
ISaGRAF® 進階使用手冊

泓格科技股份有限公司

本書適用以下的控制器，期望對使用 ISaGRAF 作為軟體作業平台的系統整合人員、程式設計師以及系統維護人員能有所助益。

XP-8047-CE6 / XP-8046-CE6 / XP-8347-CE6 / XP-8346-CE6 / XP-8747-CE6 / XP-8746-CE6 ,
WP-8147 / WP-8146 / WP-8447 / WP-8446 / WP-8847 / WP-8846 ,
WP-8137 / WP-8136 / WP-8437 / WP-8436 / WP-8837 / WP-8836 ,
VP-25W7 / VP-25W6 / VP-23W7 / VP-23W6 ,
μPAC-7186EG , I-7188EG , I-7188XG , VP-2117 ,
iP-8447 / iP-8847 / iP-8417 / iP-8817 ,
I-8417 / I-8817 / I-8437-80 / I-8837-80

泓格科技股份有限公司感謝您購買本公司以 ISaGRAF 作為軟體核心的各類控制器，本系列產品結合方便整合的特性以及 IEC 61131-3: ISaGRAF 強大的發展性，適用於各類工業用控制系統

注意

泓格科技股份有限公司對於因為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，本公司並保留在任何時候修訂本書且不需通知的權利。

泓格科技股份有限公司將儘可能地提供本系列產品可靠而詳盡的資訊。然而，本公司並無義務需提供此系列產品詳盡的應用資訊，或對因非法使用本系列產品所遭受的損害負任何責任。

商標&著作權

本書提到的所有公司商標、商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

Data Sheet :

請上網 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 畫面右方 點選 Data Sheet 或 http://www.icpdas.com/products/PAC/i-8000/data%20sheet/data%20sheet_c.htm

FAQ :

請上網 www.icpdas.com - “FAQ” - “Software” - “ISaGRAF” 查看 FAQ (常被詢問的問題) 或 http://www.icpdas.com/faq/isagraf_c.htm

Copyright Sep 2011, by ICP DAS CO., LTD. All Rights Reserved

目 錄

ISAGRAF 進階使用手冊	1
注意	1
商標&著作權	1
目 錄	2
相關參考資料.....	7
ISAGRAF 控制器性能比較表一	8
ISAGRAF 控制器性能比較表二	9
XPAC, WinPAC 系列 與 iPAC:.....	9
ViewPAC 系列:.....	10
I-8000, iPAC 系列 與 WinPAC:.....	11
I-7188, μ PAC 系列:	12
如何選用 ISAGRAF PAC.....	15
第 1 章 安裝軟體及硬體.....	1-1
1.1: 安裝 ISaGRAF Ver.3 作業平台	1-1
1.1.1 保護鎖注意事項 (Dongle 與 USB Key-Pro).....	1-2
1.1.2 Windows 2000 使用者的注意事項 :.....	1-3
1.1.3 Windows NT 使用者的注意事項 :.....	1-4
1.1.4 Windows Vista 或 Windows 7 (32-bit) 使用者的注意事項.....	1-4
1.1.5 Windows 7 (64-bit) 使用者的注意事項	1-6
1.1.6 使用 Variable Arrays 變數陣列的注意事項.....	1-6
1.2: 安裝 ICP DAS Utilities For ISaGRAF.....	1-6
1.3: PAC 硬體相關設定	1-7
第 2 章 ISAGRAF 程式設計基礎	2-1
2.1: 一個簡單的階梯圖程式 (LD)	2-1
2.1.1: 程式撰寫	2-4
2.1.2: 連接輸入輸出(I/O)設備.....	2-26
2.1.3: 編譯 LD 範例程式	2-28
2.1.4: 模擬 LD 專案	2-29
2.1.5: 下載 及 除錯 LD 範例程式	2-31
2.2: 一個簡單的結構化文字 (ST) 程式.....	2-35
2.3: 簡單的功能方塊圖程式	2-43
2.3.1: 編寫 FBD 範例程式.....	2-43
2.3.2: 模擬 FBD 程式.....	2-49
2.4: 簡單的指令集程式	2-51
2.5: 簡單的順序式功能圖程式(SFC).....	2-54
2.5.1: 撰寫 SFC 範例程式.....	2-56
2.5.2: 編輯 SFC 程式	2-59
2.5.3: 模擬 SFC 程式	2-64
2.6: 使用 變數陣列	2-65
2.6.1 指定 網路位址號碼 給變數陣列.....	2-70
2.6.2 將 變數陣列 使用成 可保留變數.....	2-71
第 3 章 建立 I/O 連結	3-1
3.1 如何將 I/O 控制模組引入 ISaGRAF 專案	3-3

3.1.1:	連結 I/O 板	3-4
3.1.2:	連結輸入及輸出變數	3-5
3.2:	連結類比型態的 I/O 板	3-7
3.2.1:	設定 類比 I/O 卡的 Range 參數 與 類比 IO 數值轉換函式	3-7
3.2.2:	設定 特別的 “range” 參數來直接取得 溫度卡片的 “攝氏度” 或 “華式度” 值	3-9
3.2.3:	I-87017ZW 板卡使用說明	3-11
3.2.4:	I-8017HW 板卡使用說明	3-14
3.2.5:	I-8084W 板卡使用說明	3-15
3.2.6:	I-87015W 與 I-87015PW 板卡使用說明	3-17
3.2.7:	I-87019ZW 板卡使用說明	3-18
3.2.8:	I-8024W 板卡使用說明	3-19
3.2.9:	I-87018ZW 板卡使用說明	3-20
3.3:	一些特殊虛卡的連結使用說明	3-21
3.3.1:	連結 Push4Key 與 Show3Led 虛卡	3-21
3.3.2:	連結 io_state 虛卡 來偵測 真實 IO 板卡的運作狀態	3-22
3.3.3:	連結 echo_tim 虛卡 來延遲 Modbus RTU Slave 串口一段時間後再回覆	3-23
3.3.4:	WP-8xx7, VP-25W7 與 XP-8xx7 連結 RTU_Slav 虛卡來使用更多擴充的 Modbus RTU Slave 串口	3-24
3.3.5:	使用 dis_stop 虛卡 來 關閉 / 開啟 ISaGRAF Download 程式功能	3-25
3.4:	直接表示變數	3-26
3.5:	內建於 I-87xxx 及 I-7000 D/I 模組 的計數值	3-29
3.6:	自動搜尋及宣告 : Auto-Scan I/O	3-31
3.7:	PWM 輸出	3-33
3.8:	並列輸入卡的 Counter	3-37
3.10:	步進輸出	3-38
第 4 章	連接 I-8XX7 控制器與人機介面(HMI)程式	4-1
4.1:	宣告變數的網路位址	4-1
4.2:	使用 Modbus 讀寫 Words, Long Words 及 Float 資料	4-6
4.3:	將 I-8xx7 當成 Modbus I/O 或 Modbus TCP/IP I/O 來使用	4-8
4.4:	連結 ISaGRAF PAC 和 Touch 500 系列	4-13
4.4.1:	設計 ISaGRAF PAC 應用程式式	4-14
4.4.2:	設計 Touch 500 畫面	4-15
4.5:	經由 Modbus 通訊協定讀/寫 Word 及 Integer 陣列	4-35
第 5 章	MODBUS 協定	5-1
5.1:	Modbus 協定格式: RTU Serial	5-1
5.2:	Modbus Protocol Format: TCP/IP	5-6
5.3:	CRC-16 演算邏輯	5-7
第 6 章	連接 I-7000 和 I-87K 遠程 I/O 模塊	6-1
6.1:	設定 I-7000 及 I-87xx 模組	6-1
6.2:	開啟 “Bus7000b”	6-6
6.3:	編寫控制 I-7000 模組程式	6-8
6.3.1:	編寫 I_7xxx 及 I_87xxx 方塊程式	6-8
6.3.2:	直接取得 遠程 I-7000 及 I-87xxxW 溫度輸入 I/O 的溫度值	6-13
6.4:	Redundant Bus7000	6-15
第 7 章	控制器與控制器間的資料交換	7-1
7.1:	基本的 Fbus 規則	7-1
7.2:	設定 ISaGRAF PAC 為 Fbus master 或 Fbus slave	7-2
7.3:	編寫 Fbus 封包程式	7-6
7.4:	透過 Fbus 的資料傳輸範例	7-9
7.5:	編寫 Ebus 資料交換程式	7-14
7.5.1:	基本的 Ebus 規則	7-14
7.5.2:	設定 ISaGRAF PAC 為 Ebus Master 或 Slave	7-16
7.5.3:	編寫 Ebus 封包	7-18

第 8 章 連結 MODBUS RTU / ASCII 裝置	8-1
8.1: 設定 控制器 以連接 Modbus 裝置	8-2
8.2: 編寫 Modbus 方塊程式	8-4
8.3: 連接 M-7000 I/O 模塊	8-10
8.4: 連接 EKAN-Modview LED 顯示器	8-10
第 9 章 幾個 ISaGRAF 常用的功能介紹	9-1
9.1: 建立 ISaGRAF 的專案群組	9-2
9.2: 上載 ISaGRAF 專案	9-3
9.3: 設定 ISaGRAF 的密碼管制	9-6
9.4: 建立 ISaGRAF 程式日記	9-8
9.5: 程式的備份及回存	9-9
9.6: 專案及程式的複製	9-11
9.7: 為 ISaGRAF 專案增加註解	9-13
9.8: 設定所要連接的 controllers 的位址編號	9-14
9.9: 程式編譯的最佳化選項	9-15
9.10: ISaGRAF 的類比轉換表	9-16
9.11: 透過 Microsoft Excel 引用/匯出變數宣告	9-19
9.12: 執行期間顯示選定的變數資料	9-22
9.13: 如何在 ISaGRAF 內搜尋某一變數名稱 ?	9-25
第 10 章 可保存變數和資料儲存	10-1
10.1: 可保存變數	10-1
10.2: 備份資料到 EEPROM	10-6
10.3: 電池保持式 SRAM	10-8
10.3.1: 存取 SRAM	10-9
10.3.2: 從 SRAM 內 上載(抓出) 資料	10-9
10.3.3: 下載資料到 SRAM 內	10-11
10.3.4: 電池保持式 SRAM 的函式說明	10-13
10.4: 使用 I-8073 來儲存資料	10-13
10.5: 檔案存取	10-14
10.5.1: 範例 Wpdmo_51: 從檔案內讀取 10 個實數值, 共 10 行, 每行 1 個實數	10-15
10.5.2: 範例 Wpdmo_54: 從檔案內讀取 20 個實數值, 共 4 行, 每行 5 個實數	10-17
10.5.3: 範例 Wpdmo_55: 從檔案內讀取 20 個整數值, 共 2 行, 每行 10 個整數	10-19
10.5.4: 範例 Wpdmo_56: 將 1 ~ 255 個實數變數用 Flash Memory 來保存最終值	10-21
10.5.5: 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 RAM Disk 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖	10-31
10.6: 偵測 Controller Fault	10-32
第 11 章 ISaGRAF 程式範例與 FAQ	11-1
11.1: 安裝 ISaGRAF 程式範例	11-1
11.2: ISaGRAF 範例程式列表	11-3
11.3: 一些範例程式的詳細說明	11-16
11.3.0 : Demo_01A & Demo_03: 在某個時間點做某件事	11-16
11.3.1 Demo_02: 啟動, 暫停及重置 Timer	11-20
11.3.2 Demo_17: 讀/寫 EEPROM	11-22
11.3.3 Demo_29: 每 75 秒記錄 1200 筆 Short Int 並經由 Com3 傳送到 PC	11-24
11.3.4 Demo_33: 從 Com3 讀/寫 User 自行定義的通訊格式	11-29
11.3.5 Wdemo_24: 當 Alarm 1 ~ 8 發生時, 從 Com2 送出 Alarm 訊息	11-37
11.3.6: 範例 Whmi_12: 使用 Wincon-8xx7, 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 RAM Disk 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖	11-41
11.3.7: 範例 Demo_71: 使用 I-8437-80 或 I-8837-80, 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 S-256 / S-512 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖	11-54
11.3.8: 使用較準確的 固定週期 去 做某件事	11-62

11.3.9: Demo_72: I-7188EGD 連接 I-7018z 量測 6 個 4 到 20 mA 的電流輸入值 與 4 個 Thermo-Couple 溫度輸入值, 並顯示在 PC 的 VB 6.0 程式上.....	11-63
11.3.10: 範例 Whmi_13: 使用 Wincon-8xx7, 每 0.01 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 User 配置的一塊 RAM 記憶區內, 連續記錄 1~10 分鐘, 之後再儲存到 RAM Disk File 內, 最後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖.....	11-69
11.4: 常用的 FAQ.....	11-79
第 12 章 發送 EMAIL.....	12-1
第 13 章 透過 MODEM_LINK 遠程監視及下載程式.....	13-1
13.1: Introduction	13-1
13.2: 透過 Modem_Link 下載程式	13-2
第 14 章 SPOTLIGHT : 簡易圖控	14-1
14.1: Spotlight 範例:	14-1
第 15 章 編寫自行定義的 FUNCTION.....	15-1
15.1: 在 1 個專案內定義 Function	15-1
15.2: 在 ISaGRAF Library 內定義 Function	15-6
第 16 章 連接 MMICON.....	16-1
16.1: 硬體安裝.....	16-1
16.2: 建立 MMICON 的背景圖形	16-2
16.3: 設計 MMICON 的控制程式	16-2
第 17 章 SMS: 收發手機簡訊.....	17-1
17.1: 硬體安裝	17-1
17.2: SMS 範例程式	17-2
第 18 章 馬達運動控制.....	18-1
18.1: Install motion driver	18-1
18.2: Introduction	18-3
18.2.1: System Block Diagram	18-3
18.2.2: DDA Technology.....	18-3
18.3: Hardware	18-5
18.3.1: I-8000 hardware address	18-5
18.3.2: LED Indicator	18-6
18.3.3: Hardware Configuration	18-6
18.3.4: Pin assignment of connector CN2.....	18-9
18.4: Software.....	18-13
I/O connection:	18-13
Setting commands:.....	18-14
M_regist Register one I-8091	18-14
M_r_sys Reset all setting	18-15
M_s_var Set motion system parameters.....	18-16
M_s_dir Define output direction of axes	18-17
M_s_mode Set output mode.....	18-17
M_s_serv Set servo ON/OFF.....	18-18
M_s_nc Set N.O. / N.C.	18-18
Stop commands:.....	18-19
M_stpx Stop X axis	18-19
M_stpy Stop Y axis	18-19
M_stpall Stop X & Y axes	18-19
Simple motion commands:.....	18-20
M_lsporg Low speed move to ORG	18-20
M_hsporg High speed move to ORG	18-20
M_lsppmv Low speed pulse move	18-21
M_hsppmv High speed pulse move	18-21
M_nsppmv Normal speed pulse move	18-22
M_lspm Low speed move	18-22
M_hspm High speed move.....	18-23

M_cspmv	Change speed move.....	18-23
M_slwdn	Slow down to low speed.....	18-24
M_slwstp	Slow down to stop	18-24
Interpolation commands:.....		18-25
M_intp	Move a short distance on X-Y plane	18-25
M_intln	Move a long distance on X-Y plane.....	18-26
M_intln2	Move a long distance on X-Y plane.....	18-27
M_intcl2	Move a circle on X-Y plane	18-28
M_intar2	Move a arc on X-Y plane	18-29
M_intstp	Test X-Y plane moving command	18-30
I-8090 encorder commands:.....		18-31
M_r_enco	Reset I-8090's encorder value to 0.....	18-31
第 19 章 以太網通訊與安全防護		19-1
19.1:	網路安全防護	19-1
19.2:	使用 UDP 來傳遞或接收訊息 到 PC / HMI 或 設備.....	19-3
19.3:	使用 TCP / IP Client 功能來 傳送 / 接收 / 主動回報 訊息.....	19-5
第 20 章 備援系統 (冗餘系統).....		20-1
20.1:	XP-8xx7-CE6 備援系統	20-1
20.2:	WP-8xx7 備援系統.....	20-5
第 21 章 連接 M-7000 系列 MODBUS RTU I/O.....		21-1
21.1:	使用 DCON Utility 來對 M-7000 模塊 作 初始設定.....	21-2
21.2:	編寫操控 M-7000 模塊 的程式	21-6
第 22 章 連接 MODBUS TCP/IP I/O		22-1
22.1:	I-8KE8-MTCP I/O 簡介.....	22-1
22.2:	編寫程式操控 I-8KE8-MTCP I/O.....	22-4
第 23 章 連接 FRNET 快速 I/O.....		23-1
23.1:	FRnet I/O 簡介	23-3
23.2:	編寫 FRnet I/O 程式	23-5
23.3:	使用 FR_B_A 來降低程式的大小.....	23-8
第 24 章 使用 COM PORT 函式來讀/寫 RS-232/422/485.....		24-1
24.1:	主動對設備 一問一答 方式	24-2
24.2:	被動等待 設備傳資料過來 方式	24-3
24.3:	每隔一段時間 主動回報資料 方式	24-4
24.4:	有事件發生時 主動回報資料 方式	24-5

相關參考資料

本書提供中,英文版的電子檔,路徑如下:

English manual :

檔名為 “user_manual_i_8xx7.pdf“

iP-8xx7, I-8000 & μPAC-7186, I-7188 CD: \napdos\isagraf\8000\english_manu\

WP-8xx7 CD: \napdos\isagraf\wp-8xx7\english_manu\

XP-8xx7-CE6 CD: \napdos\isagraf\xp-8xx7-ce6\english_manu\

VP-2xW7 CD: \napdos\isagraf\vp-25w7-23w7\english_manu\

中文使用手冊 :

檔名為 “chinese_user_manual_i_8xx7.pdf“

iP-8xx7, I-8000 & μPAC-7186, I-7188 CD: \napdos\isagraf\8000\chinese_manu\

WP-8xx7 CD: \napdos\isagraf\wp-8xx7\chinese_manu\

XP-8xx7-CE6 CD: \napdos\isagraf\xp-8xx7-ce6\chinese_manu\

VP-2xW7 CD: \napdos\isagraf\vp-25w7-23w7\chinese_manu\

各 ISaGRAF PAC 的快速上手手冊 (使用手冊):

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 畫面右方的 Manual

網路更新資訊 :

ISaGRAF IO 函式庫、驅動程式以及手冊的更新訊息請參考

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 畫面右方的 Driver , Manual

技術服務 :

請您與當地的代理商聯絡 或 E-mail 至 service@icpdas.com

更多的產品訊息請參考 www.icpdas.com

FAQ:

請上網 www.icpdas.com - “FAQ” - “Software” - “ISaGRAF” 查看 FAQ (常被詢問問題)

或 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 畫面右方的 FAQ

或 <http://www.icpdas.com/faq/isagraf.htm>

ISaGRAF 控制器性能比較表一

PACs	CPU	相較於 I-8417		乙太網路	ISaGRAF 程式碼容量限制 (bytes)	可供程式運作的記憶體容量 (bytes)
		一般執行速度比	一般浮點運算速度比			
		(掃描時間)	(掃描時間)			
XP-8xx7-CE6	LX 800 500 MHz	約 10 ~ 50 倍 (3 ~ 15 ms)	約 10 ~ 50 倍 (3 ~ 15 ms)	2 ports 10/100 Mbps	2 MB	大約 200 ~ 400 MB
WP-8xx7	PXA270, 520 MHz 或相容	約 10 ~ 30 倍 (3 ~ 15 ms)	約 10 ~ 30 倍 (3 ~ 15 ms)	2 ports 10/100 Mbps	1 MB	大約 20 ~ 40 MB
W-8347 W-8747	Strong-ARM 206 MHz 或相容	大約	大約			
W-8337 W-8737		10 ~ 20 倍 (3 ~ 15 ms)	10 ~ 20 倍 (3 ~ 15 ms)			
VP-25W7 VP-23W7	PXA270, 520 MHz 或相容	約 10 ~ 30 倍 (3 ~ 15 ms)	約 10 ~ 30 倍 (3 ~ 15 ms)	1 port 10/100 Mbps	1 MB	大約 20 ~ 40 MB
VP-2117	80186, 80 MHz 或相容	約 4 倍 (2 ~ 25 ms)	約 0.8 倍 (10 ~ 125 ms)	1 port 10/100 Mbps	64 KB	大約 768 KB
iP-8447 iP-8847	80186, 80 MHz 或相容	大約 4 倍 (2 ~ 25 ms)	大約 0.8 倍 (10 ~ 125 ms)	2 ports 10/100 Mbps	64 KB	大約 768 KB
I-8437-80 I-8837-80				1 port 10 Mbps		
I-8437 I-8837 I-8417 I-8817	80188, 40 MHz 或相容	大約 1 倍 (5 ~ 100 ms)	大約 0.2 倍 (25 ~ 500 ms)	-		
μPAC-5x07(P)	80186, 80 MHz 或相容	大約 4 倍 (2 ~ 5 ms)	大約 0.8 倍 (10 ~ 125 ms)	1 port 10/100 Mbps	64 KB	大約 768 KB
μPAC-7186EG						大約 640 KB
μPAC-7186PEG						
I-7188EG I-7188XG	80188, 40 MHz 或相容	大約 1 倍 (5 ~ 100 ms)	大約 0.2 倍 (25 ~ 500 ms)	1 port 10 Mbps		大約 512 KB
				-		

注意: W-8xx7 與 I-8x37 均已停產, 請選用相容產品 WP-8x47 與 iP-8x47.

ISaGRAF 控制器性能比較表二

XPAC, WinPAC 系列 與 iPAC:

OS	WinCE		MiniOS7
型號	XP-8xx7 *1	WP-8x37/WP-8x47 *1	iP-8447/iP-8847 *1
Modbus TCP Master (最大連接數)	100 台設備		-
Modbus RTU/ASCII Master 功能方塊 (Max.)	(每 port) 256		(全部) 128
Modbus RTU/ASCII Master 通訊串口 (Max.) *3	33 port COM1 ~ 33	10 port 1 ~ 14	2 port 1 ~ 5
Modbus RTU Slave 通訊串口 (Max.) *3	9 port COM1 ~ 33	5 port 1 ~ 8	2 port 1 或 2/3
Modbus TCP/IP Slave 連線數 *4	64	32	6
Modbus 位址範圍	1 ~ 8191		1 ~ 4095
VGA 解析度 (Max.)	1024x768	1024x768/800x600	-
USB 串口 *5	2	2/1	-
電池備援 SRAM *6	512 KB		
PAC 與 PAC 資料交換	Ebus		Fbus, Ebus
發送 E-mail (可夾帶附件) *7	是		
備援乙太網路串口 *8	是		
Mbus24r 與 mbus24r1 方塊	是		
Mbus_xr 與 Mbus_xr1 方塊 *9	是		-
可支援軟體功能 (需搭配選購設備)			
支援 FRnet I/O *10	是		
支援 CAN/CANopen *11	是		
支援 VW Sensor	是		
支援 新備援系統 *12	是		-
可支援之遠程 I/O 模組 (選配)			
支援 Ethernet I/O	是 (搭配 I-8KE4/E8-MTCP)		-
支援 I-7K/87K I/O 模組 (* 只支持 1 個 COM 埠)	最大可連接數量: 255		64
	COM 3 或 4	2 或 3	2 或 3 或 4

ViewPAC 系列:

OS	WinCE	MiniOS7
型號	VP-25W7/VP-23W7	VP-2117
Modbus TCP Master (最大連接數)	100 台設備	-
Modbus RTU/ASCII Master 功能方塊 (Max.)	(每 port) 256	(全部) 128
Modbus RTU/ASCII Master 通訊串口 (Max.) *3	10 port COM 2, 3, 5 ~ 14	2 port 1 ~ 3, 5
Modbus RTU Slave 通訊串口 (Max.) *3	5 port COM 2, 3, 5 ~ 8	2 port 1 或 2/3
Modbus TCP/IP Slave 連線數 *4	32	6
Modbus 位址範圍	1 ~ 8191	1 ~ 4095
LCD 螢幕	TFT 5.7"/3.5"	STN
觸控式面板	是/ -	-
VGA 解析度 (Max.)	640x480/320x240	128x64
USB 串口 *5	1	-
電池備援 SRAM *6	512 KB	
PAC 與 PAC 資料交換	Ebus	Fbus, Ebus
發送 E-mail (可夾帶附件) *7	是	
備援乙太網路串口 *8	是	-
Mbus24r 與 mbus24r1 方塊	是	
Mbus_xr 與 Mbus_xr1 方塊 *9	是	-
可支援軟體功能 (需搭配選購設備)		
支援 FRnet I/O *10	是	
支援 CAN/CANopen *11	是	
支援 VW Sensor	是	
支援 新備援系統 *12	是	-
可支援之遠程 I/O 模組 (選配)		
支援 Ethernet I/O	是 (搭配 I-8KE4/E8-MTCP)	-
支援 I-7K/87K I/O 模組 (* 只支持 1 個 COM 埠)	最大可連接數量: 255	64
	COM 2 或 3	

I-8000, iPAC 系列 與 WinPAC:

OS	MiniOS7			WinCE
型號	I-8417/8817	I-8x37-80 *1	iP-8447 iP-8847 *1	WP-8x37/ WP-8x47 *1
Modbus TCP Master (最大連接數)	-			100 台設備
Modbus RTU/ASCII Master 功能方塊 (Max.)	(全部)			(每 port)
	64		128	256
Modbus RTU/ASCII Master 通訊串口 (Max.) *3	2 port			10 port
	COM 1, 3, 4, 5		1 ~ 5	1 ~ 14
Modbus RTU Slave 通訊串口 (Max.) *3	2 port			5 port
	COM 1, 2	1, 3	1 or 2/3	1 ~ 8
Modbus TCP/IP Slave 連線數 *4	0	4	6	32
Modbus 位址範圍	1 ~ 4095			1 ~ 8191
VGA 解析度 (Max.)	-			1024x768/800x600
USB 串口 *5	-			2/1
電池備援 SRAM *6	選購		512 KB	
PAC 與 PAC 資料交換	Fbus	Fbus, Ebus		Ebus
發送 E-mail (可夾帶附件)*7	-	是		
備援乙太網路串口 *8	-	是		是
Mbus24r 與 mbus24r1 方塊	-	是		
Mbus_xr 與 Mbus_xr1 方塊*9	-	是		
可支援軟體功能 (需搭配選購設備)				
支援 FRnet I/O *10	-	是		
支援 CAN/CANopen *11	-	是		
支援 VW Sensor	是			
支援 新備援系統 *12	-	是		
可支援之遠程 I/O 模組 (選配)				
支援 Ethernet I/O (搭配 I-8KE4/E8-MTCP)	-			是
支援 I-7K/87K I/O 模組 (* 只支持 1 個 COM 埠)	最大可連接數量: 64			255
	COM 3 或 4	2 或 3 或 4		2 或 3

I-7188, μPAC 系列:

OS	MiniOS7			
型號	I-7188 XG	I-7188 EG	μPAC-7186EG/PEG *2	μPAC-5x07 *1,2
Modbus RTU/ASCII Master 功能方塊 (Max.)	(全部)			
	64		128	
Modbus RTU/ASCII Master 通訊串口 (Max.) *3	COM2, 3	COM1, 2, 3		
Modbus RTU Slave 通訊串口 (Max.) *3	2 port			
	COM1 或 2/3			
Modbus TCP/IP Slave 連線數 *4	0	4	6	
Modbus 位址範圍	1 ~ 4095			
電池備援 SRAM *6	選購			512 KB
PAC 與 PAC 資料交換	Fbus	Fbus, Ebus		
發送 E-mail (可夾帶附件) *7	-	是		
Mbus24r 與 mbus24r1 方塊	-	是		
可支援軟體功能 (需搭配選購設備)				
支援 FRnet I/O *10	-	是	-	
支援 CAN/CANopen *11	-	是		
可支援之遠程 I/O 模組 (選配)				
支援 I-7K/87K I/O 模組 (* 只支持 1 個 COM 埠)	最大可連接數量: 64			
	COM 2 或 3			

註解:

- *1. μPAC-5x07 代表產品 μPAC-5007(P)/5107(P)/5207/5307.
I-8x37/I-8x37-80 代表產品 I-8437/8837/8437-80/8837-80.
iP-8x47 代表產品 iP-8447/8847.
WP-8x37 代表產品 WP-8137/8437/8837.
WP-8x47 代表產品 WP-8147/8447/8847.
XP-8xx7-CE6 代表產品 XP-8047-CE6/8347-CE6/8747-CE6
- *2. μPAC-7186PEG 是具有 PoE (Power-over-Ethernet) 的模組。
μPAC-5007P/5107P 是具有 PoE (Power-over-Ethernet) 的模組。
- *3. I-8000 的 COM5 ~ 20, W-8x47/ 8x37 的 COM5 ~ 14 位於 I-8112/ I-8114/ I-8142/ I-8144/ I-8142i 擴充卡上;

iP-8x47 的 COM5 ~ 20, VP-2117 的 COM5 ~ 16 位於 I-8112iW/ I-8114W/ I-8114iW/ I-8144iW/ I-8144iAW 擴充卡上;

WP-8x47, WP-8x37 和 VP-25W7/23W7 的 COM5 ~ 14 位於 I-8112iW/ I-8114W/ I-8114iW/ I-8144iW/ I-8144iAW 擴充卡上;

XP-8x47-CE6 的 COM6 ~ 33, 位於 I-8112iW/ I-8114W/ I-8114iW/ I-8144iW/ I-8144iAW 擴充卡上;

I-7188/ μPAC-7186 的 COM3 ~ 8, 位於 X5xx X-board 擴充卡上.

μPAC-5x07 的 COM3 ~ 8, 位於 XW5xx XW-board 擴充卡上.

- *4. XP-8xx7-CE6 的 乙太網路通訊 比 WP-8xx7, VP-2xW7 更具效率. 最多支持 64 個 Modbus TCP/IP 連線.

W-8x47 的 driver 如果是 4.02 版或之前的版本只支持 8 個 Modbus TCP/IP connection. 由 4.03 版 driver 起支持最多 32 個 Modbus TCP/IP connection.

如果使用的控制器是 WinCon-8347/8747 (有 2 個乙太網路埠), 它的 OS image 必需是使用 2008 年 7 月 1 日發佈的版本, 網路通訊才能正常運作.

詳情請參考 www.icpdas.com > FAQ > Software > ISaGRAF > 095.

- *5. XP-8xx7-CE6 上供滑鼠使用的 USB 埠, 比 WP-8xx7, VP-2xW7 更具效率. WP-8x37 支援 2 個 USB 埠, WP-8x47 支援 1 個 USB 埠.

- *6. I-8x17/8x37-80 搭配 S256/S512, μPAC-7186EG, I-7188EG/XG 搭配 X607 (128K) / X608 (512K), 最多可支援 1024 個保存變數, 也可儲存資料與時間.

- *7. μPAC-7186EG 需使用 X607/X608 battery backup SRAM 擴充卡, 若沒有選購 battery backup SRAM 只可發送沒有附件的 E-mail.

- *8. 指的是 當 PC/HMI 與其中一個 Ethernet 埠的 Modbus TCP/IP 斷線不通時, PC/HMI 可以改去跟另一個 Ethernet 埠來繼續通訊. (請於 VP-25W7/23W7 上裝一片 I-8135W 卡, 以擴增第二個 Ethernet 埠.)

- *9. Mbus_xr 與 Mbus_xr1 功能方塊, 最多可讀取 120 個字元或 60 個長整數或 60 個實數. 請參考 www.icpdas.com > FAQ > Software > ISaGRAF > FAQ-101 取得更多資訊.

- *10. μPAC-7186EG 需選購 FX-016 擴充卡插於控制器內, 以支援 FRnet I/O.

VP-25W7/23W7, VP-2117 最多支援 **3** 片 I-8172W (最多 768 DI 和 768 DO 點)

iPAC-8x47 最多支援 **4** 片 I-8172W (最多 1024 DI 和 1024 DO 點).

WP-8x47, WP-8x37 最多支援 **8** 片 I-8172W (最多 2048 DI 和 2048 DO 點).

XP-8xx7-CE6, W-8xx7 最多支援 **7** 片 I-8172W (最多 1792 DI 和 1792 DO 點).

- *11. XP-8xx7-CE6, μ PAC-5x07, μ PAC-7186EG, iP-8x47, WP-8x47, WP-8x37, VP-25W7/23W7, VP-2117 與 W-8xx7 支援 I-7530 (RS-232 轉 CAN 轉換器) 連接至其它的 CAN/CANopen 設備.
- *12. 只有 XP-8xx7-CE6, WP-8x47, WP-8x37, VP-25W7/23W7 與 W-8x47 有支援新的備援系統, 而 W-8x37 沒有支援.

如何選用 ISaGRAF PAC

記憶體:

1. I-8417, I-8817, I-8437-80, I-8837-80, I-7188EG, μ PAC-7186EG, I-7188XG, VP-2117 與 iP-8447 / iP-8847 的記憶體比較受限. 這些機型不能執行大於 64K byte 的 ISaGRAF 程式 (編譯完後 “appli.x8m” 檔的大小)
2. WP-8147, WP-8447, WP-8847 與 VP-25W7, VP-23W7 可執行最多到 1M byte 的 ISaGRAF 程式. 程式容量為 I-8xx7 與 I-7188EG/XG 的 16 倍大 (XP-8xx7-CE6 則是 2M byte).

CPU 執行速度:

I-8417/8817, I-7188EG 與 I-7188XG 的 CPU 為 80188, 40MHz 或相容. 它是 16 位元的 CPU. 不善於進行實數(浮點數)運算. 如果你的應用會做很多實數運算的話, 最好是選用 WP-8xx7 或 VP-25W7 / VP-23W7 或 XP-8xx7-CE6 或日後較先進的 ISaGRAF 控制器. 他們的 CPU 為 32 位元, 他們的執行速度約為 I-8xx7 及 I-7188EG/XG 的 10 ~ 20 倍快, 特別是進行實數運時, 有時會比 I-8xx7 及 I-7188EG/XG 快 50 ~ 100 倍, 主要是因為 80188 CPU 處理實數(浮點數)很慢. I-8x37-80 與 iP-8xx7 與 μ Pac-7186EG 的執行速度則約為 I-8xx7 及 I-7188EG/XG 的 2 ~ 4 倍快. I-8x37-80 與 iP-8xx7 與 μ Pac-7186EG 處理實數也約為 I-8xx7 及 I-7188EG/XG 的 2 ~ 4 倍快

備援/冗餘:

XP-8xx7-CE6 與 WP-8xx7 支持備援功能. 2 台 PAC 來構成 1 個備援系統.

一個是 Main, 一個是 Backup. 一開始通常是 Main 那台會掌管所有 RS-485 I/O (I-7k & I-87K) 或 I-8KE4-MTCP 或 I-8KE8-MTCP 或 RU-87Pn 上 input 與 output 的控制權. 假如 Main 死機了, Backup 會接管控制權繼續運作. 所有的 Output 需規劃使用遠程 RS-485 或 Ethernet I/O. Input 可以是插在第 1~7 槽的 input 或 RS-485 遠程 input.

網路:

1. 目前 WP-8xx7, XP-8xx7-CE6 與 iP-8447 / iP-8847 有支持 雙網口, 10/100 M. I-7188EG, μ P-7186EG 及 I-8437-80, I-8837-80 都只支持單一網口(其中 μ PAC-7186EG 為 10/100 M, 其它是 10M). 所有 ISaGRAF 有網口的控制器都支持 Modbus TCP/IP slave 協議. I-7188XG 與 I-8417/8817 與 iP-8417/8817 則沒有網口.
2. XP-8xx7-CE6, WP-8xx7 與 VP-25W7 / VP-23W7 或日後較先進的 ISaGRAF 控制器有支持用 UDP/IP 或 TCP/IP 來收/發 使用者自定的字串 到 PC 或其它的設備上. 而 I-8417, I-8817, I-7188EG, I-8437-80, I-8837-80, iP-8417, iP-8817 無此功能(μ PAC-7186EG, iP-8x47 有支持 UDP).

視窗顯示 (Windows):

WP-8xx7, VP-25W7/23W7 與 XP-8xx7-CE6 控制器支持視窗顯示 (VP-25W7 有 Touch screen).

WP-8xx6, VP-25W6/23W6 與 XP-8xx6-CE6 支持 ISaGRAF 與 InduSoft(VP-25W6 有 Touch screen)

Modbus RTU Slave 通訊串口:

WP-8xx7, VP-25W7, VP-23W7 最多可支持 5 個串口為 Modbus RTU Slave 串口. (請參閱它的產品包裝盒內的 Getting Started Manual 內的 Appendix G, E 與 A.2)

XP-8xx7-CE6 最多可支持 9 個串口為 Modbus RTU Slave 串口

I-7188EG/XG, 7186EG, iP-8xx7, I-8xx7, VP-2117 最多可支持 2 個串口為 Modbus RTU Slave 串口. (請參閱各別產品包裝盒內的快速上手手冊)

第 1 章 安裝軟體及硬體

注意：

本書內出現的 **I-8xx7** 是 I-8417、I-8437、I-8817 和 I-8837 控制器的縮寫。

而 **WP-8xx7** 是 WP-8147/8447/8847 及 WP-8137/8437/8837 的縮寫。

而 **XP-8xx7-CE6** 是 XP-8047-CE6 / XP-8347-CE6 / XP-8747-CE6 的縮寫。

而 **iP-8xx7** 是 iP-8447 / iP-8847 的縮寫。

1.1: 安裝 ISaGRAF Ver.3 作業平台

為了使 I-8xx7, I-7188EG/XG, μ PAC-7186EG, iP-8xx7, WP-8xx7, VP-25W7 / VP-23W7 與 XP-8xx7-CE6 控制系統以及 ISaGRAF 作業平台正常運作，正確地完成各項設定是必要的。本章將花費您最少的時間完成各項設定。在您開始以 ISaGRAF 進行 PAC 嵌入式控制系統的程式設計之前，您必須在您的電腦上先安裝 ISaGRAF 3.x 版 作業平台。

硬體需求

Pentium 133(含)以上等級的電腦

最少 32 MB 的記憶體(建議能有 64 MB 以上)

最少 128 MB 儲存空間的硬碟

最少有一個 RS-232 序列埠

軟體需求

在安裝 ISaGRAF 作業平台前，您必須先安裝以下任何一種作業系統

- Windows 98 或 Windows 2000 或 Windows XP
- Windows NT Version 3.51 或 Windows NT Version 4.0
- Windows Vista 或 Windows 7 (請參考 [FAQ-117](#))

安裝 ISaGRAF 作業平台

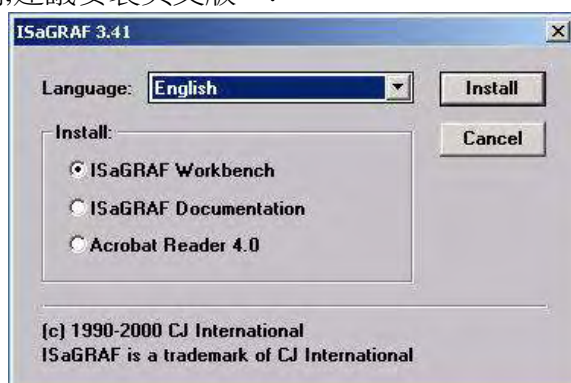


使用 Windows Vista 或 Windows 7 (32-bit)，請先參考 [1.1.4 節](#)。

使用 Windows 7 (64-bit) 作業系統，請先參考 [1.1.5 節](#)。

將 ISaGRAF 軟體光碟片放入光碟機中，通常具有自動執行程式功能的電腦將自動執行 "install.bat" 檔。如果您的電腦並不支援此項功能，您可以開啟檔案管理員，執行光碟機目錄下的 "install.bat" 檔。如果無法找到此檔，您可執行 "ISaGRAF.exe" 開始安裝程序。

當您執行 "install.bat" 檔後將出現如下的對話畫面，請選擇您想使用的語言版本，本書將以英文版作為範例，建議安裝英文版。



安裝完畢後，您可以開始使用 ISaGRAF

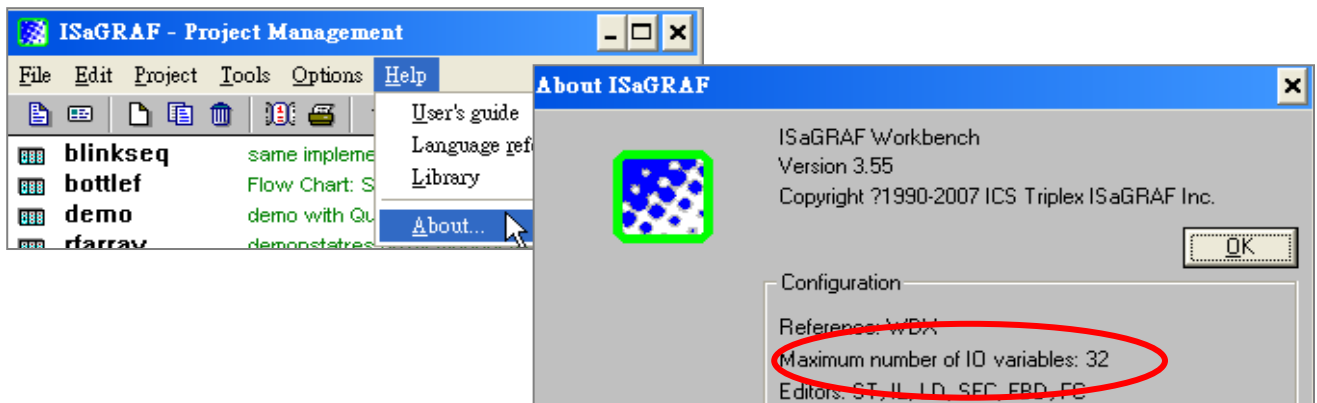
請點選左下角的[開始]->[程式集]後，您會看到類似如下的畫面



1.1.1 保護鎖注意事項 (Dongle 與 USB Key-Pro)

您必須安裝 ISaGRAF 提供的硬體保護鎖(dongle:安裝於電腦的並列埠) 或 USB 保護鎖 (ISaGRAF 3.51 版起使用)，安裝完成才能使用 ISaGRAF 所有的功能。(ISaGRAF-32-E 或 ISaGRAF-32-C 不需保護鎖)

請執行 ISaGRAF 的 [Help] > [About] 查看安裝是否正確，若出現 "Maximum number of IO variables: 32" 表示 ISaGRAF 無法找到保護鎖，請確認安插正確，然後將電腦重新開機，再查看一次。



如果仍然出現 "Maximum number of IO variables: 32"，表示保護鎖的驅動程式沒有安裝成功，請依保護鎖型態執行下列步驟。

Dongle 型保護鎖:

請執行下列驅動程式(位於 ISaGRAF 光碟)，然後將電腦重新開機:

- ISaGRAF-80 版本: \Sentinel5382\setup.exe
- 其他 ISaGRAF 版本: \Sentinel\setup.exe

USB 型保護鎖:

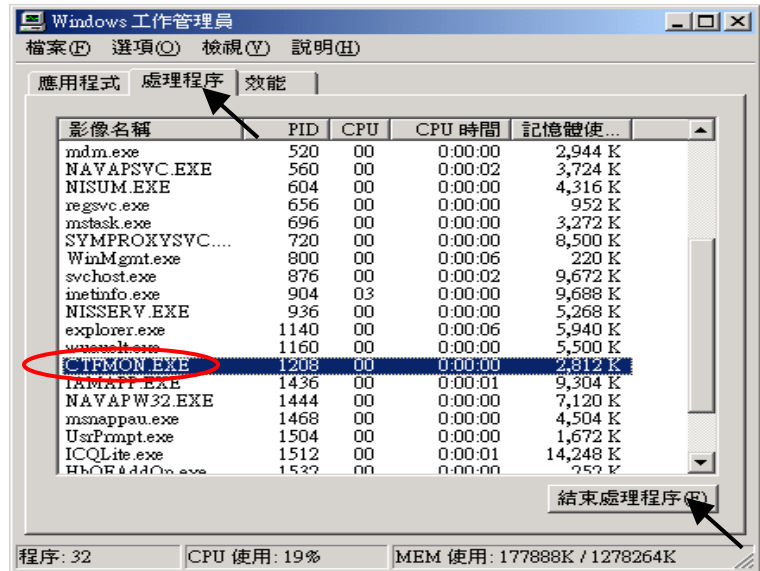
- 1.安裝 ISaGRAF 軟體後，請先暫時取下 USB 保護鎖，執行 ISaGRAF 3.51~3.55 光碟裡的程式 "\Sentinel\ SSD5411-32bit.exe"，然後重新開機。
- 2.之後，執行 ISaGRAF 時，皆需插上 USB 保護鎖。

1.1.2 Windows 2000 使用者的注意事項：

如果您使用 Windows 2000 作業系統，在關閉某些 ISaGRAF 視窗時，有些電腦會出現停滯 20~40 秒（沒有反應）的現象。可能是 Windows 2000 裡 "CTFMON.EXE" 程式的關係。

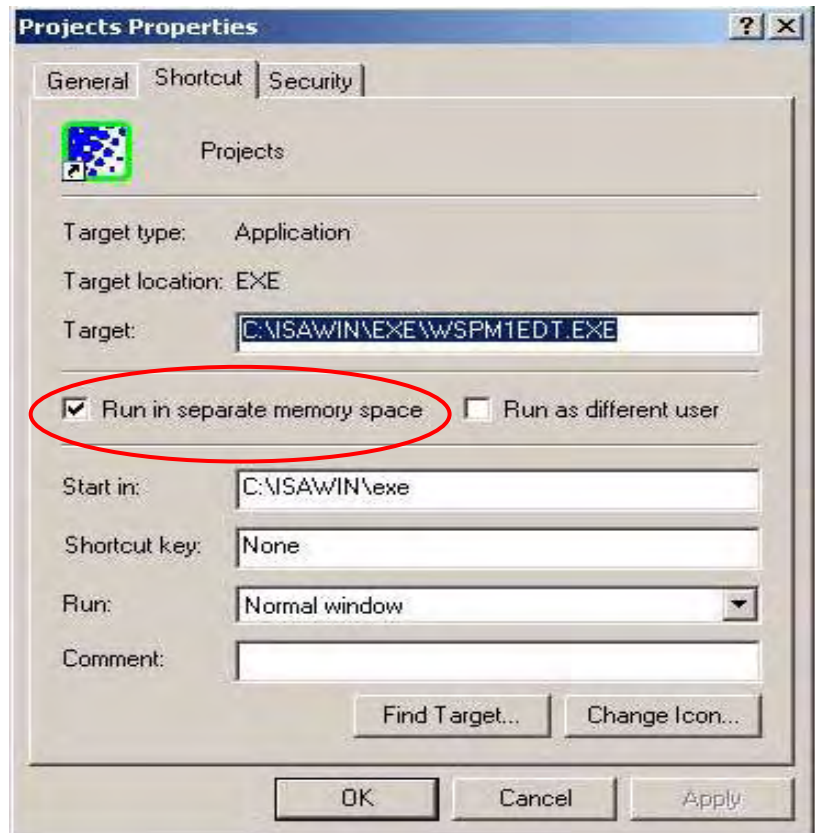
發生視窗停滯問題的處理方法：

您可以同時按住鍵盤的“Ctrl”+“Alt”+“Del”，出現如下畫面，請依照下圖指示，結束“CTFMON.EXE”的處理程序。



解決視窗停滯問題的方法：

為 "ISaGRAF project manager" 建立一個捷徑，並在捷徑的設定視窗中選取 "run in separate memory space" 的選項設定。



1.1.3 Windows NT 使用者的注意事項：

如果您使用 Windows NT 作業系統，您需要在 ISaGRAF 作業平台下的子目錄 "EXE" 內的 "isa.ini" 檔內加入一行 "NT=1"。

如果您將 ISaGRAF 安裝在 C 硬碟，您會在下面路徑找到該檔：

C:\isawin\exe\isa.ini

您可以使用任何 ASCII 文字編輯器（例如 Notepad 或 UltraEdit32）開啟 "isa.ini"，找到 [WS001] 標頭（應該在檔案的最上方），在 [WS001] 部分的任何位置加入一行 "NT=1"，為了 RS-232 通訊能正常運作，"NT=1" 這行敘述一定要加入。如下：

[WS001]

NT=1

Isa=C:\ISAWIN

IsaExe=C:\ISAWIN\EXE

Group=Samples

IsaApl=c:\isawin\smp

IsaTmp=C:\ISAWIN\TMP

1.1.4 Windows Vista 或 Windows 7 (32-bit) 使用者的注意事項

安裝 ISaGRAF 軟體之前，若您使用的是 Windows Vista 或 Windows 7 (32-bit) 作業系統，為了避免部分安裝的限制，請先變更使用者帳戶控制設定。

如何關閉使用者帳戶控制？



此設定須具備系統管理員的權限。

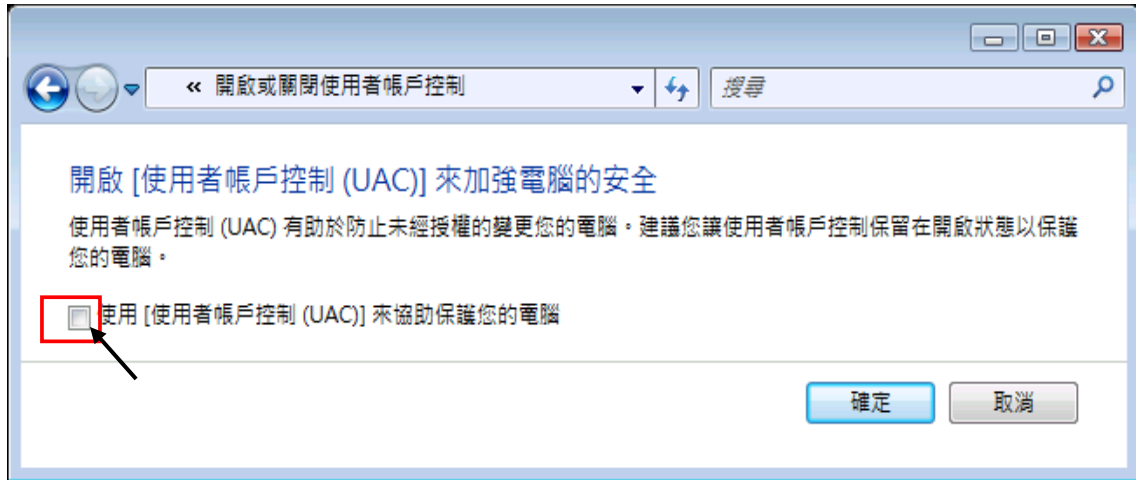
1. 請於 [開始] > [控制台] > [使用者帳戶和家庭安全] > [使用者帳戶] > 點選 [開啟或關閉使用者帳戶控制] 或 [變更使用者帳戶控制設定]。



2. 點選後，將出現以下的畫面。

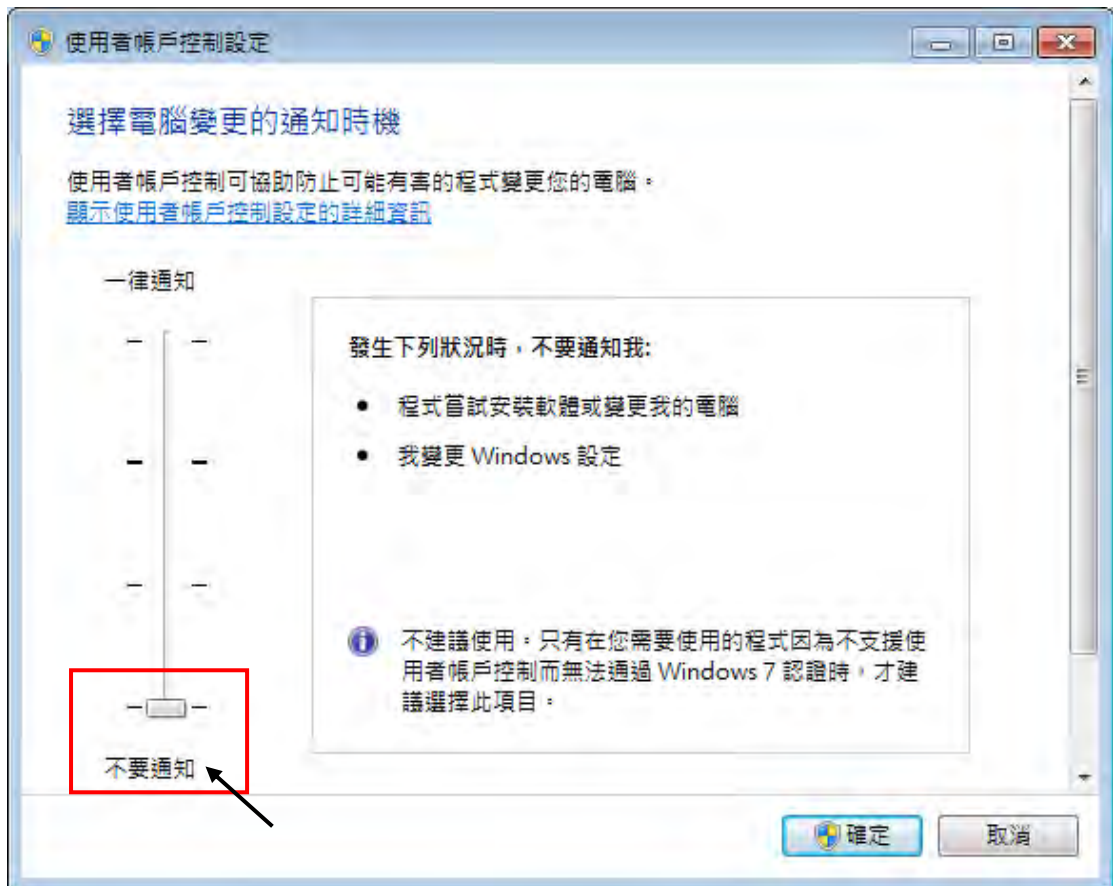
Windows Vista:

畫面中標識的地方不要打勾，最後按下“確定”鍵。



Windows 7:

將畫面中的橫桿移至“不要通知”，最後按下“確定”鍵。



3. 請重新啟動電腦。

4. 電腦重新開機後，請參考 章節 [1.1 節](#)。

1.1.5 Windows 7 (64-bit) 使用者的注意事項

若您使用 Windows 7 (64-bit) Professional/Enterprise/Ultimate 作業系統，則須在 XP Mode 下才可安裝 ISaGRAF 軟體。請先執行以下步驟來安裝 Virtual PC 和 XP Mode。

安裝 Virtual PC 和 XP Mode:

1. 請從 Windows Virtual PC 網站 (<http://go.microsoft.com/fwlink/?LinkID=160479>) 下載 Windows Virtual PC 與 Windows XP Mode 的安裝程式。
2. 滑鼠雙擊 “WindowsXPMode_nn-NN.exe” (此處的 nn-NN 代表語言環境，例如 en-US)，並依照安裝精靈的導引完成 Windows XP Mode 安裝。
3. 滑鼠雙擊 “Windows6.1-KB958559-x64.msu”，安裝 Windows Virtual PC。
4. 重新啟動電腦。
5. 請依序點選電腦 [開始] > [所有程式] > [Windows Virtual PC]，開啟 [Windows XP Mode]。
6. 依照導引精靈完成 Windows XP Mode 的設定。請記得一定要將安裝過程中提供的密碼記錄下來，因為您需要此密碼來登入虛擬機器。
7. 請回到[章節 1.1](#) 來安裝 ISaGRAF。

1.1.6 使用 Variable Arrays 變數陣列的注意事項

使用 Variable Arrays 變數陣列的設定如下

使用變數陣列必須在 [c:\isawin\ese\isa.ini](#) 檔案的最前端加入兩行程式碼:

```
[DEBUG]
Arrays=1
```

1.2: 安裝 ICP DAS Utilities For ISaGRAF

“ICP DAS Utilities For ISaGRAF” 包含 3 個主要項目

- I/O library (For ISaGRAF PAC)
- Modem_Link utility
- Auto-scan I/O utility

在安裝 ICP DAS Utilities For ISaGRAF 前需先安裝 ISaGRAF 軟體作業平台，如果您尚未安裝 ISaGRAF 作業平台，請參考 1.1 節安裝後再繼續下列步驟。

您手中的 ICP DAS CD-ROM 內包含您所需要的 ICP DAS Utilities For ISaGRAF，請將此 CD-ROM 放入光碟機內，以下詳細的步驟將引導您完成安裝。

在 CD-ROM 內的 \napdos\isagraf\ 目錄，請執行 “setup.exe”



注意:

如果你沒有 ICP DAS CD-ROM 或找不到 “\napdos\isagraf\setup.exe”，可從 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 點選畫面右方 Driver 內去下載 “ICP DAS Utilities For ISaGRAF.zip”

1.3: PAC 硬體相關設定

關於 PAC 硬體相關設定，例如 IP address, 控制器的站號, Modbus RTU slave port 的設定, 通訊串口的接腳圖 ...等, 請參考個別 PAC 的快速上手手冊, 放於 PAC 包裝盒內附的 CD-ROM 內

WP-8xx7 : \napdos\isagraf\wp-8xx7\chinese_manu\chinese_getting_started_wp-8xx7.pdf

XP-8xx7-CE6 : \napdos\isagraf\xp-8xx7-ce6\chinese_manu\getting-started-xp-8xx7-ce6-chinese.pdf

VP-2xW7 CD: \napdos\isagraf\vp-25w7-23w7\chinese_manu\chinese-getting-started-vp-2xw7.pdf

I-8000 & I-7188 CD:

iP-8xx7: \napdos\isagraf\ip8000\chinese_manu\ipac-8x47_getting_started_chinese.zip

µPAC-7186EG, I-7188EG/XG : \napdos\isagraf\7188eg\chinese_manu\

718xegxg_getting_started_chinese.zip

VP-2117 CD: \\napdos\isagraf\vp2k\chinese_manu\ vp-2117_getting_started_chinese.zip

或上網下載 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 按下畫面右方的 Manual

第 2 章 ISaGRAF 程式設計基礎

注意：

本書內的出現的 **I-8xx7** 是 I-8417、I-8437、I-8817 和 I-8837 控制器的縮寫。

而 **WP-8xx7** 是 WP-8147/8447/8847 及 WP-8137/8437/8837 的縮寫。

而 **XP-8xx7-CE6** 是 XP-8047-CE6 / XP-8347-CE6 / XP-8747-CE6 的縮寫。

而 **iP-8xx7** 是 iP-8447 / iP-8847 的縮寫。

2.1: 一個簡單的階梯圖程式 (LD)

本章使用簡單的例子來說明五種簡單但是有效率的 IEC61131-3 語言進行 I-8xx7, WP-8xx7, iP-8xx7, VP-25W7 / VP-23W7, I-7188EG/XG, VP-2117 與 XP-8xx7-CE6 控制器的程式設計，初學者請按本章的說明操作練習。

若需要知道如何使用更多 ISaGRAF 的功能,可參閱附錄 E, 或 ISaGRAF CD_ROM 中 “ISaGRAF USER’S GUIDE” 手冊的電子檔 “ISaGRAF.pdf” 或是 “ISaGRAF.doc”。

階梯圖語言程式設計基礎

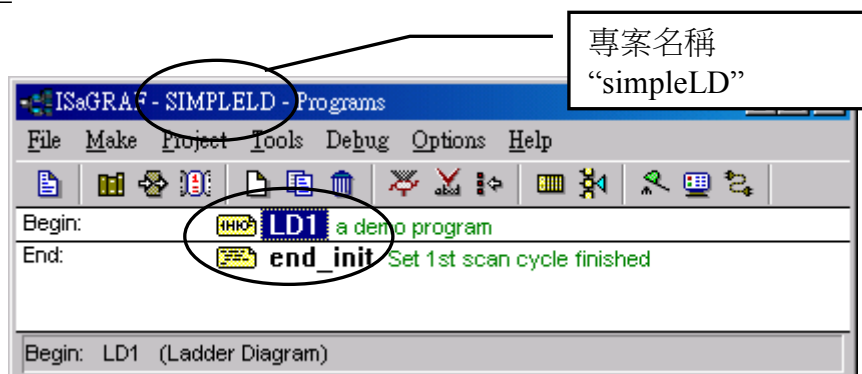
階梯圖程式語言是一種結合 contacts (輸入參數) 和 coils (輸出參數)的圖形化布林方程式語言，其與電機人員在工業控制系統上常用的電路圖表示法類似，階梯圖程式語言讓程式設計師將硬體設備的狀態及其改變情形以圖示法表示出來。

階梯圖語言程式設計範例

我們將依序引導您利用 ISaGRAF 作業平台建構一個階梯圖語言程式。

同時我們會另外建一個 結構化文字 語言程式 (ST), 如此在這個 ISaGRAF 專案範例中,我們有 2 個程式包函在裏面。

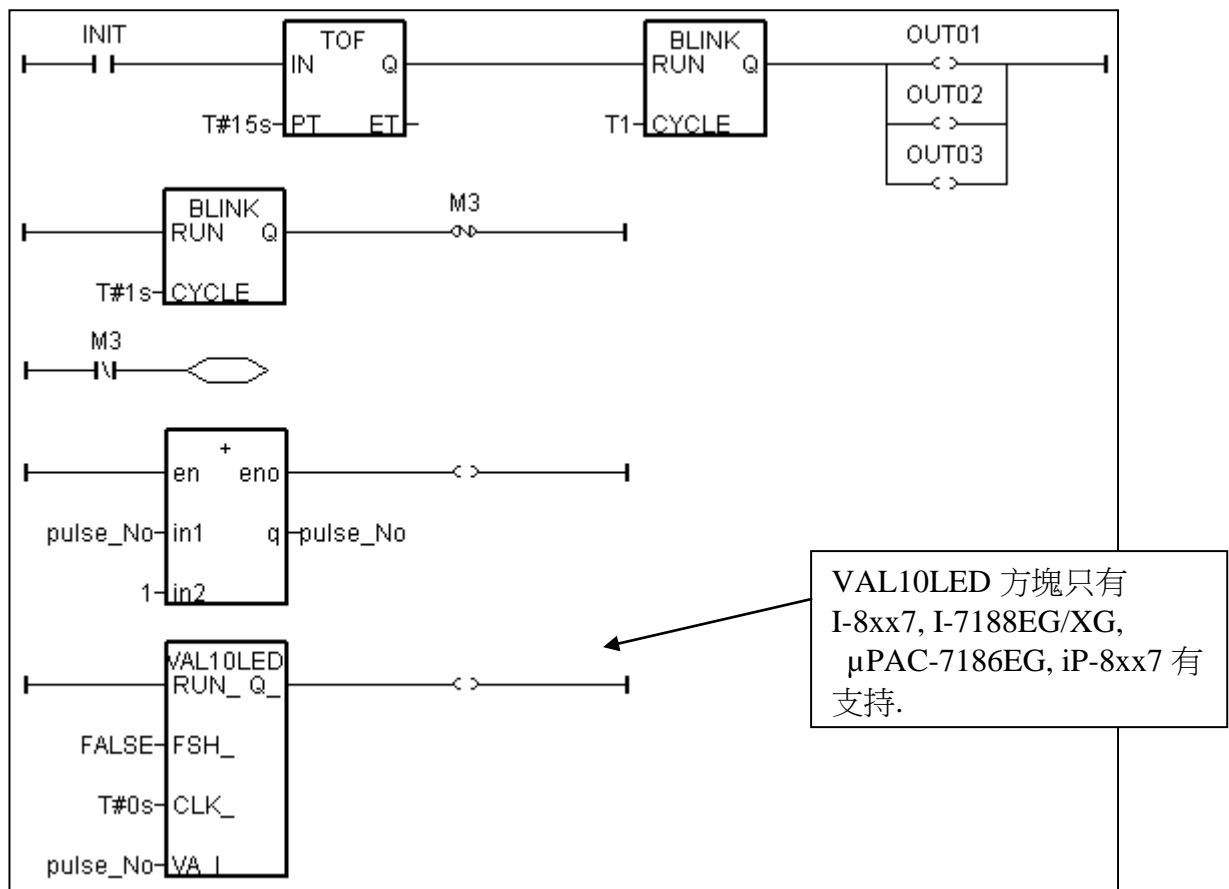
我們把這個專案取名為 “simpleLD”. 而此專案內的 LD 程式取名為 “LD1”, ST 程式取名為 ”end_init”



本範例內的變數宣告：

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 TRUE . TRUE 表示 第 1 個 Scan 周期
M3	Boolean	Internal	用來表示 pulse 有沒有產生
OUT01	Boolean	Output	Output 1
OUT02	Boolean	Output	Output 2
OUT03	Boolean	Output	Output 3
T1	Timer	Internal	閃爍周期時間, 初值設為 T#1s
Pulse_No	Integer	Internal	當 pulse 產生時, 加 1, 初值設為 0

階梯圖程式 “LD1” 全貌



結構化文字程式 “end_init” 全貌

```
INIT := FALSE ;
```

範例程式邏輯說明：

“LD1” 程式：

在程式執行起來的前 15 秒，閃爍輸出 Outputs 1, 2, & 3，閃爍周期為 “T1”，“T1” 的初值為 1 秒。15 秒過後，Outputs 1, 2, & 3 會設成 FALSE。

每秒產生一個 Pulse 到布林內部變數 “M3” 內。

每當 M3 – Pulse 產生時，將整數變數 “pulse_No” 的值加 1。

將 “pulse_No” 的值顯示到 I-8xx7, iP-8xx7 的 7 段式顯示 Led 上。

ST 程式 “end_init”：

在本專案的末端，將布林變數 “INIT” 的值設為 FALSE。如此 INIT 只會在第 1 個 Scan 周期時為 TRUE。

方塊及一些 LD 元件的描素：

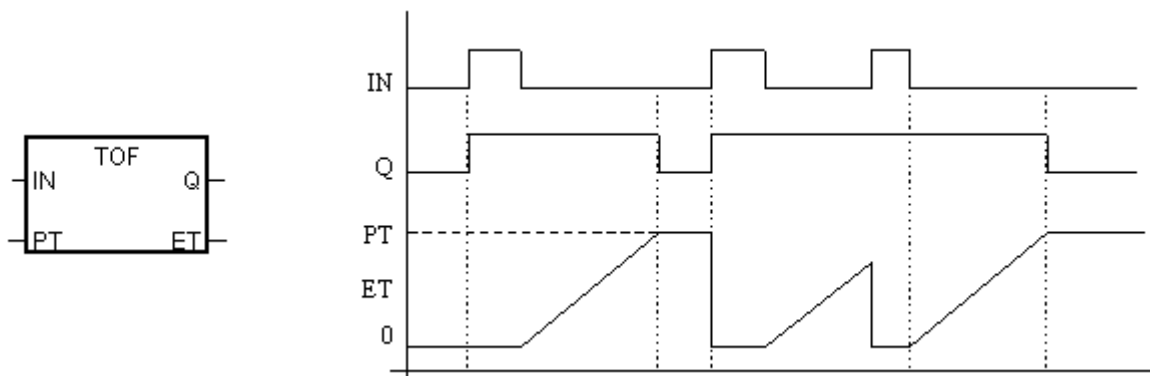
TOF: 延遲一個布林值一段時間後才設為 FALSE。

“IN” 參數為布林型態，當它剛好從 TRUE 降為 FALSE 時，此方塊開始從 0 秒計時到 “PT”

“PT” 參數為計時器型態，它指的是所要延遲的時間。

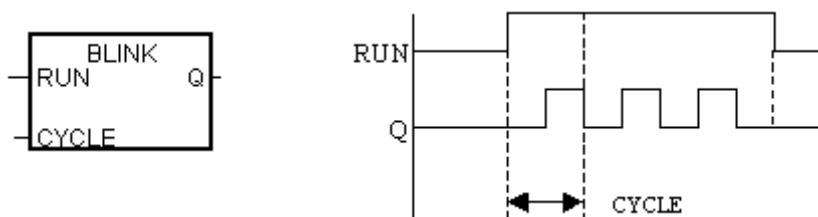
“Q” 為此方塊的布林輸出。當計時時間到達 “PT” 時，就會被設成 FALSE。

“ET” 為此方塊的計時器輸出。(本範例沒有使用它)



BLINK: 閃爍一個布林

“RUN” 參數為布林型態，當它為 TRUE 時，會產生一個閃爍的輸出到 “Q”，周期為 “CYCLE”。



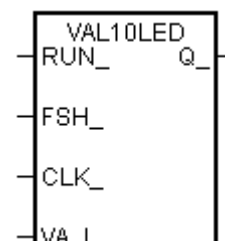
VAL10LED: 顯示 1 個整數值到 I-8xx7, iP-8xx7 的 7 段式顯示 Led 上。

“RUN_” 參數為布林型態。TRUE 就顯示。

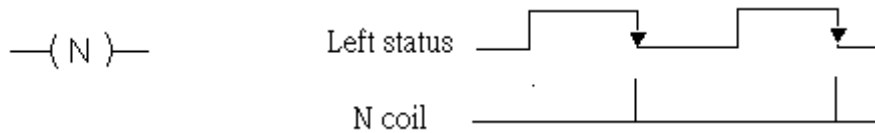
“FSH_” 參數為布林型態。TRUE 會閃爍該顯示的數字。

“CLK_” 參數為計時器型態。為閃爍的周期時間。

“VA_I_” 參數為整數型態，指的是所要顯示的數字。



“N” coil：當 coil 左端的狀態剛好由 TRUE 降為 FALSE 時，該 coil 會產生一個 Pulse 輸出。



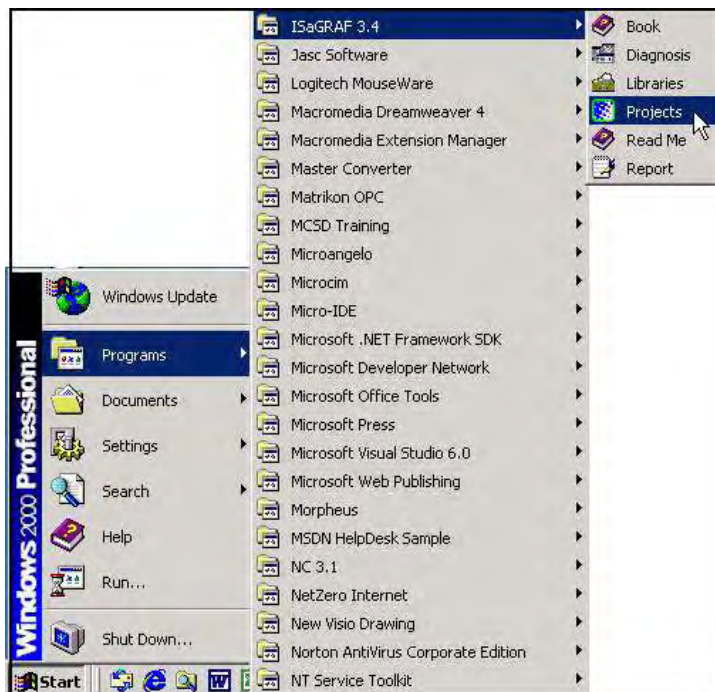
“Retrun”：當 return 左端的狀態為 TRUE 時，在此 return 以下的 LD 程式不會被執行。



2.1.1: 程式撰寫

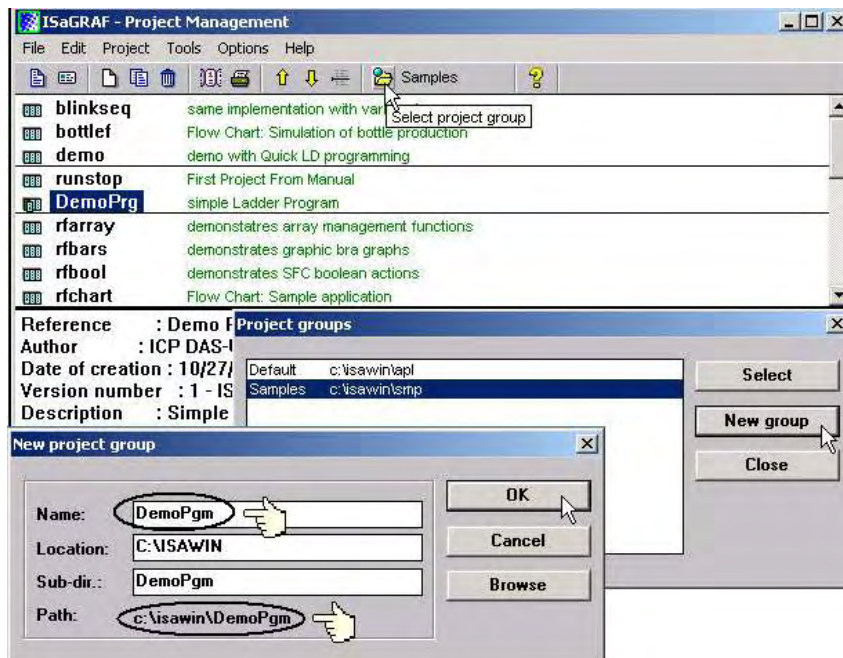
啟動 ISaGRAF 作業平台

如下圖所示，選擇左下方的[開始]->[程式集]->[ISaGRAF 3.4]->[Projects]



2.1.1.1: 建立 ISaGRAF 使用者群組

如下圖所示，選擇 [Select Program Group] -> [New Group]，輸入您想建立的群組名稱，按下 "OK" 鍵。

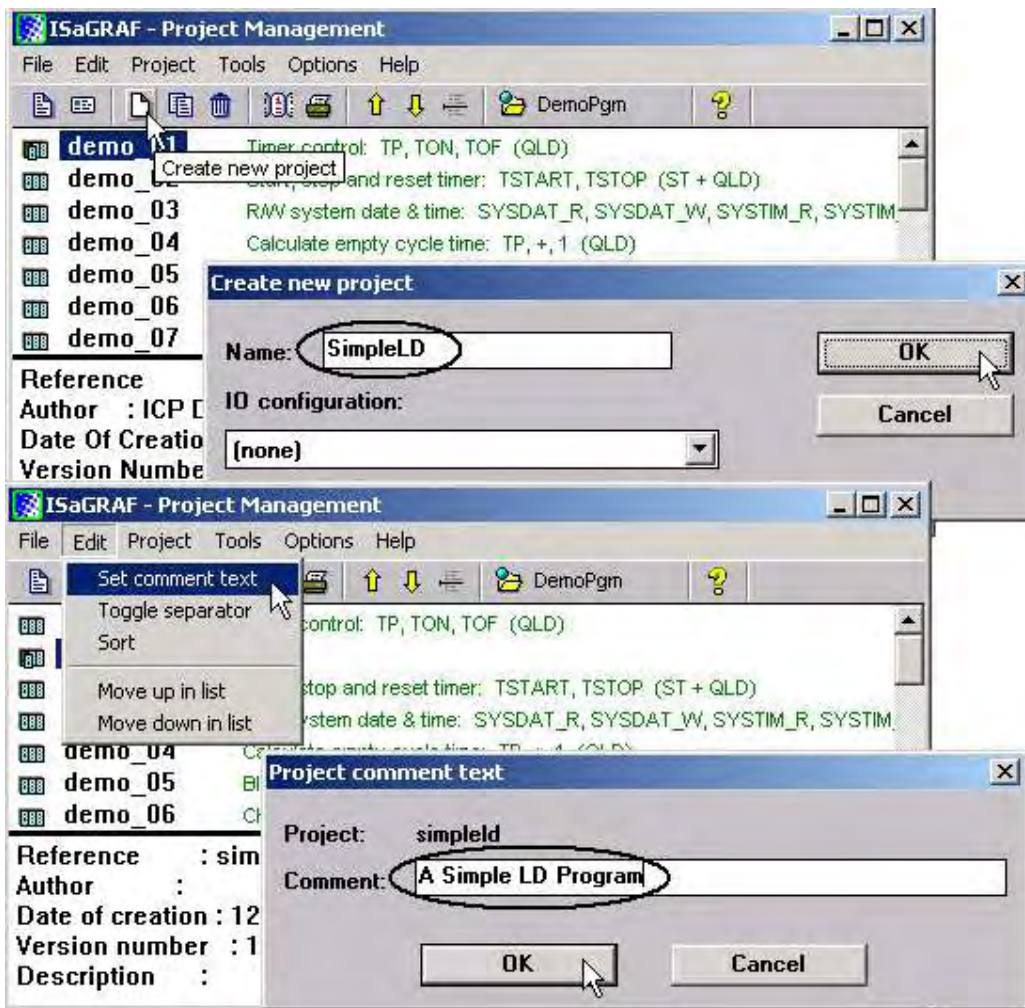


ISaGRAF 將依照您輸入的群組名稱在 "New Project Group" 加入一個新的資料夾，並且在 "c:\isawin" 下亦同時產生一個對應的資料夾。

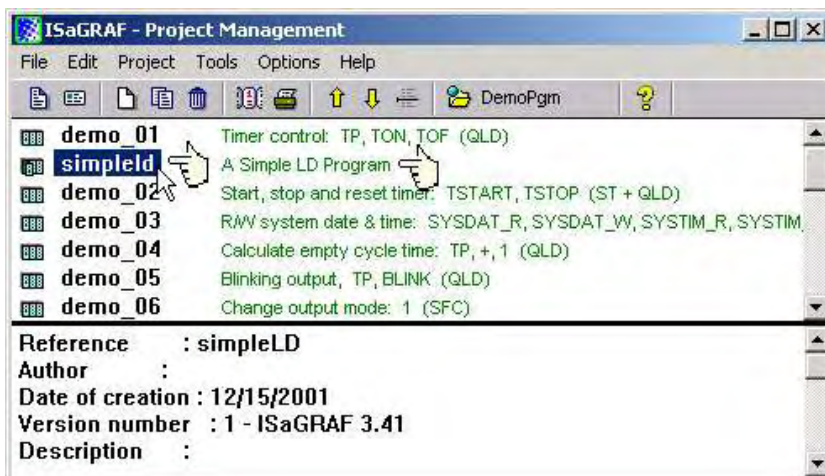
您可以雙擊這個新群組名稱以進入該新群組，或者選擇這個新群組的名稱(群組名稱會變亮)，然後按下 "Select" 鍵進入新群組。

2.1.1.2: 建立新的 ISaGRAF 專案

您可以點選 "Create New Project" 圖示，並鍵入您要的專案名稱以建立您的 ISaGRAF 專案，如果您想要替您的專案加註解，請選擇 [Edit] > [Set Comment Text]

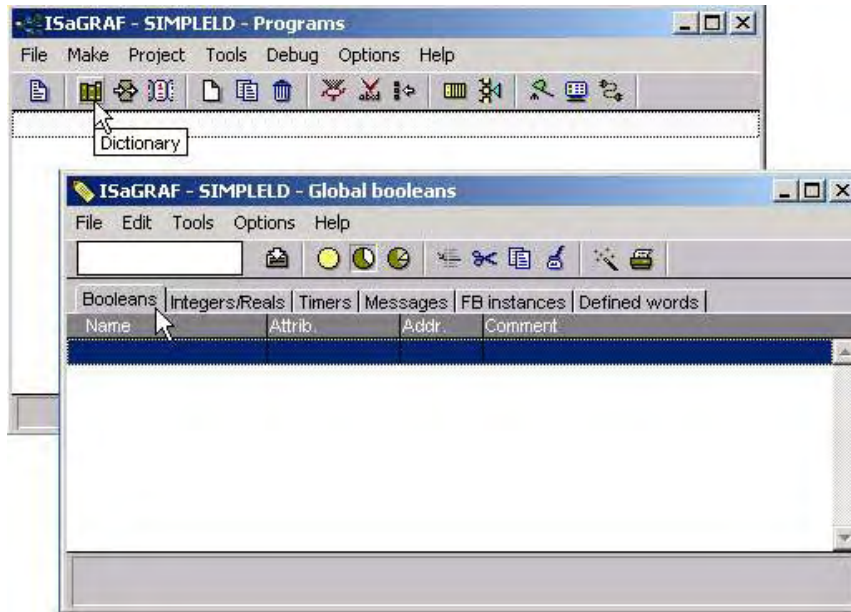


您現在在 "Project Management" 視窗可以見到您的專案名稱，您可以用滑鼠雙擊您的專案名稱來開啟它。

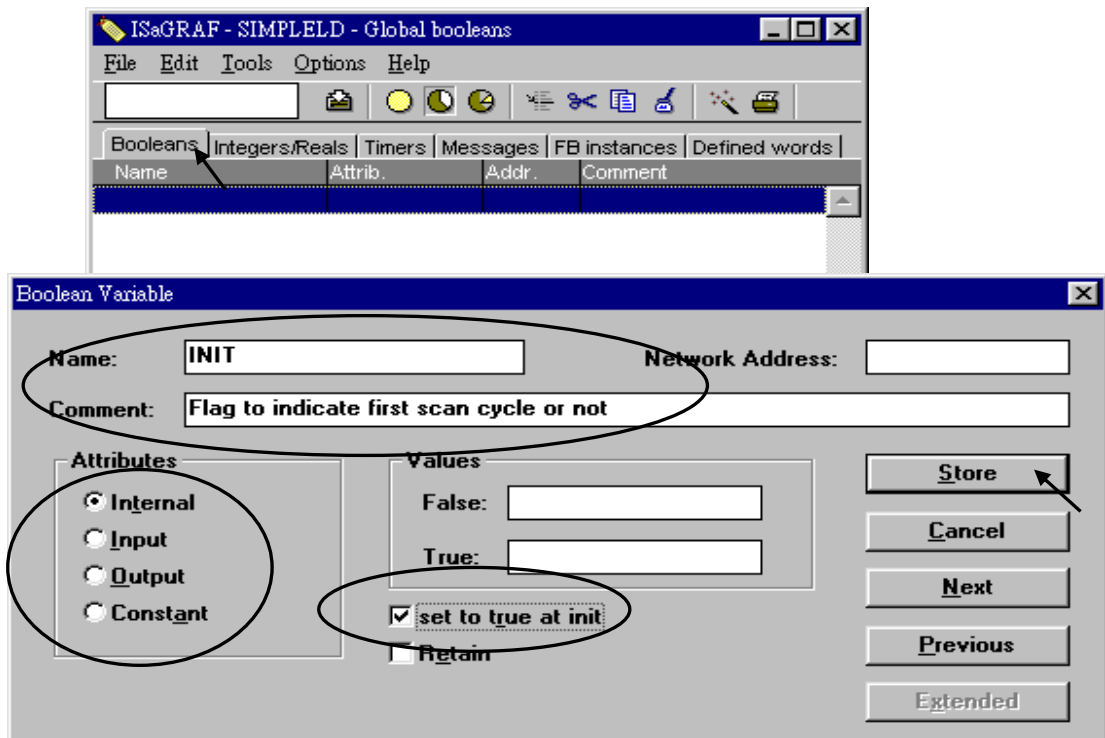


2.1.1.3: 宣告 ISaGRAF 專案變數

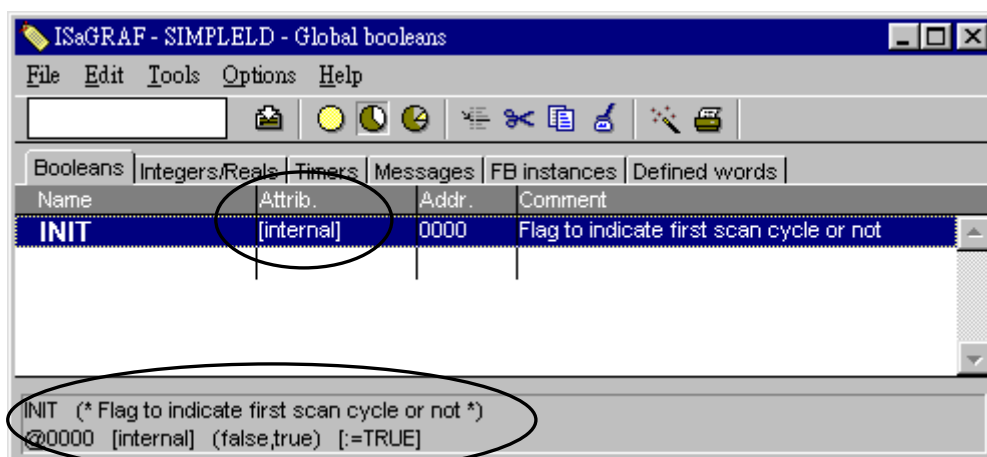
在您開始撰寫 ISaGRAF 程式前您必須先宣告在程式中的各個變數，首先，請點選 "Dictionary" 圖像，再點選 "Boolean" 標籤宣告在程式中會用到的布林變數。



您可以利用滑鼠雙擊 "Boolean" 標籤下有顏色的區域開啟 "Boolean Variable" 視窗，並依序鍵入您需要的變數名稱，本範例中輸入布林參數名稱為 "INIT"，在 "Comment" 欄加入 "Flag to indicate first scan cycle or not" 註解，然後須選擇 "INIT" 變數的屬性，在此範例中 "INIT" 的屬性為 "Internal"，並將 "set to true at init" 打勾，完成上述步驟後，按下 "Store" 鍵，儲存相關的設定。

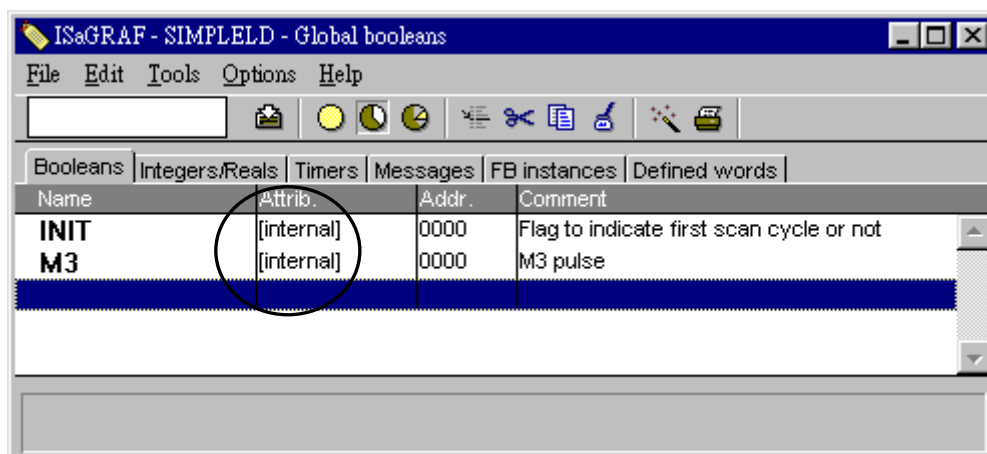


如此便宣告一個布林變數了，在變數後方各欄提供程式設計者詳盡的變數相關訊息。

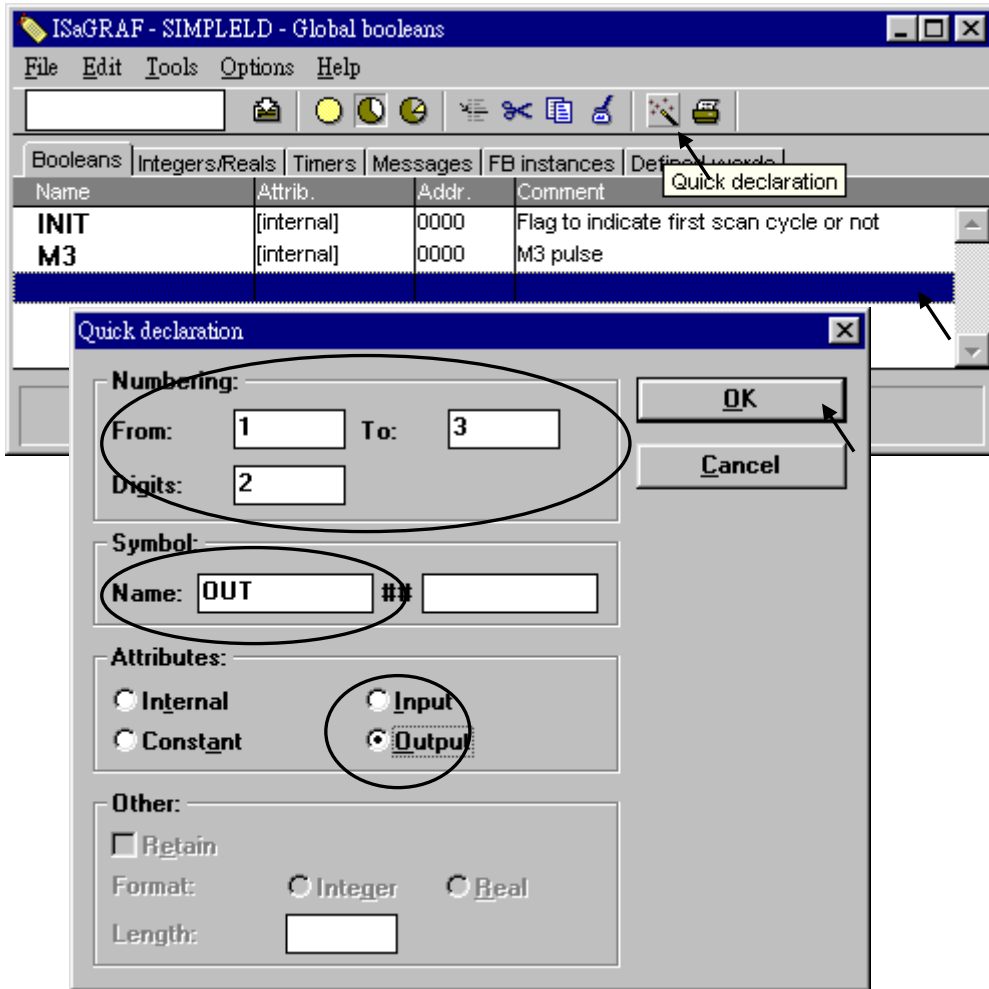


注意：您必須確認您欲宣告的變數有正確的屬性，如果您需要改變變數的屬性，您只需用滑鼠雙擊變數名稱，您便可重新設定變數的屬性。

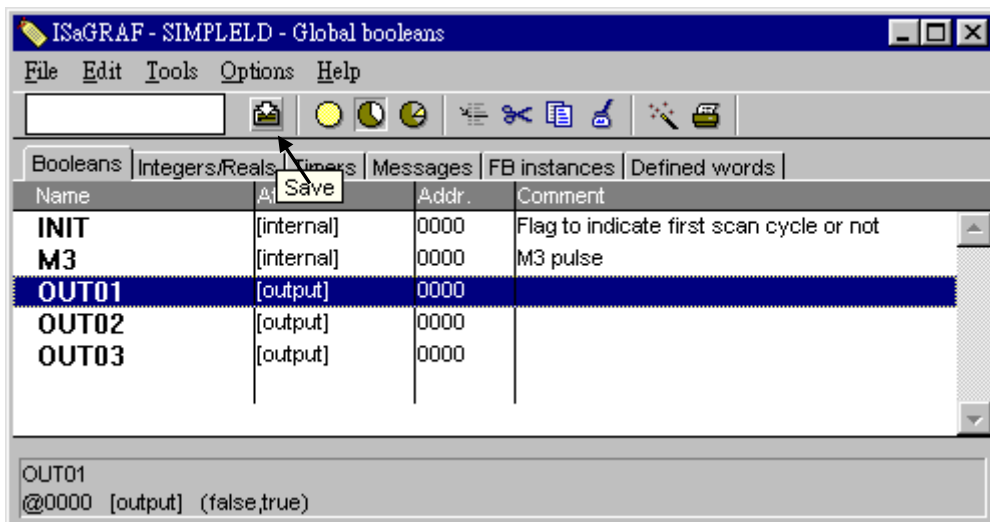
重複上述的步驟您可以完成 "M3" 的宣告，結果會如下圖所示。



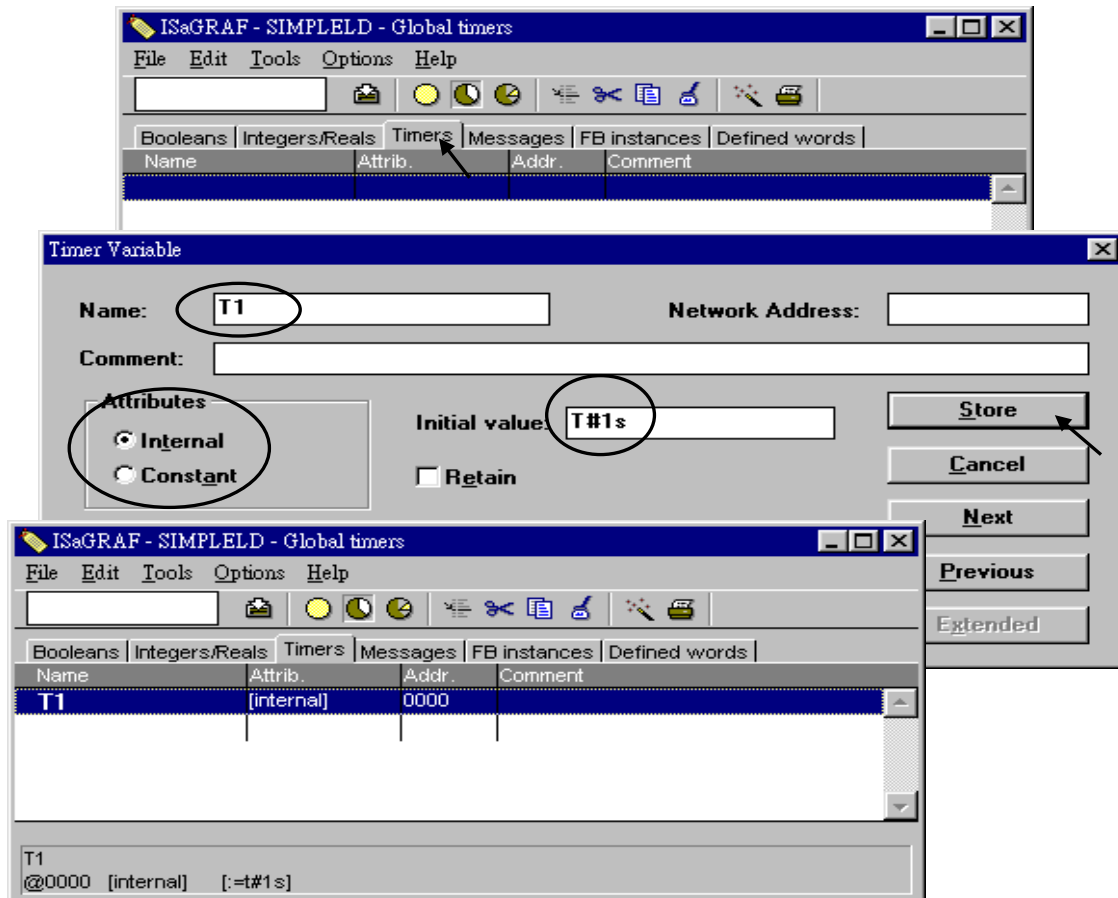
有 3 個輸出變數 "OUT01, OUT02, 和 OUT03" 還沒有宣告，ISaGRAF 提供一種簡便快速的方法宣告這一系列同性質的變數，您如果想使用這種方法只需要用滑鼠單擊 "Quick Declaration" 圖像，並在 "NAME" 欄輸入 "OUT"，在 "Numbering" 內的 "from" 及 "To" 欄輸入序號數 (此範例為 1 到 3)，最後選擇 "Output" 屬性，按下 "OK" 鍵



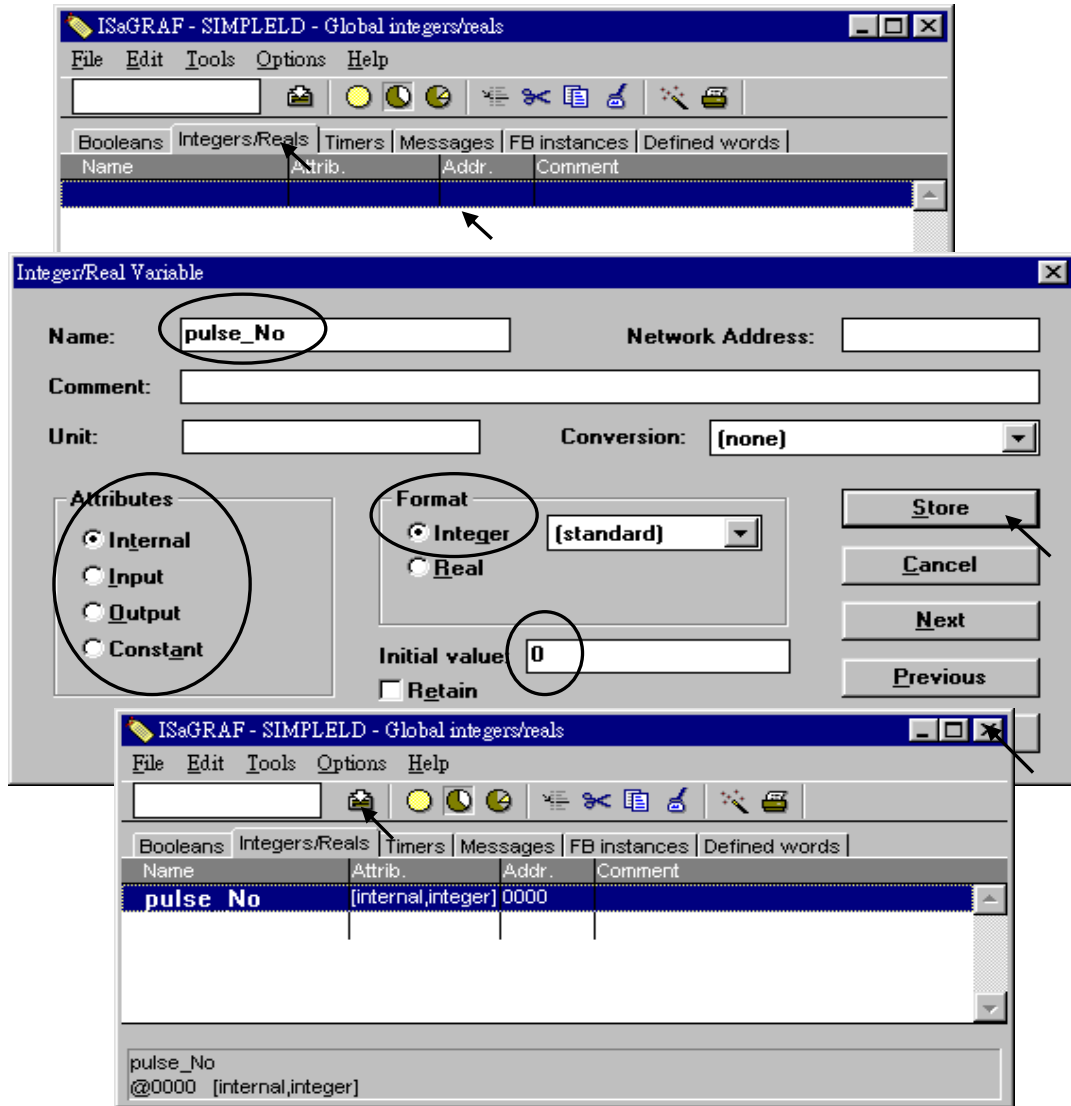
然後你會得到如下的視窗，按 1 下 “Save” 儲存所宣告的資料。



接下來宣告計時器變數(T1)，類似上述的步驟，用滑鼠單擊 Global 設定視窗上的 "Timers" 標籤並雙擊有顏色的區域開啟 "Timer Variable" 視窗，鍵入變數名稱 "T1"，設定屬性為 "Internal"，初始值(Initial Value)設定為 "T#1s"，最後按 "Store" 鍵儲存設定，如下圖所示。



要宣告整數變數 (pulse_No)，用滑鼠單擊 Global 設定視窗上的 " Integers/Reals " 標籤，並雙擊有顏色的區域開啟 " Integers/Reals Variable " 視窗，鍵入變數名稱 "pulse_No"，設定屬性為 "Internal"，格式為 "Integer"，初始值(Initial Value)設定為 "0"，最後按 "Store" 鍵儲存設定，如下圖所示。

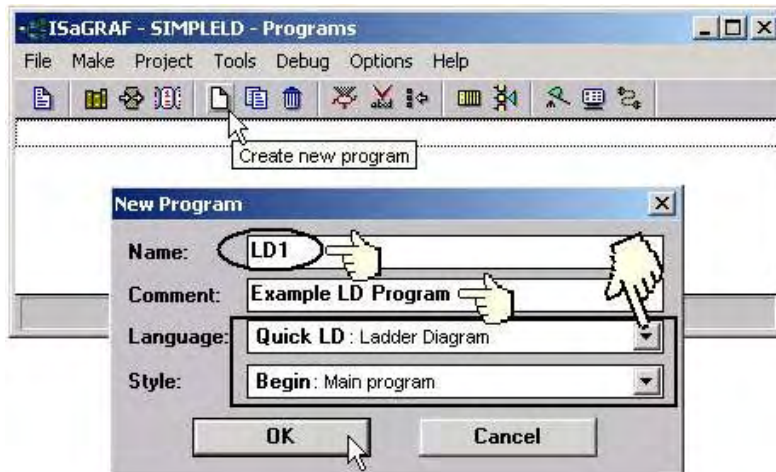


當您設定好全部的變數後您可以點選右上角 "X" 圖像來關閉視窗。

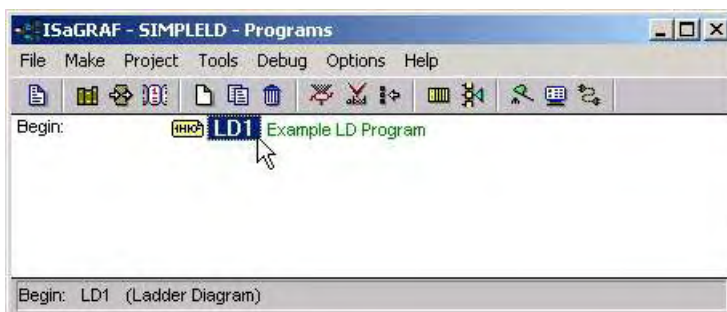
2.1.1.4: 建立 ISaGRAF 範例程式

當您完成所有的變數設定之後，現在可以開始建立 LD 範例程式了，請先點選"Create New Program"圖像，您會見到"New Program"視窗。

首先，在"Name"欄輸入"LD1"(本 LD 範例的名稱)，然後在"Language"欄內選擇"Quick LD: Ladder Diagram"，在"Style"欄內選擇"Begin: Main Program"，您可以在"Comment"內加入註解，不過這不是必須的。

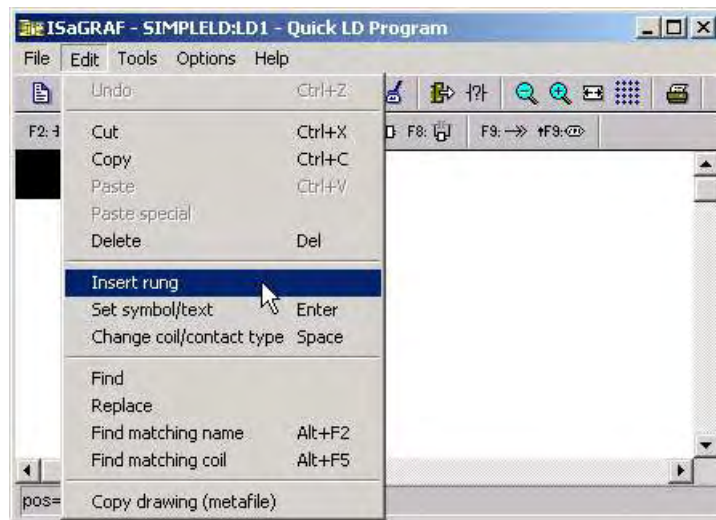


此時您已經建立了一個空的 LD 程式，您可以用滑鼠雙擊"LD1"名稱來開啟程式編輯器。

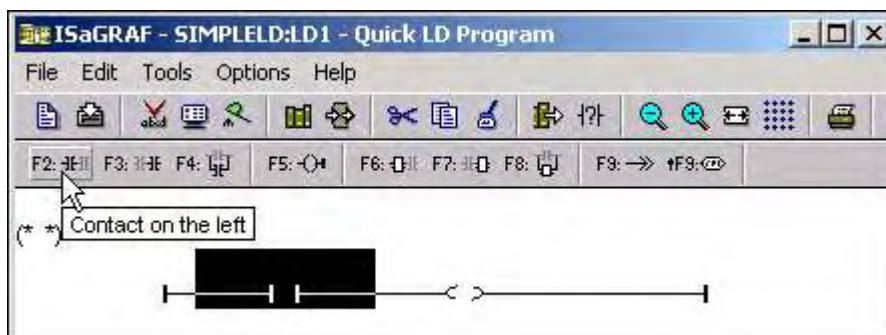


2.1.1.5: 編輯"LD1"範例程式

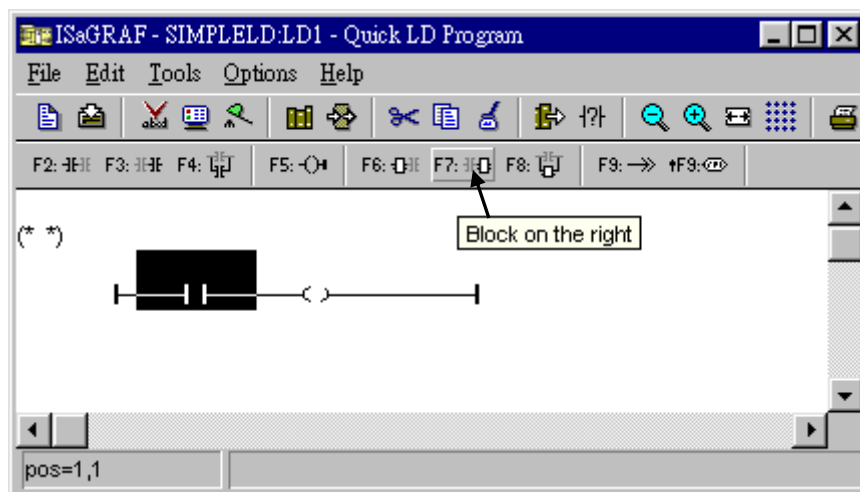
當您用滑鼠雙擊"LD1"後您會看到"Quick LD Program"視窗，如下圖所示，在"Edit"選單中點選"Insert Rung"



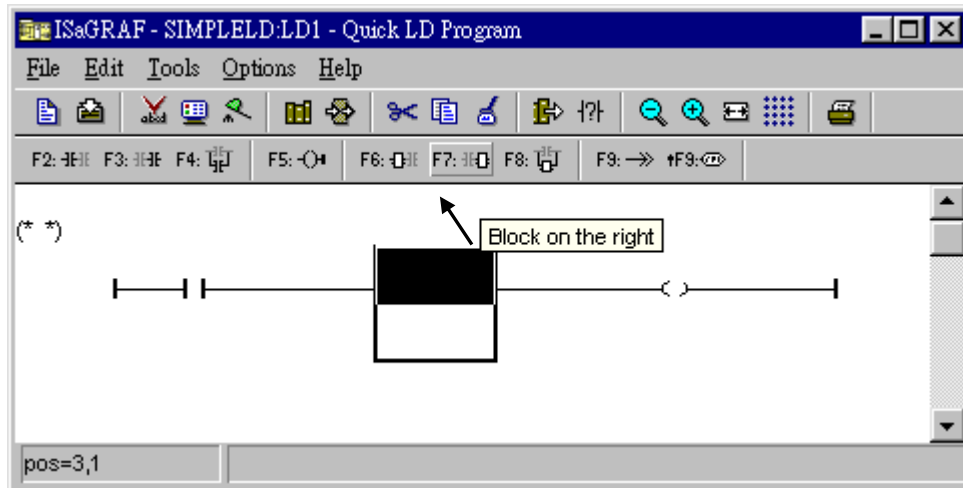
或者，您可以用滑鼠直接點選"F2 (Contact On The Left)"圖像，將會產生如下圖所示的"Quick LD Program"視窗。



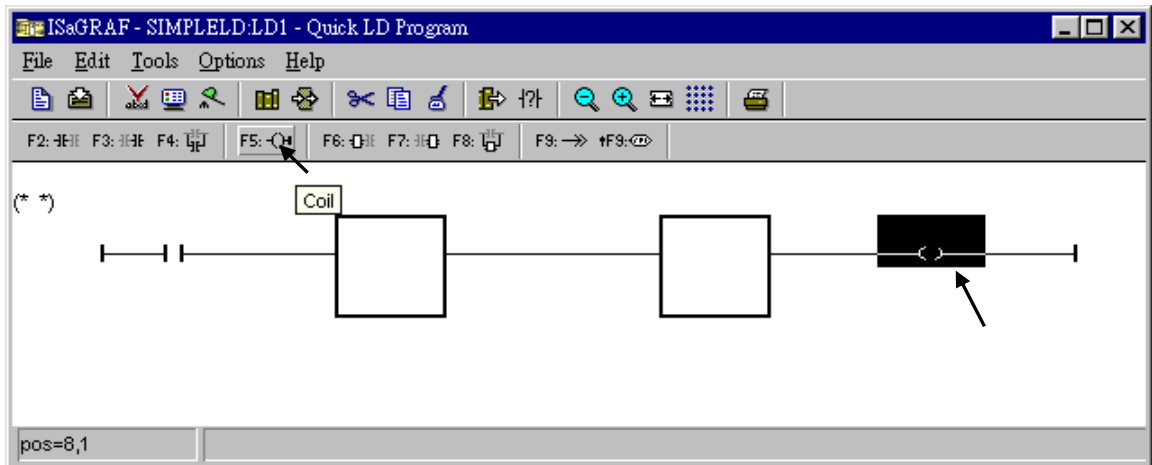
用滑鼠點選 "F7 (Block on the right)", 可以在郵標的右方加入 1 個空的方塊.



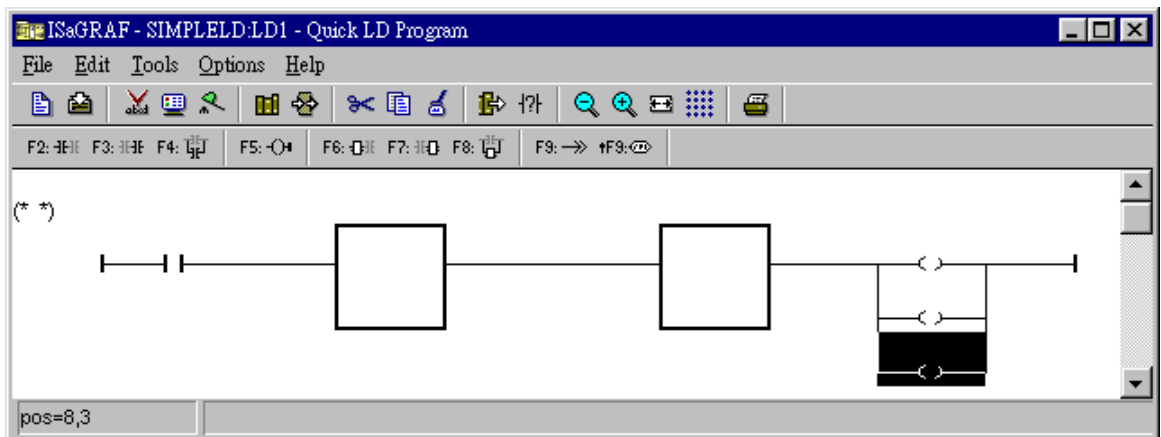
再用滑鼠點選 "F7 (Block on the right)"一次 ,在右方再加入 1 個空的方塊。



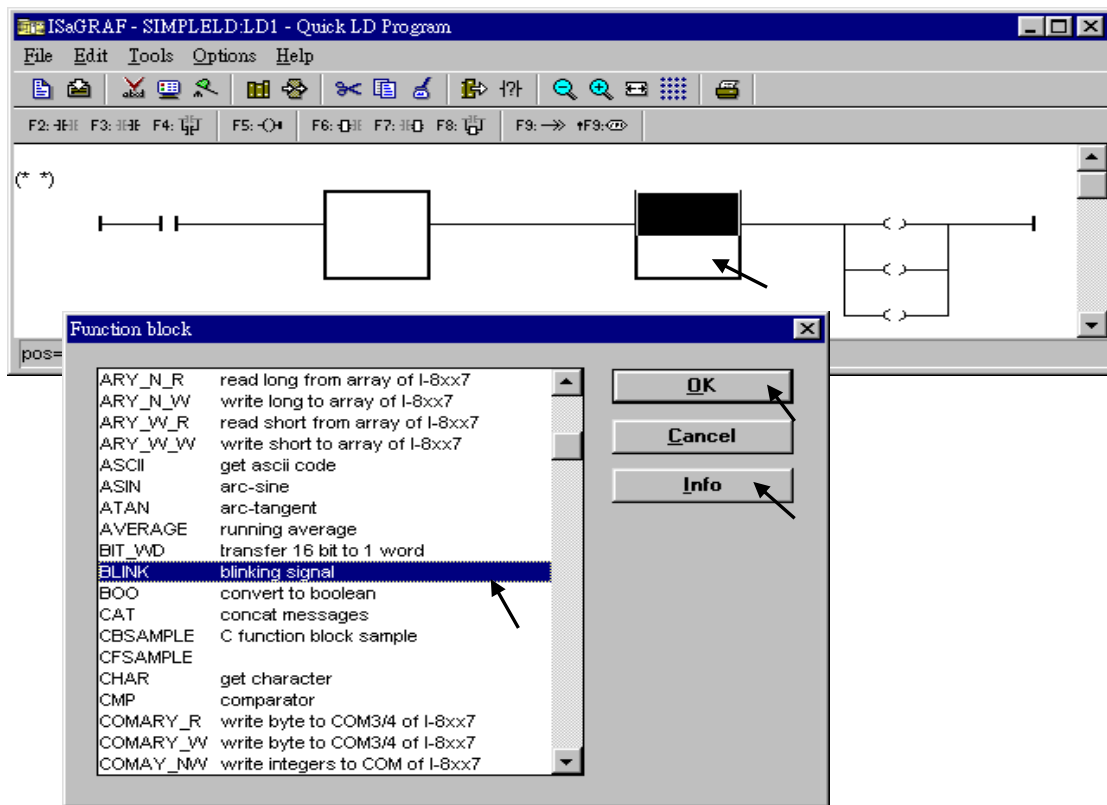
然後得到如下的視窗. 用滑鼠點選右方的 coil 將郵標移到右方. 然點選 “F5 (Coil)” 以增加 1 個 coil, 之後再點選 “F5 (Coil)” 1 次以增加第 3 個 coil.



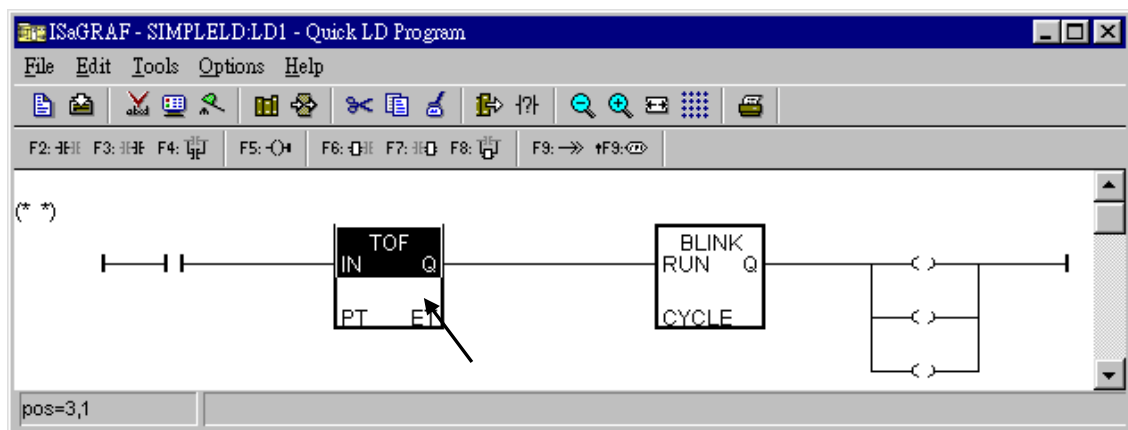
然後得到如下的視窗.



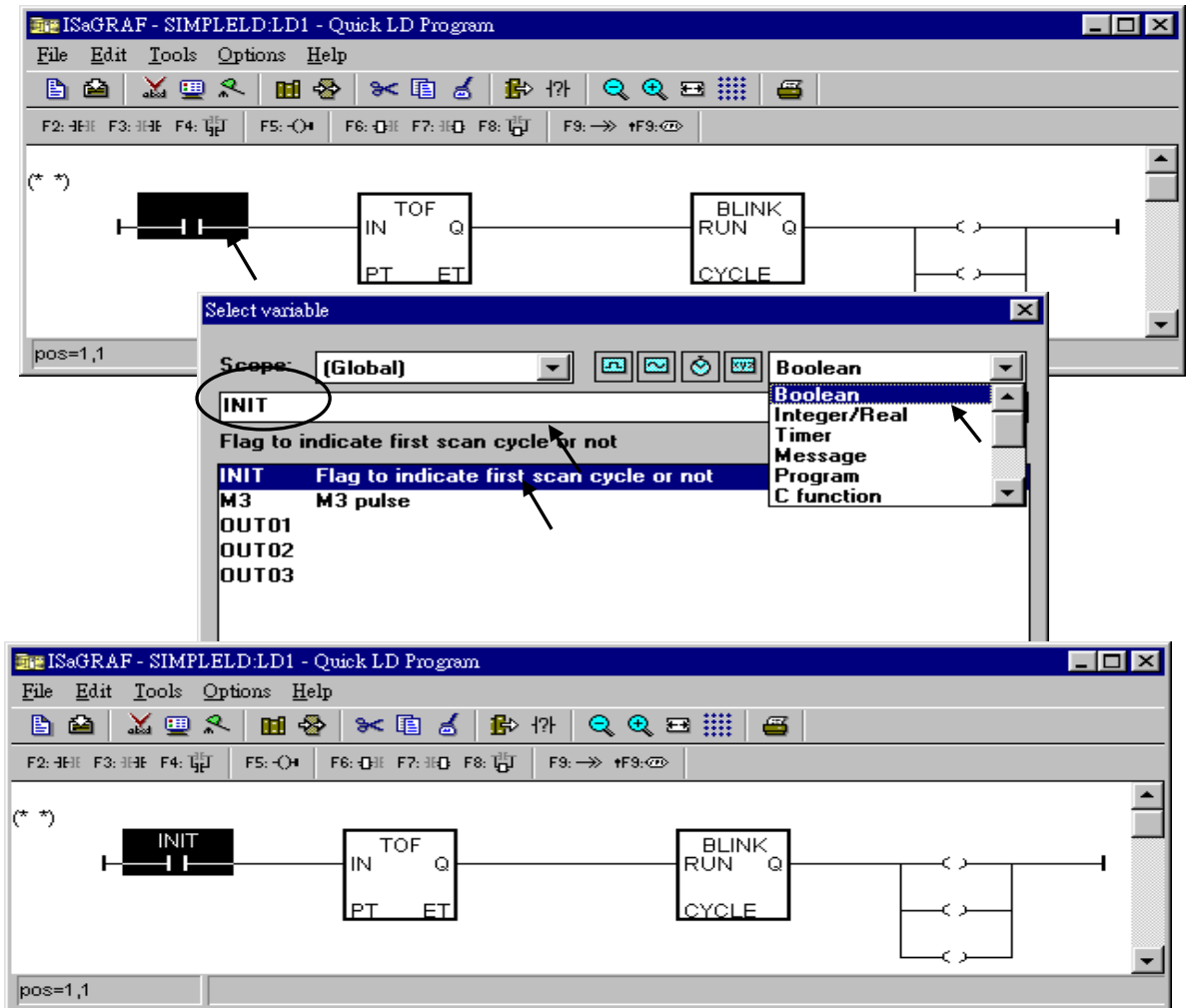
用滑鼠雙擊在右方的方塊，之後會出現 "Function Block" 視窗。 選取"BLINK" 方塊,之後再單擊 "OK"。 若要查看該方塊的說明，可以單擊 "Info"。



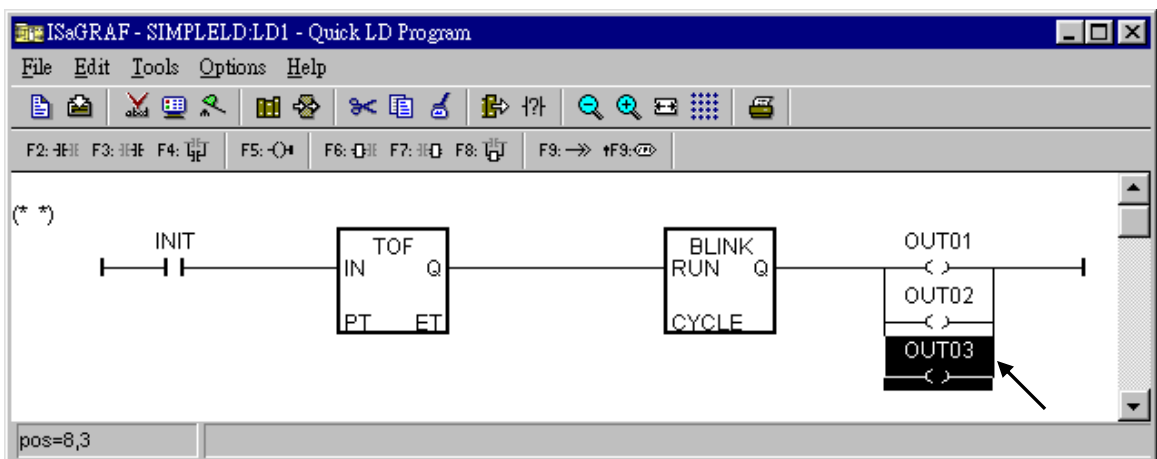
使用同樣的方法，把 "TOF" 指定給左側的方塊，如下圖所示。



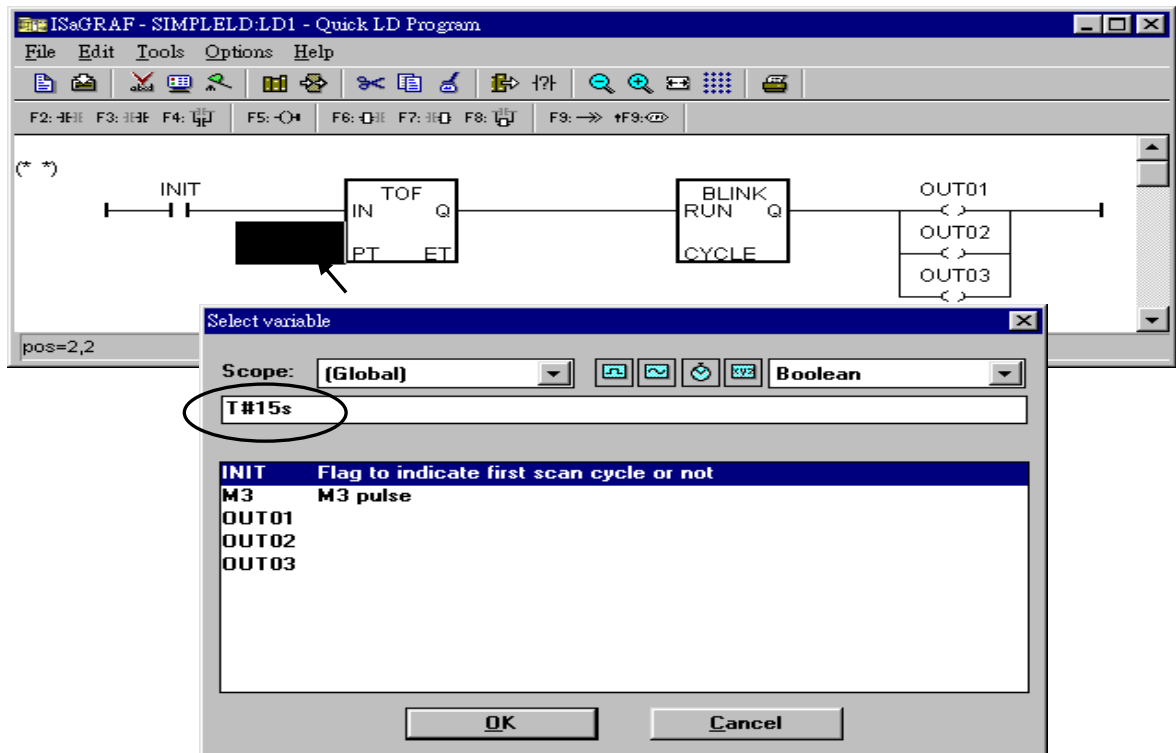
現在我們要開始指定變數或常數給每個元件. 用滑鼠雙擊左側的 Contact, 之後 “Select variable” 視窗會跑出來. 先選取 “Scope” 為 “(Global)”, 再選取 “Boolean”. 之後雙擊 “INIT” (你也可以用鍵盤輸入).



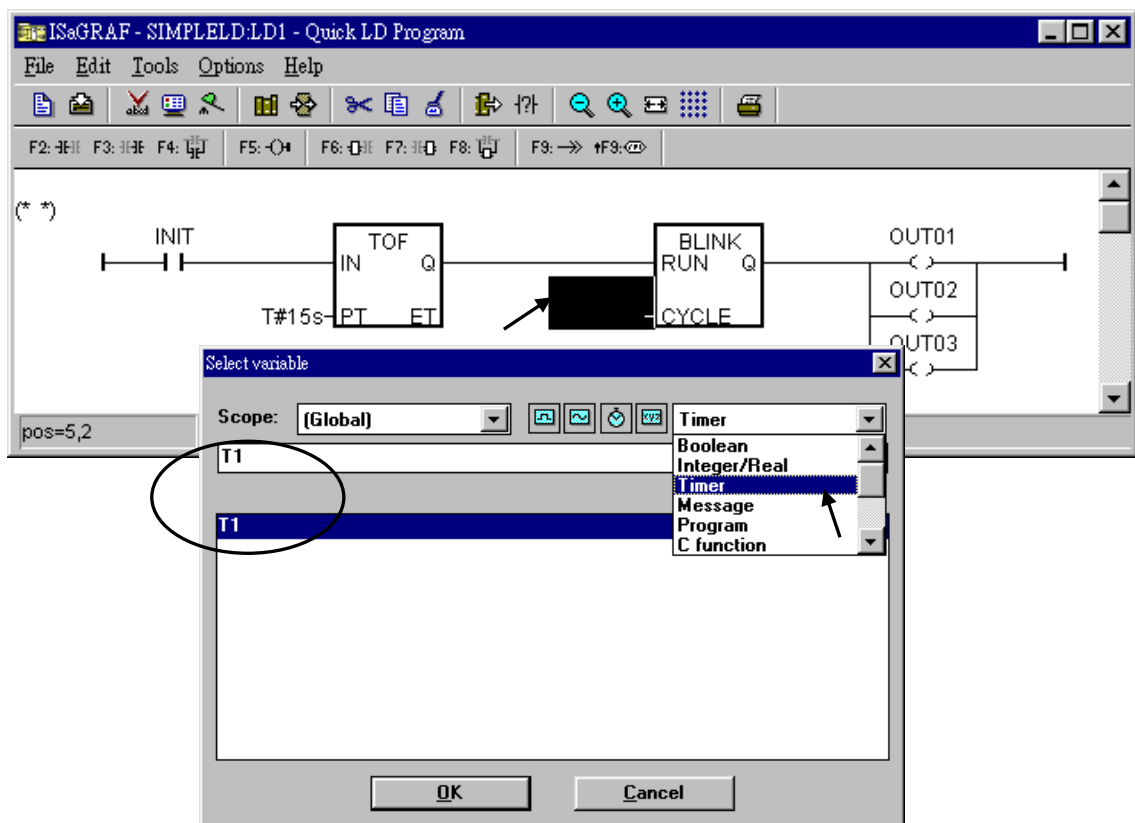
使用同樣的方法指定 OUT01~OUT03 到個別對應的 Coil, 如下圖所示.



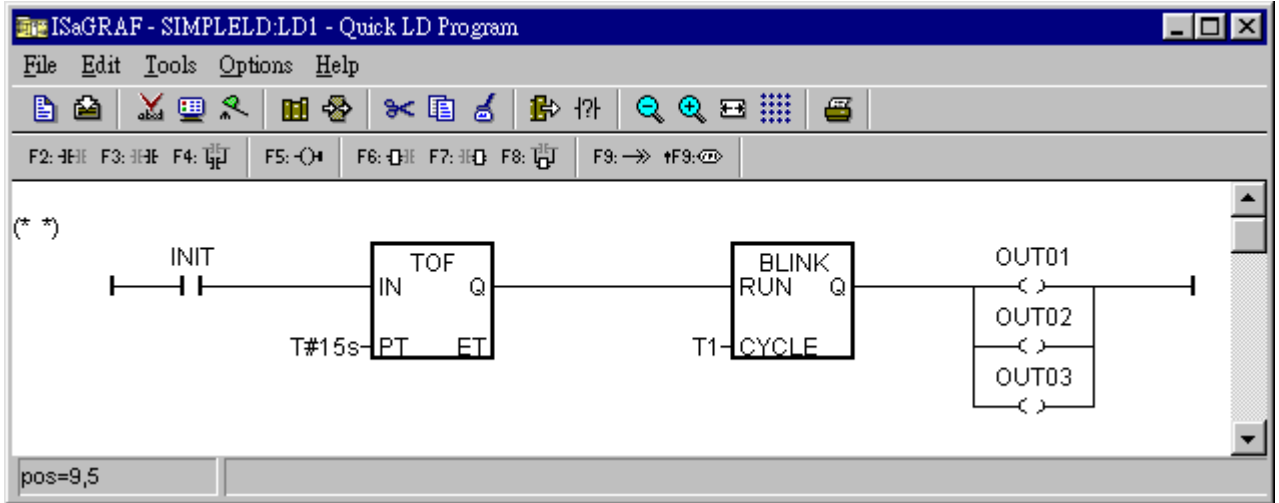
用滑鼠移動郵標到 ”TOF” 方塊 ”PT” 參數的左邊一格，雙擊該位置，用鍵盤輸入 T#15s (表示 15 秒)，然後按下 “OK”。



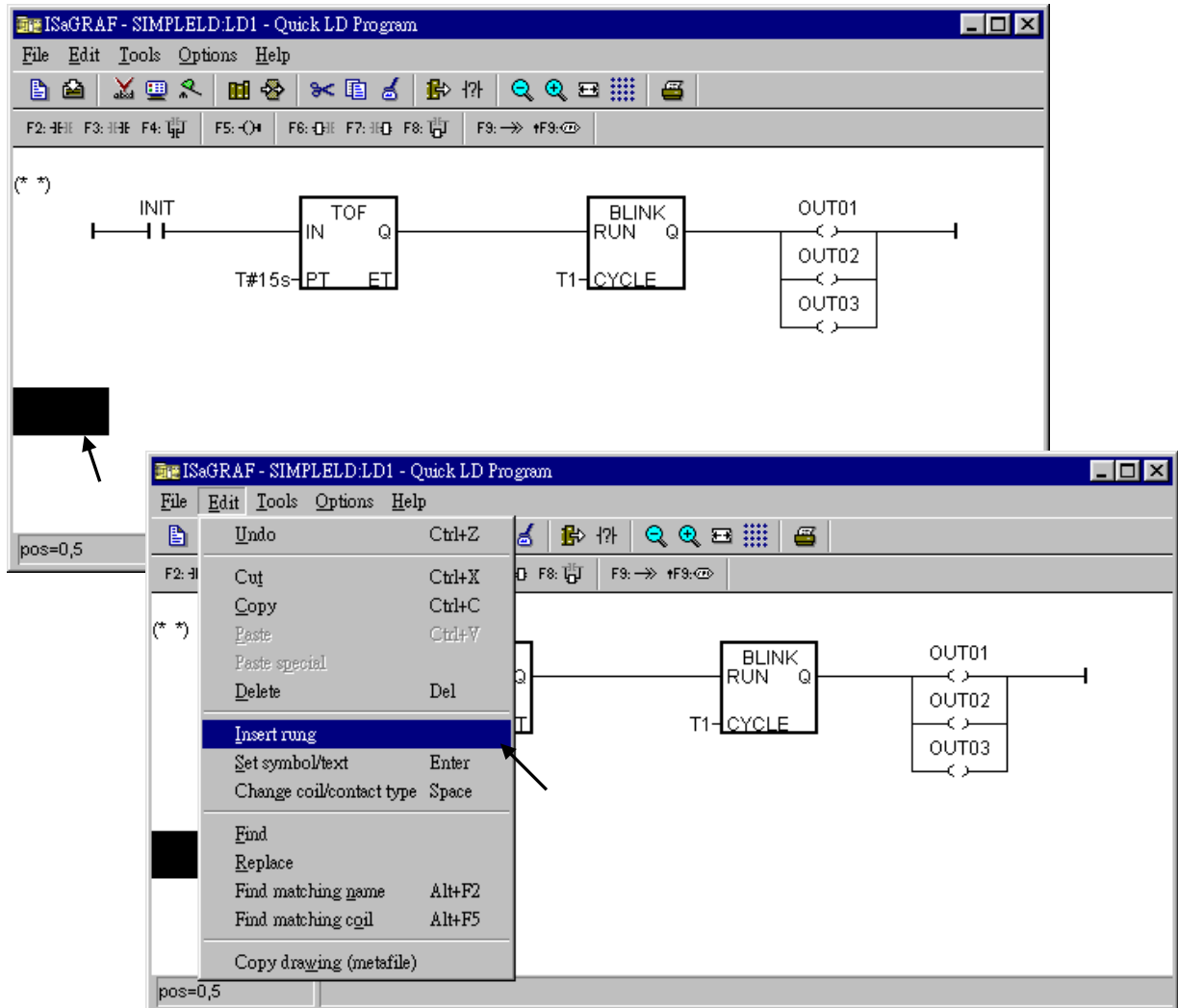
以同樣的方法指定 T1 給 “BLINK” 方塊的 “CYCLE” 參數。



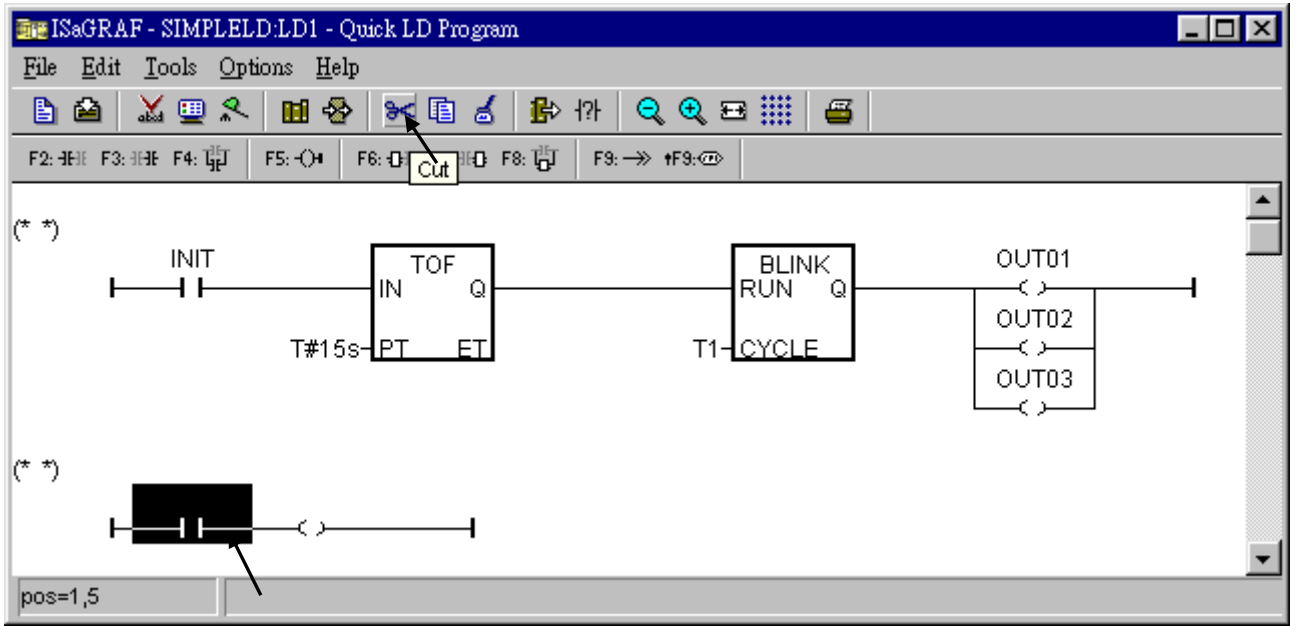
此時視窗會如下所示。



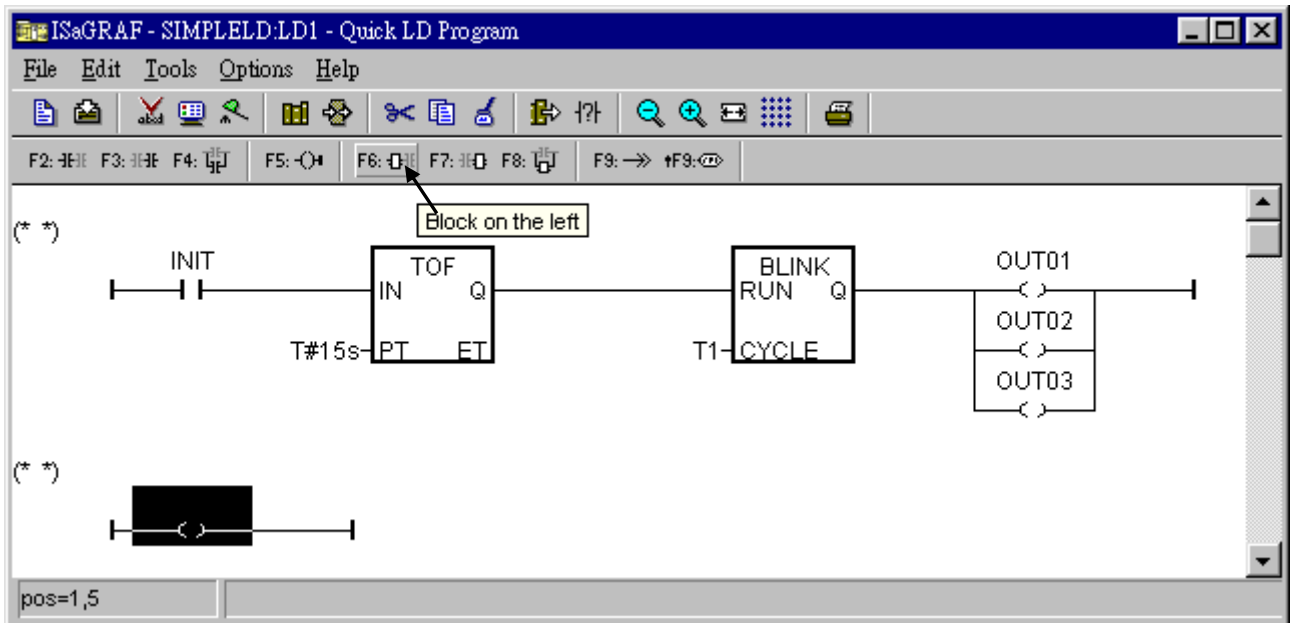
接下來要在下方新增一行，先用滑鼠移動郵標到下方的適當位置，之後點選“Edit – Insert rung”



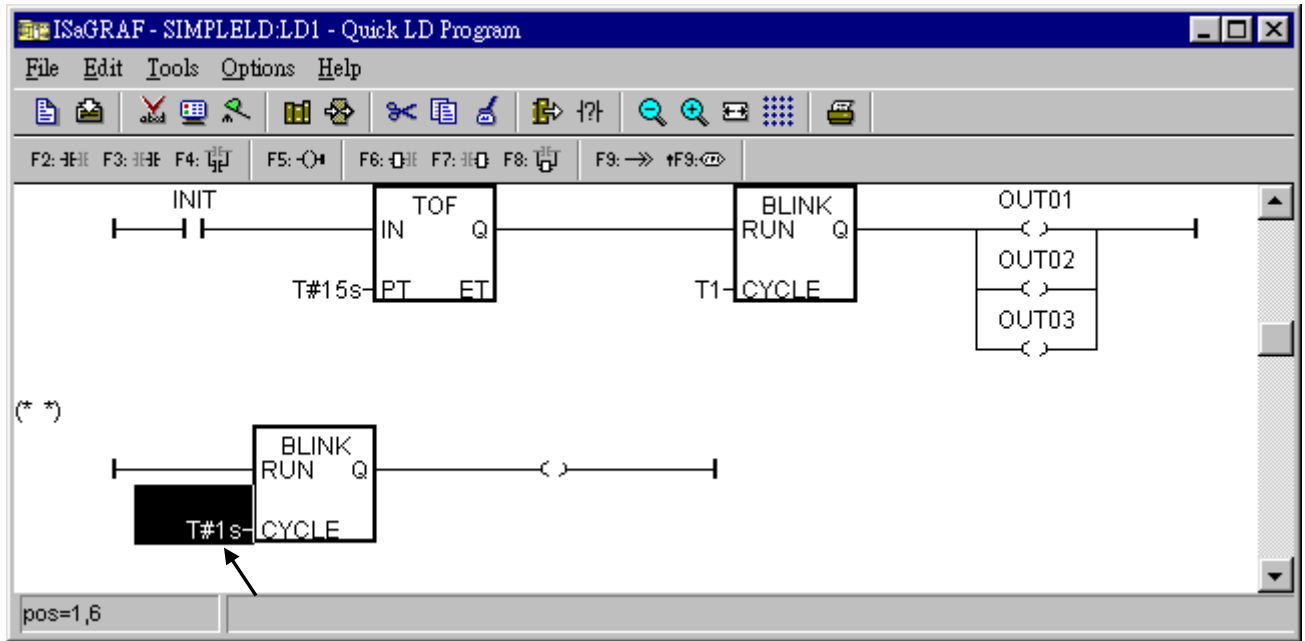
我們在這個範例內不須要左側的 Contact，先用滑鼠點選它，然後再點選“Cut”把它刪掉。



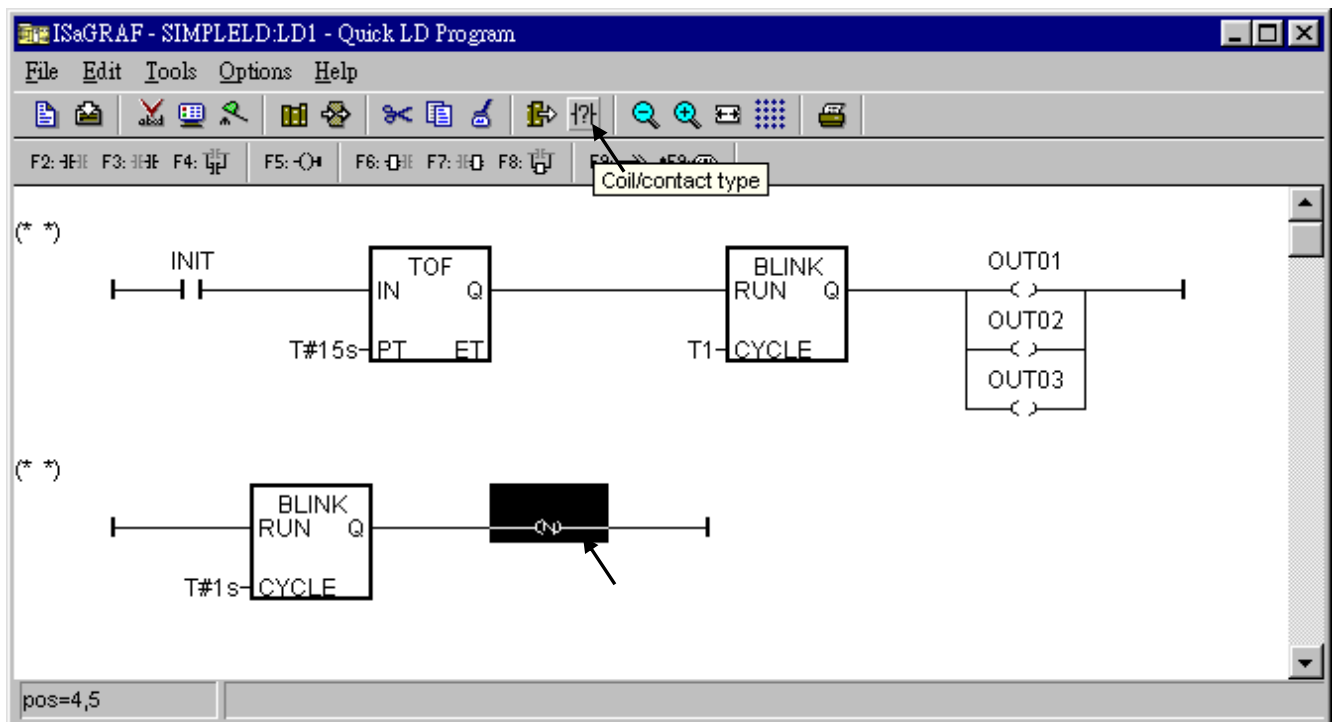
點選“F6 (Block on the left)”，之後雙擊新增的方塊，並指定“BLINK”給該方塊。



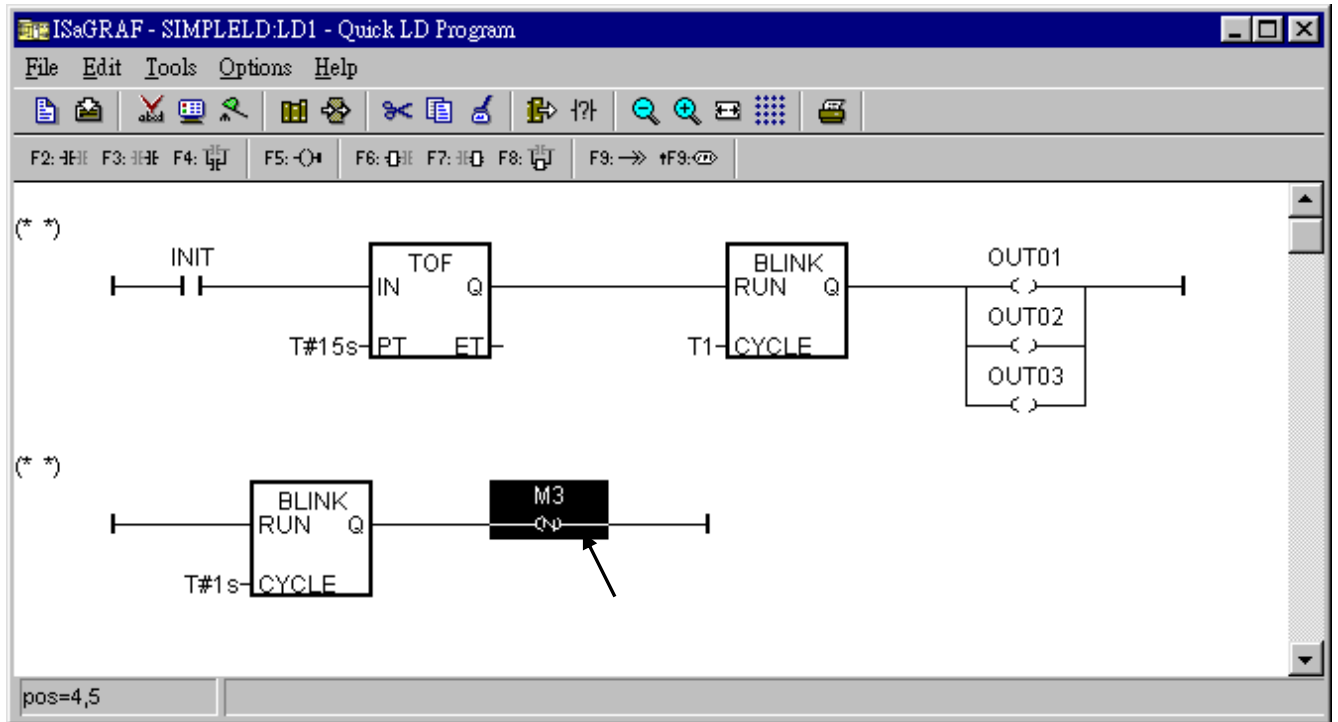
指定 T#1s 給 “BLINK” 方塊的 “CYCLE” 參數, 之後會得到如下的視窗.



