
RPAC-2658M 使用手冊

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 **Linux** PAC 包括:

RPAC: RPAC-2658M

本公司出產的 Win-GRAF **WinCE** PAC 包括:

ViewPAC: VP-x208-CE7 [x: 2, 3, 4, 5, 6]

VP-x238-CE7 [x: 1, 4, 6]

WinPAC: WP-5238-CE7

WP-8x28-CE7 [x: 1, 4, 8]

WP-9x28-CE7 [x: 2, 4, 8]

XPAC: XP-8x38-CE6 [x: 0, 1, 3, 7]

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 © 2020 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。本書頁尾皆列有 版本號 與 發佈日期，可至網頁下載最新文件:

<https://www.icpdas.com/en/download/index.php?nation=US&kind1=&model=&kw=win-graf>

技術支援

- Win-GRAF 網站:
https://www.icpdas.com/en/product/guide+Software+Development__Tools+Win-GRAF
- Win-GRAF 下載區:
<https://www.icpdas.com/en/download/index.php?nation=US&kind1=&model=&kw=win-graf>
- Win-GRAF 範例程式:
請參考 [第 12 章](#) 了解更多關於 Win-GRAF 範例程式的說明。

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

RPAC-2658M 使用手冊	1
目 錄.....	2
第 1 章 軟體安裝與硬體設定	7
1.1 安裝 Win-GRAF 軟體	7
1.2 開啟 Win-GRAF 軟體	8
1.2.1 軟體介面說明.....	9
1.2.2 函式庫管理員 (Library Manager) 功能.....	10
1.3 設定 Win-GRAF PAC 的 IP 位址.....	11
1.4 硬體說明.....	12
第 2 章 編寫一個簡單的 WIN-GRAF 範例.....	13
2.1 建立 Win-GRAF 專案	13
2.1.1 建立樣版專案 (Demo01).....	13
2.1.2 重要專案設定.....	15
2.2 專案介紹.....	17
2.2.1 Demo01 - LD 程式.....	17
2.2.2 Demo01 - 變數說明.....	19
2.3 小試身手.....	20
2.3.1 宣告變數.....	20
2.3.2 建立 LD 程式.....	22
2.3.3 編譯程式.....	27
2.3.4 下載程式到 PAC	28
2.3.5 測試程式.....	31
第 3 章 MODBUS SLAVE: 開放 WIN-GRAF PAC 與 圖控/HMI 軟體來相互溝通	33
3.1 啟用 Win-GRAF PAC 為 Modbus TCP Slave	33
3.2 啟用 Win-GRAF PAC 為 Modbus RTU Slave.....	39
第 4 章 使用 “I/O BOARD” 功能	42
4.1 Ping_ip: 測試遠端 Ethernet/Internet 設備的連線.....	43
4.2 i_scale: 比例轉換.....	45
4.3 RPAC_2000_LED: 控制 RPAC-2658M 的 L1 與 L2 LED.....	47
4.4 RPAC_PAC_state: 偵測 RPAC-2658M 的 LAN 狀態.....	48
第 5 章 MODBUS MASTER: 連接其它 MODBUS SLAVE 設備	49
5.1 啟用 Win-GRAF PAC 為 Modbus RTU/ASCII Master.....	49
5.1.1 讀取 DI 資料	52
5.1.2 寫入 DO 資料.....	55

5.1.3	讀取 AI 資料.....	57
5.1.4	寫入 AO 資料 (16-bit)	60
5.1.5	寫入 AO 資料 (32-bit)	62
5.1.6	停止/啟用 Modbus RTU/ASCII Master Port	64
5.2	啟用 Win-GRAF PAC 為 Modbus TCP/UDP Master	65
5.2.1	連接 ET-7000 系列 I/O 模組.....	69
5.2.2	連接 ET-7060 (6 DI, 6 Relay Output).....	72
5.2.3	連接 ET-7018Z (10 AI)	74
5.2.4	停止/啟用 Modbus TCP/UDP Master Port	75
5.3	連接具有 2 個 IP 位址的 Modbus TCP Slave 設備	76
5.4	透過 tGW-700 閘道器來存取多個 Modbus RTU Slave	80
5.4.1	設定 tGW-700 (Modbus TCP 轉 Modbus RTU/ASCII 閘道器).....	80
5.4.2	透過 tGW-700 存取 LC 系列模組.....	83
5.4.3	測試範例程式 (demo_tgw725.zip)	87
第 6 章	可保存變數與資料儲存	88
6.1	使用 RETAIN_xxx 函式 (Function)	89
6.1.1	RETAIN_VAR: 設定保存一個變數	90
6.1.2	RETAIN_ARY: 設定保存一個陣列變數.....	91
6.1.3	RETAIN_FLAG_SET/GET/CLR (設定/取得/刪除 Flag 的狀態).....	92
6.2	將保存變數與資料存成檔案.....	94
6.3	將保存變數資料存到 FRAM	98
6.3.1	EEP_READ (讀取 FRAM).....	99
6.3.2	EEP_WRITE (寫入 FRAM)	99
第 7 章	在 PAC 間互傳資料 (DATA BINDING).....	100
第 8 章	連接 DCON I/O 模組	104
8.1	設定 "DCON" I/O 卡.....	105
8.2	使用 I/O 功能方塊.....	107
8.2.1	"D_7065" 功能方塊.....	108
8.2.2	"D_7018Z" 功能方塊	109
8.2.3	"D_7083" 功能方塊.....	111
8.2.4	"D_87084_FREQ" 功能方塊	112
8.2.5	"D_87084_CNT4" 功能方塊	113
8.2.6	"D_87084_CNT8" 功能方塊	114
8.2.7	"DL_100T485" 功能方塊	115
8.2.8	"D_GPS721" 功能方塊.....	116
8.3	I-87082W, I-87084W, I-7083 與 I-7080 模組的計數功能.....	118
8.3.1	COUNTER_START (開始計數)	118
8.3.2	COUNTER_STOP (停止計數).....	120
8.3.3	COUNTER_GET (取得計數值).....	121

8.3.4	COUNTER_STATE (取得計數狀態).....	122
8.3.5	COUNTER_RESET (重置計數值).....	123
第 9 章	即時線上更新 (ON-LINE CHANGE).....	124
9.1	"On-Line Change" 功能的使用限制.....	124
9.2	使用 "On-Line Change" 功能.....	126
第 10 章	資料/型態轉換與使用 PAC 時間.....	129
10.1	AI 資料轉換 (i_scale).....	129
10.2	AO 資料轉換 (i_scale).....	130
10.3	型態轉換函式 (ANY_TO_xxx).....	132
10.4	BCD 轉換 (BIN_TO_BCD, BCD_TO_BIN).....	133
10.5	組合/拆解 整數或布林值.....	134
10.6	組合/拆解 BYTE, WORD, DWORD.....	136
10.7	將變數拆成 Byte Array 或將 Byte Array 組成變數.....	139
10.8	取得/設定 PAC 時間.....	141
第 11 章	一般常用工具與有用的技巧.....	142
11.1	更新 Win-GRAF 函式庫 (Library).....	142
11.2	更新 Win-GRAF 驅動程式 (Driver).....	143
11.3	觀測清單 (Spy List).....	145
11.4	備份/回存一個 Win-GRAF 專案 (Project).....	147
11.5	以軟體重新啟動 PAC.....	149
11.6	在 LD 與 FBD 程式內使用 ST 語法.....	150
11.7	PAC 內套用配方表 (Recipe).....	151
11.8	Win-GRAF PAC 有支援的 Function 與 Function Block.....	153
11.9	上傳 Win-GRAF 專案原始碼.....	155
11.10	設定 PAC 的密碼.....	157
11.11	在 ST 程式內使用 Function 或 Function Block.....	159
11.12	保護您的 Win-GRAF 程式，讓盜用者無法使用.....	160
第 12 章	範例程式說明.....	164
12.1	範例程式列表.....	165
12.2	計時器 (Timer) 操作.....	167
12.2.1	啟動、停止、重置計時.....	167
12.2.2	週期性的操作.....	168
12.2.3	偵測穩定的 ON 或 OFF 訊號.....	170
12.2.4	觸發後維持 "ON" 一段時間.....	171
12.3	序列埠的通訊操作.....	172
12.3.1	使用 COM Port 來傳送一個字串.....	173
12.3.2	使用 COM Port 對設備一問一答.....	174
12.3.3	等待遠端設備傳送資料到 COM Port.....	176

12.3.4	使用 COM Port 定期回覆資料給遠端設備	178
12.4	讀/寫 PAC 內儲存裝置的檔案 (File).....	179
12.4.1	寫入資料到 PAC 內的檔案.....	180
12.4.2	讀取 PAC 內的檔案資料.....	182
12.4.3	資料紀錄功能.....	185
第 13 章	使用 C 程式來讀/寫 WIN-GRAF 變數	190
13.1	開放 Win-GRAF 變數給 C 程式使用	190
13.2	下載與編譯 C 程式.....	192
13.3	存取 Win-GRAF 變數	195
13.4	"Quicker.a" 內的函式說明	196
13.4.1	讀/寫 Boolean 變數	196
13.4.2	讀/寫整數變數	197
13.4.3	讀/寫實數變數	199
13.4.4	讀/寫字串變數	200
第 14 章	備援 (冗餘) 系統.....	201
14.1	哪些資料會自動備援到 Passive PAC?	203
14.2	備援系統 (旋轉開關 7 & 9).....	204
14.2.1	Win-GRAF 備援架構	204
14.2.2	設定備援系統.....	207
14.2.3	測試備援程式 (demo_rdn_2).....	210
14.3	範例程式介紹.....	212
14.3.1	[重要] "I/O Boards" 設定 (i_redundancy、i_redundancy_rs485).....	212
14.3.2	變數宣告 (demo_rdn_2).....	216
14.3.3	程式介紹 (demo_rdn_2).....	217
14.4	其他備援程式說明.....	219
14.4.1	測試備援程式 (demo_rdn_3、demo_rdn_1).....	219
14.4.2	程式介紹 (demo_rdn_4).....	223
14.4.3	測試備援程式 (demo_rdn_4).....	228
第 15 章	排程控制 (SCHEDULE CONTROL).....	235
15.1	安裝 Schedule-Control Utility 與回存 Win-GRAF 範例程式	235
15.2	Win-GRAF 範例程式說明	236
15.2.1	"I/O Boards" 設定 (Schedule)	237
15.2.2	開放變數資料 (Data Binding).....	238
15.3	Schedule-Control Utility 範例說明	239
15.4	測試 PAC 內的排程控制	241
15.5	如何使用 Schedule-Control Utility	243
15.5.1	BOOL、DINT、REAL 控制變數的位址	243
15.5.2	設定 Target	244

15.5.3	設定 Season	246
15.5.4	設定 Normal Day / Holiday / Special Day	248
15.5.5	設定 Schedule 時段	250
15.5.6	儲存或傳送設定檔	251
15.5.7	時間同步 (Time Synchronization).....	252
第 16 章	WIN-GRAF 簡訊功能	253
16.1	"GSM_Open"、"Send_SMS" 與 "Read_SMS" 函式.....	254
16.2	程式介紹 (Demo_SMS).....	257
16.3	簡訊傳送測試.....	260
附錄 A	資料型態與數值範圍	264
附錄 B	錯誤訊息排除	265
附錄 C	讓 WIN-GRAF WORKBENCH 透過 PAC 的序列埠連線	267
附錄 D	PAC 上序列埠的接腳圖	271

第 1 章 軟體安裝與硬體設定

1.1 安裝 Win-GRAF 軟體

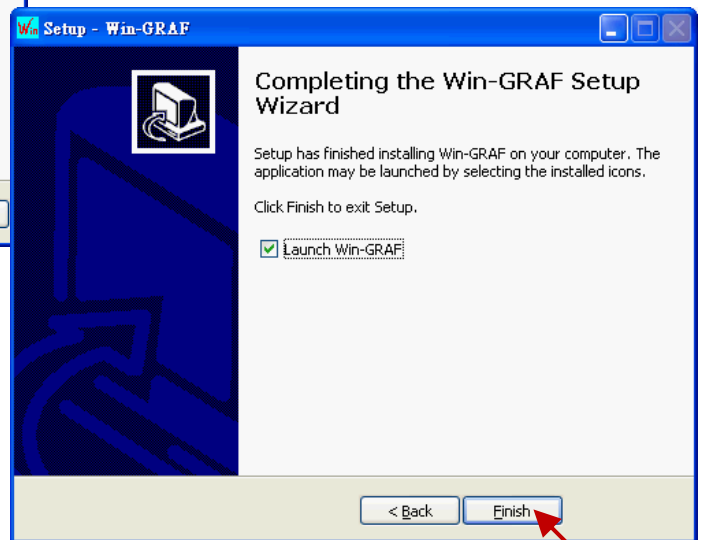
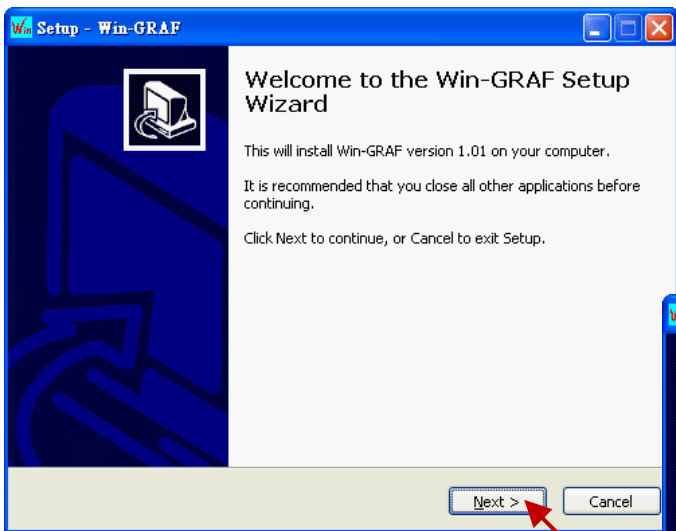
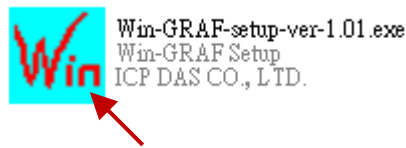
安裝 Win-GRAF 軟體前，請先確認您電腦中的安裝環境。

系統需求:

- 作業系統: Windows 7, Windows 8, Windows 10 (32-bit 或 64-bit)
- Microsoft .Net Framework 3.5 (可在微軟官方網站下載:
<https://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. 請於 ICP DAS 網站下載 Win-GRAF Workbench 檔案。
(<https://www.icpdas.com/en/download/show.php?num=711>, Win-GRAF-setup-ver-1.xx.zip)
2. 滑鼠雙擊 “Win-GRAF-setup-ver-x.xx.exe” 檔，並依照畫面指示安裝。



1.2 開啟 Win-GRAF 軟體

開啟軟體前，請確認您的 PC 已裝上 USB 保護鎖 (Win-GRAF Dongle)，否則將以 Demo Mode 來運行 Win-GRAF Workbench。

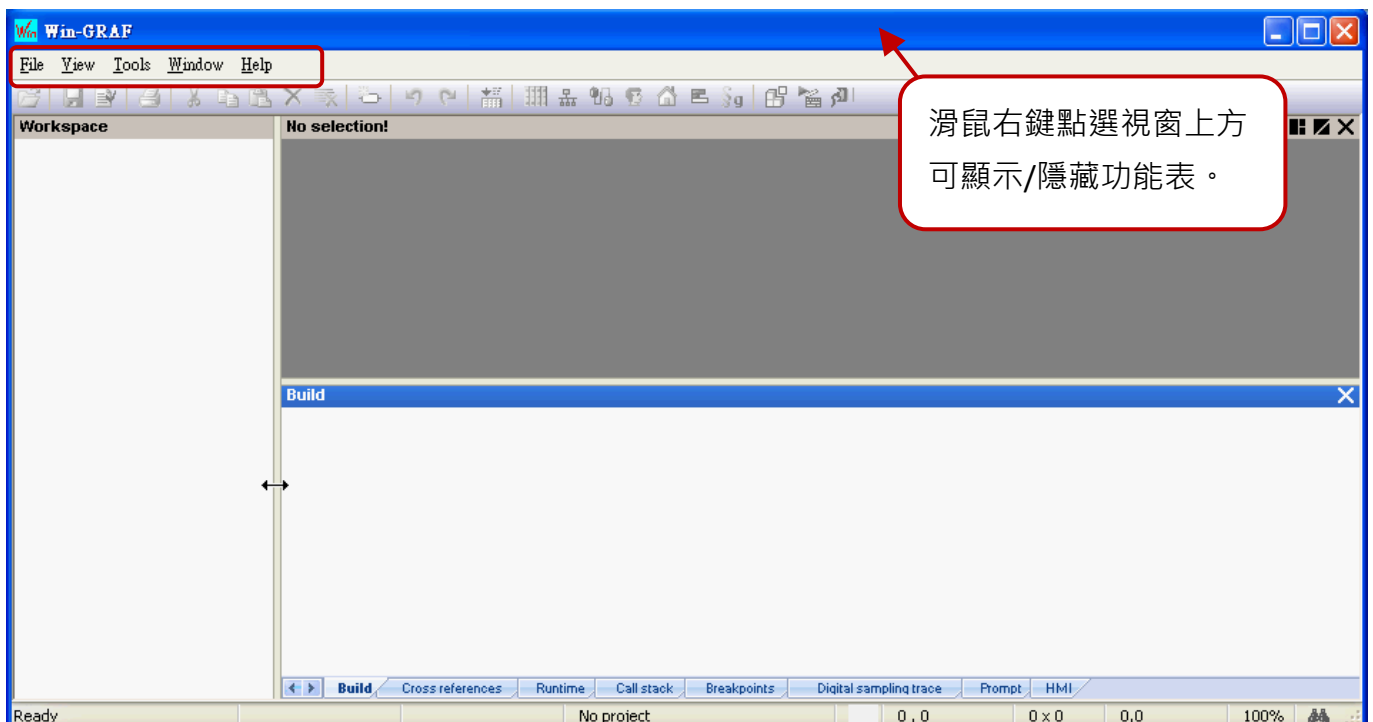
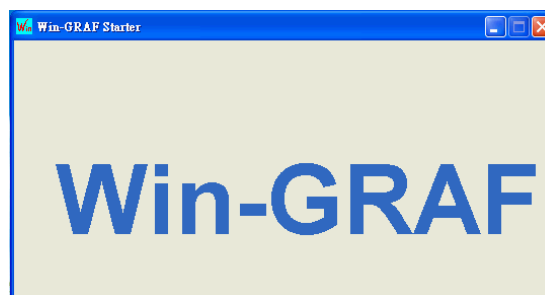
Demo Mode 使用限制:

1. 支援 40 個 I/O 點。
2. 編譯好的程式，可在 PAC 內運行 15 分鐘。
3. 可執行模擬功能 15 分鐘。

您可在“開始”選單內，點選“Win-GRAF”來開啟軟體。

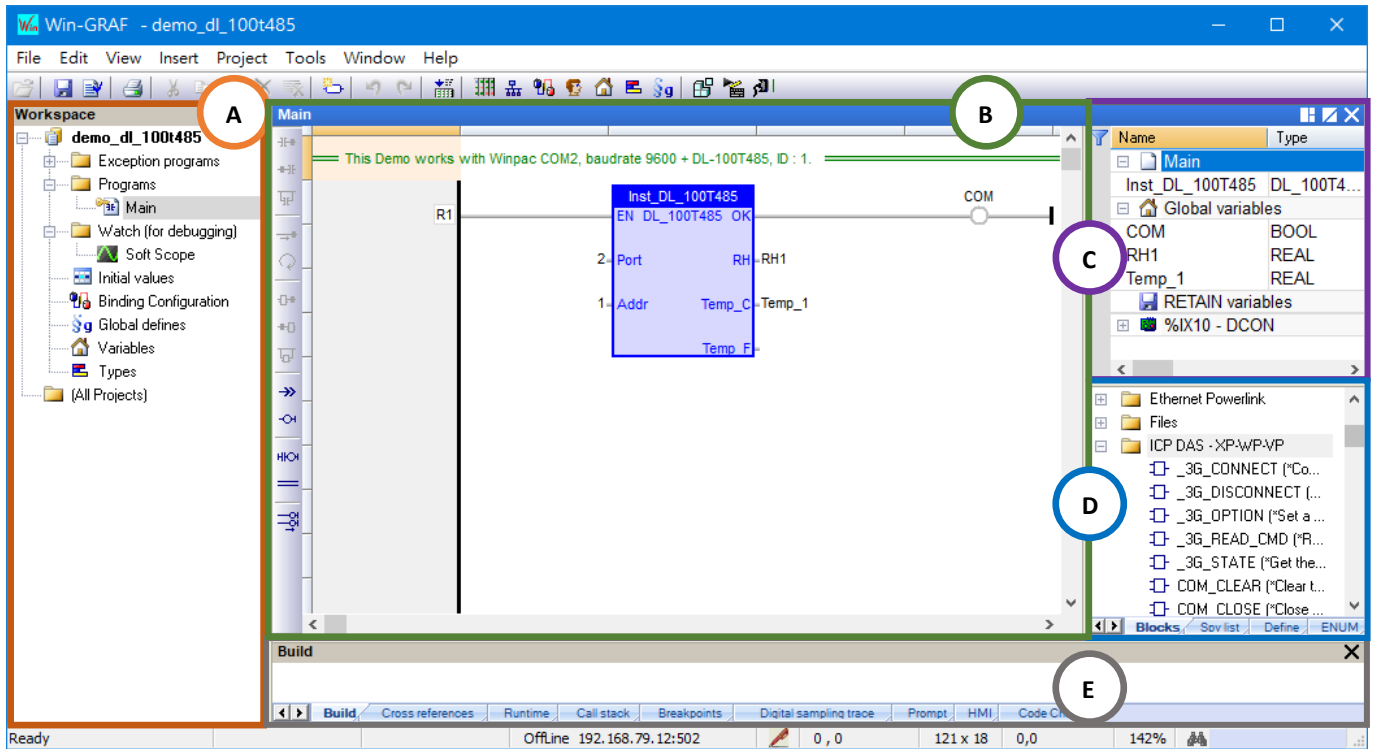


註: 若開啟 Win-GRAF 後才裝上 USB 保護鎖，請重新開啟軟體來進入完整版。



1.2.1 軟體介面說明

Win-GRAF Workbench 主要的工作區域，畫面如下：



A. 工作區 (Workspace):

Workbench 可在同一個工作區中，編輯多個專案。每個專案與相關資訊會存在磁碟上的資料夾，而專案清單會存在副檔名為 ".W5L" 的檔案，此檔案只儲存專案資料夾的清單與一些設定資料。

B. 編輯區:

滑鼠雙擊 Workspace 列的項目來開啟編輯頁面，例如: Program 與 Variables。

C. 變數區:

在開發專案期間，可快速宣告變數 或 實體變數 (Instance)。

D. 功能方塊 (Block): 可在程式中使用的 Function 與 Function Block。

觀測清單 (Spy list): 在 Debug 期間，可快速檢視變數的動態。

E. 輸出視窗:

用來顯示訊息，並提供多個診斷工具。

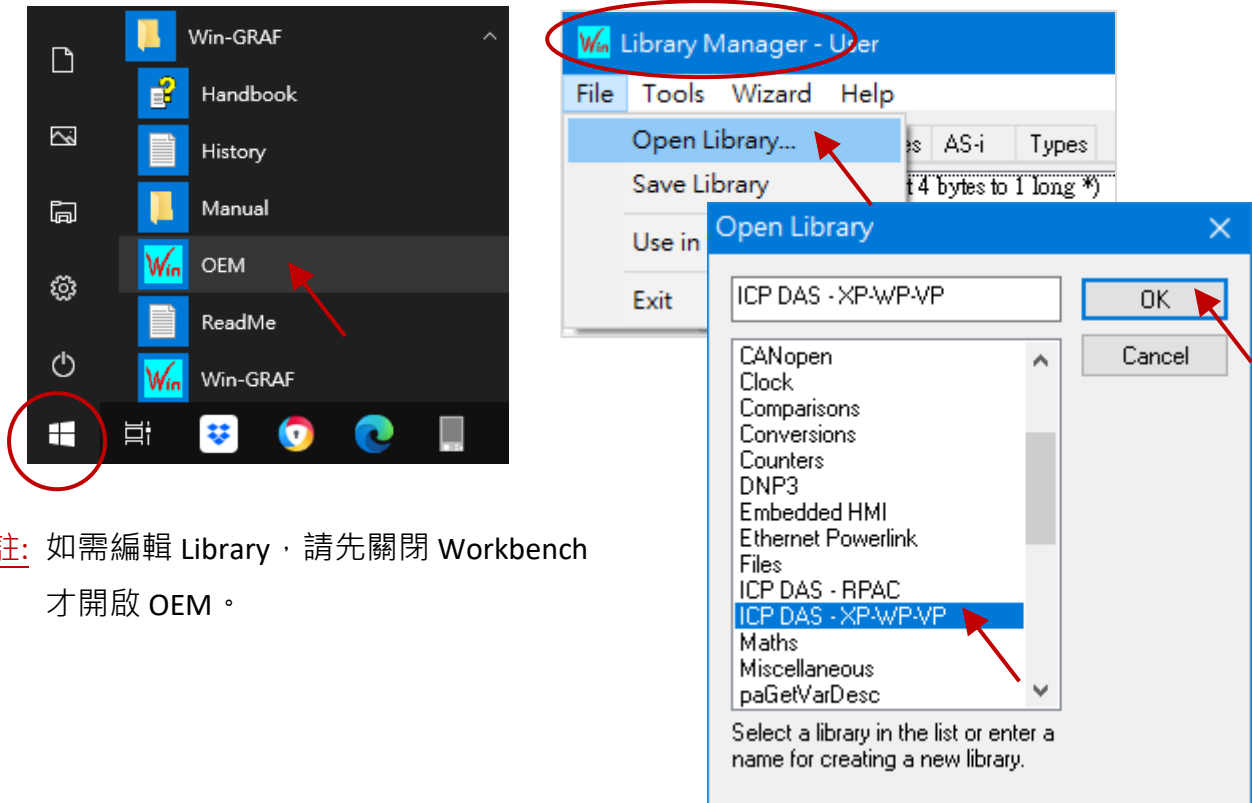
使用小技巧:

1. 可在工作區上按 “F1” 鍵，開啟對應的使用說明。
2. 可使用滑鼠拖曳的方式，任意調整視窗的大小。
3. 點選功能表 “View - Output / Infos Tab1 / Infos Tab2”，若該窗格不小心被關閉。
4. 點選功能表 “Help - Language” 可切換語系，Workbench 會自動重新開啟，並套用設定。

1.2.2 函式庫管理員 (Library Manager) 功能

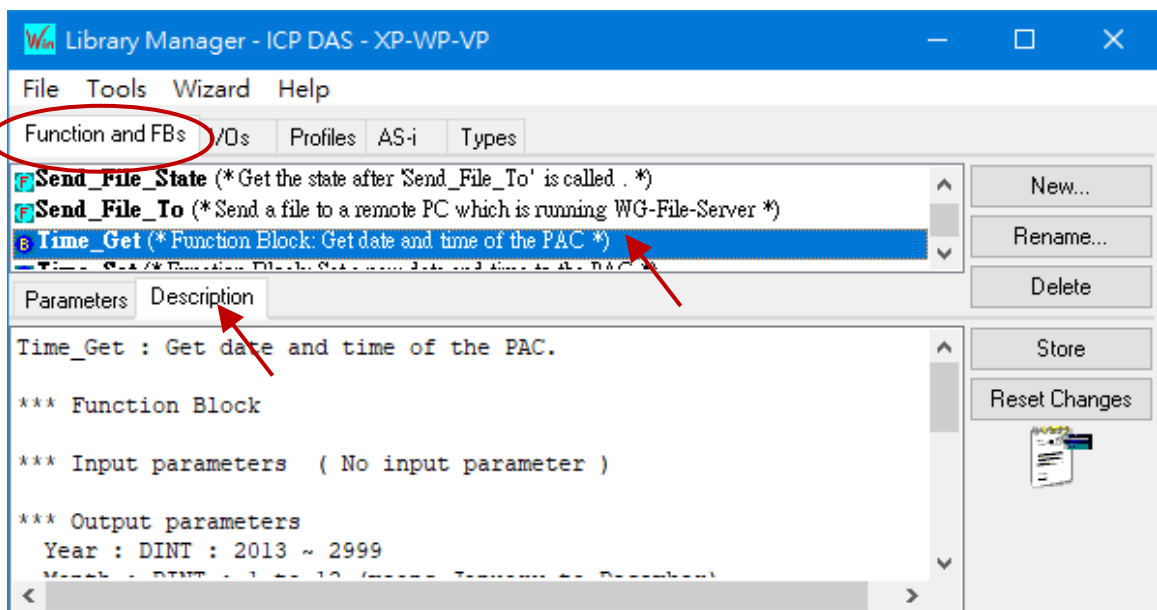
欲查詢 Function 與 Function Block 的說明，請開啟 Win-GRAF 的函式庫管理員 (Library Manager)。

1. 點選“開始”選單，再點選 "Win-GRAF" 目錄內的 "OEM"。
2. 在“Library Manager”視窗點選“File - Open Library”，接著點選“ICP DAS – XP-WP-VP”或“ICP DAS – RPAC”再點選“OK”。



註: 如需編輯 Library，請先關閉 Workbench 才開啟 OEM。

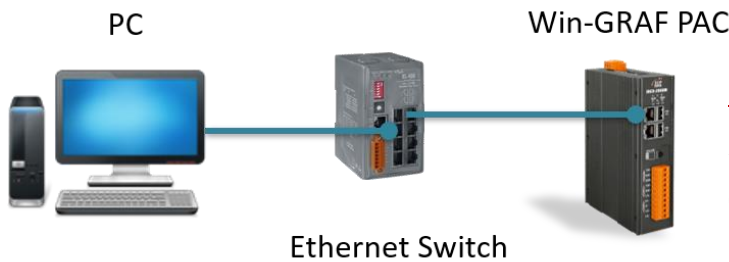
3. 點選“Function and FBs”頁籤下的任一項目，再點選“Description”來查看該 Function/Function Block 的說明。



1.3 設定 Win-GRAF PAC 的 IP 位址

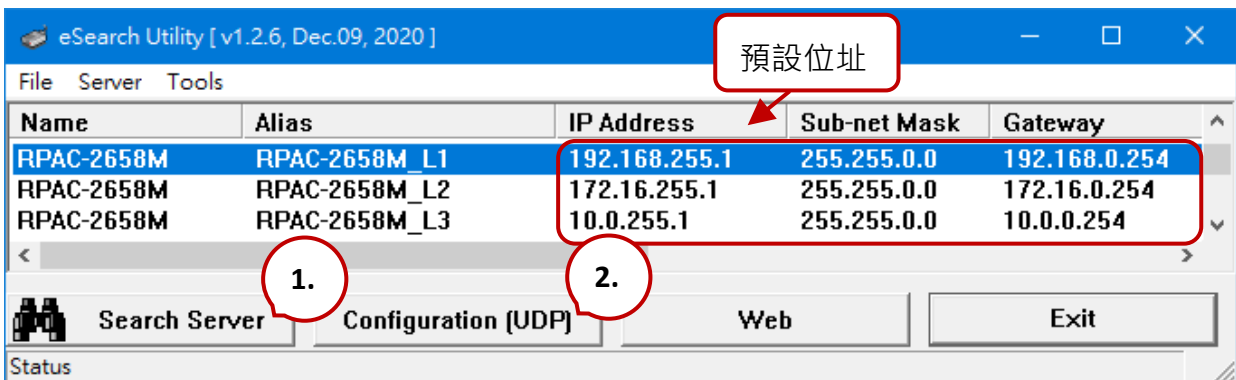
請依照以下步驟來設定 RPAC-2658M 的 IP、Mask 與 Gateway 位址。

硬體連接圖

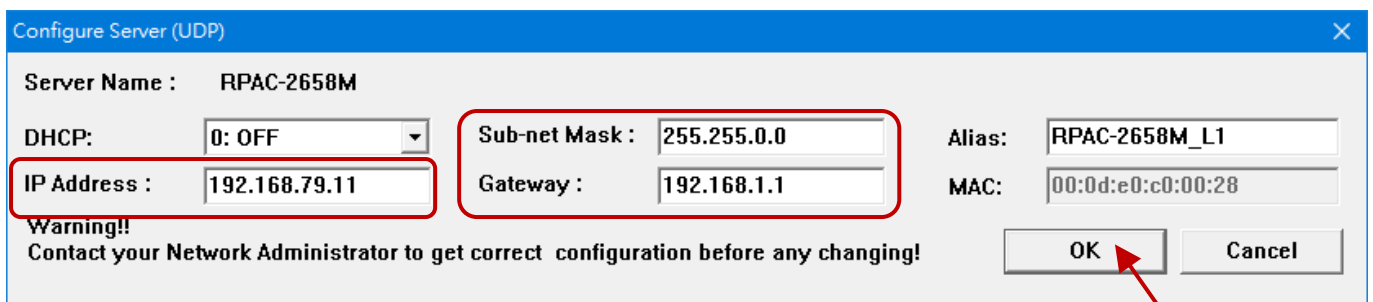


注意:
建議使用 LAN1 與 PC 軟體 (例如: Win-GRAF) 連線，且需設定相同的網段。

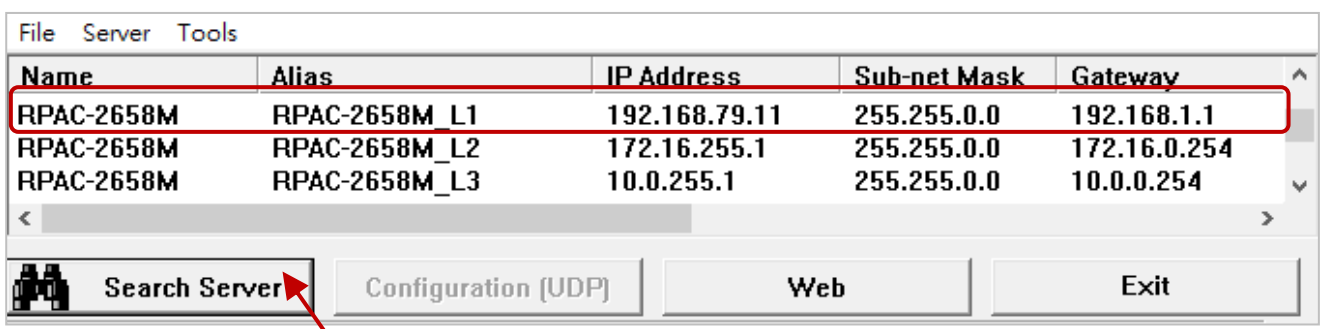
1. 請在 [ICP DAS 網站](#) 下載並安裝 “eSearch Utility” 軟體。
2. 點選 Search Server 按鈕搜尋網路上的設備，一台 RPAC 會顯示 3 個 LAN 設定。



3. 點選 Configuration 按鈕，並設定以下欄位，再點選 OK 按鈕。

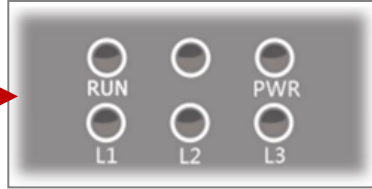


4. 請再重新搜尋一次，以更新狀態。**註:** LAN1, LAN2 與 LAN3 需設定為不同的網段。



1.4 硬體說明

LED 指示燈



LED 指示燈	顏色	說明
RUN	綠	開機 或 運行 OS (閃爍狀態)
PWR	紅	電源為 ON
L1	綠	使用者可程式 LED 燈 與 硬體錯誤指示燈
L2	橘	使用者可程式 LED 燈
L3	紅	備援狀態指示燈

旋轉開關 (Rotary Switch)



旋轉開關	說明
0	正常模式
1	不執行用戶的應用程式
2	LAN1: DHCP 模式
3	LAN1: 靜態 IP 模式 (Static IP) (IP: 192.168.255.1, Mask: 255.255.0.0)
4	更新韌體 (Firmware) 模式

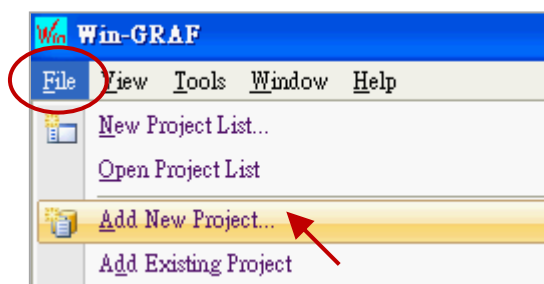
第 2 章 編寫一個簡單的 Win-GRAF 範例

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。

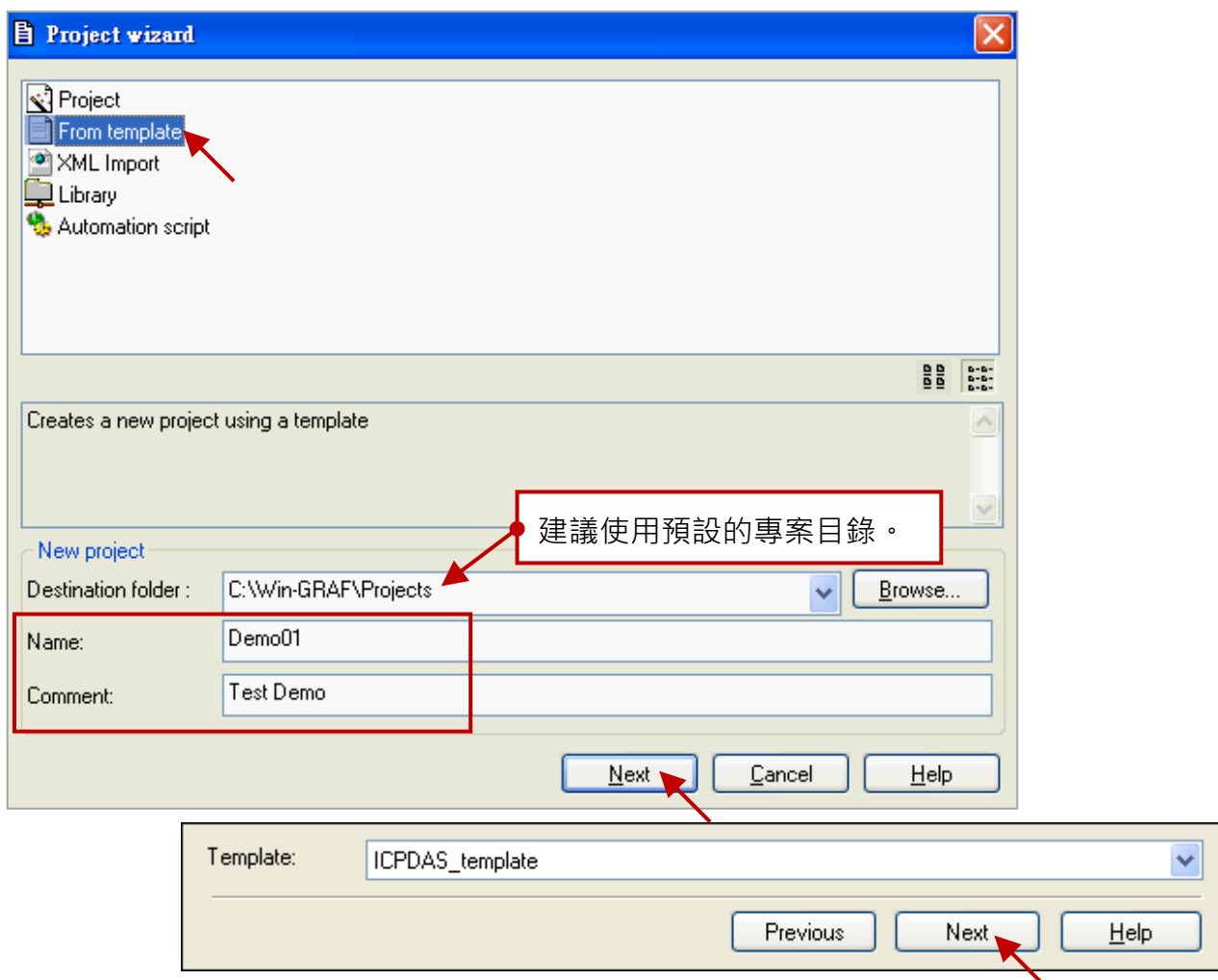
2.1 建立 Win-GRAF 專案

2.1.1 建立樣版專案 (Demo01)

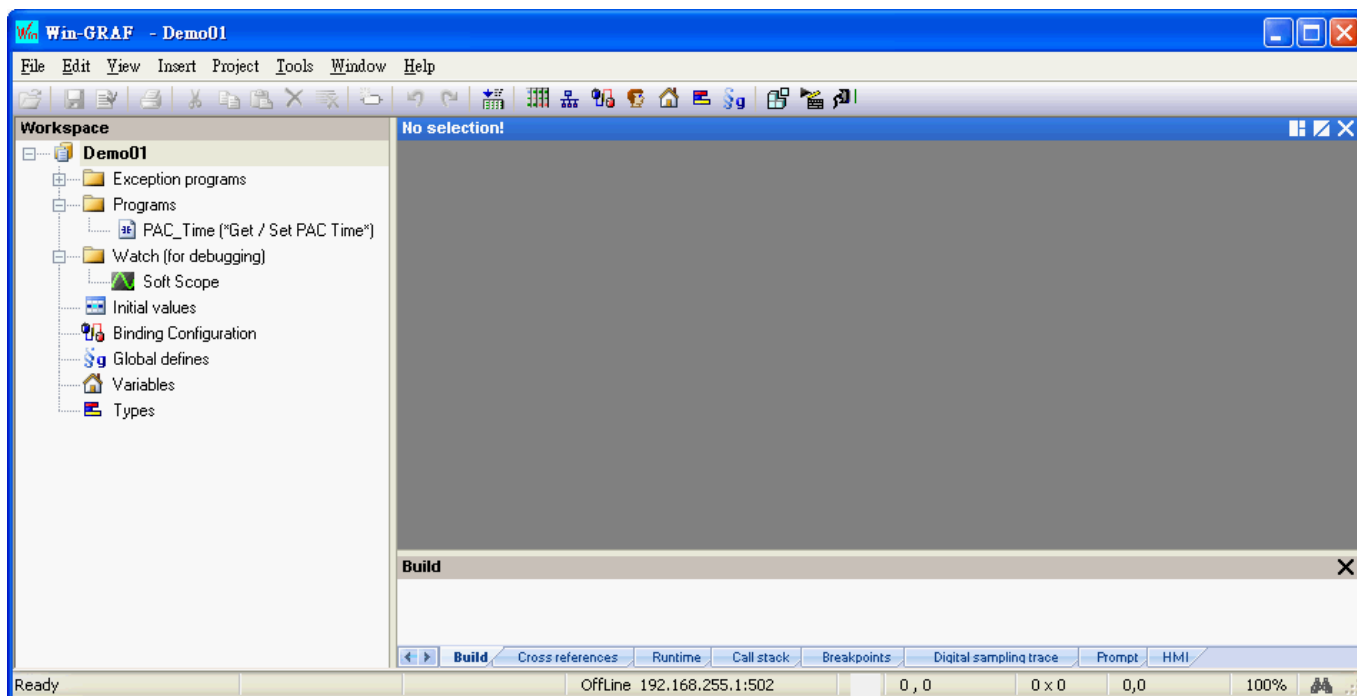
1. 開啟 Win-GRAF 軟體，並點選功能表 "File - Add New Project..."。



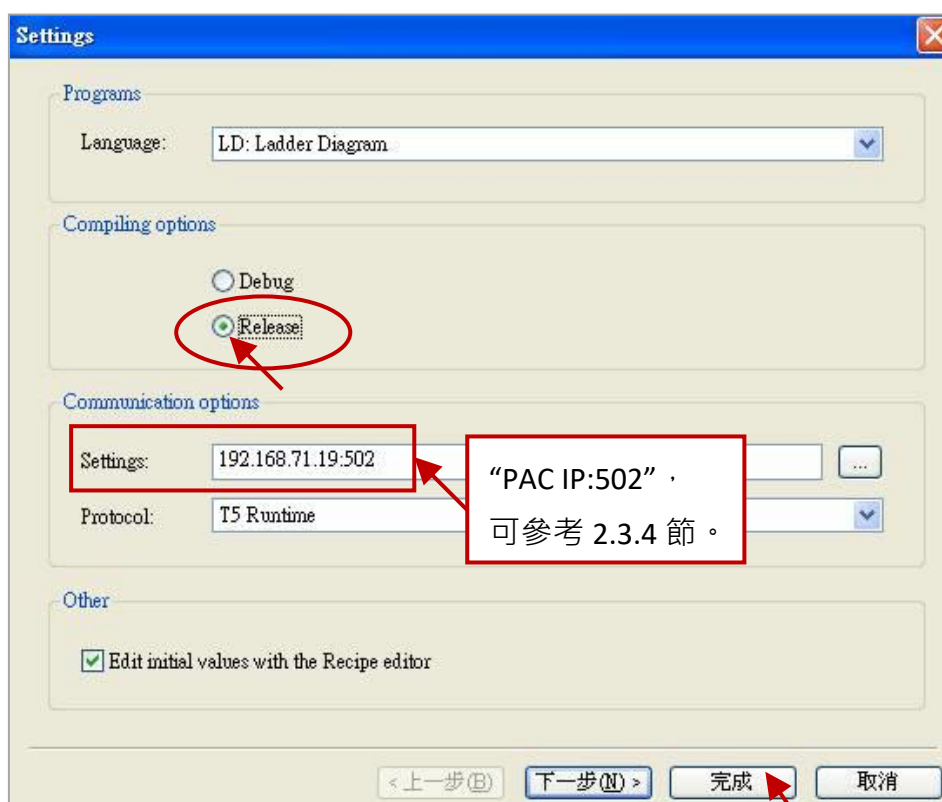
2. 點選 "From template" 來建立一個樣版專案，在 "Name" 欄位中輸入專案名稱 (例如: Demo01)，"Comment" 欄位可輸入註解，點選 "Next" 並選取 Win-GRAF 提供的樣版 "ICPDAS_template"，再點選 "Next" 按鈕。



3. 您已建立一個 “Demo01” 樣版專案。



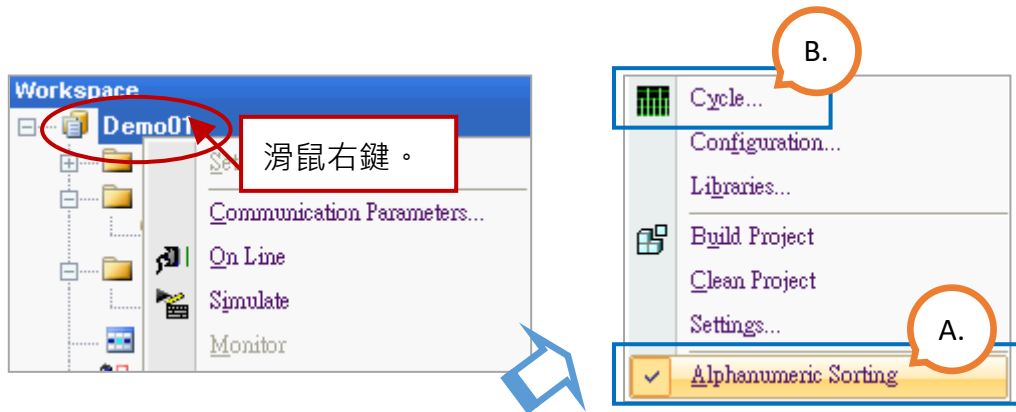
注意: 若您在步驟 2，改用 "Project" 方式來建立專案，請設定 “Compiling options” 為 "Release"，並點選 "完成"，其餘項目可在後續章節中進行設定。



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。

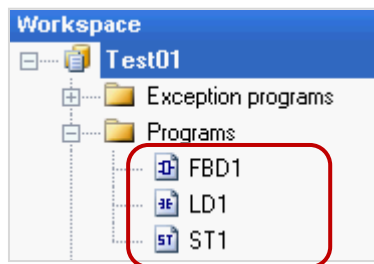
1. 程式的執行順序:



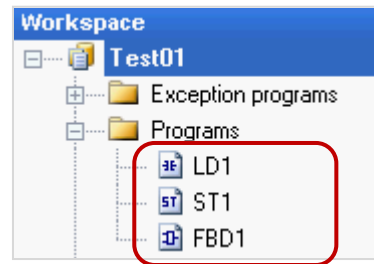
A. 查看程式的執行順序

滑鼠右鍵點選專案名稱 (例如: Demo01)，勾選 "Alphanumeric Sorting" 表示依**字母順序**來排列程式；若取消勾選，表示依**執行順序**來排列程式。

(勾選，依字母順序)

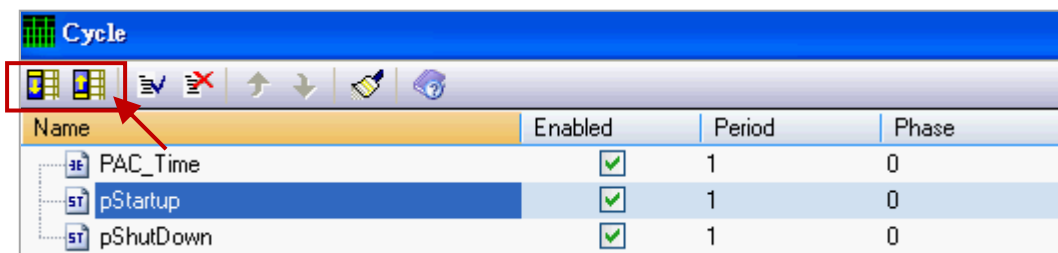


(不勾選，依執行順序)



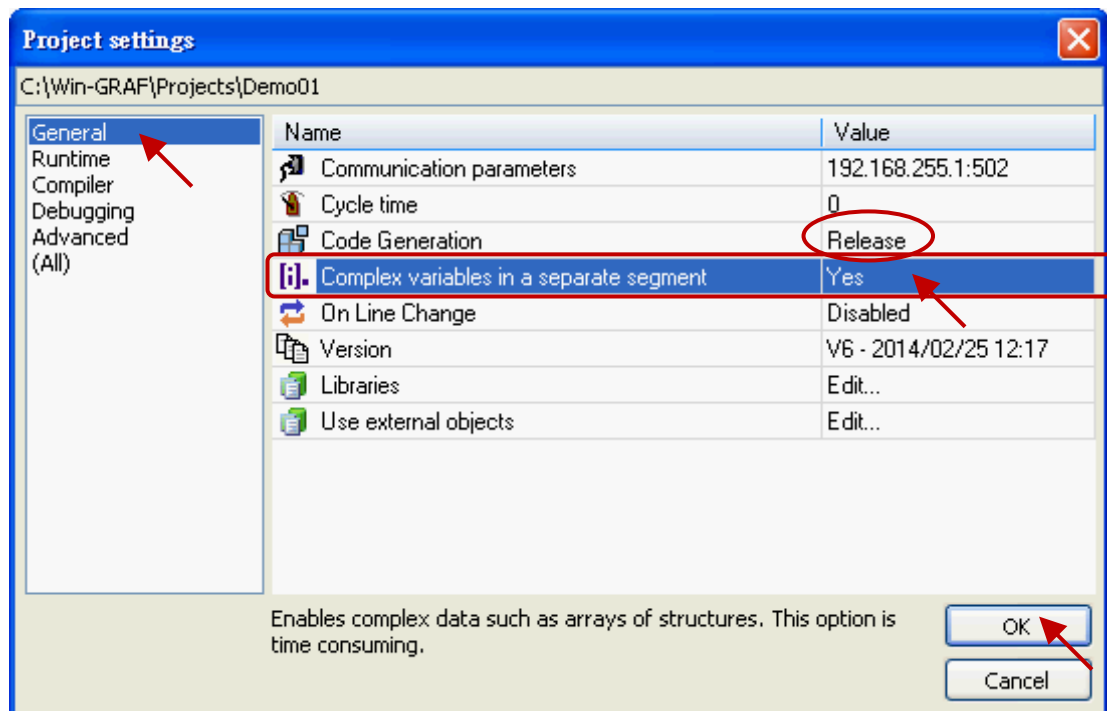
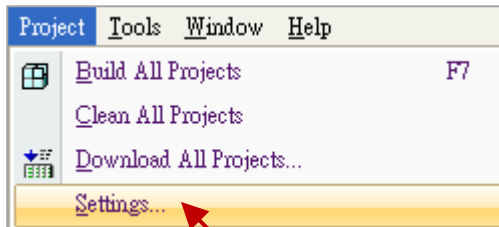
B. 修改程式的執行順序:

滑鼠右鍵點選專案名稱 (例如: Demo01)，並點選 "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. 啟用 "Complex variables in a separate segment" 選項

若專案中會使用複合變數 (Array、Structure 與 FB Instance)，請點選功能表 "Project - Settings..." 開啟 "Project settings" 視窗，接著點選 "General" 選項，並將 "Complex variables in a separate segment" 設定為 "Yes"，再點選 "OK"。



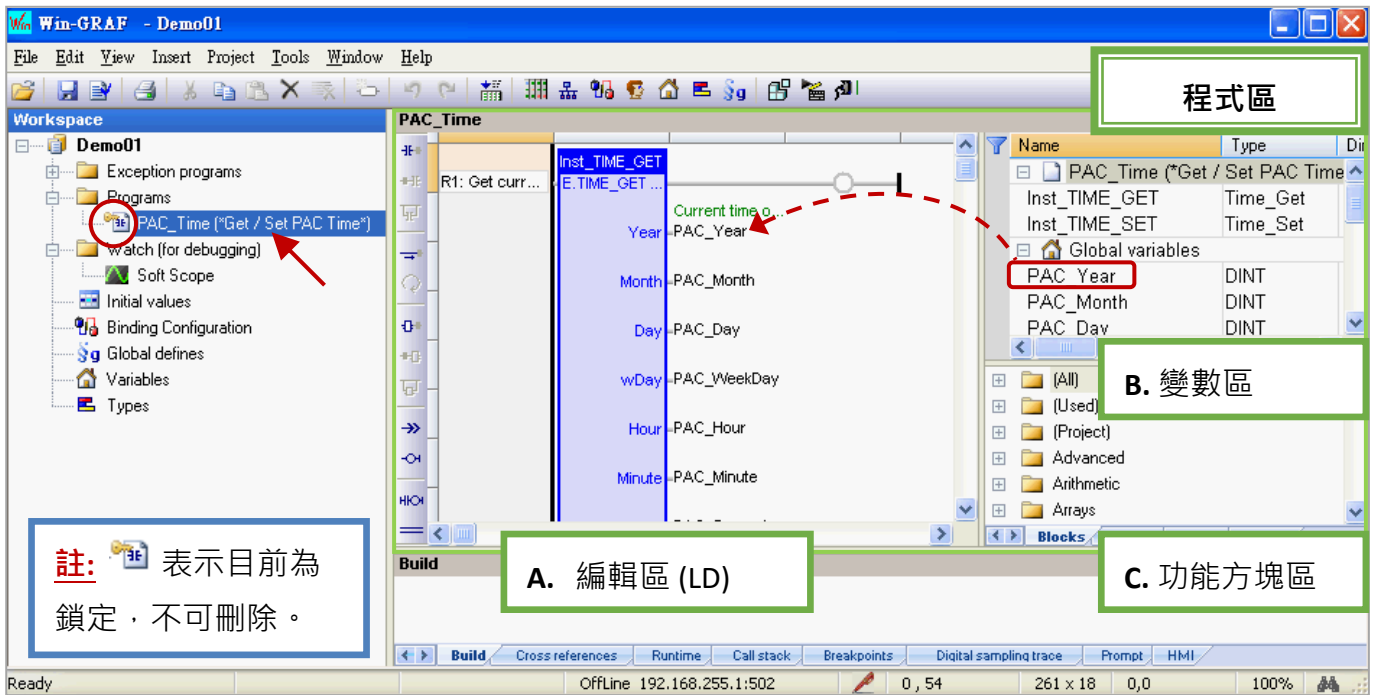
註:

1. 可點選功能表 "Help - Topics" 並搜尋 "Complex variables in a separate segment" 取得詳細說明。
2. 請確認 **Code Generation** 設為 **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：

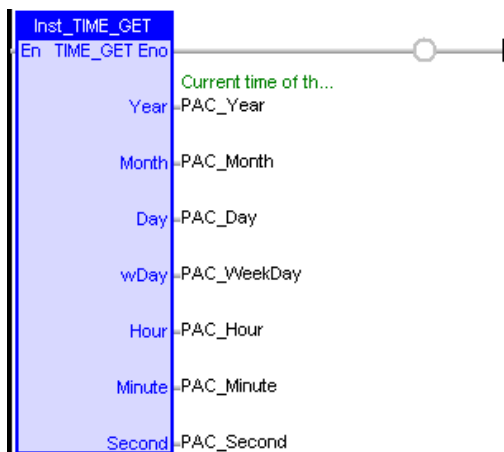


使用小技巧： 滑鼠點選一下編輯區，再按 “+” 或 “-” 鍵，可放大或縮小程式內容。

A. 編輯區：

此區塊用來編輯程式，您可點選左邊的元件按鈕來進行編程，也可將變數區的變數拖曳到程式內使用。

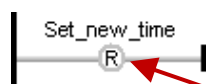
R1: 讀取 PAC 目前時間



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



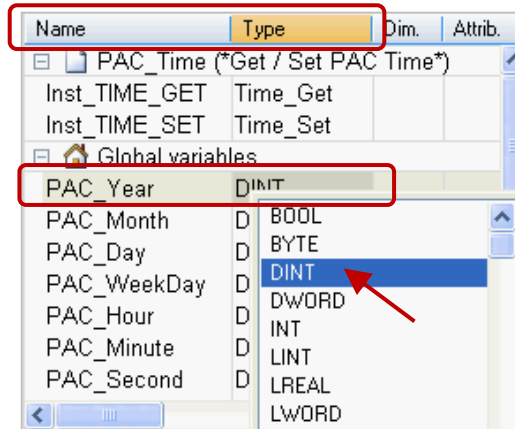
R3: 重置為 “False” 狀態



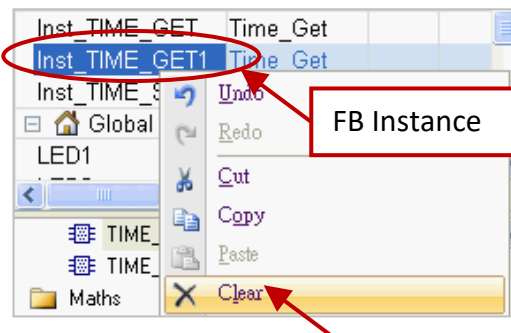
註： 您可點選空白鍵來變更其型態 (/, S, R) · 按 F1 鍵可取得詳細說明。

B. 變數區:

此區塊顯示了程式中使用的 FB Instance 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。

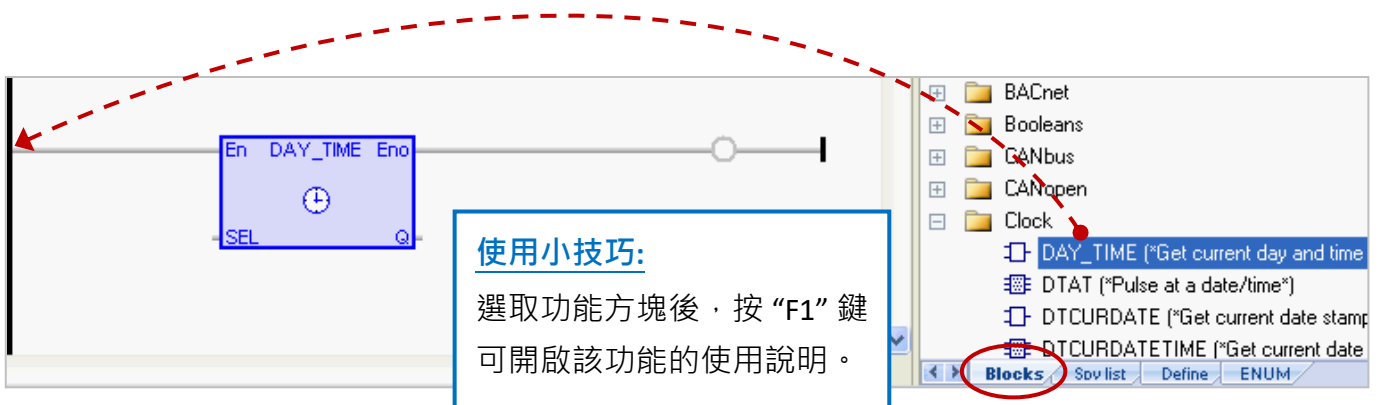


註: 為了在程式內正確的使用功能方塊，每增加一個功能方塊 (FB) 都會自動建立一個 Instance 名稱 “Inst_xxx..”，而為了安全考量，若刪除程式的功能方塊時，並不會自動刪除 FB Instance。如需要，使用者可用滑鼠右鍵點選該名稱再選擇 “Clear” 來手動刪除它。



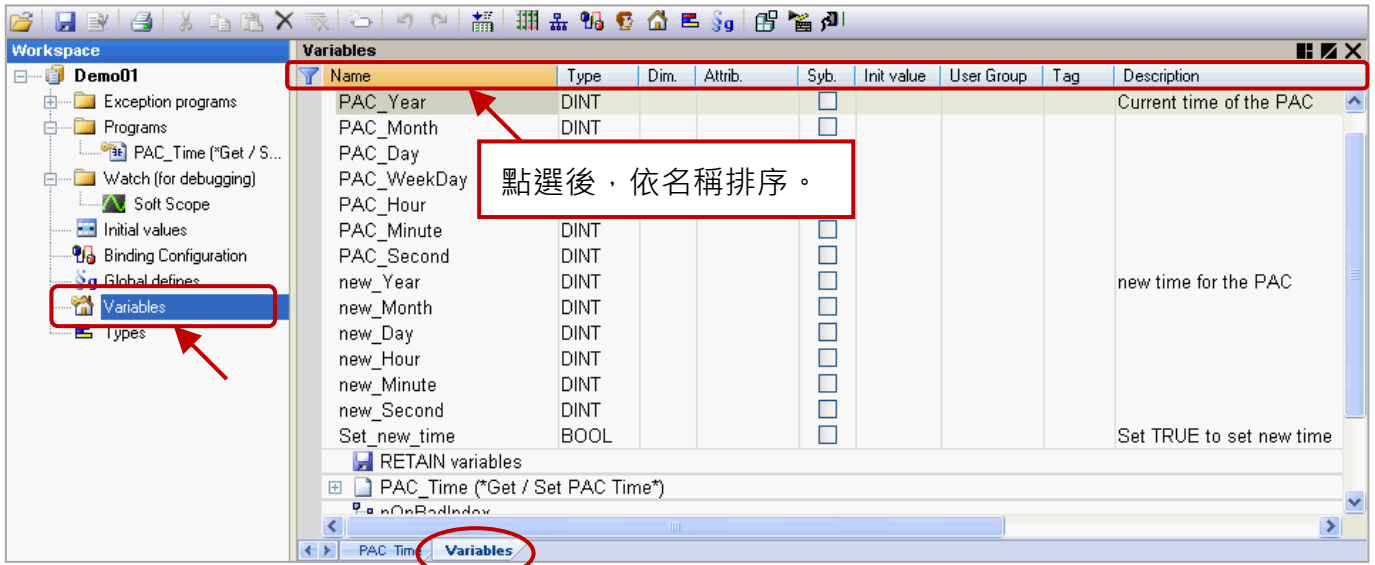
C. 功能方塊區:

在 “Blocks” 頁籤內，您可將功能方塊拖曳到程式內進行編輯。

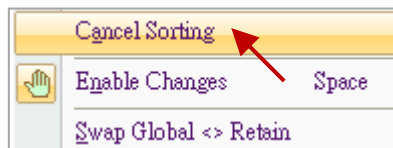


2.2.2 Demo01 - 變數說明

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，可點選任一欄位名稱 (例如: “Name”) 來排序資料。另外，也可滑鼠右鍵點選視窗內任一處，並點選“Cancel Sorting”恢復原有排序。



欄位說明: (可按“F1”鍵，查看詳細說明)

滑鼠雙擊欄位中的任一個項目，來設定/修改資料。

- 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.:** 此功能即將支援。選取的名稱會嵌入到應用程式中，可在 PAC 呼叫名稱來存取參數的資料。
- 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 (Nickname)。
- Description:** 可輸入變數的用途說明。


2.3 小試身手

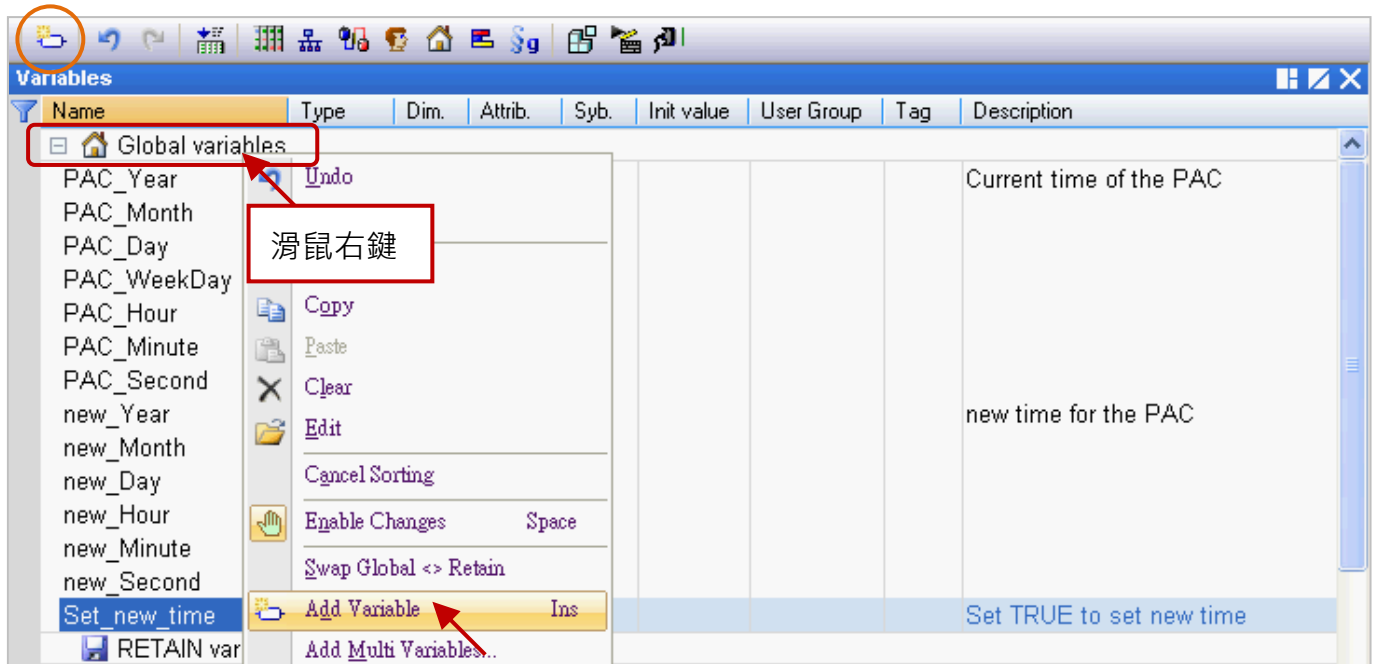
前面章節介紹了“Demo01”專案中的 LD 程式 (2.2.1 節) 與 變數說明 (2.2.2 節) ，本章節將說明如何宣告變數 與 新增一個有閃爍功能的 LD 程式。

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

2.3.1 宣告變數

首先，我們將宣告程式內的 2 個布林變數 (即，LED1 與 LED2)。

1. 在“Variables”視窗，滑鼠右鍵點選“Global variables”內任一處並選擇“Add Variable”來新增一個變數。(也可點選  按鈕或按“Ins”鍵)



2. 滑鼠雙擊“NewVar”並修改名稱為“LED1”，再按“Enter”完成設定。(資料型態為“BOOL”。)

Set_new_time	BOOL			<input type="checkbox"/>				Set TRUE to set new time
NewVar	BOOL			<input type="checkbox"/>				
LED1								

注意：所有的設定都必須按“Enter”才能完成。

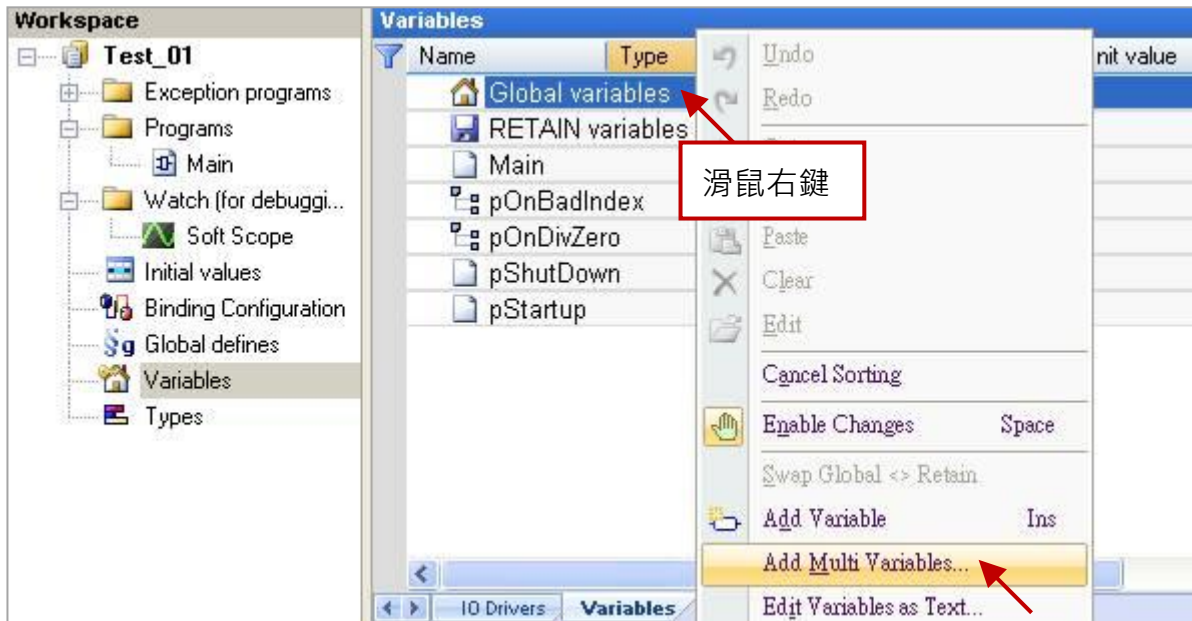
3. 如同上述步驟，請再新增“LED2”布林變數。

Set new time	BOOL			<input type="checkbox"/>				Set TRUE to set new time
LED1	BOOL			<input type="checkbox"/>				
LED2	BOOL			<input type="checkbox"/>				

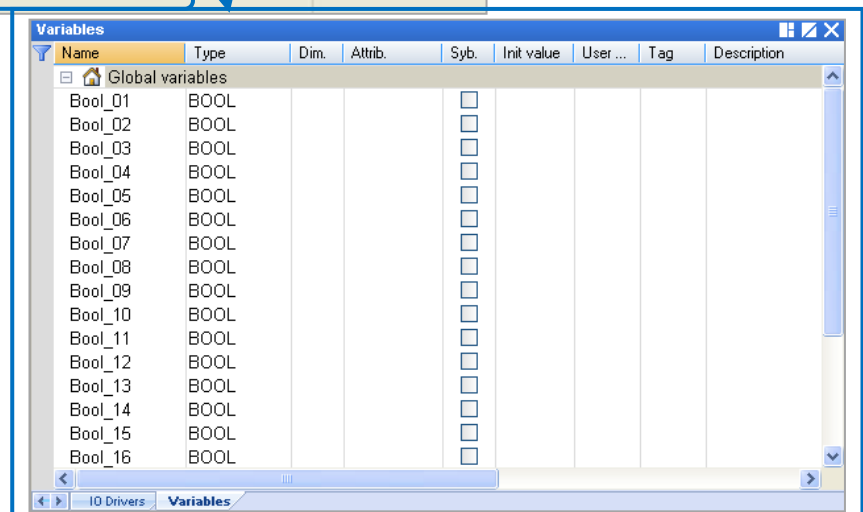
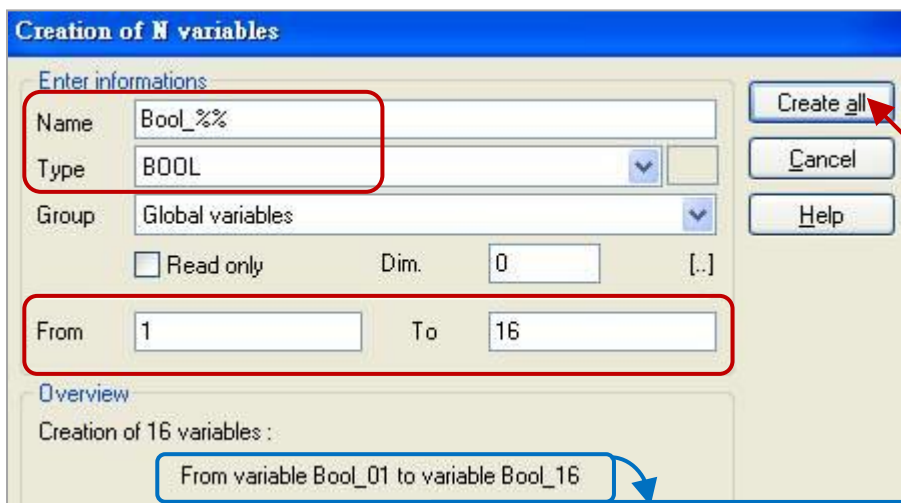
使用小技巧 1：欲新增連續順序的變數，可在步驟 2 設定名稱為“LED”，再以複製 (Ctrl+C) 並多次貼上 (Ctrl+V) 來自動產生“LED1”、“LED2”....，最後再刪除“LED”變數即可。

使用小技巧 2:

1. 欲新增多個連續順序的變數，例如：“Bool_01” ~ “Boo_16” 的布林變數。首先，滑鼠右鍵點選“Global variables” 再選擇 “Add Multi Variables”。



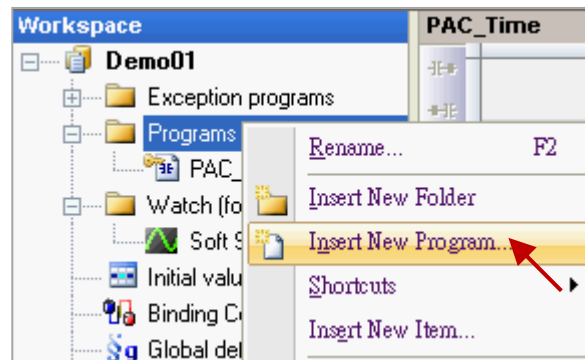
2. 請將 Name 設為 “Bool_%%”、Type 設為 “BOOL”、和 From 與 To 欄位設為 1 和 16，再點選 “Create all” 來新增變數。



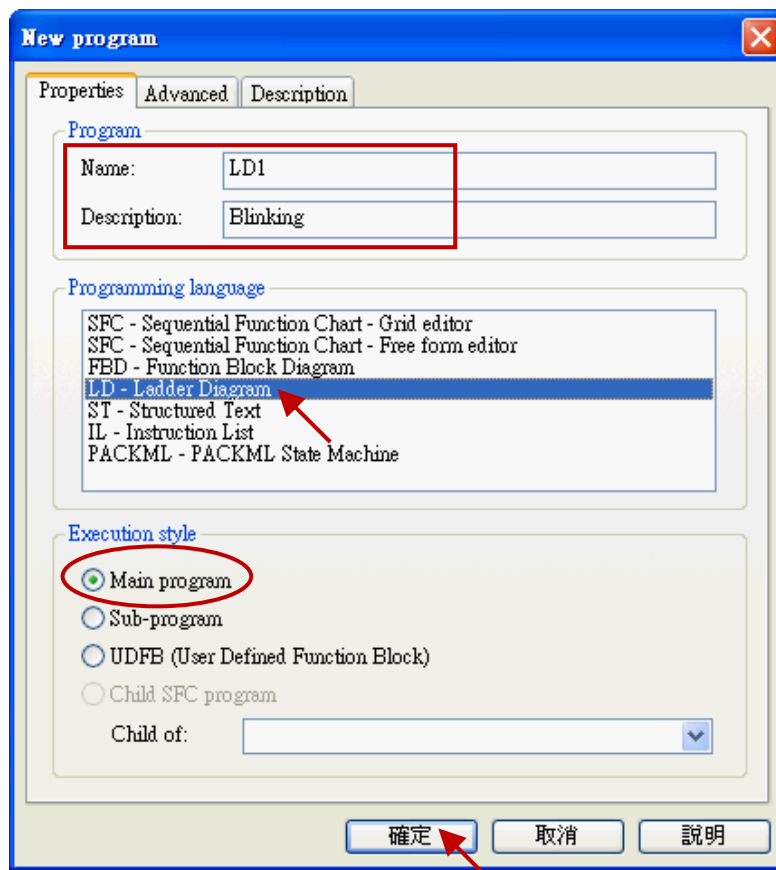
2.3.2 建立 LD 程式

此章節，我們將在“Demo01”專案中新增一個具有閃爍功能的 LD 程式。請依照以下步驟開始：

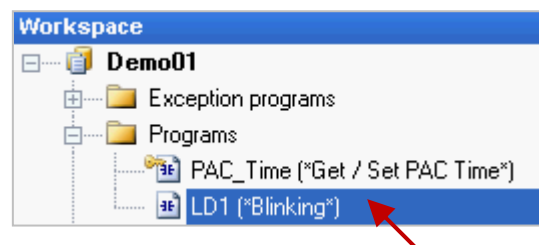
1. 工作區中，滑鼠右鍵點選“Programs”再選擇“Insert New Program...”。



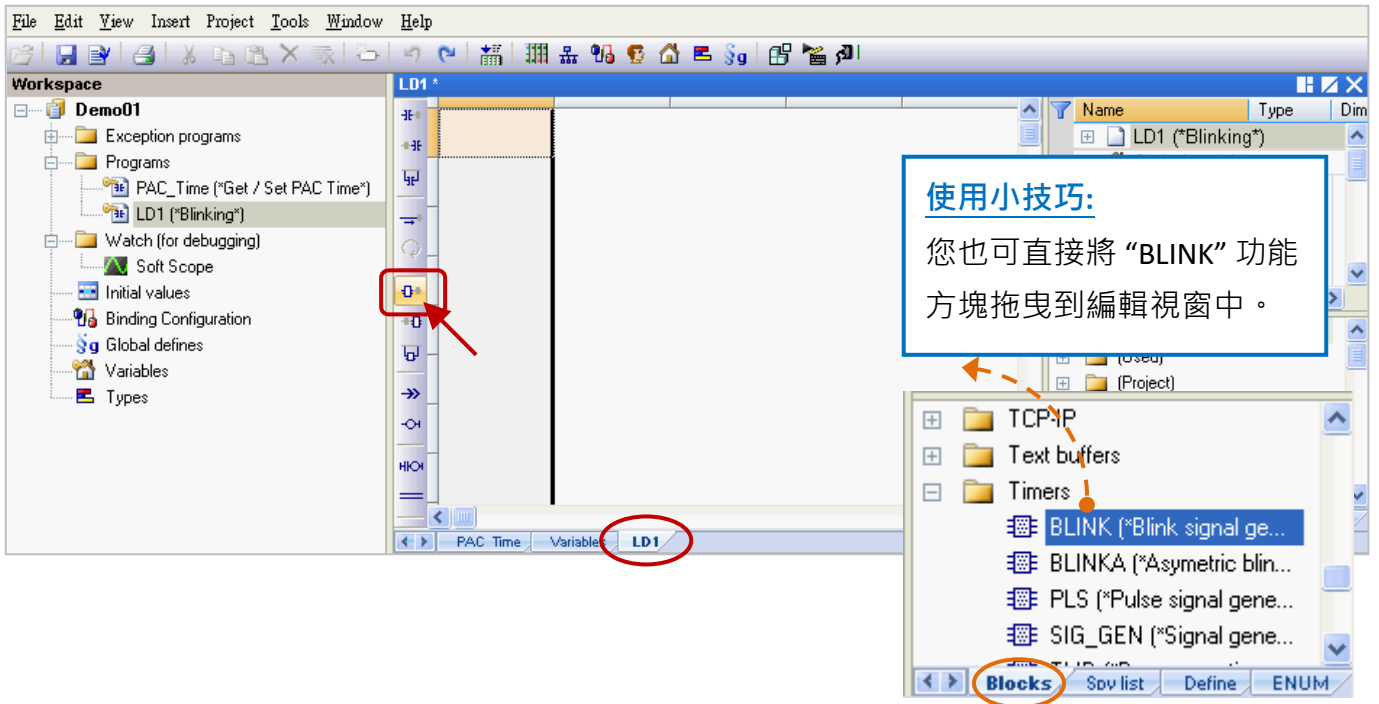
2. 於“Name”欄位中填入程式名稱，並在“Description”欄位加入簡單的描述。接著，選取編程語言“LD – Ladder Diagram”，再點選“確定”。



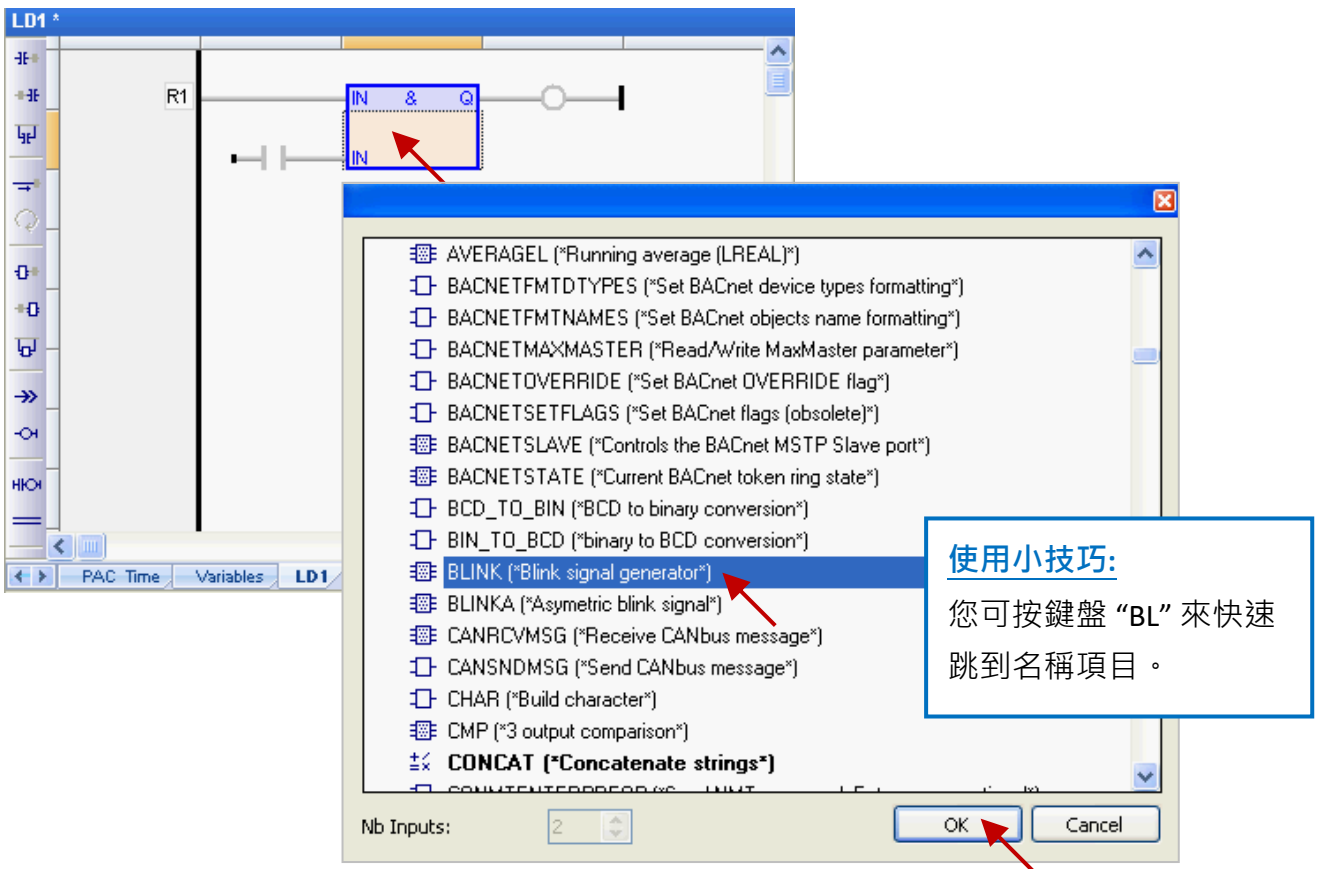
3. 滑鼠雙擊“LD1”程式來開啟編輯視窗。



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”。

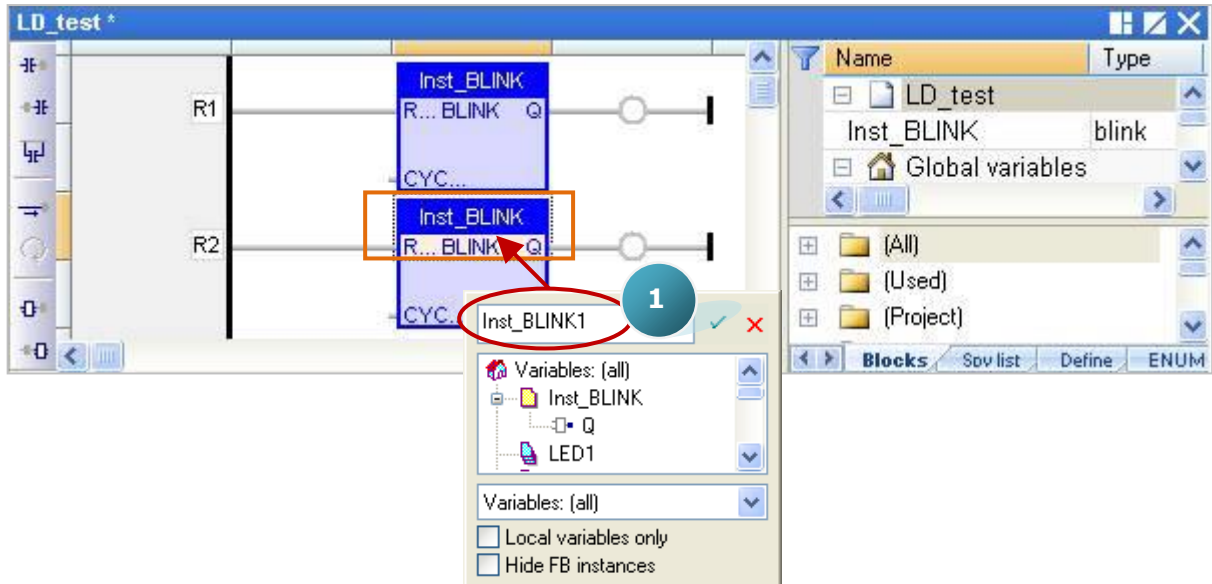




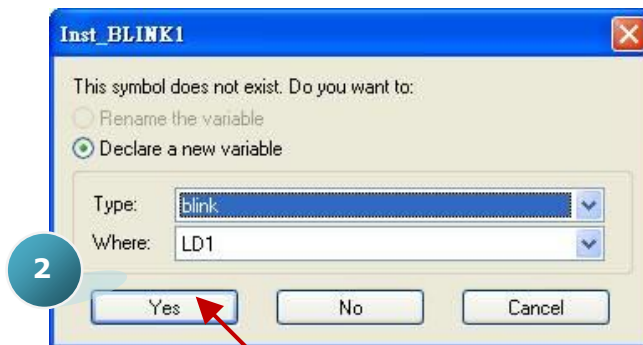
重要注意事項:

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，但這會造成 FB instance 名稱重複，而導致功能異常。因此，請為複製的 **Function Block** 建立一個新名稱。

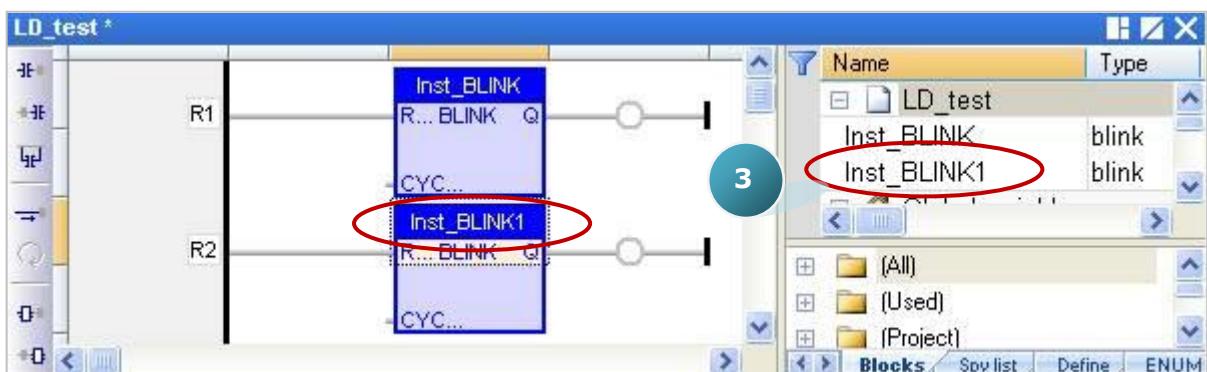
1. 滑鼠雙擊該功能方塊，並輸入新的名稱 (例如: “Inst_BLINK1”)，再點選 完成設定。



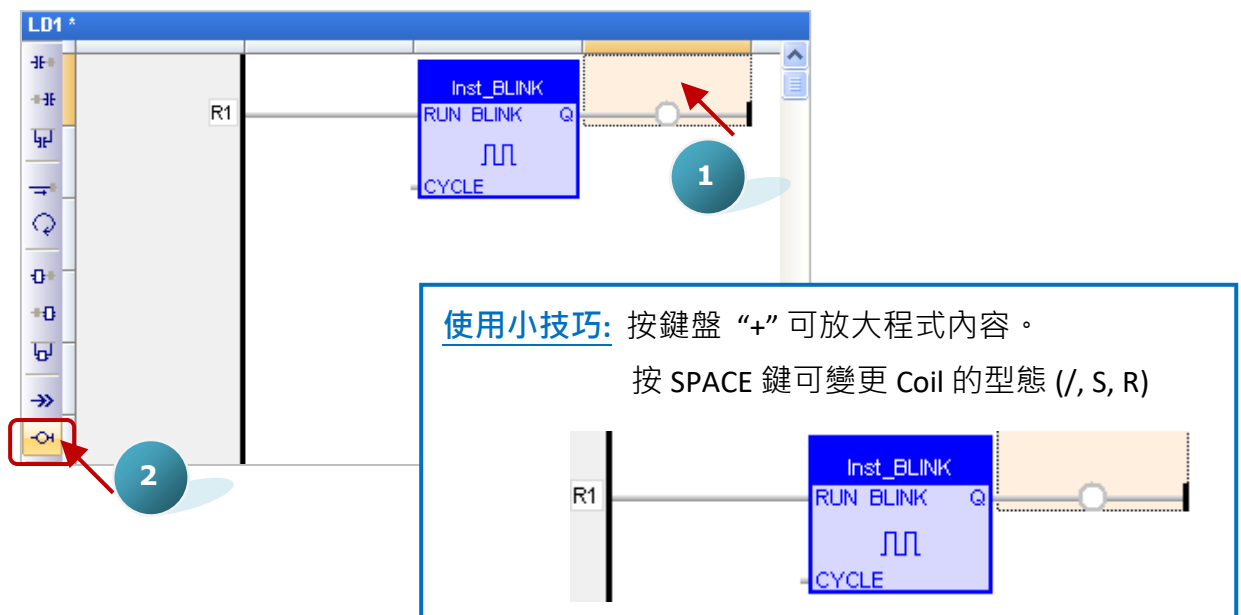
2. 於 “Inst_BLINK1” 變數視窗中，點選 “Yes” 來建立此 FB instance 名稱。



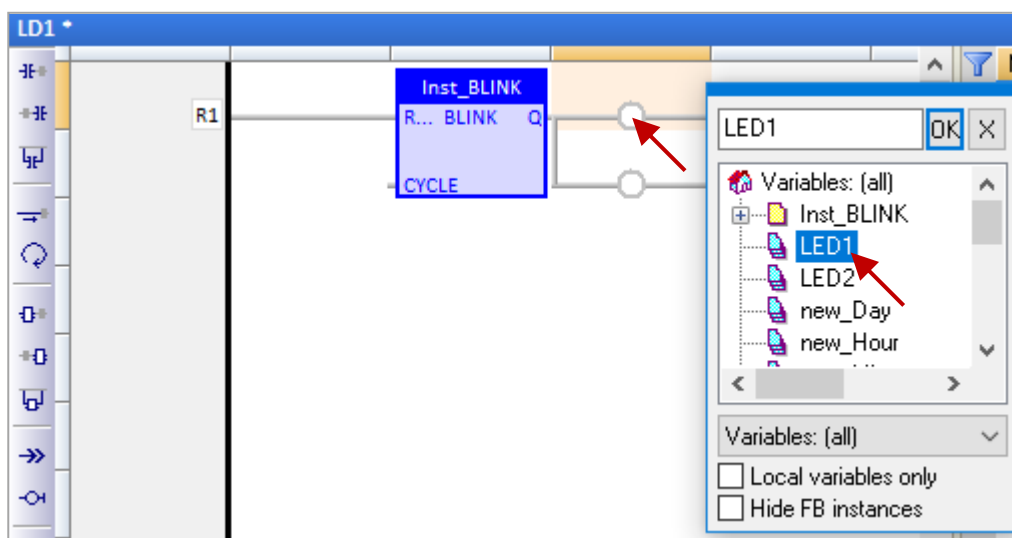
3. 變數區已建立了一個名為 “Inst_BLINK1” 的 FB instance。



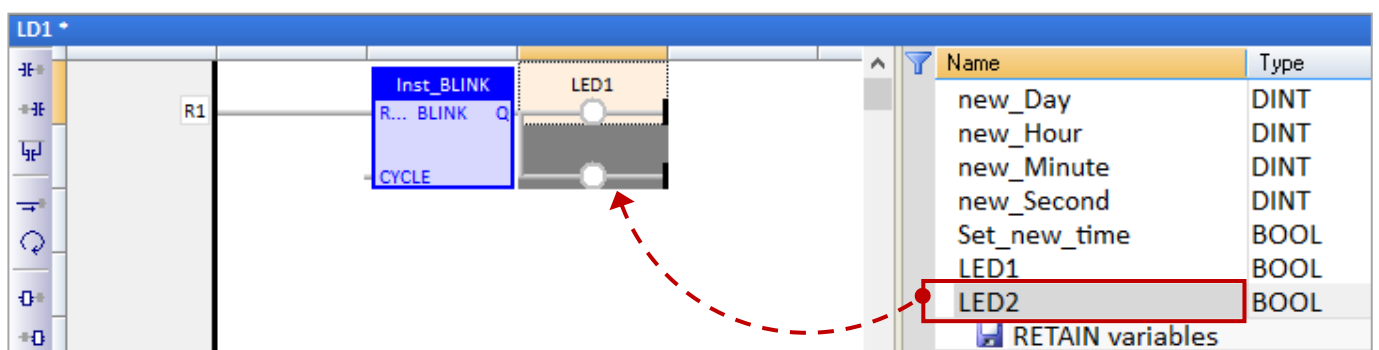
6. 點選 “BLINK” 功能方塊右方的 “Coil”，再連續點選 “Insert Coil” 按鈕來新增 2 個 “Coil”。



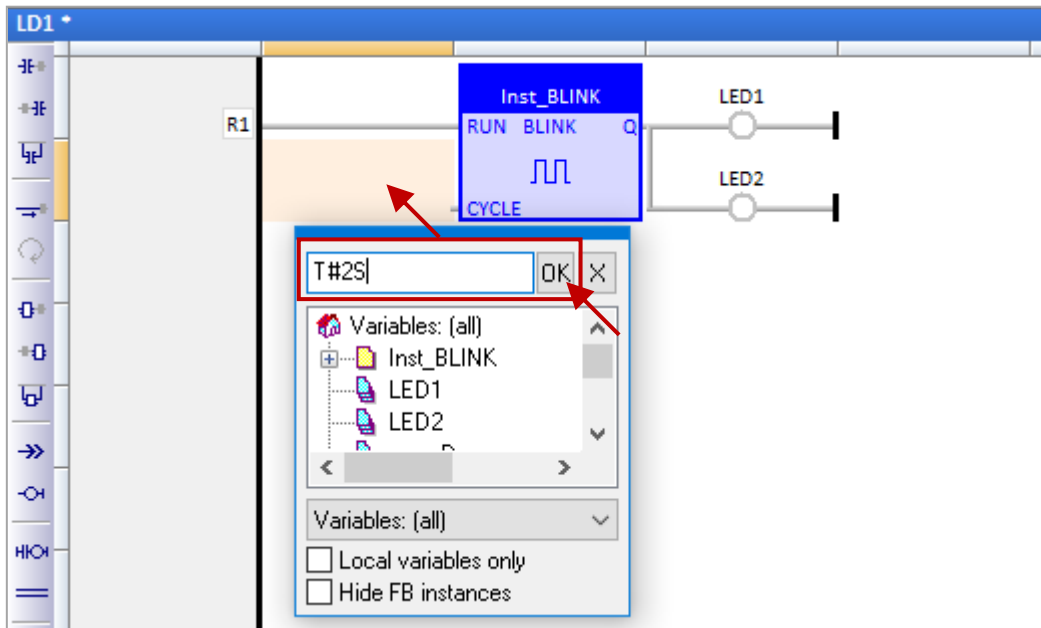
7. 滑鼠雙擊第一個 “Coil” 再雙擊 “LED1” 來指定變數。



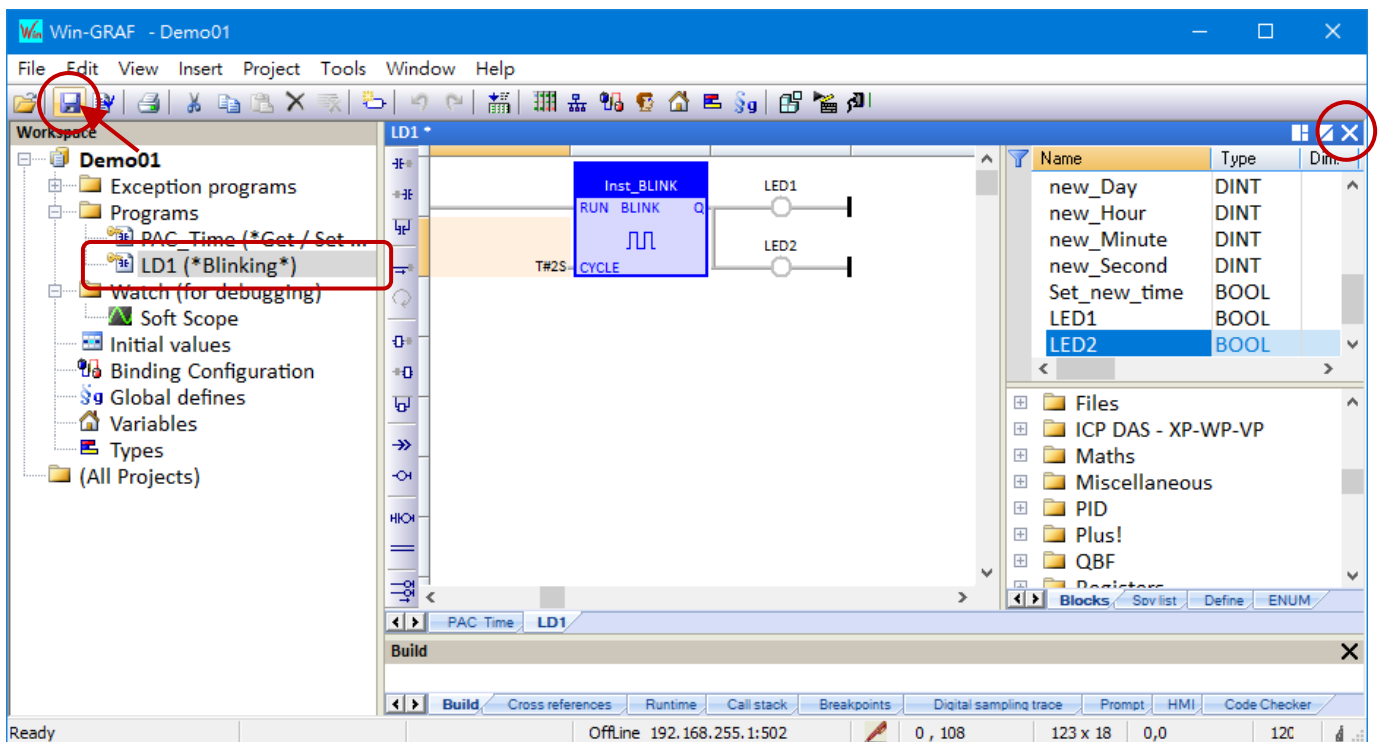
也可從變數區拖曳 “LED2” 變數。





8. 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 OK 完成設定。



9. 最後，點選“Save”按鈕，儲存“LD1”程式。

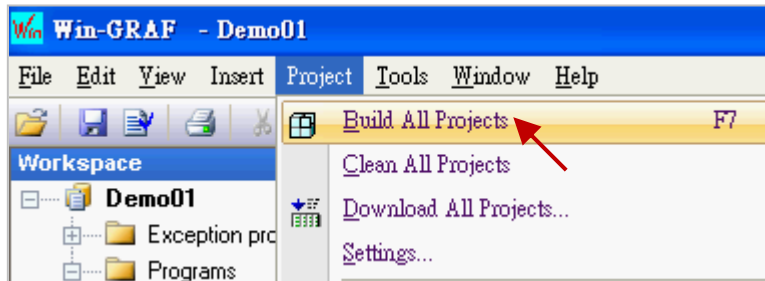


註：  表示該程式目前為開啟狀態（鎖定，不可刪除），可點選程式區右上角的“X”來關閉程式視窗（解除鎖定，）。

2.3.3 編譯程式

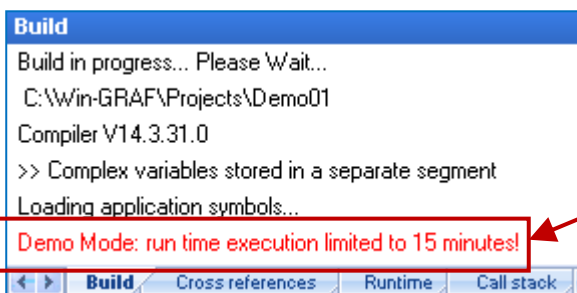
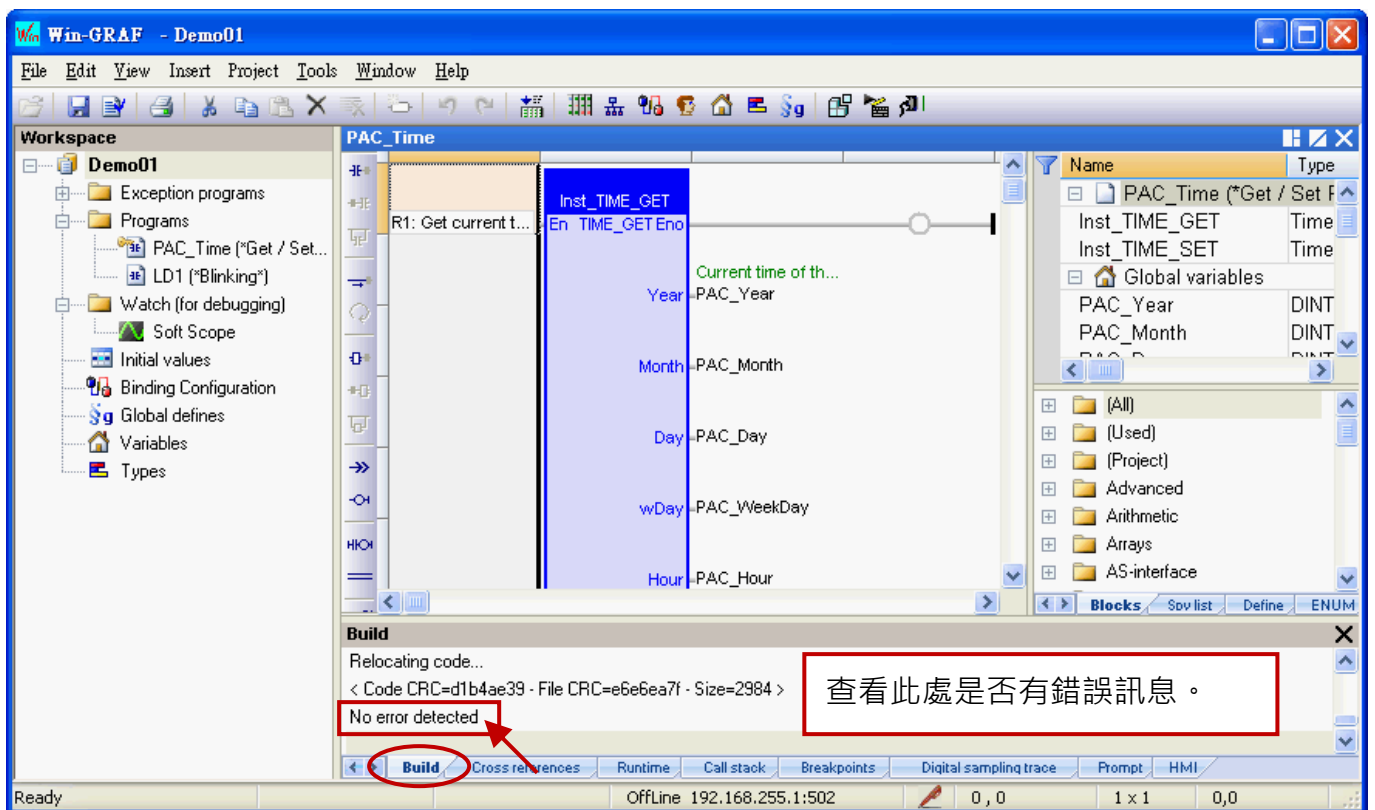
在上一節中，我們已新增並儲存了 LD 程式，為了讓 Win-GRAF 程式可在 PAC 上正常運作，接下來要進行程式編譯。

1. 點選功能表 "Project - 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：若再次修改並儲存程式，請在編譯程式前先執行 "Clean All Projects"。

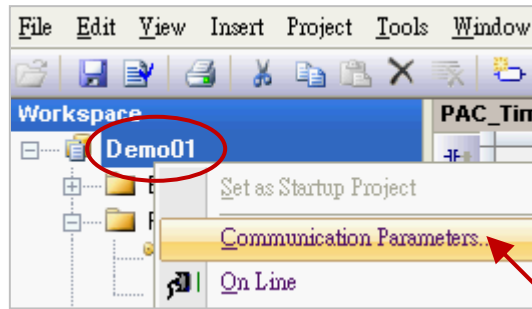


若使用 Demo Mode，此訊息表示專案只能 PAC 中的運行 15 分鐘。


2.3.4 下載程式到 PAC

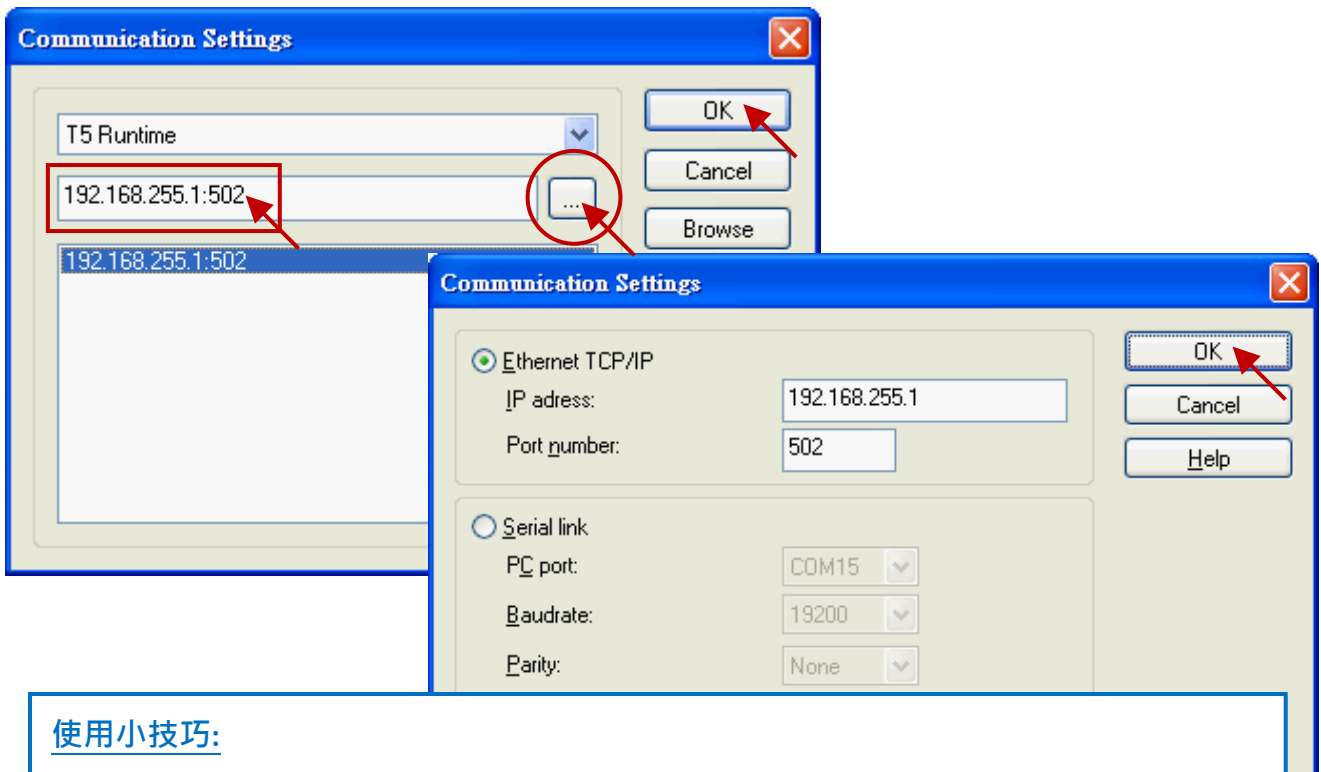
下載程式之前，請先設定好 Ethernet 通訊參數。若想使用序列埠 (例如: RS-232)，請參考 [附錄 C](#)。

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



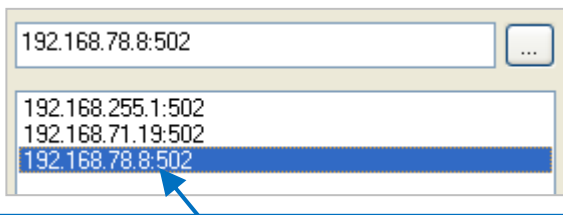
2. 填入 PAC 的 IP 位址與 TCP port 編號 (例如: “192.168.255.1:502”，出廠預設)，並按 “OK”。

點選  按鈕，可新增/修改 IP 位址。



使用小技巧:


可選取 IP 再按 “Del” 鍵，刪除多餘的項目。

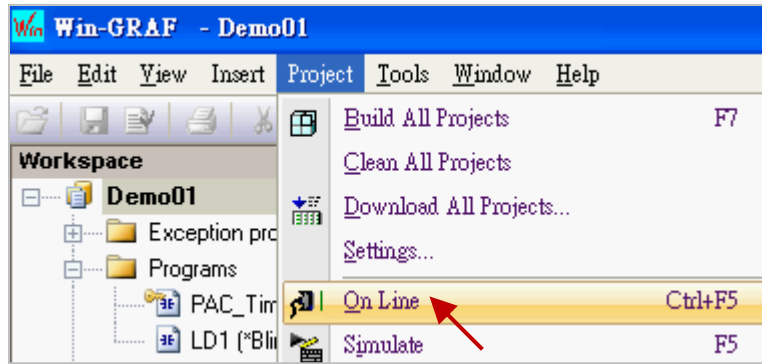


延長通訊的 Timeout 時間?

