

Win-GRAF Workbench

軟體

使用手冊

Ver. 3.0



此使用手冊適用於 Win-GRAF Workbench 10.x 及以後版本軟體。

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 WinCE-based Win-GRAF 可程式自動化控制器 包括:

ViewPAC-x208:	VP-2208-CE7, VP-3208-CE7, VP-4208-CE7, VP-5208-CE7, VP-6208-CE7
ViewPAC-x238:	VP-1238-CE7, VP-4238-CE7, VP-6238-CE7
WinPAC-5000:	WP-5238-CE7
WinPAC-8000:	WP-8128-CE7, WP-8428-CE7, WP-8828-CE7 WP-8148, WP-8448, WP-8848
WinPAC-9000:	WP-9228-CE7, WP-9428-CE7, WP-9828-CE7
XPAC-8000	XP-8038-CE6, XP-8138-CE6, XP-8338-CE6, XP-8738-CE6 XP-8048-CE6, XP-8348-CE6, XP-8748-CE6

保固說明

泓格科技股份有限公司，所生產製造的產品自交貨給原購買者起，均享有一年的保固期限。此保固僅限於產品材料與製造上的瑕疵。

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、不當使用，而導致的第三方侵權行為承擔任何法律責任。此處包涵的技術或編輯錯誤、遺漏，概不負其法律責任。

商標識別

本手冊中所提及之所有商標，均屬於其合法註冊公司所有。

版權宣告

版權所有 © 2022 泓格科技股份有限公司保留所有權利。

技術支援

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

改版記錄

版本	日期	說明	編輯
3.0	06.23.2022	OPC UA Server	Lynn Tang
	01.03.2022	10.x Workbench 後的修訂版本	
	01.11.2023	修改 Retain Variables 變數名稱	Lynn Tang

目錄

目錄	4
1 軟體安裝 & 硬體設定	6
1.1 安裝 Win-GRAF 軟體.....	6
1.2 開啟 Win-GRAF 軟體.....	8
1.2.1 軟體介面說明	9
1.3 設定 Win-GRAF PAC 的 IP 位址	11
2 編寫一個簡單的 Win-GRAF 範例	12
2.1 建立 Win-GRAF 專案.....	12
2.1.1 建立單項任務專案	12
2.2 編寫程式	14
2.3 建立/ 編譯應用程式.....	16
2.4 下載應用	18
2.5 除錯	19
3 在 Workbench 中的 I/O 模組	20
3.1 加入 PAC I/O.....	20
3.2 加入遠端 I/O	23
3.2.1 Fieldbus Configuration.....	23
3.2.2 變數綁定	25
3.3 I/O 模組設定	26
3.3.1 I-8K Series	26
3.3.2 I-87K 系列	28
4 保存變數	31
4.1 保存變數	31
4.1.1 在 Workbench 加入保存變數	31
4.2 永久保存變數	32
4.2.1 RET_RetainVariable	32
4.2.2 RET_RetainArray	33
4.2.3 RET_RetainFormat	33
4.3 匯出& 匯入保存變數	33
4.4 保存記憶體空間	34
5 Modbus Master: 連接 Modbus Slave 設備.....	35
5.1 可啟用 Win-GRAF PAC 為 Modbus RTU/ASCII Master	35
5.1.1 設定通訊介面	35
6 Modbus Slave: 連接 Modbus Master 設備.....	44

6.1	Slave Data Block 設定.....	44
6.1.1	選擇 Slave.....	44
6.1.2	定義 Slave Register.....	45
6.1.3	定義 Holding Register	45
6.1.4	定義 Input Bit Data Block	48
6.2	Slave 型態設定.....	49
7	備援(冗餘)系統.....	51
7.1	硬體連接	53
7.2	軟體設定	54
7.2.1	XPAC Utility	54
7.2.2	Win-GRAF 備援設定 (LAN1/LAN2).....	56
7.3	切換機制	56
7.3.1	Main PAC 和 Backup PAC 切換.....	57
7.3.2	正常運作同步狀態	57
7.3.3	同步概念	57
7.4	功能方塊	58
7.4.1	GetActiveRTStat	58
7.4.2	GetActiveExStat	58
7.4.3	ActiveRTSwitch	59
7.4.4	ActiveSYNSwitch	59
8	OPC UA Server	60
8.1	OPC UA Server 設定	60
8.1.1	Server 設定程序	60
8.2	Server 設定參數.....	65
8.2.1	Server 資訊	65
8.2.2	對話設定	66
8.2.3	Server 訂閱設定	66
8.3	用戶認證設定	67
8.3.1	匿名身份識別	68
8.3.2	用戶名稱身分識別	69

1 軟體安裝 & 硬體設定

1.1 安裝 Win-GRAF 軟體

安裝 Win-GRAF 軟體之前，請先確認您電腦中的安裝環境。

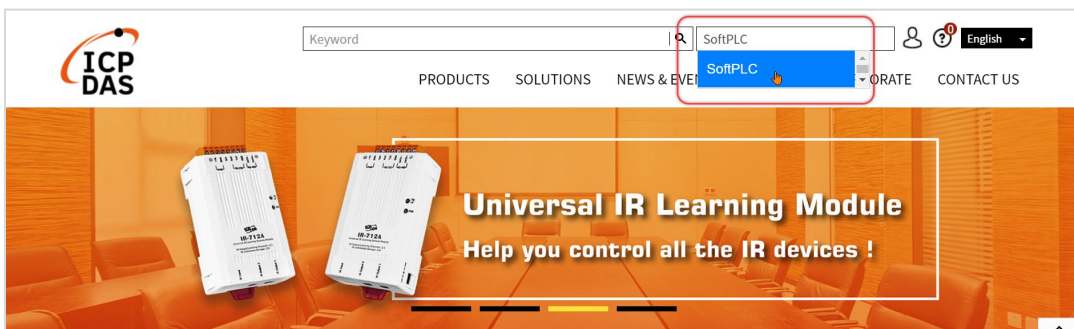
系統需求:

- 作業系統: Windows 7, Windows 8, Windows 10 (32-bits or 64-bits)
- Microsoft .Net Framework 3.5 (可在微軟官方網站下載:
<http://www.microsoft.com/en-US/download/details.aspx?id=22>)
- RAM: 至少 1 GB (建議: 2 GB)
- 可用硬碟空間: 至少 200 MB

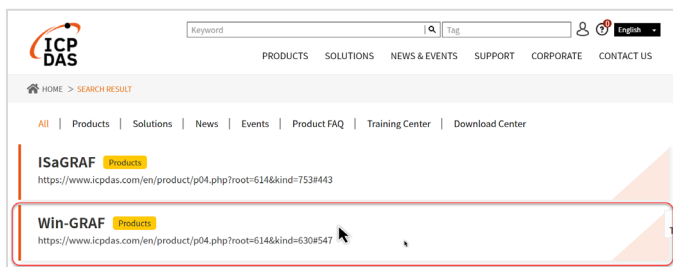
安裝步驟:

請至 ICP DAS 網頁下載 Win-GRAF Workbench 安裝檔案

(<https://www.icpdas.com>) 在 Tag 搜索框中輸入 'SoftPLC'，然後點選它。



點選 "Win-GRAF"，稍後打開新頁籤。



此圖顯示 Win-GRAF 標題。點選 "Download Center" 將會打開另一個頁籤。



滑鼠向下滾動尋找 Utility & Tools 的部分，點選 "Win-GRAF Workbench"。

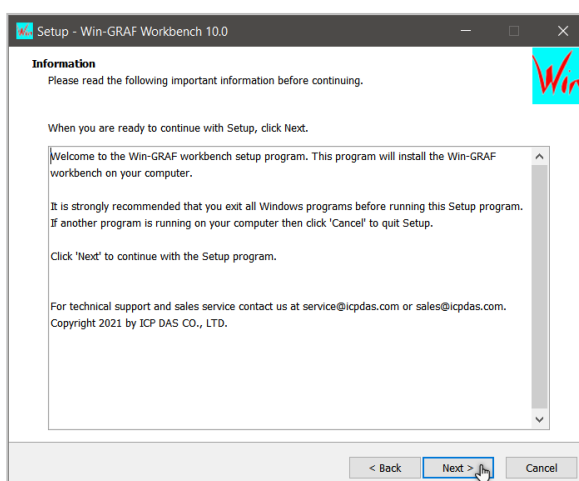
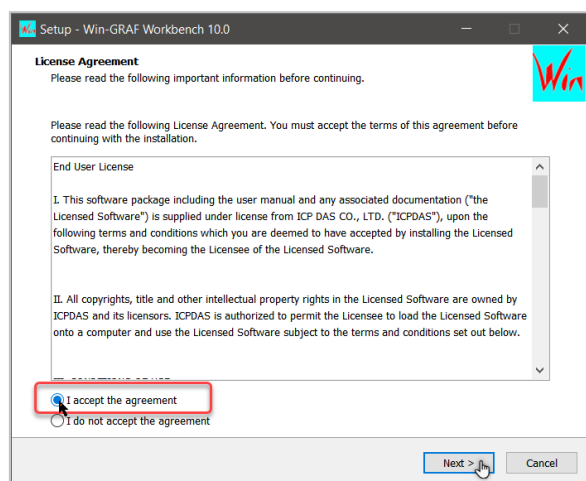
Utility & Tools				
	FILE NAME	DESCRIPTION	MODEL	LAST UPDATE
	Win-GRAF Tools	Win-GRAF Tools		2021-03-03
	Win-GRAF Workbench	Win-GRAF Development Tool		2020-11-25

點選 "Win-GRAF_Workbench_##.##_Setup.exe"圖標來下載執行檔。雙擊下載檔案並依照以下步驟安裝。

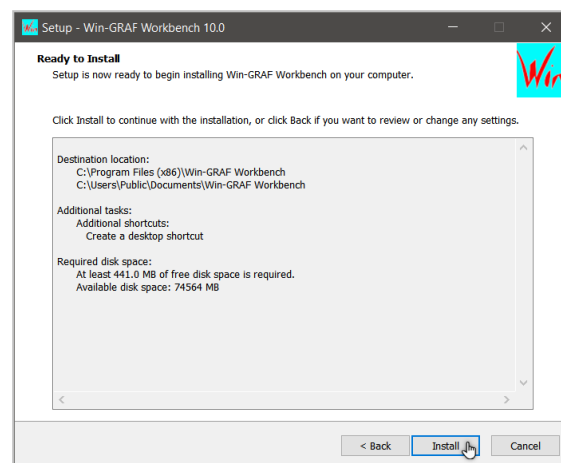
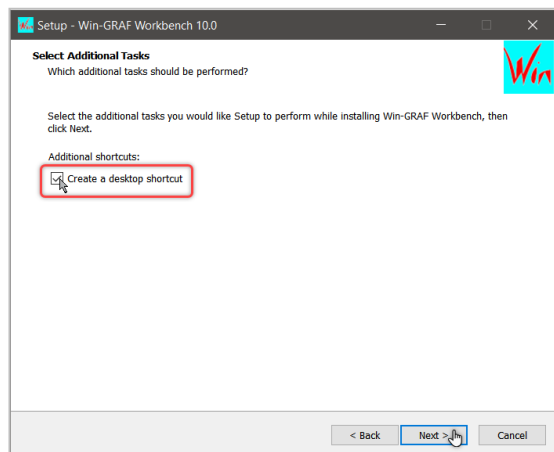
FILE NAME	VERSION	FILE DATE	SIZE	NOTE	
Win-GRAF-setup-ver-1.08.zip	1.08	2015-07-28	55.6 MB	Win-GRAF Workbench	

 Win-GRAF_Workbench_10.0_Setup.exe

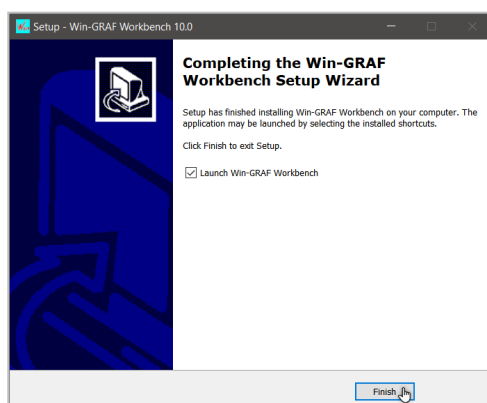
勾選 "I Accept the License Agreement" 並點選 "Next"，將會顯示資訊對話框，請閱讀並點選 "Next"。



可勾選 "Create a desktop shortcut" 建立桌面捷徑，再點選 "Next" 進行下一步。點選 "Install" 開始安裝 Win-GRAF 軟體。



已安裝完成並啟動 Workbench。



1.2 開啟 Win-GRAF 軟體

安裝過程成功完成後，在開始選單中點擊 "Win-GRAF Workbench"，workbench 準備開始。

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

Demo mode 限制:

1. 應用程式只能使用 40個 I/O點。
2. 編譯產生的 code 只能在 PAC 內運行 15 分鐘。
3. 模擬在 15 分鐘後停止。

開啟 Windows 的開始選單，點擊 "Win-GRAF" 資料夾和"Win-GRAF"來開啟此軟體。

圖 1: Windows 7

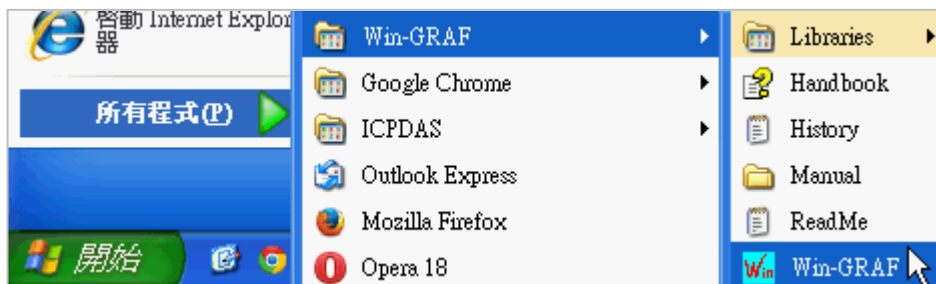
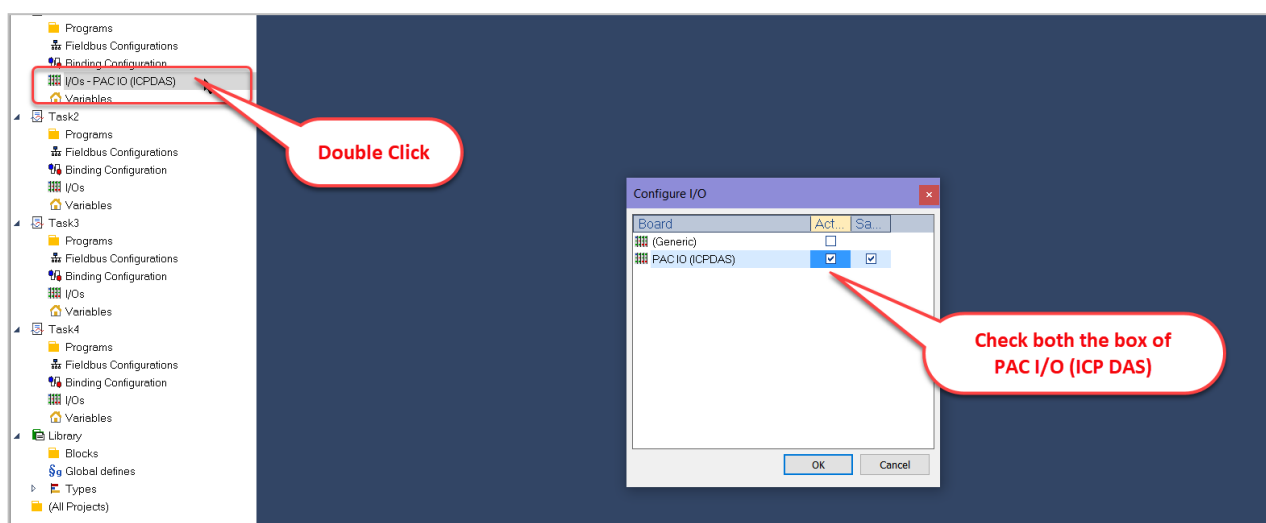
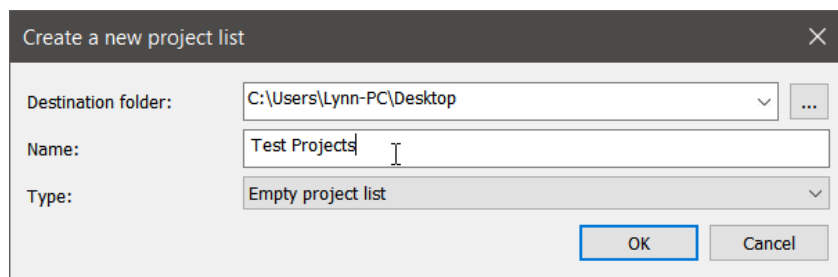


圖 2: Windows 8 & 10

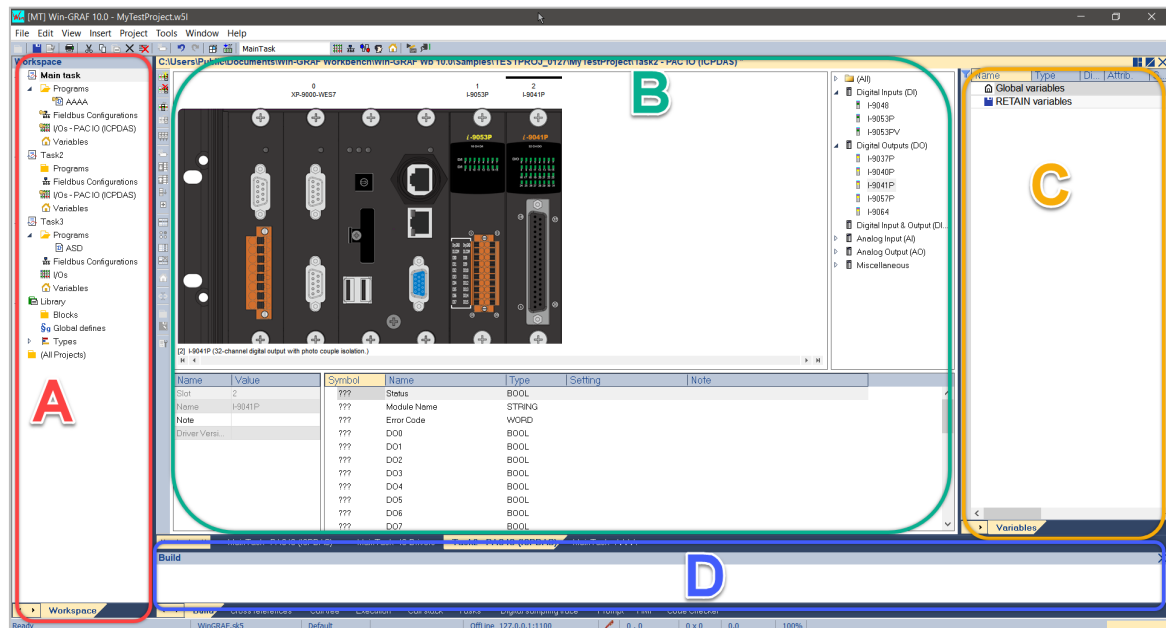


1.2.1 軟體介面說明

從上一節中找到執行路徑，運行 Win-GRAF Workbench 並按照以下步驟了解此開發工具。



Win-GRAF workbench 的新版本的用戶界面 (UI)顯示如下



A. 工作區: 任務名稱及其相關程式、fieldbus configuration、觀測清單和全域變數，顯示在Workbench 的左邊視窗工作區。工作區的项目清單內容，可透過雙擊項目顯示在主視窗。專案工作區是儲存格式“.W5L”的檔案，包含任務資料夾清單和一些設定資料。Workbench 為每個任務建立一個單獨的檔案夾，用於儲存程式原始碼、fieldbus configuration、IO 設定等。

提示: 可以將任務項目複製到工作區內的另一個任務。這可透過選單列表選擇複製命令、使用快捷鍵 CTRL+C 和 CTRL+V 或透過拖曳來完成。

B. 編輯區: 雙擊 "A. Workspace" 中列出的項目會顯示編輯區。用戶可開啟Main PAC (預設值為 XP-9000-WES7) 和相關的 PAC I/O，可透過加入和刪除產品圖來進行全部設定。習慣使用 IEC-61131 定義的 5 種 PLC 語言進行編輯的用戶，也可以在工作區加入並開始編輯程式。視窗底下顯示 PAC I/O 資訊，包括已經預加載到 Win-GRAF 工作台的程式庫資料。

C. 變數頁籤: 宣告已定義的全域變數和區域變數 (包括綁定或取消綁定) 顯示在 Workbench 主視窗的右上方區。變數編輯器是一個網格工具，可讓您宣告應用程式的所有變數。編輯區中的變數按照群組排序：

- 全域變數。
- "保留"變數。
- I/O 變數 (每個 I/O 設備為一個群組)。
- 程式變數區域 (包括自定義功能方塊 (UDFB) 輸出和輸入的參數)。

D. 輸出訊息: 如果連接設備並且執行，會顯示編譯訊息，透過執行會顯示所有運作中的訊息和每個任務的狀態。

備註: 想更改語系，請點擊工具欄位中的“Help”並選擇“語系”，Workbench 自動重新啟動後，將會轉換為您選擇的語系。

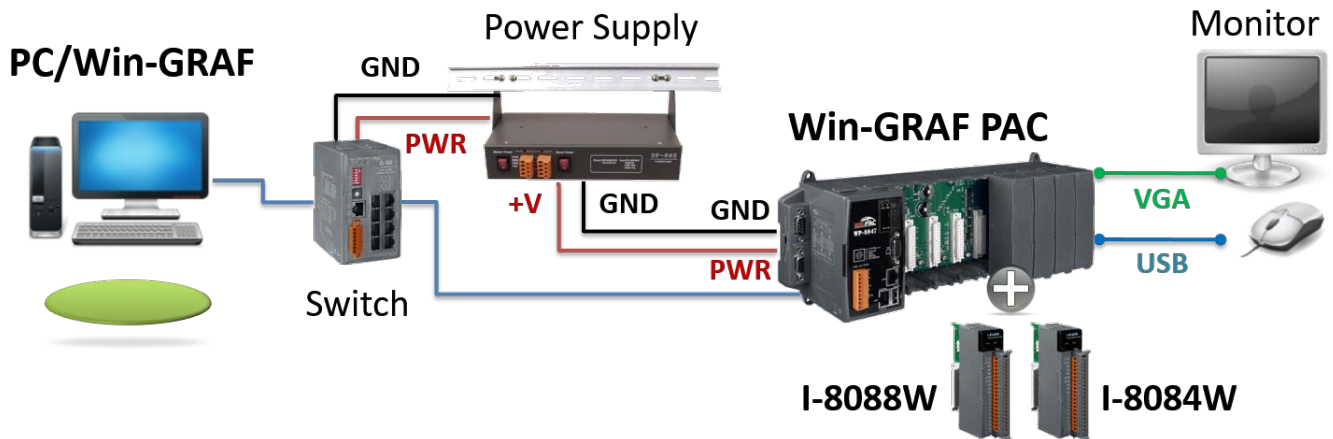
* 多國語系的支援是根據 OS 的語系來決定。例如:當 OS 是德語，Workbench 語言可在德語和英語之間互相切換。

1.3 設定 Win-GRAF PAC 的 IP 位址

僅適用於 Runtime 版本

下圖顯示如何完成接線和設定 PAC IP。以 WinPAC (WP-5xx8-CE7, WP-8x48, WP-8x28-CE7) 為例:

硬體連接圖



備註: PC/Win-GRAF IP 與 PAC IP 必須在同一個網段。

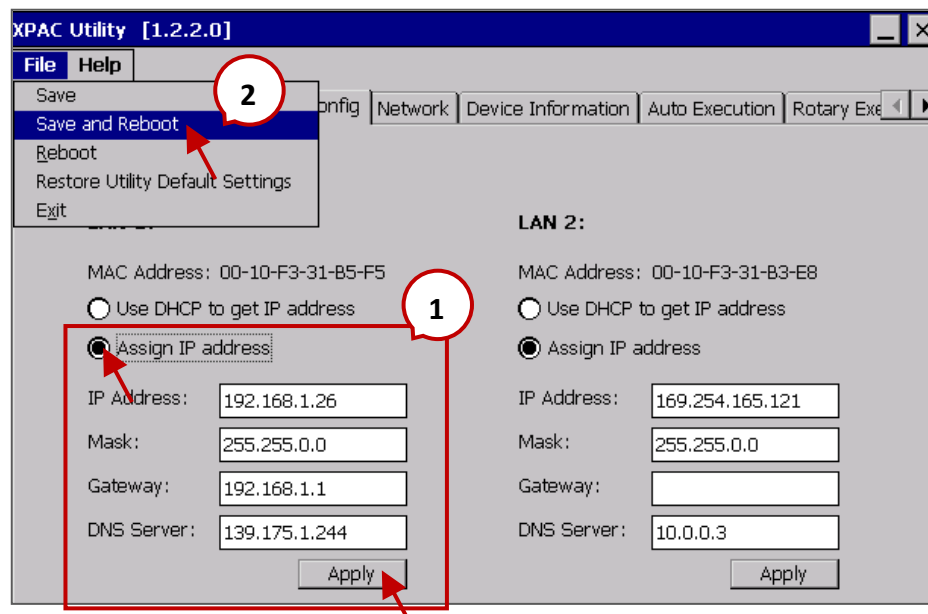
範例:

PC 的 IP: "192.168.1.20" (Mask: 255.255.255.0)

PAC 的 IP: "192.168.1.10" (Mask: 255.255.255.0)

PAC 端設定

下載並啟動 PAC_Utility，在"IP Config" 頁籤內設定 LAN1 (或 LAN2) 的 IP address 為 "Assign IP address"，並輸入適當的 IP 位址。接著，點選 "Apply" 按鈕，進入工具列，點選 File 並選擇 "Save and Reboot"，以套用設定。



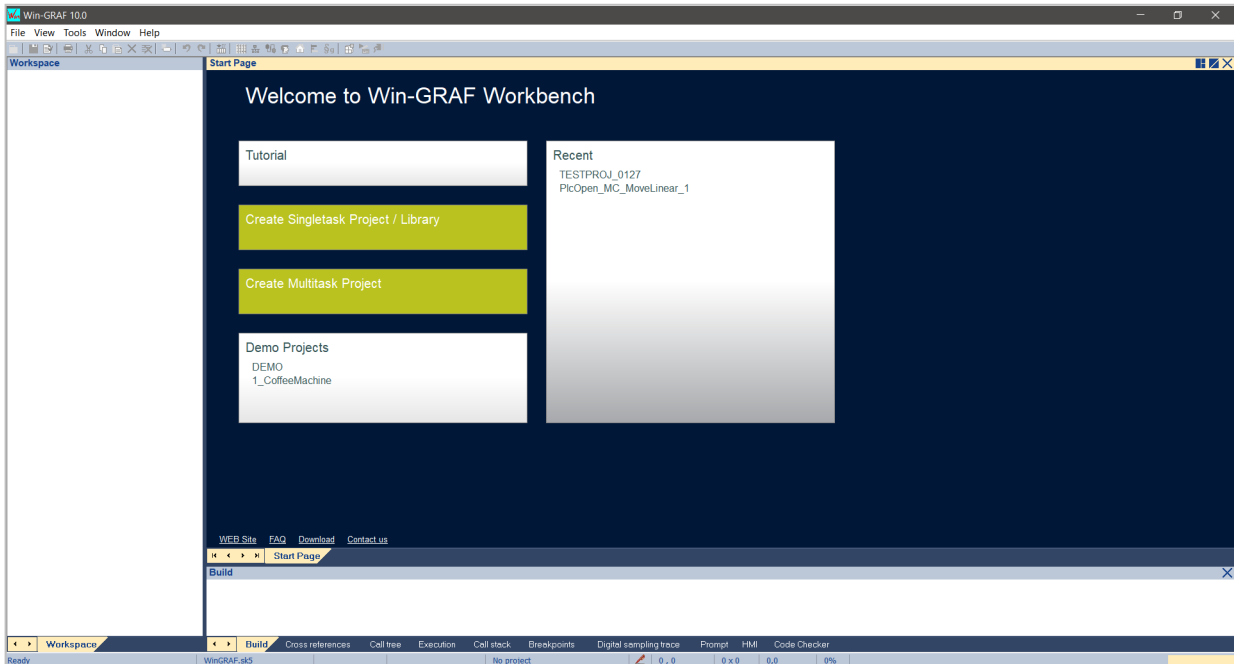
2 編寫一個簡單的 Win-GRAF 範例

2.1 建立 Win-GRAF 專案

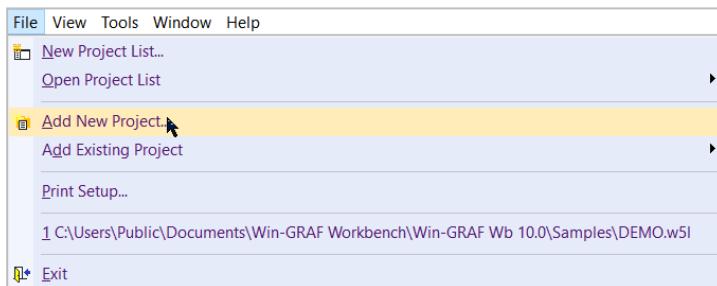
2.1.1 建立單項任務專案

使用 Win-GRAF workbench 建立單項任務 PLC 專案基本步驟:

1. 執行 Win-GRAF workbench，工作區是空白的，"Start Page"列出了可使用的手冊，專案範例 和 最近開啟的專案。此外，還列出 2 個綠色命令框，用於建立單項或多項任務專案。

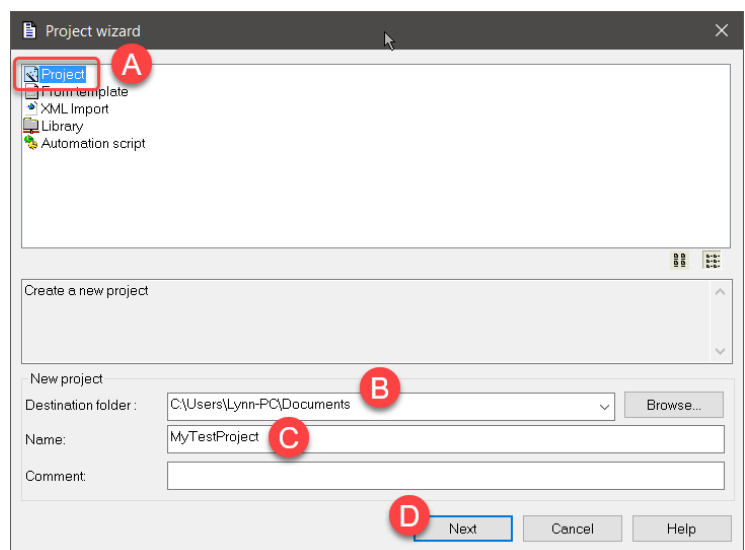


2. 要開啟單項任務專案的專案精靈，請點選 "Start Page" 的綠色命令框或到 "檔案\加入新專案"

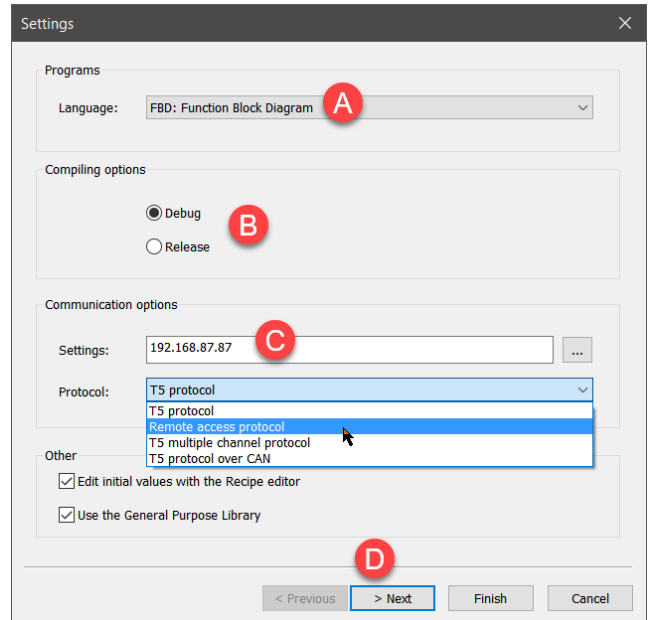


3. 點選"專案"，選擇目的地文件夾並輸入項目名稱，點選 "Next"。

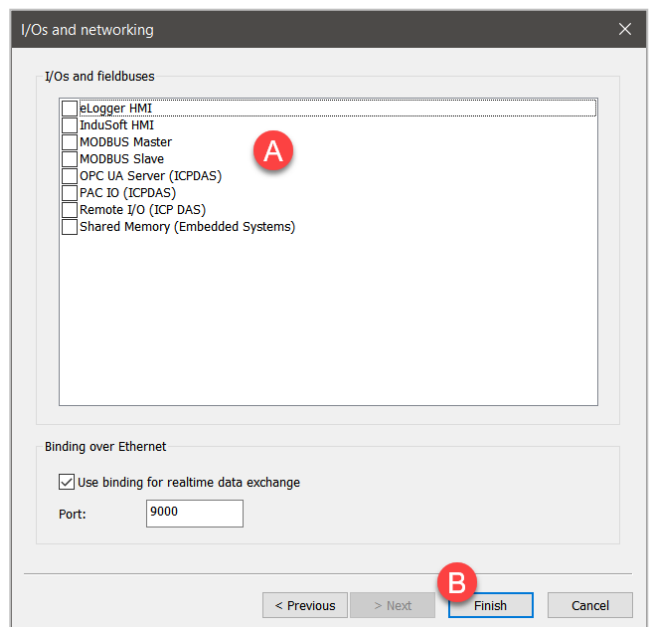
備註: 為了讓專案產生的一些檔案和文件夾更容易維護，因此，建議為專案建立一個新文件夾或使用一個空的目的地文件夾。



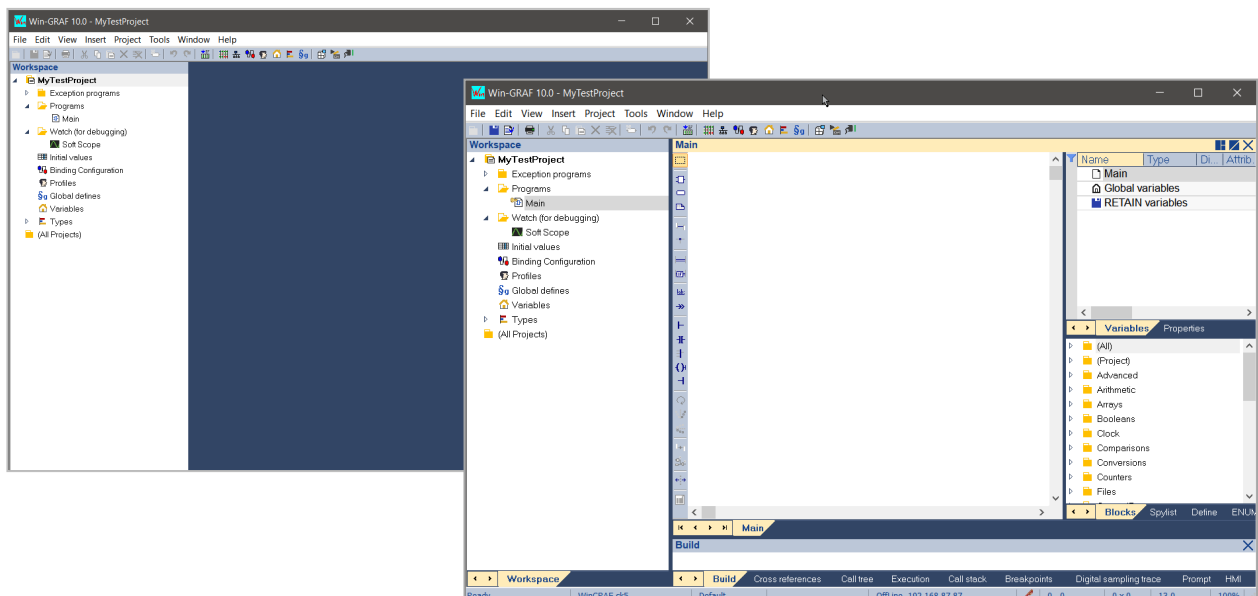
4. 設定的第一個程式(POU)編程語言，編譯選項，輸入遠端設備的 IP 位址並保留協議為 "邏輯服務"。專案建立完成後仍可修改所有設定，雖然精靈第一個程序只允許您選擇一種編程語言，但是之後可加入不同的編程語言的其他程式，點擊 "Next" 繼續設定。



5. 選擇要使用的附加組件，如果您不確定現階段專案要加入哪些組件，請勿勾選，這些選項可在專案開發中隨時手動加入，請勿更改 "Binding" 設定。點選 "Finish" 確認設定，將會產生具有目前設定的新專案。



6. 下圖顯示 "Workspace" 單個任務專案設定。雙擊 "Main" 項目來打開 main 程式編輯。

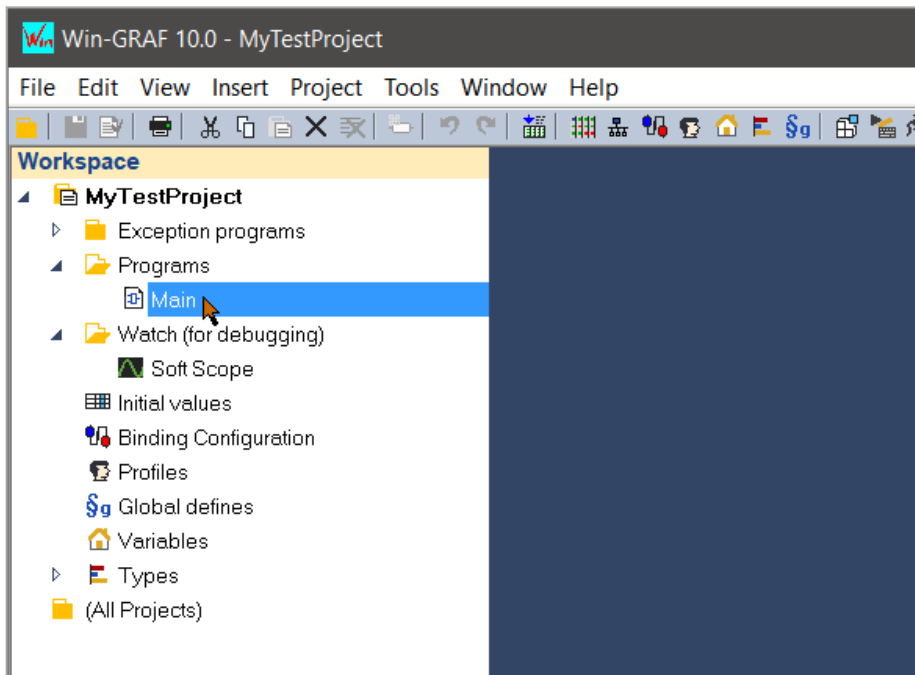


2.2 編寫程式

本節的重點是對於Workbench的用戶界面進行簡介，並顯示如何使用工作台提供的工具來編寫 PLC 程式的邏輯。Win-GRAF Workbench 支持 IEC-61131 定義的 5 種 PLC 編程語言，每一種編程語言都提供單獨編輯。

使用 FBD 程式編輯宣告變數 和 編寫 Function 的基本程序:

1. 雙擊工作區中的'Main' 程式名稱，開啟程式編輯:



備註: 雙擊程式名稱而不是圖標。

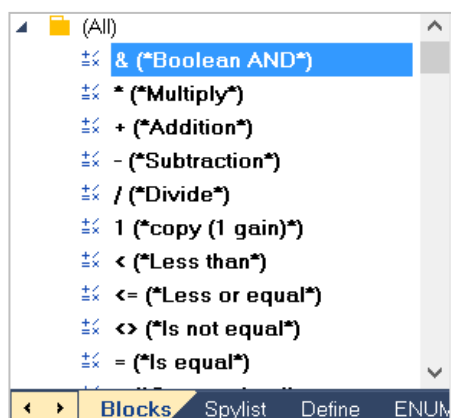
PLC 邏輯程式現在可使用 FBD 語言編寫，以下步驟示範如何加入功能方塊(function block)到編輯器，並宣告 input 和 output 變數。

2. 加入 "AND" 功能方塊:

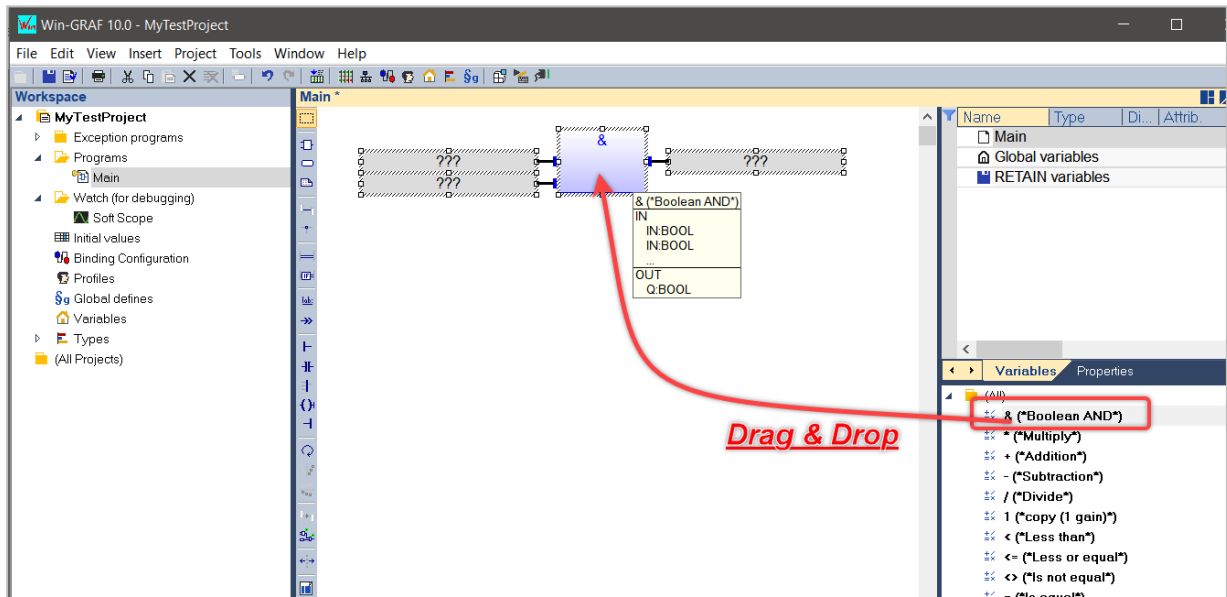
所有支援的功能方塊都列在訊息視窗右側的 "Blocks" 選項中，如果訊息視窗沒有，請至選單列中的"檢視/資訊頁籤 2"以顯示視窗。

功能方塊是根據不同的類別排序，"(All)" 類別清單支援所有功能方塊。

點擊"(All)" 樹節點，顯示所有功能方塊。



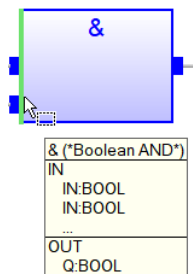
點擊 "& (*Boolean AND*)" 功能方塊並拖曳至編輯區。



"AND" 功能方塊有兩個 input 變數和一個 output 變數，在 inputs 和 output 出現"???" 表示變數尚未分配。

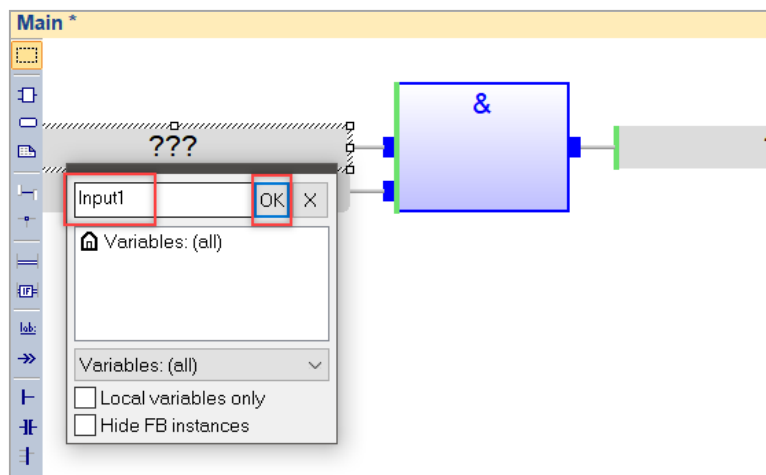
點擊功能方塊並在鍵盤上按 "+" 或 "-" 鍵，可更改功能方塊的大小。

將鼠標指針移動到函數上就會顯示所需的 input 和 output 資料型態：



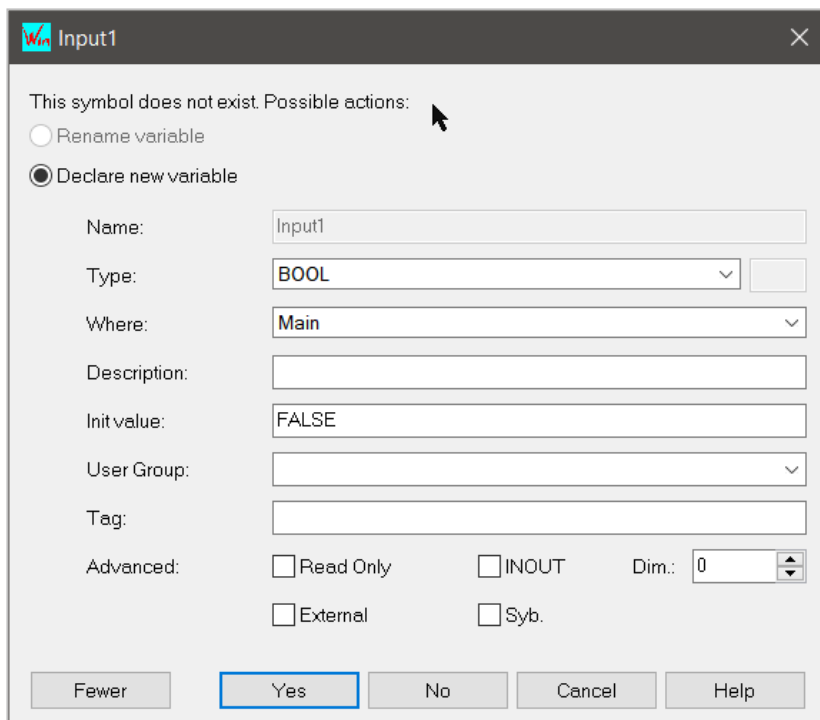
3. 分配變數到功能方塊:

- 雙擊有問號的灰色框 input
- 輸入變數名稱
- 點擊 "OK"

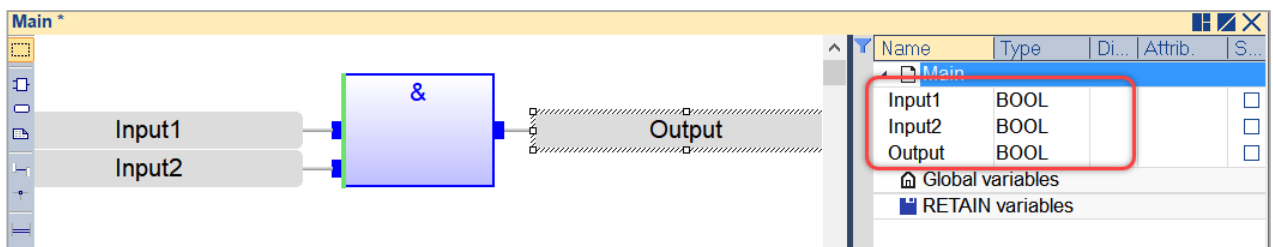


由於在跳出窗口之前專案尚未宣告變數，它允許您可選擇資料型態，初始值等。

- 功能方塊 input 資料型態是 boolean，所以選擇 BOOL。
- 初始值設定為 FALSE
- 點擊 "Yes" 把變數加到專案中



重複上述過程，加入 "Input2" 變數到第 2 個 input 功能方塊(function block)和 1 個 "Output" 變數到 output，全部新宣告的變數列在變數視窗右邊的螢幕。



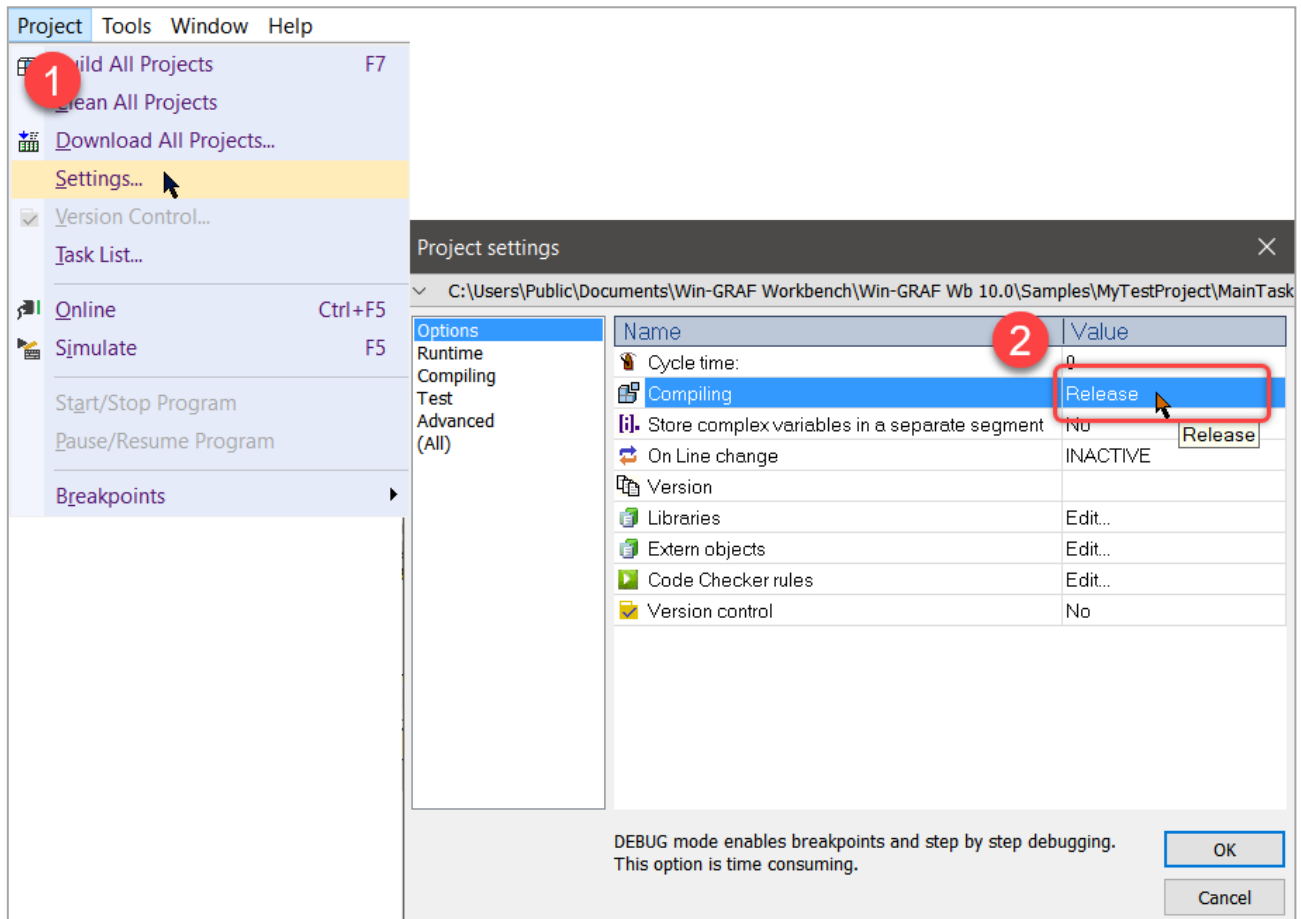
2.3 建立/ 編譯應用程式

workbench 支援 2 種編碼產生的類型: *Release* 與 *Debug* 模式。在 '*Debug*' 模式下編譯的應用程式支援週期執行、斷點和逐步除錯，斷點可以放置應用原始碼中的任何位置。除錯工具也顯示 UDFBs 和子程式在逐步執行時的呼叫堆疊。在 '*Debug*' 模式下編譯應用程式，包括步進附加的資訊，這都會導致 code size 過大效能過小。

**** 建議在 "Release" 模式下編譯您的應用程式。**

以下步驟顯示如何建立應用程式:

1. 從 Workbench 選單點選 „Project“，並選擇 „Settings“，會出現 „Project settings“ 對話視窗。
2. 在左邊選單中找到 „Options“，右邊會顯示一個 Workbench 功能列表，選擇 „Compiling“ 並雙擊 „Release“ 或 „Debug“ 切換模式，然後點選 „OK“。

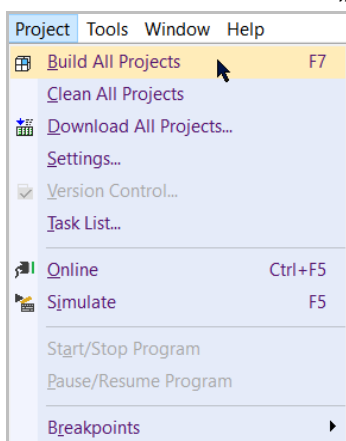


3. 要建立專案，請選擇以下方法之一：

a. 從工具列點擊 „Build all project“ 按鈕；

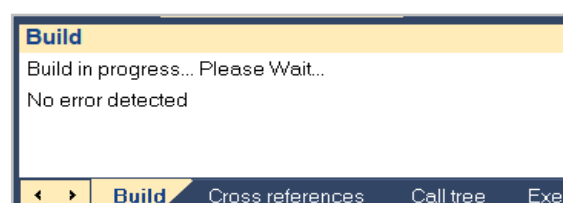


b. 從 Workbench 選單點選 „Project“，並選擇 „Build all projects“；



c. 按 „F7“。

然後您將會在輸出訊息視窗取得字串，表示過程是否成功。



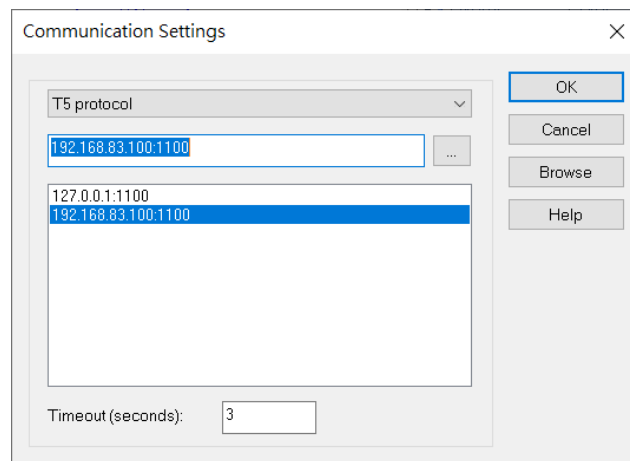
2.4 下載應用

應用程式已經編譯成功後，應用程序必須下載到 runtime 以便執行。Win-GRAF workbench 透過 TCP/IP 通訊與運行時交換資料。

為了建立 TCP/IP 通訊，Workbench 需要知道 target runtime 的 IP 位址和 socket port number。參考 target 設備的使用手冊，確定如何設定和取得通訊配置資料。

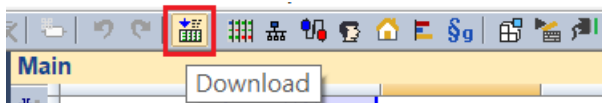
以下步驟顯示如何下載編譯應用程式：

1. 設定 workbench 和 runtime 之間的通訊參數。
 - a. Workbench: 設定 target runtime 的 IP 位址和 socket port number。
 - i. 從選單中選擇 „Tools/Communication Settings...” 或 雙擊視窗的底部狀態欄位 „offline” 部分。
 - ii. 編寫 target runtime 的 IP 位址和通訊埠號，兩者參數必須用冒號隔開。僅支援 Ethernet TCP/IP 通訊，點選 „OK”。

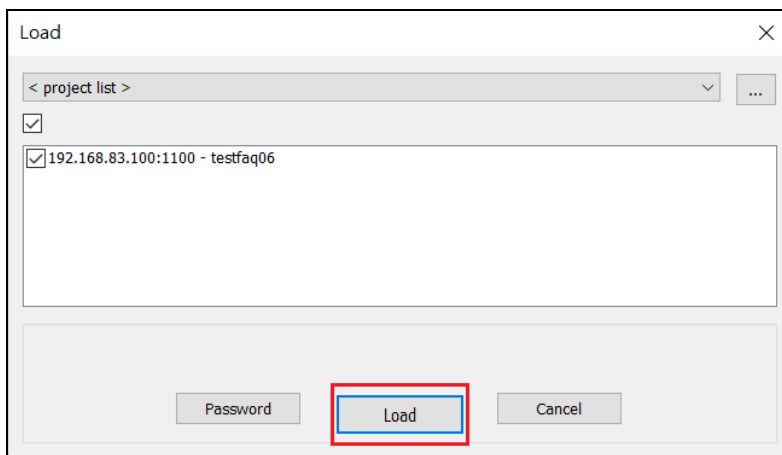


2. 下載程式至PAC

- a. 按下工具列中的Download圖示，如下圖所示。



- b. 確認好PAC 設定的 IP 及檔案後，按下按鈕Load進行下載程式至PAC上，等待上傳結束後完成下載，如下圖所示。



2.5 除錯

Workbench 允許用戶在 PLC 應用程式執行時，可直接更改變數值。Workbench 必須連接到運行時才能顯示目前的變數值。這個章節介紹如何透過 workbench 監控 PLC 程式和操控變數。

