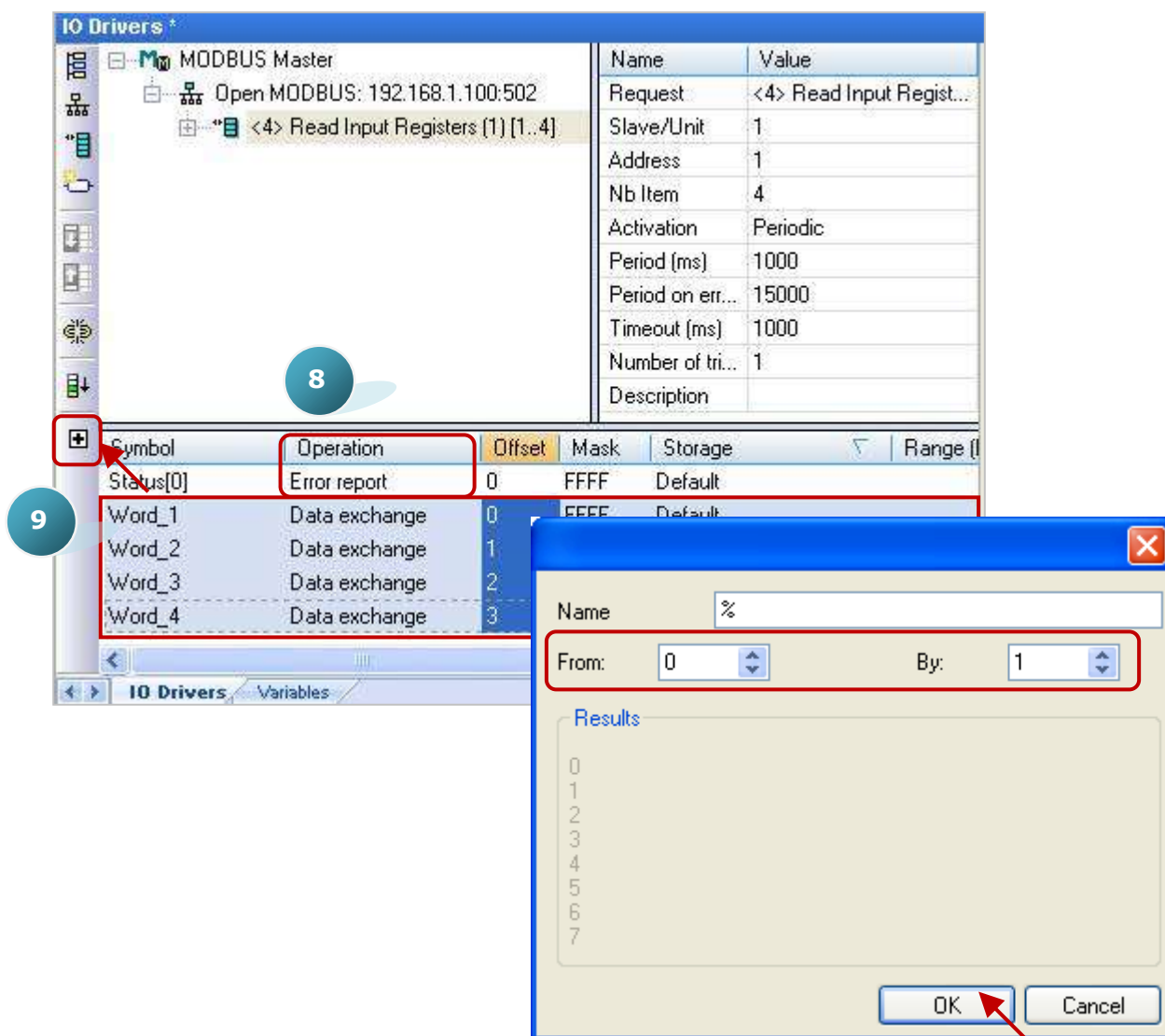
您可參考 [附錄 A](#) 來了解變數的資料形態與範圍，設定完成後，畫面如下。



7. 於 “I/O Drivers” 視窗，請將變數區中的變數 (“Word_1 ~ Word_4” 與 “Status”) 拖曳到 Data Block 的 “Symbol” 區域。**注意:** “Status” 是一個陣列變數，拖曳到 “Symbol” 區域會是 “Status[0] ~ Status[4]”，請按 “Delete” 鍵刪除 “Status[1] ~ [4]”。



8. 設定 “Status[0]” 的 “Operation” 為 “Error report” (表示讀取失敗時，該變數值為一個 “Error Code”，讀取成功時則會重置為 “0”)，按 “F1” 鍵則可查看 Modbus Master 設定說明，於標題 “Status and command variables” 中有詳細的命令、“Error Code” 說明。
9. 選取 “Word_1 ~ Word_4” 並點選 “Iterate property” 設定 Offset 值 (From: 0 ; By: 1)。



無論是 Modbus Master RTU/ASCII Port (參考 5.1 節) 或是 Modbus Master TCP/UDP Port，設定 “Modbus Master Request” 的方式是相同的，以上已完成了讀取 AI 資料的設定，其它的讀/寫方式可點選項目連結，來參考 5.1.1 ~ 5.1.5 節的範例內容。

項目	Function Code	Modbus Request	說明
1	2	Read Input Bits	讀取 DI 資料
2	5	Write single coil bit	寫出 DO 資料
3	4	Read Input Registers	讀取 AI 資料
4	6	Write single holding register	寫出 AO 資料 (16-bit)
5	16	Write Holding Registers	寫出 AO 資料 (16/32 bit)

註: 若想在程式運行過程中停用 (Disable) Modbus TCP/UDP Master Port，請參考 [第 5.2.4 節](#) 來使用 "MBTCP_M_disable" 函式 (UDP 則使用 "MBUDP_M_disable" 函式)。

5.2.1 連接 ET-7000 系列 I/O 模組

ICP DAS ET-7000 系列模組為一款支援 Modbus TCP Slave 通訊協定的 I/O 模組，Win-GRAF PAC 可啟用 Modbus TCP Master 來連接多個 ET-7000 模組。而不同的 PAC 型號，其建議可連線使用的 "最多 ET-7000 模組數量" 也會有差別，比如 XP-8xx8-CE6, WP-5238-CE7 建議最多不超過 200 個，VP-x2x8-CE7 則建議不超過 32 個。

關於更多 ET-7000 的產品訊息請參訪:

http://www.icpdas.com/root/product/solutions/remote_io/ethernet_io/ethernet_io_selection.html

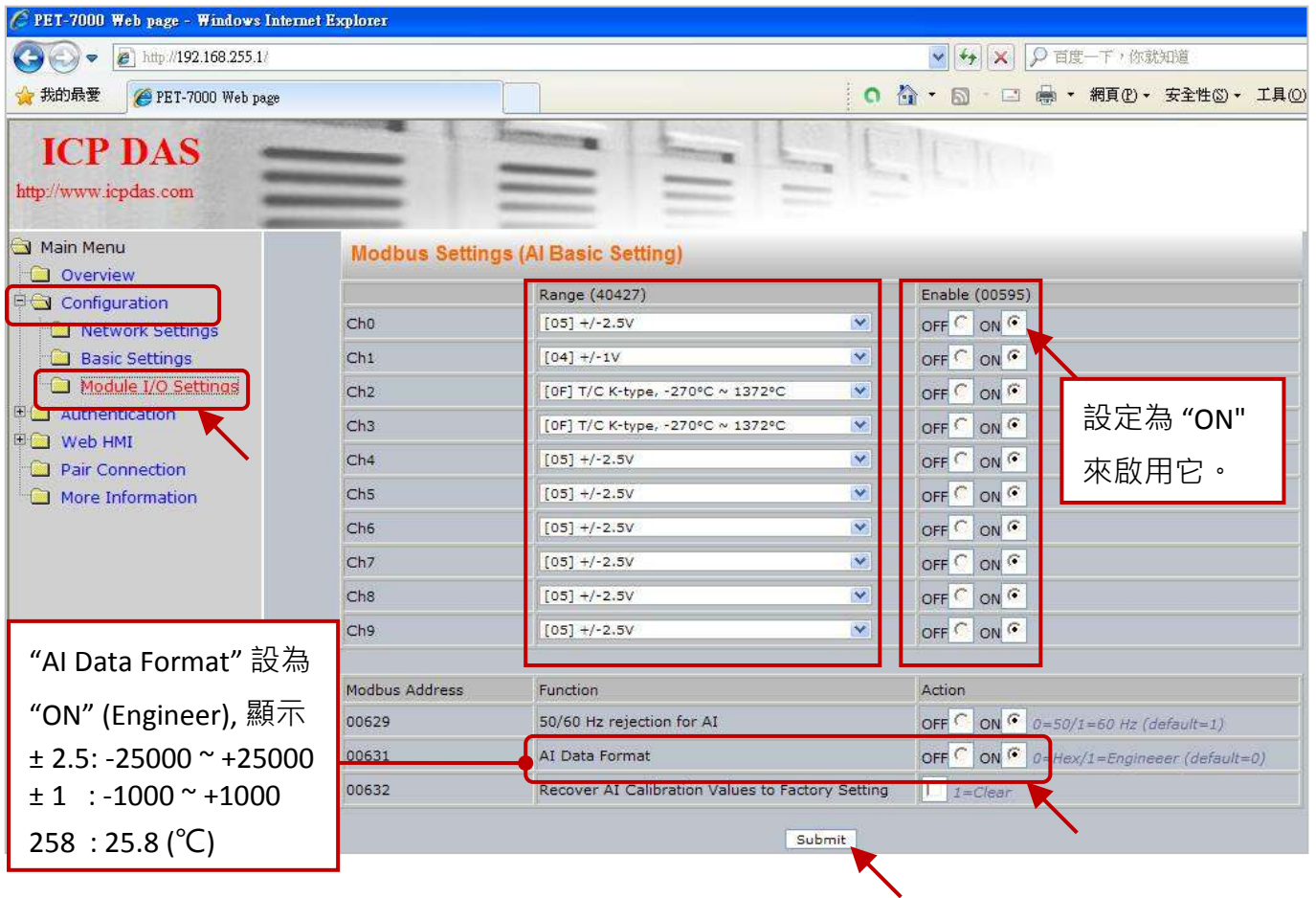
使用 Internet Browser 設定 ET-7000 模組

初次使用 ET-7000 前，需先使用 Internet Browser 對該 ET-7000 進行設定。出廠時，ET-7000 的 IP 位址為 192.168.255.1，Mask = 255.255.0.0，請先將您的 PC 的 IP 設定在同一個網段內 (例如，將 IP 設成 192.168.255.100，Mask = 255.255.0.0)，然後開啟瀏覽器 (例如: IE)，輸入該 ET-7000 的 IP 來連上它。

注意: ET-7000 後方的指撥開關 (Dip Switch) 必須保持在 "Normal" 位置。



如下，在 Configuration > Module I/O Settings 內設定好相關的通道設定，再按下“Submit”。



使用者可設定 ET-7018Z 的“AI Data Format”為“ON” (Engineering) 較方便使用。例如:

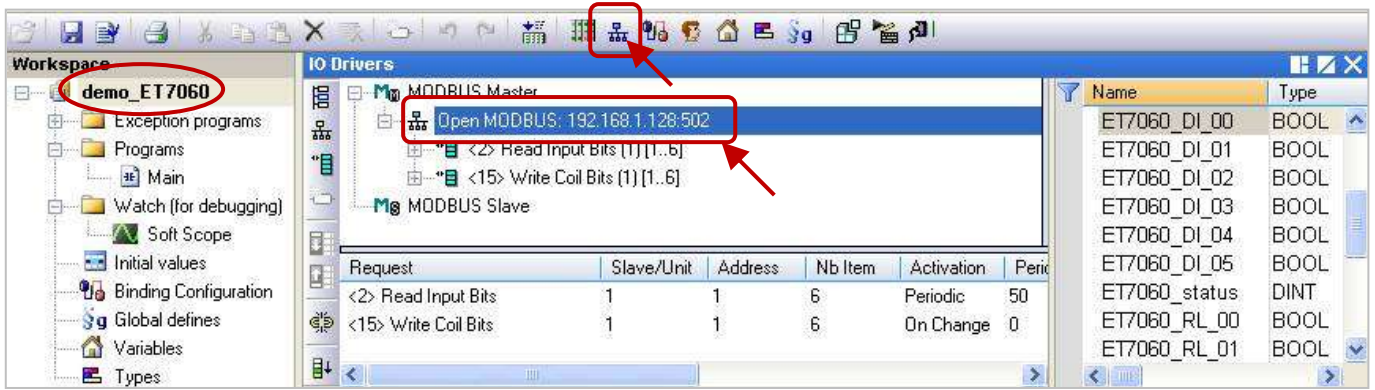
Type Code	Range	Data Format	Minimum	Maximum
04	-1 ~ +1 V	Engineering	-10000	+10000
		2's comp HEX	8000h	7FFFh
05	-2.5 ~ +2.5 V	Engineering	-25000	+25000
		2's comp HEX	8000h	7FFFh
18	Type M Thermocouple -200 ~ 100°C	Engineering	-20000	+10000
		2's comp HEX	8000h	4000h

回存/開啟範例程式:

以下章節所提供的 Win-GRAF 範例程式，可在出貨光碟中找到，請參考 [第 12 章](#)，點選功能表 "File" > "Add Existing Project" > "From Zip"，來回存範例程式並查看詳細的程式內容。
(CD-ROM:\Napdos\Win-GRAF\demo-project\)

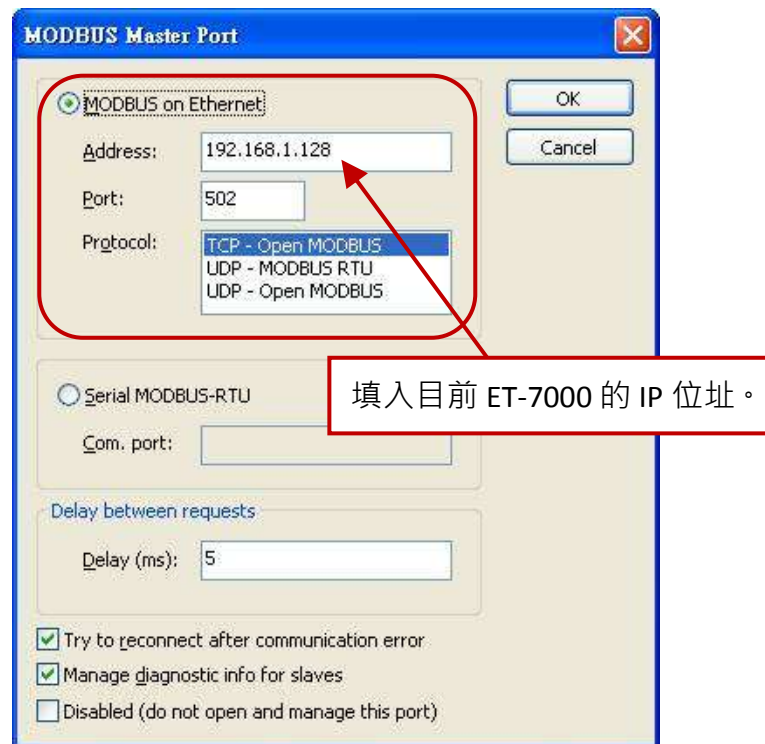
範例	檔名	說明
ET-7060	demo_ET7060.zip	讀取 6 DI，寫出 6 DO
ET-7018Z	demo_ET7018z.zip	讀取 10 AI

1. 滑鼠點選工具列上的“Open Fieldbus Configuration” 按鈕來開啟 “I/O Drivers” 視窗。



2. 滑鼠雙擊 “Open Modbus: IP:502” 可開啟 “MODBUS Master Port” 視窗。

注意: 此章節的範例，皆啟用 Win-GRAF PAC 為 Modbus TCP Master，請填入目前 ET-7000 的 IP 位址，"Port" 一般是 "502"，"Protocol" 選擇 "TCP - Open Modbus"。



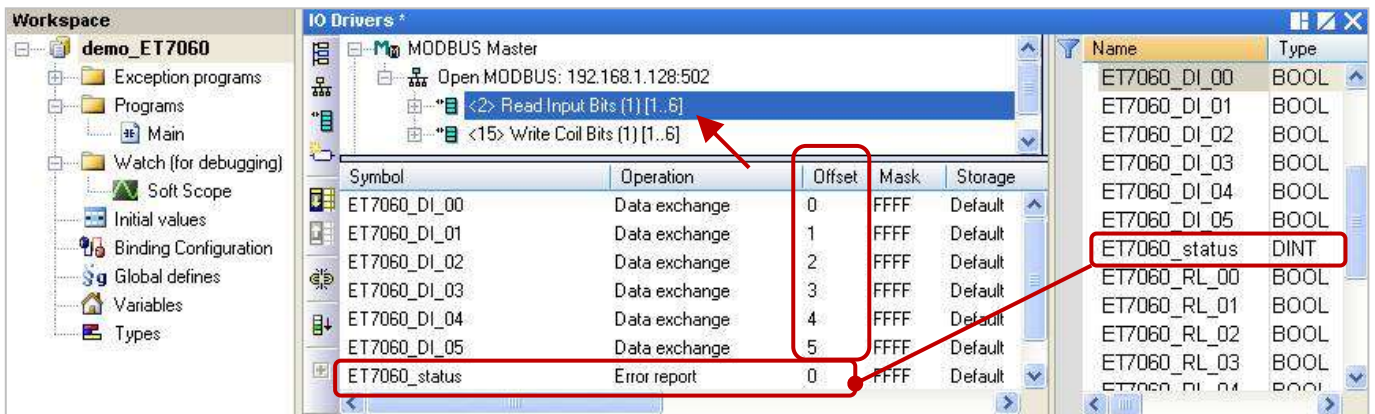
5.2.2 連接 ET-7060 (6 DI, 6 Relay)

ET-7060 是一款具有 6 DI 與 6 Relay 的 Ethernet I/O 模組，本章節說明的 Win-GRAF 範例程式為 "demo_ET7060.zip"，使用前請先參考 5.2.1 節 使用 Internet Browser 對該模組進行各通道設定，並回存 與 開啟此範例程式。

範例說明:

此範例建立了 2 個 Data Block，一個用來讀取 6 個 DI 資料，另一個用來寫出 6 個 DO 資料。

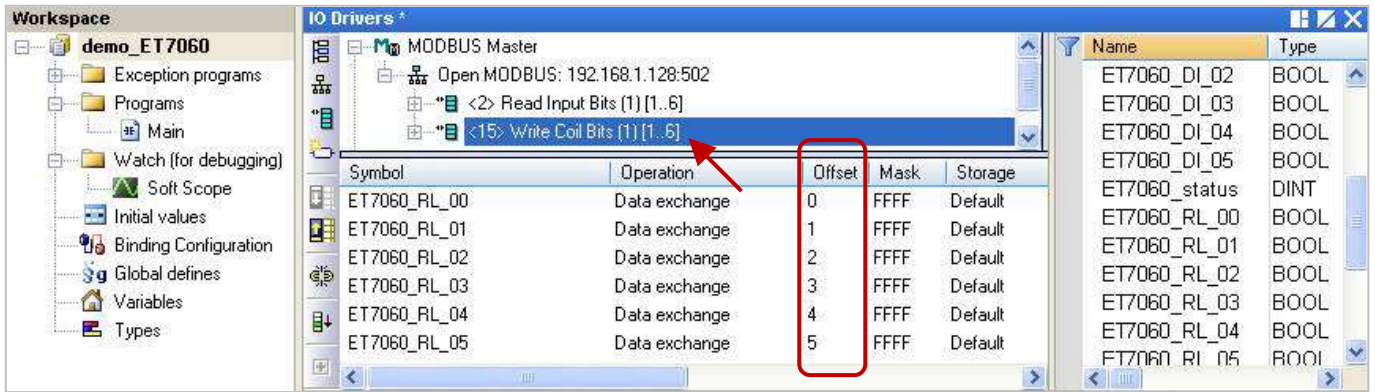
1. 滑鼠雙擊第 1 個 Data Block (即，<2> Read Input Bits) 來開啟設定視窗。



- 注意:**
1. "Offset" 的值是由 "0" 開始，而 "Offset" 值加 1 (Base address) 才是該變數的 Modbus 位址。
 2. 若 "Operation" 設定為 "Error report"，該變數 (資料型態: DINT) 的 "Offset" 值需設定為 "0"。



2. 滑鼠雙擊第 2 個 Data Block (即 · <15> Write Coil Bits) 來開啟設定視窗。

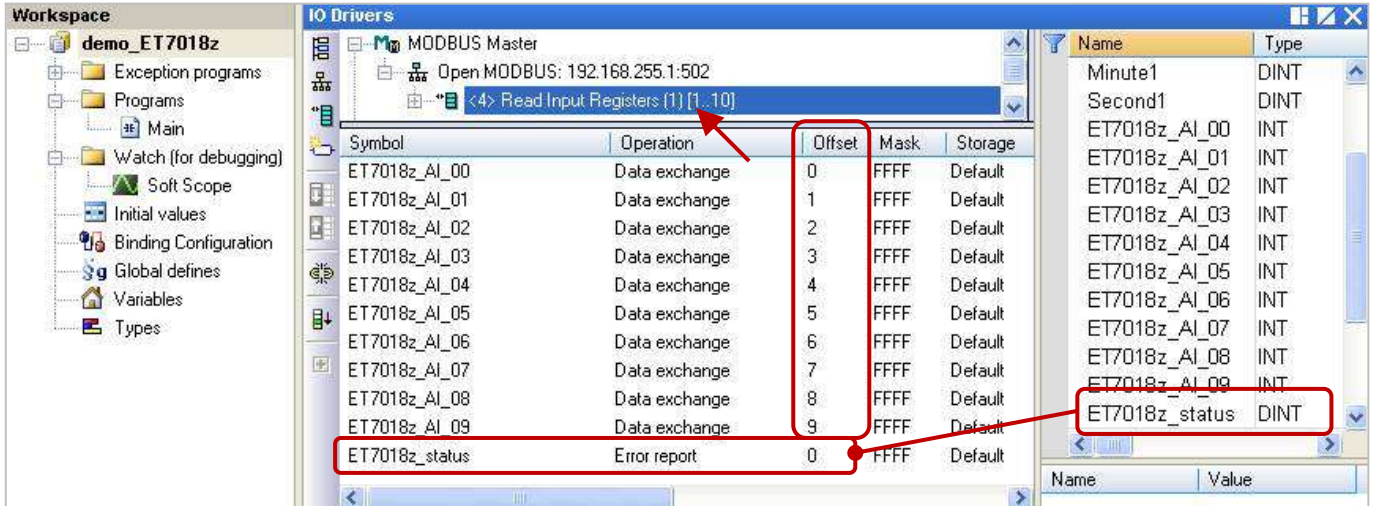


5.2.3 連接 ET-7018Z (10 AI)

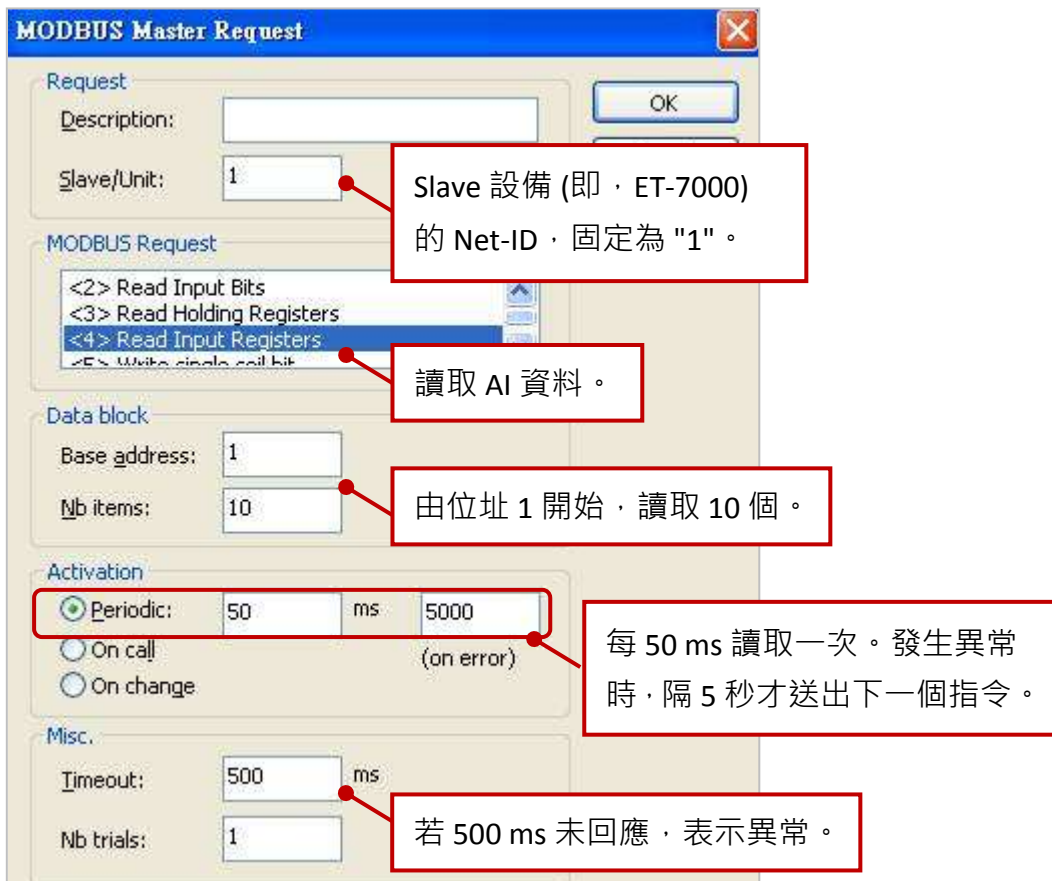
ET-7018Z 是一款具有 10 AI 的 Ethernet I/O 模組，本章節說明的 Win-GRAF 範例程式為 "demo_ET7018z.zip"，使用前請先參考 [5.2.1 節](#) 使用 Internet Browser 對該模組進行各通道設定，並回存與開啟此範例程式。

範例說明：此範例建立了 1 個 Data Block，用來讀取 10 個 AI 資料。

1. 滑鼠雙擊第 1 個 Data Block (即，<4> Read Input Registers) 來開啟設定視窗。



- 注意：**
1. "Offset" 的值是由 "0" 開始，而 "Offset" 值加 1 (Base address) 才是該變數的 Modbus 位址。
 2. 若 "Operation" 設定為 "Error report"，該變數 (資料型態: DINT) 的 "Offset" 值需設定為 "0"。
 3. AI 數值範圍為 -32768 ~ 32767，請宣告變數的資料型態為 "INT"。



5.2.4 停止/啟用 Modbus TCP/UDP Master Port

在 Win-GRAF - "Fieldbus Configuration" 的 "IO Drivers" 畫面內啟用的 Modbus TCP/UDP Master Port，都會在 PAC 開機後自動啟用。若想在程式運行中停用其中的某一個 Modbus TCP Master Port 可以使用 "MBTCP_M_disable" 這個函式 (UDP 則使用 "MBUDP_M_disable" 函式)，如下：

```
(* To_disable 宣告為 BOOL *)
If To_disable then
  To_disable := FALSE ;
  MBTCP_M_disable ( '192.168.71.9' , 502 ) ;
End_if;
```

上方的程式碼中，當把 To_disable 設為 TRUE 時，會把連到 192.168.71.9 (TCP Port_No = 502) 這個 Slave Device 的 Modbus TCP Master Port 給停用。之後若想再啟用它可以使用 "MBTCP_M_enable" (UDP 則使用 "MBUDP_M_enable" 函式)，如下：

```
(* To_enable 宣告為 BOOL
   Status_tcp 宣告為 BOOL *)
If To_enable then
  To_enable := FALSE ;
  MBTCP_M_enable ( '192.168.71.9' , 502 ) ;
End_if;
Status_tcp := MBTCP_M_status ( '192.168.71.9' , 502 ) ;
```

上方的 "MBTCP_M_status" 函式 (UDP 則使用 "MBUDP_M_status" 函式)，是用來取得該 Modbus TCP Master 的啟用 (True) 或 停用 (False) 狀態。

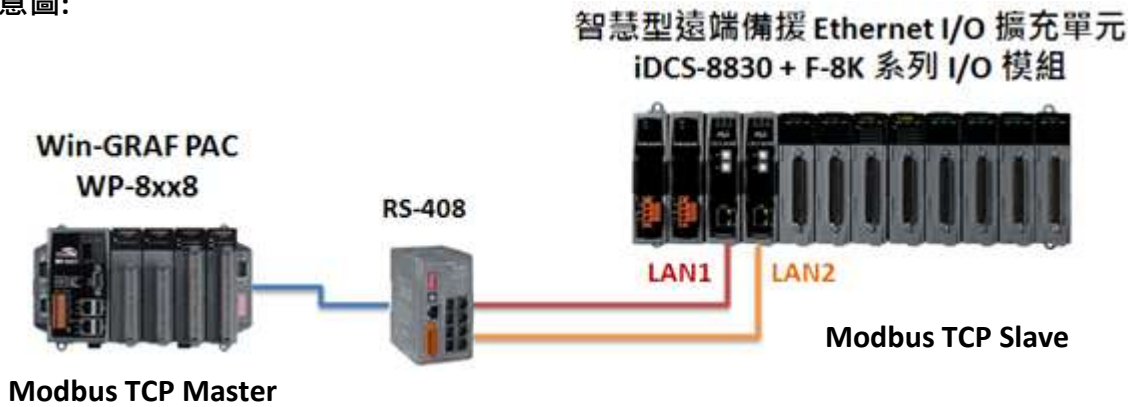
為了方便維護，可以宣告一個 STRING 變數 (長度設為 20)，比如 IP_addr2 來設定 192.168.71.9 為它的初值。這樣使用函式時比較不會出錯，如下：

```
If To_disable then
  To_disable := FALSE ;
  MBTCP_M_disable ( IP_addr2 , 502 ) ;
End_if;
Status_tcp2 := MBTCP_M_status ( IP_addr2 , 502 ) ;
```

5.3 連接具有 2 個 IP 位址的 Modbus TCP Slave 設備

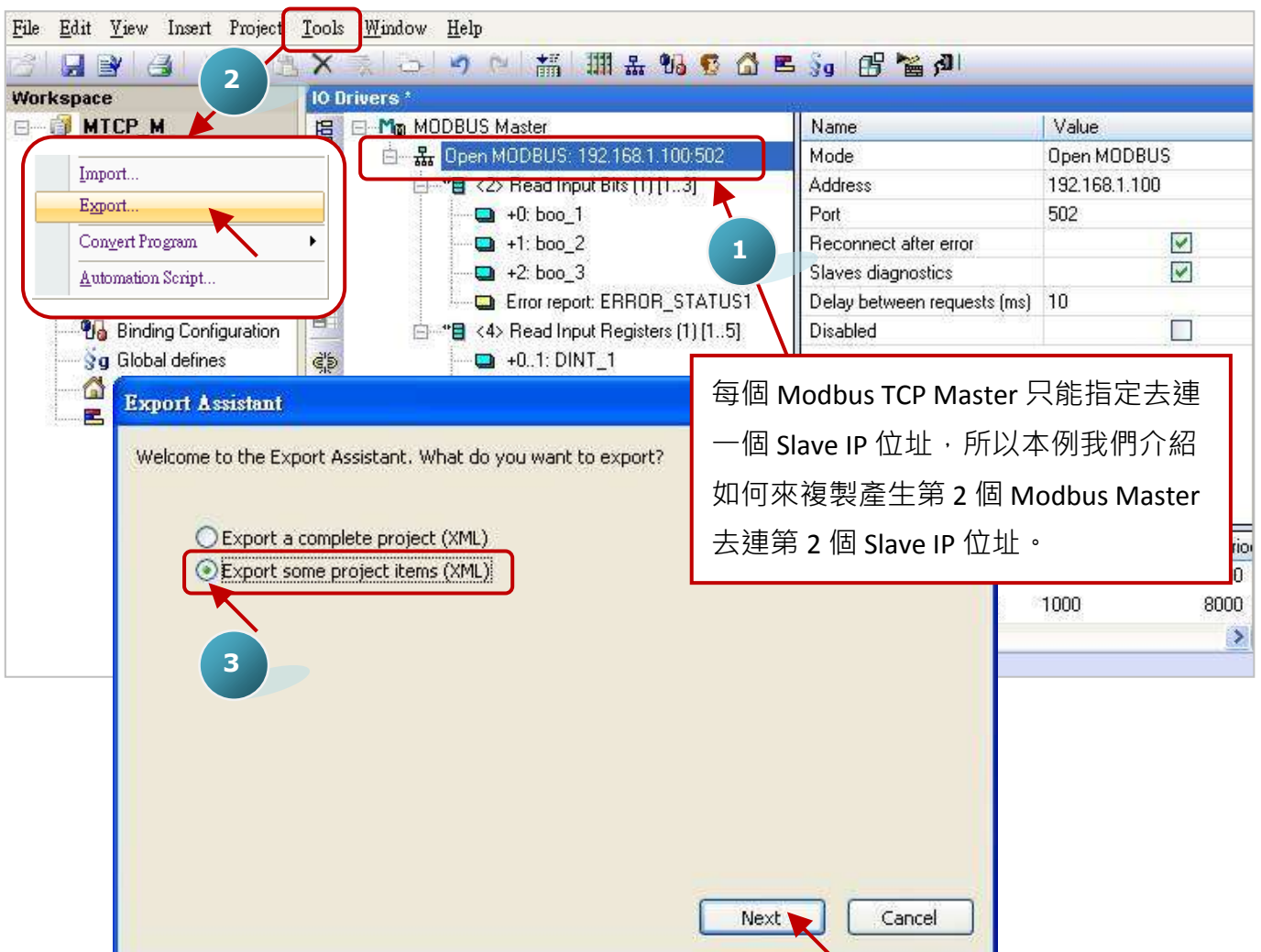
上一節說明了如何啟用 Win-GRAF PAC 為 Modbus TCP Master 設備，也說明了去讀/寫 Modbus TCP Slave 設備的方式。此章節將說明如何建立備援的“Modbus Master Request”設定，當 Modbus TCP Slave 設備的其中一個 IP 無法使用時，另一個 IP 仍會正常的讀/寫資料。

應用示意圖：

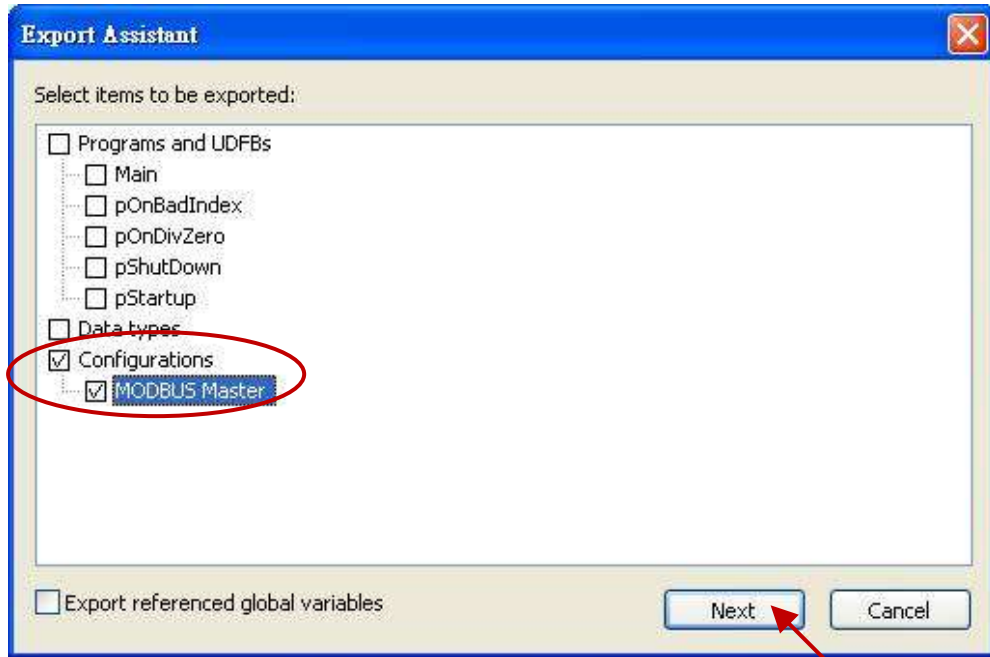


請參考以下操作步驟：

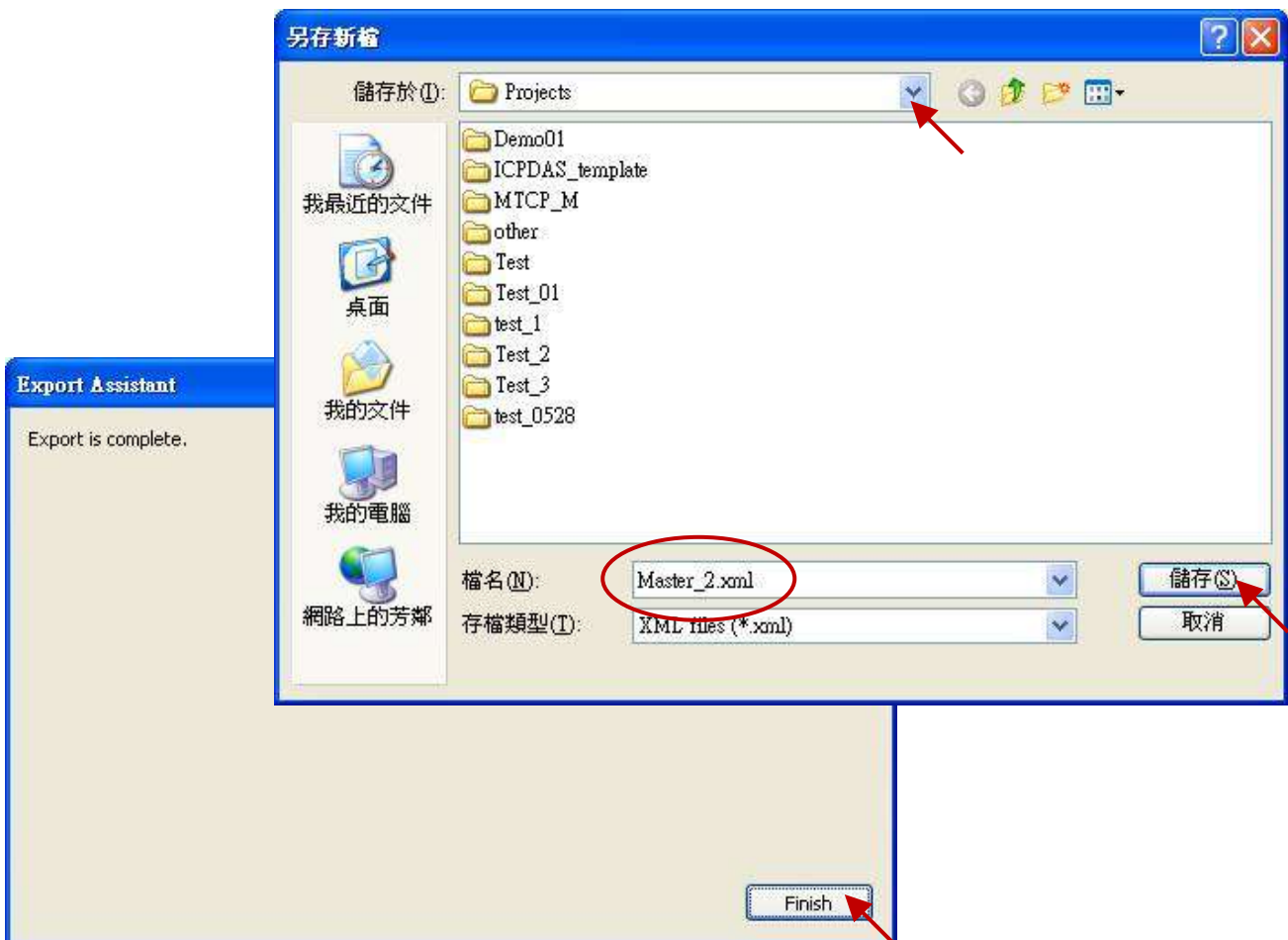
1. 滑鼠點選在“Open MODBUS:”，再點選功能表“Tools”並選擇“Export”項目。
2. 於“Export Assistant”視窗中，點選“Export some project items (XNL)”再點選“Next”進行下一步。



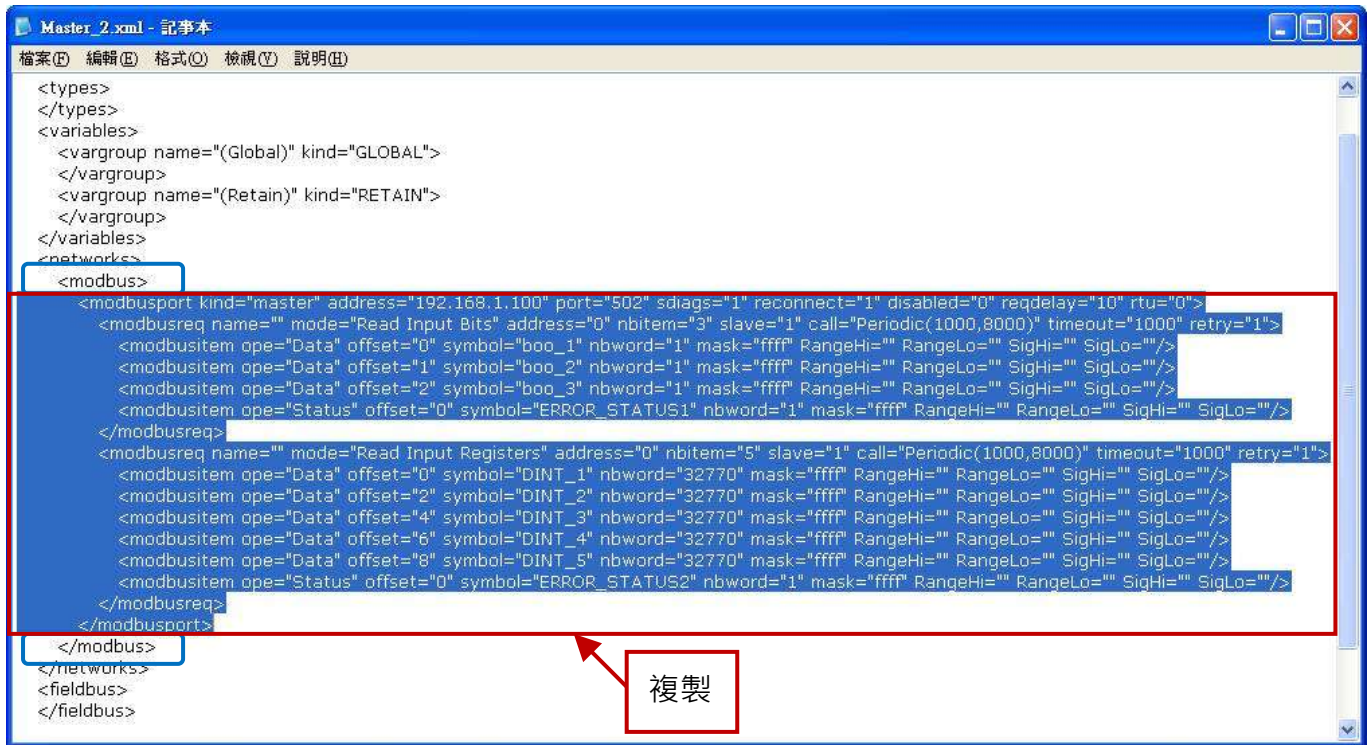
3. 取消其它勾选，仅保留“Configurations”勾选并点击“Next”进行下一步。



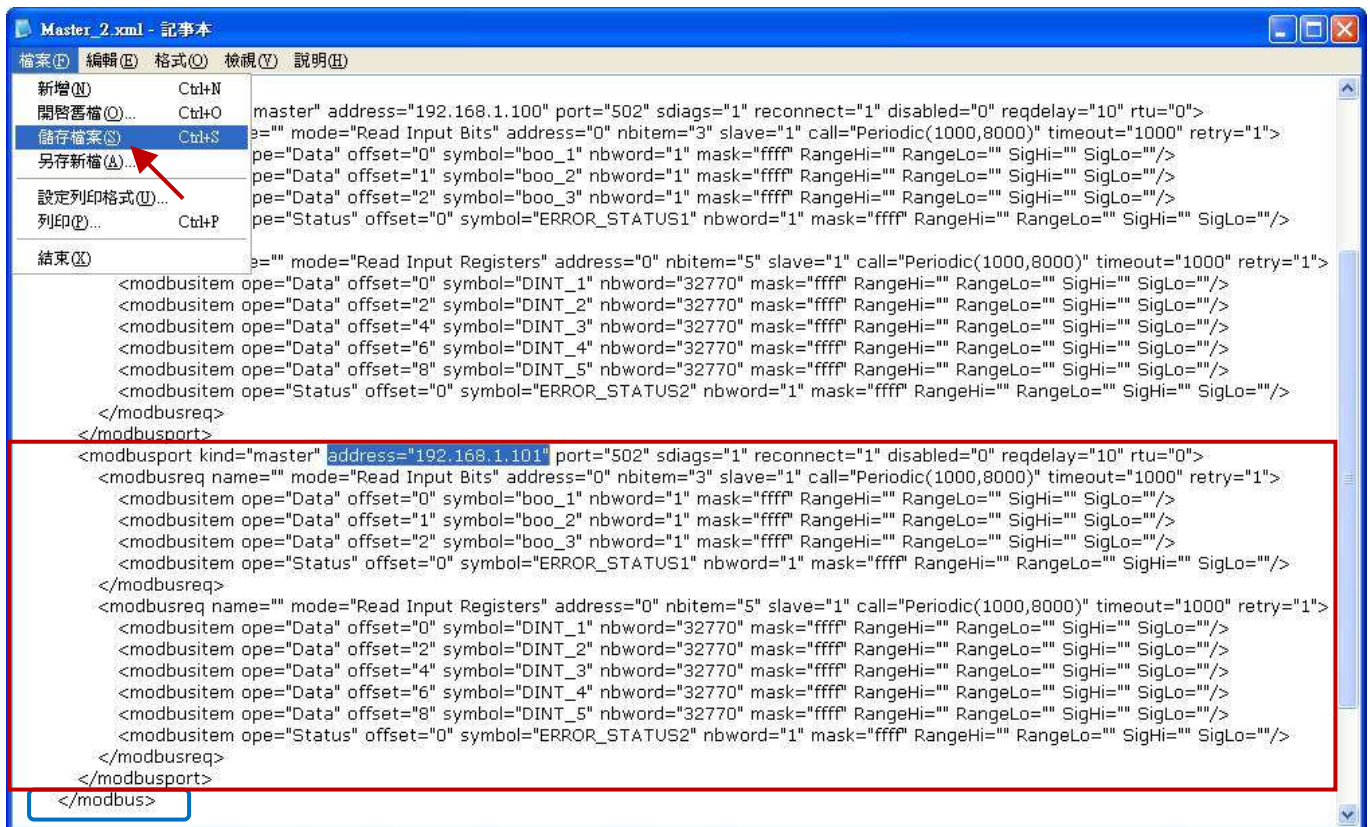
4. 寻找适合的路径 (预设设在 C:\Win-GRAF\Projects) 并为此档案命名 (例如: Master_2.xml)，再點選“儲存”按钮。最後，點選“Finish”完成匯出設定。



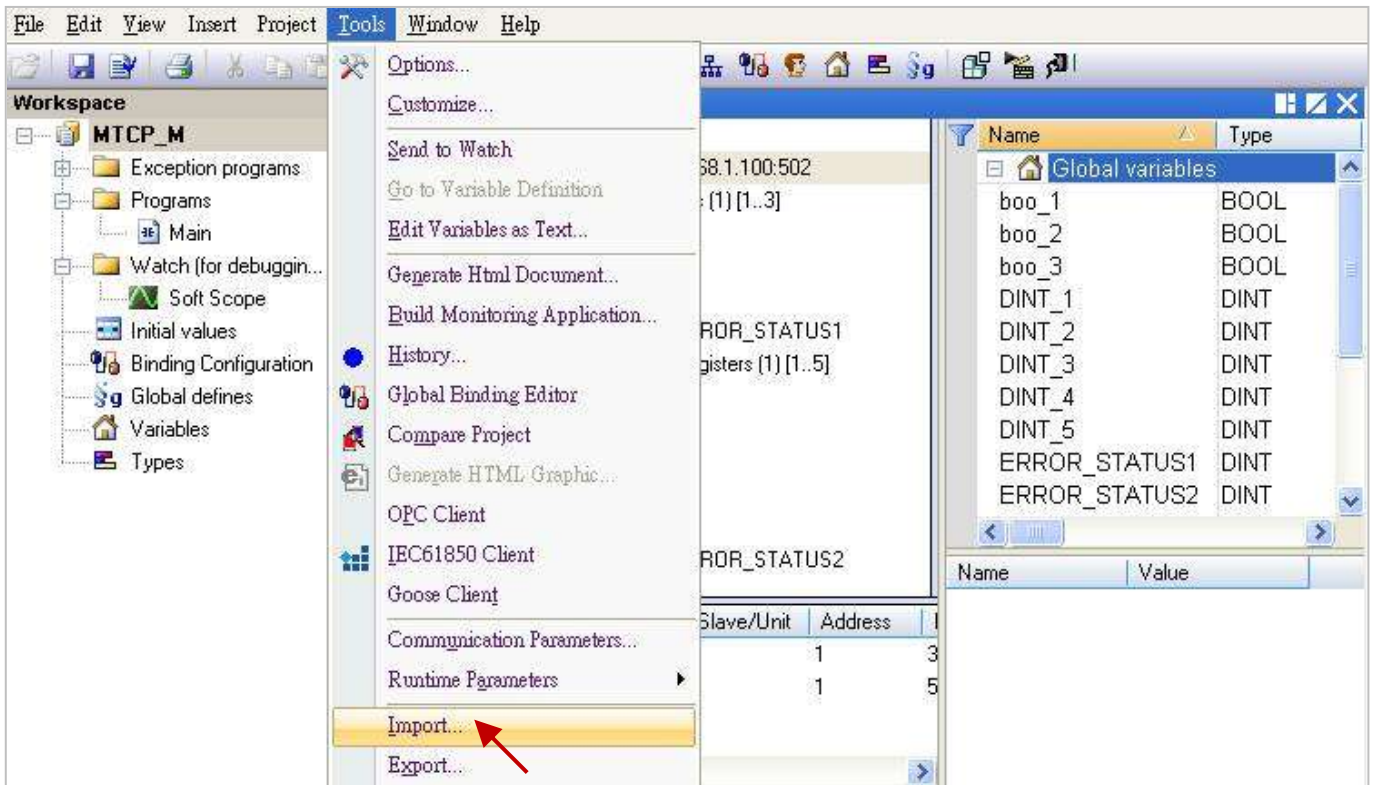
5. 以記事本 (Notepad) 開啟於步驟 4 匯出的 .xml 檔案，並複製 <modbus> 與 </modbus> 中間的內容。



6. 將複製的內容貼在 </modbus> 之上，並修改 address 為 Modbus Slave 設備的第二個 IP 位址 (例如: "192.168.1.101")，再儲存並關閉檔案。

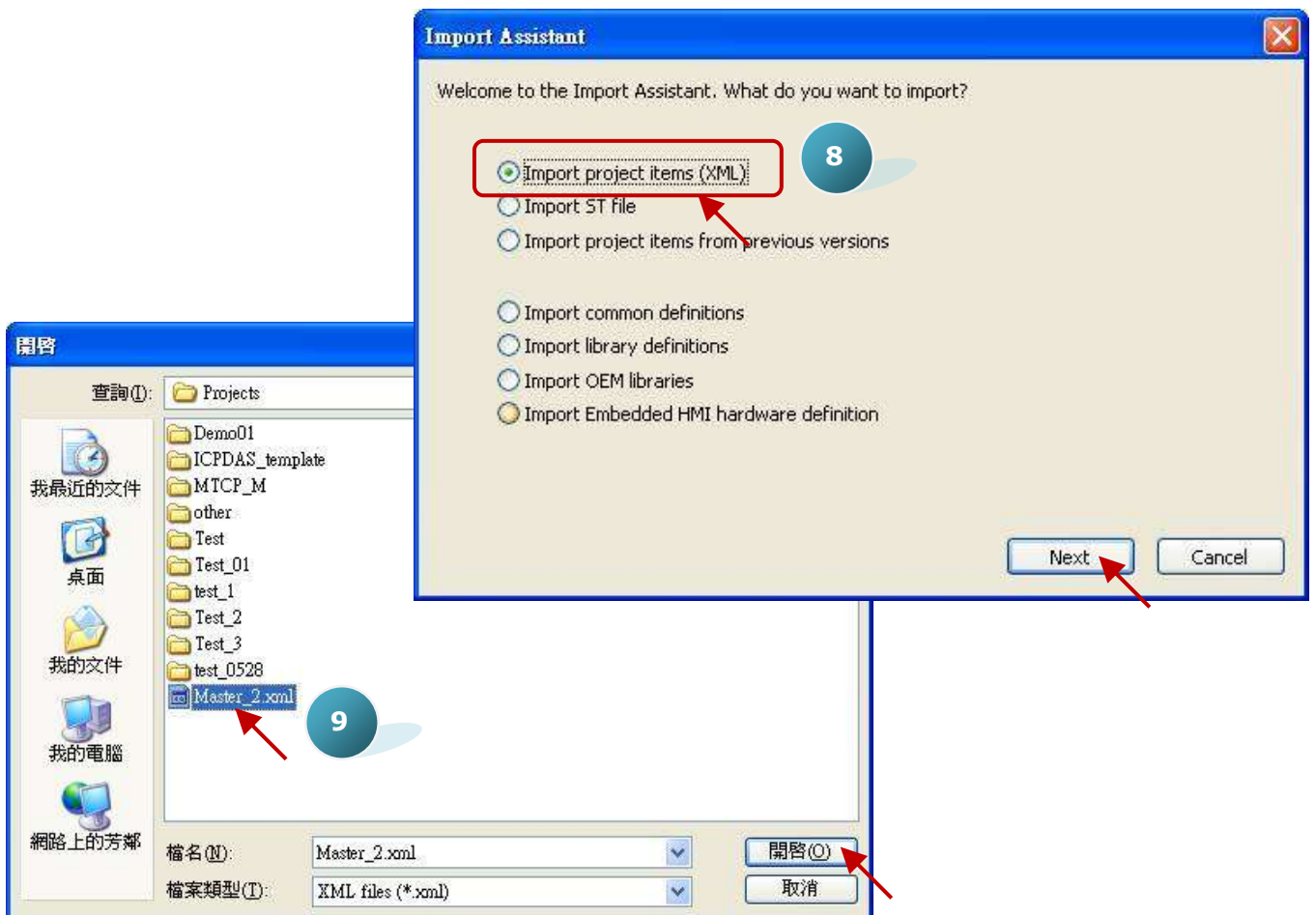


7. 點選 Win-GRAF 的功能表 “Tools” 並選擇 “Import” 項目。



8. 於 “Import Assistant” 視窗中，點選 “Import project items (XNL)” 再點選 “Next” 進行下一步。

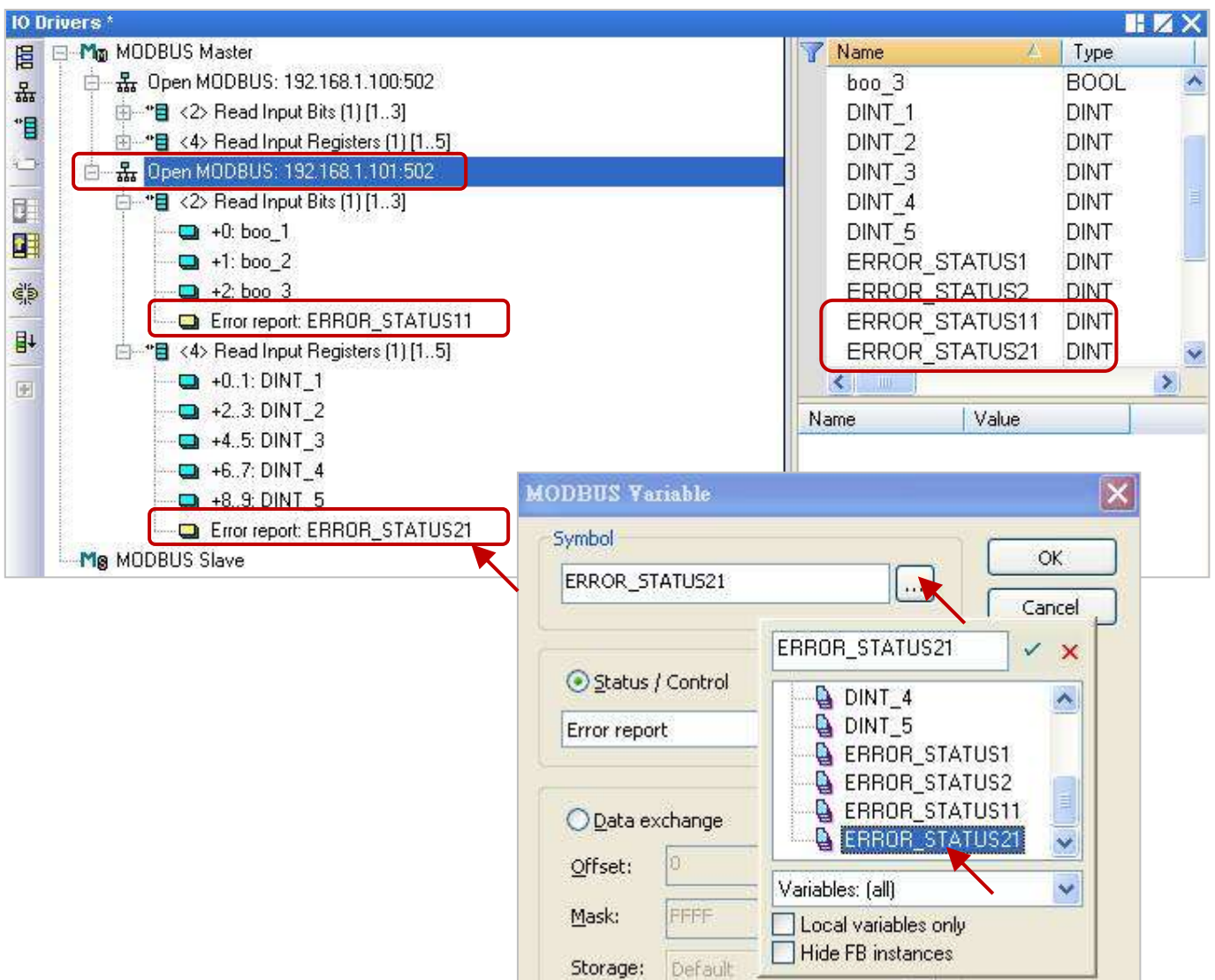
9. 選取欲匯入的檔案 (例如: “Master_2.xml”) 並點選 “開啟” 按鈕。



10. 點選 “Finish” 完成匯入設定。



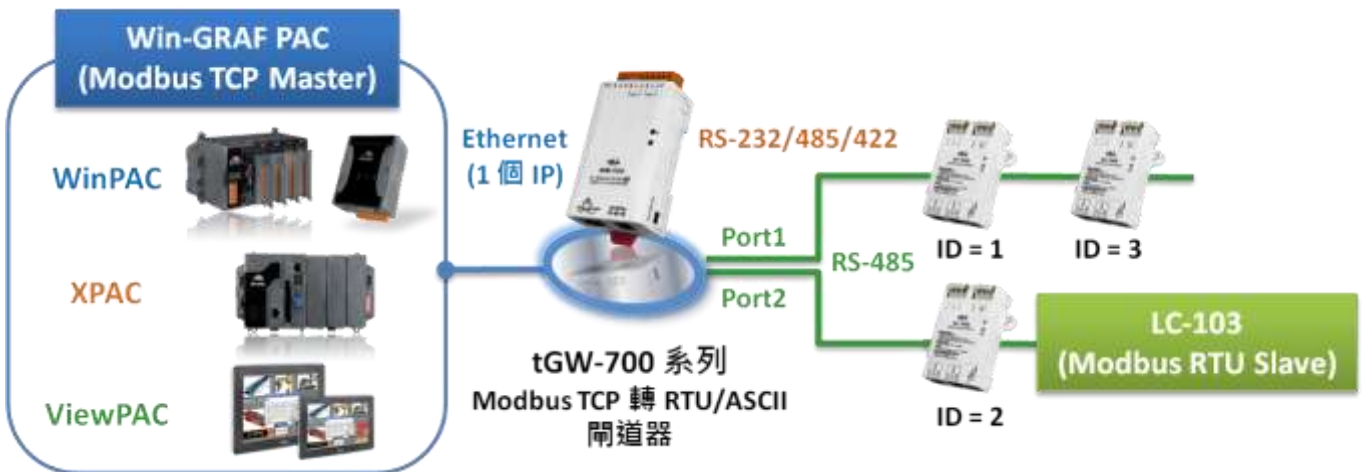
11. 在 “I/O Drivers” 視窗中，已新增了一個 “Open MODBUS” 設定，此設定包含了 2 個 “Modbus Master Request”，用來讀取 DI 與 AI，而其中 “Error report” 是用來表示 IP 的連線狀態，因此請在變數區新增 2 個 “DINT” 變數 (例如: “ERROR_STATUS11”, “ERROR_STATUS21”) 並滑鼠雙擊 “Error report” 來修改指定的變數。



5.4 連接 tGW-700 來擴充 Modbus RTU Master 串口

一般如需在遠距離傳輸的應用中，使用 Modbus RTU (RS-232/485/422) 設備來進行通訊，為了維持較好的訊號品質需選用較低的通訊速率 (Baud rate)，但這種方式卻影響了資料取得的效率。為了解決這項難題，ICP DAS 提供了 tGW-700 系列產品 (Modbus TCP 轉 RTU/ASCII 閘道器)，可將原本的 RS-485 通訊轉換成乙太網路，如此即可縮短 RS-485 訊號線的長度，進而解決通訊效率不佳的問題。

本章節會提供一個範例程式 (demo_tgw725.zip)，用來說明 Win-GRAF PAC 如何透過 tGW-700 閘道器與 LC-103 模組進行通訊 (如下圖)。



5.4.1 使用 tGW-700 系列 (Modbus TCP 轉 Modbus RTU/ASCII 閘道器)

tGW-700 系列模組擁有 Modbus TCP 轉 Modbus RTU/ASCII 的閘道器功能，能讓 Modbus TCP 主機 (例如: WP-8xx8) 透過乙太網路與序列的 Modbus RTU/ASCII 設備進行通訊，消除傳統序列通訊設備的電纜長度限制。請參訪 tGW-700 系列網頁，取得更多資訊:

http://www.icpdas.com/root/product/solutions/industrial_communication/pds/tgw-700_tc.html

tGW-700 系列 使用手冊

<http://ftp.icpdas.com/pub/cd/tinymodules/napdos/tgw-700/document/>

(請參考第 3、4 章來了解 tGW-700 的網路設定，測試方式與網頁功能配置)：

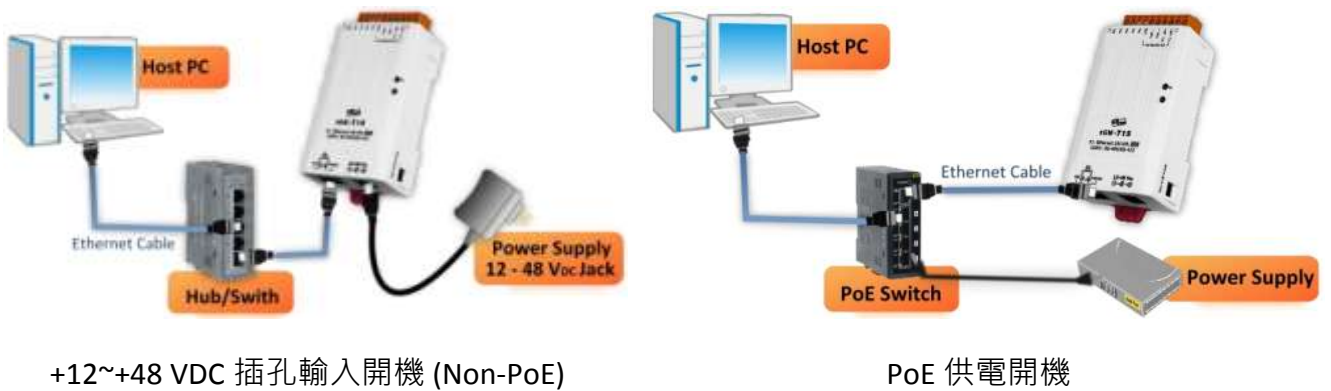
使用前，請先進行 tGW-700 的網路與 COM Port 設定：

● 連接電源和電腦主機。

1. 確認 Init/Run 開關是在 Run 位置上。

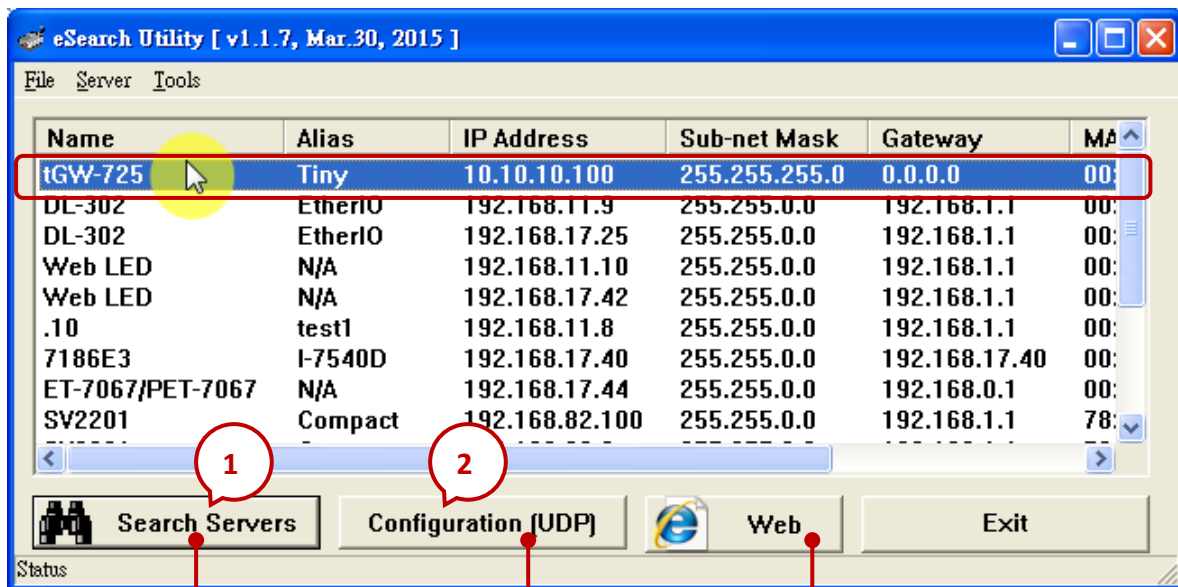


2. 將 tGW-700 系列模組與 PC 接至同一個集線器或同一個子網域，然後供電開機啟動 tGW-700 系列模組。



- 安裝 eSearch Utility 到您的電腦，搜尋並設定 tGW-700 的網路配置。

http://ftp.icpdas.com/pub/cd/tinymodules/napdos/software/modbus_utility/

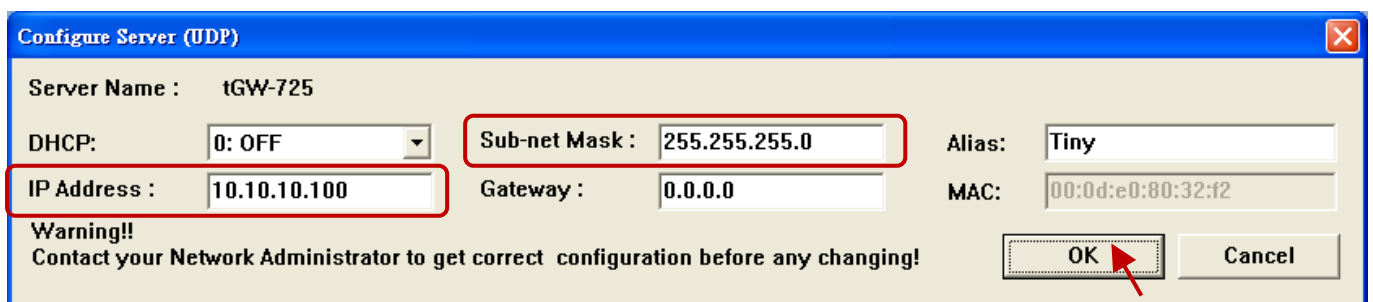


1 搜尋您的 tGW-700。

2 設定 tGW-700 的 IP / Mask / Gateway。(請聯繫您的網路管理員取得正確的配置。)

開啟 tGW-700 網頁伺服器。
(註: 需和 PC 的 IP 位於同一網域，請參考該產品手冊第四章。)

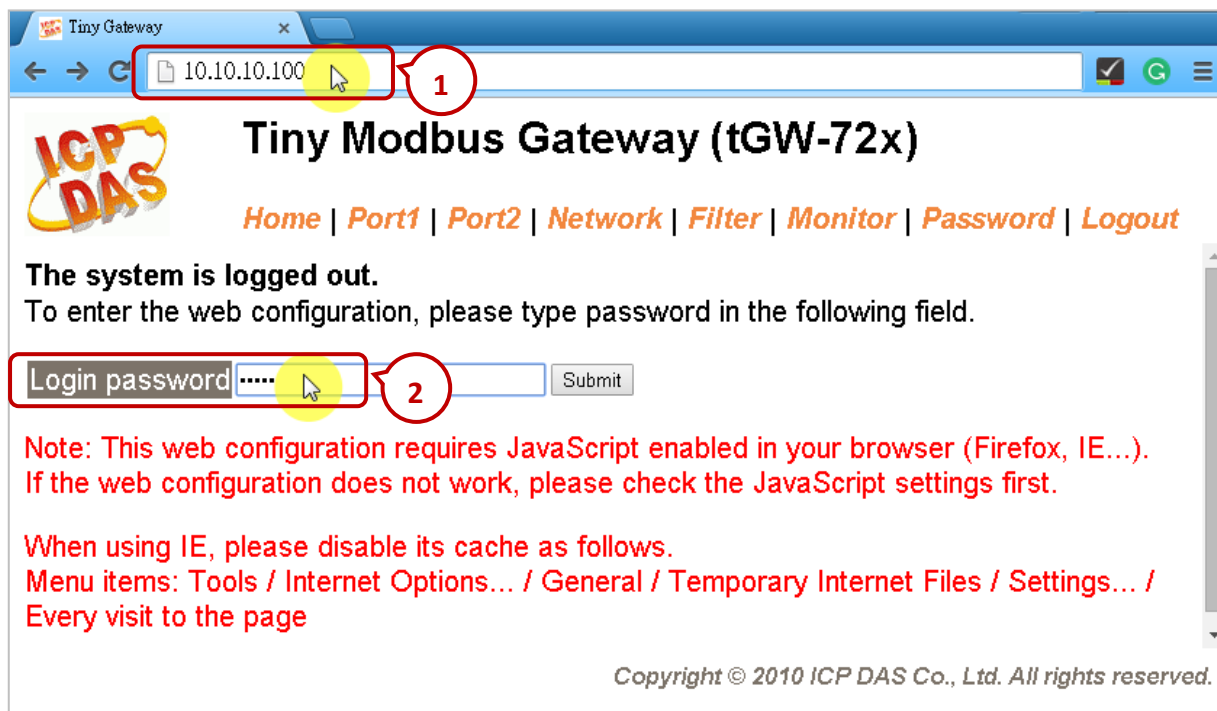
請聯繫您的網路管理員取得正確 IP / Mask / Gateway 設定。修改完成後，單擊“OK”按鈕，tGW-700 將會在 2 秒後套用新的設定。



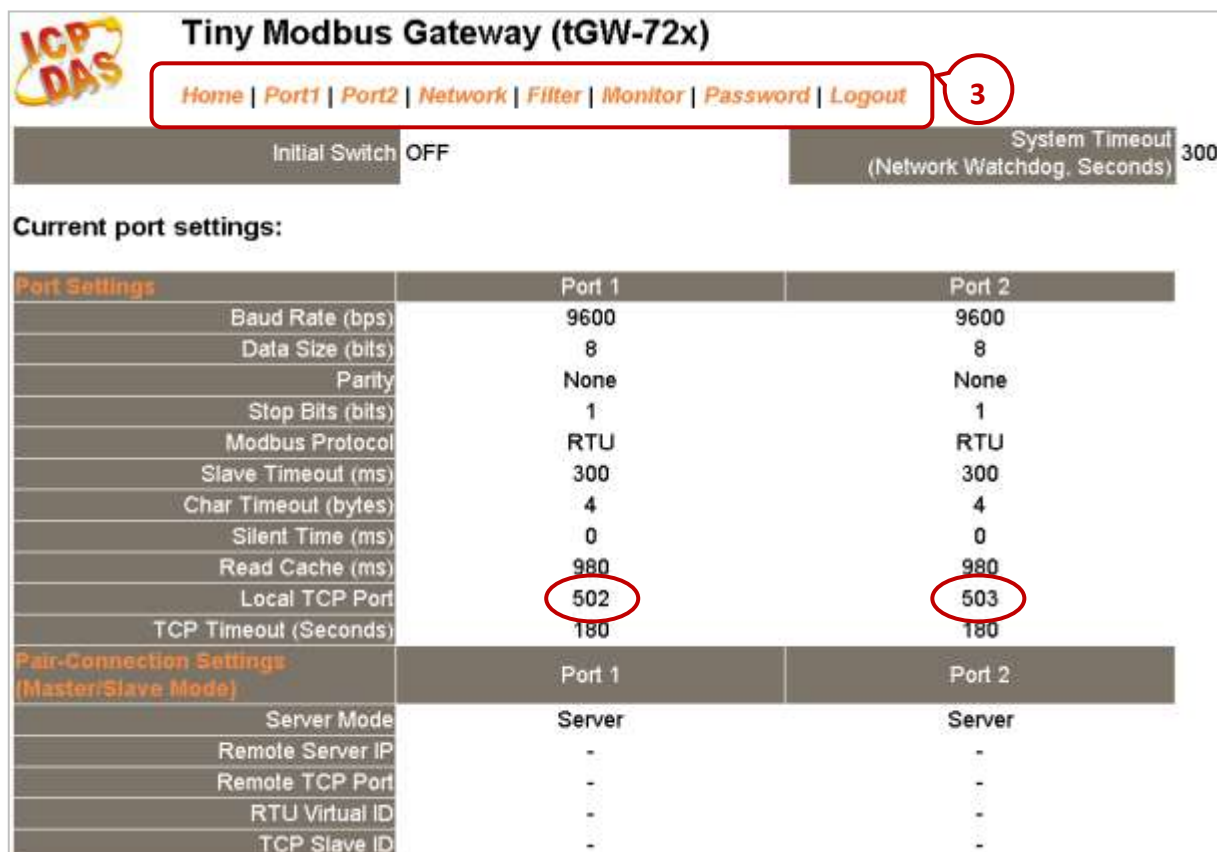
● 網頁配置

您可參考 tGW-700 使用手冊 (第 4 章) 來查看詳細的功能設定，此處將說明 COM Port 設定。

1. 在 Web 瀏覽器輸入 tGW-700 的 IP 位址 (註: 需和您 PC 的 IP 在同一網域)。
2. 輸入登入密碼 (原廠預設密碼: admin)。



3. 登入後，首頁 (Home) 會顯示目前的 Port 設定，您也可點選“Port1”或“Port2”頁籤來進行修改。



5.4.2 連接 tGW-700 系列 與 LC-103 模組 (1 DI, 3 Relay)

本章節提供了一個範例程式 (demo_tgw725.zip) ，用來說明 Win-GRAF PAC 如何透過 tGW-725 (具有 2 個 RS-285 Port 的 Modbus TCP 轉 Modbus RTU/ASCII 閘道器) 與 LC-103 模組進行通訊，您可執行 Win-GRAF Workbench 並點選 "File → Add Existing Project → From Zip..." 來開啟 Win-GRAF PAC 出貨光碟 (\Napdos\Win-GRAF\demo-project) 內的 Win-GRAF 範例程式。

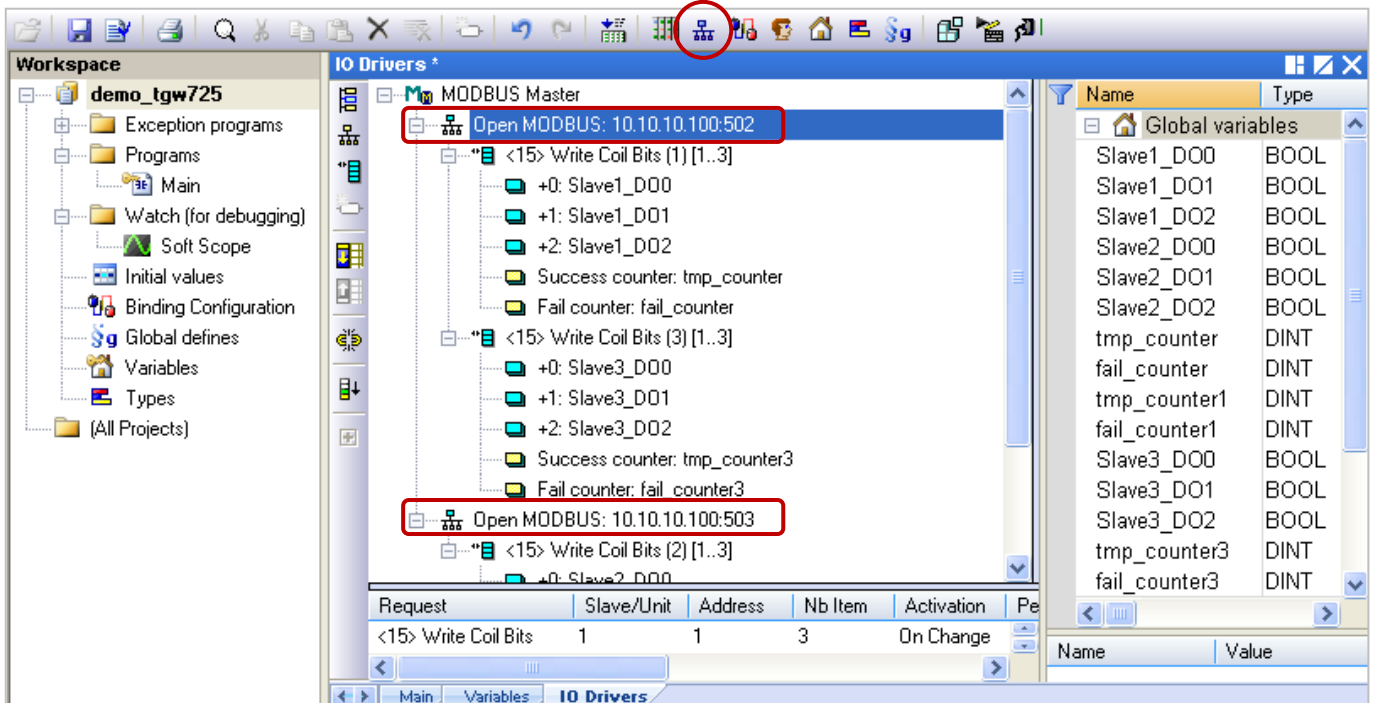
LC-103 模組 是一款支援 Modbus RTU 通訊協定且簡單易用的燈控模組，它提供了 1 個數位輸入通道與 3 個 Relay 輸出通道。使用前，請依據您的需求來設定模組的 ID 編號，例如: ID = 1，請將該模組底部的旋轉開關調整至 "1" 即可。請參訪 LC-103 模組網頁，取得更多資訊:

http://www.icpdas.com/root/product/solutions/remote_io/rs-485/lighting_control/lc-103.html



範例說明: (您可參考 5.2 節 來了解此範例的建立方式)

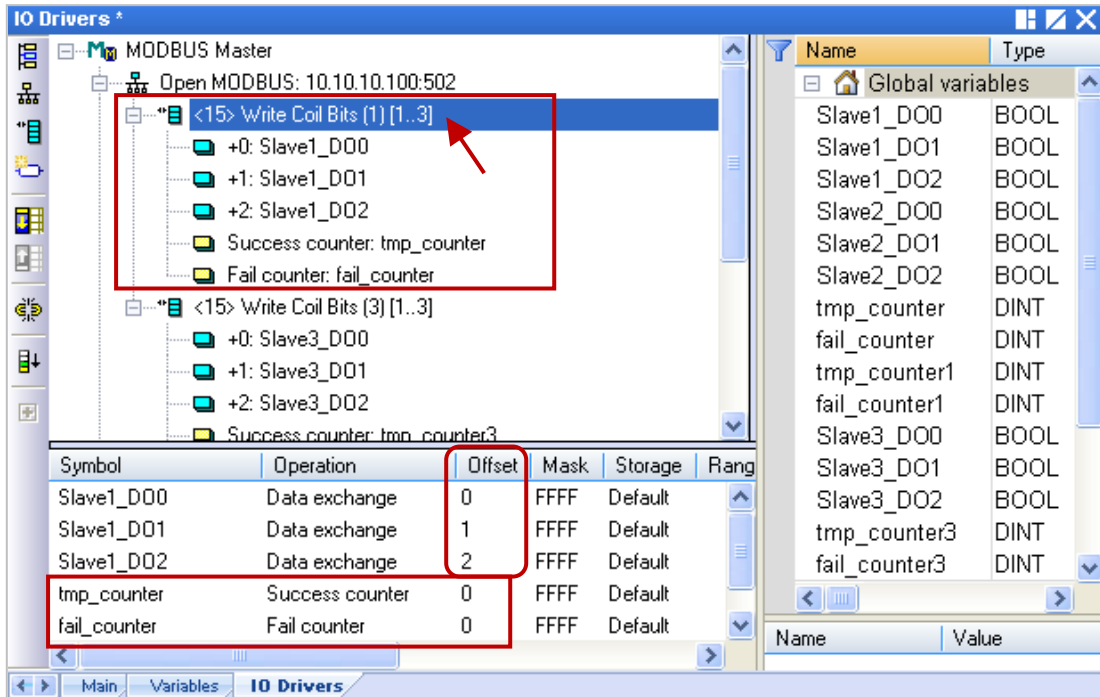
1. 滑鼠點選工具列上的 "Open Fieldbus Configuration" 按鈕來開啟 "I/O Drivers" 視窗。



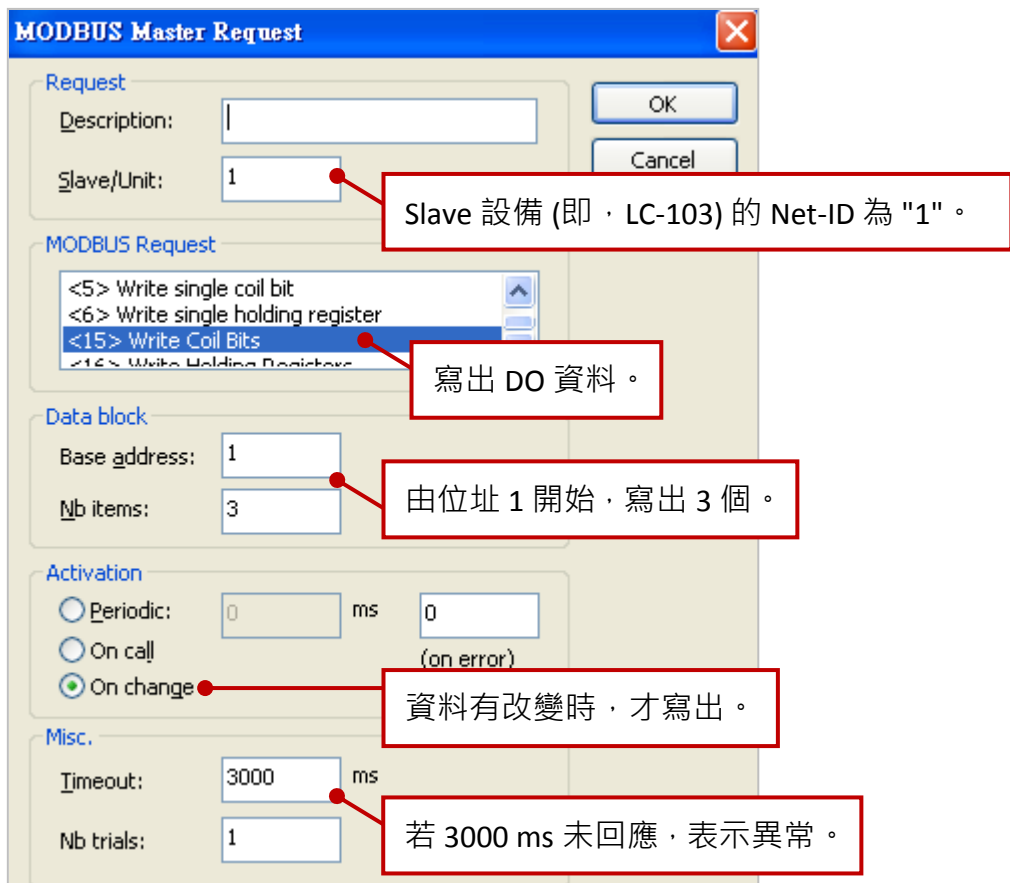
畫面中 "Open MODBUS: 10.10.10.100:502 / Open MODBUS: 10.10.10.100:503" 表示 tGW-725 的 IP 位址為 "10.10.10.100" 且使用了 2 個 COM Port (RS-485) 編號為 "502" 與 "503"，其 COM1 連接了 2 個 LC-103 模組 (Slave ID 為 1 與 3)，COM2 連接了 1 個 LC-103 模組 (Slave ID 為 2)，接下來將一一介紹每個 Modbus Master Request 的設定方式。

2. 滑鼠雙擊 COM1 (Port = 502) 下第 1 個 Data Block 來查看此 Modbus Master Request 。

此例，Win-GRAF PAC (Modbus TCP Master) 透過 tGW-725 的 **COM1 (Port = 502)** 來寫出 3 個 DO 至 **LC-103 (Slave ID = 1)**。如下圖，"Operation" 設定為 "Success counter" 表示寫出成功，該變數值會加 1；設定為 "Fail counter" 表示寫出失敗，該變數值會加 1，另外，此兩變數的 "Offset" 值必需設定為 "0"。

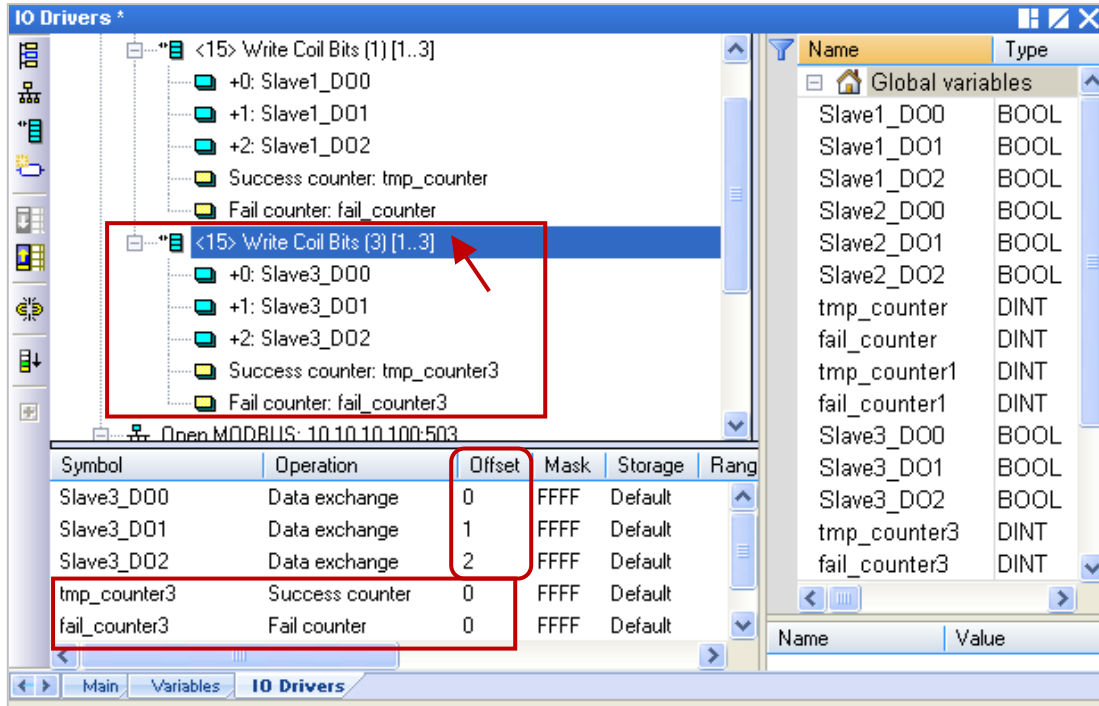


注意：“Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。

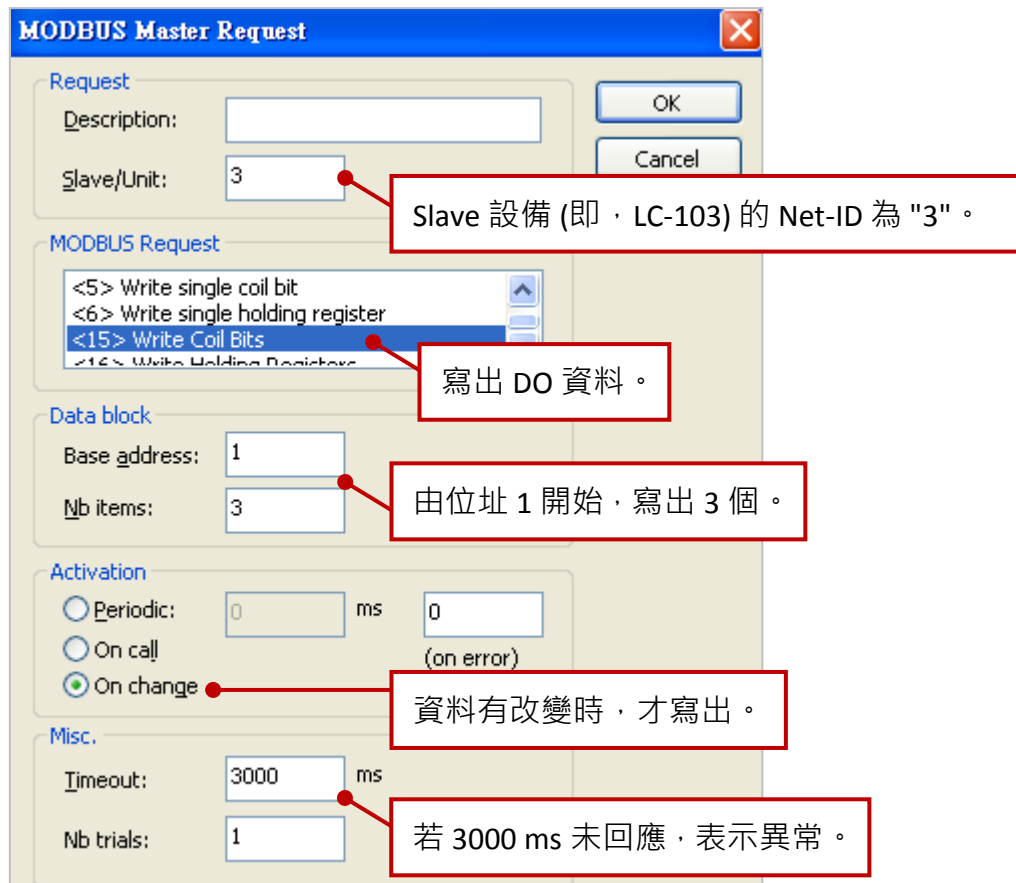


3. 滑鼠雙擊 COM1 (Port = 502) 下第 2 個 Data Block 來查看此 Modbus Master Request 。

此例，Win-GRAF PAC (Modbus TCP Master) 透過 tGW-725 的 **COM1 (Port = 502)** 來寫出 3 個 DO 至 **LC-103 (Slave ID = 3)**。如下圖，"Operation" 設定為 "Success counter" 表示寫出成功，該變數值會加 1；設定為 "Fail counter" 表示寫出失敗，該變數值會加 1，另外，此兩變數的 "Offset" 值必需設定為 "0"。

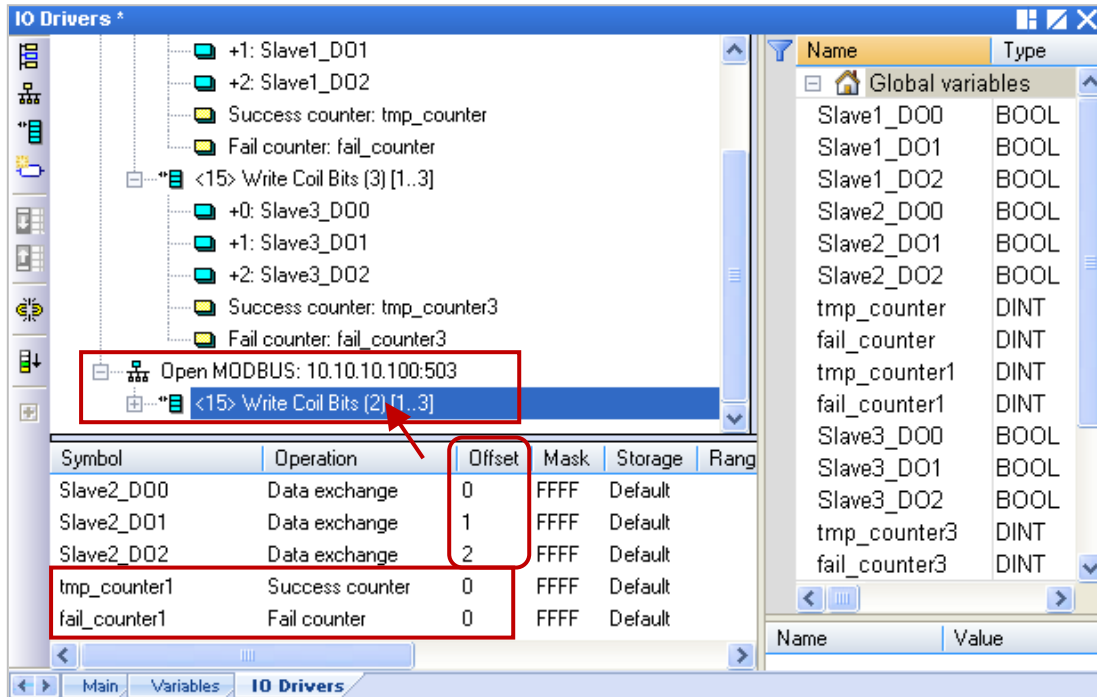


注意：“Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。

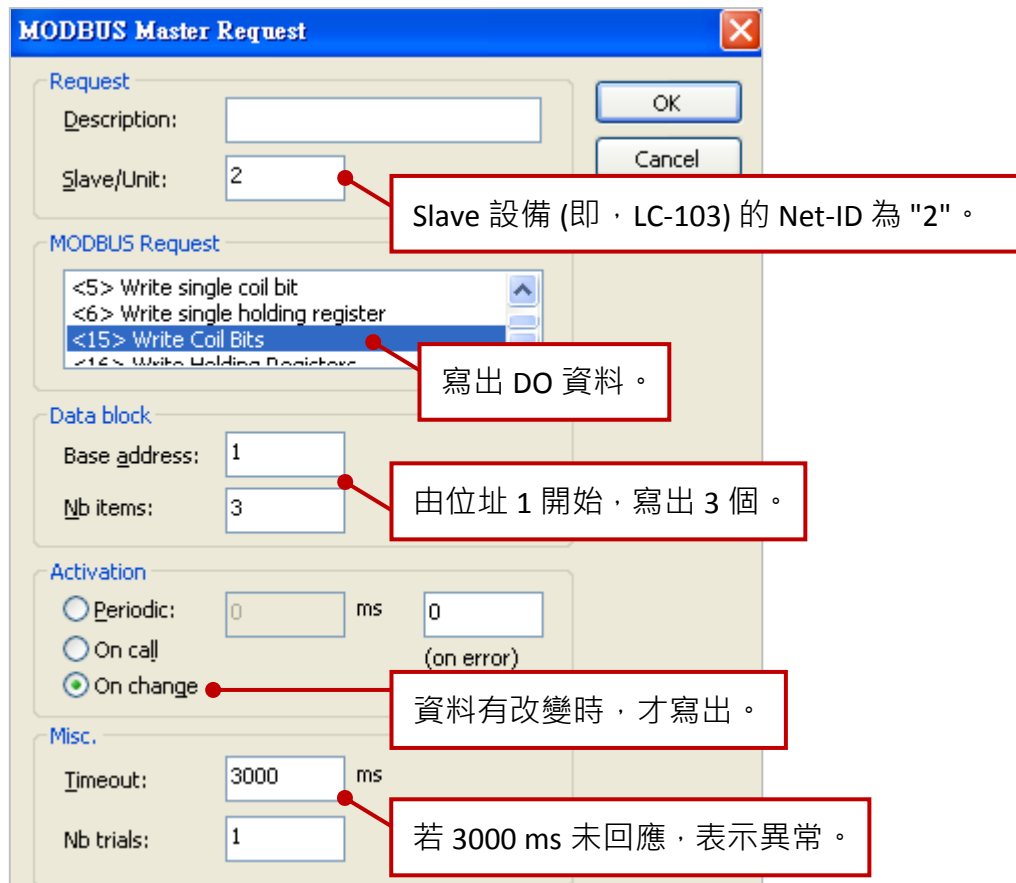


4. 滑鼠雙擊 COM2 (Port = 503) 下的 Data Block 來查看此 Modbus Master Request 。

此例，Win-GRAF PAC (Modbus TCP Master) 透過 tGW-725 的 **COM2 (Port = 503)** 來寫出 3 個 DO 至 **LC-103 (Slave ID = 2)**。如下圖，"Operation" 設定為 "Success counter" 表示寫出成功，該變數值會加 1；設定為 "Fail counter" 表示寫出失敗，該變數值會加 1，另外，此兩變數的 "Offset" 值必需設定為 "0"。



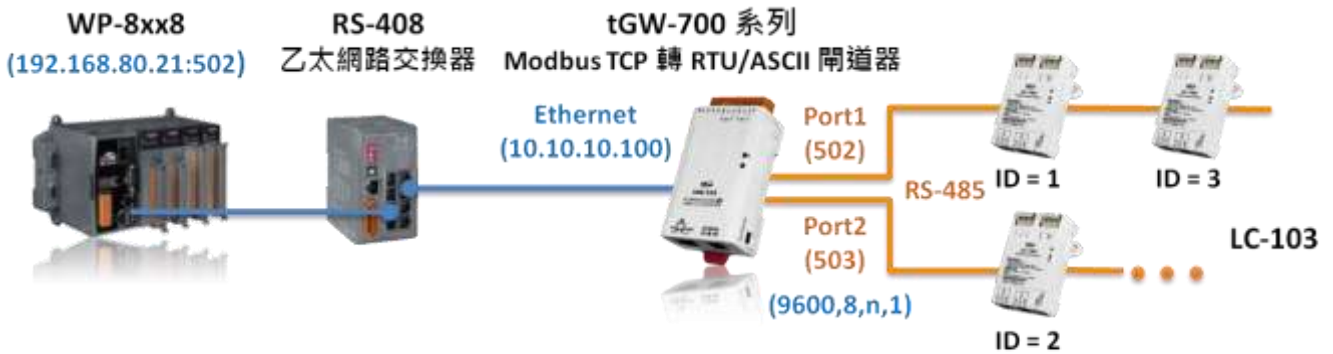
注意：“Offset” 的值是由 “0” 開始，而 “Offset” 值加 1 (Base address) 才是該變數的 Modbus 位址。



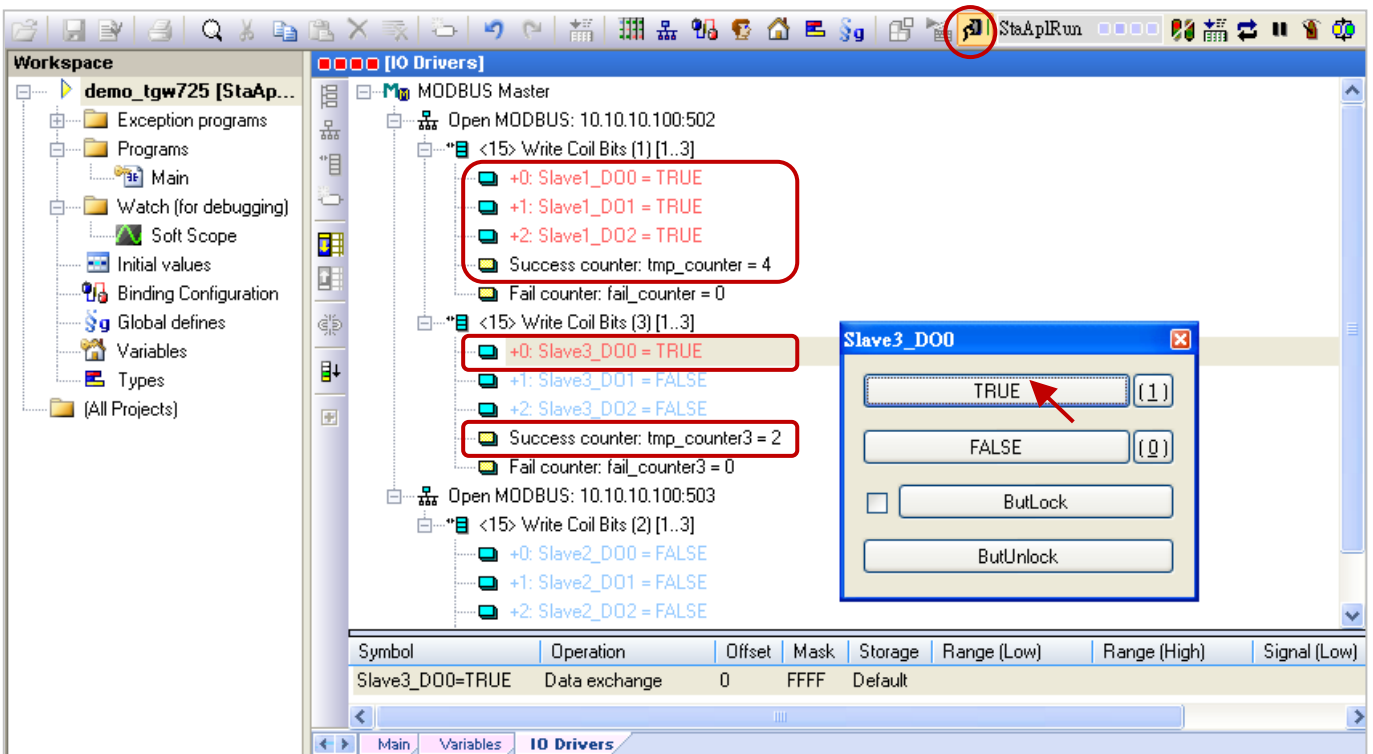
5.4.3 測試範例程式 (demo_tgw725.zip)

測試前，請先將此範例程式下載到您的 Win-GRAF PAC (若不熟悉操作，可參考 [2.3.4 節](#) 與 [2.3.5 節](#))。

硬體連接:



連線後，您可在 "I/O Drivers" 視窗中，滑鼠雙擊任一 DO 變數，並將其狀態設定為 "TRUE"。若寫出成功，則 "tmp_counter" 的值會加 1。



註: 由於 Win-GRAF PAC 一啟動就會發出 Modbus Request 給 Modbus Slave 設備，因此您會見到 "tmp_counter" 的值一開始為 "1"，表示寫出成功。

第 6 章 可保存變數與資料儲存

6.1 可保存變數 (Retain Variable)

此章節將介紹如何使用 "RETAIN_VAR"、"RETAIN_ARY"、"RETAIN_FLAG_GET"、"RETAIN_FLAG_SET" 與 "RETAIN_FLAG_CLR" 函式 (Function) · Win-GRAF 系列 PAC 內建有一個資料保存記憶區，可供使用者保存變數資料，此資料不會因關機而消失，下次開機時仍為上一次的值。

於出貨光碟中 (\Napdos\Win-GRAF\demo-project) · 有提供此章節的範例程式 (demo_retain.zip) · 請參考 [第 12 章](#) 來回存此專案 (執行 File > Add Existing Project > From Zip) 並設定好 PAC 目前的 IP 位址。

註: "Retain_Var()" 或 Retain_Ary() 函式，只能在第一個 PAC Cycle 或 執行線上更新 (On-line Change) 的那個 Cycle 內使用，於其它 Cycle 內呼叫此函式，將會回傳 "FALSE"。當保存變數尚未指定過任何初值時，PAC 程式執行此函式所讀到的值並無意義，使用者至少需為所有的保存變數指定適合的初值一次。

ST 語法: 此範例說明 Retain_Var() 與 Retain_Ary() 函式。

(* 宣告 "on_line_change_cycle" 為 "DINT" 變數，
非 "0" 表示正在執行線上更新 (On-line Change) 的那個 Cycle 內。
宣告 "retain_done" 為 "BOOL" 變數且設定初始值為 "FALSE" 一次。
宣告 "tmp_bool" 為 "BOOL" 變數。

*)

```
on_line_change_cycle := GetSysInfo (_SYSINFO_CHANGE_CYCLE);  
if (retain_done = FALSE) or (on_line_change_cycle <> 0) then  
    retain_done := TRUE; (* 只執行一次即可 *)  
  
    tmp_bool := Retain_Var ( DINT_1 , 1); (* 設定保存一個 "DINT" 變數 *)  
    tmp_bool := Retain_Var ( DINT_2 , 2);  
    tmp_bool := Retain_Var ( REAL_1 , 3); (* 設定保存一個 "REAL" 變數 *)  
    tmp_bool := Retain_Var ( BOOL_1 , 4); (* 設定保存一個 "BOOL" 變數 *)  
    tmp_bool := Retain_Var ( BOOL_2 , 5);  
  
    (* 設定保存 "INT" 陣列變數內的 10 個元素 *)  
    tmp_bool := Retain_Ary ( INT_ARY , 6 , 10);  
  
    (* 設定保存 "REAL" 陣列變數內的 20 個元素 *)  
    tmp_bool := Retain_Ary ( REAL_ARY , 16 , 20);  
  
    tmp_bool := Retain_Var ( DINT_3 , 36);
```

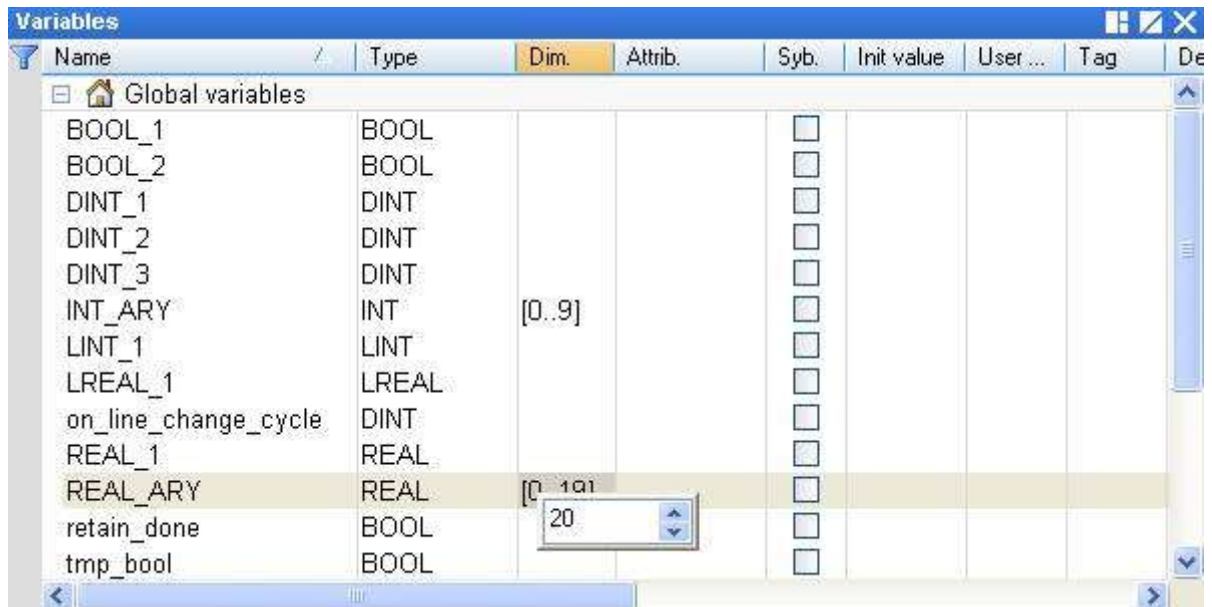
(* 64-bit 變數只能使用位址編號 10001 ~ 12000 *)

```
tmp_bool := Retain_Var ( LINT_1 , 10001); (* 設定保存一個 "LINT" 變數 (64-bit) *)
```

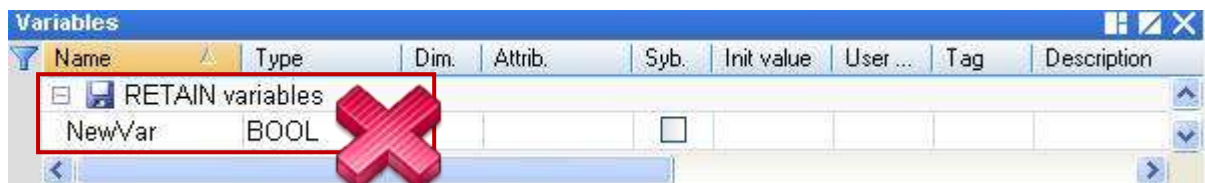
```
tmp_bool := Retain_Var ( LREAL_1 , 10002); (* 設定保存一個 "LREAL" 變數 (64-bit) *)
```

```
end_if;
```

於 "Variables" 視窗可查詢/設定變數，若您不熟悉變數的宣告方式，可參考 [2.2.2 節](#) 與 [2.3.1 節](#)。



注意: ICP DAS Win-GRAF PAC 並不支援 "Variable" 視窗中的 "Retain Variable" 功能，因此請參考下列章節的 5 個函式來使用可保存變數功能。



6.1.1 RETAIN_VAR (設定保存一個變數)



使用小技巧:

按“F1”鍵，可查看詳細的設定說明。

Name:

想要保存資料的變數名稱 (勿使用陣列變數 或 字串) · 資料型態可為 BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT, LREAL。

Addr: (資料型態: “DINT”)

用來保存變數的位址編號，可設定為 1 ~ 12000。

Q: 資料型態為 “BOOL”，“TRUE”: 表示 OK ; “FALSE”: 表示錯誤。

註:

1. 一個位址只能儲存一個變數，請勿將相同的位址指定給兩個變數 (或多個)，否則保存值將會出現錯誤。
2. 64-bit 的資料型態 (例如: LINT, LREAL) 只能使用位址編號 10001 ~ 12000。
3. 其它資料型態 (例如: BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME) 可使用位址編號 1 ~ 12000。

6.1.2 RETAIN_ARY (設定保存一個陣列變數)



使用小技巧:

按“F1”鍵，可查看詳細的設定說明。

Name[]:

想要保存資料的陣列變數名稱 (勿使用字串 或 非陣列變數) · 資料型態可為 BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT, LREAL。

Addr: (資料型態: “DINT”)

用來保存陣列變數的起始位址編號 · 可設定為 1 ~ 12000。

Num: (資料型態: “DINT”)

陣列變數內想要保存的資料數量。

例如 · 若陣列變數有 100 個元素 · 設定 "Num" 為 "1 ~ 100" 是正確的 · 但若設為大於 100 就不對。

若陣列變數有 5 個元素 · 則設定 "Num" 為 "1 ~ 5" 是正確的 · 但若設為大於 5 就不對。

Q: 資料型態為 “BOOL” · “TRUE”: 表示 OK ; “FALSE”: 表示錯誤。

註:

1. 一個位址只能儲存一個變數 (或陣列中的一個元素) · **請勿**將相同的位址指定給兩個變數 (或多個) · 否則保存值將會出現錯誤。
2. 64-bit 的資料型態 (例如: LINT, LREAL) 只能使用位址編號 10001 ~ 12000。
3. 其它資料型態 (例如: BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME) 可使用位址編號 1 ~ 12000。

6.1.3 RETAIN_FLAG_SET/GET/CLR (設定/取得/刪除 Flag 的狀態)

如何使用:

"RETAIN_FLAG" 是一個儲存在保存記憶體中的旗標 (TRUE / FALSE) · 可讓使用者判斷保存資料是否為有效值。若先前保存變數未設置過任何初值 · 則 PAC 一開機從保存記憶體所讀到的值並不是正確的 (一般會是個亂數值) · 因此為了讓程式可正常運作 · 使用者至少需為所有的保存變數設定過一次適當的值 · 然後可呼叫 "Retain_Flag_Set()" 將保存旗標設定為 "TRUE" · 表示 "所有的保存變數已設置了適當的值"; 呼叫 "Retain_Flag_Get()" 可取得保存旗標的狀態; 呼叫 "Retain_Flag_Clr()" 則可刪除保存旗標的狀態。

ST 語法:

(* 宣告 "on_line_change_cycle" 為 "DINT" 變數 · 非 "0" 表示正在執行線上更新 (On-line Change) 的那個 Cycle 內。

宣告 "retain_done" 為 "BOOL" 變數且初始值為 "FALSE"。

宣告 "tmp_bool" · "retain_flag" 與 "to_set_flag" 為 "BOOL" 變數。

*)

```
on_line_change_cycle := GetSysInfo (_SYSINFO_CHANGE_CYCLE);
```

```
if (retain_done = FALSE) or (on_line_change_cycle <> 0) then
```

```
  retain_done := TRUE; (* 只執行一次即可 *)
```

```
  tmp_bool := Retain_Var( DINT_1 , 1); (* 設定保存一個 "DINT" 變數 *)
```

```
  tmp_bool := Retain_Var( DINT_2 , 2);
```

```
  tmp_bool := Retain_Var( REAL_1 , 3); (* 設定保存一個 "REAL" 變數 *)
```

```
  tmp_bool := Retain_Var( BOOL_1 , 4); (* 設定保存一個 "BOOL" 變數 *)
```

```
(* ... 在執行所有的 Retain 函式後 ... *)
```

```
retain_flag := Retain_Flag_Get();
```

```
if (retain_flag = FALSE) then
```

```
  (* 若先前保存變數並未設定過適當的值 · 您可在此執行一些適當的操作。*)
```

```
  (* ... *)
```

```
end_if;
```

```
end_if;
```

(* 當您為所有的保存變數都指定了適當的值後 · 請記得將以下的 "to_set_flag" 設定為 "TRUE" · 以便調用 "Retain_Flag_Set()" 一次 · 如此 · 下次使用到 "Retain_Flag_Get()" 時就會回傳 "TRUE"。*)

```
if (to_set_flag = TRUE) then
```

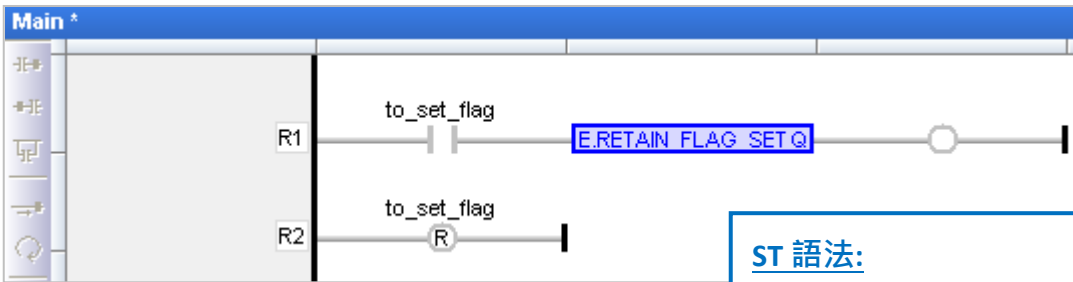
```
  to_set_flag := FALSE;
```

```
  tmp_bool := Retain_Flag_Set();
```

```
end_if;
```


LD 語法: (按 “F1” 鍵，可查看詳細的設定說明。)

RETAIN_FLAG_SET : 設定保存旗標。

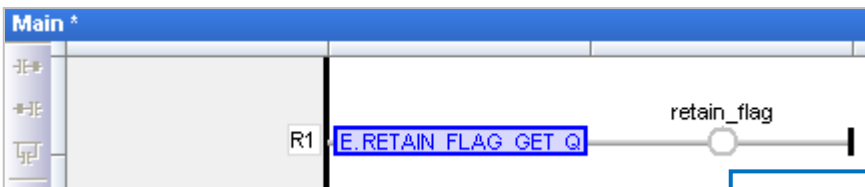


Q: 資料型態為 “BOOL”，只回傳 “TRUE”。

ST 語法:

```
if to_set_flag then  
    to_set_flag := FALSE ;  
    TMP_BOOL := Retain_Flag_Set() ;  
end_if ;
```

RETAIN_FLAG_GET : 取得保存旗標的狀態。

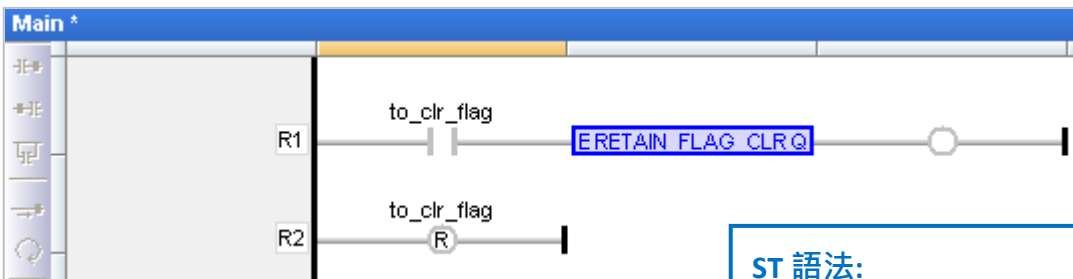


Q: 資料型態為 “BOOL”，
“TRUE”: 表示已設立旗標 (Flag) ;
“FALSE”: 表示未設立旗標 (Flag)。

ST 語法:

```
retain_flag := Retain_Flag_Get() ;
```

RETAIN_FLAG_CLR : 刪除保存旗標的狀態。



Q: 資料型態為 “BOOL”，只回傳 “TRUE”。

ST 語法:

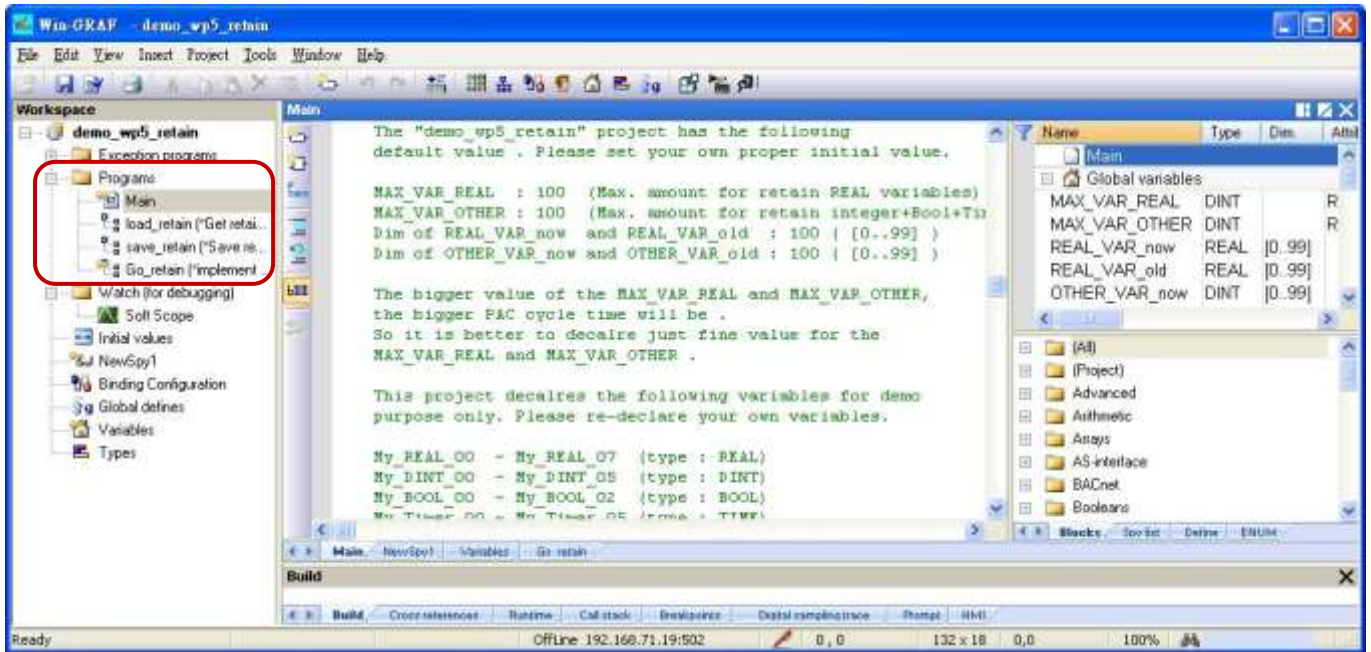
```
if to_clr_flag then  
    to_clr_flag := FALSE ;  
    TMP_BOOL := Retain_Flag_Clr() ;  
end_if ;
```

6.2 可保存變數 (使用檔案)

若您想使用 File 來保存變數，您可使用此章節所提供的範例程式 (demo_wp5_retain.zip) 將需保存的變數資料儲存在 PAC 內 \System_disk\Win-GRAF\ 目錄下的檔案中。

於出貨光碟中 (\Napdos\Win-GRAF\demo-project) 有提供此範例程式 (demo_wp5_retain.zip)，請參考 [第 12 章](#) 來回存此專案 (執行 File > Add Existing Project > From Zip) 並設定好 PAC 目前的 IP 位址。

此專案包含了 1 個 ST 主程式 (Main) 與 3 個 ST 副程式 (load_retain, save_retain 與 Go_retain)。



使用限制:

此專案並不適合處理數值會頻繁變動的保存變數 (例如: 每秒或每分鐘需變更數值)。這是因為變數資料是儲存在 \System_disk\ 目錄下，若保存數值常常變動，則檔案更新會消耗很多 CPU 時間進而造成 PAC 效能低弱。

以下為 "demo_wp5_retain" 專案中設定的初值，請依實際需求設定適當的值。

MAX_VAR_REAL: **100** (最多可使用 REAL 保留變數的數量)

MAX_VAR_OTHER: **100** (最多可使用 Integer, BOOL, TIMER 保留變數的全部數量)

"REAL_VAR_now" 與 "REAL_VAR_old" 的 Dim. 值: **100** ([0..99]，需和 "MAX_VAR_REAL" 值相同)

"OTHER_VAR_now" 與 "OTHER_VAR_old" 的 Dim. 值: **100** ([0..99]，需和 "MAX_VAR_OTHER" 值相同)

註: "MAX_VAR_REAL" 與 "MAX_VAR_OTHER" 的設定值越大，表示 PAC 的 Cycle Time 也越大。因此，請將此兩個數值設定為接近實際的使用數量即可。

以下為此專案中宣告的變數，請依實際需求來宣告變數。

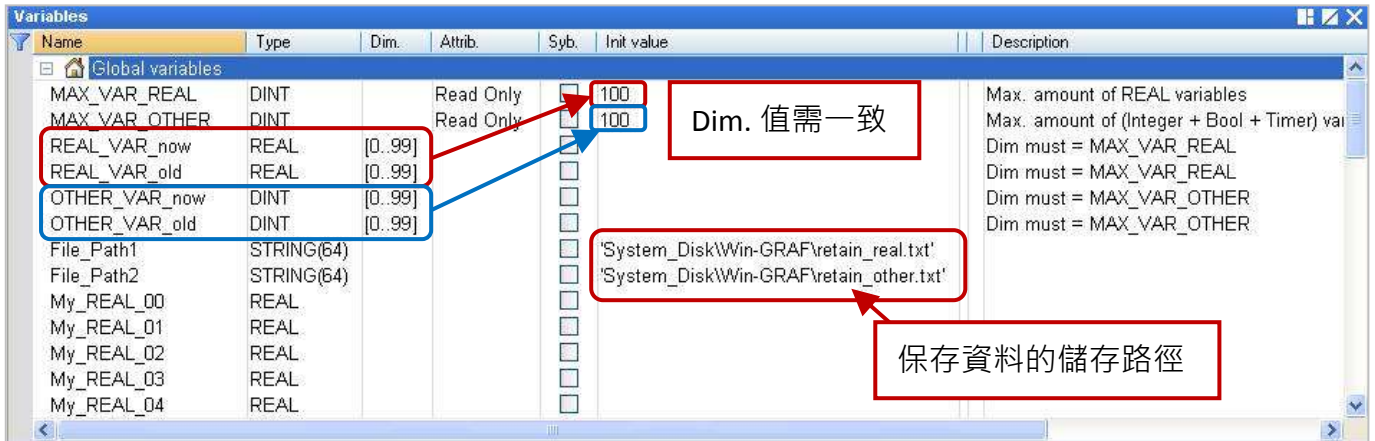
My_REAL_00 ~ My_REAL_07 (資料型態: REAL)

My_DINT_00 ~ My_DINT_05 (資料型態: DINT)

My_BOOL_00 ~ My_BOOL_02 (資料型態: BOOL)

My_Timer_00 ~ My_Timer_05 (資料型態: TIME)

您可在 "Variables" 視窗查看更多的變數。



"Go_retain" 副程式是用來執行檔案中的保存動作，請記得修改以下的 4 個區塊 (註解為 "Add your REAL variables for retain here" 與 "Add your integer, BOOL, Timer variables for retain here")，請依實際宣告的變數來修改：

(* Add your REAL variables for retain here *)

(* ----- *)

```
My_REAL_00 := REAL_VAR_now[0];
My_REAL_01 := REAL_VAR_now[1];
My_REAL_02 := REAL_VAR_now[2];
My_REAL_03 := REAL_VAR_now[3];
My_REAL_04 := REAL_VAR_now[4];
My_REAL_05 := REAL_VAR_now[5];
My_REAL_06 := REAL_VAR_now[6];
My_REAL_07 := REAL_VAR_now[7];
```

(* ----- *)

(* Add your integer, BOOL, Timer variables for retain here *)

(* ----- *)

```
My_DINT_00 := OTHER_VAR_now[0];
My_DINT_01 := OTHER_VAR_now[1];
My_DINT_02 := OTHER_VAR_now[2];
My_DINT_03 := OTHER_VAR_now[3];
My_DINT_04 := OTHER_VAR_now[4];
My_DINT_05 := OTHER_VAR_now[5];
```

```
My_BOOL_00 := Any_to_BOOL( OTHER_VAR_now[6] );
My_BOOL_01 := Any_to_BOOL( OTHER_VAR_now[7] );
My_BOOL_02 := Any_to_BOOL( OTHER_VAR_now[8] );
```

```
My_Timer_00 := Any_to_TIME( OTHER_VAR_now[9] );
My_Timer_01 := Any_to_TIME( OTHER_VAR_now[10] );
My_Timer_02 := Any_to_TIME( OTHER_VAR_now[11] );
My_Timer_03 := Any_to_TIME( OTHER_VAR_now[12] );
My_Timer_04 := Any_to_TIME( OTHER_VAR_now[13] );
My_Timer_05 := Any_to_TIME( OTHER_VAR_now[14] );
(* ..... *)
```

(* Add your REAL variables for retain here *)

```
(* ..... *)
REAL_VAR_now[0] := My_REAL_00 ;
REAL_VAR_now[1] := My_REAL_01 ;
REAL_VAR_now[2] := My_REAL_02 ;
REAL_VAR_now[3] := My_REAL_03 ;
REAL_VAR_now[4] := My_REAL_04 ;
REAL_VAR_now[5] := My_REAL_05 ;
REAL_VAR_now[6] := My_REAL_06 ;
REAL_VAR_now[7] := My_REAL_07 ;
(* ..... *)
```

(* Add your integer, BOOL, Timer variables for retain here *)

```
(* ..... *)
OTHER_VAR_now[0] := My_DINT_00 ;
OTHER_VAR_now[1] := My_DINT_01 ;
OTHER_VAR_now[2] := My_DINT_02 ;
OTHER_VAR_now[3] := My_DINT_03 ;
OTHER_VAR_now[4] := My_DINT_04 ;
OTHER_VAR_now[5] := My_DINT_05 ;

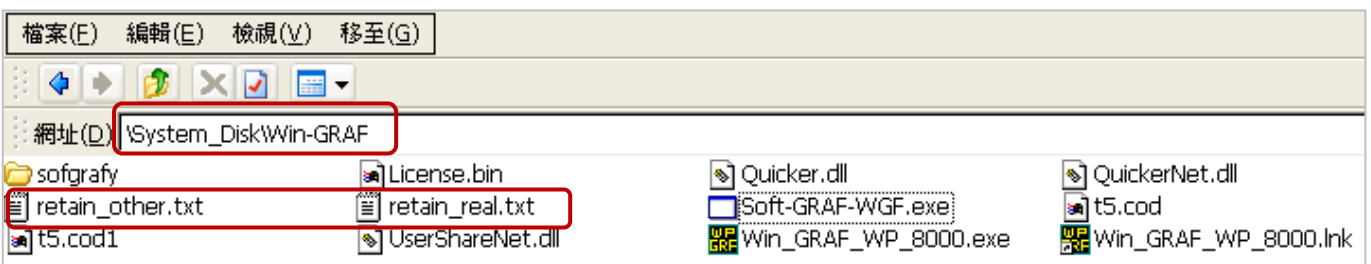
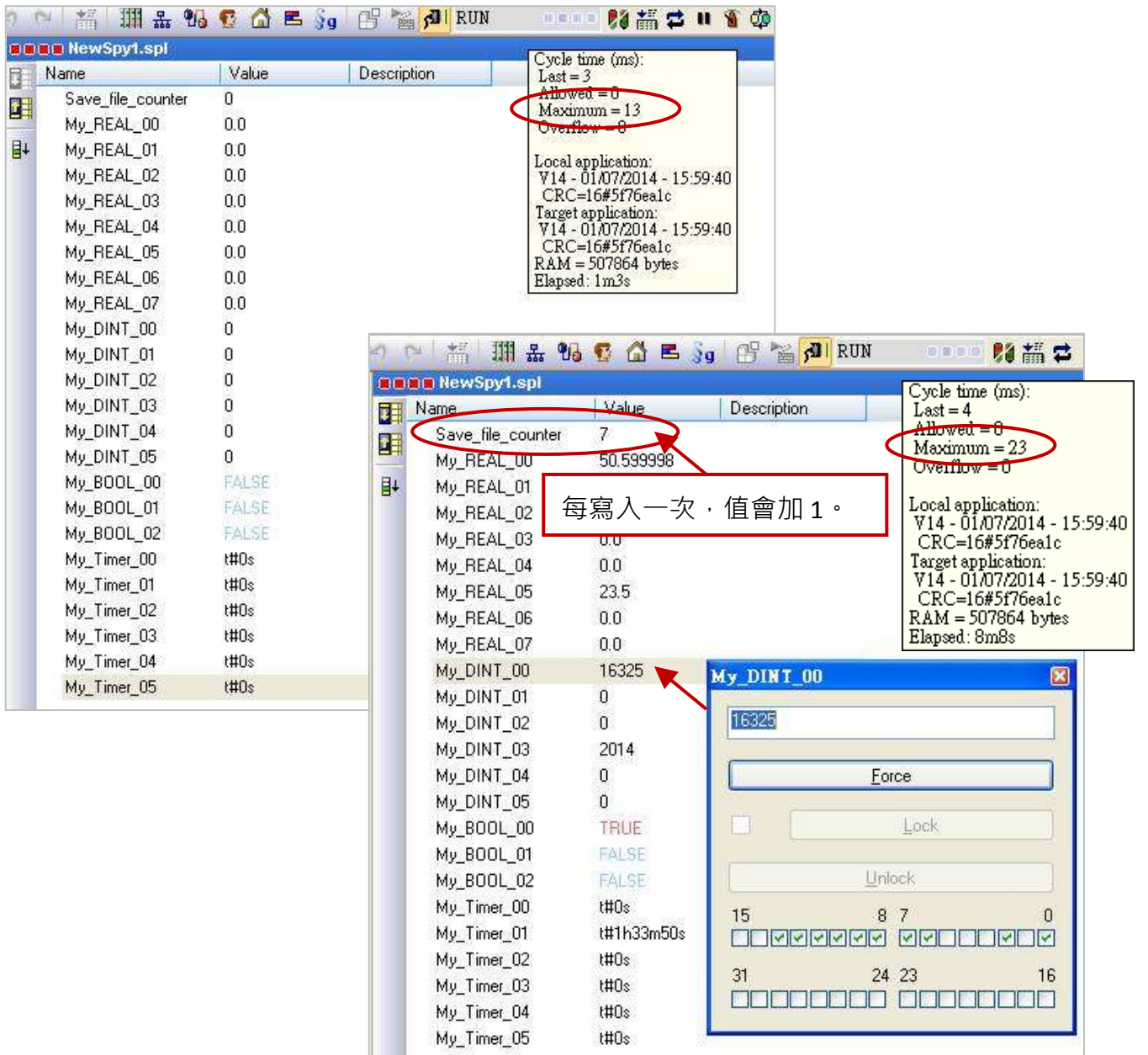
OTHER_VAR_now[6] := Any_to_DINT( My_BOOL_00 ) ;
OTHER_VAR_now[7] := Any_to_DINT( My_BOOL_01 ) ;
OTHER_VAR_now[8] := Any_to_DINT( My_BOOL_02 ) ;

OTHER_VAR_now[9] := Any_to_DINT( My_Timer_00 ) ;
OTHER_VAR_now[10] := Any_to_DINT( My_Timer_01 ) ;
OTHER_VAR_now[11] := Any_to_DINT( My_Timer_02 ) ;
OTHER_VAR_now[12] := Any_to_DINT( My_Timer_03 ) ;
OTHER_VAR_now[13] := Any_to_DINT( My_Timer_04 ) ;
OTHER_VAR_now[14] := Any_to_DINT( My_Timer_05 ) ;
(* ..... *)
```

測試程式:

測試之前，請確認已設定好 PAC IP，再編譯/下載程式到 PAC 中 (可參考 2.3.4 節 與 2.3.5 節)。與 PAC 連線時，在觀測清單中 (Spy list，設定方式可參考 11.3 節) 一開始所有數值為 "0" (或 "FALSE")，請任意的輸入幾個數值，當數值有變更時會在 PAC 內建立文字檔 (\System_disk\Win-GRAF\retain_real.txt 與 retain_other.txt) 並將資料寫入到該檔案中。

註: "Save_file_counter" 會顯示寫入檔案的次數，若此數值變化很快 (例如: 每秒/分鐘寫入好幾次)，則不適合此應用 (因為頻繁地在 \System_disk 下寫入檔案，會降低 PAC 效能)。



6.3 備份資料到 EEPROM

Win-GRAF 系列 PAC 內建有一個 EEPROM 記憶體可供使用者用來讀取與寫入資料，此資料不會因 PAC 關機而消失。相較於 SRAM 的讀寫方式，EEPROM 有以下優缺點：

註：有些 PAC 沒有 EEPROM 記憶體 (例如: WP-5238-CE7) 不支援 EEP_Read() 與 EEP_Write()。

優點： 提供 "可保存變數" (Retain Variable，參考 6.1 節) 之外，另一個可保存重要資料的方式。

缺點： 1. EEPROM 的讀寫動作很耗 CPU 時間 (約 5 ~ 50 ms)，而使用 "可保存變數" 方式，CPU 時間遠小於 1 ms。因此，**請勿**太頻繁的使用 "EEP_Read" 與 "EEP_Write" 函式，會增加許多 PAC Cycle 時間。

2. EEPROM 有寫入限制 (視 PAC 而定)，並不適合對同一筆資料進行多次寫入，因此，**請勿**在每個 PAC Cycle 內呼叫 "EEP_Write" 函式來進行寫入動作。

ST 語法： (下列分為 安全 與 危險 寫法)

```
(* 宣告 "FIRST_CYCLE" 為 "BOOL" 變數且初始值為 "TRUE"。
```

```
 宣告 "tmp_bool" 為 "BOOL" 變數。
```

```
 宣告 "New_Val" 與 "Old_Val" 為 "DINT" 變數。 *)
```

```
(* 於第一個 Cycle 讀取 EEPROM 一次 *)
```

```
if FIRST_CYCLE then
```

```
  FIRST_CYCLE := FALSE ; (* 表示不再是第一個 Cycle *)
```

```
  tmp_bool := EEP_Read ( 1 , New_Val ) ;
```

```
end_if ;
```

```
(* 安全寫法： 只有數值改變時才寫入到 EEPROM *)
```

```
if New_Val <> Old_Val then
```

```
  Old_Val := New_Val ;
```

```
  tmp_bool := EEP_Write ( 1 , New_Val ) ;
```

```
end_if ;
```

```
(* 危險寫法： EEPROM 可能很快被損毀 *)
```

```
(* 宣告 "FIRST_CYCLE" 為 "BOOL" 變數且初始值為 "TRUE"。
```

```
 宣告 "tmp_bool" 為 "BOOL" 變數。
```

```
 宣告 "New_Val" 與 "Old_Val" 為 "DINT" 變數。 *)
```

```
(* 於第一個 Cycle 讀取 EEPROM 一次 *)
```

```
if FIRST_CYCLE then
```

```
  FIRST_CYCLE := FALSE ; (* 表示不再是第一個 Cycle *)
```

```
  tmp_bool := EEP_Read ( 1 , New_Val ) ;
```

```
end_if ;
```

```
(* 危險寫法： 於每個 Cycle 內將 "New_Val" 值寫入到 EEPROM 一次 *)
```

```
tmp_bool := EEP_Write ( 1 , New_Val ) ;
```

6.3.1 EEP_READ (讀取 EEPROM)



使用小技巧:

按“F1”鍵，可查看詳細的設定說明。

Addr: (資料型態: “DINT”)

位址可設定為 1 ~ 1200。若 "@Name" 參數的資料型態為 64-bit 資料 (例如: LINT 或 LREAL) ，則 "Addr" 只能設定為 1001 ~ 1200。

@Name :

指定一個變數名稱，用來儲存讀到的 EEPROM 的資料。

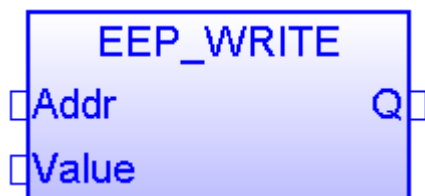
(請勿使用字串變數，資料型態可為 BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT, LREAL。)

Q:

資料型態為 “BOOL”，“TRUE”: 表示 OK ； “FALSE”: 表示錯誤。

若 "@Name" 參數的資料型態為 "REAL" 或 "LREAL" ，當讀取的資料不是實數或有其它錯誤發生，"Q" 會回傳 "FALSE"。若此資料非實數資料，"REAL" 或 "LREAL" 變數將會讀到 "0.0"。

6.3.2 EEP_WRITE (寫入 EEPROM)



Addr: (資料型態: “DINT”)

位址可設定為 1 ~ 1200。若 "Value" 參數的資料型態為 64-bit 資料 (例如: LINT 或 LREAL) ，則 "Addr" 只能設定為 1001 ~ 1200。

Value :

將資料寫入 EEPROM。

(請勿使用字串變數，資料型態可為 BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT, LREAL。)

Q:

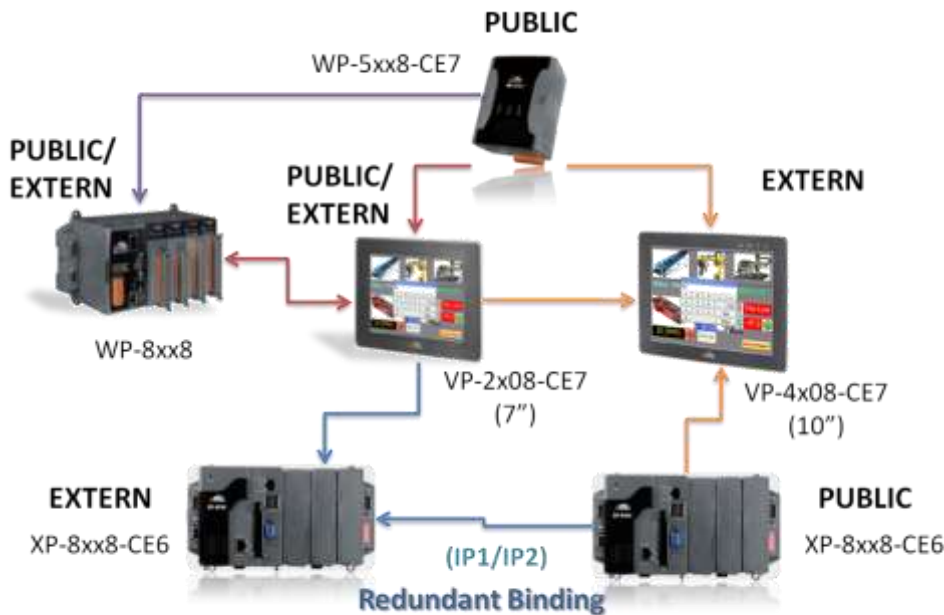
資料型態為 “BOOL”，“TRUE”: 表示 OK ； “FALSE”: 表示錯誤。

第 7 章 在 PAC 間互傳資料 (Data Binding)

"Binding" 功能可讓多台 Win-GRAF PAC 之間相互傳遞資料，資料是以事件觸發方式傳遞，比輪詢 (Polling) 方式更具效率。Win-GRAF 提供了兩種設定 Binding 方式：

- **PUBLIC:** 是指公開 PAC 自己的資料，或給同一台 PAC 內的 VB .net, C# 或 C 來使用。
- **EXTERN:** 是指從別台 PAC 取資料回來。

應用示意圖：



注意:

每台 Win-GRAF PAC 最多可使用的 "Binding" 功能 (EXTRN):

XPAC: Max. 32

WinPAC: Max. 16

ViewPAC: Max. 16

(詳細型號，請見 P1-1)

"PUBLIC" 設定步驟如下：

當 PAC 有指定 "PUBLIC" 區表示公開自己的資料。

1. 滑鼠點選工具列上的 "Open Binding Configuration" 按鈕來開啟 "Binding" 視窗。
2. 點選 "PUBLIC (:9000)" 來設定要公開的資料，"Address" 欄位無需填寫，"Port" 欄位固定為 "9000"，請勿更動。

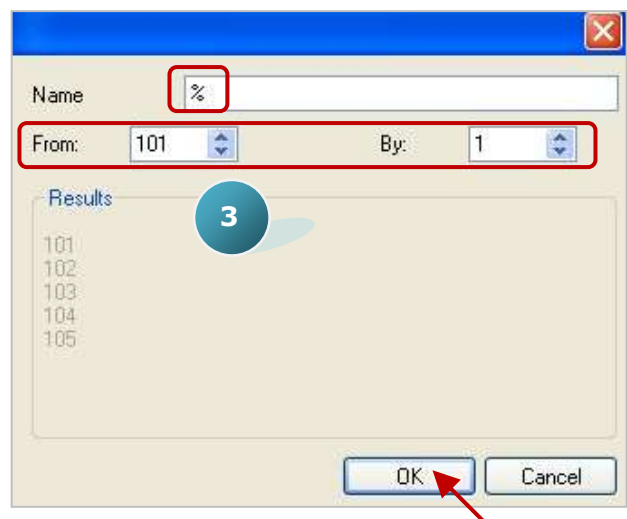
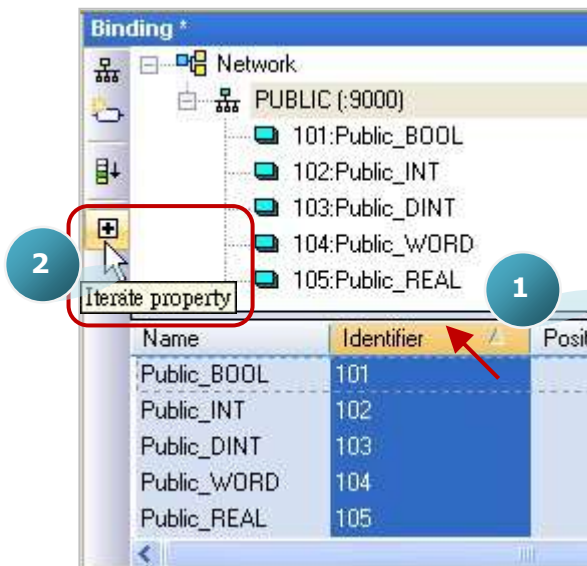
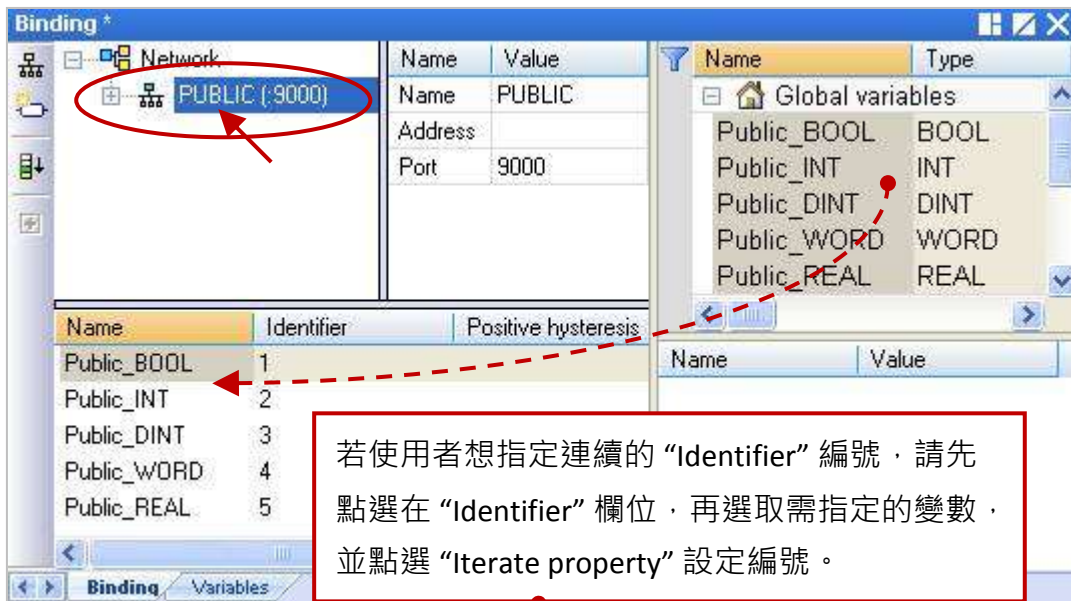