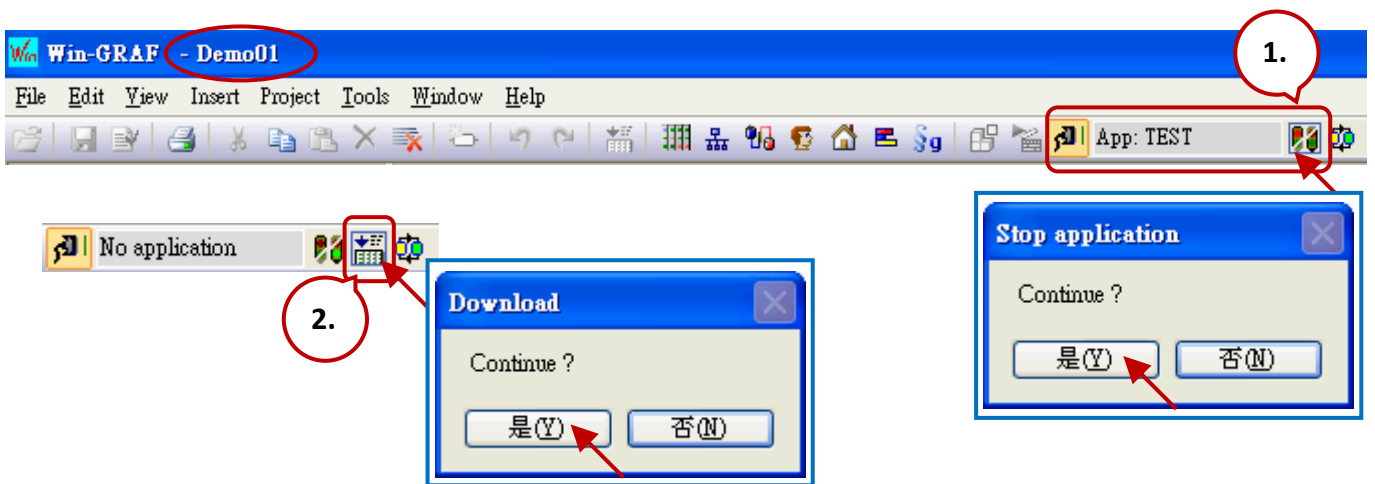
預設為 3 秒，若輸入 “PAC IP:502(n)” 表示 Timeout 設為 n 秒。例如：
192.168.255.1:502(10)，表示 10 秒。

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 - On Line” (或點選工具按鈕 ) 來與 PAC 建立連線。

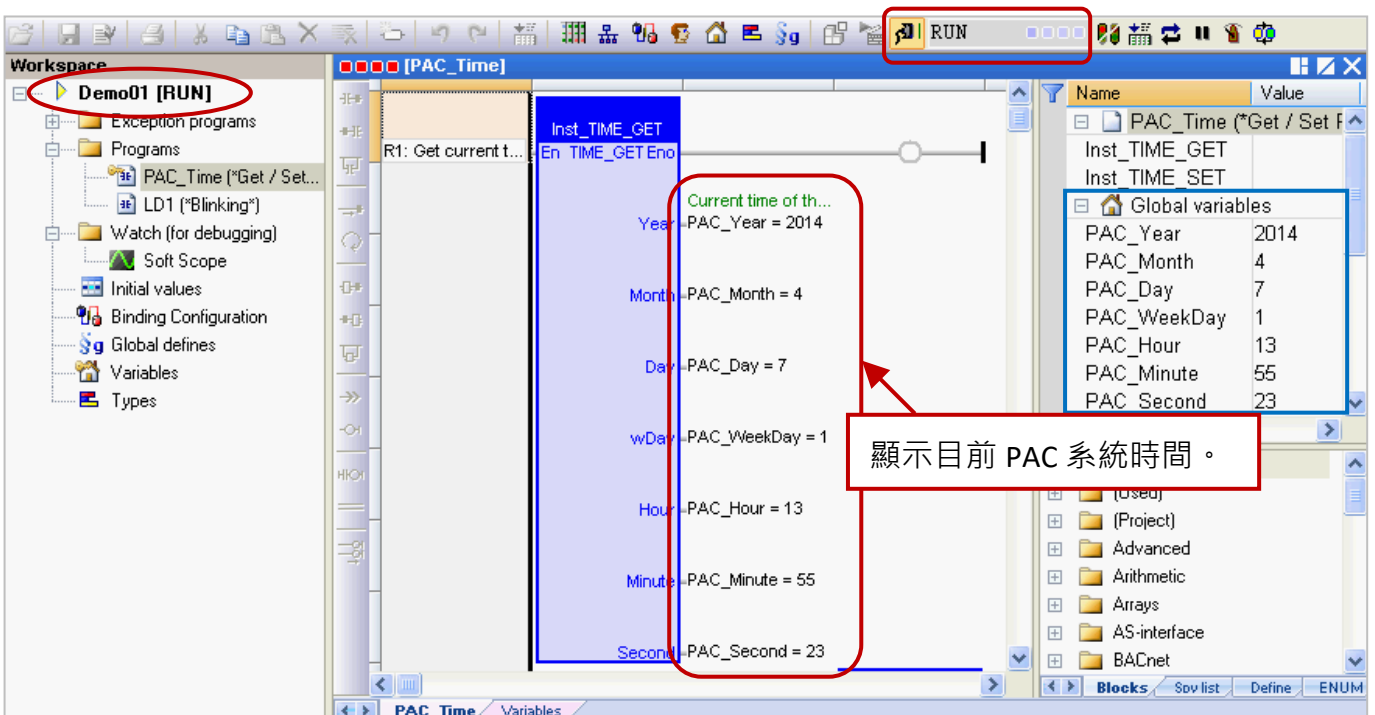


5. 若顯示的專案名稱 (TEST) 與目前的 (Demo01) 不同，請點選“Stop application” 按鈕停止運行舊的專案。接著，點選“Download” 按鈕來下載“Demo01” 專案。



6. 若出現“RUN” 字樣，表示“Demo01” 專案正在 PAC 中運行。

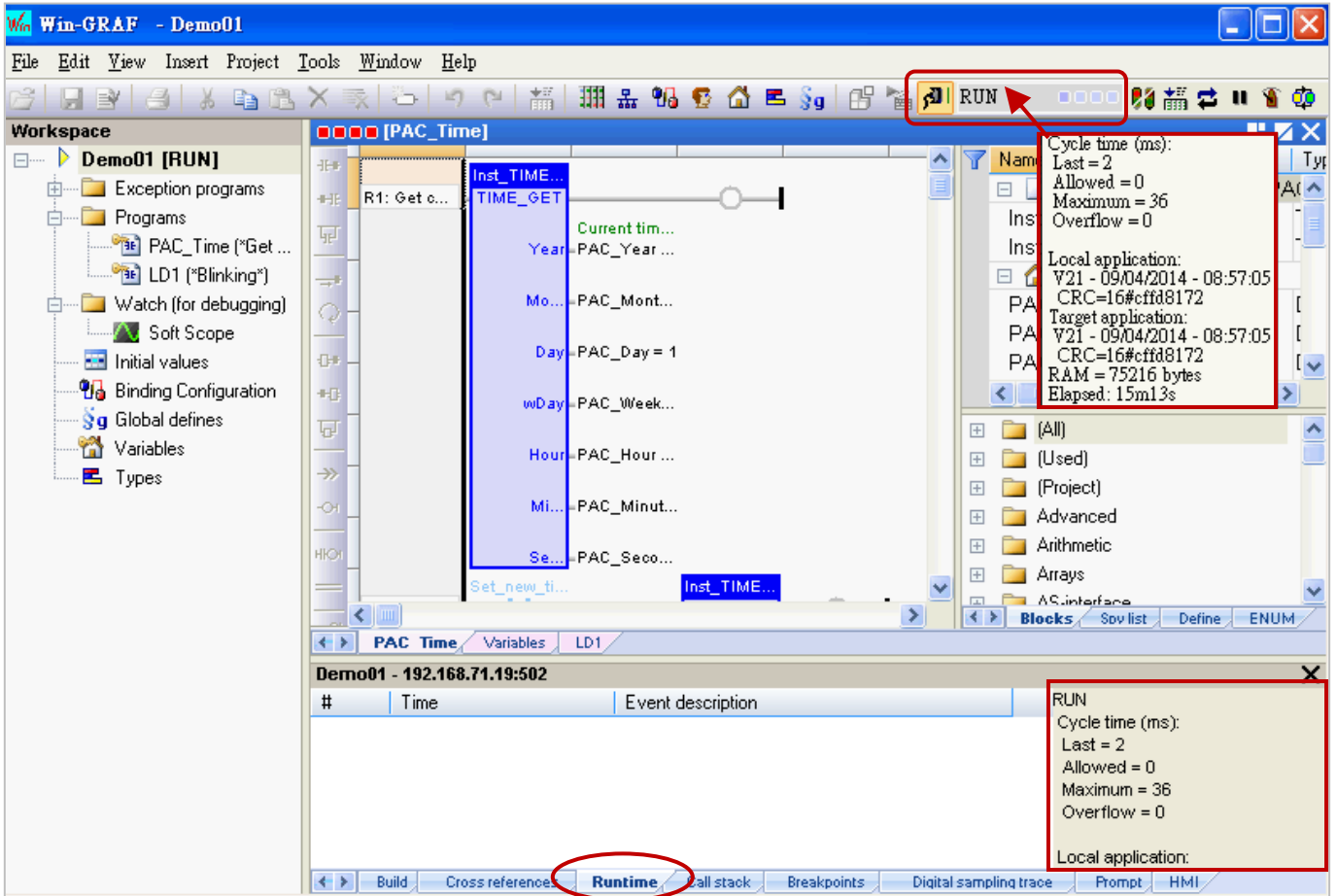
註: 如有錯誤訊息，可參考 [附錄 B](#) 來排除問題。



Cycle time

Workbench 和 PAC 連線的狀態下，將游標移到工具列上“RUN”的位置，可查看 PAC 中程式的 Cycle time。您也可在右下方的訊息區中查看此資訊。

此外，您可在 Runtime 視窗查看是否有程式運行中的錯誤訊息。

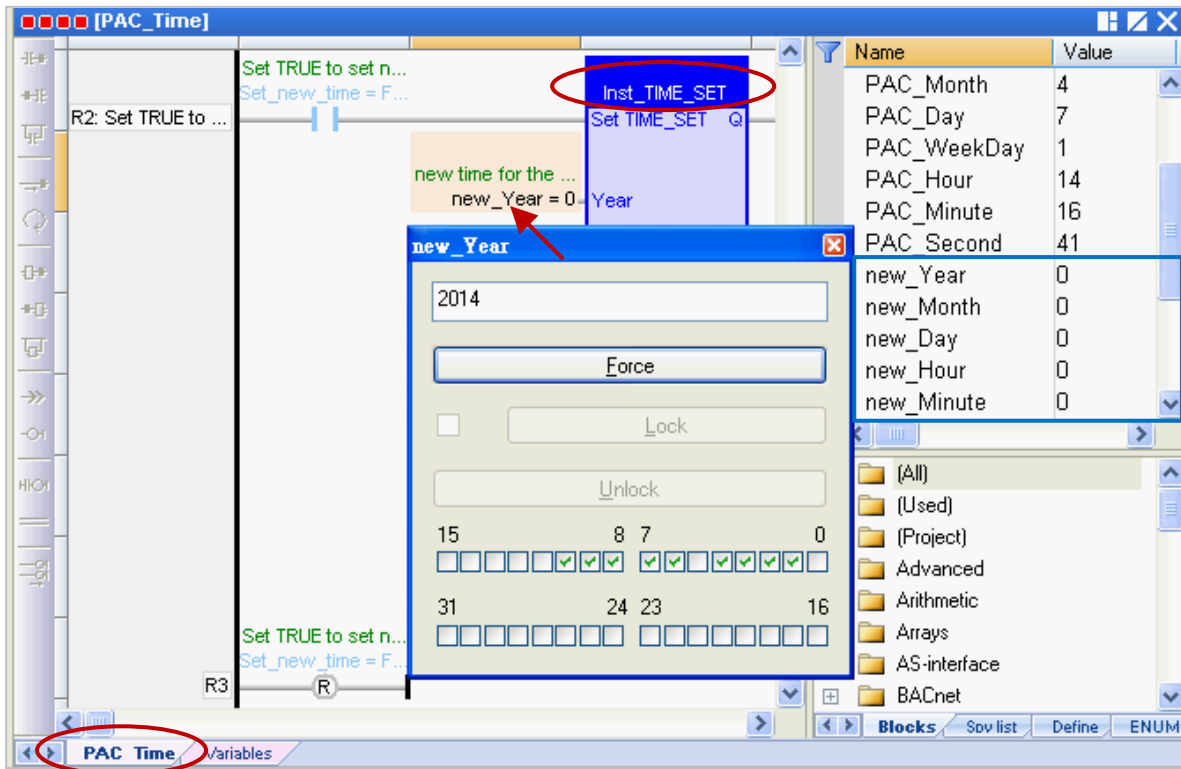


2.3.5 測試程式

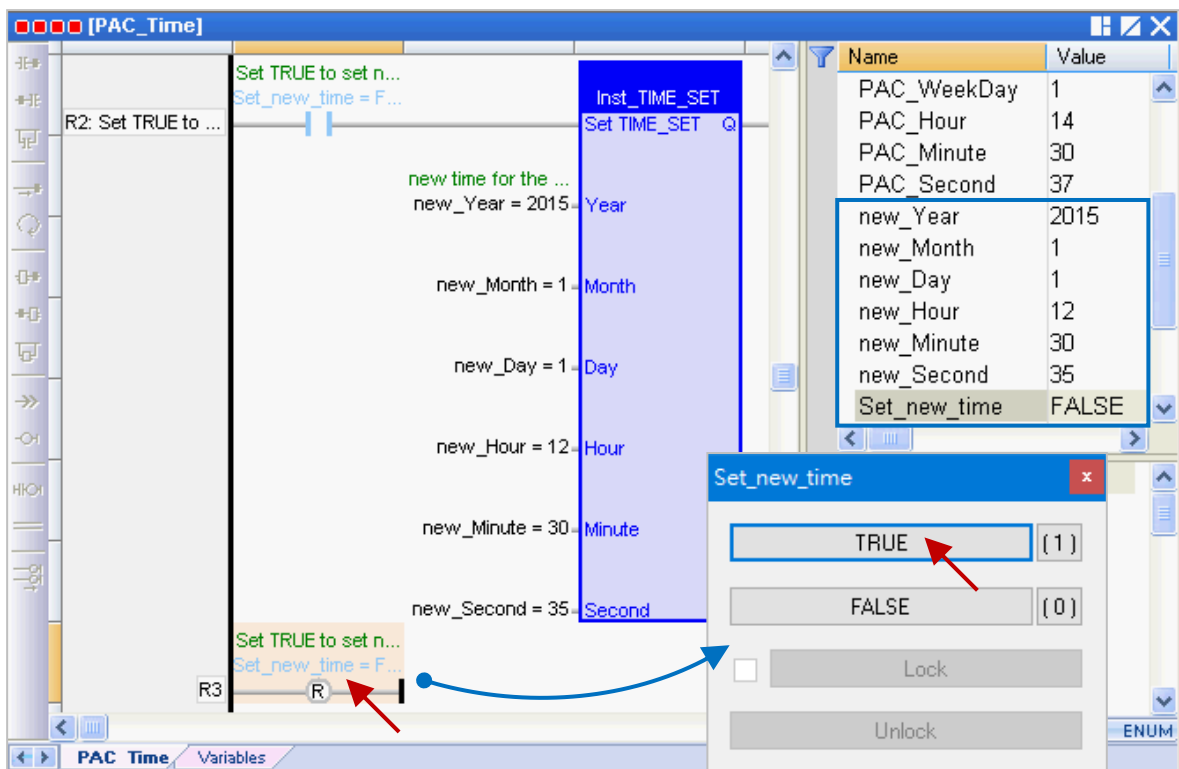
上一章節中，已成功下載“Demo01”專案，接下來要介紹如何測試程式。

“PAC_Time”程式：

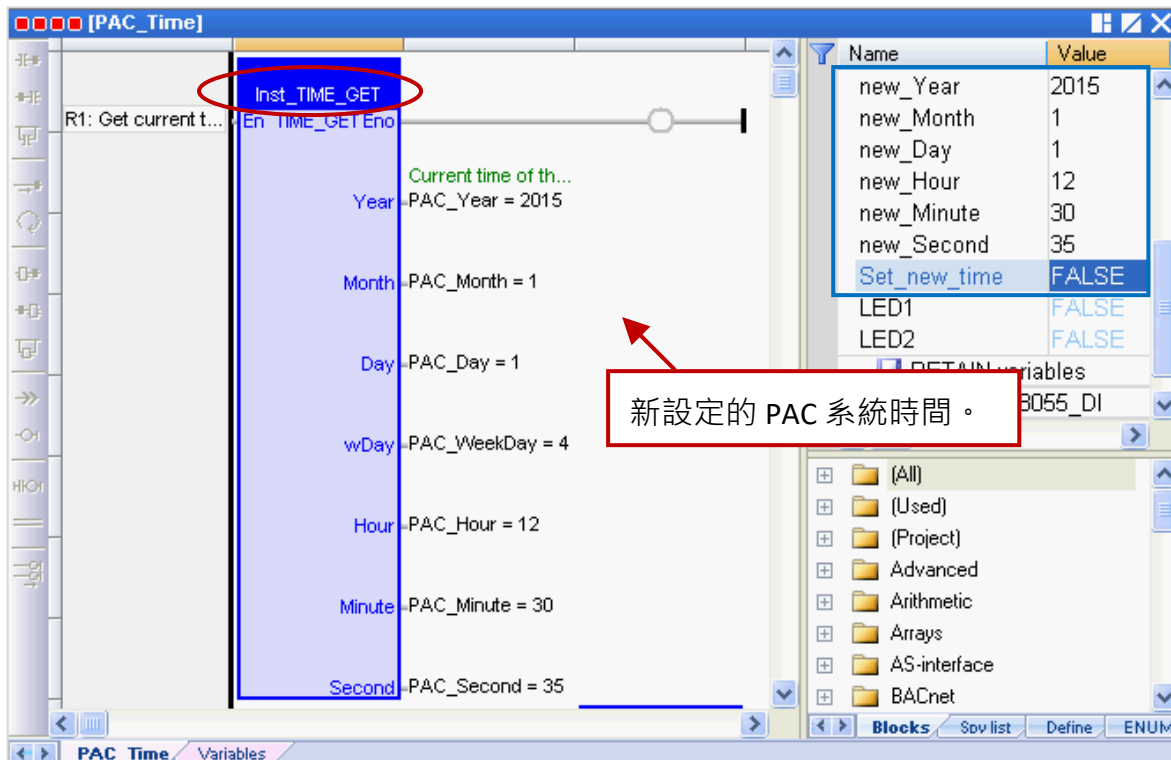
1. 在“TIME_SET”功能方塊中(或變數區中)，滑鼠雙擊變數名稱(例如：“new_Year”)來修改系統時間。



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

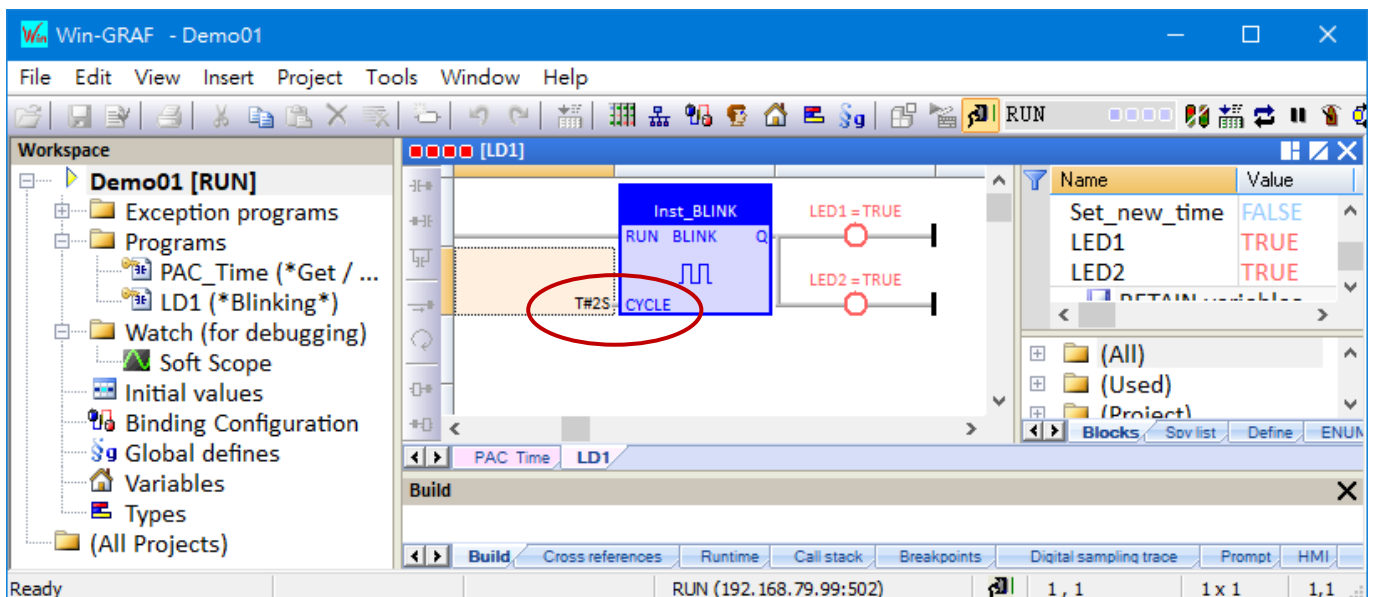


3. 在“TIME_GET”功能方塊中(或變數區中)，顯示了新設定的 PAC 系統時間，且“Set_new_time”變數會自動重置為“FALSE”。



“LD1” 程式:

4. 當“Demo01”專案運行時，可看到 LED1 與 LED2 每 2 秒閃爍一次(即，“T#2S”)，但是無法變更設定值。如需變更，您可再次點選工具按鈕  來取消與 PAC 的連線，並指定一個資料型態為“TIME”的變數。



注意: 請點選“On Line”而不要點選“Stop Application”按鈕，否則 PAC 中的程式將會停止運行。

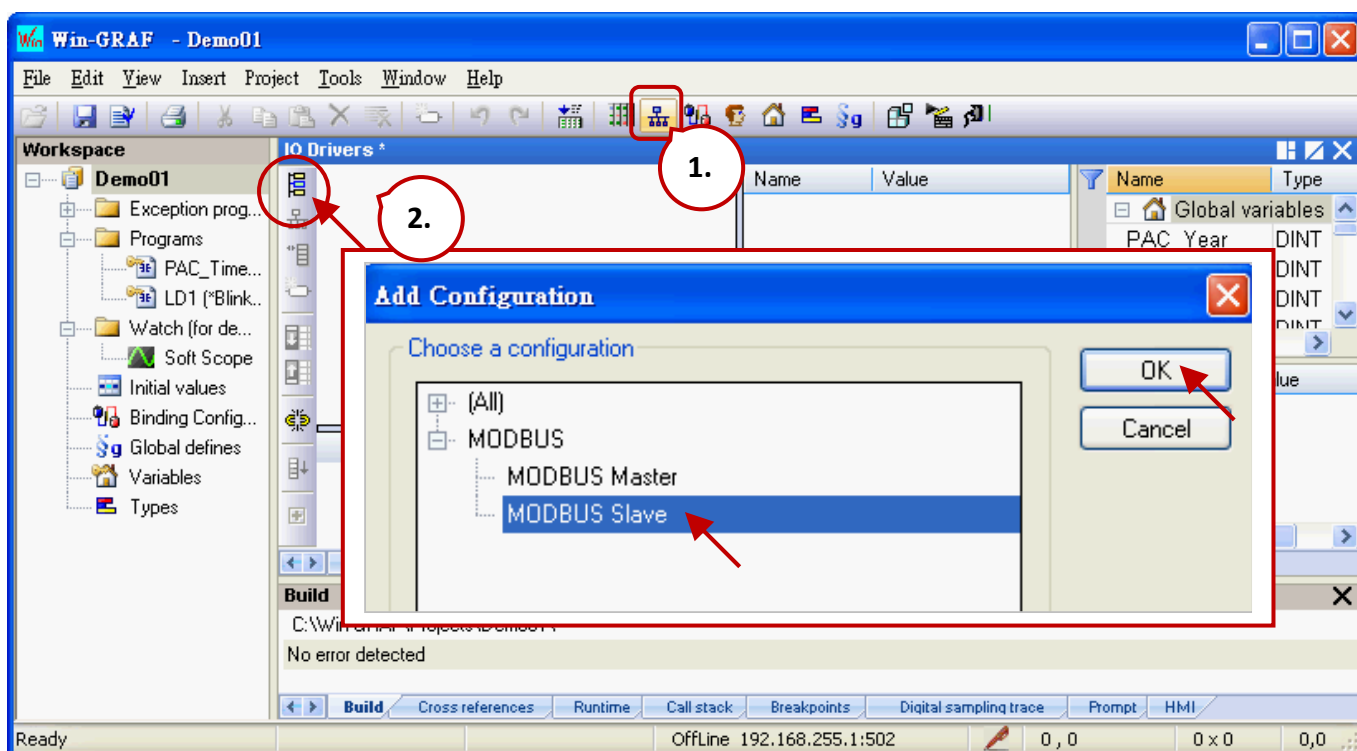
第 3 章 Modbus Slave: 開放 Win-GRAF PAC 與 圖控/HMI 軟體來相互溝通

此章將介紹如何讓圖控軟體 (例如: 泓格科技的“[InduSoft](#)”) 或 HMI 軟體透過 Modbus TCP 或 Modbus RTU 通訊協定來存取 Win-GRAF 的變數資料。請依照以下步驟，進行設定。

3.1 啟用 Win-GRAF PAC 為 Modbus TCP Slave

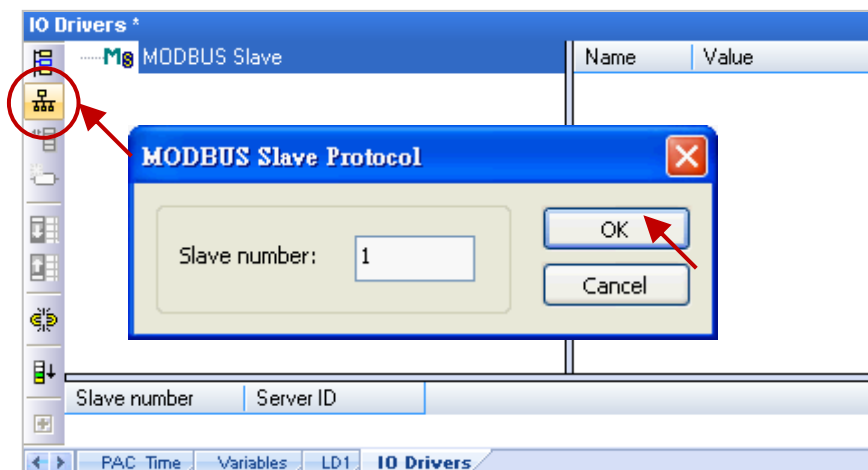
啟用 PAC 為 Modbus Slave

1. 點選工具列上的“Open Fieldbus Configuration” 按鈕，來開啟“IO Drivers” 視窗。
2. 點選“IO 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。



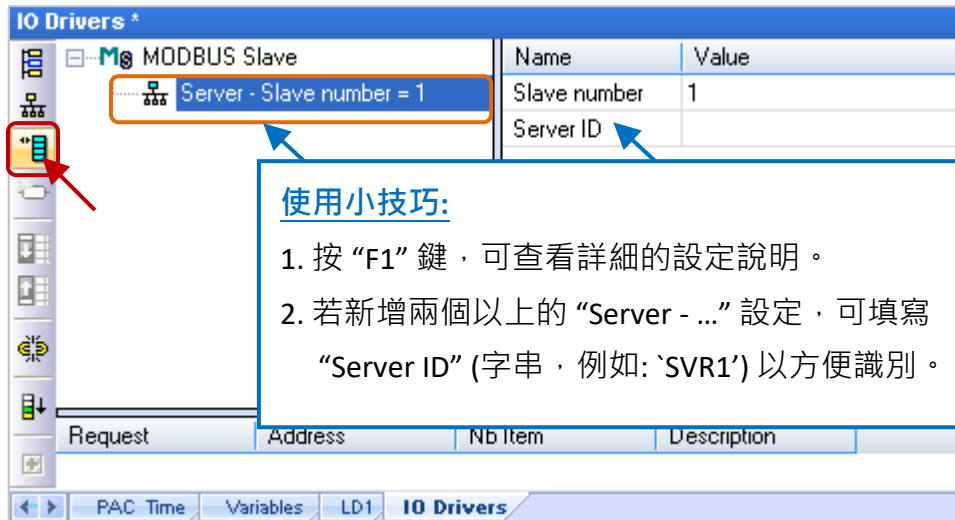
設定 Slave ID (或 Server ID)

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



建立 Data Block

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



5. 讀取 AI 變數:

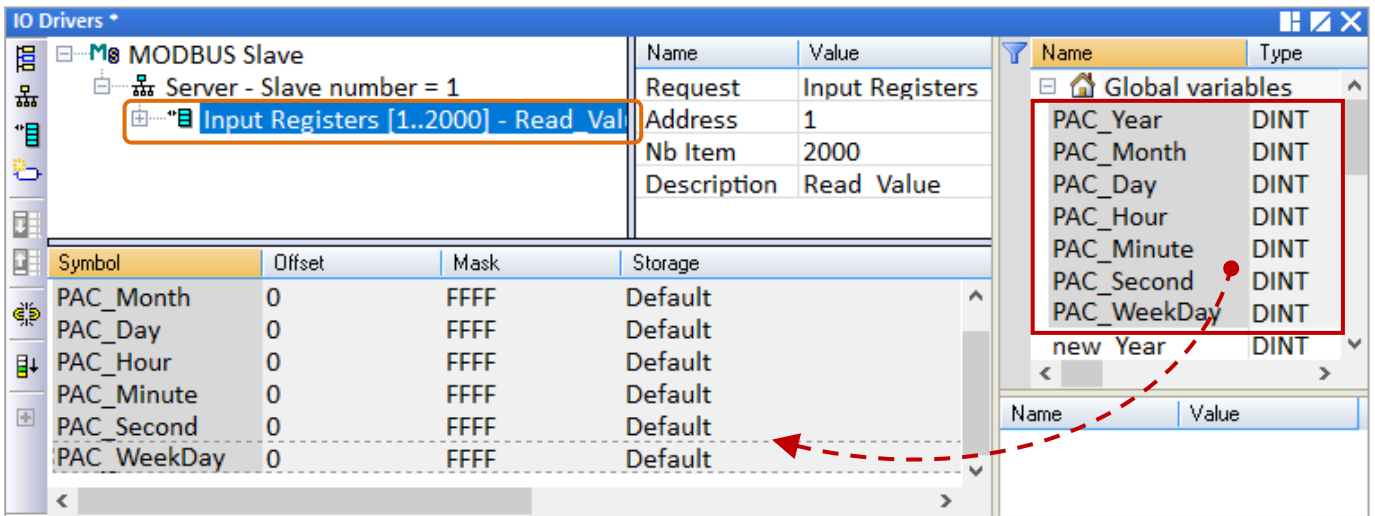
在 “Description” 填入識別的資訊並點選 “Input Registers” 選項，建議設定 “Base address” 為 “1” 並設定 “Nb items” 為大於 200 的數值。



“Nb items” 表示一個 “Data block” 有多少個變數資料 (Bit, Word) 能被存取。若 Modbus Master (即·圖控軟體/HMI) 要求的資料，其位址大於此數值 (例如: “2000”)，則 Modbus Slave (即·Win-GRAF PAC) 將不回應。

指定可讓 Modbus Master 存取資料的變數

6. 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，拉到位址對應區。



指定與 Modbus 位址對應的 Offset

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

注意:

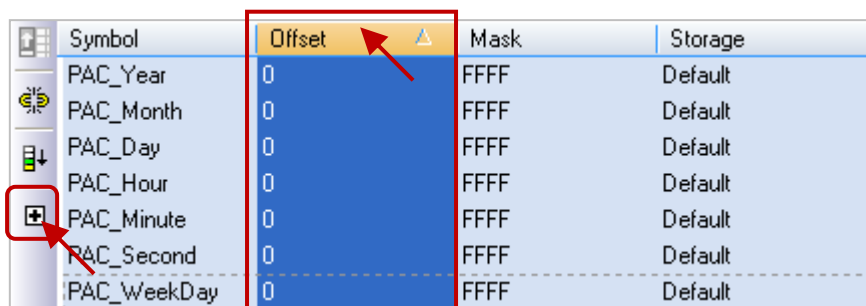
- (1) “Offset” 是 base 0 的位址，因此，Modbus 位址 (base 1) 為 Offset 加 1。
- (2) 若選用的是 32-bit (或以上) 的變數 (此例為 DINT)，則需佔用 2 個連續的位址。

如下表，“0, 2, 4, 6, ...”。

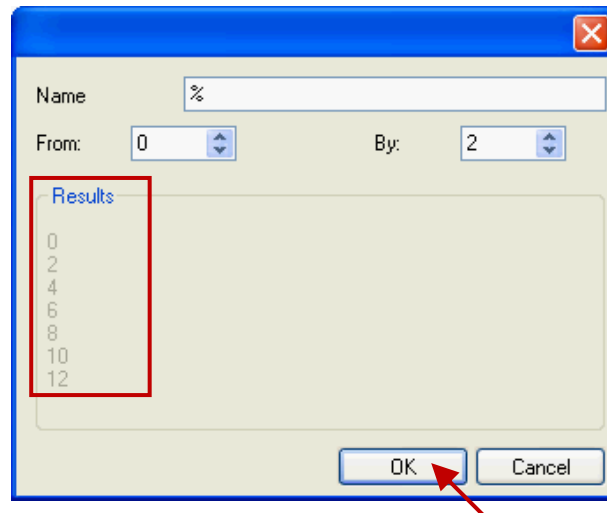
Symbol	Offset	Mask	Storage
PAC_Year	0	FFFF	Default
PAC_Month	2	FFFF	Default
PAC_Day	4	FFFF	Default
PAC_Hour	6	FFFF	Default
PAC_Minute	8	FFFF	Default
PAC_Second	0	FFFF	Default
PAC_WeekDay	0	FFFF	Default

使用小技巧:

滑鼠點選 “Offset” 以選取所有的欄位，點選左側的 “Iterate Property” 按鈕，開啟設定視窗。

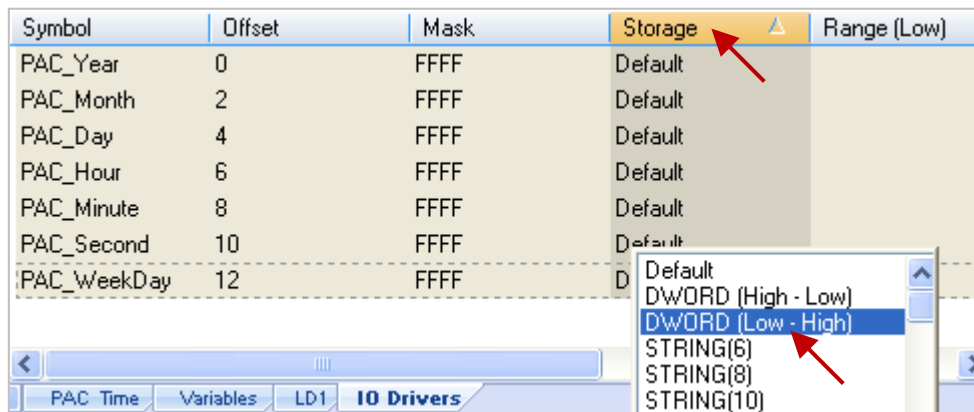


“Name” 維持設定 “%”、將 “From” 填入 0、“By” 填入 2，再點選 “OK”。

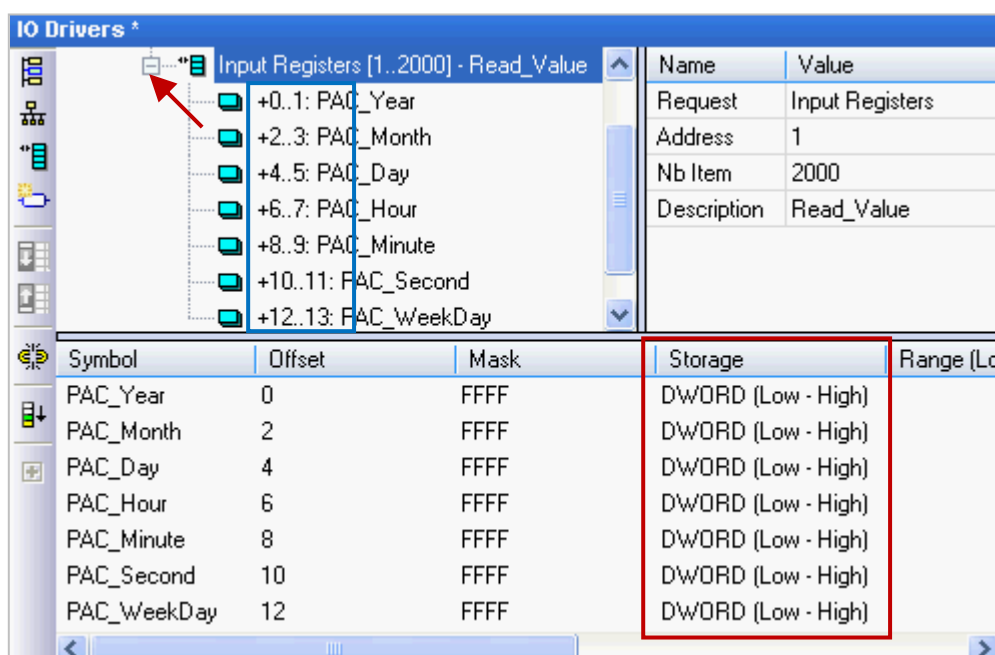


32-bit (或以上) 變數需設定 Storage

- 滑鼠點選 “Storage” 來選取整欄，再按 “Enter” 鍵顯示下拉選單，滑鼠雙擊 “DWORD (Low - High)” 以設定所有項目。



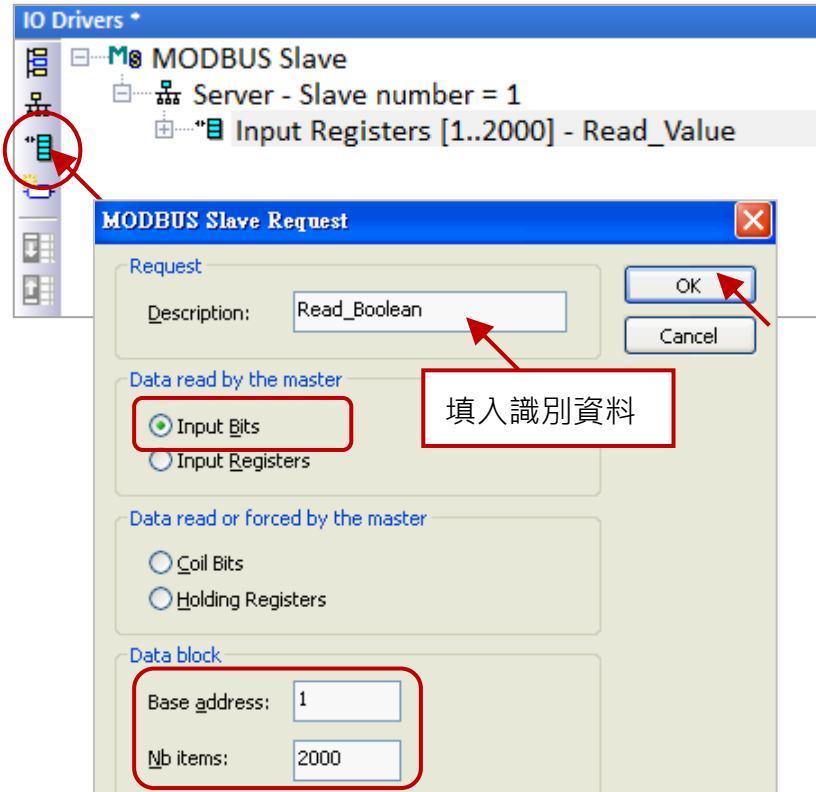
展開這個 “Data Block”，您可見到資料將拆成 2 個位址來儲存。



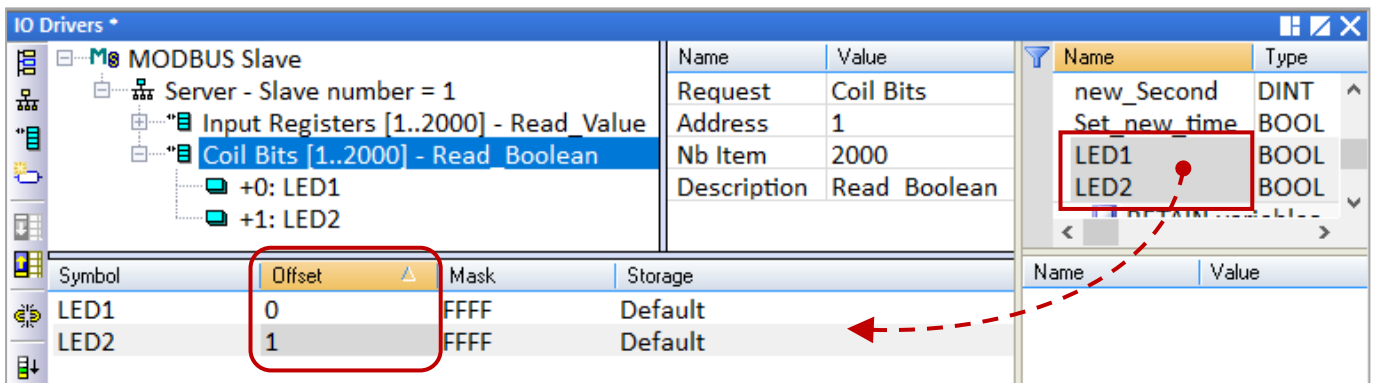
建立第 2 個 Data Block

9. 新增一個 “Data Block” 來讓 Modbus Master 讀取 Boolean 資料。設定方式與步驟 4 ~ 8 類似:

- 1) 滑鼠點選左側的 “Insert Slave/Data Block” 按鈕，來開啟設定視窗。
- 2) 在 “MODBUS Slave Request” 視窗中，填入識別資料、選擇 “Input Bits” 選項、設定 “Base address” 為 “1”、設定 “Nb items” 為 “2000”。




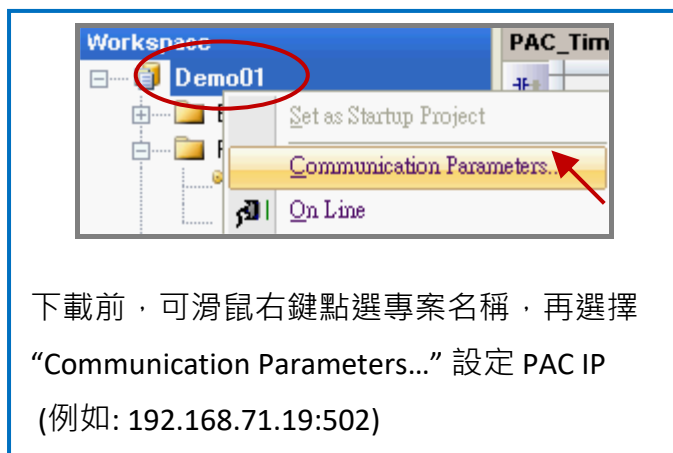
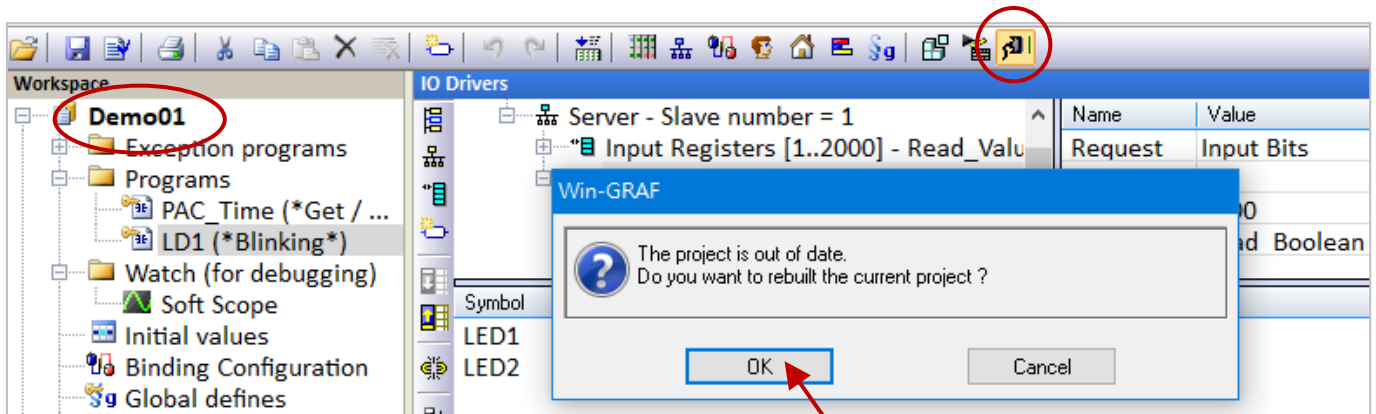
(3) 將變數區的 BOOL 變數 “LED1”、“LED2” 拖曳到地址對應區，並設定 “Offset” 為 “0” 與 “1”。



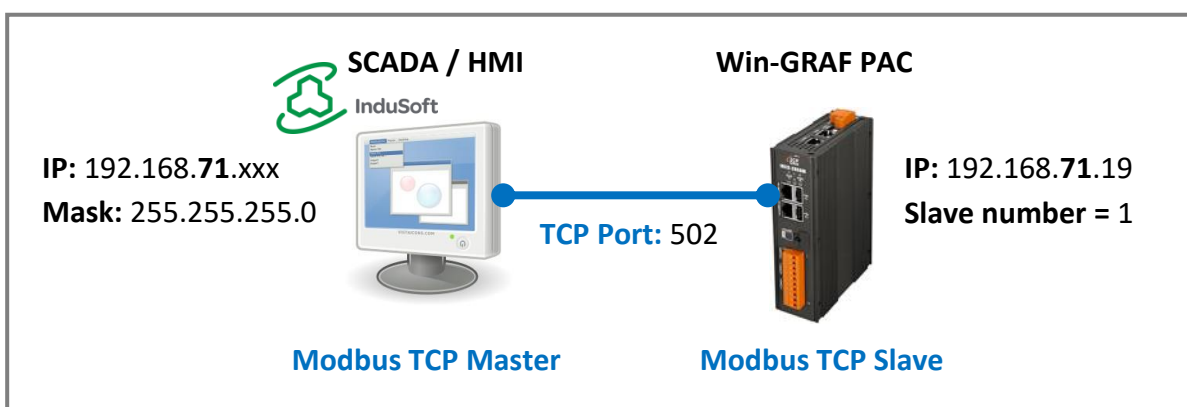
您已完成 Modbus Slave 設定，接下來請下載程式到 Win-GRAF PAC 中。

下載此專案

10. 點選 “On Line” () 來建立連線，並將專案下載到 Win-GRAF PAC 中。若曾編譯 或修改過專案，將會出現以下對話框，請點選 OK 重新編譯並下載專案。



完成所有步驟後，SCADA/HMI 軟體即可透過 Modbus TCP 來存取 Win-GRAF 變數。

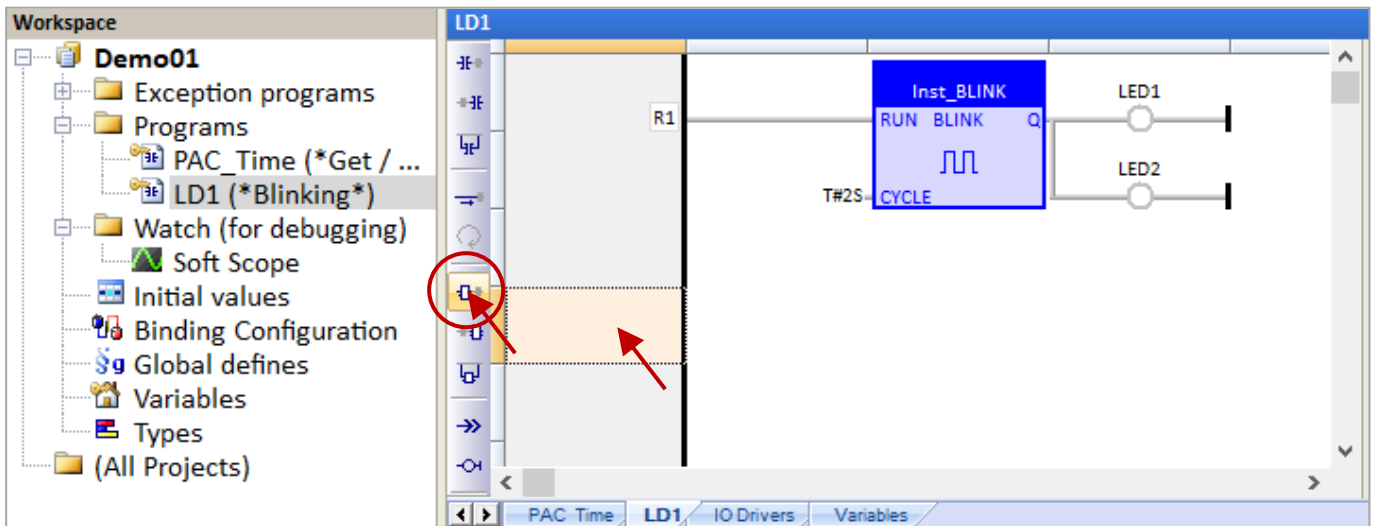


3.2 啟用 Win-GRAF PAC 為 Modbus RTU Slave

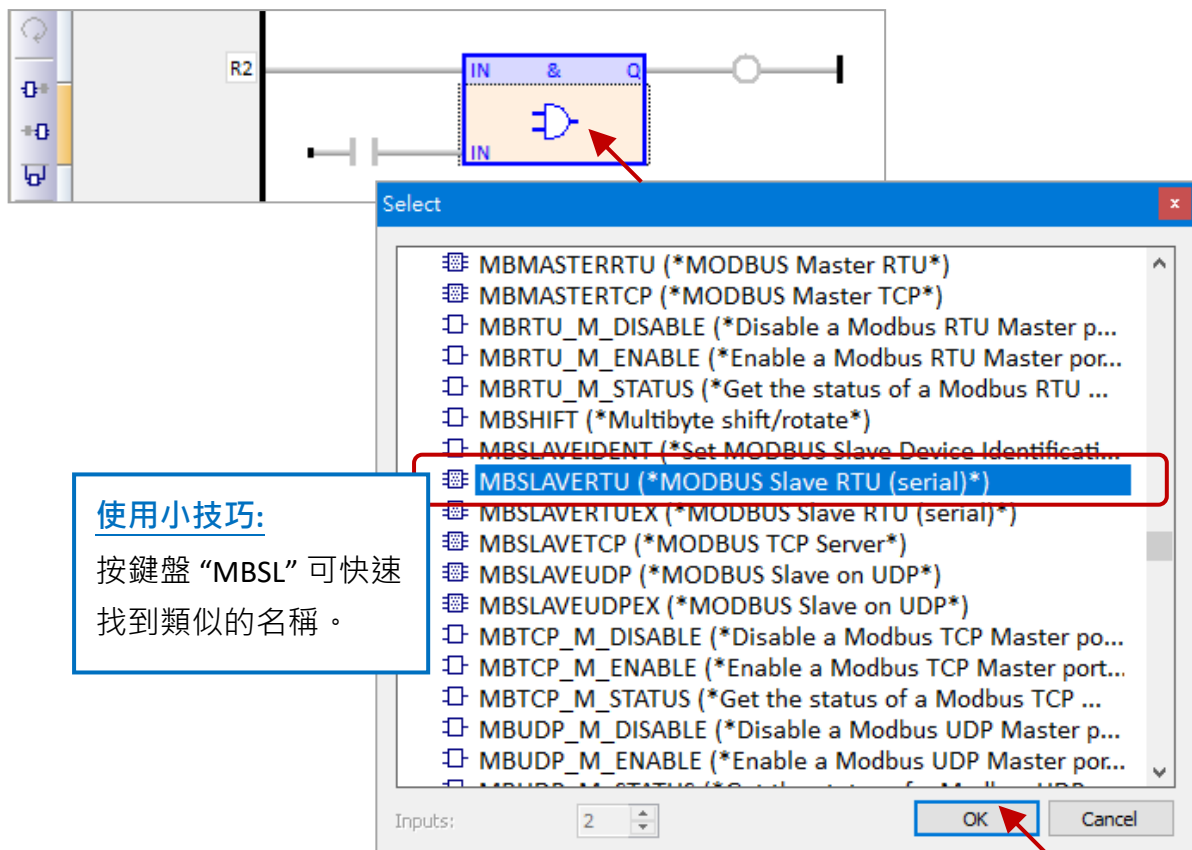
開始前，請確認已啟用 Win-GRAF PAC 為 Modbus Slave，並設定好需開放的變數 (參考 3.1 節)。若要透過 Modbus RTU 來存取資料，請在程式中加入“MBSLAVERTU”或“MBSLAVERTUEX”功能方塊。請依照下列步驟來完成設定。

加入“MBSLAVERTU”功能方塊

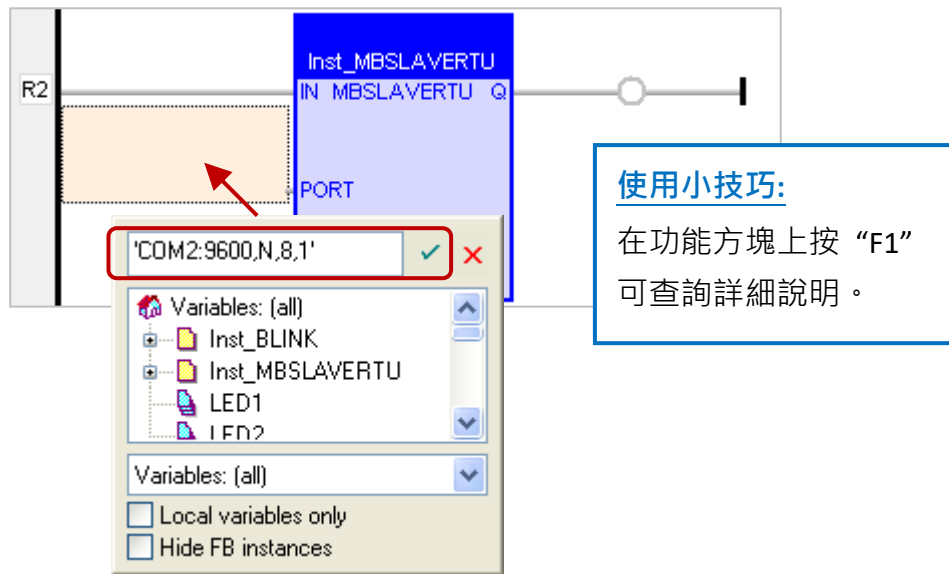
1. 在“LD1”視窗中，滑鼠點選需加入功能方塊的位置，再點選左側的“Insert FB..”按鈕。



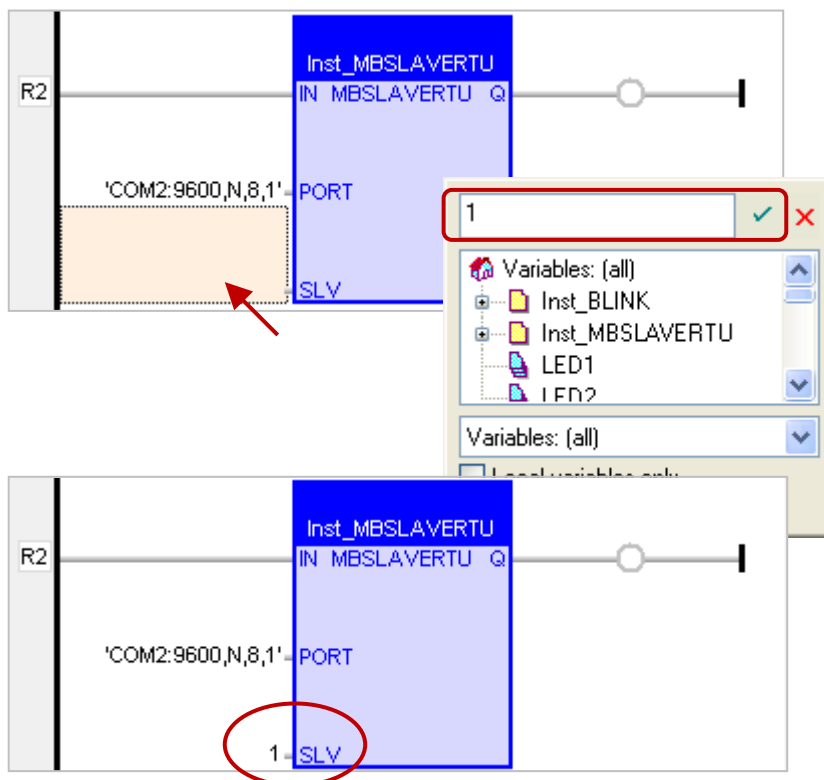
2. 滑鼠雙擊此功能方塊，並選取“MBSLAVERTU”再按“OK”。



3. 於 “MBSLAVERTU” 功能方塊，滑鼠雙擊 “PORT” 的左側區域 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。

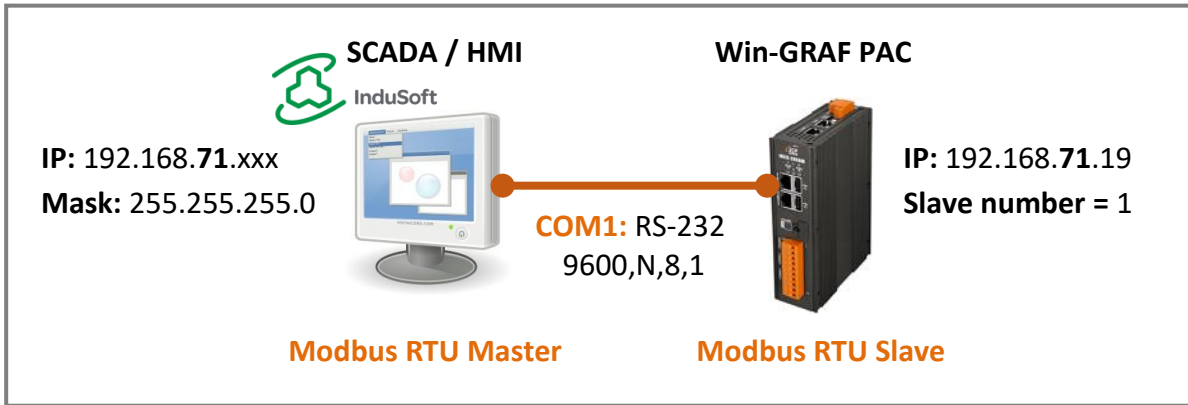


4. 滑鼠雙擊 “SLV” 左側的區域並輸入 “1” (即 [3.1 節](#) - 步驟 3 設定的值)，再點選 完成設定。



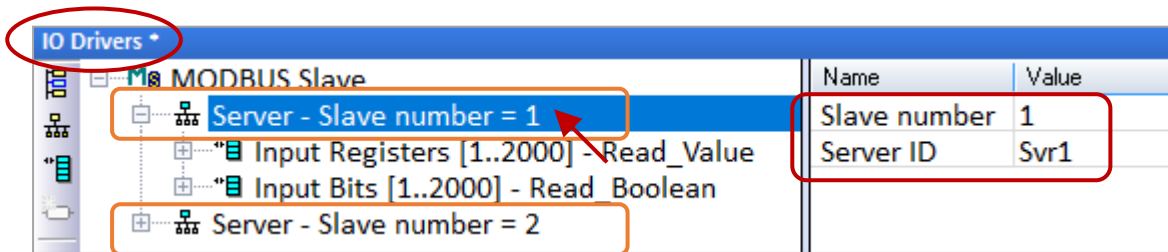
您已完成 “MBSLAVERTU” 功能方塊的設定，請將專案下載到 Win-GRAF PAC 中。

注意: 同一台 PAC 內可啟用多個 Modbus RTU Slave Port (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。



加入 “MBSLAVERTUEX” 功能方塊

若 “IO Drivers” 視窗中有多個 Modbus Slave 配置 (建議只使用一個) , 則可使用 “MBSLAVERTUEX” 功能方塊。



您可依照上述步驟 1 ~ 4 在 “LD1” 視窗中建立一個 “MBSLAVERTUEX” 功能方塊。相較於 “MBSLAVERTU” 它多了一個 “SrvID” 設定，滑鼠雙擊 “SrvID” 左側再填入 Modbus Slave 設定中的 “Server ID” (例如: ‘Svr1’，字串格式)。



注意:

- 使用 MBSlaveRTUex，您可依照 MODBUS Slave 配置來指定 Server ID。若設為空字串，將會採用第一個 Server 的配置。
- MBSlaveRTU 則永遠使用第一個 Server 的配置。

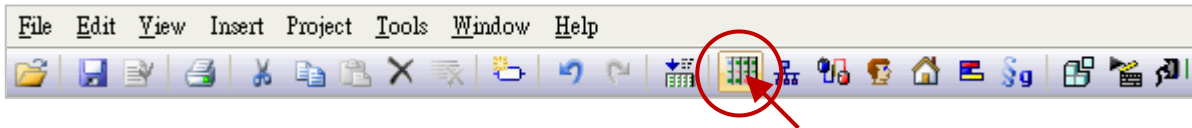
您已完成 “MBSLAVERTUEX” 功能方塊的設定，請將專案下載到 Win-GRAF PAC 中。

第 4 章 使用 “I/O Board” 功能

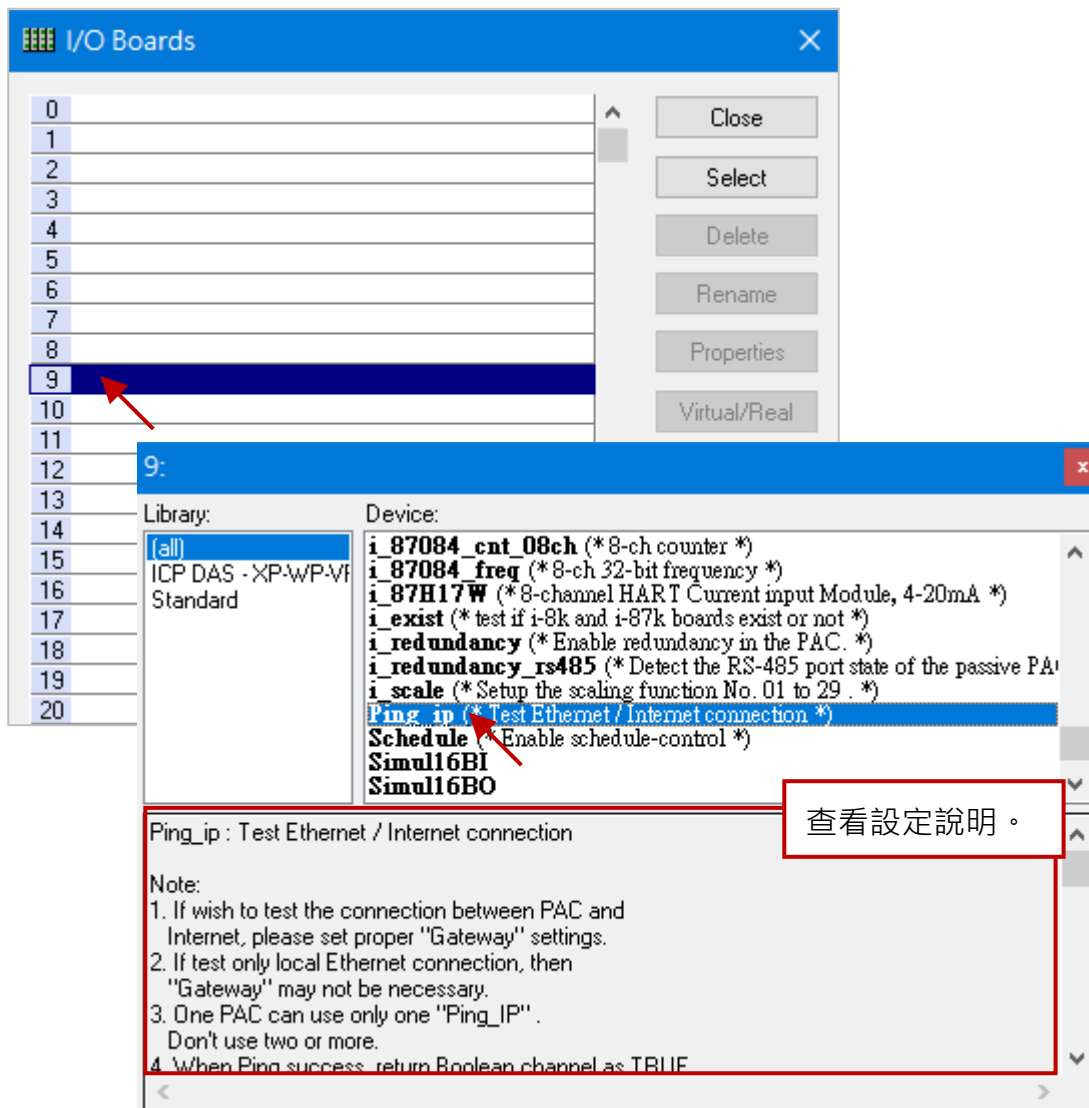
注意:

RPAC-2658M 不支援 Local I/O。如需連接遠程 I/O，請參考第 8 章 - 連接 DCON I/O 模組。

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



2. 滑鼠雙擊任一個編號，來選用需要的功能。

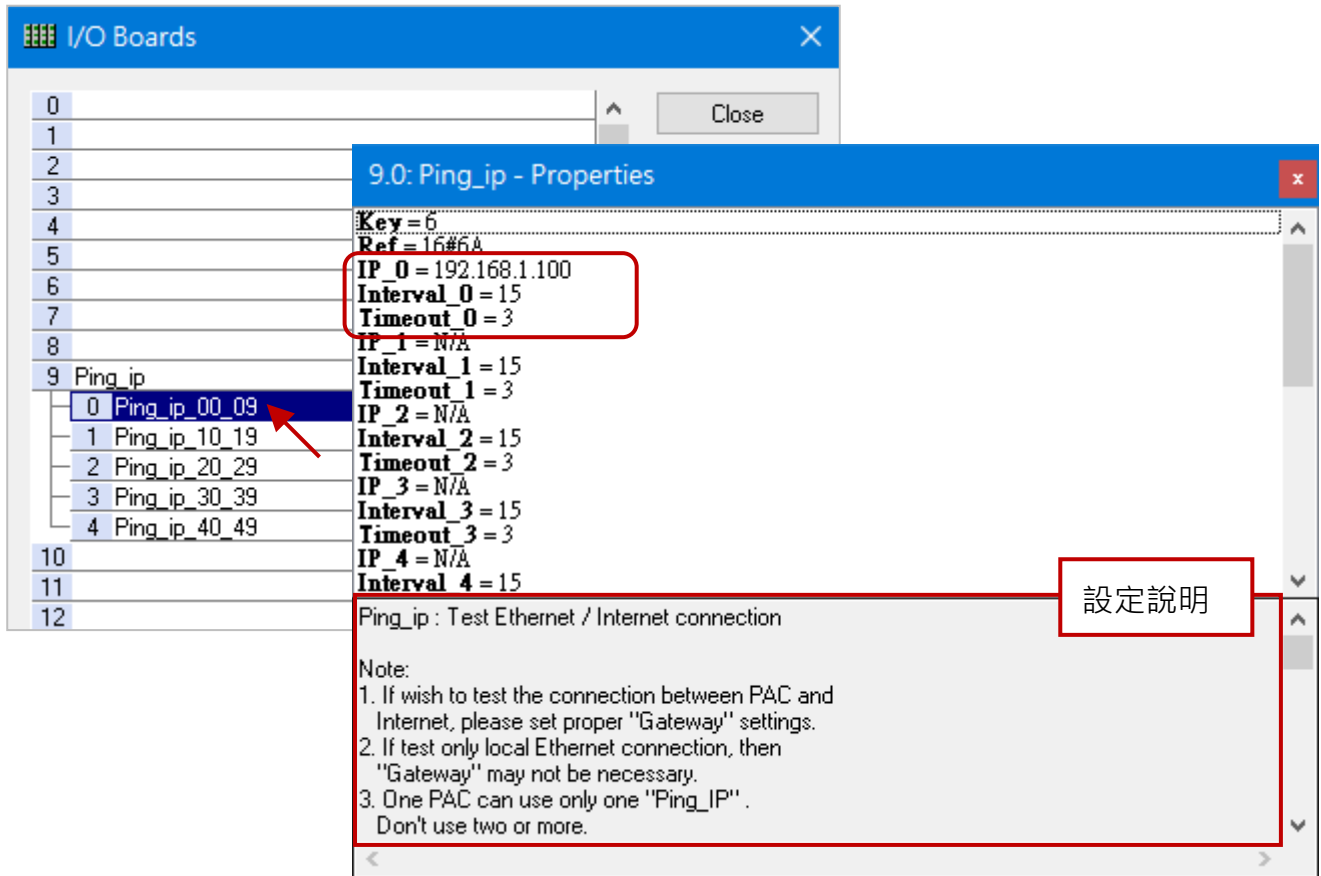


註: “i_8xxx” 與 “i_87xxx” 只適用在有支援 Local I/O 的 PAC。

4.1 Ping_ip: 測試遠端 Ethernet/Internet 設備的連線

"Ping_ip" 功能可用來知道遠端設備的 Ethernet/ Internet 連線是否正常? 最多可設定 50 個 IP 位址。可參考 [第四章](#) 來加入此 "I/O board" 。

1. 滑鼠雙擊 "0: Ping_ip_00_09" 可開啟 "Properties" 視窗。



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. 若是測試區域網路 (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。
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"。)

單位為秒，傳送 "ping" 命令的間隔時間，預設為 15 秒。PAC 內採用的有效值為 6 ~ 86,400 秒。

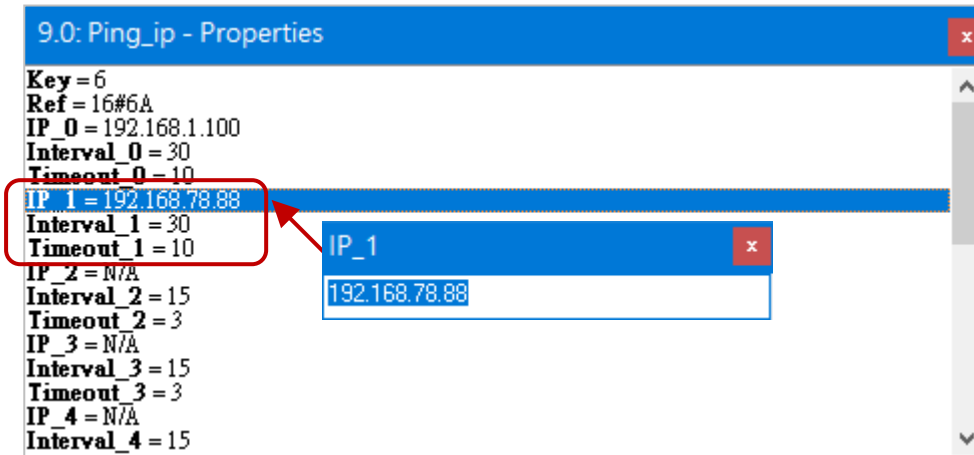
Timeout_01 ~ Timeout_49: (資料型態: "DINT"。)

單位為秒，"ping" 命令的逾時時間，預設為 3 秒。PAC 內採用的有效值為 2 ~ 30 秒。

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， $10 \times 3 = 30$)。

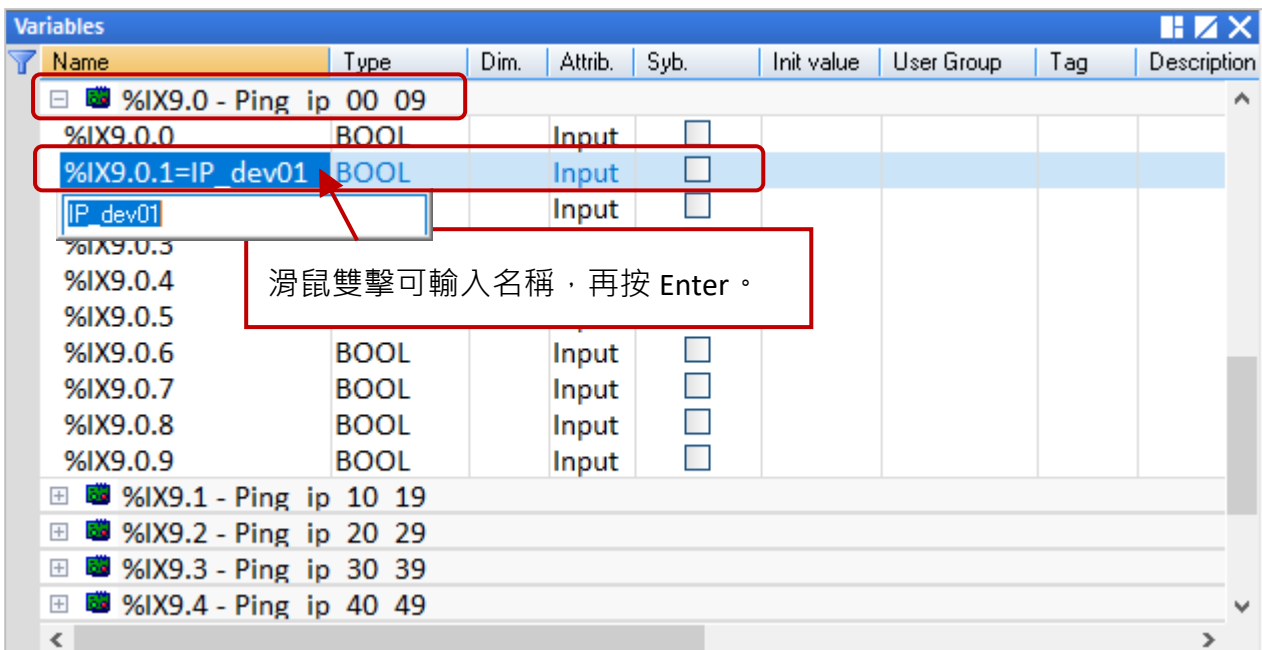
2. 滑鼠雙擊所需項目，來輸入值並按 "Enter" 鍵完成設定。



3. 在 "I/O Boards" 視窗內設定 "Ping_ip" 後，會自動在 "Variables" 視窗中新增 50 個 "BOOL" Input 變數，當 Win-GRAF 有連上 PAC 時，會顯示遠端設備的連線狀態。

TRUE: 表示連線 OK。

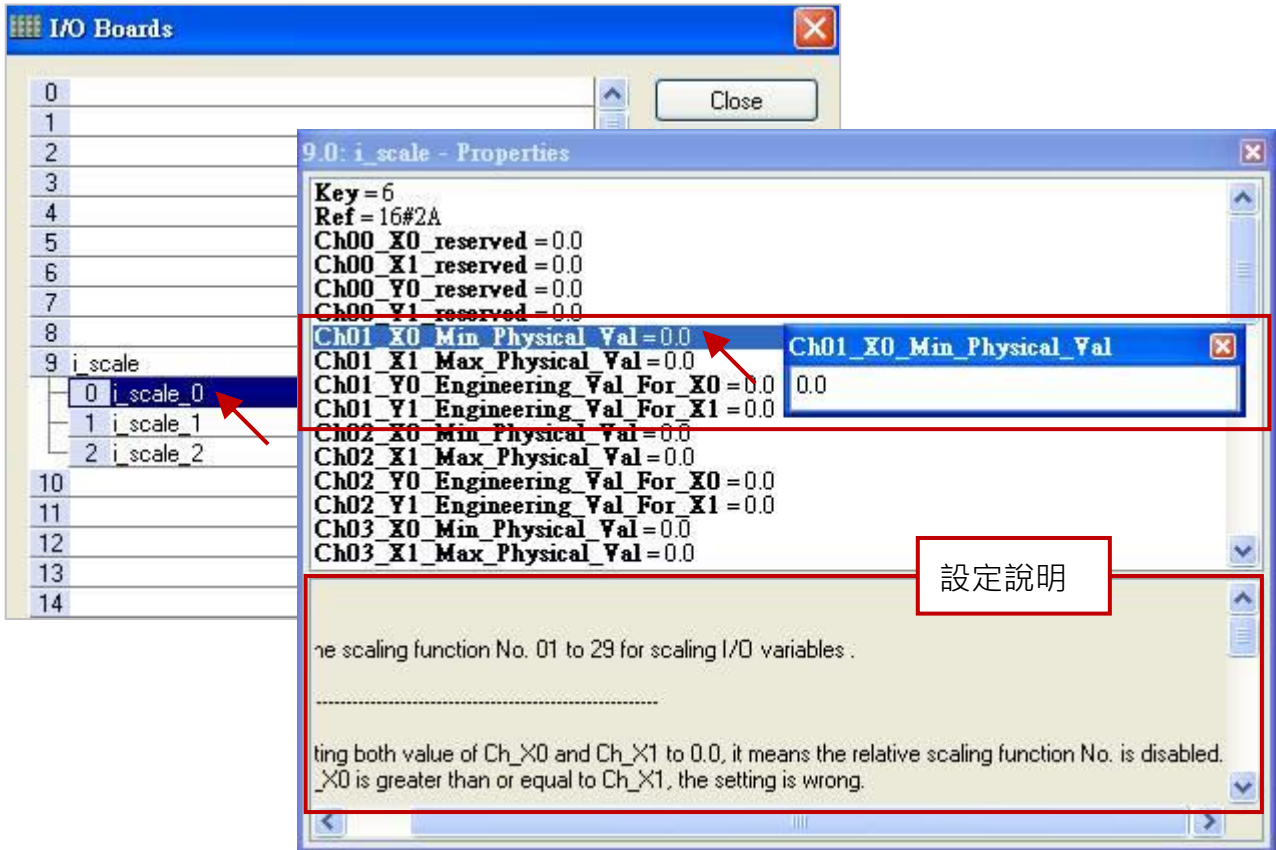
FALSE: 表示連線失敗 或 接線問題。



4.2 i_scale: 比例轉換

“i_scale” 功能提供最多 29 組比例轉換設定，可將 I/O 實體值轉換成工程值。可參考 [第四章](#) 來加入此 “I/O board”。

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。



參數說明: (Ch: 表示 Ch01 ~ 29; Ch00 為保留值。)

- Ch_X0_Min_Physical_Val: AI (或 AO) 模組的最小值 (X0)。
- Ch_X1_Max_Physical_Val: AI (或 AO) 模組的最大值 (X1)。
- Ch_Y0_Engineering_Val_For_X0: 轉換 X0 後的工程應用值。
- Ch_Y1_Engineering_Val_For_X1: 轉換 X1 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，若想將 “4 ~ 20 mA” 的 AI 值轉換為 “0 ~ 10000” 的工程值，請設定

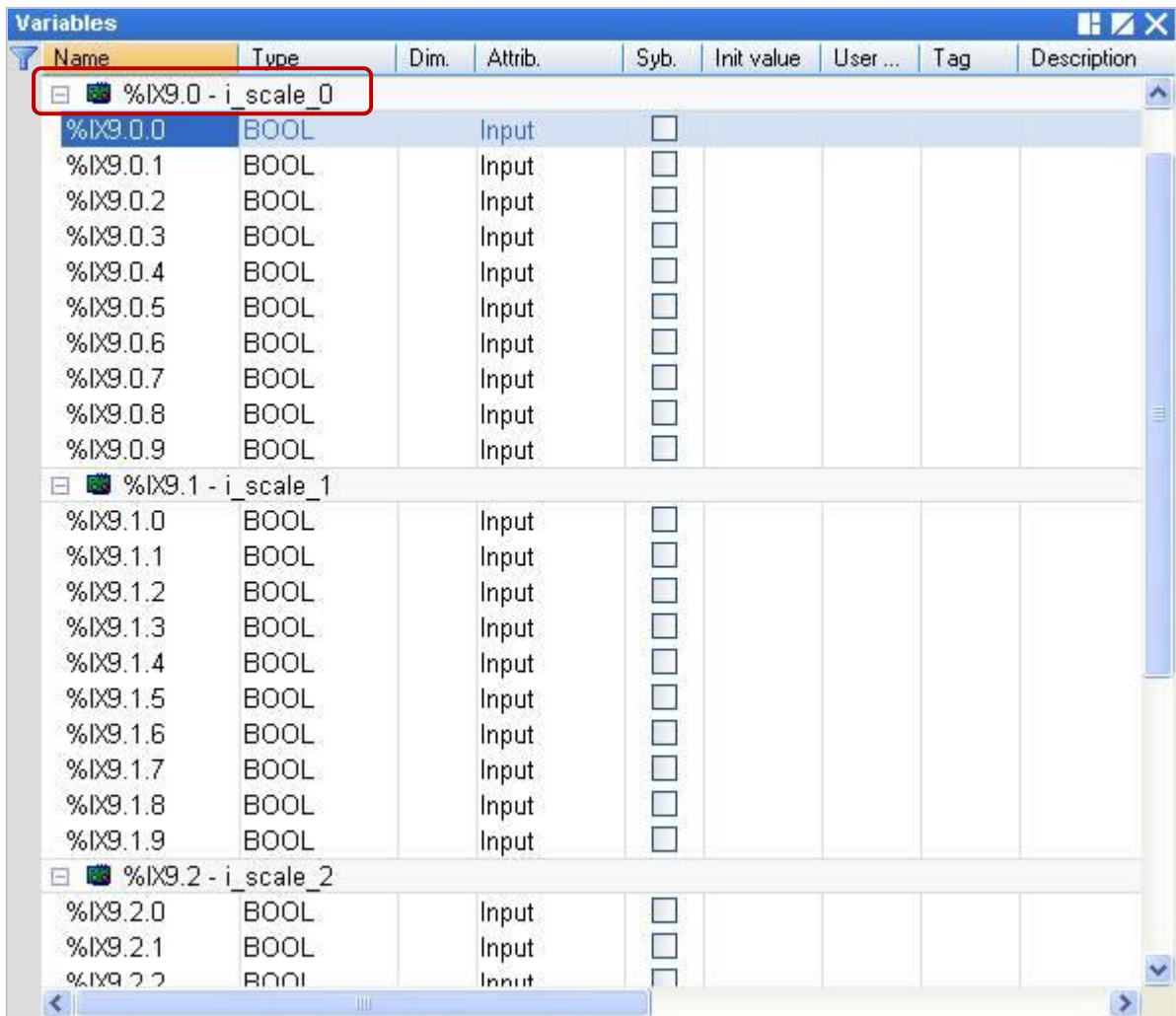
Ch_X0 為 “4.0”、Ch_X1 為 “20.0”、Ch_Y0 為 “0.0” 與 Ch_Y1 為 “10000.0”。

例如，若想將“-10 ~ +10 V”的 AO 值轉換為“-50 ~ 1200”的工程值，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”視窗內設定“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，會顯示比例轉換的狀態。

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

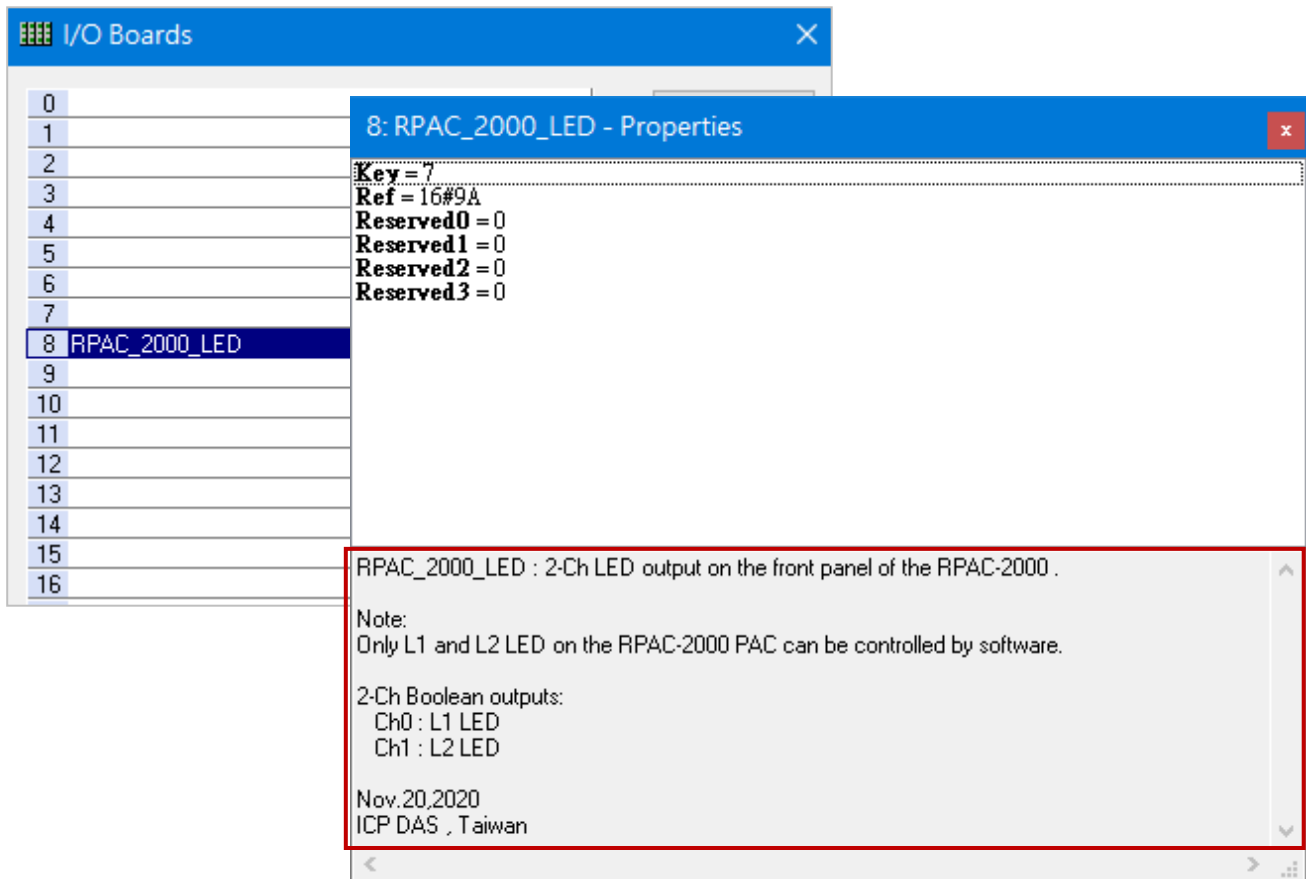


4.3 RPAC_2000_LED: 控制 RPAC-2658M 的 L1 與 L2 LED

RPAC_2000_LED 功能可用來控制 RPAC-2658M 面板上的 L1 與 L2 LED 燈號。

注意: 只有 PAC 上的 L1 與 L2 LED 可由軟體控制。可參考 [第四章](#) 來加入此 “I/O board”。

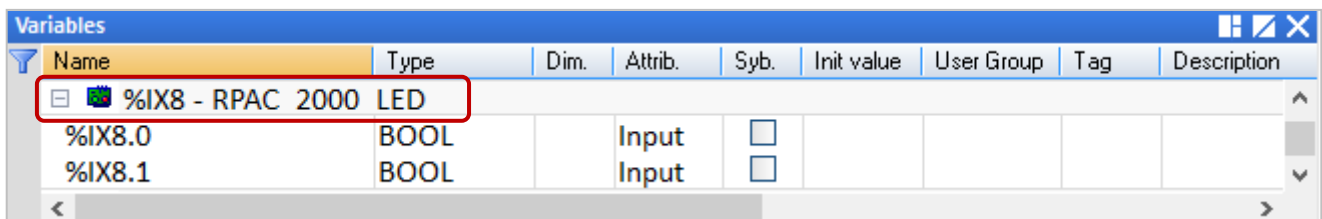
1. 滑鼠雙擊 “RPAC_2000_LED” 開啟 “Properties” 視窗，來查看設定說明。



參數說明:

Ch0、**Ch1**: (資料型態: "BOOL") 代表 L1 與 L2 LED。

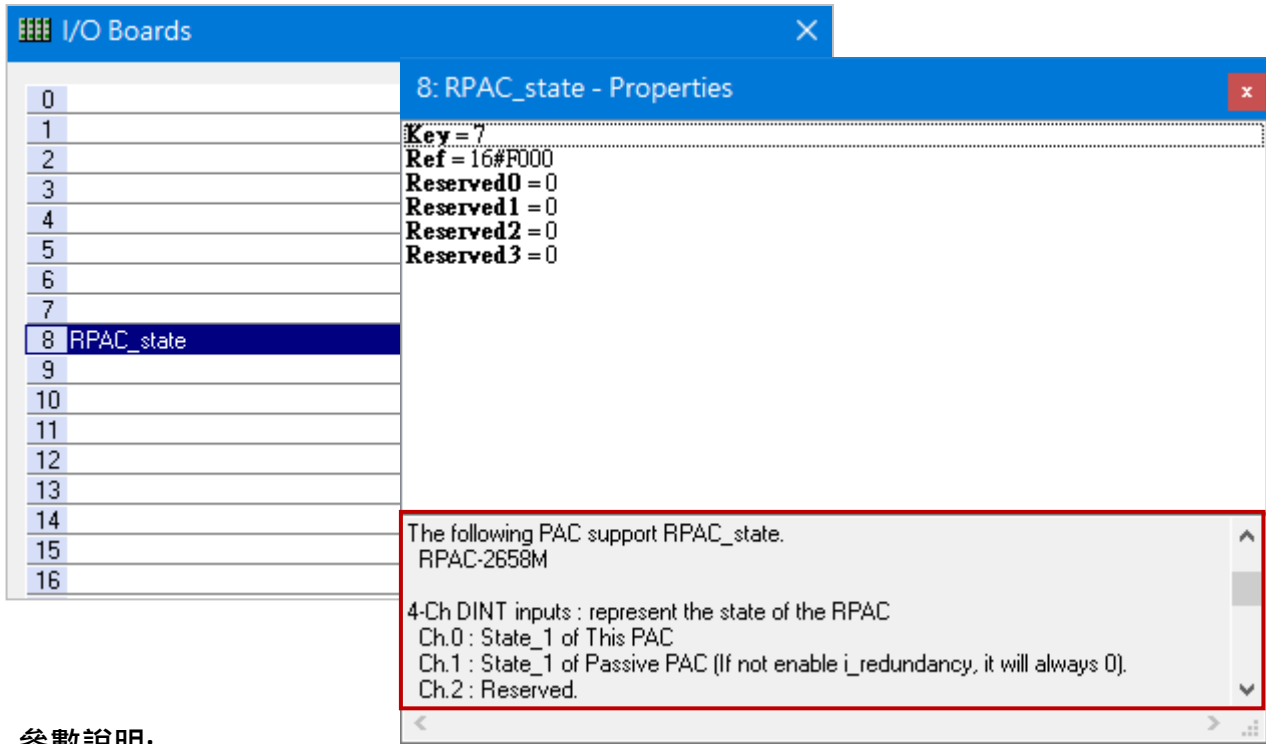
2. 在 “I/O Boards” 視窗內設定 “RPAC_2000_LED” 後，會自動在 “Variables” 視窗中新增 2 個布林變數，當 Win-GRAF 有連上 PAC 時，會顯示 LED 的狀態。



4.4 RPAC_PAC_state: 偵測 RPAC-2658M 的 LAN 狀態

RPAC_PAC_state 功能可用來偵測 RPAC-2658M 的狀態。可參考 [第四章](#) 來加入此 “I/O board”。

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



參數說明:

4-Ch DINT 輸入參數，表示 RPAC PAC 的狀態。

Ch.0: 此台 PAC 的 State_1。

Ch.1: Passive PAC 的 State_1 (如果 i_redundancy 未啟用，會是 0)。

Ch.2, Ch.3: 保留。

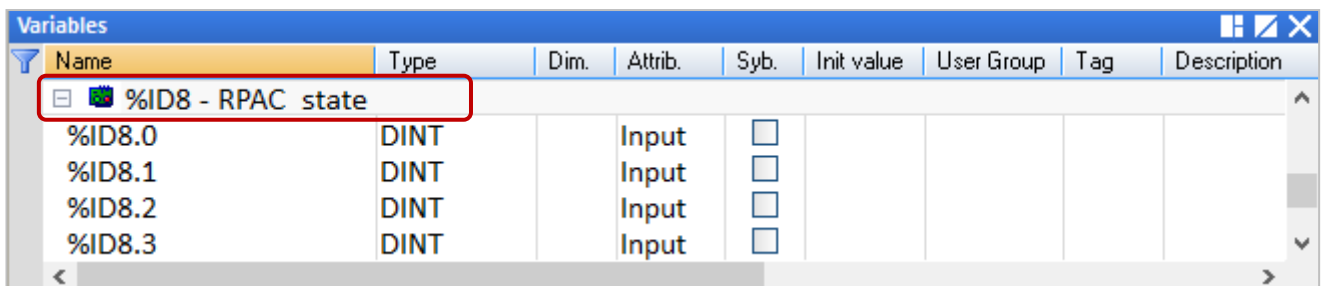
State:

bit0 ~ bit2: LAN1 ~ LAN3 的狀態，0: 連線正常，1: 失敗 或 未接線。

bit3: 硬體的狀態，0: 連線正常，1: 失敗。(像是 EEPROM、Disk、RAM... 的狀態)

bit4: IO 插槽的狀態，0: 連線正常，1: 失敗。(僅供有 Local I/O 的 PAC)

2. 在 “I/O Boards” 視窗內設定 “RPAC_PAC_state” 後，會自動在 “Variables” 視窗中新增 4 個 DINT 變數，當 Win-GRAF 有連上 PAC 時，會顯示 PAC 的狀態。

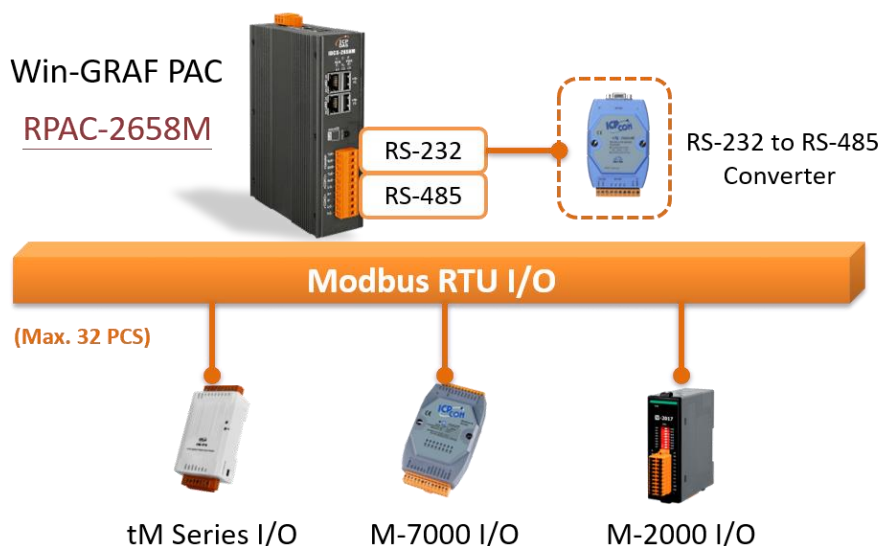


第 5 章 Modbus Master: 連接其它 Modbus Slave 設備

此章節將介紹如何啟用 Win-GRAF PAC 作為 Modbus Master 來連接 Modbus RTU/ASCII Slave 或 Modbus TCP/UDP Slave 設備。

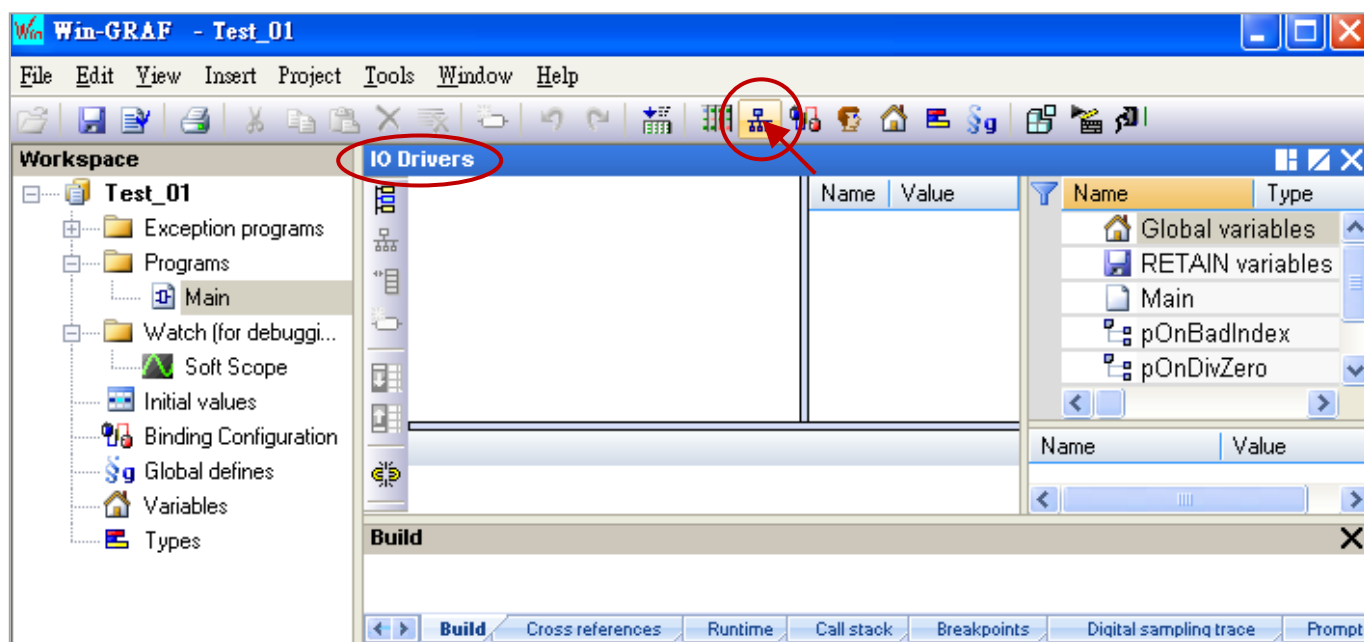
5.1 啟用 Win-GRAF PAC 為 Modbus RTU/ASCII Master

應用示意圖:

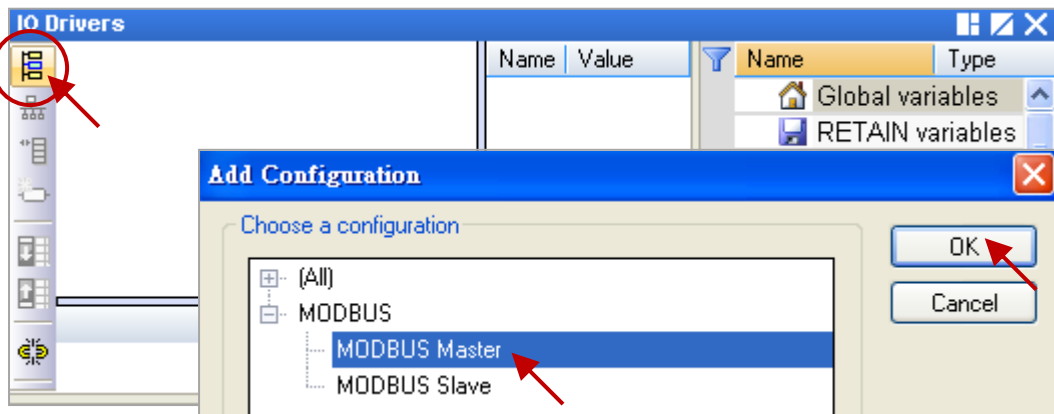


請依照以下步驟:

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



2. 設定 PAC 為 Modbus Master。點選 “IO Drivers” 視窗左側的 “Insert Configuration” 按鈕，再點選 “MODBUS Master” 並點選 “OK”。



3. 選用序列通訊埠。點選左側的 “Insert Master/Port” 按鈕，開啟設定視窗。接著，點選 “Serial MODBUS-RTU” 並設定 COM Port 與 Delay，再點選 “OK”。

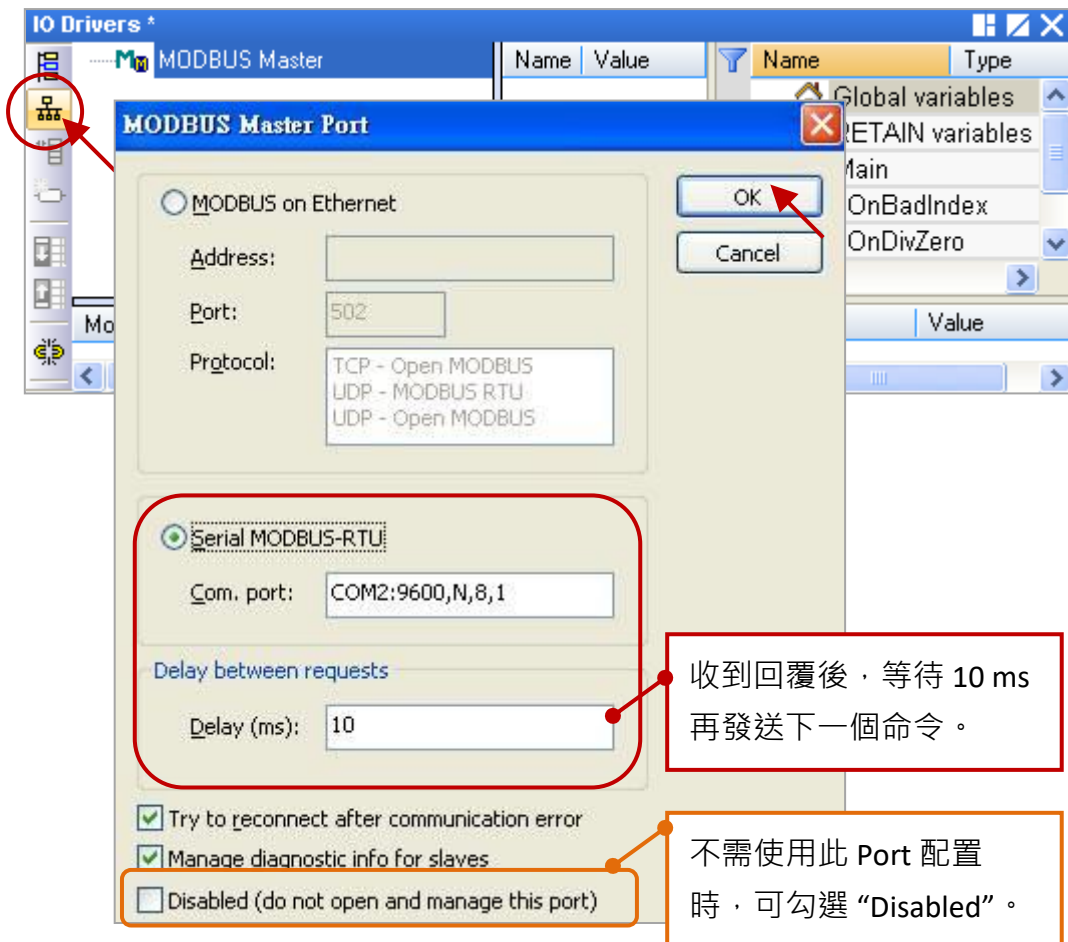
COM Port:

若使用 Modbus RTU 通訊協定，請填入 **COM2:9600,N,8,1**。

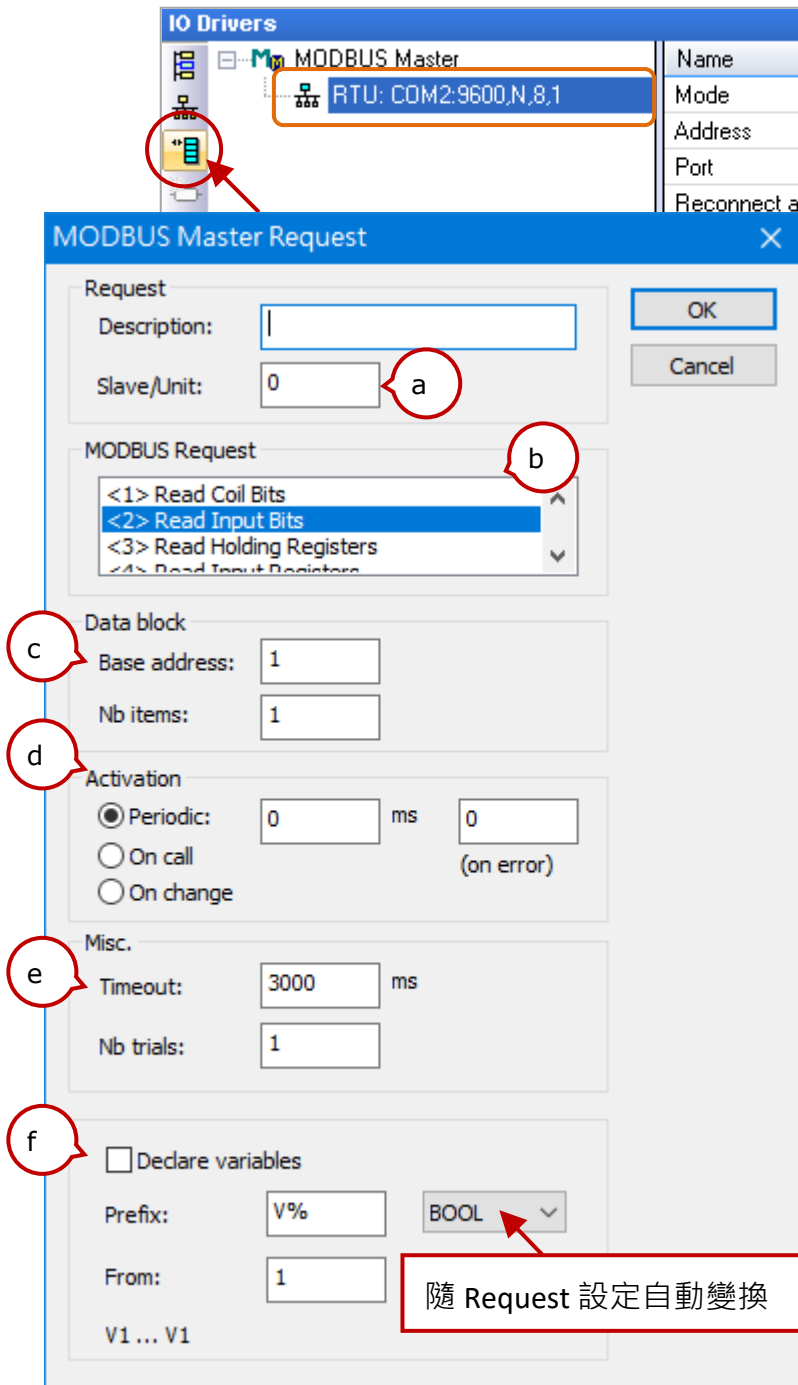
若使用 Modbus ASCII 通訊協定，請填入 **ASCII:COM2:9600,N,8,1**。

Delay (ms):

建議設為 10 (ms)，可設為 0 ~ 10000。



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



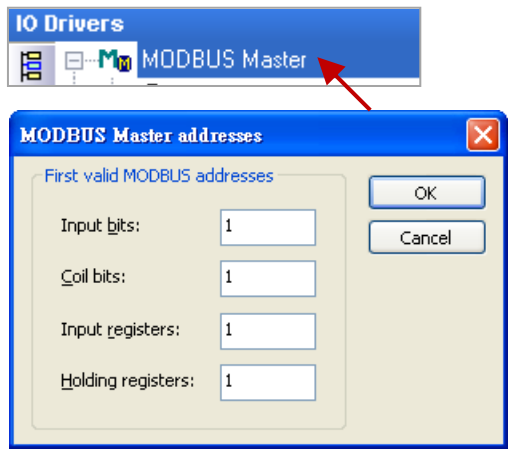
Name	Value
Mode	RTU
Address	COM2:9600,N,8,1
Port	502
Reconnect after error	<input checked="" type="checkbox"/>
Logistics	<input checked="" type="checkbox"/>
Response time requests (ms)	10
	<input type="checkbox"/>

- a. Slave/Unit:
填入 Slave 設備的站號。
- b. MODBUS Request:
讀/寫命令。
- c. Base address:
預設由 1 開始。
- Nb items:
讀/寫的資料數量。
- d. Activation: 發送命令的方式。
Periodic: 週期性的發送，“on error”表示異常後多久發送命令。
On call: 程式呼叫時，才發送命令。
On change:
資料有改變時，才發送命令。
- e. Timeout: 多久未回應，即表示異常。
(Modbus RTU/ASCII: 200 ~ 1000 ms)

f. 依據 Request 和 Nb items 宣告一個或多個連續變數。
若 Nb items 設為 5，勾選可新增 BOOL 變數 V1 ~ V5。

註:

- 如果必需修改“Base address”，可使用滑鼠右鍵點選“MODBUS Master”再選擇“MODBUS Master Addresses”修改其值。
- 您也在程式中使用"MBRTU_M_disable" 函式，來停用 Modbus RTU/ASCII Master Port。(參考 5.1.6 節)



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

Function Code	Modbus Request	說明
2	Read Input Bits	5.1.1 讀取 DI 資料
5	Write single coil bit	5.1.2 寫入 DO 資料
4	Read Input Registers	5.1.3 讀取 AI 資料
6	Write single holding register	5.1.4 寫入 AO 資料 (16-bit)
16	Write Holding Registers	5.1.5 寫入 AO 資料 (32-bit)

5.1.1 讀取 DI 資料

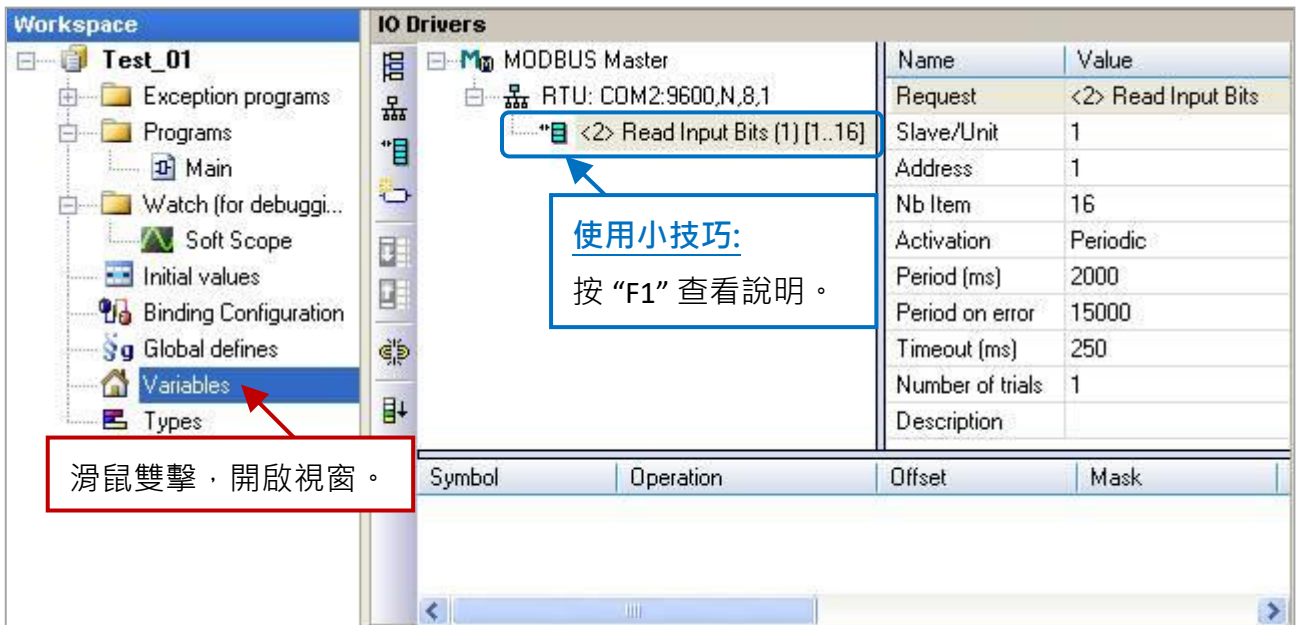
說明: 從設備 (Slave ID=1) 讀取 16 個 DI 值，每 2 秒發送一次命令，如有異常發生 (on error) 則 15 秒後再發送一次。250 ms 未回應會視為異常。

- 於 “MODBUS Master Request” 視窗，完成以下設定項目並點選 “OK”。



註: 細項說明可見 5.1 節 – 步驟 4。

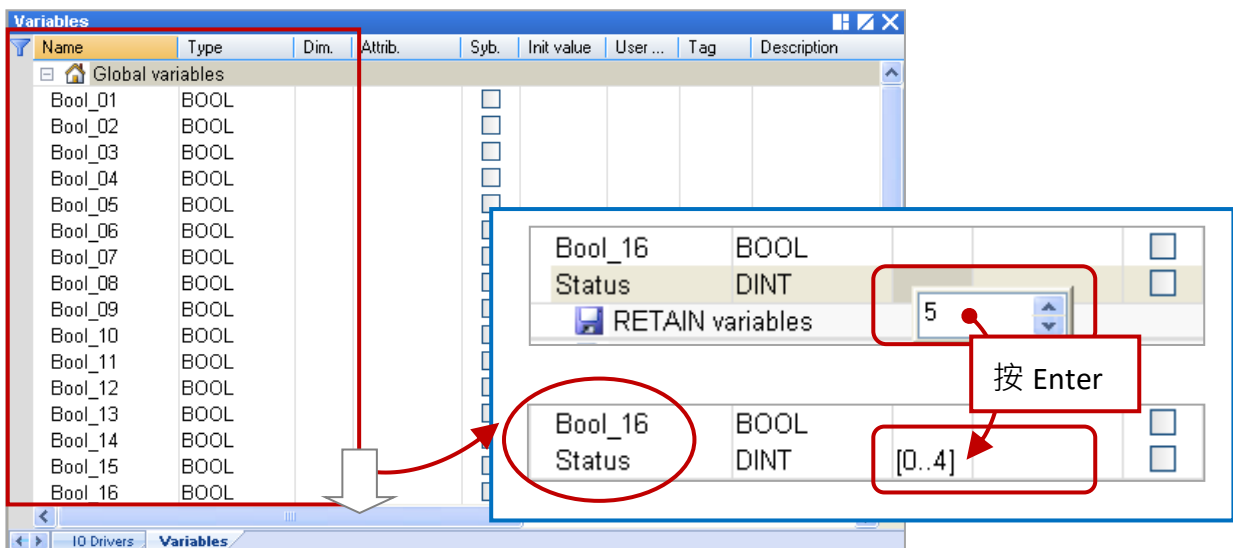
2. 接著，開啟 “Variables” 視窗，來新增需要的變數。



“**Boo_01 ~ Boo_16**” (BOOL): 新增 16 個布林變數，用來讀取資料。

“**Status**” (DINT, Dim. = 5): 新增 1 個陣列變數，用來顯示資料的存取狀況。

設定完成後，畫面如下。



3. 選取變數。

於 “IO Drivers” 視窗，將變數 “Boo_01” ~ “Boo_16” 與 “Status” 拖曳到 Data Block 的位址對應區。

注意: “Status” 是一個陣列變數，包含 “Status[0] ~ [4]”，位址對應區內只需保留 **Status[0]**。

4. 設定 Offset 位址。

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)。

5. 設定 Operation 狀態。

將 **Status[0]** 項目設為 “**Error report**”。表示讀取失敗時，會回傳 “Error Code”；讀取成功時，會重置為 “0”。

The screenshot shows the 'IO Drivers' configuration window for a MODBUS Master. The 'Request' is set to '<2> Read Input Bits (1) [1..16]'. The 'Name' column lists Bool_01 through Bool_16, and Status. The 'Operation' column is set to 'Data exchange' for all. The 'Offset' column is set to 0 for Status[0]. The 'Mask' column is set to FFFF for all. The 'Storage' column is set to 'Default' for all. The 'Range' column is set to 'Default' for all. A callout box labeled '1.' points to the 'Name' column. A callout box labeled '2.' points to the 'Offset' column. A callout box labeled '3.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '4.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '5.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '6.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '7.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '8.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '9.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '10.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '11.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '12.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '13.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '14.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '15.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '16.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '17.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '18.' points to the 'Offset' column, which is set to 0 for Status[0]. A callout box labeled '19.' points to the 'Operation' column, which is set to 'Error report' for Status[0]. A callout box labeled '20.' points to the 'Offset' column, which is set to 0 for Status[0].