以下步驟介紹如何透過 workbench 直接修改 PLC 變數。假設 PLC 應用 已經在下載並執行。

步驟 1: 在 workbench 和運行時之間建立 TCP/IP 連接:

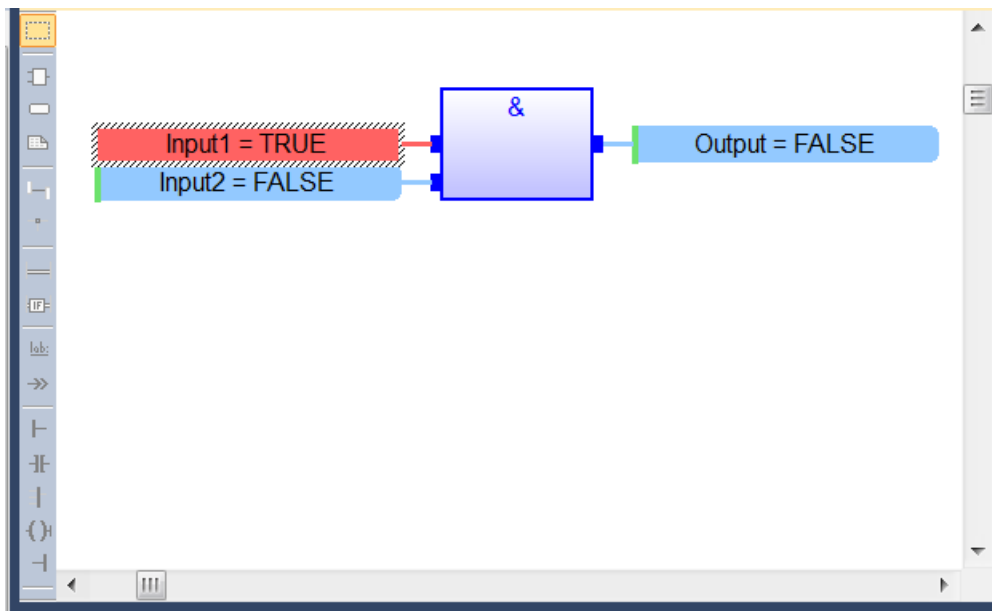
點擊 'Online' button:

建立連接後，每個變數的所有當前值都會顯示在變數名稱旁邊。如果運行空轉，這些變數值是在每個任務周期中更新。

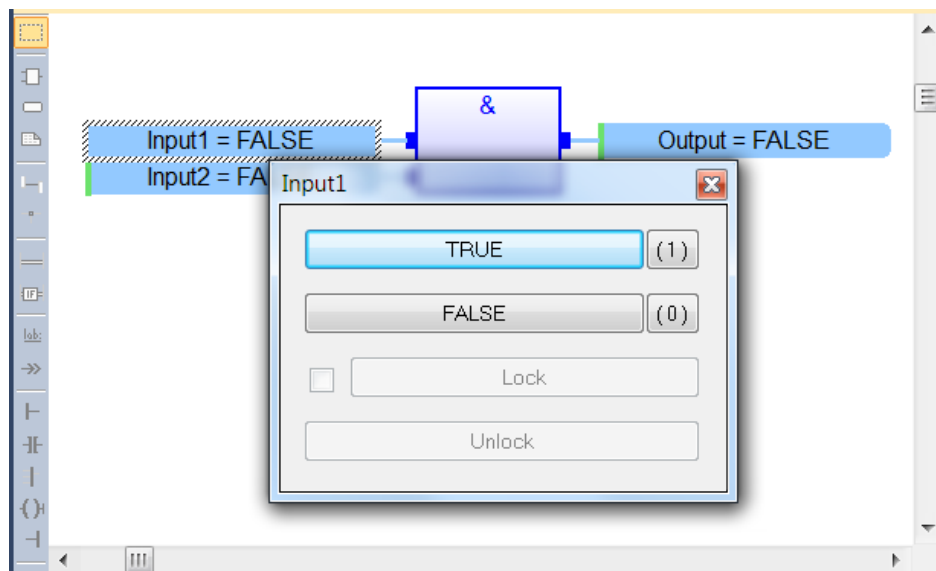
步驟 2: 變數值可以透過 workbench 直接更改:

雙擊 'Input1' 變數旁的功能方塊，並點擊跳出視窗的 TRUE 按鈕。

input 變數現在從 FALSE 變為 TRUE:



所有 PLC 資料型態都可以按照介紹方式操作。這允許直接測試 PLC 程式。



3 在 Workbench 中的 I/O 模組

泓格幾乎把所有 I/O 模組程式庫都放入此版本的 Win-GRAF Workbench 中，因此用戶現在可以輕鬆的雙擊並選擇主要 PAC 單元和放在 PAC 插槽上的模組。

請注意這個程式庫並非一直更新。如果您發現選擇的模組尚未列出，請聯繫您的業務窗口。

3.1 加入 PAC I/O

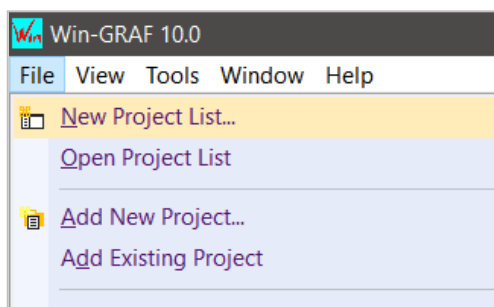
首先，查看下表中的 PAC 插槽編號和對應的 I/O 模組：

PAC 模組	插槽編號 (左至右)	支援的 PAC I/O 模組
WP-9x28-CE7	0 to 7	支援： I-9K 與 I-97K 系列 I/O 模組。
WP-8x28-CE7	0 to 7	支援： I-8K 與 I-87K 系列 I/O 高卡模組。 (例如: I-8017HW 與 I-87055W)
XP-8x38-CE6 (*)	0 to 7	
VP-x2x8-CE7 (*)	0 to 2	不支援： I-8K 與 I-87K 系列 I/O 低卡模組。 (例如: I-8017H 與 I-87055)
WP-5238-CE7	-	掌上型 PAC，可裝一片 XV board。 (例如: XV107，XV116，XV308 等。)

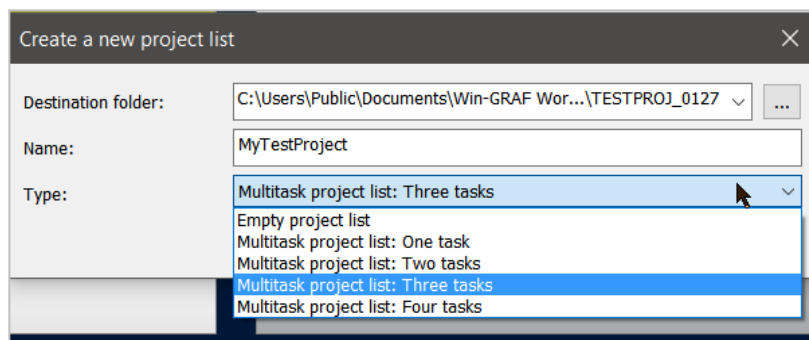
(*) The XP-8038-CE6, VP-x208, WP-5xx8-CE7 are the PAC without slot.

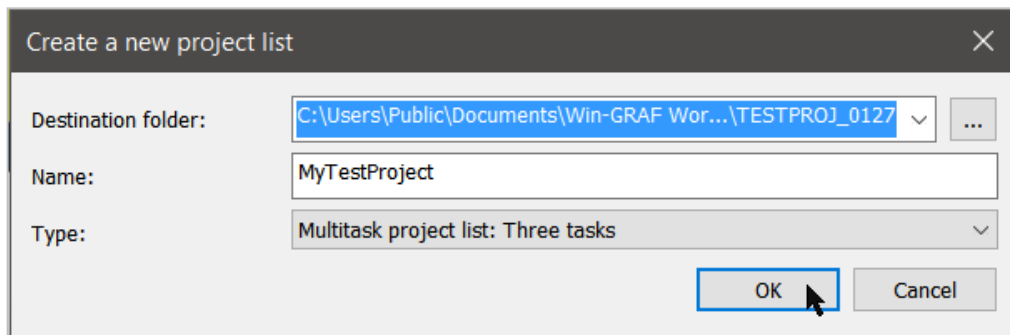
下方我們將介紹如何加入 PAC 以及相關的 PAC I/O 模組。

1. 點擊工具列，選擇並建立一個 "New Project List"。

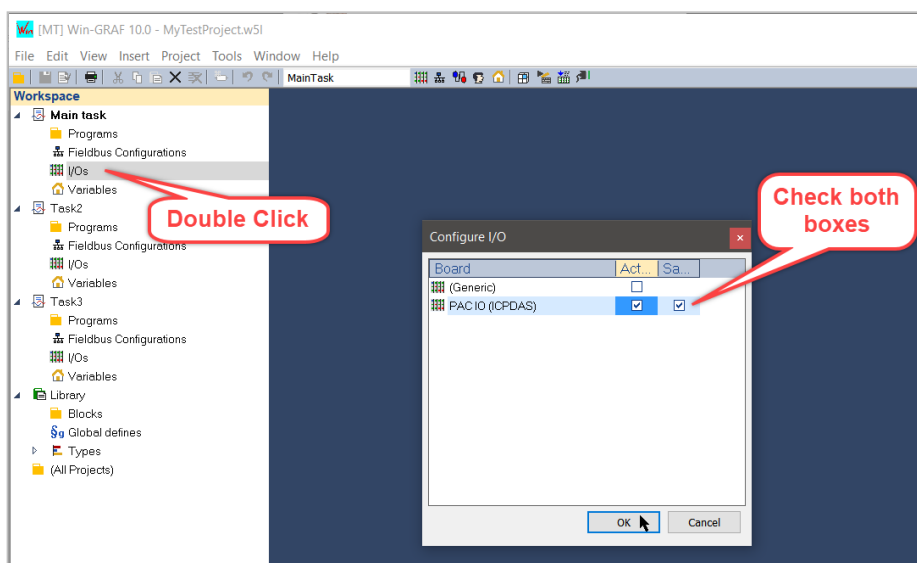


2. 輸入名稱並選擇您的專案清單類類型，將會自動選擇目的文件夾，我們建議使用預設值路徑，點擊 OK。

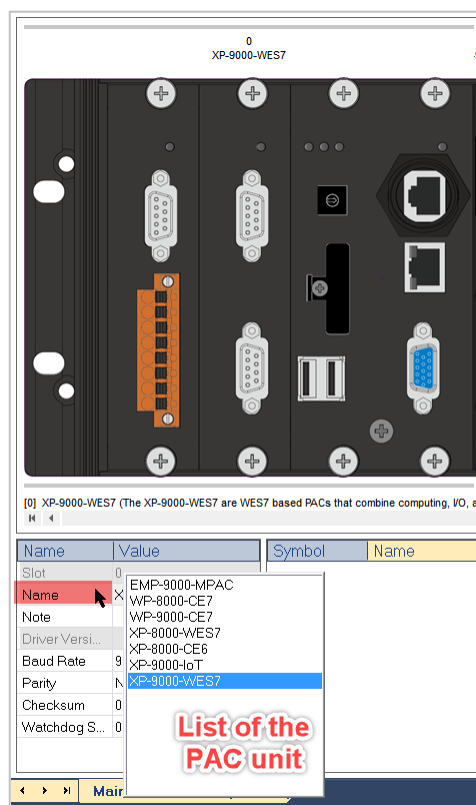




3. 雙擊 I/O 會出現一個小視窗，取消 Generic 複選框，並勾選 PAC IO (ICPDAS) 的 2 個複選框，然後點擊 OK。



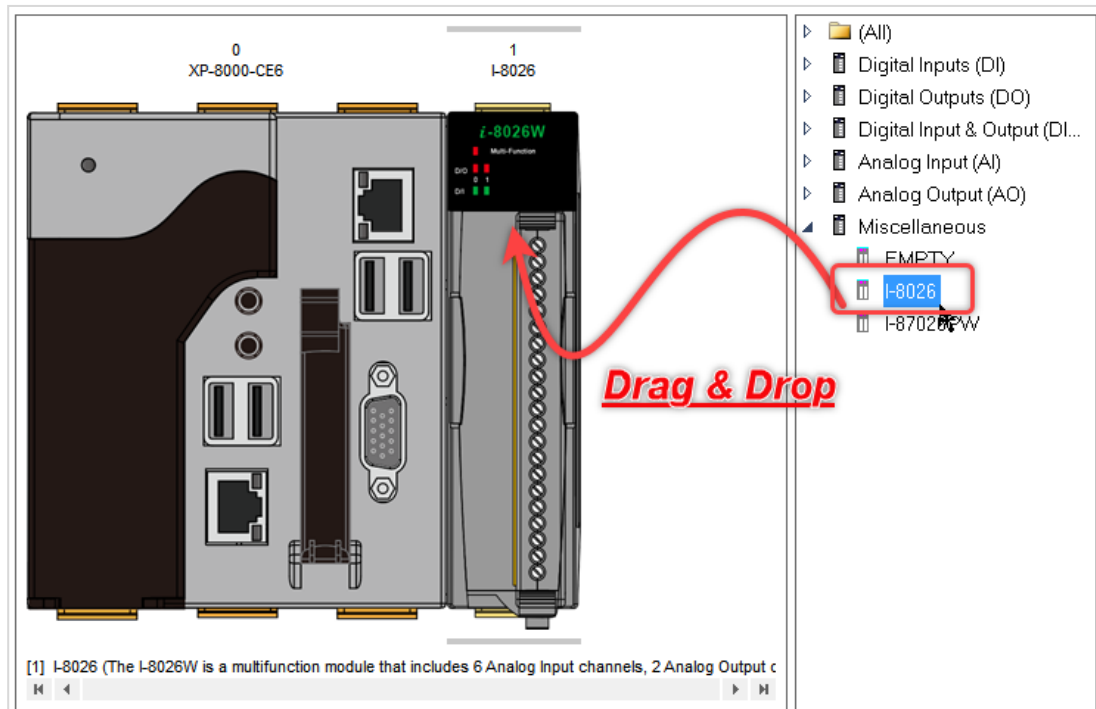
4. 點擊 OK 後將如下圖(預設值 PAC 是 XP-9000-WES7，尚未發行)。
更改在手邊的 PAC，雙擊產品圖下方的"名稱"欄，這會顯示 PAC unit 列表。



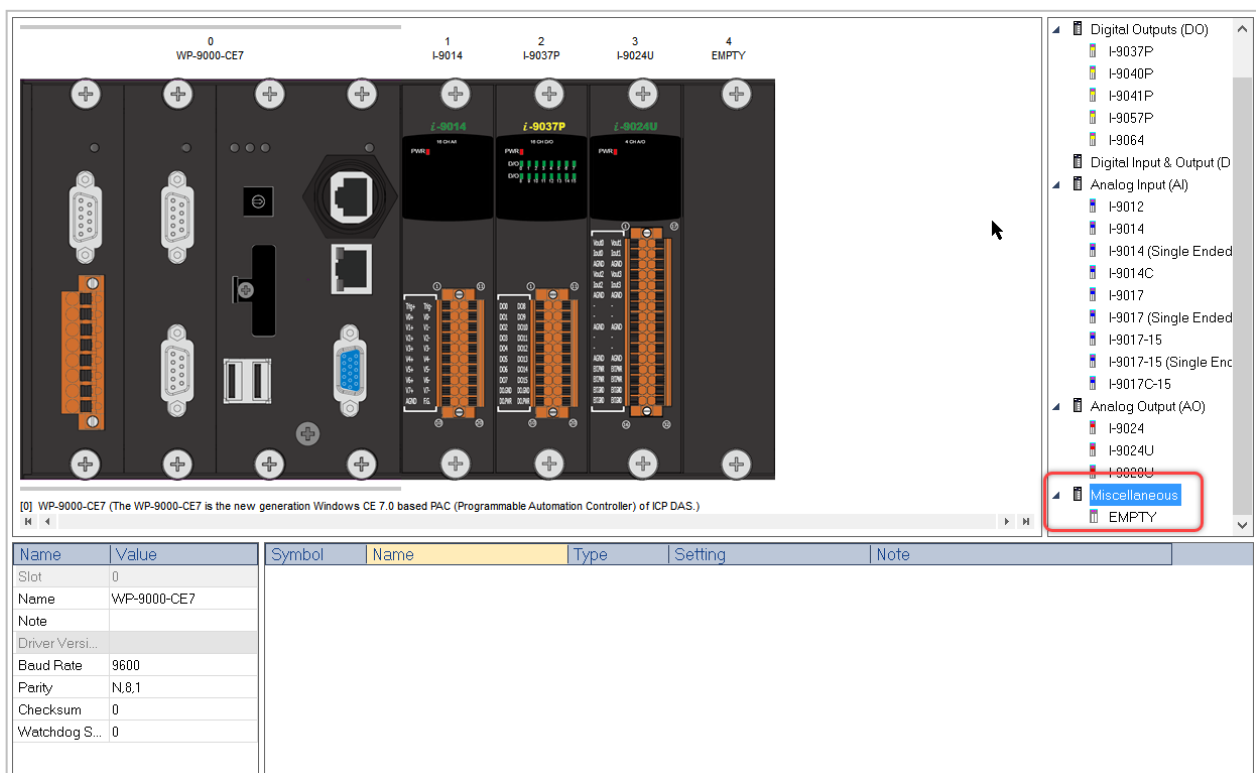
5. 加入 PAC I/O 模組到 PAC 擴充插槽

打開右上角視窗的模組列表，選擇模組型號並拖曳到左邊 main PAC 的空白區。

在下圖，加入多功能模組 I-8026 到第一個插槽。



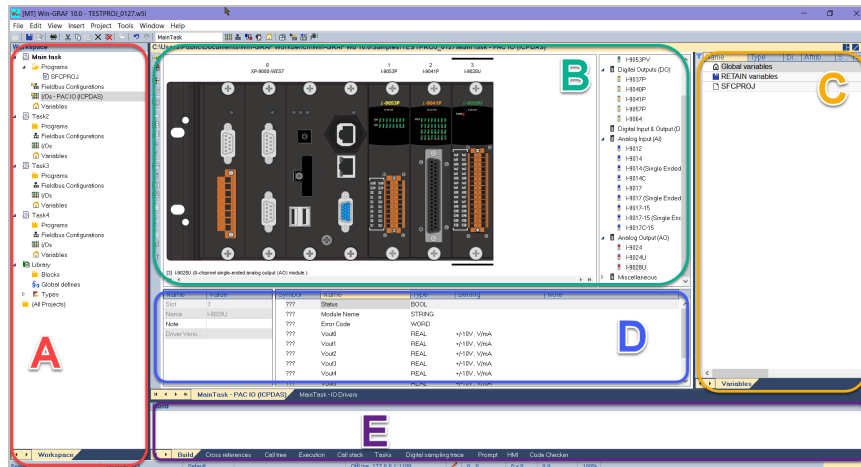
6. 下圖顯示 1 個 4 插槽 WP-9428-CE7 的 PAC 與加入的 3 個 I/O 模組。ICP DAS 為不同的 PAC 提供不同數量的 I/O 插槽，例如:1/4/8 或 1/3/7。請再次確認網站，確認您手頭上的 PAC I/O 最大插槽數。如果在 Win-GRAF Workbench 中，您需要的 I/O 模組少於確切的插槽數，請從 Miscellaneous 列表拖曳一個"EMPTY"模組。



3.2 加入遠端 I/O

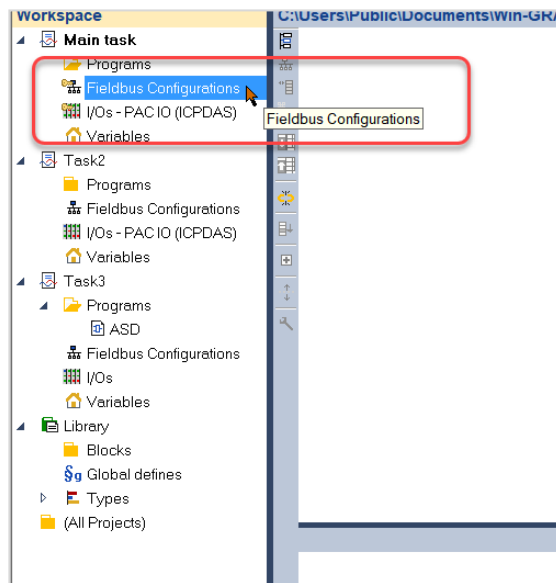
3.2.1 Fieldbus Configuration

加入 main PAC 和 I/O 模組後，我們可以取得一個小系統，如下圖。矩形 B+D 類似於前面第 1 章我們介紹的用戶界面，我們在這裡不做詳細介紹。

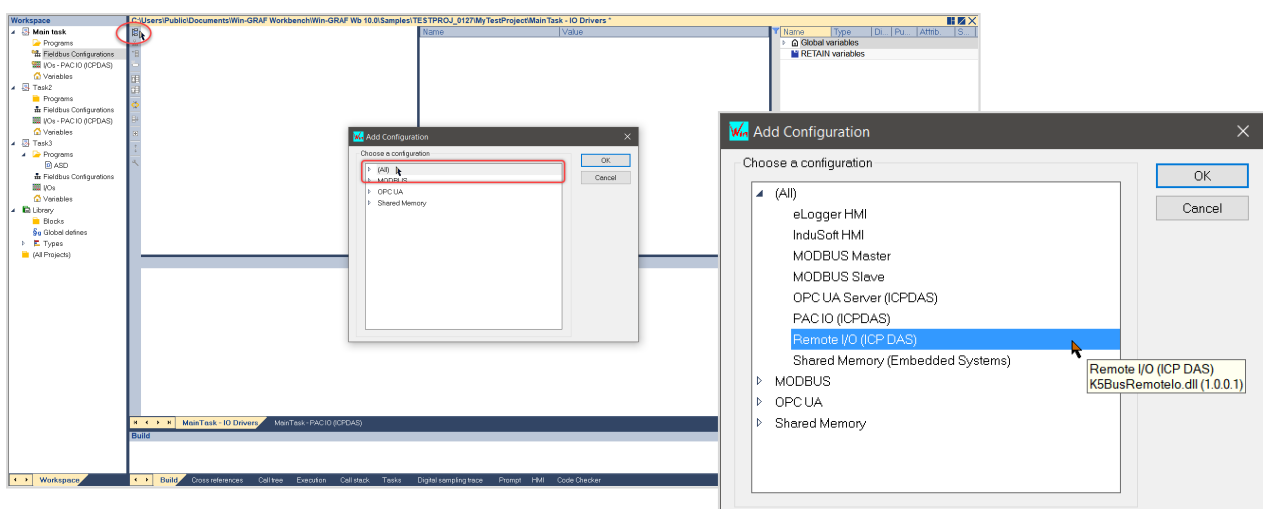


接下來，我們必須定義變數，並綁定到各個通道來使 I/O 運作。

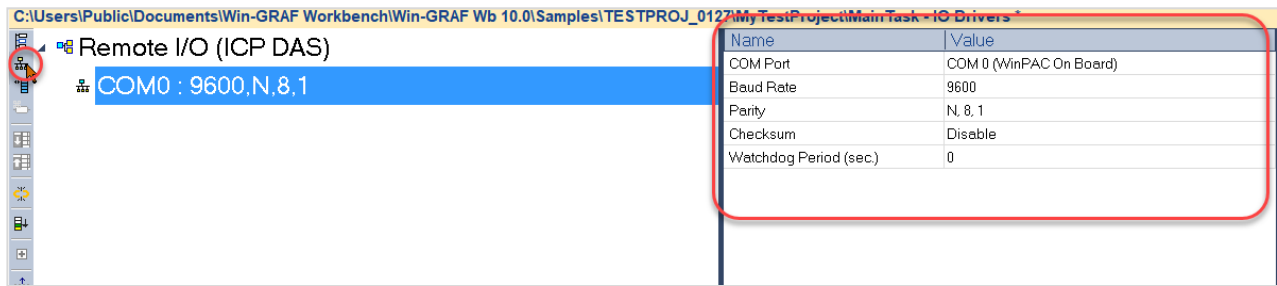
1. 雙擊" Fieldbus Configuration " 取得視窗如下圖。



2. 點擊左上角的 " Insert Configuration "，這會顯示一個對話視窗，雙擊 "All" 選擇 "Remote I/O (ICPDAS)"，點擊 OK。



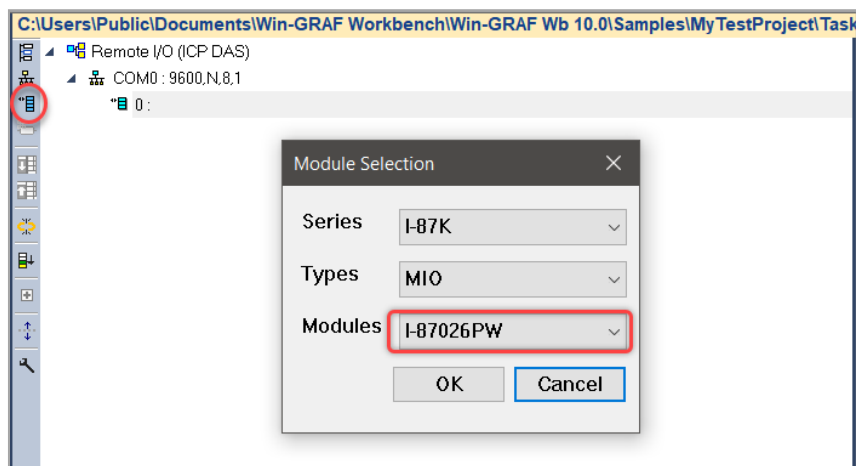
3. 然後點擊左上角第二個"插入 Master/Port" 加入 1 個 COM0，在右側顯示 COM0 的資訊。



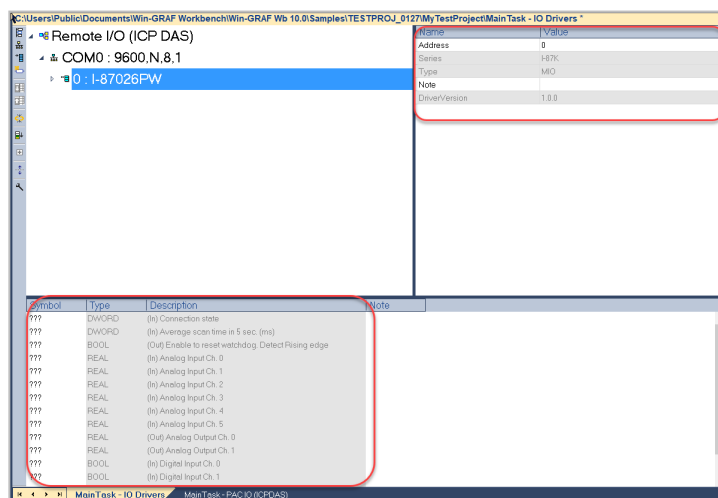
備註: 基本上不同的 PAC 項目 (例如: XPAC/WinPAC...)，他們使用不同的通訊埠來做插槽通訊。如需詳細訊息，請在需要幫助時參考硬體手冊

備註: 波特率、同位元和校驗需符合硬體設定。硬體設定可透過在 PAC 上的 DCON Utility 取得。

4. 點擊左上角的第三個圖標"Insert Slave/Data Block"，以加入您的 main PAC 模組。這裡我們有 XP-8000 作為 main 單元，並且我們選擇 I-87026PW 的 I/O 模組，點擊 OK。



5. 加入 I-87026PW 模組後，在右上視窗顯示這個遠端 I/O 的插槽位址，I/O 系列和 I/O 型態。視窗下方顯示資料型態的類型與詳細說明。

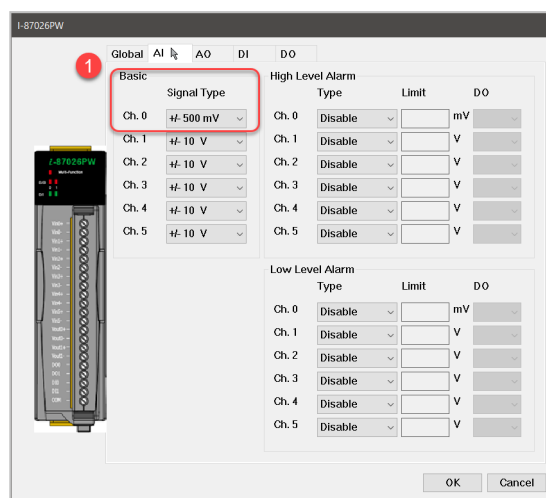


備註: 位址需符合硬體設定。硬體設定可透過在 PAC 上的 DCON Utility 取得。

3.2.2 變數綁定

在大多情況下，變數應該被 "綁定" 攜帶設定。以 I-87026PW 為範例:

步驟 1. 雙擊 I-87026PW 圖片會顯示此對話框，選擇 "AI" 頁籤，設定 Ch.0 訊號型態為 +/-500mV 並點擊 OK。



步驟 2. 在 PAC 圖片下方的變數區，雙擊 AI0 的 "???" 欄位，在空白處輸入變數名稱 (這裡我們輸入 Temp) 並點擊 OK。

Name	Value
Slot	1
Name	I-87026PW
Note	
Driver Versi...	1.0.0

Symbol	Name	Type	Setting
???	Connection State	DWORD	
???	Scan Time	DWORD	
???	Reset Watchdog	BOOL	
???	AI 0	REAL	08,D,D
???	AI 1	REAL	08,D,D
???	AI 2		08,D,D
???	AI 3		08,D,D
???	AI 4		08,D,D
???	AI 5		08,D,D
???	AO 0		33,0,0,0,0,0
???	AO 1		33,0,0,0,0,0
???	DI 0		D
???	DI 1		D
???	DO 0		0,0,D
???	DO 1		0,0,D

Variables: (all)
Temp

Variables: (all)
☐ Local variables only
☐ Hide FB instances

因此，訊號類型 (或通道的設定) 將透過綁定變數攜帶到運行時，並且綁定設定正確的完成。

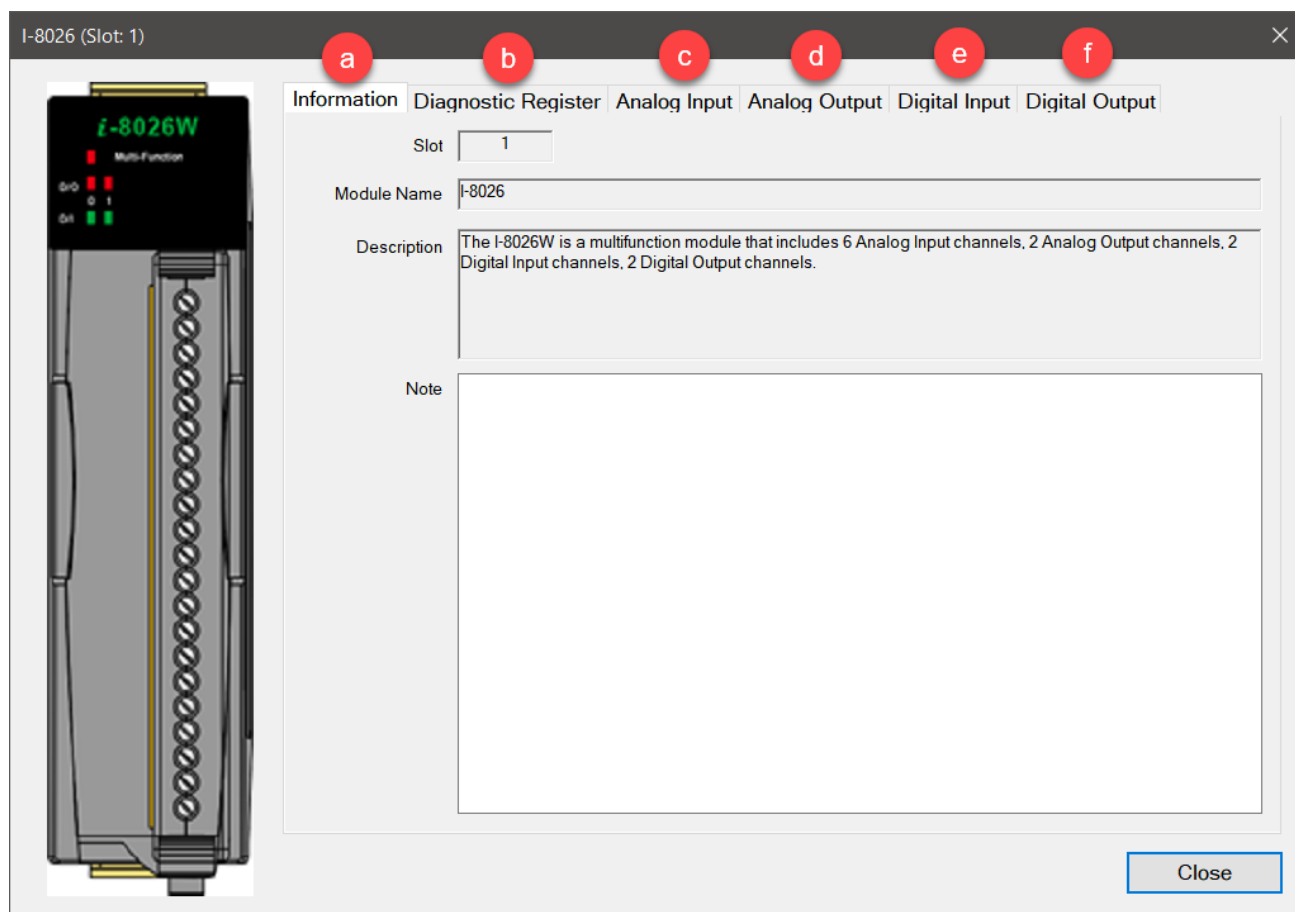
3.3 I/O 模組設定

在上一節中，我們有加入 PAC 與 I/O 模組到 Win-GRAF Workbench，我們也完成變數綁定的設定以允許資料讀取與寫入。接下來的部分，我們將開始對這些 PAC I/O 產品做一些的基本設定，我們也提供一些讓用戶完成設定時應該知道的提示。

3.3.1 I-8K Series

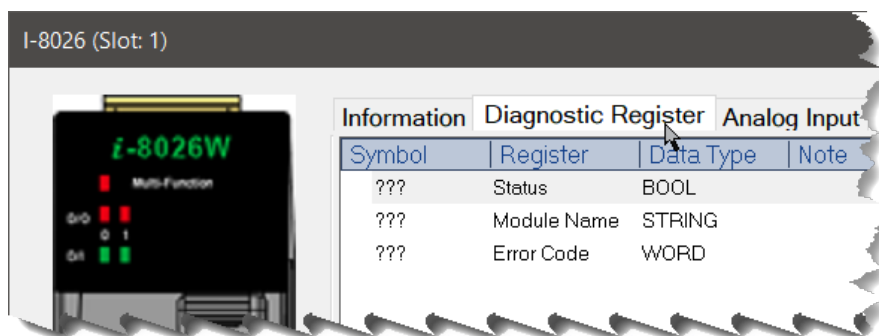
以 I-8K 系列 (I-8026W 具有 6AI, 2AO, 2DI 和 2DO) 為例：

只要您雙擊想要設定的模組，下圖就會顯示關於這個模組的產品圖片和一些書籤。

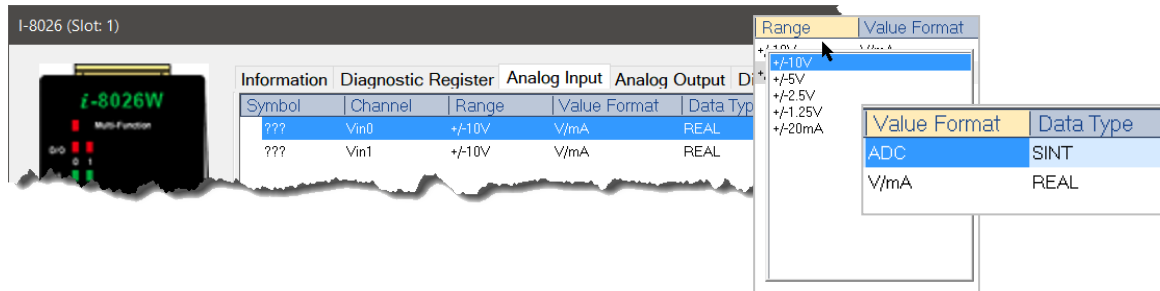


a. Information: 顯示基本模組詳情和產品介紹。

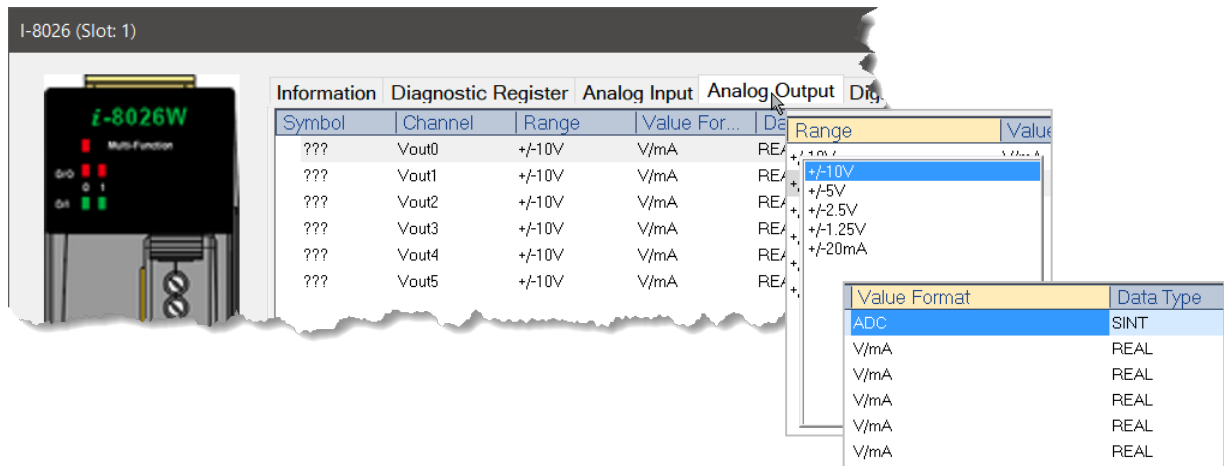
b. Diagnostic Register: 顯示這個模組的資料型態，並允許 PLC 程式直接存取模組名稱、通訊狀態和存取模組失敗的錯誤型態。



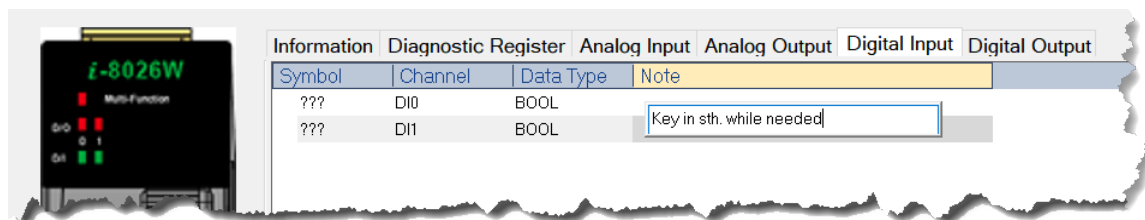
c. **Analog Input:** 找到 "Range" 並雙擊以下欄位選擇 Voltage input，再雙擊 "Value Format" 欄位改為 ADC，"資料型態" 將相對應的更改為 SINT。



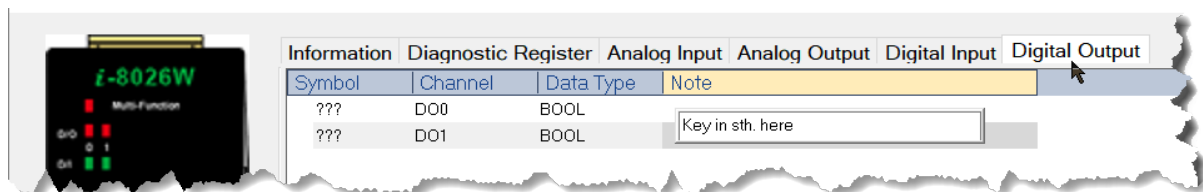
d. **Analog Output:** 與步驟 c 相同，更改 "Range" 和 "Value Format"。



e. **Digital Input:** 此圖顯示這個模組的 Digital Input，如果需要輸入任何內容，雙擊 "Note" 欄位。



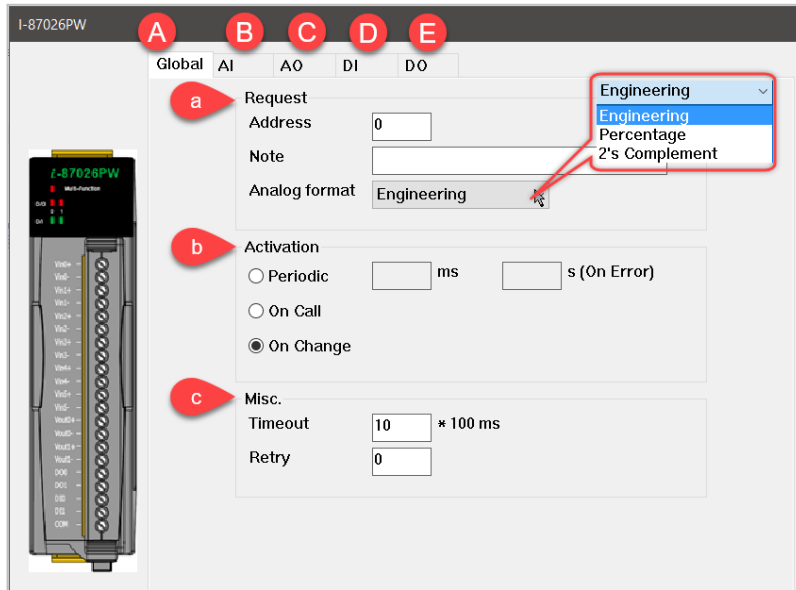
f. **Digital Output:** 同步步驟 e，如果需要輸入任何內容，雙擊 "Note" 下方欄位。



3.3.2 I-87K 系列

以 I-87K 系列 (I-87026PW 具有 6AI, 2AO, 2DI 與 2DO) 為例:

與 I-8K 相比，我們在設定 I-87K 模組時提供不同的環境，請查看下圖。



A. Global: 全域(Global)頁籤顯示有關模組的圖片和資訊。

a. Request

a-1. *Address:* 模組序列埠上的唯一編號。

a-2. *Note:* 使用者備註。

a-3. *Analog Format* (僅限 AIO 模組): 這裡有類比格式 3 個不同的類型，請從下拉選單選擇一個適合您的設定。

b. Activation

b-1. Periodic:

當模組連接後，PAC 會定期發送帶有設定的 IO 命令。

當模塊連接中斷時，PAC 將定期發送初始命令連接模塊，設置為“On Error”。

b-2. On Call:

當模組連接並啟用變數(自動生成)被觸發時，PAC 將會發送 IO 命令。

當模組連接中斷並啟用變數被觸發時，PAC 將會發送初始化命令給連接模組。

b-3. On Change:

當模組連接後，PAC 會在輸出值有變化時，發送 DO 或 AO 命令，並連續發送 DI 或 AI 命令。

當模組連接中斷時，PAC 將每 1 秒發送一次初始值命令。

c. Misc.

c-1. Timeout:

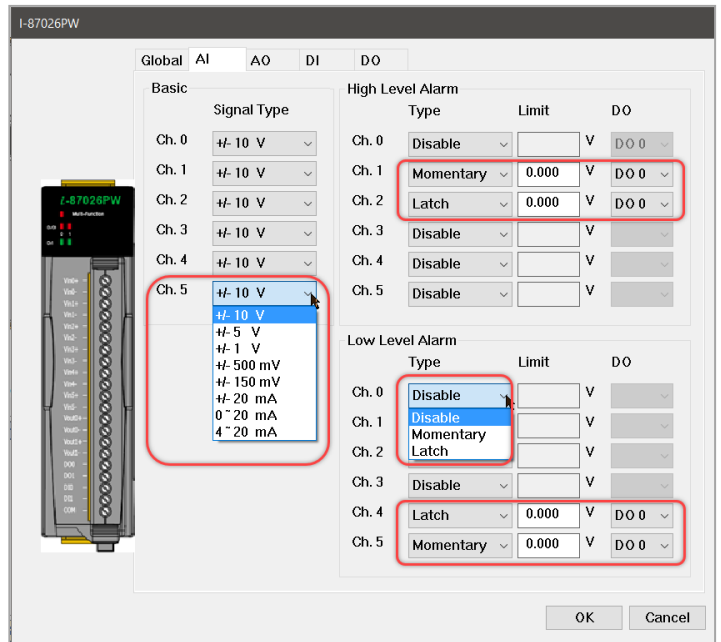
當模組回應超時，PAC 會判定模組已中斷連接。

c-2. Retry:

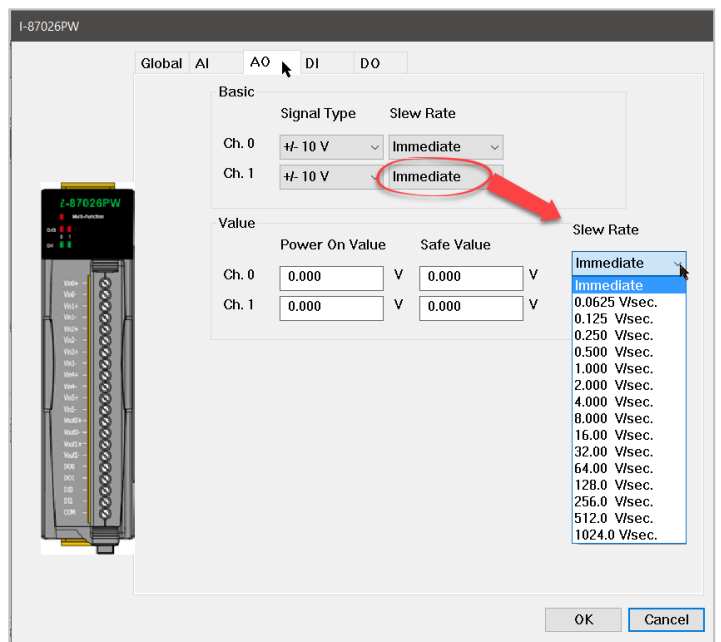
當模組回應超時，PAC 將根據設定的次數重新發送最後一條命令。重發次數達到設定次數後，會重發初始化命令。

B. AI: 頁籤顯示類比 input 訊號的"Signal Type"，左擊下拉選單列表，並從這裡選擇訊號範圍。

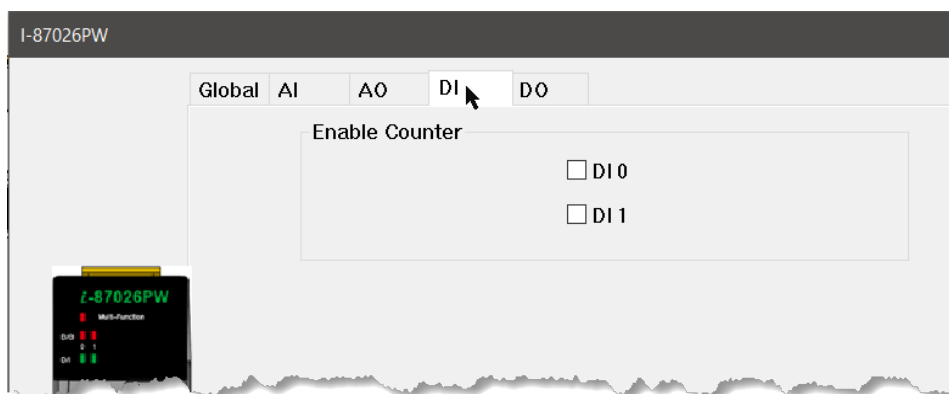
此外，I-87026PW 模組的 AI 通道可進行 "High Level Alarm" 或 "Low Level Alarm"，左擊下拉式選單列表，選擇型態 "Momentary" 或 "Latch"，再輸入訊號限制資料，並在最後 DO 的部分設定輸出通道。



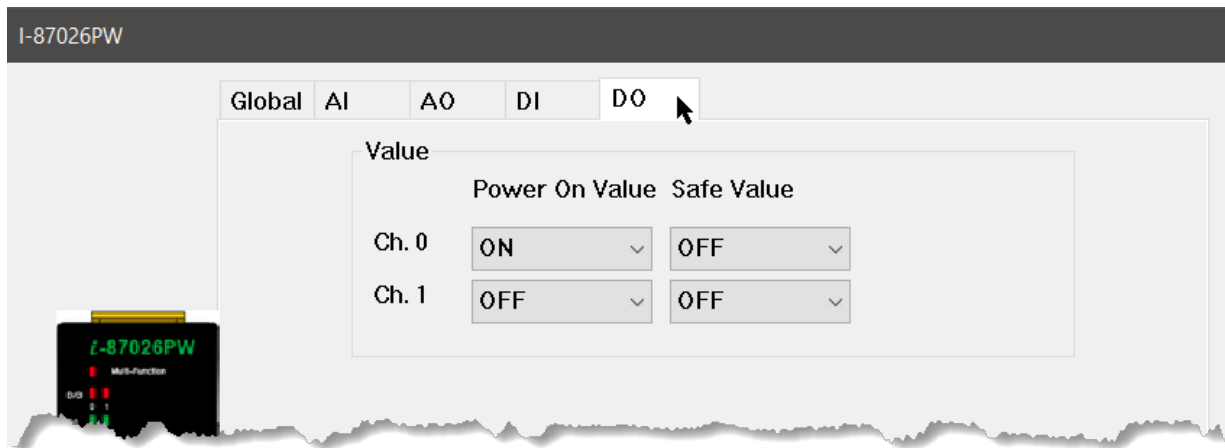
C. AO: 讓我們進入 AO 頁籤，這裡顯示 I-87026PW 的 2 通道 AO 和訊號型態同上。AO 最重要的部分是設定 "Slew Rate"，意思是輸出通道可以多快回應 input 範圍突然的變化。最大速率範圍從 0.0625V/sec. 到 1024V/sec.



D. DI: 勾選此框來開啟 DI 計數器。I-87K 模組的 DI 通道支援 digital input 和低速 (100Hz) 計數器輸入。2 個功能可同時工作，開啟計數器，模組的更新率變慢。



E. DO: 如果需要設定開機值和安全值，請拉下選單中的 "ON" 或 "OFF" 來開啟此功能。



4 保存變數

什麼是保存變數? 保存變數是一個變數:

- 是非揮發性記憶體是儲存在一般硬碟檔案。
- 所有程式都知道 (當內容有改變時，會傳播到使用該變數的所有方程式。)
- 通常不包含即時關鍵資料。

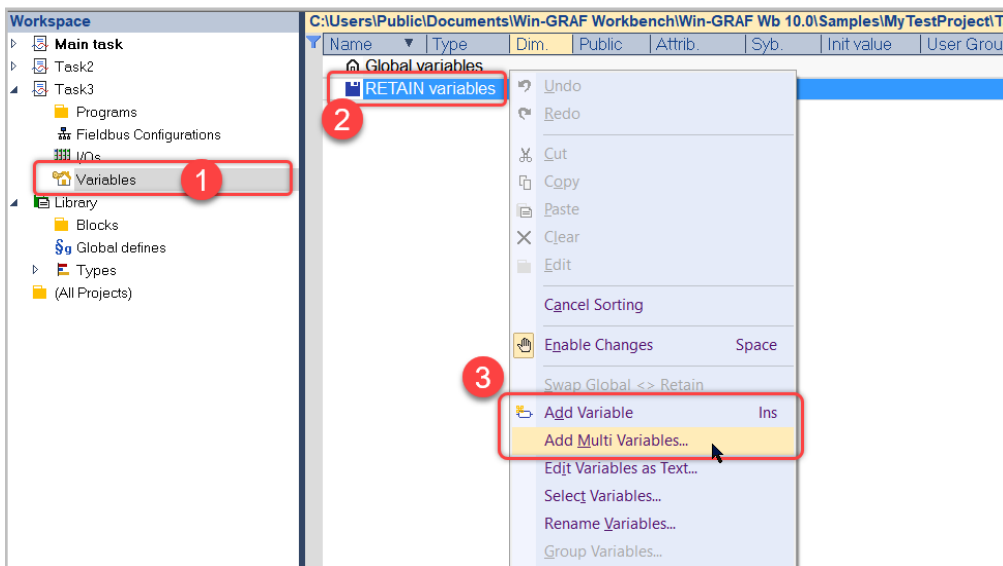
換句話說，保存變數用來保護重要資料。當發生意外關閉時，系統會保存最新的變數資料。

4.1 保存變數

4.1.1 在 Workbench 加入保存變數

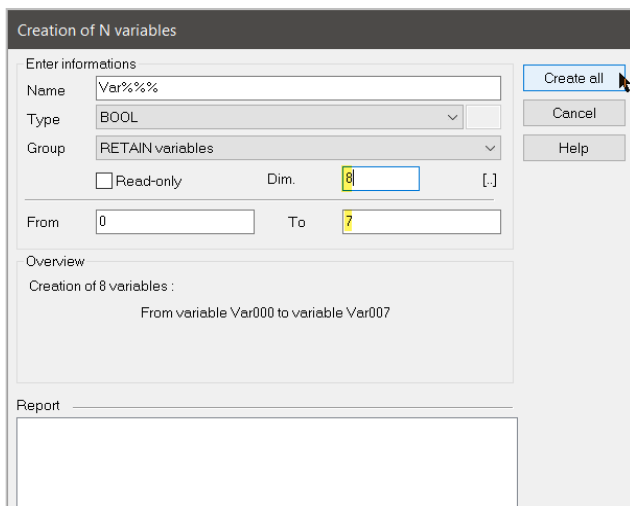
保存變數是在變數編輯器的 'RETAIN variables' 區塊宣告。(下圖)

1. 在工作區，雙擊 „Variables“。
2. 在編輯區，右鍵點擊 „RETAIN variables“。
3. 從清單中選擇加入 1 個 或多個變數。



<圖 4.1.1:保存變數宣告>

在這作個範例，我們選擇 „Add Multi Variables“，然後會顯示一個對話視窗。



首先確認變數的 „Group“ 顯示 „RETAIN variables“。

輸入變數的總數 (dimensions)，這裡我們輸入 „8“ 作為範例。

接下來，我們宣告變數從 0 到 7，因此我們將建立 8 個保存變數。

點擊 „Create all“ 按鈕，然後再次點擊 „Cancel“ 關閉這個對話視窗。

關閉對話視窗後，在 RETAIN variables 區域，會顯示 8 個添加變數在清單。

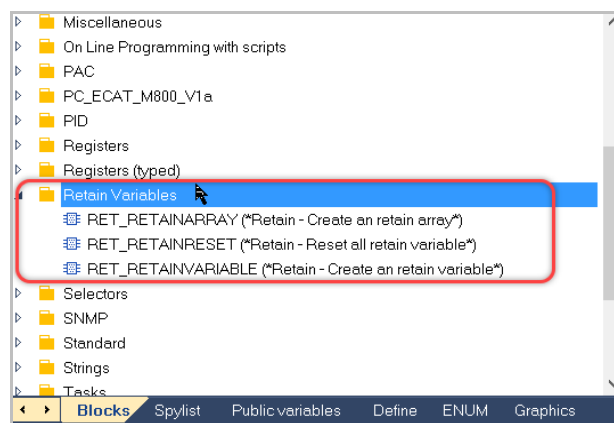
Name	Type	Dim.	Public	Attrib.	Syb.	Init value	User Group	Tag	Description
Global variables									
RETAIN variables									
Var000	BOOL	[0..7]	☐		☐				
Var001	BOOL	[0..7]	☐		☐				
Var002	BOOL	[0..7]	☐		☐				
Var003	BOOL	[0..7]	☐		☐				
Var004	BOOL	[0..7]	☐		☐				
Var005	BOOL	[0..7]	☐		☐				
Var006	BOOL	[0..7]	☐		☐				
Var007	BOOL	[0..7]	☐		☐				

備註: 當保存變數修改過，已儲存的保存值將會全部清除。

4.2 永久保存變數

由於上一節中有說明，ICP DAS 提供另一個 „Persistent Retain“來防止所有值被清除。一般來說，每個都有自己任務。我們讀取或寫入資料是辨別變數的 ID 而不是名稱。

這裡有 3 個功能方塊可以實現永久保存:



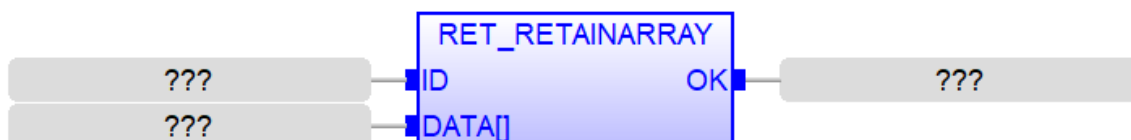
4.2.1 RET_RetainVariable

RET_RetainVariable		
This Function Block is used to create a Persistent Retain Variable.		
NAME	TYPE	DESCRIPTION
Input		
ID	UINT	A unique ID number in the Project
Data	ANY	Retained value
Output		
OK	BOOL	TRUE: Successful / FALSE: Fail