3. 在設定公開資料之前，您必須在變數區先建立好要公開的變數。滑鼠點選“Global variables”再按“Ins”鍵來新增變數項目，下表為此範例所使用的變數，您可依實際需求來設定，設定完成後，畫面如下。

變數名稱	資料型態
Public_BOOL	BOOL
Public_INT	INT
Public_DINT	DINT
Public_WORD	WORD
Public_REAL	REAL



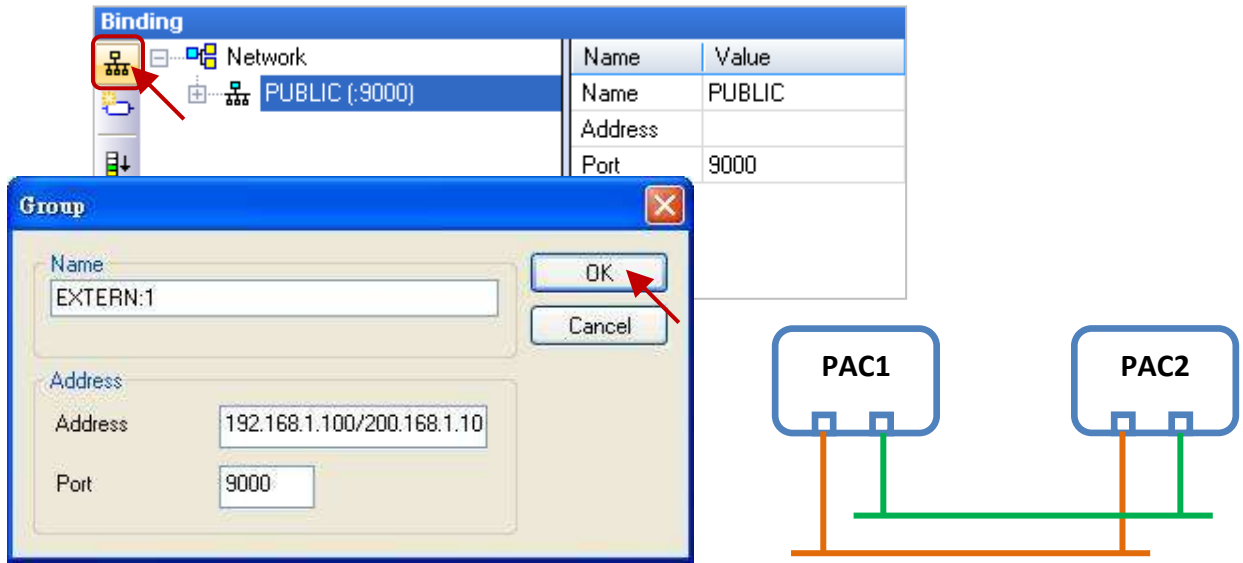
4. 滑鼠點選在“PUBLIC (:9000)”，再選取變數區欲公開的變數資料，並拖曳到“Name”區域。“Identifier”欄位會自動產生編號(若其它台 PAC 想取用該資料，需設定一樣的 ID 編號)。
注意: “PUBLIC”最多可使用 8192 個變數，“Identifier”編號只能是“1 ~ 8192”。



"EXTERN" 設定步驟如下:

當 PAC 有指定 "EXTERN" 區表示要從別台 PAC 取回它的資料。

5. 點選左邊的 "Insert Master/Port" 按鈕，將會出現 "Group" 視窗，請設定好以下欄位並按 "OK"。



Name: 可修改為所需的名稱。

Address: 輸入要取得資料的那台 PAC 的 IP 位址 (例如: "192.168.1.100")，也可輸入兩個 IP 位址 (例如: "192.168.1.100/200.168.1.100"; 對方 PAC 需有使用 2 個 Ethernet Port)，如此當其中之一發生問題，會嘗試去連第 2 個 IP 位址。

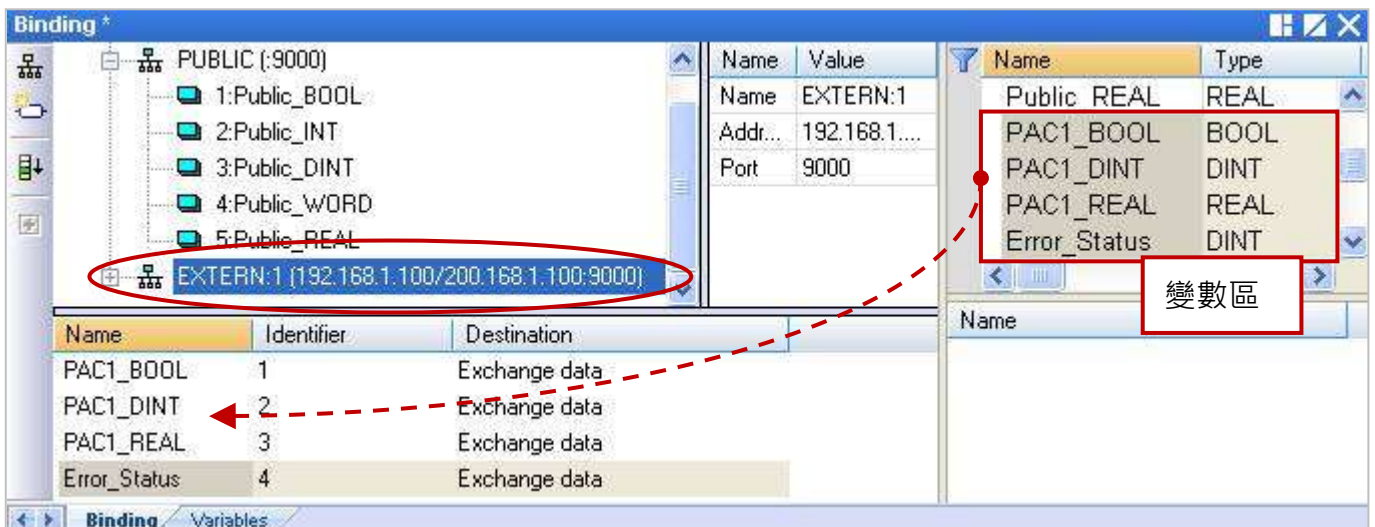
Port: 固定為 "9000"，請勿更動。

6. 您必須在變數區先設定好要取得的資料型態。
(可參考步驟 3 - 滑鼠點選 "Global variables" 再按 "Ins" 鍵來新增變數項目)，右邊列表是此範例想取用的資料型態，您可依實際需求來設定，設定完成後，畫面如下。

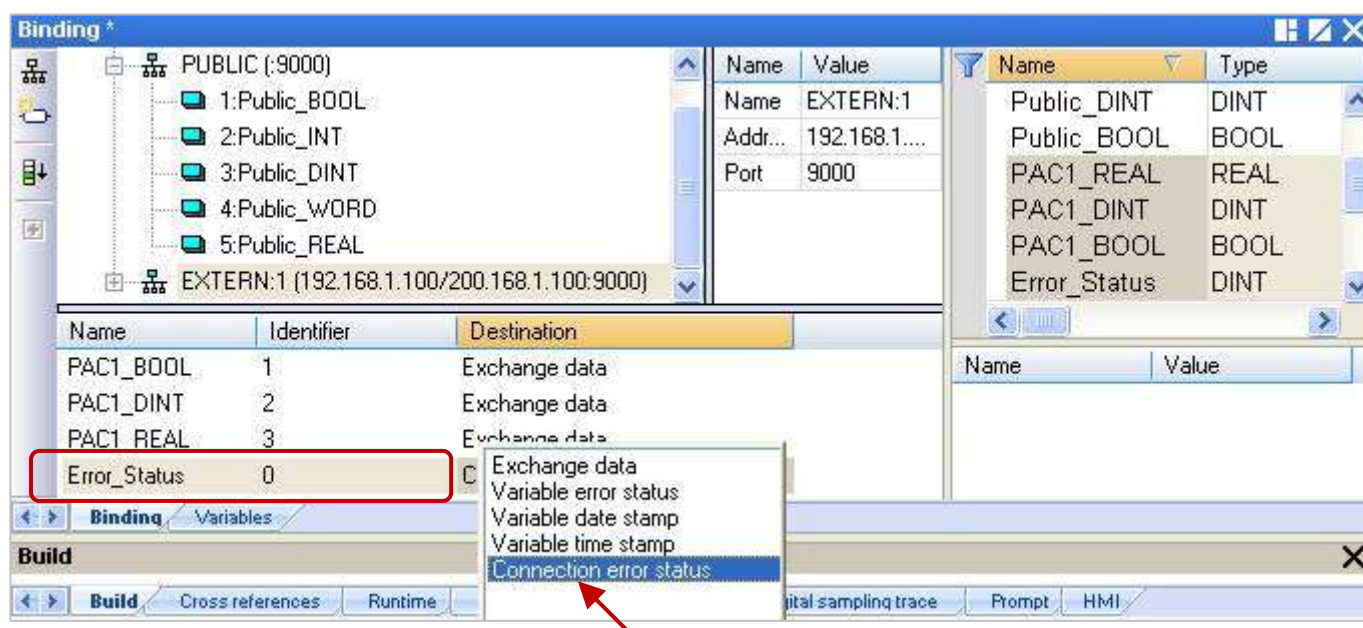
變數名稱	資料型態
PAC1_BOOL	BOOL
PAC1_DINT	DINT
PAC1_REAL	REAL
Error_Status	DINT

7. 請將所需的變數拖曳到 "EXTERN:1" 的 "Name" 區域。

註: "Identifier" 欄位會自動產生編號，請修改為和 要取得資料的那台 PAC 所開放的 ID 一致。



8. 如圖，“Error_Status”變數是用來判斷該 PAC 的通訊狀況，請將此 ID 設定為 "0" 再雙擊 "Destination" 欄位將其設定為 "Connection error status"。



註:

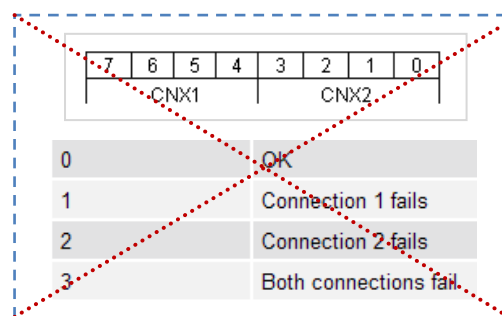
- a. 若“EXTERN”設定了 2 個 IP 位址 (步驟 5)，則“Error_Status”會回傳 2 個通訊狀態。如下表，以 8 個 bit 來表示通訊狀態，bit 0 ~ 3 表示第 1 個 IP 的通訊狀態 (bit 皆為 1 時，值為 15)，bit 4 ~ 7 表示第 2 個 IP 的通訊狀態 (bit 皆為 1 時，值為 240)，只要不等於 0，即表示有通訊異常。

IP2 的通訊狀態				IP1 的通訊狀態				狀態說明
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
0				0				通訊 OK
0				≠0 (1 ~ 15)				IP1 通訊異常
≠ 0 (16 ~ 240)				0				IP2 通訊異常
≠ 0				≠ 0				IP1 · IP2 皆通訊異常

- b. “Error_Status”的回傳值是一個整數值，以下提供了一個除法的判斷方式，將此數值除以 16，商數代表第 2 個 IP 的通訊狀態，餘數代表第 1 個 IP 的通訊狀態，不等於 0，即表示有通訊異常。例如，若“Error_Status” = 16，除以 16 的結果，商數 = 1 (≠0，IP2 通訊異常) 且 餘數 = 0 (IP1 通訊 OK); 若“Error_Status” = 3，除以 16 的結果，商數 = 0 (IP2 通訊 OK) 且 餘數 = 3 (≠0，IP1 通訊異常);

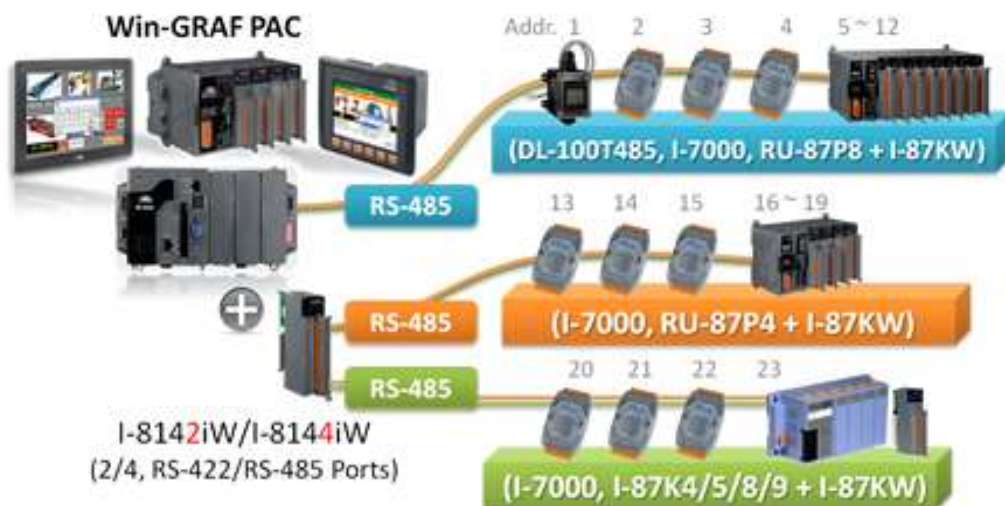
注意:

若按 "F1" 鍵開啟 "HTML Help" 說明，主題 - "Networked applications - Dual binding on redundant ETHERNET" 中 "Connection status" 與 "Variable status" 的說明是錯誤的，不符合 ICP DAS Win-GRAF PAC 的使用方式，請忽略該說明。



第 8 章 連接 DCON I/O 模組

Win-GRAF PAC 可透過 COM Port (RS-485) 來連接 ICP DAS 的 "I-7000" 與 "I-87KW" 遠端 DCON I/O 模組。每台 PAC 最多可啟用 16 個 DCON Port，而每個 Port 最多可連接 50 個遠端 DCON 模組 (建議不要超過 32 個)。若選用 "I-87KW" 系列 I/O 模組，必須搭配使用 RS-485 I/O 擴充單元 (例如: I-87K4/5/8/9 或 RU-87P4/8)。您可在泓格科技的網站上查看詳細的產品資訊：www.icpdas.com/root/product/solutions/remote_io/remote_io_products_tc.html



在開始使用 "I-7000" 或 "I-87KW" 遠端 DCON I/O 模組前，需先以 "DCON Utility" 軟體設定每一個模組的 Protocol (請選 DCON 模式)、Address (1 ~ 255)、Baudrate (需與 Win-GRAF PAC 一樣，建議設為 9600)、Checksum (需與 Win-GRAF PAC 設定一樣，為了通訊安全，建議設為 "Enabled")、Data format 與其它 Input/Output 設定 (依需求來設定)。

注意:

- A. 若 [I-7000](#) 及 [I-87KW](#) 是 AI 模組，則資料格式 (Data format) 必需設定為 "2's Complement"。
例如: I-7005, I-7013, I-7014D, I-7015, I-7016, I-7017R, I-7018Z, I-7019R, I-7033; I-87005W, I-87013W, I-87015W, I-87015PW, I-87016W, I-87017W, I-87017RCW, I-87017ZW, I-87017DW, I-87018W, I-87018RW, I-87018ZW, I-87019RW, I-87019ZW, ...等類比輸入模組。
- B. 若 [I-7000](#) 及 [I-87KW](#) 是 AO 模組，則資料格式 (Data format) 必需設定為 "Engineering"。
例如: I-7021, I-7022, I-7024, I-7024R; I-87024W, I-87024UW, I-87024CW, I-87028UW, I-87028CW, I-87028VW, I-87028VW-20V 等類比輸出模組。

"DCON Utility" 是一個方便好用的軟體工具，可以協助網路搜尋、設定與測試 I/O 模組。請至以下網址取得軟體程式與使用手冊：

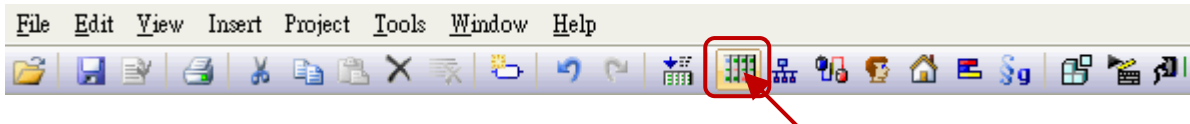
www.icpdas.com/products/dcon/introduction.htm

接下來將說明，Win-GRAF Workbench 中的設定方式。

8.1 設定 "DCON" I/O 卡

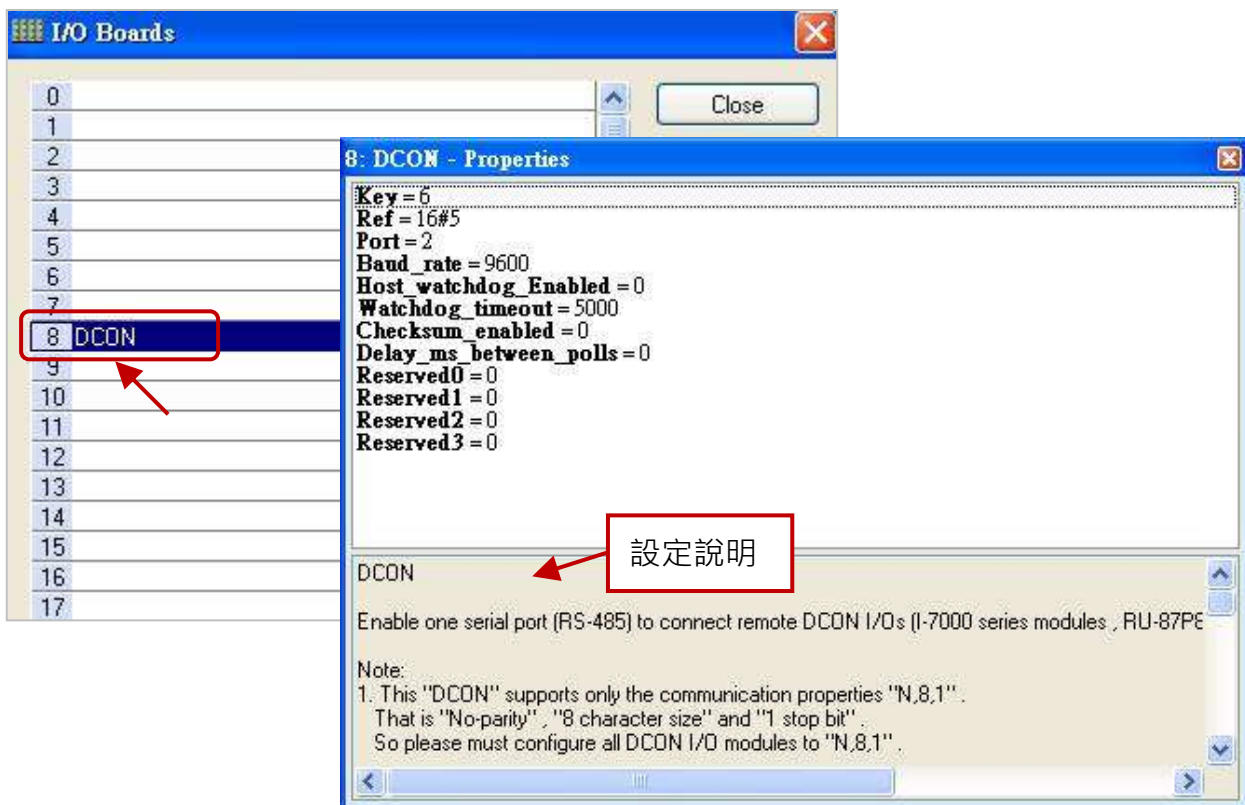
"DCON" 可用來啟用一個 RS-485 Port 來連接遠端的 DCON I/O 模組 (例如: I-7000 系列模組、RU-87P8 I/O 擴充單元 + I-87KW I/O 模組 或 I-87K8 I/O 擴充單元 + I-87KW I/O 模組)。如需啟用多個 DCON Port，請設定多個 "DCON" I/O 卡。(一台 PAC 最多可啟用 16 個 "DCON")

1. 點選 Win-GRAF 工具列的 "Open I/Os" 按鈕來開啟 "I/O Boards" 視窗。



2. 於 Slot8 加入 "DCON" I/O 卡 (參考第四章)，再以滑鼠雙擊該項目來開啟 "Properties" 視窗。

注意: Slot0 ~ 7 是保留給 PAC I/O 模組，Slot8 (含) 以上供給其它用途使用。



參數說明:

註: "DCON" 僅支援通訊屬性為 "N,8,1"，表示 "無同位元"、"8 個資料位元" 與 "1 個停止位元"，因此請將所有的 DCON I/O 模組設置為 "N,8,1"。

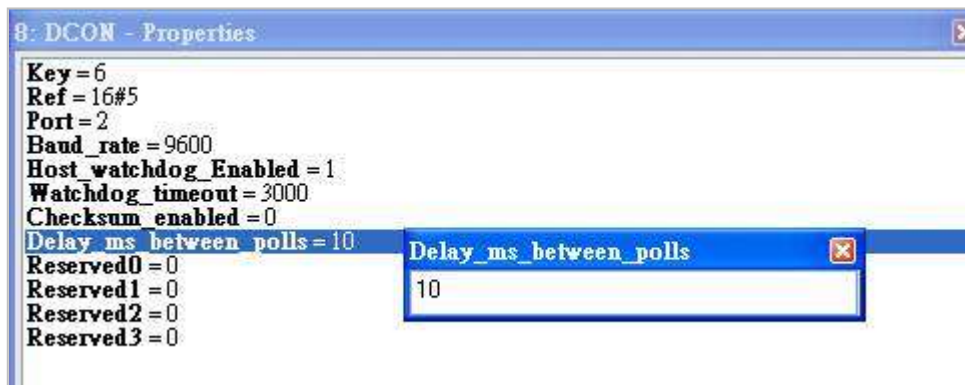
Port: COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。

Baud_rate: 通訊速率，可設定為 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 (bps)，若設定為其它值，將會採用預設值 "9600"。

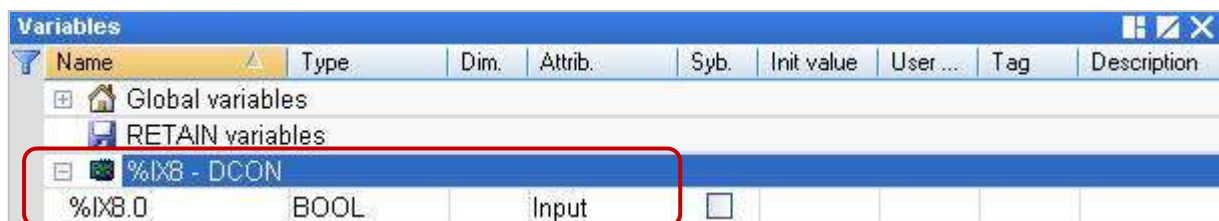
Host_watchdog_Enabled: 1: 表示啟用 Host-watchdog，0: 表示不啟用
設定為非 0 的值，將會採用 "1"。

- Watchdog_timeout:** 單位: ms · 可設定為 “3000 ~ 25500” · 設定為大於 “25500” 將會採用 25500 ms (即 25.5 秒); 設定為小於 “3000” 將會採用 3000 ms (即 3 秒) 。若 "Host_watchdog_Enabled" 設定為 "0" 會忽略此設定。
- Checksum_enabled:** 1: 表示啟用 · 0: 表示不啟用。
若設定為非 0 的值 · 將會採用 "1"。(為了通訊安全, 建議啟用 Checksum)
- Delay_ms_between_polls:** 單位: ms · 預設值為 0 ms · 有效範圍為 "0 ~ 1000"。設定為小於 “0” 將會採用 0 ms ; 設定為大於 “1000” 將會採用 1000 ms 。
若沒有連接無線模組 · 請設定為較小的值 (例如: 0 ~ 10) 。
若連接無線模組 (例如: ICP DAS 的 [ZT-2570](#)、[ZT-2571 Ethernet/RS-485/RS-232 至 ZigBee 轉換器](#) 或 [ZB-2000 系列 DIO/AIO 模組](#)) · 請設定為較大的值 (例如: 30 ~ 100 或 其它值) · 而設定較大的值 · 輪詢 (Polling) 效率會較慢。

3. 滑鼠雙擊欲設定的項目 · 並輸入設定值。

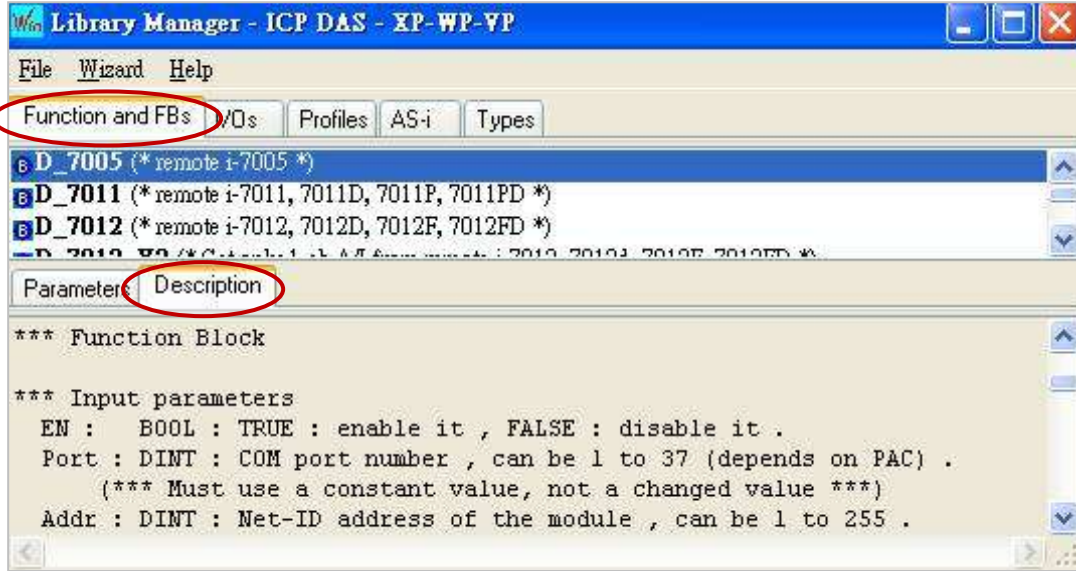


4. 在 “I/O Boards” 視窗內連上 “DCON” 後 · 會自動在 “Variables” 視窗中新增 1 個 “BOOL” 輸入變數 · 當 Win-GRAF 有連上 PAC 時 · 會顯示出 COM Port 的連接狀態。(TRUE: 表示 OK ; FALSE: 表示錯誤。)

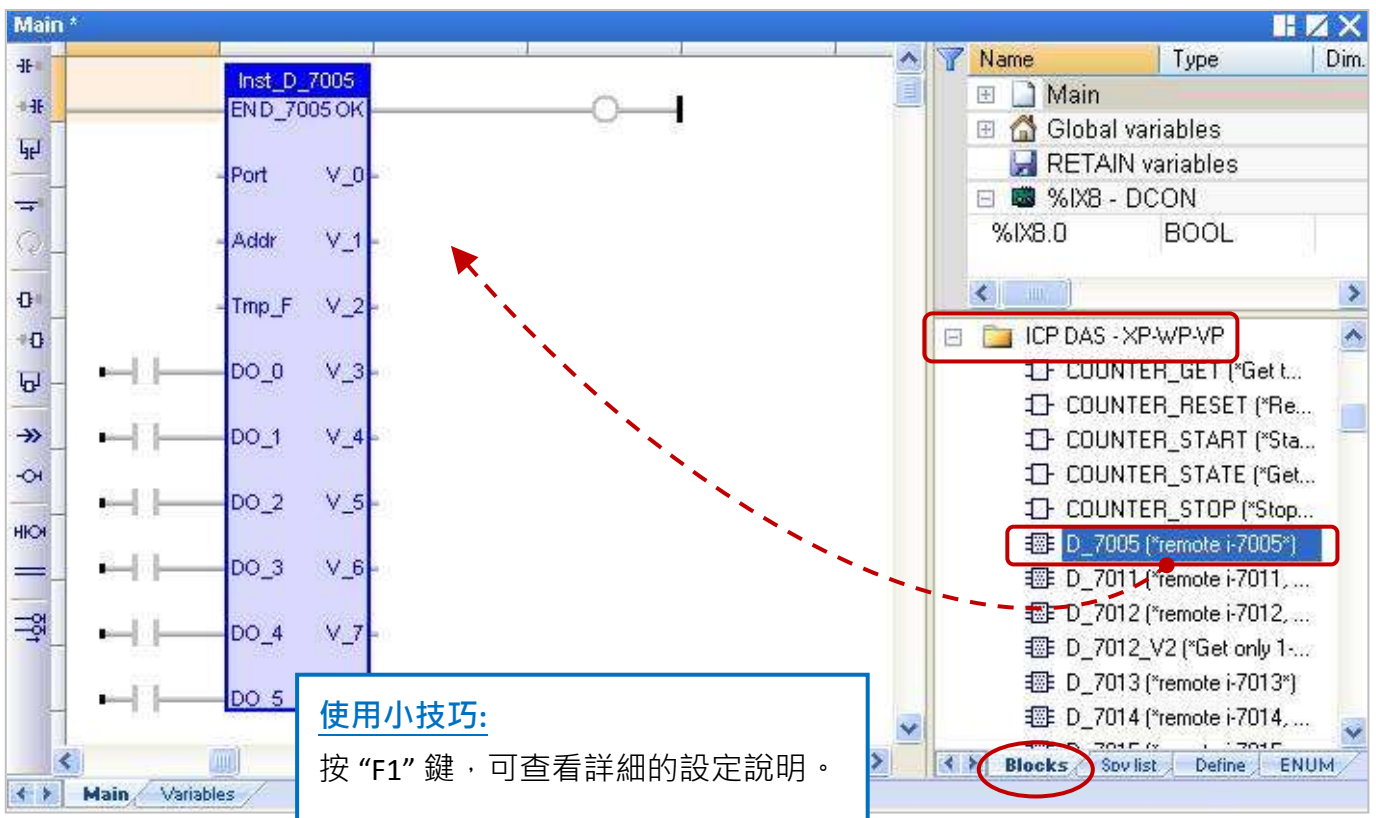


8.2 使用 I/O 功能方塊

Win-GRAF 支援許多 ICP DAS 的 DCON 遠端 I/O 模組，您可開啟“程式庫管理員”(參考 1.2.3 節)或在 I/O 功能方塊上按“F1”鍵來查看這些 I/O 功能方塊的設定說明。本章節將介紹“D_7065”，“D_7018Z”，“D_7083”，“D_87084_freq”，“D_87084_cnt4”，“D_87084_cnt8”，“DL_100T485”...等 功能方塊。



在 LD 程式 - 功能方塊區中，可在“Blocks”面板中展開“ICP DAS - XP-WP-VP”資料夾，裡面列有許多函式與功能方塊，您可選擇所需的項目，並將它拖曳到程式編輯區來使用。



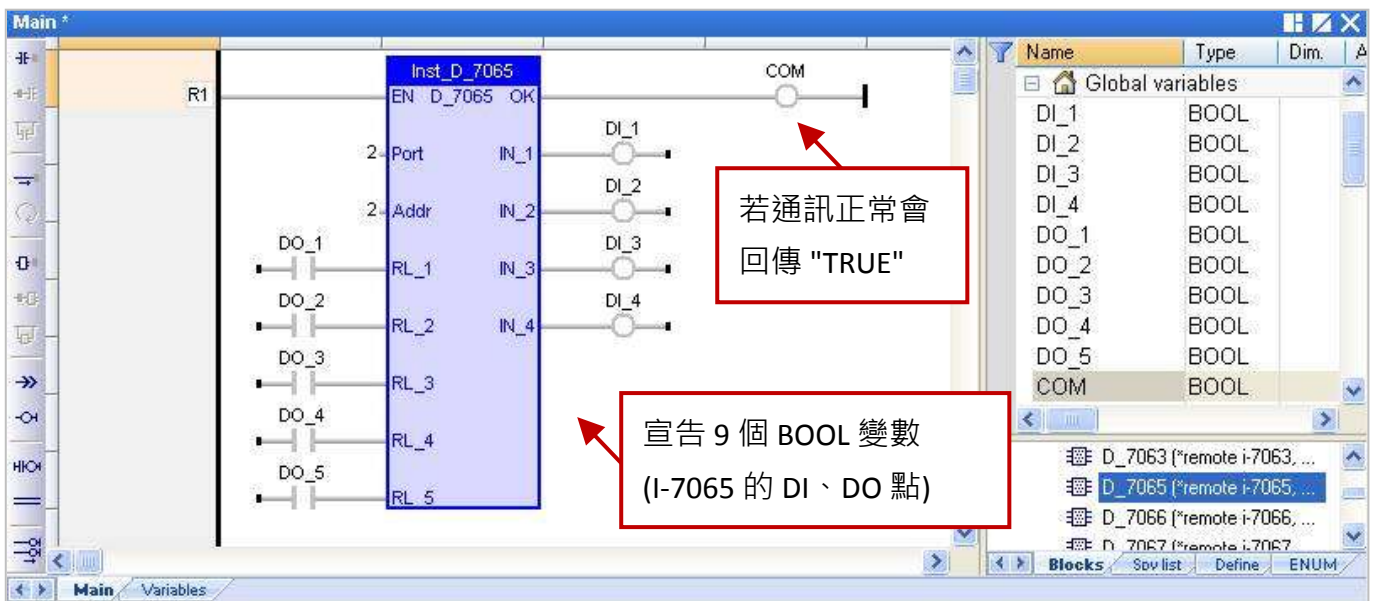
8.2.1 “D_7065” 功能方塊

“D_7065” 可用來連結一個遠端的 I-7065, I-7065D 繼電器 (Relay) 輸出模組 與 I-7065A, I-7065AD, I-7065B, I-7065BD (固態繼電器輸出模組)。

註:

1. 所有連接的 DCON I/O 模組需使用 "DCON Utility" 軟體 (見 [p8-1](#)) 設定過一次。
2. 請在 “I/O boards” 視窗加入 “DCON” (見 [8.1 節](#))，並填入正確的參數 (例如: Port、Baud_rate...)。
3. 只有在通訊狀態為 “TRUE” 的情況下 (若 “OK” 會回傳 “TRUE”)，DI 通道的回傳值才具意義。
4. 可參考 [第 12 章](#)，點選功能表 “File” > “Add Existing Project” > “From Zip”，來回存出貨光碟中的範例程式 (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO_D_7065.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 I-7065 (位址 = 2)，使用 4 個數位輸入通道 與 5 Relay 輸出通道。



輸入參數:

- EN:** 資料型態: BOOL，設定為 “TRUE”: 啟用；設定為 “FALSE”: 不啟用。
- Port:** 資料型態: DINT，COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。
(** 需設定為常數，即不可變動的值得值 **)
- Addr:** 資料型態: DINT，模組的 Net-ID 位址 (可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值得值 **)
- RL_1 ~ RL_5:** 資料型態: BOOL，5 通道 DO 值。

輸出參數:

- OK:** 資料型態: BOOL，“TRUE”: 表示通訊正常；“FALSE”: 表示通訊失敗。
- IN_1 ~ IN_4:** 資料型態: BOOL，4 通道 DI 值。

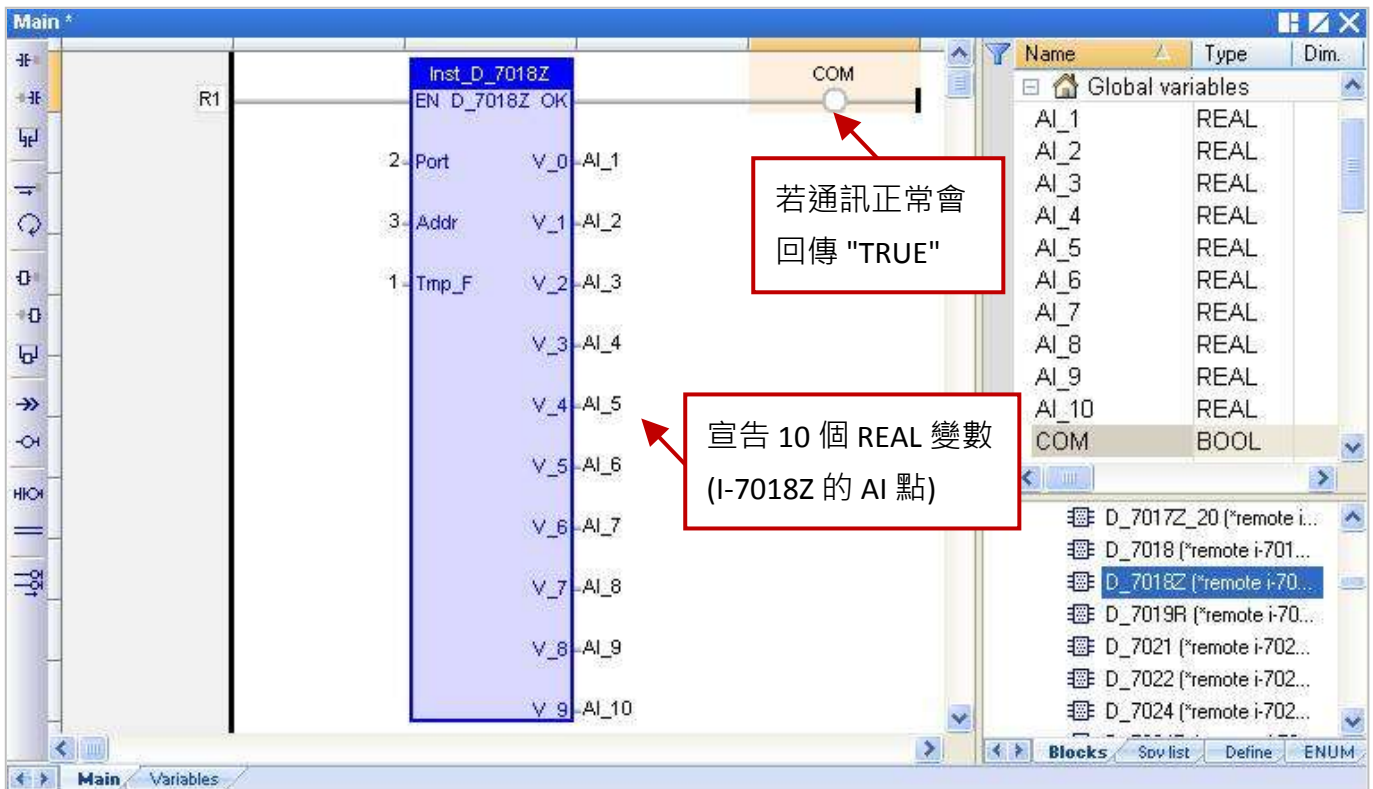
8.2.2 “D_7018Z” 功能方塊

“D_7018Z” 可用來連結一個遠端的 I-7018Z 模組，此模組是 10 通道熱電耦類比輸入模組，可用來量測電壓、電流或溫度，通道可個別設定並具有斷線偵測、過電壓保護功能。

註:

1. 請先使用 "DCON Utility" 軟體 (見 [P8-1](#)) 設定好該模組的適當參數 (例如: Address、Baudrate...)，AI 模組的資料格式需設定為 "2's Complement"，否則 Win-GRAF PAC 無法正確地讀取該值。
2. 請在 "I/O boards" 視窗加入 "DCON" (見 [8.1 節](#))，並填入正確的參數 (例如: Port、Baud_rate...)。
3. 只有在通訊狀態為 "TRUE" 的情況下 (若 "OK" 會回傳 "TRUE")，AI 通道的回傳值才具意義。
4. 可參考 [第 12 章](#)，點選功能表 "File" > "Add Existing Project" > "From Zip"，來回存出貨光碟中的範例程式 (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO_D_7018z.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 I-7018Z (位址 = 3)，且用來量測攝氏溫度。



輸入參數:

- EN:** 資料型態: BOOL，設定為 "TRUE": 啟用；設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT，COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。
(** 需設定為常數，即不可變動的值得值 **)
- Addr:** 資料型態: DINT，模組的 Net-ID 位址 (可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值得值 **)

Tmp_F: 資料型態: DINT, 溫度格式可設定為 1 或 2。

1: 表示為攝氏溫度

2: 表示為華氏溫度

若設定為其它值, 則會取用 "1: 攝氏溫度"。

輸出參數:

OK: 資料型態: BOOL, "TRUE": 表示通訊正常; "FALSE": 表示通訊失敗。

V_0 ~ V_9: 資料型態: REAL, 10 通道 AI 值。

若在 "DCON Utility" 選用了通道為電壓類型, 表示該回傳值的單位為 "V"。

例如, 回傳值為 0.85421 表示 0.85421 V 或 854.21 mV。

若在 "DCON Utility" 選用了通道為電流類型, 表示該回傳值的單位為 "mA"。

例如, 回傳值為 1.5567 表示 1.5567 mA。

若在 "DCON Utility" 選用了通道為溫度類型, 表示該回傳值的單位為 "度"。

例如, 回傳值為 25.75 表示 25.75 度。

斷線偵測:

若溫度值大於 "9000.0" 表示,

1. 溫度感測器可能斷線。
2. 溫度感測器可能損毀。
3. DCON 模組的設定與溫度感測器不符。
4. 感測器量到錯誤的電阻值。

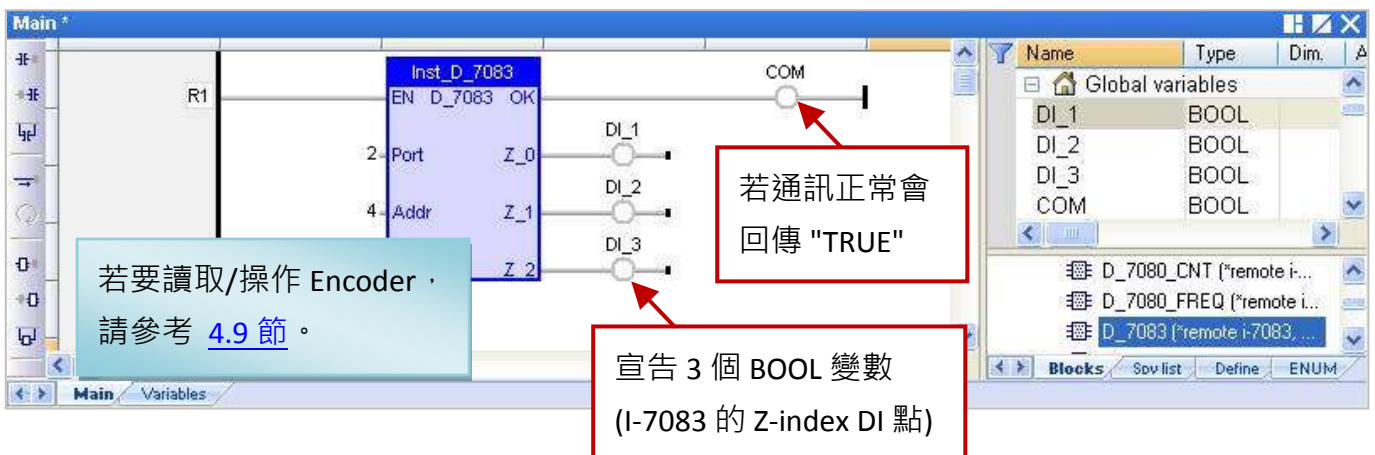
8.2.3 “D_7083” 功能方塊

“D_7083” 可用來連結一個遠端的 I-7083, I-7083D, I-7083B, I-7083BD 模組，此模組是 3 軸 32 位元編碼器輸入模組。

註:

1. 為了取得 I-7083, I-7083D, I-7083B, I-7083BD 模組的 Encoder 值，需先使用 "D_7083" 功能方塊之後，再搭配 "Counter_Start", "Counter_Stop", "Counter_Get", "Counter_State" 與 "Counter_Reset" 函式 (可參考 4.9 節) 來操作這些模組的 Encoder 通道。
2. 請先使用 "DCON Utility" 軟體 (見 P8-1) 設定好該模組的適當參數 (例如: Address、Baudrate...)，AI 模組的資料格式需設定為 "2's Complement"，否則 Win-GRAF PAC 無法正確地讀取該值。
3. 請在 "I/O boards" 視窗加入 "DCON" (見 8.1 節)，並填入正確的參數 (例如: Port、Baud_rate...)。
4. 只有在通訊狀態為 "TRUE" 的情況下 (若 "OK" 會回傳 "TRUE")，AI 通道的回傳值才具意義。
5. 可參考 第 12 章，點選功能表 "File" > "Add Existing Project" > "From Zip"，來回存出貨光碟中的範例程式 (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO_D_7083.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 I-7065 (位址 = 4)，使用 3 個數位輸入通道。



輸入參數:

- EN:** 資料型態: BOOL，設定為 "TRUE": 啟用；設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT，COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。
(** 需設定為常數，即不可變動的值 **)
- Addr:** 資料型態: DINT，模組的 Net-ID 位址 (可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值 **)

輸出參數:

- OK:** 資料型態: BOOL，"TRUE": 表示通訊正常；"FALSE": 表示通訊失敗。
- Z_0 ~ Z_2:** 資料型態: BOOL，3 軸 Z-index DI 值。

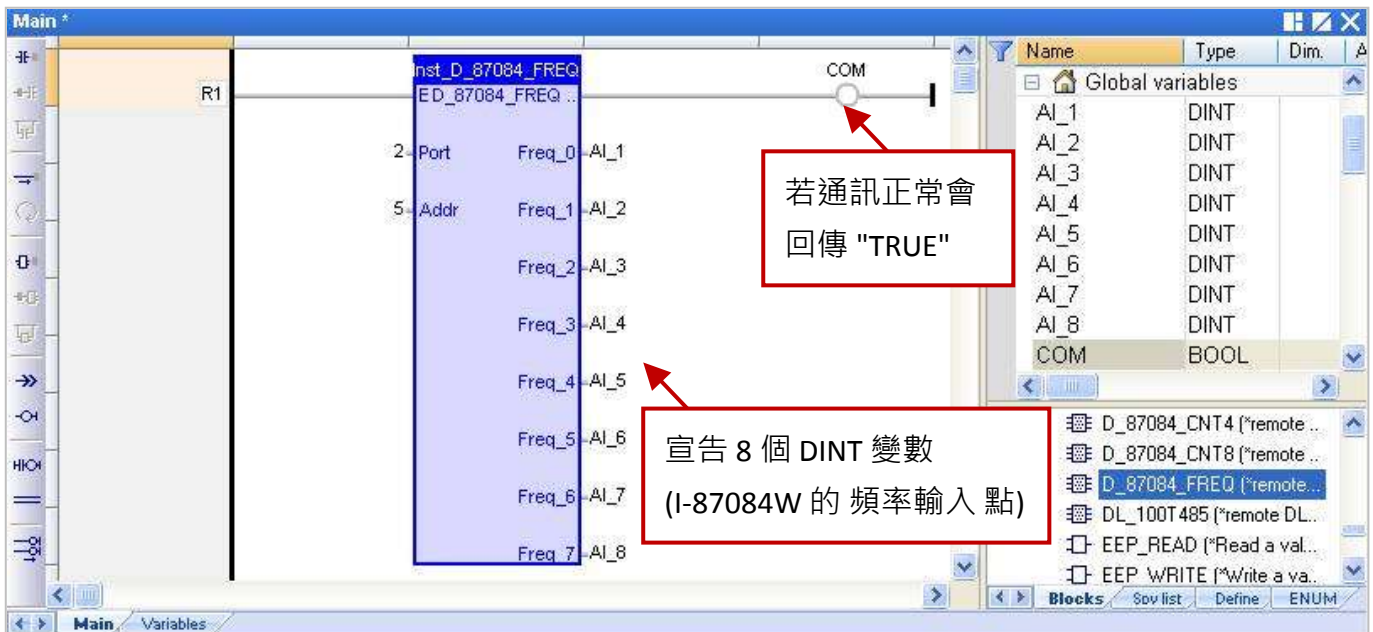
8.2.4 “D_87084_FREQ” 功能方塊

"D_87084_freq" 可用來連結一個在 I/O 擴充單元 (例如: I-87K4/5/8/9、RU-87P4 或 RU-87P8) 上的 I-87084W 模組，並用來量測 8 通道的頻率值。

註:

1. 請先使用 "DCON Utility" 軟體 (見 [P8-1](#)) 設定好該模組的適當參數 (例如: Address、Baudrate...)，頻率的資料格式需設定為 "Hex format"，否則該功能將無效。
2. 請在 "I/O boards" 視窗加入 "DCON" (見 [8.1 節](#))，並填入正確的參數 (例如: Port、Baud_rate...)。
3. 只有在通訊狀態為 "TRUE" 的情況下 (若 "OK" 會回傳 "TRUE")，AI 通道的回傳值才具意義。
4. 可參考 [第 12 章](#)，點選功能表 "File" > "Add Existing Project" > "From Zip"，回存出貨光碟中的範例程式 (CD-ROM:\Napdos\Win-GRAF\demo-project\DEMO_D_87084_FR.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 I-87084W (位址 = 5)，使用 8 個頻率輸入通道。



輸入參數:

- EN:** 資料型態: BOOL，設定為 "TRUE": 啟用；設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT，COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。
(** 需設定為常數，即不可變動的值得值 **)
- Addr:** 資料型態: DINT，模組的 Net-ID 位址 (可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值得值 **)

輸出參數:

- OK:** 資料型態: BOOL，"TRUE": 表示通訊正常；"FALSE": 表示通訊失敗。
- Freq_0 ~ Freq_7:** 資料型態: DINT，8 通道頻率值 (單位: Hz)。

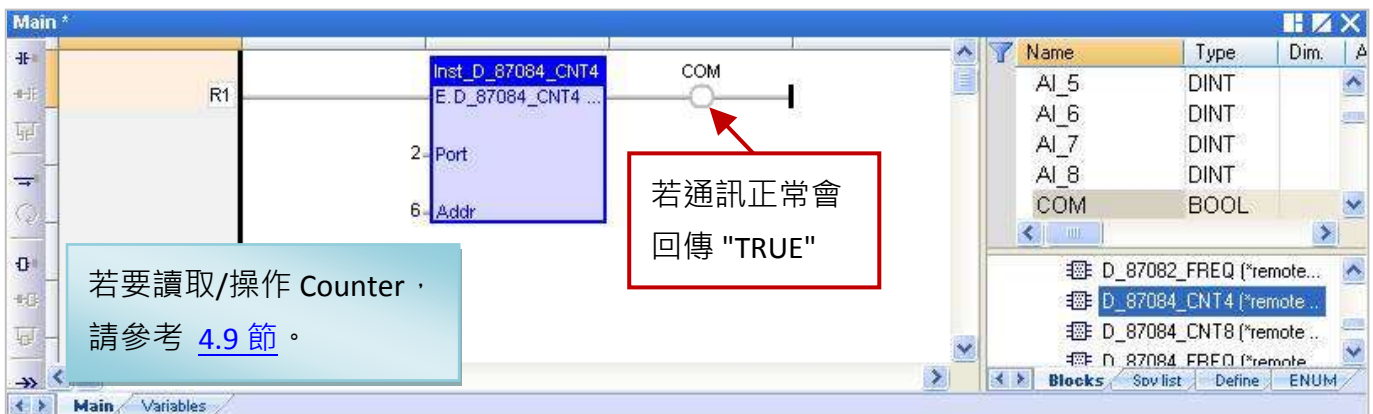
8.2.5 “D_87084_CNT4” 功能方塊

"D_87084_CNT4" 可用來連結一個在 I/O 擴充單元 (例如: I-87K4/5/8/9、RU-87P4 或 RU-87P8) 上的 I-87084W 模組，並用來量測 4 通道的計數值。

註:

1. 請先使用 "DCON Utility" 軟體 (見 [P8-1](#)) 設定好該模組的適當參數 (例如: Address、Baudrate...)，頻率的資料格式需設定為 "Hex format"，否則該功能將無效。
2. 請在 "I/O boards" 視窗加入 "DCON" (見 [8.1 節](#))，並填入正確的參數 (例如: Port、Baud_rate...)。
3. 為了由遠端 I-87084W 模組取得 4 通道的 Counter 值，需先使用 "D_87084_CNT4" 功能方塊，並搭配 "Counter_Start", "Counter_Stop", "Counter_Get", "Counter_State" 與 "Counter_Reset" 函式 (可參考 [4.9 節](#)) 來操作該模組的 Counter 通道。
4. 只有在通訊狀態為 "TRUE" 的情況下 (若 "OK" 會回傳 "TRUE")，AI 通道的回傳值才具意義。
5. 可參考 [第 12 章](#)，點選功能表 "File" > "Add Existing Project" > "From Zip"，回存出貨光碟中的範例程式 (CD-ROM:\Napdos\Win-GRAF\demo-project\DEMO_D_87084_C4.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 I-87084W (位址 = 6)，使用 4 個計數通道。



輸入參數:

- EN:** 資料型態: BOOL，設定為 "TRUE": 啟用；設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT，COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。
(** 需設定為常數，即不可變動的值 **)
- Addr:** 資料型態: DINT，模組的 Net-ID 位址 (可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值 **)

輸出參數:

- OK:** 資料型態: BOOL，"TRUE": 表示通訊正常；"FALSE": 表示通訊失敗。

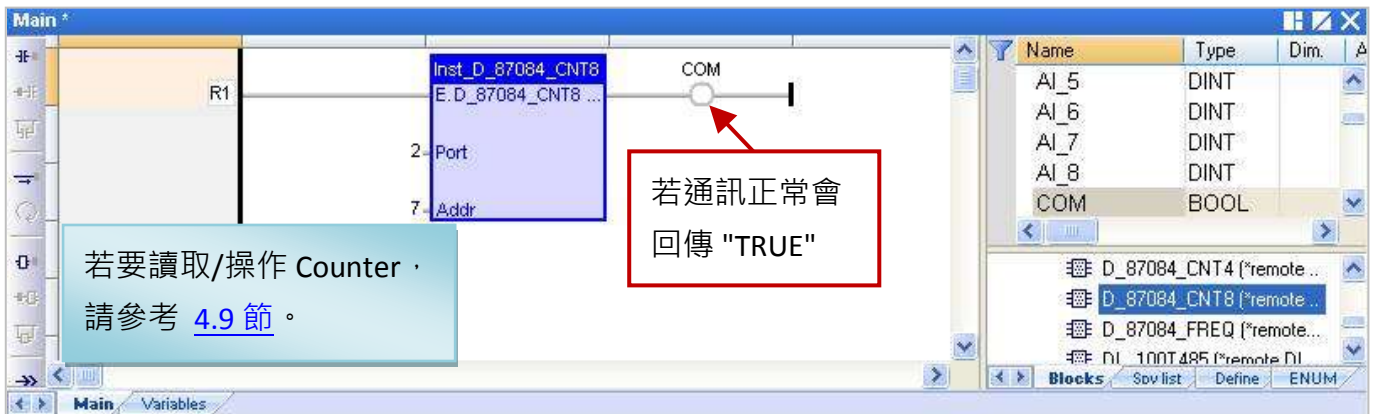
8.2.6 "D_87084_CNT8" 功能方塊

"D_87084_CNT8" 可用來連結一個在 I/O 擴充單元 (例如: I-87K4/5/8/9、RU-87P4 或 RU-87P8) 上的 I-87084W 模組，並用來量測 8 通道的計數值。

註:

1. 請先使用 "DCON Utility" 軟體 (見 [P8-1](#)) 設定好該模組的適當參數 (例如: Address、Baudrate...)，頻率的資料格式需設定為 "Hex format"，否則該功能將無效。
2. 請在 "I/O boards" 視窗加入 "DCON" (見 [8.1 節](#))，並填入正確的參數 (例如: Port、Baud_rate...)。
3. 為了由遠端 I-87084W 模組取得 8 通道的 Counter 值，需先使用 "D_87084_CNT8" 功能方塊，並搭配 "Counter_Start", "Counter_Stop", "Counter_Get", "Counter_State" 與 "Counter_Reset" 函式 (可參考 [4.9 節](#)) 來操作該模組的 Counter 通道。
4. 只有在通訊狀態為 "TRUE" 的情況下 (若 "OK" 會回傳 "TRUE")，AI 通道的回傳值才具意義。
5. 可參考 [第 12 章](#)，點選功能表 "File" > "Add Existing Project" > "From Zip"，回存出貨光碟中的範例程式 (CD-ROM:\Napdos\Win-GRAF\demo-project\DEMO_D_87084_C8.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 I-87084W (位址 = 7)，使用 8 個計數通道。



輸入參數:

- EN:** 資料型態: BOOL，設定為 "TRUE": 啟用；設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT，COM Port 編號 (可設定為 1 ~ 37，視 PAC 而定)。
(** 需設定為常數，即不可變動的值 **)
- Addr:** 資料型態: DINT，模組的 Net-ID 位址 (可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值 **)

輸出參數:

- OK:** 資料型態: BOOL，"TRUE": 表示通訊正常；"FALSE": 表示通訊失敗。

8.2.7 “DL_100T485” 功能方塊

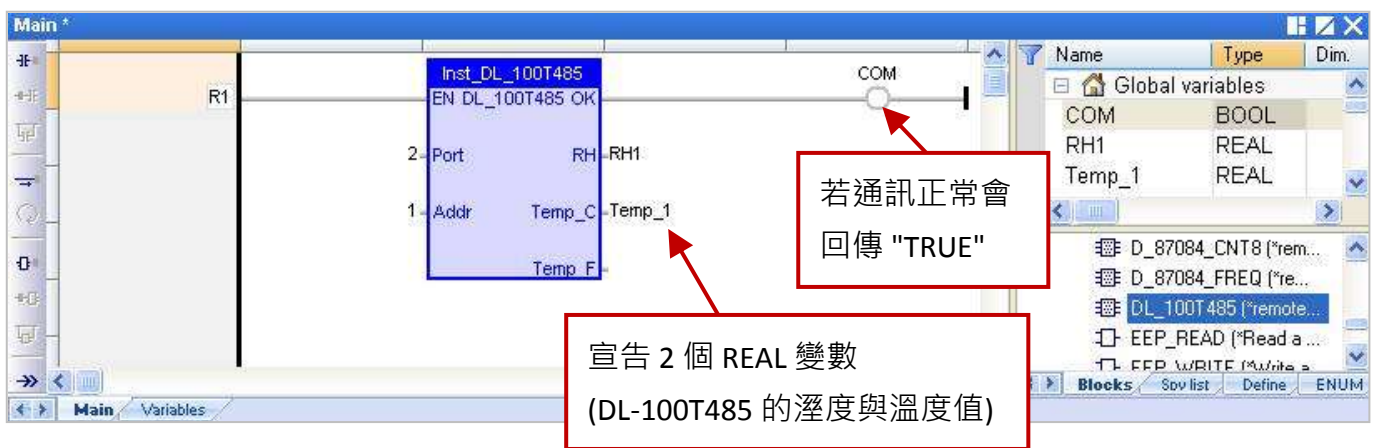
"DL_100T485" 可用來連結一個遠端的 DL_100T485 模組，來讀取溫度與濕度值。

產品網頁: www.icpdas.com/root/product/solutions/remote_io/rs-485/dl_series/dl-100t485.html

註:

1. 請先使用隨貨光碟中的 "DL-100T485 Utility" 軟體，設定好該模組的適當參數 (例如: Module ID) ，DL-100T485 預設 Address (ID) 為 "1" ，Baudrate 為 "9600" ，Checksum 為 "Disable" 。
2. 請在 "I/O boards" 視窗加入 "DCON" (見 8.1 節) ，並填入正確的參數 (例如: Port 、Baud_rate...) 。
3. 只有在通訊狀態為 "TRUE" 的情況下 (若 "OK" 會回傳 "TRUE") ，AI 通道的回傳值才具意義。
4. 可參考 第 12 章，點選功能表 "File" > "Add Existing Project" > "From Zip" ，回存出貨光碟中的範例程式 (CD-ROM:\Napdos\Win-GRAF\demo-project\DEMO_DL_100T485.zip) 並查看詳細的程式內容。

假設: 使用 PAC 的 COM2 來連接 DL_100T485 (位址 = 1) ，用來量測濕度值。



輸入參數:

- EN:** 資料型態: BOOL ，設定為 "TRUE": 啟用；設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT ，COM Port 編號 (可設定為 1 ~ 37 ，視 PAC 而定)。
(** 需設定為常數，即不可變動的值得值 **)
- Addr:** 資料型態: DINT ，模組的 Net-ID 位址 (預設為 "1" ，可設定為 1 ~ 255)。
(** 需設定為常數，即不可變動的值得值 **)

輸出參數:

- OK:** 資料型態: BOOL ，"TRUE": 表示通訊正常；"FALSE": 表示通訊失敗。
- RH:** 資料型態: REAL ，回傳值為相對溼度 (單位: %)。
例如: 回傳值為 "45.7" 表示 45.7%。
- Temp_C:** 資料型態: REAL ，回傳值為攝氏溫度
例如: 回傳值為 "25.7" 表示 25.7 °C。
- Temp_F:** 資料型態: REAL ，回傳值為華氏溫度
例如: 回傳值為 "78.26" 表示 78.26 °F。

8.2.8 “D_GPS721” 功能方塊

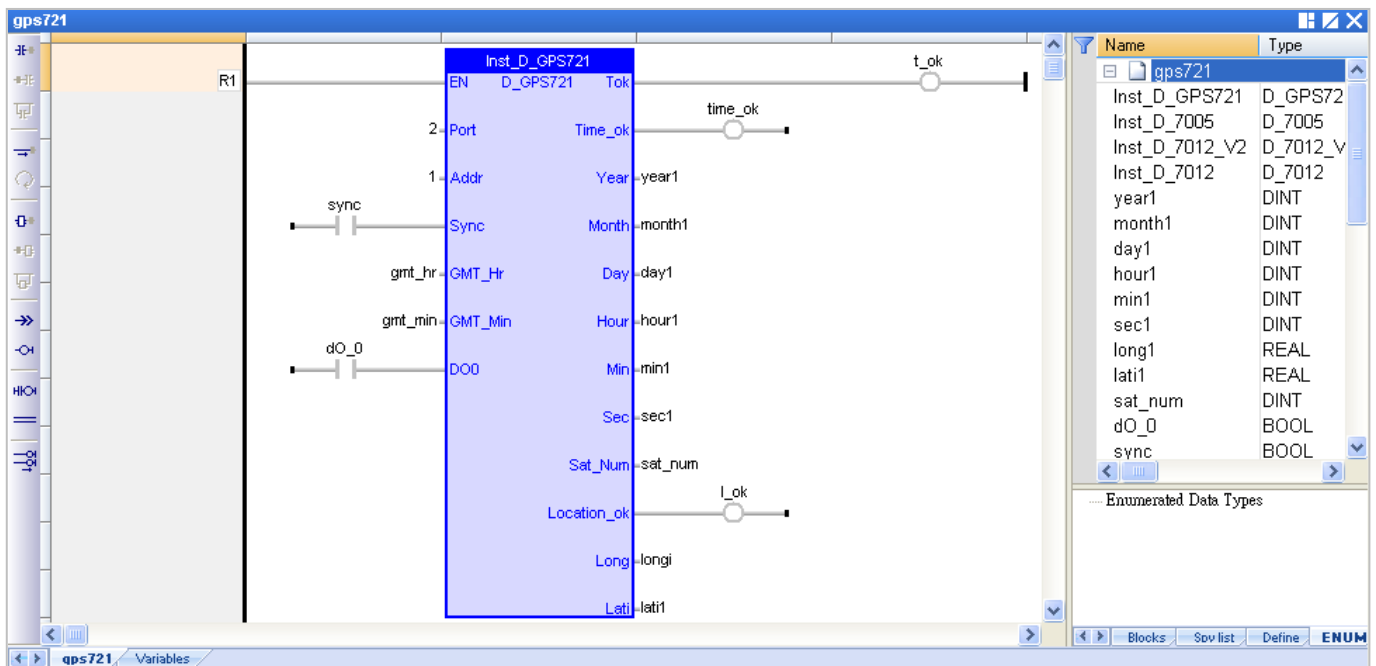
“D_GPS721” 可用來連結一個 “GPS-721” 遠端 GPS 接收模組 (內含 1 DO, 1 PPS 輸出通道) · 用來接收 GPS 衛星訊號 · 即時取得精確的時間 · 日期 · 經/緯度 · 作為定位與校時使用。另外 · “GPS-721” 具有 RS-232 介面 · 遠端主機可透過 RS-485 的 DCON 命令來詢問 GPS-721 的 GPS 資訊 · 也可遠端控制其內建的 DO 通道 · 而 PPS (Pulse Per Second) 功能 · 更可用來做為簡單的時間同步。

產品網頁:

http://www.icpdas.com/root/product/solutions/industrial_wireless_communication/wireless_solutions/gps-721_tc.html

註:

1. 一台 PAC 只能使用一個 GPS-721 模組。
2. 所有連接的模組皆需設定過一次 · 請先使用 "DCON Utility" 軟體 (見 [P8-1](#)) 設定 GPS-721 模組的相關參數 · 其預設 Address (ID) 為 "1" · Baudrate 為 "9600" · Checksum 為 "Disable"。
3. 請在 "I/O boards" 視窗加入 "DCON" (見 [8.1 節](#)) · 並填入正確的參數 (例如: Port · Baud_rate...)。
4. 可參考 [第 12 章](#) · 點選功能表 "File" > "Add Existing Project" > "From Zip" · 回存出貨光碟中的範例程式 (CD-ROM:\Napdos\Win-GRAF\demo-project\dmeo_gps721.zip) 並查看詳細的程式內容。



輸入參數:

- EN:** 資料型態: BOOL · 設定為 "TRUE": 啟用; 設定為 "FALSE": 不啟用。
- Port:** 資料型態: DINT · COM Port 編號 (可設定為 1 ~ 37 · 視 PAC 而定)。
(** 需設定為常數 · 即不可變動的值得值 **)
- Addr:** 資料型態: DINT · 模組的 Net-ID 位址 (預設為 "1" · 可設定為 1 ~ 255)。
(** 需設定為常數 · 即不可變動的值得值 **)
- Sync:** 資料型態: BOOL · 設定為 "TRUE" 來啟用時間自動同步功能 · 當 GPS-721 與 PAC 時間差距 5 秒 (或更多) 會自動校正 PAC 時間。設定為 "FALSE" 來關閉此功能。
("Time_ok" 為 "TRUE" 時 · 此功能才有效。)

GMT_Hr & GMT_Min:

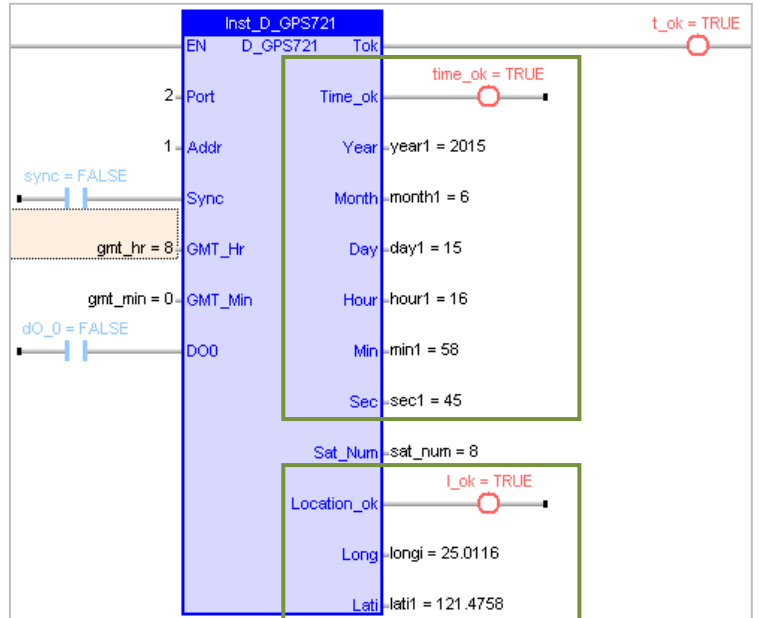
資料型態: DINT，目前所在位置與格林威治標準時間的時差。例如: 北京與台北是加 8 小時 (GMT_Hr=8，GMT_Min=0)，美國是減 6 小時 (GMT_Hr=-6，GMT_Min=0)，印度是加 5.5 小時 (GMT_Hr=5，GMT_Min=30)。

DO0: 資料型態: BOOL，GPS-721 模組的數位輸出通道 (DO0)。

輸出參數:

Tok: 資料型態: BOOL，
TRUE: GPS-721 通訊正常。
FALSE: GPS-721 通訊失敗，以下的回傳值皆無意義。

Time_ok: 資料型態: BOOL，
TRUE: 此時 Year, Month, Day, Hour, Min, Sec 的值是有意義的。
FALSE: 此時 Year, Month, Day, Hour, Min, Sec 的值無意義 (即，錯誤的或非即時的數據)。



註: 每天的 23:59:00 – 00:00:59 (2 分鐘)，"Time_ok" 會自動設為 "FALSE" 並且不會進行時間校正。

Year: 資料型態: DINT，年 (2009 ~ ...)。

Month: 資料型態: DINT，月 (1 ~ 12)。

Day: 資料型態: DINT，日 (1 ~ 31)。

Hour: 資料型態: DINT，時 (0 ~ 23)。

Min: 資料型態: DINT，分 (0 ~ 59)。

Sec: 資料型態: DINT，秒 (0 ~ 59)。

Sat_Num: 資料型態: DINT，使用衛星數量 (0: 未發現衛星 或 使用 1 ~ 9 顆衛星)。

Location_ok: 資料型態: BOOL，

FALSE: 此時 Long, Lati 的值無意義 (即，錯誤的或非即時的數據)。

TRUE: 已取得目前位置的經/緯度。

(只有在 "Location_ok" 為 "TRUE" 時，Long 與 Lati 值才正確。)

Long: 資料型態: REAL，經度 (正數: 表示東方，負數: 表示西方)。

(例如: "25.0121" 表示 25.0121 度。)

Lati: 資料型態: REAL，緯度 (正數: 表示北方，負數: 表示南方)。

(例如: "121.4576" 表示 121.4576 度。)

第 9 章 即時線上更新 (On Line Change)

"On Line Change" 功能允許 Win-GRAF PAC 在運行的同時更新小幅修改過的程式，此修改的程式名稱必須與 PAC 中目前正在 Run 的程式是同一個。"On Line Change" 功能主要提供給緊急狀況使用，像是應用場所不允許因為要更換應用程式而短暫的停機或停止運作，也找不到時間可供更換新的應用程式 (例如，需 24 小時運作不能停止的設備)。若非以上狀況，最好不要使用此功能，您可先停止運行中的應用程式，再將修改過的程式下載到 PAC 中 (見 2.3.5 節)，這是比較安全的作法。

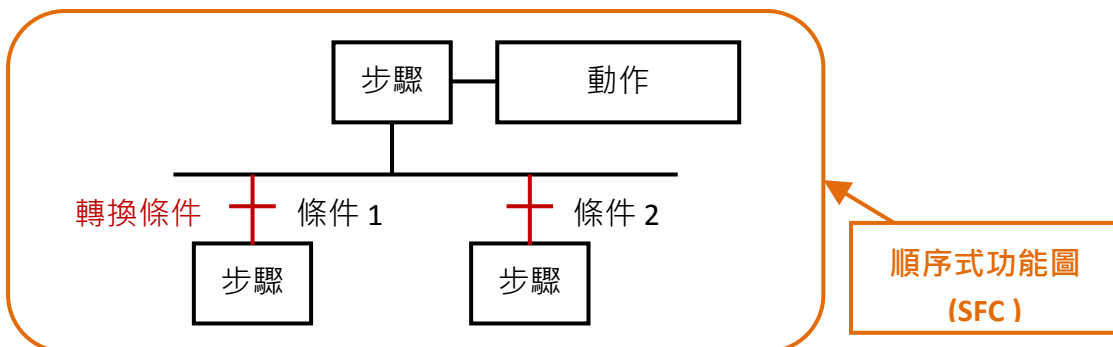


9.1 "On Line Change" 功能的使用限制

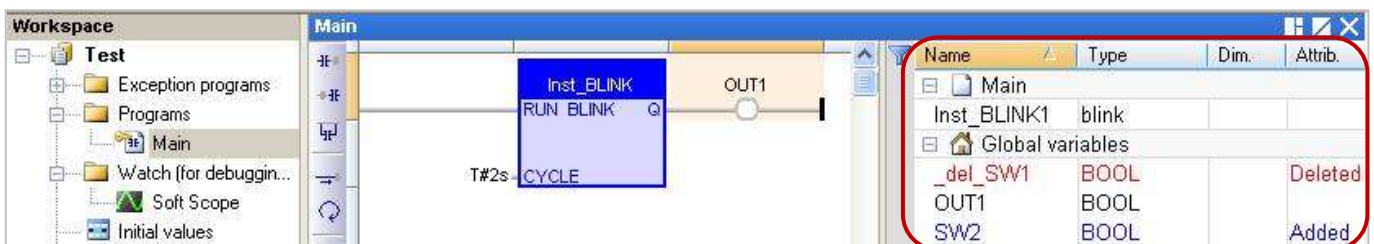
 在啟用 "On Line Change" 功能之前，請先了解下列使用限制:

啟用 "On Line change" 功能後，可執行下列變更 (不停止原有程式):

- 變更單一程式中的程式碼。
- 變更單一順序式功能圖 (SFC) 的條件或動作。

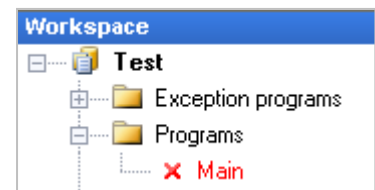
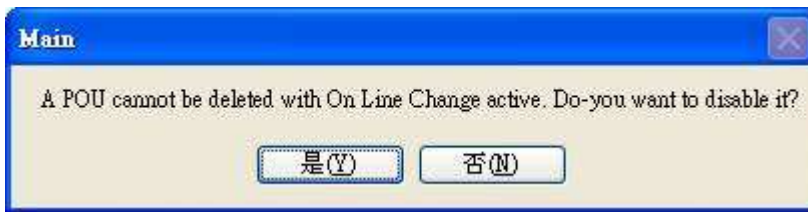


- 新增、更名、刪除全域 (Global) 或 區域 (Local) 變數。
- 新增、更名、刪除全域 (Global) 或 區域 (Local) 功能方塊的樣例變數。



啟用 "On Line change" 功能後，**不允許**以下變更:

- 新增、更名、刪除程式。(若刪除程式，會出現以下警告訊息)



- 變更 順序式功能圖 (SFC)。
- 變更 使用者自定功能方塊 (UDFB) 的區域參數或變數。
- 變更 變數 或 功能方塊 (FB) 樣例變數 的類型、陣列 Dim 長度與字串長度。
- 變更 "I/O boards" 視窗中的設定。

此外，啟用 "On Line change" 功能後，以下的程式寫法是**不安全的**:

- 脈波 (P 或 N) 接點 與 線圈 (邊緣偵測)。
- ✍ 請改成使用已定義的 "R_TRIG" 與 "F_TRIG" 功能方塊的樣例變數。

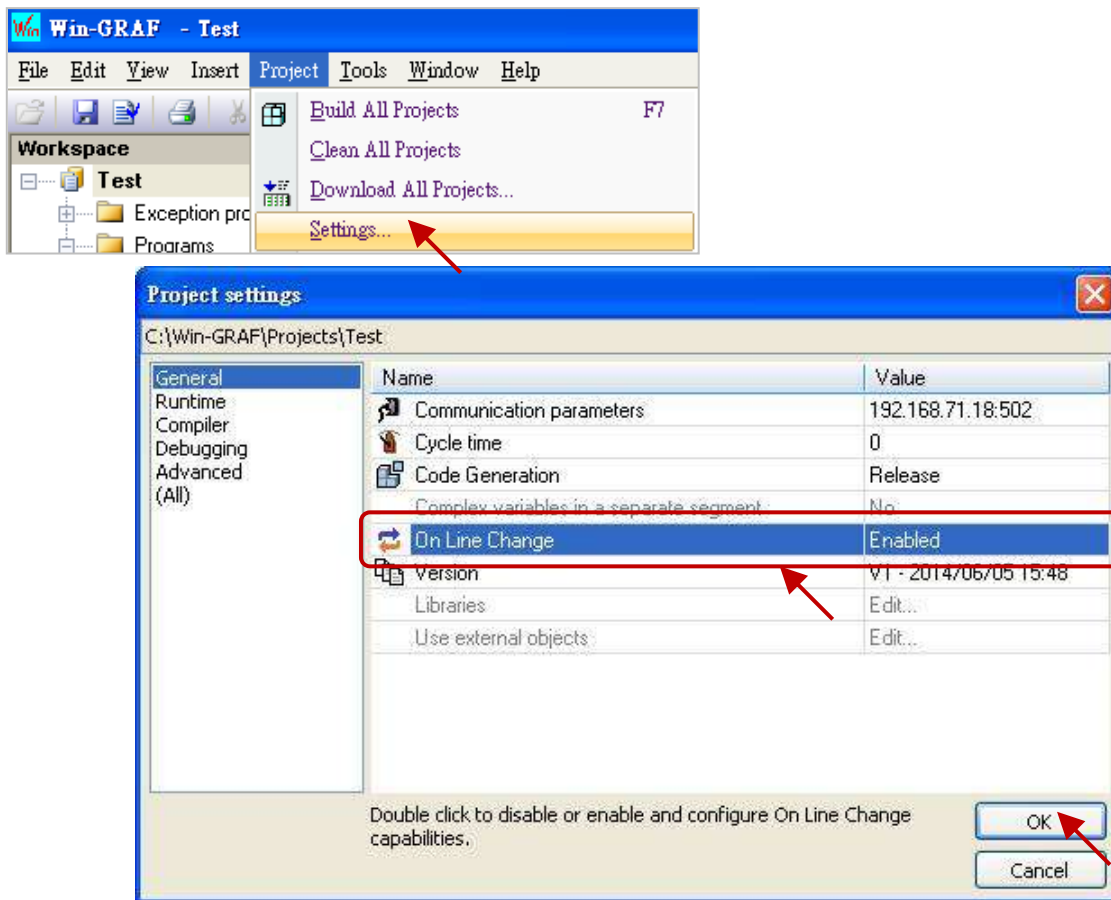
上升脈波偵測		
	修改前	修改後
P (False > True)		
下降脈波偵測		
N (True > False)		

- FBD 的迴圈中含有未定義的變數連結。
- ✍ 您必須明確地指定迴圈中的變數。

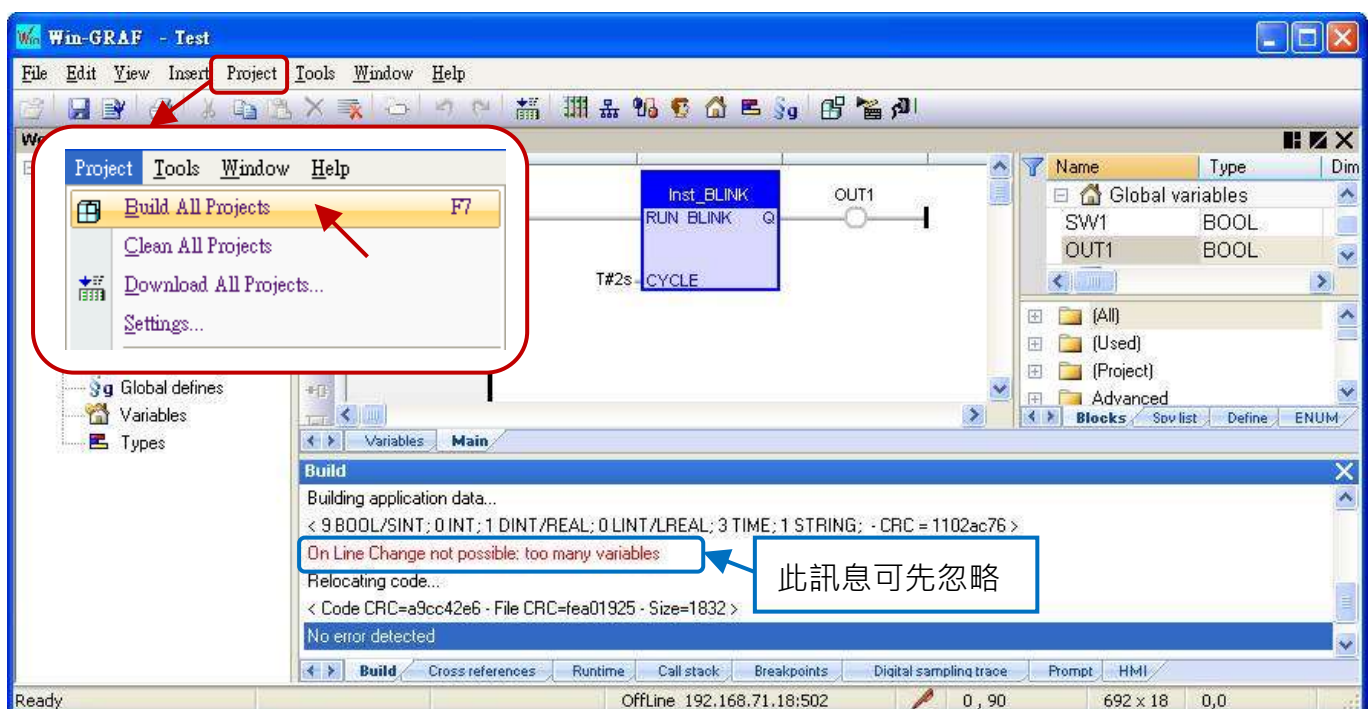
9.2 使用 "On Line change" 功能

啟用 "On Line change" 功能:

1. 滑鼠點選功能表 "Project > Settings..."，再雙擊 "On Line Change" 項目將其設定為 "Enabled"。



2. 接著，需點選功能表 "Project > Build All Projects"，執行程式編譯才能執行後續設定。



配置變數使用量:

當啟用 "On Line Change" 功能，為了允許新的變數與方塊宣告，您必須為各種資料型態的變數預留可使用的數量。

3. 同步驟 1，滑鼠點選功能表 "Project > Settings..."，再雙擊 "On Line Change" 項目進入設定視窗。請在 "Value" 或 "Margin" 欄位設定需新增的數量。

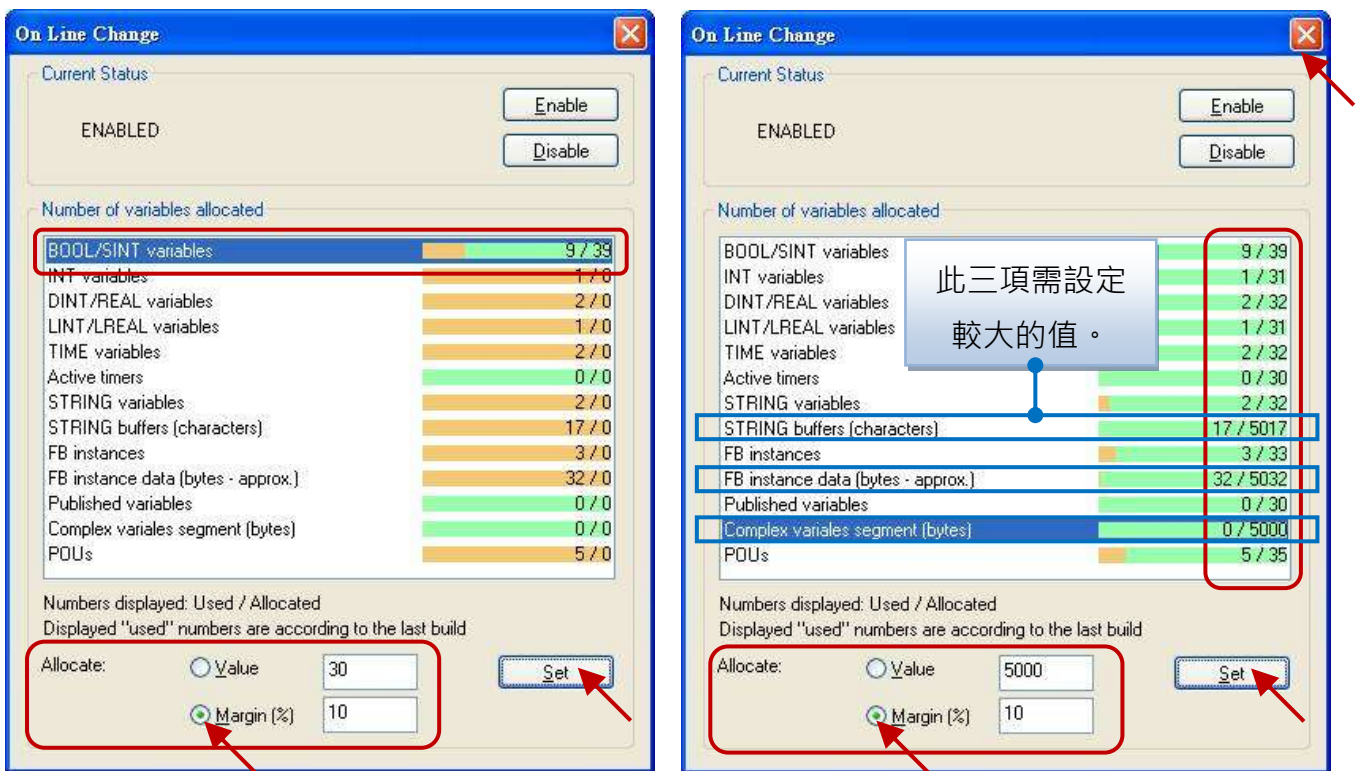
註: 若 "Value" 與 "Margin" 兩者皆有設定值，會取用較大的值。此例，"Value" 填入 "30" 且 "Margin" 填入 "10"，由於顯示數值 x 10% 小於 30，因此會啟用較大值 "30"。


4. 接著，點選所需的資料型態項目並按 "Set" 按鈕以完成設定。

(例如: 點選 "BOOL/SINT variables" 並按 "Set" 後，數量為 9 + 30 = 39)。

註: "STRING buffers (characters)"、"FB instance data (bytes – approx.)" 與 "Complex variables segment (bytes)"，此三項需設定較大的值 (此例設定為 "5000")。

5. 設定完成後畫面如下，可點選右上角的 "X" 按鈕離開設定畫面。

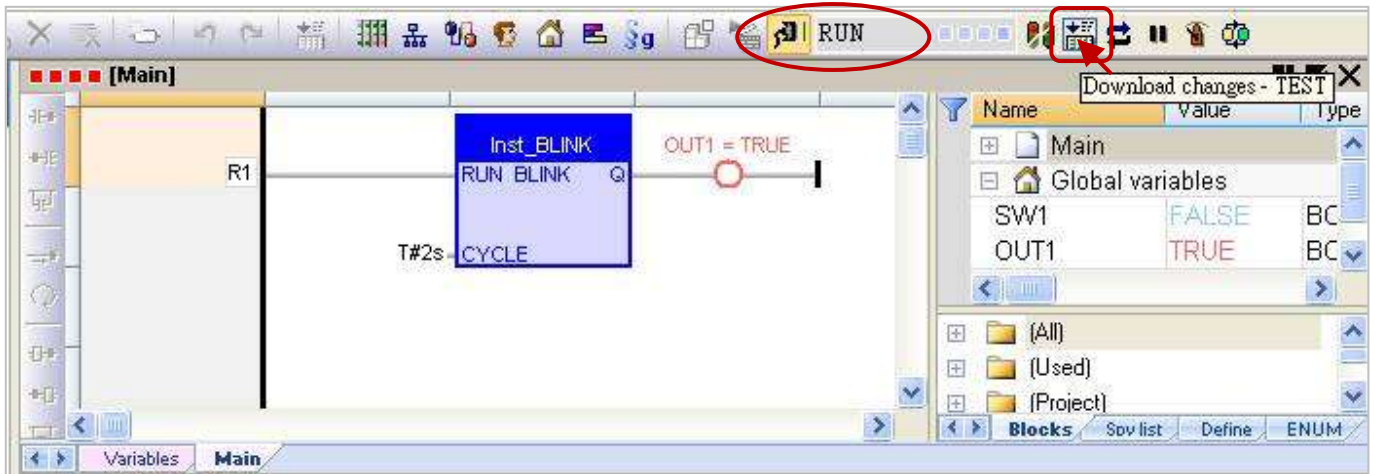


6. 點選功能表 "Project" 再選擇 "Build All Projects"，再次編譯程式。接著，再選擇 "On Line"，或點選工具按鈕  來與 PAC 建立連線。(若不熟悉連線設定，可參考 2.3.5 節)



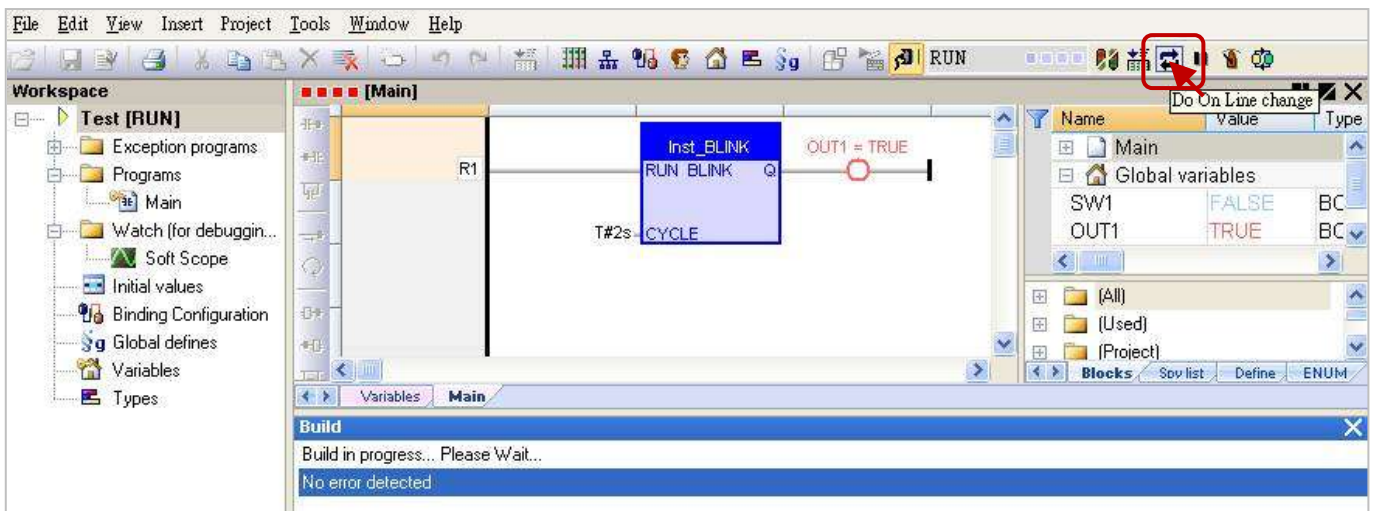
7. 連線成功後，點選工具列 "Download changes" 按鈕，將程式下載到 PAC 中。

注意： "On Line Change" 功能，僅適用在對原本的程式進行小幅度的修改 (不需停止運行)，若 PAC 中原先運行的程式與目前檔名不同，則需停止程式運行再重新下載 (參考[附錄 B](#))。



8. 點選工具列 "Do On Line change" 按鈕，來執行此功能。

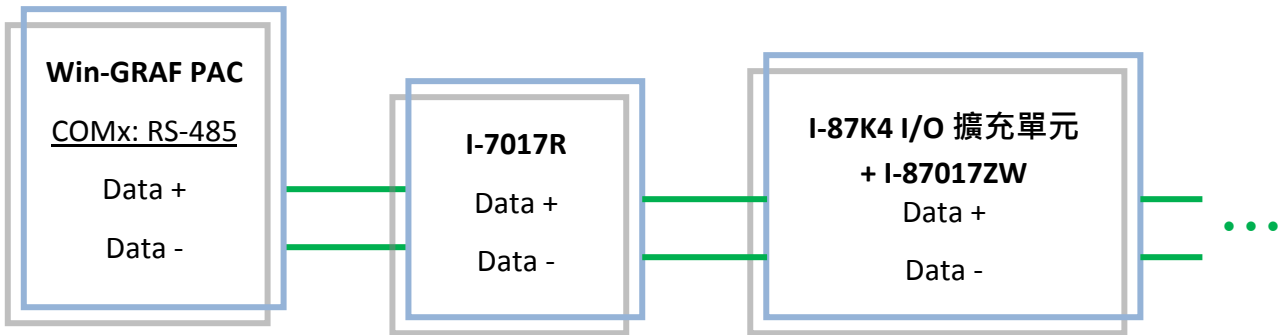
注意： 由於執行 "On Line Change" 後，為了保護系統的正常運作會有一些使用限制 (參考 [9.1 節](#))，因此請確認程式無誤後，再執行此功能。



第 10 章 資料/型態轉換 與 使用 PAC 時間

10.1 AI 資料轉換

若在 PAC 的 Slot 0 ~ 7 使用 AI 模組 (例如: I-8017HW) , 並且想把 AI 輸入訊號 (例如: "4 ~ 20 mA" 或 "0 ~ 10 V") 轉換為使用者所需的工程應用值 (例如: 0 ~ 10000) 可參考 4.3 節。然而, 若是使用遠端的 AI 模組 (例如: 透過 PAC 的 RS-485 Port 去連接的 I-87017ZW 或 I-7017R 模組) , 則可參考以下設定:

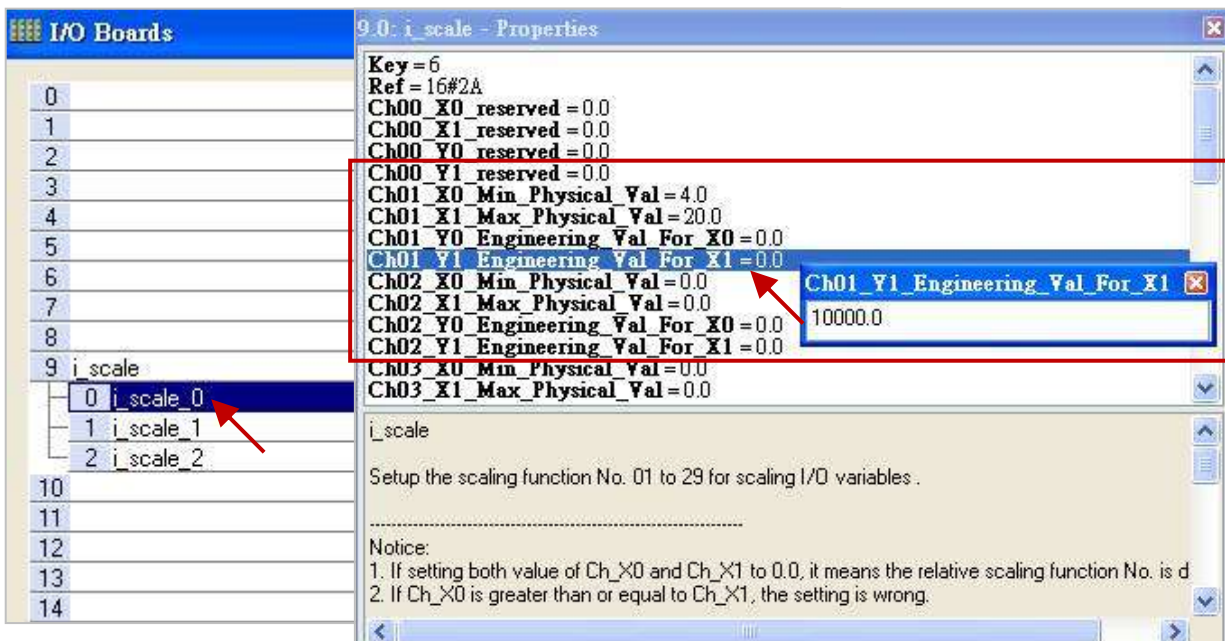


1. 首先, 在 "I/O Boards" 視窗中, 連上 "i_scale", 並滑鼠雙擊 "i_scale_x" 開啟屬性視窗。(若不熟悉操作, 請參考 4.2 節)

注意: "I/O Boards" 視窗中只能使用一個 "i_scale" (請勿連接兩個或多個)。

2. 設定需啟用的轉換函式編號 與 數值 (例如: 使用函式 1 並將 "4 ~ 20 mA" 轉換為 "0 ~ 10000")。

Ch01_X0_Min_Physical_Val: "4.0"。
Ch01_X1_Max_Physical_Val: "20.0"。
Ch01_Y0_Engineering_Val_For_X0: "0.0"。
Ch01_Y1_Engineering_Val_For_X1: "10000.0"。



3. 編寫一個 ST 程式，用來將實體值 (例如: Phy_V[0] ~ [7]) 轉換為 工程值 (例如: Eng_V[0] ~ [7])。
(若不熟悉程式建立方式，可參考 [2.3.3 節](#))

```

(* ii is declared as DINT
Phy_V is declared as REAL array with Dim 8
Eng_V is declared as REAL array with Dim 8
*)

for ii := 0 to 7 do

(* Using conversion function 1 to convert a
physical value to an engineering value *)
Eng_V[ii] := Convert_to_Eng (1, Phy_V[ii]);

end_for;

```

(* ii 宣告為 DINT 變數
Phy_V 宣告為 REAL 陣列，且 Dim. = 8
Eng_V 宣告為 REAL 陣列，且 Dim. = 8 *)

for ii := 0 to 7 do

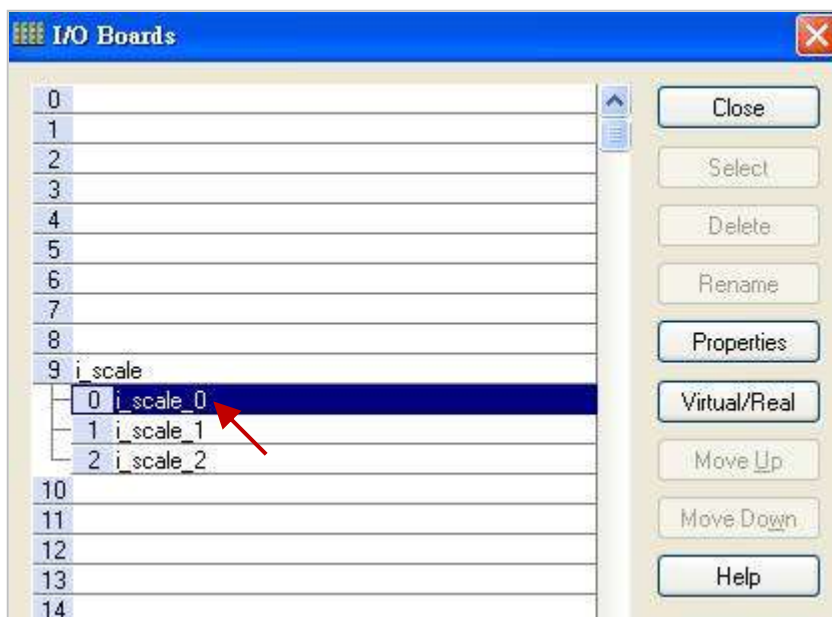
(* 套用轉換函式 1，將實體值轉換為工程值 *)
Eng_V[ii] := Convert_to_Eng (1, Phy_V[ii]);

end_for;

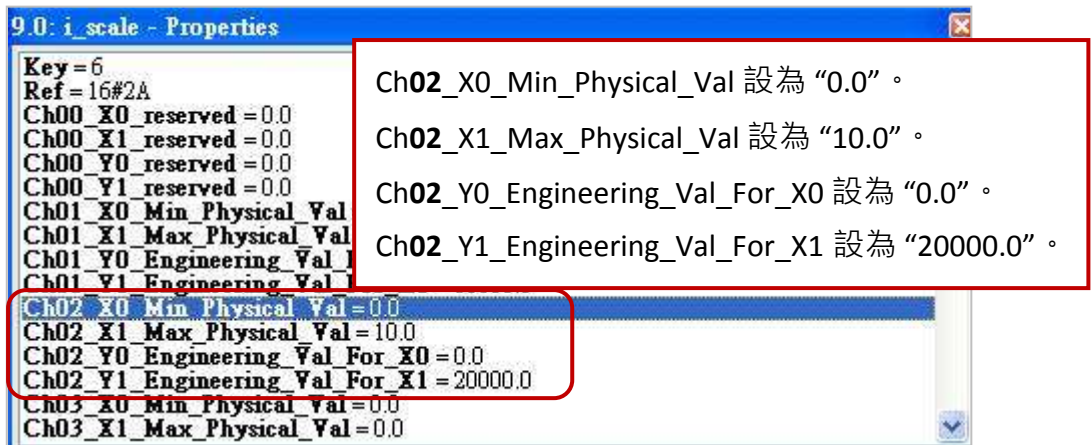
10.2 AO 資料轉換

若在 PAC 的 Slot 0 ~ 7 使用 AO 模組 (例如: I-8024W)，並且想把使用者自己的 工程應用值 轉換為 AO 輸出訊號 (例如: "0 ~ 20000" 轉換為 "0 ~ 10 V") 可參考 [4.4 節](#)。然而，若是使用 DCON 遠端 AO 模組 (例如: 透過 PAC 的 RS-485 Port 去連接的 I-7024 模組)，則可參考以下設定:

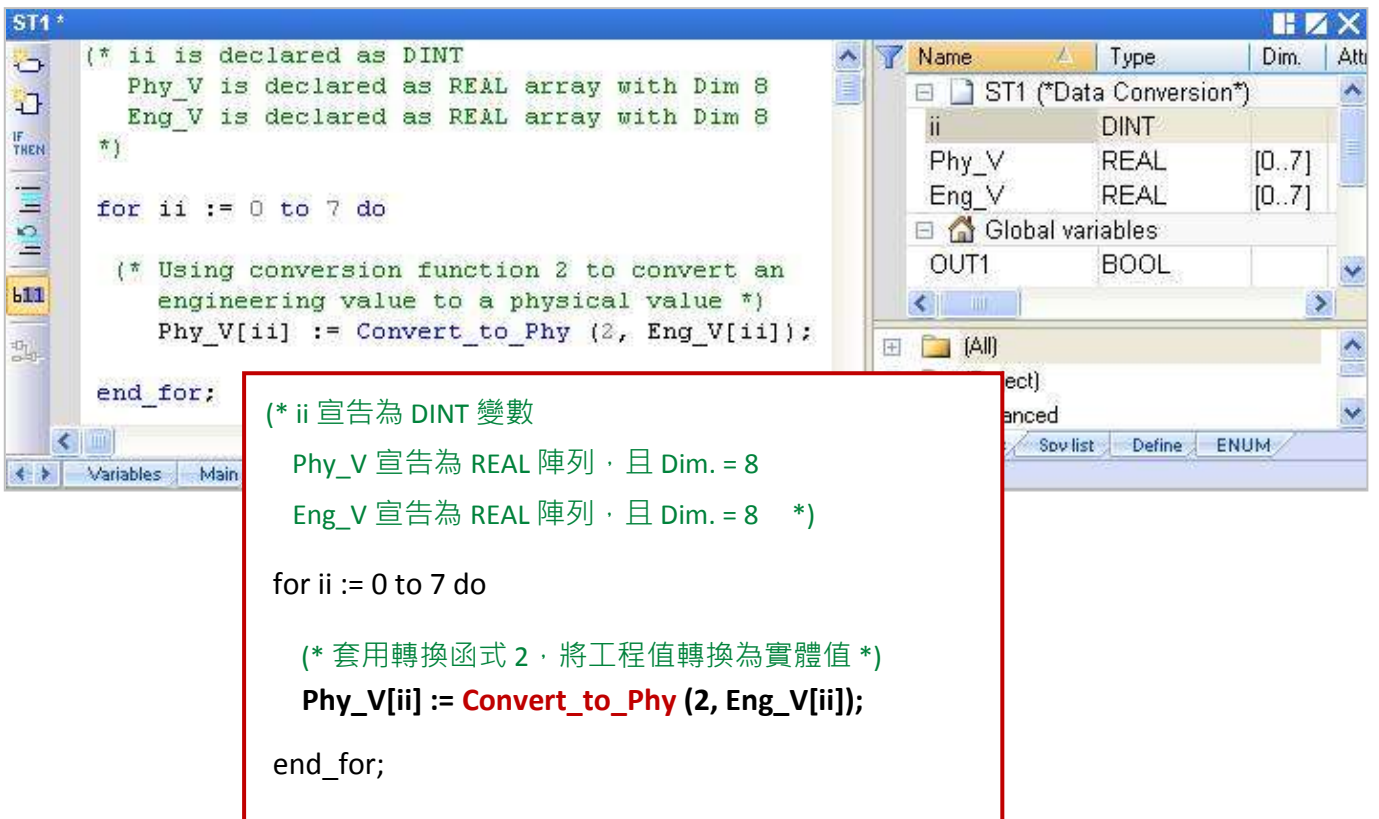
1. 在 "I/O Boards" 視窗中，連上 "i_scale"，並滑鼠雙擊 "i_scale_x" 開啟屬性視窗。(參考 [10.1 節](#))。
注意: "I/O Boards" 視窗中只能使用一個 "i_scale" (請勿連接兩個或多個)。



2. 設定需啟用的轉換函式編號與數值 (例如: 使用函式 2 並將 "0 ~ 20000" 轉換為 "0 ~ 10 V")。



3. 編寫一個 ST 程式，用來將工程值 (例如: Eng_V[0] ~ [7]) 轉換為實體值 (例如: Phy_V[0] ~ [7])。
(若不熟悉程式建立方式，可參考 [2.3.3 節](#))



10.3 資料型態轉換

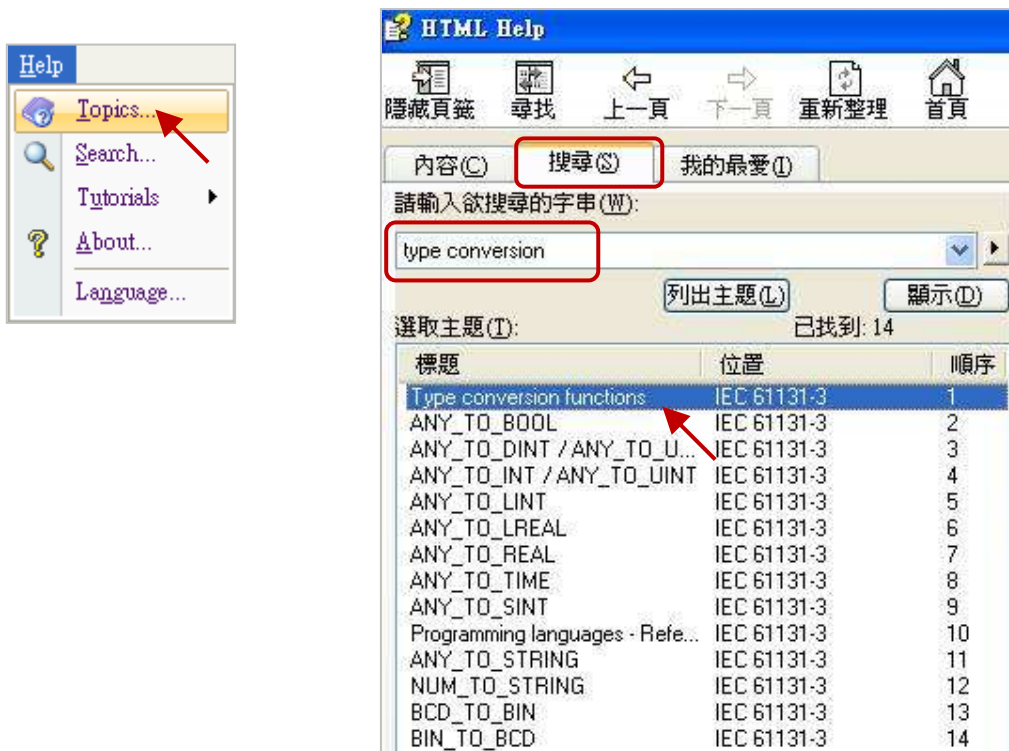
當不同的資料型態要進行 "+, -, *, /" 運算 或 ">, <, =, <=, >=, <>" (不等於) 判斷，或使用函式內參數的變數或型態不一樣時，需先使用型態轉換函式 (如下表) 來轉換為一樣的資料型態，才能正常使用。

型態轉換函式	說明	型態轉換函式	說明
ANY_TO_BOOL	轉換為 布林 (Boolean)	ANY_TO_REAL	轉換為 實數
ANY_TO_SINT	轉換為 短整數 (8 bit)	ANY_TO_LREAL	轉換為 雙精準實數
ANY_TO_INT	轉換為 整數 (16 bit)	ANY_TO_STRING	轉換為 字元字串
ANY_TO_DINT	轉換為 長整數 (32 bit – 預設)	NUM_TO_STRING	將數字轉為字串， 可指定小數點後轉換位數
ANY_TO_LINT	轉換為 Large 整數 (64 bit)	ATOH	將 16 進制字串轉為整數
ANY_TO_TIME	轉換為 時間格式	HTOA	將整數轉為 16 進制字串

例如，以下的 ST 程式，會將 DINT 變數先轉換為實數後，再來進行運算。

```
REAL_Val_1 := ANY_TO_REAL (DINT_Val_1) * 3.5 + 4.8 ;
```

您可在功能表中開啟 "HTML Help"，並輸入欲搜尋的字串，來了解詳細的設定說明。



10.4 BCD 轉換

BCD 碼是用 4-bit 來表示十進制數字 0~9。假設有個數值十進制是 "1 3 2" 換算為 BCD 碼是 "0001 0011 0010"，換算為二進制則是 10000100 (即 $2^7 + 2^2 = 128 + 4 = 132$)。

十進制	BCD				說明
	2^3	2^2	2^1	2^0	
0	0	0	0	0	0
1	0	0	0	1	$2^0 = 1$
2	0	0	1	0	$2^1 = 2$
3	0	0	1	1	$2^1 + 2^0 = 3$
4	0	1	0	0	$2^2 = 4$
5	0	1	0	1	$2^2 + 2^0 = 5$
6	0	1	1	0	$2^2 + 2^1 = 6$
7	0	1	1	1	$2^2 + 2^1 + 2^0 = 7$
8	1	0	0	0	$2^3 = 8$
9	1	0	0	1	$2^3 + 2^0 = 9$

註:

BCD 碼只能表示 0~9 的數字。此 6 種數值 (1010, 1011, 1100, 1101, 1110, 1111) 將無法使用，會回傳 "0"。

下表中的函式可用來進行 BCD (Binary Coded Decimal) 數值轉換。

型態轉換功能	說明
BIN_TO_BCD	將二進碼轉換為 BCD 值。
BCD_TO_BIN	將 BCD 值轉換為二進碼。

註: 若輸入 ≤ 0 ，皆會顯示 "0"。

IN1, IN2, Q1, Q2 宣告為 DINT。

BIN_TO_BCD:

$$19_{(10)} = 0001\ 1001_{(BCD)}$$

$$= 2^4 + 2^3 + 2^0 = 16 + 8 + 1 = 25$$



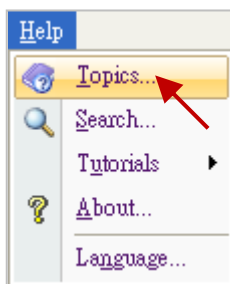
BCD_TO_BIN:

$$33_{(10)} = 0010\ 0001_{(2)} = 21_{(BCD)}$$

$$15_{(10)} = 0000\ 1111_{(2)} = 0_{(BCD)}$$



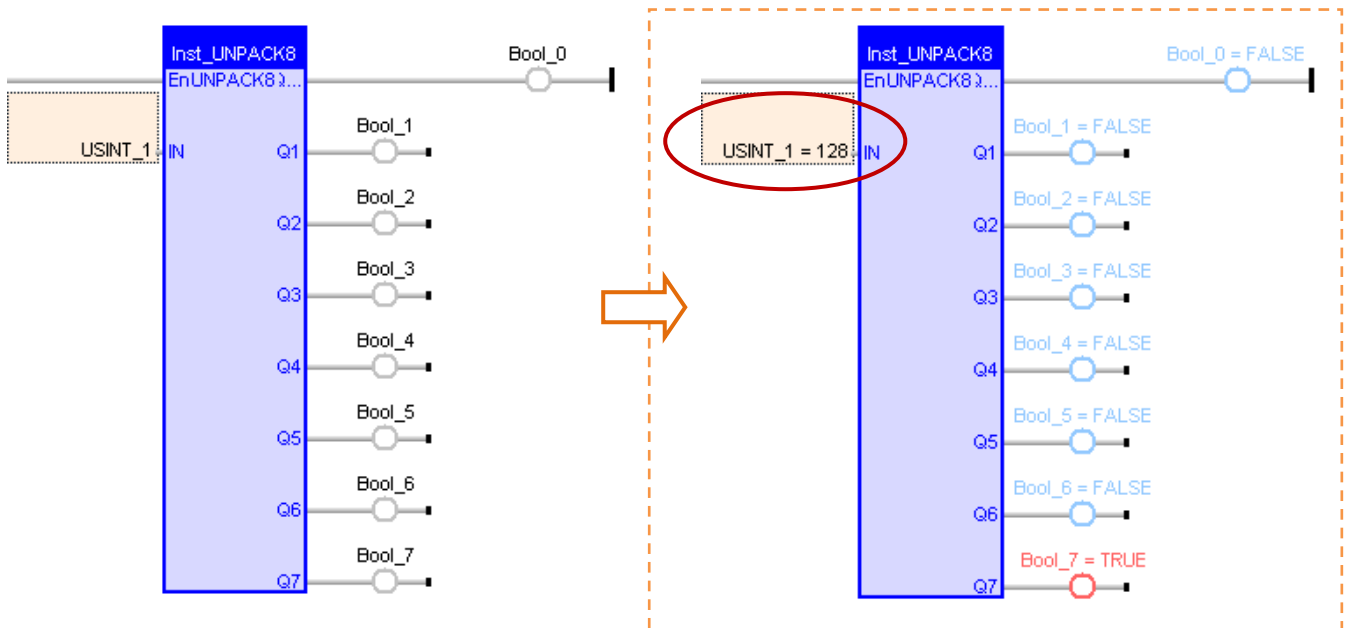
您可在功能表中開啟 "HTML Help"，並輸入欲搜尋的字串，來了解詳細的設定說明。



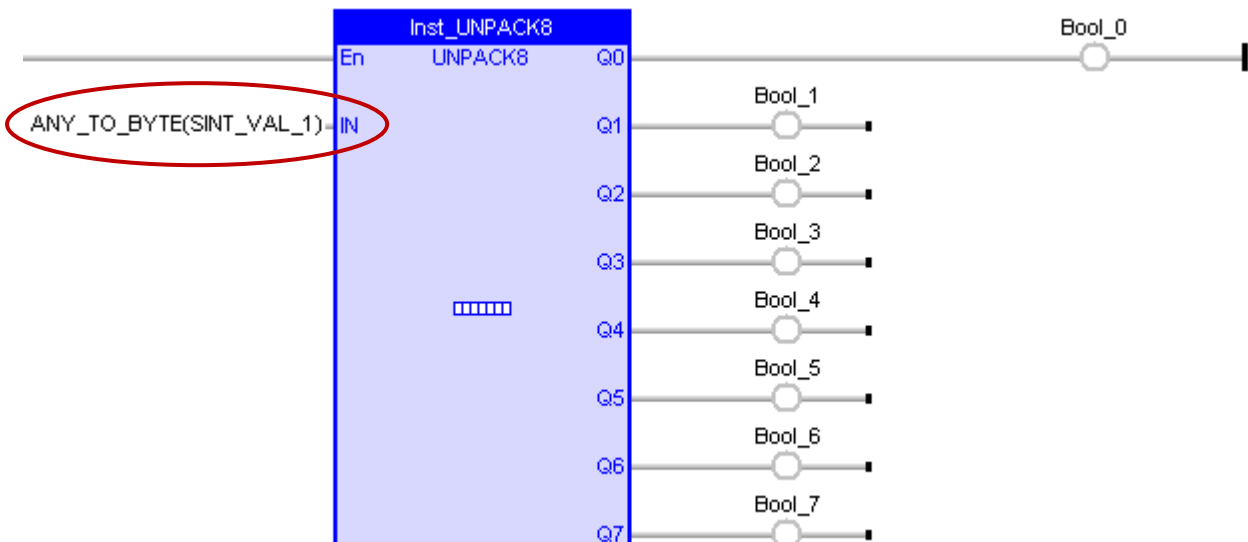
10.5 組合/拆解 整數或布林值

將整數值拆解為布林值:

若想把 1 個 BYTE (或 USINT · 範圍: 0 ~ 255) 拆解為 8 個 Boolean · 可使用 "UNPACK8" 函式。



若想把 1 個 SINT 拆解為 8 個 Boolean · 需先使用 ST 程式 "ANY_TO_BYTE()" 將 SINT 轉為 BYTE 型態 · 方法如下:



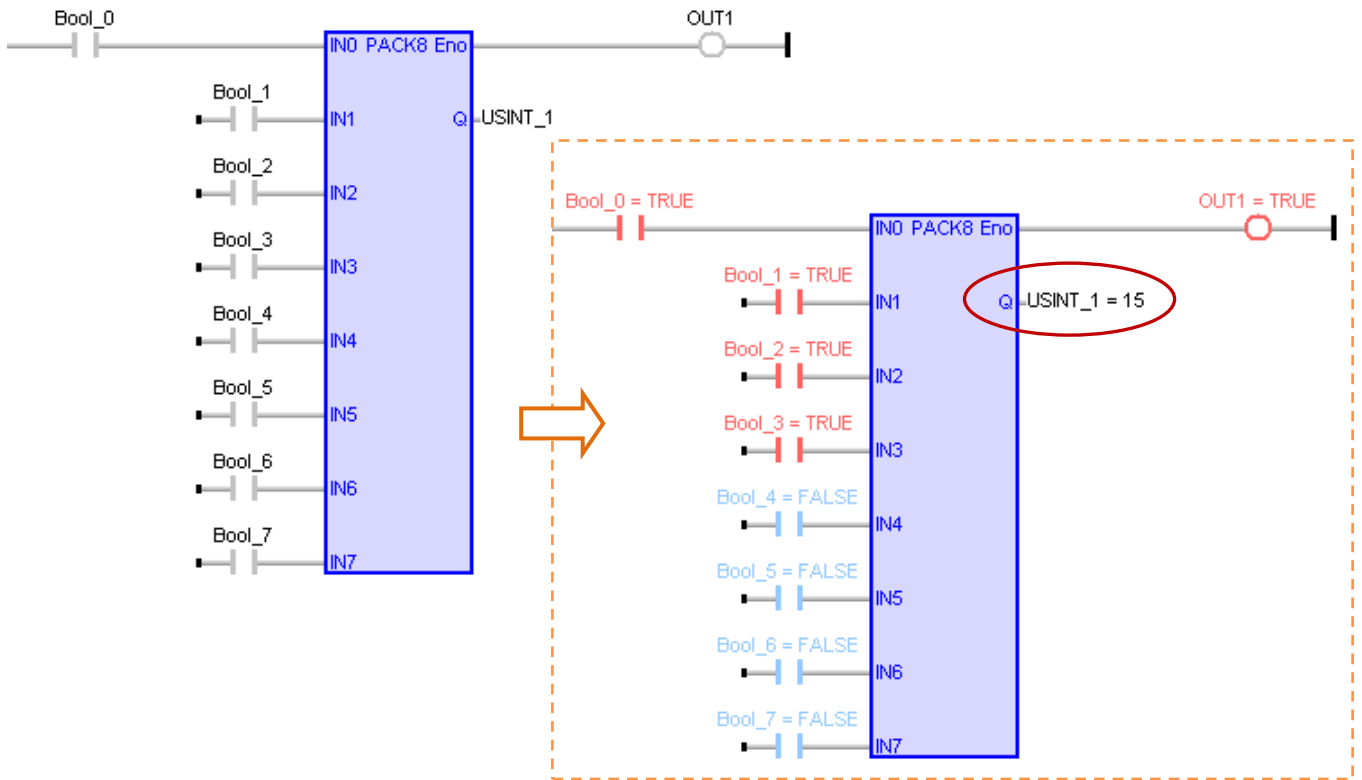
將布林值組合為整數值:

若想把 8 個 Boolean 組合成 1 個 BYTE (或 USINT · 範圍: 0 ~ 255) · 可使用 "PACK8" 函式。

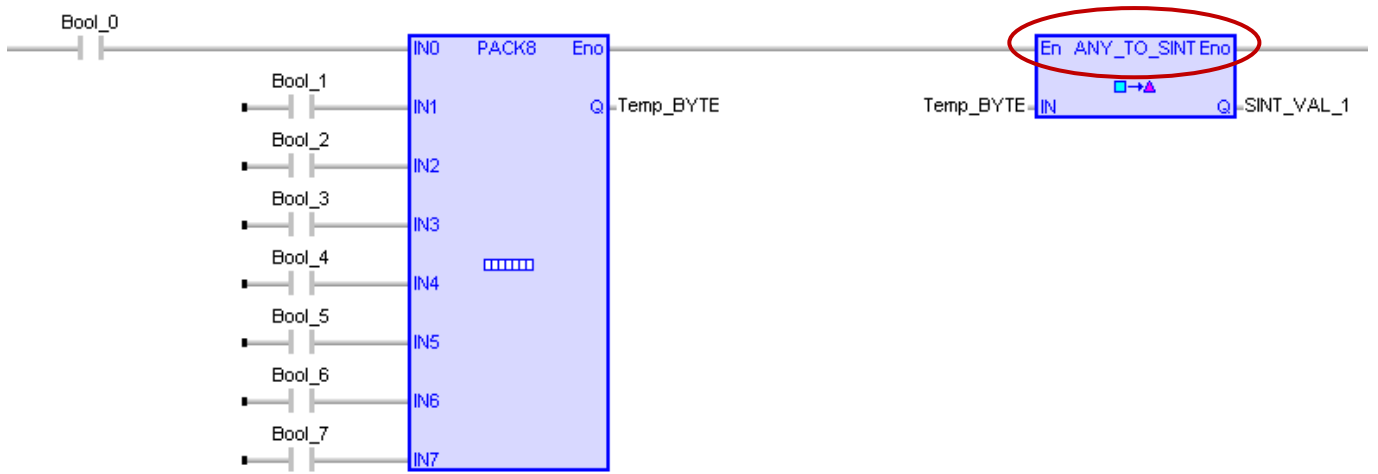
ST 語法:

```
USINT_1 := PACK8 (Bool_0, Bool_1, Bool_2, Bool_3, Bool_4, Bool_5, Bool_6, Bool_7);
```

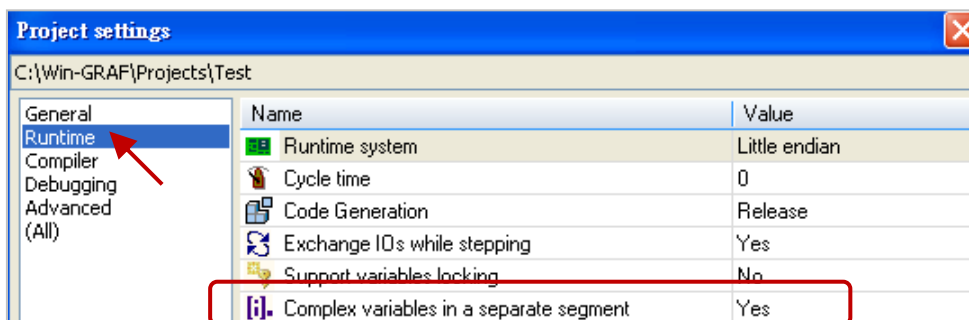
LD 語法:



若想把 8 個 Boolean 組合成 1 個 SINT，輸出 (Q) 需指定一個 "BYTE" 變數來暫存數值，並使用一個 "ANY_TO_SINT" 函式將 BYTE 轉為 SINT 型態，方法如下：



注意: 如有編譯失敗的情況，請點選功能表 "Project" > "Settings"，並檢查 "Runtime" 設定中的 "Complex variables in a separate segment" 是否為 "Yes"。



10.6 組合/拆解 BYTE, WORD, DWORD

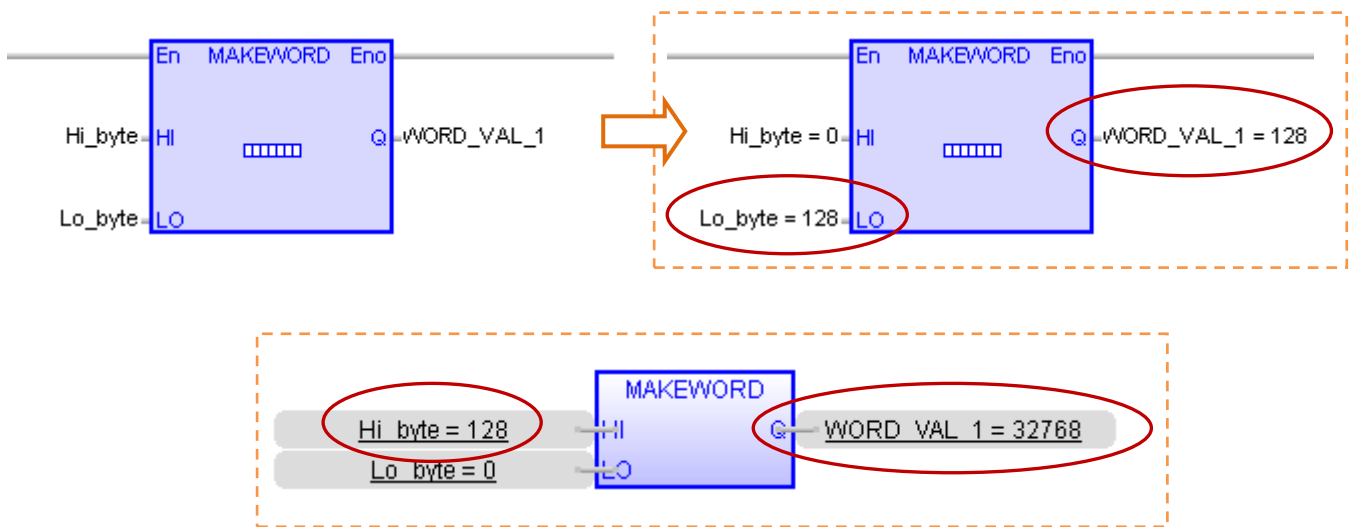
將 2 個 8-bit 組成 16-bit 資料

若想將 2 個 Byte (或 USINT) 組成 1 個 WORD (或 UINT) , 可使用 "MAKEWORD" 函式。

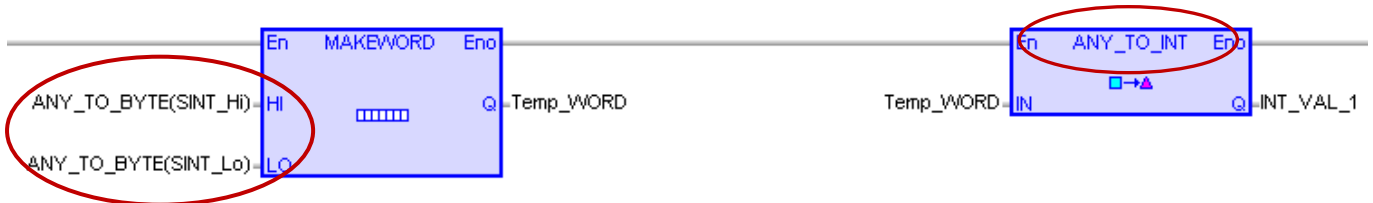
ST 語法:

```
WORD_VAL_1 := MAKEWORD (Hi_byte, Lo_byte);
```

LD/FBD 語法:



若想將 2 個 SINT 組成 1 個 INT , 需先使用 ST 程式 "ANY_TO_BYTE()" 將 SINT 轉為 BYTE , 再使用一個 "ANY_TO_INT" 函式 , 將組成的 WORD 轉為 INT 型態。



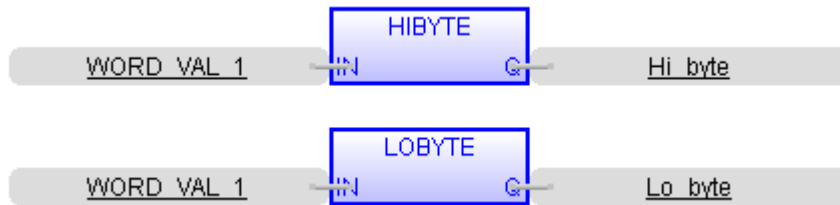
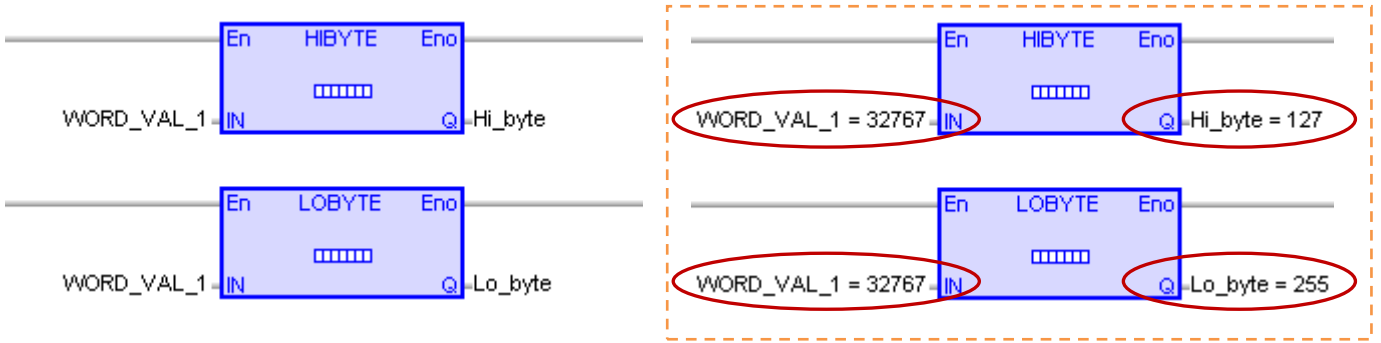
將 1 個 16-bit 拆解為 2 個 8-bit 資料

若想將 1 個 WORD (或 UINT) 拆解為 2 個 Byte (或 USINT) , 可使用 "HIBYTE" 、 "LOBYTE" 函式。

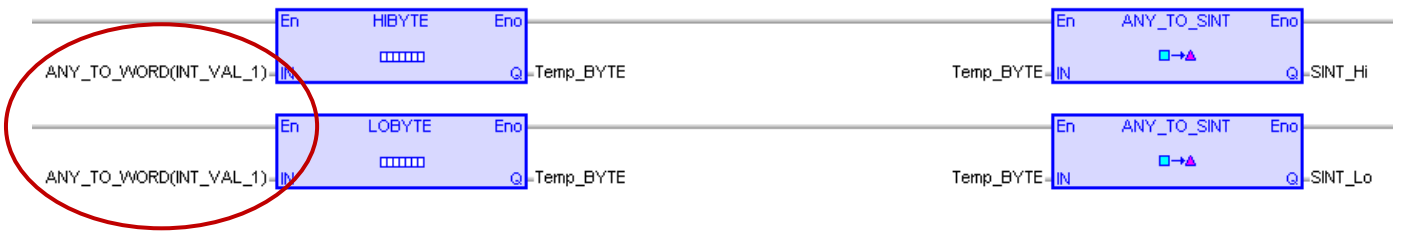
ST 語法:

```
Hi_byte := HIBYTE (WORD_VAL_1);  
Lo_byte := LOBYTE (WORD_VAL_1);
```

LD/FBD 語法:



若想將 1 個 INT 拆解為 2 個 SINT，需先使用 ST 程式 "ANY_TO_WORD()" 將 INT 轉為 WORD，再使用一個 "ANY_TO_SINT" 函式將拆解的 BYTE 轉為 SINT 型態。



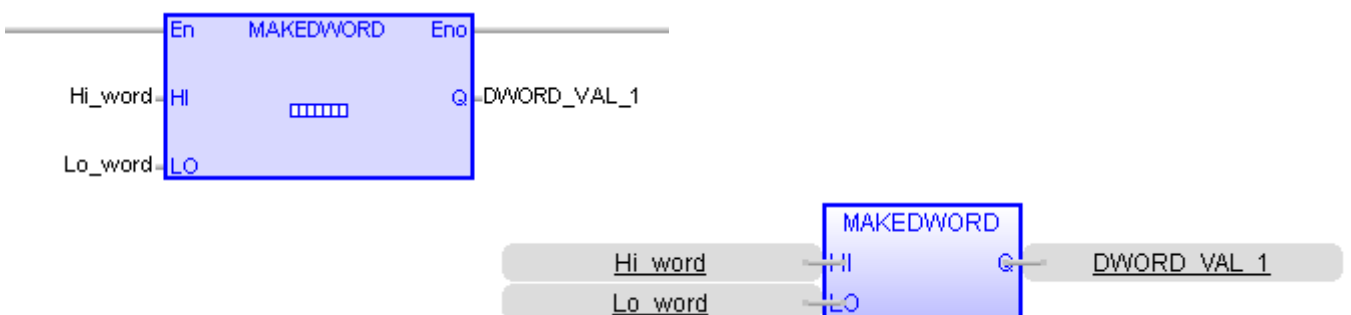
將 2 個 16-bit 組成 32-bit 資料

若想將 2 個 WORD (或 UINT) 組成 1 個 DWORD (或 UDINT)，可使用 "MAKEDWORD" 函式。

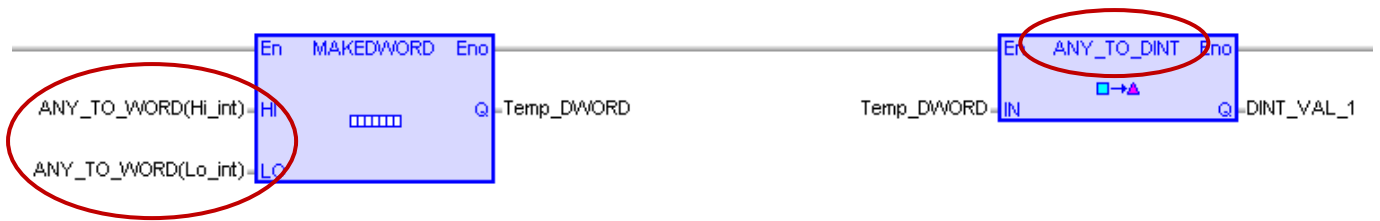
ST 語法:

```
DWORD_VAL_1 := MAKEDWORD (Hi_word, Lo_word);
```

LD/FBD 語法:



若想將 2 個 INT 組成 1 個 DINT，需先使用 ST 程式 "ANY_TO_WORD()" 將 INT 轉為 WORD，再使用一個 "ANY_TO_DINT" 函式，將組成的 DWORD 轉為 DINT 型態。



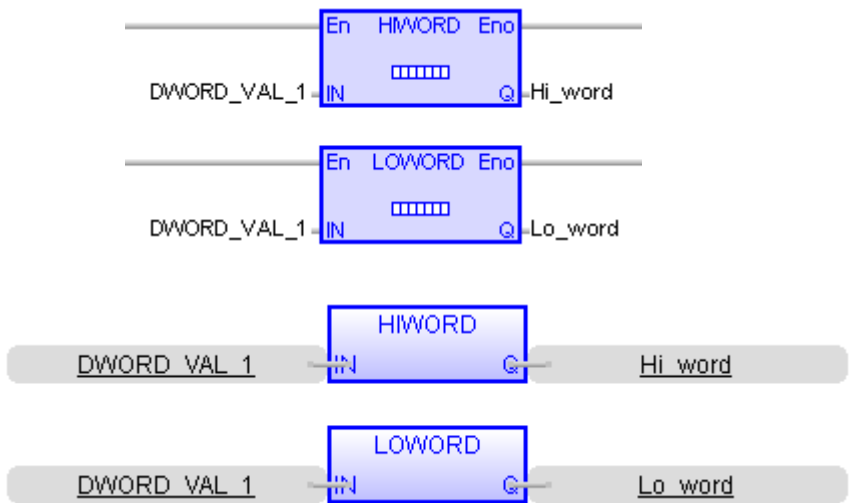
將 1 個 32-bit 拆解為 2 個 16-bit 資料

若想將 1 個 DWORD (或 UDINT) 拆解為 2 個 WORD (或 UINT)，可使用 "HIWORD"、"LOWORD" 函式。

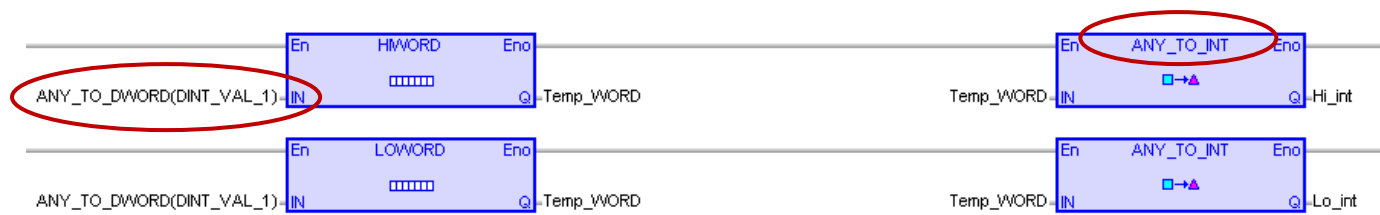
ST 語法:

```
Hi_word := HIWORD (DWORD_VAL_1);
Lo_word := LOWORD (DWORD_VAL_1);
```

LD/FBD 語法:



若想將 1 個 DINT 拆解為 2 個 INT，需先使用 ST 程式 "ANY_TO_DWORD()" 將 DINT 轉為 DWORD，再使用一個 "ANY_TO_INT" 函式，將拆解的 WORD 轉為 INT 型態。



10.7 將變數拆成 Byte Array 或將 Byte Array 組成變數

"SerializeOut" 函式可以將一個 Win-GRAF 變數值拆解成 Byte Array (或 USINT Array) ;
 "SerializeIn" 函式則可以將一個 Byte Array (或 USINT Array) 組成一個 Win-GRAF 變數值。

- 註:** 1. Array 的 Dim. 至少要設定為 "8" 。
 2. 此 "Serialize" 函式不可使用 STRING 變數。

您可在功能表中開啟 "HTML Help" ，並輸入欲搜尋的字串，來了解詳細的說明。



SerializeOut() 與 SerializeIn() 若回傳 0 ，表示儲存位置錯誤或 Array 空間不足。

(* 宣告 TMP_DINT 為一個 DINT,
 buf 為一個 BYTE Array, Dim 為 10,
 DINT_Val 為一個 DINT,
 Word_Val 為一個 WORD,
 REAL_Val 為一個 REAL *)

註:

資料型態	Byte
BOOL, SINT, USINT, BYTE	1
INT, UINT, WORD	2
DINT, UDINT, DWORD, REAL	4
LINT, LREAL	8

範例 1

(* 將 DINT_Val 拆解成 4 個 byte ，從 BYTE Array 位置 2 開始分別存放在 buf[2], buf[3], buf[4], buf[5] ，
 採用 "Little Endian" 排序 *)

```
TMP_DINT := SerializeOut (buf, DINT_Val, 2, FALSE);
```

註: 最後一個參數為 "FALSE" 表示採用 "Little Endian" 排序 (即，將低序位元組存儲在起始地址)

範例 2

(* 將 Word_Val 拆解成 2 個 byte · 從 BYTE Array 位置 0 開始分別存放在 buf[0], buf[1] · 採用 "Big Endian" 排序 *)

```
TMP_DINT := SerializeOut (buf, Word_Val, 0, TRUE);
```

註: 最後一個參數為 "TRUE" 表示採用 "Big Endian" 排序 (即 · 將高序位元組存儲在起始地址)

範例 3

(* 將 BYTE Array 內的 buf[0], buf[1], buf[2], buf[3] 組合成 1 個 REAL_Val · 採用 "Little Endian" 排序 *)

```
TMP_DINT := SerializeIn (buf, REAL_Val, 0, FALSE);
```

註: 最後一個參數為 "FALSE" 表示採用 "Little Endian" 排序 (即 · 將低序位元組存儲在起始地址)

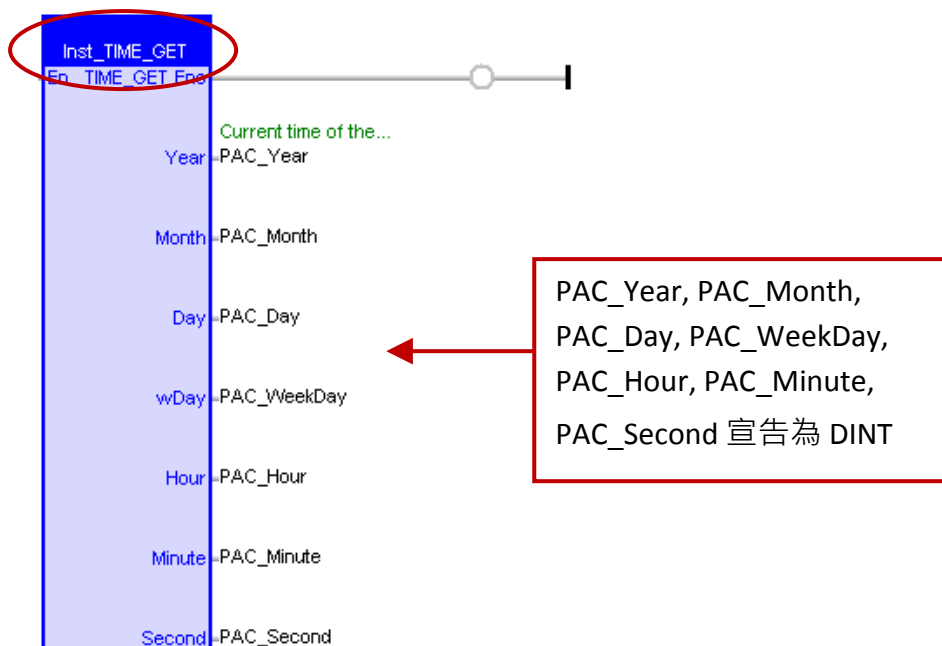
範例 4

(* 將一個 DINT_Val 對應成 1 個 REAL_Val · 採用 "Little Endian" 排序 *)

```
TMP_DINT := SerializeOut (buf, DINT_Val, 0, FALSE);  
TMP_DINT := SerializeIn (buf, REAL_Val, 0, FALSE);
```

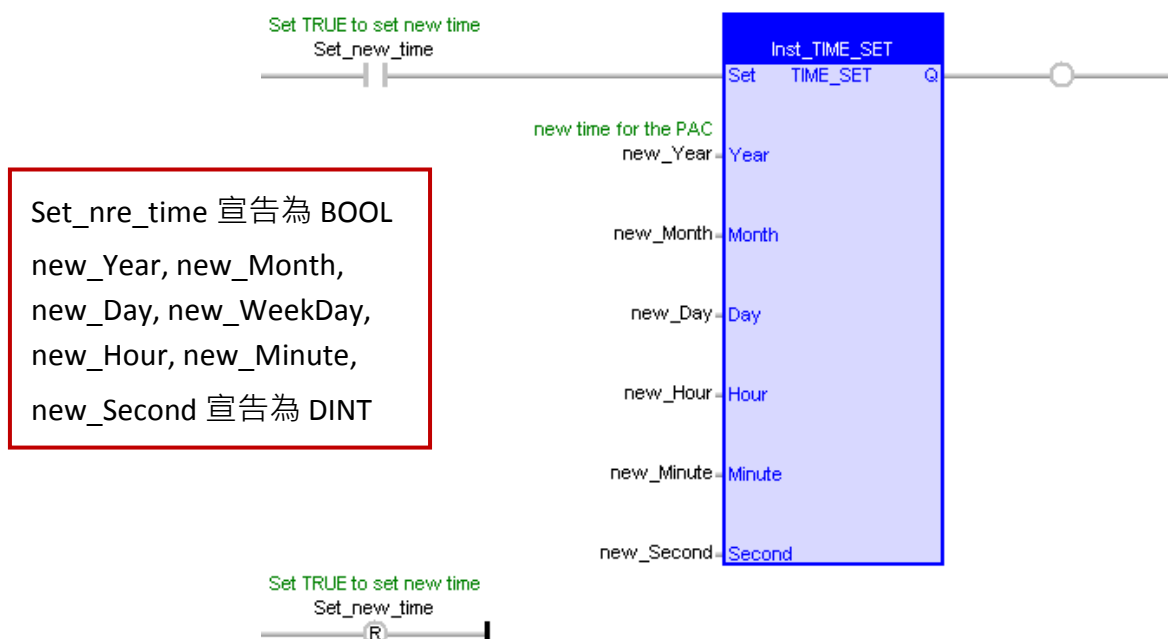
10.8 取得/設定 PAC 時間

若要取得 Win-GRAF PAC 目前的時間，可使用 "TIME_GET" 功能方塊。(可參考 [2.2.1 節](#))



若要調整 Win-GRAF PAC 目前的時間，可使用 "TIME_SET" 功能方塊。(可參考 [2.3.6 節](#))

先填入新的時間到 new_Year, new_Month, new_Day, new_WeekDay, new_Hour, new_Minute, new_Second 變數後，再將 Set_new_time 變數設定為 "TRUE" 一次。