設定為 “Error report” 時，
“Offset” 需是 “0”。

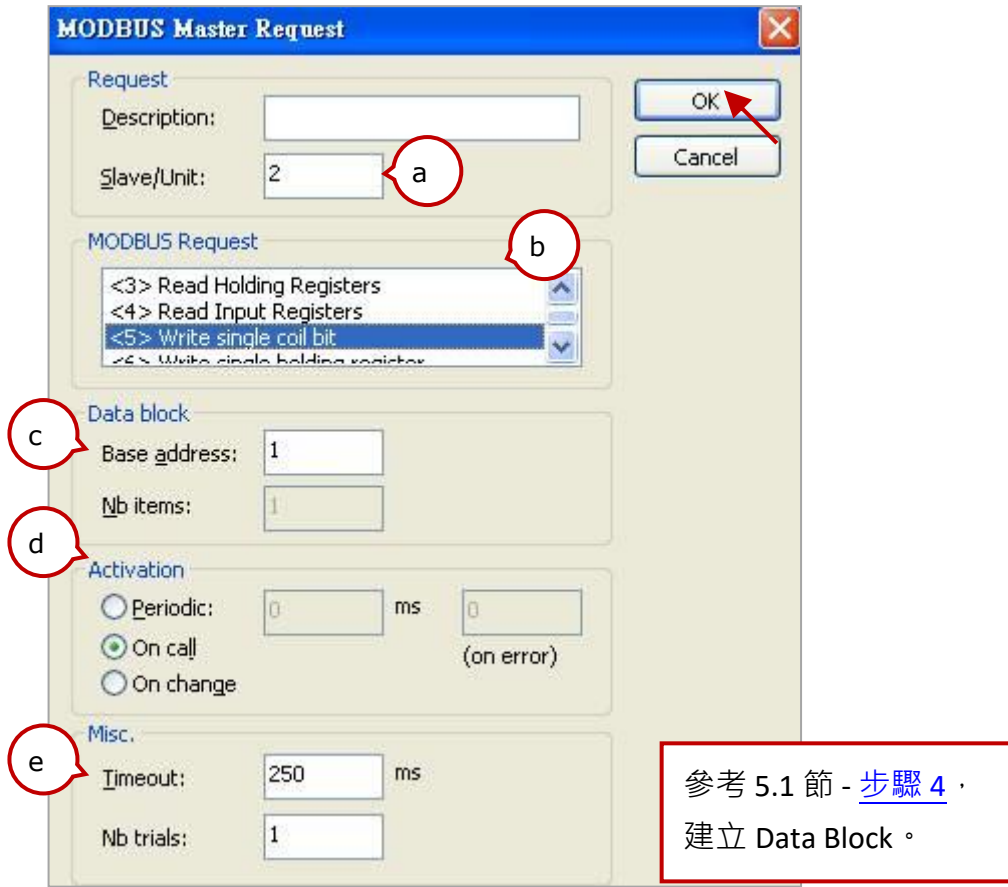
您可在此 “IO Drivers” 視窗中，按 “F1” 鍵了解 Modbus Master 設定的詳細資訊。

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。

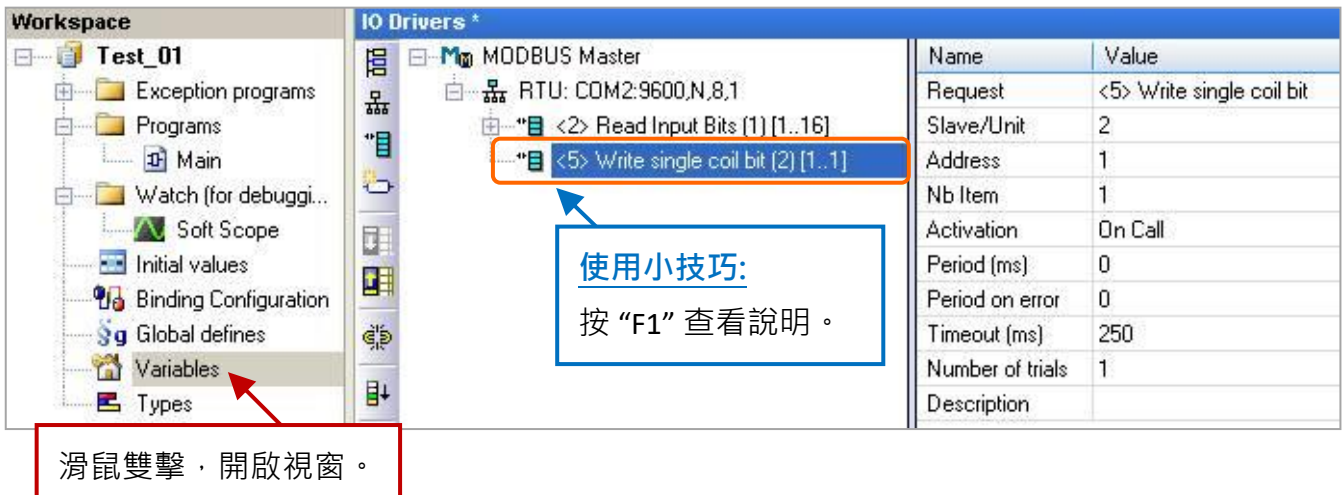
5.1.2 寫入 DO 資料

說明: 於設備 (Slave ID=2) 寫入一個 DO 值，程式呼叫時，才發送命令，250 ms 未回應會視為異常。

1. 於 “MODBUS Master Request” 視窗，完成以下設定項目並點選 “OK”。



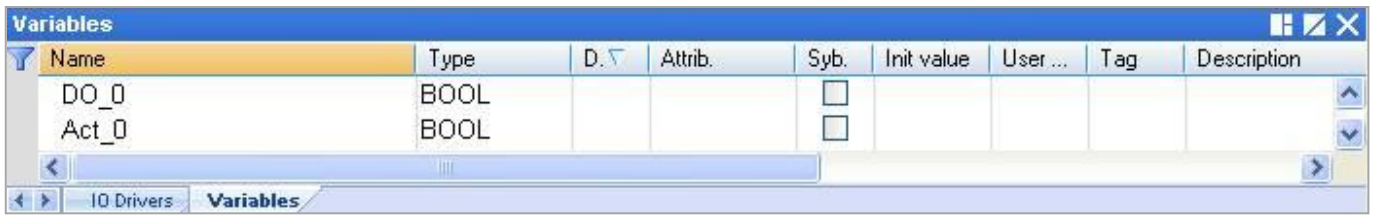
2. 開啟 “Variables” 視窗，新增所需的變數。



DO_0 (BOOL): 新增一個布林變數，用來寫入 DO 值。

Act_0 (BOOL): 新增一個布林變數，用來啟動 “On call” 程序。

設定完成後，畫面如下。



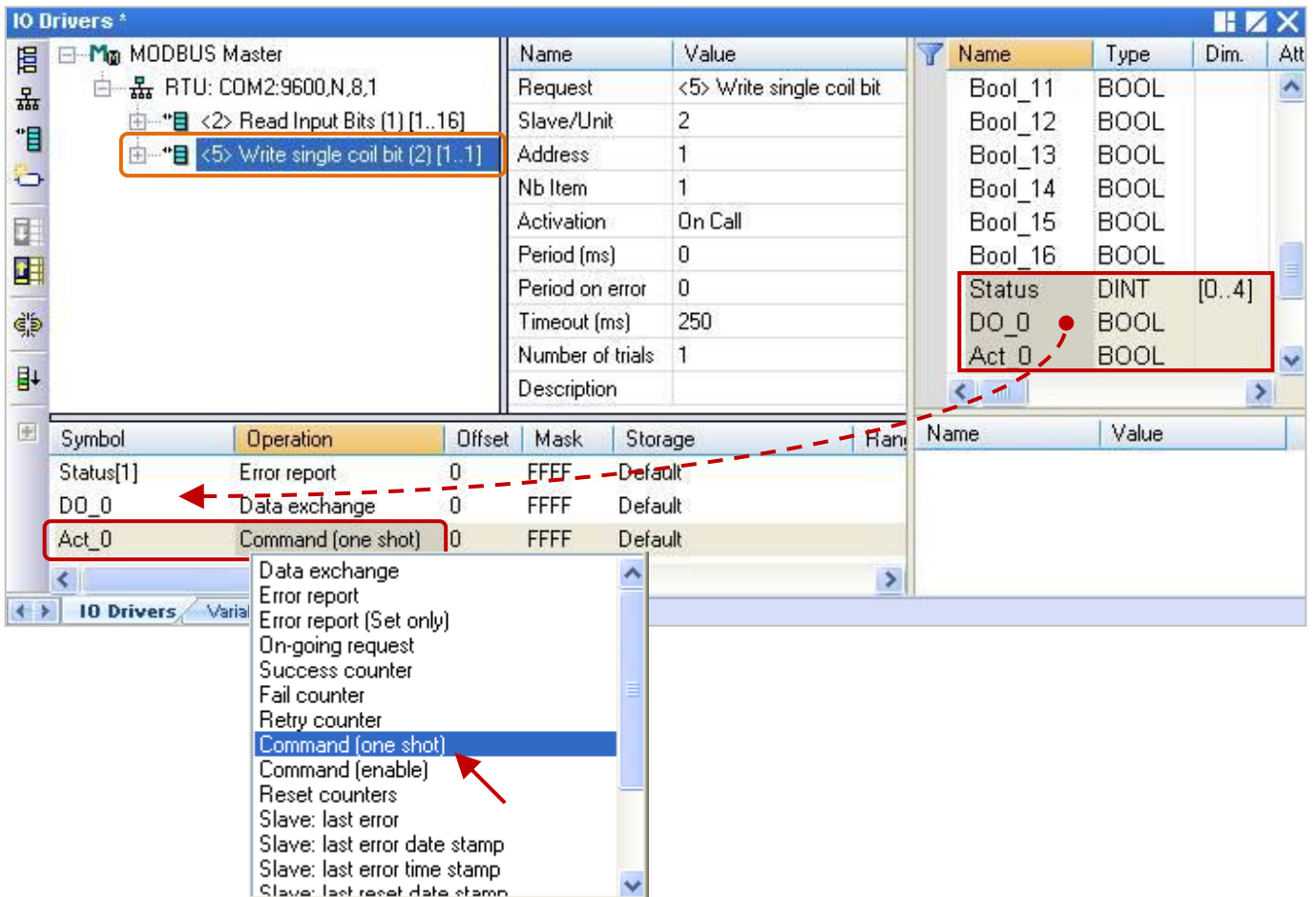
3. 選取變數。

於“IO Drivers”視窗，將變數“DO_0”、“Act_0”與“Status”拖曳到 Data Block 的位址對應區。

注意：“Status”是一個陣列變數，包含“Status[0]~[4]”，位址對應區內只需保留 **Status[1]**。

4. 設定 Operation 狀態。

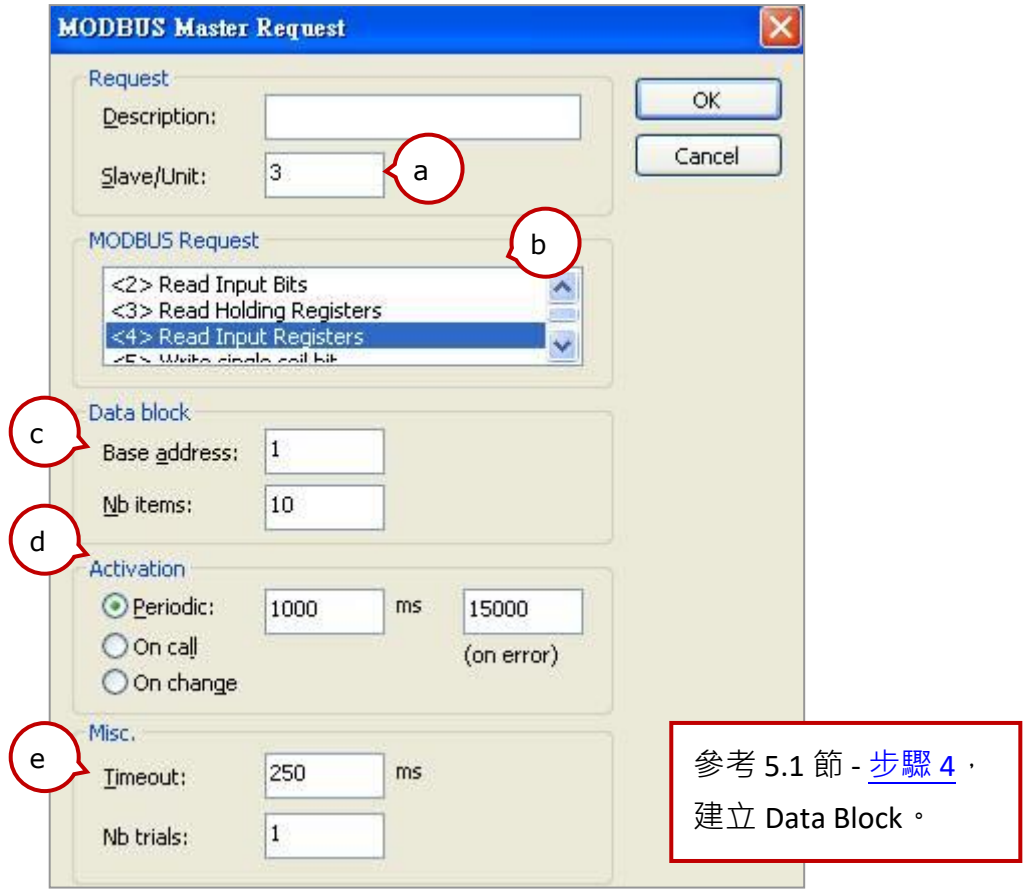
- 1) 將 **Status[1]** 的 Operation 設為“**Error report**”。表示讀取失敗時，會回傳“Error Code”；讀取成功時，會重置為“0”。
- 2) 將 **Act_0** 項目設為“**Command (one shot)**”，表示“Act_0”設為“TRUE”時，會發送指令一次，並自動重置為“FALSE”；若選用“Command (Enable)”，表示“Act_0”設為“TRUE”時，會輪流發送連續指令，直到“Act_0”設為“FALSE”時，才會停止發送指令。



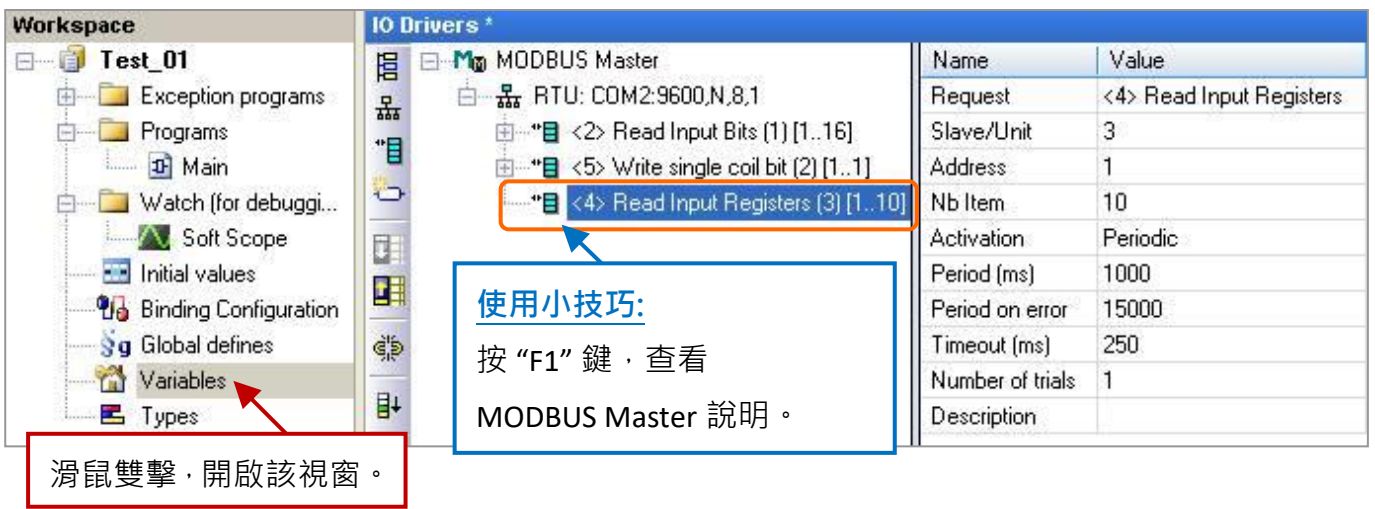
5.1.3 讀取 AI 資料

說明: 從設備 (Slave ID=3) 讀取 10 個 AI 值，每 1 秒發送一次命令，如有異常發生 (on error) 則 15 秒後再發送一次。250 ms 未回應會視為異常。

1. 於 “MODBUS Master Request” 視窗，完成以下設定項目並點選 “OK”。



2. 開啟 “Variables” 視窗，新增所需的變數。

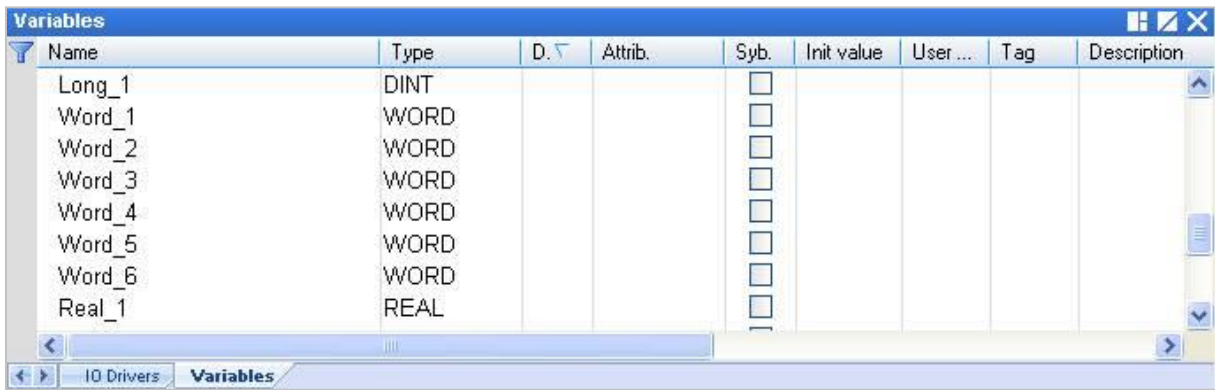


Word_1 ~ Word_6 (WORD): 新增 6 個 WORD (16-bit) 變數，用來讀取 AI 值。

Long_1 (DINT): 新增 1 個 DINT (32-bit) 變數，用來讀取 AI 值。

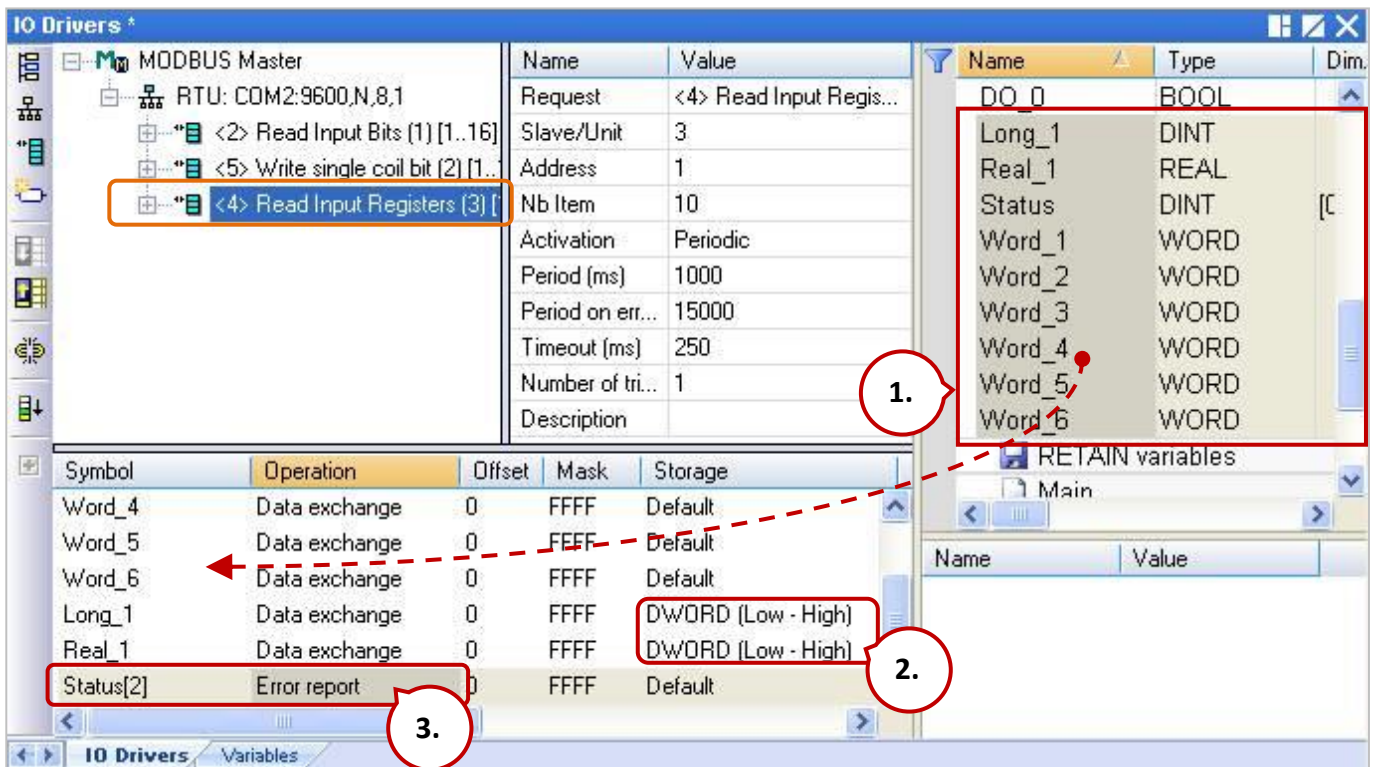
Real_1 (REAL): 新增 1 個 REAL (32-bit) 變數，用來讀取 AI 值。

設定完成後，畫面如下。



3. 選取變數。

於“IO Drivers”視窗，將變數 Word_1 ~ Word_6、“Long_1”、“Real_1”與“Status”拖曳到 Data Block 的位址對應區。**注意：**“Status”是一個陣列變數，包含“Status[0] ~ [4]”，位址對應區內只需保留 Status[2]。



4. 設定 Storage 格式。

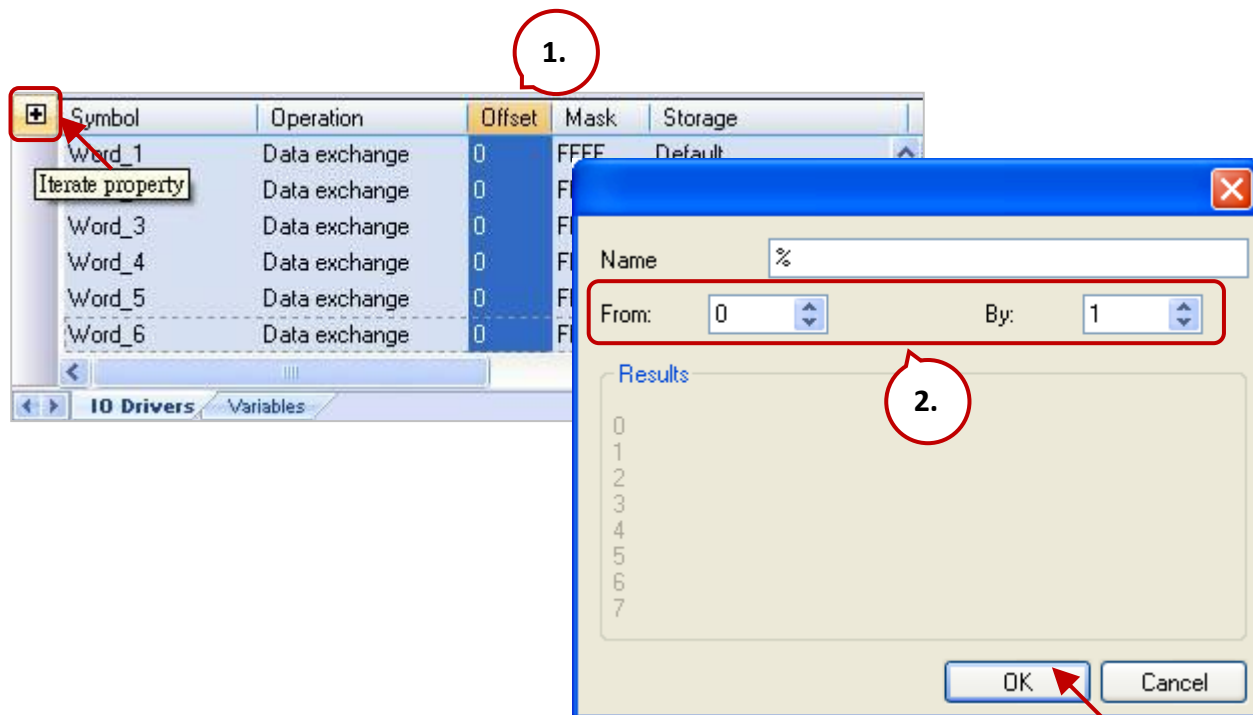
將 Long_1 與 Real_1 (32-bit) 的 Storage 設為“D'WORD (Low – High)”，對應 2 個 Word (16-bit)。

5. 設定 Operation 狀態。

將 Status[2] 的 Operation 設為“Error report”。表示讀取失敗時，會回傳“Error Code”；讀取成功時，會重置為“0”。

6. 設定 Offset 位址。

- 1) 選取 **Word_1 ~ Word_6** 的 Offset，並點選左側的“Iterate Property”按鈕，再設定 Offset 值 (From: 0 ; By: 1)。
- 2) 滑鼠雙擊 **Long_1 (32-bit)** 的 Offset，輸入 **6** 並按 **Enter** 鍵。
- 3) 滑鼠雙擊 **Real_1 (32-bit)** 的 Offset，輸入 **8** 並按 **Enter** 鍵。

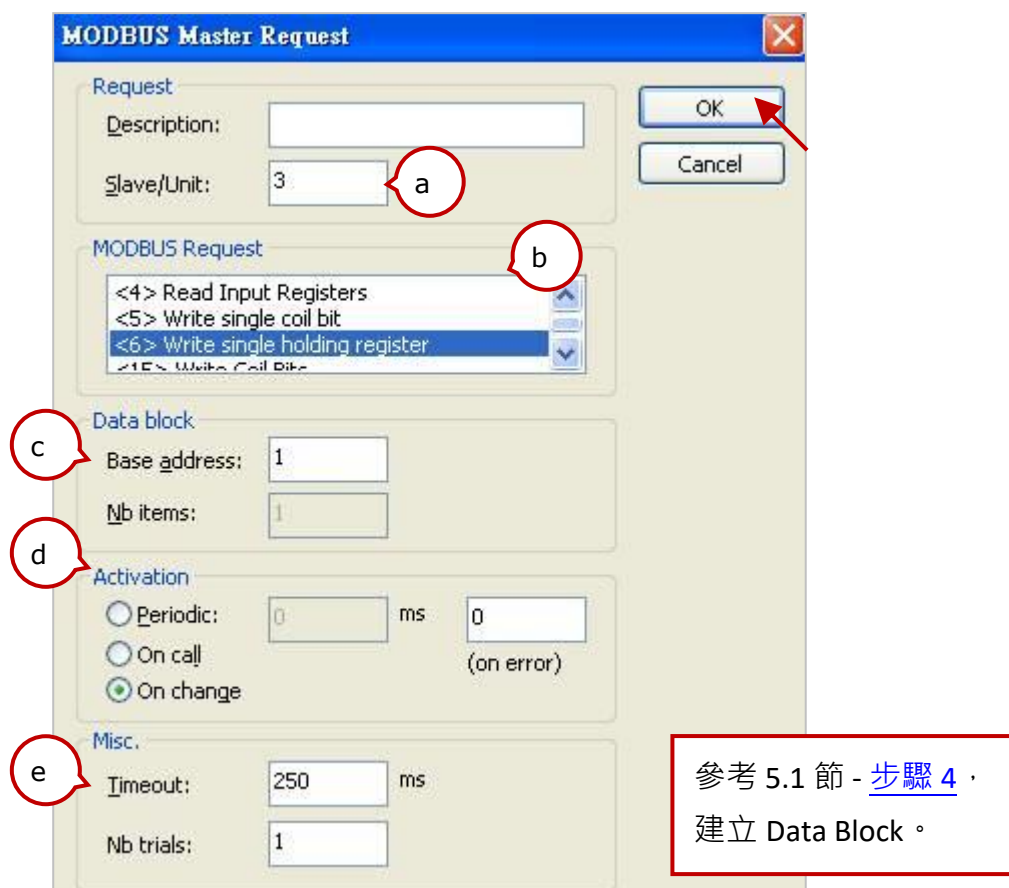


註: 32-bit 資料需占用 2 個 Modbus 位址。

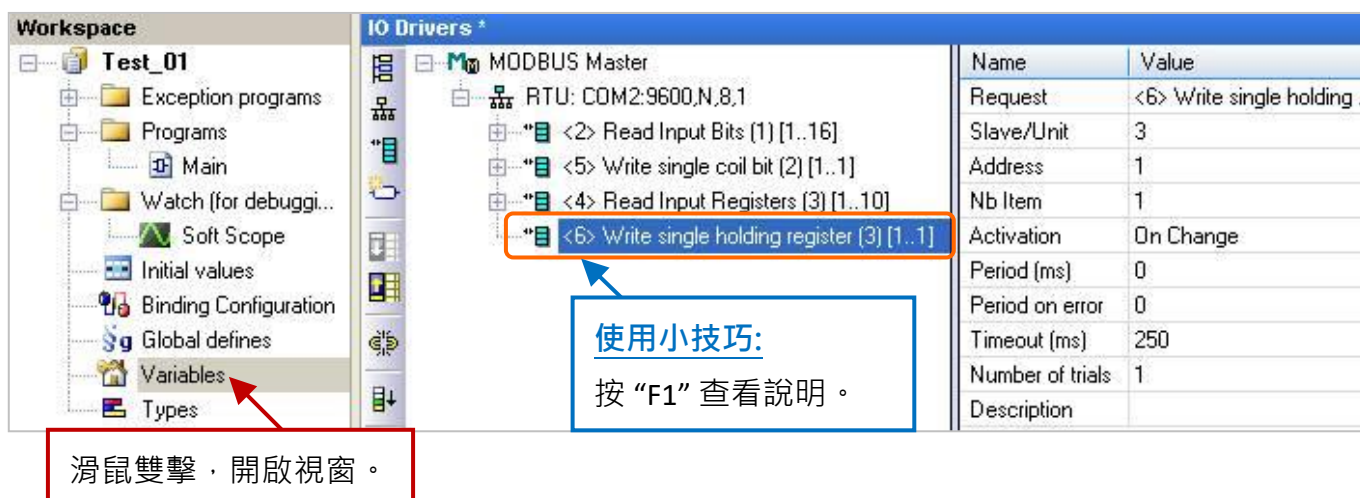
5.1.4 寫入 AO 資料 (16-bit)

說明: 於設備 (Slave ID=3) 寫入一個 AO 值，數值改變時，才發送命令，250 ms 未回應會視為異常。

1. 於 “MODBUS Master Request” 視窗，完成以下設定項目並點選 “OK”。

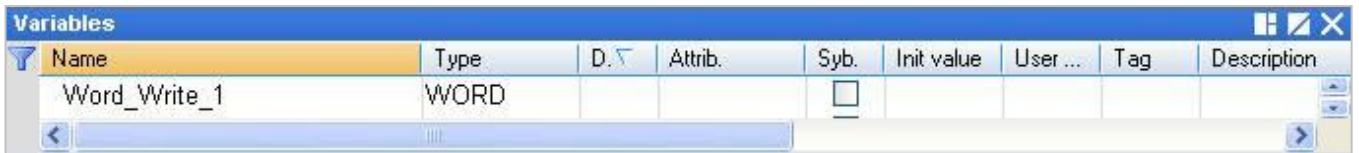


2. 開啟 “Variables” 視窗，設定需使用的變數。



Word_Write_1 (WORD): 新增 1 個 WORD (16-bit) 變數，用來寫入 AO 值。

設定完成後，畫面如下。



3. 選取變數。

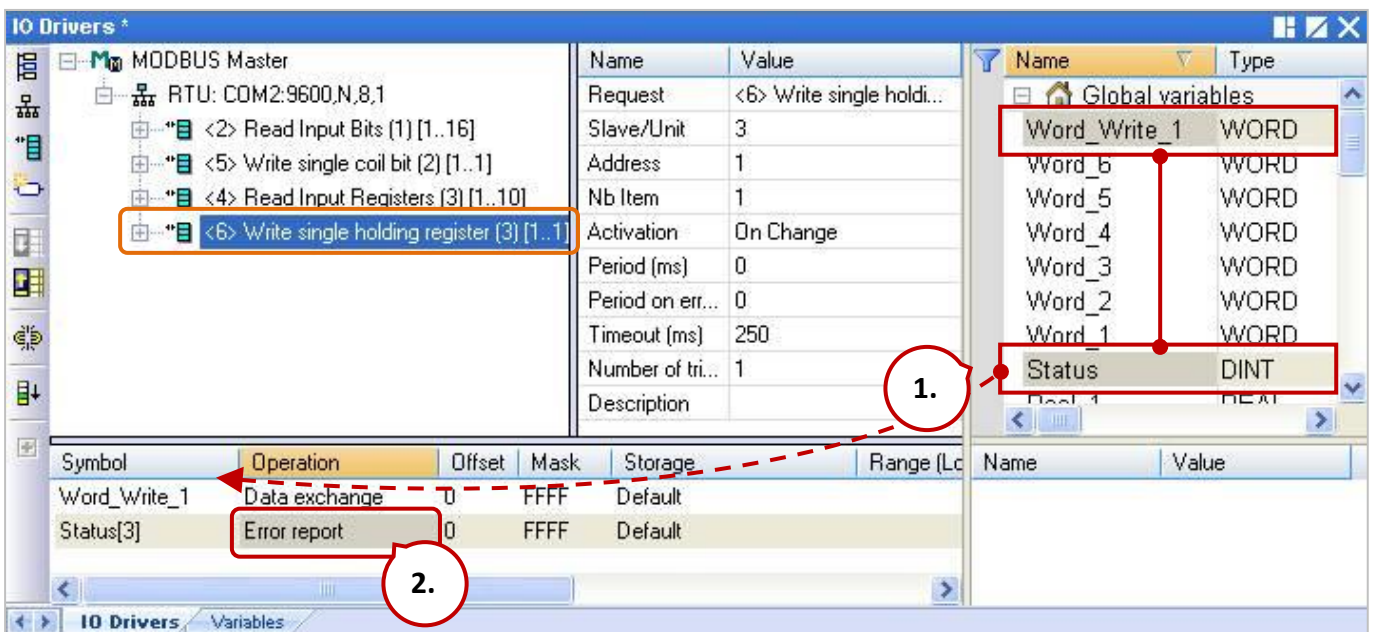
於“IO Drivers”視窗，將變數“Word_Write_1”與“Status”拖曳到 Data Block 的位址對應區。

注意:“Status”是一個陣列變數，包含“Status[0] ~ [4]”，位址對應區內只需保留 **Status[3]**。

4. 設定 Operation 狀態。

將 **Status[3]** 的“Operation”設為“**Error report**”。表示讀取失敗時，會回傳“Error Code”；

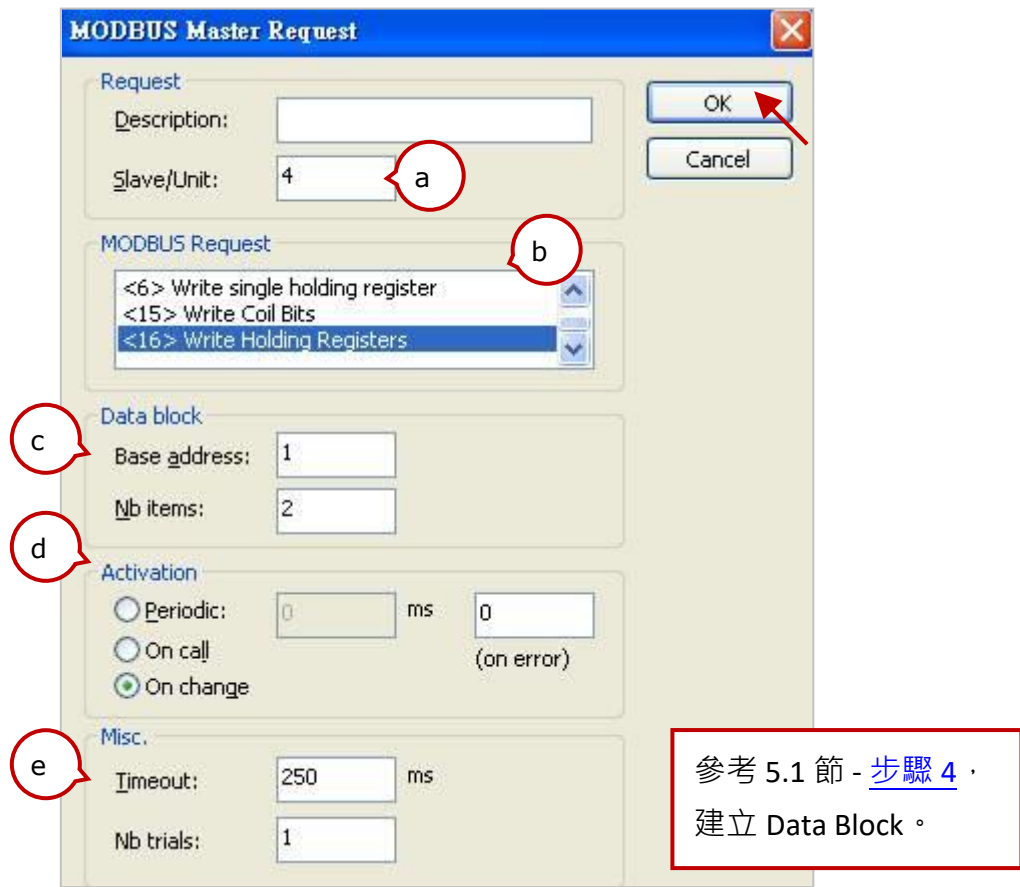
讀取成功時，會重置為“0”。



5.1.5 寫入 AO 資料 (32-bit)

說明: 於設備 (Slave ID=4) 寫入一個 AO 值，數值改變時，才發送命令，250 ms 未回應會視為異常。

1. 於 “MODBUS Master Request” 視窗，完成以下設定項目並點選 “OK”。

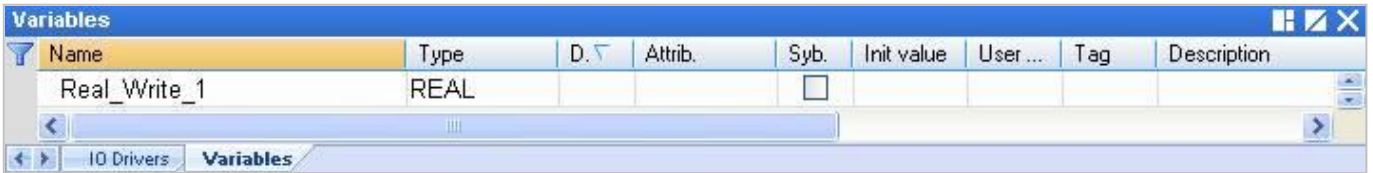


2. 開啟 “Variables” 視窗，設定需使用的變數。



Real_Write_1 (REAL): 新增 1 個 REAL (32-bit) 變數來寫入 AO 值。

設定完成後，畫面如下。



3. 選取變數。

於“IO Drivers”視窗，將變數“Real_Write_1”與“Status”拖曳到 Data Block 的位址對應區。

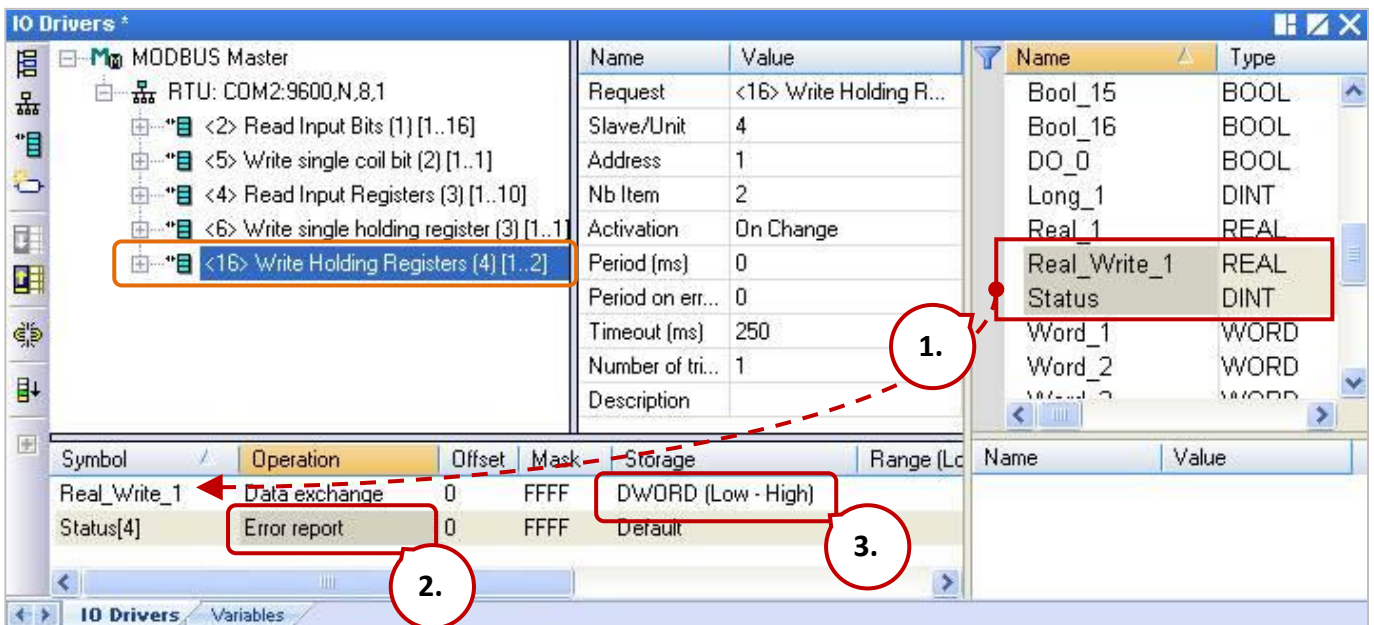
注意：“Status”是一個陣列變數，包含“Status[0] ~ [4]”，位址對應區內只需保留 **Status[4]**。

4. 設定 Operation 狀態。

將 **Status[4]** 的“Operation”設為“**Error report**”。表示讀取失敗時，會回傳 Error Code；讀取成功時，會重置為“0”。

5. 設定 Storage 格式。

將 **Real_Write_1** (32-bit) 的“Storage”設為“**DWORD (Low – High)**”，對應 2 個 Word (16-bit)。



5.1.6 停止/啟用 Modbus RTU/ASCII Master Port

每次 PAC 開機後，會自動啟用在 Win-GRAF "Fieldbus Configuration" 的 "IO Drivers" 畫面內設定的 Modbus **RTU/ASCII** Master Port。

若使用者想在程式運行中停用某個 Modbus Master Port 可以使用 "**MBRTU_M_disable**" 函式，如下：

```
(* To_disable 宣告為 BOOL *)  
If To_disable then  
  To_disable := FALSE ;  
  MBRTU_M_disable(2) ;  
End_if;
```

上方的程式碼中，當把 To_disable 設為 TRUE 時，會停用 COM2 這個 Modbus RTU/ASCII Master Port。之後若想再啟用它可以使用 "**MBRTU_M_enable**"，如下：

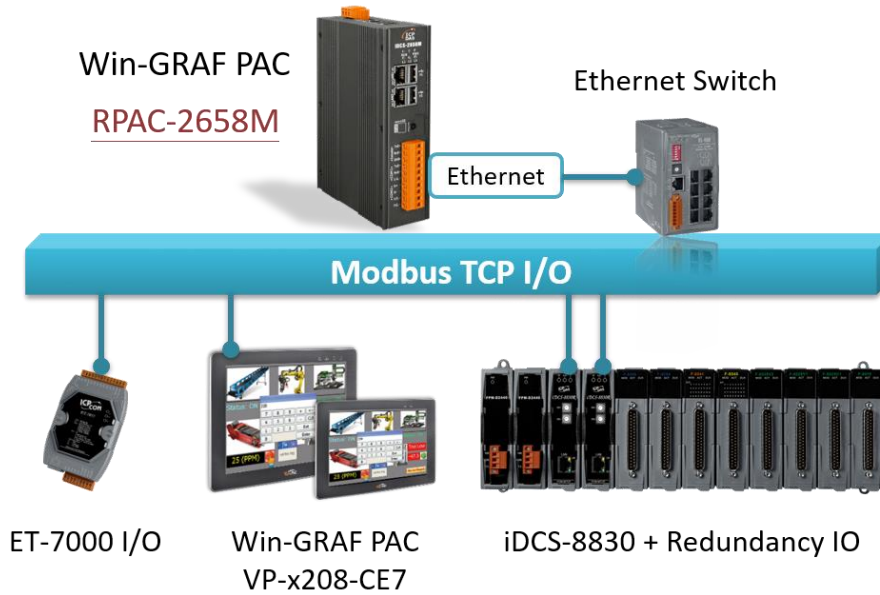
```
(* To_enable 宣告為 BOOL  
  Status_com2 宣告為 BOOL *)  
If To_enable then  
  To_enable := FALSE ;  
  MBRTU_M_enable(2) ;  
End_if;  
Status_com2 := MBRTU_M_status(2) ;
```

上方的 "**MBRTU_M_status**" 函式是用來取得 Modbus RTU/ASCII Master Port 的狀態。

True: 表示啟用 ; False: 表示停用

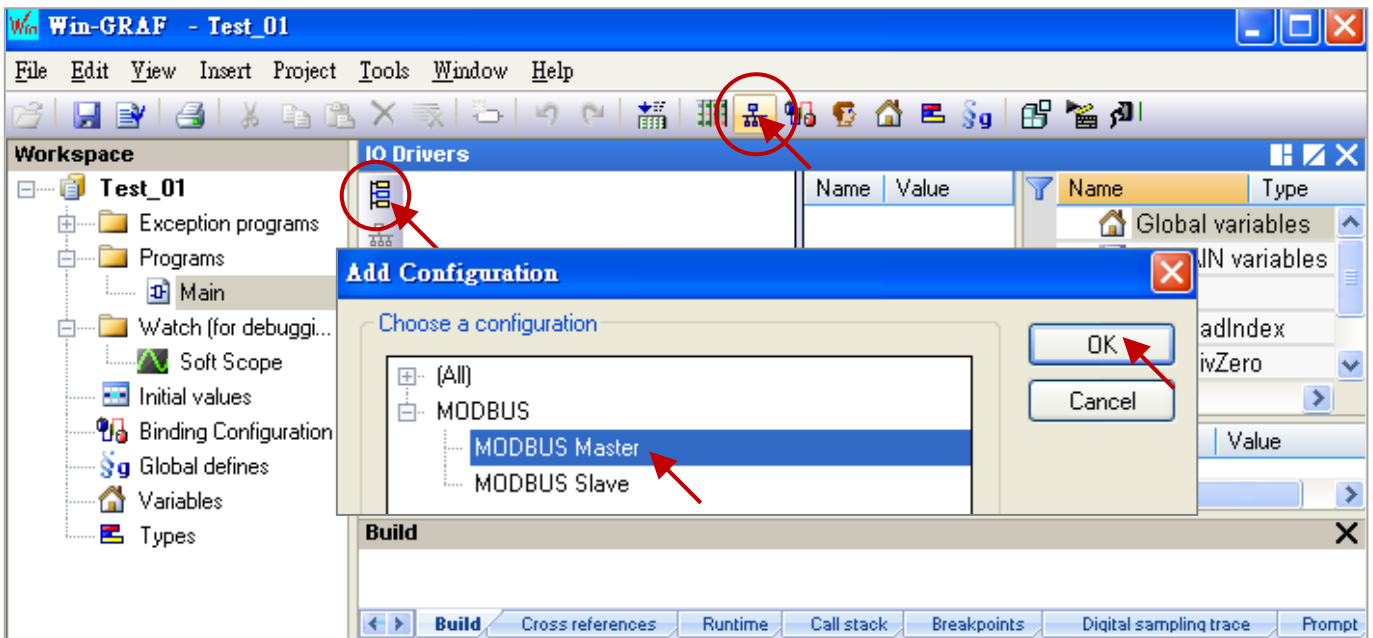
5.2 啟用 Win-GRAF PAC 為 Modbus TCP/UDP Master

應用示意圖：



請依照以下步驟：

1. 滑鼠點選工具列上的“Open Fieldbus Configuration” 按鈕來開啟“IO Drivers” 視窗。
2. 設定 PAC 為 **Modbus Master**。點選“IO Drivers” 視窗左側的“Insert Configuration” 按鈕，再點選“MODBUS Master” 並點選“OK”。



註： Modbus Master 設定內，可使用多個 Modbus TCP/UDP 或 Modbus RTU/ASCII Port。
若您想在程式運行時，停用此 Modbus TCP/UDP Port，請參考 5.2.4 節。

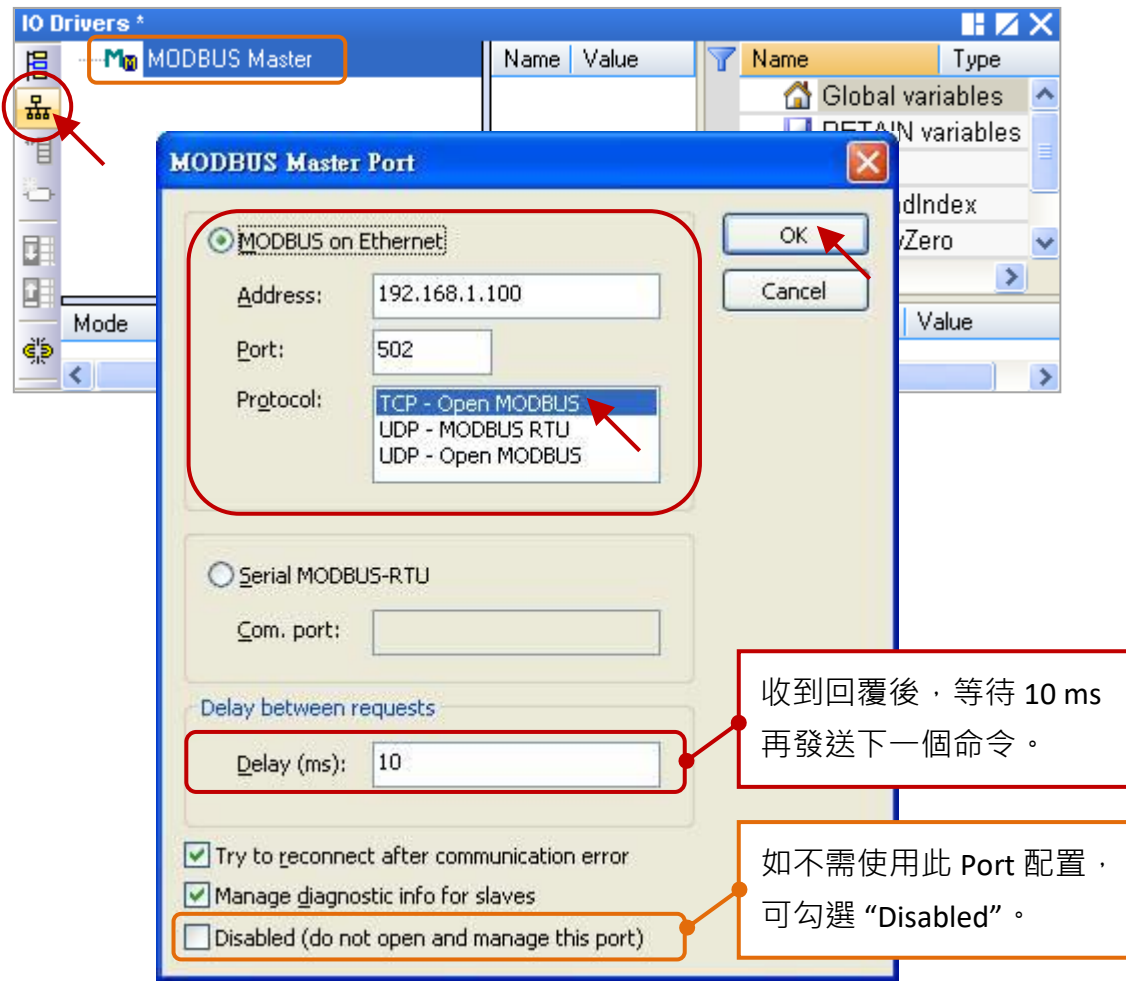
3. 選用 **Ethernet 通訊埠**。點選左側的 “Insert Master/Port” 按鈕，開啟設定視窗。接著，點選 “MODBUS on Ethernet” 並設定 Address、Port、Protocol 與 Delay，再點選 “OK”。

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

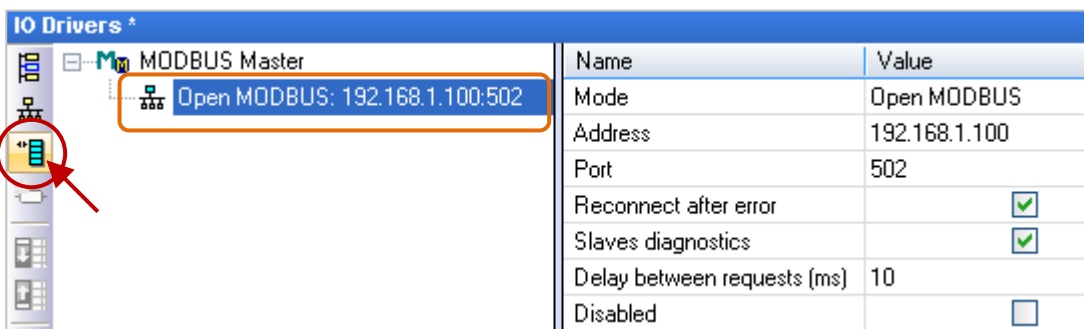
Port: 填入 Modbus TCP Slave 設備的 Port 編號，一般是 “502”。

Protocol: 若使用 Modbus TCP，請選擇 “TCP – Open MODBUS”。
若使用 Modbus UDP，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)。



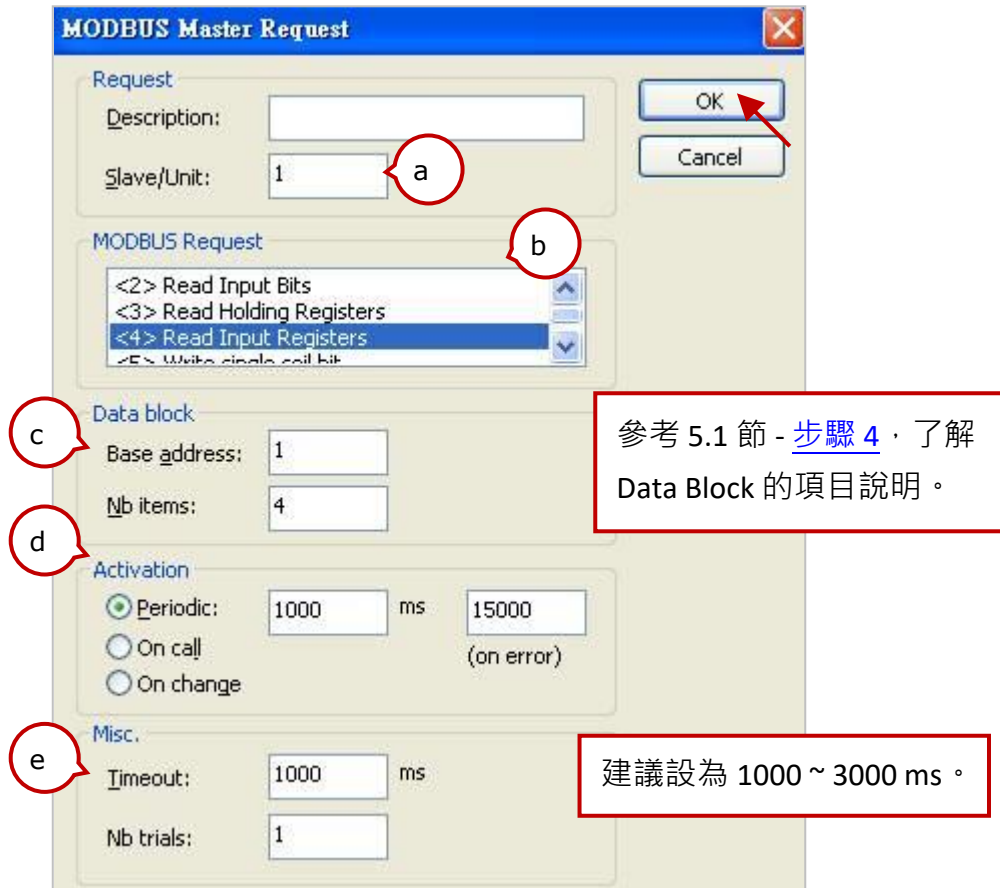
4. 建立 **Data Block**。點選左側的 “Insert Slave/Data Block” 按鈕，開啟設定視窗。



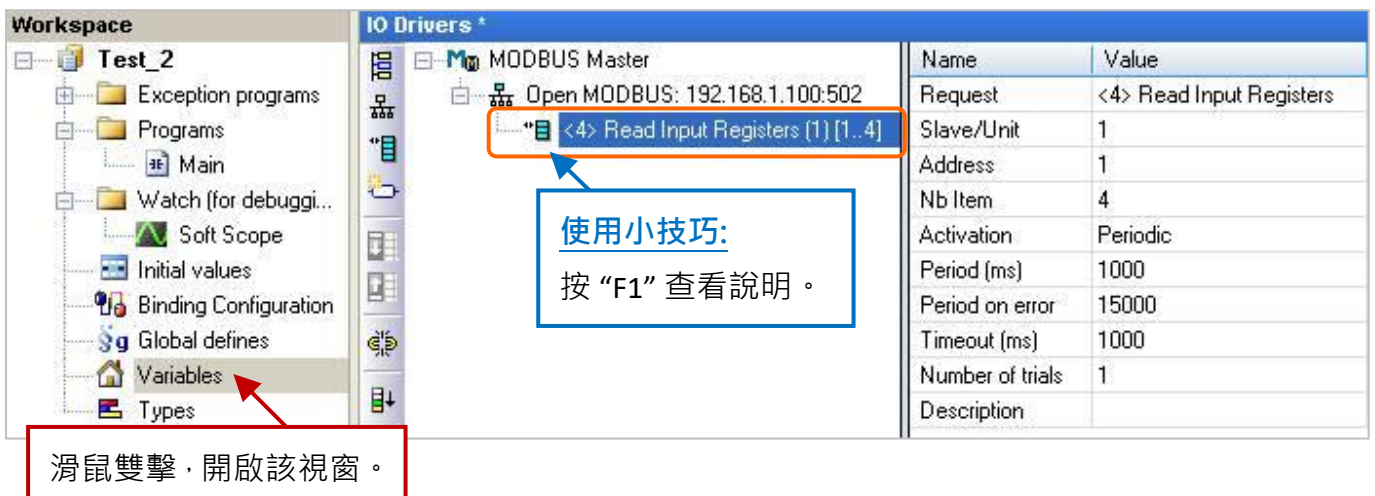
讀取 AI 資料

從設備 (Slave ID=1) 讀取 4 個 AI 值，每 1 秒發送一次命令，如有異常發生 (on error) 則 15 秒後再發送一次。1 秒未回應會視為異常。

1. 於 “MODBUS Master Request” 視窗，完成以下設定項目並點選 “OK”。



2. 開啟 “Variables” 視窗，新增所需的變數。



“Word_1 to Word_4” (WORD): 新增 4 個 WORD (16-bit) 變數，來讀取 AI 值。

“Status” (DINT, Dim.=5): 新增 1 個陣列變數，來顯示資料的存取狀況。

設定完成後，畫面如下。

Name	Type	Dim.	Attrib.	Syb.	Init value	User ...	Tag	Description
Status	DINT	[0..4]		<input type="checkbox"/>				
Word_1	WORD			<input type="checkbox"/>				
Word_2	WORD			<input type="checkbox"/>				
Word_3	WORD			<input type="checkbox"/>				
Word_4	WORD			<input type="checkbox"/>				

3. 選取變數。

於“IO Drivers”視窗，將變數“Word_1 ~ Word_4”與“Status”拖曳到 Data Block 的位址對應區。

注意：“Status”是一個陣列變數，包含“Status[0] ~ [4]”，位址對應區內只需保留 **Status[0]**。

4. 設定 Operation 狀態。

將 **Status[0]** 的“Operation”設為“**Error report**”。表示讀取失敗時，會回傳 Error Code；讀取成功時，會重置為“0”。

5. 設定 Offset 位址。

選取 **Word_1 ~ Word_4** 的 Offset，並點選左側的“Iterate Property”按鈕，再設定 Offset 值 (From: 0；By: 1)。

Symbol	Operation	Offset	Mask	Storage	Range (Low)	Range (High)
Status[0]	Error report	0	FFFF	Default		
Word_1	Data exchange	0	FFFF	Default		
Word_2	Data exchange	1	FFFF	Default		
Word_3	Data exchange	2	FFFF	Default		
Word_4	Data exchange	3	FFFF	Default		

Dialog Box: Iterate Property

Name: %

From: 0 By: 1

Results:

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7

Buttons: OK, Cancel

註：請使用類似的方法建立 Data Block 來讀/寫 IO 值，也可參考 5.1.1 ~ 5.1.5 節。

5.2.1 連接 ET-7000 系列 I/O 模組

ET-7000 系列是一款支援 Web_based 介面的 Ethernet I/O 模組，您可將 Win-GRAF PAC 設置為 Modbus TCP Master 來連接多個 ET-7000 模組。建議一台 RPAC-2658M 最多可連接 100 個 ET-7000 模組。

關於更多 ET-7000 的產品訊息請參訪：

https://www.icpdas.com/en/product/guide+Remote_I_O_Module_and_Unit+Ethernet_I_O_Modules+ET-7000_ET-7200

設定 ET-7000 模組

使用 ET-7000 模組前，需使用網頁瀏覽器進行設定。ET-7000 出廠預設值為 IP = 192.168.255.1，Mask = 255.255.0.0。請確認模組和您的 PC 都在同一個網段內 (例如，IP = 192.168.255.100，Mask = 255.255.0.0)。

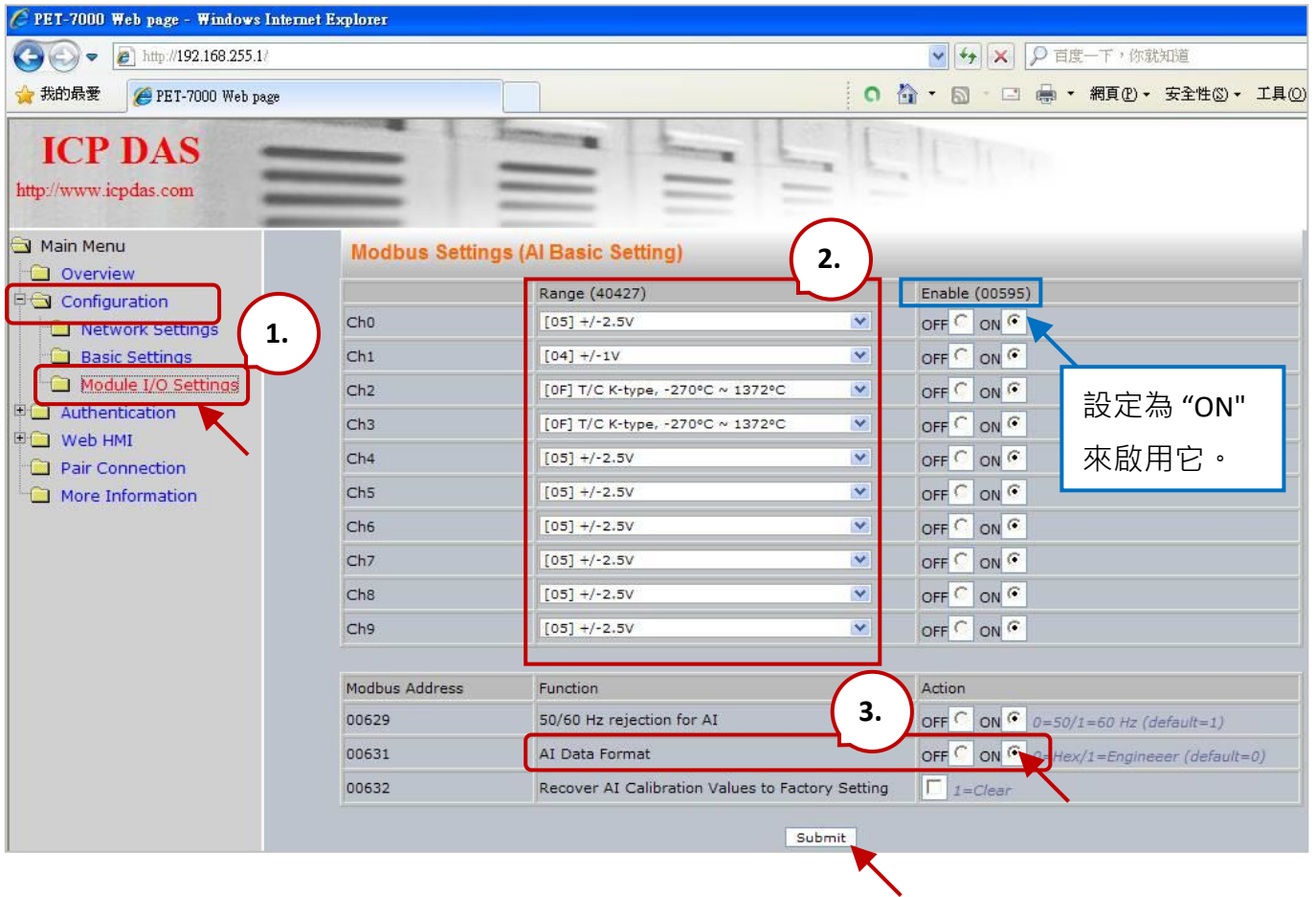
注意: ET-7000 後方的指撥開關 (Dip Switch) 必須保持在 “Normal” 位置。

1. 開啟瀏覽器，輸入該 ET-7000 的 IP 來連上它。



2. 輸入使用者名稱與密碼，來登入 ET-7000。





3. 點選 Configuration 目錄下的 Module I/O Settings，來設定通道的資料範圍 (Range)，再點選 "Submit"。使用 AO 模組時 (例如: ET-7018Z)，請設定 "AI Data Format" 為 "ON" (Engineering)。

例如:

數值 -10000 ~ +1000 表示 **-1 ~ +1 V**。

數值 -25000 ~ +25000 表示 **-2.5 ~ +2.5 V**。

數值 258 表示 **25.8 (°C)**。

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

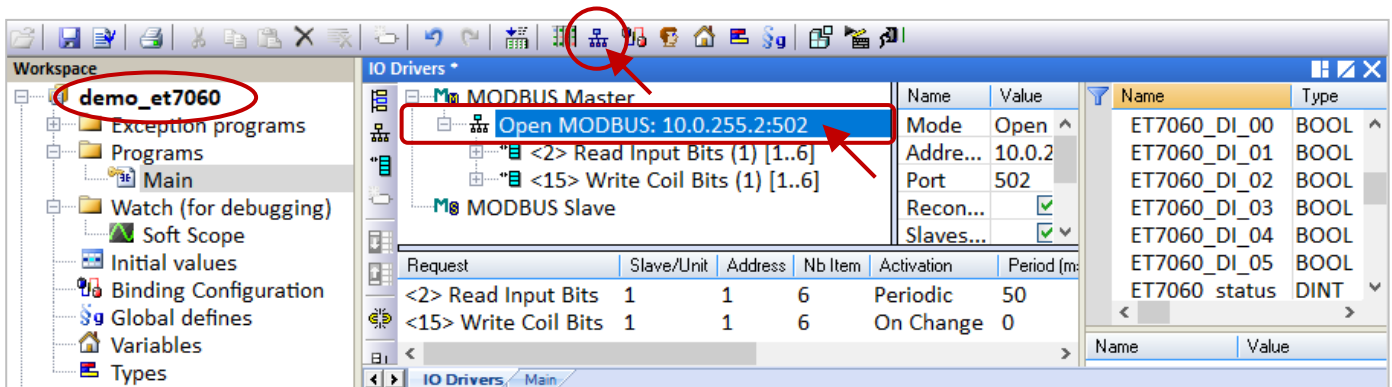
回存 並 修改 Win-GRAF 專案:

您可在 [網站](#) 下載此章節說明的 Win-GRAF [範例程式](#):

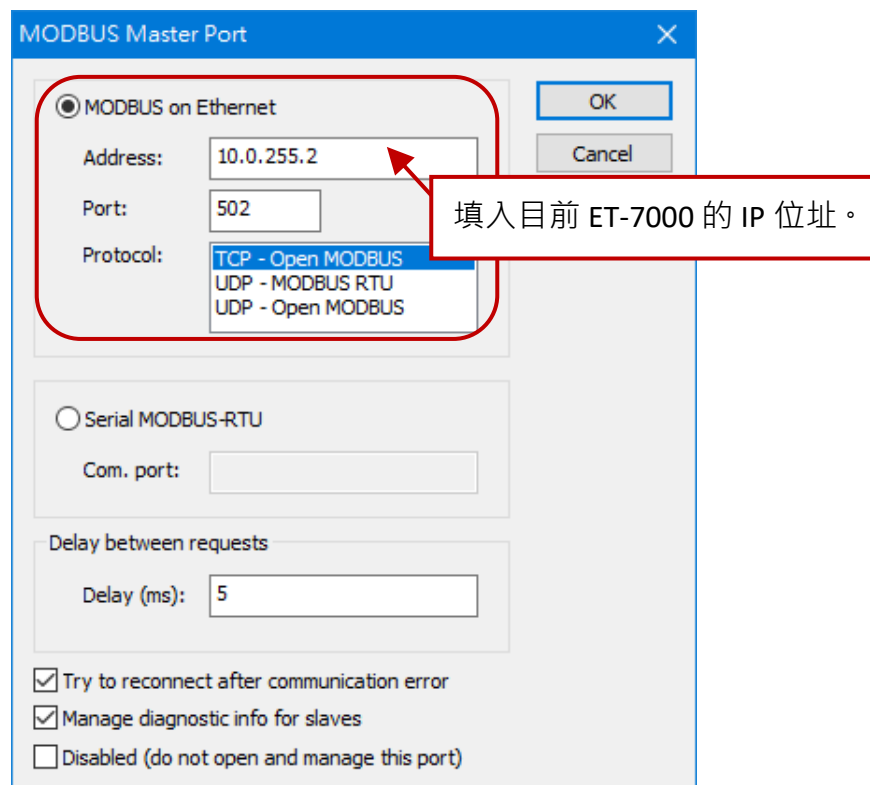
點選 "File - Add Existing Project - From Zip" 將專案回存到 Win-GRAF workbench (可參考第 11.4 節)。

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

1. 滑鼠點選 "Open Fieldbus Configuration" 按鈕，來開啟 "IO Drivers" 視窗。



2. 滑鼠雙擊 "Open Modbus: IP:502" 可開啟 "MODBUS Master Port" 視窗。此例，啟用 Modbus TCP Master 來連接 ET-7000，請填入 ET-7000 的 IP 位址、"Port" 固定是 "502"、"Protocol" 選擇 "TCP - Open Modbus"。



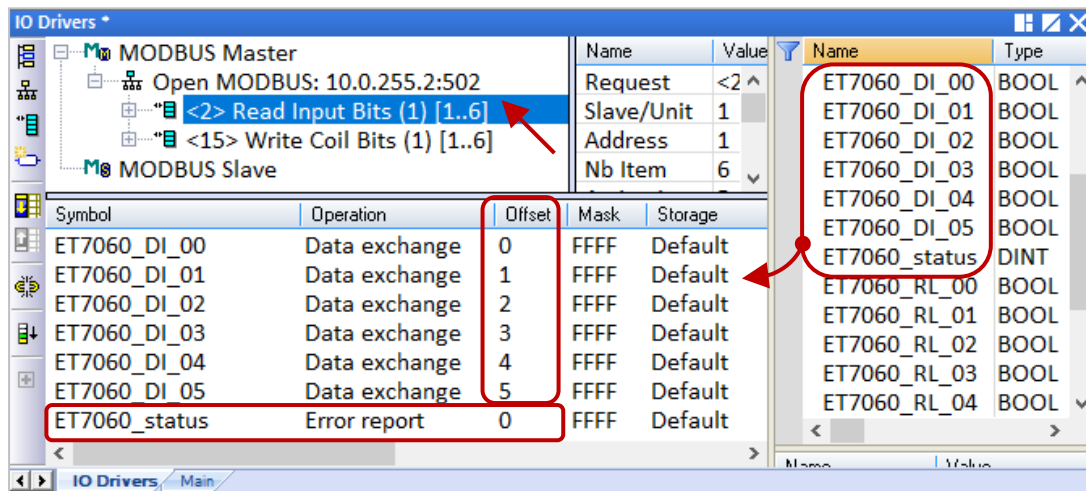
5.2.2 連接 ET-7060 (6 DI, 6 Relay Output)

ET-7060 是一款具有 6 DI 與 6 Relay Output 的 Ethernet I/O 模組，本章節說明的 Win-GRAF 範例程式為 "demo_ET7060.zip"。使用模組前，請確認您已完成 [5.2.1 節](#) 中說明的模組與 Port 設定。

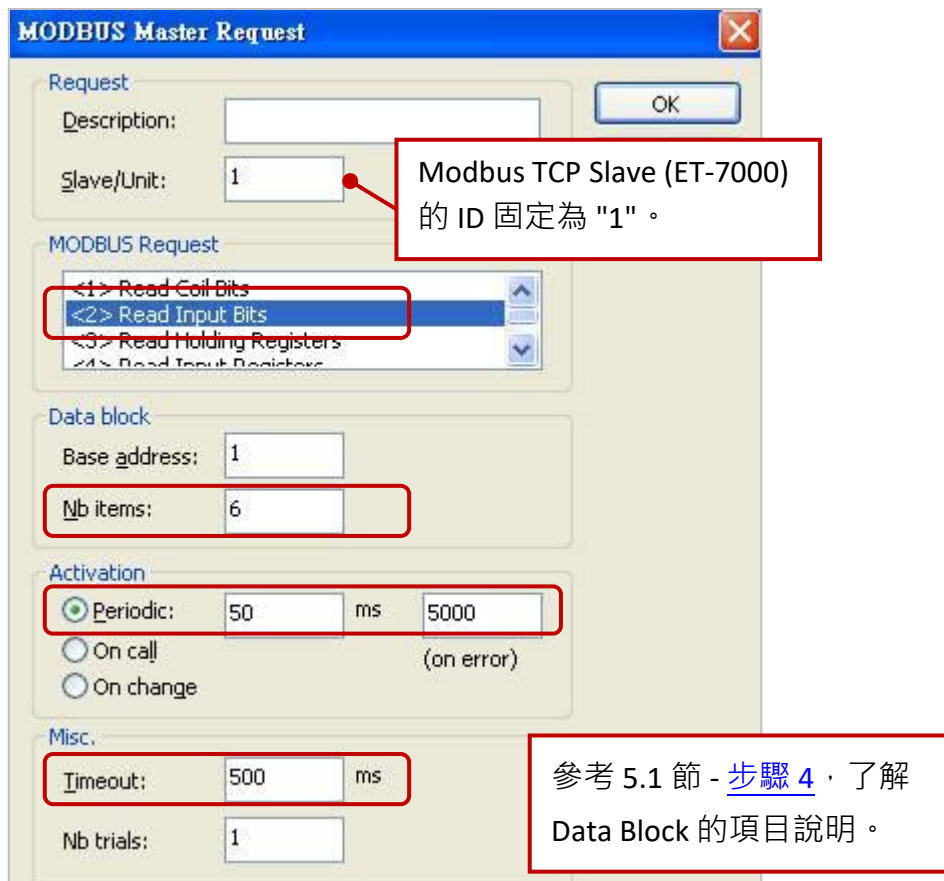
範例說明:

透過 RPAC 的 LAN3 來連接 ET-7060，此範例建立了 2 個 Data Block，一個用來讀取 6 個 DI 資料，另一個用來寫入 6 個 DO 資料。

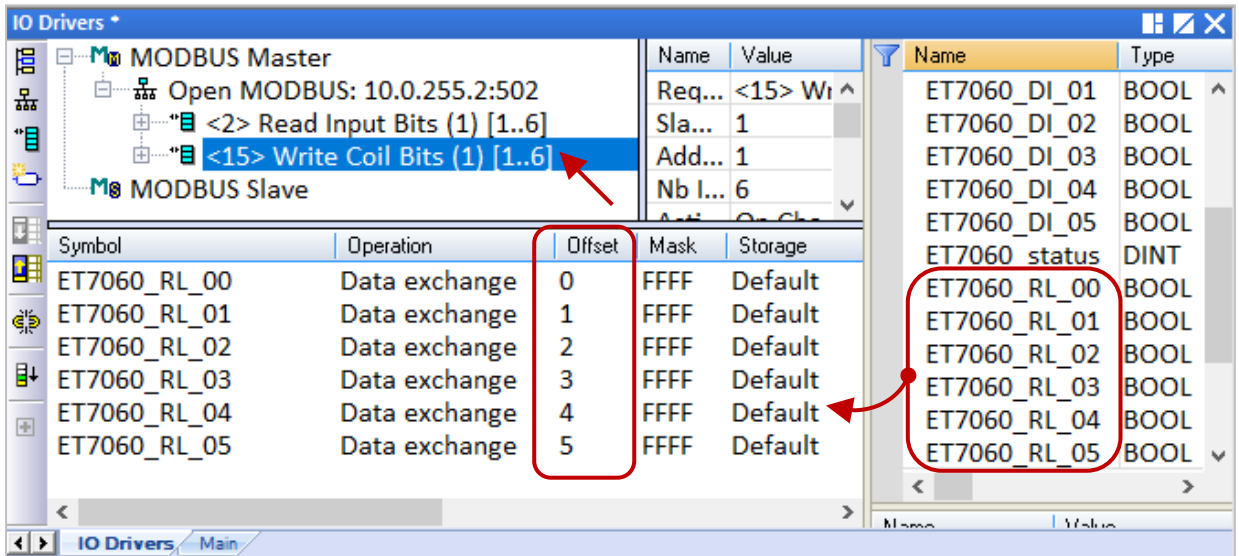
1. 點選 (或滑鼠雙擊) 第 1 個 Data Block (即，<2> Read Input Bits) 來查看設定。



說明: 從設備 (Slave ID=1) 讀取 6 個 DI 值，每 50 ms 發送一次命令，如有異常發生 (on error) 則 5 秒後再發送一次。500 ms 未回應會視為異常。



2. 點選 (或滑鼠雙擊) 第 2 個 Data Block (即 · <15> Write Coil Bits) 來查看設定。



說明: 於設備 (Slave ID=1) 寫入 6 個 DO 值，當資料改變時，發送一次命令。500 ms 未回應會視為異常。



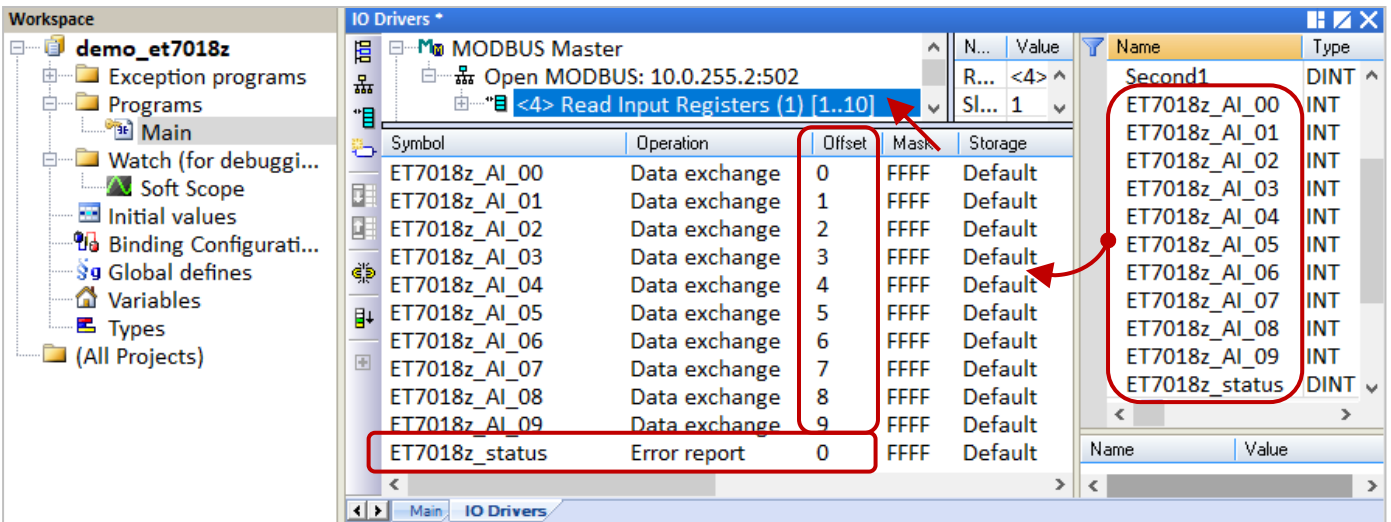
5.2.3 連接 ET-7018Z (10 AI)

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_ET7060.zip"。使用前，請確認您已完成 [5.2.1 節](#) 中說明的模組與 Port 設定。

範例說明:

透過 RPAC 的 LAN3 來連接 ET-7018Z，此範例建立了 1 個 Data Block，用來讀取 10 個 AI 資料。

1. 點選 (或滑鼠雙擊) Data Block (即，<4> Read Input Registers) 來查看設定。



說明: 從設備 (Slave ID=1) 讀取 10 個 AI 值，每 50 ms 發送一次命令，如有異常發生 (on error) 則 5 秒後再發送一次。500 ms 未回應會視為異常。



5.2.4 停止/啟用 Modbus TCP/UDP Master Port

每次 PAC 開機後，會自動啟用在 Win-GRAF "Fieldbus Configuration" 的 "IO Drivers" 畫面內設定的 Modbus **TCP/UDP** Master Port。

若您想在程式運行中停用某個 Modbus **TCP/UDP** Master Port 可以使用 "**MBTCP_M_disable**" 或 "**MBUDP_M_disable**" 函式，如下：

```
(* To_disable 宣告為 BOOL *)  
If To_disable then  
  To_disable := FALSE ;  
  MBTCP_M_disable ( '10.0.255.2' , 502 ) ;  
End_if;
```

當 To_disable 設為 TRUE 時，會停用連往 TCP Slave 設備 (IP 位址 = 10.0.255.2，Port_No = 502) 的 Modbus TCP Master Port。如需要，可再使用 "**MBTCP_M_enable**" (或 "**MBUDP_M_enable**") 函式，來啟用它。如下：

```
(* To_enable 宣告為 BOOL  
   Status_tcp 宣告為 BOOL *)  
If To_enable then  
  To_enable := FALSE ;  
  MBTCP_M_enable ( '10.0.255.2' , 502 ) ;  
End_if;  
Status_tcp := MBTCP_M_status ( '10.0.255.2' , 502 ) ;
```

"**MBTCP_M_status**" 或 "**MBUDP_M_status**" 函式，可用來取得 Modbus TCP/UDP Master 的狀態。(True: 啟用；False: 停用)。

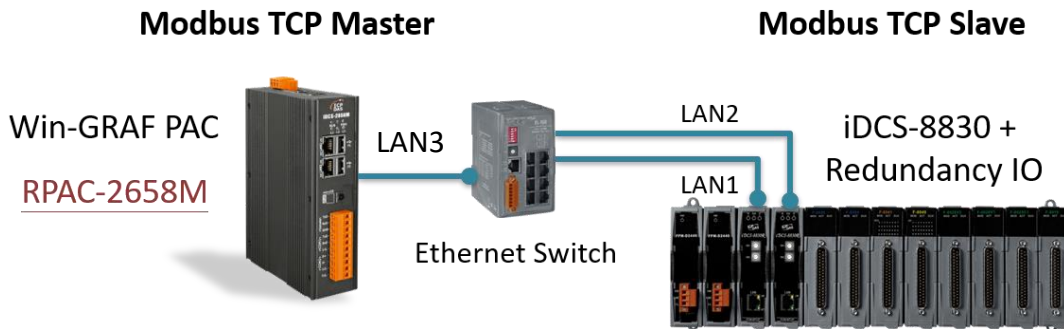
為了程式維護方便與避免錯誤，可以宣告一個 STRING 變數 (例如: IP_addr2，長度設為 20)，並設定初始值 (例如: 10.0.255.2)，如下：

```
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

此章節將說明如何設定備援的“Modbus Master Request”，意即當 Modbus TCP Slave 的其中一個 IP 無法使用時，仍可連到另一個 IP 來讀/寫資料。請參考 5.2 節啟用 Modbus Master 並建立 Data Block。

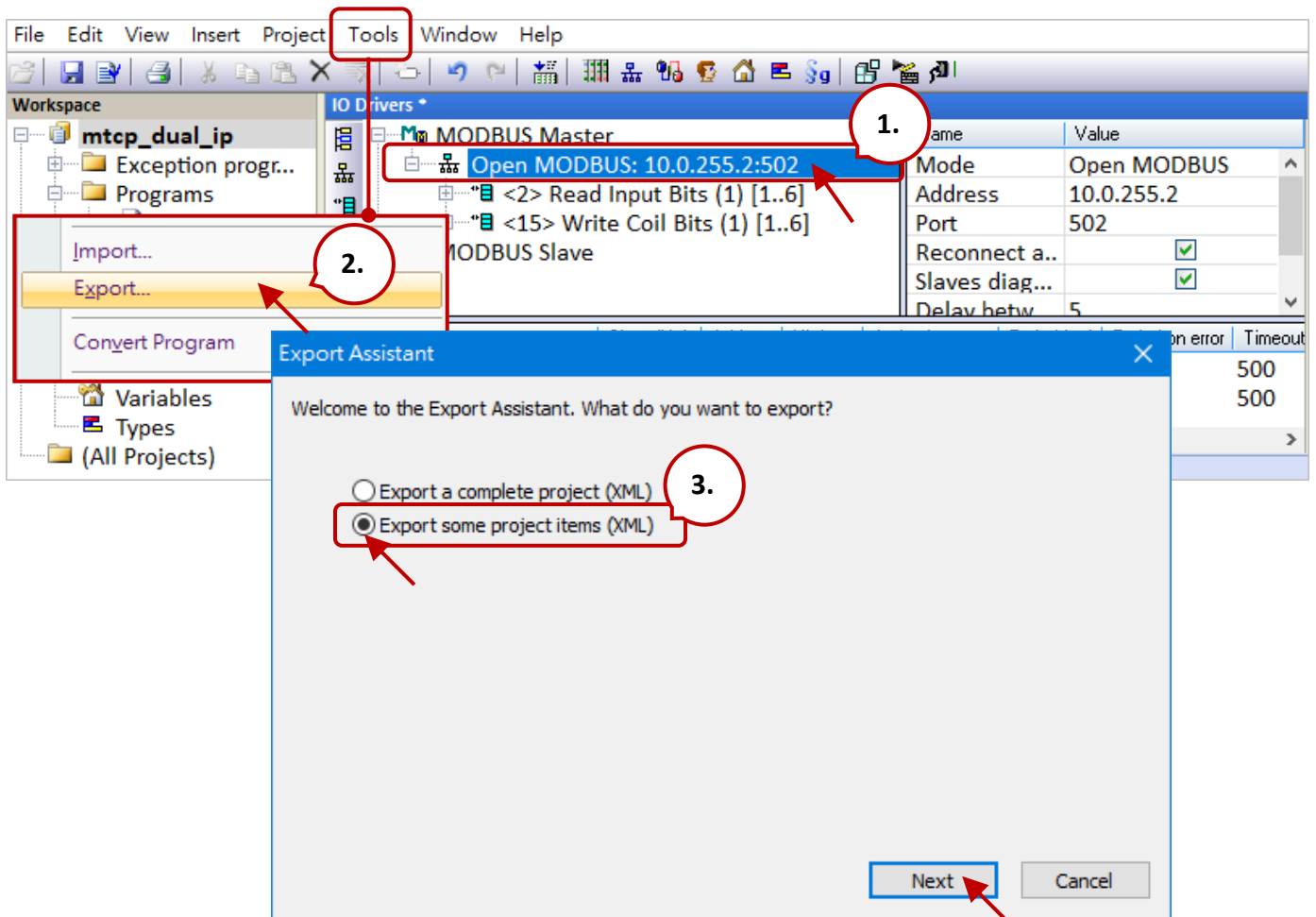
應用示意圖：



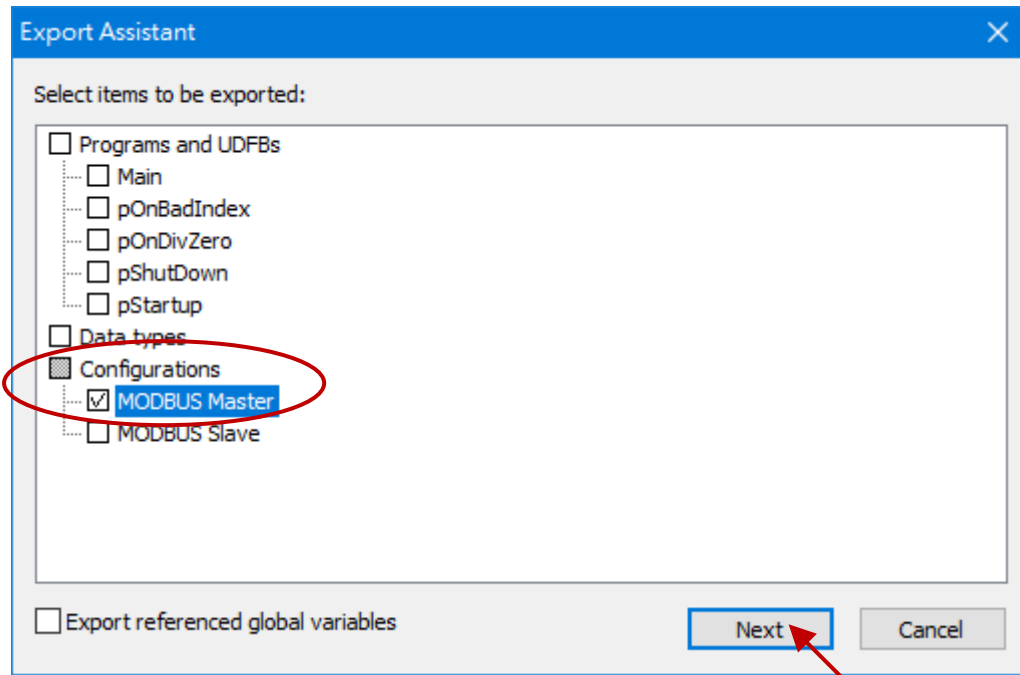
請依照以下步驟：

在 Modbus Master 設定，只能指定去連一個 Modbus Slave 的 IP 位址。以下將介紹如何複製讀/寫設定，並使用第 2 個 Slave IP 位址存取資料。

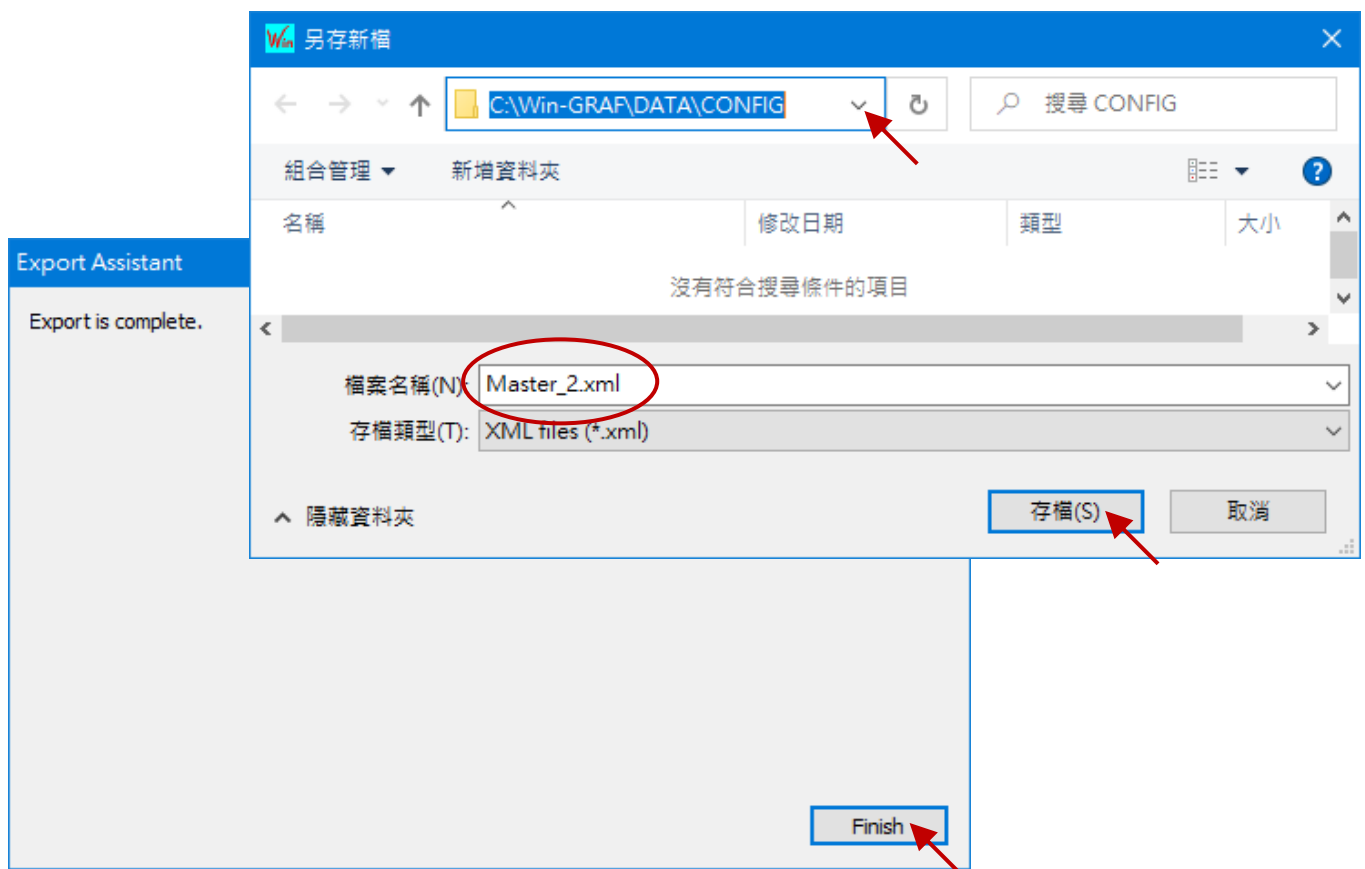
1. 在“IO Drivers”視窗點選“Open MODBUS:”，再點選功能表“Tools - Export”。
2. 在“Export Assistant”視窗點選“Export some project items (XML)”，再點選“Next”。



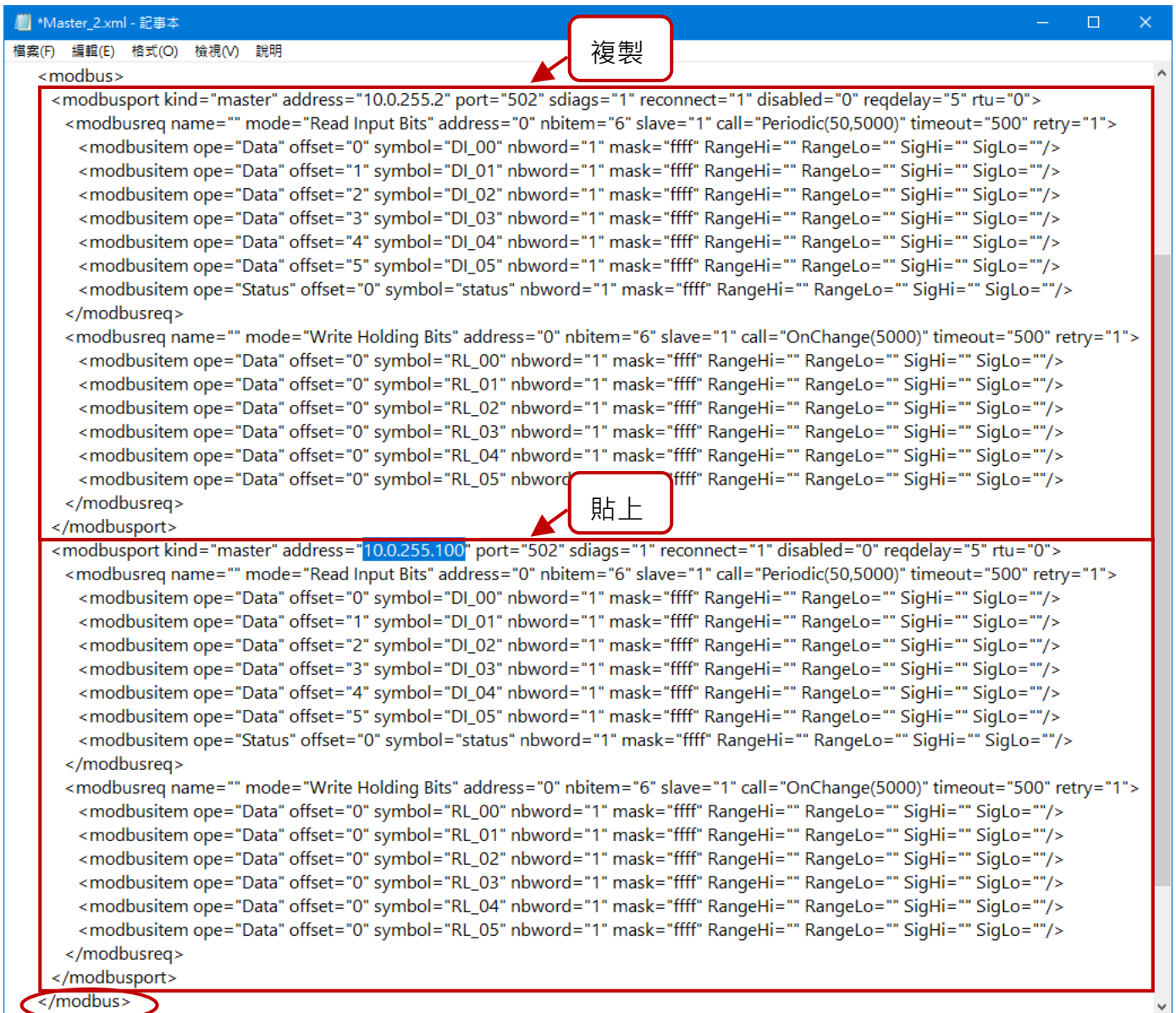
3. 只需勾選“MODBUS Master”配置並點選“Next”按鈕。



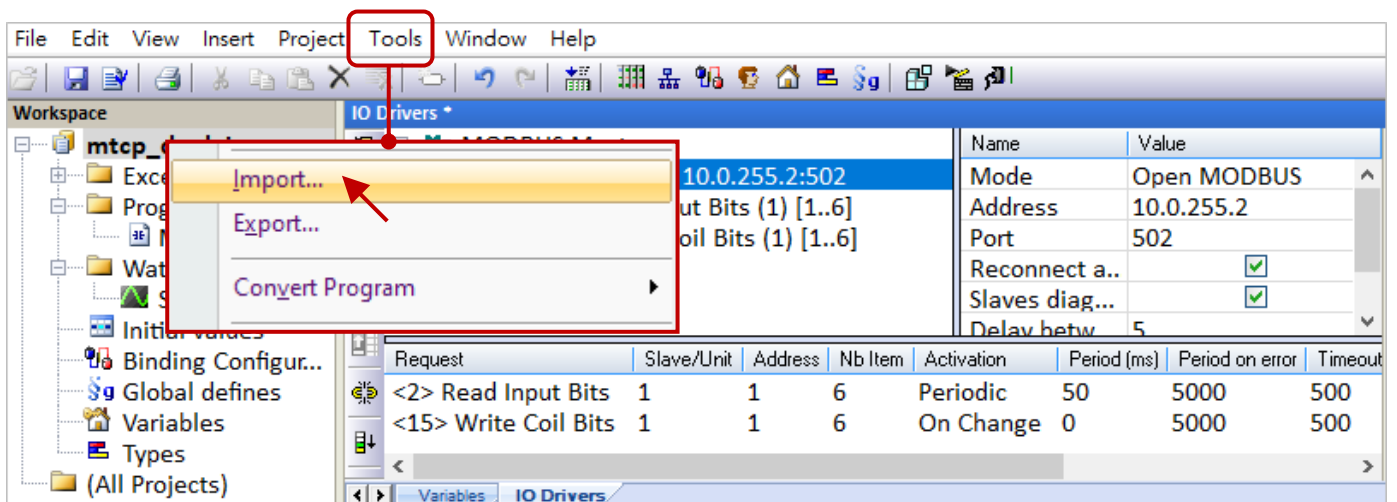
4. 指定檔案位置 (預設在 C:\Win-GRAF\DATA\CONFIG) 並輸入檔名 (例如: Master_2.xml) , 再點選“存檔”按鈕。最後, 點選“Finish”匯出檔案。



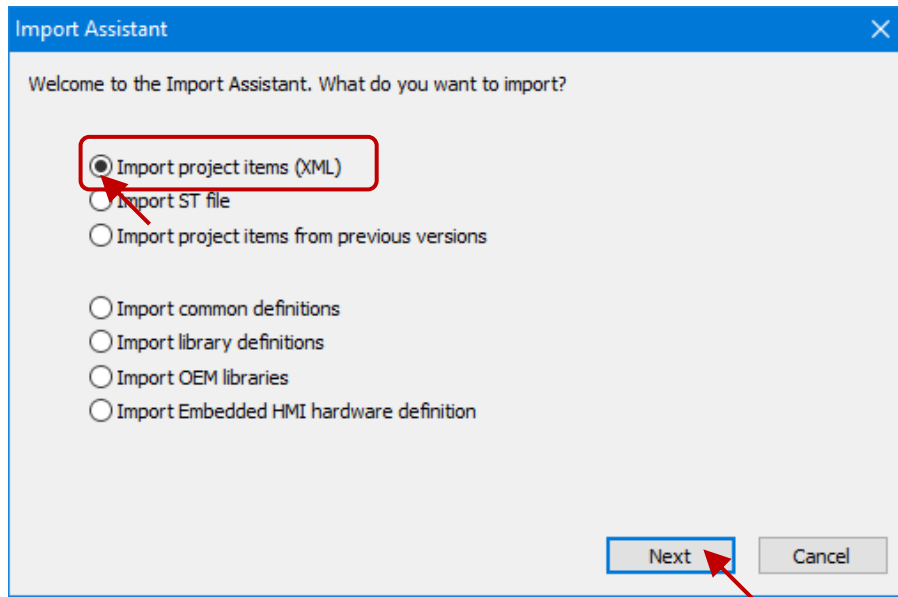
- 以記事本開啟 .xml 檔，並複製 <modbusport> 至 </modbusport> 的程式碼，將其貼到 </modbus> 之上，並修改 address 為 Modbus Slave 的第二個 IP 位址 (例如: "10.0.255.100")，再儲存並關閉檔案。



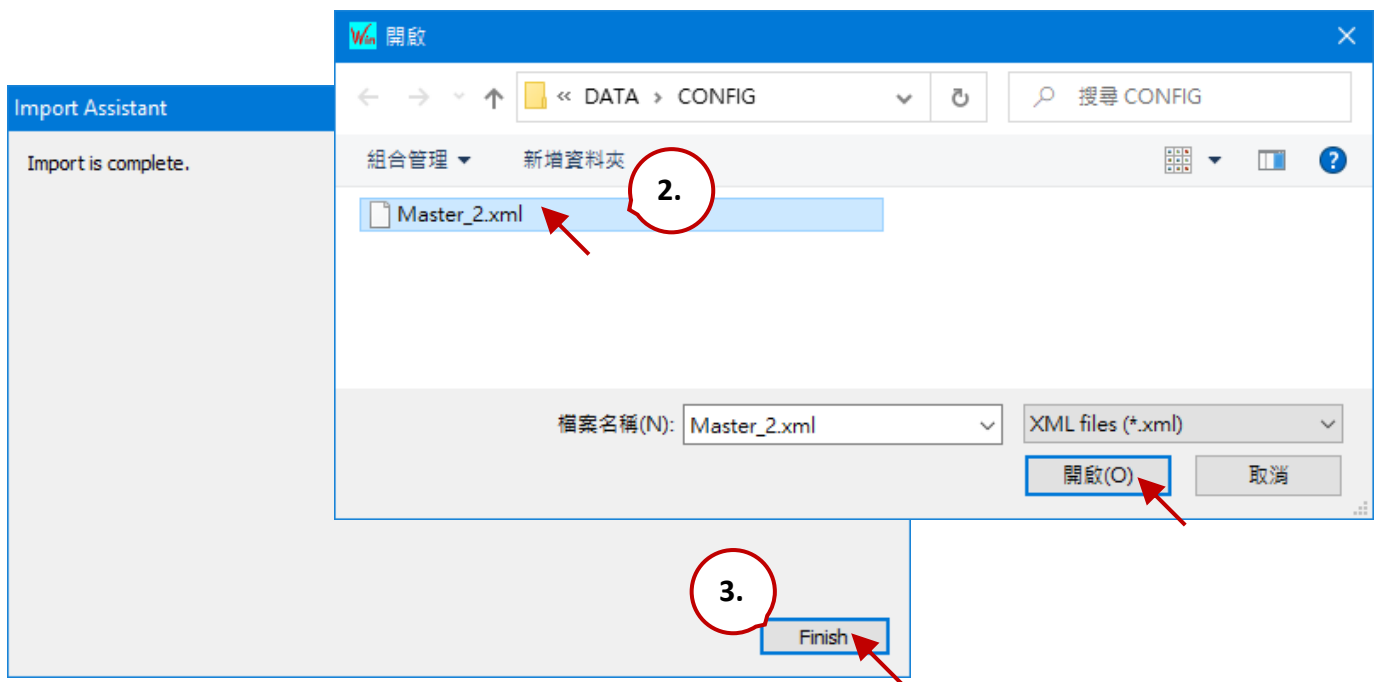
- 在 Win-GRAF Workbench 點選功能表 “Tools - Import”。



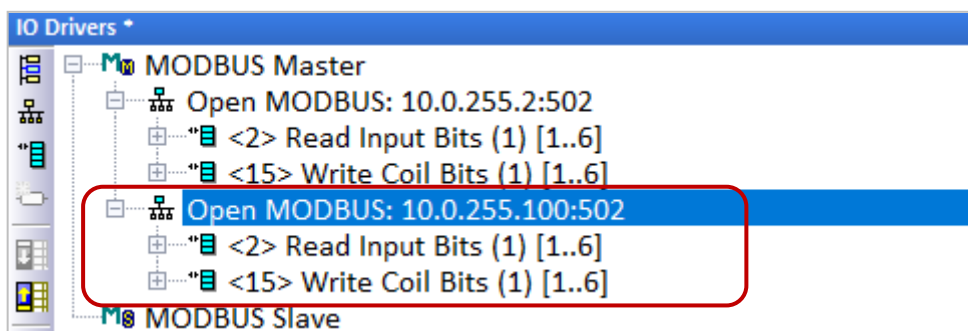
7. 於 “Import Assistant” 視窗中，點選 “Import project items (XML)” 再點選 “Next” 按鈕。



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



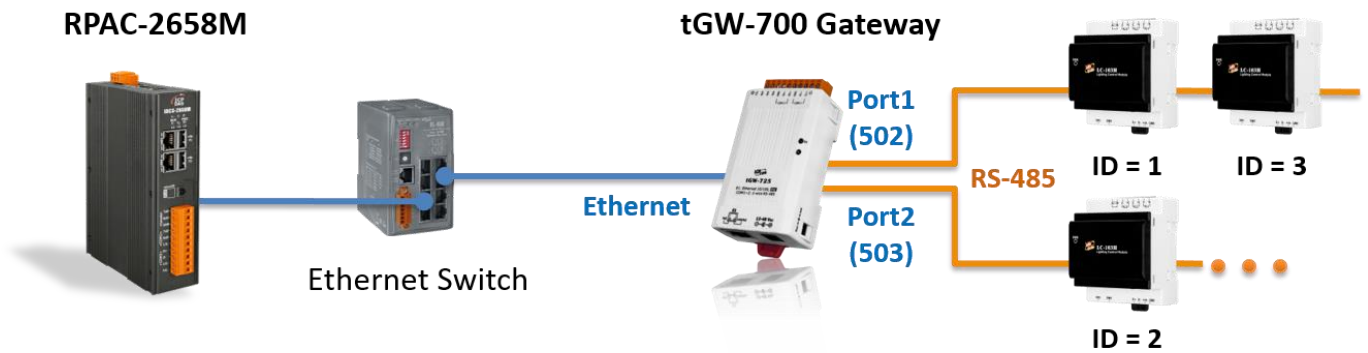
9. 目前已可使用 Modbus Slave 的第二個 IP 位址來存取資料。



5.4 透過 tGW-700 閘道器來存取多個 Modbus RTU Slave

使用 Modbus RTU (RS-232/485/422) 設備進行遠距離傳輸時，選用較低的通訊速率 (Baud rate)，穩定傳輸的距離較長，但傳輸速度慢。此時，您可使用 tGW-700 系列 (Modbus TCP 轉 RTU/ASCII 閘道器) 來透過 Ethernet 傳送資料，進而改善通訊效率。

本章節將說明 Win-GRAF PAC 如何透過 tGW-700 閘道器與多個 LC-103H 模組進行通訊 (如下圖)。



您可在網站下載 [範例程式 \(demo_tgw725.zip\)](#)。

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

tGW-700 系列是一款 Modbus TCP 轉 Modbus RTU/ASCII 閘道器，它可以讓 Modbus TCP 主機 (例如: RPAC-2658M) 透過 Ethernet 與串列 Modbus RTU/ASCII 設備進行通訊，並消除傳統串列通訊設備對佈線長度的限制。

請參訪 [tGW-700 系列網頁](#)，了解更多產品資訊，並參考 [tGW-700 系列 使用手冊](#) - 第 3，4 章來了解 tGW-700 的網路設定、測試方式 與 網頁配置的說明。

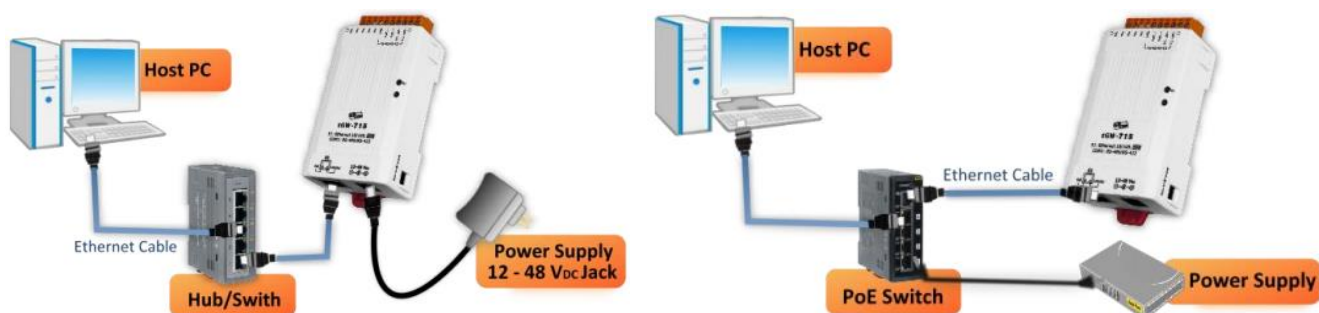
使用前，請先進行 tGW-700 的網路 與 COM Port 設定：

- 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 皆在同一個子網域上，然後供電開機啟動 tGW-700 系列模組。

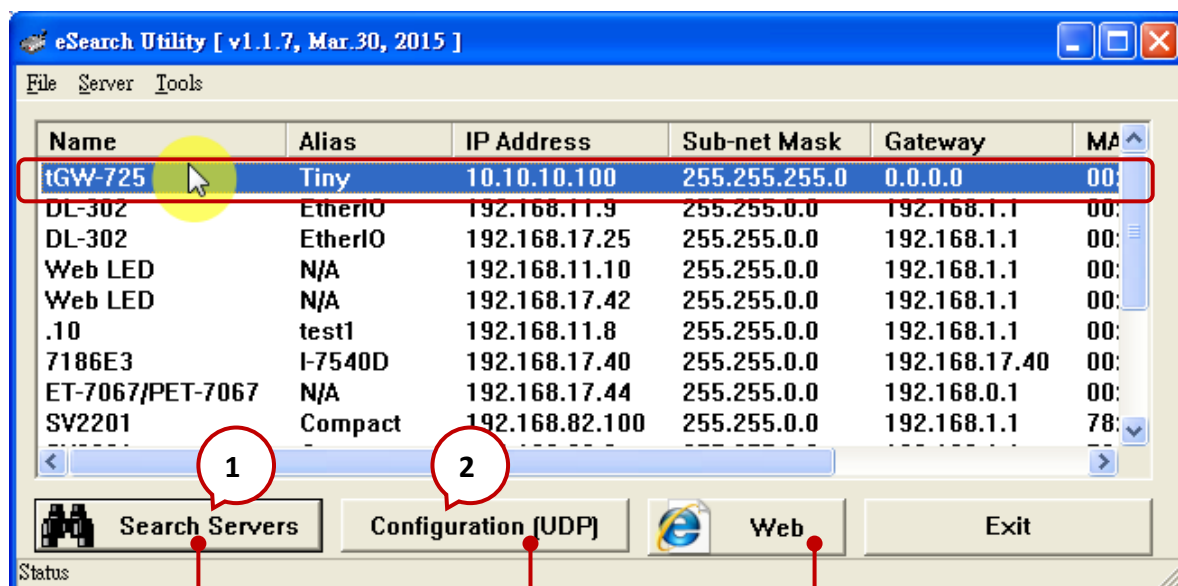


+12 ~ +48 VDC 插孔輸入開機 (Non-PoE)

PoE 供電開機

- 在 PC 安裝 eSearch Utility，搜尋並設定 tGW-700 的網路配置。

https://www.icpdas.com/en/product/guide+Software+Utility_Driver+eSearch__Utility

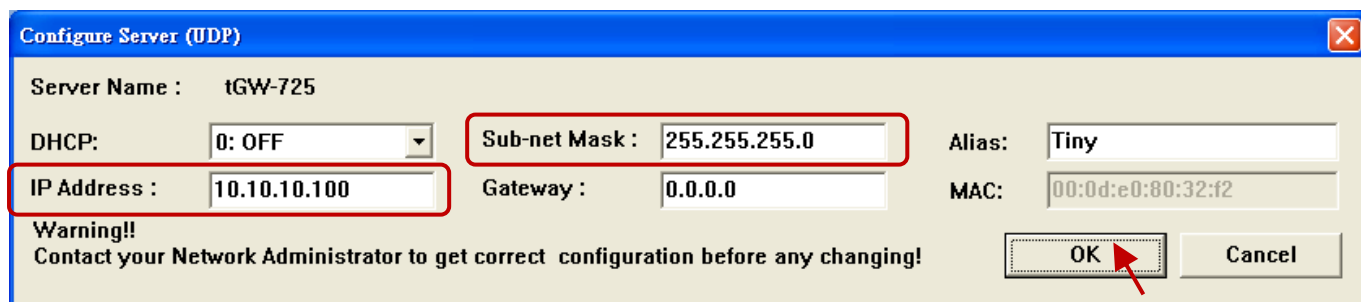


搜尋您的 tGW-700。

設定 IP / Mask / Gateway。

配置 tGW-700 Gateway。

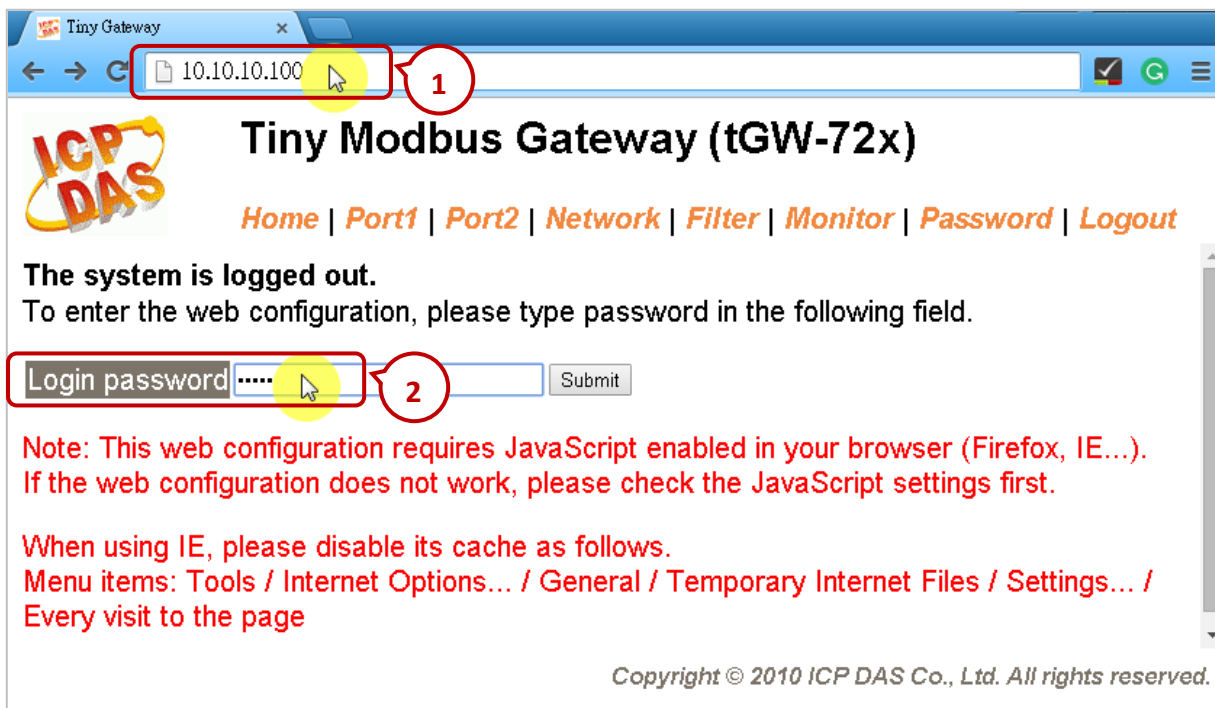
您可變更 IP Address/ Subnet Mask / Gateway 設定。修改完成後，單擊“OK”按鈕，tGW-700 將會在 2 秒後套用新的設定。



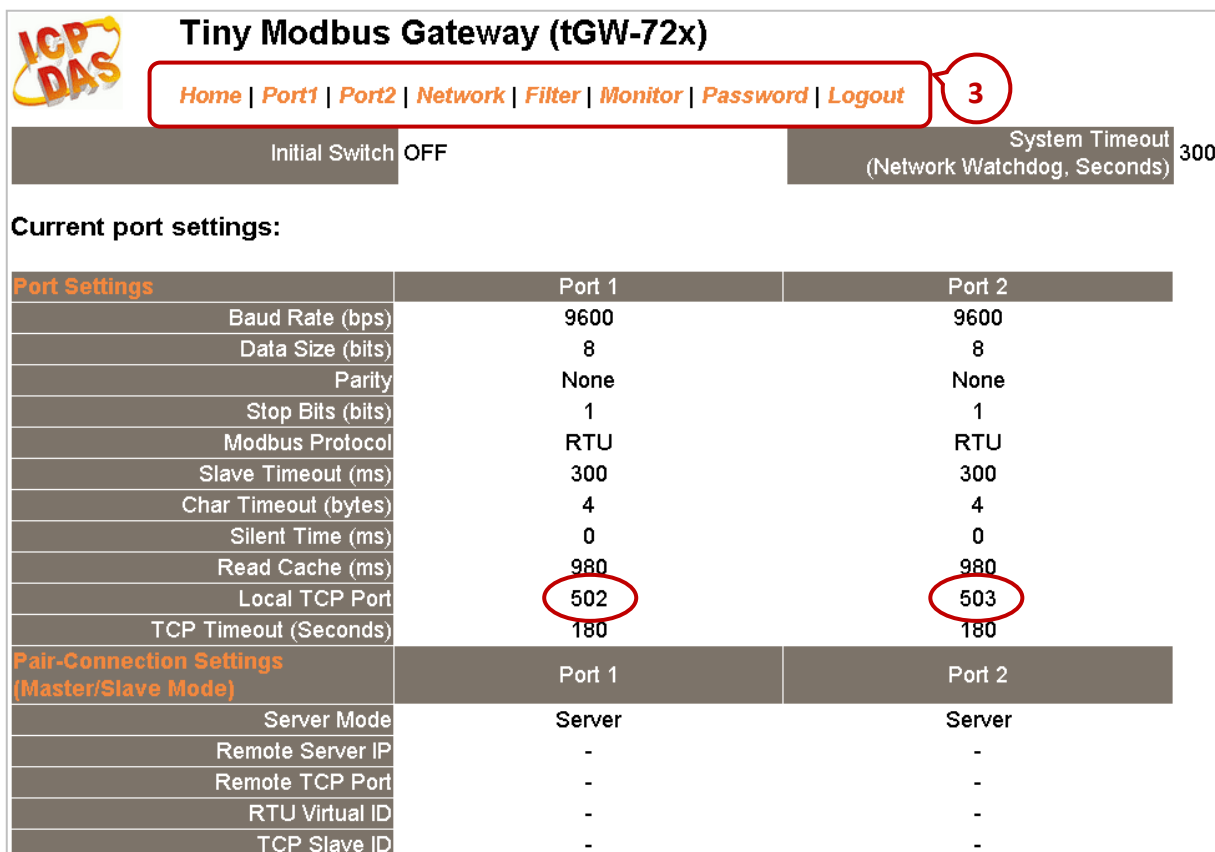
● 網頁配置

以下將說明 COM Port 設定。

1. 在網址列輸入 tGW-700 的 IP 位址 (註: tGW-700 和 PC 需在同一個網域內)。
2. 輸入登入密碼 (原廠預設密碼: admin) 並點選 Submit 登入。



3. 登入後，首頁 (Home) 會顯示目前的 Port 設定，您也可在 “Port1” 或 “Port2” 頁面修改設定。



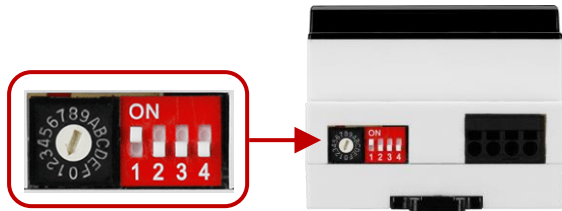
5.4.2 透過 tGW-700 存取 LC 系列模組

以下將說明 Win-GRAF PAC 如何透過 tGW-725 來存取 LC-103H 模組的資料。

tGW-725 是一款小型的 Modbus TCP 轉 Modbus RTU/ASCII 閘道器，具有 PoE 供電 與 2 個 RS-285 Port。

LC-103H 是一款支援 Modbus RTU 通訊協定的燈控模組，並提供 1 個 DI 通道 與 3 個 Relay Output。

使用前，請設定好模組的 ID 編號，例如: ID = 1，將模組底部的旋轉開關調整至 “1” 即可。



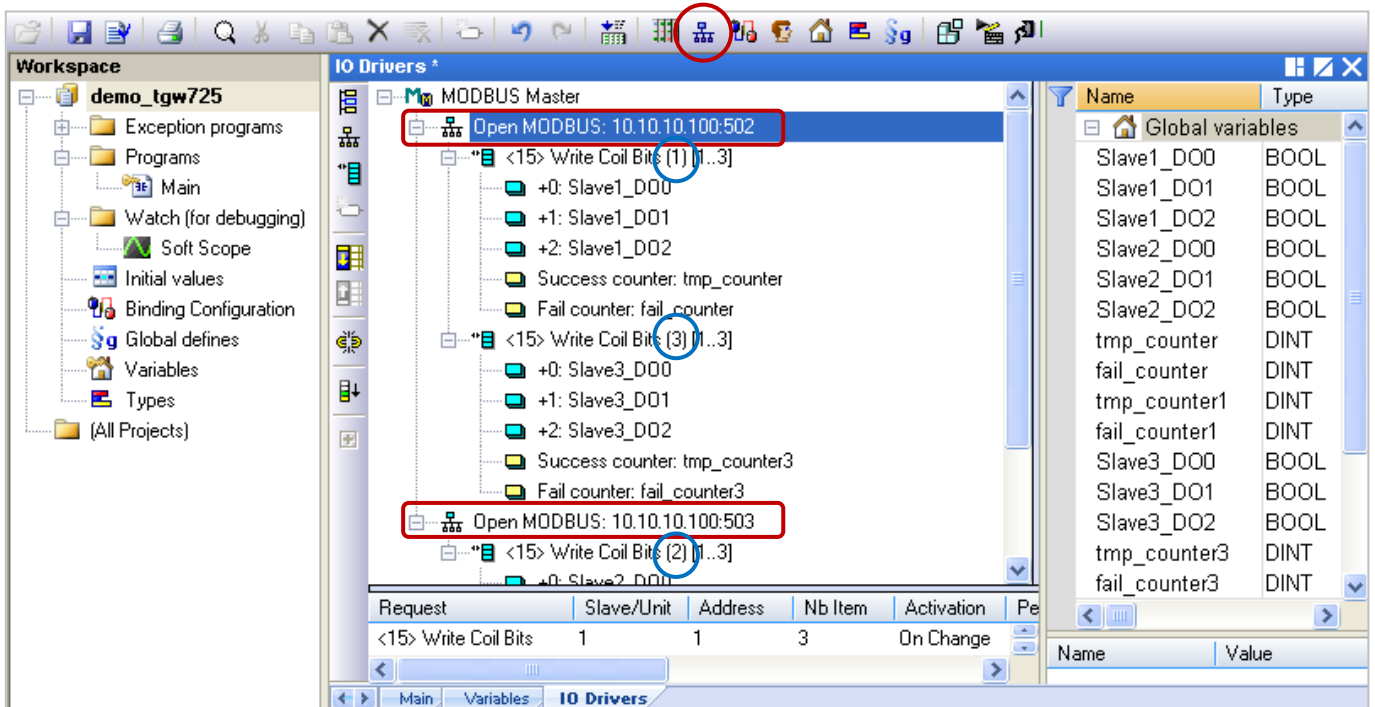
LC 系列模組使用手冊:

<http://www.icpdas.com/en/download/show.php?num=596&model=LC-103H>

請在網站下載 [範例程式 \(demo_tgw725.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。

範例說明: (可參考 5.2 節，了解 Modbus TCP Master 設定)

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

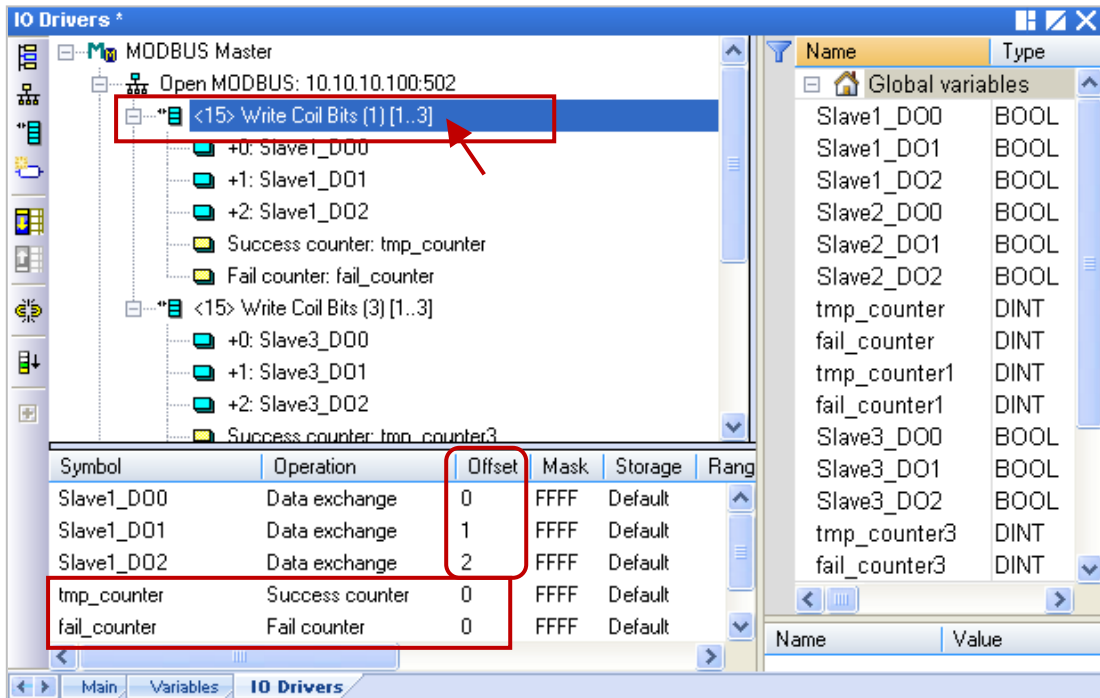


下表說明各設備的相關資訊:

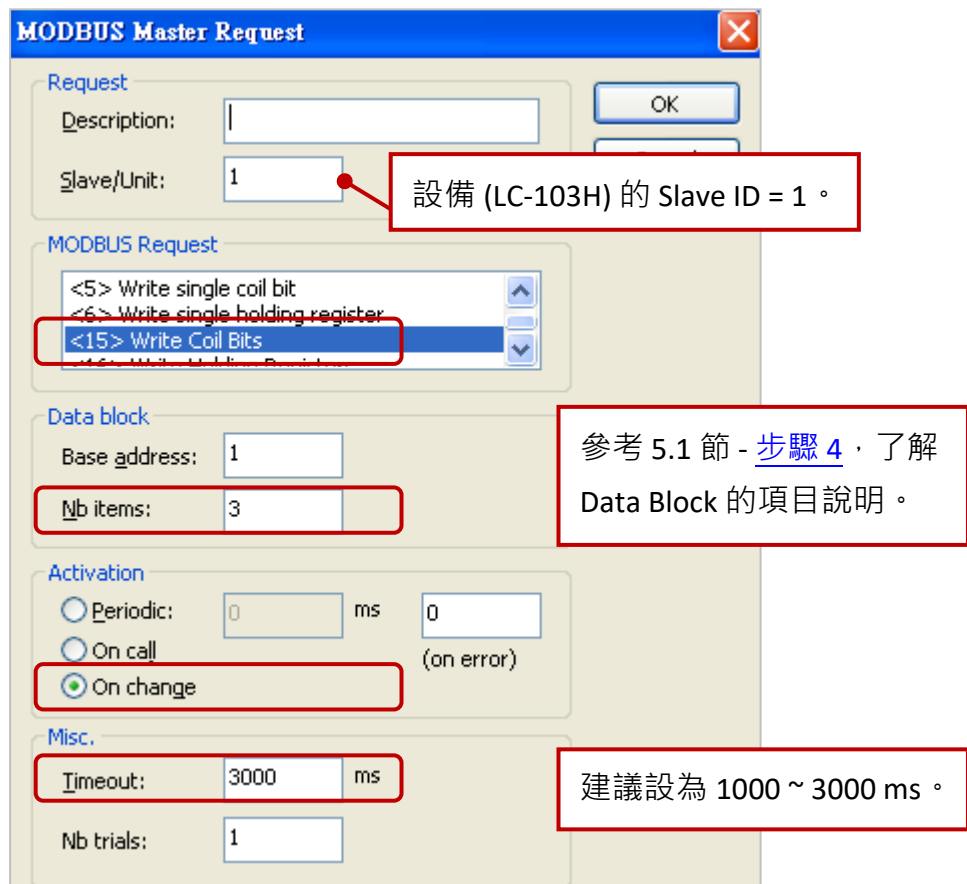
Win-GRAF PAC	tGW-725			LC-103H	
Modbus TCP Master	Modbus TCP to RTU/ASCII Gateway			Modbus RTU Slave	
	IP Address 10.10.10.100	TCP Port = 502	COM1	2 個模組	Slave ID = 1 & 3
		TCP Port = 503	COM2	1 個模組	Slave ID = 2

2. 點選 (或滑鼠雙擊) PORT 502 下的第 1 個 Data Block 來查看設定。

"Operation" 設定為 "Success counter" (或 "Fail counter") 表示寫入成功 (或失敗)，該變數值會加 1。另外，此兩項目的 "Offset" 值必需設定為 "0"。

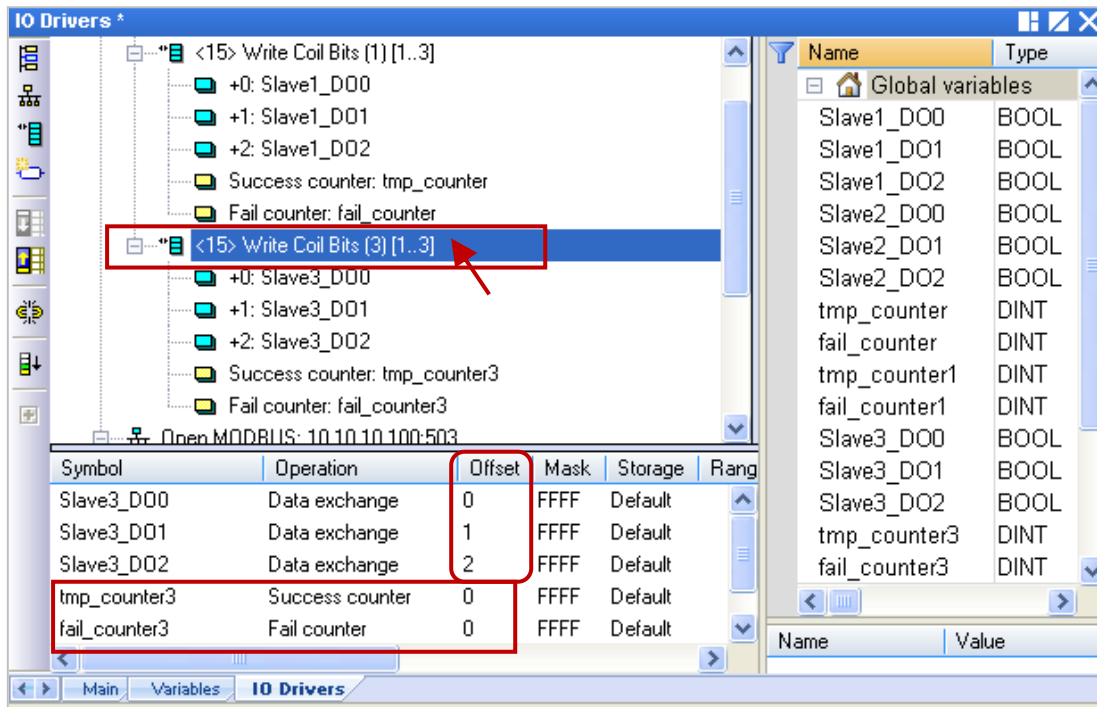


說明: 於設備 (Slave ID = 1) 寫入 3 個 DO 值，資料有改變時，才發送一次命令。
3 秒 未回應會視為異常。

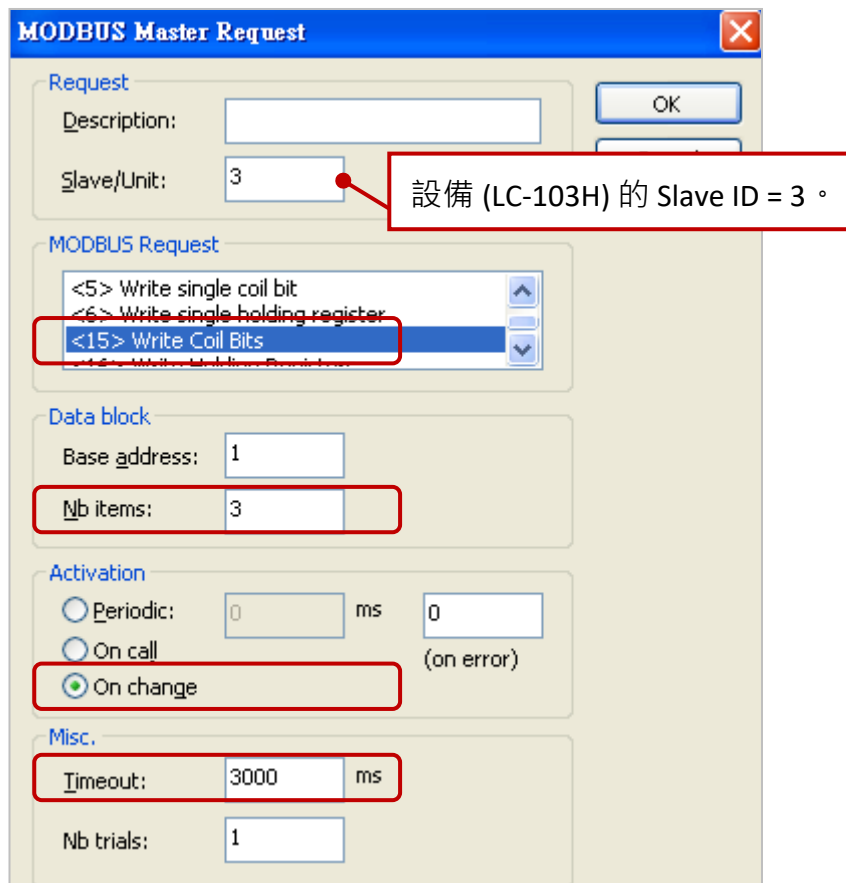


3. 點選 (或滑鼠雙擊) PORT 502 下的第 2 個 Data Block 來查看設定。

"Operation" 設定為 "Success counter" (或 "Fail counter") 表示寫入成功 (或失敗)，該變數值會加 1。另外，此兩項目的 "Offset" 值必需設定為 "0"。

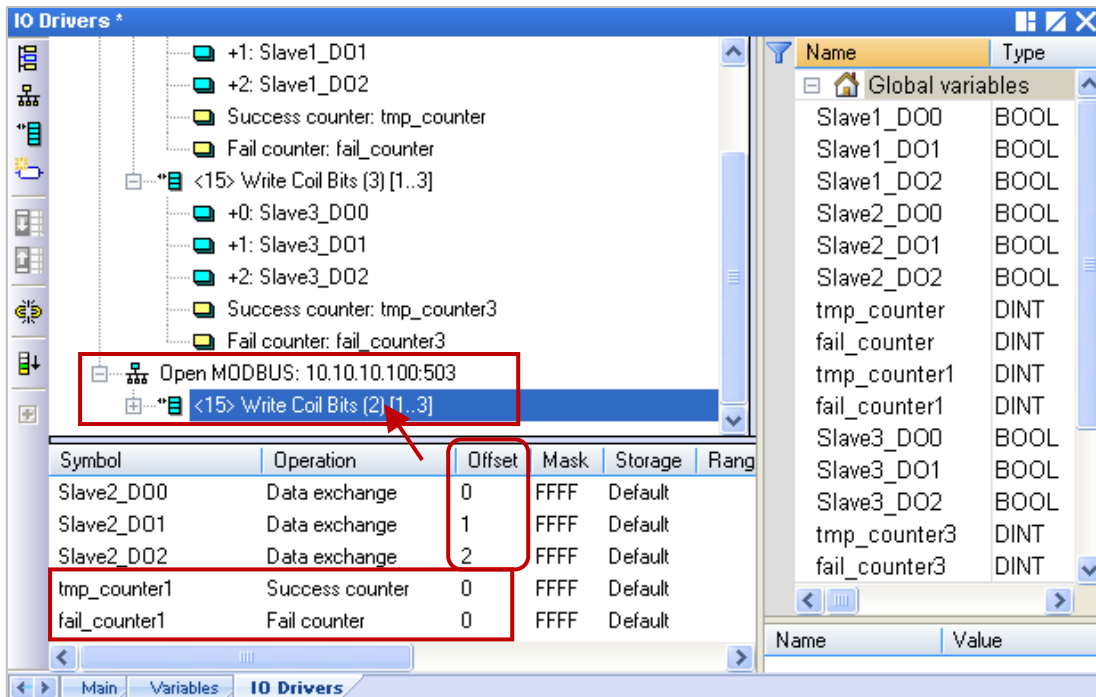


說明: 於設備 (Slave ID = 3) 寫入 3 個 DO 值，資料有改變時，才發送一次命令。
3 秒 未回應會視為異常。

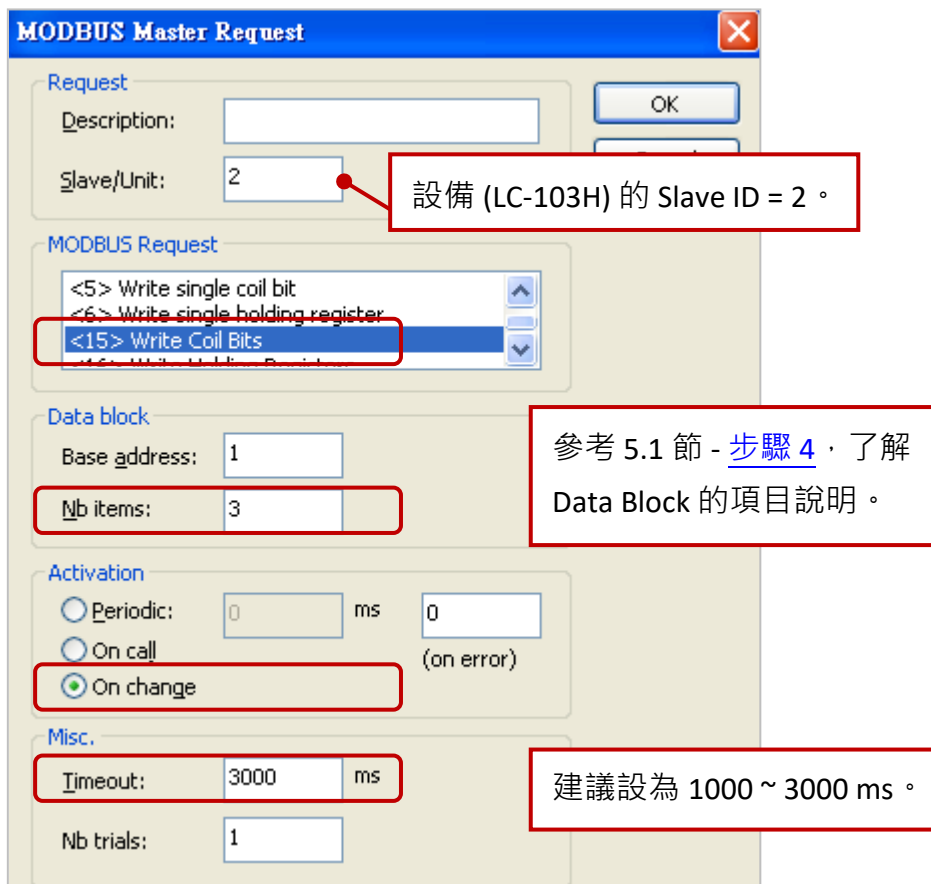


4. 點選 (或滑鼠雙擊) PORT 503 下的第 1 個 Data Block 來查看設定。

"Operation" 設定為 "Success counter" (或 "Fail counter") 表示寫入成功 (或失敗)，該變數值會加 1。另外，此兩項目的 "Offset" 值必需設定為 "0"。



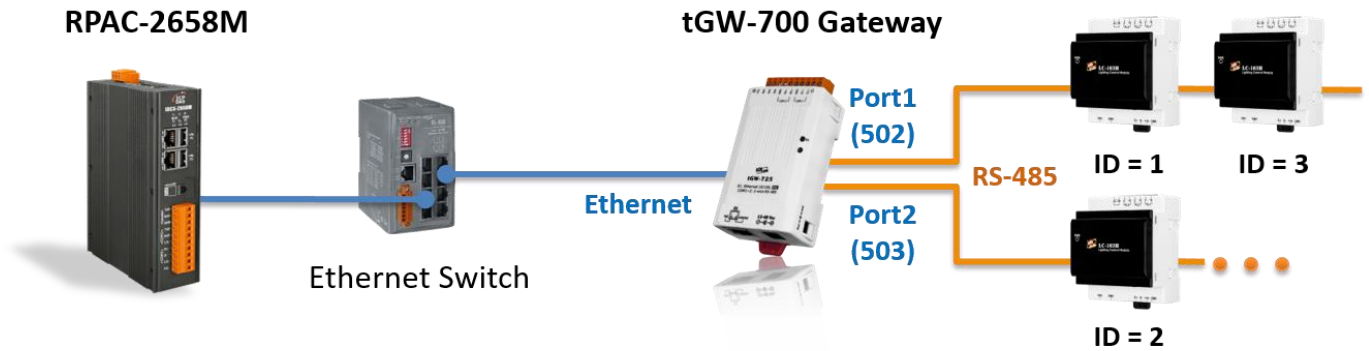
說明: 於設備 (Slave ID = 2) 寫入 3 個 DO 值，資料有改變時，才發送一次命令。
3 秒未回應會視為異常。



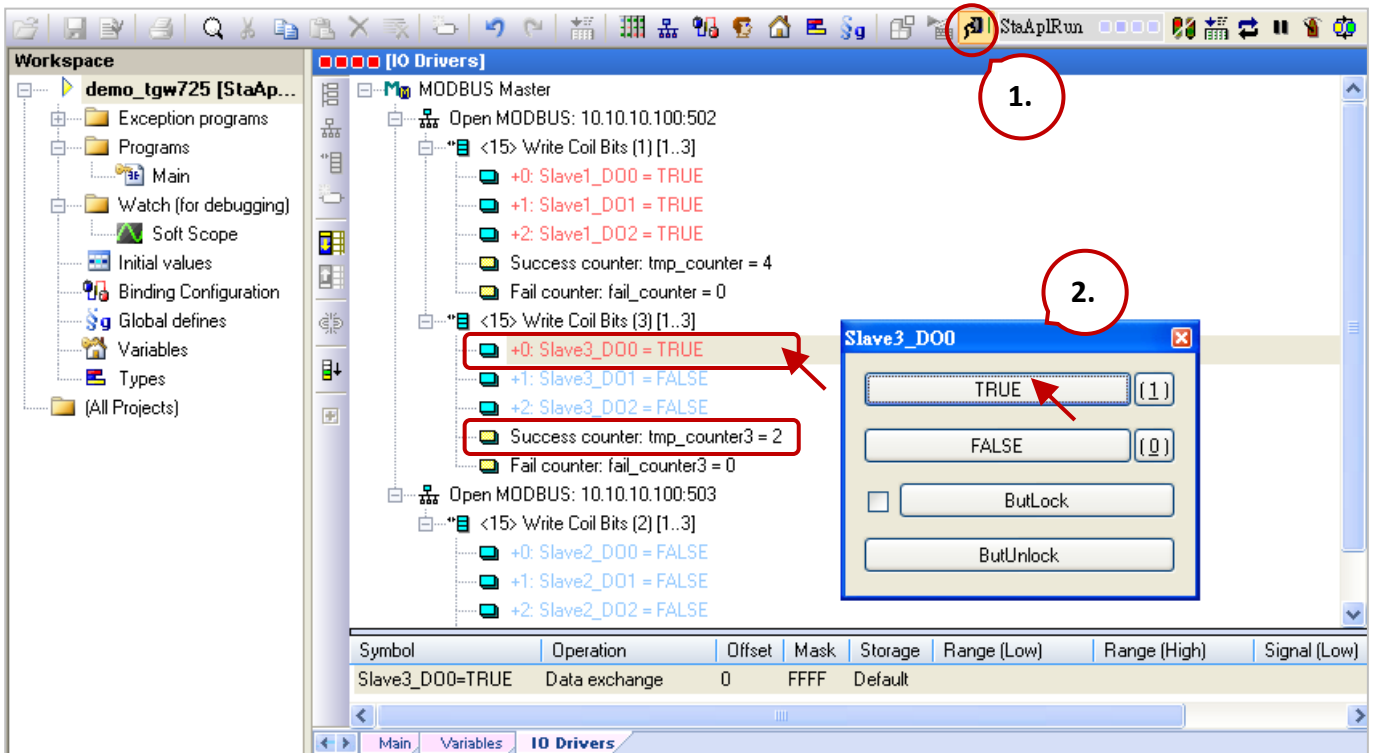
5.4.3 測試範例程式 (demo_tgw725.zip)

測試前，請先將此範例程式下載到您的 Win-GRAF PAC。

硬體連接:



點選 "On Line" 按鈕與 PAC 連線，您可在 "IO Drivers" 視窗中，滑鼠雙擊任一 DO 變數，並將其狀態設定為 "TRUE"。若寫入成功，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"，表示寫入資料成功。

第 6 章 可保存變數與資料儲存

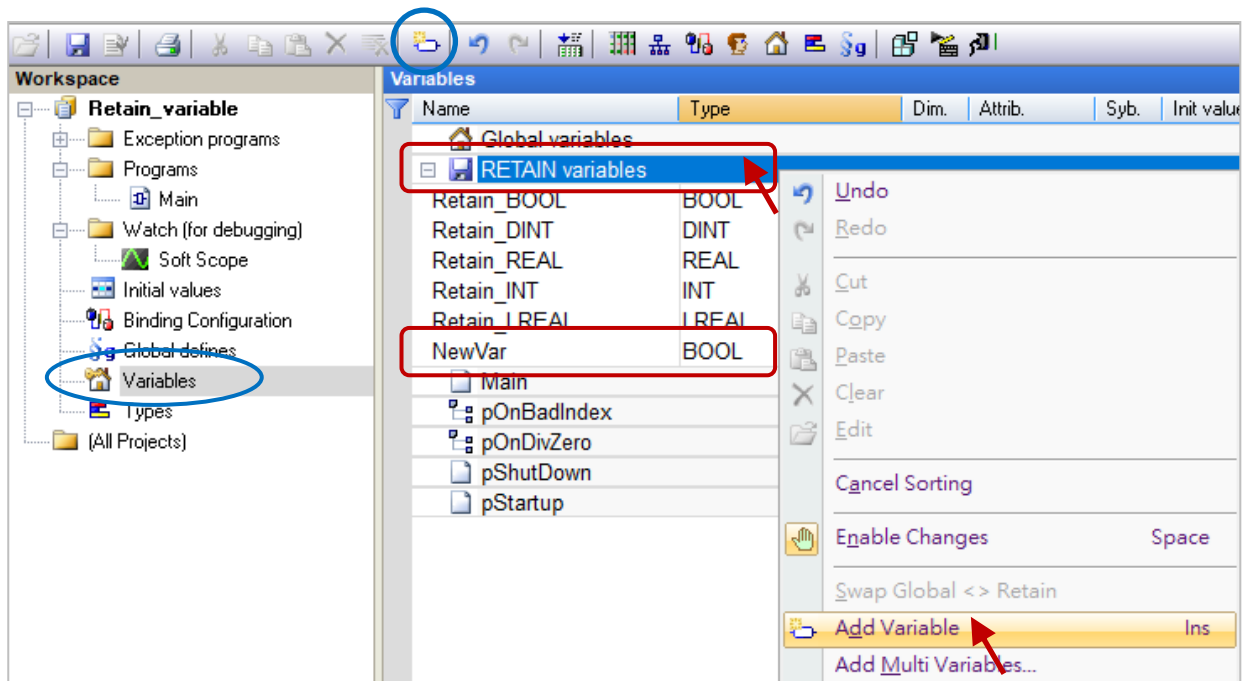
可保存變數 (Retain variable) 用來保護重要資料，可於系統斷電並重啟後，自動復原至斷電前的狀態。您可選擇在 Variables 視窗內，使用 RETAIN variable，或使用 RETAIN_xxx Function。

注意:

- 為了避免內部資料衝突，一旦專案使用了 RETAIN variables，就不能再使用 Function 方式，否則該 Function 將會失效。
- 由於可保存變數會存取硬碟資料，因此設定過多的可保存變數會造成系統效能降低。

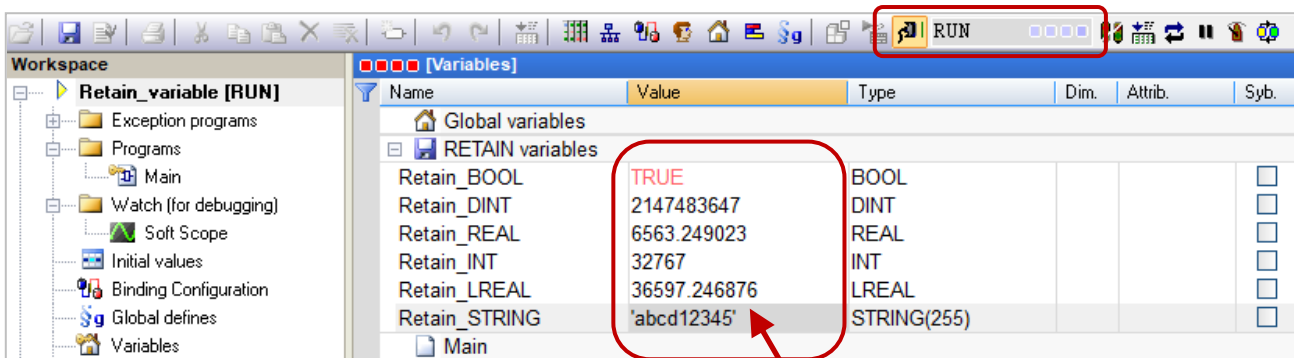
建立保存變數

在 "Variables" 視窗，滑鼠右鍵點選 "RETAIN variables"，再點選 "Add Variable" 新增保存變數。接著，輸入變數名稱與 型別 (Type)，再按 Enter。



測試保存變數

將專案下載到 Win-GRAF PAC，並於運行中修改數值。如設定正確，則 PAC 斷電重開後，會顯示先前變更的值。



6.1 使用 RETAIN_xxx 函式 (Function)

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 · 下次開機時資料仍然存在。

注意:

1. 為了避免內部資料衝突 · 一旦專案使用了 Function 方式 · 就不能再使用 RETAIN variable · 否則該 Function 將會失效。
2. "Retain_Var()" 或 Retain_Ary() 函式 · 只能在第一個 PAC Cycle 或有執行 On-line Change 的那個 Cycle 內使用 · 若在其它 Cycle 內呼叫此函式 · 將會回傳 "FALSE"。
3. 使用者需為所有的保存變數指定初始值 · 若未指定 · 則 PAC 程式執行 Function 後 · 讀到的會是無效的值。

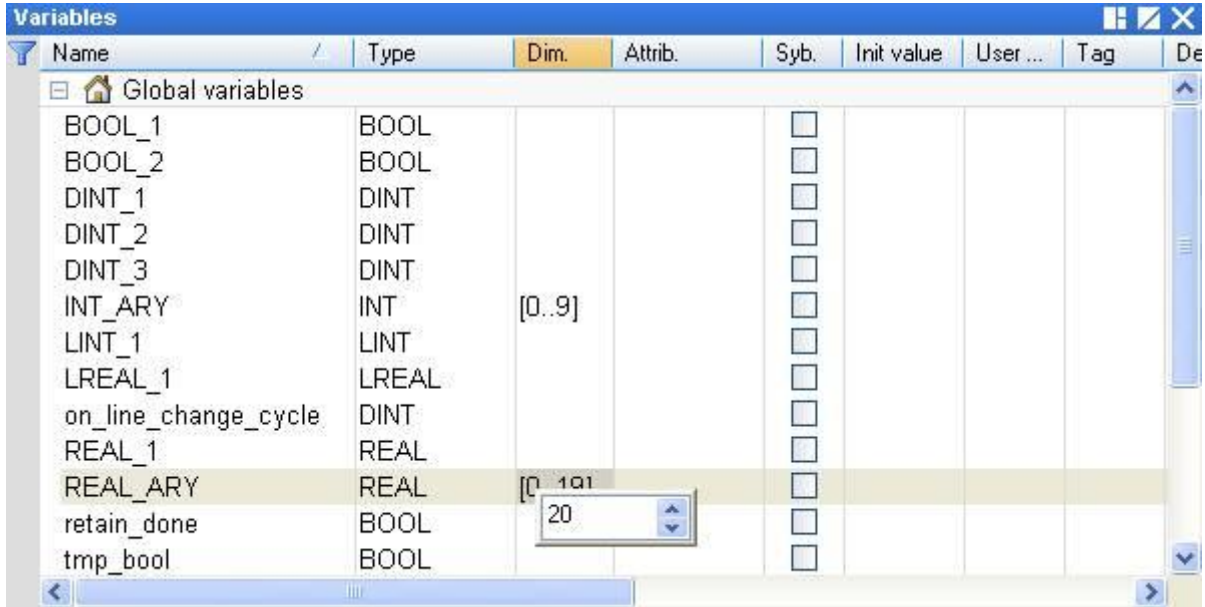
請在網站下載 [範例程式 \(demo_retain.zip\)](#) · 並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。

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 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;
```