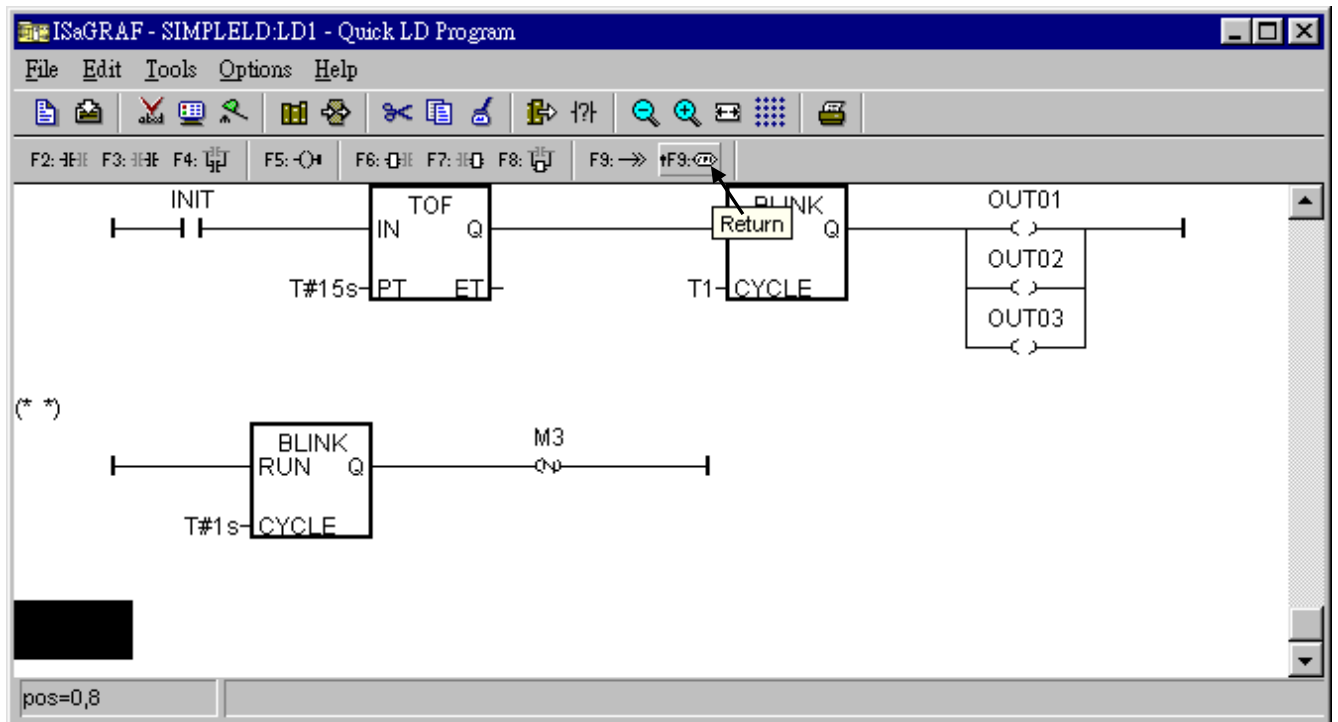
移動游標到右方的 Coil, 然後點選 “Coil/contact type” 幾次來切換 Coil 到 型態 “N”.



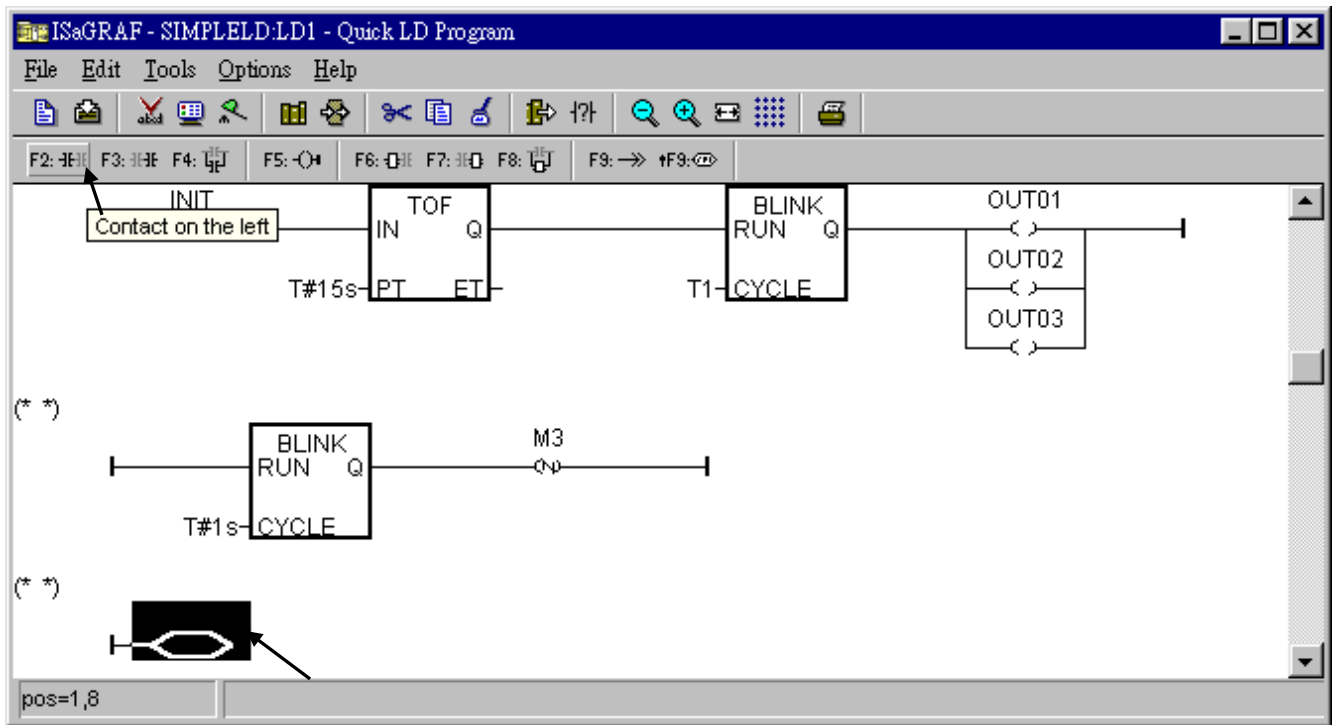
用滑鼠雙擊 “N” Coil 以指定 “M3” 給它.



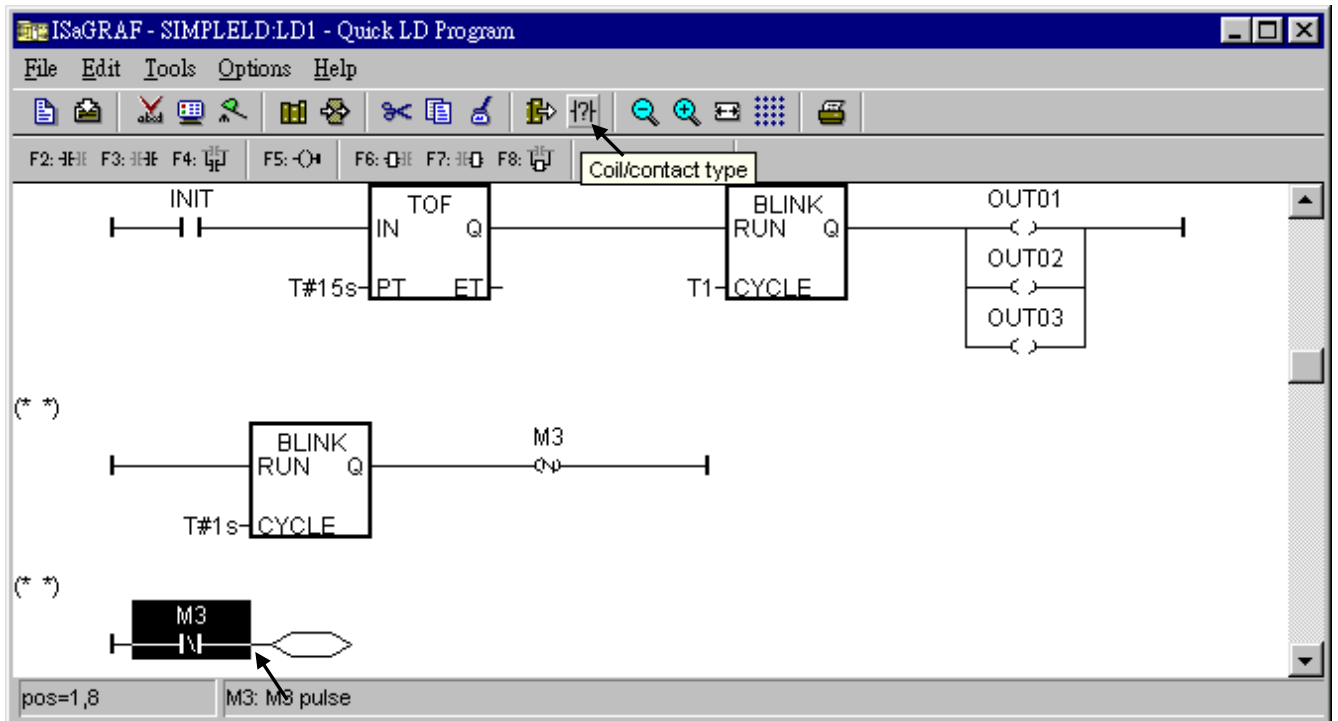
現在，我們還要新增另一行，用滑鼠移動郵標到下方的適當位置，然後點選 “F9 (Return)”。



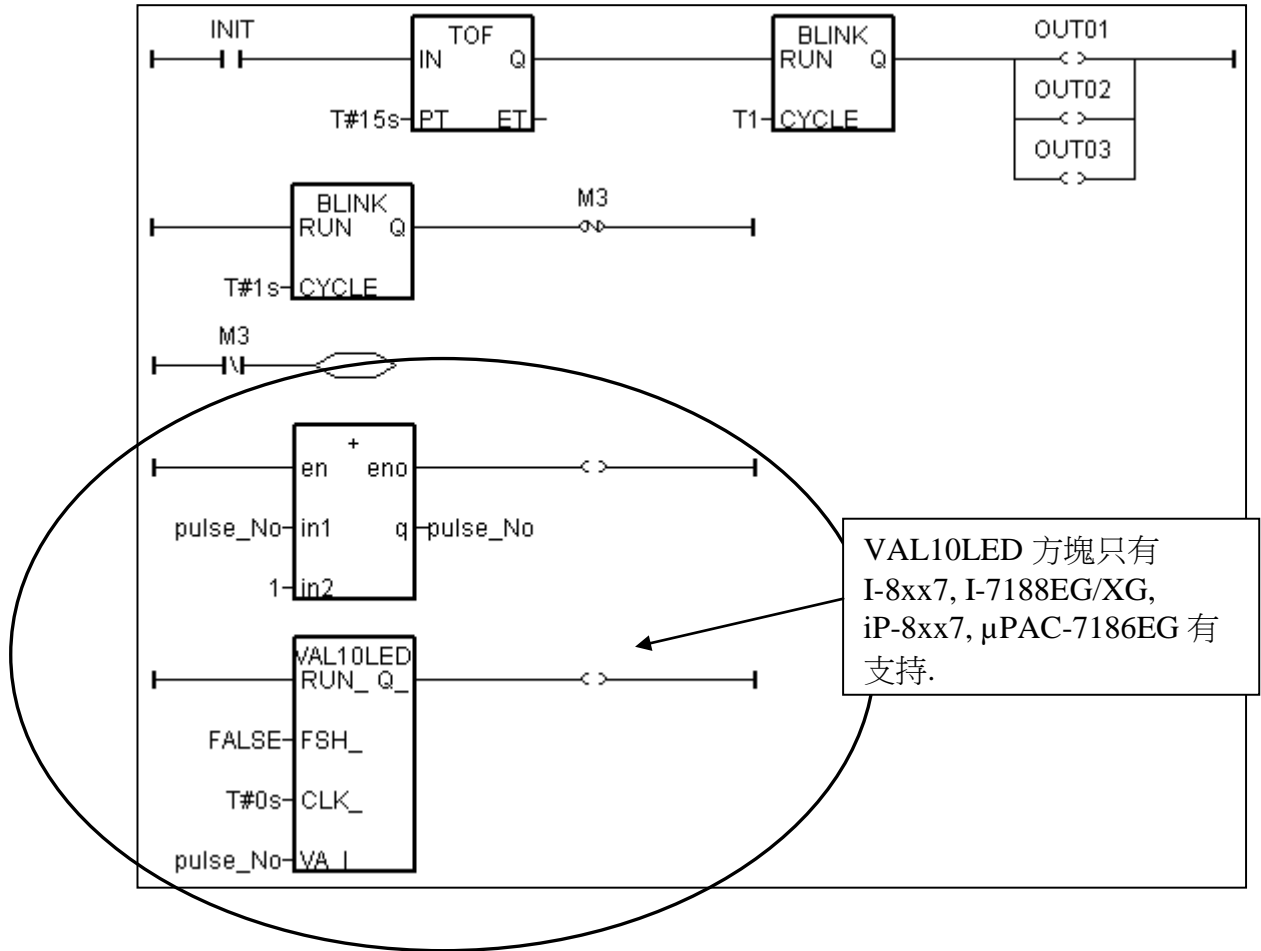
移動游標到 “return”，之後點選 “F2 (Contact on the left)” 以在左邊增加一個 Contact.



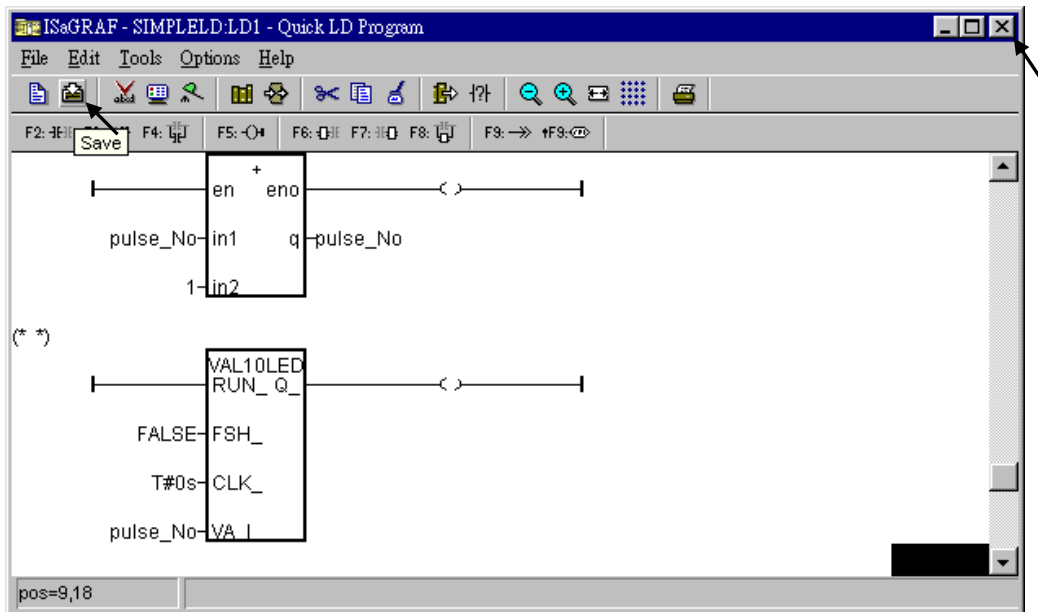
用滑鼠雙擊該 Contact，之後指定 “M3” 給它，並更改它的型態為 “∩” (反相 Contact).



接下請自行依照先前描素的方法新增第 4 及第 5 行，完成後的 LD 程式需如下所示。

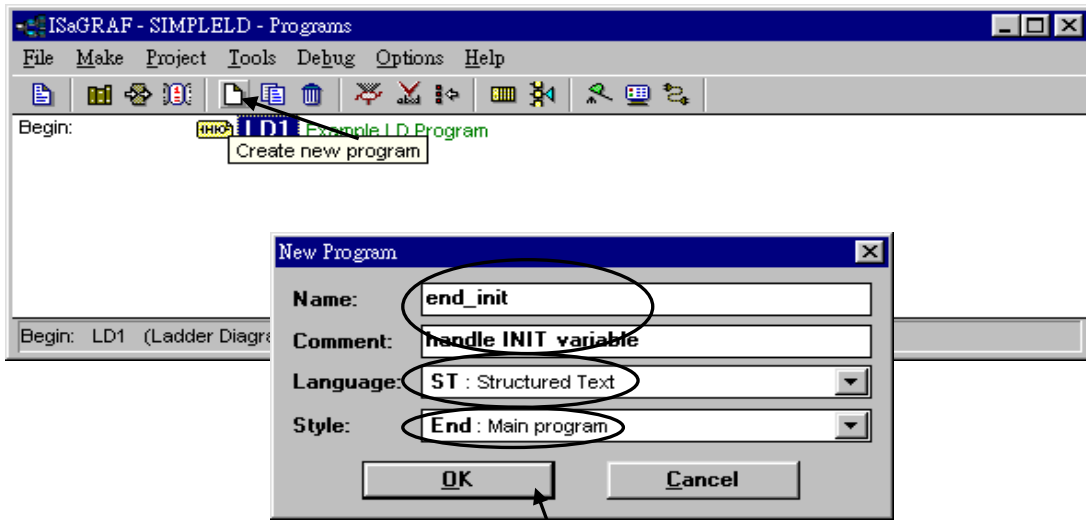


儲存這個 LD 程式，之後關閉它。

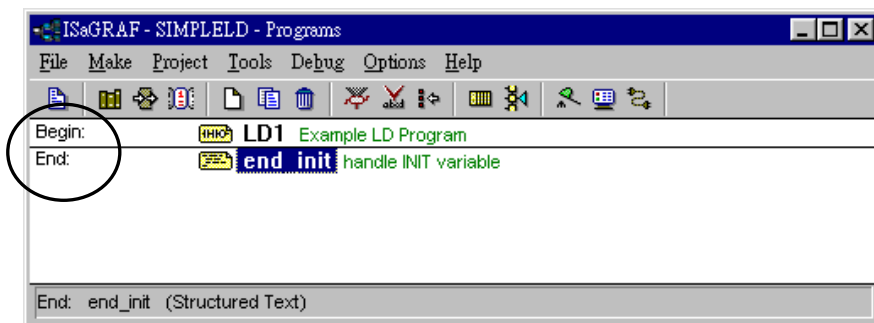


2.1.1.6: 建立 ST 程式 "end_init"

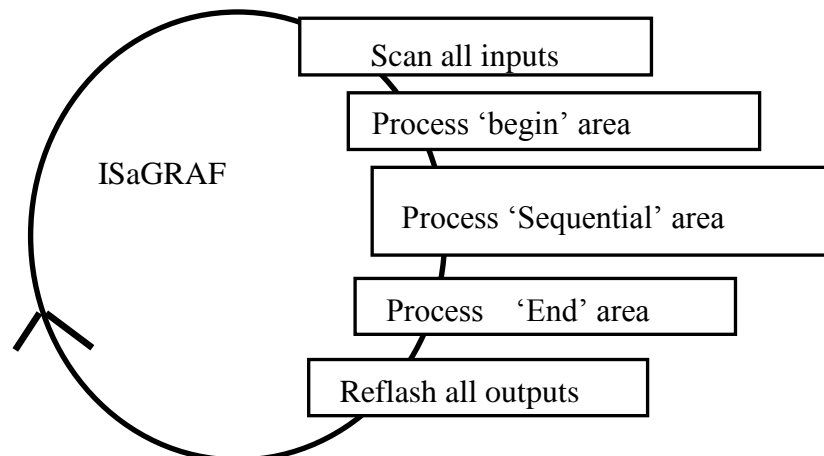
在這個專案內,我們需要一個 ST 程式來處理 INIT 變數,以滑鼠點選 “Create new program” 以建立一個新程式. 輸入程式名稱為 “end_init”, 注解為 “Handle INIT variable”, Language 為 “ST: structured Text”, Style 為 “End: Main program”. 之後點選 “OK”.



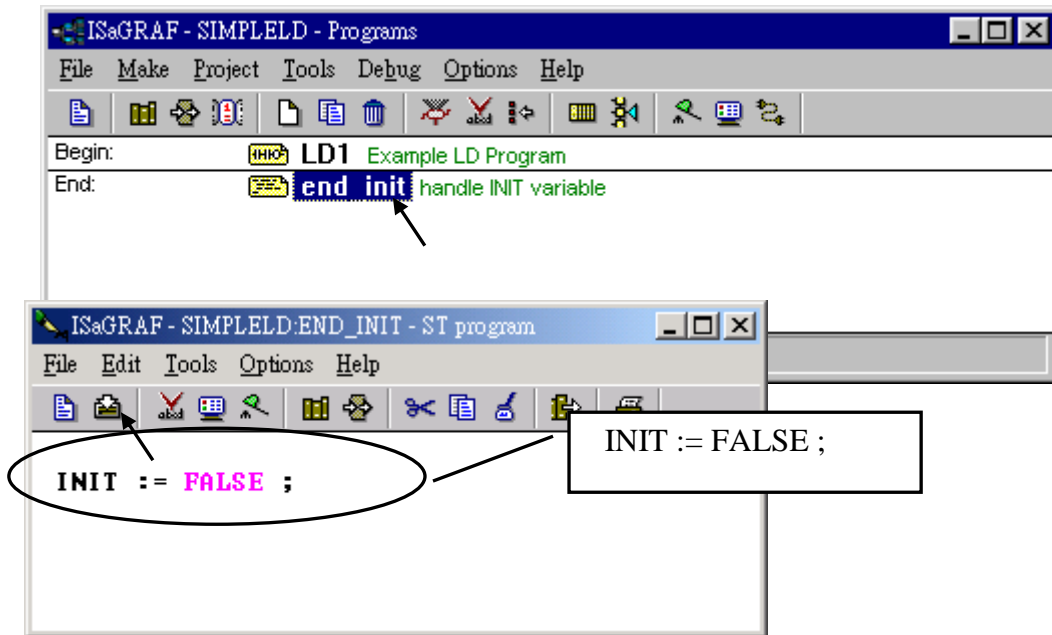
然後我們可以看到有 2 個程式在這個專案內.



ISaGRAF 每個 Scan 周期都會執行這兩個程式各 1 次. 其運作原則為先執行 “begin” 區內的程式, 之後執行 “Sequential” 區, 最後再執行 “End” 區. 一個 ISaGRAF Scan 周期會如下圖的方式運作.



用滑鼠雙擊 “end_init” 程式. 用鍵盤編好 ST 程式後, 點選 “save” 然後關閉它.任何出現在(*與 *) 內的文字為註解.

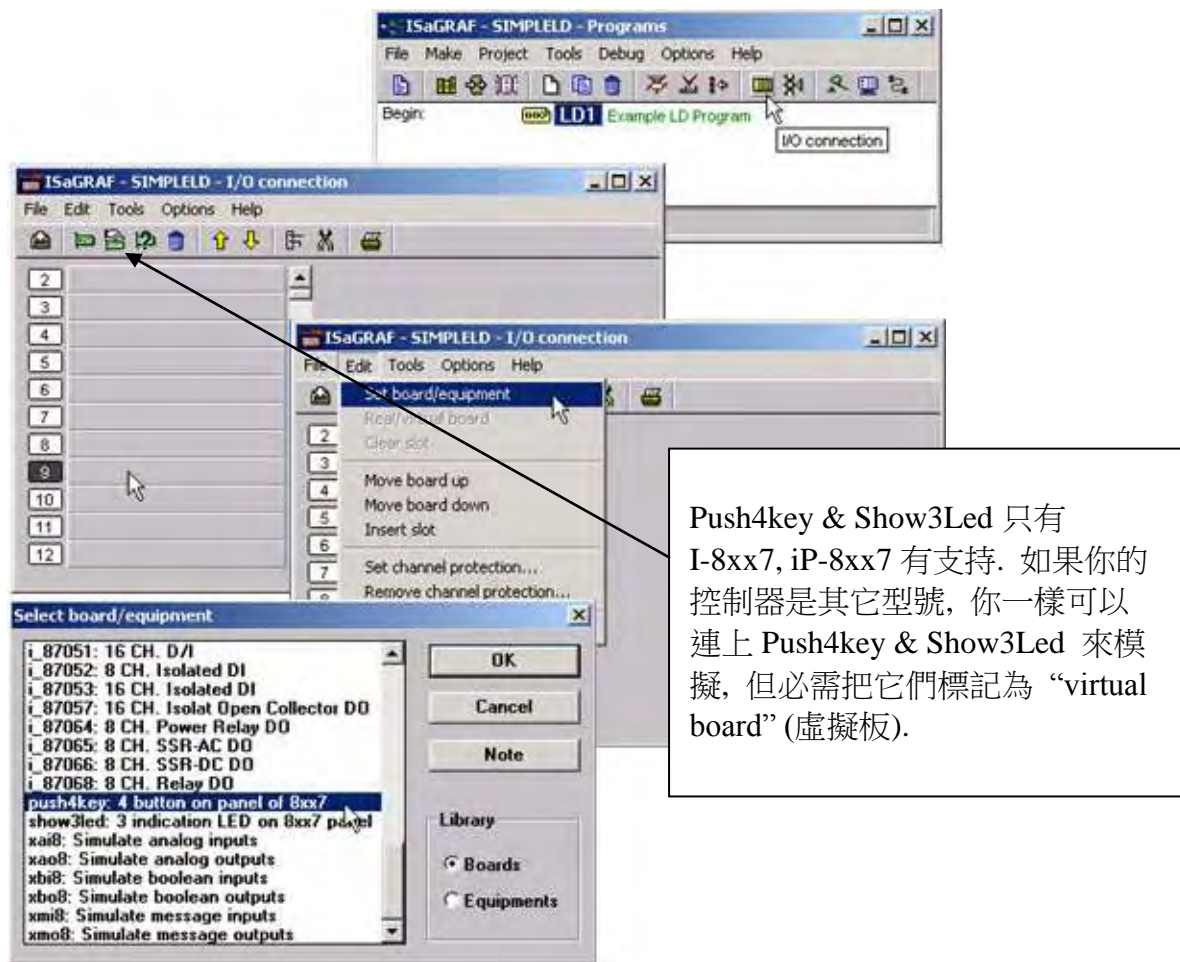


因為 “INIT” 被宣告成初值為 “TRUE”, 這個 ST 程式會在第 1 個周期的尾端把 “INIT” 設為 FALSE 之後它的值就一直是 FALSE. 也就是說, “INIT” 可用來判定目前程式是不是才剛開始執行 (TRUE: 第 1 個 Scan 周期, FALSE: 其它周期).

2.1.2: 連接輸入輸出(I/O)設備

由於 ISaGRAF 是一個開放式的軟體，很多控制器及 PLC 皆支援 ISaGRAF，然而，為了要讓這些 PLC 控制器能夠使用 ISaGRAF 程式，首先這些 PLC 控制器的硬體製造商必須將 ISaGRAF 的驅動程式嵌入控制器中，ICP DAS 全系列的 ISaGRAF PAC 皆有將 ISaGRAF 驅動程式嵌入，使其成為強大的、具高可塑性的工業用控制器。

當您完成您的 ISaGRAF 程式後您必須將這個範例程式與 PAC 的 I/O 控制系統連接，而 ICP DAS 的 ISaGRAF 控制系統有一項方便好用的特色可以讓您不需要實際插上 I/O 模板便可進行測試，I-8xx7, iP-8xx7 控制器上面板的 4 個按鍵可以作為 4 個數位輸入，面板上方的 3 個 LED 燈號可作為輸出的訊號。



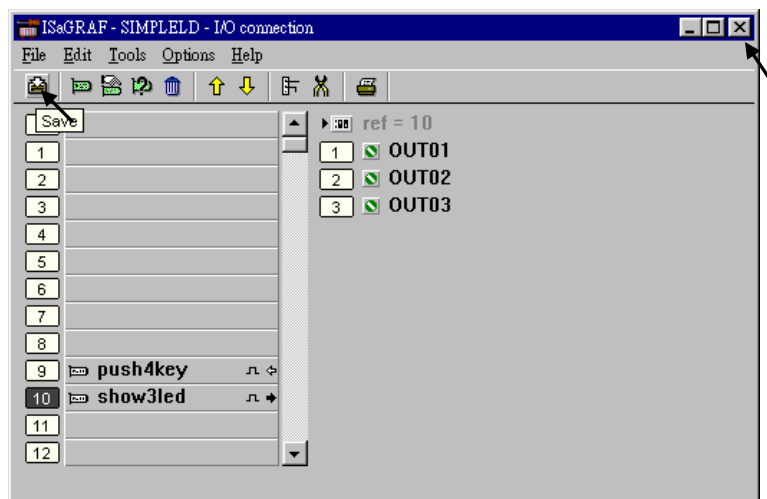
當您用滑鼠單擊"I/O Connection"圖像之後便會出現"I/O Connection"視窗，以本範例為例，您可以用滑鼠雙擊 9 號插槽，或者單擊 9 號插槽然後單擊[Edit]->[Set Board/Equipment]，也可以叫出"I/O Connection"視窗，點選"push4key:4 button on panel of 8xx7"連結以控制器面板上的 4 個按鍵作為輸入的 I/O Board (本範例並未用到這 4 個按鍵)。

注意：編號 0 到 7 的 I/O 插槽是保留給實際的 ISaGRAF 控制器 I/O 模板用的，您可以使用編號 8 以後的插槽連結虛擬 I/O 板。

為了建立 I/O 連結您可以用滑鼠雙擊 10 號插槽並點選"Show3led: 3 indication LED on 8xx7 panel"連結控制面板上的 3 個 LED 作為輸出，此時您的"I/O Connection"視窗應如下圖所示。



當您完成 I/O 連結後記得點選"SAVE"來儲存。



注意:為了能夠成功地編譯,所有輸入、輸出屬性的變數必須經由上述的方法進行 I/O 連結,而且,也只有輸出,輸入變數才會出現在"I/O Connections"視窗。

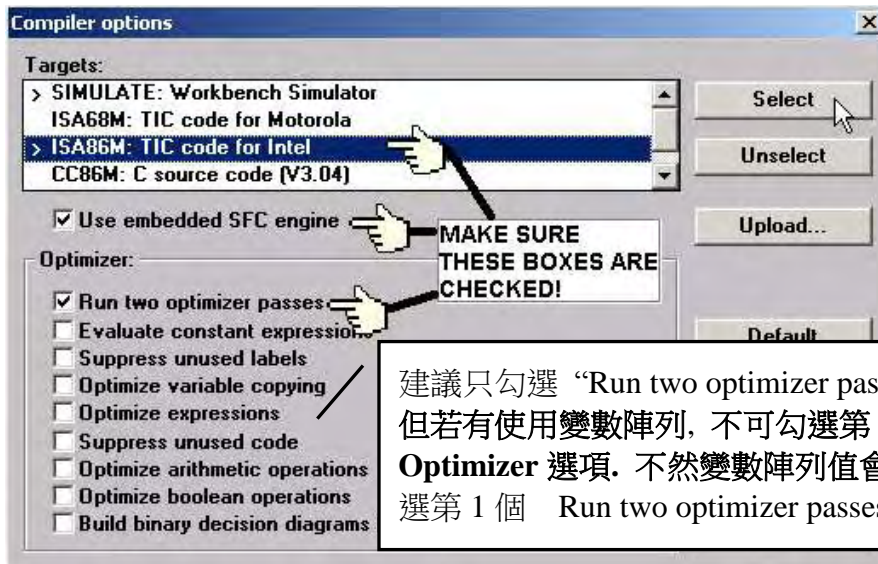
2.1.3: 編譯 LD 範例程式

為了讓所有的 ISaGRAF 程式都能夠在 ISaGRAF 控制器上正常的運作，程式設計師必須選擇正確的"Compiler Options"編譯選項，您必須選擇 "ISA86ASM: TIC Code For Intel" 選項，詳細步驟如下所示。

首先，用滑鼠在選單上點選"MAKE"選項，然後如下所示點選 "Compiler Options"。



此時您會見到如下圖所示，請確認您正確地選擇如下圖所示的選項，並按下"OK"鍵完成所有的選擇。



編譯時的訊息處理!

當您點選"Make Application Code"圖示後便開始編譯此 LD 範例程式，如果編譯過程沒有出現任何錯誤訊息，恭喜您，您已經成功地完成了一個 LD 範例程式。

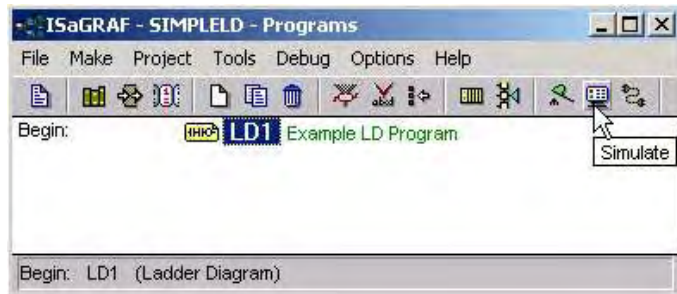
如果您在編譯過程中程式出現錯誤訊息，請點選"CONTINUE"按鍵檢視錯誤訊息代表的意義，並回到專案編輯畫面更正錯誤，並重新編譯。



2.1.4: 模擬 LD 專案

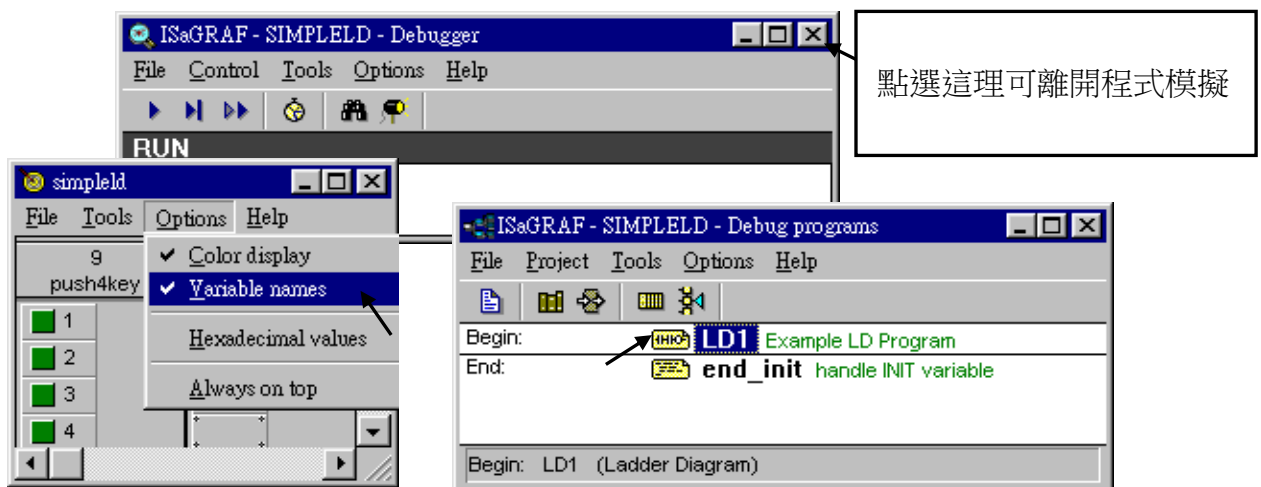
ISaGRAF 其中一項強大的功能在於它強大的除錯功能，在您還沒實際將專案下載至 ISaGRAF PAC 中，您可以利用"SIMULATE"(模擬)的功能進行除錯。

在成功地編譯完成了 LD 程式之後，如下圖所示點選"SIMULATE"圖示。



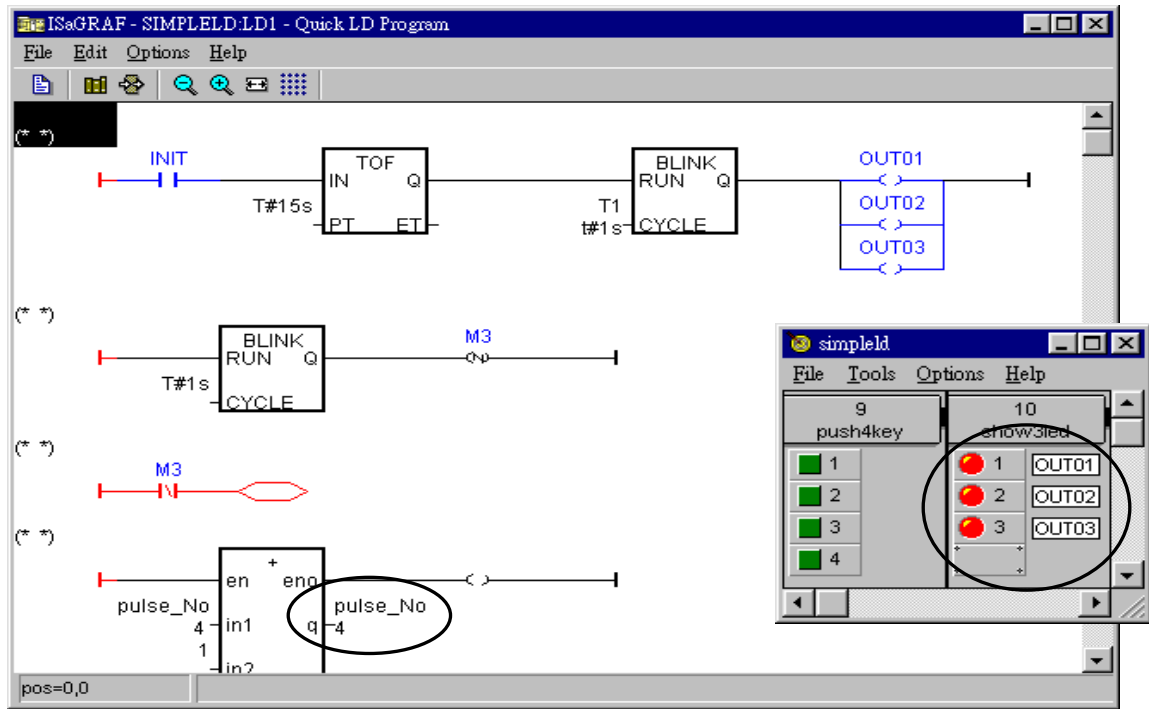
當您點選"Simulate"圖示後會出現 3 個視窗，分別是"ISaGRAF Debugger"、"ISaGRAF Debug Programs"和 "I/O Simulator"三個視窗，如果您建立的 I/O 變數名稱沒有出現在"I/O Simulator"視窗，只需要點選"Options"和"Variable Names"選項便可以在"I/O Simulator"視窗中顯示變數名稱。

如下圖所示，在"ISaGRAF Debug Program"視窗中雙擊"LD1"，可以打開 ISaGRAF 的"Quick LD Program"視窗，並看到您先前所設計的程式。



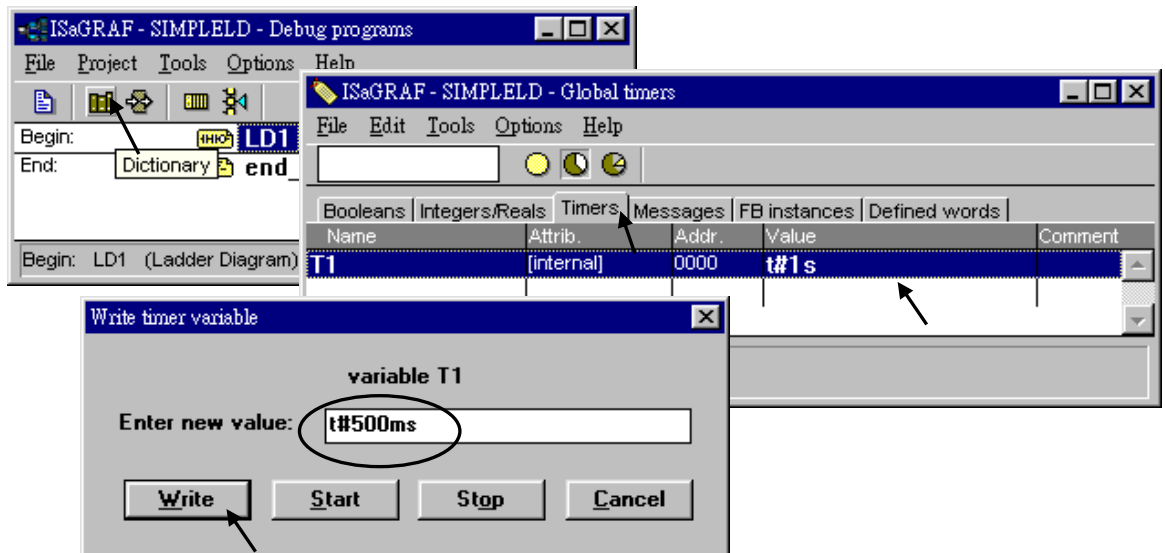
執行模擬程式

當您在"ISaGRAF Debug Programs"視窗中用滑鼠雙擊"LD1"後會見到如下畫面。



你會看到 OUT01 ~ OUT03 在程式開始執行的前 15 秒會以 1 秒為周期閃爍. 而且 pulse_No 的值會每秒加 1.

您可以在程式進行時調整 T1 計時器變數的值, 首先, 在 ISaGRAF 視窗中點選"Dictionary"圖示在"ISaGRAF Global Variables"視窗中點選"Timers"標籤, 然後以滑鼠雙擊"T1"變數名稱, 此時會開啟"Write Timer Variable"視窗, 您可以在"Enter New Value:"欄位中將"t#1s"改成"t#500ms"並按下"Write"鈕.

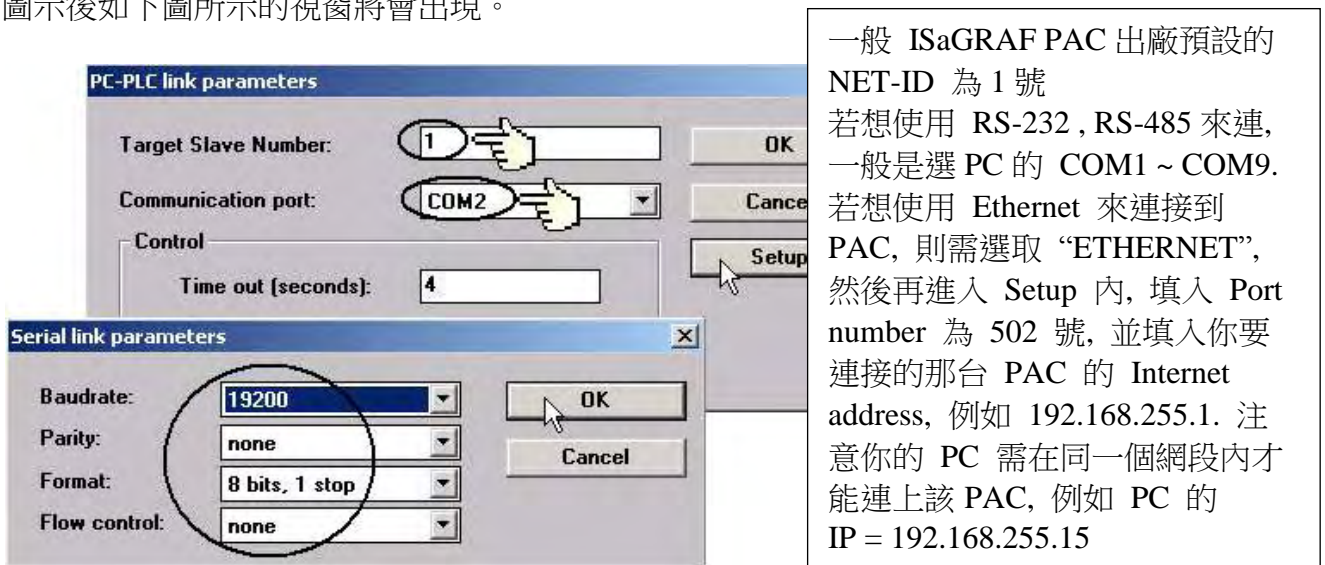


2.1.5: 下載 及 除錯 LD 範例程式

最終，您可將此 LD 範例程式下載至 IsaGRAF PAC 上運作，在下載之前您必須在 控制器 與 PC 間建立通訊。



首先，請您用滑鼠在"ISaGRAF Programs"視窗中點選"Link Setup"圖示，當您點選"Link Setup"圖示後如下圖所示的視窗將會出現。



其中"Target Slave Number"是第 1 章 1.3.1 節 I-8xx7 控制系統的 Net-ID，Net-ID 的位置在 I-8xx7 控制器的右下方。Net-ID 是設定您控制器的編號，本範例您的控制器是設為 1，而“Communication Port”是指通訊線連結到你電腦所連接的通訊埠，通常不是 Com1 就是 Com2。(關於 I-7188EG/XG, μ PAC-7186EG, iP-8xx7, WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6 的 NET-ID 設定，請參閱它們各自的“快速上手手冊”)

控制器 的通訊參數和您電腦上的通訊參數必需設定為一樣的。在 I-8xx7, I-7188EG/XG 控制器的通訊埠 Com1(RS232)及 Com2(RS232)的預設參數如下：

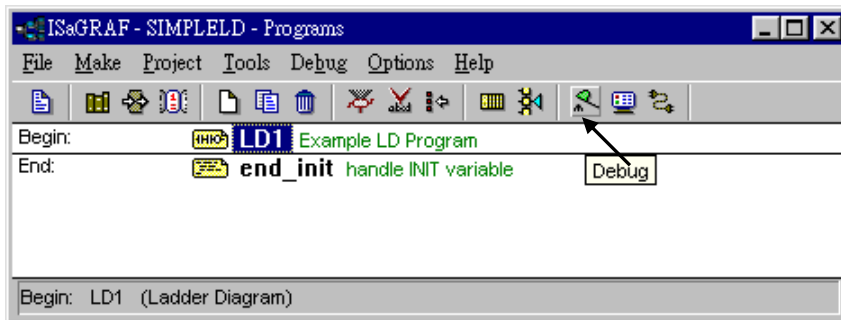
Baudrate:	19200
Parity:	none
<hr/>	
Format:	8 bits, 1 stop
Flow control:	none

注意：

您必需確定電腦上的通訊埠和 IsaGRAF PAC 是否能夠正常通訊，若不行則必需修改電腦通訊埠的設定讓電腦和 控制器 能夠順利溝通。

下載 LD 範例程式

在您下載 LD 範例程式到控制器之前，您必需先確認電腦和控制器之間的連線是否正常，若正常無誤，按下 ISaGRAF 程式視窗中的 Debug 鍵，開啟如下之畫面：



若出現程式的名字和“Run”這兩個字，則表示控制器中有一個應用程式在跑。若在 ISaGRAF Debugger 視窗中出現“Disconnect”則表示電腦和控制器並沒有連線成功。

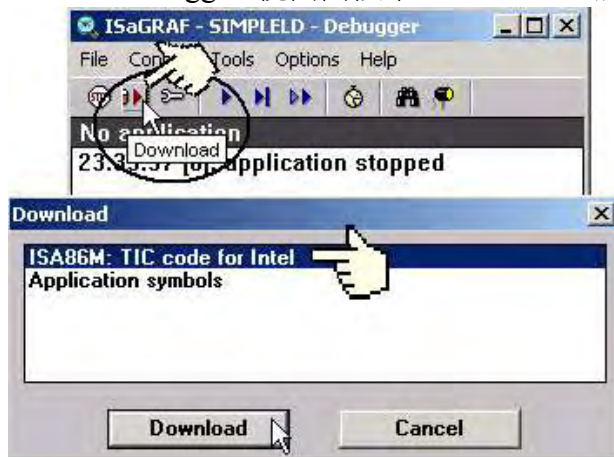
最常導致連結失敗的原因是彼此之間的通訊埠沒有插好，或是彼此間的通訊參數未設正確。這時您必需重插電腦上的通訊線到正確的通訊埠或是重設 ISaGRAF 程式的連結設定。

若原先控制器內已有應用程式在執行，需先停掉該應用程式之後才能下載新的應用程式，方法為在“ISaGRAF Debugger”視窗下選取 STOP 鍵停止應用程式，此時便可重新下載應用程式至控制器。



開始下載程式

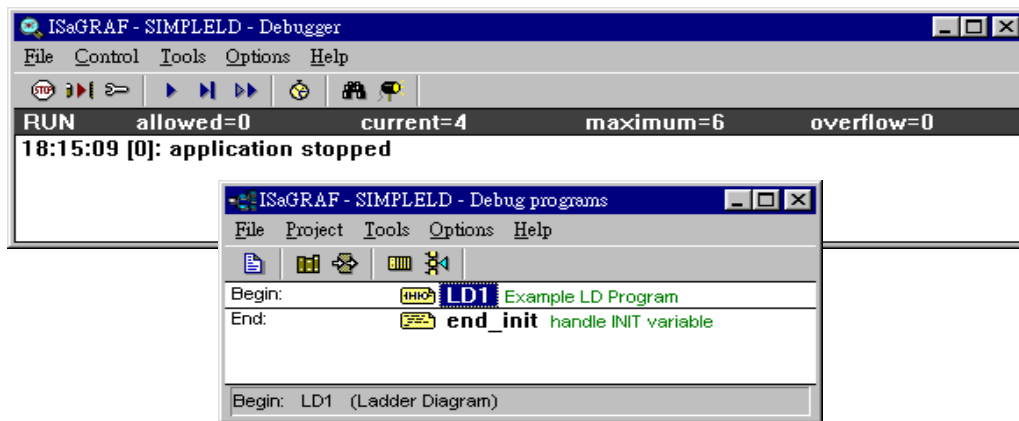
從“ISaGRAF Debugger”視窗內按下“Download”鍵，接著選擇“ISA86M:TIC Code For Intel”



將會開始下載 LD 範例程式至控制器內，而在 “ISaGRAF Debugger”視窗中會出現程式下載進度的畫面。

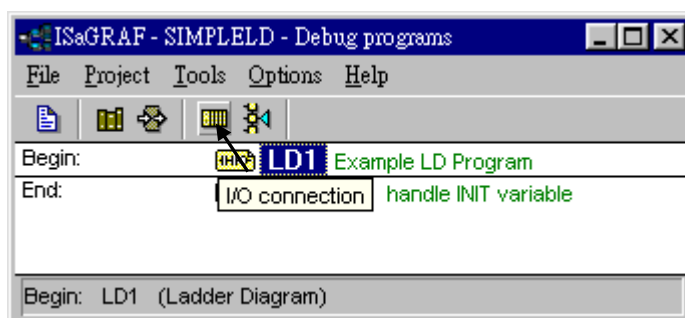


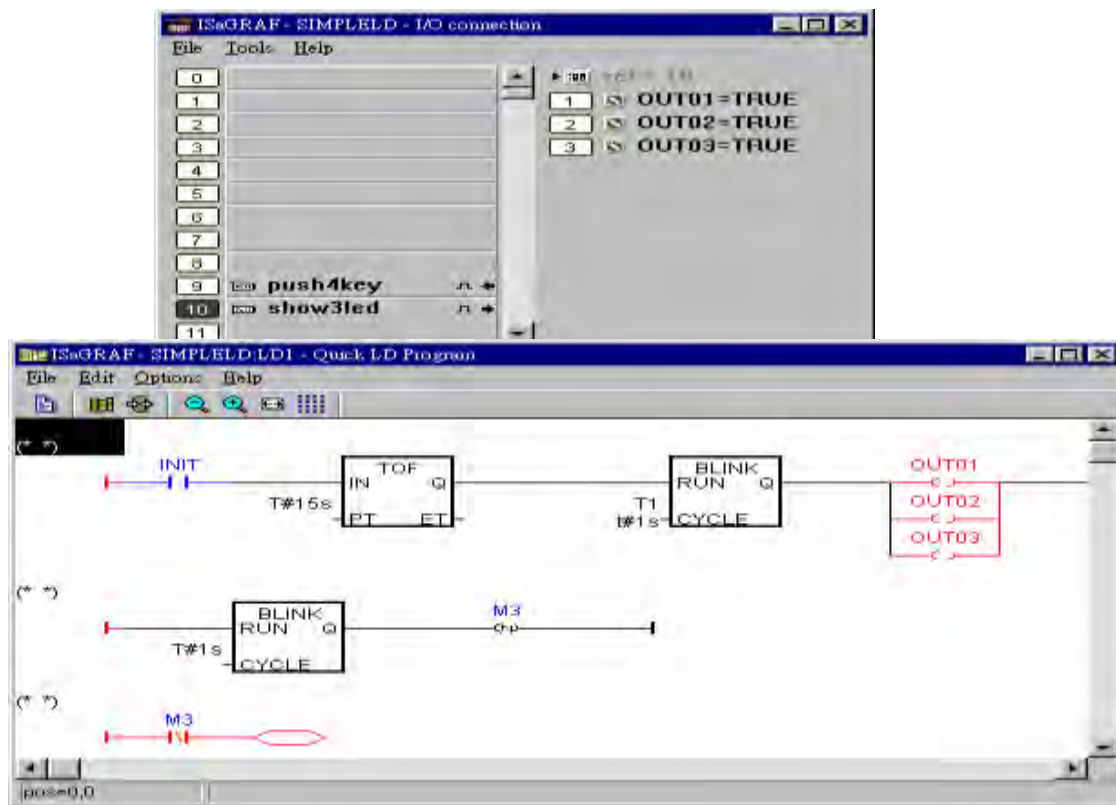
當 LD 程式下載完成後，將會出現下面兩個視窗



監視 LD 範例程式

當您在監視 LD 範例程式時，透過 ISaGRAF 視窗畫面您可以觀察即時的 I/O 狀態。在 ISaGRAF Debugger 視窗中按下 “I/O connections”鍵開啟 “I/O connection，透過“I/O Connections”視窗會顯示所有的輸出及輸入點的狀態。另外一個有用的視窗即 “Quick LD program”，經由這個視窗可以即時的看到 LD 程式的執行狀態。





雖然在編寫和下載 ISaGRAF 程式有許多的步驟，但這些步驟都是很容易去實作出來。

練習！練習！練習！

現在您已能夠成功在 ISaGRAF 器控制內建立和執行一個 ISaGRAF 程式，但您還是需要透過練習去建立更多有用的程式，只有透過練習和實作才能讓自己更得心應手，祝您好運！

2.2: 一個簡單的結構化文字 (ST) 程式

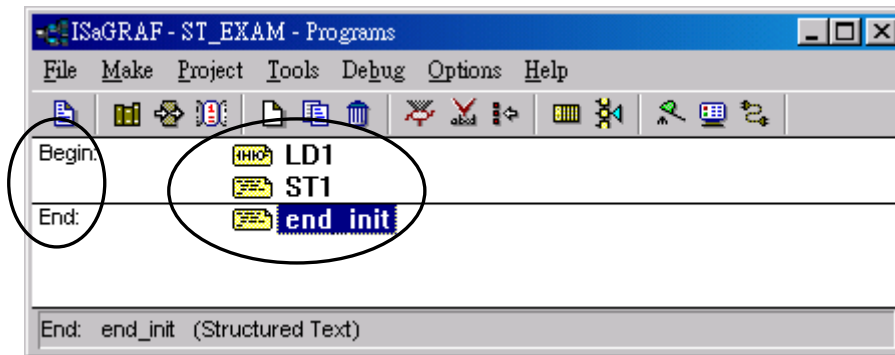
結構化文字是用於控制器的高階程式語言，結構化文字(以後皆稱為 ST)主要可用來撰寫一些其它語言(如 LD, FBD)不易表示的較複雜邏輯。

ST 程式是由敘述組成的，每個 ST 敘述必須以分號結尾“;”，在“(”與“)”中間的任何字為註解。

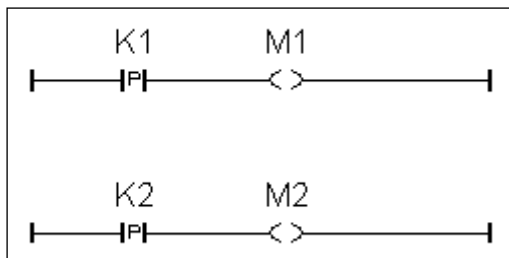
本範例內的變數宣告：

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 “TRUE”. 用 TRUE 來表示第 1 個 PLC 周期
K1	Boolean	Input	連到 I-8xx7 面板上的第 1 個按鈕
K2	Boolean	Input	連到 I-8xx7 面板上的第 2 個按鈕
M1	Boolean	Internal	用來表示 K1 剛剛被按下
M2	Boolean	Internal	用來表示 K2 剛剛被按下
TEMP	Boolean	Internal	暫時使用的 boolean 內部變數
COUNT	Integer	Internal	初值設為 0, 按 K1 & K2 會改變其值

本範例內含有 3 個程式。一個名稱叫 “LD1” 為 LD 程式，另外兩個 ST 程式名稱為 “ST1” 及 “end_init”，如下圖。



“LD1” 程式全貌：



“ST1” 程式全貌:

```
(* 在第 1 個 PLC scan 開啟 Com3 串口, 9600 baud rate, 8 char. , no parity, 1 stop bit *)
if INIT = TRUE then
    TEMP := comopen(3, 9600, 8, 0, 1) ;
end_if ;

(* 當 K1 或 K2 有動作時 *)
if ( M1 = TRUE ) or ( M2 = TRUE ) then

    (* 若為 K1 有動作, COUNT 加 1 *)
    if M1 = TRUE then
        COUNT := COUNT + 1 ;
    end_if ;

    (*若為 K2 有動作, COUNT 加 10 *)
    if M2 = TRUE then
        COUNT := COUNT + 10 ;
    end_if ;

    (* 將 COUNT 值存入 2 號 integer 陣列的第 5 個位址 *)
    TEMP := ARY_N_W(2, 5, COUNT) ;

    TEMP := COMWRITE(3, 16#2) ;    (*寫出 1 個 byte = 2 (16 進位) 到 Com3 *)

    (*寫出 1 個 integer 值 (1 個 integer 包含 4 個 byte)到 COM3, 該 integer 值放於
    2 號 integer 陣列的第 5 個位址*)
    TEMP := COMAY_NW(3, 2, 1, 5) ;

    TEMP := COMWRITE(3, 16#3) ;    (* 寫出 1 個 byte = 3 (16 進位) 到 Com3 *)

end_if ;
```

“end_init” 程式全貌:

```
INIT := FALSE ;
```


範例程式邏輯說明:

“LD1” 程式 :

取得按鈕 K1 剛被按下去的瞬間狀態, 存入此 “瞬間狀態” 於 boolean 變數 “M1”
取得按鈕 K2 剛被按下去的瞬間狀態, 存入此 “瞬間狀態” 於 boolean 變數 “M2”

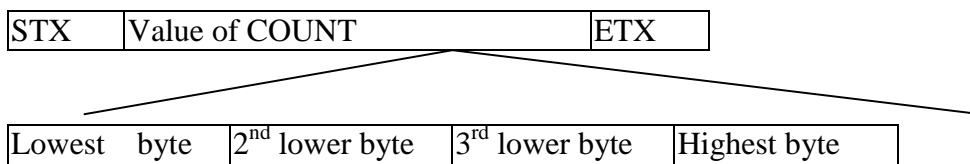
“ST1” 程式:

在第 1 個 PLC 周期時, 開啟通訊埠 Com3 為 速率 9600, 字元大小為 8 個 bit. , 沒有同位檢查, 停止位元大小為 1 個 bit

當 K1 被按下時將變數 “COUNT” 值加 1

當 K2 被按下時將變數 “COUNT” 值加 10

將”COUNT” 值用以下的格式, 從 I-8xx7 的 Com3 送到 PC



STX : 封包啟始字元(Byte), 字元值為 2

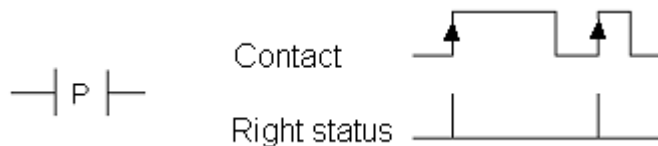
ETX : 封包結尾字元(Byte), 字元值為 3

“end_init” 程式:

在 PLC 周期的 end 區, 將 “INIT” 設成 FALSE. 如此 “INIT” 的值只會在第 1 個 PLC 周期為 TRUE.

方塊及元件說明:

“P” 接點 : 當接點的值剛好由 FALSE 上升為 TRUE 時, 會在 右側產生 1 個 TRUE 的脈衝.



Comopen(PORT, BAUD, CHAR, PARI, STOP) : 開啟 PAC 的通訊埠

輸入參數

PORT	: Integer	欲開啟的通訊埠, 3:COM3 ,4:COM4, ..., 20:COM20
BAUD	: Integer	通訊速率, 2400, 4800, 9600, 19200, 38400, 57600, 115200
CHAR	: Integer	字元大小, 7 or 8
PARI	: Integer	同位檢查, 0:none, 1:even, 2:odd
STOP	: Integer	停止位元大小, 1 or 2

回傳值 : boolean 正確: TRUE , 失敗: FALSE

Ary_N_W(NUM, ADR, DATA) : 存 1 個長整數(32 bit) 到整數陣列內.

輸入參數

NUM	:	Integer	存到幾號的陣列 (1-6)
ADR	:	Integer	存到該陣列內的那個位置 (1-256)
DATA	:	Integer	要存進去的值

回傳值 : boolean 正確: TRUE , 失敗: FALSE

ComWrite(PORT, DATA) : 寫出 1 個位元組(Byte) 到通訊埠

輸入參數

PORT	:	Integer	欲使用的通訊埠, 3:COM3 ,4:COM4, ..., 20:COM20
DATA	:	Integer	欲寫出的位元組 (0 - 255)

回傳值 : boolean 正確: TRUE , 失敗: FALSE

ComAy_NW(PORT, ARY_NO, NUM, POS) : 寫出 1 個長整數陣列到通訊埠

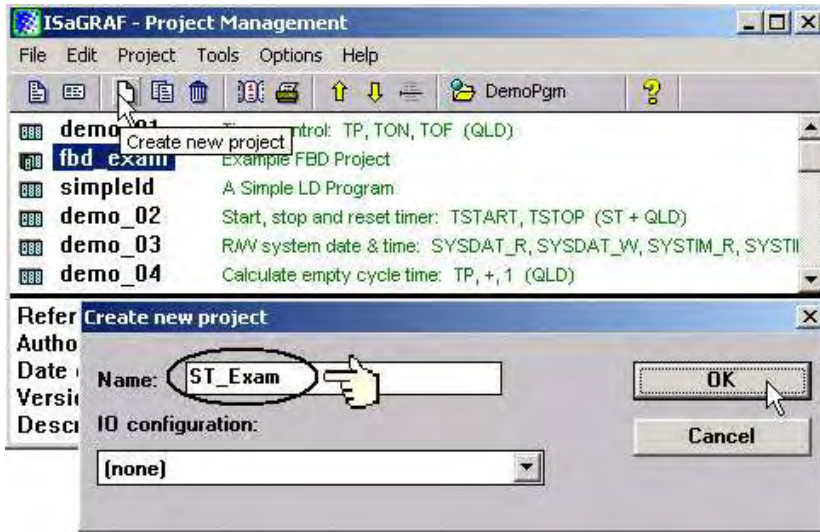
輸入參數

PORT	:	Integer	欲使用的通訊埠, 3:COM3 ,4:COM4, ..., 20:COM20
ARY_NO	:	Integer	欲寫出的陣列號碼 (1-6)
NUM	:	Integer	該陣列內有幾個整數要寫出 (0-256)
POS	:	Integer	從陣列內的第幾號位置開始寫出 (1-256)

回傳值 : boolean 正確: TRUE , 失敗: FALSE

第一步需建立 1 個專案。

在 "ISaGRAF Project Management" 視窗內以滑鼠點選"Create New Project"，之後輸入一個專案名稱。



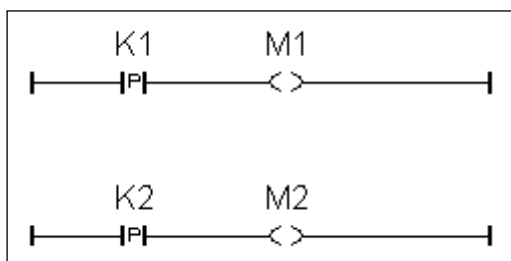
宣告變數如下表

請參閱第 2.1.1.3 節的方法。

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 “TRUE”. 用 TRUE 來表示第 1 個 PLC 周期
K1	Boolean	Input	連到 I-8xx7 面板上的第 1 個按鈕
K2	Boolean	Input	連到 I-8xx7 面板上的第 2 個按鈕
M1	Boolean	Internal	用來表示 K1 剛剛被按下
M2	Boolean	Internal	用來表示 K2 剛剛被按下
TEMP	Boolean	Internal	暫時使用的 boolean 內部變數
COUNT	Integer	Internal	初值設為 "0", 按 K1 & K2 會改變其值

建立新的 LD 程式，命名為 “LD1” 並編成如下圖的程式內容。

請參閱第 2.1.1.4 節的方法。



使用與第 2.1.1.6 節相同的步驟，建立一個 ST 程式 “end_init” 如以下的程式內容，切記 此 “end_init” 要放在 “End” 區

```
INIT := FALSE ;
```

建立另一個 ST 程式 “ST1”，放在 “Begin” 區，輸如以下的程式碼。方法可參閱 第 2.1.1.6 節

```
(* 在第 1 個 PLC scan 開啟 Com3 串口, 9600 baud rate, 8 char. , no parity, 1 stop bit *)
if INIT = TRUE then
    TEMP := comopen( 3 , 9600 , 8 , 0 , 1 ) ;
end_if ;

(* 當 K1 或 K2 有動作時 *)
if ( M1 = TRUE ) or ( M2 = TRUE ) then

    (* 若為 K1 有動作, COUNT 加 1 *)
    if M1 = TRUE then
        COUNT := COUNT + 1 ;
    end_if ;

    (*若為 K2 有動作, COUNT 加 10 *)
    if M2 = TRUE then
        COUNT := COUNT + 10 ;
    end_if ;

    (* 將 COUNT 值存入 2 號 integer 陣列的第 5 個位址 *)
    TEMP := ARY_N_W( 2 , 5 , COUNT ) ;

    TEMP := COMWRITE( 3 , 16#2 ) ;    (*寫出 1 個 byte = 2 (16 進位) 到 Com3 *)

    (*寫出 1 個 integer 值 (1 個 integer 包含 4 個 byte)到 COM3, 該 integer 值放於
    2 號 integer 陣列的第 5 個位址*)
    TEMP := COMAY_NW( 3 , 2 , 1 , 5 ) ;

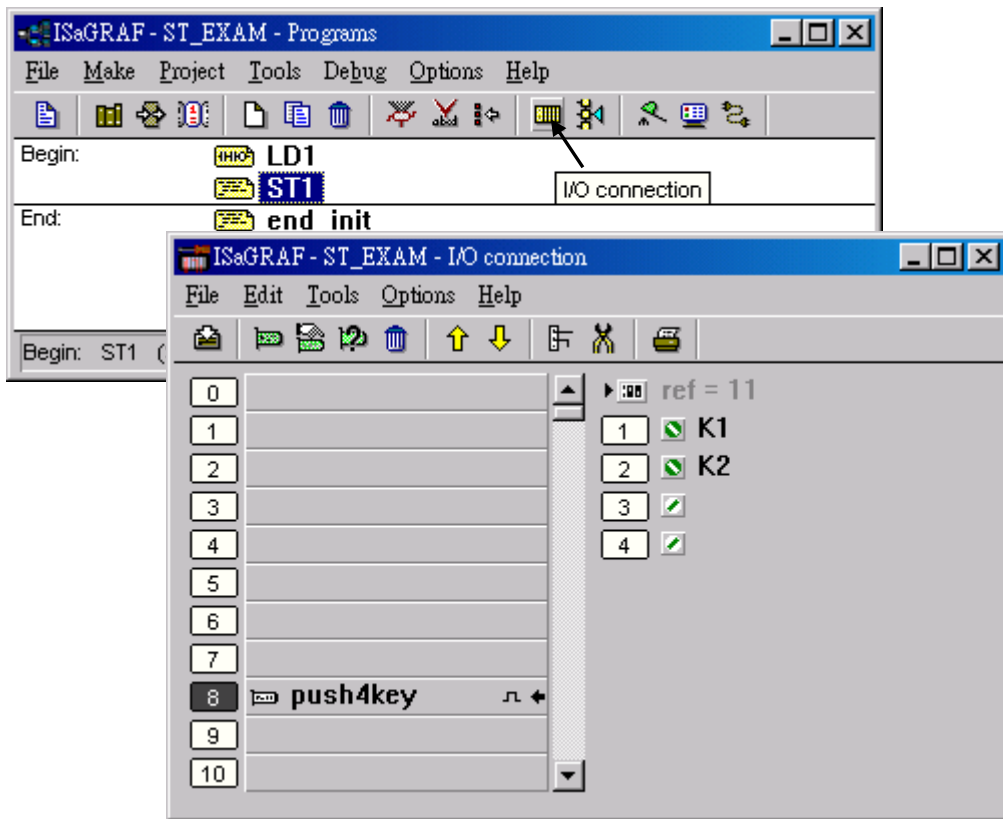
    TEMP := COMWRITE( 3 , 16#3 ) ;    (* 寫出 1 個 byte = 3 (16 進位) 到 Com3 *)

end_if ;
```

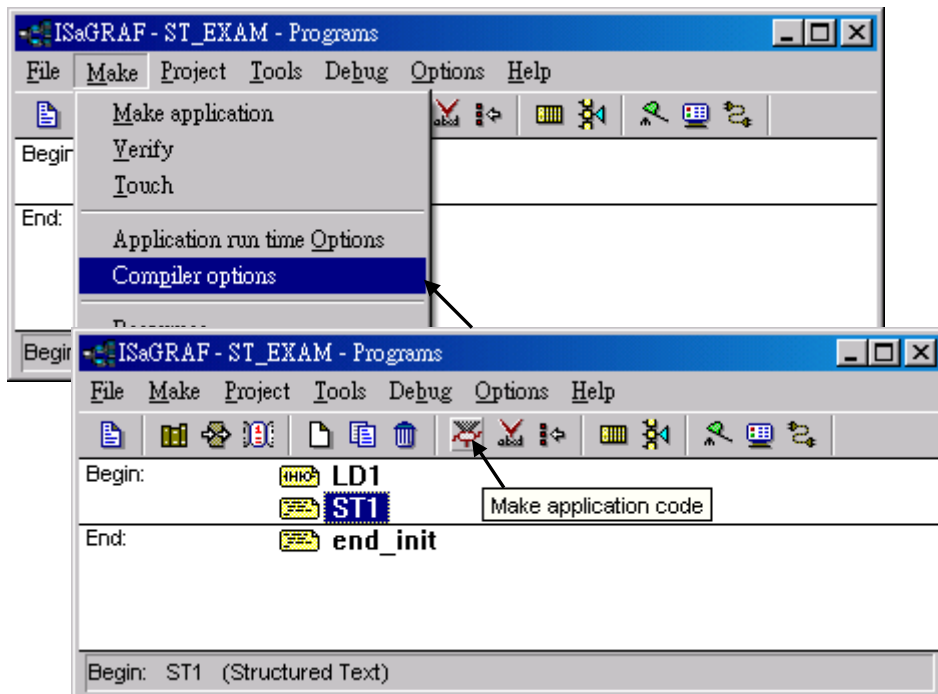
注意:

每個 ST 敘述需以分號 ";" 結尾。建議隨時以滑鼠去點選”Save”，以保安全

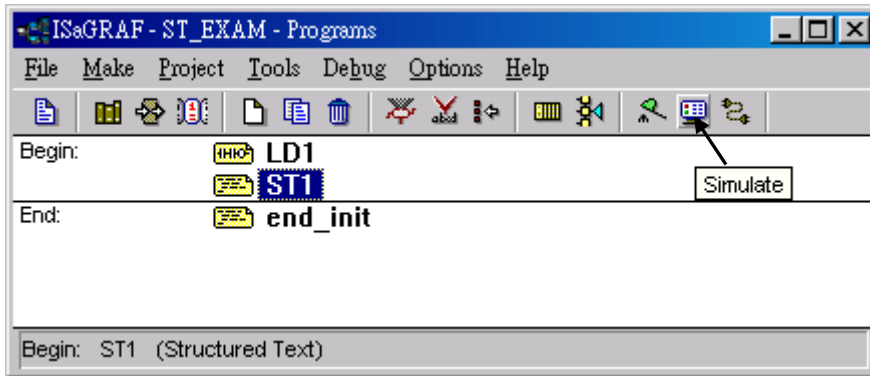
請使用如 2.1.2 節的方法完成 I/O 連結



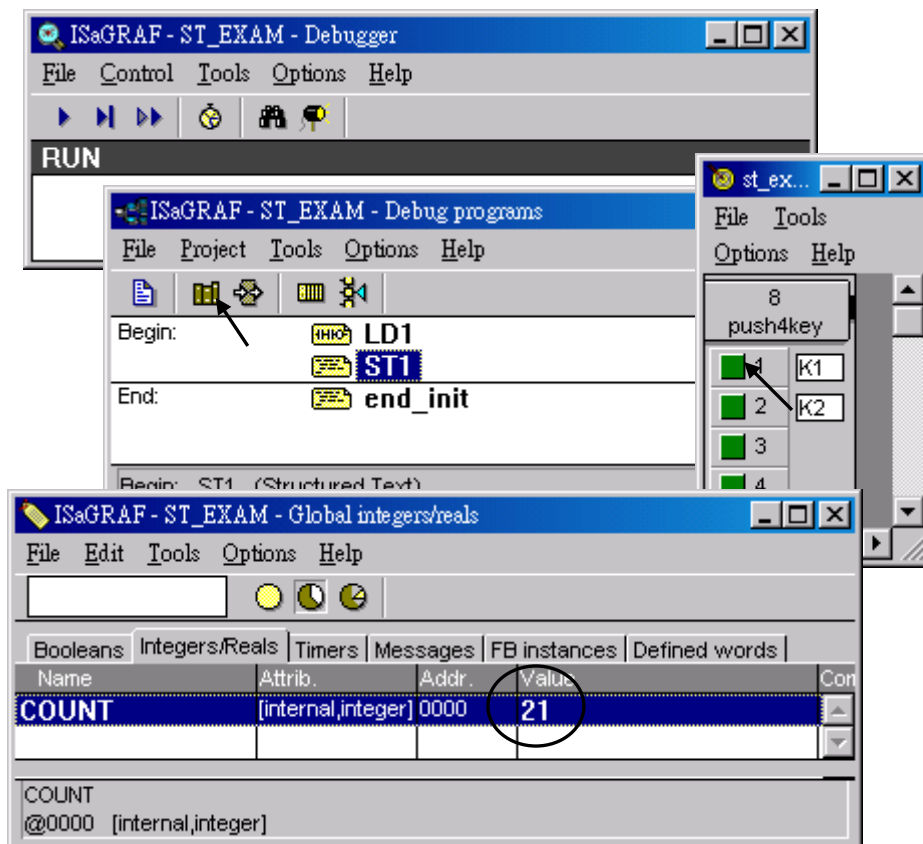
請使用如 2.1.3 節的方法完成編譯選項的設定，之後編譯(Compile)此專案。



編譯成功後，點選 "Simulate" 來模擬專案的執行結果。



模擬時可開啟 "dictionary" 視窗去檢視 "COUNT" 的值。以滑鼠點選 "K1" 或 "K2"，可看出 "COUNT" 的值會發生變化。



模擬完後接下來可將此專案下載到 ISaGRAF 控制器內。(請參閱第 2.1.5 節的方法)

控制器在執行此專案，每當 K1 或 K2 按鈕被按下時，會寫出 6 個位元組(Byte) 到 Com3。假如你有可監視 RS232 傳輸狀況的程式，你就可觀查到

2.3: 簡單的功能方塊圖程式

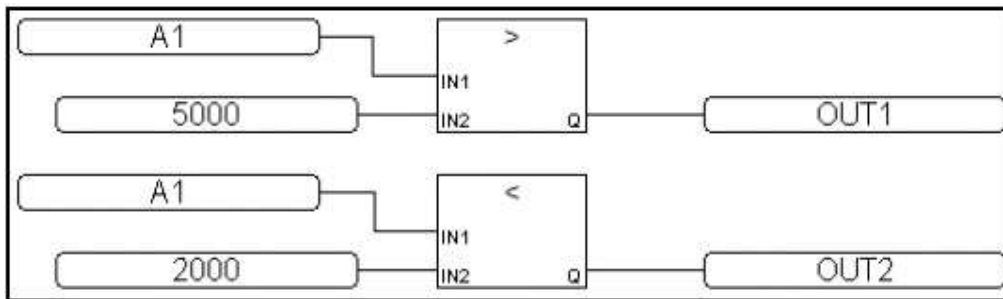
功能方塊圖程式 (FBD) 是一種圖形程式語言，它允許使用者透過 ISaGRAF 函式庫內的 "Function" 或 "Function Block" 來編寫複雜的邏輯程序程式。在下面的小節裡，我們會詳細的說明如何用 ISaGRAF 建立功能方塊程式。

FBD 控制程式範例

下面為在這一節裡 FBD 程式中所使用的變數名稱。

名稱	型態	屬性	描述
OUT1	Boolean	Output	輸出變數
OUT2	Boolean	Output	輸出變數
A1	Integer	Internal	內部變數，初始值為 0

FBD 程式圖



FBD 程式說明

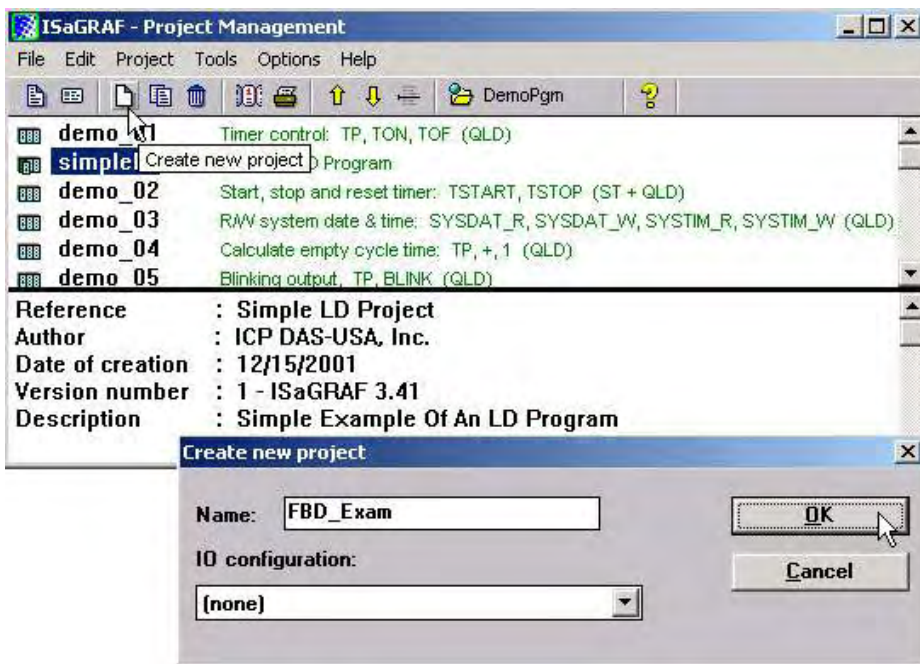
假如 "A1" > 5000, 輸出變數 "OUT1" 等於 "TRUE".
假如 "A1" < 2000, 輸出變數 "OUT2" 等於 "TRUE".
其他狀況, 輸出變數 "OUT1" 及 "OUT2" 均是 "FALSE"

2.3.1: 編寫 FBD 範例程式

建立 FBD 程式和在 2.1 節所建立對 LD 程式是非常類似的，下面我們將詳細的介紹如何建立一個簡易的 FBD 程式。

建立一個新的 FBD 專案

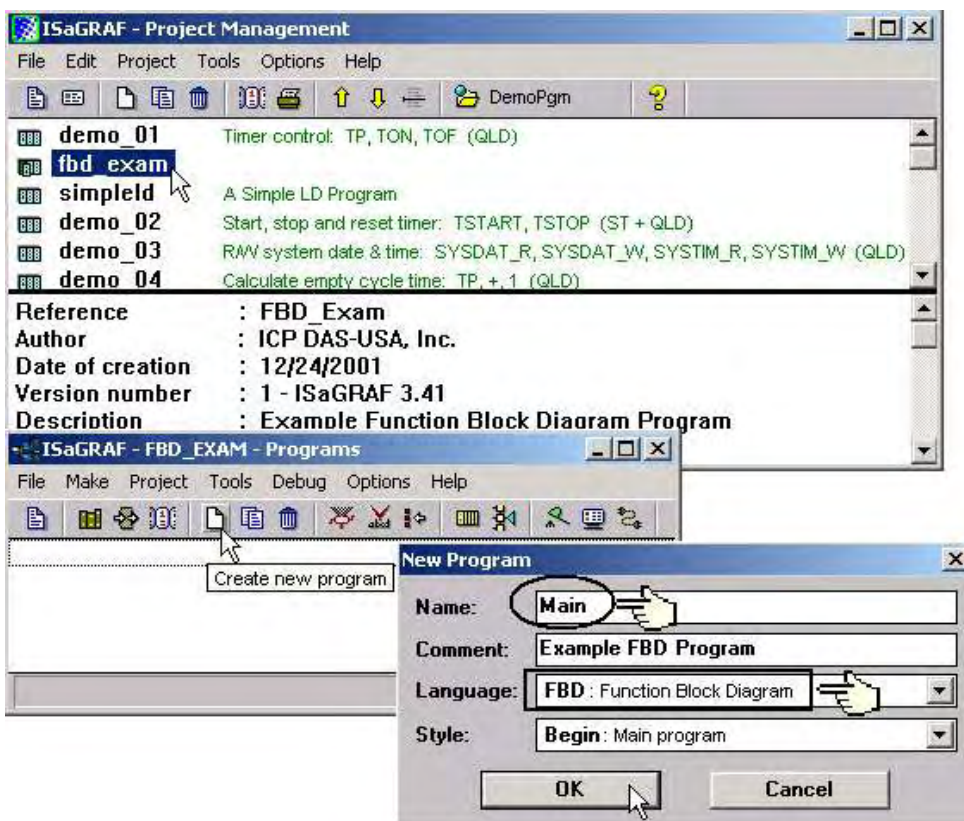
從 "ISaGRAF Project Management" 視窗單擊 "Create New Project" 鍵，然後輸入程式名稱 "FBD_Exam"。



在您建立新的專案後，在“ISaGRAF Project Management”視窗內單擊“FBD_Exam”開啟 FBD 專案，在“ISaGRAF Programs”視窗內單擊“Create New Program”以開啟“New Programs”視窗。

首先：在“Name”欄輸入“Main”（本範例的名稱），然後在“Language”欄內選擇“FBD: Function Block Diagram.”，在“Style”欄內選擇“Begin: Main Program”，您可以在“Comment”內加入註解。

此時您已建立了一個空的 FBD 程式，你可以用滑鼠雙擊“Main”名稱來開啟編輯器。

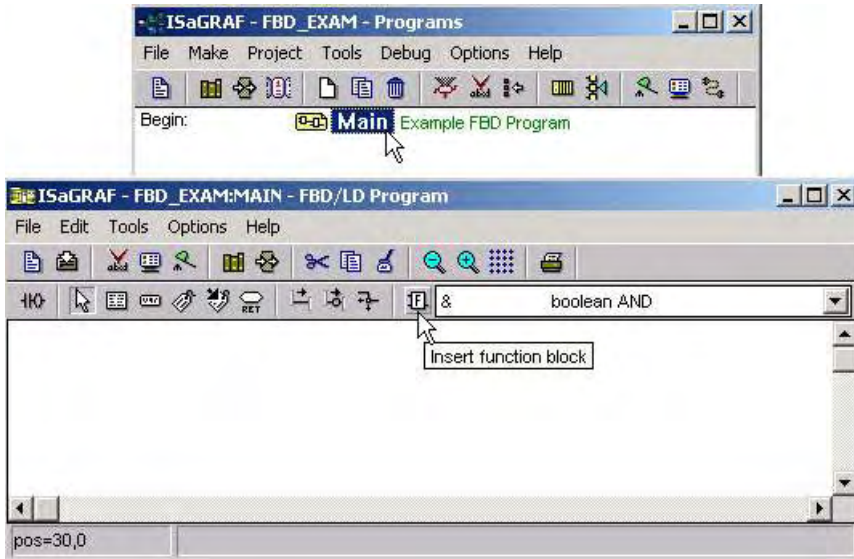


宣告變數

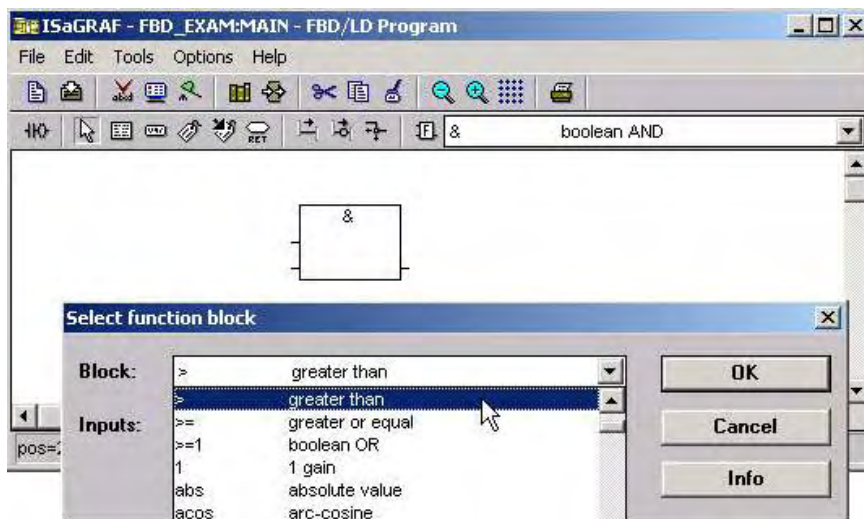
在 FBD 範例程式中我們將宣告 3 個變數，這 3 個變數分別為 “OUT1”，“OUT2”及 “A1”。宣告 FBD 程式變數的方法和 LD 程式中宣告的一樣，請參考 2.1.1.3 節。

編輯 FBD 程式

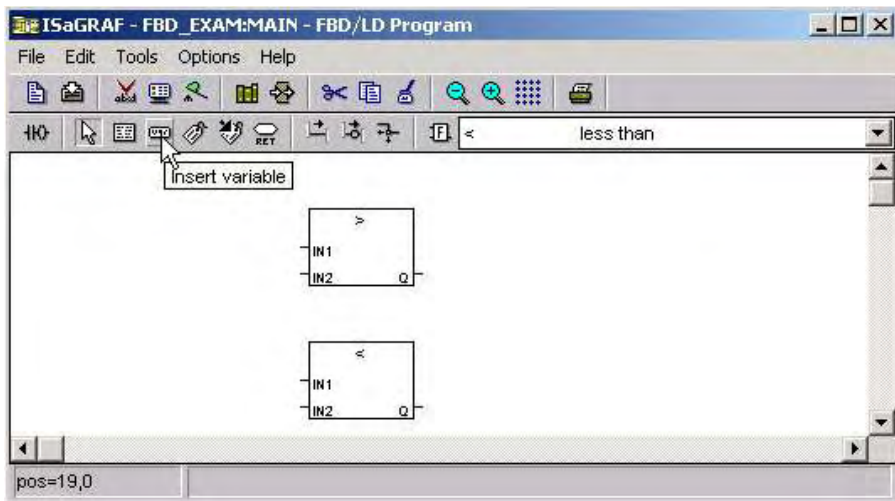
在 “ISaGRAF Program”視窗中雙擊 “Main”，然後單擊 “Insert function block”鍵。如下所示。



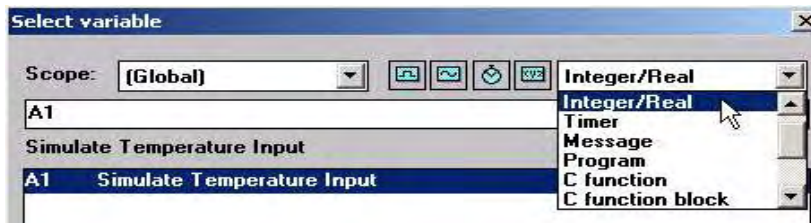
在所欲插入功能方塊的空白處以滑鼠按一下，即會新增一個功能方塊，以滑鼠雙擊新增的功能方塊內，選擇 “> Greater Than”。(單擊 “info”鍵會出現方塊的功能說明)



使用相同的程序，在“Greater Than”功能方塊下增加“< Less Than”功能方塊。

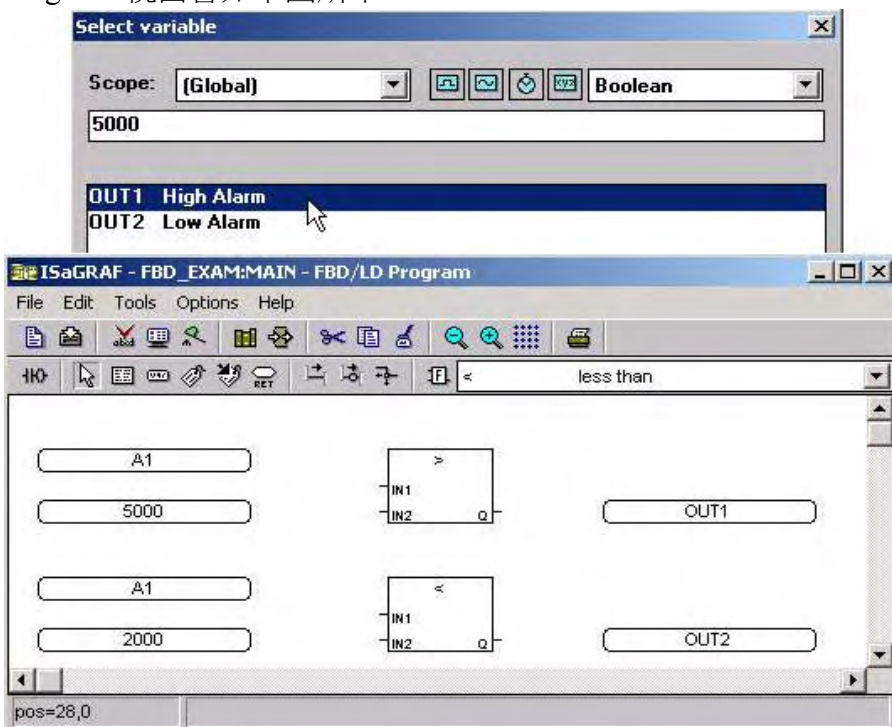


如上圖所示，單擊“Insert Variable”，在“ISaGRAF Select Variable”視窗上選取“Integer/Real”格式，A1 變數會出現在“ISaGRAF Select Variable”視窗上。

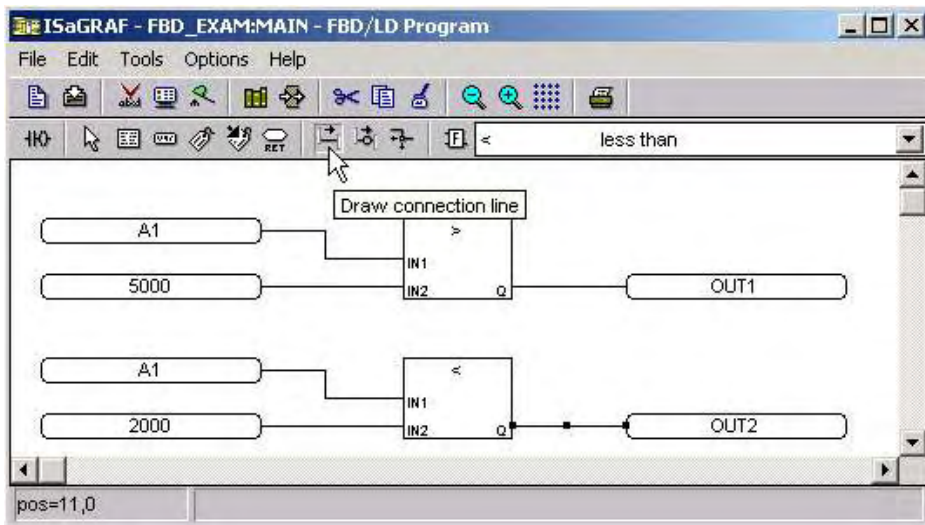


雙擊“A1”輸入變數，把“A1”變數放在“ISaGRAF FBD/LD”視窗中適當的位置，接著再增加另一個“A1”及“OUT1”、“OUT2”變數，放置的位置如下圖所示。

接著再增加兩個常數變數 2000 和 5000，分別放置在 A1 的下方。最後“ISaGRAF FBD/LD Program”視窗會如下圖所示。



接下來我們要完成每個變數和函式方塊的连接，單擊 “Draw Connection Line”作如下圖所示的连接。

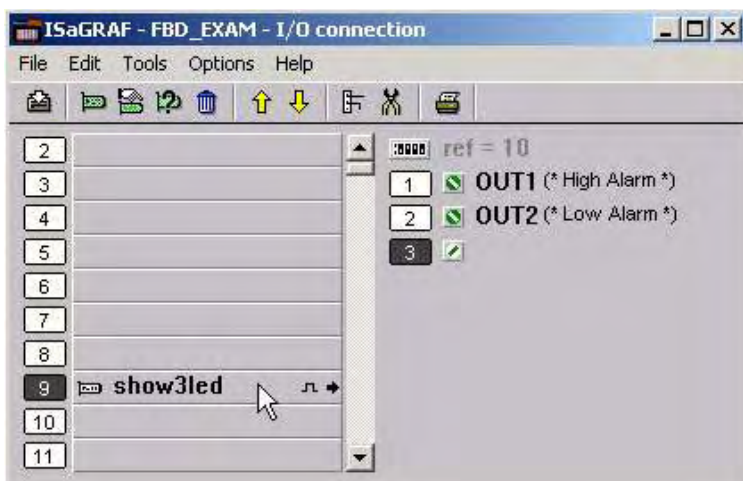


上方的 A1 變數連接到 “> Greater Than”函式方塊的 IN1 點，常數變數 5000 連接到 “>Greater Than”函式方塊的 IN2 點，下方的 A1 變數連接到 “< Less Than”函式方塊的 IN1 點，常數變數 2000 連接到 “< Less Than”函式方塊的 IN2 點。

最後連接 OUT1 變數到 “> Greater Than”函式方塊的 Q 點，連接 OUT2 變數到 “< Less Than”函式方塊的 Q 點。

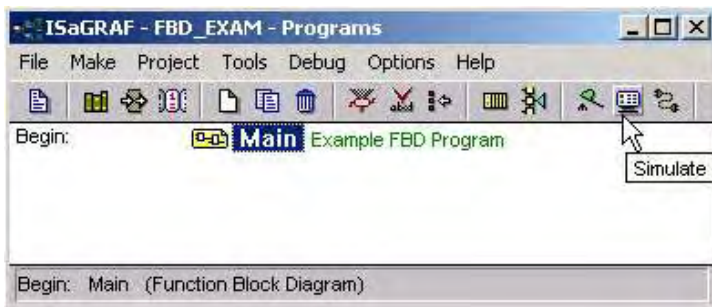
連結 I/O 及編譯專案

用跟 2.1.2 節及 2.1.3 節一樣的程序來連結 I/O 和編譯 FBD 範例程式。“ISaGRAF I/O Connect”視窗如下圖所示。



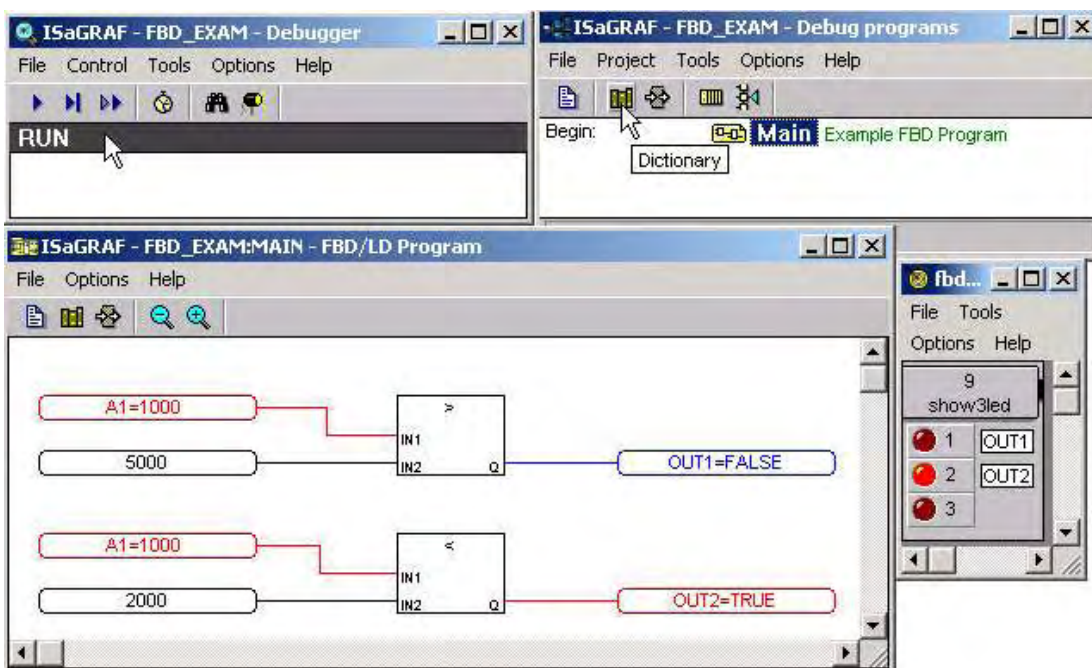
2.3.2: 模擬 FBD 程式

單擊 “ISaGRAF Programs”視窗下的 “Simulate”鍵來執行模擬 FBD 程式。

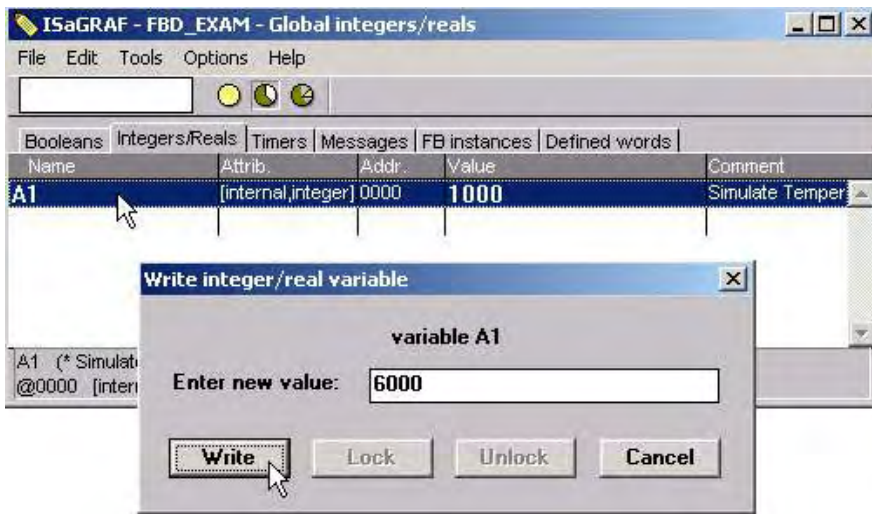


當您按下 “Simulate”鍵時，“ISaGRAF Debugger”，“I/O Simulate”及 “ISaGRAF Debug Programs”將會開啟。假如你單擊 “ISaGRAF Debug programs”視窗內的 “Main”，”ISaGRAF FBD/LD Program”視窗會開啟並顯示目前的執行狀況。

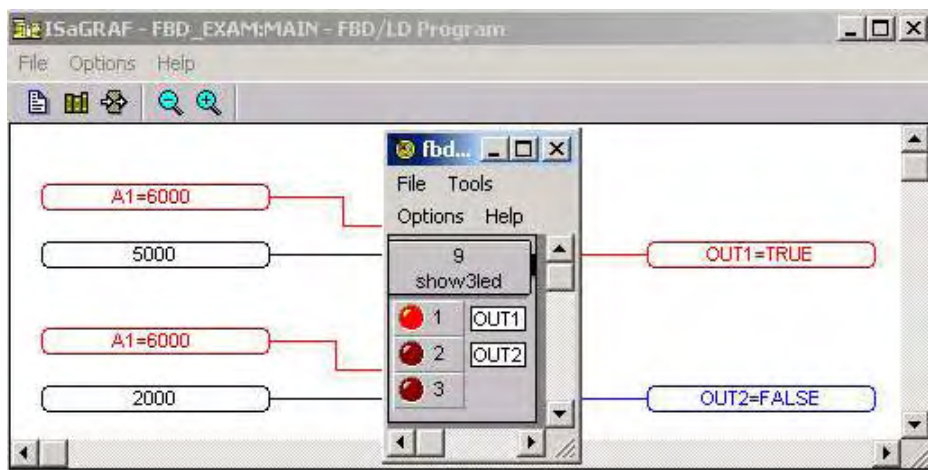
若現在 A1 變數設小於 2000(目前設定為 1000)，則 “OUT2”輸出變數為 TRUE， “OUT”輸出變數為 FALSE。



在“ISaGRAF Debug Programs”視窗中按下“Dictionary”鍵來開啟“Global Dictionary”視窗，接著選擇“Integer/Real”設定 A1 變數的值。



在"Enter New Value" 方格內輸入 6000，則“FBD/LD program”的畫面將會產生改變。



您可以參考 2.1.5 節的步驟下載 FBD 範例程式到 ISaGRAF 控制器中。

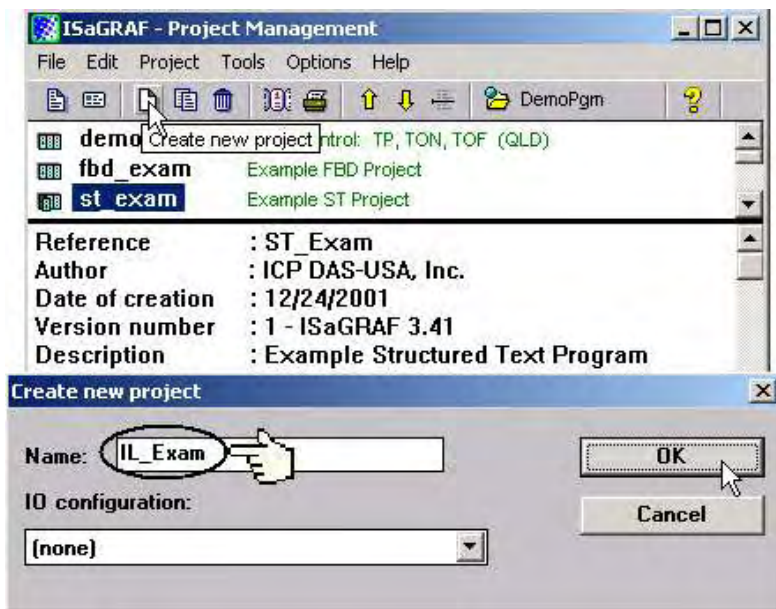
2.4: 簡單的指令集程式

指令集程式為包含了一系列指令的低階程式語言，它的指令一定關聯於目前結果(或 IL 暫存器)運算元顯示了目前數值與運算子之間的運算操作，而結果會再儲存於目前結果內。

IL 程式是一連串的命令。每一個指令必須寫在每一列的開始，而且必須包含一個運算元，一個或多個運算子(以逗號分開“,”)。標籤及其冒號必須寫在指令前。假使指令有註解時，註解必須寫在每一列的最後，而且必須起始於“(”終止於”)”。指令之間可以插入空的列。空的列可以加入註解。

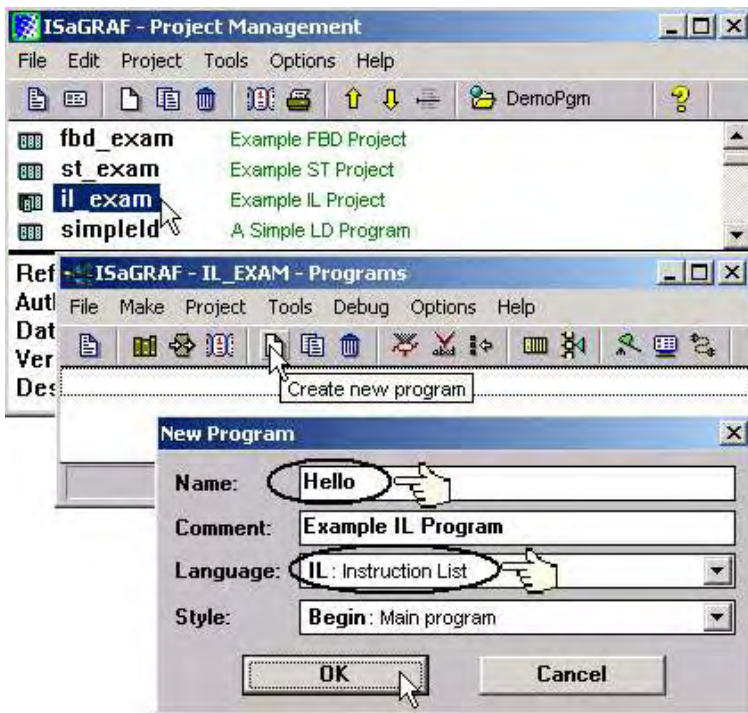
本節將描述如何撰寫 IL 程式，這個 IL 程式和 ST 及 FBD 程式具有相同的程式功能和變數要求。(請參考 2.2 及 2.3 節)

先為 IL 程式建立一個專案。

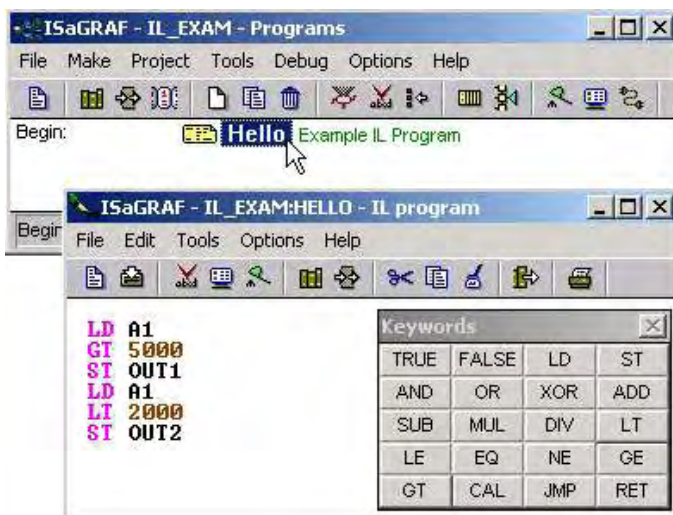


為 IL 程式建立一個 “IL_Exam”的專案名稱。

雙擊 “IL_Exam”，在開啟的 “ISaGRAF Programs” 程式視窗內建立一個 “Hello” 程式，在語言種類選項中選擇 “IL : Instruction List”。



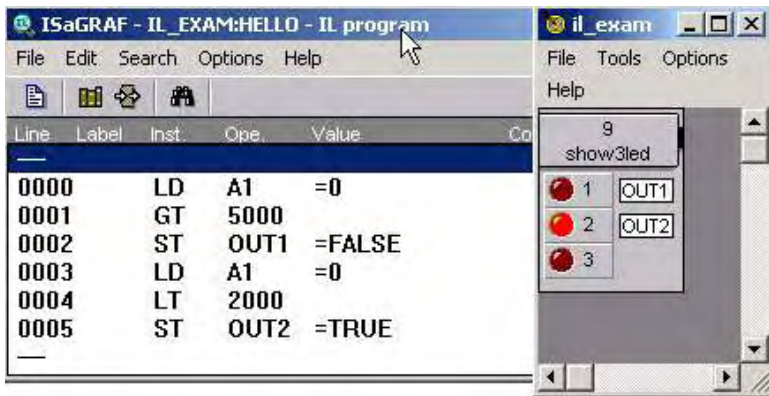
當您按下 “OK” 鍵時， “ISaGRAF Programs” 視窗將會開啟，按下 “Hello” 鍵開啟 IL 程式。



宣告 IL 程式變數

本 IL 程式使用和 FBD 程式中相同的變數名稱， “OUT1” 和 “OUT2” 變數及整數變數 “A1”。請參考 2.1.1.3 節。 “Connecting I/O”， “Compiling” 及 “Simulate” 的方法均和 FBD 程式相同。

當您連結 I/O 及編譯 IL 程式後，單擊 “Simulate” 鍵，執行結果將如下圖所示：



因為 “A1” 初做值為 0，所以 OUT1 為 FLASE，OUT2 為 TRUE，改變 A1 的變數值為 5001，OUT1 將變為 TRUE，而 OUT2 將變為 FALSE。



2.5: 簡單的順序式功能圖程式(SFC)

順序式功能圖(Sequential Function Chart，簡稱 SFC)是一種描述循序操作的繪圖式語言。圖中是用一連串定義明確的步驟(steps)來表示，步驟間以轉移條件(transition)連結起來。每一個轉移條件都有一個布林判斷條件(Boolean condition)。步驟中的行為可以用其他語言(ST、IL、LD、和 FDB)來詳述。

SFC 程式是以一連串的步驟和轉移條件圖形來表示，而步驟與轉換條件之間是用有方向性的線(oriented links)連結在一起。多連結線可以用來表示發散和收斂。完整程式的某些部份可以被分隔出來，然後用一個所謂巨集步驟(macro steps)的單一符號表示在主要圖形裡。SFC 的基本繪圖規則如下：

- 1、步驟之後不能緊跟著另外一個步驟
- 2、轉移條件之後不能緊跟著另一個轉移條件

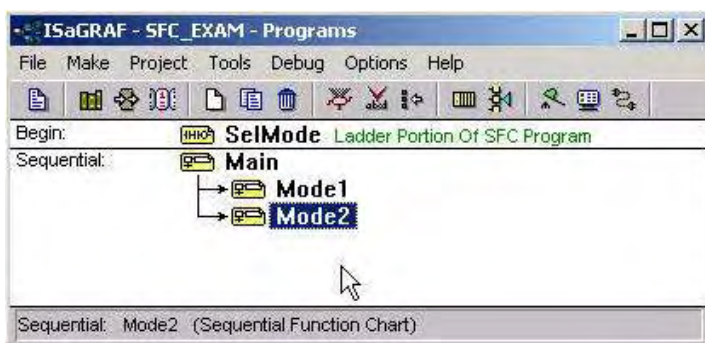
SFC 語言的基本元件(圖形符號)是步驟(steps)、初始步驟(initial steps)、轉移條件(transitions)、有方向性的線(oriented links)、跳躍到步驟(jumps to a step)。

下面為在 SFC 程式中所使用的變數名稱

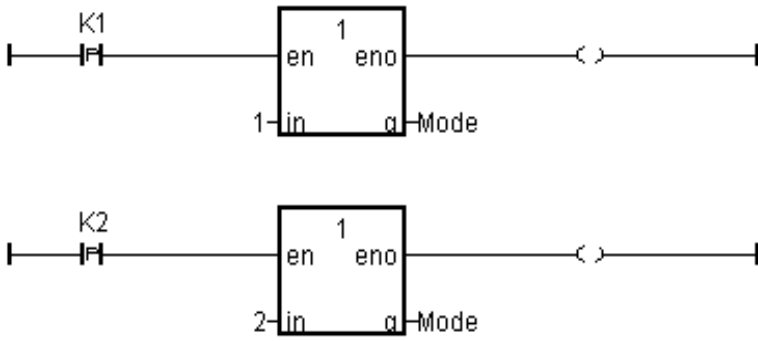
名稱	型態	屬性	描述
OUT1	Boolean	Output	輸出變數
OUT2	Boolean	Output	輸出變數
K1	Boolean	Input	模式 1 的輸入變數
K2	Boolean	Input	模式 2 的輸入變數
TMR1	Timer	Internal	計時器變數, 初始值為 "T#1s"
Mode	Integer	Internal	1 代表模式 1, 2 代表模式 2, 初始值為 1

SFC 程式

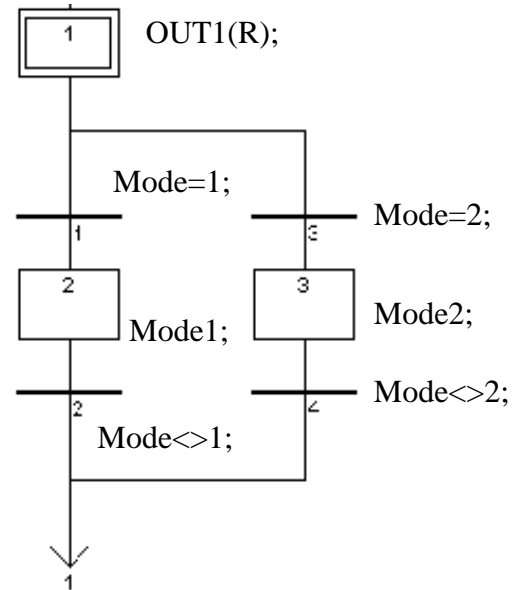
當您完成了“ISaGRAF Programs”視窗內的設定，將會如下圖所示：



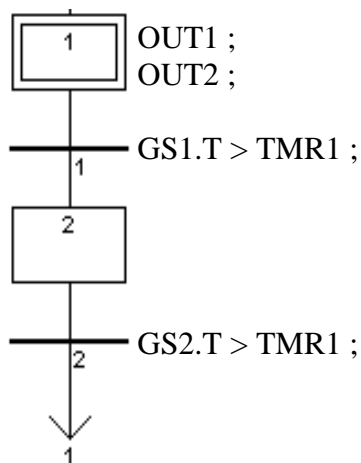
LD 程式 "SelMode"



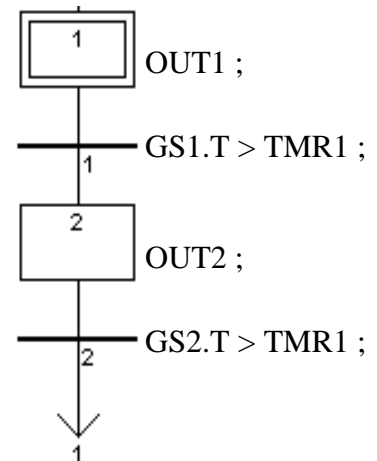
SFC 程式 "Main"



SFC 子程式 "Mode1"



SFC 子程式 "Mode2"

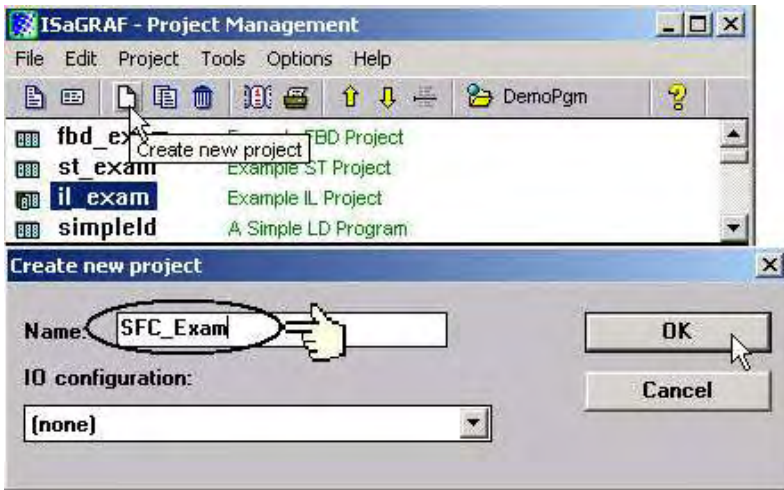


SFC 程式行為:

- 1.當 "K1" 的行為被觸發, 執行"Mode1" 程式
- 2.當 "K2" 的行為被觸發, 執行"Mode2" 程式

2.5.1: 撰寫 SFC 範例程式

建立 SFC 範例程式和第 2.1 小節所描述的方法相同，但記得要宣告變數 “K1”、“K2”、“OUT1”、“OUT2”、“TMR1” 及 “Mode”。

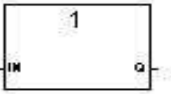


在建立完 SFC 專案之後，建立一個 “SelMode” 的 LD program，如下所示:



注意:

在這個 SFC 範例程式中，我們使用一個之前沒使用過的函式方塊 “1 Gain”，加入我們的 LD 程式中。



Arguments:

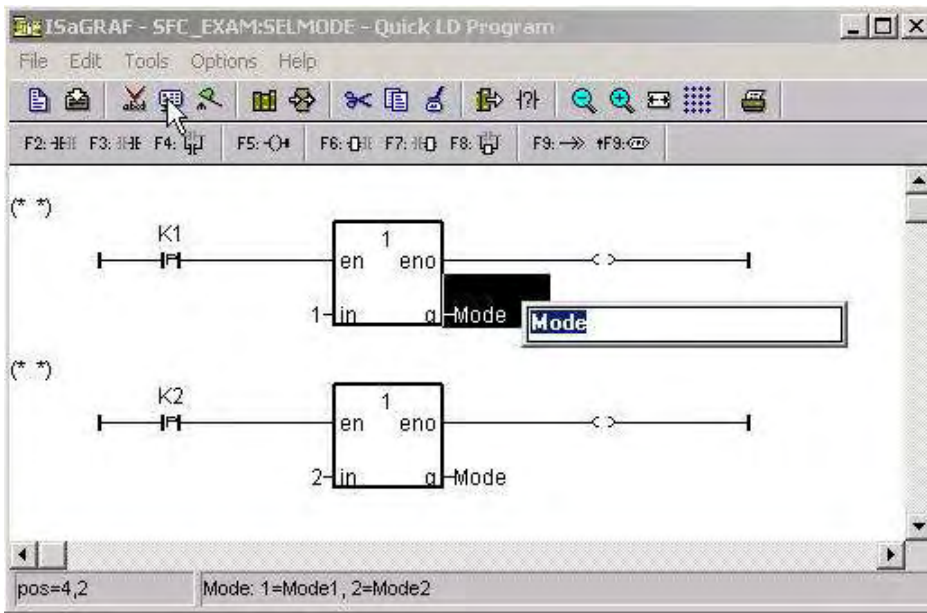
IN	any type
Q	any type

Description:

assignment of one variable into another one

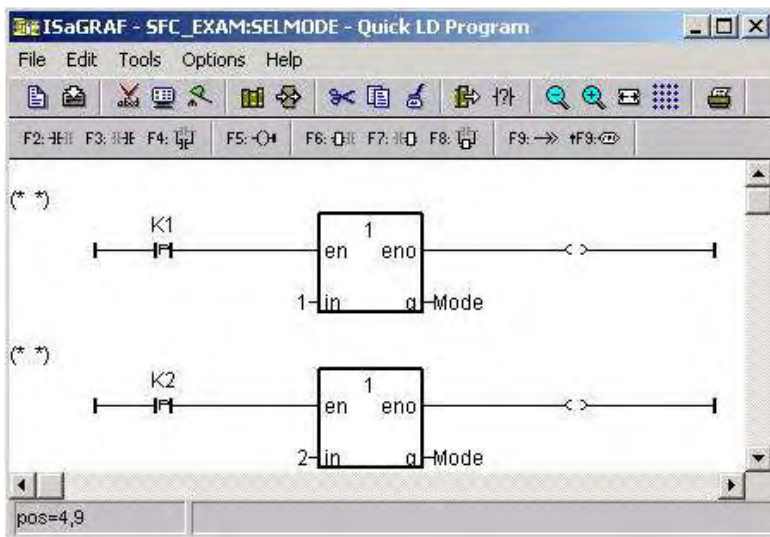
This block is very useful to directly link a diagram input and a diagram output. It can also be used (with a boolean negation line) to invert the state of a line connected to a diagram output.

即使 "EN" (input) 及 "ENO" (output) 參數 在上面的範例中並未使用過, 但當您在程式中加入 "1 Gain" 函式方塊時, 就多了這兩個變數。



您必須改變 "K1" 和 "K2" 的接點型態為 "P", 當 "P" 接點的左連接線的布林狀態由 false 變成 true 時, 線圈上的變數值就設定為真。將滑鼠移到 "Q" 點的右邊單擊一下, 輸入 "Mode"。將滑鼠移到 "in" 的左邊單擊一下, 輸入 1, 而下方的 "1Gain" 函式方塊的左邊 "in" 則輸入 2。

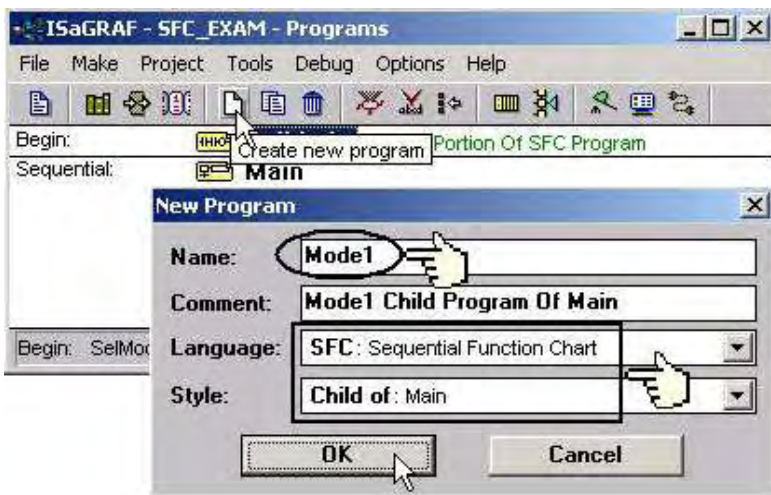
當您編輯完 "ISaGRAF Quick LD Program" 程式時, 應該如下圖所示:



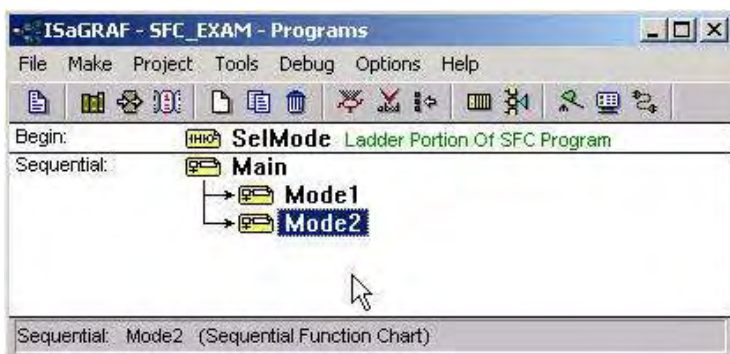
建立 “main” SFC 程式



建立 “Mode1” 子程式

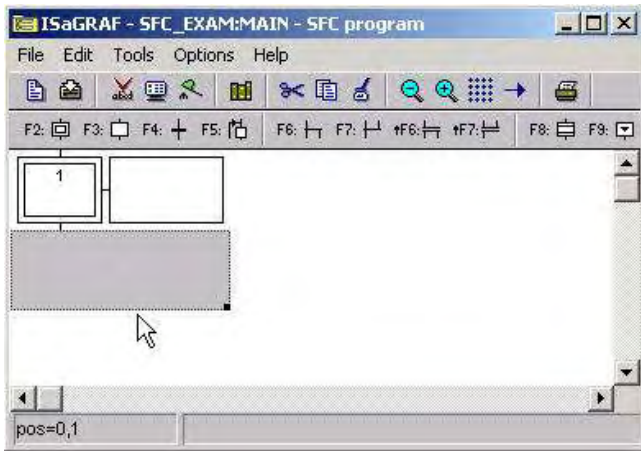


依同樣的方式，建立 “mode2” 子程式

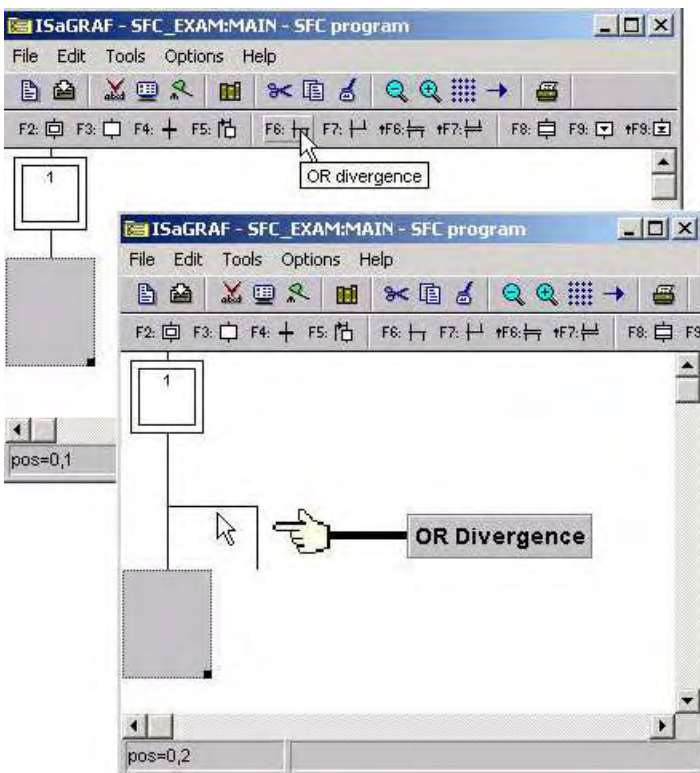


2.5.2: 編輯 SFC 程式

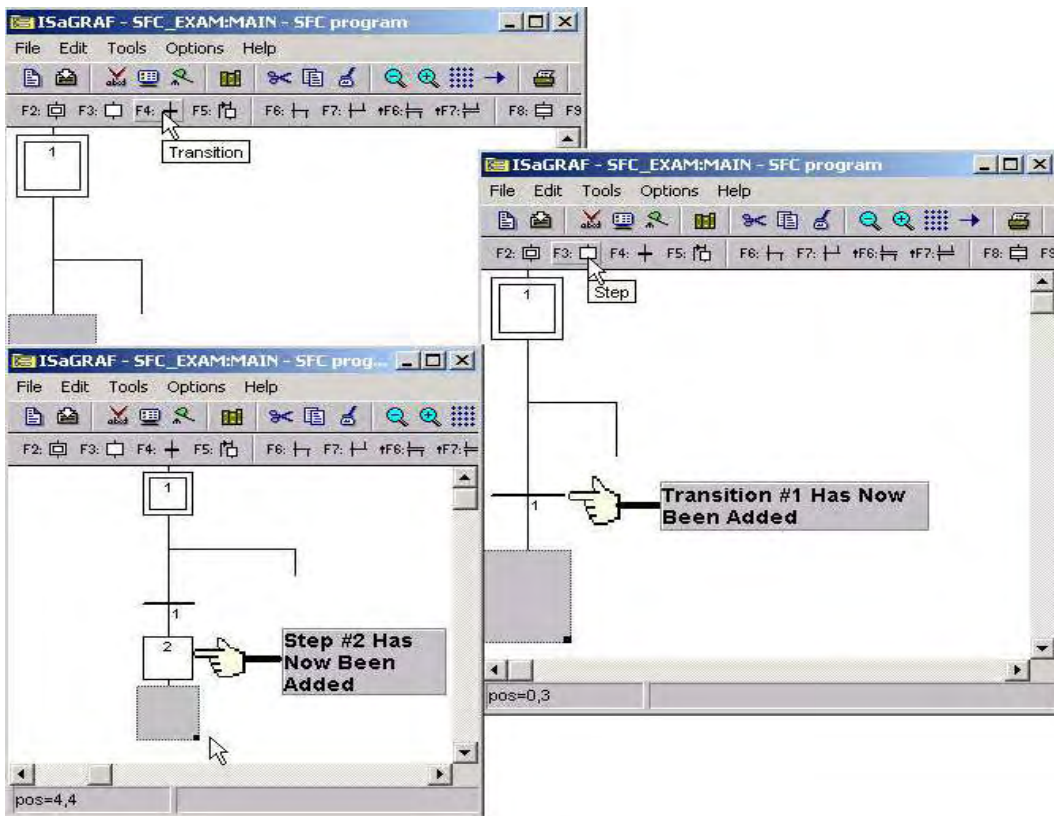
在 “ISaGRAF Programs” 視窗中雙擊 “main” 來編輯 SFC 程式。



您應該會注意到在初始步驟方塊的右邊有一塊空白的格子，這格子是用來在 FSC 程式中為步驟或轉移條件編寫註解用的。在 SFC 程式中，這個註解方塊並不是必須的所以您可以拖曳灰色方塊中的黑點來調整初始步驟適當的大小

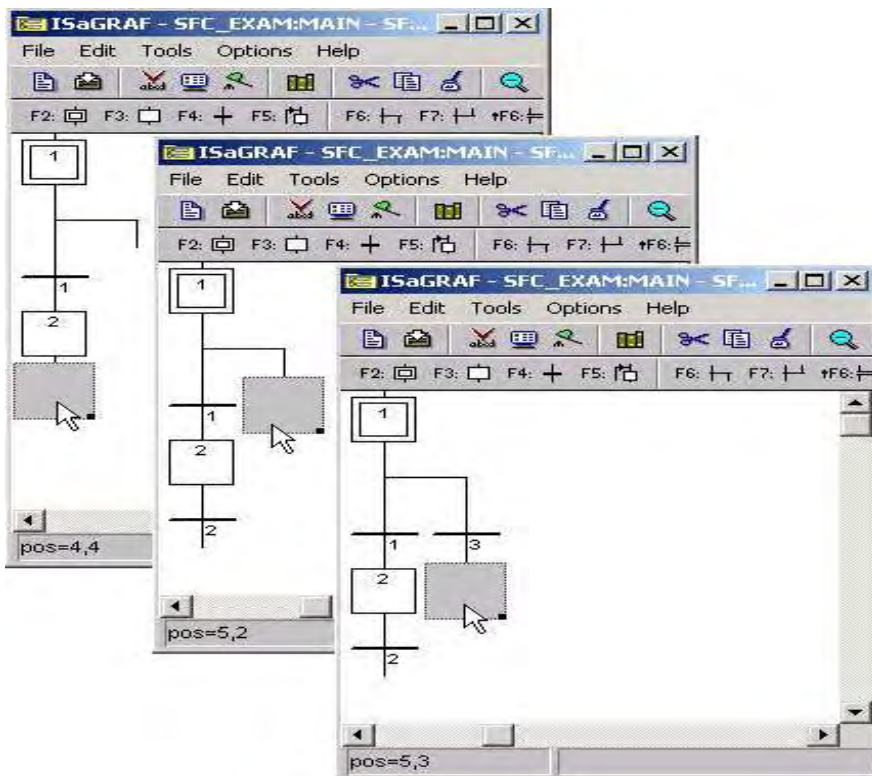


當您單擊 “OR Divergence”時，灰色的方塊會自動的往下移。接下來，單擊 “Transition”鍵來建立 “Transition 1”，然後按 “Step”鍵來建立 Step2，如下所示:

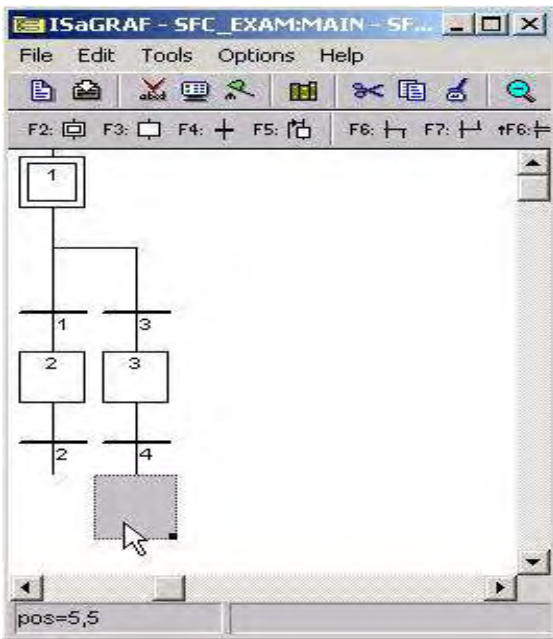


在灰色格子內按一下轉移鍵，來增加第二個轉移條件(transition 2)。在完成第 2 個轉移條件後，在 直接在“OR Divergence“在新增第 3 個轉移條件(transition 3)

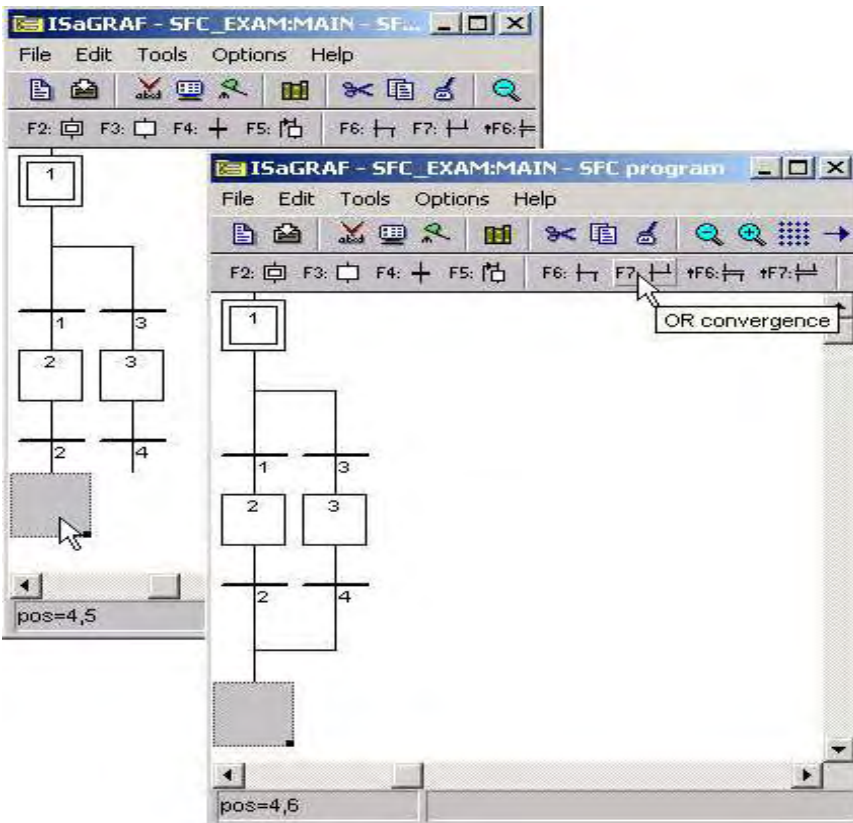
當您完成以上之設定時，應該如下圖所示：



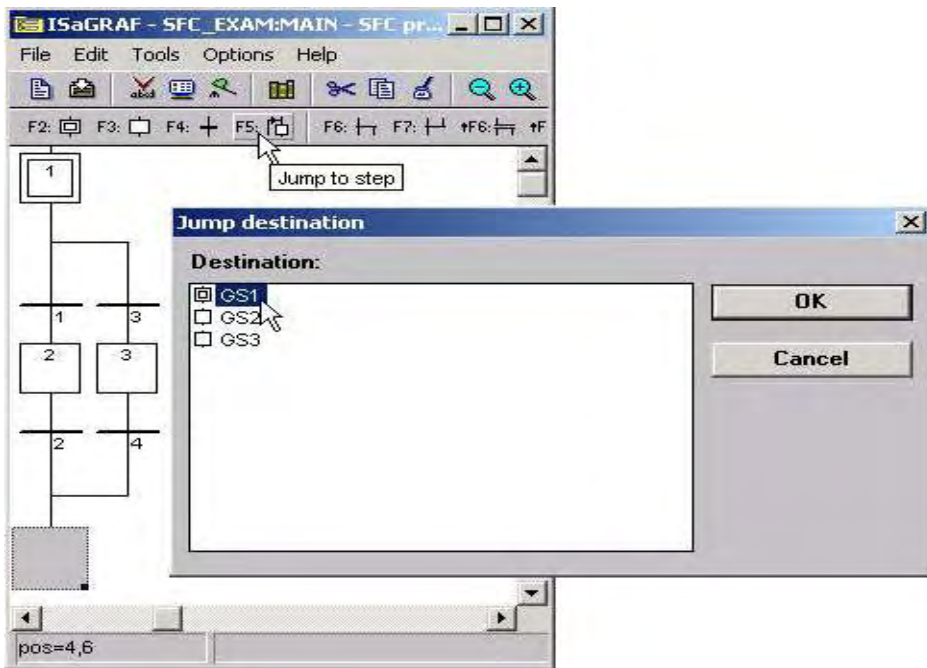
接下來在灰格子裡單擊”Step”來增加 Step 3，在增加完 Step 3 後，在 step 3 的下方再新增一個轉移條件(transition 4)，新增完後，應該如下圖所示：



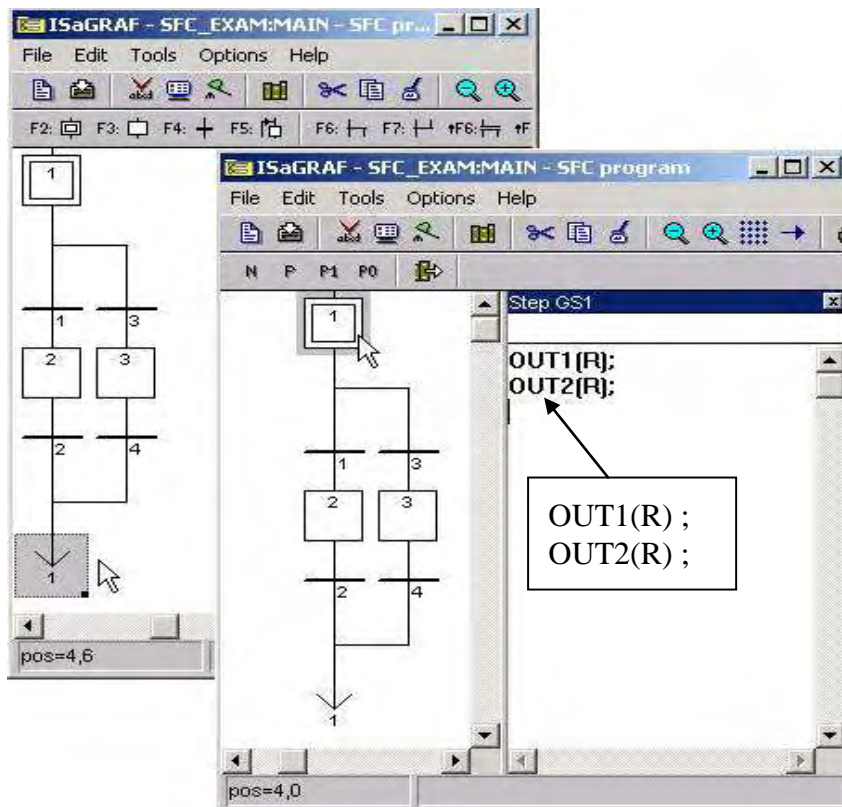
單擊轉移條件 2 下方的灰色方塊，然後按下 “OR Convergence” 鍵。



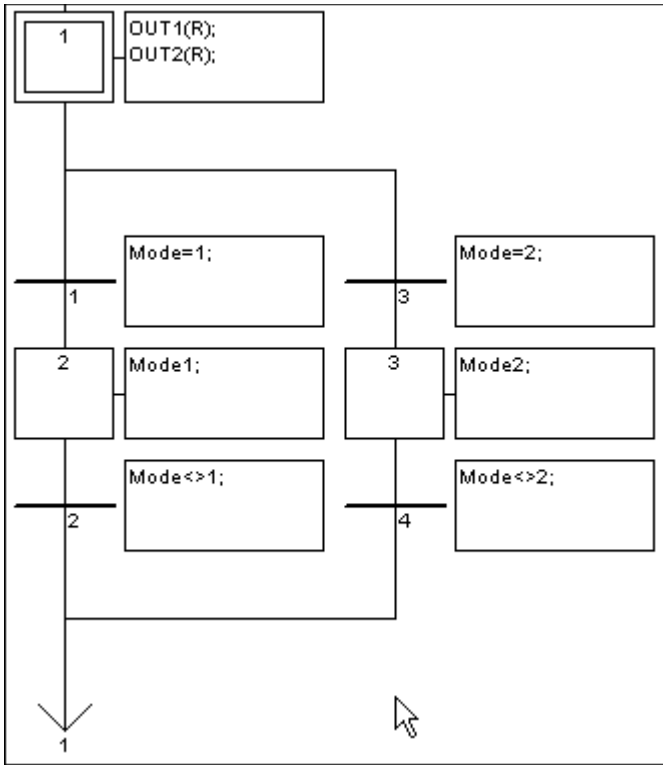
單擊 “Jump To step”視窗，將會產生如下圖所示的畫面，在 “Jump Destination”視窗內雙擊 “GS1”。



現在我們已經完成 SFC 專案中的 “Main” 程式，接下來我們要為每一個步驟和轉移條件編寫程式碼。雙擊 step #1 開啟 “ISaGRAF SFC 程式”將下圖所標示的程式碼填入，記得必須在每一行程式碼後加上一個分號(“;”) 作結尾。



依照上面的方式，在每一個步驟和轉移條件加上程式碼，如下圖所示：

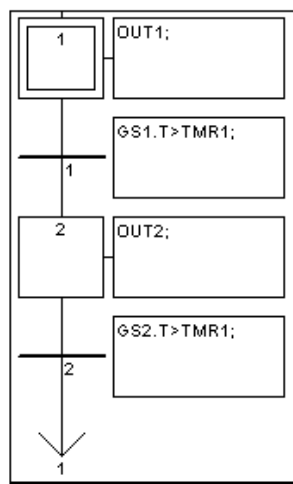
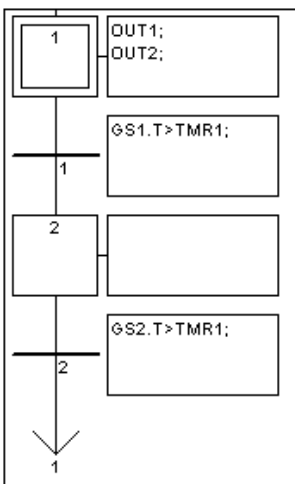


您現在已經完成 SFC 程式中的 main 主程式。

接下來您需要建立及編輯兩個子程式，編輯子程式的方法和編輯 main 程式是一樣的。當您建立兩個子程式後，應該如下圖所示：

SFC Child Program "Mode1"

SFC Child Program "Mode2"

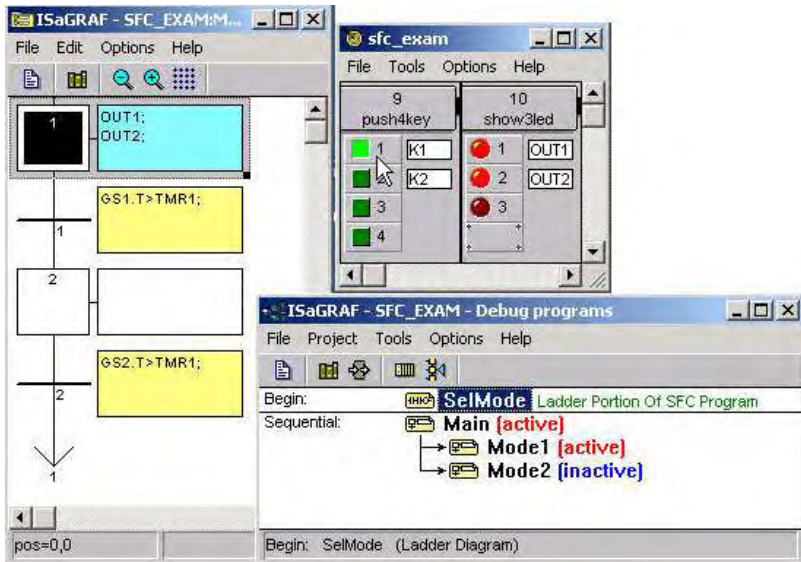


您還是須要 “Connecting I/O”及 “Compiling The Project”，請參考 2.1.2 節及 2.1.3 節。

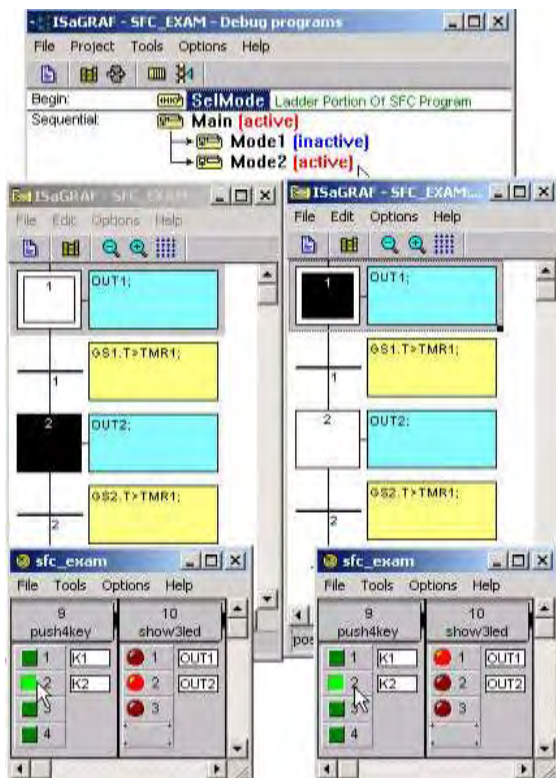
2.5.3: 模擬 SFC 程式

在您成功的編譯完 SFC 程式之後，您可以執行模擬的動作來觀察兩個子程式和 main 程式是如何運作的。

當 “K1”開啟時，“Mode 1”為 TRUE。“OUT1”及 “OUT2”同時開然後同時關時，此時”Mode2”為 FALSE。



當 “K2”開啟時 “Mode 2”為 TRUE，OUT1 會先開然後換 OUT2 開，此時 Mode 1 為 FALSE。



2.6: 使用 變數陣列

如果您的 ISaGRAF Workbench 為 3.4 或 3.5 版, 可以使用 變數陣列.

請先把所有開啟的 ISaGRAF Window 關閉. 之後在您的 ISaGRAF 所安裝的路徑, 更改以下的檔案內容, 此路徑通常是 c:\isawin\exe\

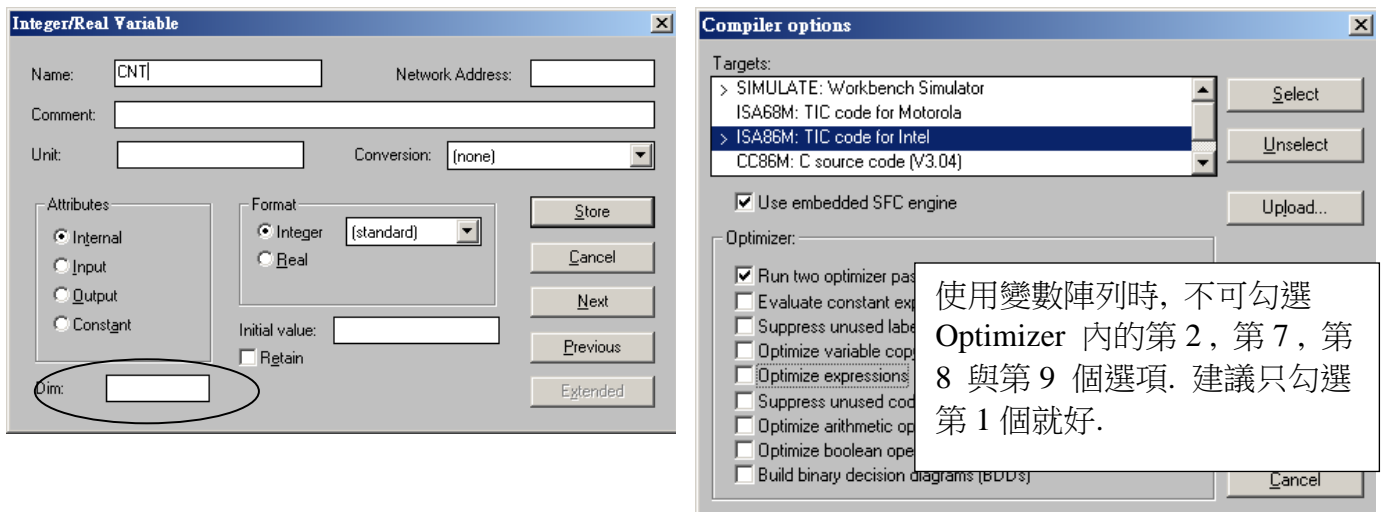
請在 C:\ISAWIN\EXE\ISA.INI 檔案的最上方, 新增 2 行(加第 3 行為空白行) 如下.

```
[DEBUG]
arrays=1
```

編輯好 “isa.ini” 檔案後儲存它. 之後當你開啟 ISaGRAF 內的 Dictionary 時, 您會發現每個變數宣告 Window 內多了一個 “Dim” 欄位. 可填入的數字為 1 到 255.