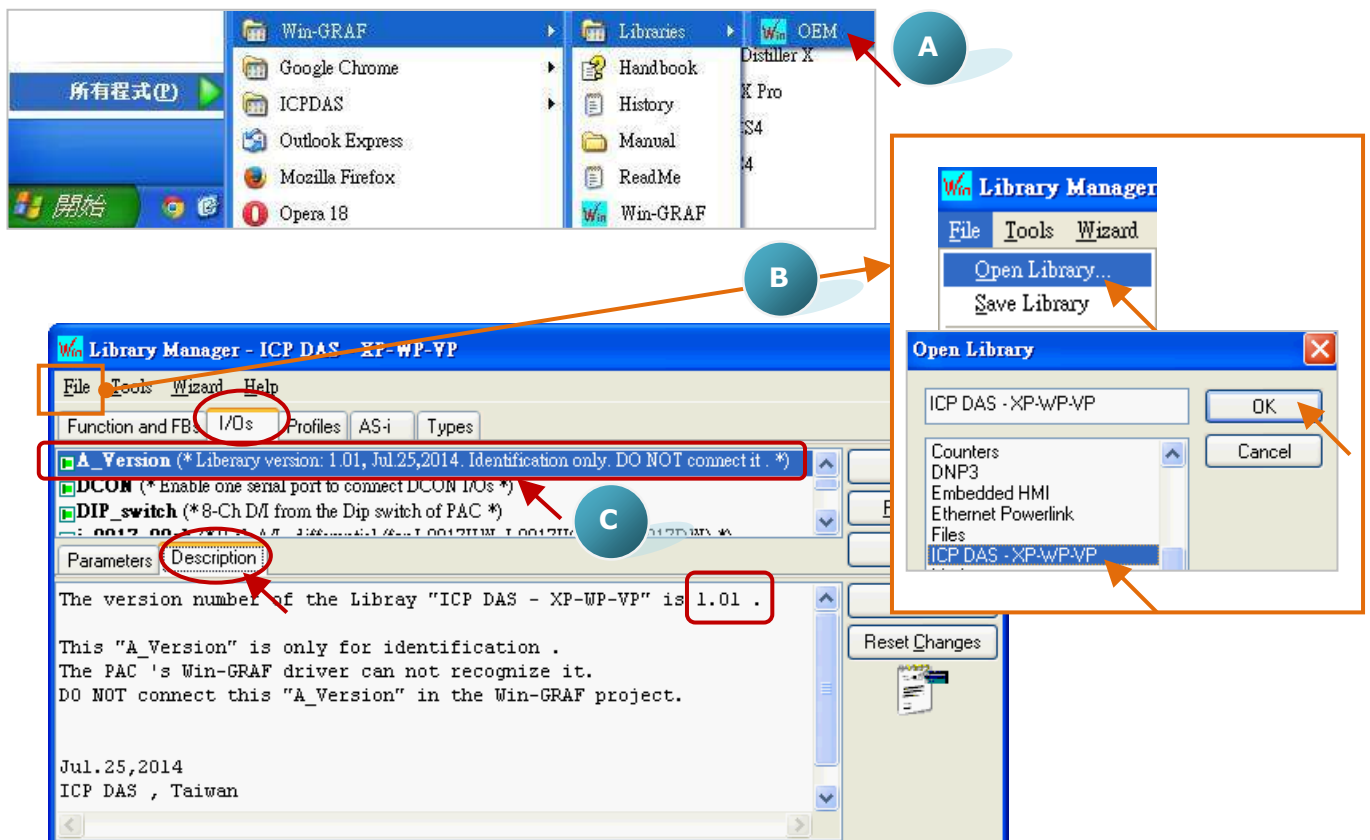
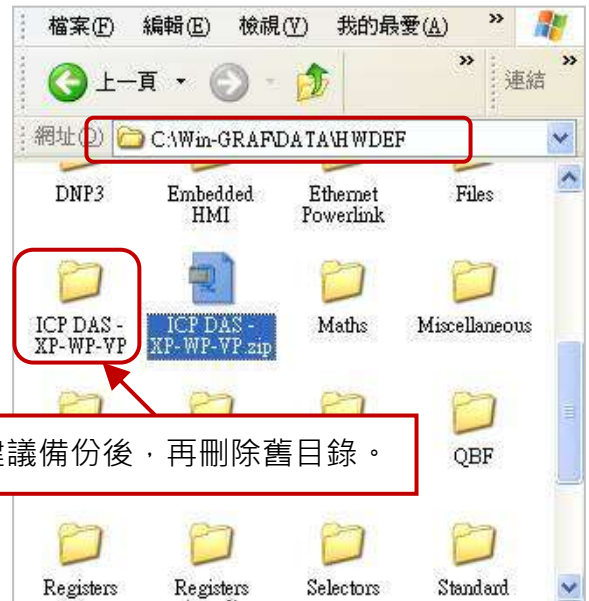
第 11 章 一般常用工具 與 有用的技巧

11.1 更新 Win-GRAF 程式庫 (Library)

您可在 http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-driver.html 下載最新的程式庫 (Win-GRAF Lib，例如: "win-graf-lib-1.xx.zip")。Win-GRAF 的程式庫 (包括 Function、Function Block 與 I/O Board)，一般儲存在 C:\Win-GRAF\DATA\HWDEF 路徑下的 "ICP DAS - XP-WP-VP" 目錄中。基於某一些情況，會將程式庫更新至新版本以支援更多的 Function 或新的 I/O Board，請依照以下步驟來執行此程序：

1. 首先，關閉所有 Win-GRAF Workbench 的視窗。
2. 可先壓縮 "ICP DAS - XP-WP-VP" 目錄並備份在其它路徑 (例如: D:\temp\xxx.zip)，再刪除舊目錄。
3. 將新的 "ICP DAS - XP-WP-VP" 目錄複製到 C:\Win-GRAF\DATA\HWDEF 路徑內，再重新開啟 Win-GRAF Workbench。

註：若您想查詢目前 Win-GRAF 程式庫的版本號，請如下圖開啟 "Library Manager" 並點選 "I/Os" 頁籤下的 "A_Version"，再點選 "Description" 查看版本號 (例如: "1.01")。



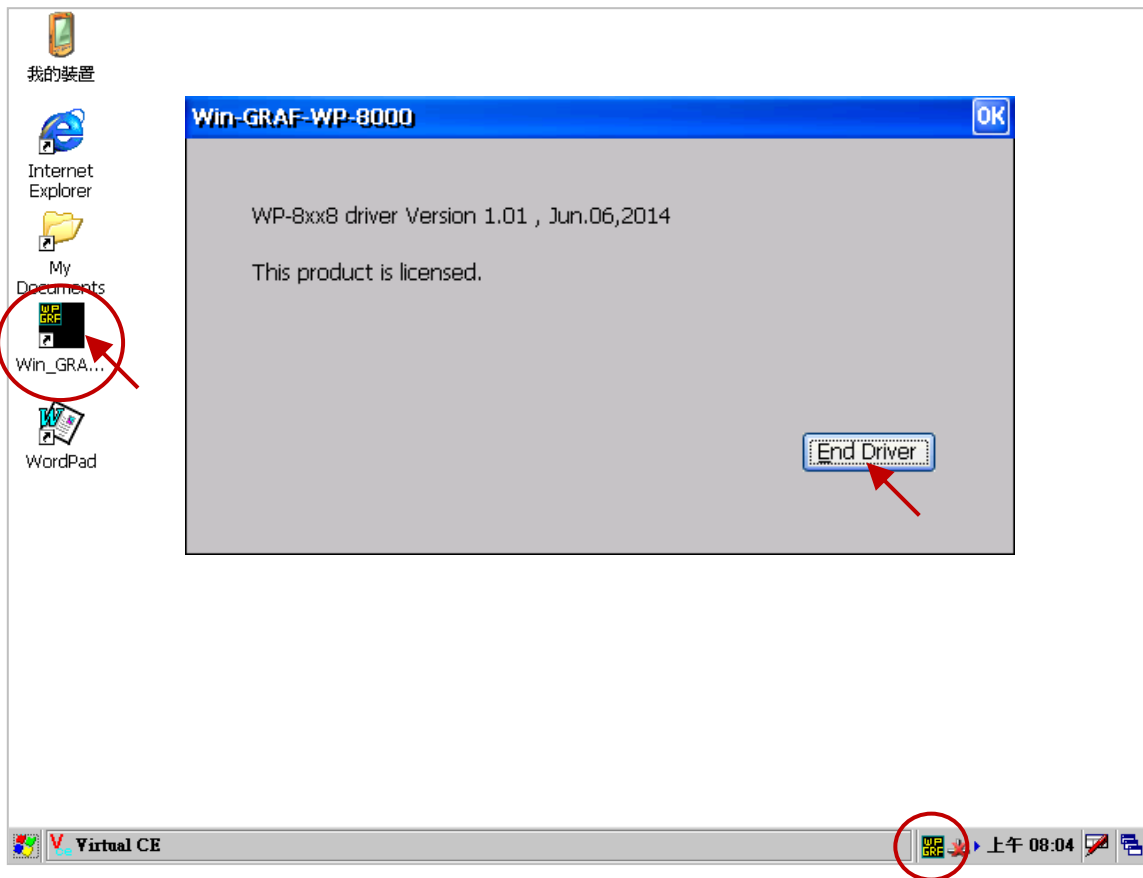
11.2 更新 Win-GRAF 驅動程式 (Driver)

為了因應新增加的功能、IO Board 或其他用途，將會不定時的發佈 PAC 新版的 Win-GRAF Driver，您可在 Web 上取得最新的 PAC Driver，並依照以下步驟更新到 PAC 中。

http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-driver.html

註：XPAC (XP-8xx8-CE6)、WinPAC (WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7) 與 ViewPAC (VP-x2x8-CE7) 會把 Win-GRAF Driver 放在 PAC 的 `\System_Disk\Win-GRAF\` 路徑內。

1. PAC 中 (以 WP-8xx8 為例)，滑鼠雙擊桌面上的 "Win-GRAF_WP_8000" 再點選 "End Driver" 停止正在運行的 Driver。



2. PC 中，以 FTP 方式先將新的 Driver 丟到 PAC 的 `\Temp\` 路徑內。
3. PAC 中，將 `\Temp\` 路徑內的新 Driver 覆寫到 `\System_Disk\Win-GRAF\` 路徑內，再將 PAC 重新開機即可。

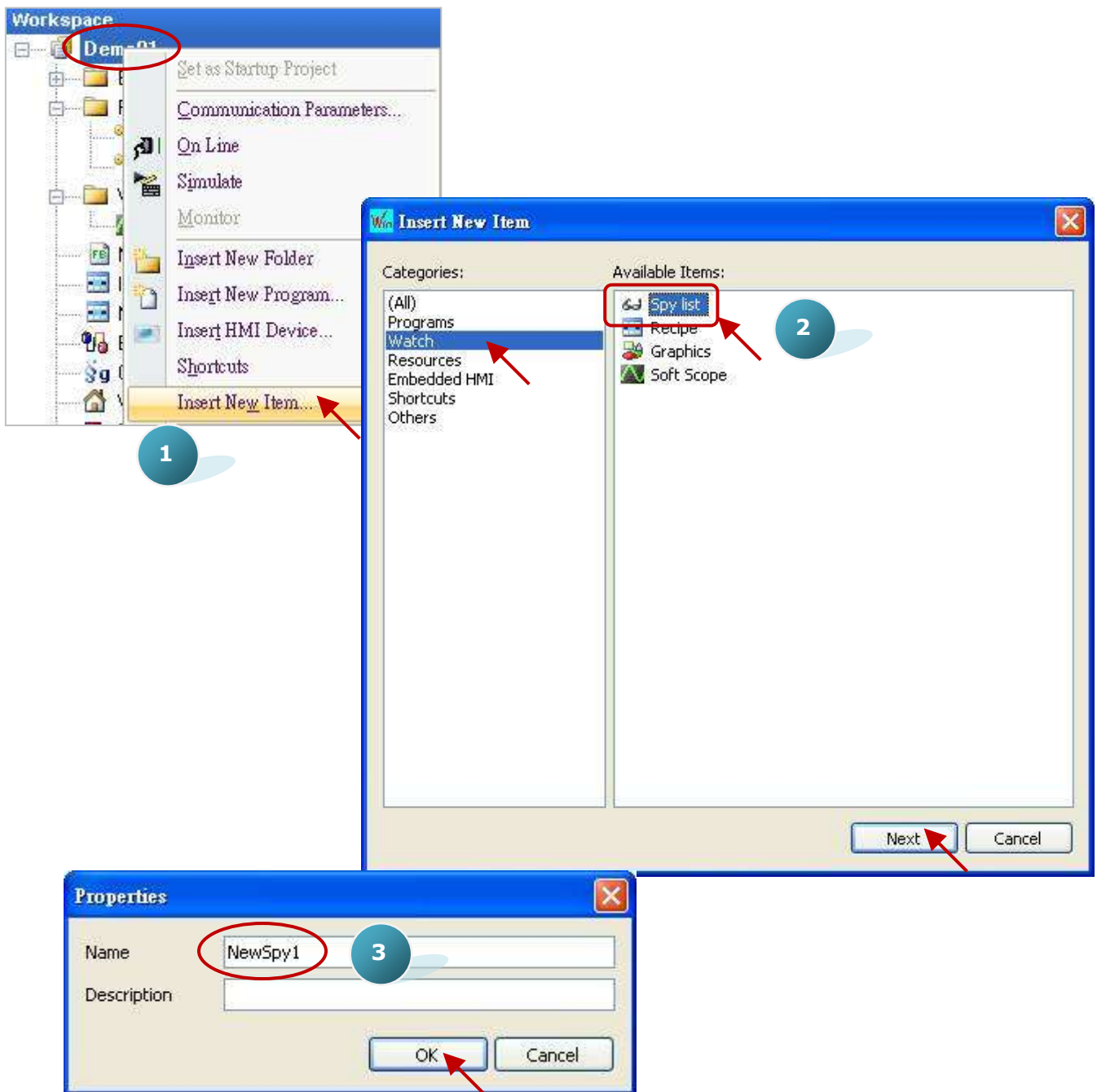


11.3 觀測清單 (Spy List)

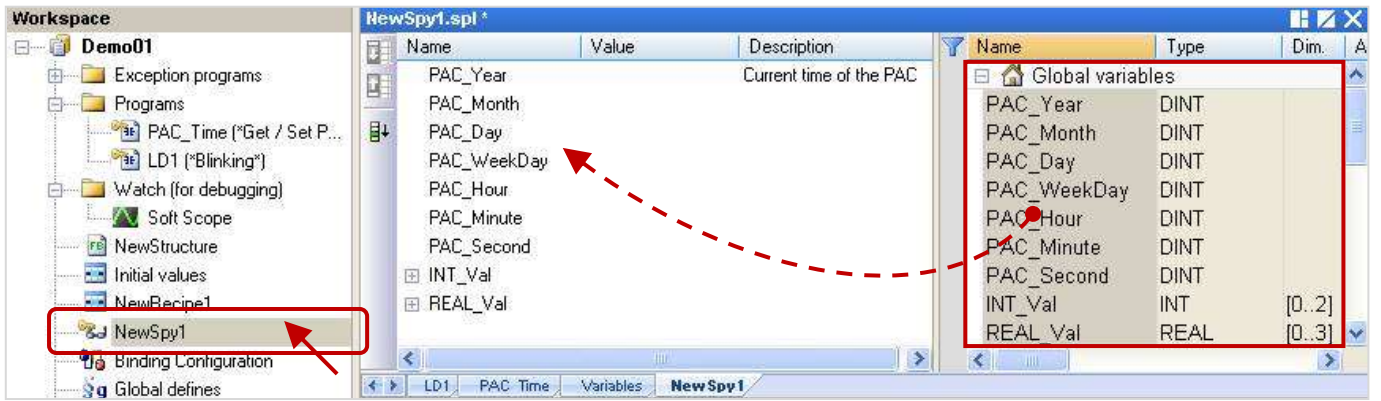
在程式運行時，觀測清單 (Spy List) 可提供使用者快速掌握變數的數值或狀態。有時，一個程式可能宣告了數百~數千個變數，使用者無需費心尋找，只需切換到自己建立的 Spy List 即可看到所需的資料。

方法如下：

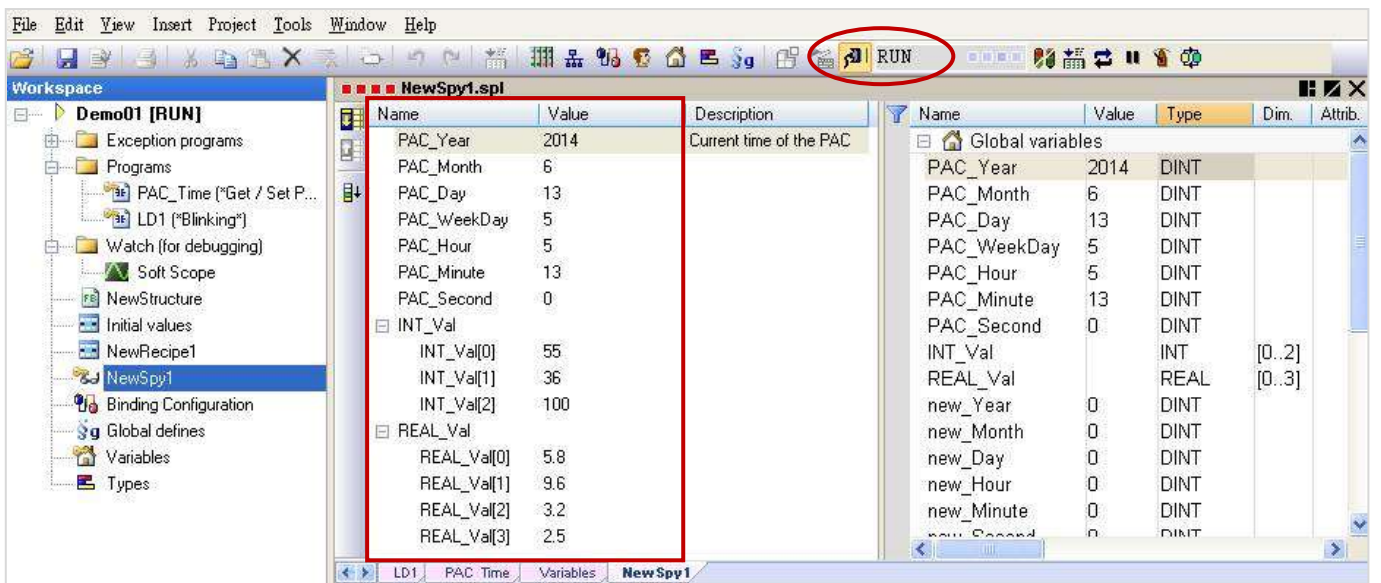
1. 滑鼠右鍵點選專案名稱 (例如: "Demo01") 並選擇 "Insert New Item" 項目。
2. 在 "Watch" 區中選擇 "Spy List"，再點選 "Next" 進行下一步驟。
3. 接著，再輸入清單名稱 (例如: "NewSpy1") 並按 "OK"。



4. 滑鼠雙擊左側的 "NewSpy1" 開啟視窗，並將想要顯示的變數拖曳到此視窗內。



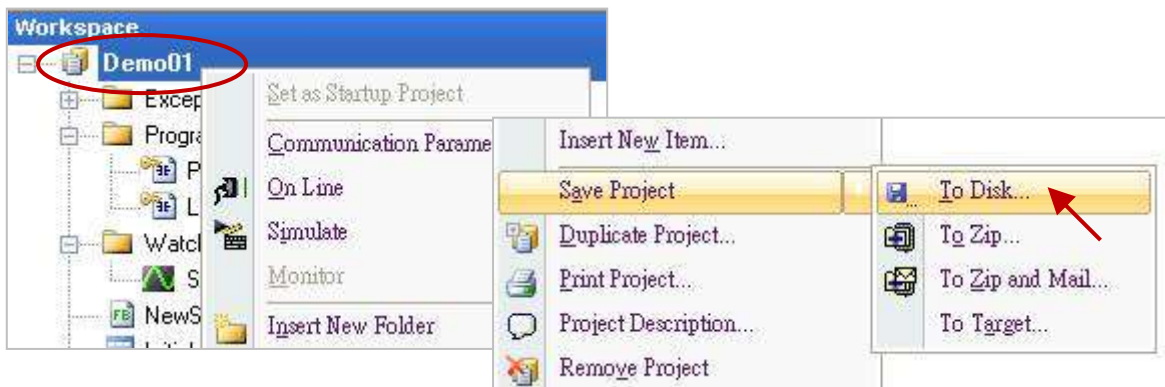
5. 當 Win-GRAF 與 PAC 連線時，此 "NewSpy1" 視窗將會清楚顯示這些變數資料。



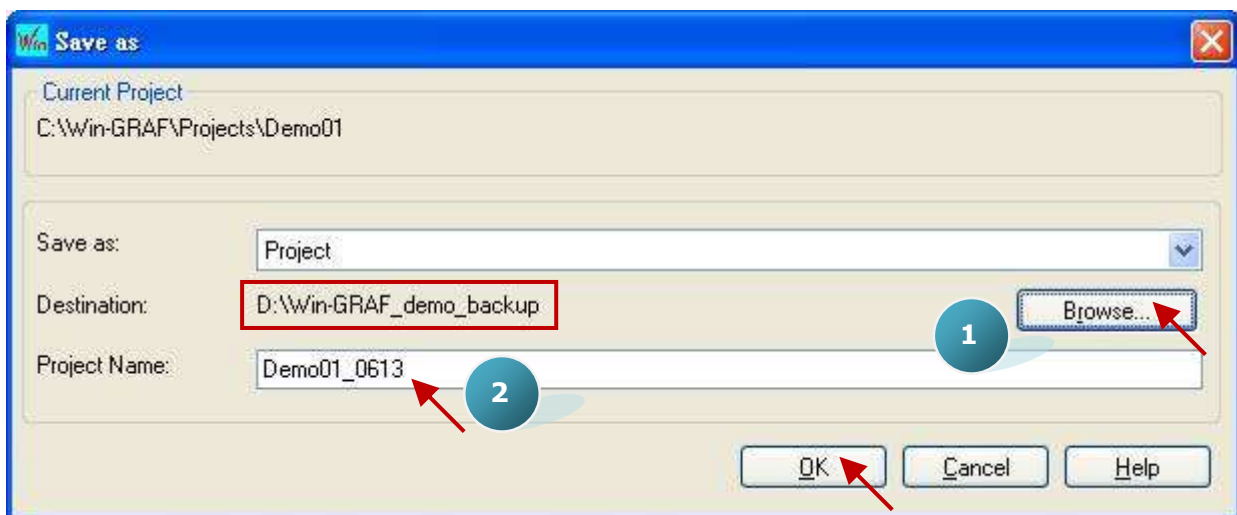
11.4 備份/回存一個 Win-GRAF 專案 (Project)

備份 Win-GRAF 專案:

1. 滑鼠右鍵點選專案名稱 (例如: "Demo01") 並選擇 "Save Project" 內的 "To Disk" 。

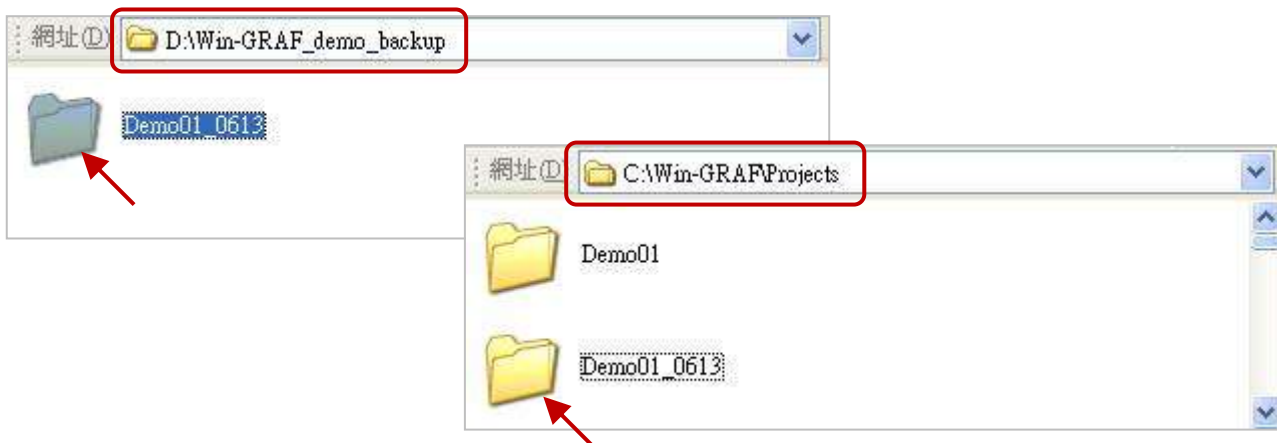


2. 點選 "Browse" 按鈕來指定存檔的路徑 (例如: D:\Win-GRAF_demo_backup) , 並輸入專案名稱 (例如: "Demo01_0613") , 再點選 "OK" 即完成備份 。

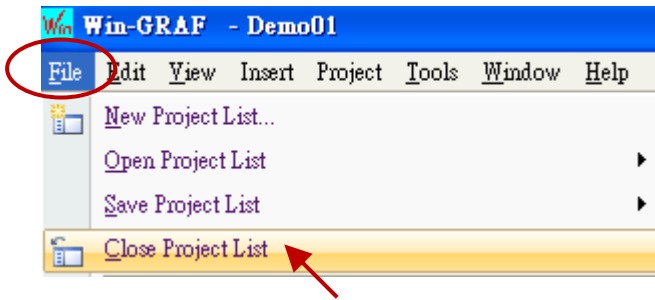


回存 Win-GRAF 專案:

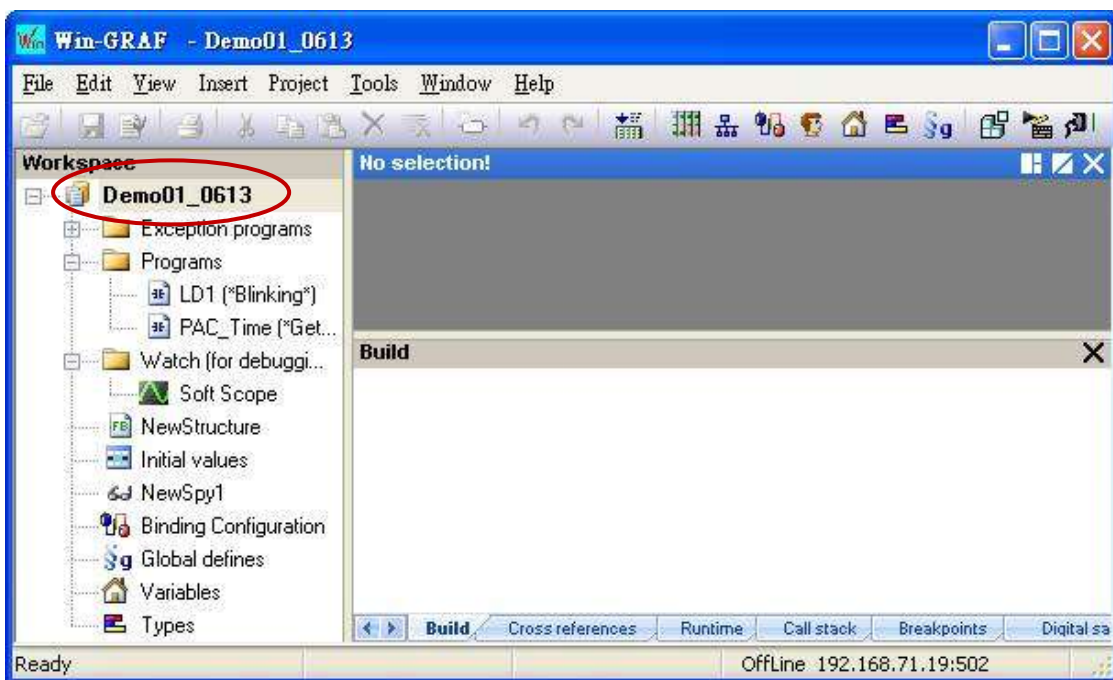
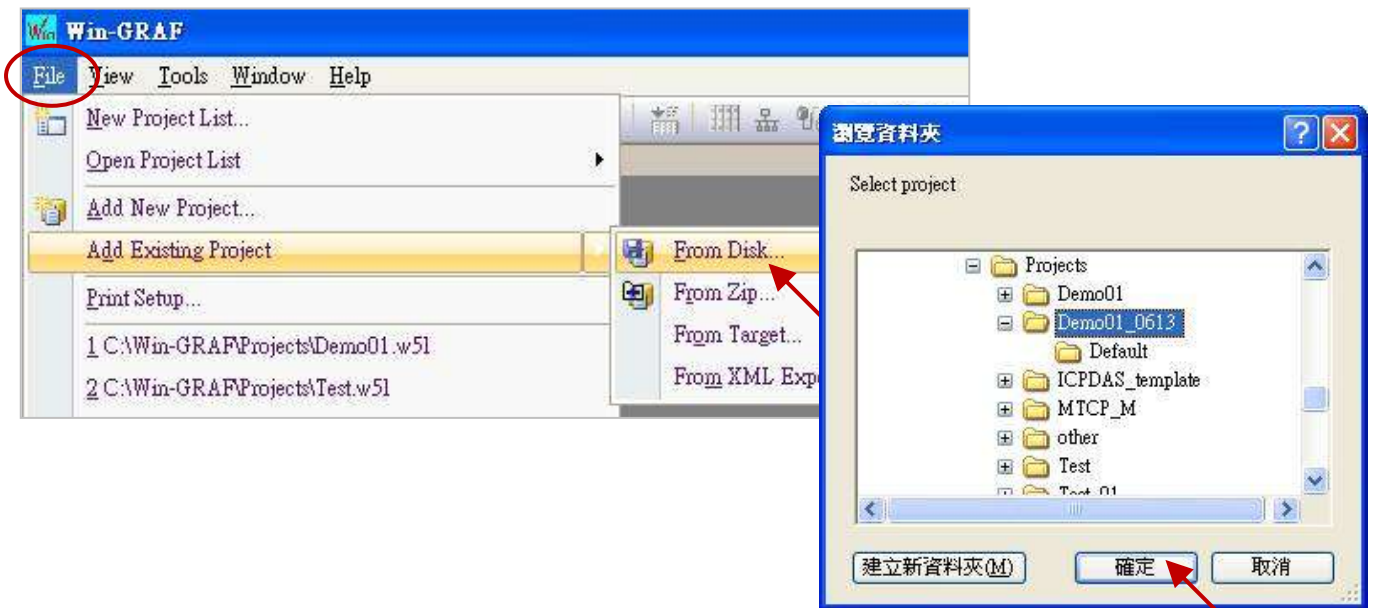
1. 請將先前備份的專案目錄 (例如: "Demo01_0613") 複製到 C:\Win-GRAF\Projects 路徑內 。



2. 滑鼠點選功能表 "File" > "Close Project List" 關閉所有已開啟的視窗。



3. 接著，再點選功能表 "File" > "Add Existing Project" > "From Disk" 項目，並選取 C:\Win-GRAF\Projects 路徑內，要回存的專案目錄 (例如: "Demo01_0613")，再點選 "確定" 即完成回存。



11.5 以軟體重新啟動 PAC

基於某一些情況，使用者可能需要以軟體的方式來重新啟動 PAC。Win-GRAF 提供了 "PAC_Reboot" 函式 (Function) 可調用它來重開機。

註: 請勿在每個 PAC Cycle 調用此功能，否則 PAC 會一直重新開機。

安全寫法:

```
(* "reset_PAC" 宣告為 BOOL 且初值為 "FALSE"  
"TMP_BOOL" 宣告為 BOOL *)  
(* 只有 "reset_PAC" 設定為 "TRUE" 時，才會重開機 *)  
if reset_PAC then  
  reset_PAC := FALSE;  
  TMP_BOOL := PAC_Reboot();  
end_if;
```



危險的寫法:

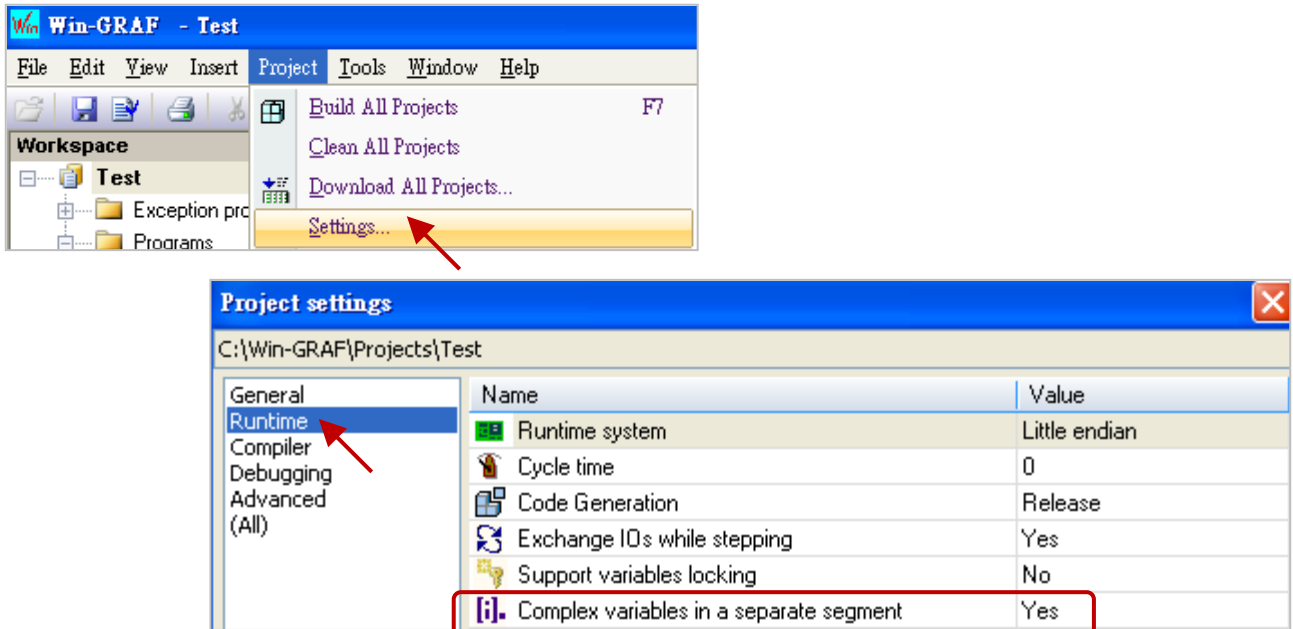
```
(* "TMP_BOOL" 宣告為 BOOL *)  
(* 危險！此寫法會導致 PAC 一直重新開機。 *)  
TMP_BOOL := PAC_Reboot();
```

若出現 PAC 一直重開機的錯誤，請將 Win-GRAF PAC 上的旋轉式開關 (Rotary Switch) 轉到 "1" 並重新開機。開機後會進入安全模式，請修改 PAC 中 Win-GRAF 應用程式為無效的名稱 (例如: "t5.cod1")，再將旋轉式開關轉到 "0"，重新開機後會進入一般模式 (且無應用程式)。

註: Win-GRAF PAC (即，XP-8xx8-CE6, WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7, 與 VP-x2x8-CE7) 裡的應用程式存放在 "\System_Disk\Win-GRAF\t5.cod"。

11.6 在 LD 與 FBD 內使用 ST 語法

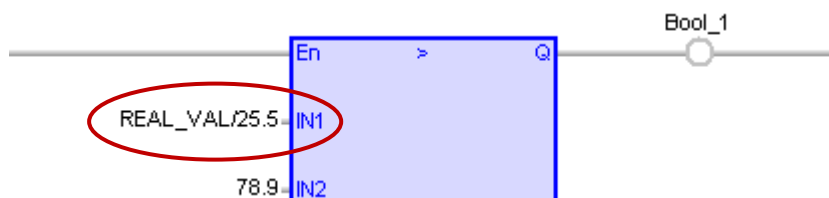
Win-GRAF Workbench 允許使用者在階梯圖 (LD) 與 功能方塊圖 (FBD) 內使用簡單的 ST 語法，以方便程式設計。使用前，請先至功能表 "Project" > "Settings" 的 "Runtime" 區內將 "Complex variables in a separate segment" 啟用為 "Yes"。



以下為使用的例子:

LD 語法:

使用除法 (REAL_VAL/25.5)。



FBD 語法:

呼叫一個 ANY_TO_BYTE() 函式，將 SINT 轉為 BYTE 型態。

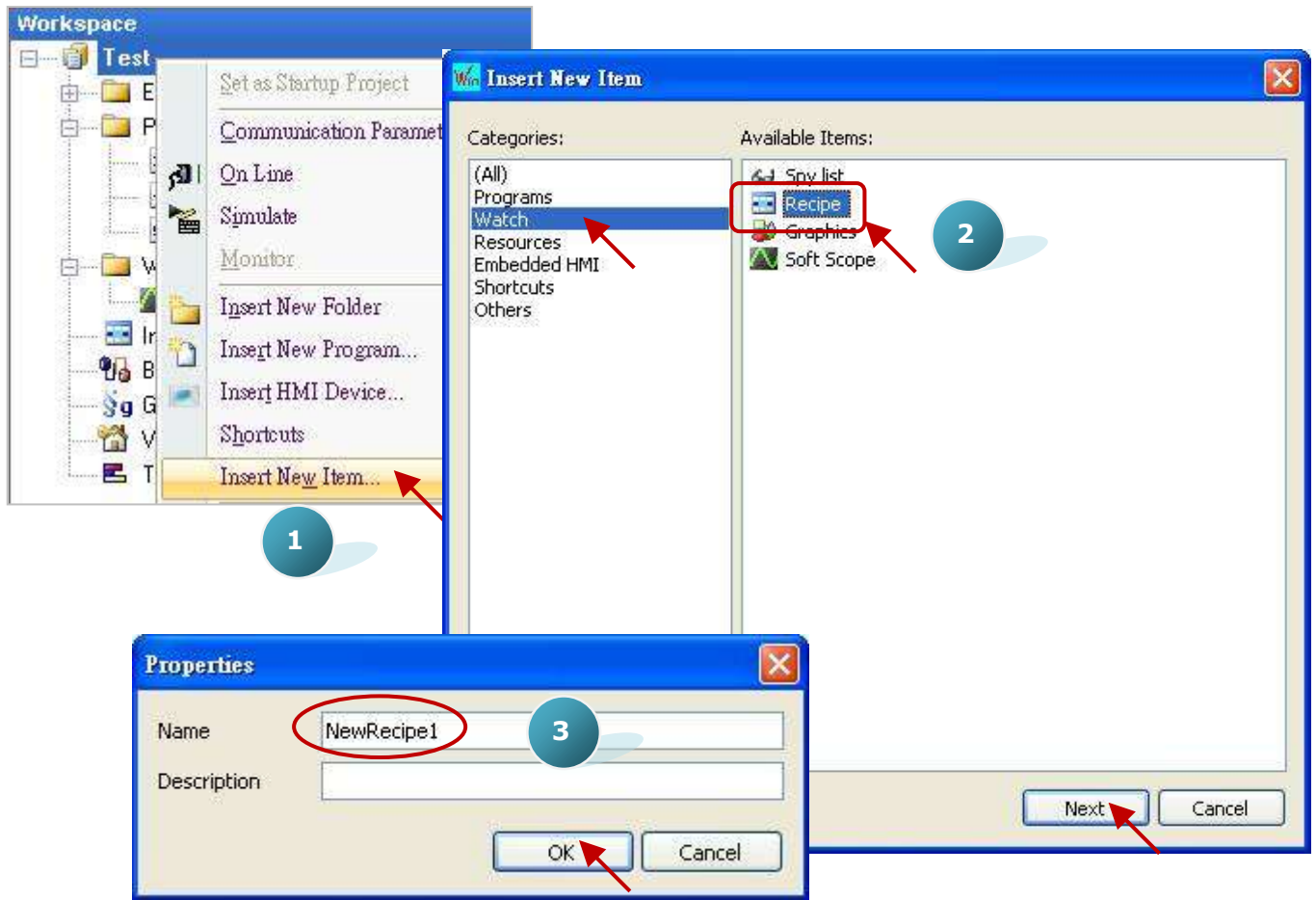


11.7 置換 PAC 內的配方表 (Recipe)

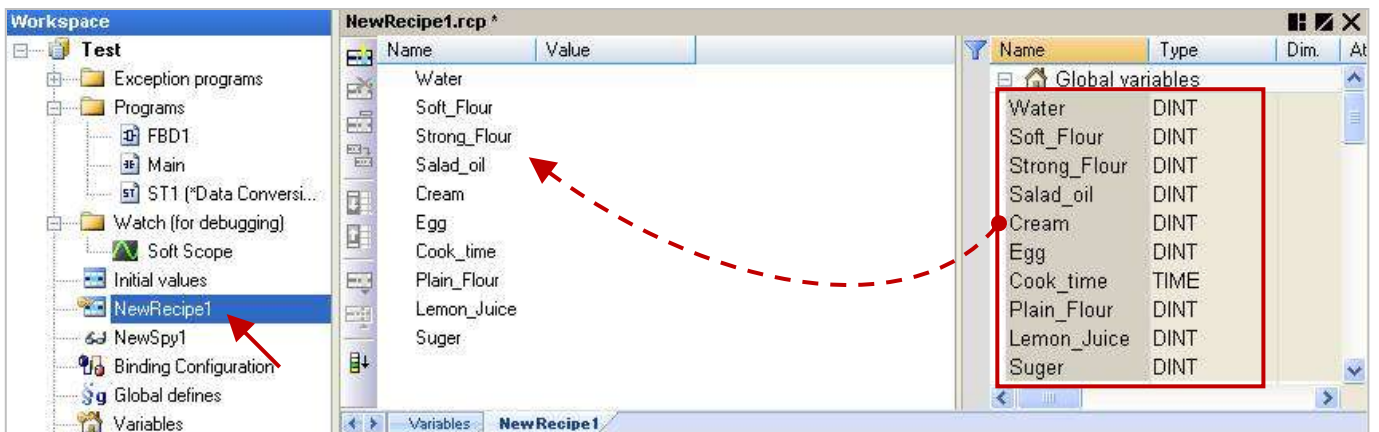
有些應用會預先定義好不同產品所需的配方表 (Recipe) 與數值，而此配方表 (Recipe) 可對應到 Win-GRAF PAC 內一群變數的組合，當某天要變更 PAC 製程來生產不同的產品時，可使用 Win-GRAF Workbench 與 PAC 連線，並選取想要更換的新配方表，再套用到 PAC 內即可。

方法如下：

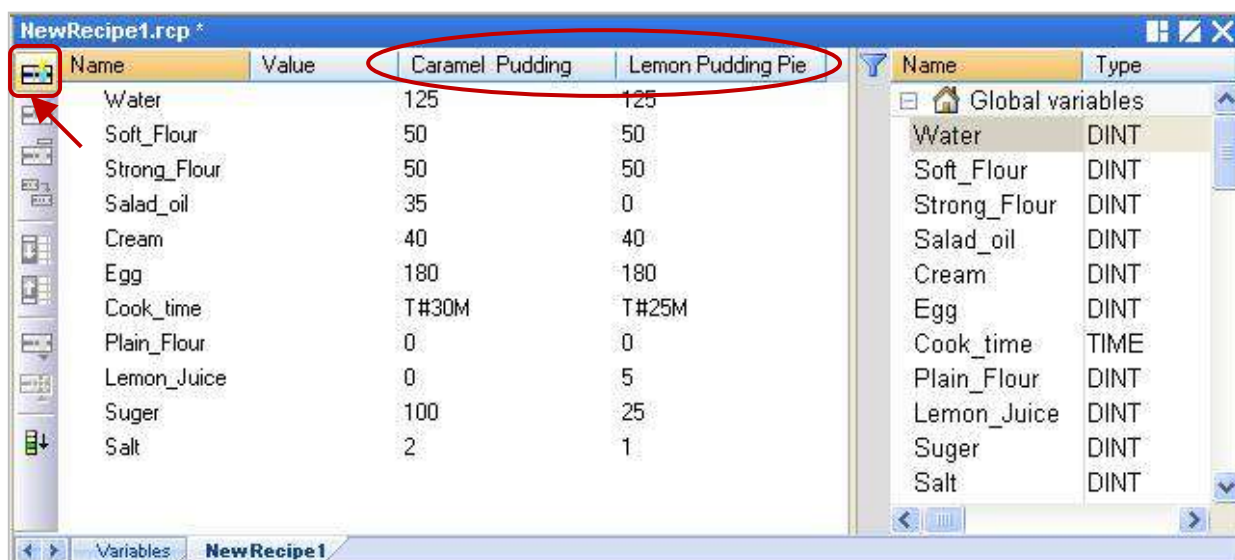
1. 滑鼠右鍵點選 專案名稱 (例如: "Test") 並選擇 "Insert New Item" 項目，在 "Watch" 區中選擇 "Recipe" 並點選 "Next"，接著，再輸入清單名稱 (例如: "NewRecipe1") 並按 "OK"。



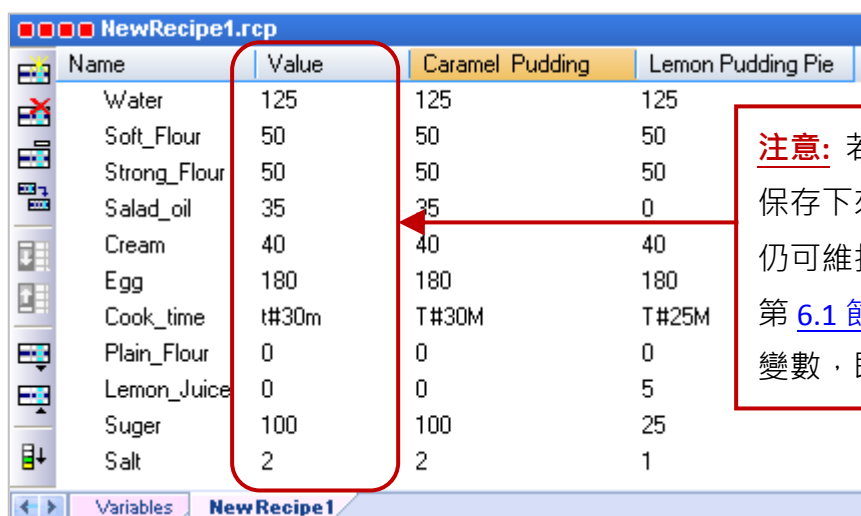
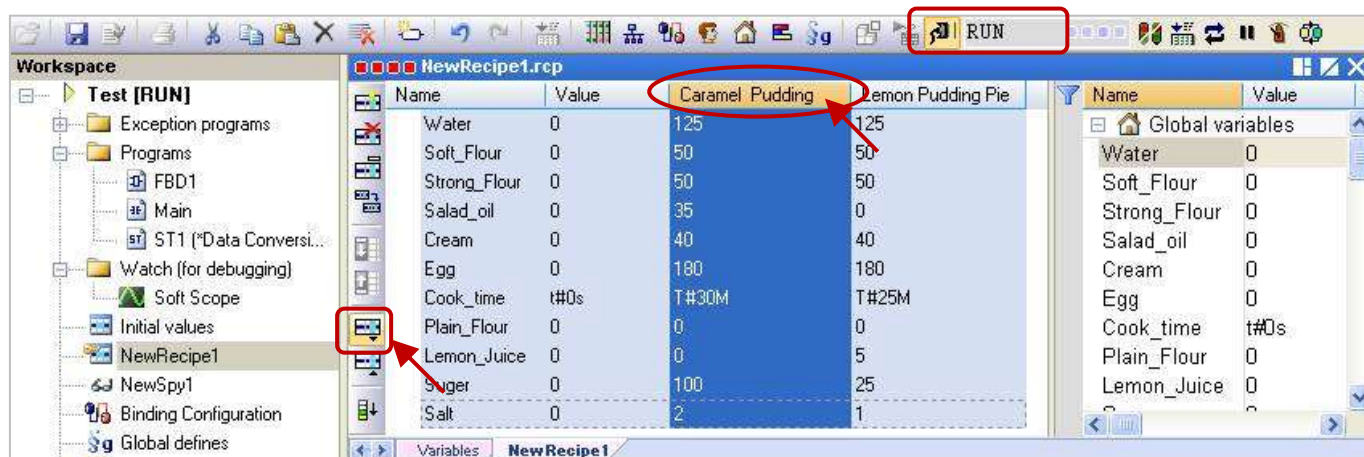
2. 滑鼠雙擊左側的 "NewRecipe1" 開啟視窗，並將需使用的變數拖曳到此視窗內。



3. 點選 "Insert Column" 來新增配方表，並填入適當的數值。



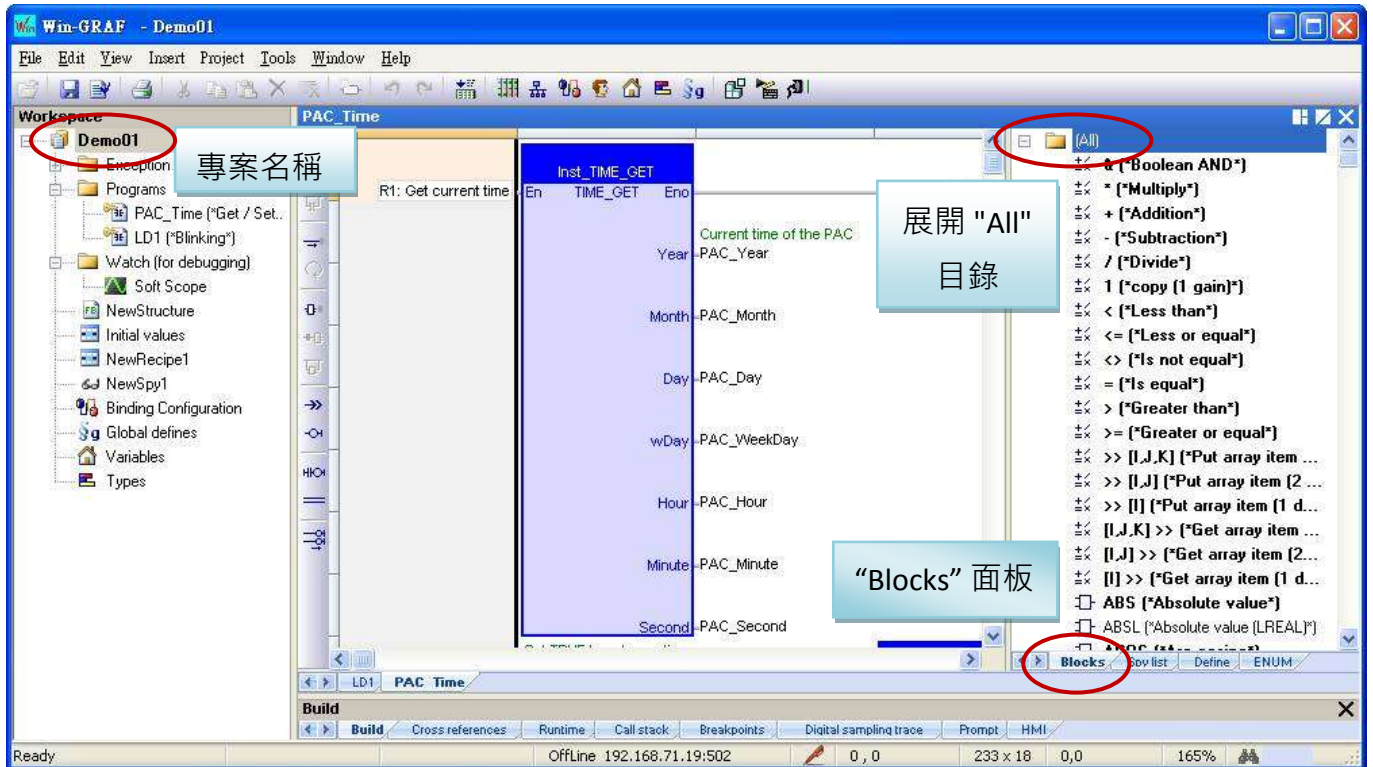
4. 點選 "On Line" 按鈕與 PAC 連線。一開始值皆為 "0"，請點選產品名稱欄位再點選 "Send recipe" 將此配方表套用到 PAC 中。



注意: 若想將 PAC 內的配方數值保存下來 (即 PAC 關機後再開機，仍可維持之前的數值)，請參考第 6.1 節的方法來使用可保存變數，即可達成。

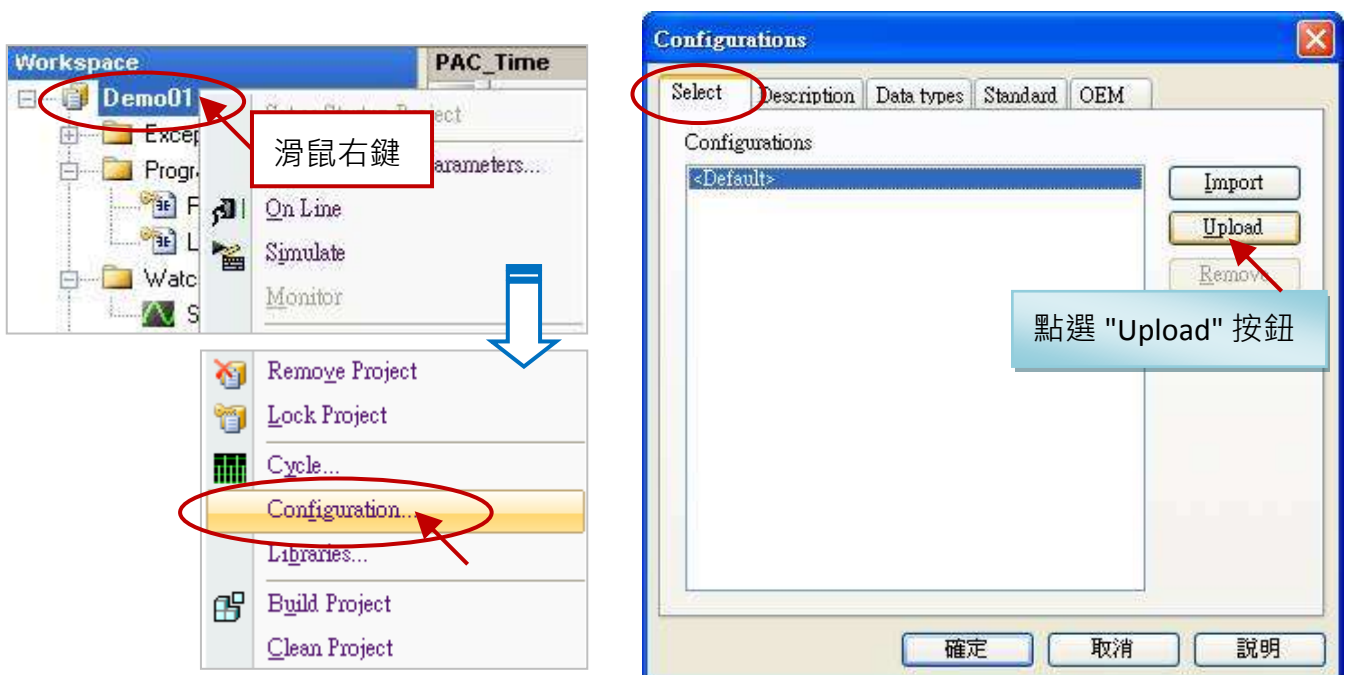
11.8 取得 PAC 所支援的函式 (Function) 與功能方塊 (Function Block)

在 Win-GRAF Workbench 的 "Blocks" 面板內，展開 "All" 目錄可見到相當多的函式 (Function) 與功能方塊 (Function Block)，然而有一些在 Win-GRAF PAC 內並未支援，以下將說明如何快速分辨 PAC 內所支援的函式與功能方塊。

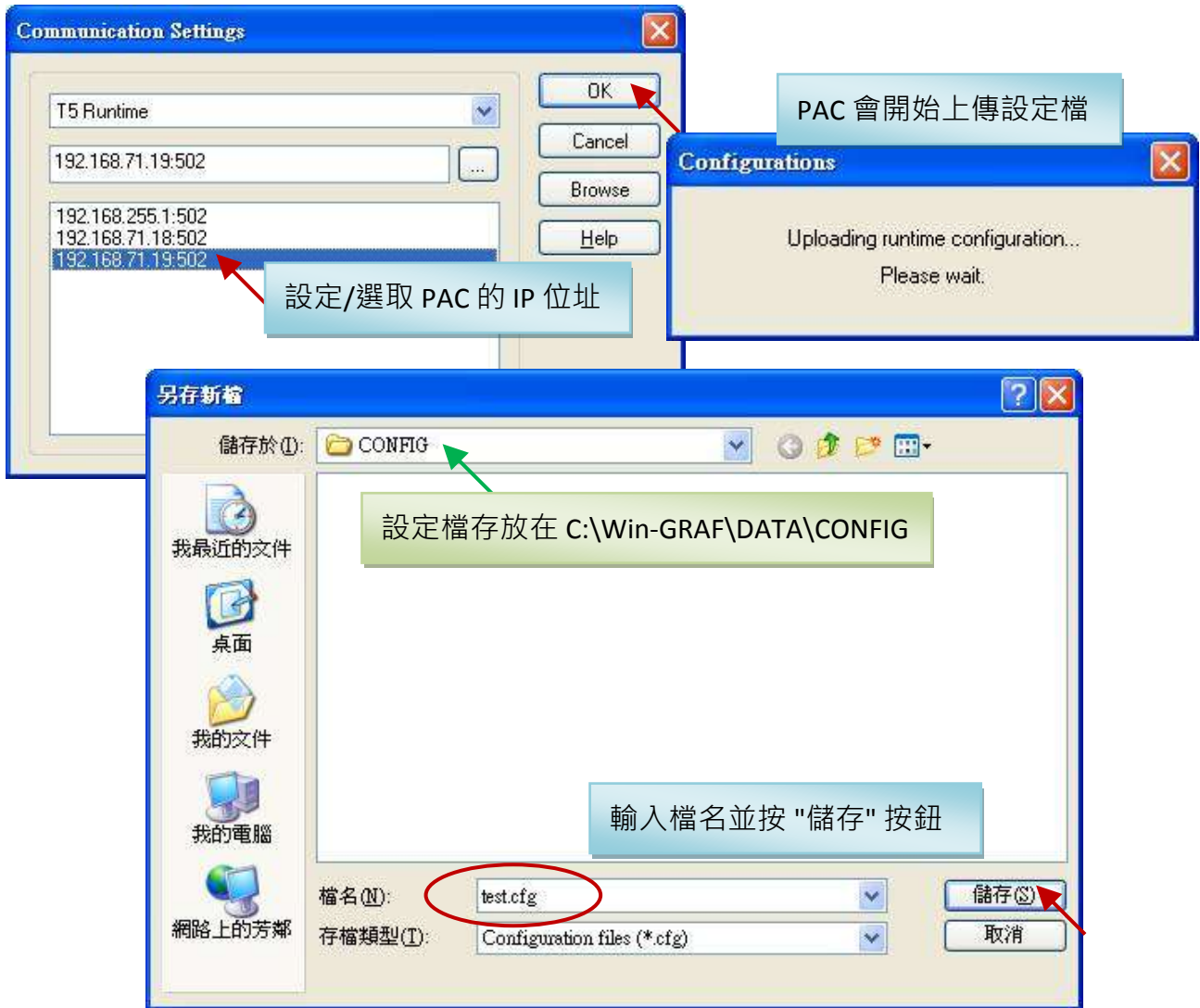


設定方式如下：

1. 請確認 PAC 有開機且和 PC 有用 Ethernet 連上。
2. 於 Win-GRAF Workbench，滑鼠右鍵點選專案名稱 (例如: "Demo01") 再選擇 "Configuration"，並於 "Select" 頁籤中點選 "Upload" 按鈕開啟設定視窗。



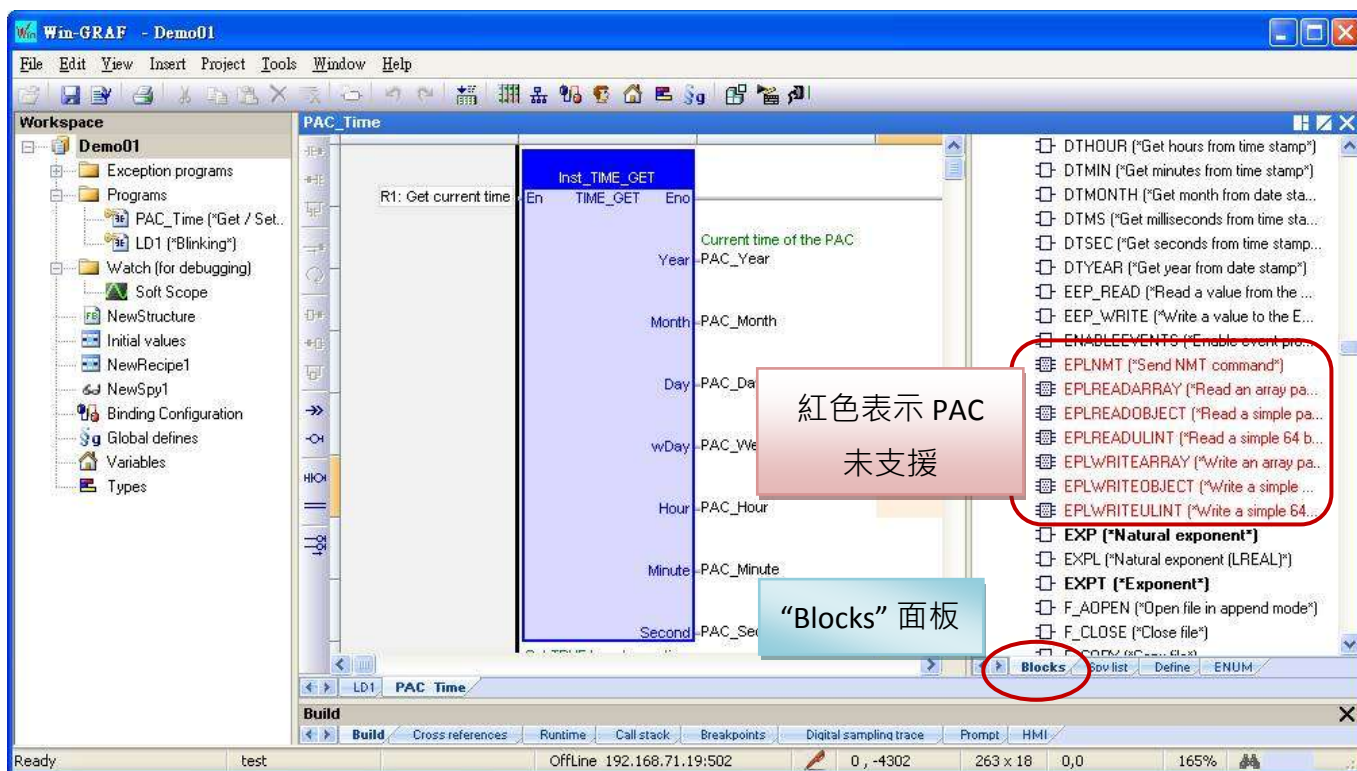
3. 設定/選取 PAC 的 IP 位址並點選 "OK" 按鈕後，PAC 會開始上傳設定檔。接著，請輸入檔名 (例如: "test.cfg") 並按 "儲存" 來儲存此設定檔。



4. 回到 "Configurations" 視窗，可見到剛剛設定的檔名 (test)，請按 "確定" 離開此視窗。



5. 在 "Blocks" 面板內，可見到紅色部分表示 PAC 未支援。

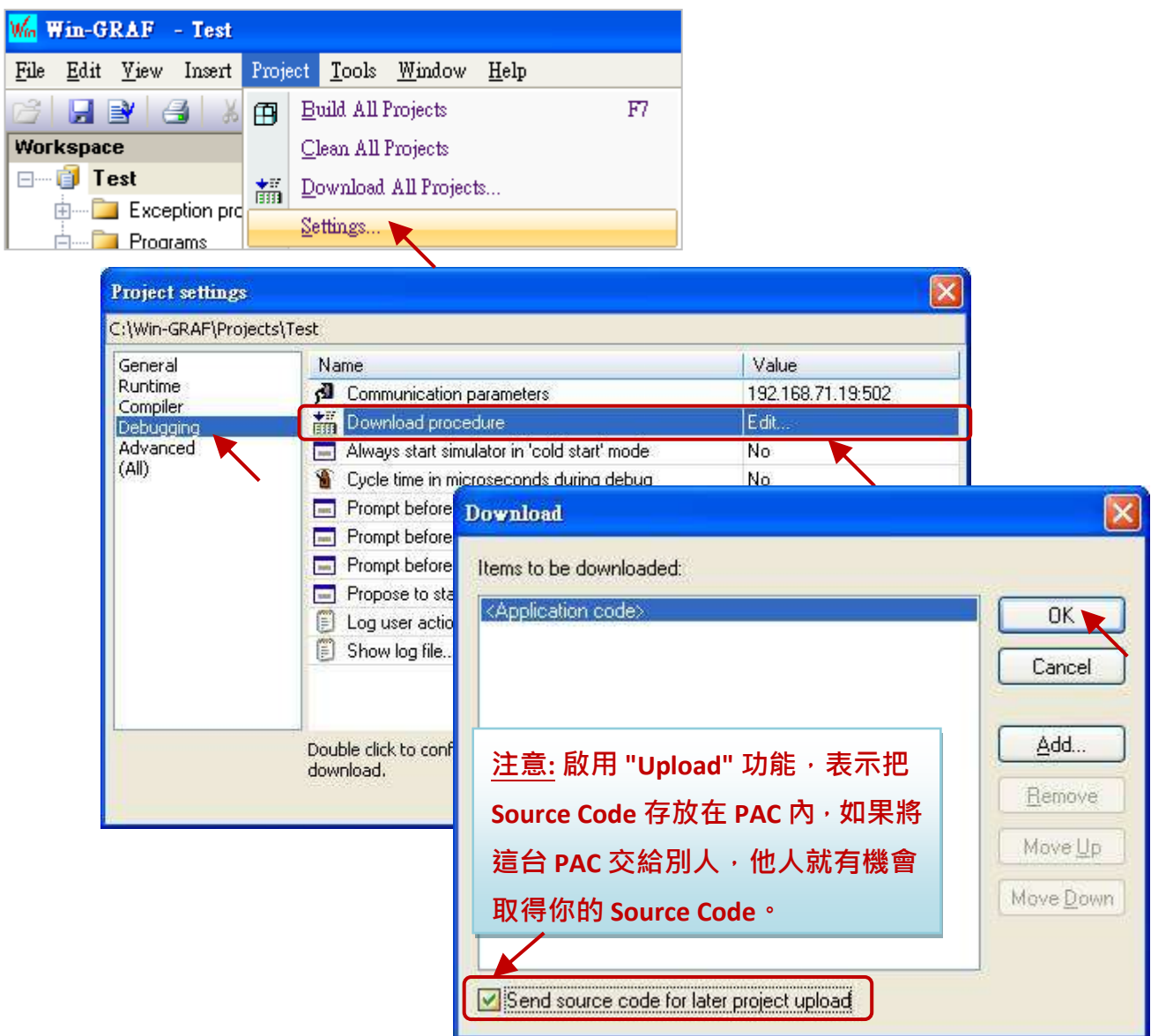



11.9 上傳 Win-GRAF 專案原始碼

有些應用可能會在一段時日後，需要從 PAC 內將 Win-GRAF 專案的原始碼 (Source Code) 抓回到 PC 內，這個功能就叫做 "Upload"。此功能可防止專案的原始碼遺失或前一位工作者移交原始碼不完全，仍可取得 PAC 內的專案原始碼。

啟用 "下載專案原始碼":

1. 滑鼠點選功能表 "Project" > "Settings" 開啟設定視窗。
2. 滑鼠雙擊 "Debugging" 內的 "Download procedure" 項目並勾選 "Send source code for later project upload" 再點選 "OK" 完成設定。

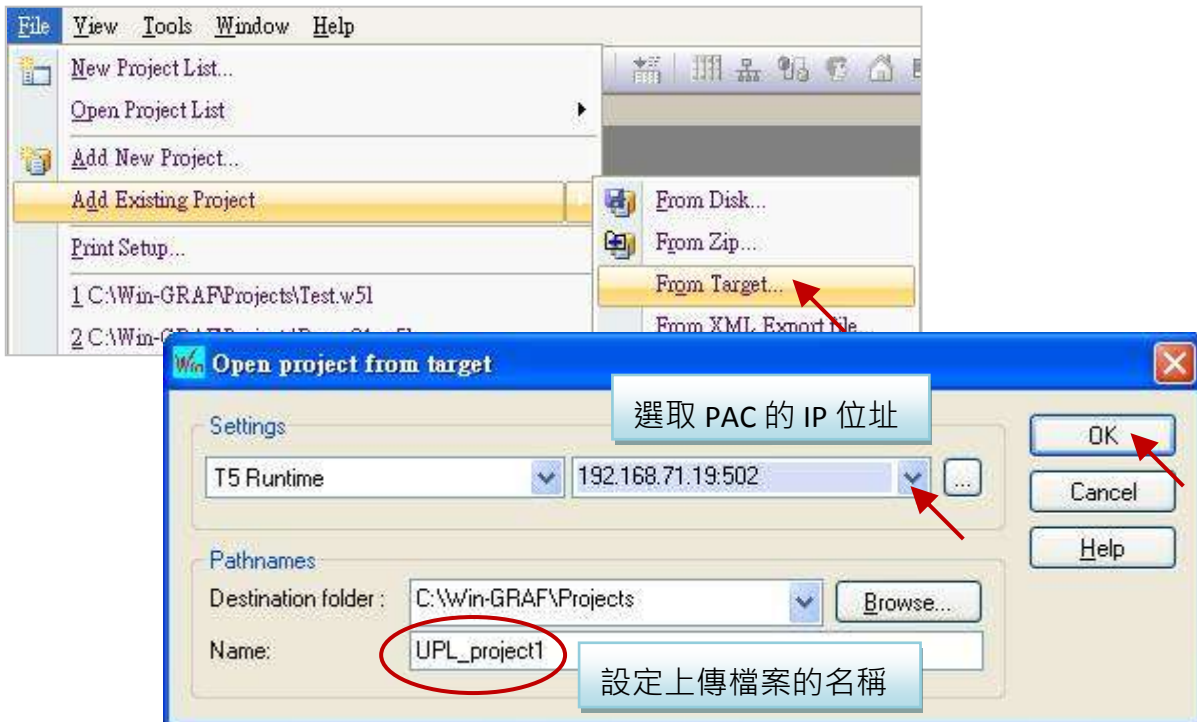


3. 點選功能表 "Project" > "Build All Projects" 編譯程式，並點選工具按鈕  與 PAC 建立連線，接著再將目前的專案下載到 PAC 中 (若不熟悉操作，可參考 [2.3.5 節](#))。下載專案後，Source Code 會存放在 PAC 的 \System_Disk\Win-GRAF\t5.upl，此檔案會隨著專案內容的增加而變大，當專案很大很複雜時，檔案大小有可能會達到幾百 K Byte 或甚至超過 1 MB。

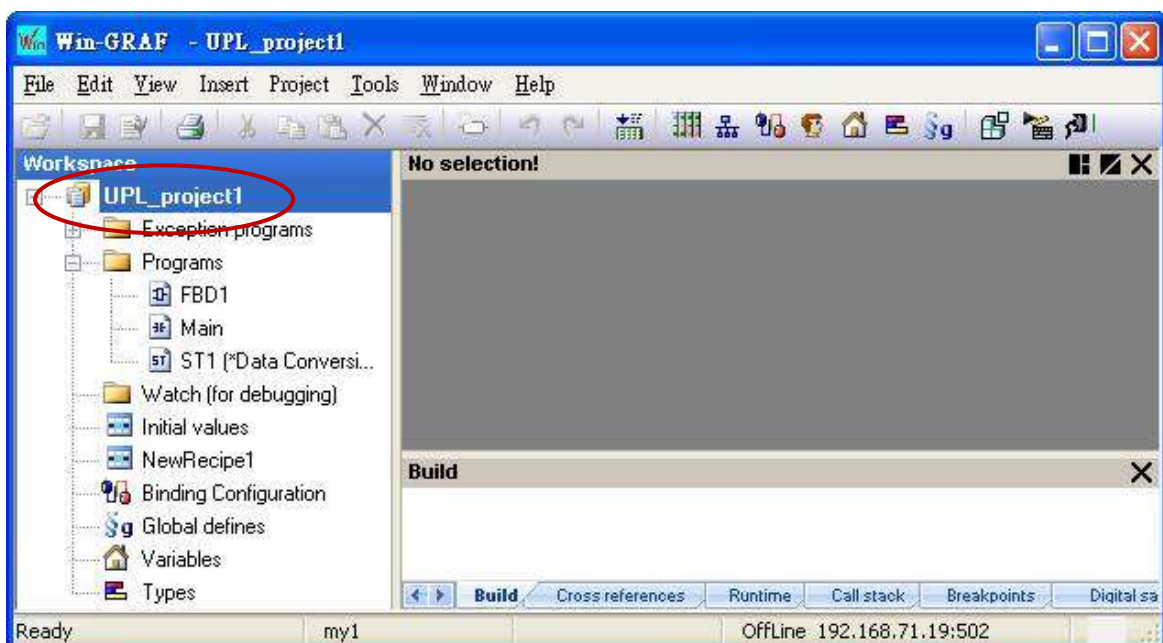
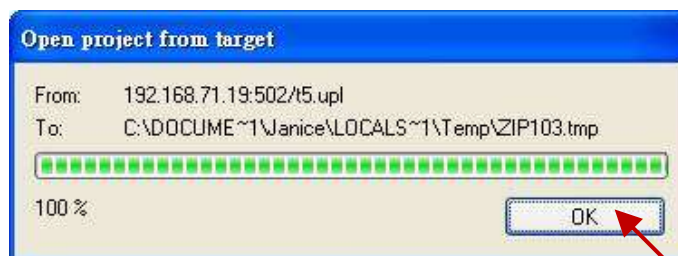
上傳專案原始碼:

請先關閉所有已開啟的 Win-GRAF 視窗 (滑鼠點選功能表 "File" > "Close Project List")。

4. 點選功能表 "File" > "Add Existing Project" > "From Target"，接著 選取 PAC 的 IP 位址 並設定上傳檔案的名稱 (例如: "UPL_project1")，再點選 "OK" 開始上傳檔案。



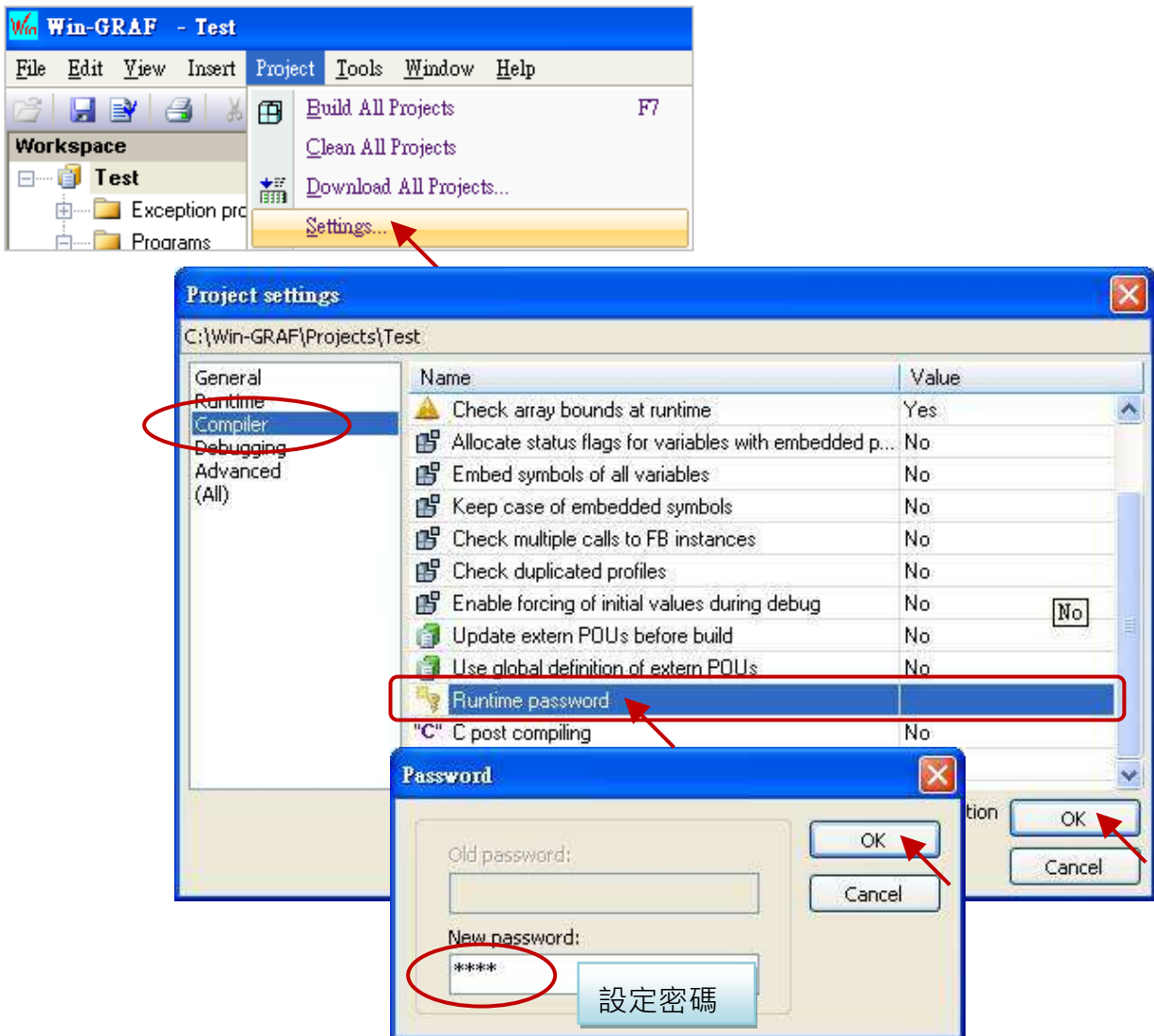
5. 上傳完成後，請點選 "OK" 按鈕。此時，Win-GRAF 會自動開啟此專案。



11.10 設定 PAC 的密碼

為了避免 PAC 內正在運行的重要程式，被某一台 PC 連線並任意的更動或停止運行，您可為 PAC 設定一組密碼來防止未經授權的操作。

1. 滑鼠點選功能表 "Project" > "Settings" 開啟設定視窗。
2. 滑鼠雙擊 "Compiler" 內的 "Runtime password" 項目，並設定密碼 再點選 "OK" 完成設定。



3. 點選功能表 "Project" > "Build All Projects" 再次編譯程式，並將目前的專案下載到 PAC 中即完成設定 (若不熟悉操作，可參考 [2.3.5 節](#))。下次執行 "On Line" 連線時，將會要求輸入密碼。



若執行 "On Line" 連線，需輸入密碼。

注意: 啟用密碼後，請務必記得您所設置的密碼，否則日後您將無法連上 PAC。

唯一的解決方法:

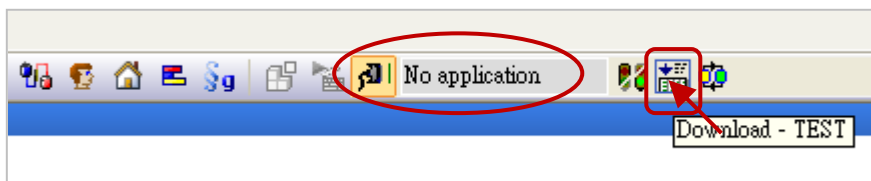
1. 將 PAC 接上 USB 滑鼠與螢幕。
2. PAC 中，開啟 Win-GRAF Driver 並執行 "End Driver" (可參考 [11.2 節](#))。



3. 將 \System_Disk\win-graf 內的 "t5.cod" 檔案重新命名 (例如: "t5.cod1") 或刪除後，再將 PAC 重新開機。



如此，PAC 內將變成 "No application"，此時 Win-GRAF Workbench 可再重新下載應用程式。



11.11 使用 ST 語法來操作功能方塊

使用者如需在 ST 語法內操作函式 (Function) 方法很簡單，只要呼叫該函式 與 填入對應的參數即可。如下的程式在一開始會開啟 COM3，之後每隔 5 秒會從 COM3 送出一個字串 'Hello'。

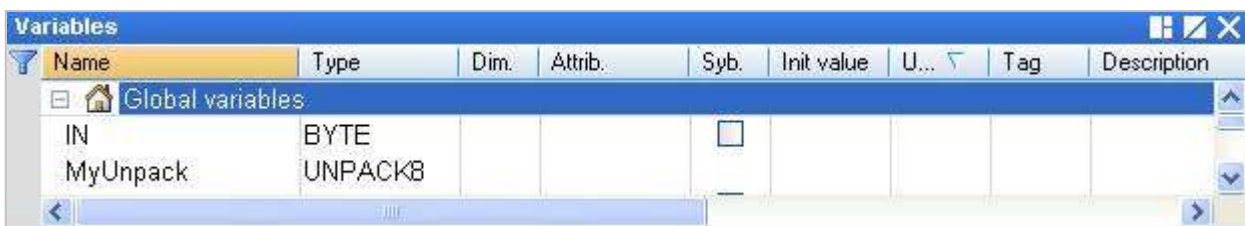
```
(* 宣告 "INIT1" 為 BOOL 並指定初值為 TRUE,
   宣告 "TMP_BOO" 為 BOOL, "TMR1" 為 TIME *)

IF INIT1 THEN
  INIT1 := FALSE;
  TMR1 := T#0s;
  TSTART (TMR1);
END_IF;
IF COM_Status(3) = FALSE THEN
  TMP_BOO := COM_open (3, '19200,N,8,1');
END_IF;
IF TMR1 >= T#5s THEN
  TMR1 := T#0s;
  COM_send_str (3, 'Hello: ');
END_IF;
```

若要在 ST 內使用功能方塊 (Function Block)，需先在變數區宣告要使用的功能方塊的樣例變數 (Instance)，之後的用法就類似函式的用法如下：

以下的程式碼可以把 1 個 byte 拆解成 8 個 BOOL：

1. 宣告 "MyUnpack" 變數為 UNPACK8 (FB Instance)，"IN" 變數為 BYTE。



Name	Type	Dim.	Attrib.	Syb.	Init value	U...	Tag	Description
Global variables								
IN	BYTE			<input type="checkbox"/>				
MyUnpack	UNPACK8							

2. 然後寫一個 ST 程式。

```
MyUnpack(IN);
Q0 := MyUnpack.Q0;
Q1 := MyUnpack.Q1;
Q2 := MyUnpack.Q2;
Q3 := MyUnpack.Q3;
Q4 := MyUnpack.Q4;
Q5 := MyUnpack.Q5;
Q6 := MyUnpack.Q6;
Q7 := MyUnpack.Q7;
```

11.12 如何保護您的 Win-GRAF 程式，讓盜用者無法使用？

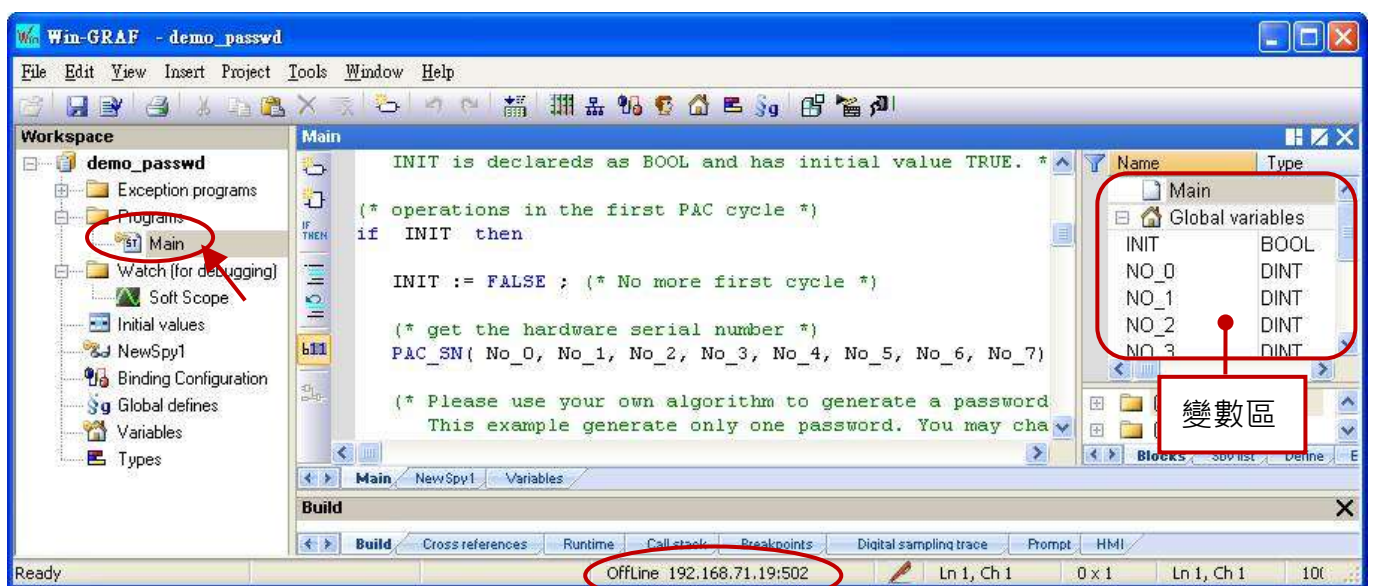
當您完成 Win-GRAF 應用程式開發，並準備交貨給客戶時，請先想一想，是否有可能您交出去的 Win-GRAF PAC 內的應用程式，將被複製到另一台同型號的 PAC 內？！小心！或許第三者就是這樣盜用了您辛苦開發完的成果，以下提供一個簡單好用的方法來保護您的應用程式。

注意：如果您是把 Win-GRAF 應用程式的 Source Code 交給了客戶，那很抱歉，以下的方法將無法保護您的程式不被盜用。因為有了 Source Code，盜用者就可以自行修改程式碼並套用在另一台 PAC 內。

每一台 ICP DAS Win-GRAF PAC 都會有一個序號 (Serial Number)，此序號有 8 個 Byte (或稱 64 Bit)，而且每台 PAC 的序號都不同。因此，可利用這個序號再加上您自訂的運算來產生一組密碼，再將此密碼預先存到 PAC 的檔案內。之後，在您的應用程式裡去驗證它，若不通過，該應用程式將不允許運作。其方法如下：

此範例使用了 2 個 Win-GRAF 專案，一個是 "demo_passwd" 用來產生 PAC 的密碼並存到 PAC 的檔案內，另一個是 "demo_my_ap" 已開發好並準備出貨給客戶的應用程式。每次 PAC 要出貨之前，必需先把 "demo_passwd" 程式下載到 PAC 內 Run 一次，它會產生專屬於那台 PAC 的密碼。然後，使用者再下載 "demo_my_ap" 到同一台 PAC 內，之後再出貨給客戶。此後，第三者若複製此 PAC 內的 Win-GRAF 應用程式到另一台同型號的 PAC 內，會因為密碼驗證不通過而運作失敗。

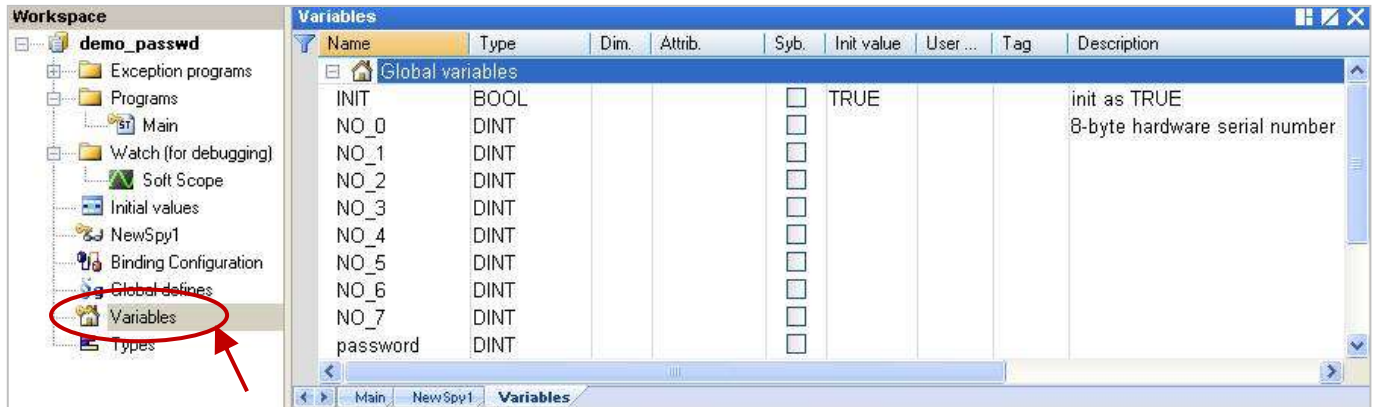
於出貨光碟中 (\Napdos\Win-GRAF\demo-project) 提供了此 2 個範例程式 (demo_passwd.zip 與 demo_my_ap.zip)，請參考 [第 12 章](#) 來回存此專案 (執行 File > Add Existing Project > From Zip) 並設定好 PAC 目前的 IP 位址。



"demo_passwd" 應用程式:

此程式先使用 "PAC_SN" 來讀出序號，再經過使用者自訂的演算法來產生密碼。最後，將該密碼存到 PAC 的檔案內 (使用者可自行決定要存放何處)。

變數宣告:



ST 程式:

(* 此 "demo_passwd" 範例程式會依據 PAC 內 8-byte 的硬體序號來產生一組密碼，並將它儲存到 PAC 內 \System_disk\Win-GRAF\ my_product.pwd 檔案中。)

(* 宣告 "No_0" ~ "No_7" 與 "password" 變數為 DINT。
宣告 "INIT" 變數為 BOOL 且初值 (Initial value) 為 TRUE。*)

(* 第一個 PAC Cycle 的操作 *)

if INIT then

 INIT := FALSE ; (* 表示不再是第一個 Cycle *)

(* 取得硬體的序號 *)

PAC_SN(No_0, No_1, No_2, No_3, No_4, No_5, No_6, No_7);

(* 請使用您自定的演算法來產生一組密碼。本範例僅產生一組密碼，您可修改程式來產生多組密碼 *)

password := (No_0 * No_1) + (No_2 * 12345) + No_3 + (No_4 * No_5) + No_6 + No_7 ;

(* 將密碼儲存到 \System_disk\Win-GRAF\ my_product.pwd 檔案中 *)

file_name := '\System_Disk\Win-GRAF\my_product.pwd' ;

file_id := f_wopen(file_name) ;

if file_id = 0 then

 (* 若失敗，不做任何事 *)

else

 (* 若檔案開啟成功，將密碼儲存進去 *)

 fm_write(file_id , Any_to_String(password)) ;

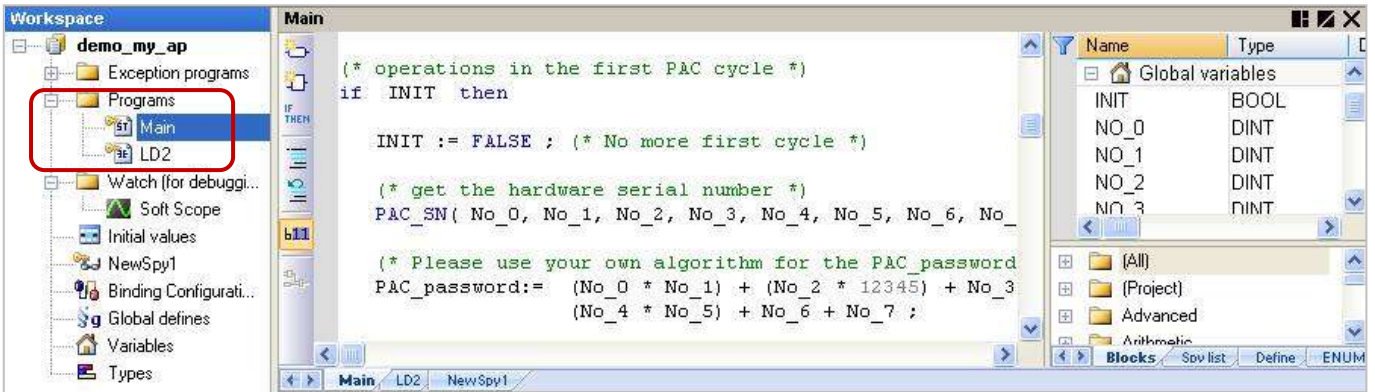
 f_close(file_id) ; (* 關閉檔案 *)

end_if ;

end_if ;

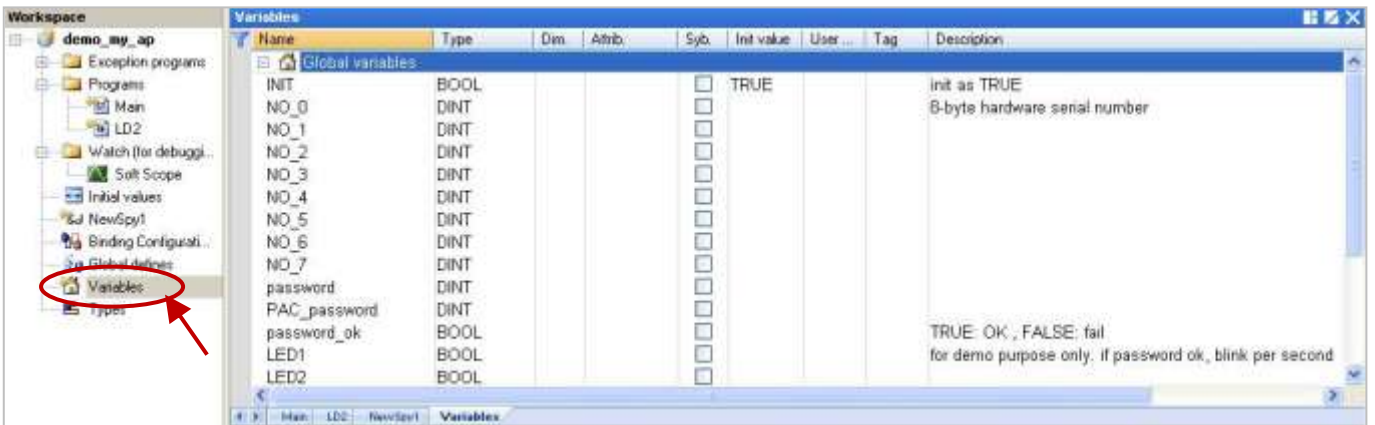
"demo_my_ap" 應用程式:

一開始會使用 "PAC_SN" 函式來讀出序號，接著再算出密碼，另外也會從 PAC 的檔案內讀出密碼來比對密碼是否正確？



(註: 您可參考 [2.1.2 節](#)，讓程式依照執行的順序來排列)

變數宣告:



ST 程式 - Main:

(* 此 "demo_my_ap" 範例程式可由 PAC 內 \System_disk\Win-GRAF\ my_product.pwd 檔案中讀出密碼，並與使用者自訂的演算結果相互比對看密碼是否正確? *)

(* 宣告 "No_0" ~ "No_7", "password" 與 "PAC_password" 變數為 DINT。
宣告 "INIT" 變數為 BOOL 且初值 (Initial value) 為 TRUE。
宣告 "password_ok" 變數為 BOOL *)

(* 第一個 PAC Cycle 的操作 *)

if INIT then

INIT := FALSE ; (* 表示不再是第一個 Cycle *)

(* 取得硬體的序號 *)

PAC_SN(No_0, No_1, No_2, No_3, No_4, No_5, No_6, No_7);

(* 使用您自訂的演算法來產生 "PAC_password" 的值 *)

```
PAC_password:= (No_0 * No_1) + (No_2 * 12345) + No_3 + (No_4 * No_5) + No_6 + No_7 ;
```

(* 讀出先前儲存在 \System_disk\Win-GRAF\ my_product.pwd 檔案中的密碼 *)

```
file_name := '\System_disk\Win-GRAF\my_product.pwd' ;
```

```
file_id := f_ropen( file_name ) ;
```

```
if file_id = 0 then
```

```
  (* 若無法開啟檔案, 設定密碼為 0 *)
```

```
  password := 0 ;
```

```
else
```

```
  (* 若開啟成功, 讀取密碼 *)
```

```
  if f_eof( file_id ) then
```

```
    (* 到達檔案的最末端 *)
```

```
  else
```

```
    (* 尚未抵達 file 尾端, 讀取一個 String *)
```

```
    Tmp_string := fm_read( file_id ) ;
```

```
    (* 將 String 轉換為 DINT 值 *)
```

```
    password := Any_to_DINT(Tmp_string) ;
```

```
  end_if ;
```

```
  f_close(file_id) ; (* 關閉檔案 *)
```

```
end_if ;
```

(* 比對密碼是否正確? *)

```
password_ok := FALSE ; (* 一開始先設為 "FALSE" *)
```

```
if password = PAC_password then
```

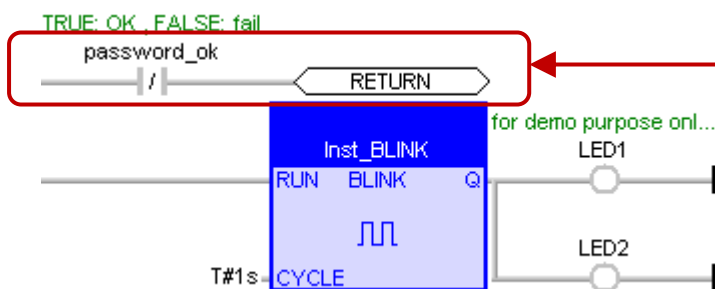
```
  password_ok := TRUE ; (* 密碼正確 *)
```

```
end_if ;
```

```
end_if ;
```

LD 程式 – LD2

如果 "password_ok" 為 "FALSE" 表示密碼不正確，會離開此程式。只有密碼正確時，才能執行後續的程式，如此即可保護您的應用程式無法被盜用者使用。

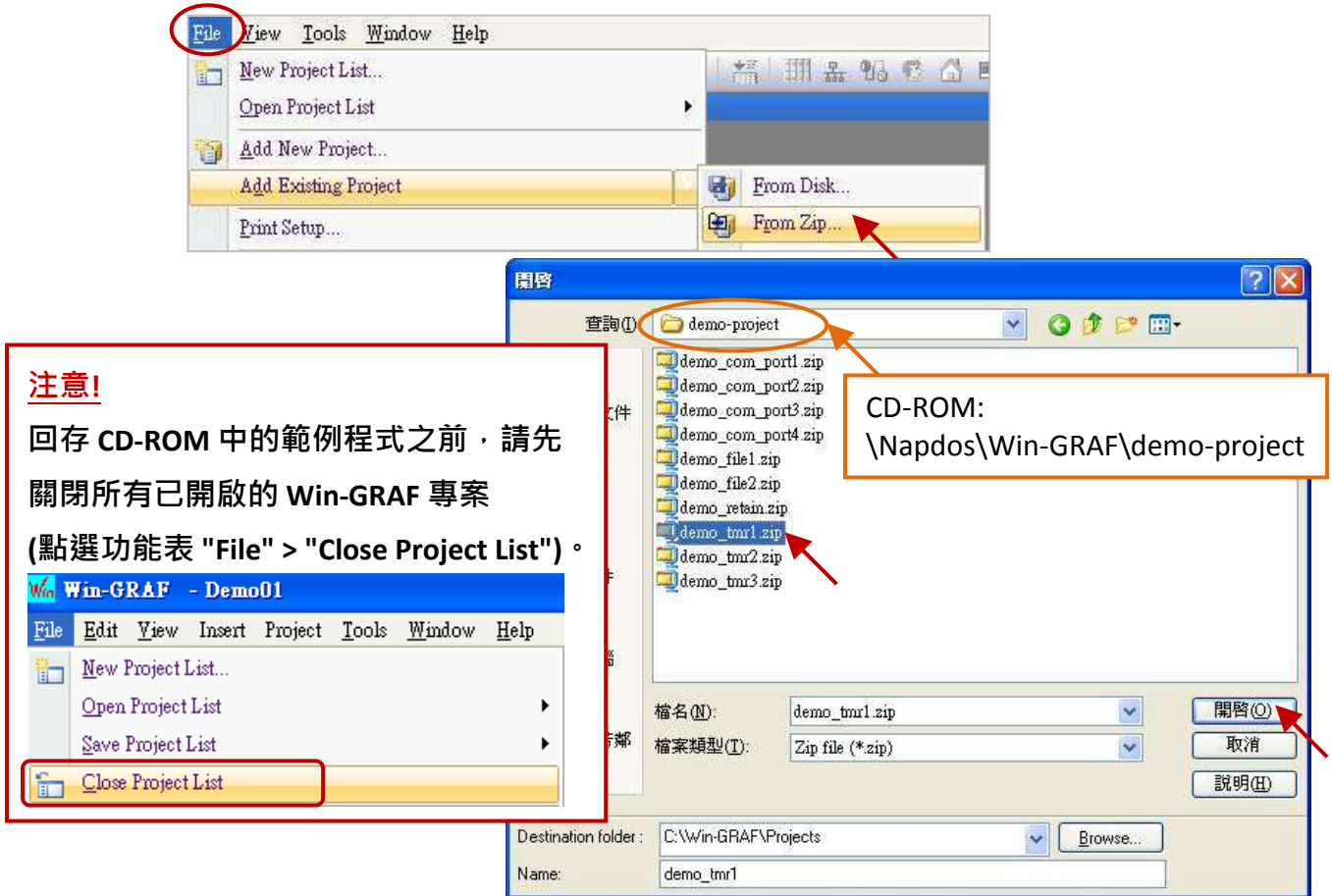


本例只使用一個 "LD2" 程式，若您的應用還有其它程式，請為每個程式加上判斷 "password_ok" 的程式碼。

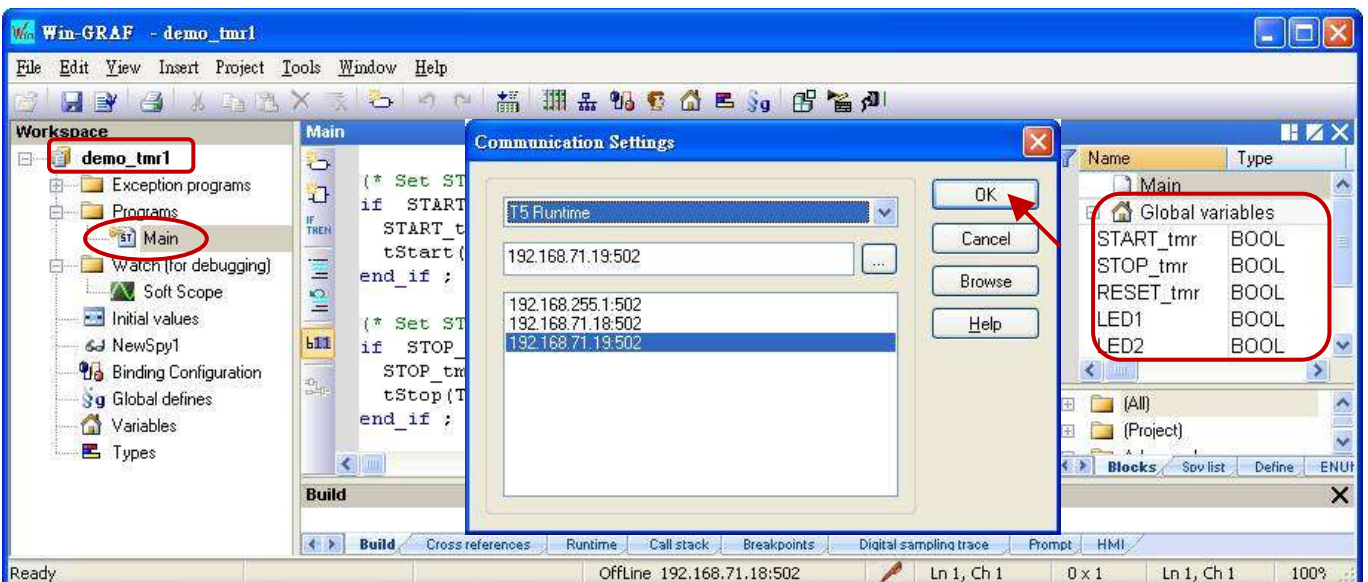
第 12 章 範例程式說明

在 Win-GRAF PAC 的出貨光碟中 (CD-ROM: \Napdos\Win-GRAF\demo-project) ，提供了以下章節的範例程式。使用前，請先進行以下步驟：

1. 點選功能表 "File" > "Add Existing Project" > "From Zip" ，來開啟專案 (例如: "demo_tmr1.zip") 。



2. 滑鼠雙擊 "Main" 可開啟此 ST 程式，於變數區可查看/建立變數。
3. 滑鼠右鍵點選專案名稱 ("demo_tmr1") 再選擇 "Communication Parameters" 設定 PAC 目前的 IP 位址。(可參考 [2.3.5 節](#))



12.1 範例程式列表

Win-GRAF PAC 出貨包裝盒內包含一張名為 Win-GRAF-PAC-CD 的 CD-ROM，在它的 \Napdos\Win-GRAF\demo-project 路徑內可以找到很多關於使用 Win-GRAF PAC 的範例程式，請參考以下列表來回存範例程式到 Win-GRAF Workbench 內。

以下是範例程式列表：

檔案名稱	說明
demo_tmr1	使用 tStart 與 tStop 函式來操作計時器 (參考 12.2.1 節)。
demo_tmr2	對計時器進行周期性的操作 (參考 12.2.2 節)。
demo_tmr3	較精準的對計時器進行周期性的操作 (參考 12.2.2 節)。
demo_com_port1	使用 COM Port 來傳送一個字串 (參考 12.3.1 節)。
demo_com_port2	使用 COM Port 對設備一問一答 (參考 12.3.2 節)。
demo_com_port3	COM Port 等待接收遠端設備傳來的資料 (參考 12.3.3 節)。
demo_com_port4	使用 COM Port 定期回報資料給遠端設備 (參考 12.3.4 節)。
demo_file1	寫入資料到 PAC 內的檔案 (參考 12.4.1 節)。
demo_file2	讀取 PAC 內的檔案 (參考 12.4.2 節)。
demo_retain	使用可保存變數來保有 PAC 停電前的變數資料 (參考 6.1 節)。
demo_wp5_retain	使用檔案來保存變數 (參考 6.2 節)。
demo_extra_port	啟用一個序列埠讓 Win-GRAF Workbench 連進來 (參考 附錄 E)。
demo_my_ap	保護您的 Win-GRAF 程式，讓盜用者無法使用 (參考 11.12 節)。
demo_passwd	
demo_XV310	WP-5xx8-CE7 使用 XV-Board (參考 5.1.6 ~ 5.1.12 節)。
demo_XV308_1	
demo_XV308_2	
demo_XV308_3	
demo_XV116	
demo_XV111	
demo_XV110	
demo_XV107	
demo_vb01	使用 VB.net 2008 程式來讀/寫 Win-GRAF 變數 (參考第 13 章)。
demo_vb02	
demo_vb03	
demo_vb04	
demo_PID_simple	PID 應用 與 調節器應用。
demo_user_C	開發屬於自己的 Function 與 Function Block (參考第 18 章)。

檔案名稱	說明
demo_rdn_1	備援 (冗余) 系統 (參考第 16 章)。
demo_rdn_2	
demo_rdn_3	
demo_ET7060	連接遠端 ET-7060 網路 I/O 模組 (參考 5.2.2 節)。
demo_ET7018z	連接遠端 ET-7018Z 網路 I/O 模組 (參考 5.2.3 節)。
DEMO_DL_100T485	連接遠端 DL-100T485 模組來測量溫度與濕度 (參考 8.2.7 節)。
demo_schedule	排程控制 (參考第 17 章)。
demo_d_7065	連接遠端 I-7065 I/O 模組 (參考 8.2.1 節)。
demo_d_7018z	連接遠端 I-7018Z I/O 模組 (參考 8.2.2 節)。
DEMO_D_7083	連接遠端 I-7083 I/O 模組 (參考 8.2.3 節)。
demo_8088w	PWM 輸出 (參考 4.11 節)。
DEMO_D_87084_FR	連接遠端 I-87084W 模組來量測頻率 (參考 8.2.4 節)。
DEMO_D_87084_C4	連接遠端 I-87084W 模組來量測 Counter 值 (參考 8.2.5 與 8.2.6 與 4.9 節)。
DEMO_D_87084_C8	
demo_SMS	使用一個 2G/3G GSM Modem 來收發簡訊 (參考第 21 章)。
demo_send_file	透過 Ethernet 或 3G 來傳送檔案到一台遠方的 PC (參考第 20 章)。
demo_3G	3G 無線通訊。

12.2 計時器 (Timer) 操作

12.2.1 啟動、停止、重置計時

請參考 [P12-1](#) 來開啟此專案 ("demo_tmr1.zip")，於變數區可查看/建立變數。

ST 程式:

```
(* 宣告 "START_tmr", "STOP_tmr", "RESET_tmr", "LED1", "LED2" 為 BOOL  
宣告 "TMR1" 為 TIME *)
```

```
(* 設定 "START_tmr" 為 TRUE, 以開始 "TMR1" 計時 *)
```

```
IF START_tmr THEN  
  START_tmr := FALSE;  
  TSTART (TMR1);  
END_IF;
```

```
(* 設定 "STOP_tmr" 為 TRUE, 以停止 "TMR1" 計時 *)
```

```
IF STOP_tmr THEN  
  STOP_tmr := FALSE;  
  TSTOP (TMR1);  
END_IF;
```

```
(* 設定 "RESET_tmr" 為 TRUE, 以重置 "TMR1" 為 T#0s *)
```

```
IF RESET_tmr THEN  
  RESET_tmr := FALSE;  
  TMR1 := T#0s;  
END_IF;
```

```
(* 當 "TMR1" 為 T#3s ~ T#10s 時, "LED1 ~ LED2" 為 ON *)
```

```
LED1 := FALSE;  
LED2 := FALSE;  
IF (TMR1 >= T#3s) and (TMR1 <= T#10s) THEN  
  LED1 := TRUE;  
  LED2 := TRUE;  
END_IF;
```

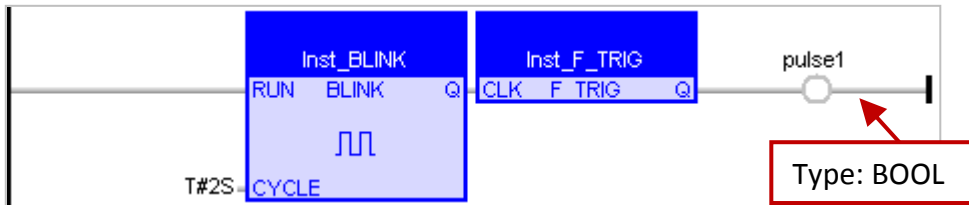
```
(* 當 "TMR1" 為 T#15s 時, 自動重置 "TMR1" 為 T#0s *)
```

```
IF TMR1 >= T#15s THEN  
  TMR1 := T#0s;  
END_IF;
```

12.2.2 週期性的操作

請參考 [P12-1](#) 來開啟此專案 ("demo_tmr2.zip")，於變數區可查看/建立變數。"BLINK" 功能搭配 "F_TRIG" 功能方塊，可每隔一段時間產生一個 Pluse TRUE，因此它可以應用在週期性的操作上。

LD 程式:



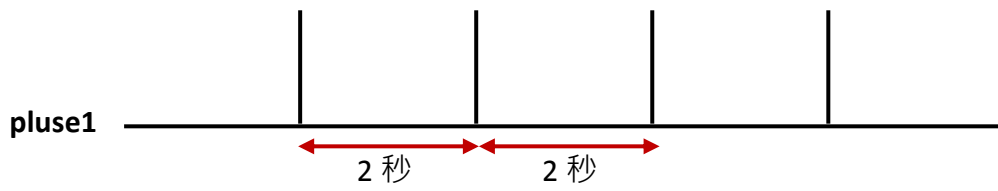
ST 程式:

```

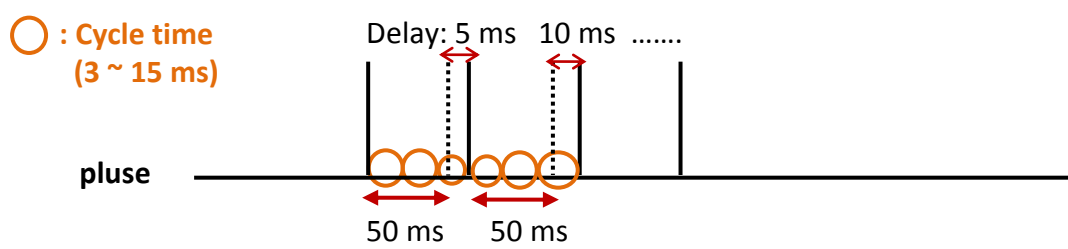
IF pluse1 THEN
  (* 執行操作 *)
  ...
END_IF;

```

上方的 "BLINK" 與 "F_TRIG" 將會每 2 秒產生一個 Pluse TRUE，但是此方式有個缺點:



若週期的間隔時間較短時 (例如: 間隔時間為 100 ms 或更小；或是 PAC 的 Cycle time 較大時，像是 20~50 ms，一般為 3~15 ms)，則此週期性的操作會不準確。例如，每 50 ms 執行一次週期性操作，由於相較於 250 ms 或 2 秒，間隔時間為 50 ms 相當接近 PAC 的 Cycle time，若使用功能方塊的方式很容易累積延遲輸出的時間，因此最終的操作時間也變得不準確。



為了改善此狀況，使用以下的編寫方式會較精確:

請參考 [P12-1](#) 來開啟此專案 ("demo_tmr3.zip")，於變數區可查看/建立變數。

ST 程式:

(* 宣告 "INIT" 為 BOOL 且設定初值為 TRUE
宣告 "TMR1", "TMR1_next" 為 TIME *)

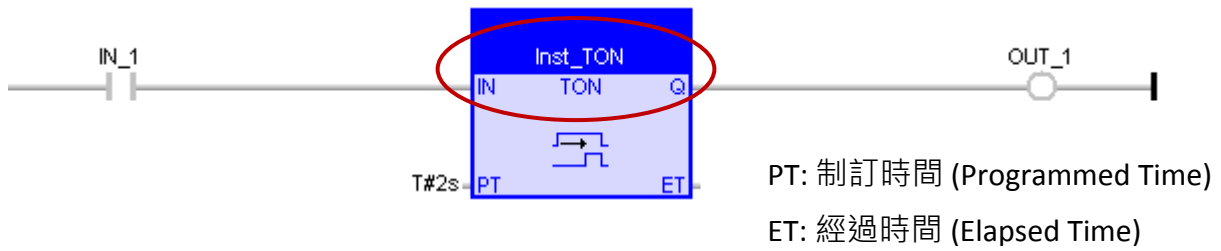
```
IF INIT THEN
  INIT := FALSE ;
  TMR1 := T#0s ;
  TMR1_next := TMR1 + T#50 ms ;
  TSTART (TMR1);
END_IF;

IF TMR1 >= TMR1_next THEN
  IF TMR1 > T#10h THEN
    TMR1 := T#0s ;
    TMR1_next := T#0s ;
  END_IF;
  TMR1_next := TMR1_next + T#50 ms ;
  (* 執行操作 *);
  ...
END_IF;
```

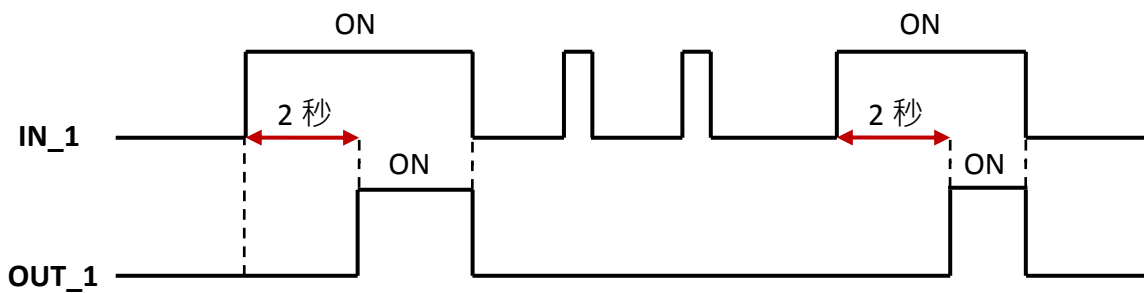
當計時到 T#23h59m59s999ms 時，
Timer 的值會產生溢位 (Overflow)。
因此，可設定它在 10 或 18 小時，
自動重置為 "0"。

12.2.3 偵測穩定的 ON 或 OFF 訊號

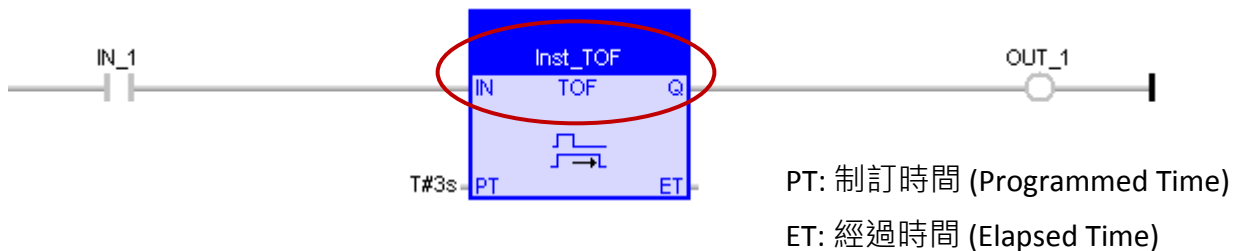
"TON" 功能方塊可偵測經過一段時間，仍維持 "ON" 的穩定訊號。



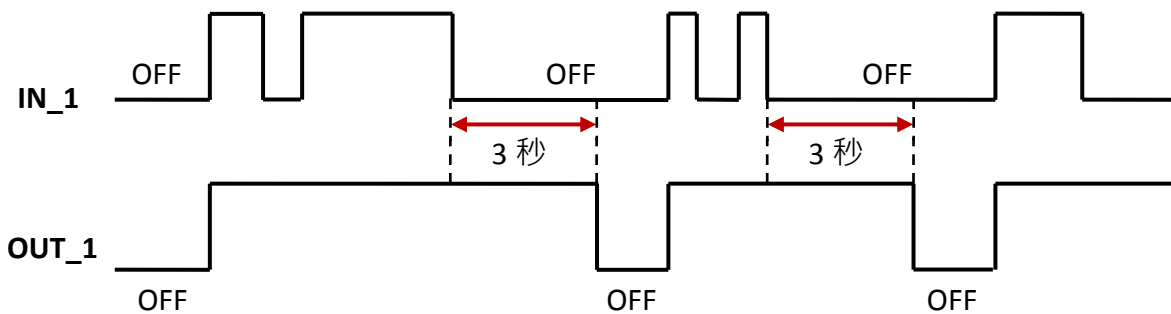
如上圖，此功能可偵測維持 2 秒仍 "On" 的穩定訊號。



"TOF" 功能方塊可偵測經過一段時間，仍維持 "OFF" 的穩定訊號。

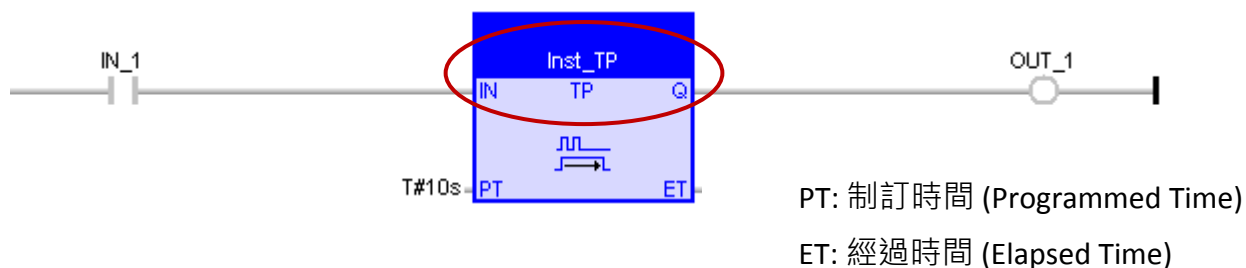


如上圖，此功能可偵測維持 3 秒仍 "OFF" 的穩定訊號。

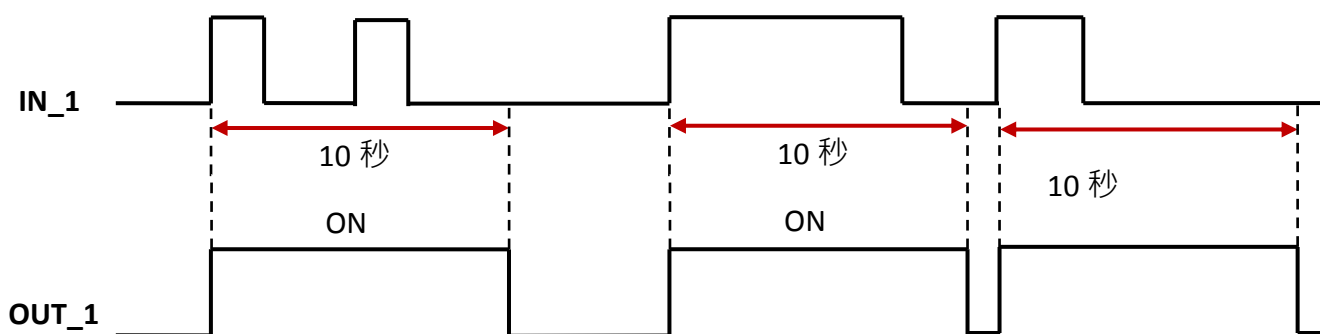


12.2.4 觸發後維持 "ON" 一段時間

"TP" 功能方塊可在觸發後 (即，由 OFF 變成 ON 時) 維持輸出 "ON" 一段時間。



如上圖，觸發後維持 10 秒 "ON" 的訊號輸出。

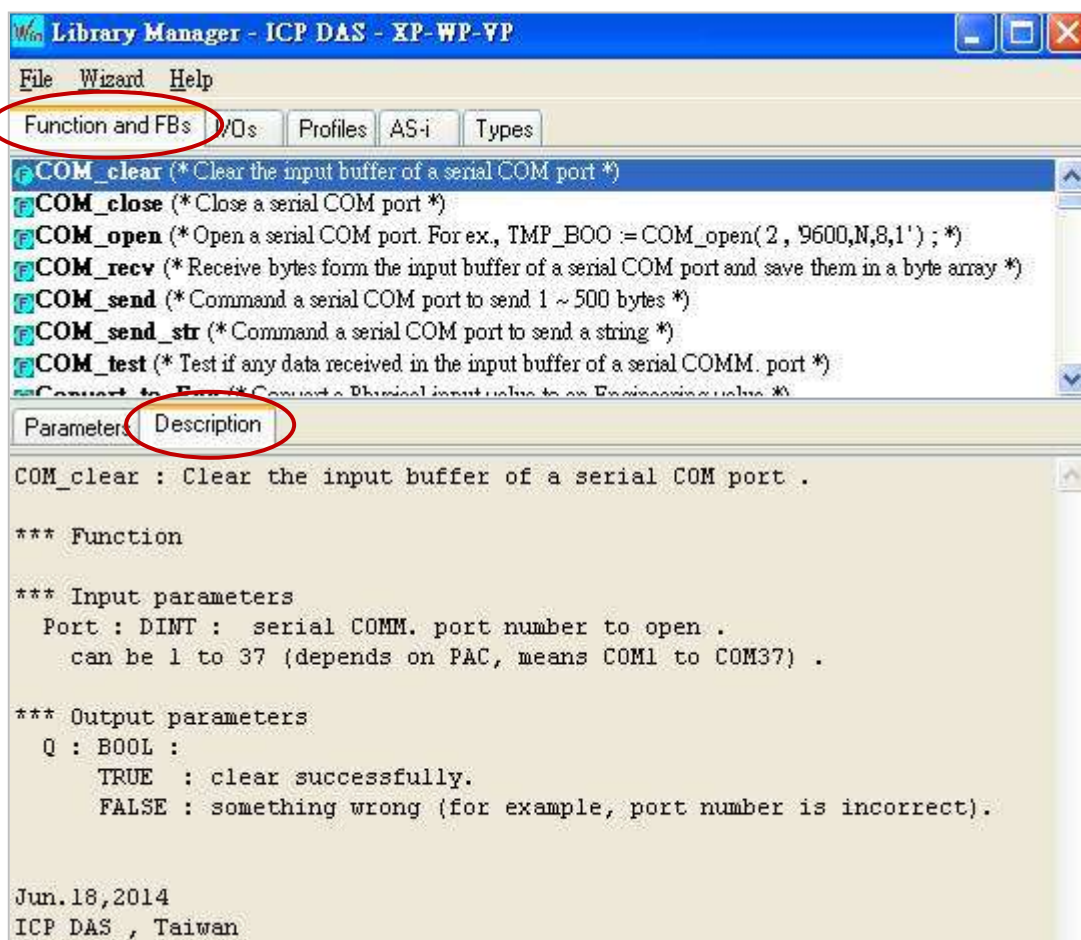


12.3 序列埠的通訊操作

使用者可直接操作序列埠 (即，RS-232, RS-485 或 RS-422 Port) 來實現一些特定的通訊協定。以下有 7 個函式 (Function) 可用來直接操作序列埠。

函式 (Function)	說明
COM_open	開啟一個序列埠。
COM_close	關閉一個序列埠。
COM_clear	清除序列埠的輸入暫存區。
COM_test	測試序列埠是否有接收到任何 Byte?
COM_send	傳送一些 Byte 給序列埠。
COM_send_str	傳送一些 String 給序列埠。
COM_recv	接收一些 Byte 給序列埠。
COM_status	取得 COM Port 是否已正常開啟的狀態。

可參考 [1.2.3 節](#) 來開啟程式庫管理員 (Library Manager) 並查詢各函式 (Function) 的使用說明。



12.3.1 使用 COM Port 來傳送一個字串

請參考 [P12-1](#) 來開啟此專案 ("demo_com_port1.zip")，於變數區可查看/建立變數。

ST 程式: 此程式可以每 2 秒從 PAC 的 COM1 (通訊參數: '9600, N, 8, 1') 送出一個字串 (例如:

< CNT1 = 1 > 或 < CNT1 = 25 >, 等)。

(* 在第一個 PAC cycle 執行操作 *)

if INIT then

INIT := FALSE; (* 非第一個 cycle *)

CNT1 := 0;

TMR1 := T#0s;

TMR1_next := TMR1 + T#2s;

(* 開始 "TMR1" 計時 *)

tStart(TMR1);

end_if;

(* 若 COM port 的狀態變為 FALSE (未開啟), 則開啟它 *)

if COM_Status(Port_number) = FALSE then

(* 開啟一個序列 COM port *)

Port_OK := COM_open(Port_number, '9600,N,8,1');

end_if;

(* 到達時間時, ... *)

if TMR1 >= TMR1_next then

(* 避免 "TMR1" 達到 T#23h59m59s999ms 發生溢位 *)

if TMR1 > T#10h then

TMR1 := T#0s;

TMR1_next := T#0s;

end_if;

(* 為 "TMR1_next" 設定新值 *)

TMR1_next := TMR1_next + T#2s;

(* 由 COM port 傳送字串 *)

COM_send_str(Port_number, '<CNT1=' + Any_to_STRING(CNT1) + '>');

(* 到達 100 時, 重置 CNT1 為 0 *)

CNT1 := CNT1 + 1;

if CNT1 >= 100 then

CNT1 := 0;

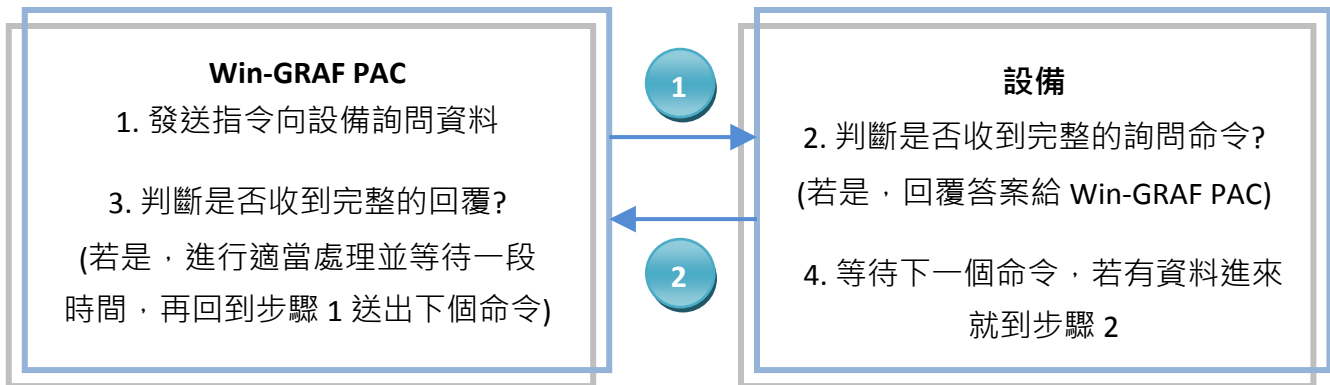
end_if;

end_if;

宣告 "INIT" 為 BOOL 且初值為 TRUE ;
"Port_OK" 為 BOOL ;
"CNT1" 為 DINT ;
"TMR1", "TMR1_next" 為 TIME
"Port_number" 為 DINT 且初值為 "1"。

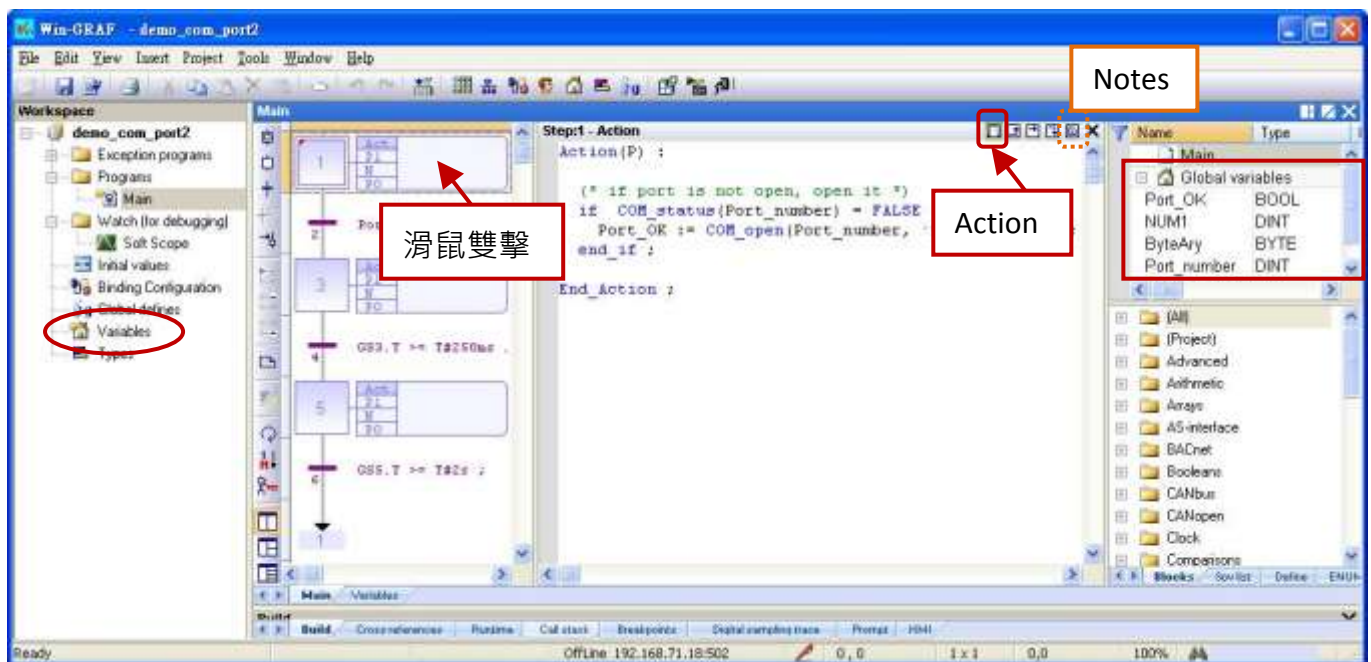
12.3.2 使用 COM Port 對設備一問一答

如有應用需使用 RS-232/485/422 Port 來取得其它設備的資料，其一問一答的方式如下：

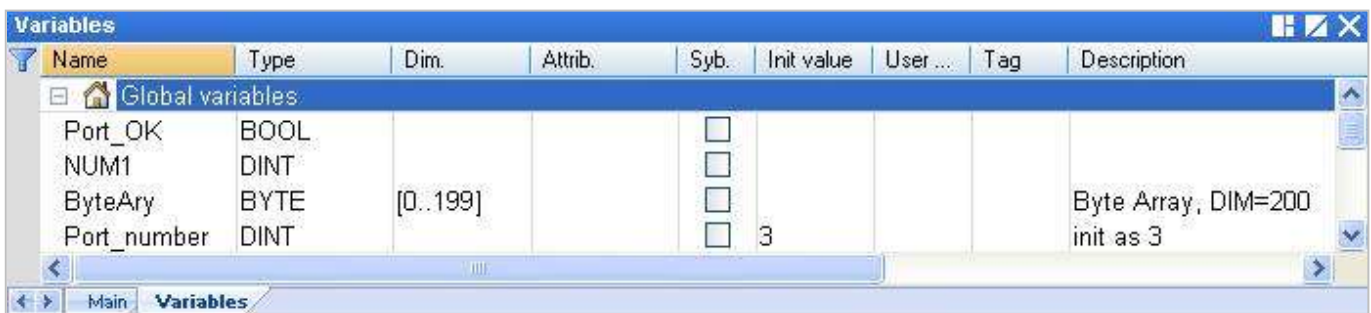


請參考 [P12-1](#) 來開啟此專案 ("demo_com_port2.zip")，於變數區可查看/建立變數。

註：滑鼠雙擊“Action”會先開啟“Notes”視窗，請切換到“Action”視窗來查看程式碼。



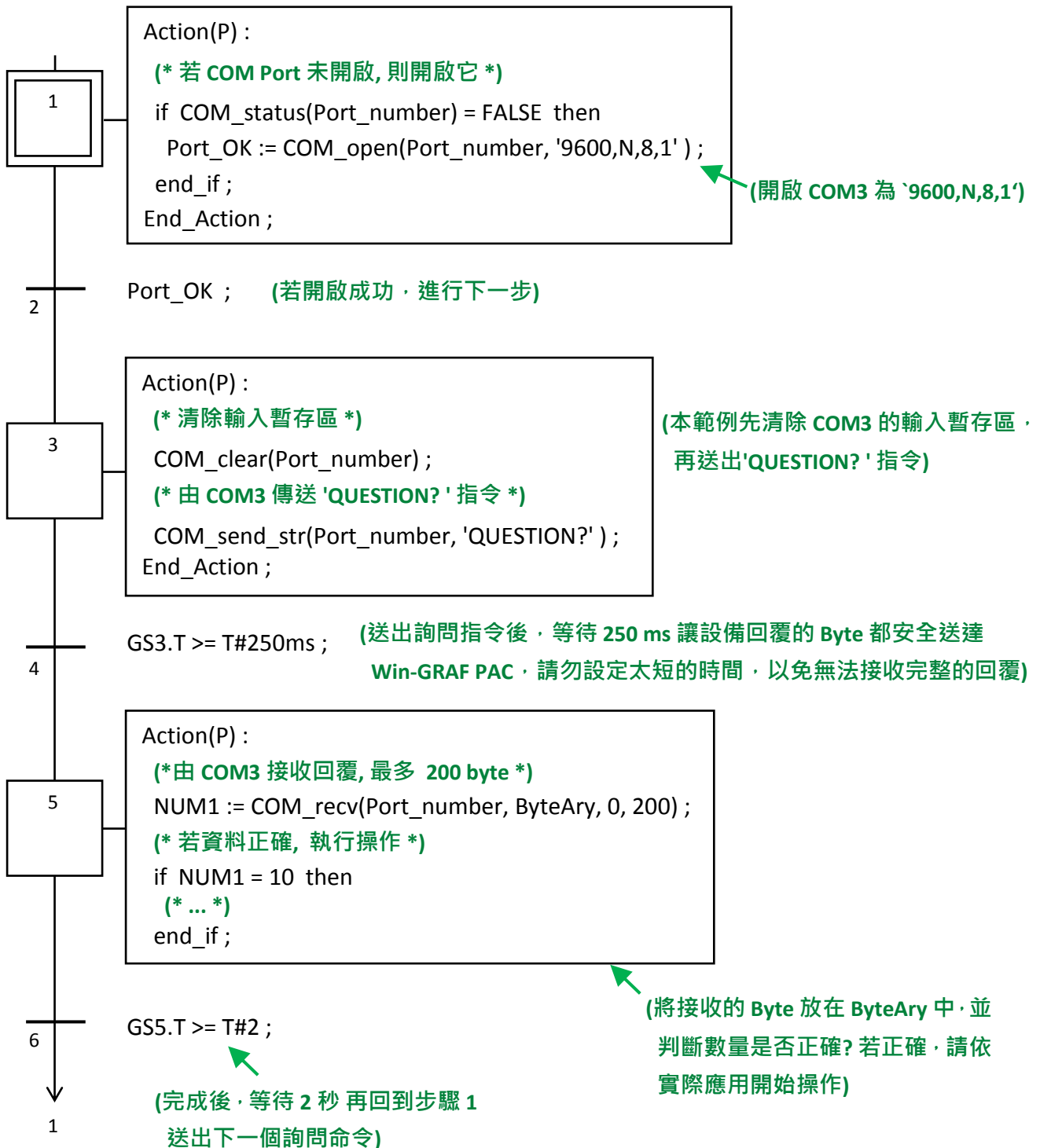
您也可點選“Variables”開啟變數視窗。



本範例先由 Win-GRAF PAC 的 COM3 送出一個字串 'QUESTION?' 給設備，接著等待設備的回覆並做處理，完成後等待 2 秒，再送出同樣的 'QUESTION?' 命令，...如此重複進行。

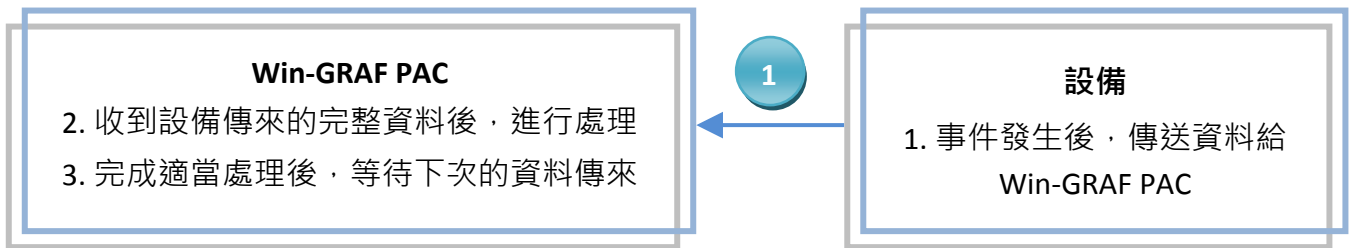
SFC 程式:

(變數 "Port_OK" 為 BOOL; "NUM1" 為 DINT; "ByteAry" 為 BYTE 且 Dim. 為 "200";
 "Port_number" 為 DINT 且初值 為 "3"。)



12.3.3 COM Port 等待遠端設備傳來的資料

此種方式在一般的商場或便利商店很常見，像是使用條碼閱讀機。當讀取到商品的條碼後，它會傳送條碼資料到 Win-GRAF PAC 的 COM Port (RS-232/485/422)，並且不需回覆任何訊息。



請參考 [P12-1](#) 來開啟此專案 ("demo_com_port3.zip")，於變數區可查看/建立變數。

ST 程式:

(* 在第一個 PAC cycle 執行操作 *)

```
if INIT then
  INIT := FALSE ;
  T1 := T#0s ;
  STEP1 := 0 ;
end_if ;
```

(* 若 COM Port 未開啟, 則開啟它 *)

```
if COM_status(Port_number) = FALSE then
  Port_OK := COM_open( Port_number , '9600,N,8,1' ) ;
end_if ;
```

(* 若開啟 COM Port 失敗，離開此 ST 程式 *)

```
if Port_OK = FALSE then
  return ;
end_if ;
```

CASE STEP1 OF

(* 是否有至少 1 Byte 的資料傳來 *)

```
0:
  if COM_test(Port_number) then
    STEP1 := 1 ;
    T1 := T#0s ;
    Tstart(T1) ;
  end_if ;
```

```
宣告 "INIT" 為 BOOL 且初值為 TRUE ;
"Port_OK" 為 BOOL ;
"STEP1", "NUM1" 為 DINT ;
"T1" 為 TIME ;
"ByteAry" 為 BYTE 且 Dim. 為 "200" ;
"Port_number" 為 DINT 且初值 為 "3" 。
```

```
STEP1 = 0，表示等待中並測試 COM3 是否有傳資料進來？
若回傳 TRUE，表示有資料。
將 STEP1 設定為 "1"，T1 設定為 "0" 並開始計時。
```

(* 等待 250 ms, 並由 COM port 接收所有的 Byte *)

1:

```
if T1 >= T#250ms then
  Tstop(T1);
  T1 := T#0s;
  STEP1 := 0;
```

STEP = 1，表示資料正傳送進來，等待 250 ms 將所有接收的資料放在陣列中，此等待時間和設備規格與 Baud Rate 有關，若設定的時間太短可能會接收不完全。記得將 STEP1 設定為 "0" 來等待下次資料再傳進來。

(* 最多可接收 200 Byte *)

```
NUM1 := COM_recv(Port_number, ByteArray, 0, 200);
```

(* 若資料是正確的, 此例為 25 Byte 則執行操作 *)

```
if NUM1 = 25 then
  (* ... *)
end_if;
```

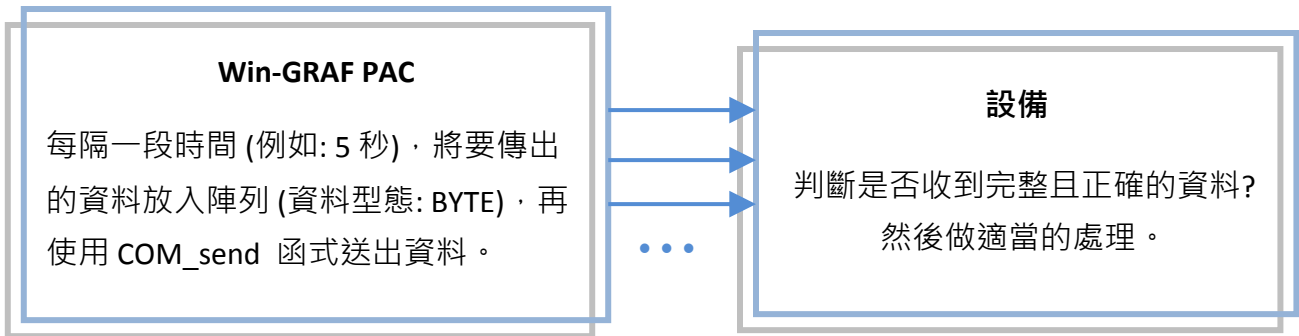
收到資料後，判斷資料對不對？再依實際應用開始操作。

```
end_if;
```

```
END_CASE;
```

12.3.4 使用 COM Port 定期回報資料給遠端設備

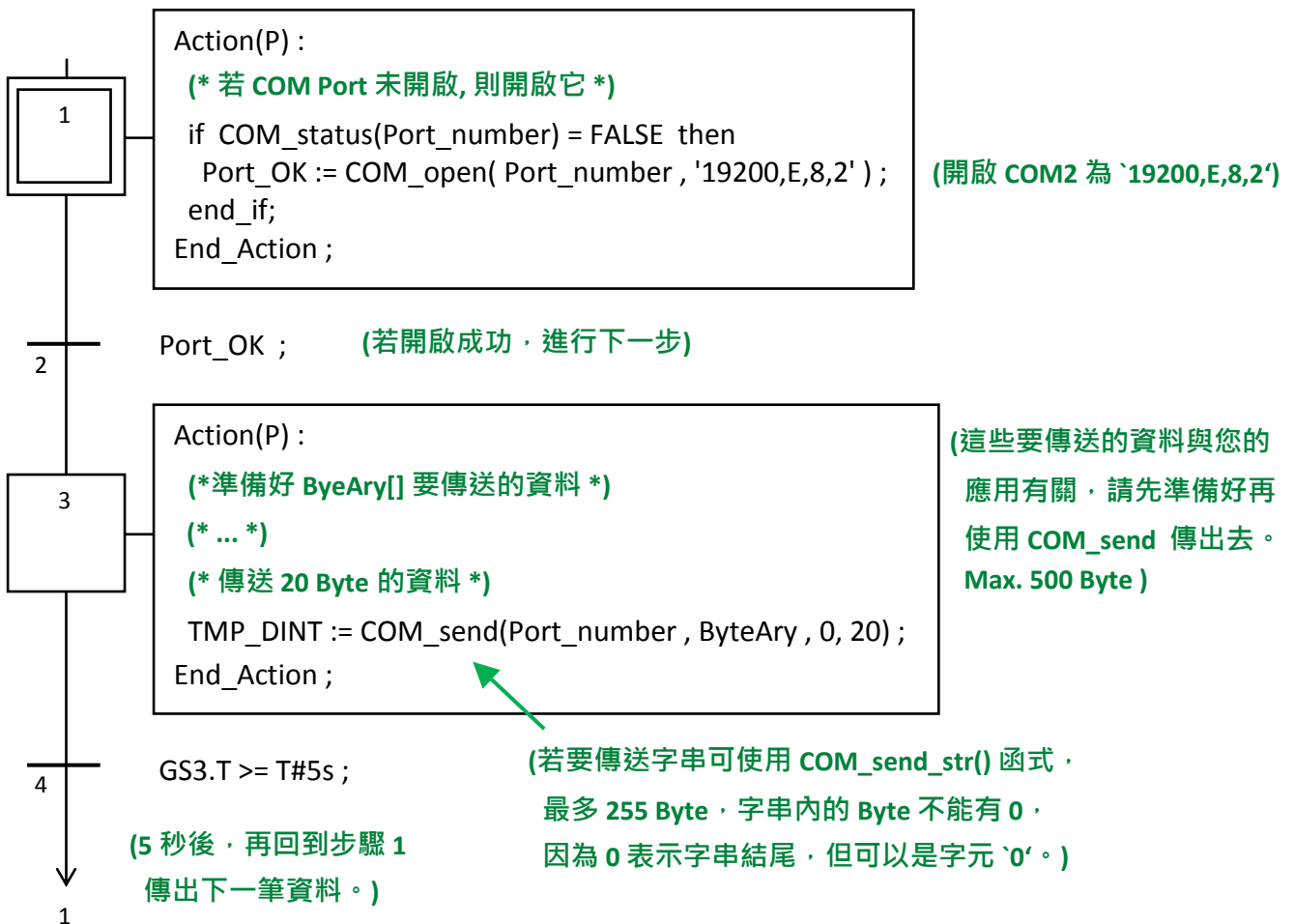
如需每隔一段時間使用 RS-232/485/422 Port 將資料回報給其它的設備，方式如下：



請參考 [P12-1](#) 來開啟此專案 ("demo_com_port4.zip")，於變數區可查看/建立變數。

SFC 程式: (可參考 [12.3.2 節](#) 來開啟 "Action" 視窗。)

(宣告變數: "Port_OK" 為 BOOL ; "TMP_DINT" 為 DINT ; "ByteAry" 為 BYTE 且 Dim. 為 100 ; "Port_number" 為 DINT 且初值為 "2"。)



12.4 讀/寫 PAC 內儲存裝置的檔案 (File)

Win-GRAF Workbench 提供了下列函式 (Function) ，可對 PAC 內的儲存裝置進行讀/寫操作。

函式名稱	說明
<p>請點選功能表 "Help" > "Topics" 並輸入搜尋字串 "File" ，於 "File Management functions" 主題中來了解詳細的使用說明。</p> 	
F_ROPEN	開啟/建立一個檔案以供讀取
F_WOPEN	開啟/建立一個檔案以供寫入 若檔案不存在，會自動建立；若檔案已存在，會先清空內容
F_AOPEN	以附加方式來建立/開啟檔案
F_CLOSE	關閉一個開啟中的檔案
F_EOF	測試若到達檔案尾端時，開啟檔案以供讀取
FA_READ	從二進位 (Binary) 檔案中，讀出整數值 (DINT)
FA_WRITE	從二進位 (Binary) 檔案中，寫入整數值 (DINT)
FM_READ	從文字檔中，讀出字串 (STRING)
FM_WRITE	從文字檔中，寫入字串 (STRING)
FB_READ	從檔案中，讀出二進位值 (Binary)
FB_WRITE	從檔案中，寫入二進位值 (Binary)
F_EXIST	測試檔案是否存在？
F_GETSIZE	取得檔案大小
F_COPY	複製檔案
F_DELETE	刪除檔案
F_RENAME	將檔案重新命名
請參考 1.2.3 節 來查詢以下 Function (函式) 的使用說明	
F_dir	建立一個目錄
F_cp_dir	複製一個目錄與其內的檔案到另一個目錄 (包含子目錄， 註 1)
F_del_dir	刪除一個目錄與其內的檔案 (包含子目錄， 註 1)

註: ICP DAS 的 Win-GRAF PAC 並不支援 "F_SAVERETAIN" 與 "F_LOADERRETAIN" 函式。

註 1: F_cp_dir · F_del_dir 從以下 PAC Driver 版本起才有包含子目錄。

WP-8xx8 : 1.04 · VP-x2x8-CE7 : 1.01 · XP-8xx8-CE6 : 1.02 · WP-5xx8-CE7 : 發行日起。

12.4.1 寫入資料到 PAC 內的檔案

請參考 [P12-1](#) 來開啟此專案 ("demo_file1.zip")，於變數區可查看/建立變數。

ST 程式: 此程式可用來寫入 10 個 "REAL" 數值到 PAC 內的檔案。

(* 此 "demo_file1" 專案會把 10 個 REAL 數值儲存到 \System_Disk\Real_data1.txt

檔案格式：

每一列有一個 REAL 數值且包含了換行字元 (即，<CR><LF>)。

```
1.08
2.786
38.45
41.5
59.875
60.76
71.23
80.5
99.8
100.7 *)
```

(* 變數宣告:

```
Write_File      : BOOL
Tmp_string      : String, len=255
File_ID         : DINT
REAL_val[0..9] : REAL
ii              : DINT
File_Status     : String, len=128 *)
```

由於 \System_Disk\ 的容量較小，建議您將檔案路徑改為以下位置:

WinPAC, ViewPAC 系列:

\Micro_SD\ 或是

XPAC 系列:

\System_Disk2\

(* 將 "Write_File" 設定為 "TRUE"，將資料寫入檔案中 *)

```
if Write_File then
```

```
Write_File := FALSE ;
```

```
File_ID := F_Wopen( '\System_Disk\Real_data1.txt' );
```

```
if File_ID = 0 then
```

(* 在寫入模式將無法開啟檔案 *)

```
File_Status := 'Can not open file in write mode !' ;
```

```
else
```

(* 若開啟檔案成功, 將 REAL[0] ~ [9] 的值儲存到檔案中, 每列包含 1 個 REAL 值與

結束字元 <CR><LF> *)

```
File_Status := 'Open file ok.' ;
```

```
for ii := 0 to 9 by 1 do
```

```

    Tmp_string := Any_to_string( REAL_val[iii] );
    FM_write( File_ID , Tmp_string );
end_for ;

```

(* 關閉檔案 *)

```

F_close( File_ID );

```

```

end_if ;
end_if ;

```

若想存成整數資料，可修改為：

```

Tmp_string := Any_to_string( DINT_val[iii] );

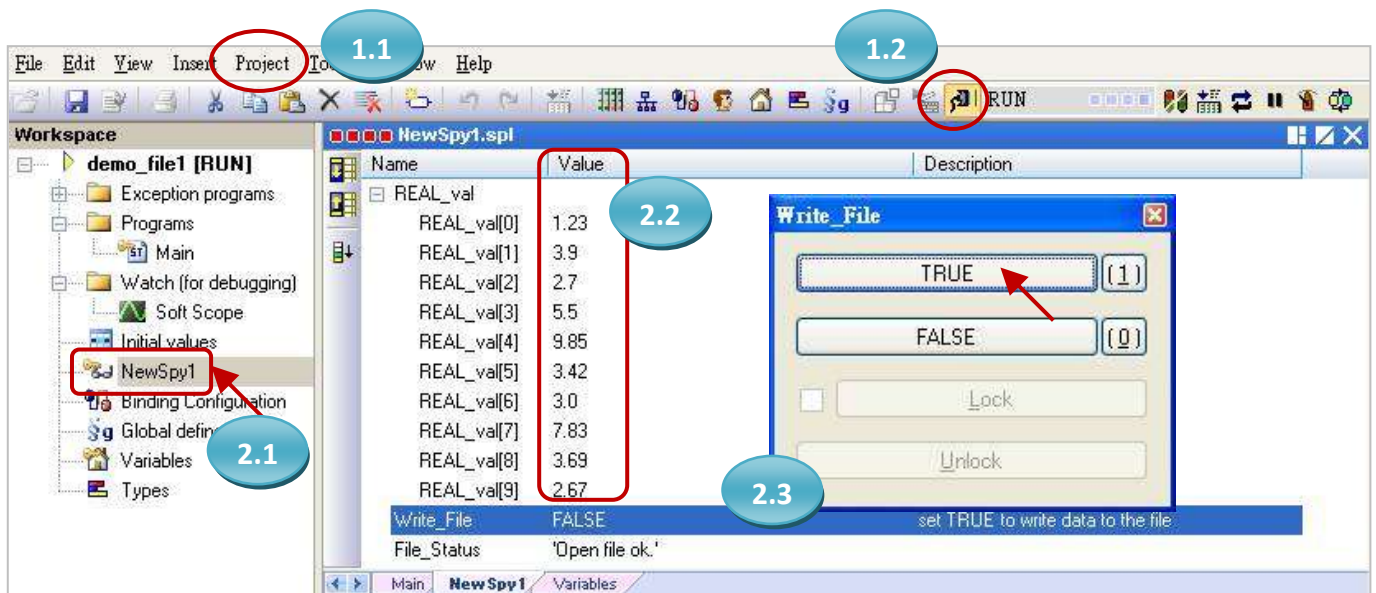
```

且變數區也需宣告 "DINT_val" 為 DINT 且 Dim. 至少為 "10"。

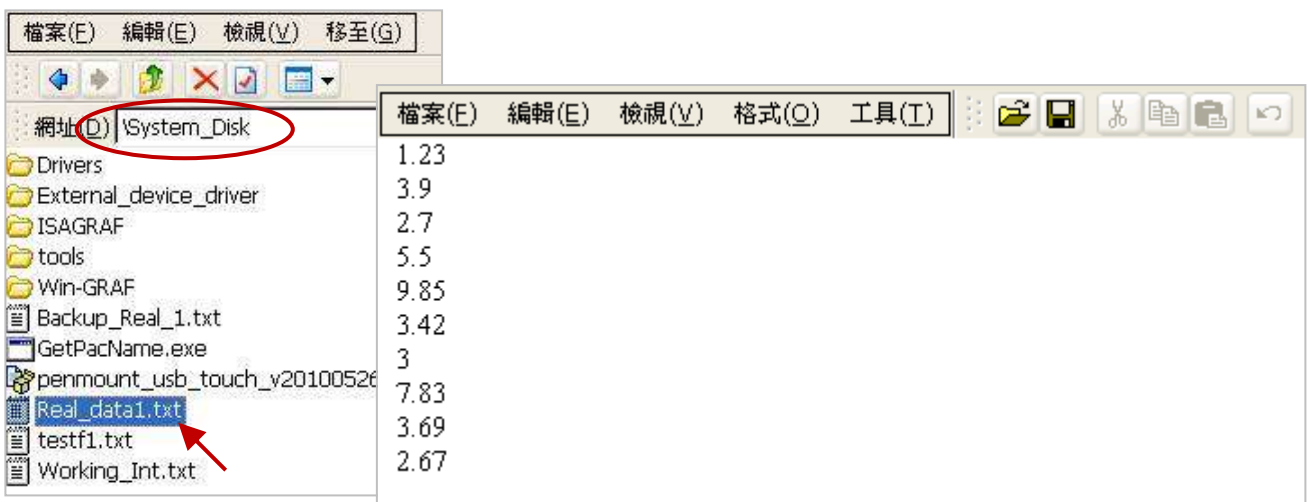
測試程式：

此例當 "Write_File" 設為 "TRUE" 時，會將數值寫入 PAC 內的 \System_Disk\Real_data1.txt 檔案中。

1. 請先完成 IP 設定 (見 [P12-1](#))，編譯並下載程式到 PAC 中。
(點選 "Project" > "Build All Projects" / "On Line"，若不熟悉操作，可參考 [2.3.4 節](#)、[2.3.5 節](#))
2. 點選 "NewSpy1" 開啟觀測清單，並填入要寫入的數值，再將 "Write_File" 設定為 "TRUE"，來寫入資料。(若 OK，"File_Status" 會顯示 "Open file ok")



3. PAC 中，開啟 "Real_data1.txt" 檔案，可見到步驟 2 所填入的值。



12.4.2 讀取 PAC 內的檔案資料

請參考 [P12-1](#) 來開啟此專案 ("demo_file2.zip")，於變數區可查看/建立變數。

ST 程式: 此程式可用來讀取 PAC 內檔案中的 10 個 "REAL" 數值。

(* 此 "demo_file2" 專案會從 \System_Disk\Real_data2.txt 檔案中讀取 10 個 REAL 數值

檔案格式:

每一列有一個 REAL 數值且包含了換行字元 (即, <CR><LF>)。

```
1.08
2.786
38.45
41.5
59.875
60.76
71.23
80.5
99.8
100.7
```

*)

(*

變數宣告:

```
Write_File      : BOOL
Tmp_string      : String, len=255
File_ID         : DINT
REAL_val[0..9] : REAL
ii              : DINT
File_path       : String, len = 128, initial val = '\System_Disk\Real_data2.txt'
File_Status     : String, len=128
```

*)

(* 將 "Read_File" 設定為 "TRUE"，來讀取檔案資料 *)

```
if Read_File then
```

```
  Read_File := FALSE ;
```

```
  (* 檢查檔案是否存在? *)
```

```
  if F_exist( File_path ) = FALSE then
```

```
    (*檔案不存在 *)
```

```
    File_Status := 'File "' + File_path +'" does not exist !' ;
```

```
  else
```

```
    (* 若檔案存在, 開啟並讀取檔案 *)
```

```
    File_ID := F_Ropen( File_path );
```

```
  if File_ID = 0 then
```

```
    (* 開啟檔案失敗 *)
```

```
    File_status := 'Can not open File "' + File_path +'" !' ;
```

由於 \System_Disk\ 的容量較小，建議您將檔案路徑改為以下位置:

WinPAC, ViewPAC 系列:

\Micro_SD\ 或是

XPAC 系列:

\System_Disk2\

(* 開啟檔案成功, 讀取檔案中 REAL[0] ~ [9] 的數值,
每列包含 1 個 REAL 值與結束字元 <CR><LF> *)

```
File_status := 'Open File "' + File_path + "' Ok .';  
for ii := 0 to 9 by 1 do
```

(* 測試是否讀到檔案的最後一列? *)

```
if F_EOF( File_ID ) then
```

(* 讀到檔案的最後一列, 離開 "for" 迴圈 *)

```
exit ;  
end_if ;
```

(* 讀取檔案中的一列 String 資料 *)

```
Tmp_string := FM_READ( File_ID );
```

(* 將 String 轉換為 REAL 數值 *)

```
REAL_val[ii] := Any_to_REAL( Tmp_string );
```

```
end_for ;
```

(* 關閉檔案 *)

```
F_close( File_ID );
```

```
end_if ;
```

```
end_if ;
```

```
end_if ;
```

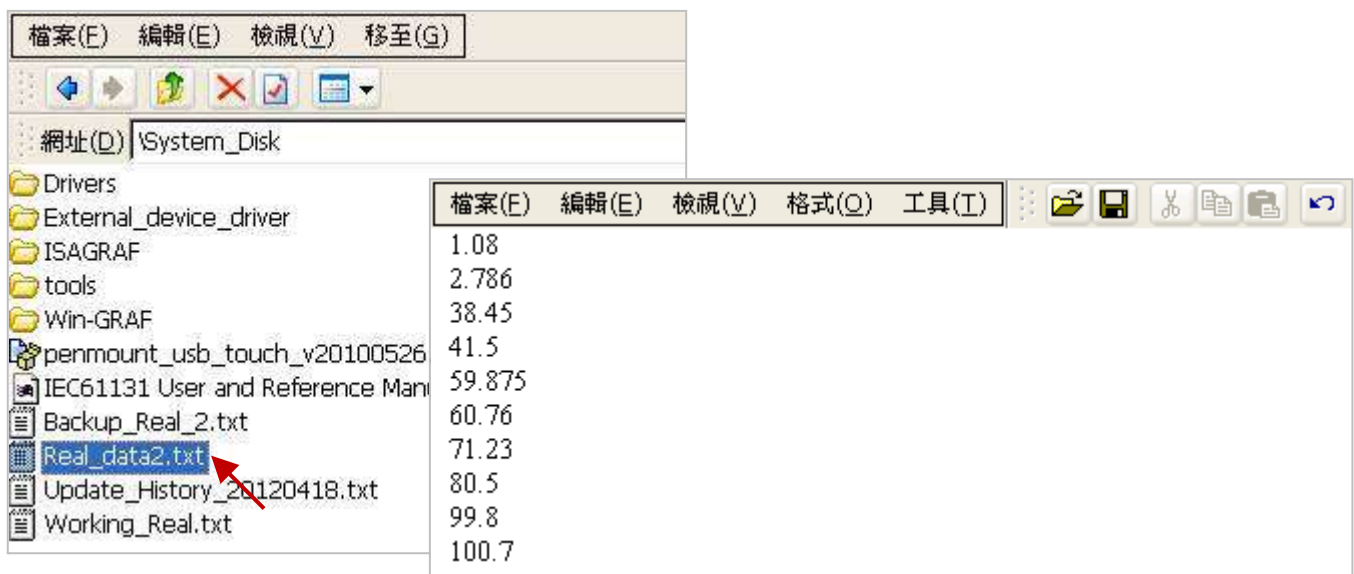
若想讀取整數資料, 可修改為:

```
DINT_val[ii] := Any_to_DINT( Tmp_string );
```

且變數區也需宣告 "DINT_val" 為 DINT 且

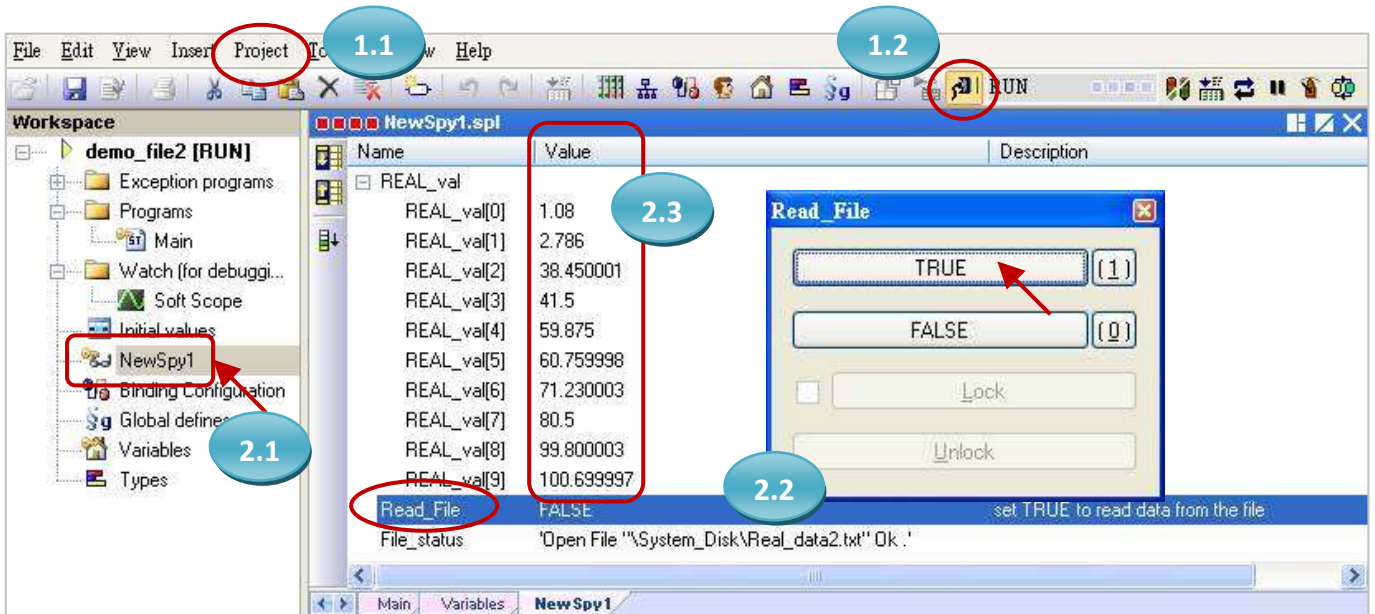
Dim. 至少為 "10" 。

註: 此範例當 "Read_File" 設定為 "TRUE" 時, 會去讀取 PAC 內 "\System_Disk\Real_data2.txt" 檔案, 請確認此檔案已存放在 PAC 中。



測試程式:

1. 請先完成 IP 設定 (見 [P12-1](#))，編譯並下載程式到 PAC 中。
(點選 "Project" > "Build All Projects" / "On Line"，若不熟悉操作，可參考 [2.3.4 節](#)、[2.3.5 節](#))
2. 點選 "NewSpy1" 開啟觀測清單，再將 "Read_File" 設定為 "TRUE"，來讀取資料。
(若 OK，"File_Status" 會顯示 "Open File "\System_Disk\Real_data2.txt" Ok.")