4.2.2 RET_RetainArray

RET_RetainArray		
This Function Block is used to create a Retain Array.		
NAME	TYPE	DESCRIPTION
Input		
ID	UINT	A unique ID number in the Project
Data[]	ANY[]	Retained array
Output		
OK	BOOL	TRUE: Successful / FALSE: Fail



提示: 如果您需要大量的保存資料，我們建議使用 „ARRAY“ 方法更有效率。

4.2.3 RET_RetainFormat

RET_RetainFormat		
This Function Block is used to clear all the Retain Variables.		
NAME	TYPE	DESCRIPTION
Input		
ENABLE	BOOL	Rising edge to trigger CLEAR
Output		
Q	ANY	No output

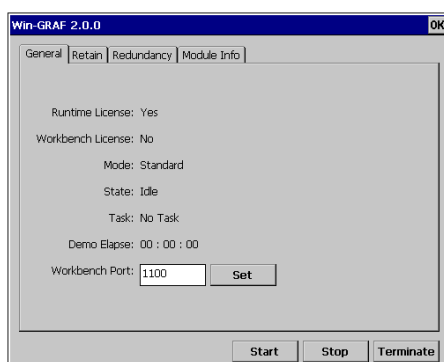


4.3 匯出& 匯入保存變數

大多數的情況下，我們會無法取得或知道暫存在 SRAM 中的變數。

在這一節, 我們將介紹如何在 SRAM 中取出變數/ 設定並備份和移動到另一個運行。

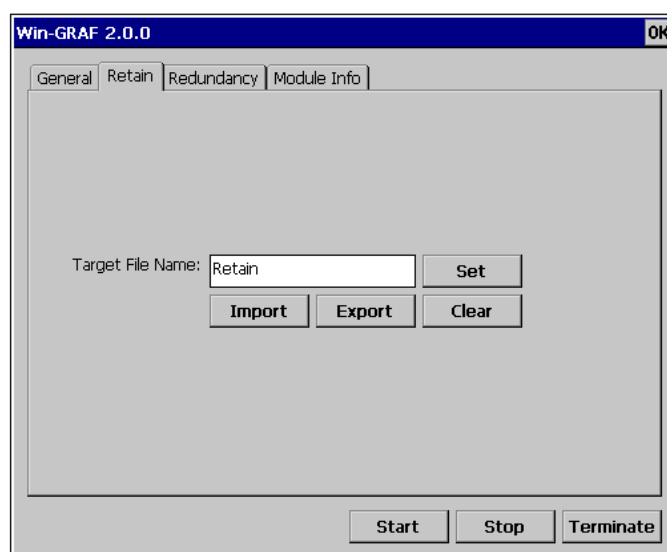
1. 停止您的運行專案。



2. 輸入 Target 檔案名稱並點擊 „Set“ 按鈕，然後點擊 „Export“ 按鈕，跳出對話視窗點擊 „OK“ 來關閉它。

Export: SRAM 到二進位(binary)檔案。

Import: 二進位(binary)檔案 到 SRAM。



3. 打開 PAC 的系統硬碟，找出名為 „Retain“，檔案類型 „.RET“ 的檔案。左鍵點擊複製和備份這個檔案到你的 PC 或其他外部的儲存設備。

4. 將 RET 檔案複製回到新 PAC 的系統硬碟，點擊 „Import“ 按鈕，跳出對話視窗。點擊 „OK“ 關閉它。

5. 點擊 „Start“ 再次執行運行，現在資料在新 PAC 繼續記錄。

4.4 保存記憶體空間

儲存保存變數的總記憶體大小的不同，取決於 PACs 型號不同。詳情如下表:

PAC 型號	記憶體類型	總尺寸	保存變數尺寸	永久保存尺寸
WP-5238	FRAM	8 KB	4 KB	4 KB
WP-8x28	MRAM	7 KB	3.5 KB	3.5 KB
WP-9x28	MRAM	7 KB	3.5 KB	3.5 KB
XP-8x38	MRAM	131 KB	65.5 KB	65.5 KB
VP-x238	MRAM	7 KB	3.5 KB	3.5 KB
VP-x208	MRAM	7 KB	3.5 KB	3.5 KB

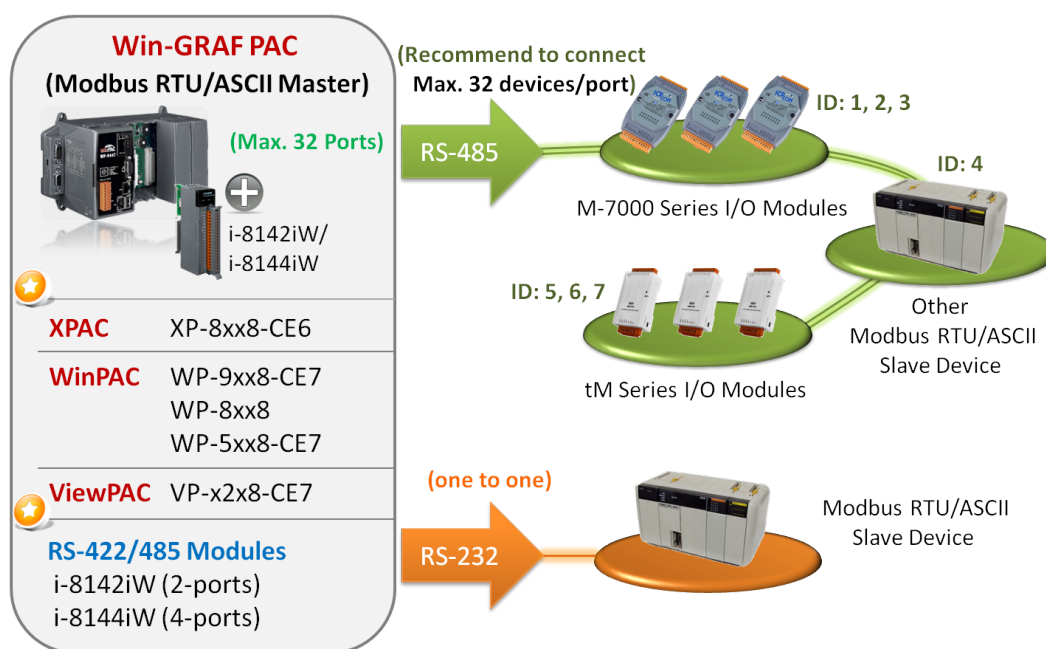
5 Modbus Master: 連接 Modbus Slave 設備

Win-GRAF Modbus master 支援 3 種通訊協議: Modbus TCP, Modbus RTU 與 Modbus ASCII。同一個 PLC 應用程式可以實現多個 Modbus master。每個序列通訊埠 (RS232、RS485、RS422) 和多個 Modbus TCP (以太網) 主站只能建立一個主站 (Modbus RTU、Modbus ASCII)。

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

以下顯示如何建立 Modbus RTU master，它透過序列埠 COM1 與 Modbus slaves 網路進行通訊。

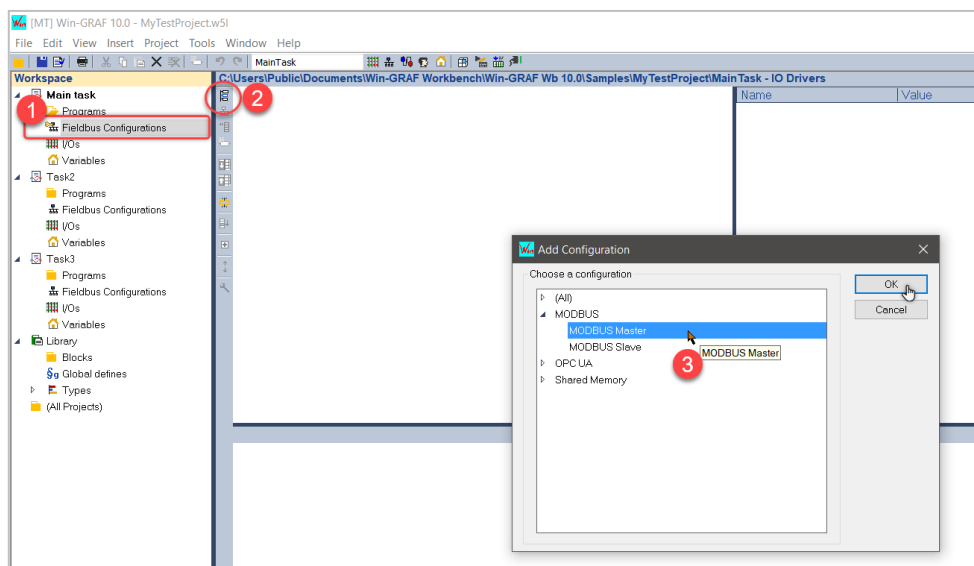
應用示意圖:



5.1.1 設定通訊介面

❖ 選擇 Modbus Master 作為 fieldbus configuration

1. 雙擊 "Fieldbus Configuration" 按鈕，並點擊 "Insert Configuration"。
2. 從跳出的對話視窗下拉式列表，選擇 "MODBUS Master" 並點擊 OK。

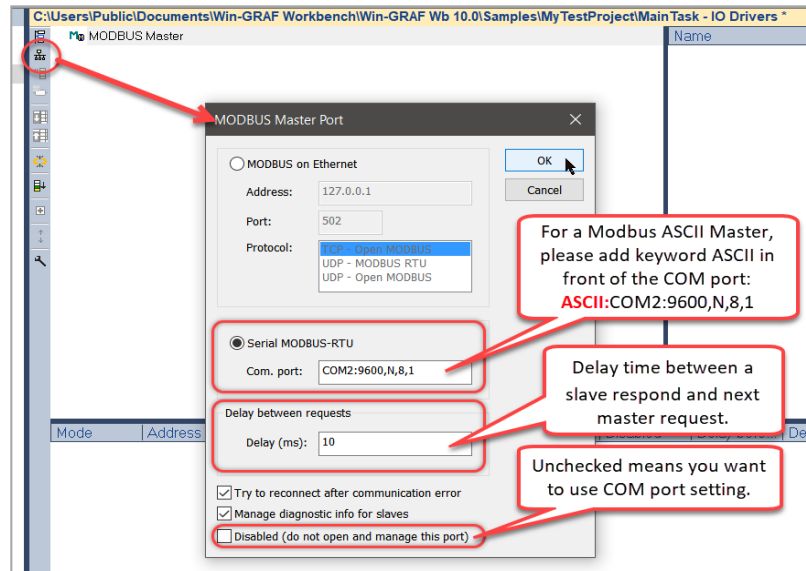


❖ 加入 Modbus RTU master 到 workbench 並作設定

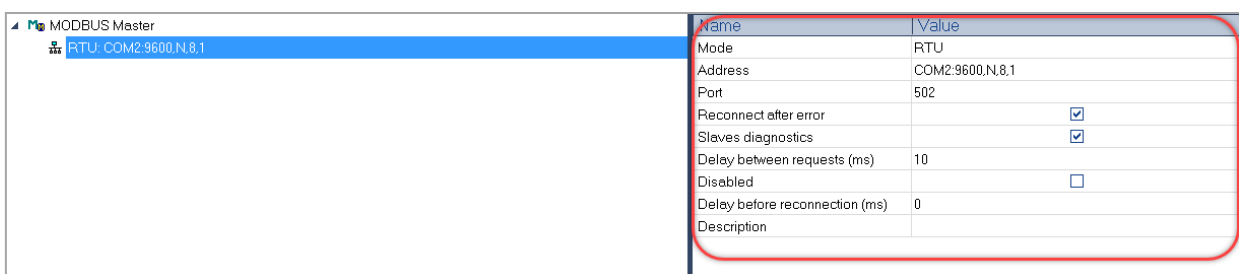
1. 點擊 “Insert Master/Port” 來打開跳出的對話框，選擇 “Serial MODBUS-RTU” 下方，並輸入 COM port 設定的相關資料，點擊 OK。

備註: 序列埠配置格式

[通訊埠,速度,同位元,資料位元,停止位元] → 範例: COM3,9600,N,8,1



2. Master 通訊參數顯示在屬性視窗，可透過雙擊 master 修改或在屬性視窗直接修改設定。



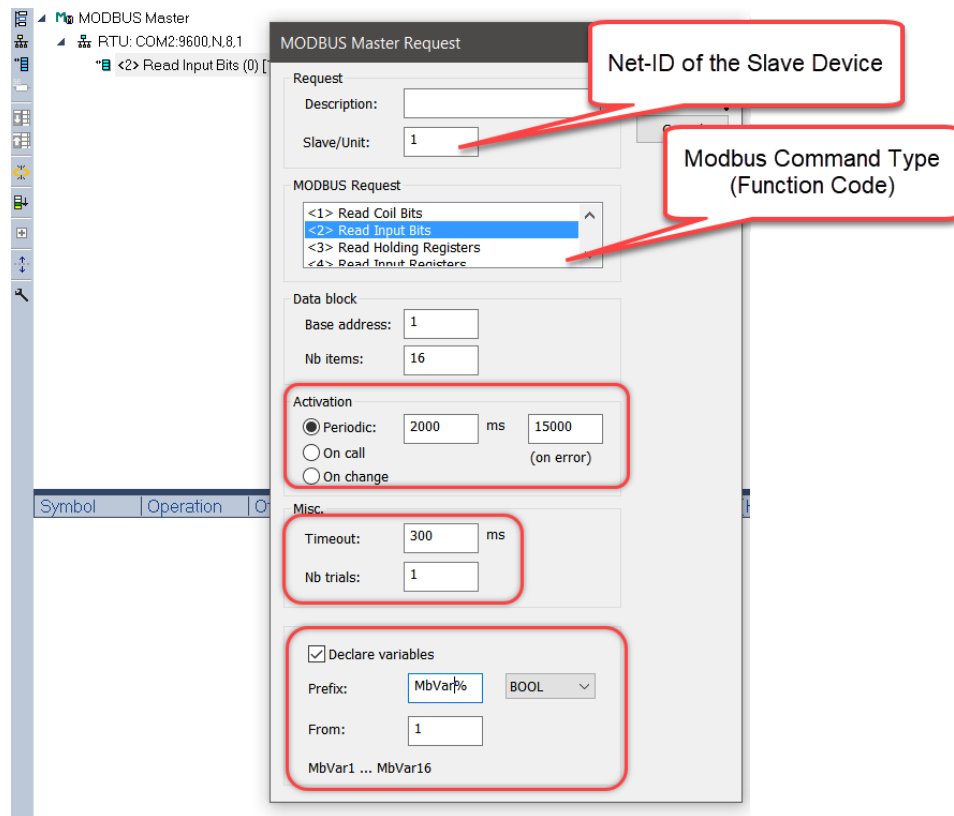
❖ Master 加入一個 data block，儲存從 slave 接收到的 input 資料和發送到 slave 的 output 資料。

設定保存 slave 資料的 data block。master 的 data block 設定需是 slave 表的精確表示。換句話說它需是 slave data block 的大小和資料型態的圖像。從 slave 讀取的資料被儲存在 data block 讀取部分，並且在 data block 寫入部分被修改的資料已被發送到 slave。

型態	Function Code	Modbus Request	說明
Read	1	Read coil bits	讀取數位輸出 (DO) 資料
	2	Read input bits	讀取數位輸入 (DI) 資料
	3	Read holding registers	讀取類比輸出 (AO) 資料
	4	Read input registers	讀取類比輸入 (AI) 資料
Write	5	Write single coil bit	寫入數位輸出 (DO) 資料
	6	Write single holding register	寫入單個類比輸出 (AO) 資料 (16-bits)
	15	Write coil bits	寫入多個數位輸出 (DO) 資料
	16	Write Holding Registers	寫入多個類比輸出 (AO) 資料 (16/32 bits)

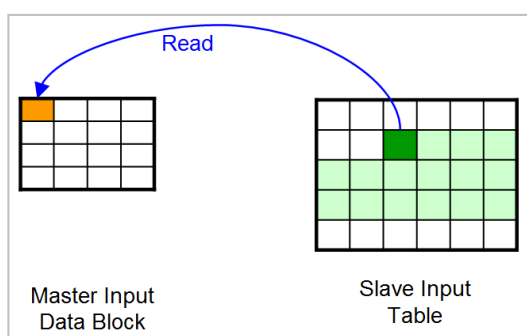
<表 5.1: 功能碼表>

- 下圖建立 1 個包含 16 個 input bits 的 input data block，並由位址 1 開始讀取 slave discrete 輸入表。(範例: 設定數量為16)。
 a) 在左側點擊 'Insert Slave/Data Block' 按鈕，來建立 data block。
 b) Slave/Unit: 輸入 Slave 設備的站號。(範例:遠端 slave ID 為 '1')。
 c) MODBUS Request: 選擇用於存取 slave 表<功能碼表>的 Modbus 命令型態 (功能碼)。範例: <2> Read Input Bits



<圖 5.1: Modbus Data Block 設定>

- 每個 data block 都由一個 MODBUS slave 編號、base address 和項目/條目 (bits 或 words) 的來識別。項目/條目的數量受 MODBUS 限制 (讀取 2000 bits、強制 1968 bits、讀取 125 words 或強制 120 words)。
- Base address:** 預設值由 '1' 開始，它表示 master 開始讀取 Modbus slave 表起始位址。從起始位址讀取的資料將會被複製到 master input block data 區的的第一個位置。
 注意: 如果你想要更改 'Base address'，右鍵點擊 'MODBUS Master'，然後選擇 'MODBUS Master Addresses' 修改其值。
- Nb items:** 讀取 slave 表條目的數量



< Base address 設定為 9 和 數量設定為 16>

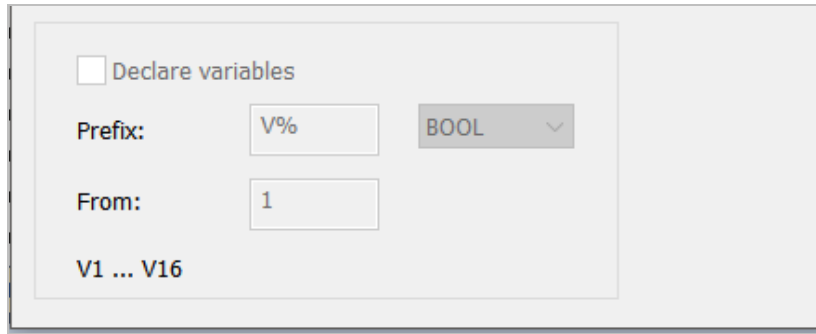
5. **Activation:** 設定 Master 如何觸發 Modbus 請求。
6. **Periodic:** 定期發送請求。(範例:為每 2 秒發送一次。)"on error" 表示 Modbus 發生異常後的下一次發送時間(範例:為 15 秒)。
7. **On call:** 該請求是通過使用變數來呼叫程式。如果對應到'命令(單次)'操作的變數從 false 變為 true，則會觸發 Modbus 請求。
8. **On change:** 一旦對應到輸出區的變數有改變，就會觸發請求。這個選項只能用於寫入命令，不能用於讀取命令。
9. **Timeout:** 設定超時值。如果 slave 在超時期間不回應 master 請求，將會產生超時錯誤。(建議 Modbus RTU/ASCII 通訊的超時值在 200 和 1000 ms之間)。
10. **Nb Trials:** 如果 slave 回應失敗 或 master 接收到無效的回應，master 將會重試設定的重試次數，然後繼續執行列表中的下一個命令。
11. **Declare variables:** 這個選項允許用戶宣告新變數，並且自動對應到 Modbus master data block，輸入變數名稱並在名稱尾末加入符號 '%'，然後輸入 '%' 符號的起始編號。workbench 用數字代替 '%'，該數字對於每個新變數都會遞增，如視窗的底部所示。新變數會自動對應到 Modbus master data block (見下圖)。

備註: 如果變數已經在編輯器宣告，則禁用這個選項。在這種情況下，變數必須從變數編輯器拖曳到對應區域來手動對應變數。

Symbol	Operation	Offset	Mask	Storage	Range (Lo...	Range (Hi...	Signal (Low)	Signal (Hi...
MbVar1	Data exchan...	0	FFFF	Default				
MbVar2	Data exchan...	1	FFFF	Default				
MbVar3	Data exchan...	2	FFFF	Default				
MbVar4	Data exchan...	3	FFFF	Default				
MbVar5	Data exchan...	4	FFFF	Default				
MbVar6	Data exchan...	5	FFFF	Default				
MbVar7	Data exchan...	6	FFFF	Default				
MbVar8	Data exchan...	7	FFFF	Default				
MbVar9	Data exchan...	8	FFFF	Default				
MbVar10	Data exchan...	9	FFFF	Default				
MbVar11	Data exchan...	10	FFFF	Default				
MbVar12	Data exchan...	11	FFFF	Default				
MbVar13	Data exchan...	12	FFFF	Default				

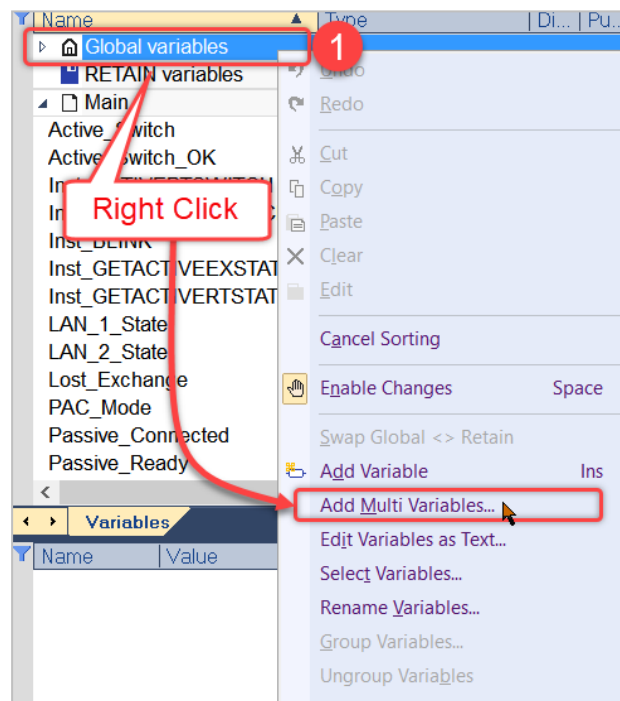
Mapping Area

❖ 只有在未選擇 'Declare variables' 選項時才需要完成以下步驟，如下圖。

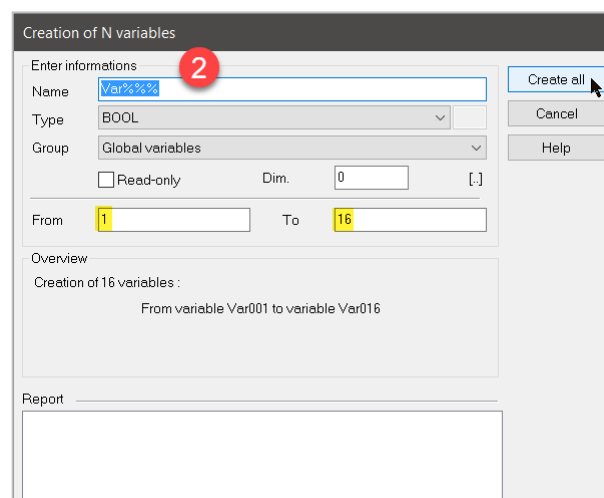


宣告新變數並拖曳變數到 data block 對應區域。

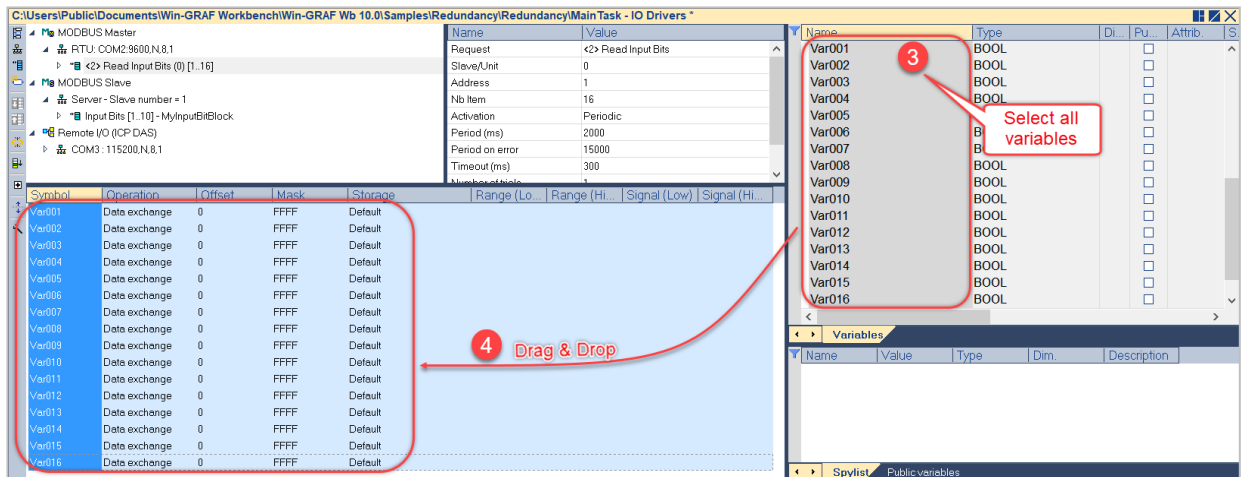
1. 右鍵點擊 ① “Global Variables” 加入多個變數。



2. ②輸入尾末帶有“%%”的變數名稱，選擇類型和變數數量 (在這作個舉例從 1 到 16)，並點擊 “Create all”。變數將顯示在右上欄。點擊 Cancel 來關閉 N 變數對話框。

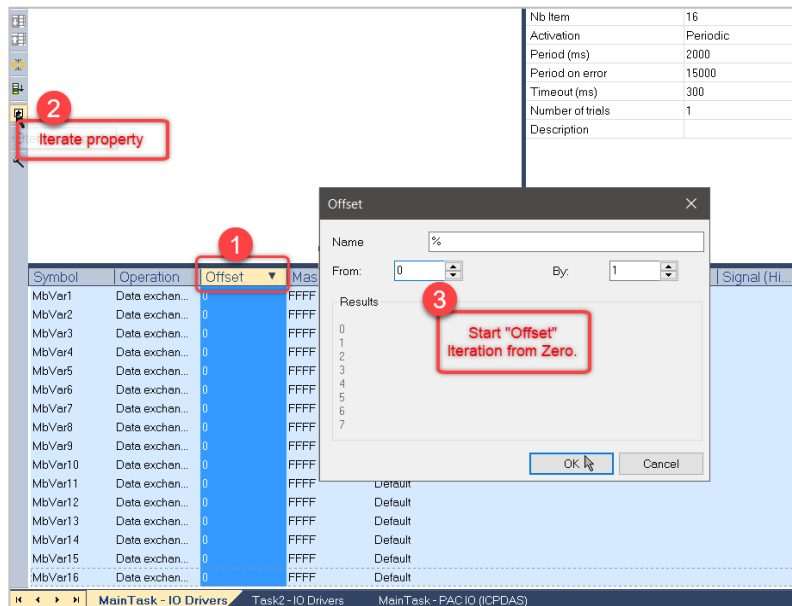


3. ③選擇剛才建立的所有 16 個變數並 ④拖曳回 Tabs 區，現在 offset 數全部是“0”。



4. Offset:

設定每個 data block 的 offset 變數。每個變數需有不同 offset 數，否則會發生記憶體重疊。



Symbol	Operation	Offset	Mask	Storage
MbVar1	Data exchan...	0	FFFF	Default
MbVar2	Data exchan...	1	FFFF	Default
MbVar3	Data exchan...	2	FFFF	Default
MbVar4	Data exchan...	3	FFFF	Default
MbVar5	Data exchan...	4	FFFF	Default
MbVar6	Data exchan...	5	FFFF	Default
MbVar7	Data exchan...	6	FFFF	Default
MbVar8	Data exchan...	7	FFFF	Default
MbVar9	Data exchan...	8	FFFF	Default
MbVar10	Data exchan...	9	FFFF	Default
MbVar11	Data exchan...	10	FFFF	Default
MbVar12	Data exchan...	11	FFFF	Default
MbVar13	Data exchan...	12	FFFF	Default
MbVar14	Data exchan...	13	FFFF	Default
MbVar15	Data exchan...	14	FFFF	Default
MbVar16	Data exchan...	15	FFFF	Default

5. Mask:

對應區域的 Mask 只需為 input 和 holding registers 設定，而對於 coil 和 input bits 將被忽略。
Modbus register 內存大小為 16 位元 (與 INT, UINT, WORD 相同)。mask (十六進制數) 可用於過濾掉 register 位元。

<範例>

Register Value	Mask (hex)	Result
65565	0001	1
65565	00FF	255
65565	FFFF	65565

6. Storage:

儲存欄僅對於 input 與 holding register 設定相關。為了交換 32 位元變數 (DINT, REAL...)，可以將兩個連續的 Modbus register 對應到一個變數。為了交換字串，可以將多個 register 對應到一個字串。