非常重要:

1. 當需要使用 變數陣列 時, 請永遠只宣告真正所需要的 Dim 數量. 宣告過多的 Dim 會消耗 控制器過多的記憶容量, 特別是像 I-7188EG/XG, I-8xx7 這種 記憶容量比較受限的 Controller.
2. 使用變數陣列時, 不可勾選 Make – Compiler options - Optimizer 內的第 2, 第 7, 第 8 與第 9 個選項. 建議只勾選 第 1 個 – “Run two optimizer passes” 就好. 不然變數陣列的值會發生錯誤.



變數陣列的第 1 個元素由 0 開始. 比如若宣告一個整數 變數陣列 CNT, 設其 Dim 欄位為 10, 此變數陣列將會顯示為 CNT[0..9], 表示它可以使用的元素為 CNT[0], CNT[1], ..., CNT[9] 共 10 個元素.

如何在程式內使用 變數陣列 ？

以下範例程式使用 ST 語法, 將初值 100 到 109 指定給 CNT[0], ... , 到 CNT[9]

(* INIT 宣告為 internal Boolean 變數, 且宣告初值為 TRUE *)

(* CNT 宣告為 internal integer 變數陣列, “Dim” 欄位為 10 *)

(* ii 宣告為 internal integer 變數 *)

```
IF INIT THEN
```

```
  INIT := FALSE ; (* INIT 設為 False, 這些 code 只有第 1 個 PLC scan 會執行一次而已 *)
```

```
  For ii := 0 to 9 do
```

```
    CNT[ii] := 100 + ii ;
```

```
  End_For ;
```

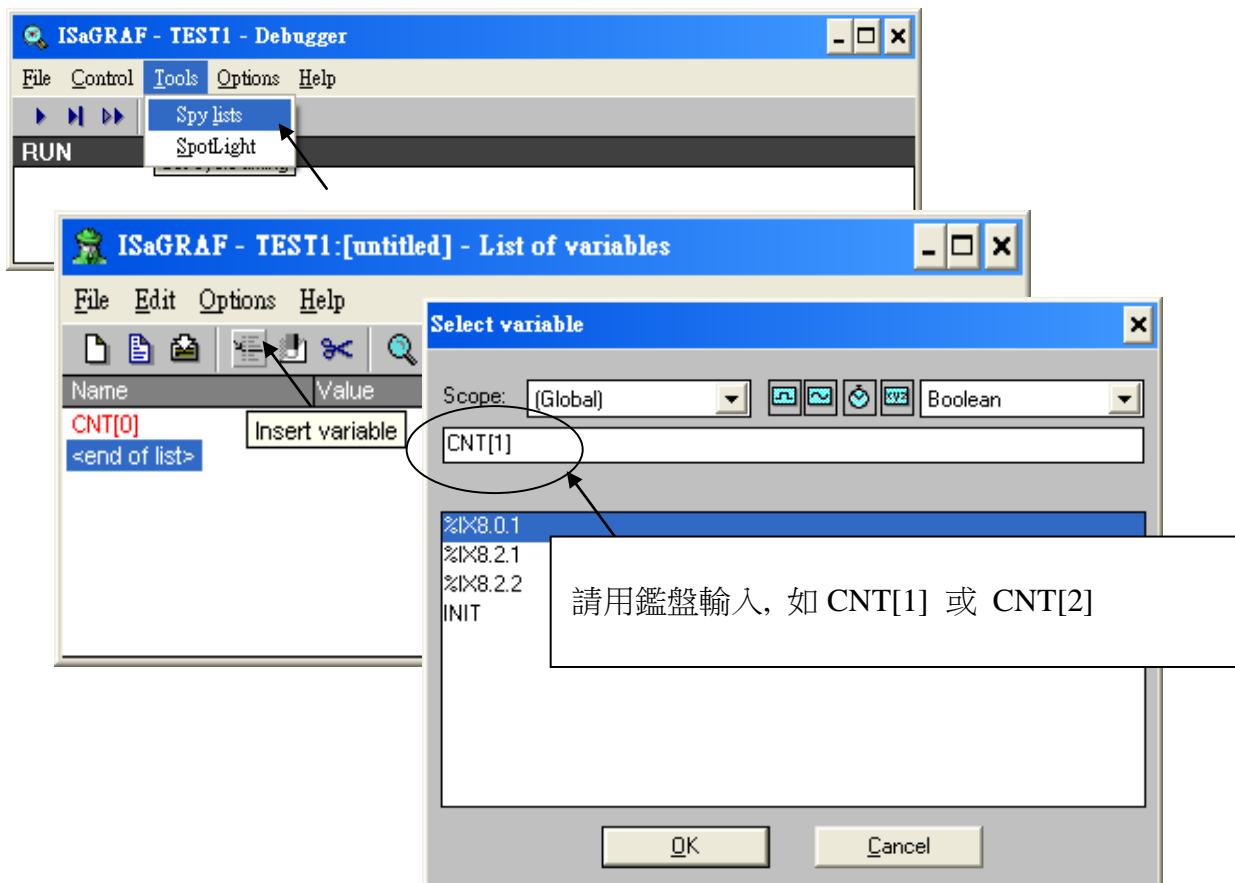
```
END_IF ;
```

非常重要: 請不要在程式內使用超過 變數陣列 所宣告的合法元素數量. 比如上方的例子, CNT 變數陣列的 “Dim” 欄位宣告為 10, 所以可以使用的合法元素為 CNT[0], CNT[1], ..., CNT[9]. 如果在程式內去使用 CNT[10], CNT[11], ... 將會導致 Controller 發生錯誤.

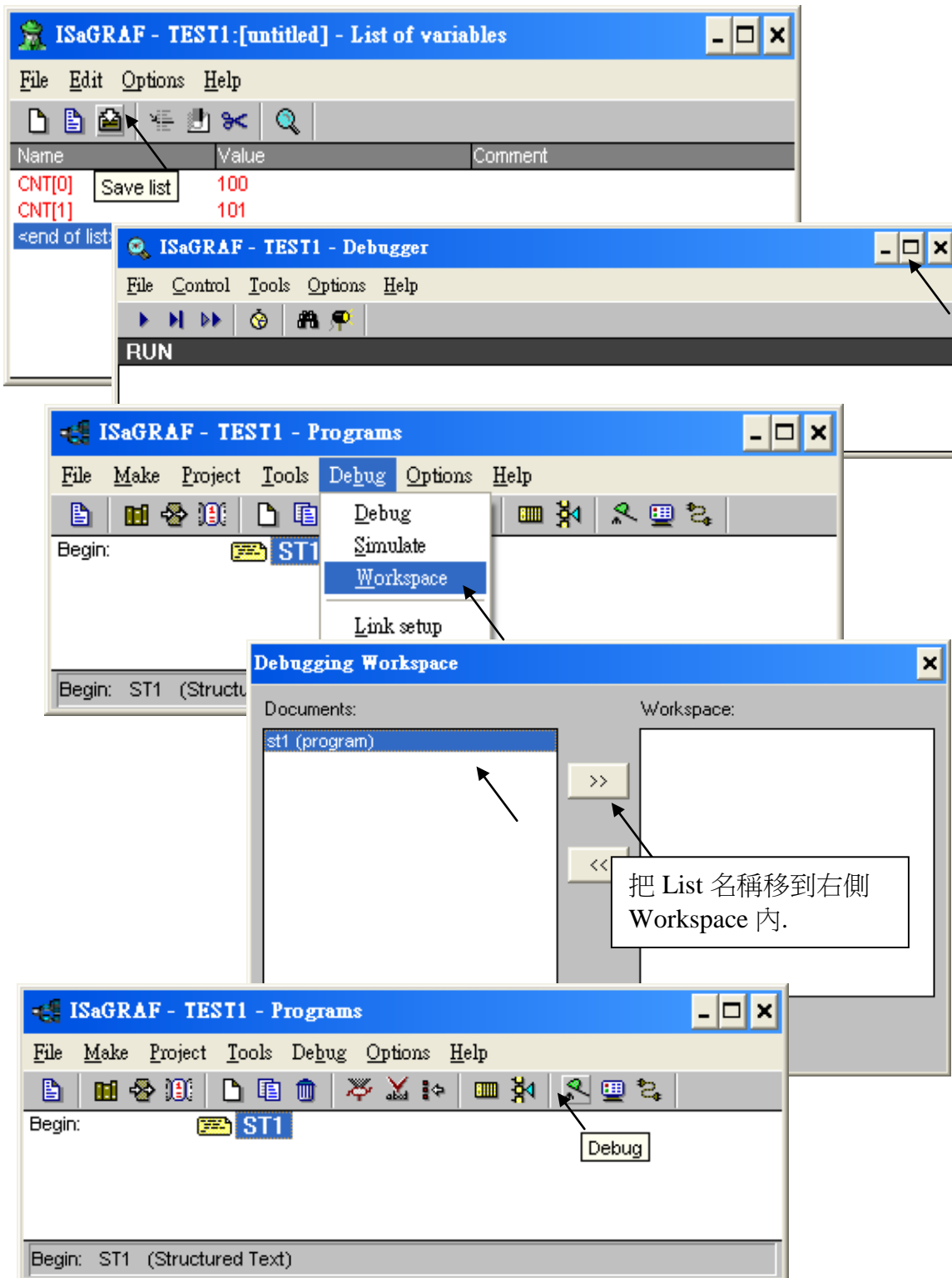
如何 監看 / 除錯 變數陣列 ？

當程式寫完並編譯(Compile) 完, 成功下載到 Controller 後. 可以在 “Debugger” 視窗, 使用 “Tools” - “Spy lists” 工具來監看 變數陣列 的值. (關於 “Spy Lists” 的更多使用說明, 請參考第 9.12 節).

請在 Spy Lists 視窗內輸入變數陣列的元素名字, 如下.

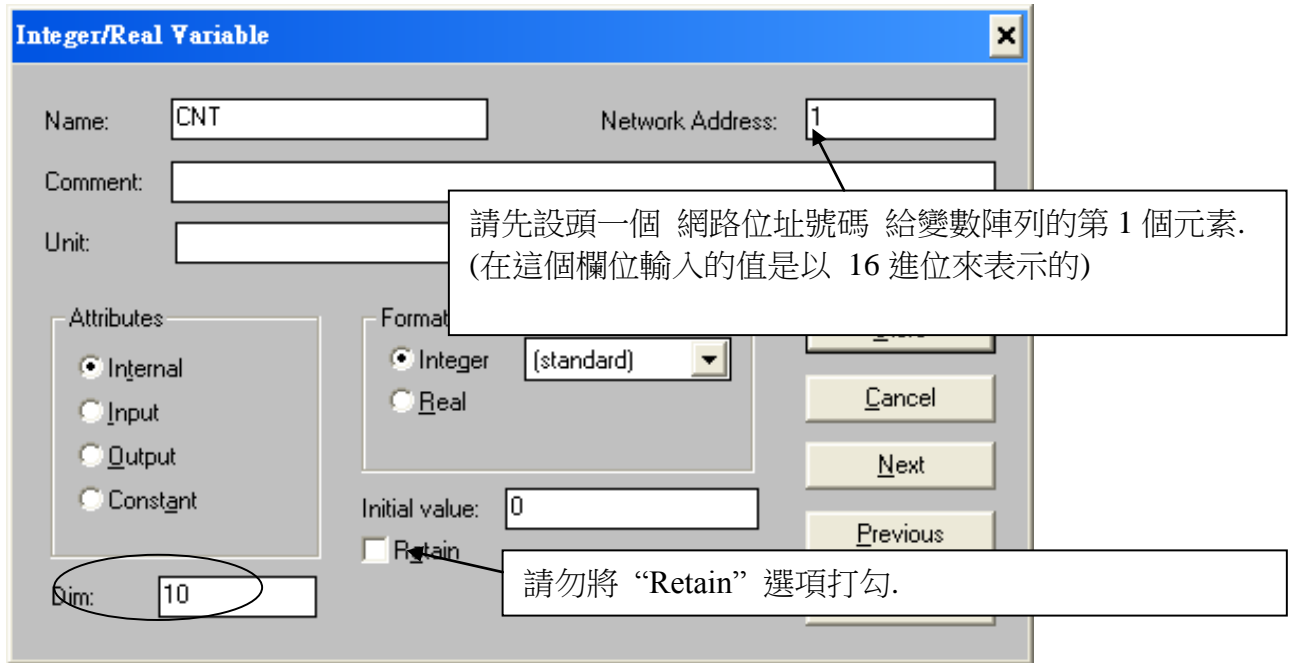


輸入完所有想監看的名字後，請記得將此“spy list”儲存起來，比如存成“list1”，之後可移到“workspace”內。如此每次用 ISaGRAF 跟此 Controller 連線時，“list1”的內容就會自動顯示出來。



2.6.1 指定 網路位址號碼 給變數陣列

要指定 網路位址號碼 (變數的“網路位址號碼”, Network Address No. 的用途為用來跟其它 SCADA / HMI 交換資料, 請參考第 4 章的說明) 給變數陣列, 請先設頭一個網路位址號碼給變數陣列的第 1 個元素, 比如設 1 號給 CNT[0]. 然後可使用“S_MB_ADR()” 函式來指定給其它元素.



以下為使用 “S_MB_ADR()” 的例子

1. 設定連續的 網路位址號碼 1, 2, 3, ..., 10 給 CNT[0], CNT[1], CNT[2], ..., CNT[9]

(* INIT 宣告為 internal Boolean 變數, 並宣告初值為 TRUE *)

(* TMP 宣告為 internal Boolean 變數 *)

IF INIT THEN

INIT := FALSE ; (* 只有第 1 個 PLC scan 會執行到 *)

TMP := S_MB_ADR(1, 10, 0) ; (* 從網路位址 1 的 變數陣列 起, 指定 10 個連續號. *)

END_IF ;

2. 設定跳 1 號的 網路位址號碼 1, 3, 5, ..., 19 給 CNT[0], CNT[1], CNT[2], ..., CNT[9]

(* INIT 宣告為 internal Boolean 變數, 並宣告初值為 TRUE *)

(* TMP 宣告為 internal Boolean 變數 *)

IF INIT THEN

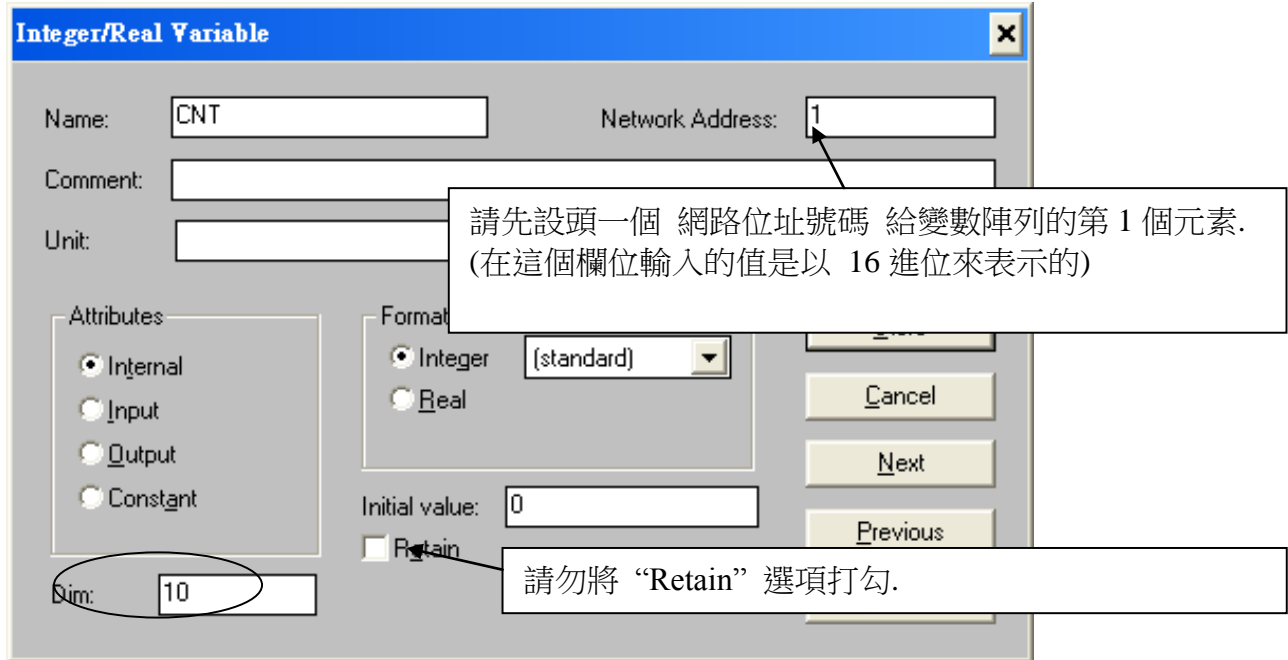
INIT := FALSE ; (* 只有第 1 個 PLC scan 會執行到 *)

TMP := S_MB_ADR(1, 10, 1) ; (* 從網路位址 1 的 變數陣列 起, 指定 10 個跳號. *)

END_IF ;

2.6.2 將 變數陣列 使用成 可保留變數

要將 變數陣列使用成 可保留變數, 請先設頭一個 網路位址號碼 給變數陣列的第 1 個元素, 比如設 1 號給 CNT[0]. 之後可使用 “Retain_A()” 函式來指定 如下.(關於 新的可留變數, 請參考第 10.1 節的說明).



以下的程式將 Integer 變數陣列 CNT[0..9] 設定為 可保留變數, 其可保留位址從 20, 21, ... 到 29.

(* INIT 宣告為 internal Boolean 變數, 並宣告初值為 TRUE *)

(* TMP 宣告為 internal Boolean 變數 *)

IF INIT THEN

INIT := FALSE ; (* only do it at 1st PLC scan *)

TMP := Retain_A('N', 1, 10, 20) ;

(* 第 1 個參數 : 'B' : 表示變數陣列為 boolean 型態, 'N' : Integer, 'F' : Real, 'T' : Timer

第 2 個參數 : 變數陣列的第 1 個元素的 網路位址號碼 .

第 3 個參數 : 可以是 1 - 255, 想要設定成可保留變數 的元素數量.

第 4 個參數 : 使用的 可保留位址從幾號開始 ..

7188EG/XG + X607/608, I-8xx7 + S256/512, iP-8xx7 : 'B' 與 'T' 是 1 到 256,

'N' 與 'F' 是 1 到 1024 .

WP-8xx7, XP-8xx7-CE6, VP-25W7 : 'B' 與 'T' 是 1 到 1024, 'N' 與 'F' 是 1 到 4096 *)

END_IF ;

第 3 章 建立 I/O 連結

在您使用 ISaGRAF 來控制 iP-8xx7, I-8xx7, I-7188EG/XG , μPAC-7186EG, WP-8xx7, VP-25W7, XP-8xx7-CE6 控制系統之前,您必需先確定 I/O 函式庫是否已經安裝完成。假如您還未安裝,請參閱 1.2 節的"安裝 ICP DAS Utilities For ISaGRAF",裡面有詳細的說明引導你如何安裝函式庫。

請訪問以下網址 來取得更多 I-8K 與 I-87K I/O 卡片的說明
http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm

以下是可使用的 Slot 0 到 7 的 I-8K parallel I/O :

注意:

卡片有分為低板卡與高板卡,高板卡名稱後方會有一個 W,例如 I-87057 是低板卡,而 I-87057W 是高板卡. iP-8xx7, WP-8xx7, XP-8xx7-CE6, VP-2117 與 VP-25W7 / 23W7 只能使用高板卡插在主機旁的 I/O slot 編號 0 ~ 7. 若將低板卡插入這些 PAC 內有可能會發生 串口通訊不正常, Retain 變數異常, I/O 不正確 .

I-8K parallel I/O 卡片只可插在 PAC 的第 0 到 7 槽 (XP-8xx7-CE6 是第 1 到 7 槽上使用),不可插在 I-87K4/5/8/9 或 RU87Pn 等擴充槽內使用.

數位 輸入 / 輸出 (Digital Input / Output)	I-8037W, I-8040W, I-8040PW , I-8041W , I-8042W, I-8046W, I-8050W , I-8051W , I-8052W , I-8053W, I-8053PW, I-8054W , I-8055W , I-8056W , I-8057W, I-8058W, I-8060W , I-8063W, I-8064W , I-8068W , I-8069W
類比 輸入 (V , mA) (Anlog Input)	I-8017HW (8 Ch. Differential / 16-Ch. Single-end) iP-8xx7: I-8017HW Sample rate 最快約 125Hz (8-Ch.) , 62Hz (16-Ch.) WP-8xx7, XP-8xx7-CE6, VP-25W7: I-8017HW Sample rate 最快約 200Hz (8-Ch.) , 100Hz (16-Ch.) 跟 scan time 有關, 越大的 scan time, Sample rate 小
類比 輸出 (V , mA)	I-8024W (4 Ch.)
高速計數器 / 頻率輸 入(Counter / Frequency)	I-8084W (4/8 Ch counter or Frequency) I-8088W (8 Ch. PWM output)
馬達控制卡 , Encoder	I-8092F, I-8094, I-8094F : 只有 XP-8347-CE6 與 XP-8747-CE6 有支持. 參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-132 . I-8091W (2 Axes. Pulse output) , I-8090W (Encorder) : iP-8xx7, WP-8xx7, VP-25W7 與 XP-8xx7-CE6 有支持. I-8093W(Encoder) : iP-8xx7, WP-8xx7, VP-25W7 與 XP-8xx7-CE6 有支持
CANopen Master	I-8123W: 只有 XP-8xx7-CE6, WP-8xx7 與 VP-25W7, VP-23W7 有支持 (參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-145)
FRNET Master 卡	I-8172W (2-port FRNET) , 需搭配 FRNET I/O 使用, 請訪問 http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm 或 http://www.icpdas.com/faq/isagraf.htm FAQ-048
擴充 serial 通訊埠	I-8112iW (2-Ch. RS-232) , I-8114iW (4-Ch. RS-232) I-8142iW (2-Ch. RS-485 / 422) , I-8144iW (4-Ch. RS-485 / 422) (I-8xx7 , iP-8xx7 最多可擴充 16 埠,可插在第 0 到 4 槽) (WP-8xx7, VP-25W7 最多可擴充 10 埠, 可插在第 0 到 2 槽) (XP-8xx7-CE6 最多可擴充 28 埠, 可插在第 1 到 7 槽)

以下是可用在 Slot 0 到 7 的 I-87K parallel I/O ：

數位 輸入 / 輸出 (Digital Input / Output)	I-87040W, I-87041W, I-87046W, I-87051W, I-87052W, I-87053W, I-87053PW, I-87053W-A5, I-87053W-AC1, I-87053W-E5, I-87054W, I-87055W, I-87057W, I-87057PW, I-87058W, I-87059W, I-87061W, I-87063W, I-87064W, I-87065W, I-87066W, I-87068W, I-87069W
類比 輸入 (V, mA) (Anlog Input)	I-87016W, I-87017W, I-87017W-A5, I-87017RW, I-87017RCW, I-87017DW, I-87017ZW, I-87019PW, I-87019RW, I-87019ZW
類比 輸出 (V, mA) (Anlog Output)	I-87024W, I-87024DW, I-87024RW, I-87028CW
多種類 I/O 卡	I-87026PW
溫度 輸入 (Temperature input)	I-87005W, I-87013W, I-87015W, I-87015PW, I-87018W, I-87018RW, I-87018PW, I-87018ZW, I-87019RW, I-87019ZW
高速計數器 / 頻率, PWM	I-87082W, I-87084W, I-87088W (PWM)
HART Master	I-87H17W: 只有 XP-8xx7-CE6, WP-8xx7 與 VP-25W7, VP-23W7 有支持 (參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-136)
Vibrating Wire Input	I-87089W (參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-091)

注意:

卡片有分為低板卡與高板卡，高板卡名稱後方會有一個 W，例如 I-87057 是低板卡，而 I-87057W 是高板卡。iP-8xx7, WP-8xx7, XP-8xx7-CE6, VP-2117 與 VP-25W7 / 23W7 只能使用高板卡插在主機旁的 I/O slot 編號 0 ~ 7。若將低板卡插入這些 PAC 內有可能會發生 串口通訊不正常, Retain 變數異常, I/O 不正確。

I-8K parallel I/O 卡片只可插在 PAC 的第 0 到 7 槽 (XP-8xx7-CE6 是第 1 到 7 槽上使用), 不可插在 I-87K4/5/8/9 或 RU87Pn 等擴充槽內使用。

關於 I-7188EG/XG 或 μ PAC-7186EG 可使用的 X board, 請參考該本“快速上手手冊”或 http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm

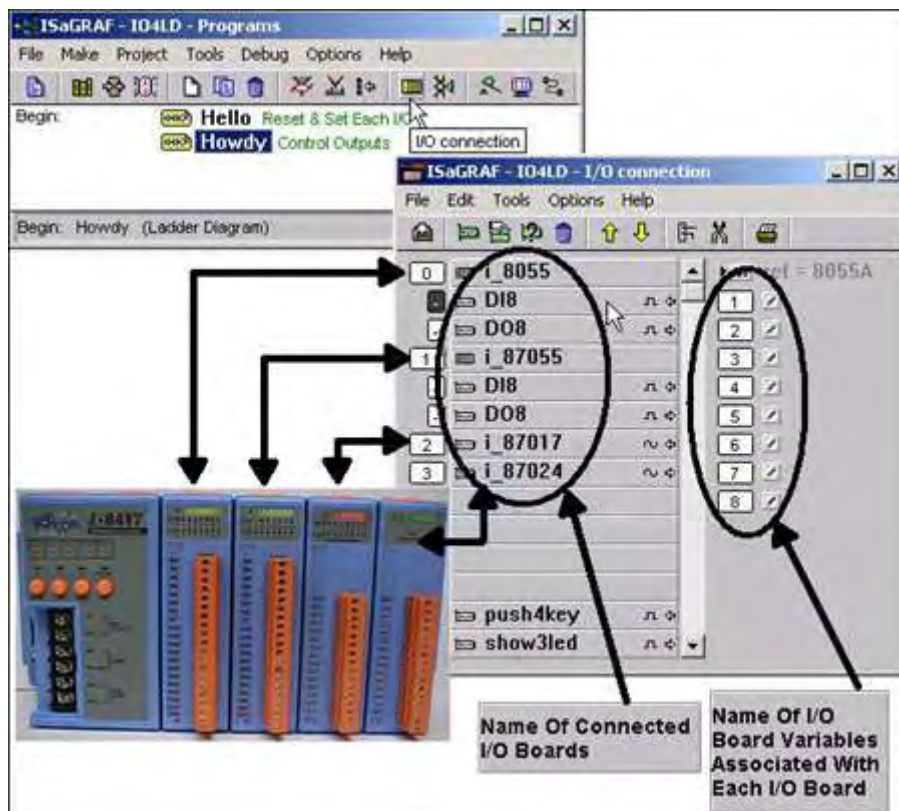
請訪問以下網址 來取得更多 I-8K 與 I-87K I/O 卡片的說明 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm

3.1 如何將 I/O 控制模組引入 ISaGRAF 專案

在您開始引用 ISaGRAF PAC 的 I/O 板卡到 ISaGRAF 專案之前，您必需在 ISaGRAF 專案中將 I-I/O 板連結起來。在”I/O connections” 視窗最左邊的數字代表著控制器的插槽號碼、插槽 0 到 7 只提供給 PAC 的 I/O 控制板使用（注意: XP-8xx7-CE6, Wincon-8xx7 沒有插槽 0,是插槽 1 到 7），編號 8 以上的插槽則可以提供給其他虛擬控制板,例如 ”Push4Key” 和 “Show3Led”使用。對於 I-7188EG/XG, μ PAC-7186EG 而言，插槽 0 是給 X-xxx board (例如 X-107) 使用，編號 1 以上的插槽給其他虛擬控制板使用。

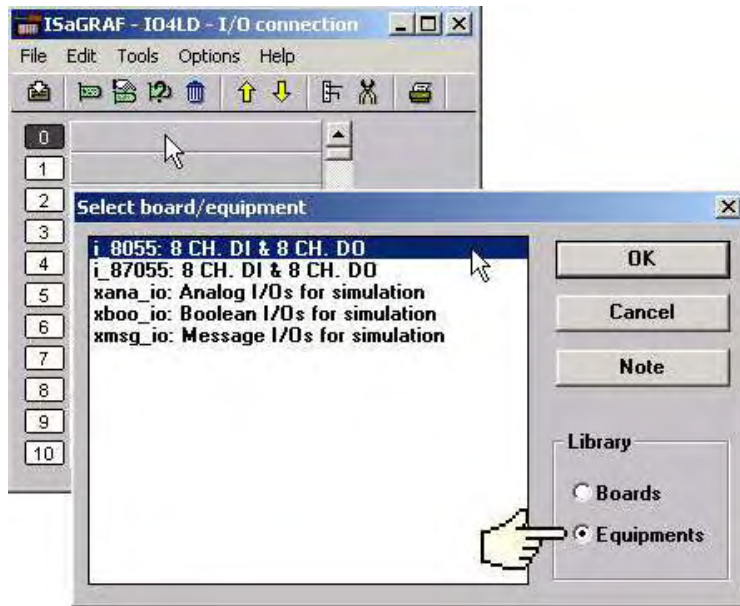
在下面的例子中，我們使用 I-8417 控制器並插著下面所列出的控制板。

- 插槽 0: I-8055 (8 個 D/I 及 8 個 D/O)
- 插槽 1: I-87055(8 個 D/I 入 及 8 個 D/O)
- 插槽 2: I-87017(8 個 A/I)
- 插槽 3: I-87024(4 個 A/O)
- 插槽 8: "Push4Key"
- 插槽 9: "Show3Led"



3.1.1: 連結 I/O 板

在”I/O Connection”視窗內的插槽上用滑鼠雙擊將會開啟所有可連結的 I/O 板，您可以在指定的插槽上選擇你所需要的 I/O 板來作連結。ISaGRAF 控制器的函式庫內定義兩種 I/O 板的格式，即 “Boards”和 “Equipment”。”Boards”即單一型態的板子，就是在板子上的輸出入點都是單一型態或單一屬性。“Equipment”即多種型態的板子，即板子上的輸出入點是多種型態的(例如 I-8055 就具有 8 個數位輸出和 8 個數位輸入)。在開始連結 I/O 板的過程中，先在插槽上雙擊以便選擇你所需要的 I/O 板。



假如您把 I/O 板設錯插槽，您可以用”Clear Slot”把這個錯誤連結砍掉，接下來在你所想要的插槽上設置正確的 I/O 板連結。

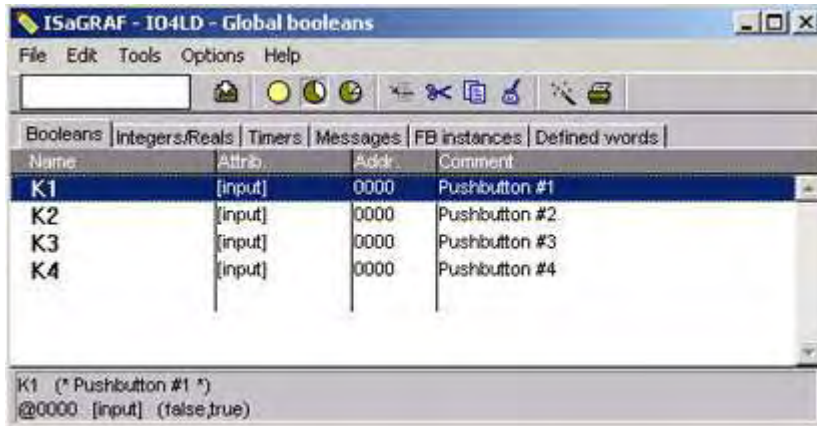


3.1.2: 連結輸入及輸出變數

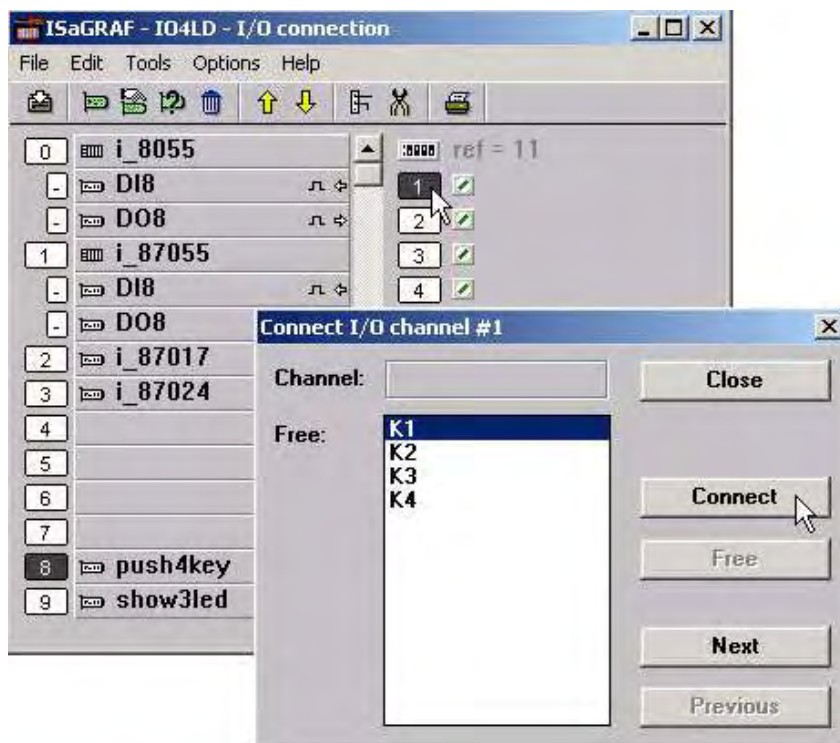
所有的輸出,輸入板的 ”變數(或名字)”一定要連結到 ”I/O Connection” 視窗內。按下你所要設定屬性的插槽(slot), 雙擊 ”I/O connection” 右手邊的 I/O 點的號碼, 選擇變數名字, 接著按下 ”connect” 鍵即完成變數連結。

注意:

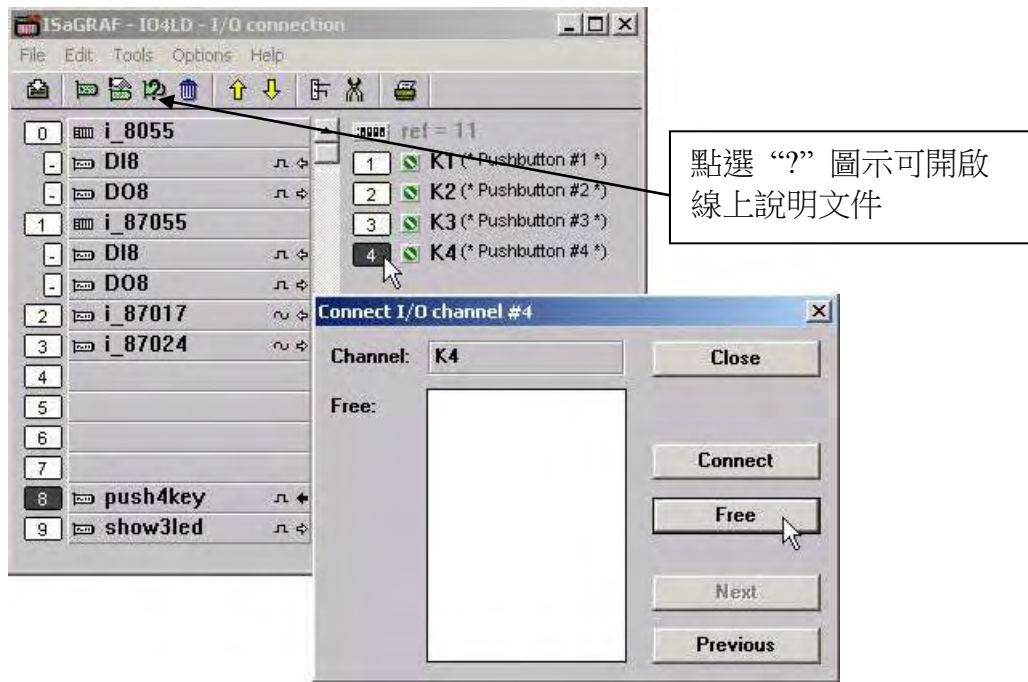
在您指定變數給各個 I/O 板的輸出入點時, 記得要先在”ISaGRAF Global Variable”視窗內對每個變數做宣告。



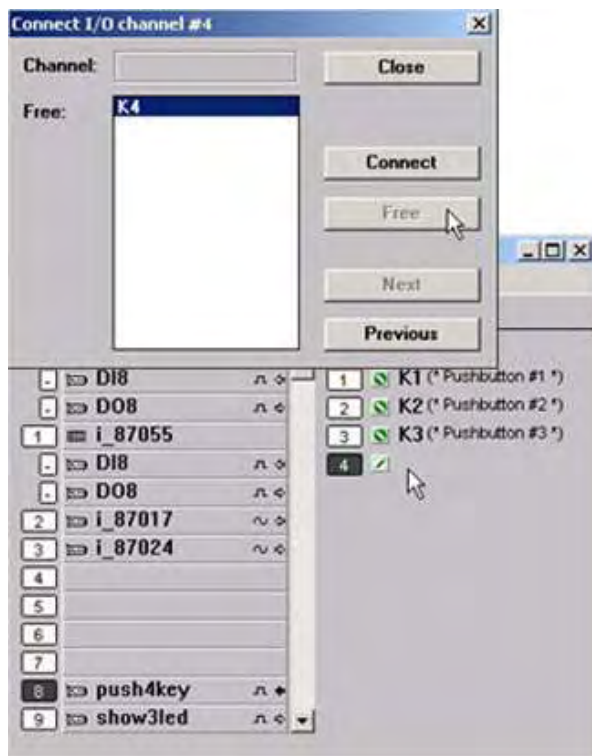
以滑鼠點一下插槽 8, 接著在”IsaGraf I/O Connection”視窗的右手邊雙擊 ”1”, ”Connect I/O Channel #1” 將會被開啟, 按下 ”connect” 鍵來建立變數 ”K1” 和 ”Push4Key” 的第一個輸入點的連結。



假如你連結輸入變數或輸出變數到不是你期望的 I/O 位置，雙擊您希望移除的輸入點。"Connect I/O Channel #x" 將會開啟，接著按下"Free"鍵，即可把變數移除掉。



當你按下"Free"鍵你將會看到變數從輸出輸入點移到"ISaGRAF I/O Connect"視窗內，此變數可以重新指定給其他位置的輸出入點使用。



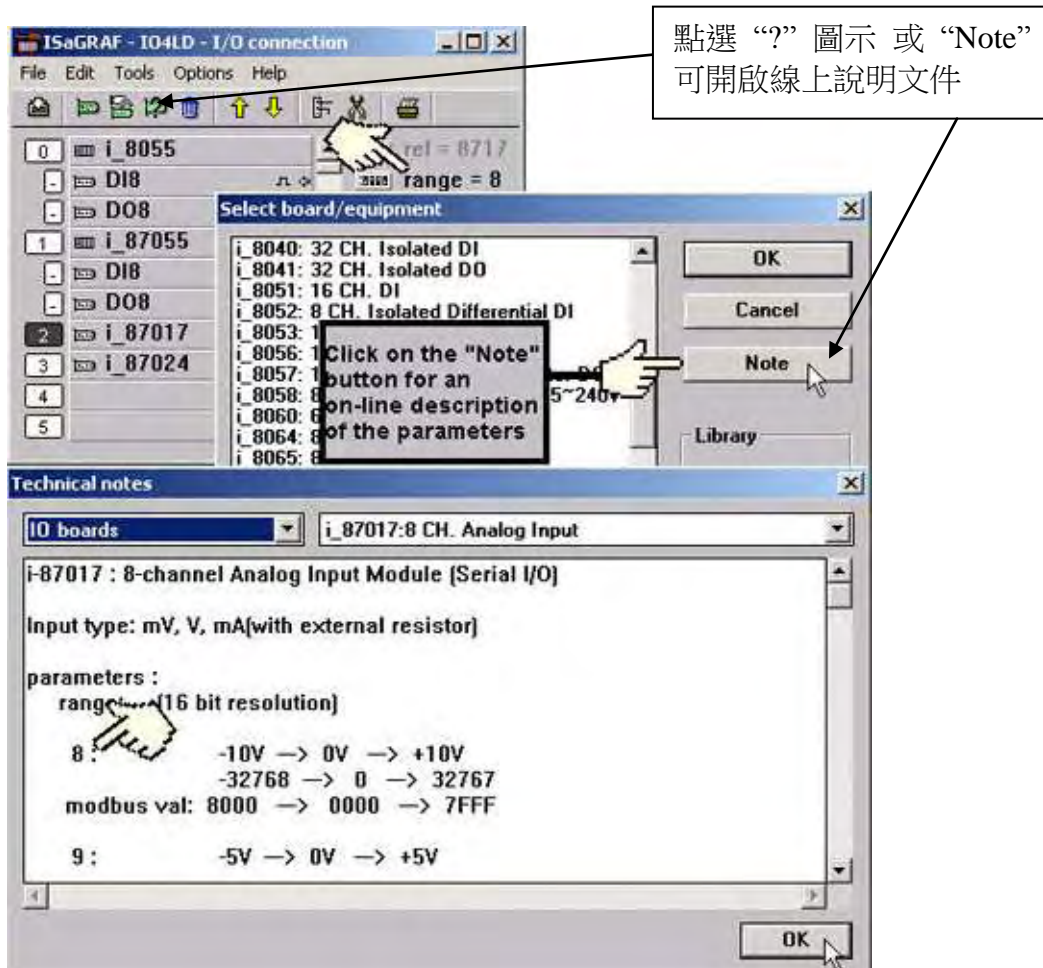
3.2: 連結類比型態的 I/O 板

注意:

1. 請不要選用 I-87017 與 I-87017C, 改選用適合工業環境用的 I-87017ZW 較好. I-87017ZW 可以用 jumper 調整為 10-Ch. Differential input 或 20-Ch. Single-end input, 且每個 channel 可以設為不同的 range 比較好用, 設為 10-Ch. 時可量測 0~20 mA, 4 ~ 20 mA, +/- 20mA, 且不需外接 125 ohm 電阻, 使用 jumper 調整就好. (I-87017W, I-87017RW, I-87017DW 量電流需外加 125 ohm 電阻)
2. 請不要選用 I-87018, I-87018W, 改選用適合工業環境用的 I-87018ZW 或 I-87019ZW 較好, 量測值也較準確.

3.2.1: 設定 類比 I/O 卡的 Range 參數 與 類比 IO 數值轉換函式

連結類比型態的 I/O 板到 ISaGRAF 控制器的方法和連結數位 I/O 板的方式大同小異。最主要的不同點在於類比 I/O 板的“range”參數必須要設定。



修改類比 I/O 板的 “range” 參數的方法為，在 ”ISaGRAF I/O Connection” 視窗中雙擊 “Range”，然後 ”I/O Board Parameter” 視窗將會開啟，輸入您所想要的 range 範圍。



在下面的表格裡有幾項關於 “range” 參數的選項。在註記內的預設值是 ”8”，代表的意思是您可以將 -10v 到 10v 的訊號值對映成 -32768~32768 的數值。若參數改為 ”9”，代表的意思為您可將 -5v ~ 5v 的訊號值對映成 -32768 ~ 32767 的數值。

若 “range” 參數改為 ”A”、代表的意思為您可將 -1v~1v 的訊號值對映成 -32768~32767 的數值。這 ”range” 數值在類比 I/O 的應用程式中可以處理很多範圍的反應控制(如溫度控制)

range: [16 bit resolution]	
8:	-10V → 0V → +10V -32768 → 0 → 32767 modbus val: 8000 → 0000 → 7FFF
9:	-5V → 0V → +5V -32768 → 0 → 32767 modbus val: 8000 → 0000 → 7FFF
A:	-1V → 0V → +1V -32768 → 0 → 32767

請參考 Appendix D - “Table of The Analog IO Value”，您可以得到一些不同型態的類比 I/O 板和他所對應的 “range” 值。

注意: 類比 IO 數值轉換函式:

有幾個好用的函式 可用來將 類比卡片 的 輸入/輸出 值 (通常是 -32768 ~ +32767) 轉換成 工程上的應用值. 比如想將 4 到 20 mA 轉換成 0 ~ 1000 Psi, 可以使用 “A4_20_To” 函式, 詳細的函式 說明, 請參考 附錄 A.4

1. **A4_20_To**: 轉換 4 ~ 20 mA 的類比輸入值(整數) 為 工程上的應用值(浮點數)
2. **V0_10_To**: 轉換 0 ~ 10 V 的類比輸入值(整數) 為 工程上的應用值(浮點數)
3. **To_A4_20**: 轉換 工程上的應用值(浮點數) 為 4 ~ 20 mA 的類比輸出值(整數)
4. **To_V0_10**: 轉換 工程上的應用值(浮點數) 為 0 ~ 10 V 的類比輸出值(整數)
5. **BIN2ENG**: 轉換類比輸入值(整數, 值需在 -32768 ~ +32767 之間) 為 工程上的應用值 (整數, 值需在 -32767 ~ +32768 之間)

3.2.2: 設定特別的“range”參數來直接取得溫度卡片的“攝氏度”或“華式度”值

ICP DAS 提供很多溫度輸入板卡(模組), 如下:

(I-7000 系列 I/O 只可用於 RS-485 遠程 I/O, 不可用於 第 0 ~ 7 槽)

有“感測器斷線偵測功能”的溫度模組型號:

Thermocouple 型式:	I-87018RW, 87018PW, 87018ZW, 87019RW, 87019PW, 87019ZW, I-7018R, 7018BL, 7019, 7019R, 7018Z
RTD 型式:	I-87013W, 87015W, 87015PW, I-7013, 7015, 7033
Thermister 型式:	I-87005W, I-7005

無“感測器斷線偵測功能”的模組型號:

Thermocouple 型式:	I-87018, 7018, 7018P
------------------	----------------------

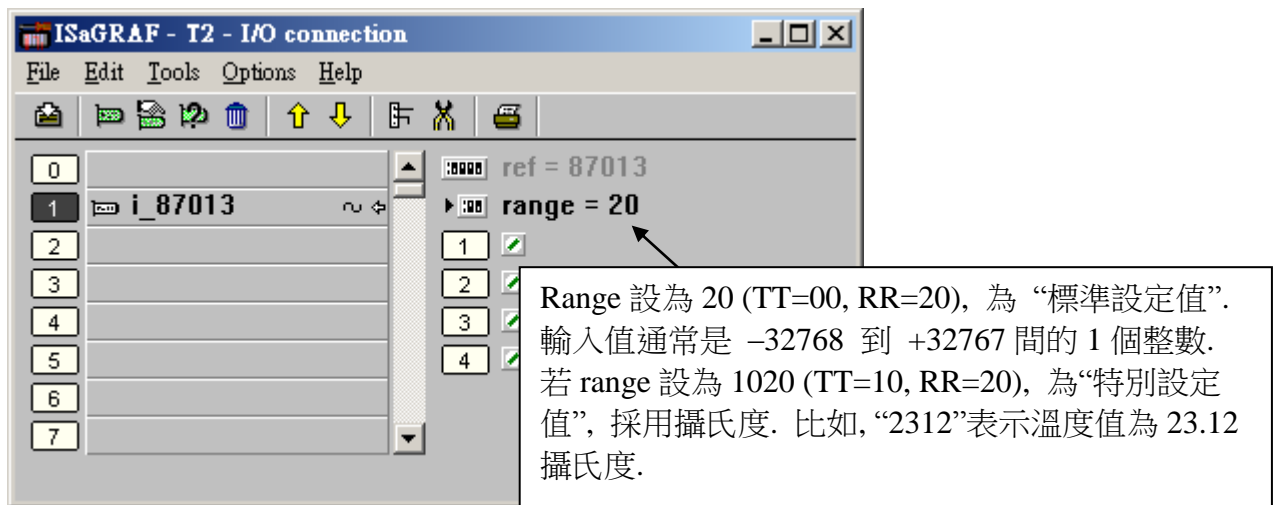
溫度 I-87K 板卡的“range”參數可以是“標準設定值”或“特別設定值”。

比如, I-87013W: 4 個 RTD 輸入. 它的“range”可以設成

20 : Platinum 100, a=0.00385, -100 ~ +100 攝氏度

...

2F : Platinum 100, a=0.003916, -200 ~ +200 攝氏度



若 I-87013 的“range”設為 20 (或 21 到 2F), 為“標準設定值”. 輸入值通常是 -32768 到 +32767 間的 1 個整數(跟 range 有關). 比如 range 為 20, 值 -32767 表示溫度約為 -100 度, +32766 約 +100 度. 值 16383 表示約 +50 度 (注意: 若卡片是有“感測器斷線偵測功能”的, 值 -32768 或+32767 表示“感測器斷線了”)

若想直接取得溫度卡量到的“攝氏度”或“華氏度”，將“range”設為“特別設定值”。格式如下：

重要：“特別設定值”從 驅動器版本 I-8xx7:3.11 , W-8xx7:3.24 起才有支持. iP-8xx7, WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6 目前也都有支持.

格式: TTRR (16 進位值)

TT = 10 (使用“特別設定值”，轉換為“攝氏度”)

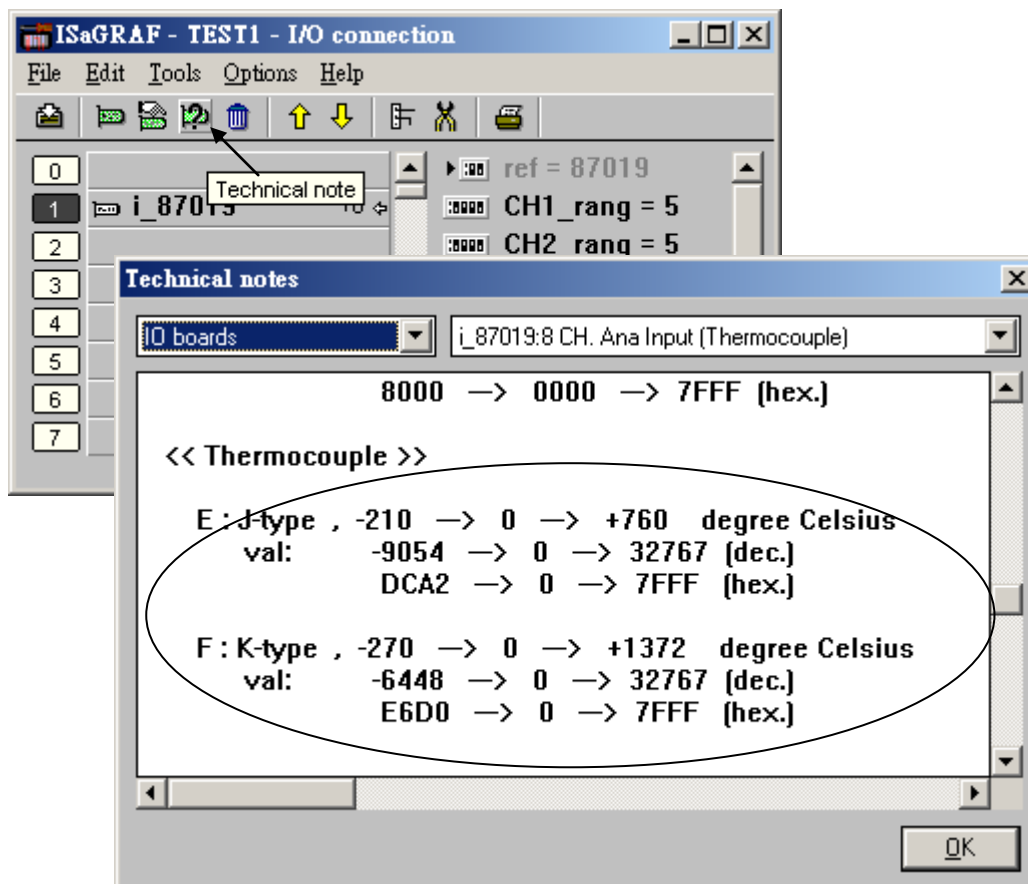
TT = 20 (使用“特別設定值”，轉換為“華氏度”)

TT = 00 (使用“標準設定值”，溫度輸入值通常為 -32768 to +32767 間的 1 個整數)

RR : 原始的 "range" 設定值 (跟卡片型號會有差別)

比如，設 I-87013 的“range”為以下：

- A. 1020 : (TT=10, RR=20) 表示使用“特別設定值”，轉換為“攝氏度”，單位為 0.01 度, range 為 "20 : Platinum 100, a=0.00385, degree Celsius". 所以若輸入值為 "2356" 表示 23.56 攝氏度, 若為"-489" 則是 -4.89 攝氏度, 若值為 "999990" 對於有“感測器斷線偵測功能”的卡片, 代表“感測器斷線了”。
- B. 202A : (TT=20, RR=2A), 表示使用“特別設定值”，轉換為“華氏度”，單位為 0.01 度, Range 為 "2A: Platinum 1000, a=0.00385, degree Celsius". 所以若輸入值為"4512" 表示 45.12 華氏度, 若是 "500" 則為 5.00 華氏度, 若值為 "999990" 對於有“感測器斷線偵測功能”的卡片, 代表“感測器斷線了”。
- C. 21 : (TT=00, RR=21), 表示使用“標準設定值”，值在 -32768 to +32767 之間, range 為 "21 : Platinum 100, a=0.00385, degree Celsius", 需參考附錄 D 或下圖 取得 On-Line 說明 來求得溫度值。



3.2.3: I-87017ZW 板卡使用說明

I-87017ZW 為工業用 電壓 或 電流量測 板卡, 可以使用板卡上的 Jumper 來調整為 10 個輸入點 (differential input), 或 20 個輸入點. 若有任一個 Channel 要量測電流必須設定為 10 個輸入點 (設為 20 個輸入點只能量測 電壓), 量電流的 輸入點 不需在 I/O 端子上並接 125 ohm 電阻, 只要調整板卡內的 相對的 jumper 就可使用.

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-87017ZW.

使用 ISaGRAF 來量測它時, 若設為 10 個輸入點, 它的 Sample Rate 約為 $10/10 = 1$ Hz. 即每秒對全部 10 接點可取樣 1 次 (設為 20 個輸入點的 Sample Rate 約為 $10/20 = 0.5$ Hz). (若您的應用是需要較快的 Sample Rate, 請採用 I-8017HW)

它的類比輸入值為 -32768 到 +32767, 與實際訊號的關係如下表.

電壓:

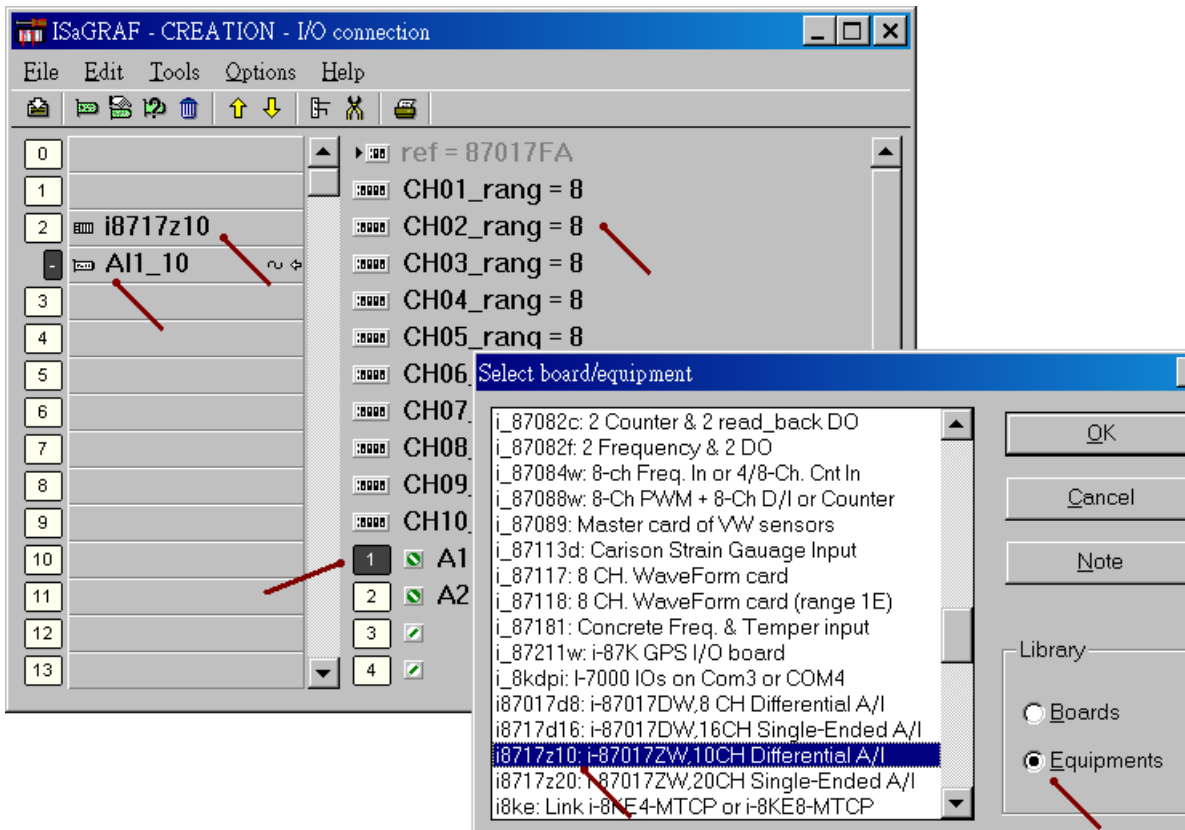
Range Type Code (Hex)	Data Format	Max value	Min value
08 (Default)	Input Range	+10.0 V	-10.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
09	Input Range	+5.0 V	-5.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0A	Input Range	+1.0 V	-1.0 V
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0B	Input Range	+500.0 mV	-500.0 mV
	Decimal Value	+32767	-32768
	2's Complement HEX	7FFF	8000
0C	Input Range	+150.0 mV	-150.0 mV
	Decimal Value	+32767	-32768

電流:

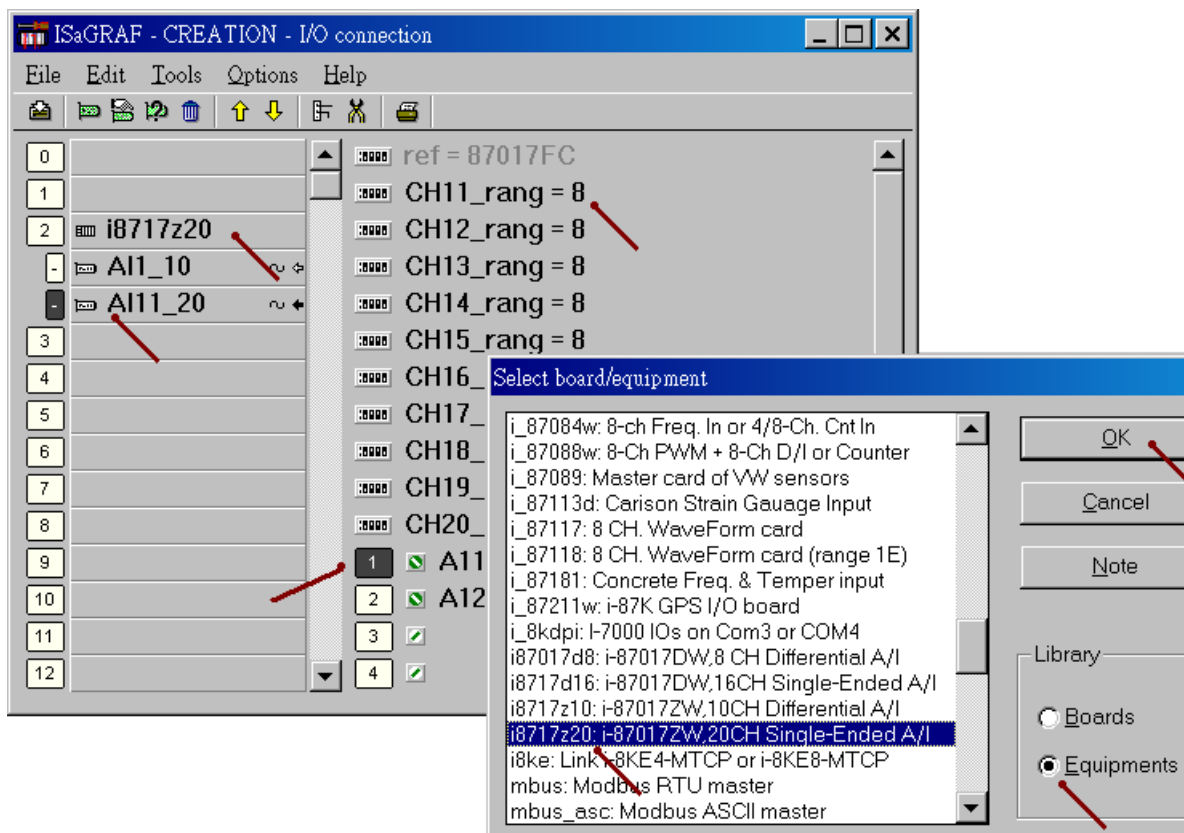
Range Code (16 進位)	項目	最大值	最小值
7	實際訊號	+20.0 mA	+4.0 mA
	類比輸入值 (10 進位)	+32767	0
	類比輸入值 (16 進位)	7FFF	0
D	實際訊號	+20.0 mA	-20.0 mA
	類比輸入值 (10 進位)	+32767	-32768
	類比輸入值 (16 進位)	7FFF	8000
1A	實際訊號	+20.0 mA	0 mA
	類比輸入值 (10 進位)	+32767	0
	類比輸入值 (16 進位)	7FFF	0

若是要使用設定為 10 個輸入點的 I-87017ZW，請在 IO connection 視窗內，在相對應的槽編號連上 “equipments” 內的 “i8717z10”，並設好每個 Channel 想使用的 Range 設定 (用 Mouse 左鍵雙擊 Ch01_rang 來設定)，之後再將對應的 Integer Input 變數 連接在 Ch.1 到 Ch.10 上 (用 Mouse 左鍵雙擊對應的 Channel 編號來設定)。

注意，若 I-87107ZW 的 jumper 是設定成 10-Ch. 可以量測電壓或電流，需注意 板卡上各 Channel 對應的 JP2 – JP11 需設定為正確的 Voltage input 或 Current Input 的方式



若是要使用設定為 20 個輸入點的 I-87017ZW，請在 IO connection 視窗內，在相對應的槽編號連上“equipments”內的“i8717z20”。(注意，若 I-87107ZW 的 jumper 是設定成 20-Ch. 只能用來量測電壓，所以 range 不能設成電流式的，即不能設成 07, 0D, 1A, 另外需注意 板卡上的 JP2 – JP11 也需設定為 Voltage input 的方式)



電流量測 4 ~ 20 mA 的重要訊息:

如果使用的 Sensor 為 4 到 20 mA 的電流量測型態，可以考慮將 range 設為 [D]: +/- 20 mA 或 [1A]: 0~ 20 mA 的 Range Type. (此時設 "[7]: 4 to 20 mA" 比較不好)

原因為:

如果 Range type 設為 [7]: 4 到 20 mA, Analog Input 值若為 0 或接近 0 可以表示 Sensor 輸入為 4 mA, 但也有可能是 Sensor 斷線. 因此光看 Analog Input 數值, 區分不出這 2 者.

然而若是將 Range 設為 [D]: +/- 20 mA 或 [1A]: 0 ~ 20 mA, 當值接近 0 時, 只有可能是 Sensor 斷線. 而 Sensor 輸入的 4 到 20mA, 值會是 (6553 到 32767), 4 mA 時值是在 6553 附近.

因此有需要分辨出電流 Sensor (4 到 mA) 是否正常時, 最好是規劃 range 為 [D]: +/- 20 mA 或 [1A]: 0 ~ 20 mA range type. 如此你的程式內就可比如說當 A1, A2 輸入值 小於 5000 或 小於 4000, 就可判定為 Sensor 斷線 或 Sensor 異常. (如果你是設成 [7]: 4 ~ 20 mA 的 Range Type, 就無法分辨, 到底是 4 mA 還是 Sensor 異常)

3.2.4: I-8017HW 板卡使用說明

I-8017HW 為 I-8K parallel 類比輸入卡, I-8017HW 可提供 8 個類比輸入點(differential input 方式) 或 16 個類比輸入點(single-end input 方式). 若用在 iP-8xx7 上, I-8017HW 的 Sample rate 最快約 125Hz (8-Ch.) 與 62 Hz (16-Ch.). 若用在 WP-8xx7, VP-25W7, XP-8xx7-CE6 上, I-8017HW 的 Sample rate 最快約約 200Hz (8-Ch.) 與 100 Hz (16-Ch.). 但要看 PLC 的 scan time 是多少, 若 scan time 較大, 比如 50 ms, 那結果會變成只有 $1000 / 50 = 20$ Hz.

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-8017HW

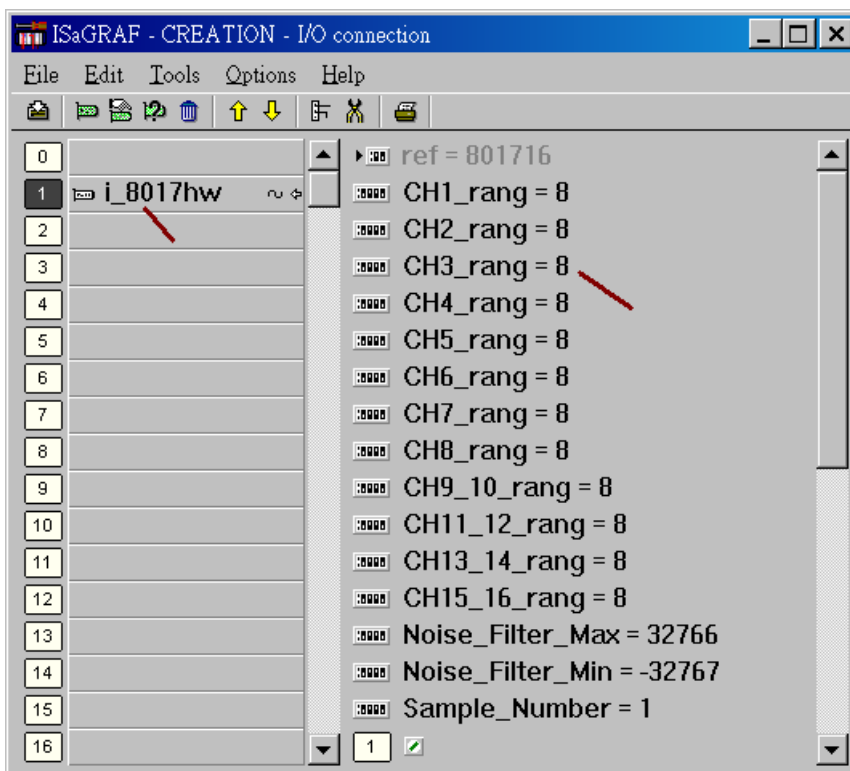
Range 設定	下限 實際值 (板卡輸入值)	實際值 (板卡輸入值)	上限 實際值 (板卡輸入值)
5	-2.5 V (-32768)	0 V (0)	+ 2.5 V (+32767)
6 (要外接 125 ohm 的 電阻於正/負端 並接)	-20 mA (-32768)	0 mA (0)	+ 20 mA (+32767)
7	-1.25 V (-32768)	0 V (0)	+ 1.25 V (+32767)
8	-10 V (-32768)	0 V (0)	+ 10 V (+32767)
9	-5 V (-32768)	0 V (0)	+ 5 V (+32767)

Noise_Filter_Max 參數: 為允許最大的上限值, 可為 -32468 ~ +32767, 板卡輸入值若超過此值, 會設成此 “Noise_Filter_Max” 之值

Noise_Filter_Min 參數: 為允許最小的下限值, 可為-32768 ~ +32467, 板卡輸入值若低於此值, 會設成此 “Noise_Filter_Min” 之值

Sample_Number 參數: 設定 多少次 I-8017HW 的 sample 計算成為 1 個平均值並 更新成為板卡的輸入值. 值可以是 1 到 500. 內定值為 1, 表示最快速的 Sample rate. 每 sample 一次就更新.

若設為 500, 表示最慢速的 Sample rate. 每 500 個 sample, 才會算出平均值更新 1 次. 此值設越大, I-8017HW 的 Sample rate 會降低, 輸入的波形會變的較平均.

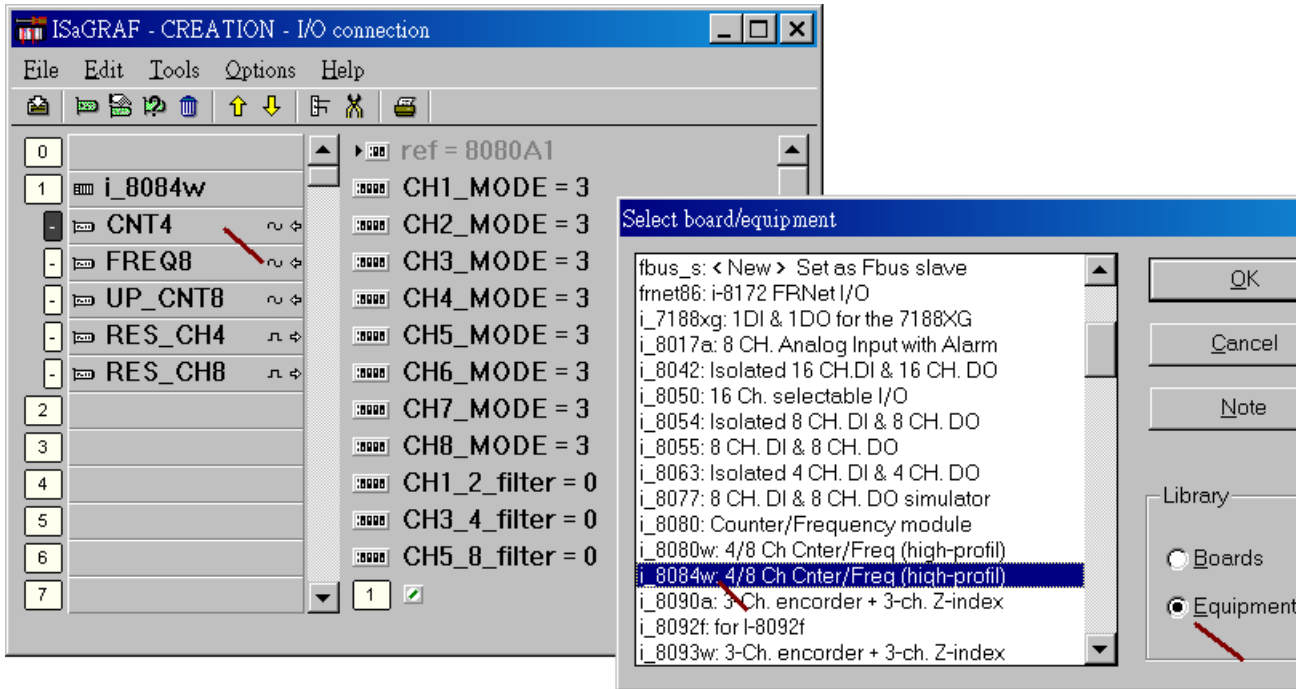


3.2.5: I-8084W 板卡使用說明

是 iP-8xx7, WP-8xx7, XP-8xx7-CE6 與 VP-25W7 / 23W7 有支持 I-8084W 板卡。

I-8084W 可用來當成 “8-Ch. 的 Up Counter” 或 “4-Ch. 的 Dir/Pulse Counter” 或 “4-Ch. 的 Up/Down Counter” 或 “4-Ch. A/B phase Counter (Quard. mode)”，也可用來當成 “8-Ch. 的 Frequency input”

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-8084W



內定的 "CHx_x_filter" 參數是 0 (0 表示不啟用慮波功能)。若值不是 0, 是用做 “慮波”使用, 寬度太小的波形會被過慮掉

CH1_2_filter: 為 “8-Ch. 的 Up Counter” 或 “8-Ch. Frequency” 的 Ch. 1 與 Ch 2 使用 或 “4-Ch. Dir/Pulse Counter” 或 “4-Ch. Up/Down Counter“ 或 “4-Ch. A/B phase” 的 Ch.1 使用

CH3_4_filter: 為 “8-Ch. 的 Up Counter” 或 “8-Ch. Frequency” 的 Ch. 3 與 Ch 4 使用 或 “4-Ch. Dir/Pulse Counter” 或 “4-Ch. Up/Down Counter“或 “4-Ch. A/B phase” 的 Ch.2 使用

CH5_8_filter: 為 “8-Ch. 的 Up Counter” 或 “8-Ch. Frequency” 的 Ch. 5, 6, 7, 8 使用 或 “4-Ch. Dir/Pulse Counter” 或 “4-Ch. Up/Down Counter“或 “4-Ch. A/B phase” 的 Ch.3, 4 使用

請依據你的實際的輸入訊號 允許的最大頻率來設這個值。

最大允許輸入的訊號頻率 (Hz)	CHx_x_filter 值
0 ~ 1K	200
0 ~ 2K	100
2K ~ 5K	40
5K ~ 10K	20
10K ~ 20K	10
20K ~ 100K	2
100K ~ 450K	1
0 ~ 450K	0 (不啟用慮波功能)

CHx_MODE 則是 用來設定 該 Channel 是要使用成為 那種類型的訊號輸入, 如下

CH1_MODE ~ CH8_MODE 是給 “8-Ch Up Counter” 與 “8-Ch Frequency” 的 Ch 1 到 Ch 8 使用

若是要使用 “4-Ch. DIR / Pulse Counter” 或 “4-Ch. Up / Down Counter” 或 “4-Ch. A/B phase Counter”

CH1_MODE 與 CH2_MODE : 設定值必須一樣, 表示 Ch1 的設定值

CH3_MODE 與 CH4_MODE : 設定值必須一樣, 表示 Ch2 的設定值

CH5_MODE 與 CH6_MODE : 設定值必須一樣, 表示 Ch3 的設定值

CH7_MODE 與 CH8_MODE : 設定值必須一樣, 表示 Ch4 的設定值

比如, CH1_MODE 設為 0 : “Dir / Pulse” (4-Ch), 就必需將 CH2_MODE 也設為 0.

CH1_MODE 設為 3 : “Up Count” (8-Ch), 那 CH2_MODE 可以設成 83 或 2, 6, A, 82, 86 或 8A.

以下的設定值是給 Counter 輸入時使用

0 : Dir / Pulse (4-Ch.)

1 : Up / Down (4-Ch.)

3 : Up Count (8-Ch)

4 : A/B phase (Quard. Mode, 4-Ch)

80 : Dir / Pulse (4-Ch. 輸入訊號為 反相)

81 : Up/Down (4-Ch. 輸入訊號為 反相)

83 : Up Count (8-Ch. 輸入訊號為 反相)

84 : A/B phase (Quard. Mode, 4-Ch, 輸入訊號為 反相)

以下的設定值是給 Frequency 輸入時使用

2 : Frequency (採用 “AutoTT” 設定)

6 : Frequency (採用 “LowTT” 設定)

A : Frequency (採用 “HighTT” 設定)

82 : Frequency (採用 “AutoTT” 設定, 輸入訊號為 反相)

86 : Frequency (採用 “LowTT” 設定, 輸入訊號為 反相)

8A : Frequency (採用 “HighTT” 設定, 輸入訊號為 反相)

注意:

1. DIR / Pulse 與 Up / Down Counter 與 A/B phase 模式類似於 Encoder Input.

此 Counter 值需控管在 -2,147,483,648 到 2,147,483,647 之間

2. Up Counter 模式的輸入值是一個 32-bit 整數, 此值從 0 開始, 一直依照實際的輸入訊號, ... 增加, 最大到 +2,147,483,647 然後 如再多輸入 1 個 Pulse, 值會突然降到 -2,147,483,648, 然後依照實際的輸入訊號再...增加, 到 -1, 0, ...再到 +2,147,483,647.

ISaGRAF 的整數變數為 (32-bit signed, 有正負號), 因此無法取得 大於+2,147,483,647 的正值.

如果使用者搭配的 圖控軟體或 HMI 可以接受 (32-bit unsigned, 無負號) 的整數, 就可以在 該 圖控軟體或 HMI 上得到 0, 1, ..., +2147483647, +2147483648, +2147483649, ..., +4294967295 再回到 0, 1, ...的 無負號的 Counter 值

3.2.6: I-87015W 與 I-87015PW 板卡使用說明

如果 RTD sensor 放的位置距離 I-87015W 板卡比較遠, 請改成購買 I-87015PW 較好.

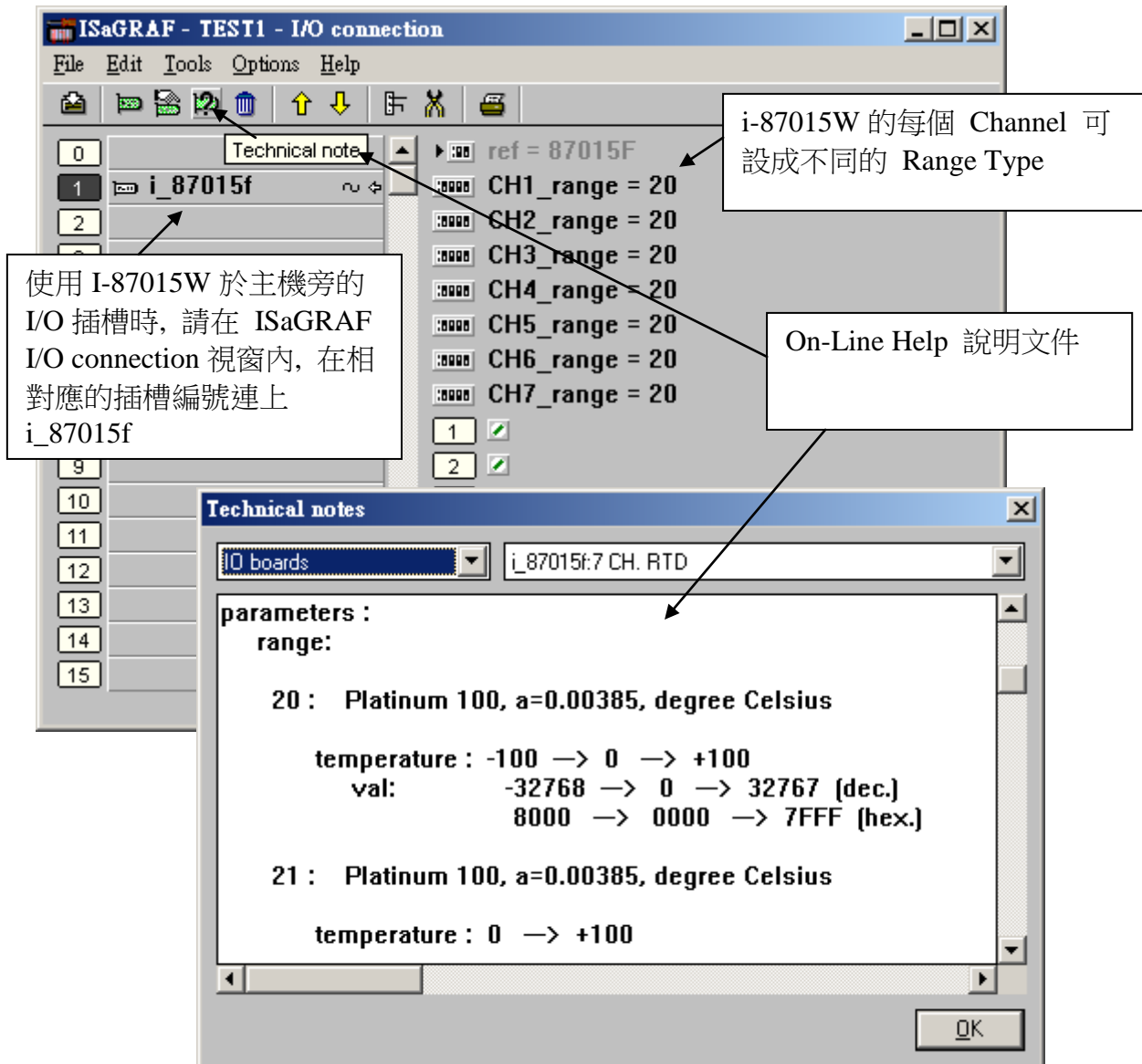
i-87K I/O 板卡 (如 I-87015W, I-87015PW) 可插在 PAC 的第 0 ~ 7 槽. 也可當成 RS-485 遠程 I/O 模塊來使用 (此時需插在 I-87K4/5/8/9 或 RU-87P4/8 擴充底板上, 使用說明請參考第 6 章).

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-87015W

它的 Sample Rate 約為 $12/7 = 1.7$ Hz. 即每秒對全部 7 接點可取樣約 1.7 次

下圖為 把 I-87015W / I-87015PW 插在主機的第 1 槽的使用例.

I-87015W / 87015PW 為 7 點 (路) RTD 溫度輸入卡, 使用 I-87015W 於主機旁的 I/O 插槽時, 請在 ISaGRAF I/O connection 視窗內, 在相對應的插槽編號連上 “i_87015f”, 每個輸入點可設為不同的 Range, 請參閱 On-Line Help 內的說明, 如下. (若要取得 攝式度 或 華式度 請參閱 3.2.2 節來設 Range 參數)



3.2.7: I-87019ZW 板卡使用說明

I-87019ZW 為工業用 溫度 Thermo-Couple 量測 板卡，或可為一般類比輸入量測板卡，如“±10V”或“±20mA”，每個板卡上有 10 個輸入點，每個接點可設成量測不同的 訊號 與 Range.

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-87019ZW 它的類比輸入值一般為 -32768 到 +32767，與實際訊號的關係請參閱附錄 D. 當使用成溫度輸入時，可以使用如第 3.2.2 節的方式來直接取得溫度值.

I-87019ZW 若 Range 是設成 06, 07, 0D 或 1A 來量測電流時 不需在 I/O 端子上並接 125 ohm 電阻，只需調整板卡上各接點的 Jumper 為電流輸入方式就可使用.

它的 Sample Rate 約為 10/10 = 1 Hz. 即每秒對全部 10 接點可取樣 1 次. (若您的應用是如“±10V”或“±20mA”的訊號量測,且需要較快的 Sample Rate, 請採用 I-8017HW)

請在 IO connection 視窗內，在相對應的槽編號連上“i_87019z”，並設好想使用的 Range 設定，之後再將對應的 Integer Input 變數 連接在 Ch.1 到 Ch.10 上，如下

若設為 100F, 表示使用 T/C K-Type, 轉換成 攝氏度, 單位為 0.01 度.

若設為 200F, 表示使用 T/C K-Type, 轉換成 華氏度, 單位為 0.01 度.

當使用特殊設定 回傳的溫度值為 999990 時, 表示 溫度輸入 Sensor 斷掉了 (參考第 3.2.2 節)

3.2.8: I-8024W 板卡使用說明

I-8024W 為工業用類比輸出板卡，可輸出 “±10V” 或 “0 到 20mA” 的訊號。每個 I-8024W 上面有 4 個輸出點，每點可設為使用不同的 Range 輸出 (電壓 或 電流)。

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-8024W

它的輸出 速率跟 PLC 的 Scan Time 有關，如果 PLC 的 scan time 約是 10 ms，那若程式的輸出值有變，那它會在約 10 ms 更新輸出值。 但若 PLC scan time 是 100ms，那若程式的輸出值有變，那它會在約 100 ms 更新輸出值。

跟 I-87024W 相比，I-8024W 的更新速率較快，且 I-87024W 不像 I-8024W 那樣每個接點都可設成不同 Range，I-87024W 的 4 個輸出點都需設成相同的一個 Range。

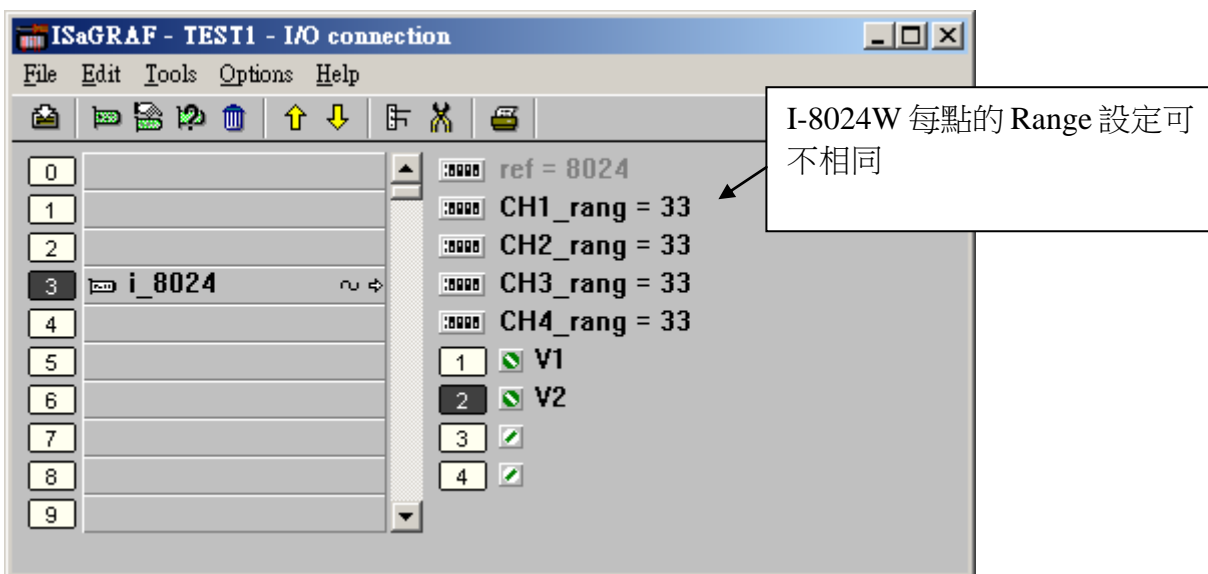
不過 I-87024W 可在主機旁的第 0 到 7 槽使用，也可在 RS-485 遠程擴充模組 I-87K4/5/8/9 或 RU-87P4/8 上使用。而 I-8024W 只能在主機旁的第 0 到 7 槽使用。

I-8024W 的輸出值為：

電流：(0 到 32767) 代表 (0 到 20 mA) 或 電壓：(-32768 到 32767) 代表 (-10 到 10 V)

Range Code	項目	最大值	最小值
30	實際電流輸出訊號	+20.0 mA	+0.0 mA
	類比輸出值	+32767	+0
33	實際電壓輸出訊號	+10.0 V	-10.0 V
	類比輸出值	+32767	-32768

要使用 I-8024W，請在 IO connection 視窗內相對應的槽編號上，連上 “i_8024”，之後請設定每個接點的 Range，再把 Integer Output 變數連在各別的 Channel 上。



3.2.9: I-87018ZW 板卡使用說明

I-8701ZW 為工業用 溫度 Thermo-Couple 量測 板卡, 或可為一般類比輸入量測板卡, 如 $\pm 20\text{mA}$ 或 $0 \sim 20 \text{ mA}$ 或 $4 \sim 20 \text{ mA}$ 或 $\pm 2.5\text{V}$, $\pm 1\text{V}$, $\pm 500\text{mV}$, $\pm 100\text{mV}$, $\pm 50\text{mV}$, $\pm 15\text{mV}$. 每個板卡上有 10 個輸入點, 每個 Channel 的量測準確度較 i-87018W 與 i-87018RW 高, 且每個 Channel 可設成使用不同的輸入型態, 比如可使用第 1 到 4 Ch. 來量測 $4 \sim 20\text{mA}$, 第 5 到 8 Ch. 來量測 T/C K-Type, 第 9 Ch 量測 $\pm 2.5\text{V}$, 第 10 Ch. 量測 T/C R-Type

請參考 http://www.icpdas.com/products/PAC/i-8000/8000_IO_modules.htm > I-87018ZW

它的類比輸入值一般為 -32768 到 $+32767$, 與實際訊號的關係請參閱附錄 D. 當使用成溫度輸入時, 可以使用如第 3.2.2 節的方式來直接取得溫度值.

I-87018ZW 若 Range 是設成量測電流, 例如 06: $\pm 20\text{mA}$ 或 07: $4 \sim 20 \text{ mA}$ 或 1A: $0 \sim 20 \text{ mA}$, 必需在 I/O 端子上該 Channel 的正, 負端, 並接 1 個 125 ohm 電阻, 才能量測電流值.

它的 Sample Rate 約為 $10/10 = 1 \text{ Hz}$. 即每秒對全部 10 接點可取樣 1 次 (若您的應用是如 “ $\pm 10\text{V}$ ” 或 “ $\pm 20\text{mA}$ ” 的訊號量測, 且需要較快的 Sample Rate, 請採用 I-8017HW)

請在 IO connection 視窗內, 在相對應的槽編號連上 “i_87018z”, 並設好想使用的 Range 設定, 之後再將對應的 Integer Input 變數 連接在 Ch.1 到 Ch.10 上, 如下

點選 “?” 可查詢關於 i_87018z 的 range 設定說明.

87018ZW 的每個接點可設成不同的 Range.

本例, 第 1 到 4 Ch. 設為 1A: $0 \sim 20 \text{ mA}$.

第 5 到 8 Ch. 設為 100F, 表示使用 T/C K-Type, 轉換成 攝氏度, 單位為 0.01 度.

第 9 到 10 Ch. 設為 200F, 表示使用 T/C K-Type, 轉換成 華氏度, 單位為 0.01 度.

當使用特輸設定 回傳的溫度值為 999990 時, 表示 溫度輸入 Sensor 斷線了. (參考第 3.2.2 節)

3.3: 一些特殊虛卡的連結使用說明

3.3.1: 連結 Push4Key 與 Show3Led 虛卡

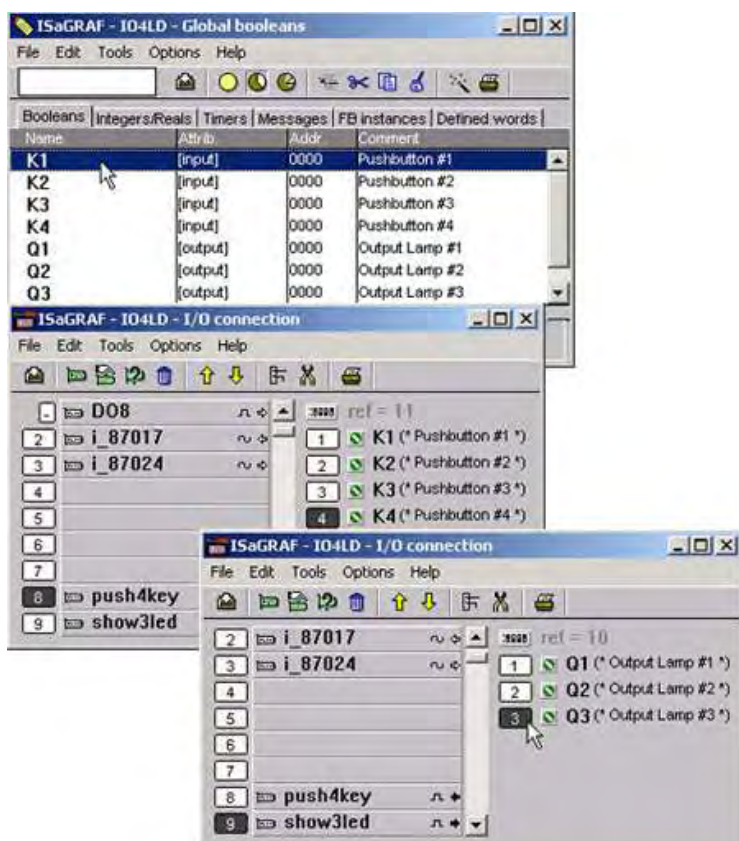
I-8xx7 與 iP-8xx7 控制器有一些可以幫忙程式進行測式和除錯的工具。這些工具即在 I-8xx7 控制器前方面板上的 "Push4Key" 和 "Show3Led" (其他 PAC, 如 XP-8xx7-CE6, WP-8xx7, VP-25W7 / 23W7, 7188EG/XG, μ PAC-7186EG 不支持 pushkey 與 show3Led)。

"Push4Key" 主要有四個可以處理數位輸入的按鍵，分別位於控制器的前方面板。而"Show3Led"是在控制器前方面板上的左邊三個 LED 燈(從左到右的三個 LED 燈，而第四個 LED 燈為控制器的電源燈)其主要為處理數位輸出訊號。

"Push4Key" 和 "Show3Led" 透過 "I/O Connect" 視窗可以連結到 ISaGRAF 程式，用來幫助程式除錯。若您要在"I/O Connection"上使用"Push4Key" 和 "Show3Led"這兩個虛擬 I/O 板，建議您把他們設在插槽編號 8 或 8 以上的插槽上(因為插槽 0~7 是給真實的 I/O 板用的)。

注意:

在 ISaGRAF 程式中使用"Push4Button"和"Show3Led"時，您必需在"ISaGRAF Global Variables"宣告變數名稱給每一個"Push4Key"的輸入變數和"Show3Led"的輸出變數。



3.3.2: 連結 io_state 虛卡 來偵測 真實 I/O 板卡的運作狀態

“io_state” 虛卡可用來偵測 是否 對應的第 0 到 7 槽的 真實的 I/O 板卡 是否有以下情況發生

- (1) 插錯卡片 或 插錯槽編號 (無法偵測如 I-8112iW/8114iW/8142iW/8144iW 與 I-8212W 與 I-8072/I-8073 等卡片)
- (2) 卡片被拔掉
- (3) 卡片損壞導至 板卡 ID 讀取錯誤, 所以可判讀為 卡片故障.

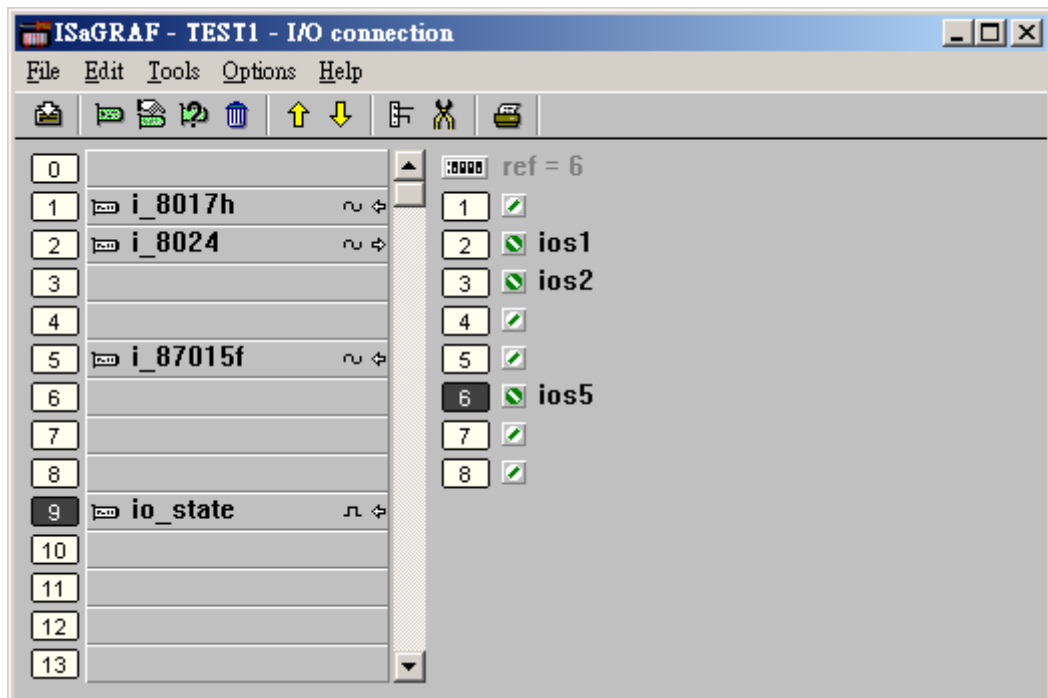
重要:

請不要 I-8xxx 與 I-8xxxW 等 Parallel 卡片在控制器 有上電 的情況下 將 I/O 板卡插入 第 0 到 7 的 IO 插槽. 此動作有很大的可能會造成 IO 板卡 甚至是控制器的損壞. 只有 I-87xxxW 等高板卡片可以在 iP-8xx7, XP-8xx7-CE6, WP-8xx7 與 VP-25W7, VP-23W7 與 VP-2117 等 PAC 內進行熱插拔.

“io_state” 只可偵測 有連在 ISaGRAF I/O connection 視窗內 第 0 到 7 槽的 真實板卡, 沒作 IO connection 但確有插在第 0 到 7 槽的 真實板卡 不會去 偵測.

使用方式如下, 將 io_state 連結在第 8 或以上的槽編號上.

此例會偵測 第 1, 2 與第 5 槽的對應的真實卡片是否正常運作. “io_state” 的 Ch.1 到 Ch.8 為對應於第 0 到 7 槽 真實 IO 卡片的運作狀態, 傳回 TRUE 表示運作正常, 傳回 FALSE 表示異常或故障. (XP-8xx7-CE6 與 Wincon 沒有 Ch.1 → 第 0 槽)



3.3.3: 連結 echo_tim 虛卡 來延遲 Modbus RTU Slave 串口一段時間後再回覆

I-8xx7, iP-8xx7 與 W-8xx7/8xx6 控制器 有支持 “echo_tim” 虛卡，可用來 延遲 Modbus RTU Slave 通訊串口一段時間後再回覆。 每台 I-8xx7 與 W-8xx7/8xx6 控制器 只有 1 個 Modbus RTU Slave 通訊串口可設為 延遲回覆，其它的 Modbus RTU Slave 通訊串口都是 收到 PC / HMI / Other 的 Modbus Request 後，會馬上回覆 ()。

可設為延遲回覆的 Modbus RTU Slave 通訊串口：

I-8xx7, iP-8xx7: COM1 或 COM2 或 COM3 的其中一個

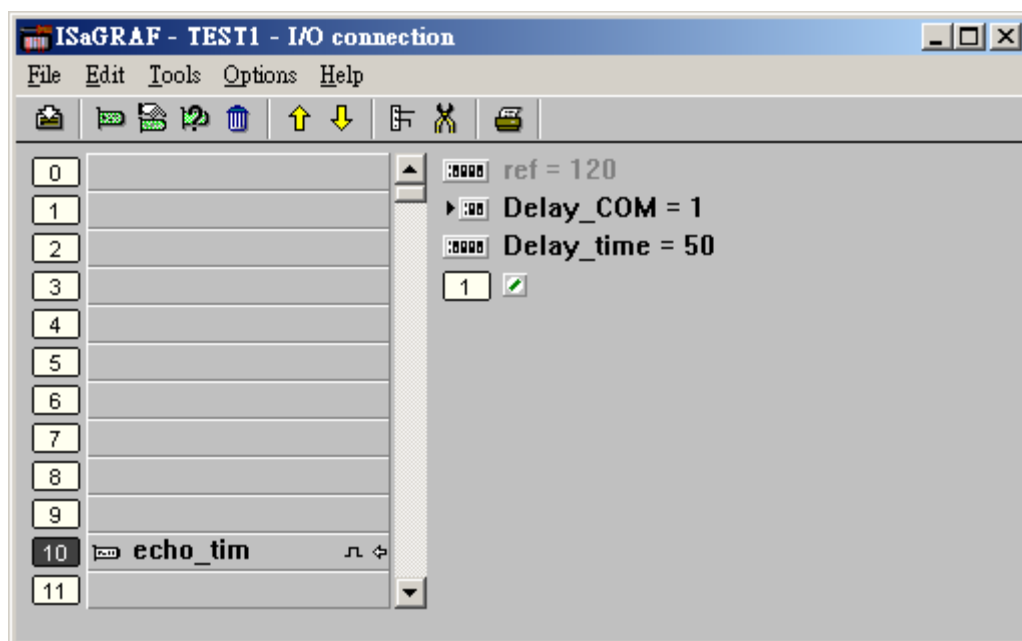
W-8xx7 / 8xx6: COM2 或 COM3 的其中一個

使用方法為將 echo_tim 連結在槽編號 8 或以上的槽編號上。

第 1 個 Delay_COM 參數為 要延遲那個 通訊串口, I-8xx7, iP-8xx7 可以是 1, 2 或 3. W-8xx7 可以是 2 或 3. 第 2 個 Delay_time 參數為 “要延遲多久?“, 可以是 1 到 10000, 單位為 ms (0.001 秒)

Ch.1 若回傳 FALSE 表示 Ok, 若回傳 TRUE 表示 有問題, 比如用錯 串口編號。

以下的例子, 會將 I-8xx7 的 COM1: Modbus RTU Slave 串口的回覆時間延遲 0.05 秒後再回覆。(Wincon 只能延遲 COM2 或 COM3)



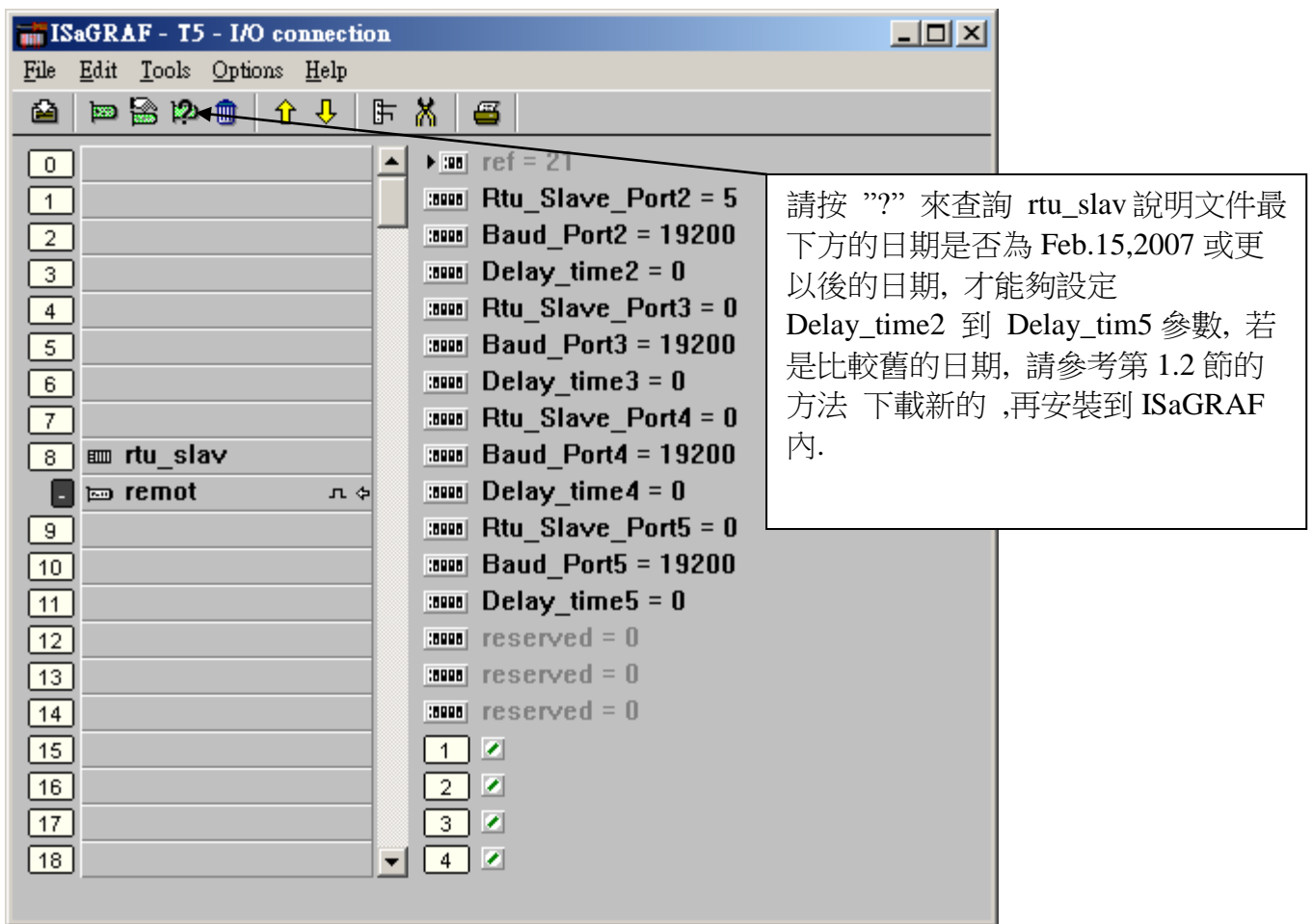
為何需要延遲 Modbus RTU Slave 通訊串口的回覆時間？

有些應用的通訊設備需要切換 收 或 送 的狀態，而切換狀態時會需要一小段時間，比如像某些 無線傳輸設備。當主控端 發出 Modbus RTU 詢問命令 經由 “此無線傳輸設備” 發到 I-8xx7, iP-8xx7 或 W-8xx7 的 Modbus RTU Slave 串口 之後, I-8xx7 或 W-8xx7 不能馬上回覆，需等一段時間等 “此無線傳輸設備” 收/送 狀態切換完畢後才可回覆。

3.3.4: WP-8xx7, VP-25W7 與 XP-8xx7 連結 RTU_Slav 虛卡來使用更多擴充的 Modbus RTU Slave 串口

WP-8xx7, VP-25W7 除了 COM2:RS-232 與 COM3:RS-485 的其中一個可以規劃成為 Modbus RTU Slave 串口外, 另外可再規劃最多 4 個串口來支持 Modbus RTU Slave 串口 於擴充的 i-8112iW/8114iW/8142iW/8144iW 板卡上 (COM5 to COM8). XP-8xx7-CE6 則是可再規劃最多 8 個串口來支持 Modbus RTU Slave 串口. 請先參考 個別 PAC 的“快速上手手冊” 先將擴充的 COM port 先設立起來.

在 ISaGRAF 程式內要規劃 擴充的 Modbus RTU Slave 串口, 需在 I/O connection 視窗內連上 “RTU_Slav” 虛卡 於 Slot 編號 8 或 8 以上 (XP-8xx7-CE6 還有另外支持 RTU_Slv2), 如下



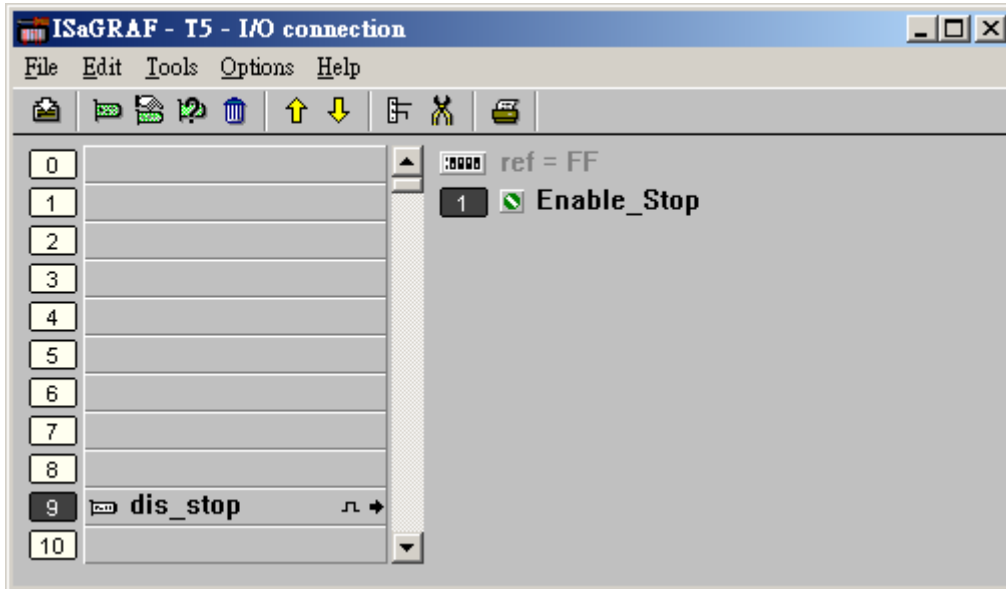
Rtu_Slave_Port2 到 Rtu_Slave_Port5 參數 可以設成 0, 5, 6, 7 或 8. 設為 0 表示該 Port 不想規劃為 Modbus RTU Slave 串口.

Baud_Port2 到 Baud_Port5 參數可以設成 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 或 115200.

Delay_time2 到 Baud_Port5 參數可以設成 0 或 1 到 10000, 單位為 0.001 秒. 表示收到 PC/HMI/圖控軟體 的詢問命令後, 要延遲多少時間 再回覆. 關於延遲回覆的需求說明 請參考第 3.3.3 節.

3.3.5: 使用 dis_stop 虛卡 來 關閉 / 開啟 ISaGRAF Download 程式功能

有時當 ISaGRAF PAC 程式開發完, 為了預防 被人誤用 或故意 使用 ISaGRAF 從 Internet 來將 Program Stop, 可以連接 Dis_stop 虛卡, 將該輸出點的 初值 設為 TRUE, 就可防止被人使用 ISaGRAF 將程式 Stop.



若是需要再啟用它, 請先用原來的 ISaGRAF 程式 連上 controller 後, 將 該輸出點 設為 False. 如此就可 download 修改後的 ISaGRAF 程式進去.

關於更多網路安全的說明, 請參考第 19 章.

3.4: 直接表示變數

若您是使用 ISaGRAF-256 或 ISaGRAF-L workbench，您就不需使用本章節所提到的技巧。

ISaGRAF Workbench 程式有一個很有用的特色，就是能夠使用”直接表示變數”。直接表示變數不須實際連結到輸入或輸出點。有四種版本的 ISaGRAF Workbench 程式支援 ISaGRAF PAC 控制程式。如:ISaGRAF-32、ISaGRAF-80、ISaGRAF-256 和 ISaGRAF-L。ISaGRAF 後面的數字代表在所指定的 ISaGRAF Workbench 程式中可允許使用的輸出/輸入的變數數目。

ISaGRAF Workbench 程式有一個硬體保護裝置直接裝在電腦的平行埠上，每當你在 ISaGRAF 內編譯程式時，硬體保護裝置就會被讀取以判斷你是否有連結更多的輸出/輸入變數名稱以致超過您所購買的 ISaGRAF 所允許的輸出/輸入變數數目。

這些”直接表示變數”可以被使用在真正的輸出,輸入當中，所以你可以建立額外的變數而不超過 ISaGRAF 所允許的輸出,輸入變數數量。“直接表示變數”必需小心使用，存取這些“直接表示變數”比使用輸出/輸入變數複雜。**對一個專業的程式設計師而言，建議購買 ISaGRAF-256。**

單一型態板(Board)的直接表示變數程式設計:

單一型態板: "s" 表示插槽號碼, "c" 表示 channel 號碼

%IXs.c	布林輸入板, 如	%IX2.3
%QXs.c	布林輸出板, 如	%QX0.2
%IDs.c	整數輸入板, 如	%ID3.1
%QDs.c	整數輸出板, 如	%QD2.4
%ISs.c	訊息輸入板, 如	%IS3.1
%QSs.c	訊息輸出板, 如	%QS2.4

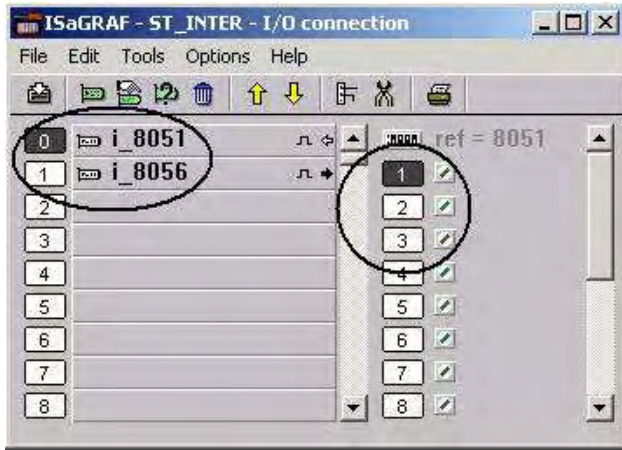
複合型態板(Equipment)的直接表示變數程式設計

複合型態板: "s"表示插槽號碼, "b" 代表複合板內各個子板的編號 "c"表示 channel 號碼.

%IXs.b.c	布林輸入板,	如	%IX2.3.2
%QXs.b.c	布林輸出板,	如	%QX0.2.1
%IDs.b.c	整數輸入板,	如	%ID3.1.3
%QDs.b.c	整數輸出板,	如	%QD2.4.3
%ISs.b.c	訊息輸入板,	如	%IS3.3.1
%QSs.b.c	訊息輸出板,	如	%QS2.1.4

一個直接變數的程式例子

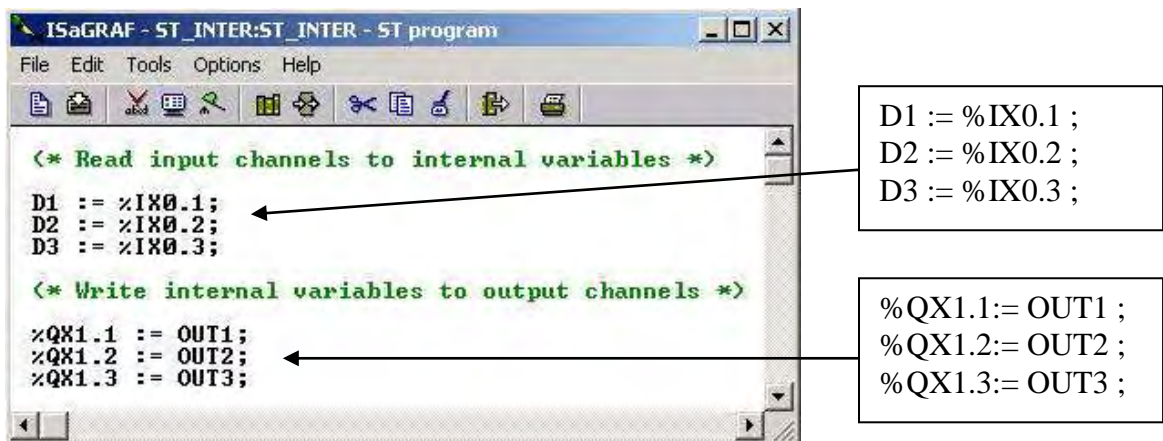
在 ISaGRAF ST 程式中建立一個新的專案，接著在視窗下連結到 I/O boards。在插槽 0 上連結一個 I-8051 板並為它宣告三個內部屬性的變數(D1、D2、D3)，在插槽 1 上連結一個 I-8056 板並為它宣告三個內部屬性的變數(OUT1、OUT2、OUT3)。切記為各個變數設定他們的屬性為”內部”以代替”輸出”和”輸入”(這表示這些變數不需實際連到實體的 I/O)



建立一個新的”ST”程式



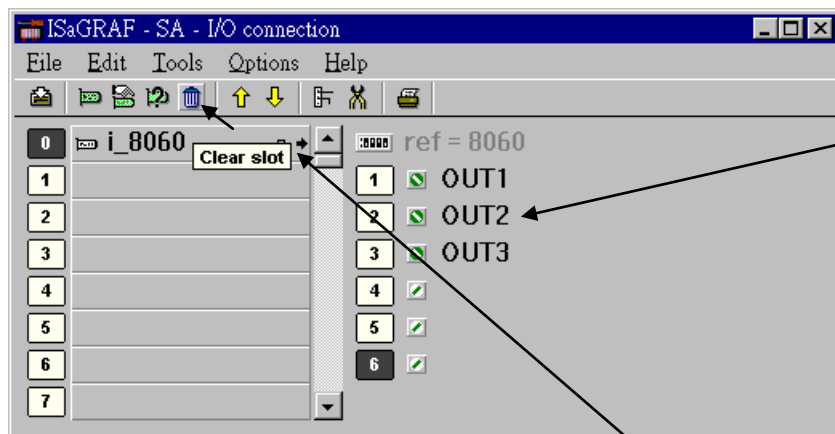
雙擊”ST_inter” 接著 ”ISaGRAF ST Program” 程式視窗將會開啟，接著依照下方視窗上的程式碼輸入，記得每行的結尾一定要輸入分號(“;”)。



現在我們可以在別的程式中使用所建立的內部變數 D1~D3 和 OUT1~OUT3. 這些內部變數將會在 ST 程式中的相關 I/O 板裡產生輸出輸入的動作。

注意:

假如輸入或輸出變數有連結到 “Boards”或 “Equipment”、而現在想要更改變數內容以使用”直接表示變數”時，這些輸出輸入變數必需重新設定成”內部”屬性且”Board”或 “Equipment”必需再重新連結一下到插槽上。



假如您想使變數的屬性改為”內部”屬性，您必須在 “dictionary”視窗內重新更改變數的屬性。

將插槽刪除再重新連結.

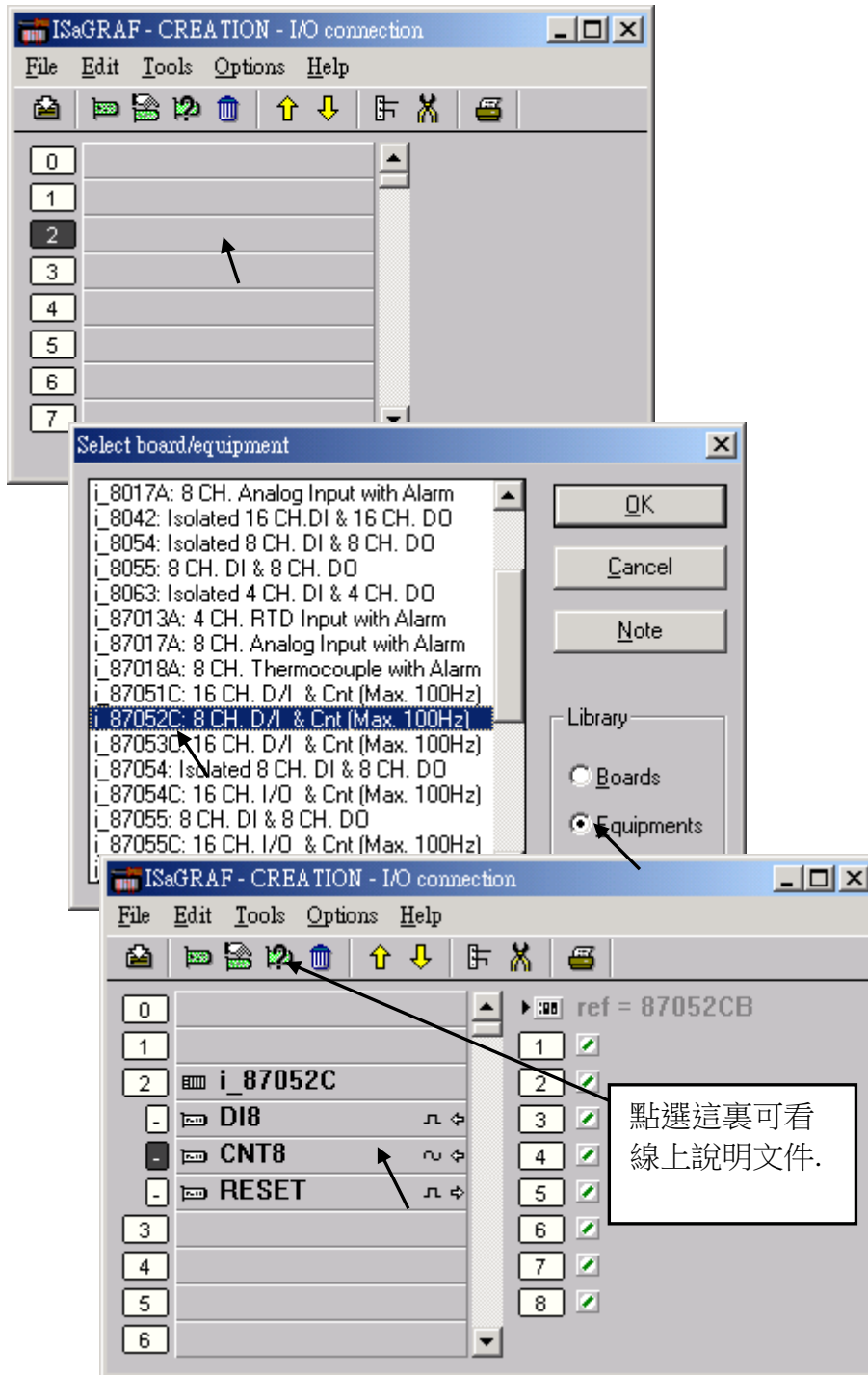
注意:

若要使用上載功能且又想同時使用 “直接表示變數”，在編譯選項中的 “Upload”裡有一個 “Command for not connect I/O channel” 必需要選取。(參考 9.2 節).

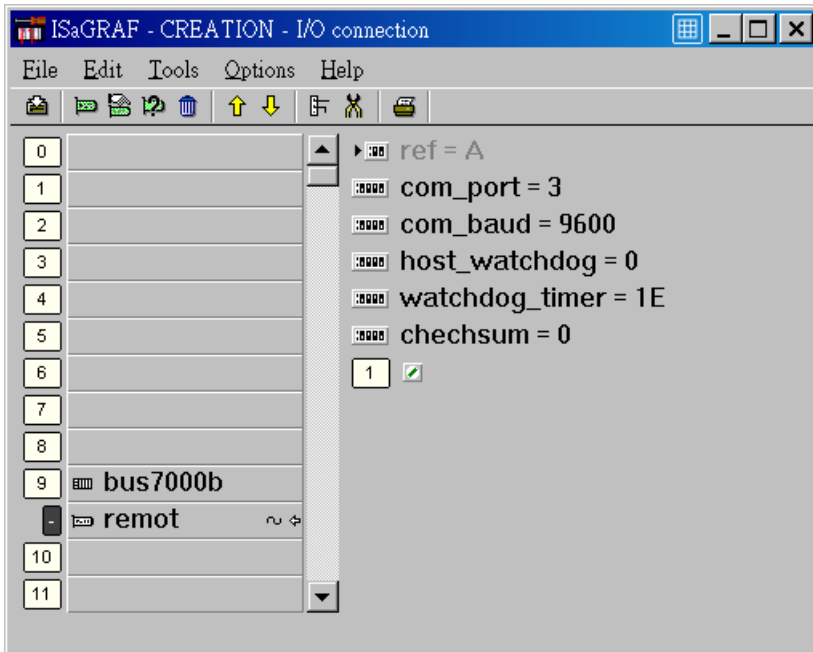
3.5: 內建於 I-87xxx 及 I-7000 D/I 模組 的計數值

87051W, 87052W, 87053W, 87054W, 87055W, 87058W, 87063W & i-7050, 7052, 7053, 7041, 7044, 7060, 7063, 7065 等模組的每個 D/I 點有內建的計數值(Counter). 這些 D/I 計數值只能接受最快到 100 Hz. 計數值的範圍為 0 到 65535, 可重置為 0.

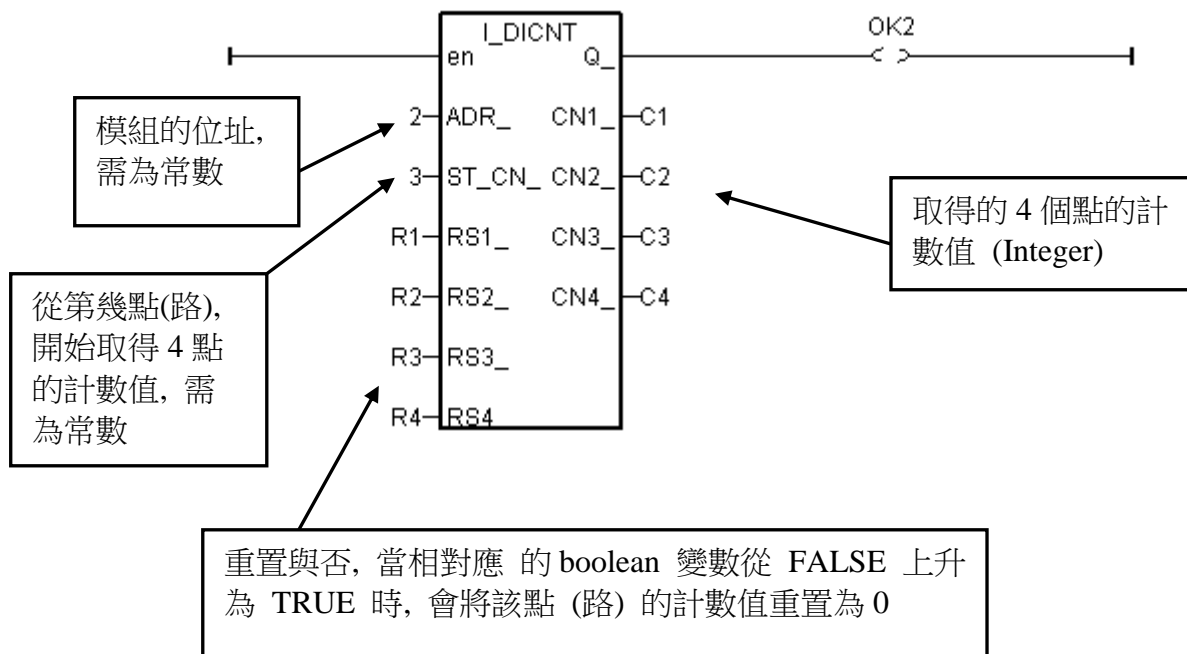
若是 將 I-87xxx D/I 模組插在主機的第 0 ~ 7 槽, 只要在”I/O connection”視窗內連接以 “C” 結尾的 I-87xxx D/I 模組即可. 例如 “i_87052C”.



若是 將 I-87xxx D/I 模組插在 87K4, 87K5, 87K8 & 87K9 等遠程擴展模組上, 或是使用 I-7000 D/I 模組, 首先請依照第 6 章的方法, 以 DCON Utility 設好每個模組的位址編號, 接著在”I/O connection” 內連接 “Bus7000b”.



然後可在 LD 程式內使用 “I_DiCnt” 方塊來取得 D/I 計數值(Counter). 每個 “I_DiCnt” 方塊可取得 4 個 計數值.



3.6: 自動搜尋及宣告 : Auto-Scan I/O

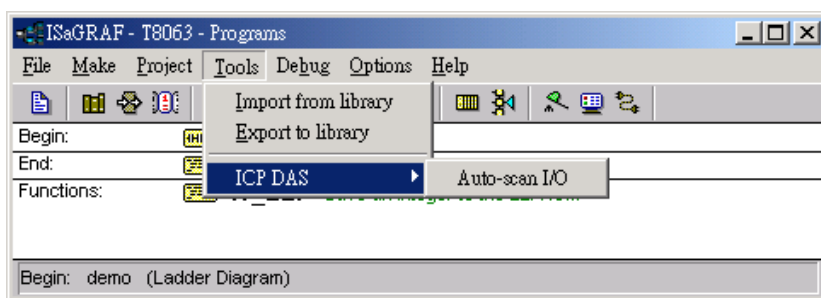
在可以正確使用 Auto-scan I/O 功能前, 請參閱第 1.2 節確認 “ICP DAS Utilities For ISaGRAF” 是否已經安裝好 (注意: 並非所有的卡片都支持 Auto-scan).

什麼是 Auto-scan I/O :

這是 ICP DAS 為了 ISaGRAF 控制器而設計在 ISaGRAF 內, 可以自動搜尋插在第 0~7 槽的 I/O 模組, 及自動宣告 I/O 變數的功能.

如何使用 ?

- A. 開啟你的 ISaGRAF 專案 (Project).
- B. 以滑鼠點選 “Tools - ICP DAS - Auto-scan I/O” 以開始自動搜尋.

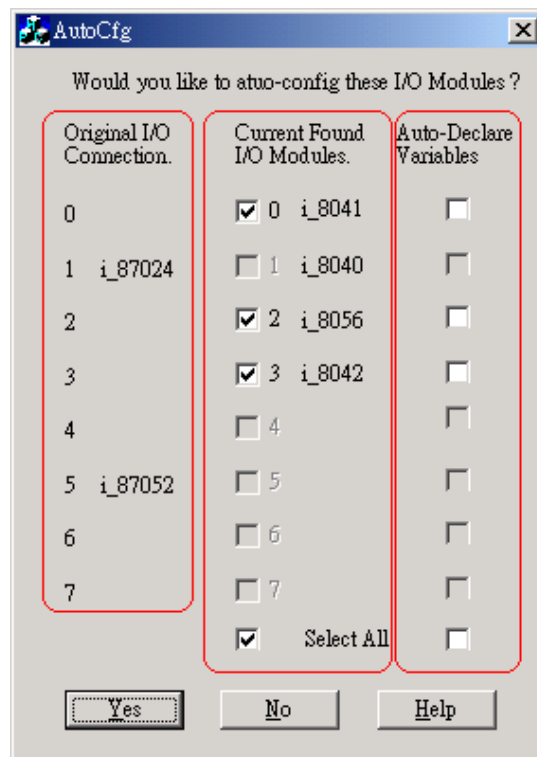


- C. Auto-scan I/O 區分為 3 個區域

Original I/O Connection : 左側, 顯示你的專案(Project) 原本連結在 0~7 槽的 I/O 模組.

Current Found I/O Modules : 中間, 顯示目前有搜尋到已插在第 0~7 槽的 I/O 模組.

Auto-Declare Variables : 右側, 顯示有那些搜尋到的 I/O 模組, 可以自動宣告 I/O 變數.



D. 在 “Current Found I/O Modules” 區：

只有原先專案沒有連結的槽位(Slot) 可以選取.

E. 在 “Auto-Declare Variables” 區：

只有在“Current Found I/O Modules” 區內有選取的槽位(Slot) 才可以選取.

F. 你可以選取下方”Select ALL” 去選取全部可選取的項目.

Auto-scan I/O 的必要項目？

A. 通訊的連接設定(Link setup) 要對 (Com1, Com2, Ethernet ,...)

B. I/O 模組要先插好在 PAC 上.

自動宣告變數的命名規則:

變數名稱格式：型態_槽編號_接點(路) 編號

型態:

DI 點 : DI

DO 點 : DO

AI 點 : AI

AO 點 : AO

槽編號：1 個位數的號碼, 0 ~ 7.

接點(路) 編號：1 個位數的號碼, 01 ~ 32.

範例.:

DI_0_02, 在第 0 槽第 2 號接點的 DI.

AI_5_06, 在第 5 槽第 6 號接點的 AI.

DO_2_12, 在第 2 槽第 12 號接點的 DO.

AO_1_03, 在第 1 槽第 3 號接點的 AO.

3.7: PWM 輸出

ISaGRAF 控制器的週期時間(scan time) 與使用者寫的 ISaGRAF 程式及控制器的驅動程式(Driver)有關. 一般的應用約介於 5 到 40 ms 之間. 當使用者的 ISaGRAF 程式寫的很複雜時有可能會增加到 100 ms. 因此要用 ISaGRAF 寫程式來產生固定週期精確的脈波輸出並不容易, 比如, 脈波輸出 2 ms OFF & 1 ms ON.

注意: 若想使用高速的 PWM, 請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-105 來使用 I-8088W, I-87088W 或 I-7088W.

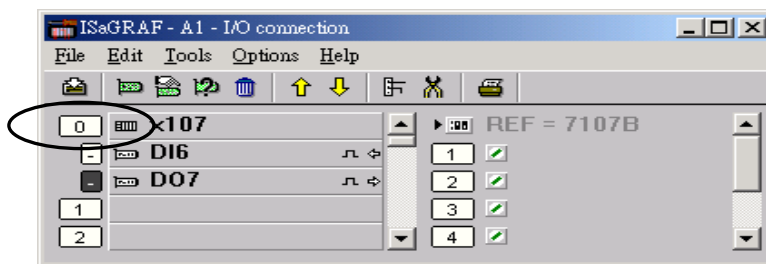
只有並列輸出(parallel output) 卡有支援 PWM 功能, 如下.

I-8037W, 8041W, 8042W, 8054W, 8055W, 8056W, 8057W, 8060W, 8063W, 8064W, 8065W, 8066W, 8068W, 8069W (output 若是 Relay 不適合使用 PWM, 因為 relay 可能會來不及反應)

而對 I-7188EG/XG, μ PAC-7186EG 而言, 只要是含有輸出點的 X-xxx 卡就可支援 PWM 功能.

注意: 最多只可同時開啟 8 個 PWM 輸出.

I-7188EG/XG, μ PAC-7186EG 必需先在 slot 0 連結 X-xxx 卡後, PWM 輸出才能使用.

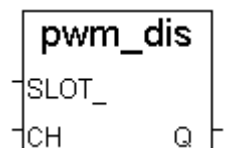


以下函式可用來 處理 PWM 輸出的相關功能.

PWM_dis 關閉 PWM 輸出

輸入參數:

SLOT_	整數	那個 slot ?	0 ~ 7
CH_	整數	第幾點 ?	1 ~ 32.



傳回值:

Q_	布林	TRUE: 正確 .
		FALSE: 錯誤的輸入參數, 或太多點被開啟(>8), 或找不到.

注意:

1. 當 PWM 輸出以 PWM_dis 關閉後, 該相對應的輸出點改被 ISaGRAF cycle engine 控制.
2. 最多 8 個輸出點可使用 PWM_en, PWM_en2, pwm_ON, pwm_OFF.

範例: I-8xx7, iP-8xx7: demo_63 , W-8xx7: Wdemo_22

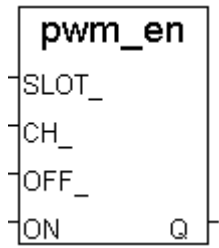
PWM_en 開啟 PWM 輸出功能直到使用 PWM_dis 來關閉為止

輸入參數:

SLOT_ 整數 那個 slot ? 0 ~ 7
 CH_ 整數 第幾點 ? 1 ~ 32.
 OFF_ 整數 Off 時間, 單位為 ms.
 ON_ 整數 On 時間, 單位為 ms.

I-8xx7, I-7188EG/XG, 7186EG, iP-8xx7: 1 ~ 32767

WP-8xx7, XP-8xx7-CE6, VP-25W7 : 2 ~ 32766 (←須設為 2 的倍數)



傳回值:

Q_ 布林 TRUE: 正確 .

FALSE: 錯誤的輸入參數, 或太多點被開啟(>8), 或找不到.

範例: I-8xx7, iP-8xx7: demo_63 , W-8xx7: Wdemo_22

PWM_en2 開啟 PWM 輸出功能來輸出一給定數量的脈波

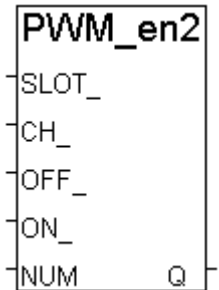
輸入參數:

SLOT_ 整數 那個 slot ? 0 ~ 7
 CH_ 整數 第幾點 ? 1 ~ 32.
 OFF_ 整數 Off 時間, 單位為 ms.
 ON_ 整數 On 時間, 單位為 ms.

I-8xx7, I-7188EG/XG, 7186EG, iP-8xx7: 1 ~ 32767

WP-8xx7, XP-8xx7-CE6, VP-25W7 : 2 ~ 32766 (←須設為 2 的倍數)

NUM_ 整數 要輸出幾個脈波, 1 - 2,147,483,647, 若給負值, 將永遠不會停, 除非用 Pwm_dis 命令

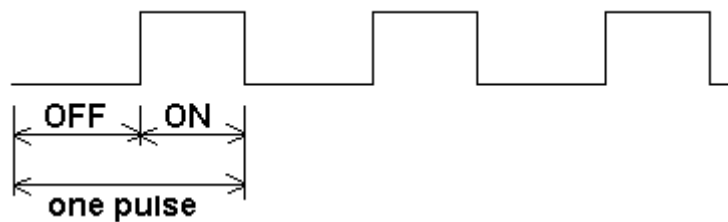


傳回值:

Q_ 布林 TRUE: 正確 .

FALSE: 錯誤的輸入參數, 或太多點被開啟(>8), 或找不到.

PWM 輸出波形:



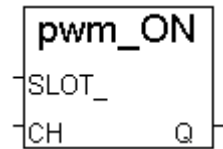
注意:

1. 使用 PWM_en 或 PWM_en2 後, 該輸出點會以 OFF, ON, OFF, ON, ... 的順序來輸出.
2. 當使用 PWM_en2 輸出達到所要求的脈波數量時, 該點的 PWM 輸出會停止並自動關閉(所以不需再使用 PWM_dis 來關閉它).
3. 可使用 PWM_sts 來測試 PWM_en2 是否到達所要求的脈波數量.
4. 最多 8 個輸出點可使用 PWM_en, PWM_en2, pwm_ON, pwm_OFF.
5. 不可再開啟 已經 開啟 PWM 輸出的 D/O 點.

pwm_ON 將 並列 D/O (Parallel D/O) 立刻輸出為 TRUE

輸入參數:

SLOT_	整數	那個 slot ? 0 ~ 7
CH_	整數	第幾點 ? 1 ~ 32.



傳回值:

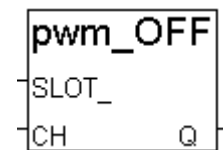
Q_	布林	TRUE: 正確 . FALSE: 錯誤的輸入參數, 或太多點被開啟(>8), 或找不到.
----	----	--

範例: I-8xx7, iP-8xx7: demo_63 , W-8xx7: Wdemo_22

pwm_OFF 將 並列 D/O (Parallel D/O) 立刻輸出為 FALSE

輸入參數:

SLOT_	整數	那個 slot ? I-8xx7: 0 ~ 7 , I-7188EG/XG: 0 W-8xx7: 1 ~ 7
CH_	整數	第幾點 ? 1 ~ 32.



傳回值:

Q_	布林	TRUE: 正確 . FALSE: 錯誤的輸入參數, 或太多點被開啟(>8), 或找不到.
----	----	--

範例: I-8xx7: demo_63 , W-8xx7: Wdemo_22

注意:

1. 最多 8 個輸出點可使用 PWM_en, PWM_en2, pwm_ON, pwm_OFF.
2. pwm_ON 會馬上將該 D/O 點設為 TRUE.
3. pwm_OFF 會馬上將該 D/O 點設為 FALSE .
4. 假如某 D/O 點已使用過 pwm_ON 或 pwm_OFF, 請先以 PWM_dis 關閉該點後, 才能改使用 PWM_en 或 PWM_en2.

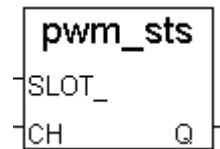
PWM_sts 測試 PWM 的狀態

輸入參數:

SLOT_	整數	那個 slot ? 0 ~ 7
CH_	整數	第幾點 ? 1 ~ 32.

傳回值:

Q_	布林	TRUE: 該輸出點 PWM 還在動作中. FALSE: 該輸出點 PWM 已關閉 (如果是先前是使用 pwm_en2 來開啟 PWM, FALSE 表示已到達所要求的脈波數量).
----	----	---



注意:

1. 最多 8 個輸出點可使用 PWM_en, PWM_en2, pwm_ON, pwm_OFF.
2. 此函式可用來測試"PWM_en2" 是否到達所要求的脈波數量.

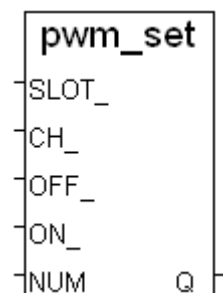
PWM_set 動態改變 PWM 的 ON / OFF 波形寬度

輸入參數:

SLOT_	整數	那個 slot ? 0 ~ 7
CH_	整數	第幾點 ? 1 ~ 32.
OFF_	整數	Off 時間, 單位為 ms.
ON_	整數	On 時間, 單位為 ms.
		I-8xx7, I-7188EG/XG, 7186EG, iP-8xx7: 1 ~ 32767
		WP-8xx7, XP-8xx7-CE6, VP-25W7 : 2 ~ 32766 (←須設為 2 的倍數)
NUM_	整數	要輸出幾個脈波, 1 - 2,147,483,647, 若給負值, 將永遠不會停, 除非用 Pwm_dis 命令

回傳值:

Q_	布林	TRUE: 正確 . FALSE: 發生錯誤.
----	----	-------------------------



PWM_sts2 取得 pwm_en2 與 pwm_en 啟用後已經輸出的 脈波數

輸入參數:

SLOT_	整數	那個 slot ? 0 ~ 7
CH_	整數	第幾點 ? 1 ~ 32.

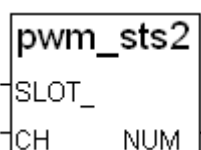
回傳值:

NUM_	整數	pwm_en2 與 pwm_en 啟用後已經輸出的 脈波數
------	----	-------------------------------

注意:

1. 此函數只有 在 pwm_en2 或 pwm_en 對該點 有啟用才有作用.
2. 回傳的脈波數有可能會比實際的輸出脈波數小一些. 比如啟用"pwm_en2" 輸出 20000 個脈波數, 有可能輸出完時只回傳數量為 19998.
3. 若 "pwm_en2" 給的 'NUM_' 參數為負值, 脈波的輸出將不會停止, 除非碰到 PWM_DIS 命令它停. 這種若一直不停的脈波輸出值會從 0, 1, 2, ... 到 2,147,483,647 然後會回到 0, 1, 2, ...

範例: I-8xx7, iP-8xx7: demo_63 , W-8xx7: Wdemo_22



3.8: 並列輸入卡的 Counter

只有插在 slot 0 的 並列輸入卡 有 counter 功能 (XP-8xx7-CE6 則是 Slot 0). 如下.
I-8040W, 8042W, 8051W, 8052W, 8053W, 8054W, 8055W, 8058W, 8063W

I-7188EG / XG 與 μ PAC-7186EG 只有含有 D/I 點的 X-xxx 卡有 D/I counter.

注意: 若想使用高速 Counter , Frequency, 請參考 http://www.icpdas.com/faq/isagraf_c.htm
> FAQ-100 來使用 I-8084W 或 I-87084W.

每台控制器最多只能使用 8 個並列 D/I counter. Counter 的輸入頻率最高為 500Hz. 最小脈波長度需大於 1 ms.

以下的 c 功能方塊可 取得/重置(reset) slot 0 的 D/I counter.

輸入參數:

RS1_ ~ RS8_ 布林 當由 FALSE 上升到 TRUE 時重置該點的 counter

傳回值:

Q_ 布林 正確: TRUE. 如果 Q_ 傳回 FALSE, 表示“slot 0 找不到並列輸入點 ”

CN1_ ~ CN8_ 整數 第 1 到 8 點的 counter 值. 範圍介於 0 到 2,147,483,647. 假如值超過 2,147,483,647, 會從 0 開始.

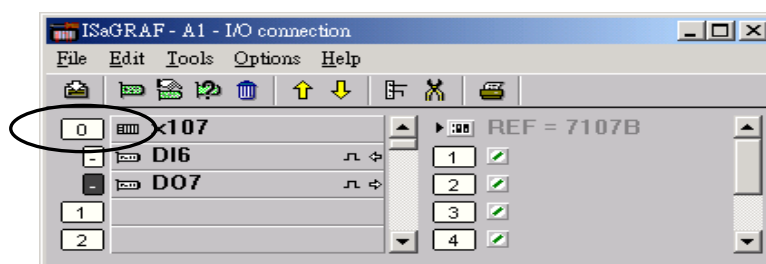
Di_Cnt	
	Q_
RS1_	CN1_
RS2_	CN2_
RS3_	CN3_
RS4_	CN4_
RS5_	CN5_
RS6_	CN6_
RS7_	CN7_
RS8_	CN8_

注意:

只有插在 slot 0 的並列輸入(parallel D/I)卡可使用“Di_Cnt”, 其它 slot 不行.

只有前 8 個 D/I 點可使用“Di_Cnt”.

I-7188EG/XG, μ PAC-7186EG 必須在 slot 0 連結 Xxxx 卡, “Di_Cnt” 才能使用.



範例: 請參閱 I-8xx7, iP-8xx7 的 demo_52 & demo_53.

3.10: 步進輸出

I-8417/8817/8437/8837 (其它 PAC 不支持步進輸出) 從驅動程式 2.37 版起支援 D/O 卡來控制步進輸出. 只有以下的板卡提供此功能.

I-8037, 8041, 8042, 8054, 8055, 8056, 8057, 8060, 8063, 8064, 8065, 8066, 8068, 8069

每台控制器最多支援 2 軸.

每軸由 4 點 D/O 來驅動. 請用以下方式接線.

第 1 軸: A --- Ch.1 B --- Ch.2 A_ --- Ch.3 B_ --- Ch.4

第 2 軸: A --- Ch.5 B --- Ch.6 A_ --- Ch.7 B_ --- Ch.8

注意:

請勿在同一個 D/O 點同時啟動 PWM 及 步進輸出 功能.

I-7188EG/XG , 7186EG, W-8xx7, WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 不支援 步進輸出 功能.

可用的函式方塊:

STP_en 啟動 步進輸出

輸入參數:

SLOT_ 整數 那個 slot ? 0 ~ 7

AXIS_ 整數 第幾軸 ? 1 - 2

第 1 軸: (Ch.1 - Ch.4), 第 2 軸: (Ch.5 - Ch.8)

MODE_ 整數 那種 mode ? 1 - 3, (A, B, A_, B_) =

Mode 1: (1, 0, 0, 0) --> (0, 1, 0, 0) --> (0, 0, 1, 0) --> (0, 0, 0, 1)

Mode 2: (1, 1, 0, 0) --> (0, 1, 1, 0) --> (0, 0, 1, 1) --> (1, 0, 0, 1)

Mode 3: (1, 0, 0, 0) --> (1, 1, 0, 0) --> (0, 1, 0, 0) --> (0, 1, 1, 0) -->

(0, 0, 1, 0) --> (0, 0, 1, 1) --> (0, 0, 0, 1) --> (1, 0, 0, 1)

MS_ 整數 步距, 1 - 1000, 單位為 ms.

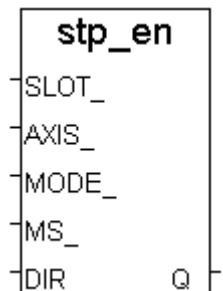
比如. 設為 5 表示每秒跑 200 步.

DIR_ 布林 True: 正轉, False: 反轉

傳回值:

Q_ 布林 TRUE: 正確 .

FALSE: 錯誤的輸入參數, 或找不到.



注意:

1. 可使用 "STP_dis" 去停止 步進輸出

2. 假如找不到 "STP_en", "STP_en2", "STP_sts" 及 "STP_dis", 請到以下網站下載 "ICP DAS Utilities For ISaGRAF.zip", 然後執行 setup.exe 來安裝

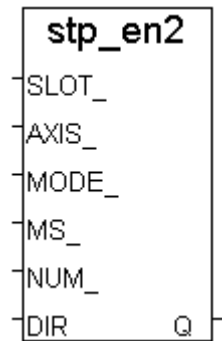
<http://www.icpdas.com/products/8000/isagraf.htm>

範例: demo_58 & demo_59.

STP_en2 啟動 步進輸出 去輸出給定的 步數

輸入參數:

SLOT_	整數	那個 slot ? 0 ~ 7
AXIS_	整數	第幾軸 ? 1 - 2 第 1 軸: (Ch.1 - Ch.4), 第 2 軸: (Ch.5 - Ch.8)
MODE_	整數	那種 mode ? 1 - 3, (A, B, A_, B_) = Mode 1: (1, 0, 0, 0) --> (0, 1, 0, 0) --> (0, 0, 1, 0) --> (0, 0, 0, 1) Mode 2: (1, 1, 0, 0) --> (0, 1, 1, 0) --> (0, 0, 1, 1) --> (1, 0, 0, 1) Mode 3: (1, 0, 0, 0) --> (1, 1, 0, 0) --> (0, 1, 0, 0) --> (0, 1, 1, 0) --> (0, 0, 1, 0) --> (0, 0, 1, 1) --> (0, 0, 0, 1) --> (1, 0, 0, 1)
MS_	整數	步距, 1 - 1000, 單位為 ms. 比如. 設為 5 表示每秒跑 200 步.
NUM_	整數	要輸出幾步? 0 - 2,147,483,647
DIR_	布林	True: 正轉, False: 反轉



傳回值:

Q_	布林	TRUE: 正確 . FALSE: 錯誤的輸入參數, 或找不到.
----	----	-------------------------------------

注意:

可使用 "STP_sts" 去測試 "STP_en2" 是否已到達給定的 步數.