註: 在 [6.2 節](#) 還有另外一個與操作 File 有關的範例程式說明，它可以處理 File 內很多資料，您可以參考它。

12.4.3 資料紀錄功能

請參考 [P12-1](#) 來開啟此專案 ("demo_DataLog.zip")，此程式為簡易的資料記錄功能 (Data Logging)。本程式會建立一個含有字串 (write_date)、整數 (int_data) 與 實數 (float_data) 變數的觀測清單 (Spy List)，且每隔 1 分鐘會將此 3 個變數，紀錄在 PAC 的 "\\System_Disk2\" 下，該記錄檔會依照日期來命名 (例如: 2015 年 5 月 6 日，檔名就是 "2015-5-6.csv")，另外，每一天會更換紀錄檔，並將前一個檔案放到 "\\System_Disk2\\當月日期\" 的資料夾底下 (例如: 2015 年 5 月 6 日，會放在 "\\System_Disk2\\2015-5\\")。

註: 本範例是以 XP-8xx8-CE6 為例，若您使用了其它 PAC 型號，請修改程式中的存檔路徑為 \\Micro_SD\ 或其它路徑。

CSV 檔案格式範例如下:

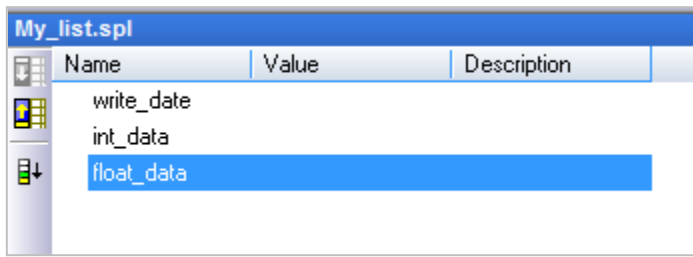
```
Time , int_val , float_val
20:18:30, 1236, 14.56
20:18:40, 3456, 34.56
20:18:50, 8932, 89.32
```

變數說明: 您可在變數區 (Variables) 查看/建立變數。

變數名稱	資料型態	說明
Year1	DINT	於 "PAC_Time" 程式中，用來取得 PAC 的系統時間。
Month1	DINT	
Day1	DINT	
WeekDay1	DINT	
Hour1	DINT	
Minute1	DINT	
Second1	DINT	
old_day	DINT	於 "PAC_Time" 程式中，用來時間變更時切換紀錄檔的名稱。
old_month	DINT	
log_tmr1	TIME	Timer。
log_tmr2	TIME	
CSV_Path	STRING	CSV 檔的儲存路徑。
CSV_Dir	STRING	CSV 檔的儲存目錄。
write_date	STRING	用於紀錄每筆資料寫入 CSV 檔的時間。
init	BOOL	用於初始化，預設為 TRUE。
int_data	DINT	用於紀錄資料。
float_data	REAL	用於紀錄資料。
writcsv	BOOL	用於觸發寫入資料。
File_ID	DINT	用於檔案開啟函數。
tmp_bval	BOOL	暫存變數。
tmp_msg	STRING	

觀測清單 (Spy List) – “My_List”:

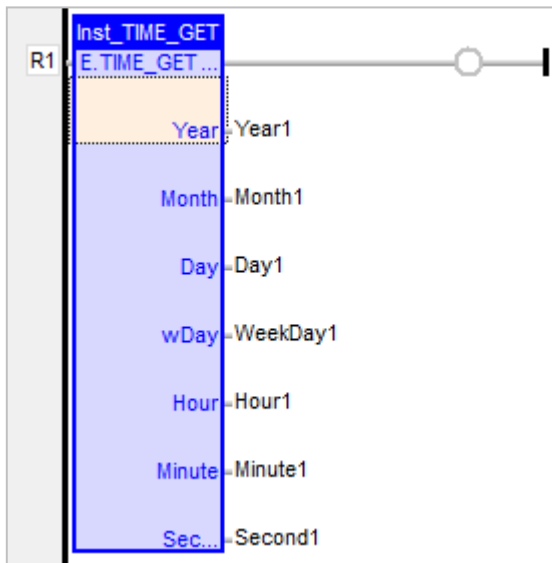
此範例建立了一個名為 “My_list” 的觀測清單 (請參考 [11.3 節](#))。



Name	Value	Description
write_date		
int_data		
float_data		

LD 程式 – “PAC_Time”:

用來取得/設定 PAC 的系統時間。



ST 程式 – “Main”:

用來產生隨機變數作為紀錄檔 (.CSV) 的資料與 變更檔案路徑。

(* 宣告 “init” 為 BOOL，宣告 “log_tmr1” 以及 “log_tmr2” 為 TIME *)

(* 設定 “init” 為 TRUE, 以開始 “TMR1” 計時 *)

```
IF init THEN
```

```
  init := FALSE;
```

```
  (*啟用 Data Logger 的計時器*)
```

```
  TSTART (log_tmr1);
```

```
  (* 設定 .csv 紀錄檔內的分隔符號 與小數點 *)
```

```
  SetCsvOpt( ', ', '. ' );
```

```
END_IF;
```

```
IF old_day<> day1 THEN
```

```
  tmp_msg:='\System_Disk2\'+'CSV_Dir+'\'+ANY_TO_STRING(year1)+'-'+'+
```

```
  ANY_TO_STRING(month1)+'-'+'+ANY_TO_STRING(old_day)+''.csv';
```


(* 複製檔案到 System_Disk2 *)

```
tmp_bval:=F_COPY (CSV_Path,tmp_msg);
```

(* 刪除原本在 Temp 的檔案 *)

```
tmp_bval:=F_DELETE(CSV_Path);
```

(* 建立 CSV 檔案與其欄位名稱 *)

```
tmp_msg:='\Temp\'+any_to_string(year1)+'-'+any_to_string(month1)+'-'+  
any_to_string(day1)+'.csv';
```

```
CSV_Path:=tmp_msg;
```

```
File_ID:= F_WOPEN(tmp_msg);
```

```
IF File_ID<>0 THEN
```

```
tmp_msg:='Time,Int_val,Float_val';
```

```
tmp_bval:=FM_WRITE(File_ID,tmp_msg);
```

```
tmp_bval:=F_CLOSE(File_ID);
```

```
END_IF;
```

```
old_day:=day1;
```

```
END_IF;
```

(*每過一個月 或是 PAC 剛啟動時會建立一個新資料夾*)

```
if old_month<>month1 then
```

```
CSV_Dir:=ANY_TO_STRING(year1)+ANY_TO_STRING(month1);
```

```
tmp_msg:='\System_Disk2\' + CSV_Dir;
```

```
F_DIR(tmp_msg);
```

```
old_month:=month1;
```

```
end_if;
```

由於 \System_Disk\ 的容量較小，建議將檔案存放在 - WinPAC, ViewPAC series: \Micro_SD\ 或 XPAC series: \System_Disk2\

(* 觸發寫入檔案功能 *)

```
IF log_tmr1>=log_tmr2 THEN
```

```
log_tmr1:=t#0s;
```

(* 產生隨機的 "Int_val" 與 "Float_val" 資料值 *)

```
int_data:=rand(1000);
```

```
float_data:=ANY_TO_REAL(int_data)/100;
```

(* "Time" 資料值 *)

```
write_date:=ANY_TO_STRING(hour1)+'-'+ANY_TO_STRING (Minute1)+'-'+  
ANY_TO_STRING (Second1);
```

(* 刪除原本的檔案 *)

```
tmp_msg:='\System_Disk2\' + CSV_Dir + '\' + ANY_TO_STRING(Year1) + '-' +  
ANY_TO_STRING(month1) + '-' + ANY_TO_STRING(Day1) + '.csv';
```

(* 刪除在 System_Disk2 的檔案 *)

```
tmp_bval:=F_DELETE(tmp_msg);
```

(* 觸發檔案寫入 *)

```
writcsv:=true;
```

```
END_IF;
```

LD 程式 – “WriteFile”

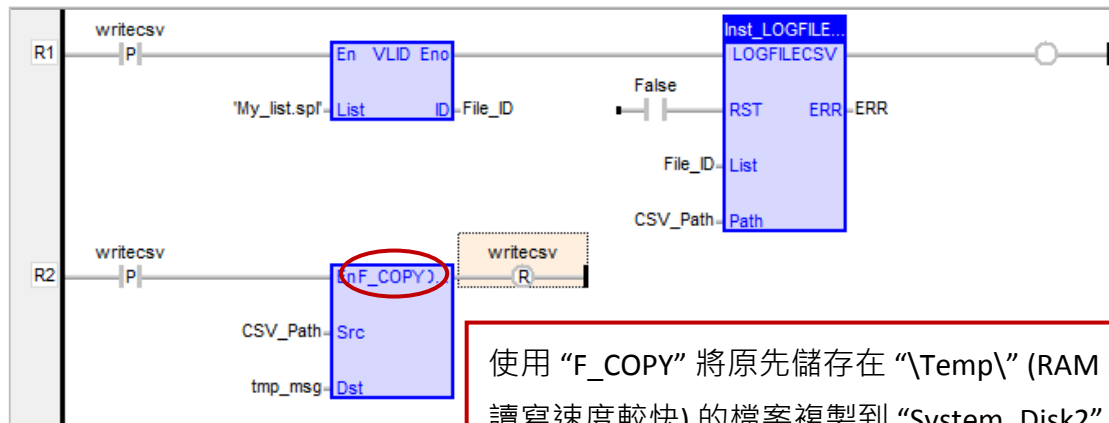
用來觸發並寫入一筆資料到 .CSV 檔案。

您可點選以下功能方塊，並按 “F1” 來查看功能說明。

VLID: 取得觀測清單 (Spy List: My_list.spl) 的值。

LogFileCSV: 將變數值存成 csv 格式的記錄檔。

F_COPY: 複製一個檔案。



使用 “F_COPY” 將原先儲存在 “\Temp\” (RAM Disk · 讀寫速度較快) 的檔案複製到 “System_Disk2” 。
(參考 ST 程式 – “Main”)

第 13 章 使用 VB.net 2008 程式來 讀/寫 Win-GRAF 變數

本章以 Visual Studio .NET 2008 開發工具建立一個範例程式的方式來說明，範例程式可以在 XP-8xx8-CE6 , WP-8xx8, WP-8xx8-CE7,VP-x2x8-CE7 , WP-5xx8-CE7 產品盒內附的 CD-ROM 內找到。

VB .NET 範例:

光碟 : \napdos\Win-GRAF\demo-project\vb.net_2008_demo\
demo_vb01 : 數位 I/O 範例 · 搭配 I-87055W 模組 (於 Slot 0)

demo_vb02 : 類比 I/O 範例 · 搭配 I-87024W (Slot 1) 與 I-8017HW (Slot 2) 模組

demo_vb03 : 讀/寫 Win-GRAF Internal Integers, Timer, Real 以及 String 變數 (無需 I/O 模組)

demo_vb04 : 讀/寫 Win-GRAF Internal String 變數 (無需 I/O 模組)

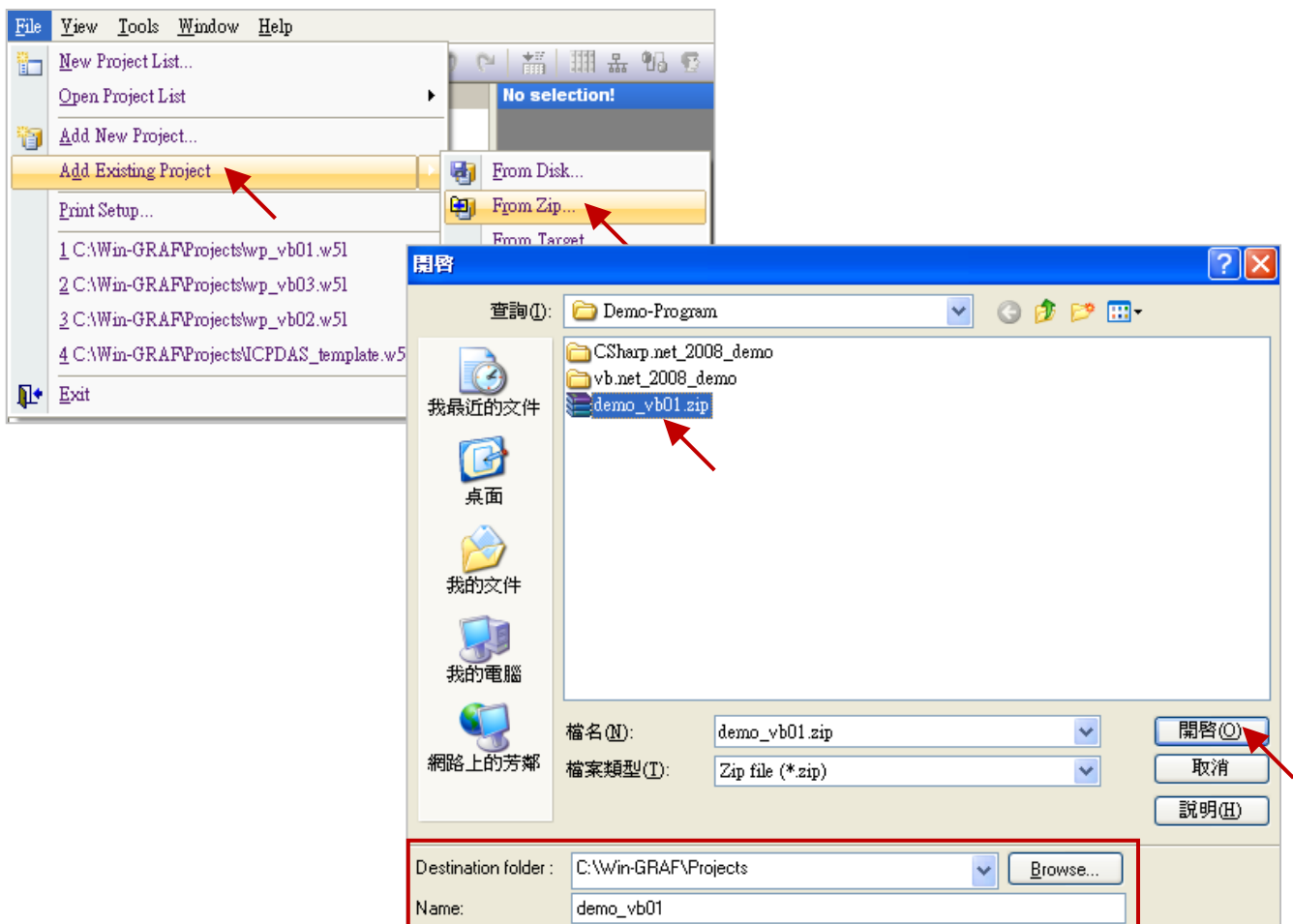
Win-GRAF 範例:

光碟 : \napdos\Win-GRAF\demo-project\
"demo_vb01.zip", "demo_vb02.zip", "demo_vb03.zip", "demo_vb04.zip"

13.1 如何回存 Win-GRAF 專案?

以下介紹如何把 Win-GRAF 範例程式，回存到 Win-GRAF Workbench 中。

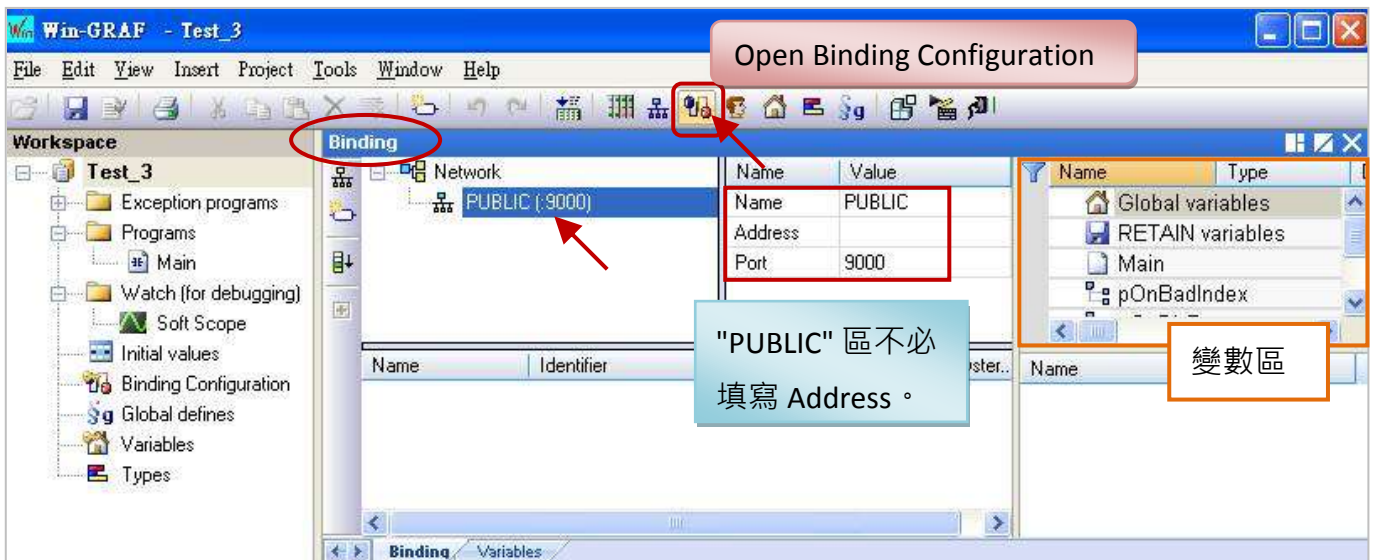
首先，滑鼠點選工具列“File”->“Add Existing Project”->“From Zip...”，並選擇想要回存的 Win-GRAF 專案 (zip 檔)，即可回存該專案。之後，請執行編譯並將專案下載到 PAC 內。



13.2 如何開放 Win-GRAF 變數給 .NET 程式使用?

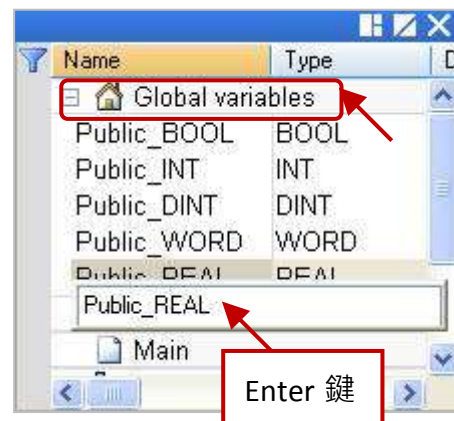
Win-GRAF 中，除了字串 (String) 變數之外，其它欲開放的變數需在 "Binding" 設定中產生一個位址才能提供 .NET 程式進行存取。以下將示範如何開放變數：

1. 滑鼠點選工具列上的 "Open Binding Configuration" 按鈕來開啟 "Binding" 視窗。
2. 點選 "PUBLIC (:9000)" 來設定要公開的資料，"Address" 欄位無需填寫，"Port" 欄位固定為 "9000"，請勿更動。



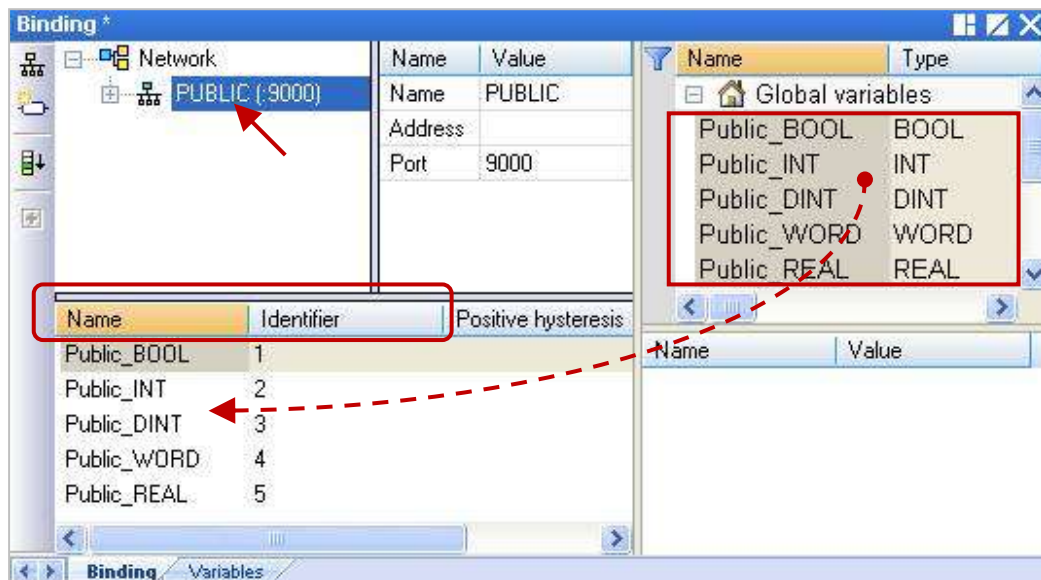
3. 在設定公開資料之前，您必須在變數區先建立好要公開的變數。滑鼠點選 "Global variables" 再按 "Ins" 鍵來新增變數項目，下表為 "Test_3" 範例所使用的變數，您可依實際需求來設定，設定完成後，畫面如下。

變數名稱	資料型態
Public_BOOL	BOOL
Public_INT	INT
Public_DINT	DINT
Public_WORD	WORD
Public_REAL	REAL



4. 如下圖，滑鼠點選在 "PUBLIC(:9000)"，接著選取變數區內欲公開的變數資料，並將其拖曳到 "Name" 區域。"Identifier" 欄位會自動產生編號，若其他 VB 或 .Net 程式想取用變數資料，需設定一樣的 ID 編號。

注意: "PUBLIC" 最多可使用 8192 個變數，"Identifier" 編號只能是 "1 ~ 8192"。



另外，如需開放 Win-GRAF 的 String 變數，可使用 "Pub_String" 函式。在 Win-GRAF 專案的程式裡，加入此段 ST 語言程式。(您可回存 Win-GRAF demo_vb04 範例程式，來查看以下程式內容)

```
Pub_string(Address, String_val);
```

Address: 開放的位址編號，範圍可以是 1 ~ 1024
String_val: String 變數的名稱

變數說明:

名稱	型態	說明
Init	BOOL	初始化用，初值設定為 True。
Tmp_val	BOOL	判斷開放位址是否成功。 "True" 表示成功，"False" 表示失敗。
msg1	STRING, 字串長度為 100	欲開放的 String 變數。 注意 : 字串長度可以是 1 ~ 255。
msg2	STRING, 字串長度為 32	
msg3	STRING, 字串長度為 60	

程式內容:

```

If init then
  Init := false;
  (* 開放位址為 1 的字串值 *)
  Tmp_val := pub_string(1,msg1);

  (* 開放位址為 2 的字串值 *)
  Tmp_val := pub_string(2,msg2);

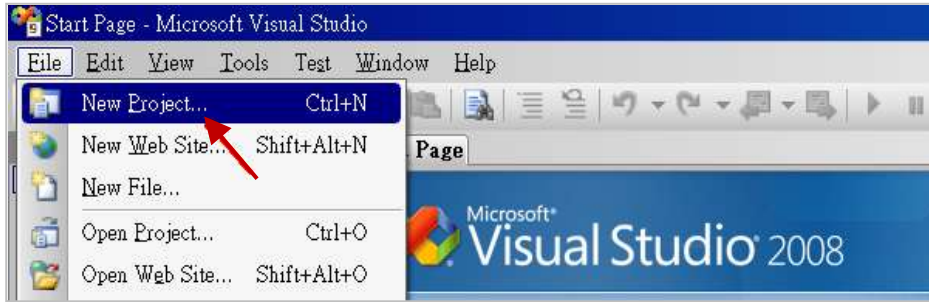
  (* 開放位址為 3 的字串值 *)
  Tmp_val := pub_string(3,msg3);

End_if;

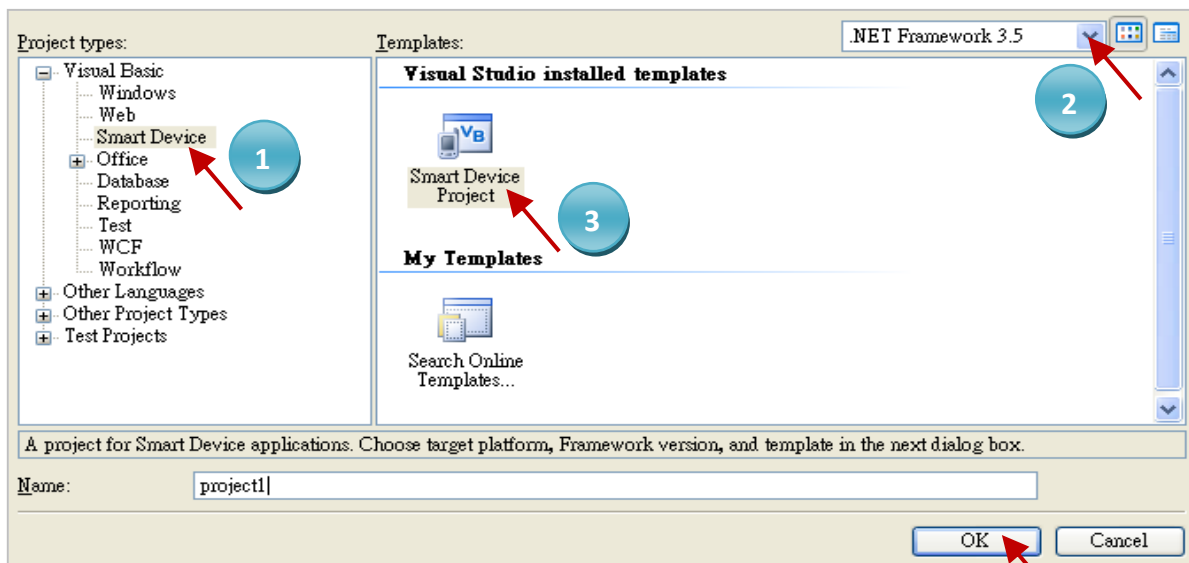
```

13.3 建立 VB.NET 新專案

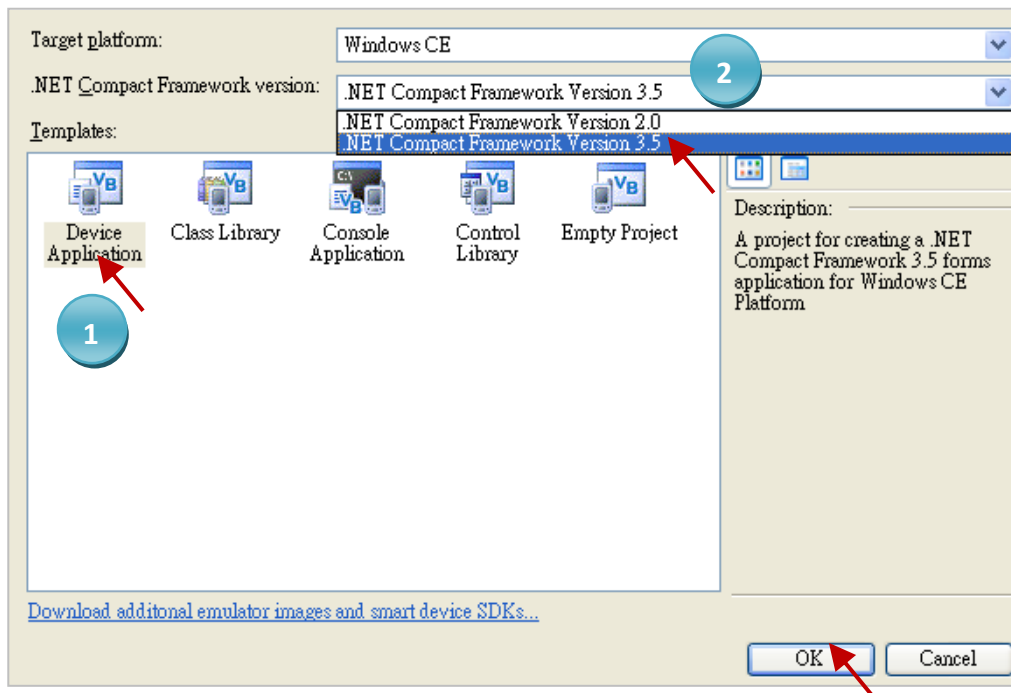
1. 開啟微軟 Visual Studio .NET 2008 軟體，點選 [File] > [New Project]。



2. 點選 [Smart Device] > [.NET framework 3.5] > [Smart Device Project]，在下方輸入專案名稱 (本例: project1)，然後按“OK”。



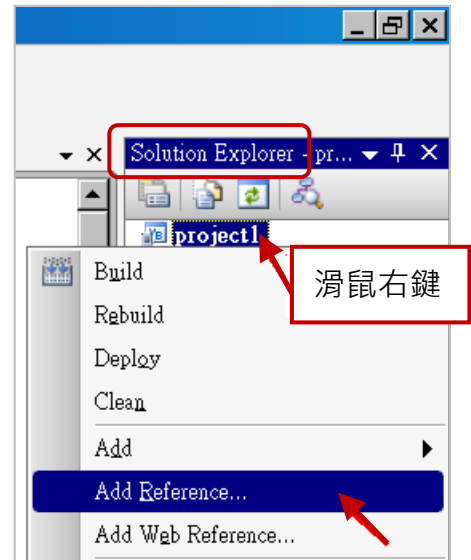
3. 點選 [Device Application] > [Windows CE] > [.NET Compact Framework Version 3.5]，然後按“OK”。



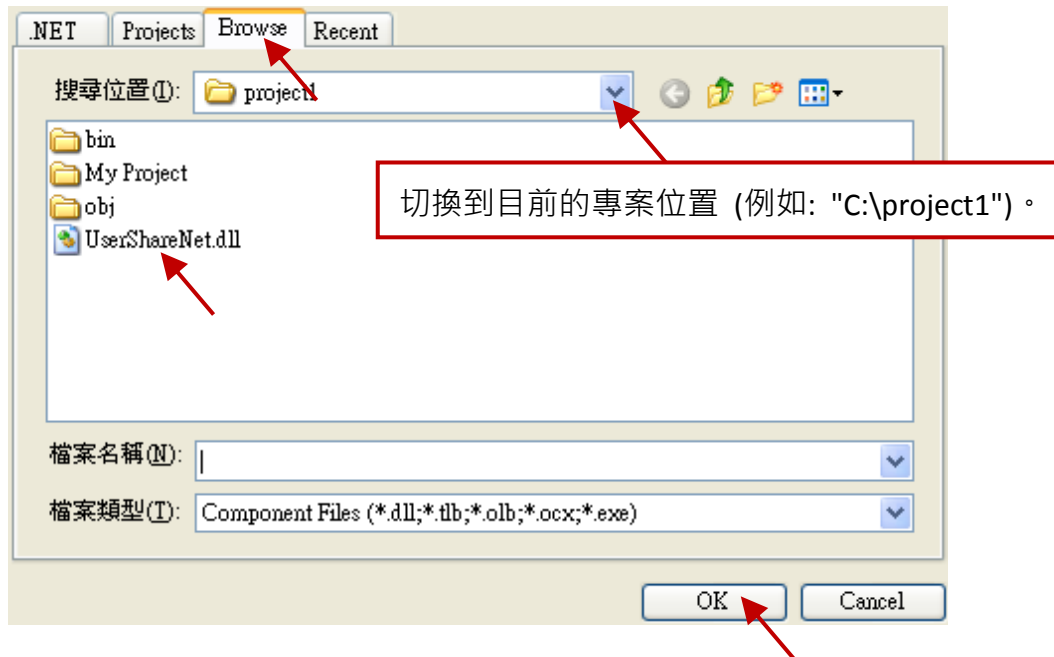
13.3.1 加入專案參考 (Project Reference)

“UserShareNet” 程式庫 (Library) 中包含了所有要跟 Win-GRAF 專案交換資料用的函式 (Function) ，而在程式裡使用 “UserShare” 關鍵字之前，您必須在應用程式的參考清單裡加入參考：“UserShareNet.dll” 。

1. 請於出貨光碟中 (\napdos\Win-GRAF\demo-project\demo_vb01\vb01\) ，將 “UserShareNet.dll” 檔複製到目前專案的位置底下 (例如: "C:\project1") 。
2. 滑鼠右鍵點擊 “Solution Explorer” 視窗中的專案名稱 (例如: "project1") ，再選擇 “Add Reference ...” 。

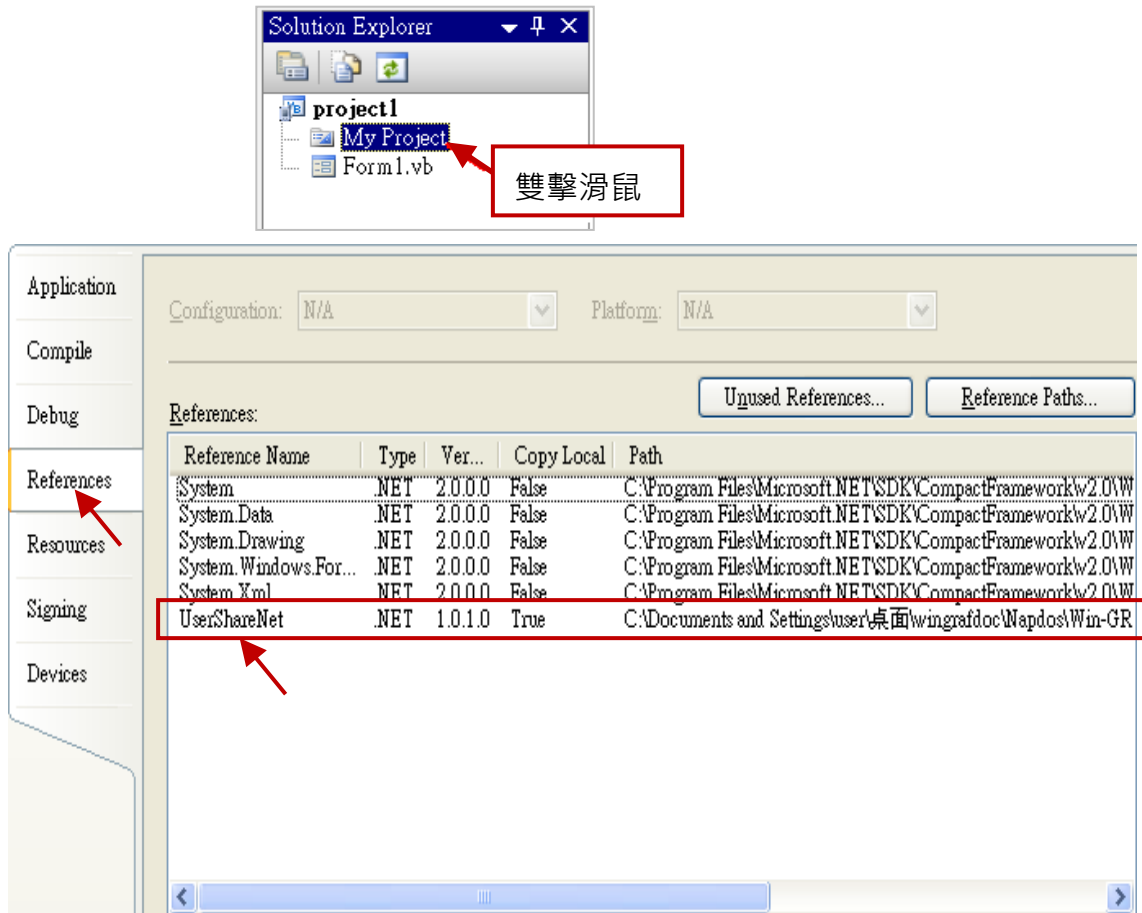


3. 點選 “Browse” 頁籤，並切換到目前的專案位置 (例如: "C:\project1") ，再選擇 “UserShareNet.dll” ，並按 “OK” 即完成了。

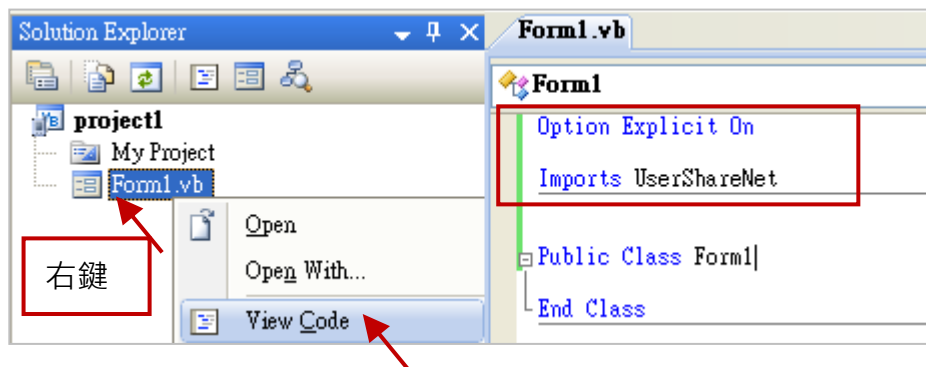


註: 加入 “UserShareNet.dll” 後，下次需開發新專案時，即可從先前開發的專案目錄中 (例如: "C:\project1") ，將此檔案複製並加入到新的專案中，或是，可預先從光碟中複製 “UserShareNet.dll” 到固定路徑中 (例如: "C:\dll_lib") 。

4. 加入 “UserShareNet.dll” 後，請雙擊專案的 “My Project”，確認是否已加入 “UserShareNet.dll”。



5. 以滑鼠右鍵點選 “Form1.vb”，選擇 “View Code”，並在第一、二行插入 “Option Explicit On” 與 “Imports UserShareNet” (如下圖)。

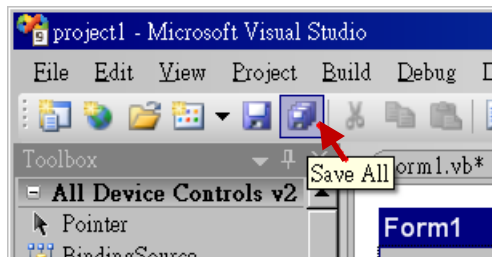


接著，就可以在您的 VB 表格中設計所需的物件與動作了。請參考 [13.5 節](#) 的說明來使用 “UserShareNet.dll” 內的函式，來讀/寫 Win-GRAF 專案內的變數資料。

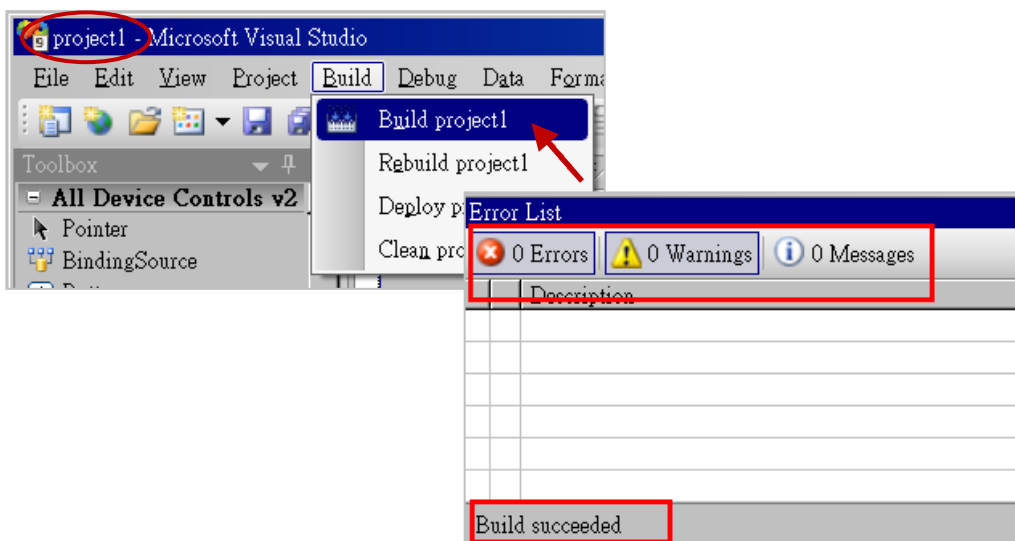
13.4 編譯應用程式

程式編寫完成後，請依照以下步驟來編譯 (Build) 並產生一個執行檔 (.exe)。

1. 請記得隨時按下 “Save All” 工具按鈕來存檔。



2. 點選 “Build” > “Build project1” 來編譯此專案 (project1)，於 “Error List” 視窗中會顯示編譯結果。



3. 編譯完成後，執行檔會存放在以下目錄中。

<您的 VB.net 專案資料夾> \bin\Release\ <project_name>.exe

請將此執行檔 (例如: “project1.exe”) 複製到 PAC 的 \System_Disk\Win-GRAF\ 目錄下來執行。

注意:

使用者可複製 VB.net 執行檔到其他目錄下執行，但請記得同時複製相關的 DLL 檔案，否則執行會有錯誤。例如: 要在 \Micro_SD\ 目錄下執行 “project1.exe”，該目錄必須有以下 3 個檔案，即 “project1.exe”、“UserShareNet.dll”、“Quicker.dll” 檔。

(可在光碟的 “\System_disk\Win-GRAF\” 目錄下取得 “UserShareNet.dll” 與 “Quicker.dll” 檔案)

13.5 "UserShareNet.DLL" 內的函式說明

本節著重於 "UserShareNet.DLL" 內的函式 (Function) 來進行說明。

"UserShareNet.DLL" 提供了許多函式，可用來讀/寫 Win-GRAF 專案裡的變數，以下分為幾類：

1. 讀/寫 Boolean
2. 讀/寫 8-bit 整數
3. 讀/寫 16-bit 整數
4. 讀/寫 32-bit 整數
5. 讀/寫 64-bit 整數
6. 讀/寫 32-bit 實數
7. 讀/寫 64-bit 實數

※請參考附錄 A 來查看 Win-GRAF 變數的資料型態與範圍

13.5.1 讀/寫 Boolean 的函式

■ Set_BOOL

說明：

設定指定位址編號的 Boolean 變數值。

語法：

```
UserShare.Set_BOOL ( iUserAddress As System.UInt16 , ByVal iStatus As byte) as Byte
```

參數：

iUserAddress：指定變數的位址編號。(1 ~ 8192)

iStatus：設定變數的狀態；例如：iStatus = 1 表示 "True"，iStatus = 0 表示 "False"

範例：

‘設定位址“1”的 Win-GRAF Boolean 變數為 "True"。

```
UserShare.Set_BOOL(Convert.ToUInt16(1), 1)
```

範例程式：

光碟：\napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb01

■ Get_BOOL

說明:

讀出指定位址編號的 Boolean 變數值。

語法:

```
UserShare.Get_BOOL ( iUserAddress As System.UInt16 , ByRef iStatus As byte)
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 取得變數的狀態 ; iStatus = 1 表示 "True" , iStatus = 0 表示 "False"

範例:

‘取得 Win-GRAF Boolean 位址編號 “1” 的變數狀態。

```
Dim iStatus As Byte
```

```
UserShare.Get_BOOL(Convert.ToUInt16(1), iStatus)
```

範例程式 :

光碟: \napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb01

13.5.2 讀/寫 整數 的函式

■ [Set_SINT](#) ■ [Set_INT](#) ■ [Set_DINT](#) ■ [Set_LINT](#)

說明:

設定指定位址編號的 Win-GRAF 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 變數值。

語法:

```
UserShare.Set_SINT (ByVal iUserAddress As System.UInt16 , ByVal iStatus As SByte) As Byte  
UserShare.Set_INT (ByVal iUserAddress As System.UInt16 , ByVal iStatus As Short) As Byte  
UserShare.Set_DINT (ByVal iUserAddress As System.UInt16 , ByVal iStatus As Integer) As Byte  
UserShare.Set_LINT (ByVal iUserAddress As System.UInt16 , ByVal iStatus As long) As Byte
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 設定 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 值

範例:

‘設定 32-bit integer 值 “1234567” 到位址編號 “1” 的變數.

```
UserShare.Set_DINT(Convert.ToUInt16(1), Convert.ToInt32(1234567) )
```

‘設定 integer 值 “-1234” 到位址編號 “2” 的變數.

```
UserShare.Set_INT(Convert.ToUInt16(3), Convert.ToInt16(-1234) )
```

‘設定 64-bit integer 值 “123456789012345” 到位址編號 “3” 的變數.

```
UserShare.Set_LINT(Convert.ToUInt16(3), Convert.ToInt64(123456789012345) )
```

‘設定 8-bit integer 值 “125” 到位址編號 “4” 的變數.

```
UserShare.Set_SINT(Convert.ToUInt16(3), Convert.ToSByte(125) )
```

範例程式:

光碟 :

1. 讀/寫 類比 I/O: \napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb02
2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:
 \napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb03

■ Get_SINT ■ Get_INT ■ Get_DINT ■ Get_LINT

說明:

讀出指定位址編號的 Win-GRAF 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 變數值。

語法:

```
UserShare. Get_SINT (ByVal iUserAddress As System.UInt16 , ByRef iStatus As SByte) As Byte  
UserShare. Get_INT (ByVal iUserAddress As System.UInt16 , ByRef iStatus As Short) As Byte  
UserShare. Get_DINT (ByVal iUserAddress As System.UInt16 , ByRef iStatus As Integer) As Byte  
UserShare. Get_LINT (ByVal iUserAddress As System.UInt16 , ByRef iStatus As long) As Byte
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 取得 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 值

範例:

```
Dim Dlong_val As Int64  
Dim short_val As Int16  
Dim long_val As Int32  
Dim Sbyte_val as sbyte
```

‘取得位址編號“7”的 64-bit 整數的變數值。

```
UserShare.Get_LINT(Convert.ToUInt16(7), Dlong_val)
```

‘取得位址編號“8”的 32-bit 整數的變數值。

```
UserShare.Get_DINT(Convert.ToUInt16(8), long_val)
```

‘取得位址編號“9”的 16-bit 整數的變數值。

```
UserShare.Get_INT(Convert.ToUInt16(9), short_val)
```

‘取得位址編號“10”的 8-bit 整數的變數值。

```
UserShare.Get_SINT(Convert.ToUInt16(9), sbyte_val)
```

範例程式:

光碟:

1. 讀/寫 類比 I/O:

\napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb02

2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:

\napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb03

13.5.3 讀/寫 實數 的函式

■ Get_REAL ■ Get_LREAL

說明:

讀出指定位址編號的 Win-GRAF 32-bit Real、64-bit Real 變數值。

語法:

```
UserShare.Get_REAL (ByVal iUserAddress As System.UInt16 , ByRef iStatus As Single) As Byte
```

```
UserShare.Get_LREAL(ByVal iUserAddress As System.UInt16 , ByRef iStatus As Double) As Byte
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 取得浮點數數值

範例:

```
Dim float_val As Single  
Dim double_val As Double
```

‘取得位址編號“7”的 double 變數值。

```
UserShare.Get_LREAL(Convert.ToUInt16(7), double_val)
```

‘取得位址編號“8”的 Single 變數值。

```
UserShare.Get_REAL(Convert.ToUInt16(8), float_val)
```

範例程式:

光碟:

1. 讀/寫 類比 I/O:
 \apidos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb02
2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:
 \apidos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb03

■ Set_REAL ■ Set_LREAL

說明:

寫入指定位址編號的 Win-GRAF 32-bit Real、64-bit Real 變數值。

語法:

```
UserShare.Set_REAL (ByVal iUserAddress As System.UInt16, ByVal iStatus As Single) As Byte
```

```
UserShare.Set_LREAL (ByVal iUserAddress As System.UInt16, ByVal iStatus As Double) As Byte
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 設定浮點數數值

範例:

‘寫入 11234.234567 到位址編號 "7" 的變數

```
UserShare.Set_LREAL(Convert.ToUInt16(7),Convert.ToDouble(11234.234567))
```

‘寫入 123.12 到位址編號 "8" 的變數

```
UserShare.Set_REAL(Convert.ToUInt16(8), Convert.ToSingle (123.12))
```

範例程式:

光碟:

1. 讀/寫 類比 I/O: \napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb02
2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:
 \napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb03

13.5.4 讀/寫字串的函式

■ Get_STRING

說明:

讀取指定位址編號的 Win-GRAF 字串變數值。

語法:

```
UserShare.Get_STRING (ByVal iUserAddress As System.UInt16, ByVal msg() As Byte) As Byte
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 1024)

msg() : 取得/設定 Win-GRAF 的字串值

範例:

```
Dim str_val As String
```

```
Dim msg() As Byte
```

‘取得位址編號“7”的 String 變數值。

```
UserShare.Get_STRING(Convert.ToUInt16(7),msg )
```

```
str_val= byte_array_to_unicode(msg)
```

```
Private Function byte_array_to_unicode(ByVal buf() As Byte) As String
```

```
    Dim tmpmsg As String
```

```
    If buf.Length > 255 Then
```

```
        Return Nothing
```

```
    End If
```

```
tmpmsg = System.Text.Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length)
```

```
    Return tmpmsg
```

```
End Function
```

範例程式:

光碟:

1. 讀/寫 STRING 的值:

```
\napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb04
```


■ Set_STRING

說明:

寫入指定位址編號的 Win-GRAF 字串值。

語法:

```
UserShare.Set_STRING (ByVal iUserAddress As System.UInt16, ByVal msg() As Byte) As Byte
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 1024)

msg() : 設定 Win-GRAF 的字串值

範例:

```
Dim str_val As String="Hellow World"
```

```
Dim msg() As Byte
```

```
msg= unicode_to_byte_array(str_val)
```

‘寫入位址編號“7”的 String 變數值。

```
UserShare.Set_STRING(Convert.ToUInt16(7),msg )
```

‘轉換字串成 byte array

```
Private Function unicode_to_byte_array(ByVal msg As String) As Byte()
```

```
    Dim tmpbuf() As Byte
```

```
    If msg.Length > 255 Then
```

```
        Return Nothing
```

```
    End If
```

```
    tmpbuf = System.Text.Encoding.GetEncoding("UTF-8").GetBytes(msg)
```

```
    Return tmpbuf
```

```
End Function
```

範例程式:

光碟:

1. 讀/寫 STRING 的值:

```
\napdos\Win-GRAF\demo-project\vb.net_2008_demo\demo_vb04
```

13.5.5 如何讓 VB.NET 程式讀取 Win-GRAF 字串變數?

.NET 應用程式如果要寫入字串變數，必須根據目前使用的語言編碼 (例如: UTF-8) 轉換成 byte 陣列才能寫入，若是讀出字串變數的陣列內容，則需要根據語言編碼轉換成字串。以下提供 VB.NET 程式轉換的範例

(編碼為 UTF-8) :

String 轉成 byte 陣列

```
Private Function unicode_to_byte_array(ByVal msg As String) As Byte()  
    Dim tmpbuf() As Byte  
    If msg.Length > 255 Then  
        Return Nothing  
    End If  
  
    tmpbuf = System.Text.Encoding.GetEncoding("UTF-8").GetBytes(msg)  
    Return tmpbuf  
End Function
```

byte 陣列轉字串

```
Private Function byte_array_to_unicode(ByVal buf() As Byte) As String  
    Dim tmpmsg As String  
    If buf.Length > 255 Then  
        Return Nothing  
    End If  
    tmpmsg = System.Text.Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length)  
  
    Return tmpmsg  
End Function
```

第 14 章 使用 C# 程式來讀/寫 Win-GRAF 變數

本章以 Visual Studio .NET 2008 開發工具建立一個範例程式的方式來說明，範例程式可以在 XP-8xx8-CE6, WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7, WP-5xx8-CE7 產品盒內附的 CD-ROM 內找到。

C# 範例:

光碟 : \napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\
demo_CSharp01 : 數位 I/O 範例，搭配 I-87055W 模組 (於 Slot 0)

demo_CSharp02 : 類比 I/O 範例，搭配 I-87024W (Slot 1) 與 I-8017HW (Slot 2) 模組

demo_CSharp03 : 讀/寫 Win-GRAF Internal Integer, Timer, 及 Real 變數 (無需 I/O 模組)

demo_CSharp04 : 讀/寫 Win-GRAF String 變數 (無需 I/O 模組)

Win-GRAF 範例:

光碟 : \napdos\Win-GRAF\demo-project\
"demo_vb01.zip", "demo_vb02.zip", "demo_vb03.zip", "demo_vb04.zip"

14.1 如何回存 Win-GRAF 專案?

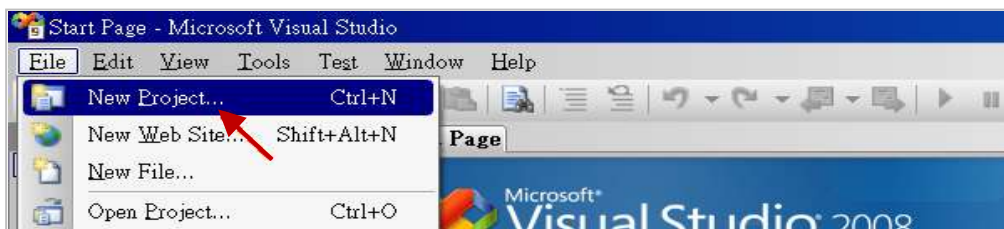
請參考 [13.1 節](#) 來回存 Win-GRAF 專案。

14.2 如何開放 Win-GRAF 變數給 C# 程式使用?

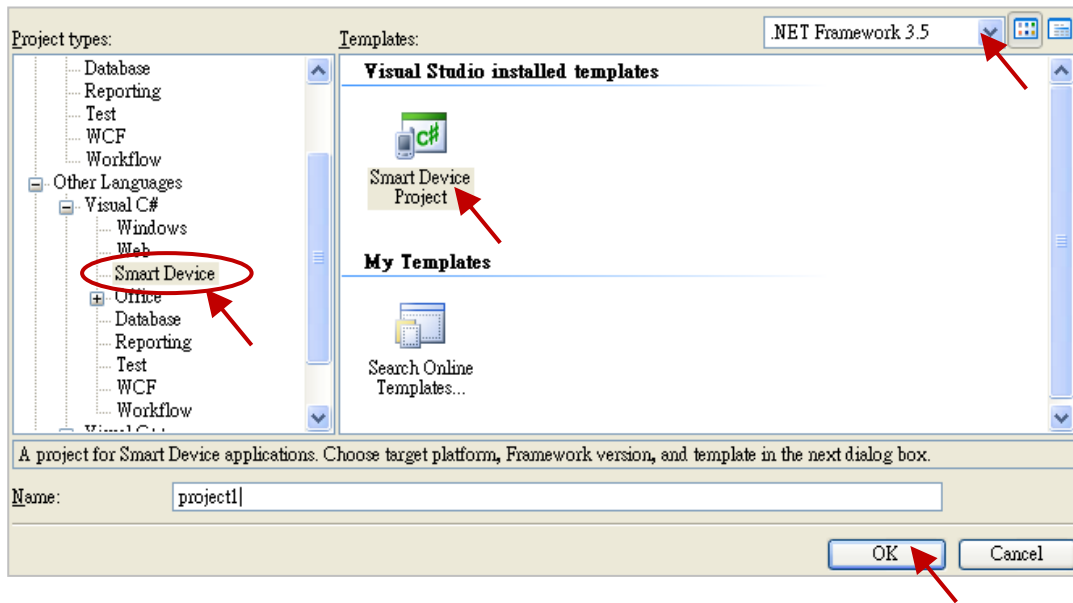
請參考 [13.2 節](#) - 開放 Win-GRAF 變數給 .NET 程式使用的方式。

14.3 建立 C# 新專案

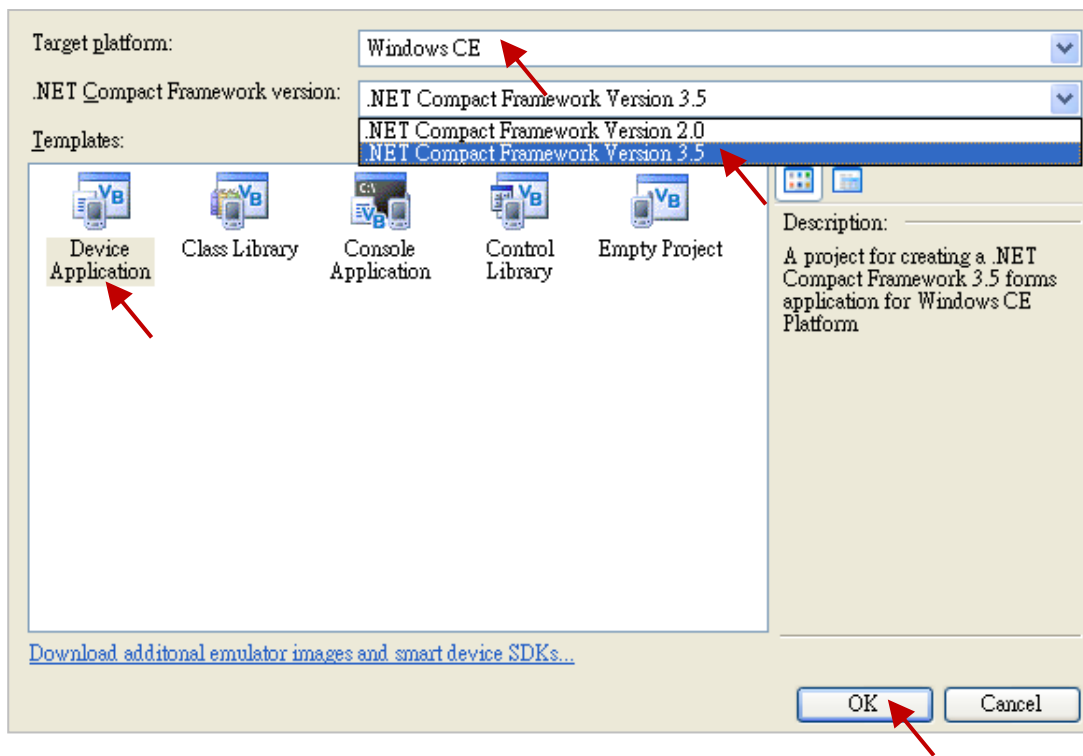
1. 開啟微軟 Visual Studio .NET 2008，點選功能表 [File] > [New Project]。



2. 點選 [Smart Device] > [.NET frame work 3.5] > [Smart Device Project]，並輸入專案名稱 (例如: "project1")，然後點選 "OK"。



3. 點選 [Device Application] > [Windows CE] > [.NET Compact Framework Version 3.5]，然後點選 "OK"。

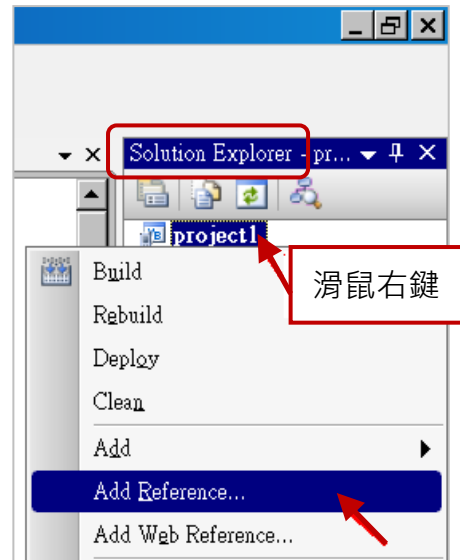


14.3.1 加入 C# 專案參考

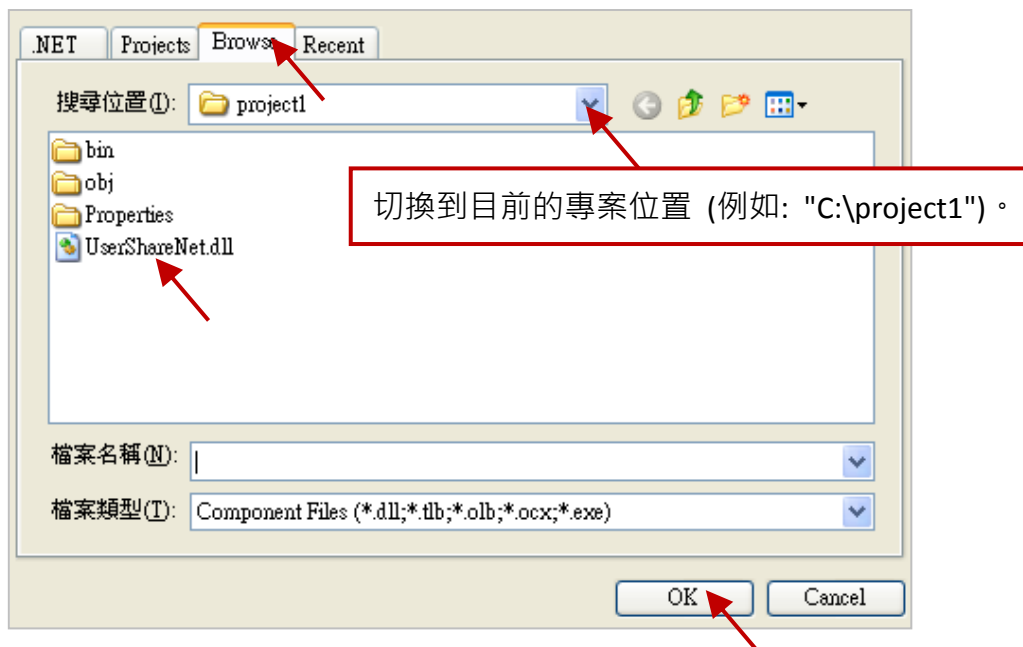
“UserShareNet” 程式庫 (Library) 中包含了所有要跟 Win-GRAF 專案交換資料用的函式，而在程式裡使用 “UserShare” 關鍵字之前，您必須在應用程式的參考清單裡加入參考: “UserShareNet.dll”。

1. 請於出貨光碟中 (\napdos\Win-GRAF\demo-project\demo_CSharp01\demo_CSharp01\)，將 “UserShareNet.dll” 檔複製到目前專案的位置底下(例如: "C:\project1")。

2. 滑鼠右鍵點選 “Solution Explorer” 視窗中的專案名稱 (例如: "project1")，再選擇 “Add Reference ...”。

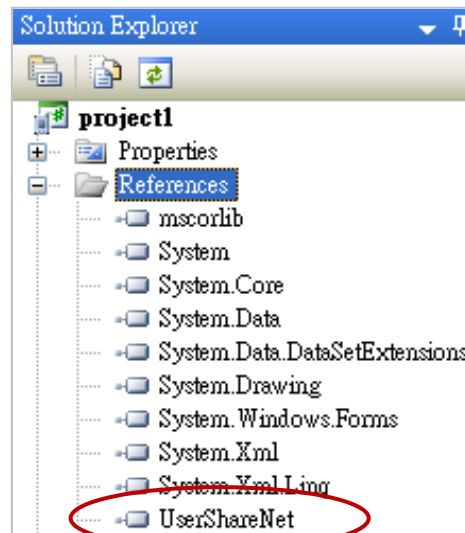


3. 點選 “Browse” 頁籤，並切換到目前的專案位置 (例如: "C:\project1")，再選擇 “UserShareNet.dll”，並按 “OK” 即完成了。

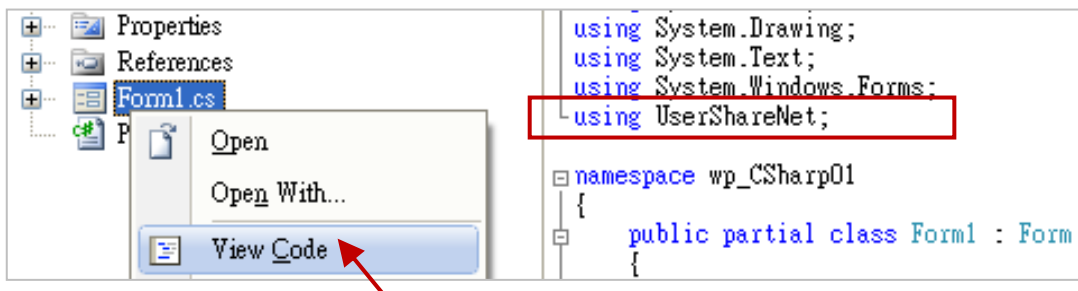


註: 加入 “UserShareNet.dll” 後，下次需開發新專案時，即可從先前開發的專案目錄中 (例如: "C:\project1")，將此檔案複製並加入到新的專案中。或是，可預先從光碟中複製 “UserShareNet.dll” 到固定路徑中 (例如: "C:\dll_lib")。

4. 加入 “UserShareNet.dll” ，會出現在 “Solution Explorer” 視窗。



5. 以滑鼠右鍵點選 “Form1.cs” ，選擇 “View Code” ，將游標移到最上方 ，在第一個區段中加入 “using UserShareNet” 。

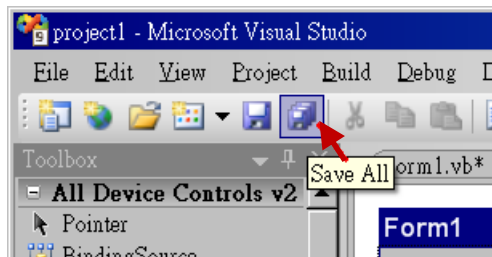


接著，就可以在您的 C# 表單中設計所需的物件與動作了。請參考 14.5 節 的說明來使用 “UserShareNet.dll” 內的函式，來讀/寫 Win-GRAF 內的變數資料。

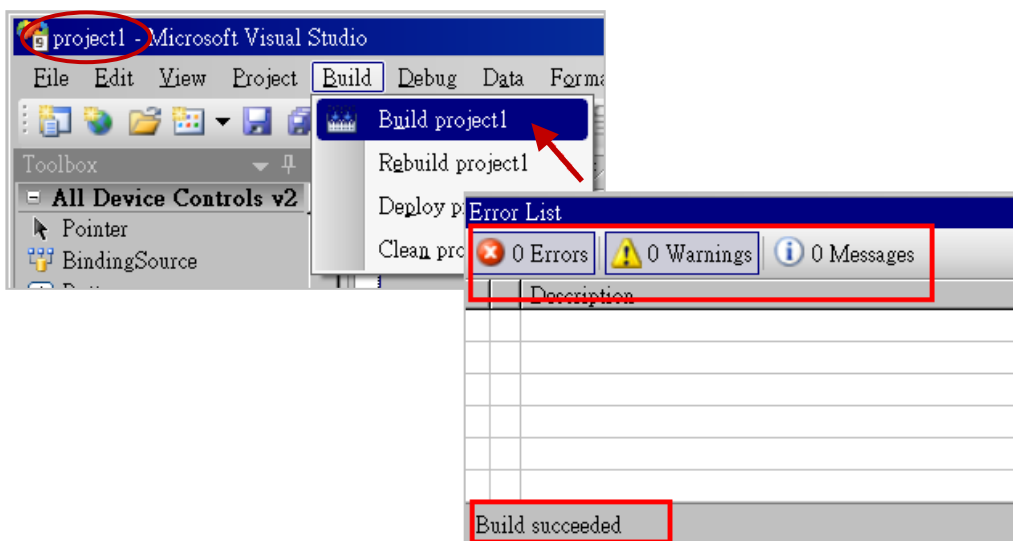
14.4 編譯應用程式

程式編寫完成後，請依照以下步驟來編譯 (Build) 並產生一個執行檔 (.exe)。

1. 請記得隨時按下 “Save All” 工具按鈕來存檔。



2. 點選 “Build” > “Build project1” 來編譯此專案 (project1)，於 “Error List” 視窗中會顯示編譯結果。



3. 編譯完成後，執行檔會存放在以下目錄中。

<您的 C# .net 專案資料夾> \bin\Release\ <project_name>.exe

請將此執行檔 (例如: “project1.exe”) 複製到 PAC 的 \System_Disk\Win-GRAF\ 目錄下來執行。

注意:

使用者可複製 C# .net 執行檔到其他目錄下執行，但請記得同時複製相關的 DLL 檔案，否則執行會有錯誤。例如: 要在 \Micro_SD\ 目錄下執行 “project1.exe”，該目錄必須有以下 3 個檔案，即 “project1.exe”、“UserShareNet.dll”、“Quicker.dll” 檔。

(可在光碟的 “\System_disk\Win-GRAF\” 目錄下取得 “UserShareNet.dll” 與 “Quicker.dll” 檔案)

14.5 "UserShareNET.DLL" 內的函式說明

本節將針對 "UserShareNet.DLL" 內的函式 (Function) 來進行說明。

"UserShareNet.DLL" 提供了許多函式，可用來讀/寫 Win-GRAF 專案裡的變數，以下分為幾類：

1. 讀/寫 Boolean
2. 讀/寫 8-bit 整數
3. 讀/寫 16-bit 整數
4. 讀/寫 32-bit 整數
5. 讀/寫 64-bit 整數
6. 讀/寫 32-bit 實數
7. 讀/寫 64-bit 實數

※請參考附錄 A 來查看 Win-GRAF 的變數資料型態與範圍

14.5.1 讀/寫 Boolean 的函式

■ Set_BOOL

說明：

設定指定位址編號的 Win-GRAF Boolean 變數值。

語法：

```
UserShare.Set_BOOL(ushort iUserAddress, byte iStatus)
```

參數：

iUserAddress：指定變數的位址編號 (1 ~ 8192)

iStatus：設定變數的狀態；例如: iStatus = 1 表示 "True"，iStatus = 0 表示 "False"。

範例：

```
//設定位址 "1" 的 Win-GRAF 變數為 True.
```

```
UserShare.Set_BOOL(Convert.ToUInt16(1), 1);
```

範例程式：

光碟：\napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp01

■ Get_BOOL

說明:

讀出指定位址編號的 Win-GARF Boolean 變數值。

語法:

```
UserShare.Get_BOOL(ushort iUserAddress, out byte iStatus)
```

參數:

iUserAddress : 指定變數位址編號 (1 ~ 8192)

iStatus : 取得變數的狀態 ; iStatus = 1 表示 "True" , iStatus = 0 表示 "False" 。

範例:

```
Byte iStatus=0;  
//取得位址編號 "1" 的變數狀態.  
UserShare.Get_BOOL(Convert.ToUInt16(1),out iStatus);
```

範例程式:

光碟: \napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp01

14.5.2 讀/寫 整數 的函式

■ Set_SINT ■ Set_INT ■ Set_DINT ■ Set_LINT

說明:

設定指定位址編號的 Win-GRAF 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 變數值。

語法:

```
UserShare.Set_SINT(ushort iUserAddress , sbyte iStatus)

UserShare.Set_INT(ushort iUserAddress , short iStatus)

UserShare.Set_DINT(ushort iUserAddress, int iStatus)

UserShare.Set_LINT(ushort iUserAddress, long iStatus)
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 設定 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 值

範例:

//設定 32-bit integer 值 “1234567” 到位址編號 “1” 的變數.

```
int temp1=1234567;
UserShare.Set_DINT(Convert.ToUInt16(1), temp );
```

//設定 16-bit Integer 值 “-1234” 到 Modbus 位址編號 “2” 的變數.

```
short temp2= -1234;
UserShare.Set_INT(Convert.ToUInt16(2), temp2 );
```

//設定 64-bit Integer 值 “123456789012345” 到位址編號 “3” 的變數.

```
long temp3=123456789012345;
UserShare.Set_LINT(Convert.ToUInt16(3), temp3 );
```

//設定 8-bit Integer “125” 值到位址編號 “4” 的變數.

```
Sbyte temp4=125;
UserShare.Set_SINT(Convert.ToUInt16(4), temp4 );
```

範例程式:

光碟:

1. 讀/寫 類比 I/O: \napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp02
2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:
 \napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp03

■ Get_SINT ■ Get_INT ■ Get_DINT ■ Get_LINT

說明:

讀出指定位址編號的 Win-GRAF 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 變數值。

語法:

```
UserShare.Get_SINT(ushort iUserAddress, out sbyte iStatus)
```

```
UserShare.Get_INT(ushort iUserAddress, out short iStatus)
```

```
UserShare.Get_DINT(ushort iUserAddress, out int iStatus)
```

```
UserShare.Get_LINT(ushort iUserAddress, out long iStatus)
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 取得 8-bit Integer, 16-bit Integer, 32-bit Integer 及 64-bit Integer 值

範例:

```
Int64 Dlong_val;  
Int16 short_val;  
Int32 long_val ;  
sbyte sbyte_val;
```

```
//取得位址編號“7”的 64-bit 整數的變數值.
```

```
UserShare.Get_LINT(Convert.ToUInt16(7),out Dlong_val);
```

```
//取得位址編號“8”的 32-bit 整數的變數值.
```

```
UserShare.Get_DINT(Convert.ToUInt16(8),out long_val);
```

```
//取得位址編號“9”的 16-bit 整數的變數值.
```

```
UserShare.Get_INT(Convert.ToUInt16(9),out short_val);
```

```
//取得位址編號“10”的 8-bit 整數的變數值.
```

```
UserShare.Get_SINT(Convert.ToUInt16(9),out sbyte_val)
```

範例程式:

光碟:

1. 讀/寫 類比 I/O:

```
\napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp02
```

2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:

```
\napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp03
```

14.5.3 讀/寫 實數 的函式

■ Get_REAL ■ Get_LREAL

說明:

讀出指定位址編號的 Win-GRAF 32-bit Real、64-bit Real 變數值。

語法:

```
UserShare. Get_REAL (System.UInt16 iUserAddress, out float iStatus)
```

```
UserShare. Get_LREAL(ByVal iUserAddress As System.UInt16 , out Double iStatus)
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 取得浮點數數值

範例:

```
float float_val;
```

```
double double_val;
```

```
// 取得位址編號 "7" 的 double 變數值.
```

```
UserShare.Get_LREAL(Convert.ToUInt16(7),out double_val);
```

```
//取得位址編號 "8" 的 float 變數值.
```

```
UserShare.Get_REAL(Convert.ToUInt16(8),out float_val);
```

範例程式:

光碟:

1. 讀/寫 類比 I/O:

```
\napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp02
```

2. 讀/寫 Internal long integer、Timer 及 Real (浮點數) 的值:

```
\napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_demo_CSharp03
```

■ Set_REAL ■ Set_LREAL

說明:

寫入指定的位址編號的 Win-GRAF 32-bit Real、64-bit Real 變數值。

語法:

```
UserShare. Set_REAL ( ushort iUserAddress , float iStatus )  
UserShare. Set_LREAL( ushort iUserAddress , Double iStatus)
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 8192)

iStatus : 設定浮點數數值

範例:

// 寫入 11234.234567 到位址編號 "7" 的變數

```
UserShare.Set_LREAL(Convert.ToUInt16(7),Convert.ToDouble(11234.234567));
```

//寫入 123.12 到位址編號 "8" 的變數

```
UserShare.Set_REAL(Convert.ToUInt16(8), Convert.ToSingle (123.12));
```

範例程式:

光碟:

1. 讀/寫 類比 I/O:
 \ napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp02
2. 讀/寫 internal long integer、Timer 及 Real (浮點數) 的值:
 \ napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp03

14.5.4 讀/寫 字串的函式

■ Set_STRING

說明:

寫入指定位址編號的 Win-GRAF 字串值。

語法:

```
UserShare.Set_STRING (ushort addr , Byte [] msg)
```

參數:

addr : 指定變數的位址編號 (1 ~ 1024)

msg[] : 設定 Win-GRAF String 變數值。

範例:

```
String str_val;  
Byte[] msg;
```

```
// 寫入位址編號 "7" 的 String 變數值。  
msg= unicode_to_byte_array(str_val);  
UserShare.Set_STRING(Convert.ToUInt16(7),msg );
```

```
//String 轉成 byte 陣列  
private byte[] unicode_to_byte_array(string msg)  
{  
    byte[] tmpbuf;  
    if (msg.Length > 255)  
        return null;  
  
    tmpbuf = Encoding.GetEncoding("UTF-8").GetBytes(msg);  
    return tmpbuf;  
}
```

範例程式:

光碟:

1. 讀/寫 STRING 的值:
 \
 \apidos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp04

■ Get_STRING

說明:

讀取指定位址編號的 Win-GRAF 字串變數值。

語法:

```
UserShare.Set_STRING (ushort addr , Byte [] msg)
```

參數:

addr : 指定變數的位址編號 (1 ~ 1024)

msg[] : 設定 Win-GRAF String 變數值。

範例:

```
String str_val= "Hello World";  
Byte[] msg;
```

```
// 設定位址編號 "7" 的 String 變數值。  
UserShare.Get_STRING(Convert.ToUInt16(7),msg );  
str_val= byte_array_to_unicode(msg);
```

```
//byte 陣列轉字串  
private string byte_array_to_unicode(byte[] buf)  
{  
    string tmpmsg;  
    if (buf.Length > 255)  
        return null;  
  
    tmpmsg = Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length);  
    return tmpmsg;  
}
```

範例程式:

光碟:

1. 讀/寫 STRING 的值: \napdos\Win-GRAF\demo-project\CSharp.net_2008_demo\demo_CSharp04

14.5.5 如何讓 C# 程式讀取 Win-GRAF 字串變數?

.NET 程式如果要寫入字串變數，必須根據目前使用的語言編碼 (例如: UTF-8) 轉換成 byte 陣列才能寫入，若是讀出字串變數的陣列內容，則需要根據語言編碼轉換成字串。以下提供 C# 程式轉換的範例 (編碼為 UTF-8)：

//String 轉成 byte 陣列

```
private byte[] unicode_to_byte_array(string msg)
{
    byte[] tmpbuf;
    if (msg.Length > 255)
        return null;

    tmpbuf = Encoding.GetEncoding("UTF-8").GetBytes(msg);
    return tmpbuf;
}
```

//byte 陣列轉字串

```
private string byte_array_to_unicode(byte[] buf)
{
    string tmpmsg;
    if (buf.Length > 255)
        return null;

    tmpmsg = Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length);
    return tmpmsg;
}
```

第 15 章 如何在 Win-GRAF PAC 上使用 eLogger HMI

eLogger 是泓格科技 (ICP DAS) 開發的 HMI 軟體工具，具備簡單好用的特性，它不僅支援 Local 的 HMI 畫面，也提供 Web Server 的 HMI 畫面。使用者可在 eLogger 中設計好 HMI 畫面，並和 Win-GRAF 程式一起運行在同一台 PAC 內。

15.1 Win-GRAF 專案

1. 請將光碟中的範例檔案，複製到 PC 的桌面上並解壓縮。

Win-GRAF PAC CD-ROM: \napdos\Win-GRAF\demo-project\Soft-GRAF-demo\demo_faq018_all.zip

Win-GRAF 與 eLogger 專案對照表:

Win-GRAF 專案	eLogger 專案	說明
eL01.zip	eL_01.wez	設計 Local HMI 與 Web HMI
eL02.zip	eL_02.wez	使用一個可瞬間 ON 之後再 OFF 的控制按鈕
eL03.zip	eL_03.wez	讀取與變更 PAC 的日期/時間 與 進行時間控制

2. 回存 Win-GRAF 專案 (.zip) 到 PC 的 Win-GRAF workbench 內 (可參考 [13.1 節](#))。

3. 再將 Win-GRAF 專案 (例如: eL01) 下載到 PAC 中 (可參考 [2.3.4 節](#) 與 [2.3.5 節](#))。

更多關於 Win-GRAF + eLogger HMI 的使用說明，請參訪:

www.icpdas.com > Support > FAQ > [Win-GRAF Soft-Logic PAC](#) > [FAQ-018](#) 或
http://www.icpdas.com/root/support/faq/win-graf_tc.php

15.2 eLogger 專案

您可在 eLogger 網頁中，下載並安裝 eLogger 軟體。接著將 eLogger 專案 (.wez) 全都複製到 C:\ICPDAS\eLogger\eLogger_Developer\Project，再將專案 (例如: eL_01) 下載到 PAC 中。

eLogger 網頁:

www.icpdas.com > Product > Solution > Software > [SCADA/HMI](#) > [eLogger](#)
http://www.icpdas.com/root/product/solutions/software/scada_hmi/elogger/elogger.html

eLogger 軟體:

<http://ftp.icpdas.com/pub/cd/winpac/napdos/elogger/setup/>

eLogger 手冊:

<http://ftp.icpdas.com/pub/cd/winpac/napdos/elogger/document/>

第 16 章 備援 (冗余) 系統

16.1 特色與架構

ICP DAS 的 **Win-GRAF PAC - XP-8xx8-CE6** 系列有支援備援 (冗余) 系統:

Win-GRAF PAC 備援是使用 2 台 PAC 來達成，其中一台 PAC 的旋轉式開關 (Rotary switch) 需設為 7 (表示 Main-PAC)，另一台 PAC 則需設為 9 (表示 Backup-PAC)。當其中一台 PAC 意外損壞或由 User 的應用程序判定必需切換 PAC 控制權時，會自動切換到可以正常運作的那台 PAC。

Win-GRAF 備援系統的特色

1. 安全性佳:

2 台 PAC 間有 3 條 Cable (LAN1, LAN2, Alive Port) 可互相通訊，不會因為其中 1 條 或其中 2 條 Cable 斷線，備援功能就發生失控。Win-GRAF PAC 備援系統只要仍有 1 條 Cable 通訊正常，此備援系統就仍然可正常控制整個程序。

2. 單一的 Public IP:

Win-GRAF 備援系統提供單一的對外 IP 位址給圖控/HMI 來連接，圖控就不需判斷要挑哪一台 PAC 的 IP 位址來連接。

3. 維修安裝方便:

當備援系統開始上線運作後，日後若有某一台 PAC 發生故障，可將此故障 PAC 單獨斷電拆下 (**注意:** 不要斷電 與 不要拆下 正常運作的那台 PAC，需讓它繼續運作。) 後續可拿另外一台備品 PAC (或修好的 PAC)，不需額外再燒錄 Win-GRAF 應用程序，直接調整好該 PAC 的旋轉式開關，與接好通訊線 (LAN1, LAN2, Alive port, I/O)，並確認原先正常的 PAC 仍在運作中，才將此備品開機，如此，原先正常的 PAC 會自動把 Win-GRAF 應用程序 與全部備援資料都自動傳給剛上線的那台 PAC。維修安裝方便簡單，操作員不必擔心是否要安裝應用程序，PAC 會自動安裝好。

例外: 若 User 在備援系統內除了 Win-GRAF 應用程式之外，還有 Run 其它應用程式，比如 C, VB.net, C# 應用程式 或 eLogger HMI 應用程式，則需把這些應用程式 預先安裝 到那台要新裝上的備品 PAC (或修好的 PAC) 內，之後才可以把這台備品 PAC (或修好的 PAC) 裝置到備援系統內。

4. 應用程序設計簡單:

User 只需設計好應用程序。不需額外設計哪些備援資料需傳到另一台 PAC，Win-GRAF 備援系統會自動把 備援資料備份到另一台 PAC。

5. User 可在應用程序內 額外制定一些其它 安全機制:

比如若主控的那台 PAC 的 LAN1 斷線 (導致 SCADA 連不上) 或某 RS-485 通訊 Port 斷線或故障。諸如此類，User 的應用程序可以判定出來，然後進行切換控制權到正常的那台 PAC。

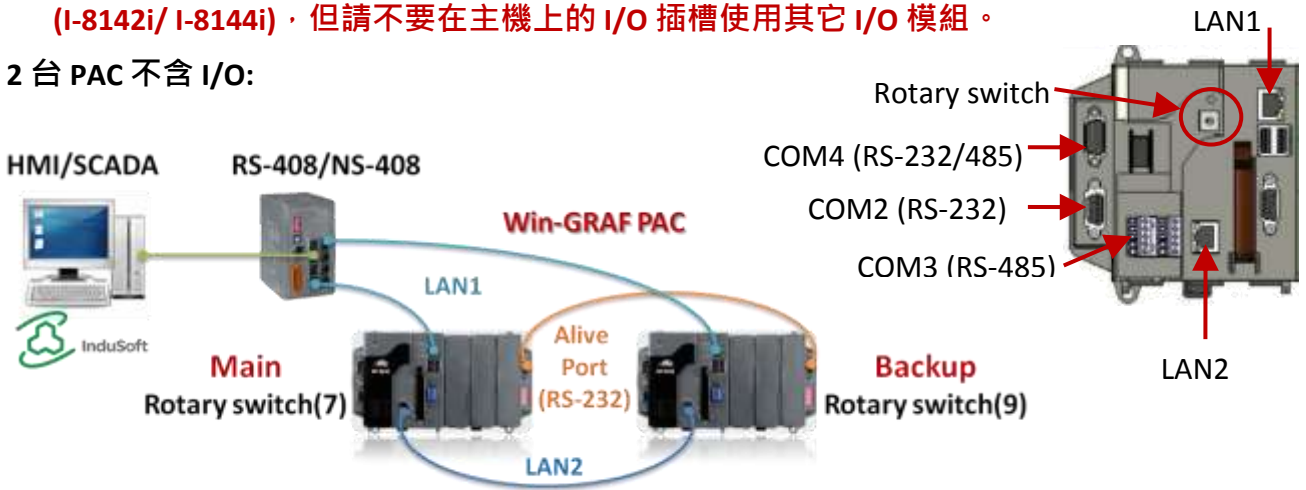
6. I/O 備援:

除了 PAC 有備援外，若 I/O 是選用 [IDCS-8830 系列 I/O](#)，則 I/O 模組一樣可以備援。

Win-GRAF 備援系統的架構 (以 XP-8xx8-CE6 為例):

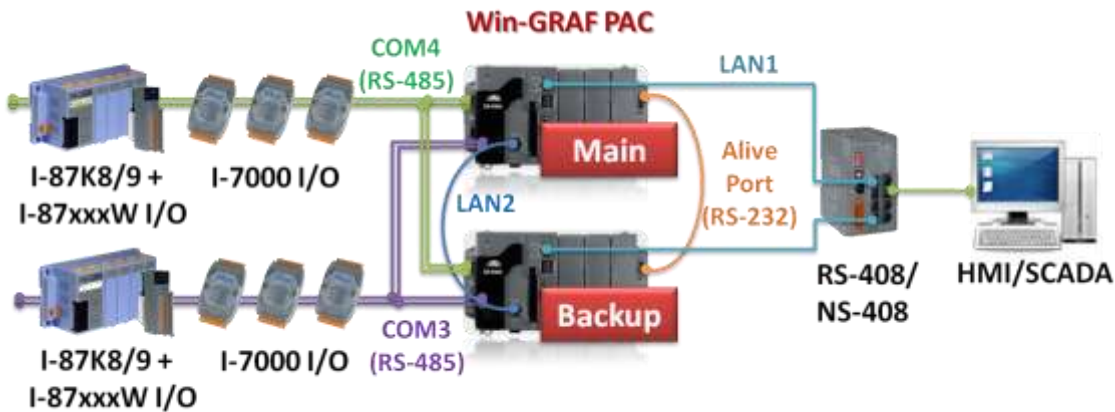
注意: Win-GRAF 備援 PAC 的主機上的 I/O 插槽，可視應用需要插上 RS-485/ RS-422 擴充卡 (I-8142i/ I-8144i)，但請不要在主機上的 I/O 插槽使用其它 I/O 模組。

1. 2 台 PAC 不含 I/O:



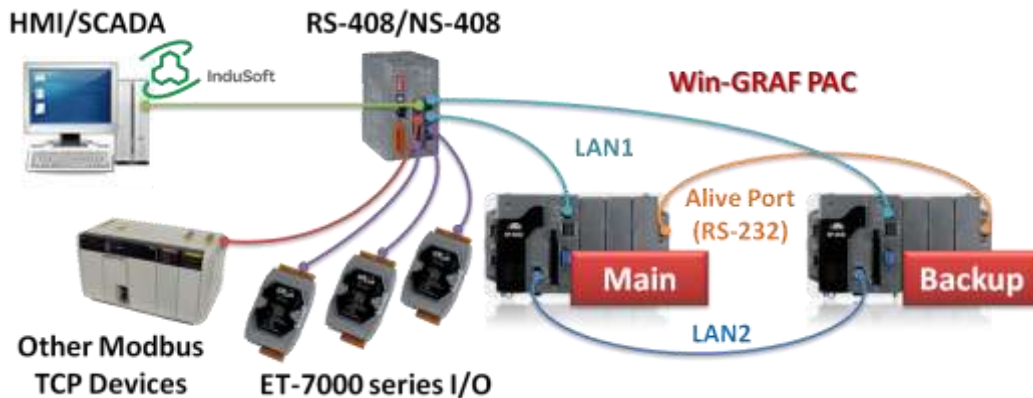
註: LAN1: 一般網路線, LAN2: 乙太網路跳線 (Cross, 交叉線), Alive Port: RS-232 跳線 (Cross, 交叉線)。

2. 2 台 PAC 搭配 DCON I/O:



註: LAN1: 一般網路線, LAN2: 乙太網路跳線 (Cross, 交叉線), Alive Port: RS-232 跳線 (Cross, 交叉線)。
COM3, COM4 (RS-485): Data+ 接 Data+ ; Data- 接 Data- 。

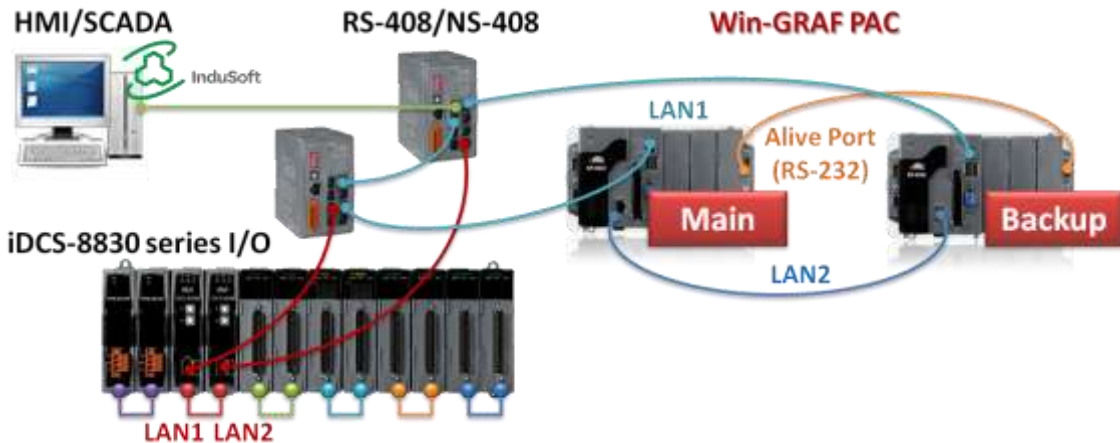
3. 2 台 PAC 搭配 Modbus TCP I/O:



註: LAN1: 一般網路線, LAN2: 乙太網路跳線 (Cross, 交叉線), Alive Port: RS-232 跳線 (Cross, 交叉線)。

4. 2 台 PAC 搭配 iDCS-8830 I/O:

本架構除了 CPU 備援外，I/O 模組也有備援。

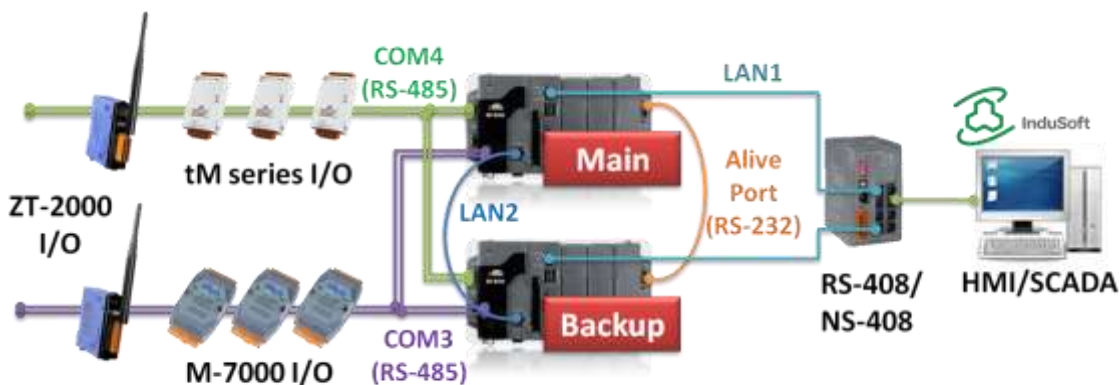


註: LAN1 (PAC), LAN1 (iDC-8830), LAN2 (iDC-8830): 一般網路線。

LAN2 (PAC): 乙太網路跳線 (Cross, 交叉線), Alive Port: RS-232 跳線 (Cross, 交叉線)。

每台 iDCS-8830 內插的 I/O 模組是 2 個相同型號的模組為一組。

5. 2 台 PAC 搭配 其它 Modbus RTU/ASCII I/O:



註: LAN1: 一般網路線, LAN2: 乙太網路跳線 (Cross, 交叉線), Alive Port: RS-232 跳線 (Cross, 交叉線)。

COM3, COM4 (RS-485): Data+ 接 Data+ ; Data- 接 Data- 。

6. (2) ~ (5) 所連接的 I/O 也可以混合搭配 2 種 或 以上 來使用。

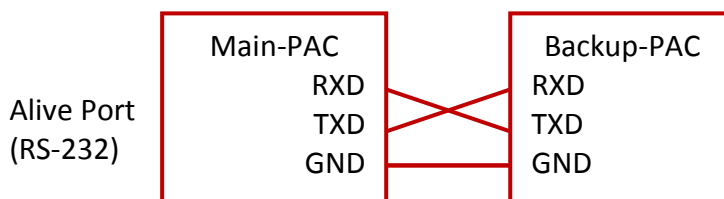
Win-GRAF 備援系統是由 2 台 PAC 所構成，其中一台 PAC 需把旋轉式開關 (Rotary switch) 設定為 7 (此台稱呼為 Main-PAC)，而另一台的 Rotary switch 需設成 9 (此台稱呼為 Backup-PAC)，不可拿 2 台 Main-PAC 或 2 台 Backup-PAC 來組成一個 Win-GRAF 備援系統。

16.2 重要通訊 Port 與安裝事項

Win-GRAF 備援系統的 2 台 PAC 間需有以下 3 個 通訊 Port 可以互相通訊。

1. Alive Port:

Win-GRAF 備援系統的 2 台 PAC 會使用一個 RS-232 Port 當成 Alive Port (或稱 Heart-beat Port)。此 Alive Port 的連接線是使用一條 RS-232 跳線 (Cross Cable 或 NULL Modem Cable)，它的 RXD、TXD、GND 這 3 個 RS-232 訊號接腳，必需如下方互相接到 2 台 PAC。



2. Replication Port:

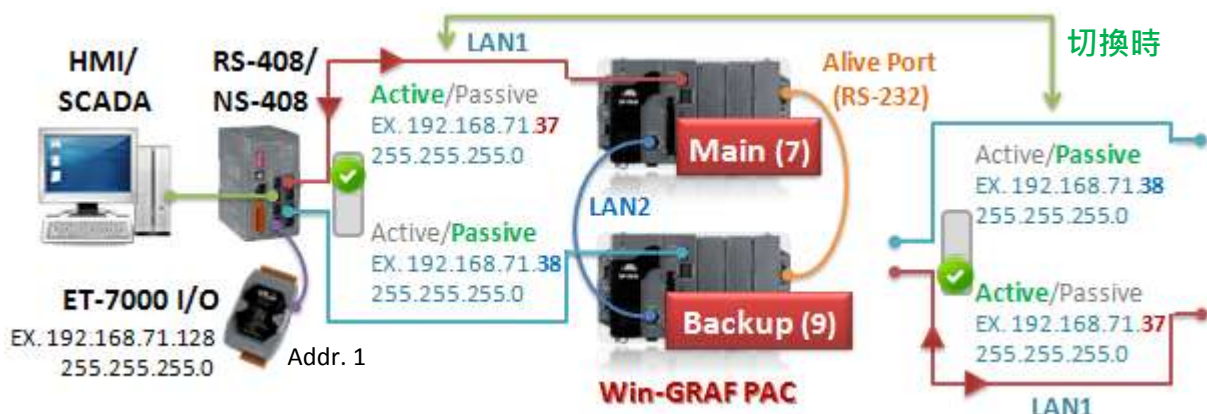
Win-GRAF 備援系統的 2 台 PAC 會使用 LAN2 這個 Ethernet Port 來當成備援資料傳輸 Port。2 台 PAC 的 LAN2 必需直接使用一條 CROSS Ethernet Cable (交叉線) 互相接好，中間不可接任何 Ethernet Switch 或 Hub。不然可能會發生錯誤 或 常發生來不及通訊 (Timeout) 的狀況。此 LAN2 必需專屬於這 2 台 PAC 間互相高速傳遞備援資料，不可再外接任何設備與 Switch/Hub。



LAN2 Port 需使用乙太網路跳線 (Crossover)，直接連接兩台 PAC，不可使用一般網路線。

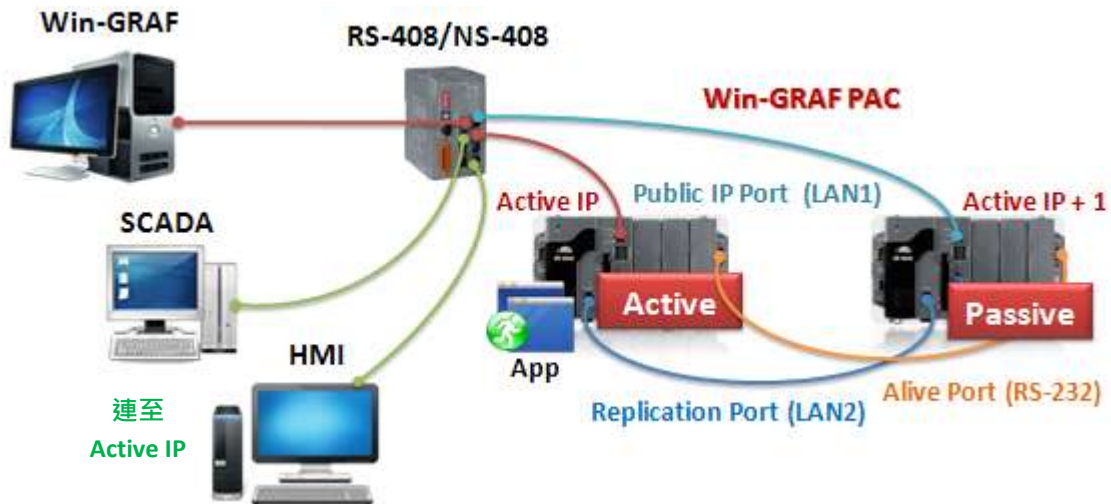
3. Public IP Port:

Win-GRAF 備援系統的 2 台 PAC 的 LAN1 這個 Ethernet Port 請使用一般網路線接到 Ethernet Switch 來對外通訊，可以用來跟圖控(組態軟件)，或 HMI 來通訊。也可用來連接控制 Modbus TCP I/O 或設備或其它 Ethernet 設備。這個 LAN1 會自動切換 IP 位址，即當該 PAC 為主控 (Active) 時就會切換它的 LAN1 為 User 在程序內指定的 "Active_IP" 位址。當該 PAC 為 Passive 時，它的 LAN1 會自動切換為 "Active_IP+1" 位址。SCADA/HMI 請都使用 "Active_IP" 這個位址來跟 Win-GRAF 備援系統通訊。



安裝注意事項 (非常重要):

1. 在該 PAC 通電前，必需確認旋轉式開關 (Rotary switch) 的設定是否正確，一台是 7，另一台必須是 9。若設定錯誤，可能會發生備援系統失控。
2. Win-GRAF 備援系統 安裝到工作現場時，必需至少確保以下 3 條 通訊線有安裝好 (若還有其它通訊線，比如 RS-485，也都先裝好)，之後才可以對該 PAC 通電。
若以下 3 條 通訊線還沒接好就對 PAC 通上電源，可能會發生 備援系統失控。
 - A. 2 台 PAC 的 Alive Port 要使用一條 CROSS cable 直接互接。
 - B. 2 台 PAC 的 LAN2 要使用一條 CROSS Ethernet Cable 直接互接，不可經過任何 乙太網路交換器/集線器 (Ethernet Switch/Hub)。
 - C. 2 台 PAC 的 LAN1 請使用一般網路線接到 Ethernet Switch。
3. 若原先已經有一台 PAC 已經在現場運作了，不可把該運作中的 PAC 關電 (或關機)。
請把要新裝上的 PAC 先經過上方 (1) 與 (2) 設定 與 安裝好後，才可對此新裝上的 PAC 通上電源。



只有目前為主控 (Active) 的那台 PAC 會運行 Win-GRAF 應用程式，即擁有備援系統的控制權。而 Passive 那台並不會運行 Win-GRAF 應用程式，它只是等待日後可以接收控制權並隨時接收 Active PAC 丟過來的備援資料。

16.3 範例程式介紹

Win-GRAF PAC 出貨 CD-ROM 內有提供 demo_RDN_1.zip、demo_RDN_2.zip、demo_RDN_3.zip 與 demo_RDN_4.zip 這 4 個備援系統範例程式。請參考 [第 12 章](#) 的方式，將專案回存到 Win-GRAF Workbench 內。

範例名稱	說明
demo_RDN_1	2 台 XP-8xx8-CE6，使用 COM3 去連 3 個 DCON I/O 模組。
demo_RDN_2	2 台 XP-8xx8-CE6，沒有連接任何 I/O 模組。
demo_RDN_3	2 台 XP-8xx8-CE6，使用 LAN1 經過 Ethernet Switch 去連接一個 ET-7050 (Modbus TCP I/O 模組)。
demo_RDN_4	2 台 XP-8xx8-CE6，使用 LAN1 經過 Ethernet Switch 去連接一個 iDCS-8830 (PAC 與 I/O 皆備援)。

以下會使用 "demo_RDN_2.zip" 這個範例程式來介紹。

16.3.1 "I/O Board" 設定

demo_RDN_2, demo_RDN_3, demo_RDN_4:

為了能在 PAC 中使用備援功能，需在 "I/O Board" 視窗內連上 "i_redundancy" 功能 (參考 [第 4 章](#))。

The screenshot shows the "I/O Boards" window with a list of slots. Slot 10 is selected and labeled "i_redundancy". A "Close" button is visible. The "10: i_redundancy - Properties" dialog is open, showing the following configuration:

- Key = 6
- Ref = 16#3
- Active IP = 192.168.71.37
- Passive IP = auto
- Mask = 255.255.255.0
- Gateway IP = disabled
- Reserved0 = 0
- Reserved1 = 0
- Reserved2 = 0
- Reserved3 = 0
- Reserved4 = 0

Below the properties, there is a section for "i_redundancy" with the text: "Enable Redundancy in the PAC. The following PAC support redundancy. XP-8xx8-CE6, XP-9xx8-CE6, WP-5248".

Red callout boxes provide the following instructions:

- "注意: 請使用 Slot 8 之後的位置。" (Note: Please use the position after Slot 8.)
- "注意: Active IP 位址的最後一個數字不可能是 0, 254, 255, 必須是 1 ~ 253。" (Note: The last digit of the Active IP address cannot be 0, 254, or 255, it must be 1 ~ 253.)
- "請參考設定說明來獲得更多資訊。" (Please refer to the configuration manual for more information.)

設定說明:

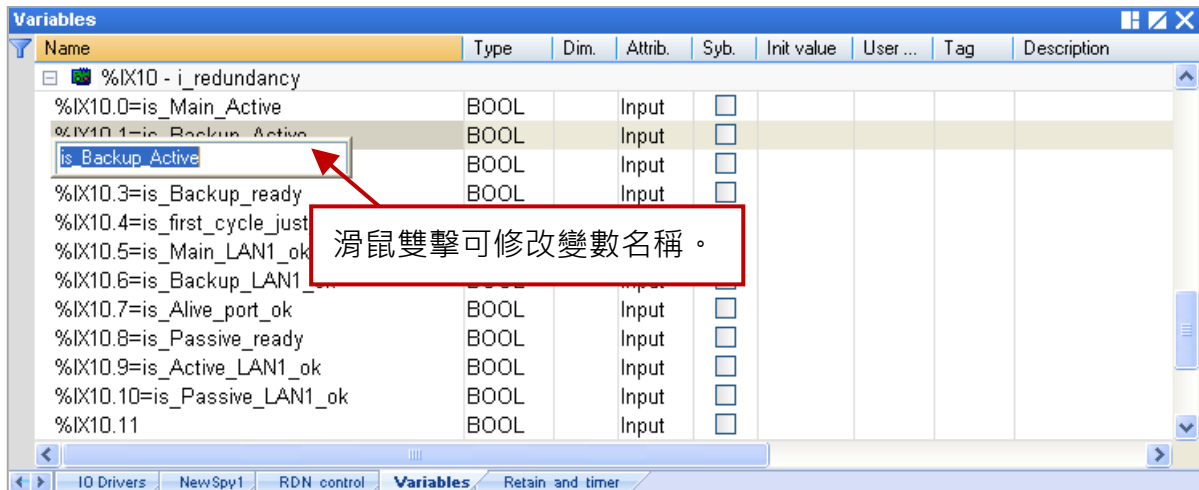
Active_IP: 此備援系統對外公開的 IP 位址，主要給 HMI/SCADA 來連接通訊用。
(**注意:** Active IP 位址的最後一個數字不可能是 0, 254, 255, 必須是 1 ~ 253。)

Passive_IP: 固定為 Auto，它會等於目前 Passive PAC 的 LAN1 的 IP 位址，此 IP 位址會自動指定為 Active_IP + 1。

(例如: 若 Active_IP 為 192.168.71.37，則 Passive_IP 為 192.168.71.38)

Mask: 通常設為 255.255.255.0 或 255.255.0.0 (視網路環境而定)。

在 "I/O Board" 視窗內連上 "i_redundancy" 後，會自動在 "Variables" 視窗中新增 12 個 "BOOL" 輸入變數，用來表示目前備援系統的狀態。



Ch.0 (is_Main_Active): Main-PAC (旋轉開關 = 7) 是否為 Active PAC (擁有控制權)?

TRUE: 為 Active PAC，FALSE: 為 Passive PAC。

Ch.1 (is_Backup_Active): Backup-PAC (旋轉開關 = 9) 是否為 Active PAC (擁有控制權)?

TRUE: 為 Active PAC，FALSE: 為 Passive PAC。

Ch.2 (is_Main_ready): Main-PAC 目前是否就緒? 若 Ch.2 回傳 FALSE，可能是以下原因:

(1) Main-PAC 與 Backup-PAC 之間的 LAN2 連線損毀或無法通訊。

(2) Main-PAC 當機或損毀。

(3) Main-PAC 的旋轉開關未設定為 7。

Ch.3 (is_Backup_ready): Backup-PAC 目前是否就緒? 若 Ch.3 回傳 FALSE，可能是以下原因:

(1) Main-PAC 與 Backup-PAC 之間的 LAN2 連線損毀或無法通訊。

(2) Backup-PAC 當機或損毀。

(3) Backup-PAC 的旋轉開關未設定為 9。

Ch.4 (is_first_cycle_just_after_switch): 僅適用於 Active PAC。

True: 目前為切換後的第一個週期。

False: 目前不是切換後的第一個週期。

Ch.5 (is_Main_LAN1_ok): Main-PAC 的 LAN1 Port 通訊正常嗎?

TRUE: 正常，FALSE: 失敗或網路線斷訊。

Ch.6 (is_Backup_LAN1_ok): Backup-PAC 的 LAN1 Port 通訊正常嗎?

TRUE: 正常，FALSE: 失敗或網路線斷訊。

Ch.7 (is_Alive_port_ok): True : Alive Port 通訊正常。

False : Alive Port 通訊失敗 或 Passive PAC 當機或損毀。

Ch.8 (is_Passive_ready): Passive PAC 目前是否就緒？若 Ch.8 回傳 FALSE，可能是以下原因：

(1) Main-PAC 與 Backup-PAC 之間的 LAN2 網路連線損毀。

(2) Passive PAC 當機或損毀。(3) Passive PAC 的旋轉開關設定錯誤。

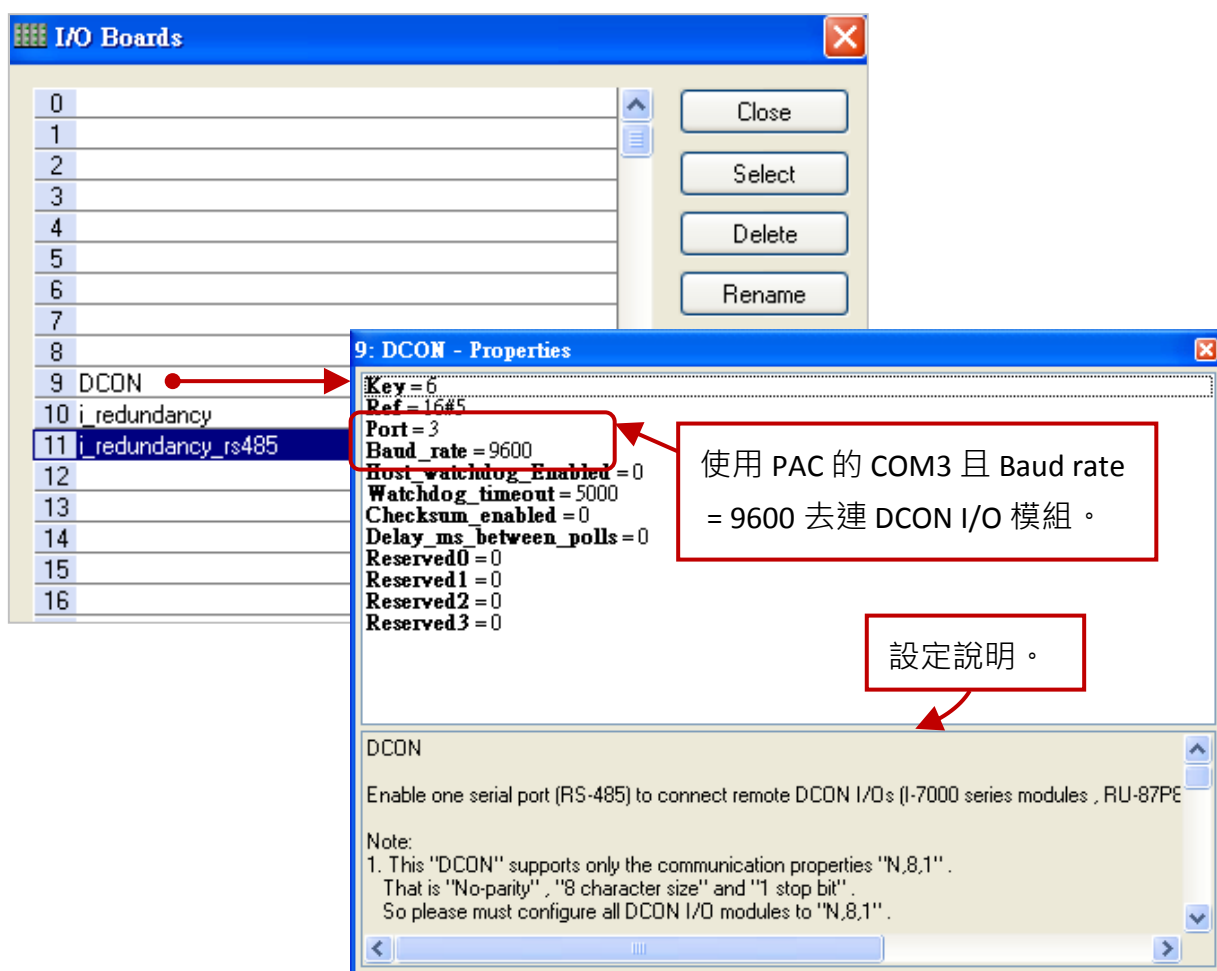
Ch.9 (is_Active_LAN1_ok): Active PAC 的 LAN1 Port 通訊正常嗎？

TRUE: 正常，FALSE: 失敗 或 網路線斷訊。

Ch.10 (is_Passive_LAN1_ok): Passive-PAC 的 LAN1 Port 通訊正常嗎？

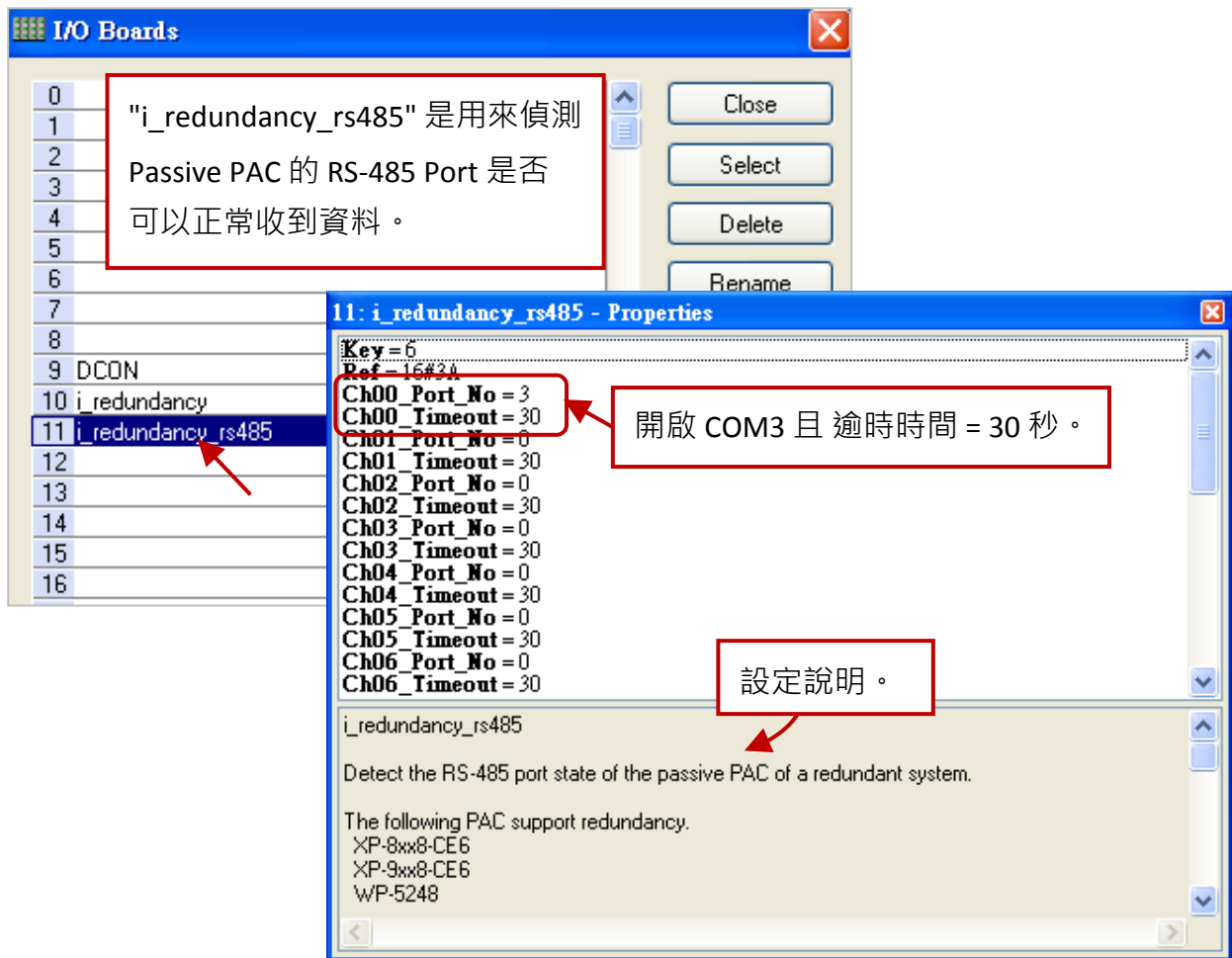
TRUE: 正常，FALSE: 失敗或 網路線斷訊。

demo_RDN_1: 此程式使用 PAC 的 COM3 (RS-485) 去連 DCON I/O 模組。



重要事項:

1. 必同時使用 "i_redundancy" 否則 "i_redundancy_rs485" 將無效。
2. "i_redundancy_rs485" 只會開啟 Passive PAC 相關的 RS-485 Port 來接收資料並不會傳送資料。
3. "i_redundancy_rs485" 是用來偵測 Passive PAC 的 RS-485 Port 是否可以正常收到資料。



設定說明:

Ch00_Port_No ~ Ch15_Port_No :

Passive PAC 的 RS-485 Port 編號。

可以是 0 或 1 ~ 33 (不同的 PAC 型號 Port 編號會不同) , 設定為 0 表示不啟用。

Ch00_Timeout ~ Ch15_Timeout :

單位為秒 , 可以是 1 ~ 60 秒。

若相關的 RS-485 Port 在該時間內皆沒有接收到資料 , 則該通道狀態會是 FALSE。

16 通道 Boolean 輸入 :

用來表示 Passive PAC 的 RS-485 Port 狀態。

TRUE : 正常開啟相關的 RS-485 Port 且可接收資料。

FALSE: 相關的 RS-485 Port 開啟失敗 或 逾時間隔時間內沒有接收到資料。

16.3.2 變數宣告 (demo_RDN_2)

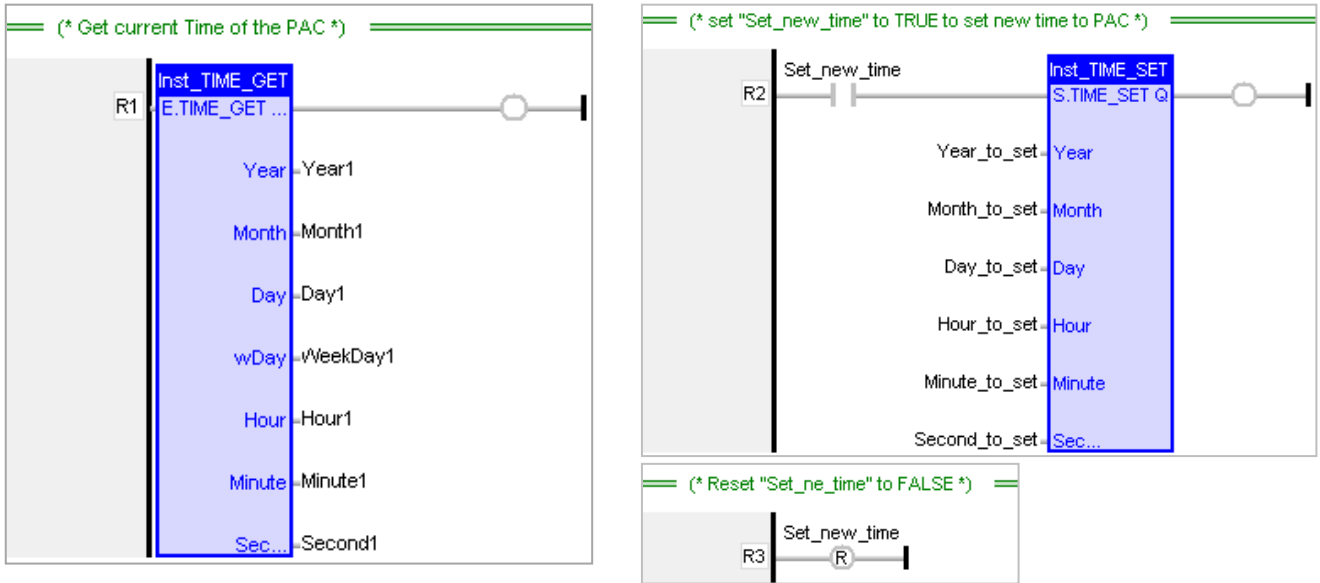
您可在 "Variable" 視窗中查看/建立變數 (參考 [2.3 節](#))。

變數名稱	資料型態	說明
Year1	DINT	於 "PAC_Time" 程式中，用來取得 PAC 的系統時間。
Month1	DINT	
Day1	DINT	
WeekDay1	DINT	
Hour1	DINT	
Minute1	DINT	
Second1	DINT	
Set_new_time	BOOL	設定為 "TRUE" 以設定新的時間。
Year_to_set	DINT	於 "PAC_Time" 程式中，用來設定 PAC 的系統時間。
Month_to_set	DINT	
Day_to_set	DINT	
Hour_to_set	DINT	
Minute_to_set	DINT	
Second_to_set	DINT	
DINT_1	DINT	於 "Retain_and_timer" 程式中，設定為保存變數。
DINT_2	DINT	
REAL_1	REAL	
REAL_2	REAL	
TMR_1	TIME	Timer
TMR_2	TIME	
retain_done	BOOL	TRUE: 已設定過可保存變數；FALSE: 尚未設定。
on_line_change_cycle	DINT	非 0，表示本週期為 On-Line Change 後的第 1 個週期。
tmp_bool	BOOL	回傳 Retain 狀態用。
TMR_1_last_state	BOOL	TRUE: 計時中；FALSE: 未計時，"Sleep" 中。
TMR_2_last_state	BOOL	TRUE: 計時中；FALSE: 未計時，"Sleep" 中。
To_tick_TMR_1	BOOL	設定為 TRUE 以開始計時 TIMER1。
To_tick_TMR_2	BOOL	設定為 TRUE 以開始計時 TIMER2。
To_stop_TMR_1	BOOL	設定為 TRUE 以停止計時 TIMER1。
To_stop_TMR_2	BOOL	設定為 TRUE 以停止計時 TIMER2。

16.3.3 程式介紹 (demo_RDN_2)

LD 程式 – “PAC_Time”

用來取得/設定 PAC 的系統時間。



LD 程式 – “RDN_control”

當 Active PAC 的 LAN1 發生問題時，若 Passive PAC 就緒且 Passive PAC 的 LAN1 通訊正常，則 Active PAC 會在 10 秒後重開機，將控制權切換到 Passive PAC。



ST 程式 – "Retain_and_timer"

(* 宣告 "on_line_change_cycle" 為 DINT (非 0，表示正在 On-Line Change 後的第一個執行週期)。
 宣告 "retain_done" 為 BOOL 且初始值為 FALSE。
 宣告 "tmp_bool" 為 BOOL。 *)

```

on_line_change_cycle := GetSysInfo (_SYSINFO_CHANGE_CYCLE);
if (retain_done = FALSE) or
(is_first_cycle_just_after_switch = TRUE) or
(on_line_change_cycle <> 0) then
    retain_done := TRUE; (* 執行保存變數一次 *)
    tmp_bool := Retain_Var( DINT_1, 1); (* 設定保存 DINT 變數 *)
    tmp_bool := Retain_Var( DINT_2, 2);
    tmp_bool := Retain_Var( REAL_1, 3); (*設定保存 REAL 變數 *)
    tmp_bool := Retain_Var( REAL_2, 4);
    
```

(* 若保存變數尚未初始化，則使用以下預設值 *)

```
if (DINT_1 < -1000000) or (DINT_1 > 1000000) or
   (DINT_2 < -2000000) or (DINT_2 > 2000000) or
   (REAL_1 < -9.9E10) or (REAL_1 > 9.9E10) or
   (REAL_2 < -9.9E10) or (REAL_2 > 9.9E10) then
  DINT_1 := 0 ;
  DINT_2 := 0 ;
  REAL_1 := 0.0 ;
  REAL_2 := 0.0 ;
end_if ;
end_if ;
```

(* is_first_cycle_just_after_switch :

TRUE : 正在 PAC 取得控制權後的第一個執行週期。

FALSE : 於其它執行週期。 *)

```
if is_first_cycle_just_after_switch then
```

(* 由於 Timer 並不會自動備援它的計時狀態，因此必須加入以下處理程序。

切換 PAC 後，若 Timer 的上一個狀態為 "ticking"，則在該 Cycle 中啟動計時。 *)

```
if TMR_1_last_state then
  tStart(TMR_1) ;
end_if ;
if TMR_2_last_state then
  tStart(TMR_2) ;
end_if ;
end_if ;
```

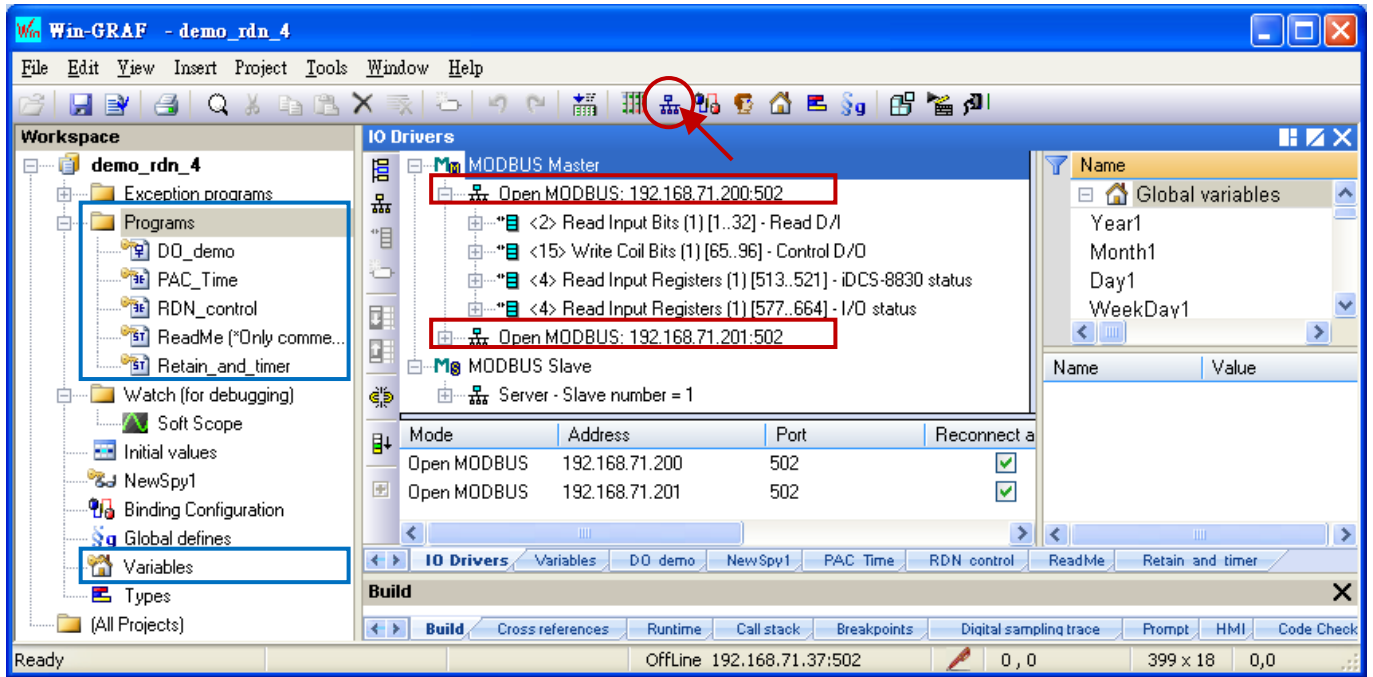
(* Timer 操作 *)