您可在 "Variables" 視窗查詢/設定變數。



6.1.1 RETAIN_VAR: 設定保存一個變數



使用小技巧:
按 "F1" 鍵，可查看說明。

Name:

保存變數的名稱 (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")

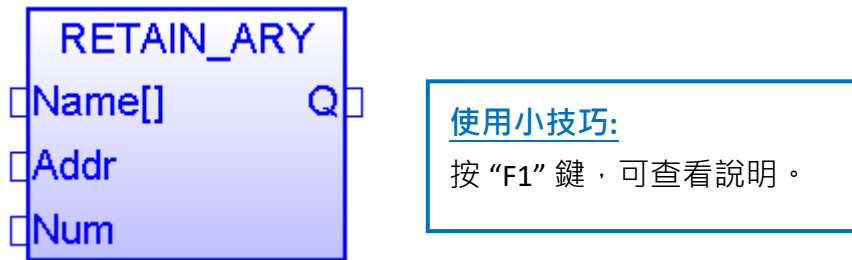
用來保存變數的位址編號 · 可設定為 1 ~ 12000。

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 (或陣列的一個元素) · 請勿將相同的位址指定給兩個 (或多個) 變數 · 否則變數資料將會出錯。
2. 若使用 64-bit 的資料型態 (例如: LINT, LREAL) · 請設定位址為 10001 ~ 12000。

6.1.2 RETAIN_ARY: 設定保存一個陣列變數



Name[]:

陣列變數的名稱 (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”)

陣列變數的起始位址 · 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 · 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 · 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 (或陣列的一個元素) · **請勿**將相同的位址指定給兩個 (或多個) 變數 · 否則變數資料將會出錯。
2. 若使用 64-bit 的資料型態 (例如: LINT, LREAL) · 請設定位址為 10001 ~ 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。為了讓程式可正常運作，使用者必需為所有的保存變數指定一個初值，如未設定，則 PAC 一開機從記憶體所讀到的值一般會是個亂數值。

將 "Retain_Flag_Set()" 設定為 "TRUE"，表示所有的保存變數已設置了初值；

而 "Retain_Flag_Get()" 可取得旗標的狀態；"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 On-line Change 的 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
```

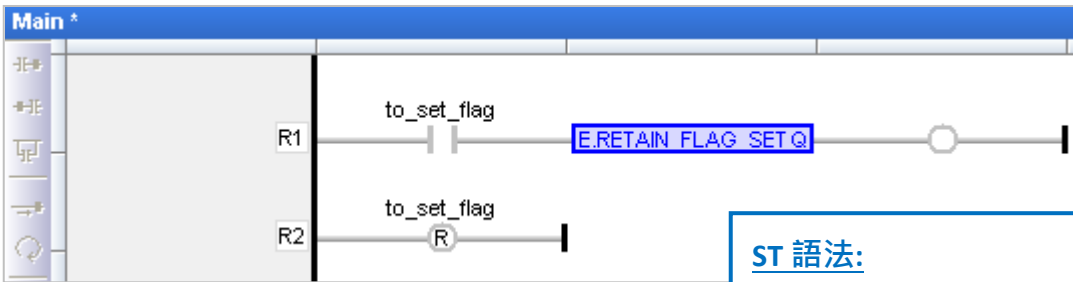
```
  to_set_flag := FALSE;
```

```
  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。

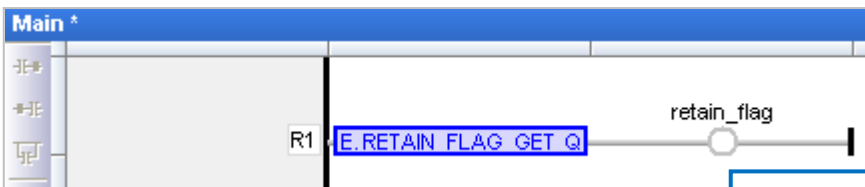


Q: 資料型態為 “BOOL”，只回傳 “TRUE”。

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。

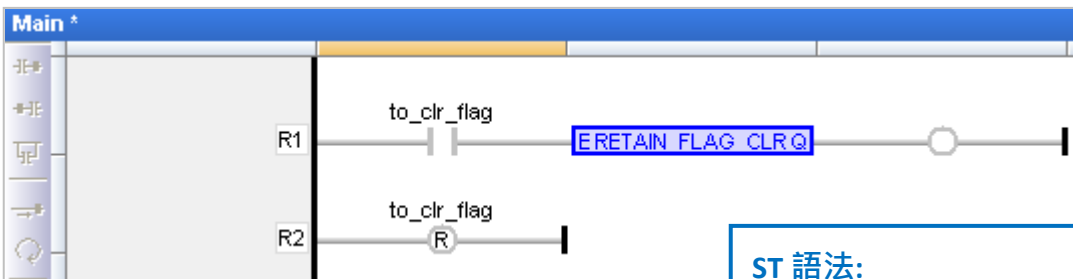


Q: 資料型態為 “BOOL”，
“TRUE”: 表示已設立旗標 (Flag) ;
“FALSE”: 表示未設立旗標 (Flag)。

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”。

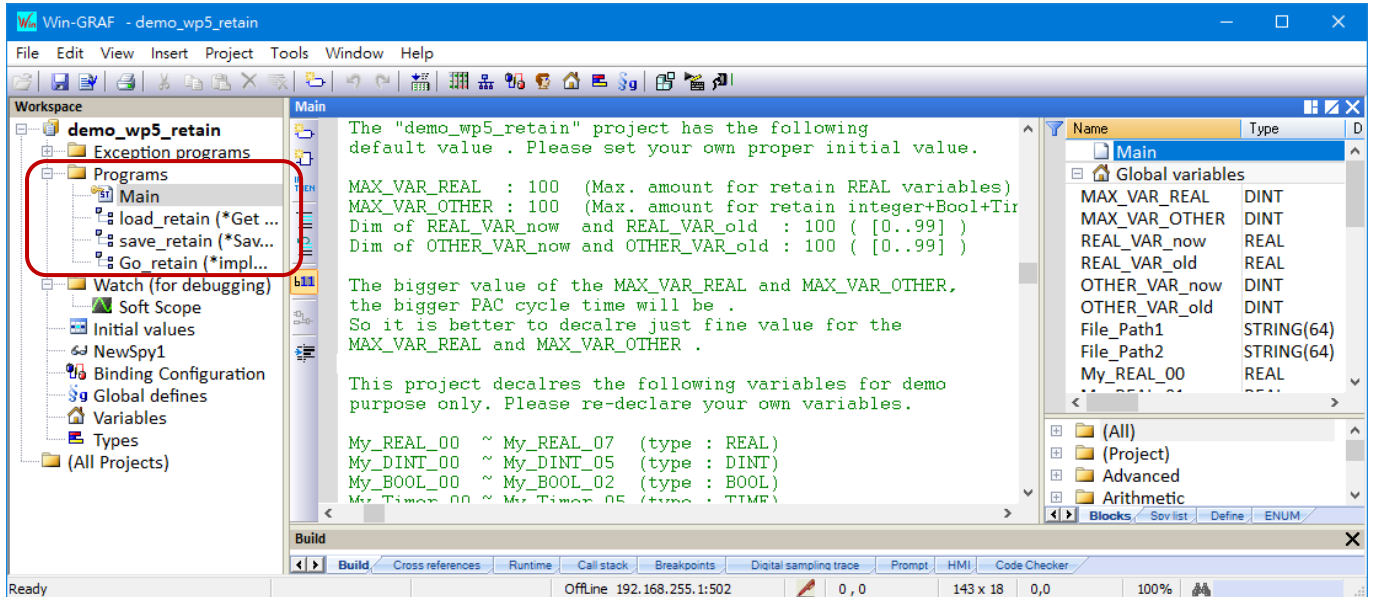
ST 語法:

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

6.2 將保存變數與資料存成檔案

請在網站下載 [範例程式 \(demo_wp5_retain.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。

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會頻繁變動的保存變數資料 (例如: 每秒或每分鐘需變更數值)。此例，保存變數資料是儲存在 /System_disk/ 路徑下的檔案中，若數值常常變動，則更新檔案會消耗很多 CPU 時間進而造成 PAC 效能低下。

下表為 "demo_wp5_retain" 專案中的設定值，請依實際需求設定適當的值。

變數名稱	值	說明
MAX_VAR_REAL	Init. = 100	最多可使用 REAL 保留變數的數量
MAX_VAR_OTHER	Init. = 100	最多可使用 Integer、BOOL、TIMER 保留變數的全部數量
REAL_VAR_now REAL_VAR_old	Dim. = 100	[0..99]，需和 "MAX_VAR_REAL" 值相同
OTHER_VAR_now OTHER_VAR_old	Dim. = 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，請設定為接近實際使用的數量即可。

以下為此專案中宣告的變數:

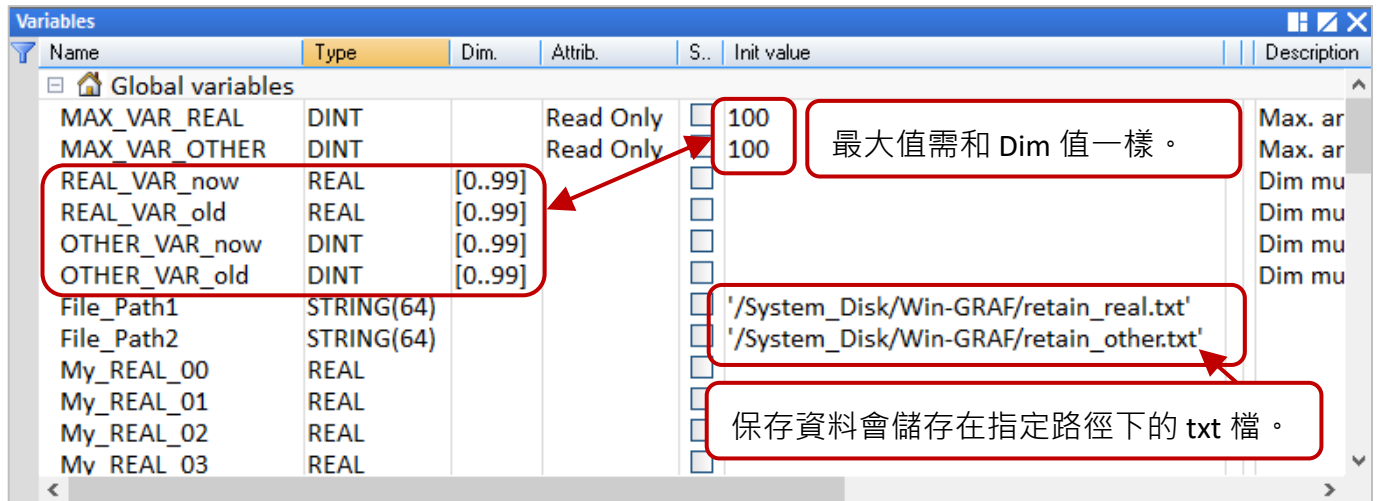
My_REAL_00 ~ My_REAL_07 (資料型態: REAL)

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)

保存資料會儲存在指定路徑下的 txt 檔您可在 "Variables" 視窗查看更多的變數。



"Go_retain" 副程式是用來執行保存動作。您可使用自己宣告的變數，來修改以下 4 個區塊:

(* 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 ;

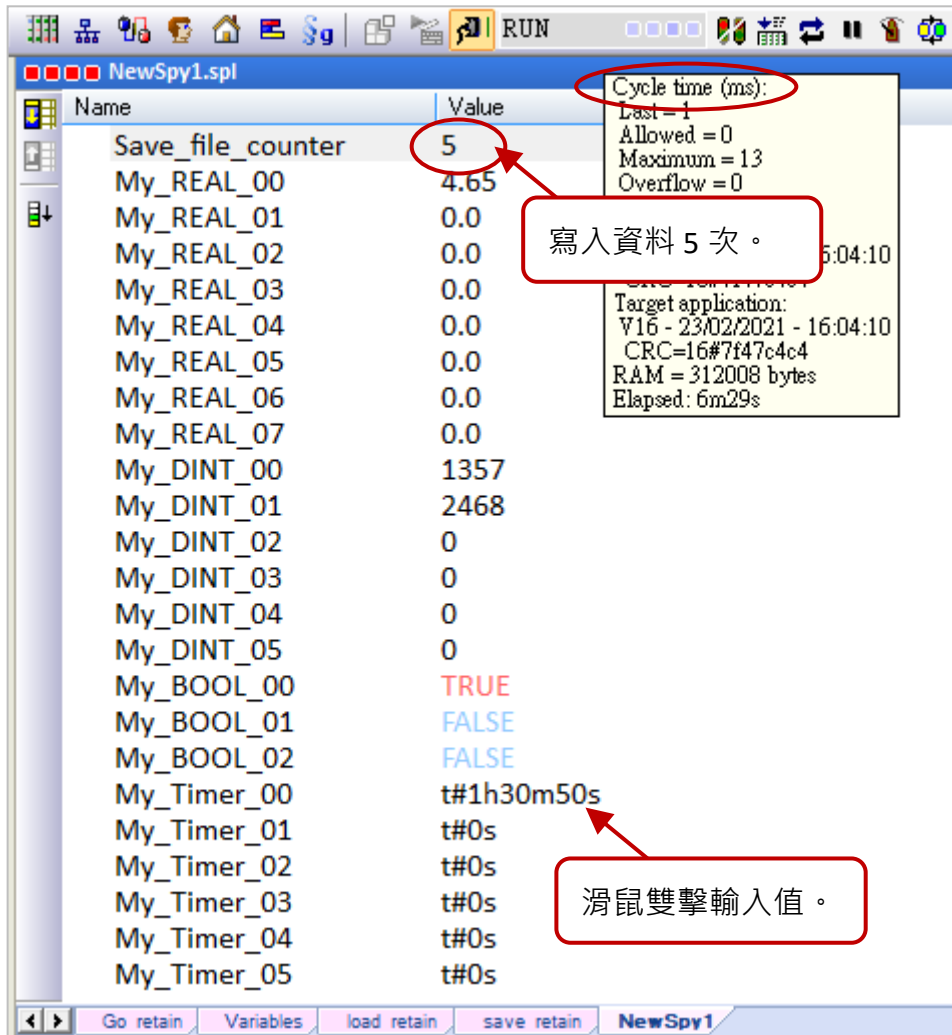
OTHER_VAR_now[6] := Any_to_DINT( My_BOOL_00 ) ;
OTHER_VAR_now[7] := Any_to_DINT( My_BOOL_01 ) ;
OTHER_VAR_now[8] := Any_to_DINT( My_BOOL_02 ) ;

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

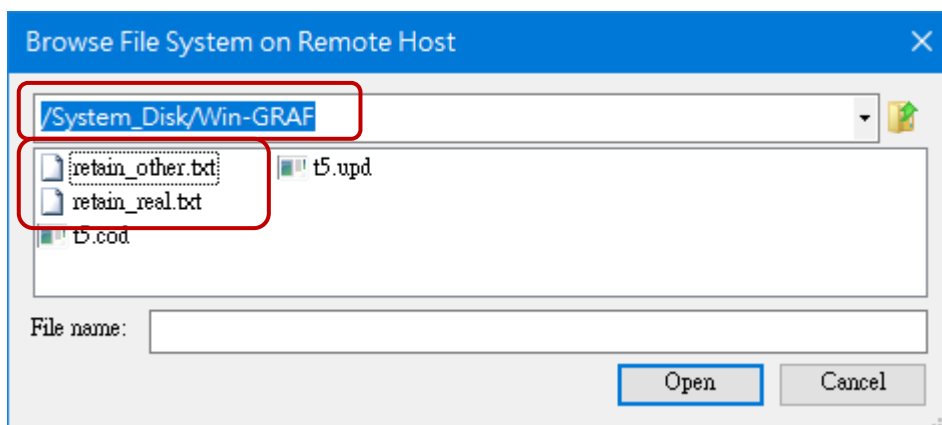

測試程式:

請確認通訊 IP 正確，並將專案下載到 PAC 中。在觀測清單中 (Spy list)，一開始所有數值皆為 "0" (或 "FALSE")，請變更幾個數值，然後 PAC 會在指定目錄下自動建立文字檔 (即，retain_real.txt 或 retain_other.txt) 並將資料寫入到該檔案中。

註: "Save_file_counter" 會顯示檔案更新的次數。此範例不適用在需頻繁更新 /System_disk/... 下檔案的需求。



您可使用 SSH 工具軟體來查看 Win-GRAP PAC 內的檔案。(參考 13.2 節)



6.3 將保存變數資料存到 FRAM

RPAC-2658M 內建有一個 FRAM 記憶體，可讓使用者讀/寫資料，此資料不會因 PAC 關機而消失。

FRAM 有以下特點：

優點： 在 "可保存變數" (Retain Variabl) 之外，提供了另一個可保存重要資料的方式。

缺點： 若使用 "可保存變數"，CPU 時間遠小於 1 ms；而 FRAM 的讀/寫動作會耗 CPU 時間。因此，
請勿太頻繁的調用 "EEP_Read" 與 "EEP_Write" 函式，會增加許多 PAC Cycle 時間。

註： `EEP_Read()` 與 `EEP_Write()` 可用來讀/寫 FRAM 或 EEPROM。

ST 語法：

```
(* 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 讀取記憶體一次 *)  
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記憶體 *)  
if New_Val <> Old_Val then  
    Old_Val := New_Val ;  
    tmp_bool := EEP_Write ( 1, New_Val );  
end_if;
```

```
(* 危險寫法： 記憶體可能很快損毀 *)
```

```
(* 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 讀取記憶體 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" 值寫入到記憶體一次 *)  
tmp_bool := EEP_Write ( 1, New_Val );
```

6.3.1 EEP_READ (讀取 FRAM)



使用小技巧:

按“F1”鍵，可查看詳細的設定說明。

Addr: (資料型態: “DINT”)

可設定位址為 1 ~ 1200。

若使用 64-bit 的資料類型 (例如: LINT 或 LREAL)，請設定位址為 1001 ~ 1200。

@Name :

指定一個變數名稱，用來儲存從 FRAM 讀到的資料。

(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。)

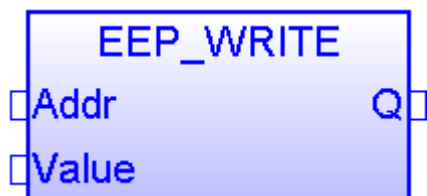
Q:

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

若變數的資料型態為 “REAL” 或 “LREAL”，但讀到的資料不是實數或發生其它錯誤，

“Q” 會回傳 “FALSE”。變數值 “0.0” 表示此資料不是實數值。

6.3.2 EEP_WRITE (寫入 FRAM)



Addr: (資料型態: “DINT”)

可設定位址為 1 ~ 1200。

若使用 64-bit 的資料類型 (例如: LINT 或 LREAL)，請設定位址為 1001 ~ 1200。

Value :

指定寫入 FRAM 的值。

(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。

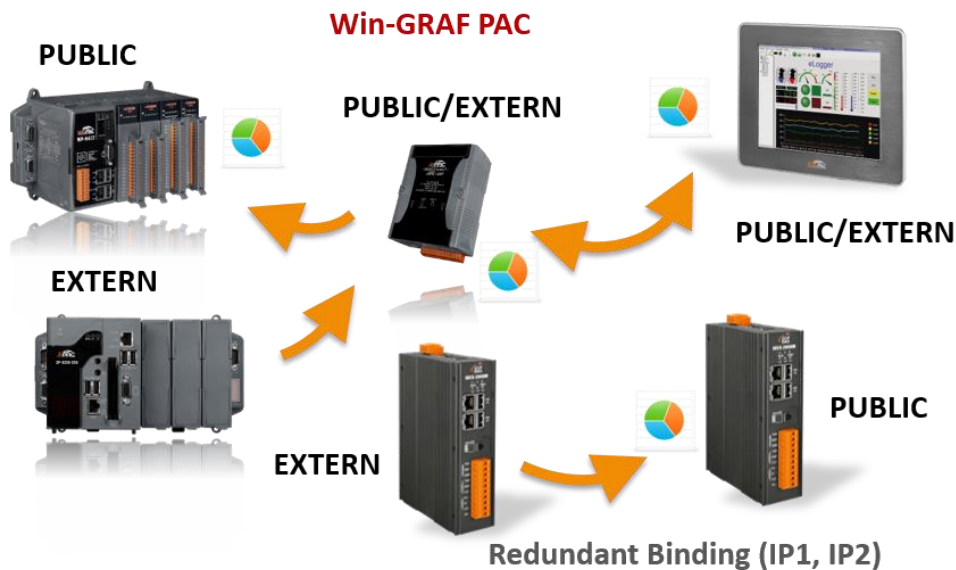
第 7 章 在 PAC 間互傳資料 (Data Binding)

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

PUBLIC: 是指公開資料給其他 PAC，或給同一台 PAC 內的 C 程式來使用。

EXTERN: 是指從別台 PAC 取回資料。

應用示意圖：

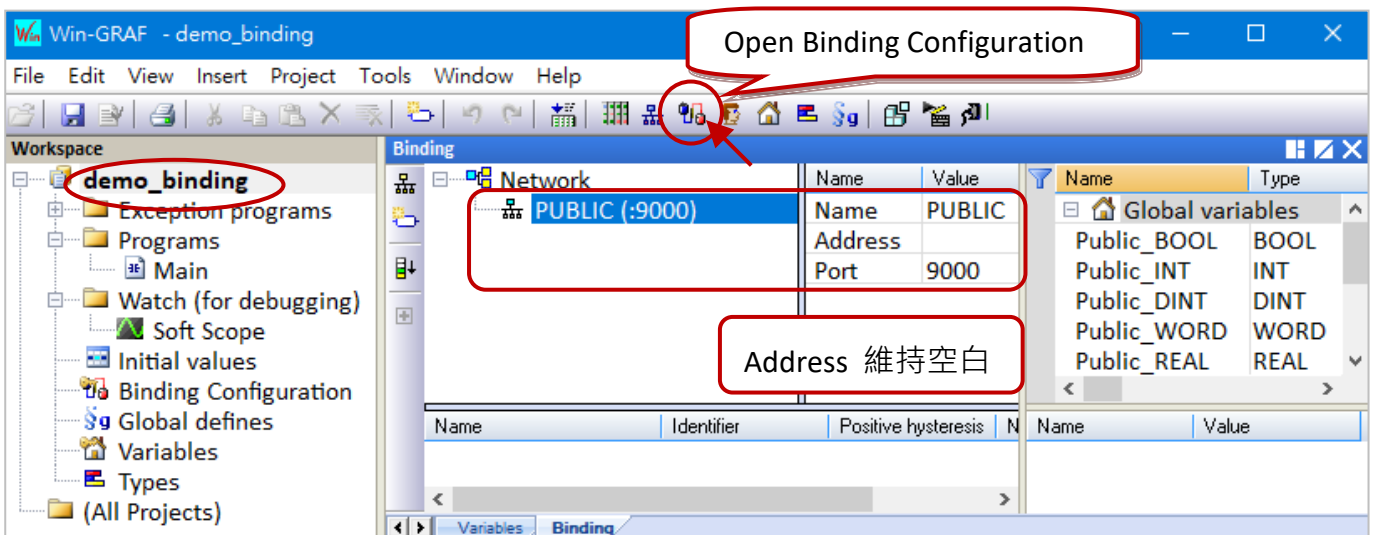


注意： 每台 RPAC-2658M 最多可使用的 "Binding" 功能為: 32 (EXTRN)。

"PUBLIC" 設定： 讓其他 PAC 可存取變數資料。

1. 滑鼠點選 "Open Binding Configuration" 按鈕來開啟 "Binding" 視窗。

"PUBLIC (:9000)" 表示透過 "Port: 9000" (編號固定，請勿更動) 來開放變數資料。

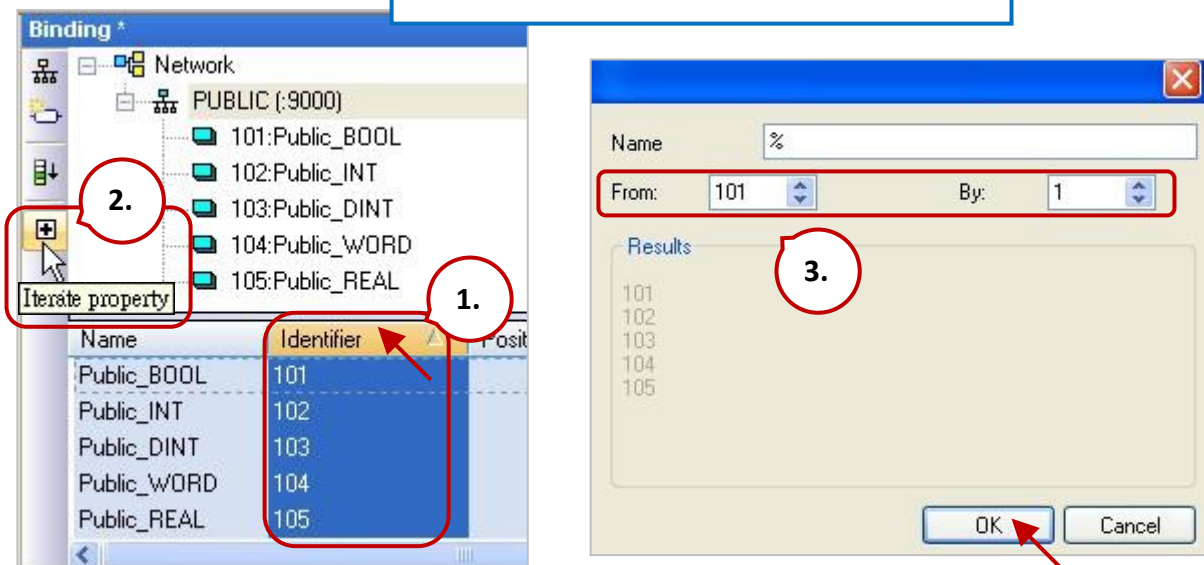
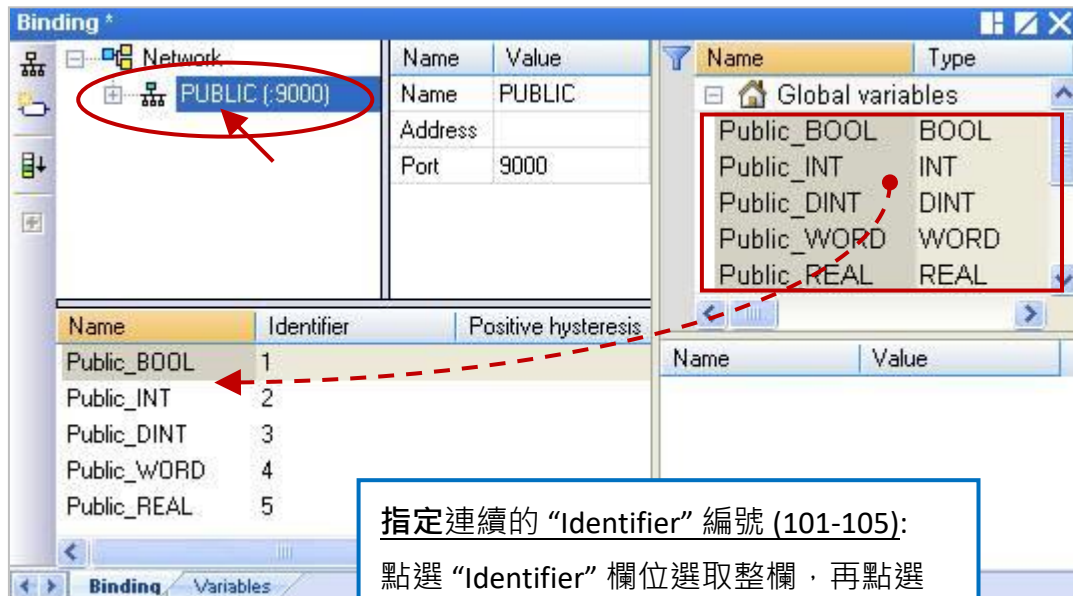


2. 在設定公開資料之前，請先建立好要公開的變數。下表為此範例所使用的變數：

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

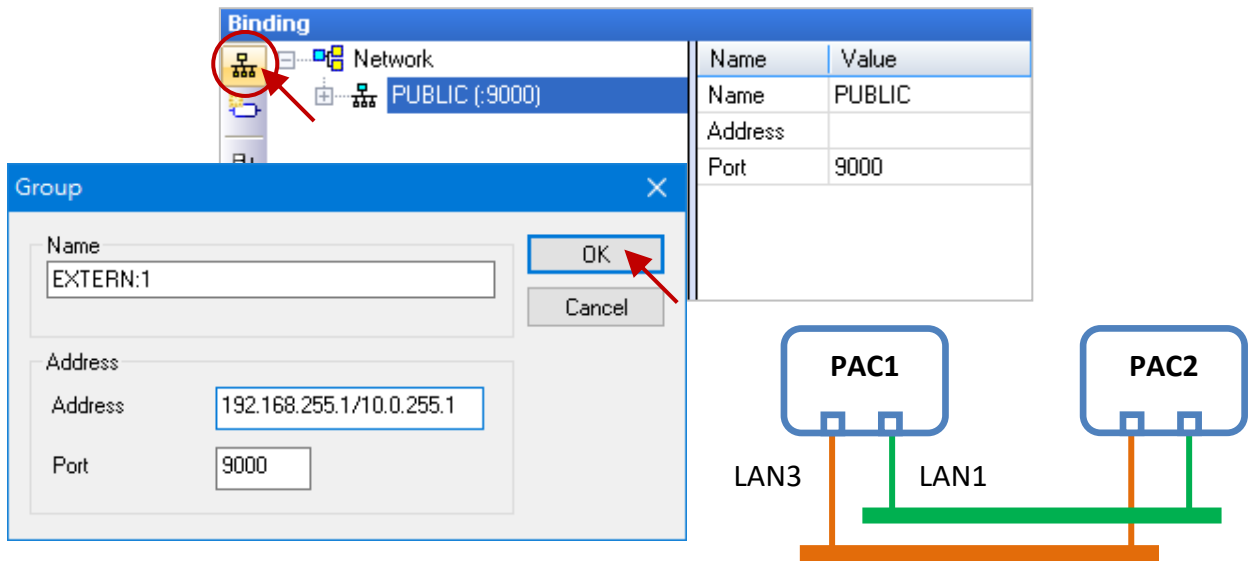
3. 滑鼠點選在 "PUBLIC (:9000)"，將所需的變數資料拖曳到 ID 對應區，"Identifier" 欄位會自動產生編號 (想取用該資料的 PAC，需設定一樣的 ID 編號)。

注意："PUBLIC" 最多可使用 8192 個變數，"Identifier" 編號只能是 "1 ~ 8192"。



"EXTERN" 設定: 取得其他 PAC 的資料。

- 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。

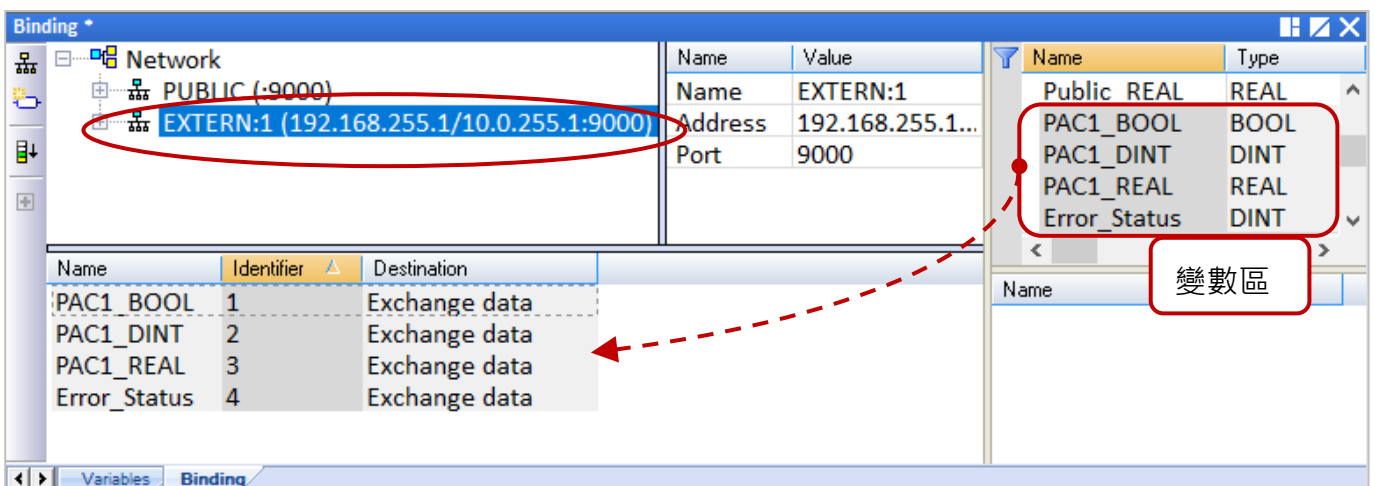
Address: 輸入要取得資料的那台 PAC 的 IP 位址 (例如: "192.168.255.1")。若該 PAC 有 2 個 Ethernet Port 也可輸入兩個 IP 位址 (LAN1: 192.168.x.x, LAN3: 10.0.x.x)。當一個 IP 位址發生問題，它會嘗試連線第 2 個 IP 位址。

Port: 固定為 "9000"，請勿更動。

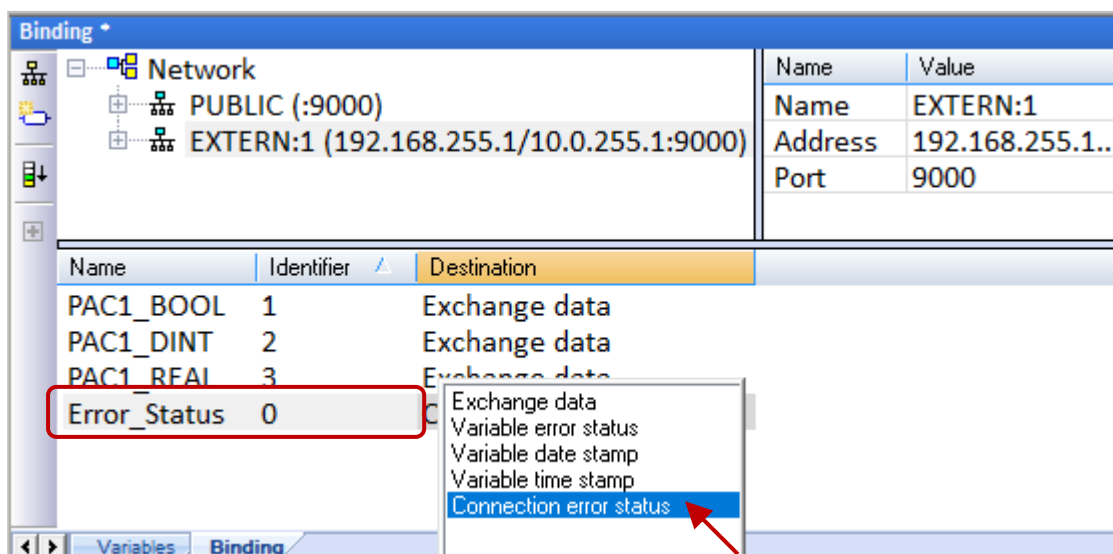
- 請在變數區建立好要取得資料的變數。右表為此範例所使用的變數:

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

- 請將所需的變數拖曳到 "EXTERN:1" 的 ID 對應區，"Identifier" 欄位會自動產生編號，請修改為和遠端 Public PAC 的公開變數一樣的 ID。



7. "Error_Status" 變數是用來了解 PAC 的通訊狀況，請將此 ID 設定為 "0" 再雙擊其 "Destination" 欄位將其設定為 "Connection error status"。



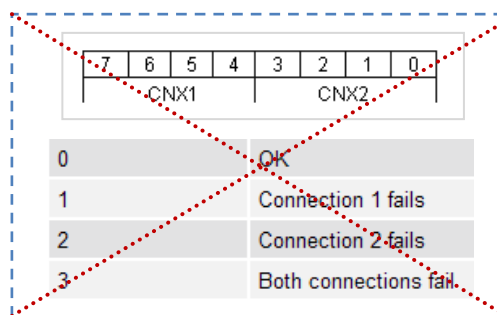
註:

若 "EXTERN" 內設定了 2 個 IP 位址，則 "Error_Status" 會回傳 2 個通訊狀態 (8-bit)。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

注意:

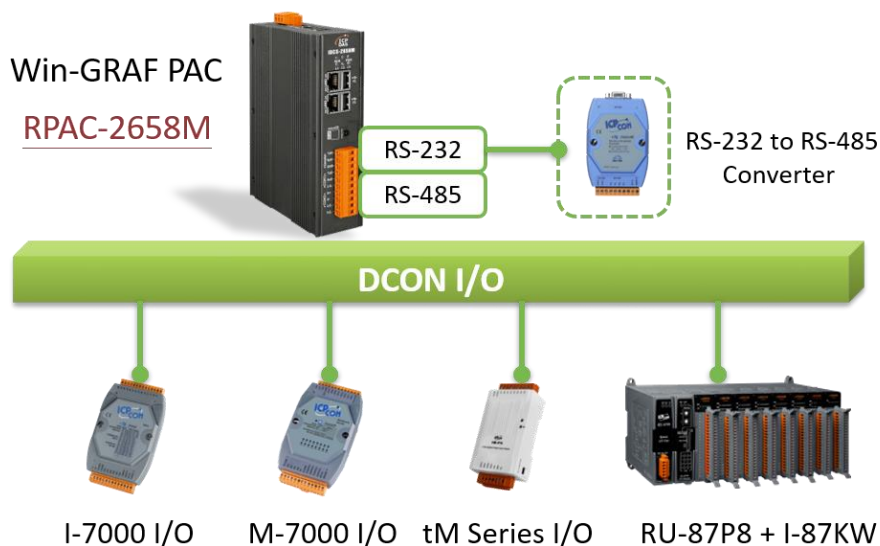
"HTML Help" (按 "F1" 鍵)，主題 - "Networked applications - Dual binding on redundant ETHERNET" 中 "Connection status" 與 "Variable status" 的說明是錯誤的，請忽略，並不符合 Win-GRAP PAC 的使用方式。



第 8 章 連接 DCON I/O 模組

Win-GRAF PAC 可透過 COM Port (RS-485) 來連接遠端 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: RU-87P4/8)。



產品網頁 - [遠程 I/O 模組與擴充單元](#)

產品網頁 - [DCON Utility Pro](#)

使用遠端 DCON I/O 模組之前，需使用 "DCON Utility Pro" 軟體設定每一個模組。

參數	設定值
Protocol	DCON
Address	1 ~ 255
Baudrate	建議設為 9600，與 Win-GRAF PAC 一樣
Checksum	建議設為 "Enabled"，與 Win-GRAF PAC 一樣
Data format 與 I/O 設定	依應用需求來設定

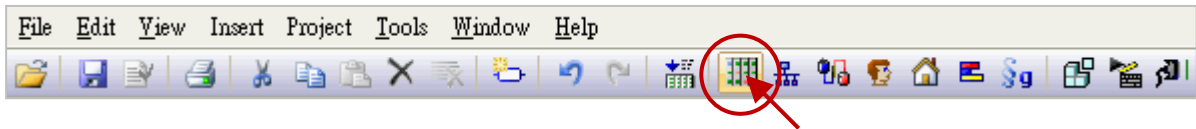
注意:

1. 若使用 類比輸入 (AI) 模組，則資料格式 (Data format) 必需設定為 "2's Complement"。
2. 若使用 類比輸出 (AO) 模組，則資料格式 (Data format) 必需設定為 "Engineering"。

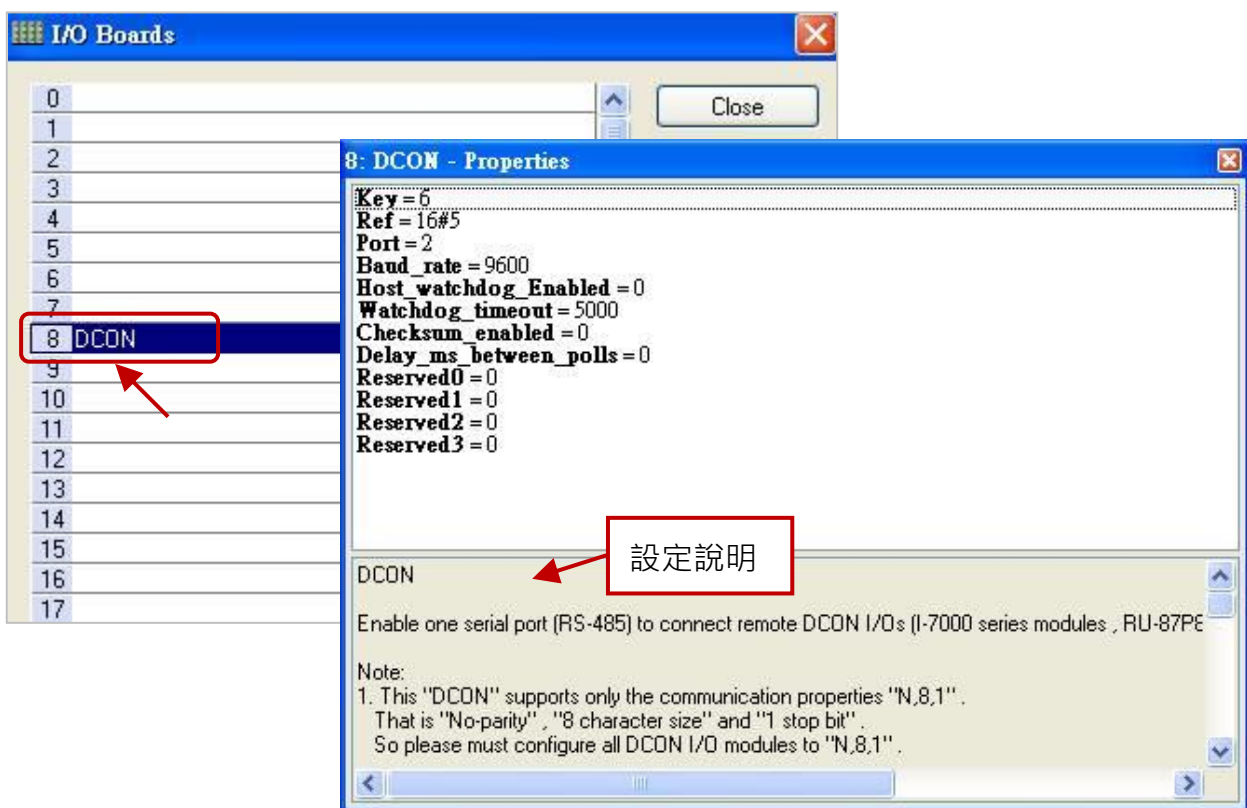
8.1 設定 "DCON" I/O 卡

"DCON" I/O 卡可用來啟用一個 RS-485 Port 來連接遠端的 DCON I/O 模組。一台 PAC 最多可啟用 16 個 DCON Port。

1. 點選 "Open I/Os" 按鈕來開啟 "I/O Boards" 視窗。



2. 滑鼠雙擊 Slot 編號加入 "DCON" I/O 卡，再滑鼠雙擊 "DCON" 來開啟 "Properties" 視窗。



參數說明:

註: 使用 "DCON" 通訊協定時，請設定 I/O 模組的資料格式 (Data Format) 為 "N,8,1"，表示 "無同位元"、"8 個資料位元" 與 "1 個停止位元"。

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

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

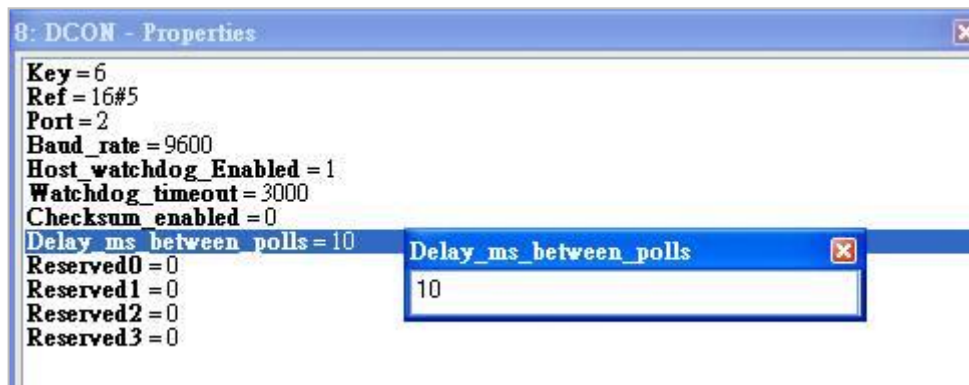
Host_watchdog_Enabled: 1: 表示啟用 Host-watchdog，0: 表示不啟用
設為非 0 的值，將會採用 "1"。

Watchdog_timeout: 單位: ms , 可設定為 “3000 ~ 25500” (即 3 ~ 25.5 秒)。
設為小於 “3000” , 將會採用 3000;
設為大於 “25500” , 將會採用 25500。
若 "Host_watchdog_Enabled" 設為 "0" , 會忽略此設定。

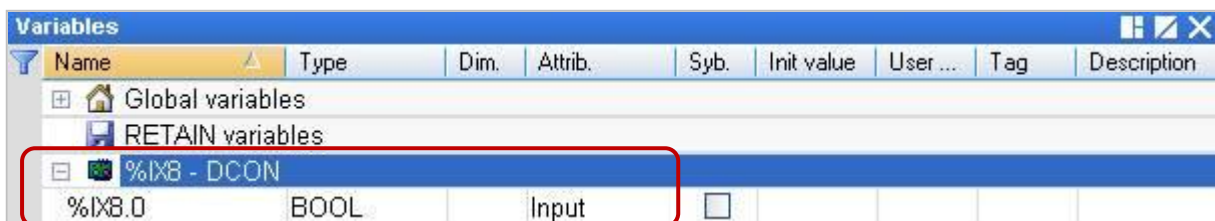
Checksum_enabled: 1: 表示啟用 Checksum , 0: 表示不啟用。
若設為非 0 的值 , 將會採用 "1"。
(為了通訊安全, 建議啟用 Checksum)

Delay_ms_between_polls: 單位: ms , 預設值為 0 , 有效範圍為 "0 ~ 1000"。
設為小於 “0” , 將會採用 0 ;
設為大於 “1000” , 將會採用 1000。
若沒有連接無線模組 , 可設為較小的值 (例如: 0 ~ 10)。
若連接無線模組 (例如: ZigBee I/O) , 可設為較大的值 (例如: 30 ~ 100
或 其它值) , 而值越大 , 輪詢 (Polling) 效率會較慢。

3. 滑鼠雙擊任一項目 , 來輸入設定值。



4. 在 “I/O Boards” 視窗加入 “DCON” 後 , 會自動在 “Variables” 視窗中新增 1 個 “BOOL” 輸入變數 , 可用來顯示 Win-GRAF PAC 上 COM Port 的連接狀態。(TRUE: 表示 OK ; FALSE: 表示錯誤。)



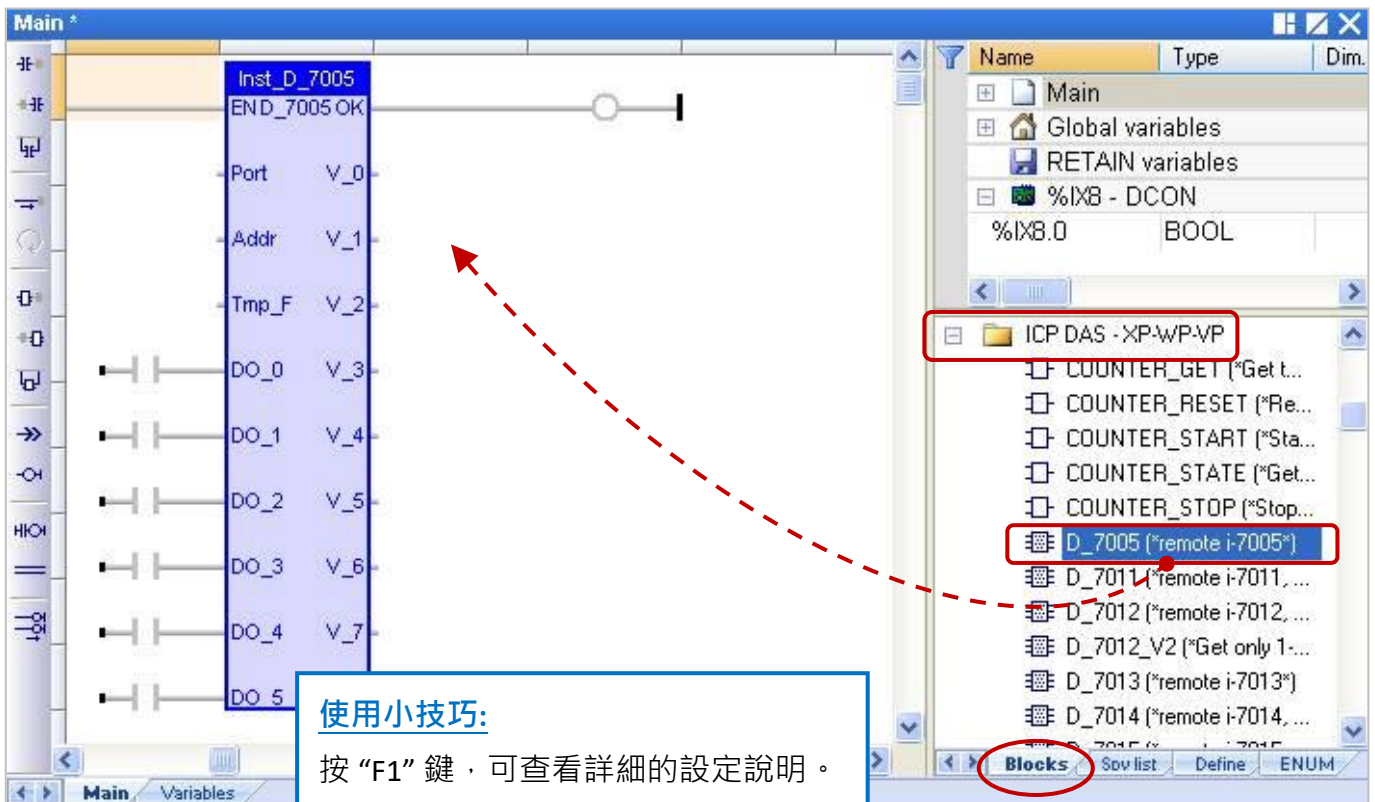
8.2 使用 I/O 功能方塊

Win-GRAF 支援許多遠端 DCON I/O 模組，您可開啟“函式庫管理員”(參考 1.2.2 節)或在 Help 說明(按“F1”鍵)來查看這些 Function 與 Function Block 的說明。

本章節將介紹“[D_7065](#)”、“[D_7018Z](#)”、“[D_7083](#)”、“[D_87084 freq](#)”、“[D_87084 cnt4](#)”、“[D_87084 cnt8](#)”、“[DL_100T485](#)”與 [D_GPS721](#) 功能方塊。

型號	說明
I-7065D-G	4 通道隔離型 DI 與 5 通道 Power Relay 模組 (有 LED 顯示器)
I-7065AD-G	4 通道隔離型 DI 與 5 通道 AC-SSR Output 模組 (有 LED 顯示器)
I-7065BD-G	4 通道隔離型 DI 與 5 通道 DC-SSR Output 模組 (有 LED 顯示器)
I-7018Z-G/S	10 通道熱電偶 (Thermocouple) 輸入模組
I-7083D-G	3 軸、32-bit Encoder/Counter 輸入模組 (有 LED 顯示器)
I-7083BD-G	3 軸、32-bit Encoder/Counter 輸入模組 (有虛擬電池備援與 LED 顯示器)
I-87084W-G	4/8 通道 Counter/Frequency/Encoder 輸入模組
DL-100T485	遠端溫/溼度資料紀錄器 (RS-485，有 LED Display)
GPS-721	遠端 GPS 接收模組，具有 RS-485 (Asia Only)

在 LD 程式的 **Blocks** 面板中，“ICP DAS – XP-WP-VP”資料夾內有許多 Function 與 Function Block。請拖曳所需的機能，到程式編輯區來使用。



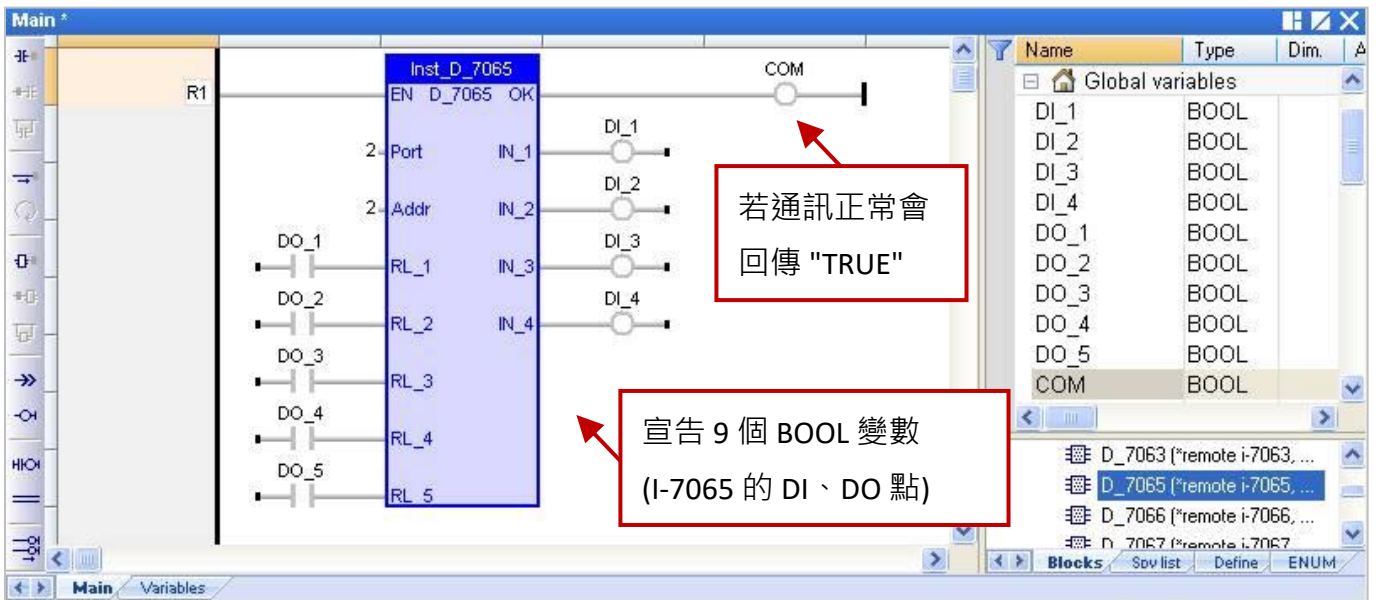
8.2.1 “D_7065” 功能方塊

“D_7065”: 用來連結一個遠端的 I-7065D、I-7065AD 或 I-7065BD DCON 模組。

註:

1. 請確認已使用 “DCON Utility Pro” 設定過 I/O 模組 (見第 8 章)。
2. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_d_7065.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 “File - Add Existing Project - From Zip...” 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時，接收的 DI 值才正確。

範例: 透過 PAC 的 COM2 來連接 I-7065 (位址 = 2)，它具有 4 個 DI 與 5 個 Relay Output 通道。



輸入參數	資料型態	說明
EN	BOOL	“TRUE”: 啟用; “FALSE”: 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)
RL_1 ~ RL_5	BOOL	5 通道 DO 值

輸出參數	資料型態	說明
OK	BOOL	“TRUE”: 通訊正常; “FALSE”: 通訊失敗。
IN_1 ~ IN_4	BOOL	4 通道 DI 值

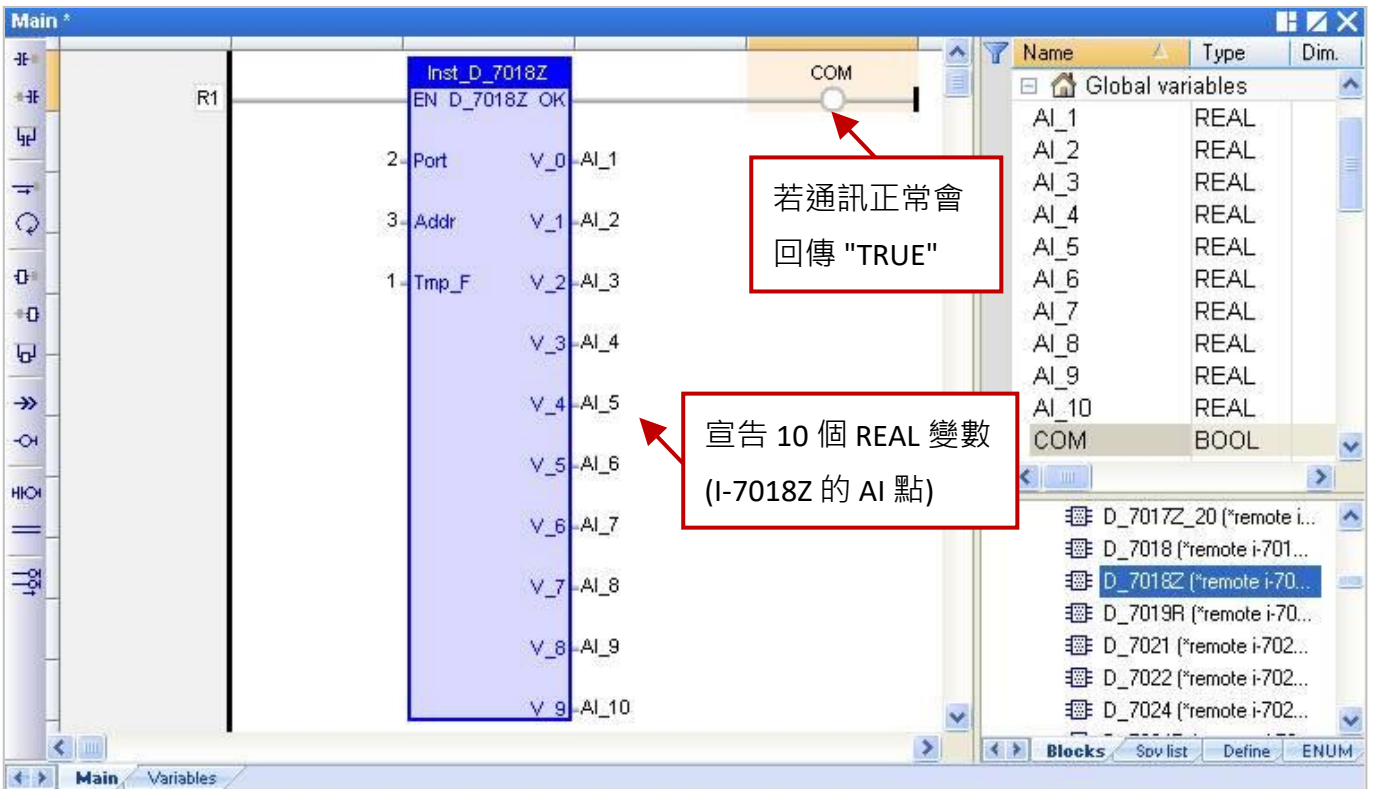
8.2.2 “D_7018Z” 功能方塊

“D_7018Z”：用來連結一個遠端的 I-7018Z DCON 模組，來量測電壓、電流或溫度。

註：

1. 請確認已使用 “DCON Utility Pro” 設定過 I/O 模組 (見第 8 章)，且 AI 模組的資料格式設為 “2’s Complement”，否則 Win-GRAF 無法讀到正確的值。
2. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_d_7018z.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 “File - Add Existing Project - From Zip...” 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時，接收的 AI 值才正確。

範例: 透過 PAC 的 COM2 連接 I-7018Z (位址 = 3)，來量測攝氏溫度。



輸入參數	資料型態	說明
EN	BOOL	“TRUE”: 啟用；“FALSE”: 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)
Tmp_F	DINT	溫度格式可設定為 1 或 2。 1: 攝氏；2: 華氏 若設定為其它值，則會取用 “1: 攝氏溫度”。

輸出參數	資料型態	說明
OK	BOOL	"TRUE": 通訊正常；"FALSE": 通訊失敗。
V_0 ~ V_9	REAL	<p>10 通道 AI 值，可在 "DCON Utility Pro" 設定 Type Code。</p> <p>若設定為溫度類型，則回傳值的單位為 "度"。</p> <p>例如，回傳值為 25.75 表示 25.75 度。</p> <p>若設定為電壓類型，則回傳值的單位為 "V"。</p> <p>例如，回傳值為 0.85421 表示 0.85421 V 或 854.21 mV。</p> <p>若設定為電流類型，則回傳值的單位為 "mA"。</p> <p>例如，回傳值為 1.5567 表示 1.5567 mA。</p>

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 與 溫度感測器的設定不相符。
4. 感測器量到錯誤的電阻值。

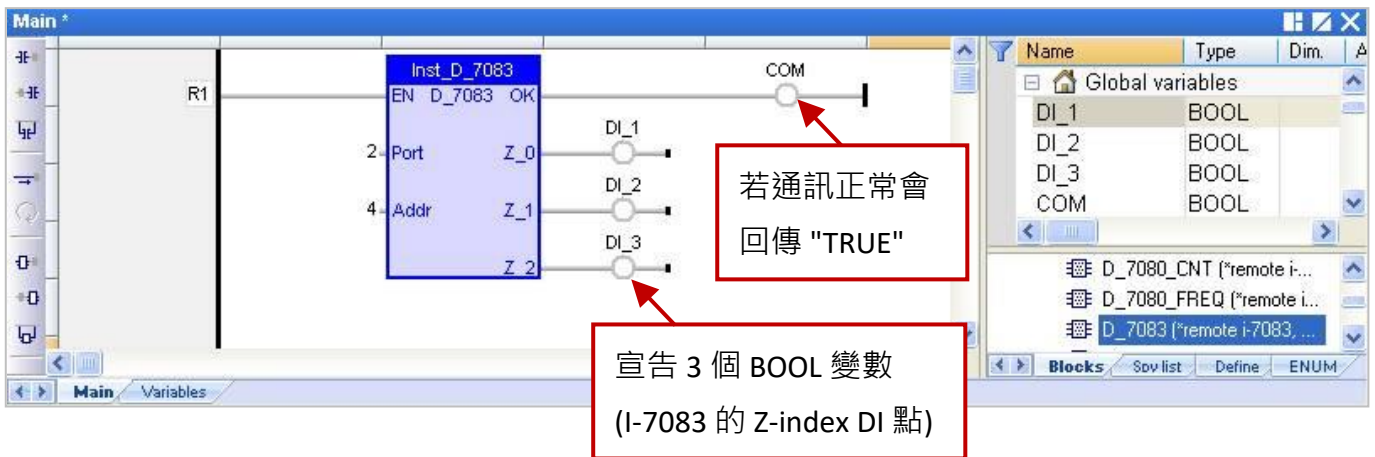
8.2.3 “D_7083” 功能方塊

“D_7083”: 用來連結一個遠端的 I-7083D 或 I-7083BD DCON 模組。

註:

1. 請確認已使用 “DCON Utility Pro” 設定過 I/O 模組 (見第 8 章) , 且 AI 模組的資料格式設為 “2’s Complement” , 否則 Win-GRAF 無法讀到正確的值。
2. 請確認已在 “I/O boards” 視窗加入 “DCON” , 並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_d_7083.zip\)](#) , 並在 Win-GRAF Workbench 點選功能表 “File - Add Existing Project - From Zip...” 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時 , 接收的 DI 值才正確。
5. 為了取得 I-7083D/BD 模組的 Encoder 值 , 需先使用 “D_7083” 功能方塊 , 再搭配 “Counter_xxxx” 函式 ([參考 8.3 節](#)) 來實作 Encoder 功能。

範例: 透過 PAC 的 COM2 來連接 I-7083 (位址 = 4) , 它具有 3 個 DI 通道。



輸入參數	資料型態	說明
EN	BOOL	“TRUE”: 啟用 ; “FALSE”: 不啟用。
Port	DINT	COM Port 編號 , 視 PAC 而定可設定為 1 ~ 37。 (需設定為常數 , 不能是變數)。
Addr	DINT	模組的 Net-ID 位址 , 可設定為 1 ~ 255。 (需設定為常數 , 不能是變數)

輸出參數	資料型態	說明
OK	BOOL	“TRUE”: 通訊正常 ; “FALSE”: 通訊失敗。
Z_0 ~ Z_2	BOOL	3 軸 Z-index DI 值。

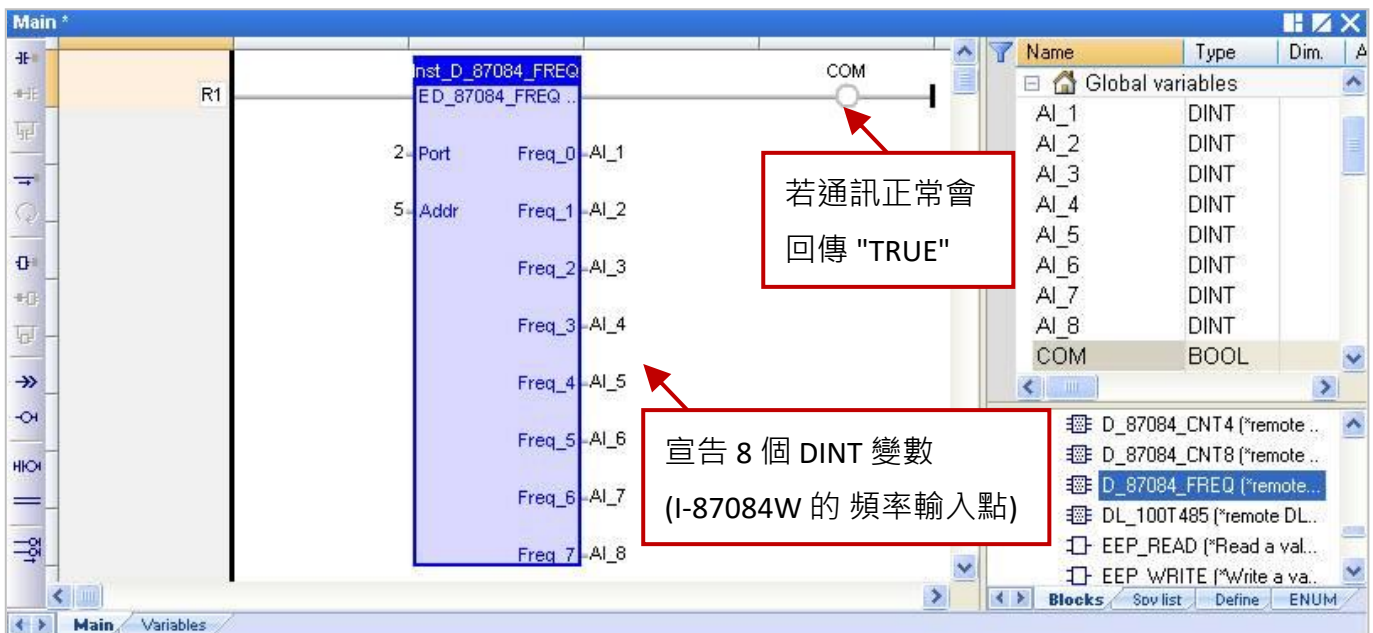
8.2.4 “D_87084_FREQ” 功能方塊

"D_87084_freq": 用來連結一個在 RS-485 I/O 擴充單元 (例如: RU-87P4/8 或 I-87K4/5/8/9) 上的 I-87084W 模組，來量測頻率值。

註:

1. 請確認已使用 “DCON Utility Pro” 設定過 I/O 模組 (見第 8 章)，且 I-87084W 的資料格式 (Frequency format) 需設為 "Hex format"，否則該功能會無效。
2. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_d_87084_fr.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時，接收的 AI 值才正確。

範例: 透過 PAC 的 COM2 連接 I-87084W (位址 = 5)，來量測 8 個通道的頻率值。



輸入參數	資料型態	說明
EN	BOOL	"TRUE": 啟用；"FALSE": 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)

輸出參數	資料型態	說明
OK	BOOL	"TRUE": 通訊正常；"FALSE": 通訊失敗。
Freq_0 ~ Freq_7	DINT	8 通道頻率值 (單位: Hz)。

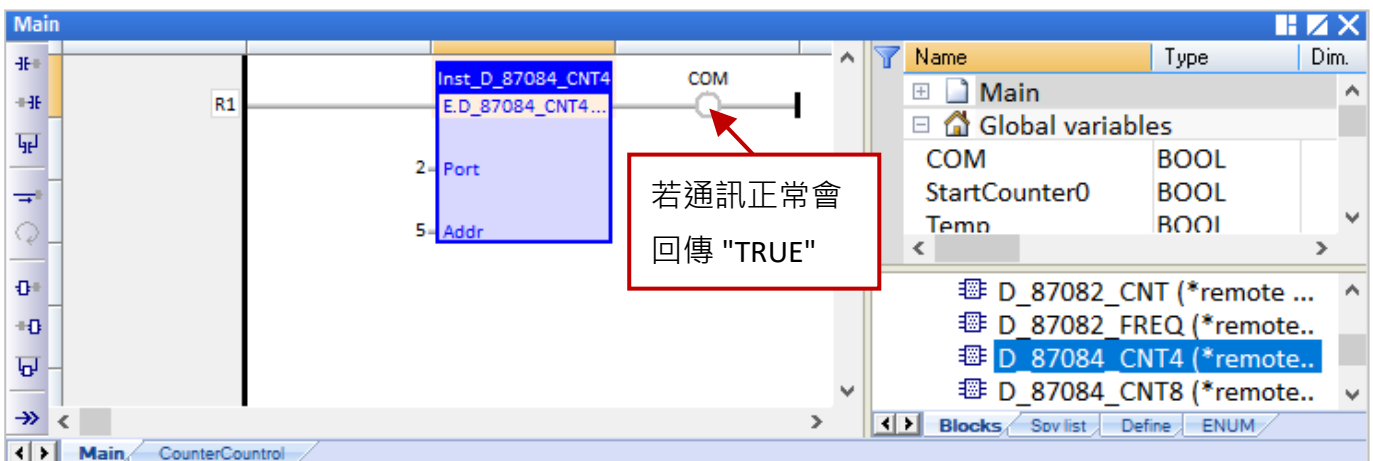
8.2.5 “D_87084_CNT4” 功能方塊

"D_87084_CNT4": 用來連結一個在 RS-485 I/O 擴充單元 (例如: I-87K4/5/8/9、RU-87P4/8) 上的 I-87084W 遠端模組，來量測 4 個通道的 Counter 值。

註:

1. 請確認已使用 “DCON Utility Pro” 設定過 I/O 模組 (見第 8 章)，且 I-87084W 的資料格式設為 "Hex format"，否則該功能會無效。
2. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_d_87084_c4.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時，Counter 值才正確。
5. 為了取得 I-87084W 模組的 Counter 值，需先使用 "D_87084_CNT4" 功能方塊，再搭配 "Counter_xxxx" 函式 ([參考 8.3 節](#)) 來實作 Counter 功能。

範例: 透過 PAC 的 COM2 連接 I-87084W (位址 = 5)，來量測 4 個通道的 Counter 值。



輸入參數	資料型態	說明
EN	BOOL	"TRUE": 啟用；"FALSE": 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)

輸出參數	資料型態	說明
OK	BOOL	"TRUE": 通訊正常；"FALSE": 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. 請確認已使用 “DCON Utility Pro” 設定過 I/O 模組 (見第 8 章)，且 I-87084W 的資料格式設為 "Hex format"，否則該功能會無效。
2. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_d_87084_c8.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時，Counter 值才正確。
5. 為了取得 I-87084W 模組的 Counter 值，需先使用 "D_87084_CNT8" 功能方塊，再搭配 "Counter_xxxx" 函式 ([參考 8.3 節](#)) 來實作 Counter 功能。

範例: 透過 PAC 的 COM2 連接 I-87084W (位址 = 5)，來量測 8 個通道的計數值。



輸入參數	資料型態	說明
EN	BOOL	"TRUE": 啟用；"FALSE": 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)

輸出參數	資料型態	說明
OK	BOOL	"TRUE": 通訊正常；"FALSE": 通訊失敗。

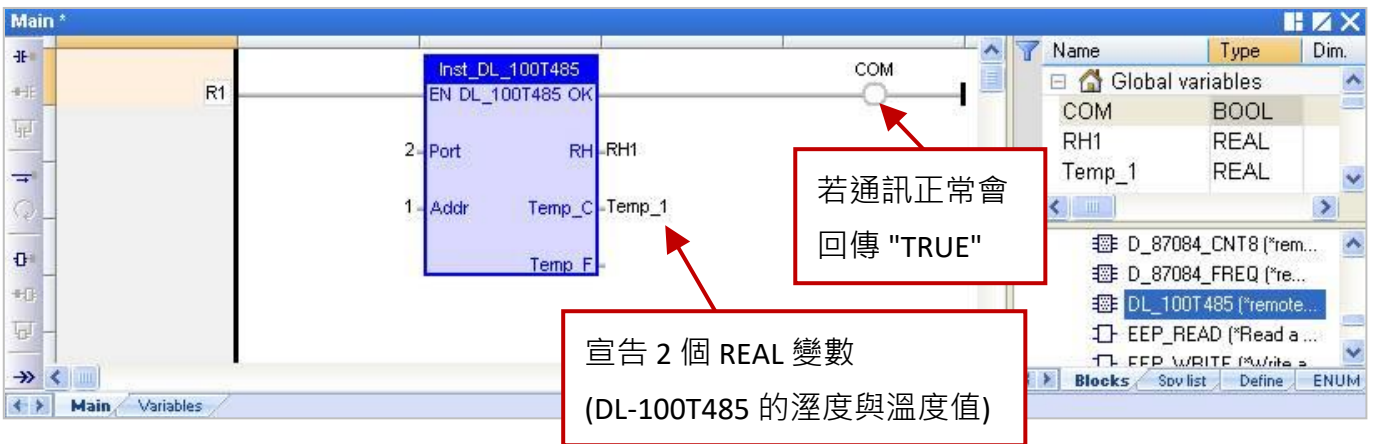
8.2.7 “DL_100T485” 功能方塊

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

註:

1. 請確認已使用 "[DL Utility](#)" 設定過 I/O 模組，DL-100T485 預設值 - Address (ID) 為 "1"、Baudrate 為 "9600"、Checksum 為 "Disable"。([下載 DL-100T485 使用手冊](#))
2. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
3. 請在網站下載 [範例程式 \(demo_dl_100T485.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。
4. 只有當模組的通訊狀態正常 (TRUE) 時，接收的 AI 值才正確。

範例: 透過 PAC 的 COM2 連接 DL_100T485 (位址 = 1)，來量測溫/濕度值。



輸入參數	資料型態	說明
EN	BOOL	"TRUE": 啟用；"FALSE": 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)

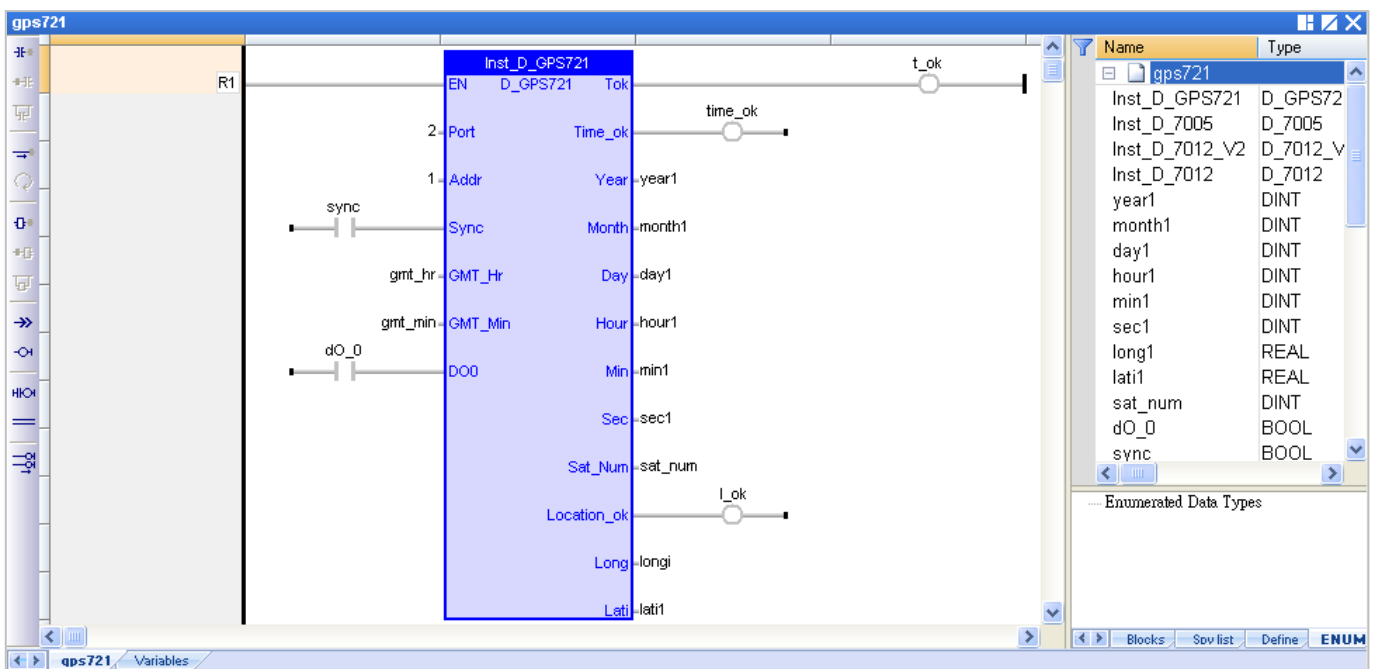
輸出參數	資料型態	說明
OK	BOOL	"TRUE": 通訊正常；"FALSE": 通訊失敗。
RH	REAL	相對溼度 (單位: %)。 例如: 回傳值為 "45.7" 表示 45.7%。
Temp_C	REAL	攝氏溫度 例如: 回傳值為 "25.7" 表示 25.7 °C。
Temp_F	REAL	華氏溫度 例如: 回傳值為 "78.26" 表示 78.26 °F。

8.2.8 “D_GPS721” 功能方塊

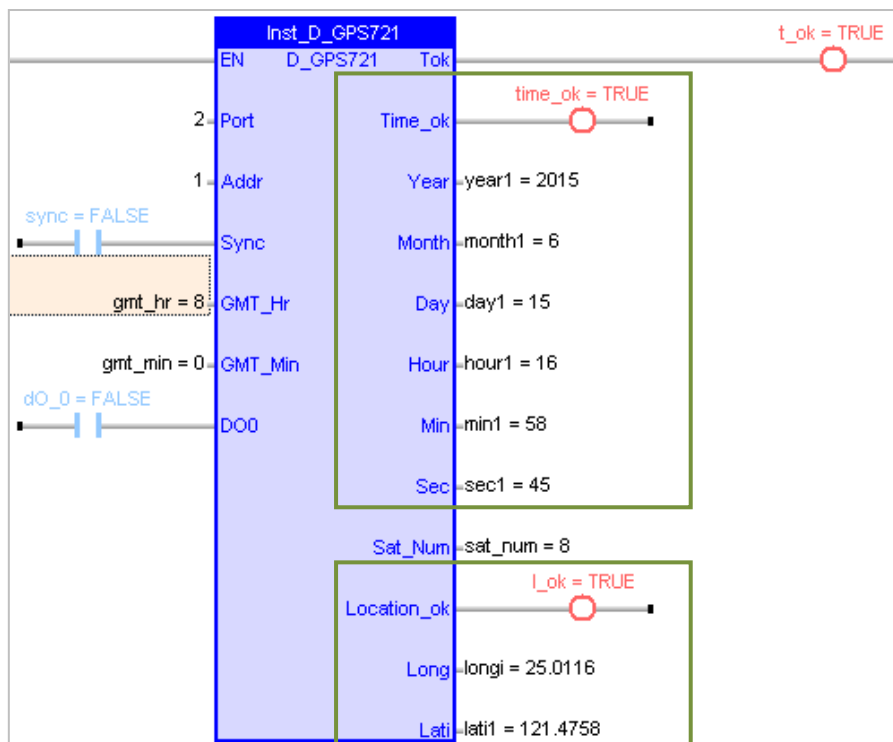
“D_GPS721” 可用來連結一個 “GPS-721” 遠端 GPS 接收模組 (內含 1 DO, 1 PPS 輸出通道)。

註:

1. 一台 PAC 只能使用一個 GPS-721 模組。
2. 請確認已使用 “DCON Utility Pro” 設定過模組 (見第 8 章)，GPS-721 的預設值 - Address (ID) 為 "1"、Baudrate 為 "9600"、Checksum 為 "Disable"。
3. 請確認已在 “I/O boards” 視窗加入 “DCON”，並填入正確的參數 (見 8.1 節)。
4. 請在網站下載 [範例程式 \(dmeo_gps721.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。



輸入參數	資料型態	說明
EN	BOOL	“TRUE”: 啟用; “FALSE”: 不啟用。
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。 (需設定為常數，不能是變數)。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。 (需設定為常數，不能是變數)。
Sync	BOOL	時間同步功能，設為 “TRUE” 啟用; “FALSE”: 不啟用。 當 GPS-721 與 PAC 時間差距 5 秒 (或更多) 會自動校正 PAC 時間。
GMT_Hr & GMT_Min	DINT	時區，例如: 北京 與 台北是 GMT+8 (GMT_Hr = 8, GMT_Min = 0)， 美國是 GMT-6 (GMT_Hr = -6, GMT_Min = 0) 與 新德里是 GMT+5.5 (GMT_Hr = 5, GMT_Min = 30)。
DO0	BOOL	GPS-721 模組的數位輸出通道。



輸出參數	資料型態	說明
Tok	BOOL	TRUE: GPS-721 通訊正常 ; FALSE: 通訊失敗 , 以下回傳值皆無意義。
Time_ok	BOOL	TRUE: Year、Month、Day、Hour、Min、Sec 為有效值。 FALSE: Year、Month、Day、Hour、Min、Sec 為無效值。
Year、Month Day、Hour Min、Sec	DINT	年 (2009 ~ ...)、月 (1 ~ 12)、日 (1 ~ 31)、時 (0 ~ 23)、分 (0 ~ 59)、秒 (0 ~ 59)。 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。
Sat_Num	DINT	衛星的使用數量 (0: 未發現衛星 或 使用 1 ~ 9 顆衛星)。
Location_ok	BOOL	注意: 只有 "Location_ok" 為 "TRUE" 時 , Long 與 Lati 值才正確。 TRUE: GPS-721 已取得目前位置的經/緯度。 FALSE: Long, Lati 為無效值。
Long	REAL	經度 (正數: 表示東方 , 負數: 表示西方)。 例如: "25.0121" 表示東經 25.0121 度。
Lati	REAL	緯度 (正數: 表示北方 , 負數: 表示南方)。 例如: "121.4576" 表示北緯 121.4576 度。

8.3 I-87082W, I-87084W, I-7083 與 I-7080 模組的計數功能

此章節將介紹如何在 LD 或 ST 程式中使用 "COUNTER_START"、"COUNTER_STOP"、"COUNTER_GET"、"COUNTER_STATE" 與 "COUNTER_RESET" 函式 (Function) 來操作 Counter/Encoder 功能。

注意:

- 請先建立好 I/O Function Block，否則 "COUNTER_XXX" Function 會無效。
- 可將 I-87K I/O 模組裝在 RS-485 I/O 擴充單元上 (例如: RU-87P4/8)。

Model	Function Block	Encoder	Counter	Frequency
I-7080	"D_7080"		✓	✓
I-7083	"D_7083"	✓ (8.2.3 節)	✓	
I-87082W	"D_87082"		✓	✓
I-87084W	"D_87084_cnt4" 或 "D_87084_cnt8"	✓	✓ (8.2.5 節)	✓ (8.2.4 節)

8.3.1 COUNTER_START (開始計數)

說明: 開始計數器或編碼器的計數。

ST 語法:

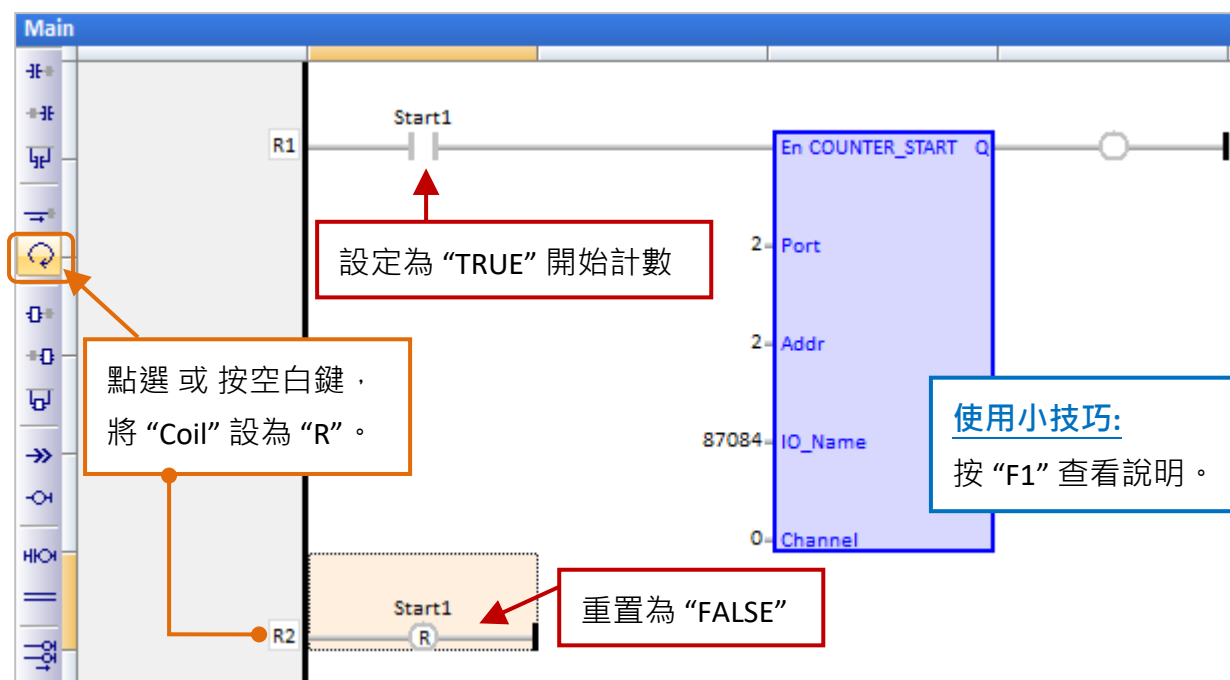
```
IF Start1 = TRUE THEN
  Start1 := FALSE ;
  TMP_BOOL := Counter_Start (Port, Addr, IO_Name, Channel) ;
END_IF ;
```

註:

可先在變數區宣告 "Start1" 與 "TMP_BOOL" 布林變數。

LD 語法:

“Start1” (BOOL): 設定為 “TRUE” 啟動計數，並重置 “Start1” 為 “FALSE”。



輸入參數	資料型態	說明
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。
IO_Name	DINT	Counter/Encoder 模組的名稱，可設定為 "87084"、"87082"、"7083" 與 "7080"。
Channel	DINT	Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...。

輸出參數	資料型態	說明
Q	BOOL	"TRUE": OK ; "FALSE": 錯誤。

8.3.2 COUNTER_STOP (停止計數)

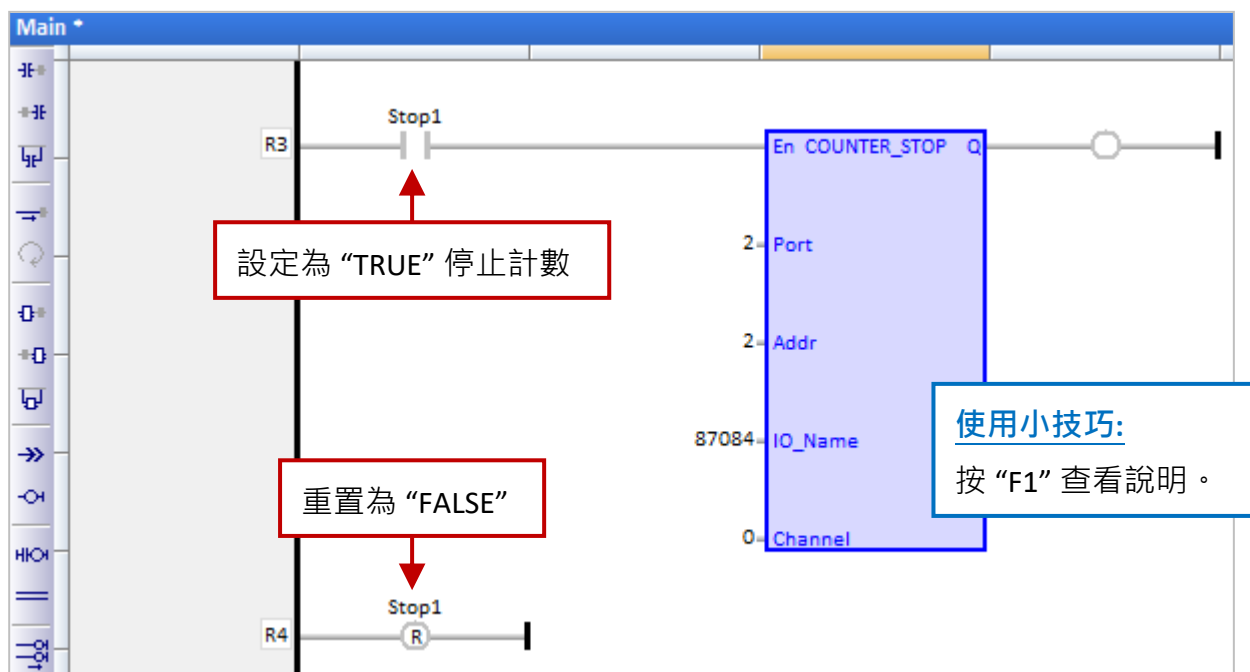
說明: 停止計數器或編碼器的計數。

ST 語法: 可先在 "變數區" 建立一個 "Stop1" 與 "TMP_BOOL" 布林變數。

```
IF Stop1 = TRUE THEN
    Stop1 := FALSE;
    TMP_BOOL := Counter_Stop (Port, Addr, IO_Name, Channel);
END_IF;
```

LD 語法:

"Stop1" (BOOL): 設定為 "TRUE" 停止計數，且 "Start1" 會重置為 "FALSE"。



輸入參數	資料型態	說明
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。
IO_Name	DINT	Counter/Encoder 模組的名稱，可設定為 "87084"、"87082"、"7083" 與 "7080"。
Channel	DINT	Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...。

輸出參數	資料型態	說明
Q	BOOL	"TRUE": OK ; "FALSE": 錯誤。

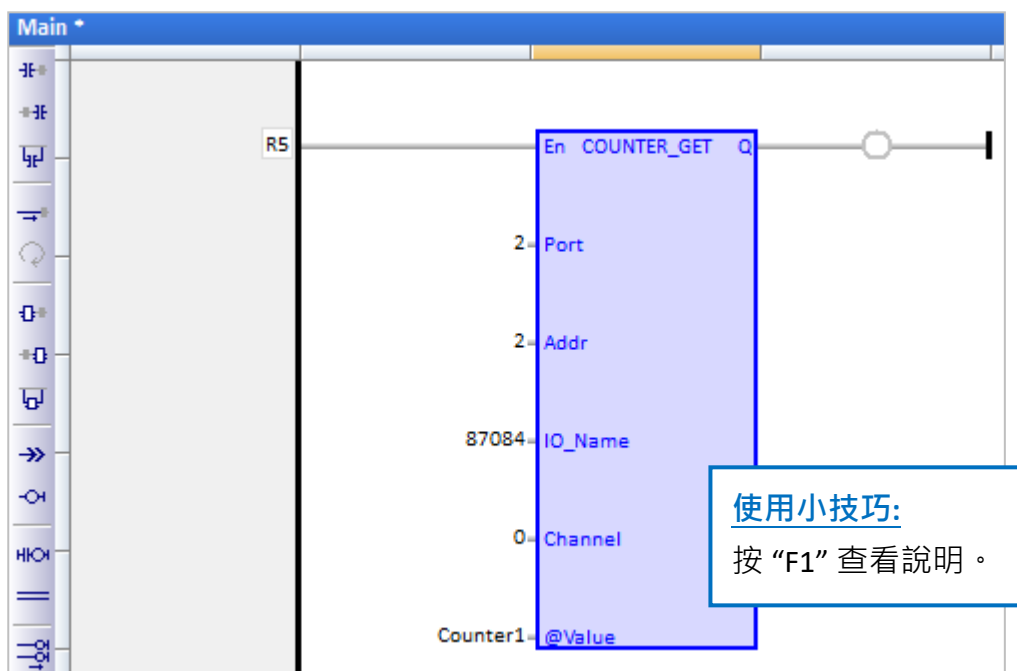
8.3.3 COUNTER_GET (取得計數值)

說明: 取得計數器或編碼器的計數值。

ST 語法: 可先在 "變數區" 建立 "TMP_BOOL" (BOOL) 與 "Encoder_1" (DINT) 變數。

TMP_BOOL := Counter_Get (Port, Addr, IO_Name, Channel, Counter1);

LD 語法:



輸入參數	資料型態	說明
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。
IO_Name	DINT	Counter/Encoder 模組的名稱，可設定為 "87084"、"87082"、"7083" 與 "7080"。
Channel	DINT	Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...。
@Value	"DINT", "UDINT", "DWORD" 或 "LINT"	回傳計數器或編碼器的值。 (關於資料範圍，可參考 附錄 A)

輸出參數	資料型態	說明
Q	BOOL	"TRUE": OK ; "FALSE": 錯誤。

8.3.4 COUNTER_STATE (取得計數狀態)

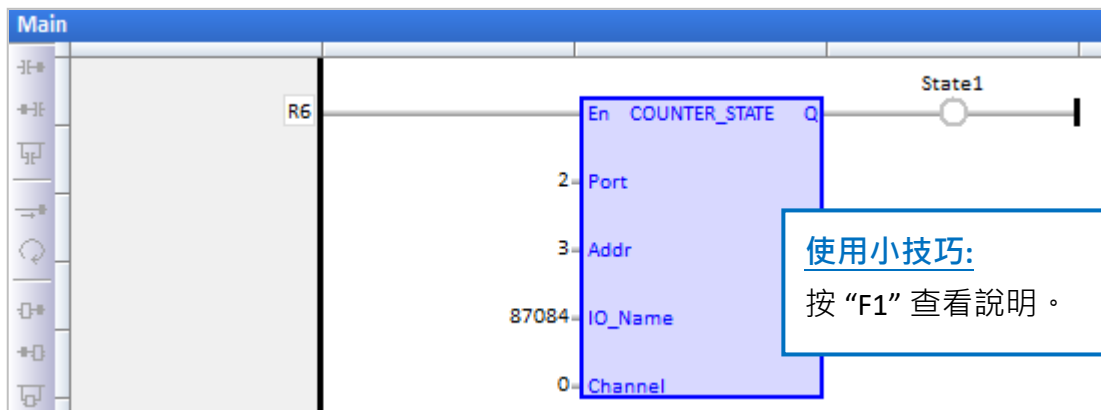
說明: 取得計數器或編碼器的計數狀態。

ST 語法: 可先在 "變數區" 建立一個 "TMP_BOOL" 布林變數。

```
TMP_BOOL := Counter_State (Port, Addr, IO_Name, Channel);
```

LD 語法:

"State1" (BOOL): 顯示計數的狀態。



輸入參數	資料型態	說明
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。
IO_Name	DINT	Counter/Encoder 模組的名稱，可設定為 "87084"、"87082"、"7083" 與 "7080"。
Channel	DINT	Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...。

輸出參數	資料型態	說明
Q	BOOL	"TRUE": OK ; "FALSE": 錯誤。

8.3.5 COUNTER_RESET (重置計數值)

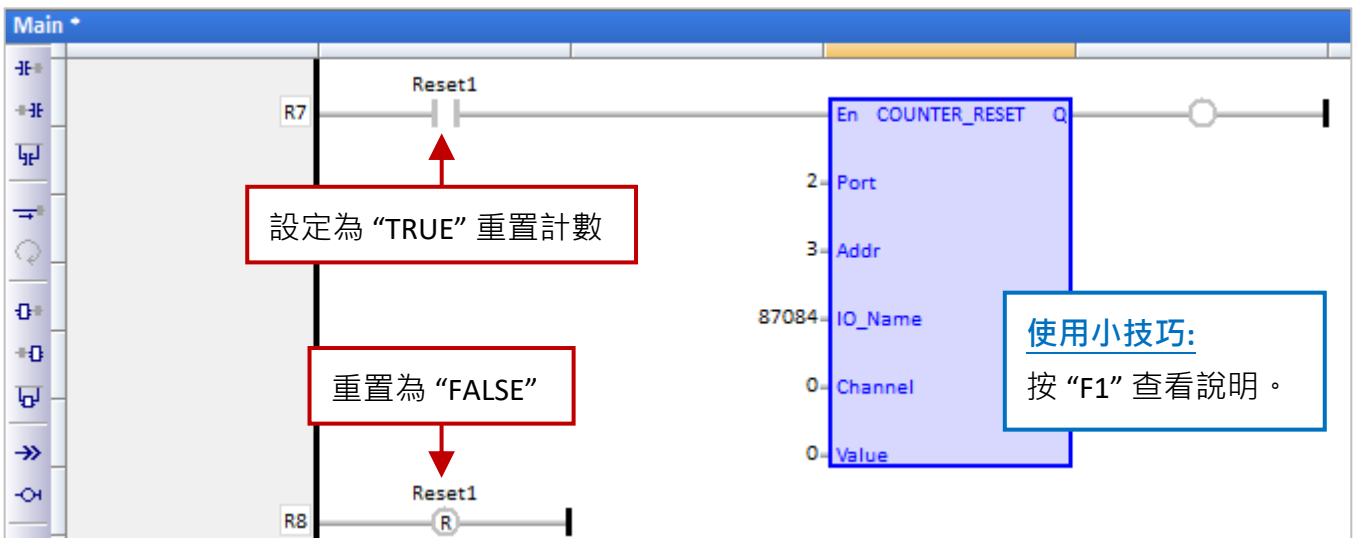
假設：將計數器或編碼器重置為指定的值。

ST 語法: 可先在 "變數區" 建立一個 "Reset 1" 與 "TMP_BOOL" 布林變數。

```
IF Reset1 = TRUE THEN
    Reset1 := FALSE;
    TMP_BOOL := Counter_Reset (Port, Addr, IO_Name, Channel, Value);
END_IF;
```

LD 語法:

"Reset1" (BOOL): 設定為 "TRUE" 重置計數值，並將 "Reset1" 重置為 "FALSE"。



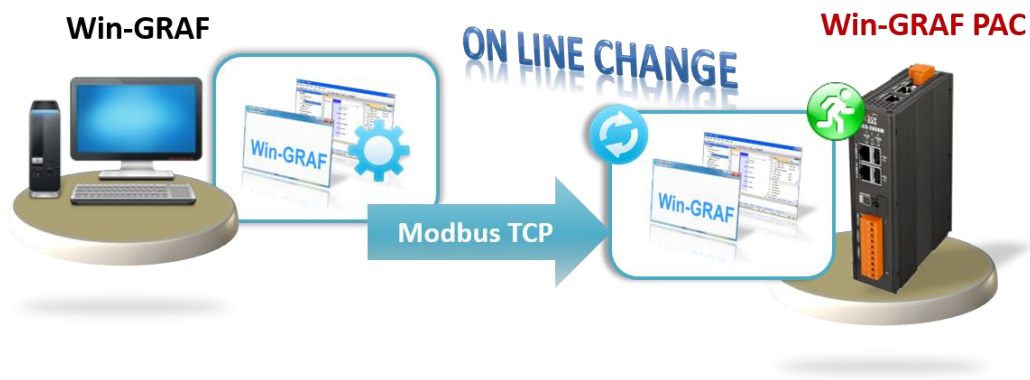
輸入參數	資料型態	說明
Port	DINT	COM Port 編號，視 PAC 而定可設定為 1 ~ 37。
Addr	DINT	模組的 Net-ID 位址，可設定為 1 ~ 255。
IO_Name	DINT	Counter/Encoder 模組的名稱，可設定為 "87084"、"87082"、"7083" 與 "7080"。
Channel	DINT	Counter/Encoder 模組的通道編號，可設定為 "0"、"1"、...。
Value	"DINT", "UDINT", "DWORD" 或 "LINT"	指定一個新的 Counter/Encoder 值。

輸出參數	資料型態	說明
Q	BOOL	"TRUE": OK ; "FALSE": 錯誤。

第 9 章 即時線上更新 (On-Line Change)

"On-Line Change" 功能允許 Win-GRAF PAC 在運行時更新小幅修改過的程式。此程式和 PAC 正在 Run 的程式必需是同一個，且檔名相同。

此功能主要提供緊急狀況使用，像是某些情況下系統需 24 小時運作，不希望 (或不能) 為了更新程式而停止運作。若非此狀況，建議不要進行線上修改。您可先停止運行中的程式，再將修改過的程式下載到 PAC 中，這是比較安全的作法。



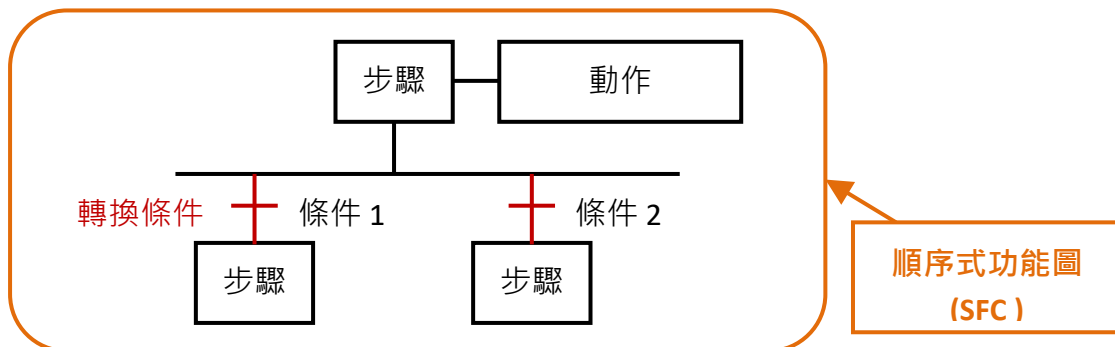
9.1 "On-Line Change" 功能的使用限制



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，可在線執行下列變更：

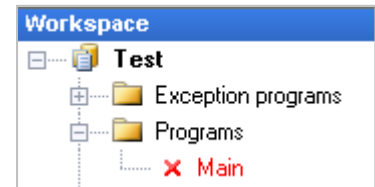
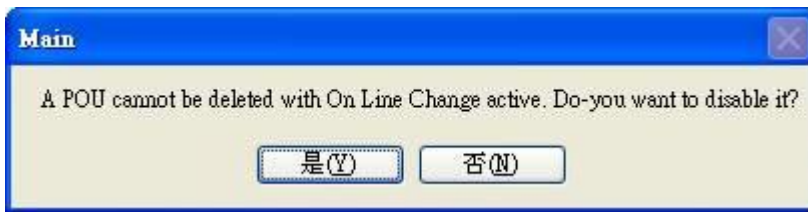
- 變更一個程式中的程式碼。
- 變更一個 SFC 程式中的條件 或 一個步驟的動作。



- 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) 的 FB Instance。

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 FB Instance 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，在變更過程中，不應使用以下**不安全的編程功能**:

- 脈波 (P 或 N) 接點 與 線圈 (邊緣偵測)。
- ✍ 請改成使用已宣告的 "R_TRIG" 與 "F_TRIG" FB Instance。

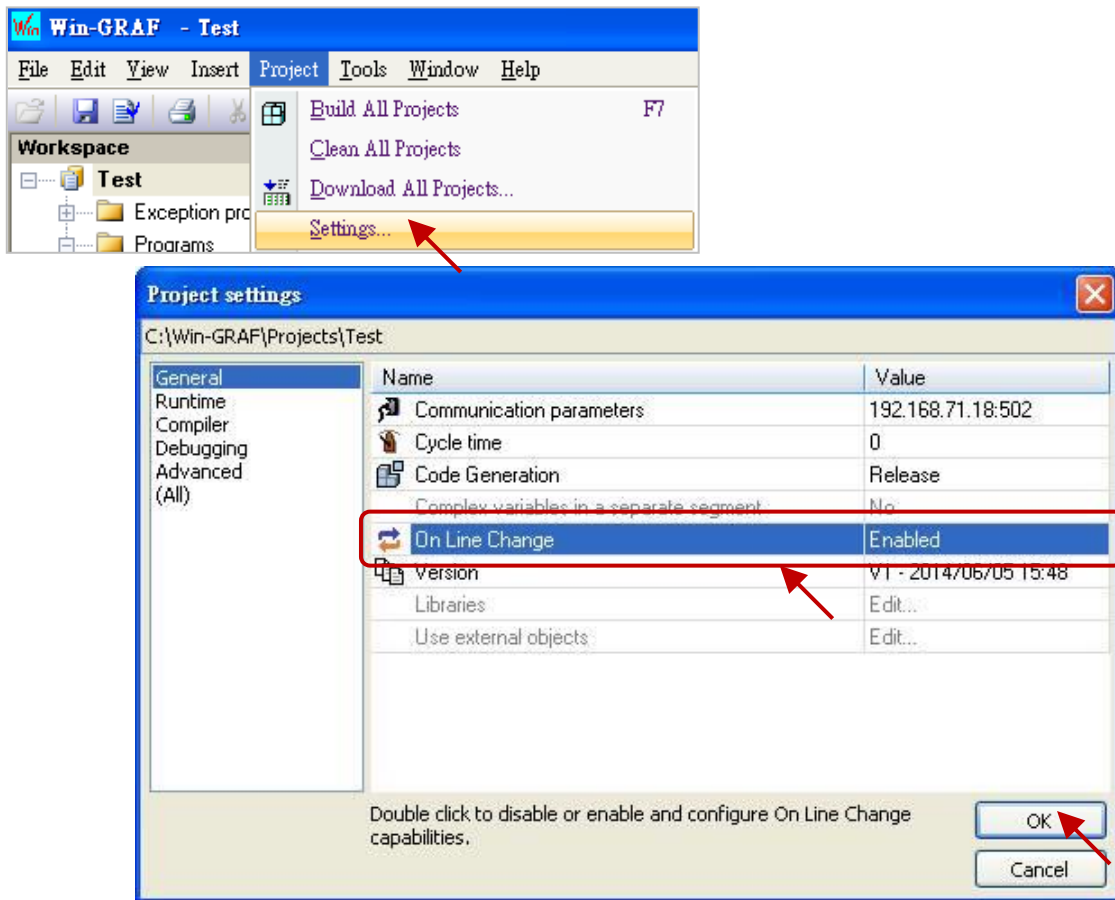
上升脈波偵測		
	不安全	安全
P (False 至 True)		
下降脈波偵測		
N (True 至 False)		

- FBD 的迴圈中含有未定義的變數連結。
- ✍ 您必需在迴圈中明確地指定變數。

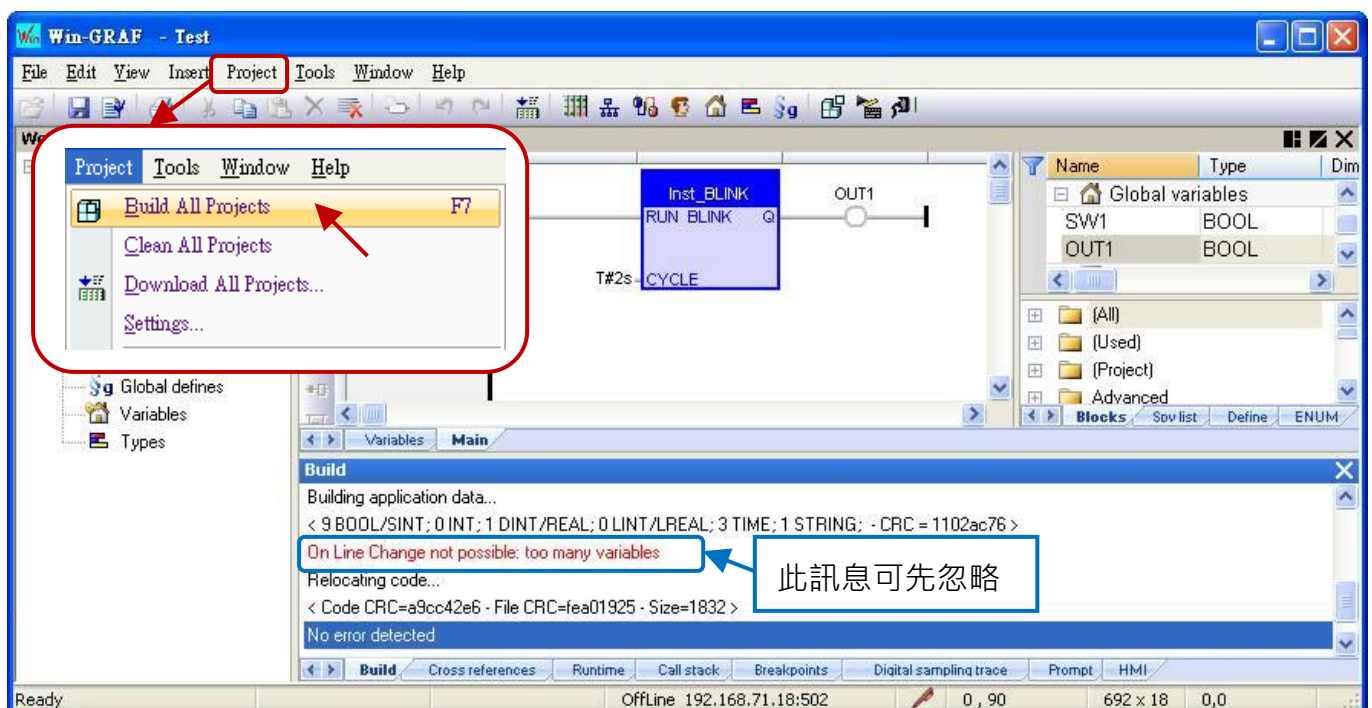
9.2 使用 "On-Line Change" 功能

啟用 "On-Line Change" 功能:

1. 點選功能表 "Project - Settings..."，再雙擊 "On-Line Change" 將其設定為 "Enabled" 並點選 OK。



2. 點選功能表 "Project - Build All Projects" 編譯程式，同時更新 On-Line Change 顯示的變數數量。



分配變數的可使用量:

啟用 "On-Line Change" 功能後，為了允許宣告新變數或 POU (副程式 或 UDFB 的名稱)，必需定義 PAC 中為各種資料型態分配的記憶體空間。

- 依照步驟 1，開啟 "On-Line Change" 視窗。如下圖，請點選所需的變數類型，並設定 "Value" 或 "Margin" 值來配數量，再按 Set 按鈕。

Allocate 設定	說明
Value	總數量 (如左圖)。 點選 BOOL 變數並設為 30，按 Set 後，已分配數量 =30。
Margin	已使用數量 + "已使用" *百分比。例如， 點選 BOOL 變數並設為 100%，按 Set 後，已分配數量 = 9+9*100% = 18。

注意:

- 若 Value 與 Margin 皆有設定值，會加上數量較大者。(如右圖)

此例，Margin (100%=9) 的數量小於 Value (30)，按 Set 後，已分配數量 = 9 + 30 = 39。

已使用/已分配數量

此三項需設定較大的值。

+30 大於 +9，已分配 = 9+30=39。

- 即使是空的專案，預設有 "已使用" 的數量。
- "STRING buffers (characters)"、"FB instance data (bytes – approx.)" 與 "Complex variables segment (bytes)"，此三項需設定較大的值 (此例為 "5000")。

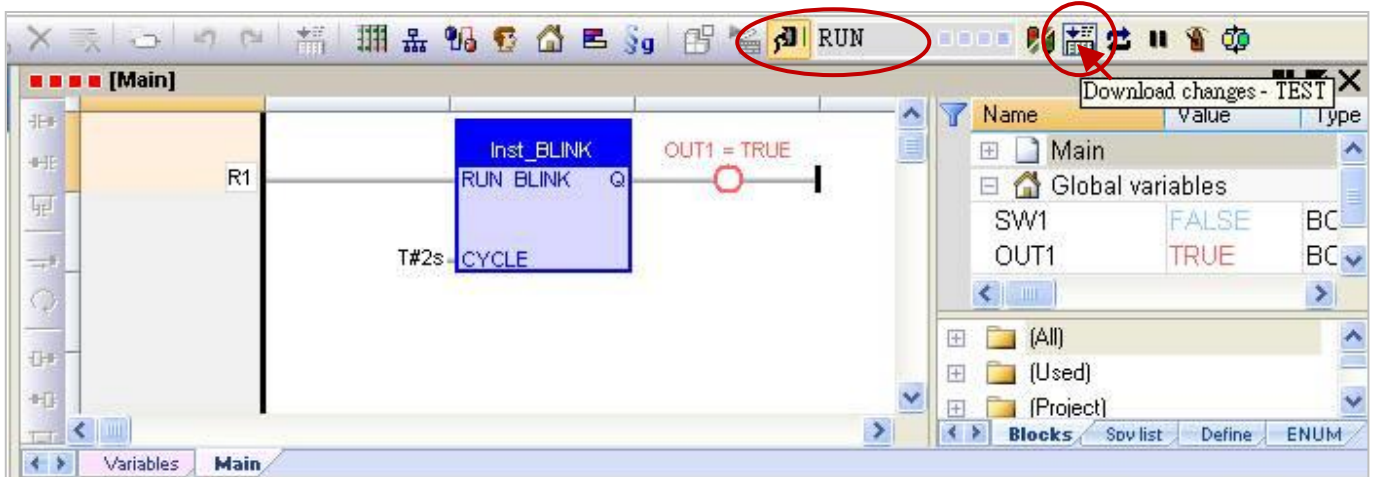
設定完成後，點選右上角的 "X" 按鈕離開設定畫面。

4. 點選功能表 "Project - Build All Projects" 再次編譯程式，並選擇 "On Line" 與 PAC 建立連線。



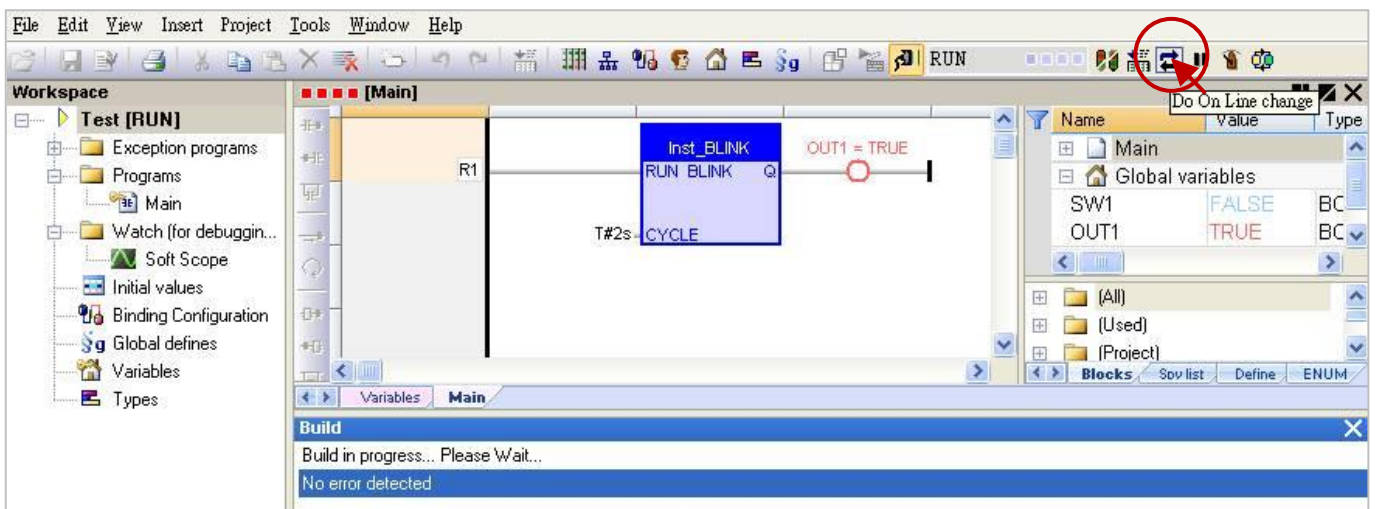
5. 連線成功後，點選 "Download changes" 按鈕，將程式下載到 PAC。

注意： "On-Line Change" 功能，只支援上線中對同一專案進行小幅度的修改，若專案名稱不同，則需停止目前運行的程式，再下載修改過的專案。



6. 點選 "Do On-Line Change" 按鈕，來更新程式。

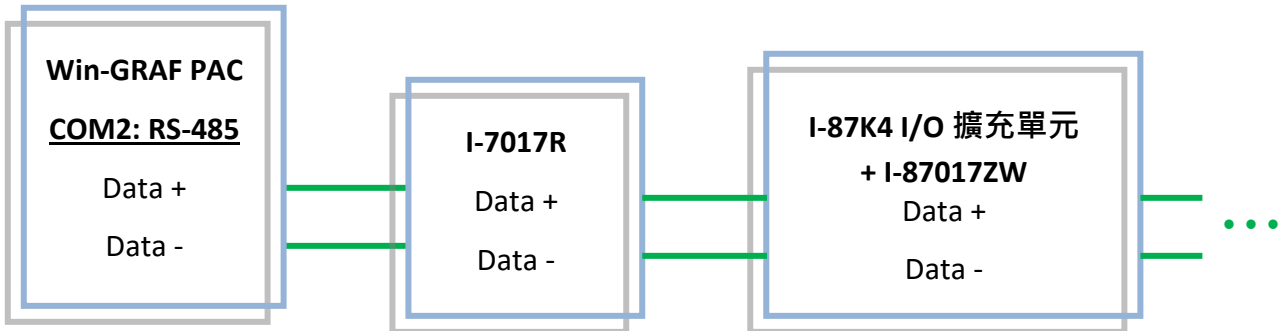
注意： 為了保護系統的正常運作，啟用 "On-Line Change" 後，會有一些使用限制 (參考 9.1 節)，請確認程式無誤後，才執行此功能。



第 10 章 資料/型態轉換 與 使用 PAC 時間

10.1 AI 資料轉換 (i_scale)

連接遠端的 AI 模組 (例如: 透過 PAC 的 RS-485 Port 去連接的 I-87017ZW 或 I-7017R 模組) , 請參考以下設定:



1. 首先, 在 "I/O Boards" 視窗中, 加上 "i_scale", 並滑鼠雙擊 "i_scale_x" 開啟屬性視窗。(第 4 章)

注意: Win-GRAF 專案中只能使用一個 "i_scale" 功能。

2. 在指定的功能編號上設定數值。

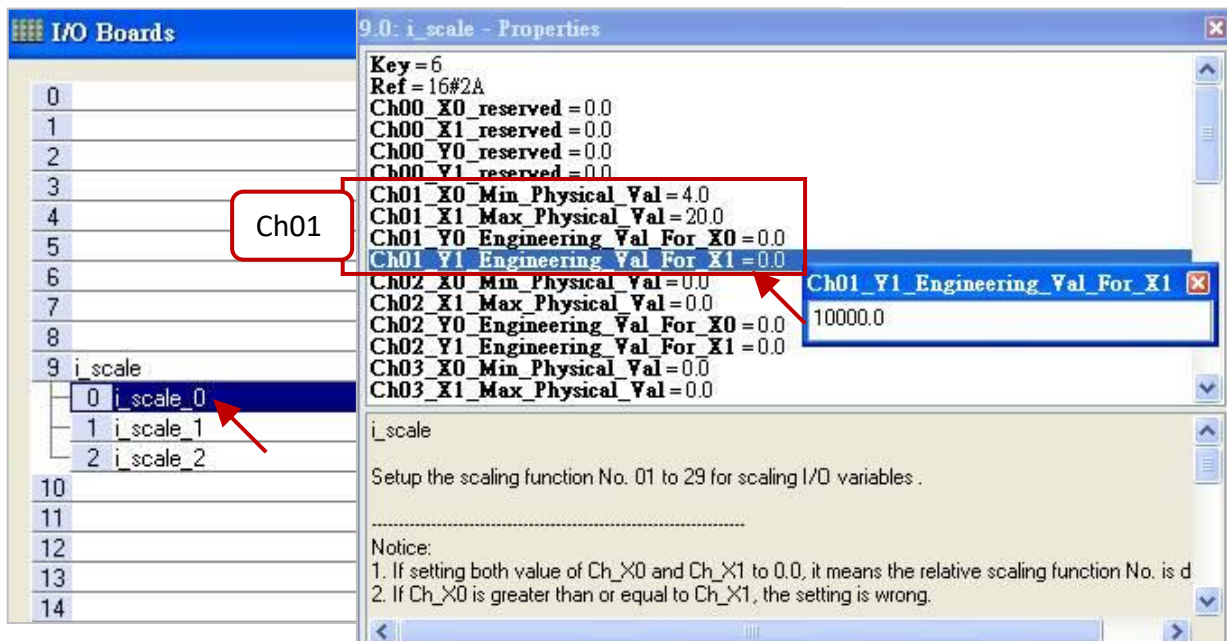
例如: 使用功能編號 01 將 "4 ~ 20 mA" 的電流輸入值轉換為 "0 ~ 10000" 的工程值。

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])。

```

(* 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;
    
```

Name	Type	Dim.	Att
ii	DINT		
Phy_V	REAL	[0..7]	
Eng_V	REAL	[0..7]	
Global variables			
OUT1	BOOL		

(* 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]);

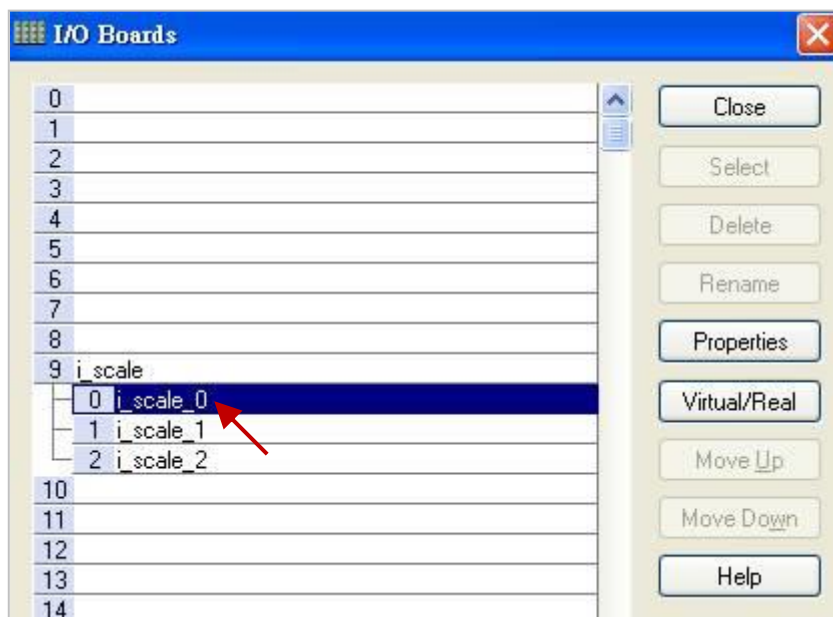
end_for;

10.2 AO 資料轉換 (i_scale)

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)，可參考以下設定:

1. 首先，在 "I/O Boards" 視窗中，加上 "i_scale"，並滑鼠雙擊 "i_scale_x" 開啟屬性視窗。(第 4 章)

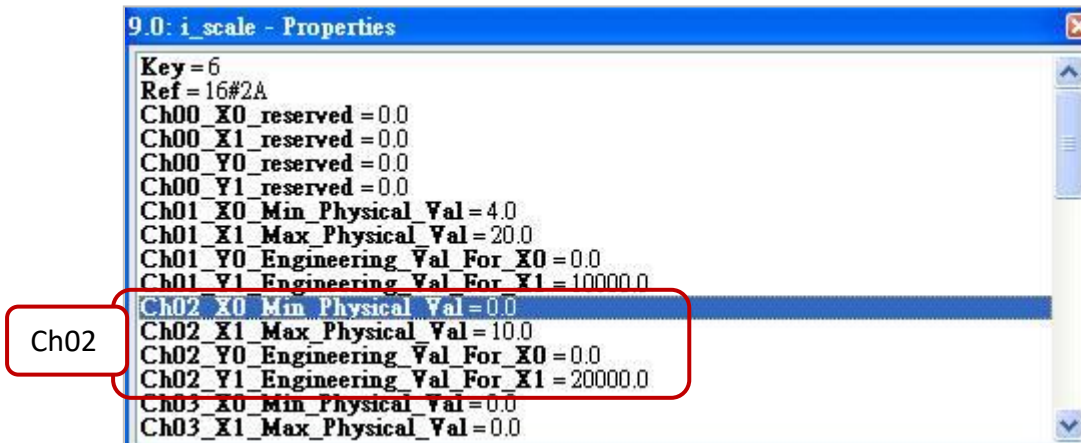
注意: Win-GRAF 專案中只能使用一個 "i_scale" 功能。



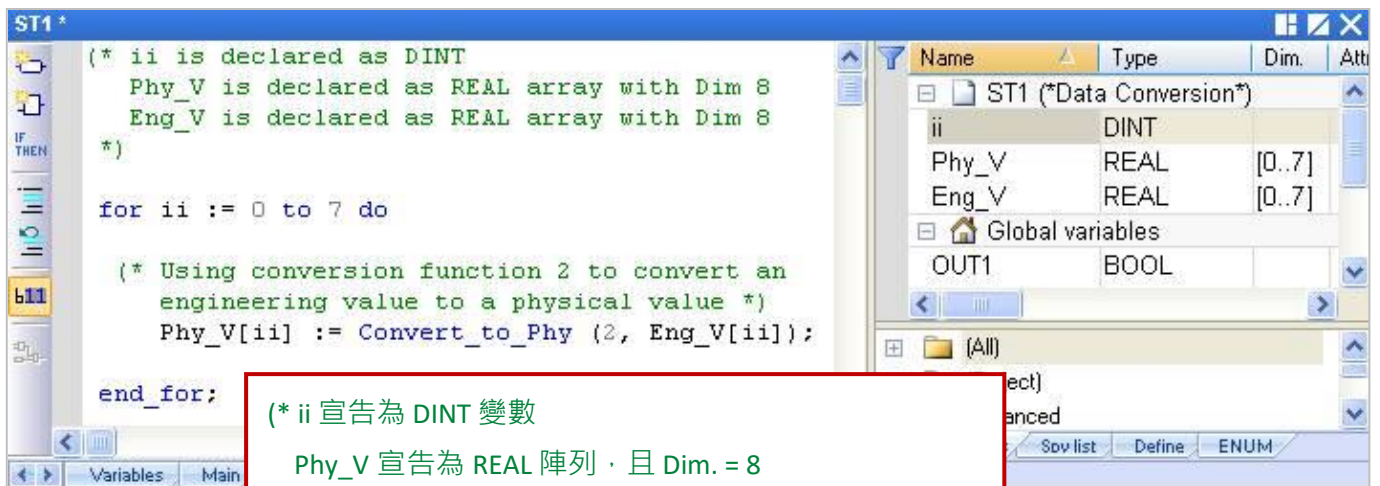
2. 在指定的功能編號上設定數值。

例如: 使用功能編號 02 將 "0 ~ 20000" 的工程值轉換為 "0 ~ 10 V" 的電壓輸出值。

Ch02_X0_Min_Physical_Val: "0.0"
Ch02_X1_Max_Physical_Val: "10.0"
Ch02_Y0_Engineering_Val_For_X0: "0.0"
Ch02_Y1_Engineering_Val_For_X1: "20000.0"



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])。



(* 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 2，將工程值轉換為實體值 *)
 Phy_V[ii] := Convert_to_Phys (2, Eng_V[ii]);

end_for;

10.3 型態轉換函式 (ANY_TO_xxx)

在程式中對不同型態的資料或變數進行比較 (>, <, =, <=, >=, <>) 或運算 (+, -, *, /) 時，請參考下表將其轉換為相同的資料型態，才能正常執行。

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

例如，此 ST 程式用來將資料型態 DINT 轉換為 REAL 後，再進行運算。

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

您可在功能表中開啟 "HTML 說明"，並輸入關鍵字，來搜尋相關的說明。



10.4 BCD 轉換 (BIN_TO_BCD, BCD_TO_BIN)

BCD 碼是用 4 個 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 $2^7 + 2^2 = 128 + 4 = 132$)。

註: BCD 碼 只能用來表示數字 0~9，無法使用數值 1010、1011、1100、1101、1110、1111，會回傳 "0"。

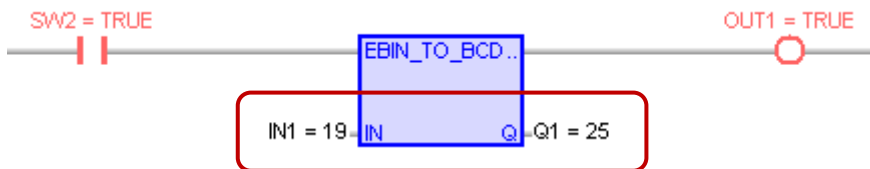
十進制	BCD				說明
	2^3	2^2	2^1	2^0	
0	0	0	0	0	0
1	0	0	0	1	$2^0 = 1$
2	0	0	1	0	$2^1 = 2$
3	0	0	1	1	$2^1 + 2^0 = 3$
4	0	1	0	0	$2^2 = 4$
5	0	1	0	1	$2^2 + 2^0 = 5$
6	0	1	1	0	$2^2 + 2^1 = 6$
7	0	1	1	1	$2^2 + 2^1 + 2^0 = 7$
8	1	0	0	0	$2^3 = 8$
9	1	0	0	1	$2^3 + 2^0 = 9$

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: IN1, IN2, Q1, Q2 宣告為 DINT，若 IN ≤ 0，Q 會顯示 "0"。

BIN_TO_BCD: $19_{(10)} = 0001\ 1001_{(BCD)} = 2^4 + 2^3 + 2^0 = 16 + 8 + 1 = 25$



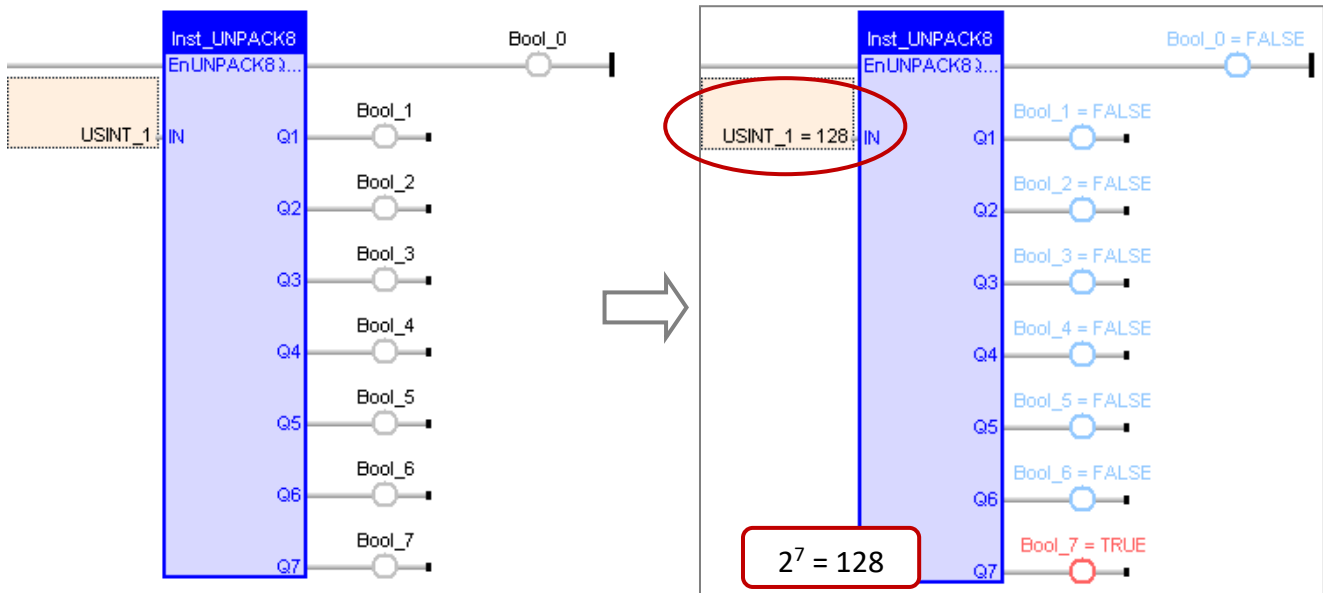
BCD_TO_BIN: $33_{(10)} = 0010\ 0001_{(2)} = 21_{(BCD)}$
 $15_{(10)} = 0000\ 1111_{(2)} = 0_{(BCD)}$



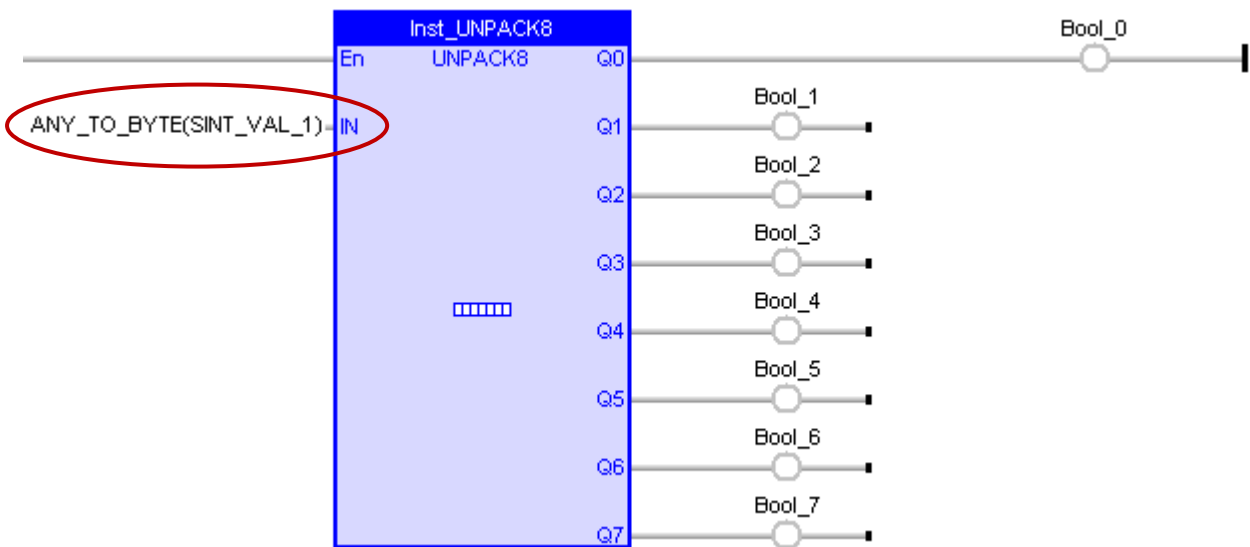
10.5 組合/拆解 整數或布林值

將整數值拆解為布林值:

"UNPACK8" 功能可用來將 1 個 BYTE (或 USINT, 範圍: 0 ~ 255) 拆解為 8 個 Boolean。



若想把 1 個 SINT 拆解為 8 個 Boolean, 可使用 ST 語法 "ANY_TO_BYTE()" 先將 SINT 轉為 BYTE 型態, 方法如下:



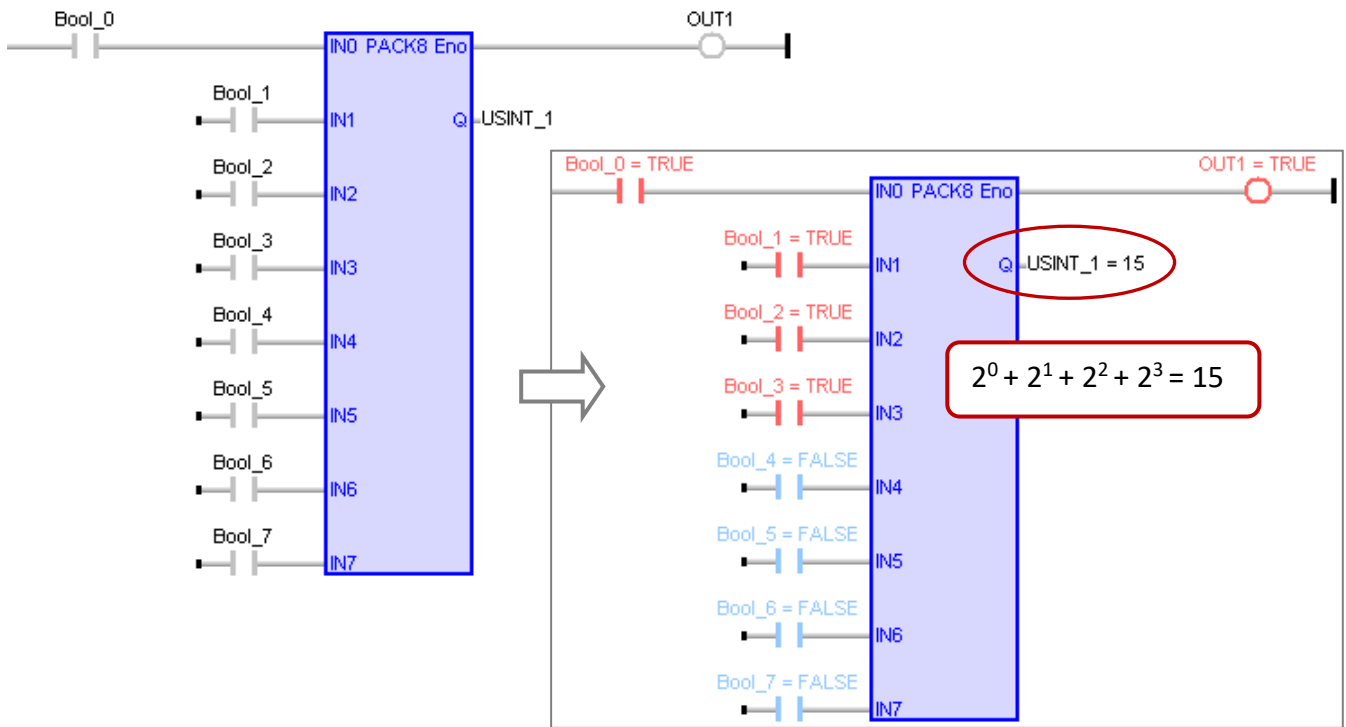
將布林值組合為整數值:

"PACK8" 功能可用來將 8 個 Boolean 組合成 1 個 BYTE (或 USINT, 範圍: 0 ~ 255)。

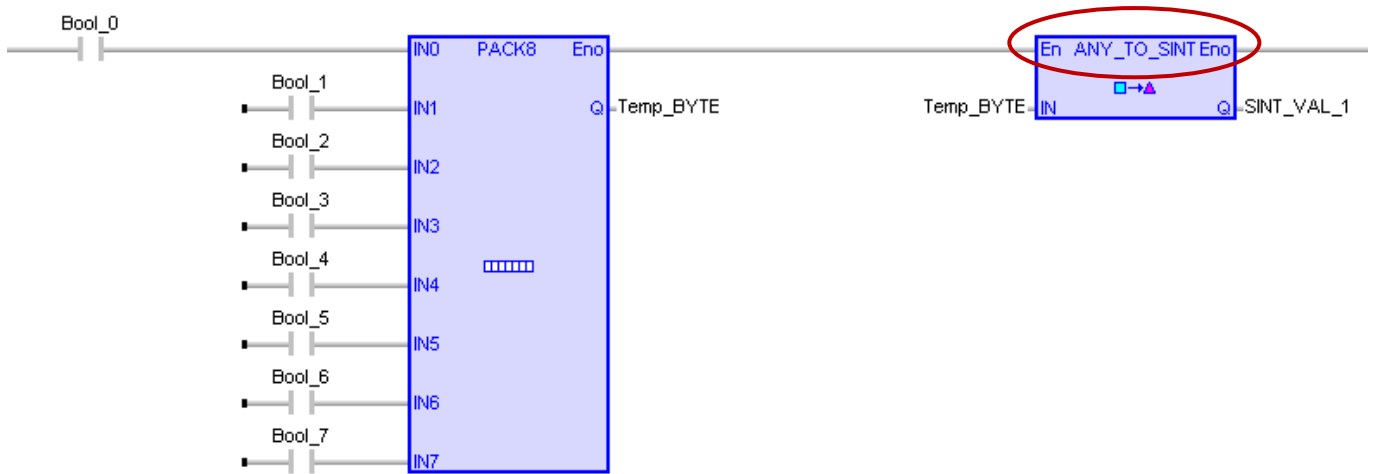
ST 語法:

```
USINT_1 := PACK8 (Bool_0, Bool_1, Bool_2, Bool_3, Bool_4, Bool_5, Bool_6, Bool_7);
```

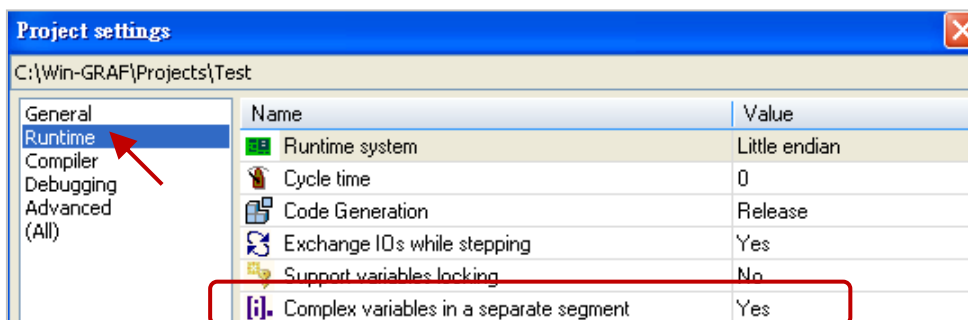
LD 語法:



若想把 8 個 Boolean 組合成 1 個 SINT，需在輸出 (Q) 指定一個 "BYTE" 變數來暫存數值，並加入一個 "ANY_TO_SINT" 功能將 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

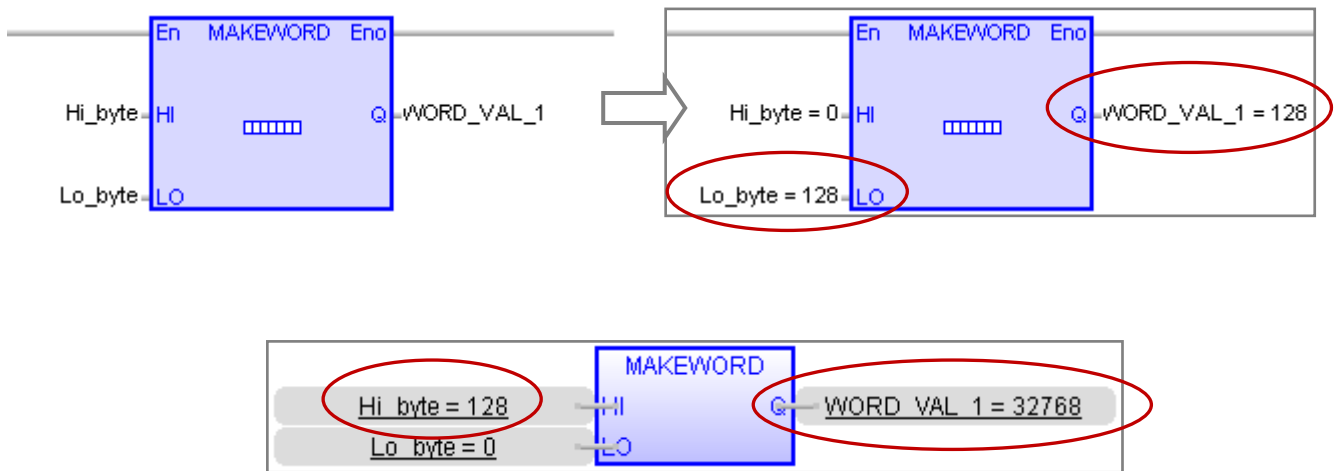
將 2 個 8-bit 組成 16-bit 資料

"MAKEWORD" 功能可用來將 2 個 Byte (或 USINT) 組成 1 個 WORD (或 UINT)。

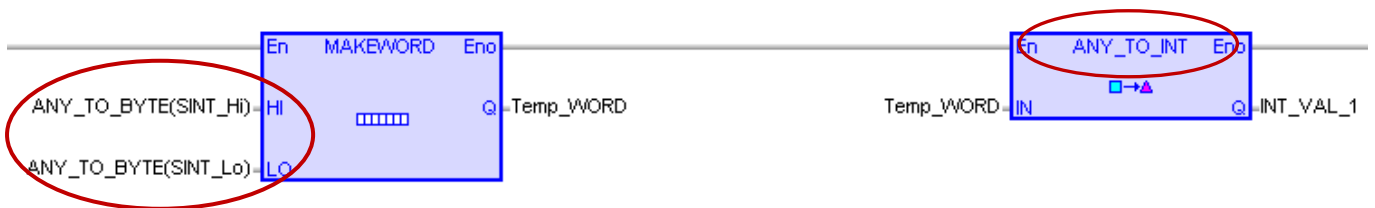
ST 語法:

```
WORD_VAL_1 := MAKEWORD (Hi_byte, Lo_byte);
```

LD/FBD 語法:



若想將 2 個 SINT 組成 1 個 INT，可使用 ST 語法 "ANY_TO_BYTE()" 先將 SINT 轉為 BYTE，再使用一個 "ANY_TO_INT" 功能，將組成的 WORD 轉為 INT 型態。



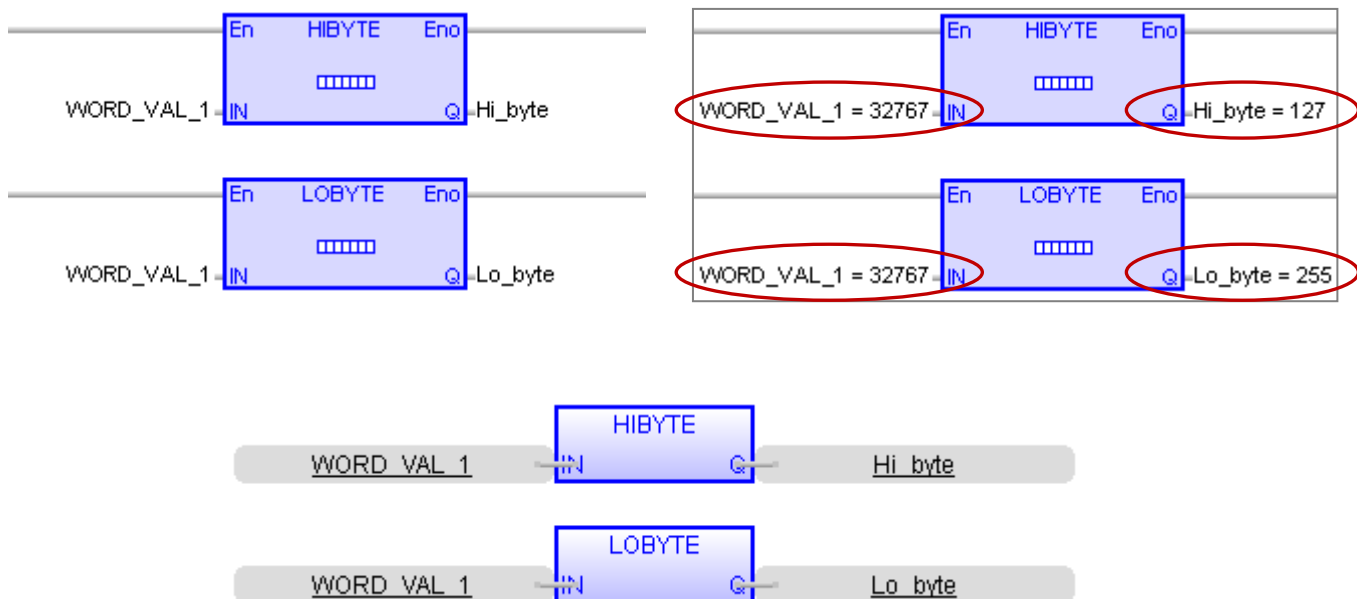
將 1 個 16-bit 拆解為 2 個 8-bit 資料

"HIBYTE" 與 "LOBYTE" 功能可用來將 1 個 WORD (或 UINT) 拆解為 2 個 Byte (或 USINT)。

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，可使用 ST 語法 "ANY_TO_WORD()" 先將 INT 轉為 WORD，再使用一個 "ANY_TO_SINT" 功能將拆解的 BYTE 轉為 SINT 型態。



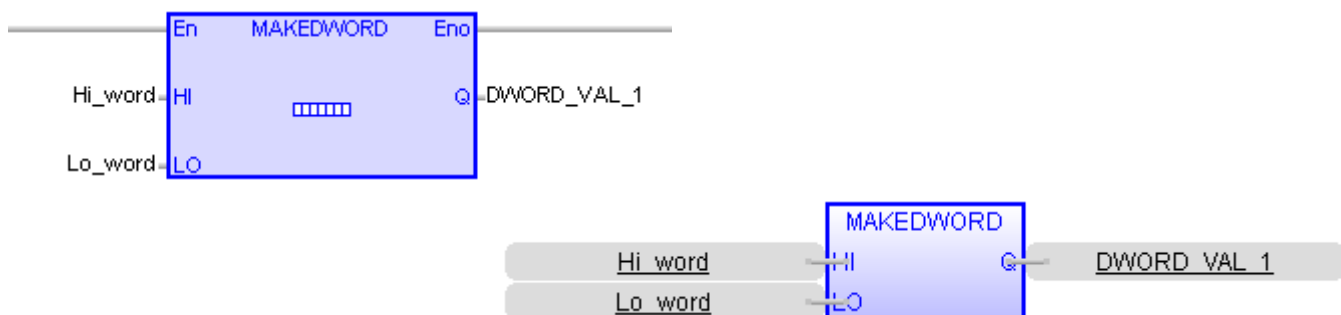
將 2 個 16-bit 組成 32-bit 資料

"MAKEDWORD" 功能可用來將 2 個 WORD (或 UINT) 組成 1 個 DWORD (或 UDINT)。

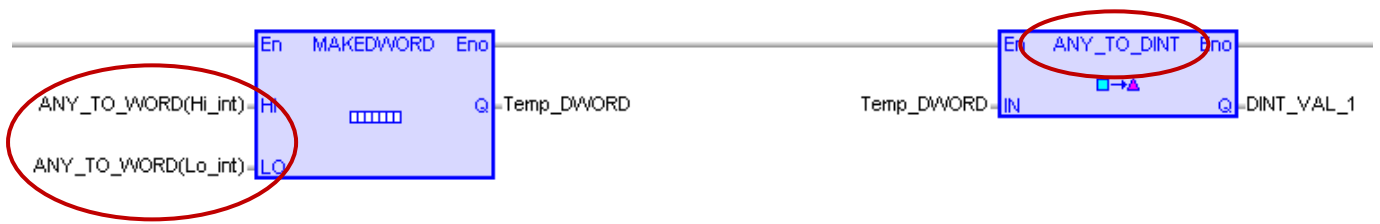
ST 語法:

```
DWORD_VAL_1 := MAKEDWORD (Hi_word, Lo_word);
```

LD/FBD 語法:



若想將 **2 個 INT** 組成 **1 個 DINT**，可使用 ST 語法 "ANY_TO_WORD()" 先將 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

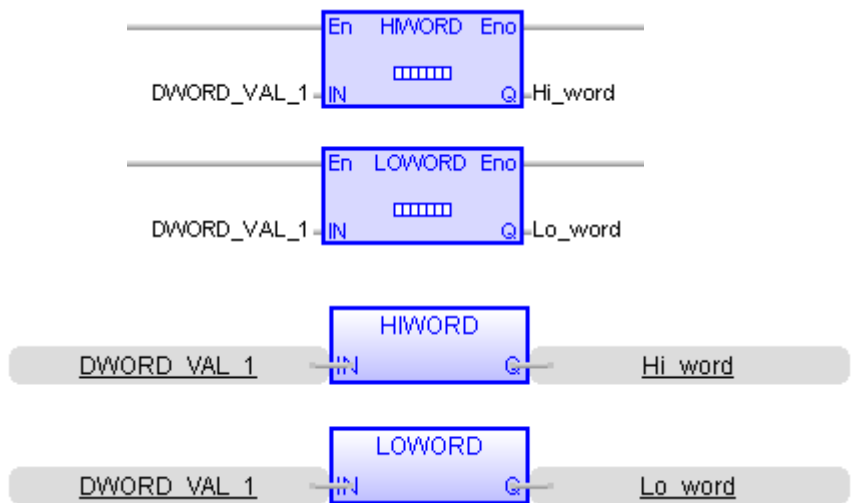
"HIWORD" 與 "LOWORD" 功能可用來將 **1 個 DWORD** (或 UDINT) 拆解為 **2 個 WORD** (或 UINT)。

ST 語法:

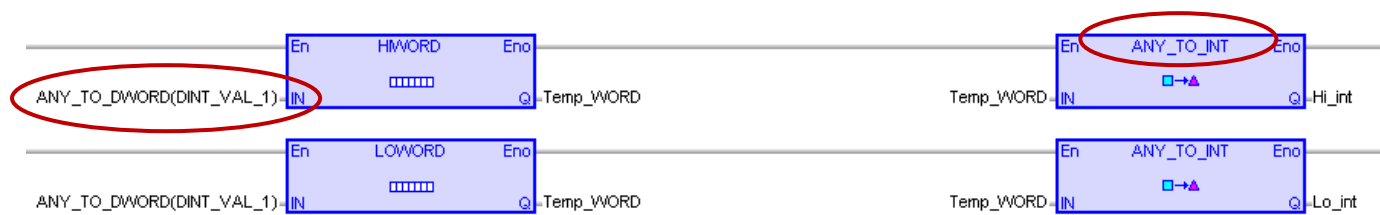
```

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**，可使用 ST 語法 "ANY_TO_DWORD()" 先將 DINT 轉為 DWORD，再使用一個 "ANY_TO_INT" 功能，將拆解的 WORD 轉為 INT 型態。



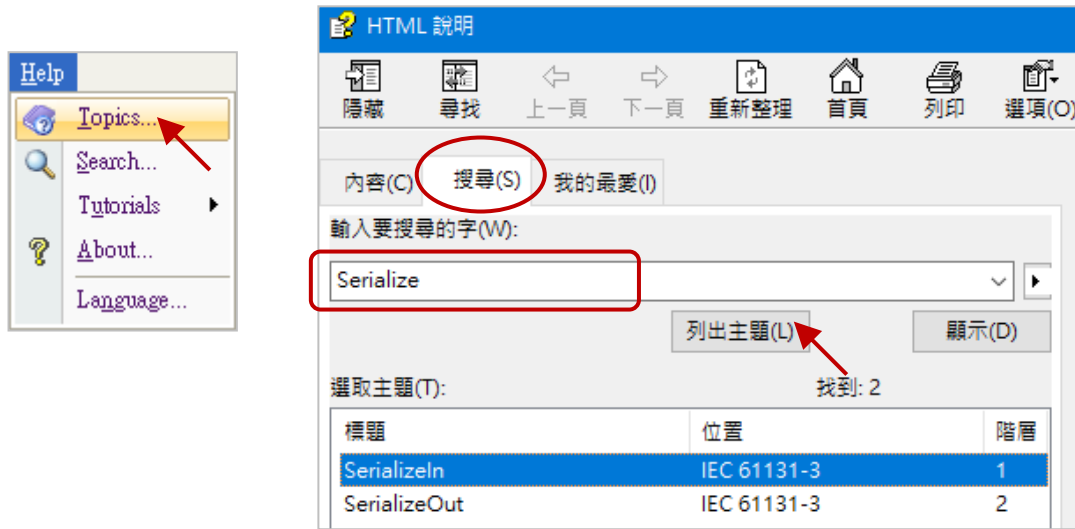
10.7 將變數拆成 Byte Array 或將 Byte Array 組成變數

"SerializeOut" 函式，可將一個 Win-GRAF 變數值拆解成 Byte Array (或 USINT Array)；

"SerializeIn" 函式，可將一個 Byte Array (或 USINT Array) 組成一一個 Win-GRAF 變數值。

- 註:** 1. Array 的 Dim. 至少要設定為 "8"。
2. "Serialize" 函式中，不可使用 STRING 變數。

您可在功能表中開啟 "HTML 說明"，並輸入關鍵字，來搜尋相關的說明。



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)



若要設定 Win-GRAF PAC 目前的時間，可使用 "TIME_SET" 功能方塊。填入新的時間到 new_Year、new_Month、new_Day、new_WeekDay、new_Hour、new_Minute 與 new_Second 變數後，再將 Set_new_time 變數設定為 "TRUE" 一次。

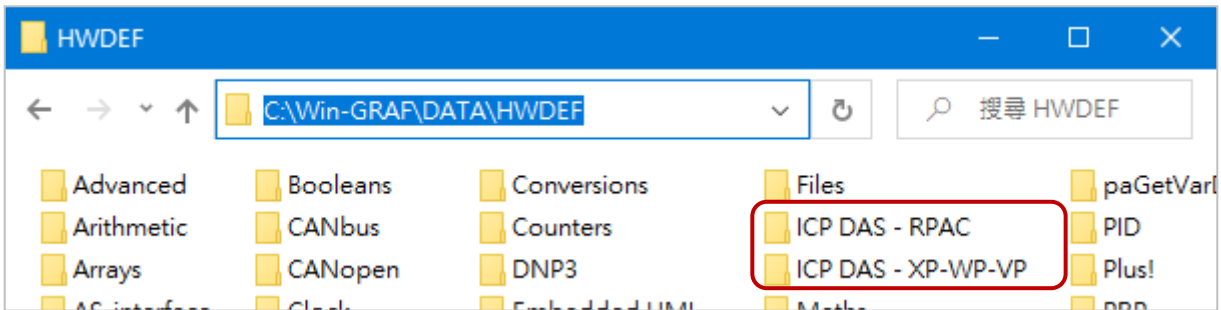


第 11 章 一般常用工具 與 有用的技巧

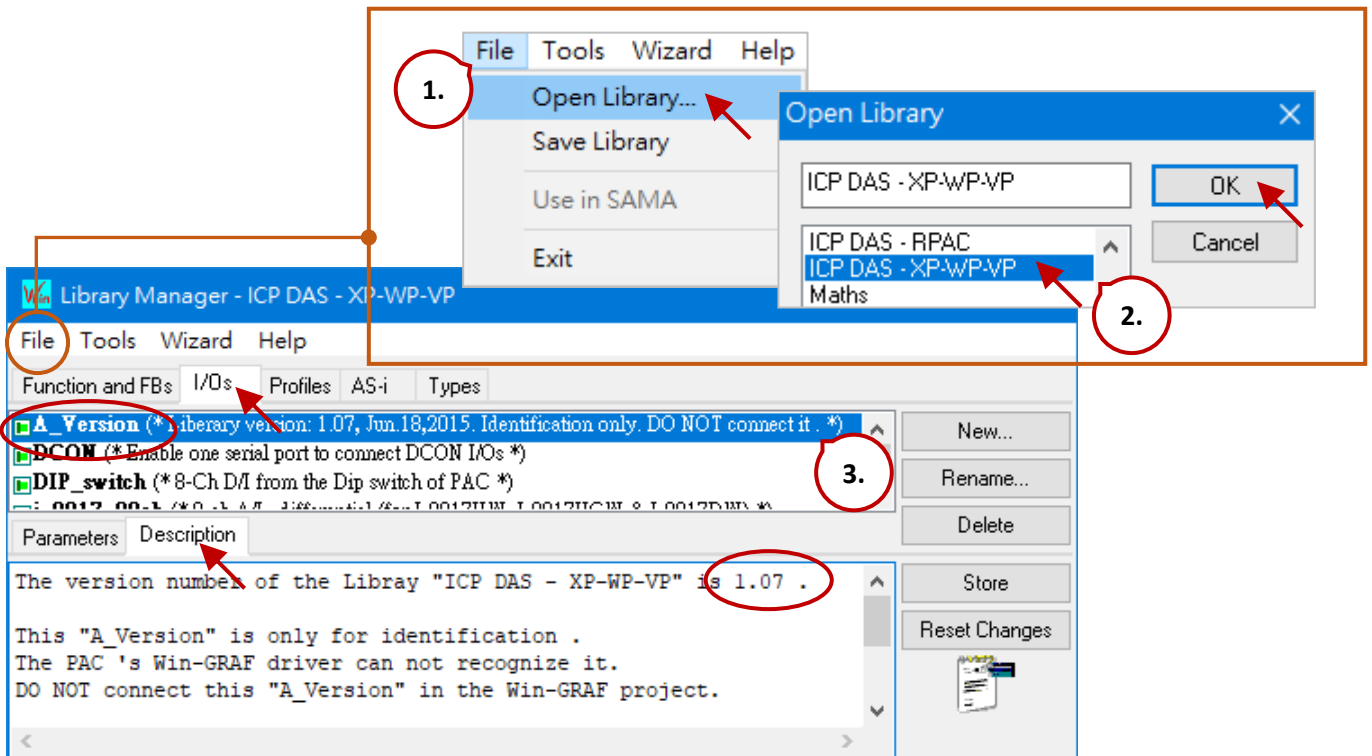
11.1 更新 Win-GRAF 函式庫 (Library)

您可在網站下載最新的 [函式庫](#) (例如: "win-graf-lib-x.xx.zip")。ICP DAS 的 Win-GRAF 的函式庫包括 Function、Function Block...，儲存在 C:\Win-GRAF\DATA\HWDEF 下的 "ICP DAS - XP-WP-VP" 與 "ICP DAS - RPAC" 目錄中。

1. 更新前，請關閉所有 Win-GRAF Workbench 的視窗。建議先備份舊的函式庫到其他目錄 (例如: D:\temp\xxx.zip)，再刪除目前位置的目錄。
2. 將新的 "ICP DAS - XP-WP-VP" 與 "ICP DAS - RPAC" 目錄，複製到 C:\Win-GRAF\DATA\HWDEF 內，再開啟 Win-GRAF Workbench。



註: 若想查詢 Win-GRAF 函式庫的版本號，請在 "Library Manager" (參考 1.2.2 節) 開啟函式庫 - "ICP DAS - XP-WP-VP"，並在 "I/Os" 頁面點選 "A_Version"，再點選 "Description" 查看版本號 (例如: "1.07")。




11.2 更新 Win-GRAF 驅動程式 (Driver)

使用者可在網站下載最新的 [Driver](#)，並依照下列步驟更新韌體。

<https://www.icpdas.com/en/download/index.php?model=RPAC-2658M>

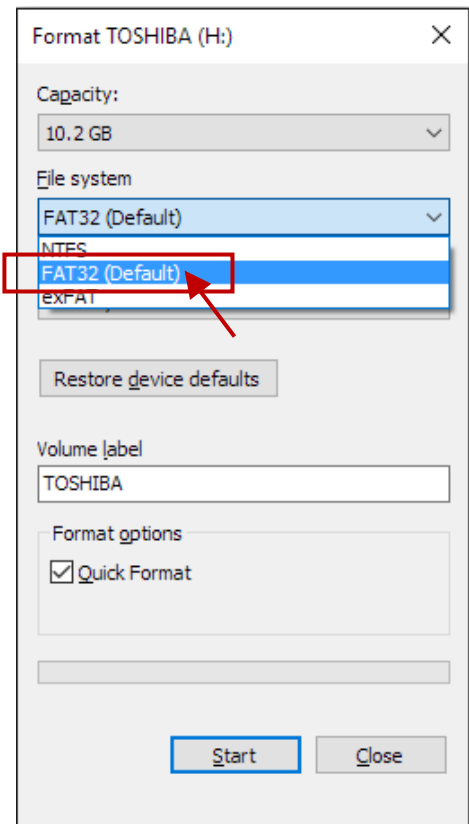
步驟 1:

- 請準備一個 USB 隨身碟。
- 將隨身碟格式化為 **FAT32** 格式。
- 將批次檔放入隨身碟，檔名是 **patch_vxx.xx.zip**。

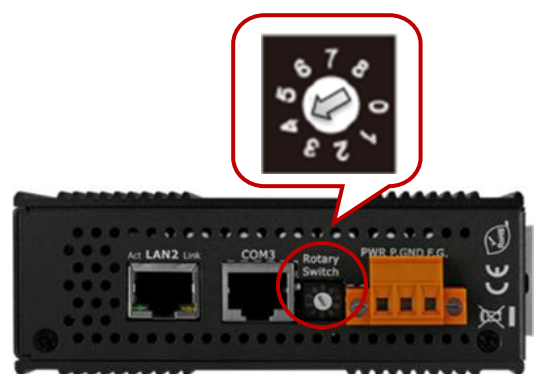
 patch_v0.11.zip

步驟 2:

將隨身碟插入 RPAC-2658M 的 USB Port，並將旋轉開關調到 **4** 的位置。

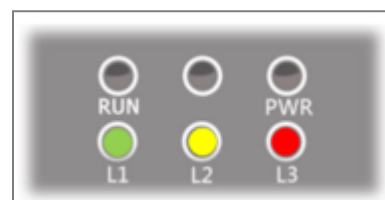


位置 4



步驟 3:

打開電源，將會自動執行韌體更新程序。更新完成後，**L1, L2, L3 LED** 皆為 **ON**。



步驟 4:

關閉電源，並取出您的隨身碟。

步驟 5:

將旋轉開關調到 **0** 的位置，再開啟電源。韌體版本顯示在 Console Port 或網頁上，表示更新成功。

Console port:

請參考 13.2 節，使用 SSH 軟體與 PAC 連線。

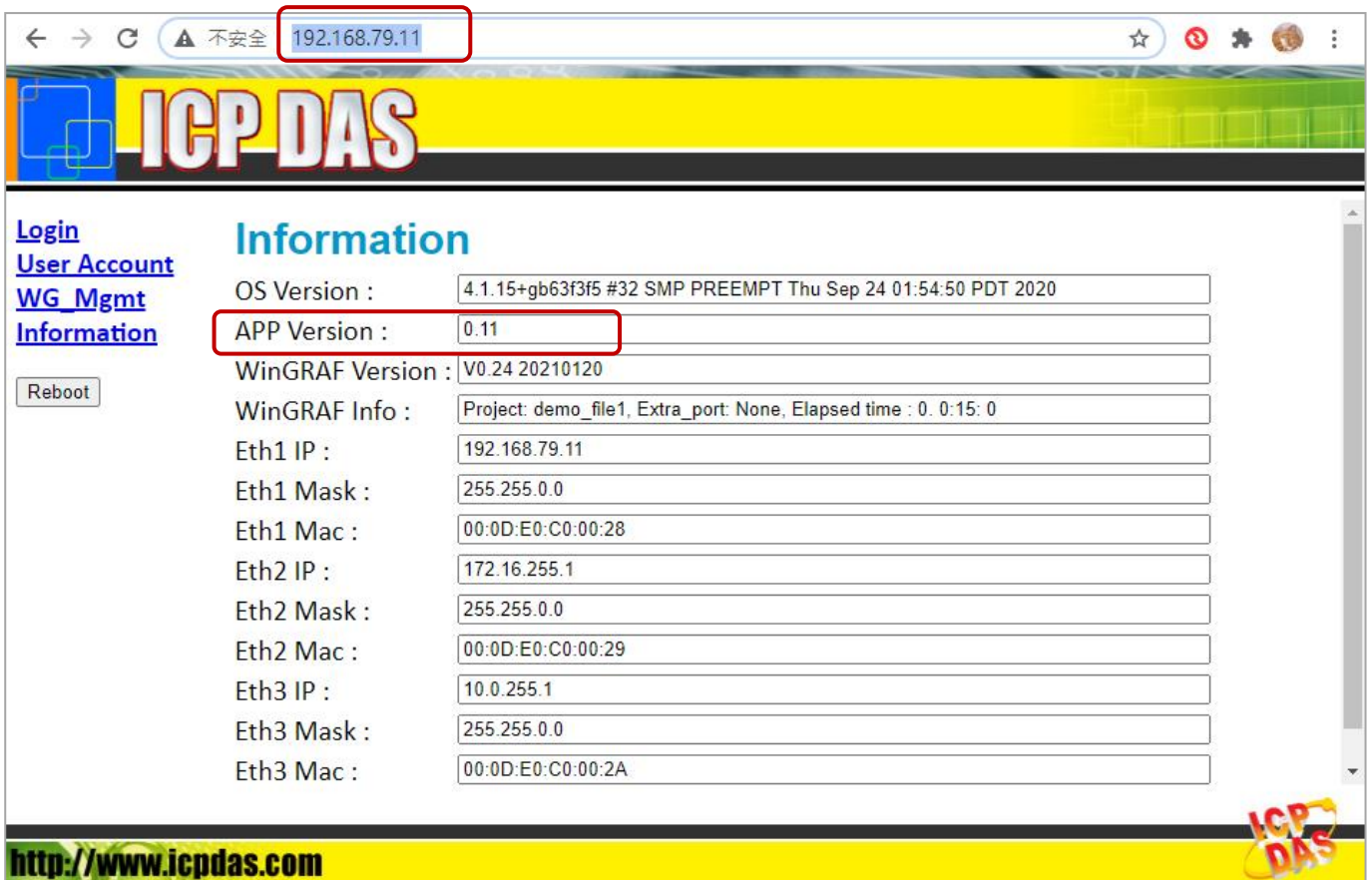
```
Welcome to RPAC-2658M (armv7l-Linux-4.1.15+gb63f3f5@ttymxc1/115200)

Copyright (C) 2021 ICP DAS Corp. <www.icpdas.com>
Last update : Ver=0.11 Time=20210122-22:44:46

RPAC-2658M login: [ ]
```

網頁:

輸入 PAC 的 IP 位址。



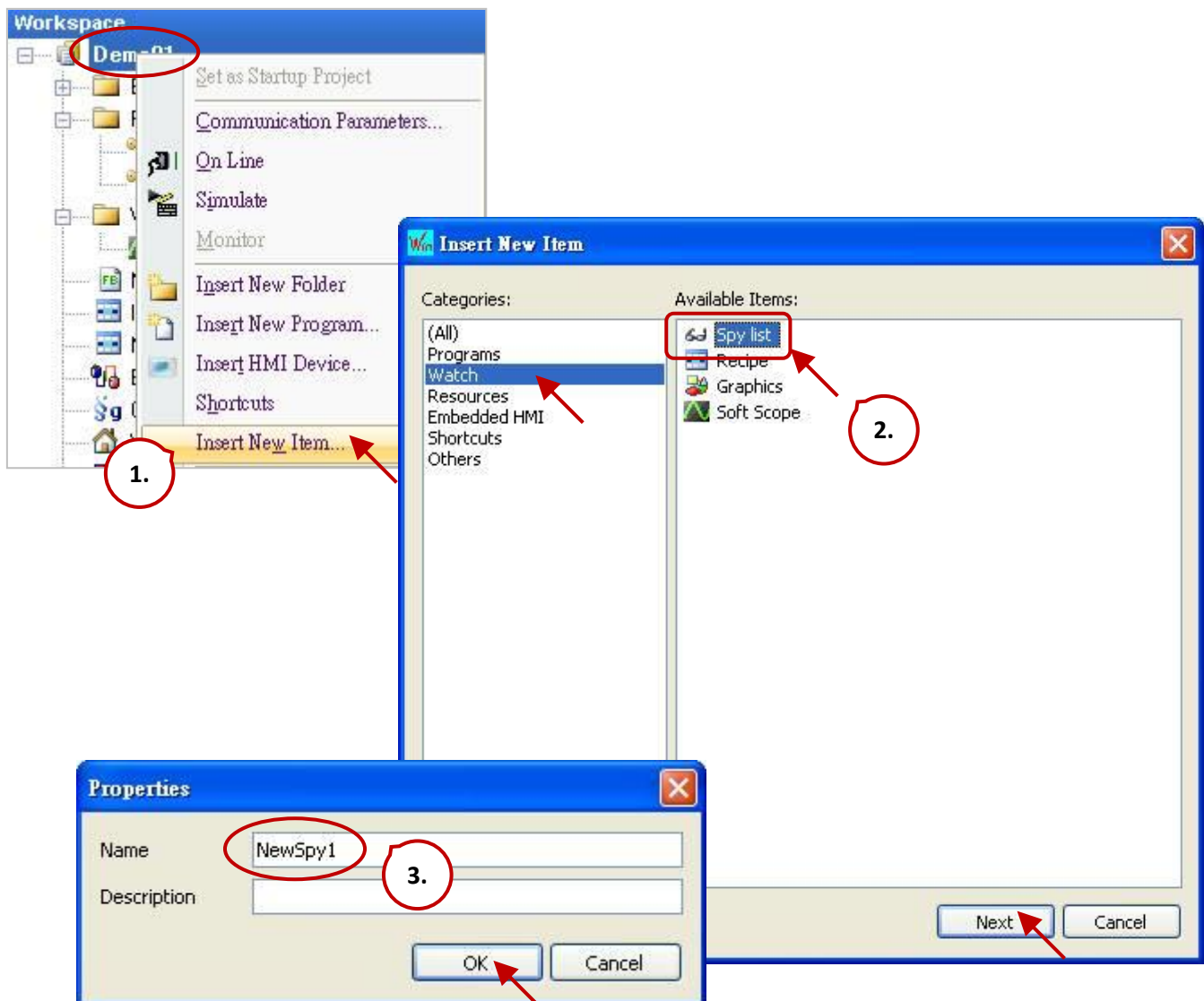
11.3 觀測清單 (Spy List)

Spy List 是一個監測功能，可讓使用者在 PAC 運行時觀察 Win-GRAF 變數。一個應用程式中可能宣告了數百個變數，Workbench 可讓使用者加入指定的變數到 Spy List，並可在運行中監測變數當前的數值與狀態。

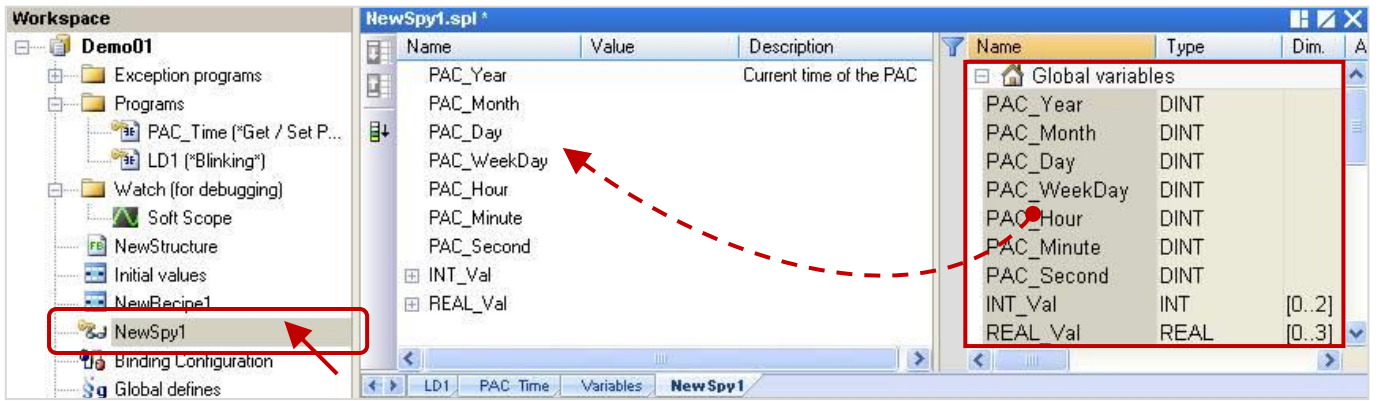
連線模式下，您可在 Spy List 中寫入或設定變數資料，來直接掌控 PAC 的應用動作。

方法如下：

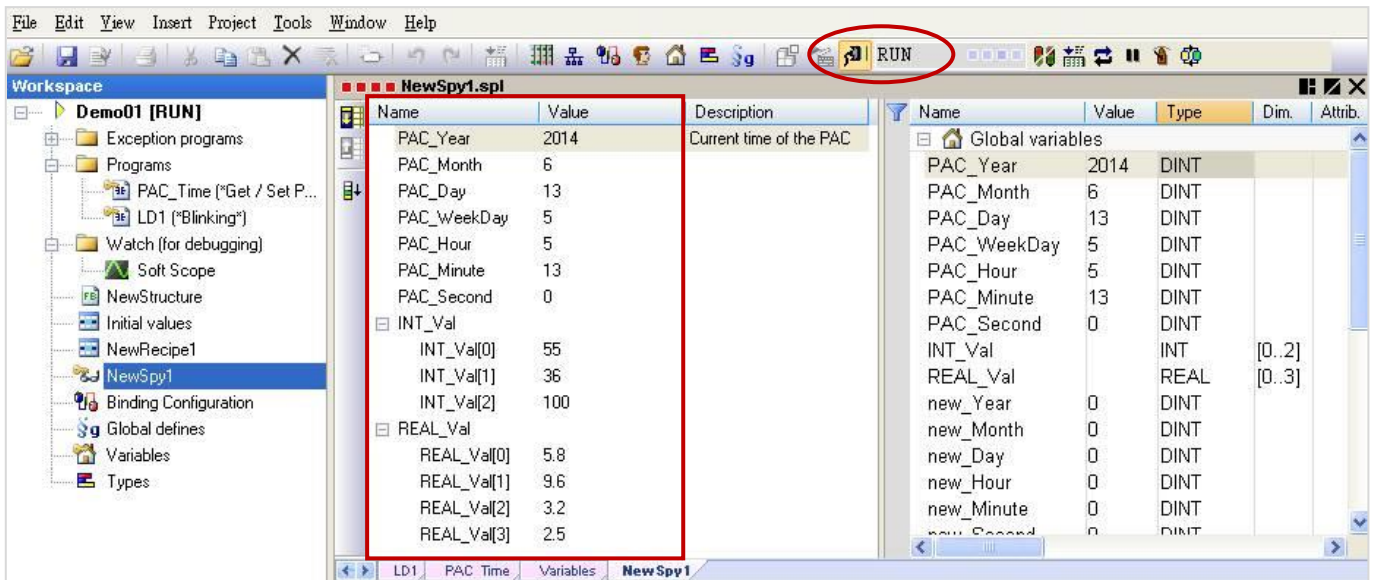
1. 滑鼠右鍵點選專案名稱 (例如: "Demo01") 並選擇 "Insert New Item" 項目。
2. 點選 "Watch" 類別並選擇 "Spy List"，再點選 "Next" 進行下一步驟。
3. 輸入清單名稱 (例如: "NewSpy1") 並按 "OK"。



4. 滑鼠雙擊左側的 "NewSpy1" 開啟視窗，並將想要觀察的變數拖曳到此視窗內。



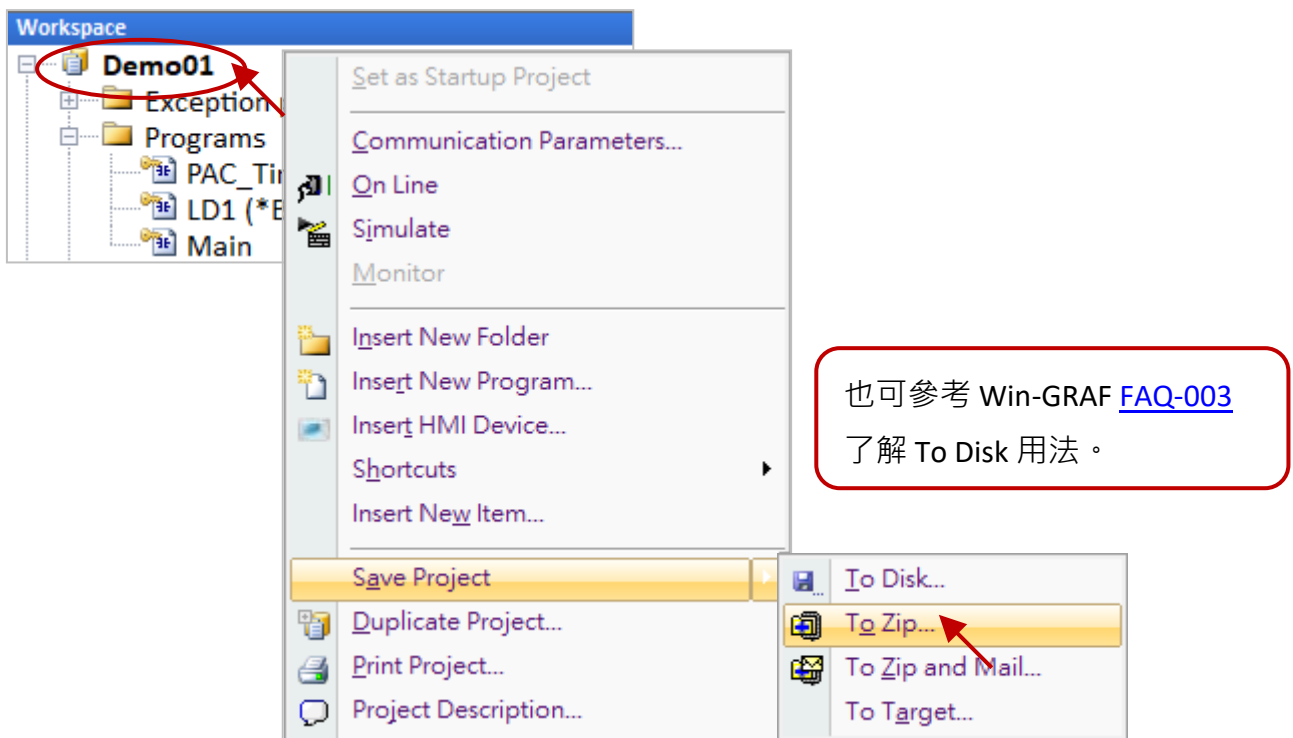
5. 當 Win-GRAP 與 PAC 連線時，將會顯示當前的變數值。



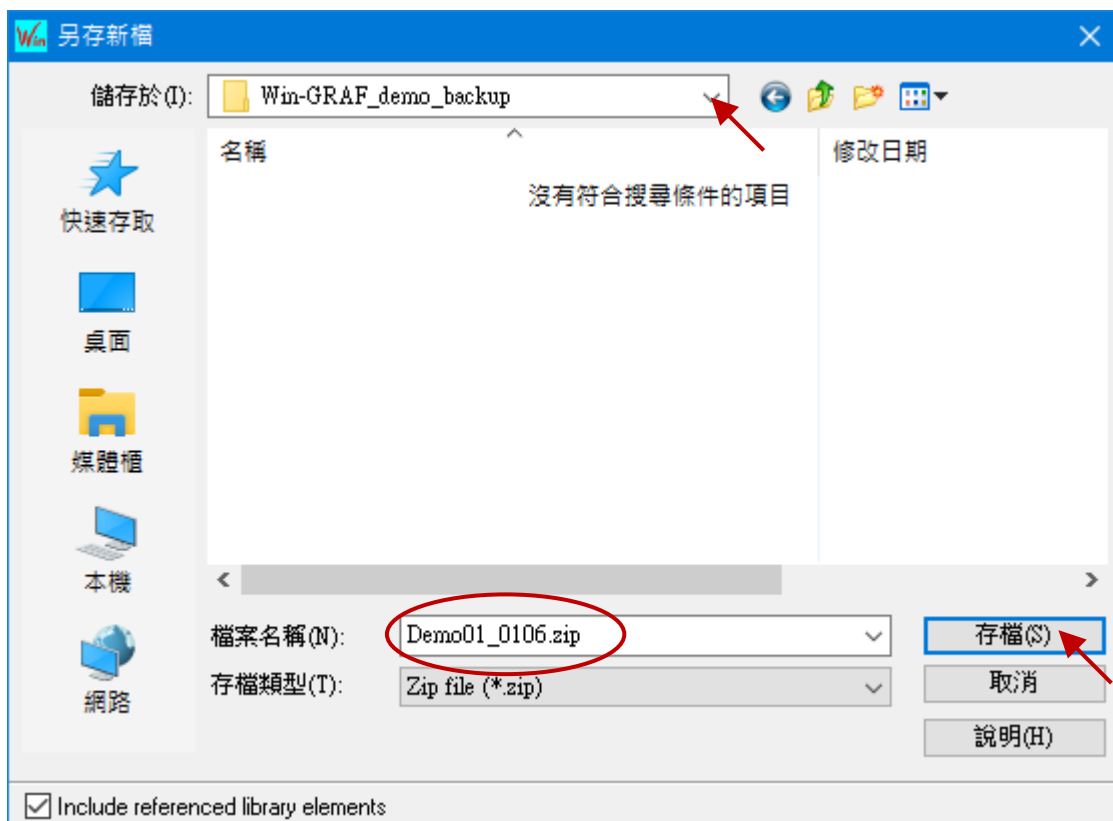
11.4 備份/回存一個 Win-GRAF 專案 (Project)

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 Zip" 。

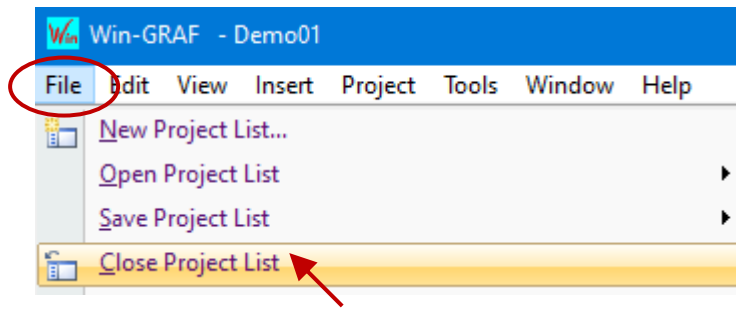


2. 指定要存檔的位置 (例如: D:\Win-GRAF_demo_backup) , 並輸入專案名稱 (例如: "Demo01_0106") , 再點選 "存檔" 即完成備份。

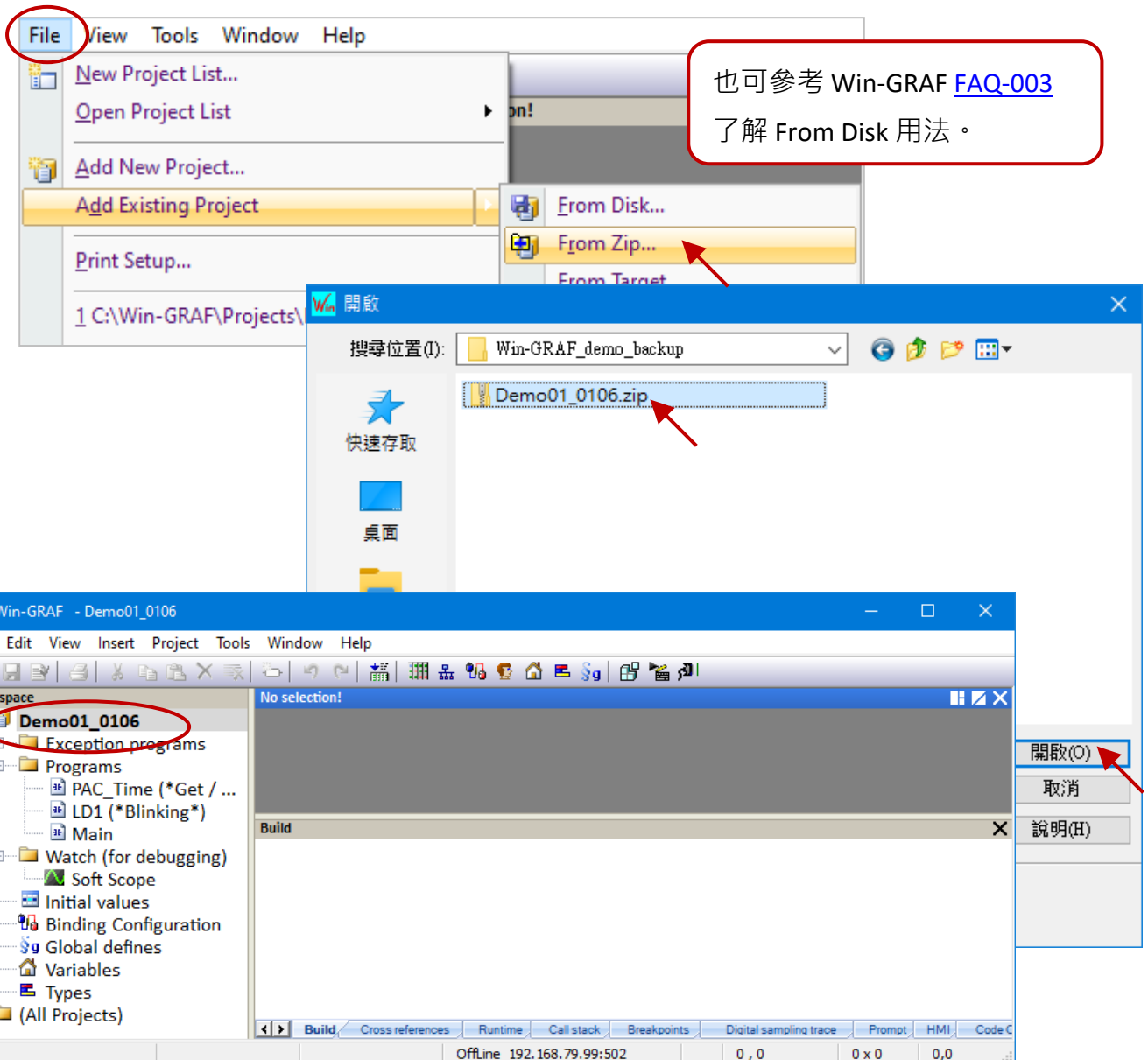


回存 Win-GRAF 專案:

註: 回存前，請點選功能表 "File - Close Project List"，關閉所有的專案。



1. 點選功能表 "File - Add Existing Project - From Zip"，接著選取專案名稱 (例如: "Demo01_0106")，再點選 "開啟" 即完成回存。



11.5 以軟體重新啟動 PAC

在某些情況下，使用者可能需要以軟體方式來重新啟動 PAC。Win-GRAF 提供了 "PAC_Reboot" 函式 (Function) 可調用它來自動重開機。

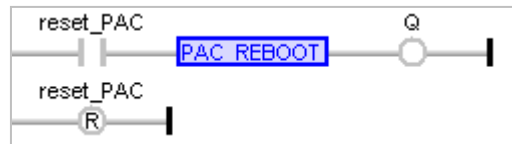
註: 請勿在每個 PAC Cycle 調用此 Function，否則 PAC 會一直重新開機。

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。 *)
```

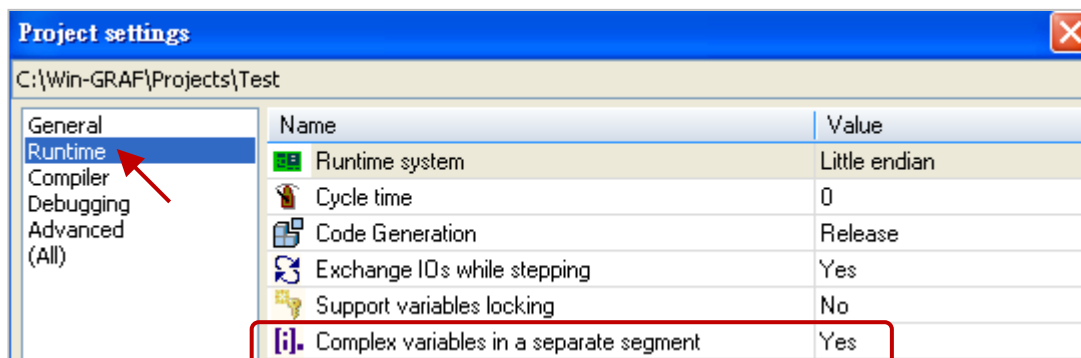
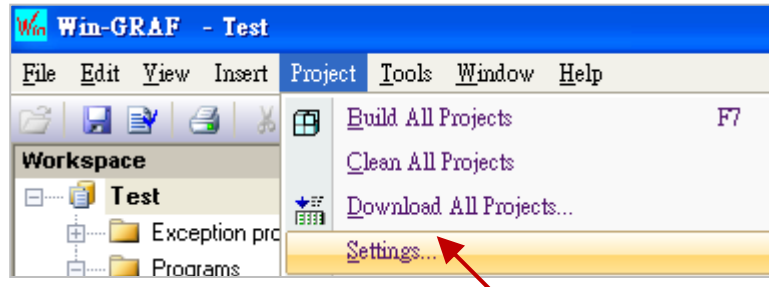
```
TMP_BOOL := PAC_Reboot() ;
```

若出現 PAC 一直重開機的錯誤，

可將 Win-GRAF PAC 上的旋轉開關調到 "1" (即，安全模式) 並重新開機。開機後，重新下載專案，並將旋轉開關調回 "0" (即，一般模式) 即可。

11.6 在 LD 與 FBD 程式內使用 ST 語法

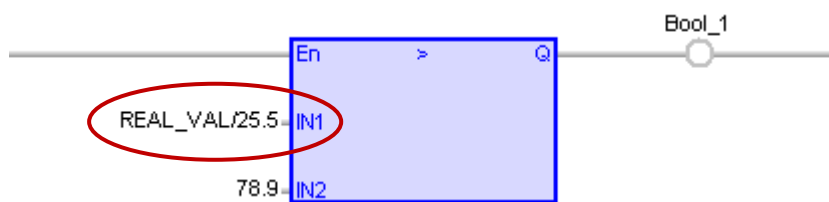
Win-GRAF Workbench 允許使用者在 LD 或 FBD 程式內，使用簡單的 ST 語法，以便利程式設計。使用前，請點選功能表 "Project - Settings"，接著點選 "Runtime" 並將 "Complex variables in a separate segment" 設為 "Yes" 來啟用此功能。



使用範例:

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。

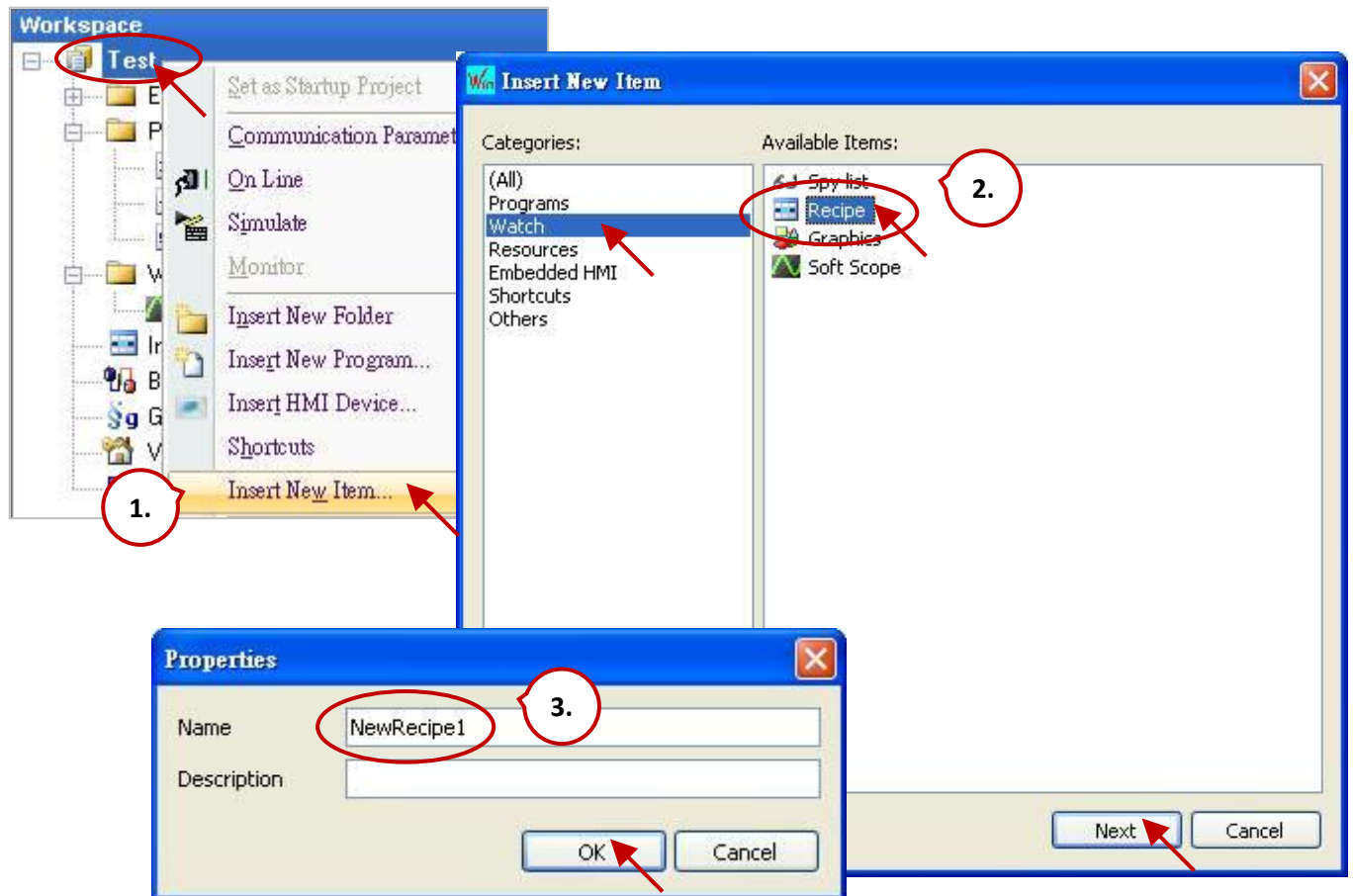


11.7 PAC 內套用配方表 (Recipe)

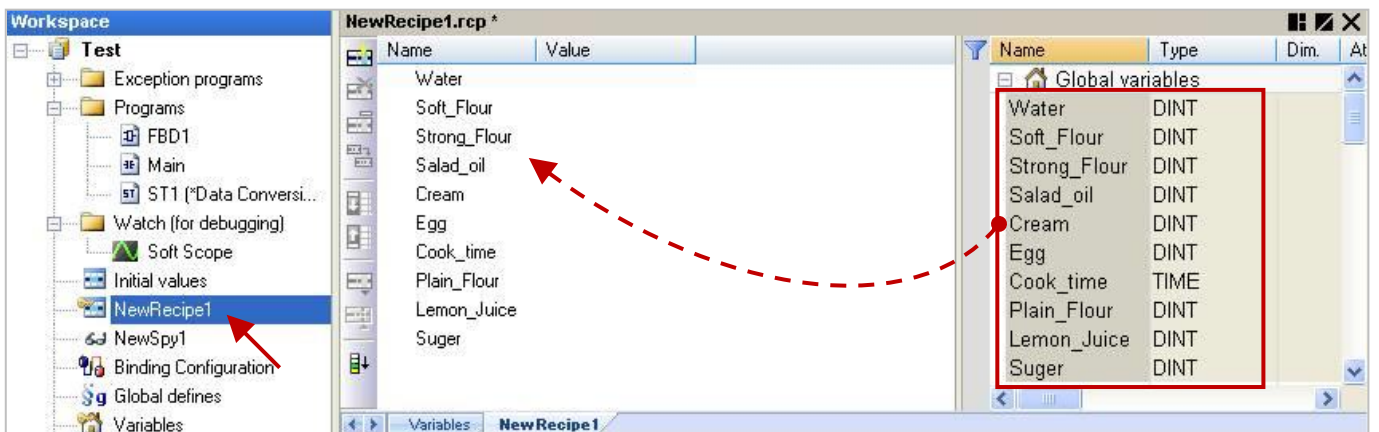
某些應用會需要因應不同的製程來替換配方表 (Recipe) 中的產品配置，使用者可預先定義好每個產品所需使用的變數與數值。當要生產其他品項時，可選取想要更換的產品配置，並套用到 PAC 中。

請依照以下步驟：

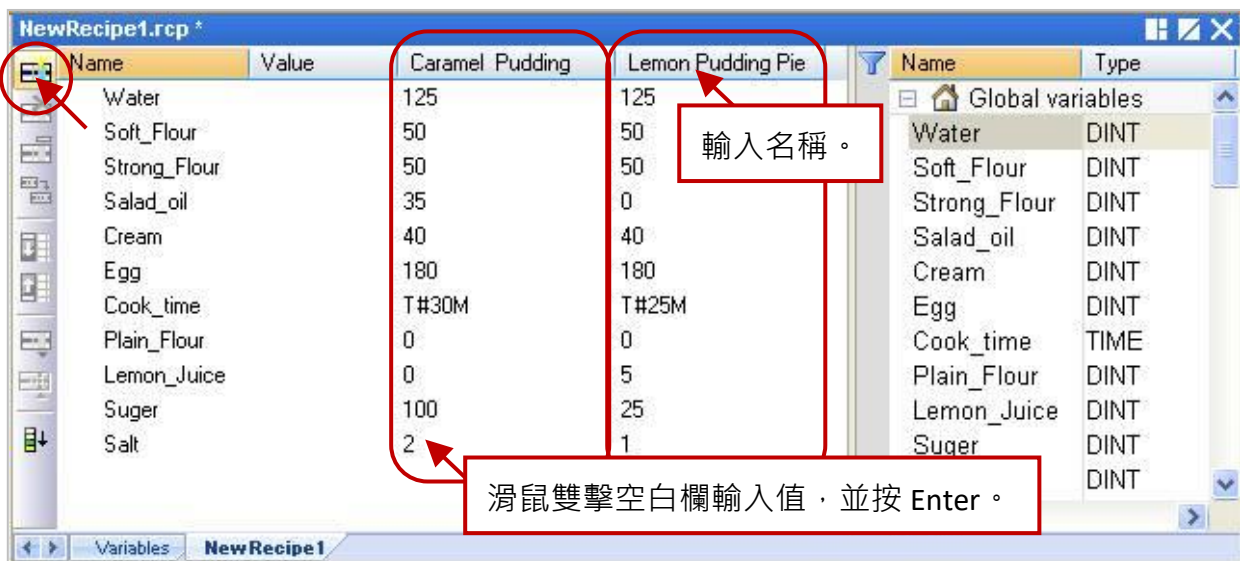
1. 新增配方表。滑鼠右鍵點選專案名稱並選擇 "Insert New Item" 項目，點選 "Watch/ Recipe" 並點選 "Next"，接著輸入名稱 (例如: "NewRecipe1") 並按 "OK"。



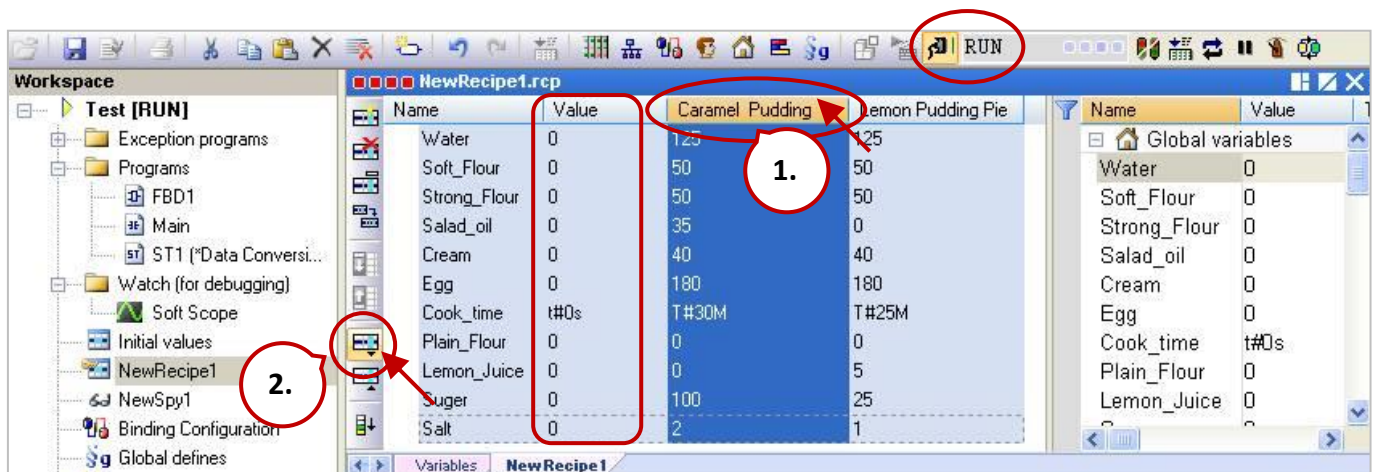
2. 指定變數。滑鼠雙擊 Workspace 內的 "NewRecipe1" 開啟視窗，並將所需的變數拖曳到視窗內。



3. 設定配方表。點選 "Insert Column" 按鈕來新增欄位並輸入名稱，滑鼠雙擊空白欄輸入值，並按 Enter 鍵。



4. 套用配方表。將專案下載到 PAC，一開始值皆為 "0"。請點選所需的品項，再點選 "Send Recipe" 將此配置套用到 PAC 中。



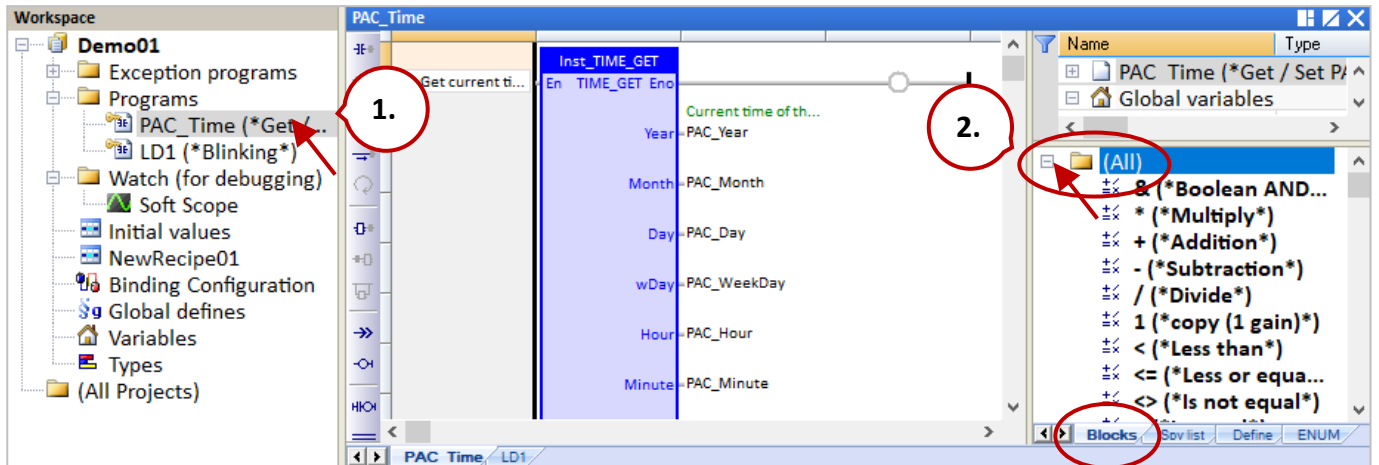
Name	Value	Caramel Pudding	Lemon Pudding Pie
Water	125	125	125
Soft_Flour	50	50	50
Strong_Flour	50	50	50
Salad_oil	35	35	0
Cream	40	40	40
Egg	180	180	180
Cook_time	t#30m	T#30M	T#25M
Plain_Flour	0	0	0
Lemon_Juice	0	0	5
Suger	100	100	25
Salt	2	2	1

註: 您也可參考第 6 章，來使用可保存變數，以確保數值不會因突然斷電而消失。

11.8 Win-GRAF PAC 有支援的 Function 與 Function Block

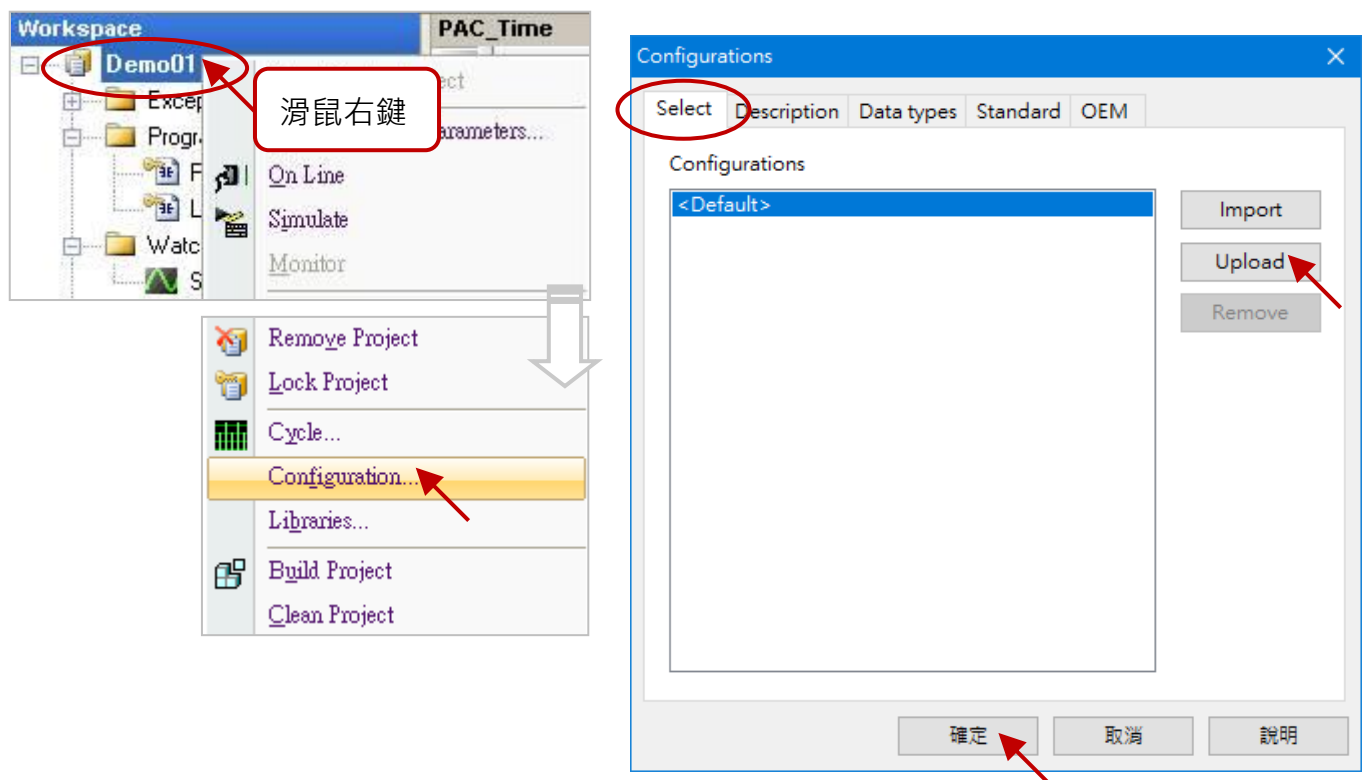
Win-GRAF 的 "Blocks" 面板內，列有許多的 Function 與 Function Block，然而有一些 Win-GRAF PAC 並未支援，以下將說明如何分辨是否有支援。

為了方便檢視所有的 Function 與 FB，請在任一個程式的 **Blocks** 面板中，展開 **All** 資料夾。

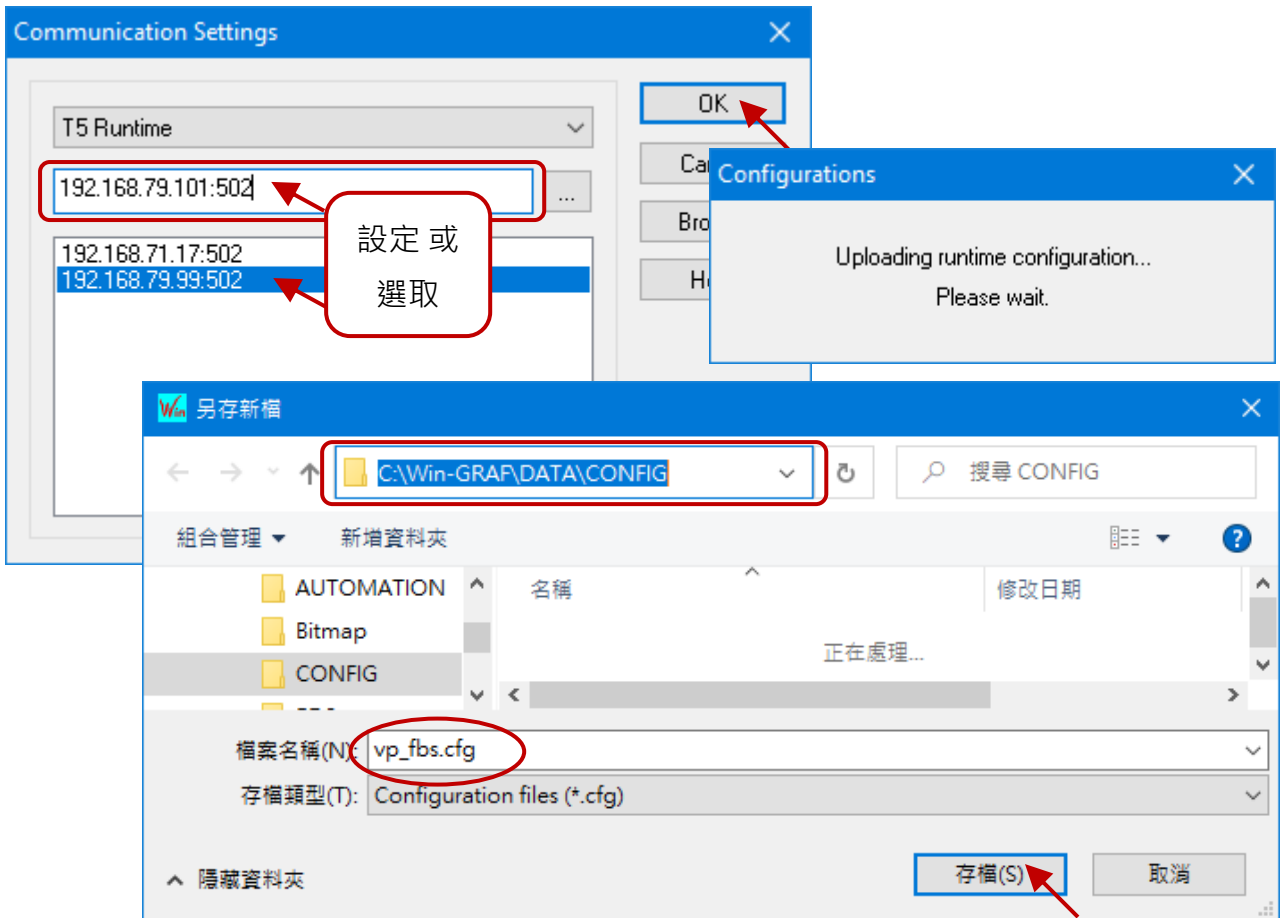


從 PAC 上傳組態檔，請依照以下步驟:

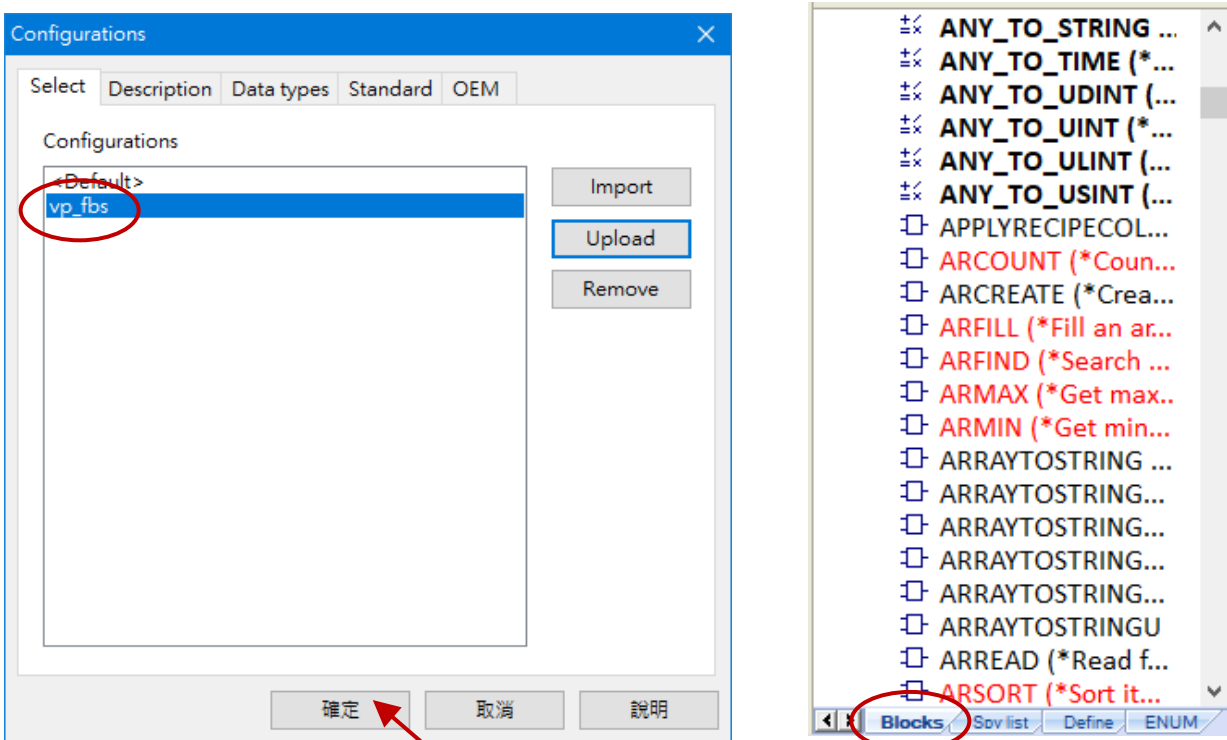
1. 請確認 PAC 有開機，且與 Win-GRAF/PC 在相同的網段上。
2. 滑鼠右鍵點選 Win-GRAF 專案名稱再選擇 "Configuration"，接著在 "Select" 頁面中點選 "Upload" 按鈕開啟設定視窗。



3. 設定/選取 PAC 的 IP 位址，並點選 "OK" 按鈕開始上傳檔案。組態檔預設儲存至 C:\Win-GRAF\DATA\CONFIG 目錄下，請輸入檔名 (例如: "vp_fbs.cfg") 並按 "存檔" 按鈕。



4. "Configurations" 視窗內會顯示檔案名稱，請點選 "確定" 離開。接著，請再檢視 **Blocks** 面板，所有標示紅色的 Function 與 FB，Win-GRAF PAC 皆未支援。

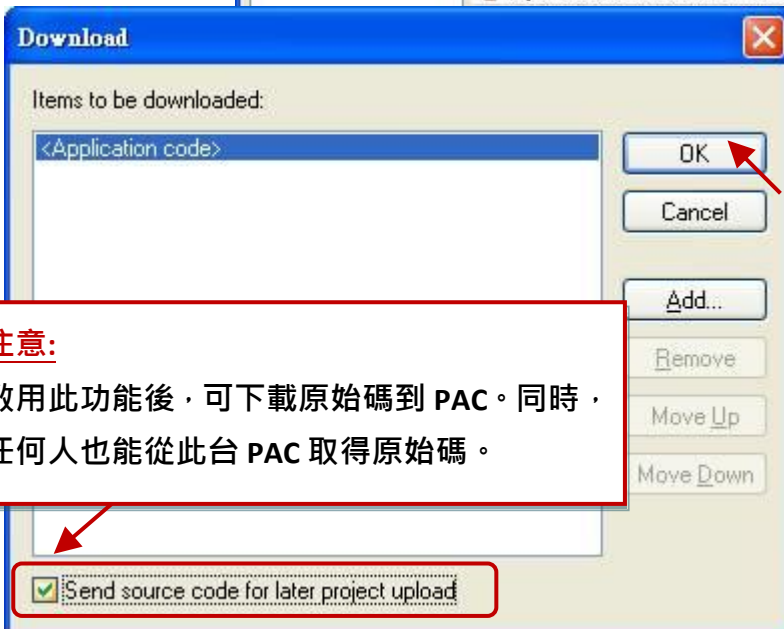
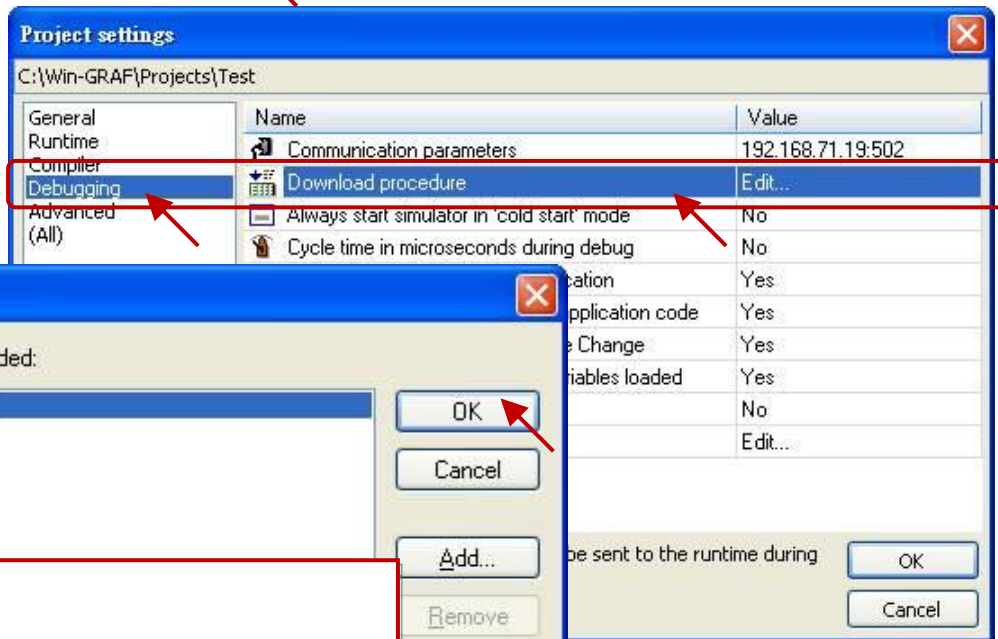
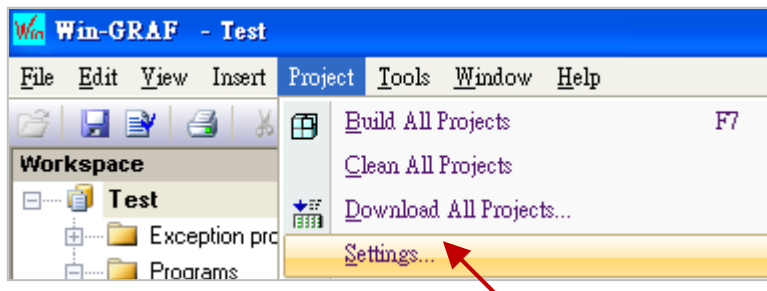


11.9 上傳 Win-GRAF 專案原始碼

因某些原因，像是為了避免原始碼 (Source Code) 遺失 或 上一位工作者移交不完整的原始碼，因此必需從 PAC 上傳原始碼到 PC。然而，使用者必需先開啟 **Download Prodedure** 功能並下載專案至 PAC 一次，下次才能上傳有原始碼的 Win-GRAF 專案。

開啟 Download Prodedure 功能:

1. 點選功能表 "Project - Settings" 開啟設定視窗。
2. 滑鼠雙擊 "Debugging" 設定中的 "Download procedure"，接著勾選 "Send source code for later project upload" 再點選 "OK" 完成設定。



注意:

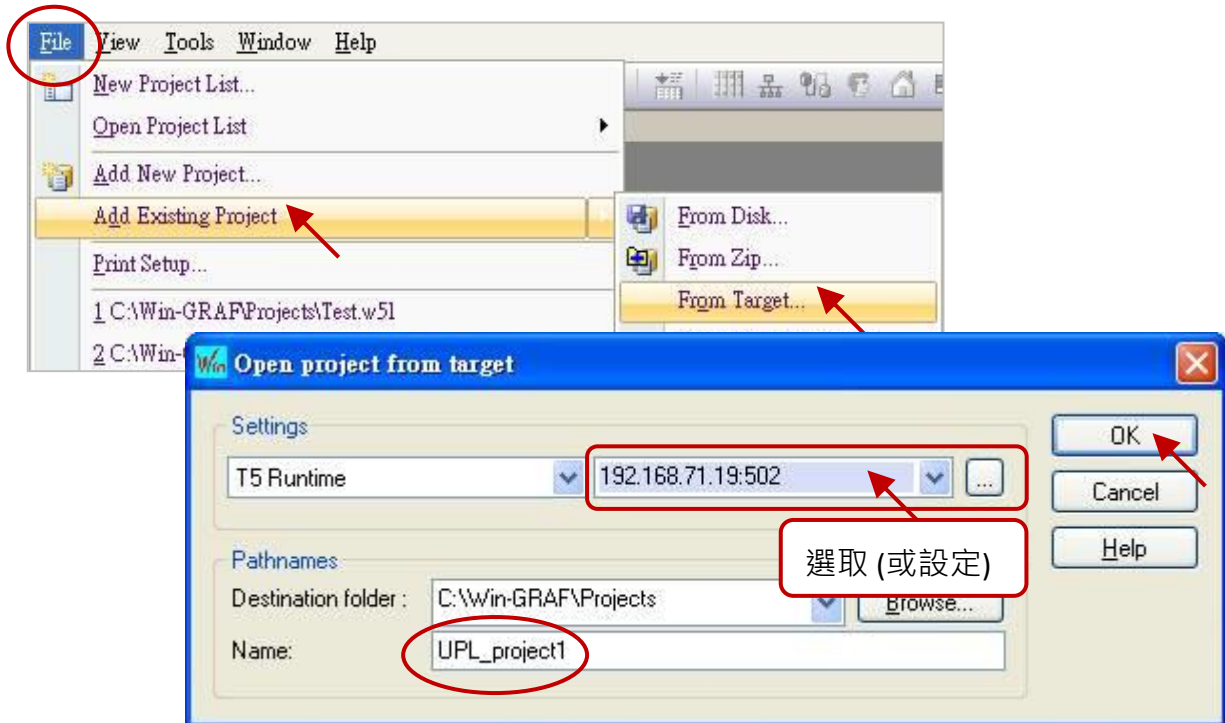
啟用此功能後，可下載原始碼到 PAC。同時，任何人也能從此台 PAC 取得原始碼。

3. 請編譯程式 (點選功能表 "Project - Build All Projects") 並將專案下載到 PAC 中。Source Code 會存放在 PAC 的 `/System_Disk/Win-GRAF/t5.upl`，此檔案大小會隨著專案內容而改變，當專案變得龐大且複雜時，此檔案可能會達到幾百 K Byte 或甚至超過 1 MB。