2. 停止 "STP_en2" 的方法為 - 使用 "STP_dis" 或 - 等 它到達所要求步數

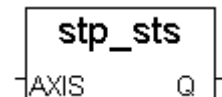
STP_sts 取得步進輸出狀態

輸入參數:

AXIS_	整數	第幾軸? 1 - 2 第 1 軸: (Ch.1 - Ch.4), 第 2 軸: (Ch.5 - Ch.8)
-------	----	--

傳回值:

Q_	布林	TRUE: 仍在動作中, FALSE: 停止 (對 stp_en2 而言, 表示已到達給定的 步數)
----	----	--



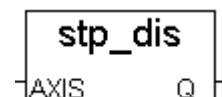
STP_dis 停止步進輸出

輸入參數:

AXIS_	整數	第幾軸? 1 - 2 第 1 軸: (Ch.1 - Ch.4), 第 2 軸: (Ch.5 - Ch.8)
-------	----	--

傳回值:

Q_	布林	TRUE: 正確 . FALSE: 錯誤的輸入參數, 或找不到.
----	----	-------------------------------------



範例: demo_58 & demo_59.

第 4 章 連接 I-8xx7 控制器與人機介面(HMI)程式

這個章節詳細的描述如何使 ISaGRAF 控制器與 SCADA 或人機介面(HMI)程式交換資料。這是一個很有用的特色允許使用者建立他們自己的人機介面程式來與 I-8xx7, I-7188EG/XG, μ PAC-7186EG, iP-8xx7, VP-2117, WP-8xx7, XP-8xx7-CE6 與 VP-25W7, VP-23W7 控制系統連結。

在您詳細了解 4.1 節後，假如您想使用 ISaGRAF 控制器當做 ”Modbus RTU 或 Modbus TCP/IP I/O”，您可以參考 4.3 節。ICP DAS 所提供的 ”Touch series” 觸控螢幕支援 ”Modbus” 通訊協定，且 ”Touch series” 可以連結 ISaGRAF 控制器以存取資料。在 4.4 章節裡詳細的介紹如何連結 ”Touch 500” 和 ISaGRAF 控制系統。

注意:

1. 請參閱各別 PAC 的 “快速上手手冊” 來得知 如何啟用 Modbus RTU slave port.
2. 所有 有 Ethernet 網口的 ISaGRAF 控制器都有支持 Modbus TCP/IP Slave 規約.
3. I-8417/8817 與 iP-8x47 的 COM1:RS232 與 COM2:RS485 內定支持 Modbus RTU Slave 規約.
4. I-8437-80/8837-80 的 COM1:RS232 內定支持 Modbus RTU Slave 規約. 4. I-7188EG/XG, μ PAC-7186EG 的 COM1 內定支持 Modbus RTU Slave 規約
5. WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7 出貨時並未啟用 Modbus RTU Slave 串口. 但是他們的 Ethernet port 都已經啟用了 Modbus TCP/IP Slave.

4.1: 宣告變數的網路位址

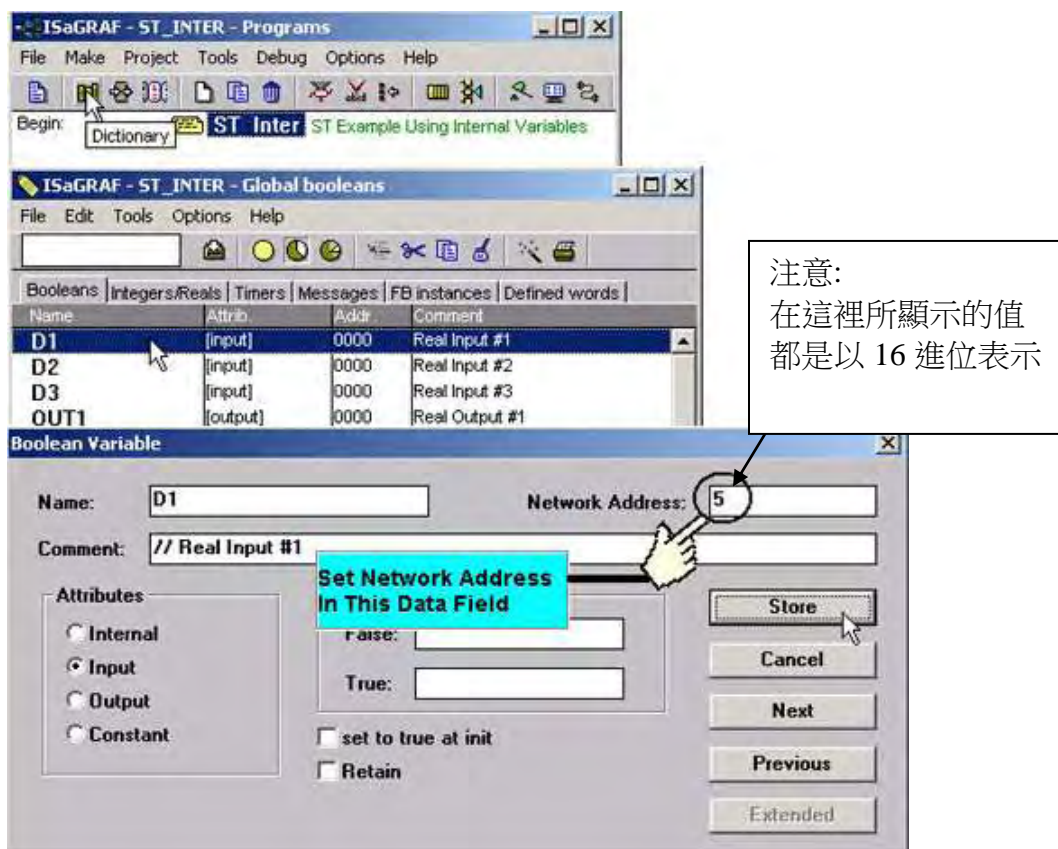
為了讓 ISaGRAF 控制器和 SCADA 或 其他軟體程式(HMI) 或 人機介面程式中能交換處理資料，您必需先為每一個變數宣告其網路位址號碼,其他的軟體程式或人機介面程式要 讀/寫控制器的資料必需透過這個網路位址號碼。

注意:

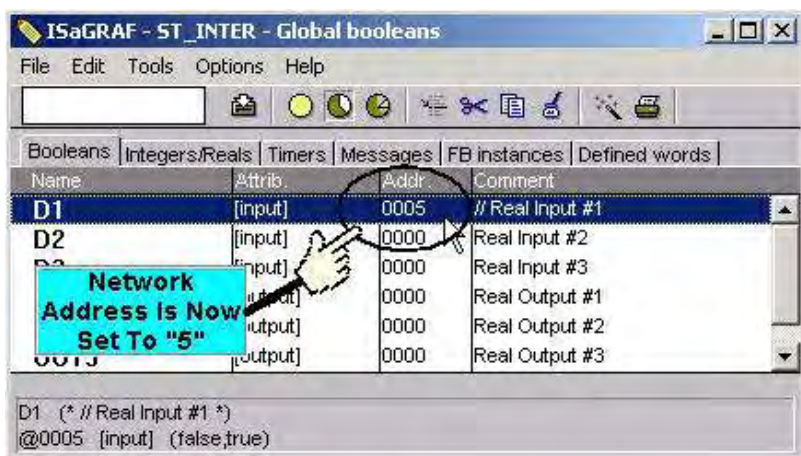
1. **I-8xx7, I-8437-80, I-8837-80 , I-7188EG/XG, VP-2117, μ PAC-7186EG 與 iP-8xx7** 可使用的變數網路位址號碼為 1 到 FFF(若以十進位表示為 1 ~ 4095). 位址號碼 **5001 到 8072** 是給 Word 和 Integer 陣列使用的(請參閱 4.5 節).
2. WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7 可使用的變數網路位址號碼為 1 到 1FFF(若以十進位表示為 1 ~ 8191). 位址號碼 **10,001 到 19,216** 是給 Word 和 Integer 陣列使用的(請參閱 4.5 節).

要宣告變數的網路位址有兩種方法是可行的，第一種即為下面圖示的宣告方式。(若要設定 網路位址 給 “變數陣列”，請參考第 2.6 節的說明).

開啟 ”ISaGRAF Programs” 視窗並且單擊一下 ”Dictionary” 圖示，接著雙擊變數名稱來指定網路位址號碼。

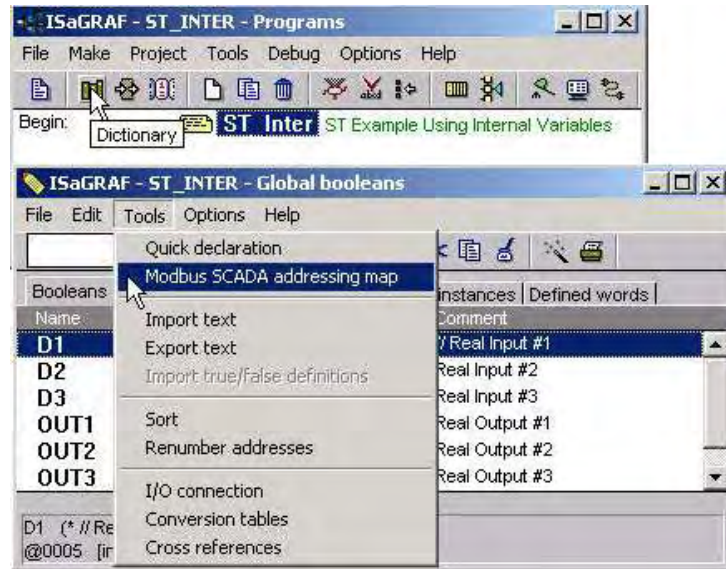


上圖,當您單擊”Store”按鈕時,你將會看到下圖 ”ISaGRAF Global Variables”視窗上的網路位址即馬上變成您所設定的數值。

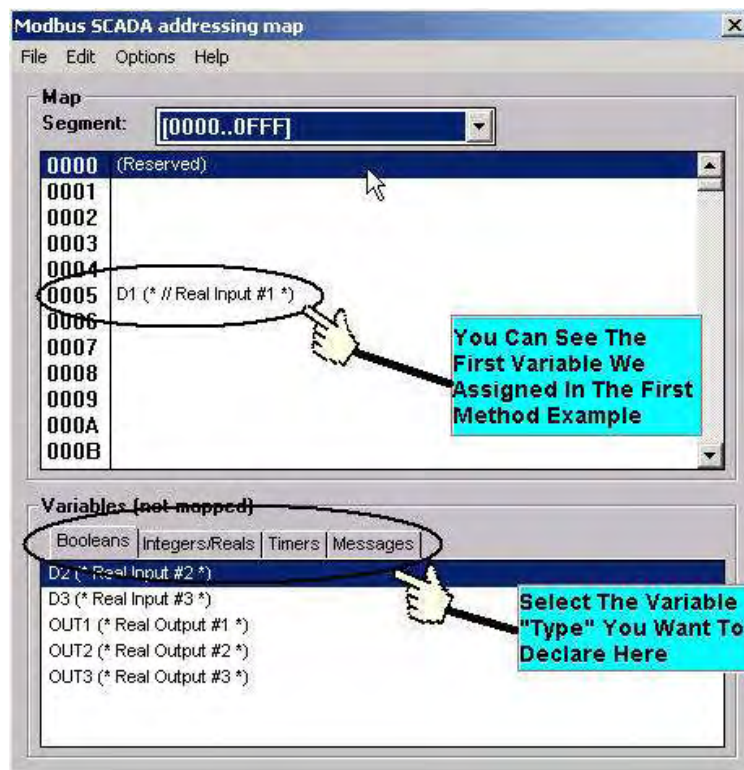


第二種方式為在您宣告變數完之後再設定網路位址，這種方式允許您連續指派網路位址給很多已經宣告好的變數。

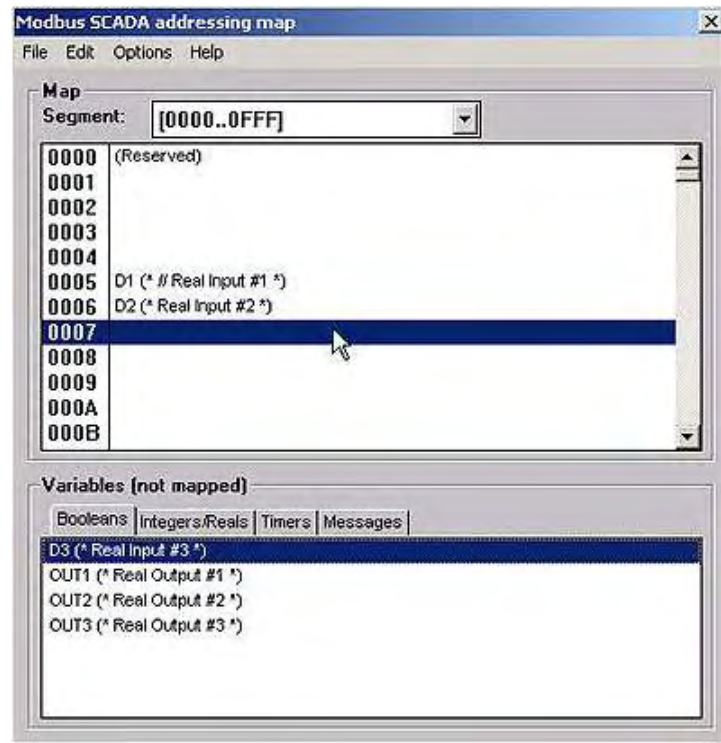
當您單擊 ”Modbus SCADA Address Map” (SCADA 是一個工業監控系統的縮寫 ”Supervisory Control And Data Acquisition), ”Modbus SCADA Address Map”視窗將會開啟。



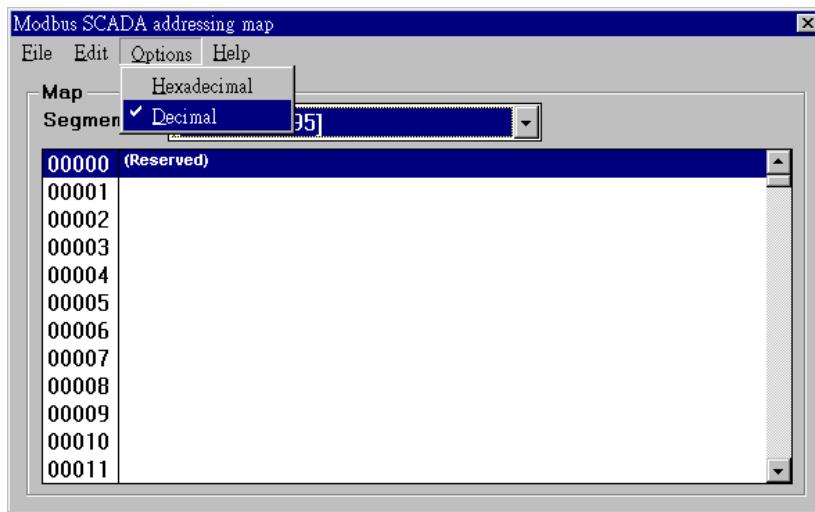
在下面的圖示中，變數 D1 在我們上一個網路位址範例就設定過了，所以它會放置在視窗上方的部份。而其他尚未設定網路位址的變數，將會顯示在視窗下方的部份。



若你想指定網路位址號碼給變數，您可以單擊視窗上方未被指派的號碼，然後雙擊視窗下方的變數名稱，這樣網路位址即會指派給你所選定的變數。



一般來說，網路位址編號用 16 進位的方式來表示是很不方便的,容易增加犯錯的機率。因此，將 16 進位格式轉換成 10 進位格式來顯示是很需要的。作法如下圖所示



關於 Modbus 網路位址的重要事項

“Modbus”網路位址編號的定義在 HMI 或其他的軟體程式中有時是不同的。這不同之處就是在某些其他的軟體程式所指派的網路位址號碼會比 ISaGRAF 控制器的變數位址號碼少 1 號。

像 InduSoft, Iconics、Citech、Wizcon、Kepware’s OPC server、Intellution’s”iFix”、Wonderware’s “Intouch”、National Instruments”Labview”、還有 ICP DAS’s Touch 506L, Touch 506T 和 Touch 510T 等裝置的變數網路位址編號定義都和 ISaGRAF 控制器相同。

而某些人機界面和 ISaGRAF PAC 的網路位址編號定義就不同，假如你指派”B”這個網路位址編號給這些產品，在 控制器 上對應的變數網路位址編號將需設定成”C”。若網路位址編號是”2”，在控制器上必須設定為”3”。

另外需注意，市面上賣的很多 SCADA 與 HMI，其網路位址編號的第 1 個數字通常指的是資料的型態及讀寫權限，這個數字在 ISaGRAF 內是不需要的 (I-8xx7, I-7188EG/XG, 7186EG, iP-8xx7 最大的編號只到 4095, WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 則到 8191)。以 iFix 為例，它與 ISaGRAF PAC 的編號對應如下：

iFix (10 進位)	ISaGRAF PAC 網路位址編號 (10 進位)
00001 (iFix 第 1 個數字 0 表示 讀/寫 Boolean)	1
00002	2
10010 (iFix 第 1 個數字 1 表示 讀 Boolean)	10
10011	11
31000(iFix 第 1 個數字 3 表示讀 Word)	1000
31001	1001
40101(iFix 第 1 個數字 4 表示讀/寫 Word)	101
42001	2001

ICP DAS 無法測試所有 SCADA 與人機界面程式或裝置的網路位址編號差異，假如你想用 Modbus 協定連接人機界面程式或其他裝置系統到 ISaGRAF 控制器系統，記得您必需先測試一下這些產品的位址編號偏移量是否相同或差 1 號，以方便它們之間的溝通。

程式設計者若要用 Visual Basic 或 Visual C++來開發他們自己的界面程式，建議參考第 5 章，裡面說明關於在 Modbus 上開發界面程式的通訊協定。

注意：

在 I-8xx7, μPAC-7186EG, I-7188EG/XG, iP-8xx7, VP-2117 控制系統裡，一個 Modbus 命令不能詢問超過 255 位元(bit 或 Boolean)，且不能一次 Modbus 命令詢問超過超過 122 個字組 (word)，它必需拆成兩個或兩個以上的 Modbus 命令來傳送。

但對 WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 來說，一個 Modbus 命令可以詢問達 1968 位元 (bit 或 Boolean)，而一次 Modbus 命令一樣不可詢問超過超過 122 個字組(word)

4.2: 使用 Modbus 讀寫 Words, Long Words 及 Float 資料

Modbus protocol 提供函式 3 及 4 來讀取多個字組(word)，函式 6 和 16 來寫入字組(word)。在第五章裡有更多 Modbus 的相關訊息。

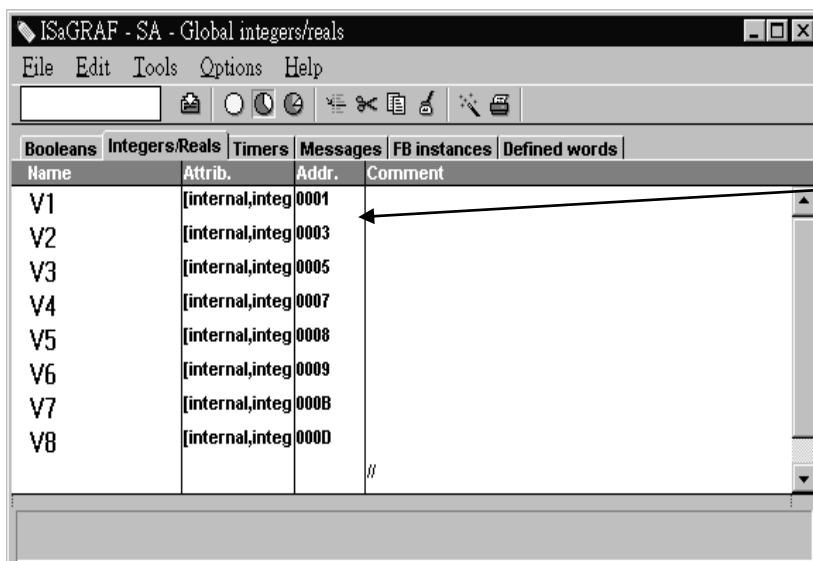
定義在 ISaGRAF 控制器 Modbus 協定上的 Word 是一個 Signed Short Integer，佔了兩個位元組 (bytes)，範圍值從-32768(16 進位表示法為 8000) ~ 32767(16 進位表示法為 7FFF)。它通常都是被用來描述類比點的物理量。例如，I-87017W- I/O 板(請參考 3.2 節)

I-87017W :

Range 識別號碼 (16 進位表示)	物理量數值	輸入點範圍值(10 進位)		
		-32768	0	+32767
8 (default)	± 10V	- 10V	0V	+ 10V
9	± 5V	- 5V	0V	+ 5V
A	± 1V	- 1V	0V	+ 1V
B	± 500mV	- 500mV	0mV	+ 500mV
C	± 150mV	- 150mV	0mV	+ 150mV
D	± 20mA	- 20mA	0mA	+ 20mA

定義在 ISaGRAF 控制器 Modbus 協定上的一個 Long Word，佔了四個位元組，它的範圍值從最小的 -2,147,483,648(16 進位表示法為 8000 0000) 到 +2,147,483,647(16 進位表示法為 7FFF FFFF)。它通常是被用來表示宣告在 ISaGRAF Workbench 的內部整數變數的值。

宣告在 ISaGRAF 內的整數變數都是 Signed 32-bit Format。然而在 Modbus 傳輸協定裡，有宣告網路位址編號的整數變數，僅佔 1 個 word(2 byte)，而 1 個 Long Integer 佔兩個 word(4 bytes)，所以要透過 Modbus 來讀寫 Long Word，網路位址編號必須如下方圖示的方法來指定。

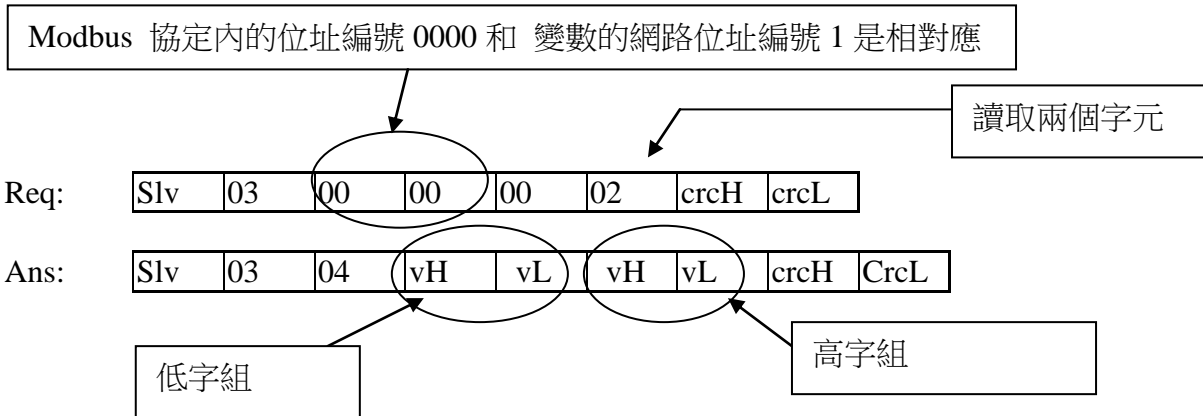


在 Modbus 協定中，若 V1 的網路位址編號被指定為 1，而且網路位址編號 2 未指定給任何變數使用，這時 V1 在 Modbus 協定內將佔有兩個 Word (共 4 bytes)

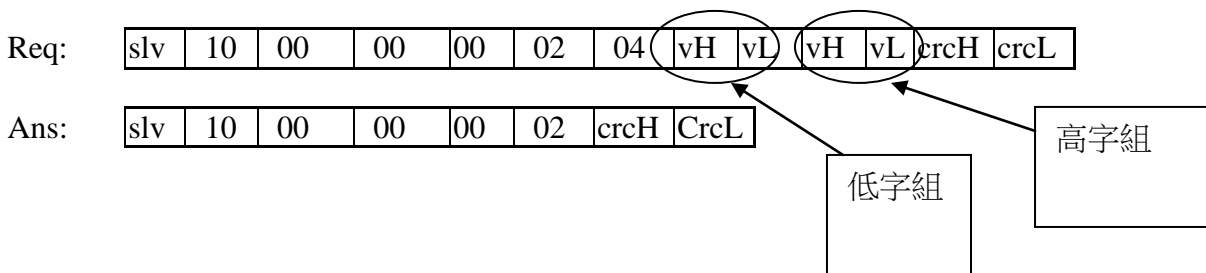
若網路位址編號 2 有指定給別的變數使用，這時 V1 在 Modbus 協定內就只佔有一個 Word (2 bytes)

在左方的例子中，V1,V2,V3,V6,V7 和 V8 佔有兩個 Word (4 bytes)，而 V4 和 V5 只佔有一個 Word (2 bytes)

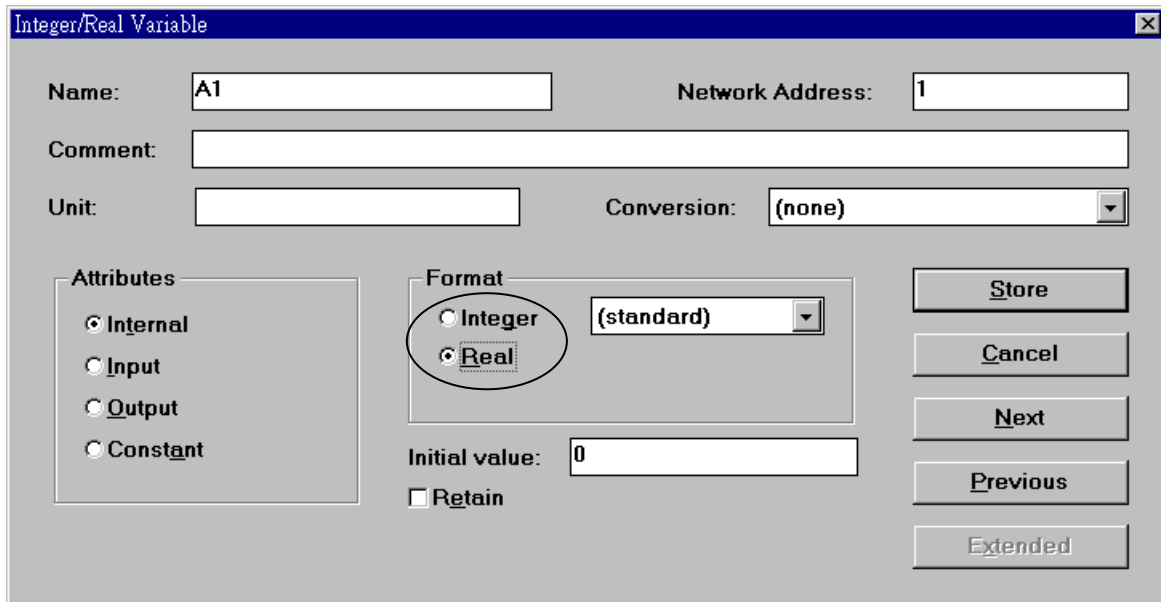
要讀取長字組 V1 的值，可藉由 Modbus 函式 3，讀取二個字組的資料 (請參考 5.1 節)



要寫入長字組 V1 的值，可藉由 Modbus 函式 16，寫入二個字組



讀寫 Float 和讀寫 Long Word 是大同小異的。唯一不同的地方是您在宣告變數時，變數格式必需設為 "Real" 型態，且下一個網路位址編號不可指派給其他變數。



在市面上有很多現成的 HMI 套裝軟體，您需要注意它們是否有支援 Modbus 協定，且重要的是您在 ISaGRAF 中是否設 對 變數的網路位址編號。

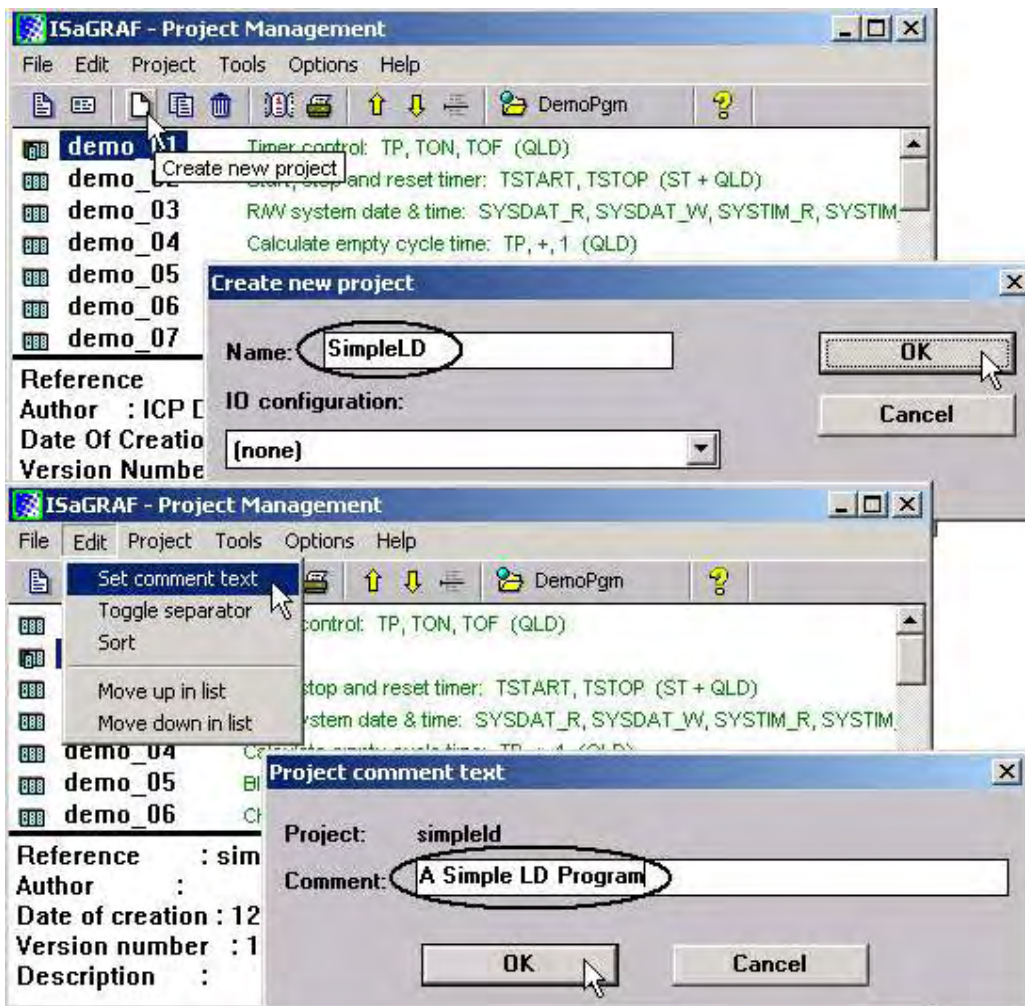
4.3: 將 I-8xx7 當成 Modbus I/O 或 Modbus TCP/IP I/O 來使用

有一些人機界面軟體會從 Modbus I/O 模組中收集 I/O 資料。這些 I/O 模組會定期掃描輸入板和更新輸出板的資料值。基本上，在這些 I/O 模組內並沒有控制邏輯在裡面，而是靠外部的人機界面軟體來進行資料的處理。為了配合這種使用方法，I-8xx7, iP-8xx7, I-8437-80, I-8837-80 可被當成 Modbus I/O 模組來使用。另外 I-8437-80, I-8837-80 與 iP-8x47 可當成 Modbus TCP/IP I/O 模組來使用。使用方法請參照下列的範例(若您還不熟悉 ISaGRAF 程式設計，建議參閱第二章)。

建立一個新的專案

您可以參考 2.1.1.2 節

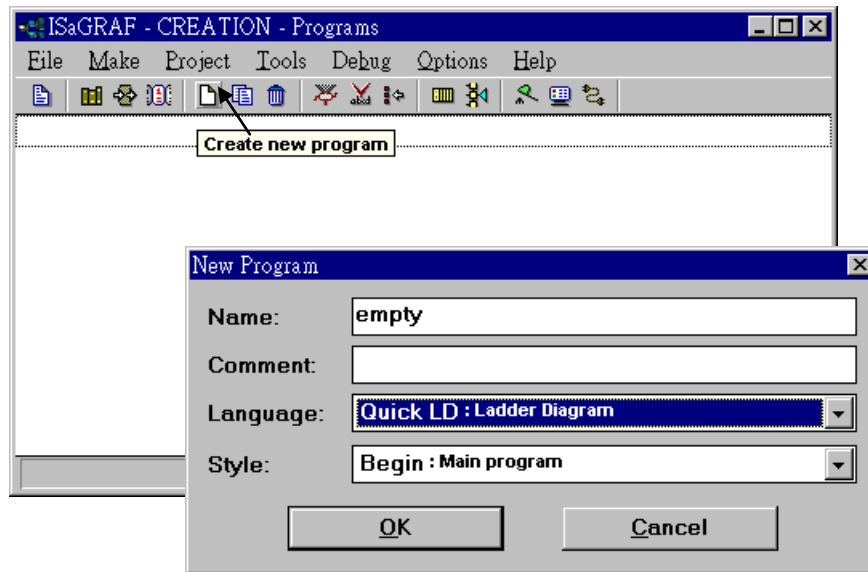
例子:



建立一個新的程式

不需在程式內寫控制邏輯

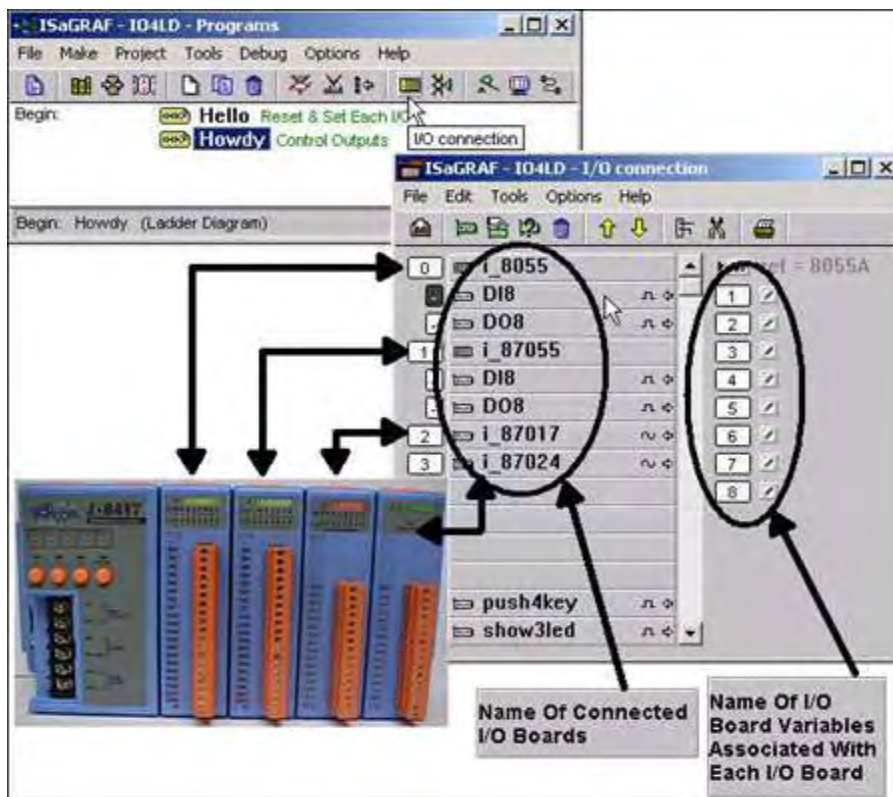
例子:



連結 I/O 模組

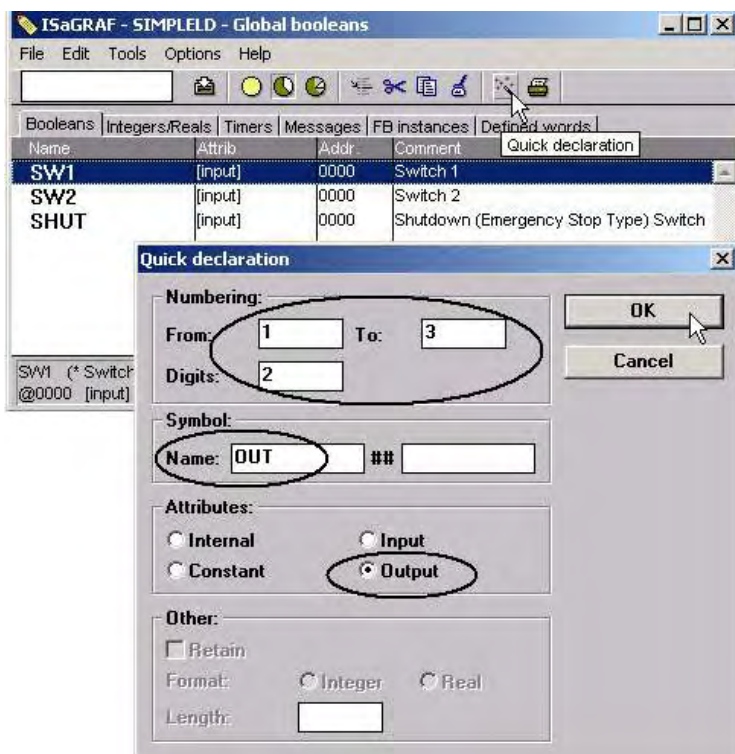
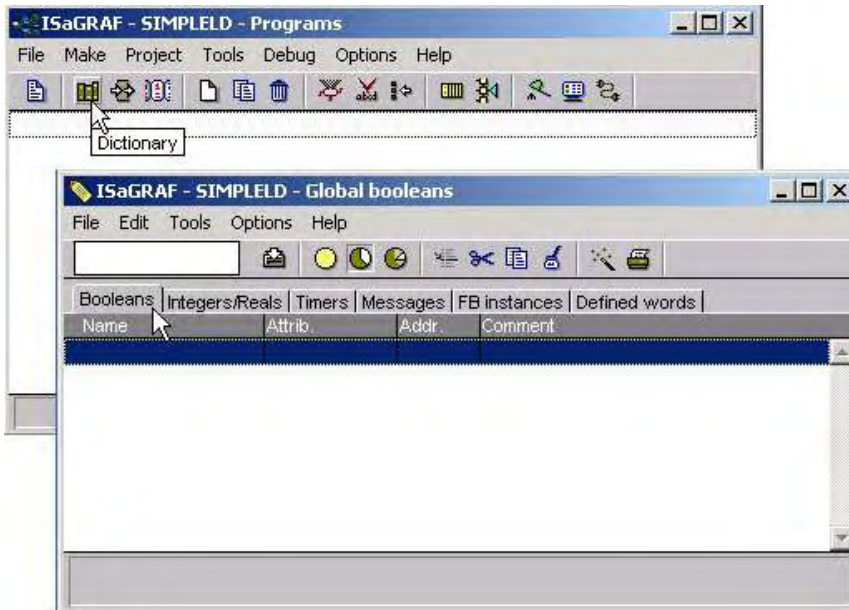
您可以參閱 3.1 節

例子:

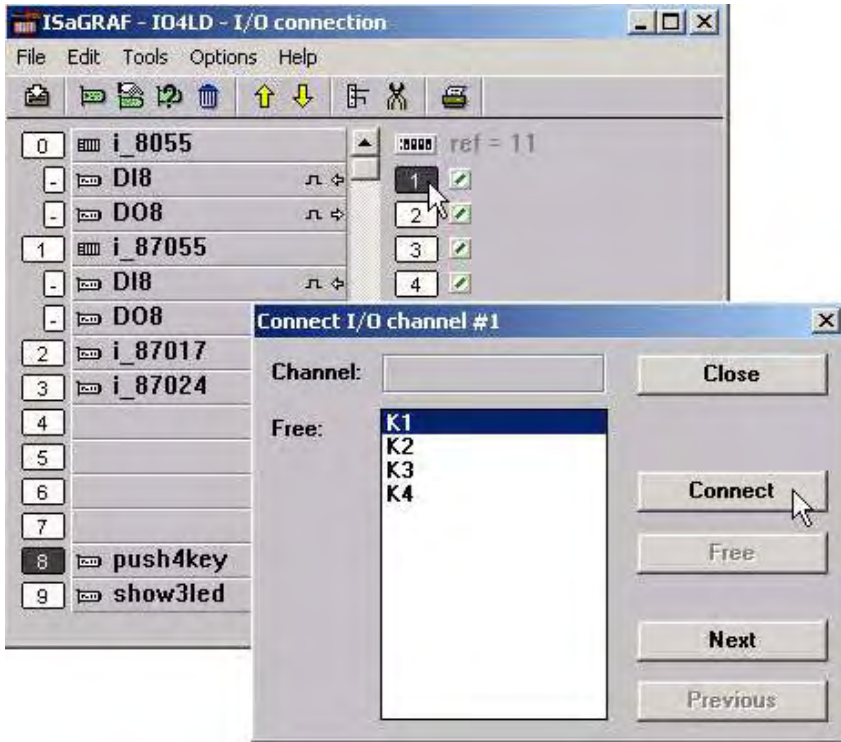


宣告 I/O 模組內的輸出/入點的變數名稱
你可以參閱 2.1.1.3 節

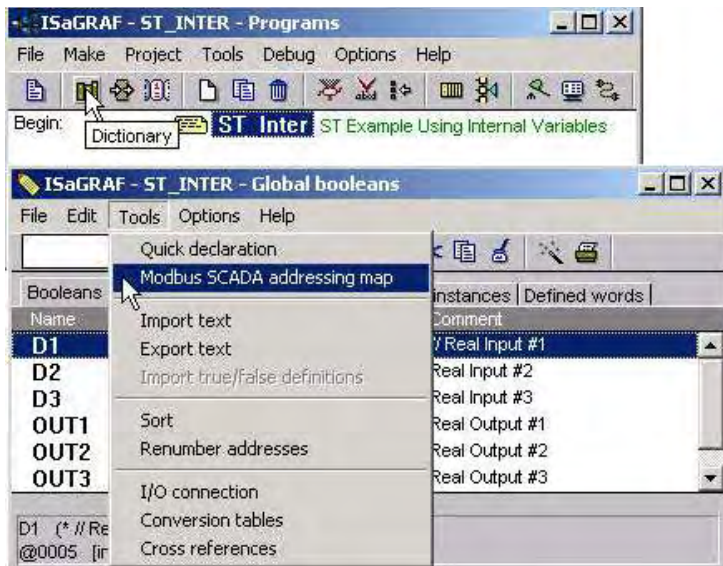
例子:



連結變數到 I/O 模組的輸出入點。
 您可以參閱 3.1.2 節
 例子:



指派網路位址編號給變數
 您可以參閱 4.1 節
 例子:



編譯 & 下載 專案

您可以參閱 2.1.3 & 2.1.5 節

注意:

若使用 Modbus TCP/IP 協定, 請確認 I-8437-80, I-8837-80, iP-8x47 的 NET-ID (1.3.1 節)、IP、Mask(附錄 B)是否設定正確。若設定無誤, 你的人機介面(HMI)程式才可以透過 Modbus TCP/IP 網址來存取 I/O 模板的資料。

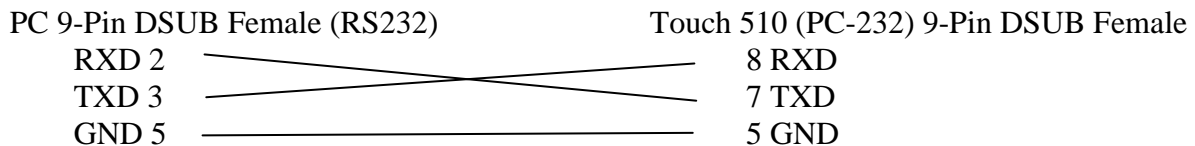
4.4: 連結 ISaGRAF PAC 和 Touch 500 系列

Touch500 系列 HMI 可以使用下表的通訊規約，來與 ICP DAS 的 ISaGRAF 控制器相連接。

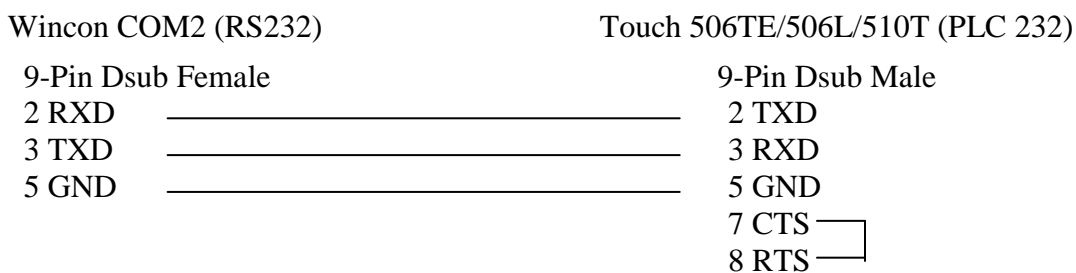
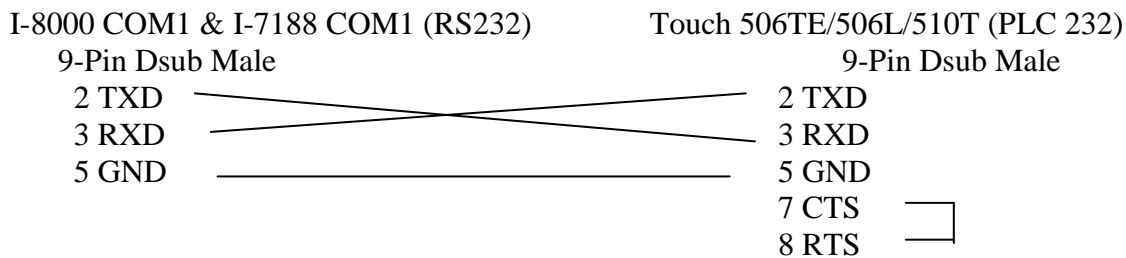
型號	支持的通訊規約
Touch-506L	Modbus RTU RS-232 , Modbus RTU RS-485
Touch-506TE	Modbus RTU RS-232 , Modbus RTU RS-485 , Modbus TCP/IP
Touch-510T	Modbus RTU RS-232 , Modbus RTU RS-485

您必需在電腦上先安裝 ”EasyBuilder 500” 這套軟體,才能開發 Touch 506L, 506TE, 510T 的人機界面程式. 請由 <http://www.icpdas.com/download/others/touch/touch.htm> 下載最新的軟體 “**setup.zip**”跟手冊 或由 I-8000 CD-ROM:\napdos\others\touch\500series\setup\ 執行 “**setup.exe**” (2.7.1 版起)

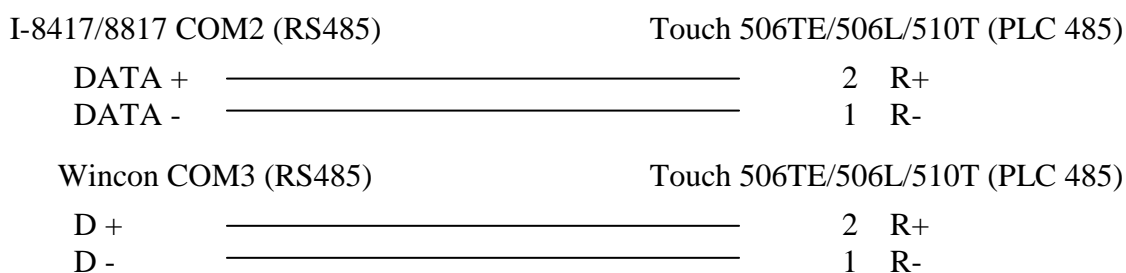
PC 連接到 Touch 500 系列產品的 RS232 通訊線接腳如下 (PC 下載 HMI 畫面時使用).



PC 下載畫面到 ”Touch 500 系列” 後, 若是使用 RS-232 接線由 Touch 500 連到 控制器, 如下



若是使用 RS-485 接線, 由 Touch 500 連到 控制器, 接線 的接腳圖如下



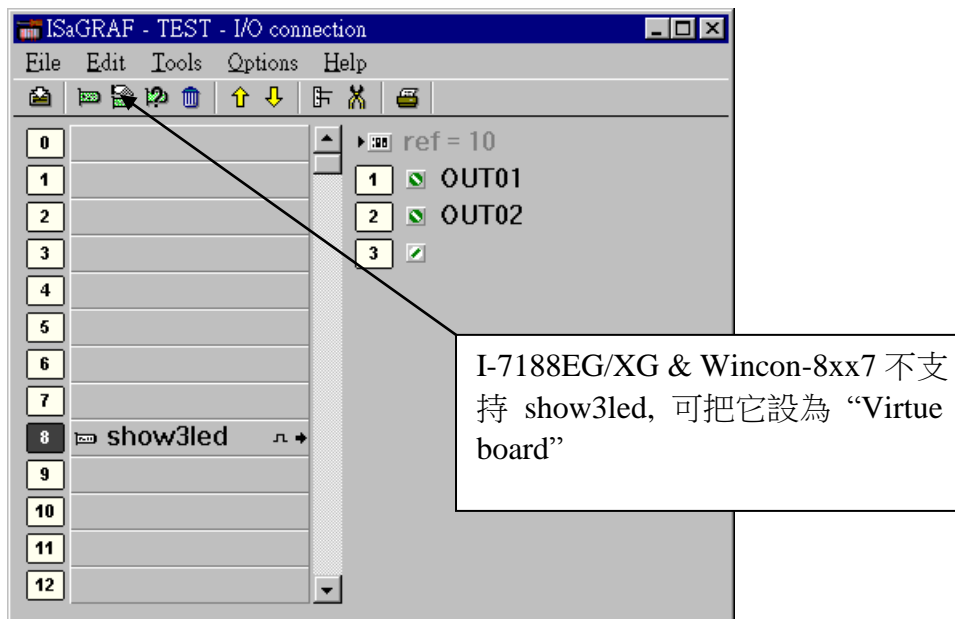
4.4.1: 設計 ISaGRAF PAC 應用程式

ISaGRAF 控制器內變數的網路位址編號一定要設定，不然 "Touch 500" 將無法存取控制器的資料。(請參考 4.1、4.2 節) 若你對 ISaGRAF 的程式語法不熟悉，建議您複習第二章。

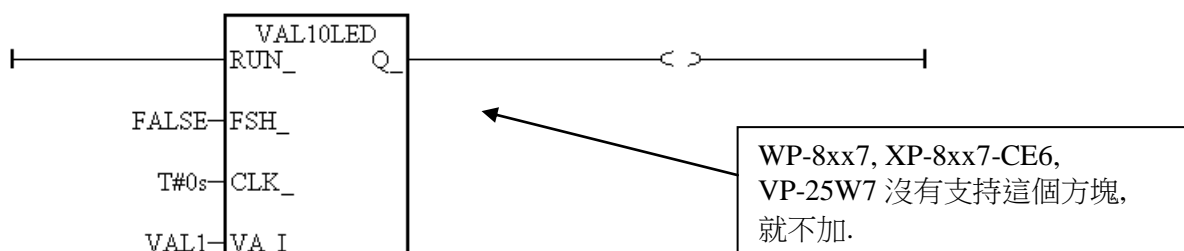
在這個範例中所使用的變數名稱

名稱	型態	屬性	網路位址	其他
OUT01	布林	輸出	0001	-
OUT02	布林	輸出	0002	-
VAL1	整數	內部	000A (10)	-

IO 連結:



寫一個階梯圖程式，將 "VAL1" 的結果顯示在 7 段式 LED 燈。



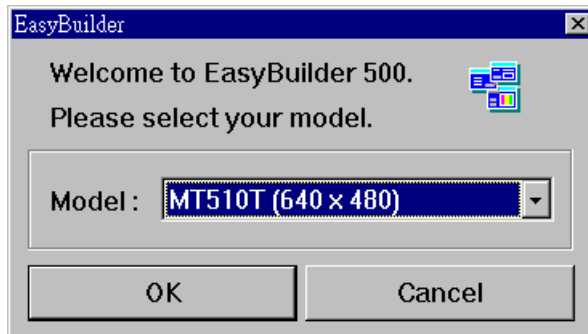
當你完成這個專案程式後，編譯和下載程式到 控制器。

4.4.2: 設計 Touch 500 畫面

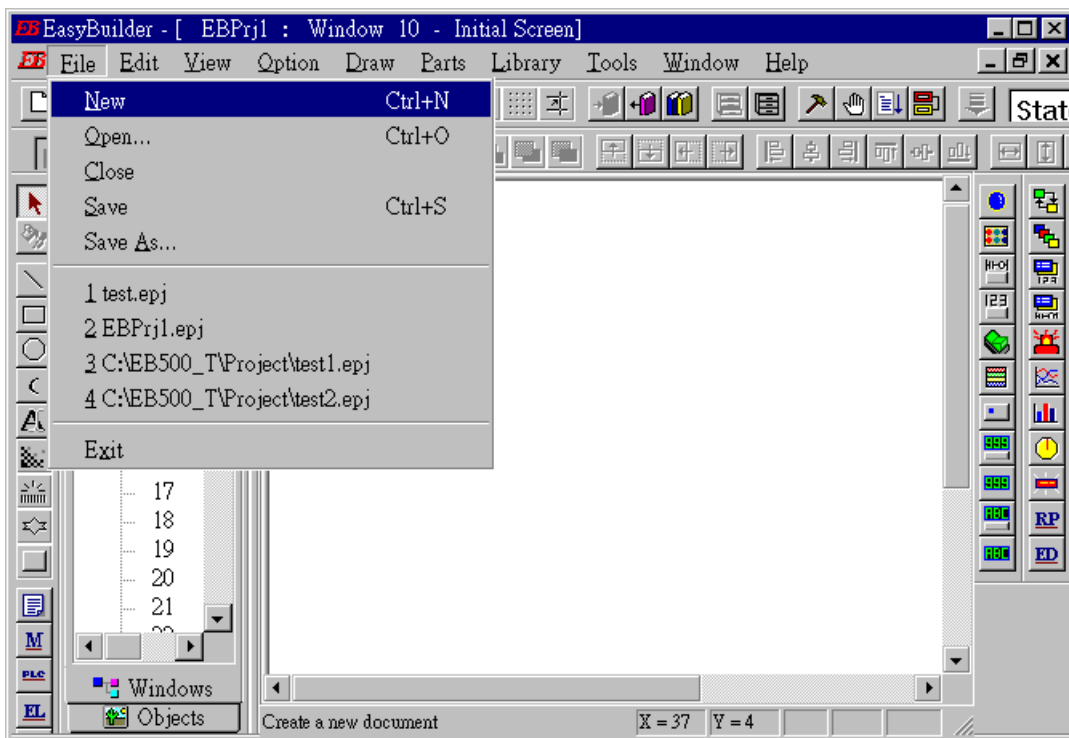
“EasyBuilder 500”可以用來設計”Touch 500”系列的人機介面圖示，所以在這一節我們用 ”EasyBuilder 500” 來為 ”Touch 510” 寫一段圖示，幫助各位了解如何使用”EasyBuilder 500”。

(更多關於 Touch series 的程式設計技巧，參考我們所提供的 Touch series 使用手冊)

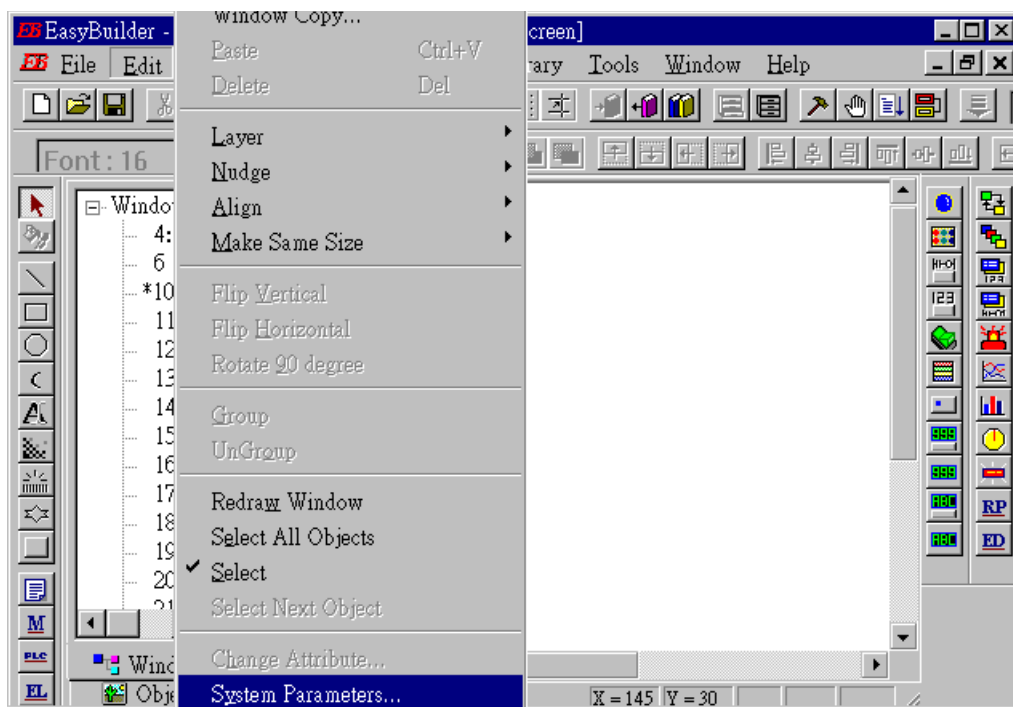
請點選左下角的[開始]->[程式集]->[EasyBuilder]->”EasyBuilder” 按鍵，則將會出現下面的視窗畫面，選擇適當的應用開發模式。



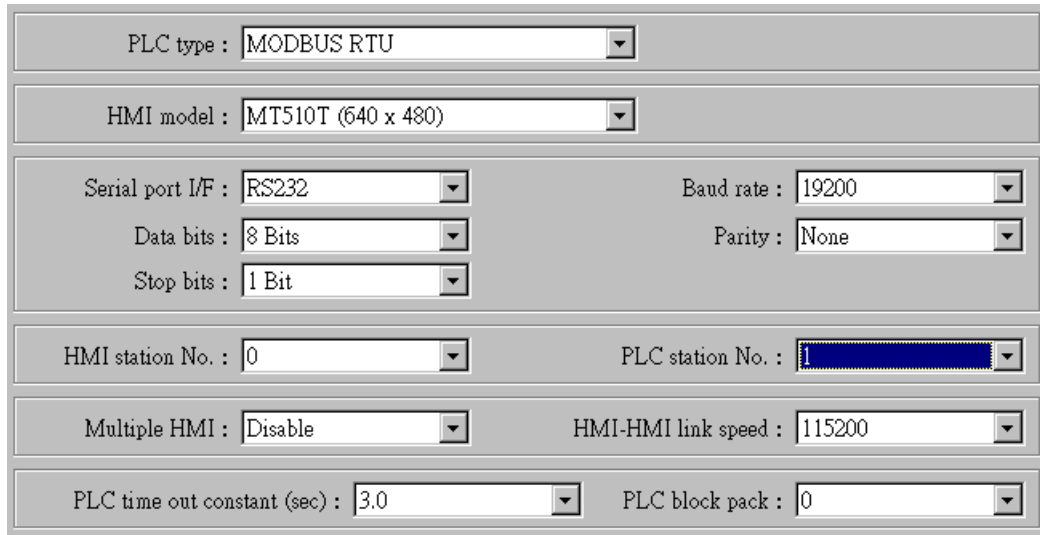
按下[File]->”new”，建立一個新的專案。



按下[Edit]->“System Parameters” 來設定”Touch 510” 和 ISaGRAF 控制器之間的通訊參數。



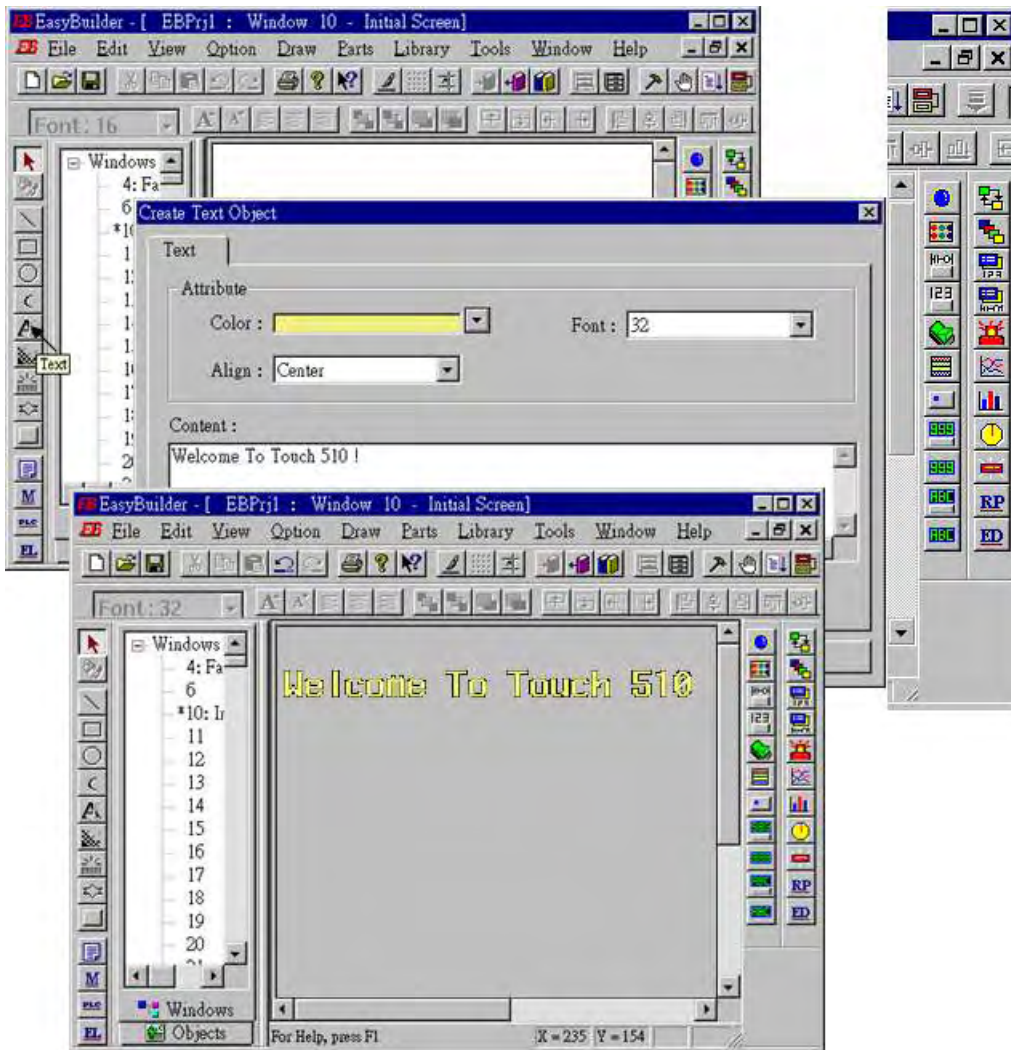
在下面視窗中的 PLC type 要設成”MODBUS RTU”, Serial port 設成 “RS232”, Data bits 設成 “8 Bits”, Stop bits 設成“1 Bit”, Baud rate 設成“19200”, Parity 設成 “None”, PLC station No. 和 ISaGRAF 控制器的 Net-ID 都設成 1。



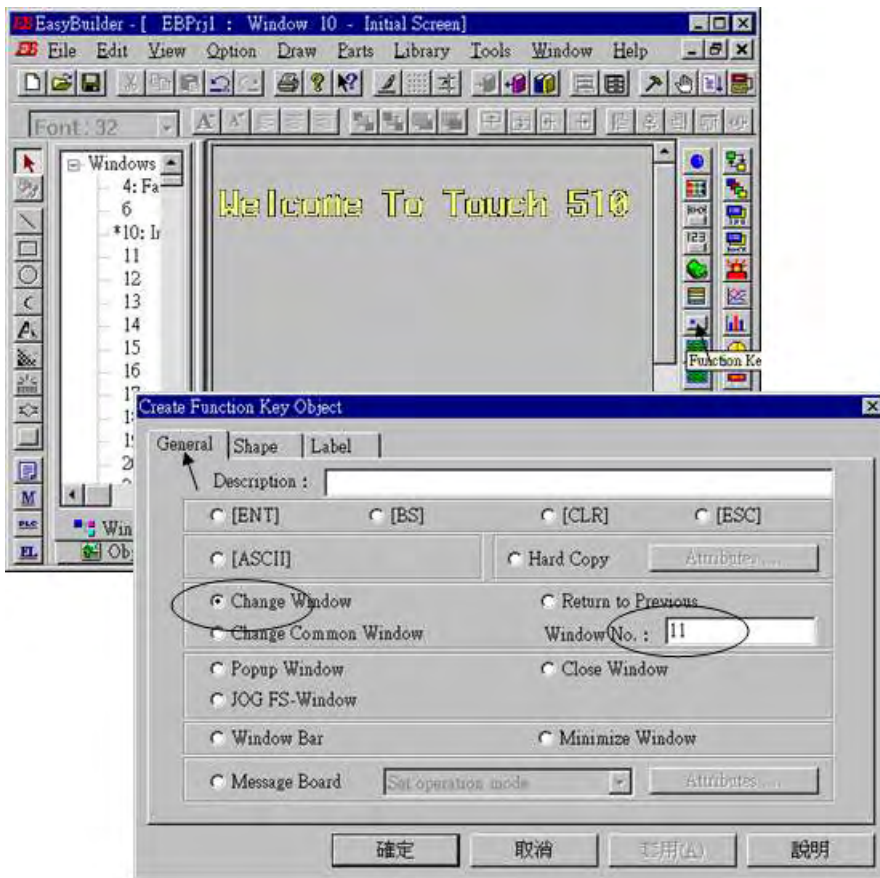
注意:

- 1.若是使用 **Touch506TE** 的 **Ethernet** 來連接, PLC type 要設成 “MODBUS RTU TCP/IP”, PLC I/F port 要設成 “Ethernet”, Local IP address 為 Touch506TE 要設成的 IP, 而 Server IP address 則是 控制器的 IP 地址, PLC station No 則需設為 控制器的 NET-ID (一般是 1 號)
2. 若是使用 **2 線式 RS-485** 來連接, 則 PLC type 要設為 “MODBUS RTU(485 2W)”, PLC I/F port 要設成 ”RS-485 2W”, 其它則與使用 RS-232 的設法相同.

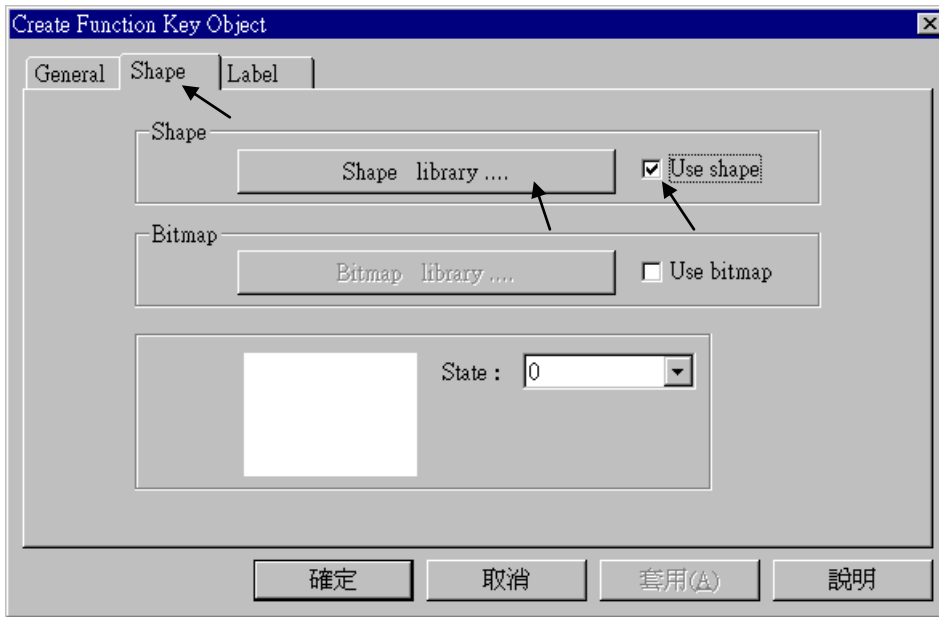
按下”Text”，選擇自己喜歡的顏色、字型，在 ”content” 內編輯你想要輸入的文字。



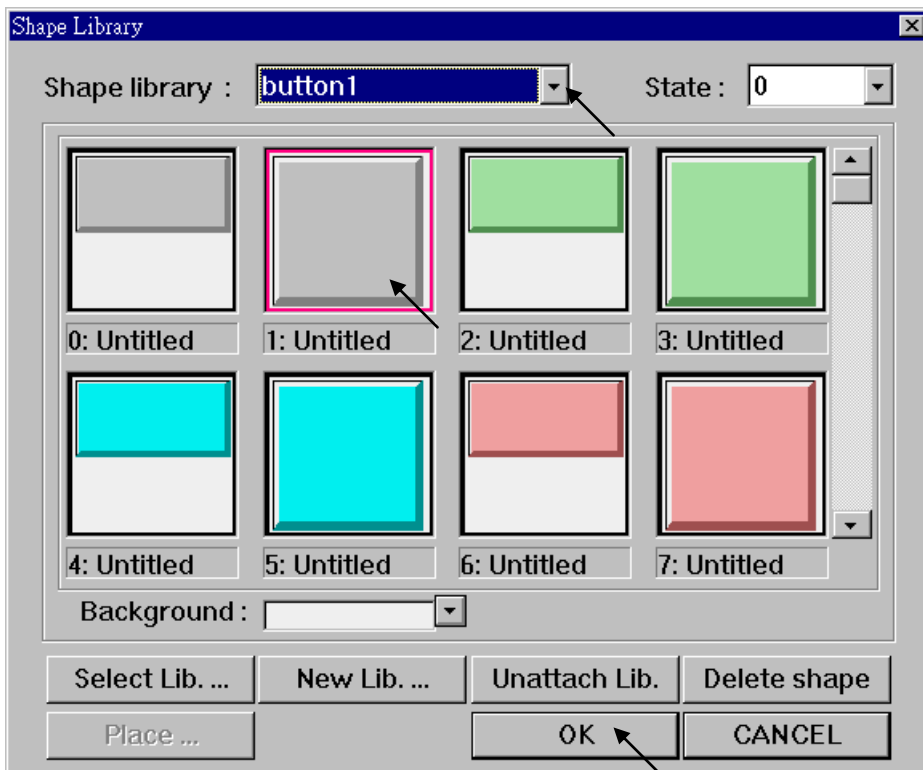
按下[Function Key]→”General”，選擇 ”Change Window” 和設定 ”Window No.”的值為 11。來增加一個視窗的按鈕。



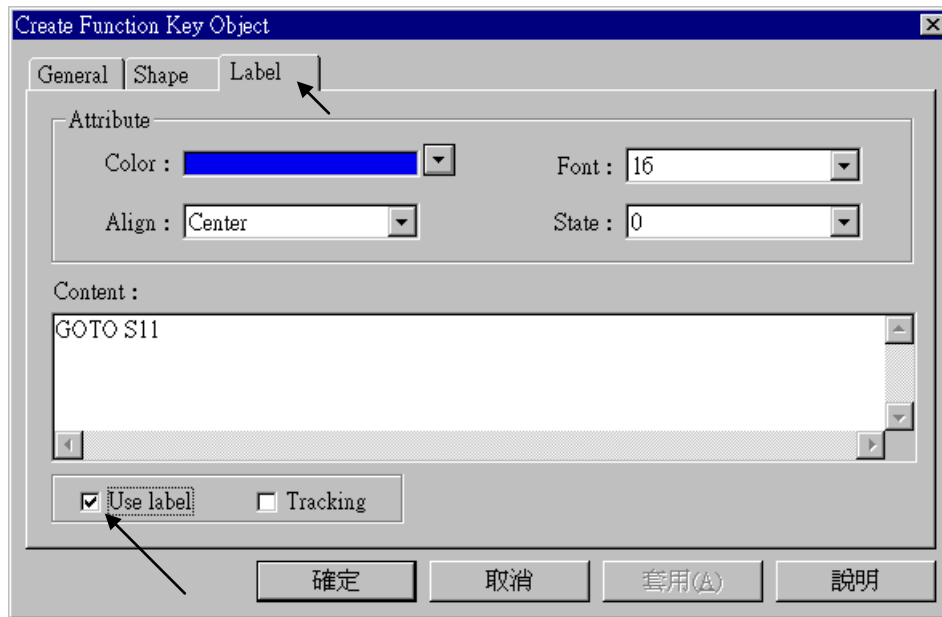
按下 "Shape"，然後選擇 "Use shape"，單擊 "Shape library"。



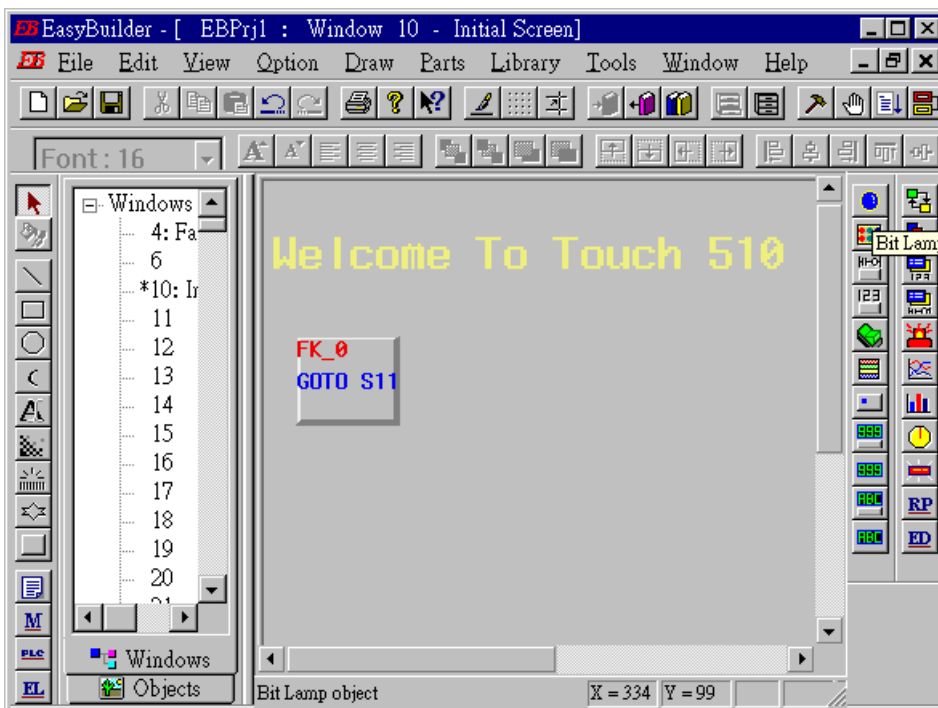
選擇您比較喜歡的 "Shape library"，然後按 "OK"。



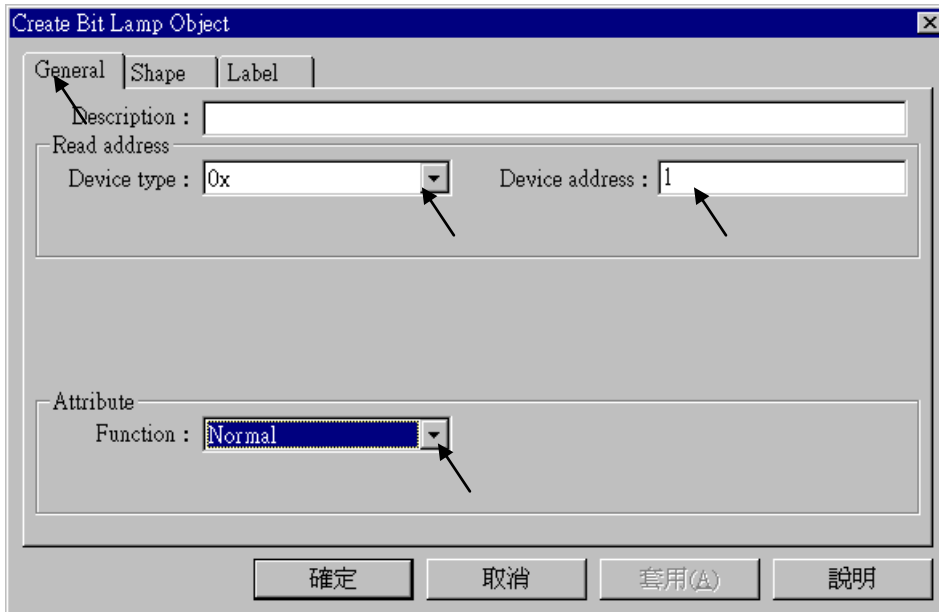
按下“Label”選擇您所喜歡的顏色、字型，在下面的“Content”內容打上”GOTO S11”。確定視窗下方的”Use label”一定要選取。



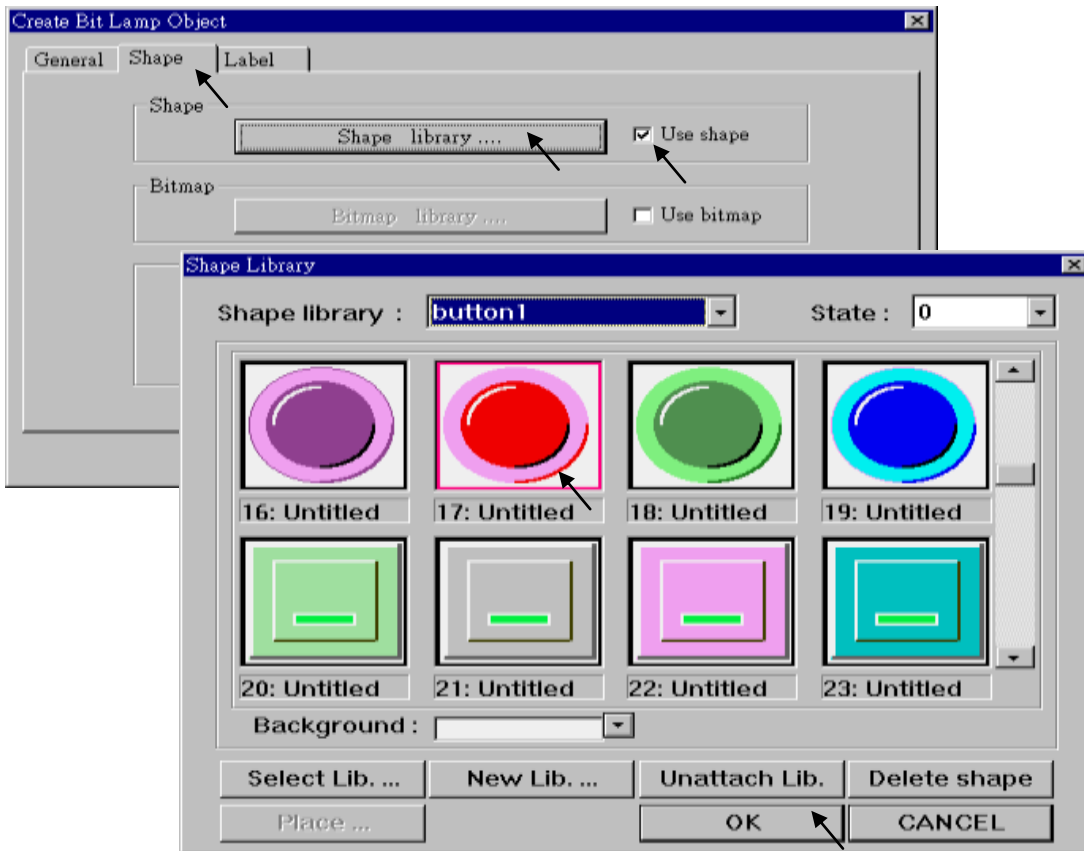
按下“Bit Lamp”鍵



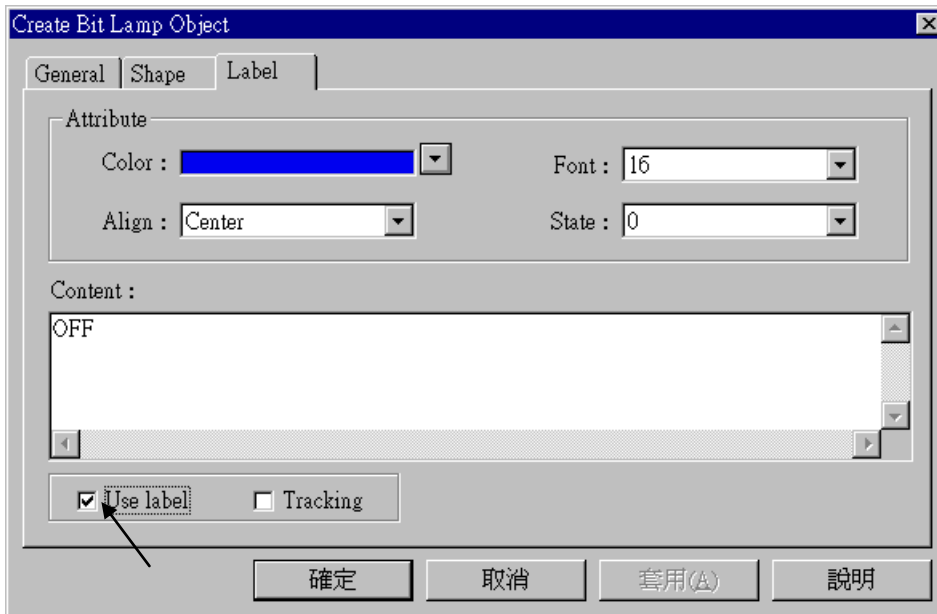
按下“General”鍵，將“Device type”設定為“0x”(0x 代表布林變數)，”Device address”設定為”1”(這個值即為 I-8xx7 的變數網路位址編號)，然後設定“function”為”Normal”。



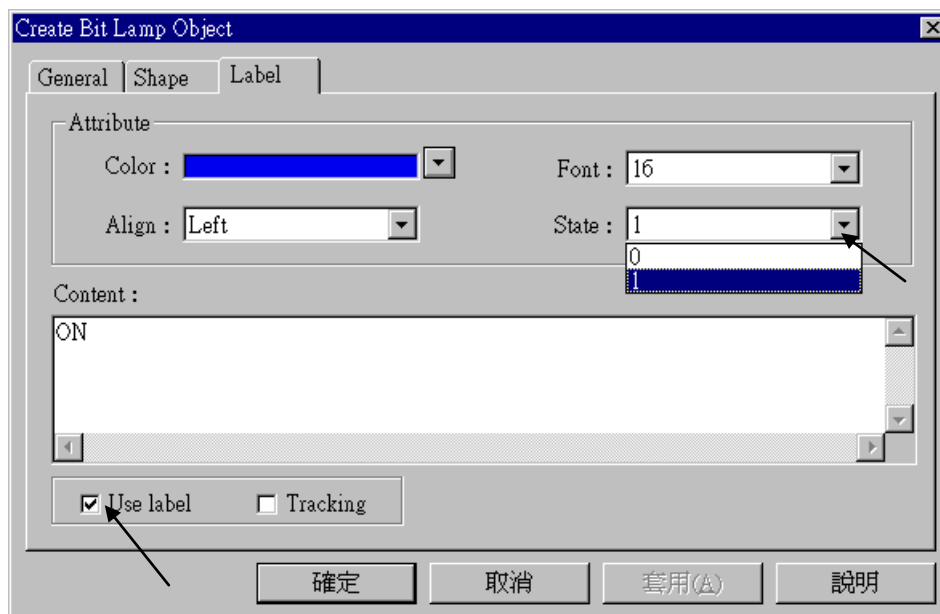
跟之前一樣，選擇您喜歡的”Shap library”



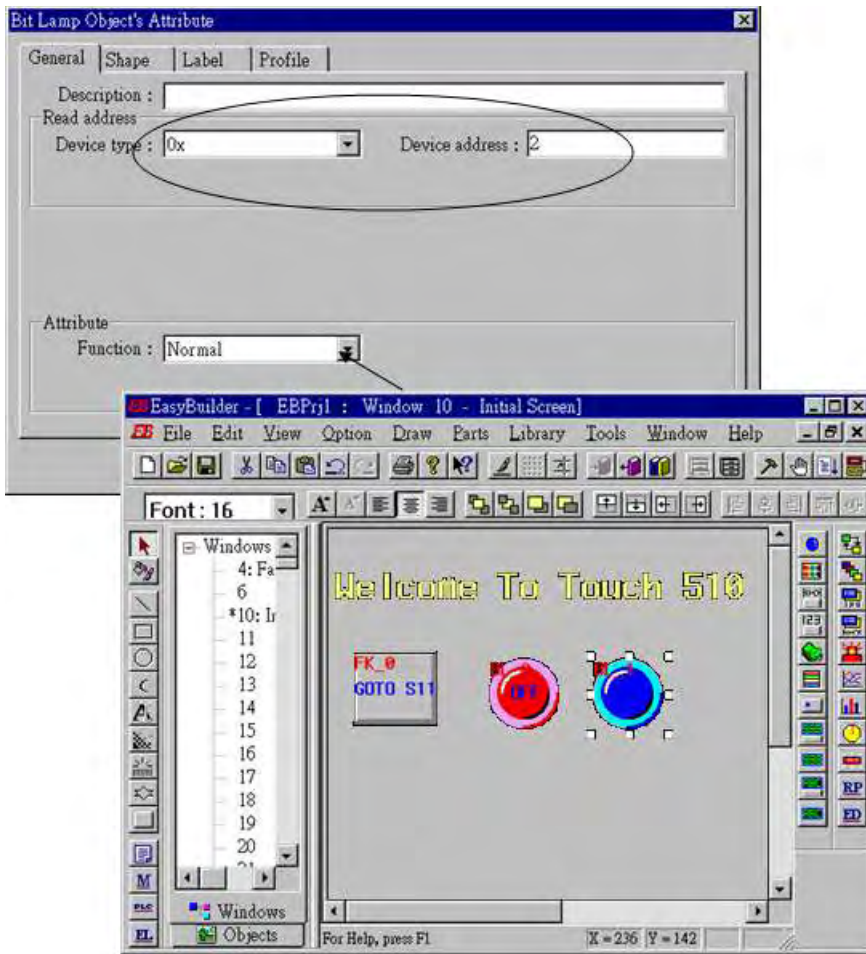
接著按下 “Label” 鍵，在 “Content” 內填入 “OFF”、”State” 設定成 0，確定視窗下方的 “Use label” 一定要選取。



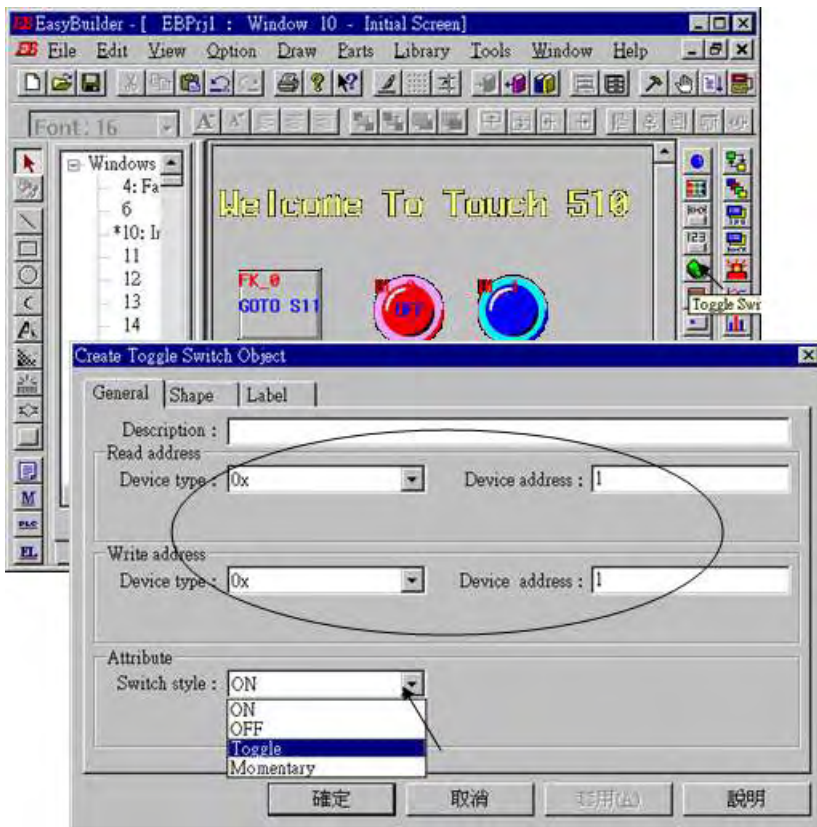
將 ”State” 改變成 1，在 “content” 內填入 ”ON”，確定 “Use label” 是否有選取。



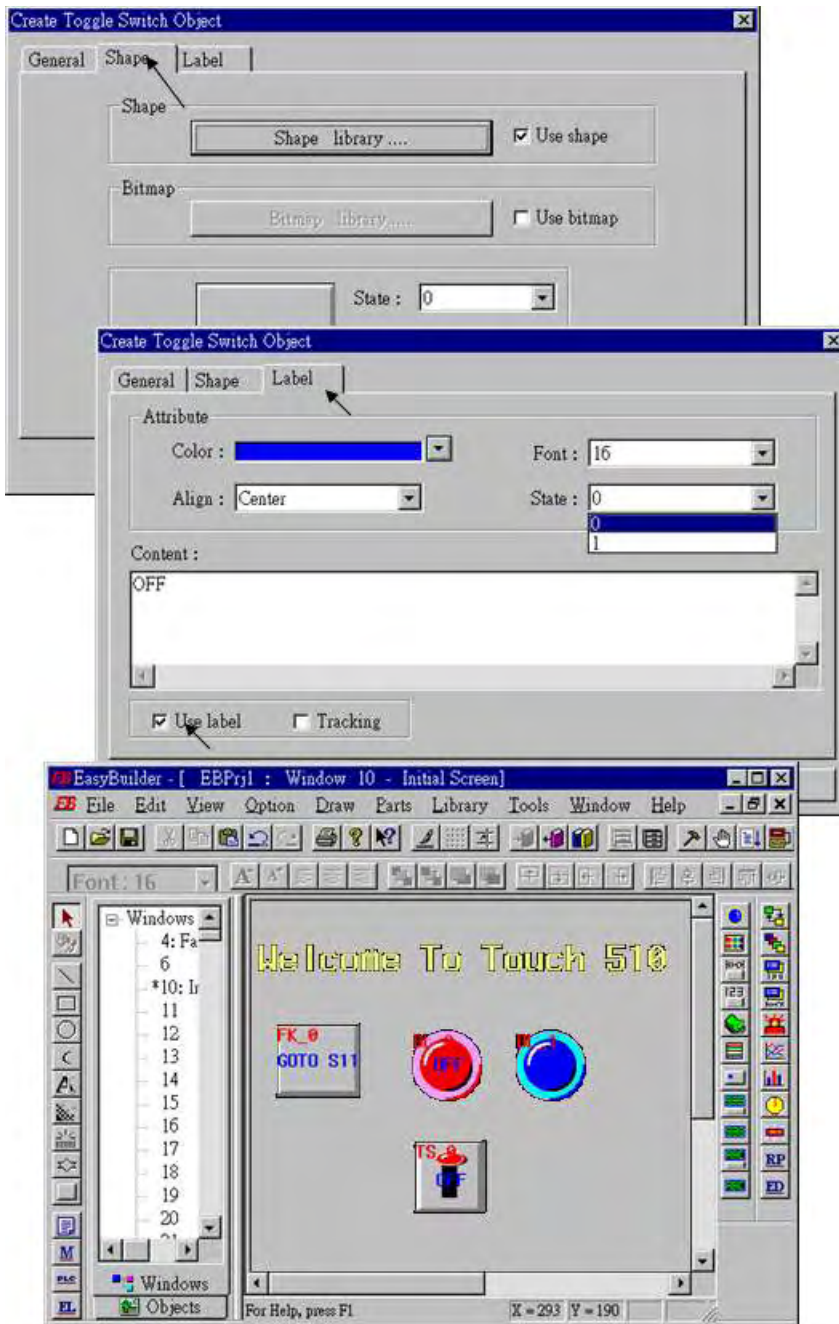
跟之前同樣的方式，建立一個燈的圖示，”Device address”設成 2。



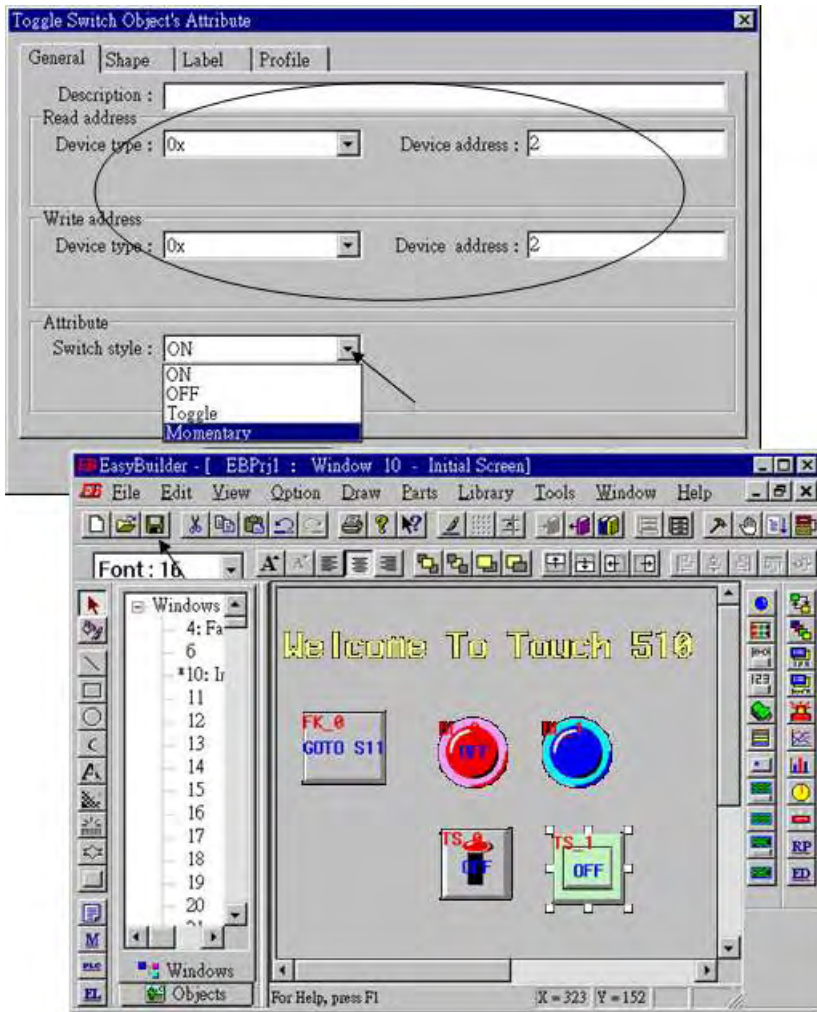
按下 "Toggle switch" 將 "Dvice Type"均設成 "0x" , "Device address"均設成 1 , "Switch Type" 設為"Toggle"。



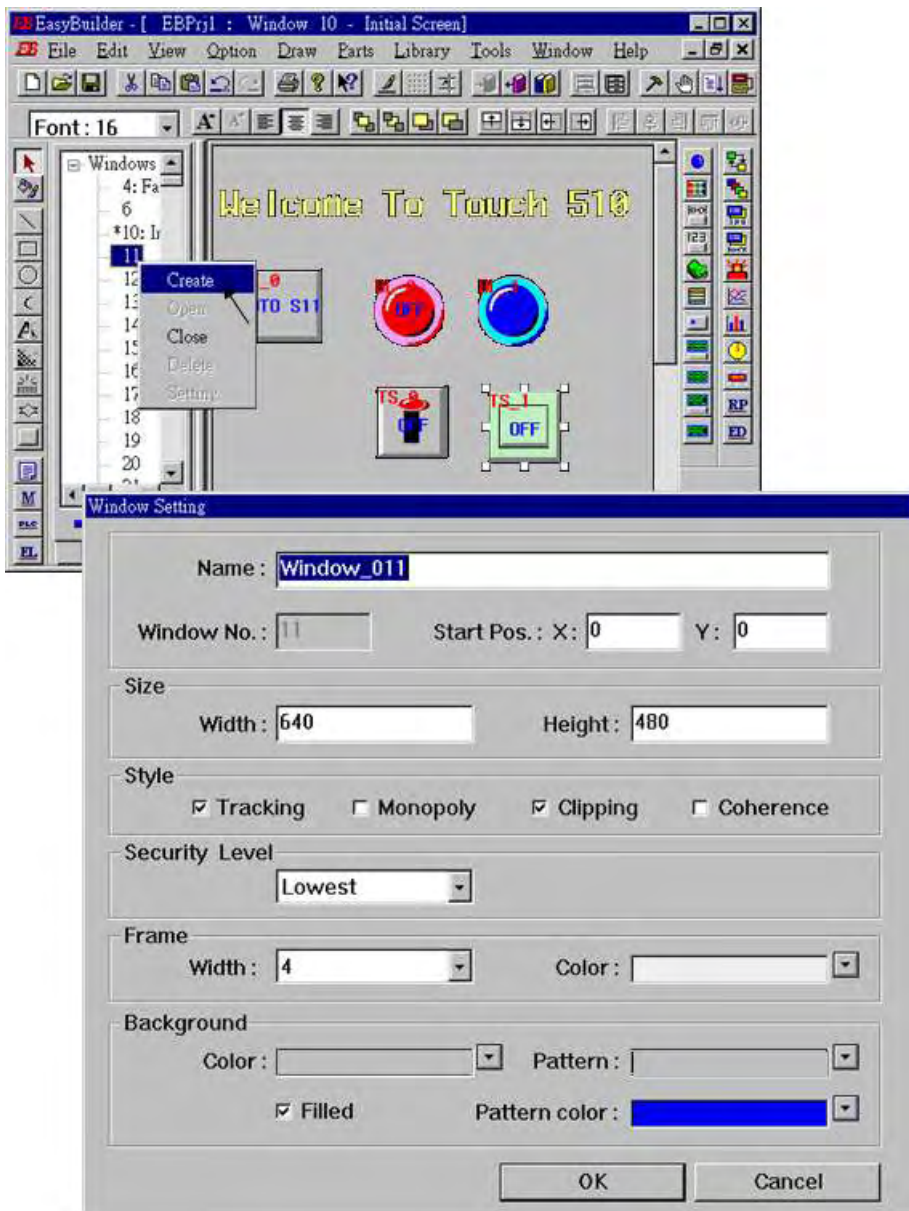
選擇自己喜歡的形狀和顏色。



建立另一個 “Toggle Switch”，將其” Device address”設成 2、”Switch style” 設成 ”Momentary”，接著把這個專案儲存起來。



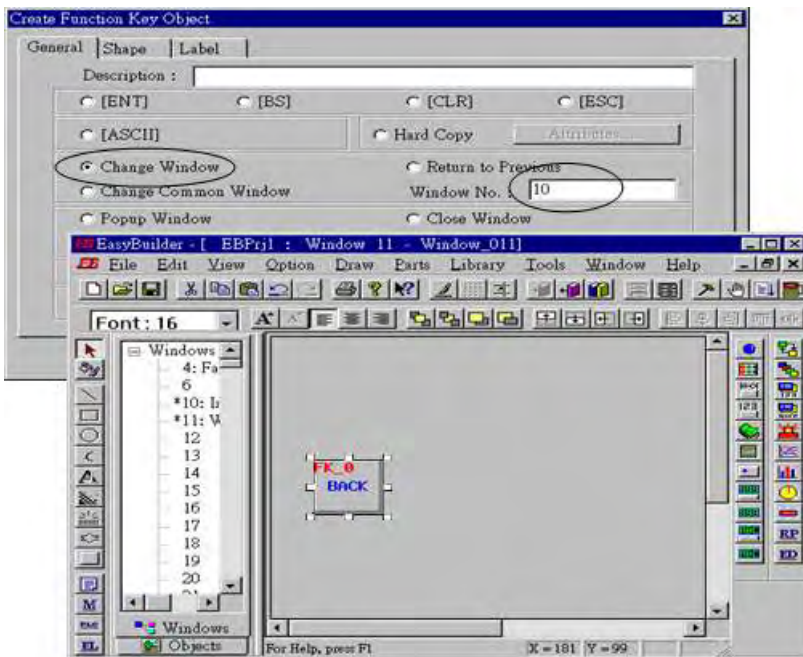
接著我們來設計另一個新的視窗畫面，按著[windows]→11，然後按右鍵，選擇”Create”來建立一個新的視窗。



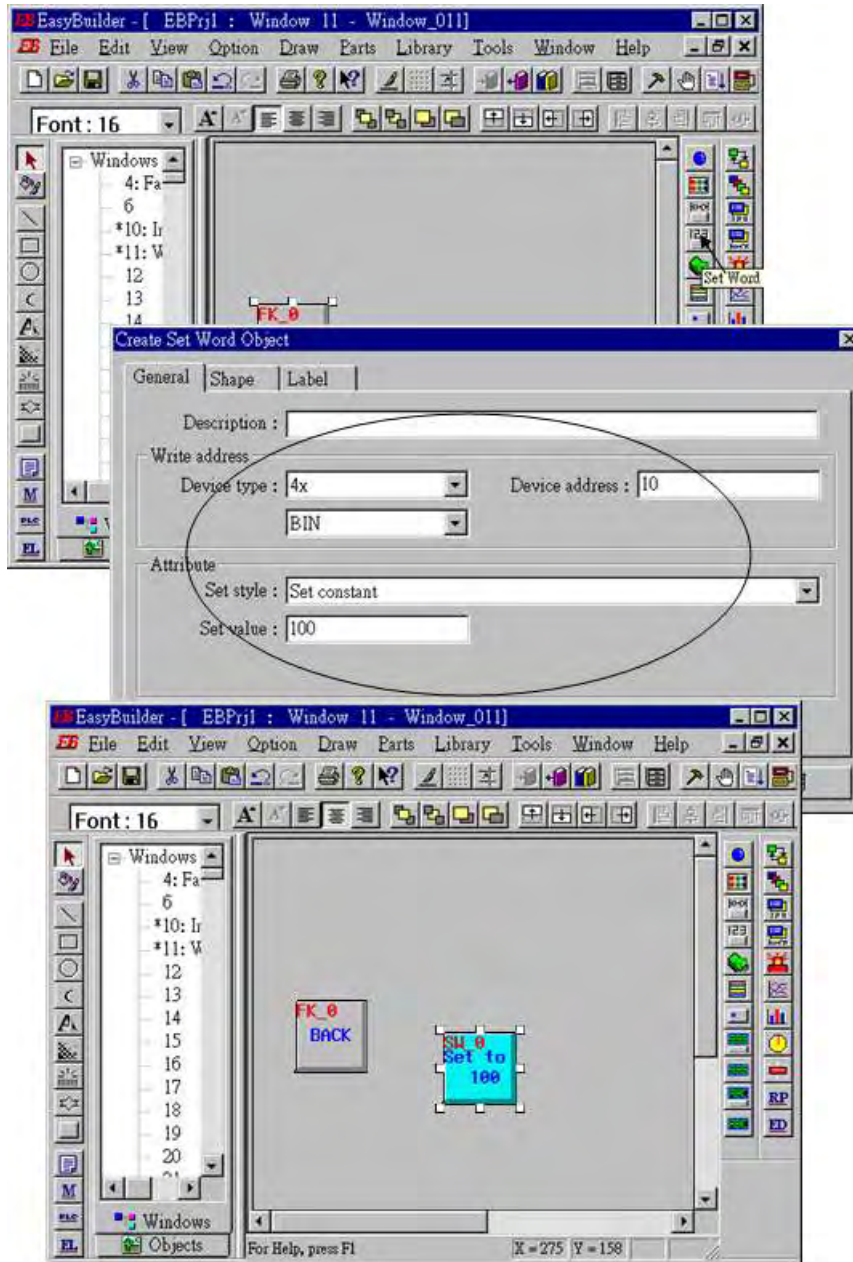
雙擊 “Windows_011”



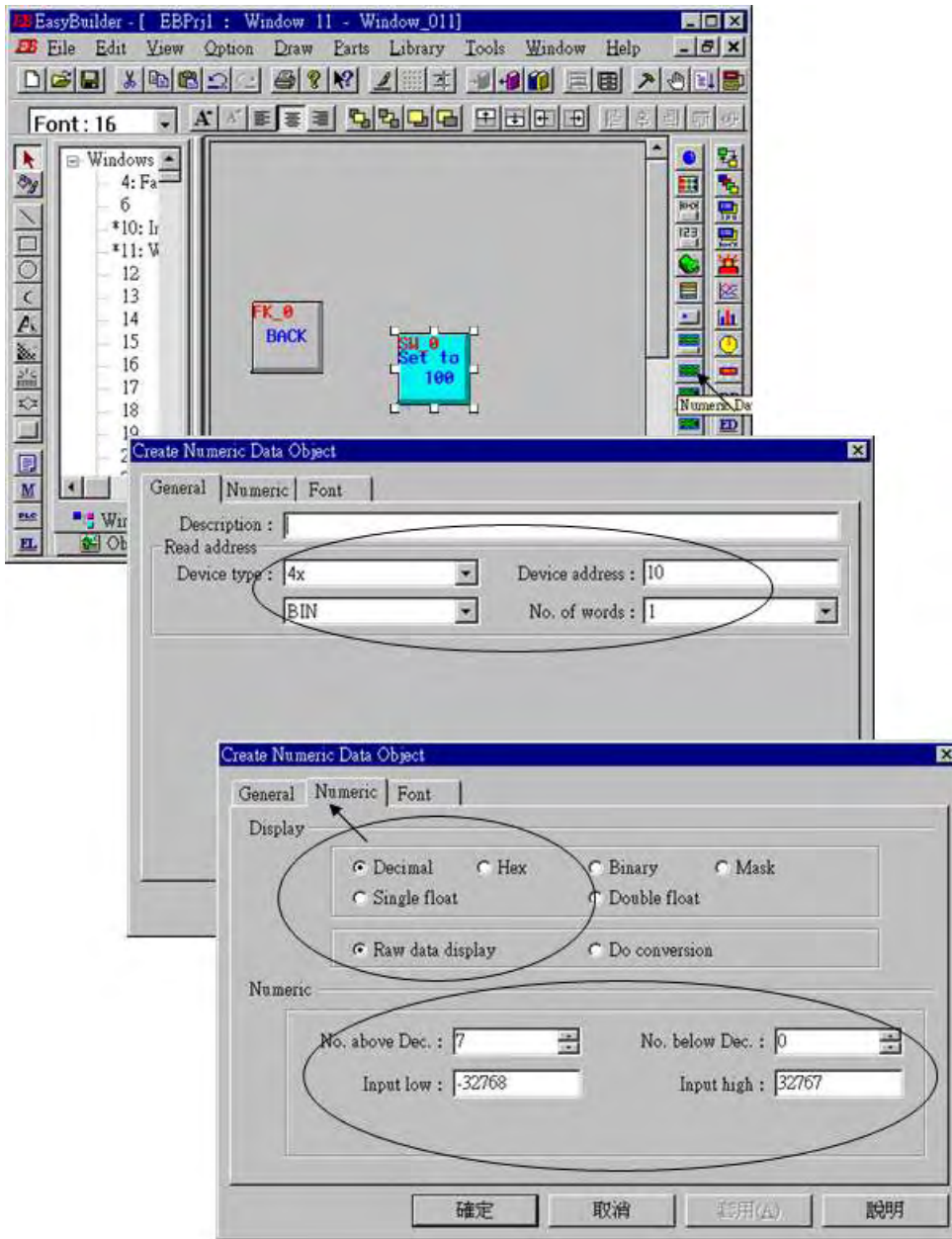
跟先前的方法一樣，按下[Function_key]→ “General”鍵，建立一個 ”change Windows”，”Windows No.”設定成 10，在”Label”選項上填入”BACK”。



按下 “Set Word” 鍵，將 “Device Type” 設成 “4x” (4x 代表 short 整數, 4L 代表 long 整數)，
“Device address” 設為 10，選取 “BIN” 選項，將 “Set style” 設為 “Set Constant”，“Set Value”
設為 100。接著選擇您所喜歡的圖型和將 “Label” 的文字填入 “Set to 100”。

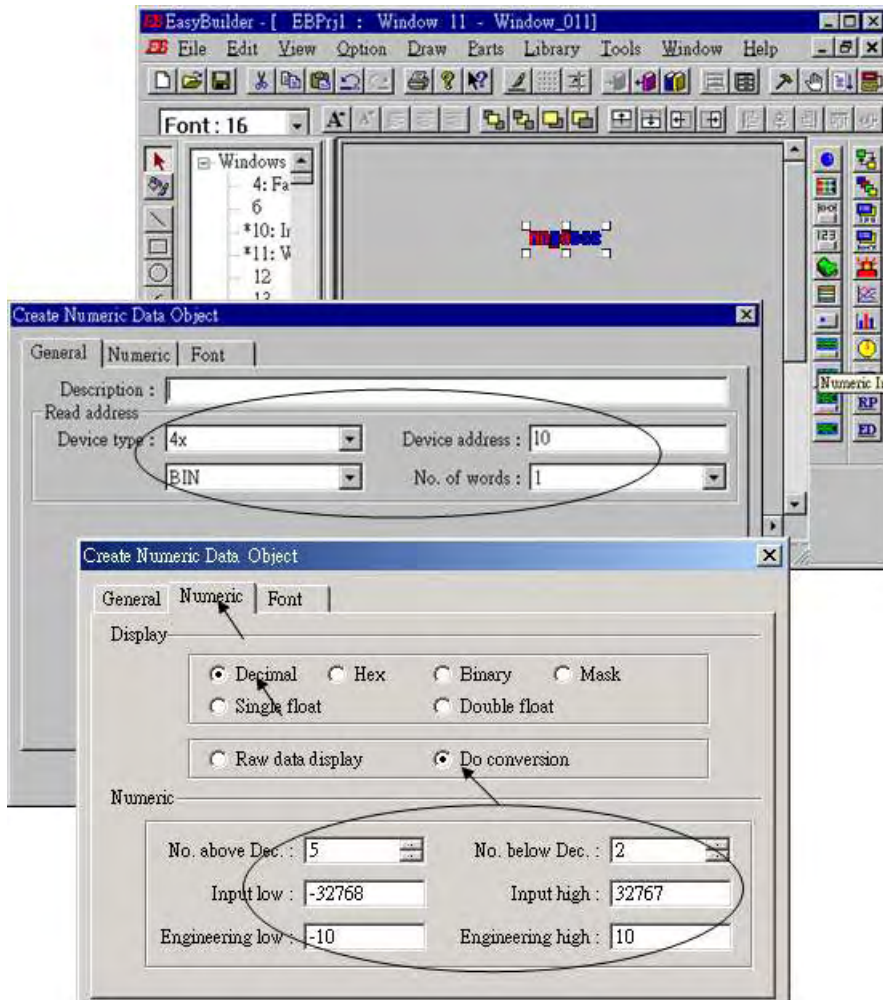


按下 “Numerical Data” 鍵，將 “Device Type” 設為 “4x” (4x 代表 short 整數, 4L 代表 long 整數)， “Device address” 設為 10，選擇 “BIN” 選項， “Number of words” 設為 1， “NO.above Dec” 設為 7， “No.below Decimal” 設為 0， “Input low” 設成 -32768， “Input high” 設成 32767，接著選擇您所喜歡的 Font。

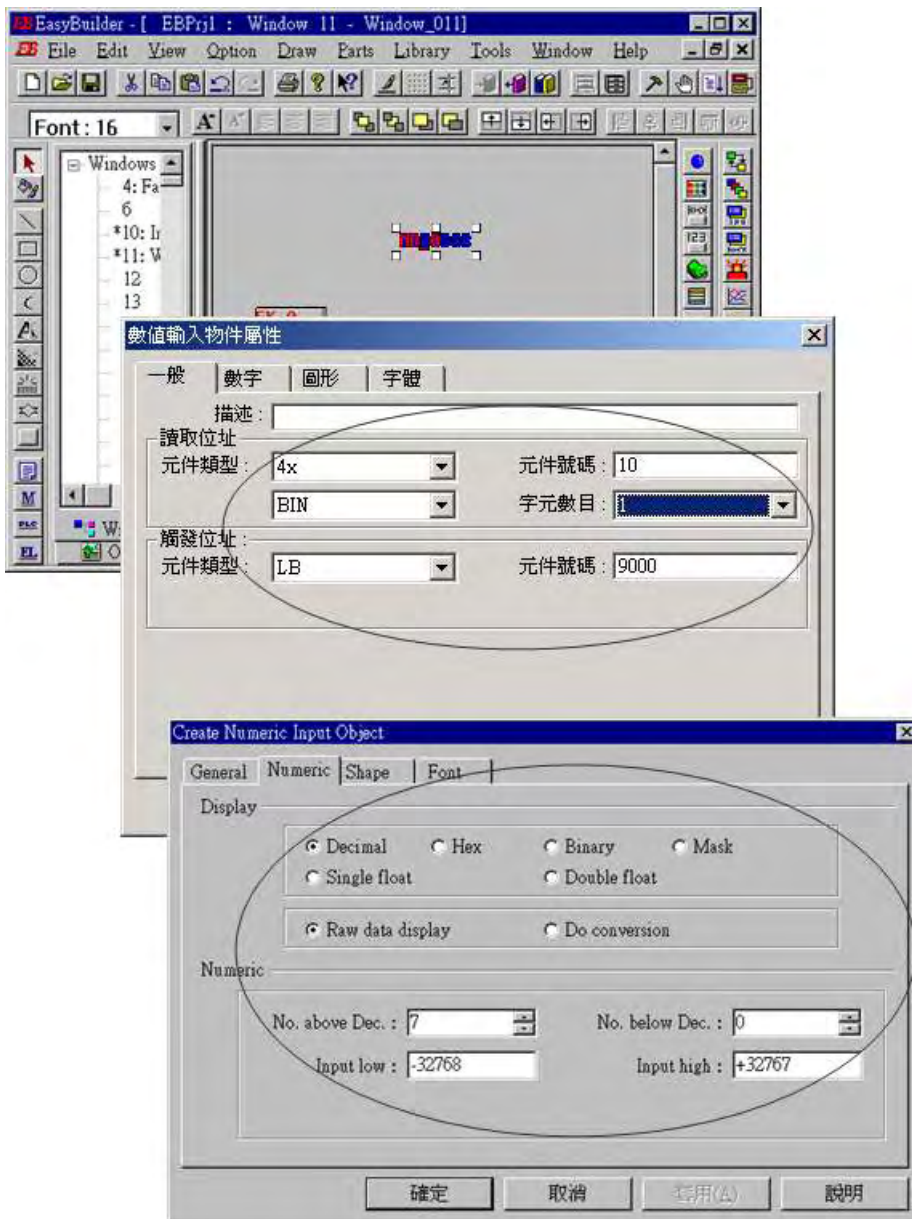


現在要來新增另一個有轉換功能 (conversion) 的 “Numerical Data”。

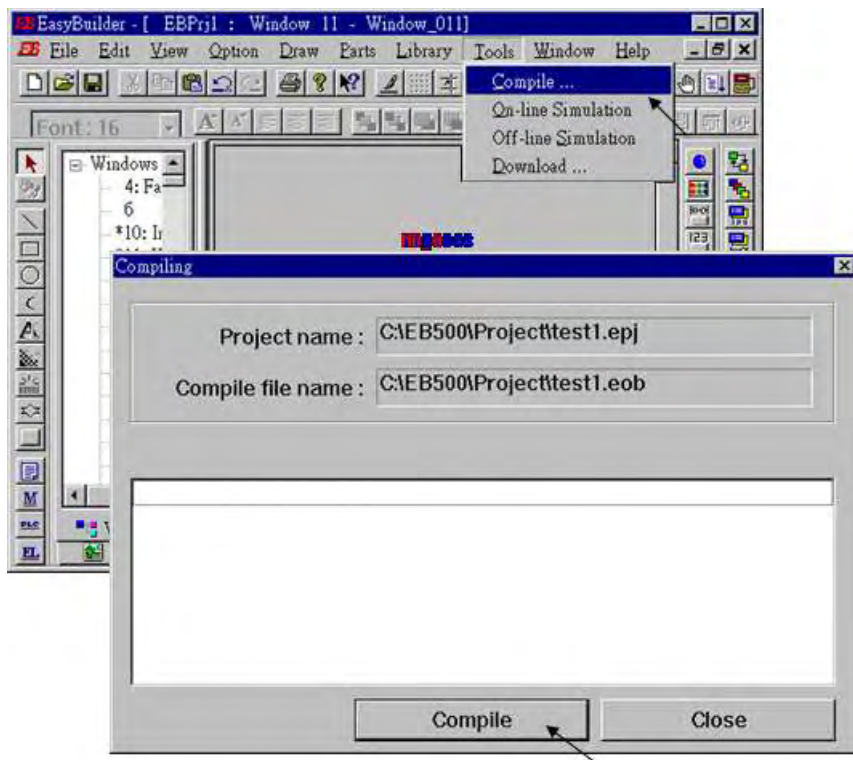
按下 “Numerical Data” 鍵，將 “Device Type” 設為 “4x”，“Device address” 設為 10，選擇 “BIN” 選項，“Number of words” 設為 1，選取 “Do conversion”，“NO.above Dec” 設為 5，“No.below Decimal” 設為 2，“Input low” 設成 -32768，“Input high” 設成 32767，“Engineering low” 設成 -10，“Engineering high” 設成 +10 (轉換 [-32768,+32767] 為 0 [-10,+10])，接著選擇您所喜歡的 Font。



按下 “Numerical Input” 鍵，將 “Device Type” 設為 “4x”，“Device address” 設為 10，選擇 “BIN” 選項，“Number of words” 設為 1，“Trigger Device Type” 設為 “LB”，“Trigger Device address” 設為 “9000”，“NO. above Dec” 設為 7，“No. below Decimal” 設為 0，“Input low” 設成 -32768，“Input high” 設成 32767，接著選擇您所喜歡的圖形。記得儲存

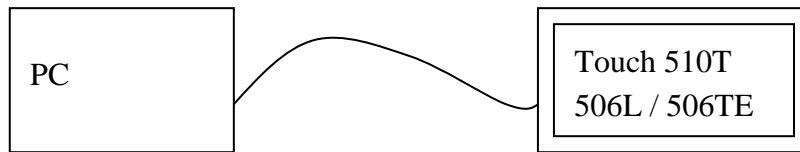


按下[Tools]→“Compile ...”來編譯您所寫的專案畫面。



按下[Start]→[Program]→[EasyBuilder]→“EasyManager”鍵，裡面有個“DownLoad”鍵可用來將專案下載到 Touch 510 中。選擇適當的 COM 埠，通訊速率設為 115200。

透過 RS232 傳輸線，您可以將 PC 和 Touch 510 連接(參考 4.4 節一開始的那頁)。

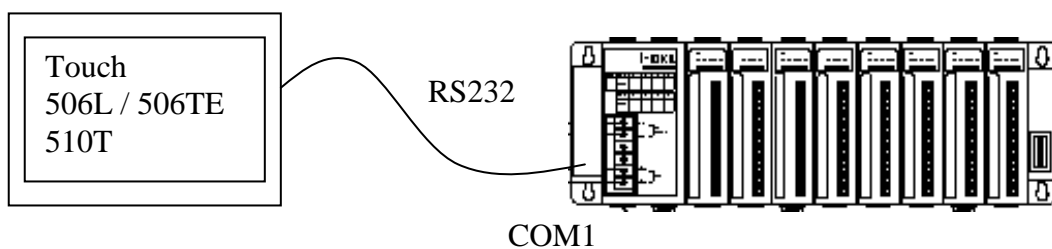


按下“Jump To RDS”，Touch 510 會出現等待專案下載的畫面，按下“DownLoad”即可開始下載人機界面圖示到 Touch 510。



若下載完成，您可以選擇“Jump To Application”或是將 Touch 510 重新開機，接著透過 RS232 傳輸線將 Touch 510 和 ISaGRAF 控制器連接。(參考 4.4 節一開始的那頁)

現在您可以在 Touch 510 上觸碰你所開發的圖示來測試 Touch 510 和控制器之間的資料傳輸。



4.5: 經由 Modbus 通訊協定讀/寫 Word 及 Integer 陣列

以下的 function 可在 ISaGRAF 程式內用來讀/寫 word 及 integer 陣列. 更詳細的 function 說明, 請參閱附錄 A.4.

ARY_N_R	從 integr (4 個 byte,有正負號) 陣列讀取一個 integer
ARY_N_W	寫一個 integer(4 個 byte,有正負號) 到 integer 陣列內
ARY_W_R	從 word (2 個 byte,有正負號) 陣列讀取一個 word
ARY_W_W	寫一個 word(2 個 byte,有正負號) 到 word 陣列內

在 I-8xx7, I-7188EG /XG, μ PAC-7186EG, iP-8xx7, VP-2117 內的 word 及 integer 陣列占用相同的記憶區, 請小心安排使用. 在 PC 上執行的軟體(如 HMI, OPC server, ...) 可以經由 Modbus 通訊協定來讀/寫這些陣列. I-8xx7, I-7188EG /XG, μ PAC-7186EG, iP-8xx7,VP-2117 內可以使用的 **network address** 編號為 **5001** 到 **8072**, 而 WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 則為 **10,001** 到 **19,216** 其對照表如下.

I-8xx7, I-7188EG /XG, μ PAC-7186EG, iP-8xx7, VP-2117:

Network Address (10 進位)	Word 陣列	Integer 陣列
5001	(1,1)	(1,1)
5002	(1,2)	
5003	(1,3)	(1,2)
5004	(1,4)	
...
...	...	
8071	(12,255)	(6,256)
8072	(12,256)	

WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7:

Network Address (10 進位)	Word 陣列	Integer 陣列
10001	(1,1)	(1,1)
10002	(1,2)	
10003	(1,3)	(1,2)
10004	(1,4)	
...
...	...	
19215	(36,255)	(18,256)
19216	(36,256)	

注意:

- Network address 1** 到 **4095** 號可被 I-8xx7, I-7188EG /XG, μ PAC-7186EG, iP-8xx7, VP-2117 的使用者自行定義, 而 WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 的 **Network address 1** 到 **8191** 號可被使用者自行定義,請參閱第 4.1 節
- 實際在傳送時用的是 **Modbus address**, 它的編號等於 **Network address** 減 1(請參閱第 5 章). 所以給 word 及 integer 陣列用的 **Modbus address** 為 5000 到 8071(I-8xx7, iP-8xx7, ...), 10000 到 19215(WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7).

第 5 章 Modbus 協定

Modbus 是一個功能非常強大且又很有彈性的通訊協定，它容許眾多的軟體程式和硬體裝置之間的溝通傳輸。ISaGRAF 控制器程式內的變數透過 Modbus 通訊協定，可以用來和其他裝置進行資料交換，當然這些變數的網路位址編號必須要先設定好(參考第 4.1 節)。

5.1: Modbus 協定格式: RTU Serial

ISaGRAF 控制系統內的變數若有設定網路位址編號時，則可以透過此網路位址編號與其他的軟體程式或人機界面裝置做資料的存取交換(請參考第四章)。關於如何連接 PC 和 ISaGRAF 控制器，請參考請參閱各自的“快速上手手冊”。

你必需撰寫 Modbus 通訊程式或使用市面上可買到的現成圖控軟體，來和 I-8xx7, I-7188EG/XG, μ PAC-7186EG, iP-8xx7, VP-2117, WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7 控制器溝通。這些 控制器 支援下面所示的 Modbus function。

Modbus function	Action
1	Read N bits (booleans)
2	Read N bits (booleans)
3	Read N words (signed short integers)
4	Read N words (signed short integers)
5	Write 1 bit (boolean)
6	Write 1 word (signed short integer)
15	Write N bits (booleans)
16	Write N words (signed short integers)

函式 1 和函式 3 都能用於讀取布林變數。如果使用函式 3，值會用一個 word 來儲存，變數為 TRUE 則值為 0xFFFF

函式 5 和函式 15 可以用於寫入布林變數。當使用函式 5 時，寫入 byte-vH 第 0 個 bit 為 1，將設定該 boolean 變數為 TRUE。例如，寫 vH=1 or 3 or 255 將會設 boolean 變數為 TRUE

函式 3 可以用來讀取類比變數。

函式 6 和函式 16 可以用來寫入類比變數。

函式 3 可以用來讀取長整數(Signed Long Integer)及浮點數(Float)，若是寫入長整數及浮點數，可用函式 16，在第 4.2 節有詳細的說明。

為了讓您了解 Modbus 通訊協定，下表提供了此節所使用的專有名詞，幫助您閱讀本節。

Slv	控制器的 NET-ID
Nbw	字組(Word)數目
Nbb	位元組(Byte)數目
Nbi	位元(Bit)數目
AddH	Modbus 網路位址編號的高位元組(High byte), 0 ~ 3F
AddL	Modbus 網路位址編號的低位元組(Low byte), 0 ~ FE
VH	字組值(Word)的高位元組(High byte)
VL	字元值(Word)的低位元組(Low byte)
V	位元組(Byte)值
CrcH	檢查碼的高位元組 (High byte of CRC-16)
CrcL	檢查碼的低位元組 (Low byte of CRC-16)

注意:

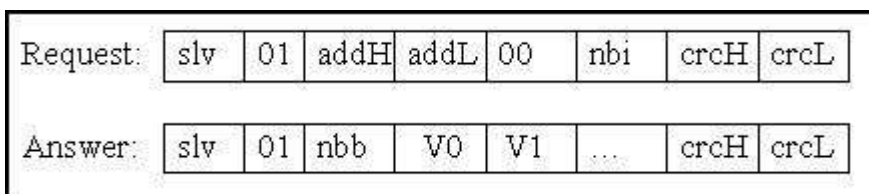
本章的封包在詢問和回答中均使用 16 進位表示法。

在本章所提到的 Modbus “位址編號” 等於 ISaGRAF 變數的 “網路位址編號” 減 1。

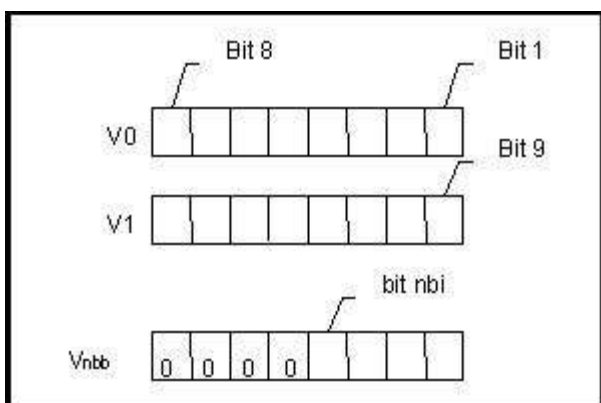
例如，若 Modbus 的網路位址編號為 0，則 ISaGRAF 變數網路位址編號為 1。若 Modbus 的網路位址編號為 FFE(4094)，則 ISaGRAF 變數網路位址編號為 FFF(4095)。

Function 1: 讀取 N 個位元 (Bit)

Function 1 用來讀取 N 個 Bits，由 Modbus 網路位址 addH/addL 開始。



V0、V1 為回傳的布林資料，使用下面的格式



Bit 1 相對應到 Modbus 網路位址 addH/addL 的 Boolean 值，Bit Nbi 相對應到 Modbus 網路位址 addH/addL+nbi-1 的 Boolean 值。如果 Boolean 變數的值是 True，則相對應的 Bit 將為 1。如果值是 False，則相對應的 Bit 為 0。

Function 2: 讀取 N 個位元(Bits)

Function 2 的格式和 Function 1 相同。

Function 3: 讀取 N 個字組(Words)

Function 3 由 Modbus 網路位址 addH/addL 開始，用來讀取 N 個 Signed 16-Bit integer。

Request:	slv	03	addH	addL	00	nbw	crcH	crcL
Answer:	slv	03	nbb	vH	vL	...	crcH	crcL

nbb 指的數目是由第 1 個 word 的高位元組(vH)算到最後 1 個 word 的低位元組(vL)所得到 byte 總數。

Function 3 注意事項

Integer 值可被 Function 3 讀取。在 Modbus 協定中的一個 word 是一個 16-bit 整術值(Signed Short integer)，然而在 ISaGRAF 內的一個 integer 變數是一個 32-bit 整術值，所以在 Modbus 協定內該值只傳回較低的 16 bits。如果使用者想讀取 I-8xx7 控制器內的 32-bit 的整數(Signed Long integer)，必需要設定適當的網路位址編號，如 4.2 節所述。

Function 4: 讀取 N 個字組(word)

Function 4 格式與 Function 3 相同。

Function 5: 寫入一個位元

Function 5 由 Modbus 網路位址 addH/addL 開始，寫入一個位元值到布林變數。

Request:	slv	05	addH	addL	V	0	crcH	crcL
Answer:	slv	05	addH	addL	V	0	crcH	crcL

寫入 0xFF 值到 V 將設定 Boolean 變數為 True。寫入一個零值到 V，Boolean 變數則會設為 False。

Function 6: 寫入一個字組(word)

Function 6 寫一個 word (16 bits)到 Modbus 網路位址 addH/addL 的 integer 變數

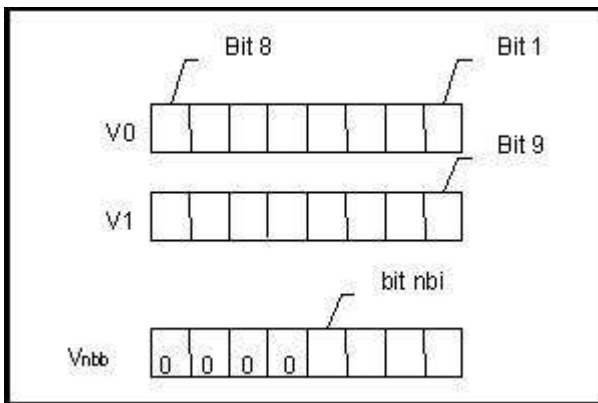
Request:	slv	06	addH	addL	vH	vL	crcH	crcL
Answer:	slv	06	addH	addL	vH	vL	crcH	crcL

Function 15: 寫入 N 個位元

Function 15 由 Modbus 網路位址 addH/addL 至 addH/addL+nbi-1 寫入數個 bits 到布林變數。Bytes 的總數(nbb)是 nbi bits 所占用的 bytes 總數， $nbb=(nbi+7)/8$ 。例如 nbi=1~8, nbb=1; nbi=9~16, nbb=2; nbi=17~24, nbb=3



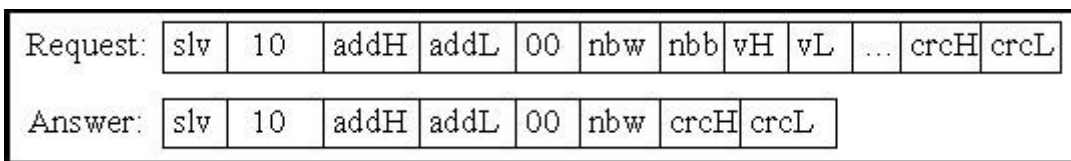
V0、V1 是布林資料，使用下面的格式



Bit1 相對應於 Modbus 網路位址 addH/addL 之 Boolean 變數值。Bit nbi 相對應於 Modbus 網路位址 addH/addL+nbi-1 之 Boolean 變數值。寫入 1 至一個 bit，會將相對之 Boolean 變數設為 True，寫入 0 到一個 bit，會將相對之 Boolean 變數設為 False

Function 16: 寫入 N 個字組(word)

Function 16 由 Modbus 網路位址 addH/addL 至 addH/addL+nbw-1 寫入數個字組(nbw) 到 integer 變數。Nbb 是 nbw Word 所占用的 byte 數， $nbb=2*nbw$ 。



Modbus 通訊格式的一些例子:

Function 1 : 由 Modbus address 0x1020 開始讀取 15 個 bits。控制器的 NET-ID 為 1。

Request:	01	01	10	20	00	0F	79	04
Answer:	01	01	02	00	12	39	F1	

在這個例子 function 1 傳回 2 bytes,其值為 0x0012，意指 ISaGRAF 變數網路位址 0x102A 與 0x102D 的布林值為 True (Modbus 網路位址 為 0x1029 與 0x102C)，其餘的為 False

Function 5 : 寫一個 bit 到 Boolean 變數，Modbus 網路位址為 0x0006。NET-ID 為 1,值被寫入為 0xFF。

Request:	01	05	00	06	FF	00	6C	3B
Answer:	01	05	00	06	FF	00	6C	3B

函式 5 的布林變數值會被設定為 “TRUE”

Function16 : 寫 2 個 words(4 bytes)到 integer 變數，Modbus 位址由 0x2100 開始。第一個 word 值寫入為 0x1234。第二個 word 寫入為 0x5678。NET-ID 為 1

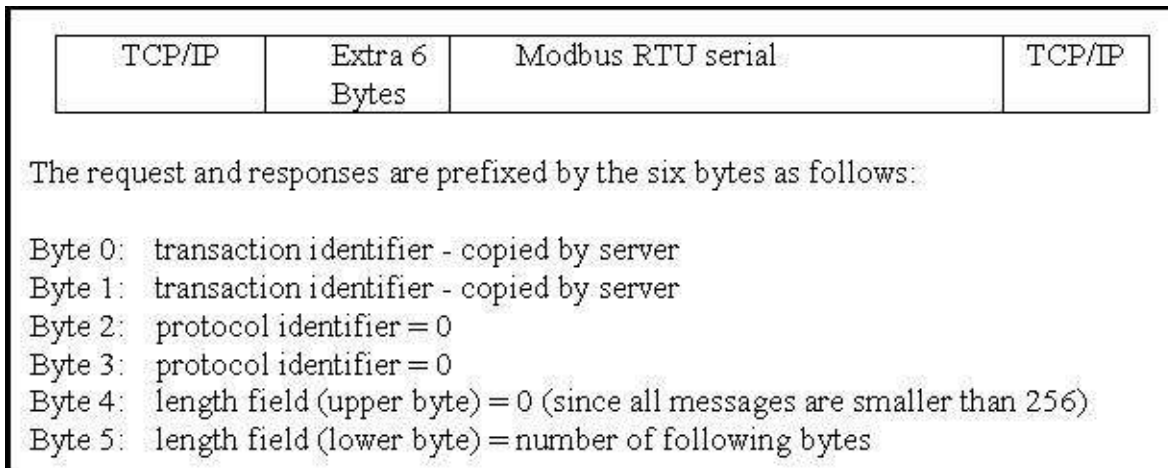
Request:	01	10	21	00	00	02	04	12	34	56	78	1C	CA
Answer:	01	10	21	00	00	02	4B	F4					

5.2: Modbus Protocol Format: TCP/IP

I-8437-80, I-8837-80, I-7188EG, μ PAC-7186EG, iP-8x47, VP-2117, WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7 等控制器的 Ethernet 埠都有支援 Modbus TCP slave 通訊協定

所有的通訊資料都透過 Port No = 502 送出

在 Modbus RTU series 協定之前，Modbus TCP/IP 協定增加 6 個額外的 bytes，這 6 個 bytes 和 Modbus RTU series 協定都被包在 TCP/IP 協定中



在 byte 號碼 6 之後，除了 CRC-16 在 Modbus TCP/IP 協定中是不需要的，其餘的 Modbus TCP/IP 協定與 Modbus RTU Serial 協定相同。

Modbus TCP/IP 協定例子

第一個 Modbus TCP/IP 協定的例子，是在 Modbus address 4 讀一個 word，從 NET-ID=9 回傳一個數值為 8，協定內容如下：

Request:	01	02	00	00	00	06	09	03	00	04	00	01
Response:	01	02	00	00	00	05	09	03	02	00	08	

第二個 Modbus TCP/IP 協定的例子，從 NET-ID=7，Modbus address 2 開始讀 8 個 bits，傳回值為 0x49(bit field: 01001001)如下：

Request:	03	29	00	00	00	06	07	01	00	02	00	08
Response:	03	29	00	00	00	04	07	01	01	49		

5.3: CRC-16 演算邏輯

下面的 C 語言演算法，只適用於 Modbus RTU Serial，這個 CRC-16 程式提供一個加總確定，用於確認資訊透過 Modbus RTU Serial 協定傳送的正確性

在通訊資料的程式之初，需先呼叫 `crc_init()` 一次，初始加總檢核表(checksum table)。然後無論何時當你想計算 checksum 時，呼叫 `crc_make()`

```
#define POLY_CRC16 0xA001
static BYTE TABLE1[256];
static BYTE TABLE2[256];

void crc_init(void) /* set crc table */
{
    WORD mask,bit,crc,mem;
    for(mask=0;mask<0x100;mask++)
    {
        crc=mask;
        for(bit=0;bit<8;bit++)
        {
            mem=crc & 0x0001;
            crc/=2;
            if(mem!=0) crc ^= POLY_CRC16;
        }
        TABLE2[mask]=crc & 0xff;
        TABLE1[mask]=crc >> 8;
    }
}

void crc_make(WORD size, BYTE *buff, BYTE *hi, BYTE *lo) /* calculate crc */
{
    BYTE car,i;
    BYTE crc[2];
    crc[0]=0xff;
    crc[1]=0xff;
    for(i=0;i<size;i++)
    {
        car = buff[i];
        car ^= crc[0];
        crc[0]=crc[1] ^ TABLE2[car];
        crc[1]=TABLE1[car];
    }
    *hi=crc[0];
    *lo=crc[1];
}
```

第 6 章 連接 I-7000 和 I-87K 遠程 I/O 模塊

在 1.5 節內有詳細的接線說明如何使 ISaGRAF 控制器 與 I-7000 和 I-87xx 系列模組相連接。

1. 請不要選 **I-87017** 與 **I-87017C**，改選用適合工業環境用的 **I-87017R** 與 **I-87017RC**。
2. 請不要選 **I-87018**，改選用適合工業環境用的 **I-87018Z** (I-87018z 有 10 個 Channel，每個 Channel 的量測準確度較 I-87018 與 I-87018R 與 I-87019R 為高，且每個 Channel 可設定使用不同的輸入型態，比如可使用第 1 到 4 ch. 來量測 4 ~ 20mA，第 5 到 8 ch. 來量測 T/C K-Type，第 9 ch. 量測 +/- 2.5V，第 10 ch. 量測 T/C R-Type)。

3. 請不要選 **I-7018**，改選用適合工業環境用的 **I-7018Z** (i-7018Z 的優點與 i-87018z 相同)

I-7018z: http://www.icpdas.com/products/Remote_IO/i-7000/i-7018z-g.htm

I-87018z: http://www.icpdas.com/products/Remote_IO/i-87k/i-87018z.htm

關於使用 i-7018z 的範例程式，請參考第 11.3.9 節。

6.1: 設定 I-7000 及 I-87xx 模組

注意:

A. I-7000 及 I-87xxxW 模組的型態若是類比輸入，則他們的資料格式必須設定成 “2’s complement”。例如: I-7005, I-7013, I-7015, I-7016, I-7017, I-7017R, I-7018, I-7018R, I-7019, I-7019R, I-7033, I-87005W, I-87013W, I-87015W, I-87015PW, I-87016W, I-87017W, I-87017RCW, I-87017ZW, I-87017DW, I-87018W, I-87018RW, I-87018ZW, I-87019RW, I-87019ZW, ... 等類比輸入模組。

B. I-7000 及 I-87xxxW 模組的型態若是類比輸出，則他們的資料格式必須設定成 “Engineer Unit”。

例如: I-7021, I-7022, I-7024, I-87022W, I-87024W, 及 I-87026W 等類比輸出模組。

控制器在連結 I-7000 及 I-87K 遠程 I/O 模組前，必需先以 DCON Utility 設定每一個 I-7000 及 I-87K 遠程模組的 站號(必須是唯一的 NET-ID)，並將控制器與遠程 I/O 模組設定相同的通訊速率。

DCON Utility 是一個方便好用的軟體工具，可以協助網路搜尋、設定與測試 I/O 模組。

DCON Utility 軟體程式及其使用手冊 可由下列網址取得

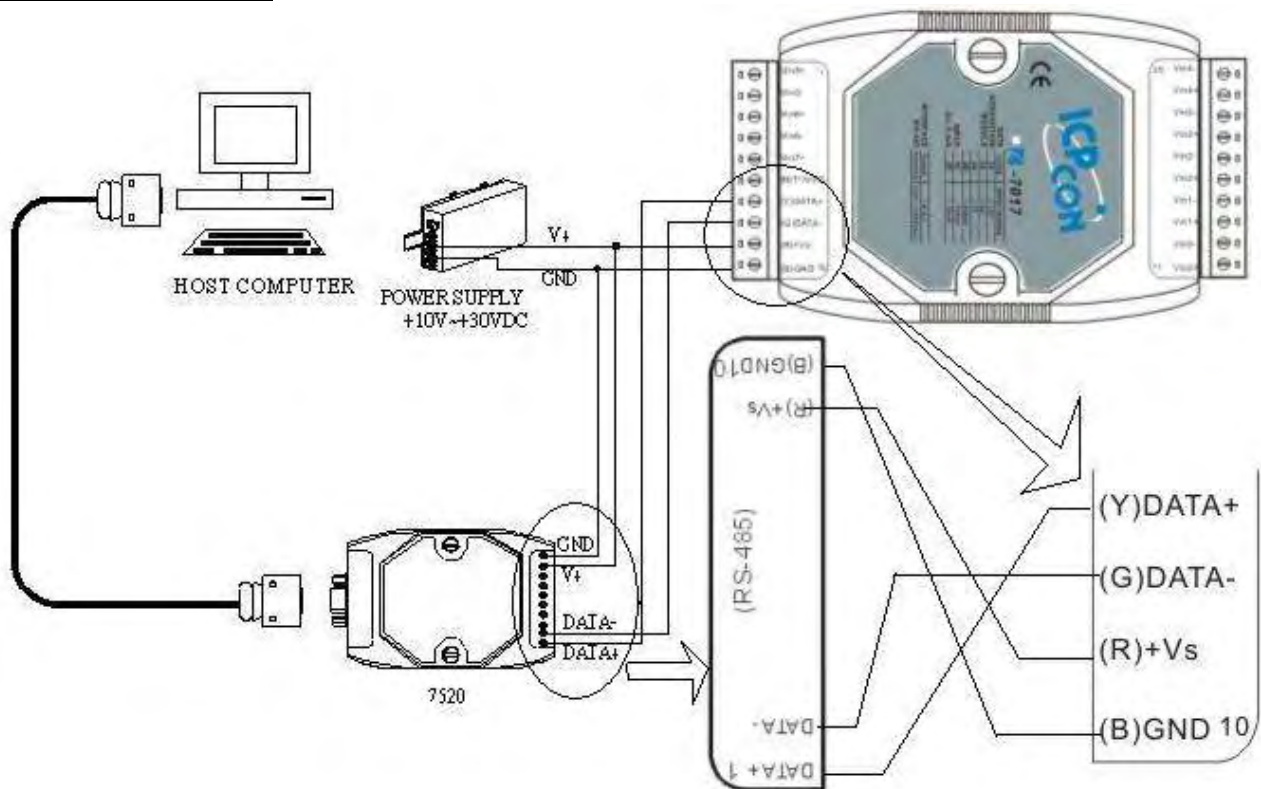
ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/。

注意：

1. 需確認硬體連接正確。
2. 一次只搜尋及設定規劃一個模組，以確保設定正確。
3. 設定前先將 I/O 模組的 Jumper 切換為 INIT 模式，然後重新啟動模組。
(有些較早期設計的 IO 模組是需要連接 IO 模組的 INIT* 及 GND 然後重新啟動模組)。

非常重要: 請在 ISaGRAF PAC 連接 I-7000 與 I-87K 模塊使用的 RS-485 串口的 D+ 與 D- 接腳中間接一個約 110 到 330 ohm 的電阻。(比如可以先試接 125 ohm，若還不行，再試其它)。這樣做可以確保 I-7000 與 I-87K 輸出模塊的 watchdog 功能可以正確動作。比如你可以先不接終端電阻，然後啟用 host watchdog 功能 (6.2 節，將 bus7000b 的 host_watchdog 參數設為 1)，之後若只拔開 I-7000 與 I-87K 輸出模塊的 DATA+ 接腳 (將 DATA- 維持在連接狀態)，您會發現 I-7000 與 I-87K 輸出模塊的 watchdog 無法正確動作。但若有接上 比如 125 ohm 電阻於 controller 的 RS-485 串口的 D+ 與 D- 接腳中間，當您拔掉 I-7000 與 I-87K 輸出模塊的 DATA+ 或 DATA- 接腳的任何一條，它的 watchdog 會在 timeout 時間抵達後正確 動作起來。

步驟 1: 硬體連接 : PC 連到 I-7000 做 設定動作



注意: 連接 I-7000 I/O 模組, 需準備一個 I-7520R (或 I-7520) RS232/RS485 轉換器 (或 其它 USB 轉 RS-485 的轉換器, 例如 tM-7561, I-7561). I-7520 / 7520R, tM-7561, I-7561 為 ICP DAS 高功能的 RS485 轉換器, 具備有 Self Turner 可自動切換 Baud rate.

http://www.icpdas.com/products/Industrial/communication_module/communication_list.htm .

不像某些廠牌的 RS232/RS485 轉換器, 會設定在只能用某個 Baud rate (這些 沒有 Self Turner 廠牌的 RS232/RS485 轉換器 要換 Baud rate 就得改設一次, 很不方便)

步驟 2: 將 I/O 模組設定為初始狀態

新的模組出廠時, 會將其設定為**預設狀態**(如下第一個表)以方便使用. 若您不知道模組的設定值, 請直接將 I/O 模組設定為**初始狀態** (如下第二個表).

*** 設定 I-7000 模組成為初始狀態, 請連接 INIT* 和 GND, 然後重新啟動模組電源. 該模組即可設定為初始狀態. (某些新的 I-7000 模塊背面有支持一個 Dip Switch, 這些模塊只需 切換 Switch 到 “INIT” 位置, 然後重新啟動模組電源, 不必去將 INIT* 和 GND 相連接)

*** 大部份的 I-87xxxW 卡片內有一個 Jumper 可以設成 Normal (運作狀態) 或 INIT (初始狀態).

*** 設定早期設計的 I-87K 模組成為初始狀態, 請切換 I-87K4/5/8/9 的相對應的 DIP 開關. 例如, 切換 DIP-2 到“ON”的位置, 然後重新啟動電源, 第二個插槽就會設為初始狀態. (若使用的擴充底板是 I-87K9 與 I-87K5, 請不要將 I-87K IO 插在 最左邊的那個 IO 插槽來做設定, 因為沒有對應的 DIP 可設, DIP-1 是對應到 I-87K9 / 87K5 左邊數過來的第 2 個 I/O 插槽. 若是 I-87K8 / 87K4 則沒有這個問題, 每個 I/O 插槽都有對應的 DIP switch) .

較新設計的高卡的 I-87K I/O, 比如 I-87019ZW, 在板卡上會有一個 Jumper 可撥成 INIT (初始狀態) 或 Normal (一般狀態), 此類的板卡 不需 去撥 I-87K4/5/8/9 Base 的 Dip 來設. 但切記 設定完後 要把該 I-87K IO 卡的 Jumper 撥回 Normal 位置 .