```
if To_tick_TMR_1 then
  To_tick_TMR_1 := FALSE ;
  tStart(TMR_1) ;
  TMR_1_last_state := TRUE ;
end_if ;
if To_tick_TMR_2 then
  To_tick_TMR_2 := FALSE ;
  tStart(TMR_2) ;
  TMR_2_last_state := TRUE ;
end_if ;
if To_stop_TMR_1 then
  To_stop_TMR_1 := FALSE ;
  tStop(TMR_1) ;
  TMR_1_last_state := FALSE ;
end_if ;
if To_stop_TMR_2 then
  To_stop_TMR_2 := FALSE ;
  tStop(TMR_2) ;
  TMR_2_last_state := FALSE ;
end_if ;
```

16.3.4 專案介紹 (demo_RDN_4)

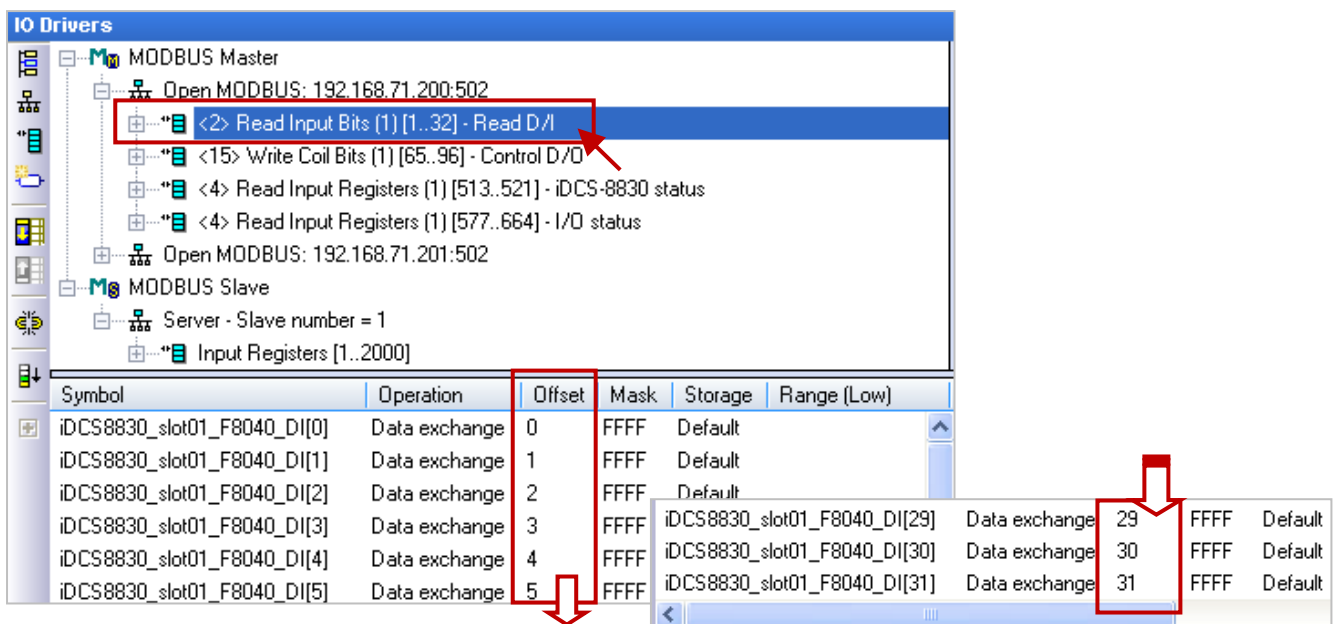
在 "demo_rdn_4" 專案中，您可點選程式名稱來查看內容，點選 "Variables" 查看使用的變數。另外，請參考 [16.3.1 節](#) 查看 I/O Board 設定 ("i_redundancy")，本章節將針對專案中的 "Modbus Master" 功能進行介紹 (詳細的操作步驟，與設定多個變數連續 Offset 的方法，請參考 [第 5 章](#))。

本範例使用的 iDCS-8830 在其 I/O Slot 編號 0、1 有插上兩張備援 DI 卡 (F-8040) 與在 I/O Slot 2、3 有插上兩張備援 DO 卡 (F-8041)。開始測試本範例前，請參考 [16.4.2 節](#) 將 iDCS-8830 先設定好，才能正確使用它。請點選 "Open Fieldbus Configuration" 按鈕來開啟 "I/O Drivers" 視窗。



此處我們啟用了 Modbus Master 來連接 2 個 Modbus TCP Slave 設備 (每台 iDCS-8830 備援 I/O 具有 2 個 IP 位址)，其 IP 位址為 192.168.71.200 與 192.168.71.201 (Port: 502)，並進行資料與狀態的讀/寫。

讀取 DI (使用 2 張備援 DI 卡 : F-8040 於 I/O 插槽 0、1)



注意: 本範例使用 iDCS-8830 備援 I/O，請參考 iDCS-8000 使用手冊 (CH4 Modbus Addresses Mapping) 來填寫 "Base address" 位址。

http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/fcm-mtcp_software_usermanual_en.pdf

MODBUS Master Request

Request
Description: Read D/I
Slave/Unit: 1

MODBUS Request
<1> Read Coil Bits
<2> Read Input Bits
<3> Read Holding Registers
<4> Read Input Registers

Data block
Base address: 1
Nb items: 32

Activation
 Periodic: 0 ms 3000
 On call
 On change (on error)

Misc.
Timeout: 1000 ms
Nb trials: 1

Annotations:
- Slave/Unit: 1
- Base address: 1, Nb items: 32: 由位址 1 開始讀取 32 個 DI 狀態。
- Periodic: 0 ms 3000: 一直發送。若出現異常，於 3 秒後再發送。
- Timeout: 1000 ms: 1 秒未回應，表示異常。

寫入 DO (使用 2 張備援 DO 卡 : F-8041 於 I/O 插槽 2、3)

此例，“Base address”由位址 65 開始寫入 32 個 DO 狀態，其餘如上圖。

IO Drivers

- MODBUS Master
 - Open MODBUS: 192.168.71.200:502
 - <2> Read Input Bits (1) [1..32] - Read D/I
 - <15> Write Coil Bits (1) [65..96] - Control D/O**
 - <4> Read Input Registers (1) [513..521] - iDCS-8830 status
 - <4> Read Input Registers (1) [577..664] - I/O status
 - Open MODBUS: 192.168.71.201:502
- MODBUS Slave
 - Server - Slave number = 1
 - Input Registers [1..2000]

Symbol	Operation	Offset	Mask	Storage
iDCS8830_slot23_F8041_DO[0]	Data exchange	0	FFFF	Default
iDCS8830_slot23_F8041_DO[1]	Data exchange	1	FFFF	Default
iDCS8830_slot23_F8041_DO[2]	Data exchange	2	FFFF	Default
iDCS8830_slot23_F8041_DO[3]	Data exchange	3	FFFF	Default
iDCS8830_slot23_F8041_DO[4]	Data exchange	4	FFFF	Default
iDCS8830_slot23_F8041_DO[5]	Data exchange	5	FFFF	Default
iDCS8830_slot23_F8041_DO[28]	Data exchange	28	FFFF	Default
iDCS8830_slot23_F8041_DO[29]	Data exchange	29	FFFF	Default
iDCS8830_slot23_F8041_DO[30]	Data exchange	30	FFFF	Default
iDCS8830_slot23_F8041_DO[31]	Data exchange	31	FFFF	Default

MODBUS Master Request

Request
Description: Control D/O
Slave/Unit: 1

MODBUS Request
<5> Write single coil bit
<6> Write single holding register
<15> Write Coil Bits
<16> Write Holding Registers

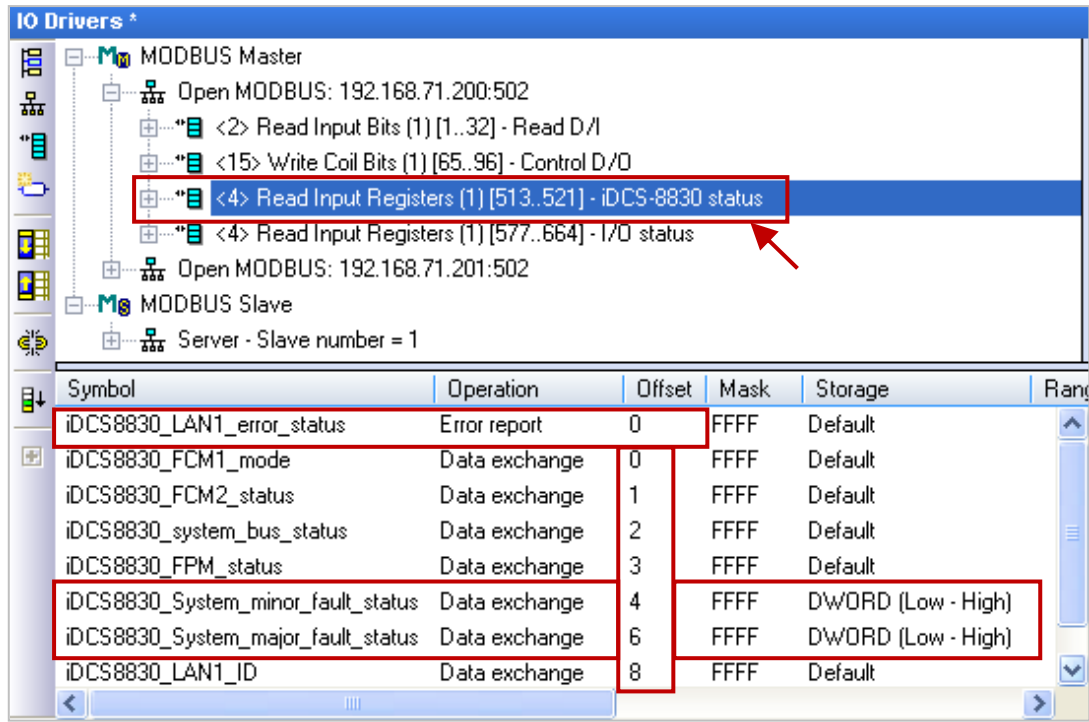
Data block
Base address: 65
Nb items: 32

Activation
 Periodic: 0 ms 3000
 On call
 On change (on error)

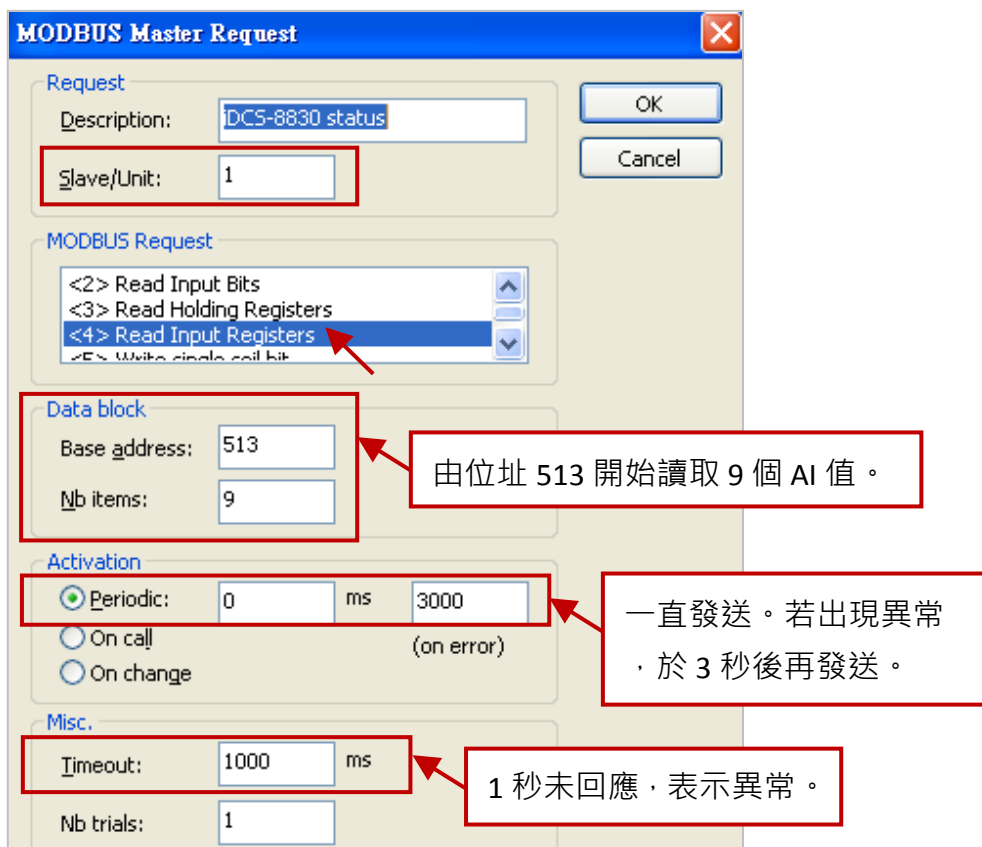
Misc.
Timeout: 1000 ms
Nb trials: 1

讀取 iDCS-8830 狀態

- "iDCS8830_LAN1_error_status" 的 "Operation" 設定為 "Error report" 表示讀取失敗時，該變數值為一個 "Error Code"，讀取成功時則會重置為 "0"。另外，"Offset" 需設定為 "0"。
- "iDCS8830_System_minor_fault_status" 與 "iDCS8830_System_major_fault_status" 的資料型態為 "DWORD" (32 bit)，"Offset" 需要使用 2 個位址且 "Storage" 需設定為 "DWORD (Low-High)"。



注意: 請參考 iDCS-8000 使用手冊 (CH4 Modbus Addresses Mapping) 來填寫 "Base address" 位址。
http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/fcm-mtcp_software_usermanual_en.pdf



讀取 iDCS-8830 的 I/O 狀態

Symbol	Operation	Offset	Mask	Storage	Ra
iDCS8830_io_slot_status[0]	Data exchange	16	FFFF	Default	
iDCS8830_io_slot_status[1]	Data exchange	17	FFFF	Default	
iDCS8830_io_slot_status[2]	Data exchange	18	FFFF	Default	
iDCS8830_io_slot_status[3]	Data exchange	19	FFFF	Default	
iDCS8830_io_slot_status[4]	Data exchange	20	FFFF	Default	
iDCS8830_io_slot_status[5]	Data exchange	21	FFFF	Default	
iDCS8830_io_slot_status[6]	Data exchange	22	FFFF	Default	
iDCS8830_io_slot_status[7]	Data exchange	23	FFFF	Default	
iDCS8830_io_emergency_status[0]	Data exchange	32	FFFF	Default	
iDCS8830_io_emergency_status[1]	Data exchange	33	FFFF	Default	
iDCS8830_io_emergency_status[2]	Data exchange	34	FFFF	Default	
iDCS8830_io_emergency_status[3]	Data exchange	35	FFFF	Default	
iDCS8830_io_emergency_status[4]	Data exchange	36	FFFF	Default	
iDCS8830_io_emergency_status[5]	Data exchange	37	FFFF	Default	
iDCS8830_io_emergency_status[6]	Data exchange	38	FFFF	Default	
iDCS8830_io_emergency_status[7]	Data exchange	39	FFFF	Default	
iDCS8830_io_channel_break_status[0]	Data exchange	72	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[1]	Data exchange	74	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[2]	Data exchange	76	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[3]	Data exchange	78	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[4]	Data exchange	80	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[5]	Data exchange	82	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[6]	Data exchange	84	FFFF	DWORD (Low - High)	
iDCS8830_io_channel_break_status[7]	Data exchange	86	FFFF	DWORD (Low - High)	

資料型態為 "DWORD" (32 bit) · 需要占用 2 個 Modbus 位址。

注意:

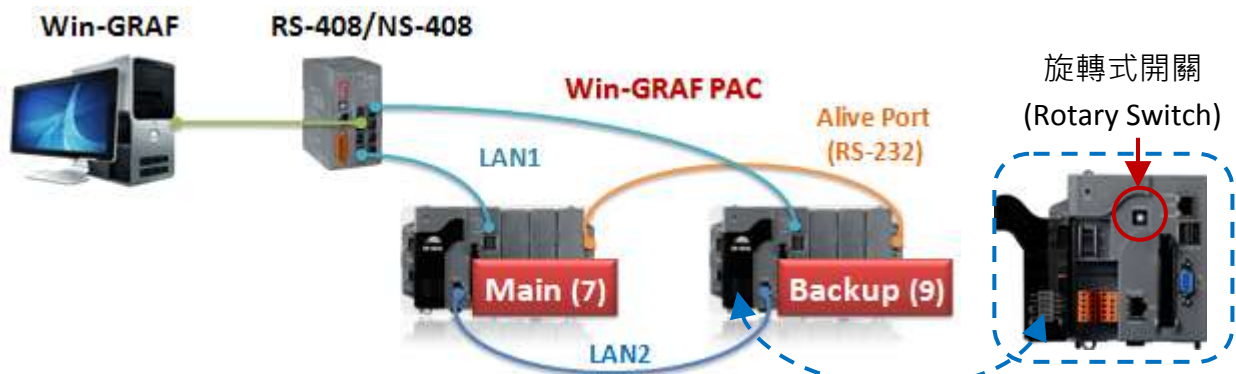
請參考 iDCS-8000 使用手冊 (CH4 Modbus Addresses Mapping) 來填寫 "Base address" 位址。
http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/fcm-mtcp_software_usermanual_en.pdf
 此例，由位址 577 開始讀取 88 個 AI 值。

以上關於 Modbus Master 的詳細設定方式，可參考第 5 章；關於 Modbus Slave 設定請參考第 3 章。

16.4 測試範例程式

16.4.1 測試 demo_RDN_2 · demo_RDN_3

demo_RDN_2: 2 台 XP-8xx8-CE6 · 沒有連接任何 I/O 模組。



1. 裝置硬體設備 (以 XP-8xx8-CE6 為例):
請參考 [16.2 節](#) 的安裝注意事項，確認 PAC 的 3 個通訊 Port 已連接好，且 2 台 PAC 的旋轉開關分別設定為 7 (Main-PAC) 或 9 (Backup-PAC)。
2. 若本備援系統尚未下載任何備援程式 (即，無主控權切換程序)，請先啟動 Main-PAC (7) 再啟動 Backup-PAC (9)，則主控權 (Active) 會在 Main-PAC。
(後續如需判斷主控權在哪一台，可開啟 PAC 螢幕上的 Win-GRAF Driver 來查看。)

PAC 端:

Win-GRAF Driver

- (1) 請先查看目前 Active PAC 硬體的 LAN1 IP (出廠預設 IP=192.168.255.1, Mask=255.255.255.0) 並確認與 PC 是在相同的網段 (例如: IP=192.168.255.x)，才可正常連線。(參考 [1.3 節](#))

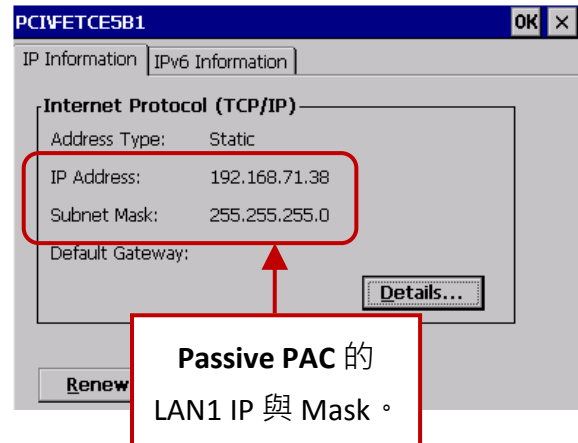
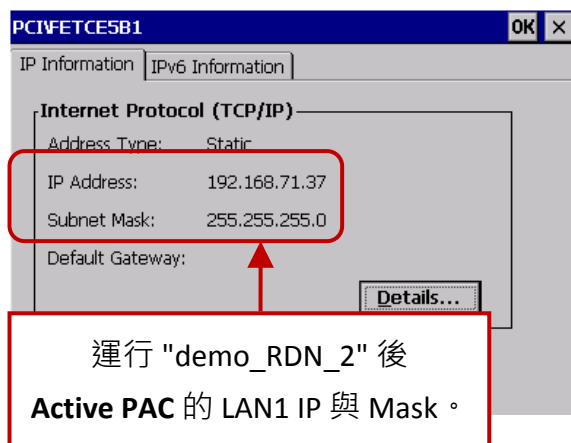
Active PAC 的 LAN1 IP 與 Mask。

- (2) 第一次下載備援應用程式時，請將此 "demo_RDN_2" 程式中的通訊 IP (參考 2.3.5 節的 "Communication Parameters") 改為目前 Active PAC 硬體的 LAN1 IP。

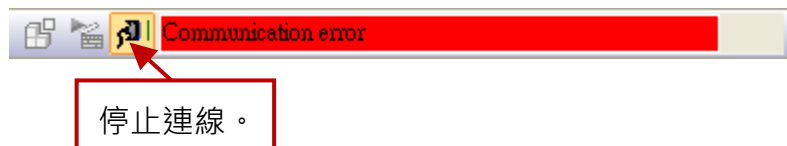


註: 如需設定通訊的 Timeout 時間 (預設: 3 秒) , 請參考 2.3.5 節。(例如: 設定 IP 為 "192.168.255.1:502(10)" , 表示 Timeout 為 10 秒。)

3. 再重新編譯程式並下載到 Active PAC 中 (若不熟悉操作，可參考 2.3.4 節、2.3.5 節)。下載後，Active PAC 的 LAN1 會設定為 Active IP (此例為 192.168.71.37，可參考 16.3.1 節)，而 Passive PAC 的 LAN1 會設成 Active IP + 1 (即，192.168.71.38)。




4. 此時，程式會顯示 "Communication error"，因為通訊 IP 與連線的 PAC IP 在不同的網段。請停止連線並將此 "demo_RDN_2" 程式中的通訊 IP (參考 2.3.5 節的 "Communication Parameters") 改回 "192.168.71.37"，同時，確認 PC 在相同的 IP 網段 (例如: IP=192.168.71.x)，往後如需下載程式，皆會下載到 Active 那台 PAC。

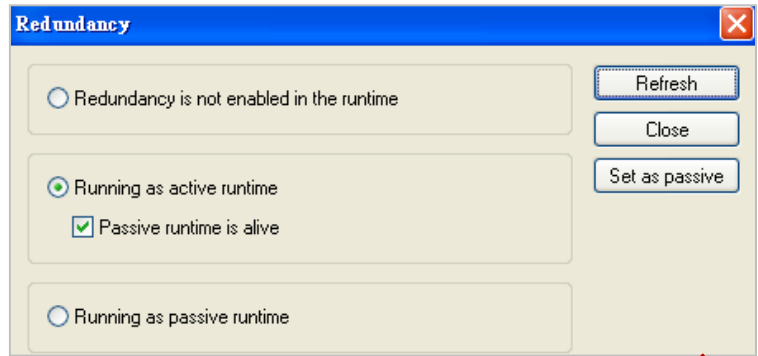


註: 如需設定通訊的 Timeout 時間 (預設: 3 秒) , 請參考 2.3.5 節。(例如: 設定 IP 為 "192.168.71.37:502(10)" , 表示 Timeout 為 10 秒。)



5. 連線後，點選 "NewSpy1" 開啟觀測清單，下圖顯示目前主控的 (Active) 是 Main-PAC。

注意：若 Passive PAC 尚未就緒 (即，`is_Passive_ready = FALSE`)，此時不能切換 PAC 的控制權，因為 Passive PAC 會無法接收到 Active PAC 的備援資料。另外，您也可點選 "Redundancy" 工具按鈕 () 來切換主控權。



Name	Value	Description
Hour1	12	
Minute1	35	
Second1	21	
is_Main_Active	TRUE	
is_Backup_Active	FALSE	
is_Main_ready	TRUE	
is_Backup_ready	TRUE	
is_first_cycle_just_after_switch	FALSE	
is_Main_LAN1_ok	TRUE	
is_Backup_LAN1_ok	TRUE	
is_Alive_port_ok	TRUE	
is_Passive_ready	TRUE	
is_Active_LAN1_ok	TRUE	
is Passive LAN1 ok	TRUE	
DINT_1	0	
DINT_2	0	
REAL_1	0.0	
REAL_2	0.0	
TMR_1	t#0s	
TMR_1_last_state	FALSE	TRUE: ticking , FALSE: sleep
To_tick_TMR_1	FALSE	Set TRUE to start ticking timer1
To_stop_TMR_1	FALSE	Set TRUE to stop the ticking of timer1
TMR_2	t#0s	
TMR_2_last_state	FALSE	TRUE: ticking , FALSE: sleep
To_tick_TMR_2	FALSE	Set TRUE to start ticking timer2
To_stop_TMR_2	FALSE	Set TRUE to stop the ticking of timer2

6. 請將 "DINT_1"、"DINT_2"、"REAL_1" 與 "REAL_2" 設定任意值，再分別將 "To_tick_TMR_1" 與 "To_tick_TMR_2" 設定為 "TRUE" (會自動重置為 FALSE) 以開始計時 TMR_1 或 TMR_2，此時，TIMER 的狀態會由 FALSE 變成 TRUE。

DINT_1	5	Setup as Retain variable in the program "Retain_and_timer"
DINT_2	1234	Setup as Retain variable in the program "Retain_and_timer"
REAL_1	2.5	variable in the program "Retain_and_timer"
REAL_2	99.5	variable in the program "Retain_and_timer"
TMR_1	t#18s947ms	
TMR_1_last_state	TRUE	TRUE: ticking , FALSE: sleep
To_tick_TMR_1	FALSE	Set TRUE to start ticking timer1
To_stop_TMR_1	FALSE	Set TRUE to stop the ticking of timer1
TMR_2	t#14s355ms	
TMR_2_last_state	TRUE	TRUE: ticking , FALSE: sleep
To_tick_TMR_2	FALSE	Set TRUE to start ticking timer2

7. 確認 Passive PAC 已就緒後 (即，is_Passive_Ready 為 TRUE)，將 Main-PAC 的 LAN1 拔除 或將 Main-PAC 關電後再開電，並等待一段時間 (參考“RDN_control”程式)，Main-PAC 會自動重開機 並將主控權交給 Backup-PAC。此時，主控為 Backup PAC 而且可看到各變數的值仍為原先所設定的 值 且 Timer 仍維持先前的狀態持續計時中。

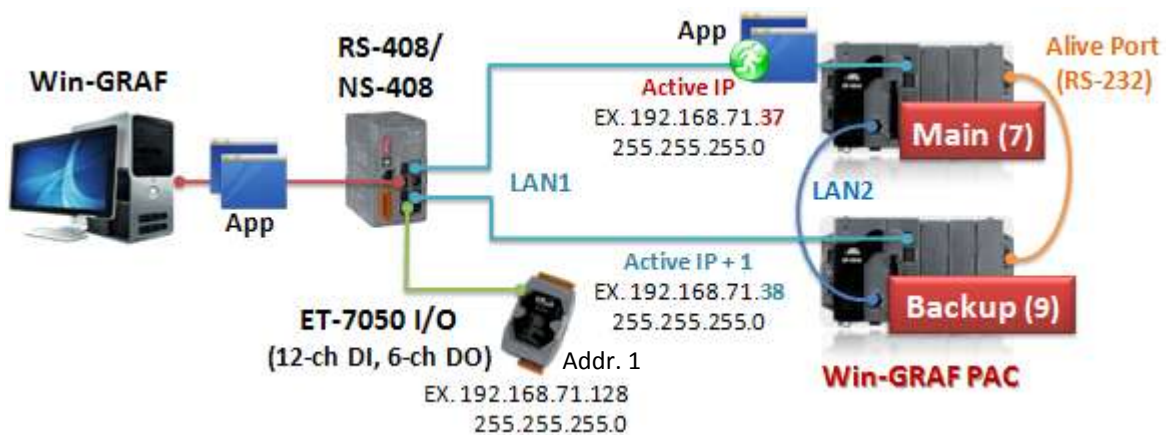
is_Main_Active	FALSE	
is_Backup_Active	TRUE	主控為 Backup PAC
is_Main_ready	TRUE	
is_Backup_ready	TRUE	
is_first_cycle_just_after_switch	FALSE	
is_Main_LAN1_ok	FALSE	
is_Backup_LAN1_ok	TRUE	
is_Alive_port_ok	TRUE	
is_Passive_ready	TRUE	
is_Active_LAN1_ok	TRUE	
is_Passive_LAN1_ok	FALSE	

DINT_1	5
DINT_2	1234
REAL_1	2.5
REAL_2	99.5
TMR_1	t#11m18s711ms
TMR_1_last_state	TRUE
To_tick_TMR_1	FALSE
To_stop_TMR_1	FALSE
TMR_2	t#11m14s119ms
TMR_2_last_state	TRUE
To_tick_TMR_2	FALSE

(註: 之後，再將 Main-PAC 的 LAN1 插上，則 "is_Main_LAN1" 與 "is_Passive_LAN1" 將回復為 TRUE。)

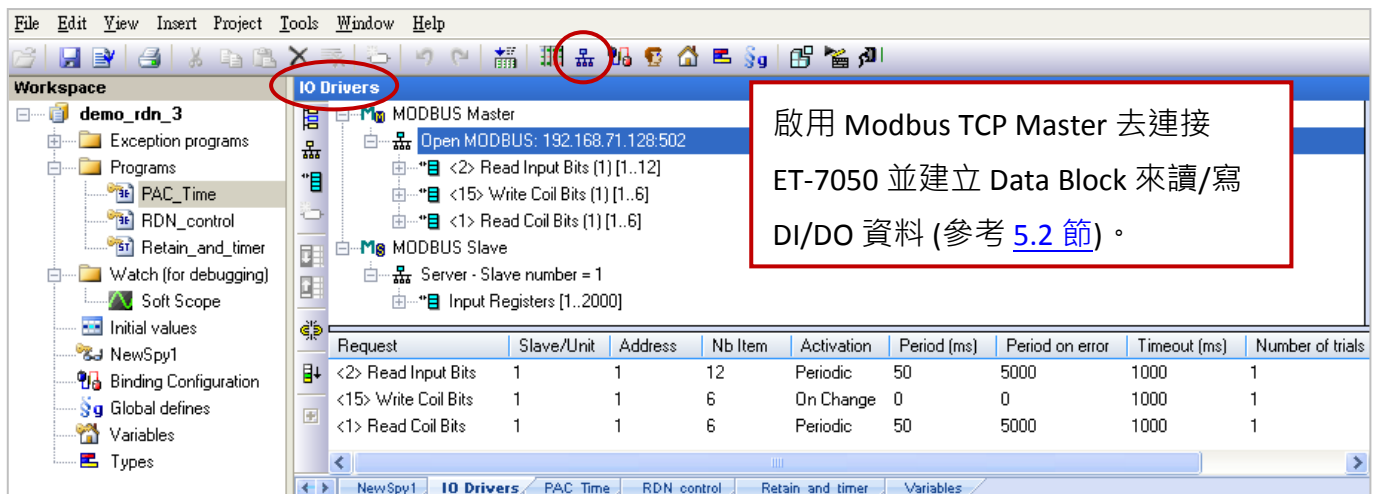
demo_RDN_3:

2 台 XP-8xx8-CE6，使用 LAN1 經過 Ethernet Switch 去連接一個 ET-7050 (Modbus TCP I/O 模組)。

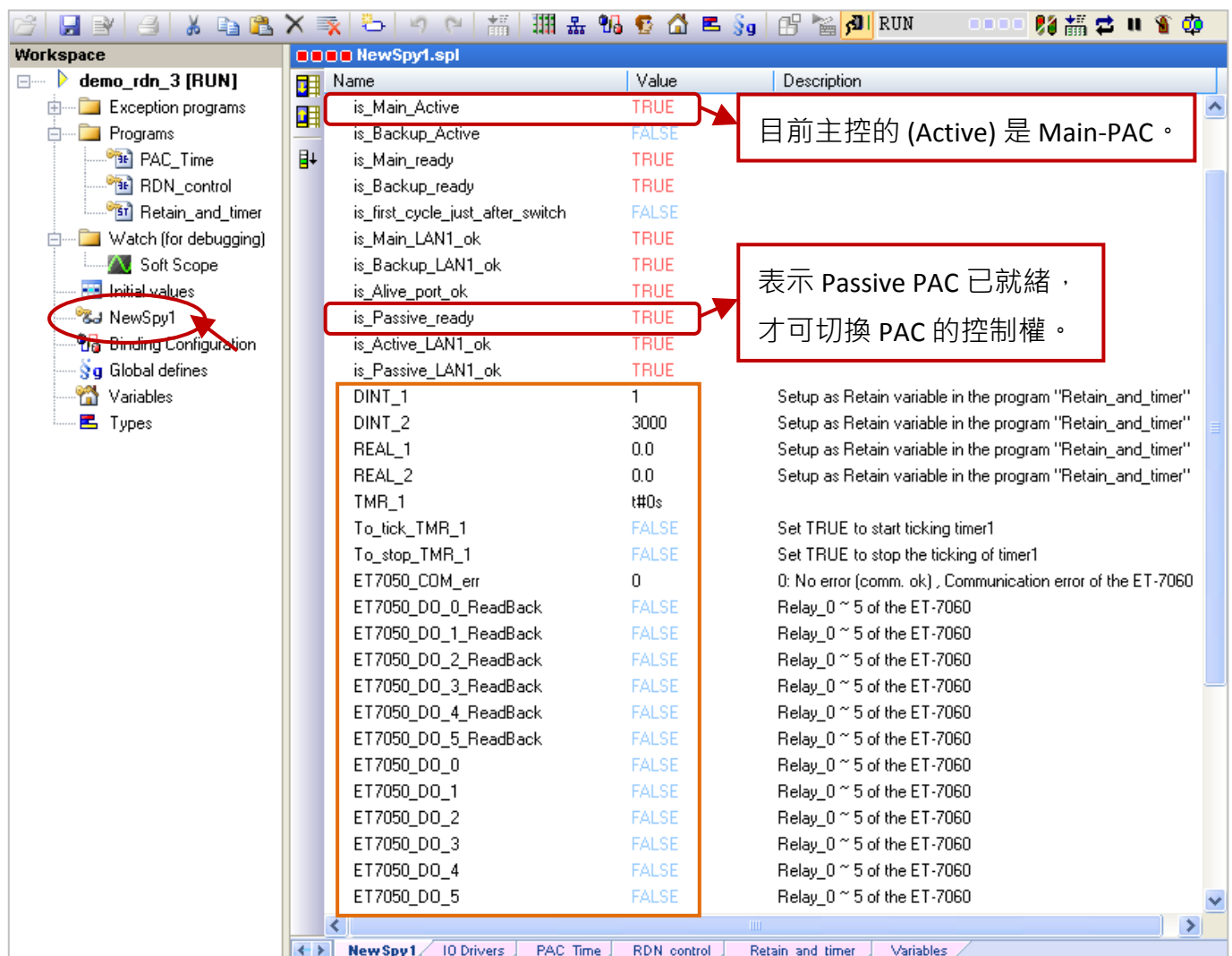


1. 請參考 ET-7000 手冊來設定 [ET-7050](#) 的 IP 位址與所需設定 (參考 [5.2.1 節](#))。
手冊: http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7000_et7200/document/
2. 開啟並下載 "demo_RDN_3" 程式，下載前，請將通訊 IP (參考 [2.3.5 節](#) 的 "Communication Parameters") 改為目前 Active PAC 硬體的 LAN1 IP (參考 [16.4 節](#) "demo_RDN_2" 程式測試步驟 2)。

"I/O Drivers" 視窗，此處啟用 Modbus TCP Master 去連接 ET-7050 (Modbus TCP Slave · Addr. = 1) 並建立 Data Block 來讀/寫 DI/DO 資料 (參考 5.2 節)，其它相關細節請參考範例程式。



3. 點選 "NewSpy1" 開啟觀測清單，目前主控的 (Active) 是 Main-PAC。



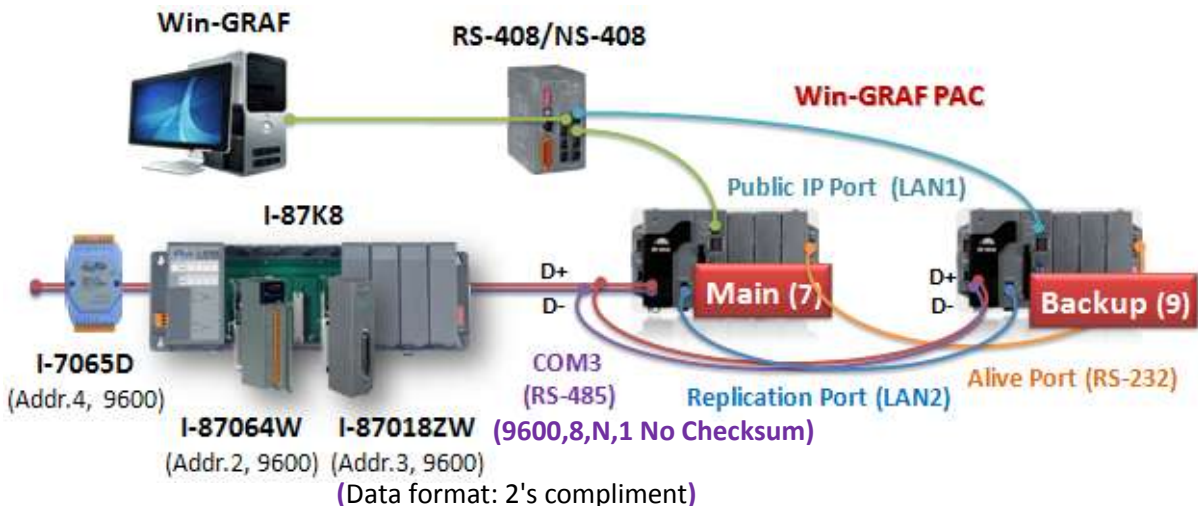
4. 您可試著修改變數值 (DINT_1, DINT_2, REAL_1, REAL_2) 的值，並將 "To_tick_TMR_1" 設定為 TRUE 來啟動 Timer 計時。若將 ET-7050_DOx 設定為 TRUE，則 ET-7050_DOx_ReadBack 會回傳 TRUE。若將 ET-7050 的網路線移除，ET-7050_COM_error 將會回傳非 0 的值，表示有錯誤。

5. 確認 Passive PAC 已就緒後 (即，is_Passive_Ready 為 TRUE) ，將 Main-PAC 的 LAN1 拔除，或將 Main-PAC 關電後再開電，並等待一段時間 (參考 “RDN_control” 程式) ，Main-PAC 會自動重開機並將主控權交給 Backup-PAC 。

Name	Value	Description
is_Main_Active	FALSE	
is_Backup_Active	TRUE	主控為 Backup PAC 。
is_Main_ready	TRUE	
is_Backup_ready	TRUE	
is_first_cycle_just_after_switch	FALSE	
is_Main_LAN1_ok	FALSE	之後，將 Main-PAC 的 LAN1 插上，來回復為 TRUE 。
is_Backup_LAN1_ok	TRUE	
is_Alive_port_ok	TRUE	
is_Passive_ready	TRUE	
is_Active_LAN1_ok	TRUE	
is_Passive_LAN1_ok	FALSE	

DINT_1	9	Setup as Retain variable in the program "Retain_and_timer"
DINT_2	3700	
REAL_1	3.9	主控切換為 Backup PAC 後，各變數仍為原先的值且 Timer 仍維持計時中。
REAL_2	5.8	
TMR_1	t#5m9s526ms	
To_tick_TMR_1	FALSE	Set TRUE to start ticking timer1
To_stop_TMR_1	FALSE	Set TRUE to stop the ticking of timer1
ET7050_COM_err	130	0: No error (comm. ok) . Communication error of the ET-7060
ET7050_DO_0_ReadBack	TRUE	非 0 表示有錯誤。
ET7050_DO_1_ReadBack	FALSE	
ET7050_DO_2_ReadBack	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_3_ReadBack	TRUE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_4_ReadBack	FALSE	
ET7050_DO_5_ReadBack	TRUE	將 ET-7050_DOx 設定為 TRUE，則 ET-7050_DOx_ReadBack 會回傳 TRUE。
ET7050_DO_0	TRUE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_1	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_2	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_3	TRUE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_4	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_DO_5	TRUE	Relay_0 ~ 5 of the ET-7060

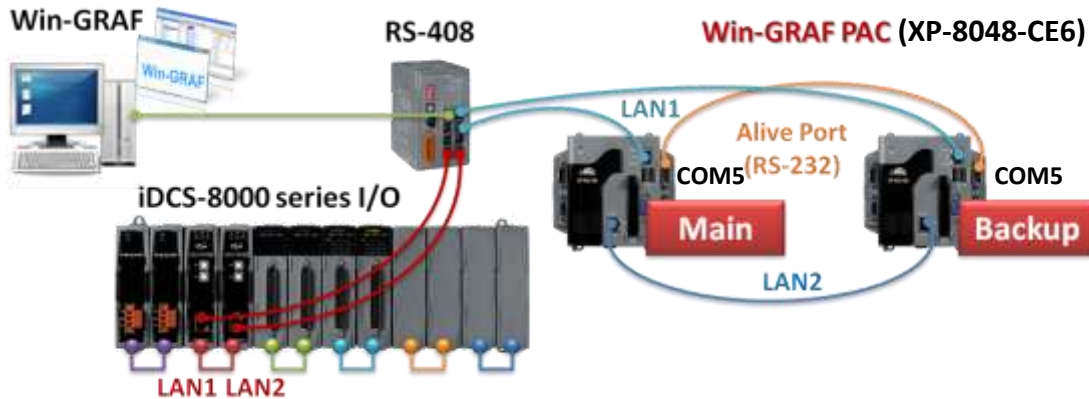
demo_RDN_1: 2 台 XP-8x8-CE6，使用 COM3 去連 3 個 DCON I/O 模組。



於 "demo_RDN_1" 範例程式，開始使用 "I-7000" 或 "I-87KW" 遠端 DCON I/O 模組前，需先以 "DCON Utility" 軟體設定每一個模組。請參考 [第 8 章](#) 說明並至 "DCON Utility" 網頁，下載軟體與使用手冊: www.icpdas.com/products/dcon/introduction.htm，其它相關細節請回存並參考範例程式，其 "I/O Board" 設定: 可參考 [16.3.1 節](#) 說明。

16.4.2 測試 demo_RDN_4

2 台 XP-8048-CE6，使用 LAN1 經過 Ethernet Switch 去連接一台 iDCS-8830 (PAC 與 I/O 皆備援)。



下表中列出了此範例程式所使用的設備:

產品型號	數量	產品型號	數量
XP-8048-CE6	2	DN-DI-32DW	1
RS-408	1	DN-DO-16DR-A	1
iDCS-8830	1	DN-DO-16DR-B	1
F-8040	2	CA-3710AM (1M Cable) 或 CA-3720AM/30AM/50AM/100AM	4
F-8041	2		

接線方式 與 設定:

您可參考 [16.1 節](#)，[16.2 節](#) 來了解 Win-GRAF 備援系統的架構(4) 與 XP-8048-CE6 的接線方式。其中 iDCS-8830 由左到右分別是插入以下的模組，請先下載並安裝相關的 Utility 程式與手冊。

iDCS-8000 Utility: <http://ftp.icpdas.com/pub/cd/idcs-8000/utility/> (軟體安裝: Ch2.2)

MiniOS7 Utility: <http://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/>

iDCS-8000 手冊/網頁:

<http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/>

http://www.icpdas.com/root/product/solutions/remote_io/dcs_redundancy_io/idcs_introduction.html

FPM-D2440 * 2: 電源模組 1、電源模組 2 (要接上 24V 電源輸入)

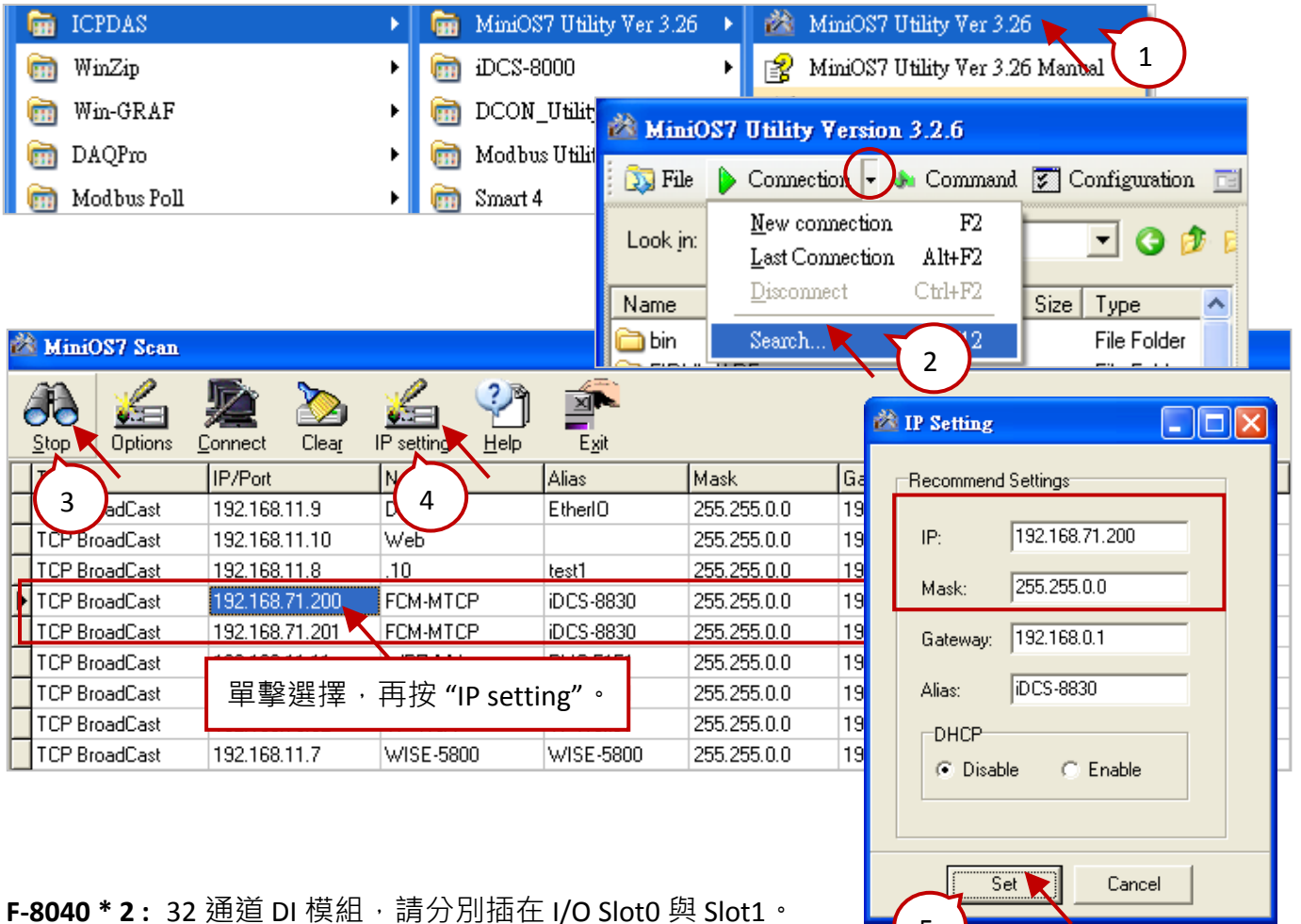
FCM-MTCP * 2: 主控模組 1 (先將 SW2 轉到 C, SW1 轉到 8，再去設定 IP 位址為 192.168.71.200)。

主控模組 2 (先將 SW2 轉到 C, SW1 轉到 9，再去設定 IP 位址為 192.168.71.201)。

SW2/SW1 表示 MCU (Main Control Unit) 的第 4 位 IP 位址 (C8₁₆ = 200 ; C9₁₆ = 201)，

請參考 iDCS-8000 手冊 - Ch2.3 或下圖，使用 MiniOS7 Utility 設定其 IP 位址。

開啟 MiniOS7 Utility，並點選“Search”來找到 iDCS-8830 上的 2 組 IP 設定，找到後先按“Stop”，待停止搜尋後，再依此範例更改這 2 個 IP 為 192.168.71.200 與 192.168.71.201，更改 Mask 為 255.255.0.0，點選“Set”後，關閉 MiniOS7 Utility。

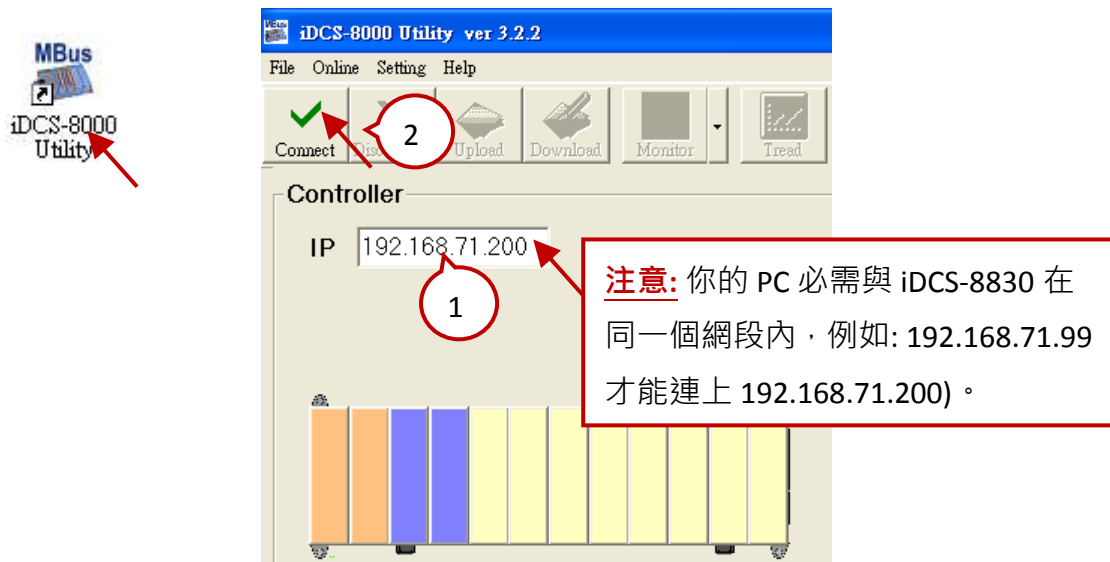


F-8040 * 2：32 通道 DI 模組，請分別插在 I/O Slot0 與 Slot1。

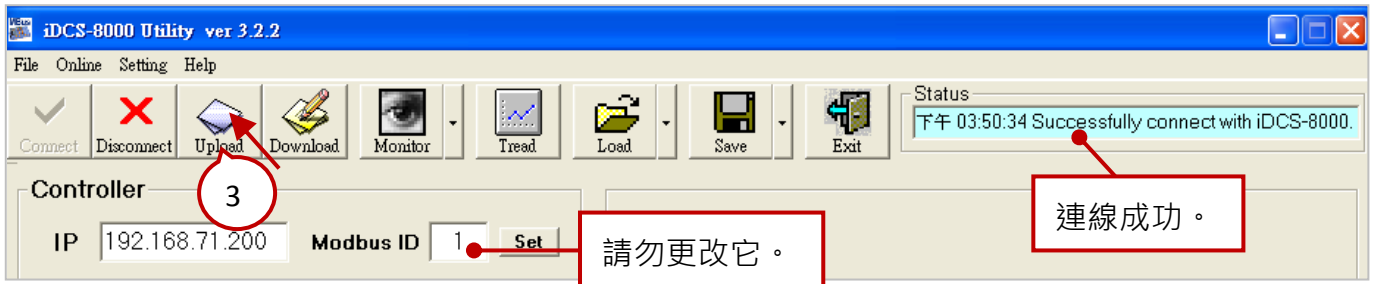
F-8041 * 2：32 通道 DO 模組，請分別插在 I/O Slot2 與 Slot3。

(請參考 iDCS-8000 手冊 - Ch.2.3 或下圖，使用 iDCS-8000 Utility 設定 I/O 模組。)

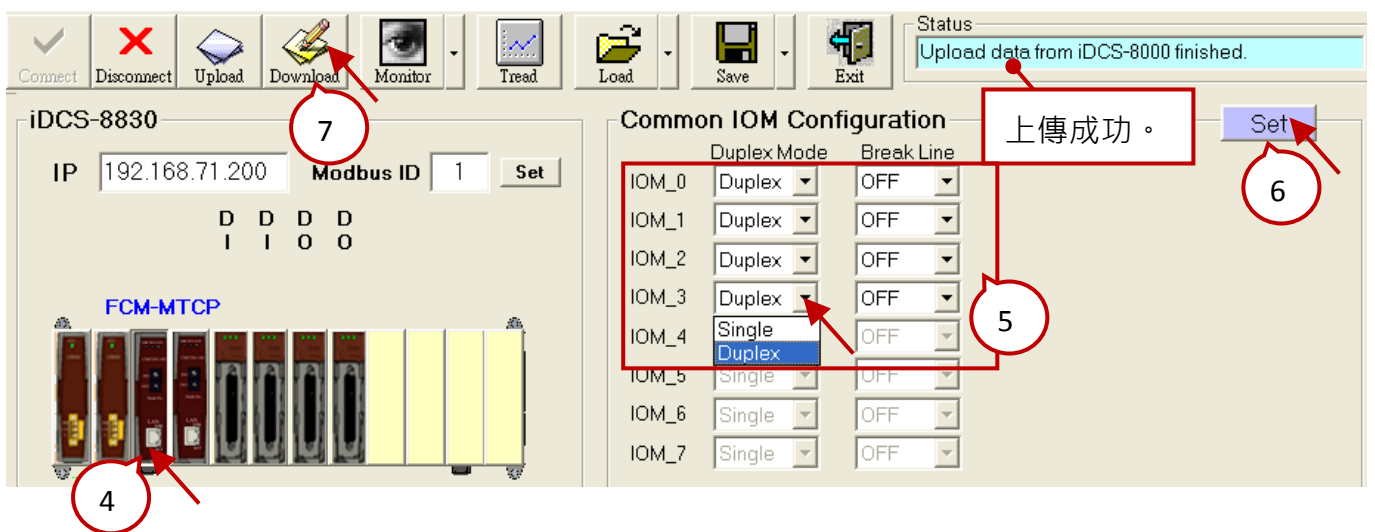
開啟 iDCS-8000 Utility，並輸入 iDCS-8830 的 IP 位址進行連線。



點選 "Upload" 將目前 iDCS-8830 內已插上的 I/O 設定傳送上來。



點選第 1 個 FCM-MTCP (MCU1) 並設定 F-8040, F-8041 (Slot 0 ~ 3) 為 "Duplex" Mode (Break Line 為斷線偵測，如果 F-8040, F-8041 尚未連接下列端子板，請先設定為 "OFF")，接著點選 "Set" 再點選 "Download" 將以上的設定下載到 iDCS-8830 內，即可關閉 iDCS-8000 Utility。



若您已經接好下列端子板，請將 "Break Line" (上圖) 設定為 "ON"。

DN-DI-32DW * 1: 32 通道 DI 端子板，請分別接到 F-8040 DI 模組 (Slot0, Slot1)。

DN-DO-16DR-A * 1: 16 通道 DO 端子板 (用於通道 0 ~ 15)

DN-DO-16DR-B * 1: 16 通道 DO 端子板 (用於通道 16 ~ 31)

請將 DN-DO-16DR-A (CN1, CN2) 分別接到 F-8041 DO 模組 (Slot2, Slot3)，
再將 DN-DO-16DR-A (CN3) 與 DN-DO-16DR-B (CN1) 相連接。

CA-3710AM * 4: 1 米 37-pin Male-Female D-sub 連接線，用於連接 I/O 模組與端子板。

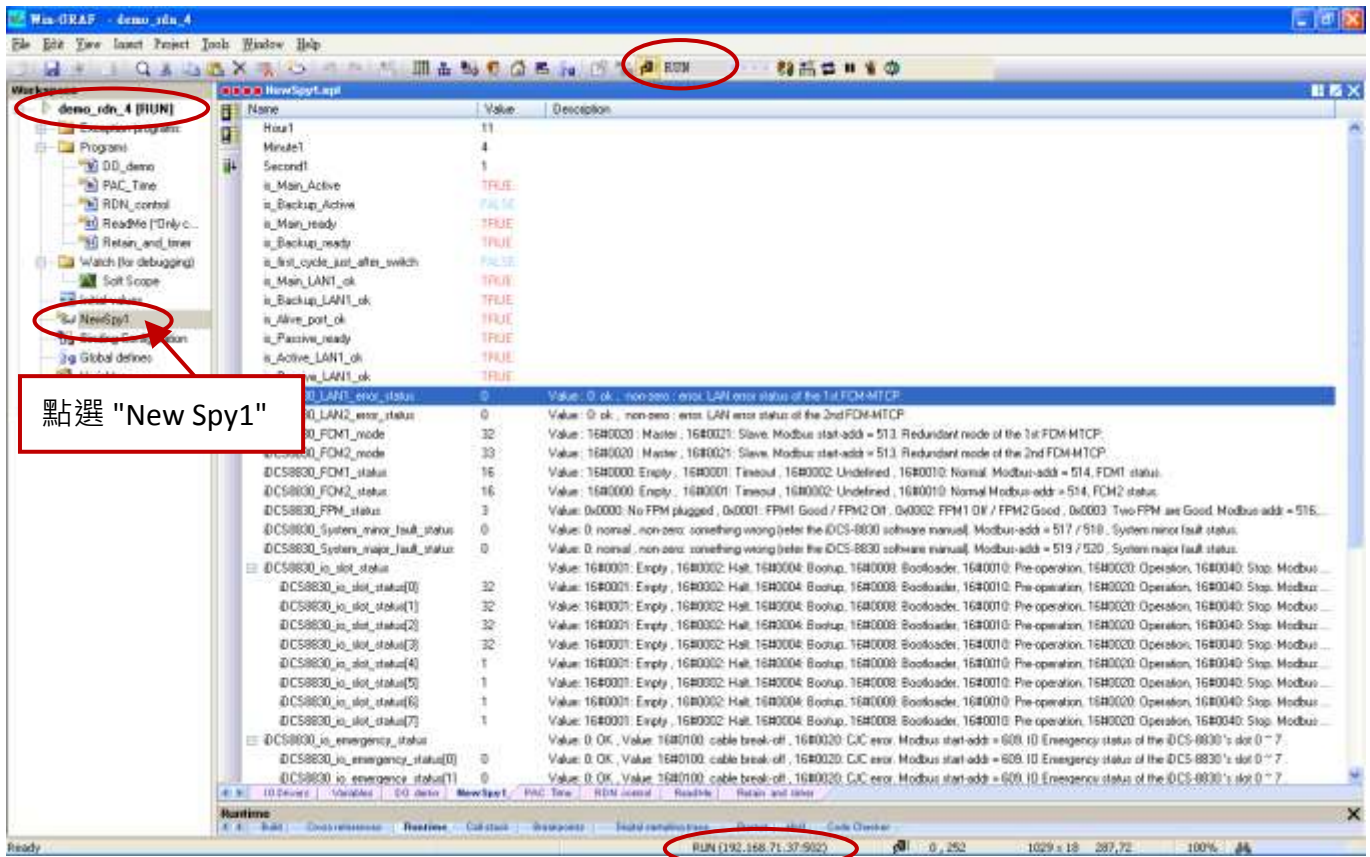
開始測試：

首先，請務必先至 [16.4.1 節](#) 了解相關 IP 設定，Win-GRAF 專案 ("demo_RDN_4") 的下載方式與 PAC 備援的測試步驟。以下將針對 iDCS-8830 的部分進行說明，開始前，請確認所有設備都已經連接好，再使用 Win-GRAF Workbench 連線至 PAC。

註： 連線時，請注意您 PC 與 PAC 的 IP 位址是在相同的網段。

(例如: PAC 的 IP = 192.168.71.37，Mask = 255.255.255.0，請設定您 PC 的 IP = 192.168.71.x)

連線後，請點選 "New Spy1" 開啟變數清單。



iDCS-8830 的網路通訊狀態:

變數名稱	iDCS8830_LAN1_error_status	iDCS8830_LAN2_error_status
測試方法	1. 拔除 (或插上) 第 1 個 FCM-MTCP 的 LAN 接線，狀態值: 非 0 (或 0)。 2. 拔除 (或插上) 第 2 個 FCM-MTCP 的 LAN 接線，狀態值: 非 0 (或 0)。	
狀態值	0: 通訊正常; 非 0: 通訊異常	
	iDCS8830_LAN1_error_status 130	Value: 0: ok, non-zero: error. LAN error status of the 1st FCM-MTCP.
	iDCS8830_LAN2_error_status 0	Value: 0: ok, non-zero: error. LAN error status of the 2nd FCM-MTCP.

iDCS-8830 的電源狀態:

變數名稱	iDCS8830_FPM_status	
測試方式	1. 拔除 (或插上) 第 1 個 FPM-D2440 的電源接線，狀態值: 2 (或 3)。 2. 拔除 (或插上) 第 2 個 FPM-D2440 的電源接線，狀態值: 1 (或 3)。	
狀態值	0: 2 個 FPM 皆未接好。	1: FPM 1 電源正常; FPM 2 電源異常。
	2: FPM 1 電源異常; FPM 2 正常。	3: 2 個 FPM 電源正常。
	iDCS8830_LAN1_error_status 0	Value: 0: ok, non-zero: error. LAN error status of the 1st FCM-MTCP.
	iDCS8830_LAN2_error_status 0	Value: 0: ok, non-zero: error. LAN error status of the 2nd FCM-MTCP.
	iDCS8830_FCM1_mode 32	Value: 16#0020: Master, 16#0021: Slave. Modbus start-addr = 513. Redundant mode of the 1st FCM.
	iDCS8830_FCM2_mode 33	Value: 16#0020: Master, 16#0021: Slave. Modbus start-addr = 513. Redundant mode of the 2nd FCM.
	iDCS8830_FCM1_status 16	Value: 16#0000: Empty, 16#0001: Timeout, 16#0002: Undefined, 16#0010: Normal. Modbus-addr =
	iDCS8830_FCM2_status 16	Value: 16#0000: Empty, 16#0001: Timeout, 16#0002: Undefined, 16#0010: Normal Modbus-addr =
	iDCS8830_FPM_status 2	Value: 0x0000: No FPM plugged, 0x0001: FPM1 Good / FPM2 Off, 0x0002: FPM1 Off / FPM2 Good,

FCM-MTCP 的備援模式與狀態:

變數名稱	iDCS8830_FCM1_mode iDCS8830_FCM1_status	iDCS8830_FCM2_mode iDCS8830_FCM2_status																		
測試方式	拔除第 1 個 FCM-MTCP 模組 (FCM1)。																			
	<table border="1"> <tr><td>iDCS8830_LAN1_error_status</td><td>130</td></tr> <tr><td>iDCS8830_LAN2_error_status</td><td>0</td></tr> <tr><td>iDCS8830_FCM1_mode</td><td>32</td></tr> <tr><td>iDCS8830_FCM2_mode</td><td>32</td></tr> <tr><td>iDCS8830_FCM1_status</td><td>1</td></tr> <tr><td>iDCS8830_FCM2_status</td><td>16</td></tr> </table>	iDCS8830_LAN1_error_status	130	iDCS8830_LAN2_error_status	0	iDCS8830_FCM1_mode	32	iDCS8830_FCM2_mode	32	iDCS8830_FCM1_status	1	iDCS8830_FCM2_status	16	<table border="1"> <tr><td>Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.</td></tr> <tr><td>Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.</td></tr> <tr><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> <tr><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> </table>	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16
iDCS8830_LAN1_error_status	130																			
iDCS8830_LAN2_error_status	0																			
iDCS8830_FCM1_mode	32																			
iDCS8830_FCM2_mode	32																			
iDCS8830_FCM1_status	1																			
iDCS8830_FCM2_status	16																			
Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.																				
Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.																				
Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																				
Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																				
Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																				
Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																				
說明	<p>iDCS8830_LAN1_error_status = 130 表示 FCM1 通訊異常。</p> <p>iDCS8830_FCM1_mode = 32 · FCM1 被拔除，其備援模式停留在 32 (Master)。</p> <p>iDCS8830_FCM2_mode = 32 · FCM2 已接手，備援模式由 33 (Slave) 變為 32 (Master)。</p> <p>iDCS8830_FCM1_status = 1 表示 FCM1 已經逾時 (Timeout)。</p>																			
狀態值	<p>iDCS8830_FCM1_mode / iDCS8830_FCM2_mode 32 : Master 33: Slave</p> <p>iDCS8830_FCM1_status / iDCS8830_FCM2_status 0: 空的 1: 逾時 2: 未定義 16: 正常工作</p>																			
請插上 FCM1，其備援模式更新為 33 (Slave)，狀態變為 16 (正常)。																				
<table border="1"> <tr><td>iDCS8830_LAN1_error_status</td><td>0</td><td>Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.</td></tr> <tr><td>iDCS8830_LAN2_error_status</td><td>0</td><td>Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.</td></tr> <tr><td>iDCS8830_FCM1_mode</td><td>33</td><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>iDCS8830_FCM2_mode</td><td>32</td><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>iDCS8830_FCM1_status</td><td>16</td><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> <tr><td>iDCS8830_FCM2_status</td><td>16</td><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> </table>			iDCS8830_LAN1_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.	iDCS8830_LAN2_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.	iDCS8830_FCM1_mode	33	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	iDCS8830_FCM2_mode	32	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	iDCS8830_FCM1_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16	iDCS8830_FCM2_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16
iDCS8830_LAN1_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.																		
iDCS8830_LAN2_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.																		
iDCS8830_FCM1_mode	33	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																		
iDCS8830_FCM2_mode	32	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																		
iDCS8830_FCM1_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																		
iDCS8830_FCM2_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																		
您可試著測試拔除 FCM2，此時 FCM1 接手為 32 (Master)。																				
<table border="1"> <tr><td>iDCS8830_LAN1_error_status</td><td>0</td><td>Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.</td></tr> <tr><td>iDCS8830_LAN2_error_status</td><td>130</td><td>Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.</td></tr> <tr><td>iDCS8830_FCM1_mode</td><td>32</td><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>iDCS8830_FCM2_mode</td><td>33</td><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>iDCS8830_FCM1_status</td><td>16</td><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> <tr><td>iDCS8830_FCM2_status</td><td>1</td><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> </table>			iDCS8830_LAN1_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.	iDCS8830_LAN2_error_status	130	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.	iDCS8830_FCM1_mode	32	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	iDCS8830_FCM2_mode	33	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	iDCS8830_FCM1_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16	iDCS8830_FCM2_status	1	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16
iDCS8830_LAN1_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.																		
iDCS8830_LAN2_error_status	130	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.																		
iDCS8830_FCM1_mode	32	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																		
iDCS8830_FCM2_mode	33	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																		
iDCS8830_FCM1_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																		
iDCS8830_FCM2_status	1	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																		
再將 FCM2 插上。																				
<table border="1"> <tr><td>iDCS8830_LAN1_error_status</td><td>0</td><td>Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.</td></tr> <tr><td>iDCS8830_LAN2_error_status</td><td>0</td><td>Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.</td></tr> <tr><td>iDCS8830_FCM1_mode</td><td>32</td><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>iDCS8830_FCM2_mode</td><td>33</td><td>Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R</td></tr> <tr><td>iDCS8830_FCM1_status</td><td>16</td><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> <tr><td>iDCS8830_FCM2_status</td><td>16</td><td>Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16</td></tr> </table>			iDCS8830_LAN1_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.	iDCS8830_LAN2_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.	iDCS8830_FCM1_mode	32	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	iDCS8830_FCM2_mode	33	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R	iDCS8830_FCM1_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16	iDCS8830_FCM2_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16
iDCS8830_LAN1_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP.																		
iDCS8830_LAN2_error_status	0	Value : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP.																		
iDCS8830_FCM1_mode	32	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																		
iDCS8830_FCM2_mode	33	Value : 16#0020: Master , 16#0021: Slave. Modbus start-addr = 513. R																		
iDCS8830_FCM1_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																		
iDCS8830_FCM2_status	16	Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16																		

I/O 插槽狀態 與 I/O 緊急狀態:

變數名稱	iDCS8830_io_slot_status	iDCS8830_io_emergency_status																																	
測試方式	拔除第 1 個 F-8040 模組上 (Slot 0) 與 DN-DI-32DW 連接的 Cable 。																																		
	<table border="0"> <tr> <td>Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008</td> <td></td> </tr> <tr> <td>iDCS8830_io_slot_status</td> <td></td> </tr> <tr> <td>iDCS8830_io_slot_status[0]</td> <td>64</td> </tr> <tr> <td>iDCS8830_io_slot_status[1]</td> <td>32</td> </tr> <tr> <td>iDCS8830_io_slot_status[2]</td> <td>32</td> </tr> <tr> <td>iDCS8830_io_slot_status[3]</td> <td>32</td> </tr> <tr> <td>iDCS8830_io_slot_status[4]</td> <td>1</td> </tr> <tr> <td>iDCS8830_io_slot_status[5]</td> <td>1</td> </tr> <tr> <td>iDCS8830_io_slot_status[6]</td> <td>1</td> </tr> <tr> <td>iDCS8830_io_slot_status[7]</td> <td>1</td> </tr> <tr> <td>Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008</td> <td></td> </tr> <tr> <td>iDCS8830_io_emergency_status</td> <td></td> </tr> <tr> <td>iDCS8830_io_emergency_status[0]</td> <td>256</td> </tr> <tr> <td>iDCS8830_io_emergency_status[1]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_emergency_status[2]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_emergency_status[3]</td> <td>0</td> </tr> <tr> <td>Value: 0: OK , Value: 16#0100: cable break-off , 16#0020: CJC error.</td> <td></td> </tr> </table>		Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008		iDCS8830_io_slot_status		iDCS8830_io_slot_status[0]	64	iDCS8830_io_slot_status[1]	32	iDCS8830_io_slot_status[2]	32	iDCS8830_io_slot_status[3]	32	iDCS8830_io_slot_status[4]	1	iDCS8830_io_slot_status[5]	1	iDCS8830_io_slot_status[6]	1	iDCS8830_io_slot_status[7]	1	Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008		iDCS8830_io_emergency_status		iDCS8830_io_emergency_status[0]	256	iDCS8830_io_emergency_status[1]	0	iDCS8830_io_emergency_status[2]	0	iDCS8830_io_emergency_status[3]	0	Value: 0: OK , Value: 16#0100: cable break-off , 16#0020: CJC error.
Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008																																			
iDCS8830_io_slot_status																																			
iDCS8830_io_slot_status[0]	64																																		
iDCS8830_io_slot_status[1]	32																																		
iDCS8830_io_slot_status[2]	32																																		
iDCS8830_io_slot_status[3]	32																																		
iDCS8830_io_slot_status[4]	1																																		
iDCS8830_io_slot_status[5]	1																																		
iDCS8830_io_slot_status[6]	1																																		
iDCS8830_io_slot_status[7]	1																																		
Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008																																			
iDCS8830_io_emergency_status																																			
iDCS8830_io_emergency_status[0]	256																																		
iDCS8830_io_emergency_status[1]	0																																		
iDCS8830_io_emergency_status[2]	0																																		
iDCS8830_io_emergency_status[3]	0																																		
Value: 0: OK , Value: 16#0100: cable break-off , 16#0020: CJC error.																																			
說明	<p>iDCS8830_io_slot_status[0] = 64 表示模組 (Slot0) 停止工作。</p> <p>iDCS8830_io_emergency_status[0] = 256 表示模組 (Slot0) 的 Cable 斷線。</p> <p>請插上 Cable，您可試著拔除其它模組上的 Cable 來進行測試。</p>																																		
狀態值	<p>iDCS8830_io_slot_status[x]</p> <p>1: 空的 2: 中止 4: 啟動 8: 啟動載入 16: 前置工作 32: 工作 64: 停止</p> <p>iDCS8830_io_emergency_status[x]</p> <p>0 : 正常 32: CJC 錯誤 256: 連接線斷開</p>																																		

註: 需先在 iDCS-8000 Utility 將 IOM_x 設定為 "Duplex" Mode，iDCS8830_io_emergency_status[x] 才有作用。

DO 模組備援:

變數名稱	iDCS8830_slot23_F8041_DO																																				
測試方式	拔除第 1 個 F-8041 DO 模組 (Slot2) 。																																				
	<table border="0"> <tr> <td>Value: 16#0001: Empty , 16#0002: Halt,</td> <td></td> </tr> <tr> <td>iDCS8830_io_slot_status</td> <td></td> </tr> <tr> <td>iDCS8830_io_slot_status[0]</td> <td>32</td> </tr> <tr> <td>iDCS8830_io_slot_status[1]</td> <td>32</td> </tr> <tr> <td>iDCS8830_io_slot_status[2]</td> <td>2</td> </tr> <tr> <td>iDCS8830_io_slot_status[3]</td> <td>32</td> </tr> <tr> <td>Value: 16#0001: Empty , 16#0002: Halt,</td> <td></td> </tr> </table> <table border="0"> <tr> <td>Value: F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830</td> <td></td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO</td> <td></td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[0]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[1]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[2]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[3]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[4]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[5]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[6]</td> <td>TRUE</td> </tr> <tr> <td>iDCS8830_slot23_F8041_DO[7]</td> <td>TRUE</td> </tr> <tr> <td>Value: F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830</td> <td></td> </tr> </table>		Value: 16#0001: Empty , 16#0002: Halt,		iDCS8830_io_slot_status		iDCS8830_io_slot_status[0]	32	iDCS8830_io_slot_status[1]	32	iDCS8830_io_slot_status[2]	2	iDCS8830_io_slot_status[3]	32	Value: 16#0001: Empty , 16#0002: Halt,		Value: F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830		iDCS8830_slot23_F8041_DO		iDCS8830_slot23_F8041_DO[0]	TRUE	iDCS8830_slot23_F8041_DO[1]	TRUE	iDCS8830_slot23_F8041_DO[2]	TRUE	iDCS8830_slot23_F8041_DO[3]	TRUE	iDCS8830_slot23_F8041_DO[4]	TRUE	iDCS8830_slot23_F8041_DO[5]	TRUE	iDCS8830_slot23_F8041_DO[6]	TRUE	iDCS8830_slot23_F8041_DO[7]	TRUE	Value: F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830
Value: 16#0001: Empty , 16#0002: Halt,																																					
iDCS8830_io_slot_status																																					
iDCS8830_io_slot_status[0]	32																																				
iDCS8830_io_slot_status[1]	32																																				
iDCS8830_io_slot_status[2]	2																																				
iDCS8830_io_slot_status[3]	32																																				
Value: 16#0001: Empty , 16#0002: Halt,																																					
Value: F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830																																					
iDCS8830_slot23_F8041_DO																																					
iDCS8830_slot23_F8041_DO[0]	TRUE																																				
iDCS8830_slot23_F8041_DO[1]	TRUE																																				
iDCS8830_slot23_F8041_DO[2]	TRUE																																				
iDCS8830_slot23_F8041_DO[3]	TRUE																																				
iDCS8830_slot23_F8041_DO[4]	TRUE																																				
iDCS8830_slot23_F8041_DO[5]	TRUE																																				
iDCS8830_slot23_F8041_DO[6]	TRUE																																				
iDCS8830_slot23_F8041_DO[7]	TRUE																																				
Value: F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830																																					
說明	<p>一開始 F-8041 (Slot2) 的 LED (DO0 ~ 7) 會依序亮燈，拔除該模組後，會立即切換到第 2 個 F-8041 (Slot3) 依序亮燈。此時，可見到 iDCS8830_io_slot_status[2] = 2 表示模組 (Slot2) 中止工作。(請再插上該模組。)</p>																																				

16.5 哪些資料會自動備援到 Passive PAC?

Win-GRAF PAC 備援系統並非會將 Active PAC 內的全部資料都自動備援到 Passive PAC。

以下的資料會自動備援:

1. 使用者寫的 Win-GRAF 應用程式。
2. 程式運行到哪裡。
3. 變數的數值。
4. 功能方塊的樣例變數 (FB instance) 裡的內部 (Private) 資料。
5. PAC 硬體的 RTC (Real Time Clock) 時間。
6. 可保存記憶體 (Retain Memory)。
7. 排程控制 (Schedule-control) 的設定檔 (參考第 17 章)。

以下是常見，不會自動備援到 Passive PAC 的項目:

1. Timer 變數的計時狀態 (Ticking 或 Sleep)。
2. Active PAC 內的 File (比如 \system_disk 或 \Micro_SD 內的 File 或其它非屬於 Win-GRAF 應用程式的檔案，如 C、VB.net、C# 或 eLogger 等應用程式檔，這些並不會自動備援。因此要安裝新的備品 PAC 或修好的 PAC 進去備援系統之前，需預先把這些 File 都安裝好)。
(不包含使用者寫的 Win-GRAF 應用程式與 Schedule-control 的設定檔，這兩個會自動備援。)
3. 使用者自行使用 COM_OPEN () 函式來 Open 序列埠 (Serial Port)，不會在切換 PAC 後自動再被 Open 起來。
4. PAC 的 EEPROM 記憶體不會自動備援。

不會自動備援的項目，使用者需使用類似以下的程式來處理:

(參考 "demo_RDN_2" 內的 "Retain_and_timer" 程式)

```
if is_first_cycle_just_after_switch then  
  
    (* 在接手控制權後的第一圈處理 *)  
  
    .....  
  