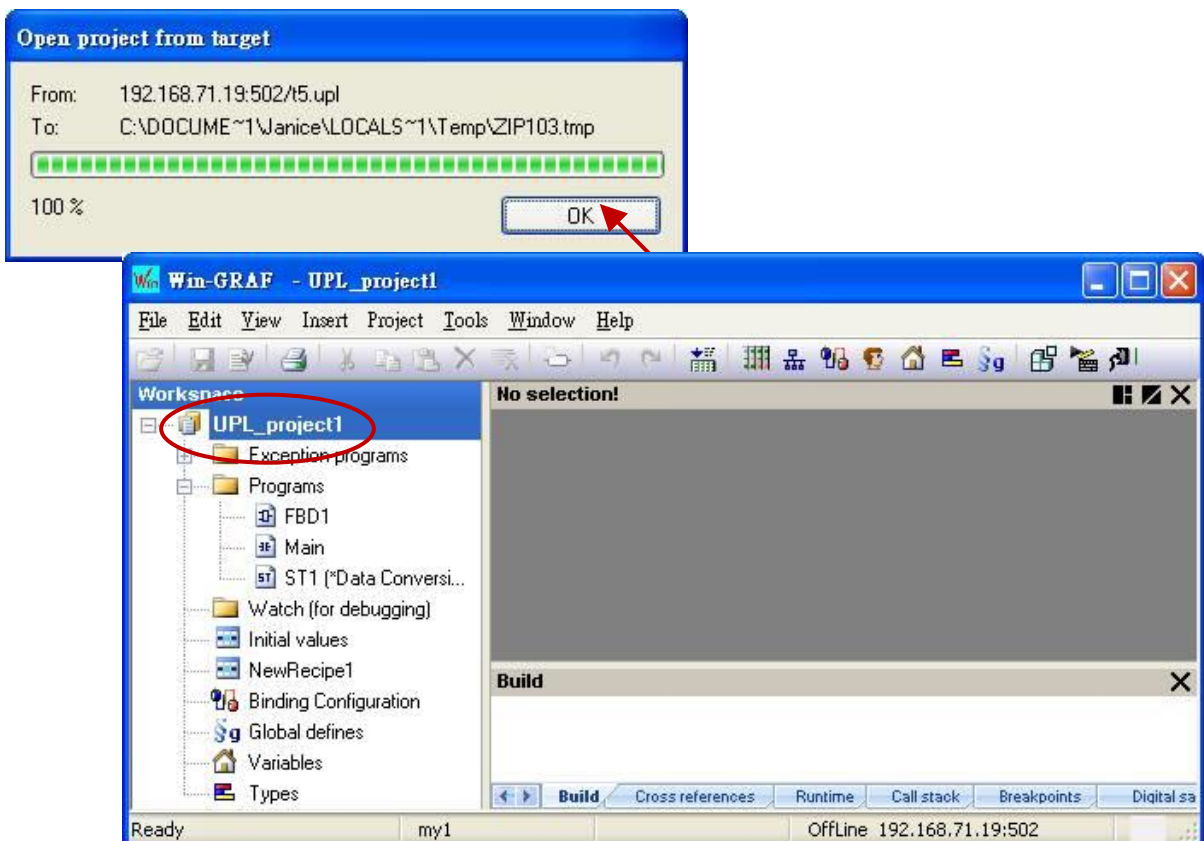
上傳專案原始碼:

請先關閉所有的 Win-GRAF 專案 (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 (或設定) PAC 的 IP 位址並輸入專案名稱 (例如: "UPL_project1")，再點選 "OK" 開始上傳。



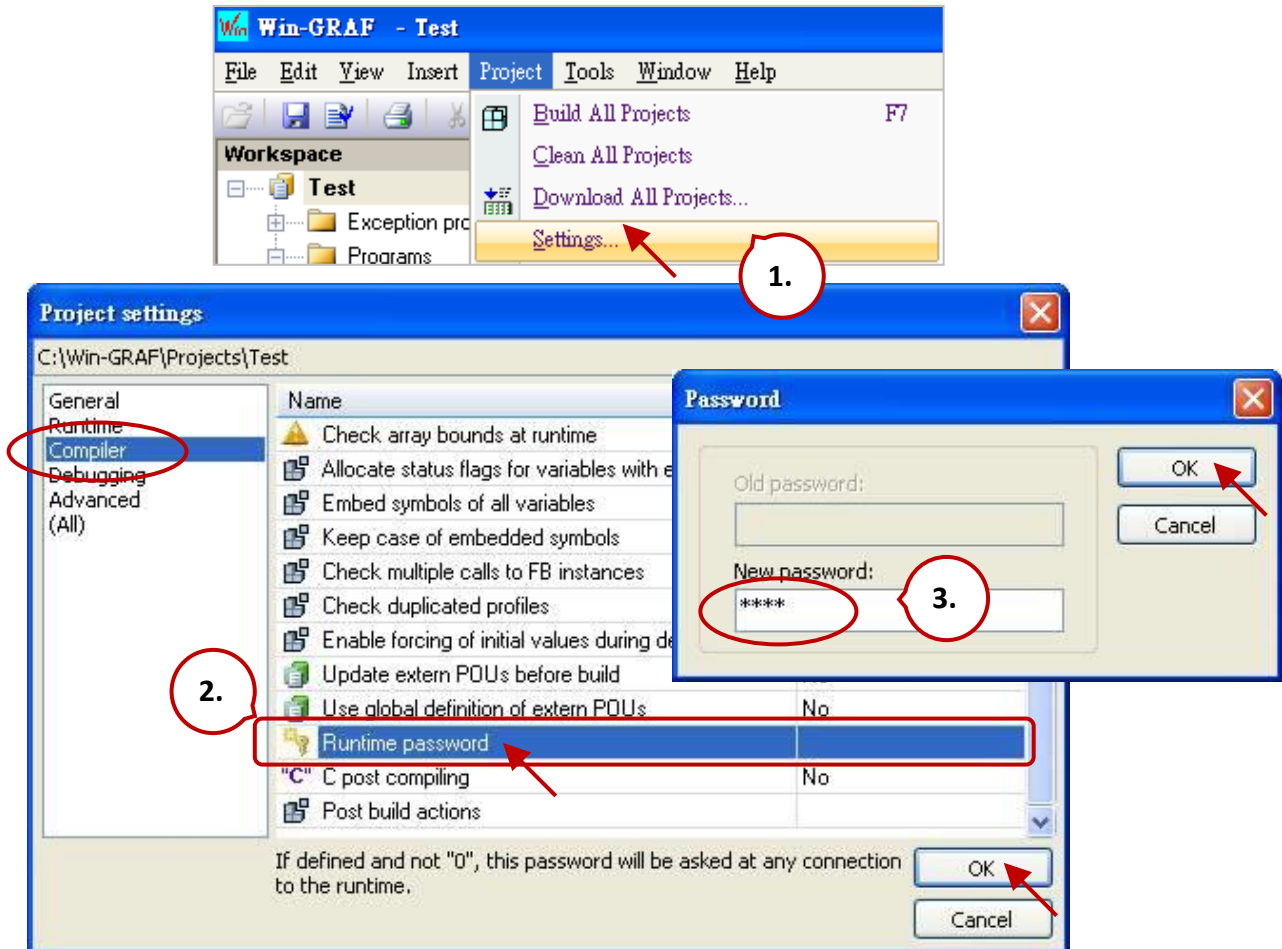
5. 上傳完成後，請點選 "OK" 按鈕會自動開啟此 Win-GRAF 專案。



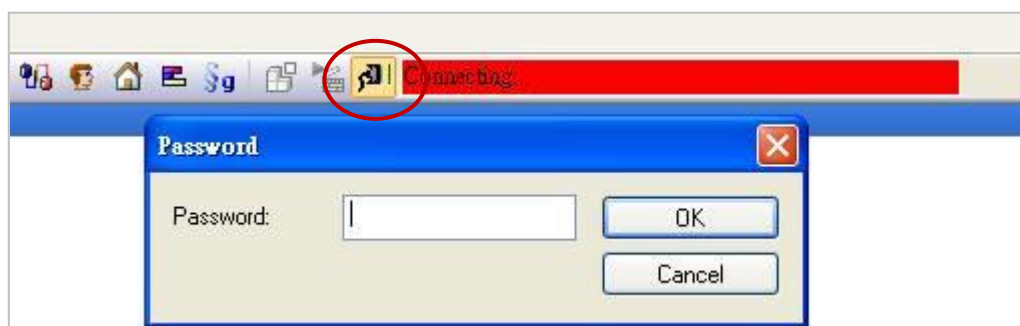
11.10 設定 PAC 的密碼

為了避免 PAC 在運行重要程式時，被某一台 PC 連線並惡意更動 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. 點選功能表 "Project - Settings" 開啟設定視窗。
2. 滑鼠雙擊 "Compiler" 設定中的 "Runtime password" 並設定密碼，再點選 "OK" 按鈕。



3. 請編譯程式 (點選功能表 "Project - Build All Projects") 並將專案下載到 PAC 中。下次執行 "On Line" 與 PAC 連線時，將會要求輸入密碼。



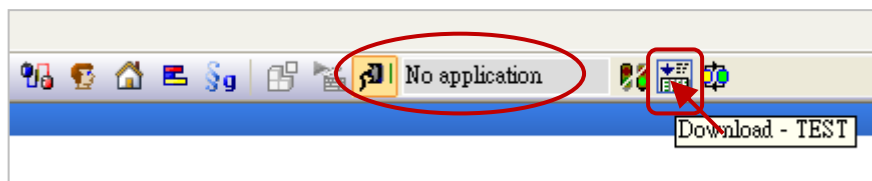
注意： 啟用密碼後，請務必記得密碼，否則 Win-GRAF 將無法連上 PAC。

唯一的解決方法:

可將 Win-GRAF PAC 上的旋轉開關調到 "1" (即，安全模式) 並重新開機。開機後，重新下載專案，並將旋轉開關調回 "0" (即，一般模式) 即可。



旋轉開關位置	說明
0	正常模式
1	不執行用戶的應用程式
2	LAN1: DHCP 模式
3	LAN1: 靜態 IP 模式 (Static IP) IP: 192.168.255.1 Mask: 255.255.0.0
4	更新韌體 (Firmware) 模式



11.11 在 ST 程式內使用 Function 或 Function Block

在 ST 程式內使用 Function 的方法很簡單，只需呼叫該 Function 並填入對應的參數即可。以下程式會先開啟 COM1，然後每隔 5 秒會從 COM1 送出一個 'Hello' 字串。

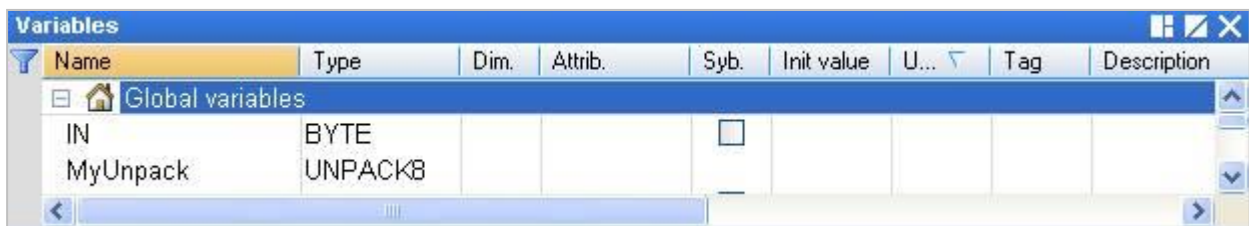
```
(* 宣告 "INIT1" 為 BOOL 並指定初值為 TRUE,
   宣告 "TMP_BOO" 為 BOOL, "TMR1" 為 TIME *)

IF INIT1 THEN
  INIT1 := FALSE ;
  TMR1 := T#0s ;
  TSTART (TMR1) ;
END_IF;
IF COM_Status(1) = FALSE THEN
  TMP_BOO := COM_open (1, `19200,N,8,1');
END_IF;
IF TMR1 >= T#5s THEN
  TMR1 := T#0s ;
  COM_send_str (1, `Hello: ');
END_IF;
```

若要在 ST 程式內使用 Function Block，需先在變數區宣告該 FB 的實例變數 (Instance)，之後，和 Function 的用法類似，如下：

範例: 1 個 byte 拆解成 8 個 BOOL:

1. 宣告 "MyUnpack" 為 UNPACK8 (FB) 變數、"IN" 為 BYTE 變數。



Name	Type	Dim.	Attrib.	Syb.	Init value	U...	Tag	Description
Global variables								
IN	BYTE			<input type="checkbox"/>				
MyUnpack	UNPACK8							

2. 編寫一個 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 保護您的 Win-GRAF 程式，讓盜用者無法使用

當您完成 Win-GRAF 專案開發，並準備交付給客戶時，請考慮到，Win-GRAF PAC 內的專案，是否可能被第三方複製到另一台同型號的 PAC 內？！注意！別人可能會盜用了您辛苦開發的成果，以下提供一個簡單好用的方法來保護您的程式。

注意：如果您將 Win-GRAF 專案的 Source Code 交給了其他人，那很抱歉，以下方法無法保護您的程式。因為任何取得 Source Code 的人，即可修改程式碼並套用到另一台 PAC 上。

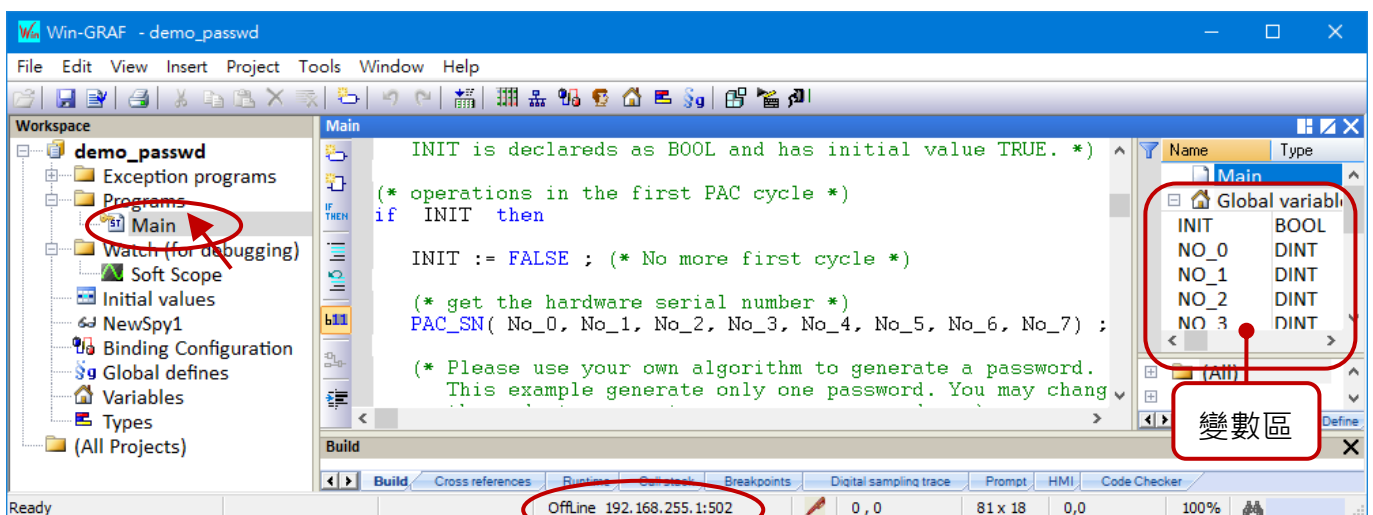
每一台 ICP DAS Win-GRAF PAC 都會有一個獨一無二的 64-Bit 序號 (Serial Number)，可用來在程式中加上自訂的運算來產生一組密碼。此外，可在 Win-GRAF 專案中加入驗證方式，當專案下載到 PAC 後，會檢查密碼是否正確？若不正確，則不允許程式運作。

範例說明

- 1) "demo_passwd": 用來產生密碼並存到 PAC 的檔案內。
- 2) "demo_my_ap": 要給客戶的專案，具有密碼驗證功能。

您必需先將 "demo_passwd" 程式下載到 PAC 內 Run 一次，來產生專屬的密碼。然後，再下載 "demo_my_ap" 到同一台 PAC 內。此後，即使 Win-GRAF 專案被複製到另一台同型號的 PAC，也會因為密碼驗證失敗而無法運作。

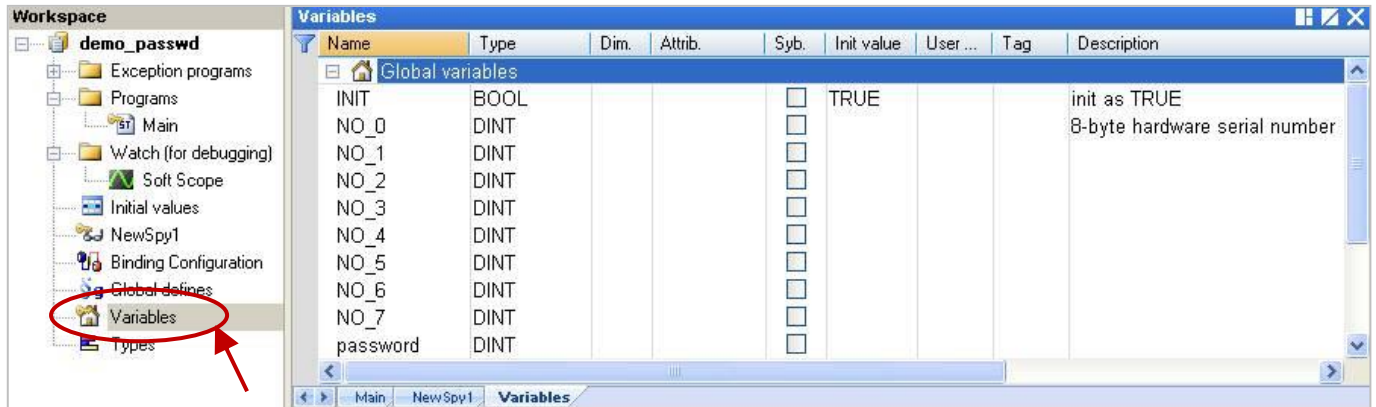
請在網站下載 [範例程式](#) (demo_passwd.zip 與 demo_my_ap.zip)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。



Win-GRAF 專案: "demo_passwd"

一開始先使用 "PAC_SN" 函式來讀出 PAC 的序號，接著加上自訂的演算法來產生密碼，然後儲存到 PAC 的檔案內。使用者可自行決定檔案位置。

變數宣告:



ST 程式 - Main:

(* 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);

(* 請使用您自訂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));

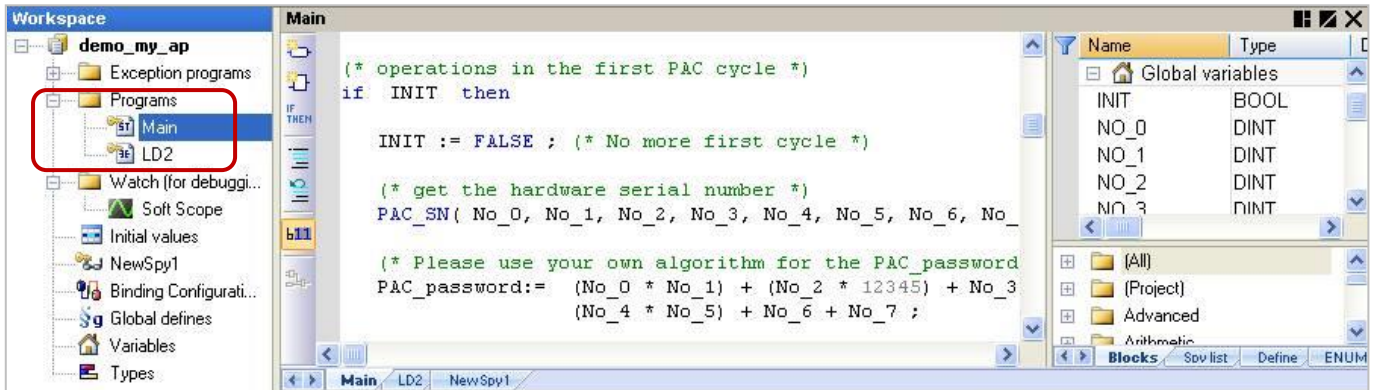
f_close(file_id); (* 關閉檔案 *)

end_if ;

end_if ;

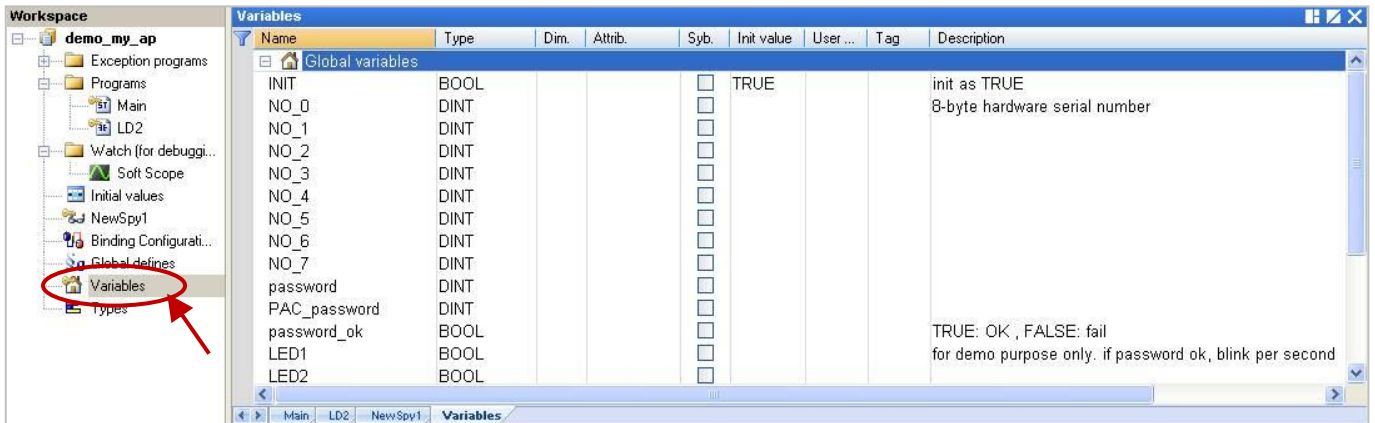
Win-GRAF 專案: "demo_my_ap"

一開始會用上述相同方式產生密碼，接著讀取 PAC 檔案中的密碼進行比對，若密碼不正確，則會離開程式。



(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
```

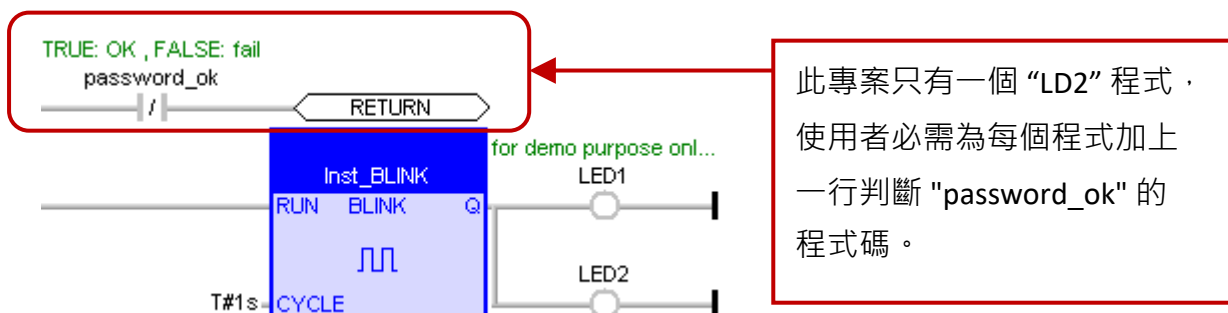
```
  password_ok := TRUE ; (* 密碼正確 *)
```

```
end_if ;
```

```
end_if ;
```

LD 程式 – LD2

如果 "password_ok" 為 "FALSE" 表示密碼不正確，將會退出此程式。只有密碼正確時，才能執行後續的程式，如此即可保護您的程式無法被盜用者使用。

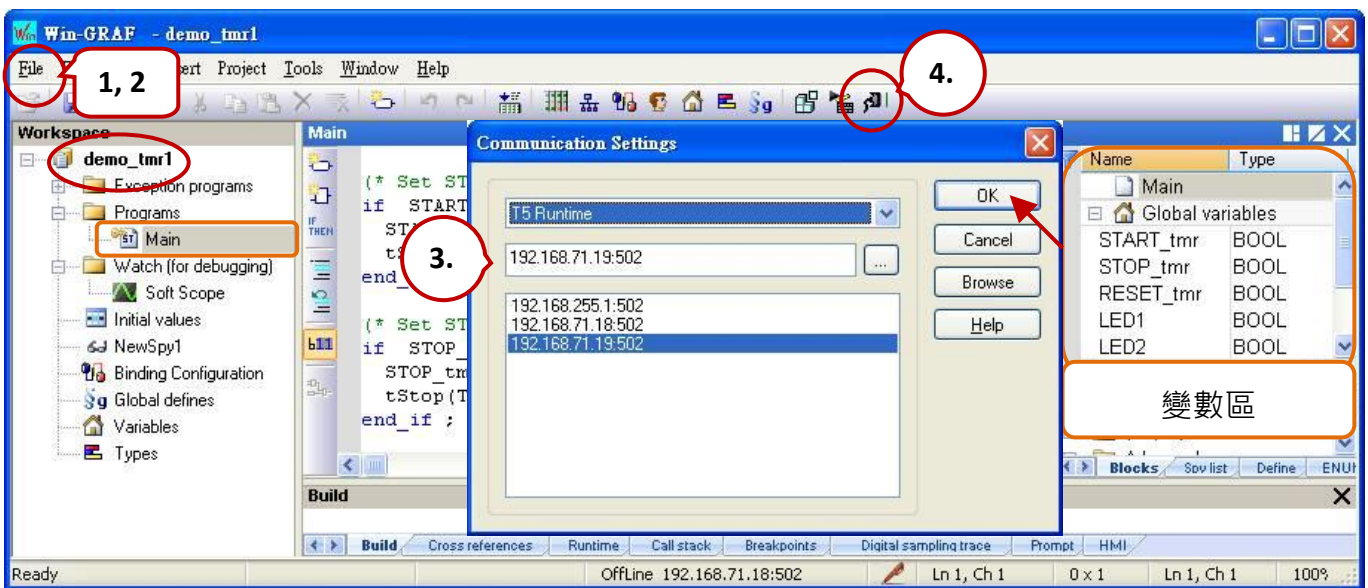


第 12 章 範例程式說明

您可在 [網站下載](#) Win-GRAF 範例程式:

請依照以下步驟，來執行範例程式:







1. 點選功能表 "File - Close Project List" 關閉所有開啟中的專案。
2. 點選功能表 "File - Add Existing Project - From Zip" 回存專案 (例如: "demo_tmr1.zip")。
(您可參考 11.4 節來備份/回存專案。)
3. 滑鼠右鍵點選專案名稱，再選擇 "Communication Parameters" 設定 PAC 的 IP 位址。
4. 點選功能表 "Project - On Line" 下載專案到 PAC。(參考 2.3.4 節)



將範例程式下載到 PAC 前，請確認以下設定是正確的:

1. Win-GRAF 通訊 IP、Modbus TCP Slave IP。
RPAC 預設 IP · LAN1: 192.168.255.1 · LAN2: 172.16.255.1 · LAN3: 10.0.255.1
2. COM Port 或 Baudrate 設定。
COM1: RS-232 · COM2: RS-485 · COM3: RS-422
3. 檔案位置 (例如: /mnt/microSD/ 或 /System_Disk/Win-GRAF/)

您可參考 12.1 節，在下列視窗修改為適當的參數:

 Main	LD 程式		I/O Boards
 Main	ST 程式		I/O Drivers
 Variables	Variables		Binding

12.1 範例程式列表

請依照上述步驟，將範例程式回存到 Win-GRAF Workbench 內。

檔案名稱	說明	章節
win-graf-linux-demo-all.zip : 下載所有的 Win-GRAF 範例程式 (674 KB)		
demo_et7060	連接遠端 ET-7060 Ethernet I/O 模組 I/O Drivers: Modbus Slave IP (10.0.255.2:502)	5.2.2
demo_et7018z	連接遠端 ET-7018Z Ethernet I/O 模組 I/O Drivers: Modbus Slave IP (10.0.255.2:502)	5.2.3
demo_tgw725	使用 tGW-700 閘道器來存取多個 Modbus RTU Slave I/O Drivers: Modbus Slave IP (Port1): 10.0.255.2:502 Modbus Slave IP (Port2): 10.0.255.2:503	5.4
demo_retain	使用可保存變數來保有 PAC 停電前的變數資料	6.1
demo_wp5_retain	使用檔案來保存變數 Variables: File_Path1 ('/System_Disk/Win-GRAF/retain_real.txt') Variables: File_Path2 ('/System_Disk/Win-GRAF/retain_other.txt')	6.2
demo_binding	PAC 間互傳資料 (Data Binding) Binding: Exern (192.168.255.1/10.0.255.1)	7
遠端 DCON I/O 模組 (8.2.1 ~ 8.2.8) I/O Boards: DCON (Port = 2, Baud rate = 9600)		
demo_d_7065	連接遠端 I-7065 I/O 模組 Function Block: D_7065 (Port = 2, Address = 2)	8.2.1
demo_d_7018z	連接遠端 I-7018Z I/O 模組 Function Block: D_7018Z (Port = 2, Address = 3)	8.2.2
demo_d_7083	連接遠端 I-7083 I/O 模組 Function Block: D_7083 (Port = 2, Address = 4)	8.2.3
demo_d_87084_fr	連接遠端 I-87084W 模組來量測頻率 Function Block: D_7084_FREQ (Port = 2, Address = 5)	8.2.4
demo_d_87084_c4	連接遠端 I-87084W 模組來量測 Counter Function Block: D_7084_CNT4 (Port = 2, Address = 5)	8.2.5
demo_d_87084_c8	Function Block: D_7084_CNT8 (Port = 2, Address = 5)	8.2.6
demo_dl_100T485	連接遠端 DL-100T485 (溫度/濕度) 模組 Function Block: DL_100T485 (Port = 2, Address = 1)	8.2.7
dmeo_gps721	連結 GPS-721 遠端 GPS 接收模組 Function Block: D_GPS721 (Port = 2, Address = 1)	8.2.8
demo_my_ap	保護您的 Win-GRAF 程式，讓盜用者無法使用	11.12
demo_passwd	Main (ST): file_name ('/System_Disk/Win-GRAF/my_product.pwd')	
demo_tmr1	使用 tStart 與 tStop 函式，來操作計時器	12.2.1
demo_tmr2	對計時器進行周期性的操作	12.2.2
demo_tmr3	較精準的對計時器進行周期性的操作	12.2.2

demo_com_port1	使用 COM Port 來傳送一個字串 Variables: Port_number (1: RS-232), Main (ST): '9600,N,8,1'	12.3.1
demo_com_port2	使用 COM Port 對設備一問一答 Variables: Port_number (2: RS-485), Main (SFC): '9600,N,8,1'	12.3.2
demo_com_port3	COM Port 等待接收遠端設備傳來的資料 Variables: Port_number (2: RS-485), Main (ST): '9600,N,8,1'	12.3.3
demo_com_port4	使用 COM Port 定期回報資料給遠端設備 Variables: Port_number (2: RS-485), Main (SFC): '19200,E,8,2'	12.3.4
demo_file1	寫入資料到 PAC 內的檔案 Main (ST): '/System_Disk/Real_data1.txt'	12.4.1
demo_file2	讀取 PAC 內的檔案 Variables: File_path ('/System_Disk/Real_data2.txt')	12.4.2
demo_datalog	資料紀錄功能 Main (ST): /mnt/microSD/yyyy-mm/yyyy-mm-dd.csv	12.4.3
demo_vb03	使用 C 程式來讀取 Win-GRAF 變數	13
demo_vb04		
◇ 備援 (冗餘) 系統 i_redundancy: Active_IP = 192.168.71.37, Mask=255.255.0.0 LAN2 IP 固定: 199.193.195.17 (Main)/ 199.193.195.9 (Backup) 自訂安全機制: 請見 RDN_control 程式		14
demo_rdn_1	Function Block: D_87064、D_87018Z、D_7065 (Port = 2, Address = 2、3、4) I/O Boards: DCON (Baud rate = 9600)、i_redundancy、i_redundancy_rs485	
demo_rdn_2	I/O Boards: i_redundancy	
demo_rdn_3	LAN3, ET-7050 I/O Drivers: Modbus Slave IP (10.0.255.2:502) I/O Boards: i_redundancy	
demo_rdn_4	LAN3, iDCS-8830 I/O Drivers: Modbus Slave IP (10.0.79.200/10.0.79.201) I/O Boards: i_redundancy	
demo_schedule	排程控制 I/O Boards: Schedule (Password: 0) Binding: 控制變數 (ID: 5001 ~ 5030)	15
demo_sms	使用一個 2G/3G GSM Modem 來收發簡訊 Variables: Phone_Nb ('0900629879')	16
demo_extra_port	啟用一個序列埠，讓 Win-GRAF Workbench 連進來 Variables: file1_name ('/System_Disk/Win-GRAF/Extra_Ports.txt') ST1 (ST): 'COM2:115200,N,8,1'	C
demo_pid_simple	PID 應用與調節器應用。	-

12.2 計時器 (Timer) 操作

12.2.1 啟動、停止、重置計時

開啟此專案 ("demo_tmr1.zip")，於變數區可查看變數 (參考第 12 章)。

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;
```

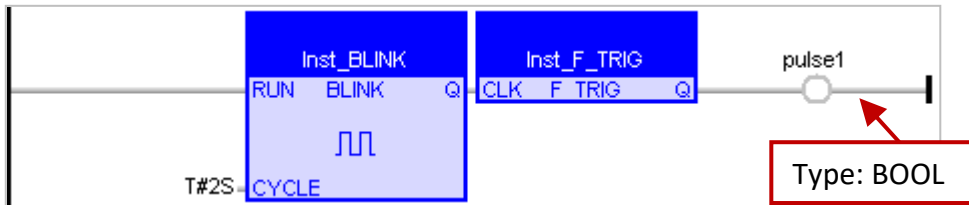
```
(* 當 "TMR1" 為 T#15s 時, 自動重置 "TMR1" 為 T#0s *)
```

```
IF TMR1 >= T#15s THEN  
    TMR1 := T#0s;  
END_IF;
```

12.2.2 週期性的操作

開啟此專案 ("demo_tmr2.zip")，於變數區可查看變數 (參考第 12 章)。此例，使用 "BLINK" Function 與 "F_TRIG" Function Block，可在指定的間隔時間產生一個 Pluse TRUE，適用於週期性的操作。

LD 程式:



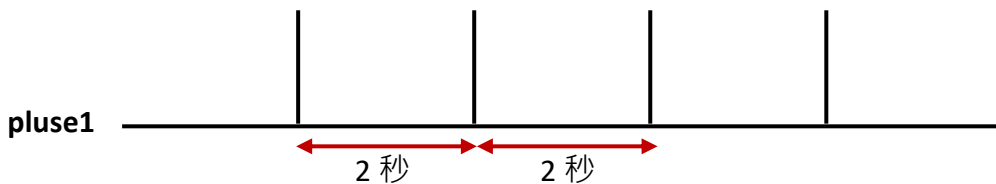
ST 程式:

```

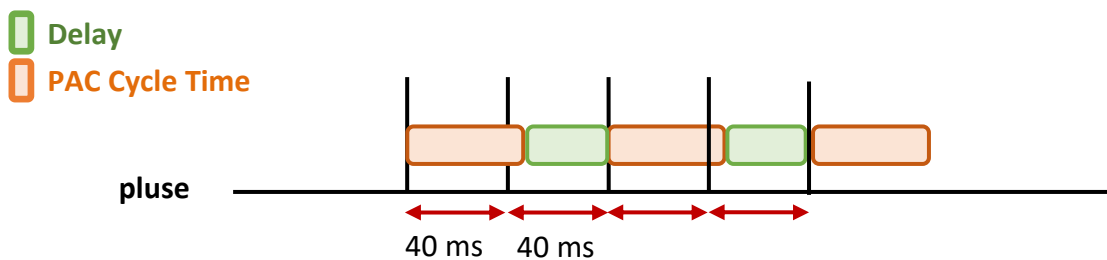
IF pluse1 THEN
  (* 執行操作 *)
  ...
END_IF;

```

上方的 LD 程式，會每隔 2 秒產生一個 Pluse TRUE，但是此方式有個缺點:



若指定的間隔時間較短 (例如: 小於 100 ms) 或 PAC 的 Cycle Time 較大 (例如: 20 ~ 50 ms，一般為 3 ~ 15 ms) 時，則此操作時間會變得不準確。例如，指定每 40 ms 執行一次操作，由於相較於 2 秒 或 250 ms，間隔時間為 40 ms 相當接近 PAC 的 Cycle Time，此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:

開啟此專案 ("demo_tmr3.zip")，於變數區可查看變數 (參考第 12 章)。

ST 程式:

(* 宣告 "INIT" 為 BOOL 且設定初值為 TRUE
宣告 "TMR1", "TMR1_next" 為 TIME *)

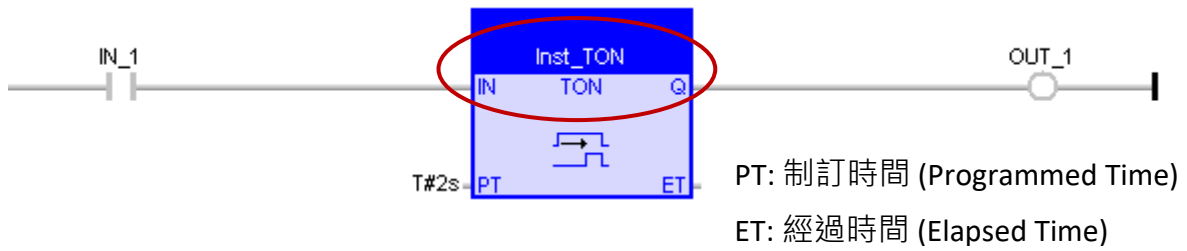
```
IF INIT THEN
  INIT := FALSE ;
  TMR1 := T#0s ;
  TMR1_next := TMR1 + T#50 ms ;
  TSTART (TMR1);
END_IF;

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;
```

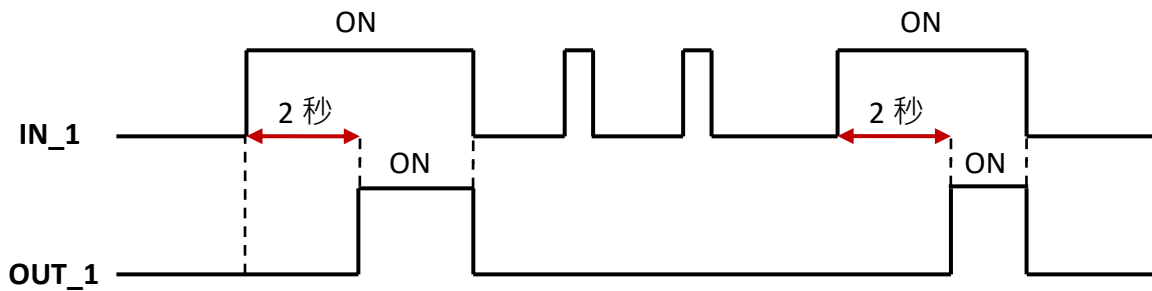
當計時到 T#23h59m59s999ms 時，
Timer 的值會產生溢位 (Overflow)。
因此，可設定它在 10 或 18 小時，
自動重置為 "0"。

12.2.3 偵測穩定的 ON 或 OFF 訊號

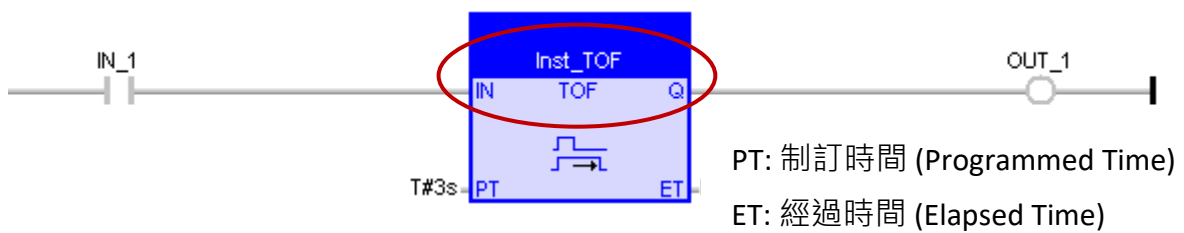
"TON" 功能方塊，可偵測在指定時間內仍維持 "ON" 的穩定訊號。



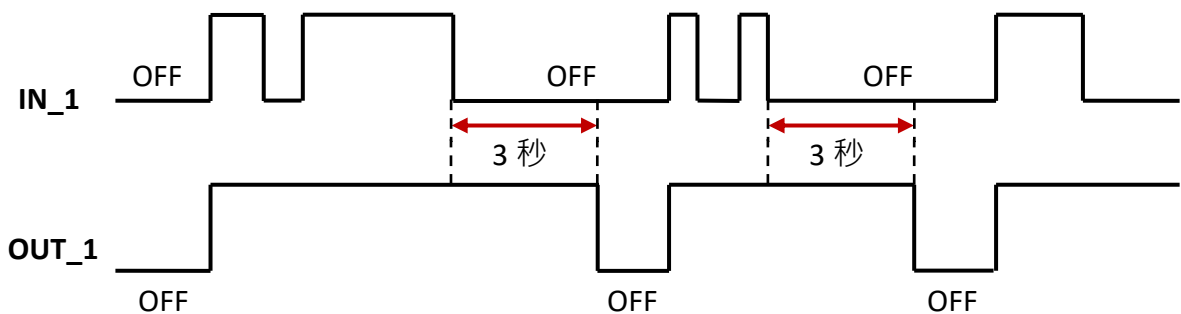
此例，此功能可偵測維持 2 秒仍 "On" 的穩定訊號。



"TOF" 功能方塊，可偵測在指定時間內仍維持 "OFF" 的穩定訊號。

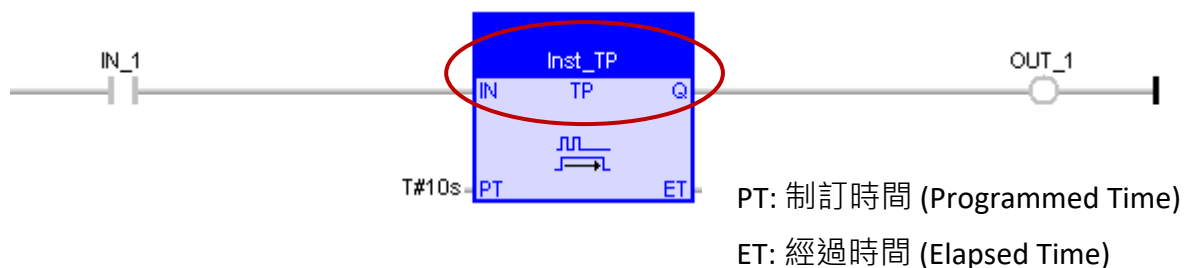


此例，此功能可偵測維持 3 秒仍 "OFF" 的穩定訊號。

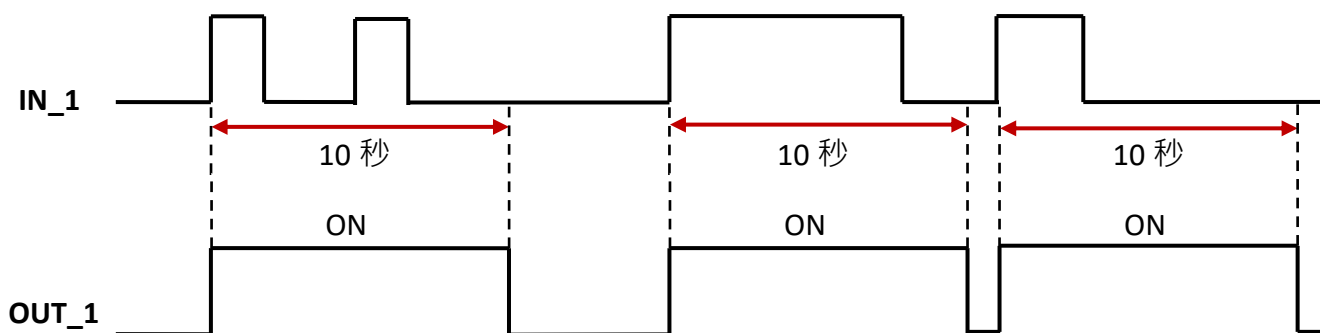


12.2.4 觸發後維持 "ON" 一段時間

"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。



此例，IN_1 為 ON 時，會維持輸出 10 秒的 "ON" 訊號。

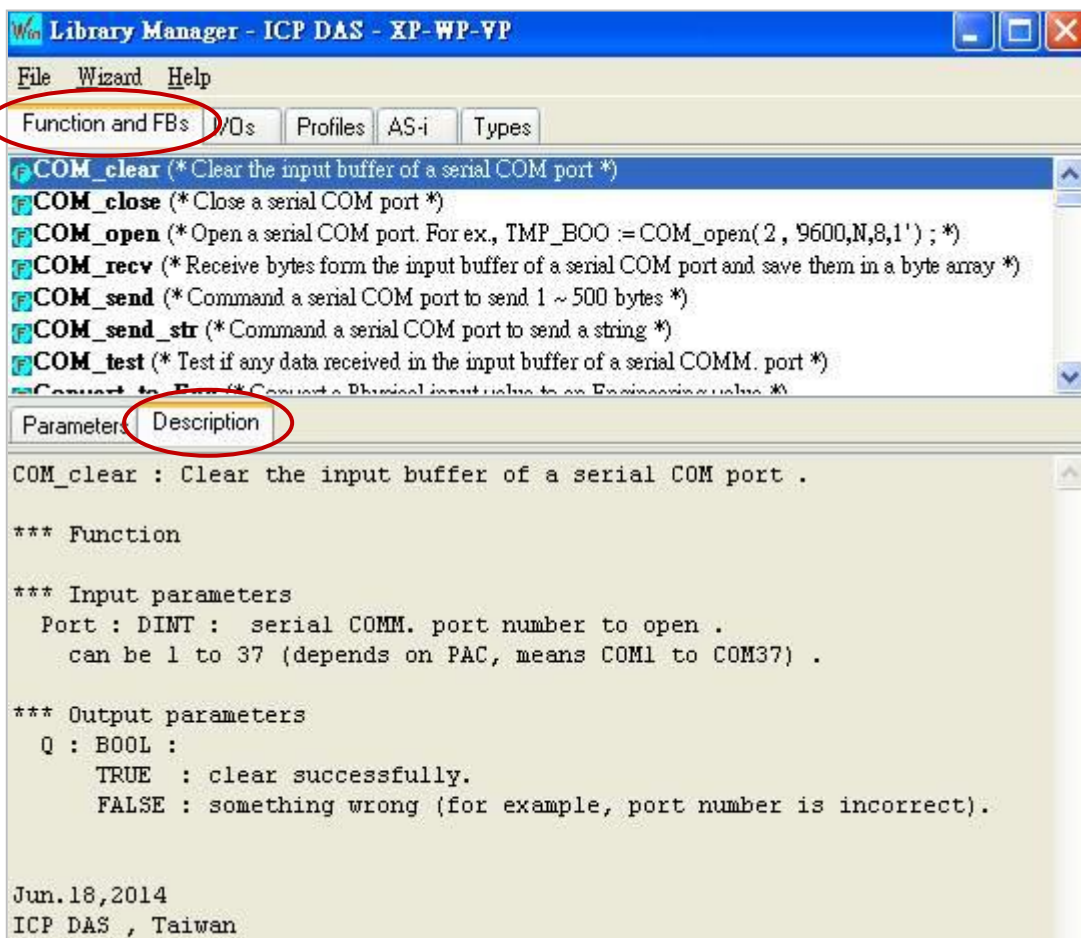


12.3 序列埠的通訊操作

序列通訊通常用來控制或接收資料，您可使用下列函式 (Function) 直接對序列埠 (即，RS-232, RS-485 或 RS-422) 進行指定的通訊動作。

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 資料?
COM_send	傳送 Byte 資料給序列埠。
COM_send_str	傳送 String 資料給序列埠。
COM_rcv	由序列埠的輸入暫存區接收 Byte 資料，並儲存到 Byte Array。
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 函式庫管理員 (Library Manager) 並查詢各個函式的使用說明。



12.3.1 使用 COM Port 來傳送一個字串

開啟此專案 ("demo_com_port1.zip")，於變數區可查看變數 (參考第 12 章)。

ST 程式:

用來透過 PAC 的 COM 1 ('9600, N, 8, 1') 每 2 秒送出一個字串 (例如: <CNT1=1>...<CNT1=100>)。

(* 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;

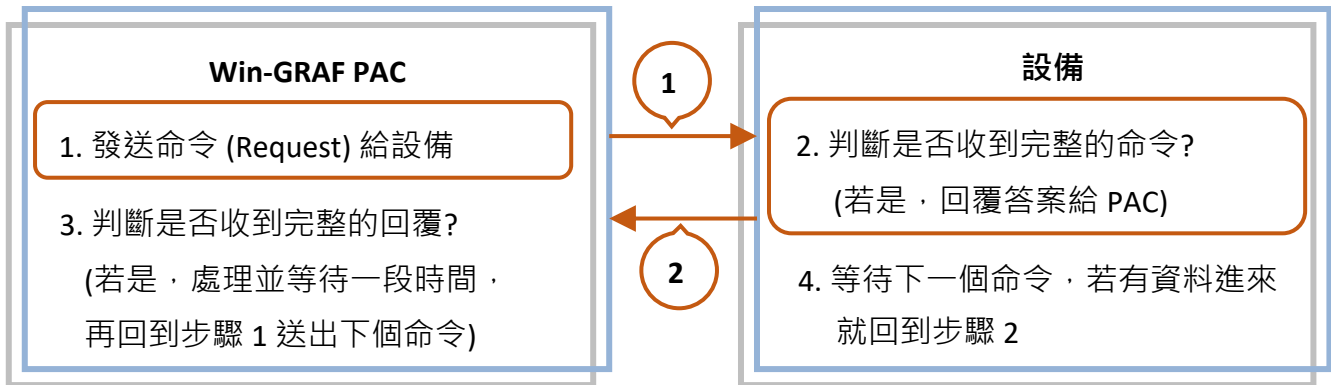
end_if;

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"。

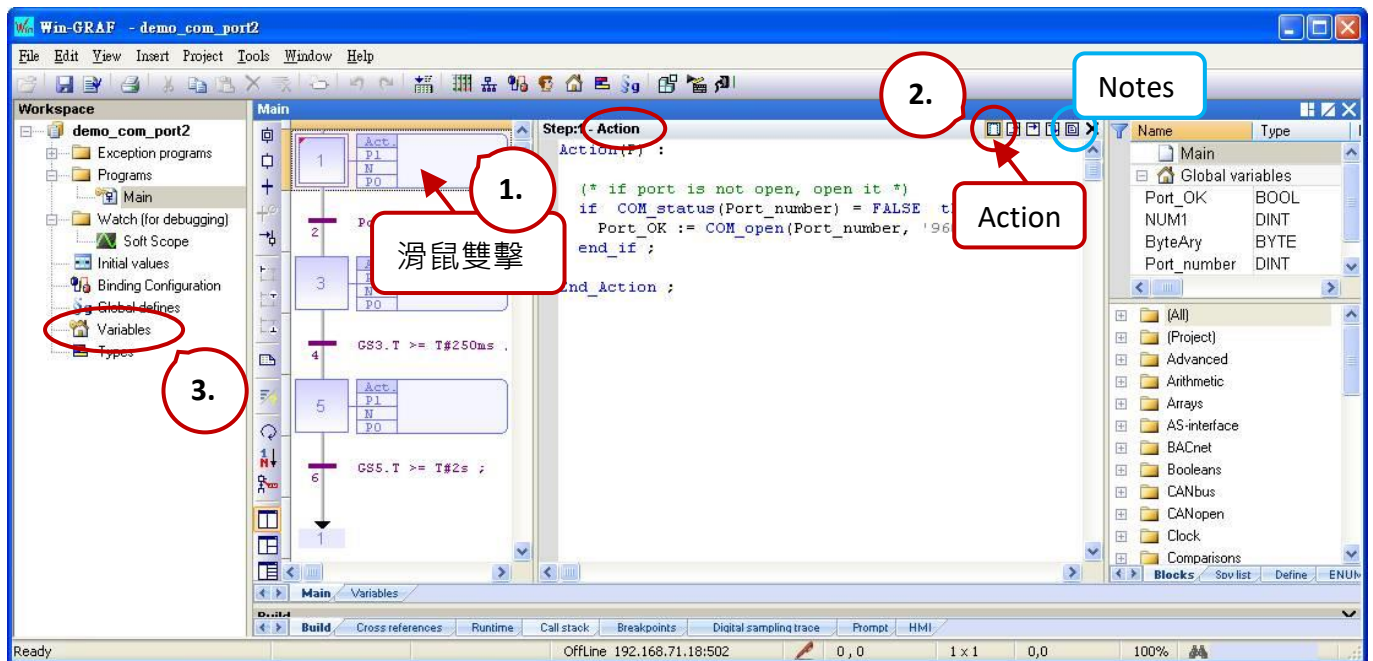
12.3.2 使用 COM Port 對設備一問一答

如需透過 RS-232/RS-485/RS-422 Port 來取得其它設備的資料，其一問一答的方式如下：



開啟此專案 ("demo_com_port2.zip")，於變數區可查看變數。

1. 滑鼠雙擊 "Action" 會開啟 "Notes" 視窗，請切換到 "Action" 視窗來查看程式碼。



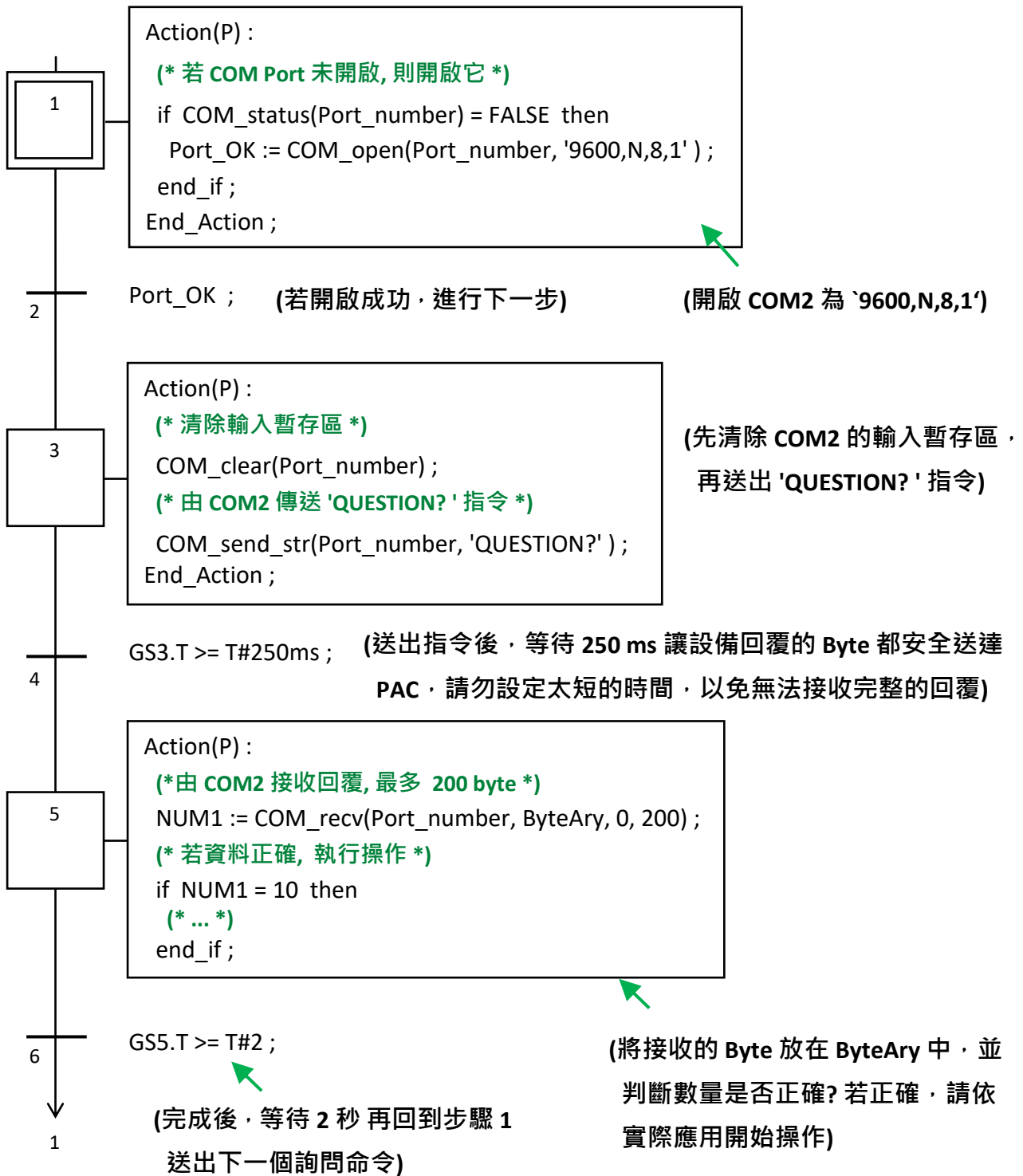
2. 您也可點選 "Variables" 開啟變數視窗。

Name	Type	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description
Global variables								
Port_OK	BOOL			<input type="checkbox"/>				
Port_number	DINT			<input type="checkbox"/>	2			init as 2
NUM1	DINT			<input type="checkbox"/>				
ByteAry	BYTE	[0..199]		<input type="checkbox"/>				Byte Array, DIM=200

此例，會透過 Win-GRAF PAC 的 COM2 送出一個 `QUESTION?` 字串給設備，並等待完整的回覆後再做處理，完成後等待 2 秒，再送出下一個命令，並重複進行。

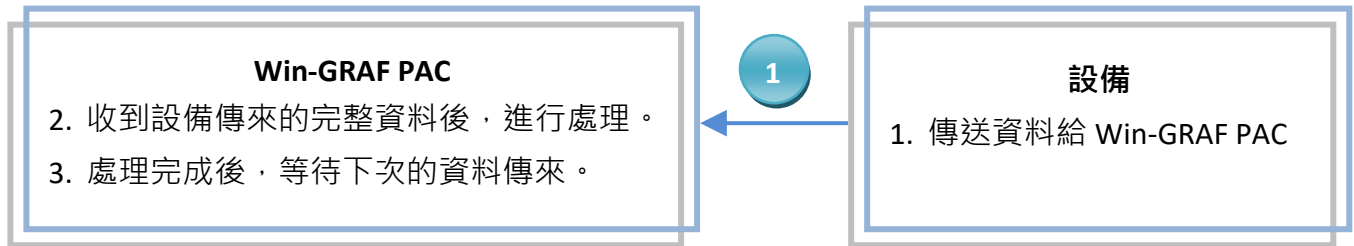
SFC 程式:

(變數 "Port_OK" 為 BOOL; "NUM1" 為 DINT; "ByteAry" 為 BYTE 且 Dim. 為 "200";
 "Port_number" 為 DINT 且初值 為 "2"。)



12.3.3 等待遠端設備傳送資料到 COM Port

此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。當讀取到商品條碼後，它會傳送資料到 Win-GRAF PAC 的 COM Port (RS-232/485/422)，並且不需回覆任何訊息。



開啟此專案 ("demo_com_port3.zip")，於變數區可查看變數 (參考第 12 章)。

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 "2" 。

STEP1 = 0，表示等待中並測試 COM2 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，表示正傳入資料，等待 250 ms 將所有接收的資料放到陣列中。此等待時間與設備規格或 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;
```

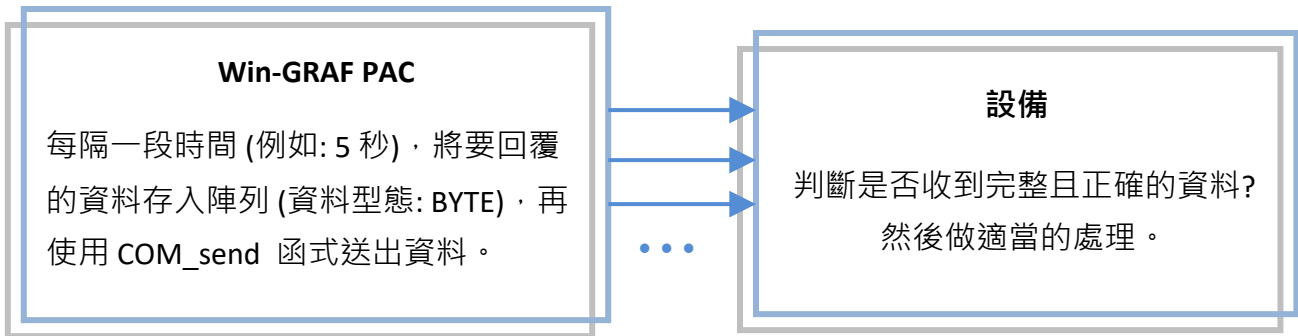
收到資料後，判斷資料對不對？再依實際應用開始操作。

```
end_if;
```

```
END_CASE;
```

12.3.4 使用 COM Port 定期回覆資料給遠端設備

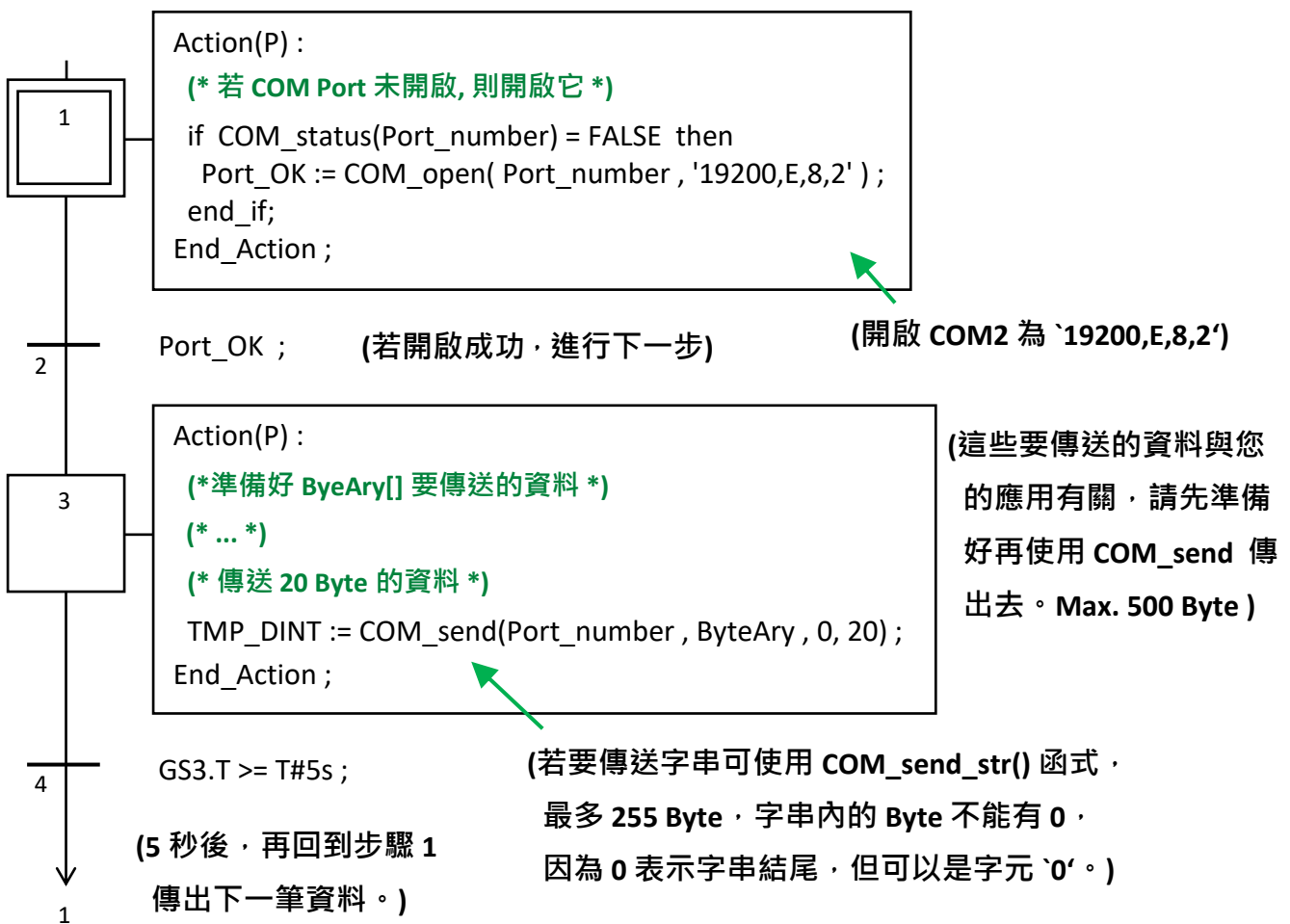
如需定期透過 RS-232/RS-485/RS-422 Port 回覆資料給其它的設備，方法如下：



開啟此專案 ("demo_com_port4.zip") , 於變數區可查看變數 (參考第 12 章)。

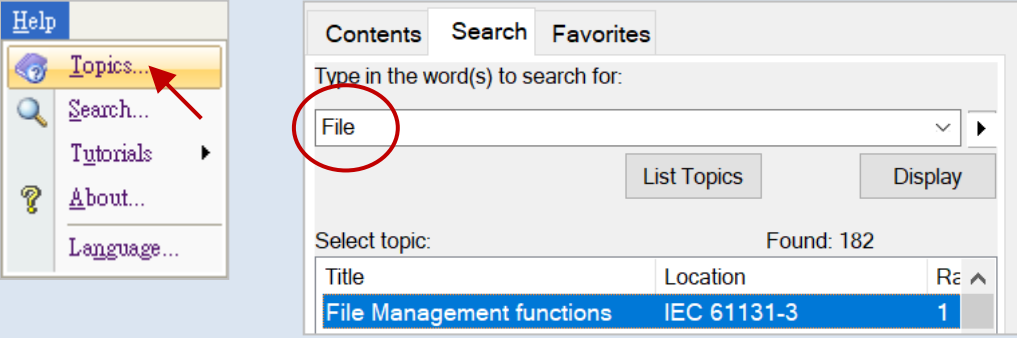
SFC 程式:

(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 內的檔案進行讀/寫操作。請開啟 "HTML 說明" 視窗並搜尋關鍵字 "File" 並選擇標題 "File Management functions" 取得更多資訊。

函式名稱	說明
	
F_OPEN	開啟檔案，以供讀取
F_WOPEN	建立 或 清空檔案，並開啟以供寫入
F_AOPEN	以添加 (append) 模式，來建立或開啟檔案
F_CLOSE	關閉一個開啟中的檔案
F_EOF	測試已開啟供讀取的檔案中，是否到達檔案末尾
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 來查看以下 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 其內所有檔案到另一個目錄
F_del_dir	刪除一個目錄 與 其內的所有檔案

註: Linux PAC 不支援 F_SAVERETAIN 與 F_LOADERRETAIN 函式。

12.4.1 寫入資料到 PAC 內的檔案

開啟此專案 ("demo_file1.zip")，於變數區可查看變數 (參考第 12 章)。

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 *)
```

(* 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
```

由於 /System_Disk/ 的容量較小，您可將檔案路徑改為 /mnt/microSD/，同時 PAC 需有安裝 SD Card。

```

Tmp_string := Any_to_string( REAL_val[ii] );
FM_write( File_ID , Tmp_string );
end_for ;

```

(* 關閉檔案 *)

```
F_close( File_ID );
```

```

end_if ;
end_if ;

```

若想寫入**整數**資料，可修改為：

```

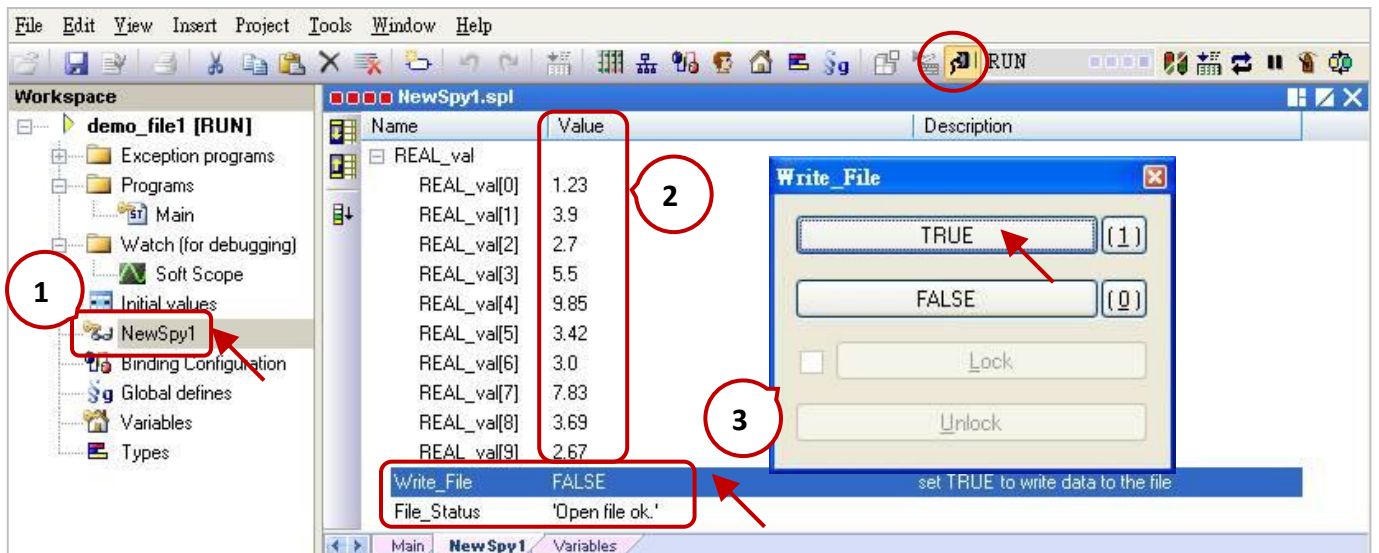
Tmp_string := Any_to_string( DINT_val[ii] );
且宣告 "DINT_val" 為 DINT 且 Dim. 至少為 "10" 。

```

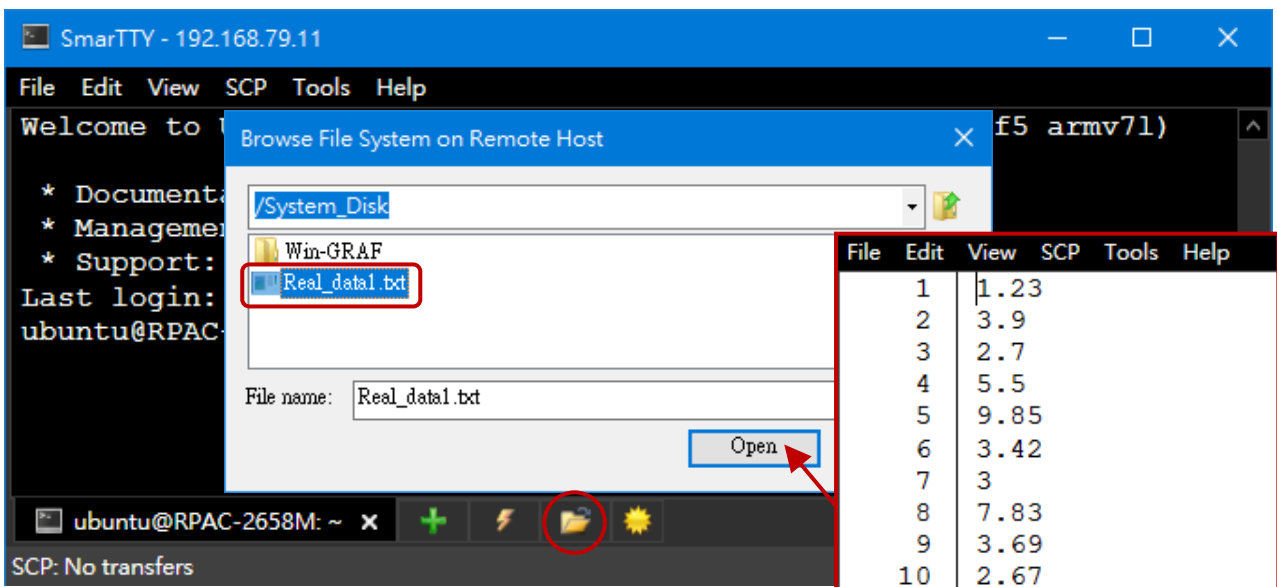
測試程式:

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. 將專案下載到 PAC 後，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")



2. 可透過 SSH 連線 (參考 13.2 節) 查看 PAC 中的 "Real_data1.txt" 檔案。



12.4.2 讀取 PAC 內的檔案資料

開啟此專案 ("demo_file2.zip")，於變數區可查看變數 (參考第 12 章)。

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 +"' !' ;

else

由於 /System_Disk/ 的容量較小，您可將檔案路徑改為 /mnt/microSD/，同時 PAC 需有安裝 SD Card。

(* 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:

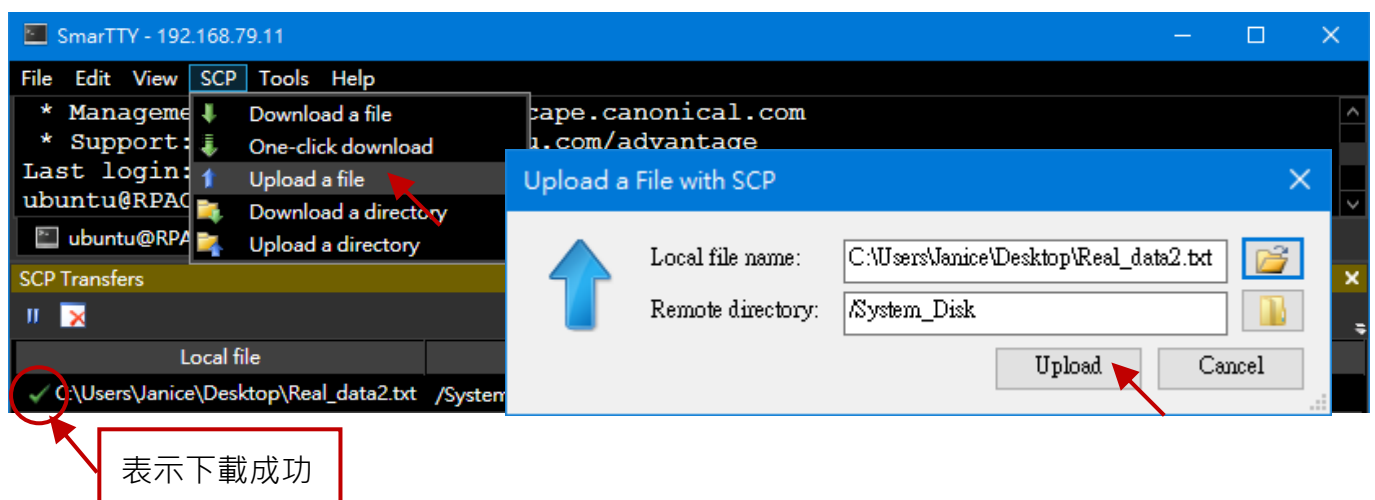
```
DINT_val[ii] := Any_to_DINT( Tmp_string );
```

且宣告 "DINT_val" 為 DINT 且 Dim. 至少為 "10" 。

測試程式:

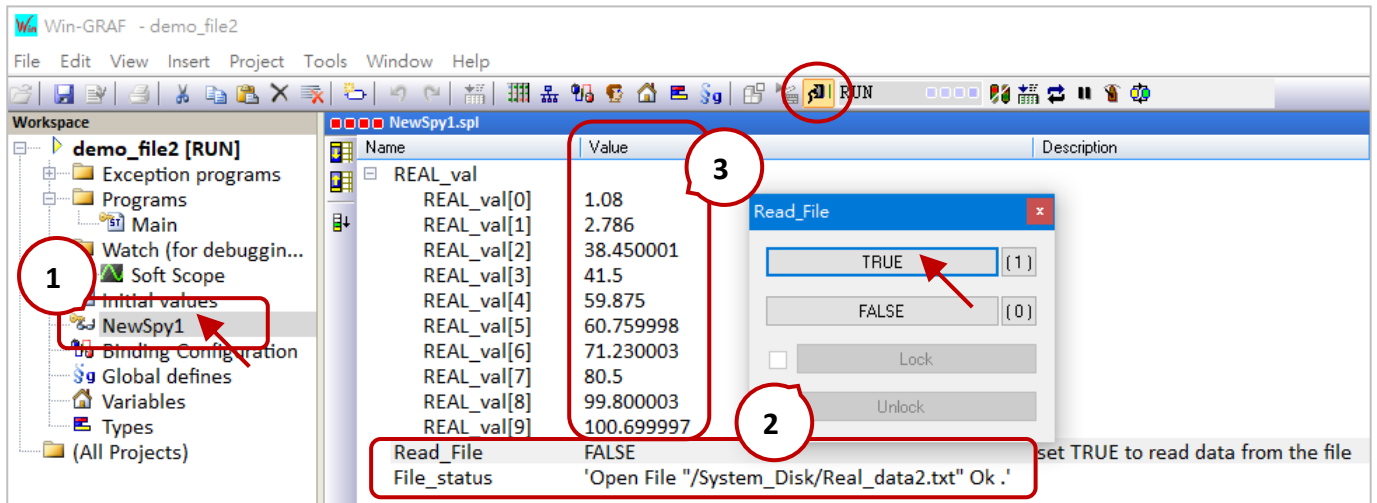
此範例當 "Read_File" 設定為 "TRUE" 時, 會去讀取 PAC 內 `"/System_Disk/Real_data2.txt"` 檔案。

1. 使用 SSH 工具, 將 `Real_data2.txt` 下載到 `/System_Disk/` 資料夾內。(可參考 13.2 節)



2. 下載專案 ("demo_file2.zip") 到 PAC 後，點選 "NewSpy1" 開啟觀測清單，再將 "Read_File" 設定為 "TRUE"，來讀取資料。

若 OK，會顯示讀到的資料且 "File_Status" 會顯示 'Open File "/System_Disk/Real_data2.txt" Ok.'



註:

- 使用 Linux PAC 時，路徑符號為 / (斜線)。
- 您也可參考 6.2 節，使用保存變數將多筆資料寫入檔案中。

12.4.3 資料紀錄功能

請參考第 12 章開啟此專案 ("demo_datalog.zip")，此程式為簡易的資料記錄功能 (Data Logging)。

範例說明:

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 個變數資料，紀錄在 PAC 的指定目錄下。

該記錄檔會依照日期來命名 (例如: 2021 年 1 月 2 日，檔名就是 "2021-1-2.csv")，另外，每日會將前一天的紀錄檔移到 "/指定目錄/當月日期/" 的資料夾底下 (例如: 2021 年 1 月 1 日，會放在 "/mnt/microSD/2021-1/")。

註: 本範例的紀錄檔是存在 PAC 上的 /mnt/microSD/，PAC 需有安裝 SD Card。

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
```

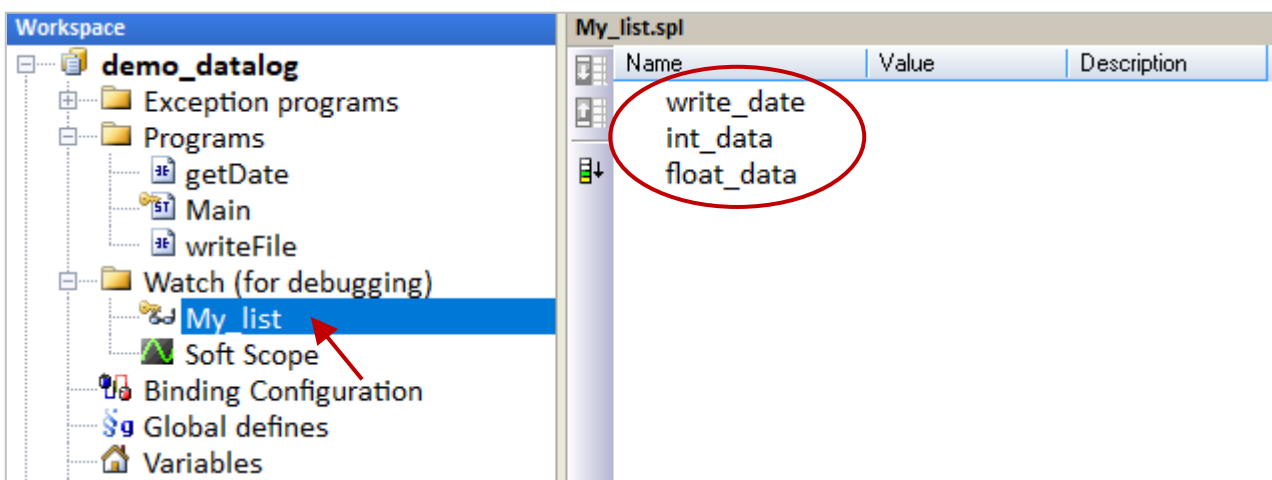
變數說明: 您可在變數區 (Variables) 查看/建立變數。

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		
Day1		
WeekDay1		
Hour1		
Minute1		
Second1		
old_day		
old_month	TIME	Timer。
log_tmr1		
log_tmr2	STRING	CSV 檔的儲存路徑。
CSV_Path		CSV 檔的儲存目錄。
CSV_Dir		用於紀錄每筆資料寫入 CSV 檔的時間。
write_date	BOOL	用於初始化，預設為 TRUE。
init		

變數名稱	資料型態	說明
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 "F_WOPEN" 函數。
tmp_bval	BOOL	暫存變數。
tmp_msg	STRING	

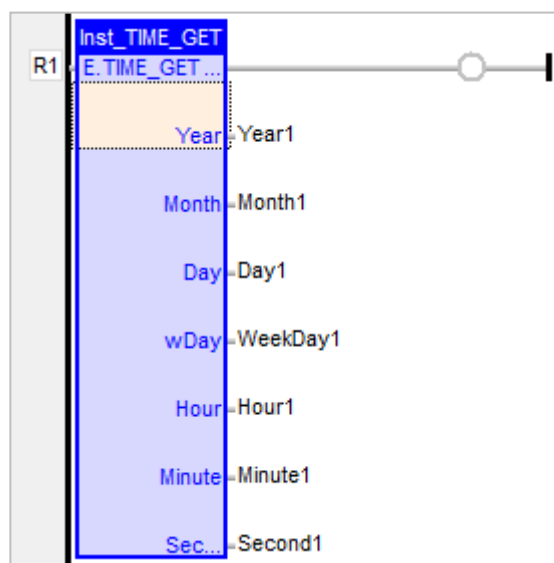
觀測清單 (Spy List) – “My_List”:

此範例建立了一個名為 “My_list” 的觀測清單 (可參考 11.3 節，了解詳細設定)。



LD 程式 – “getDate”:

用來取得/設定 PAC 的系統時間。



註: 此範例程式的檔案目錄是 `/mnt/microSD/`，PAC 需有安裝 SD Card。

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:='/mnt/microSD/'+CSV_Dir+'/' +ANY_TO_STRING(year1)+'-'+
    ANY_TO_STRING(month1)+'-'+ANY_TO_STRING(old_day)+'.csv';

  (* 複製檔案到 /mnt/microSD/ *)
  tmp_bval:=F_COPY (CSV_Path,tmp_msg);
  (* 刪除原本在 /run/ 的檔案 *)
  tmp_bval:=F_DELETE(CSV_Path);
  (* 建立 csv 檔案與其欄位名稱 *)
  tmp_msg:='/run/'+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;
```

(*每週一個月 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:='/mnt/microSD/'+CSV_Dir;
    F_DIR(tmp_msg);
```

```
end_if;
```

(* 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:='/mnt/microSD/'+CSV_Dir+'/' +ANY_TO_STRING(Year1) +'-'+
    ANY_TO_STRING(month1)+'-'+ANY_TO_STRING(Day1)+'.csv';
```

由於 /System_Disk/ 的容量較小，您可將檔案路徑改為 /mnt/microSD/，同時 PAC 需有安裝 SD Card。

(* 刪除在 /mnt/microSD/ 的檔案 *)

```
tmp_bval:=F_DELETE(tmp_msg);
```

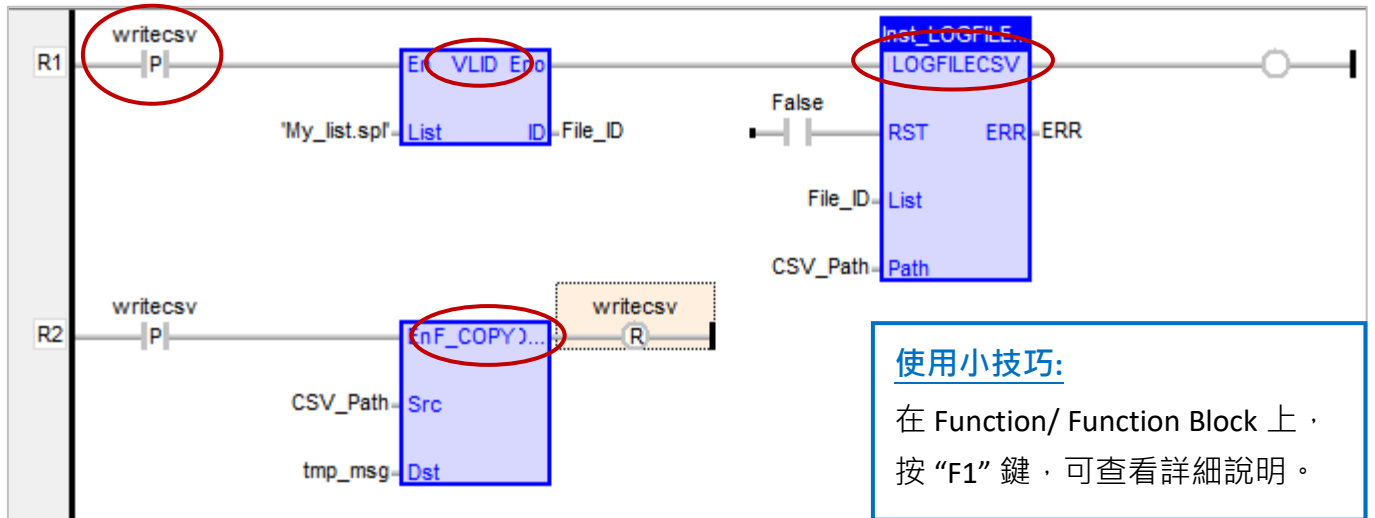
(* 觸發檔案寫入 *)

```
writcsv:=true;
```

```
END_IF;
```

LD 程式 – “WriteFile”

用來觸發並寫入一筆資料到 .CSV 檔案。



VLID: 取得觀測清單 (Spy List: **My_list.spl**) 的值。

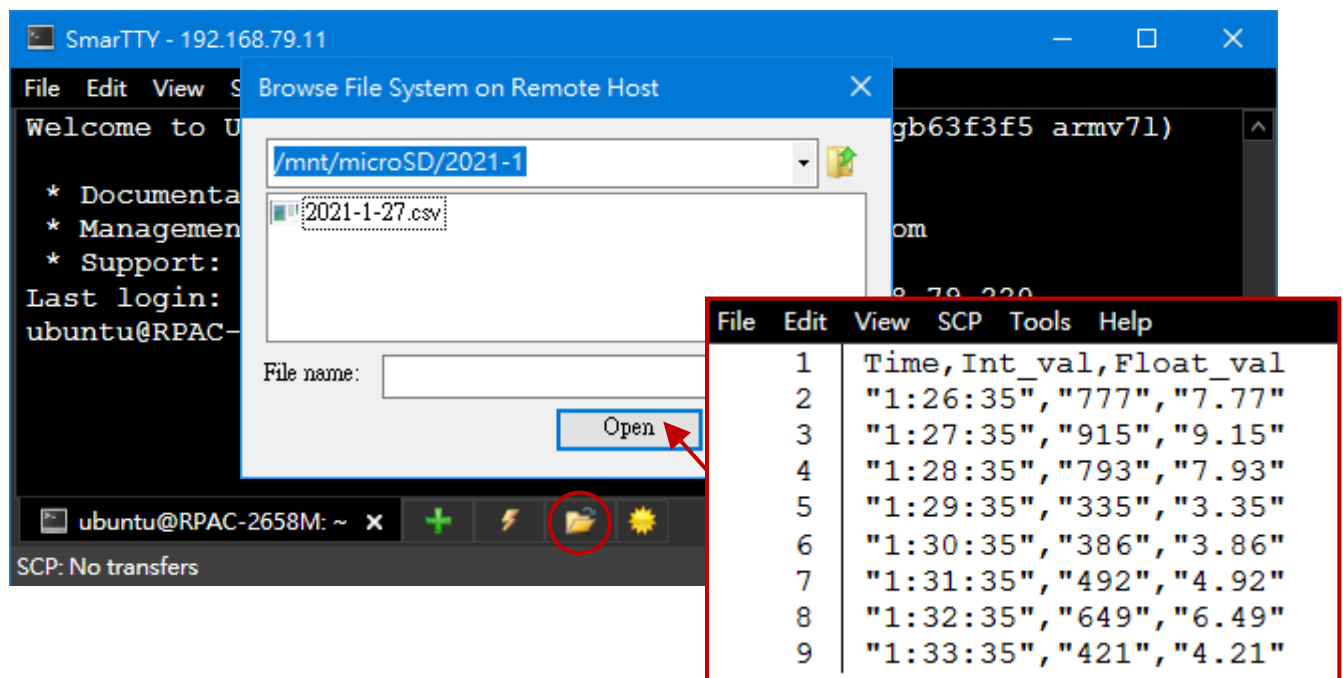
LogFileCSV: 將清單內的變數值存成 csv 格式的記錄檔。

F_COPY: 複製一個檔案。使用 “F_COPY” 將原先儲存在 “/run/” (RAM Disk, 讀寫速度較快) 的檔案複製到 “/mnt/microSD/”。(參考 ST 程式 – “Main”)

測試程式:

此例, 每隔 1 分鐘會將 write_date、int_data 與 float_data 變數資料, 紀錄並儲存在 /mnt/microSD/YYYY-MM/YYYY-MM-DD.csv (例如: /mnt/microSD/2021-1/2021-1-27.csv)。

您可使用 SSH 工具, 找到 PAC 內的 csv 檔案並開啟查看資料。(可參考 13.2 節)



第 13 章 使用 C 程式來讀/寫 Win-GRAF 變數

本章節說明如何在 Win-GRAF 專案內開放變數，讓 C 程式讀/寫資料。此 C 範例程式是以 GCC 開發工具建立的，另外提供了 2 個 Win-GRAF 範例程式。

C 範例程式 (Quicker_Demo.tgz):

用來讀/寫 WinGRAF Char、Short、Long、ULong、Int64、Float、Double、String 變數。

Win-GRAF 範例程式 (demo_vb03.zip 與 demo_vb04.zip):

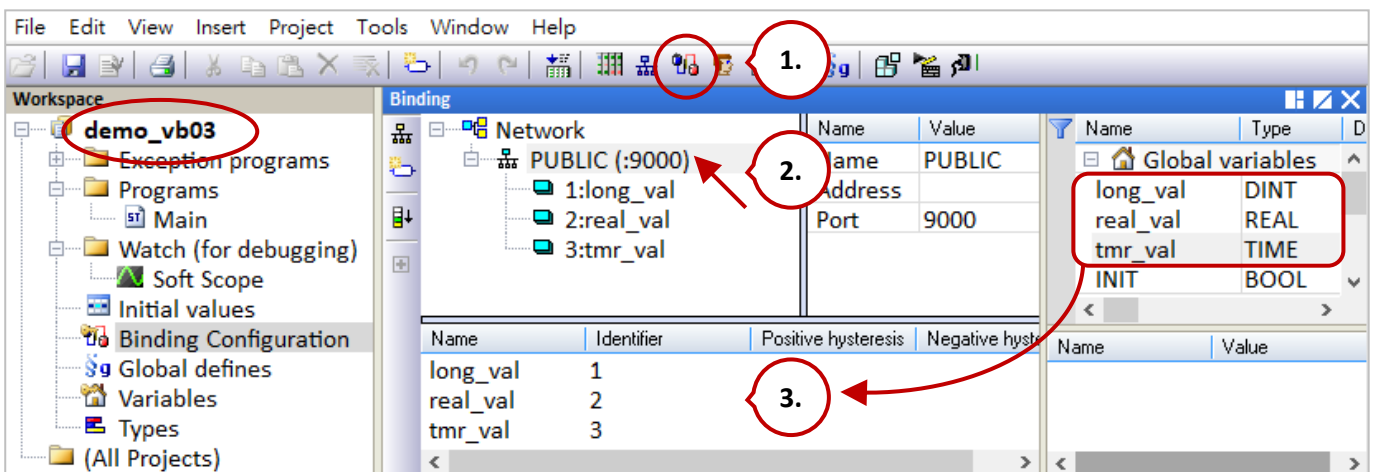
請在網站下載 [範例程式](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。

13.1 開放 Win-GRAF 變數給 C 程式使用

Win-GRAF 中，除了 String 變數之外，皆需在 "Binding" 視窗中設定好要開放的變數與 ID，這樣 C 程式才能使用一樣的位址來存取資料。

請參考 [第 7 章](#) 來設定要開放的變數，設定完成後，畫面如下。

注意: "PUBLIC" 最多可使用 8192 個變數，"Identifier" 編號只能是 "1 ~ 8192"。



ST 程式 (Main):

(* 在第一個 scan cycle 執行 *)

if INIT then

INIT := False; (* 第一個 scan cycle 後，不再執行 *)

tstart(tmr_val);

end_if;

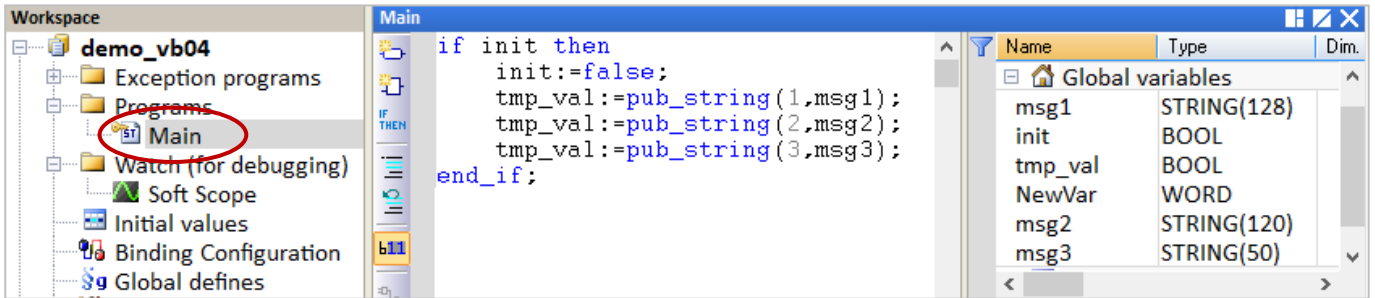
另外，如需開放 String 變數，可在 Win-GRAF 專案的 ST 程式裡，使用 "Pub_String" 函式。

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。

您可回存 [範例程式 \(demo_vb04.zip\)](#)，來查看以下內容。



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 Binding 設定是否成功。(True: 成功，False: 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	

ST 程式 (Main):

```

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;

```

13.2 下載與編譯 C 程式

請依照以下步驟來下載並編譯 C 程式:

1. 請先下載 SSH 工具，像是 **SmartTTY** (有支援 SCP 指令)。

<https://sysprogs.com/SmartTTY/download/>

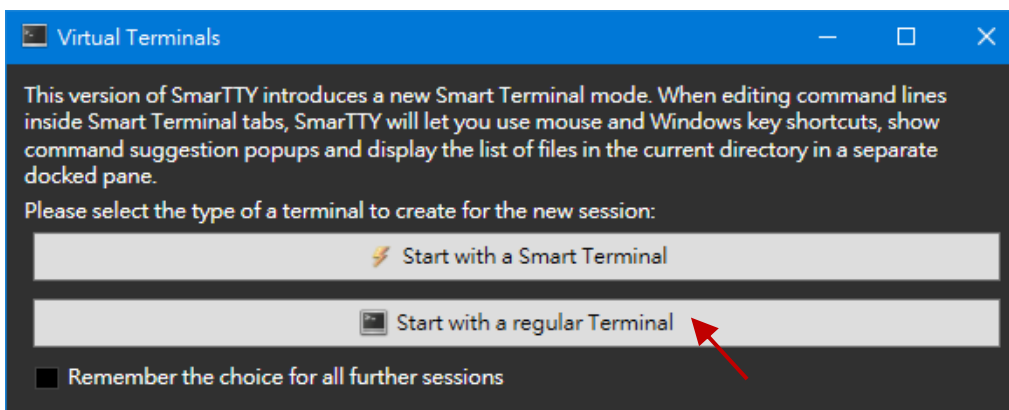
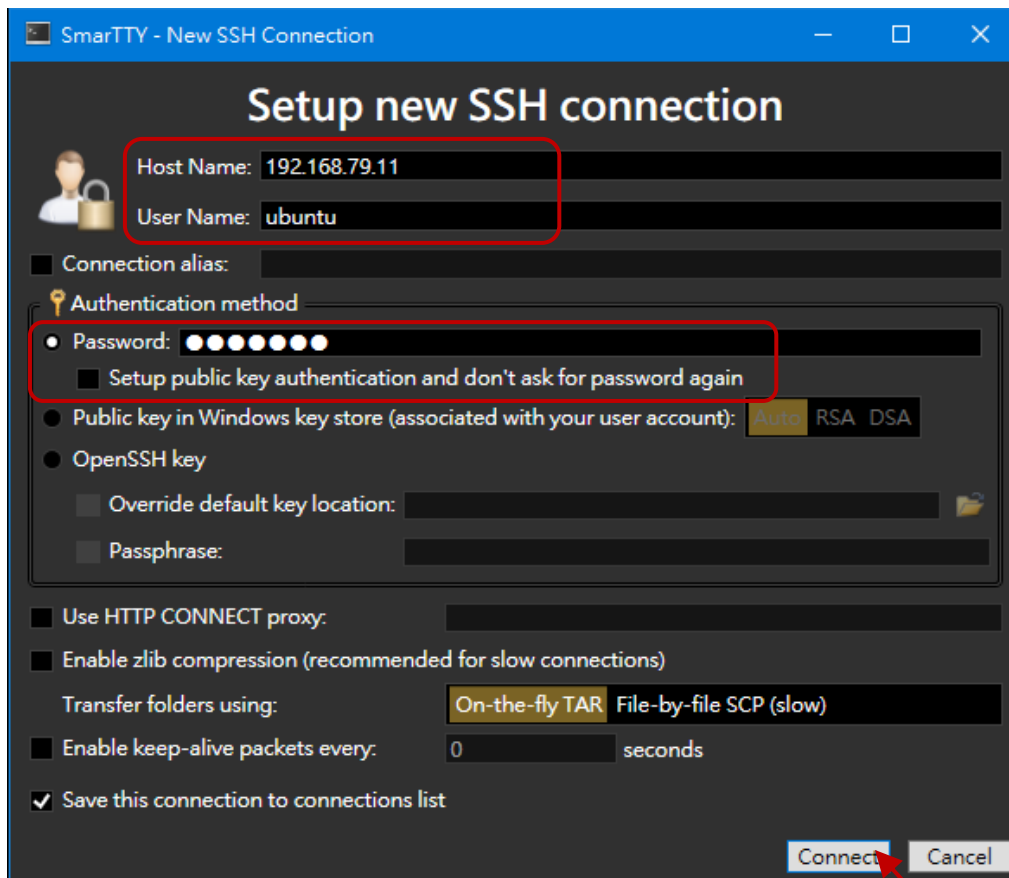
2. 使用 SmartTTY 與 PAC 連線。

執行 SmartTTY 並設定新的 SSH 連線。

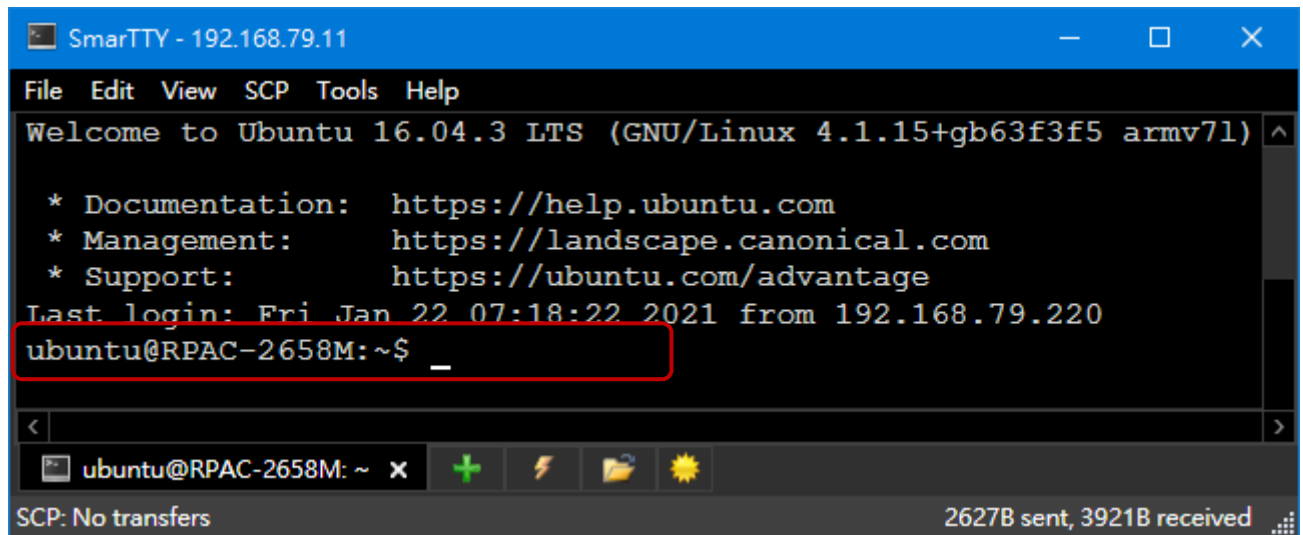
Host name: 輸入 PAC 的 IP 位址。(LAN1 預設 IP 為 '192.168.255.1')

User name: 輸入 'ubuntu'。

Password: 輸入 'temppwd'。



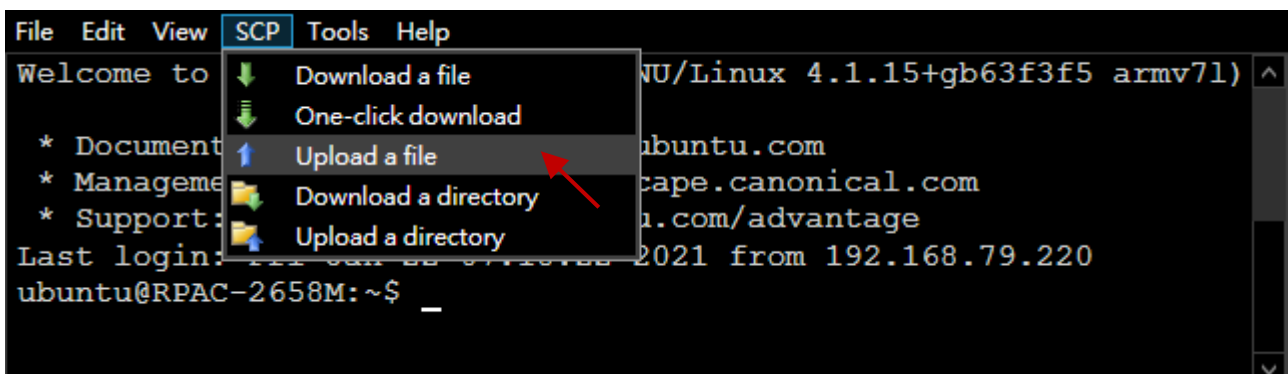
若連線成功，將會顯示以下訊息。



註：第一次登入 PAC，會要求修改密碼。

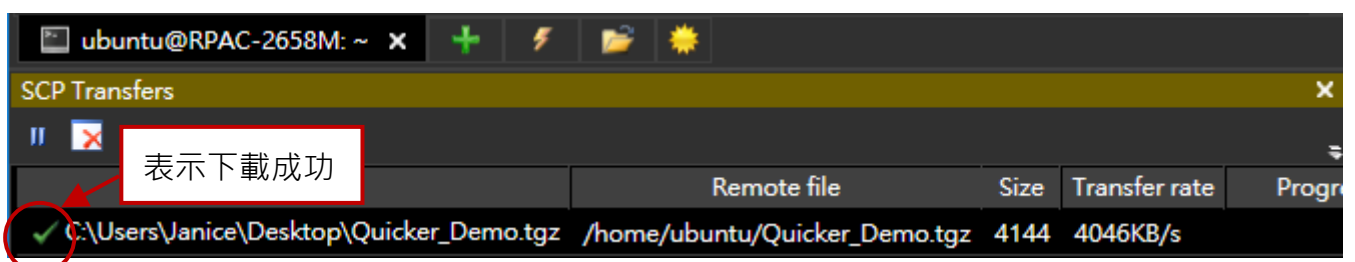
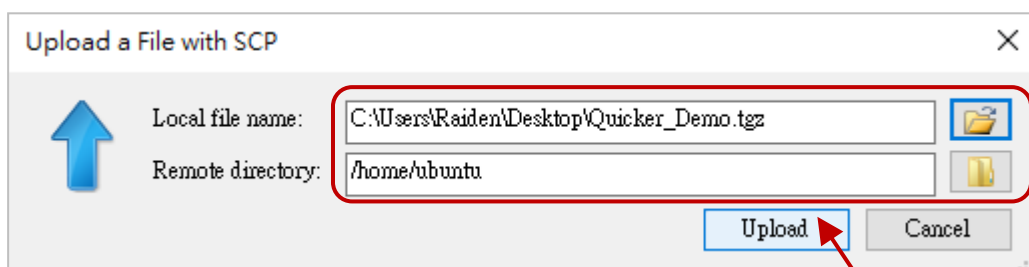
3. 下載 **Quicker_Demo.tgz** 檔案到 PAC。

1) 點選功能表“SCP - Upload a file”命令。



註：LAN1, LAN2 與 LAN3 需設定為不同的網段，否則可能會造成路由錯誤，無法下載檔案。

2) 將 **Quicker_Demo.tgz** 檔案下載到 **/home/Ubuntu** 目錄。



4. 解壓縮 Quicker_Demo.tgz 壓縮檔。

使用 'tar -zxvf ./Quicker_Demo.tgz' 指令，將檔案解壓縮到 Quicker_Demo 目錄。

```
SmarTTY - 192.168.79.11
File Edit View SCP Tools Help
ubuntu@RPAC-2658M:~$ tar -zxvf ./Quicker_Demo.tgz
./Quicker_Demo/
./Quicker_Demo/Makefile
./Quicker_Demo/Quicker_demo.c
./Quicker_Demo/inc/
./Quicker_Demo/inc/Quicker.h
./Quicker_Demo/lib/
./Quicker_Demo/lib/Quicker.a
ubuntu@RPAC-2658M:~$ _
```

5. 執行 make 指令來編譯範例程式。

- 1) 輸入 'cd Quicker_Demo/' 指令進入該目錄。
- 2) 輸入 'make' 指令編譯當前目錄中的程式。

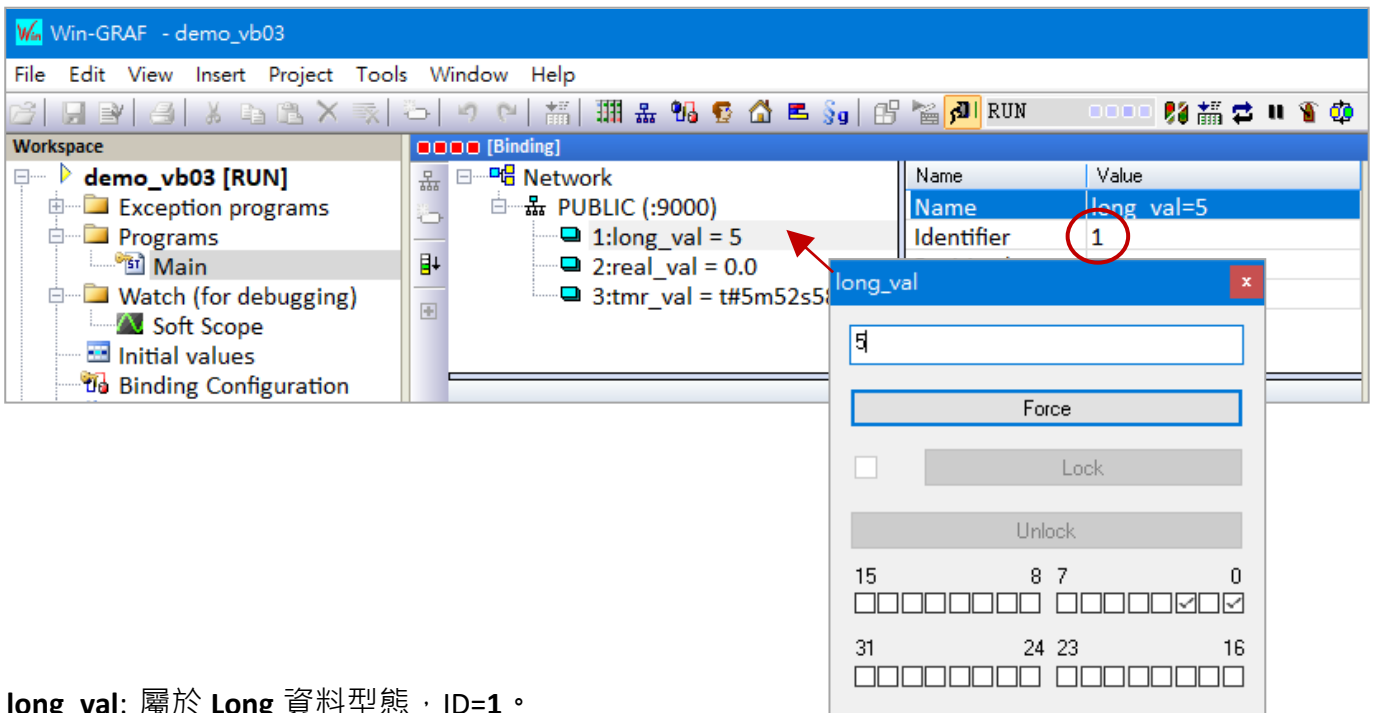
```
ubuntu@RPAC-2658M:~$ cd Quicker_Demo/
ubuntu@RPAC-2658M:~/Quicker_Demo$ make
gcc -I./ -I./inc -o Quicker_demo Quicker_demo.c ./lib/Quick
er.a -lm ./lib/Quicker.a -lm -lpthread
ubuntu@RPAC-2658M:~/Quicker_Demo$ ls
inc lib Makefile Quicker_demo Quicker_demo.c
```

6. 執行範例程式來存取 Win-GRAF 變數。

```
ubuntu@RPAC-2658M:~/Quicker_Demo$ ./Quicker_demo
=====
Usage : ./Quicker_demo Set/Get Char/Short/Long/ULong/Int64/Float/
Ex : ./Quicker_demo Set Char 1 0
Ex : ./Quicker_demo Get Char 1
Ex : ./Quicker_demo Set Str 1 hello
Ex : ./Quicker_demo Get Str 1
=====
ubuntu@RPAC-2658M:~/Quicker Demo$ ./Quicker_demo Set Char 1 0
Set_Char: ret=0, addr=1, value=0
ubuntu@RPAC-2658M:~/Quicker Demo$ ./Quicker_demo Get Char 1
Get_Char: ret=0, addr=1, value=0
ubuntu@RPAC-2658M:~/Quicker_Demo$
```

13.3 存取 Win-GRAF 變數

在 Win-GRAF 專案 - **demo_vb03**，共有三個 Public 變數，分別為 **long_val**、**real_val** 與 **tmr_val**。



long_val: 屬於 Long 資料型態，ID=1。

- 1) 滑鼠雙擊 Binding 視窗內的 **long_val** 可輸入數值 (例如: 5)。
- 2) 輸入 **./Quicker_Demo Get Long 1** 指令來讀取資料。

```
ubuntu@RPAC-2658M:~/Quicker_Demo$ ./Quicker_demo Get Long 1
Get_Long: ret=0, addr=1, value=5
ubuntu@RPAC-2658M:~/Quicker_Demo$ _
```

real_val: 屬於 Float 資料型態，ID=2。

- 1) 輸入 **./Quicker_Demo Set Float 2 1.1** 指令來寫入資料。
- 2) 輸入 **./Quicker_Demo Get Float 2** 指令來讀取資料。

```
ubuntu@RPAC-2658M:~/Quicker_Demo$ ./Quicker_demo Set Float 2 1.1
Set_Float: ret=0, addr=2, value=1.100000
ubuntu@RPAC-2658M:~/Quicker_Demo$ ./Quicker_demo Get Float 2
Get_Float: ret=0, addr=2, value=1.100000
ubuntu@RPAC-2658M:~/Quicker_Demo$ _
```

tmr_val: 屬於 Ulong 資料型態，ID=3。輸入 **./Quicker_Demo Get ULong 3** 指令來讀取資料。

Timer 的單位為 ms，因此數值為 1529275 (ms) = 25 m 29 s 275 ms。

```
Get_ULong: ret=0, addr=3, value=1529275
ubuntu@RPAC-2658M:~/Quicker_Demo$ _
```

13.4 "Quicker.a" 內的函式說明

本章節將介紹 "Quicker.a" 靜態函式庫內的 Function 說明。

"Quicker.a" 提供了許多函式，可用來讀/寫 Win-GRAF 變數資料，如下：

1. 讀/寫 Boolean
2. 讀/寫 8-bit、16-bit、32-bit 或 64-bit 整數
3. 讀/寫 32-bit 或 64-bit 實數

13.4.1 讀/寫 Boolean 變數

■ Set_Coil

說明：

設定特定位址編號的 Win-GRAF Boolean 變數值。

語法：

```
unsigned char UserShare_Set_Coil(unsigned short iUserAddress, unsigned char iStatus)
```

參數：

iUserAddress：Win-GRAF 變數的位址編號 (1 ~ 8192)

iStatus：設定 Boolean 變數值 (1: "True" ; 0: "False")

範例：

‘設定位址編號 "1" 的 Boolean 變數值為 "True"’

```
UserShare_Set_Coil(1, 1)
```

■ Get_Coil

說明：

取得特定位址的 Win-GRAF Boolean 變數值。

語法：

```
unsigned char UserShare_Get_Coil(unsigned short iUserAddress, unsigned char *iStatus)
```

參數：

iUserAddress：Win-GRAF 變數的位址編號 (1 ~ 8192)

iStatus：取得 Boolean 變數值 (1: "True" ; 0: "False")

範例：

‘取得位址編號 "1" 的 Win-GRAF Boolean 變數值’

```
unsigned char value;
```

```
UserShare_Get_Coil(1, &value);
```

13.4.2 讀/寫整數變數

■ Set_Char ■ Set_Short ■ Set_Long ■ Set_Int64
■ Set_UChar ■ Set_UShort ■ Set_ULong ■ Set_UInt64

說明:

設定特定位址編號的 Win-GRAF 8-bit, 16-bit, 32-bit 或 64-bit Integer 變數值。

語法:

```
unsigned char UserShare_Set_Char (unsigned short iUserAddress, char iStatus)  
unsigned char UserShare_Set_Short (unsigned short iUserAddress, short iStatus)  
unsigned char UserShare_Set_Long (unsigned short iUserAddress, long iStatus)  
unsigned char UserShare_Set_Int64 (unsigned short iUserAddress, long long iStatus)  
unsigned char UserShare_Set_ULong (unsigned short iUserAddress, unsigned long iStatus)  
unsigned char UserShare_Set_UChar (unsigned short iUserAddress, unsigned char iStatus)  
unsigned char UserShare_Set_UShort (unsigned short iUserAddress, unsigned short iStatus)  
unsigned char UserShare_Set_UInt64 (unsigned short iUserAddress, unsigned long long iStatus)
```

參數:

iUserAddress : Win-GRAF 變數的位址編號 (1 ~ 8192)

iStatus : 設定 8-bit, 16-bit, 32-bit 或 64-bit 的 Integer 值

範例:

‘設定位址編號 “1” 的 32-bit integer 變數值為 “1234567”

UserShare_Set_Long(1, 1234567)

‘設定位址編號 “2” 的 32-bit integer 變數值為 “-1234”

UserShare_Set_Long(2, -1234)

‘設定位址編號 “3” 的 64-bit integer 變數值為 “123456789012345”

UserShare_Set_Int64(3, 123456789012345)

‘設定位址編號 “4” 的 8-bit integer 變數值為 “125”

UserShare_Set_Char(4, 125)

■ Get_Char ■ Get_Short ■ Get_Long ■ Get_Int64
■ Get_UChar ■ Get_UShort ■ Get_ULong ■ Get_UInt64

說明:

取得特定位址編號的 Win-GRAF 8-bit, 16-bit, 32-bit 或 64-bit Integer 變數值。

語法:

```
unsigned char UserShare_Get_Char (unsigned short iUserAddress, char *iStatus)  
unsigned char UserShare_Get_Short (unsigned short iUserAddress, short *iStatus)  
unsigned char UserShare_Get_Long (unsigned short iUserAddress, long *iStatus)  
unsigned char UserShare_Get_Int64 (unsigned short iUserAddress, long long *iStatus)  
unsigned char UserShare_Get_UChar (unsigned short iUserAddress, unsigned char *iStatus)  
unsigned char UserShare_Get_UShort (unsigned short iUserAddress, unsigned short *iStatus)  
unsigned char UserShare_Get_ULong (unsigned short iUserAddress, unsigned long *iStatus)  
unsigned char UserShare_Get_UInt64 (unsigned short iUserAddress, unsigned long long *iStatus)
```

參數:

iUserAddress: Win-GRAF 變數的位址編號 (1 ~ 8192)

iStatus: 取得 8-bit, 16-bit, 32-bit 或 64-bit 的 Integer 值

範例:

long long Dlong_val;

long long_val;

short short_val;

char sbyte_val;

‘取得位址編號 “7” 的 64-bit integer 變數值

UserShare_Get_Int64(7, &Dlong_val);

‘取得位址編號 “8” 的 32-bit integer 變數值

UserShare_Get_Long(8, &long_val);

‘取得位址編號 “9” 的 16-bit integer 變數值

UserShare_Get_Short(9, &short_val);

‘取得位址編號 “10” 的 8-bit integer 變數值

UserShare_Get_Char(10, &sbyte_val);

13.4.3 讀/寫實數變數

■ Set_Float ■ Set_Double

說明:

設定特定位址的 Win-GRAF 32-bit 或 64-bit REAL 變數值。

語法:

```
unsigned char UserShare_Set_Float(unsigned short iUserAddress, float iStatus)
unsigned char UserShare_Set_Double(unsigned short 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:

‘設定位址編號 “7” 的 64-bit REAL 變數值為 “11234.234567”

UserShare_Set_Double(7, 11234.234567)

‘設定位址編號 “8” 的 32-bit REAL 變數值為 “123.12”

UserShare_Set_Float(8, 123.12);

■ Get_Float ■ Get_Double

說明:

取得特定位址編號的 Win-GRAF 32-bit 或 64-bit REAL 變數值。

語法:

```
unsigned char UserShare_Get_Float(unsigned short iUserAddress, float *iStatus)
unsigned char UserShare_Get_Double(unsigned short iUserAddress, double *iStatus)
```

參數:

iUserAddress : Win-GRAF 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_Double(7, &double_val);

‘取得位址編號 “8” 的 float 變數值

UserShare_Get_Float(8, &float_val);

13.4.4 讀/寫字串變數

■ Set_STRING

說明:

設定特定位址編號的 Win-GRAF String 資料。

語法:

```
int UserSetReg_Str(unsigned short iUserAddress, char iStatus, unsigned short len)
```

參數:

iUserAddress : Win-GRAF 變數的位址 (1 ~ 1024)

iStatus : 取得/設定 Win-GRAF 的字串指標

len: 設定 Win-GRAF 的字串長度

範例:

```
char str_value[256];  
//設定位址編號“7”的 String 資料  
sprintf(str_value, "hello world \n");  
UserSetReg_Str(7, str_value, strlen(str_value));
```

■ Get_STRING

說明:

取得指定位址編號的 Win-GRAF String 資料。

語法:

```
int UserGetReg_Str(unsigned short iUserAddress, char *iStatus, unsigned short len);
```

參數:

iUserAddress : 指定變數的位址編號 (1 ~ 1024)

iStatus : 取得/設定 Win-GRAF 的字串指標

len: 取得 Win-GRAF 的字串長度

範例:

```
char str_value[256];  
//取得位址編號“7”的 String 變數值  
UserGetReg_Str(7, str_value, 256);
```

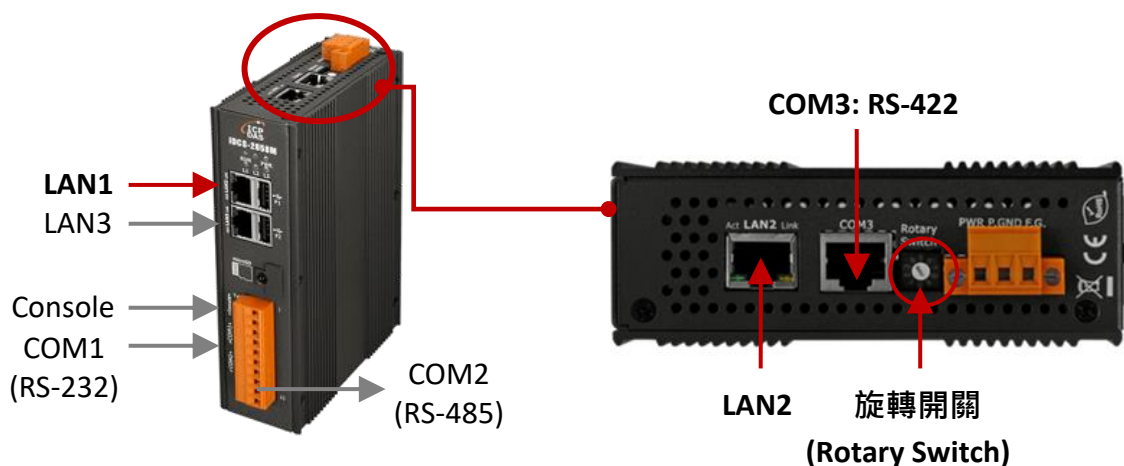

第 14 章 備援 (冗餘) 系統

ICP DAS 的 **Win-GRAF Linux PAC - RPAC-2658M** 有支援備援 (冗餘) 系統:

Win-GRAF PAC 備援是使用 2 台 PAC 來達成，當運行程式的 PAC 喪失系統控制能力時，會經由切換機制自動切換到備援 PAC，或由 Win-GRAF 程式判斷於特定條件下時進行切換。

名稱定義	說明
Main-PAC	旋轉開關設為 7 的那台 PAC
Backup-PAC	旋轉開關設為 9 的那台 PAC
Active-PAC	正在運行程式的那台 PAC
Passive PAC	正在備援資料的那台 PAC

註: Main-PAC 一開始為 Active-PAC，會依照狀況自動切換主控權到另一台 PAC。



注意:

1. 備援系統內的 PAC 不可同時為 Main PAC 或 Backup PAC。
2. 兩台 PAC 之間，需有以下 3 個 通訊 Port 可互相通訊。

通訊 Port	說明
A. Public IP Port	LAN1 ，用來跟 Win-GRAF、圖控 (SCADA) 或 HMI 進行通訊。 註: 若需連接 Ethernet I/O 或設備，建議使用 LAN3。
B. Replication Port	LAN2 ，資料同步通道。 2 台 PAC 使用 Ethernet 跳線對接，互相高速傳遞備援資料，中間不可接任何 Ethernet Switch 或 Hub。
C. Alive Port (或 Heart-beat Port)	COM3 (RS-422) ，用來偵測兩台 PAC 是否正常運作。 使用 RJ-45 接頭，2 台 PAC 可使用 Ethernet 跳線對接。

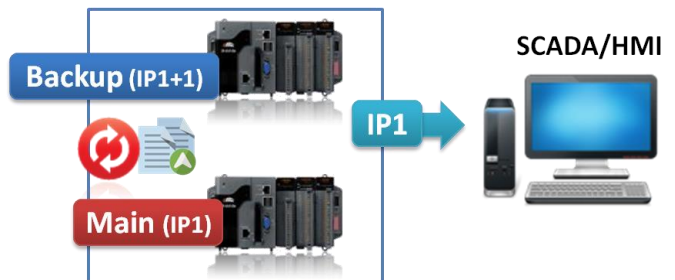
Win-GRAF 備援系統的特色

1. 安全性佳:

備援系統主要由 LAN1、LAN2 與 COM3 (RS-422) 進行通訊，當 Active-PAC 偵測到軟硬體或通訊異常，則會觸發切換機制，由另一台 PAC 進行系統控制。即使任 2 條 Cable 中斷通訊，仍可繼續正常運作程式。

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:

若備援系統發生故障，操作人員可將故障的 PAC 單獨斷電拆下並換上備品，不需重新燒錄程式。正常的 PAC 會自動把 Win-GRAF 專案與備援資料，全都傳給剛上線的那台 PAC。

注意:

- 正常運作的那台 PAC，不可斷電或拆下，需讓它繼續運作。
- 備品需調整好旋轉開關、接好通訊線或 I/O 模組，才開機。

例外:

若 User 在備援系統內除了 Win-GRAF 專案之外，還有 Run 其它程式，例如: C 應用程式或 HMI 專案，則需在上線運作前，把這些檔案安裝到備品 RPAC (或修好的 RPAC) 內。

4. 簡化程式設計:

User 只需設計好程式，不需額外設計哪些資料需傳送到另一台 PAC，Win-GRAF 備援系統會自動把備援資料備份到另一台 PAC。

5. 自訂安全機制:

User 可在程式內制定安全機制，例如: 若主控 PAC 的 LAN1 斷線，而無法連上 SCADA，或是某個 RS-485 Port 斷線/故障時，將自動重開機並切換控制權到另一台 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。

14.1 哪些資料會自動備援到 Passive PAC?