工廠出貨時的預設狀態表：

I/O 模組	i-7000	M-7000	87K 系列
站號	1	1	1
通訊速率	9600	9600	115200
Checksum	Disabled	Not defined	Disabled
通訊協定	DCON Protocol	Modbus Protocol	DCON Protocol

初始狀態表：



I/O 模組	7000 系列 (i-7000 及 M-7000)	87K 系列
站號 , 通訊速率	0 , 9600	0 , 115200
Checksum	Disabled	Disabled

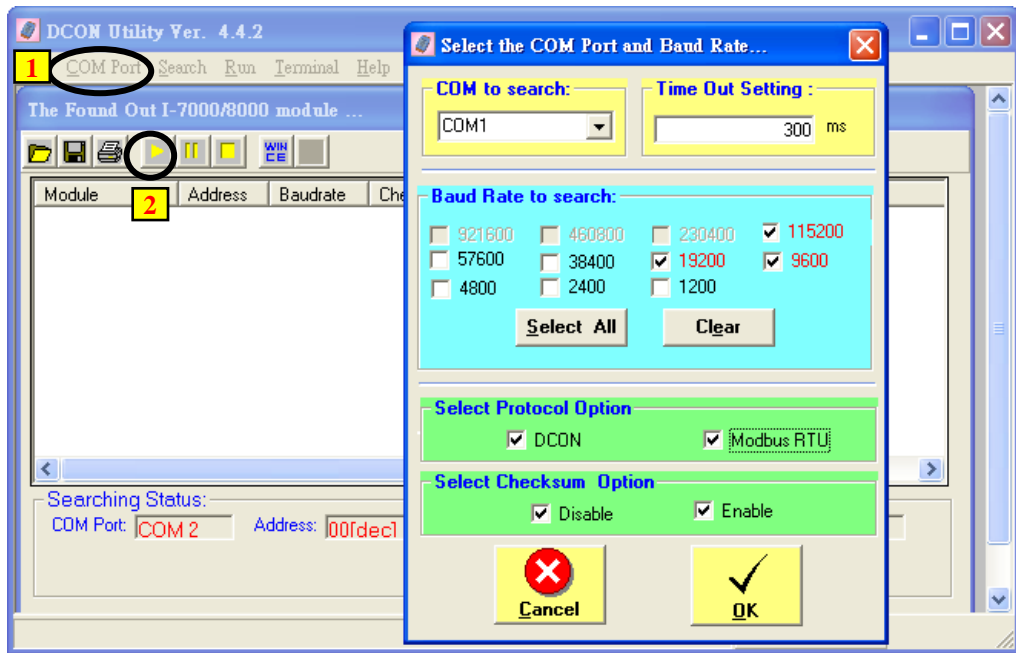
步驟 3: 選擇搜尋的 COM port 和通訊速率(baud rate)

從 “開始/所有程式/DAQPro/DCON Utility/” 啟動 DCON Utility.

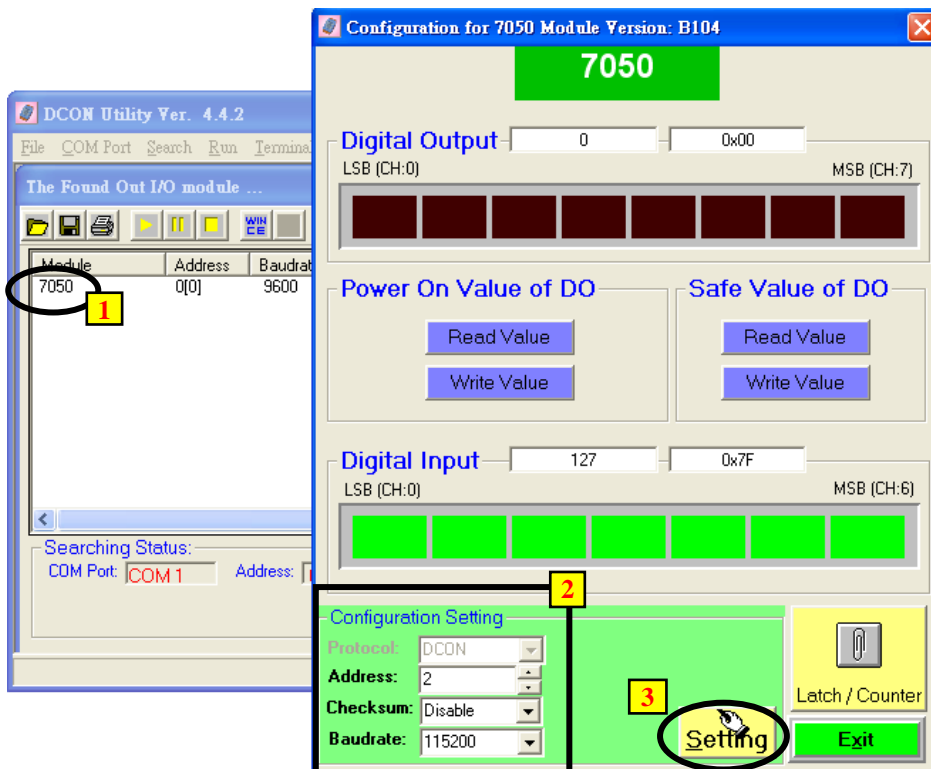


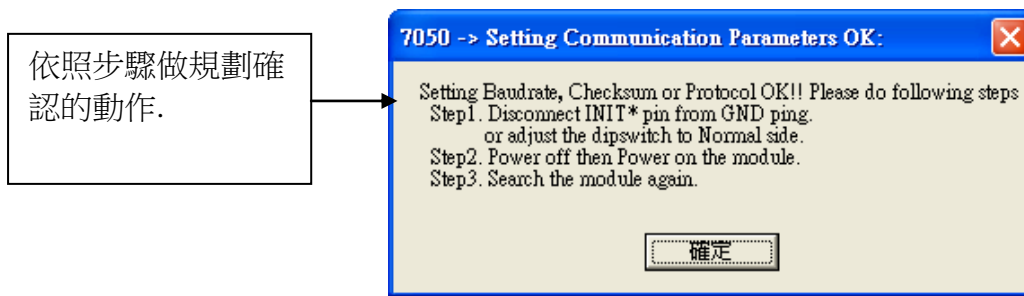
1. 點選 “COM Port” 功能表來選擇搜尋的 COM port 和 baud rate. 如果您不知道先前模組的設定, 可以選擇多個 baud rate, protocol 和 checksum, 搜尋網路時會花較多時間. 選擇完畢, 按 “OK” 鍵.

2. 點選  “Start Search” 按鈕開始搜尋模組。 找到模組請點  按鈕。



步驟 4: 點選找到的模組型號， 然後在出現的視窗中規劃新的設定





注意： 設定完成後，記得一定要移除 I-7000 的 INIT* 和 GND 的連接線。然後重新開啟電源。若是 I-87K I/O 模組，則需記得將相關的 Dip 開關切回 “OFF”，然後重新啟動電源。不然該 I/O 模塊會一直處於 INIT 狀態而無法使用。

關於 I-7000 和 I-87xxx 模組的重要訊息：

一台 I-8xx7, I-7188EG/XG, μ PAC-7186EG, iP-8xx7, VP-2117 控制器最多可以連接 64 個 I-7000 和 I-87xxx 模組(WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 則為 255 個)，但建議最多一個控制器只連接 40 個模組。每一個 I-7000 或 I-87xxx 模組必需擁有一個唯一的位址號碼，讓 ISaGRAF 控制器能夠與它們溝通。在 DCON Utility 中，“Checksum” 的選項通常設定為 “Disable”，且每一個 I-7000 及 I-87xx 模組的通訊速率必須與 控制器 設定的一樣。

I-7000 及 I-87xxx 模組的型態若是類比輸入，則他們的資料格式必須設定成 “2’s complement”。
例如: I-7005, I-7013, I-7015, I-7016, I-7017, I-7017R, I-7018, I-7018R, I-7019, I-7019R, I-7033, I-87005W, I-87013W, I-87015W, I-87015PW, I-87016W, I-87017W, I-87017RCW, I-87017ZW, I-87017DW, I-87018W, I-87018RW, I-87018ZW, I-87019RW, I-87019ZW, ... 等類比輸入模組。

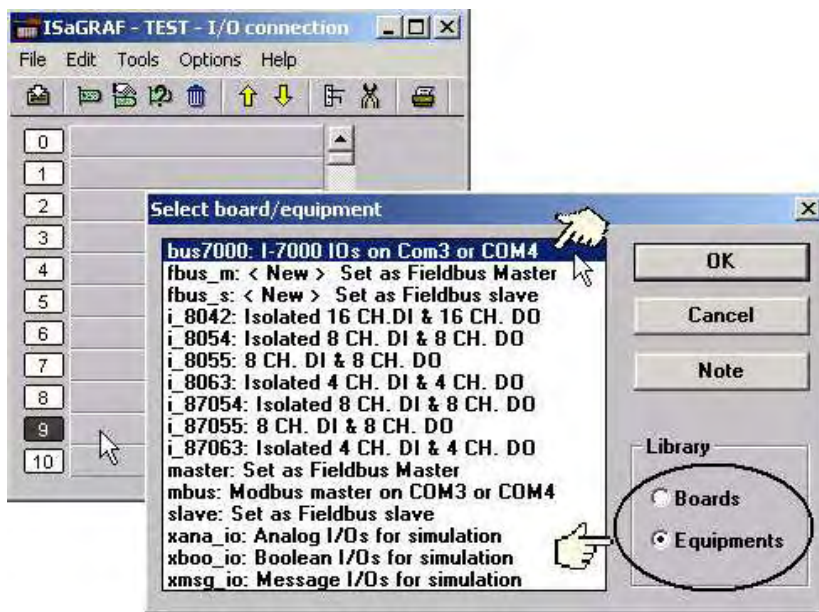
I-7000 及 I-87xx 模組的型態若是類比輸出，則他們的資料格式必須設定成 “Engineer Unit”。

例如: I-7021, I-7022, I-7024, I-87022W, I-87024W, 及 I-87026W 等類比輸出模組。

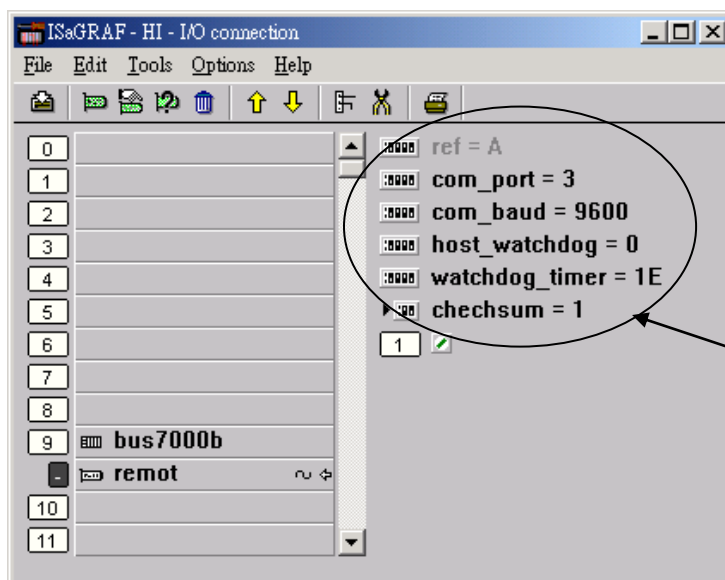
6.2: 開啟 “Bus7000b”

您必需透過連結 “ISaGRAF I/O Connection”內的 “Bus7000” (或 Bus7000b, 可設定 Checksum 是 Enable 或 Disable, 而 “Bus7000”只能用在 Checksum Disable 的場合), 才可以將 ISaGRAF 控制器和 I-7000 及 I-87xxx 模組作連結。“Bus7000b” 是一個虛擬模組, 在 “Select Board/Equipment” 視窗中, 您要選擇 “Equipment” 才會出現。

“Bus7000b”必須連接在 “ISaGRAF I/O Connect”視窗中編號 8 或 8 以上的插槽(插槽 0~7 均給真實的 I-8xxxW 與 I-87xxxW 等 I/O 卡片使用)。若您企圖連接兩個以上的 “Bus7000b” 到 ISaGRAF PAC 控制系統上, 只有一個 “Bus7000b”會產生作用。



下列顯示連結 “Bus7000b” 在插槽 9 上。



Checksum 設為 1 表示所要連接的 I-7000 & I-87xxx 都是設為 Checksum Enable, 若設為 0 則表示 Disable.

“**com_port**” 參數值可以設定成 I-8xx7, iP-8xx7: 3(Com 3)或4(Com 4), I-7188EG/XG, 7186EG: 2(Com 2)或3(Com 3), WP-8xx7, VP-25W7, VP-23W7: 2(Com2), XP-8xx7-CE6: 3 (COM3)。這個參數的指的是 控制器 連接到 I-7000 及 I-87xx 模組所使用的 COM 埠編號。

“**com_baud**” 參數指的是 “控制器和 I-7000 和 I-87xxx 模組相連的通訊速率。可選擇的值為 2400, 4800, 9600, 19200, 38400, 57600 及 115200 bps。您必需確定 控制器 和 I-7000 和 I-87xxx 模組是否設定為一樣的通訊速率，以讓它們之間能順利的溝通。

“**host_watchdog**” 參數的設定關係到 I-7000 和 I-87xxx 模組的 watchdog 函式的開啟或關閉。將 “host_watchdog” 參數設為 1 會開啟 “host_watchdog” 功能, 設為 0 則是關閉。

“**watchdog_timer**” 參數為啟動 “host_watchdog” 後所設定的 Time out 的時間。

“watchdog_timer” 的值是 16 進位的數字且單位為 0.1 秒，若您設定為 1E, “watchdog_timer” 即被設定為 3 秒，若你設定為 2A, “watchdog_timer” 即被設定為 4.2 秒。

若 控制器 和 I-7000 或 I-87xxx 模組通訊斷線超過 watchdog_timer 的時間時，則 host_watchdog 會讓 I-7000 及 I-87xxx 模組的 Output Channel 進入你 DCON utility 對它所設定的 安全值(通常 D/O 會是設成 False)。

在 “Bus7000: Remote” 虛擬板上有一個類比輸入點。這個類比輸入點主要的功能為傳回現在的通訊速率, 若值為 0 表示該 通訊串口開啟失敗。

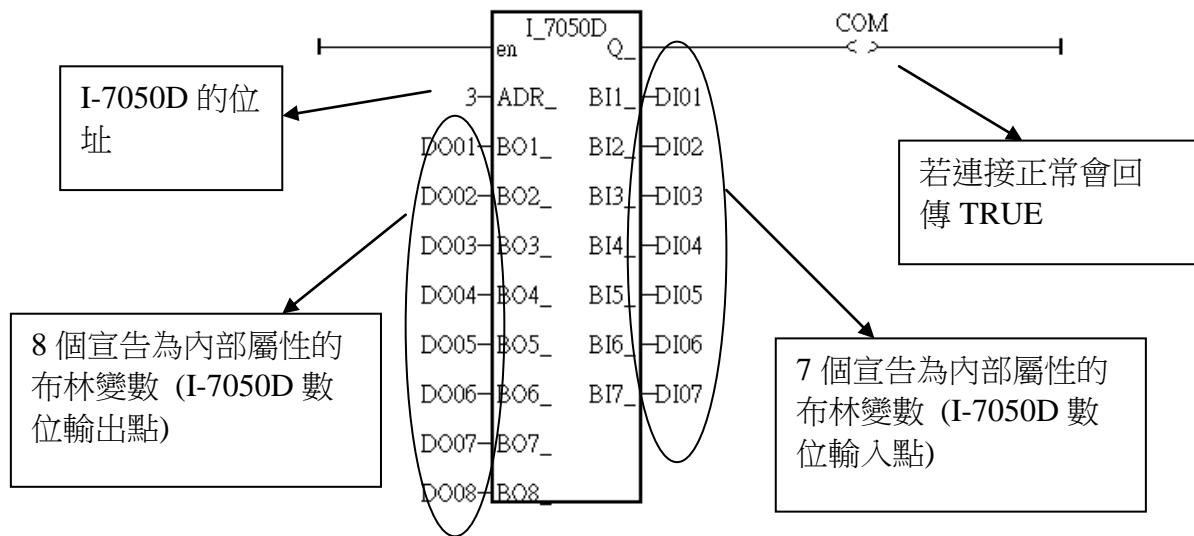
6.3: 編寫控制 I-7000 模組程式

6.3.1: 編寫 I_7xxx 及 I_87xxx 方塊程式

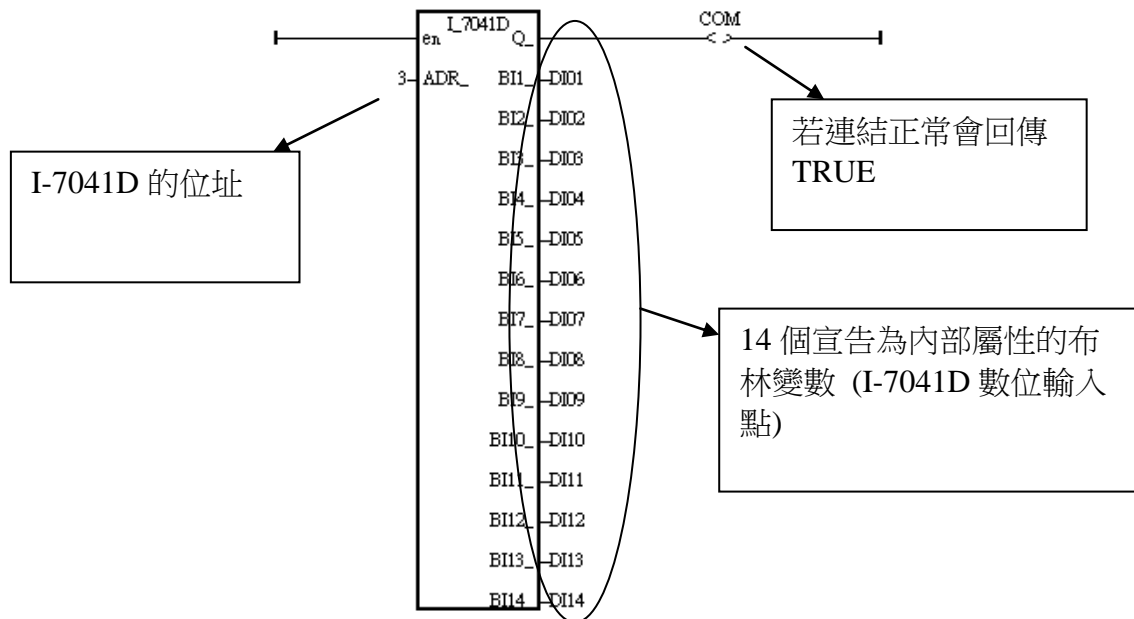
在您連結任何 I-7000 及 I-87xxx 模組到 ISaGRAF 控制器之前，您必須先開啟“Bus7000b”。一旦“Bus7000b”被開啟，你就可以在 控制器 程式內編寫程式來存取 I-7xxx 或 I-87xxx 上的 I/O 到對應的變數內，之後您可將那些變數使用在同個 ISaGRAF 專案的其他程式上。

注意： 您必需宣告所有連接在 I-7000 及 I-87xxx 的方塊的 變數 為 內部(internal) 屬性

範例 1: 編寫 I-7050D 方塊程式



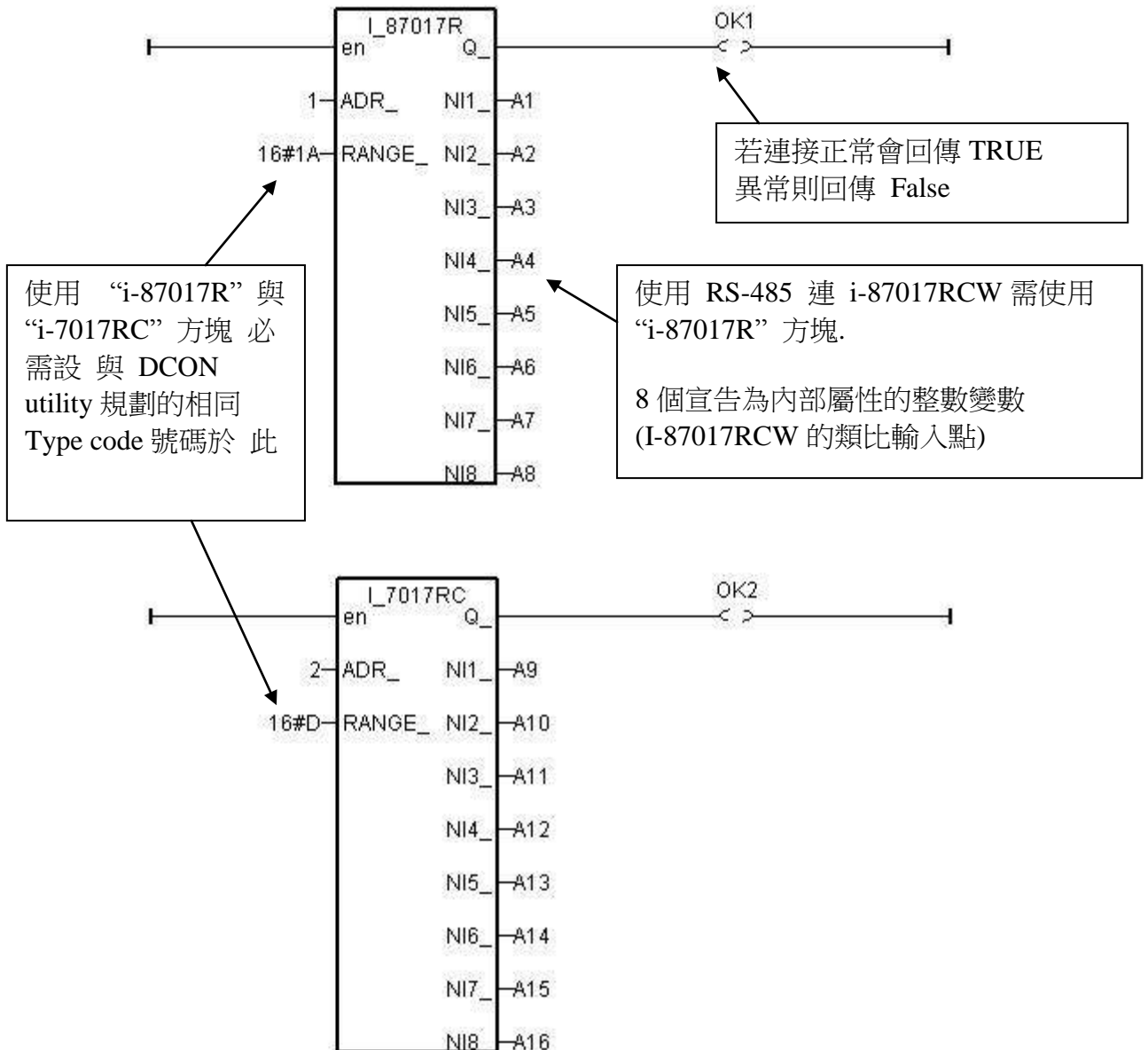
範例 2: 編寫 I-7041D 方塊程式



範例 3: 編寫 I-87017R 與 I-7017RC 方塊程式 (硬體為 I-87017RCW 與 I-7017RC 時使用)

I-87017RCW 與 I-7017RC 可用來量測 $\pm 20\text{ mA}$, $0 \sim 20\text{mA}$ 與 $4 \sim 20\text{mA}$ 的電流輸入訊號, 不需外接 125 ohm 電阻. 它的資料格式型態必須使用 DCON Utility 設成 2 的補數 (2's complement) 才能使用. (可參考 附錄 A.4 內 A4_20_to 的說明, 將輸入值轉換為工程值)

Range 識別號碼 (藉由 DCON Utility 設定)	物理量數值	I-7017RC /87017RCW 方塊的輸入點數值 (10 進位)		
		- 32768	0	+32767
7	4 ~ 20 mA		4 mA	20 mA
D	$\pm 20\text{mA}$	- 20mA	0 mA	20mA
1A	0 ~ 20 mA		0 mA	20 mA



使用 RS-485 遠程 IO 量測電流 4 ~ 20 mA 的重要訊息:

如果使用的 Sensor 為 4 到 20 mA 的量測型態, 可以考慮將 Analog input 模塊的 Range type 設為 [D]: +/- 20 mA 或 [1A]: 0~ 20 mA 的 Type. (此時設 "[7]: 4 to 20 mA" type 比較不好)

原因為:

如果的 Range type 為 [7]: 4 到 20 mA type, Analog Input 值若為 0 或接近 0 可以表示 Sensor 輸入為 4 mA, 但也有可能是 Sensor 斷線. 因此光看 Analog Input 數值, 區分不出這 2 者.

然而若是將 Range type 規劃為 [D]: +/- 20 mA 或 [1A]: 0 ~ 20 mA type, 當值接近 0 時, 只有可能是 Sensor 斷線. 而 Sensor 輸入的 4 到 20mA, 值會是 (6553 到 32767), 4mA 時值是在 6553 附近, 不是在 0 附近.

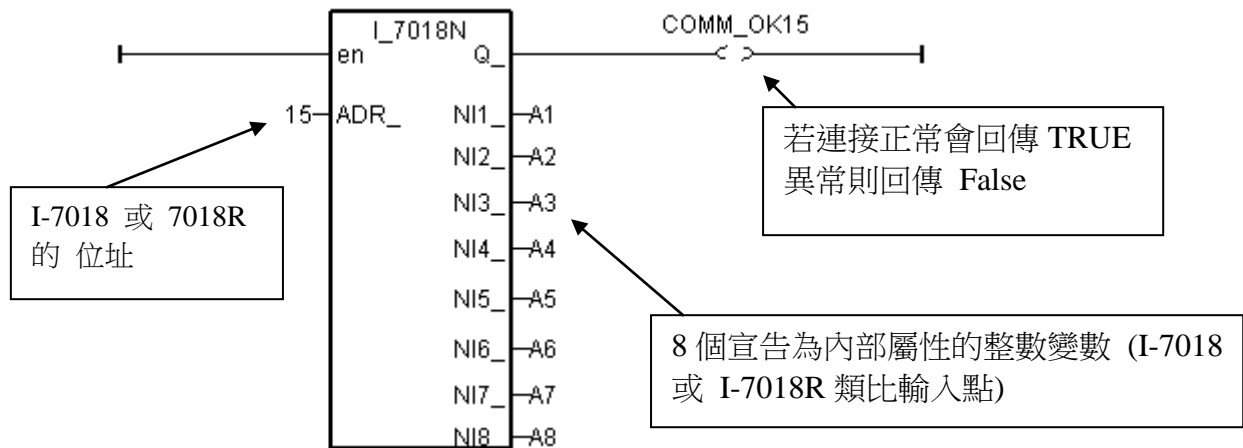
(當然 Analog input 模塊 與 控制器間的 RS-485 連線需正常, 以上本例的變數 OK1 與 OK2 可用來判斷遠程 I/O 模塊的 RS-485 連線是否正常, 值若為 False 表通訊有問題, 你可在程式內作適當處理).

因此有需要分辨出 Sensor (4 到 mA) 是否正常時, 最好是規劃 為 [D]: +/- 20 mA 或 [1A]: 0 ~ 20 mA type. 如此你的程式內就可比如說當 A1 ~ A16 輸入值 小於 5000 或 小於 4000, 就可判定為 Sensor 斷線 或 Sensor 異常. (如果你是設成 [7]: 4 ~ 20 mA 的 Type, 就無法分辨, 到底是 4 mA 還是 Sensor 異常)

範例 4: 編寫 I-7018 方塊程式 (使用 I_7018n 方塊)

(硬體請改選用 I-7018z 較佳, 第 11.3.9 節有使用 I-7018z 的範例程式)

I-7018 or I-7018R 的資料格式型態必須使用”DCON Utility” 設成 2 的補數 (2’s complement) 才能使用. 請用 “I_7018n” 方塊 而非 “I_7018” 方塊 (“I_7018n” 方塊一次問完 8 個 Channel, 而 “I_7018” 方塊 要問 8 次才問完 8 個 Channel)



編寫遠程 I-7000 或 I-87K I/O 的方法都與上方類似, 要注意的是 :

若 **I-7000** 及 **I-87xxx** 遠程 IO 模組的型態是類比輸入, 則他們的資料格式必須設定成 “2’s complement”. 例如: I-7005, I-7013, I-7015, I-7016, I-7017, I-7017R, I-7018, I-7018R, I-7019, I-7019R, I-7033, I-87005W, I-87013W, I-87015W, I-87015PW, I-87016W, I-87017W, I-87017RCW, I-87017ZW, I-87017DW, I-87018W, I-87018RW, I-87018ZW, I-87019RW, I-87019ZW, ... 等類比輸入模組.

若 **I-7000** 及 **I-87xx** 模組的型態是類比輸出, 則他們的資料格式必須設定成 “**Engineer Unit**”.

例如: I-7021, I-7022, I-7024, I-87022W, I-87024W, 及 I-87026W 等類比輸出模組.

下表為 I-7017, 7017R, 87017W, 87017RW 類比訊號的物理量轉換成整數值的轉換表 (線性關係). (這些模塊量測 D: $\pm 20\text{mA}$ 時, 必需外接 125 ohm 於各 Channel 的正 / 負端, 若不想使用 外接電阻來量測電流值, 請選用 I-7017RC 或 I-87017RCW 或 I-87017ZW 或 I-87019ZW)

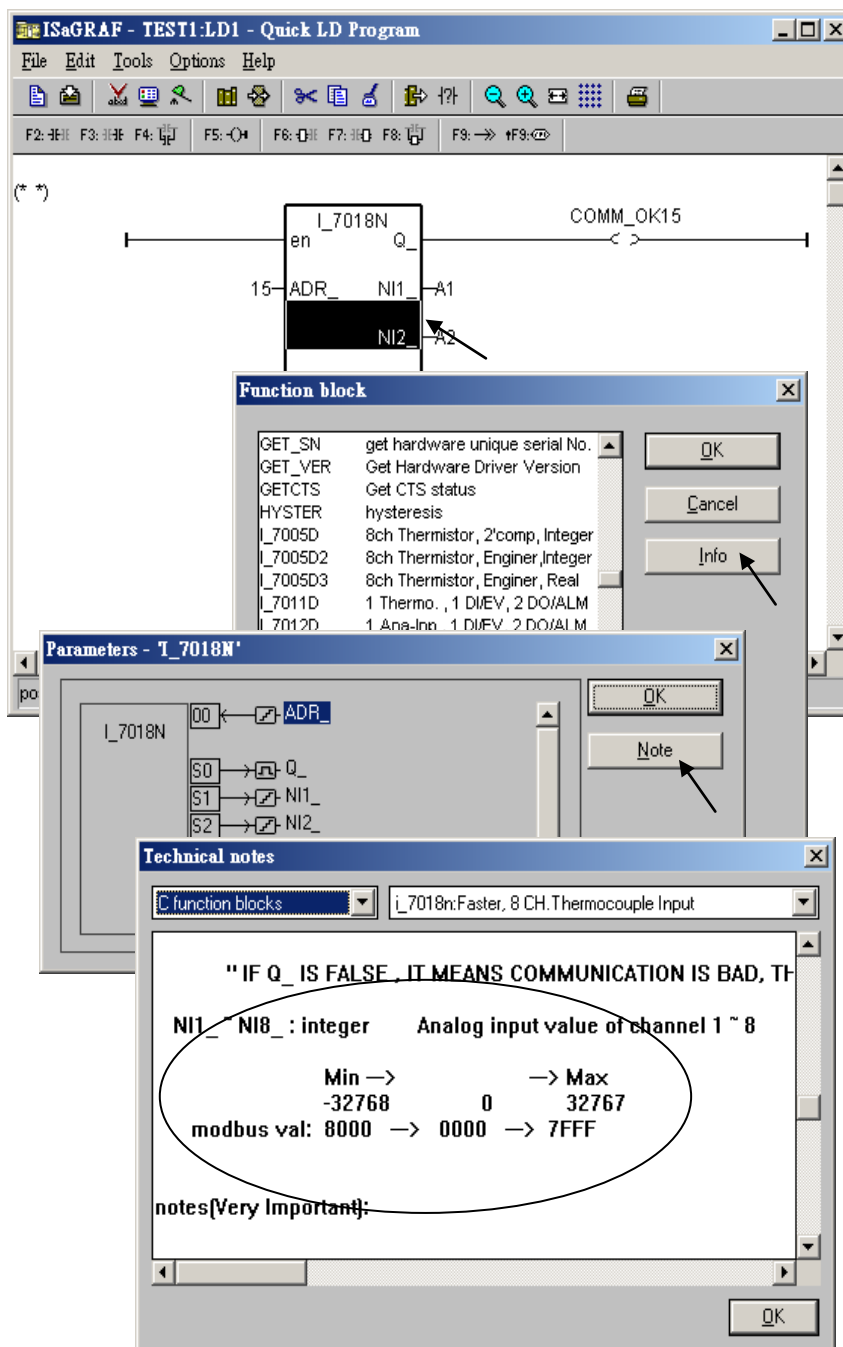
Range 識別號碼 (藉由 DCON Utility 設定)	物理量數值	I-7017/87017 方塊的輸入點數值(10 進位)		
		-32768	0	+32767
8	$\pm 10\text{V}$	- 10V	0V	+ 10V
9	$\pm 5\text{V}$	- 5V	0V	+ 5V
A	$\pm 1\text{V}$	- 1V	0V	+ 1V
B	$\pm 500\text{mV}$	- 500mV	0mV	+ 500mV
C	$\pm 150\text{mV}$	- 150mV	0mV	+ 150mV
D	$\pm 20\text{mA}$	- 20mA	0mA	+ 20mA

若硬體是選用 I-7017RC 或 I-87017RCW 則請看下表

Range 識別號碼 (藉由 DCON Utility 設定)	物理量數值	I-7017RC /87017RCW 方塊的輸入點數值 (10 進位)		
		- 32768	0	+32767
7	4 ~ 20 mA		4 mA	20 mA
D	± 20mA	- 20mA	0 mA	20mA
1A	0 ~ 20 mA		0 mA	20 mA

其它各個 IO 模塊的轉換表，請查看 ISaGRAF 的 On-Line Help 如下圖所示 及 本手冊的附錄 D

先在方塊內用滑鼠雙擊，然後點選 “Info”，最後點選 “Note”



6.3.2：直接取得 遠程 I-7000 及 I-87xxxW 溫度輸入 I/O 的溫度值

ICP DAS 提供很多遠程溫度輸入模塊，如下。

有 ”感測器斷線偵測功能” 的 RS-485 遠程溫度模塊型號：

Thermocouple : I-87018zW, 87018RW, 87019RW, 87019zW, 7018R, 7018BL, 7019, 7019R, 7018Z

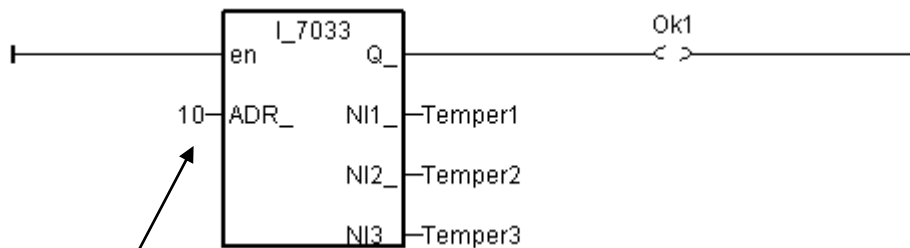
RTD 型式: I-87013W, 87015W, I-87015PW, 7013, 7015, 7033

Thermister 型式: I-87005W, 7005

無 ”感測器斷線偵測功能” 的模塊型號：

Thermocouple 型式: I-87018, 7018, 7018P

I-7000 方塊中的 “ADR_” 參數可以是 “標準設定值” 或 “特別設定值”。比如將 ”I_7033” 的 “ADR_” 參數 設成 1 到 255 表示使用“標準設定值”，此 1 到 255 的值表示 I-7033 遠程溫度模塊的位址。取得的 溫度輸入值 會是一個 在 -32768 到 + 32767 區間的整數。必需參考附錄 D 來求得真正的溫度值。此溫度轉換會跟該 遠程溫度模塊的 “Type code” 設定有關 (用 DCON utility 在初始設定時設的)。(若值是等於 -32768 或 +32767，對有 ”感測器斷線偵測功能” 的 I/O 模塊而言，指的是 ”感測器斷線了”)



ADR_ 為 10 (TT=00, RR=00, AA=0A ←16 進位.) 表示使用 “標準設定值”，此例模塊 address 為 10，右側的溫度輸入值通常會是 -32768 到 + 32767 間的 1 個整數值

若設 ADR_為 16#10201A (TT=10, RR=20, AA=1A ←16 進位) 表示使用 “特別設定值”，採用 “攝氏度”，該模塊的 初始”Type Code”設定是 20 (使用 DCON utility 設的)，模塊 address 為 26。因此，若右方的溫度傳回值若是，比如”4556”表示為 45.56 度，“-500”則為-5.00 度。

如果想直接取得遠程溫度模塊的溫度值，比如，值 “2312” 來表示 23.12 攝氏度。請將 “ADR_” 參數設成 “特別設定值”。

注意：從下列 硬件的 Driver 版本開始，才有支持 “特別設定值” 功能

I-8xx7:3.11, I-7188EG:2.09, I-7188XG:2.07, μPAC-7186EG:1.01 版
iP-8xx7: 1.01 版, WP-8xx7: 1.01 版, XP-8xx7-CE6: 1.01 版, VP-25W7/23W7: 1.01 版

遠程溫度模塊的“ADR_”參數格式說明: 16#TTRRAA (16 進位表示.)

TT = 10 (表示使用 "攝氏度"), 單位 0.01 度

TT = 20 (表示使用 "華氏度"), 單位 0.01 度

TT = 00 (表示使用“標準設定值”, RR 需同時設為 00)

RR: 該 溫度 I/O 模塊的 "Type code" 設定 (初始設定時用 DCON Utility 設的)

AA: 該 溫度 I/O 模塊的 Address (01 ~ FF)

例如, 以下的 "ADR_" 設定, 分別表示:

- A. 16#102011 : (TT=10, RR=20, AA=11 ←16 進位表示), 使用 "攝氏度", “Type Code” 為 "20 : Platinum 100, a=0.00385, degree Celsius", Address 為 17 (10 進位.). 結果會是, 若溫度輸入值回傳為"2356" 表示 23.56 攝氏度, 若是 "-489" 表示 -4.89 攝氏度, 若是“999990" 則為“感測器斷線了”.
- B. 16#202A03 : (TT=20, RR=2A, AA=03 ←16 進位表示), 使用 "華氏度", “Type Code” 為 "2A : Platinum 1000, a=0.00385, degree Celsius", Address 為 3 (10 進位.). 結果會是, 若溫度輸入值回傳為 "4512" 表示 45.12 華氏度, 若是 "500" 表示 5.00 華氏度, 若是“999990" 則為“感測器斷線了”
- C. 16#01 : (TT=00, RR=00, AA=1 ←16 進位表示), 使用“標準設定值”, 通常溫度輸入值回傳值會是 -32768 到 +32767 間的 1 個整數值, Address 為 1. 必需參考附錄 D 來求得真正的溫度值.

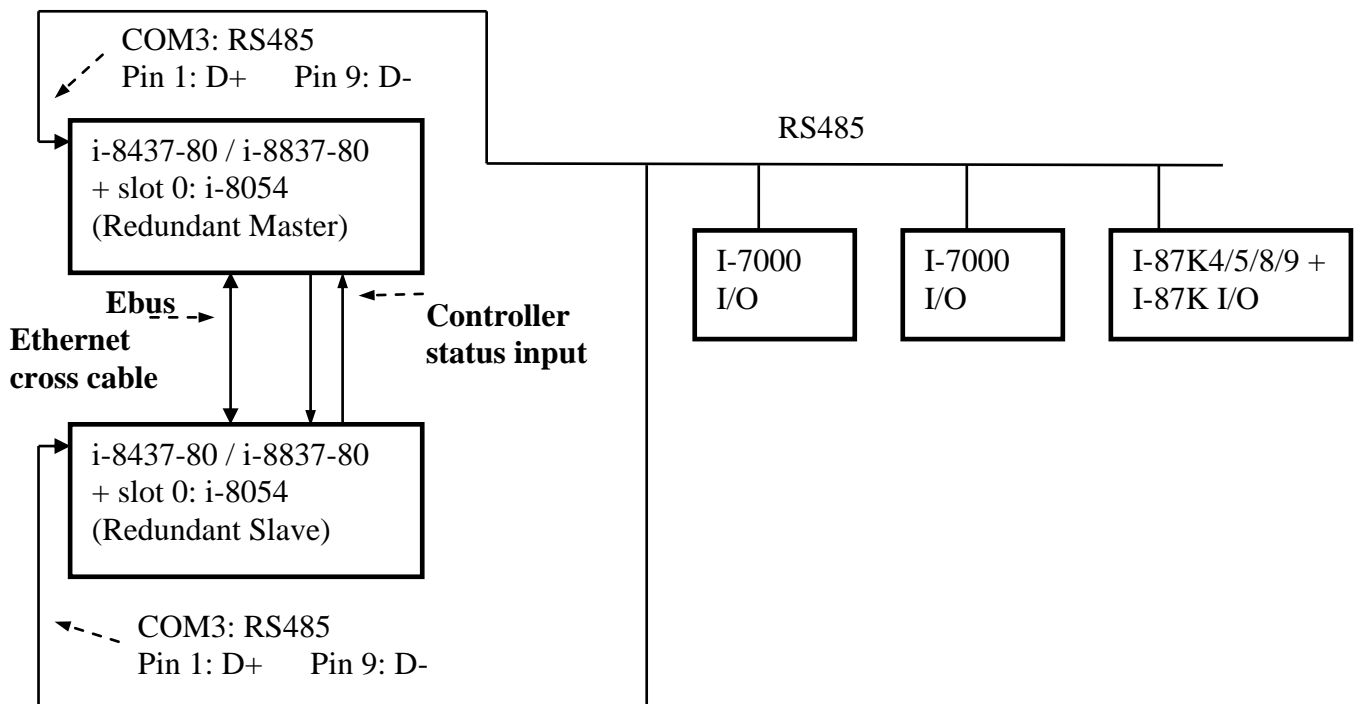
6.4: Redundant Bus7000

注意:

1. XP-8xx7-CE6 是比較好的 備援 (冗余) 系統, 請參考 FAQ-125 (http://www.icpdas.com/faq/isagraf_c.htm) 或 FAQ-093.
2. 若是採用 i-8437-80 或 i-8837-80 或 P-8x47 也可設立一套如下圖的 Bus7000 備援 (冗余) 系統. 他們的 CPU 是 80MHz. 執行速度大約是 i-8417/8817/8437/8837 (40MHz) 的 2 到 4 倍.
3. 40 MHz 的 i-8417/8817/8437/8837 與 40MHz 的 i-7188EG 與 i-7188XG 不適合用來做 Bus7000 備援 (冗余) 系統. 請採用 (1) 的方法最好, 若不想用(1), 可採用 (2)的方法.

I-8437-80 / i-8837-80 (驅動 3.20 版起 或 更新版本) 與 iP-8x47 可設立一套如下圖的 Bus7000 備援 (冗余) 系統. 2 台控制器間的 Ebus 是用來交換控制資料. 另外請將 Master 上 i-8054 的 Ch.1 D/O 連到 Slave 上 i-8054 的 Ch.1 D/I. 同樣的請將 Slave 上 i-8054 的 Ch.1 D/O 連到 Master 上 i-8054 的 Ch.1 D/I. 這 2 個 Status input 是用來告訴對方 – “我還活著”.

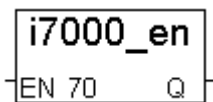
I-8437-80 : Bus7000 備援 (冗余) 系統



運作原理:

1. 當系統啟動時, 最初的 Bus7000 控制權屬於 “Redundant Master”.
2. 假如 “Redundant Master” 死機(或沒電), “Redundant Slave” 會接管 Bus7000 的控制權.
3. 假如 “Redundant Master” 又活過來了, 它會再接管 Bus7000 的控制權.
4. 2 台 controller 間的控制資料透過 Ebus 來傳輸 (可用一條 對接的 Ethernet 線 (Cross cable), 不需接到 Ethernet Switch)

“i7000_en” 函式用來 開啟/關閉 Bus7000 的控制權. 工控器開機後的內定值為 Enable.



輸入參數:

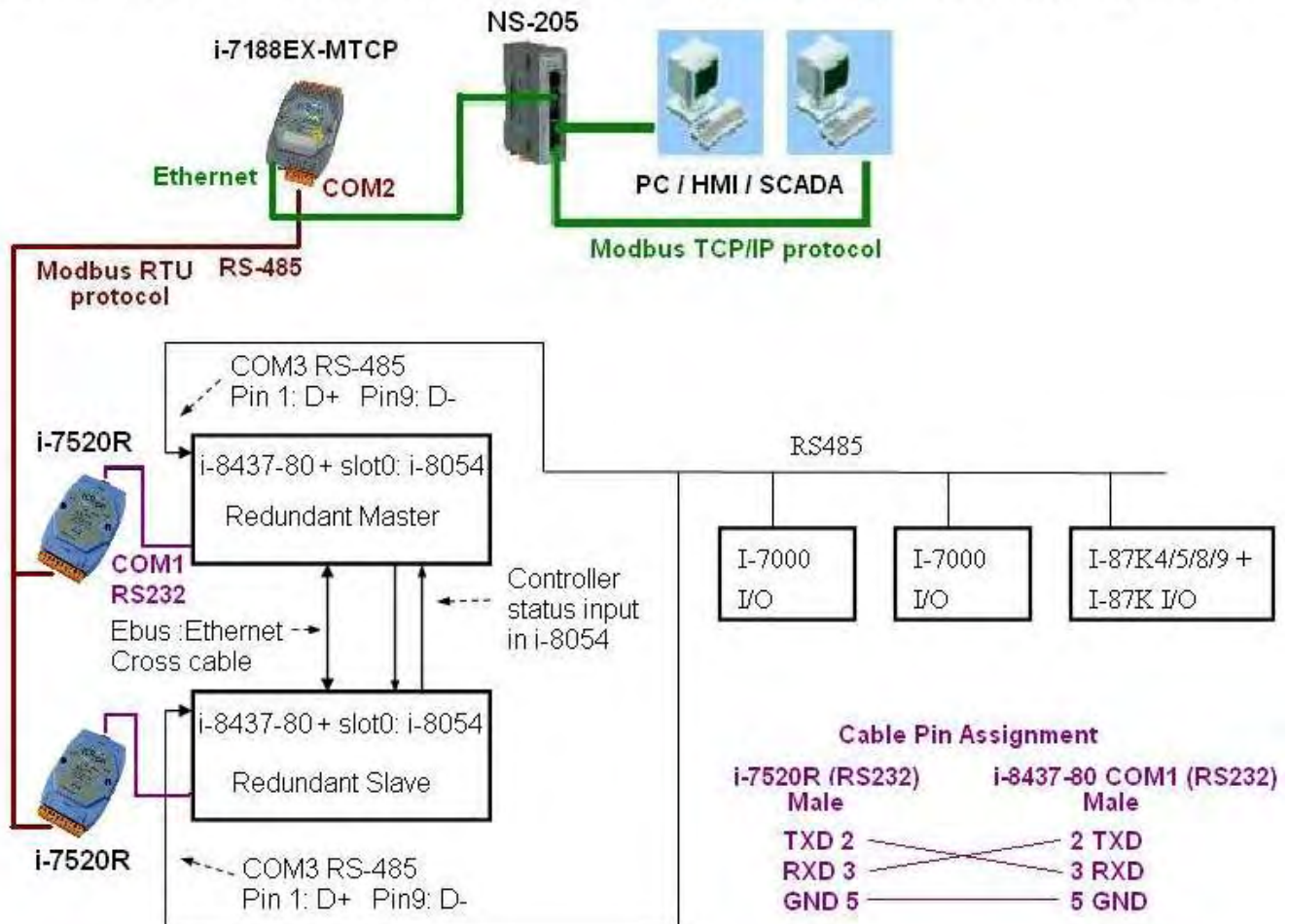
EN_7000_ 整數 True: 開啟, False: 關閉

傳回值:

Q_ 布林 永遠傳回 True.

如果該 i-8437-80 不是在 redundant 激活 (Active) 狀態, 可以在程式內使用 “COM_MRTU” 函式來 關閉 i-8437-80 COM1 的 Modbus RTU 功能, 這樣它就不會回覆任何 PC / HMI / SCADA 送來的詢問命令. 反之若該 i-8437-80 是在 redundant 激活 (Active) 狀態, 也可以在程式內使用 COM_MRTU 函式來 開啟 i-8437-80 COM1 的 Modbus RTU 功能. 這樣做的用意在於可以確保任一時間只有 redundant 激活 那台會回覆 PC / HMI 送來的詢問命令 (請參考範例 demo_49a 與 demo_49b). 關於 i-7188EX-MTCP (Modbus TCP/IP to Modbus RTU gateway) 的使用方法, 請參考第 20.5 節或 www.icpdas.com – FAQ – Software – ISaGRAF – 062. (重要, 2 台 i-8437-80 的 Net-ID 要設為一樣, 比如都設成 1 號. IP 地址則要設成不一樣, 但要在同一網域內. 比如 192.168.1.8 與 192.168.1.9)

PC / HMI / SCADA can connect to this bus7000 redundancy system with only one IP of the i-7188EX-MTCP



範例程式: demo_49a 與 demo_49b.

可至 <http://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/> 下載.

第 7 章 控制器與控制器間的資料交換

注意:

從以下 Driver 版本起, Fbus 及 Ebus 最大可用的布林封包/整數封包的編號從 256 降到 128
I-8xx7:2.42 , I-7188EG:1.32 , I-7188XG:1.29 , iP-8xx7: 1.01 , 7186EG: 1.01 , VP-2117: 1.01

7.1: 基本的 Fbus 規則

任何 I-8xx7 , I-7188EG/XG, 7186EG, iP-8xx7, VP-2117 可以透過 Fbus 資料交換系統和其他的 I-8xx7 & I-7188EG/XG 控制器交換資料 (**WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 不支持 Fbus 功能, 它只支持 Ebus 功能, 請參閱 7.5 節**)。而且在 Fbus 協定裡只有布林和整數這兩種型態的資料能被交換, 若您想要交換實數資料, 請搭配使用”Int_Real”及”Real_Int”方塊(請參考附錄 A.4)。Fbus 會建立 8 個布林值的封包和 8 個 32 位元長整數的封包。布林封包與整數封包可以利用 Fbus 發送, 允許從 1 台控制器傳送資料到另 1 台或多台控制器。

Fbus 基本規則:

規則 1: 每一個布林封包都有一個識別編號, 編號的範圍從 1 到 128, 也就是說最多只可容納 128 個布林封包在 Fbus 裡作資料交換。

每一個布林封包包含 8 個布林值, 而這些布林值只能是 TURE 或 FALSE 這兩個數值。在布林封包內的布林值可指定成“內部”, “輸入”或“輸出”等布林變數或布林常數。

規則 2: 每一個整數封包都有一個識別編號, 編號的範圍從 1 到 128, 也就是說最多只可容納 128 個整數封包在 Fbus 裡作資料交換。

每一個整數封包包含 8 個 32 位元的長整數值, 整數值的範圍從 -2147483648 到 2147483647。

在整數封包內的整數值可指定成“內部”, “輸入”或“輸出”等整數變數或整數常數。

規則 3: 指定給布林封包及整數封包的識別號碼, 只能被同一個 Fbus 網路上的 1 台控制器寫出。

控制器中 不能寫出相同識別號碼的布林封包或是整數封包, 必須確定只有 1 台控制器寫出該號碼的封包。在下面的例子中, 有 5 台 I-8xx7, I-7188EG/XG 透過 Fbus 互相傳輸資料, 控制器分別稱為 S1, S2, S3, S4 和 S5。假使 S1 企圖寫出識別號碼為 1 的布林封包和識別號碼為 1 的整數封包, 其他的控制器就不能寫出具有相同識別號碼的布林封包或整數封包, 但其他的控制器可以寫出識別號碼為 3 的布林封包或識別號碼為 3 的整數封包。

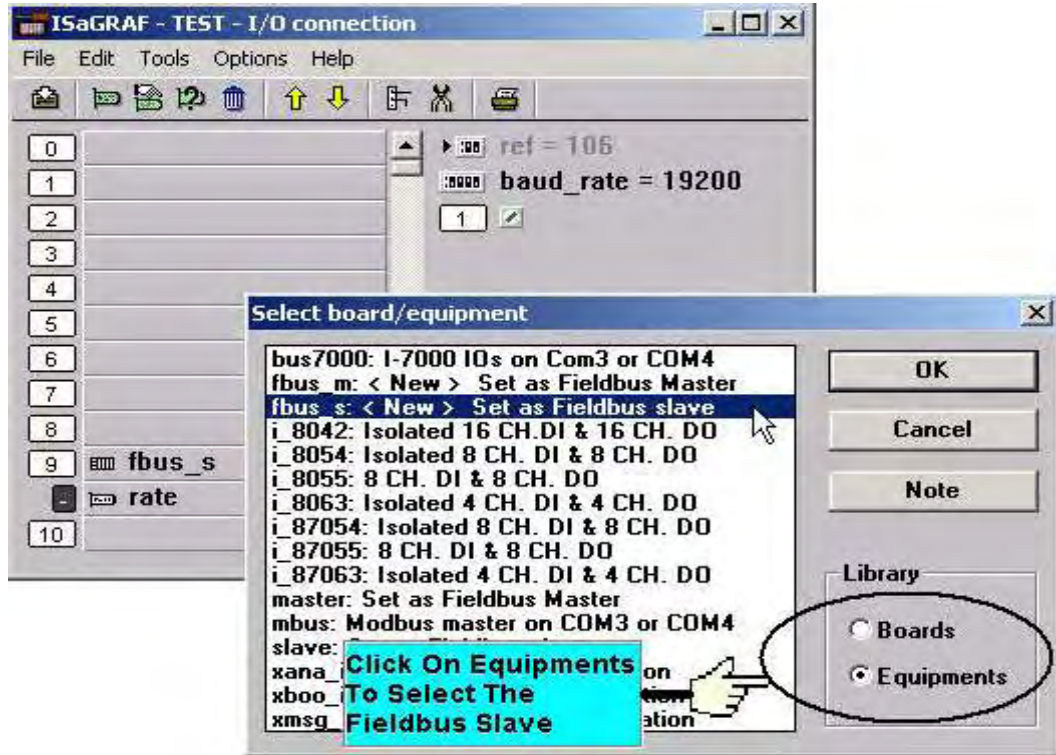
透過 Fbus 並沒有限制可以讀取多少個相同識別號碼的封包。任何的 S2, S3, S4 和 S5 都可以讀取識別號碼為 1 的布林封包或識別號碼為 1 的整數封包。

規則 4: 只有 1 台而且必需有 1 台控制器能被設定成 Fbus master, 其他的必需設成 Fbus slave。

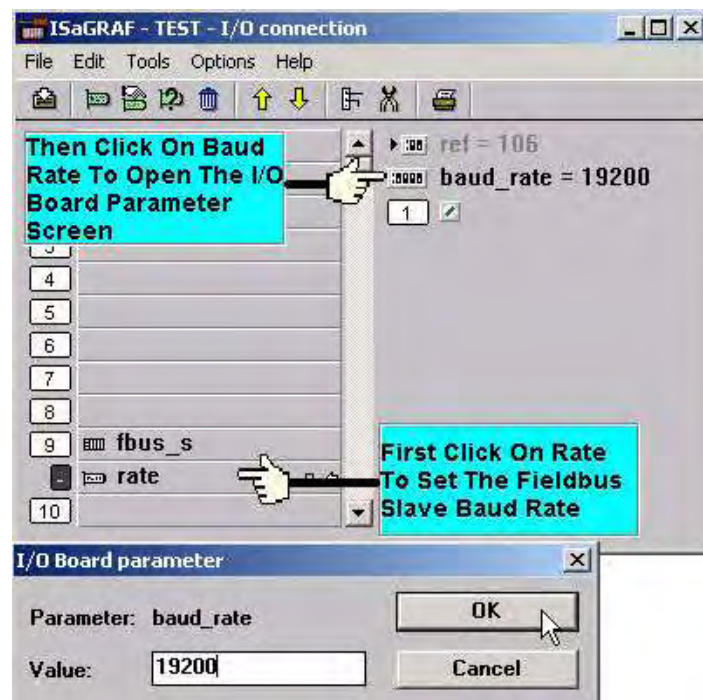
被設為 Fbus master 的控制器會傳送訊息處理該傳送及接收的命令。假使你設定兩台以上為 Fbus master, 所有資料將不會進行交換處理。

7.2: 設定 ISaGRAF PAC 為 Fbus master 或 Fbus slave

要設定控制器為 Fbus master 或 Fbus slave (WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 不支持 Fbus 功能, 它只支持 Ebus 功能, 請參閱 7.5 節), 開啟 “ISaGRAF I/O Connections” 視窗, 雙擊編號 7 以上的插槽, “Select Board/Equipments” 開啟後, 選擇 “Equipment”, 然後雙擊 “Fbus_s” 選擇設定 Fbus slave, 或雙擊 “Fbus_m” 選擇設定 Fbus master。切記, 只能設定一個控制器為 master, 也且也不能同一個控制器 master 和 slave 都設。

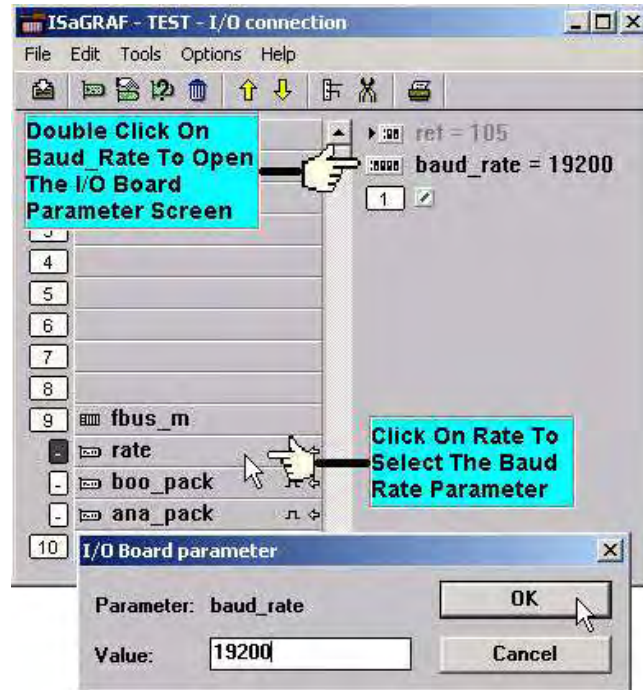


若您設定控制器為 Fbus slave, 您必須設定控制器的通訊速率參數。通訊速率的參數值分別有 2400, 4800, 9600, 19200, 38400, 57600 和 115200。控制器的預設值為 19200。在 Fbus 網路中, 所有的控制器必須設定相同的通訊速率。



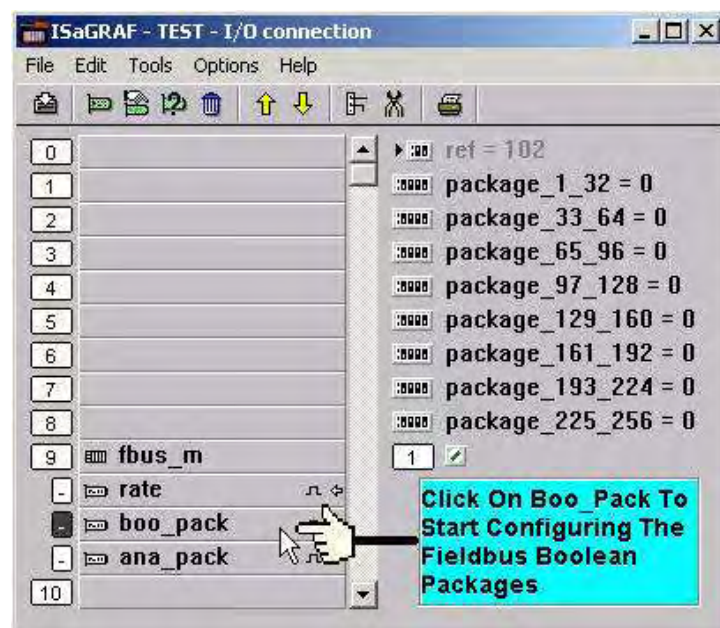
在 “fbus_s: rate”中有一個數位輸入點，這數位輸入點的功能為當 Fbus 功能開啟時，它會回傳 TRUE，若 Fbus 功能未開啟，則回傳 FALSE。

若你設控制器為 Fbus master, “Fbus_m :rate”內的 “baud_rate”這參數可設定成 2400，4800，9600，19200，38400，57600 和 115200。控制器的預設值為 19200。在 Fbus 網路中，所有的控制器必須設定相同的通訊速率。



在 “fbus_m: rate”中有一個數位輸入點，這數位輸入點的功能為當 Fbus 功能開啟時，它會回傳 TRUE，若 Fbus 功能未開啟，則回傳 FALSE

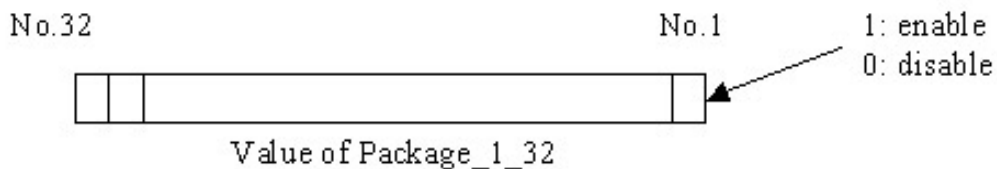
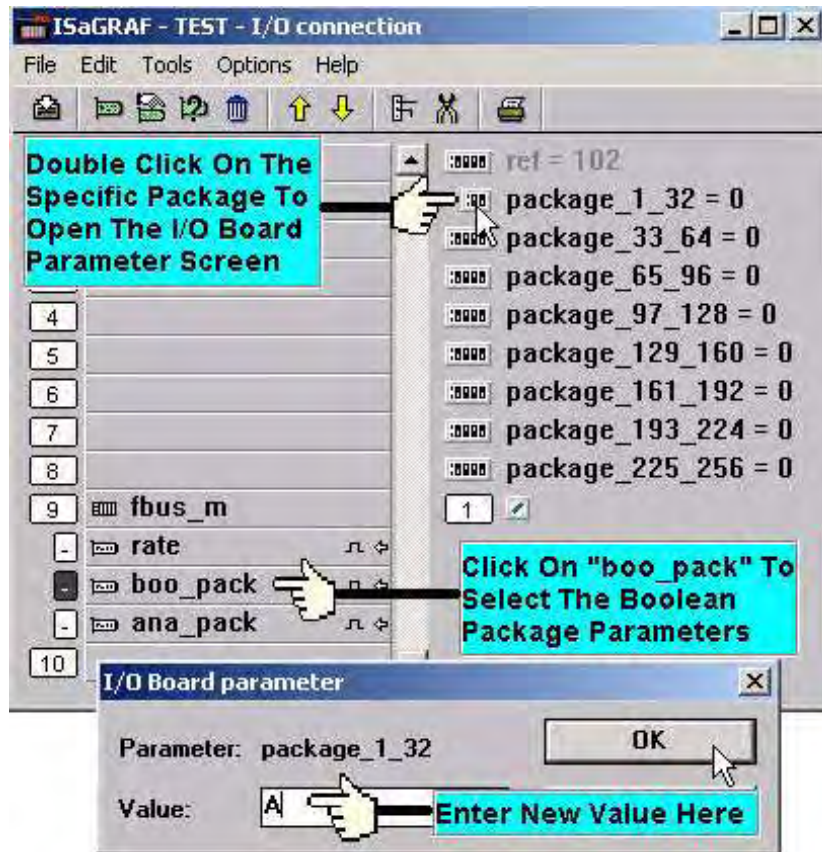
在 “fbus_m” I/O 連結畫面中，選取 “boo_pack” 選項，boo_pack 選項可以幫助我們設定布林封包。



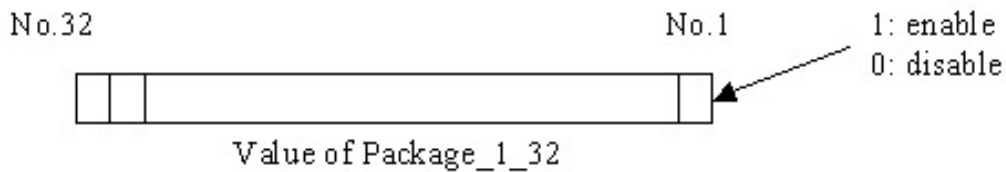
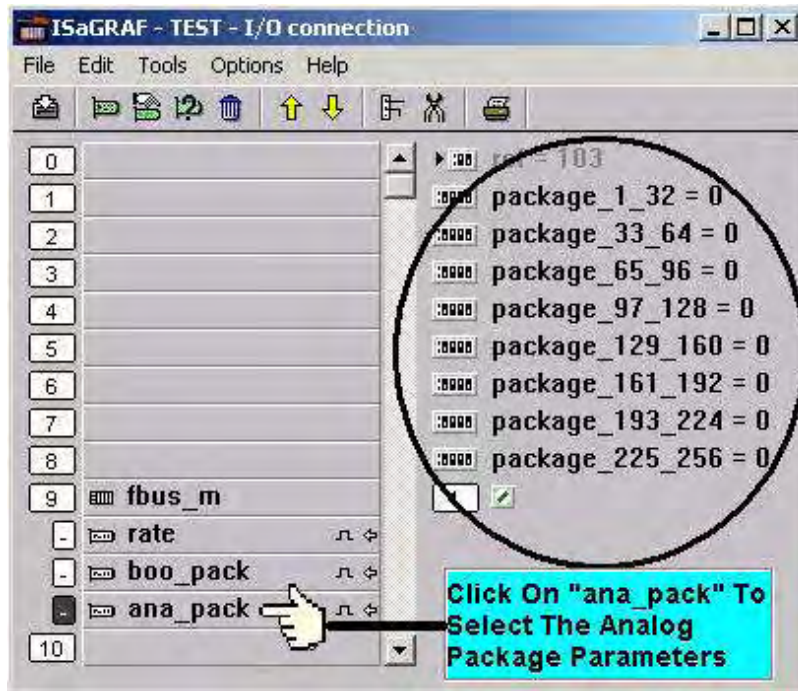
“fbus_m: boo_pack” 的 “package_xxx_xxx” 參數表示在 Fbus 網路中可被讀取或寫入的布林封包編號。這個參數是一個 32-bit 的 16 進位整數

例子：

如果 package_1_32 被設定為 FFFFFFFF，這將使得 1 號至 32 號的封包可透過 Fbus 網路被讀或寫。如果 Package_1_32 被設為 A 這將使得 2 號及 4 號的布林封包可透過 Fbus 網路被讀或寫。Fbus 網路上有愈多封包被開啟，通訊的效能會愈慢。僅記在心，總是只開啟在 Fbus 上的應用程式所需要的封包編號，所以你在 Fbus 網路上將有較大的通訊效能。



在“fbus_m:ana_pack”的“package_xxx_xxx”參數表示在 Fbus 網路中可被讀取或寫入的的整數封包號碼。“fbus_m:ana_pack”主要為透過 Fbus 網路讀寫 32 位元整數值，每一個參數值用 16 進位 32 位元整數值表達，架構規則如同布林封包。



7.3: 編寫 Fbus 封包程式

在您透過 Fbus 網路作資料交換之前，請先確定每一個控制器是否已設定為 fbus master 或 fbus slave(注意:同一個 Fbus 網路只有一個 控制器 能當 master)，請參考 7.2 節。

(WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 不支持 Fbus 功能, 它只支持 Ebus 功能, 請參閱 7.5 節)

下面的 Fbus 函式能被用於 LD 程式，透過 Fbus 網路交換資料。

Fbus_b_r	讀取一個布林封包.
Fbus_b_w	寫出一個布林封包.
Fbus_n_r	讀取一個整數封包.
Fbus_n_w	寫出一個整數封包.
Fbus_f_w	寫出一個實數封包.
Fbus_f_r	讀取一個實數封包.

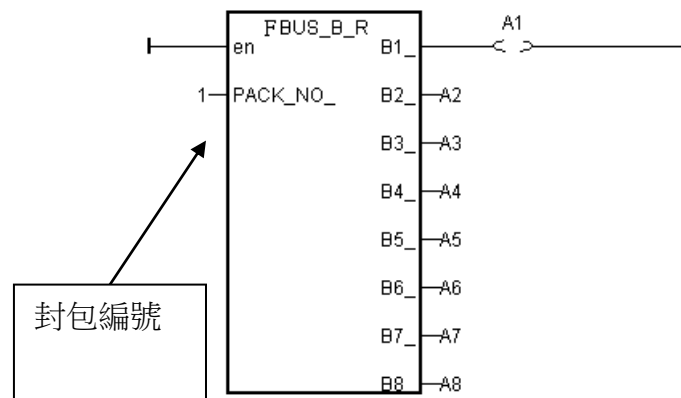
(實數封包與整數封包使用同一個記憶區, 請不要把同一個封包編號 使用成實數 又同時使用成整數, 不然值會錯誤, 有時 Fbus_f_r 更會發生 Local Fault 編號 116,請參考第 10.6 節)

下面的函式可用來偵測每個 Fbus 封包目前的通訊正常與否.

Fbus_sts	取得 Fbus 封包的狀態
----------	---------------

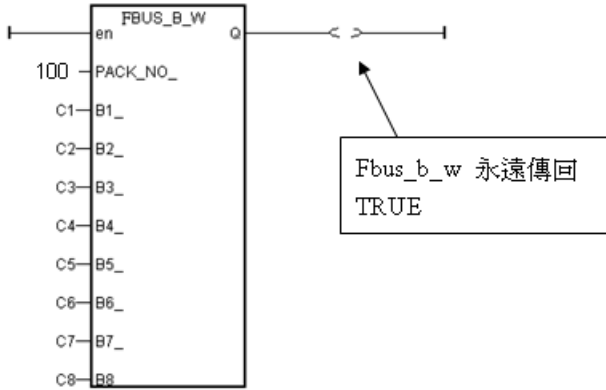
Fbus 函式 1: "Fbus_b_r"

"Fbus_b_r"函式透過 Fbus 網路可以讀取布林封包。 在下面的例子中，“Fbus_b_r” 的布林封包識別號碼為 1，A1 變數為讀取編號為 1 的布林封包的第一個布林值，A2 變數為讀取第二個布林值，A3 變數為讀取第三個布林值，其他的變數都有相同的作用，直到 A8 變數。



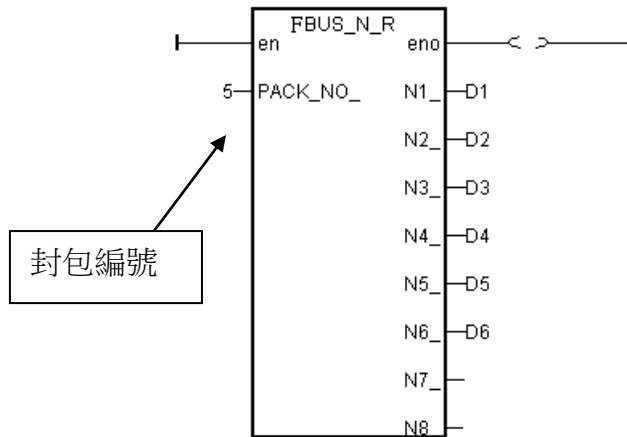
Fbus 函式 2: "Fbus_b_w"

“Fbus_b_w”函式透過 Fbus 網路寫出布林封包，在下面的例子中，”Fbus_b_w”的布林封包識別號碼為 100，C1 變數可以寫出到布林封包號碼為 100 的第一個布林值，C2 變數可以寫出第二個布林值，C3 變數可以寫出到第三個布林值，其他的變數都有相同的作用，直到 A8 變數。



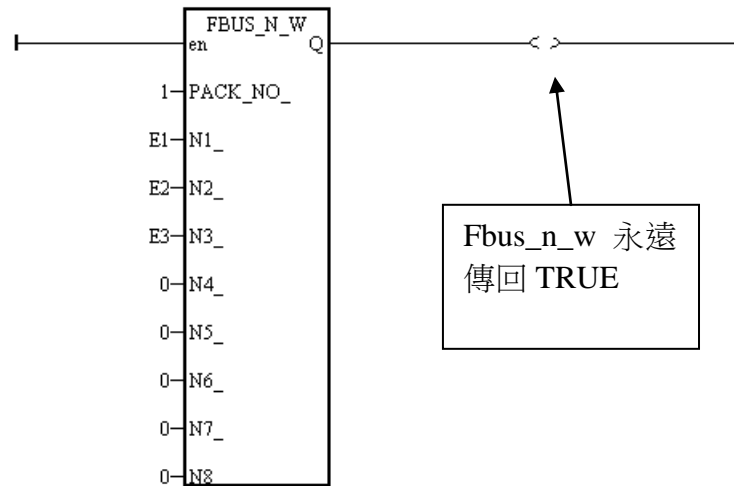
Fbus 函式 3: "Fbus_n_r"

"Fbus_n_r"函式透過 Fbus 網路可以讀取整數封包。 在下面的例子中，“Fbus_n_r”的整數封包識別號碼為 5，D1 變數為讀取編號為 5 的整數封包的第一個整數值，D2 變數為讀取第二個整數值，D3 變數為讀取第三個整數值，其他的變數都有相同的作用，直到 D6 變數。



Fbus 函式 4: "Fbus_n_w"

"Fbus_n_w" 函式透過 Fbus 網路寫出整數封包，在下面的例子中，"Fbus_n_w" 的整數封包識別號碼為 1，E1 入變數可以寫出到整數封包號碼為 1 的第一個整數值，E2 變數可以寫出的第二個整數值，E3 變數可以寫入第三個整數值。



7.4: 透過 Fbus 的資料傳輸範例

範例描述:

在下面這個範例裡，3 個 I-8xx7, I-7188EG/XG 控制器經由 Fbus 互相連接，這三個控制器分別稱為 “SA”(master 控制器 #1)， “SB”(slave 控制器 #2)， “SC”(slave 控制器 #3)。

SA 控制器內的一個輸入變數值必需分享出來讓其他的控制器能夠存取，而這個整數輸入值我們稱它為 “ZZ”。

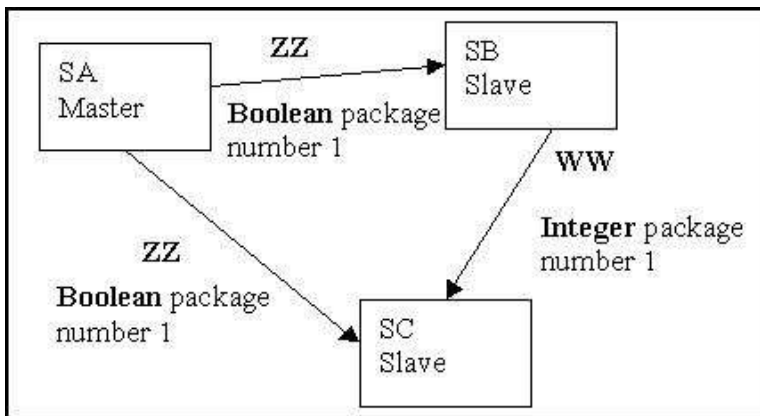
在這個範例中，我們必需在 SA 控制器中建立布林輸入值 “ZZ”。所以使用 “ISaGRAF project” 視窗程式來宣告 “ZZ” 為布林輸入變數，接著開啟 “ISaGRAF I/O connections” 視窗程式，將 “ZZ” 變數與 SA 控制器作連結。

接下來您可在 SB 和 SC 控制器上宣告一個布林內部變數 “ZZ”(取其它的名字亦可)。在 SB 和 SC 中，ZZ 變數一定要宣告成內部變數，因為 ZZ 的值是由 SA 的某個 D/I 得來，所以在 SB 和 SC 中就不須設定為布林輸入變數。

在這個範例中有個額外的設定，即在 SB 控制器中內宣告一個 WW 內部整數變數，讓 SC 控制器能夠讀取 WW。

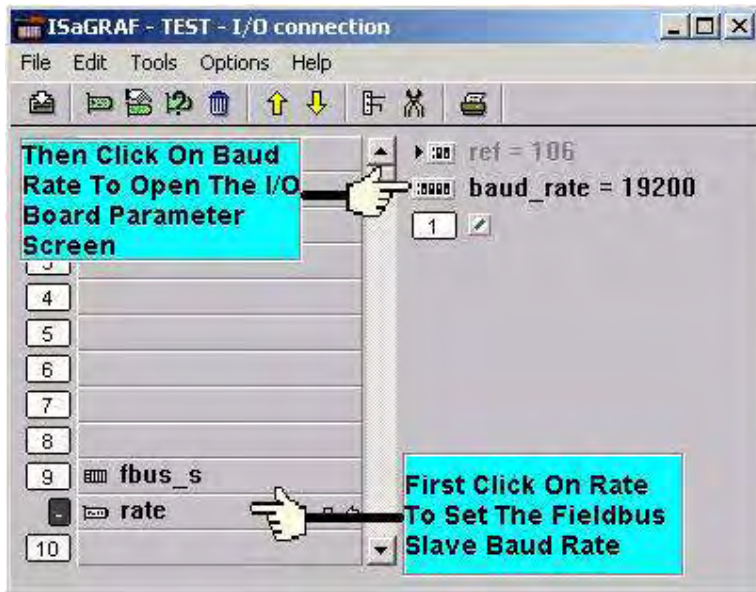
例子:

SA 控制器為 Fbus master 控制器，而 SB 和 SC 控制器為 Fbus slave 控制器，他們的通訊速率均為 19200 bps。



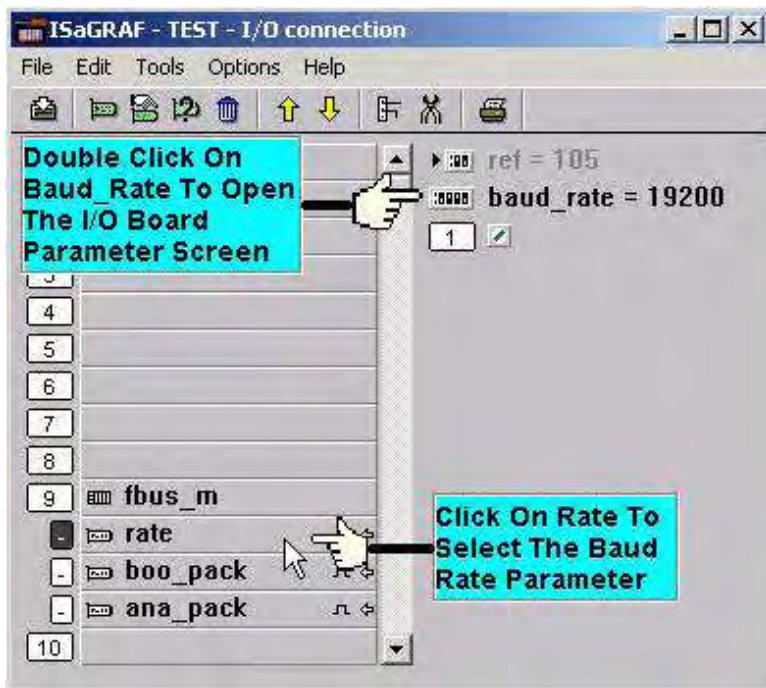
設定 SB 及 SC 控制器 為 Fbus Slave:

開啟 SB 專案及 SC 專案的 “ISaGRAF I/O connects” 視窗程式，設定 SB 和 SC 控制器為 Fbus slave。

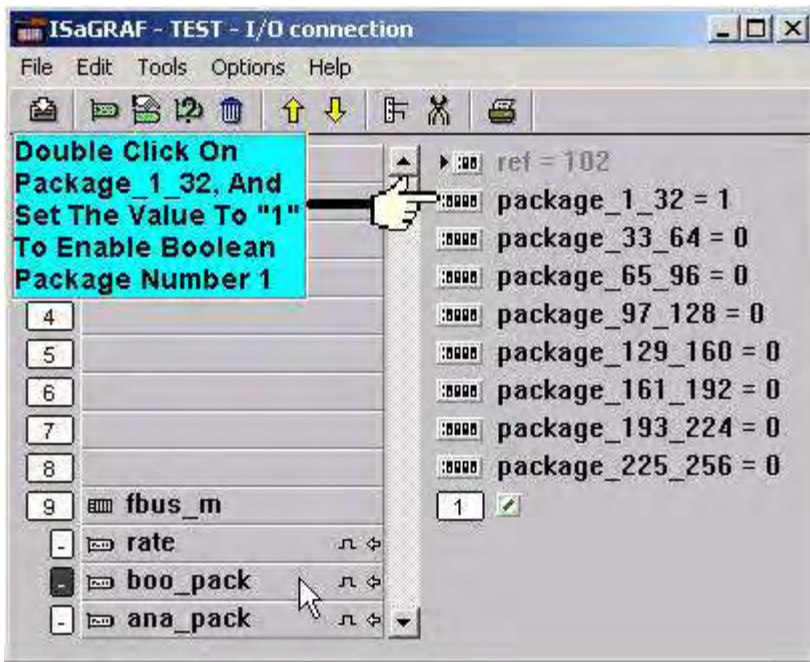


設定 SA 控制器 為 Fbus Master:

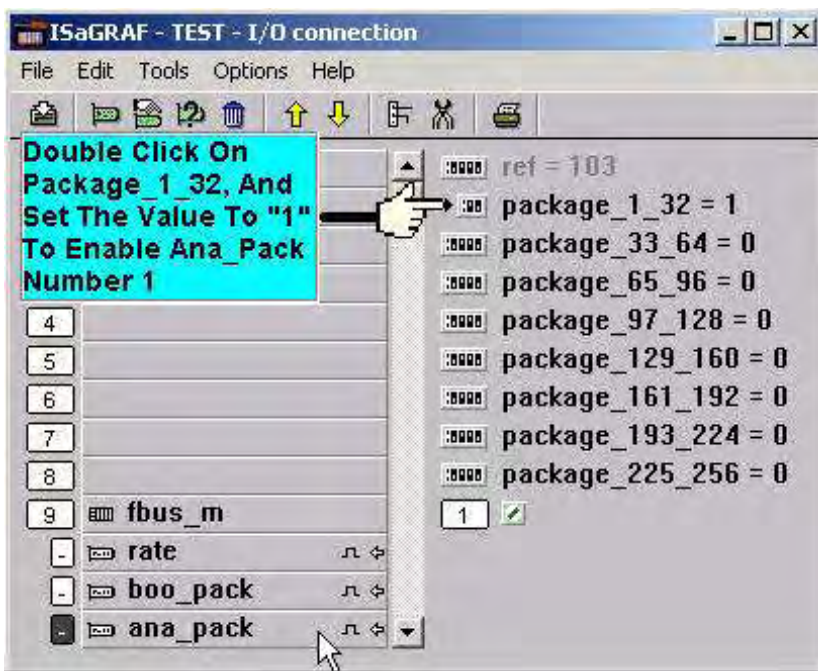
開啟 SA 專案的 “ISaGRAF I/O Connections” 視窗程式，設定 SA 控制器為 Fbus master。



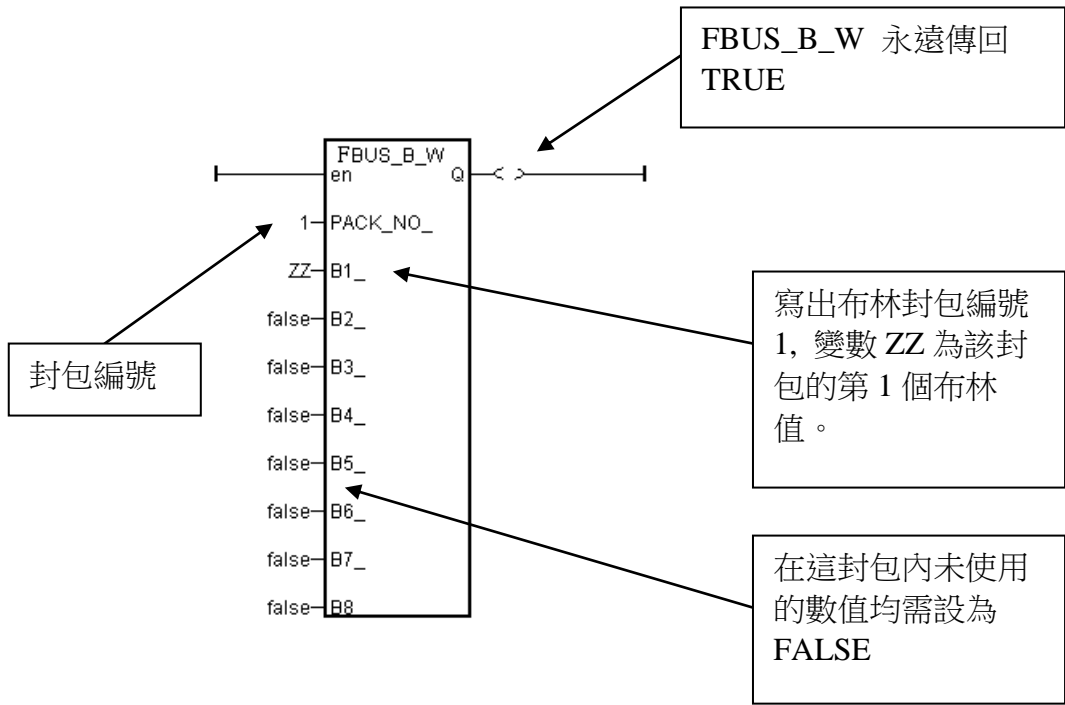
將 SA 控制器的布林參數封包參數設為 1，意思是只有編號為 1 的布林封包可以在此 Fbus 網路上收送。



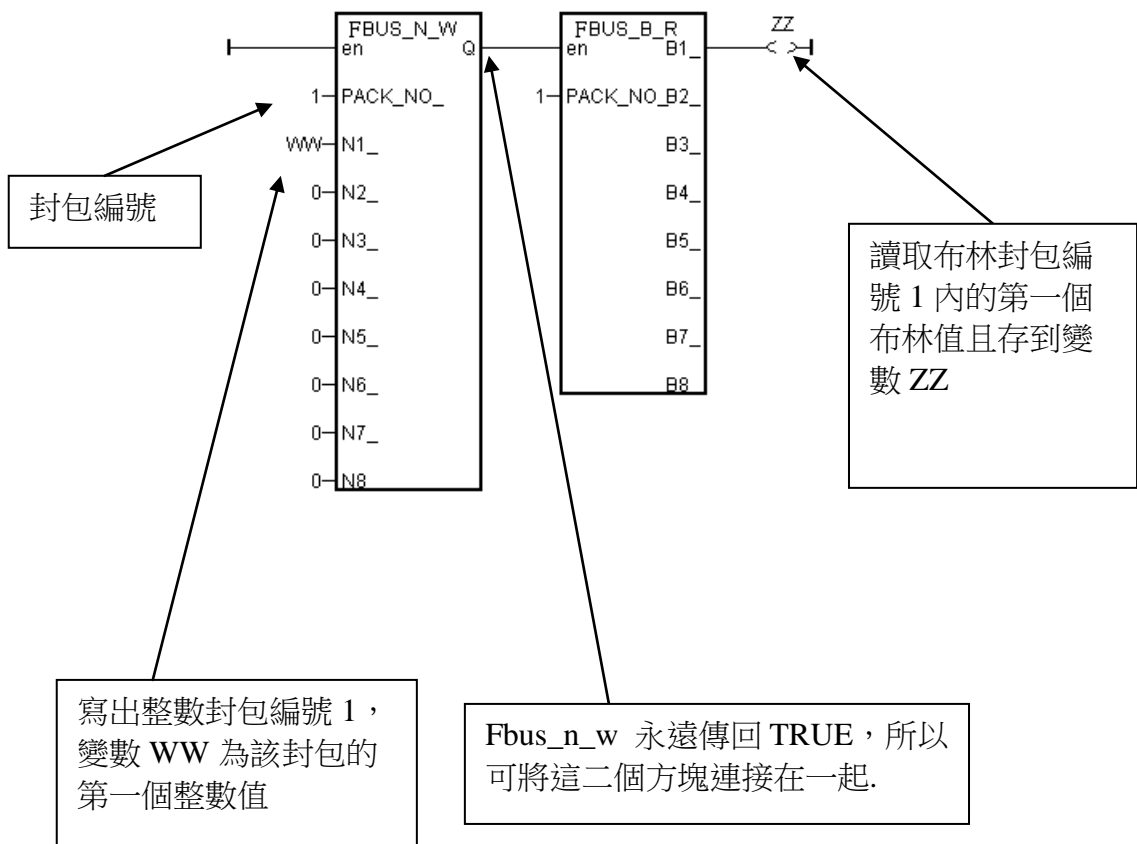
亦將 SA 控制器的整數參數封包參數設為 1，意思是只有編號為 1 的整數封包可以在此 Fbus 網路上收送



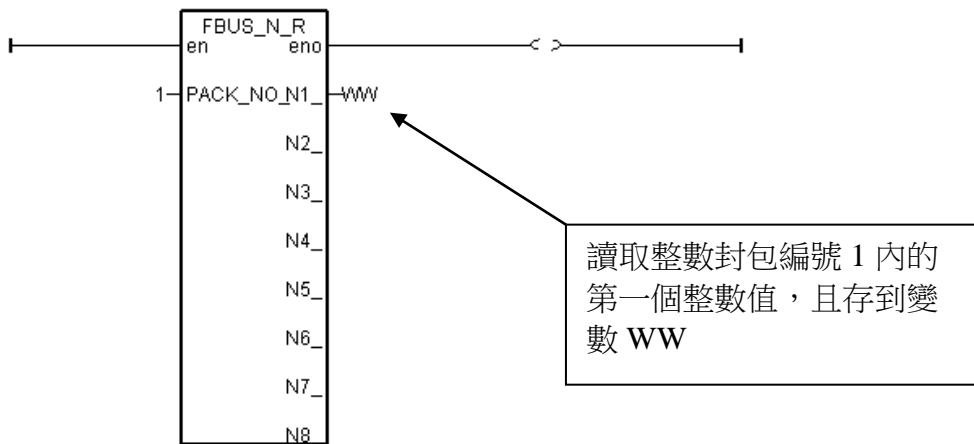
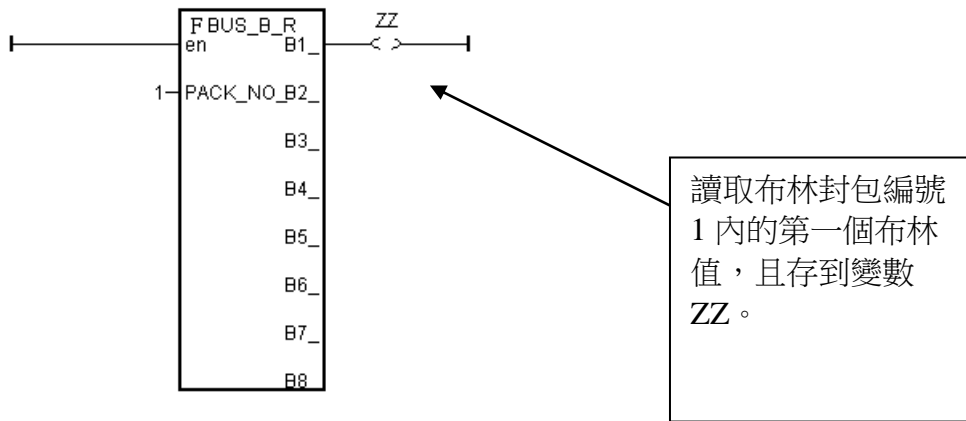
SA 控制器的 ISaGRAF 階梯圖程式



SB 控制器的 ISaGRAF 階梯圖程式

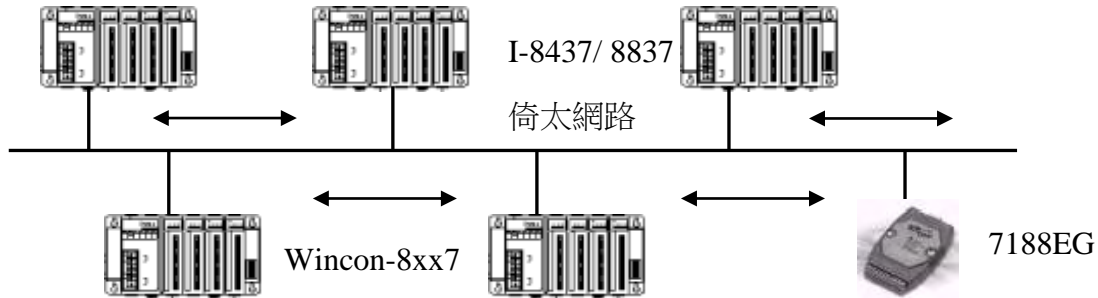


SC 控制器的 ISaGRAF 階梯圖程式



7.5: 編寫 Ebus 資料交換程式

Ebus 是一種軟體的機制，允許工控器之間可透過倚太網路互相傳遞資料。Ebus 只可用於區域網路，無法透過 gateway 連到其它網路。



I-8437-80, I-8837-80 控制器的驅動程式從 2.15 版起，開始提供 Ebus 功能，而 I-7188EG 則從 1.08 版起開始提供 Ebus 功能， μ PAC-7186EG, iP-8x47, WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6 則都有支持 Ebus。可以從下列網址找到所需的驅動程式：

<http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

並且 I/O 函式庫也要連帶更新到 ISaGRAF 內。可由下列網址下載。

<http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

之後請參閱第 1.2 節的方法安裝。

注意：

1. 從以下 Driver 版本起，Fbus 及 Ebus 最大可用的 布林封包 & 整數封包 的編號 從 256 降到 128。
I-8xx7:2.42 , I-7188EG:1.32 , I-7188XG:1.29 , iP-8xx7: 1.01 , 7186EG: 1.01 , VP-2117: 1.01
2. 假如使用的 Controller 型號為 Wincon-8x47 / 8x46 (雙網口 Wincon), WP-8xx7, XP-8xx7-CE6, Ebus 的網線必須接在 它的 LAN2 網口。
3. 同一個 Ebus 網路內的 Controller 必須將 IP 地址設在同一個網域內，且 Mask 必須設成 255.255.255.0，比如，(192.168. 1. 3), (192.168. 1. 5), (192.168. 1. 20)。

7.5.1: 基本的 Ebus 規則

Ebus 會建立 8 個布林值的封包和 8 個 32 位元長整數的封包。布林封包與整數封包可以利用 Ebus 發送，允許從 1 台控制器傳送資料到另 1 台或多台控制器，在 Ebus 協定裡只有布林和整數這兩種型態的資料能被交換，若您想要交換實數資料，請搭配使用”Int_Real”及”Real_Int”函數(請參考附錄 A.4)。

基本的 Ebus 規則與 Fbus 類似 (參閱第 7.1 節)。

規則 1: 每個 Ebus 網路有一個 “Group_No”，其值從 1 到 10。只有在同一個 “Group_No” 內的工控器可以互相傳遞資料。

例如，若一個區域倚太網路內有 5 台控制器，分別命名為 A1, A2, A3, A4, A5。若 A1, A2 & A3 被指定為 Group_No = 1，然而 A4 & A5 被指定為 Group_No = 2。則，A1 可以和 A2 及 A3 交換資料，卻不能和 A4 及 A5 交換資料。

規則 2: 每一個 Ebus 布林封包都有一個識別編號，編號的範圍從 1 到 128，也就是說最多只可容納 128 個布林封包在同一個 Ebus: Group No 裡作資料交換

每一個布林封包包含 8 個布林值，而這些布林值只能是 TRUE 或 FALSE 這兩個數值。在布林封包內的布林值可指定成“內部”，“輸入”或“輸出”等布林變數或布林常數。

規則 3: 每一個 Ebus 整數封包都有一個識別編號，編號的範圍從 1 到 128，也就是說最多只可容納 128 個整數封包在同一個 Ebus: Group No 裡作資料交換

每一個整數封包包含 8 個 32 位元的長整數值，整數值的範圍從 -2147483648 到 2147483647。

在整數封包內的整數值可指定成“內部”，“輸入”或“輸出”等整數變數或整數常數。

規則 4: 指定給布林封包及整數封包的識別號碼，只能被同一個 Ebus: Group No 網路上的 1 台控制器寫出。

不同台的控制器不能在同一個 Ebus: Group No 網路上寫出相同識別號碼的布林封包或是整數封包，必須確定只有 1 台寫出該號碼的封包。

在下面的例子中，有 5 台 ISaGRAF PAC 透過 Ebus 互相傳輸資料, Group No 皆設為 1, 控制器名稱分別稱為 S1, S2, S3, S4 和 S5。假使 S1 企圖寫出識別號碼為 1 的布林封包和識別號碼為 2 的整數封包，其他的控制器就不能寫出具有相同識別號碼的布林封包或整數封包，但其他的控制器可以寫出識別號碼為 2 的布林封包或識別號碼為 1 的整數封包。

在控制器中透過 Ebus 並沒有限制可以讀取多少個相同識別號碼的封包. S2,S3,S4 和 S5 的任何 1 台都可以讀取識別號碼為 1 的布林封包或識別號碼為 2 的整數封包。

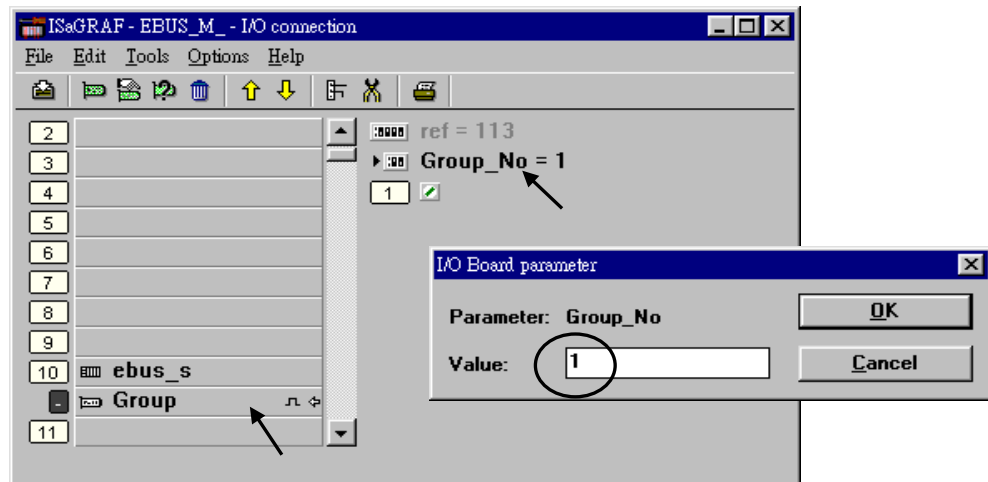
規則 5: 在同一個 Ebus: Group No 內，只有一台控制器能被設定成 Ebus master, 其他的控制器都必需設成 Ebus slave.

被設為 Ebus master 的控制器會傳送訊息處理如何傳送及接收的命令。假使你設定兩台以上為 Ebus master，所有資料將不會進行交換處理

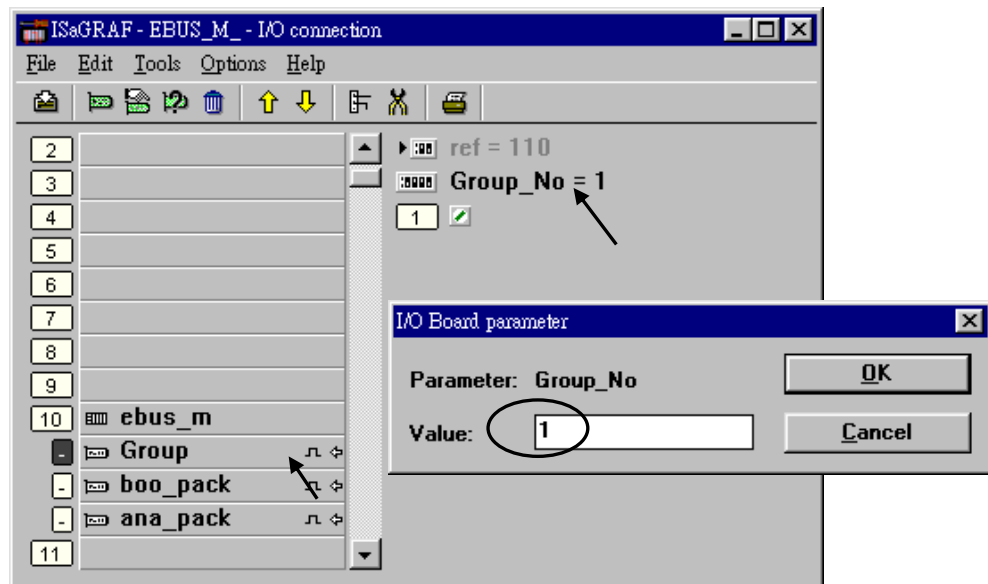
7.5.2: 設定 ISaGRAF PAC 為 Ebus Master 或 Slave

要設定控制器為 Ebus master 或 Ebus slave, 請開啟 “ISaGRAF I/O Connections” 視窗, 雙擊編號 7 以上的插槽號碼, 在 “Select Board/Equipments” 開啟後, 選擇 “Equipment”, 然後雙擊 “Ebus_s” 選擇設定為 Ebus slave, 或雙擊 “Ebus_m” 選擇設定 Ebus master. 切記, 在同一個 Ebus: Group No 內, 只能設定 1 台控制器為 Ebus master, 也且也不能在同 1 台控制器內 Ebus master 和 Ebus slave 都設.

假如設為 **Ebus slave**, 只有 “Group_No” 參數要指定, 其值可設為 1 到 10 的任一個.

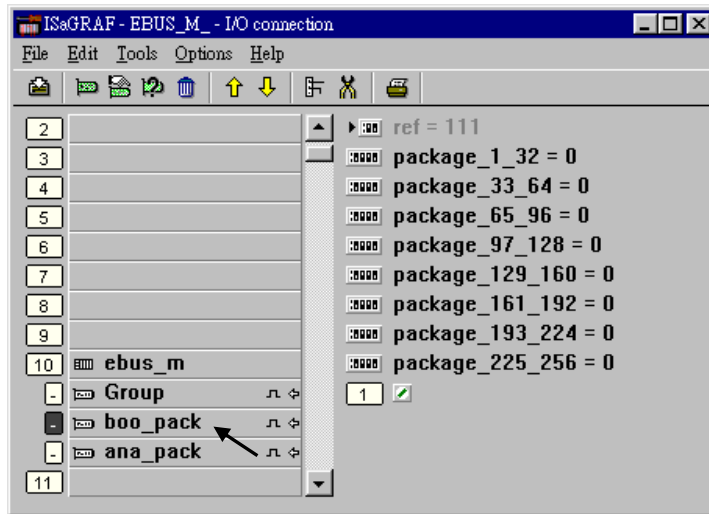


假如設為 **Ebus master**, “Group_No” 參數要指定其值為 1 到 10 的任一個. 之後還有一些參數要指定.



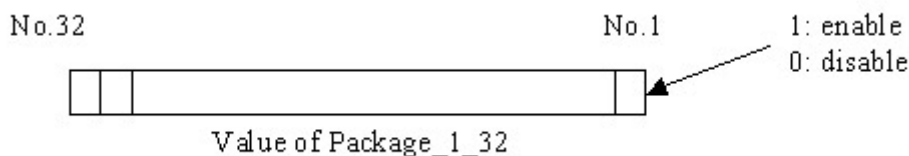
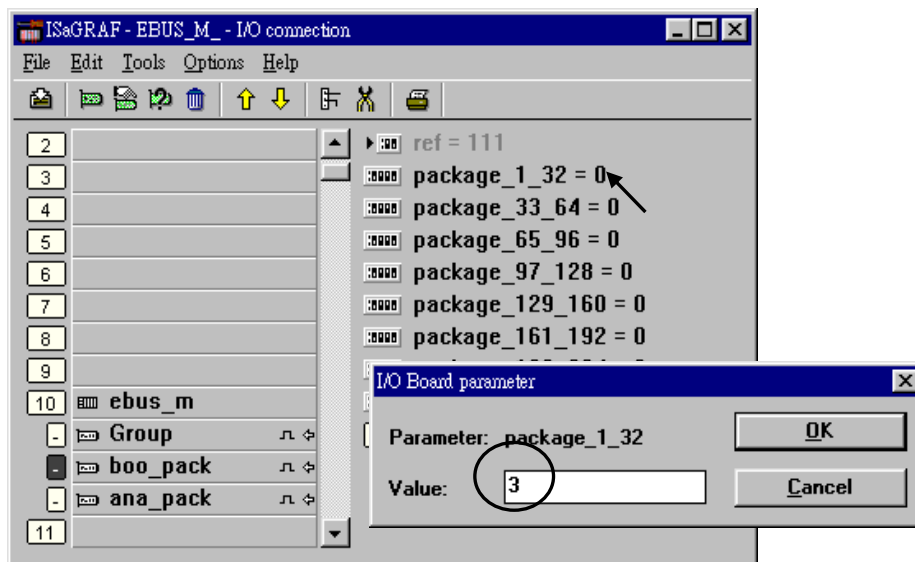
設定 Ebus master 布林封包:

在“Ebus_m” I/O 連結畫面中，選取 “boo_pack” 選項，boo_pack 選項可以讓我們設定布林封包。

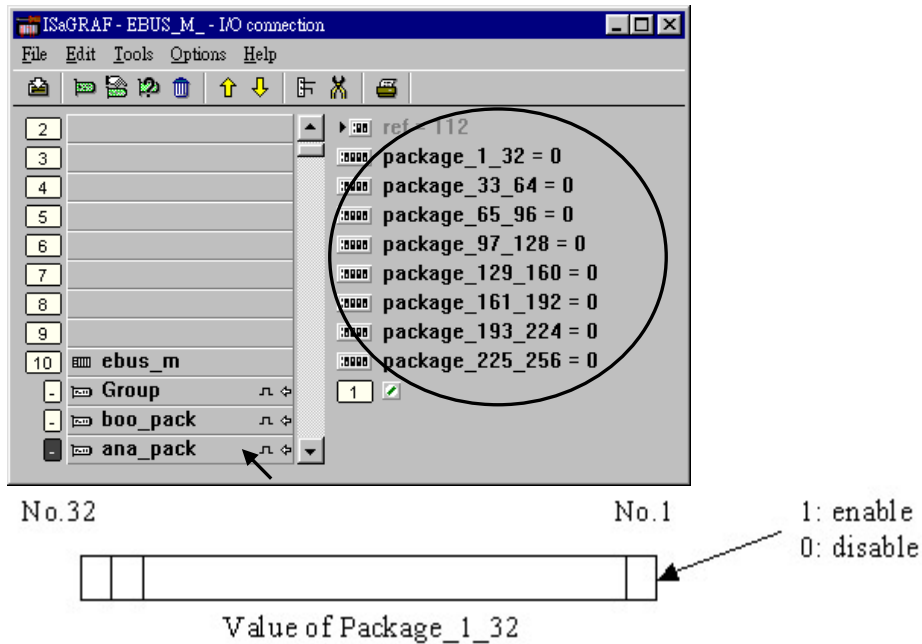


“Ebus_m: boo_pack” 的 “package_xxx_xxx” 參數表示在 Ebus 網路中可被讀取或寫入的的布林封包編號。這個參數是一個 32-bit 的 16 進位整數

例子, 如果 package_1_32 被設定為 FFFFFFFF，這將使得 1 號至 32 號的封包都可透過 Ebus 網路被讀或寫。如果 Package_1_32 被設為 A 這將只使得第 2 號及第 4 號的布林封包可透過 Ebus 網路被讀或寫。Ebus 網路上有愈多封包被開啟，通訊的效能會愈慢。僅記在心，總是只開啟在 Ebus 上的應用程式所需要的封包編號，所以你在 Ebus 網路上將有較大的通訊效能。



在“Ebus_m: ana_pack”的“package_xxx_xxx”參數表示在 Ebus 網路中可被讀取或寫入的整數封包號碼。每一個參數值用 16 進位的 32 位元整數值表達,規則如同前面布林封包”boo_pack”。



7.5.3: 編寫 Ebus 封包

在您透過 Ebus 網路作資料交換之前,請先確定每一個控制器是否已設定為同 1 個 Group No, 且設定為 Ebus master 或 Ebus slave (注意:同一個 Ebus: Group No 網路只有一個 控制器 能當 Ebus master), 請參考 7.5.2 節。

下面的 Ebus 函式與 Fbus 函式很類似(參考 7.3 節),可用於 LD 程式, 透過 Ebus 網路交換資料。

- | | |
|----------|-----------|
| Ebus_b_r | 讀取一個布林封包. |
| Ebus_b_w | 寫出一個布林封包. |
| Ebus_n_r | 讀取一個整數封包. |
| Ebus_n_w | 寫出一個整數封包. |
| Ebus_F_r | 讀取一個整數封包. |
| Ebus_F_w | 寫出一個整數封包. |

(實數封包與整數封包使用同一個記憶區, 請不要把同一個封包編號 使用成實數 又同時使用成整數, 不然值會錯誤, 有時 Ebus_f_r 更會發生 Local Fault 編號 115, 請參考第 10.6 節)

下面的函式可用來偵測每個 Ebus 封包目前的通訊正常與否。

- | | |
|----------|---------------|
| Ebus_sts | 取得 Ebus 封包的狀態 |
|----------|---------------|

第 8 章 連結 Modbus RTU / ASCII 裝置

注意：不同的控制器型號能夠啟用的 Modbus Master 通訊埠數量有差別，如下表。
 (使用 WP-8xx7, VP-25W7/23W7 與 Wincon 的 COM5 到 COM14 需先 做好設定，請參閱它的
 “Getting Started Manual” 的附錄 E 或到
http://www.icpdas.com/products/PAC/i-8000/getting_started_manual.htm 去下載。

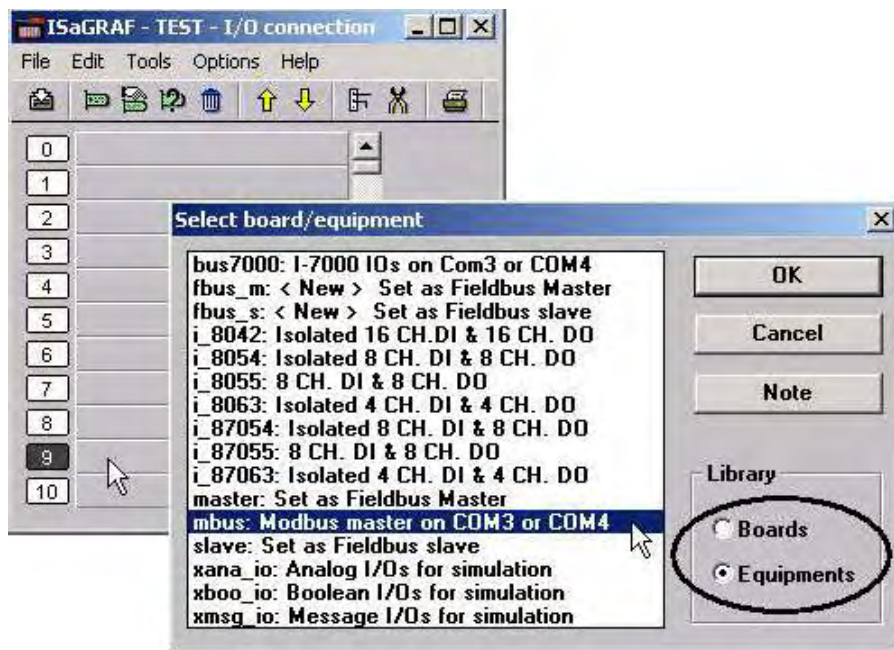
WP-8xx7, VP-25W7, VP-23W7 與 XP-8xx7-CE6 的 Ethernet port 有支持 Modbus TCP Master,
 請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-113

有時無法正確連上 Modbus device 時可以參考 http://www.icpdas.com/faq/isagraf_c.htm >
 FAQ-075 的說明來 排除 (或設定特別的 “Delay_Time”)

(表 1)	μPAC-7186EG I-7188EG/XG	iP-8xx7 I-8xx7 I-8437/8837-80	WP-8xx7 VP-25W7 / 23W7	XP-8xx7-CE6
可啟用的最大數量 Modbus Master	最多 2 port	最多 2 port	最多 10 port	最多 32 port
可用於	COM1 COM2 COM3 (3 位於 X5xx 擴 充卡上)	COM1 , COM3 , COM4 , COM5 (5 位於 I-8142iW/8144iW 擴充卡上)	COM2 , COM3, COM5 ~ COM14 (5 ~ 14 位於 I-8142iW/8144iw 擴充卡上)	COM2 ~ COM33 (6 ~ 33 位於 I-8142iW/8144iw 擴充卡上)

8.1: 設定 控制器 以連接 Modbus 裝置

要設定 ISaGRAF 控制器與其他 Modbus 裝置連接，您必須在 ISaGRAF 專案中設定“Mbus”或“Mbus_asc”連結，Mbus 是指 設立 該通訊埠為 Modbus RTU Master，而 Mbus_asc 則是設立該通訊埠為 Modbus ASCII Master。首先請開啟“ISaGRAF I/O Connections”視窗，然後雙擊插槽編號 7 或 7 以上的插槽開啟“Select Board/Equipments”視窗畫面，選擇“Equipment”選項，單擊“Mbus :Modbus Master ...”選項 (ASCII 請選“MBbus_asc”)，接著按下 OK 鍵完成設定。當同時啟用多個 Modbus Master port 時，就需連結多個“Mbus”或“Mbus_asc”，每個“Mbus”或“Mbus_asc”的“port_no”參數，需都不相同。(跟 controller 型號有關,如表 1)



“Mbus”與“Mbus_asc”的參數設定描述如下：

"port_no" 參數：啟用那個 通訊埠 為 Modbus Master，值可以是 1 ~ 33 (跟 控制器型號有關，如表 1)，當同時啟用多個 Modbus Master port 時，就需連結多個“Mbus”或“Mbus_asc”，每個“Mbus”或“Mbus_asc”的“port_no”參數，需都不相同

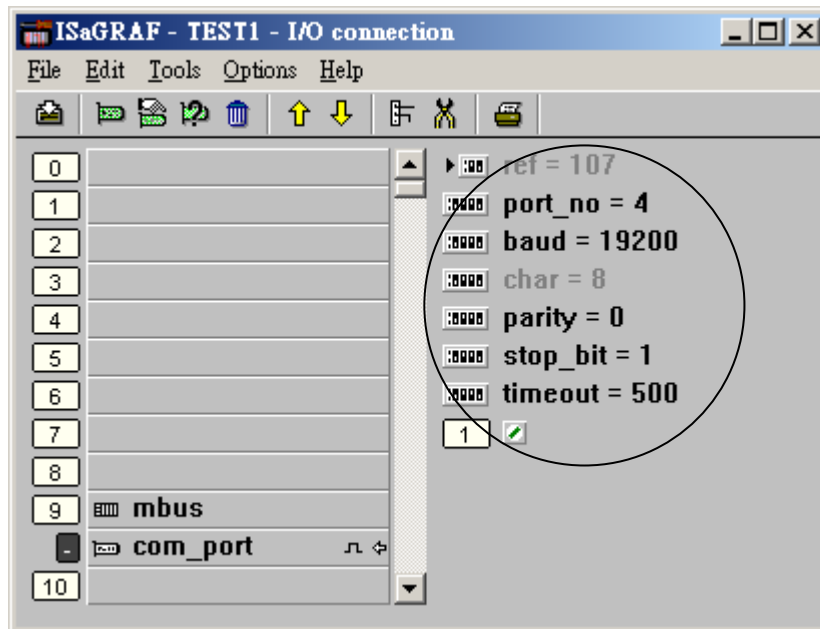
"baud" 參數：通訊速率，必須設定成 控制器和其他的 Modbus 裝置一樣的通訊速率來順利溝通，值 可以是 2400, 4800, 9600, 19200, 38400, 57600 或 115200

"char" 參數：內定值為 8，如果是使用“Mbus”此值不可變更。若是 使用“Mbus_asc”此值可以設成 7 或 8，需跟所連接的 Modbus slave 設備的設定一樣才能通起來。

"parity"參數：設定通訊的同位檢查參數，設 0 即設成同位檢查為“none”，設 1 即設同位檢查為偶數，設 2 即設同位檢查為奇數。

"stop_bit" 參數：設定 Modbus 通訊中的停止位元數字。設為 1 即表示有一個停止位元，設為 2 即表示有兩個停止位元。

"timeout" 參數：設定 1 個 Modbus Master 命令發出給 設備後，該設備需於多少時間內就需答覆，不然就當成 Time out 來處理。時間單位為 ms = 0.001 秒 (這個參數很重要，內定值是 500，即 0.5 秒內需收到 Modbus 設備的答覆，若您的 Modbus 設備回覆時間比較慢，請設大一點的值)，值設大的後果是 比較慢回覆的 設備可以通上，但若遇到有通訊斷線時 controller 會等的比較久，才會再送出下個 Modbus Master command. 此值建議請勿設小於 200. 最大建議不超過 4000.



注意:

如果要在 I-8xx7, I-7188EG, μ PAC-7186EG, iP-8xx7 的 COM1 上使用 Modbus Master 功能，請先參閱各別的“快速上手手冊”來“設定 COM1 為 非 Modbus RTU”的方法來關掉 COM1 原先的 Modbus RTU slave 的功能.

8.2: 編寫 Modbus 方塊程式

您必須先連結“Mbus”或“Mbus_asc”才能存取 Modbus 裝置的資料，若您尚未連結，請參考 8.1 節。若您已連結，您可以透過 Modbus 通訊協定在 ISaGRAF PAC 上和其他 Modbus 裝置之間存取資料。下面的功能方塊可以讓您應用在 LD 或 FBD 程式中透過 Modbus 通訊協定作資料的存取。

注意: Mbus24R, Mbus24R1, Mbus_XR, Mbus_XR1 只有 WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6 才有支持 (參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-101)。

Mbus_R Mbus24R	若設定“CODE_”為 Modbus function 編號 3 或 4 1. 可用來讀取最多達 12 個 word 資料 (Mbus24R 是 24 個 word), 每個 word 值為 -32768 ~ +32767 2. 也可用來讀取最多達 6 個 32-bit 整數資料 (Mbus24R 是 12 個 32-bit 整數), -2,147,483,648 ~ +2,147,483,647, 需配合使用“WD_LONG”方塊將 2 個 word 轉換成一個 32-bit 整數 3. 或用來讀取最多達 6 個 REAL 資料 (32-bit 符點數), 需配合使用“WD_LONG”方塊將 2 個 word 轉換成 1 個 32-bit 整數, 然後再將該 32-bit 整數用“INT_REAL”方塊對應成 1 個 32-bit 符點數 (Mbus24R 是 12 個 32-bit 符點數) 若設定“CODE_”為 Modbus function 編號 1 或 2 4. 可用來讀取最多達 192 個 Boolean (Bit) 資料, 需配合使用“WD_Bit”方塊將 1 個 word 轉換成 16 個 Boolean 資料 (Mbus24R 是 384 個 Boolean)
Mbus_R1 Mbur24R1	同“MBUS_R”, 但多了一個參數“PERIOD_”, 單位為秒. 可設成 1 ~ 600. 每過一段 PERIOD 時間才對該 Modbus 設備詢問一次.
Mbus_XR Mbus_XR1	可用來讀取最多達 120 個 word 資料, 或 60 個 32-bit 整數, 或 60 個 32-bit 實數. 參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-101
Mbus_N_R	使用 Modbus function 編號 3, 一次詢問 8 個 Word 資料 (-32768 ~ +32767) (每個 Modbus command 都會問 8 個 Word, 若設備不支持 一次被問這麼多個 Word, 或是只支持 Modbus function 編號 4, 請改用“MBUS_R”)
Mbus_NR1	同“MBUS_N_R”, 但多了一個參數“PERIOD_”, 單位為秒. 可設成 1 ~ 600. 每過一段 PERIOD 時間才對該 Modbus 設備詢問一次.
MBUS_B_R	使用 Modbus function 編號 1, 一次詢問 8 個 Boolean (Bit) 資料 (True 或 False) (每個 Modbus command 都會問 8 個 Bit, 若設備不支持 一次被問這麼多個 Bit, 或是只支持 Modbus function 編號 2, 請改用“MBUS_R”)
MBUS_BR1	同“MBUS_B_R”, 但多了一個參數“PERIOD_”, 單位為秒. 可設成 1 ~ 600. 每過一段 PERIOD 時間才對該 Modbus 設備詢問一次.
Mbus12W	可寫出 1 ~ 12 word (-32768 ~ +32767) 給 Modbus 設備 參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-144
MBUS_N_W	1. 使用 Modbus function 編號 6 或 16 來寫出 1 ~ 4 個 Word(-32768 ~ +32767) 給 Modbus 設備. 當“NUM_W_”為 1 時, 使用 Modbus function 編號 6. 當“NUM_W_”為 -1 時, 使用 Modbus function 編號 16 來寫出 1 個 word. 當“NUM_W_”為 2 ~ 4 時, 使用 Modbus function 編號 16 2. 也可用來寫出 1~2 個 32-bit 整數給 Modbus 設備需先使用 LONG_WD 方塊將 1 個 32-bit 整數轉成 2 個 Word 再送入 MBUS_N_W 內,此時 NUM_W_需設為 2 或

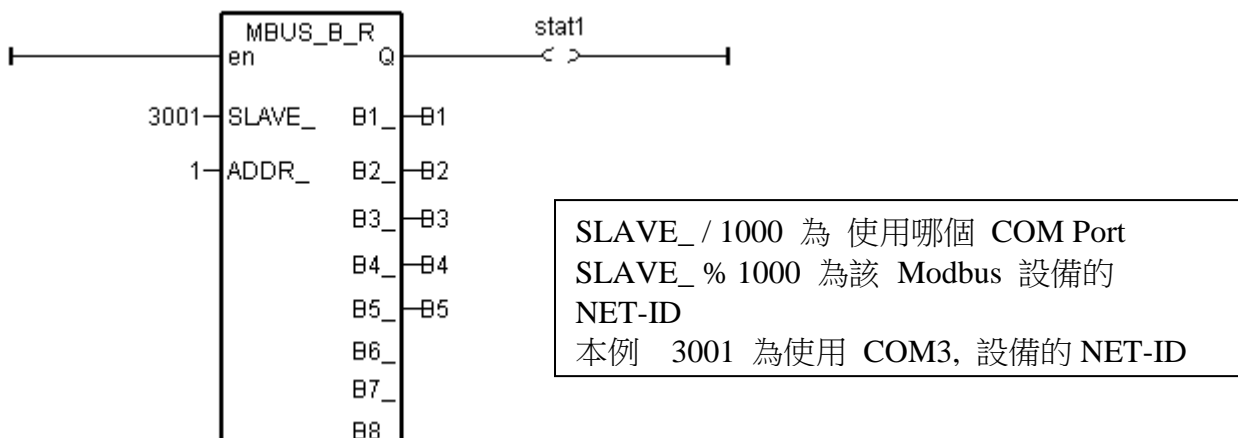
	4 3. 也可用來 寫出 1 或 2 個 32-bit 浮點數給 Modbus 設備. 需先使用 “REAL_INT” 將 1 個 32-bit 浮點數 對應成 1 個 32-bit 整數, 再用 “LONG_WD” 方塊將 這個 32-bit 整數轉成 2 個 Word 再送入 “MBUS_N_W” 內, 此時“NUM_W_”需設為 2 或 4
MBUS_B_W	使用 Modbus function 編號 5 或 15 來寫出 1~4 個 Bit 給 Modbus 設備. 當 “NUM_W_” 為 1 時, 使用 Modbus function 編號 5. 當 “NUM_W_” 為 2~4 時, 使用 Modbus function 編號 15
MBUS_WB	使用 Modbus function 編號 15 來寫出 1~16 個 Bit 給 Modbus 設備

注意:

- 同性質的“Mbus_x_x”方塊可以在同 1 台 I-8xx7, I-8xx7-80, I-7188EG/XG, μPAC-7186EG, iP-8xx7, VP-2117 中所有 port 使用的最大數量總和為 64 個。
- 同性質的“Mbus_x_x”方塊可在 WP-8xx7, XP-8xx7-CE6, VP-25W7 的每單獨 1 個 port 中使用的最大數量 為 256 個。
- “MBUS_R”, “MBUS_R1”, “MBUS24R”, “MBUS24R1”, “MBUS_N_R” 與 “MBUS_NR1” 是屬於同性質的
“MBUS_B_R” 與 “MBUS_BR1” 是屬於同性質的
“MBUS_B_W” 與 “MBUS_WB” 是屬於同性質的
“Mbus_N_W” 與 “Mbus12W” 是屬於同性質的
- 一台 WP-8xx7, XP-8xx7-CE6, VP-25W7 控制器內最多只能使用 128 個 ”Mbus_XR” 或 ”Mbus_XR1” (即 ”Mbus_XR” 加上 ”Mbus_XR1” 的數量不可超過 128 個)

Modbus 範例函式 #1: "Mbus_b_r"

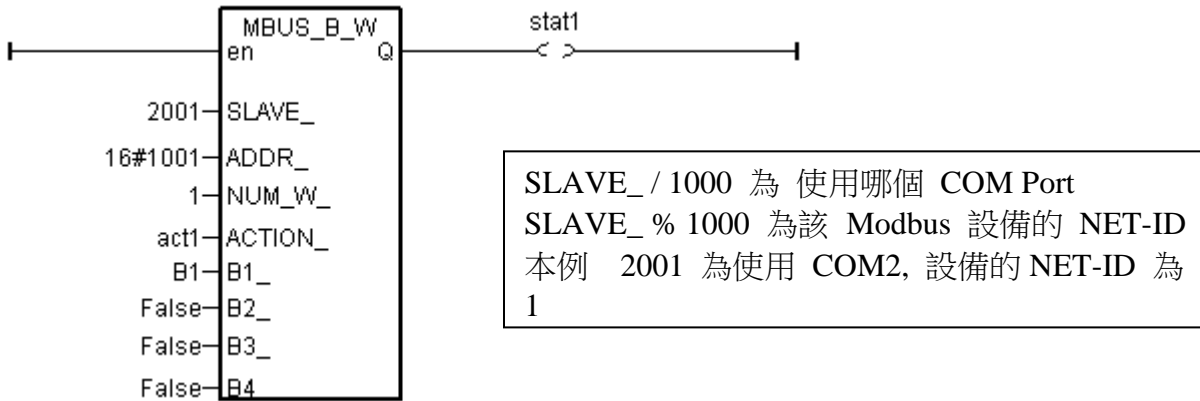
在這個例子中 “Modbus_b_r” 方塊會從 COM3, NET ID 為 1 的 Modbus 裝置讀取 8 個位元， B1 會取得 modbus 網路位址編號 1 的值， B2 會取得 modbus 網路位址編號 2 的值， B5 會取得 modbus 網路位址編號 5 的值。



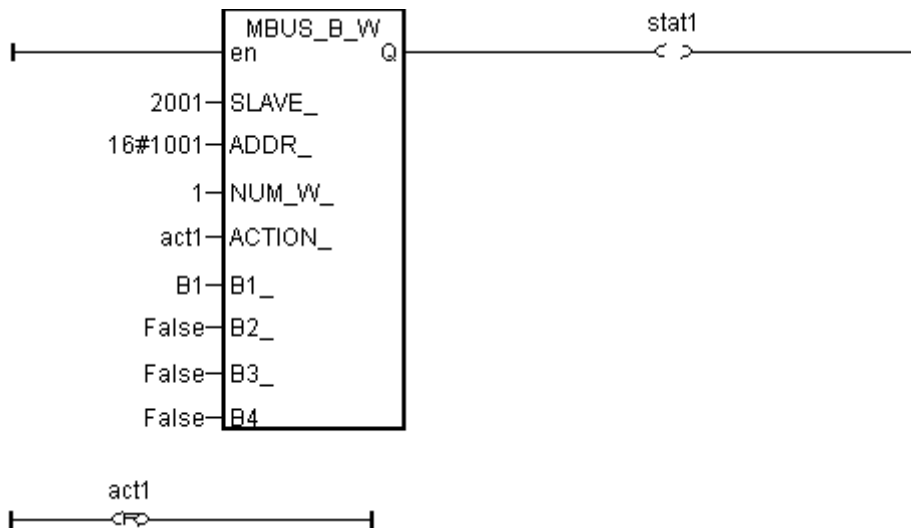
Modbus 範例函式 #2: "Mbus_b_w"

在這個例子中“Mbus_b_w”方塊會寫出 1 個位元到 COM2, NET ID 為 1 的 Modbus 裝置。且“Mbus_b_w”只有在“ACTION_”為 TURE 時，才會寫出位元。在下面的例子中，當 act1 為 TRUE 時，B1 會寫出到 Modbus 網路位址編號為 16#1001(或 4097)的地方。

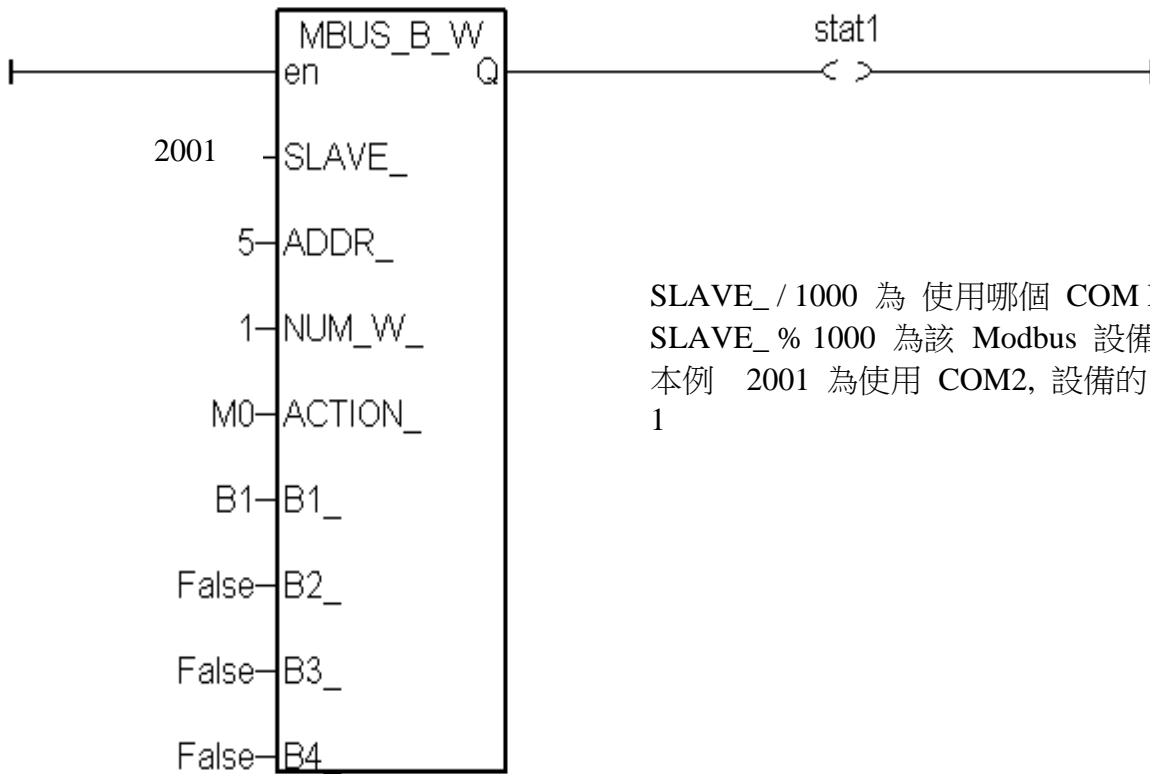
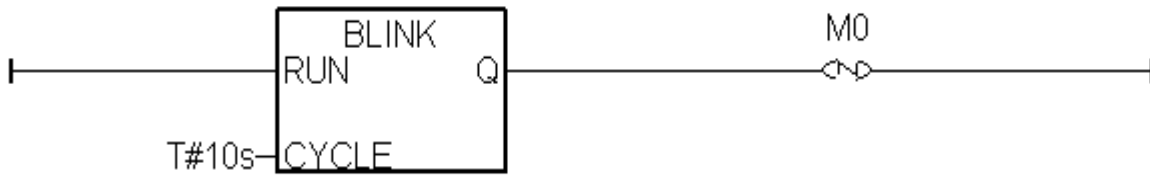
當成功的連接到 Modbus 裝置時， stat1 為 TRUE，否則 stat1 為 FALSE。



若“ACTION_”一直保持為 TRUE，則會將 B1 一直寫出到 Modbus 裝置，直到“ACTION_”為 FALSE。若您只想讓 B1 只寫出一次到 Modbus 裝置內，您可以寫一個如下圖所示的 LD 程式，將 M0 宣告為內部布林變數，如此當 act1 由 False 上升為 True 時，只會寫一次出去。



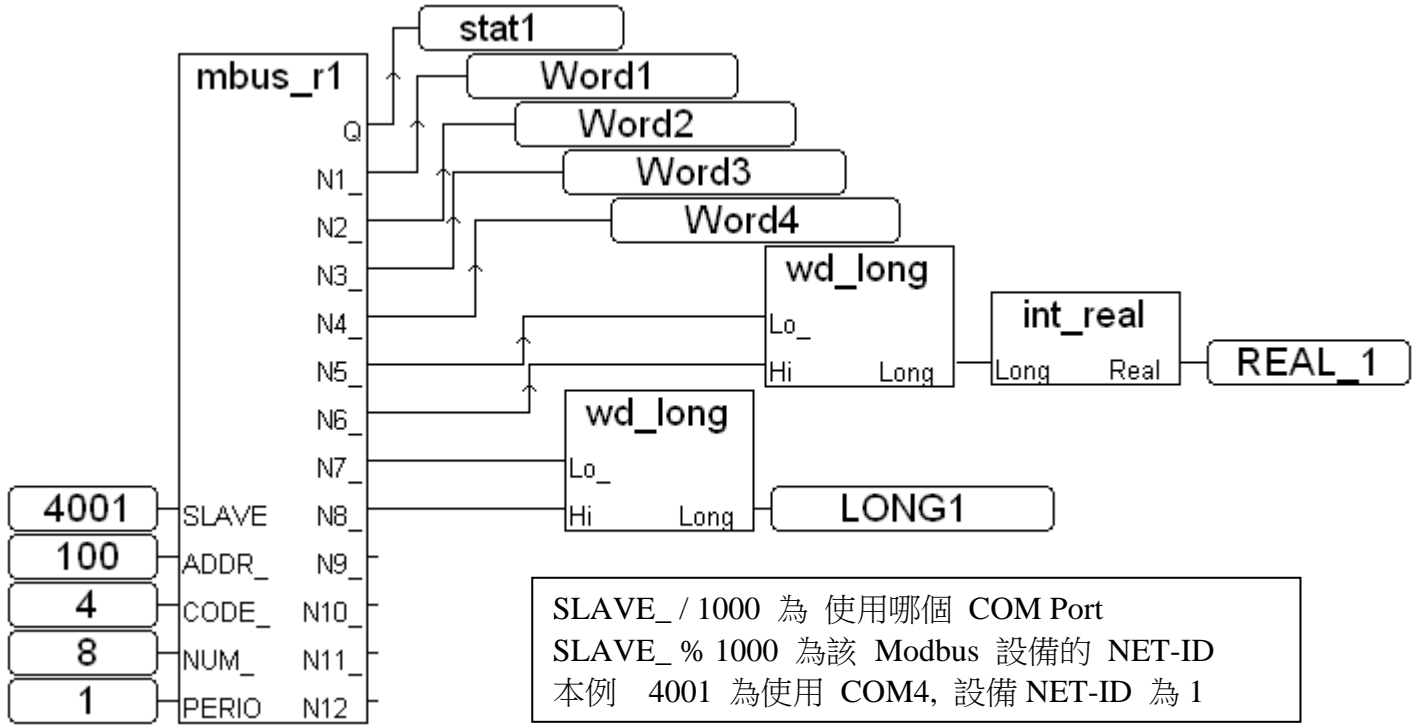
或者 如果想每過一段時間 就寫出一次到 Modbus 設備, 可使用以下方式 (此例每過 10 秒寫一次 B1 值 到 COM2, Modbus 設備 NET-ID 為 1 的 第 5 號 Modbus 網路位址內)



SLAVE_ / 1000 為 使用哪個 COM Port
 SLAVE_ % 1000 為該 Modbus 設備的 NET-ID
 本例 2001 為使用 COM2, 設備的 NET-ID 為 1

Modbus 範例程式 #3: "Mbus_r1"

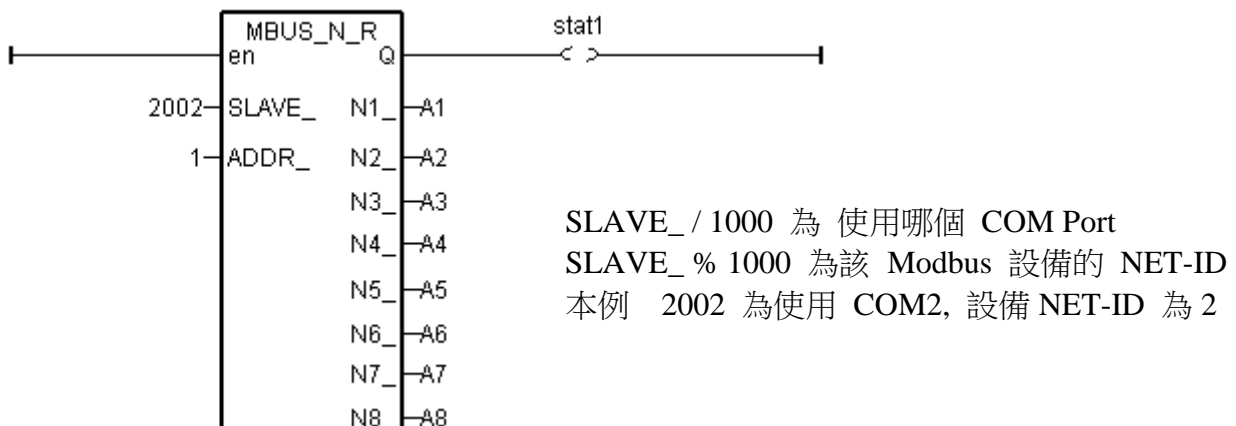
此例使用 Function Block 程式, 用 Modbus function 編號 4 從 COM4, NET-ID 為 1 的 Modbus 設備的 Modbus 網路位址 100 起, 每過 1 秒讀出 8 個 Word 資料, 前 4 個 Word 會存入 Word1 ~ Word4, 第 5 & 6 個 word 會轉換成 REAL 值, 最後 2 個 Word 用 "WD_LONG" 方塊轉換成一個 32-bit 整數。



Modbus 範例程式 #4: "Mbus_n_r"

在這個例子中 "Modbus_n_r" 方塊會從 COM2, NET ID 為 2 的 Modbus 裝置讀取 8 個字組 (Signed Word)。在這個例子中 A1 會取得 modbus 網路位址編號為 1 的值, A2 會取得 modbus 網路位址編號為 2 的值, ..., A8 會取得 modbus 網路位址編號為 8 的值。

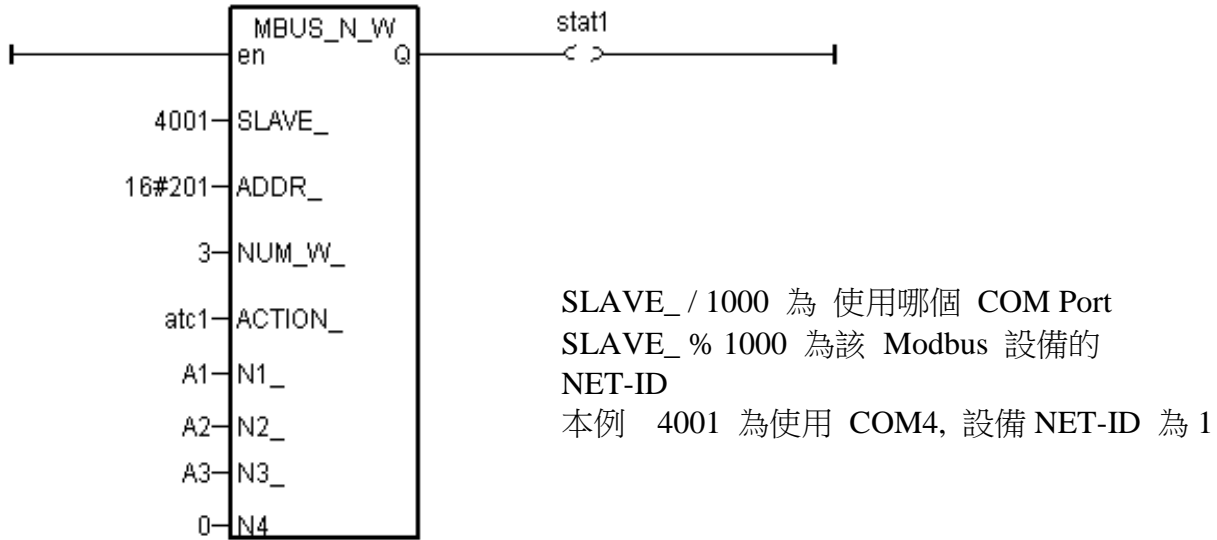
當成功的連接到 Modbus 裝置, stat1 為 TRUE, 否則 stat1 為 FALSE。



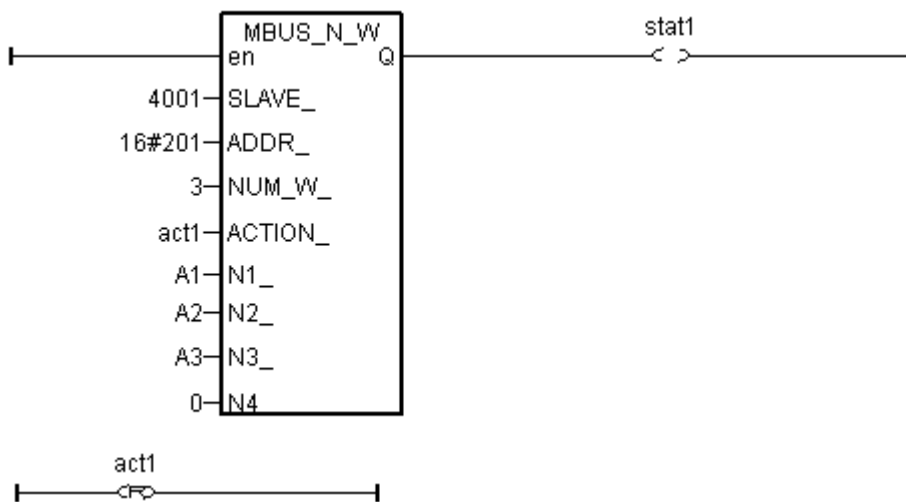
Modbus 範例程式 #5: "Mbus_n_w"

在這個例子中“Modbus_n_w”方塊會寫出 3 個字組(Signed Word)到 COM4, NET ID 為 1 的 Modbus 裝置。在這個例子中 modbus 的網路位址編號從 16#201 開始，當“ACTION_”為 TRUE 時，A1 會寫出到 modbus 網路位址為 16#201 的地方，A2 會寫出到 modbus 網路位址為 16#202 的地方，A3 會寫出到 modbus 網路位址為 16#203 的地方。

當成功的連接到 Modbus 裝置，stat1 為 TRUE，否則 stat1 為 FALSE。



若“ACTION_”一直保持為 TRUE，則會將 A1 到 A3 的值一直寫出到 Modbus 裝置，直到“ACTION_”為 FALSE。若您只想讓變數只寫出一次到 modbus 裝置，您可以寫一個如下圖所示的 LD 程式，將 M0 宣告為內部布林變數，當 act1 由 False 上升為 True，只會寫出一次



更多的 Modbus Master 使用說明可訪問

http://www.icpdas.com/faq/isagraf_c.htm > FAQ-144, 113, 101, 096, 075, 047, 046, 027, 028, 045

8.3: 連接 M-7000 I/O 模塊

ICP DAS 的 M-7000 系列 I/O 模塊 支持 Modbus RS-485 RTU 通信規約. ISaGRAF PAC 可以 啟用 Modbus RTU Master 功能來與它們相連接. 詳細內容 與 demo 程序, 請參考第 21 章 與 http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm

(或 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-050)

注意: 每個 ISaGRAF PAC 的 RS-485 通訊串口, 最多只能連 32 顆 M-7000. 或要接更多, 必需搭配 RS-485 repeater (I-7510) 來使用

8.4: 連接 EKAN-Modview LED 顯示器

ICP DAS 的 EKAN-Modview LED 顯示器的 RS-485 串口 支持 Modbus RTU 通信規約. ISaGRAF PAC 可以啟用 Modbus RTU Master 功能來與它們相連接, 來控制所要顯示的資料 訊息. 詳細內容與 demo 程序 請訪問

www.icpdas.com – FAQ – Software – ISaGRAF – FAQ045 與

<http://www.icpdas.com/products/HMI/led/ekan.htm>



第 9 章 幾個 ISaGRAF 常用的功能介紹

注意：

本書內的出現的 **I-8xx7** 是 I-8417、 I-8437、 I-8817 和 I-8837 控制器的縮寫.

而 **WP-8xx7** 是 WP-8147/8447/8847 及 WP-8137/8437/8837 的縮寫.

而 **XP-8xx7-CE6** 是 XP-8047-CE6 / XP-8347-CE6 / XP-8747-CE6 的縮寫.

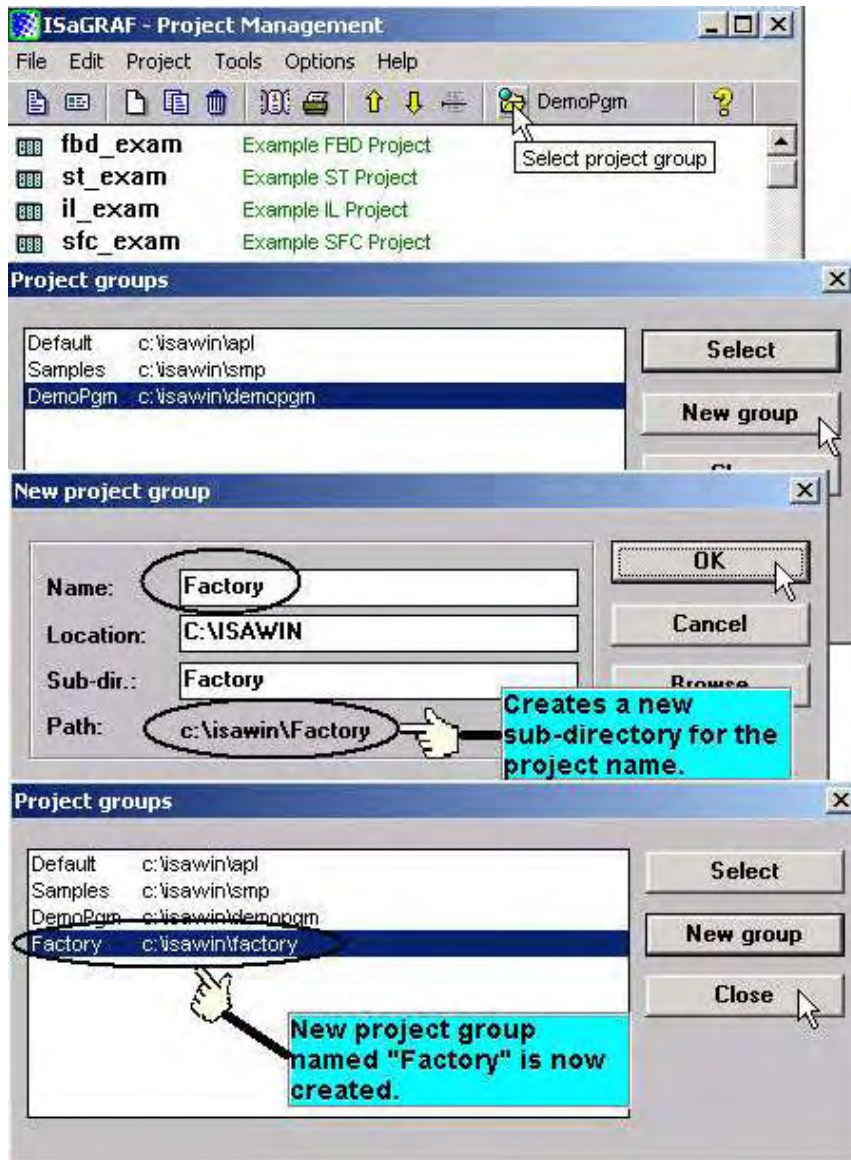
而 **iP-8xx7** 是 iP-8447 / iP-8847 的縮寫.