end_if;
```

第 17 章 排程控制 (Schedule Control)



簡介:

Win-GRAF WinCE 系列 PAC 皆支援排程控制功能 (Schedule Control)，一台 Win-GRAF PAC 可以控制最多 10 個控制設備 (Target) 的排程 (Schedule)。每個控制設備 (Target) 可控制 1 個布林 (BOOL)、1 個 32-bit 整數 (DINT) 與 1 個實數 (REAL) 共三個變數，ICP DAS 提供免費的排程工具程式 - "Schedule-Control Utility"，讓 User 在 PC 端與 PAC 端皆可快速簡單完成排程之設定與監控。

檢查 PAC Driver 版本:

下列 Win-GRAF PAC 驅動程式 (Driver) 版本起才有支援 排程控制功能:

WinCE 系列	Win-GRAF PAC	Driver 版本
WP-8000	WP-8148, WP-8448, WP-8848	1.02 版起
WP-8000-CE7	WP-8128-CE7, WP-8428-CE7, WP-8828-CE7	1.01 版起
WP-5000	WP-5238-CE7	
ViewPAC	VP-x2x8-CE7	
XP-8000-CE6	XP-8048-CE6, XP-8348-CE6, XP-8748-CE6	

若您的 PAC Driver 版本較舊，請先安裝最新版本。驅動程式下載網址:

http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-driver.html

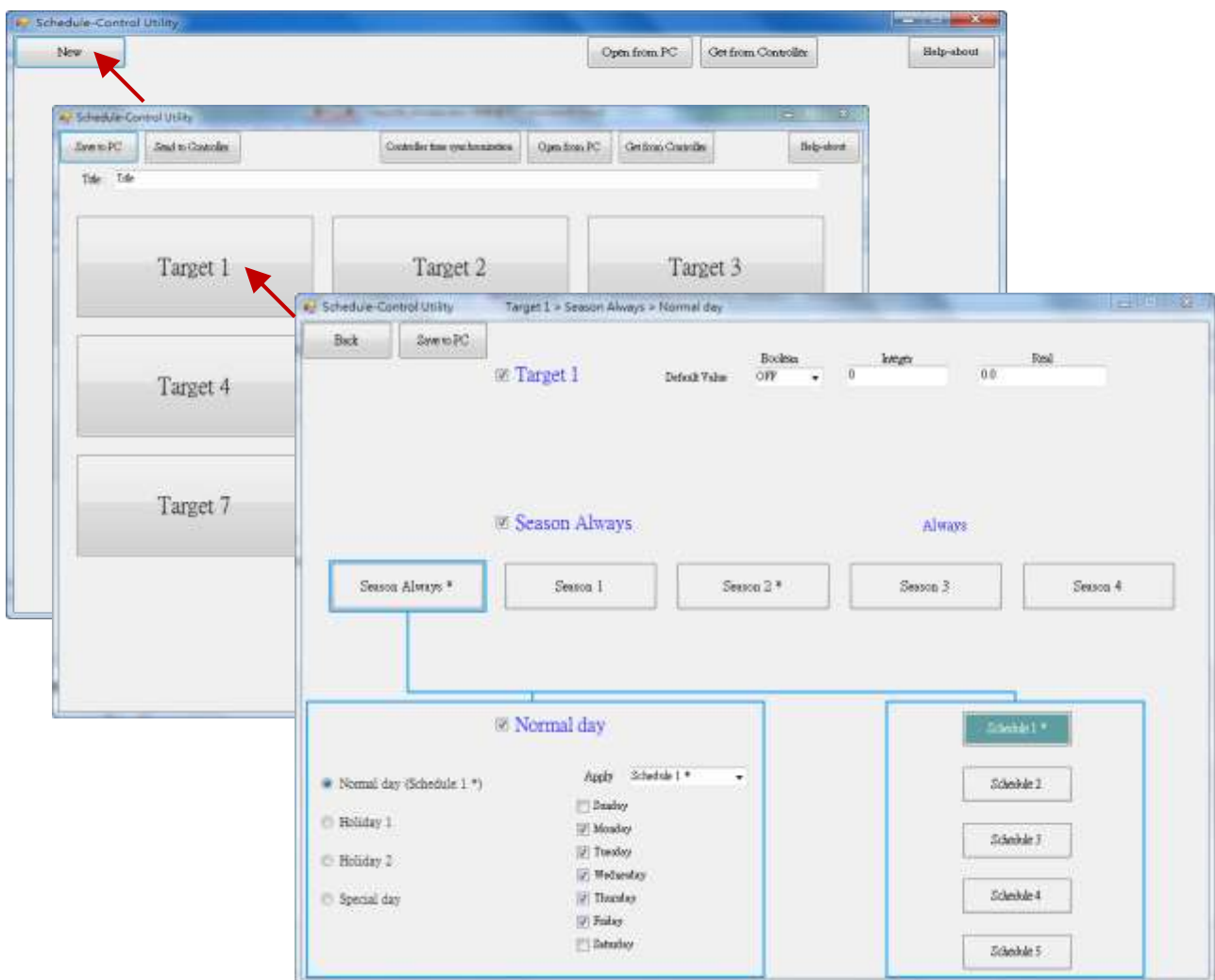
17.1 安裝 Schedule-Control Utility 與回存 Win-GRAF 範例程式

Win-GRAF PAC 包裝盒裡附有一張 Win-GRAF-PAC-CD，於路徑 CD:\napdos\Win-GRAF\Tools_Utility\ 內有一個 Schedule-Control 的工具程式，檔名為“Schedule_in_PC.exe”。

請將此 **Schedule-Control Utility** (Schedule_in_PC.exe 與 label_name.txt) 複製到您的 PC 上，建議存放在與 Win-GRAF 專案相同的目錄內 (例如，D:\Schedule-Control\Station1\Schedule_in_PC.exe)，然後直接執行該檔案。

接著，可見到 "Schedule-Control Utility" 操作畫面如下：

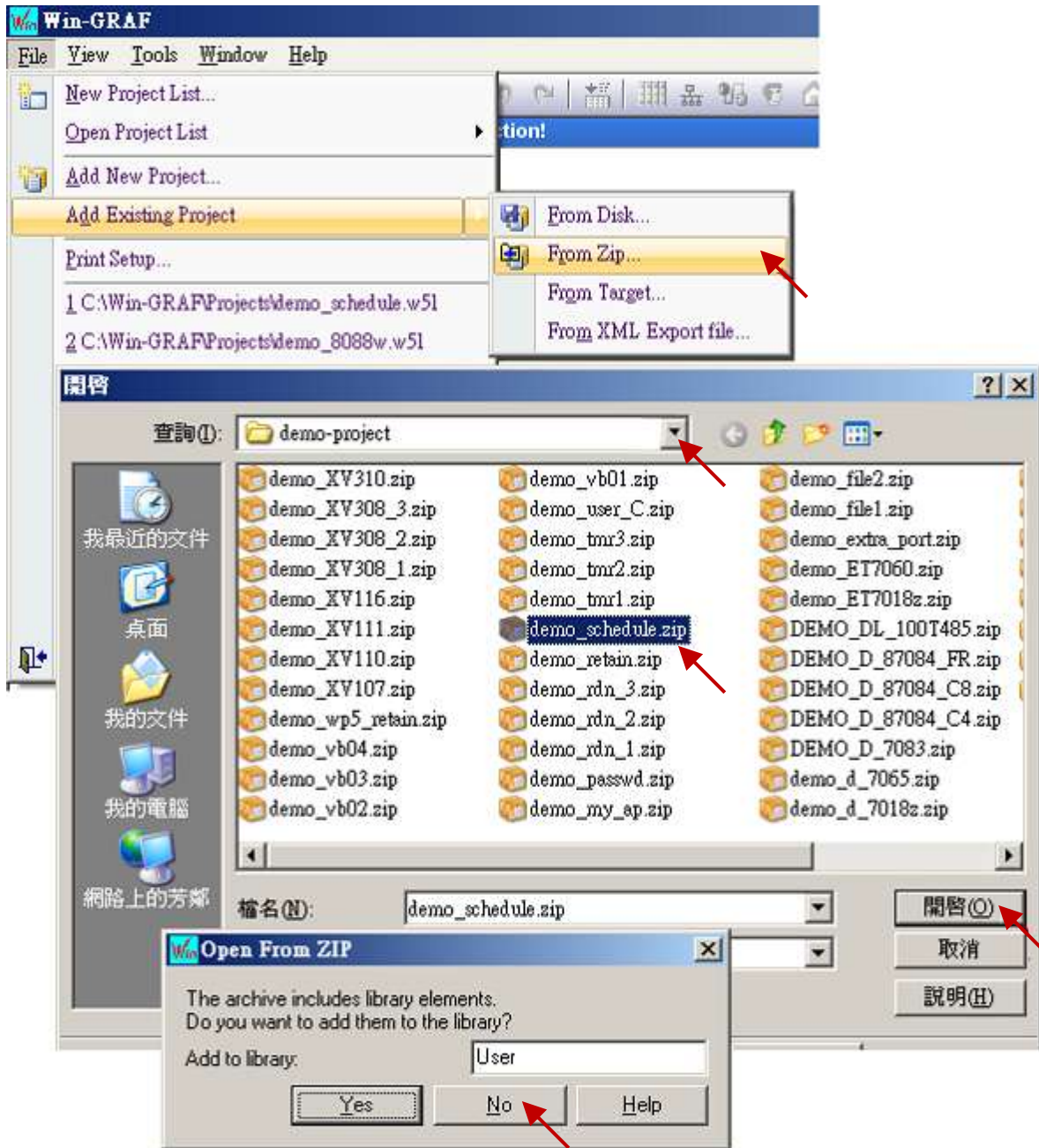
(點選 "New" 再點選 "Target 1")



在 Win-GRAF PAC 內也有一個排程控制的工具程式 (即，“Schedule_in_PAC.exe”)，您可在 PAC 的 \System_Disk\Win-GRAF\ 路徑內找到“Schedule_in_PAC.exe”。

回存 demo_schedule 範例程式:

在 Win-GRAF-PAC-CD 的路徑 \napdos\Win-GRAF\demo-project\ 內可以找到此排程控制的 “demo_schedule.zip” 範例程式，請依下圖將此 zip 檔回存到 Win-GRAF Workbench 內。



17.2 "demo_schedule" 範例程式說明

本章節將介紹排程控制的“demo_schedule”範例程式，請準備一台 Win-GRAF PAC (例如，WP-8448 或 VP-22x8-CE7)。一台 PAC 最多可設定 10 個“控制標的”(Target 1 ~ Target 10) 的排程表 (Schedule) 來控制 10 個設備，而每個 Target 可控制 1 個 BOOL 變數、1 個 DINT 變數與 1 個 REAL 變數。

I/O board 設定：

要在 Win-GRAF PAC 內啟用 Schedule-control，請先點選“Open I/Os”來加入一個“Schedule”(請加在 Slot 8 或更大的 Slot 編號內)。在其“Properties”視窗內有個“Password”參數可用來設定，當連線到 Win-GRAF PAC 時，供 PC 上 Run 的“Schedule-Control Utility”識別權限的密碼，此例設定為 0。在“I/O Boards”內啟用“Schedule”後，可在“Variables”視窗內看到它有 10 個 BOOL 輸入通道，它會回傳 Target 1 ~ Target 10 使用排程控制的狀態，TRUE 表示“Schedule-Control Utility”有啟用該 Target 的排程控制，FALSE 表示未啟用。

The screenshot displays the Win-GRAF software interface with several windows and annotations:

- 1.** Points to the "Open I/Os" button in the "Binding" toolbar.
- 2.** Points to the "Schedule" entry in the "I/O Boards" list, with a callout box: "滑鼠雙擊，加入 'Schedule'" (Double-click mouse, add 'Schedule').
- 3.** Points to the "Properties" button for the selected "Schedule" board.
- 4.** Points to the "Password" field in the "10: Schedule - Properties" dialog, which is set to 0.
- 5.** Points to the "Variables" window, which shows a list of variables for the "%IX10 - Schedule" board.

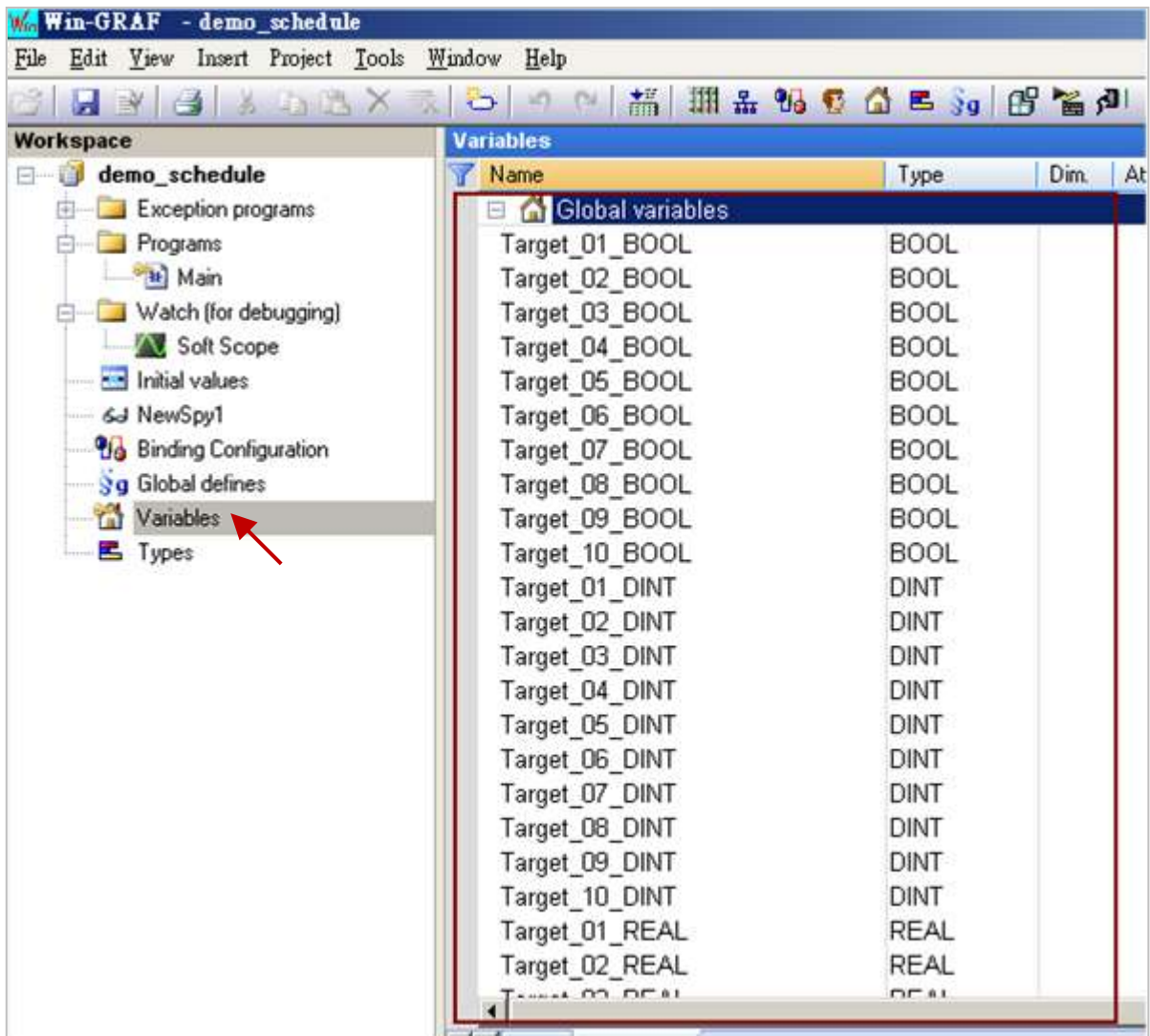
The "Variables" window shows the following table:

Name	Type	Dim	Attrib
Hour1			
Minute1			
Second1			
To_Page			
Current_Page			
RETAIN variables			
%IX10 - Schedule			
%IX10.0=Target01	BOOL		Input
%IX10.1=Target02	BOOL		Input
%IX10.2=Target03	BOOL		Input
%IX10.3=Target04	BOOL		Input
%IX10.4=Target05	BOOL		Input
%IX10.5=Target06	BOOL		Input
%IX10.6=Target07	BOOL		Input
%IX10.7=Target08	BOOL		Input
%IX10.8=Target09	BOOL		Input
%IX10.9=Target10	BOOL		Input

I/O Board – Schedule, 有 10 個 BOOL 輸入通道，用來表示 Target 1 ~ 10 是否已經啟用排程控制。

定義變數：

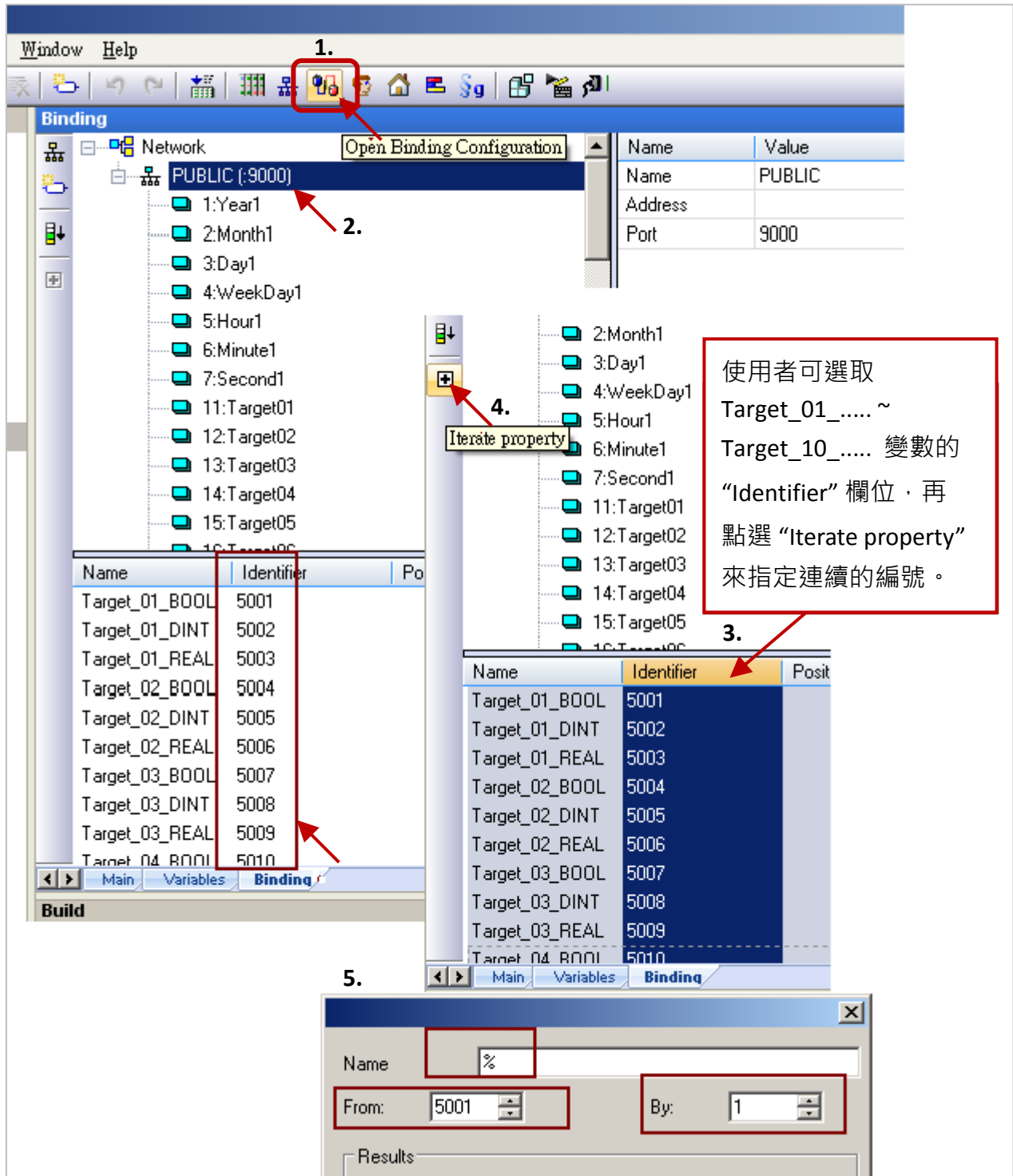
點選 “Variables” 可查看此範例中已宣告的 Win-GRAF 變數。



這些變數 – “Target_01_BOOL ~ Target_10_BOOL” · “Target_01_DINT ~ Target_10_DINT” 與 “Target_01_REAL ~ Target_10_REAL” – 是用來控制 10 個 Target 的 BOOL · DINT 與 REAL 變數。

Data Binding

在 [第 7 章](#)，我們已詳細說明了 Data Binding 的使用方式。若您想開放這台 Win-GRAF PAC 的變數，供其它台 PAC 讀取資料，則需將上述的變數拖曳到 “Binding” 視窗的 “Public” 區域內，並指定好這些變數的識別碼 (Identifier)，為了正確的控制排程設定，ID 編號 **必須為 5001 ~ 5030**。設定完成後，就會依照 User 在 Schedule-Control Utility 內設定的 10 個 Target 來進行排程控制。

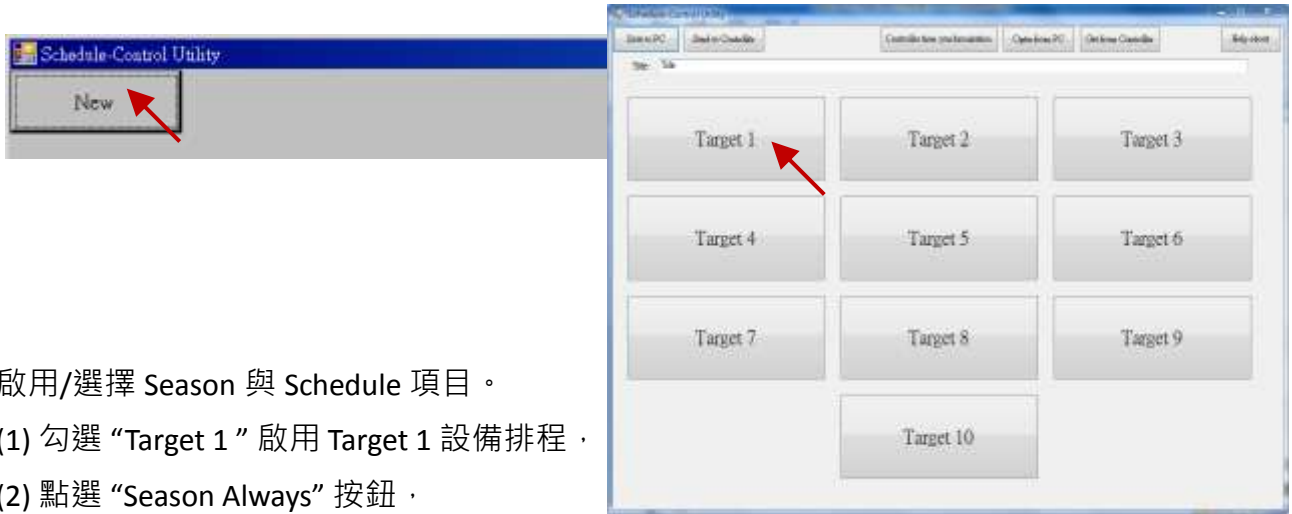


註: 若您想開放 Win-GRAF 變數，來讓 eLogger HMI 讀取資料，則需使用 [3.1 節 啟用 Win-GRAF PAC 為 Modbus TCP Slave](#) 的方式，來進行設定。

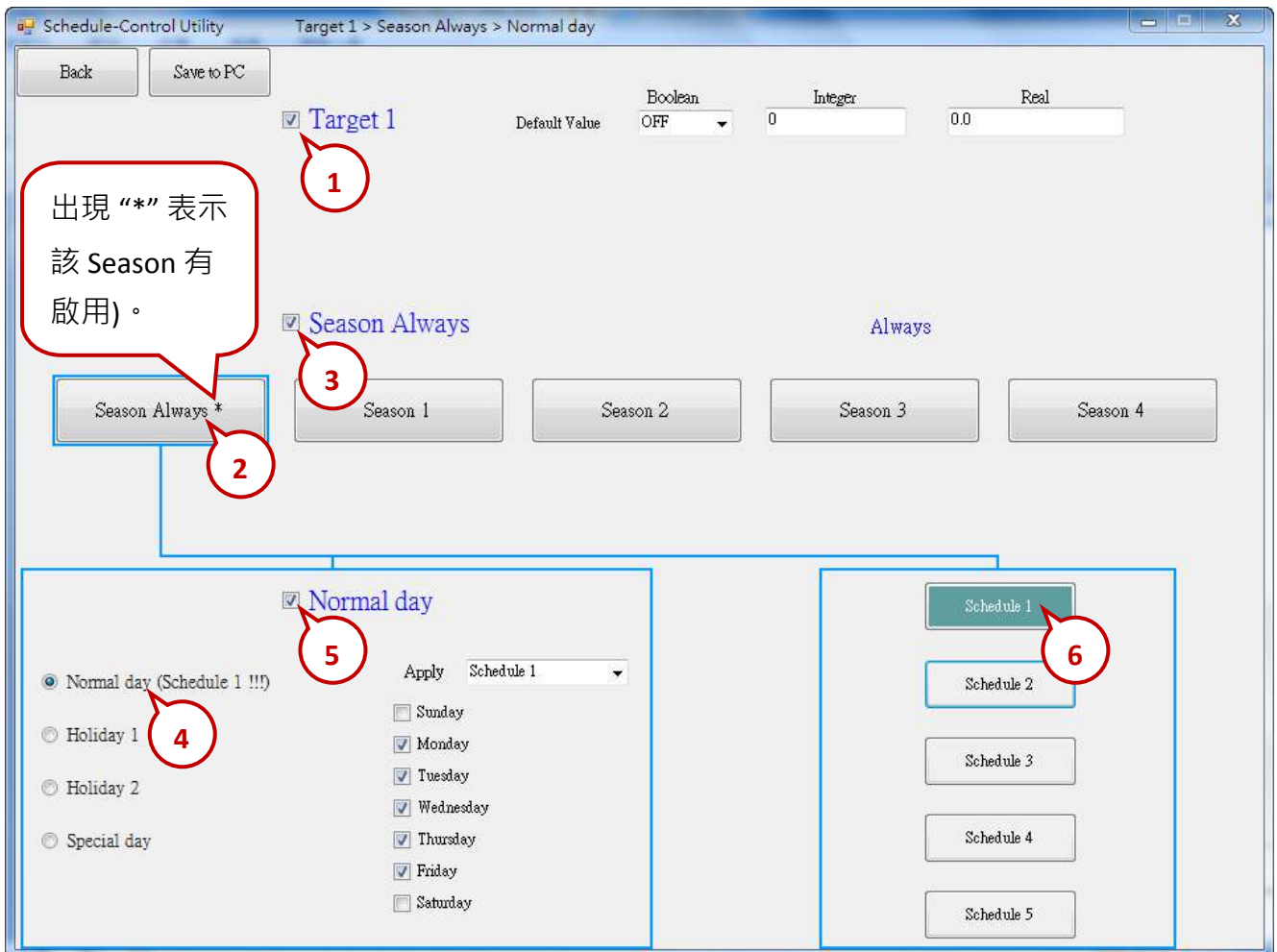
17.3 使用 Schedule-Control Utility 來設定 Target 1 ~ 10 的排程

以下將介紹使用 Schedule-Control Utility 的簡單範例，詳細的排程功能請見 [17.5 節](#)。

1. 在 PC 上開啟 Schedule-Control Utility (“Schedule_in_PC.exe”)。
(點選 New，再點選 “Target 1” 按鈕來設定 Target 1 設備的排程表)。



2. 啟用/選擇 Season 與 Schedule 項目。
 - (1) 勾選 “Target 1” 啟用 Target 1 設備排程，
 - (2) 點選 “Season Always” 按鈕，
 - (3) 勾選 “Season Always” 來設定全年不分季節的排程表，
 - (4) 點選 “Normal Day” 項目 (一般日，通常是週一~週五)，
 - (5) 勾選 “Normal Day” 來啟用與設定排程表，
 - (6) 點選 “Schedule 1” 來設定 排程表一。



3. 設定排程時段

於上述步驟 2 – (6) 選擇 “Schedule 1” 之後，繼續進行以下操作。

(1) 勾選 “01” 啟用該排程表內的第 1 個時段。

(A) 可設定為如下圖相同的時間，或自行設定方便測試的時間。

(B) 設定 3 個控制變數 (Boolean · Integer · Real) 所需的值 (如下圖)，或自行設定。

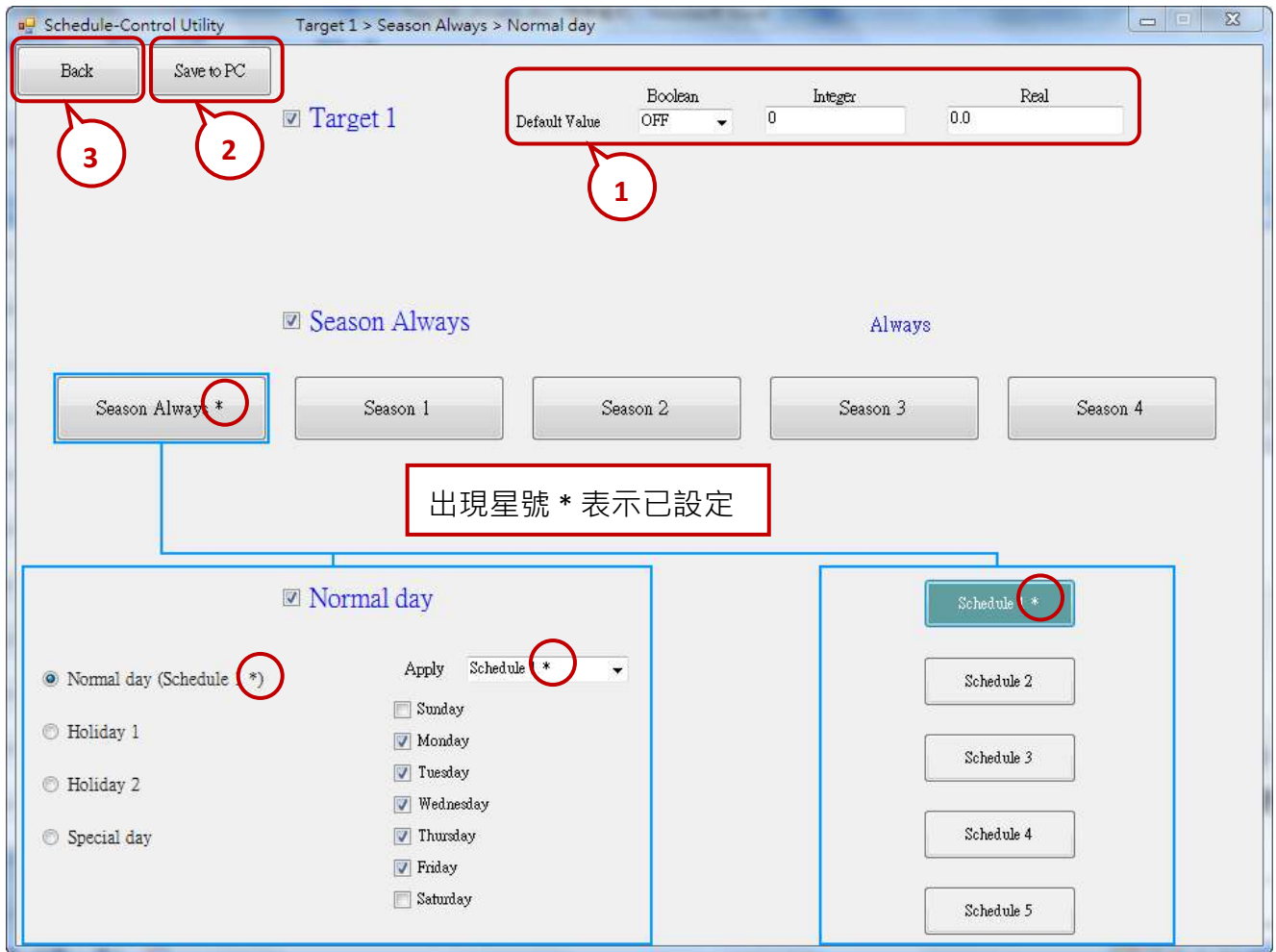
(2) 勾選 “02” 啟用第 2 個時段，設定方法同 (1)。

每個排程表最多可以設定 15 個時段，設定完成，請按 “Save and exit” 儲存並離開視窗。

The screenshot shows the "Schedule 1" dialog box. It contains a table for scheduling 15 time slots. The first two slots are selected. The first slot is from 8:30 to 12:00, and the second is from 13:00 to 17:30. To the right, there are three columns for control variables: Boolean (ON/OFF), Integer (10, 20), and Real (12.34, 25.67). A "Save and exit" button is highlighted with a red arrow.

	Hour	Minute	To	Hour	Minute	Boolean	Integer	Real
<input checked="" type="checkbox"/> 01:	8	30		12	0	ON	10	12.34
<input checked="" type="checkbox"/> 02:	13	0		17	30	ON	20	25.67
<input type="checkbox"/> 03:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 04:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 05:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 06:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 07:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 08:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 09:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 10:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 11:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 12:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 13:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 14:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 15:	0	0		0	0	OFF	0	0

4. 接著，會回到上一層設定視窗 (如下圖)，已設定過排程的項目，右方會標示星號 “*”。



5. 右上方的“Default Value”項目，可設定預設值。如未設定任何排程，或找不到排程設定時，則 Target 設備會依預設值動作，此範例預設值為“Boolean: OFF · Integer: 0 · Real : 0.0”。

“Default Value” 的優點: 活用 "Default Value" 可以減化 Schedule 1 ~ 5 內的時段設定，

比如: 若想 設定 Schedule 1 為以下，必需設定好 5 個時段

1. 00:00 ~ 08:00 OFF 0 0.0
2. 08:00 ~ 09:50 ON 0 0.0
3. 09:50 ~ 10:00 OFF 0 0.0
4. 10:00 ~ 11:50 ON 0 0.0
5. 11:50 ~ 24:00 OFF 0 0.0

但若充分利用 "Default Value" 值，將它設為 "OFF · 0 · 0.0"，則只需設定以下 2 個時段即可

1. 08:00 ~ 09:50 ON 0 0.0
2. 10:00 ~ 11:50 ON 0 0.0

6. 設定完成後，請按“Save to PC”於 PC 中儲存設定檔。(可自行設定檔名，此範例為 test1.txt)
7. 點選“Back”回上一層視窗，可繼續設定其他 Target 設備。(本範例僅設定 Target 1)

17.4 測試範例程式

本章節將示範如何下載範例程式與排程設定到連接的 Win-GRAF PAC，並測試排程控制功能。

1. 使用 Win-GRAF Workbench 下載 “demo_schedule” 專案至 PAC。(詳細步驟請參見 [2.3.5 節](#))

2. 使用 Schedule-Control Utility 下載 排程設定 到 PAC。

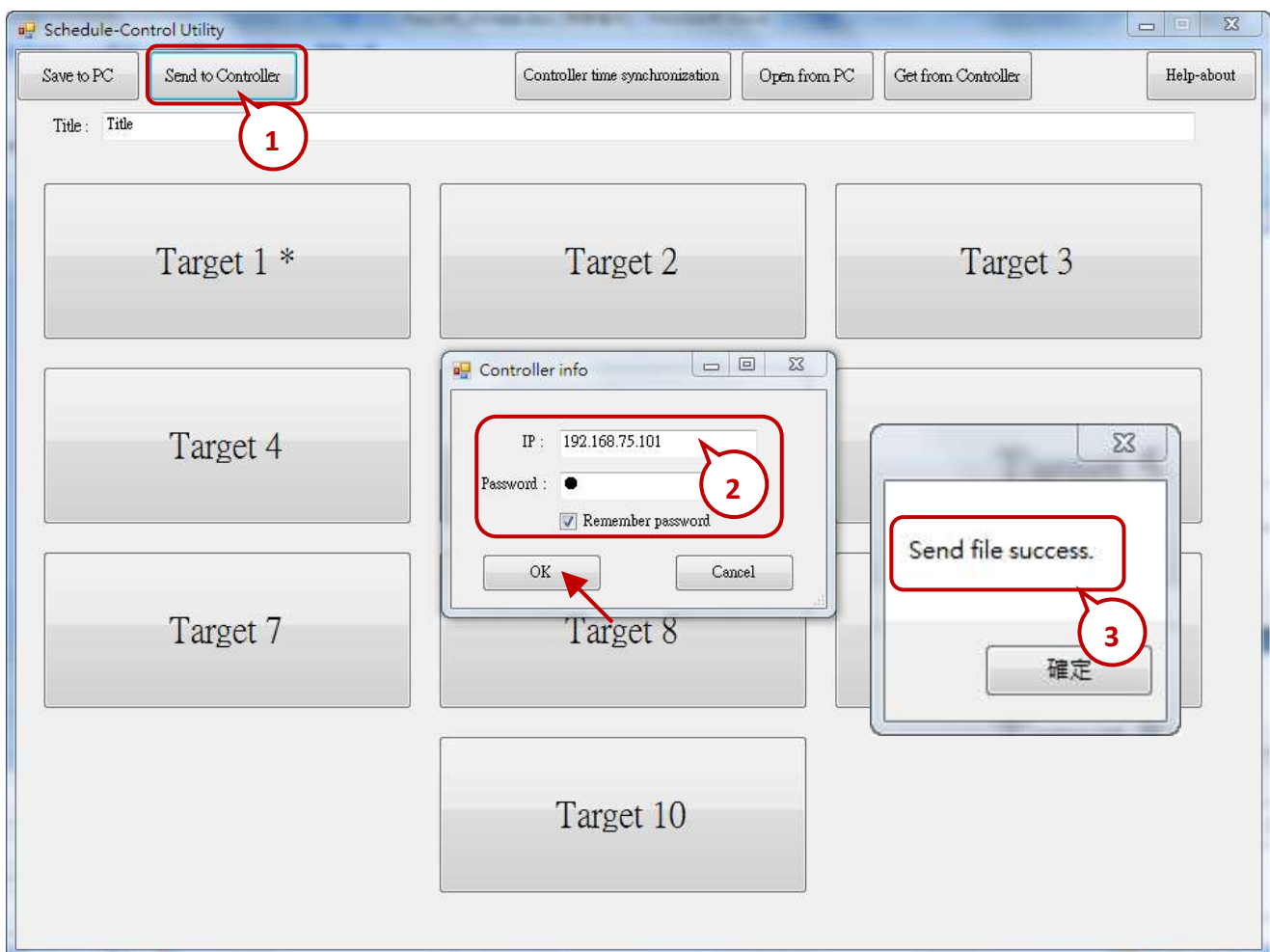
(1) 點選 “Send to Controller”。

(2) 輸入 控制器的 IP 位址 (請輸入您的 PAC 的 IP 位址編號)。

輸入 密碼 (本範例預設 0)，可勾選 “Remember Password” 來記憶密碼，點選 “OK” 按鈕，開始傳送到控制器。

(3) 傳送成功會跳出 “Send file success” 視窗。

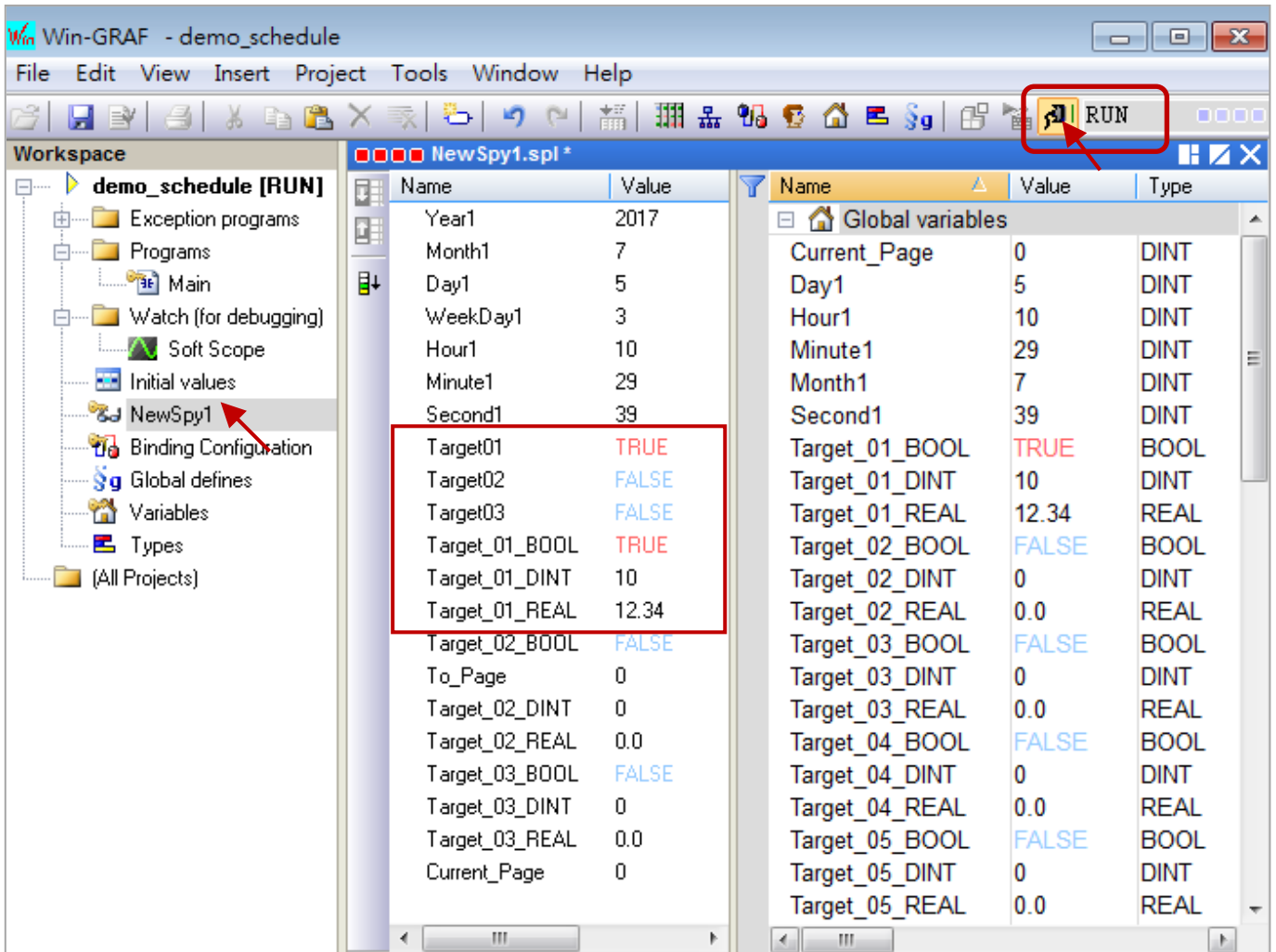
注意： 下載專案前，請確認已開啟 PAC 內的 Schedule-Control Utility。此外，請在 PAC Utility 的 Auto Execution 新增其路徑 \System_Disk\Win-GRAF\Schedule_in_PAC.exe。



3. 測試 Win-GRAF 專案:

點選 Win-GRAF Workbench 內的 "On Line" 與 PAC 連線 (如下圖) , 並開啟 "NewSpy1" 視窗畫面。
若連線成功 , 可見到 "Target_01_xxx ~ Target_10_xxx" 變數 , 皆依照在 Schedule-Control Utility 中設定的排程來顯示狀態或數值。

User 可再使用 Schedule-Control Utility 來修改排程設定 , 並下載至 Win-GRAF PAC。然後 , 再次查看這些變數是否依然控制得當。



17.5 Schedule-Control Utility 設定說明

17.5.1 每個 Target 控制變數的 Address

Schedule-Control Utility 最多可配置 10 個“控制標的”(Target 1 ~ Target 10) 的排程表 (Schedule) 來控制最多 10 個設備，而每個 Target 可控制 1 個 BOOL 變數、1 個 DINT 變數與 1 個 REAL 變數。

要在 Win-GRAF PAC 內啟用排程控制，必需先在 Win-GRAF Workbench 的 "I/O Boards" 視窗內連上一個“Schedule”(參考 [17.2 節](#))，請在 "Variables" 視窗定義所需的變數(參考 [17.2 節](#) - 定義變數)，並在 Binding 視窗內加入這些變數與設定其識別碼 (Identifier)，編號必需是 5001 ~ 5030。將專案下載至 PAC 後，PAC 端的排程機制就會自動去控制這些變數的值。

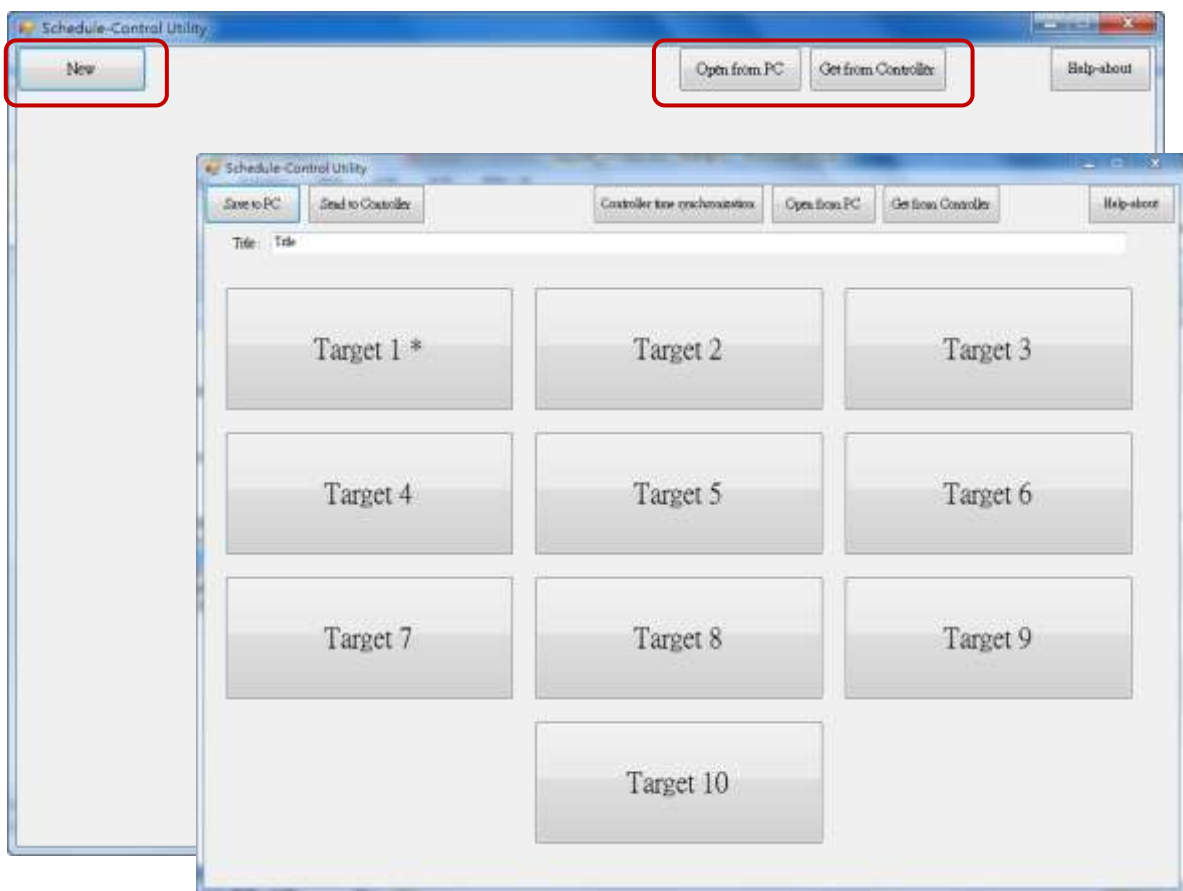
Address	型態	說明	Address	型態	說明
5001	BOOL	Target 1 控制的 BOOL, DINT 與 REAL 變數	5016	BOOL	Target 6 控制的 BOOL, DINT 與 REAL 變數
5002	DINT		5017	DINT	
5003	REAL		5018	REAL	
5004	BOOL	Target 2 控制的 BOOL, DINT 與 REAL 變數	5019	BOOL	Target 7 控制的 BOOL, DINT 與 REAL 變數
5005	DINT		5020	DINT	
5006	REAL		5021	REAL	
5007	BOOL	Target 3 控制的 BOOL, DINT 與 REAL 變數	5022	BOOL	Target 8 控制的 BOOL, DINT 與 REAL 變數
5008	DINT		5023	DINT	
5009	REAL		5024	REAL	
5010	BOOL	Target 4 控制的 BOOL, DINT 與 REAL 變數	5025	BOOL	Target 9 控制的 BOOL, DINT 與 REAL 變數
5011	DINT		5026	DINT	
5012	REAL		5027	REAL	
5013	BOOL	Target 5 控制的 BOOL, DINT 與 REAL 變數	5028	BOOL	Target 10 控制的 BOOL, DINT 與 REAL 變數
5014	DINT		5029	DINT	
5015	REAL		5030	REAL	

17.5.2 控制標的 (Target) 設定說明

每台 Win-GRAF PAC 最多可設定 10 個“控制標的”(Target 1 ~ 10)。請先開啟 Schedule-Control Utility，並點選 "New" 按鈕來建立設定檔，您可見到如下圖 10 個 Target 按鈕，其預設名稱為“Target 1” ~ “Target 10”。一個 Target 可設定用於不同季節的排程，已設定過排程的 Target 則會顯示星號“*”以方便辨識。

除了建立 (New) 新的排程設定，User 也可開啟 PC 現存的 (或從 PAC 載入) 設定檔來修改：

1. New: 建立新檔。
2. Open from PC: 開啟 PC 裡現有的設定檔。
3. Get from Controller: 從 PAC 載入現有的設定檔，需輸入 PAC IP 與密碼。

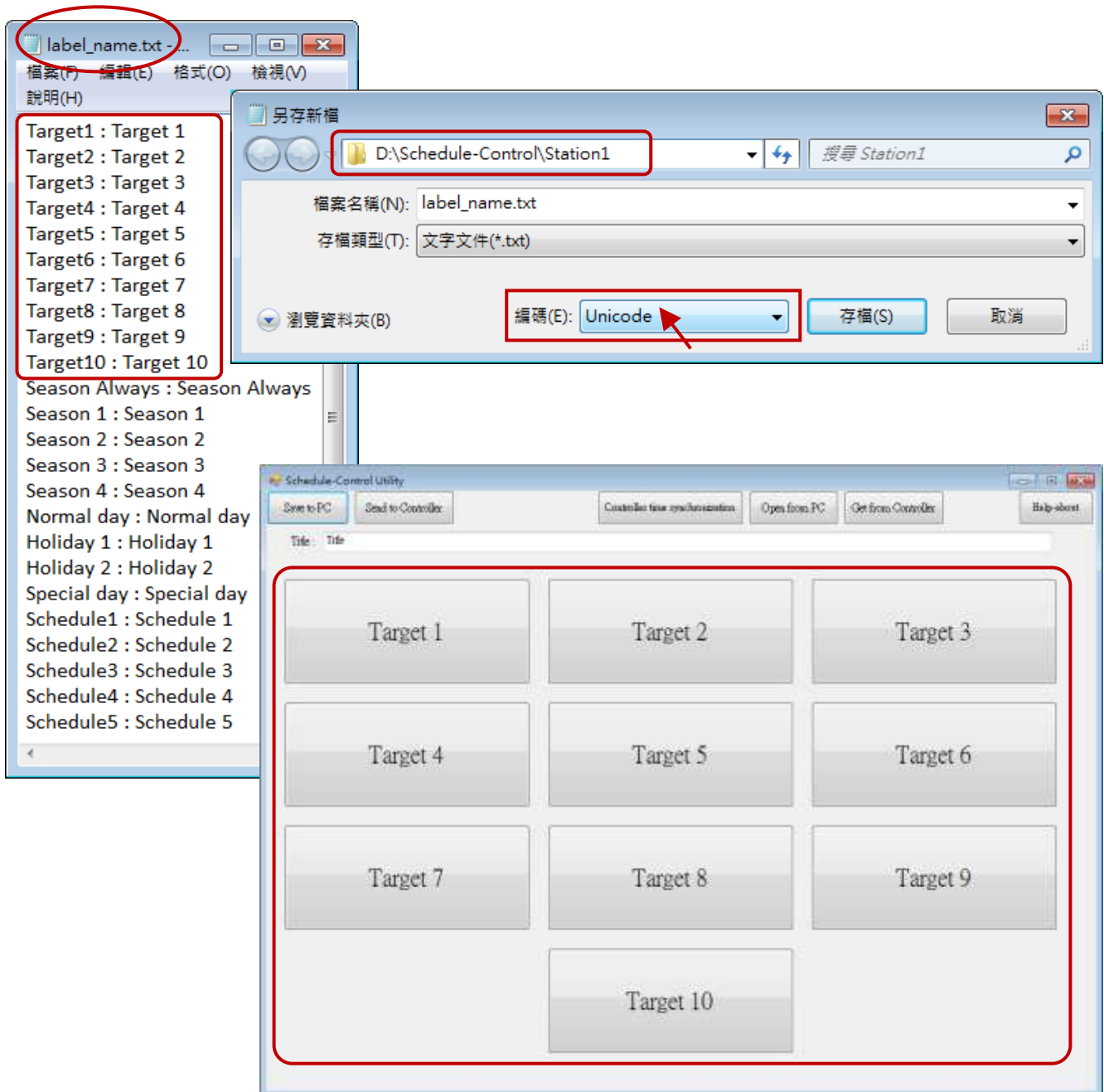


配合案場變更項目名稱：

為配合案場設備，User 可變更 Target，Season，...等名稱。請建立一個檔名為“Label_Name.txt”的文字檔 (如下圖)，並與 Schedule-Control Utility (即，“Schedule_in_PC.exe”) 存放在同一資料夾內 (比如，存放到 D:\Schedule-Control\Station1\Label_Name.txt)。

“Label_Name.txt” 檔案的建立原則:

1. 若此檔案不存在，畫面上會顯示預設的英文項目名稱 (例如: Target 1，Target 2)。
2. 檔案內冒號 (":") 後，可輸入想變更的 Target 名稱 (例如: “ 工廠 ”，頭尾空白會自動刪除)，或其他項目名稱 (例如: Season，Normal day，Holiday，Schedule ... 等)。
3. 可使用記事本 建立/編輯 檔案，但檔案需儲存為 “Unicode” 編碼格式。
4. 需將檔案複製到 PAC 內，與 “Schedule_in_PAC.exe” 相同的目錄中 \System_Disk\Win-GRAF\。

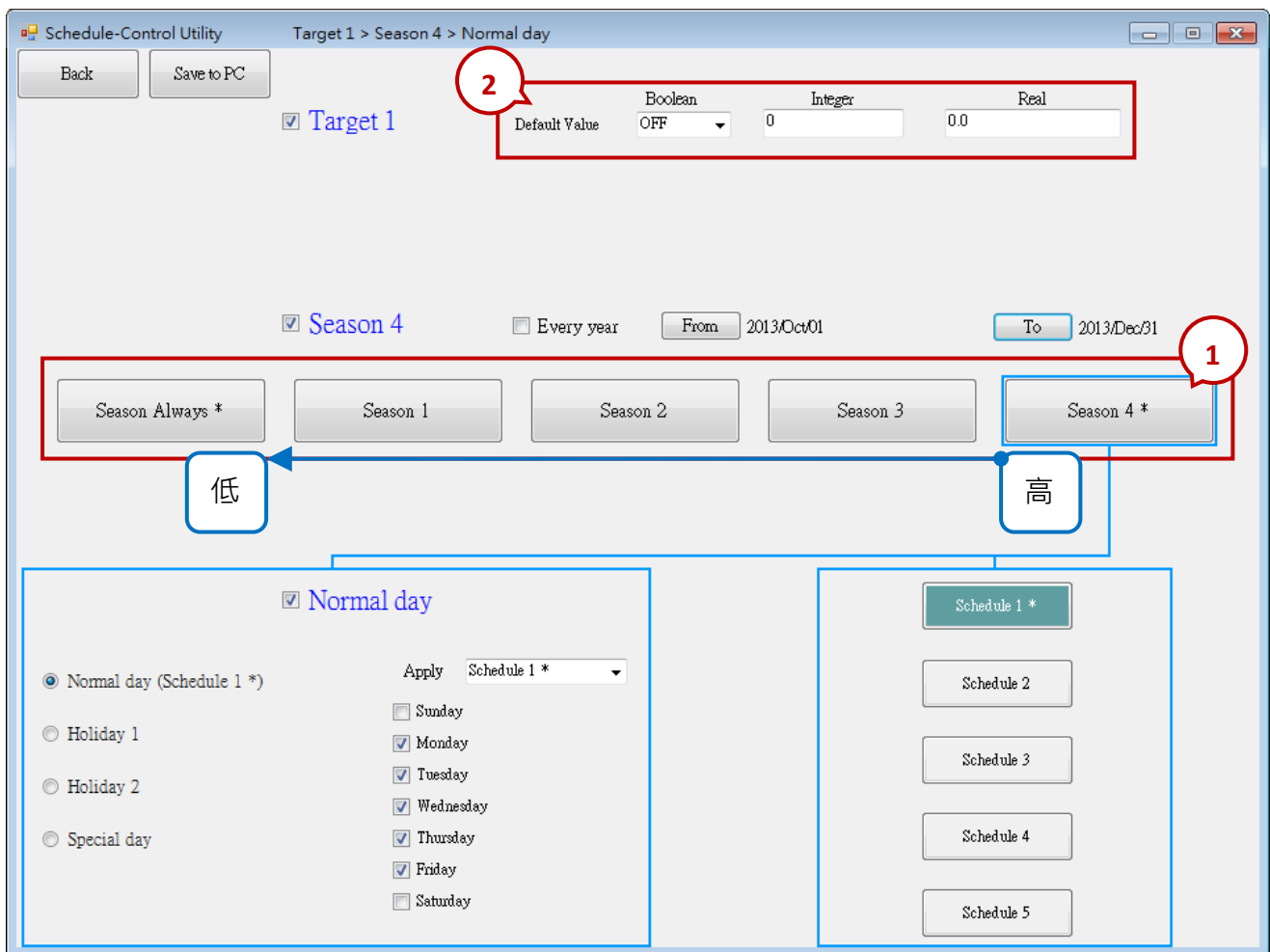


17.5.3 季節 (Season) 設定說明

每個Target (1 ~ 10) 內都有 "Season Always", "Season 1", "Season 2", "Season 3" 與 "Season 4" 設定項目，建議 User 可直接勾選 "Season Always"，表示啟用全年不分季節的排程表。

PAC 搜尋 Season 設定的優先順序:

1. 先搜尋 **Season 4** (若它有啟用才搜尋)
若找到該日的排程就進行 BOOL / DINT / REAL 的控制。
2. 若找不到就改為搜尋 **Season 3**，**Season 2...**，如此一直搜尋到 **Season 1**。
3. 若找不到就進行 **Season Always** 設定的控制。
4. 若都找不到就會採用該 Target 所設定的 "Default Value"。



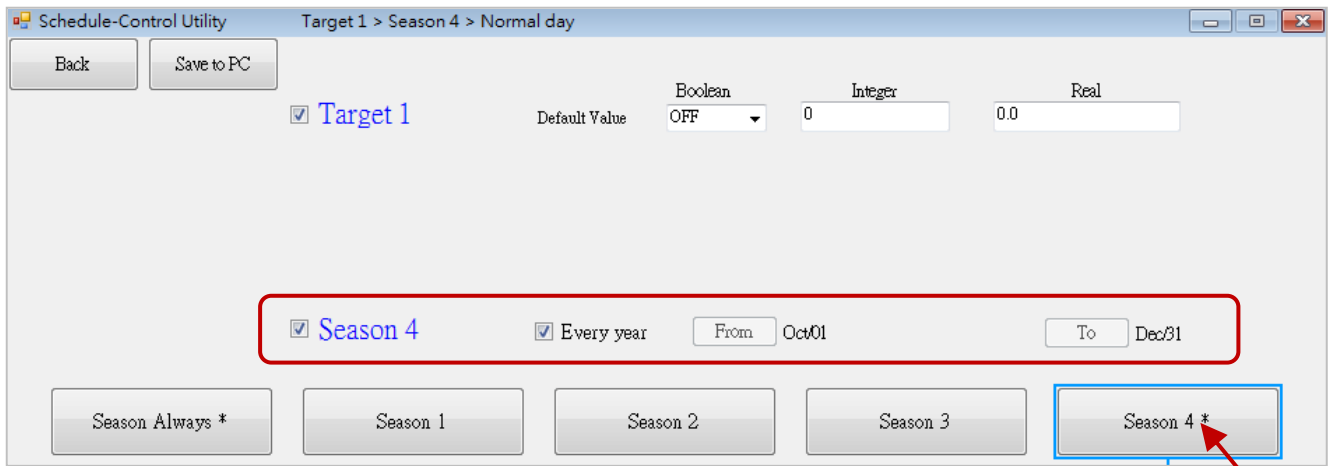
Season 的日期區間:

Season 1 ~ 4 需要設定它的 "日期區間"。設定後，建議勾選 "Every Year"，這樣每一年的該月/日區間內皆適用。

原則 1: 這 4 個 Season 設定的日期區間不可重疊。

原則 2: 若勾選 "Every Year"，會只判斷月/日 不重疊即可，不會去判斷年份。

但若沒勾選 "Every Year"，則會判斷 起始年/月/日 需早於 結束年/月/日。



註: 您必須取消勾選 “Every Year”，才可設定日期區間，並請注意日期順序。

例如:

1. 正確的設定:

可勾選 “Every Year”，這樣會適用在每一年的該月/日區間內。

Season 1	01/01 ~ 03/31
Season 2	04/01 ~ 07/15
Season 3	07/16 ~ 09/30
Season 4	10/01 ~ 12/31

2. 錯誤的設定:

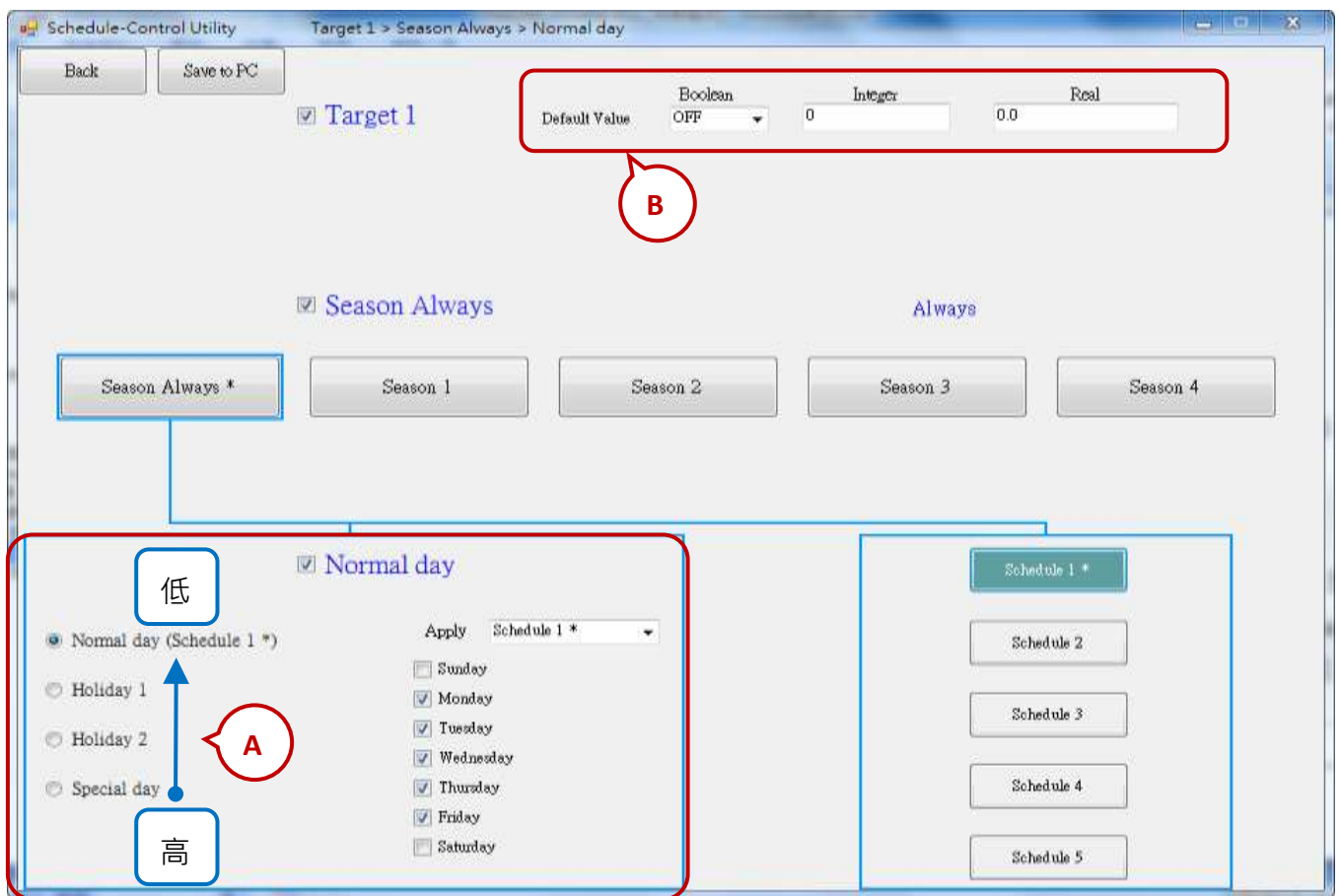
因為 2013/03/16 ~ 2013/03/31，重覆出現在 Season 1 與 Season 2 的區間內。

Season 1	01/01 ~ 03/31
Season 2	03/16 ~ 07/15
Season 3	07/16 ~ 12/31
Season 4	Disabled

17.5.4 一般日，假日，特別日 設定說明

每個 Season 內會區分為 一般日 (Normal day)，假日 1、2 (Holiday 1、2) 與 特別日 (Special day)。啟用此設定時，必需選擇一個 Schedule 編號 (1 ~ 5) 來套用時段。

Normal day	一般日，一般的習慣是星期一 ~ 星期五
Holiday 1	假日 1，通常設為星期六 與 星期日。
Holiday 2	假日 2，某些場所有第 2 種排程不一樣的假日 (比如，設為 星期三...)
Special Day	特別日，用來指定一些特別假日 或 補上班日的排程。 (比如，10 月 10 日、7 月 4 日、10 月 1 日、12 月 25 日...，一個 Season 內，最多可指定 50 個特別日。)



A. 一般日，假日，特別日的優先順序:

先搜尋 特別日，若找不到該日期的排程，再搜尋 假日 2，然後若還是找不到再搜尋 假日 1，再找不到換搜尋 一般日。

B. Boolean / Integer / Real 變數的預設值 (Default Value) :

每個 Target 必需設定 Boolean、Integer 與 Real 控制變數的預設值，主要用來處理當所有 啟用的 Season 設定中都找不到符合的排程日 或 時段時，就會以這個 "Default value" (一般為 Boolean: OFF、Integer: 0 與 Float value : 0) 來進行控制，User 也可自行設定所需的值。

C. Normal day · Holiday 1 與 Holiday 2 的日期設定:

請注意日期不可重疊。例如，

正確設定:

Normal day	Monday · Tuesday · Wednesday · Thursday · Friday
Holiday 1	Sunday · Saturday
Holiday 2	Disabled

錯誤設定: (因為 “Normal day” 與 “Holiday 2” 內的 “Friday” 重疊了)

Normal day	Monday · Tuesday · Wednesday · Thursday · Friday
Holiday 1	Saturday
Holiday 2	Sunday · Friday

D. Special Day 的日期設定:

“Special Day” 是用來設定一些特別日的排程，比如 特別假日 或 補上班日，每個 Season 最多可以設定 50 個日期。“Special Day” 的優先順序高於 “Holiday 2” 與 “Holiday 1” 與 “Normal day”。每個啟用的 “Special Day” 日期，必需選擇一個要套用的 Schedule 編號 (1 ~ 5)。

The screenshot shows the 'Special day' configuration window. On the left, a sidebar lists 'Normal day (Schedule 1 *)', 'Holiday 1', 'Holiday 2', and 'Special day *' (selected). A red circle '1' is around 'Special day *'. The main window has a 'Date Setting' button (red circle '3') and a 'Special day' checkbox (checked, red circle '2'). Below, a 'Calendar' window is open for July 2017, with the date '7' selected (red circle '4'). The 'Calendar' window has 'Confirm' and 'Cancel' buttons. Below the calendar, there is an 'Add a new date' button (red circle '4'), a 'Delete' button, and a 'Schedule 1' dropdown menu (red circle '5'). At the bottom, there is a 'Save Setting' button (red circle '6'), 'Clear all', and 'Exit' buttons.

17.5.5 排程 (Schedule) 設定說明

每個 "Season" 內最多可以設定 5 個排程表 (Schedule 1 ~ 5) · 每個排程 (Schedule) 內最多可以設定 15 個時段 · 時間最小單位為分鐘 · 必需在 00:00 ~ 24:00 的範圍內。



例如: 以下排程是正確的。

編號	時段	Boolean	Integer	Real
01	00:00 ~ 08:00	OFF	100	30
02	08:00 ~ 12:00	ON	150	25.5
03	12:00 ~ 13:00	OFF	120	27
04	13:00 ~ 17:00	ON	150	25.5
05	17:00 ~ 24:00	OFF	100	30

排程時段搜尋的優先順序:

排程時段搜尋的順序，會先搜尋較大編號的排程時段，若找不到再往編號小的搜尋。

例如: 下表顯示了 5 項排程時段設定。

編號	時段	Boolean	Integer	Real
01	00:00 ~ 08:00	OFF	100	30
02	08:00 ~ 12:00	ON	150	25.5
03	12:00 ~ 13:00	OFF	120	27
04	13:00 ~ 17:00	ON	150	25.5
05	17:00 ~ 24:00	OFF	100	30

1. 會依序由編號 05，04，03，02，01 來搜尋，若發生 排程時段設定重疊時，就會採用 編號較大的時段來進行控制。
2. 若在編號 "15" ~ "01" 的時段設定中皆找不到符合的時間點，就會套用 "Default Value" 的設定。

The screenshot shows a software window titled "Schedule 1" with a "Copy from" button at the top. Below it is a table with columns for "Hour", "Minute", "To", "Hour", "Minute", "Boolean", "Integer", and "Real". The rows represent time slots from 01 to 15. A red arrow on the left side points upwards from row 15 to row 01, indicating the search order. The data in the table matches the table provided in the text above.

	Hour	Minute	To	Hour	Minute	Boolean	Integer	Real
<input checked="" type="checkbox"/> 01:	0	0		8	0	OFF	100	30
<input checked="" type="checkbox"/> 02:	8	0		12	0	ON	150	25.5
<input checked="" type="checkbox"/> 03:	12	0		13	0	OFF	120	27
<input checked="" type="checkbox"/> 04:	13	0		17	0	ON	150	25.5
<input checked="" type="checkbox"/> 05:	17	0		24	0	OFF	100	30
<input type="checkbox"/> 06:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 07:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 08:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 09:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 10:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 11:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 12:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 13:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 14:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 15:	0	0		0	0	OFF	0	0

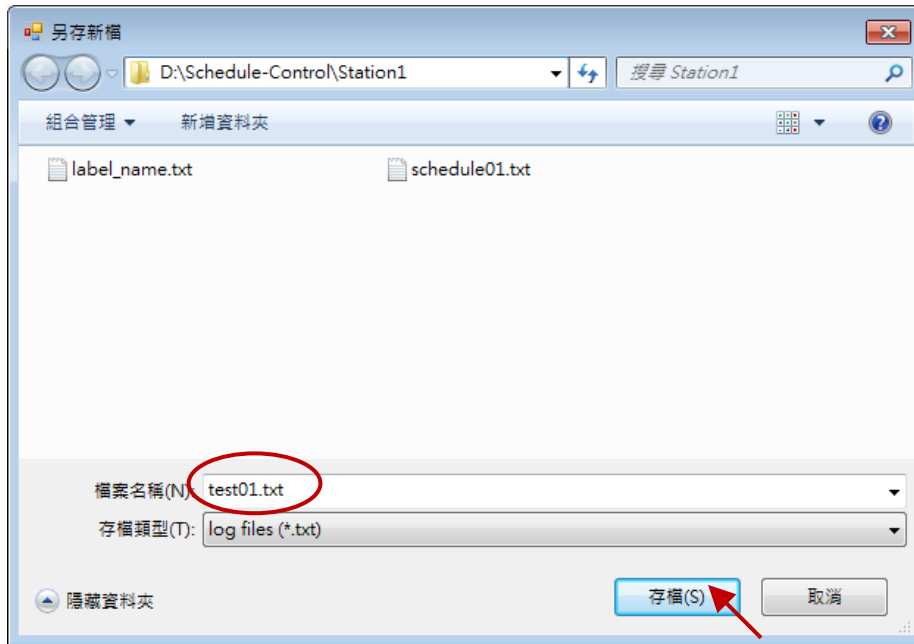
Buttons: "Save and exit" and "Cancel".

17.5.6 儲存與傳送 設定檔至 Win-GRAF PAC

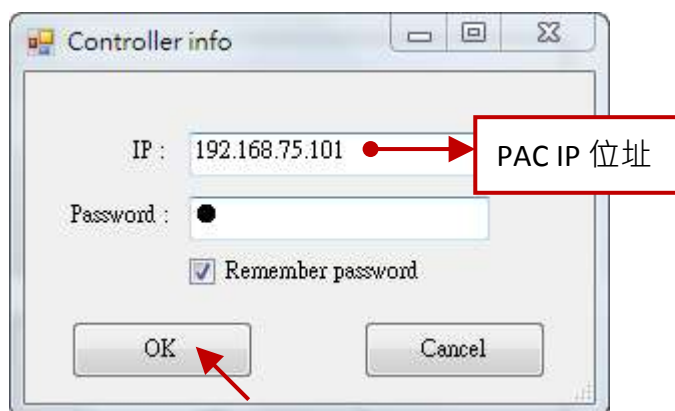
完成排程設定後，儲存並傳送 設定檔至 PAC 中：



1. 點選“Save to PC”，儲存設定檔案 (“*.txt”)。

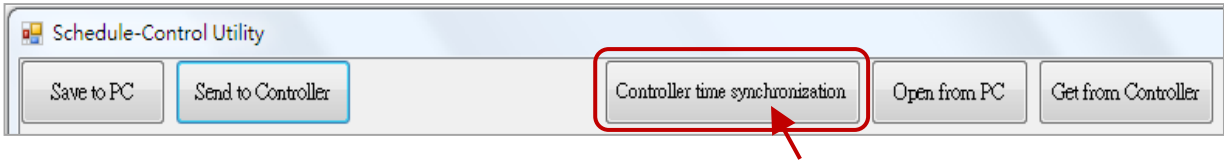


2. 點選“Send to Controller”，傳送設定檔到 PAC，請填入 PAC IP 與密碼 (預設: 0)。
(可勾選儲存密碼，加快下次的傳送程序。)



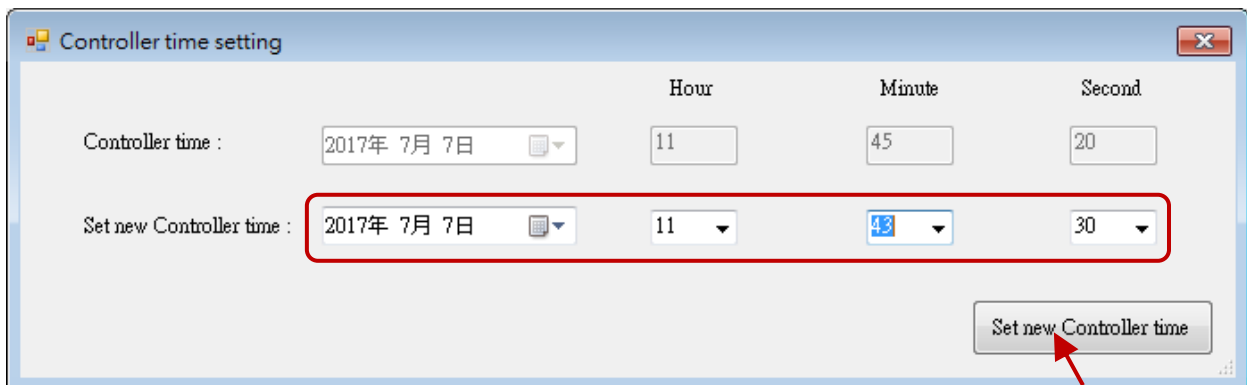
17.5.7 時間同步 (Time Synchronization)

若經過長時間 (比如 1 年) 都沒有校對 PAC 的時間，時間可能會誤差 10 多秒到數分鐘，為了控制排程時間的一致性，Schedule-Control Utility 提供校時功能來校正控制器的時間。



步驟:

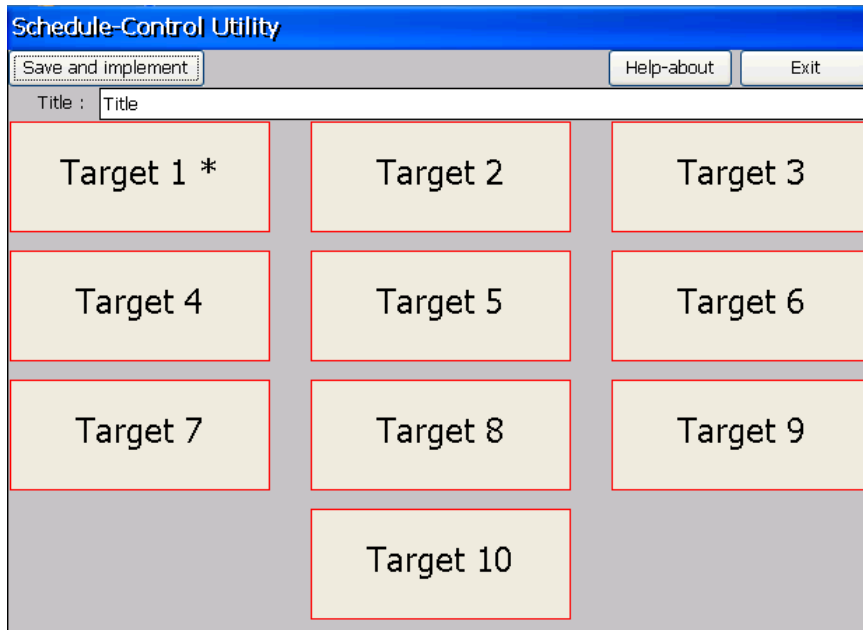
1. PC 需使用 Ethernet cable 連上 Ethernet Switch 再去連該 Win-GRAF PAC。
2. 點選 "Controller Time Synchronization" 按鈕，並輸入目前的 PAC IP 與密碼 (預設: 0)，再點選 OK。
3. 設定新的日期、時、分、秒。
4. 點選 "Set new Controller time 按鈕"，即可設定新的 PAC 時間。



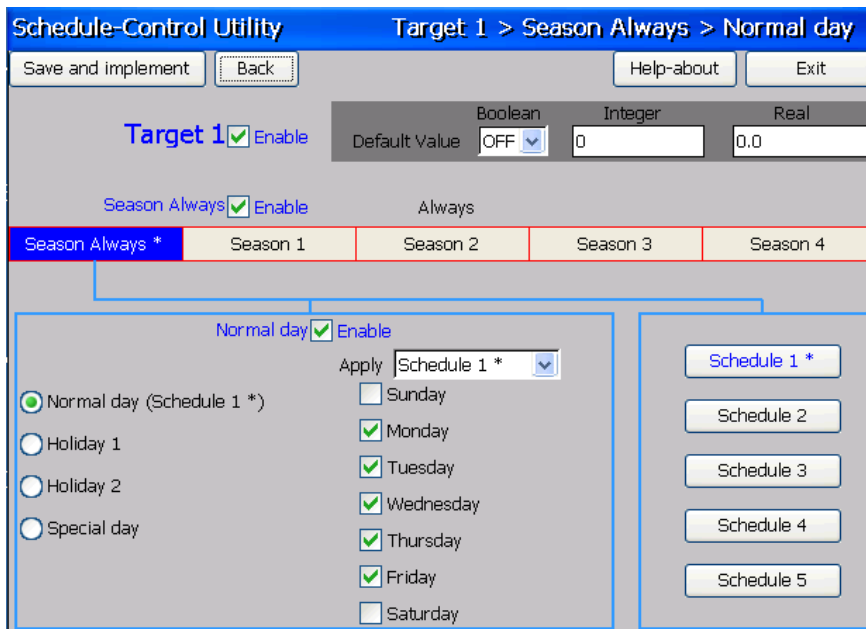
17.5.8 控制器端的 Schedule-Control 工具程式

PAC 控制器端的設定方式與 PC 端相同，只是畫面有點差異，設定方式請參考 [17.5.2](#) ~ [17.5.6](#) 節。

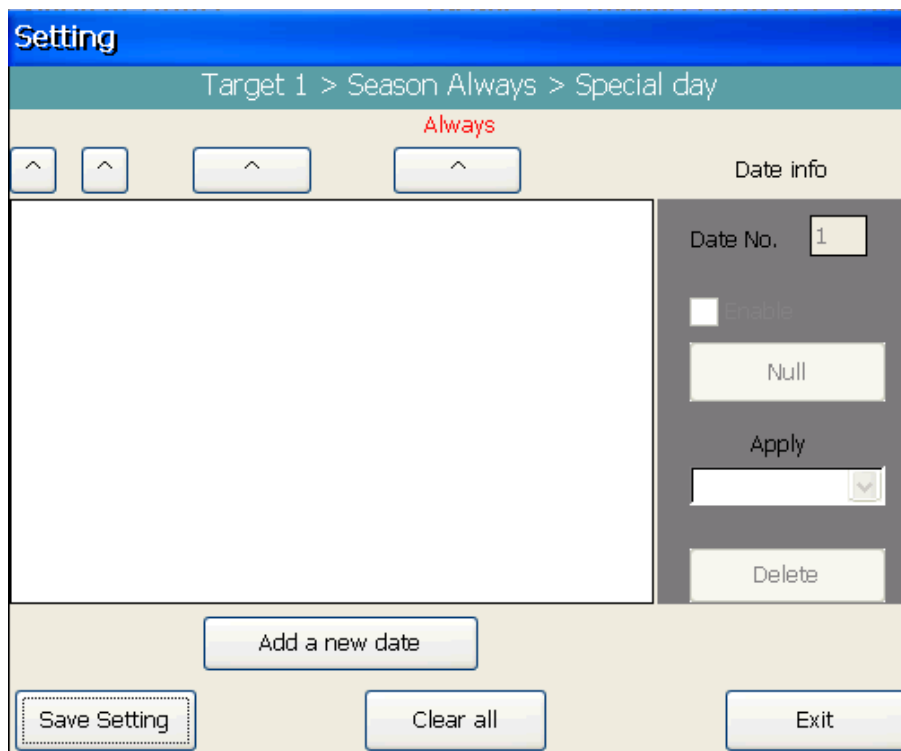
1. “Target” 設定:



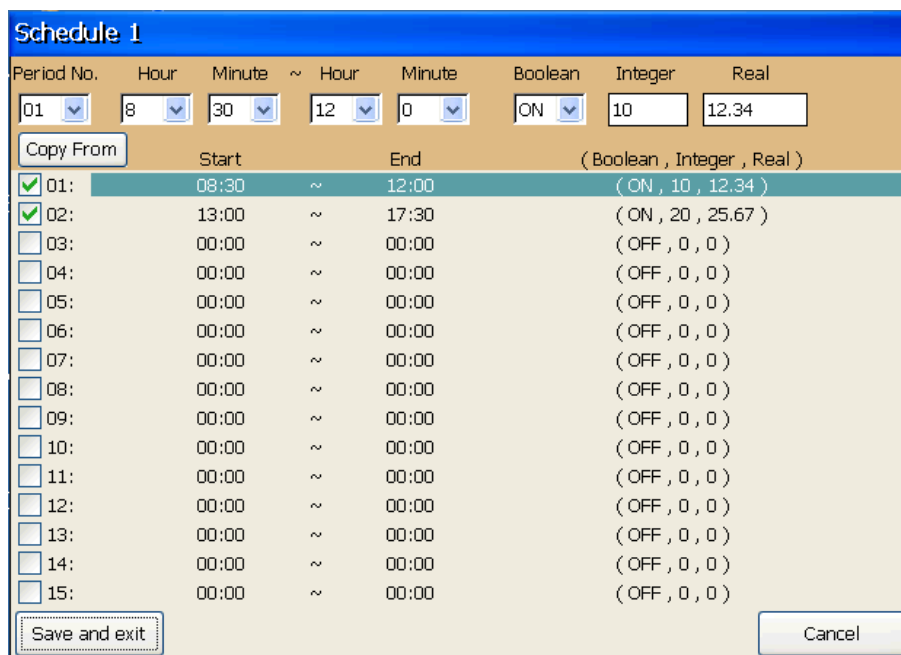
2. “Season” 設定:



3. 排程的 “Special day” 設定:



4. “Schedule” 設定:



17.5.9 在 eLogger HMI 內應用 Schedule-Control

eLogger 是泓格科技開發的一套免費且簡單易用的 HMI (Human Machine Interface) 軟體，可用來設計 Local HMI 與 Web Server HMI，並支援 PC、手機透過網頁瀏覽的方式對 PAC 進行遠端操作。所有 Win-GRAF PAC 都支援 eLogger HMI。

關於 eLogger HMI 的相關說明，請至以下網頁參見 Win-GRAF FAQ-018 與 FAQ-019:

[泓格科技首頁](#) > [支援服務](#) > [產品問答](#) > [Win-GRAF Soft-Logic PAC](#) > [FAQ-018](#)、[FAQ-019](#)。或
http://www.icpdas.com/root/support/faq/win-graf_tc.php



您可在 Win-GRAF FAQ 網頁，直接下載範例程式 (“demo_faq018_all.zip” 或 demo_faq019_all.zip)，或在 Win-GRAF-PAC-CD (\napdos\win-graf\demo_project\) 內取得，並依照文件的內容，來操作並測試專案。

第 18 章 開發屬於自己的 Function 與 Function Block

本章介紹以 Visual Studio 2008 開發工具來開發自定義的 Function/Function Block 的 DLL 檔。相關範例程式/專案 可以在 XP-8xx8-CE6, WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7, VP-x2x8-CE7 產品盒內附的 CD-ROM 內找到。

相關範例:

光碟 : \napdos\Win-GRAF\demo-project\user_c_lib\

\demo_user_c : VS 2008 的專案資料夾，可用來產生自定的 Function 或 Function Block 的 “user_c.dll” 檔案。(參考 [18.4 節](#))

..\user_c.dll : 預先編譯好的 DLL 檔，包含此範例中 Function (“bytes_to_long”) 和 Function Block (“long_to_bytes”) 的 “user_c.dll” 檔案。
適用於 WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7, WP-5xx8-CE7: \wp_vp\user_c.dll。
適用於 XP-8xx8-CE6: \xpac\user_c.dll。

\user : Win-GRAF Library 資料夾，包含此範例中 Function (“bytes_to_long”) 和 Function Block (“long_to_bytes”) 的 Library 相關檔案。(參考 [18.3 節](#))

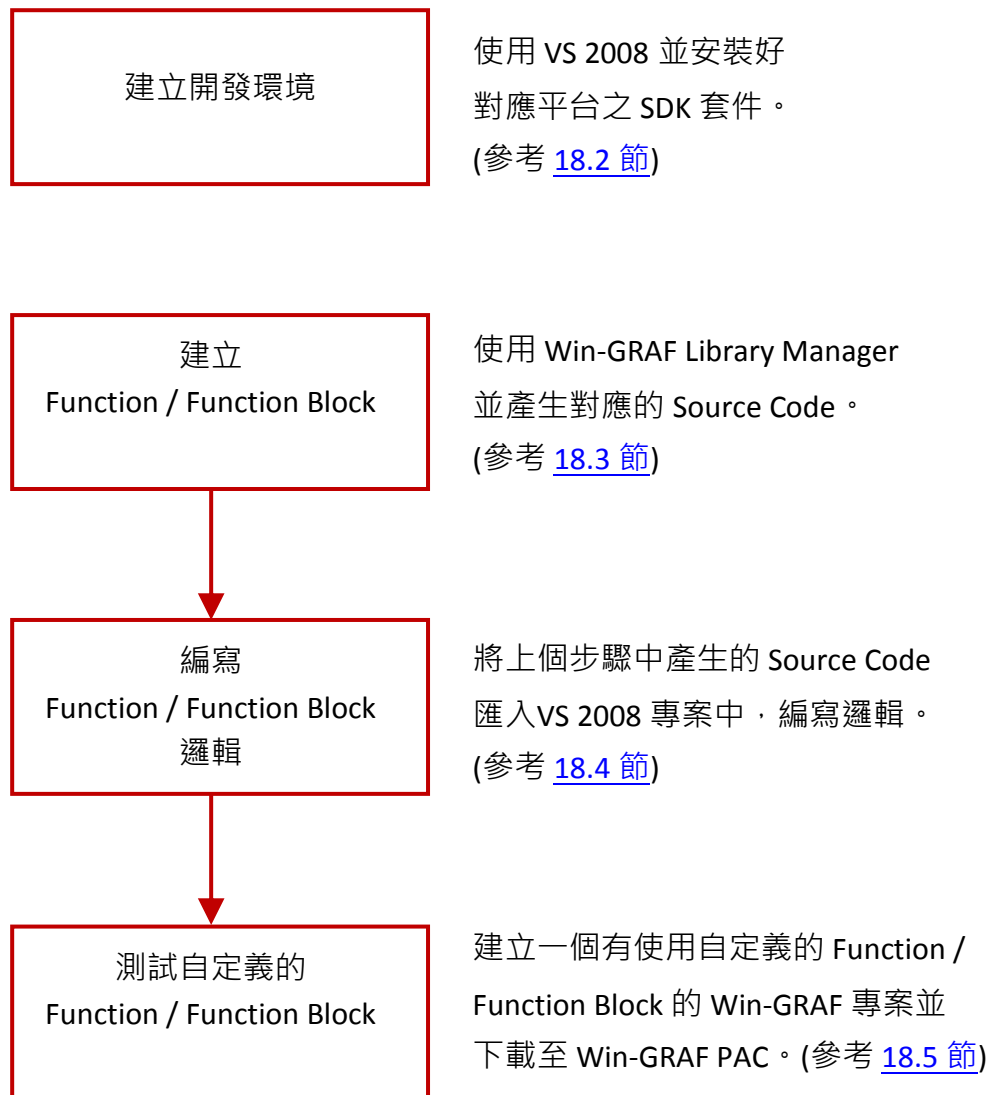
Win-GRAF 範例:

光碟 : \napdos\Win-GRAF\demo-project\demo_user_c.zip

ICP DAS 標準出貨的 Win-GRAF PAC 沒有包含 “user_c.dll” 檔案，若想在 PAC 內新增自定的 Function 與 Function Block，使用者就必需加入自己編譯產生的 DLL 檔。請將該檔案放在 PAC 內與 Win-GRAF Driver 相同的資料夾內 (即，\System_disk\Win-GRAF\)

然後，將該 PAC 重新開機一次，它就會支持使用者自定的 Function 與 Function Block。

18.1 自定義的 Function/Function Block 開發流程



18.2 建立開發編譯環境

下載相關 SDK 開發套件:

客戶可在下列網頁下載相關的開發套件 (SDK)

1. 適用 XPAC (XP-8xx8-CE6)

<ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/platformsdk/>

(pacsdk_ce_x.x.x_vs2008.msi)

2. 適用 ViewPAC (VP-x2x8-CE7) 、 WinPAC (WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7)

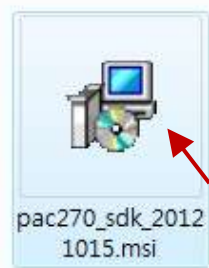
ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/sdk/platformsdk/

(pac270_sdk_yyyymmdd.msi)

18.2.1 安裝 ViewPAC 或 WinPAC 的開發套件 (SDK)

注意: 請確認您的 PC 內已經安裝了 Microsoft VS2008，才能進行以下步驟。

1. 滑鼠雙擊下載的 SDK 檔案 (例如: pac270_sdk_20121015.msi)，將其安裝至 VS2008 中。



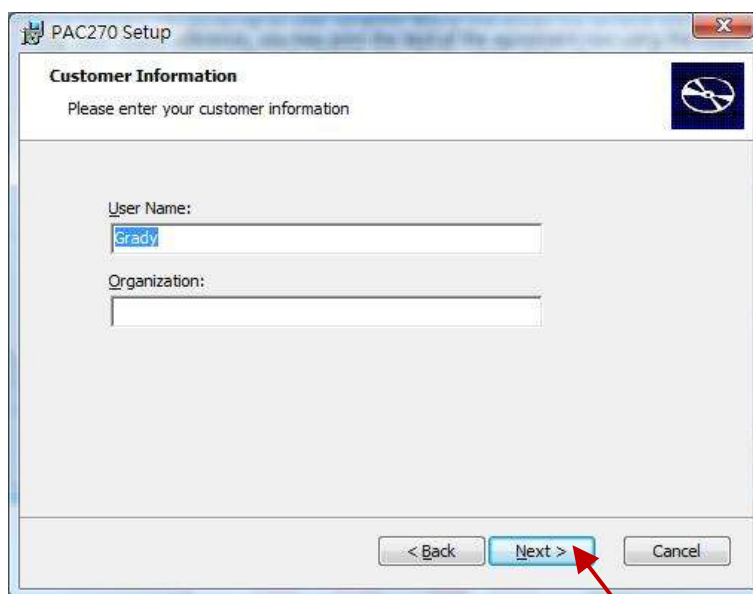
2. 點擊“Next”按鈕。



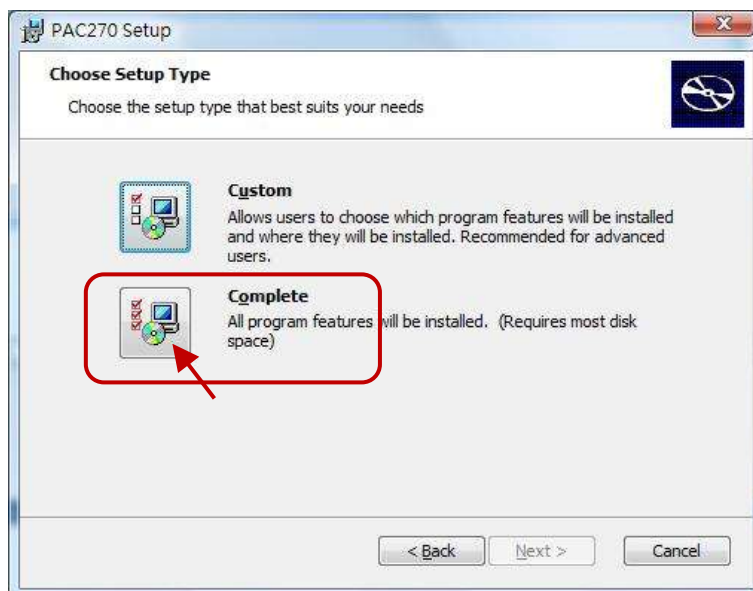
3. 點選 “Accept” ，再點擊 “Next” 按鈕。



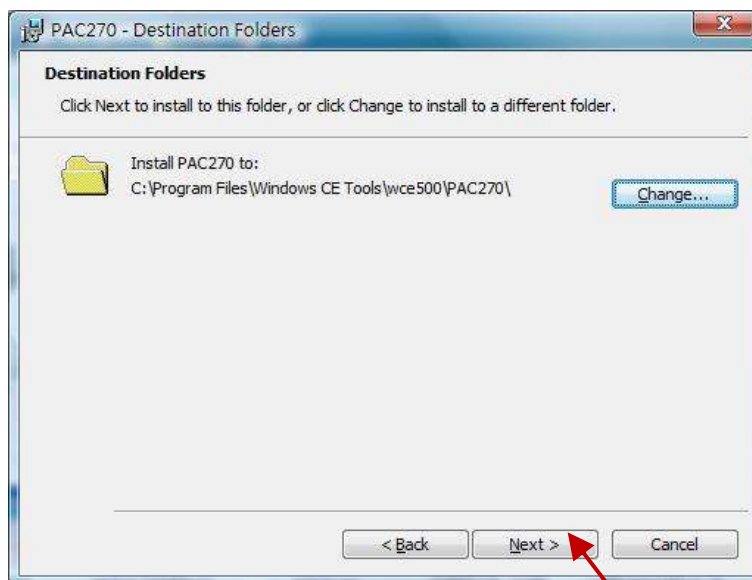
4. 點擊 “Next” 按鈕。



5. 點擊 “Complete” 按鈕。



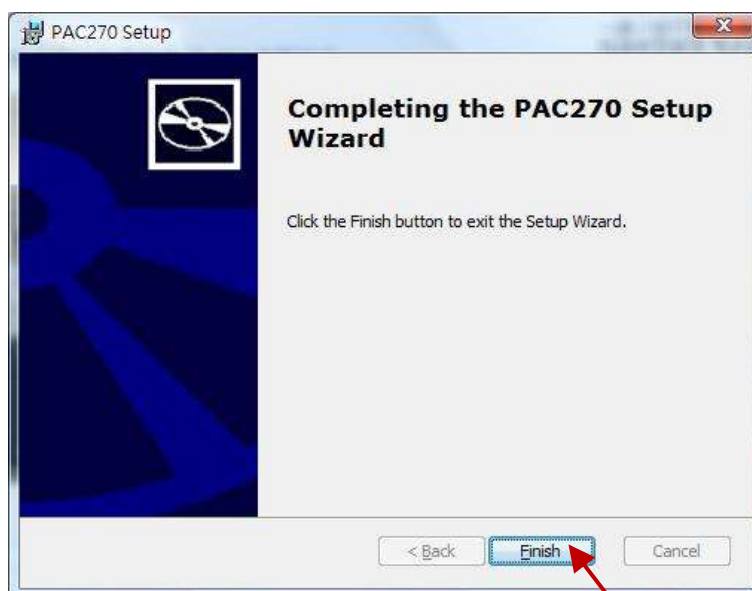
6. 點擊 “Next” 按鈕。



7. 點擊 “Install” 按鈕，開始安裝 SDK。



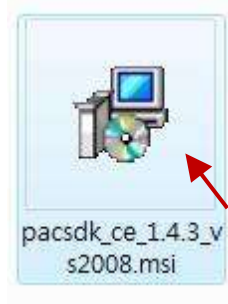
8. 待安裝結束後，點擊 “Finish” 按鈕離開安裝程序。



18.2.2 安裝 XPAC (XP-8xx8-CE6, XP-8xx8-Atom-CE6) 的開發套件 (SDK)

注意: 請確認您的 PC 內已經安裝了 Microsoft VS2008，才能進行以下步驟。

1. 滑鼠雙擊下載的 SDK 檔案 (例如: pacsdk_ce_1.4.3_vs2008.msi)，將其安裝至 VS2008 中。



2. 其他步驟，請參考 [18.2.1 節](#) 的步驟 (2) ~ (8)。

18.3 定義 Function 或 Function Block

18.3.1 定義 Function Lib

這個範例說明如何建立 Function 的 Library - "bytes_to_long"，該 Function 的功能是將 4 個 Byte (0 ~ 255) 轉換成一個長整數 (32-bit Signed Integer)。

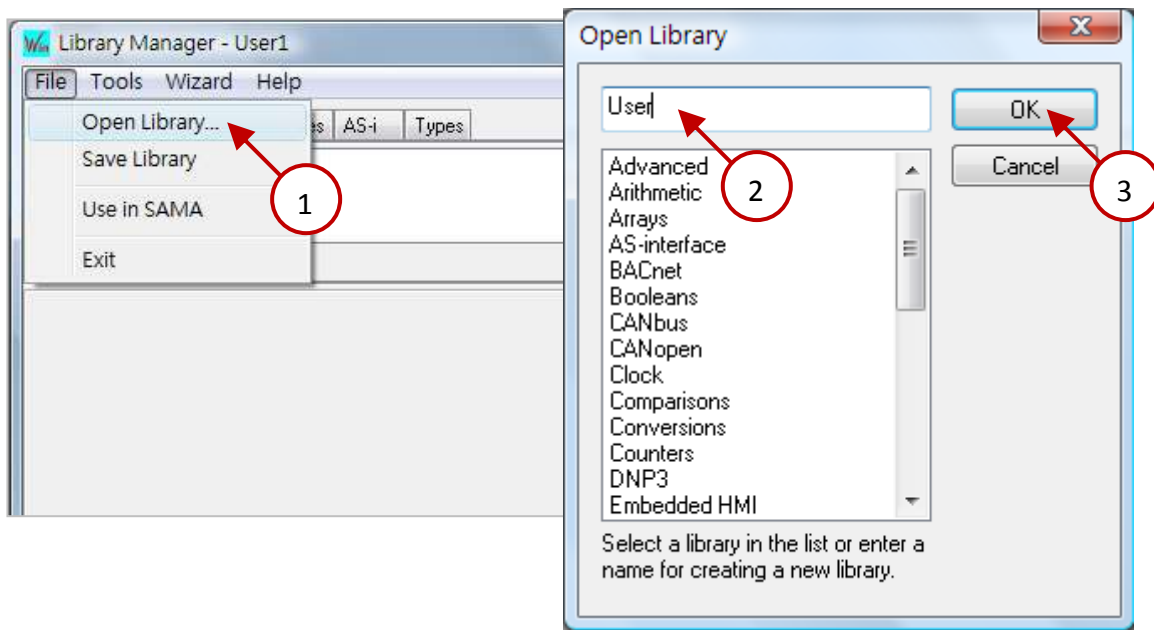
步驟如下：

1. 請開啟 程式集 → Win-GRAF → Libraries → OEM (如下圖，或參考 1.2.3 節)。

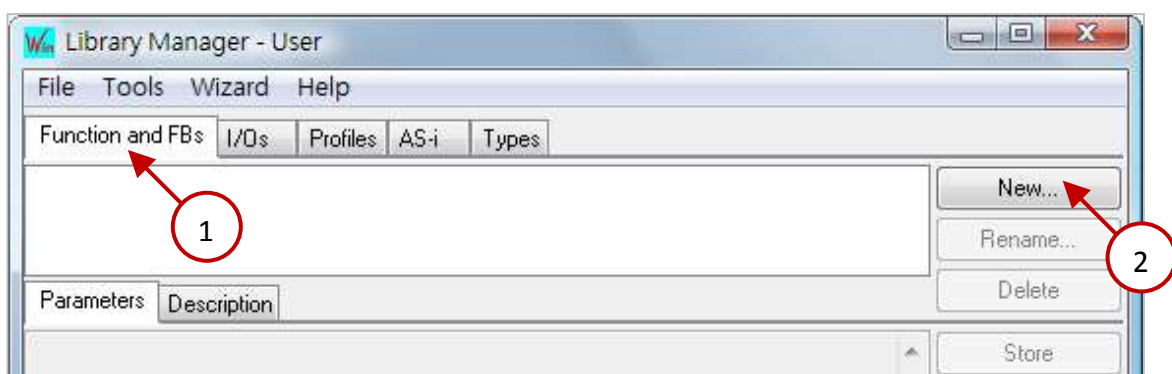
注意：若打開 Win-GRAF Workbench，會無法新增/編輯 Win-GRAF Library。



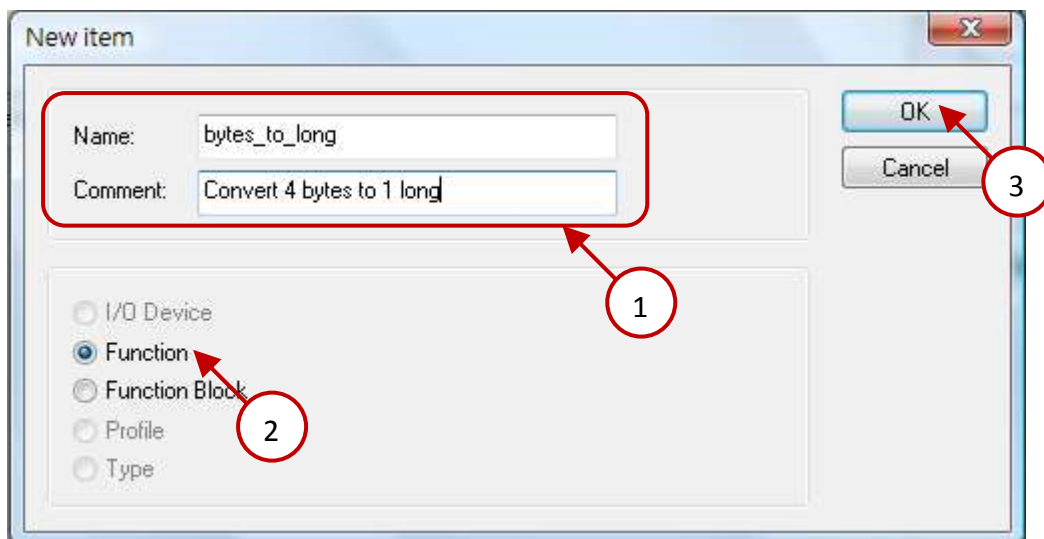
2. 建立一個名為 "User" 的 Library 群組，以方便維護及管理。



3. 選擇 "Function and FBs" 頁籤，並按下 "New" 按鈕。

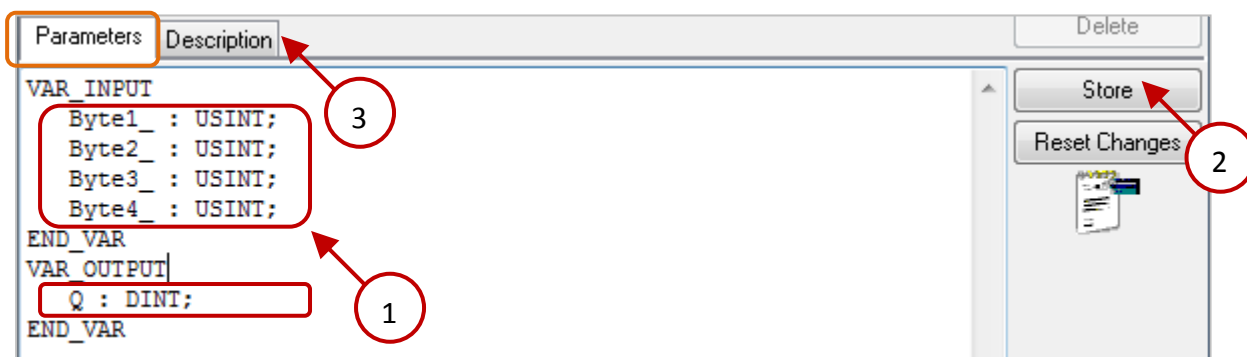


4. 填入 Function 的名稱、註解，並選擇 Function 的型態為“Function”，最後按下“OK”按鈕。

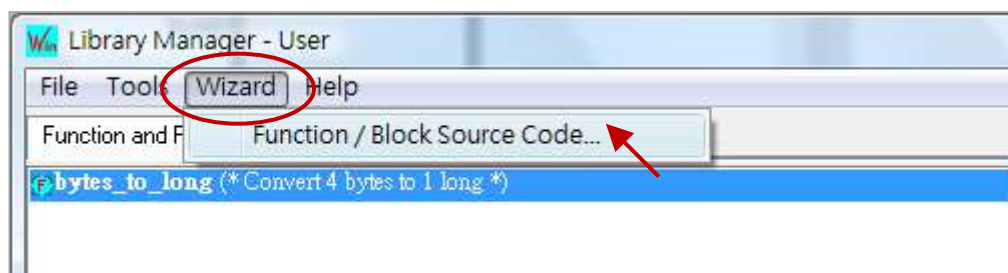


5. 宣告該 Function 的原型。

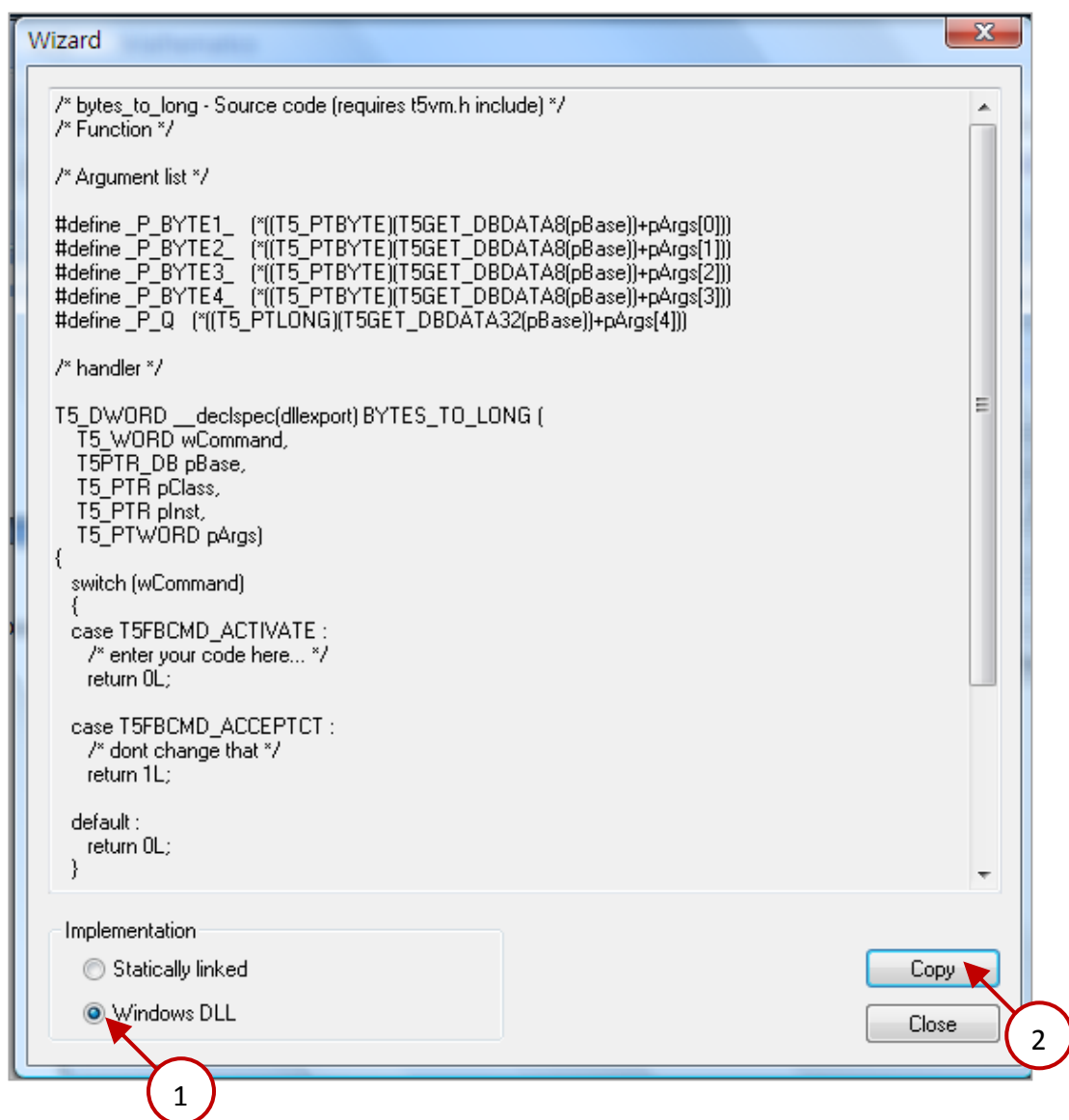
參數的宣告請依照“變數名稱：變數型態；”的方式來宣告參數，而變數型態可以參考[附錄 A](#)，“VAR_INPUT”與“END_VAR”之間為傳入參數；“VAR_OUTPUT”與“END_VAR”之間為回傳參數，完成後按“Store”按鈕儲存，再點選“Description”頁籤，可在此處編輯該 Function 的技術說明，完成後也請按“Store”按鈕儲存。



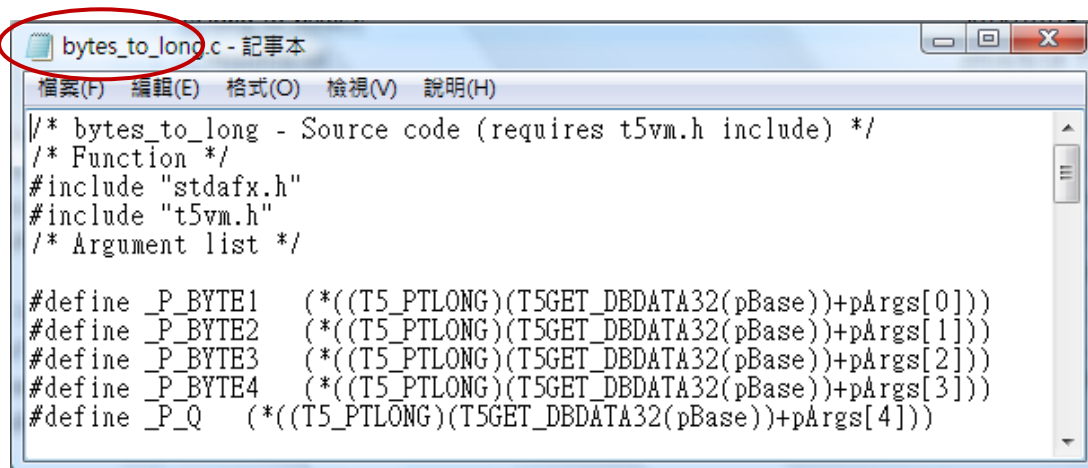
6. 點選“Wizard”→“Function/Block Source Code”產生該 Function 的 Source Code。



7. 點選 “Windows DLL” 按下 “Copy” 按鈕，複製產生的 Source Code。



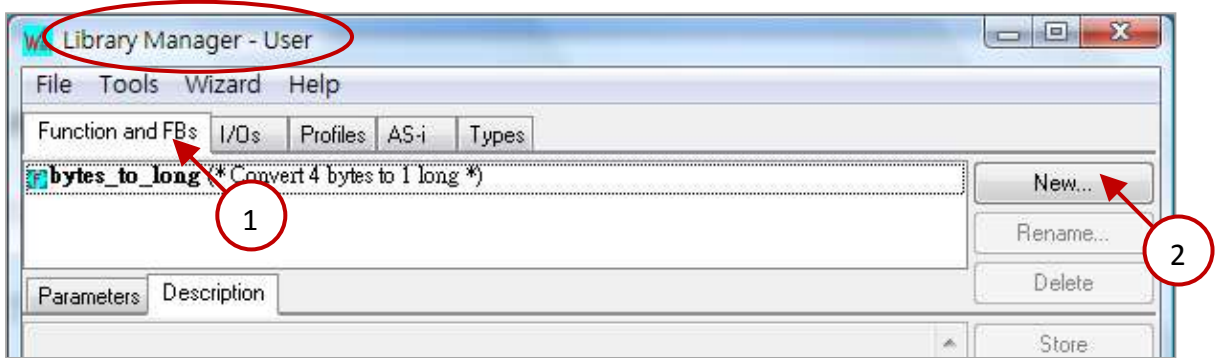
8. 新增一個純文字檔案且命名為 “bytes_to_long.c”，並將剛剛複製的內容貼入並儲存以供備用。



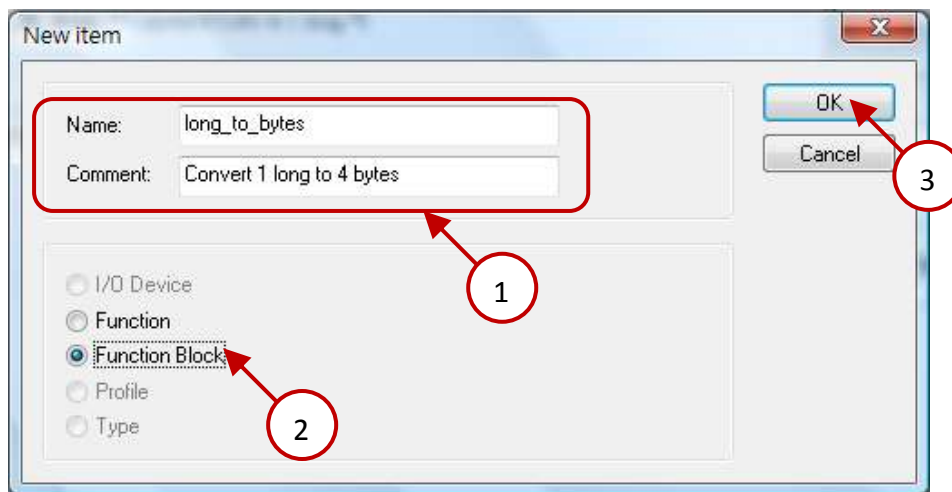
18.3.2 定義 Function Block Lib

本節的方法與前一節類似，本範例是要定義出 Function Block ("long_to_bytes") 的 Win-GRAF Lib 並產生出 Source code ("long_to_bytes.c") 檔案。

1. 參考 [前一節](#)，開啟 "Library Manager" (程式集 → Win-GRAF → **Libraries** → **OEM**)，再開啟先前新增的 Library 群組 - "User"，並按下 "New" 按鈕來新增一個 Function Block。

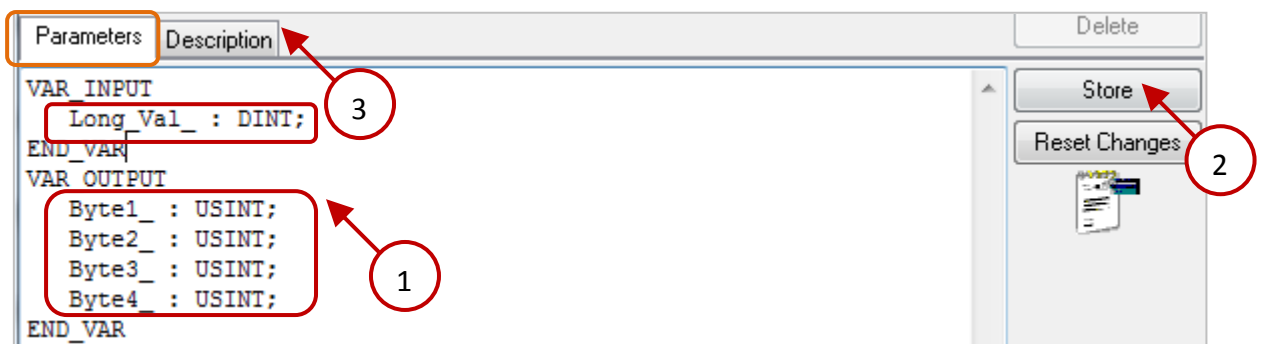


2. 填入 Function 的名稱、註解，並選擇型態為 "Function Block"，最後按下 "OK" 的按鈕。

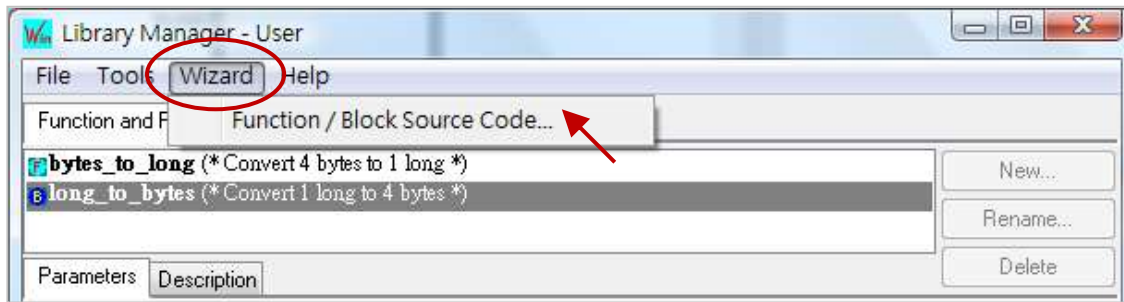


3. 宣告該 Function 的原型。

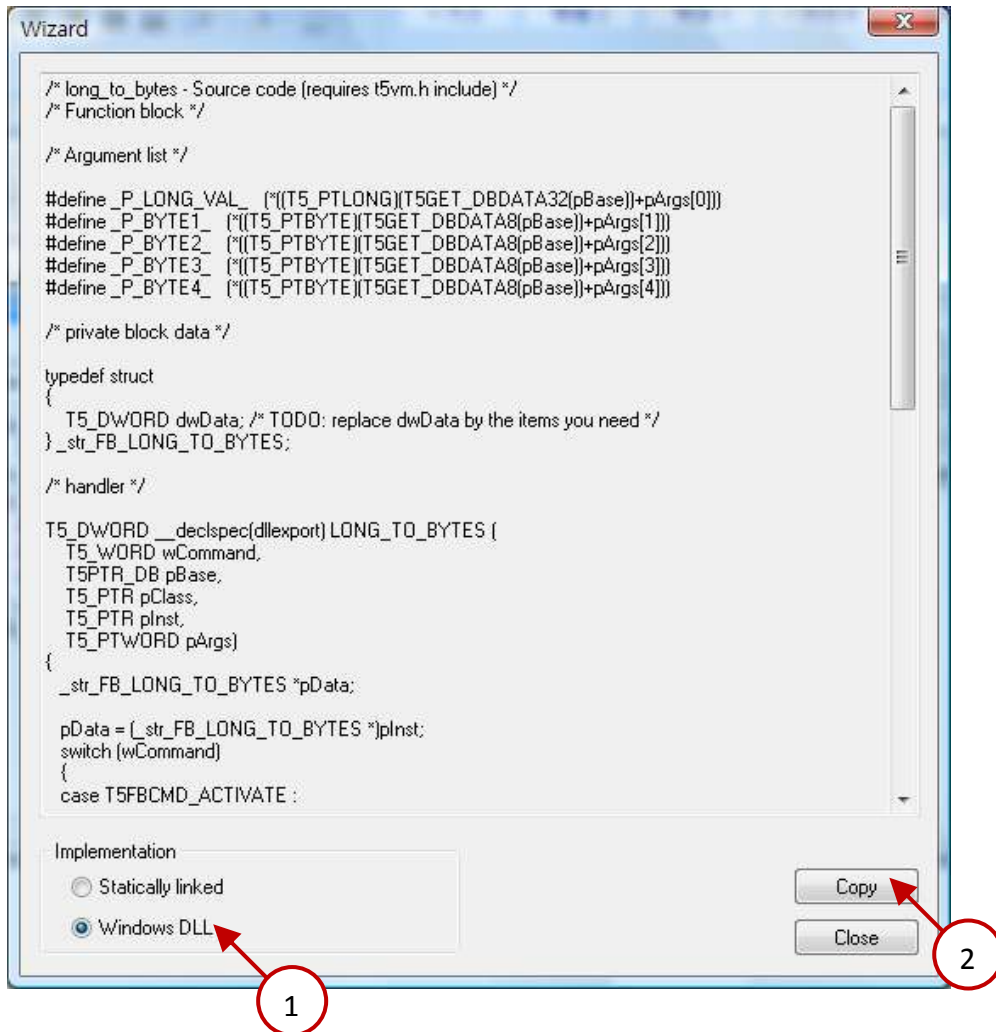
參數的宣告請依照 "變數名稱：變數型態；" 的方式來宣告參數，而變數型態可以參考 [附錄 A](#)，"VAR_INPUT" 與 "END_VAR" 之間為傳入參數；"VAR_OUTPUT" 與 "END_VAR" 之間為回傳參數，完成後按 "Store" 按鈕儲存，再點選 "Description" 頁籤，可在此處編輯該 Function Block 的技術說明，完成後也請按 "Store" 按鈕儲存。



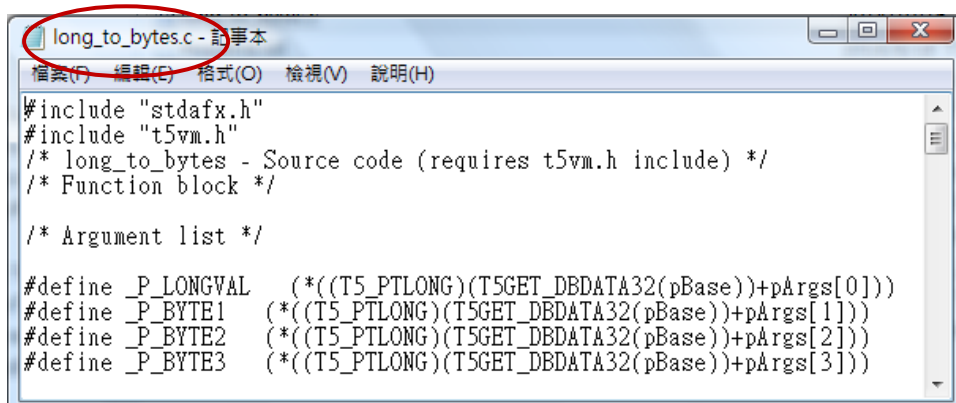
4. 點選 “Wizard” → “Function/Block Source Code” 產生該 Function Block 的 Source Code 。



5. 點選 “Windows DLL” 按下 “Copy” 按鈕，複製產生的 Source Code 。



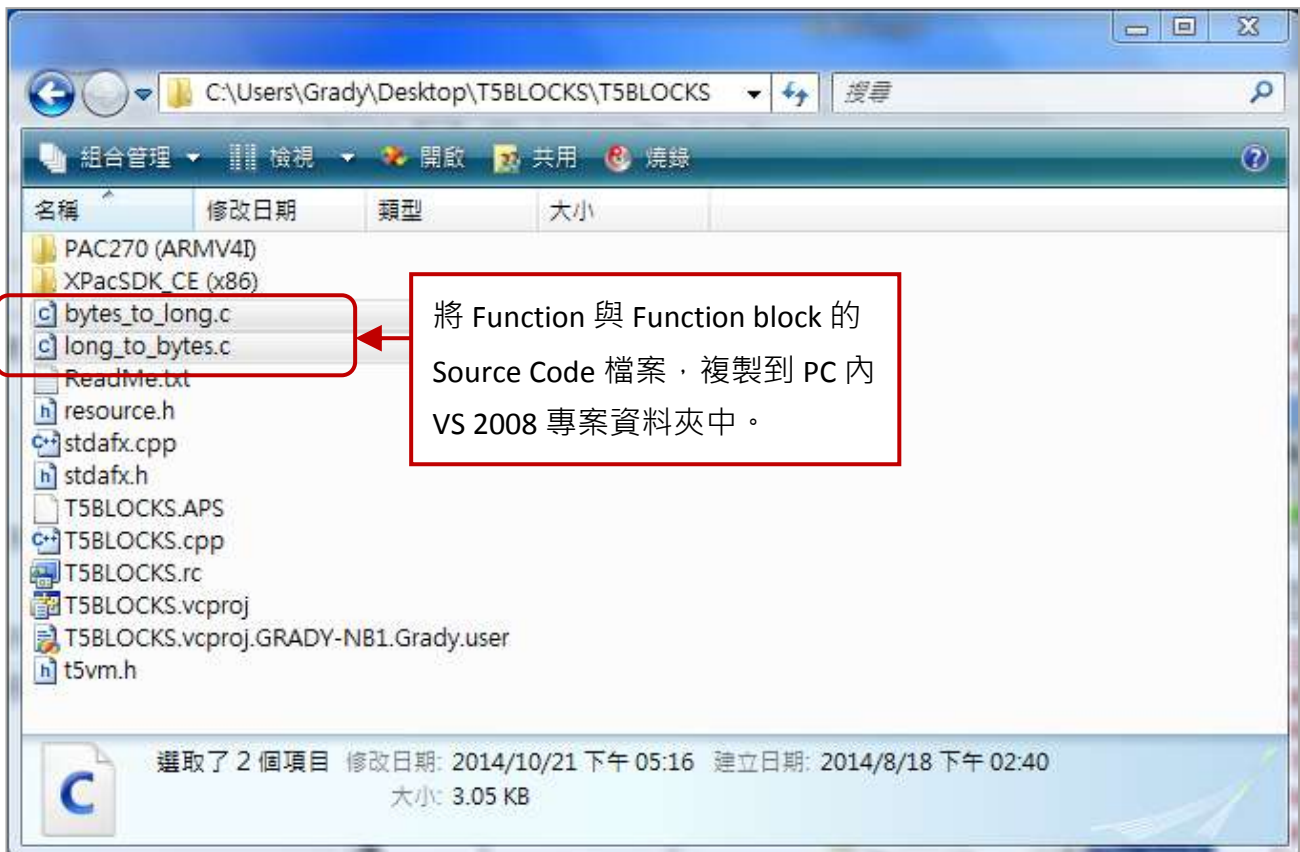
6. 新增一個純文字檔案且命名為 “long_to_bytes.c”，並將剛剛複製的內容貼入並儲存以供備用。



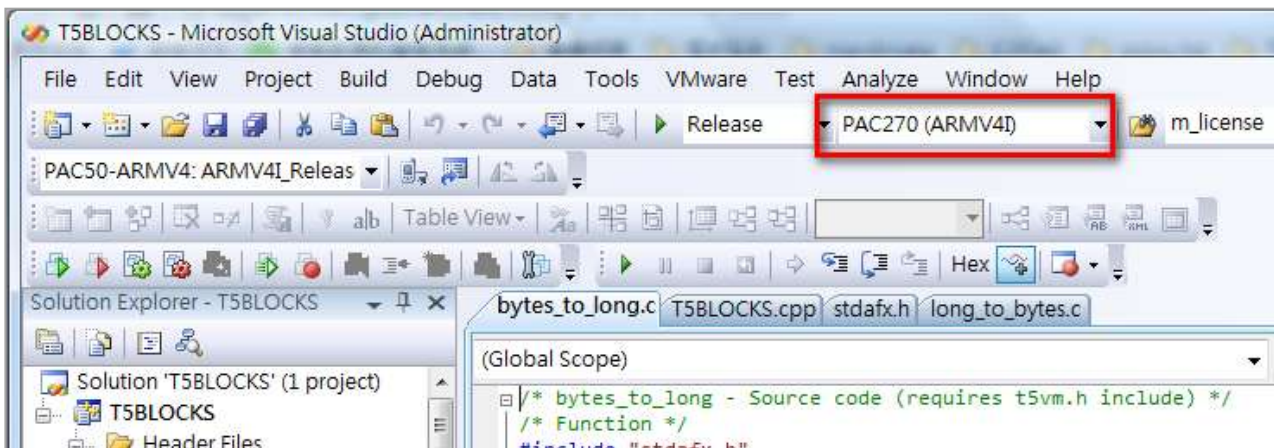
18.4 編寫 Function 或 Function Block 的邏輯程式

注意: 請確認您的 PC 內已經安裝了 Visual Studio 2008 軟體且已安裝 WinPAC SDK / XPAC SDK，才能進行以下步驟。

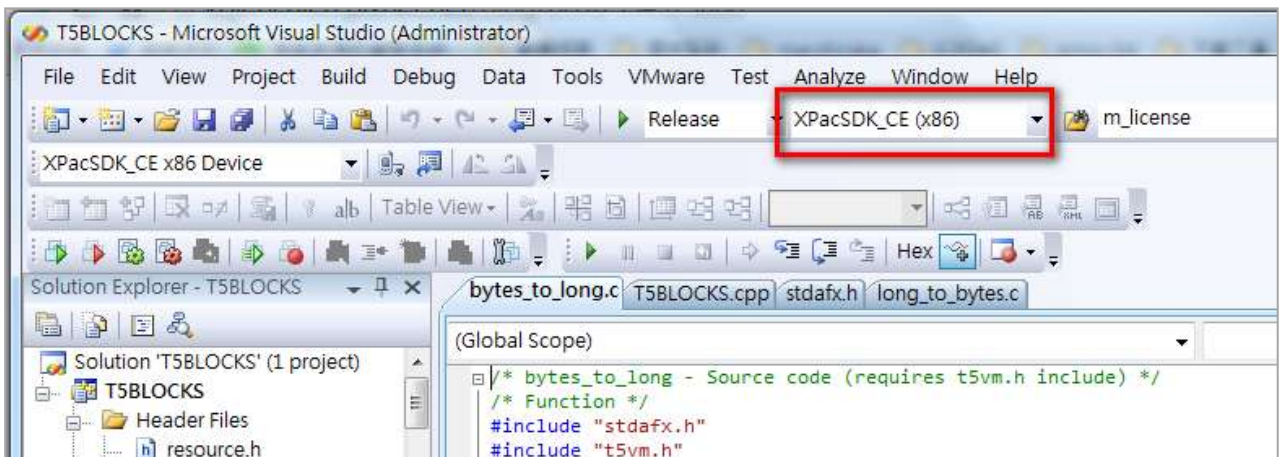
1. 把剛剛產生的 Function 與 Function block 的 Source Code 檔案 (如圖)，複製到 PC 內相對應的專案資料夾內。(您可在光碟的路徑中找到該範例專案:
\\napdos\Win-GRAF\demo-project\user_c_lib\demo_user_c)



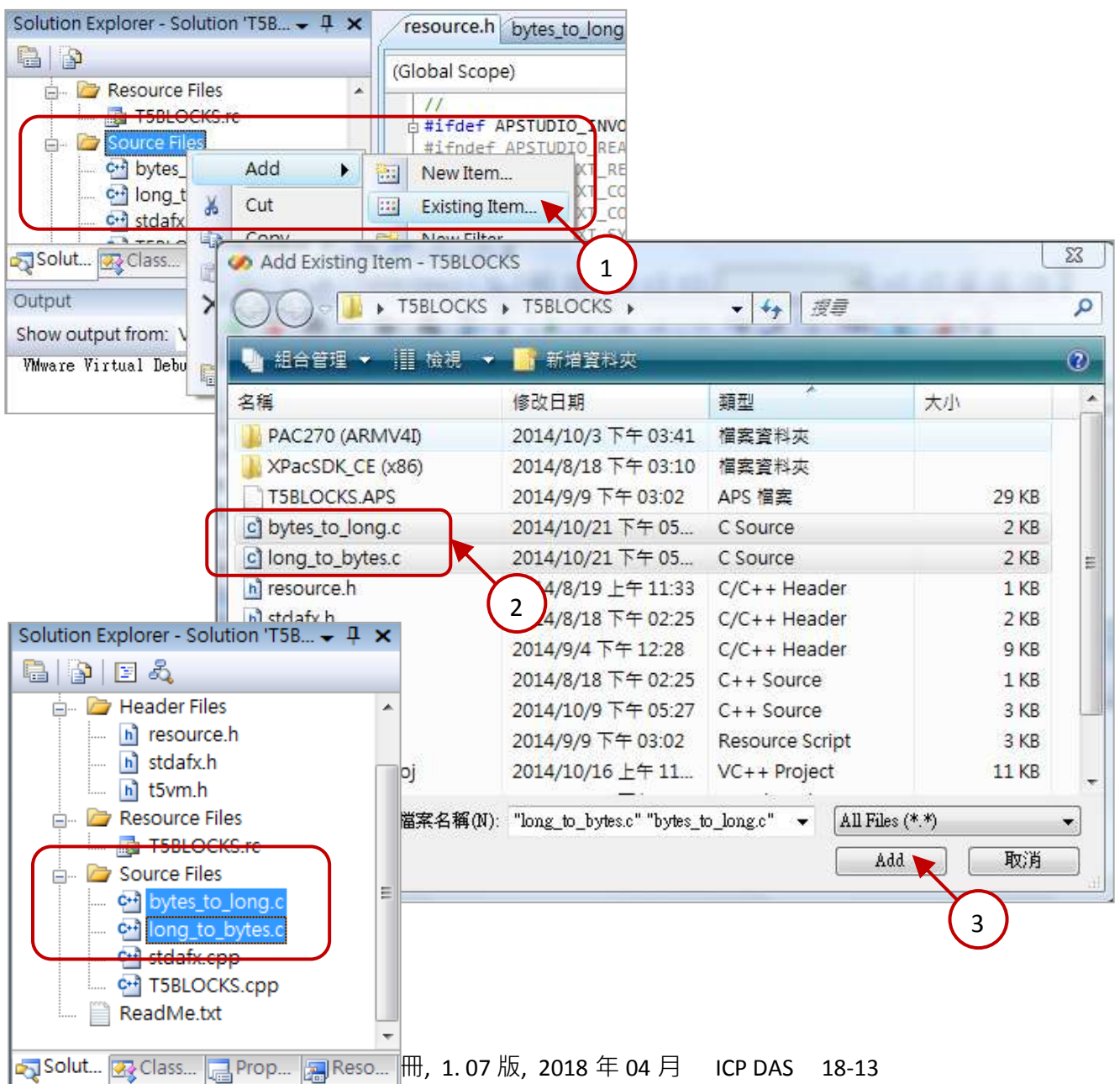
2. 接下來，需確認 VS 2008 的 Project 設定是否正確? (**注意:** 不同的 PAC 採用的設定會不相同) 若 PAC 使用 WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7 或 WP-5xx8-CE7 必需設定為 “PXA270(ARMV4I)”。



若 PAC 使用 XP-8xx8-CE6 必需設定為 "XPacSDK (x86)" 。

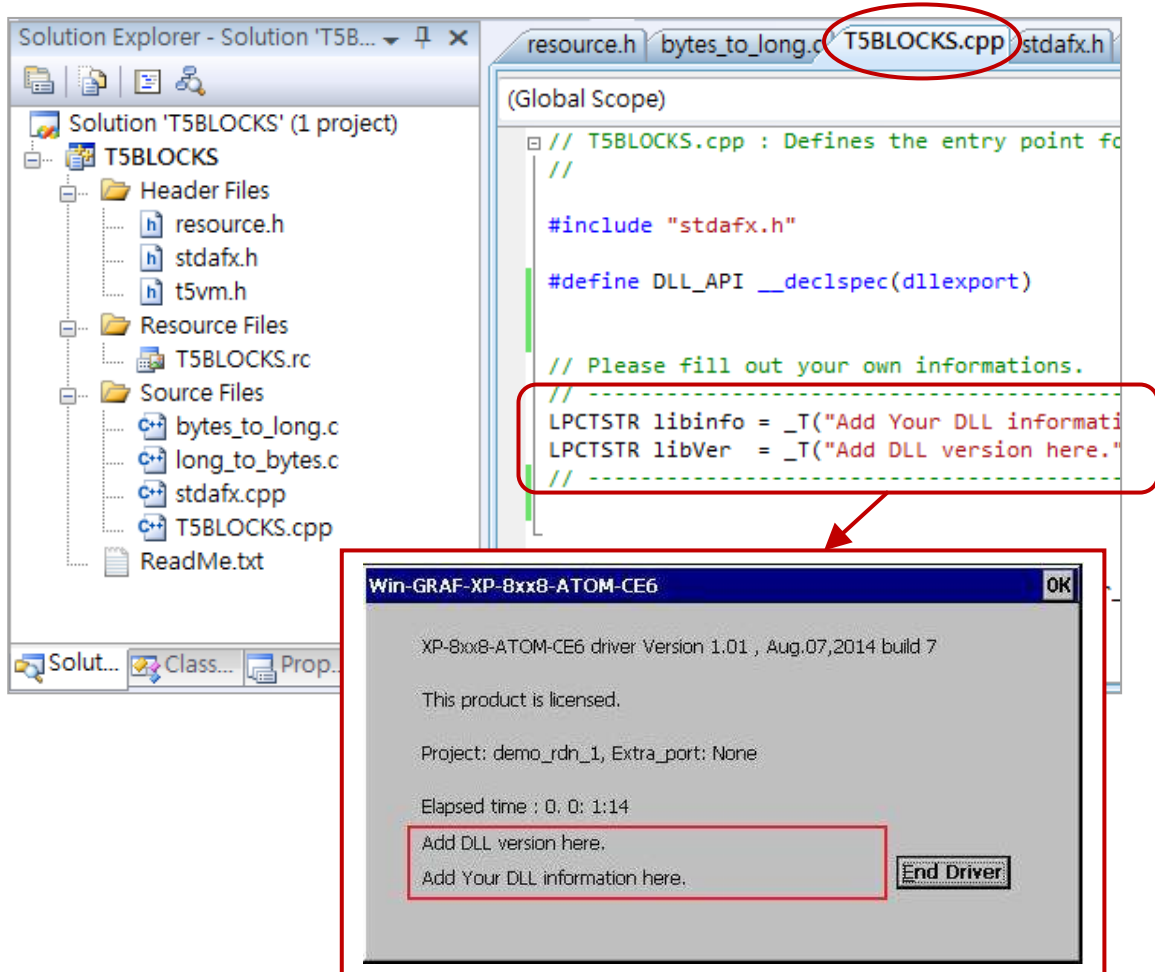


3. 將 Function 與 Function block 的 C 檔案加入此 VS 2008 Project 內。在 "Source Files" 上叫出右鍵選單 → "Add" → "Existing Item..."，並選擇剛剛加入的 C 檔案，再按下 "Add" 按鈕。

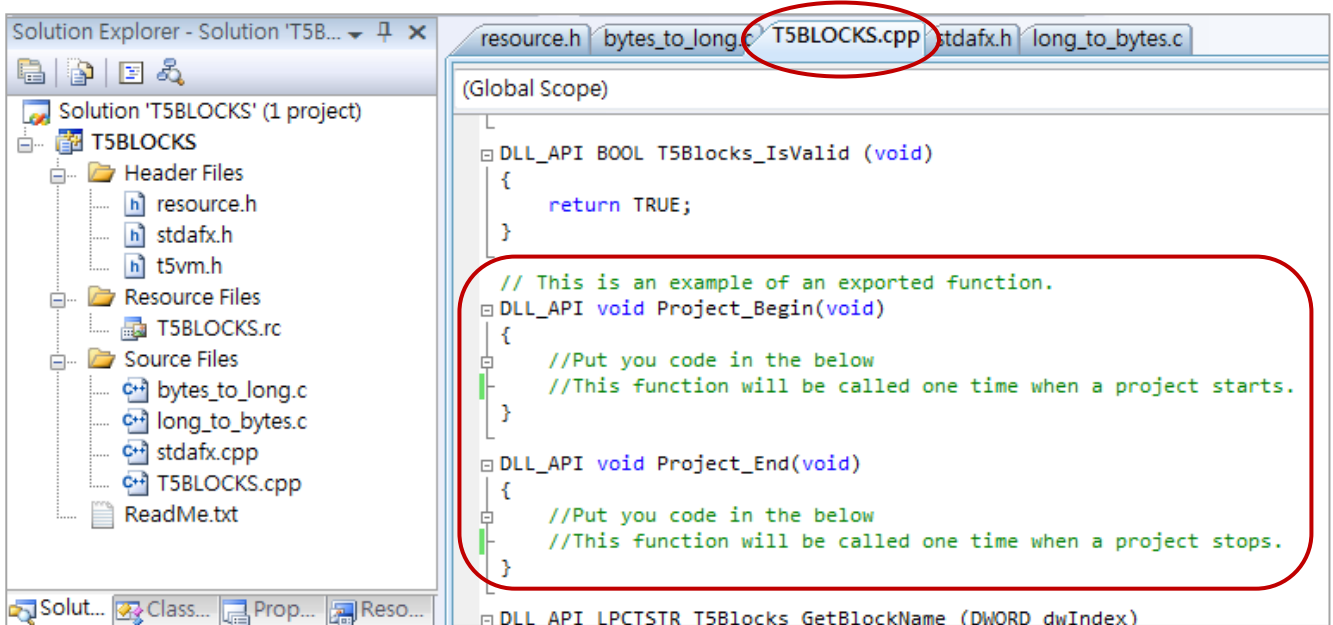


18.4.1 編寫 “T5BLOCKS.cpp”

將 “T5BLOCKS.cpp” 內的 “libinfo” 與 “libVer” 修改為您想要顯示的資訊，這 2 項資訊會顯示在 PAC 內 Win-GRAF Driver 的視窗上 (如下圖)。

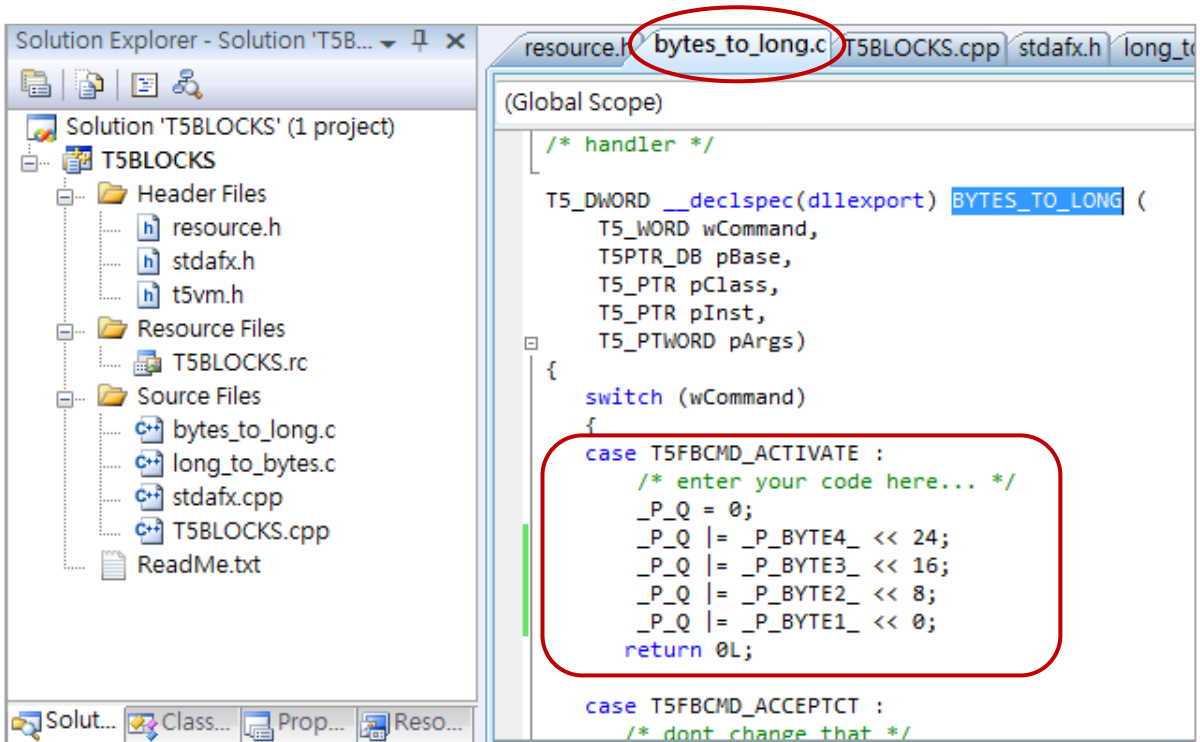


如果您想要在 Win-GRAF 專案程式 “一開始運行” 與 “結束運行前” 進行一些動作，請編寫 “T5BLOCKS.cpp” 內 “Project_Begin” 與 “Project_End” 這 2 個函式的程式。

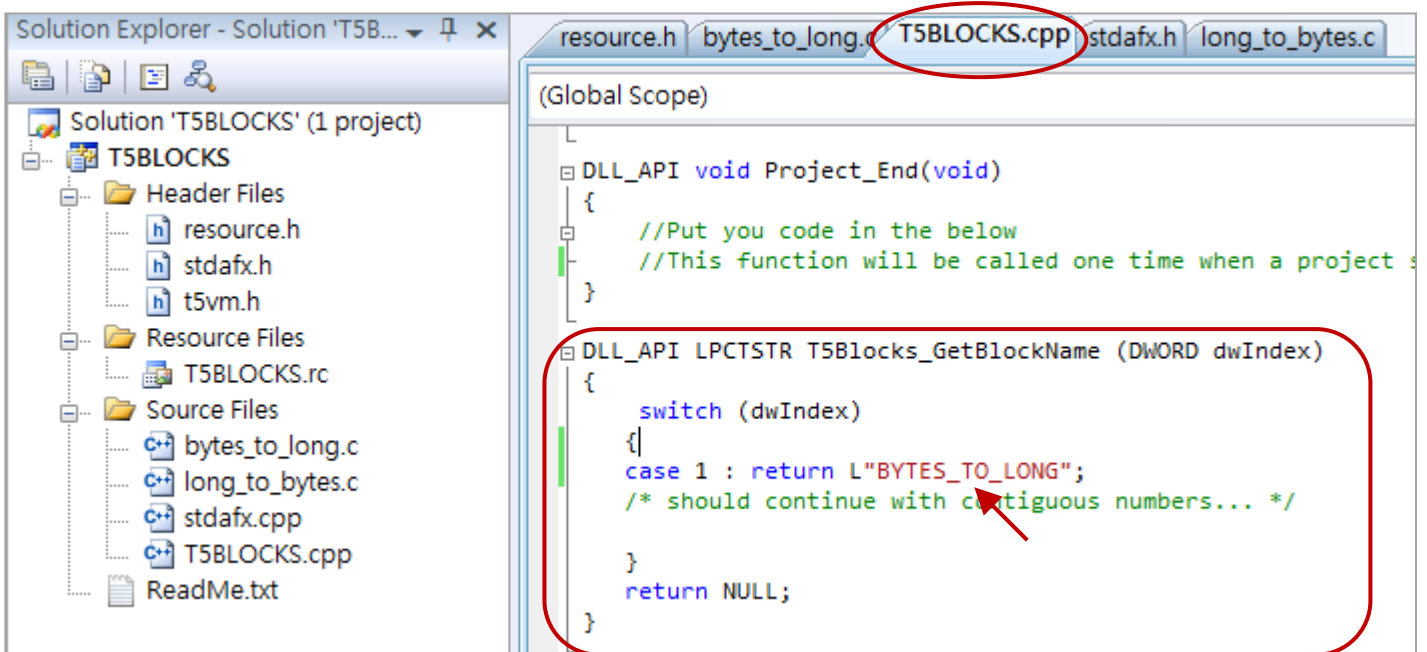


18.4.2 編寫 Function 內的邏輯程式 (本例是 “bytes_to_long.c”)

先將 Function 的處理邏輯寫在 switch case - "T5FBCMD_ACTIVE" 指令中。



再將此 Function 的名稱 (“BYTES_TO_LONG”) 加到 “T5BLOCKS.cpp” 檔案內 "T5Blocks_GetBlockName" 的 switch case 中。**注意:** switch case 的編號一定要由 1 開始，且不得跳號。



18.4.3 編寫 Function Block 內的邏輯程式 (本例是 “long_to_bytes.c”)

```
typedef struct  
{  
    T5_DWORD dwData; /* TODO: replace dwData by the items you need */  
} _str_FB_LONG_TO_BYTES;
```

每個 Function Block 都有一個配置好的 “Private structure” 記憶體空間，User 可依照自己的需求來做運用 或 改寫。

```
T5_DWORD __declspec(dllexport) LONG_TO_BYTES (  
    T5_WORD wCommand,  
    T5PTR_DB pBase,  
    T5_PTR pClass,  
    T5_PTR pInst,  
    T5_PTWORD pArgs)  
{  
    _str_FB_LONG_TO_BYTES *pData;  
  
    pData = (_str_FB_LONG_TO_BYTES *)pInst;  
    switch (wCommand)  
    {  
    case T5FBCMD_ACTIVATE :  
        /* activates the function block */  
        /* enter your code here... */  
        _P_BYTE1_ = _P_LONG_VAL_ & 0xFF;  
        _P_BYTE2_ = (_P_LONG_VAL_ >> 8) & 0xFF;  
        _P_BYTE3_ = (_P_LONG_VAL_ >> 16) & 0xFF;  
        _P_BYTE4_ = (_P_LONG_VAL_ >> 24) & 0xFF;  
        return 0L;  
  
    case T5FBCMD_INITINSTANCE :  
        /* initialize private data */  
        /* enter your code here... */  
        return 0L;  
  
    case T5FBCMD_EXITINSTANCE :  
        /* release private data */  
        /* enter your code here... */  
        return 0L;  
  
    case T5FBCMD_HOTRESTART :  
        /* actuate private data for hot restart */  
        /* enter your code here... */  
        return 0L;  
  
    case T5FBCMD_SIZEOFINSTANCE :  
        /* dont change that */  
        return (T5_DWORD)sizeof(_str_FB_LONG_TO_BYTES);  
  
    case T5FBCMD_ACCEPTCT :  
        /* dont change that */  
        return 1L;  
  
    default :  
        return 0L;  
    }  
}
```

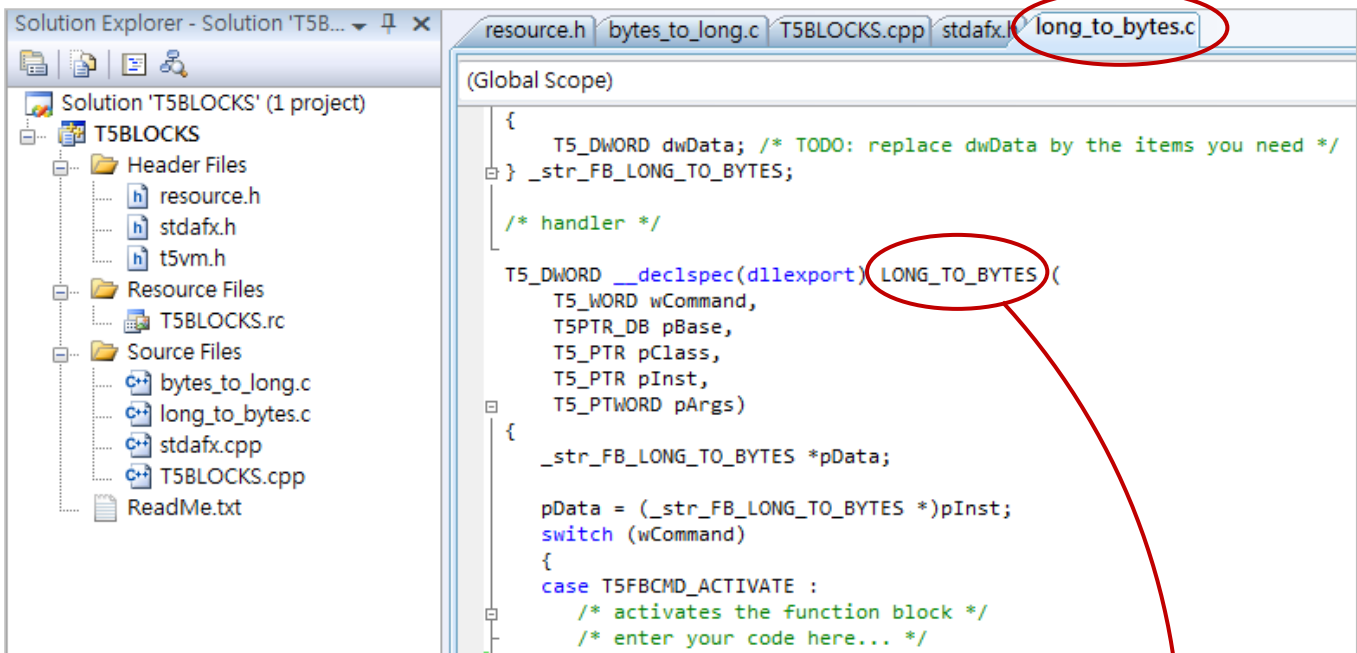
每個 Cycle 都會執行該區塊。

當 Project 啟動時，主要用來初始化 “Private structure” 實體。

當 Project 停止時，主要用來關閉 “Private structure” 實體。

當 Project 熱啟動 (Hot restart) 或 On Line Change 時，可以更新 “Private structure” 的資料。

最後，記得將該 Function Block 的名稱 ("LONG_TO_BYTES") 加到 "T5BLOCKS.cpp" 檔案內的 "T5Blocks_GetBlockName" 的 switch case 中。



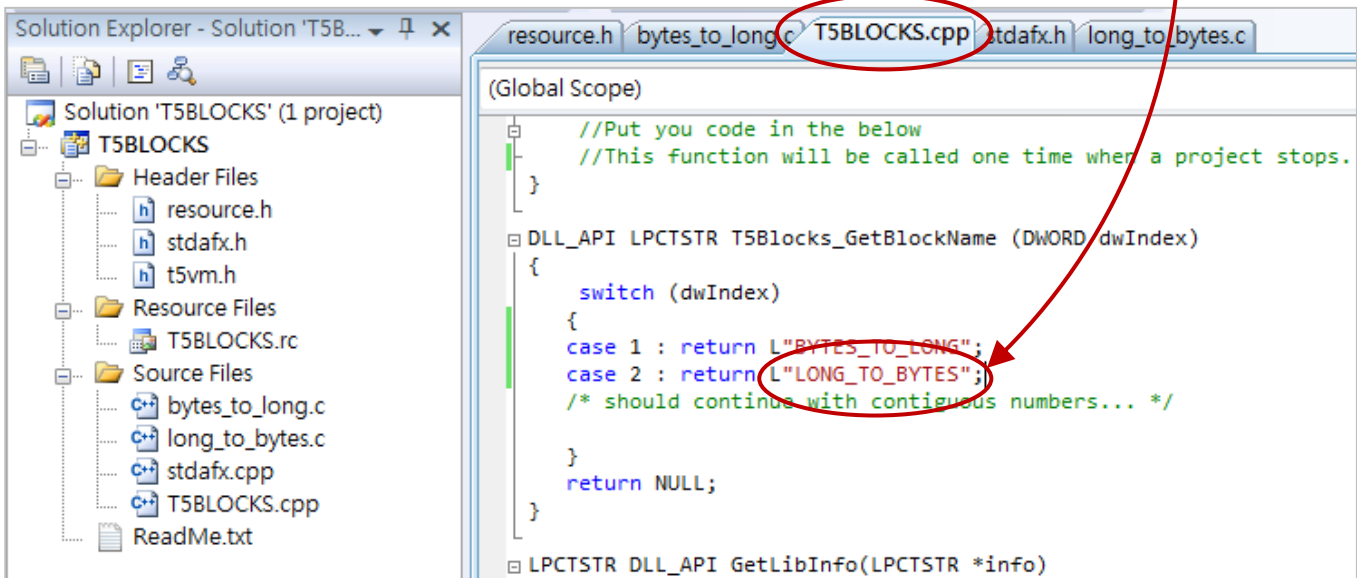
```
resource.h bytes_to_long.c T5BLOCKS.cpp stdafx.h long_to_bytes.c
(Global Scope)
{
    T5_DWORD dwData; /* TODO: replace dwData by the items you need */
} _str_FB_LONG_TO_BYTES;

/* handler */

T5_DWORD __declspec(dllexport) LONG_TO_BYTES (
    T5_WORD wCommand,
    T5PTR_DB pBase,
    T5_PTR pClass,
    T5_PTR pInst,
    T5_PTRWORD pArgs)
{
    _str_FB_LONG_TO_BYTES *pData;

    pData = (_str_FB_LONG_TO_BYTES *)pInst;
    switch (wCommand)
    {
    case T5FBCMD_ACTIVATE :
        /* activates the function block */
        /* enter your code here... */
```

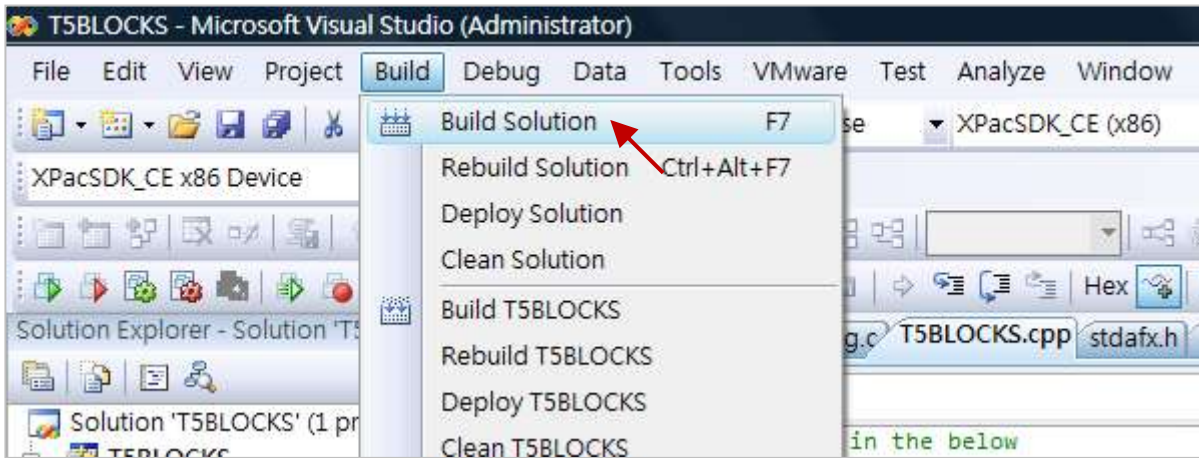
(注意: switch case 的編號一定要由 1 開始，且不得跳號。)



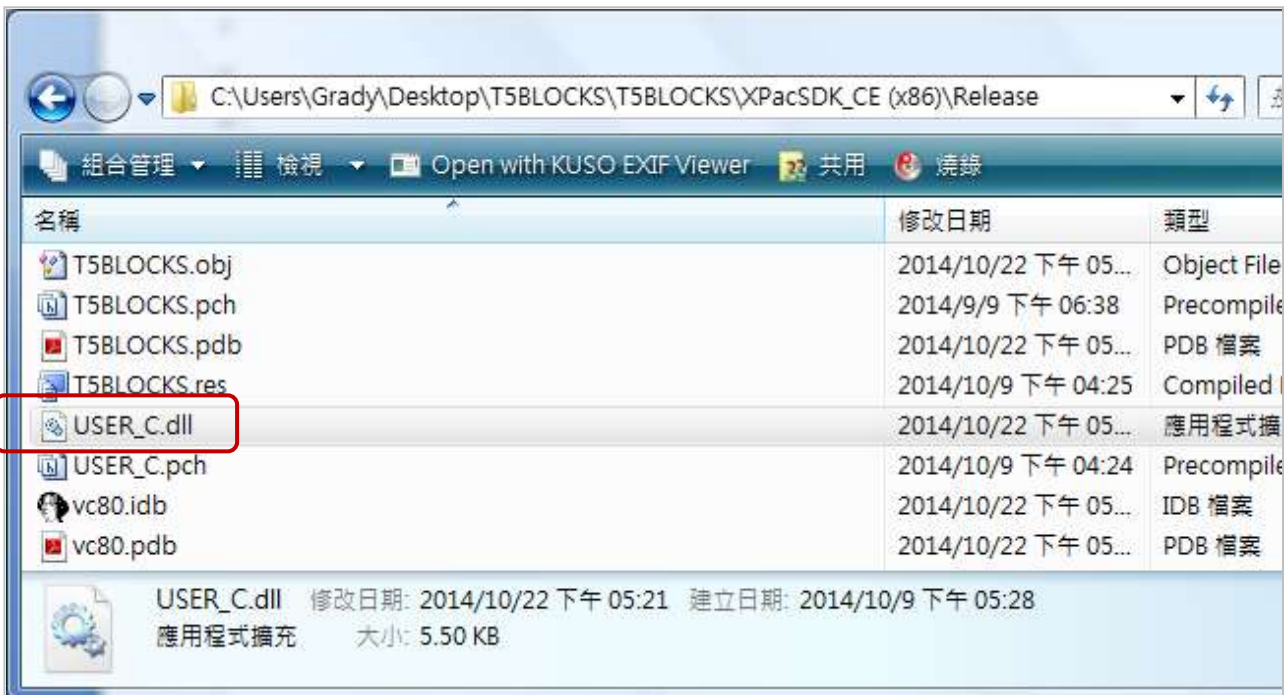
```
resource.h bytes_to_long.c T5BLOCKS.cpp stdafx.h long_to_bytes.c
(Global Scope)
//Put you code in the below
//This function will be called one time when a project stops.
}
DLL_API LPCTSTR T5Blocks_GetBlockName (DWORD dwIndex)
{
    switch (dwIndex)
    {
    case 1 : return L"BYTES_TO_LONG";
    case 2 : return L"LONG_TO_BYTES";
    /* should continue with contiguous numbers... */
    }
    return NULL;
}
LPCTSTR DLL_API GetLibInfo(LPCTSTR *info)
```


18.4.4 編譯專案

1. 點選功能表 "Build" > "Build Solution" 來執行編譯。



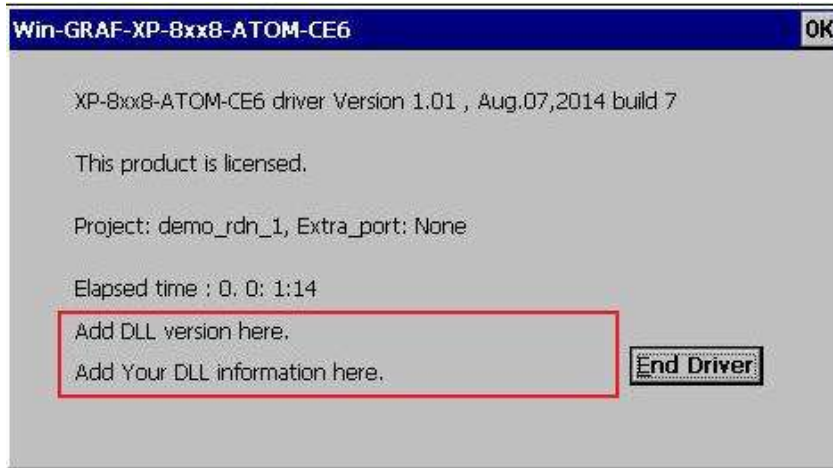
2. 若編譯成功，複製 "USER_C.dll" 到您的 PAC 內的 "\\System_disk\Win-GRAF\" 路徑下，並重開機一次即可。



18.5 測試自定義的 Function 與 Function Block

1. 使用 FTP 的方式，將 "user_c.dll" 檔案放到 PAC 內與 Win-GRAF Driver 相同的資料夾中 (即，\System_disk\Win-GRAF)，再將控制器重新上電。

若 Win-GRAF Driver 有偵測到正確的 DLL 檔，會顯示如下：



2. 開啟 Win-GRAF Workbench 中含有自定義的 Function/Function Block 的專案，編譯後下載該專案至 Win-GRAF PAC 中。

另外，Win-GRAF PAC 隨貨光碟中含有一些預先準備好的檔案，也可以用來測試一下效果：

- (1) WP-8xx8 · VP-x2x8-CE7 · WP-5xx8-CE7:

\napdos\Win-GRAF\demo-project\user_c_lib\wp_vp\user_c.dll

- (2) XP-8xx8-CE6:

\napdos\Win-GRAF\demo-project\user_c_lib\xpac\user_c.dll

- (3) 將 Win-GRAF Library 資料夾 "User" 複製到 PC 的以下路徑中。

C:\Win-GRAF\DATA\HWDEF\

- (4) 開啟 Win-GRAF 範例專案 – "demo_user_c.zip"，編譯後下載該專案至 Win-GRAF PAC 中。

(可以參考 [13.1 節](#) 使用 Win-GRAF Workbench 來開啟範例專案。)

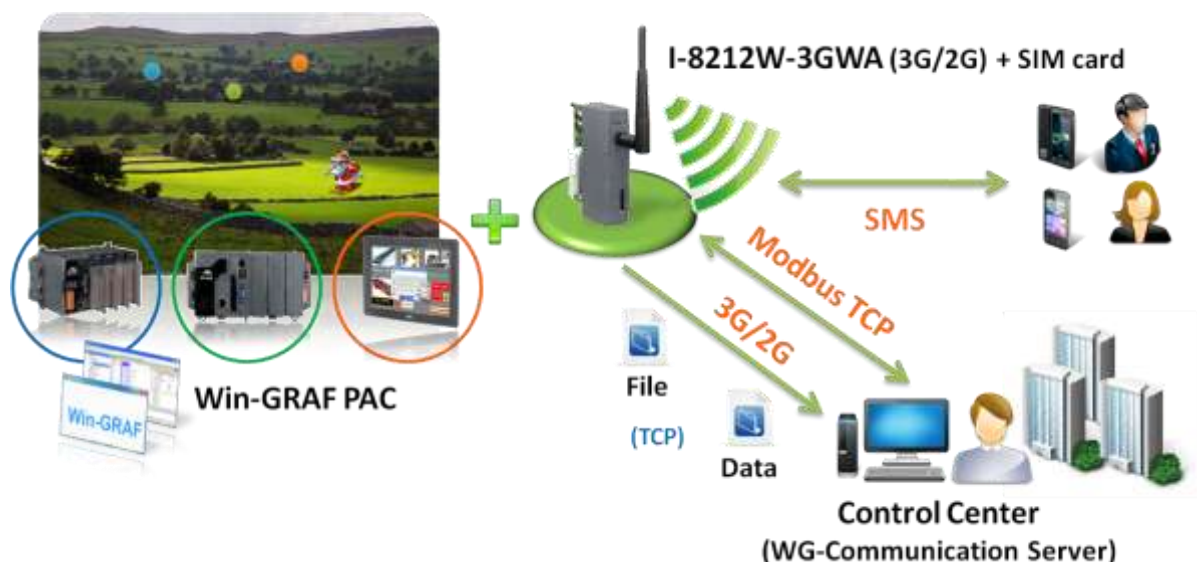
第 19 章 使用 3G 模組 – I-8212W-3GWA

註:

1. I-8212W-3GWA 因為認證問題，只能在某些地區銷售，請聯絡經銷商了解詳細情況。
2. 另外有一種 3G Solution 可供選擇，可以在當地購買 3G Router 加上 SIM 卡，這樣也能使 Win-GRAF PAC 可以利用 3G 來連上 Internet。

很多應用場合需要將採集到的資料透過網路傳回中控中心進行處理，但並沒有網路線可到達，或者佈網路線成本太高。對於此類應用，泓格科技推出 Win-GRAF PAC + I-8212W-3GWA 解決方案，設計者可以採用 Win-GRAF 軟體來編寫 PLC 應用程式 (Ladder, ST, Function Block, ...) 來採集 I/O 資料與其它數據資料，然後透過 I-8212W-3GWA (其內插上跟電信公司申請的有支持 3G/2G 的 SIM 卡) 撥接 3G/2G 上網，再透過發送 TCP 資料傳回控制中心。

3G/2G 無線通訊功能



從以下 PAC 的 Win-GRAF Driver 版本起，有支持 I-8212W-3GWA 撥接上網功能。

XP-8xx8-CE6 : 1.03 版起

WP-8xx8 : 1.05 版起

VP-x2x8-CE7 : 1.01 版起

若您的 Win-GRAF PAC Driver 是比上方還要早期的版本，請於網站下載最新的 Win-GRAF Driver，然後更新到您的 PAC 內。網址為:

http://www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-Driver.html

I-8212W-3GWA : <http://m2m.icpdas.com/i-8212w-3GWA.html>

19.1 硬體安裝與設定

I-8212W-3GWA 支持 3G/2G 無線通訊，請把跟電信公司申請的有 3G/2G 功能的 SIM 卡插入該模組的“SIM Card”插槽內，並把 I-8212W-3GWA 的天線裝好。

若 PAC 是 XP-8xx8-CE6 請將 I-8212W-3GWA 插在 Slot 1。

(即，XPC 最左方的 I/O 插槽。)

若 PAC 是 WP-8xx8 或 VP-x2x8-CE7，請將 I-8212W-3GWA 插在 Slot 0。

(即，WinPAC 最左方的 I/O 插槽，ViewPAC 則要看背面的 I/O 插槽編號。)

接下來將 PAC 開機，運行 PAC 的 Utility (比如 XPAC 就是“XPAC_UTILITY”) 來設定啟用 I-8212W-3GWA 在 COM6 這個串口上。記得最後要 Run “File > Save and Reboot” 將所做的設定存起來。

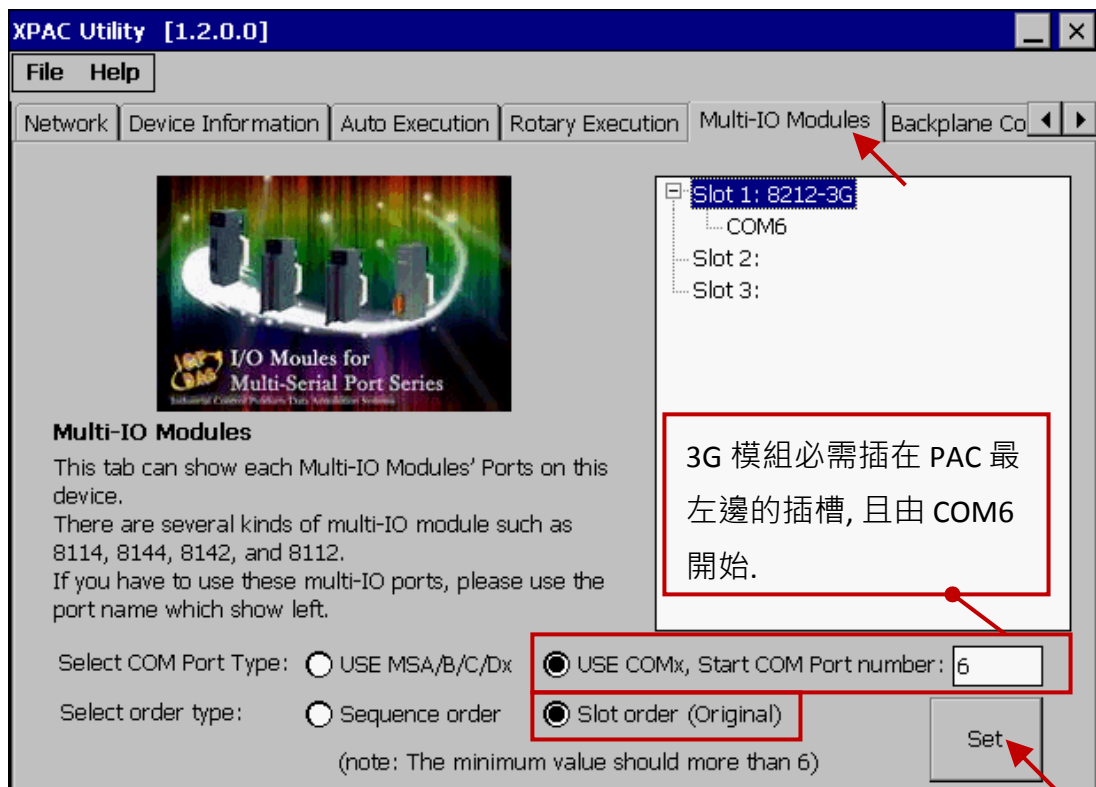
您可在 PAC 桌面或路徑下 “\System_Disk\Tools\” 找到 Utility (例如：“XPAC_UTILITY”)，也可下載最新的版本。

XPAC: ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/system_disk/tools/

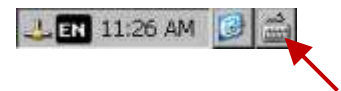
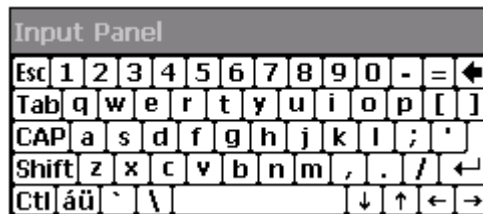
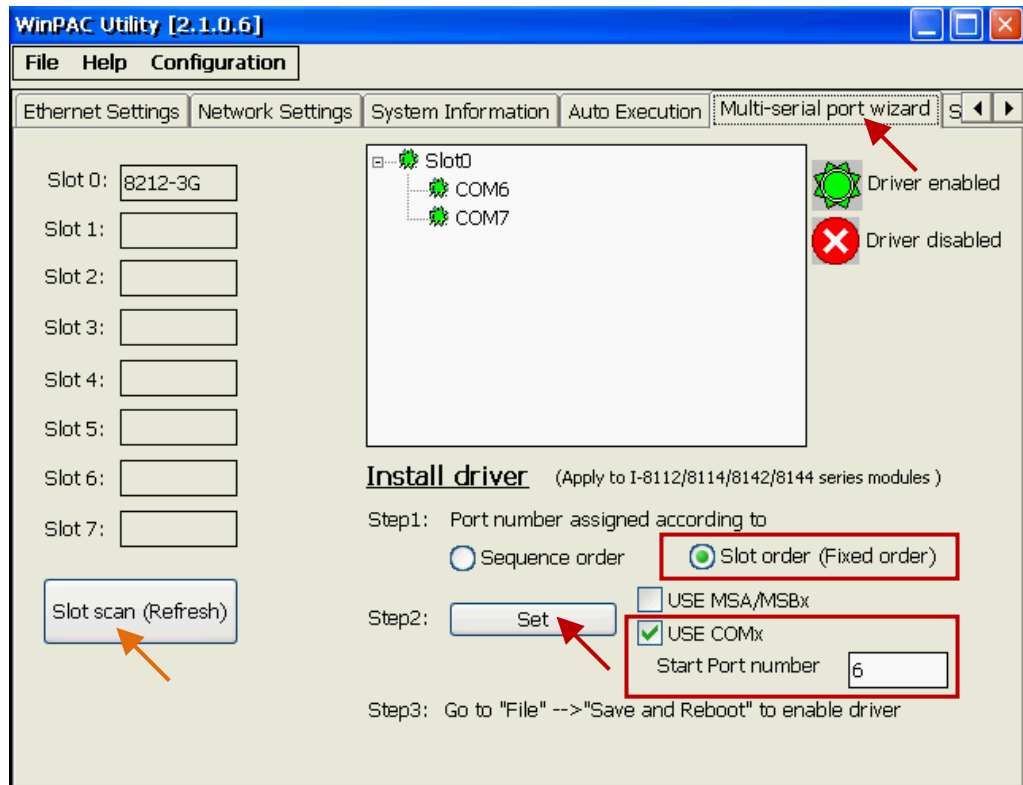
WinPAC: ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/system_disk/tools/

ViewPAC: ftp://ftp.icpdas.com/pub/cd/winpac/napdos/vp-4000_ce50/system_disk/tools/

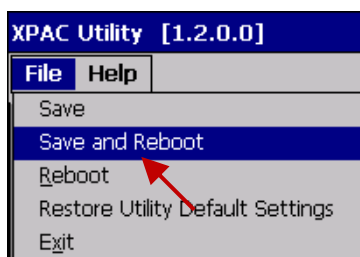
XPAC_UTILITY:



WinPAC_UTILITY:



最後，點選 PAC Utility 的 “File > Save and Reboot” 儲存設定並重新開機。



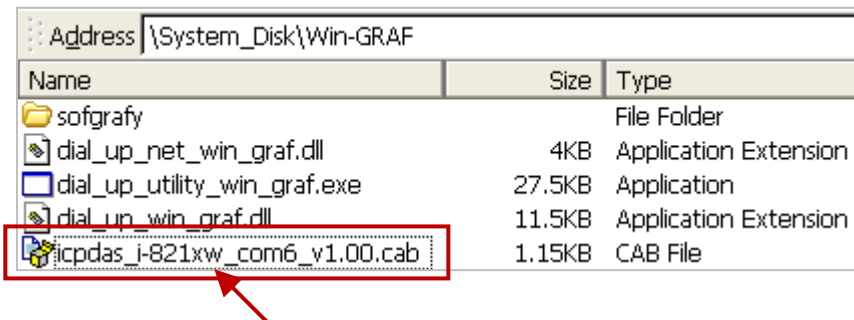
19.2 軟體安裝

請確認您的 PAC 的 Win-GRAF Driver 是否符合 [第 19 章](#) 第一頁所列的版本，若不符合，請更新它。

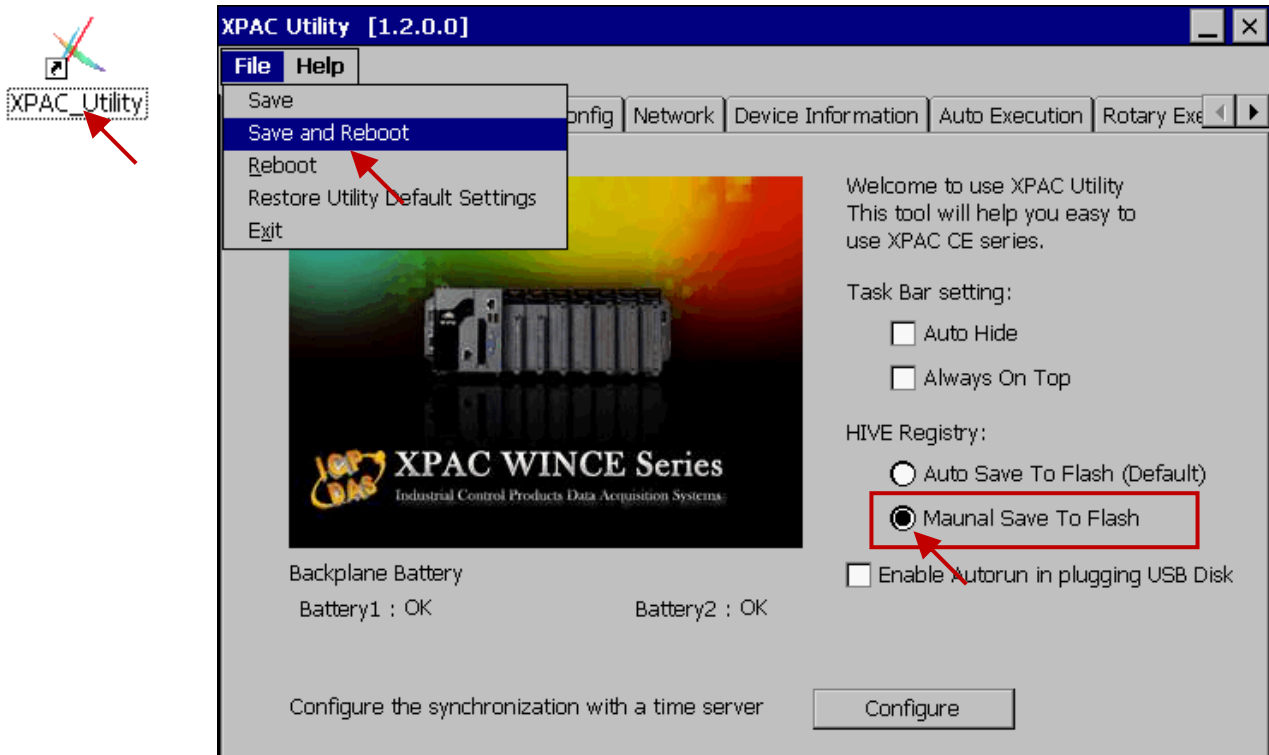
19.2.1 安裝 I-8212W-3GWA 的 Driver

注意：請把 I-8212W-3GWA 插在 WinPAC 與 ViewPAC 的 I/O 插槽 **0**，XP-8xx8-CE6 則是插槽 **1**。

若 PAC 是 XP-8xx8-CE6, WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7，請用滑鼠雙擊 PAC 路徑 `\System_Disk\Win-GRAF\` 內的 “icpdas_i-821xw_com6_vx.xx.cab” 來將 I-8212W-3GWA 的 **Driver** 安裝起來。

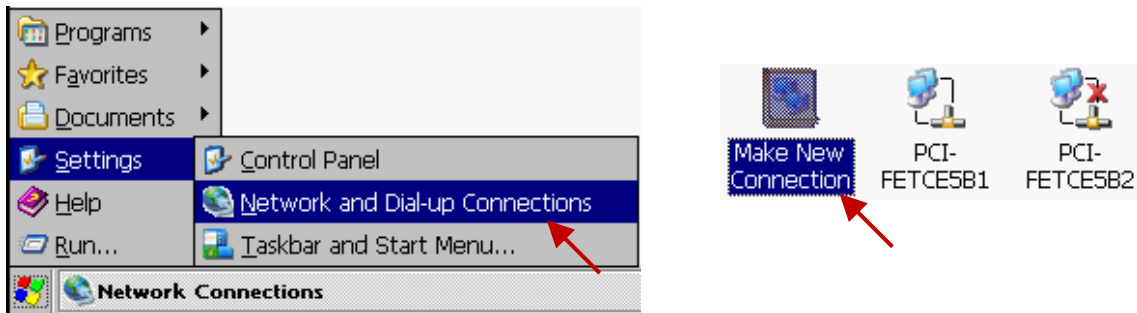


安裝好後，記得要運行 XPAC Utility (或 WinPAC Utility, ViewPAC Utility) 內的 “File > Save and Reboot” 讓該設定存起來，PAC 會自動開機一次。(下圖是以 XP-8xx8-CE6 為例子，請先選取 “Manual Save To Flash” 再執行 “File > Save and Reboot”)

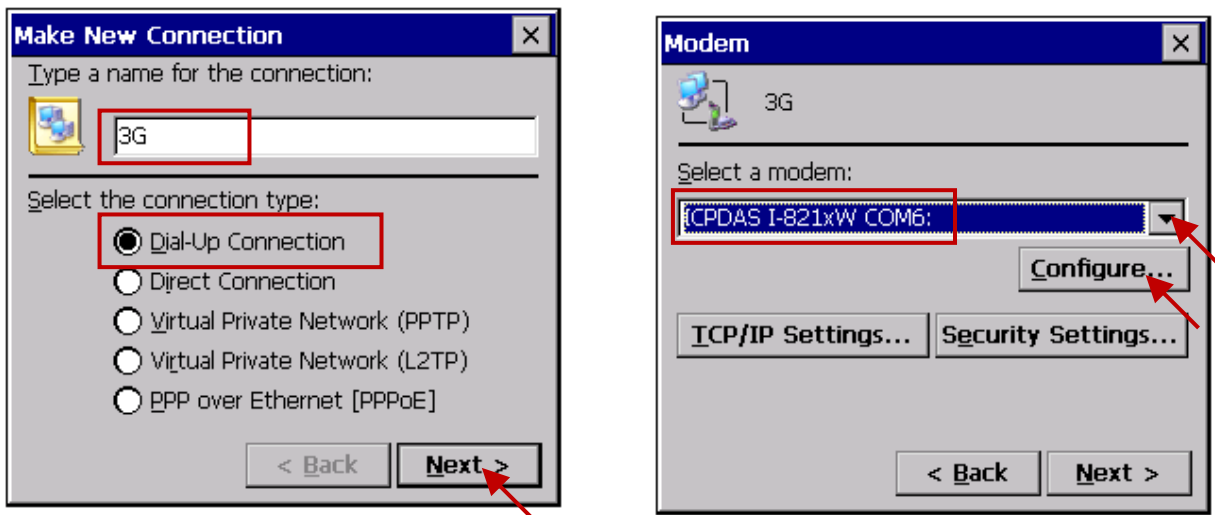


19.2.2 設定 3G/2G 撥接參數

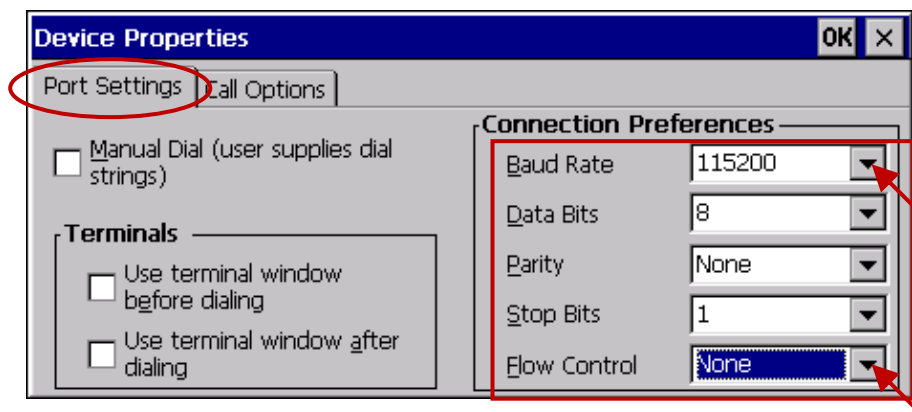
在 PAC 內，先進入“Network and Dial-up Connections”，之後 Run “Make New Connection”。



選取“Dial-Up Connection”，輸入一個英文名稱 (名稱內可含有數字 0~9)，下圖的名稱為“3G”。
按下“Next”按鈕後，選取“ICPDAS I-821xW COM6:”，然後進入“Configure...”。



在“Port Settings”內選取 Baud Rate 為 115200，Data Bits 為 8，Parity 為 None，Stop Bits 為 1，Flow Control 為 None。

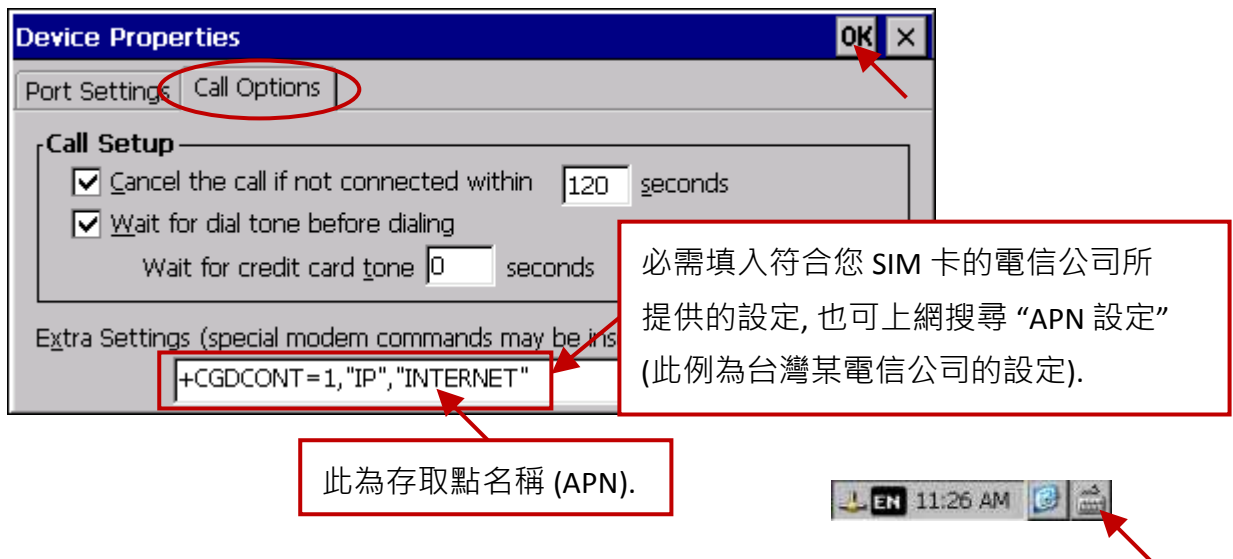


之後進入 “Call Options” 設定符合電信公司的 “Extra Settings”，比如在 Taiwan 某電信公司是使用
+CGDCONT=1,"IP","INTERNET"

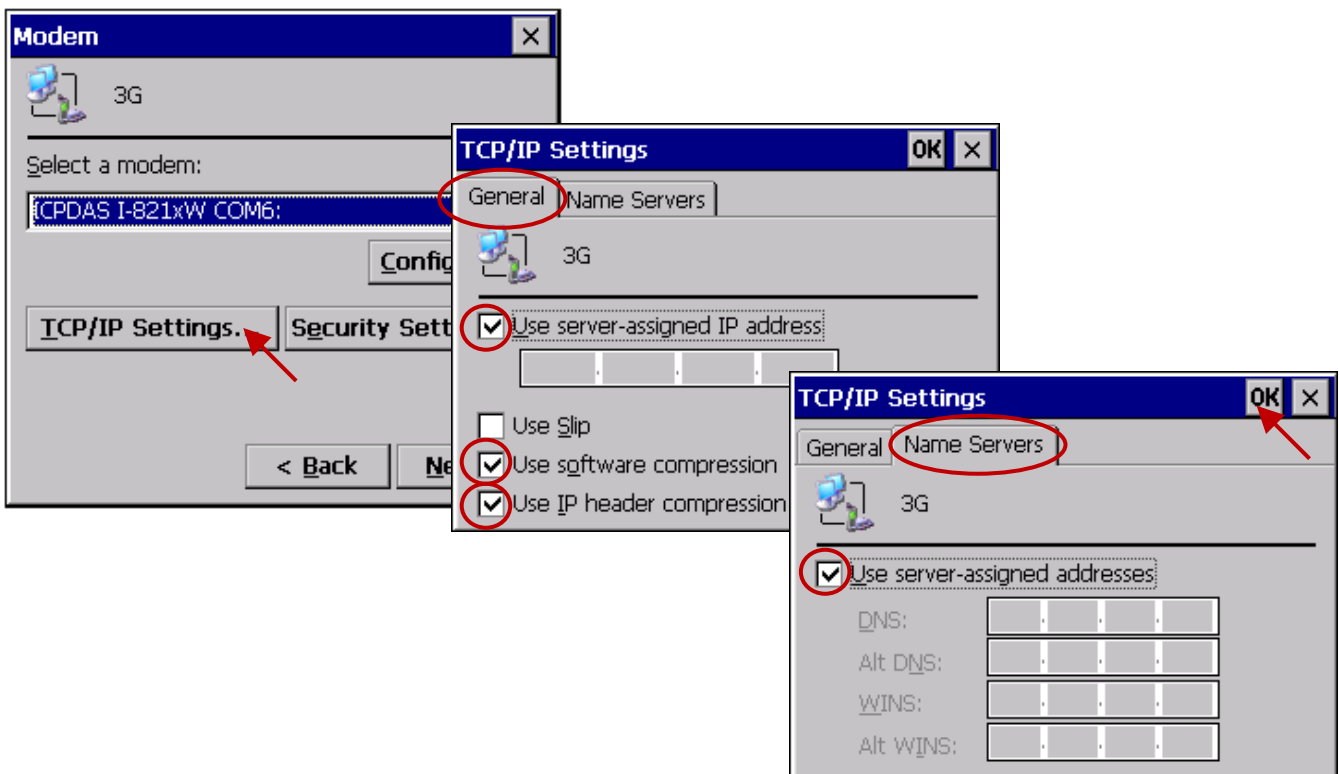
在 China 某電信公司是使用

+CGDCONT=1,"IP","CMNET"

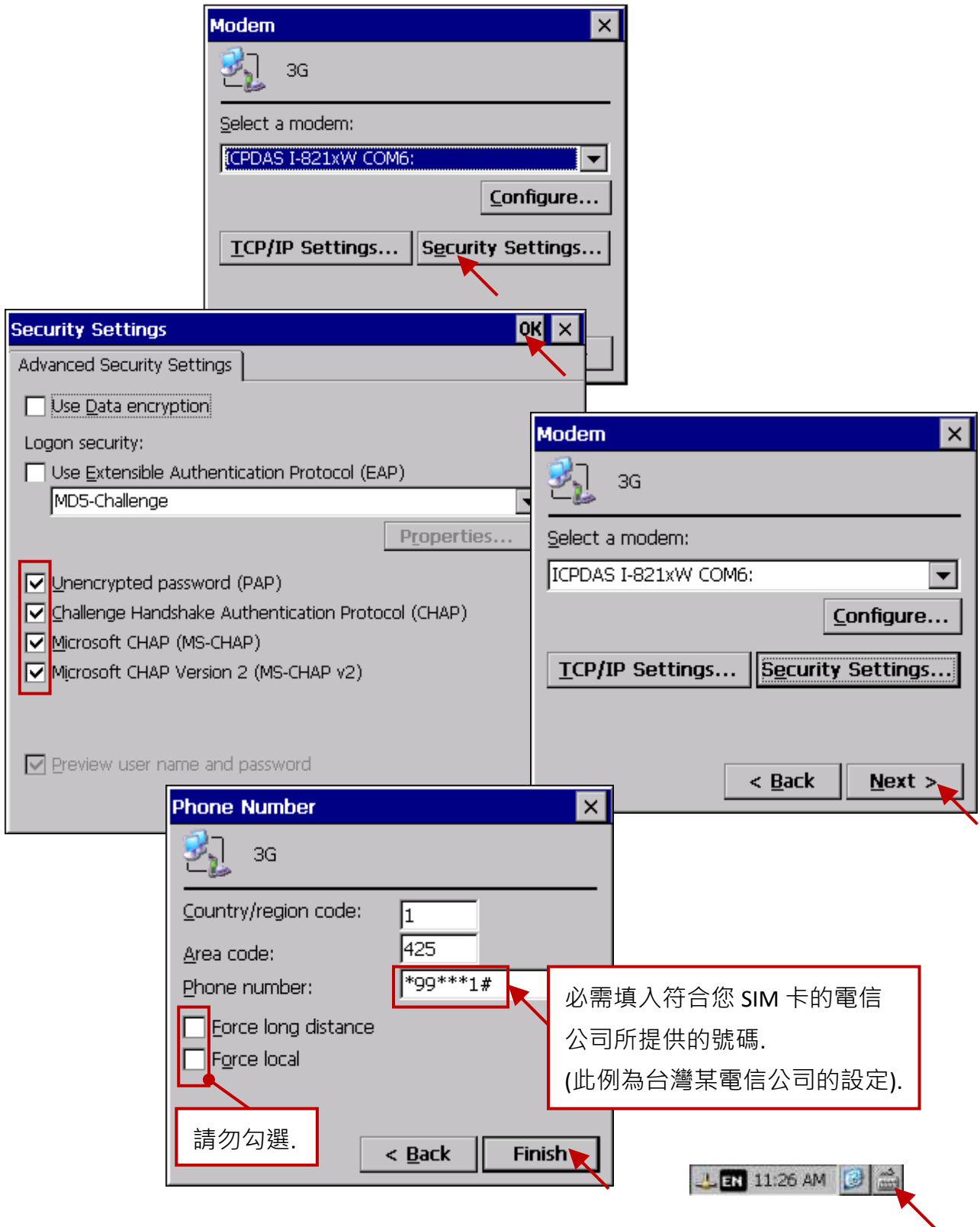
此設定內含 3G/2G 的存取點名稱 (APN)，必需配合提供 SIM 卡的電信公司的設定，也可以上網搜尋 “3G/2G APN” 查查看。



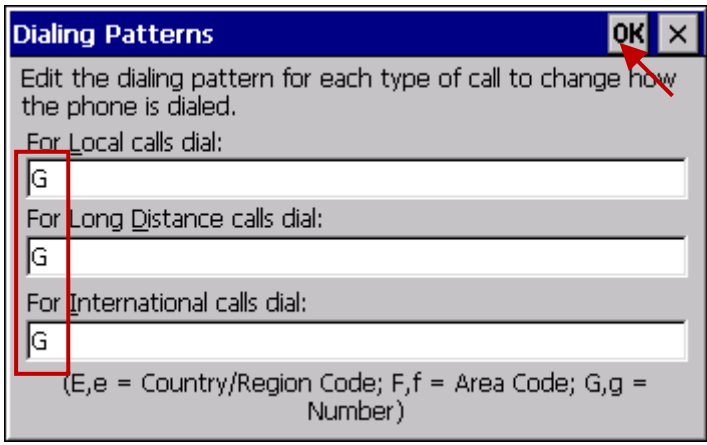
然後進入 “TCP/IP Settings ...” 進行以下相同的設定。



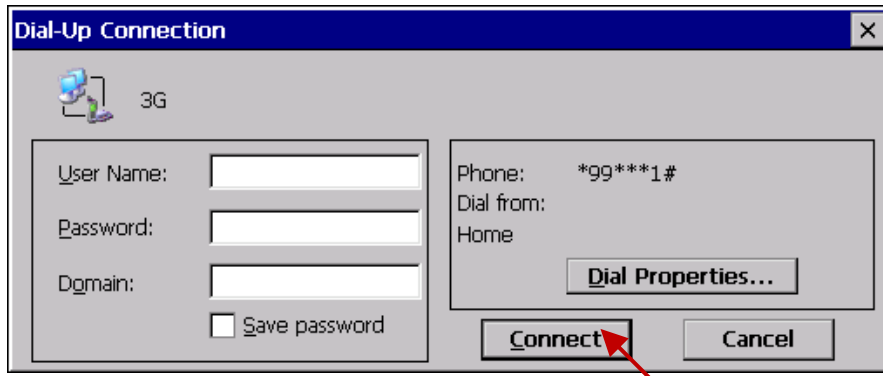
然後進入 “Security Settings ...” 進行以下相同的設定。之後，輸入撥接 3G/2G 的 “Phone Number”，此號碼需符合電信公司提供的號碼，完成後按下 “Finish”。



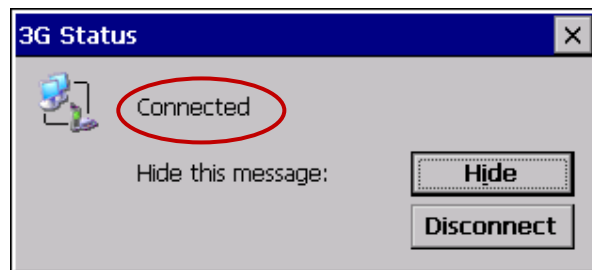
接下來用滑鼠雙擊剛才建立好的新連線，進入“Dial Properties”，再進入“Dialing Patterns”，將最下方的那 3 個欄位都改為 “G”，然後按下 “OK”。



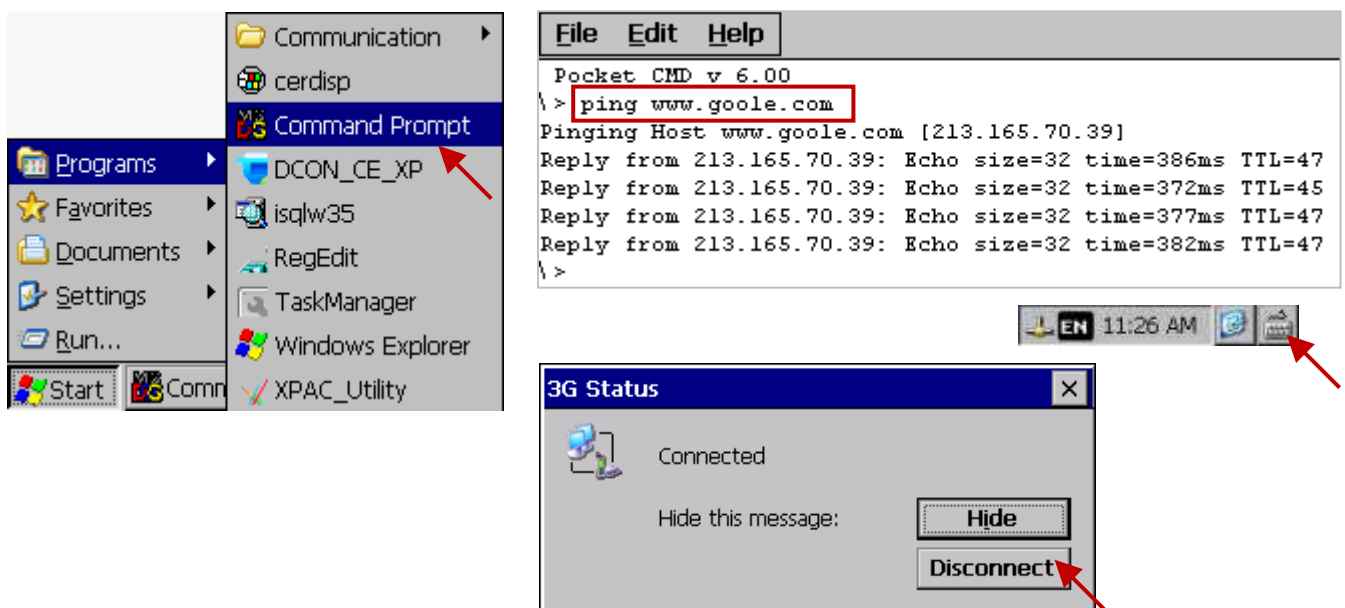
之後，需撥接一次看看能否正確連上 3G/2G 網路。下方的“User Name”與“Password”，需使用電信公司提供的設定，也可以上網搜尋“3G/2G APN”查查看。下方是使用 Taiwan 某電信公司 SIM 卡的例子 (2 個欄位都是空白)，然後按下“Connect”開始命令 I-8212W-3GWA + SIM 卡去撥號。



若連線成功會顯示“Connected” (注意: 測試成功後，要進行第 19.2.3 節的重要設定)。



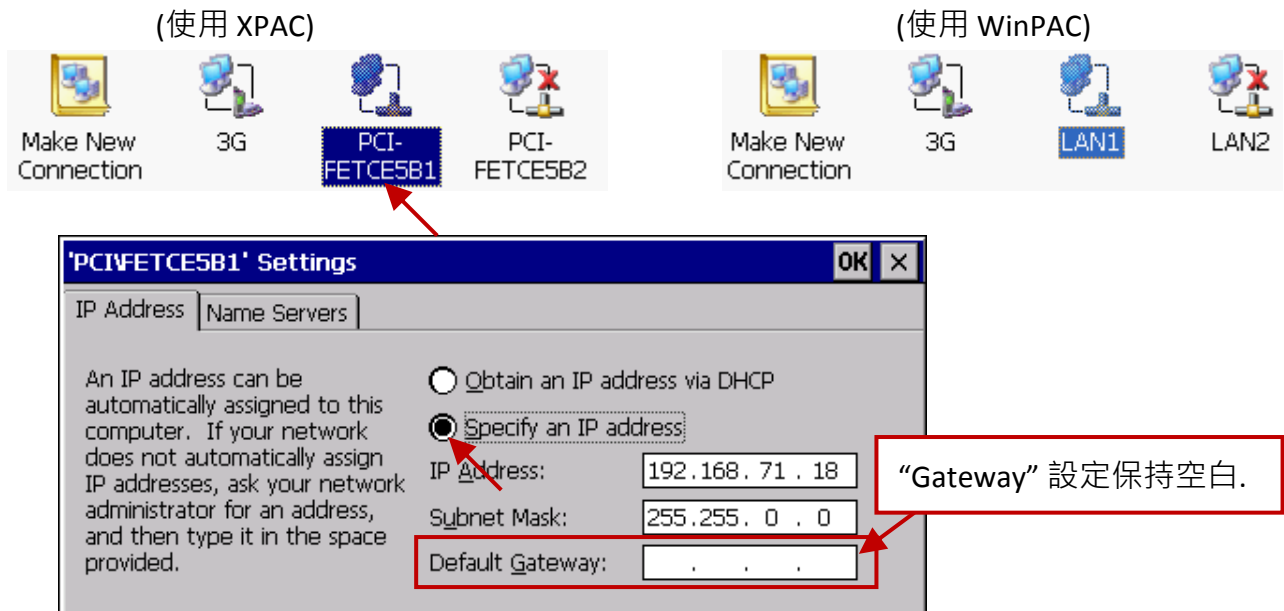
連線成功後，開啟“Command Prompt”，下個命令去 Ping 某個網站，看是否 Ping 的上。(若無法 Ping 上 Internet，請參考下一節 19.2.3 來清除 LAN1 與 LAN2 的 Gateway 設定。) 若可成功 Ping 上，先按下“Disconnect”來掛斷 3G/2G，然後進行下一頁的重要設定。



19.2.3 重要設定，不可忽略!!!

以下 2 個設定，非常重要，不可忽略!

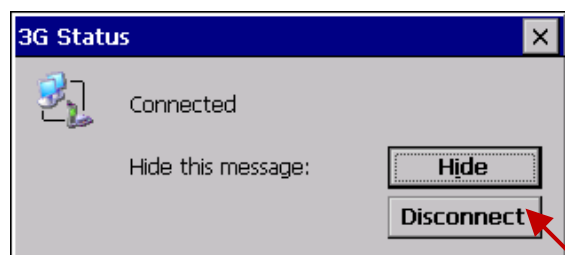
1. 若您的 PAC 是要使用 3G/2G 來連上網路 (進行 TCP 資料發送與接收，...)，則必需將 PAC 的 LAN1 與 LAN2 的 Gateway 設定清除 (保持空白)，設定完後記得最後要 Run PAC 的個別 Utility 的 “File > Save and Reboot” 將設定存起來。不然 3G/2G 雖然是 “Connected” 但卻會無法上網傳送資料。



但若您的 PAC 的 LAN1 或 LAN2 已經可以連上網路，則建議您使用 LAN1 或 LAN2 來上網收/發資料，不要使用 3G/2G (此時就必需要去設定 LAN1 或 LAN2 的 Gateway)，因為 3G/2G 雖然是無線，但速度比 LAN1/LAN2 慢，且要額外支付 3G/2G 通訊費用給電信公司。

2. 當前一節所有設定都完成後，接下來的設定一定要進行，不可忽略，不然之後使用 Win-GRAF 程序來連上 3G/2G 網路會出現問題。

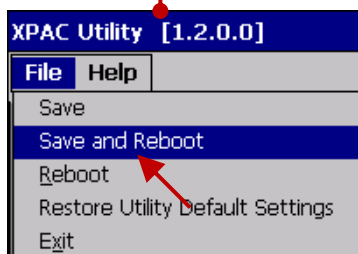
首先若您的 3G/2G 連線仍為 “Connected”，將它 Disconnect。



之後 Run 新建立的連線 (例如: “3G”) · 然後按下 “Cancel” (此時不再去按 “Connect” · 一定要按下 “Cancel”) · 最後要 Run 個別的 Utility (例如: XPAC Utility) 的 “File > Save and Reboot” 把此 “Cancel” 設定與上一節所作的全部設定都存起來 · 然後 PAC 會自動重新開機一次。



需先執行 “Cancel” 操作, 再執行
PAC Utility 來儲存此設定.



19.2.4 啟用 "Dial_up_utility_win_graf" 撥接軟體

“Dial_up_utility_win_graf” 為 ICP DAS 開發的自動撥接 3G/2G 的軟體工具程式，它可以讓 Win-GRAF 程式 (或 VB.net · C#.net 程式 與 C 程式) 可以對它下命令來連上或中斷 3G/2G，也可以讀取目前的連線狀態 與 命令狀態。請進行以下設定來啟用此撥接軟體，之後按下 “Connect”。若可以正常連上，則表示設定正確，然後再按下 “Disconnect” 看是否可正確斷線。最後必須運行 PAC Utility 把 “dial_up_utility_win_graf.exe” 加到 “Auto-Execution” 的清單內，並且執行 “File > Save and Reboot” 把設定存起來。

1. My Device

2. dial_up_utility_win_graf.exe

3. Connect

4. Address

5. Entry 3G

6. Set parameter

7. Connect when starting up

請稍待約 25 秒。

帳號, 密碼需符合 (Entry: 新連線的名稱)
電信公司的設定
(部分為空白).

如需在 PAC 開機時, 自動啟用 3G 連線, 可勾選此項目.

8. XPAC Utility

9. Auto Execution

10. Browse

11. Apply

12. Save and Reboot

Program 2: \\System_Disk\\Win-GRAF\\dial_up_utility_win

At most 10 programs can be specified to execute automatically at system startup.

19.3 控制 3G/2G 連線的函式說明

右方的 Win-GRAF 範例程式說明如何使用 `_3G_connect()` 來連線 3G/2G，只要把 `"To_connect_3G"` 設為 `"TRUE"`，它就會命令 `"Dial_up_utility_win_graf"` 去連線 3G/2G。

```
(* 將 "To_connect_3G" 設為 "TRUE" 來連線 3G 上網 *)
if To_connect_3G then
    To_connect_3G := FALSE ;
    _3G_connect() ;
end_if ;
```

右方的 Win-GRAF 範例程式說明如何使用 `_3G_disconnect()` 來停止 3G/2G 連線，只要把 `"To_disconnect_3G"` 設為 `TRUE`，它就會命令 `"Dial_up_utility_win_graf"` 去停止 3G/2G 連線。

```
(* 將 "To_disconnect_3G" 設為 "TRUE" 來停止 3G 上網*)
if To_disconnect_3G then
    To_disconnect_3G := FALSE ;
    _3G_disconnect() ;
end_if ;
```

下方的 Win-GRAF 範例程式說明如何使用 `_3G_state()` 來讀取目前 3G/2G 的連線狀態與使用 `_3G_read_cmd()` 來讀取目前 3G/2G 的命令狀態。

```
(* 取得 3G 連線狀態 *)
State_3G := _3G_state() ;
```

```
(* 讀取目前的 3G 命令狀態
0 : 無動作
1 : 連線
2 : 停止連線 *)
current_3G_cmd := _3G_read_cmd() ;
```

0 : 無動作	
(1 ~ 6) : 連線中	
1 : 開啟 COM Port	2 : 已開啟 COM Port
3 : 連接設備	4 : 已連接設備
5 : 驗證中	6 : 已驗證
7 : 密碼已過期	8 : 已連線 (Connected)
9 : 斷線 (Disconnected)	10 : 其它

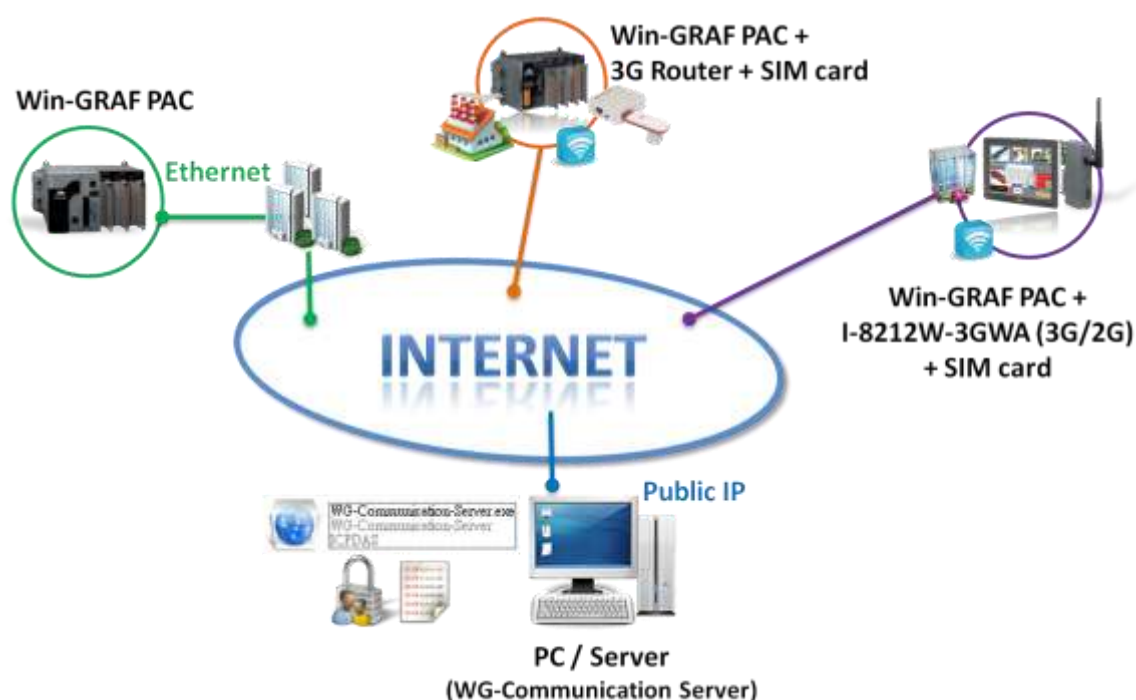
Win-GRAF PAC 在 3G/2G 撥接狀態是 `"Connected"` 下，會每隔約 15 分鐘嘗試去下達 `"ping"` 命令給 DNS Server 與 8.8.8.8 這 2 個 IP Address，若 Ping 15 秒內都收不到對方回應，會判定為 3G/2G 網路連線發生問題，此時 Win-GRAF PAC 會自動 Reset 該 3G/2G 板卡，然後再自動讓 3G/2G 板卡重新撥接去連上網路。若 User 不想去 ping 8.8.8.8 這個 IP Address，可以變更去 Ping 其它的 IP Address (比如 202.43.192.106)，方法如下：