The screenshot shows the Win-GRAF software interface with the following components and annotations:

- MODBUS Master Request Dialog:**
 - Request: <3> Read Holding Registers (1, 5)
 - Slave/Unit: 2
 - Base address: 1
 - No items: 5
- Global variables Table:**

Name	Type	Di...	Pu...	Attrib	S...	Init v...	Us...	Tag
dwVar0	UDINT							
dwVar1	UDINT							
dwVar2	UDINT							
dwVar3	UDINT							
dwVar4	UDINT							
- Variables Table:**

Name	Value	Type	Dim.	Description
AAA				
BBB				
Main				
- Symbol Table:**

Symbol	Description	Offset	Mask	Storage
dwVar0	Data exch...	0	FFFF	DWORD (Low-High)
dwVar1	Data exch...	2	FFFF	DWORD (Low-High)
dwVar2	Data exch...	4	FFFF	DWORD (Low-High)
dwVar3	Data exch...	6	FFFF	DWORD (Low-High)
dwVar4	Data exch...	8	FFFF	DWORD (Low-High)

<圖 5.2: 對應 32 位元變數到 holding register >

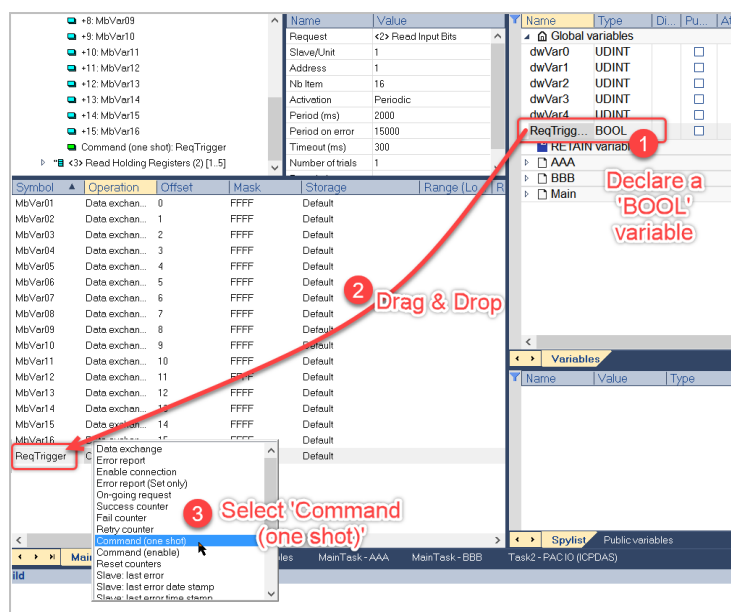
❖ 只有選中 'On call' 複選框時，才需要完成此步驟 ('Activation' 部分在圖 5.1)。如果 master 是 'On call' 模式時，Win-GRAF 運行時不會自動向 slave 發送請求。PLC 的邏輯程式必須觸發請求。有 2 種命令類型可用於發起請求：

1. 命令 (一次)

— 一旦附加標誌 (圖 5.3: '觸發請求') 從 FALSE 變為 TRUE，將發送請求。每次 PLC 邏輯將標誌切換為 TRUE 時，都會向 slave 發送一個 Modbus 命令。發送請求後，運行時標誌將自動重置為 0。

2. 命令 (啟用)

— 一旦對應到“命令 (啟用)”操作的 BOOL 變數由 PLC 邏輯設定為 TRUE，Modbus 主站將向 slave 發送請求，直到變數設定為 FALSE。如果變數設定為 FALSE，將不會發送任何命令。



<圖 5.3: 建立 “On call” 變數>

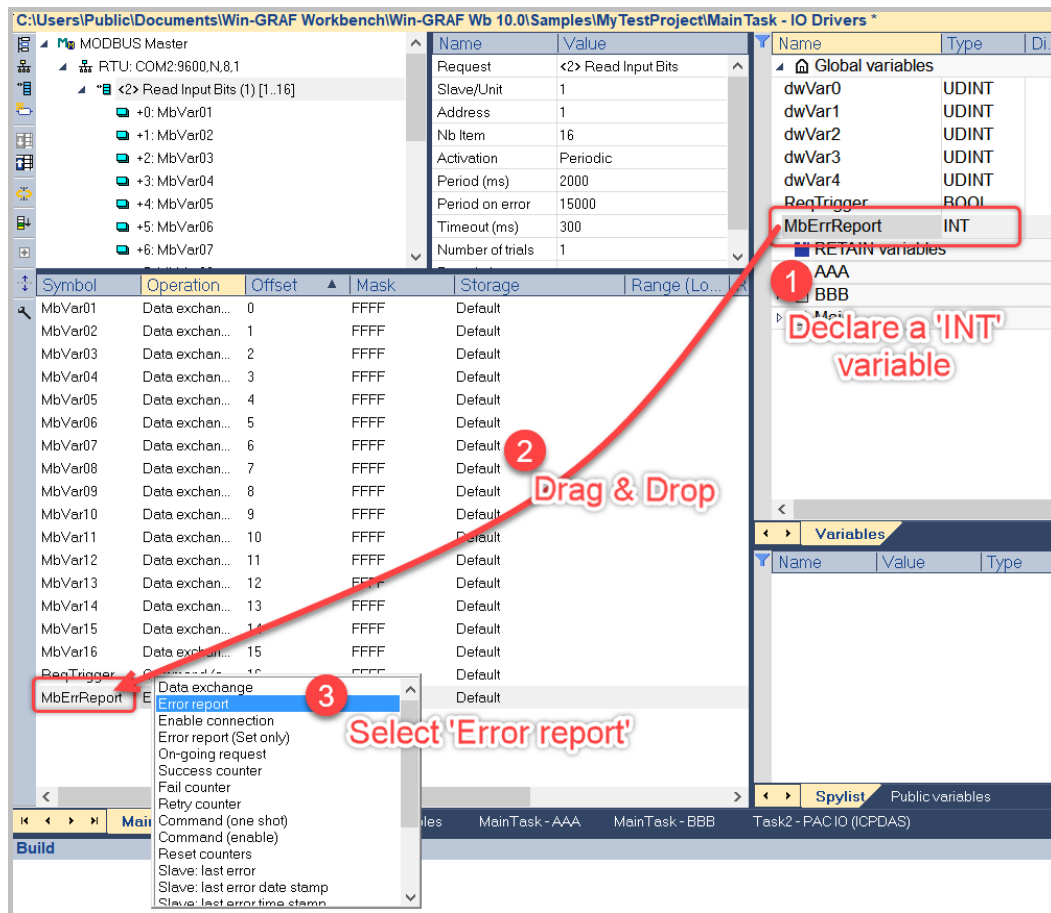
❖ 變數對應到 master 的診斷和狀態資訊。master 紀錄通訊超時，無效的資料位址，無效的命令，失敗的請求數量等...訊息。下方將變數對應到 'Error report'，該操作記錄錯誤的最後請求。表 5.2 顯示 'Error report' 操作使用所有的錯誤代碼。

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

<表 5.2: 用於 'Error report' 操作的錯誤代碼>

程序用於執行 'Error report' 程序 (圖 5.4):

1. 宣告 1 個 INT 變數 (例如: 'MbErrorReport')。
2. 拖曳新變數到對應清單。
3. 雙擊變數旁邊的 'Operation' 欄，然後選擇 'Error report'，這個操作會忽略 offset 和 storage 欄位設定。當 Modbus 發生請求錯誤時，變數將顯示錯誤代碼，並在下一次請求成功後重置為 0。



<圖 5.4: Error Report 設定>

6 Modbus Slave: 連接 Modbus Master 設備

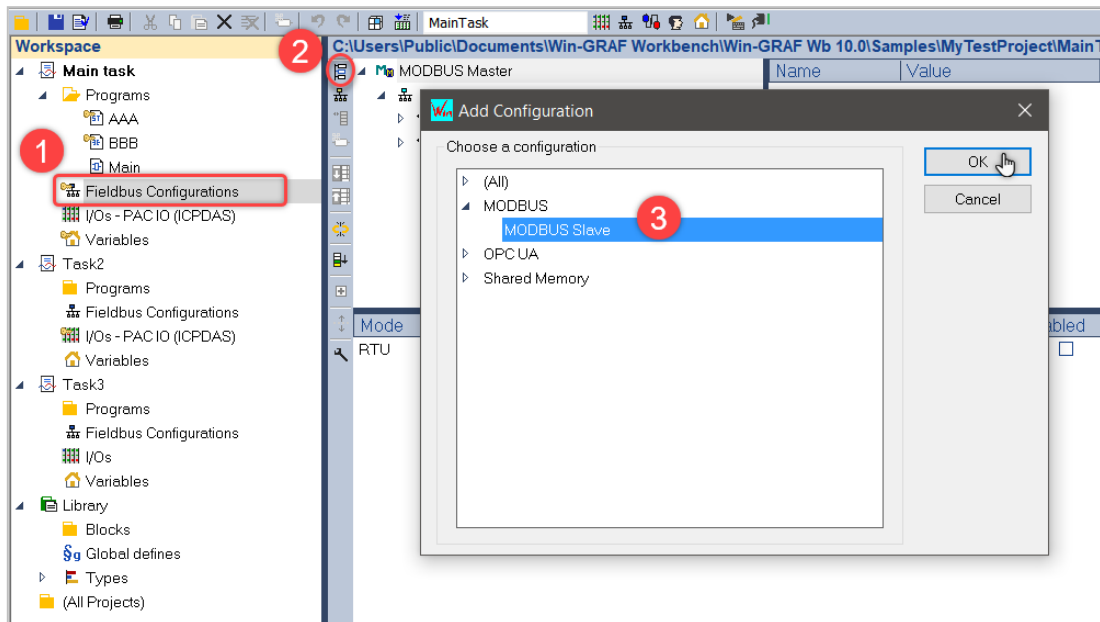
本章介紹如何設定運行時作為 Modbus slave，可安裝 3 種類型的 slave：Modbus TCP，Modbus RTU 和 Modbus ASCII。Win-GRAF 運行時可以模擬多個 slave。

Win-GRAF 運行時的 Modbus slave，首先必須通過 workbench 進行設定，以便遠端 Modbus master 存取。必須設定 slave 的通訊層 (Ethernet, serial) register 類型和 registers (data block) 數量。下方將依序介紹：

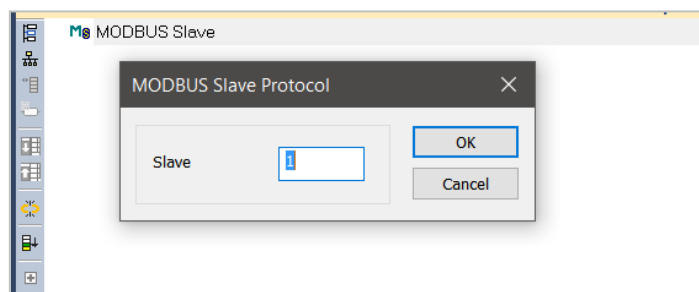
6.1 Slave Data Block 設定

6.1.1 選擇 Slave

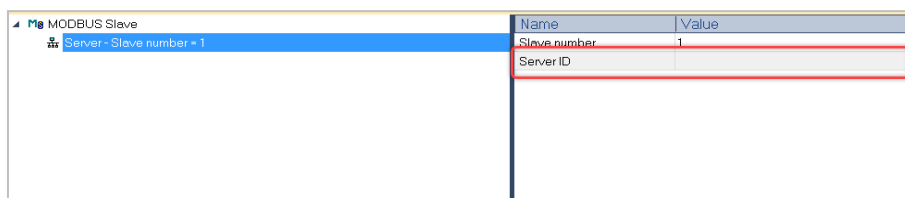
1. 雙擊“Fieldbus Configuration”打開 IO Drivers 視窗，從左上角視窗點擊“Insert Configuration”，選擇 Modbus→選擇“MODBUS Slave”然後點擊“OK”。



2. 點擊“Insert Master/Port”按鈕，設定 slave number (此範例輸入為‘1’)，然後點擊“OK”。



備註：如果你要加入更多個 Slave，請為每個 slave 分配一個 ID，以識別 PLC 程式中的 Server。

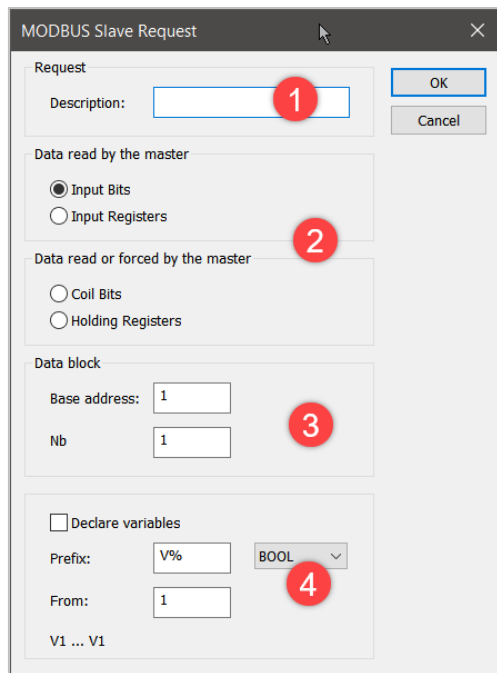


6.1.2 定義 Slave Register

Win-GRAF 支持以下標準 Modbus register 和 coils:

- Input register (masters 讀取)
- Holding register (master 讀取/寫入)
- Coils (master 讀取/寫入)
- Discrete inputs (masters 讀取)

Modbus slave register 類型和大小必須透過下圖進行設定。



如下說明:

1. Description 欄位允許你簡短的描述 data block 的用途。這個欄位可以空白。
2. 首先決定 master 是否只有讀取存取或同時擁有對 data block 的讀取和寫入存取。另外，決定 block 的資料格式

存取	選項	資料型態
Read	Input Bits	BOOL
	Input Register	BYTE, INT, DINT, REAL, etc.
Read/ Write	Coil Bits	BOOL
	Holding Register	BYTE, INT, DINT, REAL, etc.

3. 設定 data block 的 register (Nb)數量和起始位址 (base address)° Register 大小單位是 BIT 或 WORD 是取決於 2 做的

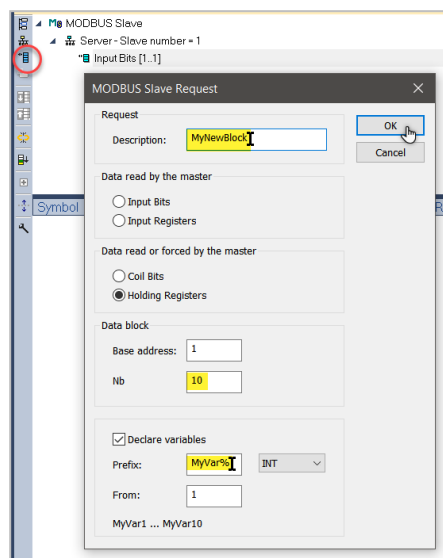
的選擇。建議將初始位址設定為 1。如果加入 2 個相同 register 類型的 blocks，請確認第 2 個 block 初始位址與第一個 block 結束位址連續。這是允許 master 存取資料包的 2 個 blocks，並且減少 master 和 slave 之間資料包交換的次數。如果 Modbus Master 請求的資料位址 (例如: SCADA 軟體)小於起始位址或大於最大位址 (起始位址 + Nb -1)，則 Win-GRAF 運行時的 slave 將不回應。

4. 請求 Workbench 自動宣告新變數，並直接 map 到新建立的 data block。

6.1.3 定義 Holding Register

❖ 此圖說明如何加入 holding register data block。

1. 點擊„Insert Slave/ Data Block“ 打開 slave 設定。名稱輸入 MyNewBlock' 並選擇,Holding Registers'。
2. Holding Register 設定為具有 10 個 registers (Nb=10) 和 base address 為 1。
3. 勾選該框宣告 10 個變數的資料型態為 INT'，並對應到 data block。輸入變數名稱為 ,MyVar%'; ,%' 代表值從 1 開始。
4. 點擊 OK' 完成。



加入“MyNewBlock”到 Modbus slave 和新 data block 後，將會自動宣告全域 (global) 新變數並分配給 data block。

New data block with the assigned variables.

Variables created by Workbench

Variables mapped to the slave data block

Symbol	Offset	Mask	Storage	Range (Lo...	Range (Hi...	Signal (Low)	Signal (Hi...
MyVar1	0	FFFF	Default				
MyVar2	1	FFFF	Default				
MyVar3	2	FFFF	Default				
MyVar4	3	FFFF	Default				
MyVar5	4	FFFF	Default				
MyVar6	5	FFFF	Default				
MyVar7	6	FFFF	Default				
MyVar8	7	FFFF	Default				
MyVar9	8	FFFF	Default				
MyVar10	9	FFFF	Default				

❖ 此圖說明如何加入 holding register 和對應全域 (global) 變數。

1. 點擊 “Insert Slave/ Data Block” 來打開 Slave Request 設定表。輸入 “MyNewBlock2” 作為名稱。
2. 加入 holding register 類型的 data block，保留 5 個 registers (Nb=5)，第一個 register 位址從 11 開始(base address)。因為前一個 block 以 10 結尾，為了讓相同的 Holding Register 位址範圍保持一致，所以這裡我們繼續用 11。
3. 稍後我們將宣告變數並對應到 holding register data block，所以我們不勾選此框，然後點擊 “OK”。

MODBUS Slave Request

Request

Description: MyNewBlock2

Data read by the master

☐ Input Bits

☐ Input Registers

Data read or forced by the master

☐ Coil Bits

☒ Holding Registers

Data block

Base address: 11

Nb: 5

☐ Declare variables

Prefix: V% INT

From: 11

V11 ... V15

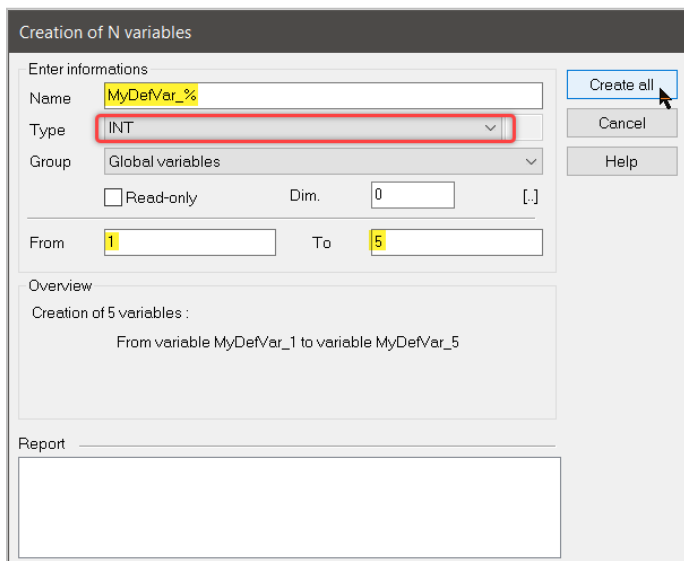
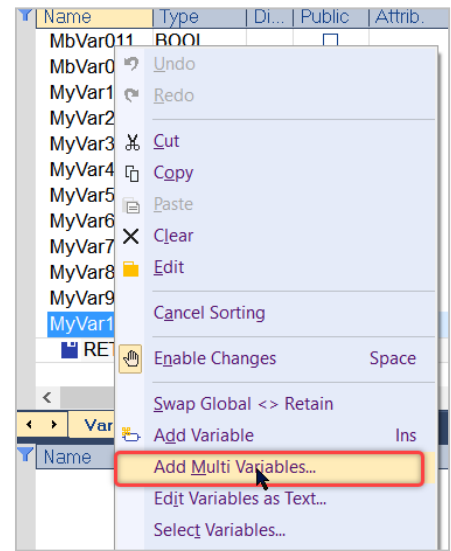
Unchecked

下方顯示 workbench “MyNewBlock2” 的擷取視窗。

Name	Value
Request	Holding Registers
Address	11
Nb Item	5
Description	MyNewBlock2

現在我們開始宣告 5 個 INT 型態的 global 變數。

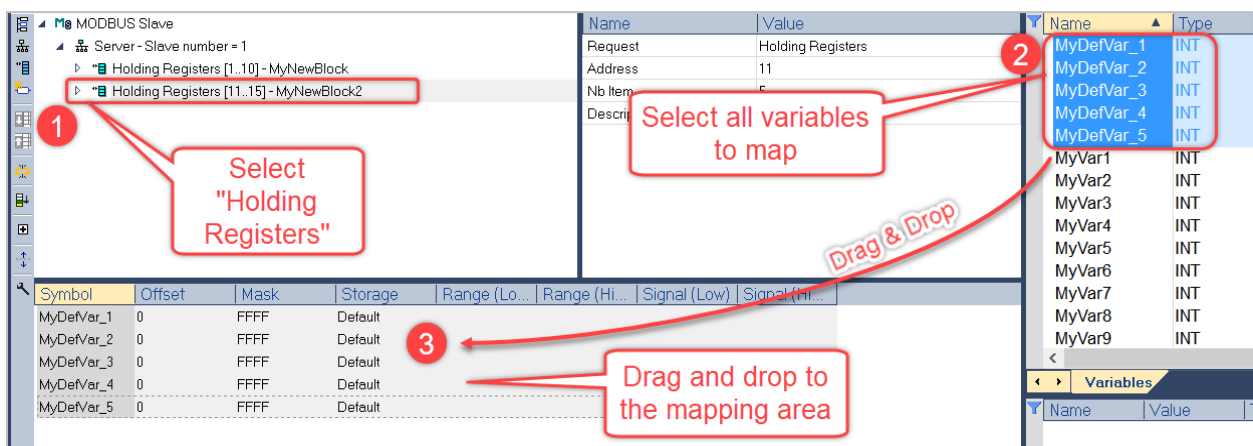
1. 右鍵點擊全域(global)變數並選擇 “Add Multi Variables...”。
2. 輸入變數名稱 (範例: 'MyDefVar_%')。 % 符號表示以遞增順序附加到變數名稱的數字。選擇宣告變數型態 (稍後我們必須對應 INT 型態，所以我們選擇 'INT')。
3. 起始編號必須輸入在 'From' 和 'To' 空白處，將變數群組定義為 'Global variables'。
4. 點擊 'Create all' 宣告變數，然後點擊 'Cancel' 來關閉這個視窗。



下圖顯示 Workbench 將宣告變數加入到變數編輯器。

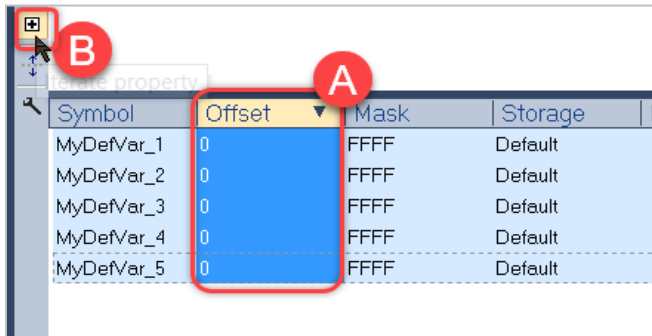
Name	Type	Di...	Public
MyDefVar_1	INT		☐
MyDefVar_2	INT		☐
MyDefVar_3	INT		☐
MyDefVar_4	INT		☐
MyDefVar_5	INT		☐

現在我們可以對應 5 個 “MyDefVar” 變數到 Modbus register data block。

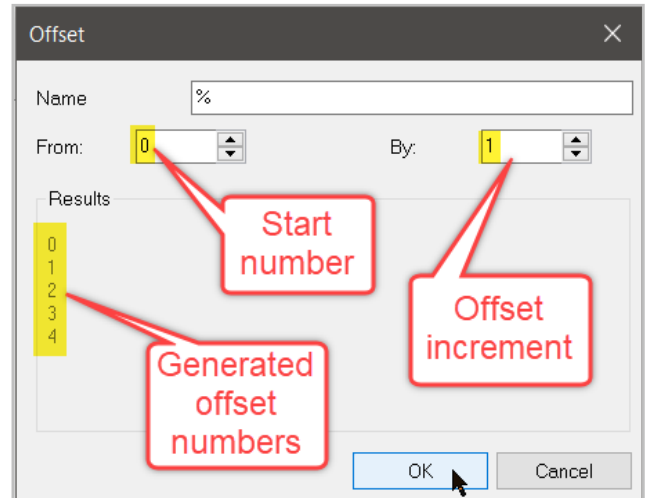


這裡有 2 種方式可改變變數的 Offset。

1. 點擊 offset 欄位的標題  來選擇所有變數。然後在左邊工具欄點擊  打開 Offset 視窗。輸入起始編號和遞增值。再次確認產生的 offset 數結果，並點擊“OK”。



Symbol	Offset	Mask	Storage
MyDefVar_1	0	FFFF	Default
MyDefVar_2	0	FFFF	Default
MyDefVar_3	0	FFFF	Default
MyDefVar_4	0	FFFF	Default
MyDefVar_5	0	FFFF	Default

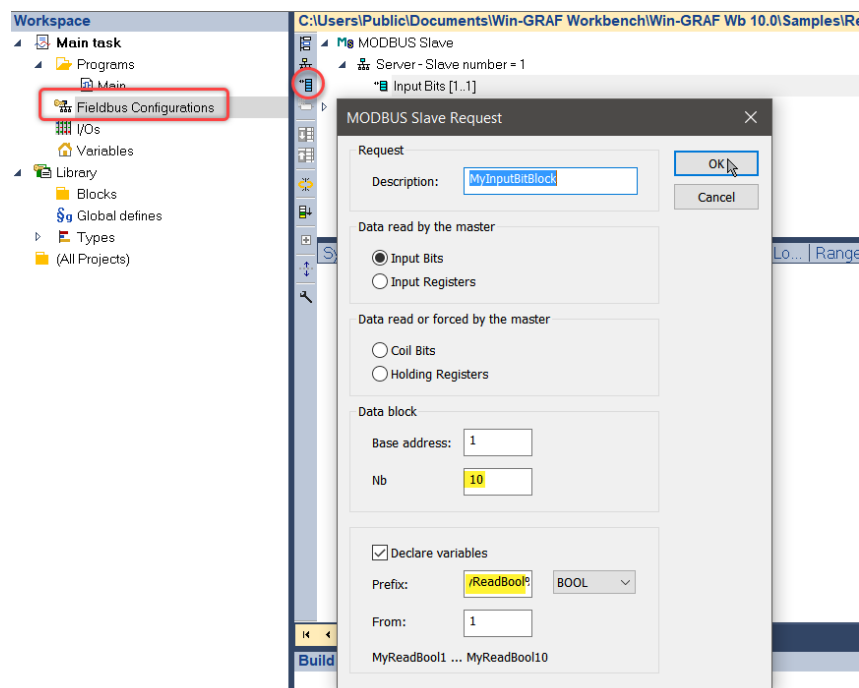


2. 雙擊 offset 欄位來分配 offset 位置。第一個變數 offset 從 0 開始，第二個變數從 1 開始，以此類推。

Symbol	Offset	Mask	Storage
MyDefVar_1	0	FFFF	Default
MyDefVar_2	1	FFF	Default
MyDefVar_3	2	FFFF	Default
MyDefVar_4	3	FFFF	Default
MyDefVar_5	4	FFFF	Default

6.1.4 定義 Input Bit Data Block

❖ 以下步驟說明如何加入只能被 master 讀取的 input bit data block。

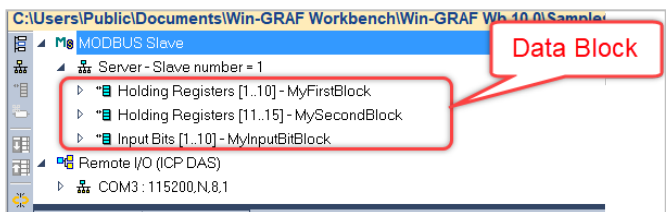


6.2 Slave 型態設定

上一節中設定完成的 Modbus data block 定義了slave 記憶體的大小和結構。本章介紹如何建立通訊介面，透過這個介面，Master 可與 slave data block 交換資料 (圖 6.2.1~6.2.3)。

1. 定義 Modbus data block 包含部分為:

- Input 與 output register
- Input 與 output bit



<圖 6.2.1>

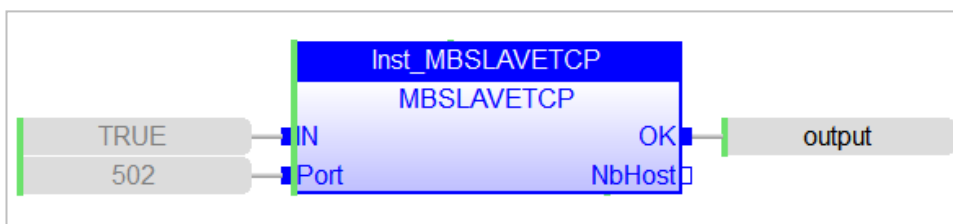
2. 對應 data block 到全域(global)變數。

Symbol	Offset	Mask	Storage
V1	0	FFFF	Default
V2	1	FFFF	Default
V3	2	FFFF	Default
V4	3	FFFF	Default
V5	4	FFFF	Default
V6	5	FFFF	Default
V7	6	FFFF	Default
V8	7	FFFF	Default
V9	8	FFFF	Default
V10	9	FFFF	Default

<圖 6.2.2>

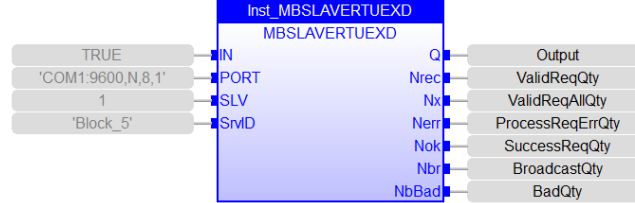
3. 設定 Modbus 通訊協定去存取資料區塊

- Modbus TCP, Modbus RTU, Modbus UDP



<圖 6.2.3>

Win-GRAF 提供 Modbus slave 功能塊，透過處理從 master 接收到的請求來處理主機和 slave data block之間的通信。如果 master 請求 slave data block 的內容，那 slave 功能方塊會從 data block 讀取資料並寫入回應資料給 master。提供支援乙太網和串列通信的功能方塊。(表 6.2).