在下面的章節中我們將描述 ISaGRAF Workbench 程式整合環境的一些有用的功能，這些功能可以使開發 ISaGRAF 程式更為容易管理。

在本章中我們不探討 ISaGRAF 的程式設計技巧，因為這些資訊，均可在 ISaGRAF Workbench 的 CD ROM 內找到，檔名為 ISaGRAF.pdf 或 ISaGRAF.doc。

9.1: 建立 ISaGRAF 的專案群組

ISaGRAF 可分成數個群組來管理不同類型或不同使用者的專案，如此可讓工作案件分門別類，方便管理。

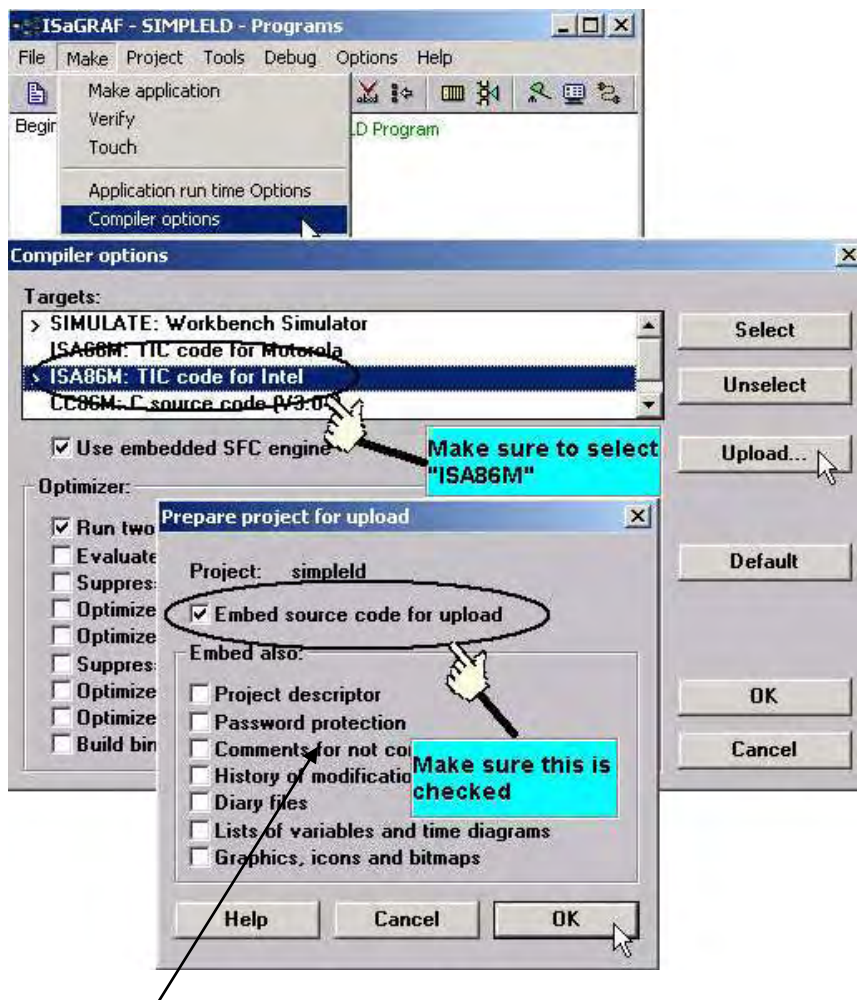


若你想要刪除已存在的專案群組，在 Windows 下將此專案群組的目錄路徑刪除即可，例如 C:\isawin\factory 目錄。

9.2: 上載 ISaGRAF 專案

ISaGRAF 提供程式碼上載的功能 (由於啟動上載功能會使程式碼變大 1.5 ~ 3 倍, 在小容量的 Controller 如 I-7188EG/XG, I-8xx7 上,建議儘量不要使用), 可將原先已經在控制器內執行的程式上載到 ISaGRAF 內, 但在上載之前您必須先作一些設定。

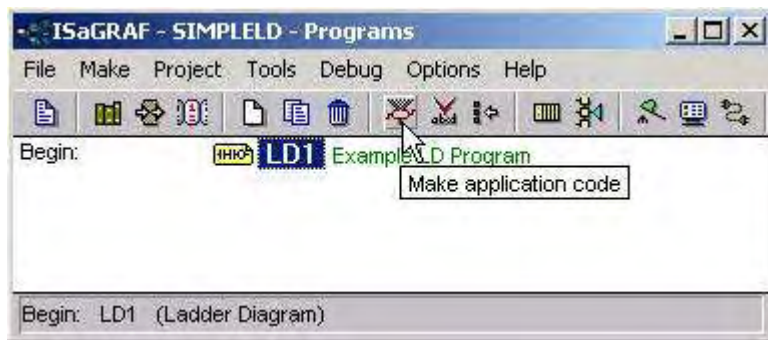
在專案內點選 “Compiler Option”, 選取 “ISA86M: TIC Code For Intel”, 單擊 “Upload” 鍵, “Prepare Project For Upload” 視窗會被開啟, 選擇 “Embed Souce Code For Upload”, 接著按下 OK 鍵。



注意:

若您有使用 ”直接表示變數” 則 “Comments for not connected I/O channels” 選項必須被選取 (請參考 3.4 節)

在您確定有勾選“Embed Source Code For Upload”後，按下 OK 鍵，該專案必須重新編譯過，再下載至控制器中。



注意:

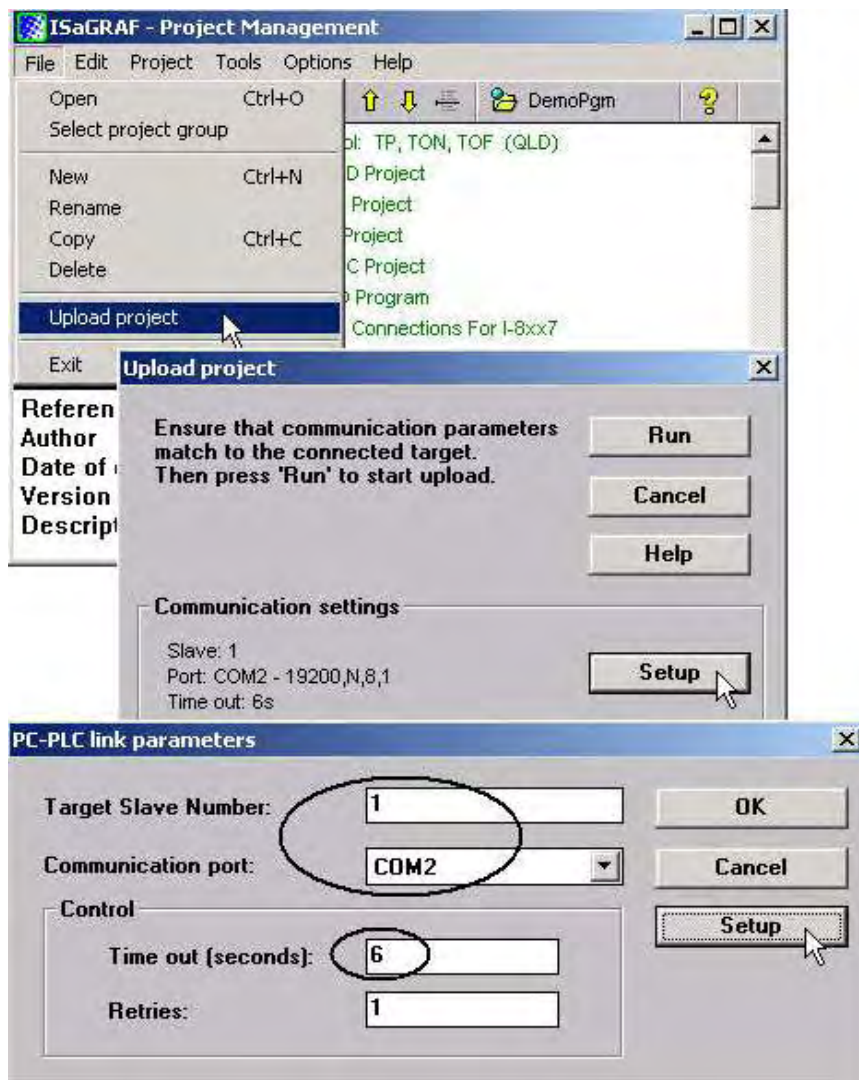
一旦您完成了”上載”的設定後，編譯出來的程式碼會比原先未設定過”上載”的原始程式碼還大。假如這可上載的程式碼大於 64K bytes，您將無法把這些程式碼下載到 I-8xx7, iP-8xx7, I-7188EG/XG, μ PAC-7186EG, VP-2117 控制器中 (Wincon-8xx7 可接受最大 512K bytes 的程式碼, WP-8xx7, VP-25W7/23W7 是 1M bytes, XP-8xx7-CE6 是 2M bytes)。

在您開始下載程式之前，建議您先檢查一下要下載的程式碼大小，找出程式碼所放置的子目錄位置。若是“SIMPLELD”編譯出來的程式碼，其程式碼是放在 c:\isawin\demopgm\simpleld 目錄下。

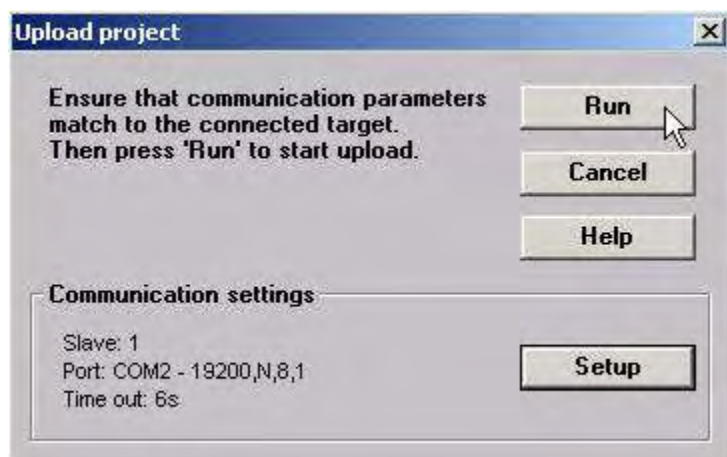
“DEMOPGM” 即為該專案的群組名稱，而”SIMPLELD” 則為 ”DEMOPGM” 下的專案名稱，而編譯出來的程式碼放在 SIMPLELD 目錄底下，在此目錄下我們可以找到“APPLI.X8M”這個檔案，確定他的檔案大小。“APPLIC.X8M” 包含上載或下載的程式碼，確定用在 I-8xx7, iP-8xx7, I-7188EG/XG, μ PAC-7186EG, VP-2117 上其檔案大小沒有超過 64K (Wincon-8xx7 可接受最大 512K bytes 的程式碼, WP-8xx7, VP-25W7/23W7 是 1M bytes, XP-8xx7-CE6 是 2M bytes)。

上載 ISaGRAF 專案

要從控制器中上載 ISaGRAF 專案到電腦，要開啟[File]→[Upload Project]，在設定之前必須確定您的控制器與電腦之間的連線是否正常，若不正常可透過“Setup”來修改控制器與電腦之間的通訊參數。



若確定您的通訊數設定沒問題，單擊“RUN”來進行專案的上載。

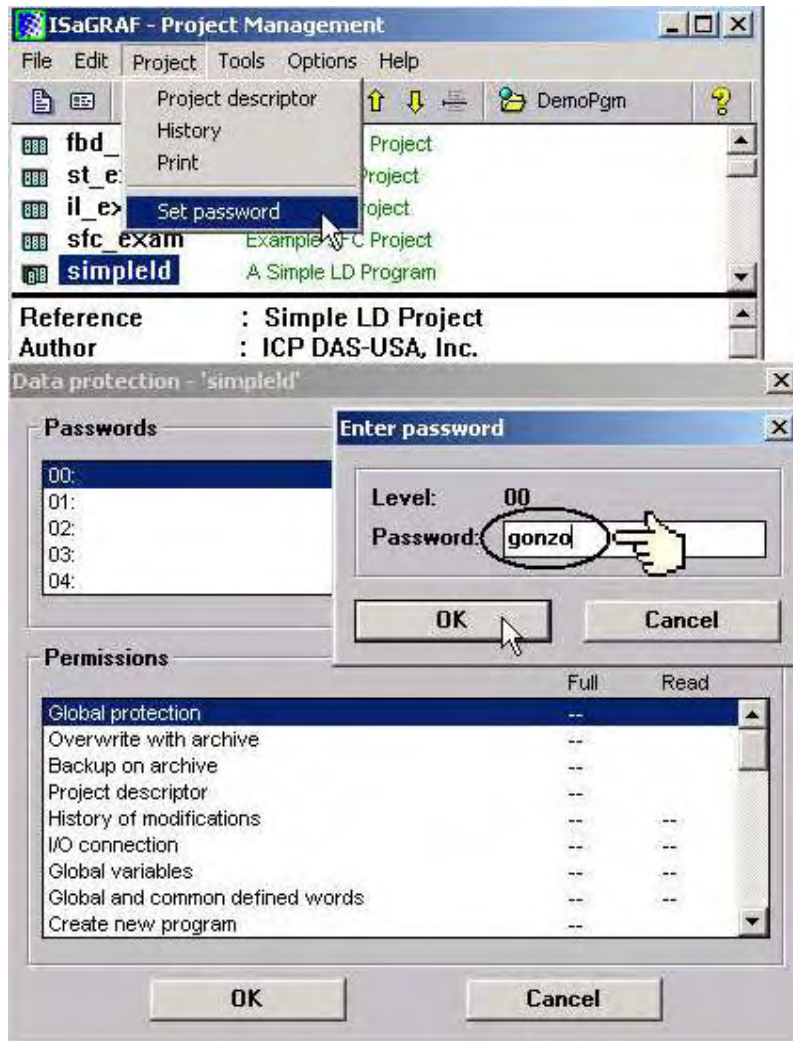


9.3: 設定 ISaGRAF 的密碼管制

ISaGRAF 提供使用者針對每個專案可設定 16 組密碼，可以管制不同的使用權限、且密碼的使用可以保護程式不被他人盜用。

在 “ISaGRAF Project Window” 視窗中選 [Project] → [Set passwd]，將會開啟 “Data Protection” 視窗，密碼共可設 16 組，編號 00~15，任選一組設定其密碼。

您也可以為您的 ISaGRAF 專案設定其保護型態，在下面的例子中，我們為這個專案設定 “Global Protection”。

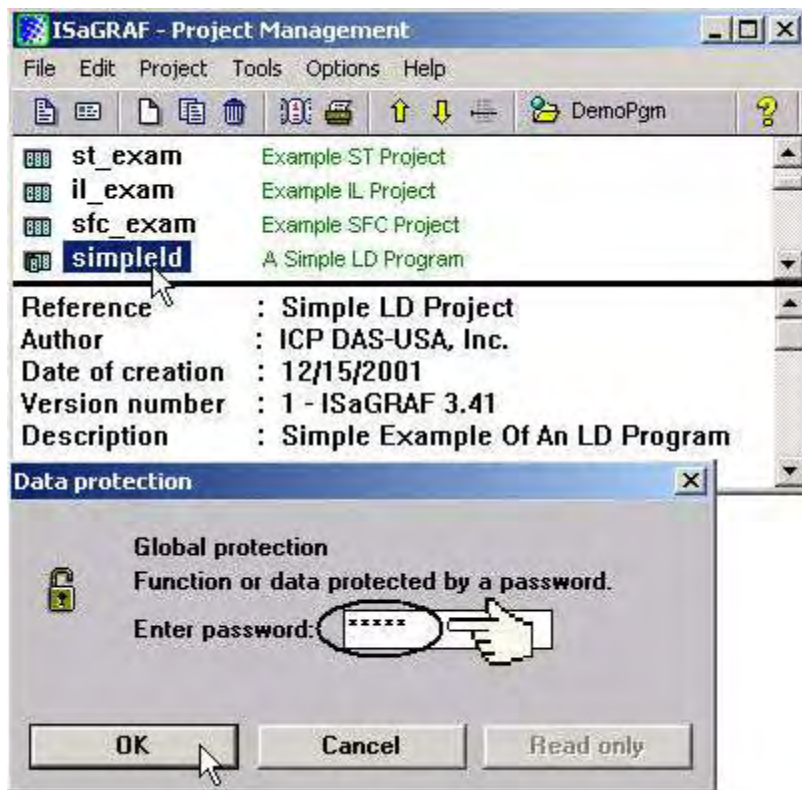


在您輸入密碼按下 OK 鍵後離開，就完成了本專案的密碼設定。

在 ISaGRAF 專案中的 “Permissions” 可以定義保護型態，在 “Data Protection Permission” 中您可以選擇全部存取和輸入密碼來保護您的專案。

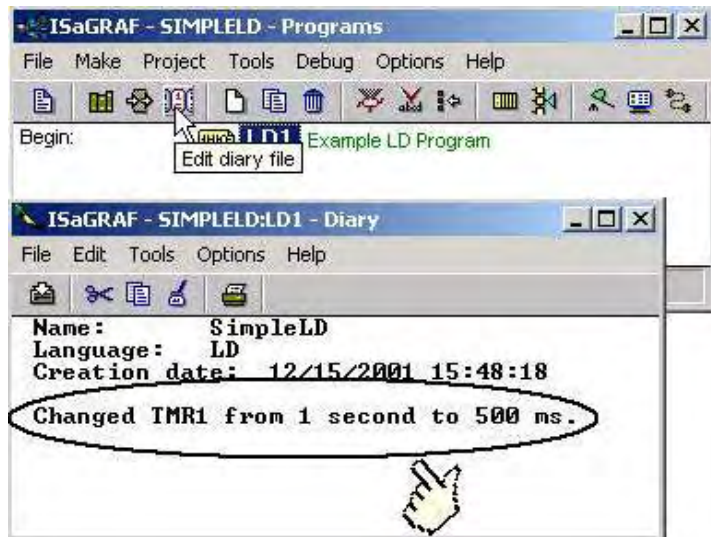


離開 ISaGRAF 後，再由 Windows 重新進入 “ISaGRAF Project Management” 視窗，以滑鼠雙擊設定密碼的專案，你可以發現此時必須設定正確的密碼才可以進去該專案中。



9.4: 建立 ISaGRAF 程式日記

在您每次修改 ISaGRAF 程式時，您可以進入“Edit diary”來為程式編寫程式日記，使用者可以輸入自己的修改記錄在程式日記中，如此可在日後重新進入本案件時，可查閱以前的變更日記。



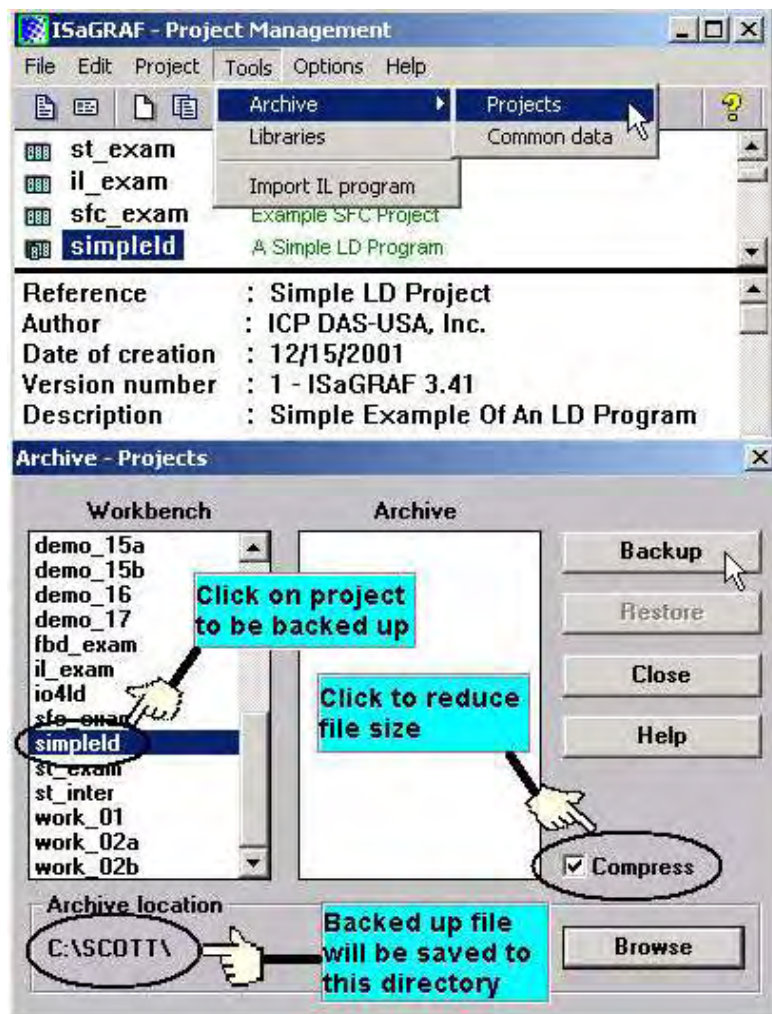
當您完成編輯“ISaGRAF Diary”後，按下“Save”鍵，即可儲存您所編寫的程式日記。

9.5: 程式的備份及回存

使用者可使用 ISaGRAF 提供的程式備份工具，將專案備份到磁碟片上或指定的檔案路徑內。

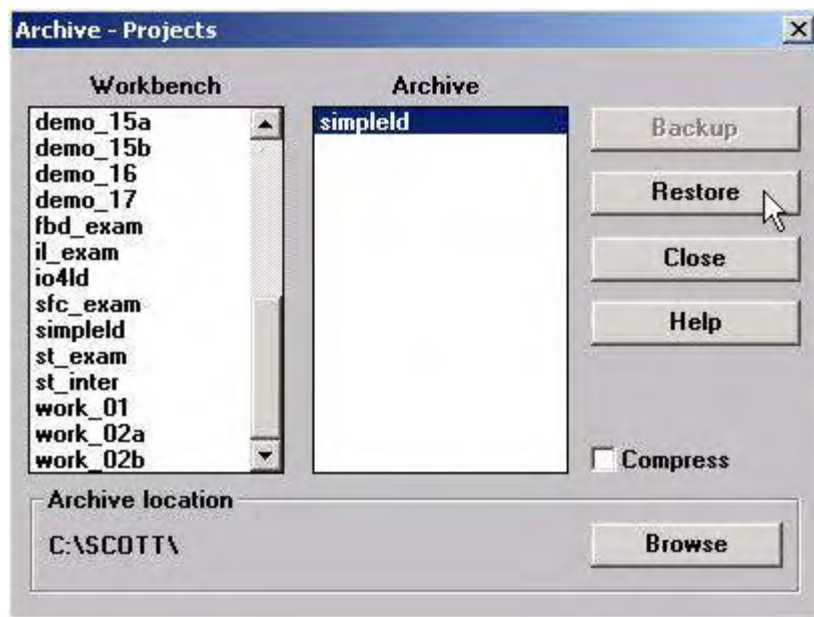
備份 ISaGRAF 專案

在 “ISaGRAF Project Management” 視窗，以滑鼠點選 [Tools] → [Archive] → [Projects]，接下來在 [Archive] 視窗內，以滑鼠點選 “Browse” 所要備份的目錄位置，之後再選取 “Archive” 視窗左方所要備份的專案名稱，按下 “backup” 即可備份到前面所選取的目錄位址。



回存 ISaGRAF 專案

若要從指定的目錄位置回存專案到 ISaGRAF 內則在 “Archive”視窗的右方，以滑鼠選取所要回存的案件名稱，按下 “restore”即可。



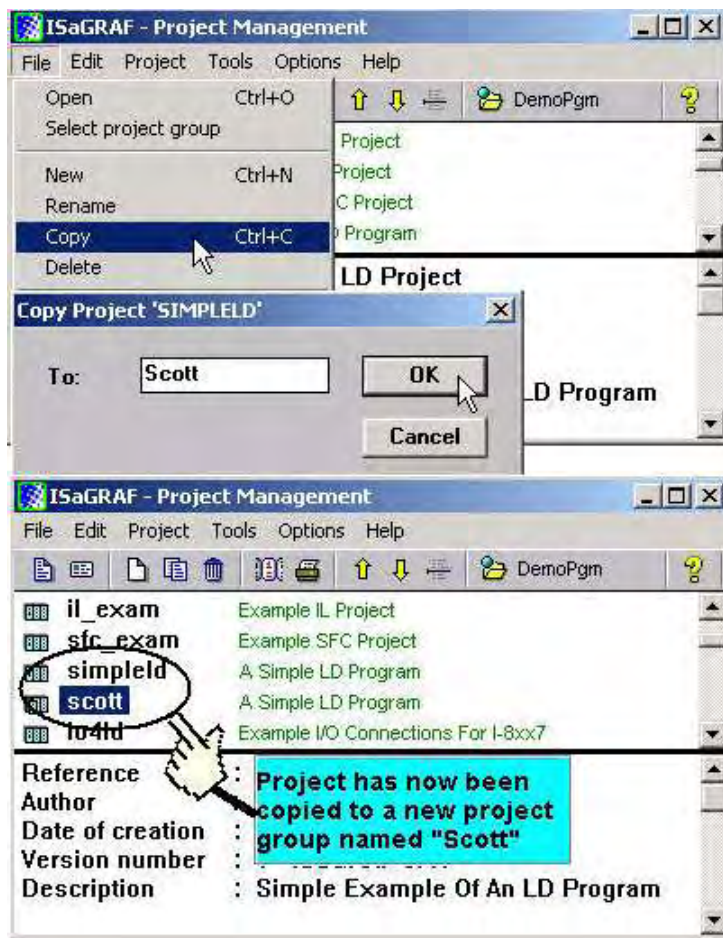
現在您就可以開啟，編輯或下載這個回存的 ISaGRAF 檔案。

9.6: 專案及程式的複製

ISaGRAF 具有複製和更改專案名稱或程式的功能，這對使用者在維護或複製程式到另一個子目錄是非常方便的。

複製 ISaGRAF 程式

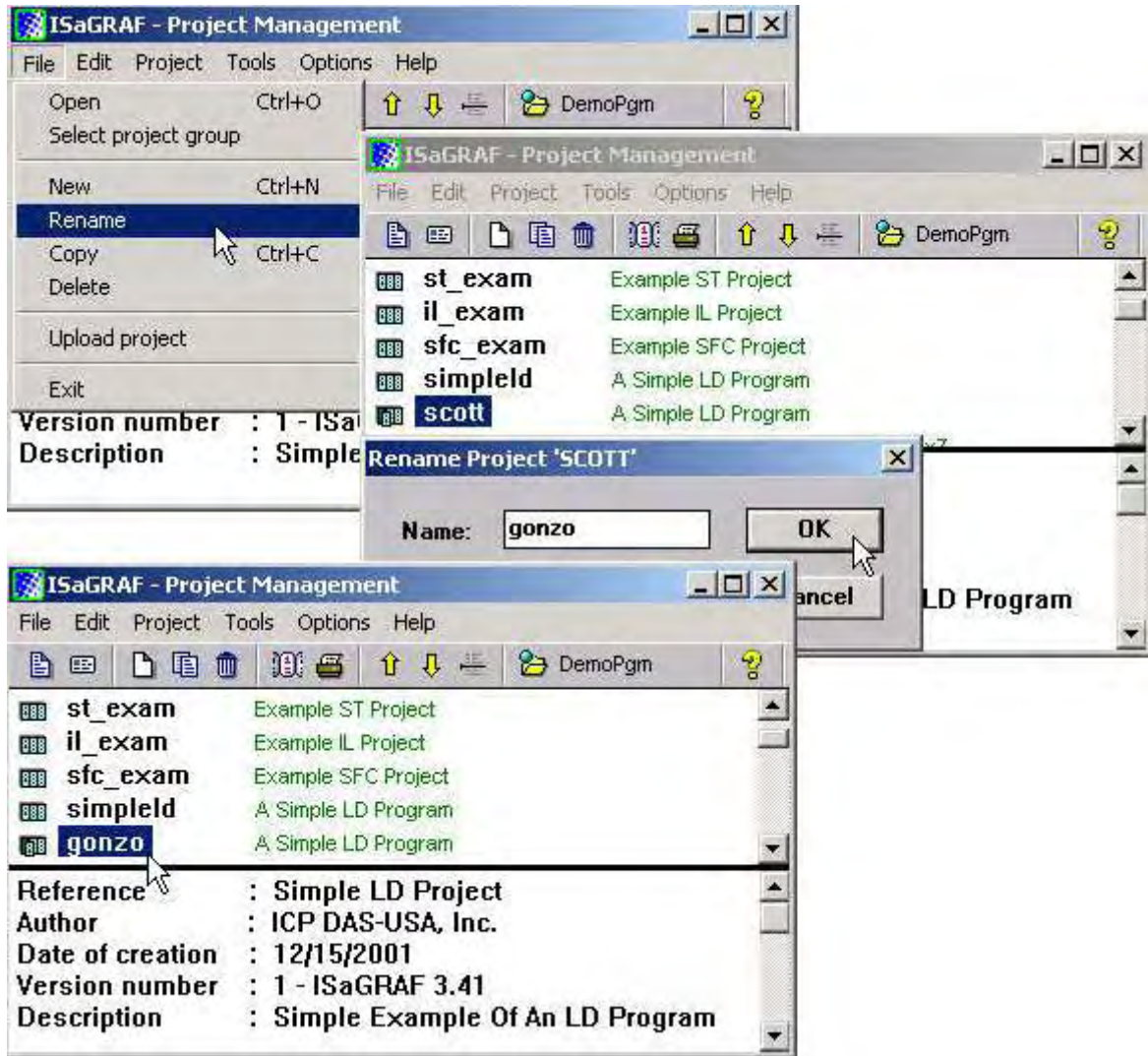
在“ISaGRAF Project Management”視窗，以滑鼠點選所要複製的專案，之後在點選“Copy Project”，輸入欲複製成的案件名稱後，按下“OK”鍵即可。



在上面的圖示中，我們將“simpleld”複製到“Scott”專案中。

變更 ISaGRAF 專案名稱

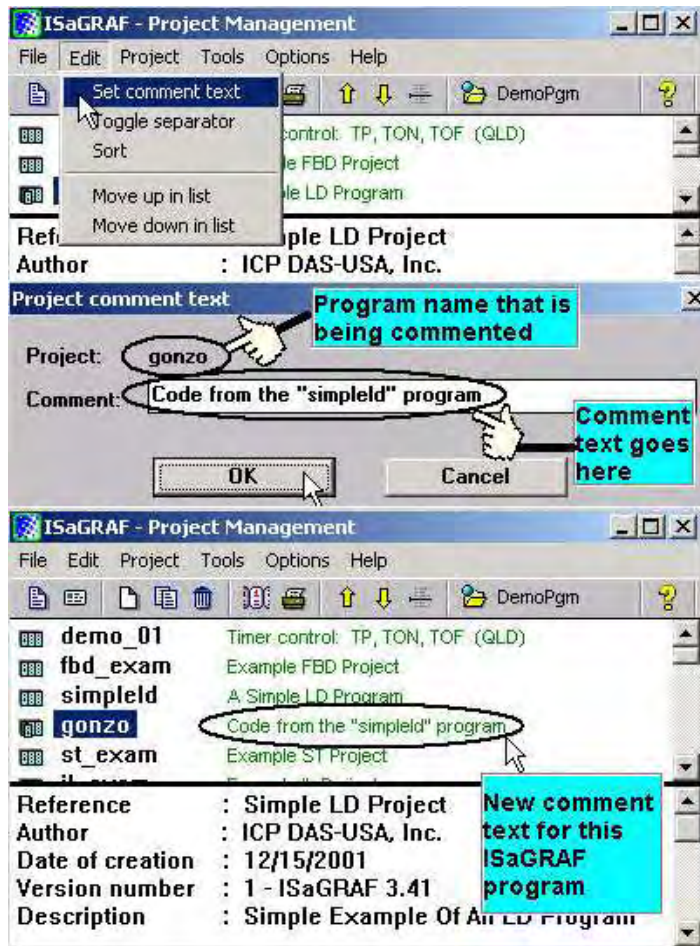
在“ISaGRAF Project Management”視窗，以滑鼠點選所要更改名字的專案，接著以滑鼠點選 [File]→[Rename]來開啟[rename]視窗，您只要輸入新的名稱，按下 OK 鍵即完成更改專案名稱。



之前的“scott”專案已改變名稱為“gonzo”，而“simpleld”專案還是依然沒有任何更動。

9.7: 為 ISaGRAF 專案增加註解

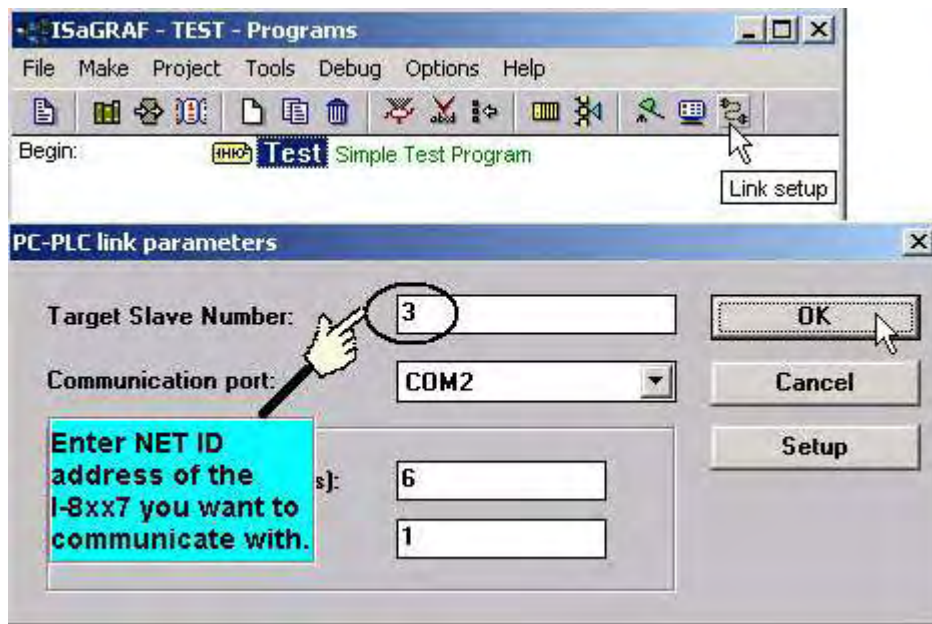
在“ISaGRAF Project management”視窗中的[Edit]→[Set comment text]選項，可以幫每一個專案加上註解，讓您可以一目了然的知道每個專案的作用。



9.8: 設定所要連接的控制器的位址編號

ISaGRAF 可同時連結多台控制器進程式上，下載或連結偵錯，每台被連結的控制器必須要有不同的位址編號(NET-ID)，如此 ISaGRAF 才能分辨要和那一台連線。假如只有一台控制器，其預設的位址編號為 1。

當您按下 “Link Setup”時，將會開啟 “PC-PLC Link Parameters”，在 “Target Slave Number”輸入你欲連接的控制器的位址編號。

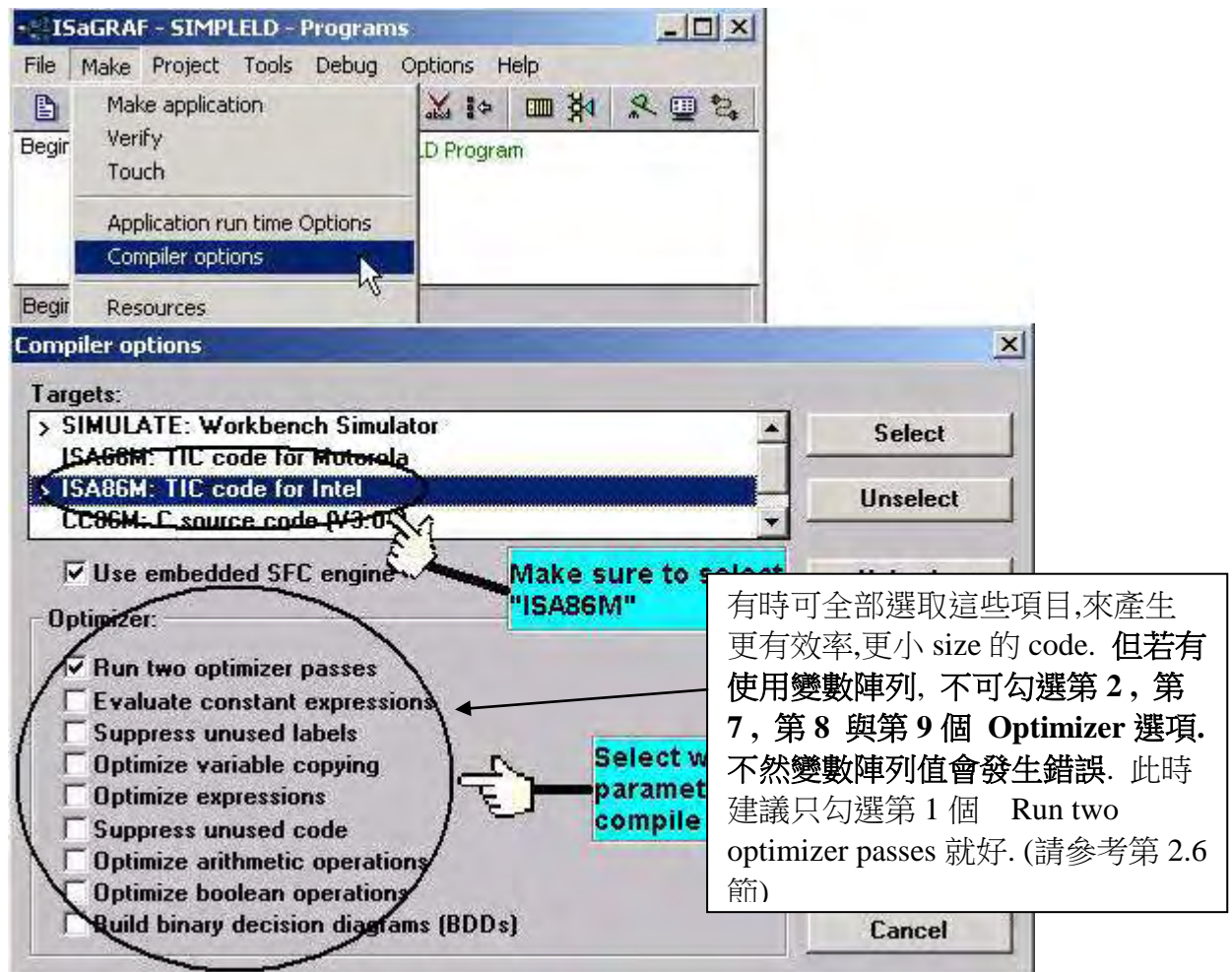


注意:

請參閱各自的 “快速上手手冊”來設定控制器的 NET-ID .

9.9: 程式編譯的最佳化選項

你可以在 ISaGRAF 的專案內，點選 [Make]→[Compiler option]選項，來編譯(compile)出執行效率更好的程式碼，但會付出較長的編譯時間。



9.10: ISaGRAF 的類比轉換表

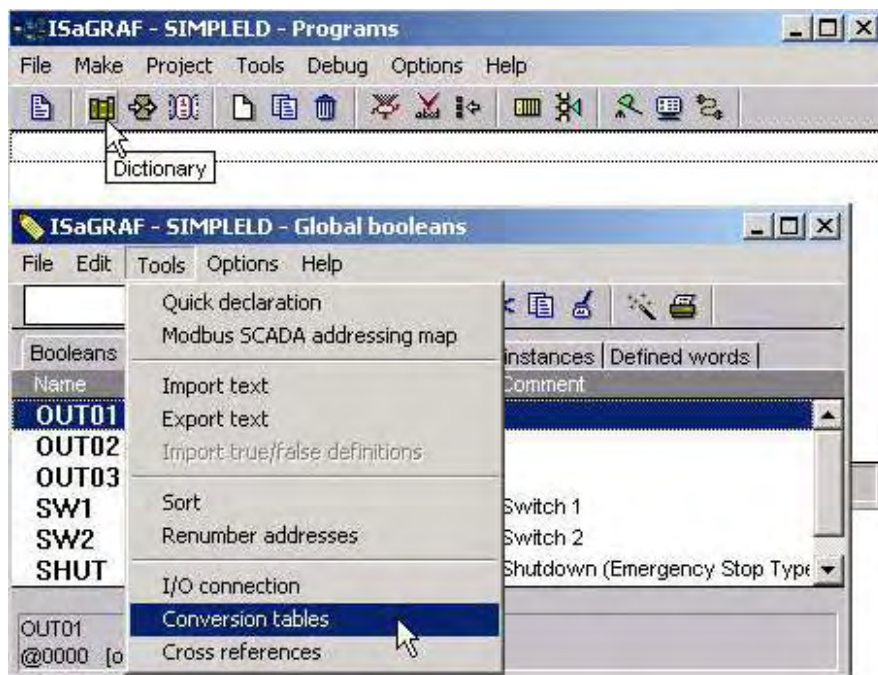
通常在讀取類比 I/O 接點的數值，例如 I-87017 類比輸入模組，其傳回的數值是由-32768 到 32767，-32768 可能表示-10V 而 32767 可能表示+10V，如果有一個輸入值是 2573，請問你能馬上知道它是幾 V 嗎？

ISaGRAF 提供轉換表的功能，可讓使用者將電子的數值轉換成你想要的數值，這兩者的關係必需是線性的，其方法為定義一些轉換點，至少要有兩點，如此就可決定出所要的轉換表。例如轉換前 -32768 代表-10V，32767 代表+10V，而轉換後-10000 將代表-10V，而 10000 將代表 10V，如此就比較好理解了，像 2573 就馬上知到為 $2573\text{mV}=2.573\text{V}$ 。

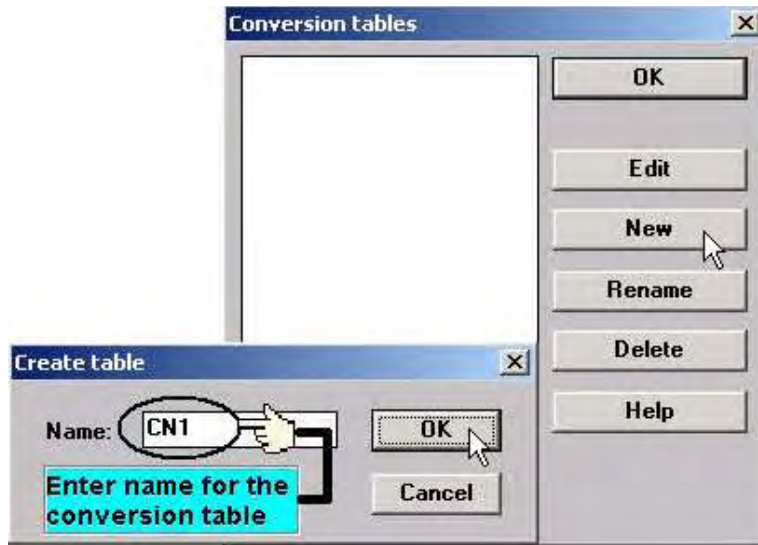
注意：

ISaGRAF 控制器在轉換前只支援 -32768 到 32767 之間的數值，而轉換後只支援-10000 到 10000 之間的數值，所以您在設定轉換表時若設超過這些範圍，其轉換值可能會出錯。

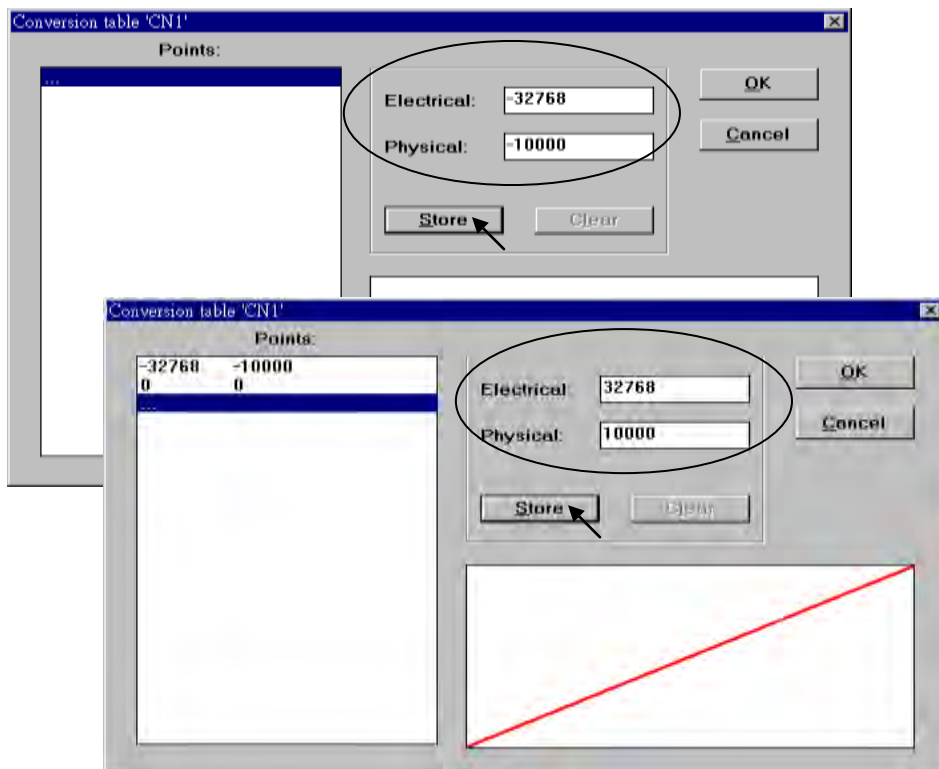
單擊 [ISaGRAF Programs]→[Dictionary]，將開啟 “ISaGRAF Global Variable”視窗，選擇 [Tools]→[Conversion Tables]即可製作你所想要的轉換表。

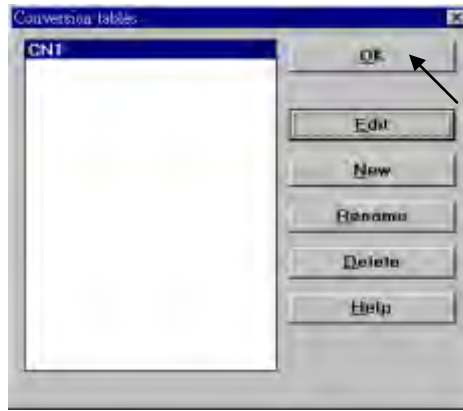


進入 “Conversion Tables”後，選取新增表格並取一個名字。

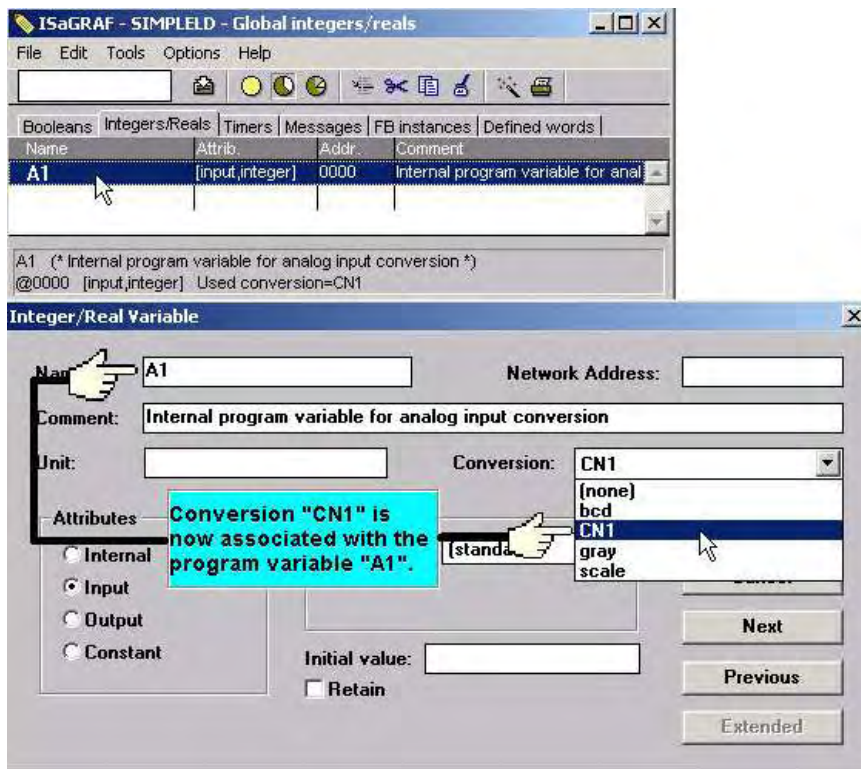


定義出轉換表，至少要定出 2 點，每定出 1 點就按 “store”，全部定完後按 “OK”。





之後又會回到“conversion table”視窗，按下“OK”，如此即完成一個轉換表的建立。接下來在需要被轉換的變數內，選取該類比變數所要套用的轉換表即可。

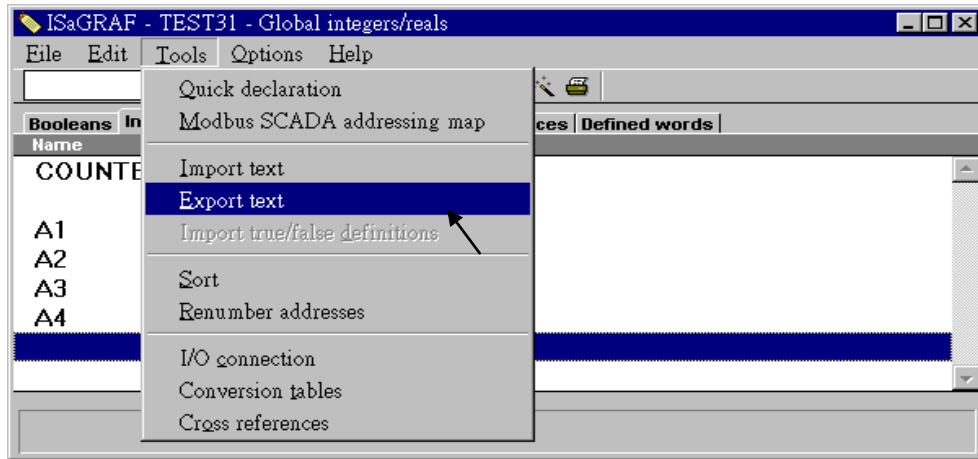


注意：

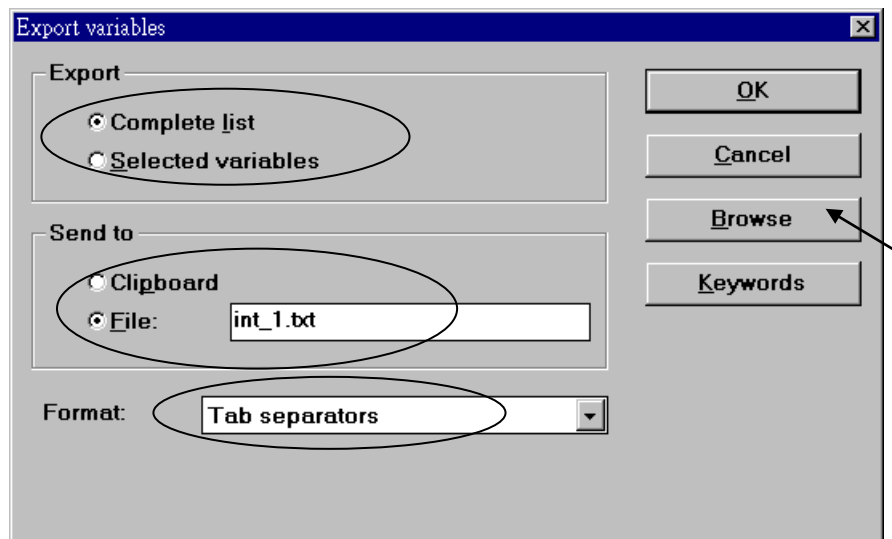
只有宣告為輸入或輸出屬性的整數變數才可以使用轉換表的功能。若要轉換 內部(internal) 屬性的變數 可使用“Bin2Eng”(參閱附錄 A.4)

9.11: 透過 Microsoft Excel 引用／匯出變數宣告

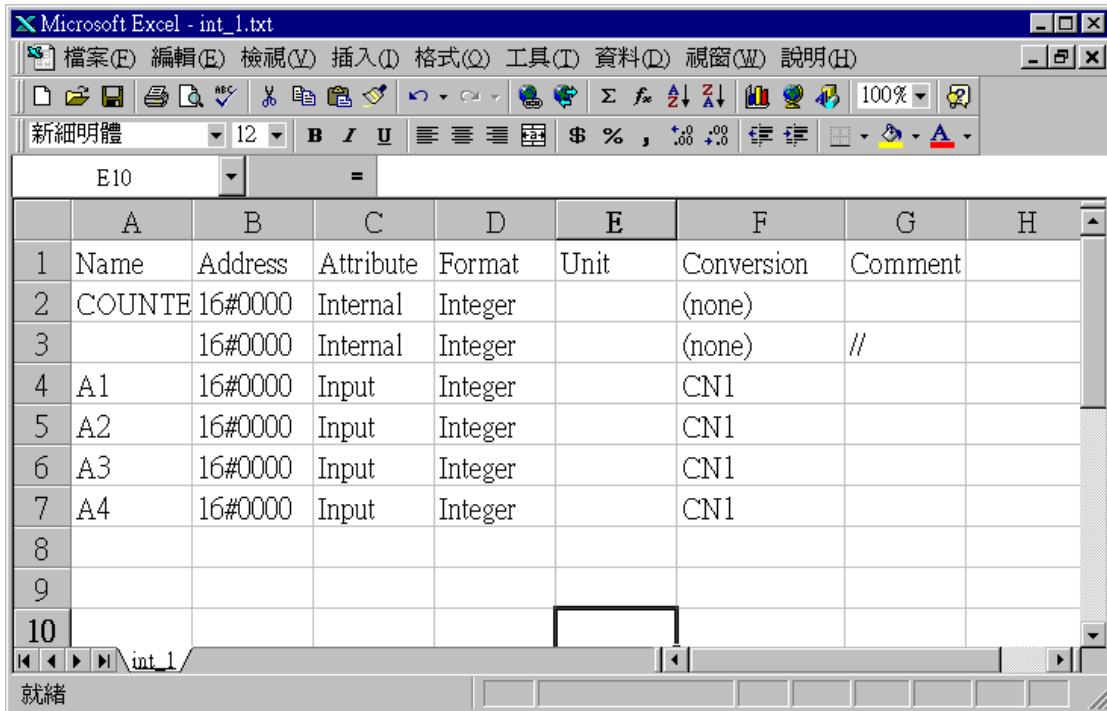
變數可定義在 Microsoft Excel 內，然後引入到 ISaGRAF，也可從 ISaGRAF 匯出到 Excel 內。從 [dictionary]→[Tools]→[Export text]可將 ISaGRAF 內所定義的變數資料匯出成文字檔



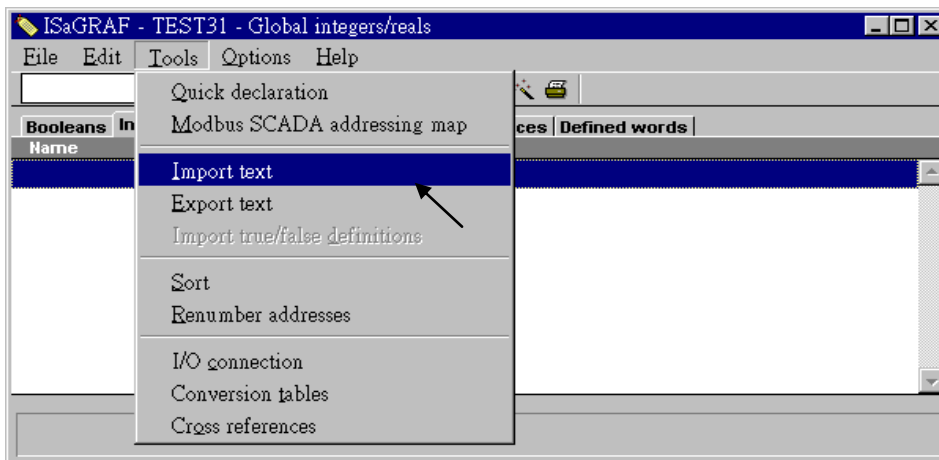
在這個例子中，我們將要存的資料存成 “int_1.txt”檔，然後按下 “Browse”選擇您所要存放的目錄。



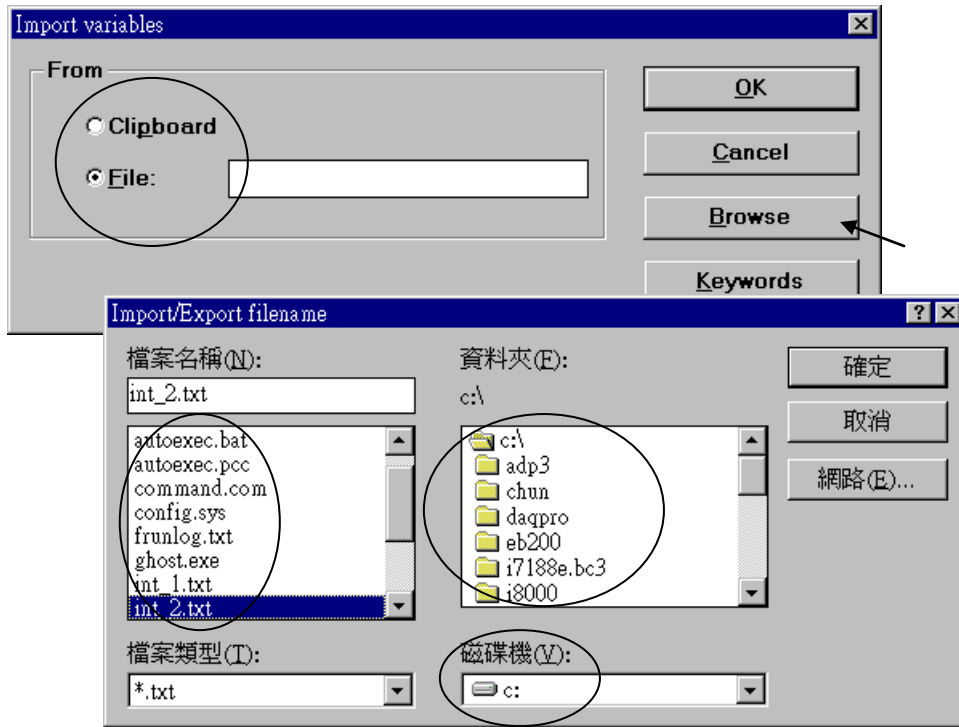
您可以在 Excel 內編輯 “int_1.txt”檔。



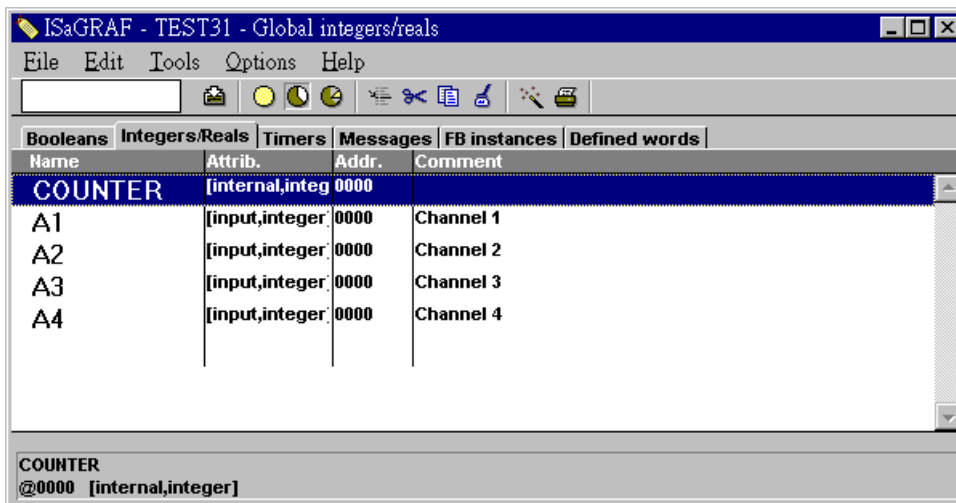
選取[[dictionary]→[Tools]→[Import text]選項，可將 “.txt”檔給引進 ISaGRAF 程式。



單擊 “Browser”來選取適當的檔案。



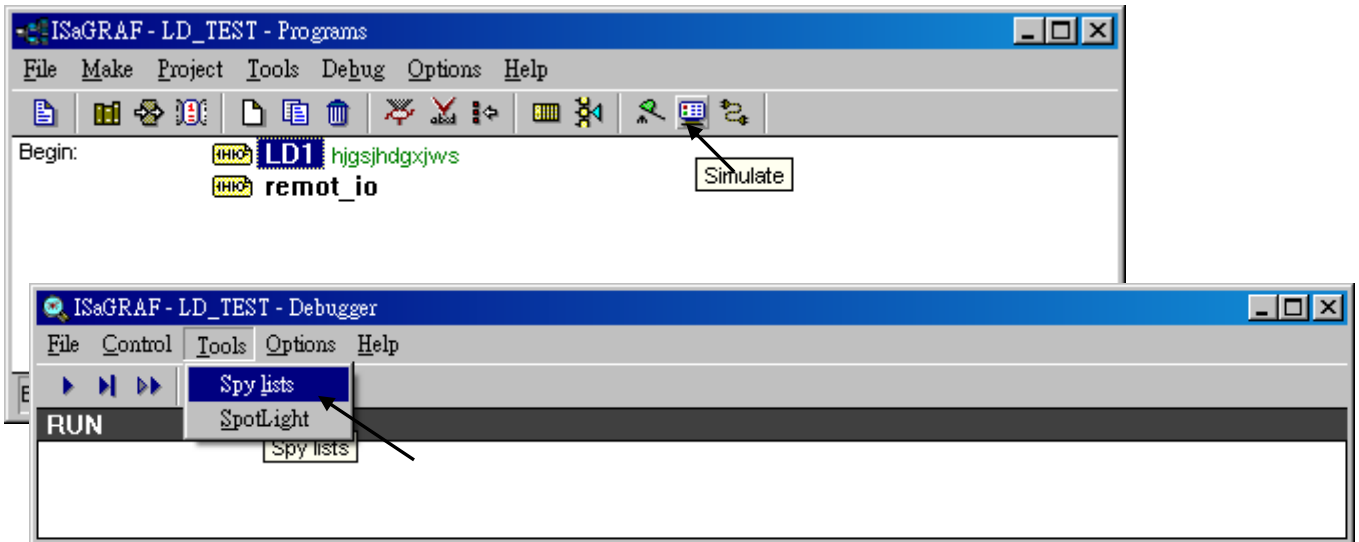
結果如下所示。



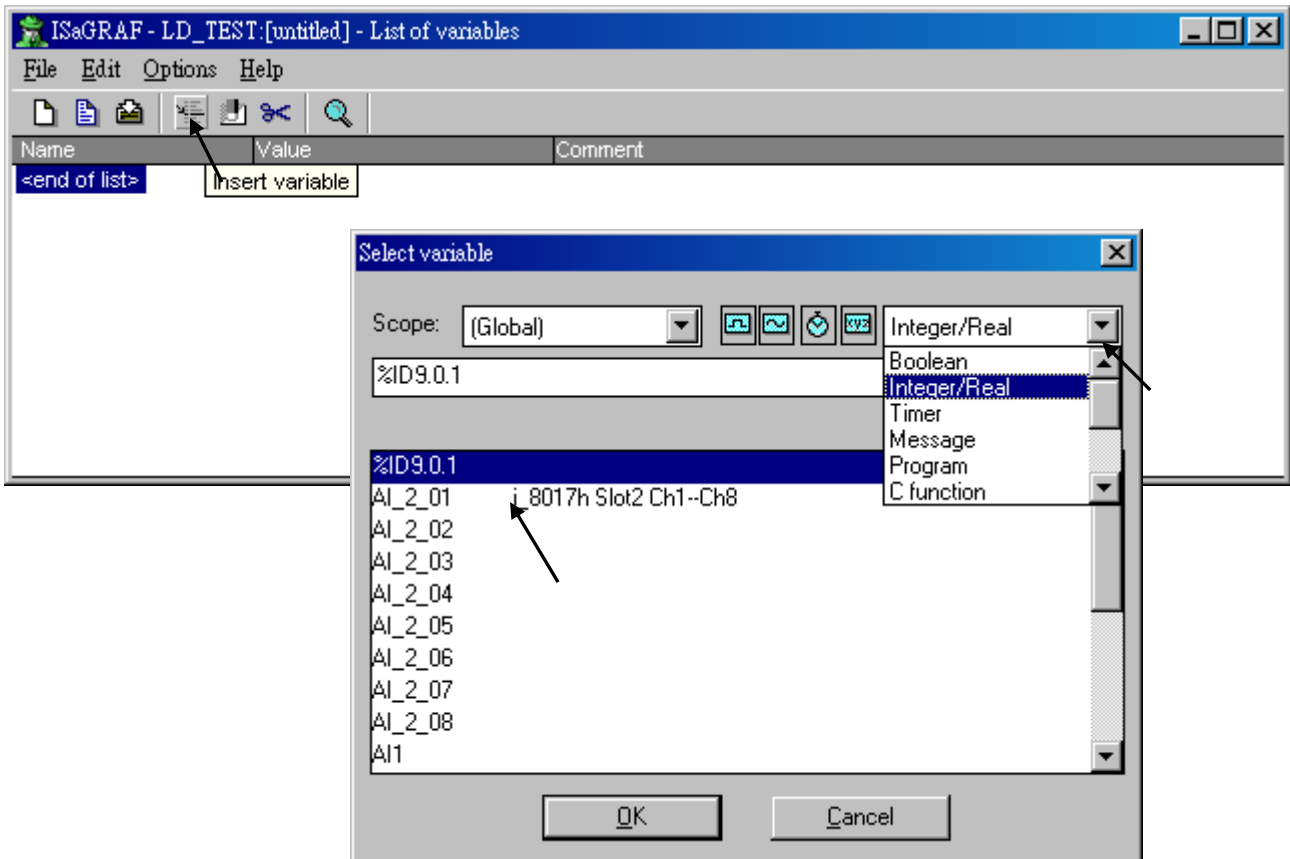
9.12: 執行期間顯示選定的變數資料

有時想在 ISaGRAF 與控制器連線時可以顯示某些特定的變數資料 以利觀看，可以參照以下的步驟建立 “Spy List”。

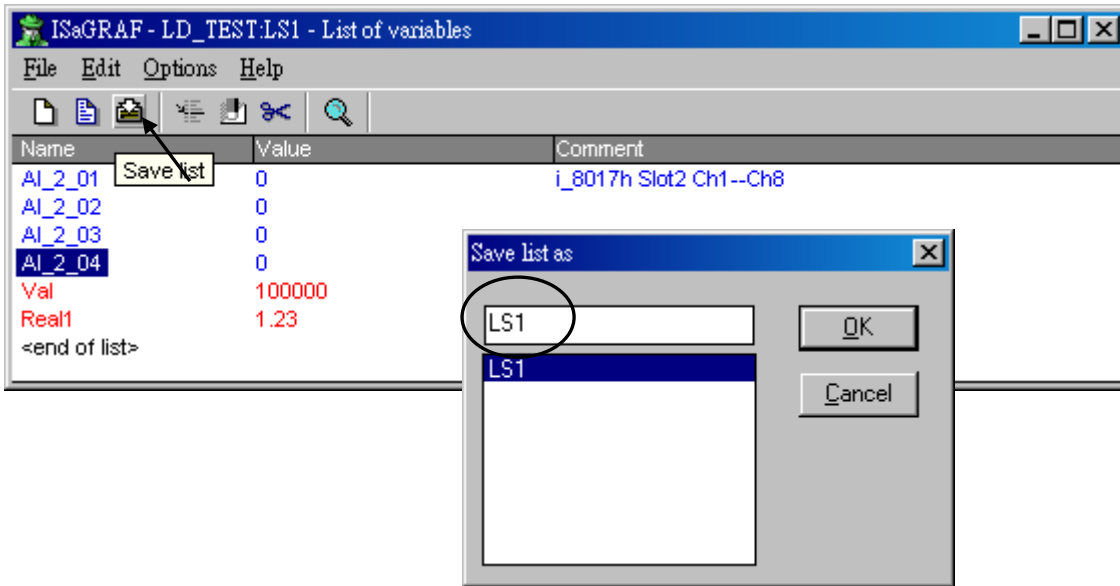
首先點選 “Simulate”，之後執行 “Tools – Spy list”。



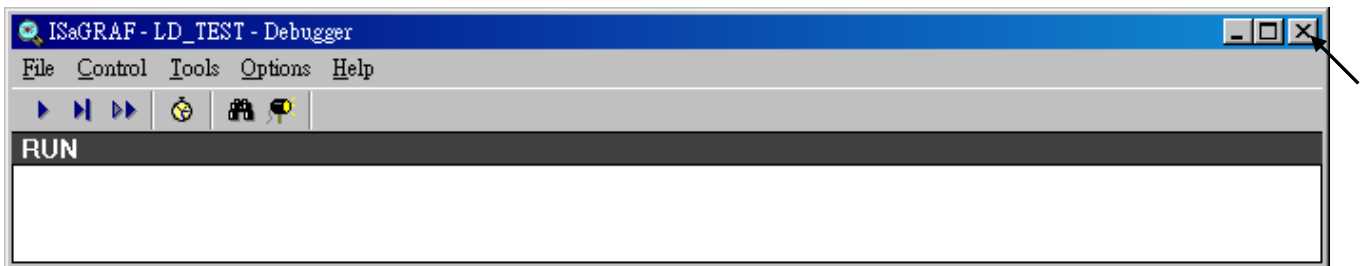
接著 點選 “Insert variable”，選取想要觀看的變數名稱。



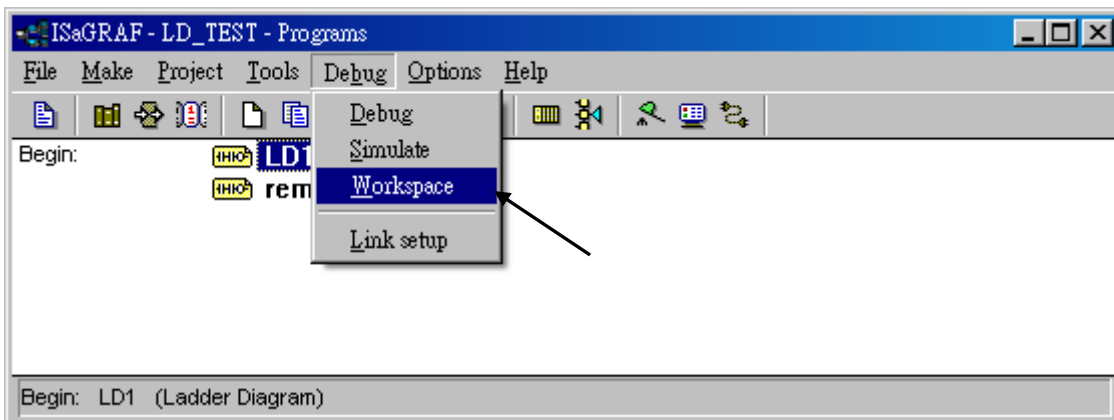
當想要觀看的變數都選完後，記得 點選 “Save list” 再離開。



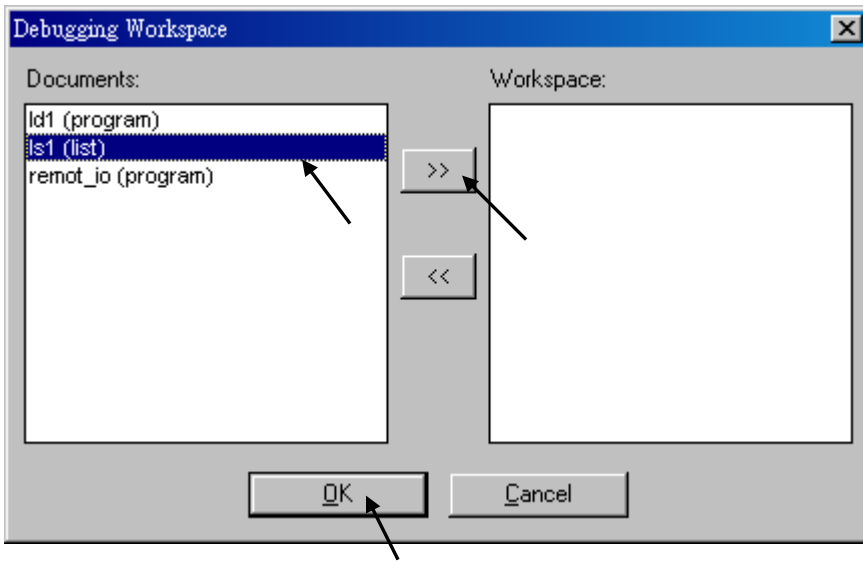
接著關閉 ”Debugger” 視窗。



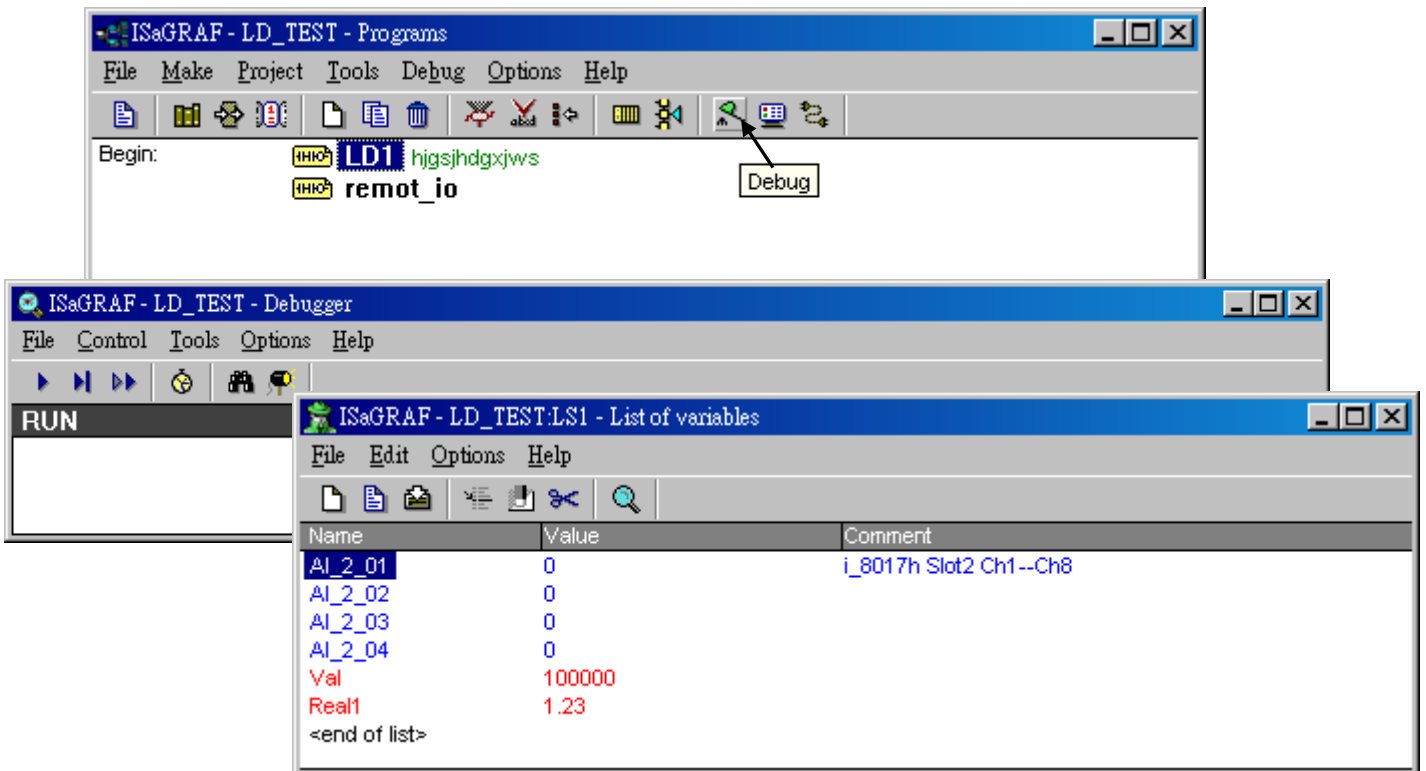
點選 “Debug – Workspace”



將 List 搬到右方.

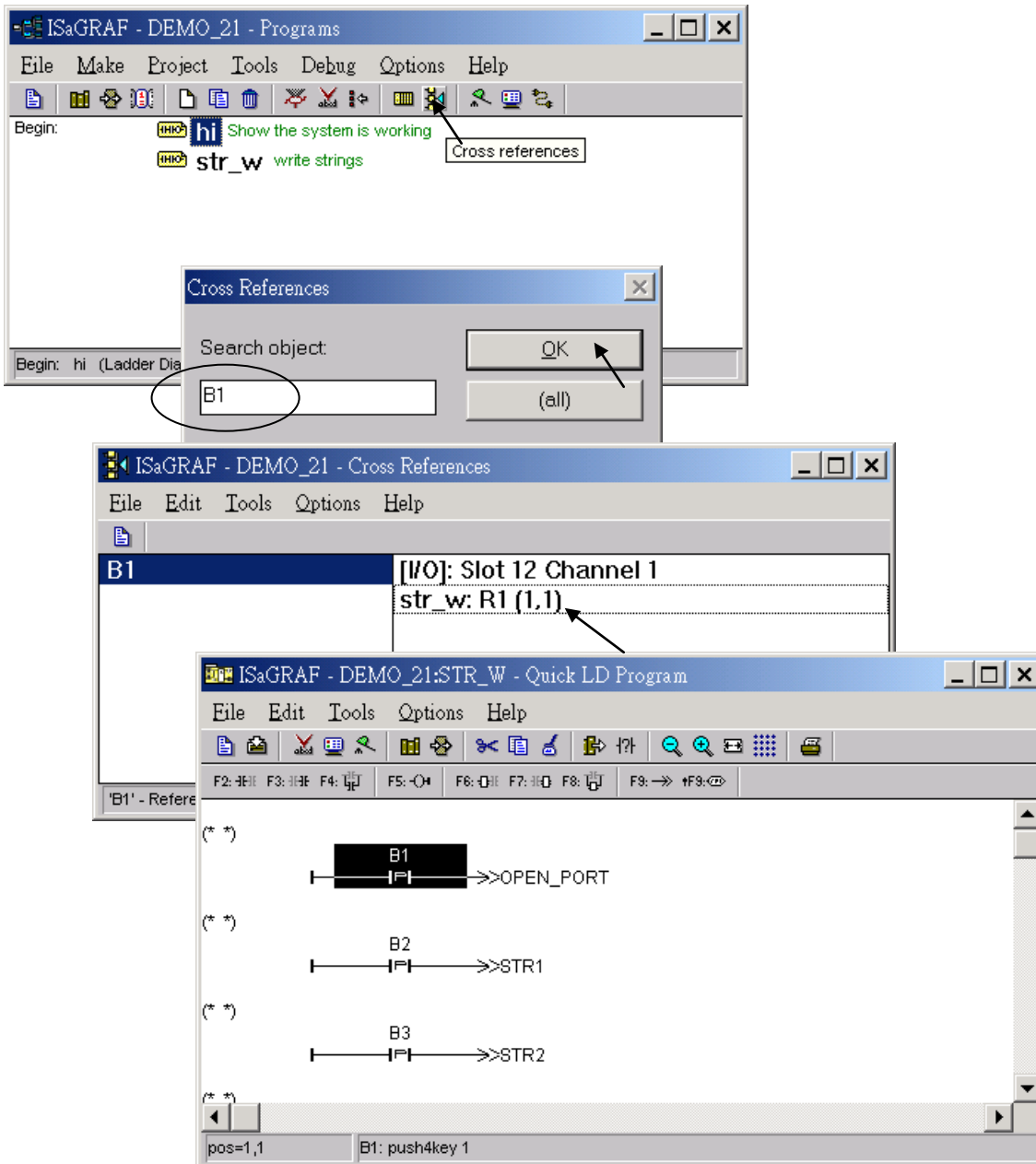


如此當下一次與 控制器連線時, 該變數的 List 就會自動顯示出來.



9.13: 如何在 ISaGRAF 內搜尋某一變數名稱 ?

請點選 “Cross references”，之後輸入要搜尋的變數名稱。搜尋到的結果會出現在右側，用滑鼠點選它就會跳到該所在位置。



第 10 章可保存變數和資料儲存

10.1: 可保存變數

注意: 假如在 PAC 的 Battery backup Memory 內 (或 S-256/512 & X607/608 內) 並未儲存 浮點數 (實數), 但卻要從其內讀取 浮點數(實數), RETAIN_F, RETAIN_X 與 RETAIN_A 可能會發生”Controller Fault”. 請參考第 10.6 節 – “偵測 Controller Fault”

I-8417/8817/8437/8837, I-7188EG/XG, W-8xx7, iP-8xx7, WP-8xx7, VP-2xW7, XP-8xx7-CE6, VP-2117, μ PAC-7186EG, 從以下驅動程式版本起, 支持新的可保存變數方式.

I-7188EG + X607 / X608:	2.05 版起, 建議採用 2.17 版 或 更高的版本
I-7188XG + X607 / X608:	2.04 版起, 建議採用 2.15 版 或 更高的版本
I-8xx7+ S256 / S512 :	3.07 版起, 建議採用 3.19 版 或 更高的版本
iP-8xx7 (內建 battery backup SRAM) :	產品 Release 起
WP-8xx7 (內建 battery backup SRAM) :	產品 Release 起
XP-8xx7-CE6 (內建 battery backup SRAM) :	產品 Release 起
VP-2xW7 (內建 battery backup SRAM) :	產品 Release 起
μ PAC-7186EG + X607 / X608 :	產品 Release 起
VP-2117 (內建 battery backup SRAM) :	產品 Release 起
W-8347/8747/8337/8737 + S256 / S512 :	3.18 版起, 建議採用 3.36 版 或 更高的版本 使用以下的 Wincon 新底板(back-plane)
W-8346/8746/8336/8736 + S256 / S512 :	3.18 版起, 建議採用 3.36 版 或 更高的版本 要使用以下的 Wincon 新底板
WB-831 (3 槽):	Rev 2.6 版起 (2006 年起出貨用的底板)
WB-871 (7 槽):	Rev 2.8 版起 (2006 年起出貨用的底板)

iP-8xx7, VP-2117, WP-8xx7, VP-25W7/23W7 與 XP-8xx7-CE6 出貨時已經內建 512K byte Battery SRAM (它們不需要添購 S-256/512).

假如有插 電池供應的 SRAM 於控制器的背板上 (I-8xx7: S256/S512, I-7188EG/XG: X607/X608, Wincon-8xx7: S256/S512), 最大的可保存變數數量為如下所示. 新的可保存變數方式是由以下的 ISaGRAF 函式來達成的

控制器 1 : μ PAC-7186EG/I-7188EG/XG + X607/608, I-8xx7+S256/512, iP-8xx7, VP-2117

控制器 2 : W-8xx7 + S256/512 (使用 Wincon 2006 年起出貨用的 新底板) 與 WP-8xx7, VP-2xW7, XP-8xx7-CE6

Retain_B : 設定 Boolean 可保存變數	控制器 1: 最多 256 個變數, 控制器 2: 最多 1024 個
Retain_N : 設定 Integer 可保存變數	控制器 1: 最多 1024 個變數, 控制器 2: 最多 4096 個
Retain_F : 設定 Real 可保存變數	控制器 1: 最多 1024 個變數, 控制器 2: 最多 4096 個
Retain_T : 設定 Timer 可保存變數.	控制器 1: 最多 256 個變數, 控制器 2: 最多 1024 個
Retain_X : 使用變數的 Network address 來設定可保存變數	
Retain_A : 使用變數的 Network address 來設定可保存變數陣列 (請參考第 2.6.2 節)	

以下的 ST 程序可以讀出 iP-8xx7, VP-25W7/23W7, WP-8xx7, XP-8xx7-CE6 底板的 2 顆電池的狀態

(* 請宣告 battery_state1 與 battery_state2 為 internal integer *)

battery_state1 := R_MB_ADR(1, 9992); (*電池 1 的狀態, 回傳 0: 電量不足, 99: 電量充足 *)

battery_state2 := R_MB_ADR(1, 9993); (*電池 2 的狀態, 回傳 0: 電量不足, 99: 電量充足 *)

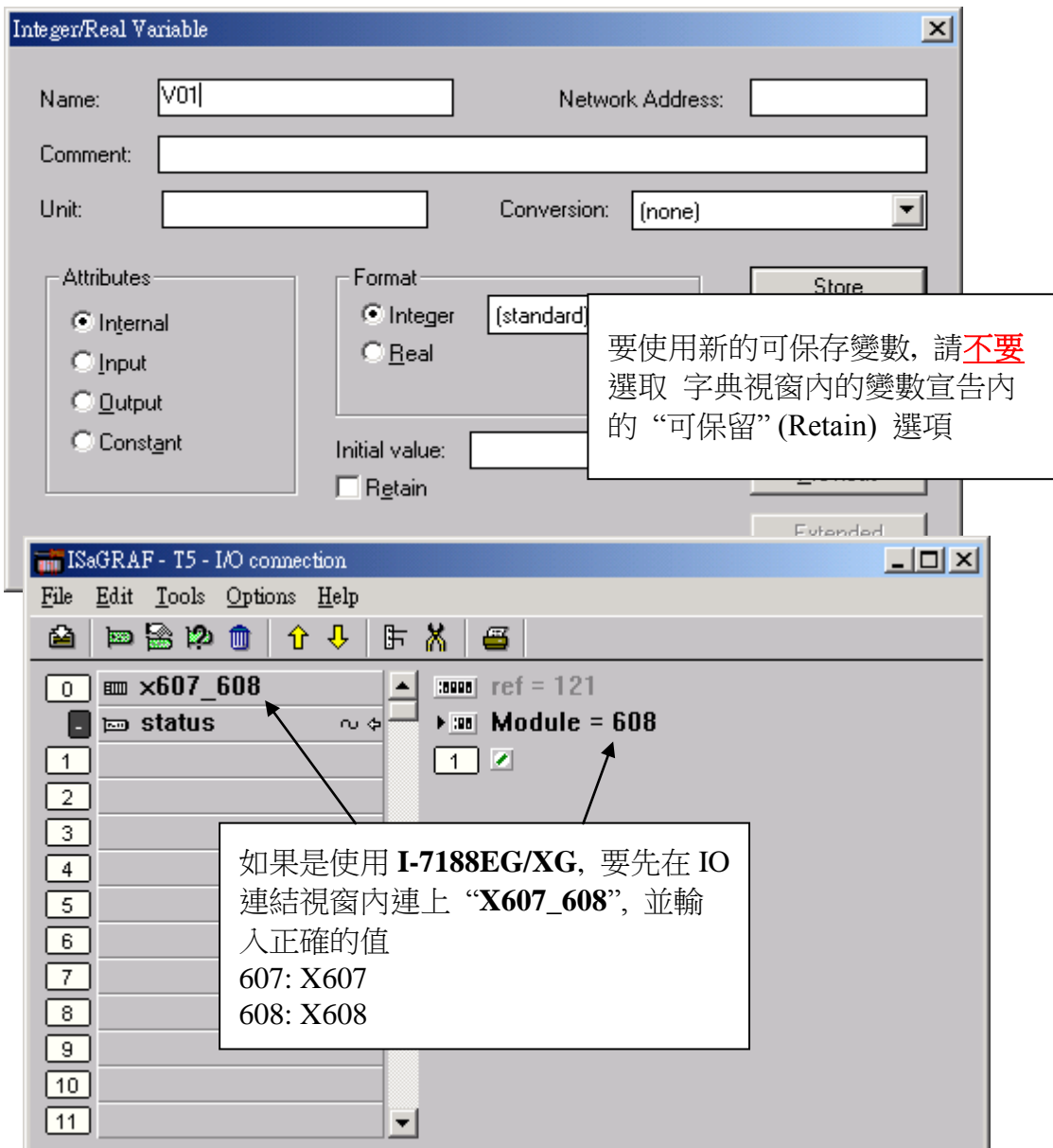
新的可保存變數的優點:

1. 值會一直保存, 不論是 "控制器斷電" 或 "重新 compile" 與 "重新下載 ISaGRAF 程式", 除非是 S-256/512, X607/608 上面的電池沒電了. (舊方式 碰到 "重新下載 ISaGRAF 程式", 值會清為 0)
2. 新的可保存變數 可保存的變數量 較舊方式多.

請參考下一頁的範例程式來實作新的可保存變數.

注意:

1. 要使用新的可保存變數, 請不要 選取字典視窗內的變數宣告內的 "可保留" (Retain) 選項.
2. 如果是使用 **μPAC-7186EG, I-7188EG/XG**, 要先在 IO 連結視窗內連上 "X607_608".



程式範例: (* 使用 變數名稱 來設定 為 可保存變數*)

(* 設定變數為 Retain variable 的動作, 一定要在第一個 PLC scan 內只做一次, 如下方的 code *)

(* To_Retain 宣告為 Boolean, Internal, 初值設為 True *)

(* Tmp 宣告為 Boolean, Internal *)

(*

Check1 宣告為 Integer, Internal, 不可勾選 “可保留”(Retain), 此變數用來 當作 一個 標記.
比如可以定義 若值為 1357246, 表示所有該指定初值的 Retain Variable, 初值都已經設定過了.
那就可讓 本應用的 控制程式開始動作 (因為該給的值已經設好了).

但若 Check1 的值不是 1357246, 表示 所有該指定初值的 Retain Variable, 還有一些沒設好初值. 那就不能讓本應用的 控制程式動作 (因為該給的值,還沒設好,若讓控制程式動作 ,會出錯, 必需要等 User 把該設的 Retain 值設好後,之候再設 Check1 為 1357246, 讓控制程式開始動作 .日後每次重新開機, 因為是 Retain Variable, 所以以前設過的值 都還在, 包含 Check1 也會是 1357246, 所以程式開機後就可動作, 不需再重新設 Retain 的初值 , 因為以前已經至少設過一次了.).

*)

(* B1 , B2 宣告為 Boolean, Internal, 不可勾選 “可保留”(Retain) *)

(* N1 , N2 宣告為 Integer, Internal, 不可勾選 “可保留”(Retain) *)

(* F1 , F2 宣告為 Real, Internal, 不可勾選 “可保留”(Retain) *)

(* T1 , T2 宣告為 Timer, Internal, 不可勾選 “可保留”(Retain) *)

if To_Retain then (* 當控制器剛啟動時, 使用變數名稱來設定可保存變數 *)

To_Retain := False ;(* 設為 False, 如此只有第 1 個 PLC Scan 會執行到這一段*)

Tmp := Retain_N(Check1 , 1) ; (* 此變數用來 當作 一個 標記 *)

Tmp := Retain_B(B1 , 1) ; Tmp := Retain_B(B2 , 2) ; (* Boolean *)

Tmp := Retain_N(N1 , 11) ; Tmp := Retain_N(N2 , 12) ; (* Integer *)

Tmp := Retain_F(F1 , 1) ; Tmp := Retain_F(F2 , 2) ; (* Real *)

Tmp := Retain_T(T1 , 1) ; Tmp := Retain_T(T2 , 2) ; (* Timer *)

end_if ;

(*以下的程序可以讀出 iP-8xx7, VP-25W7/23W7, WP-8xx7, XP-8xx7-CE6 底板的 2 顆電池的狀態*)

(* 請宣告 battery_state1 與 battery_state2 為 internal integer *)

battery_state1 := R_MB_ADR(1 , 9992) ;(*電池 1 的狀態, 回傳 0: 電量不足 , 99:電量充足*)

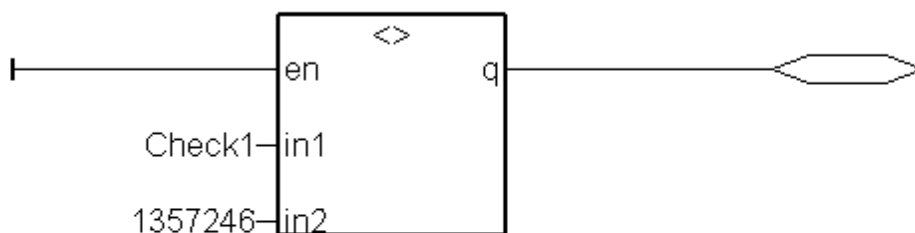
battery_state2 := R_MB_ADR(1 , 9993) ;(*電池 2 的狀態, 回傳 0: 電量不足 , 99:電量充足*)

(* 如此之後 Check1, B1, B2, N1, N2, F1, F2, T1, T2 的值就會一值被保存下來 *)

接下來 不管是寫 Ladder program 或 ST 都可判斷 Check1 這個標記值是否正確, 再 開始做控制動作. (用意再於讓 User 把 該設的 Retain 值, 至少設過一次初值, 然後才去設 Check1 為 1357246, 讓 以下的 控制程式開始動作)

之後用 Ladder 寫的 program, 第一行可加 Check1 的判斷為:

(* If Check1 is not equal to 1357246, exit this program to run next program *)



■ ■ ■

之後用 ST 寫的 program, 第一行可加 Check1 的判斷為:

(* If Check1 is not equal to 1357246, exit this program to run next program *)

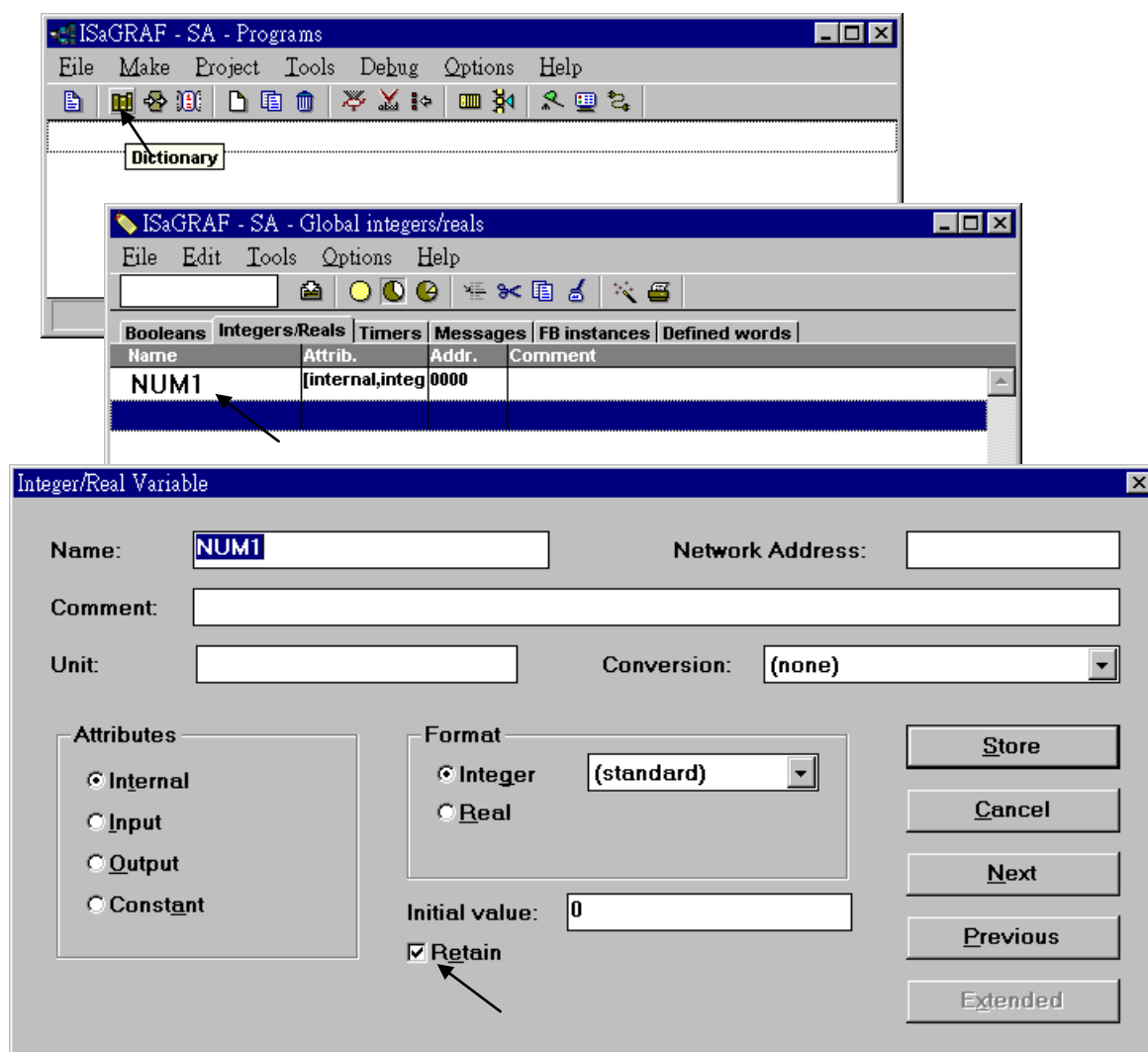
```
if Check1 <> 1357246 then
  return ;
end_if ;
```

...

舊的 可保存變數 方式:

如果沒插上 電池供應的 SRAM 於控制器的底板上 (I-8xx7: S256/S512, I-7188EG/XG: X607/X608, Wincon-8347/8747/8337/8737: S256/S512). 若是 I-8xx7 及 I-7188EG/XG 仍可支持可保存變數, 但數量很少, 而 Wincon 若不插 S-256/512,則不支持可保存變數. I-8xx7 與 I-7188EG/XG 的主機板內有 31 個 byte 的 NVRAM, 可儲存 6 個 **Integer/Real** (32-bit) 與 16 個 **boolean** 的可保存變數.

要使用舊可保存變數, 只需勾選 “可保留”(Retain) 項目就好.



注意:

假如有插電持供應的 SRAM 於控制器的背板上 (I-8xx7: S256/S512, I-7188EG/XG: X607/X608, Wincon-8347/8747/8337/8737: S256/S512), 請使用先前介紹的新的 可保存變數方式. 舊的可保存變數方式有 2 個缺點: (1) 當重新下載一個有修改過的 ISaGRAF 程式後, 原先已保存的值會被清為 0. (2) 舊方式的數量較少.

10.2: 備份資料到 EEPROM

EEPROM 是一種可以不必供電即能長久保存資料的記憶體，它可以無限制的被讀取，但最多只能被寫入 1,000,000 次，因此適合用來儲存設定性的資料。

ISaGRAF 控制器使用下列函數來存取 EEPROM 記憶體。

EEP_B_R	讀取一個布林值
EEP_BY_R	讀取一個字元 (Byte: 0 ~ 255)
EEP_WD_R	讀取一個字組 (Word: 2 bytes, signed, -32768 ~ +32767)
EEP_N_R	讀取一個整數 (Integer: 4 bytes, signed)
EEP_F_R	讀取一個實數 (Real: 4 bytes, signed float)
(若 EEP_F_R 讀取的 EEPROM 內位址內存的不是實數，讀取值會錯誤，有時更會產生 Local Fault 編號 114, 請參考第 10.6 節)	

在您將資料寫入 EEPROM 之前,須先確認 EEPROM 的保護是否已解除，您可以參考下列函式。

EEP_EN	解除 EEPROM 記憶體的保護功能以寫入資料
EEP_PR	關閉 EEPROM 記憶體以防止寫入資料
EEP_B_W	寫一個布林值到 EEPROM 記憶體
EEP_BY_W	寫一個字元到 EEPROM 記憶體 (Byte: 0 ~ 255)
EEP_WD_W	寫一個字組到 EEPROM 記憶體 (Word: 2 bytes, signed, -32768 ~ +32767)
EEP_N_W	寫一個整數到 EEPROM 記憶體(Integer: 4 bytes, signed)
EEP_F_R	寫一個實數到 EEPROM 記憶體(Real: 4 bytes, signed float)

注意:

1. EEP_F_R, EEP_N_R, EEP_N_W, EEP_F_W 使用相同的 EEPROM 位址，程式不可在同一位址同時使用成實數與整數，只能挑一個來使用。
2. 假如在 EEPROM 內該位址並未儲存 浮點數(實數)，但卻要使用 EEP_F_R 從 EEPROM 內讀取 浮點數(實數) 可能會發生"Controller Fault 編號 114"。請參考第 10.6 節 – “偵測 Controller Fault”

在 ISaGRAF PAC 內，字元 (byte)，字組(Word)，整數(Integer) 與 實數 (Real) 均存放在 EEPROM 的相同記憶區塊中，在您使用寫入函式時請小心配置這些變數的位址。不同 PAC 機型可以使用的 EEPROM 記憶體大小不一樣，以 I-8xx7, I-7188EG/XG 為例只有 1,512 個字元，位址編號從 1 到 1512，字元的位址分布從 1 到 1512，字組則從 1 到 756，而整數的位址分布則從 1 到 378。下面的位址編號將使用相同的記憶體位址。

I-8xx7, I-7188EG/XG :

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... 378 *)
Word	2n-1, 2n	
Integer, Real	n	

iP-8xx7, μPAC-7186EG, VP-2117, Wincon-8xx7 :

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... 3568 *)
Word	2n-1, 2n	
Integer , Real	n	

WP-8xx7, VP-25W7, VP-23W7, XP-8xx7-CE6 :

Byte	4n-3, 4n-2, 4n-1, 4n	(* n = 1, 2, ... 1536 *)
Word	2n-1, 2n	
Integer , Real	n	

當寫入 EEPROM 次數超過 100,000 次時，將會折損 EEPROM。在下面的程式中，EEPROM 每一個 PLC Scan 週期就被寫入一次，這是很危險的。

```
(* ST 程式, 宣告 Val 為整數變數,宣告 TEMP 為布林變數 *)
TEMP := eep_n_w(1, Val);    (* 危險 *)
```

若 Val 變數沒有經常改變，寫成下面的程式就比較不會發生問題。

```
(* ST 程式, 宣告 Val, Old_Val 為整數變數, 宣告 TEMP 為布林變數 *)
IF Val <> Old_Val THEN
    TEMP := eep_n_w(1, Val);
    Old_Val := Val;
END_IF;
```

對 EEPROM 作讀寫動作時，將會消耗很多控制器的 CPU 時間。

下面為每個函式被呼叫時所耗用的大約時間。

EEP_EN	~ 0.08 ms	EEP_PR	~ 0.08 ms
EEP_B_R	~ 0.8 ms	EEP_B_W	~ 6 ms
EEP_BY_R	~ 0.8 ms	EEP_BY_W	~ 6 ms
EEP_WD_R	~ 1.5 ms	EEP_WD_W	~ 12 ms
EEP_N_R	~ 2.9 ms	EEP_N_W	~ 23 ms

當控制器的電源恢復時，建議只讀取一次 EEPROM 資料，且當資料有變動時再做更新，請參考第 11 章的 “demo_17” 範例程式。若資料常常在作更新變動，則不適合儲存在 EEPROM 內。

10.3: 電池保持式 SRAM

注意: WP-8xx7, VP-25W7, XP-8xx7-CE6 等 PAC 有支持 10.1 節列出的 新的可保留變數 方法, 但不支持本節列出的採用 S_X_X 等 function 來直接 讀/寫 電池保池式 SRAM 的用法.

假如在 S-256/512 & X607/608 內並未儲存 浮點數(實數), 但卻要從 S-256/512 & X607/608 內讀取 浮點數(實數) 可能會發生”Controller Fault”. 請參考第 10.6 節 – “偵測 Controller Fault”

I-8417/8817/8437/8837 可以選購 1 個 S256 或 S512 電池保持式 SRAM 來保存資料, 記錄 Alarm, 及某些訊息, 而 I-7188EG/XG 則需搭配 1 個 X607 或 X608. 存放於這些電池保持式 SRAM 的資料會一直保存, 除非它的電池沒電了. 它們的記憶容量如下, 最上方的 12K byte 是保留給 ISaGRAF 使用的 (W-8337/8737/8336/8736 保留 64K).

I-8417/8817/8437/8837		I-7188EG/XG	
S256	244K bytes (256-12=244)	X607	116K bytes (128-12=116)
S512	500K bytes (512-12=500)	X608	500K bytes (512-12=500)
W-8337/8737/8336/8736			
S256	192K bytes (256-64=192)		
S512	448K bytes (512-64=448)		

假如控制器有插電池保持式 SRAM, 可宣告的 ”可保留變數” (Retained Variable)可增加如下(請參考 10.1 節).

I-7188EG/XG+X607/608 and I-8417/8817/8437/8837+S256/512		
	新的可保存變數方式	舊的可保存變數方式
Boolean	256	256
Integer	1024	256 (Integer + Real)
Real	1024	
Timer	256	32
W-8337/8737/8336/8736+S256/512 使用 Wincon 新底板(10.1 節)		
	新的可保存變數方式	舊的可保存變數方式
Boolean	1024	1024
Integer	4096	4096 (Integer + Real)
Real	4096	
Timer	1024	1024

ICP DAS 提供 1 個工具程式 “**ICPDAS UDloader**”, 可安裝在 PC 上從工控器的 SRAM 內 上載 (抓取)資料, 或 下載(存入) 資料到 SRAM 內. 請由 ICP DAS ‘光碟內的 CD-ROM:\napdos\isagraf\some_utility\ 路徑複製 “**UDloader.exe**” 到你的 PC 內.

I-8417/8817/8437/8837 從驅動程式 2.25 版 (建議採用 3.19 版 或 更高的版本) 起支援 S256/S512, 而 7188EG 從 1.18 版起 (建議採用 2.17 版 或 更高的版本) 支援 X607/X608, 7188XG 則由 1.16 版起 (建議採用 2.15 版 或 更高的版本) 支援 X607/X608. W-8337/8737/8336/8736 從 3.18 版起 (建議採用 3.36 版 或 更高的版本) 支持 S256/S512 (請參考 10.1 節), 如果你的控制器內的是較舊的版本, 請更新到最新的版本. 驅動程式可從

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 下載

10.3.1: 存取 SRAM

電池保持式 SRAM 可存取 boolean, byte, word, integer, real & message 等資料. 格式如下.
(請參考第 11.3.7 節使用 S-256/512 與 UDownloader 的範例程式)

Boolean:	True=1, False=0	1 byte
Byte:	0 ~ 255	1 byte
Word:	-32768 ~ 32767	2 bytes
Integer:	signed 32-bit	4 bytes
Real:	float	4 bytes
Message:	string (長度<=255)	len bytes

ISaGRAF 程式可使用以下函式來存取 SRAM (請參閱附錄 A).

S_B_R, S_B_W, S_BY_R, S_BY_W, S_M_R, S_M_W
S_WD_R, S_WD_W, S_N_R, S_N_W, S_R_R, S_R_W
S_MV

10.3.2: 從 SRAM 內 上載(抓出) 資料

為了讓 PC 能從 SRAM 內上載資料出來, ISaGRAF 程式需劃分 SRAM 成 1 ~ 8 個檔案. 每個檔案會有 1 個編號(1~8), 檔案名稱最多 8 個字+3 個延伸檔名. 歸劃檔案的相關函式為

S_FL_INI, S_FL_AVL, S_FL_RST, S_FL_STS

請使用 S_FL_INI 與 S_FL_AVL 來規劃 “檔案配置區” 及 “目前可取得資料區”(參閱附錄 A 及 demo_40, 41 or 42).

電池保持式 SRAM 是由 Byte 組成的. 最大可存取的記憶容量與型號有關 (最上方的 12K byte 是保留給 ISaGRAF 使用的).

型號	Byte 編號
I-8xx7: S256	1 ~ 249,856 (244K), (256-244=12K 保留)
I-8xx7: S512	1 ~ 512,000 (500K), (512-500=12K 保留)
I-7188XG/EG: X607	1 ~ 118,784 (116K), (128-116=12K 保留)
I-7188XG/EG: X608	1 ~ 512,000 (500K), (512-500=12K 保留)

SRAM 內的檔案可配置在上表中的 Byte 編號內. 每個檔案的”檔案配置區” 可表示為(**Begin, End**). Begin 為檔案配置區的開頭 Byte 編號, End 為檔案配置區的結尾 Byte 編號, Begin 需小於 End.

SRAM 內檔案的 “目前可取得資料區” 可表示為 (**Head, Tail**). Head 為開頭的 Byte 編號, Tail 為結尾的 Byte 編號. Head 可以 大於,等於, 或小於 Tail.

例, 有個 SRAM 檔案的” 檔案配置區” (Begin, End) = (1, 20000)

1. 如果 (Head, Tail) = (1001,5100), 表示目前可取得資料從 Byte 編號 1001 到 5100. 此檔目前含有 4100 個 byte.
2. 如果 (Head, Tail) = (10001,5000), 表示目前可取得資料從 Byte 編號 10001 到 20000, 緊接著再由 1 到 5000. 此檔目前含有 15000 個 byte.

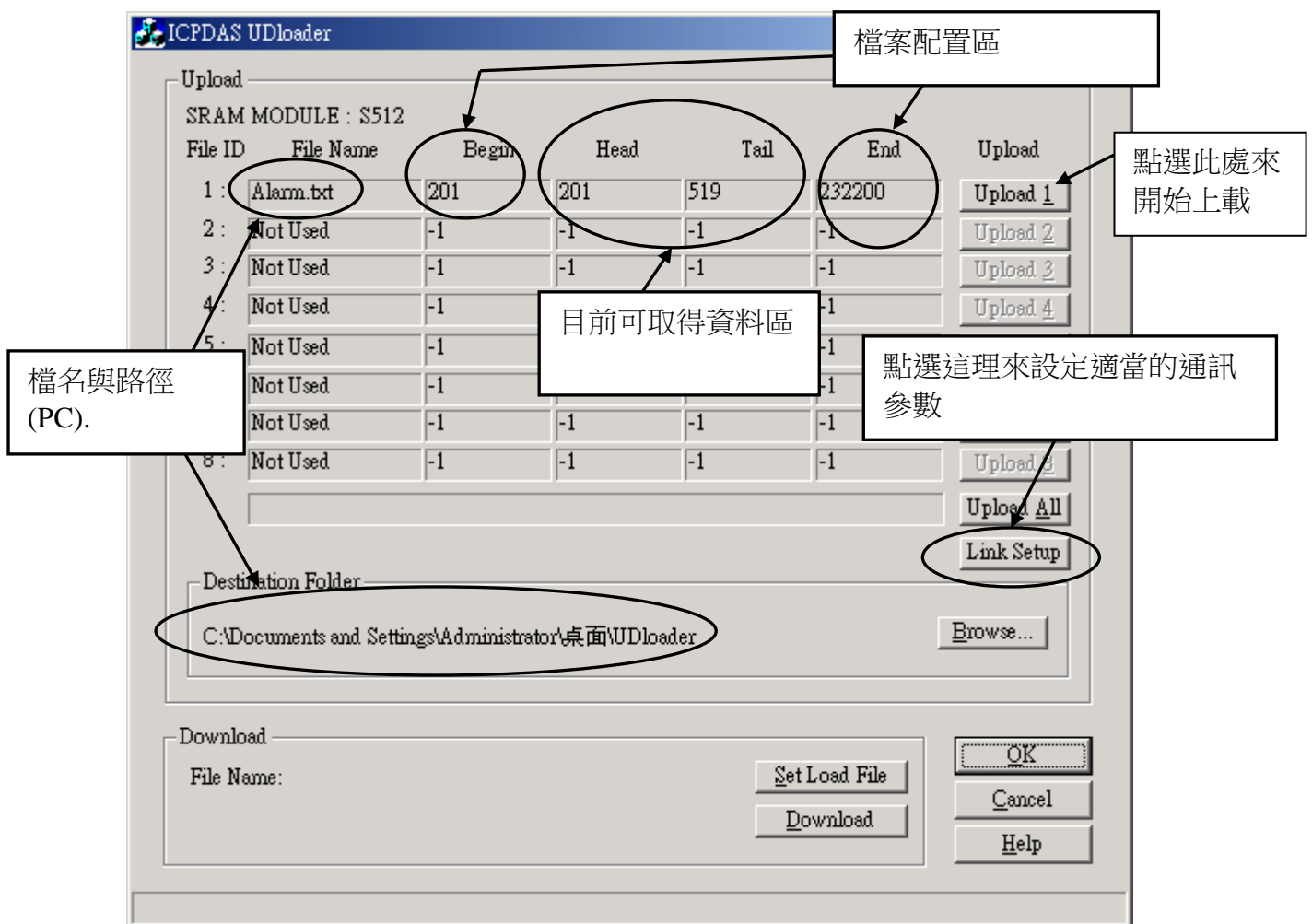
3. 如果 (Head, Tail) = (5001,5000), 表示目前可取得資料從 Byte 編號 5001 到 20000, 緊接著再由 1 到 5000. 此檔目前含有 20000 個 byte.
4. 如果 (Head, Tail) = (5000,5000), 表示此檔目前無資料.
5. 如果 (Head, Tail) = (-1,-1), 表示此檔目前無資料.

要從 SRAM 內上載資料出來, 請先確定 “ICPDAS UDloader” 是否已安裝.

在 PC 上執行 “UDloader.exe”, 之後點選 “Link Setup” 來設定適當的通訊參數, 之後點選 “Upload 1” 來上載編號為 1 的檔案.

範例:

請下載 demo_41 程式到 I-8417/8817/8437/8837 內. 之後按壓工控器面板上按鈕 1, 2, 3 或 4 數次數次. 之後執行 Udloader.exe 上載 SRAM 檔案.



10.3.3: 下載資料到 SRAM 內

下列函式可用來處理資料下載到 SRAM 內(參閱附錄 A 及 demo_44).

S_DL_EN, S_DL_DIS, S_DL_RST, S_DL_STS

請使用 S_DL_EN 來允許 PC 下載資料到 SRAM 內.

控制器能夠接受的下載資料為 Binary 格式, 如下

Byte: 0 ~ 255	1 byte
Word: -32768 ~ +32767	2 byte [low byte] [high byte]
Integer: 32-bit, signed integer	4 byte [lowest] [2nd] [3rd] [highest]
Real: 32-bit float	4 byte [lowest] [2nd] [3rd] [highest]
String: 最長 255 byte	

要使用 “UDloader.exe” 下載資料到 SRAM, 必需將資料編成 1 個 Text 檔. 資料需使用以下格式輸入.

第 1 行為 1 個數字, 表示要下載到 SRAM 的那個 Byte 編號起. 合法的編號為

S256: 1 ~ 249,856	S512: 1 ~ 512000
X607: 1 ~ 118,784	X608: 1 ~ 512000

第 2 行起為資料

A. 字串 (String)

字串需開始及結束於 ‘ 符號. 例, ‘Abcd123’ (7 個 byte). \$NN (NN 為 16 進位數字, 且不為 00) 可用來表示 ASCII 符號表的字元. 例, ‘ABC\$0D’ 包含 4 個 byte, 第 4 個 byte 為 <CR> 字元.

B. Byte

Byte 需開始於 (符號 及 結束於) 符號. 例, (0), (123), (255). 值需介於 (0) 到 (255).

C. Word

Word 需開始於 [符號 及 結束於] 符號. 例, [-100], [20000], [32767]. 值需介於 [-32768] 到 [32767].

D. Integer

Integer 需開始於 { 符號 及 結束於 } 符號. 例, {-1234567}, {200000}. 值需介於 {-2147483648} 到 {2147483647}.

E. Real

Real 需開始於 < 符號 及 結束於 > 符號. 例, <123>, <1.56E-2>, <-123.456>

3. 同一行的每個資料間, 需使用 空格 <SP> 或 逗號 , <Comma> 或 <Tab> 等字元 來區隔

例,

201 ← 下載到 SRAM 內 Byte 編號 201 開始的位址
‘Hello’ (10) (20) (30) (40) [-10000] {70000} ‘End’ ← 資料共 18 個 byte

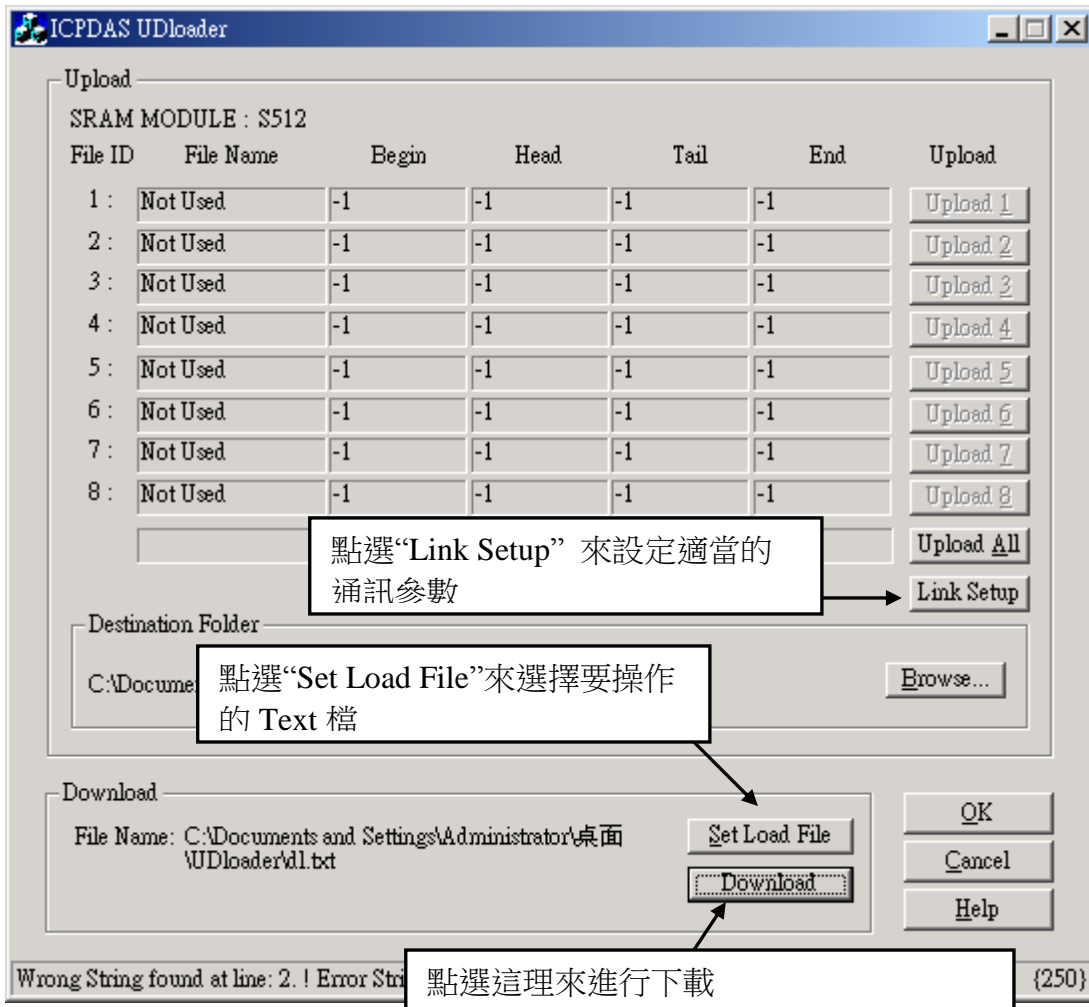
1 ← 下載到 SRAM 內 Byte 編號 1 開始的位址
 (23) ← 資料共 57 個 byte
 {-1},{2},{-3},{4},{-5},{6} {-7} {8} {-9} {10} ← 逗號, 空白, 及 <Tab> 都可為 間隔字元
 <0.123> <456.789> <100> . <2.3E3>

範例: 請下載 demo_44 程式到 I-8417/8817/8437/8837 內. 然後在 PC 編輯 1 個 Text 檔, 如下.

```
1
{1000} {250} {100} 'sTART'
```

- {1000} 表示 L1 的閃爍週期為 1000 ms.
- {250} 表示 L2 的閃爍週期為 250 ms.
- {100} 表示 L3 的閃爍週期為 100 ms. .

之後執行 “UDloader.exe” 並下載此 Text 檔. 你會看到 I-8xx7 控制器面板上的 LCD 有些動作.



10.3.4: 電池保持式 SRAM 的函式說明

詳細的參數說明請參閱附錄 A.

S_FL_INI 設定電池保持式 SRAM 檔案的 檔案配置區
S_FL_AVL 設定電池保持式 SRAM 檔案的目前可取得資料的所在位置
S_FL_STS 取得電池保持式 SRAM 檔案的狀態, PC 上載資料的結尾 byte 編號
S_FL_RST 重置電池保持式 SRAM 檔案狀態為 "PC 未上載資料過"

S_B_R: 從電池保持式 SRAM 讀出 1 個 Boolean 值
S_BY_R: 從電池保持式 SRAM 讀出 1 個 Byte 值(0~255)
S_WD_R: 從電池保持式 SRAM 讀出 1 個 Word 值 (-32768 ~ +32767)
S_N_R: 從電池保持式 SRAM 讀出 1 個 Integer 值 (32 bit, signed)

S_R_R: 從電池保持式 SRAM 讀出 1 個 Real 值 (32 bit, float)
(若電池保持式 SRAM 相對位置內存的不是 Real 值, 卻使用 S_R_R 去讀取, 讀取值會不正確, 有時更會產生 Local Fault 編號 102 號, 請參閱第 10.6 節)

S_M_R: 從電池保持式 SRAM 讀出 1 個 字串

S_B_W: 寫入 Boolean 值到電池保持式 SRAM
S_BY_W: 寫入 Byte 值到電池保持式 SRAM (0 ~ 255)
S_WD_W: 寫入 Word 值到電池保持式 SRAM (-32768 ~ +32767)
S_N_W: 寫入 Integer 值到電池保持式 SRAM (32 bit, signed)
S_R_W: 寫入 Real 值到電池保持式 SRAM (32 bit, float)
S_M_W: 寫入字串到電池保持式 SRAM

S_DL_EN 開啟下載的授權, 如此 PC 可以下載資料到 電池保持式 SRAM 內
S_DL_DIS 關閉下載的授權, 如此 PC 無法下載資料到 電池保持式 SRAM 內
S_DL_STS 取得 PC 對電池保持式 SRAM 的 下載狀態
S_DL_RST 重置電池保持式 SRAM 下載狀態 為 "-1: 無動作"

S_MV 在電池保持式 SRAM 內複製資料

10.4: 使用 I-8073 來儲存資料

ISaGRAF PAC 不支持 I-8073 與 I-8072.

10.5: 檔案存取

- 注意:** 1. 假如在 File 內的相對位置並未儲存 浮點數(實數), 但卻要從 File 內對該位置使用 F_READ_F 來讀取 浮點數(實數) 可能會發生"Controller Fault 編號 117". 請參考第 10.6 節的說明.
2. WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6 支持檔案存/取功能, 而 I-8xx7, I-7188EG/XG, VP-2117, 7186EG, iP-8xx7 則沒有.
 3. 若檔案位置在 '\System_Disk\' 或 '\Micro_SD\' 等儲存裝置內 (例如 \Micro_SD\data1.txt), 將 PAC 關機後檔案仍然存在, 但在這 2 個路徑內 Read / Write 檔案速度很慢.
 4. 若檔案位置是屬於 RAM 內 (例如 \Temp\data2.txt), 則會儲存在 RAM 內, PAC 關機後檔案就消失,但優點是 RAM 檔案的 Read / Write 速度很快 ('\System_Disk\' 或 '\Micro_SD\' 等儲存裝置的 Read / Write 很慢)

WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7 支持以下的 **ISaGRAF** 標準函式

F_OPEN	以 Binary 格式開啟已經存在的檔案以供 讀取 .
F_WRITE	以 Binary 格式開啟已經存在的檔案以供 讀取 & 寫入 .
F_CLOSE	關閉已經開啟的檔案
F_EOF	測試是否 檔案指標 已達 檔案結尾
FA_READ	從檔案內讀出 1 個 binary 長整數 (4 bytes, signed).
FA_WRITE	寫 1 個 binary 長整數(4 bytes, signed) 到檔案內.
FM_READ	從檔案內讀出 1 個 字串.
FM_WRITE	寫 1 個字串到檔案內, 會附加 <CR> <LF> 符號於字串後方

WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7 支持以下 **ICP DAS** 提供的函式

F_APPEND	附加一個檔案於另一個檔案的尾端.
F_COPY	覆製 1 個檔案.
F_CREAT	建立 1 個新的檔案.
F_DELETE	刪除 1 個檔案.
F_DIR	建立 1 個新資料夾 (目錄).
F_END	移動 檔案指標 到檔案的尾端.
F_SEEK	移動 檔案指標 到某個位置
F_READ_B	從檔案內讀出 1 個 binary 字元(0 - 255) (1 byte, unsigned).
F_WRITE_B	寫 1 個 binary 字元(0 -255) 到檔案內 (1 byte, unsigned).
F_READ_W	從檔案內讀出 1 個 binary 字組(-32768 to +32767) (2 byte, signed) .
F_WRITE_W	寫 1 個 binary 字組(-32768 to +32767) 到檔案內 (2 byte, signed).
F_READ_F	從檔案內讀出 1 個 binary 實數(例如 123.45, -2.15E-03, ...) (4 byte).
F_WRITE_F	寫 1 個 binary 實數 到檔案內 (4 byte) .
F_WRITE_S	寫 1 個字串到檔案內, 不附加 <CR> <LF> 符號於字串後方.
F_EXIST	偵測一個 檔案 或 資料夾 (目錄) 是否存在.

第 11.3.6 節 或 以下路徑可找到對於 檔案存取 的範例程式

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ "wpdmo_54.pia", 55, 56, 51, 50, 1 或 2 或 <ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/> 內

10.5.1: 範例 Wpdmo_51: 從檔案內讀取 10 個實數值, 共 10 行, 每行 1 個實數

本範例程式為 Wpdmo_51.pia 放於

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ 內 或
<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> > Driver 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

本 Wpdmo_51 範例程式 一開機 或 將 “RE_LOAD” 變數 再設為 TRUE 時, 會再去讀
\System_Disk\data51.txt 來更新 10 個 REAL 變數之值.

(在 System_Disk 卡內操作檔案很花 CPU 時間, 讀/寫 完後請 將檔案 Close, 並且避免每個 PLC Scan 都在 Read / Write 檔案, 不然 PLC Scan Time 會變很大)

要使這個範例仍夠正確運作, 請在 PC 用 NotePad (記事本) 編一個 “data51.txt” 文字檔, 共 10 行, 每行有 1 個實數值, 然後用 ftp 傳到 WP-8xx7 的 \System_Disk\ 路徑內.

“data51.txt” 文字檔 內容可如下:

```
2.345
999.03
-1.01
456.789
2
456.77
5.9E-12
32.3
45.1
33.3
```

變數宣告:

Name	Type	Attribute	Description
RE_LOAD	Bool	Internal	設為 True 會去讀 File 一次, 宣告初值為 TRUE
TMP	Bool	Internal	暫時使用的 Boolean 變數
File_name1	Message	Internal	長度 64, 初值為 \System_Disk\data51.txt
Msg1	Message	Internal	長度 128, 會顯示 File 處理狀態
str1	Message	Internal	長度 255, 用來儲存讀到 File 內的一行字串
F_VAL[0..9]	REAL	Internal	為變數陣列, Dim 設為 10. 讀到的 10 個 REAL 值
TMP_F	REAL	Internal	暫時使用的 REAL 變數
File1	Integer	Internal	File ID
ii	Integer	Internal	給 for 迴圈使用的 index

ST 程式:

```
if RE_LOAD then      (* 若將 RE_LOAD 設為 TRUE, 會去讀 File 一次 *)

RE_LOAD := FALSE ;

File1 := f_wopen( File_name1 ) ;    (* 開啟 File 為 可讀 / 可寫 模式 *)

if File1 = 0 then    (* 若回傳 0, 表示開啟 File 失敗 *)
  Msg1 := 'Can not Open file ' + File_name1 ;
  return ;    (* 無法開啟該 File, 用 Return 離開本 ST 程式 *)
end_if ;

for ii := 0 to 9 do    (* 共要讀 10 行 *)

  if f_eof(File1) = TRUE then    (* 偵測是否抵達 File 的 尾端 *)
    Msg1 := 'Data number is not enough in ' + File_name1 ;
    Exit ;    (* 離開 for 迴圈 *)
  end_if ;

  str1 := fm_read(File1) ;    (* 讀取 File 內的一行字串 *)
  TMP_F := str_real(str1);    (* 將字串轉成 實數 *)
  if TMP_F = 1.23E-20 then    (* 若傳回 1.23E-20 表示 格式錯誤 *)
    Msg1 := 'The ' + Msg(ii+1) + 'th Data format is not correct !' ;
    exit ;    (* 離開 for 迴圈 *)
  end_if ;

  F_VAL[ii] := TMP_F ;    (* 前面 有轉換成功, 值存入 F_VAL[0..9] 內 *)

end_for ;

TMP := f_close(File1) ;    (* File 只要成功開啟過, 處理完就需 close *)
If ii=10 then    (* 10 行資料都讀到了 *)
  Msg1 := 'Read ' + File_name1 + ' Ok ' ;
end_if ;

end_if ;
```

10.5.2: 範例 WPdmo_54: 從檔案內讀取 20 個實數值, 共 4 行, 每行 5 個實數

本範例程式為 Wpdmo_54.pia 放於

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ 內 或
<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> > Driver 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

本 Wpdmo_54 範例程式 一開機 或 將 “RE_LOAD” 變數 再設為 TRUE 時, 會再去讀
\System_Disk\data54.txt 來更新 20 個 REAL 變數之值.

(在 System_Disk 卡內操作檔案很花 CPU 時間, 讀/寫 完後請 將檔案 Close, 並且避免每個 PLC Scan 都在 Read / Write 檔案, 不然 PLC Scan Time 會變很大)

要使這個範例仍夠正確運作, 請在 PC 用 NotePad (記事本) 編一個 “data54.txt” 文字檔, 共 4 行, 每行有 5 個實數值, 然後用 ftp 傳到 WP-8xx7 的 \System_Disk\ 路徑內.

“data54.txt” 文字檔 內容可如下:

```
23 , 65.9 , 0.012 , 5.87 , 88.2
0.34 , 8.0005 , -2.0E8 , 4.08 , 5.32E-6
2 , -7 , 6666.8 , 456.07 , 1.01
5 , 6 , 7 , 8 , 9
```

變數宣告:

Name	Type	Attribute	Description
RE_LOAD	Bool	Internal	設為 True 會去讀 File 一次, 宣告初值為 TRUE
TMP	Bool	Internal	暫時使用的 Boolean 變數
File_name1	Message	Internal	長度 64, 初值為 \System_Disk\data54.txt
Msg1	Message	Internal	長度 128, 會顯示 File 處理狀態
str1	Message	Internal	長度 255, 用來儲存讀到 File 內的一行字串
F_VAL[0..19]	REAL	Internal	為變數陣列, Dim 設為 20. 讀到的 20 個 REAL 值
NUM1	Integer	Internal	接收 Msg_F() 的回傳值, 若為 -1 表示格式錯誤
File1	Integer	Internal	File ID
ii	Integer	Internal	給 for 迴圈使用的 index
jj	Integer	Internal	給 另一個 for 迴圈使用的 index

ST 程式:

```
if RE_LOAD then      (* 若將 RE_LOAD 設為 TRUE, 會去讀 File 一次 *)

  RE_LOAD := FALSE ;

  File1 := f_wopen( File_name1 ) ;      (* 開啟 File 為 可讀 / 可寫 模式 *)

  if File1 = 0 then   (* 若回傳 0, 表示開啟 File 失敗 *)
    Msg1 := 'Can not Open file ' + File_name1 ;
    return ;      (* 無法開啟該 File, 用 Return 離開本 ST 程式 *)
  end_if ;

  for ii := 0 to 3 do  (* 共要讀 4 行 *)

    if f_eof(File1) = TRUE then  (* 偵測是否抵達 File 的 尾端 *)
      Msg1 := 'There should be at least 4 rows in ' + File_name1 + '!!!' ;
      Exit ;      (* 離開 for 迴圈 *)
    end_if ;

    str1 := fm_read(File1) ;      (* 讀取 File 內的一行字串 *)

    NUM1 := Msg_F(str1, 1) ;      (* 轉換該字串為 數個 REAL 值, 並存入 1 號 Float
    陣列內 *)

    if NUM1 <> 5 then  (* 轉換得到的 REAL 值數量不是 5 個, 缺資料 .若為 -1
    表示格式錯誤 *)
      Msg1 := 'The ' + Msg(ii+1) + 'th row data format is not correct or data
      number is not 5!' ;
      Exit ;      (* 離開 for 迴圈 *)
    end_if ;

    for jj := 0 to 4 do
      (* 從 1 號 Float 陣列 1 到 5 位址內 取出轉換後的 REAL 值 存入 F_VAL[0..19] 內 *)
      F_VAL[ 5 * ii + jj ] := ARY_F_R( 1, jj + 1 ) ;
    end_for ;

  end_for ;

  TMP := f_close(File1) ;      (* File 只要成功開啟過, 處理完就需 close *)

  If ii = 4 then  (* 4 行資料都讀到了 *)
    Msg1 := 'Read ' + File_name1 + ' Ok ' ;
  end_if ;

end_if ;
```

10.5.3: 範例 Wpdmo_55: 從檔案內讀取 20 個整數值, 共 2 行, 每行 10 個整數

本範例程式為 Wpdmo_55.pia 放於

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ 內 或
<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> > Driver 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

本 Wpdmo_55 範例程式 一開機 或 將 “RE_LOAD” 變數 再設為 TRUE 時, 會再去讀
\System_Disk\data55.txt 來更新 20 個 Integer 變數之值.

(在 System_Disk 卡內操作檔案很花 CPU 時間, 讀/寫 完後請 將檔案 Close, 並且避免每個 PLC Scan 都在 Read / Write 檔案, 不然 PLC Scan Time 會變很大)

要使這個範例仍夠正確運作, 請在 PC 用 NotePad (記事本) 編一個 “data55.txt” 文字檔, 共 2 行, 每行有 10 個整數值, 然後用 ftp 傳到 WP-8xx7 的 \System_Disk\ 路徑內.

“data55.txt” 文字檔 內容可如下:

-1 , 1 , 2 , 3 , 4 , 5 , -6 , 7 , 8 , 9 100001 , 200002 , +300003 , 404 , -505 , 606 , 7007 , 8008 , 9009 , 10

變數宣告:

Name	Type	Attribute	Description
RE_LOAD	Bool	Internal	設為 True 會去讀 File 一次, 宣告初值為 TRUE
TMP	Bool	Internal	暫時使用的 Boolean 變數
File_name1	Message	Internal	長度 64, 初值為 \System_Disk\data55.txt
Msg1	Message	Internal	長度 128, 會顯示 File 處理狀態
str1	Message	Internal	長度 255, 用來儲存讀到 File 內的一行字串
N_VAL[0..19]	Integer	Internal	為變數陣列, Dim 設為 20. 讀到的 20 個 Integer 值
NUM1	Integer	Internal	接收 Msg_F() 的回傳值, 若為 -1 表示格式錯誤
File1	Integer	Internal	File ID
ii	Integer	Internal	給 for 迴圈使用的 index
jj	Integer	Internal	給 另一個 for 迴圈使用的 index

ST 程式:

```
if RE_LOAD then      (* 若將 RE_LOAD 設為 TRUE, 會去讀 File 一次 *)

  RE_LOAD := FALSE ;

  File1 := f_wopen( File_name1 ) ;    (* 開啟 File 為 可讀 / 可寫 模式 *)

  if File1 = 0 then    (* 若回傳 0, 表示開啟 File 失敗 *)
    Msg1 := 'Can not Open file ' + File_name1 ;
    return ;    (* 無法開啟該 File, 用 Return 離開本 ST 程式 *)
  end_if ;

  for ii := 0 to 1 do  (* 共要讀 2 行 *)

    if f_eof(File1) = TRUE then    (* 偵測是否抵達 File 的 尾端 *)
      Msg1 := 'There should be at least 2 rows in ' + File_name1 + ' !!!' ;
      Exit ;    (* 離開 for 迴圈 *)
    end_if ;

    str1 := fm_read(File1) ;    (* 讀取 File 內的一行字串 *)

    NUM1 := Msg_N(str1, 2); (*轉換該字串為 數個 Integer 值, 並存入 2 號 Integer 陣列內
    *)

    if NUM1 <> 10 then    (* 轉換得到的 Integer 數量不是 10 個, 缺資料 .若為 -1
    表示格式錯誤 *)
      Msg1 := 'The ' + Msg(ii+1) + 'th row data format is not correct or data number is not 10 !';
      Exit ;    (* 離開 for 迴圈 *)
    end_if ;

    for jj := 0 to 9 do
      (* 從 2 號 Integer 陣列 1 到 10 位址內 取出轉換後的 Integer 值 存入 N_VAL[0..19] 內 *)
      N_VAL[ 10 * ii + jj ] := ARY_N_R( 2, jj + 1 ) ;
    end_for ;

  end_for ;

  TMP := f_close(File1) ;    (* File 只要成功開啟過, 處理完就需 close *)

  If ii = 2 then    (* 2 行資料都讀到了 *)
    Msg1 := 'Read ' + File_name1 + ' Ok ' ;
  end_if ;

end_if ;
```


10.5.4: 範例 Wpdmo_56: 將 1 ~ 255 個實數變數用 Flash Memory 來保存最終值

本範例程式為 Wdemo_56.pia 放於

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ 內 或
<http://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問
<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> > Driver 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

本 Wpdmo_56 範例程式 一開機 會從 \System_Disk\data56.txt (Flash memory) 內讀取 1 ~ 255 個實數變數的 最終值, 若檔案不存在, 則指定 每個的 初值為 0.0.

之後 只要任何一個數值有變, 就會將全部的 1 ~ 255 個新的值 存入 \System_Disk\data56.txt 內. 檔案若不存在, 本範例程式會自動 建一個新的 \System_Disk\data56.txt 檔案.

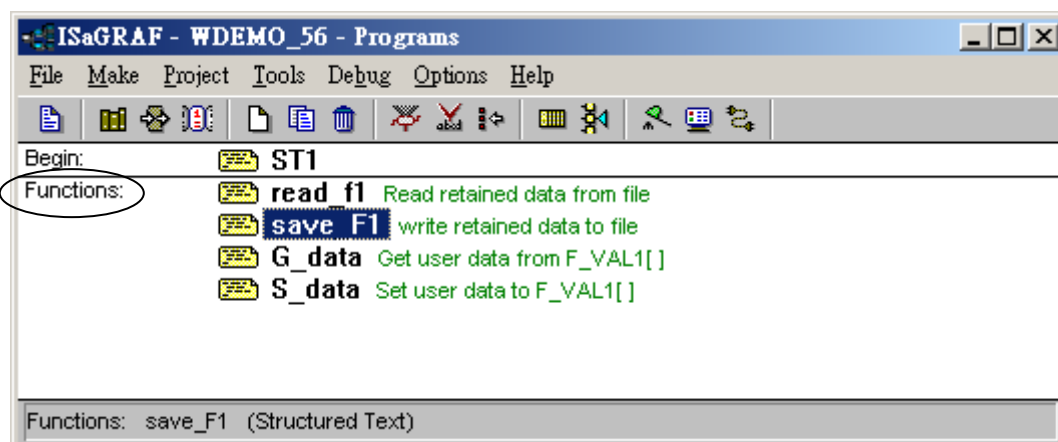
(在\System_Disk\ (Flash memory)內 寫檔案 很花 CPU 時間, 讀/寫 完後請 將檔案 Close, 並且避免每個 PLC Scan 都在 Read / Write 檔案 , 不然 PLC Scan Time 會變很大
若值一直常常更改, 則不適合用 Flash memory 來保存, 請參考第 10.1 節改使用 “新的可保留變數” 來快速保存資料)

專案程式架構:

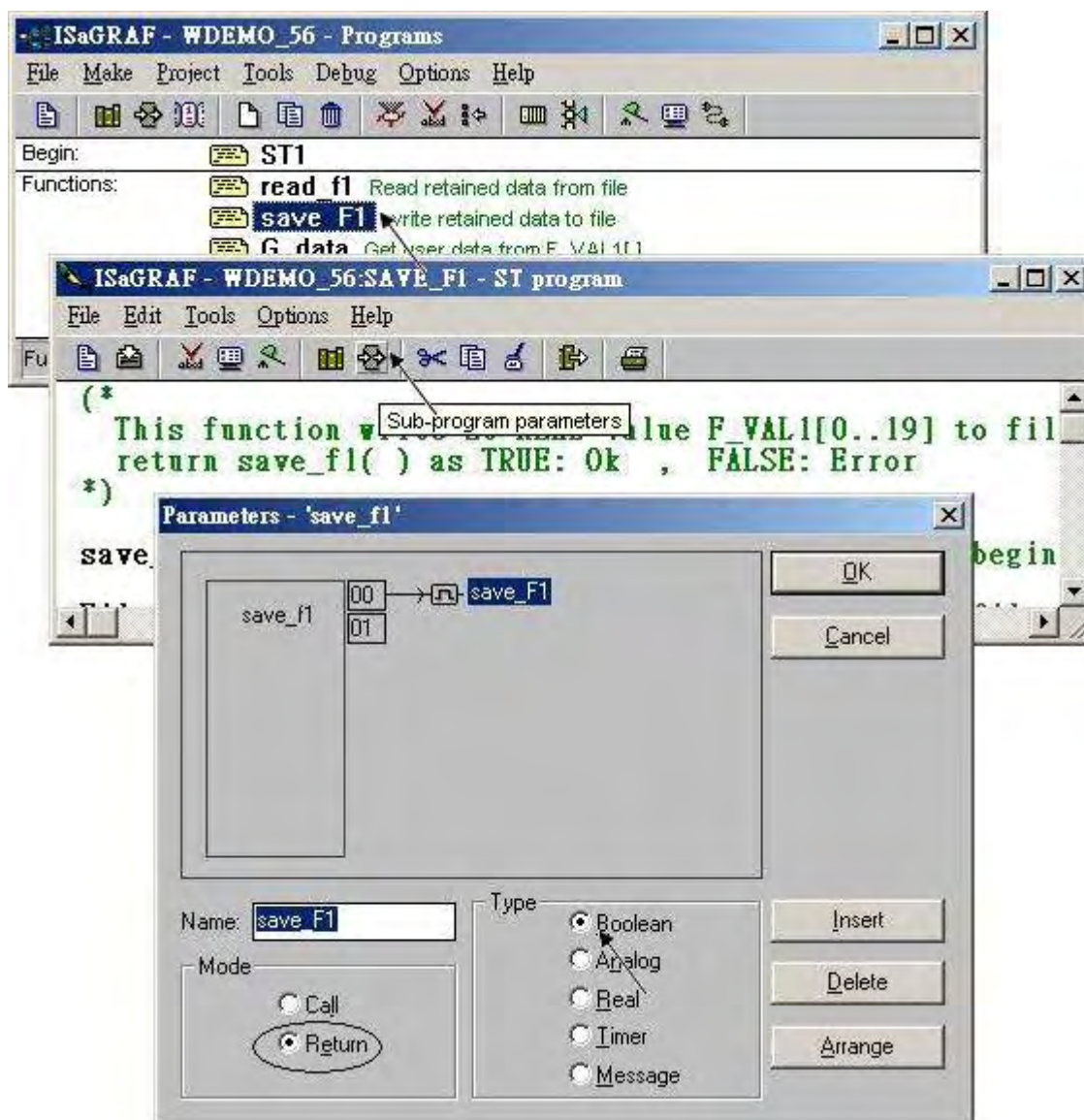
共有 5 個 ST 程式, 其中 reaf_f1, save_f1, G_data 與 S_data 為 functions

重要:

1. 本範例程式 可以依照實際應用狀況來修改 常數變數 SIZE1 之值為 1 到 255 之間的任一個值
2. 當 SIZE1 值有更改時, F_VAL1[] 與 Old_F_VAL1[] 的 "Dim" 欄位也要改成同一個值, 同時, "G_data" 與 "S_data" 內的程式也需改成符合 User 需要的.
3. 使用 \System_Disk\來保存資料有個優點為,這些資料的 File 可以預先在 PC 上編輯好, 再用 ftp 丟到 WP-8xx7 內, 本例為 \System_Disk\data56.txt. 之後將 RE_LOAD 變數設為 TRUE 一次, 它就會自動更新到 USER 的應用變數內.