Win-GRAF 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. Win-GRAF 應用程式 (專案)。
2. 變數值。
3. 功能方塊的 FB instance 裡的內部 (Private) 資料。
4. PAC 硬體的 RTC (Real Time Clock) 時間。
5. 可保存記憶體 (Retain Memory)。

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 內的一些檔案。
(例如: /System_Disk 或 /mnt/microSD 內的檔案，或其它非屬於 Win-GRAF 專案的檔案，像是 C 或 HMI 等應用程式檔。因此，更換 PAC 到備援系統之前，需確認這些檔案已安裝)。
3. 使用者自行使用 COM_OPEN () 函式開啟的序列埠 (Serial Port)，不會自動在 Passive-PAC 開啟。
4. 儲存在 PAC 的 FRAM 資料不會自動備援。

不會自動備援的資料，使用者仍可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 ;
```

14.2 備援系統 (旋轉開關 7 & 9)

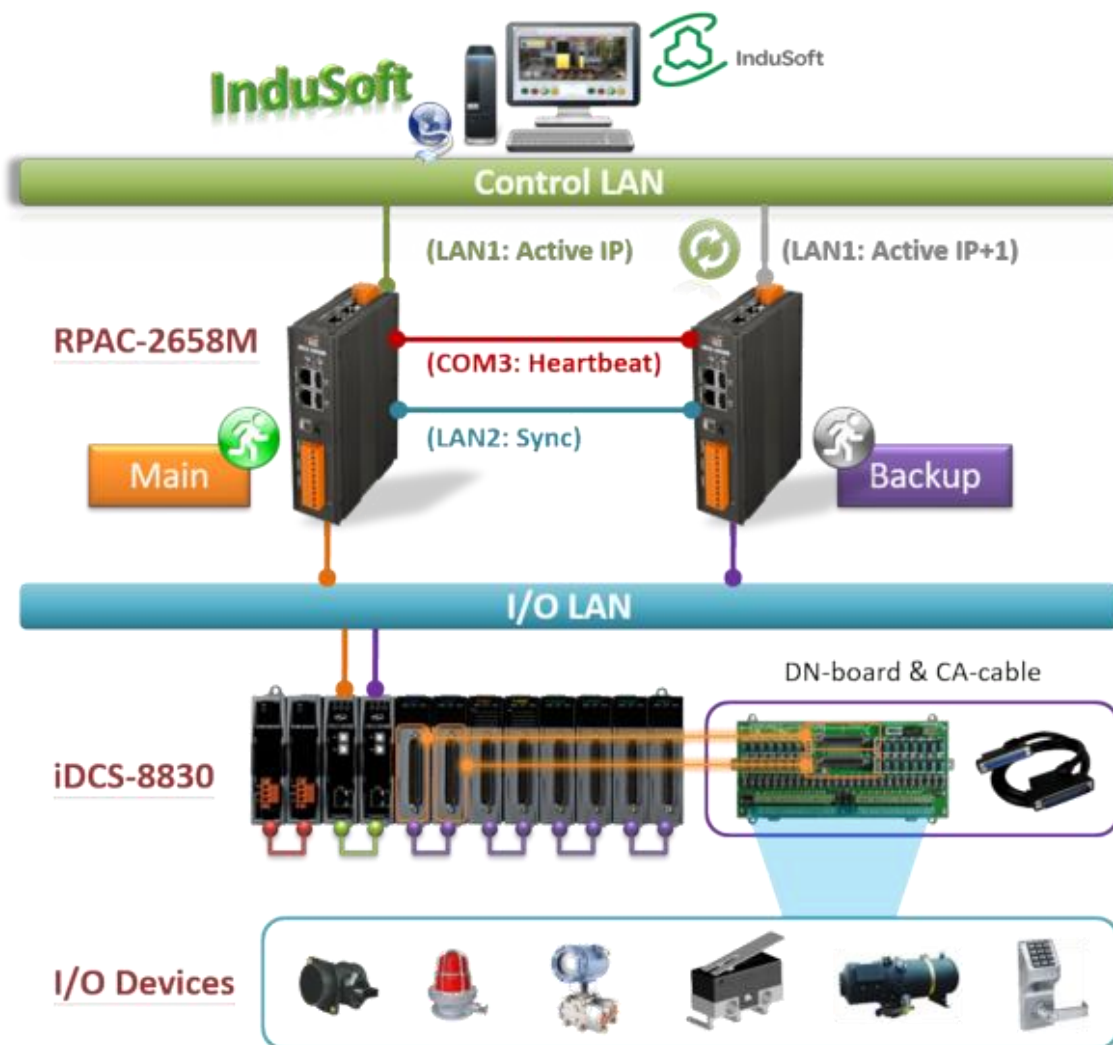
請設置旋轉開關 與 連接線如下：

硬體		
2 台 RPAC	旋轉開關 (Rotary Switch)	—台為 "7" (稱為 Main-PAC) —台為 "9" (稱為 Backup-PAC)
	LAN1	2 x 一般網路線。
	LAN2	1 x Ethernet 跳線。
	COM3 (RS-422)	1 x RS-422 跳線，也可使用 Ethernet 跳線。

14.2.1 Win-GRAF 備援架構

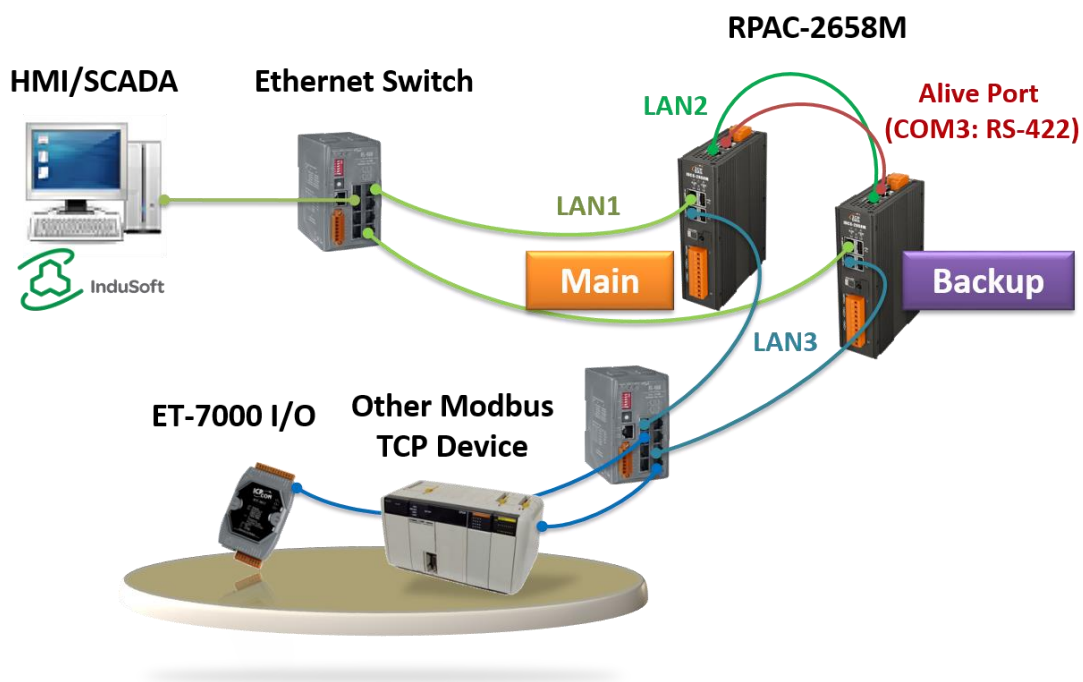
1. 備援系統搭配 iDCS-8830 I/O 模組：

iDCS-8830 Ethernet 擴充單元 + 備援 I/O 模組。

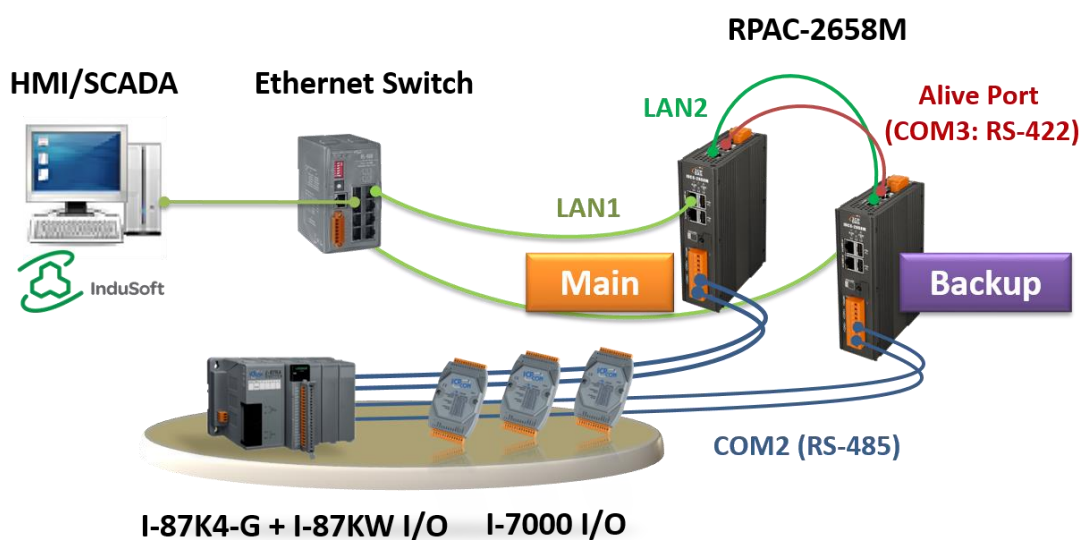


註： 每台 iDCS-8830 使用的備援 I/O 模組，是 2 個相同型號為一組的模組。

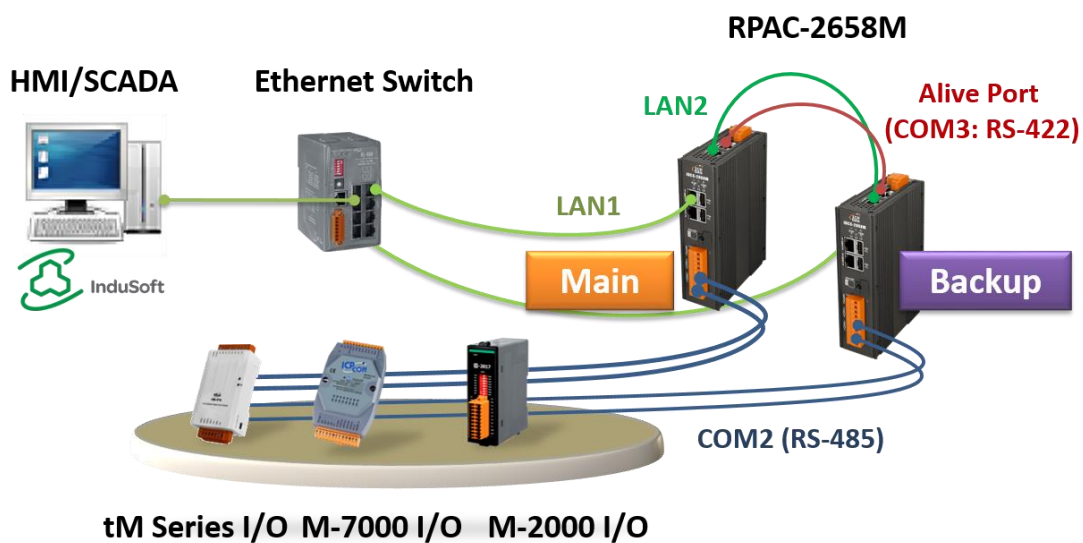
2. 備援系統搭配 Modbus TCP I/O 模組:



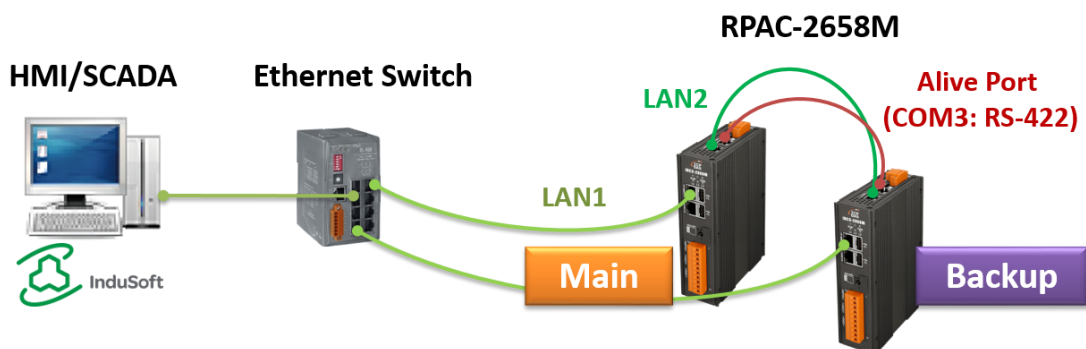
3. 備援系統搭配 DCON I/O 模組:



4. 備援系統搭配 其它 Modbus RTU/ASCII I/O 模組:

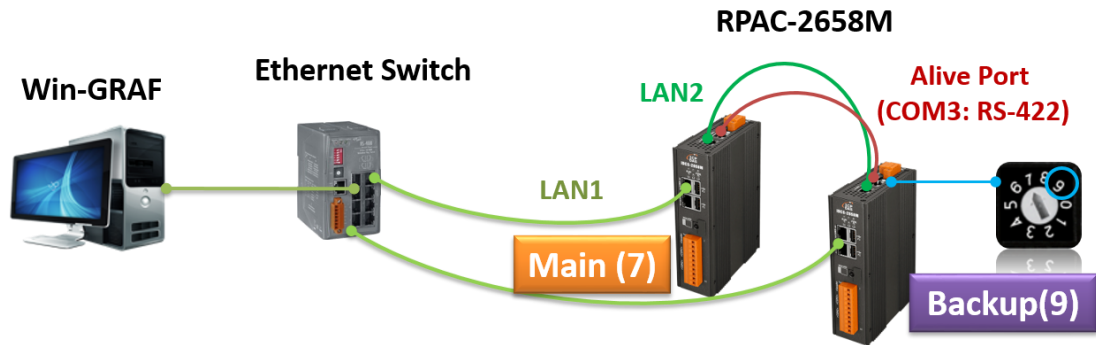


5. 備援系統不含 I/O 模組:



以上所連接的 I/O 模組也可混合搭配 2 種 (或更多) 來使用。

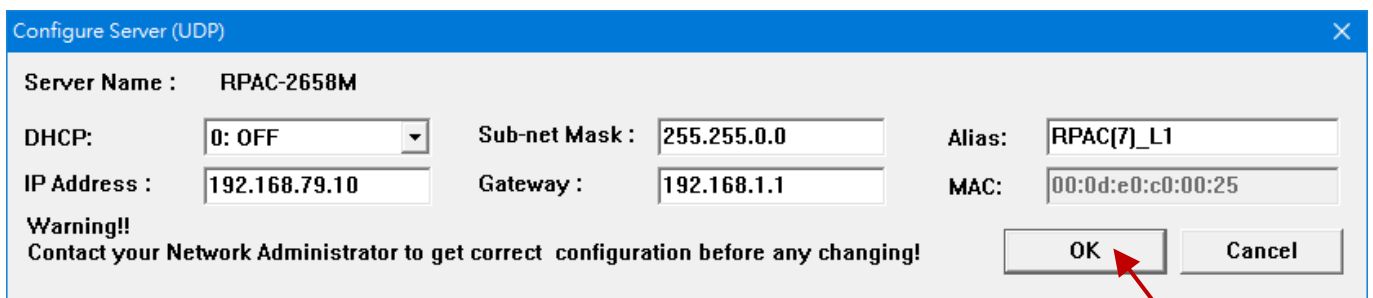
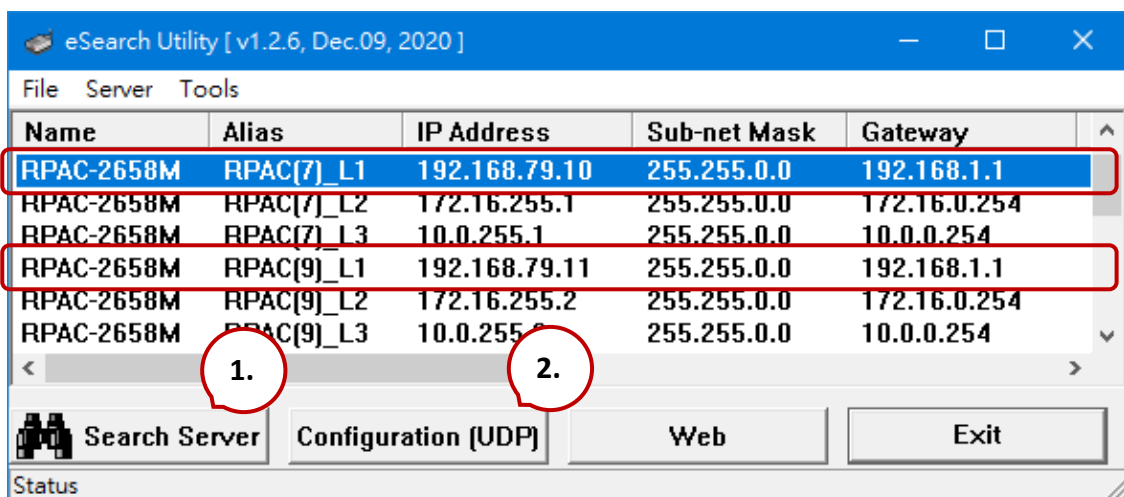
14.2.2 設定備援系統



- 步驟 1:** 兩台 PAC 以一般模式開機 (旋轉開關 = 0) , 並設定 LAN1 IP 位址。
- 步驟 2:** 兩台 PAC 各自接好 3 條通訊線 , 再以備援模式開機 (旋轉開關 = 7 & 9) 。
- 步驟 3:** 修改備援程式的通訊 IP 與 Active IP , 再下載到 Main PAC (7) 。
- 下載成功後 , 專案會透過 LAN2 自動備份到 Backup PAC (9) 。

步驟 1: 設定 LAN1 IP 位址

- 兩台 PAC 的 Rotary Switch 都轉到 0 並開機。
- 打開 eSearch Utility , 點選 Search Server 按鈕搜尋設備 , 再點選 Configuration 按鈕 , 設定 PAC 的 LAN1 IP 位址 (參考 1.3 節 , PC/Win-GRAF 與 PAC 需設定在相同網段 , 才可進行連線) 。



注意: LAN1 IP 與 Win-GRAF Active_IP 不可設為相同 , 以避免 PAC 啟動後發生 IP 衝突 。

步驟 2: 設置通訊線 與 備援模式 (旋轉開關 = 7 & 9)

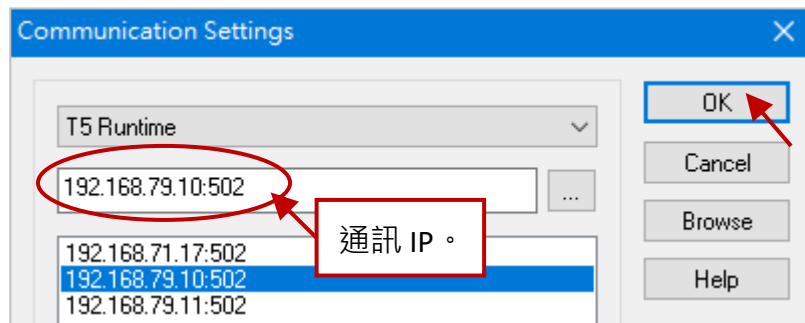
- A. 將兩台 PAC 的 LAN1 連接上 Ethernet Switch。
- B. 將兩台 PAC 的 LAN2 以跳線互接。
- C. 將兩台 PAC 的 COM3 (RS-422) 以跳線互接。
- D. 將兩台 PAC 的 Rotary Switch 轉到 7 (稱為 Main-PAC) 與 9 (稱為 Backup-PAC)。
- E. 將兩台 PAC 重新開機。

註: 備援模式時，LAN2 會自動設為 199.193.195.17 / 199.193.195.9。

步驟 3: 下載備援程式

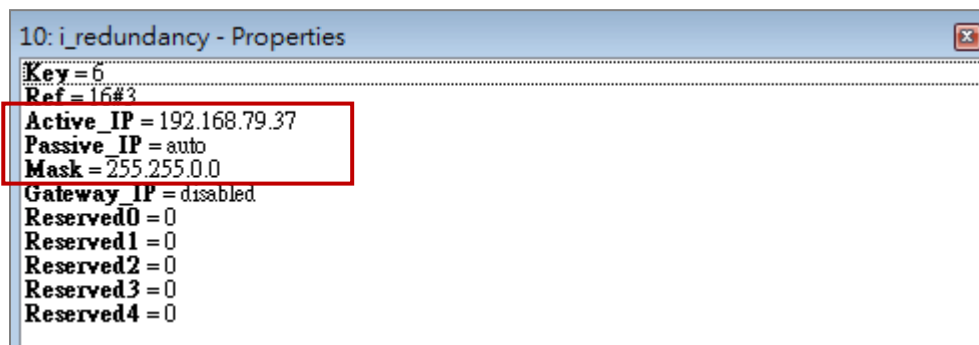
1. 設定 Win-GRAF 與 PAC 通訊的 IP

首次下載備援程式 (例如: demo_rdn_2)，需將 Win-GRAF 的通訊 IP，設為 Main PAC (7) 目前的 LAN1 IP。(若不熟悉設定，可參考 2.3.4 節)




2. 設定 Active_IP 位址

您可依照網路環境來設定 I/O Boards – i_redundancy 的 Active_IP 與 Mask。
(若不熟悉設定，可參考第 4 章)



3. 下載 Win-GRAF 專案

點選 “On Line” 按鈕 () 連線，將備援程式下載到 Main PAC (7) 中。下載後，PAC 的 LAN1 IP 會自動設成 Active_IP (此例: 192.168.79.37)，而 Backup PAC (9) 的 LAN1 IP 會自動設成 Active IP + 1 (此例: 192.168.79.38)。

4. 將 Win-GRAF 通訊 IP，修改為 Active_IP

此時，Win-GRAF 會顯示 "Communication error" 訊息，因為 Main PAC 的 IP 已自動設為 Active_IP。
請再次點選 "On Line" 按鈕，先停止 Win-GRAF 與 PAC 之間的連線。

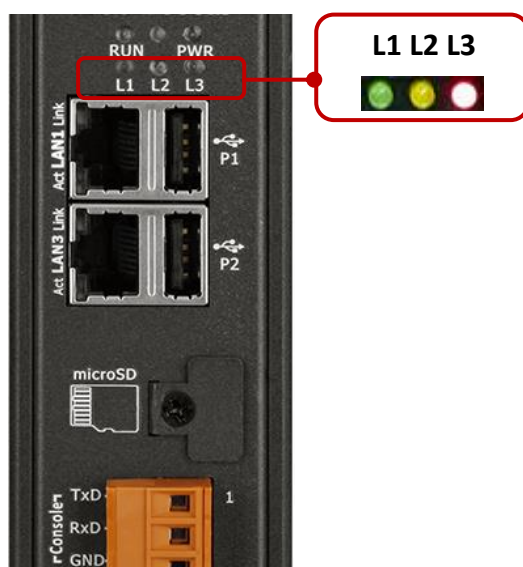


接著，將 Win-GRAF 通訊 IP 修改為 Active_IP，往後如需更新程式，皆會下載到 Active 那台 PAC。



注意：更新 Active-PAC 的程式時，請先將 Passive-PAC 斷電。下載完成後，再將 Passive-PAC 上電，以確保備援系統正常運作。

此外，PAC 上的 L3 亮紅燈表示是主控 PAC。



14.2.3 測試備援程式 (demo_rdn_2)

您可參考 14.4 節 來查看範例介紹。

1. 查看 PAC 的備援狀態。

Win-GRAF 與 PAC 連線後，可點選 "NewSpy1" 開啟觀測清單，並查看 PAC 的備援狀態。

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

目前主控的 (Active) 是 Main-PAC。

- Main-PAC 與 Backup-PAC 目前皆就緒且 LAN1 通訊正常。
- Alive Port (RS-422) 通訊正常。
- Passive PAC 已就緒。
- Active-PAC 與 Passive-PAC 的 LAN1 通訊正常。

Setup as Retain variable in the program "Retain_and_timer"

- 注意:**
- 1) 請確認 **Passive PAC 已就緒** (即，is_Passive_ready = TRUE)，才切換 PAC 的控制權。
 - 2) Win-GRAF 內，也可手動切換 PAC 的控制權，請點選 "Redundancy" 按鈕 (🔧)。

表示此台目前是 Active PAC。

2. 設定變數值。

- 1) 任意設定 "DINT_1"、"DINT_2"、"REAL_1" 與 "REAL_2" 的值。
- 2) 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 與 TMR_2 。

Name	Value	Description
is_Active_LAN1_ok	TRUE	
is Passive LAN1 ok	TRUE	
DINT_1	9	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	22.299999	Setup as Retain variable in the program "Retain_and_timer"
REAL_2	33.5	Setup as Retain variable in the program "Retain_and_timer"
TMR_1	t#1m10s26ms	
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#36s996ms	
TMR_2 last state	TRUE	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

Annotations in the image:

- A red box labeled "任意輸入值" (Arbitrary input value) points to the values of DINT_1, DINT_2, REAL_1, and REAL_2.
- A red box labeled "設定為 'TRUE'" (Set to 'TRUE') points to the values of To_tick_TMR_1 and To_tick_TMR_2.

3. 測試備援功能。(請參考 "RDN_control" 程式)

- 1) 確認 Passive PAC 已就緒 (即，is_Passive_Ready 為 TRUE)。
- 2) 將 Active PAC 的 LAN1 拔除，此時主控權會立即交給另一台 PAC，此範例程式中 (RDN_control)，PAC 會偵測 LAN1 斷線一段時間後，自動執行重開機程序。
- 3) 開機後，主控為 Backup PAC，可看到變數值仍存在且 Timer 仍計時中。

此時主控為 Backup PAC

Name	Value
is_Main_Active	FALSE
is_Backup_Active	TRUE
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

Name	Value
DINT_1	9
DINT_2	1234
REAL_1	22.299999
REAL_2	33.5
TMR_1	t#29m38s510ms
TMR_1 last state	TRUE
To_tick_TMR_1	FALSE
To_stop_TMR_1	FALSE
TMR_2	t#29m5s480ms
TMR_2 last state	TRUE
To_tick_TMR_2	FALSE
To_stop_TMR_2	FALSE

註: 請將 LAN1 插上，則 "is_Main_LAN1_ok" 與 "is_Passive_LAN1_ok" 將回復為 TRUE。

14.3 範例程式介紹

請在網站下載 [範例程式](#) (demo_rdn_1.zip、demo_rdn_2.zip、demo_rdn_3.zip 與 demo_rdn_4.zip) · 並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。

範例名稱	說明	
demo_rdn_1	使用 2 台 PAC 的 COM2，連接 3 個 DCON I/O 模組。	測試
demo_rdn_2	使用 2 台 PAC，沒有連接任何 I/O 模組。	測試 、 程式
demo_rdn_3	使用 2 台 PAC 的 LAN3，經由 Ethernet Switch 去連接一個 ET-7050 (Modbus TCP I/O 模組)。	測試
demo_rdn_4	使用 2 台 PAC 的 LAN3，經由 Ethernet Switch 去連接一個 iDCS-8830 (可支援備援 I/O 模組)。	測試 、 程式

14.3.1 [重要] "I/O Boards" 設定 (i_redundancy、i_redundancy_rs485)

◇ 相關範例: demo_rdn_1、demo_rdn_2、demo_rdn_3 與 demo_rdn_4。

為了備援系統可正常運作，需在 "I/O Board" 視窗內連上 "i_redundancy" 功能 (可參考第 4 章)，並設定 Active_IP 與 Mask。

The screenshot shows the "I/O Boards" configuration window. A list of boards is shown on the left, with "10 i_redundancy" selected. A "Properties" dialog box for "10: i_redundancy" is open, displaying the following settings:

- Key = 6
- Ref = 15#3
- Active_IP = 192.168.79.37
- Passive_IP = auto
- Mask = 255.255.0.0
- Gateway_IP = disabled
- Reserved0 = 0
- Reserved1 = 0
- Reserved2 = 0
- Reserved3 = 0
- Reserved4 = 0

A red box highlights the "Active_IP" field with the following note:

注意:
Active_IP 的最後一個數字，不可以是 0、254、255，必須是 1 ~ 253。

設定說明 (i_redundancy): 用來表示目前備援系統的狀態。

Active_IP: 此 IP 為備援系統對外公開的 IP 位址，主要給 HMI/SCADA 來連接通訊用。

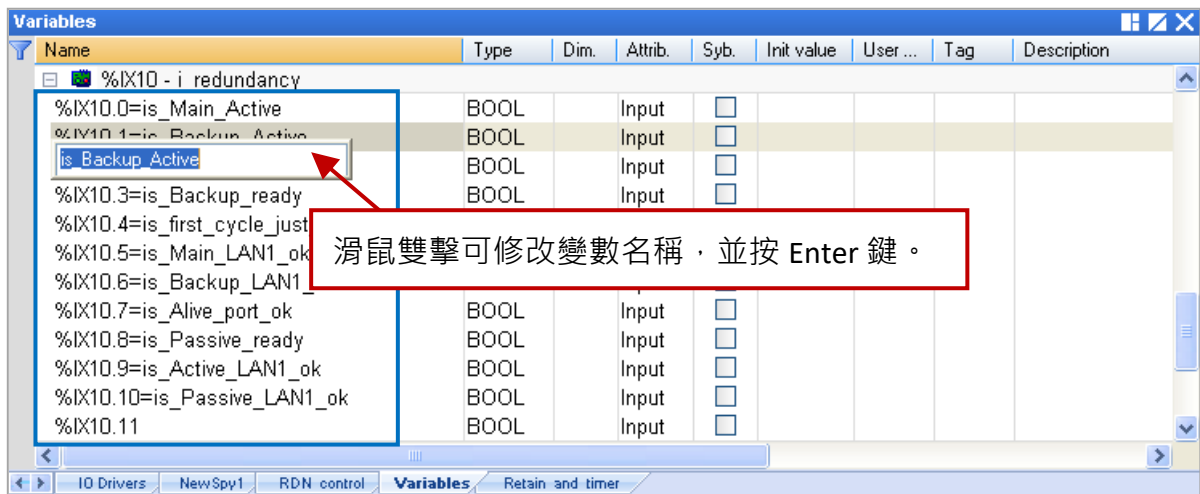
(**注意:** Active IP 位址的最後一個數字不可以是 0, 254, 255, 必須是 1 ~ 253。)

Passive_IP: 固定為 Auto，表示 Passive PAC 的 LAN1 IP 位址，它會自動指定為 **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)。

註: 在 "I/O Boards" 視窗內加入 "i_redundancy" 後，"Variables" 視窗內會自動新增 12 個 "BOOL" 輸入通道用來顯示備援系統的當前狀態，您可滑鼠雙擊來設定名稱。

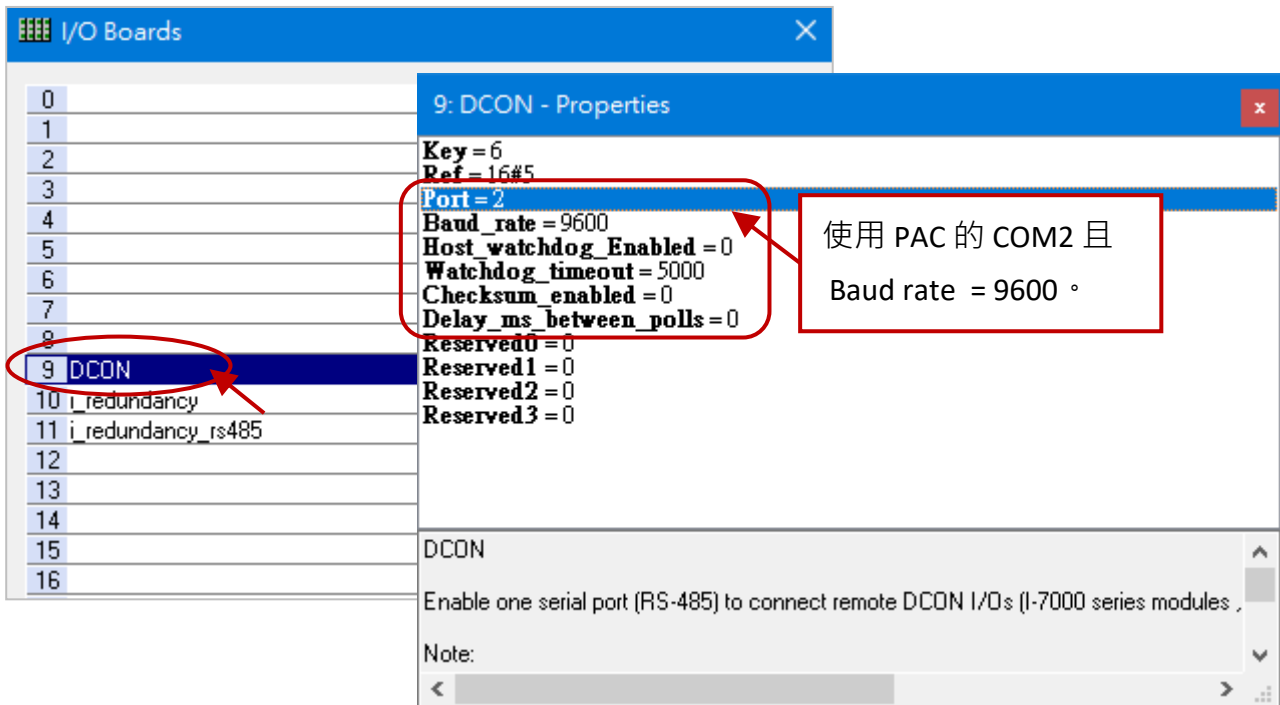


通道	變數名稱	說明
Ch.0	is_Main_Active	Main-PAC 是否為 Active PAC? TRUE: 為 Active PAC (擁有控制權) FALSE: 為 Passive PAC。
Ch.1	is_Backup_Active	Backup-PAC 是否為 Active PAC? TRUE: 為 Active PAC (擁有控制權) FALSE: 為 Passive PAC。
Ch.2	is_Main_ready	Main-PAC 目前是否就緒? 若 Ch.2 回傳 FALSE，可能是以下原因: (1) Main-PAC 與 Backup-PAC 之間的 LAN2 連線損毀 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 (或 6)。

通道	變數名稱	說明
Ch.3	is_Backup_ready	<p>Backup-PAC 目前是否就緒 ?</p> <p>若 Ch.3 回傳 FALSE，可能是以下原因：</p> <p>(1) Main-PAC 與 Backup-PAC 之間的 LAN2 連線損毀或無法通訊。</p> <p>(2) Backup-PAC 當機或損毀。</p> <p>(3) Backup-PAC 的旋轉開關未設定為 9 (或 8)。</p>
Ch.4	is_first_cycle_just_after_switch	<p>僅適用於 Active PAC。</p> <p>True: 目前為切換後的第一個週期。</p> <p>False: 目前不是切換後的第一個週期。</p>
Ch.5	is_Main_LAN1_ok	<p>Main-PAC 的 LAN1 Port 通訊正常嗎?</p> <p>TRUE: 正常。</p> <p>FALSE: 失敗 或 網路線斷訊。</p>
Ch.6	is_Backup_LAN1_ok	<p>Backup-PAC 的 LAN1 Port 通訊正常嗎?</p> <p>TRUE: 正常。</p> <p>FALSE: 失敗 或 網路線斷訊。</p>
Ch.7	is_Alive_port_ok	<p>Alive Port 通訊正常嗎?</p> <p>True : 正常。</p> <p>False : 失敗 或 Passive PAC 當機或損毀。</p>
Ch.8	is_Passive_ready	<p>Passive PAC 目前是否就緒 ?</p> <p>若 Ch.8 回傳 FALSE，可能是以下原因：</p> <p>(1) Main-PAC 與 Backup-PAC 之間的 LAN2 網路連線損毀。</p> <p>(2) Passive PAC 當機或損毀。</p> <p>(3) Passive PAC 的旋轉開關設定錯誤。</p>
Ch.9	is_Active_LAN1_ok	<p>Active PAC 的 LAN1 Port 通訊正常嗎?</p> <p>TRUE: 正常。</p> <p>FALSE: 失敗 或 網路線斷訊。</p>
Ch.10	is_Passive_LAN1_ok	<p>Passive-PAC 的 LAN1 Port 通訊正常嗎?</p> <p>TRUE: 正常。</p> <p>FALSE: 失敗或 網路線斷訊。</p>

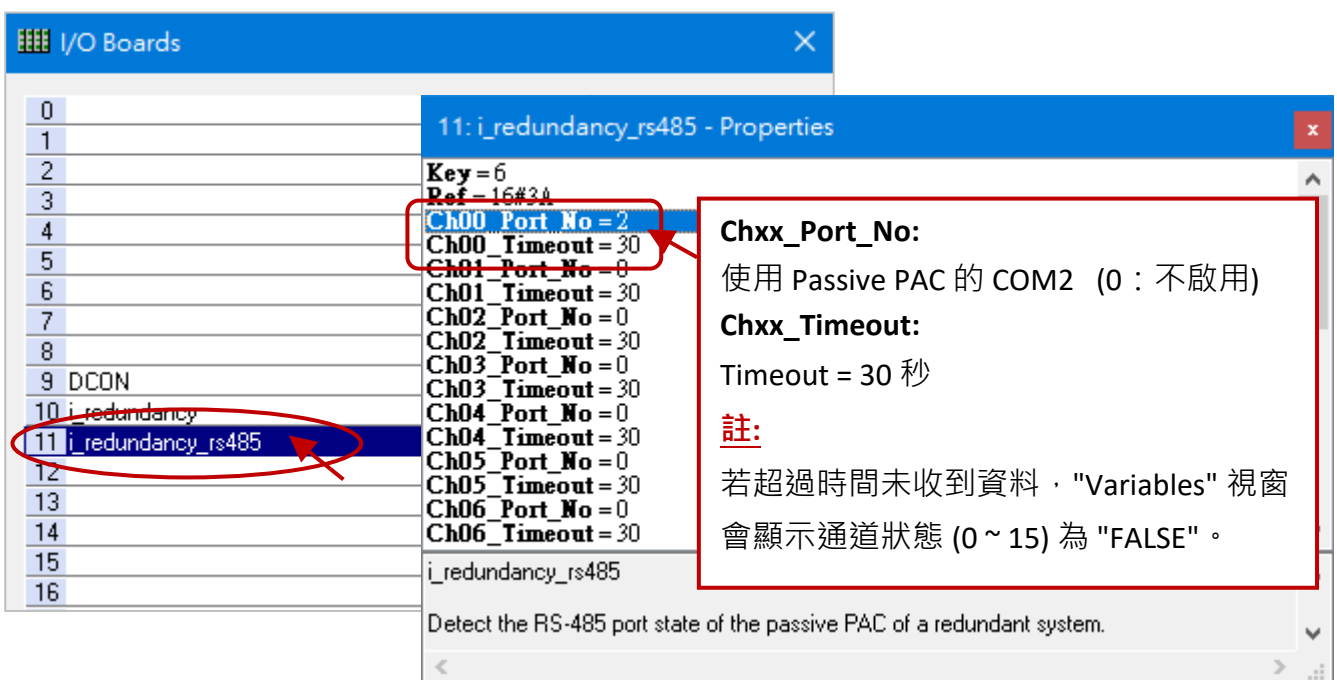
◇ 相關範例: demo_rdn_1

此例，有使用 PAC 的 COM2 (RS-485) 連接 DCON I/O 模組，因此除了在 "I/O Boards" 視窗內加入 "i_redundancy" 功能，也需加入 "DCON" 功能。而 "i_redundancy_rs485" 可用來偵測 Passive PAC 的 RS-485 Port 是否可以正常收到資料？



注意:

1. "i_redundancy_rs485" 必需和 "i_redundancy" 同時使用，否則會無效。
2. "i_redundancy_rs485" 只會開啟 Passive PAC 的 RS-485 Port 來接收資料，並不會傳送資料。



14.3.2 變數宣告 (demo_rdn_2)

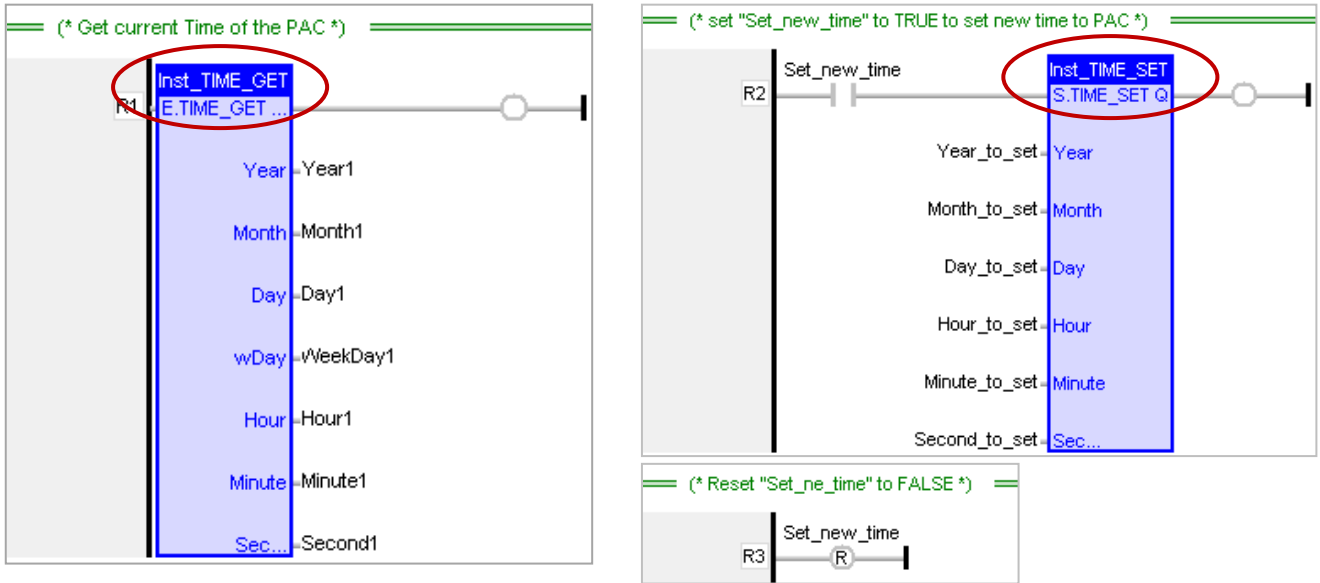
您可在 "Variable" 視窗中查看/建立變數。

變數名稱	資料型態	說明
"PAC_Time" 程式內:		
Year1	DINT	用來取得 PAC 的系統時間。
Month1		
Day1		
WeekDay1		
Hour1		
Minute1		
Second1		
Set_new_time	BOOL	設為 "TRUE"，以設定新的時間。
Year_to_set	DINT	用來設定 PAC 的系統時間。
Month_to_set		
Day_to_set		
Hour_to_set		
Minute_to_set		
Second_to_set		
"Retain_and_timer" 程式內:		
DINT_1	DINT	設定為保存變數。
DINT_2		
REAL_1	REAL	
REAL_2		
TMR_1	TIME	Timer
TMR_2		
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	用來回傳 Retain 狀態。
TMR_1_last_state		TRUE: 計時中；FALSE: 未計時，“Sleep”中。
TMR_2_last_state		TRUE: 計時中；FALSE: 未計時，“Sleep”中。
To_tick_TMR_1		設為 TRUE，以開始計時 TIMER1。
To_tick_TMR_2		設為 TRUE，以開始計時 TIMER2。
To_stop_TMR_1		設為 TRUE，以停止計時 TIMER1。
To_stop_TMR_2		設為 TRUE，以停止計時 TIMER2。

14.3.3 程式介紹 (demo_rdn_2)

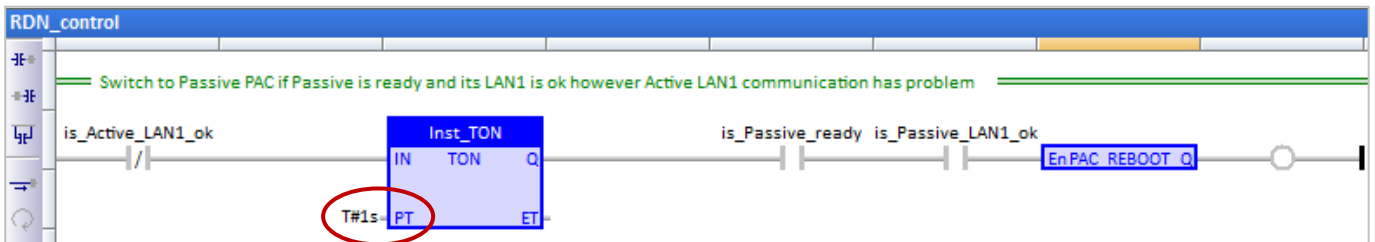
LD 程式 – “PAC_Time”

用來取得/設定 PAC 的系統時間。



LD 程式 – “RDN_control”

當 Active PAC 有問題時，若 Passive PAC 的通訊及健康狀態皆優於 Active PAC，控制權將立即切換到 Passive PAC。此範例程式中，PAC 偵測 LAN1 斷線超過 1 秒後，將自動執行重開機程序”。



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 ;
```

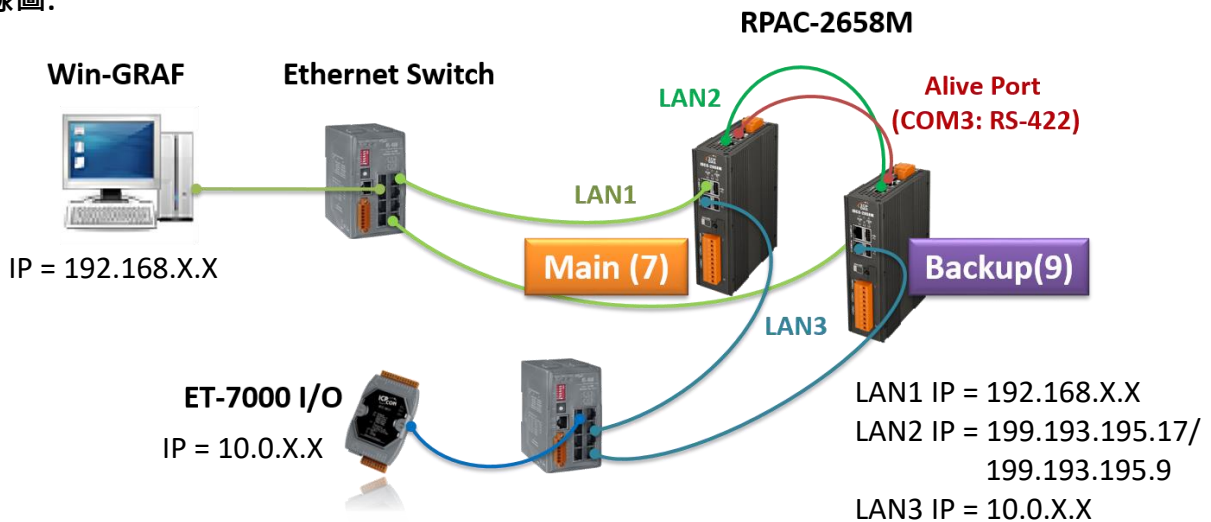
14.4 其他備援程式說明

14.4.1 測試備援程式 (demo_rdn_3、demo_rdn_1)

範例說明 (demo_rdn_3):

2 台 PAC · 使用 LAN3 經過 Ethernet Switch 去連接一個 ET-7050 (Modbus TCP I/O 模組 · 12 DI, 6 DO)。

硬體接線圖:



1. 設定 ET-7000 模組。

請參考 ET-7000 手冊來設定的 IP 位址 與 I/O 設定 (也可參考 5.2.1 節)。

手冊: <https://www.icpdas.com/en/download/show.php?num=2217&model=ET-7050>

2. 查看 Win-GRAF 專案設定。

在 "IO Drivers" 視窗中，啟用了 PAC 為 Modbus TCP Master 來連接 ET-7050 (Modbus TCP Slave · Addr. = 1) 並建立 Data Block 來讀/寫 DI/DO 資料 (設定方式可參考 5.2 節)。

Win-GRAF - demo_rdn_3

File Edit View Insert Project Tools Window Help

Workspace

demo_rdn_3

Exception programs

Programs

PAC_Time

RDN_control

Retain_and_timer

Watch (for debuggin...)

Soft Scope

Initial values

NewSpy1

Binding Configuration

Global defines

Variables

Types

(All Projects)

IO Drivers

MODBUS Master

Open MODBUS: 10.0.79.128:502

<2> Read Input Bits (1) [1..12]

<15> Write Coil Bits (1) [1..6]

<1> Read Coil Bits (1) [1..6]

MODBUS Slave

Server - Slave number = 1

Input Registers [1..2000]

+0: Year1

ET-7050 IP = 10.0.79.128
TCP Port = 502
Data block 用來讀取 12 DI、
寫入 6 DO 與 讀取 6 DO

Request	Slave/Unit	Address	Nb Item	Activation	Period (ms)	Period on error	Timeout (ms)	Number
<2> Read Input Bits	1	1	12	Periodic	50	5000	1000	1
<15> Write Coil Bits	1	1	6	On Change	0	0	1000	1
<1> Read Coil Bits	1	1	6	Periodic	50	5000	1000	1

PAC Time RDN_control Variables Retain and timer NewSpy1 IO Drivers

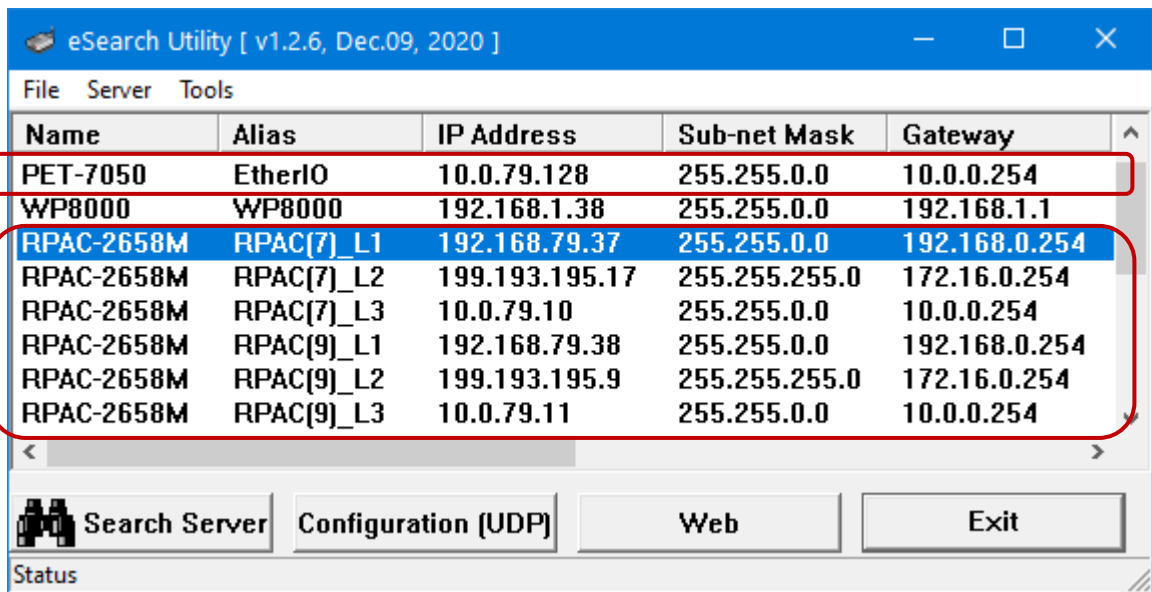
您也可滑鼠雙擊來開啟左方列的任一程式 (Program)，了解更多資訊。

3. 下載 Win-GRAF 專案 ("demo_rdn_3")

請設置適當的 Win-GRAF 的通訊 IP、Active_IP、Mask 與 ET-7000 的 IP 位址，並下載專案到 PAC。

註:

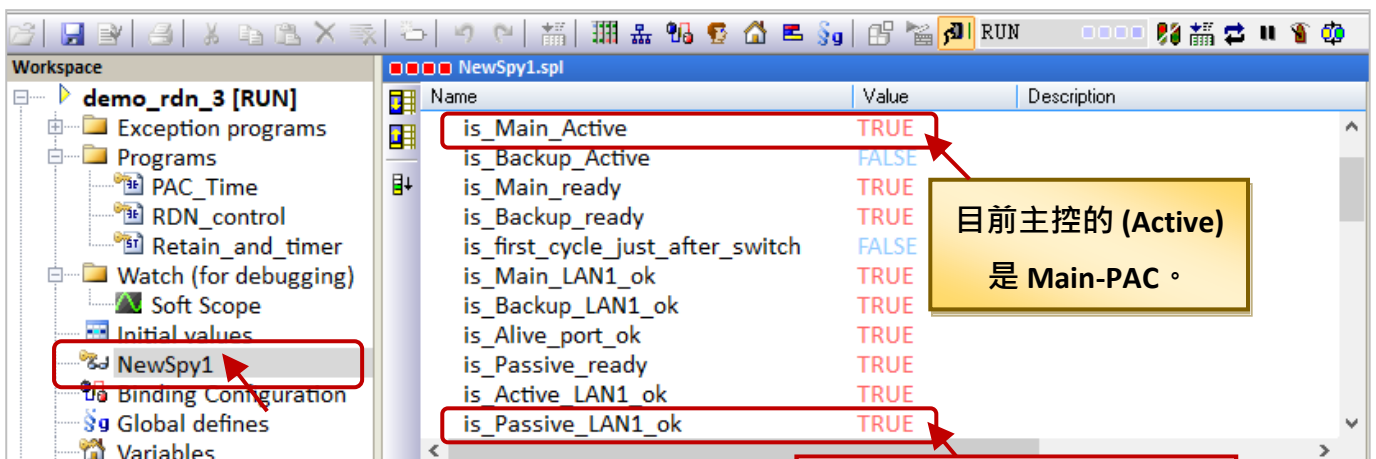
- 您可使用 eSearch Utility 來查看目前設備的 IP 位址。
- 此例，LAN3 與 PET-7050 的 IP 位址必需在相同的網段上。
- 備援模式時，兩台 PAC 的 LAN2 IP 會自動設為 199.193.195.17 與 199.193.195.9。
- 更新專案到 Active-PAC 前，請將 Passive-PAC 斷電，更新完成後，才重新上電 Passive-PAC，以確保備援系統正常運作。



Name	Alias	IP Address	Sub-net Mask	Gateway
PET-7050	EtherIO	10.0.79.128	255.255.0.0	10.0.0.254
WP8000	WP8000	192.168.1.38	255.255.0.0	192.168.1.1
RPAC-2658M	RPAC(7)_L1	192.168.79.37	255.255.0.0	192.168.0.254
RPAC-2658M	RPAC(7)_L2	199.193.195.17	255.255.255.0	172.16.0.254
RPAC-2658M	RPAC(7)_L3	10.0.79.10	255.255.0.0	10.0.0.254
RPAC-2658M	RPAC(9)_L1	192.168.79.38	255.255.0.0	192.168.0.254
RPAC-2658M	RPAC(9)_L2	199.193.195.9	255.255.255.0	172.16.0.254
RPAC-2658M	RPAC(9)_L3	10.0.79.11	255.255.0.0	10.0.0.254

4. 查看 PAC 的備援狀態。

點選 "NewSpy1" 開啟觀測清單，目前主控的 (Active) 是 Main-PAC。



Name	Value	Description
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	

目前主控的 (Active)
是 Main-PAC。

確認 Passive PAC 已就緒後，
才可切換主控權。

5. 設定變數值。

- 1) 修改變數值 (DINT_1, DINT_2, REAL_1, REAL_2)。
 - 2) 將 "To_tick_TMR_1" 設定為 TRUE 來啟動 Timer 計時。
 - 3) 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。

Name	Value	Description
is_Passive_LAN1_ok	TRUE	
DINT_1	1234	Setup as Retain variable in the program "Retain_and_timer"
DINT_2	6006	Setup as Retain variable in the program "Retain_and_timer"
REAL_1	2356.75	Setup as Retain variable in the program "Retain_and_timer"
REAL_2	6592.5	Setup as Retain variable in the program "Retain_and_timer"
TMR_1	t#12s378ms	
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	0	0: No error (comm. ok), Communication error of the ET-7050
ET7050_DO_0_ReadBack	TRUE	If ET-7050_DOx is set to TRUE, ET-7050_DOx_ReadBack will return TRUE.
ET7050_DO_1_ReadBack	TRUE	
ET7050_DO_2_ReadBack	TRUE	
ET7050_DO_3_ReadBack	FALSE	
ET7050_DO_4_ReadBack	FALSE	
ET7050_DO_5_ReadBack	FALSE	
ET7050_DO_0	TRUE	DOx of the ET-7050
ET7050_DO_1	TRUE	
ET7050_DO_2	TRUE	
ET7050_DO_3	FALSE	
ET7050_DO_4	FALSE	
ET7050_DO_5	FALSE	
ET7050_DI_00	FALSE	INx of the ET-7050
ET7050_DI_01	FALSE	
ET7050_DI_02	FALSE	
ET7050_DI_03	FALSE	
ET7050_DI_04	FALSE	
ET7050_DI_05	FALSE	
ET7050_DI_06	FALSE	
ET7050_DI_07	FALSE	
ET7050_DI_08	FALSE	
ET7050_DI_09	FALSE	
ET7050_DI_10	FALSE	
ET7050_DI_11	FALSE	

6. 測試備援功能。

- 1) 確認 Passive PAC 已就緒後 (即，is_Passive_Ready 為 TRUE)。
- 2) 將 Active PAC 的 LAN1 拔除，此時主控權會立即交給另一台 PAC，此範例程式中 (RDN_control)，PAC 會偵測 LAN1 斷線一段時間後，自動執行重開機程序。

Name	Value	Description
is_Main_Active	FALSE	
is_Backup_Active	TRUE	
is_Main_ready	FALSE	
is_Backup_ready	TRUE	
is_first_cycle_just_after_switch	FALSE	

Name	Value	Description
is_Main_LAN1_ok	FALSE	
is_Backup_LAN1_ok	TRUE	
is_Alive_port_ok	FALSE	
is_Passive_ready	FALSE	
is_Active_LAN1_ok	TRUE	
is_Passive_LAN1_ok	FALSE	
DINT_1	1234	
DINT_2	6006	
REAL_1	2356.75	
REAL_2	6592.5	
TMR_1	t#18m32s459ms	
To_tick_TMR_1	FALSE	Set TRUE to start ticking timer
To_stop_TMR_1	FALSE	Set TRUE to stop the ticking
ET7050_COM_err	0	0: No error (comm. ok), Cor
ET7050_DO_0_ReadBack	TRUE	If ET-7050_DOx is set to TR
ET7050_DO_1_ReadBack	TRUE	
ET7050_DO_2_ReadBack	TRUE	
ET7050_DO_3_ReadBack	FALSE	
ET7050_DO_4_ReadBack	FALSE	
ET7050_DO_5_ReadBack	FALSE	
ET7050_DO_0	TRUE	
ET7050_DO_1	TRUE	
ET7050_DO_2	TRUE	
ET7050_DO_3	FALSE	
ET7050_DO_4	FALSE	
ET7050_DO_5	FALSE	
ET7050_DI_00	FALSE	
ET7050_DI_01	FALSE	
ET7050_DI_02	FALSE	
ET7050_DI_03	FALSE	
ET7050_DI_04	FALSE	
ET7050_DI_05	FALSE	
ET7050_DI_06	FALSE	
ET7050_DI_07	FALSE	
ET7050_DI_08	FALSE	
ET7050_DI_09	FALSE	
ET7050_DI_10	FALSE	
ET7050_DI_11	FALSE	

將 Main-PAC (Passive) 的 LAN1 接上，來回復為 TRUE。

切換主控後，各變數仍為原先的值，且 Timer 仍維持計時。

ET-7050_COM_err 顯示非 0，表示有錯誤。設定 ET-7050_Dox = TRUE，若通訊正常，會回傳 ET-7050_DOx_ReadBack = TRUE。

INx of the ET-7050

測試備援程式 (demo_rdn_1)

範例說明:

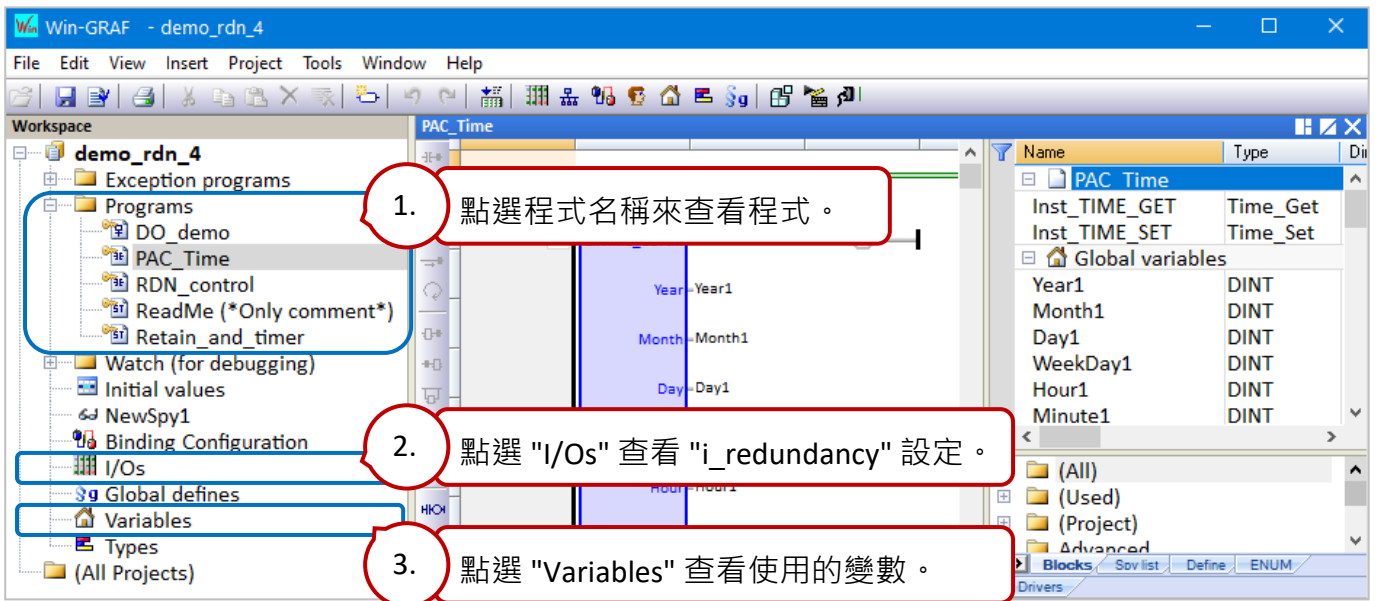
2 台 PAC，使用 **COM2 (RS-485)** 去連接 3 個 DCON I/O 模組 (I-87064, I-87018ZW, I-7065)。

1. 兩台 PAC 與遠端 DCON I/O 模組連線之前，請參考 [第 8 章](#) 使用 "DCON Utility Pro" 設定模組與 "I/O Boards" 設定 - DCON。
2. 在 Win-GRAF 查看程式與 "I/O Boards" 設定 - i_redundancy、i_redundancy_rs485 (見 14.3.1 節)。
3. 下載 Win-GRAF 專案 ("demo_rdn_1") 後，可參考 "demo_rdn_3" 步驟 5、6 進行測試。

14.4.2 程式介紹 (demo_rdn_4)

範例說明:

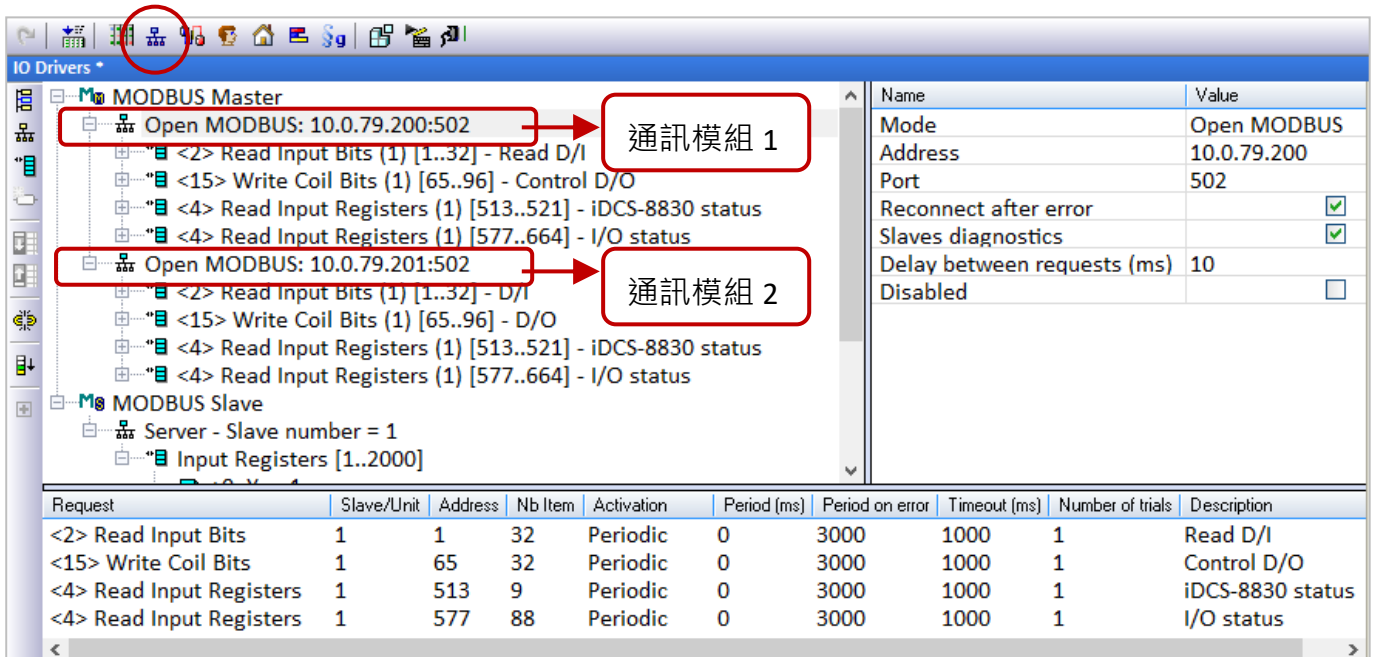
使用 2 台 PAC 的 LAN3，經由 Ethernet Switch 去連接一個 iDCS-8830 (可支援備援 I/O 模組)。



iDCS-8830 配備有備援電源輸入、備援 Ethernet 通訊模組與 8 個 I/O 插槽，此範例使用一組備援 DI 模組 (F-8040，Slot 0、1) 與一組備援 DO 模組 (F-8041，Slot 2、3)。

"Modbus Master" 設定

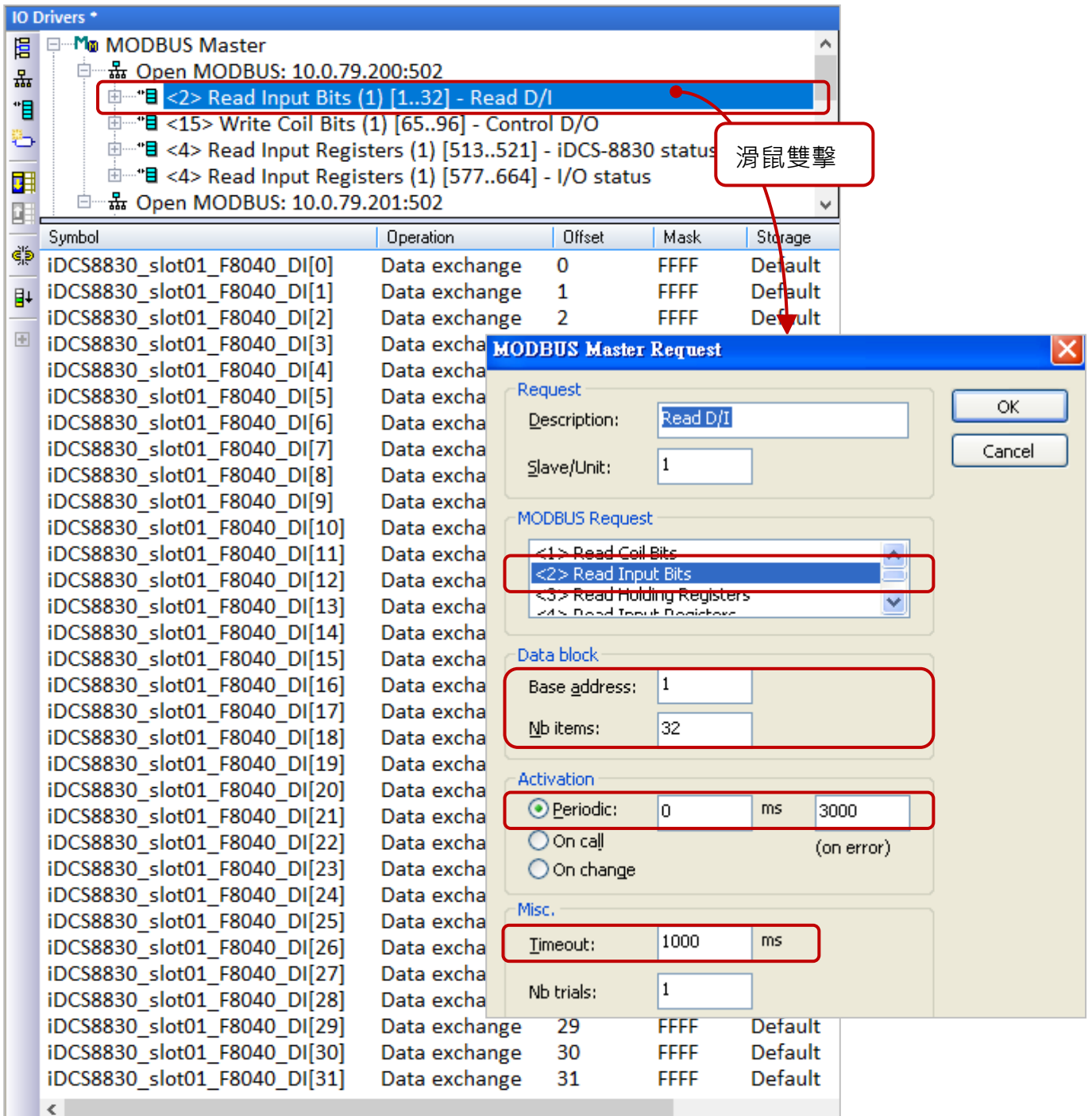
點選 "Open Fieldbus Configuration" 按鈕來開啟 "IO Drivers" 視窗。此例，啟用 Modbus Master 來連接 2 個 Modbus TCP Slave (即 iDCS-8830 的通訊模組) 來讀/寫資料或通訊狀態。



註: LAN3 與通訊模組的 IP 位址必需在相同的網段上。

讀取 DI 說明 (F-8040 · Slot 0 · 1):

由位址 1 開始讀取 32 個 DI 值，一直發送命令，若出現異常，於 3 秒後再發送。1 秒未回應會視為異常。



注意: 本範例使用 iDCS-8830 Ethernet 備援 I/O 單元，請參考 [FCM-MTCP 軟體手冊](#) (4.1.1 節) 來填寫 "Base address" 位址。

讀取 DO 說明 (F-8041 · Slot 2、3):

由位址 65 開始寫入 32 個 DO 值，一直發送命令，若出現異常，於 3 秒後再發送。1 秒未回應會視為異常。

滑鼠雙擊

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[6]	Data exchange	6	FFFF	Default
iDCS8830_slot23_F8041_DO[7]	Data exchange	7	FFFF	Default
iDCS8830_slot23_F8041_DO[8]	Data exchange	8	FFFF	Default
iDCS8830_slot23_F8041_DO[9]	Data exchange	9	FFFF	Default
iDCS8830_slot23_F8041_DO[10]	Data exchange	10	FFFF	Default
iDCS8830_slot23_F8041_DO[11]	Data exchange	11	FFFF	Default
iDCS8830_slot23_F8041_DO[12]	Data exchange	12	FFFF	Default
iDCS8830_slot23_F8041_DO[13]	Data exchange	13	FFFF	Default
iDCS8830_slot23_F8041_DO[14]	Data exchange	14	FFFF	Default
iDCS8830_slot23_F8041_DO[15]	Data exchange	15	FFFF	Default
iDCS8830_slot23_F8041_DO[16]	Data exchange	16	FFFF	Default
iDCS8830_slot23_F8041_DO[17]	Data exchange	17	FFFF	Default
iDCS8830_slot23_F8041_DO[18]	Data exchange	18	FFFF	Default
iDCS8830_slot23_F8041_DO[19]	Data exchange	19	FFFF	Default
iDCS8830_slot23_F8041_DO[20]	Data exchange	20	FFFF	Default
iDCS8830_slot23_F8041_DO[21]	Data exchange	21	FFFF	Default
iDCS8830_slot23_F8041_DO[22]	Data exchange	22	FFFF	Default
iDCS8830_slot23_F8041_DO[23]	Data exchange	23	FFFF	Default
iDCS8830_slot23_F8041_DO[24]	Data exchange	24	FFFF	Default
iDCS8830_slot23_F8041_DO[25]	Data exchange	25	FFFF	Default
iDCS8830_slot23_F8041_DO[26]	Data exchange	26	FFFF	Default
iDCS8830_slot23_F8041_DO[27]	Data exchange	27	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

Data block

Base address: 65

Nb items: 32

Activation

Periodic: 0 ms 3000
 On call (on error)
 On change

Misc.

Timeout: 1000 ms

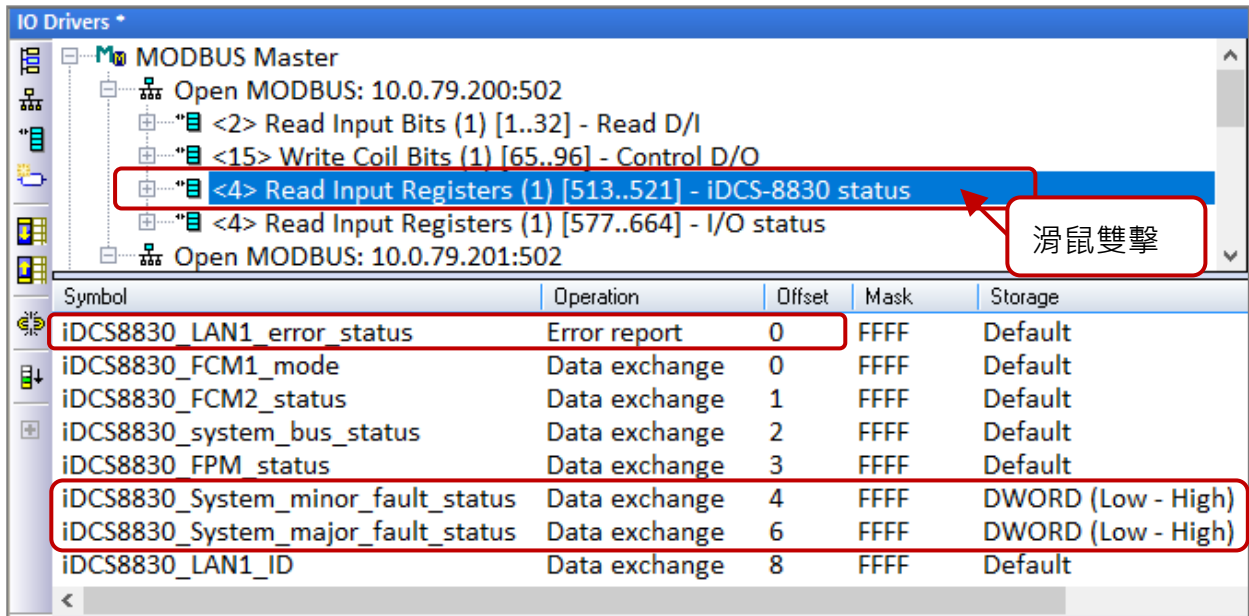
Nb trials: 1

注意:

請參考 [FCM-MTCP 軟體手冊](#) (4.1.2 節) 來填寫 "Base address" 位址。

讀取 iDCS-8830 狀態:

- "iDCS8830_LAN1_error_status" 的 "Operation" 設定為 "Error report" 表示讀取失敗時，會回傳 "Error Code"；讀取成功時，會重置為 "0"，且 "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 [FCM-MTCP 軟體手冊](#) (4.2.1 節) 來填寫 "Base address" 位址。



讀取 iDCS-8830 的 I/O 狀態

由位址 577 開始讀取系統資訊，一直發送命令，若出現異常，於 3 秒後再發送。1 秒未回應會視為異常。請參考 [FCM-MTCP 軟體手冊](#) (4.2.2 節) 來填寫 "Base address" 與 "Offset" 位址。

起始位址:

Module ID: 00577

I/O slot status: 00593

Emergency: 00609

Channel break status: 00649

Symbol	Operation	Offset	...
iDCS8830_io_slot_status[0]	Data exchange	16	F
iDCS8830_io_slot_status[1]	Data exchange	17	F
iDCS8830_io_slot_status[2]	Data exchange	18	F
iDCS8830_io_slot_status[3]	Data exchange	19	F
iDCS8830_io_slot_status[4]	Data exchange	20	F
iDCS8830_io_slot_status[5]	Data exchange	21	F
iDCS8830_io_slot_status[6]	Data exchange	22	F
iDCS8830_io_slot_status[7]	Data exchange	23	F
iDCS8830_io_emergency_status[0]	Data exchange	32	F
iDCS8830_io_emergency_status[1]	Data exchange	33	F
iDCS8830_io_emergency_status[2]	Data exchange	34	F
iDCS8830_io_emergency_status[3]	Data exchange	35	F
iDCS8830_io_emergency_status[4]	Data exchange	36	F
iDCS8830_io_emergency_status[5]	Data exchange	37	F
iDCS8830_io_emergency_status[6]	Data exchange	38	F
iDCS8830_io_emergency_status[7]	Data exchange	39	F
iDCS8830_io_channel_break_status[0]	Data exchange	72	FFFF
iDCS8830_io_channel_break_status[1]	Data exchange	74	FFFF
iDCS8830_io_channel_break_status[2]	Data exchange	76	FFFF
iDCS8830_io_channel_break_status[3]	Data exchange	78	FFFF
iDCS8830_io_channel_break_status[4]	Data exchange	80	FFFF
iDCS8830_io_channel_break_status[5]	Data exchange	82	FFFF
iDCS8830_io_channel_break_status[6]	Data exchange	84	FFFF
iDCS8830_io_channel_break_status[7]	Data exchange	86	FFFF

MODBUS Master Request

Request Description: I/O status

Slave/Unit: 1

MODBUS Request: <4> Read Input Registers

Data block: Base address: 577, Nb items: 88

Activation: Periodic: 0 ms, 3000

Misc: Timeout: 1000 ms, Nb trials: 1

Symbol: iDCS8830_io_channel_break_status[0] - Offset: 72 - Data type: DWORD (Low - High)

資料型態為 "DWORD" (32 bit) · 需要占用 2 個 Modbus 位址 · Storage 需設為 "DWORD (Low - High)" 。

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](#)。

14.4.3 測試備援程式 (demo_rdn_4)

範例說明:

2 台 PAC · 使用 LAN3 經過 Ethernet Switch 去連接一台 iDCS-8830 備援 I/O 單元。

下表中列出了此範例所需使用的設備:

產品型號	數量	產品型號	數量
RPAC-2658M	2	DN-DI-32DW	1
NS-208	1	DN-DO-16DR-A	1
iDCS-8830	1	DN-DO-16DR-B	1
F-8040 (DI)	2	CA-3710AM (1M Cable) 或	4
F-8041 (DO)	2	CA-3720AM/30AM/50AM/100AM	

以下將說明如何設定 iDCS-8830 · 請參訪網頁來下載軟體與手冊。

◇ iDCS-8000 網頁: <https://www.icpdas.com/en/product/iDCS-8830>

◇ 下載區: <https://www.icpdas.com/en/download/index.php?model=iDCS-8830>

◇ 軟體手冊: <https://www.icpdas.com/en/download/show.php?num=1677&model=iDCS-8830>

FPM-D2440 * 2 : 電源模組 1、電源模組 2 (要接上 24V 電源輸入)

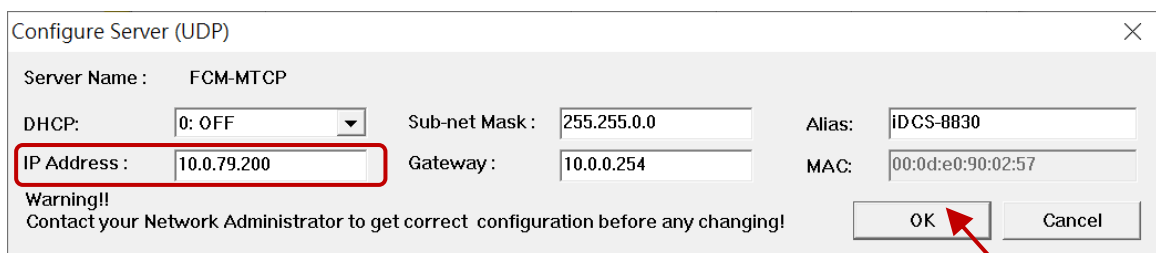
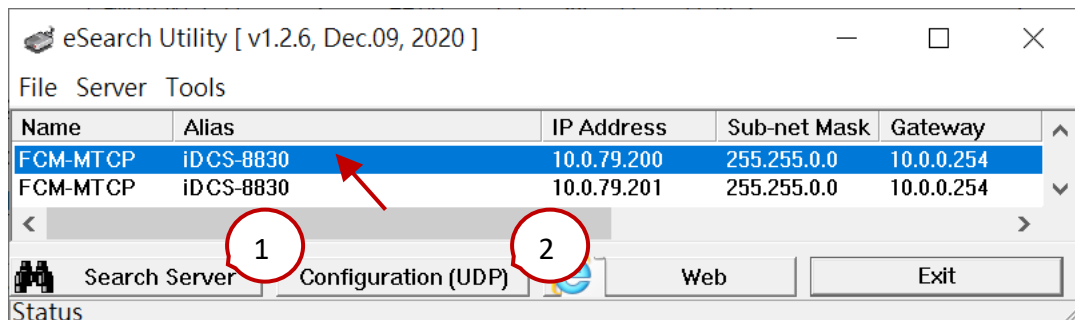
FCM-MTCP * 2 : 通訊模組 1 (先將 SW2 轉到 C, SW1 轉到 8, 再設定 IP 位址為 10.0.79.200)。

通訊模組 2 (先將 SW2 轉到 C, SW1 轉到 9, 再設定 IP 位址為 10.0.79.201)。

通訊模組上的 SW2/SW1 開關, 可表示第 4 位 IP 位址 (C8₁₆ = 200 ; C9₁₆ = 201) ,

使用 eSearch Utility 設定 IP 位址:

開啟 eSearch Utility, 並點選 “Search Server” 搜尋設備, 然後點選 “Configuration” 按鈕來設定 IP 位址。此例, 設定 2 個 IP 位址為 10.0.79.200 與 10.0.79.201, 並設定 Mask 為 255.255.0.0。

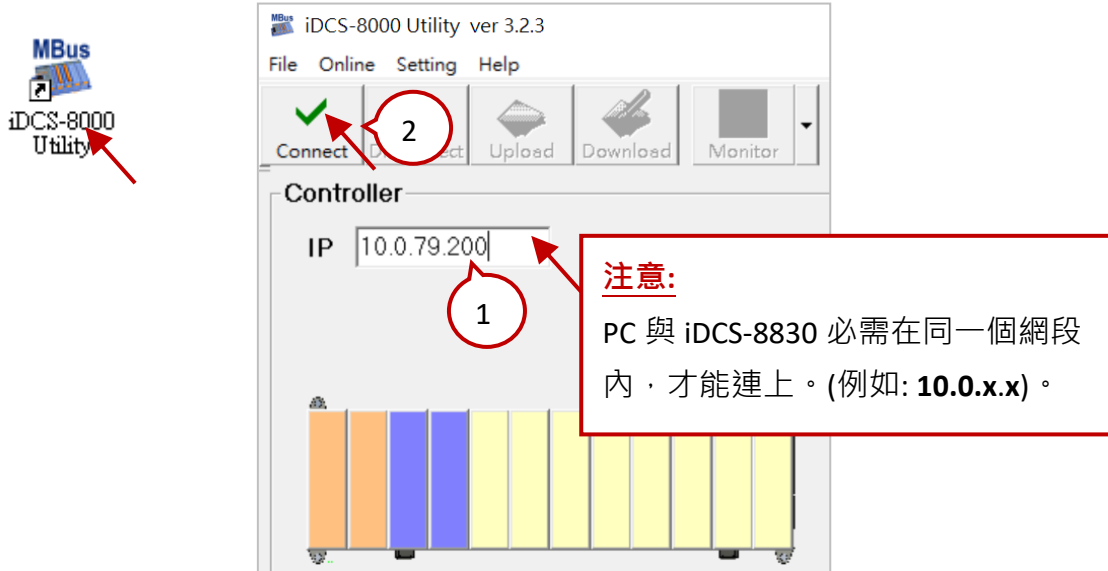


F-8040 * 2: 32 通道 DI 備援模組，請分別插在 I/O 插槽 0 與 1。

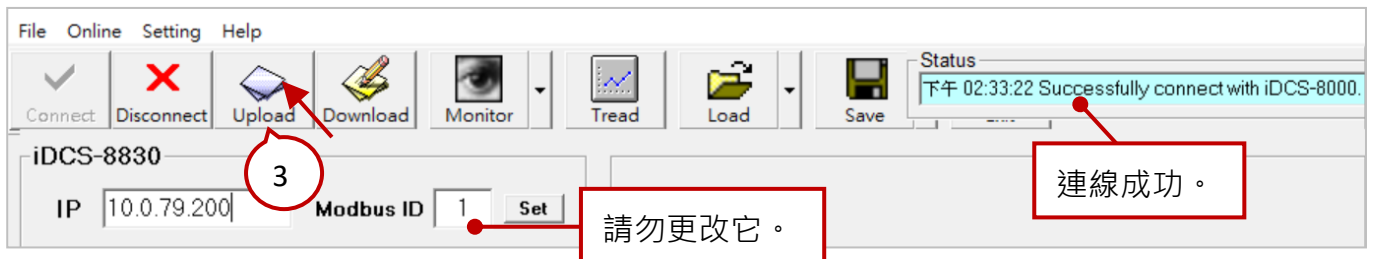
F-8041 * 2: 32 通道 DO 備援模組，請分別插在 I/O 插槽 2 與 3。

使用 iDCS-8000 Utility 設定 I/O 模組:

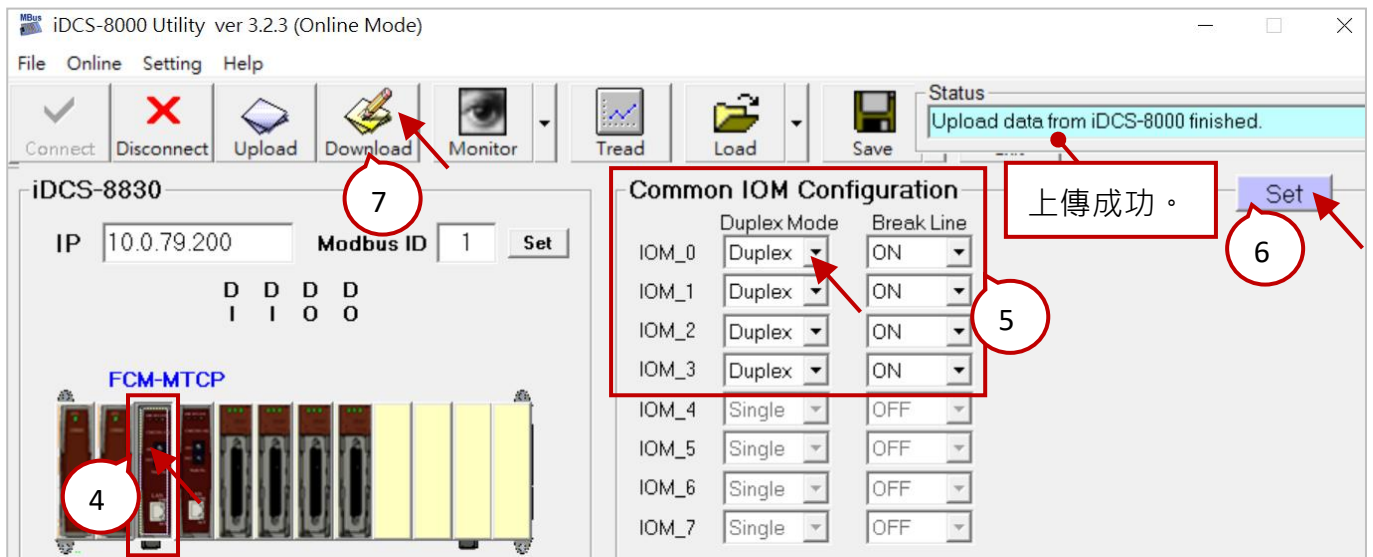
1) 開啟 iDCS-8000 Utility，並輸入 iDCS-8830 的 IP 位址進行連線。



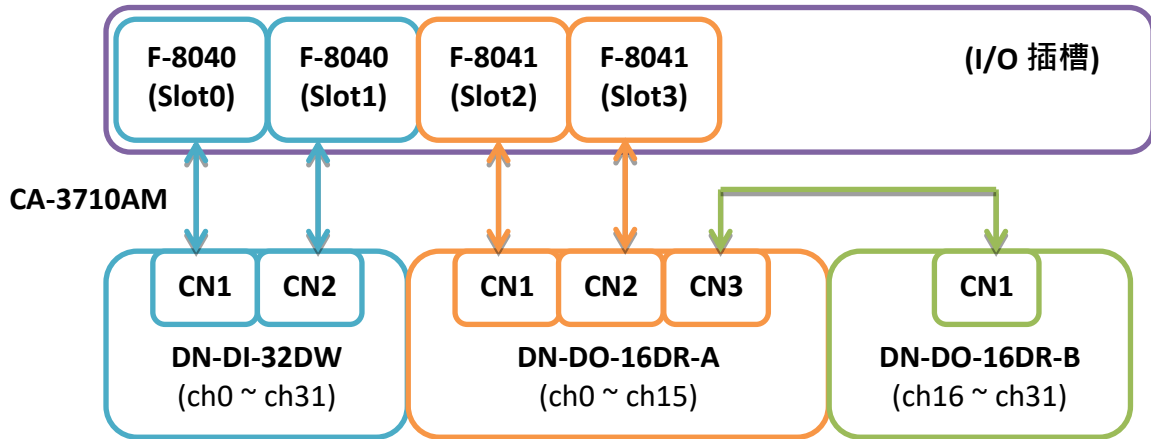
2) 點選 "Upload" 按鈕，上傳目前 iDCS-8830 上的 I/O 模組設定。



3) 點選第 1 個 FCM-MTCP (MCU1)，來設定 F-8040 與 F-8041 (IOM_0 ~ IOM_3) 為 "Duplex" Mode。



4) 確認 F-8040、F-8041 已連接好下列端子板，並將 "Break Line" (斷線偵測) 設定為 "ON"。



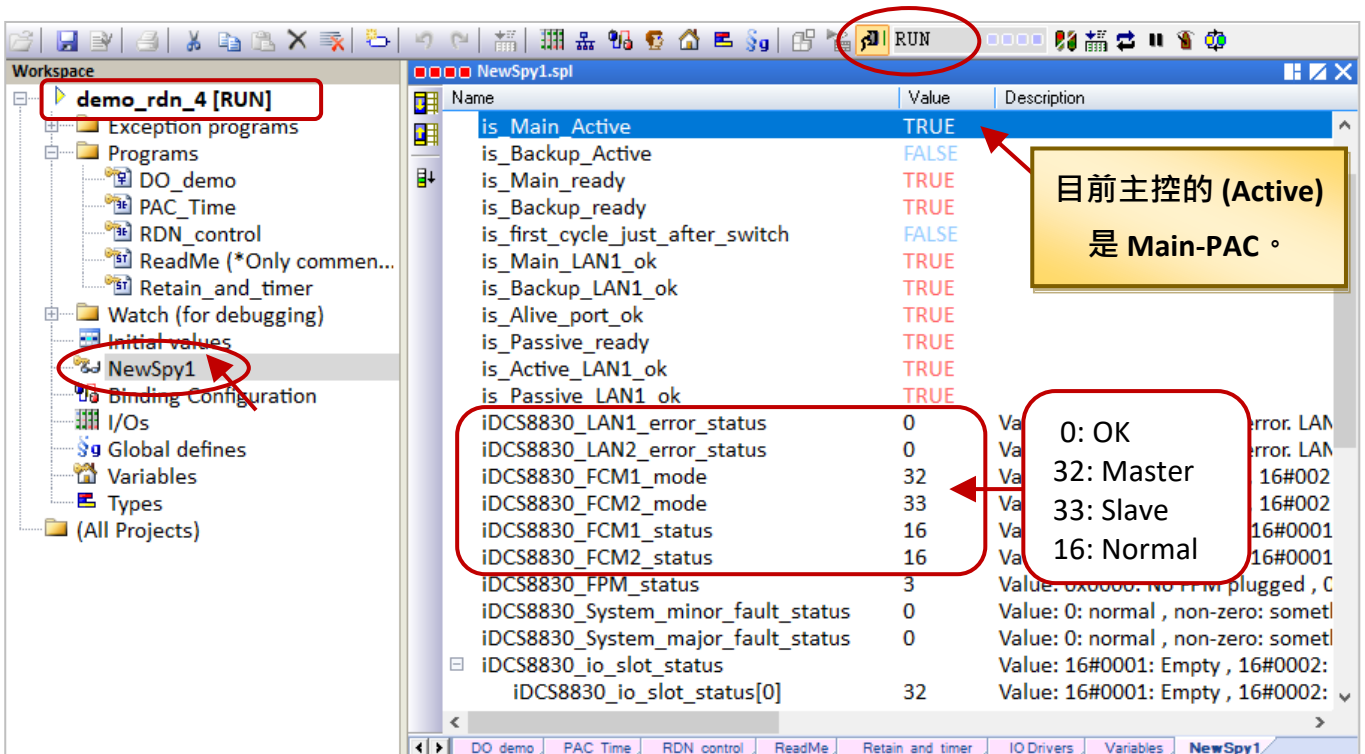
5) 點選 "Set" 再點選 "Download"，將設定值下載到 iDCS-8830 內，即可關閉 iDCS-8000 Utility。

開始測試：

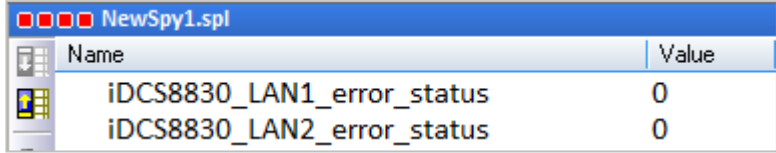
請確認所有設備都已經連接好，並修改適當的 Win-GRAF 的通訊 IP、Active_IP、Mask 位址，再下載專案 ("demo_rdn_4") 到 PAC。請確認 PC 與 PAC 的 IP 位址是在相同的網段。

註：更新專案到 Active-PAC 前，請將 Passive-PAC 斷電，更新完成後，才重新上電 Passive-PAC，以確保備援系統正常運作。

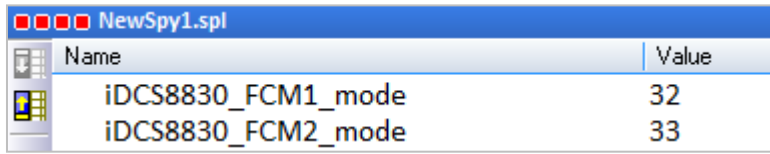
請點選 "New Spy1" 開啟變數清單。

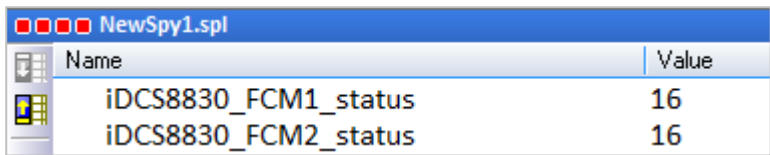


iDCS-8830 的網路通訊狀態:

變數名稱	iDCS8830_LAN1_error_status	iDCS8830_LAN2_error_status
說明	LAN1 的通訊狀態	LAN2 的通訊狀態
狀態值	0: 通訊正常; 非 0: 通訊異常	
		

FCM-MTCP 的備援模式與狀態:

變數名稱	iDCS8830_FCM1_mode	iDCS8830_FCM2_mode
說明	FCM1 的備援模式	FCM2 的備援模式
	Modbus 起始位址: 513	
狀態值	32 (16#0020): Master 33 (16#0021): Slave	
		

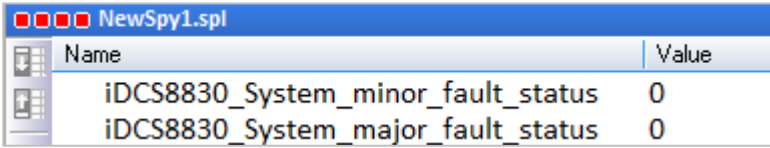
變數名稱	iDCS8830_FCM1_status	iDCS8830_FCM2_status
說明	FCM1 的狀態	FCM2 的狀態
	Modbus 起始位址: 514	
狀態值	0 (16#0000): 空的 1 (16#0001): 逾時 2 (16#0002): 未知 16 (16#0010): 正常	
		

測試方式	1. 拔除 LAN1，此時 FCM2 接手為 (32: Master)。	
	iDCS8830_LAN1_error_status=130	iDCS8830_LAN2_error_status = 0
	iDCS8830_FCM1_mode = 33	iDCS8830_FCM2_mode = 32
	iDCS8830_FCM1_status = 1	iDCS8830_FCM2_status = 16
	2. 再將 LAN1 插上。	
	iDCS8830_LAN1_error_status=0	iDCS8830_LAN2_error_status = 0
iDCS8830_FCM1_mode = 33	iDCS8830_FCM2_mode = 32	
iDCS8830_FCM1_status = 16	iDCS8830_FCM2_status = 16	

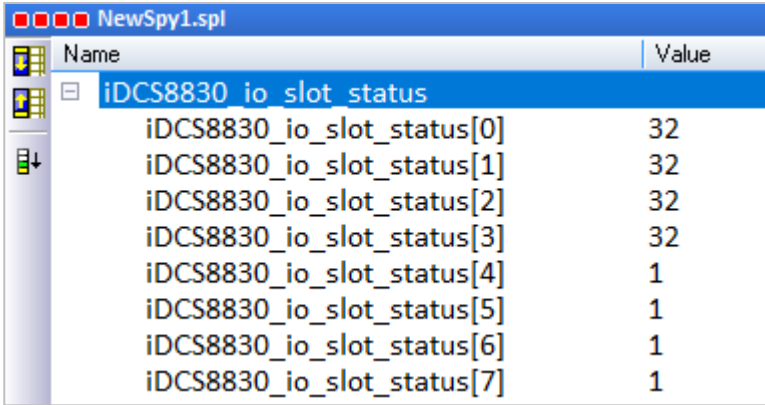
iDCS-8830 的電源狀態:

變數名稱	iDCS8830_FPM_status	
說明	兩個電源模組的狀態	
	Modbus 起始位址: 516	
狀態值	0: 未安裝 FPM 1: FPM1 正常 / FPM2 關閉 2: FPM1 關閉 / FPM2 正常 3: 2 個 FPM 電源正常	
		

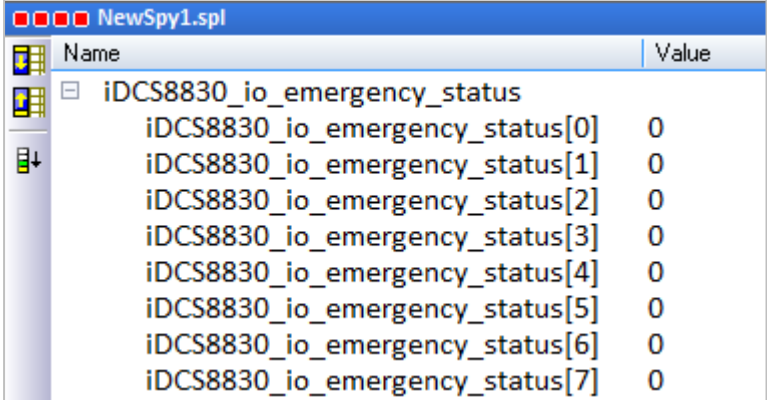
iDCS-8830 的故障狀態:

變數名稱	iDCS8830_System_minor_fault_status	iDCS8830_System_major_fault_status
說明	系統的小故障 或 重大故障狀態	
	Modbus 起始位址: 517 / 518	Modbus 起始位址: 519 / 520
狀態值	0: 正常 非 0: 有異常，請參考 FCM-MTCP 軟體手冊 (4.2.1 節)	
		

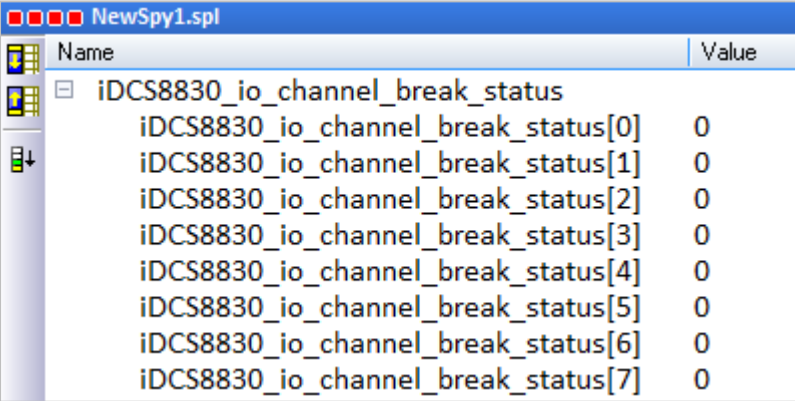
I/O 插槽狀態:

變數名稱	iDCS8830_io_slot_status	
說明	I/O 插槽 0~7 的狀態	
	Modbus 起始位址: 593	
狀態值	1 (16#0001): 空的 2 (16#0002): 中斷 4 (16#0004): 啟動 8 (16#0008): 啟動載入 16 (16#0010): 前置處理 32 (16#0020): 運行 64 (16#0040): 停止	
		

I/O 緊急狀態:

變數名稱	iDCS8830_io_emergency_status																				
說明	I/O 插槽 0~7 的緊急狀態 註: 需先在 iDCS-8000 Utility (參考 14..3 節) 將 IOM_x 設定為 "Duplex" Mode · iDCS8830_io_emergency_status[x] 才有作用。																				
	Modbus 起始位址: 609																				
狀態值	<p>0: OK, 32 (16#0020): CJC 錯誤, 256 (16#0100): 線材斷線</p>  <table border="1" data-bbox="497 568 1267 967"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>iDCS8830_io_emergency_status</td> <td></td> </tr> <tr> <td>iDCS8830_io_emergency_status[0]</td> <td>0</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>iDCS8830_io_emergency_status[4]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_emergency_status[5]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_emergency_status[6]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_emergency_status[7]</td> <td>0</td> </tr> </tbody> </table>	Name	Value	iDCS8830_io_emergency_status		iDCS8830_io_emergency_status[0]	0	iDCS8830_io_emergency_status[1]	0	iDCS8830_io_emergency_status[2]	0	iDCS8830_io_emergency_status[3]	0	iDCS8830_io_emergency_status[4]	0	iDCS8830_io_emergency_status[5]	0	iDCS8830_io_emergency_status[6]	0	iDCS8830_io_emergency_status[7]	0
Name	Value																				
iDCS8830_io_emergency_status																					
iDCS8830_io_emergency_status[0]	0																				
iDCS8830_io_emergency_status[1]	0																				
iDCS8830_io_emergency_status[2]	0																				
iDCS8830_io_emergency_status[3]	0																				
iDCS8830_io_emergency_status[4]	0																				
iDCS8830_io_emergency_status[5]	0																				
iDCS8830_io_emergency_status[6]	0																				
iDCS8830_io_emergency_status[7]	0																				

I/O 通道狀態

變數名稱	iDCS8830_io_channel_break_status																				
說明	I/O 通道 0~7 的斷線偵測																				
	Modbus 起始位址: 649																				
狀態值	<p>0: 正常, 非 0: 對應 bit = 1 · 表示該通道斷線</p>  <table border="1" data-bbox="497 1388 1295 1787"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>iDCS8830_io_channel_break_status</td> <td></td> </tr> <tr> <td>iDCS8830_io_channel_break_status[0]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[1]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[2]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[3]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[4]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[5]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[6]</td> <td>0</td> </tr> <tr> <td>iDCS8830_io_channel_break_status[7]</td> <td>0</td> </tr> </tbody> </table>	Name	Value	iDCS8830_io_channel_break_status		iDCS8830_io_channel_break_status[0]	0	iDCS8830_io_channel_break_status[1]	0	iDCS8830_io_channel_break_status[2]	0	iDCS8830_io_channel_break_status[3]	0	iDCS8830_io_channel_break_status[4]	0	iDCS8830_io_channel_break_status[5]	0	iDCS8830_io_channel_break_status[6]	0	iDCS8830_io_channel_break_status[7]	0
Name	Value																				
iDCS8830_io_channel_break_status																					
iDCS8830_io_channel_break_status[0]	0																				
iDCS8830_io_channel_break_status[1]	0																				
iDCS8830_io_channel_break_status[2]	0																				
iDCS8830_io_channel_break_status[3]	0																				
iDCS8830_io_channel_break_status[4]	0																				
iDCS8830_io_channel_break_status[5]	0																				
iDCS8830_io_channel_break_status[6]	0																				
iDCS8830_io_channel_break_status[7]	0																				

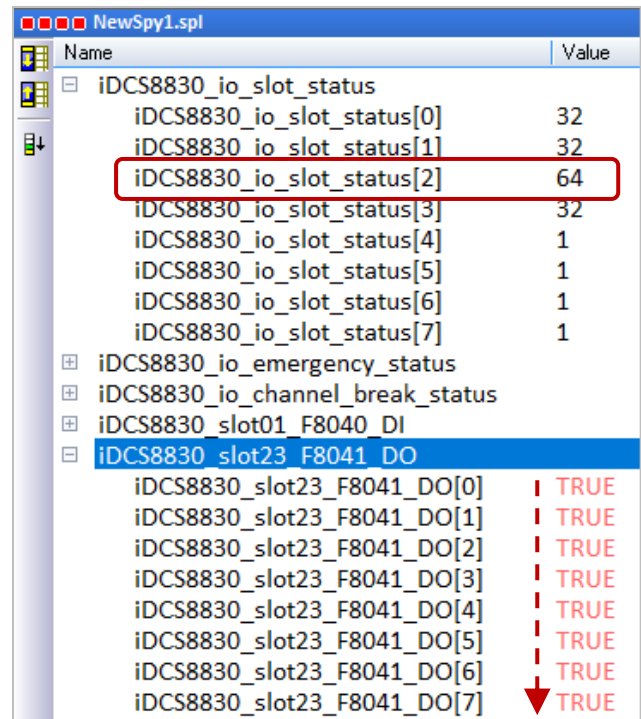
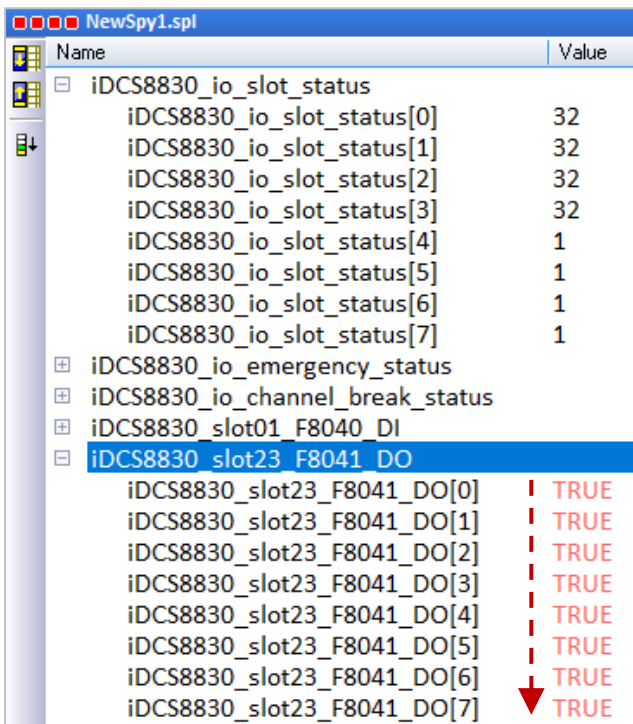
F8040 (DI) 、 F8041 (DO) 模組狀態

變數名稱	iDCS8830_slot01_F8040_DI	iDCS8830_slot23_F8041_DO
說明	I/O 插槽 0、1 (Duplex Mode) 上的 F8040 (32 DI) 模組狀態	I/O 插槽 2、3 (Duplex Mode) 上的 F8041 (32 DO) 模組狀態
	Modbus 起始位址: 1	Modbus 起始位址: 65

測試方式

一開始 F-8041 (Slot2) 的 LED (DO0 ~ 7) 會依序亮燈。

1. 拔除 F-8041 (Slot2) 模組後，會立即切換到第 2 個 F-8041 (Slot3) 依序亮燈。
2. 此時，可見到 iDCS8830_io_slot_status[2] = 64 表示模組 (Slot2) 中止工作，請再插上該模組。



以上是針對 iDCS-8830 備援 I/O 的測試說明，您也可參考 [14.4.1 節](#) 測試 RPAC 備援系統。

第 15 章 排程控制 (Schedule Control)



簡介:

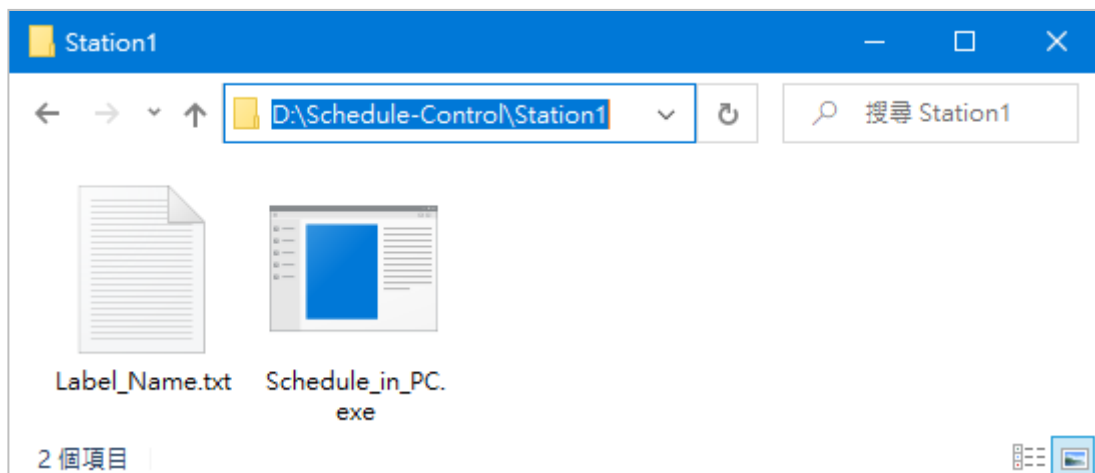
Win-GRAF PAC 皆支援排程控制功能 (Schedule Control) ，一台 Win-GRAF PAC 最多可控制 10 個 Target 的排程。每個 Target 可控制 1 個布林 (BOOL) 、1 個整數 (DINT) 與 1 個實數 (REAL) 變數，且可依據全年、四季來設定各 5 個一周排程，而一個排程可設定 15 個時段。ICP DAS 提供免費的排程工具 - “Schedule-Control Utility”，讓 User 可簡單快速地完成排程設定。

15.1 安裝 Schedule-Control Utility 與回存 Win-GRAF 範例程式

您可在 [網站](#) 下載 Schedule-Control 工具程式 - “Schedule_in_PC.zip”。

安裝 Schedule-Control Utility:

請解壓縮 zip 檔並將 Schedule_in_PC.exe 與 label_name.txt 複製到您的 PC 上，建議存放在專案目錄內 (例如，D:\Schedule-Control\Station1\)

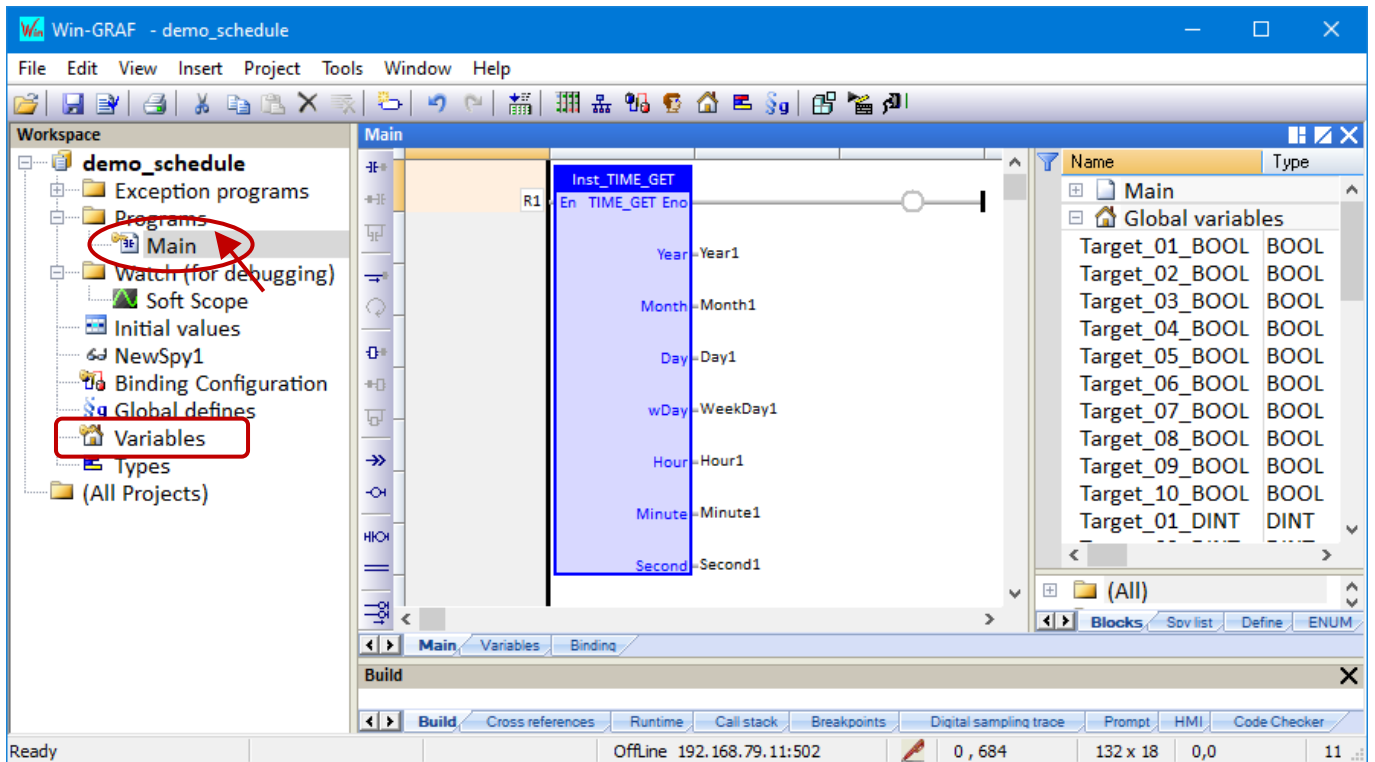


回存 Win-GRAF 範例程式:

請在網站下載 [範例程式 \(demo_schedule.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。(可參考 11.4 節)

15.2 Win-GRAF 範例程式說明

此範例包含一個 LD 程式 (Main)，用來取得系統時間。

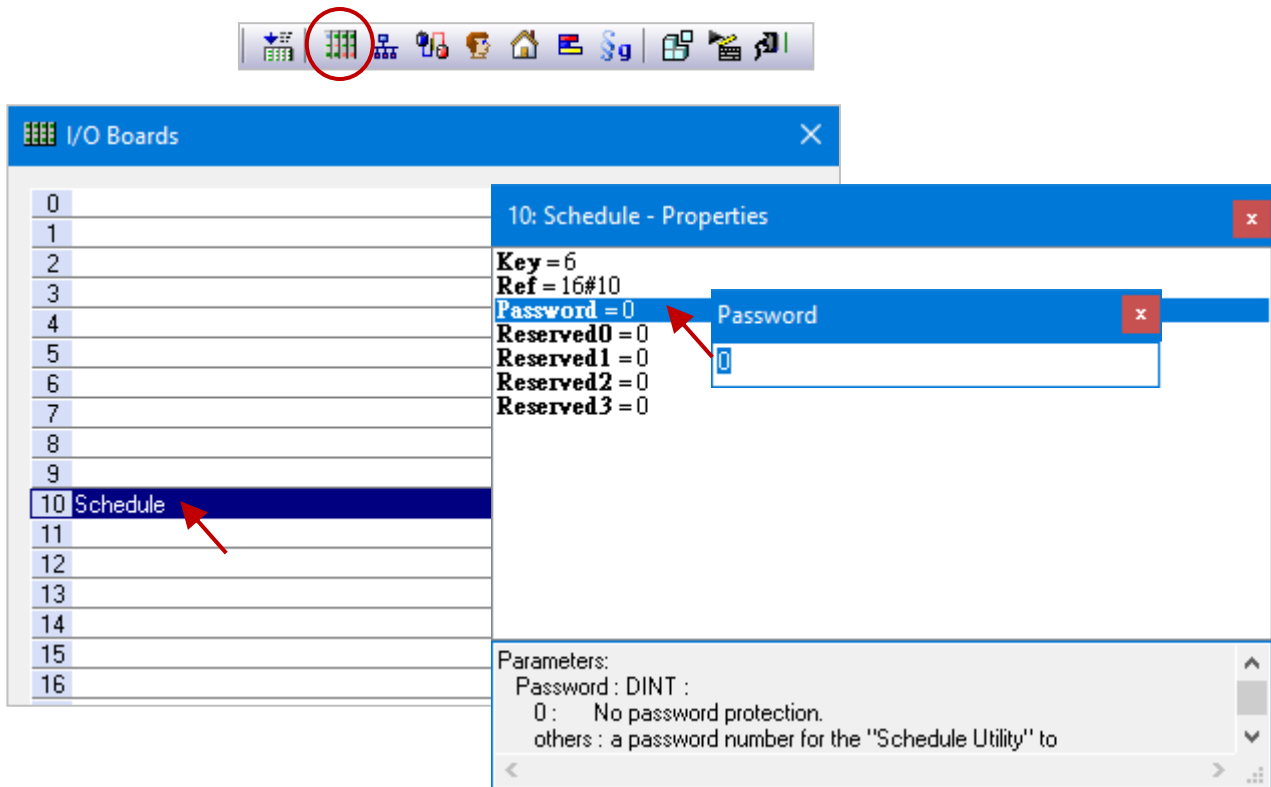


滑鼠雙擊 "Variable" 可查看/設定變數。

變數名稱	資料型態	說明
Target_01_BOOL ~ Target_10_BOOL	BOOL	用來控制 10 個 Target 的 Boolean 變數
Target_01_DINT ~ Target_10_DINT	DINT	用來控制 10 個 Target 的 DINT 變數
Target_01_REAL ~ Target_10_REAL	REAL	用來控制 10 個 Target 的 REAL 變數
Year1	DINT	用於 "TIME_GET" 功能方塊
Month1		
Day1		
WeekDay1		
Hour1		
Minute1		
Second1		