Modbus Slave 功能方塊	
乙太網	串列 (RS-232, RS-485)
MBSLAVETCP 	MBSLAVERTU 
MBSLAVETCPEX 	MBSLAVERTUEX 
MBSLAVEUDP 	MBSLAVERTUEXD 
MBSLAVEUDPEX 	

<表 6.2: 用於設定 Modbus 協議類型的功能塊 >

7 備援(冗餘)系統

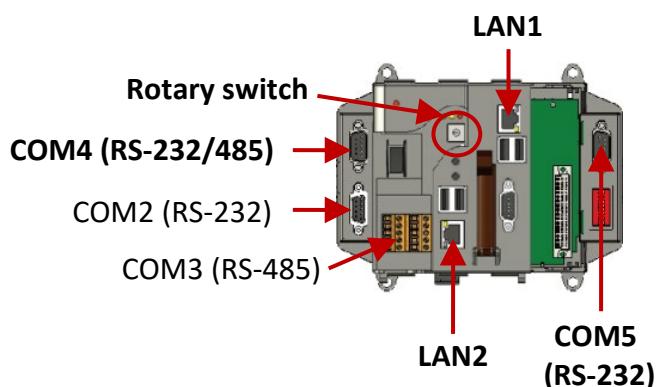
在工程中，備援(冗餘)是系統關鍵組件或功能的複製，目的是提高系統的可靠性，通常以備份或故障安全的形式，或提高實際系統性能。備援(冗餘)有時會產生更少的結果，而不是更高的可靠性-它會創建一個更複雜的系統，容易出現各種問題，它可能導致人為疏忽職責，並可能導致更高的生產需求，這可能會導致系統壓力過大而降低安全性。

ICP DAS 的 Win-GRAF PAC - XP-8xx8-CE6 系列有支援備援 (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 PLC 程式判斷於特定條件下時進行切換。

名稱定義	備援(冗餘)設定
Main-PAC	旋轉開關設為 2
Backup-PAC	旋轉開關設為 4
Active-PAC	正在運行程式的 PAC
Passive-PAC	正在備援資料的 PAC

備註: Main-PAC 一開始是 Active-PAC，它的控制權會根據情況自動切換到其他 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	備援(冗餘)機制 (旋轉開關: 2 & 4)
A. Public IP Port	LAN1: 用來跟 Win-GRAF, SCADA, 或 HMI 進行通訊。
B. Replication Port	LAN2: 資料同步通道。 兩個 PAC 使用乙太網交叉電纜連接，以實現高速資料同步。

C. Alive Port (or Heart-beat Port)	COM5 (RS-232): 用來偵測兩台 PAC 是否正常運作。
---	--

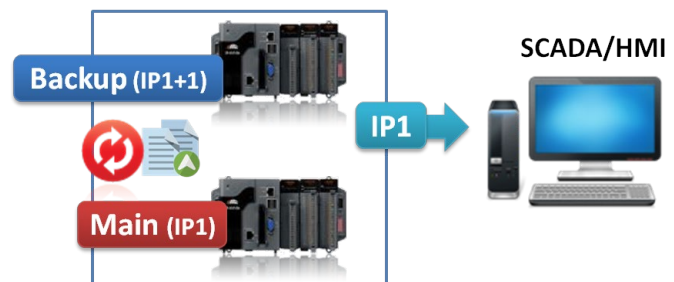
Win-GRAF 備援系統的特色

1. 安全性佳:

備援系統主要由 LAN1、LAN2 與 Alive Port 進行通訊，Active-PAC 只要有一條 Cable 通訊正常，就可繼續運作程式。若 3 條 Cable 皆中斷通訊，則觸發切換機制，由另一台 PAC 進行系統控制。

2. 單一的 Public IP:

Win-GRAF 備援系統提供單一的對外 IP 位址給 SCADA /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、VB.net、C# 應用程式或 eLogger HMI 專案，則需在上線運作前，把這些檔案安裝到備品 XPAC (或修好的 XPAC) 內。

4. 簡化程式設計:

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

5. 自訂安全機制:

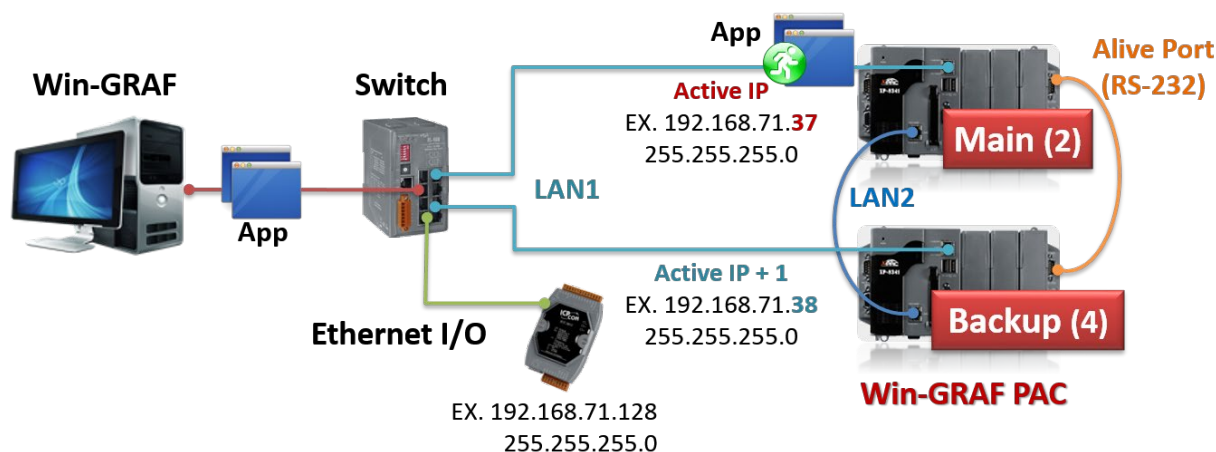
User 可在程式內制定安全機制，例如: 若主控 PAC 的 LAN1 斷線，而無法連上 SCADA，或是某個 RS-485 Port 斷線/故障時，將自動重開機並切換控制權到另一台 PAC。

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。

7.1 硬體連接

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. 使用者寫的 Win-GRAF 專案。
2. 變數的數值。
3. 功能方塊的樣例變數 (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 File (例如: \system_disk 或 \Micro_SD 內的 File 或其它非屬於 Win-GRAF 專案的檔案，如 C、應用程式檔。這些並不會自動備援。因此要安裝新的 備品 PAC 或修好的 PAC 進去備援系統之前，需預先把這些 File 都安裝好)。
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 memory 資料不會自動備援。

硬體		
2 台 XPAC	旋轉開關 Rotary Switch	一台為 "2" (稱為 Main-PAC) 一台為 "4" (稱為 Backup-PAC)
	LAN1	2 x 一般網路線
	LAN2	1 x Ethernet 跳線
	COM5 (RS-232)	1 x RS-232 跳線

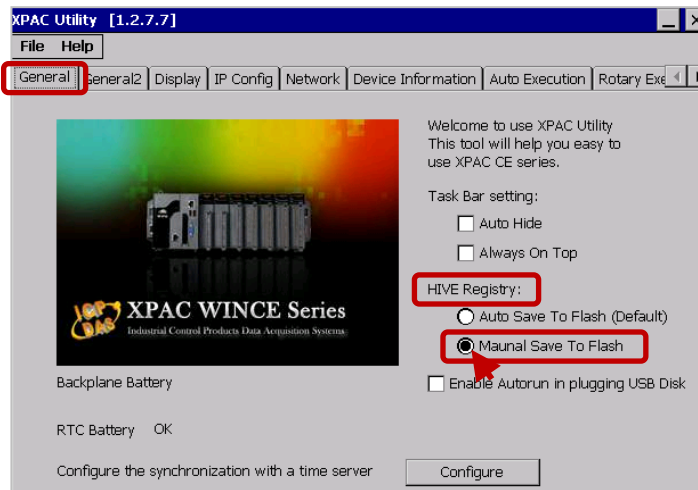
7.2 軟體設定

7.2.1 XPAC Utility

設定 LAN1 and LAN2 IP 位址：

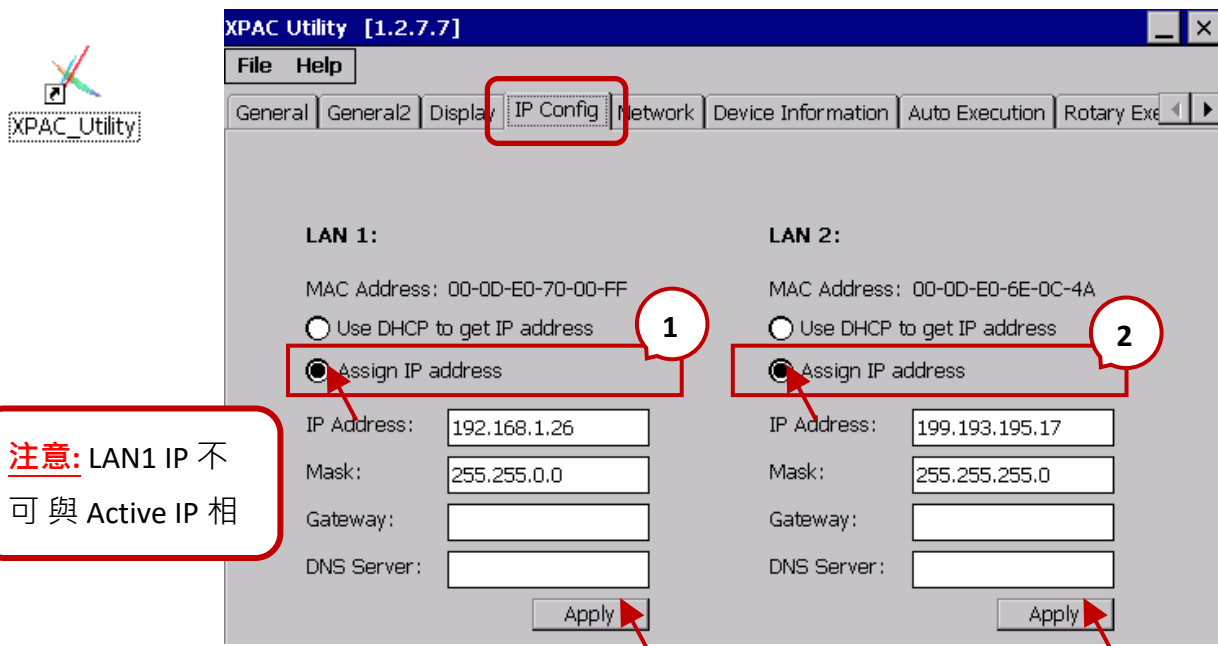
- ★ PAC 的 LAN1、LAN2 出廠預設為 DHCP，需使用 XPAC_Utility 設定為 "Assign IP address" 模式。
兩台 PAC 的 Rotary Switch 都轉到 0 並開機。

打開 XPAC Utility，於 General 頁籤，點選 HIVE Registry 的 Manual Save To Flash 選項。



點選 IP Config 頁籤，設定各自的 LAN1 IP。

- 1) LAN1 IP 不可與備援程式內的 Active_IP 相同，避免 PAC 啟動後發生 IP 衝突。
- 2) PC (Win-GRAF) 的 IP 位址，需和 PAC 的 LAN1 IP 在相同網段，才可進行連線。



設定 PAC 各自的 LAN2 為 Assign IP address 模式。

再點選 "Apply" 按鈕，保存設定。

設置通訊線 與 備援模式 (旋轉開關: 2 & 4)

- 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 COM5 (RS-232) 以跳線互接。
- D. 將兩台 PAC 的旋轉開關轉到 "2" (稱為 Main-PAC) 與"4" (稱為 Backup-PAC).
- E. 將兩台 PAC 重新開機。

開機後，LAN2 IP 與 Mask 會自動設定為:

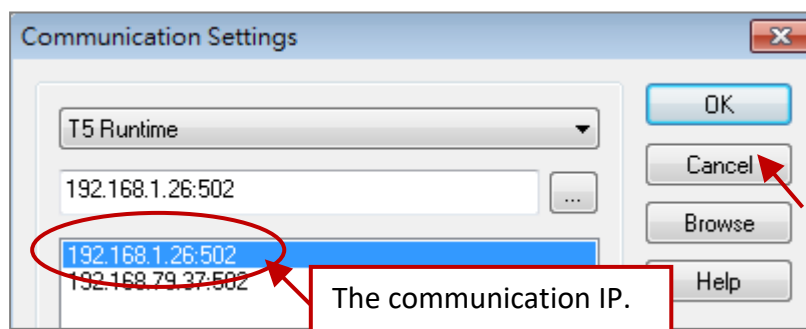
Main-PAC (2) : 199.193.195.17 / 255.255.255.0 ;

Backup-PAC (4) : 199.193.195.9 / 255.255.255.0

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 Main-PAC (2) , 目前的 LAN1 IP 。

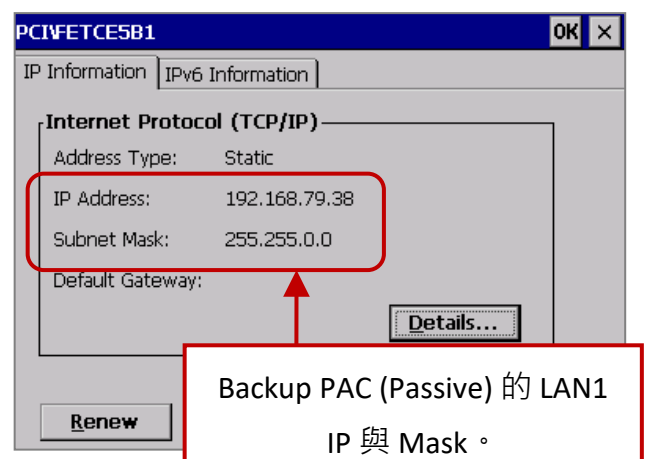
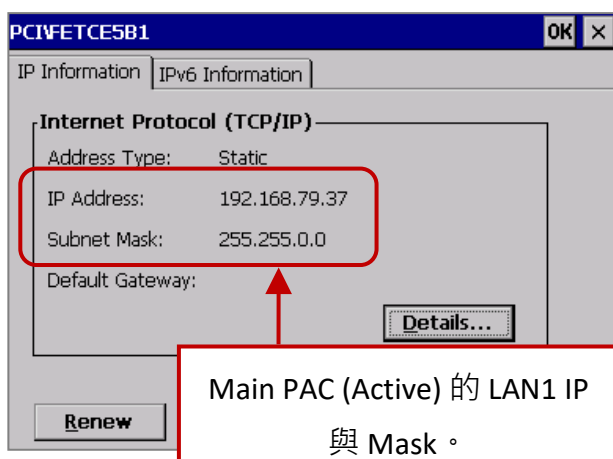


2. 設定 Active_IP 位址

您可依照網路環境來設定 I/O Boards – i_redundancy 的 Active_IP 與 Mask 。

3. 下載 Win-GRAF 專案

點選 “On Line” 按鈕 () 連線，再將備援程式下載到 Main PAC 中。下載後，Main PAC (2) 的 LAN1 會自動設成 Active_IP 。此時，Backup PAC (4) 的 LAN1 IP 會設成 Active IP + 1 。

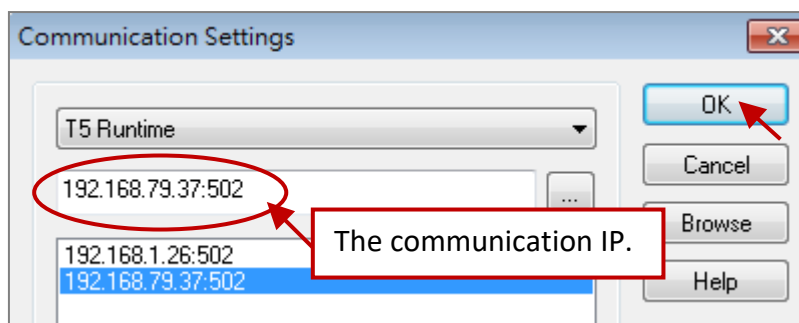


4. 將 Win-GRAF 通訊 IP，修改為 Active_IP

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，來停止 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。



備註： Win-GRAF Driver v1.10 版起，可查看 PAC 上的 L1 LED 燈，知道哪一台是主控 PAC。

L1 指示燈	
恆亮	表示此台是 Active PAC。
閃爍	表示此台是 Passive PAC。



7.2.2 Win-GRAF 備援設定 (LAN1/LAN2)

1. 查看 PACs 的備援狀態

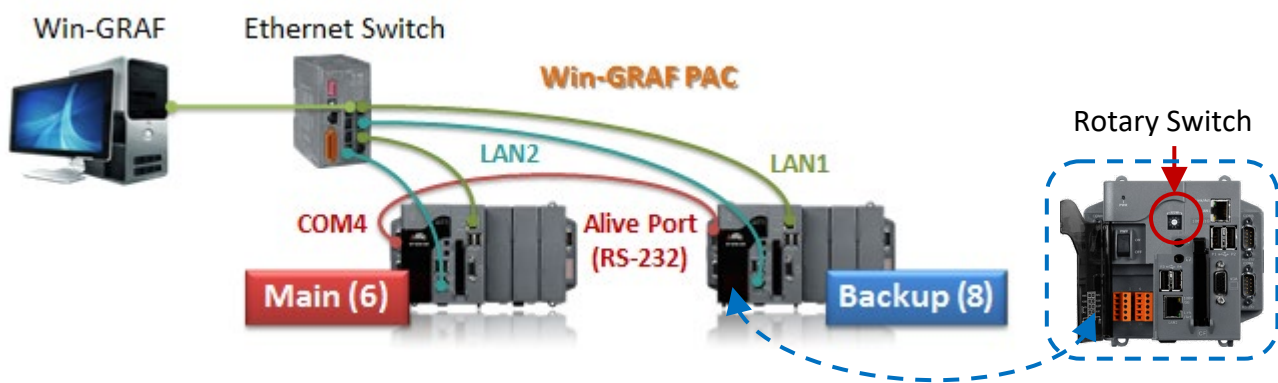
7.3 切換機制

硬體

Two XPAC	Rotary Switch	一台為 "2" (called Main-PAC) 一台為 "4" (called Backup-PAC)
	LAN1	2 x 一般網路線
	LAN2	2 x 一般網路線
	COM5 (RS-232)	1 x RS-232 跳線

7.3.1 Main PAC 和 Backup PAC 切換

7.3.2 正常運作同步狀態



步驟 1: 兩台 PAC 以一般模式開機 (旋轉開關 = 0)，並設定 LAN1、LAN2 IP 位址。

備註: 為了避免重新啟動 PAC 後的 IP 位址衝突，必須將 LAN1 與 Win-GRAF Active_IP 設定不同的 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 = 2 & 4)。

步驟 3: 修改備援程式的通訊 IP 與 Active IP，再下載到 Main PAC (2)。

下載成功後，專案會透過 LAN2 自動備份到 Backup PAC (4)。

備註: 更新 Active-PAC 的程式之前，關閉 Passive-PAC 電源，完成更新後，請在 Passive-PAC 上供電，以確保備援系統正常運作。

7.3.3 同步概念

1. Switch_Time: 上次同步。切換時間範圍。不是從 Active 到 Passive 的時間。
2. Switch_Cycle: 切換後的第一個週期
3. Lost Exchange: 總是

7.4 功能方塊

7.4.1 GetActiveRTStat

GetActiveRTStat		
此函數顯示 active runtime 的狀態。		
名稱	資料型態	說明
Output		
qRed	BOOL	如果可能存在備援，則此參數為 TRUE。如果運行時不支援備援，則為 FALSE。
qPassHere	BOOL	此參數表示 passive runtime 是否有連接到正在運行的 active runtime。
qPassOK	BOOL	此參數表示連接的 passive runtime 是否已完成初始化操作，並且現在已同步進行資料複製。
nLostExch	DINT	此參數表示超時複製交換失敗的次數。
SwitchCycle	BOOL	如果運行的 runtime 有切換為 active，並且現在正在第一個週期，此參數為 TRUE。
SwitchTime	TIME	This parameter gives an indication of the last switch duration.

7.4.2 GetActiveExStat

GetActiveExStat		
此函數顯示 active runtime 的附加狀態。		
名稱	資料型態	說明
Output		
PAC_Mode	INT	此參數顯示 PAC 模式。 0: 初始化 PAC; 1: Main PAC 備援; 2: Backup PAC 備援
Sync_LAN	INT	此參數表示目前同步的 LAN port。1: LAN1; 2: LAN2.
LAN1_State	BOOL	如果 LAN1 已連接，參數為 TRUE。 如果 LAN1 已斷線，參數則為 FALSE。
LAN2_State	BOOL	如果 LAN2 已連接，參數為 TRUE。 如果 LAN2 已斷線，參數則為 FALSE。

7.4.3 ActiveRTSwitch

ActiveRTSwitch		
用於切換 active PAC 與 passive PAC 的函數。		
名稱	資料型態	說明
Input		
bEnable	BOOL	設定此參數為 TRUE 來開啟 active 與 passive PAC 切換機制。
Output		
bOK	BOOL	如果切換成功，此參數為 TRUE。

7.4.4 ActiveSYNSwitch

ActiveSYNSwitch		
用於切換同步 LAN port 的函數。		
名稱	資料型態	說明
Input		
bEnable	BOOL	設定此參數為 TRUE 來開啟同步切換機制。
Output		
bOK	BOOL	如果切換成功，此參數為 TRUE。

8 OPC UA Server

OPC 統一架構 (UA) 是由 OPC 基金會建立的開放標準，定義了一個獨立於平台的互操作性標準。OPC UA 提供了一種安全的客戶端到 Server 連接方法，並且能夠通過防火牆和 VPN 連接進行安全連接。對於大多數用戶應用程序，UA 標準最相關的組件如下：

- 透過客戶端和 Server 端點的可信證書保護連接。
- 強大的項目訂閱模型，可在客戶端和 Server 之間提供高效的資訊更新。
- 從參與的 UA Server 發現可用資料的增強方法。

本手冊旨在介紹 Win-GRAF OPC UA Server 支援的主要功能和設定。此外，還給出了設定和測試程，以使您熟悉特定設定的特性、功能、限制和操作特性。

8.1 OPC UA Server 設定

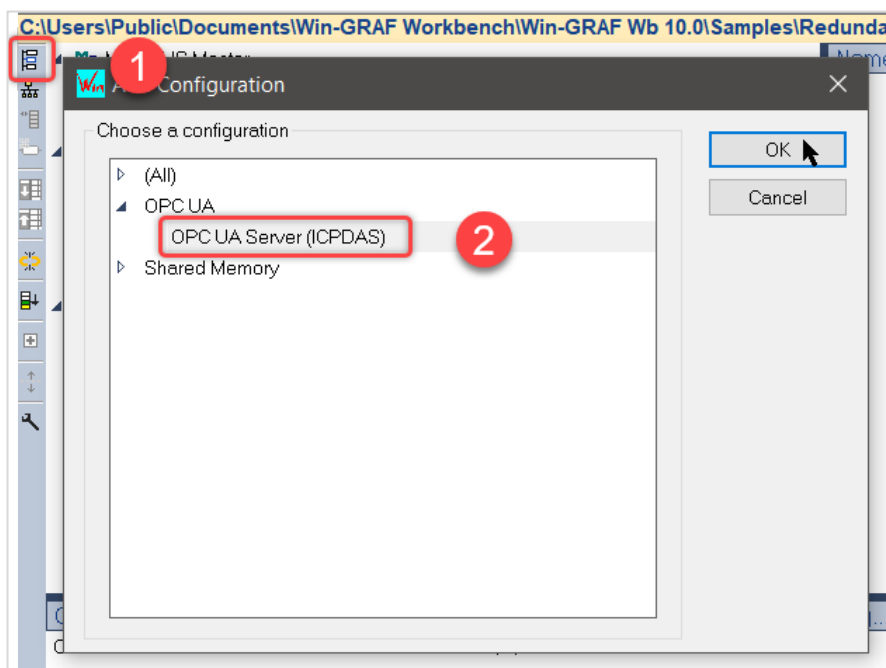
您應該熟悉 OPC UA 規範以及用於 OPC UA Server 和客戶端之間資料交換的通訊方法。如果安全性在您的應用程式中起著至關重要的作用，則需要更深入地了解 Server 和客戶端如何使用 OPC UA 證書來安全地識別並相互通信。

本章簡要說明使用 Win-GRAF 工作台配定 OPC UA Server 的過程，並描述了支援的設定參數。

8.1.1 Server 設定程序

本節簡要概述瞭如何使用 Win-GRAF 工作台加入和設定 OPC UA Server。下一章將更詳細的說明 Server 參數。

- **步驟 1:** 開啟 Win-GRAF 工作台並建立一個新項目
- **步驟 2:** 透過單擊工具欄中的 'Fieldbus Configuration' 按鈕或雙擊工作區中的 'Fieldbus Configuration' 節點，來打開 Fieldbus Configurations 視窗。
- **步驟 3:** 打開 'OPC UA Server' 插入: 點擊左邊工具欄的 'Insert Configuration' 按鈕，並從 'Add



Configuration' 對話框選擇 'OPC UA Server (ICPDAS)'。

- **步驟 4:** 基本配置參數設定:

- 單擊"OPC UA Server (ICPDAS)" 節點顯示基本配置參數的預設值。雙擊參數進入 'Value' 欄修改設定。

		Name	Value
MODBUS Master		Server Name	UaServer:[NodeName]
MODBUS Slave		Server URI	urn:Company:Product:OPCUA_Server
OPC UA Server (ICPDAS)		Product Name	Product Name
Remote I/O (ICP DAS)		Product URI	urn:Company:Product
COM3 : 115200,N,8,1		Manufacturer Name	Manufacturer Name
		Max Session	2
		Max Session Timeout	3600
		Min Session Timeout	10
		Min Keep Alive Interval	5
		Max Publish Interval	0
		Min Publish Interval	50
		Max Subscription Lifetime	0
		Min Subscription Lifetime	10000
		Use Username and Password	True
		Use Anonymous	False
		Use Certificate	False

-雙擊“OPC UA Server (ICPDAS)”節點 (左側) 打開一個單獨的對話框，在基本參數旁邊允許您進行其他設定 (例如:使用者身份識別、使用者登錄帳戶)。

OPC UA Server

Server Account Certificate

Information

Manufacturer Name: Manufacturer Name

Product Name: Product Name

Product Uri: urn:Company:Product

Server Name: UaServer:[NodeName]