本例, read_f1 , save_f1 , G_data 與 S_data 皆為 functions, 其回傳值皆定義為 Boolean 型態, 要宣告 functions 回傳值的型態, 如下 (第 15 章有詳細說明)



read_f1 與 save_f1 內有使用 區域變數 (Local variable):

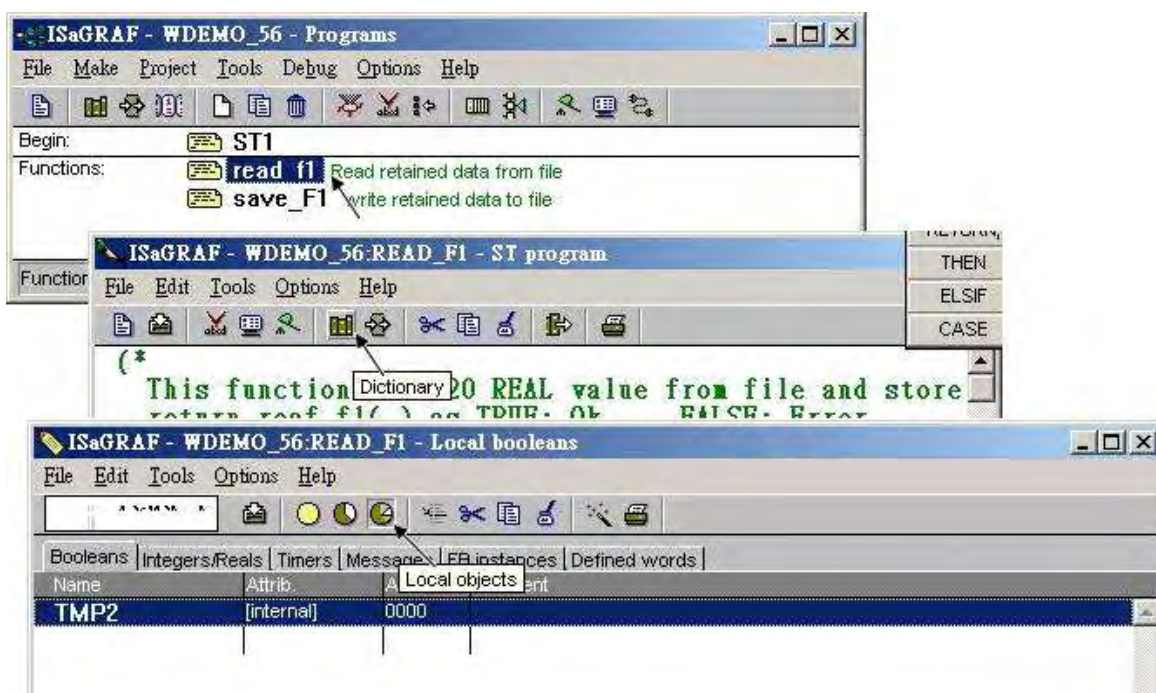
read_f1 程式的 區域變數 (Local variable) 如下:

Name	Type	Attribute	Description
TMP2	Bool	Internal	暫時使用的 Boolean 變數
ii2	Integer	Internal	給 for 迴圈使用的 index
jj2	Integer	Internal	給 for 迴圈使用的 index
num2	Integer	Internal	暫時使用的 Integer 變數

save_f1 程式的 區域變數 (Local variable) 如下:

Name	Type	Attribute	Description
TMP2	Bool	Internal	暫時使用的 Boolean 變數
ii2	Integer	Internal	給 for 迴圈使用的 index
jj2	Integer	Internal	給 for 迴圈使用的 index
num2	Integer	Internal	暫時使用的 Integer 變數

要宣告區域變數, 先雙擊 read_f1 程式進入, 之後 進入 Dictionary 內, 之後點選 “Local objects” 開始宣告 區域變數



全域變數 (Global variable)

Name	Type	Attribute	Description
SIZE1	Integer	Constant 常數	User 要保存的資料數量, 可以是 1 ~ 255 若改變此值, 需一併更改 F_VAL1[] 與 Old_F_VAL1[] 內宣告的 Dim 欄位為相同值 本例設 SIZE1 為 17
num_row1	Integer	Internal	檔案內資料共幾行, 此值由程式用 SIZE1 自動算出 本範例一行可放 10 個實數資料
Last_num1	Integer	Internal	檔案內最後一行資料的 資料有幾個 此值由程式用 SIZE1 自動算出
RE_LOAD	Bool	Internal	宣告初值為 TRUE, 所以一開機會讀 File 一次 若程式運行中 又設此值為 TRUE, 會再去讀 File 一次
TMP	Bool	Internal	暫時使用的 Boolean 變數
Data_Ok1	Bool	Internal	TRUE 表是讀 File Ok
Flag_to_save	Bool	Internal	Controller 要存 File 時, 會自動設它為 TRUE
File_name1	Message	Internal	長度 64, 初值為 \System_Disk\data56.txt
Msg1	Message	Internal	長度 128, 會顯示 File 處理狀態
str1	Message	Internal	長度 255, 操作 File 字串時會用到
F_VAL1[0..19]	REAL	Internal	為變數陣列, Dim 欄位設成與 SIZE1 之值相同
Old_F_VAL1 [0..19]	REAL	Internal	為變數陣列, Dim 欄位設成與 SIZE1 之值相同 為 F_VAL1[] 的舊值
NUM1	Integer	Internal	接收 Msg_F() 的回傳值, 若為 -1 表示格式錯誤
File1	Integer	Internal	File ID
ii	Integer	Internal	給 for 迴圈使用的 index
jj	Integer	Internal	給 另一個 for 迴圈使用的 index
Data1 ~ Data5 與 Data06 ~ Data17	REAL	Internal	用來模擬 (仿真) 為 User Data 變數, 本例因為 SIZE1 值為 17, 所以共有 17 個資料變數 User 實際的應用可以每個都使用不同的變數名稱. 若有更改名稱, G_data 與 S_data 內程式也要修改

ST 程式 ST1:

```
-----  
if RE_LOAD then (* 一開機, RE_LOAD 初值為 TRUE, 會去讀 File 一次 *)  
  
    RE_LOAD := FALSE ; (* 進來之後需馬上將 RE_LOAD 設為 FALSE *)  
  
    (* 根據 SIZE1 之值, 計算出 共有幾行資料, 與最後一行有幾個 REAL 資料 *)  
    num_row1 := SIZE1 / 10 ;  
    last_num1 := SIZE1 - 10 * num_row1 ;  
    if last_num1 <> 0 then  
        num_row1 := num_row1 + 1 ;  
    else  
        last_num1 := 10 ;  
    end_if ;  
  
    TMP := read_F1( ) ; (* 呼叫 read_f1( ) 來讀 SIZE1 個 資料到 F_VAL1[ ] 內 *)  
  
    if TMP = FALSE then (* read_f1( ) 若回傳 FALSE, 表示讀取失敗 *)  
  
        for ii := 0 to SIZE1 - 1 do  
            F_VAL1[ii] := 0.0 ; (* 讀取失敗將 SIZE1 個初值設為 0.0 *)  
        end_for ;  
  
        Data_Ok1 := FALSE ; (* 將 Data_Ok1 設為 FALSE 來表示 “讀取失敗” *)  
        Msg1 := 'File : ' + File_name1 + ' not exist or data error ! or File is open now' ;  
  
    Else (* 若讀取 File 成功 *)  
  
        Data_Ok1 := TRUE ; (* 將 Data_Ok1 設為 TRUE 來表示 “讀取成功” *)  
        Msg1 := 'Get Retained data from file Ok ' ;  
  
    end_if ;  
  
    (* 一開機時 要更新 Old_F_VAL1[ ] 之值 與 F_VAL1[ ] 相同 *)  
    for ii := 0 to SIZE1 - 1 do  
        Old_F_VAL1[ii] := F_VAL1[ii] ;  
    end_for ;  
  
    TMP := G_data( ) ; (* 每次從 File 內讀資料後 , 需更新 值 到 User Data 變數內*)  
  
end_if ;  
  
(* 每個 PLC Scan 都需 把 User Data 變數值 更新到 F_VAL1[ ] 內 *)  
TMP := S_data( ) ;
```

```

(* 在每個 PLC scan 內判斷 這 SIZE1 個資料是否有任一個值有更改 *)
for ii := 0 to SIZE1 - 1 do

    if Old_F_VAL1[ii] <> F_VAL1[ii] then (* 新/舊值不同, 表示有更改過 *)

        Flag_to_save := TRUE ; (* 將 Flag 設為 TRUE 來準備要 寫資料到 File 內*)
        Old_F_VAL1[ii] := F_VAL1[ii]; (* 值不同時, 要更新 舊值 *)

    end_if ;

end_for ;

if Flag_to_save then (* 若 Flag 被設為 TRUE, 寫 SIZE1 個資料到 File 內 *)

    TMP := save_f1( ) ; (* 呼叫 save_f1() 來寫 資料 *)

    if TMP = FALSE then (* save_f1() 回傳 FALSE, 表示寫 File 失敗,可能 File 被打開,
沒關 *)
        Msg1 := 'Can not save data to file. May be file is open now by Wincon screen !' ;
    Else (* 寫 File 成功, 將 Flag 清除為 FALSE *)
        Flag_to_save := FALSE ;
    end_if ;

end_if ;

```

ST functions 程式 – G_data :

(* 更新 User Data 變數之值 ,若 SIZE1 值有改, 或 User Data 變數名稱有改, 以下的 Code 也要修改 *)

```
Data1 := F_VAL1[0];
Data2 := F_VAL1[1];
Data3 := F_VAL1[2];
Data4 := F_VAL1[3];
Data5 := F_VAL1[4];
Data06 := F_VAL1[5];
Data07 := F_VAL1[6];
Data08 := F_VAL1[7];
Data09 := F_VAL1[8];
Data10 := F_VAL1[9];
Data11 := F_VAL1[10];
Data12 := F_VAL1[11];
Data13 := F_VAL1[12];
Data14 := F_VAL1[13];
Data15 := F_VAL1[14];
Data16 := F_VAL1[15];
Data17 := F_VAL1[16];
G_data := TRUE ; (* function 回傳 TRUE *)
```

ST functions 程式 – S_data :

(* 更新 F_VAL1[] 之值 , 若 SIZE1 值有改, 或 User Data 變數名稱有改, 以下的 Code 也要修改 *)

```
F_VAL1[0] := Data1;
F_VAL1[1] := Data2;
F_VAL1[2] := Data3;
F_VAL1[3] := Data4;
F_VAL1[4] := Data5;
F_VAL1[5] := Data06;
F_VAL1[6] := Data07;
F_VAL1[7] := Data08;
F_VAL1[8] := Data09;
F_VAL1[9] := Data10;
F_VAL1[10] := Data11;
F_VAL1[11] := Data12;
F_VAL1[12] := Data13;
F_VAL1[13] := Data14;
F_VAL1[14] := Data15;
F_VAL1[15] := Data16;
F_VAL1[16] := Data17;
S_data := TRUE ; (* function 回傳 TRUE *)
```

ST functions 程式 read_f1 :

```
-----
(* 這個 function 從 File 內讀出 SIZE1 個實數並存入 F_VAL1[ ]內
   成功: read_f1() 回傳 TRUE , 失敗回傳 FALSE *)
read_f1 := FALSE ; (* 一開始先預設為失敗 *)
File1 := f_wopen( File_name1 ) ; (* 開啟 File 為 可讀 / 可寫 模式 *)

if File1 = 0 then (* 開啟失敗, 通常為 File 不存在 *)
  return ; (* 離開此程式 *)
end_if ;

(* File 開啟成功 , 讀資料 *)
for ii2 := 0 to num_row1 - 1 do (*共 num_row1 行, 每行 10 個 REAL 值 *)

  if f_eof( File1 ) = TRUE then (* 偵測是否抵達檔案尾端 *)
    exit ; (* 若是則離開 for 迴圈 *)
  end_if ;

  str1 := fm_read( File1 ) ; (* 從 File 內讀出一行字串 *)
  NUM1 := Msg_F( str1 , 1 ); (*將該字串轉換為數個 REAL 值 , 並存放於 1 號 Float 陣列內 *)

  (* 若為 最後一行資料數量是否正確 ? 若非最後一行, 資料數量是否為 10 個 ? *)
  if ( ( ii2 = num_row1 - 1 ) and ( NUM1 <> last_num1 ) ) or
     ( ( ii2 <> num_row1 - 1 ) and ( NUM1 <> 10 ) ) then
    exit ; (* 數量不對, 離開 for 迴圈 *)
  end_if ;

  if ii2 = num_row1 - 1 then (* 若為最後一行, 設 num2 為 最後一行的資料數量 *)
    num2 := last_num1 ;
  else (* 若不是最後一行, 設資料數量 num2 為 10 個 *)
    num2 := 10 ;
  end_if ;

  (* 轉換成功, 將此行的 REAL 值 存入 F_VAL1[ ] 內 *)
  for jj2 := 0 to num2 - 1 do
    F_VAL1[ 10*ii2 + jj2 ] := ARY_F_R( 1 , jj2 + 1 ) ;
  end_for ;
end_for ;

TMP2 := f_close( File1 ) ; (* File 只要成功開啟過, 處理完就需 close *)
If ii2 = num_row1 then
  read_F1 := TRUE ; (* num_row1 行資料都已讀到, read_f1() 回傳 TRUE *)
end_if ;
-----
```

ST functions 程式 save_f1 :

```
(* 這個 function 寫 SIZE1 個實數 F_VAL1[] 到 File 內,
  成功: save_f1() 回傳 TRUE , 失敗回傳 FALSE *)

save_f1 := FALSE ; (* 一開始先預設為失敗 *)

File1 := f_creat( File_name1 ) ; (* 建立一個新 File, 若檔案已存在, 資料會刪除 *)

if File1 = 0 then
  return ; (* 建立一個新 File 失敗, 離開此程式 *)
end_if ;

(*建立一個新 File 成功, 準備寫資料進去 *)

for ii2 := 0 to num_row1 - 1 do (* 共 num_row1 行資料要寫入 *)

  str1 := ' ' ; (* 設每行字串初值為 1 個空格 *)

  if ii2 = num_row1 - 1 then (* 若為最後一行, 設 num2 為 最後一行的資料數量 *)
    num2 := last_num1 ;
  else (* 若不是最後一行, 設資料數量 num2 為 10 個 *)
    num2 := 10 ;
  end_if ;

  (* 將 REAL 資料 寫成 字串格式 *)
  for jj2 := 0 to num2 - 2 do (* 不含最後一行 *)
    str1 := str1 + REAL_STR( F_VAL1[ 10 * ii2 + jj2 ] ) + ',' ;
  end_for ;

  (* 最後一行, 結尾需為 <CR><LF> 字元 *)
  str1 := str1 + REAL_STR( F_VAL1[ 10 * ii2 + 9 ] ) + '$0D$0A' ;

  TMP2 := f_writ_s( File1 , str1 ) ; (* 將此行字串寫入 File 內 *)

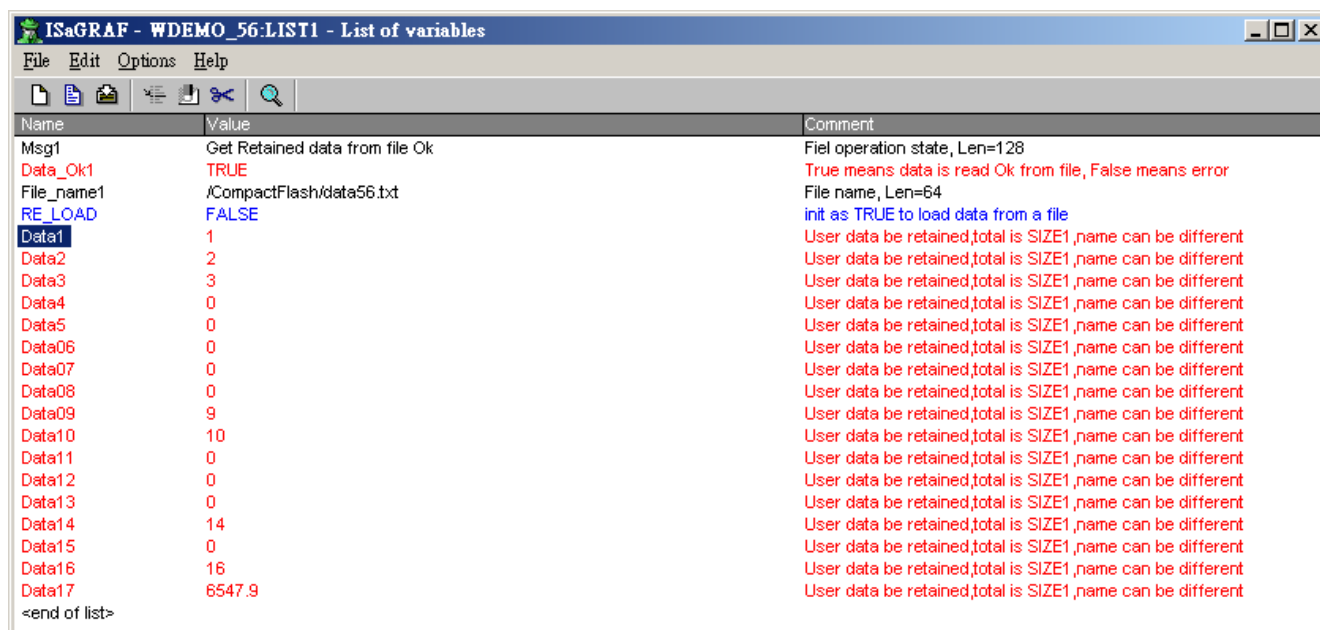
end_for ;

TMP2 := f_close( File1 ) ; (* File 只要成功開啟過, 處理完就需 close *)

save_f1 := TRUE ; (* num_row1 行資料都已寫入, save_f1() 回傳 TRUE *)
```

本 wpdmo_56 範例如何測試？

1. 將 wpdmo_56 用 ISaGRAF 下載到 WP-8xx7 後，會出現 Spy list 視窗如下



請更改 任何一個 USER Data 變數之值，改了後，它就會自動存入 \System_Disk\data56.txt 內，所以你可以在 WP-8xx7 連接的 VGA 的螢幕上開啟 此檔，會發現它的資料有跟著變動。(請不要一直開著此檔，要關掉它，不然之後 資料有變它會寫不進去，但只要有關掉，資料就會更新)

2. 將 WP-8xx7 關機後 約 5 秒 再開機，開完機後您會看到 這些 USER Data 變數之值 會是您最後輸入給它的值。

3. 在 PC 上用 NotePad (記事本) 編輯一個 data56.txt 檔類似如下，

1.1 , 2.2 , 3.3 , 4.4 , 5.5 , 6.66 , 7.77 , 8.88 , 9.99 , 10.01
0.01 , 0.02 , 0.03 , 0.04 , 0.05 , 0.06 , 0.07

將此 data56.txt 檔用 ftp 傳到 WP-8xx7 的 \System_Disk\ 目錄內。之後在 ISaGRAF 的 Spy list 視窗，雙擊 RE_LOAD，把它的值設為 TRUE。您會發現 USER Data 變數之值 會更新成上面的值。

10.5.5: 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 RAM Disk 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖

請參考第 11.3.6 節的說明 (可記錄最快達 20Hz 的資料).

第 11.3.10 節 的方法 可記錄最快達 100Hz 的資料.

10.6: 偵測 Controller Fault

有一些事件會造成 “Controller Fault”. 例如, 值除以 0 或 從 EEPROM 或 S256 或 File 內讀 浮點數(實數), 但卻沒有 浮點數 存放在裏面.

ICP DAS ISaGRAF 控制器從以下驅動程式版本起支持 Controller Fault 偵測功能.
(VP-2117, μ PAC-7186EG, iP-8xx7, WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6 都有支持)

I-7188EG	2.05	I-7188XG	2.04
I-8417/8817/8437/8837	3.07	W-8037/8337/8737	3.18

有 2 種型式的 Controller Fault. 一種稱為 Global fault. 一種稱為 Local fault.

當發生 Global fault 時, ISaGRAF 程式會停止. 等待新的程式來下載.

當發生 Local fault 時, ISaGRAF 程式會繼續 Run.

PC / HMI / OPC Server 可使用 Modbus 通訊協議 取得 控制器的 Fault 狀態.

Word 位址編號 9999 內存放 有無發生 Controller Fault. 0: 正常 , 1: 發生 Controller fault.

R_MB_ADR(1 , 9999) 取得 “有無發生 Controller Fault”

Word 位址編號 9998 內存放 Controller Fault 的型態.

R_MB_ADR(1 , 9998) 取得 ”Controller Fault 型態”

101 : Global fault

(以下皆為 Local fault)

102 : S_R_R Float error

103 : R_MB_REL Float error

104 : INT_REAL Float error

105 : RETAIN_F Float error

106 : RETAIN_X Float error

107 : Real 除以 0

108 : Integer 除以 0

109 : RETAIN_A Float error

110 : Real 乘法結果 超出 32-bit float 值的範圍

111 : Real 除法結果 超出 32-bit float 值的範圍

112 : Real 加法結果 超出 32-bit float 值的範圍

113 : Real 減法結果 超出 32-bit float 值的範圍

114 : EEP_F_R Float error

115 : EBUS_F_R Float error

116 : FBUS_F_R Float error

117 : F_REAF_F error (只有 Wincon-8xx7 使用)

118 : 在第 0 到第 7 槽找不到 相對應的 I-87K I/O 卡

119 : ARY_F_R Float error

121 : ANA() 操作錯誤, 比如 ANA(1.23E20) , ANA(-2.0e25)

122 : TMR() 操作錯誤, 比如 TMR(1.23E20) , TMR(-100)

123 : Real 運算錯誤, 比如 POW(1.23e20 , 3.0) , EXPT(5.0 , 10000000)

124 : PID_AL() 運算超出 32-bit float 值的範圍

125 : REAL 變數陣列 運算錯誤. 可能是程式內使用的陣列 Index 超出宣告的範圍

當 Controller Fault 發生, 取得 Controller 狀態後, 可用以下函式來把 9999 & 9998 號資料清為 0.
W_MB_ADR(1, 9999, 0)
W_MB_ADR(1, 9998, 0)

請參考以下的 ST 程式範例

範例:

(* Tmp 宣告為 Boolean, Internal *)

(* is_fault 與 fault_type 宣告為 Integer, Internal *)

(* PC/HMI 可使用 Modbus 通訊協議 詢問 9999 與 9998 號的 Word, 來取得 Controller fault 的狀態 *)

is_fault := R_MB_ADR(1,9999); (*程式內取得 Controller 狀態, 0:正常 ,1:發生 Controller Fault *)

(* Controller fault 的狀態

若回傳值為 101 : Global fault : 程式會停掉, 此時只有 PC / HMI 可使用 Modbus 通訊協議 詢問 9999 與 9998 號的 Word, 來取得 Controller fault 的狀態 *)

fault_type := R_MB_ADR(1,9998); (* 程式內取得 Controller Fault 的型態 *)

if is_fault = 1 then (* 發生 "Local Fault" 時 *)

(* ... *) (* "Local Fault" 時的處理 *)

(* 以下這行只適用於 Wincon-8x37/8x36: 強迫程式停止 且 第 1 到 7 槽的 output 都 Reset *)

(* tmp := Stop_APL(); *)

(* 當 Controller Fault 發生, 取得 Controller 狀態後, 把 9999 & 9998 號資料清為 0 *)

tmp := W_MB_ADR(1, 9999, 0);

tmp := W_MB_ADR(1, 9998, 0);

end_if;

第 11 章 ISaGRAF 程式範例與 FAQ

當您拿到 ISaGRAF 控制器時，我們會附上一系列的 ISaGRAF 範例程式，這些範例程式會教導您如何在 ISaGRAF Workbench 應用軟體中開發 控制器 的應用程式。

11.1: 安裝 ISaGRAF 程式範例

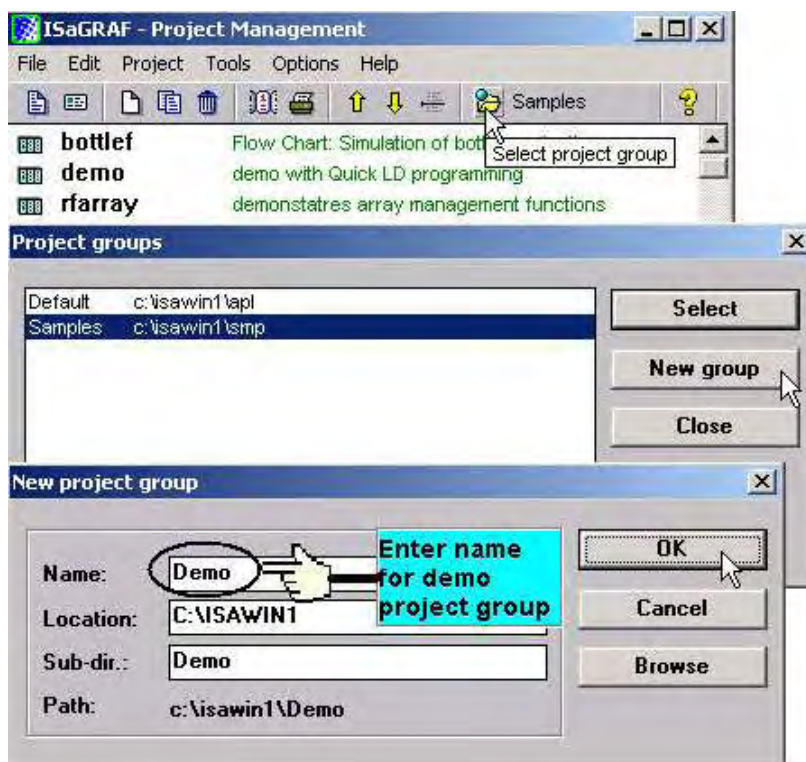
ISaGRAF 的範例程式放在 ICP DAS 的 CD-ROM 內，附在控制器的 出貨包裝盒內。

I-8xx7:	I-8000 CD-ROM: \napdos\isagraf\8000\demo\
I-7188EG, 7186EG:	I-8000 CD-ROM: \napdos\isagraf\7188eg\demo\
I-7188XG:	I-8000 CD-ROM: \napdos\isagraf\7188xg\demo\
iP-8xx7:	I-8000 CD-ROM: \napdos\isagraf\ip8000\demo\
VP-2117:	I-8000 CD-ROM: \napdos\isagraf\vp2k\demo\
XP-8xx7-CE6:	XP-8xx7-CE6 CD-ROM: \napdos\isagraf\xp-8xx7-ce6\demo\
WP-8xx7:	WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\
VP-25W7 / VP-25W7:	VP-2xW7 CD-ROM: \napdos\isagraf\vp-25w7-23w7\demo\

你也可從網路上取得範例程式:

iP-8xx7 :	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/ip8000/
µPAC-7186EG :	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/7188eg/demo/
VP-2117 :	ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vp2k/demo/
WP-8xx7 :	ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/
VP-25W7/23W7:	ftp://ftp.icpdas.com/pub/cd/vp-25w7-23w7/napdos/isagraf/vp-25w7-23w7/demo/
XP-8xx7-CE6 :	ftp://ftp.icpdas.com/pub/cd/xp-8xx7-ce6/napdos/isagraf/xp-8xx7-ce6/demo/

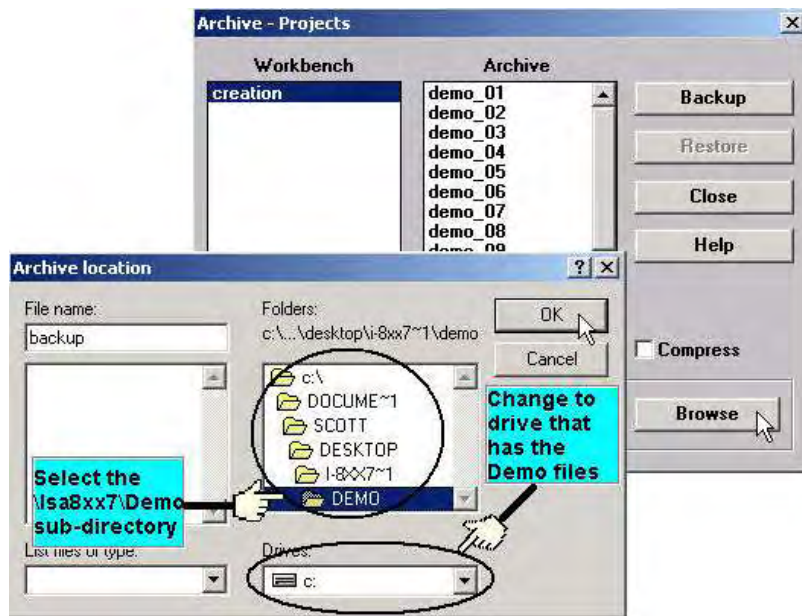
建議您在安裝 ISaGRAF 範例程式時，新建立一個“ISaGRAF Project Group”，將所有的範例程式安裝到這群組底下。



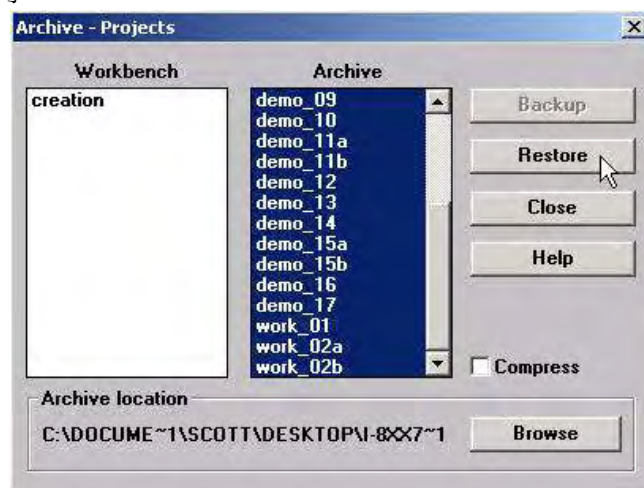
在您安裝範例程式到群組之前，您必須先開啟 “ISaGRAF Project Management”，然後選擇 [Tool]→[Archive]→[Projects]。



當你按下 “Projects”，會開啟 “Archive Projects”視窗。按下 “Browse”鍵，選擇範例檔案所放置的目錄(例如 I-8xx7, 請到 I-8000 CD-ROM 的 \Napdos\ISaGRAF\8000\Demo)



開啟 demo 目錄後，先按著 “demo_01”檔案，然後按住 “Shift”鍵不放，再用滑鼠點選最後一個檔案，此時整個 demo 目錄的檔案將被選取，最後按下 “restore”鍵。所有的 demo 檔案就會安裝在你所建立的目錄了。



11.2: ISaGRAF 範例程式列表

此節只列出 VB 範例程式， μ PAC-7186EG, iP-8xx7, WP-8xx7 與 XP-8xx7-CE6 系列的範例程式，其它 PAC 型號的範例程式請參考各別的“快速上手手冊”

所有型號最新範例程式列表，請參考網頁：

http://www.icpdas.com/products/PAC/i-8000/isagraf_demo_list.htm#VBNET

www.icpdas.com > product > [solutions](#) > [software](#) > [Development Tools](#) > [ISaGRAF](#) > Demo Files/
範例程式

Visual Basic 範例程式：

I-8000 CD-ROM:\napdos\isagraf\vb_demo\
ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vb_demo

範例名稱	說 明 (VB)	使用的 I/O 板卡 或 其他設備
Demo_1	PC 使用 Modbus TCP/IP 協定 存取 I-8437/8837 內的資料	I-8437/8837 I-8054
Demo_2	PC 使用 Modem + 電話線 存取 遠方 I-8437/8837 內的資料 (請參閱第 13 章)	I-84x7/88x7 I-87064 Modem Phone line
Demo_3	PC 跑 VB.net 2005 或 VB 6.0 用 Modbus TCP/IP 連 Controller (可以是 I-8x37-80, I-7188EG, μ PAC-7186EG, VP-2117, iP-8x47, WP-8xx7, XP-8xx7-CE6 或 VP-25W7 / VP-23W7) 參考 http://www.icpdas.com/faq/isagraf.htm FAQ-051 與 052	
Demo_4	PC 跑 VB 6.0 用 Modbus TCP/IP 連 Controller (可以是 I-8x37-80, I-7188EG, μ PAC-7186EG, VP-2117, iP-8x47, WP-8xx7, XP-8xx7-CE6 或 VP-25W7 / VP-23W7+ I-7018z) 來顯示溫度輸入值. (參考 第 11.3.9 節)	I-7018z
Demo_6	PC 跑 VB 6.0 用 Modbus TCP/IP 連 Controller (I-8437-80 / 8837-80), 參考 第 11.3.7 節	Slot2: I-8024 Slot3: I-8017H

μPAC-7186PEG、μPAC-7186EG、I-7188EG/XG 範例程式:

μPAC-7186(P)EG、I-7188EG :

I-8000 CD-ROM: \napdos\isagraf\7188eg\demo

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/7188eg/demo/>

I-7188XG :

I-8000 CD-ROM: \napdos\isagraf\7188xg\demo

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/7188xg/demo/>

範例名稱	說明 (7186/7188)	使用的 I/O 板卡 或其他設備
Demo_01	接收訊息 並 回傳 至 Com2 或 Com3	X503/4/5/6 擴充卡
Demo_02	寫入一個字串至 X503 的 Com5 和 Com6	X503
Demo_03	接收訊息 並 回傳 至 Com6 或 Com7 (使用陣列讀寫)	X503
Demo_04	使用 I-7000 和 插入 X-board 至 I-7188EG	Bus7000b X107
Demo_05	計時器控制, TP, TON, TOF	X304
Demo_06	顯示一個 值 至 S-MMI, VAL10LED	X304
Demo_07	控制 X107 和 I-7060D Relay IO	Bus7000b X107
Demo_08	接收訊息並 回傳 至 X507/8/9 的 Com3 及 控制 DO .	X507_8_9
Demo_09	使用 S-MMI 與 Timer 控制 tStart , tStop , Reset 為 0	
Demo_10	使用 S-MMI	X107
Demo_11	連結其他的 Modbus RTU 設備	mbus
Demo_12	轉換 I-7000 及顯示 類比輸入值 至 S-MMI 供學習套件使用	Bus7000b
Demo_13	轉換 I-7000 並顯示類比輸入值 至 S-MMI供學習套件使用	Bus7000b
Demo_18	PID 控制, PID_AL , 無法在 PC 模擬(仿真), 需下載到控制器內	
Demo_21	寫入一個字串 至 Com3 和 Com4	Xbi8 (虛擬 D/I) X50x
Demo_22	接收訊息 並 回傳 至 Com3 及 Com4	X50x
Demo_23	從 PC 接收一個使用者定義的通訊協定	X50x
Demo_35a	時間同步 : 35A (與 35B demo 搭配使用) 更新本控制器的日期時間可以同步更新 B 控制器	Fbus_m
Demo_35b	時間同步: 35B (與 35A demo 搭配使用)	Fbus_s
Demo_36	取得 I-7188EG 驅動程式版本	
Demo_41	記錄警報 (文字) 至 X607/X608 而且 PC 能以 "ICPDAS UDloader" 載入記錄的警報	X607_608 Xbi8 (虛擬 D/I) Xbo8 (虛擬 D/O)
Demo_43	SMS 簡訊範例, 請在 dictionary 的 message 型態中	SMS

範例名稱	說明 (7186/7188)	使用的 I/O 板卡 或 其他設備
	宣告您的手機號碼	
Demo_43a	與 Demo_43 類似, 但可發給 2 支以上的手機	SMS
Demo_44	PC 下載資料至 X607/X608	X607_608 Xbo8 (虛擬 D/O)
Demo_50	PWM I/O 範例. (Pulse Width Modulation)	X107
Demo_48a	冗餘 Redundant: I-7188XG redundant Master	Bus7000b Fbus_m
Demo_48b	冗餘 Redundant: I-7188XG redundant slave	Bus7000b Fbus_s
Demo_51a	冗餘 Redundant: I-7188EG redundant Master	Bus7000b Ebus_m
Demo_51b	冗餘 Redundant: I-7188EG redundant slave	Bus7000b Ebus_s
Demo_61	DI 計數器, 使用 DI_CNT, I-7188 + X107 當 DI 訊號發生時, 做指定動作	X107
Demo_70	當警報 1~8 發生時, 傳送字串 至 COM2/COM3 (使用陣列讀寫)	
Demo_72	Controller 用 RS-485 連接 1 顆 10 通道 I-7018z, 並將溫度值用 PC 上的 VB 6.0 程式顯示出來. 請參考第 11.3.9 節	I-7018z

注意: Demo_18 使用 CJ 公司提供的 PID_AL. 請參考:

“ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/english_manu/pid_al.complex_pid_algorithm_implementation.pdf”.

iP-8xx7、I-8417/8817/8437/8837 範例程式:

iP-8xx7 :

I-8000 CD-ROM: \napdos\isagraf\ip8000\demo 或
<ftp://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/isagraf/ip8000/demo/>

I-8417/8817/8437/8837:

I-8000 CD-ROM: \napdos\isagraf\8000\demo 或
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

範例名稱	說明 (iP-8xx7 / I-8xx7)	使用的 I/O 板
Demo_01	Timer 的控制, TP, TON, TOF	Push4Key, Show3Led
Demo_01a	於事件發生後某個時間點(例如 3 秒後) 去做某件事.	Push4Key, Show3Led
Demo_02	啟動, 暫停, 及重置 Timer, tStart , tStop	Push4Key, Show3Led
Demo_03	讀 / 寫 日期及時間 SYSDAT_R, SYSDAT_W, SYSTIM_R, SYSTIME_W 在某個時段輸出, 例如. Moday, 09:00 ~ 18:00, Sunday, 10:00 ~	NONE
Demo_04	計算周期時間	NONE
Demo_05	方波閃爍輸出	Push4Key, Show3Led
Demo_06	改變輸出模式	Push4Key, Show3Led
Demo_07	在 S-MMI 上顯示 1 個數值, VAL10LED, tStart , tStop	Push4Key, Show3Led
Demo_08	從 S-MMI 輸入數值, INP10LED	Push4Key, Show3Led
Demo_09	四則運算, +, -, *, /	NONE
Demo_10	顯示類比輸入值到 S-MMI	I-87017, I-87024, Push4Key
Demo_11a	Fbus Master, NET_ID = 1	Fbus_m, Push4Key, Show3Led
Demo_11b	Fbus Slave, NET_ID = 2	Fbus_s, Push4Key
Demo_12	使用 COM3 接收 PC 傳來的通訊 資料	Show3Led
Demo_13	每隔 3 秒由 Com3 寫出通訊資料	I-87017
Demo_14	轉換 I-7000 & I-87xxx 模組的通訊格式到 Modbus 格式	Bus7000b
Demo_15a	連到其他的 Modbus 設備	Mbus
Demo_15b	模擬成 Modbus 設備 給 Demo_15a 使用	None
Demo_16	每過 1 秒送出 Modbus 通訊命令給其它的 Modbus RTU 設備	Push4Key, Mbus
Demo_17	讀 / 寫 EEPROM	None
Demo_18	PID 控制, PID_AL , 無法在 PC 模擬(仿真), 需下載到 控制器 內	None
Demo_21	寫出字串到 Com5 及 Com6	Push4Key, Show3Led

範例名稱	說明 (iP-8xx7 / I-8xx7)	使用的 I/O 板
Demo_22	從 Com5 或 Com6 接收通訊資料並 Echo 回去	Show3Led
Demo_23	從 PC 接收 User 自定的通訊格式	Show3Led
Demo_27	運動控制: x 軸, slot 0: I-8091, Slot1:I-8090, Napdos\ISaGRAF\8000\Driver\motion.pdf	I-8091 I-8090 Show3Led
Demo_27a	X 軸移動指定數目的 pulse	I-8091
Demo_28	運動控制: x , y 軸同動, slot0: I-8091, slot1:I-8090, Napdos\ISaGRAF\8000\Driver\motion.pdf	I-8091 I-8090 Show3Led
Demo_29	每隔 75 秒, 記錄 1200 個 短整 (2 bytes) 資料, 然後使用 Com3 回報給 PC	I-87017
Demo_30	每隔 18 秒, 記錄 2880 個 短整 (2 bytes) 資料, 然後使用 Com3 回報給 PC	I-8017h
Demo_33	從 Com3 讀 / 寫 User 自行定義的通訊格式	Show3Led
Demo_35a	時間同步 : SA 更改 SA 此台的時間會將 SB 那台的時間也校正過來	Fbus_m
Demo_35b	時間同步 : SB	Fbus_s
Demo_37	Spotlight demo (簡易圖控), 請參考第 14 章	Push4Key Show3Led
Demo_38	連線到 MMICON , demo 1, 請參考第 16 章	
Demo_39	連線到 MMICON , demo 2, 請參考第 16 章	
Demo_40	每分鐘儲存 8 個 A/I (binary) 到 S256 內, 之後 PC 可使用 "ICPDAS UDownloader" 程式工具 上載資料出來	
Demo_41	儲存警報(字串)到 S256/512 內, 之後 PC 可使用 "ICPDAS UDownloader" 程式工具 上載資料出來	
Demo_42	每分鐘儲存 8 個 A/I (binary) 到 S256 內, 之後 PC 可使用 "ICPDAS UDownloader" 程式工具 上載資料出來	
Demo_43	SMS 短訊息收發範例, 請更改為自己可用的號碼	SMS
Demo_43a	與 Demo_43 類似, 但可發給 2 支以上的手機	SMS
Demo_44	PC 下載資料到 S256/512 的範例	
Demo_46	運動控制: Pulse move at a specified speed	I-8091 I-8090 Push4Key
Demo_49a	備援系統 (冗余): 8437/8837 redundant Master	Bus7000 Ebus_m
Demo_49b	備援系統(冗余): 8437/8837 redundant slave	Bus7000 Ebus_s
Demo_50	PWM I/O 請參考第 3.7 節	I-8055
Demo_52	取得 I-8K D/I 卡的 counter 值, 請參考第 3.8 節 (Counter 值有保存起來)	I-8051 Push4Key
Demo_53	取得 I-8K D/I 卡的 counter 值, 請參考第 3.8 節 (Counter 值 沒有 保存起來)	I-8051 I-8056 Push4key
Demo_54a	Modbus Master 功能 (Master)	
Demo_54b	給 Demo_54a 來連的 (Slave)	

範例名稱	說明 (iP-8xx7 / I-8xx7)	使用的 I/O 板
Demo_55	PWM I/O demo 2. 請參考第 3.7 節	I-8055
Demo_61	I-8K DI Counter demo, 8xx7 + 8051,, 請參考第 3.8 節 當 DI 訊號發生時去做某件事	I-8051
Demo_70	當警報 1~8 發生時, 傳送字串 至 COM2/COM3 (使用陣列讀寫)	
Demo_71	使用 I-8437-80 或 I-8837-80, 每 0.05 秒記錄 I-8017H 的 1~4 個電壓值於 S-256 / S-512 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1~4 條趨勢圖. 請參考第 11.3.7 節	I-8024 I-8017h
Demo_72	Controller 用 RS-485 連接 1 顆 10 通道 I-7018z, 並將溫度值用 PC 上的 VB 6.0 程式顯示出來. 請參考第 11.3.9 節	I-7018z

注意:

Demo_18 使用 CJ International for evaluation 所提供的 PID_AL, 請參考

“CD\Napdos\isagraf\8000\english_manu\ PID_AL.Complex PID algorithm implementation.htm”.

WP-8xx7 範例程式:

有個比 FAQ-131 更好用的 Soft-GRAF Studio, 請參考 FAQ-146.

WP-8xx7 CD-ROM: \napdos\isagraf\8xx7\demo\ 或

<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

專案名稱	說明 (WP-8xx7)	使用的 I/O 模組
sofgr_01	Soft-GRAF HMI 簡單範例. (sofgr_01~sofgr_08: 中文 FAQ-131)	
sofgr_02	Soft-GRAF HMI 元件範例: 使用 18 個 HMI 元件.	
sofgr_03	Soft-GRAF 範例. 顯示 10 個溫度量測值 與 8 個 D/I 值 與控制 8 個 D/O 值. 資料量少於 255.	
sofgr_04	Soft-GRAF 範例. 從 file 內讀取工作參數 或 將修改的 工作參數存入 File 內. 資料格式為實數, 只處理 1 個 File. 資料量少於 255.	
sofgr_05	Soft-GRAF 範例. 從 file 內讀取工作參數 或 將修改的 工作參數存入 File 內. 資料格式為整數, 只處理 1 個 File. 資料量少於 255.	
sofgr_06	Soft-GRAF 範例. 從 file 內讀取工作參數 或 將修改的 工作參數存入 File 內. 資料格式為實數, 可處理多個 File. 資料量少於 255.	
sofgr_07	Soft-GRAF 範例. 從 file 內讀取工作參數 或 將修改的 工作參數存入 File 內. 資料格式為整數, 可處理多個 File. 資料量多達 1000 (可更多).	
sofgr_08	Soft-GRAF 範例. 從 file 內讀取工作參數 或 將修改的 工作參數存入 File 內. 資料格式為實數, 可處理多個 File. 資料量多達 1000 (可更多).	
example1	Web HMI 人機頁面 簡單範例	slot 0: I-87055W
wp_vb01	WP-8xx7 的 VB.net 2008 demo 01: DI/DO 範例 請參考第 6 章	slot 0: I-87055W
wp_vb02	WP-8xx7 的 VB.net 2008 demo 02: AI/AO 範例 請參考第 6 章	slot 1: I-87024W slot 2: I-8017HW
wp_vb03	WP-8xx7 的 VB.net 2008 demo 03: 讀/寫 long integer, float 及 Timer. 請參考第 6 章	
wpdmo_01	WinPAC demo_01: 如何在 WinPAC 內 讀/寫 File 資料 (float 值)。(中文 FAQ > 060)	
wpdmo_02	WinPAC demo_02: 如何在 WinPAC 內 讀/寫 File 資料 (long integer 值)。(中文 FAQ > 060)	
wpdmo_03	依時間間隔輸出, 如 SYSDAT_R, SYSDAT_W, SYSTEM_R, SYSTEM_W (ST+QLD)	
wpdmo_04	WinPAC demo_04: 使用者自訂 Modbus 通訊協定 (不使	

專案名稱	說明 (WP-8xx7)	使用的 I/O 模組
	用 "Mbus")	
wpdmo_05	指定事件發生數秒後觸發某動作. (英文 FAQ > 017)	slot 0: I-87055W
wpdmo_06	使用 Message 陣列 - MsgAry_r , MsgAry_w	
wpdmo_07	轉換 float 值為 string , 使用 real_str 與 rea_str2	
wpdmo_08	PID 控制 , 參考光碟 : \napdos\isgraf\wp-8xx7\english_manu\ "PID_AL...htm"	
wpdmo_09	儲存與備份 boolean / long integer 值 到/從 檔案	
wpdmo_10	儲存與備份 boolean / long integer 值 到/從 EEPROM	
wpdmo_11	以 \Micro_SD 為目錄 , 每隔 10 秒儲存 3 個值到 3 個檔案 , 每個月改變檔名。	
wpdmo_14	Retain variable: Retain_b, Retain_N, Retain_f, Retain_t (中文 FAQ > 074)	
wpdmo_16	以 \Micro_SD 為目錄 , 每隔 1 分鐘儲存 3 個值到 1 個檔案 , 每天改變檔名。	
wpdmo19	當警告發生 , 傳遞 UDP 字串到 PC , Time_Gap 為 1 秒 (使用變數陣列) (參考 "ISaGRAF 進階手冊" 第 19.2 節)	slot0: I-87055W
wpdmo19a	3 秒後傳遞 UDP 字串到 PC , Time_Gap 為 250 ms (請參考 "ISaGRAF 進階手冊" 第 19.2 節)	slot0: I-87055W
wpdmo19b	3 秒後傳遞 UDP 字串到 PC , Time_Gap 為 250 ms (wpdmo19a 較佳) (見 "ISaGRAF 進階手冊" 第 19.2 節)	slot0: I-87055W
wpdmo_20	投過 UDP/IP 從遠端 PC 或控制器接收字串	
wpdmo_21	使用 "com_MRTU" 來 停止/啟動 Modbus RTU slave 埠	
wpdmo_22	PWM I/O 範例(WinPAC 脈寬調整, 最小 2 秒)	slot 0: I-8055W
wpdmo_23	使用 COMOPEN, COMSTR_W , 每秒傳遞時間字串到 COM3:RS-232 。 (中文 FAQ > 059)	
wpdmo_24	當發生警報 1~8 , 傳遞字串到 COM3	slot 0: I-87055W
wpdmo_26	移動 WP-8xx7 插槽 1 的 I-8091W 的 X 軸時脈 (請參考 "ISaGRAF 進階手冊" 第 18 章)	slot 1: I-8091W
wpdmo_27	運動控制: x 軸 (請參考 "ISaGRAF 進階手冊" 第 18 章)	slot 1: I-8091W slot 2: I-8090W
wpdmo_28	運動控制 : x-y 軸 (請參考 "ISaGRAF 進階手冊" 第 18 章)	slot 1: I-8091W slot 2: I-8090W
wpdmo_29	當給予 CMD 時 , 移動到絕對位置 (請參考 "ISaGRAF 進階手冊" 第 18 章)	slot 1: I-8091W slot 2: I-8090W
wpdmo_30	WP-8xx7(10.0.0.102) 連結 2 個 I-8KE8 乙太網路擴充單元 + I/O 模組 , 一個是 10.0.0.108 , 一個是 10.0.0.109 。 (中文 FAQ > 042)	

專案名稱	說明 (WP-8xx7)	使用的 I/O 模組
wpdmo_31	WP-8xx7(10.0.0.2) 連結 1 個 I-8KE8 乙太網路擴充單元 + I/O 模組 (10.0.0.109)。 (中文 FAQ > 042)	
wpdmo_32	設定 WP-8xx7 為 TCP/IP Client 並連結到其他 TCP/IP server (1 個連接)(見“ISaGRAF 進階手冊”第 19.3 節)	slot 0: I-87055W
wpdmo_33	同 Wpdmo_32，但當事件持續超過 3 秒時，只傳遞訊息	slot 0: I-87055W
wpdmo_36	從 Modbus RTU 設備讀出實數值 (中文 FAQ - 047, 075)	
wpdmo_37	將實數值寫到 Modbus RTU 設備 (中文 FAQ - 047,075)	
wpdmo_38	使用 Modbus 函式 6 來寫 16 個位元到設備 (中文 FAQ - 046 與 075)	
wpdmo_39	WP-8xx7 + I-8172W 連接 FRNET I/O 模組 (中文 FAQ - 082)	
wpdmo_41	使用 COM3 連接 1: M-7053D + 2: M-7045D (MBRTU 格式, baud=9600) (請參考“ISaGRAF 進階手冊”第 21 章)	
wpdmo_42	使用 COM3 連接 1: M-7053D 來讀取 DI 計數值 (MBRTU 格式, baud=9600)	
wpdmo_43	使用 COM3 連接 1: M-7017R + 2: M-7024 (MBRTU 格式, baud=9600)	
wpdmo_44	使用 COM3 連接 1: M-7017RC, Current input, +/- 20mA, 4-20mA (Modbus 格式)	
wpdmo_45	使用 COM3 連接 1: M-7019R (設定為 T/C K-type input) (MBRTU 格式, baud=9600)	
wpdmo_46	使用 COM3 連接 1: M-7080 (MBRTU 格式, baud=9600)	
wpdmo_48	VB.net 2005 範例 - "MBTCP_demo" (中文 FAQ > 051)	
wpdmo_50	非線性轉換，例如：給 P 值，求 V 值 (P, V 的關係列於檔案)	
wpdmo_51	使用 str_real 函式，從檔案讀取 10 個實數，共 10 列，每列 1 個實數。	
wpdmo_52	Msg_F 函式範例，ISaGRAF PAC 的驅動程式需為下列版本或更新版本。 I-8xx7: 3.19 版; I-7188EG/XG: 2.17/2.15 版; W-8xx7: 3.36 版; WP-8xx7:	
wpdmo_53	Msg_N 函式範例，ISaGRAF PAC 的驅動程式需為下列版本或更新版本。 I-8xx7: 3.19; I-7188EG/XG: 2.17/2.15; W-8xx7: 3.36; WP-8xx7:	

專案名稱	說明 (WP-8xx7)	使用的 I/O 模組
wpdmo_54	使用 msg_f 函式，從檔案讀取 20 個實數，共 4 列，每列 5 個實數。 (中文 FAQ > 060)	
wpdmo_55	使用 msg_n 函式，從檔案讀取 20 個整數，共 2 列，每列 10 個整數。	
wpdmo56	保存 17 個實數到檔案中，共 2 列，每列 10 個實數。	
wpdmo56a	保存 2 個布林 + 17 個實數到檔案，2 列，每列 10 個實數。	
wpdmo56b	保存 25 個整數到檔案中，2 列，每列 10 個整數。	
wpdmo56c	保存 2 個布林 + 25 個整數到檔案，2 列，每列 10 個實數。 (中文 FAQ > 060)	
wpdmo56d	保存 17 個實數 + 2 個布林 + 10 個整數到 2 個檔案，每列 10 個值。	
wpdmo56e	保存超過 255 個實數、255 個布林、255 個整數到 2 個檔案，最多 1024 個值。	
wpdmo_61	I-8xx7, WP-8xx7: 透過 UDP 主動資料回報到 PC。控制器=10.0.0.103，PC=10.0.0.91	
wpdmo_62	使用乙太網路埠發送 email (不附加檔案，寄給一個收件者) (中文 FAQ > 067, 071, 072, 076 或 077)	
wpdmo_63	只有 WP-8xx7 與 W-8xx7 有此功能: 發送可夾帶附件的 email 給一個收件者 (中文 FAQ > 067, 071, 072 或 077)	
wpdmo64a	工作站 1001, 使用乙太網路對多台控制器做時間校正。	
wpdmo64b	工作站 1002, 使用乙太網路對多台控制器做時間校正。	
wpdmo65a	WP-8xx7 應用: 每分鐘記錄溫度到檔案一次，每天將記錄檔以 email 寄出。 (中文 FAQ > 067, 71, 72, 76, 77)	slot 2: I-87018z
wpdmo65b	WP-8xx7: 同 wdm_65a，但增加時間校正與發出報告給 PC。 (中文 FAQ > 067, 071, 072, 076 或 077)	slot 2: I-87018z
wpdmo_66	每隔 20ms 記錄 I-8017HW 模組 1~ 4-Ch. 的電壓，並以 email 發送記錄檔	slot 2: I-8024W slot 3: I-8017HW
Wpdmo_70	FRnet 應用: WP-8xx7 或 iP-8447。 slot1: I-8172W, Port0, FR-2057(adr=4), FR-2053(adr=8)	slot 1: I-8172W FR-2057, FR-2053
wpdmo71a	CAN 應用: 以 WP-8xx7 的 COM4 連接 I-7530 -- "CANopen" ID=1 設備 (8DI, 8DO, 4AO, 8AI)。 (中文 FAQ > 086)	
wpdmo71b	類似 wdm_71A，但連接 2 個 I-7530，一個連接 COM5，一個連接 COM6。	
wpdmo71c	WP-8xx7 COM4 – I-7530 – 使用 CAN 設備取得字串(也包含 float 或 integer 資料)	

專案名稱	說明 (WP-8xx7)	使用的 I/O 模組
wpdmo71d	類似 wdm0_71c，但連接兩個 I-7530，一個連接 COM5，一個連接 COM6。	
wpdmo71e	WP-8xx7: 以 COM5 連接 I-7530 來連接 CANopen 設備，以 COM6 連接 I-7530 來連接 CAN 設備。	
wpdmo72a	新的 WP-8xx7 可熱插拔備援系統(冗餘系統): 搭配 RU-87P4 + I-87K I/O (不連接 Touch HMI) (中文 FAQ > 093)	
wpdmo72b	同 wpdmo72a，但設定 COM1 為 Modbus RTU slave 埠來連接一台 RS-232 觸控螢幕(Touch HMI) (中文 FAQ > 093)	
wpdmo72c	新的 WP-8xx7 備援(冗餘)系統 (Ethernet I/O): 搭配 I-8KE8-MTCP I/O 模組. (不連接 Touch HMI)	
wpdmo74a	求實數平均值 (中文 FAQ > 099)	
wpdmo74b	求整數平均值 (中文 FAQ > 099)	
wpdmo75	在插槽 0 使用 I-8088W(8-ch, PWM output)	slot 0: I-8088W
wpdmo75b	經由 WP-8xx7 的 COM2:RS485 連接 I-87088W (I-7088) (addr=1,baud=115200)	I-87088W (I-7088)
wpdmo77a	使用 eth_udp，eth_send()，eth_recv() 來傳送 / 接收 UDP bytes	
wpdmo77b	使用 eth_tcp，eth_send()，eth_recv() 來傳送 / 接收 TCP bytes	
wpdmo78	XP-8xx7-CE6 COM3 Mbus Master---M-7011 (ID=1, baud=9600) 來取得 AI,DI (中文 FAQ > 118)	M-7011
wpdmo78	WP-8xx7 COM2 Mbus Master---M-7011 (ID=1, baud=9600) 來取得 AI,DI (中文 FAQ-118)	M-7011
wpdmo79a	AP1 (中文 FAQ-119): Mbus RTU Master (中央工作站)	
wpdmo79b	AP1 (中文 FAQ-119): Mbus RTU Slave (區域工作站 1), 須設定 ID (Slave Number) 為 1	
wpdmo79c	AP1 (中文 FAQ-119): Mbus RTU Slave (區域工作站 2), 須設定 ID (Slave Number) 為 2	
wpdmo80a	AP2 (中文 FAQ - 119): Mbus TCP Master (中央工作站)	
wpdmo80b	AP2 (中文 FAQ - 119) (區域工作站 1), 須設定 ID 為 1, LAN1=192.168.1.178, LAN2=192.168.1.179	
wpdmo80c	AP2 (中文 FAQ - 119) (區域工作站 2), 須設定 ID 為 1, LAN1=192.168.1.180, LAN2=192.168.1.181	
wpdmo81	WP-8xx7+slot 1: I-8017HW (single-End) 來取得 動態平均值 (中文 FAQ-120)	slot 1: I-8017HW

VB.NET 2008 範例程式： 可與 ISaGRAF 程式 同時在同一台 WP-8xx7 內執行

WP-8xx7 光碟 : \napdos\isagraf\wp-8xx7\vb.net_2008_demo\
http://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/vb.net_2008_demo/

範例名稱	說明 (VB.NET 2008 搭配 WP-8xx7)	使用的 I/O 板
wp_vb01	數位 I/O 範例 (對應的專案範例檔名: "wp_vb01.pia")	I-87055W (slot 0)
wp_vb02	類比 I/O 範例 (對應的專案範例檔名: "wp_vb02.pia")	I-87024W (slot 1) I-8017HW (slot 2)
wp_vb03	讀/寫 ISaGRAF internal integers, timers 及 real 變數 (對應的專案範例檔名: "wp_vb03.pia")	

WP-8xx7 Web HMI 範例程式：

Web HMI 範例程式資料夾：

WinPAC-8xx7 光碟: \napdos\isagraf\wp-8xx7\wp_webhmi_demo\

ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/wp_webhmi_demo/

對應的專案範例資料夾：

WinPAC-8xx7 光碟: \napdos\isagraf\wp-8xx7\demo\

<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

程式名稱	說明 (WP-8xx7 Web HMI)	搭配使用的 I/O 模組
sample	Web HMI 範例	無需搭配 I/O 模組
example1	第 4 章的範例	插槽 0: I-87055W
wphmi_01	顯示控制器的日期與時間	無需搭配 I/O 模組
wphmi_02	DI 與 DO 範例	插槽 0: I-87055W
wphmi_03	讀/寫 Long, float 與 Timer 值	無需搭配 I/O 模組
wphmi_04	讀/寫控制器的字串(String)	無需搭配 I/O 模組
wphmi_05	多頁範例: 頁面選單在左方	插槽 0: I-87055W
wphmi_05a	多頁範例: 頁面選單在上方	插槽 0: I-87055W
wphmi_06	AI/AO 範例, 於 ISaGRAF 的轉換	插槽 2: I-87024W 插槽 3: I-8017HW
wphmi_07	AI/AO 範例, 於 PC 的轉換	插槽 2: I-87024W 插槽 3: I-8017HW
wphmi_08	下載控制器裡的檔案到 PC	插槽 0: I-87055W
wphmi_09	於 PC 彈出警報視窗	插槽 0: I-87055W
wphmi_11	趨勢圖	插槽 2: I-87024W 插槽 3: I-8017HW
wphmi_12	記錄 I-8017HW 每隔 50 微秒 1 ~ 8 通道的電壓, 並以微軟 Excel 軟體繪出趨勢圖	插槽 3: I-8017HW 插槽 2: I-8024W
wphmi_13	記錄 I-8017HW 每隔 10 微秒 1 ~ 4 通道的電壓, 並以微軟 Excel 軟體繪出趨勢圖	插槽 3: I-8017HW 插槽 2: I-8024W

11.3: 一些範例程式的詳細說明

11.3.0 : Demo_01A & Demo_03: 在某個時間點做某件事

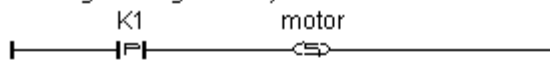
Demo_01A: 於事件發生後某個時間點(例如 3 秒後) 去做某件事.

程式存放位置: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ "demo_01a.pia"

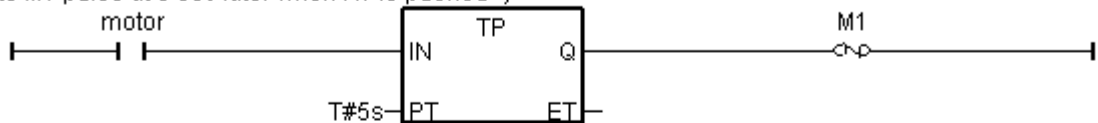
變數宣告 :

Name	Type	Attribute	Description
K1	Boolean	Input	按壓 K1 來啟動 motor (K1 為 I-8xx7 面板上的第 1 個按鈕)
Motor	Boolean	Output	True 表示 motor 已經在轉了, False 則為 停止轉動
Gate	Boolean	Output	True 表示 閘口 打開了, False 則為 關閉
M1	Boolean	Internal	存放 K1 被按壓後 5 秒產生的脈波訊號
M2	Boolean	Internal	存放 K1 被按壓後 15 秒產生的脈波訊號
M3	Boolean	Internal	存放 K1 被按壓後 18 秒產生的脈波訊號
T1	Timer	Internal	已經 經過的時間

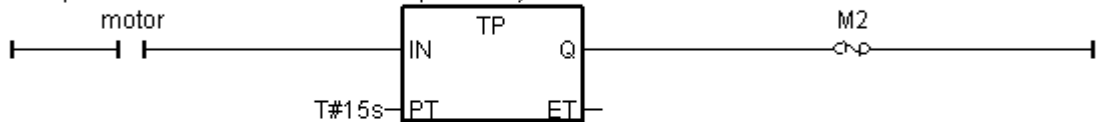
(* Push K1 to starting running motor *)



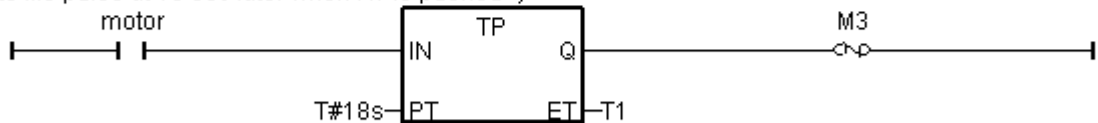
(* To generate M1 pulse at 5 sec later when K1 is pushed *)



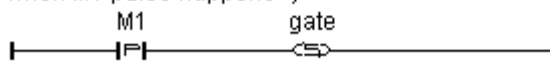
(* To generate M2 pulse at 15 sec later when K1 is pushed *)



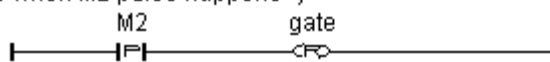
(* To generate M3 pulse at 18 sec later when K1 is pushed *)



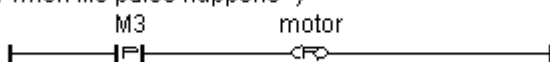
(* Open gate when M1 pulse happens *)



(* Close gate when M2 pulse happens *)



(* Stop motor when M3 pulse happens *)



Demo_03: 在星期幾的某個時段輸出

程式存放位置: I-8000 CD-ROM: \napdos\isagraf\8000\demo\ "demo_03.pia"

變數宣告 :

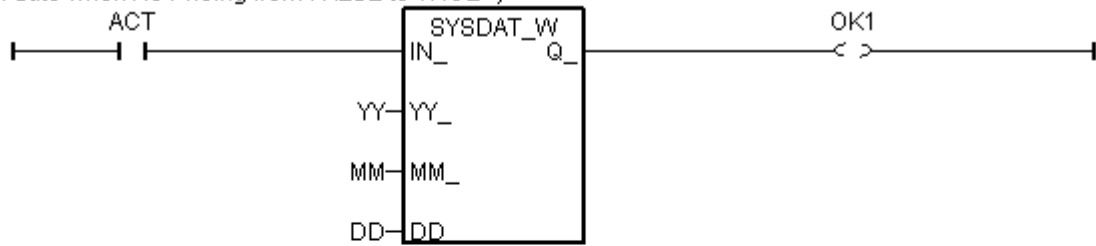
Name	Type	Attribute	Description
Year	Integer	Internal	年, 2001 ~
Month	Integer	Internal	月, 1 ~ 12
Day	Integer	Internal	日, 1 ~ 31
Wday	Integer	Internal	星期幾, 1:Monday ~ 6:Saturday, 7:Sunday
Hour	Integer	Internal	時, 0 ~ 23
Minute	Integer	Internal	分, 0 ~ 59
Second	Integer	Internal	秒, 0 ~ 59
YY	Integer	Internal	要設定的 年
MM	Integer	Internal	要設定的 月
DD	Integer	Internal	要設定的 日
HH	Integer	Internal	要設定的 時
Mn	Integer	Internal	要設定的 分
Sec	Integer	Internal	要設定的 秒
Act	Boolean	Internal	觸發它後 可設定新 日期
Act1	Boolean	Internal	觸發它後 可設定新 時間
OK1	Boolean	Internal	"SYSDAT_W" 的傳回值
OK2	Boolean	Internal	"SYSTEM_W" 的傳回值
L1 ~ L3	Boolean	Internal	模擬 輸出 1 ~ 3
Time_val	Integer	Internal	單位為秒, = 3600 x 時 + 60 x 分 + 秒, 每日為 0 ~ 86399

動作說明:

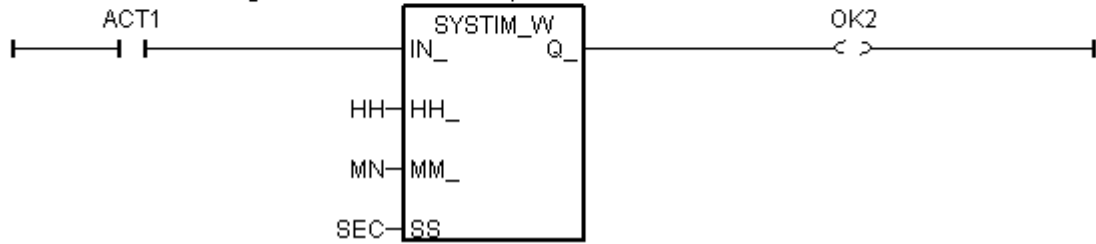
1. 星期 1 ~ 星期 6, L1 ~ L3, 09:00:00 ~ 18:00:00 輸出為 ON
2. 星期天, L1, 13:00:00 ~ 20:00:00 輸出 ON
3. 其它時間, L1 ~ L3 輸出為 OFF

階梯圖程式 : get_time

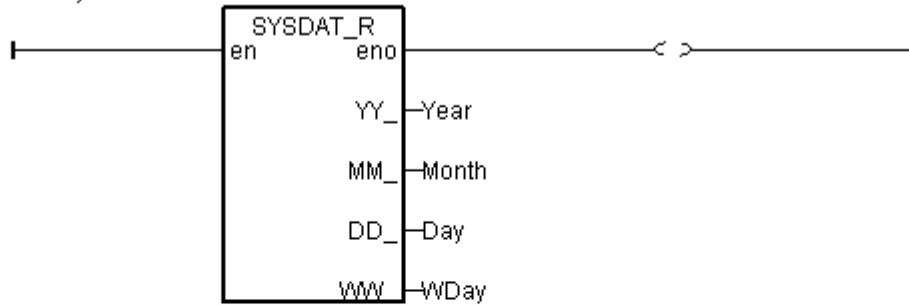
(* set system date when ACT rising from FALSE to TRUE *)



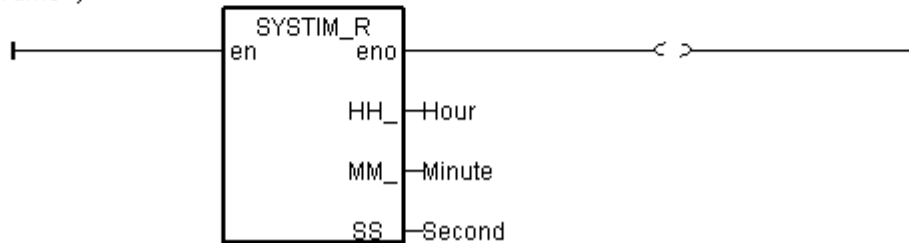
(* set system time when ACT1 rising from FALSE to TRUE *)



(* get system date *)



(* get system time *)



ST 程式 : control

```
time_val := 3600 * hour + 60 * minute + second ; (* calculate time in sec. *)
```

```
(* set as False at the beginning of this ST program*)
```

```
L1 := False ;
```

```
L2 := False ;
```

```
L3 := False ;
```

```
(* Monday ~ Saturday, L1 ~ L3, 09:00:00 ~ 18:00:00 ON *)
```

```
IF ( Wday >= 1 ) AND ( Wday <= 6 ) THEN
```

```
    IF ( time_val >= 32400 ) AND ( time_val <= 64800 ) THEN
```

```
        L1 := True ;
```

```
        L2 := True ;
```

```
        L3 := True ;
```

```
    END_IF ;
```

```
END_IF ;
```

```
(* Sunday, L1, 13:00:00 ~ 20:00:00 ON *)
```

```
IF ( Wday = 7 ) THEN
```

```
    IF ( time_val >= 46800 ) AND ( time_val <= 72000 ) THEN
```

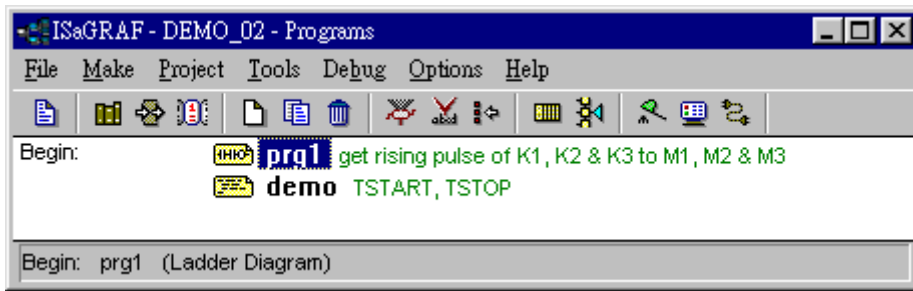
```
        L1 := True ;
```

```
    END_IF ;
```

```
END_IF ;
```

11.3.1 Demo_02 : 啟動, 暫停及重置 Timer

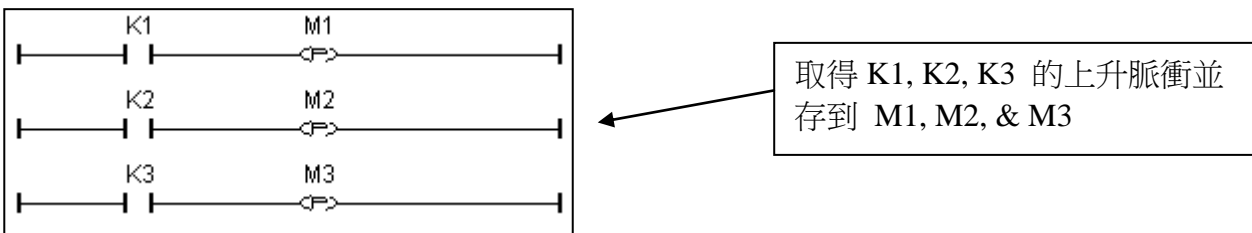
專案程式架構:



變數 :

Name	Type	Attribute	Description
M1	Boolean	Internal	表示 K1 剛被按下與否
M2	Boolean	Internal	表示 K2 剛被按下與否
M3	Boolean	Internal	表示 K3 剛被按下與否
K1	Boolean	Input	1 號按鈕
K2	Boolean	Input	2 號按鈕
K3	Boolean	Input	3 號按鈕
L1	Boolean	Output	1 號輸出
L2	Boolean	Output	2 號輸出
L3	Boolean	Output	3 號輸出
T1	Timer	Internal	操作的 timer, 初值設為 T#0s

LD 程式 “prg1”:



ST 程式 “demo” :

```
(* Start timer *)
IF M1 THEN
  TSTART(T1);
END_IF;

(* Stop timer *)
IF M2 THEN
  TSTOP(T1);
END_IF;

(* Reset timer *)
IF M3 THEN
  T1 := T#0s;
END_IF;

(* Output L1 ~ L3 *)
L1 := ( T1 > T#2s ) AND ( T1 < T#15s );
L2 := L1;
L3 := L1;
```

“TSTART” 會啟動 “T1” timer

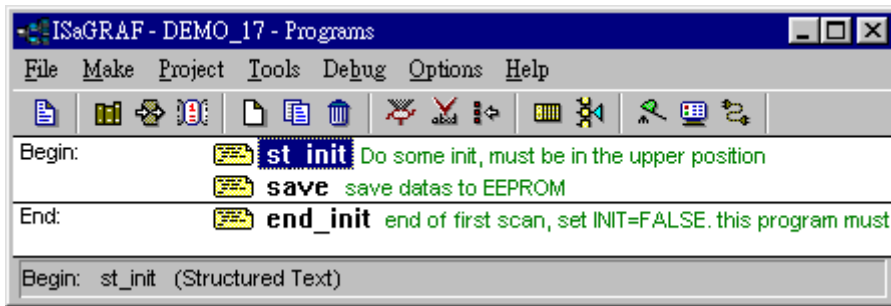
“TSOP” 會暫停 “T1” timer

重置 “T1” timer 為 0 秒

“L1” 在 “T1” Timer 介於 2 ~ 15 秒時為 TRUE

11.3.2 Demo_17 : 讀/寫 EEPROM

專案程式架構:



變數: (關於 變數陣列 的使用, 請參考第 2.6 節)

Name	Type	Attribute	Description
V[0..7]	Integer	Internal	更改 V[0..7]的值就會存新值到 EEPROM 宣告為變數陣列, Dim 設為 8
Old_V[0..7]	Integer	Internal	V[0..7] 的前一個舊值 宣告為變數陣列, Dim 設為 8
TEMP	Boolean	Internal	暫時使用的變數
ii	Integer	Internal	給 For 迴圈使用的 index
INIT	Boolean	Internal	當控制器剛開機時,宣告初值為 TRUE

ST 程式 “st_init”:

```

if INIT = TRUE then (* First scan
cycle *)

    (* Read 8 integers from EEPROM *)
    (* save them to Old_V[0..7], V[0..7] *)
    for ii := 0 to 7 do
        V[ii] := eep_n_r(ii+1) ;
        Old_V[ii] := V[ii] ;
    End_for ;

    (* remove protection of EEPROM *)
    TEMP := eep_en();

end_if ;
    
```

在第 1 個周期(剛開機時), 從 EEPROM 內讀取第 1 到 8 號位置內的長整數值。

開啟 EEPROM, 之後方可寫入資料

ST 程式 “save” :

```
(* save V[0..7] to EEPROM if it is modified *)
(* You will find write to EEPROM take lots of time, about 23ms for each eep_n_w *)
for ii := 0 to 7 do
    IF V[ii] <> Old_V[ii] THEN
        TEMP := eep_n_w(ii+1 , V[ii]) ;
        Old_V[ii] := V[ii] ;
    END_IF ;
End_for ;
```

當值有改變時，存入新值到 EEPROM 內

之後更新舊值變數=新值

ST 程式 “end_init” :

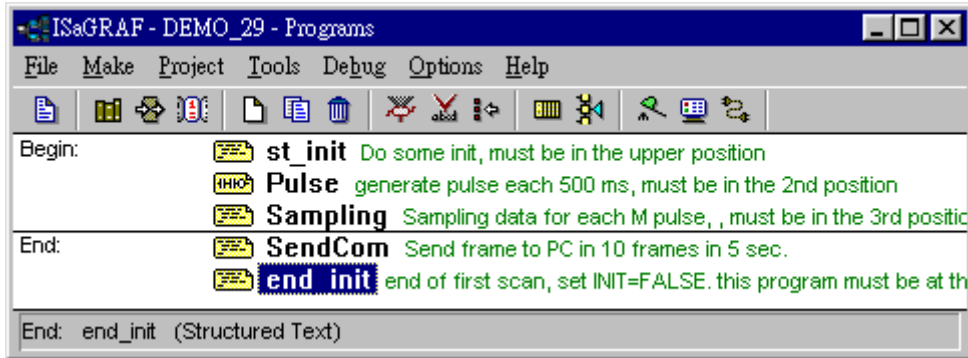
```
INIT := FALSE ; (* end of first PLC scan *)
```

在 End 區將 “INIT” 設為 FALSE, 如此 “INIT” 只有在第 1 個周期時為 TRUE

11.3.3 Demo_29: 每 75 秒記錄 1200 筆 Short Int 並經由 Com3 傳送到 PC

這個範例每 500ms 會儲存 I-87017 的 8 個類比輸入點到 short-integer 陣列. 然後當資料量達到 1200 筆時, 會分成 10 個通訊封包, 每個封包內含 120 筆, 透過 Com3 傳到 PC.

專案程式架構:



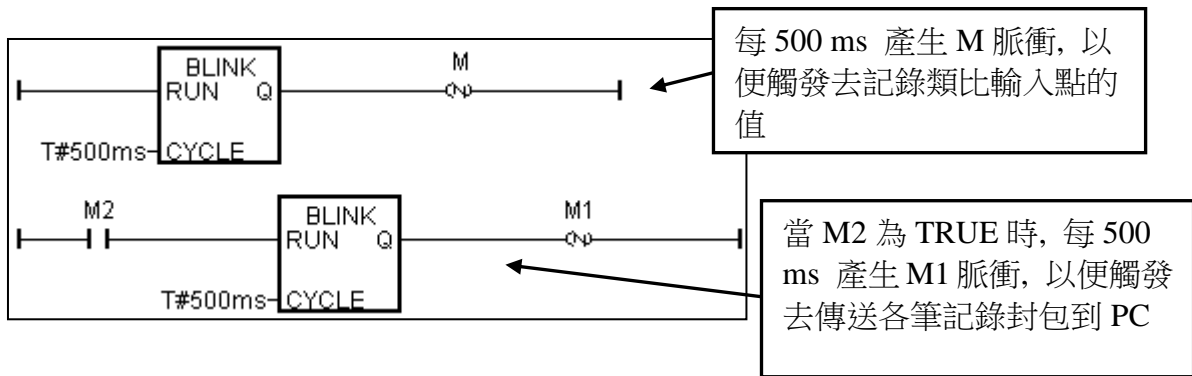
變數 :

Name	Type	Attribute	Description
M	Boolean	Internal	用來儲存資料的觸發脈衝
M1	Boolean	Internal	用來傳送資料的觸發脈衝
M2	Boolean	Internal	用來產生 M1
INIT	Boolean	Internal	只有第 1 個周期時為 TRUE, 初值為 TRUE
TMP	Boolean	Internal	暫時使用的變數
A1	Integer	Input	I-87017 的第 1 點
A2	Integer	Input	I-87017 的第 2 點
A3	Integer	Input	I-87017 的第 3 點
A4	Integer	Input	I-87017 的第 4 點
A5	Integer	Input	I-87017 的第 5 點
A6	Integer	Input	I-87017 的第 6 點
A7	Integer	Input	I-87017 的第 7 點
A8	Integer	Input	I-87017 的第 8 點
count	Integer	Internal	目前正在處理的資料編號 (1~1200), 初值為 1
position	Integer	Internal	目前正處理的 short integer 陣列內的位置編號, 1 ~ 256
No	Integer	Internal	目前正處理的 short integer 陣列編號
Frame_No	Integer	Internal	封包編號 0 ~ 10, 0 表示不傳送
TMP_VAL	Integer	Internal	暫時使用的變數

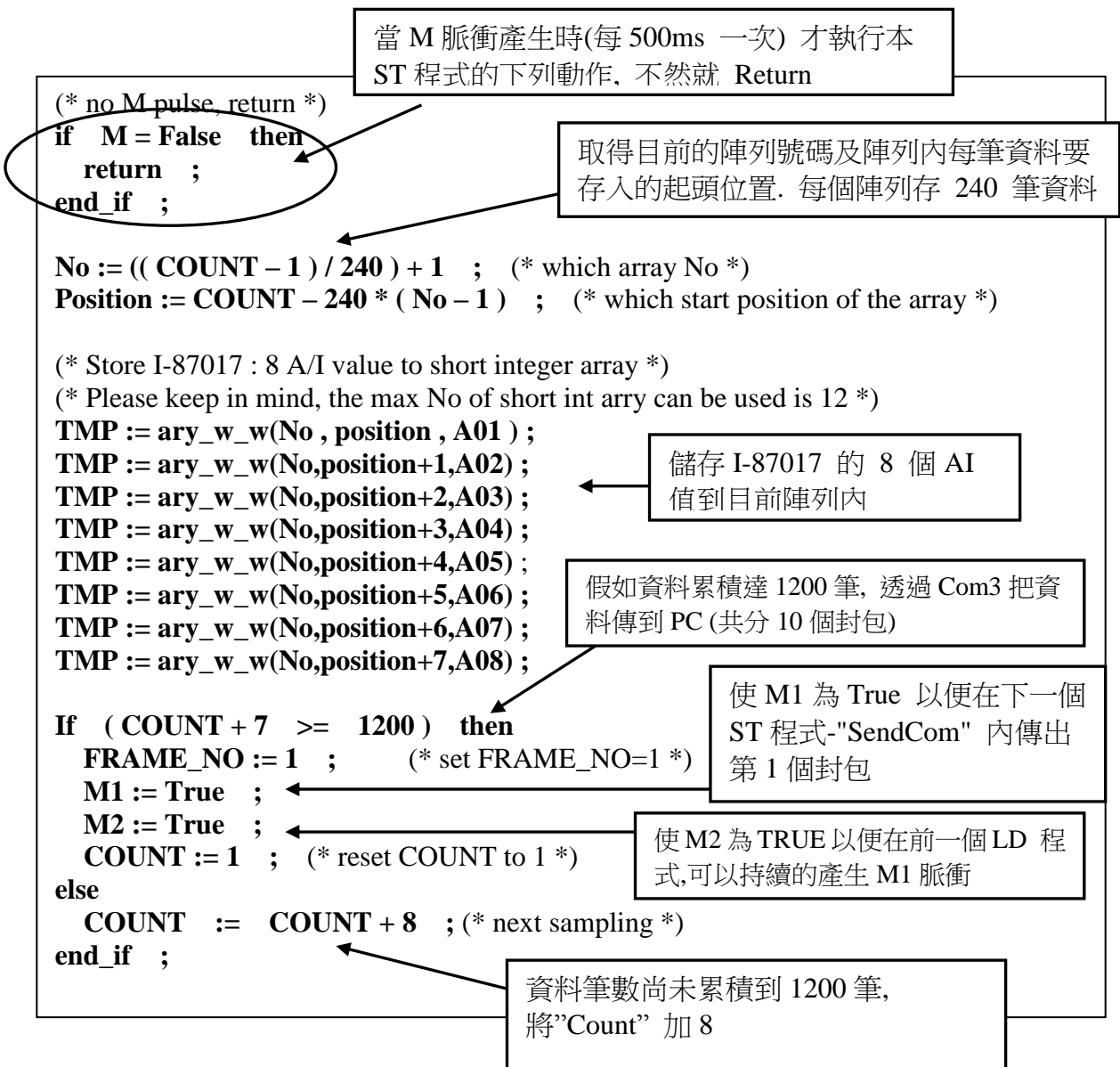
ST 程式 “st_init”:

<pre> if INIT = TRUE then (* First scan cycle *) COUNT := 1 ; (* init count to 1 *) FRAME_NO := 0 ; (* init to 0 *) M2 := False ; (* init M2 to false *) TMP := comopen(3, 9600, 8, 0, 1) ; end_if ; </pre>	<p>初始動作</p> <p>開啟 Com3 為速率 9600, 字元大小為 8, 沒有同位檢查& 停止位元為 1</p>
--	---

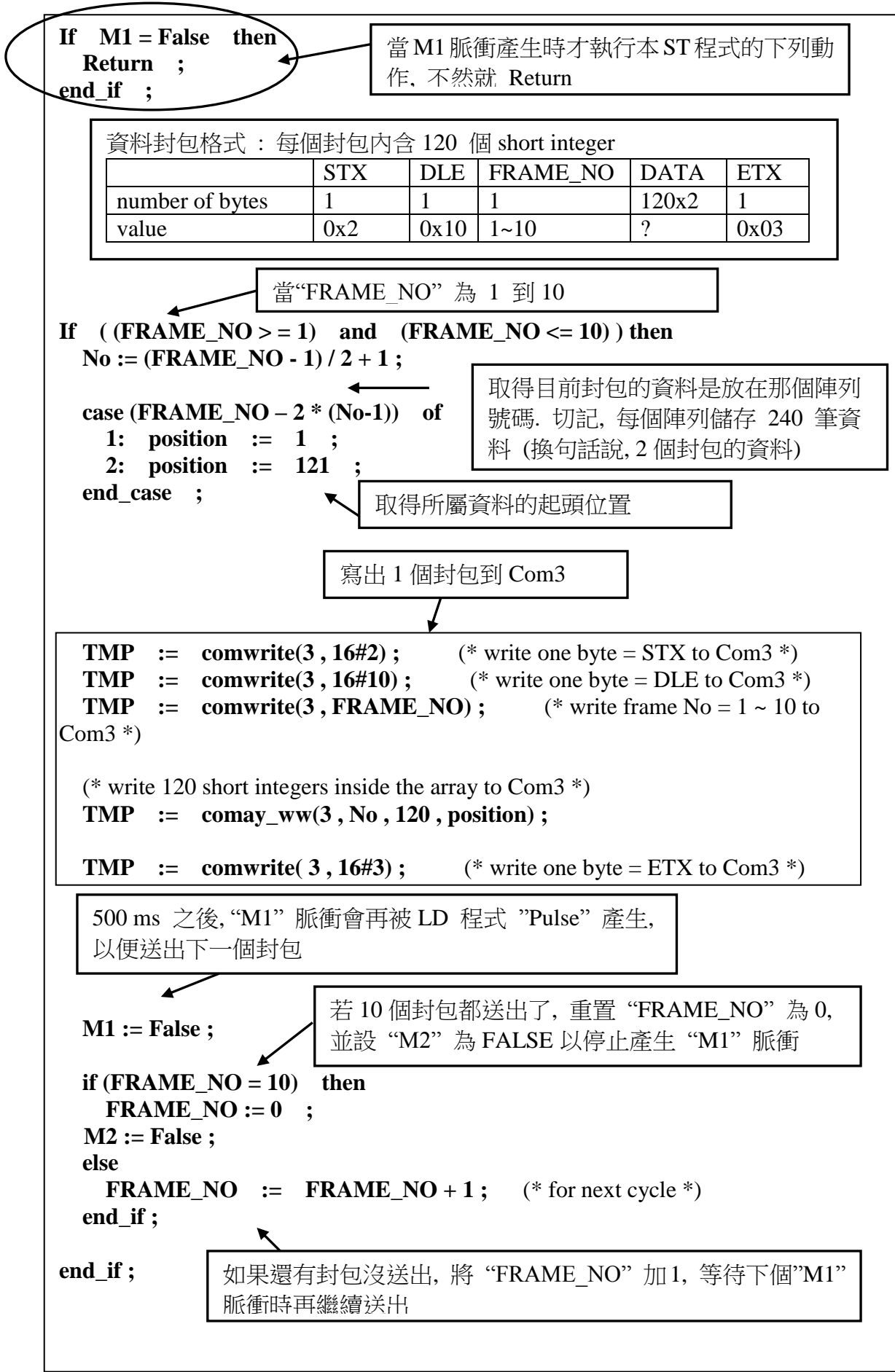
LD 程式 “Pulse”:



ST 程式 “Sampling”：



ST 程式 “SendCom” :



ST 程式 “end_init” :

```
INIT := FALSE ; (* end of first PLC scan *)
```

在 End 區將 “INIT” 設為 FALSE, 如此 “INIT” 只有在第 1 個周期時為 TRUE

如何測 ?

將 1 個 I-87017 插入第 0 槽.

下載 Demo_29

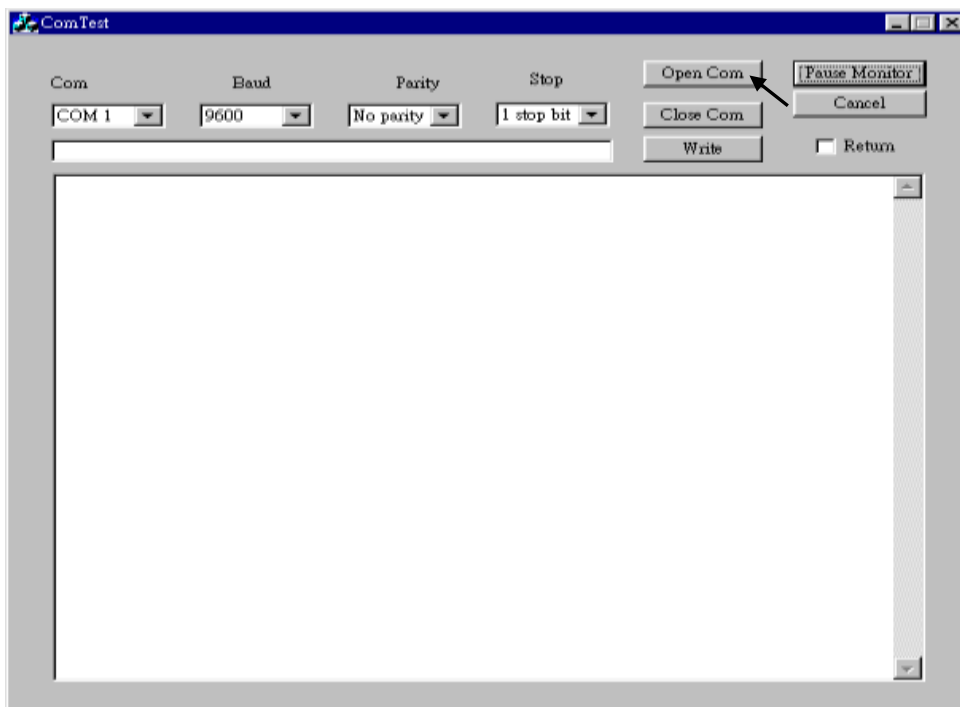
準備一條 RS232 通訊線連接 Com3 到 pc 的 Com1

有個工具程式名叫“ComTest.exe” 放在 ICP DAS 的光碟內. 覆製它到你的 PC 內.

“\Napdos\ISaGRAF\some_utility\Comtest.exe” 或由網路上下載

ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/some_utility/

執行 “ComTest”, 選取參數為 “COM1”, “9600”, “No parity”, “1 stop bit”, 然後點選 “Open Com”.



應可看到每隔 75 秒有 10 個封包傳出

11.3.4 Demo_33：從 Com3 讀/寫 User 自行定義的通訊格式

本範例程式可讓 I-8xx7, iP-8xx7, WP-8xx7, VP-25W7 接收及回答由 PC 經過 RS232 通訊線傳來的命令. 通訊格式定義如下

命令自元大小寫皆可, 比如 M1 與 m1 都可接受

通訊格式:

PC 問.

M1<CR> : 切換到 Mode 1
M2<CR> : 切換到 Mode 2
M3<CR> : 切換到 Mode 3
Txxxx<CR> : 改變周期時間到 xxxx ms
例如. T250<CR> 會改變成 250 ms

控制器回.

OK<CR>

PC 問.

M?<CR> : 詢問目前的 Mode

控制器回.

Mx<CR> : 例如 回 M1 表示 Mode 1

PC 問.

T?<CR> : 詢問目前的周期時間

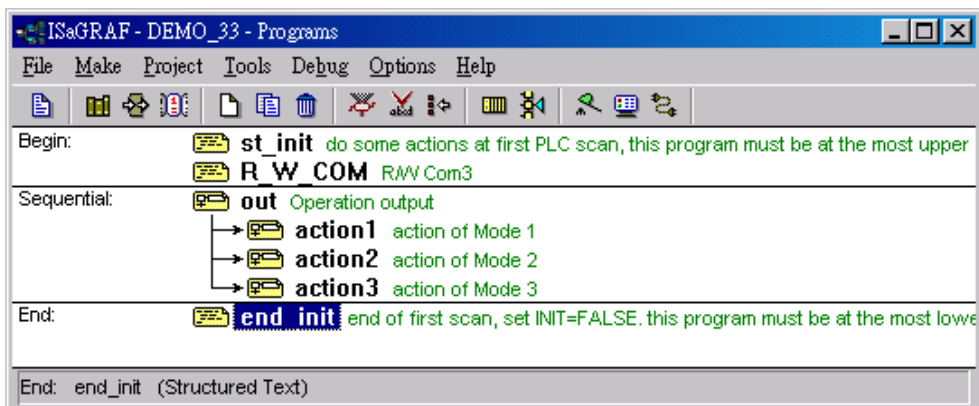
控制器回.

Txxxx<CR> : 例如 回 T1500 表示周期時間為 1500 ms

Timeout:

一個正確的命令需在 5 秒內送達, 不然視為無效

專案架構:



變數：

Name	Type	Attribute	Description
L1	Boolean	Output	1 號輸出
L2	Boolean	Output	2 號輸出
L3	Boolean	Output	3 號輸出
INIT	Boolean	Internal	只有第 1 個周期時為 TRUE, 初值為 TRUE
TMP	Boolean	Internal	暫時使用的變數
Mode	Integer	Internal	輸出模式, 1 ~ 3
Step	Integer	Internal	目前步驟
NUM	Integer	Internal	已收到的 byte 數量
Num_com3	Integer	Internal	Comary_R 函數的回傳數量
byt	Integer	Internal	目前處理的那個 byte
index	Integer	Internal	byte array 的編號
CMD	Integer	Internal	命令型態, M, m, T, 或 t
TMP_val	Integer	Internal	暫時使用的變數
ii	Integer	Internal	暫時使用的變數
T1	Timer	Internal	周期時間, 50 ~ 9999 ms
tout	Timer	Internal	用來測量是否 timeout 的 Timer, 當第 1 個 byte 收到時就開始計時

ST 程式 “st_init”：

```

if INIT = TRUE then

    (* Init *)
    Mode := 1 ;
    STEP := 0 ;
    T1 := T#500ms ;
    NUM := 0 ;
    tout := T#0s ;

    (* Open Com3 as baud=9600, char. size=8, no parity & stop bit=1 *)
    TMP := comopen( 3 , 9600 , 8 , 0 , 1 ) ;

end_if ;

```

初始動作

ST 程式 “R_W_COM”:

```

(* STEP: *)
(* 0: no valid data coming *)
(* 1: first valid byte received *)
(* 2: receive '?' command *)
(* 10: check if other byte is '0' ~ '9' *)
(* 21: receive complete command *)

num_com3 := 0 ; (* reset to 0 *)
if ComReady(3) then
    num_com3 := Comary_R(3,1);
end_if ;

(* if data coming, process it *)
index := 1 ;
while num_com3 > 0 do
    num_com3 := num_com3 - 1 ;
    byt := array_r(1, index);
    index := index + 1 ;

    case STEP of
    0:
        case byt of (* check 1st byte *)
        16#4D, 16#6D, 16#54, 16#74 : (* 1st byte is valid, M, m, T, or t *)
            STEP := 1 ; (* for next STEP *)
            NUM := 1 ; (* plus valid received byte number by 1 *)
            TSTART(tout) ; (* start ticking tout *)
            CMD := byt ; (* record command type *)
            TMP := Array_w(2, NUM, byt) ; (* save 1st valid byte to byte array 2*)
        end_case ;
    1:
        case byt of (* check 2nd byte *)
        16#3F : (* 2nd byte is '?' *)
            STEP := 2 ; (* for next STEP *)
            NUM := 2 ; (* plus valid received byte number by 1 *)
            TMP := Array_w(2, NUM, byt) ; (* save 2nd valid byte to byte array 2*)
        else
            if (CMD = 16#4D or CMD = 16#6D) (* cmd is M, m *)
                and (byt >= 16#31) and (byt <= 16#33) then (* '1' ~ '3' *)
                    STEP := 10 ; (* for next step *)
                    NUM := 2 ; (* plus valid received byte number by 1 *)
                    TMP := Array_w(2, NUM, byt) ; (* save 2nd valid byte to byte array 2*)
                elsif (CMD = 16#54 or CMD = 16#74) (* cmd is T, t *)
                    and (byt >= 16#30) and (byt <= 16#39) then (* '0' ~ '9' *)
                        STEP := 10 ; (* for next step *)
                        NUM := 2 ; (* plus valid received byte number by 1 *)
                        TMP := Array_w(2, NUM, byt) ; (* save 2nd valid byte to byte array 2*)
                    else
                        STEP := 0 ; (* not valid data, reset STEP to 0 *)
                        TSTOP(tout) ; (* stop ticking "tout" *)
                    end_if ;
                end_case ;
            end_if ;
        end_case ;
    end_case ;
end_while ;

```

測試是否有資料從 Com3 傳進來

若有,把傳進來的全部 byte 讀到 1 號陣列

處理每個進來的 byte

從陣列內取得目前要處理的 byte

步驟 0: 檢查頭一個 byte 對不對

步驟 1: 檢查第 2 個 byte 對不對

```

Tout := T#0s ; (* reset "tout" *)
NUM := 0 ; (* reset NUM *)
end_if ;
end_case ;

```

步驟 2: 收到第 2 個 byte = '?' 後, 檢查第 3 個 byte 是不是 <CR>

2:

```

if byt = 16#0D then (* check 3rd byte is <CR> or not *)
STEP := 21 ; (* complete command is received *)
(* send answer to Com3 *)

```

收到正確的“M?” 命令, 回答 “Mx”, x = ‘1’ ~ ‘3’

```

case CMD of

```

```

16#4D, 16#6D : (* M or m *)
TMP := ComWrite(3, 16#4D) ; (* M *)

```

```

TMP := ComWrite(3, Mode + 16#30) ; (* Mode *)

```

```

TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

收到正確的“T?” 命令, 回答 “Txxxx”, x = ‘0’ ~ ‘9’

```

16#54, 16#74 : (* T or t *)

```

```

TMP := ComWrite(3, 16#54) ; (* T *)

```

```

TMP := ComStr_w(3, MSG(ANA(T1))) ; (* Timer value *)

```

```

TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

```

end_case ;

```

else

```

STEP := 0 ; (* not valid data, reset STEP to 0 *)

```

```

TSTOP(tout) ; (* stop ticking "tout" *)

```

```

tout := T#0s ; (* reset "tout" *)

```

```

NUM := 0 ; (* reset NUM *)

```

```

end_if ;

```

步驟 10: 當收到的是“Mx” 或 “Txxxx” 命令, 檢查第 3 個及其它的

10:

```

if (byt = 16#0D) then (* received <CR> *)
STEP := 21 ; (* complete command is received *)

```

```

case CMD of

```

```

16#4D, 16#6D : (* M or m *)

```

```

Mode := Array_r(2, 2) - 16#30 ; (* Change Mode *)

```

```

(* send answer to Com3 *)

```

```

TMP := ComStr_w(3, 'OK') ;

```

```

TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

收到正確的“Mx” 命令, 回答“OK”, 並切換 Mode

```

16#54, 16#74 : (* T or t *)

```

```

(* get Period *)

```

```

TMP_val := 0 ;

```

```

for ii := 1 to NUM - 1 do

```

```

TMP_val := 10 * TMP_val + (Array_r(2, 1 + ii) - 16#30) ;

```

```

end_for ;

```

```

if (TMP_val >= 50) and (TMP_val < 10000) then (*T1 must be in 50~9999 ms*)

```

```

T1 := TMR(TMP_val) ; (* Change T1 *)

```

```

(* send answer to Com3 *)

```

```

TMP := ComStr_w(3, 'OK') ;

```

```

TMP := ComWrite(3, 16#0D) ; (* <CR> *)

```

```

end_if ;

```

```

end_case ;

```

收到‘0’ ~ ‘9’, 完整的命令還沒收完

```

elsif (byt >= 16#30) and (byt <= 16#39) then (* '0' ~ '9' *)

```

```

STEP := 10 ; (* for next step *)

```

```

NUM := NUM + 1 ; (* plus valid received byte number by 1 *)

```

```

TMP := Array_w( 2 , NUM , byt ) ; (* save other valid byte to byte array 2*)

if NUM > 5 then      (* command is too long, drop it *)
  STEP := 0 ;        (* reset STEP *)
  TSTOP(tout) ;     (* stop ticking "tout" *)
  Tout := T#0s ;    (* reset "tout" *)
  NUM := 0 ;        (* reset NUM *)
  EXIT ;           (* exit while loop *)
end_if ;

else
  STEP := 0 ;      (* not valid data, reset STEP to 0 *)
  TSTOP(tout) ;   (* stop ticking "tout" *)
  Tout := T#0s ;  (* reset "tout" *)
  NUM := 0 ;     (* reset NUM *)

end_if ;

end_case ;

end_while ;

(* Check timeout *)
if tout > T#5s then (* if timeout *)
  STEP := 0 ;      (* reset STEP *)
  TSTOP(tout) ;   (* stop ticking "tout" *)
  Tout := T#0s ;  (* reset "tout" *)
  NUM := 0 ;     (* reset NUM *)
end_if ;

(* reset STEP to 0 *)
if STEP = 21 then
  TSTOP(tout) ;   (* stop ticking "tout" *)
  Tout := T#0s ;  (* reset "tout" *)
  NUM := 0 ;     (* reset NUM *)
  STEP := 0 ;
end_if ;

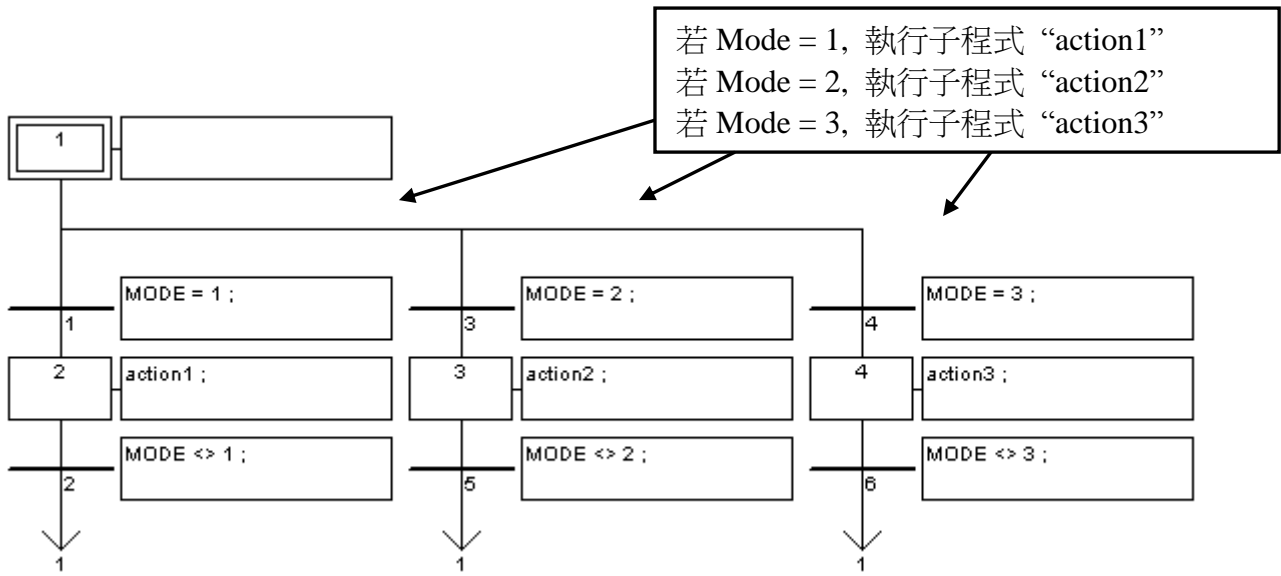
```

檢查是否 timeout, 一個正確的命令需在 5 秒內送達, 不然視為無效

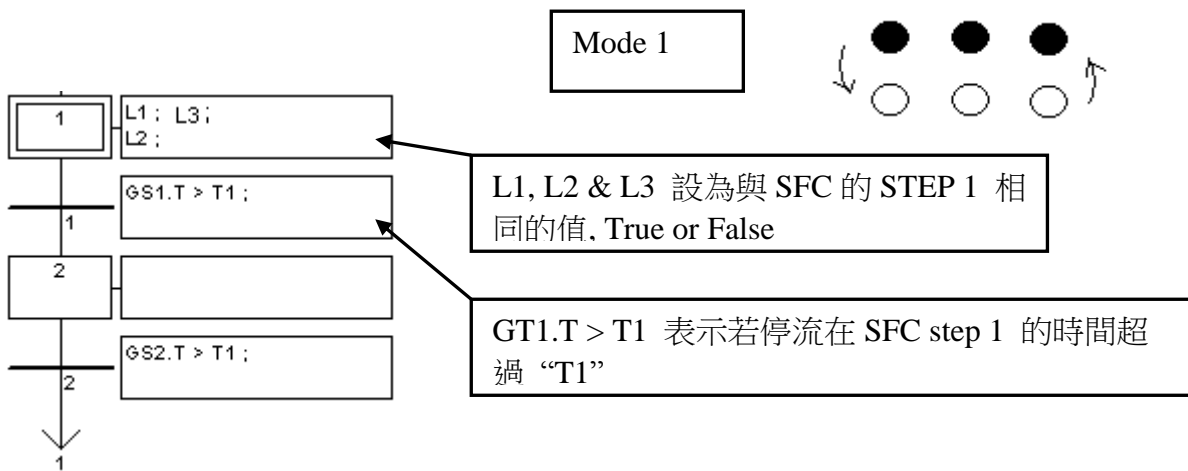
正確的命令已處理, 重置 STEP 為 0

SFC 程式 “Out”:

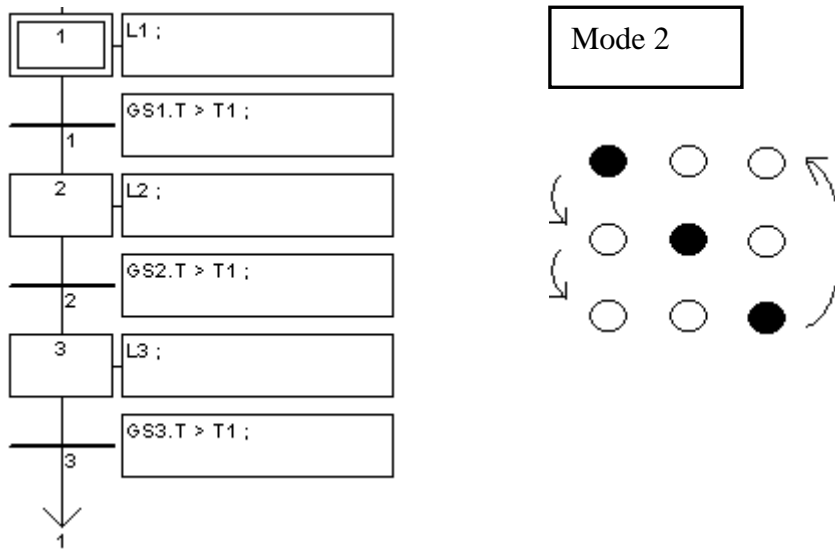
每個敘述以分號 “;” 結尾



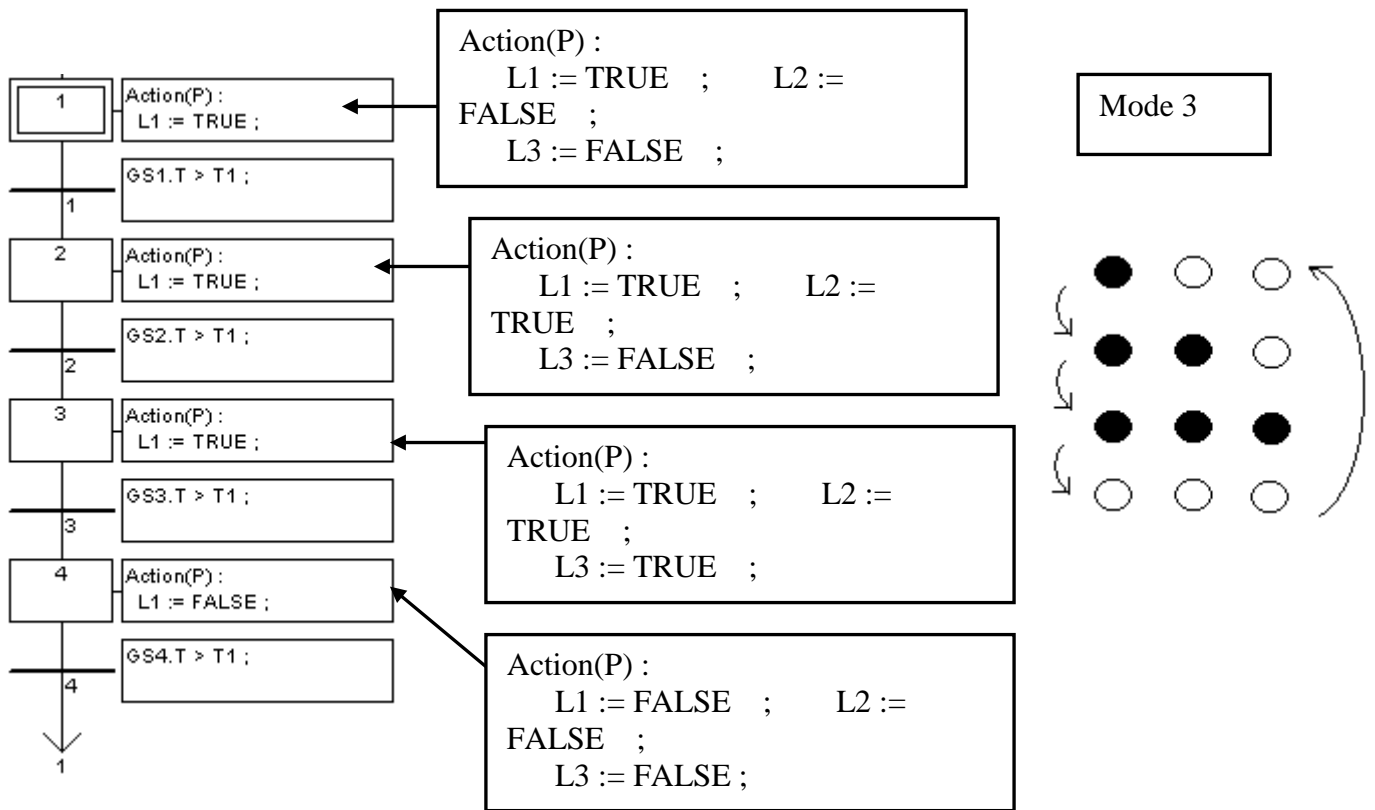
SFC 子程式 “action1”:



SFC 子程式 “action2”:



SFC 子程式 “action3”:



ST 程式 “end_init” :

```
INIT := FALSE ; (* end of first PLC scan *)
```

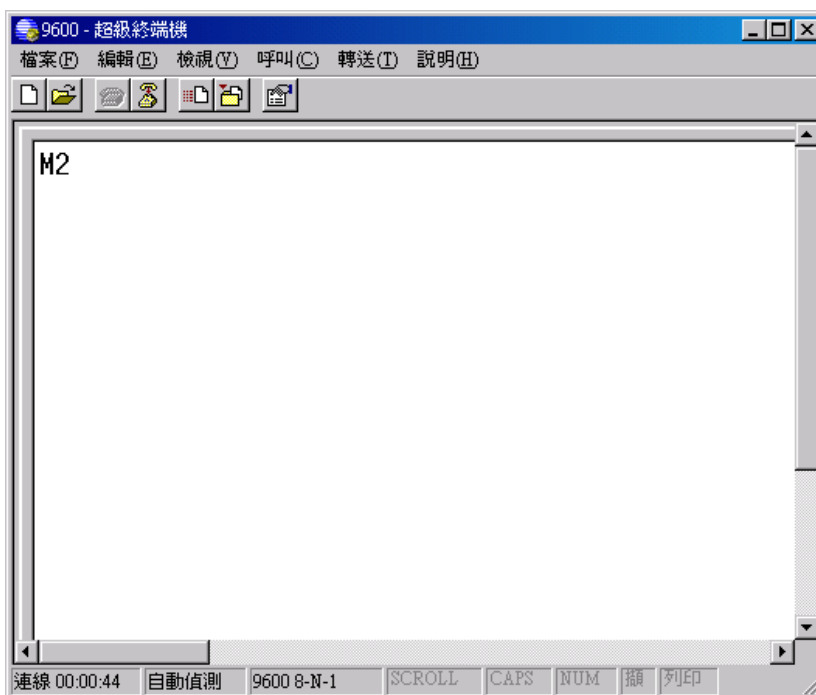
在 End 區將 “INIT” 設為 FALSE, 如此 “INIT” 只有在第 1 個周期時為 TRUE

如何測 ?

1. 下載 Demo_33 到控制器
2. 準備 1 條 RS232 線連接 Com3 到 PC 的 Com1
3. 可以開啟 1 個 “Hyper Terminal” 視窗為 Com1, 9600, N, 8, 1 並設成 “No flow control” , 之後可用鍵盤輸入

M2<CR> : 切換到 mode 2
T?<CR> : 詢問目前的輸出周期時間
T200<CR> : 改變周期時間為 200 ms
T1500<CR> : 改變周期時間為 1500 ms
M?<CR> : 詢問目前的 Mode

<CR> 代表 Enter 字元



11.3.5 Wdemo_24：當 Alarm 1 ~ 8 發生時，從 Com2 送出 Alarm 訊息

本程式可用於 Wincon-8xx7/8xx6 或 I-8xx7. 若是 Wincon，請把“PORT”變數的初值宣告為 2，若是 I-8xx7，請宣告為 3.

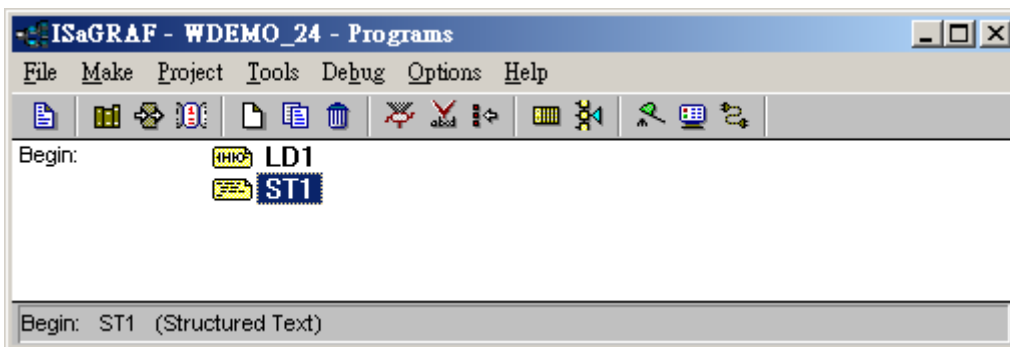
本範例放於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\“wdemo_24” 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/“wdemo_24”

本程式有使用 變數陣列，請參考 第 2.6 節關於 變數陣列 的詳細說明

變數定義：

Name	Type	Attribute	Description
INIT	Boolean	Internal	宣告初值為 TRUE
TMP	Boolean	Internal	暫時使用的變數
Tick1	Boolean	Internal	每秒用來產生 1 個 pulse True
IN[0..7]	Boolean	Input	輸入訊號，宣告為 變數陣列，Dim 欄位設為 8
OLD_IN[0..7]	Boolean	Internal	用來記錄 輸入訊號 的前一個 Scan 的舊值 宣告為 變數陣列，Dim 欄位設為 8
ii	Integer	Internal	給 ST 語言 For 迴圈使用的變數
Port	Integer	Internal	Alarm 訊息要送出去的 COM PORT 編號，宣告初值 為 2
CNT[0..7]	Integer	Internal	用來計算 輸入訊號 觸發後經過的秒數 宣告為 變數陣列，Dim 欄位設為 8
Msg1	Message	Internal	要送出的 Alarm 訊息，長度宣告為 128

專案架構：

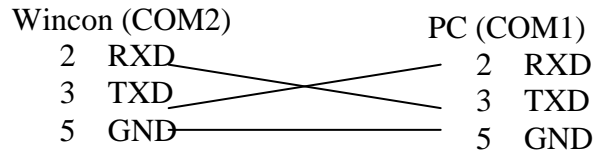


動作說明：

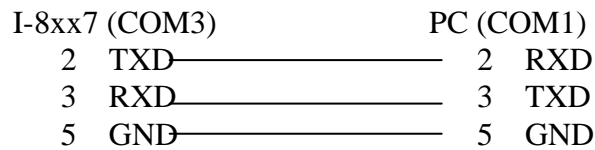
1. 當輸入訊號 IN[0..7] 由 False 上升到 True 時，會開始計時 3 秒，若這 3 秒內輸入訊號 一直維持在 True，則 3 秒時間一到，會發出第一個告警訊息到 COM2 串口
2. 當第一次告警訊息發出後，重新再計時 30 秒，若這 30 秒內輸入訊號還是 一直維持在 True，則 30 秒時間一到，會發出另一個告警訊息到 COM2 串口，之後一樣再重新再計時 30 秒，...
3. 只要 輸入訊號 IN[0..7] 下降為 False，就不再計時。

如何測試:

1. 請下載 wdemo_24 到 W-8xx7 + slot 1: I-8077 (若是 I-8xx7 請宣告 PORT 初值為 3)
2. 連接 RS232 通訊線如下



若是用 I-8xx7 則如下

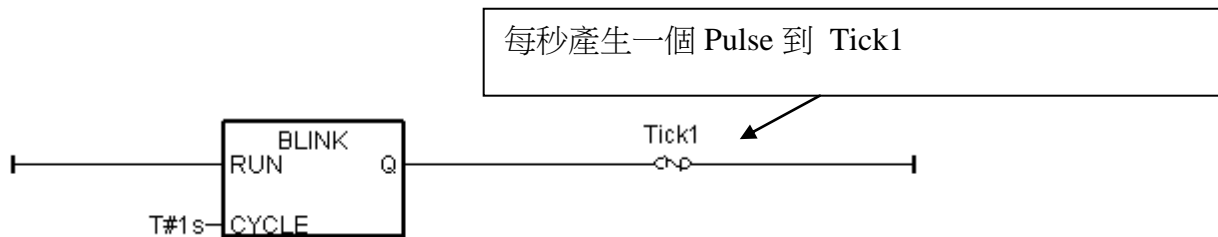


3. 開啟 PC 的“超級終端機”(Hyper terminal) 於 COM1, baud rate 設為 9600, 8 char. size, No parity, 1 stop bit 與 No flow control. 之後將 I-8077 的 Input 1 或 2 或 ... 從 FALSE 切換為 TRUE, 並維持在 True 等候約 3 秒. 若正常, Hyper terminal 應該會顯示對應的第 1 個告警訊息. 之後再等約 30 秒, 若正常, Hyper terminal 應該會顯示對應的第 2 個告警訊息. 若都不把 Input 切回 False, 將每過 30 秒就會送出一個告警訊息



程式內容:

LD1 程式:



ST1 程式:

If INIT then

INIT := FALSE ; (* 設 INIT 為 False, 因此這段 code 只有第一個 PLC Scan 會執行到 *)

TMP := COMOPEN(PORT, 9600, 8, 0, 1) ; (* 開啟 COM2 串口, 9600,8,N,1 *)

(* 設 CNT[0..7] 的初值為 -7 *)

for ii := 0 to 7 do

CNT[ii] := -7 ;

end_for ;

end_if ;

(* 測試 IN[0..7] 是否由 False 上升為 True *)

for ii := 0 to 7 do

(*測試 IN[0..7] 訊號是否上升為 True *)

if (IN[ii] = True) and (OLD_IN[ii] = False) then

(* 訊號有上升, 設對應的 CNT[] 值為 -3 *)

(* 如果 CNT[] 之值不是 -7, 表示 “輸入訊號被 觸發了” *)

(* 之後 CNT[] 之值將會每過 1 秒加 1, 除非訊號切換回 False *)

CNT[ii] := -3 ;

end_if ;

(* 如果輸入訊號切換回 False, 對應的 CNT[] 之值會設為 -7 來表示 “無訊號被觸發” *)

if IN[ii] = False then (* 若訊號切換回 False *)

(*將對應的 CNT[] 之值會設為 -7 來表示 “無訊號被觸發” *)

CNT[ii] := -7 ;

end_if ;

if Tick1 then (* Tick1 是在 LD1 程式內每秒會產生 1 次 Pulse True *)

(* 如果 CNT[] 之值 大於 -7, 表示訊號有被觸發 *)

```

if CNT[ii] > -7 then

    CNT[ii] := CNT[ii] + 1 ; (* 值加 1 , Tick1 為 True 表示時間又 1 秒過去了 *)

    (* ----- *)
    (*輸入訊號被觸發後 且 維持在觸發狀態 已經超過 3 秒, 從 COM2 送出第 1 次
    Alarm 訊息 *)
    if ( CNT[ii] = 0 ) then
        (*當 CNT[ ] 之值由 -3, -2, -1 到變為 0 時, 送出第 1 次 Alarm 訊息 *)
        CNT[ii] := 0 ; (* 將值回復為 0, 準備計時到 30 秒 *)
        (* 從 COM2 送出 Alarm 訊息 *)
        msg1 := 'Alarm ' + MSG( ii + 1 ) + '$0A$0D' ;
        TMP := comstr_w( PORT , msg1 ) ;
    end_if ;
    (* ----- *)

    (* ----- *)
    (*輸入訊號被觸發後 且 維持在觸發狀態 又經過超過了 30 秒, 從 COM2 送出第 2
    訊息 *)
    if ( CNT[ii] = 30 ) then
        (*當 CNT[ ] 之值由 1, 2, ... 到變為 30 時, 送出第 2 次訊息 *)
        CNT[ii] := 0 ; (*將值回復為 0, 準備計時到下一次 30 秒 *)
        (* 從 COM2 送出 Alarm 訊息 *)
        msg1 := 'Alarm ' + MSG( ii + 1 ) + ', 30 sec past ! $0A$0D' ;
        TMP := comstr_w( PORT , msg1 ) ;
    end_if ;
    (* ----- *)

    end_if ;      (* "if CNT[ ] > -7 then" *)

end_if ;      (* "if Tick1 then" *)

    (* 更新 OLD_IN[ ]之值 *)
    OLD_IN[ii] := IN[ii] ;

end_for ;

```

11.3.6: 範例 Whmi_12: 使用 Wincon-8xx7, 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 RAM Disk 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖

注意:

若 PAC 是 WP-8xx7, XP-8xx7-CE6 與 VP-25W7 / VP-23W7, 本節所述的功能也可以套用, 但差別是需使用各別 PAC 的 CD 的內容與 file (內容類似, 但不同於 Wincon CD). 另外 WP-8xx7, XP-8xx7-CE6 與 VP-25W7 / VP-23W7 等 PAC 並沒有 \CompactFlash\Temp\HTTP\WebHMI\, WP-8xx7 與 VP-25W7 / VP-23W7 是使用 \Micro_SD\Temp\HTTP\WebHMI\, 而 XP-8xx7-CE6 是使用 \System_Disk\Temp\HTTP\WebHMI\, 詳細內容請參考 以下網址:

WP-8xx7: “setup_web_hmi_demo.pdf” 放於 ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/wp_webhmi_demo 內.

VP-25W7 / VP-23W7: “setup_web_hmi_demo.pdf” 放於 <ftp://ftp.icpdas.com/pub/cd/vp-25w7-23w7/napdos/isagraf/vp-25w7-23w7/vp-webhmi-demo/> 內

XP-8xx7-CE6: “xpce6_setup_web_hmi_demo.pdf” 放於 <ftp://ftp.icpdas.com/pub/cd/xp-8xx7-ce6/napdos/isagraf/xp-8xx7-ce6/xpce6-webhmi-demo/> 內

本範例程式為 Whmi_12.pia 只能在 Wincon-8xx7/8xx6 的 驅動 3.36 版或更高的版本使用, 放於

新的驅動: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>
範例程式 放於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ 內 或 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

W-8xx7 提供 3 種記錄資料的方式. 一種是在 RAM Disk 內讀/寫 File, 本範例就是如此, 適合最小取樣時間可到 40ms 的應用 (最快 25Hz). 存放於 RAM Disk 內的 File, 關機後 File 會消失.

另一種為把資料記錄在 “\CompactFlash\” 路徑內. 對 CompactFlash 卡做 讀/寫 會比 RAM Disk 內讀 / 寫 File 慢很多. 它的優點是 關機後 File 不會消失, 缺點是 讀 / 寫 很慢.

第 3 種是最快的一種方式, 最小取樣時間可到 10ms 的應用, 即最快 100Hz. 請參考第 11.3.10 的範例說明.

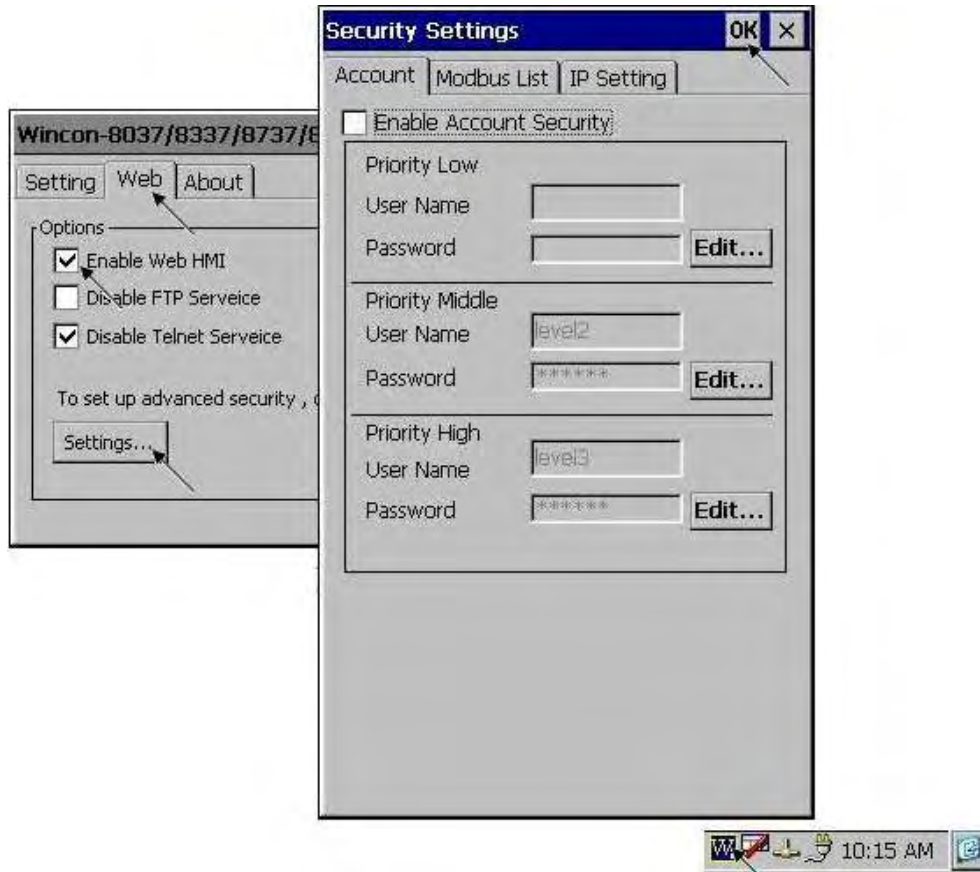
但此 最短取樣時間 也跟程式的複雜度有關, 若您的 PLC 程式 執行起來的 PLC Scan Time 已經是 100ms, 那只能做 大於 100 ms 的取樣記錄

Web HMI 的原始碼放於 (關於 Web HMI 請參考 Wincon ISaGRAF Getting Started 第 3, 4, 5 章 : W-8xx7 CD-ROM:\napdos\isagraf\wincon\english_manu\ “getting_started_w8337.pdf”) W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI_Demo\ 內的 whim_12 或 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/

若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

如何測試本範例:

1. 首先請把 I-8024 插在 W-8xx7 的 Slot 2, I-8017H 插在 Slot 3, 把 I-8024 的第 1 到 4 個電壓輸出點 連接到 I-8017H 的第 1 到 4 個電壓輸入點. 之後上電給 Wincon, 然後在 ISaGRAF Driver 上啟用 Web HMI 功能, 為方便操作, 請先不要選取 “Enable Account Security”



2. 接下來請把 Web HMI Demo 12 的所有 file 用 ftp (例如在 IE 上輸入 “ftp://10.0.0.103”), 丟到 Wincon 的 \CompactFlash\Temp\HTTP\WebHMI\ 內

Web HMI 的原始碼放於

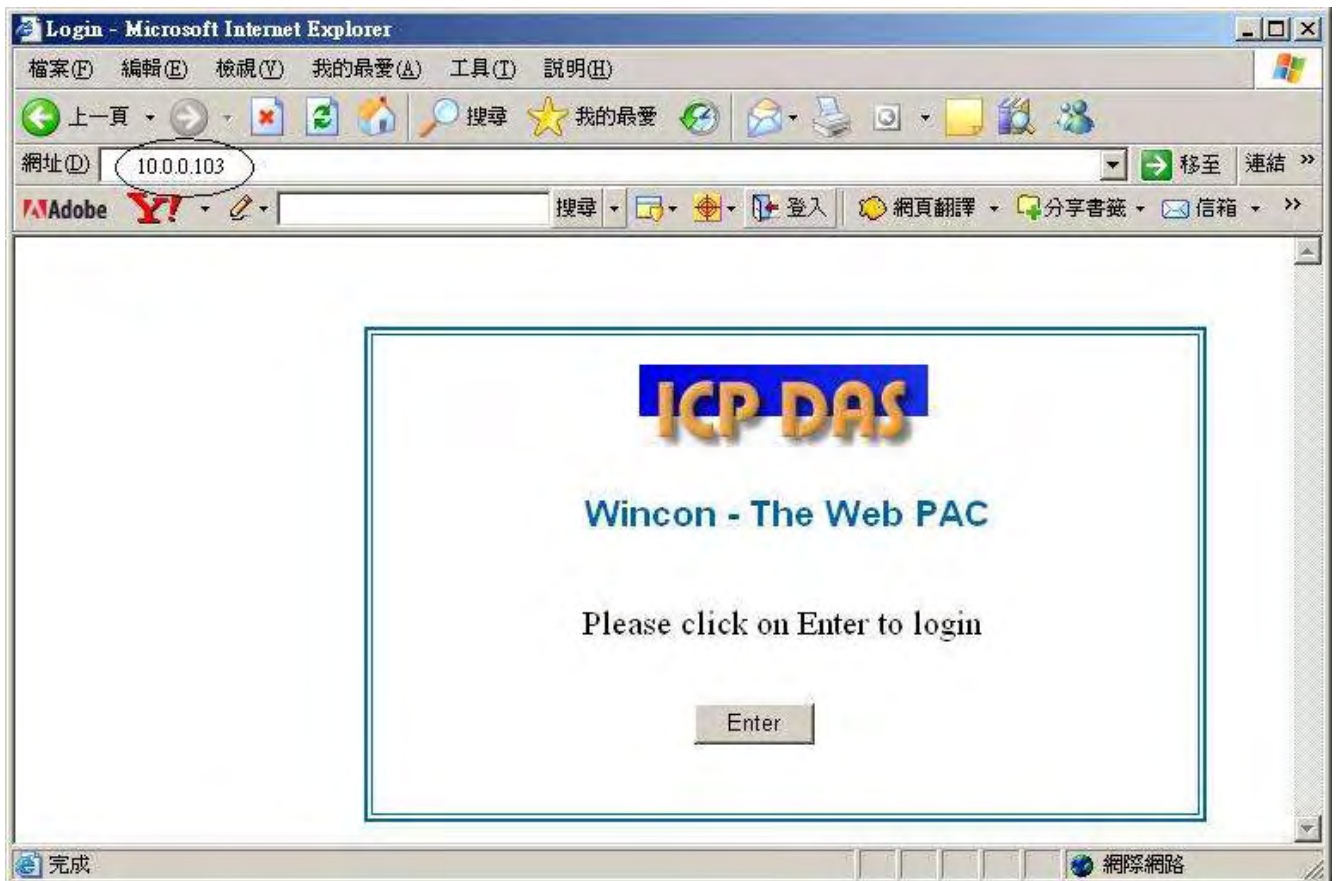
W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI_Demo\ 內的 whim_12 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/

共有 7 個 File 與 2 個目錄要 copy 到 Wincon 的 \CompactFlash\Temp\HTTP\WebHMI\ 內:
Main.htm , menu.htm , index.htm , login.htm , main.dll , login.dll , whim_filter.dll
img 目錄 與 msg 目錄

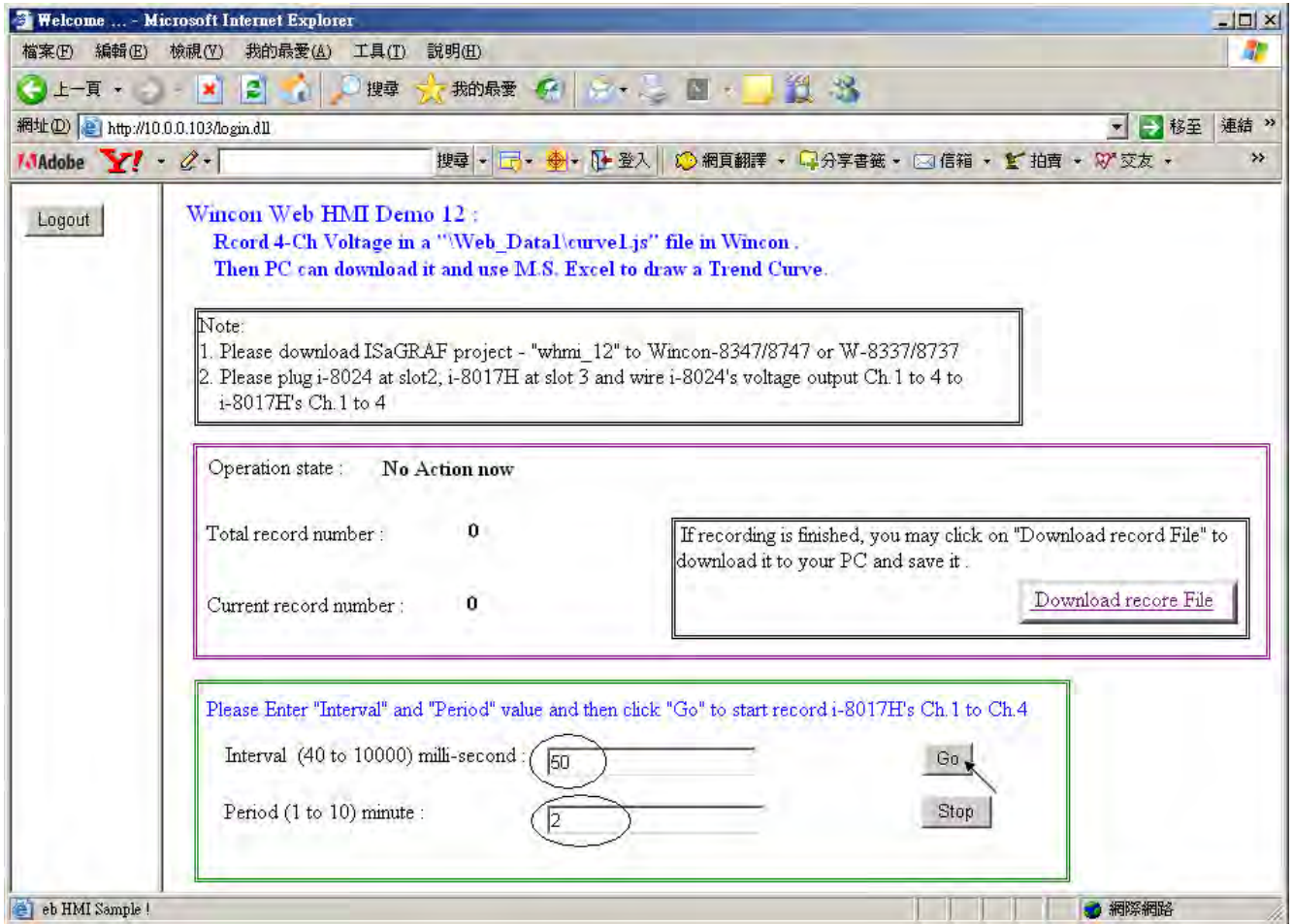
3. 之後請把 ISaGRAF 的 whim_12 程式 下載到 W-8xx7 內. (請做完步驟 2, 再做步驟 3)

本範例程式為 Whmi_12.pia 只能在 Wincon-8xx7/8xx6 內使用, 放於
W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

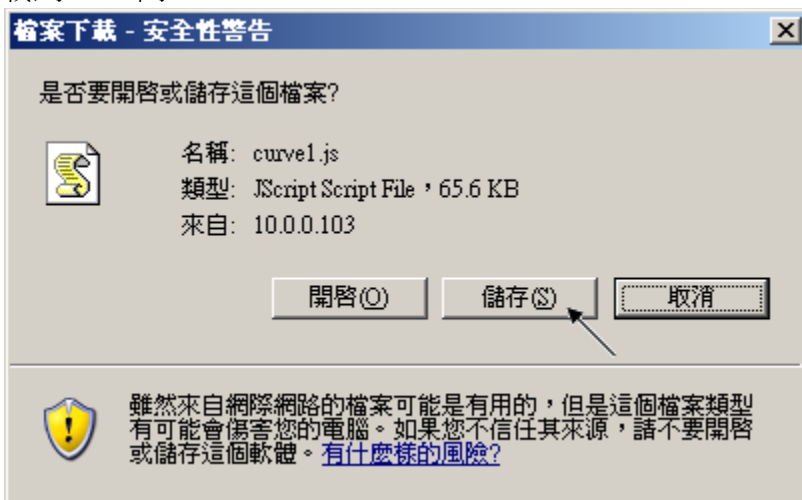
4. PC 上開啟 Internet Explorer (至少需為 IE 5.0 版, 或更高的版本), 上方輸入 W-8xx7 的 IP, 連上 Wincon 後, 請按 “Enter” 進入



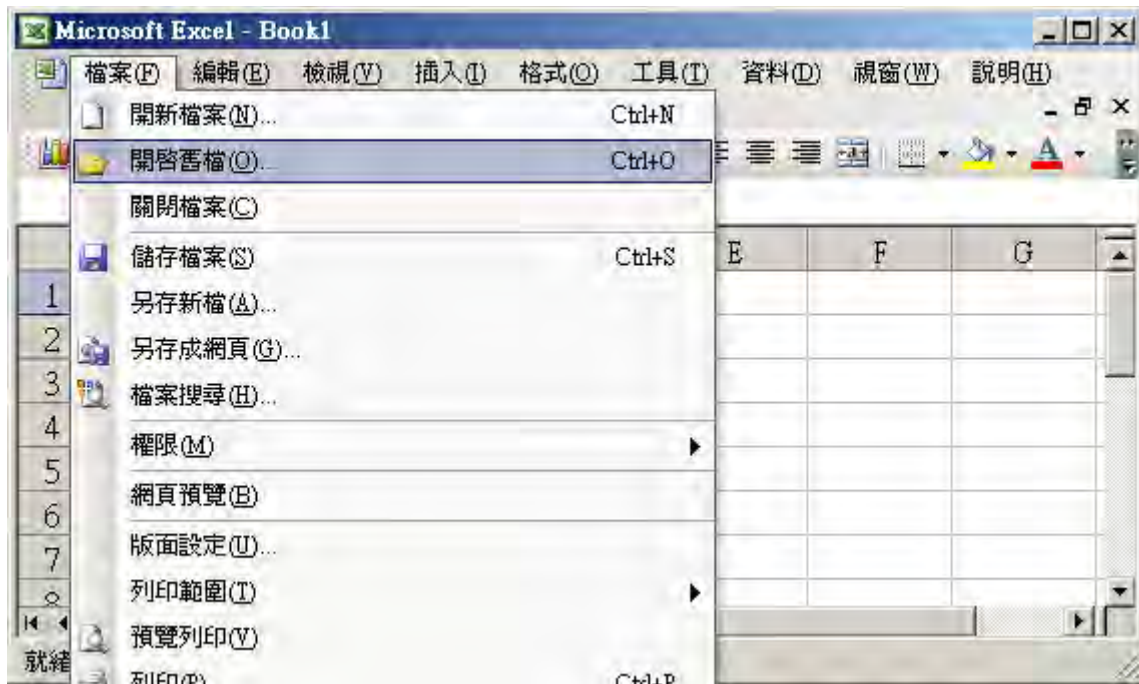
5. 之後請輸入 “Interval” 的值，單位為 0.001 秒，如給 50 指的是 每 0.05 秒，記錄一次。而”Period” 值指的是 要記錄多久，單位為分鐘。 輸入好後，按下 “Go” 可開始動作。 之後 I-8024 的 第 1 到 4 點會輸出不同的電壓，若有在 步驟 1 內將 I-8024 的 第 1 到 4 個電壓輸出點 連接到 I-8017H 的第 1 到 4 個電壓輸入點，則 I-8017H 的輸入會有不同的曲線變化會被記錄下來。



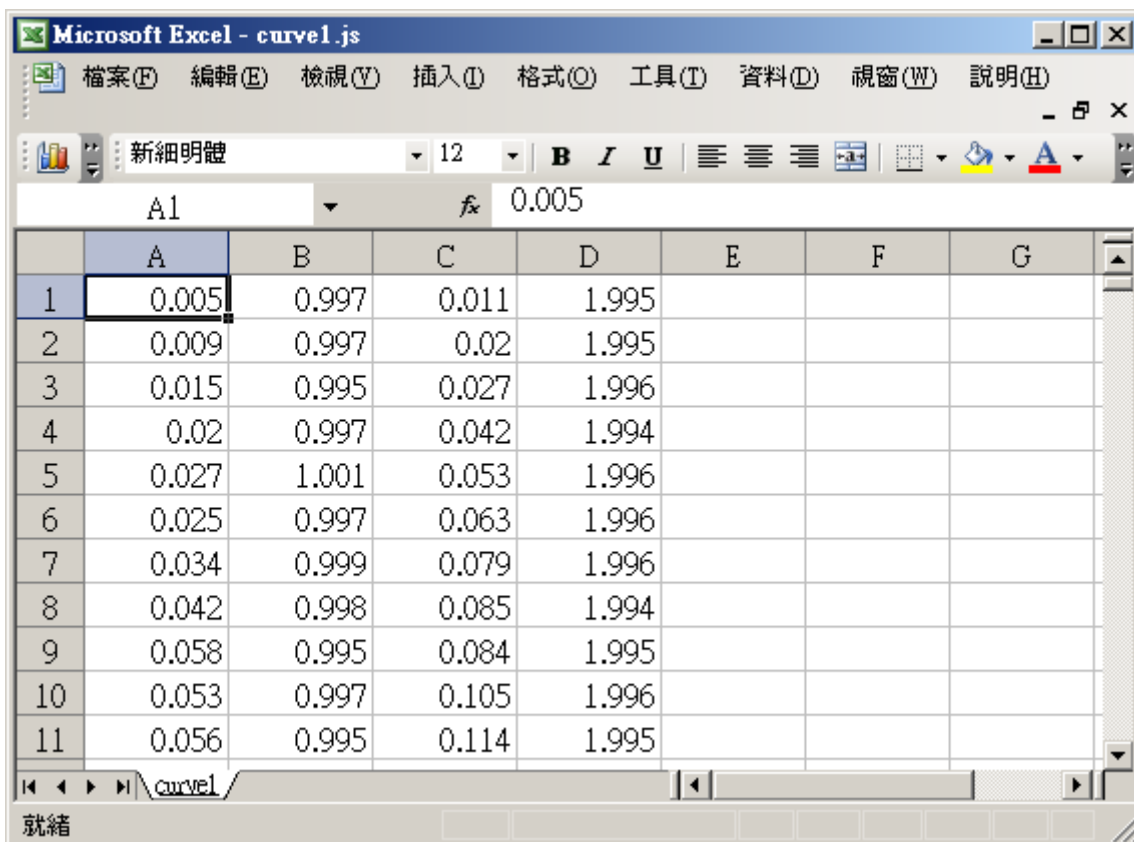
在記錄過程中，會看到 “Current record number” 的值會一直增加，當它等於 “Total record number” 之值時，就表示記錄完畢。此時可以按下 “Download record File”，點選 “儲存” 來下載到 PC 內。



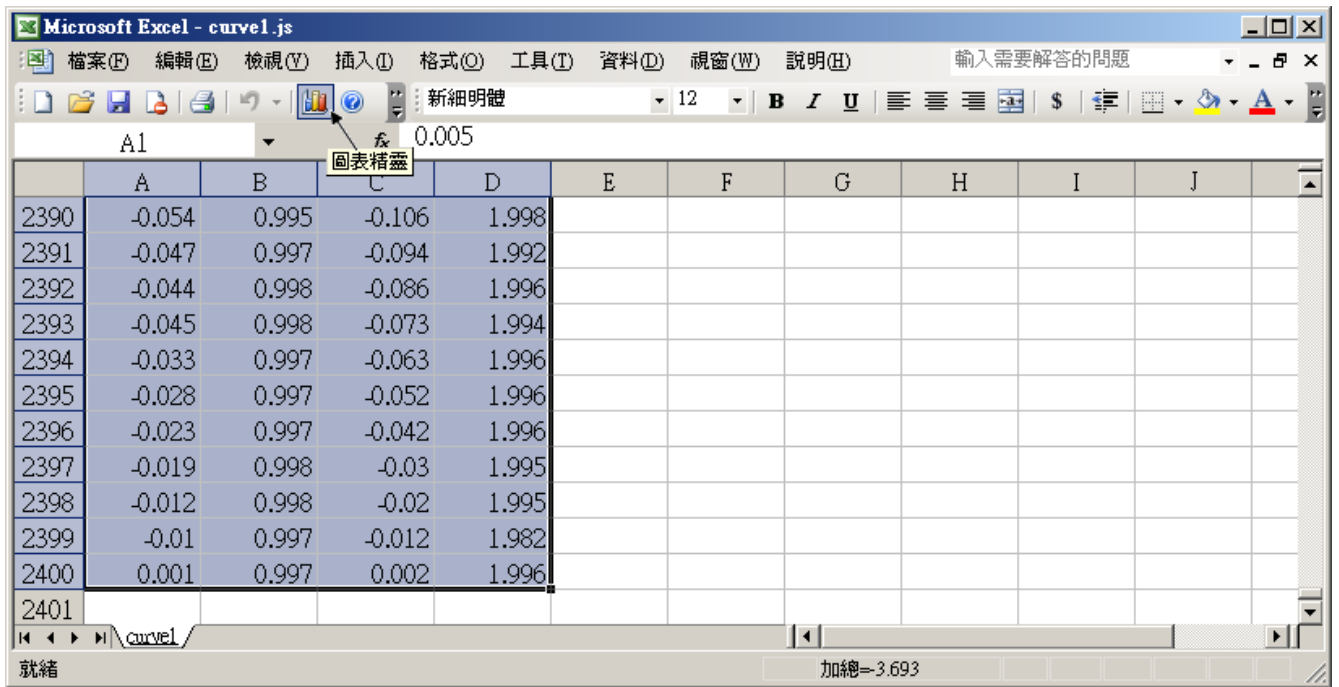
6. 記錄檔 “curver1.js” 下載到 PC 後，請在 PC 上執行 M.S. Excel，開啟 “curvel.js”



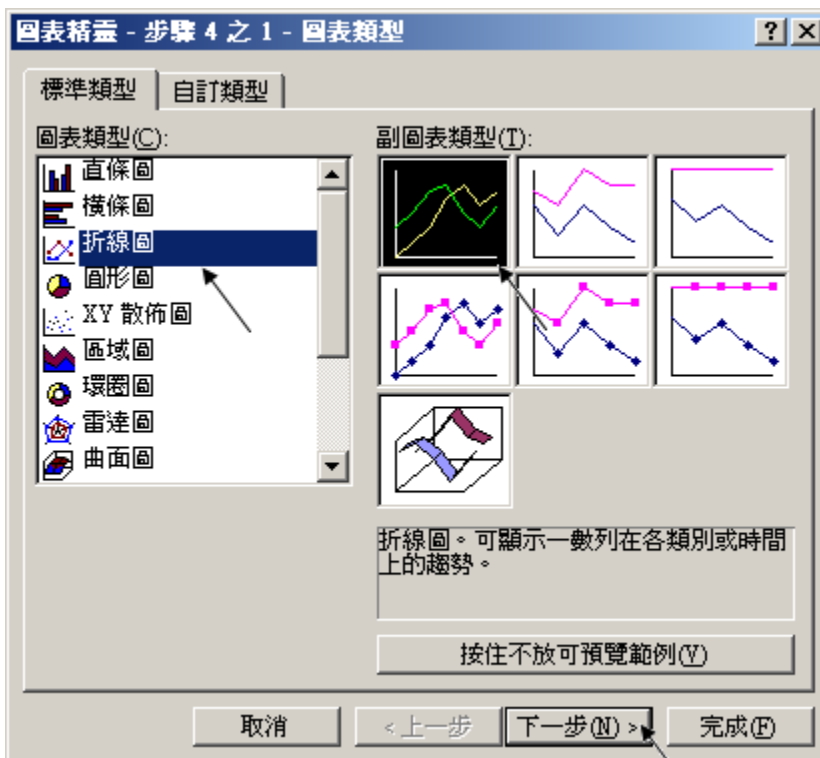
之後請先點選 最左上方的值，再按住 “Shift” 鍵，然後同時 按下 “Ctrl” – “End”，這樣會把所有資料選起來。

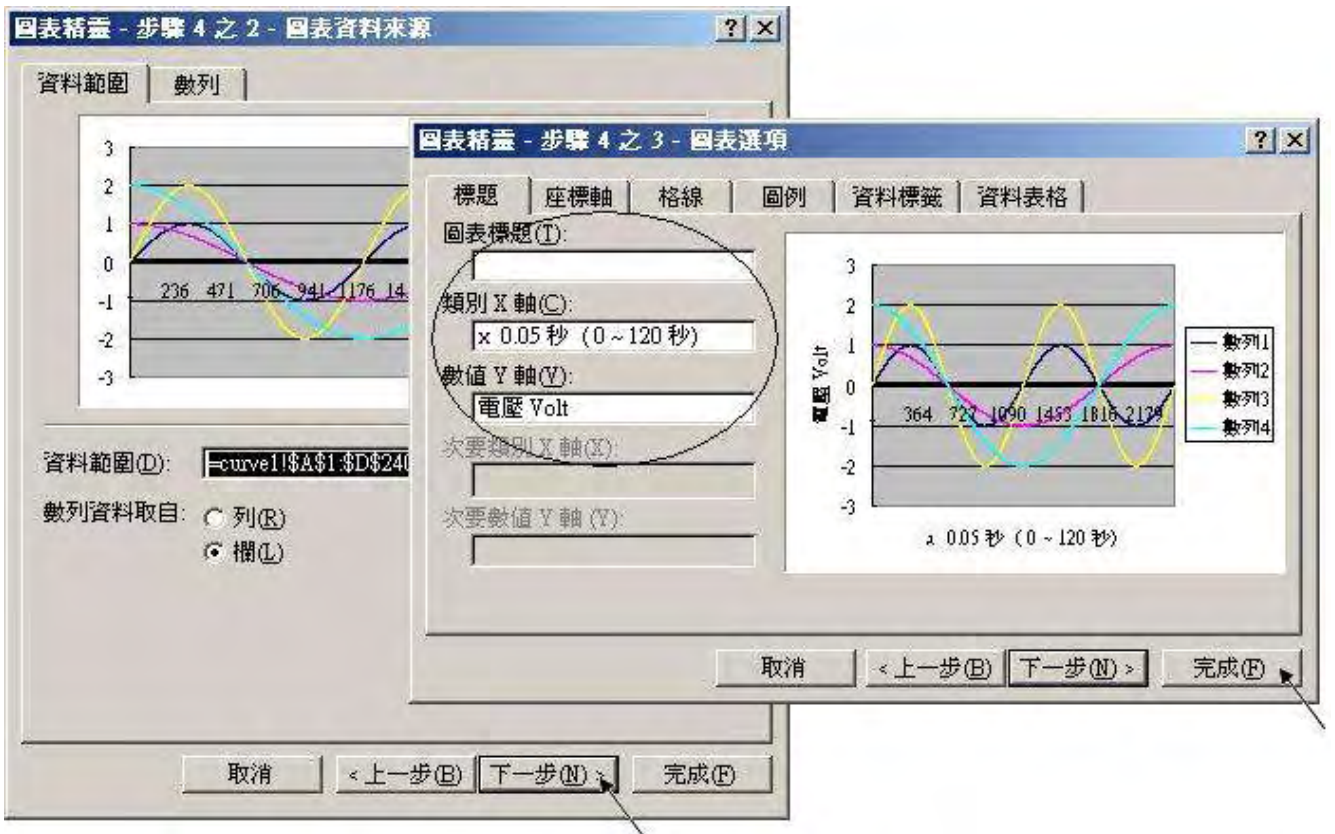


之後請按下 “圖表精靈”

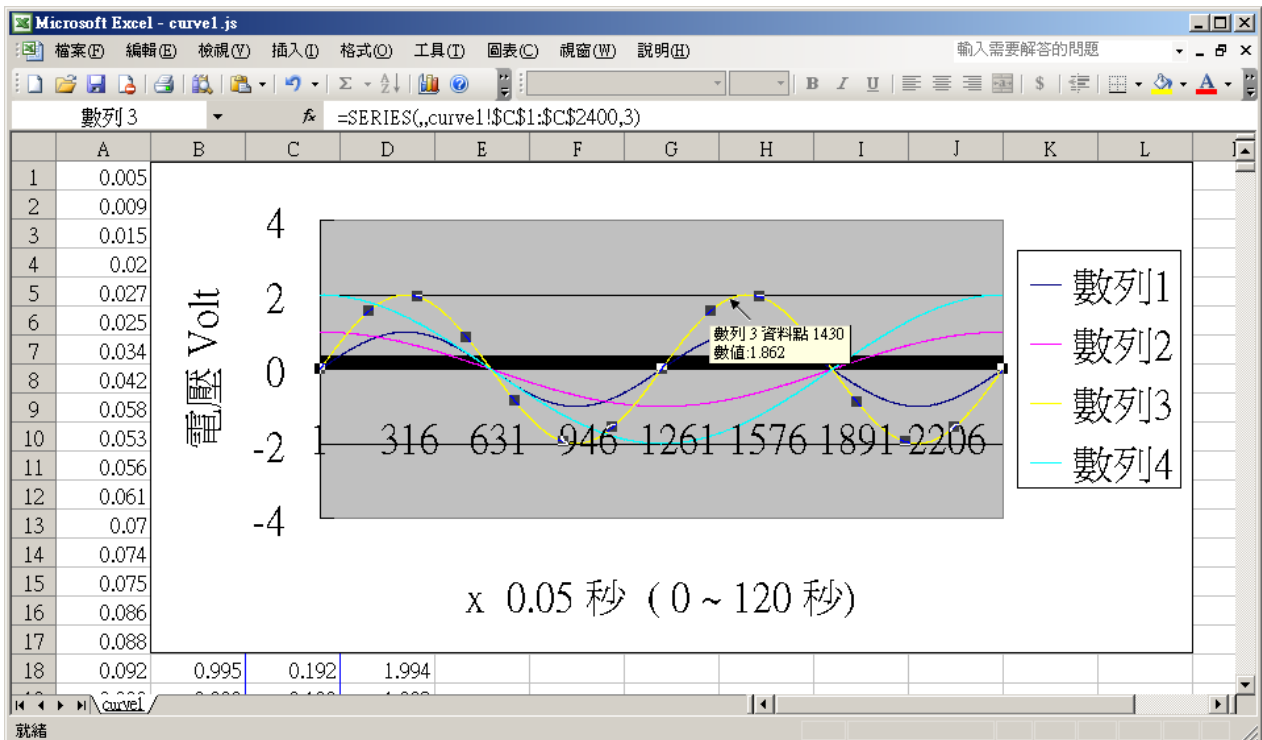


選取“折線圖”，右方請選取 左上角 那一個圖形，再按“下一步”

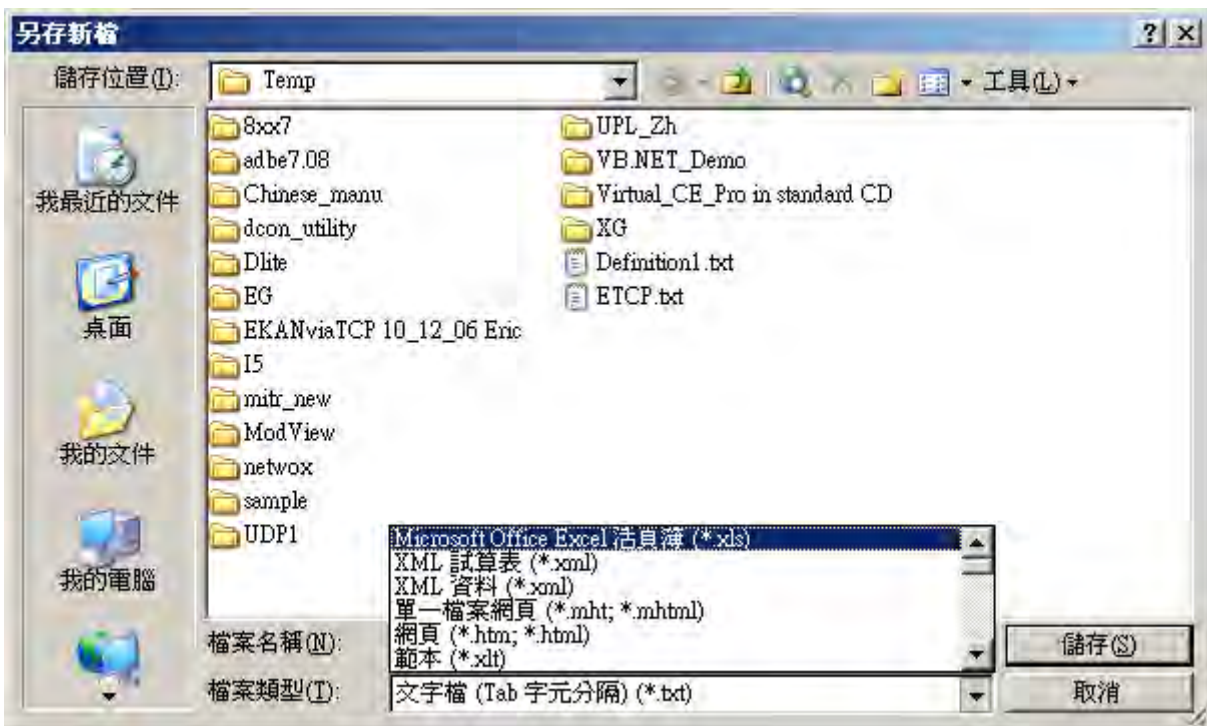
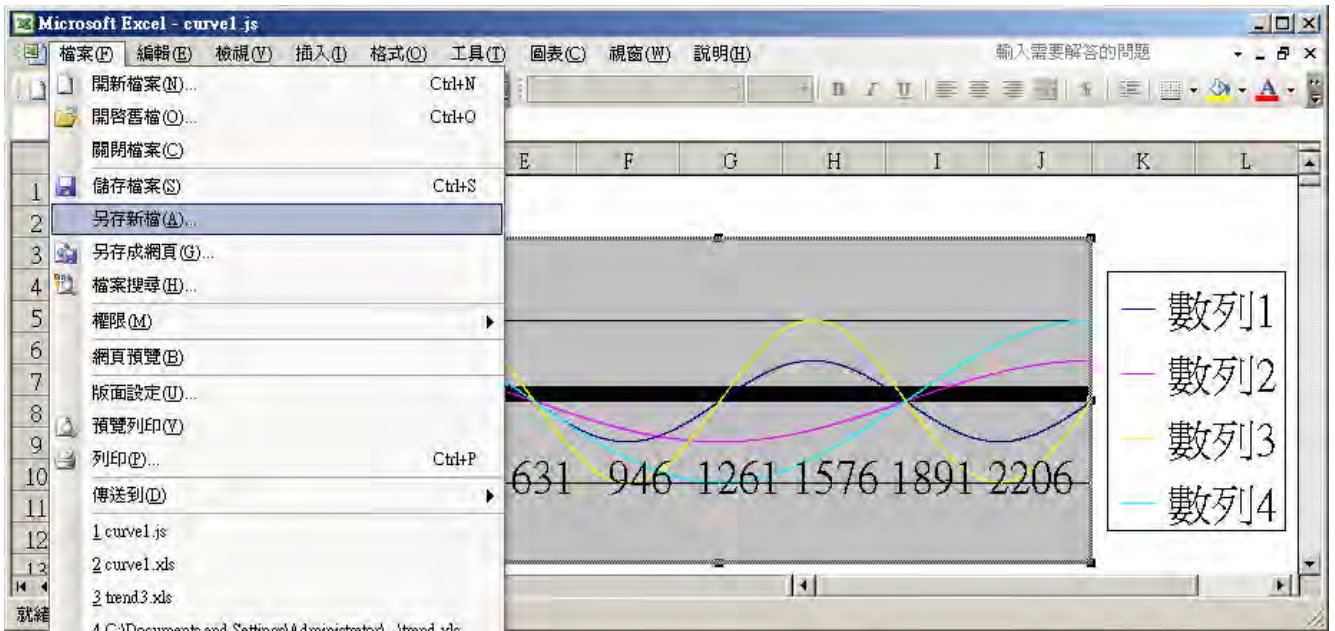




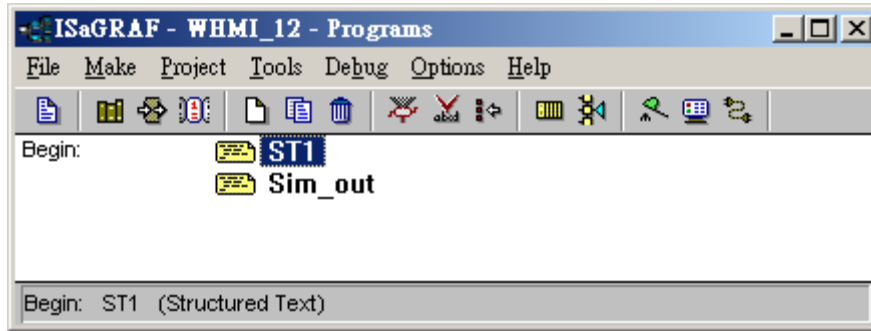
最後得到 趨勢圖如下，您可以調整圖的大小，也可以用滑鼠選取某一條顏色的曲線，當您用滑鼠指到 該曲線的某個位置時，會顯示該點的資料。



為了日後可以方便重覆查詢該曲線，請使用“另存新檔”，將曲線儲存成“Microsoft Office Excel (*.xls)”格式。這樣下回就可直接用滑鼠雙擊“curve1.xls”來直接開啟該趨勢圖。



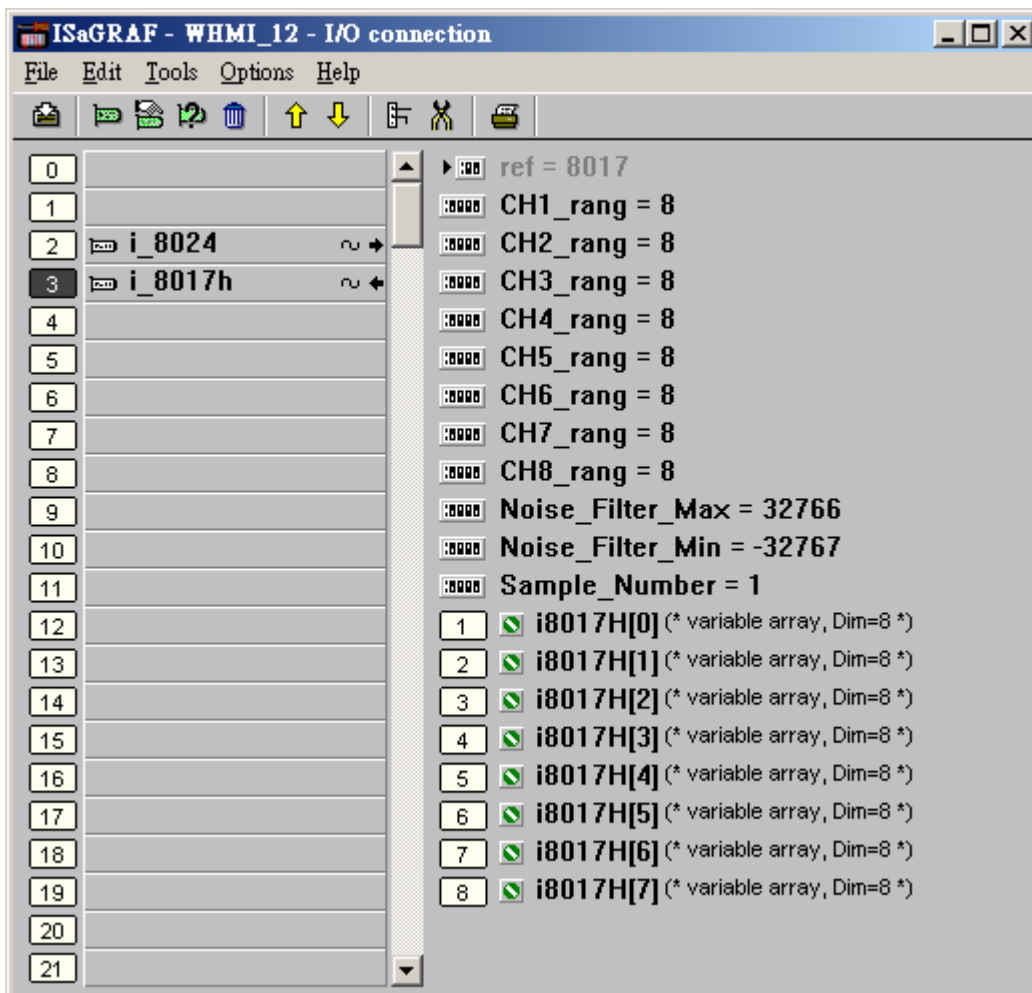
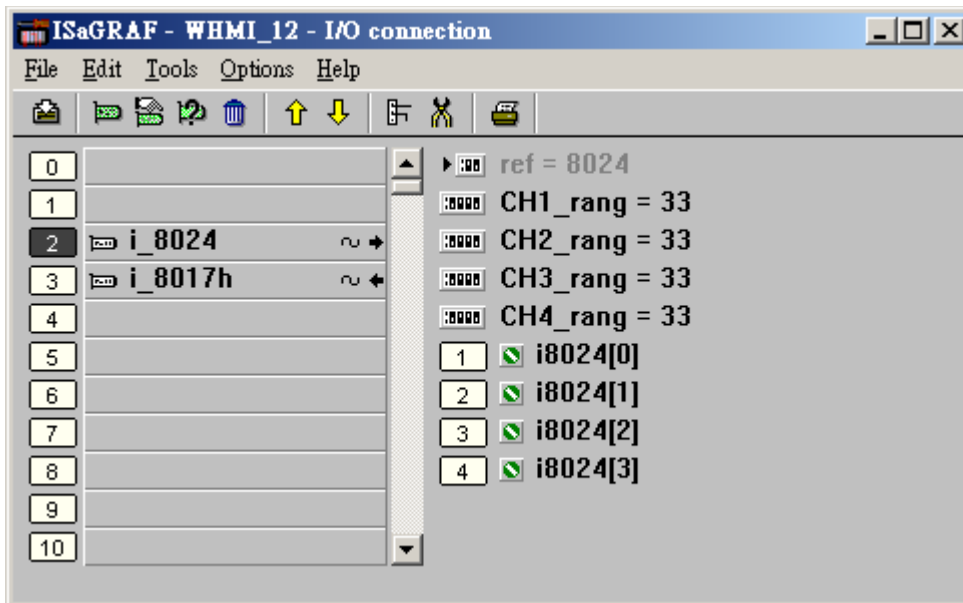
專案架構:



變數定義：

Name	Type	Attribute	Description
Go1	Boolean	Internal	若設為 True 則開始動作, addr 設為 21 (16 進位值為 15)
Stop1	Boolean	Internal	若設為 True 則停止動作, addr 設為 22 (16 進位值為 16)
TMP	Boolean	Internal	暫時使用的變數
MUM_CH	Integer	Constant 常數	I-8017H 上要記錄幾個 Channel 的電壓值 本範例 記錄 4 個 Channel (Ch.1 到 4)
File1	Integer	Internal	File ID
STEP1	Integer	Internal	0:無動作 , 1:記錄中 , 2:記錄完畢
Period1	Integer	Internal	共要記錄多久, 單位為分鐘, addr 設為 3
Interval1	Integer	Internal	多久記錄一筆資料, 單位為 0.001 秒, addr 設為 1
Total_record1	Integer	Internal	本次共有多少筆資料要記錄, 由程式根據 Period1 與 Interval1 之值 自動算出, addr 設為 5
Record_cnt1	Integer	Internal	目前已經完成記錄的資料數量, addr 設為 7
ii	Integer	Internal	給 for 迴圈使用
i8017H[0..7]	Integer	Input	為 變數陣列, Dim 設為 8, 需連接到 I-8017H 的 Ch1. 到 8
Volt1[0..7]	REAL	Internal	為 變數陣列, Dim 設為 8, I-8017H 的 Ch1. 到 8 轉換後的電壓值
i8024[0..3]	Integer	Output	為 變數陣列, Dim 設為 4, 需連接到 I-8024 Ch1. 到 4
T1	Timer	Internal	作為計時用的 Timer
T1_next	Timer	Internal	下一筆資料將於何時去記錄下來, 程式自動算出
T1_Interval	Timer	Internal	每 2 筆記錄的時間 間隔
File_name1	Message	Internal	File name, 長度 64, 初值為 \Web_Data1\curve1.js Web HMI 只支持 RAM Disk File 於 \Web_Data1\ 內 若是 CompactFlash File 則 Web HMI 只支持於 \CompactFlash\Temp\HTTP\Data\ 內 (參見 11.2 節的 Whmi_08 範例)
Msg1	Message	Internal	操作狀態, 長度 255, 初值為 “無動作”, addr 設為 41 (16 進位值為 29)
Str1	Message	Internal	長度 255, 程式內部使用

I/O 連結:



ST 程式 – Sim_out

(* 讓 I-8024 的 Ch1 到 Ch4 輸出不同的 Sin , Cos 電壓曲線 *)

(* $2 * \text{Pi} * \text{T1} / 60000 = \text{T1} * 1.047197\text{E-}4$ *)

(* $2 * \text{Pi} * \text{T1} / 120000 = \text{T1} * 5.235985\text{E-}5$ *)

i8024[0] := ANA(sin(REAL(T1) * 1.047197E-4) * 3276.8) ;

i8024[1] := ANA(cos(REAL(T1) * 5.235985E-5) * 3276.8) ;

i8024[2] := ANA(sin(REAL(T1) * 1.047197E-4) * 6553.6) ;

i8024[3] := ANA(cos(REAL(T1) * 5.235985E-5) * 6553.6) ;

ST 程式 – ST1

(* 若 Stop1 被設為 TRUE, 則停止 T1 的計時, STEP1 設為 0 *)

if Stop1 then

Stop1 := False ;

STEP1 := 0 ; (* STEP1 設為 0:無動作 *)

TStop(T1) ; (*停止 T1 的計時 *)

T1 := T#0s ;

Msg1 := 'User stop recording !' ;

end_if ;

(* 若 Go1 被設為 TRUE, 準備開始動作 *)

if Go1 then

Go1 := False ;

(* STEP1 : 0:無動作 , 1:記錄中 , 2: 記錄完畢 *)

if STEP1 = 1 then

Msg1 := 'It is still recording now ...' ; (* 更新操作狀態的顯示 為 “仍在記錄中 ...” *)

else

(* 檢查 Interval1 之值 設的 對不對, 需為 40 到 10000 之間, 單位為 0.001 秒 *)

If (Interval1 < 40) or (Interval1 > 10000) then

Msg1 := 'Wrong Interval value, it should be in 40 to 10000 milli-second !' ;

(* 檢查 Period1 之值 設的 對不對, 需為 1 到 10 之間, 單位為 分鐘 *)

elsif (Period1 < 1) or (Period1 > 10) then

Msg1 := 'Wrong Period value, it should be in 1 to 10 minute !' ;

else

(* 值設的正確 , 接下來 建立新檔案 *)

File1 := F_creat(File_name1) ;