15.2.1 "I/O Boards" 設定 (Schedule)

要在 PAC 啟用 Schedule-control 功能，需在 "I/O Board" 視窗內加入 "Schedule" 功能。(參考第 4 章)



參數說明:

Password: DINT

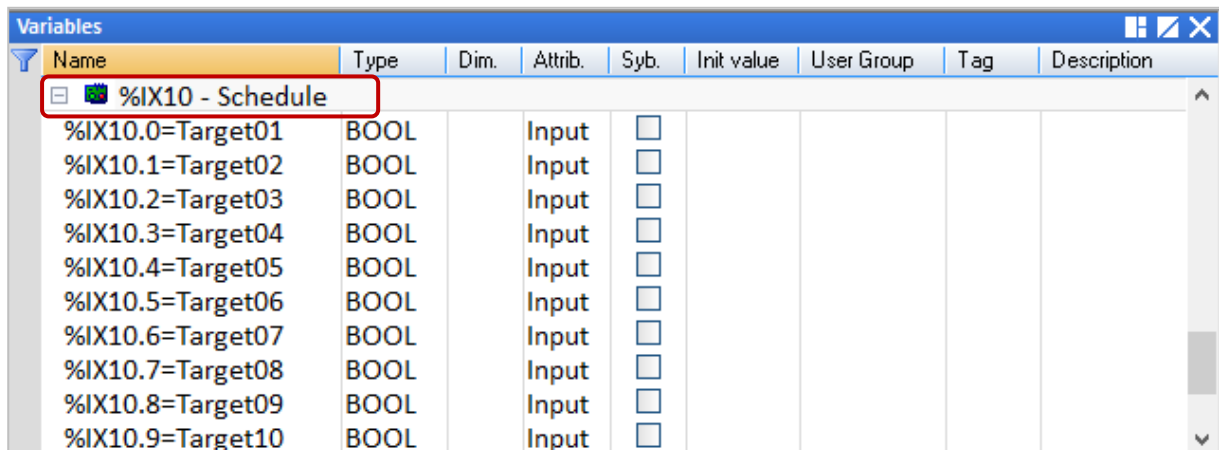
0: 無密碼保護。 其它: "Schedule Utility" 要存取 Win-GRAF PAC 的密碼。

10-Ch Boolean Input:

用來表示啟用/停用對應 Target 的排程控制。(TRUE: 啟用; FALSE: 停用)

例如: Ch.0 與 Ch.5 回傳 TRUE，表示啟用 Target 1 與 Target 6 的排程控制。

此例密碼為 0，若有設定密碼，使用 "Schedule-Control Utility" 將排程設定下載到 PAC 之前，會要求輸入密碼。加入 "Schedule" I/O Board 後，"Variables" 視窗內會自動新增 10 個 "BOOL" 輸入變數，用來顯示排程控制的狀態。

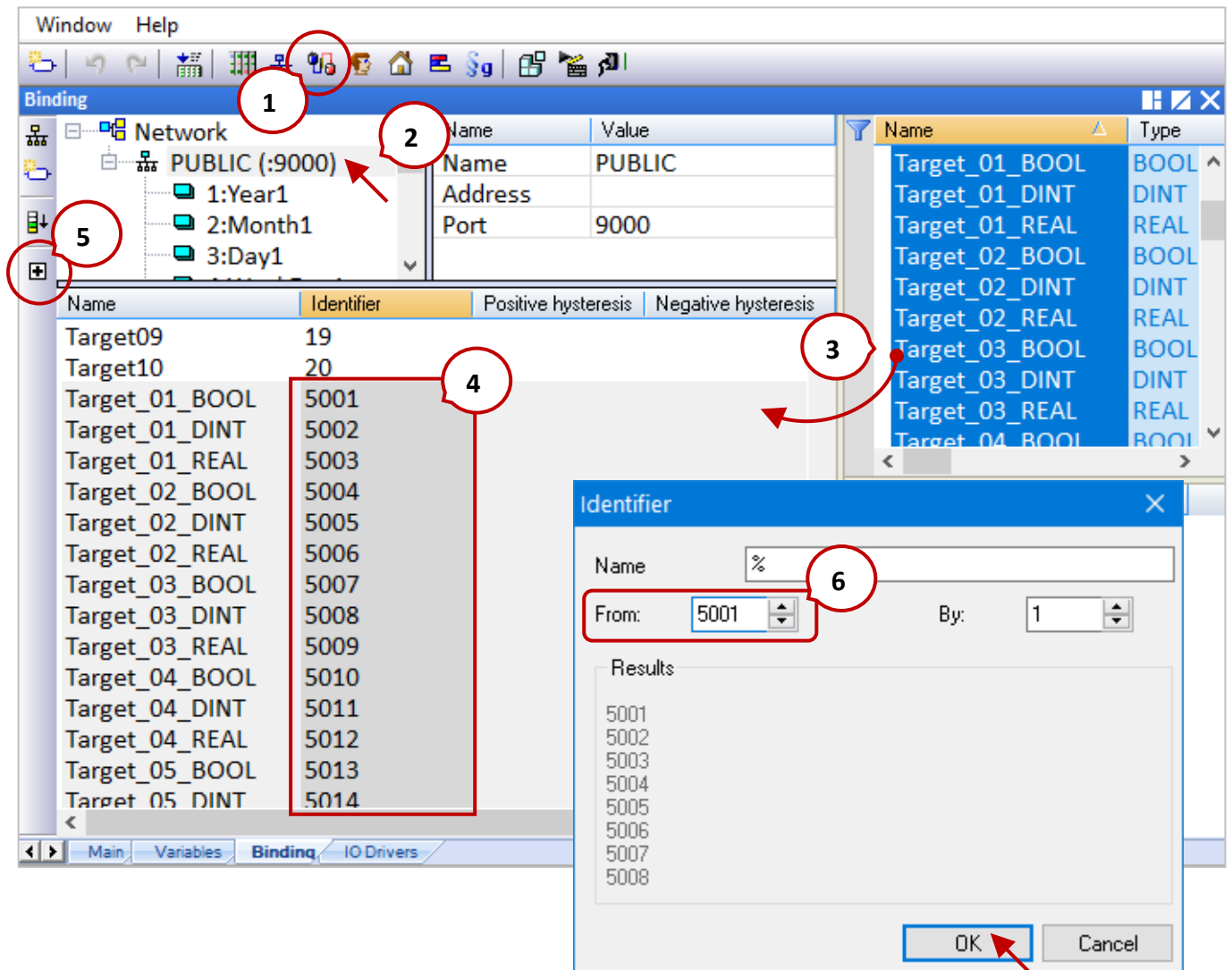


15.2.2 開放變數資料 (Data Binding)

若想開放 Win-GRAF 變數，給其它台 PAC 讀/寫資料，請在 “Binding” 視窗內加入所需的 PUBLIC 變數，並指定好識別碼 (Identifier) (可參考第 7 章)，而控制變數的 ID 必需是 5001 ~ 5030。

請依照以下步驟：

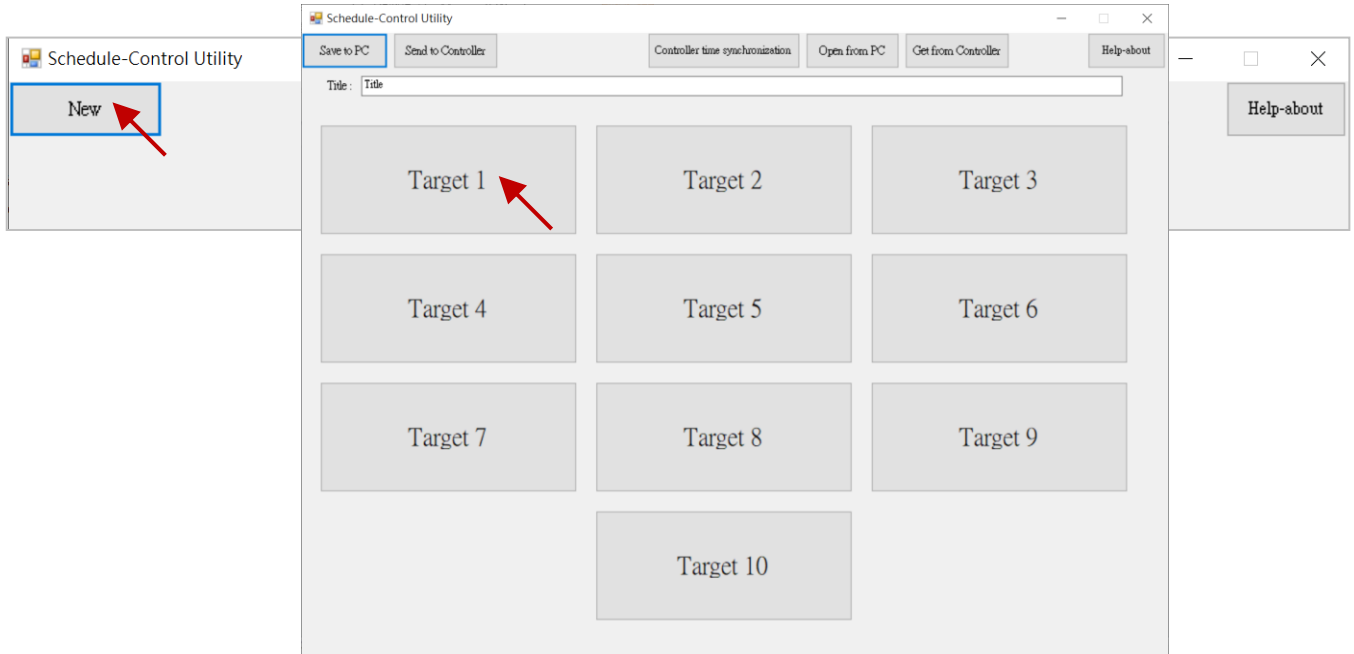
1. 點選 “Open Binding Configuration” 按鈕，開啟 Binding 視窗。
2. 點選 “PUBLIC (:9000)” 設定。
3. 將要開放的變數拖曳到 ID 對應區。
4. 選取需設定連續編號的 Identifier 欄位。
5. 點選 “Iterate Property” 按鈕。
6. From 欄位輸入起始 ID 編號。



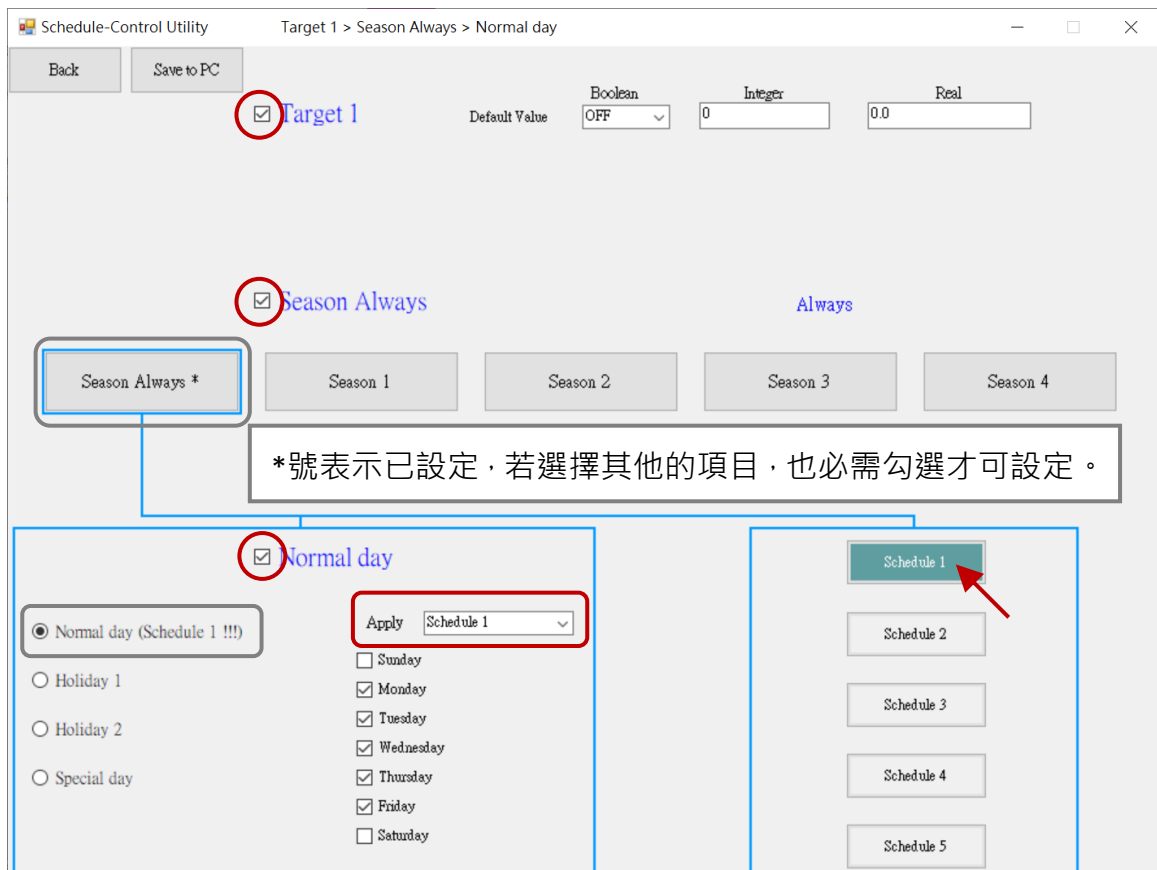
15.3 Schedule-Control Utility 範例說明

此範例簡單地介紹設定 Target 1 的方式，可參考 15.5 節了解更多的設定。

1. 開啟 Schedule-Control Utility (“Schedule_in_PC.exe”)，接著點選 New，再點選 “Target 1” 按鈕來進入設定畫面。



2. 勾選 Target 1、Season Always 與 Normal day 來啟用設定。
3. 選取想使用的排程選項 (例如: Schdule 1)，再點選 "Schdule 1" 按鈕來開啟設定視窗。



4. 每個排程表最多可以設定 15 個時段，請勾選再設定所需的時段與控制變數的狀態。設定完成後，請點選“Save and exit”按鈕，來儲存並離開視窗。

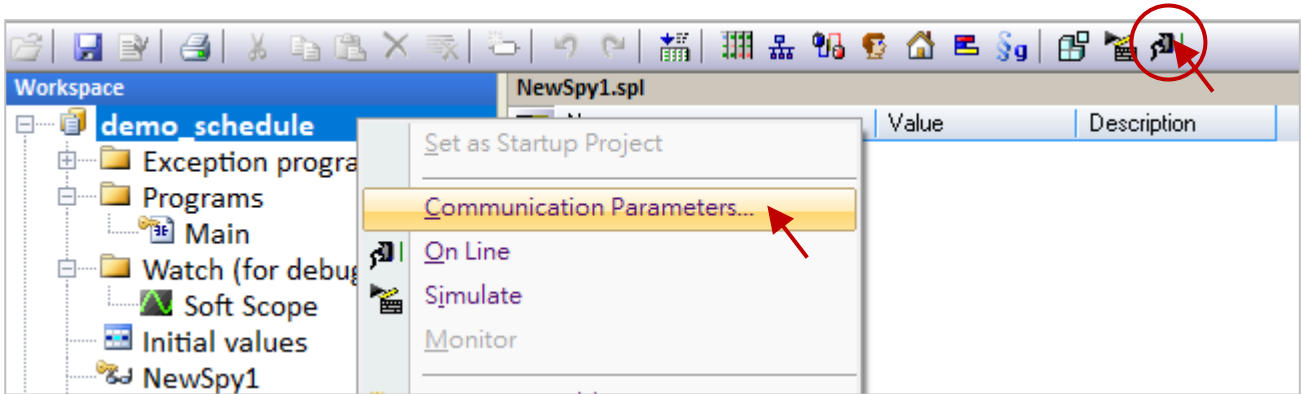
	Hour	Minute	To	Hour	Minute	Boolean	Integer	Real
<input checked="" type="checkbox"/> 01:	8	30		12	0	ON	10	12.35
<input checked="" type="checkbox"/> 02:	13	0		17	30	ON	20	27.5
<input type="checkbox"/> 03:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 04:	0	0		0	0	OFF	0	0
<input type="checkbox"/> 05:	0	0		0	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

5. 若時間不在排程時段內，則會套用“Default Value”的設定。請點選“Save to PC”按鈕，來儲存設定檔 (例如: test1.txt)。

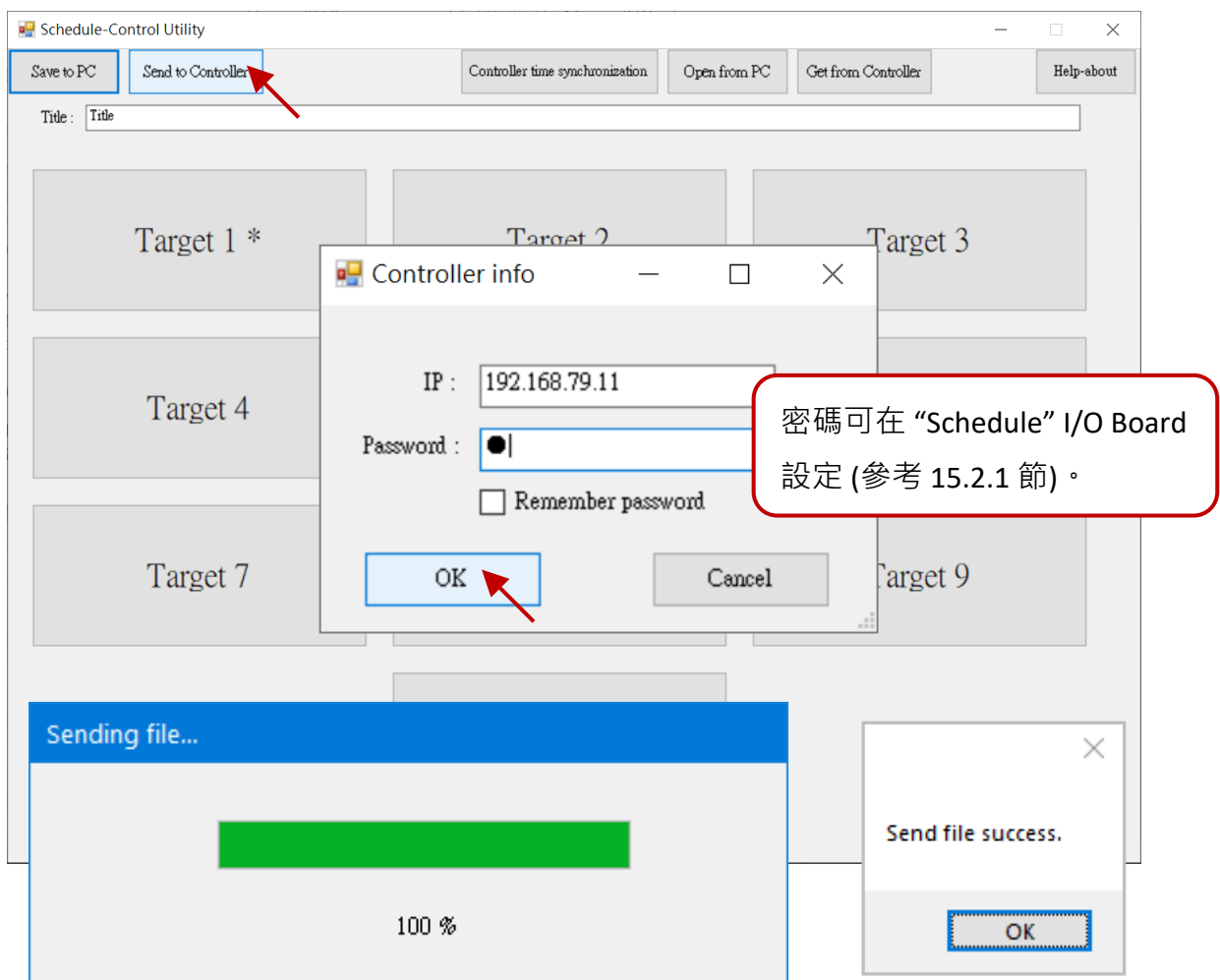
15.4 測試 PAC 內的排程控制

測試前，請將排程設定與 Win-GRAF 專案，下載到 Win-GRAF PAC。

1. 請先將 Win-GRAF 專案 (demo_schedule) 下載至 PAC。
修改通訊 IP 位址並點選 **On Line** 按鈕來下載專案。



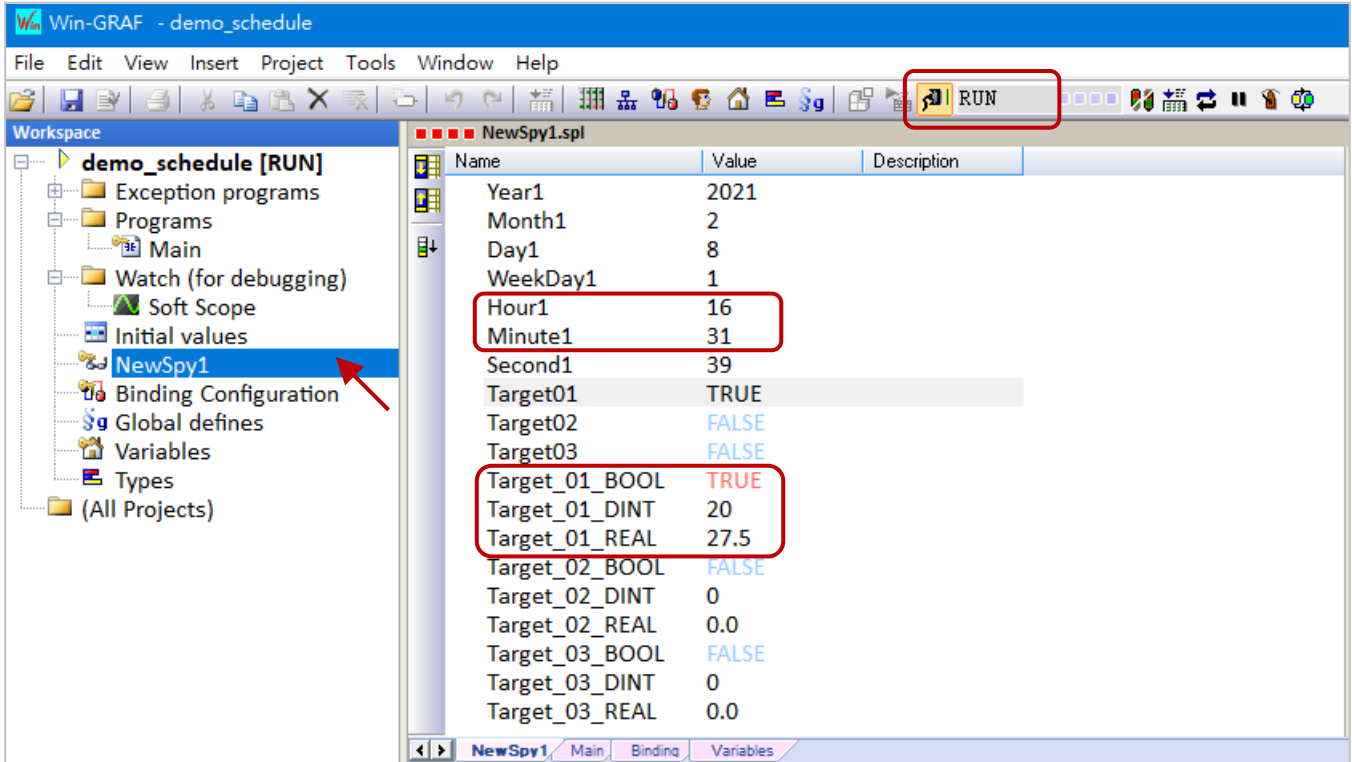
2. 使用 Schedule-Control Utility 將排程設定下載到 PAC。
點選 “Send to Controller” 按鈕，並輸入 PAC 的 IP 位址 與 排程功能的密碼 (此例為 0)，再按 OK。
傳送檔案之後，會跳出 “Send file success” 視窗，且 Utility 會自動關閉。



3. 測試排程控制：

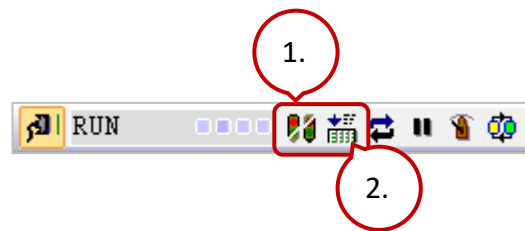
開啟 Win-GRAF 的 "NewSpy1" 視窗，您可見到 "Target_01 = TRUE" 表示啟用 **Target 1**。此例，設定 2 個排程時段：

- 01: 08:30 ~ 12:00 Boolean=ON, Integer=10, Real=12.35
- 02: 13:00 ~ 17:30 Boolean=ON, Integer=20, Real=27.5



注意:

若有變更排程設定並下載到 PAC，您可重開機 或在 Win-GRAF 點選 "Stop application" 再點選 "Download" 來讓設定生效。



15.5 如何使用 Schedule-Control Utility

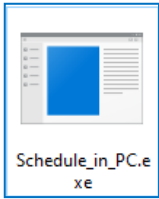
15.5.1 BOOL、DINT、REAL 控制變數的位址

Schedule-Control Utility 最多可配置 10 個 Target，而每個 Target 可控制 1 個 BOOL 變數、1 個 DINT 變數與 1 個 REAL 變數。

若想開放以下變數讓其他設備可存取資料，請在 "Variables" 視窗先宣告變數 (參考 15.2 節)，並在 Binding 視窗指定這些變數與識別碼 (Identifier)，位址必需是 5001 ~ 5030 (參考 15.2.2 節)。

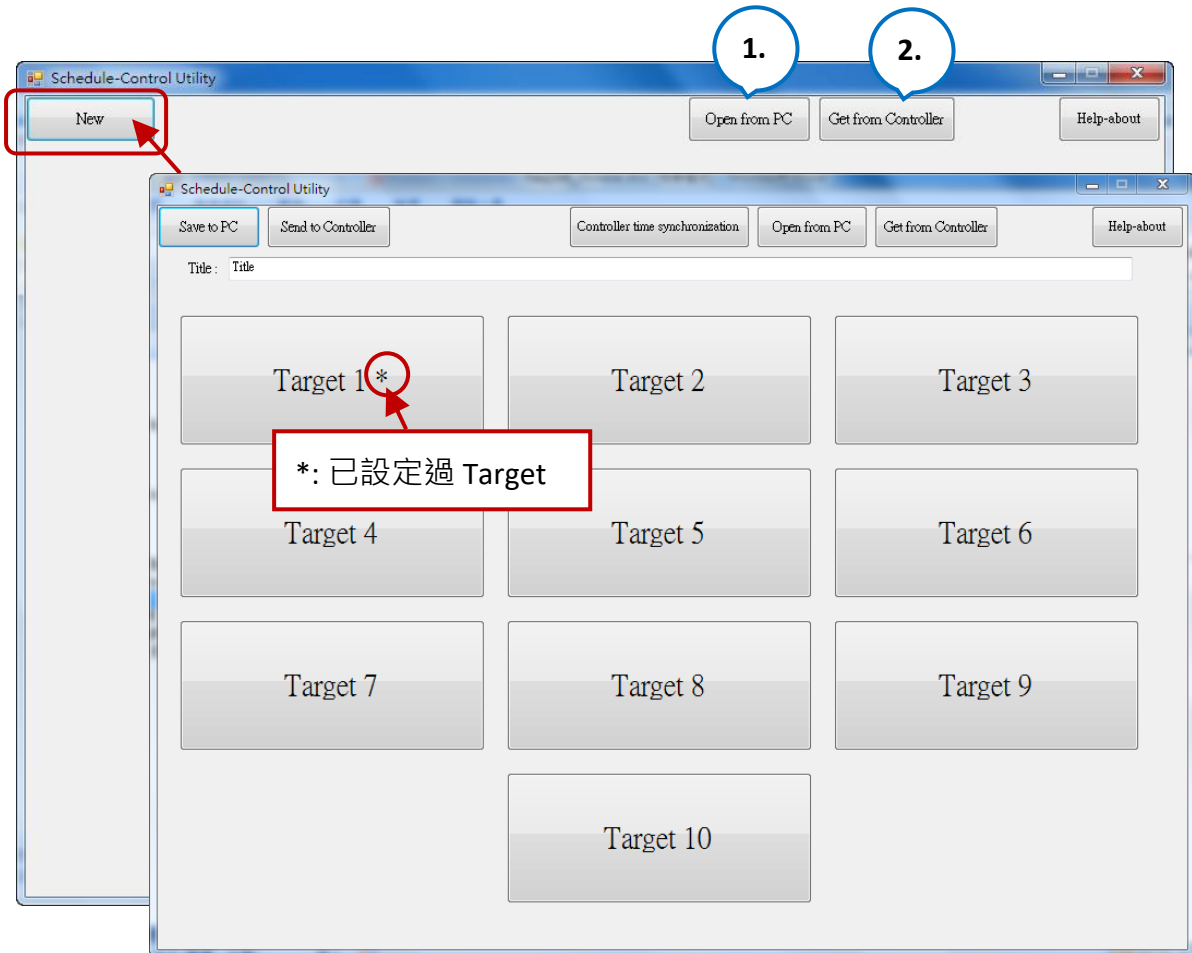
Address	型態	說明	Address	型態	說明
5001	BOOL	Target 1 控制的 BOOL, DINT 與 REAL 變數	5016	BOOL	Target 6 控制的 BOOL, DINT 與 REAL 變數
5002	DINT		5017	DINT	
5003	REAL		5018	REAL	
5004	BOOL	Target 2 控制的 BOOL, DINT 與 REAL 變數	5019	BOOL	Target 7 控制的 BOOL, DINT 與 REAL 變數
5005	DINT		5020	DINT	
5006	REAL		5021	REAL	
5007	BOOL	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	

15.5.2 設定 Target



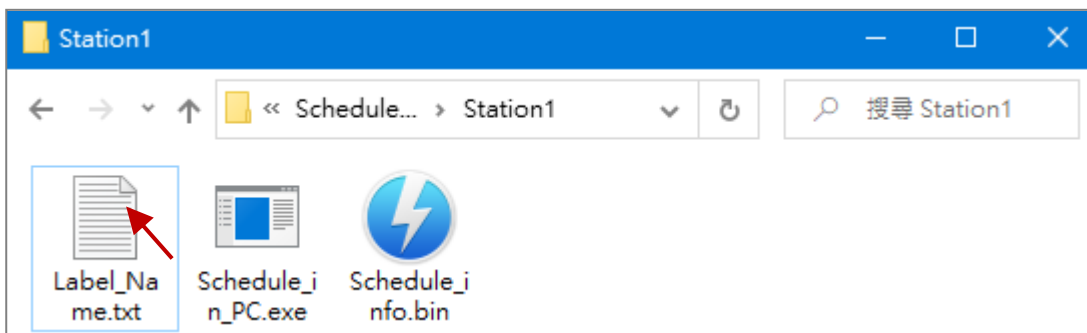
請先開啟 Schedule-Control Utility，並點選 "New" 按鈕來顯示設定畫面。每台 Win-GRAF PAC 最多可設定 10 個 Target，預設名稱為 Target 1 ~ Target 10。

- 1) **Open from PC:** 開啟現存的設定檔。
- 2) **Get from Controller:** 載入 PAC 目前的設定。



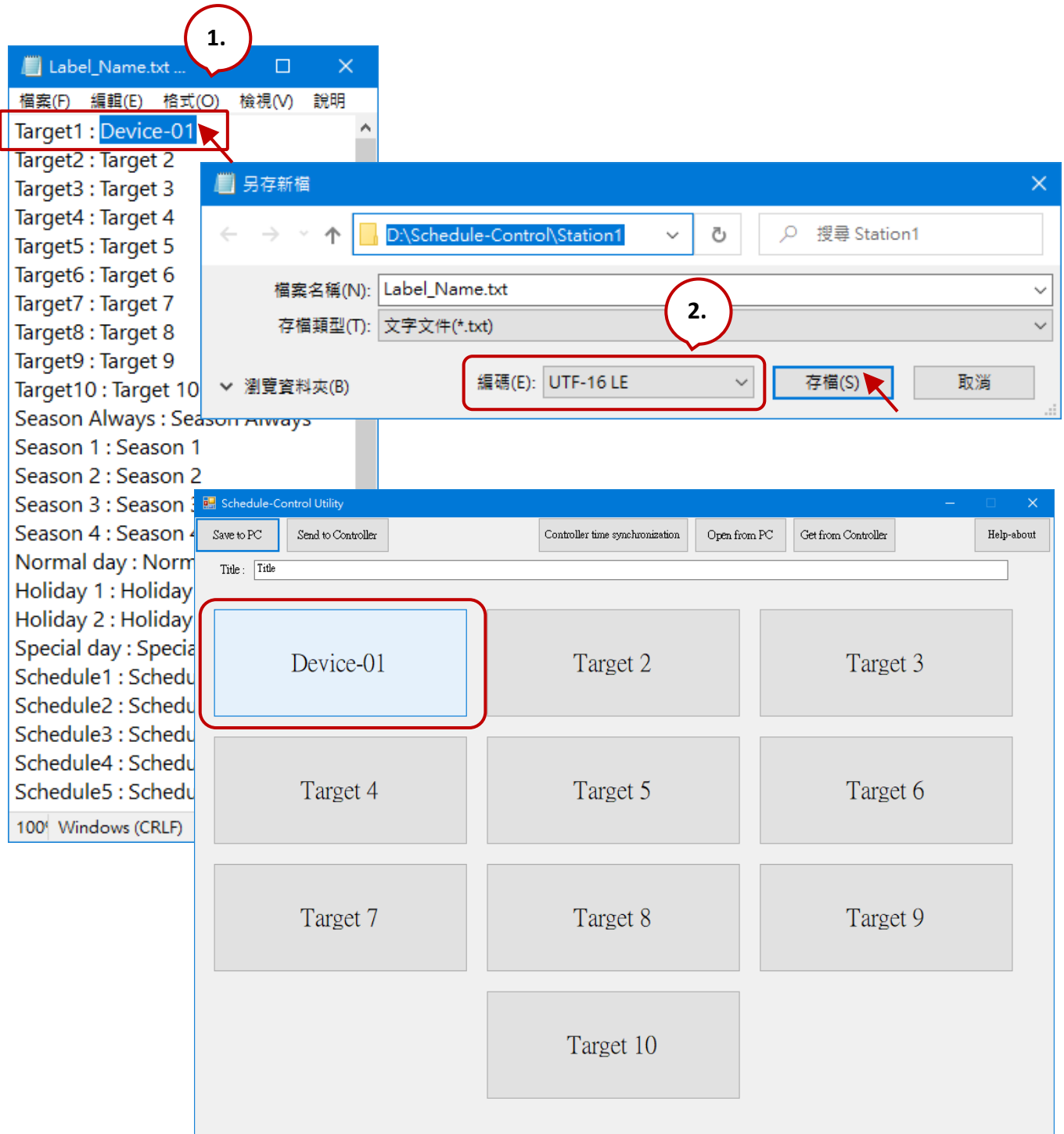
變更 Target 名稱

在專案目錄中 (例如，D:\Schedule-Control\Station1\) 有提供一個 "Label_Name.txt" 檔案，使用者可依照應用需求來變更 Target、Season、... 等名稱。若此檔案不存在，畫面上會顯示預設的名稱。



編輯 “Label_Name.txt” 檔案：

1. 請在冒號 (":") 後，輸入想變更的名稱 (例如: “ Device-01 ”，頭尾空白會自動刪除)。
2. 使用記事本編輯完成後，需儲存為 “Unicode” 編碼格式。
3. “Label_Name.txt” 與 “Schedule_in_PAC.exe” 需放在相同的目錄中。

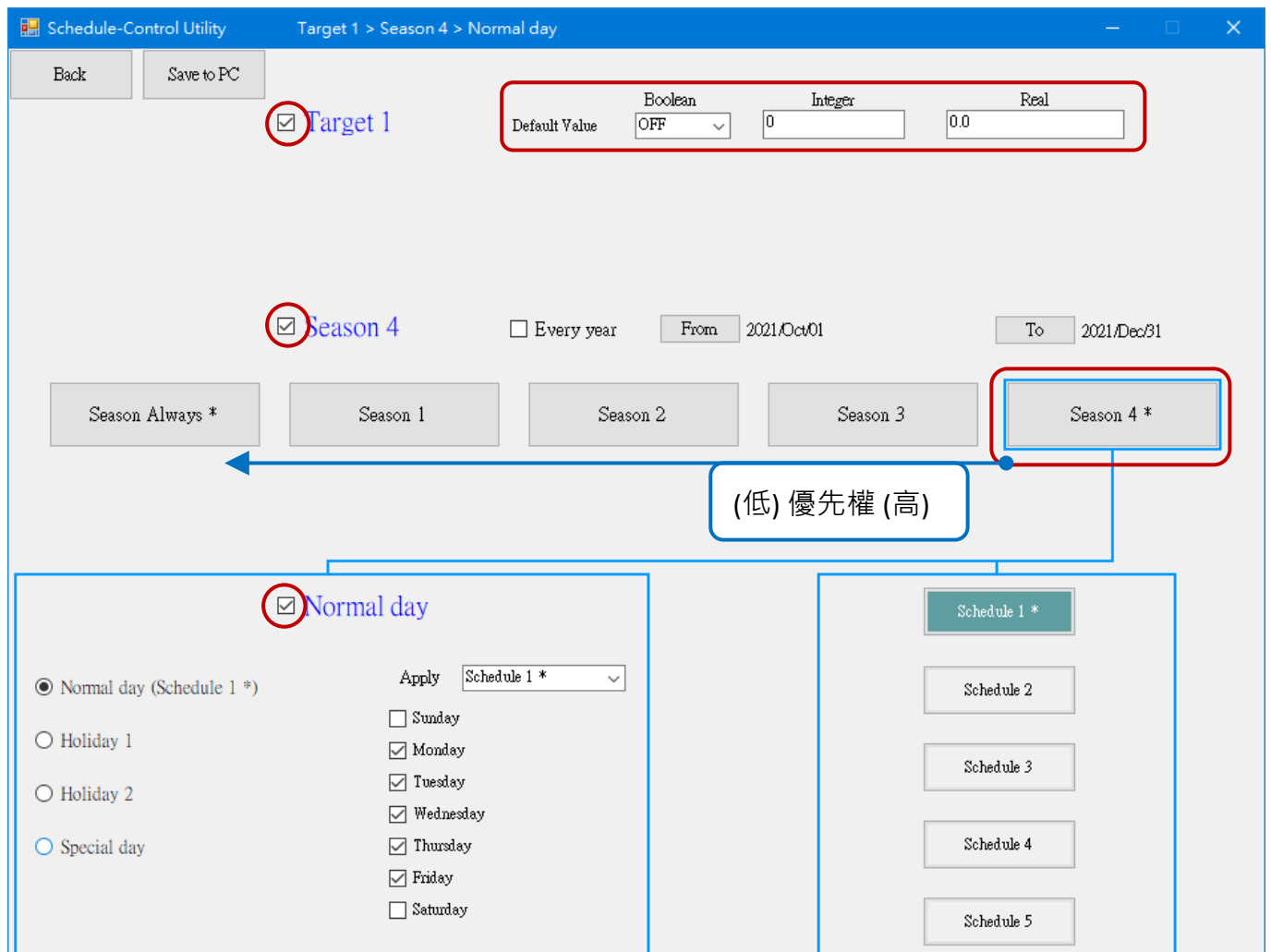
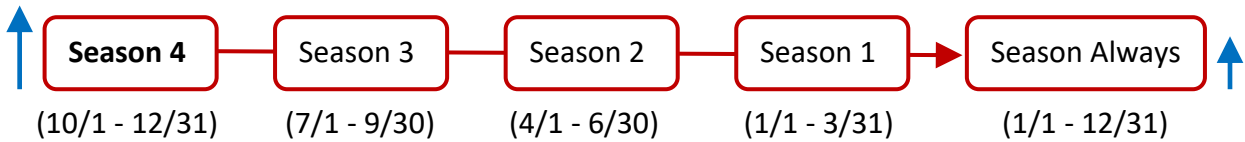


15.5.3 設定 Season

每個 Target 內提供 Season Always、Season 1、Season 2、Season 3 與 Season 4 設定，用來設定全年或四季的排程。(註: 勾選核取方塊後，才可進行設定。)

Season 的搜尋順序:

一般建議啟用 **Season Always** 設定，若啟用多個 Season 設定，PAC 會依照以下優先順序，來套用 Boolean、Integer 與 Real 設定。若找不到設定，則會使用預設值 (OFF, 0, 0.0)。



Season 設定:

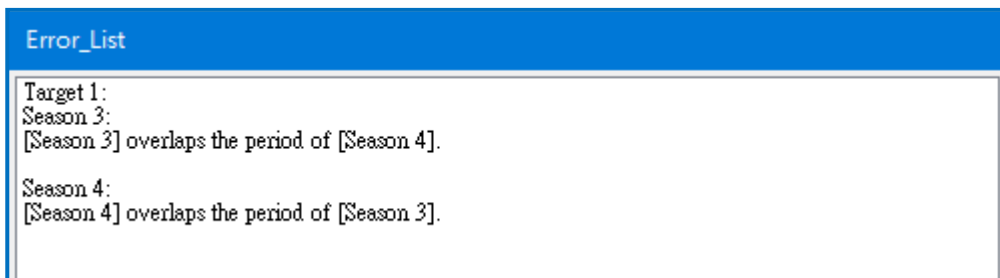
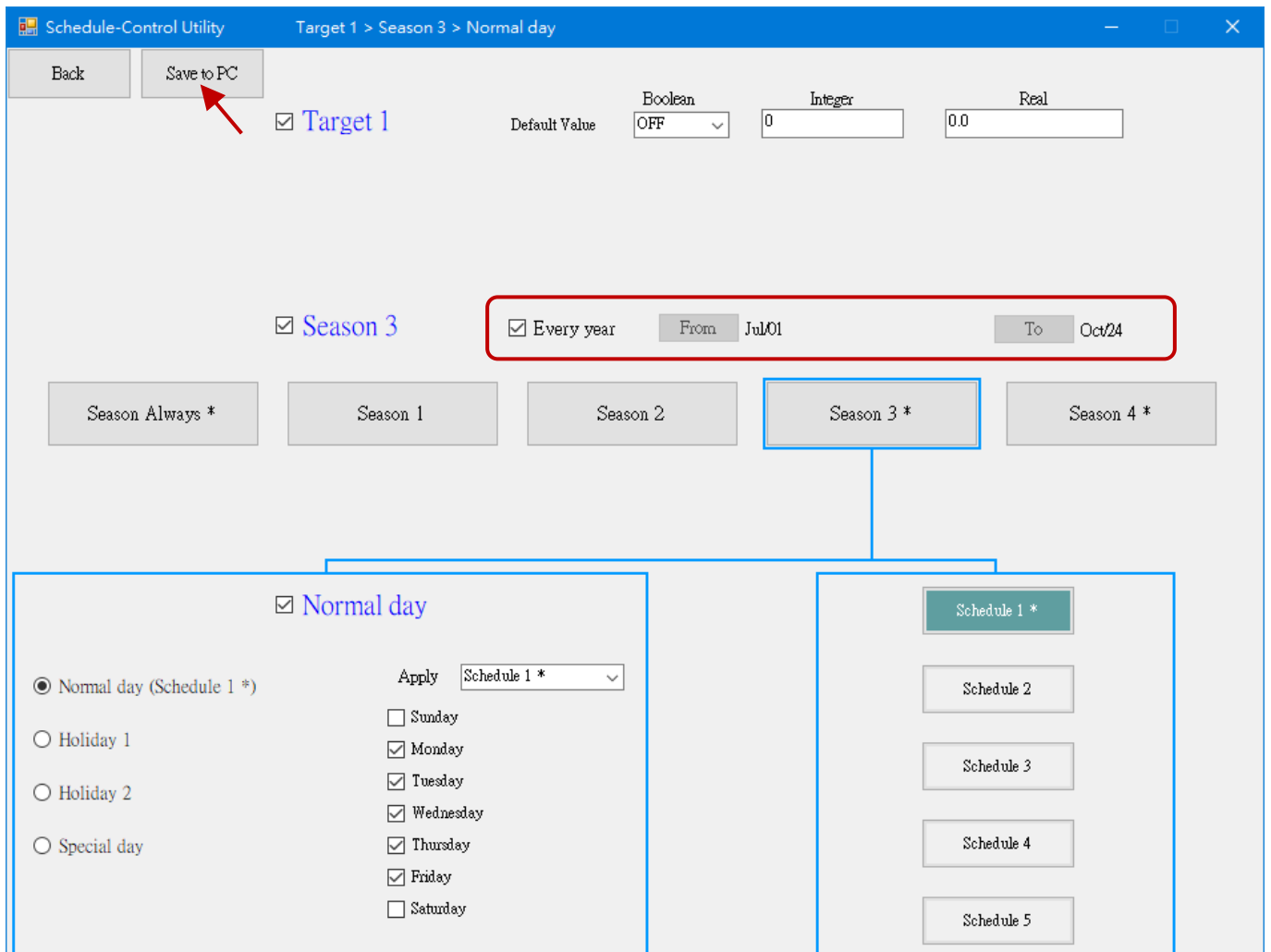
若要啟用 Season (1 ~ 4) 需設定它的“日期區間”，您可勾選“Every year”表示每年都套用此設定。

注意:

1. Season 1 ~ 4 的日期區間不可重疊，且起始日期 (From) 需早於結束日期 (To)。
2. 取消勾選 “Every year” 後，才可修改日期。

例如:

Season 4 的日期區間為 10/1 ~ 12/31，若設定 Season 3 為 7/1 ~ 10/24，則點選 **Save to PC** 存檔時，會顯示錯誤訊息。

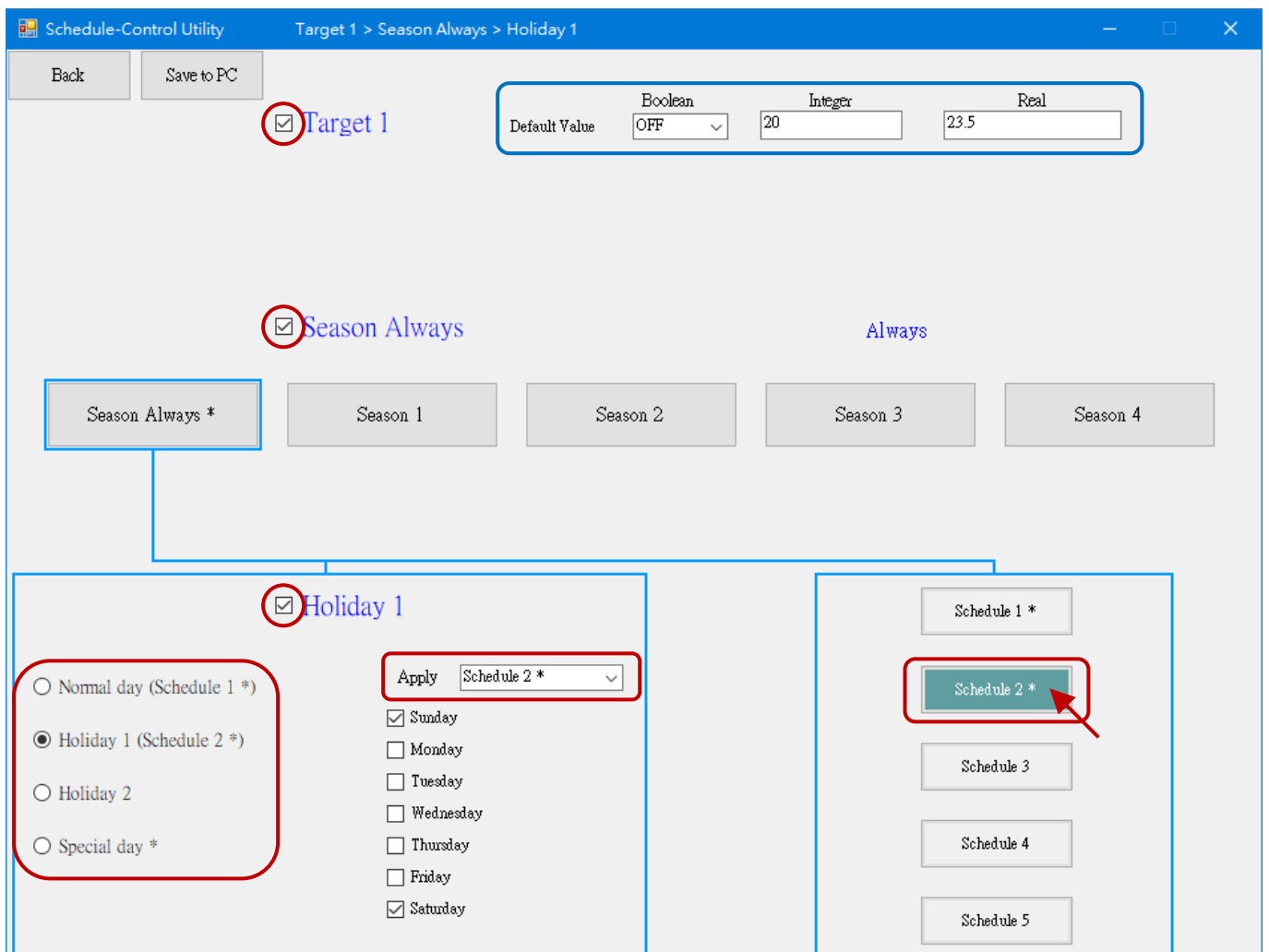
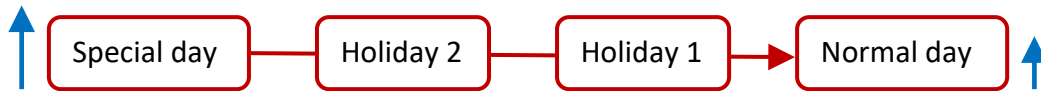


15.5.4 設定 Normal Day / Holiday / Special Day

每個 Season 內提供 Normal day、Holiday 1、Holiday 2 與 Special day 設定，用來設定工作日、假日或特定日的排程。(註: 勾選核取方塊後，才可進行設定。)

Day 的搜尋順序:

若啟用多個 Day 設定，PAC 會依照以下優先順序，來套用 Boolean、Integer 與 Real 設定。若找不到設定，則會使用預設值 (OFF, 0, 0.0)。



Normal day	工作日，一般是星期一~五	若啟用 Day 設定，至少需設定一個 Schedule (1 ~ 5)。
Holiday 1	假日 1，一般是星期六、日。	
Holiday 2	假日 2，可指定星期一~日。	
Special Day	特定日，可指定日期 (例如: 10 月 10 日、7 月 4 日...等)。一個 Season 內，最多可指定 50 個特別日。	

注意: Normal day、Holiday 1、Holiday 2 設定的日不可重疊。例如: Normal day 與 Holiday 1 都勾選了 Wednesday，則儲存設定時會跳出錯誤訊息。

The screenshot shows two overlapping configuration windows. The left window has 'Normal day' selected, and the right window has 'Holiday 1' selected. Both windows have 'Wednesday' checked under the 'Apply' dropdown menu. Below the windows is an 'Error_List' box with the following text:

```

Target 1:
Season Always:
Both [Normal day] and [Holiday 1] enable Wednesday.
Both [Holiday 1] and [Normal day] enable Wednesday.
    
```

指定 Special Day 的日期:

“Special Day” 可用來指定日期，每個 Season 最多可以設定 50 個日期。

The screenshot shows the 'Special day' configuration interface. The 'Special day' option is selected. A 'Date Setting' dialog box is open, showing a calendar for February 2021 with the date 2/9 selected. The dialog has four numbered steps:

1. Click 'Add a new date'.
2. Select the date on the calendar.
3. Click 'Apply'.
4. Click 'Save Setting'.

15.5.5 設定 Schedule 時段

每個 Season 最多可以設定 5 個 Schedule (1 ~ 5)，每個 Schedule 最多可設定 15 個時段。時間的最小單位為分鐘，必需在 00:00 ~ 24:00 的範圍內。(註: 勾選核取方塊後，才可進行設定。)

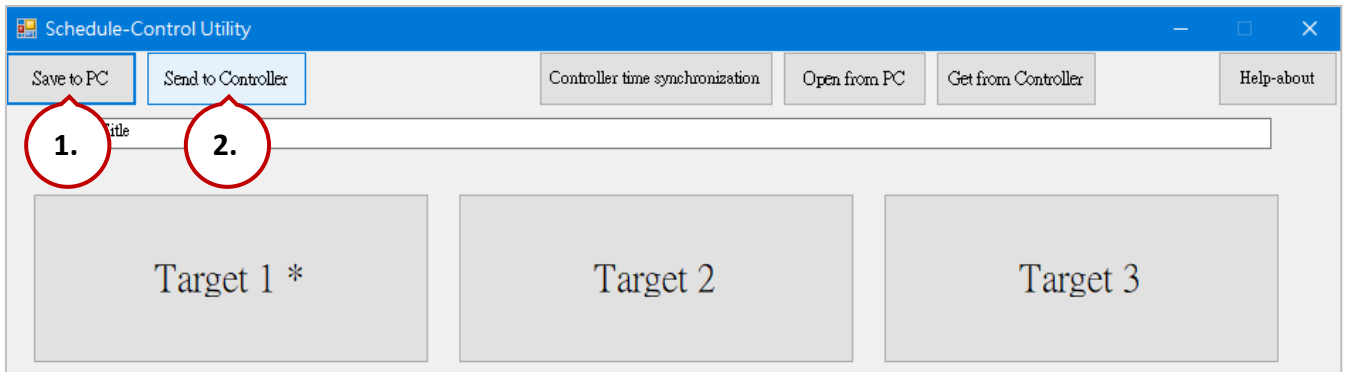
排程時段的搜尋順序:

若啟用多個時段，PAC 會先搜尋編號較大的時段，來套用 Boolean、Integer 與 Real 設定。若時間重疊會採用編號較大的設定，若找不到設定，則會使用預設值 (OFF, 0, 0.0)。

	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

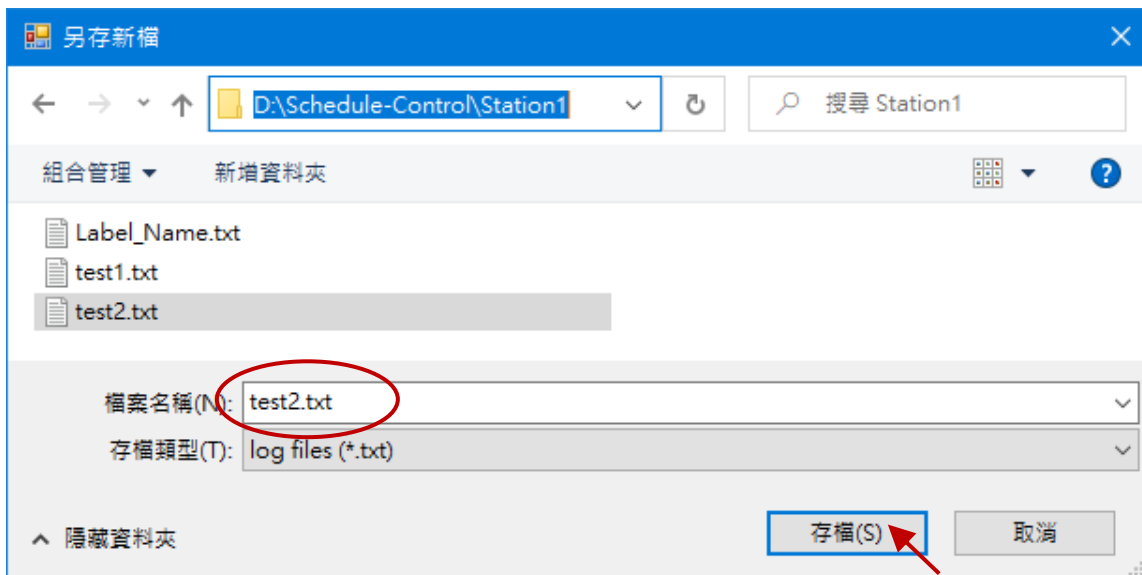
15.5.6 儲存或傳送設定檔

完成排程設定後，請先儲存一份設定檔至 PC，再傳送至 PAC 中。



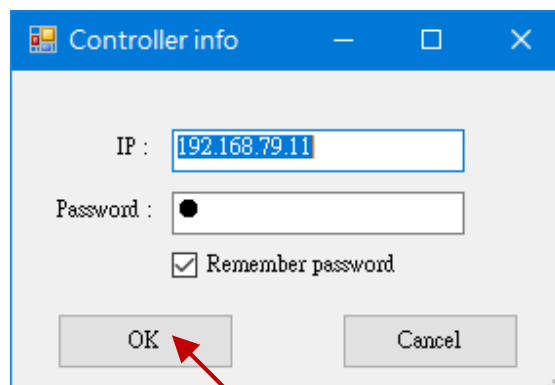
1. 儲存設定檔至 PC

點選 “Save to PC” 按鈕，在 PC 儲存設定檔 (“*.txt”)。



2. 傳送設定檔至 PAC

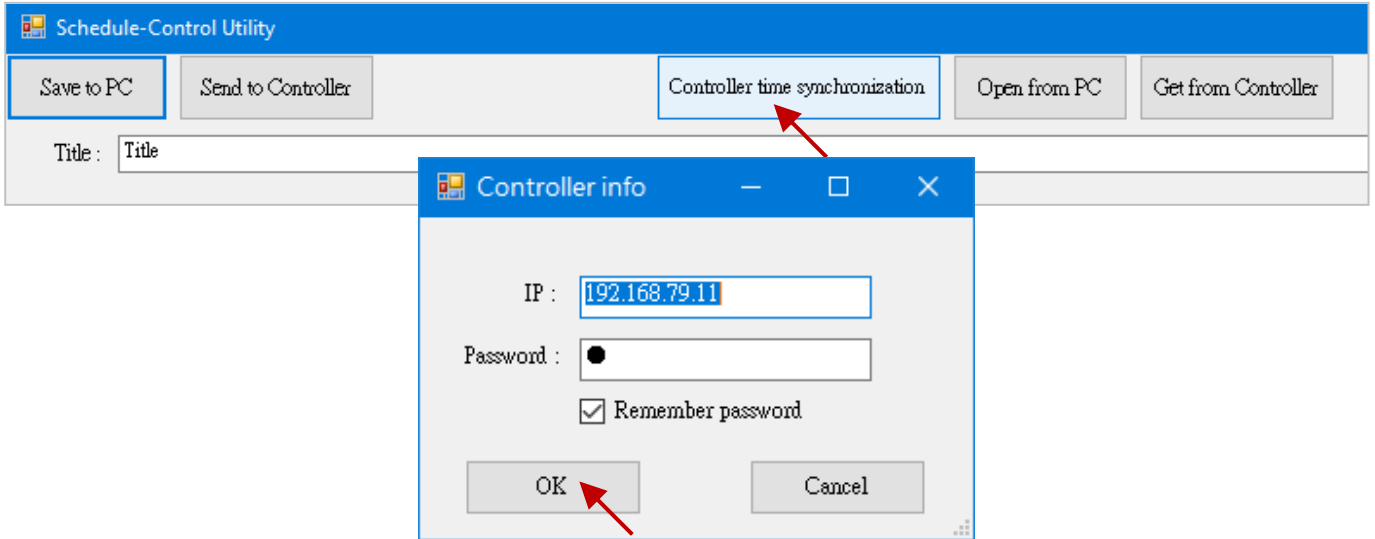
點選 “Send to Controller” 按鈕，來傳送設定到 PAC。請填入 PAC 的 IP 位址與密碼 (預設: 0)。



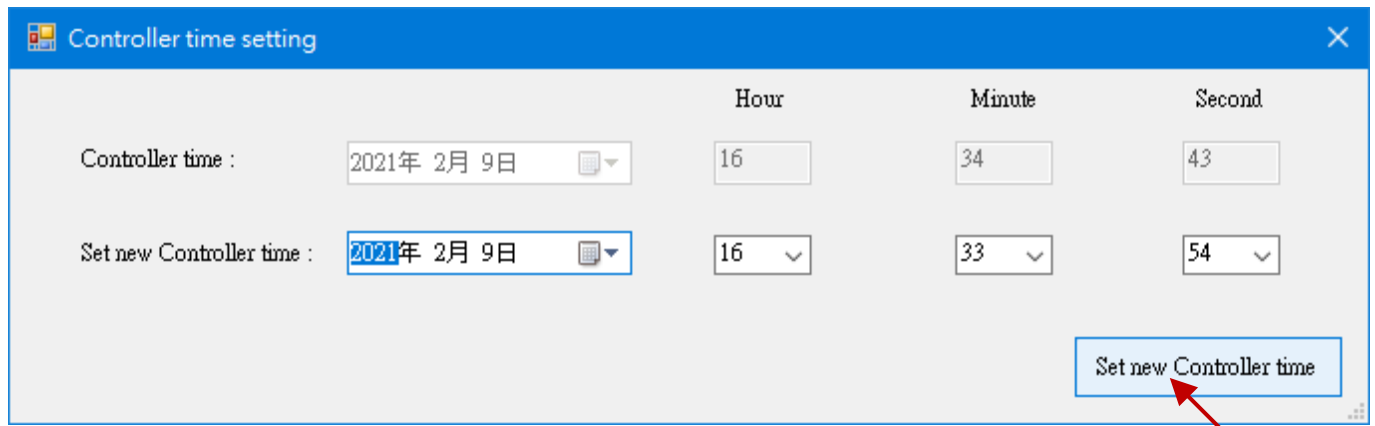
15.5.7 時間同步 (Time Synchronization)

若您發現 PAC 的時間與目前時間有誤差，為了可準時地進行排程控制，請執行 Schedule-Control Utility 的校時功能來修正 PAC 的時間。請確認 PC 與 Win-GRAF PAC 皆連上網路。

1. 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。



2. 確認日期、時、分、秒，再點選 "Set new Controller time 按鈕" 來同步時間。



第 16 章 Win-GRAF 簡訊功能

本章節使用 RPAC-2568M 搭配 [GTM-204M-4GE](#) 無線通訊模組來發送/接收簡訊。

注意:

1. 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。
2. 請在 GTM-204M-4GE 的[下載區](#)，參考 Quick Start 來安裝 USB 驅動程式，並進行快速測試。

1. 點選右鍵 – 內容 – 連接埠設定，可查看/設定 Baud Rate。

若您的電腦無超級終端機，仍可使用 PuTTY 進行測試。

2. 設定 Serial line 為 COM7，Speed 為 115200。

3. 輸入 AT 並按 Enter，出現 OK 表示測試成功。(若未顯示輸入文字，請將 Terminal 內的 Local echo 設為 Force on。)

Line discipline options:
Local echo: Auto Force on Force off
Local line editing: Auto Force on Force off

16.1 "GSM_Open"、"Send_SMS" 與 "Read_SMS" 函式

您可使用以下 3 個函式，在 Win-GRAF PAC 內處理收發簡訊的功能。

"GSM_OPEN" 函式：用來開啟/關閉 GSM 模組。



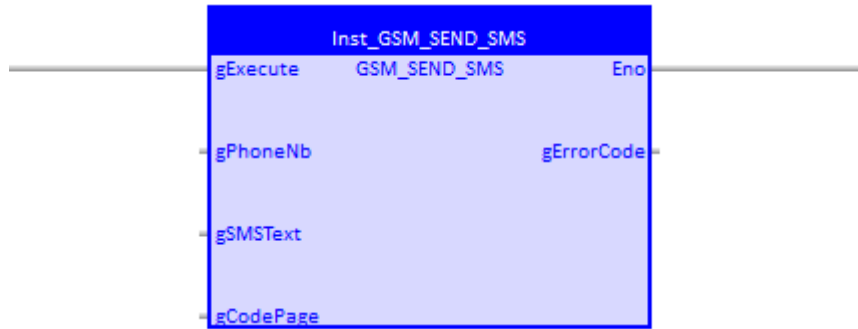
使用小技巧:

按“F1”鍵，可查看詳細的設定說明。

輸入參數	資料型態	說明
gEnable	BOOL	TRUE: 開啟指定的 COM Port 去連接 GSM 模組並初始化。 FALSE: 取消連接 GSM 模組並關閉 COM Port。
gCOMPort	DINT	連接 GSM 模組的 COM Port 編號。
gPinCode	STRING	若 SIM 卡有設定 PIN Code 可填入 (例如: '0000')。

輸出參數	資料型態	說明
gIsInitialized	BOOL	TRUE: 已開啟 COM Port 並初始化 GSM 模組。 FALSE: 未開啟 COM Port 或 初始化 GSM 模組失敗。
gSignalQuality	SINT	0 ~ 31: 數值越大，訊號越佳。 99: 訊號不明或未偵測到。
gErrorCode	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)。



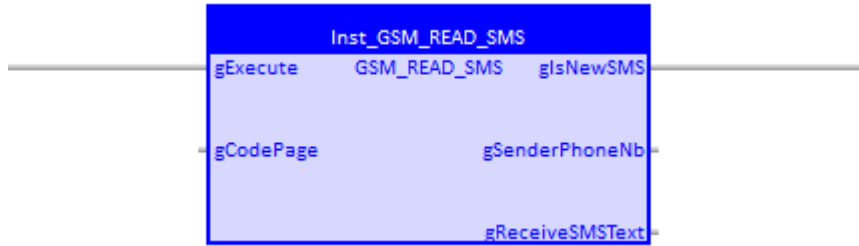
注意:

使用 "GSM_SEND_SMS" 功能之前，使用者必須先使用 "GSM_OPEN" 功能來開啟連接 GSM 模組的 PAC COM Port，否則此功能將無效。

輸入參數	資料型態	說明						
gExecute	BOOL	Pulse TRUE : 觸發以傳送簡訊。						
gPhoneNb	STRING	目的地址 (電話號碼)。						
gSMSText	STRING	簡訊內容。						
gCodePage	UDINT	文字的編碼代號。 0: 英文 866: 俄文 932: 日文 936: 簡體中文 950: 繁體中文 注意: 若超過只會傳送容許字元。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>值</th> <th>簡訊的最大長度</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>160 字元</td> </tr> <tr> <td>非 0</td> <td>70 字元</td> </tr> </tbody> </table>	值	簡訊的最大長度	0	160 字元	非 0	70 字元
值	簡訊的最大長度							
0	160 字元							
非 0	70 字元							

輸出參數	資料型態	說明
gErrorcode	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 : 無接收者的號碼。 -7 : 傳送簡訊失敗。

"GSM_READ_SMS" 函式: 用來透過 GSM 模組讀取簡訊 (SMS)。



注意:

使用 "GSM_READ_SMS" 功能之前，使用者必須先使用 "GSM_OPEN" 功能來開啟連接 GSM 模組的 PAC COM Port，否則此功能將無效。

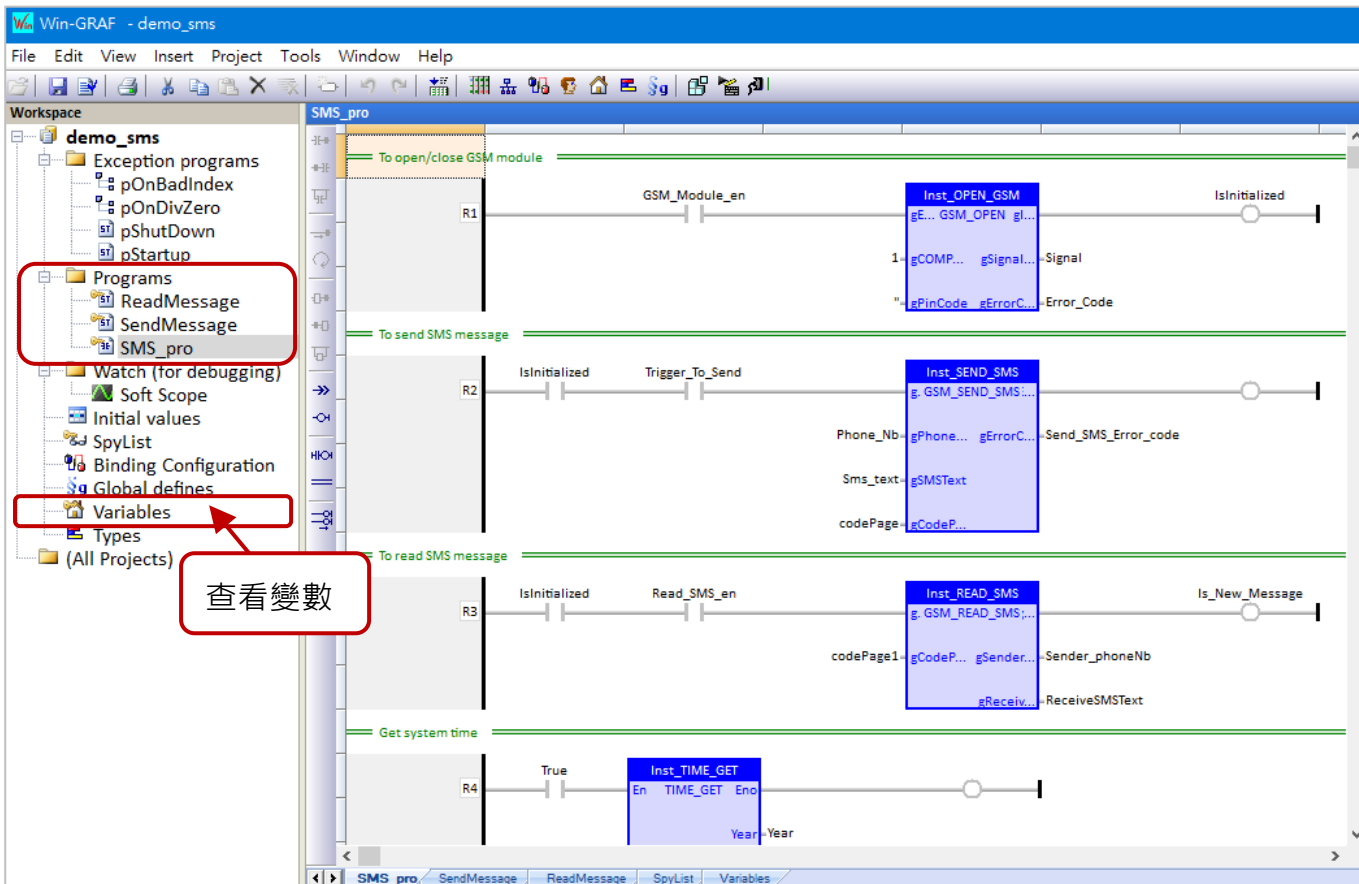
輸入參數	資料型態	說明
gExecute	BOOL	TRUE: 啟用 GSM 模組讀取簡訊的功能。 FALSE: 關閉 GSM 模組讀取簡訊的功能。
gCodePage	UDINT	文字的編碼代號。 0: 英文 866: 俄文 932: 日文 936: 簡體中文 950: 繁體中文

輸出參數	資料型態	說明
gIsNewSMS	BOOL	Pulse TRUE: 有新的簡訊傳來。
gSenderPhoneNb	STRING	來源位址 (電話號碼)。
gReceiveSMSText	STRING	簡訊內容。

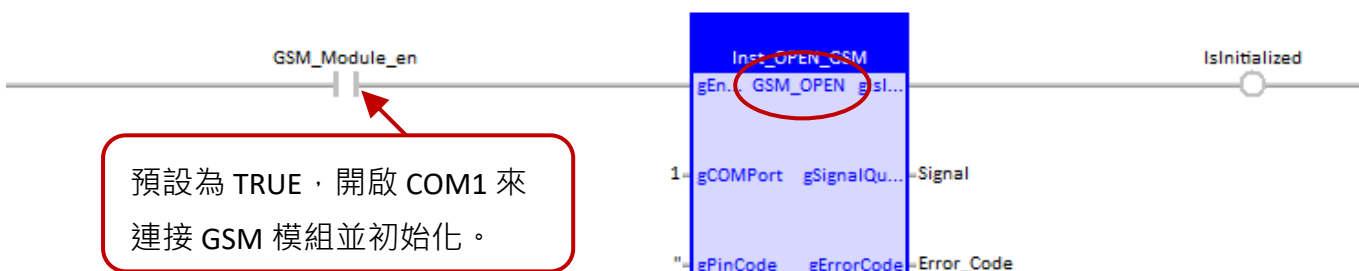
16.2 程式介紹 (Demo_SMS)

請在網站下載 [範例程式 \(demo_sms.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。(可參考 11.4 節)

此專案包含了 2 個 ST 程式 (SendMessage 與 ReadMessage) 與 1 個 LD 程式 (SMS_pro)。

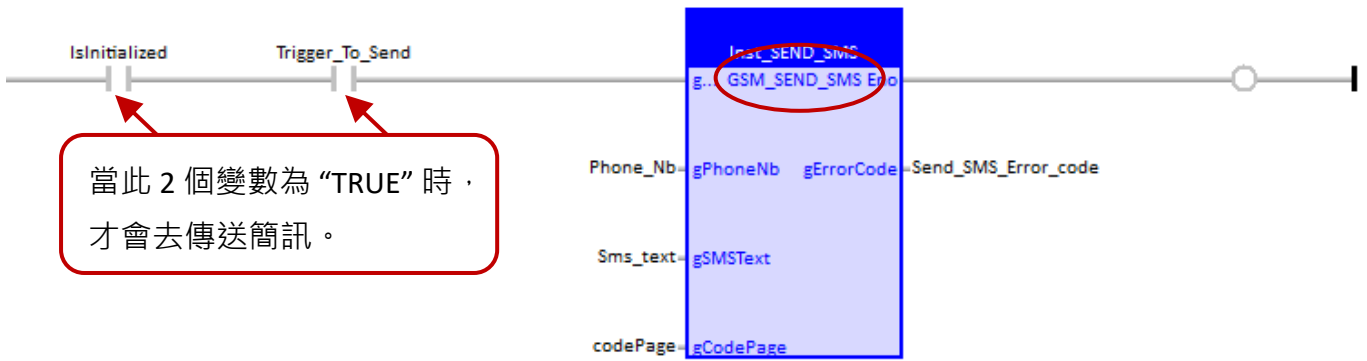


LD 程式 (SMS_pro): "GSM_OPEN" 函式 (參考 16.1 節，取得詳細說明)



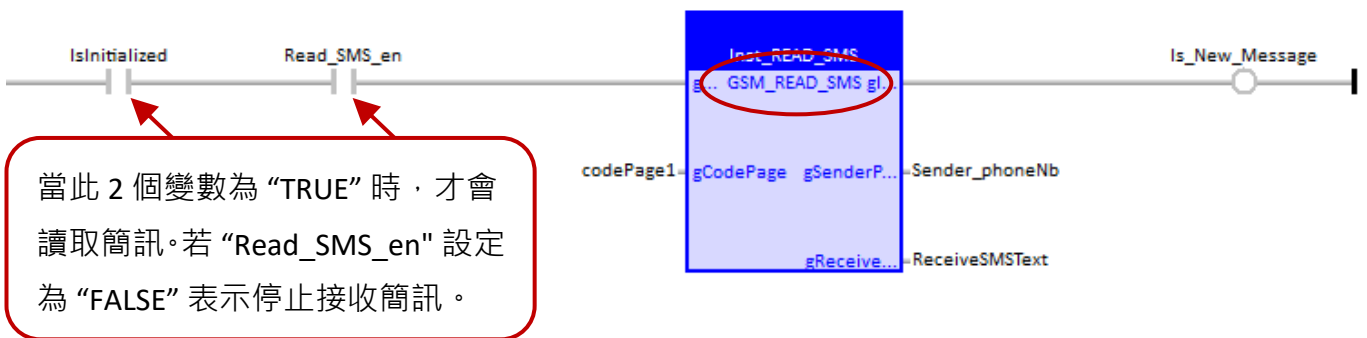
變數名稱	資料型態	說明
GSM_Module_en	BOOL	設為 TRUE 來開啟指定的 COM Port 去連接 GSM 模組並初始化。(初始值: TRUE)
Signal	SINT	GSM 模組的訊號品質。
Error_Code	INT	GSM 模組的錯誤碼。
IsInitialized	BOOL	GSM 模組初始化結果。

"GSM_SEAD_SMS" 函式 (參考 16.1 節，取得詳細說明)



變數名稱	資料型態	說明
Trigger_To_Send	BOOL	設定為 "TRUE" 以傳送簡訊。
Phone_Nb	STRING(255)	接收者的電話號碼。(初始值: '0900629879')
Sms_text	STRING(255)	簡訊的內容。 (初始值: 'This message is sent from Win-GRAF PAC')
codePage	UDINT	簡訊的編碼代號。
Send_SMS_Error_code	INT	簡訊的錯誤碼。

"GSM_READ_SMS" 函式 (參考 16.1 節)



變數名稱	資料型態	說明
Read_SMS_en	BOOL	設定為 "TRUE" 以讀取簡訊。(初始值: TRUE)
codePage1	UDINT	接收簡訊的編碼代號。(初始值: UDINT#950)
Sender_phoneNb	STRING(255)	傳送者的電話號碼。
ReceiveSMSText	STRING(255)	簡訊內容。
Is_New_Message	BOOL	回傳是否有新簡訊。

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 *)

    elsif 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; *)

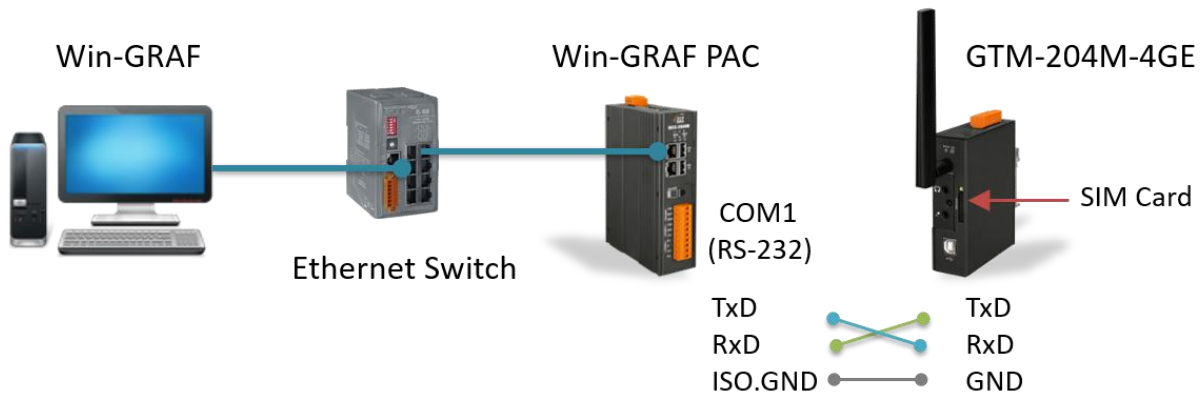
  Got_New_Message := ReceiveSMSText;
  Got_Message_from_who := Sender_phoneNb;

  (* 加入更多的操作程序 *)

end_if;
```

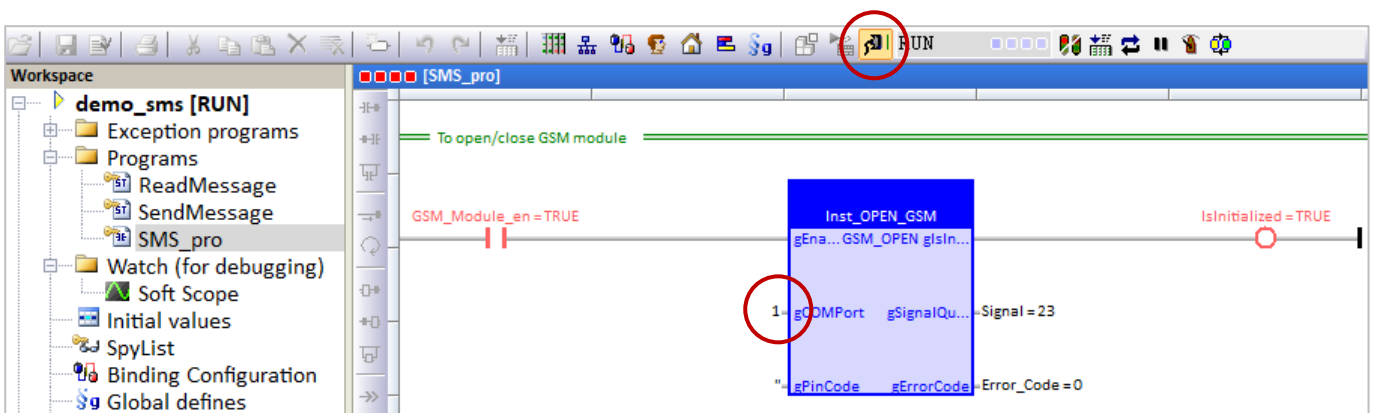
16.3 簡訊傳送測試

此例，使用 RPAC-2568M 搭配 [GTM-204M-4GE](#) 模組來發送/接收簡訊。請安裝好模組的 SIM 卡並透過 RS-232 Port 與 PAC 連接後，再開啟電源。



下載專案

1. 修改 Win-GRAF 通訊 IP 為目前的 PAC IP 位址。
 2. 填入連接 GSM 模組的 PAC COM Port 編號 (例如: 1)。
 3. 點選 On Line 按鈕，將 "dmeo_sms" 專案下載到 Win-GRAF PAC。
- (若不熟悉操作，請參考 2.3.4 節)



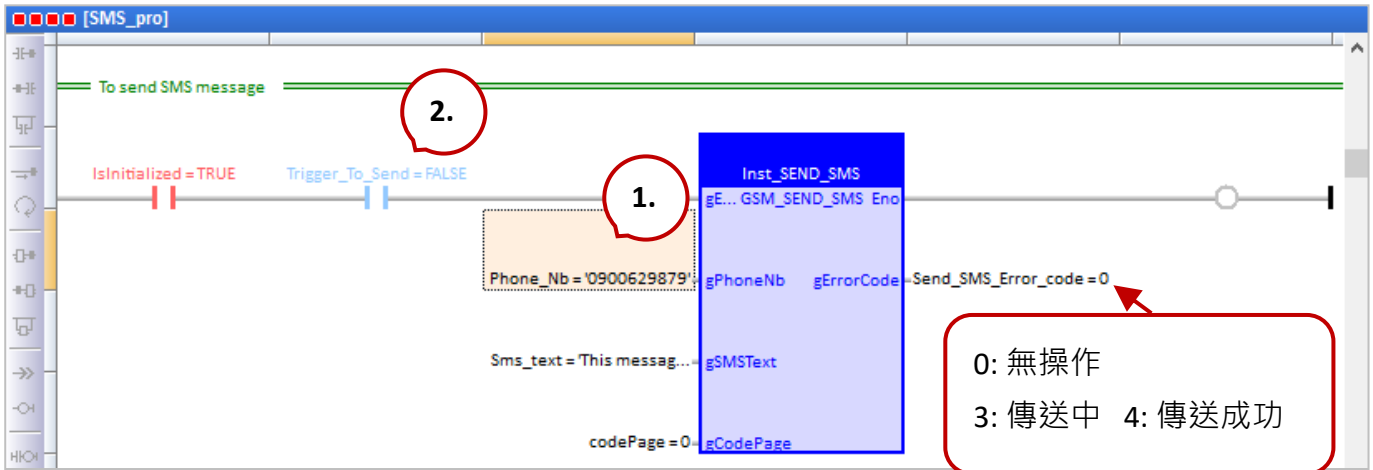
變數名稱	說明
GSM_Module_en	用來開啟 COM1 去連接並初始化 GSM 模組。 (初始值: TRUE)
Signal	GSM 模組的訊號品質 = 23。
Error_Code	GSM 模組的錯誤碼 = 0 (無錯誤)。
IsInitialized	已開啟 COM1 且 GSM 模組初始化 OK。 註: 若初始化失敗，您可將 PAC 重新啟動一次。

傳送簡訊

1. 滑鼠雙擊參數，來填入手機號碼、簡訊內容與編碼代號 (CodePage)。例如：

變數名稱	說明
Phone_Nb	'0900629879' 或 '886900629879' (初始值)
Sms_text	'This message is sent from Win-GRAF PAC' (初始值)
codePage	0: 傳送英文簡訊

2. 滑鼠雙擊 "Trigger_To_Send" 將其設為 "TRUE" 以傳送簡訊。

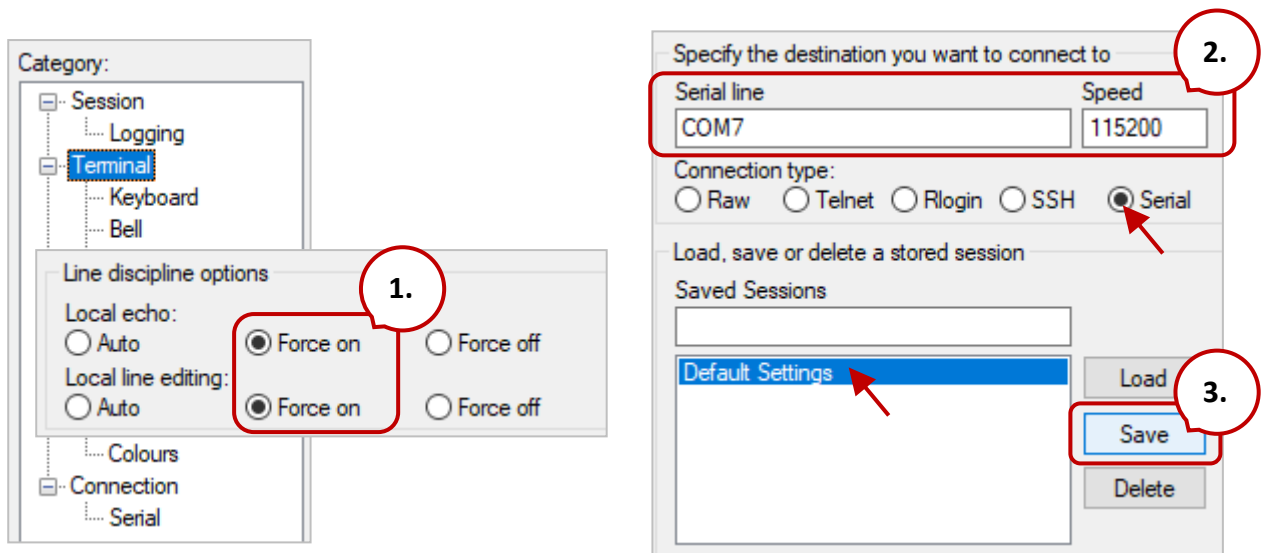


接收簡訊

GTM-204M 系列預設透過 USB Port 進行簡訊通知，可使用 AT+QURCCFG? 來詢問目前的設定狀態。

- 使用 USB 接收簡訊通知: AT+QURCCFG="urcport","usbat"
- 使用 UART 接收簡訊通知: AT+QURCCFG="urcport","uart1"

可使用 PuTTY 進行測試 (參考 [第 16 章](#))，請將 Terminal 內的 Local echo 與 Local line editing 設為 Force on，並在 Session 填入 Port 設定，再點選 Save 按鈕。



1. 輸入 AT+QURCCFG? 詢問目前設定狀態。
2. 輸入 AT+QURCCFG="urcport","uart1" 變更為使用 UART 接收簡訊通知。
3. 輸入 AT+QURCCFG? 確認目前設定。

```

COM7 - PuTTY
AT+QURCCFG?
AT+QURCCFG?
+QURCCFG: "urcport","usbat"

OK

ATE 0
ATE 0
OK

AT+QURCCFG="urcport","uart1"

OK

AT+QURCCFG?

+QURCCFG: "urcport","uart1"

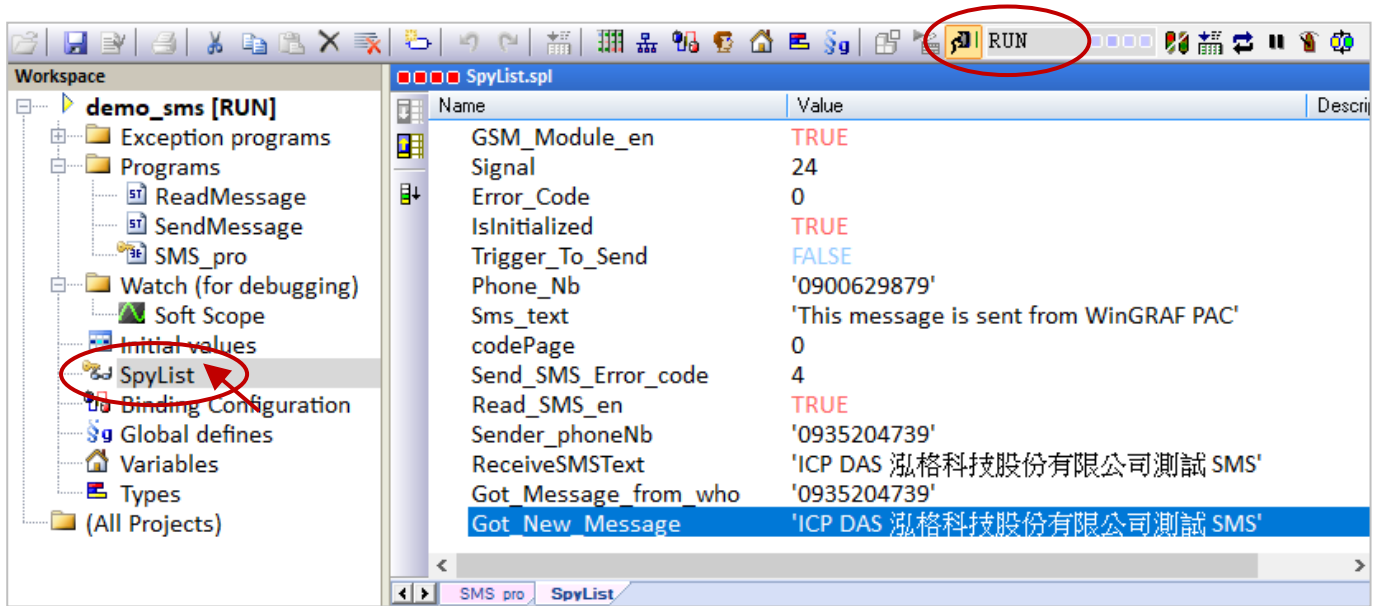
OK
  
```

請由您的手機傳送一封中文簡訊 (例如: ICP DAS 泓格科技股份有限公司測試 SMS) 至 SIM 卡的號碼。



變數名稱	說明
codePage1	950: 中文簡訊 ; 0: 英文簡訊
Sender_phoneNb	傳送者的電話號碼
ReceiveSMSText	例如: ICP DAS 泓格科技股份有限公司測試 SMS
Is_New_Message	TRUE: 有新簡訊

另外，可滑鼠雙擊 "SpyList" 開啟變數清單，來測試傳送/接收簡訊。

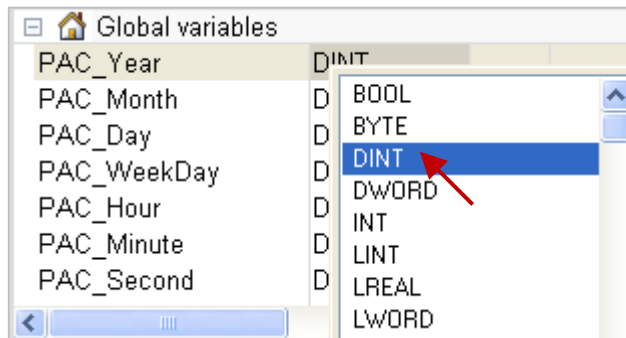


更多關於的 FAQ 請參考:

<https://www.icpdas.com/en/faq/index.php?page=2&model=GTM-204M-4GE#399>

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:

(*): 表示常用的資料型態

資料型態	位元數	數值範圍
BOOL (*)	Boolean	TRUE · FALSE
STRING (*)	Character string	最多 255 個字元
SINT	8-bit	Signed small integer
USINT		Unsigned small integer
BYTE		0 ~ +255
INT	16-bit	Signed integer
UINT		Unsigned integer
WORD		0 ~ +65535
DINT (*)	32-bit	Signed double integer
UDINT		Unsigned double integer
DWORD		0 ~ +4294967295
REAL (*)		Floating point
TIME (*)		Time
LINT	64-bit	Signed long integer
LREAL		Floating point
ULINT		Unsigned long integer
LWORD		0 ~ +(2 ⁶⁴ -1)

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

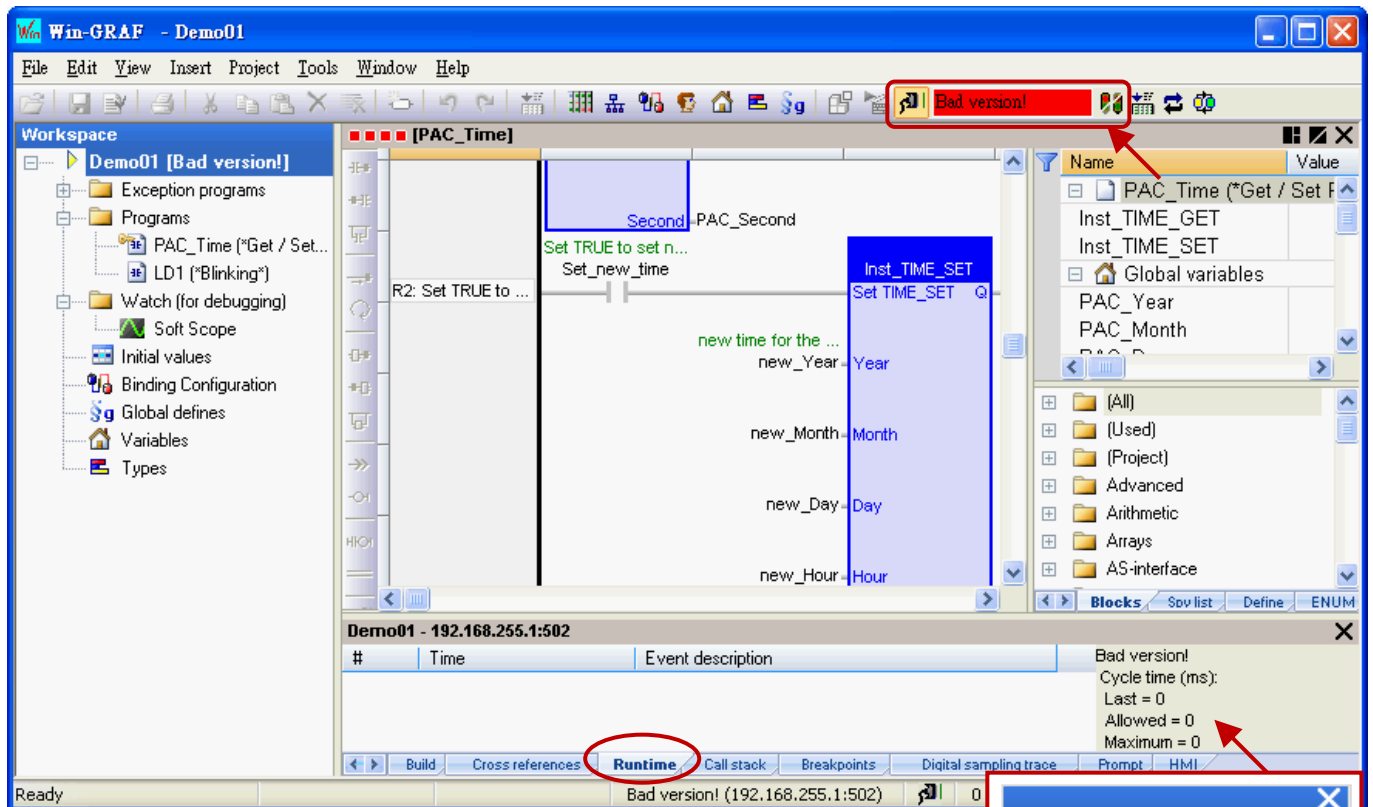
附錄 B 錯誤訊息排除

若 PC 與 PAC 之間的連線出現了錯誤訊息，請依照以下說明來排除問題。

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。

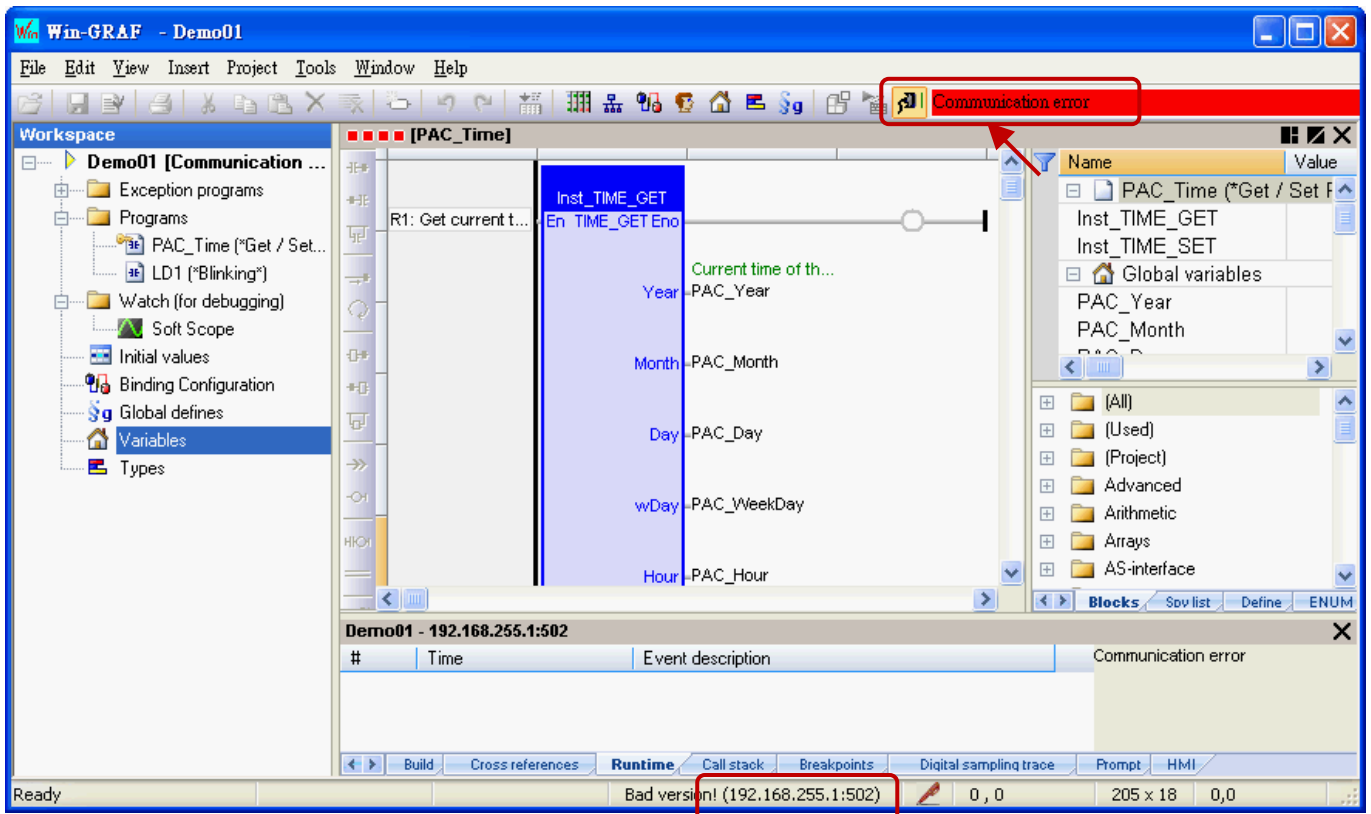


3. 出現 “RUN” 表示程式正常運行。



2. 若出現 “Communication error” 錯誤訊息:

表示 PC 與 PAC 之間通訊失敗。



解決方式:

1. 確認您的 Win-GRAF PAC 已經開機且網路通訊正常。
2. 確認 Win-GRAF 通訊 IP 已設為目前 PAC 的 IP 位址 (參考 [2.3.4 節](#)，此例為 “192.168.255.1:502”)。
3. 確認 PC 與 PAC 的 IP 位址在相同的網段上。

附錄 C 讓 Win-GRAF Workbench 透過 PAC 的序列埠連線

一般 Win-GRAF 可透過 Ethernet Port 與 PAC 連線，來下載專案或進行除錯。若需要啟用一個 PAC 的序列埠 (一般是 RS-232 或 RS-485) 讓 Win-GRAF 可連進來，其方法如下：

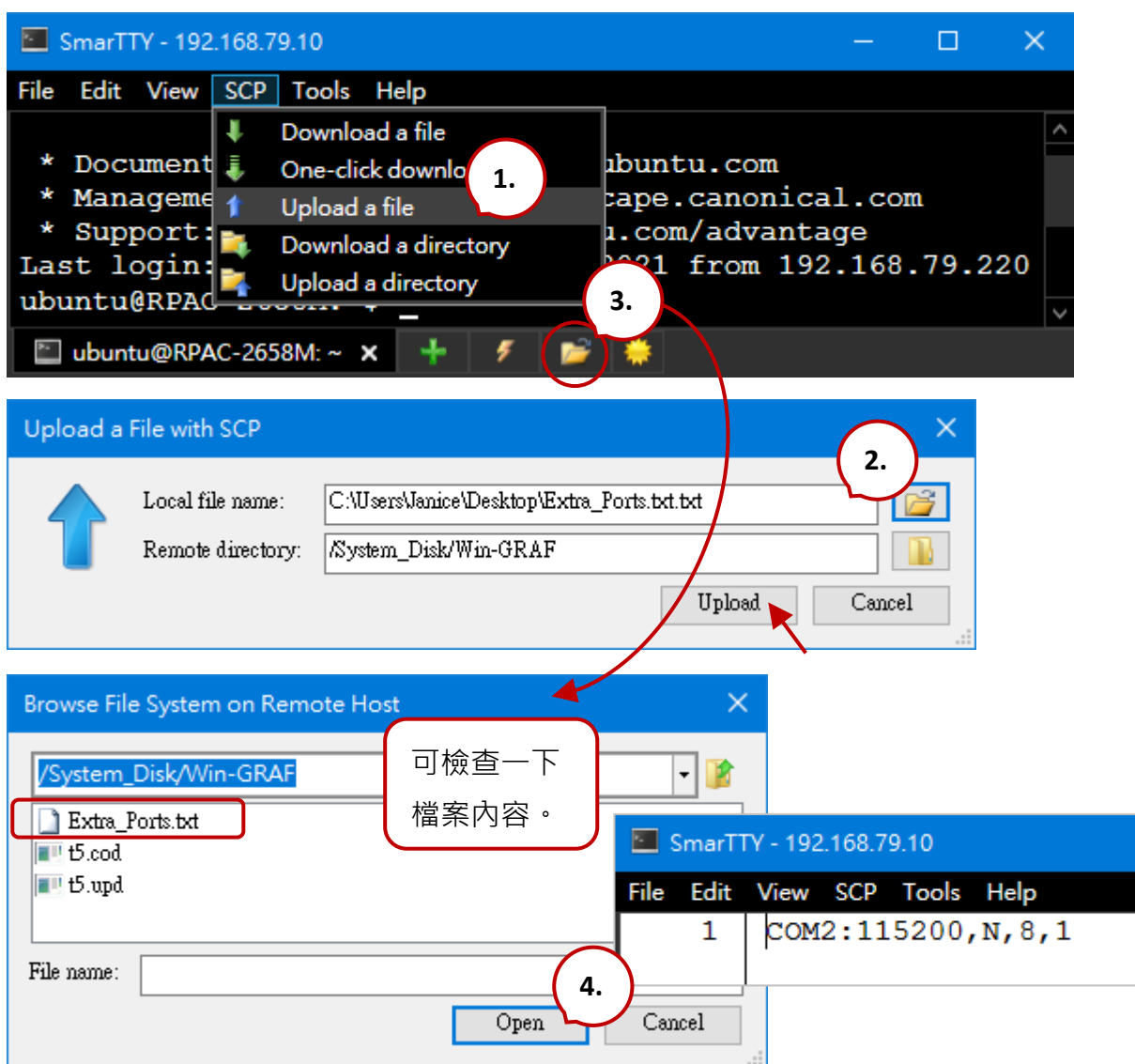
方法 1:

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。若想啟用 PAC 的 COM2 可輸入內容如下：

COM2:115200,N,8,1

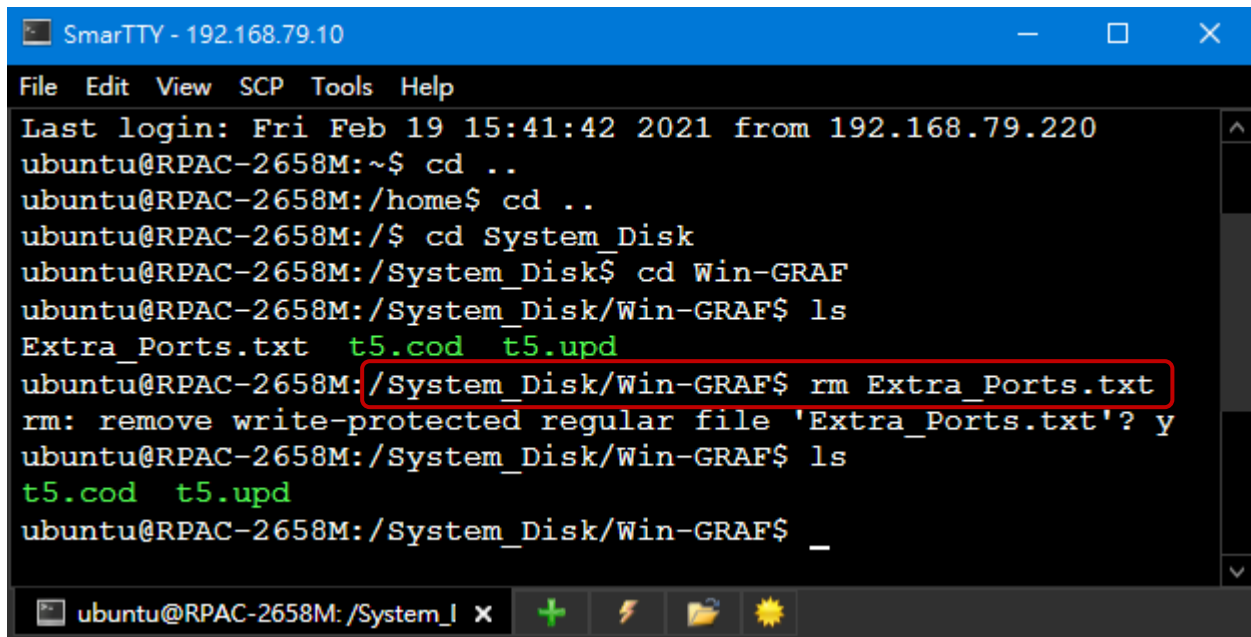
表示啟用 COM2 且
通訊速率 (Baud Rate) = 115200 bps

使用者可透過 SSH 連線 (參考 13.2 節) 將此 “Extra_Ports.txt” 檔案丟到 PAC 的 /System_Disk/Win-GRAF/ 路徑內，再將 PAC 重新開機一次即可。



取消啟用:

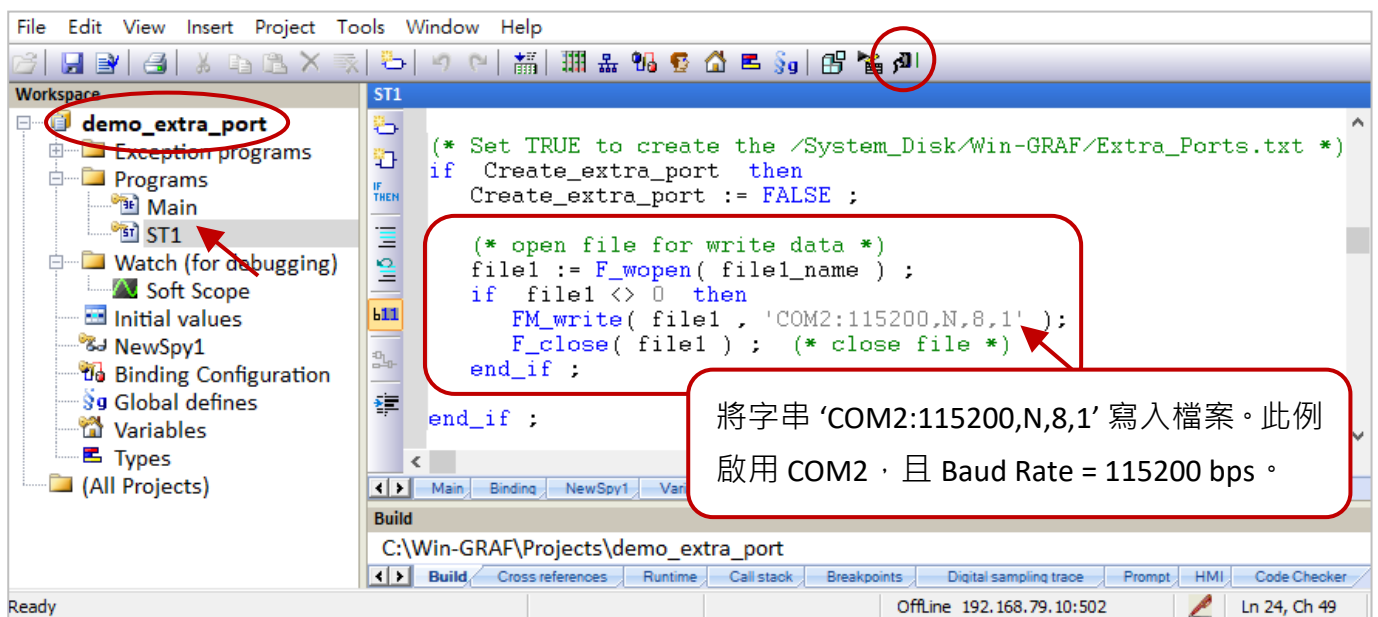
若想停用此 COM Port，可刪除 PAC 中 /System_Disk/Win-GRAF/Extra_Ports.txt 檔案，再將 PAC 重新開機即可。



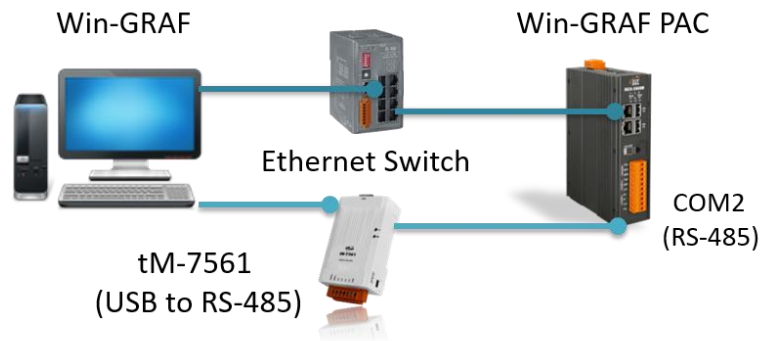
```
SmarTTY - 192.168.79.10
File Edit View SCP Tools Help
Last login: Fri Feb 19 15:41:42 2021 from 192.168.79.220
ubuntu@RPAC-2658M:~$ cd ..
ubuntu@RPAC-2658M:/home$ cd ..
ubuntu@RPAC-2658M:/$ cd System_Disk
ubuntu@RPAC-2658M:/System_Disk$ cd Win-GRAF
ubuntu@RPAC-2658M:/System_Disk/Win-GRAF$ ls
Extra_Ports.txt  t5.cod  t5.upd
ubuntu@RPAC-2658M:/System_Disk/Win-GRAF$ rm Extra_Ports.txt
rm: remove write-protected regular file 'Extra_Ports.txt'? y
ubuntu@RPAC-2658M:/System_Disk/Win-GRAF$ ls
t5.cod  t5.upd
ubuntu@RPAC-2658M:/System_Disk/Win-GRAF$ _
```

方法 2:

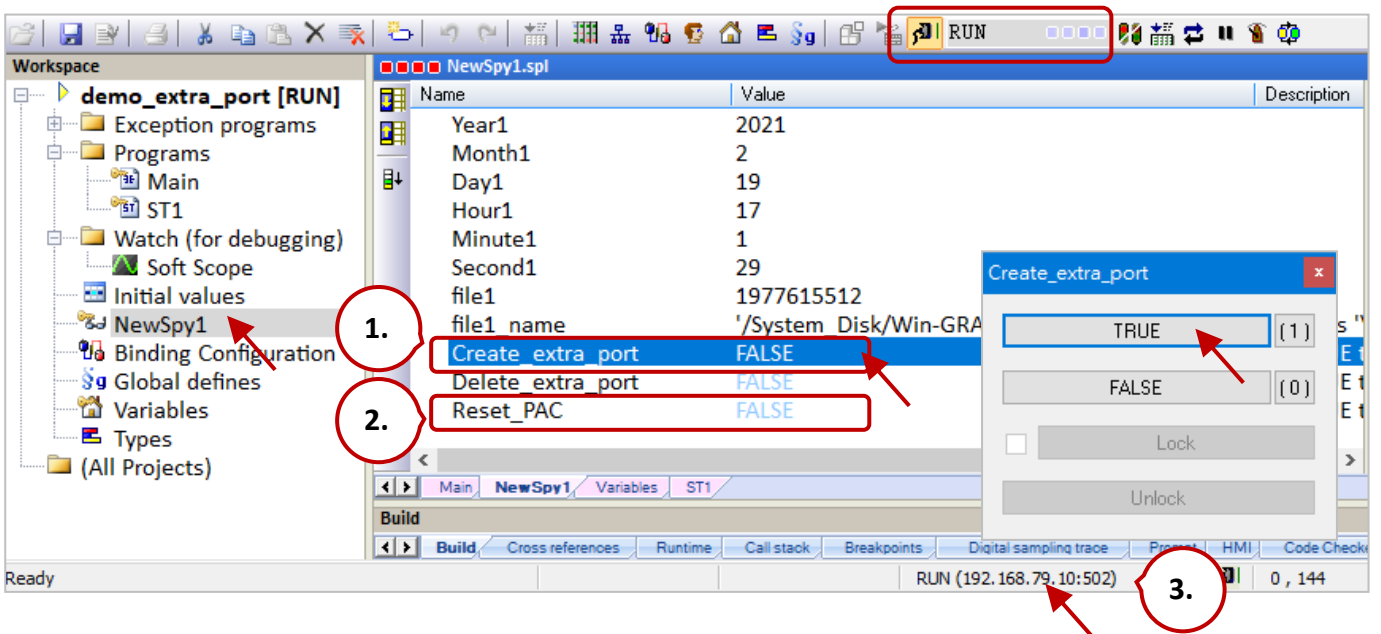
1. 請在網站下載 [範例程式 \(demo_extra_port.zip\)](#)，並在 Win-GRAF Workbench 點選功能表 "File - Add Existing Project - From Zip..." 來開啟專案。
2. 滑鼠雙擊開啟 ST1 程式，修改為 PAC 的 COM Port 編號與 baud rate (例如: 'COM2:115200,N,8,1')。
3. 點選 On Line 按鈕，先透過 Ethernet 將專案下載到 PAC 內。



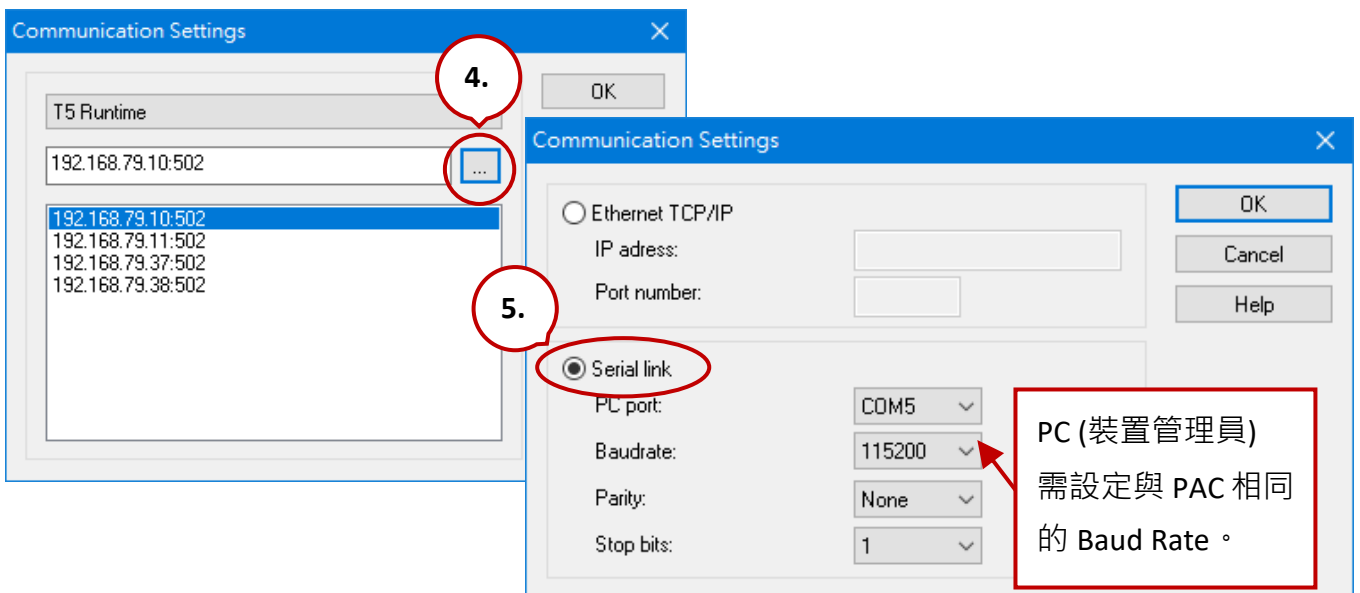
測試程式:



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/ 建立一個 "Extra_Ports.txt" 並寫入內容為 'COM2:115200,N,8,1'。
2. 設定 "Reset_PAC" 為 "TRUE"，PAC 會自動重開機並套用此設定。

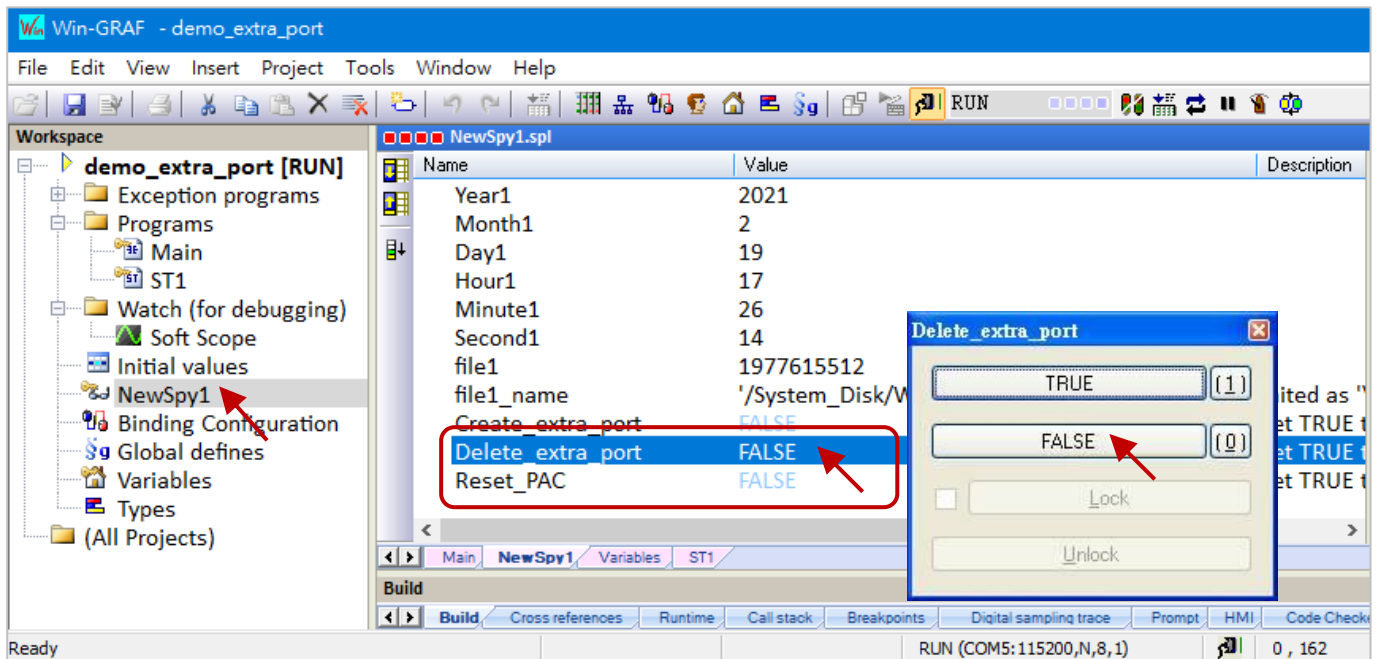


3. 取消與 PAC 連線，並修改 Win-GRAF 通訊設定。滑鼠雙擊狀態列的 IP 進入設定畫面，點選 ... 按鈕，先選擇 "Serial link" 並選擇 PC 上可使用的 COM Port。



取消啟用:

1. 將 "Delete_extra_port" 設定為 "TRUE"，會刪除 PAC 內 /System_Disk/Win-GRAF/Extra_Ports.txt
2. 將 "Reset_PAC" 設定為 "TRUE" 讓 PAC 自動重開機一次即可。



注意:

若使用方法 2 將 Extra_Ports.txt 下載到 PAC，就不可再使用方法 1 來覆蓋 PAC 內的檔案。

附錄 D PAC 上序列埠的接腳圖

RPAC-2658M:

COM1、COM2 與 COM3 的接腳圖。

Pin	Signal	Description
1	TxD	Console (RS-232)
2	RxD	
3	GND	
4	TxD	COM1 (RS-232)
5	RxD	
6	ISO.GND	
7	D+	COM2 (RS-485)
8	D-	
9	ISO.GND	
10	F.GND	Frame Ground

Pin	Signal	Description
1	Tx+	COM3 (RS-422, RJ-45)
2	Tx-	
3	Rx+	
4	-	
5	-	
6	Rx-	
7	-	
8	-	