```
(* 設定每隔約 15 分鐘要改去 Ping 的 IP Address (不同於 8.8.8.8) *)
if To_ping_a_new_ip then
    To_ping_a_new_ip := FALSE ;
    TMP_BOOL := _3G_option( 1 , '202.43.192.106' ) ;
end_if ;
```

第 20 章 透過 Ethernet 或 3G/2G 無線, 傳送 PAC 內的檔案 至遠方 PC

有些應用需在 PAC 內記錄一些應用資料, 比如溫/濕度、速度、電壓、電流...等。這些資料可被 User 設計的 Win-GRAF 程式存成 File, 然後 PAC 可透過本章的方法, 將 File 傳到 PC/Server。

在 Win-GRAF PAC 的出貨光碟中 (CD-ROM: \Napdos\Win-GRAF\demo-project), 提供了本章的 Win-GRAF 範例程式 (demo_send_file.zip), 您可參考 [第 12 章](#) 來開啟此專案。另外, 在 CD-ROM: \Napdos\Win-GRAF\Tools_Utility\, 還提供了一個在 PC (Server) 上運行的通訊 Server 軟體 (稱為 WG-Communication-Server), 可讓遠端的 PAC 傳送 File 至 Server。



注意: 有兩種方式可設置具有 3G/2G 網路的 Win-GRAF PAC:

- (1) 使用 3G 路由器 (Router) 加上一片 SIM 卡。
- (2) 使用 I-8212W-3GWA 模組加上一片 SIM 卡 (參考 [第 19 章](#))。

此外, 若 PAC 是透過 3G/2G 無線 或 WAN 傳送檔案, 則安裝 "WG-Communication Server" 的 PC/Server 必須要配置實體 IP。

從以下 Win-GRAF Driver 版本起, 才有支持本章描述的功能:

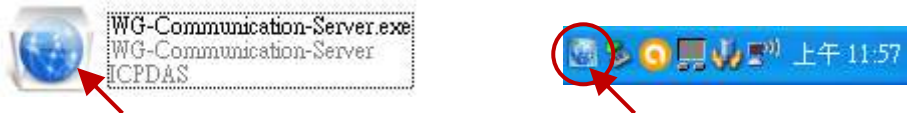
WP-8xx8 :	1.05
VP-x2x8-CE7 :	1.01
XP-8xx8-CE6 :	1.03
WP-5xx8-CE7 :	1.02

20.1 "WG-Communication-Server" 軟體說明

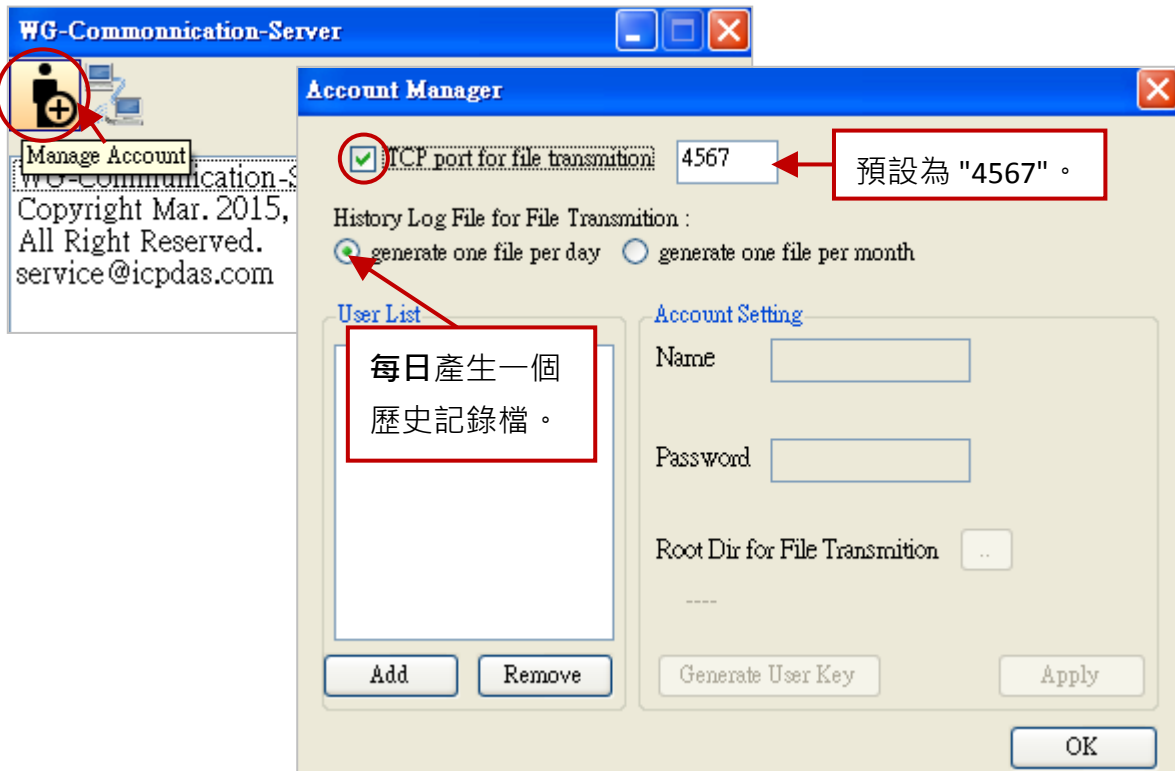
注意: 請將 "WG-Communication-Server" 資料夾 (內含 WG-Communication-Server.exe 與 DLL) 複製到 D:\ 內，且 PC/Server 必需運行此軟體，不然遠方的 PAC 無法將 File 傳到此 PC/Server。User 可使用 "WG-Communication-Server" 建立多組帳號/密碼 (最多 100 組) 來供遠端 PAC 登入到 PC/Server，並進行檔案傳送。

新增帳號/密碼:

1. 滑鼠雙擊 "WG-Communication-Server.exe" 後，程式會直接縮小至右下角的系統列中 (即，在背景運行)，如需編輯設定，請雙擊該圖示來開啟軟體。



2. 點選 "Manage Account" 圖示開啟設定畫面，並勾選 "TCP port for file transmission"。



"TCP port for file transmission":

進行檔案傳輸時，需啟用 PAC 與 PC 通訊的 TCP 埠號。(預設: 4567，範圍: 1000 ~ 9999)

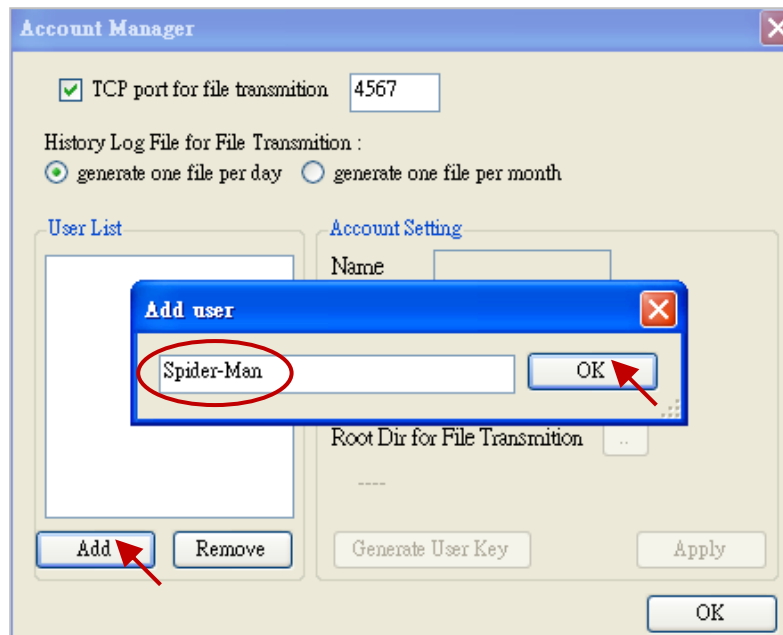
註: 勾選/取消此項目 或 修改 TCP 埠號，都必須重新開啟此軟體才會套用設定。

"History Log File": 選擇 每日 或 每月 產生一個歷史記錄檔，此例為 "每日"。

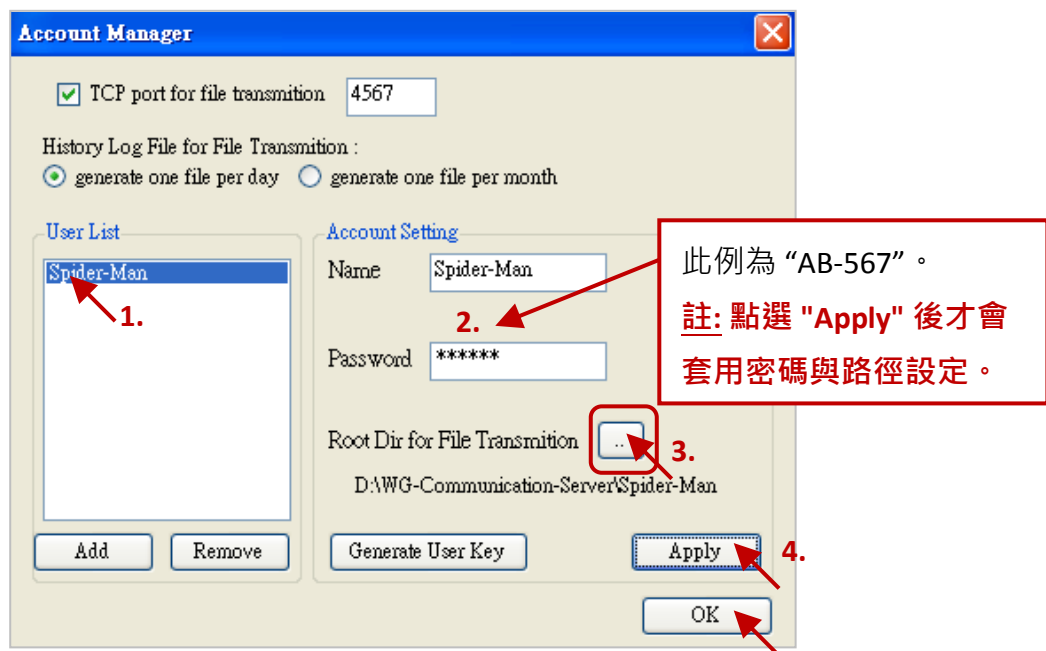
註: 請至下一步驟設定帳號/密碼，另外，您可開啟 "D:\WG-Communication-Server\account.txt" 查詢先前設定過的帳號/密碼。



3. 接著，點選 "Add" 按鈕並輸入使用者名稱 (例如: "Spider-Man")，再按 "OK"。

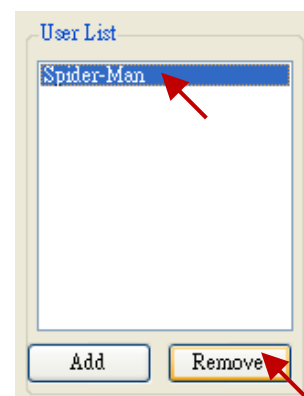


4. 滑鼠點一下使用者名稱 (例如: "Spider-Man") 來設定密碼 (例如: "AB-567")，接著設定 PAC File 的存放路徑 (建議放在預設的 D:\WG-Communication-Server\使用者名稱 或 點選 "Root Dir..." 按鈕)，再按 "Apply" 套用設定。**註:** 您可在 [22.1 節](#) 了解 "Generate User Key" 的使用方式。



刪除帳號/密碼

點選欲刪除的使用者名稱 (例如: "Spider-Man") 再點選 "Remove" 按鈕，最後按 "OK" 刪除此帳號/密碼。



20.2 "Send_File_To"、"Send_File_State" 與 "Send_File_Abort" 函式

以下 3 個函式用來處理從 PAC 到 PC/Server 的檔案傳送。

"Send_File_To" 函式:

用來傳送 PAC 內的單一檔案到遠端的 PC/Server (需 Run "WG-Communication-Server" 軟體)。



輸入參數:

註: 以下字串 (String) 的有效字元為:

A ~ Z、a ~ z (區分大小寫)、0 ~ 9、'!' (句點)、'@' (At)、'-' (減號) 與 '_' (底線)。

IP: (資料型態: String) 遠端 PC/Server 的 IP 位址。

Port: (資料型態: DINT) Server 的 TCP 通訊埠號 (範圍: 1000 ~ 9999)。

User: (資料型態: String) 使用者名稱 (最多 32 字元)。

Password: (資料型態: String) 密碼 (最多 32 字元)。

PAC_ID: (資料型態: DINT) 用來識別檔案是由哪一台 PAC 傳出的編號。

數值範圍: -2,147,483,648 ~ 0 ~ 2,147,483,647。

Des_File_Path: (資料型態: String) 需存放在 PC 的檔案目的地路徑 (最多 128 字元)。

第一個字元需是 '\', 而最後一個字元不可以是 '\'。

(例如: '\2014\12\data001.txt' 或 '\Record\recp-2014-11-08.txt'。)

Src_File_Path: (資料型態: String) 由 PAC 傳送的檔案來源路徑 (最多 128 字元)。

第一個字元需是 '\', 而最後一個字元可以不是 '\'。

(例如: '\Micro_SD\PAC\data001.txt' 或 '\System_Disk\DATA\recp-2014-11-08.txt'。)

輸出參數:

Q: (資料型態: BOOL)

TRUE: 表示通訊 OK。

FALSE: 表示輸入參數錯誤 或 "Src_File_path" 路徑不存在 或 檔案大小為 0。

"Send_File_State" 函式:

用來取得 PAC 檔案的傳送狀態 (需搭配 "Send_File_To" 函式)。



註: 您可以滑鼠點一下該函式，再按 "F1" 鍵來查看詳細的設定資訊。

輸入參數: 無。

輸出參數:

Q: (資料型態: DINT)

- 0: Sleep，尚未調用 Send_File_To() 函式。
- 1 ~ 99: Busy，正在傳送檔案中，數字為進度百分比。
- 100: Succeed，檔案已經成功傳到遠方的 PC 了。

- 1: 傳送失敗 或 發生逾時 (Timeout)。
- 2: 被 "Send_File_Abort" 命令強制停止檔案傳送了。
- 3: 使用者名稱/密碼錯誤。
- 4: PC 內 無法建立正確的子路徑 或 檔案，或
檔案大於 10,000,000 Bytes 或
PC 內 未運行 "WG-Communication-Server" 軟體。
- 5: 要傳送的檔案不存在 或 檔案大小為 0。
- 6: PAC 內 無法開啟 "\Email_ETH" 路徑下的檔案。

"Send_File_Abort" 函式:

用來取消 PAC 檔案傳送。



輸入參數: 無。

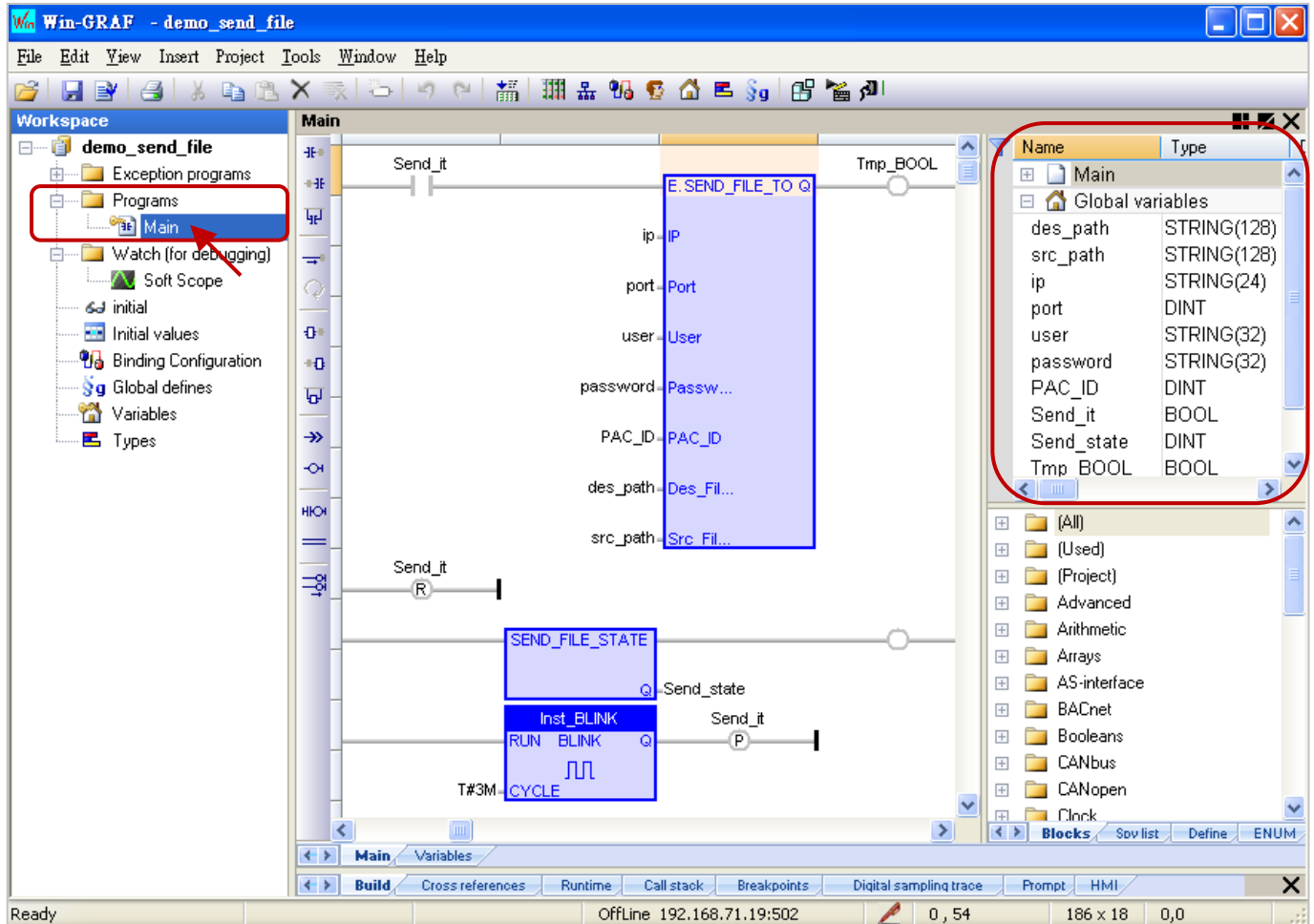
輸出參數:

Q: (資料型態: BOOL) · 永遠回傳 "TRUE"。

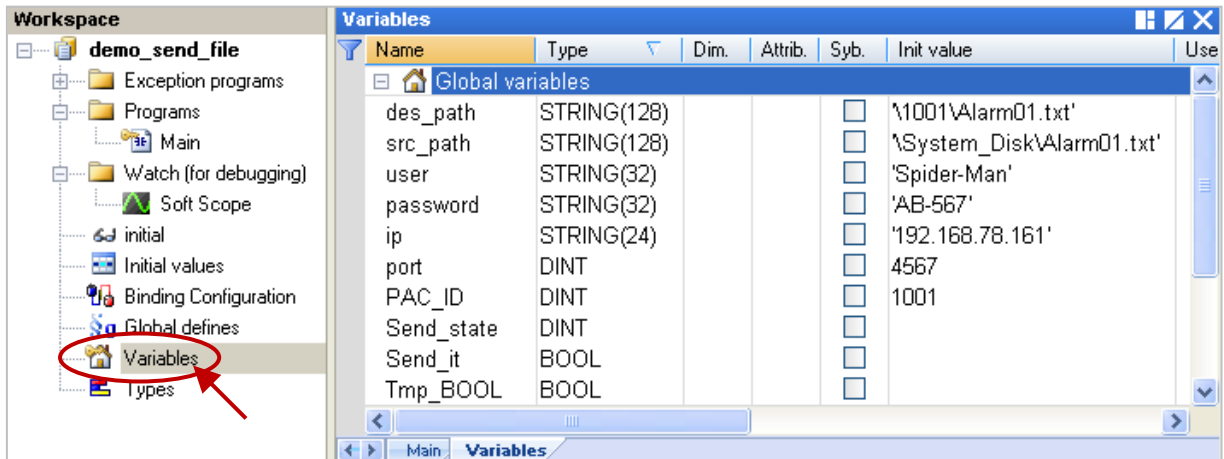
20.3 Win-GRAF 範例程式 (demo_send_file.zip)

在 Win-GRAF PAC 的出貨光碟中 (CD-ROM: \Napdos\Win-GRAF\demo-project) , 提供了此章節說明的 Win-GRAF 專案 (demo_send_file.zip) 。使用前 , 可參考 [第 12 章](#) 來開啟此專案 。

開啟 "demo_send_file" 專案後 , 滑鼠雙擊 "Main" 可開啟此 LD 程式 , 而右邊的變數區列出了所有已定義的變數 。



首先 , 我們先了解此專案使用了哪些變數與其用途。您也可點選 "Variables" 開啟變數視窗來查看/設定變數 。



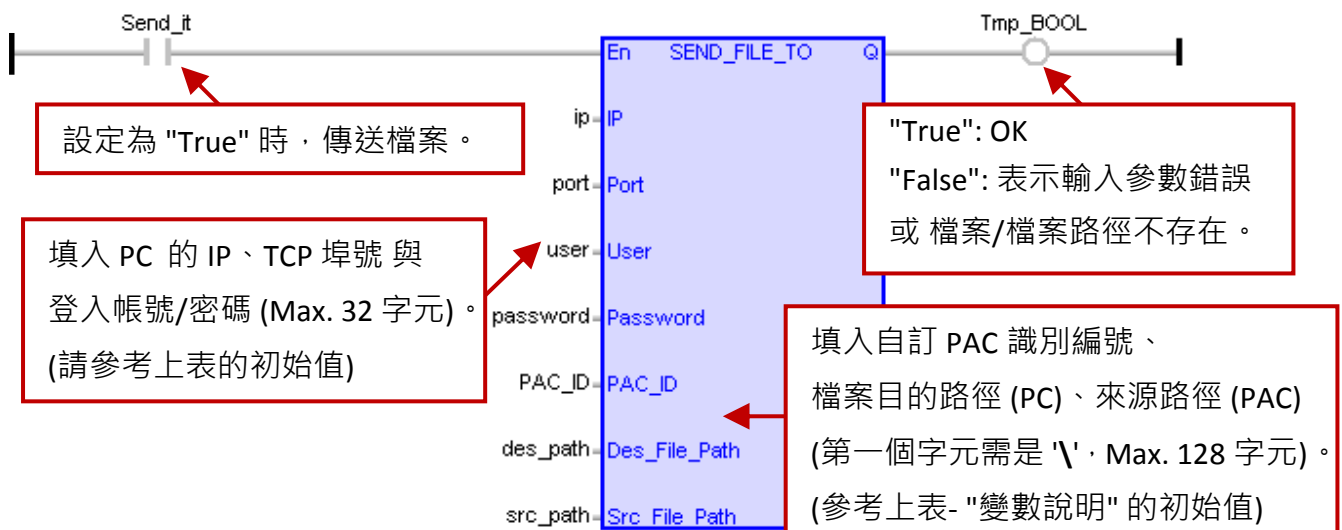
變數說明:

下表列出了在此 Win-GRAF 專案中 (demo_send_file.zip) 所使用的變數名稱、資料型態、用途與 初始值。

變數名稱	資料型態	說明
des_path	STRING(128)	檔案傳送到 PC 的目的地位置 (Max. 128 字元)。 (初始值設為: '\\1001\Alarm01.txt')
src_path	STRING(128)	PAC 傳送檔案的來源位置 (Max. 128 字元)。 (初始值設為: '\\System_Disk\Alarm01.txt')
user	STRING(32)	登入 PC 的使用者名稱 (Max. 32 字元)。 (初始值設為: 'Spider-Man')
password	STRING(32)	登入 PC 的密碼 (Max. 32 字元)。 (初始值設為: 'AB-567')
ip	STRING(24)	PC 的 IP 位址 (Max. 24 字元)。 (初始值設為: '192.168.78.161')
port	DINT	WG-Communication-Server 用來接收檔案的 TCP 埠號。 (初始值設為: 1000, 範圍: 1000 ~ 9999。)
PAC_ID	DINT	PAC 的識別編號。(初始值設為: 1001)
Send_state	DINT	檔案傳送狀態。(參考 20.2 節)
Send_it	BOOL	設定為 "True" 時, 傳送檔案。
Tmp_BOOL	BOOL	TRUE: 表示 OK。 FALSE: 表示輸入參數錯誤 或 來源路徑 (Src_File_path) 不存在 或 檔案大小為 0。

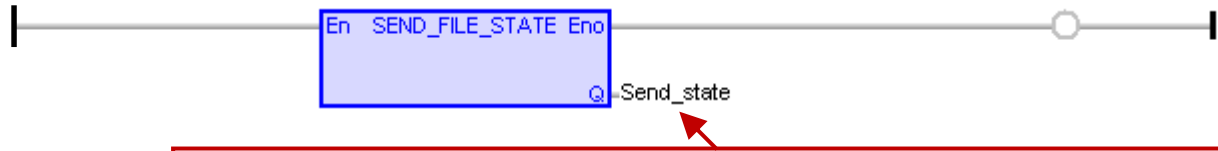
LD 程式 (Main)

此 "Send_File_To" 函式 (參考 20.2 節) 用來傳送 PAC 內的單一檔案到遠端的 PC/Server。



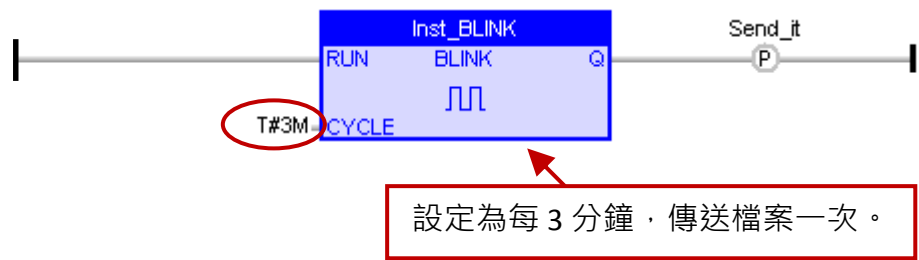


此 "Send_File_State" 函式用來取得 PAC 檔案的傳送狀態。



- 回傳檔案傳送狀態。
- 0: Sleep · 尚未調用 Send_File_To() 函式。
 - 1 ~ 99: Busy · 正在傳送檔案中，數字為進度百分比。
 - 100: Succeed · 檔案已經成功傳到遠方的 PC 了。
 - 1: 傳送失敗 或 發生逾時 (Timeout)。
 - 2: 被 "Send_File_Abort" 命令強制停止檔案傳送了。(見 [20.2 節](#))
 - 3: 使用者名稱/密碼錯誤。
 - 4: PC 內 無法建立正確的子路徑 或 檔案，或
檔案大於 10,000,000 Bytes 或
PC 內 未運行 "WG-Communication-Server" 軟體。
 - 5: 要傳送的檔案不存在 或 檔案大小為 0。
 - 6: PAC 內 無法開啟 "\Email_ETH" 路徑下的檔案。

此範例中的 "BLINK" 功能方塊用來每 3 分鐘，傳送 PAC 檔案一次。



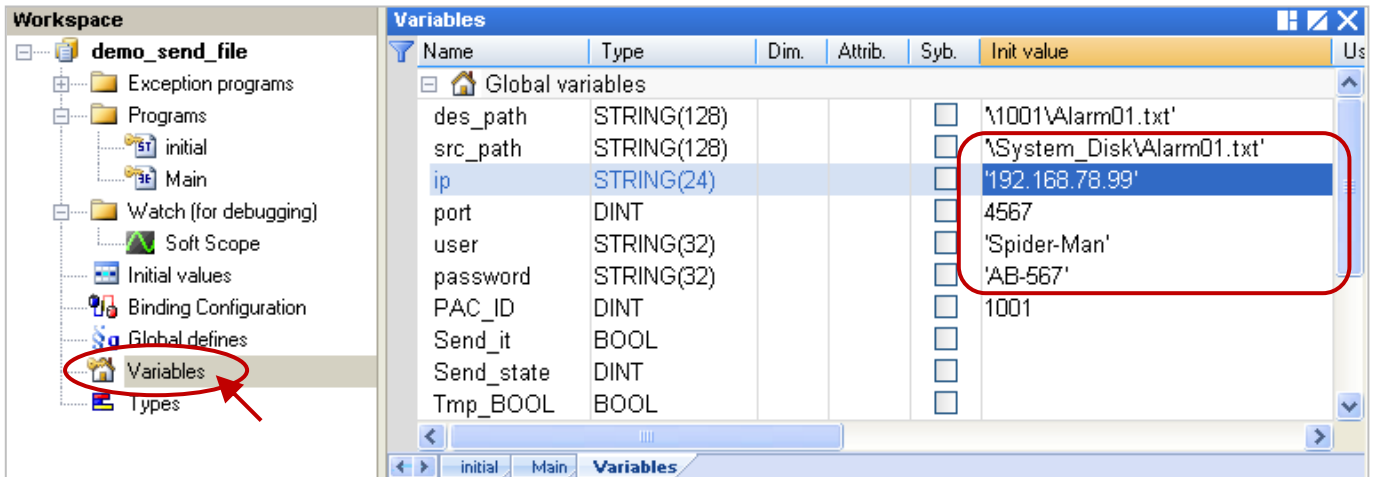
20.4 檔案傳送測試 (Ethernet)

測試之前，請先在 "Variables" 視窗確認以下資訊：

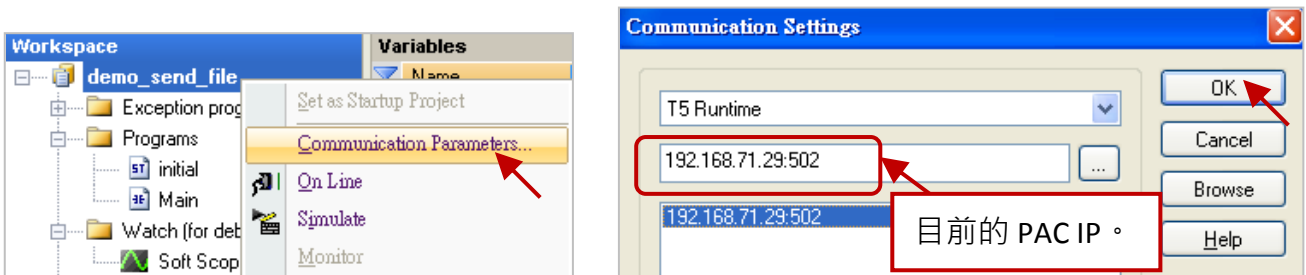
"src_path": PAC 的 "\\System_Disk\Alarm01.txt" 是否存在? 若沒有，請使用 FTP 方式放入此檔案，或改指定其它檔案。 (**注意**: 檔案不可以是 0 Byte)

"ip": 請設定為您目前 PC 的 IP (若透過 3G/2G 或 WAN 來連到 Server，則需填寫實體 IP)。

"port" (4567)、"user" 與 "password" 必需和 WG-Communication-Server 的設定值一樣 (見 [20.1 節](#))。



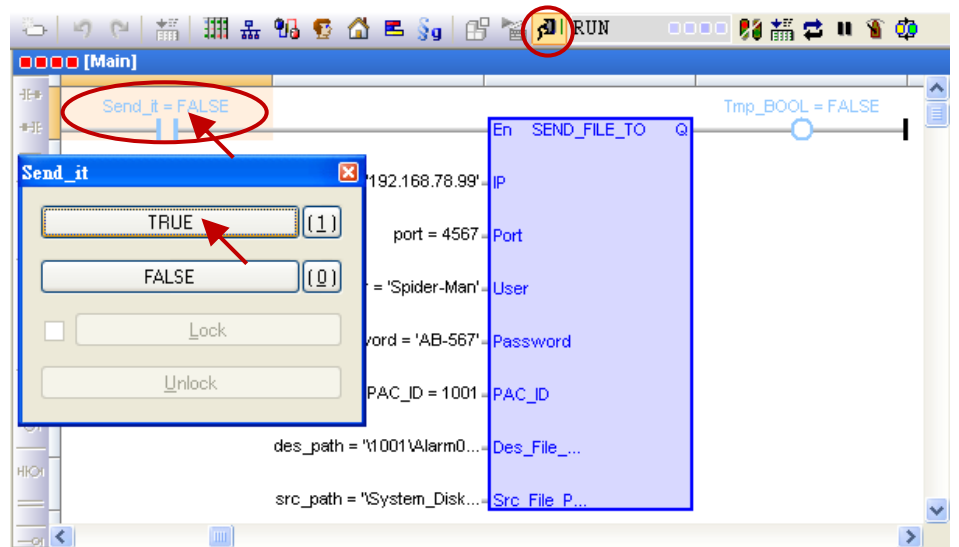
另外，通訊參數請修改為您目前 PAC 的 IP，再重新編譯並下載程式到 Win-GRAF PAC 中。



註: 如需設定通訊的 Timeout 時間 (預設: 3 秒)，請參考 [2.3.5 節](#)。

(例如: 設定 IP 為 "192.168.71.29:502(10)"，表示 Timeout 為 10 秒。)

下載後，滑鼠雙擊 LD 程式 (Main) 中的 "Send_it"，將其設定為 "TRUE" 以開始傳送檔案。



若 "Send_state" =100，表示傳送成功。

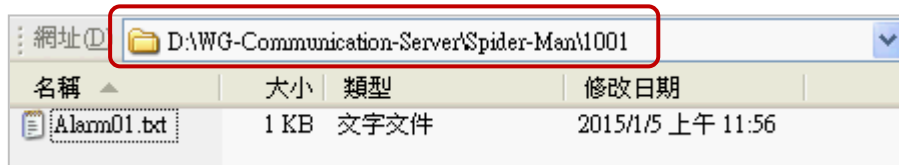


回傳檔案傳送狀態。

- 0: Sleep，尚未調用 Send_File_To() 函式。
- 1 ~ 99: Busy，正在傳送檔案中，數字為進度百分比。
- 100: Succeed，檔案已經成功傳到遠方的 PC 了。**
- 1: 傳送失敗 或 發生逾時 (Timeout)。
- 2: 被 "Send_File_Abort" 命令強制停止檔案傳送了。(見 [20.2 節](#))
- 3: 使用者名稱/密碼錯誤。
- 4: PC 內 無法建立正確的子路徑 或 檔案；檔案大於 10,000,000 Bytes；PC 內 未運行 "WG-Communication-Server" 軟體。
- 5: 要傳送的檔案不存在 或 檔案大小為 0。
- 6: PAC 內 無法開啟 "\Email_ETH" 路徑下的檔案。

檔案已傳送到 PC 的 "D:\WG-Communication-Server\Spider-Man\1001\Alarm01.txt"。

(可見 [20.1 節](#) - 步驟 4 與 [20.3 節](#) "des_path" 變數說明)



另外，在此 Win-GRAF 範例中 (demo_send_file.zip)，設計每 3 分鐘會傳送檔案一次，您可在 "WG-Communication-Server" 畫面查看 PAC 的通訊記錄。



此處會顯示通訊記錄。

第 21 章 Win-GRAF 簡訊功能

本章節說明如何使用 PAC 搭配 3G/2G 無線通訊模組來發送/接收簡訊。Win-GRAF PAC 的出貨光碟中 (CD-ROM: \Napdos\Win-GRAF\demo-project) 提供了此章節的 Win-GRAF 專案 (Demo_SMS.zip，請參考 21.2 節說明)。

軟/硬體需求:

1. Win-GRAF PAC

以下 Win-GRAF Driver 版本才有支援此章節功能 (可參考 [第 19 章](#))

WP-8xx8 : 1.05 ; VP-x2x8-CE7 : 1.01 ; XP-8xx8-CE6 : 1.03 ; WP-5xx8-CE7 : 1.02

2. 3G/2G 無線通訊模組

請先參考 [第 19 章](#) 查看軟硬體的設置方式，或詳見各產品網頁:

http://m2m.icpdas.com/2G_3G_Modems.html 支援的 GSM Modem 有

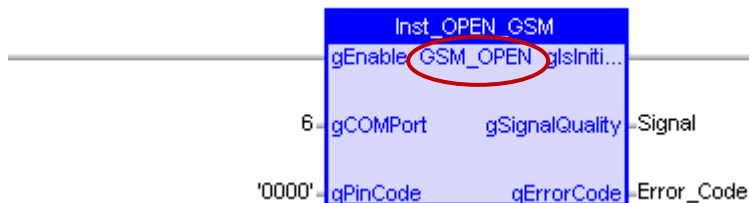
GTM-201-RS232, GTM-203M-3GWA, I-8212W, I-8212W-3GWA (參考 [第 19 章](#)), I-8213W-3GWA。

註: 以上 GSM Modem 因為認證問題，只能在某些地區銷售，請聯絡經銷商了解詳細情況。

21.1 "GSM_Open"、"Send_SMS" 與 "Read_SMS" 函式

以下 3 個函式可用來處理 Win-GRAF PAC 的簡訊傳送/接收功能。

"GSM_OPEN" 函式: 用來開啟/關閉 GSM 模組。



註: 您可以滑鼠點一下該函式，再按 "F1" 鍵來查看詳細的設定資訊。

輸入參數:

gEnable: 資料型態: BOOL

TRUE: 開啟指定的 PAC COM Port 去連接 GSM 模組並初始化。

FALSE: 取消連接 GSM 模組並關閉指定的 PAC COM Port。

gCOMPort: 資料型態: DINT

GSM 模組使用的 PAC COM Port 編號。

gPinCode: 資料型態: STRING

如果需要，可填入用來解鎖 SIM 卡的 PIN Code。

輸出參數:

gIsInitialized: 資料型態: BOOL

TRUE: 已成功開啟指定的 PAC COM Port 並初始化 GSM 模組。

FALSE: 開啟指定的 PAC COM Port 失敗 或 未初始化 GSM 模組。

gSignalQuality: 資料型態: SINT

0 ~ 31: 數值越大表示訊號越佳。

99: 訊號不明或未偵測到。

gErrorCode: 資料型態: INT

0: 無錯誤。

-1: 未連接 GSM 模組。

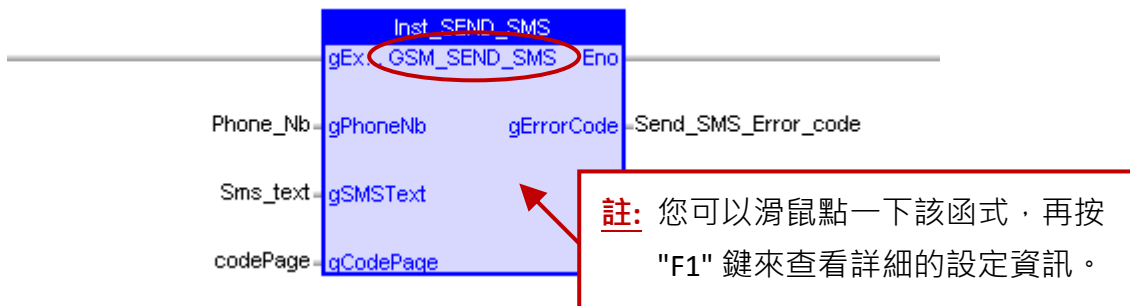
-2: 未插入 SIM 卡。

-3: SIM 卡的 PIN Code 錯誤。

-4: SIM 卡設定錯誤。

-5: 無法開啟指定的 PAC COM Port。

"GSM_SEND_SMS" 函式: 用來透過 GSM 模組傳送簡訊 (SMS)。



注意: 使用 "GSM_SEND_SMS" 功能之前, 使用者必須先使用 "GSM_OPEN" 功能來開啟 GSM 模組所連接的 PAC COM Port, 否則此功能將無效。

輸入參數:

gExecute: 資料型態: BOOL

Pulse TRUE: 觸發以傳送簡訊。

gPhoneNb: 資料型態: STRING

目的地址 (電話號碼)。

gSMSText: 資料型態: STRING

簡訊內容。

gCodePage: 資料型態: UDINT

文字的編碼代號。請參考下例子:

英文: 0 繁體中文: 950 簡體中文: 936 日文: 932 俄文: 866

注意: 若 "gCodePage" 設為 0, 則最大簡訊長度為 160 字元。

若 "gCodePage" 設為非 0, 則最簡訊長度為 70 字元。

若輸入超過最大文字長度, 則 Win-GRAF Driver 會傳送從開頭到最大長度的簡訊。

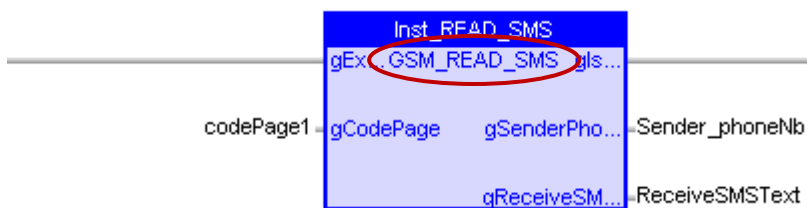
輸出參數:

gErrorCode: 資料型態: INT

- | | |
|-------------|-------------------------|
| 4 : 傳送簡訊成功。 | -1 : 找不到 GSM 模組。 |
| 3 : 傳送簡訊中。 | -2 : 未插入 SIM 卡。 |
| 2 : 等待傳送簡訊。 | -4 : SIM 卡設定錯誤。 |
| 1 : 預備傳送簡訊。 | -5 : 無法開啟 PAC COM Port。 |
| 0 : 無操作。 | -6 : 無接收的號碼。 |
| | -7 : 傳送簡訊失敗。 |

"GSM_READ_SMS" 函式:

用來透過 GSM 模組讀取簡訊 (SMS)。



註: 您可以滑鼠點一下該函式，再按 "F1" 鍵來查看詳細的設定資訊。

注意: 使用 "GSM_READ_SMS" 功能之前，使用者必須先使用 "GSM_OPEN" 功能來開啟 GSM 模組所連接的 PAC COM Port，否則此功能將無效。

輸入參數:

gExecute: 資料型態: BOOL

TRUE : 啟用讀取 GSM 模組內的簡訊。

FALSE : 禁用讀取 GSM 模組內的簡訊。

gCodePage: 資料型態: UDINT

文字的編碼代號。請參考下例子:

英文: 0 繁體中文: 950 簡體中文: 936 日文: 932 俄文: 866

輸出參數:

gIsNewSMS: 資料型態: BOOL

Pulse TRUE: 有新的簡訊傳來。

gSenderPhoneNb: 資料型態: STRING

來源位址 (電話號碼)。

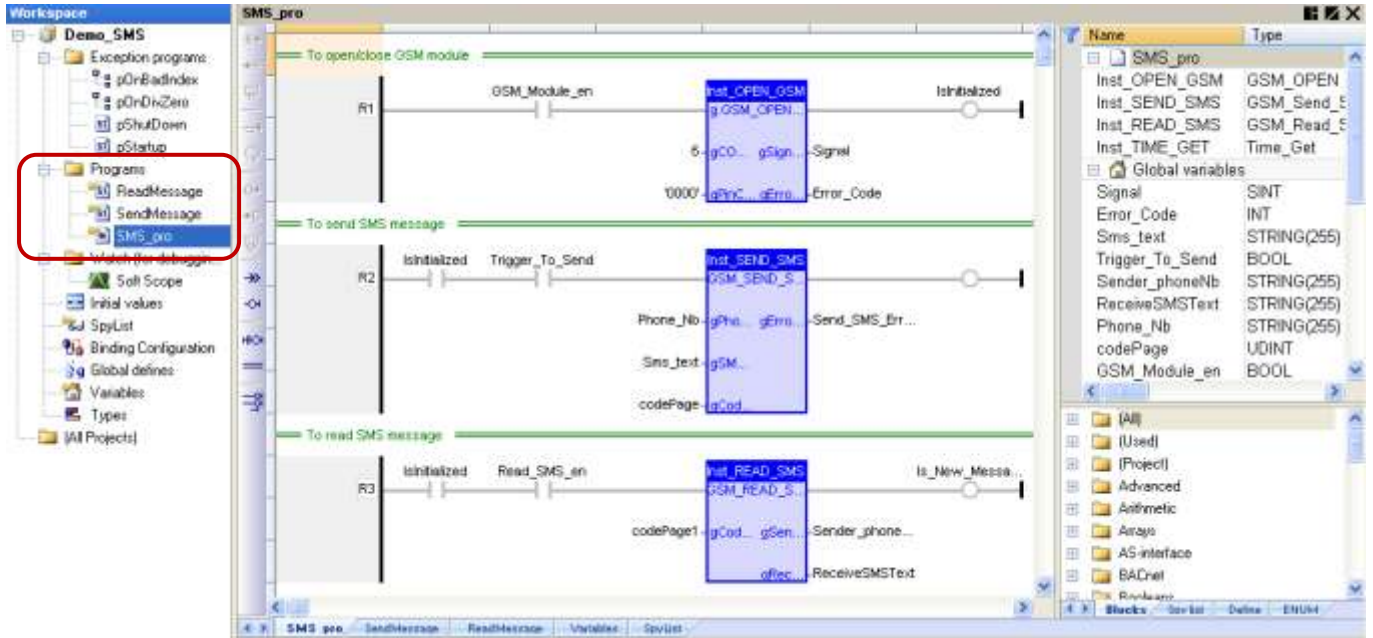
gReceiveSMSText: 資料型態: STRING

簡訊內容。

21.2 Win-GRAF 範例程式 (Demo_SMS.zip)

在 Win-GRAF PAC 的出貨光碟中 (CD-ROM: \Napdos\Win-GRAF\demo-project) ，提供了此章節說明的 Win-GRAF 專案 (Demo_SMS.zip) 。使用前，可參考 [第 12 章](#) 來開啟此專案。

"Demo_SMS" 專案包含了 1 個 LD 程式 (SMS_pro) 與 2 個 ST 程式 (SendMessage 與 ReadMessage) 。



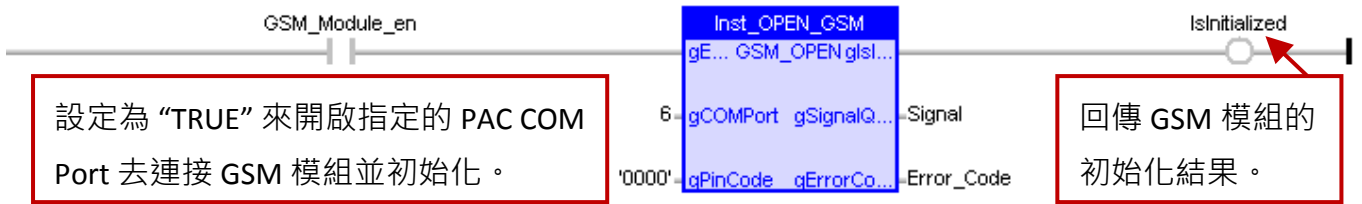
變數說明:

變數名稱	資料型態	說明
GSM_Module_en	BOOL	設定為 "TRUE" 來開啟指定的 PAC COM Port 去連接 GSM 模組並初始化。
Signal	SINT	GSM 模組的訊號品質。(參考 21.1 節)
Error_Code	INT	GSM 模組的錯誤碼。(參考 21.1 節)
IsInitialized	BOOL	用來判斷 GSM 模組是否已初始化。
Trigger_To_Send	BOOL	設定為 "TRUE" 以傳送簡訊。
Phone_Nb	STRING(255)	傳送簡訊至該電話號碼。(初始值: '0932860424')
Sms_text	STRING(255)	簡訊內容。(初始值: 'This message is sent from Win-GRAF PAC')
codePage	UDINT	傳送簡訊的編碼代號。(參考 21.1 節)
Send_SMS_Error_code	INT	傳送簡訊的錯誤碼。(參考 21.1 節)
Read_SMS_en	BOOL	設定為 "TRUE" 以讀取簡訊。
codePage1	UDINT	接收簡訊的編碼代號。(初始值: UDINT#950)
Sender_phoneNb	STRING(255)	傳來簡訊的電話號碼。
ReceiveSMSText	STRING(255)	接收的簡訊內容。
Is_New_Message	BOOL	回傳是否有新簡訊。
Got_New_Message	STRING(255)	用於 ST – ReadMessage，接收的簡訊內容。
Got_Message_from_who	STRING(255)	用於 ST – ReadMessage，傳來簡訊的電話號碼。

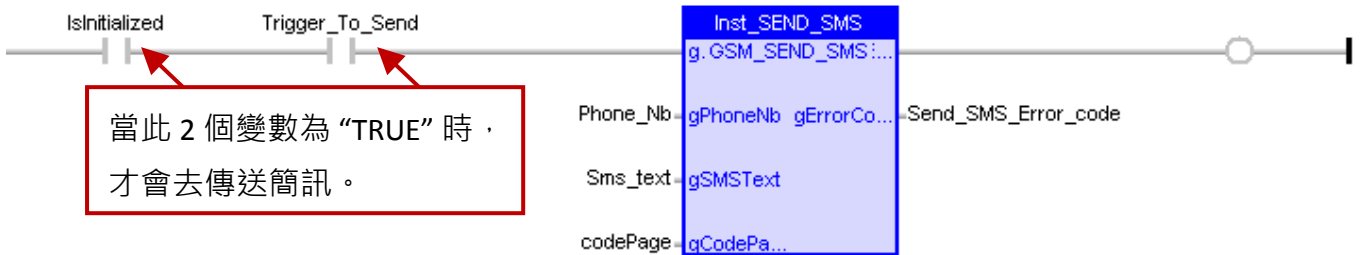
LD 程式 (SMS_pro)

請參考 [21.1 節](#) 了解以下函式的設定說明。

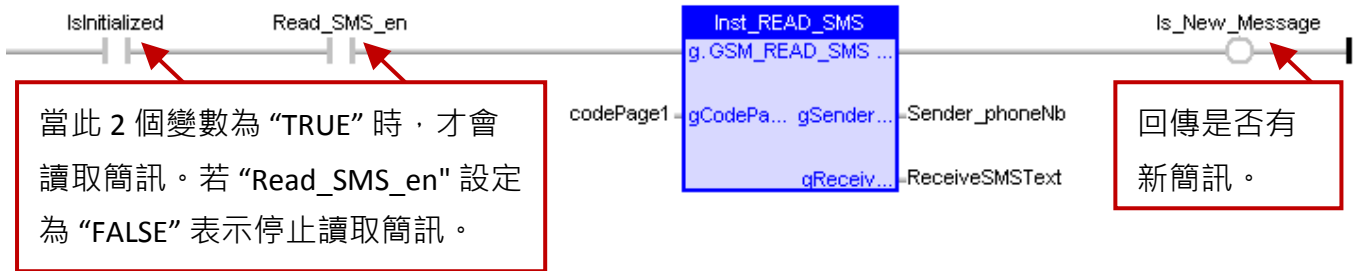
To open/close GSM module



To send SMS message



To read SMS message



ST 程式 (SendMessage)

```
if Trigger_To_Send then
```

```
  if IsInitialized then
```

```
    if Send_SMS_Error_code < 0 then
```

```
      (* 傳送簡訊失敗 *)
```

```
      Trigger_To_Send := false;
```

```
      (* 此處加入失敗時的處理程序 *)
```

```
    elseif Send_SMS_Error_code = 4 then
```

```
      (* 傳送簡訊成功 *)
```

```
      Trigger_To_Send := false;
```

```
      (* 此處加入成功時的處理程序 *)
```

```
    end_if;
```

```
  else
```

```
    (* GSM 模組未初始化 *)
```

```
    (* 此處加入失敗時的處理程序 *)
```

```
  end_if;
```

```
end_if;
```

ST 程式 (ReadMessage)

(* 取得新簡訊 *)

if Is_New_Message then

(* 若使用者需要較多的時間來處理新訊息, *)

(* 可以設定 "Read_SMS_en" 為 "FALSE" 來取消 Read_SMS 功能 *)

(* 並於處理完成後, 再設定 "Read_SMS_en" 為 "TRUE" *)

(* 注意: 若使用者設定 "Read_SMS_en" 為 "FALSE", 將不會讀取新簡訊。 *)

(* Read_SMS_en := false; *)

Got_New_Message := ReceiveSMSText;

Got_Message_from_who := Sender_phoneNb;

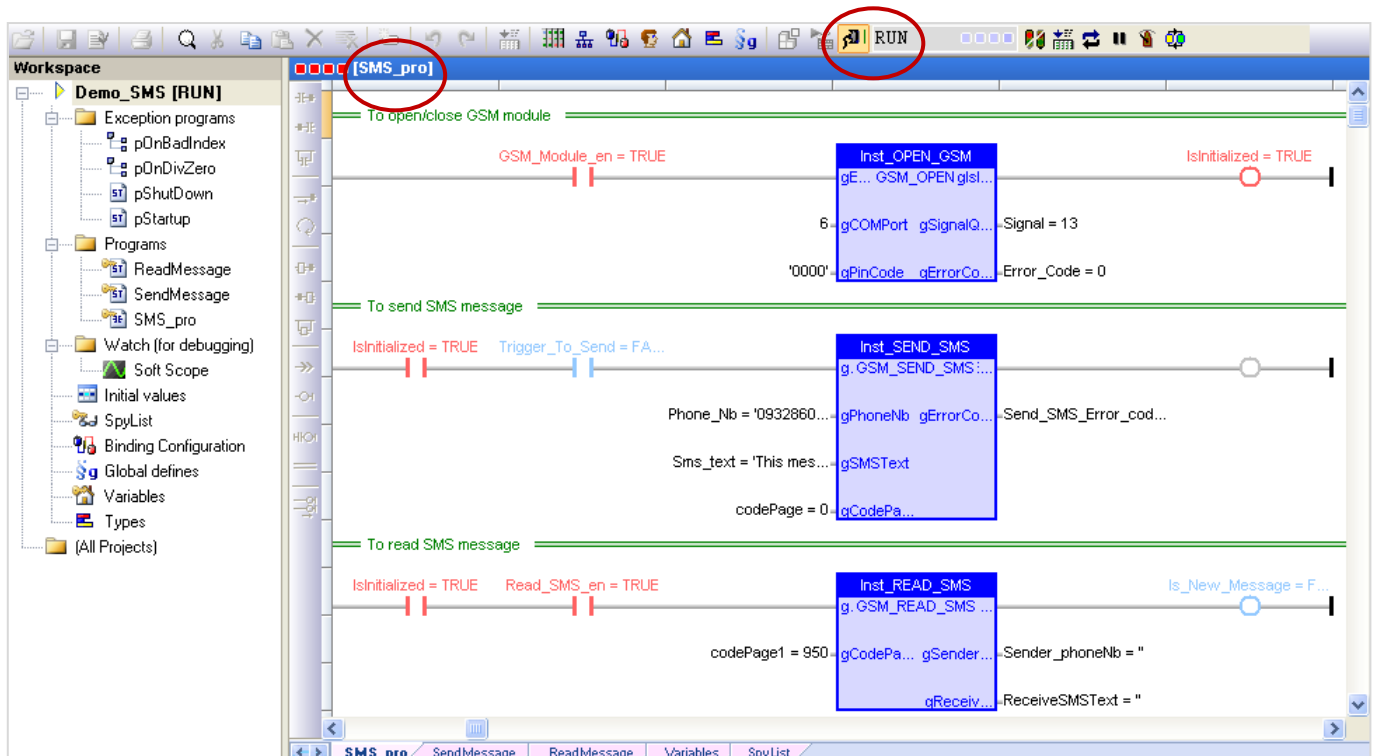
(* 加入更多的操作程序 *)

end_if;

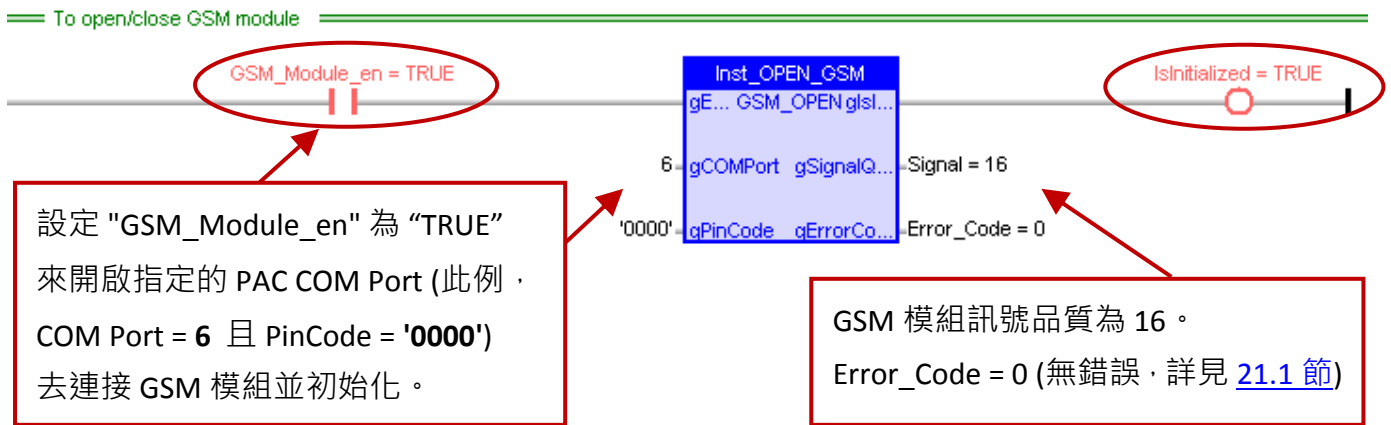
21.3 簡訊傳送測試

此例, 我們使用了一台 Win-GRAF XPAC 加上一個 3G/2G 模組 (I-8212W-3GWA + SIM 卡) 於 Slot 1。
測試前, 請將通訊參數修改為您目前 PAC 的 IP, 再重新編譯並下載程式到 Win-GRAF PAC 中。
(若不熟悉操作, 請參考 2.3.5 節)

連線後, 可見到 "SMS_pro" 程式的畫面如下:



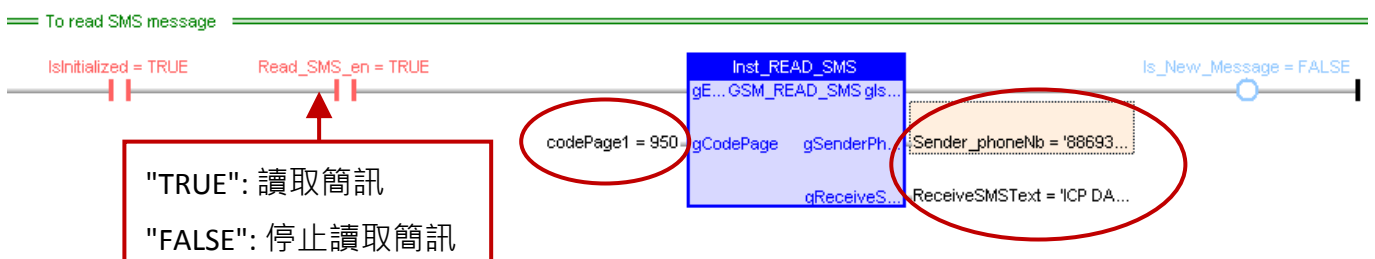
1. 填入 GSM 模組目前使用的 COM Port 與 SIM 卡的 PinCode (如果有需使用)。
此例中，GSM 模組使用了 **COM6** (參考 [19.1 節](#)) 且 SIM 卡的 PinCode = **'0000'**。



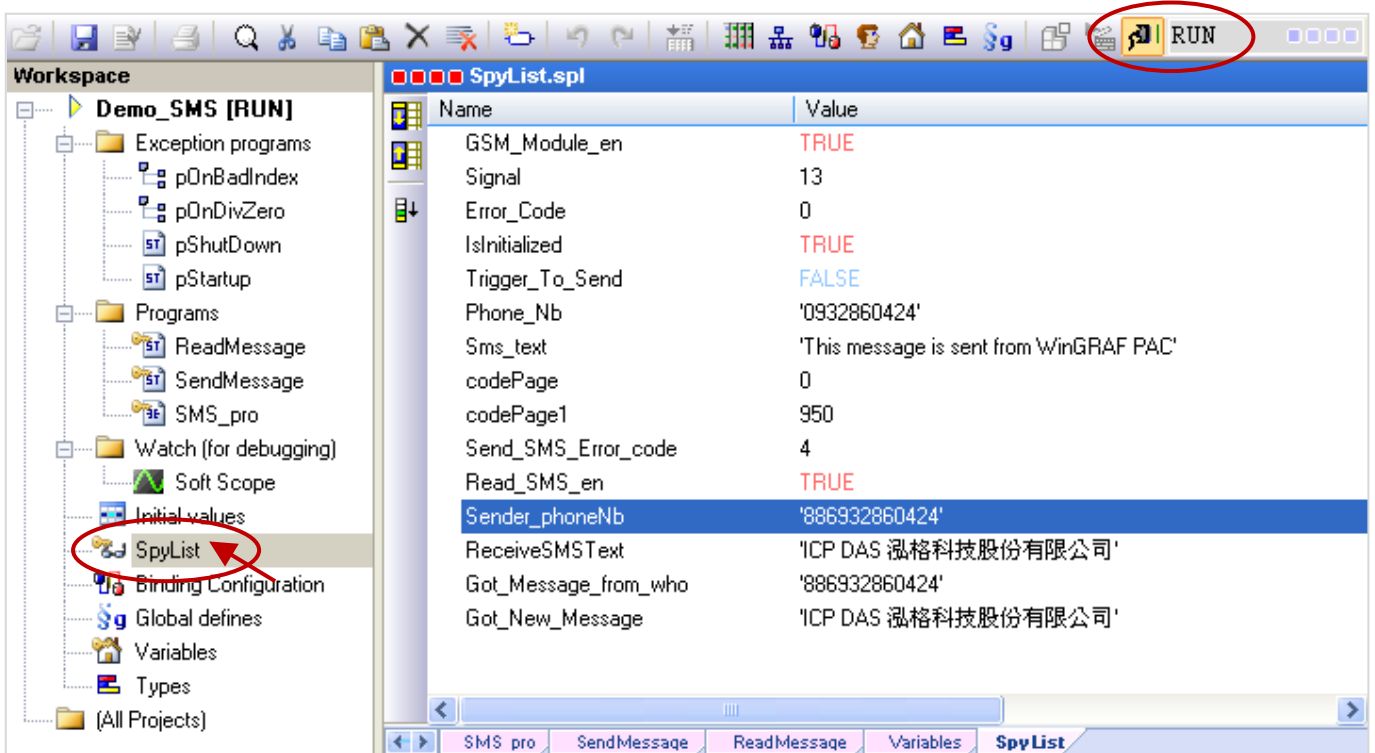
2. 填入手機號碼，簡訊內容與編碼代號 (CodePage)。
此例，PAC 將傳送 英文 (CodePage = 0) 簡訊內容 **'This message is sent from Win-GRAP PAC'** 至手機號碼 **'0932860424'** (或 **'886932860424'**)。
3. 滑鼠雙擊 "Trigger_To_Send" 設定為 "TRUE" 來傳送簡訊。此時可查看 'Send_SMS_Error_Code' 的變化，當值變為 "4" 表示傳送簡訊成功 (詳見 [21.1 節](#))。



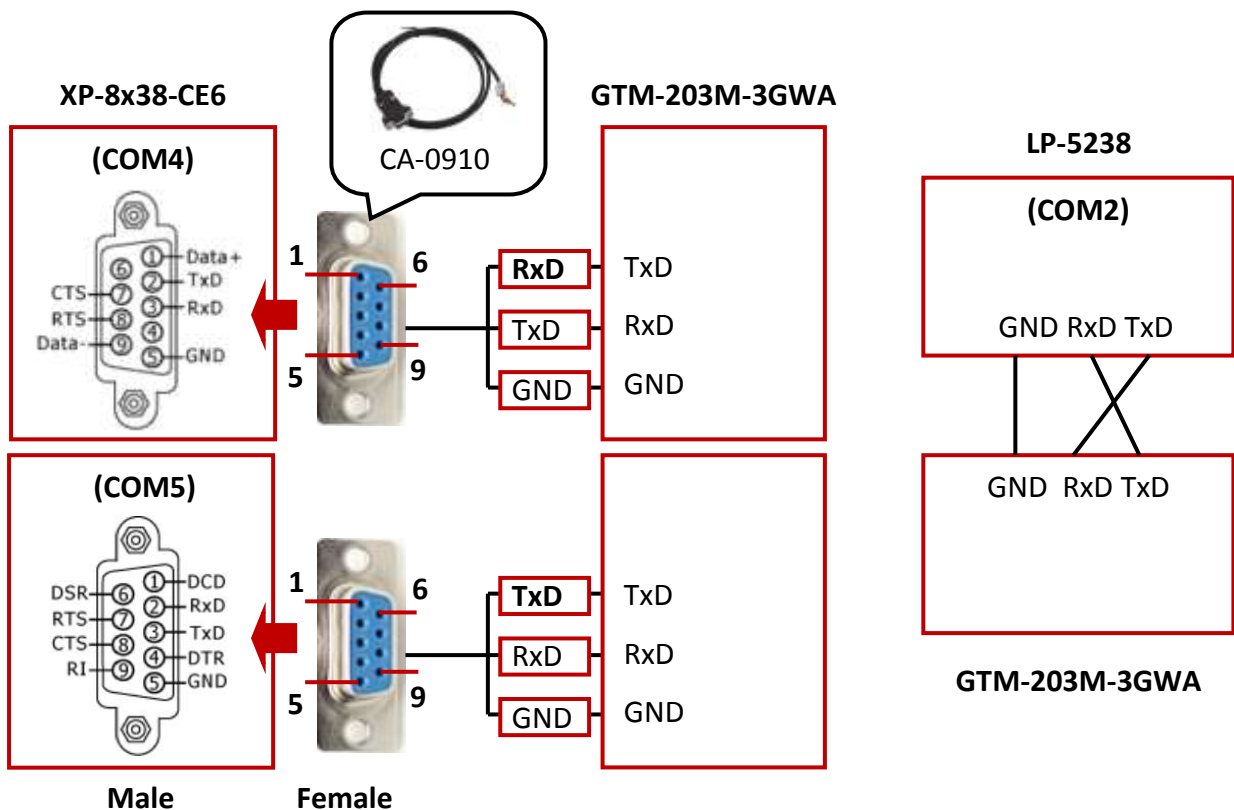
4. 為了測試 PAC 讀取簡訊的功能，請由手機號碼回覆一封簡訊至 SIM 卡。
此例，由手機號碼 **'886932860424'** 回覆一封中文 (CodePage1 = 950) 簡訊內容 **'ICP DAS 泓格科技股份有限公司'** 至 SIM 卡。



另外，使用者可滑鼠雙擊 "SpyList" 開啟變數清單，來進行傳送/接收簡訊的測試。



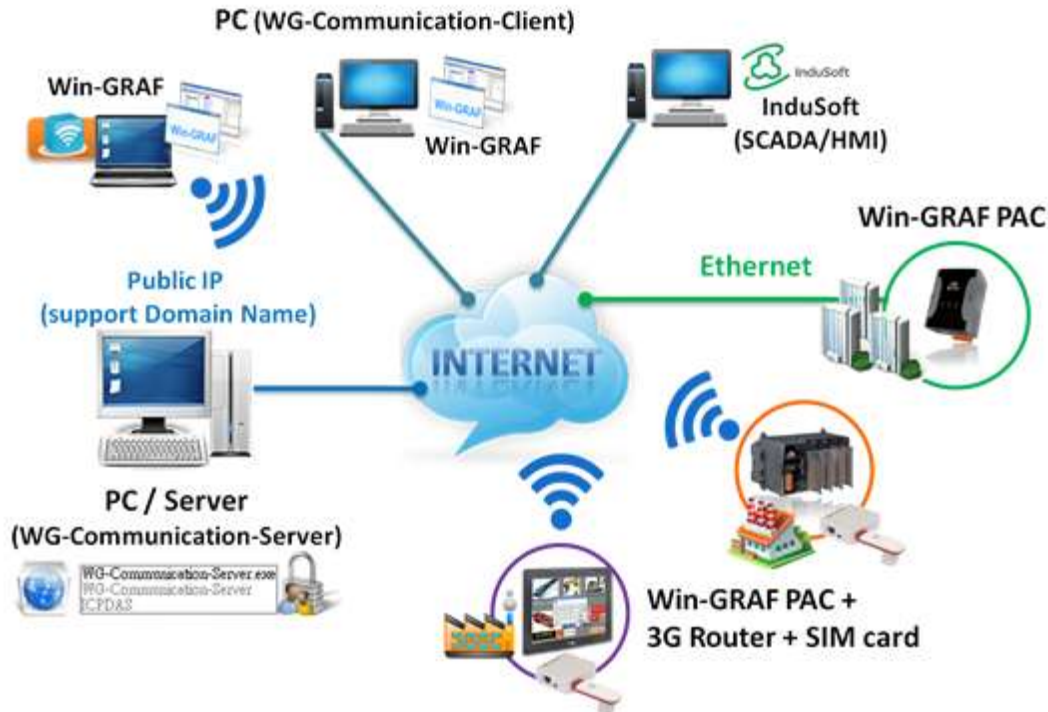
註: 若您使用 Win-GRAF PAC 的 RS-232 Port 來連接 3G/2G 通訊模組，請參考 [附錄 F](#) 來查看 COM Port 的接腳圖。(例如: 使用 XP-8xx8-CE6 的 COM4 或 COM5 的接線方式會不同)



第 22 章 Win-GRAF 3G 智能型解決方案

Win-GRAF PAC 從以下 PAC Driver 版本起，提供 3G 智能解決方案：

WP-8xx8 : 1.05 ; VP-x2x8-CE7 : 1.01 ; XP-8xx8-CE6 : 1.03 ; WP-5238-CE7 : 1.02



3G 智能解決方案的主要功能為可以讓遠方的 Win-GRAF PAC 透過 3G 訊號 或 企業網路連到一台具有實體 IP (支援網域名稱) 的 WG-Communication-Server，然後 User 的 PC/NB 即可使用 WG-Communication-Client 並上網登入 Server 來跟遠方的 PAC 通訊，之後就能達成以下功能：

1. User 的 PC/Notebook 可以執行 Win-GRAF Workbench 來對遠方的 PAC 除錯 或 更新 PAC 內的 Win-GRAF 應用程式。
2. PC 上可以執行 SCADA/HMI 軟體 (比如 InduSoft) 來監控遠方的 PAC。
3. User 的 PC/Notebook 可以視需要更新遠方 PAC 的 Win-GRAF Driver。
4. 遠方的 PAC 可以主動把 Log File 傳到 WG-Communication-Server。

註：

- (1) WG-Communication-Server 必須要具有 Internet 實體 IP 才能 Work，其它遠方的 PAC 與 User 的 PC/SCADA PC 都不必有實體 IP。
- (2) 若 PAC 所架設的工地現場已經有網路可以連上 Internet，就不需要再添購 3G Router 來上網。
- (3) 若 PAC 所架設的工地現場沒有網路，請購買 3G Router 並要插上當地申請的 3G SIM 卡來連上網路。建議 SIM 卡是選用吃到飽的方案，不然通訊費用會很龐大。
- (4) 若該工地有不只一台 PAC 要連上 WG-Communication-Server，請另架設一台 Ethernet Switch，這樣把 3G Router 接到該 Switch 後，多台 PAC 就都可以連到該 Switch 來上網。

22.1 架設一台 PC/WG-Communication-Server

User 可使用 "WG-Communication-Server" 建立多組帳號/密碼 (最多 100 組) 來供遠端 PC/NB 與 PAC 登入到 Server (**註:** PC/NB 需安裝 "[WG-Communication-Client](#)" 並使用與 PAC 相同的登入帳號/密碼) 。請將 Win-GRAF PAC 出貨光碟 (\Napdos\Win-GRAF\Tools_Utility) 內的 "WG-Communication-Server" 資料夾，複製到您 PC/Server 的 D 槽 (即，D:\WG-Communication-Server) 。

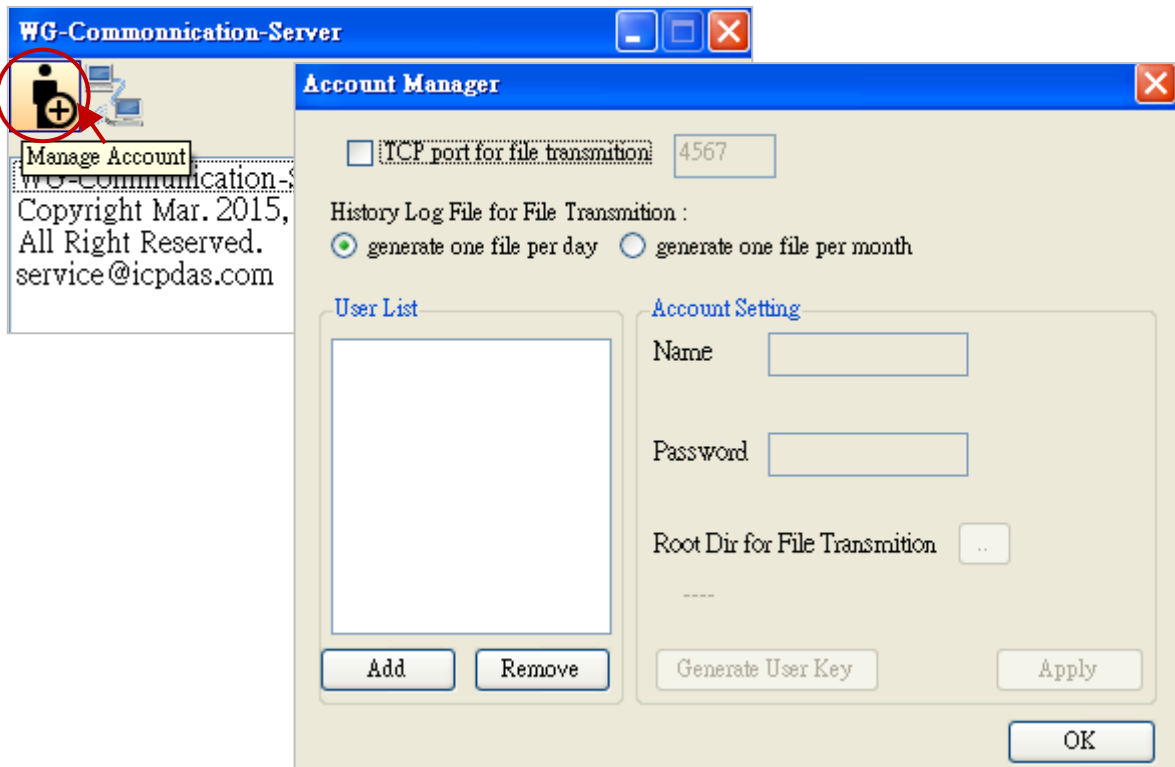
注意: [WG-Communication-Server.exe](#) 必需在此資料夾內，才可正常運作。

新增帳號/密碼:

1. 滑鼠雙擊 "WG-Communication-Server.exe" 後，程式會直接縮小至右下角的系統列中 (即，在背景運行)，如需編輯設定，請雙擊該圖示來開啟軟體。



2. 點選 "Manage Account" 圖示開啟設定畫面。



"TCP port for file transmission":

如需進行檔案傳輸，請啟用 PAC 與 PC 通訊的 TCP 埠號 (預設: 4567，範圍: 1000 ~ 9999) 。

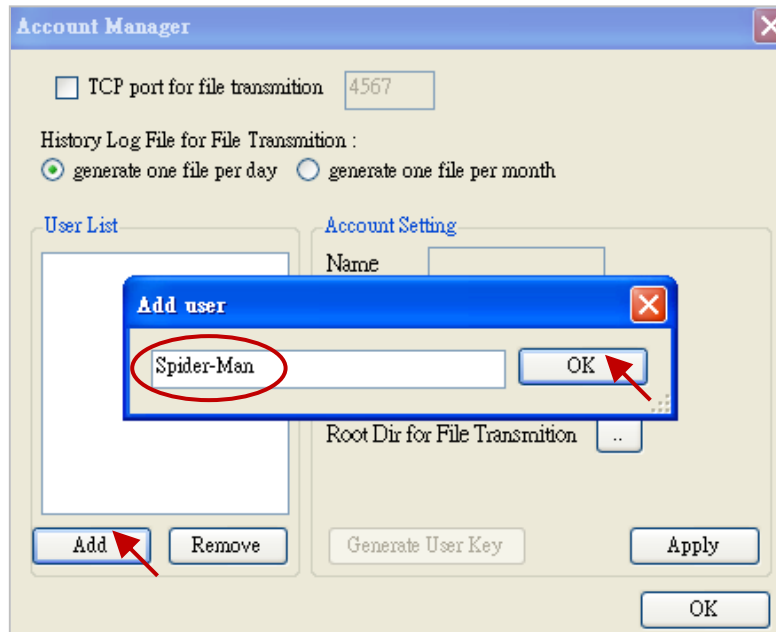
"History Log File":

選擇 **每日** 或 **每月** 產生一個歷史記錄檔 (預設: "每日") 。

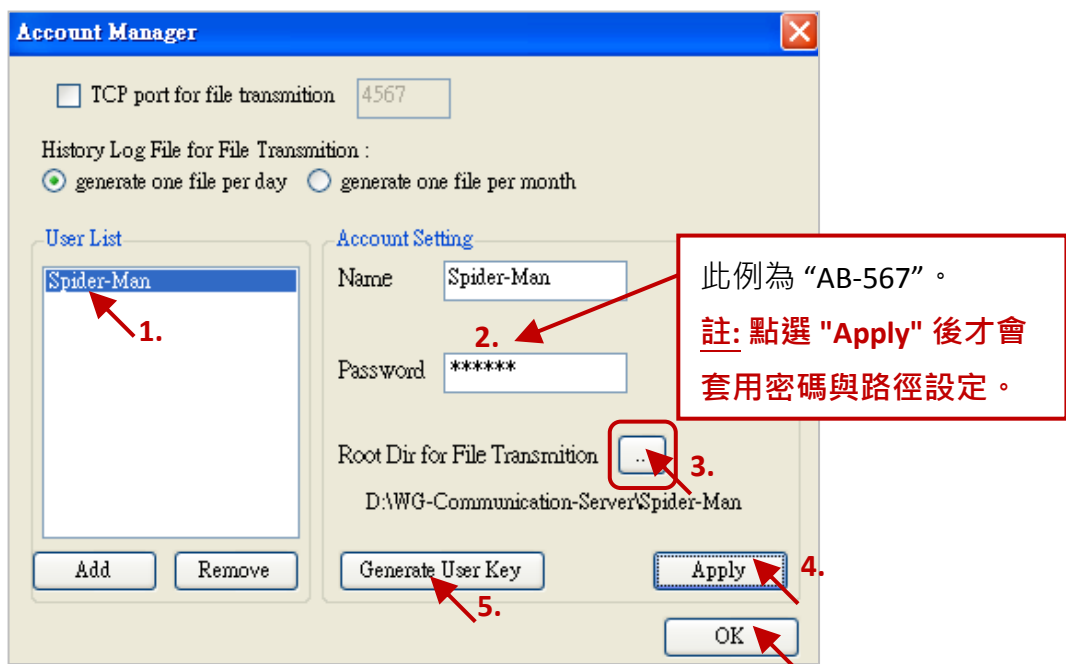
註: 此章節不需勾選 "TCP port for file transmission" (參考 [20.1 節](#))，請至下一步驟設定帳號/密碼。您可開啟 "D:\WG-Communication-Server\account.txt" 查詢已設定過的帳號/密碼。



3. 接著，點選 "Add" 按鈕並輸入使用者名稱 (例如: "Spider-Man")，再按 "OK"。



4. 滑鼠點一下使用者名稱 (例如: "Spider-Man") 來設定密碼 (例如: "AB-567")，接著設定傳輸檔案與 User Key 的存放路徑 (建議放在預設的 D:\軟體名稱\使用者名稱 或 點選 "Root Dir..." 按鈕)，再按 "Apply" 套用設定 (解開 "Generate User Key" 按鈕)。



5. 為了安全考量，遠端 PC/NB 內必須有 Server 所提供的 User Key 與帳號/密碼才能登入 Server。請點選 "Generate User Key" 產生 User Key。**注意:** 每點選 "Generate User Key" 並確認後，可產生新的 User Key，而持有舊 User Key 的 PC/NB 將會無法登入 Server (即，使用者的登入權限失效，除非再安裝新產生的 User Key)。

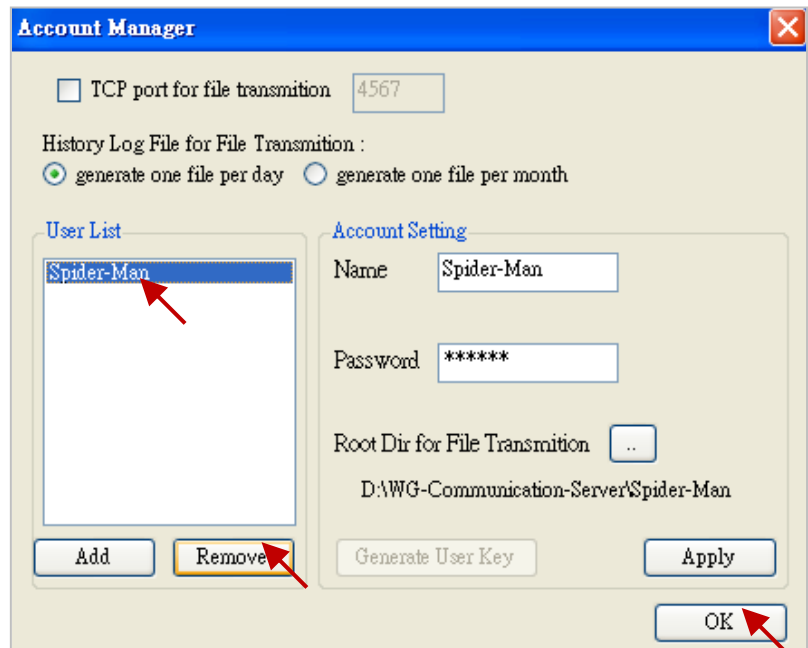


請將此 User Key 寄給遠端使用者，並複製到 User PC 的 **D:\WG-Communication-Client** 內，如此，遠端 PC 即可擁有登入權限，並使用該帳號/密碼登入 Server。



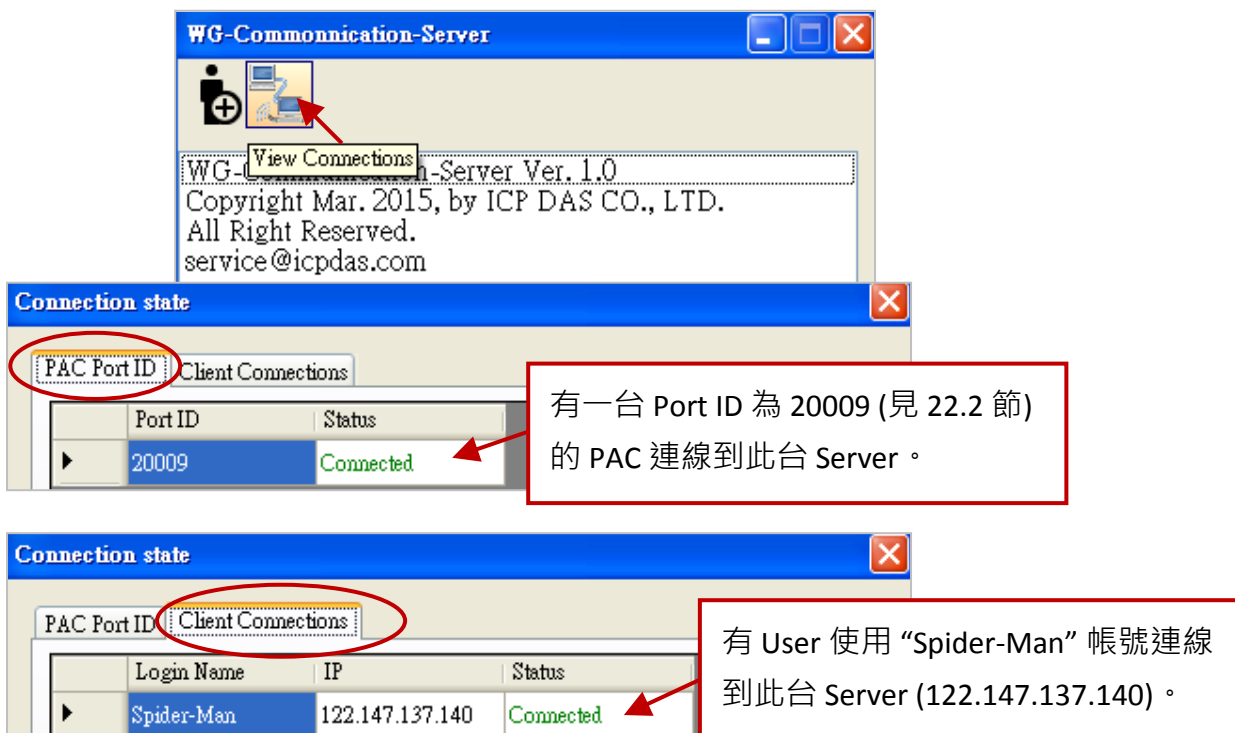
刪除帳號/密碼

點選欲刪除的使用者名稱 (例如: "Spider-Man") 再點選 "Remove" 按鈕，最後點選 "OK" 刪除此帳號/密碼。



查看連線狀態

您可使用此功能來查看連線到此 Server 的 PC/NB 或 Win-GRAF PAC。



22.2 架設遠方的 PAC 來連上 WG-Communication-Server

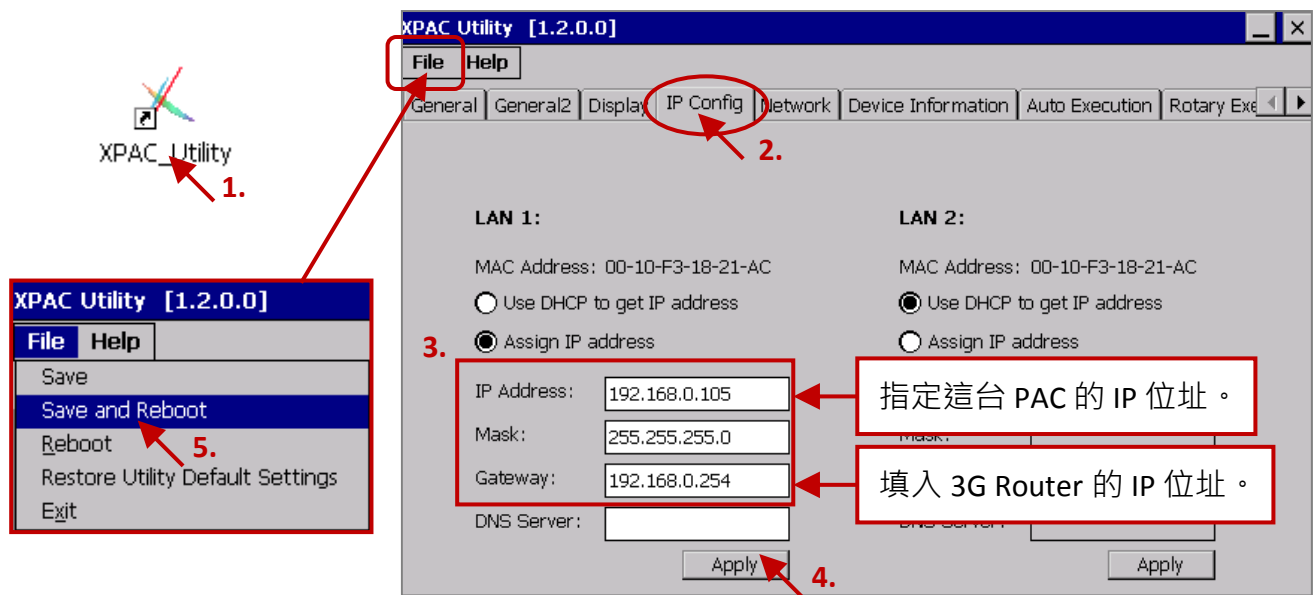
若 Win-GRAF PAC 使用 3G Router (加 SIM 卡) 來上網，請先參考該產品的使用說明書來進行設定。

注意：市面上很多 3G Router 都內建 Wireless WiFi 功能，為了 PAC 通訊上安全，建議 User 在設定 3G Router 時把它的 Wireless 功能關閉 (Disable)。

PAC 內的網路設定

請依照案場實際的網路配置來設定，假設 3G Router 當成 Gateway 的 IP 位址設定為 192.168.0.254，並開放了 192.168.0.100 ~ 192.168.0.200 給 PAC 使用，則 PAC 內可設定如下圖：

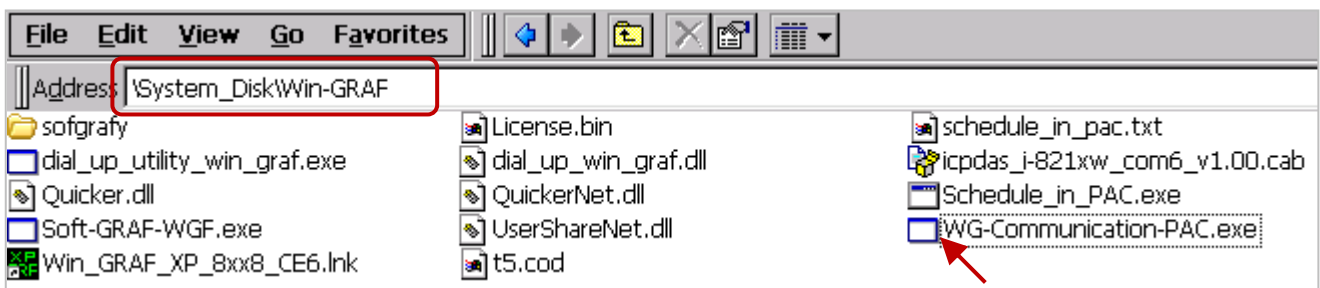
- 滑鼠雙擊桌面上的 PAC Utility (例如: XPAC_Utility)，接著切換到 IP 設定頁面 (例如: IP Config) 來填入適當的 IP/Mask/Gateway 位址，再按 “Apply”。最後，再執行 “File → Save and Reboot” 儲存並重新啟動 PAC。



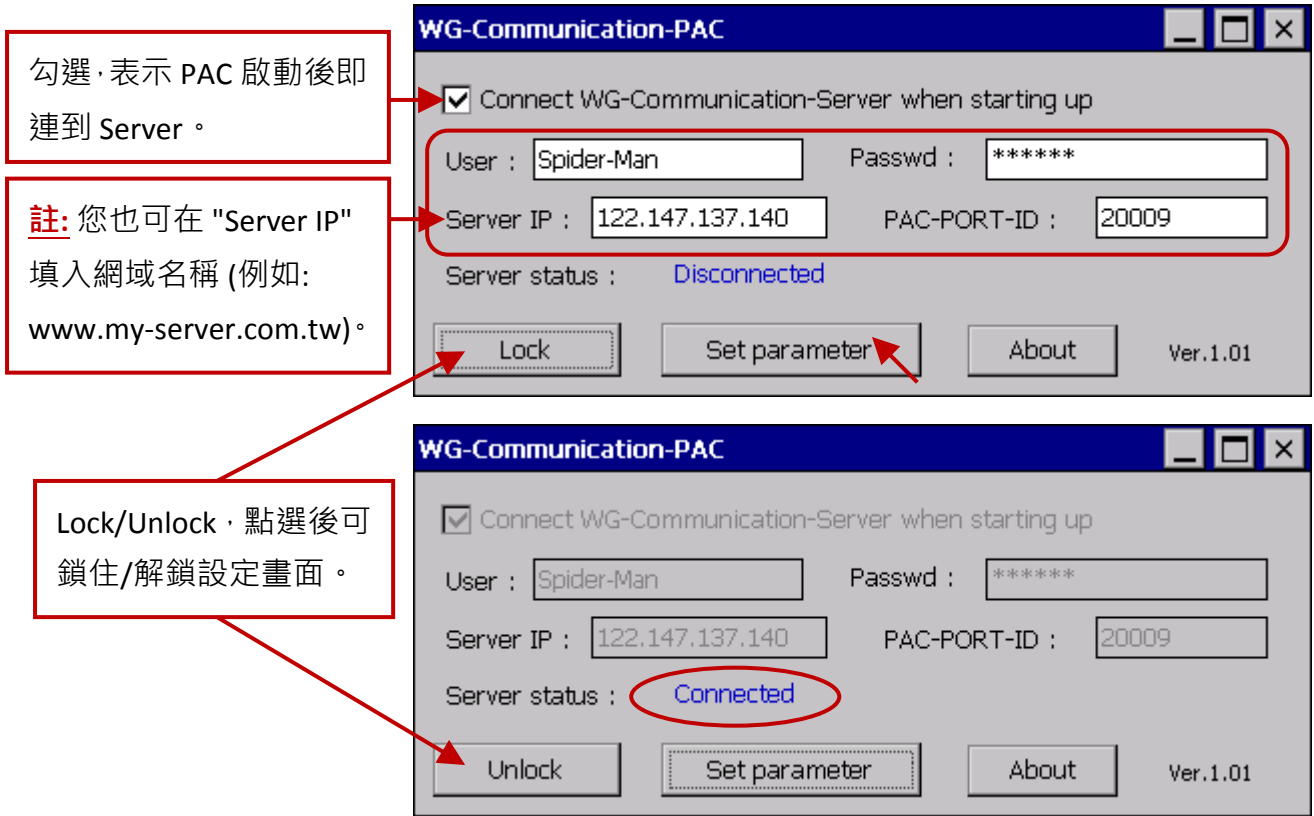
PAC 內使用 “WG-Communication-PAC” 連上 Server

您可在 PAC 內的 \System_Disk\Win-GRAF 路徑下找到 WG-Communication-PAC.exe，並使用它來連線到 WG-Communication-Server。

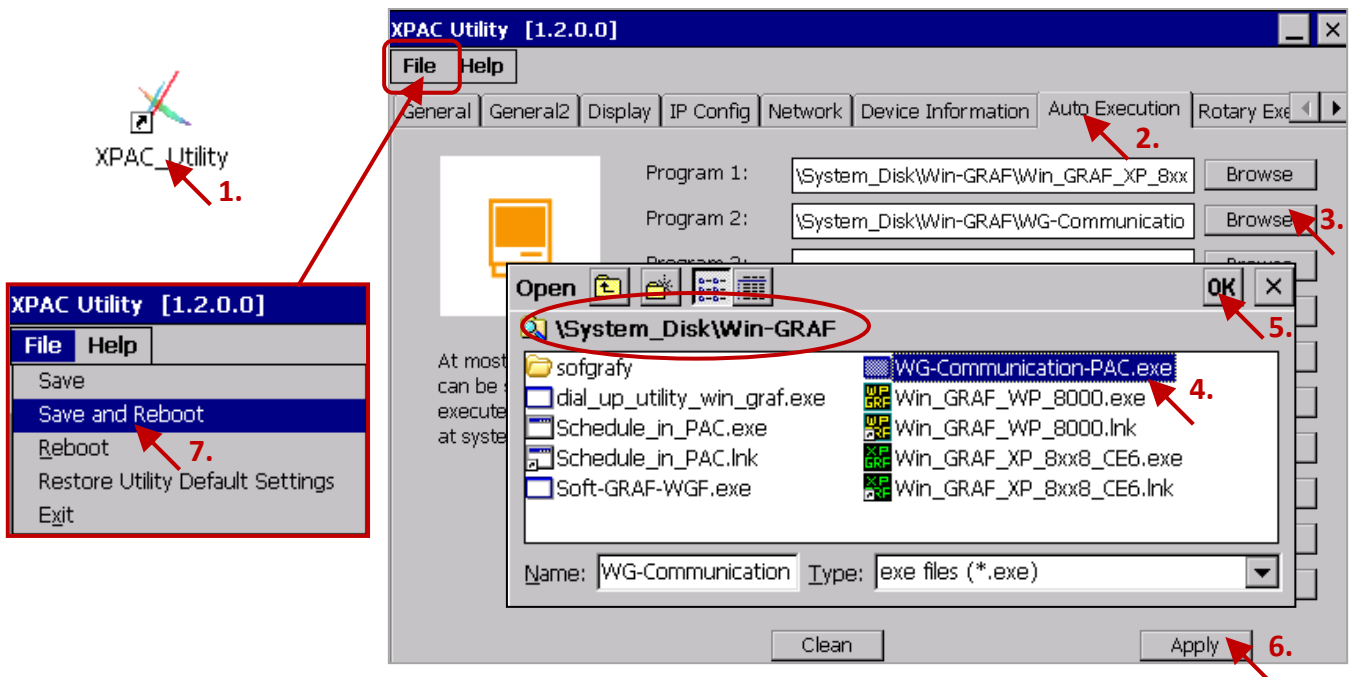
1. 滑鼠雙擊 “WG-Communication-PAC.exe” 開啟設定畫面。



- 請填入 WG-Communication-Server 所建立的登入帳號/密碼 (例如: Spider-Man/ AB-567 · 參考 22.1 節) · 再填入 Server 的 IP · 接著指定這台 PAC 的 Port ID (例如: 20009 · 範圍: 20000 ~ 22000)。
註: PAC-Port-ID 可供遠端的 Server 或 User PC 識別是和哪一台 PAC 連線。不同的 PAC 連線至同一台 Server · 需使用個別的 PAC-Port-ID。
- 最後 · 請點選 "Set parameter" 來完成參數設定。



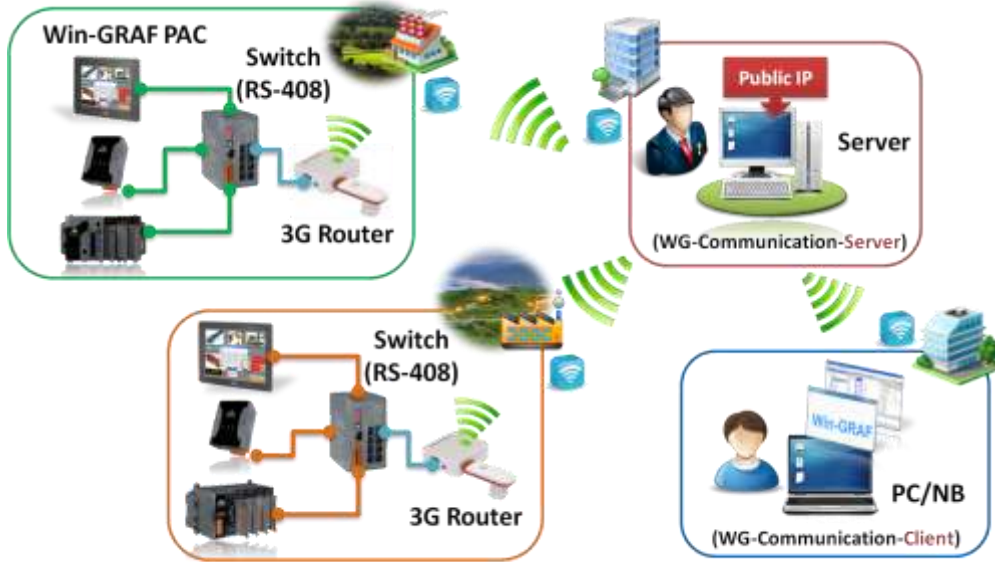
此外 · 請開啟 PAC Utility 將 “WG-Communication-PAC.exe” 加入開機後的自動執行項目 · 再執行 “File → Save and Reboot” 儲存並重新啟動 PAC。



22.3 在 User PC 與 SCADA PC 架設 WG-Communication-Client

您可在 PC 安裝 "WG-Communication-Client" 來連線至 Server，並對遠端 PAC 進行除錯、下載或更新 Win-GRAF Driver。請將 Win-GRAF PAC 出貨光碟 (\Napdos\Win-GRAF\Tools_Utility) 內的 "WG-Communication-Client" 資料夾，複製到您 PC 的 D 槽 (即，D:\WG-Communication-Client)。

注意: WG-Communication-Client.exe 必需在此資料夾內，才可正常運作。



22.3.1 由 PC 登入 Server 並編輯欲偵測的 PAC 連線

登入 Server 之前，請確認 "WG-Communication-Client" 資料夾內已有 Server 端所提供的 User Key。

1. 滑鼠雙擊 "WG-Communication-Client" 開啟設定畫面，填入帳號/密碼與 Server IP，並點選 "Connect" 按鈕。

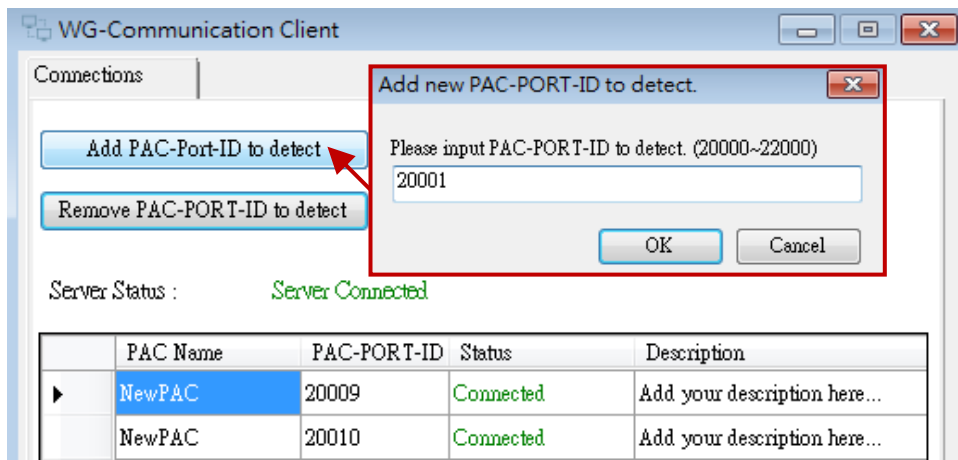
The screenshot shows the file explorer for the D:\WG-Communication-Client folder. The files listed are Spider-Man-key.bin (labeled as User Key), WG-AgentClient.dll, and WG-Communication-Client. The WG-Communication-Client file is selected. To the right, the WG-Communication Client settings window is open, showing the following fields: User Name: Spider-Man, Password: [masked], Server IP: 122.147.137.140, and a checkbox for Remember Account Setting. The Connect button is highlighted.

注意: 當 Server 斷線時, Client 端會自動登出, 並每隔 30 秒嘗試連線一次。

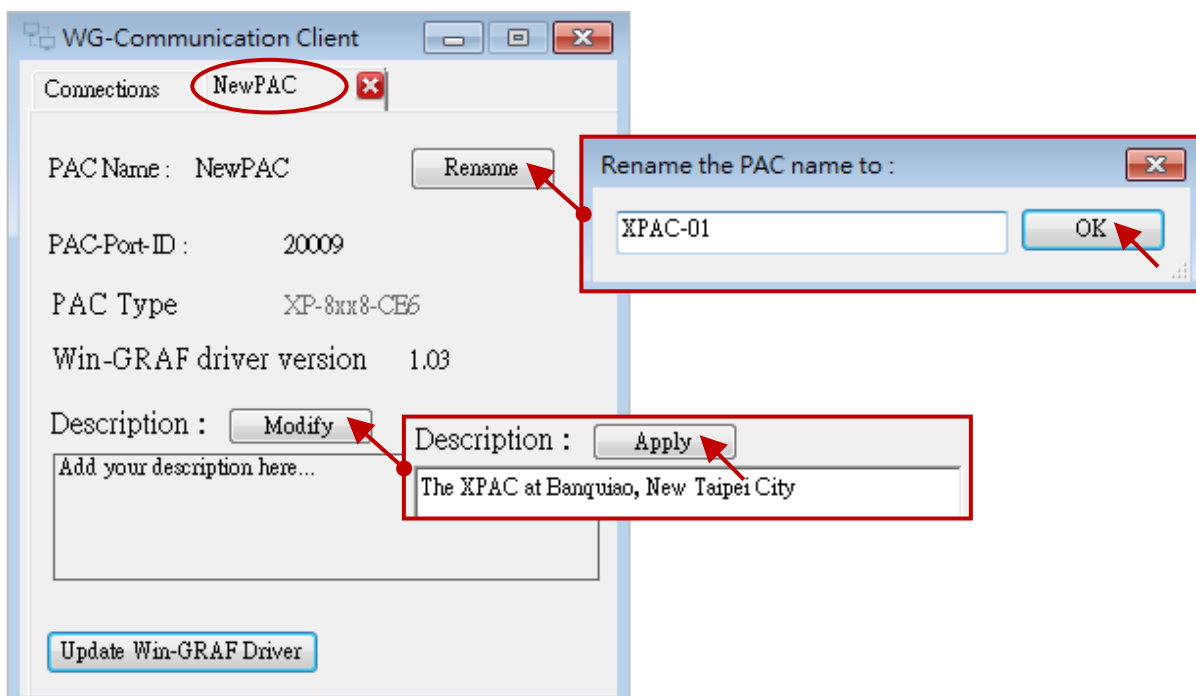
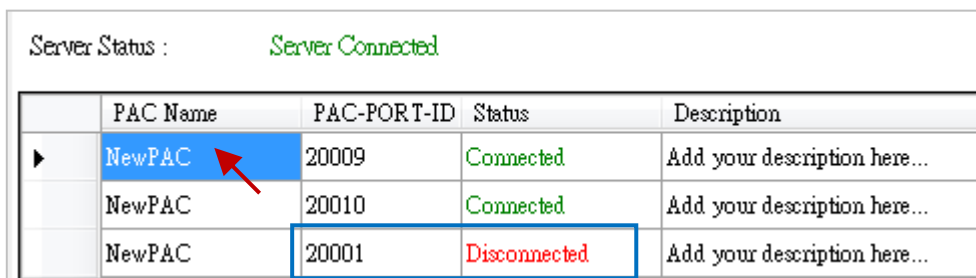
填入 Server 端提供的帳號/密碼與 IP 位址。
註: 您也可在 "Server IP" 填入網域名稱 (例如: www.my-server.com.tw)。

勾選可儲存帳號設定。

2. 連線到 Server 後，可見到 "Server Status" 顯示 "Server Connected"，且下表會列出 Server 與 PAC 的連線狀態 (參考 22.2 節)。可點選 "Add PAC-Port-ID to detect" 按鈕，並輸入 PAC-PORT-ID (例如: 20001) 來加入連線偵測清單中的 PAC 項目，也可點選 "Remove PAC-Port-ID to detect" 將該 PAC 項目從清單中移除。



3. 滑鼠雙擊任一 PAC 項目開啟設定頁籤，您可點選 "Rename" 按鈕來修改 PAC 的識別名稱，也可點選 "Modify" 按鈕新增對 PAC 的描述說明，再點選 "Apply" 套用設定。



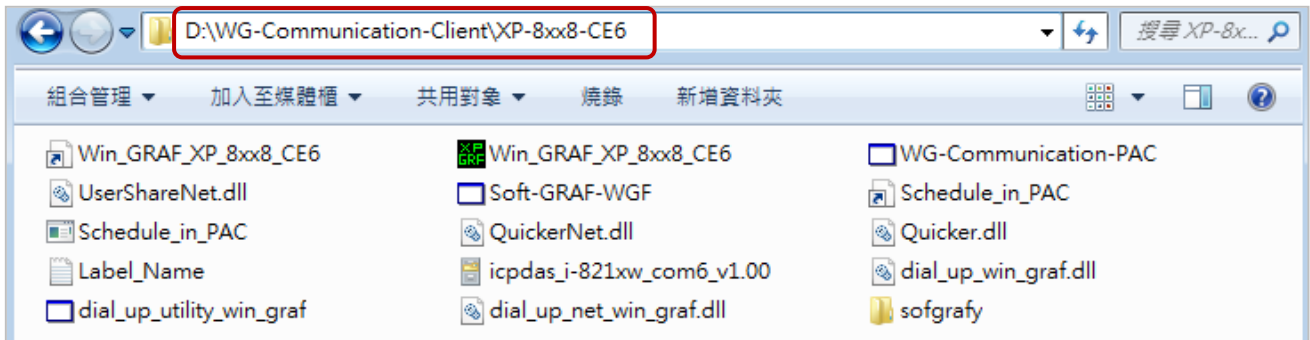
由 PC 更新遠端 PAC 的 Win-GRAF Driver

一般情況下並不需要更新 PAC 內的 Win-GRAF Driver，但若您需要使用有新功能或修正錯誤的版本時，就會需要更新 Driver。Win-GRAF PAC 最新發行的 Driver 可在網站下載：

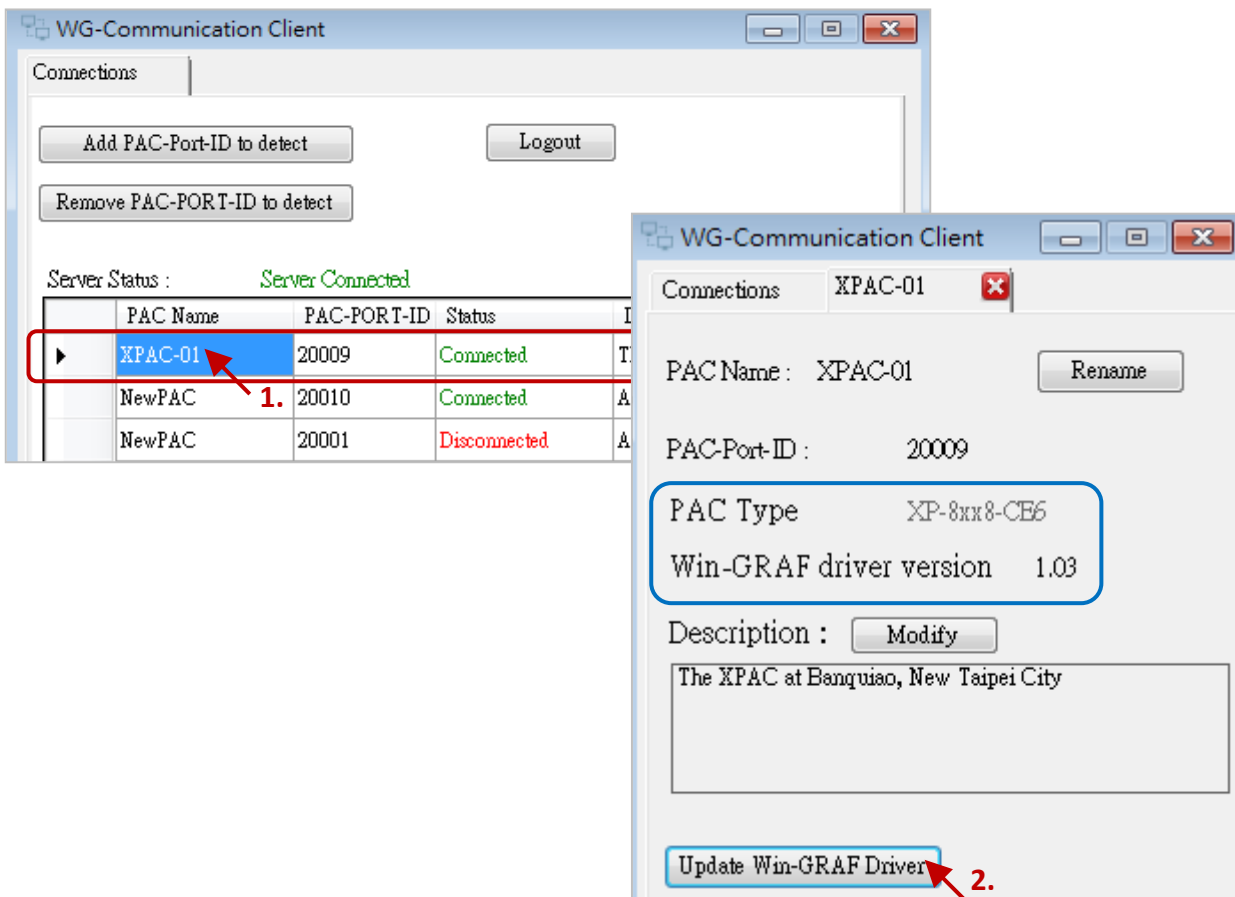
www.icpdas.com/root/product/solutions/softplc_based_on_pac/win-graf/download/win-graf-driver.html

更新前，請確認您已將 Win-GRAF Driver 檔案複製到適當的資料夾內：

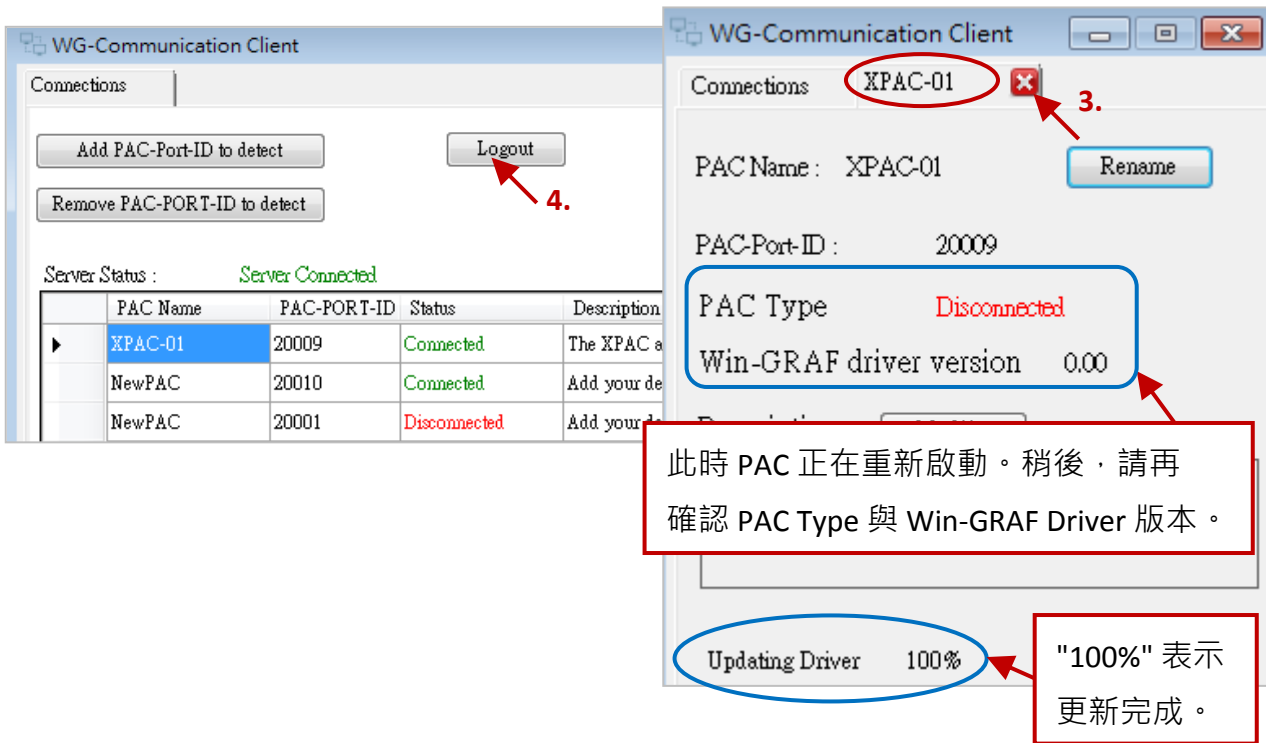
以 **XP-8xx8-CE6** 為例，請先將 Driver 檔 (.zip) 解壓縮，並將 \xp-8xx8-ce6-driver-1.xx\1.xx (例如: 1.03) 資料夾內的所有檔案複製到 D:\WG-Communication-Client\XP-8xx8-CE6。



1. 請先開啟 D:\WG-Communication-Client\WG-Communication-Client.exe 並登入到 Server。
(若不熟悉操作，可參考 [22.3.1 節](#)。)
2. 滑鼠雙擊欲更新且已連上 Server 的 PAC 名稱，再點選“Update Win-GRAF Driver” 按鈕。
註: 更新 Driver 時，請勿重啟 PAC、停止 Client 或 關閉此頁籤，且一次只能更新一台 PAC。



3. 下圖的“Updating Driver”顯示 100% 時，表示已更新完成。請等待約 60~90 秒，當 PAC Type 為 “Disconnected” 表示 PAC 正在自動重開機，之後，請再確認 PAC Type 與 Win-GRAF Driver 版本。
4. 關閉 “XPAC-01” 頁籤後，您可繼續對其它 PAC 進行操作 或 點選 “Logout” 按鈕來登出 Server。



22.3.2 由 PC/Win-GRAF Workbench 更新遠端 PAC 的 Win-GRAF 專案

本章節將說明 PC/Win-GRAF Workbench 如何藉由 WG-Communication-Client 連上 Server 並下載或更新遠端 PAC 的 Win-GRAF 專案。

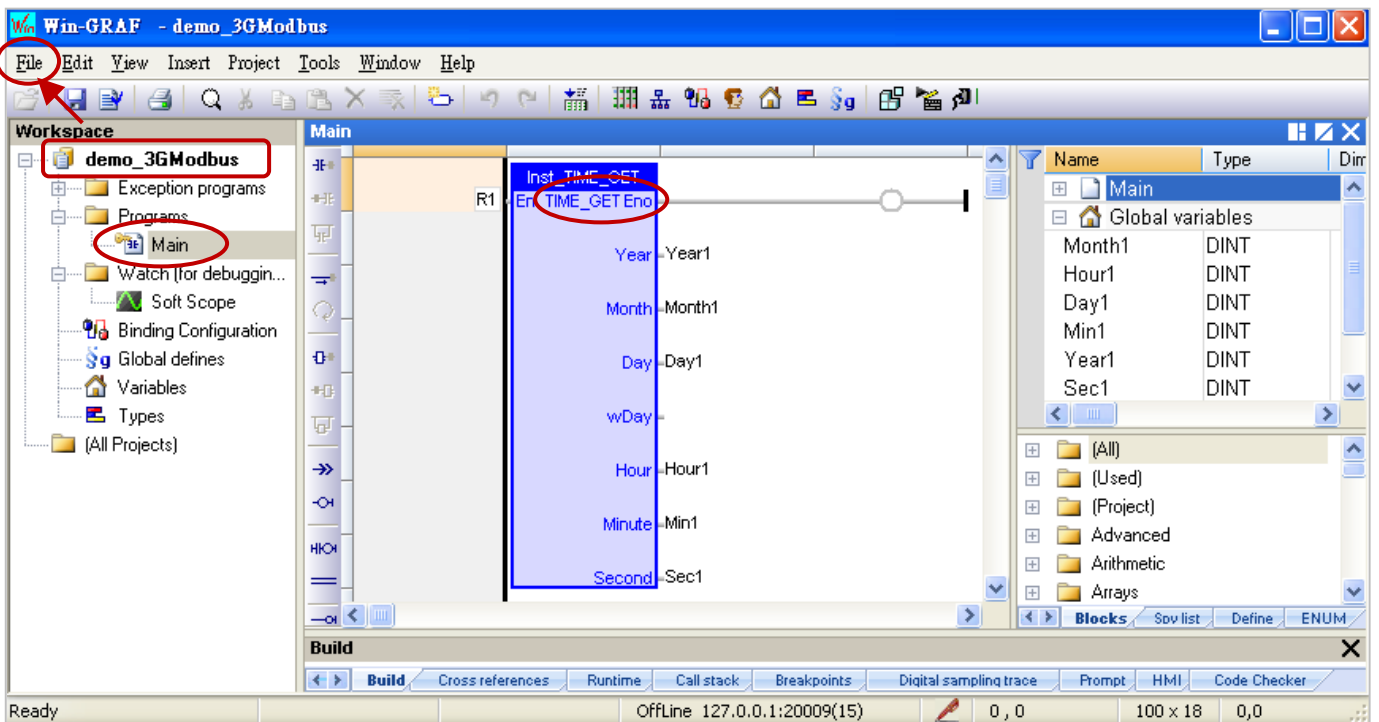


1. 請先開啟 D:\WG-Communication-Client\WG-Communication-Client.exe 並登入到 Server。(可參考 22.3.1 節。)

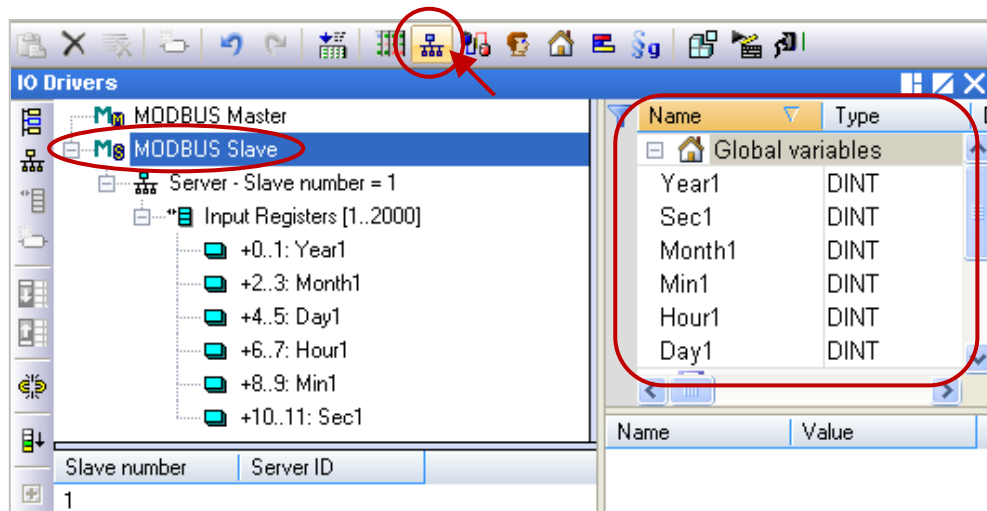


2. 執行 Win-GRAF Workbench 並點選 "File → Add Existing Project → From Zip..." 開啟 Win-GRAF PAC 出貨光碟 (\Napdos\Win-GRAF\demo-project) 內的 Win-GRAF 專案 (demo_3GModbus.zip) 。

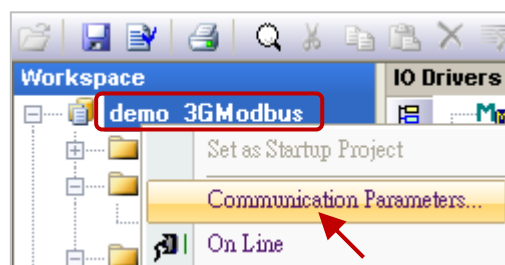
此範例包含了一個 LD 程式 (Main) ，使用了 "Time_Get" 功能方塊來讀取系統時間。



同時，啟用 Modbus Slave 來開放此專案中的變數，可供 HMI / SCADA (例如: InduSoft) / Modbus Master 進行讀取 (設定方式可參考 3.1 節) 。



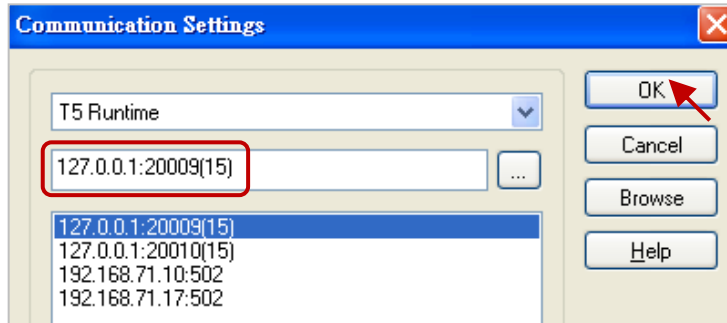
3. 滑鼠右鍵點選專案名稱 (demo_3GModbus) ，再點選 "Communication Parameters.." 設定通訊 IP 。



設定方式為 **127.0.0.1:PAC-PORT-ID(Timeout)**，此例設定為 127.0.0.1:20009(15)。

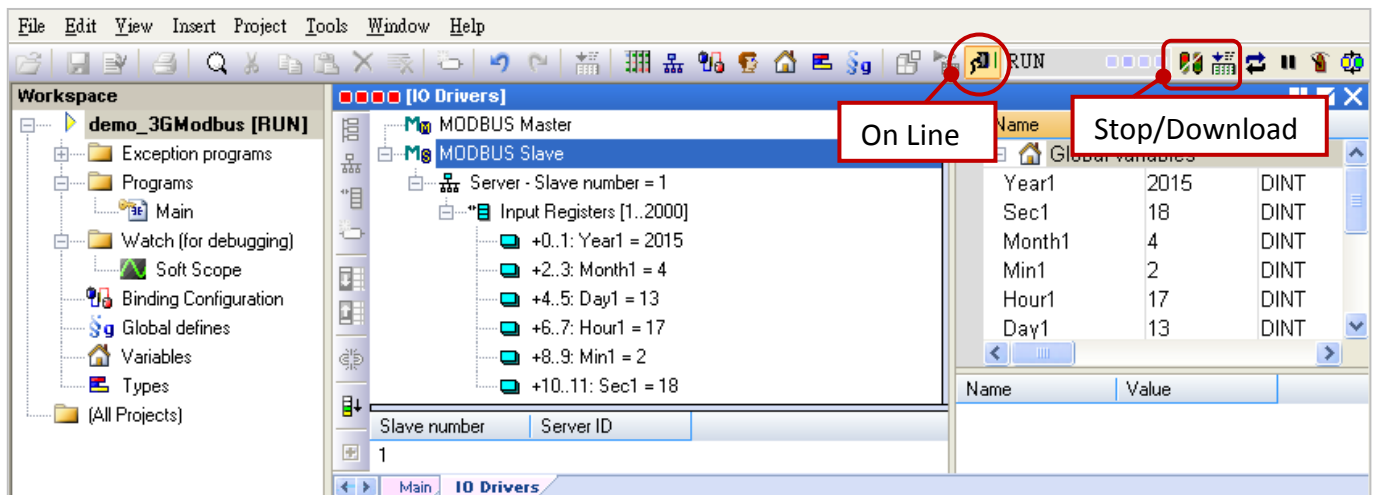
註: PAC-PORT-ID 為 PAC 的識別 ID，請確認該 PAC 已連上 Server (可參考 22.2 節)。

Timeout 預設為 3 秒，若 PAC 使用 3G Router (加 SIM 卡) 來上網，建議設定為 15 ~ 30 秒，由於 3G 是無線網路且透過基地台來傳輸資料，有可能常發生通訊延遲 或異常現象，建議將 Timeout 設定大一些 (例如: 15 ~ 30 秒)，可降低通訊 Timeout 或 讀/寫錯誤的次數。



4. 點選 “On Line” 來連線並下載專案至 PAC。

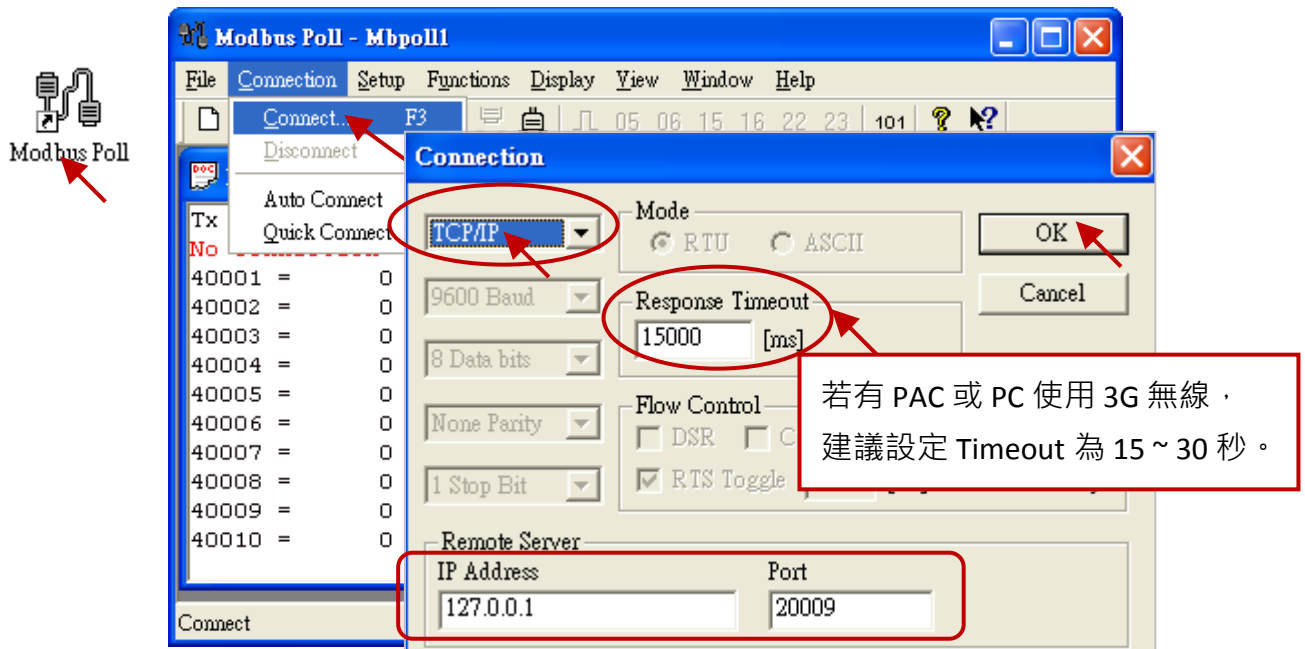
註: 若程式有修改需先編譯程式 (可參考 2.3.4 節) 再下載; 若 PAC 原先運行其它專案，需再點選 “Stop” 按鈕停止運行，再點選 “Download” 下載目前的專案 (可參考 2.3.5 節 - 步驟 4 ~ 6)。



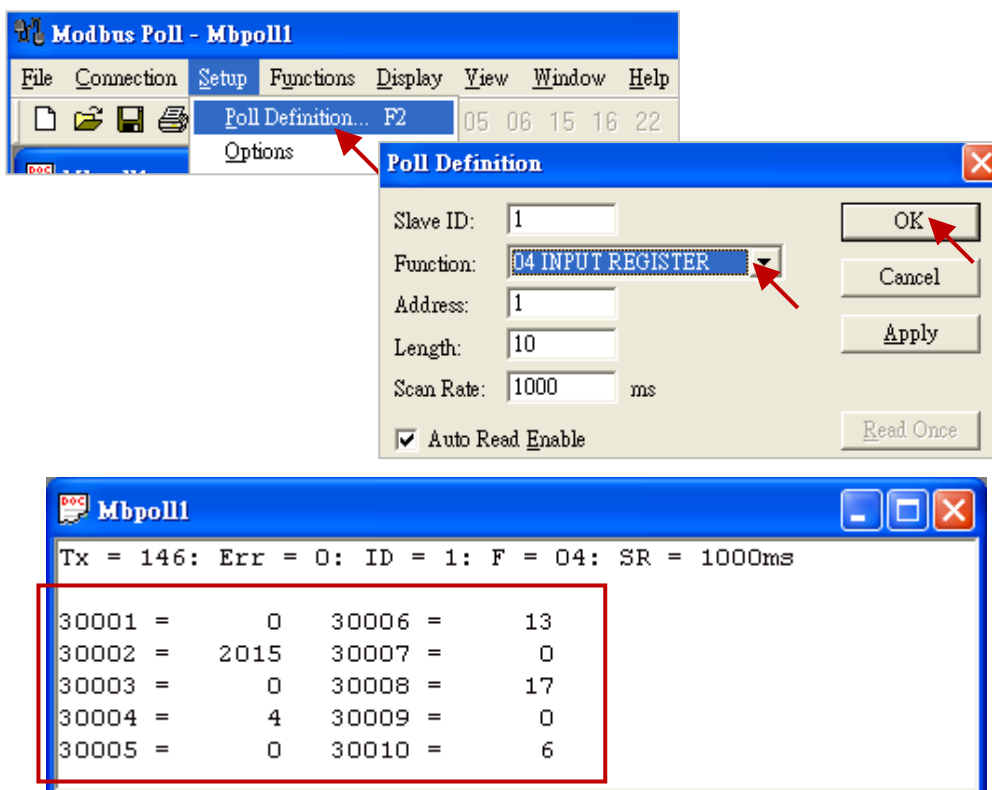
22.3.3 由 PC/SCADA 讀取/寫入遠端 PAC 的資料

不同的 HMI/SCADA 軟體若要透過 WG-Communication-Server 來跟遠方的 PAC 使用 Modbus TCP 通訊，必須在該 HMI/SCADA PC 上 Run WG-Communication-Client 並連上 Server，然後，再設定 SCADA 軟體要連上的 PAC IP 固定為 127.0.0.1，Port 為該 PAC 的 PAC-PORT-ID (20000 ~ 22000)，本章節將在 PC 上使用 Modbus Poll 軟體為例，模擬並說明 HMI/SCADA/Modbus Master 如何經由 WG-Communication-Client 讀取遠方 PAC 的資料。您可在網站下載並安裝 Modbus Poll: http://www.modbustools.com/modbus_poll.html

1. 請先開啟 D:\WG-Communication-Client\WG-Communication-Client.exe 並登入到 Server。
(若不熟悉操作，可參考 [22.3.1 節](#)。)
2. 滑鼠雙擊 “Modbus Poll” 圖示開啟此軟體，接著點選 “Connection → Connect” 進行連線設定。
連線方式選擇 “TCP/IP”，“Response Timeout” 建議設定為 15 ~ 30 秒，“IP Address” 固定填入
“127.0.0.1”，“Port” 輸入欲進行測試的 PAC 的識別 ID (例如: 20009，可參考 [22.2 節](#))。

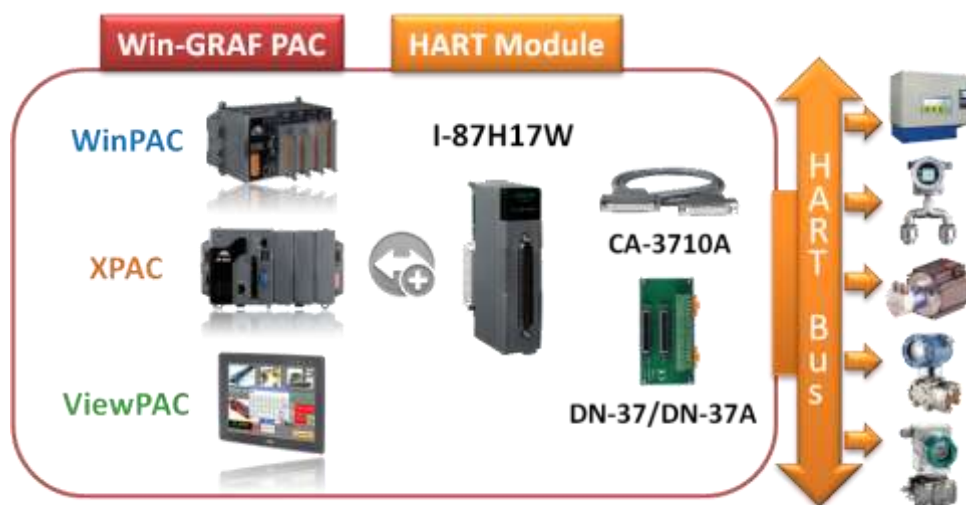


3. 點選 “Setup → Poll Definition” 設定欲讀取的資料格式，“Function” 選擇 “04 INPUT REGISTER” (參考 [上一節](#) – 步驟 2) 並按 “OK” 即可見到 Modbus Poll 讀取到的資料。



第 23 章 HART Master

Win-GRAF PAC 從以下 PAC Driver 版本起，支持 I-87H17W HART Master 模組。
WP-8xx8 : 1.07 ; XP-8xx8-CE6 : 1.05 ;



本章節會提供 3 個範例程式，用來說明 Win-GRAF PAC 搭配 I-87H17W HART 模組，與 HART 設備進行通訊的方式。

23.1 HART 模組 - I-87H17W 介紹

I-87H17W : http://www.icpdas.com/products/Remote_IO/can_bus/i-87h17w.htm

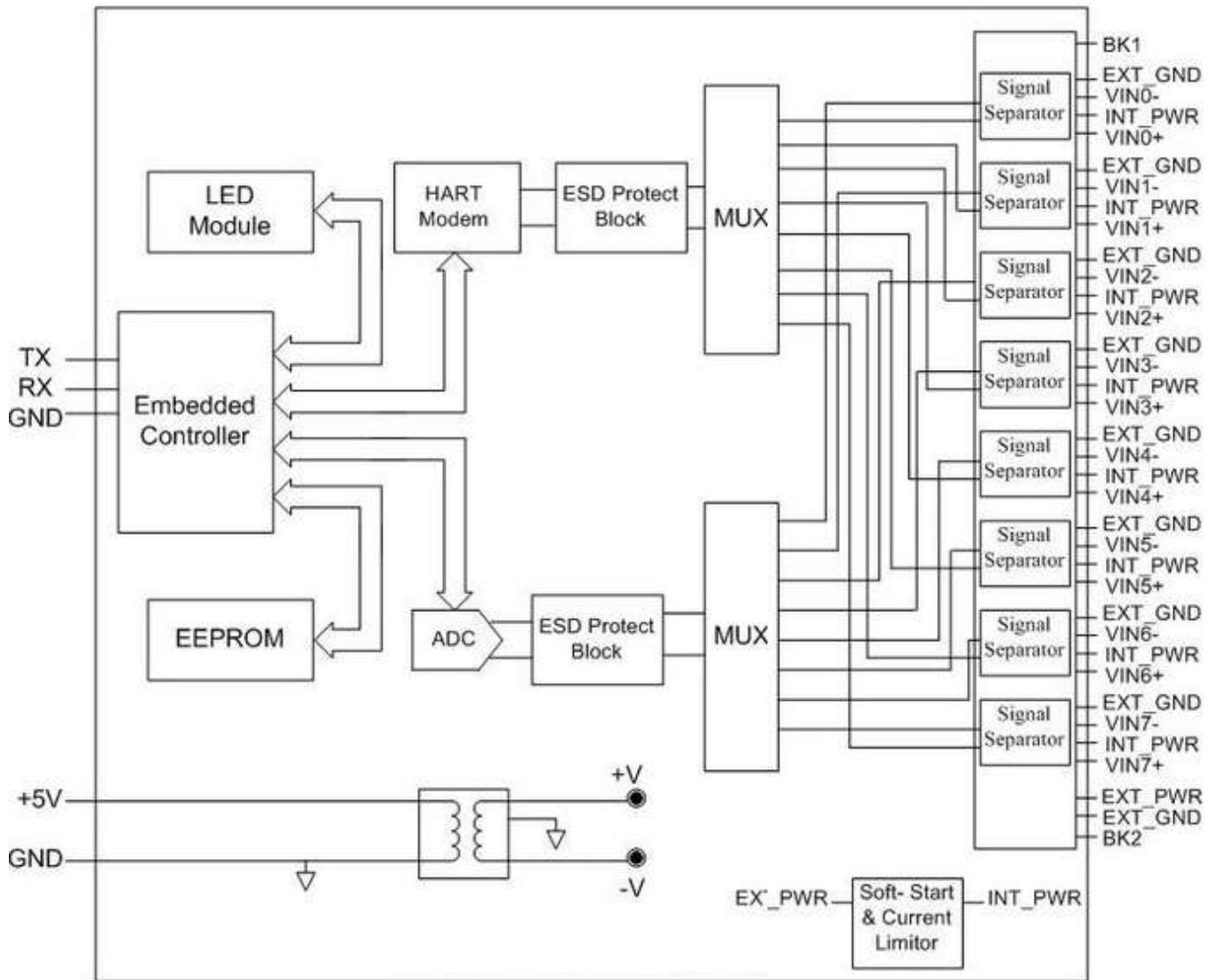
I-87H17W 可以插在 XP-8xx8-CE6 的 Slot 1 ~ 7 (第 1 個 I/O Slot 編號是 "1")，也可以插在 WP-8xx8 的 Slot 0 ~ 7。Win-GRAF 不支援 I-87H17W 插在 RS-485 擴展單元上使用 (例如: I-87K8 或 RU-87P8)，只能把它插在主機上的 I/O Slot 來使用。

每個 I-87H17W 有 8 個 AI 通道，可用來量測 4 ~ 20 mA，也可用來當成 8 通道的 HART 通訊埠，建議每個 HART 通訊埠只接一個 HART 設備。這 8 個 HART 通訊埠是共用模組上的一個通訊晶片 (Chip)，所以同一時間點只能使用 1 個 Port 來通訊，無法在同一個時間點命令同一個 I-87H17W 的 2 個或以上的 Port 同時發送/接收 HART 通訊資料，但可以在 Win-GRAF 程式內，分開時間輪流處理 1 ~ 8 個 HART 通訊埠來進行通訊 (例如，範例程式 - Demo_HART_3.zip)。

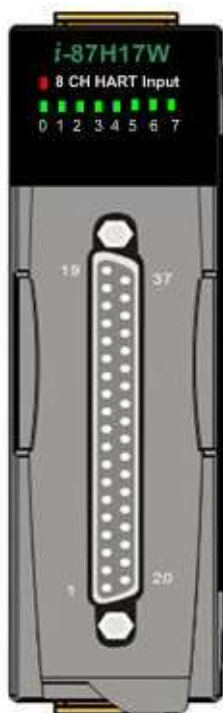
硬體接腳圖:

您可至 I-87H17W 網頁 - http://www.icpdas.com/products/Remote_IO/can_bus/i-87h17w.htm 參考相關的硬體說明，以下只列出它的內部 I/O 結構，接腳圖與接線圖。

I-87H17W 內部 I/O 結構:



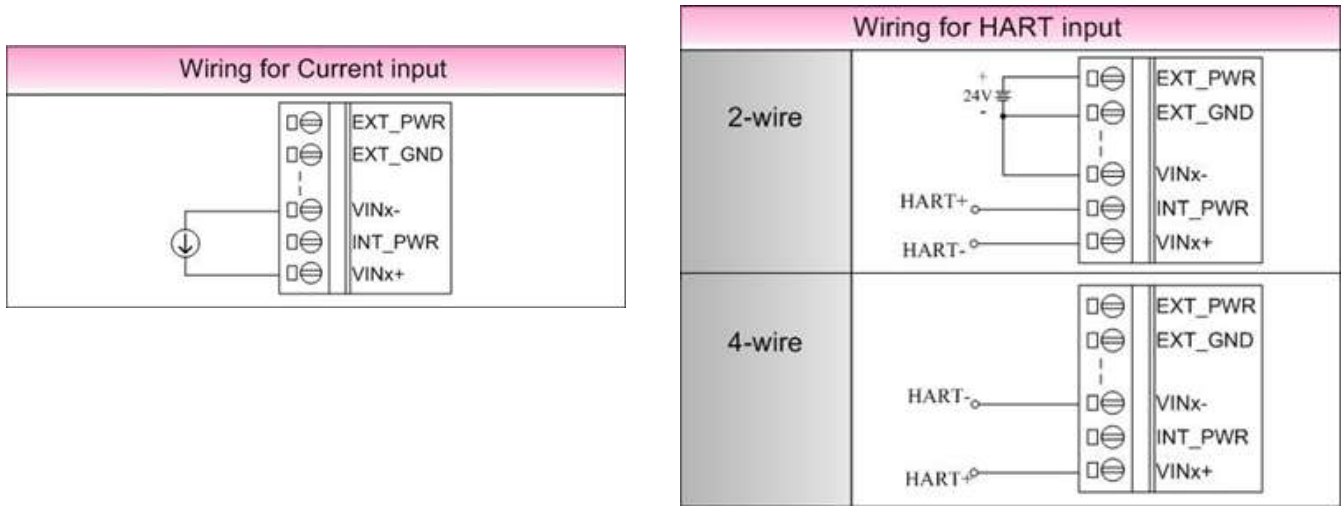
I-87H17W 接腳圖:



Pin Assignment Name	Terminal No.	Pin Assignment Name	
X	19	37	BK2
EXT_PWR	18	36	EXT_GND
VIN7-	17	35	INT_PWR7
VIN7+	16	34	EXT_GND
VIN6-	15	33	INT_PWR6
VIN6+	14	32	EXT_GND
VIN5-	13	31	INT_PWR5
VIN5+	12	30	EXT_GND
VIN4-	11	29	INT_PWR4
VIN4+	10	28	EXT_GND
VIN3-	09	27	INT_PWR3
VIN3+	08	26	EXT_GND
VIN2-	07	25	INT_PWR2
VIN2+	06	24	EXT_GND
VIN1-	05	23	INT_PWR1
VIN1+	04	22	EXT_GND
VIN0-	03	21	INT_PWR0
VIN0+	02	20	EXT_GND
BK1	01		

37-pin male D-Sub Connector

I-87H17W 接線圖:



23.2 HART 通訊 Frame 格式介紹

在開始說明範例程式之前，我們先了解 HART 通訊 Frame 的格式。HART 通訊協定是屬於主從式通訊，且採用一問一答的方式進行通訊控制與資料交換。

注意:

- A. 請參考該 HART 設備的產品說明書來了解各別 HART 設備所支援的 HART 通訊 Frame 格式。
- B. 在 Win-GRAF 程式內不需處理 “Check Byte”，I-87H17W 模組會自動處理此 “Check Byte”。
- C. HART 通訊實體層是採用 1200bps, 1 start-bit, Odd parity, 8 character-size, 1 stop-bit。

I-87H17W 發送給設備:

“Byte Count” 為 “Data” 的 Byte 數量，其值為 0 ~ 255。

Preamble	Delimiter	Address	Command	Byte Count	Data	Check byte
5 ~ 20 byte	1 byte	1 byte (short) 5 byte (long)	1 byte	1 byte	0 ~ 255 byte	1 byte

設備回覆給 I-87H17W:

“Byte Count” 為 “Response code” 加上 “Data” 的 Byte 數量，其值為 0 ~ 255。

Preamble	Delimiter	Address	Command	Byte Count	Response code	Data	Check byte
5 ~ 20 byte	1 byte	1 byte (short) 5 byte (long)	1 byte	1 byte	2 byte	0 ~ 253 byte	1 byte

Preamble: 5 ~ 20 byte

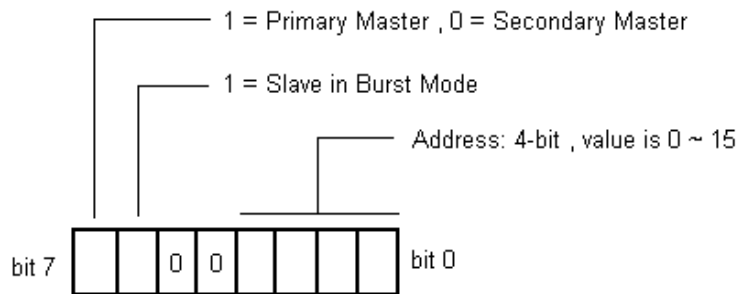
每個 Byte 的值為 255 (16#FF)。

Delimiter: 1 byte

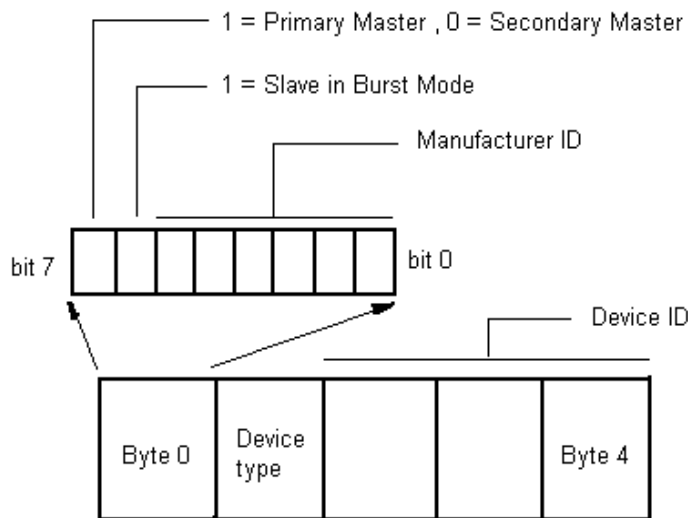
- 01 : Short frame · Burst Frame °
- 02 : Short frame · 由 Master 傳送給 Slave °
- 06 : Short frame · 由 Slave 回覆給 Master °
- 129 (16#81): Long frame · Burst Frame °
- 130 (16#82): Long frame · 由 Master 傳送給 Slave °
- 134 (16#86): Long frame · 由 Slave 回覆給 Master °

Address: 1 byte (Short frame) 或 5 byte (Long frame)

Short frame (1 byte):



Long frame (5 bytes):



Command: 1 byte · 請參考設備的定義 °

Byte Count: 1 byte °

I-87H17W 發送給設備: “Byte Count” 為 “Data” 的 Byte 數量 (0 ~ 255) °

設備回覆給 I-87H17W: “Byte Count” 為 “Response code” 加上 “Data” 的 Byte 數量 (0 ~ 255) °

Response code: 2 byte , 請參考 · 請參考設備的定義 °

Data: 請參考設備的定義 °

Check Byte: 1 byte · 在 Win-GRAF 程式內不需處理 “Check Byte” · I-87H17W 模組會自動處理此 “Check Byte” °

23.3 範例程式介紹

本章節提供了以下 3 個範例程式，來說明 Win-GRAF PAC 與 HART 設備之間的通訊方式，您可執行 Win-GRAF Workbench 並點選 "File → Add Existing Project → From Zip..." 開啟 Win-GRAF PAC 出貨光碟 (\\Napdos\Win-GRAF\demo-project) 內的 Win-GRAF 專案。

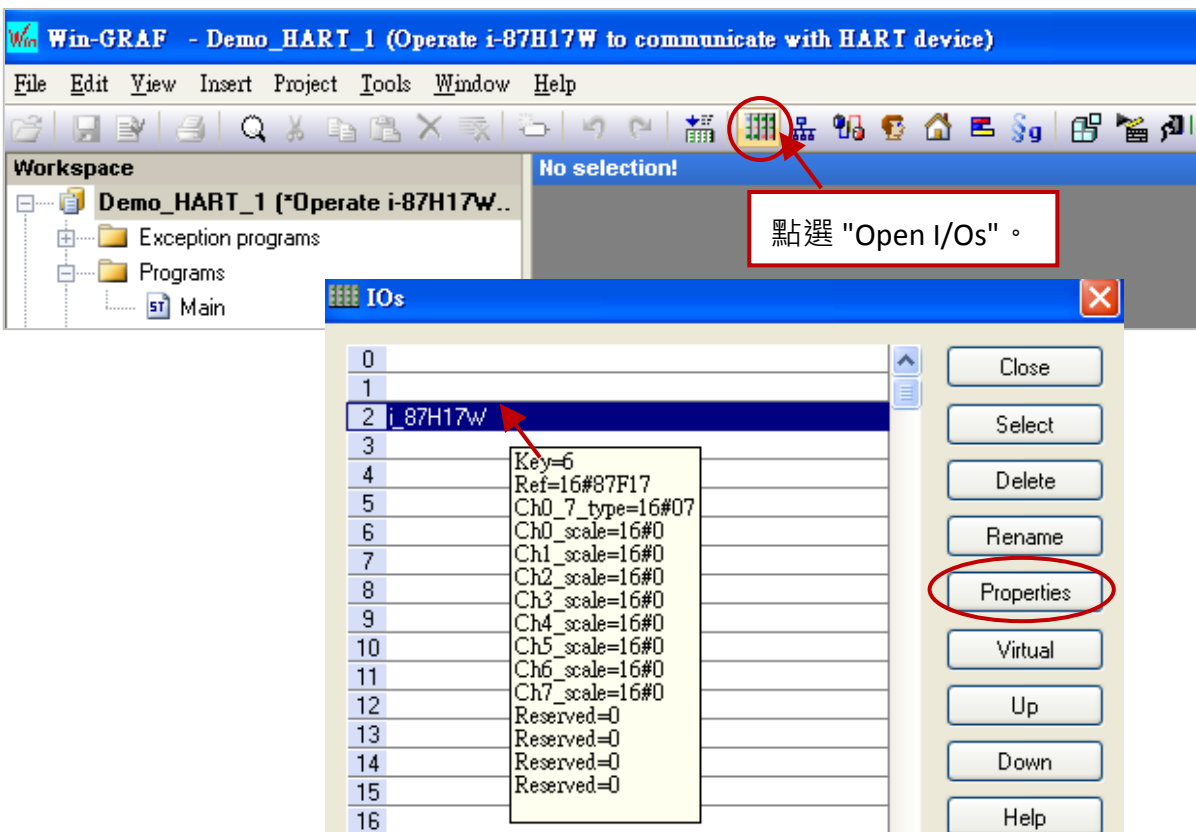
範例名稱	說明
Demo_HART_1.zip	從 Slot 2 的 通道 0 ，由 User 手動發出 HART Frame，再接收設備回覆的 Frame。
Demo_HART_2.zip	依序從 Slot 2 的 通道 0→通道 1→通道 2→通道 0 ，每隔 5 秒自動發出並接收 HART Frame。
Demo_HART_3.zip	(類似 Demo_HART_2)，但在 Slot 2, Slot 3 使用了 2 個 I-87H17W 模組，並在同一時間，依序從 通道 0~7 ，每隔 5 秒自動發出並接收 HART Frame。

注意： 同一模組一次只能處理 1 個通道發出與接收的 HART Frame 後，再處理下一個 (因為 I-87H17W 的 8 個通道是共用 1 個 HART 晶片)。

23.3.1 I/O Board 設定與 HART 函式

I/O Board 設定

"Demo_HART_1" 與 "Demo_HART_2" 範例中，都是在 Slot 2 使用一個 I-87H17W 模組，因此必須在 Win-GRAF "I/O Boards" 功能中啟用它，請點選 "Open I/Os" 按鈕，並在 Slot 2 上啟用 "i_87H17W"，點選 "Properties" 則可查看設定說明 (或可參考第四章)。



I-87H17W 是一個 8 通道的 HART 電流輸入模組 (資料型態: REAL) , 輸入範圍是 4 ~ 20mA 。

參數說明:

Ch0_scale ~ Ch7_scale : 16#SS

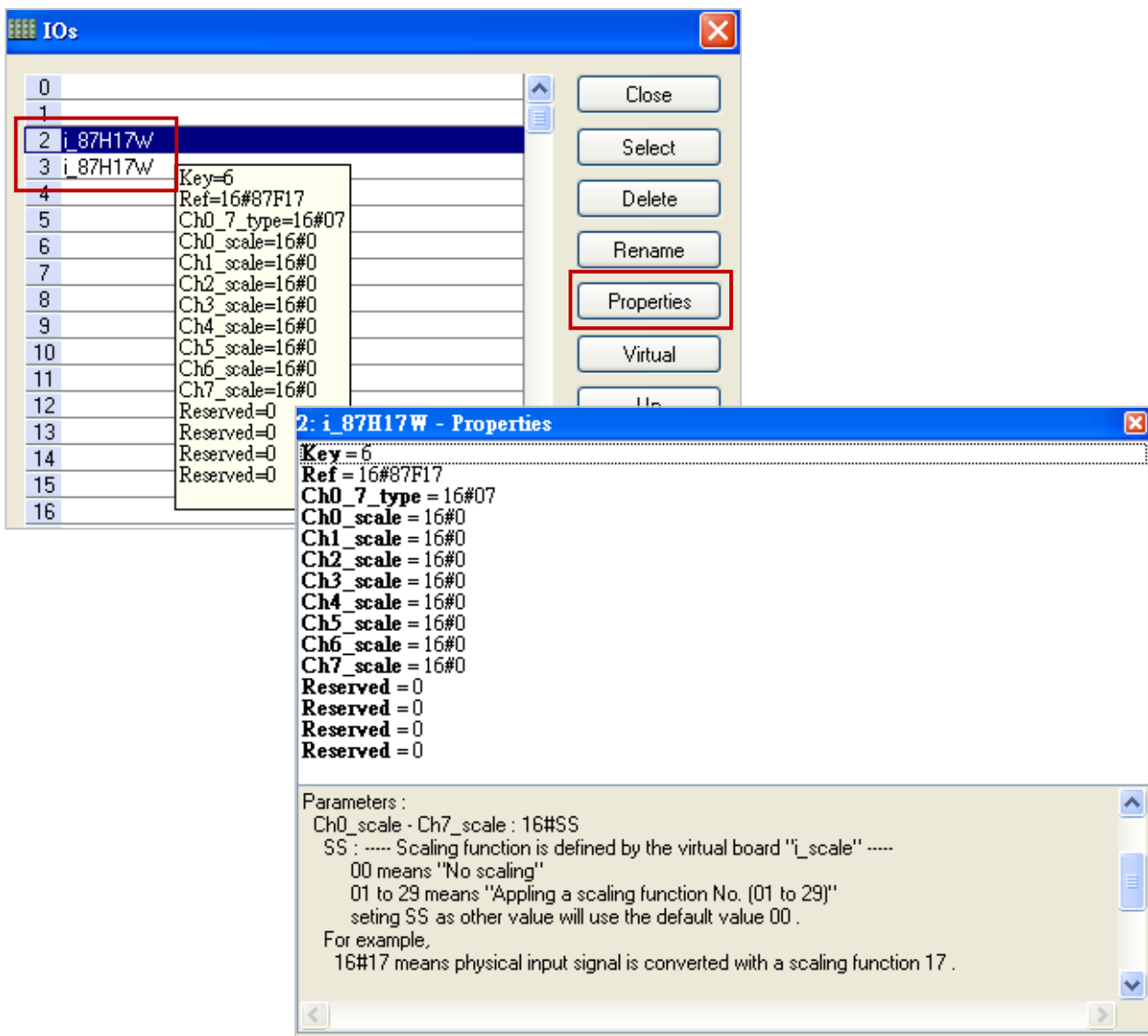
SS : 此值定義在 "i_scale" 轉換表功能中。(請參考 4.2 節)

- 00 表示不使用轉換功能。
- 01 ~ 29 表示套用轉換表 編號 01 ~ 29 的設定。
- 若設定 SS 為其它值 , 將會採用預設值 "00" 。

例如: 16#17 表示 AI 值會依據轉換表 (即 , "Ch17") 的設定 , 來轉換為一個工程值

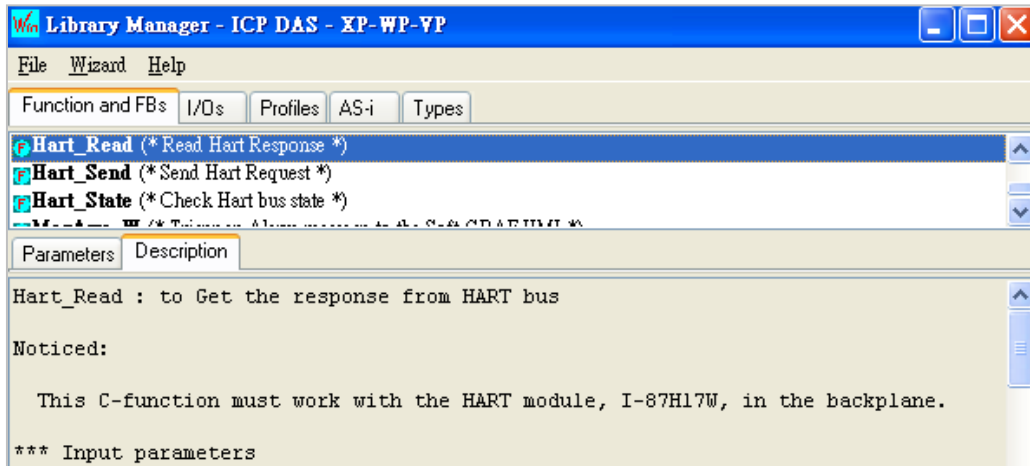
註: 若電流值大於 "9000.0" , 表示該 HART 設備可能已斷線。

在 "Demo_HART_3" 專案中 , 則是在 Slot 2, 3 啟用了 2 個 "i_87H17W" 。



HART 相關函式

您可在 ST 或 LD 程式中使用 "HART_Read"、"HART_Send" 與 "HART_Status" 功能。此外，您可在程式中點選該功能並按 "F1" 或在 "Library Manager" (參考 [1.2.3 節](#)) 查看這些說明。



Hart_Read: 取得從 HART Bus 的回應資料。

注意: 此功能需和插在 PAC 上的 HART 模組 (即，I-87H17W) 搭配使用。

輸入參數:

- gSlot:** (資料型態: DINT)
指定 I-87H17W 在 PAC 上的 Slot 編號 (0 ~ 7)，用來讀取從 HART Bus 回覆的資料。
- @gPreamble:** (資料型態: USINT) (請參考 [23.2 節](#) HART Frame 格式)。
取得 HART 設備回覆的 "Preamble" 數量。
- @gDelimiter:** (資料型態: USINT)
取得 HART 設備回覆的 "Delimiter"。
- gAddress[]:** (資料型態: USINT)
取得 HART 設備回覆的 "Address"。
- @gCommand:** (資料型態: USINT)
取得 HART 設備回覆的 "Command"。
- gData[]:** (資料型態: USINT)
取得 HART 設備回覆的 "Data"。
- @gDataLen:** (資料型態: USINT)
取得 HART 設備回覆的資料長度 (Byte)。

輸出參數:

- Q:** (資料型態: DINT)
 - 0: 尚無任何操作。
 - 1: 讀取成功。
 - 1: 於指定的 Slot 上，找不到 HART 模組 (I-87H17W)。
 - 2: 無效的 HART 回覆資料。
 - 3: 等待 HART 回覆資料。
 - 4: 無 HART 命令。
 - 5: HART 模組 (I-87H17W) 離線。

hart_Send: 傳送命令至 HART Bus。

注意: 此功能需和插在 PAC 上的 HART 模組 (即, I-87H17W) 搭配使用。

輸入參數:

- gSlot:** (資料型態: DINT)
指定 I-87H17W 在 PAC 上的 Slot 編號 (0 ~ 7), 用來傳送 HART 命令。
- gChannel:** (資料型態: DINT)
指定 I-87H17W 的通道編號 (0 ~ 7), 用來傳送 HART 命令到 HART 設備。
- gPreamble:** (資料型態: USINT) (請參考 [23.2 節](#) HART Frame 格式)。
發送 HART 命令的 “Preamble” 數量 (5 ~ 20 Bytes)。
- gDelimiter:** (資料型態: USINT)
發送 HART 命令的 “Delimiter”。
- gAddress[]:** (資料型態: USINT)
發送 HART 命令的 “Address”。
- gCommand:** (資料型態: USINT)
發送 HART 命令的 “Command”。
- gData[]:** (資料型態: USINT)
發送 HART 命令的 “Data”。
- gDataLen:** (資料型態: USINT)
發送 HART 命令的 “Data” 長度 (Byte)。

輸出參數:

- Q:** (資料型態: DINT)
- 0: 尚無任何操作。
 - 1: 讀取成功。
 - 1: 於指定的 Slot 上, 找不到 HART 模組 (I-87H17W)。
 - 2: 指定的 I-87H17W 通道 (Channel) 是錯誤的。
 - 3: 指定的 “Preamble” 數量是錯誤的。
 - 4: HART 命令傳送錯誤, 請檢查 “Delimiter” 與 “Address” 是正確的。
 - 5: HART Bus 正忙碌中。
 - 6: HART 模組 (I-87H17W) 離線。

Hart_State: 取得 HART Bus 的狀態

注意: 此功能需和插在 PAC 上的 HART 模組 (即, I-87H17W) 搭配使用。

輸入參數:

gSlot: (資料型態: DINT)

指定 I-87H17W 在 PAC 上的 Slot 編號 (0 ~ 7), 用來取得 HART Bus 的狀態。

輸出參數:

Q: (資料型態: DINT)

0: 尚無任何操作。

1: HART 命令正等待要傳送給 HART 設備。

2: HART 命令已傳送至 HART 設備。

3: 等待 HART 回覆資料。

4: 已接收 HART 回覆資料。

-1: 逾時。

-2: HART Frame 回覆的長度太短 (請參考 [23.2 節](#) HART Frame 格式)。

-3: HART Frame 回覆的 "Delimiter" 是錯誤的。

-4: HART Frame 回覆的 "Master" 位址是錯誤的。

-5: HART Frame 回覆的 "Burst" 位址是錯誤的。

-6: HART Frame 回覆的 "Command" 是錯誤的。

-7: HART Frame 回覆的檢查碼 (checksum) 是錯誤的。

-8: HART Frame 回覆錯誤。

-98: 發送與回覆 HART Frame, 出現參數不符。

-99: 不可測的錯誤。

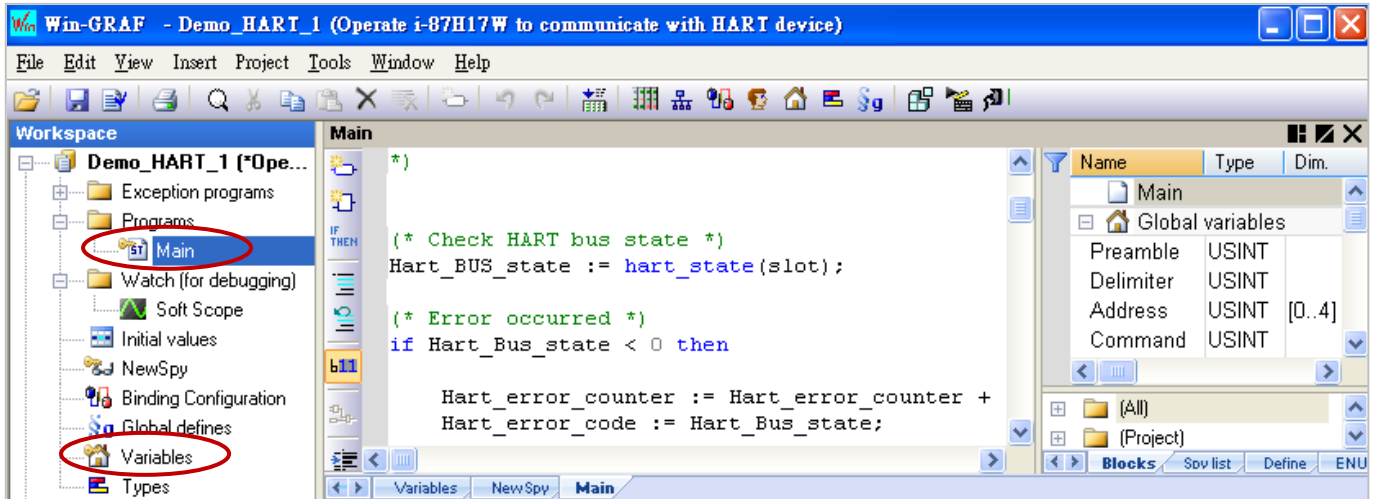
-100: 傳送 HART 命令至 HART 期間發生錯誤。

-101: 於指定的 Slot 上, 找不到 HART 模組 (I-87H17W)。

-102: HART 模組 (I-87H17W) 離線。

23.3.2 範例程式 - “Demo_HART_1”

(可見下方 ST 程式的功能說明。)



變數說明:

變數名稱	資料型態	Dim.	說明
slot	DINT	-	傳送 HART Frame 所需使用的變數。 (slot 的初始值 = 2) (參考 23.2 節 HART Frame 格式與 23.3.1 節 "HART_Send" 函式)
channel	DINT	-	
Preamble	USINT	-	
Delimiter	USINT	-	
Address	USINT	[0..4]	
Command	USINT	-	
Data	USINT	[0..254]	
Datalen	USINT	-	發送/回覆 HART Frame 的狀態值。 回覆 HART Frame 所需使用的變數。 (參考 23.2 節 HART Frame 格式與 23.3.1 節 "HART_Read" 函式)
ret	DINT	-	
Preamble_r	USINT	-	
Delimiter_r	USINT	-	
Address_r	USINT	[0..4]	
Command_r	USINT	-	
Data_r	USINT	[0..254]	
Datalen_r	USINT	-	取得 HART Bus 的狀態。 累計 HART Bus 的錯誤次數。 HART Bus 的錯誤回傳值。 HART Bus 的錯誤訊息。 發送 HART Frame 時的錯誤訊息。 讀取 HART Frame 時的錯誤訊息。 設定為 “TRUE” 以發送 HART Frame。 設定為 “TRUE” 以讀取 HART Frame。
Hart_BUS_state	DINT	-	
Hart_error_counter	DINT	-	
Hart_error_code	DINT	-	
hart_bus_err_Msg	STRING(255)	-	
hart_send_err_msg	STRING(255)	-	
hart_read_err_msg	STRING(255)	-	
send_HART_request	BOOL	-	
read_HART_request	BOOL	-	

ST 程式 (Main)

(* 此範例程式將說明如何使用一個 I-87H17W (HART Master) 並手動地與一個 HART 設備進行通訊。)

硬體環境:

1. 在 Win-GRAF PAC 的 Slot2 插上一個 I-87H17W 模組。
2. 在 I-87H17W 的通道 0 接上一個 HART 設備。 *)

(* 檢查 HART Bus 狀態 *)