Server Uri: urn:Company:Product:OPCUA_Server

Server Settings

Maximum Sessions: 2

Maximum Session Timeout: 3600 Second

Minimum Session Timeout: 10 Second

Minimum Keep Alive Interval: 5 Second

Subscription Settings

Maximum Lifetime: 0 Millisecond

Minimum Lifetime: 10000 Millisecond

Publishing Settings

Maximum Interval: 0 Millisecond

Minimum Interval: 50 Millisecond

User Identity Token

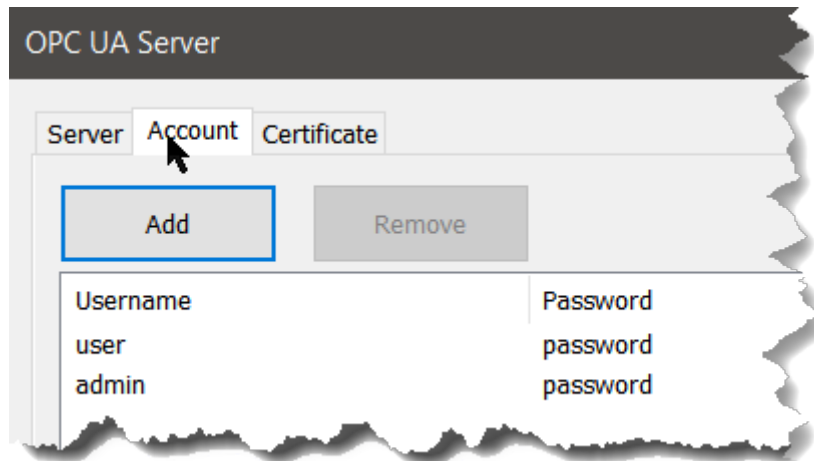
☒ Use Username and Password

☐ Use Certificate

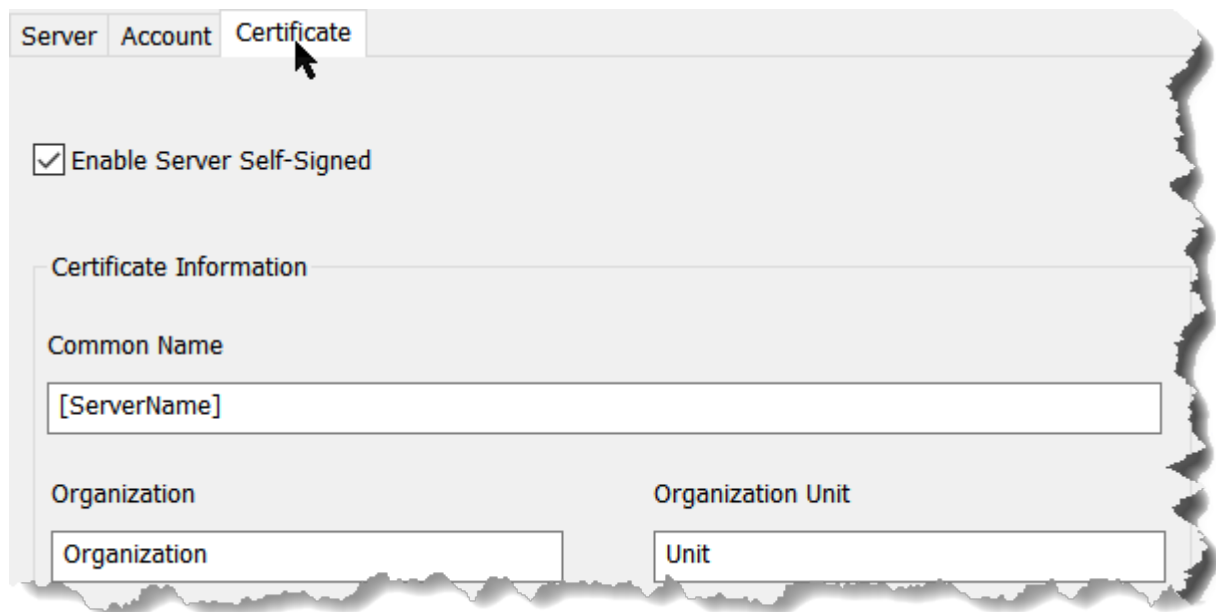
☐ Use Anonymous

OK Cancel

- 如果選擇了“使用使用者名稱和密碼”，請將用使用者名稱及密碼加入到“Account”頁籤的帳戶列表中。



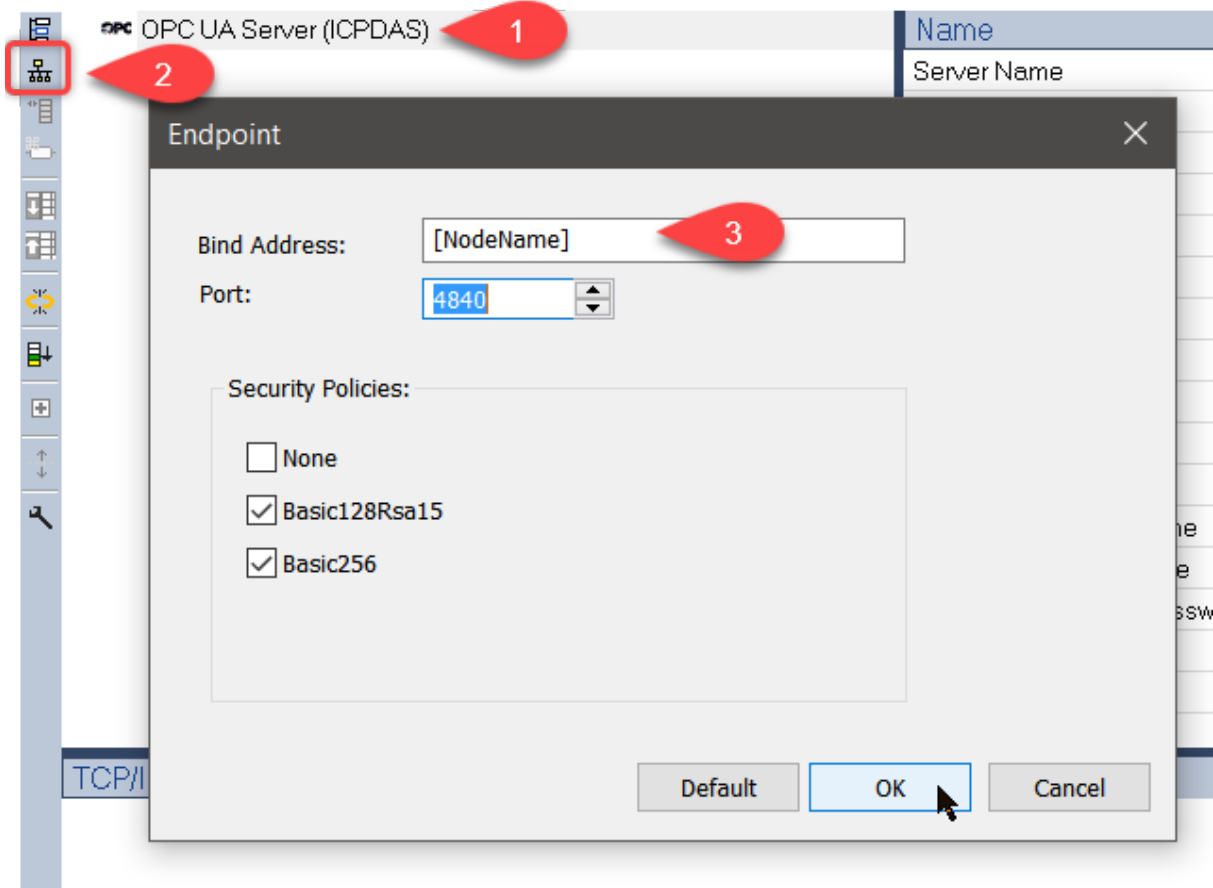
- 在 'Certificate' 頁籤確保 'Enable Server Self-Signed' 選項已啟用，其他詳細訊息如下所示。Server 將使用此資訊產生帶有其私鑰的證書並將其存儲在目錄 'pkiserver\own\certs\' 和 'pkiserver\own\private\'。僅當目錄中不存在帶有其密鑰的 Server 證書時才需要選擇 'Enable Server Self-Signed' 選項，因為沒有證書服務器無法啟動。



- **步驟 5: 端點設定:**

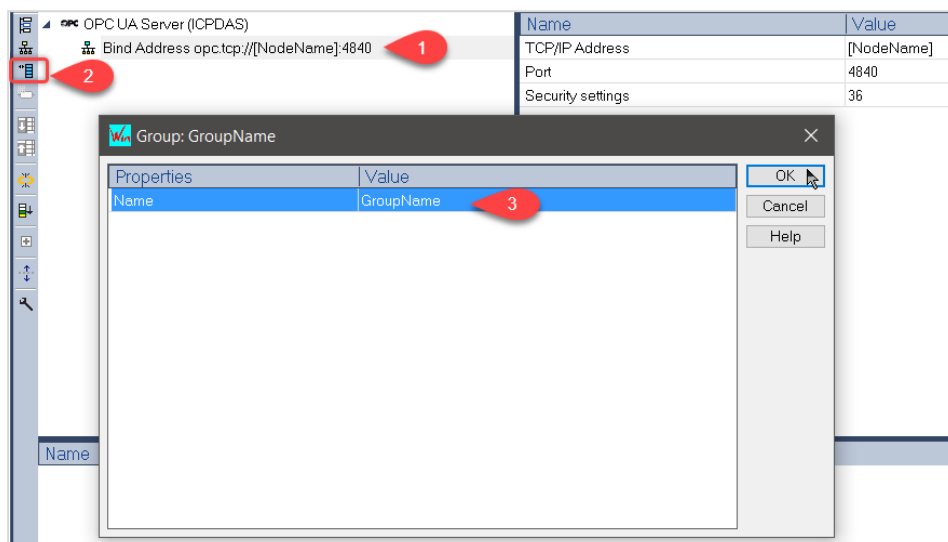
- 加入終端節點到 Server。

1. 透過雙擊節點，來啟動 'OPC UA Server (ICPDAS)' 節點。
2. 單擊左邊工具欄的 'Insert Master/Port' 按鈕，會跳出一個“Endpoint”對話框。
3. 設定端點參數。Win-GRAF Server只支援一個端點。用“OK”確認設定

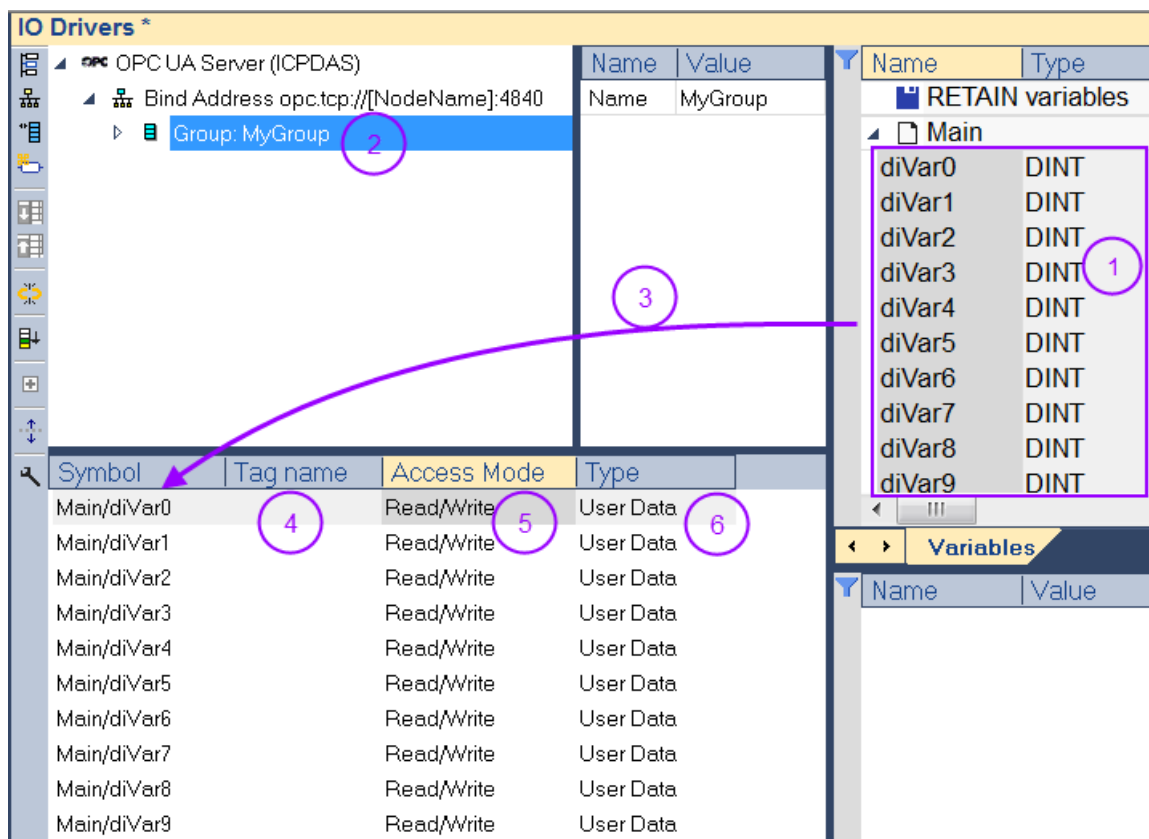


- **步驟 6: 建立監控群組節點:**

1. 建立終端節點 ('綁定 IP 位址: *opc.tcp://...*')
 2. 單擊左邊工具欄的 'Insert Slave/Data Block' 命令。將會顯示 'Group' 對話框。
 3. 通過雙擊“值”欄位，在對話框中輸入群組名稱。點擊“確定”
- 具有群組名稱的新節點將加入到樹視圖中。



- **步驟 7:** 將 PLC 變數分配給群組。此處將 PLC 變數分配給 UA Server 客戶端的存取。
 1. 宣告客戶端存取的變數
 2. 單擊選擇群組節點
 3. 將變數拖曳到對應區
 4. 為每個變數輸入一個“Tag name”。Tag name 將顯示給客戶端。如果 Tag name 為空白，則“Symbol”名稱將用於節點名稱。
 5. 選擇客戶端存取模式：“唯讀”、“唯寫”或“讀/寫”、“無法存取”。如果選擇了“無法存取”，則該變數表示 OPC UA Server 狀態。要顯示的狀態類型必須在“Type”欄位中設定。
 6. 在“Type”欄位中，設定了由變數表示的 OPC UA Serve 狀態類型。只有當選擇了“無法存取”時，“Type”欄位才有效。可用的 Server 狀態類型：
 - - Server 狀態
 - - 使用過的對話
 - - 使用過的訂閱
 - - 使用過的監控項目



- **步驟 8:** 建立程式，下載到運行時並啟動應用程式。
如果運行時執行文件目錄的 '`.\pkiserver\own\certs`' 文件夾中不存在證書，將會自動建立一個帶有私鑰的新證書。
- **步驟 9:** 使用 OPC UA 客戶端連接到 server。

8.2 Server 設定參數

OPC UA Server 模組的通訊設定，將決定server如何網路上的顯示方式，以及OPC UA 客戶端如何與它聯繫

8.2.1 Server 資訊

Information	
Manufacturer Name	Manufacturer Name
Product Name	Product Name
Product Uri	urn:Company:Product
Server Name	UaServer:[NodeName]
Server Uri	urn:Company:Product:OPCUA_Server

Server 設定	說明
Manufacturer Name	製造商的名稱。此名稱將用於 PLC 應用程式建立資訊，並顯示在追蹤紀錄檔中。
Product Name	- 應用程式的名稱。與製造商名稱相同，它被用作識別建立程式和追蹤紀錄檔的內部文件。 - 在追蹤紀錄檔中顯示為標題。
Product Uri	- server 所屬產品的全球統一資源標識符。 - 預設值名稱: 'urn:Company:Product' - 支援以下關鍵字串: '[NodeName]' -取得實際電腦的主機名稱。 '[ServerName]' -用 server 名稱替換關鍵字串。 - 範例: 'urn:MyCompany:MyProduct' 'urn:[NodeName]:MyProduct'
Server Name	- Server 的名稱。預設值的通用名稱是“Server”。 - 此名稱將儲存在關鍵字“[ServerName]”下，這意味著包含該關鍵字串由 Server 修改，方法是將關鍵字替換為 Server 名稱。例如，預設值情況下，關鍵字“[ServerName]”被分配給自簽名證書的“Common Name”，這意味著 Server 名稱將用作“Common Name”。 - 使用關鍵字串 '[NodeName]' 將自動取得實際電腦的主機名稱。 - 範例:如果電腦名為 'DESKTOP123' 字串 'MyName@[NodeName]' will generate the common name 'MyName@DESKTOP123'
Server Uri	每個 Server 都應有一個稱為 ServerUri 的全球統一資源標識符。

- 如果選擇了 UA Server，將顯示 Server URI。Server URI 可用作過濾發現結果的參考。
- 預設值名為 'urn:Company:Product:OPCUA_Server'
- 支援以下關鍵字串：
 - '[NodeName]' -取得實際電腦的主機名稱。
 - '[ServerName]' -用 server 名稱替換關鍵字串。
- 範例:
 - 'urn:MyCompany:MyProduct:MyServer'
 - 'urn:[NodeName]:MyProduct:[ServerName]'

8.2.2 對話設定

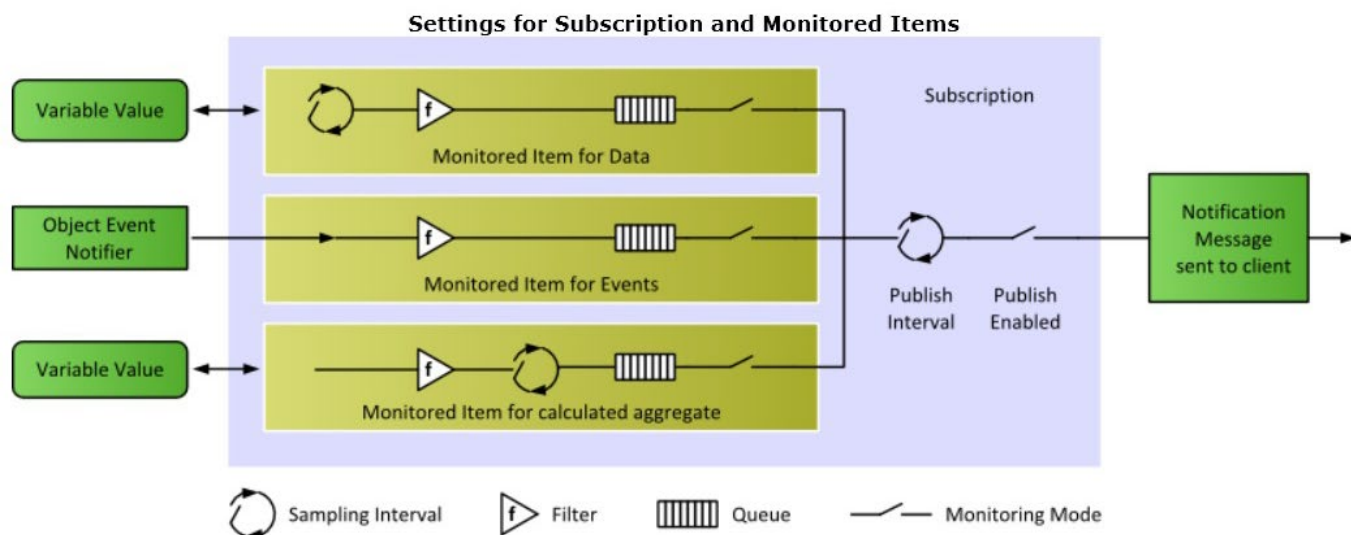
Server Settings	
Maximum Sessions	2
Maximum Session Timeout	3600 Second
Minimum Session Timeout	10 Second
Minimum Keep Alive Interval	5 Second

Server 設定	說明
Maximum Sessions	此參數指定支援的連接限制。有效的範圍是 1 到 128。預設值: 2。
Session Timeout	• 此參數指定 UA 客戶端建立對話的超時限制。值可能會根據應用程式的需要而改變。預設值為 10 到 3600。 • 最小值：此參數指定 UA 客戶端的最小超時限制。預設值: 10 秒。 • 最大值：此參數指定 UA 客戶端的最大超時限制。預設值: 3600 秒。
Keep Alive Interval	• 訂閱一個保留計數器，用於計算沒有通知報告給客戶的時間。當達到保留超時時，保留訊息通知客戶端訂閱仍然存在。 • 如果沒有要報告的內容（例如，沒有值更改），server 將向客戶端發送保留通知，這是一個空白的發佈，表示 server 仍然存取。

8.2.3 Server 訂閱設定

OPC UA 支持輪詢和訂閱模式。

- 在輪詢模式下，客戶端定義的時間間隔不斷更新對象。這會產生更高的通訊負載，因此僅建議用於少數符號。
- 在訂閱模式下，server 監控選定數量的節點。只有當節點值更改時，server 才會通知客戶端更改。這種機制極大地減少了傳輸的數據量，因此是從 OPC UA Server 讀取資訊的推薦模式。



訂閱設定說明如下。

Subscription Settings	
Maximum Lifetime	10000 Millisecond
Minimum Lifetime	0 Millisecond

Publishing Settings	
Maximum Interval	1000 Millisecond
Minimum Interval	0 Millisecond

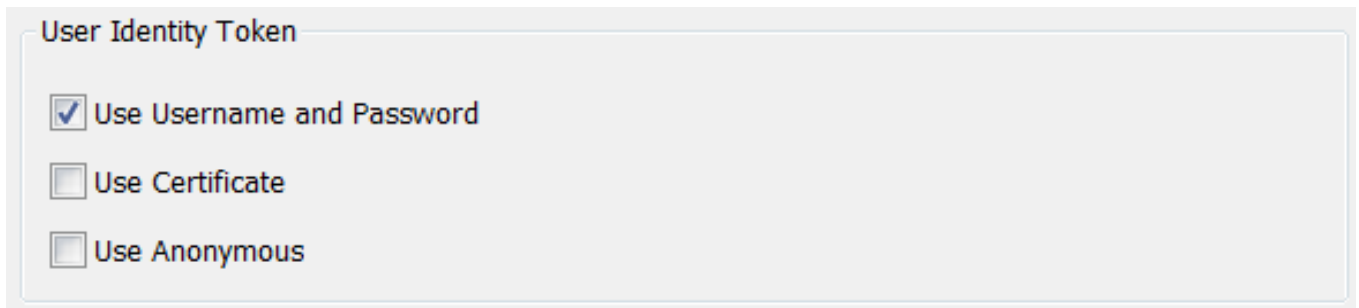
Server 設定	說明
Lifetime	- 間隔，server向客戶端發佈數據的時間間隔。 - Server將修改“最小間隔”間隔，以保護源免受可能由高採樣率引起的大量負載。 - 最大值：Server允許最長的訂閱生命週期（以毫秒為單位）。（預設值：10000） - 最小值：Server允許最短的訂閱生命週期（以毫秒為單位）；0沒有限制。（預設值：0）
Interval	在server上採樣（和儲存）數據的時間間隔，並在每個發佈時間間隔內發送。採樣間隔定義為server檢查受監控項目的數據更改的時間間隔。該時間間隔可以設定為比通知客戶端的时间間隔（Lifetime）快，在這種情況下，server可以對採樣數據進行排隊並發佈完整的順序。

8.3 用戶認證設定

當用戶嘗試從 OPC UA 客戶端連接到 OPC UA Server時，OPC UA Server 需要確認使用者的身份，然後才能允許來自 OPC UA 客戶端的連接。Win-GRAF server 目前支援在對話啟動期間對用戶進行身份驗證的三種不同方式

- 匿名身份識別
- 用戶名稱身份識別
- X509 身份識別

'User Identity Token' 區域確定允許登錄 OPC UA Server 的客戶端的身分類型。



User Identity Token

☒ Use Username and Password

☐ Use Certificate

☐ Use Anonymous

身分識別選項	說明
Use Username and Password	用戶名稱識別 • 要求客戶輸入與項目安全系統符合的使用者名稱和密碼。此選項僅在安全系統已啟用時才可使用。 • 如果選擇了這個選項，需要輸入使用者名稱和密碼。“Account”頁籤（圖 6）提供了一個列表，用戶可以在這輸入密碼。 • 預設值：啟用
Use Certificate	X509 身份識別 • 不需使用使用者名稱和密碼登錄 server，每個用戶都使用單獨的 X509 證書。 • X509 證書的通用名稱為用戶名稱，其私鑰為密碼。 • 預設值：禁用
Use Anonymous	• 允許客戶端不需輸入使用者名稱或密碼。 • 預設值：禁用

備註:

- 至少必須設定一個可用選項。如果沒有選擇任何身份選項，則客戶端將無法連接到 server。
- 選項不是唯一的；可以選擇多個選項。

下面將說明三個基本應用程式的安全設定。

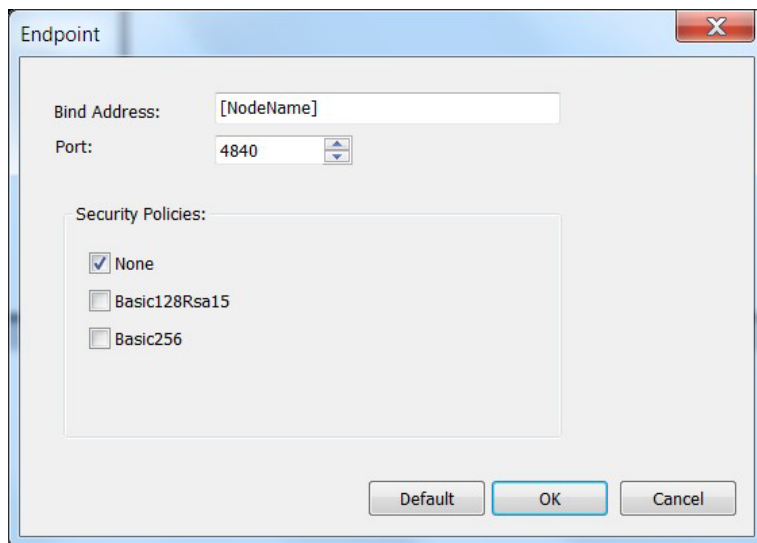
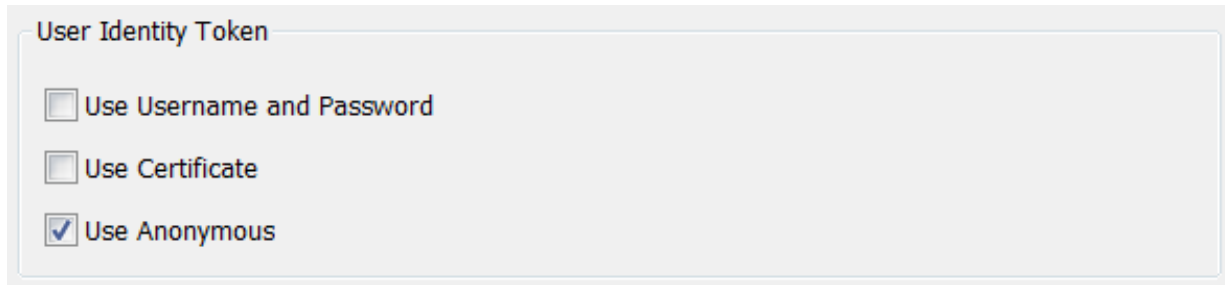
8.3.1 匿名身份識別

匿名身份識別不執行任何身份驗證（圖 3）。意思是

- 每個將證書加入到“pkiserver/trusted/certs”目錄的客戶端都可以與 server 建立連接，不須輸入使用者名和密碼。

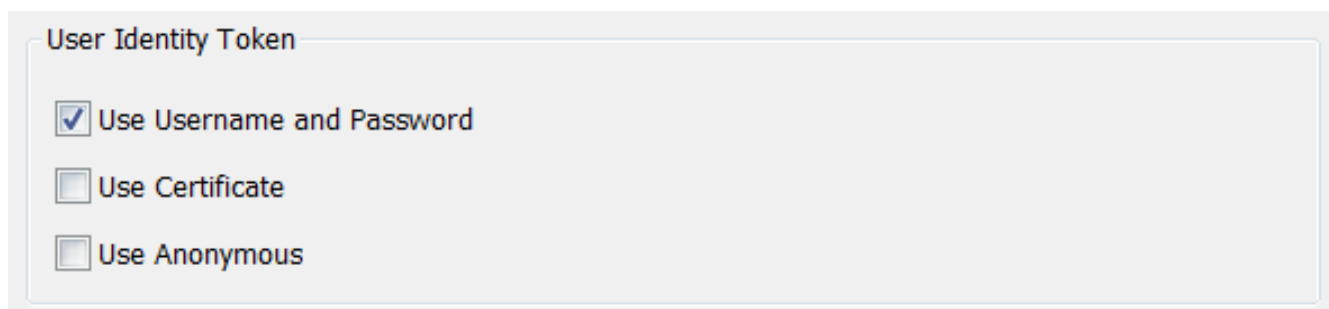
- 如果 server 的“Security Policy”設定為“None”（圖 4），所有客戶端無論是否受 server 信任，都可以建立與 server 的對話。

為了安全起見，建議使用匿名身份識別。



8.3.2 用戶名稱身分識別

這種識別類型設定（圖 5）允許使用使用者名稱和密碼對用戶進行身份驗證。密碼以加密形式傳遞給 UA Server，即使使用非安全通道也是如此。密碼永遠不會以純文本形式儲存在記憶體中，因此始終受到保護。



使用者帳號列表(圖 6) 允許你輸入多個使用者名稱與相應的密碼。此外，通過禁用“Write Authorization”複選框，可以將每個用戶的使用存取限制唯讀，預設帳號條目顯示在圖 6。