```
Hart_BUS_state := hart_state(slot);
```

(* HART Bus 發生錯誤的訊息與次數 *)

```
if Hart_Bus_state < 0 then
```

```
    Hart_error_counter := Hart_error_counter + 1;
```

```
    Hart_error_code := Hart_Bus_state;
```

```
    case Hart_error_code of
```

```
        -1 : hart_bus_err_Msg := 'Error: Timeout' ;
```

```
        -2 : hart_bus_err_Msg := 'Error: Read_data_too_short error' ;
```

```
        -3 : hart_bus_err_Msg := 'Error: Response Delimiter error' ;
```

```
        -4 : hart_bus_err_Msg := 'Error: Response addr_master error' ;
```

```
        -5 : hart_bus_err_Msg := 'Error: Response addr_burst error' ;
```

```
        -6 : hart_bus_err_Msg := 'Error: Response rcv_command error' ;
```

```
        -7 : hart_bus_err_Msg := 'Error: Response checksum error' ;
```

```
        -8 : hart_bus_err_Msg := 'Error: response error' ;
```

```
        -98 : hart_bus_err_Msg := 'Error: para_mismatch error' ;
```

```
        -99 : hart_bus_err_Msg := 'Error: impossible error' ;
```

```
        -100 : hart_bus_err_Msg := 'Error: Sending HART request error' ;
```

```
        -101 : hart_bus_err_Msg := 'Error: cannot find the HART moudule, i-87H17W' ;
```

```
        -102 : hart_bus_err_Msg := 'Error: i-87H17W is off-line' ;
```

```
    end_case;
```

```
end_if;
```

```
if Hart_BUS_state = 0 and Send_HART_request then
```

(* 表示 HART Bus 目前閒置中，可進行傳送 *)

(* 建立 HART 命令 *)

```
slot := 2;
```

```
channel := 0;
```

```
Preamble := 5;
```

```
Delimiter := 16#82;
```

```
Address[0] := 16#96;
```

```
Address[1] := 16#85;
```

```
Address[2] := 16#0B;  
Address[3] := 16#0A;  
Address[4] := 16#42;  
Command := 16#03;  
Datalen := 0;
```

```
ret := hart_send(slot, channel, Preamble, Delimiter, Address, Command, Data, Datalen);
```

```
case ret of
```

```
1: Send_HART_request := false;
```

```
    hart_send_err_msg := 'Send success';
```

```
-1: hart_send_err_msg := 'Send Error: cannot find the HART module, i-87H17W';
```

```
-2: hart_send_err_msg := 'Send Error: specified the wrong channel to send hart cmd';
```

```
-3: hart_send_err_msg := 'Send Error: specified the wrong preamble number';
```

```
-4: hart_send_err_msg := 'Send Error: wrong HART cmd, please check delimiter and  
    address is correct';
```

```
-5: hart_send_err_msg := 'Send Error: HART bus is busy';
```

```
-6: hart_send_err_msg := 'Send Error: i-87H17W is off-line';
```

```
end_case;
```

```
end_if;
```

```
(* 讀取 HART 回覆資料 *)
```

```
if Hart_BUS_state = 4 then
```

```
ret := hart_read(slot, Preamble_r, Delimiter_r, Address_r, Command_r, Data_r, Datalen_r);
```

```
case ret of
```

```
1: hart_read_err_msg := 'Read success';
```

```
    read_HART_request := true;
```

```
-1: hart_read_err_msg := 'Read Error: cannot find the HART module, i-87H17W';
```

```
-2: hart_read_err_msg := 'Read Error: invalid HART response';
```

```
-3: hart_read_err_msg := 'Read Error: waiting response';
```

```
-4: hart_read_err_msg := 'Read Error: No Request';
```

```
-5: hart_read_err_msg := 'Read Error: i-87H17W is off-line';
```

```
end_case;
```

```
end_if;
```

```
(* 處理 HART 回覆資料 *)
```

```
if read_HART_request then
```

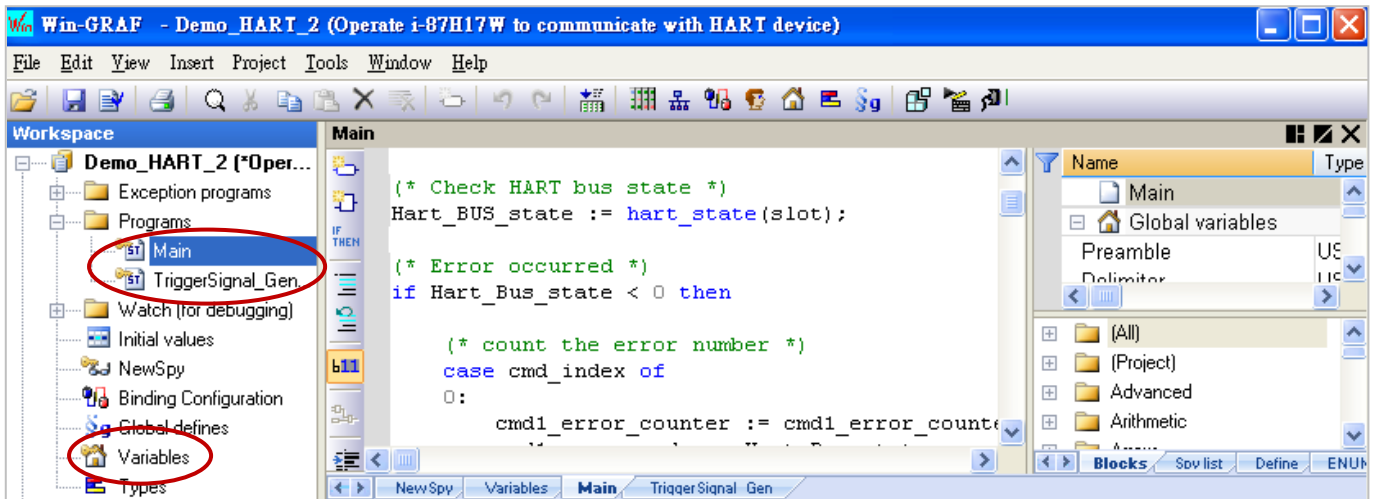
```
    read_HART_request := false;
```

```
(* ToDo: 將處理 HART 回覆資料的程式，放在此處 *)
```

```
end_if;
```

23.3.3 範例程式 - “Demo_HART_2”

(可見下方 ST 程式的功能說明。)



變數說明:

變數名稱	資料型態	Dim.	說明
slot	DINT	-	傳送 HART Frame 所需使用的變數。 (slot 的初始值 = 2) (參考 23.2 節 HART Frame 格式與 23.3.1 節 "HART_Send" 函式)
channel	DINT	-	
Preamble	USINT	-	
Delimiter	USINT	-	
Address	USINT	[0..4]	
Command	USINT	-	
Data	USINT	[0..254]	
Datalen	USINT	-	發送/回覆 HART Frame 的狀態值。
ret	DINT	-	
Preamble_r	USINT	-	
Delimiter_r	USINT	-	
Address_r	USINT	[0..4]	
Command_r	USINT	-	
Data_r	USINT	[0..254]	
Datalen_r	USINT	-	回覆 HART Frame 所需使用的變數。 (參考 23.2 節 HART Frame 格式與 23.3.1 節 "HART_Send" 函式)
Hart_BUS_state	DINT	-	
hart_bus_err_Msg	STRING(255)	-	
hart_send_err_msg	STRING(255)	-	
hart_read_err_msg	STRING(255)	-	
send_HART_request	BOOL	-	
read_HART_request	BOOL	-	
cmd_index	DINT	-	處理命令的索引值。

變數名稱	資料型態	Dim.	說明
cmd_max_num	DINT	-	發送命令的數目 (初始值 = 3)。
cmd1_response_counter	DINT	-	回覆命令 的次數。 (由通道 0 / 1/ 2)
cmd2_response_counter	DINT	-	
cmd3_response_counter	DINT	-	
cmd1_error_counter	DINT	-	HART Bus 的錯誤計數。 (由通道 0 / 1/ 2)
cmd2_error_counter	DINT	-	
cmd3_error_counter	DINT	-	
cmd1_error_code	DINT	-	HART Bus 的錯誤回傳值。 (由通道 0 / 1/ 2)
cmd2_error_code	DINT	-	
cmd3_error_code	DINT	-	
trigger_Timer	TIME	-	啟動計時。
trigger_interval	TIME	-	發出命令的時間間隔 (初始值 = T#5s)。
INIT	BOOL	-	初始化 (初始值 = TRUE)。
next_index	DINT	-	回應資料值的位置索引。
cmd1_Data	REAL	[0..3]	存放回覆的資料。 (由通道 0 / 1/ 2)
cmd2_Data	REAL	[0..3]	
cmd3_Data	REAL	[0..3]	

ST 程式 (Main)

(* 此範例程式將說明如何使用一個 I-87H17W (HART Master) 與 連接在 I-87H17W 多個通道上的 HART 設備自動地進行通訊。)

硬體環境:

1. 在 Win-GRAF PAC 的 Slot2 插上一個 I-87H17W 模組。
2. 在此 I-87H17W 的通道 0 ~ 2 各接上一個 HART 設備。 *)

(* 檢查 HART Bus 狀態 *)

```
Hart_BUS_state := hart_state(slot);
```

(* 當錯誤發生 *)

```
if Hart_Bus_state < 0 then
```

(* 計數錯誤的次數 *)

```
case cmd_index of
```

```
0:
```

```
cmd1_error_counter := cmd1_error_counter + 1;
```

```
cmd1_error_code := Hart_Bus_state;
```

```
1:
```

```
cmd2_error_counter := cmd2_error_counter + 1;
```

```
cmd2_error_code := Hart_Bus_state;
```

```
2:
  cmd3_error_counter := cmd3_error_counter + 1;
  cmd3_error_code := Hart_Bus_state;
end_case;
```

(* 嘗試傳送下一個 HART 命令 *)

```
cmd_index := cmd_index + 1;
```

```
if cmd_index = cmd_max_num then
  cmd_index := 0;
end_if;
```

```
case Hart_Bus_state of
-1 : hart_bus_err_Msg := 'Error: Timeout' ;
-2 : hart_bus_err_Msg := 'Error: Read_data_too_short error' ;
-3 : hart_bus_err_Msg := 'Error: Response Delimiter error' ;
-4 : hart_bus_err_Msg := 'Error: Response addr_master error' ;
-5 : hart_bus_err_Msg := 'Error: Response addr_burst error' ;
-6 : hart_bus_err_Msg := 'Error: Response recv_command error' ;
-7 : hart_bus_err_Msg := 'Error: Response checksum error' ;
-8 : hart_bus_err_Msg := 'Error: response error' ;
-98 : hart_bus_err_Msg := 'Error: para_mismatch error' ;
-99 : hart_bus_err_Msg := 'Error: impossible error' ;
-100 : hart_bus_err_Msg := 'Error: Sending HART request error' ;
-101 : hart_bus_err_Msg := 'Error: cannot find the HART moudule, i-87H17W' ;
-102 : hart_bus_err_Msg := 'Error: i-87H17W is off-line';
end_case;
```

```
end_if;
```

```
if Hart_BUS_state = 0 and Send_HART_request then
```

(* 表示 HART Bus 目前閒置中，可進行傳送 *)

(* 建立 HART 命令 *)

```
case cmd_index of
0:
  slot := 2;
  channel := 0;
  Preamble := 5;
  Delimiter := 16#82;
  Address[0] := 16#96;
  Address[1] := 16#85;
  Address[2] := 16#0B;
  Address[3] := 16#0A;
  Address[4] := 16#42;
  Command := 16#03;
  Datalen := 16#00;
```

```

1:
  slot := 2;
  channel := 1;
  Preamble := 5;
  Delimiter := 16#82;
  Address[0] := 16#96;
  Address[1] := 16#85;
  Address[2] := 16#0B;
  Address[3] := 16#0A;
  Address[4] := 16#42;
  Command := 16#03;
  Datalen := 16#00;
2:
  slot := 2;
  channel := 2;
  Preamble := 5;
  Delimiter := 16#82;
  Address[0] := 16#96;
  Address[1] := 16#85;
  Address[2] := 16#0B;
  Address[3] := 16#0A;
  Address[4] := 16#42;
  Command := 16#03;
  Datalen := 16#00;
end_case;

ret := hart_send(slot, channel, Preamble, Delimiter, Address, Command, Data, Datalen);

case ret of
  1: Send_HART_request := false;
     hart_send_err_msg := 'Send success';
-1: hart_send_err_msg := 'Send Error: cannot find the HART module, i-87H17W';
-2: hart_send_err_msg := 'Send Error: specified the wrong channel to send hart cmd';
-3: hart_send_err_msg := 'Send Error: specified the wrong preamble number';
-4: hart_send_err_msg := 'Send Error: wrong HART cmd, please check delimiter and address is
    correct';
-5: hart_send_err_msg := 'Send Error: HART bus is busy';
-6: hart_send_err_msg := 'Send Error: i-87H17W is off-line';
end_case;

end_if;

```

(* 讀取 HART 回覆資料 *)

```

if Hart_BUS_state = 4 then

  ret := hart_read(slot, Preamble_r, Delimiter_r, Address_r, Command_r, Data_r, Datalen_r);

  case ret of
    1: hart_read_err_msg := 'Read success';
       read_HART_request := true;

```

```

-1: hart_read_err_msg := 'Read Error: cannot find the HART moudule, i-87H17W';
-2: hart_read_err_msg := 'Read Error: invalid HART response';
-3: hart_read_err_msg := 'Read Error: waiting response';
-4: hart_read_err_msg := 'Read Error: No Request';
-5: hart_read_err_msg := 'Read Error: i-87H17W is off-line';
end_case;

```

```
end_if;
```

(* 處理 HART 回覆資料 *)

```

if read_HART_request then
  read_HART_request := false;

```

(* --- 請使用符合您 HART 設備的 <Data> 定義，來修改程式 ---

假設回應的 HART 資料為 00 00 41 A0 FF 3E 0C 3E C5 37 48 20 41 C8 3F 22 39 42 C9 8E D1

此資料是由 HART 設備所回覆，並採用大端排序 (big endian)，其意義如下：

41 A0 FF 3E = 41A0FF3E < 轉換此 4 byte (IEEE-754) 為一個 REAL 值 > : 20.1246 mA

3E C5 37 48 = 3EC53748 < 轉換此 4 byte (IEEE-754) 為一個 REAL 值 > : 0.385187

41 C8 3F 22 = 41C83F22 < 轉換此 4 byte (IEEE-754) 為一個 REAL 值 > : 25.0308

42 C9 8E D1 = 42C98ED1 < 轉換此 4 byte (IEEE-754) 為一個 REAL 值 > : 100.779 *)

```
case cmd_index of
```

```
0:
```

```
  cmd1_response_counter := cmd1_response_counter + 1;
```

```
  next_index := SerializeIn(Data_r, cmd1_Data[0], 2, true);
```

```
  next_index := next_index + 1;
```

```
  next_index := SerializeIn(Data_r, cmd1_Data[1], next_index, true);
```

```
  next_index := next_index + 1;
```

```
  next_index := SerializeIn(Data_r, cmd1_Data[2], next_index, true);
```

```
  next_index := next_index + 1;
```

```
  next_index := SerializeIn(Data_r, cmd1_Data[3], next_index, true);
```

```
1:
```

```
  cmd2_response_counter := cmd2_response_counter + 1;
```

```
  next_index := SerializeIn(Data_r, cmd2_Data[0], 2, true);
```

```
  next_index := next_index + 1;
```

```
  next_index := SerializeIn(Data_r, cmd2_Data[1], next_index, true);
```

```
  next_index := next_index + 1;
```

```
  next_index := SerializeIn(Data_r, cmd2_Data[2], next_index, true);
```

```
  next_index := next_index + 1;
```

```
  next_index := SerializeIn(Data_r, cmd2_Data[3], next_index, true);
```

```
2:
```

```
  cmd3_response_counter := cmd3_response_counter + 1;
```

```
  next_index := SerializeIn(Data_r, cmd3_Data[0], 2, true);
```

```
  next_index := next_index + 1;
```

```
next_index := Serializeln(Data_r, cmd3_Data[1], next_index, true);
next_index := next_index + 1;
next_index := Serializeln(Data_r, cmd3_Data[2], next_index, true);
next_index := next_index + 1;
next_index := Serializeln(Data_r, cmd3_Data[3], next_index, true);
end_case;
```

(* 嘗試傳送下一個 HART 命令 *)

```
cmd_index := cmd_index + 1;
Send_HART_request := true;

if cmd_index = cmd_max_num then
  cmd_index := 0;
end_if;

end_if;
```

ST 程式 (TriggerSignal)

(*
啟動計時器來產生一個觸發信號，來傳送 HART 命令
*)

```
if INIT then
  INIT := false;
  TStart(trigger_timer);
  send_HART_request := true;
end_if;

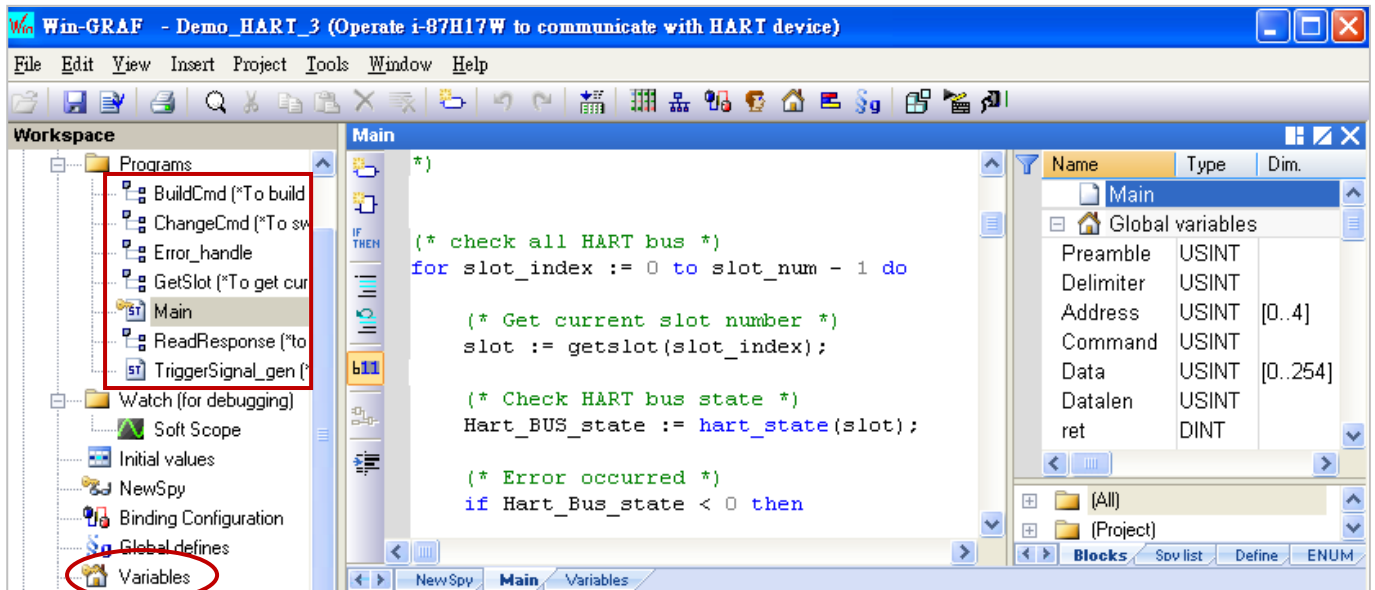
if trigger_Timer > trigger_interval then

  trigger_Timer := T#0s;
  send_HART_request := true;

end_if;
```


23.3.4 範例程式 - “Demo_HART_3”

(可見下方 ST 程式的功能說明。)



變數說明:

此範例程式與 “Demo_HART_2” 類似，您可參考其變數說明或在 Win-GRAF 點選 “Variables” 來查看所有使用的變數。

ST 程式 (Main)

此範例使用了多個子程式，您可在 Win-GRAF 中一點選來看，此處將針對 “Main” 程式來說明。

(* 此範例程式將說明如何使用 2 個 I-87H17W (HART Master) 與連接在 I-87H17W 多個通道上的 HART 設備自動地進行通訊。)

硬體環境:

1. 在 Win-GRAF PAC 的 Slot2 插上一個 I-87H17W 模組。
2. 在 Win-GRAF PAC 的 Slot3 插上一個 I-87H17W 模組。
3. 在第 1 個 I-87H17W 的通道 0~7 各接上一個 HART 設備。
4. 在第 2 個 I-87H17W 的通道 0~7 各接上一個 HART 設備。 *)

(* 檢查所有 HART Bus *)

```
for slot_index := 0 to slot_num - 1 do
```

(* 取得目前 Slot 編號 *)

```
slot := getslot(slot_index);
```

(* 檢查 HART Bus 狀態 *)

```
Hart_BUS_state := hart_state(slot);
```

(* 當錯誤發生 *)

```
if Hart_Bus_state < 0 then
```

(* 計數錯誤的次數 *)

Error_handle(any_to_byte(slot_index));

(* 嘗試傳送下一個 HART 命令 *)

ChangeCmd(any_to_byte(slot_index));

case Hart_Bus_state of

-1 : hart_bus_err_Msg := 'Error: Timeout' ;
-2 : hart_bus_err_Msg := 'Error: Read_data_too_short error' ;
-3 : hart_bus_err_Msg := 'Error: Response Delimiter error' ;
-4 : hart_bus_err_Msg := 'Error: Response addr_master error' ;
-5 : hart_bus_err_Msg := 'Error: Response addr_burst error' ;
-6 : hart_bus_err_Msg := 'Error: Response recv_command error' ;
-7 : hart_bus_err_Msg := 'Error: Response checksum error' ;
-8 : hart_bus_err_Msg := 'Error: response error' ;
-98 : hart_bus_err_Msg := 'Error: para_mismatch error' ;
-99 : hart_bus_err_Msg := 'Error: impossible error' ;
-100 : hart_bus_err_Msg := 'Error: Sending HART request error' ;
-101 : hart_bus_err_Msg := 'Error: cannot find the HART module, i-87H17W' ;
-102 : hart_bus_err_Msg := 'Error: i-87H17W is off-line';
end_case;

end_if;

if Hart_BUS_state = 0 and Send_HART_request[slot_index] then

(* 表示 HART Bus 目前閒置中，可進行傳送 *)

(* 建立 HART 命令 *)

buildCmd(any_to_byte(slot_index));

ret := hart_send(slot, channel, Preamble, Delimiter, Address, Command, Data, Datalen);

case ret of

1: Send_HART_request[slot_index] := false;
hart_send_err_msg := 'Send success';
-1: hart_send_err_msg := 'Send Error: cannot find the HART module, i-87H17W';
-2: hart_send_err_msg := 'Send Error: specified the wrong channel to send hart cmd';
-3: hart_send_err_msg := 'Send Error: specified the wrong preamble number';
-4: hart_send_err_msg := 'Send Error: wrong HART cmd, please check delimiter and address is correct';
-5: hart_send_err_msg := 'Send Error: HART bus is busy';
-6: hart_send_err_msg := 'Send Error: i-87H17W is off-line';
end_case;

end_if;

(* 讀取 HART 回覆資料 *)

if Hart_BUS_state = 4 then

ret := hart_read(slot, Preamble_r, Delimiter_r, Address_r, Command_r, Data_r, Datalen_r);

```

case ret of
  1: hart_read_err_msg := 'Read success';
    read_HART_request := true;
-1: hart_read_err_msg := 'Read Error: cannot find the HART module, i-87H17W';
-2: hart_read_err_msg := 'Read Error: invalid HART response';
-3: hart_read_err_msg := 'Read Error: waiting response';
-4: hart_read_err_msg := 'Read Error: No Request';
-5: hart_read_err_msg := 'Read Error: i-87H17W is off-line';
end_case;

```

```
end_if;
```

(* 處理 HART 回覆資料 *)

```
if read_HART_request then
  read_HART_request := false;
```

```
  ReadResponse(any_to_byte(slot_index));
  ChangeCmd(any_to_byte(slot_index));
```

```
end_if;
end_for;
```

23.4 測試範例程式

功能說明:

範例名稱	說明
Demo_HART_1.zip	從 Slot 2 的 通道 0 ，由 User 手動發出 HART Frame，再接收設備回覆的 Frame。
Demo_HART_2.zip	依序從 Slot 2 的 通道 0→通道 1→通道 2→通道 0 ，每隔 5 秒自動發出並接收 HART Frame。
Demo_HART_3.zip	(類似 Demo_HART_2)，但在 Slot 2, Slot 3 使用了 2 個 I-87H17W 模組，並在同一時間，依序從 通道 0 ~ 7 ，每隔 5 秒自動發出並接收 HART Frame。

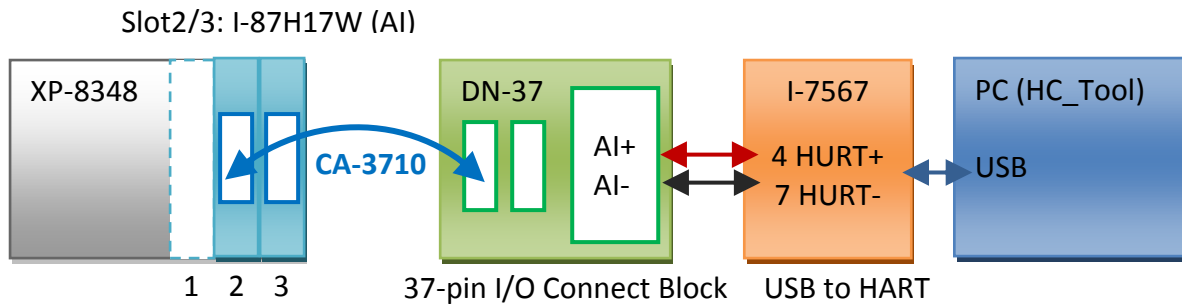
注意: 同一模組一次只能處理 1 個通道發出與接收的 HART Frame 後，再處理下一個 (因為 I-87H17W 的 8 個通道是共用 1 個 HART 晶片)。

23.4.1 測試環境與“HC_Tool”設定

測試前，請確認您的 Win-GRAF PAC 的 I/O Slot 2 (或 3) 已插上 I-87H17W 模組，且要使用的 I/O 通道 (Ch0 ~ 7) 已連接到 HART 設備 (參考 [23.1 節](#))，再將 PAC 開機，並下載範例程式到 Win-GRAF PAC。

以下會依序說明每個範例程式的測試方式，並會使用“HC_Tool”來模擬 HART 設備進行資料回覆，您可在 I-7567 產品網頁 (如下表) 下載此工具軟體與使用手冊。

您可參考各產品網頁 (Pin Assignment 或 Wire Connection) 來查看詳細的接線方式。

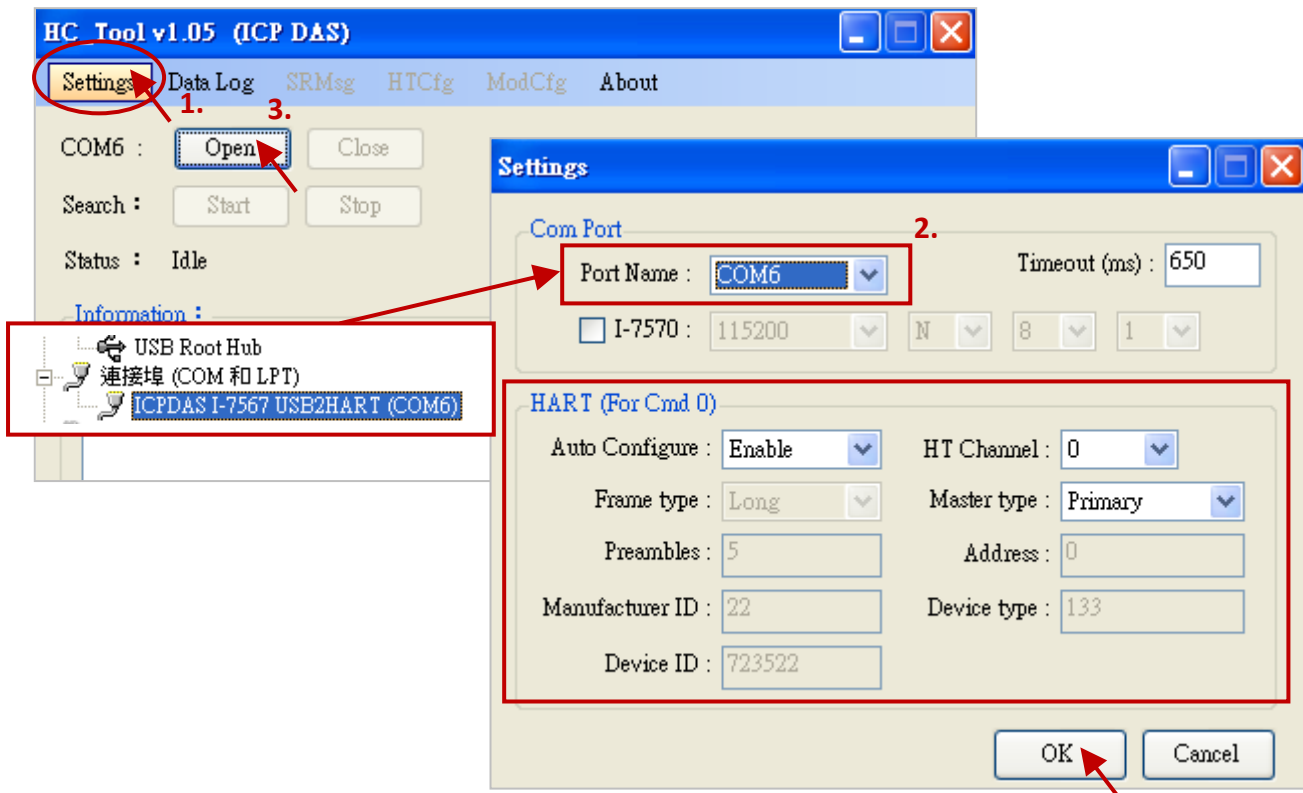


測試環境:

設備	型號	數量	備註
Win-GRAF PAC	XP-8348-CE6	1	-
HART AI 模組	I-87H17W	2	“Demo_HART_1” 與 “Demo_HART_2” 只需使用一個。
USB 至 HART 轉換器	I-7567	1	請至網頁下載驅動程式 與 工具軟體 (HC_Tool)。
Web: www.icpdas.com > 商品目錄 > 工業級通訊產品 > Fieldbus Solutions > HART Series			
I/O 端子板	DN-37	1	含 CA-3710 連接線。
Web: www.icpdas.com > 商品目錄 > PC 介面卡 > PCI I/O 介面卡 > 端子板			

HC_Tool 設定:

執行 HC_Tool.exe 並點選 “Setting” 設定 PC 端使用的 COM Port 編號 (可見 “裝置管理員”)，其它欄位請設定如下圖 (或 [參考其手冊](#))，按 “OK” 後，請再點選 “Open” 開啟此 COM Port。

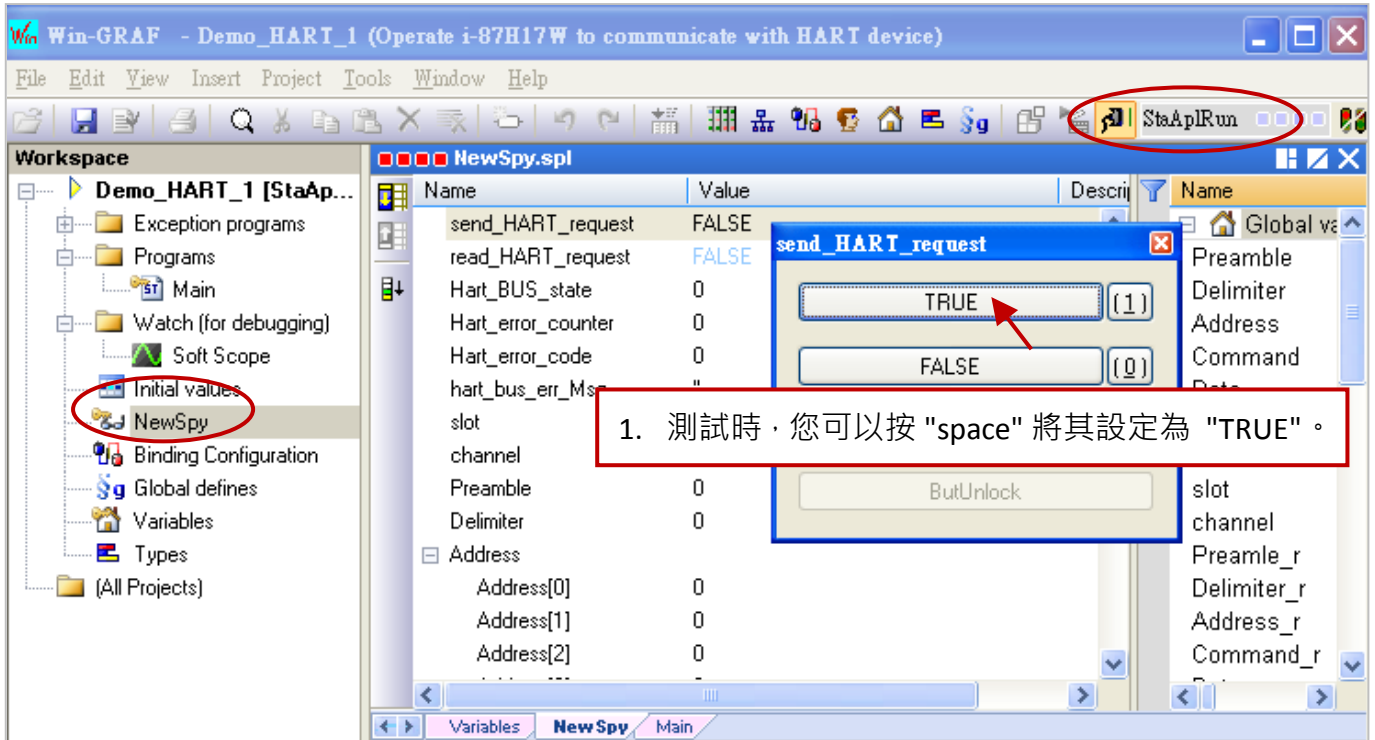


23.4.2 測試 "Demo_HART_1" , "Demo_HART_2" 與 "Demo_HART_3"

功能說明: (程式- [Demo HART 1](#) ; 接線說明 – [參考 23.4.1 節](#))

從 I-87H17W 的通道 0 (位於 Slot 2) , 由 User 手動發出 HART Frame , 再接收設備回覆的 Frame 。

下載後 , 請開啟觀測清單 (NewSpy) 並將 "send_HART_request" 設定為 "TRUE" 來傳送 HART 命令 。



在 HC_Tool 中 , 點選 "SRMsg" 開啟 "Send & Receive Msg" 視窗 , 並輸入需回覆的資料 , 再按 "Send" 。



Preamble	Delimiter	Address	Command	Byte Count	Data
I-87H17W 發出 HART 命令 (16 進制)					
FF FF FF FF FF	82	(long) 96 85 0B 0A 42	03	00	-
設備回覆的資料 (16 進制)					
FF FF FF FF FF	86	(long) 96 85 0B 0A 42	03	15 ₍₁₆₎	00 00 41 A0 FF 3E 0C 3E C5 37 48 20 41 C8 3F 22 39 42 C9 8E D1

若回覆成功，您可見到如下畫面。

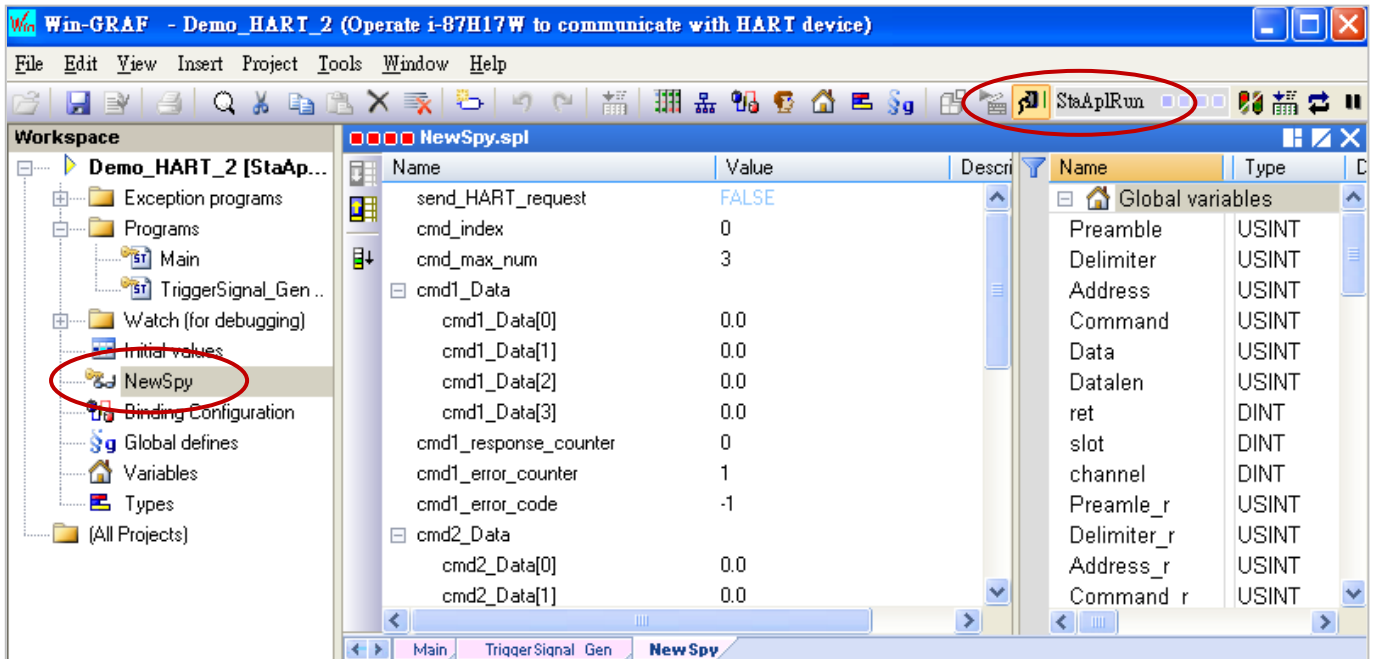
The screenshot shows the Win-GRAB interface with the following variable values:

Name	Value	Description
send_HART_request	FALSE	
read_HART_request	FALSE	
Hart_BUS_state	0	
Hart_error_counter	0	
Hart_error_code	0	
hart_bus_err_Msg	"	
slot	2	
channel	0	
Preamble	5	
Delimiter	130	
Address		
Address[0]	150	
Address[1]	133	
Address[2]	11	
Address[3]	10	
Address[4]	66	
Command	3	
Data		
Datalen	0	
hart_send_err_msg	'Send success'	
Preamble_r	5	
Delimiter_r	16#86	
Address_r		
Address_r[0]	150	
Address_r[1]	133	
Address_r[2]	11	
Address_r[3]	10	
Address_r[4]	66	
Command_r	3	
Data_r		
Data_r[0]	0	
Data_r[1]	0	
Data_r[2]	65	
Data_r[3]	160	
Data_r[250]	0	
Data_r[251]	0	
Data_r[252]	0	
Data_r[253]	0	
Data_r[254]	0	
Datalen_r	16#15	
hart_read_err_msg	'Read success'	

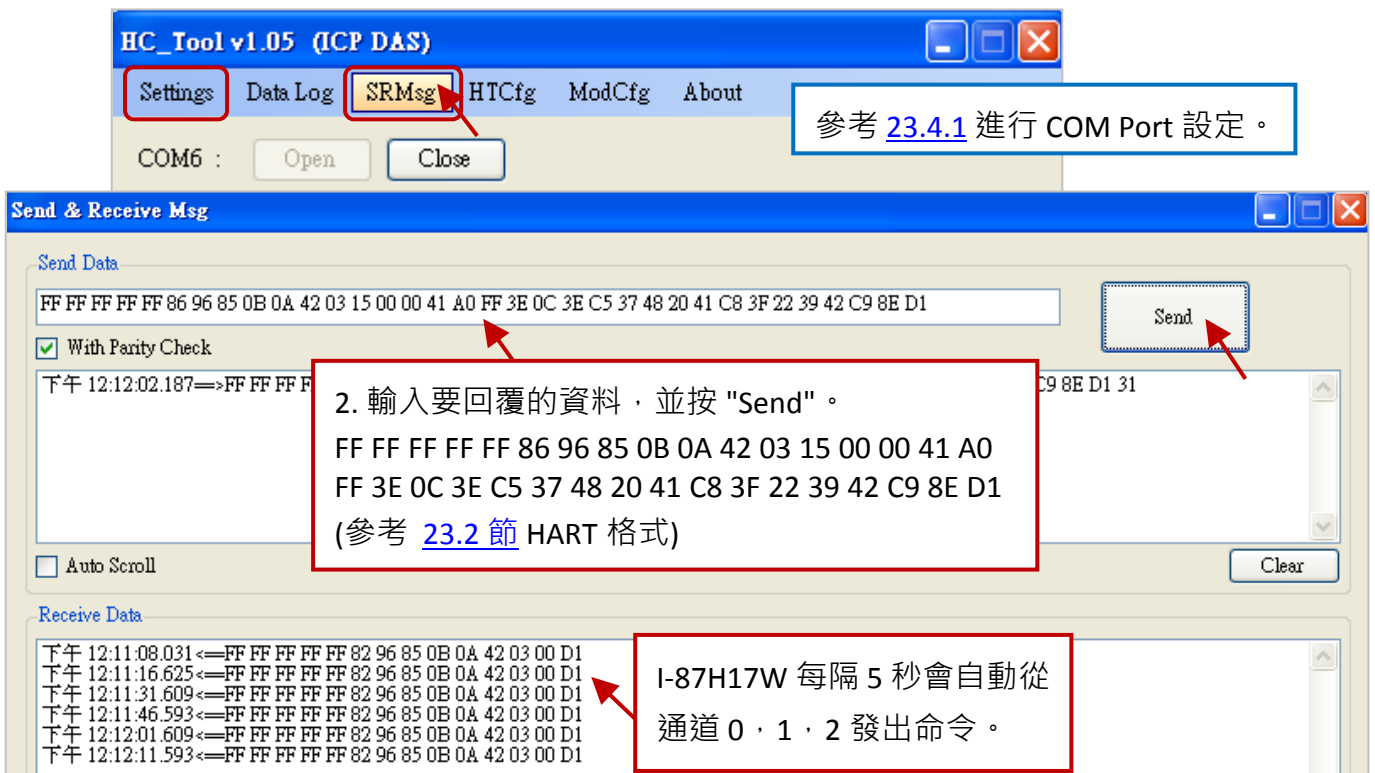
功能說明: (程式- [Demo HART 2](#) ; 接線說明 – [參考 23.4.1 節](#))

每隔 5 秒依序從 I-87H17W 的通道 0 → 1 → 2 → 0 (位於 Slot 2) , 自動發出並接收 HART Frame 。

下載後 , 請開啟觀測清單 (NewSpy) , I-87H17W 每隔 5 秒會自動從通道 0 , 1 , 2 發出命令 。



在 HC_Tool 中 , 請先設定好使用的 COM Port 與 HART 格式 (參考 [23.4.1 節](#)) , 再點選 "SRMsg" 開啟 "Send & Receive Msg" 視窗 , 並輸入需回覆的資料 , 再按 "Send" 。



(此處顯示 15 秒 , 是由於本測試只接了通道 0。)

Preamble	Delimiter	Address	Command	Byte Count	Data
I-87H17W 發出 HART 命令 (16 進制)					
FF FF FF FF FF	82	(long) 96 85 0B 0A 42	03	00	-
設備回覆的資料 (16 進制)					
FF FF FF FF FF	86	(long) 96 85 0B 0A 42	03	15 ₍₁₆₎	00 00 <u>41 A0 FF 3E</u> 0C <u>3E C5 37 48 20 41 C8</u> <u>3F 22 39 42 C9 8E D1</u>

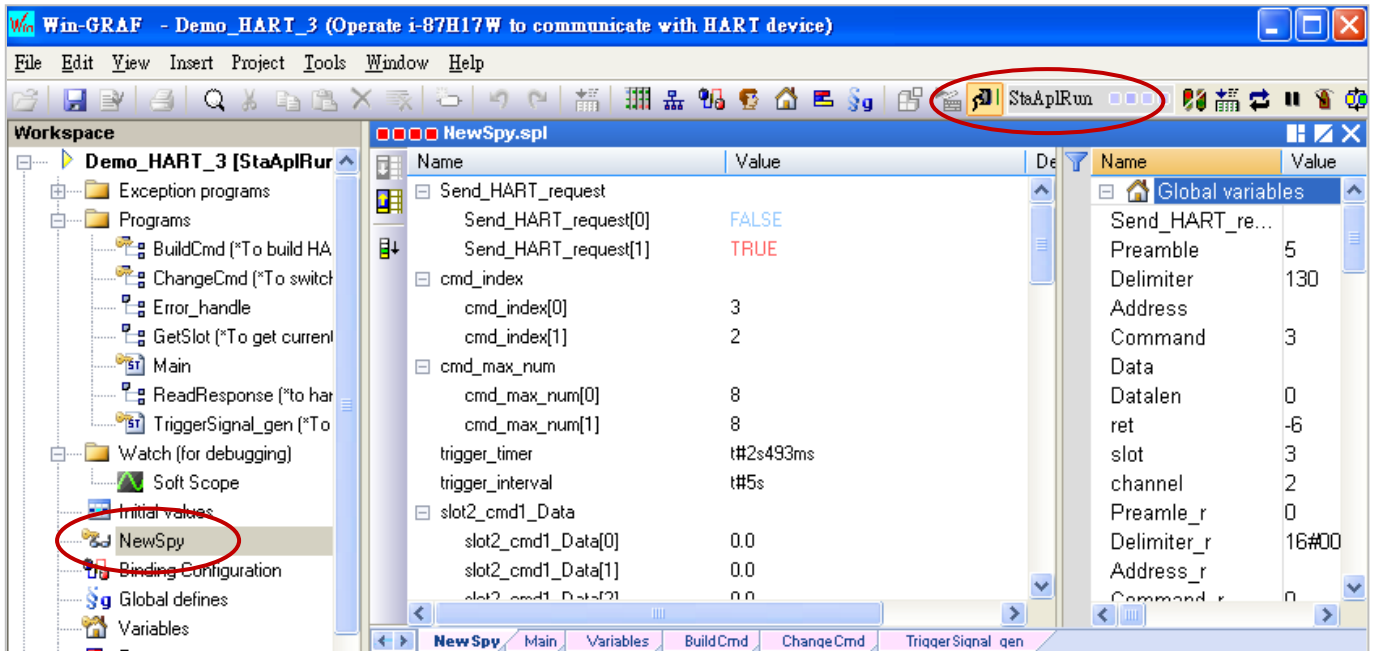
The screenshot shows the 'NewSpy.spl' interface with a table of variables and their values. Red boxes and arrows highlight specific data points and provide explanatory text in Chinese.

Name	Value	Description
send_HART_request	FALSE	
cmd_index	2	目前從 I-87H17W 的通道 2 發送命令。
cmd_max_num	3	
cmd1_Data		由通道 0 回覆的資料。程式中 (P23-17) 會將取得的 "Data" 轉成 4 個 REAL 值。
cmd1_Data[0]	20.12463	data1 = 20.12463 (41 A0 FF 3E)
cmd1_Data[1]	0.385187	data2 = 0.385187 (3E C5 37 48)
cmd1_Data[2]	25.030827	data3 = 25.030827 (41 C8 3F 22)
cmd1_Data[3]	100.778938	data4 = 100.778938 (42 C9 8E D1)
cmd1_response_counter	1	目前通道 0 已回覆 1 次。I-87H17W 每 5 秒會由通道 0~2 發送命令一次，若 Timeout 會顯示 "-1"，目前未回覆 15 次。
cmd1_error_counter	15	
cmd1_error_code	-1	
cmd2_Data		
cmd2_Data[0]	0.0	
cmd2_Data[1]	0.0	
cmd2_Data[2]	0.0	
cmd2_Data[3]	0.0	
cmd2_response_counter	0	
cmd2_error_counter	16	
cmd2_error_code	-1	
cmd3_Data		
cmd3_Data[0]	0.0	
cmd3_Data[1]	0.0	
cmd3_Data[2]	0.0	
cmd3_Data[3]	0.0	
cmd3_response_counter	0	
cmd3_error_counter	15	
cmd3_error_code	-1	
hart_bus_err_msg	'Error: Timeout'	
hart_send_err_msg	'Send success'	
hart_read_err_msg	'Read success'	

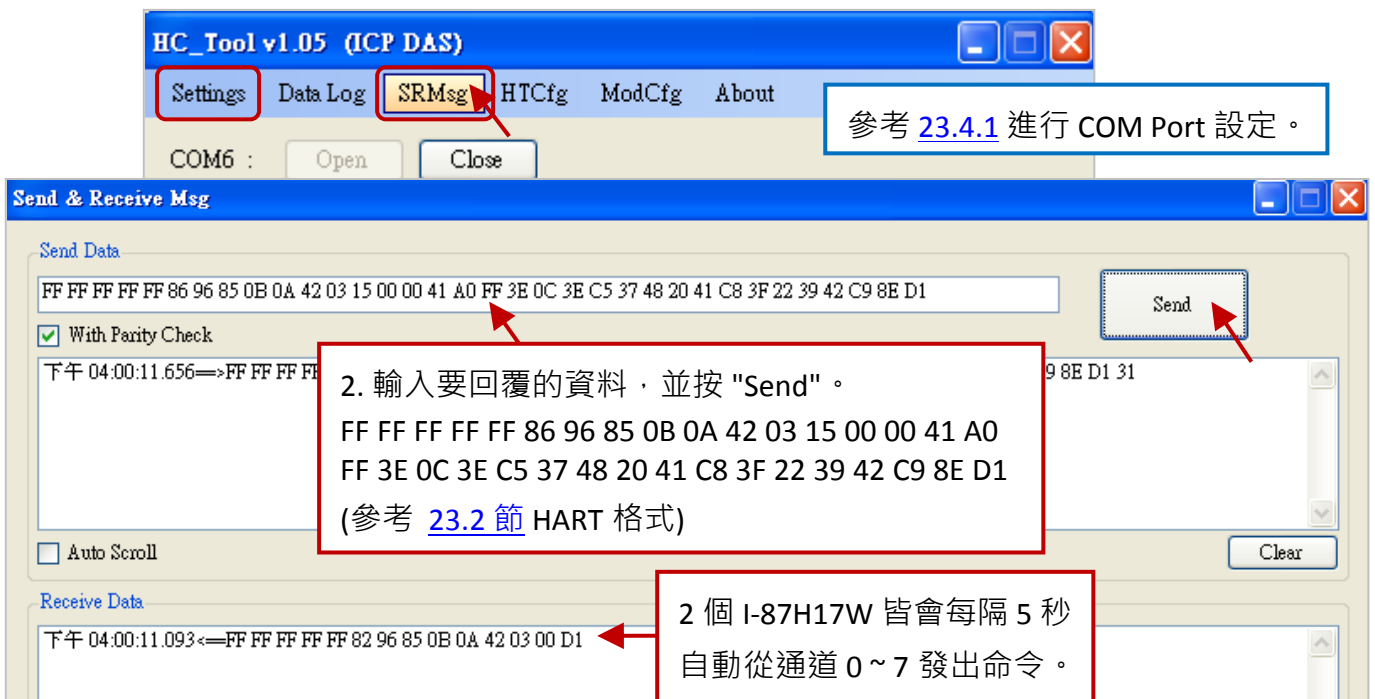
功能說明: (程式- [Demo HART 3](#) ; 接線說明 – [參考 23.4.1 節](#))

在 Slot 2 與 Slot 3 共使用了 2 個 I-87H17W 模組，並在同一時間，依序從通道 0~7，每隔 5 秒自動發出並接收 HART Frame。

下載後，請開啟觀測清單 (NewSpy)，2 個 I-87H17W 皆會每隔 5 秒自動從通道 0~7 發出命令。



在 HC_Tool 中，請先設定好使用的 COM Port 與 HART 格式 (參考 [23.4.1 節](#))，再點選 "SRMsg" 開啟 "Send & Receive Msg" 視窗，並輸入需回覆的資料，再按 "Send"。



Preamble	Delimiter	Address	Command	Byte Count	Data
I-87H17W 發出 HART 命令 (16 進制)					
FF FF FF FF FF	82	(long) 96 85 0B 0A 42	03	00	-

Preamble	Delimiter	Address	Command	Byte Count	Data
設備回覆的資料 (16 進制)					
FF FF FF FF FF	86	(long) 96 85 0B 0A 42	03	15 ₍₁₆₎	00 00 <u>41 A0 FF 3E</u> 0C 3E C5 37 48 20 <u>41 C8</u> 3F 22 39 <u>42 C9 8E D1</u>

NewSpy.spl

Name	Value	Description
Send_HART_request		
Send_HART_request[0]	FALSE	
Send_HART_request[1]	TRUE	
cmd_index		
cmd_index[0]	3	
cmd_index[1]	2	
cmd_max_num		
cmd_max_num[0]	8	
cmd_max_num[1]	8	
trigger_timer	t#2s116ms	
trigger_interval	t#5s	
slot2_cmd1_Data		
slot2_cmd1_Data[0]	20.12463	
slot2_cmd1_Data[1]	0.385187	
slot2_cmd1_Data[2]	25.030827	
slot2_cmd1_Data[3]	100.778938	
slot2_cmd2_Data		
slot2_cmd3_Data		
slot2_cmd4_Data		
slot2_cmd5_Data		
slot2_cmd6_Data		
slot2_cmd7_Data		
slot2_cmd8_Data		
slot2_cmd_response_counter		
slot2_cmd_response_counter[0]	1	
slot2_cmd_response_counter[1]	0	
slot2_cmd_response_counter[2]	0	
slot2_cmd_response_counter[3]	0	
slot2_cmd_response_counter[4]	0	
slot2_cmd_response_counter[5]	0	
slot2_cmd_response_counter[6]	0	
slot2_cmd_response_counter[7]	0	
slot3_cmd1_Data		
slot3_cmd2_Data		
slot3_cmd3_Data		
slot3_cmd4_Data		
slot3_cmd5_Data		
slot3_cmd6_Data		
slot3_cmd7_Data		
slot3_cmd8_Data		
slot3_cmd_response_counter		
slot3_cmd_error_counter		
slot3_cmd_error_code		
hart_bus_err_Msg	'Error: Timeout'	
hart_send_err_msg	'Send Error: i-87H17W is off-line'	
hart_read_err_msg	'Read success'	

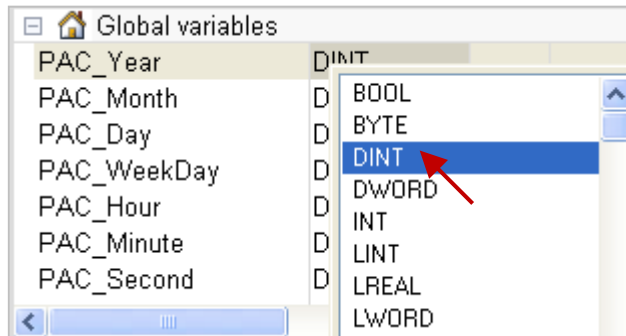
每隔 5 秒傳送命令一次。

於 Slot2，由 I-87H17W 的通道 0 回覆的資料。程式中會將取得的“Data”轉成 4 個 REAL 值。

data1 = 20.12463 (41 A0 FF 3E)
data2 = 0.385187 (3E C5 37 48)
data3 = 25.030827 (41 C8 3F 22)
data4 = 100.778938 (42 C9 8E D1)

附錄 A 資料型態與數值範圍

您可在程式的變數區 (參考 [2.2.1 節](#)) 或 變數視窗 (參考 [2.2.2 節](#)) 中設定變數的資料型態。



下表為變數的基本資料型態與數值範圍:

資料型態	位元數	數值範圍
BOOL (*)	---	TRUE · FALSE
SINT	8 bits (Small int, signed)	-128 ~ +127
USINT	8 bits (Unsigned small int)	0 ~ +255
BYTE		
INT	16 bits (Int, signed)	-32768 ~ +32767
UINT	16 bits (Unsigned int)	0 ~ +65535
WORD		
DINT (*)	32 bits (Double int, signed)	-2147483648 ~ +2147483647
UDINT	32 bits (Unsigned double int)	0 ~ +4294967295
DWORD		
LINT	64 bits (Large int, signed)	$-2^{63} \sim +(2^{63}-1)$
ULINT	64 bits (Unsigned large int)	$0 \sim +(2^{64}-1)$
LWORD		
注意: 所有的 Win-GRAF PAC 皆不支援 "ULINT" 與 "LWORD" 資料型態。		
REAL (*)	32 bits (Floating point)	$\pm 3.4 \times 10^{-38} \sim \pm 3.4 \times 10^{38}$
LREAL	64 bits (Floating point)	$\pm 1.7 \times 10^{-308} \sim \pm 1.7 \times 10^{308}$
STRING (*)	最多 255 個字元	---
TIME (*)	32 bits	T#0ms ~ T#23h59m59s999ms

(*): 表示常用的資料型態

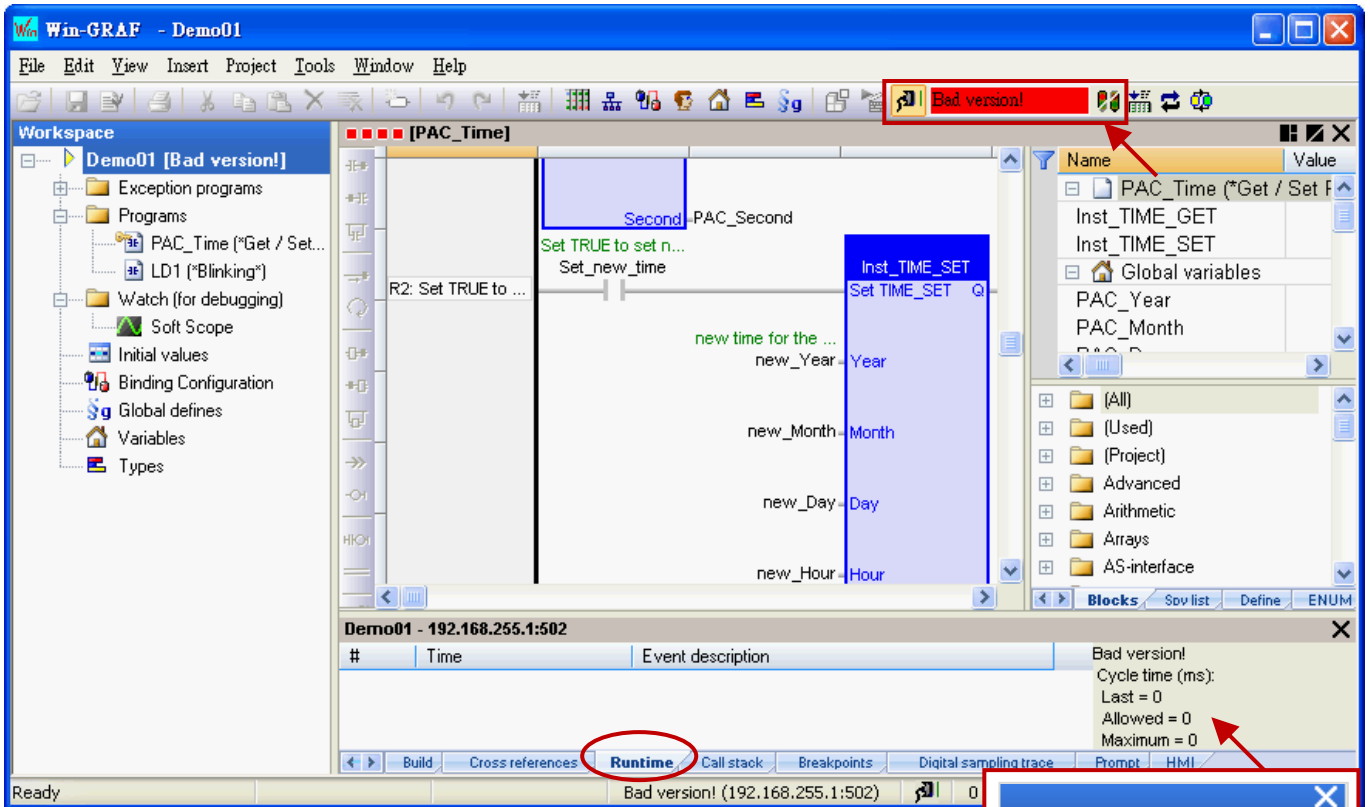
附錄 B 錯誤訊息排除

若 PC 與 PAC 之間的連線出現了錯誤訊息，請參考以下內容來排除問題。

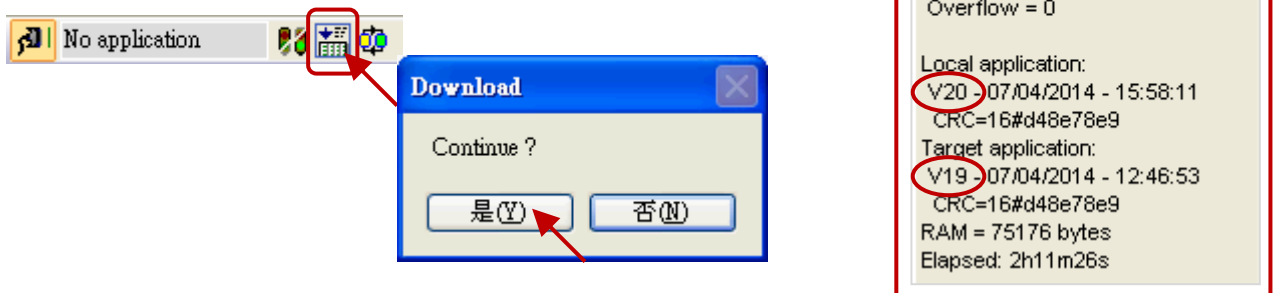
1. 若出現 “Bad version!” 錯誤訊息:

表示程式有修改並重新編譯過，因此 PC 與 PAC 中的程式 (編譯) 版本不一致。

1. 點選 “Stop application” 按鈕，停止 PAC 中運行的程式。



2. 再點選 “Download” 按鈕，重新下載程式。

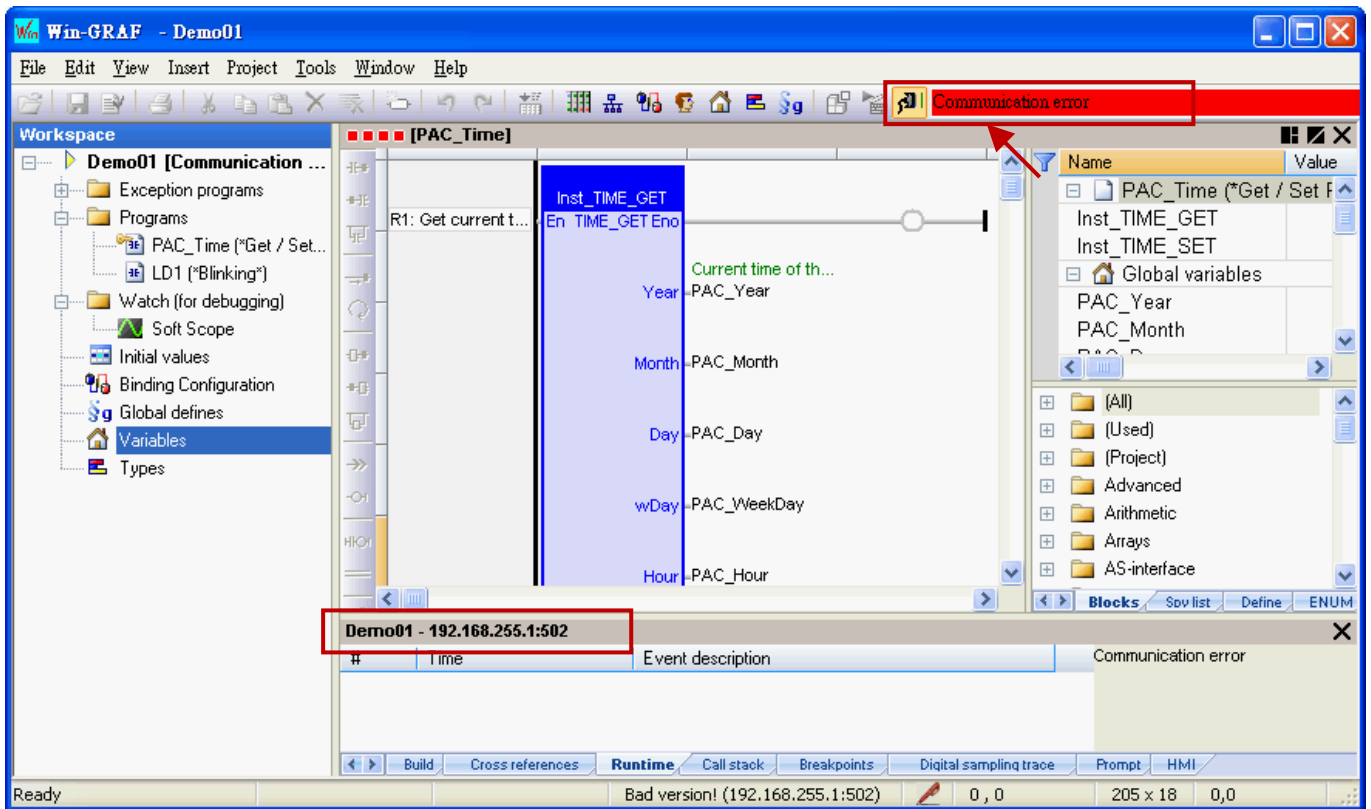


3. 出現 “RUN” 表示程式正常運行。



2. 若出現 “Communication error” 錯誤訊息:

表示 PC 與 PAC 之間通訊失敗。



1. 確認您的 Win-GRAF PAC 已經開機且網路通訊正常。
2. 確認 PAC IP 與專案中的設定一致 (參考 [2.3.5 節](#)，此例為 “192.168.255.1:502”)。
3. 確認您的 PC 網路通訊正常。

附錄 C 啟動 WinCE PAC 螢幕保護功能

啟動 WinCE PAC 的螢幕保護功能，請設定下列兩個項目。

1. 請執行“控制台”>“電源”>“配置”，在“電源配置”項目選擇“AC 電源”，並將“使用者閒置”與“系統閒置”設定同樣的值 (或設定“系統閒置”的值比“使用者閒置”的值大)。
2. 然後，記得執行“WinPAC Utility”>“File”>“Save and Reboot”存檔及重新啟動。

下方以 WP-8xx8 為例來說明:

如果使用者沒有碰觸螢幕或按鍵，設定的時間到時，WP-8xx8 會關閉背光來啟動螢幕保護功能。之後只要使用者碰觸螢幕或按鍵，WP-8xx8 就會再次開啟螢幕背光。

若不想使用螢幕保護功能，請設定“User Idle”與“System Idle”為“Never”，並記得要執行“WinPAC Utility”>“File”>“Save and Reboot”，儲存設定並重新啟動。



附錄 D 使用 RS-232/485/422 擴充卡

客戶可以在 Win-GRAF PAC (見 [P1-1](#)) 的插槽 (Slot 0 ~ 7) 使用下列模組來擴充 16 個以上的 COM 埠。

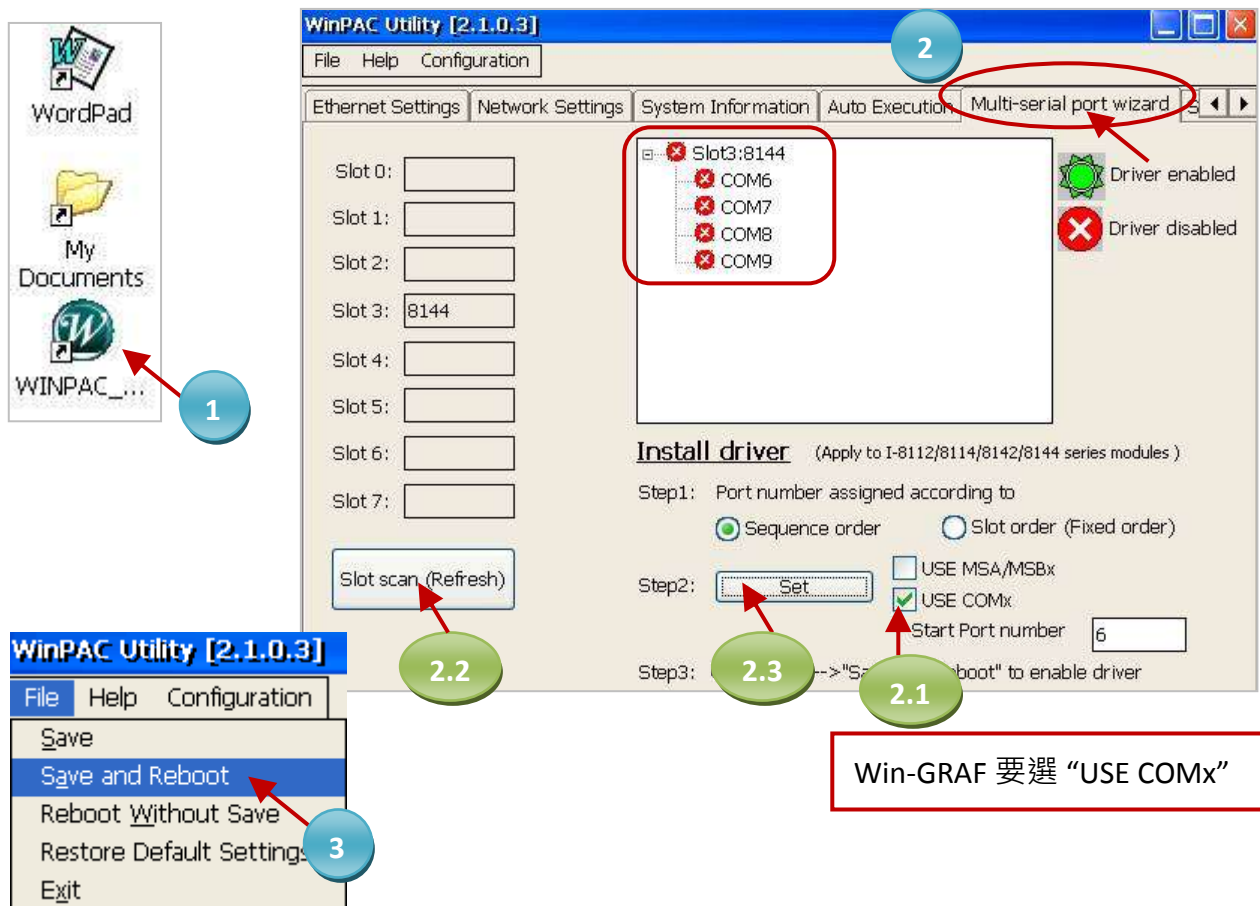
- I-8112iW : 2 埠 隔離式 RS-232 模組
- I-8114iW : 4 埠 隔離式 RS-232 模組
- I-8114W : 4 埠 無隔離式 RS-232 模組
- I-8142iW : 2 埠 隔離式 RS-422/RS-485 模組
- I-8144iW : 4 埠 隔離式 RS-422/RS-485 模組

註: WP-5xx8-CE7 不支援 XW-5xx 序列埠擴充子卡 (無法擴充序列埠)

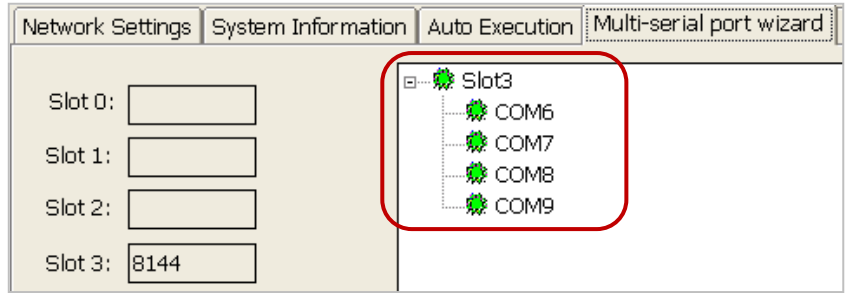
下方以 WP-8xx8 為例來說明:

使用上列模組前，必須先以“WinPAC Utility”進行設定。請將模組插入 WP-8xx8 的 0 ~ 7 槽 (最好是 0 ~ 3 槽)，再執行 WinPAC Utility :

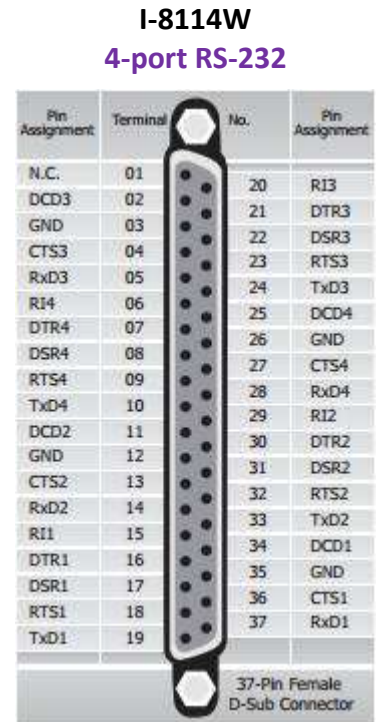
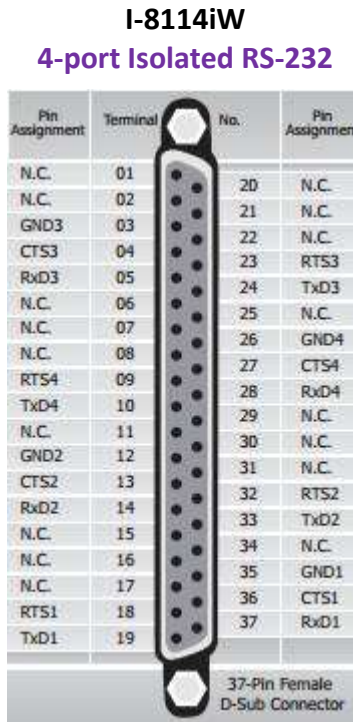
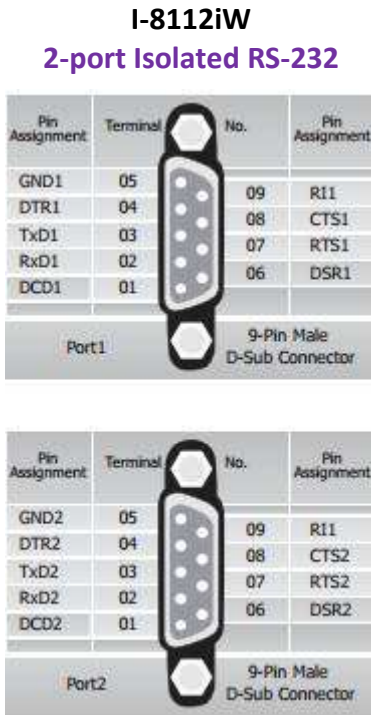
- 點選“Multi-serial port wizard”頁籤。
- 勾選“USE COMx” (**註:** Win-GRAF 不支援“USE MSA/MSBx”)。
- 點選“Slot scan”按鈕，會自動搜尋目前 PAC 插槽上的多序列埠模組，並在左欄中顯示卡號。(若先前有設定過，會在右欄中顯示 COM 埠的原始設定。)
- 按下“Set”按鈕，更新原始設定為模組目前的設定。
- 執行 [File] > [Save and Reboot] 儲存新設定，並重新啟動 WP-8xx8。



設定成功後，Win-GRAF 定義擴充卡的 COM 埠編號是 COM6 ~ COM37 (圖中的例子為 COM6 ~ COM9)。



接腳圖：



I-8142iW

Terminal No.	Pin Assignment
01	D1+/TxD1+
02	D1-/TxD1-
03	RxD1+
04	RxD1-
05	GND1
06	D2+/TxD2+
07	D2-/TxD2-
08	RxD2+
09	RxD2-
10	GND2
11	N.C.
12	N.C.
13	N.C.
14	N.C.
15	N.C.
16	N.C.
17	N.C.
18	N.C.
19	N.C.
20	N.C.

I-8144iW

Terminal No.	Pin Assignment
01	D1+/TxD1+
02	D1-/TxD1-
03	RxD1+
04	RxD1-
05	GND1
06	D2+/TxD2+
07	D2-/TxD2-
08	RxD2+
09	RxD2-
10	GND2
11	D3+/TxD3+
12	D3-/TxD3-
13	RxD3+
14	RxD3-
15	GND3
16	D4+/TxD4+
17	D4-/TxD4-
18	RxD4+
19	RxD4-
20	GND4

I-8142iW (2-port Isolated RS-422/485)

RS-485 port1: (D1+ , D1-)
RS-485 port2: (D2+ , D2-)

RS-422 port1: (TxD1+ , TxD1- , RxD1+ , RxD1-)
RS-422 port2: (TxD2+ , TxD2- , RxD2+ , RxD2-)

I-8144iW (4-port Isolated RS-422/485)

RS-485 port1: (D1+ , D1-)
RS-485 port2: (D2+ , D2-)
RS-485 port3: (D3+ , D3-)
RS-485 port4: (D4+ , D4-)

RS-422 port1: (TxD1+ , TxD1- , RxD1+ , RxD1-)
RS-422 port2: (TxD2+ , TxD2- , RxD2+ , RxD2-)
RS-422 port3: (TxD3+ , TxD3- , RxD3+ , RxD3-)
RS-422 port4: (TxD4+ , TxD4- , RxD4+ , RxD4-)

附錄 E 啟用一個序列埠 讓 Win-GRAF Workbench 連進來

(本節是以 WP-8xx8 作說明，來讓 Win-GRAF Workbench 可以連進來，其它 PAC 的操作方式一樣。)

Win-GRAF PAC 皆已啟用 Ethernet Port 來讓 Win-GRAF Workbench 可以連進來 除錯 或 下載 或 上傳 專案 (Project)，若使用者想另外啟用一個序列埠 (一般是 RS-232 或 RS-485) 來使用，其方法如下：

方法 1:

Win-GRAF PAC 一開機，會嘗試從 \System_Disk\Win-GRAF\ 內去讀取一個名為 “Extra_Ports.txt” 的檔案，其內容類似如下：

COM1:19200,N,8,1

表示啟用 COM1 且
通訊速率 (Baud Rate) = 19200 bps

若想啟用 COM2 且 通訊速率為 9600 bps，請修改內容為：

COM2:9600,N,8,1

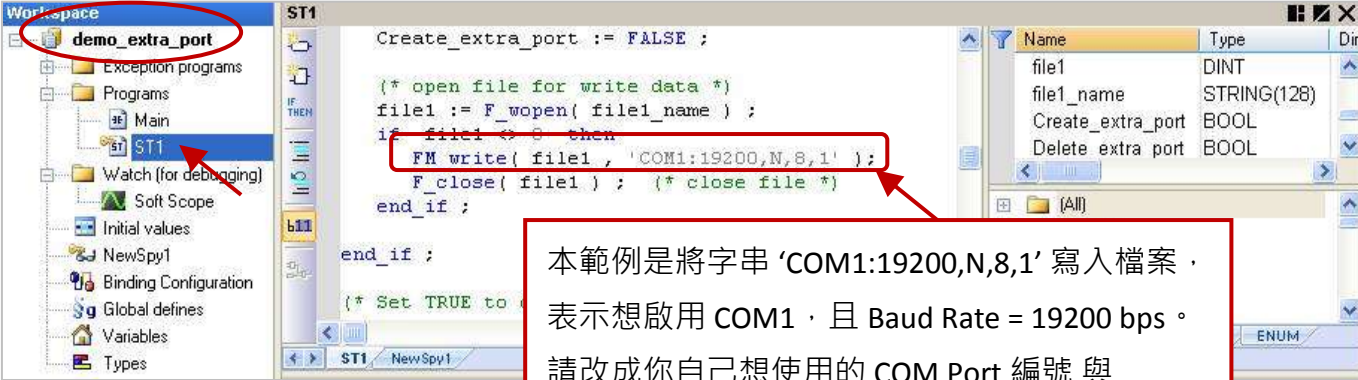
因此，只需把此 “Extra_Ports.txt” 文字檔用 FTP 的方式丟進 PAC 的 \System_Disk\Win-GRAF\ 路徑內，再將 PAC 重新開機一次即可。

取消啟用:

若想取消此 COM Port 的使用，可刪除 PAC 中 \System_Disk\Win-GRAF\Extra_Ports.txt 檔案，再將 PAC 重新開機即可。

方法 2:

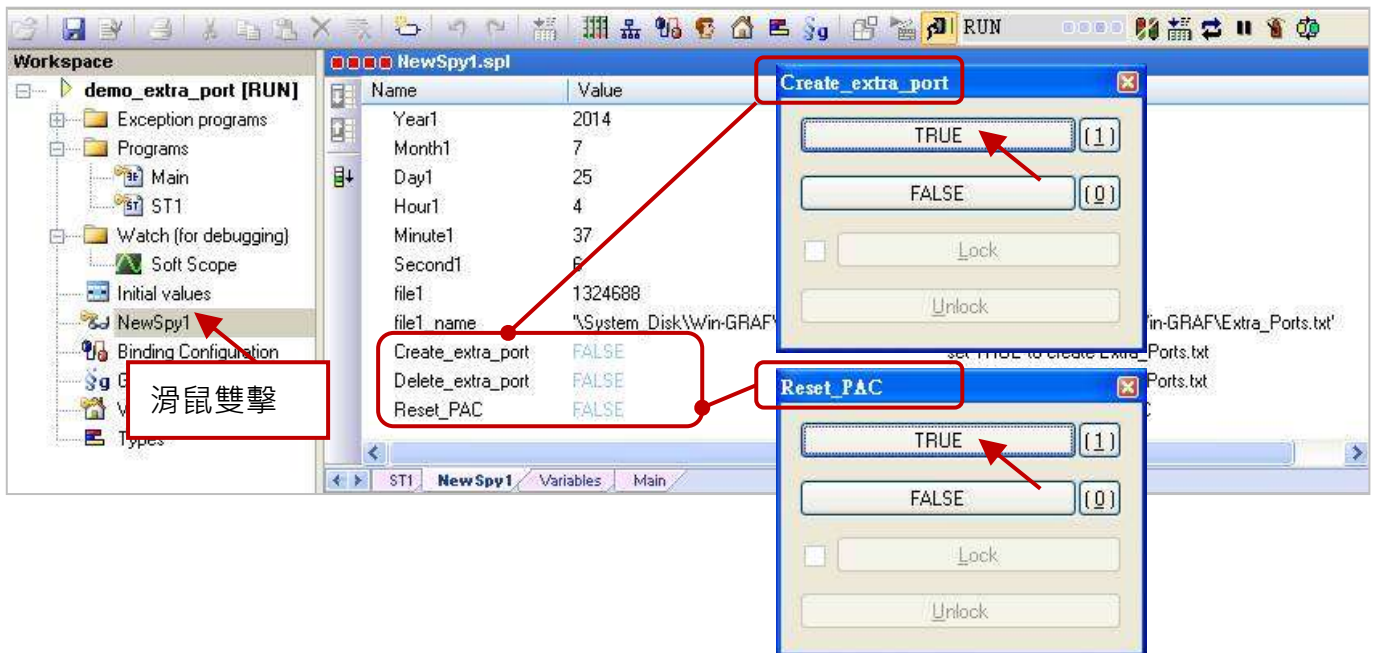
在 PAC 的出貨光碟中，有提供一個 "demo_extra_port" 範例程式，請先回存 (參考 [12 章](#)) 並以滑鼠雙擊開啟 ST1 程式來修改，之後要將程式重新編譯並下載到 PAC 內 (參考 [2.3.5 節](#))。



本範例是將字串 'COM1:19200,N,8,1' 寫入檔案，表示想啟用 COM1，且 Baud Rate = 19200 bps。請改成你自己想使用的 COM Port 編號與 baud rate 並重新編譯程式一次。

測試程式:

1. 連上 Win-GRAF PAC 後，在觀測清單中設定 "Create_extra_port" 為 "TRUE"。此例會在 \System_Disk\Win-GRAF\ 目錄下建立一個 "Extra_Ports.txt" 且內容為 "COM1:19200,N,8,1"。
2. 設定 "Reset_PAC" 為 "TRUE"，PAC 會自動重開機並套用此設定。



接下來要在 Win-GRAF Workbench 上若想要換成 Serial Port 來連到 Win-GRAF PAC。請參考 [2.3.5 節](#) 的步驟 1~2，並改以選擇 "Serial link" 即可透過 COM Port 來連到 PAC。



取消啟用:

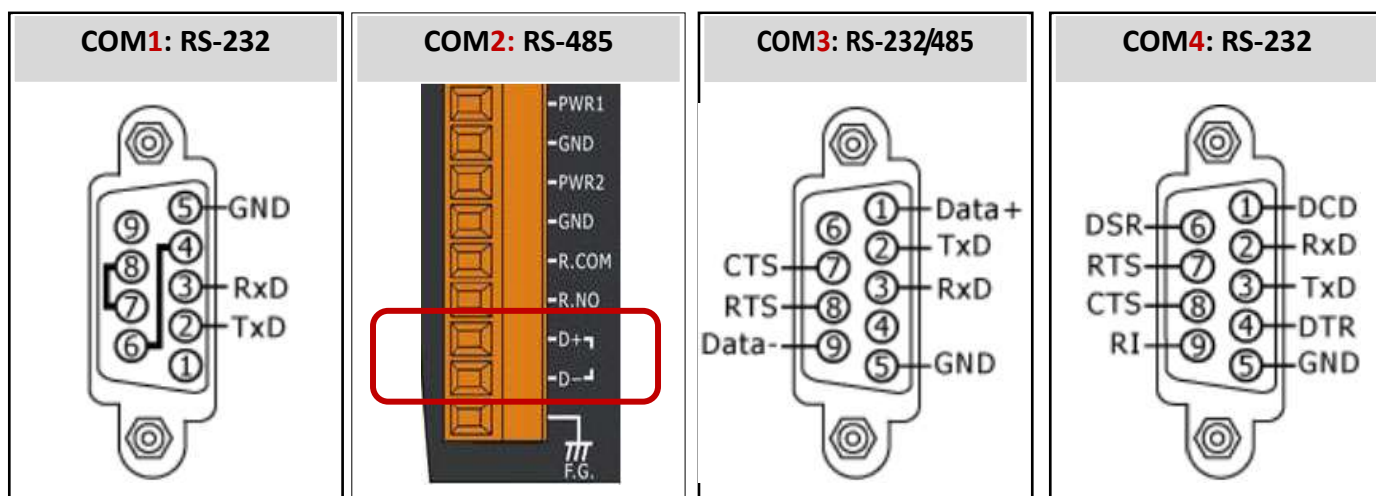
若想取消此 COM Port 的使用，請將 "Delete_extra_port" 設定為 "TRUE"，會刪除 PAC 中 \System_Disk\Win-GRAF\Extra_Ports.txt 檔案，再將 "Reset_PAC" 設定為 "TRUE" 讓 PAC 自動重開機一次即可。



附錄 F PAC 上序列埠的接腳圖

WP-8448/8848:

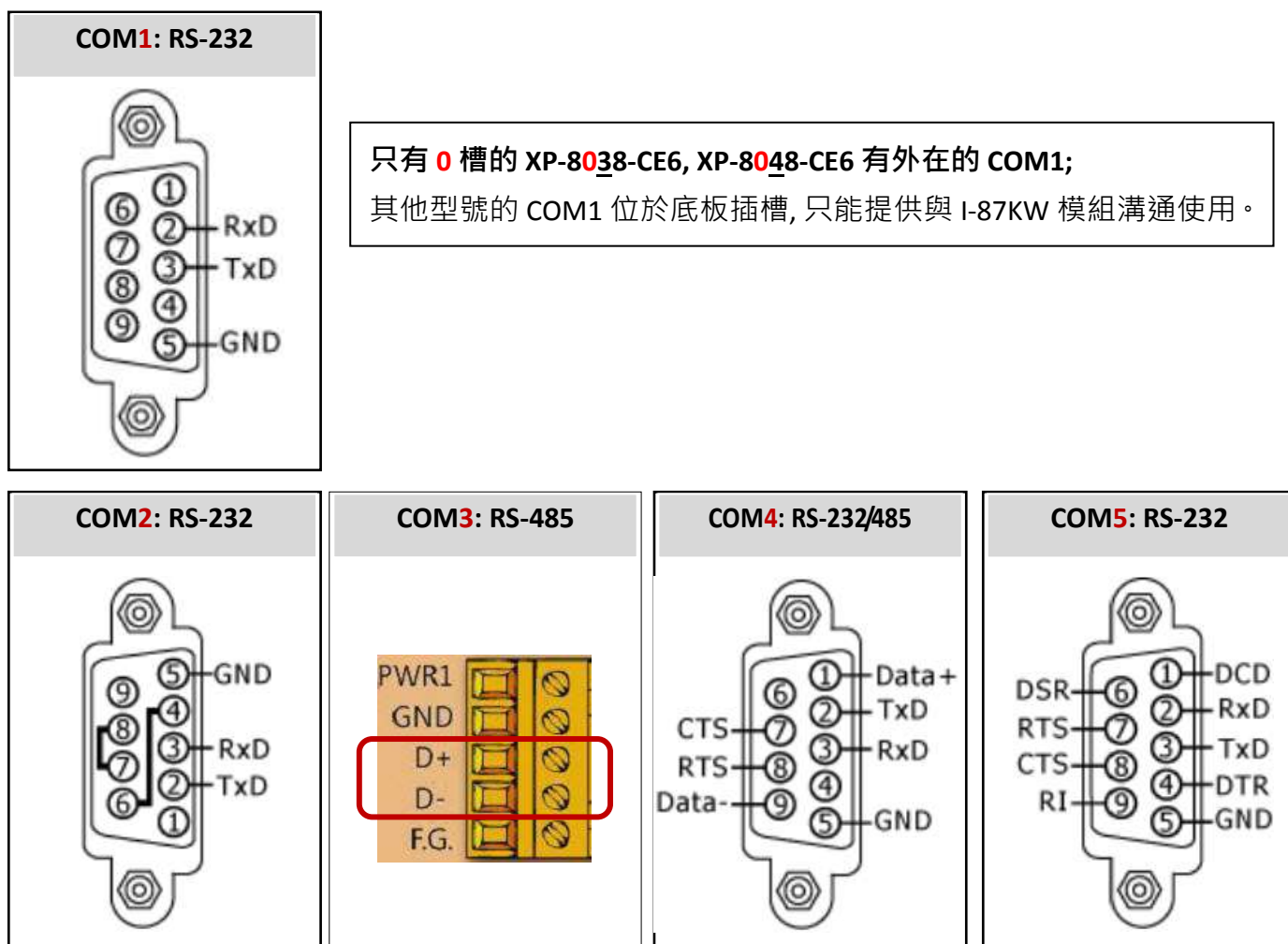
COM1 ~ COM4 的接腳圖。



Note: WP-8148 has no COM3, COM4.

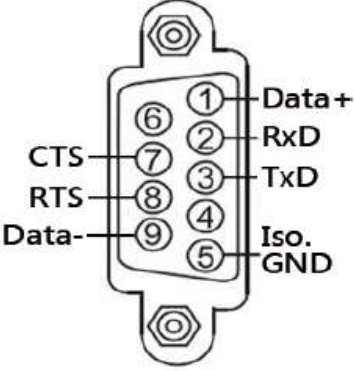
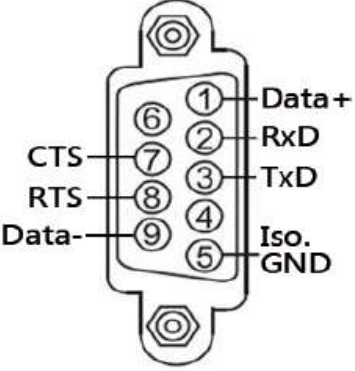
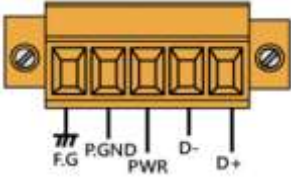
XP-8038-CE6/8138-CE6/8338-CE6/8738-CE6 與 XP-8048-CE6/8348-CE6/8848-CE6 :

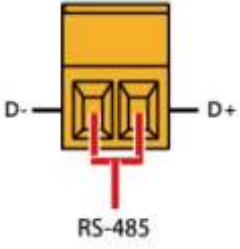
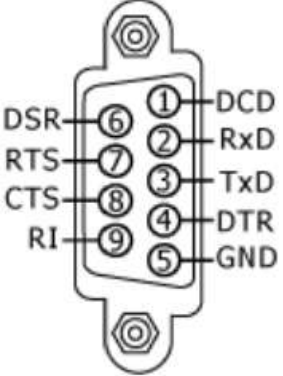
COM1 ~ COM5 的接腳圖。



VP-x2x8-CE7:

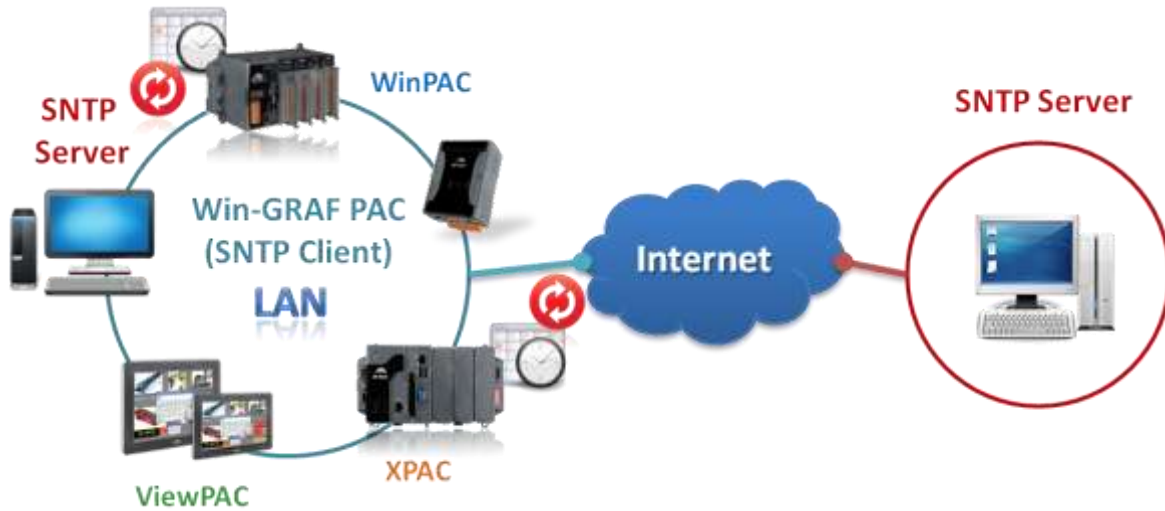
COM1 ~ COM3 的接腳圖。

VP-22x8-CE7	VP-42x8-CE7	
COM1 & COM2 : RS-232/RS-485	COM1 & COM2 : RS-232/RS-485	COM3 : RS-485 (D+, D-)
		

VP-1238-CE7	
COM2 : RS-485 (D2+, D2-)	COM3 : RS-232
	

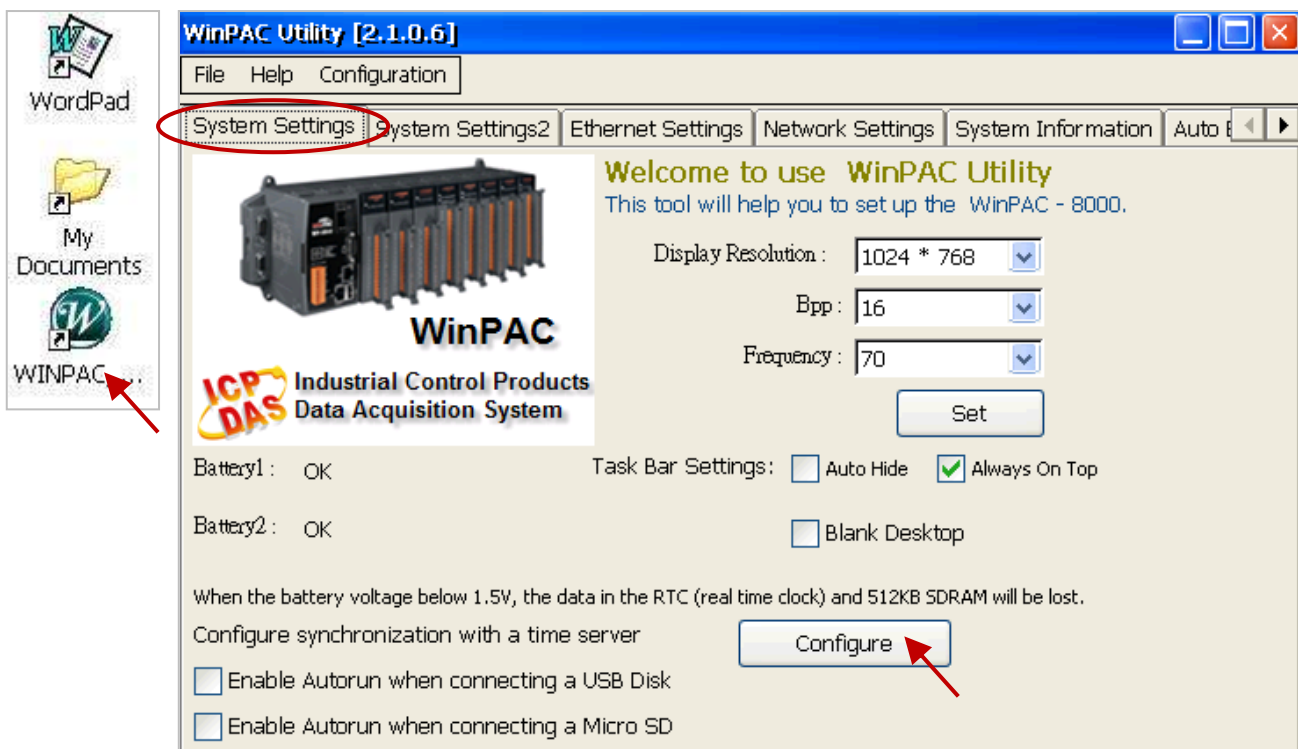
附錄 G PAC 的 SNTP 網路自動校時功能

Win-GRAF PAC (WP-8xx8, WP-8xx8-CE7, WP-5238-CE7, XP-8xx8-CE6, VP-x2x8-CE7) 支援 SNTP Client 網路自動校時功能。此章節將說明 Win-GRAF PAC 如何透過區域網路 (LAN) 或是網際網路 (Internet) 中的 SNTP Server 來進行自動校時功能。



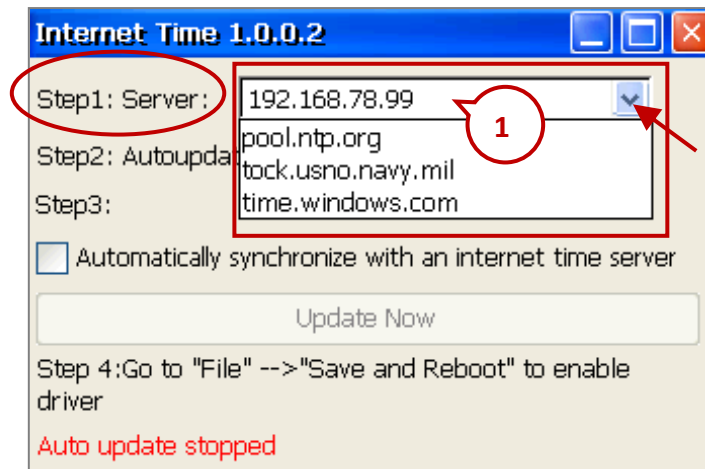
G.1 設定 SNTP Client 的網路自動校時

執行 PAC (例如: WP-8xx8) 桌面上的 "WinPAC Utility"，並點選 "System Settings" 頁面下的 "Configure" 按鈕，開啟網路校時的設定視窗。



步驟 1：指定時間伺服器 (SNTP Server)

在 "Internet Time" 視窗中，您可選擇預設的 SNTP Server (如下圖)，讓 Win-GRAF PAC 透過 Internet 來進行自動網路校時。或是，使用一台電腦並將其設定為 SNTP Server (參考 G.2 節)，並在 "Server" 欄位輸入其 IP 位址 (例如: 192.168.78.99) 來透過 LAN 或 Internet 進行自動網路校時。



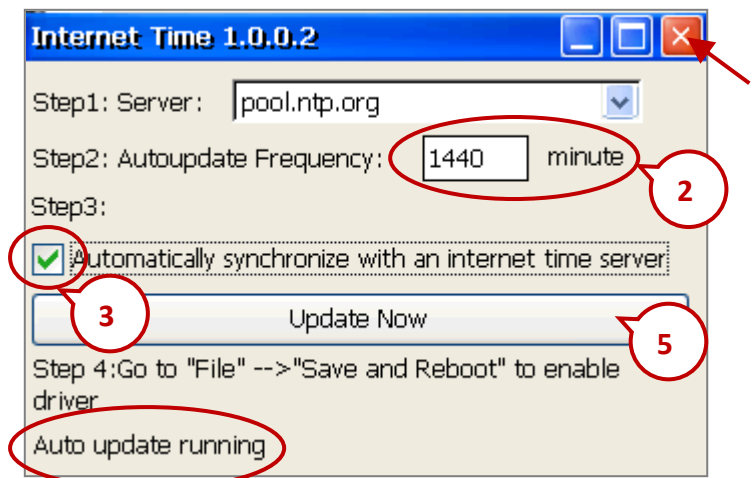
註: 如果 Win-GRAF PAC 是透過 Internet 與不同網域的 SNTP Server 進行自動網路校時，還需設定 PAC 的 "Default Gateway"，設定方式請見 G.3 節。

步驟 2：指定網路自動校時的頻率

若需要一天 (24 小時) 網路自動校時一次，則輸入 "1440" (單位: 分鐘，最小的設定時間為 5 分鐘)。

步驟 3：啟用自動校時

勾選自動校時選項，接著按右上角的 "X" 離開設定視窗。
(若取消勾選，表示停止此功能。)

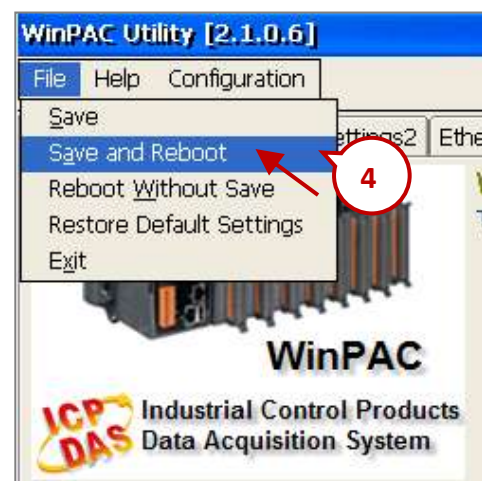


步驟 4：儲存設定

點選 WinPAC Utility 的 "File" > "Save and Reboot"，儲存設定，並重新啟動控制器。

步驟 5：檢視 SNTP 自動校時功能

PAC 開機且網路建立後，即會依照上述設定進行自動網路校時功能。如果需要立即更新時間，也可直接點選 "Internet Time" 視窗中的 "Update Now" 按鈕來更新時間。



G.2 設定一台 Windows XP 電腦為 SNTP Server 來進行測試

準備一台 Windows XP 電腦當作 WinPAC 自動網路校時的時間依據，並依照下列步驟設定。

步驟 1：啟用與設定 Windows XP SNTP Server

1. 開啟“登錄編輯程式”。

在 Windows XP 電腦，點選 [開始] > [執行]，在開啟欄中輸入“regedit”，按“確定”。



2. 啟動 SNTP Server。

一般 Windows XP 電腦的 SNTP Server 是預設開啟的。

左窗格:

請切換到下列目錄位置：

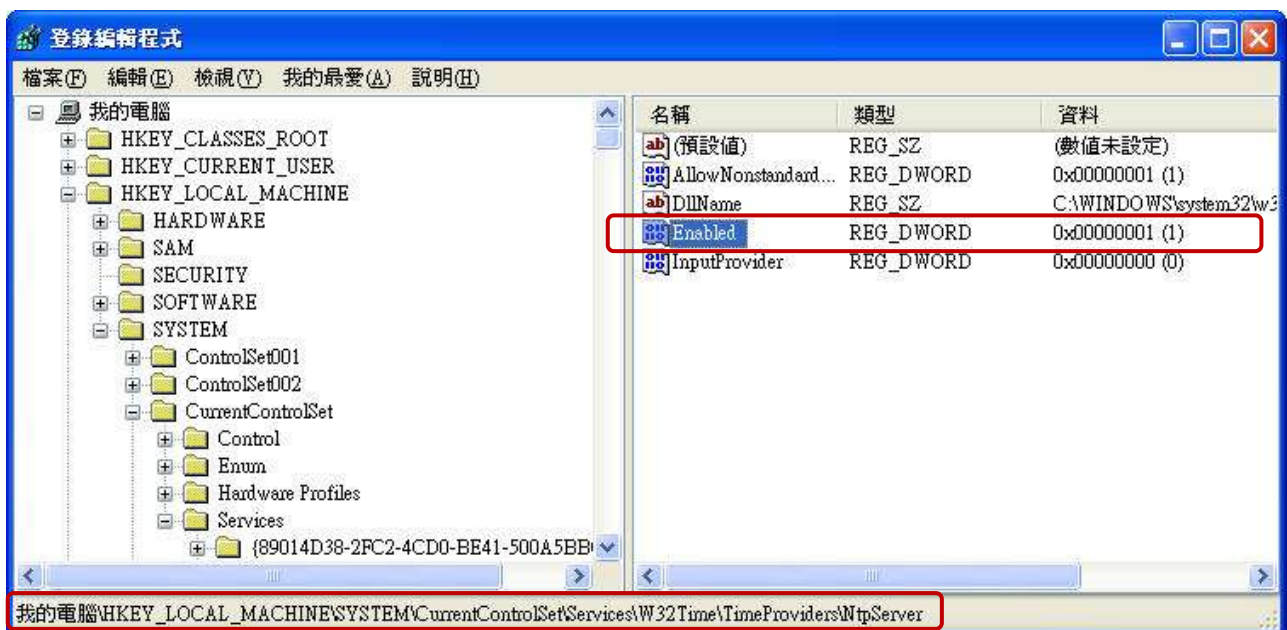
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer\

右窗格:

“**Enabled**”的“資料”欄最右邊數值是 (1)，代表 NTP Server 狀態為開啟。

若值為(0)，代表尚未開啟，請以右鍵點選“Enabled”，然後點選“修改”。

將“編輯 DWORD 值”視窗中的“數值資料”設定為 **1**，然後按“確定”。



3. 配置 Windows 時間服務以使用內部硬體時鐘。

左窗格:

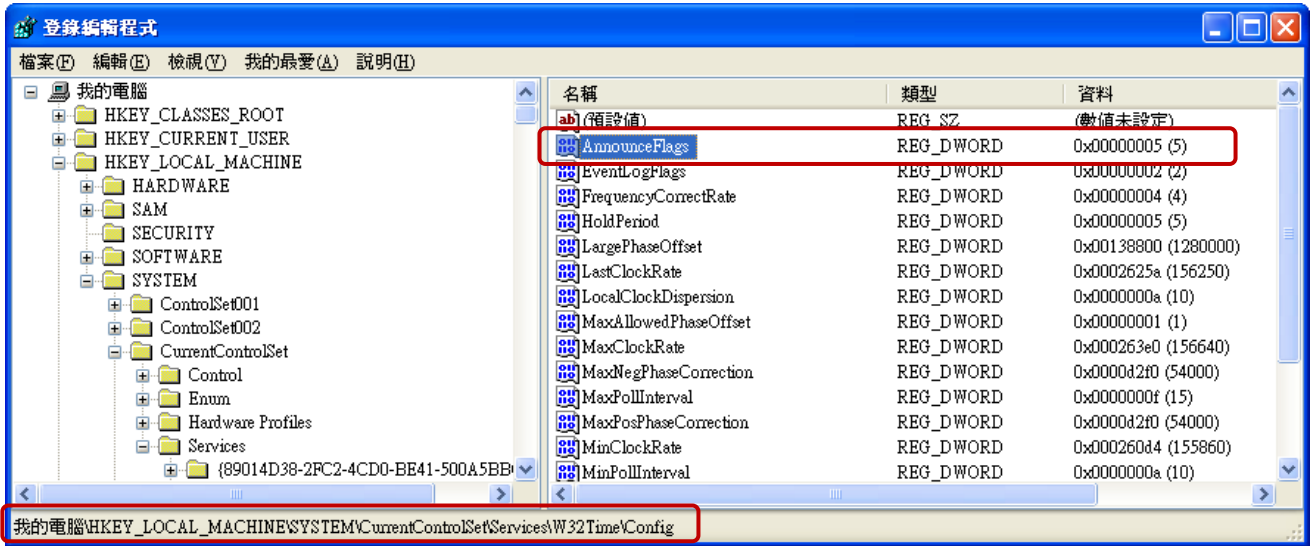
請切換到下列目錄位置:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config\

右窗格:

以滑鼠右鍵點選“AnnounceFlags”，選擇“修改”。

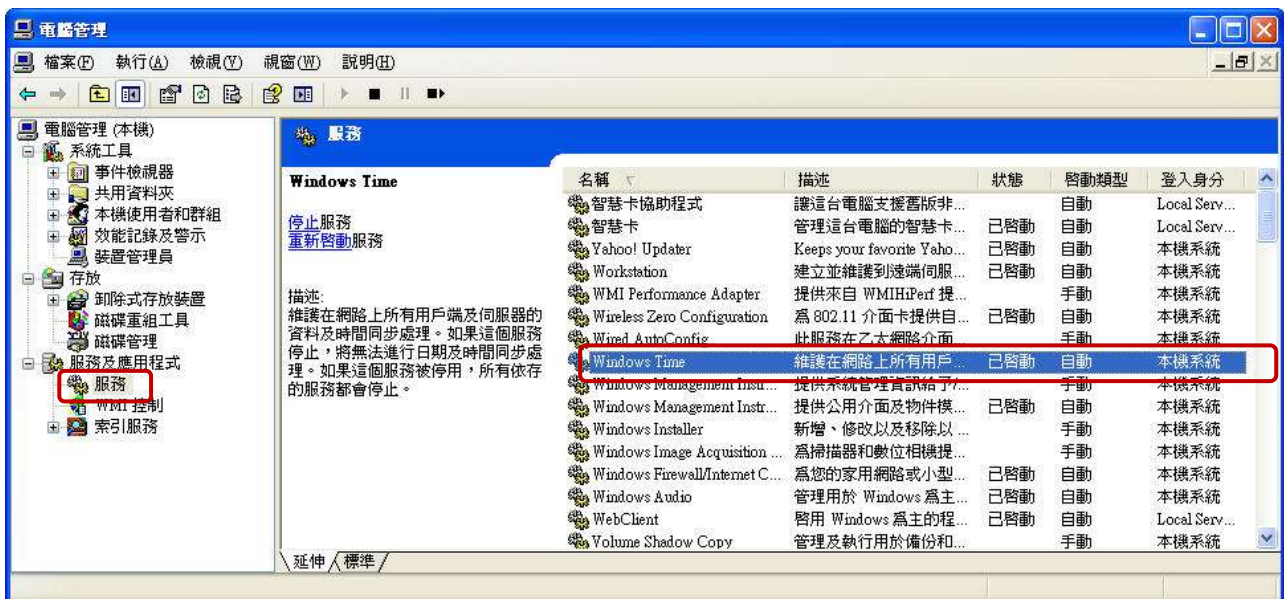
將“編輯 DWORD 值”視窗中的“數值資料”設定為 5，按“確定”並關閉“登錄編輯程式”。



步驟 2：重新啟動 Windows 時間服務

1. 在 Windows XP 電腦桌面，點選 [開始] > [執行]
2. 在開啟欄中輸入指令: “net stop w32time && net start w32time”，然後按“確定”。

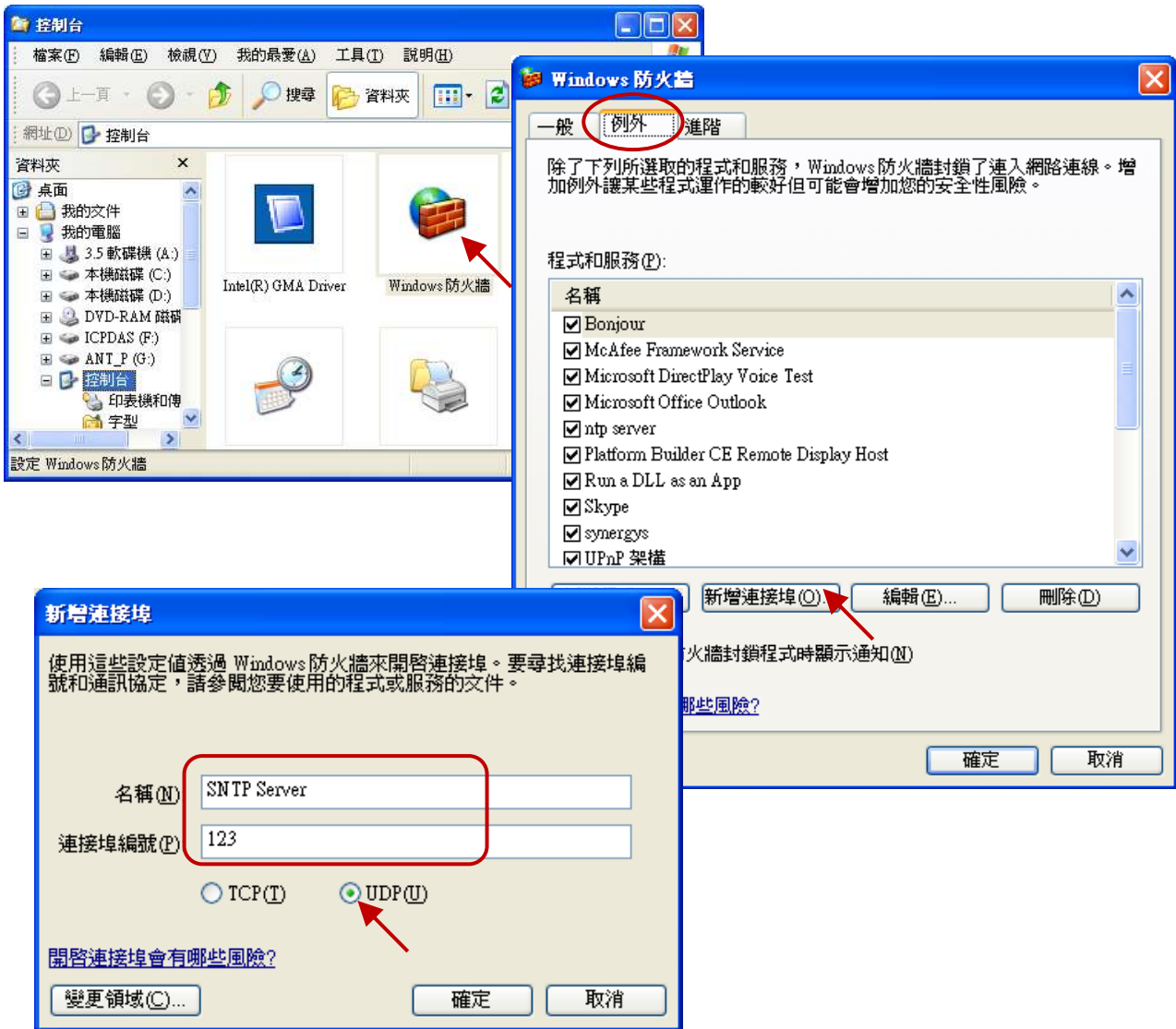
此步驟也可在視窗操作，請以滑鼠右鍵點選 [我的電腦] > 左鍵點選 [管理] > [服務] > 滑鼠雙擊右視窗的 [Windows Time]，即可停止/啟動“Windows Time”的時間服務，並設為“自動”。



步驟 3：防火牆問題

如果通訊需要通過 Windows 防火牆，則需開啟 UDP123 連接埠。

1. 在 Windows XP 電腦，點選 [開始] > [控制台] > [Windows 防火牆]。
2. 點選“例外”頁籤，並點選“新增連接埠”。
3. 給予一個名稱，如“SNTP Server”，設定連接埠為“UDP”，編號為“123”，按“確定”離開。



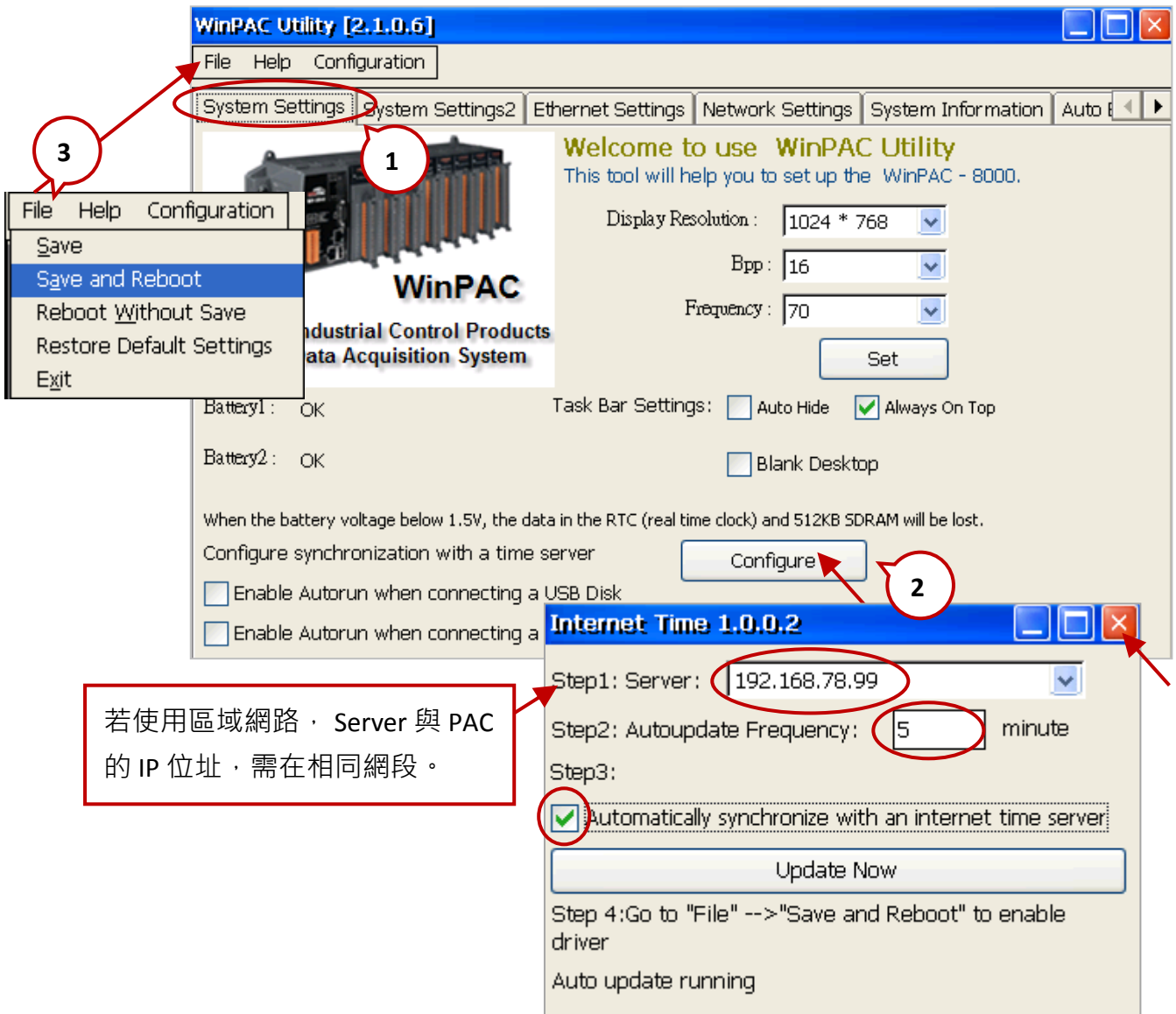
步驟 4：SNTP Server 需設定一個固定 IP 位址

無論是透過 LAN 或 Internet 來進行時間校正，SNTP Server 都需擁有一個固定的 IP 位址，且需和 WinPAC Utility 中設定的 IP 位址一致 (參考 G.1 節)。若透過 LAN 的方式，則 SNTP Server 與 WinPAC 的 IP 位址都需在相同的網段，才能成功的完成時間校正。

例如：本範例中，WinPAC 的 IP/Mask 位址是 "192.168.80.21 / 255.255.0.0"，而 SNTP Server 的 IP/Mask 位址是 "192.168.78.99 / 255.255.0.0"，此兩個設備是在相同的網段中。

步驟 5：測試

完成 SNTP Server 設定後，即可與 SNTP Client (例如: WinPAC) 進行測試。請執行 [G.1 節](#) 的步驟，填入 SNTP Server 的 IP 位址，修改自動更新頻率為 5 分鐘，勾選自動更新項目，並重新啟動 Win-GRAF PAC (例如: WP-8xx8)，即可如下圖的進行測試。



您可雙擊工具列來查看日期/時間是否已更新，之後可將自動更新頻率調回所需的時間。



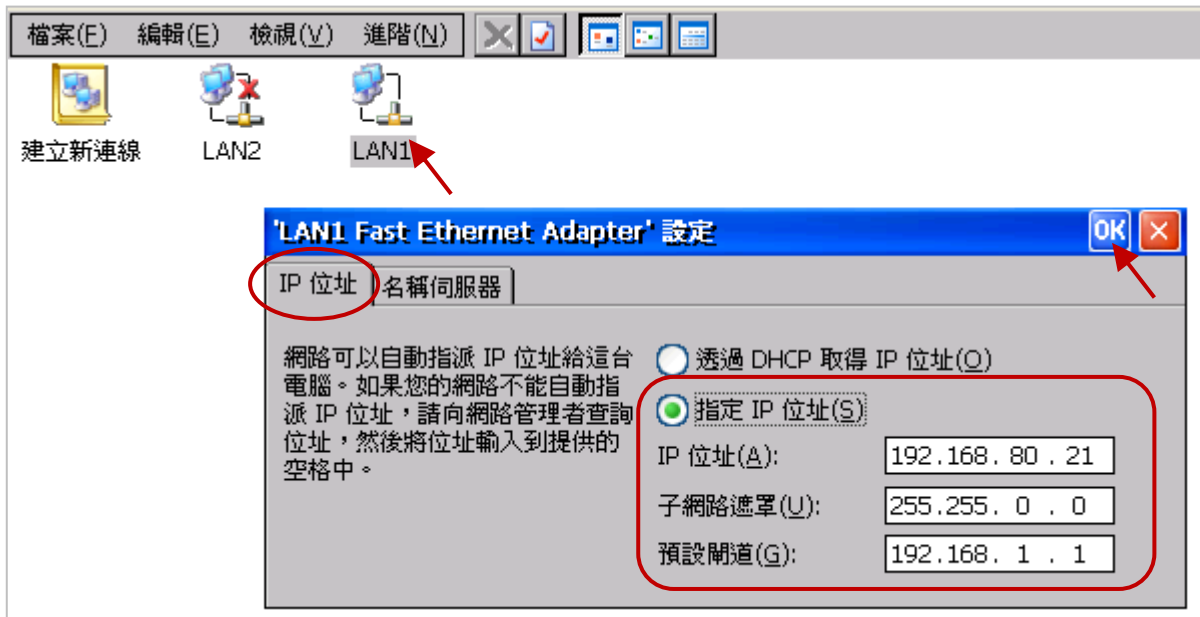
G.3 設定 PAC 的 Gateway 與 DNS Server 位址

如果您的 PAC (即, SNTP Client) 是透過 Internet 與不同網域的 SNTP Server 進行自動網路校時, 除了 IP/Mask 設定之外, 還需設定 PAC 的預設閘道 (Default Gateway) 與 DNS 伺服器 (DNS Server) 位址。

設定方法: (以下以 WinPAC 畫面為例)

1. 在 PAC 桌面, 開啟 [開始] > [設定] > [網路和撥號連線], 再雙擊 LAN1 (或 LAN2)。(可參考 1.3 節)
2. 請依照實際需求來輸入 Gateway 位址, 此例為 “192.168.1.1”。

註: Win-GRAF 應用中, 需指定固定 IP, 不可使用 DHCP。



3. 點選 “名稱伺服器” 頁籤, 設定 “主要 DNS”。完成後, 請點選 “OK”, 並重新開機。(Google 的 DNS Server 位址為 “8.8.8.8”, 中華電信的 DNS Server 位址為 “168.95.1.1”, 您可參考使用或輸入適合的位址。)