```

if File1=0 then      (* 建立新檔案若失敗, 就不動作 *)

    Msg1 := 'Create File ' + 'File_nam1 Error !!!' ;

else                (* 建立新檔案 若成功 , 開始動作 *)

    TMP := F_close( File1 ) ;      (* 檔案只要操作完, 就需關閉 *)

    total_record1 := ( Period1 * 60000 ) / Interval1 ; (* 算出 total_record1 值 *)
    record_cnt1 := 0 ;      (* 將目前已記錄資料數量設為 0 *)
    STEP1 := 1 ;      (* 將 STEP1 設為 1:記錄中 *)
    Msg1 := 'Recording now ...' ;

    (* 啟動 T1 開始從 0 起計時 *)
    T1 := T#0s ;
    T1_Interval := TMR( Interval1 ) ;
    T1_next := T1 + T1_Interval ;
    TStart( T1 ) ;

    end_if;

end_if ;

end_if ;

end_if ;

if STEP1=1 then      (* 1:記錄中 *)

    if T1 >= T1_next then (*當 T1 計時到下 1 筆該記錄的時間點時, 就準備記錄 1 筆資料 *)

        T1_next := T1_next + T1_Interval ; (* 重新設定 下 1 筆該記錄的時間點 *)

        (*因為 Timer 最多只能計時到 23 小時 59 分 59 秒, 所以提前在 剛過 20 小時之後, 就 Reset 為 0 *)
        (*本範例最多只有計時 10 分鐘, 所以不會超過, 以下這段的用意, 在於 給使用者參考用 *)
        (* Wincon 的 RAM Disk 容量若只有跑 ISaGRAF, 應該最多可以寫 File 到 16M bytes *)
        (* ----- *)
        if T1 >= T#20h then
            T1 := T#0s ;
            T1_next := T1 + T1_Interval ;
        end_if ;
        (* ----- *)

        (* 開啟 File 準備寫入一筆資料 *)
        (* ----- *)
        File1 := F_wopen( File_name1 ) ;      (* 開啟為 可讀 / 可寫 模式 *)
        if File1=0 then
            (* 若開啟 File 失敗, 不動作 *)
            Msg1 := 'Save File ' + File_name1 + ' Error !!!' ;
            STEP1 := 0 ;

```

```

    Tstop( T1 );      (* 停止 T1 的計時 *)
    return ;        (* 跳離此 ST 程式 *)
    T1 := T#0s ;
end_if ;
(* ----- *)

str1 := '' ;      (* 一開始將 str1 設為空字串 *)
for ii := 0 to NUM_CH - 1 do

    (* 轉換 i8017H 的類比輸入值為 電壓值 *)
    Volt1[ii] := Real( i8017H[ii] ) * 0.000305176 ; (* 10.0 / 32768 = 0.000305176 *)
    str1 := str1 + Rea_Str2( Volt1[ii], 3 ) + '$09' ; (* 分隔字元為 <TAB> *)

end_for ;

str1 := str1 + '$0D$0A' ; (* 加入 <CR><LF> 字元於每行尾端 *)
TMP := F_end( File1 ) ; (* 移到 檔案的最後方 來寫入 *)
TMP := F_writ_s( File1 , str1 ) ; (* 寫入 字串 *)
TMP := F_close( File1 ) ; (* 檔案只要操作完, 就需關閉 *)

record_cnt1 := record_cnt1 + 1 ; (* 目前已記錄資料數量 加 1 *)

(* 檢查是否所有資料都已記錄完畢 *)
if ( record_cnt1 >= total_record1 ) then
    STEP1 := 2 ; (* 將 STEP1 設為 2: 記錄完畢 *)
    Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;
    Tstop( T1 ); (* 停止 T1 的計時 *)
    T1 := T#0s ;
end_if ;

end_if ;

end_if;
-----

```

11.3.7: 範例 Demo_71: 使用 I-8437-80 或 I-8837-80, 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 S-256 / S-512 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖

本範例程式為 Demo_71, 適合在 I-8437-80 或 I-8837-80 內執行, (80MHz) 驅動程式請使用 3.19 或以上版本, 放於

新的驅動: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm>
範例程式 放於 I-8000 CD-ROM:\napdos\isagraf\8000\demo\ 內 或
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

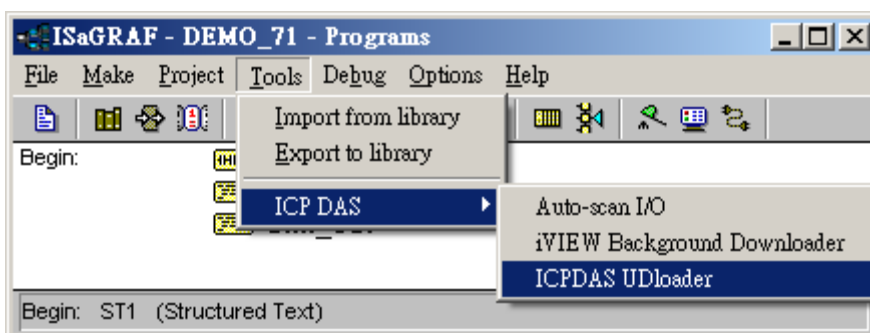
注意: iP-8xx7 也可,但必需改成使用 I-8017HW 與 I-8024W 板卡.

I-8437-80 與 I-8837-80 的 CPU 為 80MHz, 比 I-8437 與 I-8837 的 40MHz CPU 快約 2~4 倍. 適合用來記錄 最短取樣時間可小到約 25 ms 的資料記錄. 若是採用 I-8437 與 I-8837 (40MHz), 則只適合 100 ms 以上的 取樣時間. 但此 最短取樣時間 也跟程式的複雜度有關, 若您的 PLC 程式 執行起來的 PLC Scan Time 已經是 200ms, 那只能做 大於 200 ms 的取樣記錄.

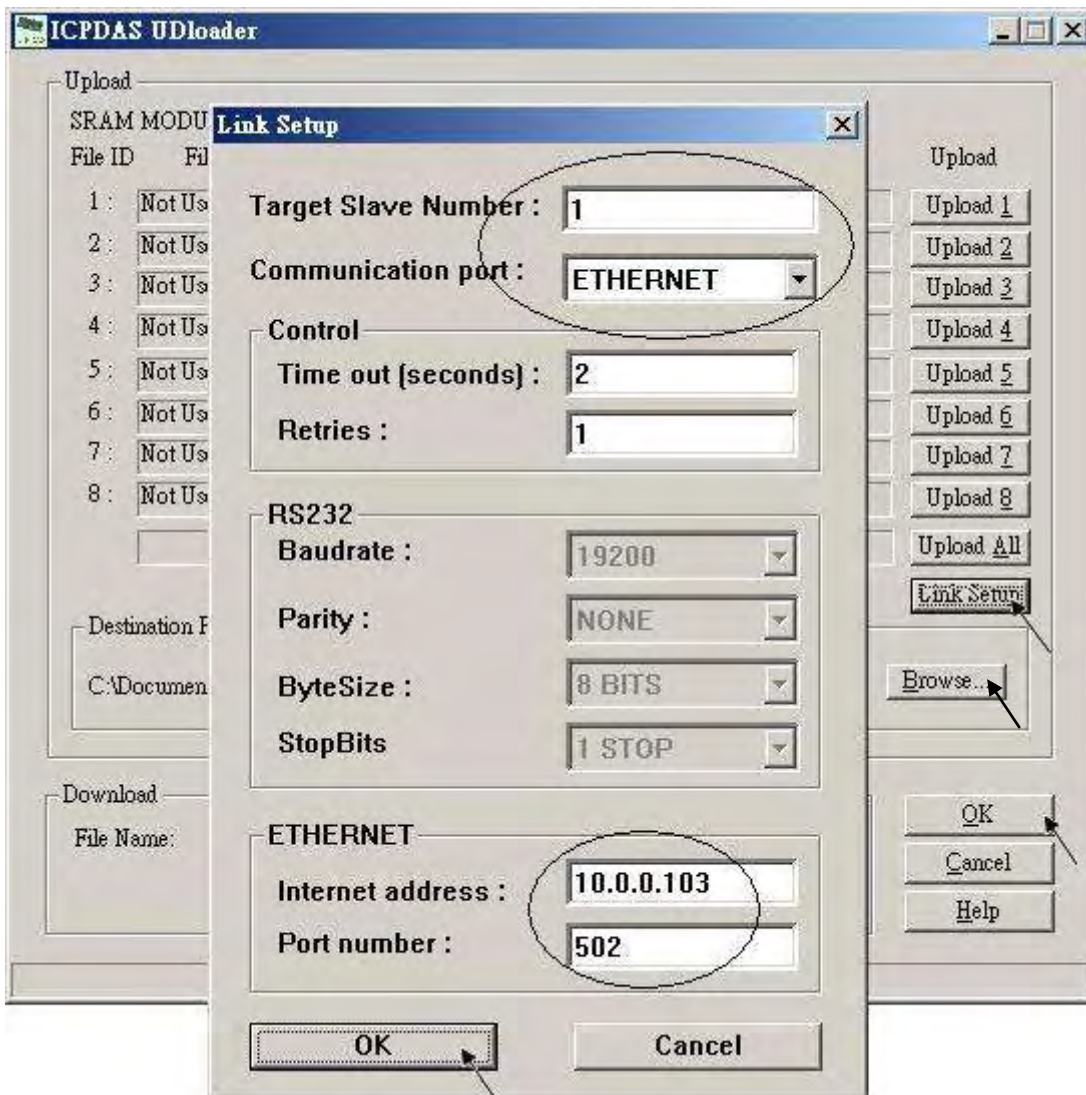
如何測試本範例:

本範例可選擇用 PC 上 VB 6.0 的 “Demo_6” 來連線操作 (請參考第 11.2 節), 或直接在 I-8437-80 / 8837-80 的面板上用按鈕來操作.

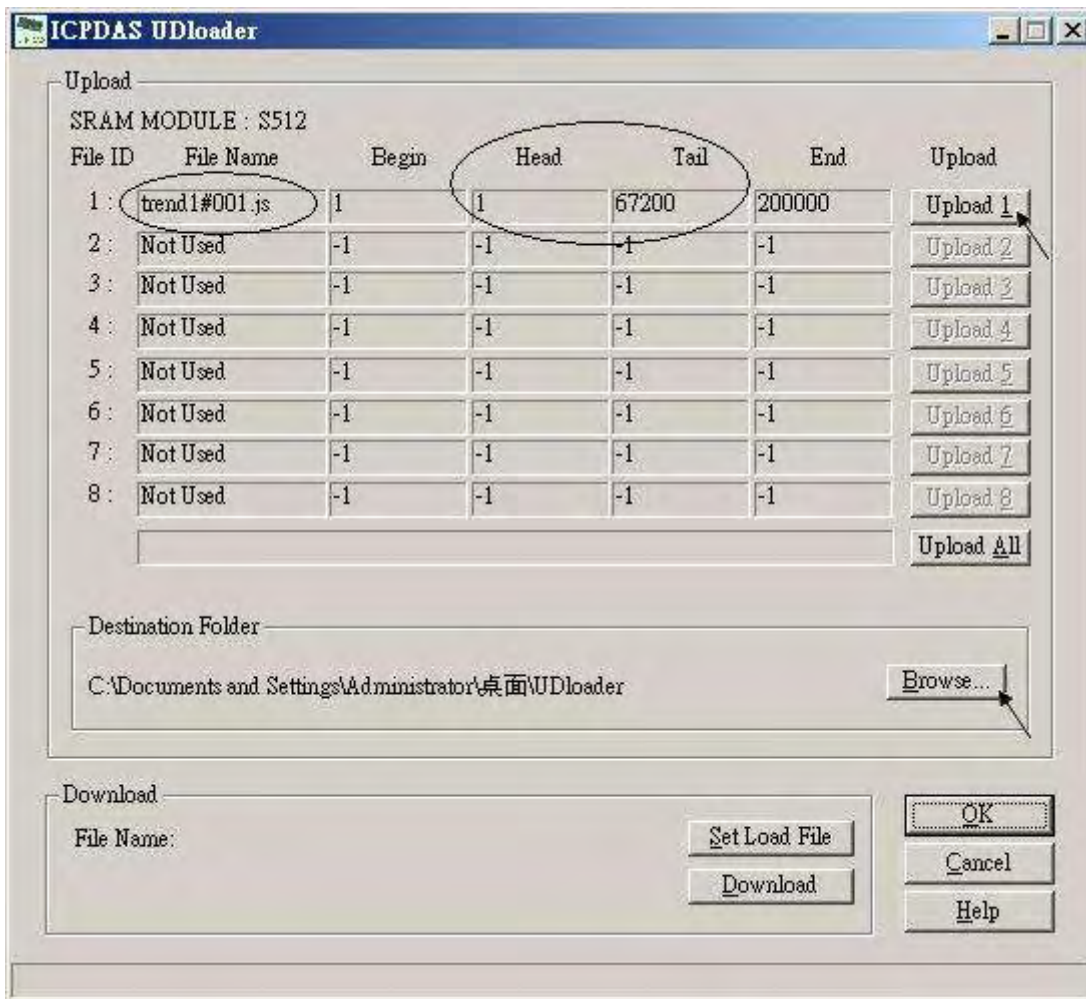
1. 首先請把 I-8024 插在 I-8437-80 的 Slot 2, I-8017H 插在 Slot 3 (注意 I-8xx7 的第 1 個 I/O Slot 為 Slot 0, 跟 W-8xx7 不同, W-8xx7 的第 1 個 I/O 為 Slot 1), 把 I-8024 的 第 1 到 4 個電壓輸出點 連接到 I-8017H 的第 1 到 4 個電壓輸入點. 之後上電給 I-8437-80 .
2. 接下來請把 ISaGRAF “Demo_71” 程式下載到 I-8437-80 內 .
3. Demo_71 執行之後, 可按下 I-8437-80 面板上的第一按鈕, 開始記錄. 之後會開始記錄 2 分鐘, 記錄過程中可看到 數字會遞減到 0 並閃爍, 記錄完畢時, 面板上的 3 個 Led 燈會一直閃爍, 且數字會是 0. (若想停止記錄, 可按第 4 個按鈕)
4. 記錄完後, 可在 PC 上執行 UDownloader 把 S-256/ S-512 內的 File 上載到 PC 上來.
若你的 PC 是設計 ISaGRAF 程式的那台 PC, 請在 ISaGRAF 的 Tools – ICP DAS – ICPDAS UDownloader 把 UDownloader 運行起來. 運行起來後應該會看到步驟 (5) 的畫面.



若你的 PC 沒有 ISaGRAF, 請將 I-8000 CD:\napdos\isagraf\some_utility\ 內的 udloader.exe 複製到 PC 的桌面上. 然後把它 執行起來. 請先點選 “Link Setup” 設好正確的 參數 (若按下 “Browse” 可選擇要把 upload 的檔案存放於 PC 的那個路徑). 之後按下 “OK” & “OK”, 儲存此設定, 並先離開 UDloader.exe , 之後再執行它.



5. 接下來 若有正確連上, 會看到有個 File Name 出現. 從 Head 與 Tail 的數字可看出該 S-256 / S-512 內的記錄檔的 大小. 以下的例子為 67,200 bytes. (注意 S-256 / S-512 有大小限制, 請參閱第 10.3 節). 若想改變 此 File 要上載後存放的位置, 可以點選 “Browse ...” 來變更, 之後請按下 “Upload 1” 來上載它.

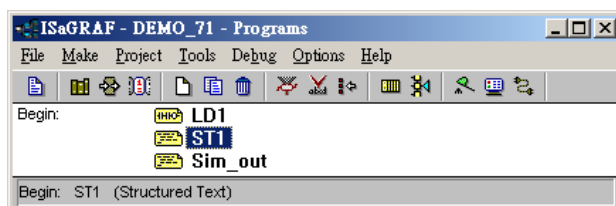


你可以到 PC 同路徑下檢查是否有上載成功.

6. 開啟 M.S. Excel 畫出趨勢圖. 請參考第 11.3.6 節的步驟 (6), 方法與它相同.

注意: 本範例可選擇用 PC 上 VB 6.0 的 “Demo_6” 來連線操作 (請參考第 11.2 節)

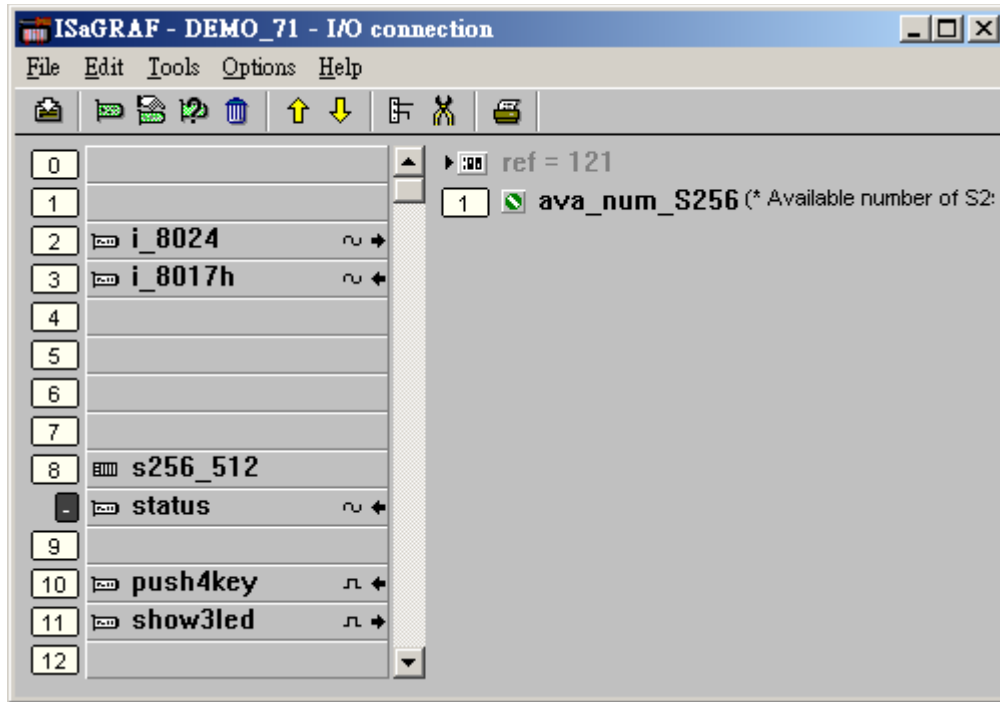
專案架構:



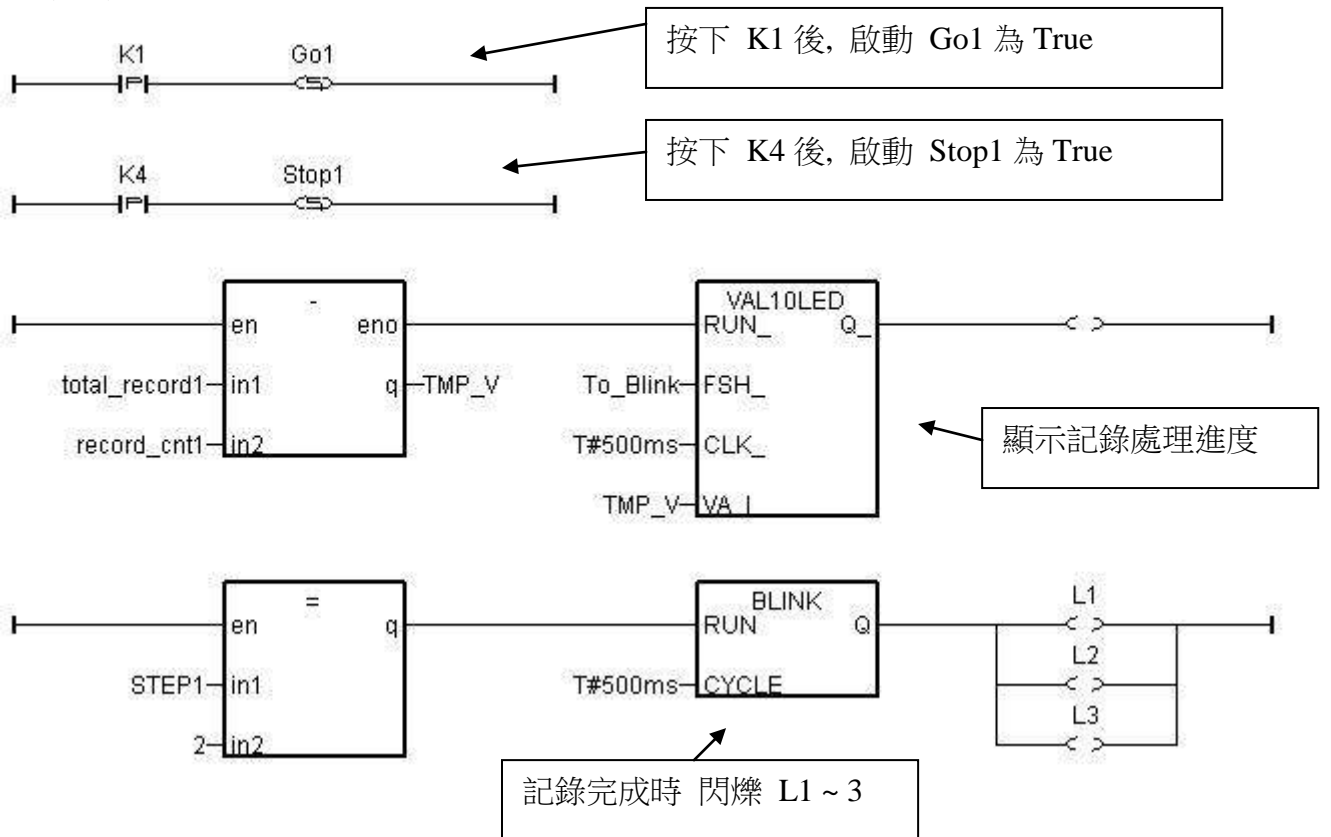
變數定義：

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 True
Go1	Boolean	Internal	若設為 True 則開始動作, addr 設為 21 (16 進位值為 15)
Stop1	Boolean	Internal	若設為 True 則停止動作, addr 設為 22 (16 進位值為 16)
TMP	Boolean	Internal	暫時使用的變數
L1 , L2 , L3	Boolean	Output	連結到 show3Led 的 Ch.1 ~ Ch.3 上
K1 , K4	Boolean	Input	連結到 push4key 的 Ch.1 與 Ch.4 上 K1 用來啟動, K4 用來停止
To_Blink	Boolean	Internal	用來控制面板數字要不要閃爍
MUM_CH	Integer	Constant 常數	I-8017H 上要記錄幾個 Channel 的電壓值 本範例 記錄 4 個 Channel (Ch.1 到 4)
Ava_num_s256	Integer	Input	連結到 S256_512 的 Ch.1 , 值若為 0 表示找不到 S256/512
Current_pos1	Integer	Internal	目前正在記錄 S256/512 的那個 byte 位址
Len1 , TMP_V	Integer	Internal	程式內部使用
File_begin1 File_end1	Integer	常數	S-256 / S-512 內要配置的 Byte 開頭 與 結尾 給 File 使用
STEP1	Integer	Internal	0:無動作 , 1:記錄中 , 2:記錄完畢
Period1	Integer	Internal	共要記錄多久, 單位為分鐘, addr 設為 3
Interval1	Integer	Internal	多久記錄一筆資料, 單位為 0.001 秒, addr 設為 1
Total_record1	Integer	Internal	本次共有多少筆資料要記錄, 由程式根據 Period1 與 Interval1 之值 自動算出, addr 設為 5
Record_cnt1	Integer	Internal	目前已經完成記錄的資料數量, addr 設為 7
ii	Integer	Internal	給 for 迴圈使用
i8017H[0..7]	Integer	Input	為 變數陣列, Dim 設為 8, 需連接到 I-8017H 的 Ch1. 到 8
Volt1[0..7]	REAL	Internal	為 變數陣列, Dim 設為 8, I-8017H 的 Ch1. 到 8 轉換後 的電壓值
i8024[0..3]	Integer	Output	為 變數陣列, Dim 設為 4, 需連接到 I-8024 Ch1. 到 4
T1	Timer	Internal	作為計時用的 Timer
T1_next	Timer	Internal	下一筆資料將於何時去記錄下來, 程式自動算出
T1_Interval	Timer	Internal	每 2 筆記錄的時間 間隔
Msg1	Message	Internal	操作狀態, 長度 255, 初值為 “無動作”, addr 設為 41 (16 進位值為 29)
Str1	Message	Internal	長度 255, 程式內部使用

I/O 連結:



LD 程式 – LD1



ST 程式 – Sim_out

(* 讓 I-8024 的 Ch1 到 Ch4 輸出不同的 Sin , Cos 電壓曲線 *)

(* 2 * Pi * T1 / 60000 = T1 * 1.047197E-4 *)
(* 2 * Pi * T1 / 120000 = T1 * 5.235985E-5 *)

i8024[0] := ANA(sin(REAL(T1) * 1.047197E-4) * 3276.8) ;
i8024[1] := ANA(cos(REAL(T1) * 5.235985E-5) * 3276.8) ;
i8024[2] := ANA(sin(REAL(T1) * 1.047197E-4) * 6553.6) ;
i8024[3] := ANA(cos(REAL(T1) * 5.235985E-5) * 6553.6) ;

ST 程式 – ST1

if INIT then
 INIT := FALSE ; (* 本區一開機只 Run 一次 *)
 if ava_num_S256 = 0 then (* 一開機 若找不到 S-256/512 就離開 *)
 Msg1 := 'S256 / S512 is not installed in I-8xx7 controller !' ;
 Return ;
 end_if ;
 (* 在 S256/512 內配置 200K bytes 給 File 編號 1 使用, 檔名為 “trend1.js” *)
 TMP := S_FL_INI(1 , 'trend1.js' , File_begin1 , File_end1) ;
 TMP := S_FL_AVL(1 , -1 , -1) ; (* 一開始先設 File 內無資料 *)
end_if ;

if ava_num_S256 = 0 then (* 每個 PLC Scan 若發現 找不到 S-256/512 就離開 *)
 return ;
end_if ;

(* 若 Stop1 被設為 TRUE, 則停止 T1 的計時, STEP1 設為 0 *)
if Stop1 then
 Stop1 := False ;
 STEP1 := 0 ; (* STEP1 設為 0:無動作 *)
 TStop(T1) ; (*停止 T1 的計時 *)
 T1 := T#0s ;
 Msg1 := 'User stop recording !' ;
 To_Blink := False ; (*停止 數字閃爍 *)
end_if ;

TMP_V := S_FL_STS(1) ; (* S_FL_STS 若回傳 非 -1 之值, 表示 PC 已上載過該 File *)
if TMP_V <> -1 then
 TMP := S_FL_RST(1) ; (* 此時再將 State 設為 -1, 讓 PC 可重覆上載該 File *)
end_if ;

(* 若 Go1 被設為 TRUE, 準備開始動作 *)
if Go1 then
 Go1 := False ;

```

(* STEP1: 0:無動作, 1:記錄中, 2: 記錄完畢 *)
if STEP1 = 1 then
    Msg1 := 'It is still recording now ...' ;    (* STEP1 為 1, 表示記錄中 )

else

    (* 檢查 Interval 值, 需在 25 到 10000 ms 之間, 本例使用 50 ms *)
    if ( Interval1 < 25 ) or ( Interval1 > 10000 ) then
        Msg1 := 'Wrong Interval value, it should be in 25 to 10000 milli-second !' ;

    (* 檢查 Period1 之值 設的 對不對, 需為 1 到 10 之間, 單位為 分鐘 *)
    elsif ( Period1 < 1 ) or ( Period1 > 10 ) then
        Msg1 := 'Wrong Period value, it should be in 1 to 10 minute !' ;

    else

        total_record1 := ( Period1 * 60000 ) / Interval1 ;    (* 算出 total_record1 值 *)
        record_cnt1 := 0 ;    (* 將目前已記錄資料數量設為 0 *)
        STEP1 := 1 ;    (* 將 STEP1 設為 1:記錄中 *)
        Msg1 := 'Recording now ...' ;

        (* 啟動 T1 開始從 0 起計時 *)
        T1 := T#0s ;
        T1_Interval := TMR( Interval1 ) ;
        T1_next := T1 + T1_Interval ;
        TStart(T1) ;
        Current_pos1 := 1 ;    (* 將 S256/512 內的資料位址 RESET 為 1 *)
        To_Blink := TRUE ;    (* 設為 TRUE 讓數字閃爍 *)

        end_if ;
    end_if ;
end_if ;

(* 以下為 STEP1 為 1: 記錄中需作的 code *)
if STEP1 = 1 then

    if T1 >= T1_next then    (*當 T1 計時到下 1 筆該記錄的時間點時, 就準備記錄 1 筆資料*)

        T1_next := T1_next + T1_Interval ;    (* 重新設定 下 1 筆該記錄的時間點 *)

    end_if ;

    (*因為 Timer 最多只能計時到 23 小時 59 分 59 秒, 所以提前在剛過 20 小時之後, 就 Reset 為 0 *)
    (* 本範例最多只有計時 10 分鐘, 所以不會超過, 以下這段 的用意, 在於 給使用者參考用 *)
    if T1 >= T#20h then
        T1 := T#0s ;
        T1_next := T1 + T1_Interval ;
    end_if ;

    str1 := '' ;    (* 一開始將 str1 設為空字串 *)
    for ii := 0 to NUM_CH - 1 do

```

```

(* 轉換 i8017H 的類比輸入值為 電壓值 *)
  Volt1[ii] := Real( i8017H[ii] ) * 0.000305176 ;    (* 10.0 / 32768 = 0.000305176 *)
  str1 := str1 + Rea_Str2( Volt1[ii] , 3 ) + '$09' ;    (* 分隔字元為 <TAB> *)
end_for ;

str1 := str1 + '$0D$0A' ;    (* 加入 <CR> <LF> 字元於每行尾端 *)
Len1 := MLEN( str1 ) ;    (* 取得字串 str1 的目前長度 *)

(* 若記錄的資料 byte 數量將超過 S256/512 配置給 File 的使用量, 就離開, 不再記錄下去 *)
if ( Current_pos1 + Len1 - 1 ) > File_end1 then
  STEP1 := 0 ;    (* 設為 0:無動作 *)
  Msg1 := 'File allocated memory is not enough to hold the data !' ;
  Tstop( T1 ) ;    (* 停止 T1 的記時 *)
  T1 := T#0s ;
  To_Blink := FALSE ;    (*停止 數字閃爍 *)
  Return ;    (* 離開此 ST 程式 *)
end_if ;

(* 將字串 str1 內的所有 byte 寫入 S-256/512 的目前位址內 *)
TMP := S_M_W( Current_pos1 , Len1 , str1 ) ;
Current_pos1 := Current_pos1 + Len1 ;    (*目前位址往前移 *)
TMP := S_FL_AVL( 1 , 1 , Current_pos1 - 1 ) ;    (*重設 S256/512 的 File 頭/尾位址 *)

record_cnt1 := record_cnt1 + 1 ;    (* 資料數量加 1 *)
if ( record_cnt1 >= total_record1 ) then    (* 檢查是否所有資料都已記錄完畢 *)
  STEP1 := 2 ;    (* 若是則設 STEP1 為 2:記錄完畢 *)
  Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;
  Tstop( T1 ) ;
  T1 := T#0s ;
  To_Blink := FALSE ;    (*停止 數字閃爍 *)
end_if ;

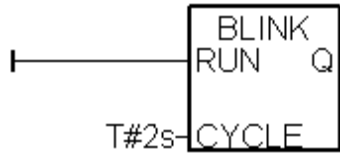
end_if ;
end_if ;

```

11.3.8: 使用較準確的 固定週期 去 做某件事

BLINK 方塊可用來每隔固定一段時間就產生一個 Pulse True, 所以可以使用在每固定一段時間就做一件事的應用上. 如下:

(* LD 程式: *)



Pulse1 宣告為 Boolean Internal
此例會每 2 秒去做一件事

(* ST 程式: *)

IF Pulse1 THEN (* 上方 LD 程式會每 2 秒產生一個 pulse TRUE 於 “pulse1” 變數內 *)

(* 做一件事 *)
(* *)

END_IF ;

但以上的程式在時間間隔較短, 比如小於 200ms 或 PLC Scan Time 較大時會變得不精確. 例如每 50ms 做一件事, 因為 50ms 是比較小的間隔, 跟 PLC Scan Time 比較接近, 就會不準確, 所以可以改成以下方法, 就可以提高準確性.

ST 程式:

```

IF INIT THEN
  INIT := False ;
  T1 := T#0s ;
  T1_next := T1 + T#50ms ;
  Tstart (T1) ;
END_IF ;
  
```

INIT 宣告為 Boolean Internal
並且初始值為 TRUE
T1 與 T1_next 為 Timer Internal

```

IF T1 >= T1_next THEN
  
```

```

  IF T1 > T#22h THEN
    T1 := T#0s ;
    T1_next := T#0s ;
  END_IF ;
  
```

因為 Timer 變數計時到 T#23h59m59s999ms 後
就會發生溢位, 所以可提前在 接近 22h 時就
Reset 為 0

```

  T1_next := T1_next + T#50ms ; (* 算出下一次動作的時間點 *)
  
```

(* 做一件事 *)
(* *)

END_IF ;

11.3.9: Demo_72: I-7188EGD 連接 I-7018z 量測 6 個 4 到 20 mA 的電流輸入值 與 4 個 Thermo-Couple 溫度輸入值，並顯示在 PC 的 VB 6.0 程式上

本範例程式為 Demo_72，可用在 I-7188EG / 7186EG 內執行，若要使用 I-8xx7, XP-8xx7-CE6 或 Wincon-8xx7 也可，但 "Bus7000b" 的 COM port 要改成使用 COM3. (若要用在 WP-8xx7, VP-25W7/23W7，則 Bus7000b" 的 COM port 要改成使用 COM2.)

範例程式 放於 I-8000 CD-ROM:\napdos\isagraf\8000\demo\ 內 或
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

VB 6.0 的 Demo_4 程式 放在

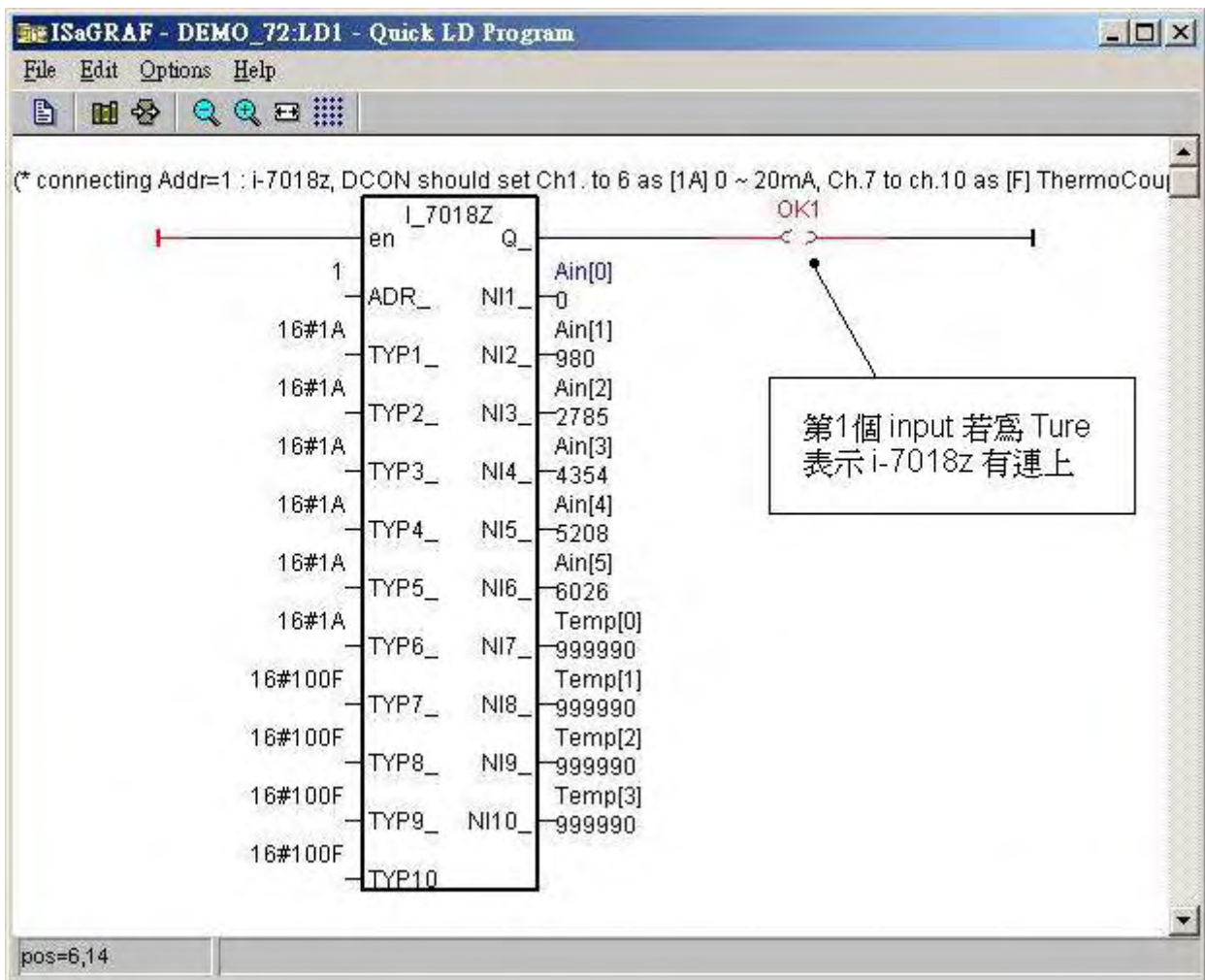
I-8000 CD: \napdos\isagraf\vb_demo\demo_4\demo_4.exe 或
<http://www.icpdas.com/faq/isagraf.htm> FAQ-055

I-7188EG 的 COM2:RS-485 可用來連接 I-7000 系列 或連接 I-87K/4/5/8/9 擴展模組加 I-87xxx I/O 板卡. 最多一台 I-7188EG 可連接 64 個 I-7000 模塊 (或 I-87xxx I/O 板卡, 總數 I-7000+I-87xxx 最多是 64 個). 使用 I-8xx7 的 COM3:RS-485 來接也可, 總數 I-7000+I-87xxx 最多也是 64 個. 若是使用 W-8xx7 的 COM3:RS-485 來接, 可連接的總數 I-7000+I-87xxx 最多是 255 個.

但接的越多 掃瞄完所有 I/O 一圈的時間 就越長, 以 9600 Baud rate 為例, 一般一顆 D/I/O 模塊掃瞄完 約花 20 ~ 40 ms. 若是 A/I/O 一般約 40 ~ 60 ms (跟模塊種類有關, 板卡上有不同 I/O 種類的, 時間會比這個數字長, 比如 I-7050D, 既有 D/I, 又有 D/O, 或 I-87082, 功能多, 花的時間也多). 所以若接了 20 個 D/I/O, 約會花 0.4 ~ 0.8 秒來掃完一圈, 若是 20 個 A/I/O, 則約花 0.8 ~ 1.2 秒 來掃完一圈. 因此建議 I-7188EG/XG 與 I-8xx7 儘量不要連接超過 24 顆. Wincon-8xx7 則建議不超過 64 顆為佳.

如何測試本範例:

1. 請將 PC 安裝好 **DCON utility**, 版本需是 **4.4.3** 版以上 才能使用 I-7018z 與 I-87018z. 可在 I-8000 CD-ROM 內找到 或到 ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/ 內的 setup 路徑內去下載 .
2. 請 參考第 6.1 節的步驟(1) 到 (4) 將 I-7018z 的 Address 設為 1, baud rate 設為 9600, Format 設為 "2's compliment", Checksum disable. 並規劃 I-7018z 的 Ch.1 ~ Ch.6 為 "[1A]: 0 ~ 20 mA", Ch.7 ~ Ch.10 為 "[0F]: T/C K-Type". 規劃完後, 請將 I-7018z 背面的 Dip Switch 撥回 "Normal".
3. 請先將 I-7188EG 的 IP 設為 192.168.1.3 (參考附錄 B 的方法), NET-ID 需為 1, 之後斷電, 連接 I-7188EG 的 COM2 到 I-7018z. 再上電給 7188EG 與 I-7018z. (PC 的 IP 也須在 192.168.1.x 的網域內, 才能連上 I-7188EG, 比如將 PC 設成 192.168.1.1, Mask=255.255.255.0)
4. PC 跑 ISaGRAF 用 Ethernet 下載 demo_72 程式到 I-7188EG 內. (不知如何下載, 請參考第 2.1.5 節 或 1.3.8 節). 下載完 會自動執行起來, 再 開啟 ISaGRAF 階梯圖程式 觀察 I-7018z 是否有連上.
(如下圖)



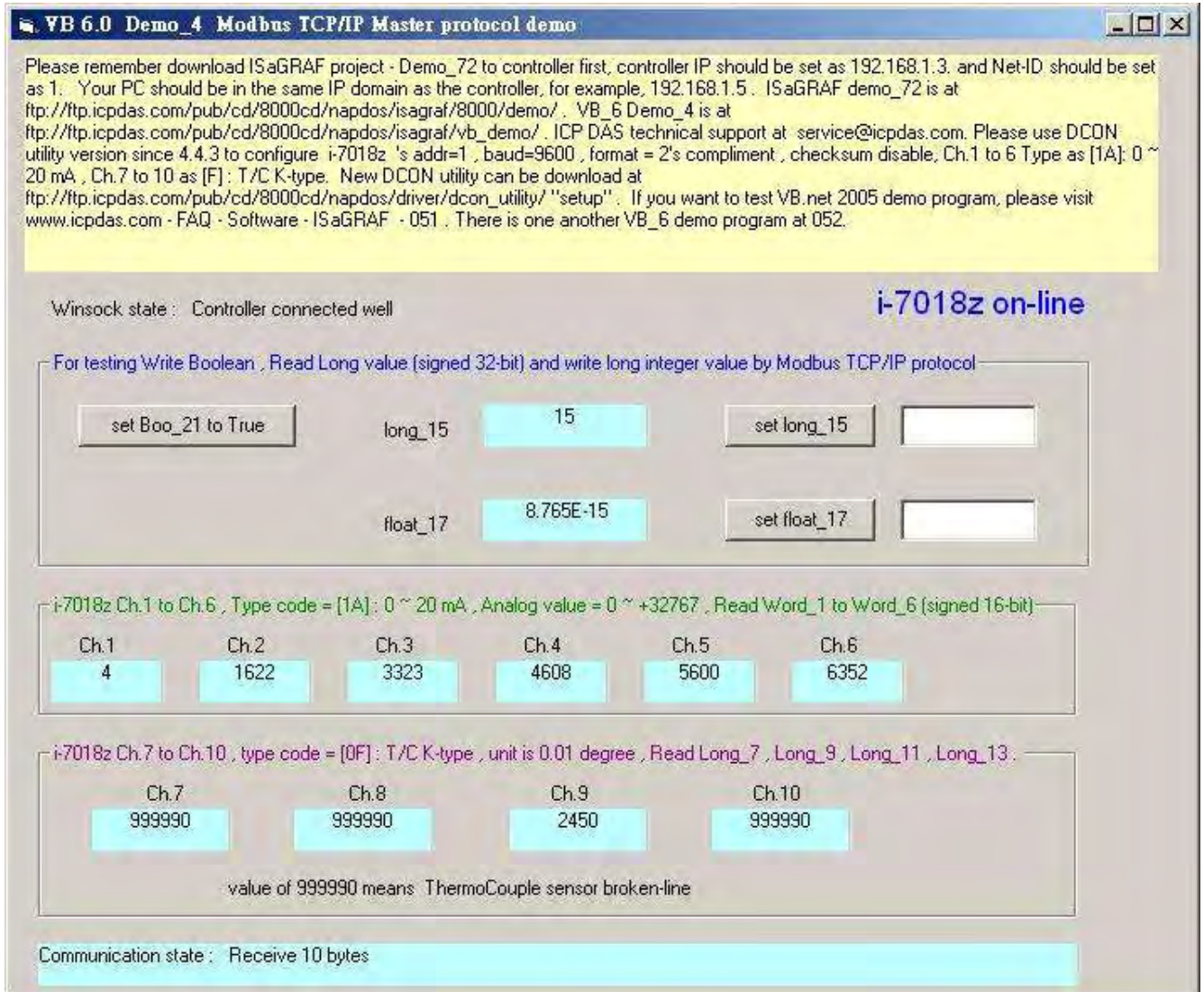
5. 接下來在 PC 上跑 VB 6.0 的 Demo_4.exe 程式，放於
 I-8000 CD: \napdos\isagraf\vb_demo\demo_4\demo_4.exe 或
<http://www.icpdas.com/faq/isagraf.htm> FAQ-055

(如下圖)

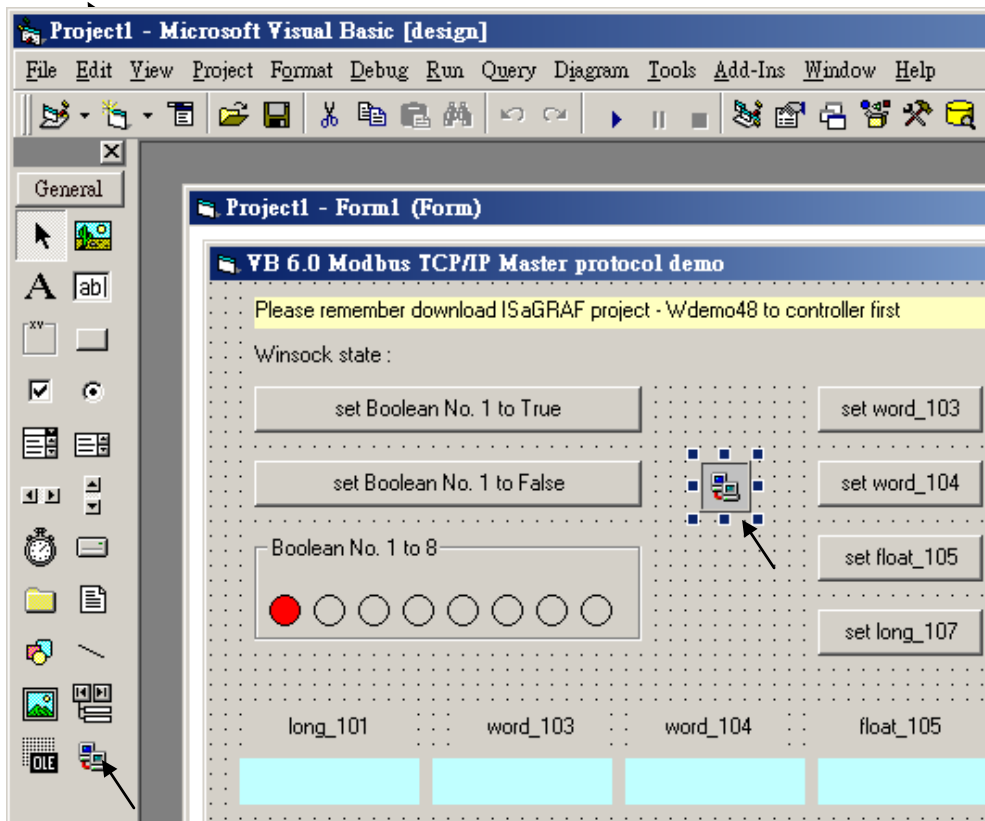
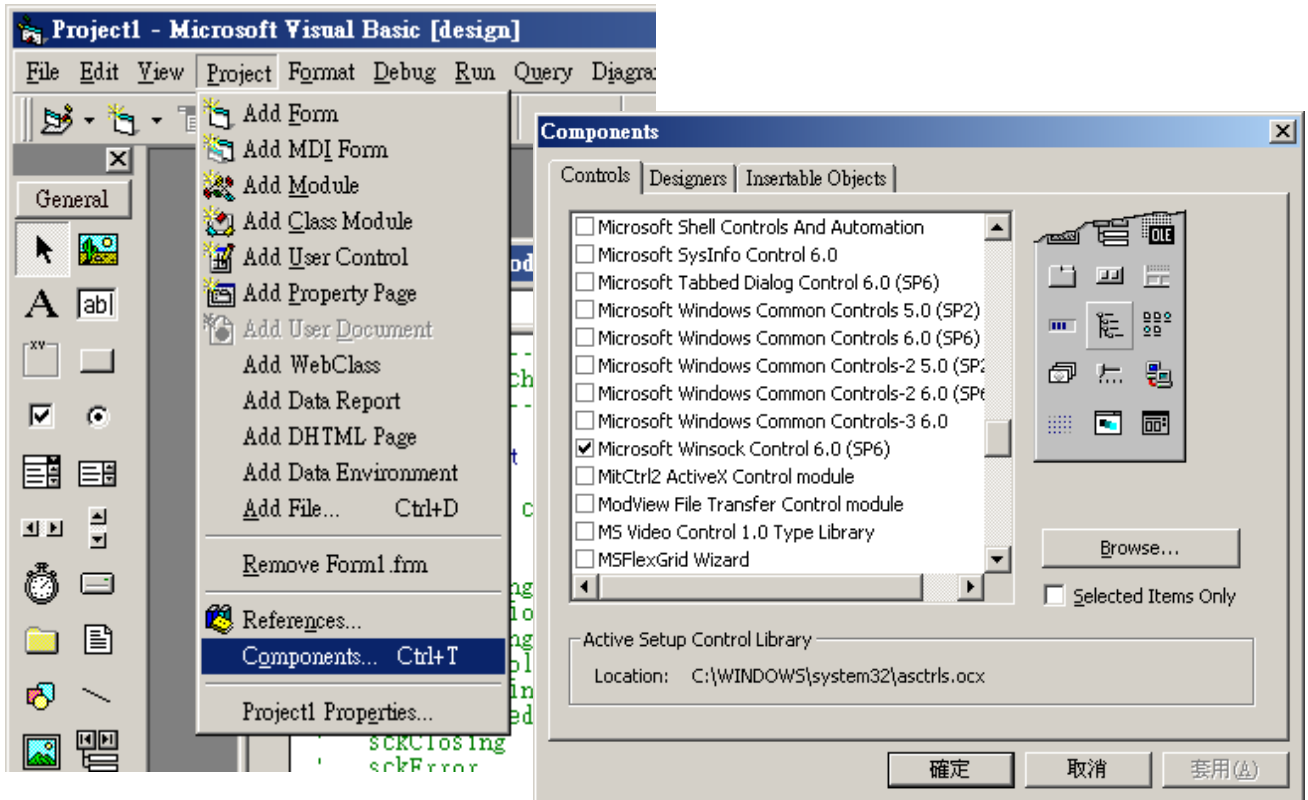
若想要看 VB.net 2005 的 demo 程式，請訪問 <http://www.icpdas.com/faq/isagraf.htm>
 FAQ-051 (或 www.icpdas.com – FAQ – Software – ISaGRAF – 051)

若 PC 無法連上，最下方的“Communication state”會顯示有錯誤發生，若 I-7018z 沒有連上，會顯示紅色的“I-7018z not on-line”

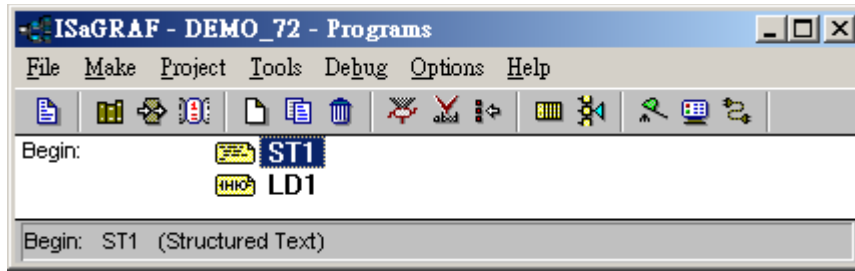
您可試著去按“set Boo_21 to True”那個按鈕，每按一次“long_15”的值會加 1。您也可以直接輸入值給“set long_15”，輸入完後要按下“set long_15”



本 VB 6.0 範例程式在設計時，需 加入”Winsock control”，才能使用 Ethernet 通訊，如下



ISaGRAF 專案架構:



變數定義：

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 True
OK1	Boolean	Internal	I-7018z 的連線狀態, addr 設為 31 (16 進位值為 1F)
M1	Boolean	Internal	測試 VB 6.0 連線用, addr 設為 21 (16 進位值為 15)
TMP	Boolean	Internal	暫時使用的變數
Ain[0..5]	Integer	Internal	為變數陣列, Dim 欄位設為 6, addr 設為 1 用來取得 I-7018z 的 Ch.1 到 Ch.6 之 類比輸入值
Temp[0..3]	Integer	Internal	為變數陣列, Dim 欄位設為 4, addr 設為 7 用來取得 I-7018z 的 Ch.7 到 Ch.10 之 溫度值
CNT1	Integer	Internal	測試 VB 6.0 連線用, addr 設為 15 (16 進位值為 F)
Float_17	Integer	REAL	測試 VB 6.0 連線用, addr 設為 17 (16 進位值為 11) 初值設為 1.02345

ST 程式 – ST1

if INIT then

INIT := False ;

(* 設定 Ain[0..5]的 addr 為 1, 2, 3, 4, 5, 6, 需先在 變數內宣告 Ain[] 的 addr = 1 *)

TMP := S_MB_ADR(1, 6, 0) ; (* 第 3 個參數 0 表示設定成連續的 addr *)

(* 設定 Temp[0..3]的 addr 為 7, 9, 11, 13, 需先在 變數內宣告 Temp[] 的 addr = 7 *)

TMP := S_MB_ADR(7, 4, 1) ; (* 第 3 個參數 1 表示設定成 跳 1 號的 addr *)

end_if ;

if M1 then

M1 := False ;

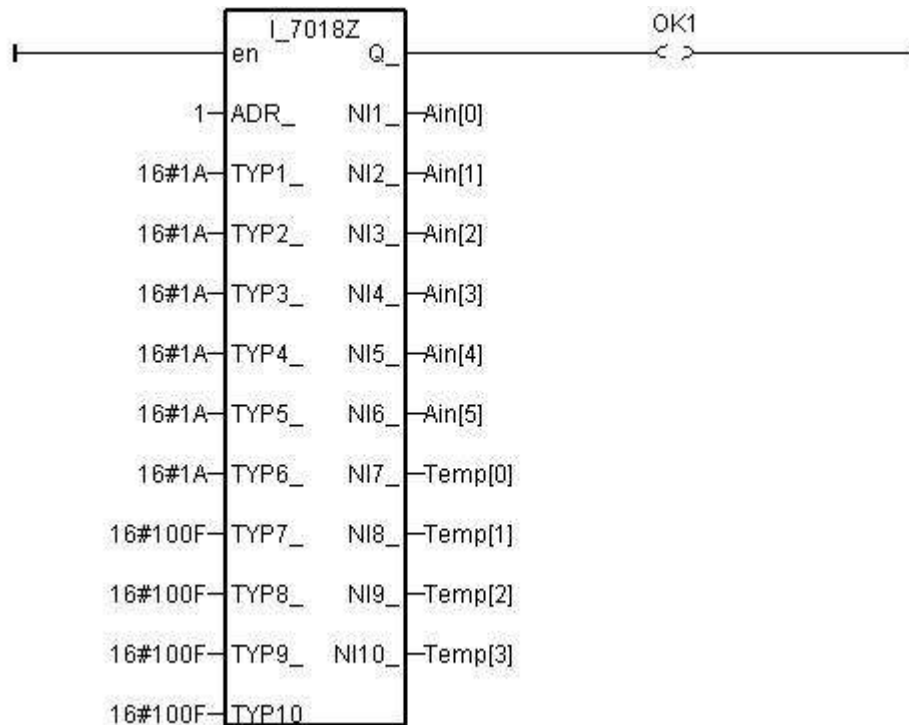
CNT1 := CNT1 + 1 ; (* 當 M1 被設為 TRUE 時, 將 CNT1 值加 1 *)

end_if ;

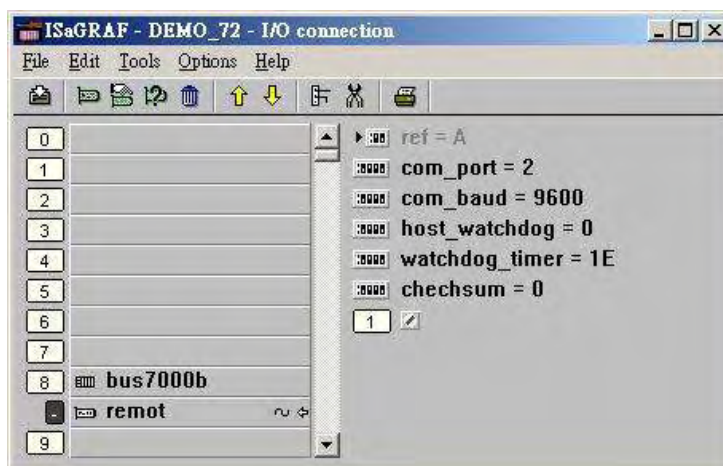
LD 程式 – LD1

TYP1_ 到 TYP6_ 需設成跟 I-7018z 用 DCON utility 規劃的 Type code 一樣 (本例使用 [1A] 0 ~ 20 mA). TYP7_ 到 TYP10_ 需設成跟 I-7018z 用 DCON utility 規劃的 Type code 一樣 (本例使用 [0F] T/C K-Type), 因為要直接轉換成 單位為 0.01 度的 攝式溫度, 所以使用 16#100F. (若想轉成 華氏度, 則需設成 16#200F). 右方的 Temp[0] 到 Temp[3], 若傳回值等於 999990, 表示溫度輸入 Sensor 斷線了.

當 I-7018z 連線正常, OK1 會傳回 TRUE, 若 連線失敗, OK1 會傳回 FALSE.



IO 連結:



11.3.10: 範例 Whmi_13: 使用 Wincon-8xx7, 每 0.01 秒記錄 I-8017H 的 1~4 個電壓值於 User 配置的一塊 RAM 記憶區內, 連續記錄 1~10 分鐘, 之後再儲存到 RAM Disk File 內, 最後可在 PC 上開 M.S. Excel 來看 1~4 條趨勢圖

注意: 若 PAC 是 WP-8xx7, XP-8xx7-CE6 與 VP-25W7 / VP-23W7, 本節所述的功能也可以套用, 但差別是需使用各別 PAC 的 CD 的內容 與 file (內容類似, 但不同於 Wincon CD). 另外 WP-8xx7, XP-8xx7-CE6 與 VP-25W7 / VP-23W7 等 PAC 並沒有 \CompactFlash\Temp\HTTP\WebHMI\, WP-8xx7 與 VP-25W7 / VP-23W7 是使用 \Micro_SD\Temp\HTTP\WebHMI\, 而 XP-8xx7-CE6 是使用 \System_Disk\Temp\HTTP\WebHMI\. 詳細內容請參考 以下網址:

WP-8xx7: “setup_web_hmi_demo.pdf” 放於
ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/wp_webhmi_demo 內.

VP-25W7 / VP-23W7: “setup_web_hmi_demo.pdf” 放於
<ftp://ftp.icpdas.com/pub/cd/vp-25w7-23w7/napdos/isagraf/vp-25w7-23w7/vp-webhmi-demo/> 內

XP-8xx7-CE6: “xpce6_setup_web_hmi_demo.pdf”
<ftp://ftp.icpdas.com/pub/cd/xp-8xx7-ce6/napdos/isagraf/xp-8xx7-ce6/xpce6-webhmi-demo/> 內

本範例程式為 Whmi_13.pia 只能在 Wincon-8xx7/8xx6 的 驅動 3.36 版或更高的版本使用, 放於

新的驅動: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>
範例程式 放於 W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

本範例 PC 的人機介面程式 可選擇用 VB6 的 “Demo_5” 放於

W-8xx7 CD-ROM:\napdos\isagraf\wincon\vb6_demo_pc\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/vb6_demo_pc/

或使用 Web HMI 來當人機介面程式, 原始碼放於 (關於 Web HMI 請參考 Wincon ISaGRAF Getting Started 第 3, 4, 5 章 : W-8xx7 CD-ROM:\napdos\isagraf\wincon\english_manu\ “getting_started_w8337.pdf”)

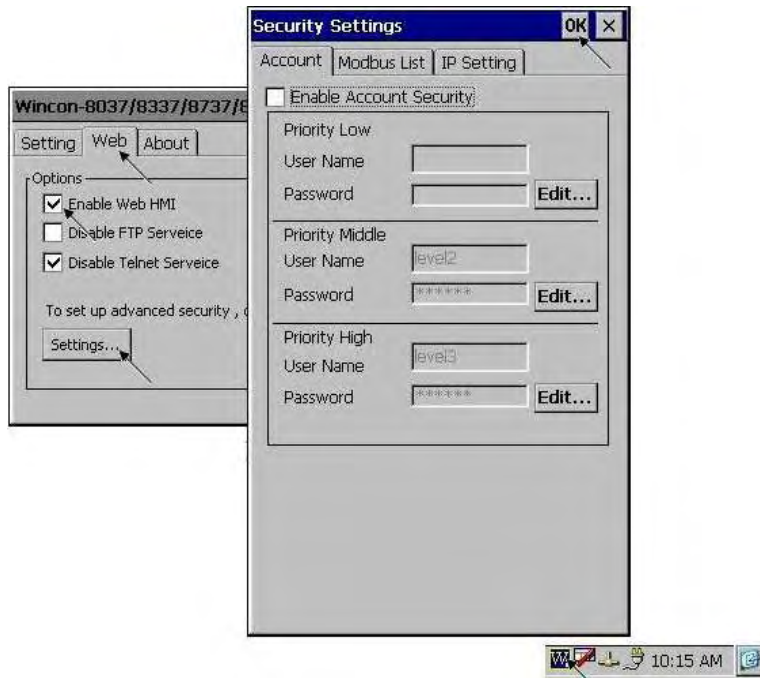
W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI_Demo\ 內的 whim_13 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/

若您 PC 上裝的 ISaGRAF 找不到 Msg_F, Msg_N, ARY_F_R, AFY_F_W 等函式, 請訪問 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 來下載 “ICP DAS utilities For ISaGRAF”, 解壓縮後, 執行其內的 setup.exe 來安裝新的 函式進去 ISaGRAF 內

如何測試本範例:

以下步驟為使用 Web HMI 當人機介面程式的方式, 若您是使用 VB6 的 “Demo_5” 來當人機介面, 只需把 “Demo_5” 的 VB6 程式在 PC 上 Run 起來, 之後只做 步驟 1 (不必啟用 Web HMI), 步驟 3 與 步驟 6.

1. 首先請把 I-8024 插在 W-8xx7 的 Slot 2, I-8017H 插在 Slot 3, 把 I-8024 的第 1 到 4 個電壓輸出點 連接到 I-8017H 的第 1 到 4 個電壓輸入點. 之後上電給 Wincon, 然後在 ISaGRAF Driver 上啟用 Web HMI 功能, 為方便操作, 請先不要選取 “Enable Account Security”



2. 接下來請把 Web HMI Demo 13 的所有 file 用 ftp (例如在 IE 上輸入 “ftp://10.0.0.103”), 丟到 Wincon 的 \CompactFlash\Temp\HTTP\WebHMI\ 內

Web HMI 的原始碼放於

W-8xx7 CD-ROM:\napdos\isagraf\wincon\WebHMI_Demo\ 內的 whim_13 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/webhmi_demo/

共有 7 個 File 與 2 個 目錄要 copy 到 Wincon 的 \CompactFlash\Temp\HTTP\WebHMI\ 內:

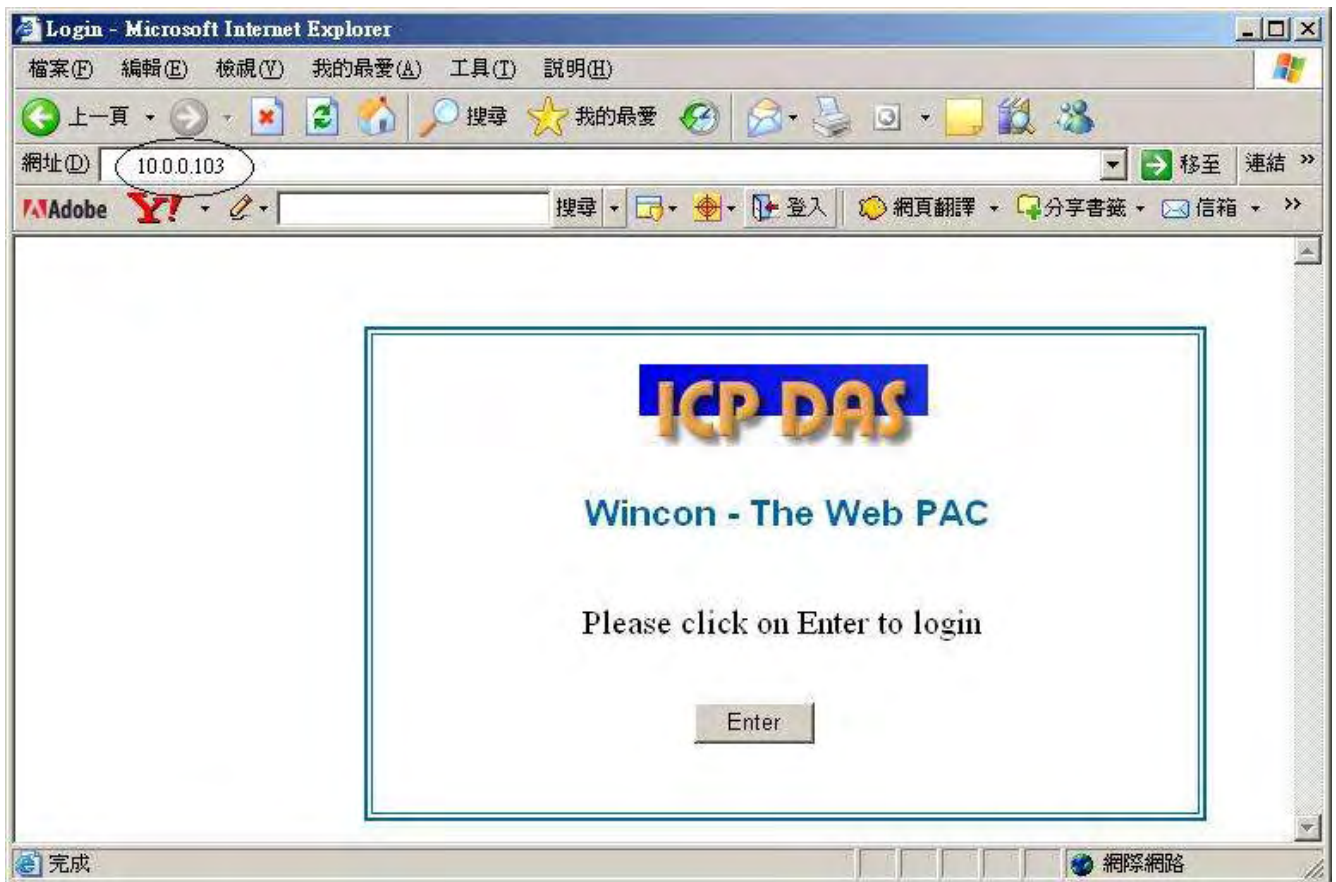
Main.htm , menu.htm , index.htm , login.htm , main.dll , login.dll , whim_filter.dll
img 目錄 與 msg 目錄

3. 之後請把 ISaGRAF 的 whim_13 程式 下載到 W-8xx7 內. (若是使用 Web HMI, 請做完步驟 2, 再做步驟 3)

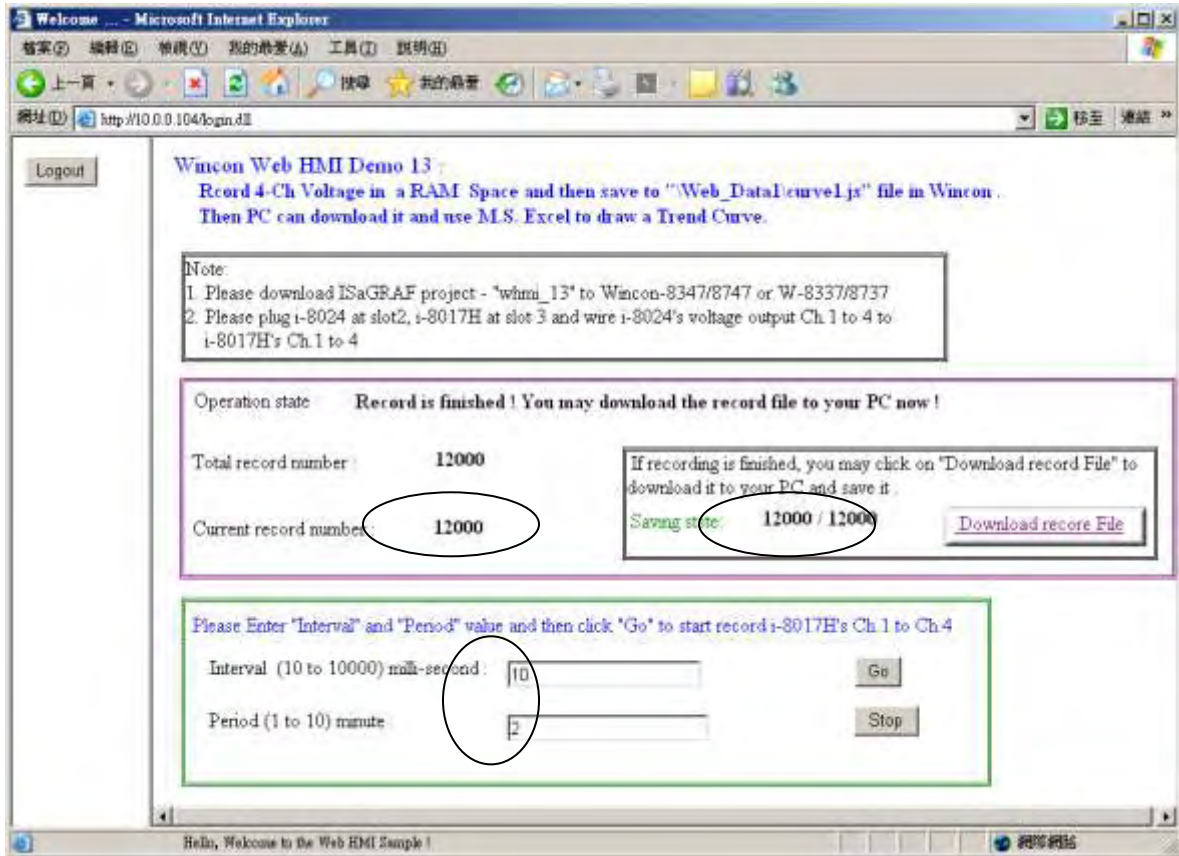
本範例程式為 Whmi_13.pia 只能在 Wincon-8xx7/8xx6 內使用, 放於

W-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

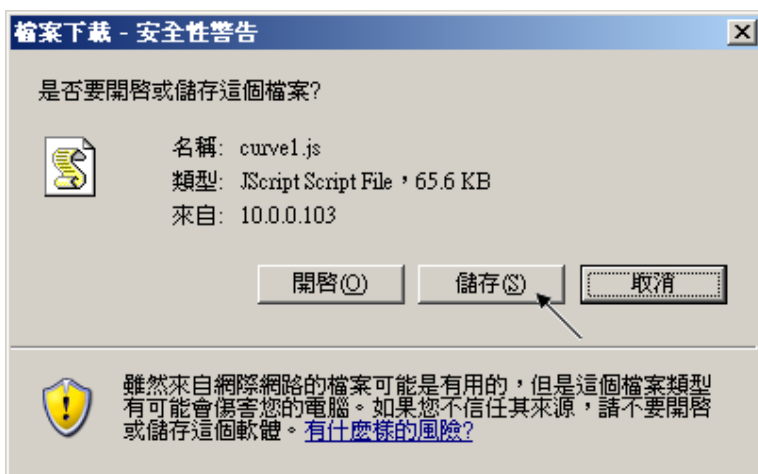
4. PC 上開啟 Internet Explorer (至少需為 IE 5.0 版, 或更高的版本), 上方輸入 W-8xx7 的 IP, 連上 Wincon 後, 請按 “Enter” 進入



5. 之後請輸入 “Interval” 的值，單位為 0.001 秒，如給 10 指的是 每 0.01 秒，記錄一次。而 “Period” 值指的是 要記錄多久，單位為分鐘。輸入好後，按下 “Go” 可開始動作。之後 I-8024 的第 1 到 4 點會輸出不同的電壓，若有在 步驟 1 內將 I-8024 的第 1 到 4 個電壓輸出點 連接到 I-8017H 的第 1 到 4 個電壓輸入點，則 I-8017H 的輸入會有不同的曲線變化會被記錄下來。

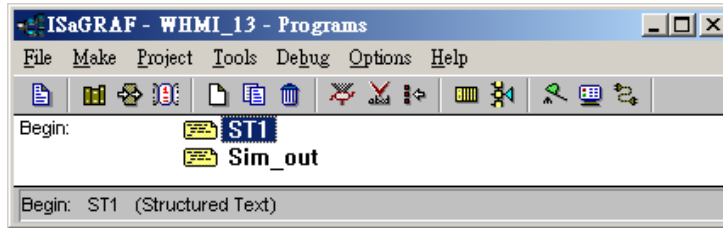


在記錄過程中，會看到 “Current record number” 的值會一直增加，當它等於 “Total record number” 之值時，就表示記錄完畢。此時 程式會開始把記錄資料存入 RAM File 內，從 “Saving state” 可以看到進度，若都完成，請按下 “Download record File”，點選 “儲存” 來下載到 PC 內。



6. 接下來 用 M.S. Excel 來畫趨勢圖，方法與第 11.3.6 節的步驟 6 相同。

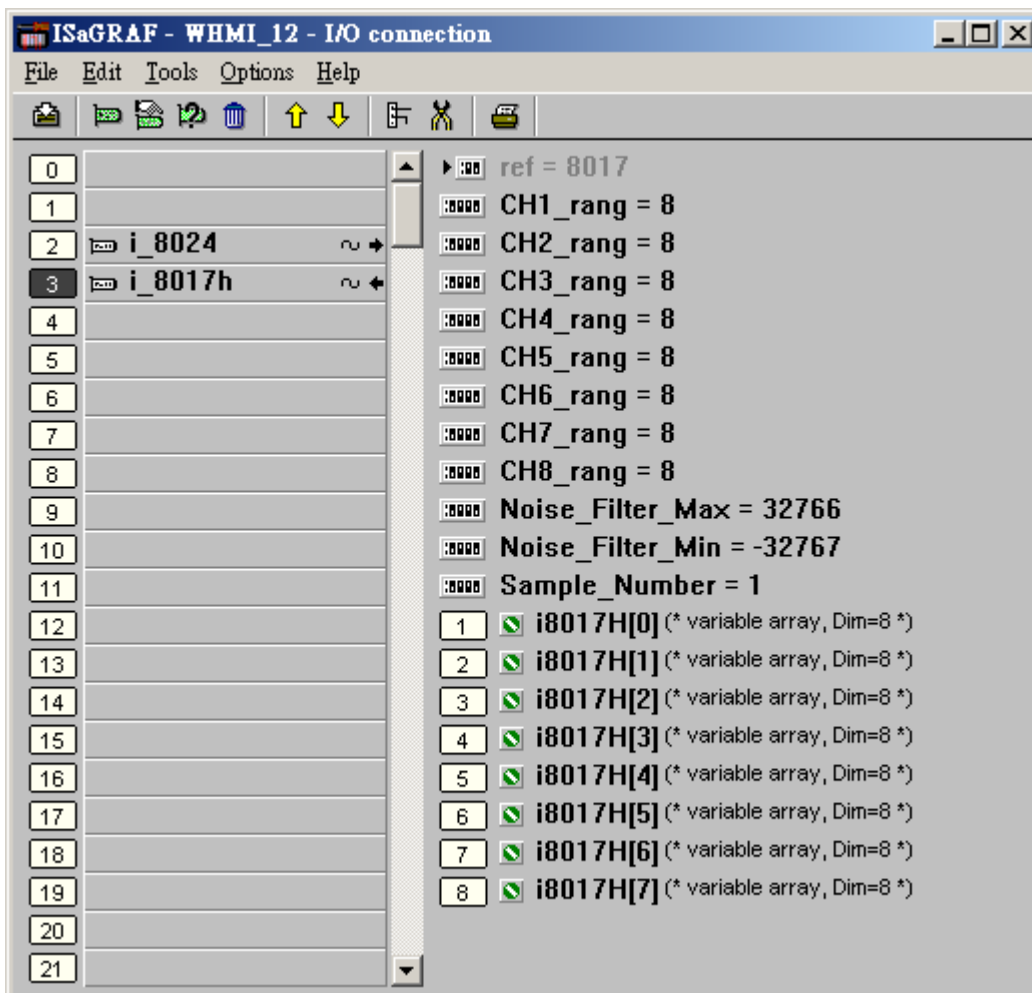
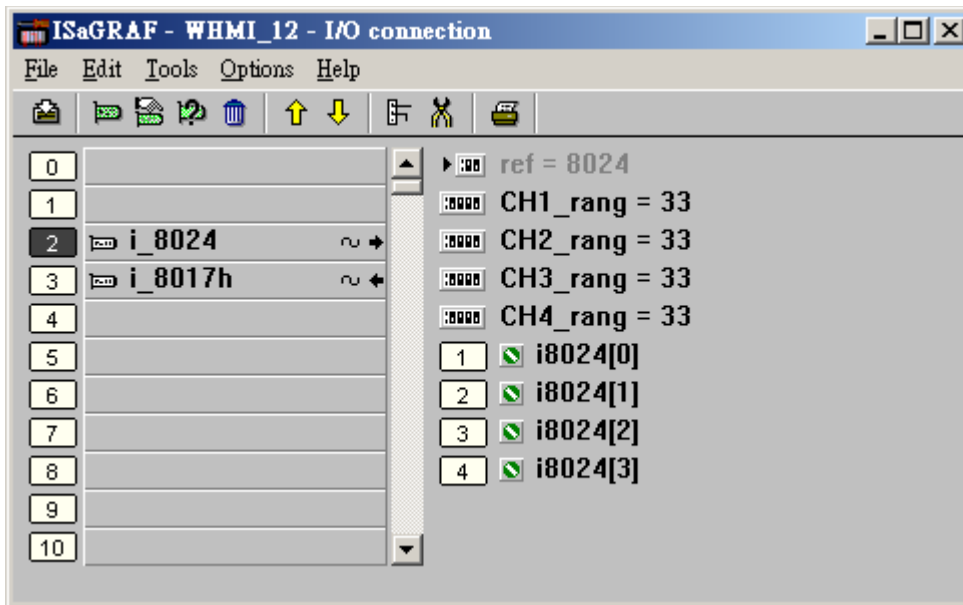
專案架構:



變數定義：

Name	Type	Attribute	Description
Go1	Boolean	Internal	若設為 True 則開始動作, addr 設為 21 (16 進位值為 15)
Stop1	Boolean	Internal	若設為 True 則停止動作, addr 設為 22 (16 進位值為 16)
TMP	Boolean	Internal	暫時使用的變數
INIT	Boolean	Internal	初值設為 True
Save_file1	Boolean	Internal	程式會自動設為 True 來儲存 RAM Disk File
MUM_CH	Integer	Constant 常數	I-8017H 上要記錄幾個 Channel 的電壓值 本範例 記錄 4 個 Channel (Ch.1 到 4)
File1	Integer	Internal	File ID
STEP1	Integer	Internal	0:無動作 , 1:記錄中 , 2:記錄完畢
Period1	Integer	Internal	共要記錄多久, 單位為分鐘, addr 設為 3
Interval1	Integer	Internal	多久記錄一筆資料, 單位為 0.001 秒, addr 設為 1
Total_record1	Integer	Internal	本次共有多少筆資料要記錄, 由程式根據 Period1 與 Interval1 之值 自動算出, addr 設為 5
Record_cnt1	Integer	Internal	目前已經完成記錄的資料數量, addr 設為 7
ii 與 ii2	Integer	Internal	給 for 迴圈使用
i8017H[0..7]	Integer	Input	為 變數陣列, Dim 設為 8, 需連接到 I-8017H 的 Ch1. 到 8
Volt1[0..7]	REAL	Internal	為 變數陣列, Dim 設為 8, I-8017H 的 Ch1. 到 8 轉換後的電壓值
i8024[0..3]	Integer	Output	為 變數陣列, Dim 設為 4, 需連接到 I-8024 Ch1. 到 4
Save_cnt1	Integer	Internal	目前儲存於 RAM disk File 內的 Record 數量, addr 設為 9
TMP_v	Integer	Internal	程式內部使用
T1	Timer	Internal	作為計時的 Timer
T1_next	Timer	Internal	下一筆資料將於何時去記錄下來, 程式自動算出
T1_Interval	Timer	Internal	每 2 筆記錄的時間 間隔
File_name1	Message	Internal	File name, 長度 64, 初值為 \Web_Data1\curve1.js Web HMI 只支持 RAM Disk File 於 \Web_Data1\ 內, 若是 CompactFlash File 則 Web HMI 只支持於 \CompactFlash\Temp\HTTP\Data\ 內 (參見 11.2 節的 Whmi_08 範例)
Msg1	Message	Internal	操作狀態, 長度 255, 初值為 “無動作”, addr 設為 41 (16 進位值為 29)
Str1	Message	Internal	長度 255, 程式內部使用

I/O 連結:



ST 程式 – Sim_out

(* 讓 I-8024 的 Ch1 到 Ch4 輸出不同的 Sin , Cos 電壓曲線 *)

(* 2 * Pi * T1 / 60000 = T1 * 1.047197E-4 *)
(* 2 * Pi * T1 / 120000 = T1 * 5.235985E-5 *)

```
i8024[0] := ANA( sin( REAL(T1) * 1.047197E-4 ) * 3276.8 ) ;  
i8024[1] := ANA( cos( REAL(T1) * 5.235985E-5 ) * 3276.8 ) ;  
i8024[2] := ANA( sin( REAL(T1) * 1.047197E-4 ) * 6553.6 ) ;  
i8024[3] := ANA( cos( REAL(T1) * 5.235985E-5 ) * 6553.6 ) ;
```

ST 程式 – ST1

(* 以下這區只在 第 1 個 PLC scan 內 Run 一次 *)

if INIT then

INIT := False ;

(* 配置 1 塊可存放 500,000 個整數 (或實數) 的 RAM 記憶區 , W-8xx7 最多只能配置 3,000,000 個 整數/實數, 一個 整數/實數 為 4 個 bytes, 所以最多是 12,000,000 bytes *)

(* ARcreate() 只能在 ISaGRAF 程式內使用一次, 且第 1 個參數只可以是 1 *)

TMP_v := ARcreate(1 , 500000) ;

if TMP_v <> 1 then (* 回傳值若不是 1 表示失敗 *)

Msg1 := 'Parameter error or Can not allocate enough memory by ARcreate()

function !' ;

end_if ;

(* 設定 W-8xx7 以最高速度執行, 此動作會使得同時在 Wincon 上 Run 的非 ISaGRAF 程式變慢 *)

TMP := PLC_mode(-1) ;

end_if ;

(* 若 Stop1 被設為 TRUE, 則停止 T1 的計時, STEP1 設為 0 *)

if Stop1 then

Stop1 := False ;

STEP1 := 0 ; (* STEP1 設為 0:無動作 *)

TStop(T1) ; (*停止 T1 的計時 *)

T1 := T#0s ;

Msg1 := 'User stop recording !' ;

end_if ;

(* 若 Go1 被設為 TRUE, 準備開始動作 *)

if Go1 then

Go1 := False ;

(* STEP1 : 0:無動作 , 1:記錄中 , 2: 記錄完畢 *)

if STEP1 = 1 then

Msg1 := 'It is still recording now ... Please wait' ; (* 更新狀態顯示為“仍在記錄中 ...” *)

else

(* 檢查 Interval1 之值 設的 對不對, 需為 10 到 10000 之間, 單位為 0.001 秒 *)

If (Interval1 < 10) or (Interval1 > 10000) then

Msg1 := 'Wrong Interval value, it should be in 10 to 10000 milli-second !' ;

(* 檢查 Period1 之值 設的 對不對, 需為 1 到 10 之間, 單位為 分鐘 *)

elsif (Period1 < 1) or (Period1 > 10) then

Msg1 := 'Wrong Period value, it should be in 1 to 10 minute !' ;

else

(* 值設的正確 , 接下來 準備開始記錄資料 *)

total_record1 := (Period1 * 60000) / Interval1 ; (* 算出 total_record1 值 *)

record_cnt1 := 0 ; (* 將目前已記錄資料數量設為 0 *)

STEP1 := 1 ; (* 將 STEP1 設為 1:記錄中 *)

Msg1 := 'Recording now ... Please wait' ;

(* 啟動 T1 開始從 0 起計時 *)

T1 := T#0s ;

T1_Interval := TMR(Interval1) ;

T1_next := T1 + T1_Interval ;

TStart(T1) ;

end_if ;

end_if ;

end_if ;

if STEP1 = 1 then (* 1:記錄中 *)

if T1 >= T1_next then (* 當 T1 計時到下 1 筆該記錄的時間點時, 就準備記錄 1 筆資料 *)

T1_next := T1_next + T1_Interval ; (* 重新設定 下 1 筆該記錄的時間點 *)

(* 因為 Timer 最多只能計時到 23 小時 59 分 59 秒, 所以提前在 剛過 20 小時之後, 就 Reset 為 0 *)

(* 本範例最多只有計時 10 分鐘, 所以不會超過, 以下這段的用意, 在於給使用者參考用 *)

(* ----- *)

if T1 >= T#20h then

T1 := T#0s ;

T1_next := T1 + T1_Interval ;

end_if ;

(* ----- *)

(* 準備寫入一筆資料於 User 配置的 RAM 記憶區內 *)

for ii := 0 to NUM_CH - 1 do

(* 轉換 i8017H 的類比輸入值為 電壓值 *)

Volt1[ii] := Real(i8017H[ii]) * 0.000305176 ; (* 10.0 / 32768 = 0.000305176 *)


```
(* 先使用 Real_int() 將實數 Map 成整數, 再存入 User 配置的 RAM 記憶區內 *)  
TMP_v := ARwrite( 1 , NUM_CH * record_cnt1 + ii , Real_int( Volt1[ii] ) );
```

```
(* 檢查是否寫入成功 *)
```

```
if TMP_v <> 1 then (* 回傳值若不是 1 表示失敗, 有可能是配置的記憶區不夠大 *)
```

```
    Msg1 := 'Can not operate ARwrite() !' ;
```

```
    STEP1 := 0 ; (* Step 設為 0: 無動作 *)
```

```
    TStop( T1 ) ; (* 停止 T1 的計時 *)
```

```
    T1 := T#0s ;
```

```
end_if ;
```

```
end_for ;
```

```
record_cnt1 := record_cnt1 + 1 ; (* 已記錄的資料數量 加 1 *)
```

```
if ( record_cnt1 >= total_record1 ) then (* 檢查是否全部資料都已記錄完畢 *)
```

```
(*全部資料都已記錄完畢, 準備開始把資料從 User 配置的 RAM 記憶區, 存入 RAM  
Disk File *)
```

```
STEP1 := 0 ; (* Save 之前, 先暫時將 Step 設為 0: 無動作 *)
```

```
Tstop( T1 ) ; (* 停止 T1 的計時 *)
```

```
T1 := T#0s ;
```

```
(* 建立一個新的 RAM Disk File, File_name1 於 \Web_Data1\ 內 *)
```

```
File1 := F_creat( File_name1 ) ;
```

```
if File1 = 0 then (* 建立 File 失敗 *)
```

```
    Msg1 := 'Create File ' + 'File_nam1 Error !!!' ;
```

```
else
```

```
    (* 將 save_file1 設為 TRUE 來開始分散於多個 PLC scan 內把資料存成 File *)
```

```
    Msg1 := ' Please wait ... Saving data to file : ' + File_name1 + ' ... ' ;
```

```
    save_file1 := True ;
```

```
    save_cnt1 := 0 ; (* 先將目前已存入記錄數量設為 0 *)
```

```
end_if ;
```

```
end_if ;
```

```
end_if ;
```

```
end_if;
```

```
(* 以下 每個 PLC scan 只存入 50 筆記錄, 請不要在 1 個 PLC scan 內就把所有記錄存完,  
因為 存資料到 File 會花較多的 CPU 時間, 若強迫在 1 個 PLC scan 內存完, 該 PLC scan 的  
Scan time 會變的很大, 會影響到其它的 I/O 邏輯操作*)
```

```
if save_file1 then
```

```
    for ii2 := 0 to 50 do (* 每個 PLC scan 只存入 50 筆記錄 *)
```

```

if save_cnt1 < total_record1 then      (* 若還未存完, 繼續存 *)

    str1 := '' ;      (* 先將 str1 設為空字串 *)
    for ii := 0 to NUM_CH - 1 do (* 將實數資料轉成字串, 分隔字元為 <TAB> *)
        str1 := str1 + Rea_Str2( Int_real( ARread(1, NUM_CH * save_cnt1 + ii) ), 3 ) + '$09' ;
    end_for ;
    str1 := str1 + '$0D$0A' ;      (* 加入 <CR><LF> 字元於每行尾端 *)
    TMP := F_writ_s( File1, str1 ) ;      (* 寫入 字串 *)
    save_cnt1 := save_cnt1 + 1 ;      (* 目前已存入記錄數量加 1 *)

else

    (* 所有資料都已存完 *)
    save_file1 := False ;      (* 設為 False 表示不再存了 *)
    TMP := F_close( File1 ) ;      (* 關閉 該 File *)
    STEP1 := 2 ;      (* 設為 2: 記錄完畢 *)
    Msg1 := 'Record is finished ! You may download the record file to your PC now !' ;

    end_if ;

end_for ;

end_if ;

```

11.4: 常用的 FAQ

FAQ (ISaGRAF Ver.3 FAQ 中文: 問題 / 說明 / 範例程式) 網頁指引

http://www.icpdas.com/faq/isagraf_c.htm

www.icpdas.com > Support > FAQ > Software > ISaGRAF Ver.3 (中文)

FAQ 問題列表:

編號	中文 ISaGRAF Ver.3 FAQ
5	Q: 如果我的 ISaGRAF 軟件不是 ISaGRAF-256 或 ISaGRAF-L, 可以使用超過 32 個 I/O 點嗎 ?
16	Q:若 D/I 量測點在切換 OFF 到 ON, 或由 ON 到 OFF 過程中有短暫的訊號跳動現象, 該如何來編程 ?
18	Q: ISaGRAF-256 軟體是否有 I/O 點數的使用限制 ?
20	Q:如何在 ISaGRAF 內搜尋某一變數名稱 ?
21	Q:有時在關閉 ISaGRAF 視窗時會卡住約 20 ~ 40 秒, 如何解決 ?
22	Q:如何使用 Proface 的人機介面 (Touch-Panel HMI) 來連接 I-7188EG/XG , I-8xx7 與 WinCon-8xx7 等 ISaGRAF 控制器 ?
24	Q:如何將 Analog / Input 或 Analog / Output 的 4 到 20 mA 值, 轉換成 工程應用值 ? 如何將 Analog / Input 或 Analog / Output 的 0 到 10 V 值, 轉換成 工程應用值 ?
30	Q:如何直接取得第 0 ~ 7 的 I/O 插槽內溫度卡片的“攝氏度”或“華式度”值 ?
31	Q:如何直接取得遠程 I-7000 及 I-87xxx 溫度輸入 I/O 的溫度值 ?
38	Q:如何 啟用 I-7188EG 或 I-7188XG 的 COM3 成為 Modbus RTU slave 埠 ?
39	Q:如何使用變數陣列 ?
40	Q:如何啟用 I-8437 / 8837 的 COM3:RS232/485 來成為 Modbus RTU Slave 串口 ? 如何改變 COM1 與 I-8417 / 8817 的 COM2 或 I-8437/8837 的 COM3 Modbus RTU port 通訊速率之設定 ?
41	Q:如何將 圖控/HMI/上位機 連接到 單一 IP 地址的備援(冗餘) 系統 ?
42	Q:如何使用 WinCon-8347 / 8747 來連接 I-8KE4-MTCP 與 I-8KE8-MTCP 這 2 款 Ethernet I/O ?
46	Q:如何使用 Modbus 函式 6 來寫 16 個 Bit 到 Modbus RTU 設備 ?
47	Q:如何對 Modbus RTU Slave 設備讀/寫實數資料 ?
48	Q:WinCon 如何連接 FRnet I/O ?
49	Q:如何使用“Mbus_R”與“Mbus_R1”方塊,來直接取得 M-7000 溫度模塊的攝氏或華氏溫度值 ?

編號	中文 ISaGRAF Ver.3 FAQ
50	Q:如何連接 M-7000 系列 Modbus RTU I/O ?
51	Q:如何使用 PC 上的 VB.net 2005 程式, 用 Modbus TCP/IP 規約來與 ISaGRAF 控制器連接 ?
52	Q:如何使用 PC 上的 VB 6.0 程式, 用 Modbus TCP/IP 規約與 ISaGRAF 控制器連接 ?
53	Q:不同型號的 ISaGRAF 控制器性能比較表
55	Q:如何連接 I-7018z 量測 6 個 4 到 20 mA 的電流輸入值 與 4 個 Thermo-Couple 溫度輸入值, 並顯示在 PC 的 VB 6.0 程式上 ?
56	Q:如何 在 ISaGRAF 控制器內 使用較準確的 固定週期 去 做某件事 ?
57	Q:如何使用 WinCon-8xx7, 每 0.01 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 User 配置的一塊 RAM 記憶區內, 連續記錄 1~10 分鐘, 之後再儲存到 RAM Disk File 內, 最後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖 ?
58	Q:如何使用 I-8437-80 或 I-8837-80, 每 0.05 秒記錄 I-8017H 的 1 ~ 4 個電壓值於 S256 / S512 內的一個檔案, 連續記錄 1~10 分鐘, 之後可在 PC 上開 M.S. Excel 來看 1 ~ 4 條趨勢圖 ?
59	Q:常用的 COM Port 函式技巧 來讀寫 RS-232 / 422 / 485
60	Q: 如何在 WinCon 內 讀 / 寫 File 資料 ?
61	Q: 如何連接 RS-485 遠程 I-7000 與 I-87xxx I/O 模塊到 I-8xx7, I-7188EG/XG 與 WinCon-8xx7 控制器 ? 如何設計 RS-485 遠程 I-7017RC, I-87017RC, I-7018Z I/O 模塊的程序 ?
62	Q: 如何規劃 備援系統 (冗餘系統)? 採用 Ethernet I/O
63	Q: 為何我的 RS-485 遠程 I-7000 與 I-87K Output 模塊 在 RS-485 通訊線斷線後 沒有啟動 wacthdog 讓 Output 點跑到 安全輸出值 ?
65	Q: 泓格科技 提供 便宜穩定 且 有效率的 主動 訊號採集 與 資料回報 系統 (提供 VC++ 6.0 與 VB 6.0 與 ISaGRAF demo 程式).
66	Q: 如何使用 RS-232 / RS-485 通訊串口來接收設備主動傳過來的數值資料 ? 比如像是 條碼機 或是 RS-232 磅秤.
67	Q: 如何使用 WinCon-8xx7 或 μ PAC-7188EG 或 iPAC-8447/8847 來發送可夾帶附件的 Email ?
68	Q: 為何 W-8xx7 或 I-8xx7 或 I-7188EG 或 I-7188XG 發生一直重啟(Reset)的現象?
69	Q: 為何我的 PC 無法用 ftp 命令來連上 W-8347 或 W-8747 ?
70	Q: 如何 自動對多台 ISaGRAF PAC 進行時間校正 與 記錄運作狀態 ?
71	Q: 應用例: 用 W-8xx7 來 記錄 10 個溫度 資料, 每分鐘 記錄 一次. 記錄完一天的資料後, 用 Ethernet 發 email 把 該 記錄 好的 file 發出來.
072	Q: 應用例: 使用 WinCon-8xx7 每 20 ms 記錄 一筆 電壓 / 電流 資料, 連續記錄 1 到 10 分鐘, 完成後用 Email 將該 記錄 檔案發出來.

編號	中文 ISaGRAF Ver.3 FAQ
073	Q: 為何 ISaGRAF 程式內用 i_7017 方塊讀到的 I-7017 模塊的輸入值是 正常值的 2 倍大或 變成負值 ?
074	Q: 如何 使用 ISaGRAF New Retain Variable 與 它的優點 ?
075	Q: 為何我的 ISaGRAF 程序無法正確連上其它廠牌的 Modbus Slave 設備 ?
077	Q: 應用例: 使用 μPAC-7186EG 每秒 記錄 一筆 電壓 / 電流 資料, 連續 記錄 1 到 10 分鐘, 完成後用 Email 將該 記錄 檔案發出來.
080	Q: 應用例: 用 μPAC-7186EG 來 記錄 10 個 溫度 資料, 每分鐘 記錄一次. 記錄完一天的 資料後, 用 Ethernet 發 email 把 該 記錄 好的 file 發出來.
081	Q: 如何在 ISaGRAF controller 內使用 I-87017W-A5 來量測 +/- 150VDC ?
082	Q: 一個 FRnet : 快速遠程 I/O 的簡單程式設計方法.
083	Q: 如何設定 I-8xx7, I-8x37-80, I-7188EG 和 μPAC-7186EG 的 TCP 回收時間?
084	Q: 應用例: 經濟且有效 的 可熱插拔 備援 (冗餘)系統 - μPAC-7186EG 或 I-8437-80 + RU-87P4 / 8.
086	Q: 如何使用 WinCon-8347 / 8747 或 μPAC-7186EG 或 iPAC-8447 / 8847 來連接一顆 I-7530 或多顆 I-7530, 來 讀取 或 控制 CAN 及 CANopen 設備 與 傳感器 ?
087	Q: 當七段顯示器出現 Err00, Err02, Err03, Err90 和 E.0001 訊息時代表什麼意思?
088	Q: 功能變更: W-8347/8747, μPAC-7186EG, I-8x37-80, I-8xx7 和 I-7188EG/XG 從新版 driver 起, S256/512 和 X607/608 不再支持舊的可保存變數 (Retain variable) 方式 , 請改用功能較好的 新的可保存變數方式.
089	Q:為何我的 μPAC-7186EG 無法更新驅動程式和 ISaGRAF 應用程式?
090	Q: 如何在 ISaGRAF PAC 內使用 I-7017Z?
091	Q: 如何使用 ISaGRAF PAC 加上 I-87089 卡片搭配 VW sensor , 來量測 水泥建築物的 VW 頻率 與 溫度 去 求得 建築物的 內部應力 ?
092	Q: 如何啟用 μPAC-7186EG, I-7188EG 和 I-7188XG 的 COM2 或 COM3 成為 Modbus RTU Slave 埠?
093	Q: WinCon-8347 / 8747 新的 可熱插拔 備援系統 (冗餘系統).
094	Q: 如何更新 WinCon-8347 / 8747 的 OS 版本 ?
095	Q: WinCon-8xx7 從第 4.03 版 Driver 起支持最多 32 個 Modbus TCP/IP 連線.
096	Q: 新增 Modbus RTU / ASCII 功能方塊來一次讀取最多 24 個 Word 或最多 384 個 Bit.
097	Q: 如何在 WinCon-8x47 沒有接 Mouse 與 VGA 顯示器的情況下, 使用 USB 隨身碟來更改它的 IP 地址 , 站號 與 Modbus RTU 等設定?

編號	中文 ISaGRAF Ver.3 FAQ
098	Q: 應用例: 使用 COM Port 的函式讀寫 Modbus RTU 設備
099	Q: 如何編寫一個每固定時間取樣 (或每個 PLC scan 取樣) 然後求整數或實數平均值的功能方塊 ?
100	Q: 如何使用 I-8084W (4 / 8 – Ch. Counter 或 8-Ch. frequency) ?
101	Q: 如何使用 MBUS_XR 或 MBUS_XR1 方塊來對 Modbus RTU / ASCII 設備, 一次讀取最多 120 個 Word 或 最多 60 個長整數 或 最多 60 個實數 (只適用於 Wincon-8xx7 / 8xx6 與 WP-8xx7 / 8xx6 與 VP-25W7/23W7/25W6/23W6) ?
102	Q: 為何 WP-8xx7 或 VP-25W7/23W7 的 FTP server 連不上 ?
103	Q: 在 WinPAC 上使用 RS-232 或 USB 觸控式螢幕.
104	Q: 為何我的 PC / ISaGRAF 無法正常連上 ISaGRAF PAC ?
105	Q: 如何在 WP-8xx7 , VP-25W7/23W7 與 iPAC-8xx7 內使用 I-8088W 來控制 8 個 PWM 輸出 ?
106	Q: 如何利用 WinPAC-8xx7 和 I-8084W 來量測頻率或轉速輸入並使用 C# .net 2008 將所記錄的值畫成趨勢圖?
107	Q: 如何在 ISaGRAF PAC 內使用 I-87211W (GPS module) 來利用衛星訊號進行自動時間校正與取得目前位置的經緯度座標 ?
108	Q: 如何利用 WinPAC-8xx7 和 I-87018z 來量測溫度並使用 C# .net 2008 將所記錄的值畫成趨勢圖 ?
109	Q: 如何利用 ISaGRAF PAC 和 I-87211w 並透過 Ebus 校正整個區域網路上全部 ISaGRAF PAC 的時間?
110	Q: ZigBee 無線通訊應用 : 如何遠距離控制 I/O 與 進行資料收集 ?
111	Q: 如何利用 GTM-201-RS232 發各別國家文字的簡訊?
112	Q: 如何使用 ISaGRAF 來編程 I-8093W (3 軸高速 Encoder 輸入模塊) ?
113	Q: 在 ISaGRAF PAC 內使用 Modbus TCP/IP Master 來連接 Modbus TCP/IP Slave 裝置.
114	Q: 列印 ISaGRAF PDF 文件時, 如何避免印出亂碼?
115	Q: 開發 eLogger HMI 加上 ISaGRAF SoftLogic 一起應用於 WP-8xx7, VP-2xW7 與 XP-8xx7-CE6 等 PAC 內. (文件版本 1.03,於 2010 年 7 月 15 日發佈)
116	Q: 如何在不修改 ISaGRAF 程式的情況下啟用 WP-8xx7 與 VP-2xW7 的第 2 到第 5 個 Modbus RTU slave 串口 ?
117	Q: 如何在 Windows Vista 或 Windows 7 中安裝 ISaGRAF?
118	Q: 一個 M.S. VC++ 6.0 範例程式跑 Modbus TCP 來連接一台 WP-8xx7.
119	Q: 如何實現中央控制站與地區控制站間的通訊備援機制 ?

編號	中文 ISaGRAF Ver.3 FAQ
120	Q: 如何使用 Aver_N 與 Aver_F 來計算連續平均值 (或稱 移動平均值) ?
121	Q: 如何建立與移除 ISaGRAF 開發環境?
122	Q: 如何解決 W-8x47 發生的 USB-Freeze 問題 ? 如何更新 Wincon-8347 / 8747 的 OS 版本 ?
123	Q: 如何加快 InduSoft + ISaGRAF 的畫面顯示速度 (W-8xx6 / WP-8xx6 / VP-25W6 / XP-8xx6-CE6) ?
124	Q: ISaGRAF 專業版控制器 XP-8xx7-CE6-PRO 的 Web HMI 應用範例 – 使用 FrontPage
125	Q: XP-8xx7-CE6 與 iDCS-8000 (或 ET-7000 或 其它 Modbus TCP Slave 設備) 備援 (冗餘) 系統
126	Q: 利用 WP-8847 連接控制 ET-7018Z 與 ET-7044D , 並使用 InduSoft 或 VS2008 C# .NET 以及 VB .NET 來當 HMI
128	Q: 如何使用 ISaGRAF PAC 加上 i-87113DW 卡片搭配 Carlson Strain Gauge Inputs (差阻式傳感器) ?
129	Q: 如何使用 ISaGRAF PAC 連接 ICP DAS 電力表 – PM-2133 與 PM-2134 ?
130	Q: 如何使用 WP-8x47/VP-23W7 SNTP Client 網路自動校時功能 ?
131	Q: Soft-GRAF: 在 XP-8xx7-CE6 與 WP-8xx7 與 VP-2xW7 等 PAC 內建立多彩多姿的 HMI 畫面 (文件版本: 1.3)
132	Q: 使用 I-8094F/8092F/8094 執行馬達控制運動 .
133	Q: 如何在 ISaGRAF PAC 內使用 UDP 或 TCP 來收取 (或 發送) 廣播資料封包 或 一般資料封包
134	Q: 如何寫 ISaGRAF 程式來 Reset ISaGRAF driver 或 Reset 整台 Controller ?
135	Q: 如何寫 ISaGRAF 程式用 WinCE PAC 去對後台的 MicroSoft SQL Server 讀取或寫出資料 ?
136	Q: HART 應用 : ISaGRAF 可程式自動控制器 (PAC) 加上 I-87H17W
137	Q: 如何使用 μ PAC-5000 系列控制器透過 GPRS 連線至遠端伺服器並且傳送資料?
138	Q:如何設計一個 XP-8xx7-CE6 備援 (冗余) 系統 (採用 I-87K8 或 Modbus I/O 或 其它 I/O) ?
139	Q:如何安裝使用 ISaGRAF 3.55 Demo 版以及有何限制 ?
140	Q:InduSoft PC 端的 HMI 如何利用 Modbus TCP 與 ISaGRAF PAC 通訊? NEW!
141	Q:iP-8xx7/ μ PAC-7186EG/I-8xx7/I-8xx7-80 提供防寫入快閃式記憶體 (Flash memory) 功能

編號	中文 ISaGRAF Ver.3 FAQ
142	Q: 如何保護你的 ISaGRAF 程式, 讓盜用它的人無法使用 ?
143	Q: 如何讓 ISaGRAF WinCE PAC 撥接 GPRS 來連上網路傳資料 ?
144	Q: 新增 Modbus RTU / ASCII / TCP 方塊“Mbus12w” 來一次寫出 1 ~ 12 個 Word .
145	Q: 如何使用 ISaGRAF 操作 I-8123W 來控制 CANopen Slave 設備 ?
146	Q: Soft-GRAF Studio : 在 XP-8xx7-CE6 與 WP-8xx7 與 VP-2xW7 等 PAC 建立多采多姿的 HMI 畫面
147	Q: 範例 4: 如何使用 VPD-130 經由 RS-485 讀取 μ PAC-7186EG 的系統日期/時間 ?
148	Q: I-87017DW、I-87017ZW 與 I-87019ZW 板卡使用說明
149	Q: 如何讓 ISaGRAF WinCE PAC 播放聲音 ?
150	Q: ISaGRAF 線上教學影片

第 12 章發送 Email

WP-8xx7, XP-8xx7-CE6, VP-25W7, VP-23W7, Wincon-8xx7, μ PAC-7186EG 與 iPAC-8447/8847 等 controller 從以下的 ISaGRAF driver 版本起,支持使用它的 Ethernet 串口來發送 Email 功能. 該 controller 所在的區域網路必需是可以上網的網路, 不然無法把 Email 發送出去.

WP-8xx7 / WP-5xx7	從開始銷售之日起
XP-8xx7-CE6 :	從開始銷售之日起
VP-25W7 / VP-23W7:	從開始銷售之日起
Wincon-8xx7 / 8xx6 :	3.42 版起
iPAC-8447/8847:	從開始銷售之日起
μ PAC-7186EG:	從開始銷售之日起

若想使用 GPRS 來連上 Internet 發 mail 或 發 TCP, UDP 資料出去, 請參考
http://www.icpdas.com/faq/isagraf_c.htm > FAQ-143

新發布的 ISaGRAF driver: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

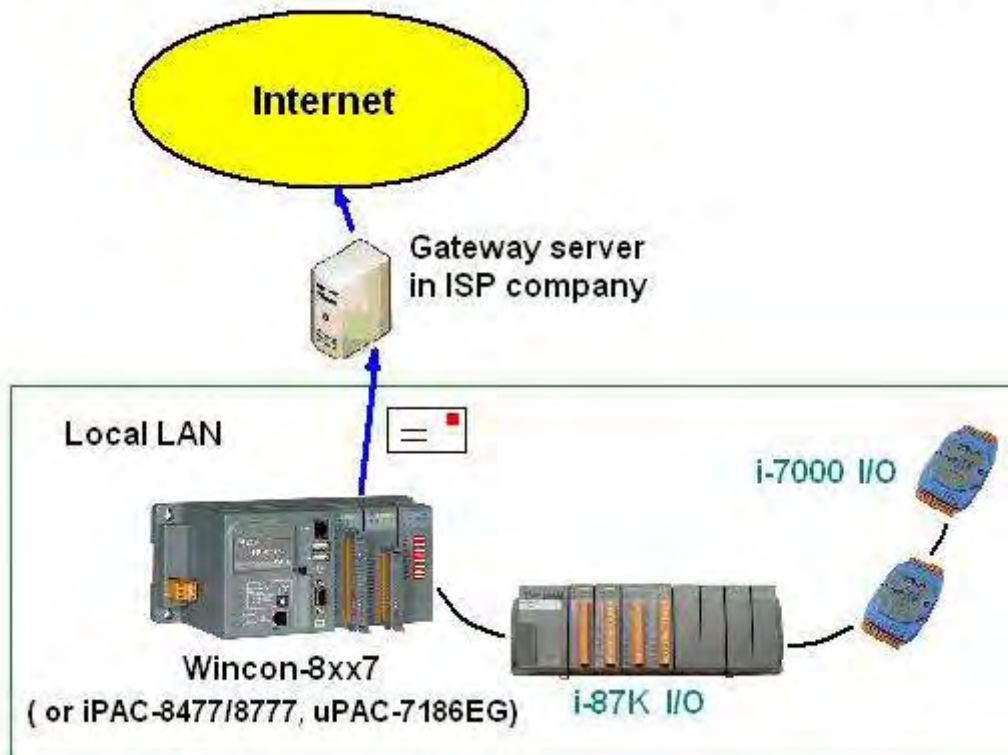
Demo program: www.icpdas.com – FAQ – Software – ISaGRAF – 067 , 071, 072, 076,077, 080
或

Wdemo_62 , Wdemo_63 , Wdmo_63a , Wdmo_65a, Wdmo_65b, demo_74a , demo_75a

或

http://www.icpdas.com/faq/isagraf_c.htm

Controller can send email without or with one attach file



基本功能如下:

1. 發送的 Email 可以夾帶一個附件 或 不夾帶附件. 附件 file 格式可以是 text , binary 或其它任何格式. 夾帶的附件最大可以如下.

WinPAC-8xx7/VP-25W7/XP-8xx7-CE6:	2M Bytes.
Wincon-8xx7:	2M Bytes.
iPAC-8447 / 8847:	488K Bytes
μPAC-7186EG + X607:	112K Bytes
μPAC-7186EG + X608-RoHs:	488K Bytes

2. Email 的主旨最長可到 128 個 byte, Email 內容最長可到 510 個 byte. 這 2 個資料可以使用各自國家的語言文字 .
3. 每封 Email 一次可以同時最多發給 10 個收件人, 若要發給更多人, 需再重覆發送多次.
4. 每封要發送出去的 Email 可以設定 Email 優先權 為 High , Low 或 Normal.
5. Mail server IP 可以只指定一台 (Mail server 1),或為了安全需要可以指定 2 台 (Mail server 1 與 Mail server 2).若有指定 2 台 mail server, 當其中一台服務失敗時, 會改使用第 2 台來服務.
6. 若 controller 是採用雙網口 WinPAC-8xx7, XP-8xx7-CE6, iPAC-8xx7 或 Wincon-8x47 當 2 個網口都 Enable 時, 若其中一個發生斷線, 會自動切換由另一個網口來發送.
7. 如果要發出的 Email 有夾帶附件, 此附件檔案資料必需在發送前就準備好.

WinPAC-8xx7 / VP-25W7 / XP-8xx7-CE6:

Wincon-8xx7:

需存放於 '\Email_ETH\' 路徑內, 例如, '\Email_ETH\A1.txt'

可以參考使用 例如 `TMP := F_copy('\Micro_SD\B9.jpg', '\Email_ETH\B9.jpg');` 來將 file 覆製到 \Email_ETH\ 路徑內.

發完 mail 後可以參考使用 例如 `TMP := F_delete('\Email_ETH\B9.jpg');` 來將 file 刪除

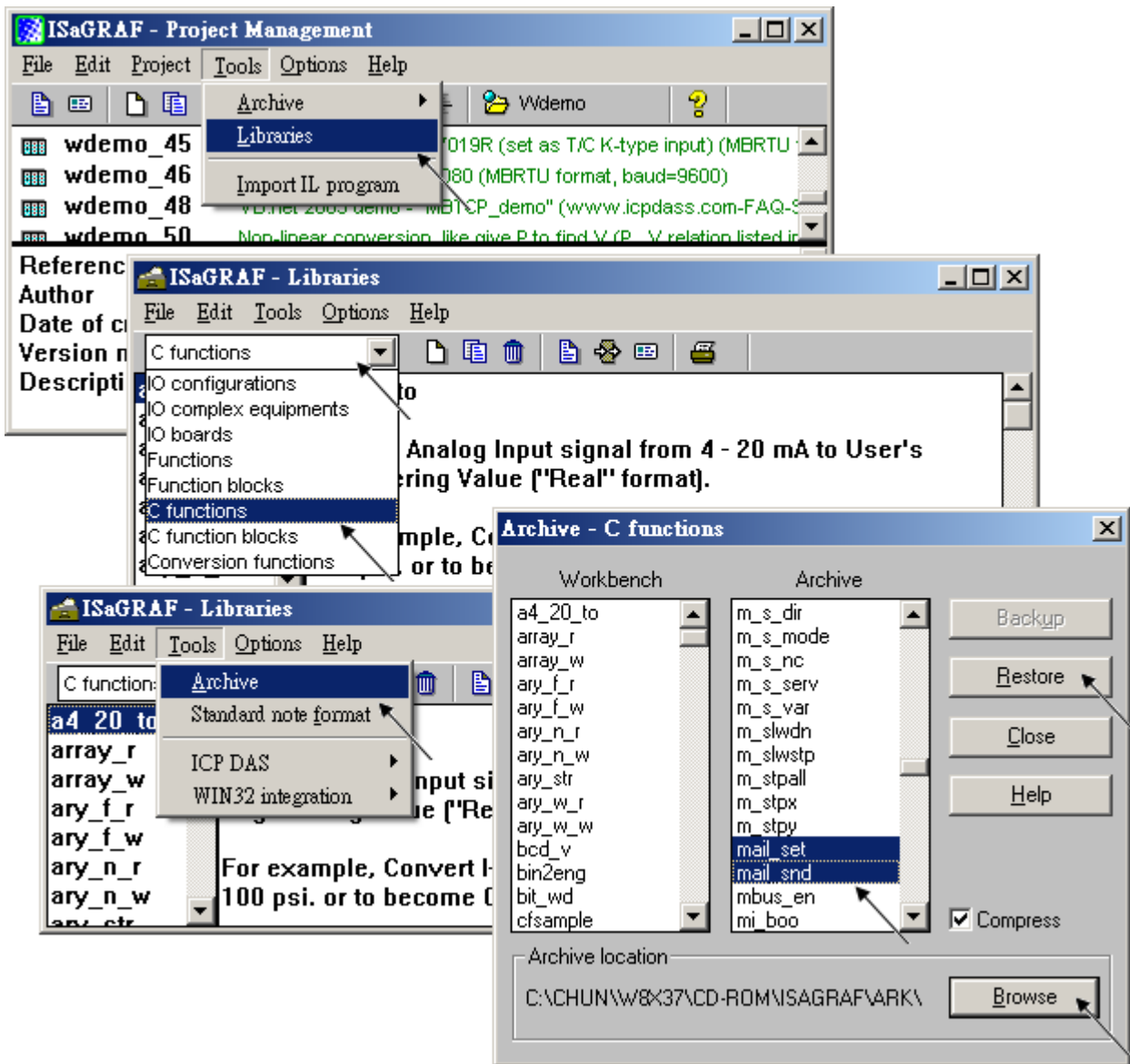
iPAC-8447 / 8847 與 μPAC-7186EG:

需存放於電池備援 SRAM 內. 可以使用 “S_xxx” 等函式來操作該 SRAM, 例如

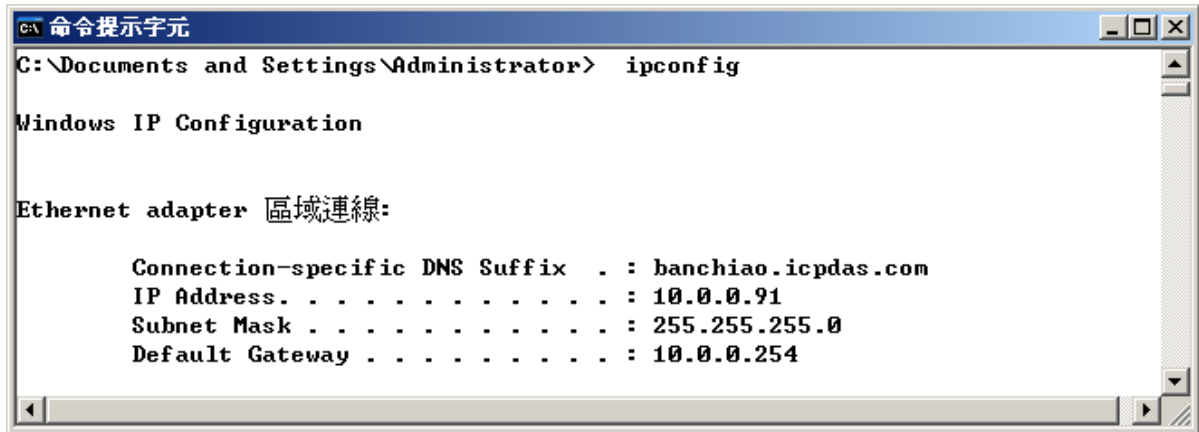
“s_fl_ini”, “s_fl_avl”, “s_m_r”, ...

(請參考 第 10.3 節 與 附錄 A.4)

請先確認您 PC 上的 ISaGRAF 軟體是否已經安裝了 Mail_snd , Mail_set 與 R_mb_adr 等 ISaGRAF c-function. 若沒有請至 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-067 或 076 下載 Demo program, 將其內的 Mail_snd.uia , Mail_set.uia 與 R_mb_adr.uia 採用以下方法安裝起來。

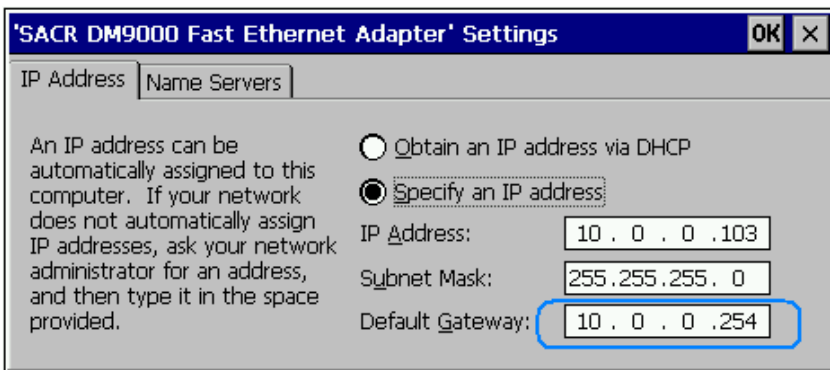


要能正確的發出 Email, 必需設定 Controller 網口的 Gateway IP, 請先用一台同區域網路內的 PC, 開啟 command prompt 視窗, 下達 ipconfig 命令, 就可查到這個區網所使用的 Gateway IP. 如下 (本例為 10.0.0.254)



然後請把此 Gateway IP address 填入 controller 的網口設定 (若使用 雙網口 W-8347/8747 可以 LAN1 與 LAN2 都 Enable, 此時 2 個都需要填入此 Gateway IP)

Wincon-8xx7 / 8xx6 :



之後請運行 Wincon 內的 Wincon Utility – Save and Reboot , 將 Wincon 的設定值存起來, 它會自動重新開機一次.



iPAC-8447 / 8847 與 μ PAC-7186EG :

請在 PC 上運行 “7188xw.exe”，之後下達 “gateway 10.0.0.254” 命令來設定 gateway IP 為 10.0.0.254. (請參考 ”ISaGRAF 進階使用手冊” 附錄 B)

該 command prompt 視窗也可用來查詢 Mail server 的 IP (ISaGRAF 程式內需用到), 比如若要查 msa.hinet.net 的 IP, 請下達 TraceRT msa.hinet.net , 如下 (本例為 168.95.4.211)

```
C:\Documents and Settings\Administrator> TraceRT msa.hinet.net

Tracing route to msa.hinet.net [168.95.4.211]
over a maximum of 30 hops:

  0  <1 ms    <1 ms    <1 ms    10.0.0.254
  1  1 ms     <1 ms     1 ms     61-218-42-1.HINET-IP.hinet.net [61.218.42.1]
  2  27 ms    27 ms    28 ms    10.218.42.254
  3  26 ms    28 ms    27 ms    tp-s2-c76r5.router.hinet.net [168.95.82.206]
  4  27 ms    27 ms    *        220-128-2-234.HINET-IP.hinet.net [220.128.2.234]
  5  27 ms    27 ms    27 ms    220-128-2-225.HINET-IP.hinet.net [220.128.2.225]
  6  27 ms    27 ms    28 ms    msa.hinet.net [168.95.4.211]
```

從 www.icpdas.com > FAQ > Software > ISaGRAF > 067 與 076 下載的 program, 其內包含 3 個.

Wdemo_62.pia 為不夾帶附件的 demo program.

Wdemo_63.pia 為夾帶附件的 demo (只在 W-8xx7, WP-8xx7, VP-25W7, XP-8xx7-CE6 內運作).

Wdmo_63a.pia 為夾帶附件的 demo program (只能在 μ PAC-7186EG, iPAC-8447/8847 內運作).

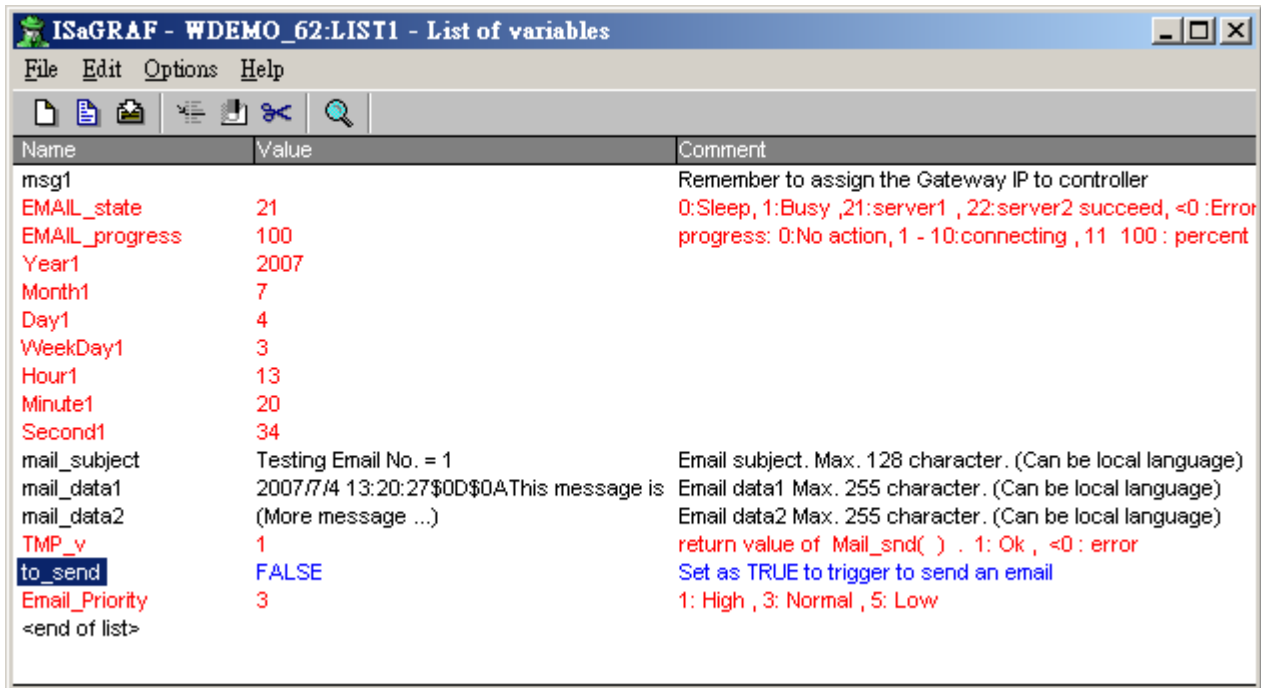
這 3 個 demo program 內使用的 Mail server 與 收件人設定 必需要修改成你自己使用的.

```
TMP := MAIL_SET( 1, 'chun@icpdas.com' );      (* 收件人 1, 請修改 *)
TMP := MAIL_SET( 100, 'go_mao@hotmail.com' ); (* 寄件人, 請修改 *)
TMP := MAIL_SET( 101, '168.95.4.211' );      (* Mail server 1 的 IP, 請修改 *)
```

(* 有些 MAIL server 是需要先 login 後, 才能利用它來發出 mail, 此時就需要設定以下 2 項. 若使用的 Mail server 1 與 server 2 並不需要先 login 就可發出 mail, 請不要設定以下 2 項 *)

```
TMP := MAIL_SET( 104, 'MY_ACCOUNT' ); (* 有在該 mail server 註冊的 account *)
TMP := MAIL_SET( 105, 'MY_PASSWORD' ); (* 有在該 mail server 註冊的 password *)
```

之後重新 Compile , 再 download 該程式 到你的 controller 內運行 會出現如下視窗.
 請將 to_send 設為 TRUE 來發送一封 Email. 幾秒後, 若成功 Email_state 的值會是 21 或 22,
 若失敗 Email_state 的值會小於 0. Email_progress 若為 100 表示資料已經 100% 發送出去了.



以下是發送 Email 使用的函式說明.

MAIL_Set(CMD_ , MSG_)

輸入參數:

CMD_ Integer 可以使用以下的值.

- 1 : 設定 receiver 1 的郵箱地址.例如, `TMP := Mail_set(1 , 'chun_tsai@icpdas.com') ;`
郵箱地址最長不可超過 48 個字.
- 2 ~ 10 : 設定 receiver 2 到 10 的郵箱地址 (可以不設定, 需要使用時才要設).
- 100 : 設定發信人的郵箱地址. 例如 `TMP := Mail_set(100 , 'sender1@icpdas.com') ;`
郵箱地址最長不可超過 48 個字.
- 101 : 設定 mail server 1 的 IP 地址. 例如, `TMP := Mail_set(101 , '168.95.4.211') ;`
- 102 : 設定 mail server 2 的 IP 地址 (可以不設定).
- 103 : 設定使用新的 TCP port No. 來發 email.
(一般都不必設定, 內定為使用 25 : “SMTP protocol”)

***** CMD_ 104, 105 與 106 從以下 ISaGRAF driver 版本起才有支持 *****
µPAC-7186EG (ISaGRAF driver Ver. 1.14 or later)
iPAC-8xx7 (ISaGRAF driver Ver. 1.10 or later)
WinPAC-8xx7 / 8xx6 (ISaGRAF driver Ver. 1.37 or later)
VP-25W7 / 23W7 / 25W6 / 23W6 (ISaGRAF driver Ver. 1.29 or later)
XP-8xx7-CE6 / XP-8xx6-CE6 (ISaGRAF Ver. 1.17 or later)

- 104 : 設定 user_account 給需要先 login 才能發 mail 的 mail server,
例如, `TMP := Mail_set(104 , 'my_account') ;`
- 105 : 設定 user_password 給需要先 login 才能發 mail 的 mail server,
例如, `TMP := Mail_set(105 , 'my_password') ;`

若使用的 mail server 不需要 login 就可以發出 mail, 請不要設定 CMD_ 104 與 105.

- 106 : 設定 Timeout, 單位是 秒, 值可以是 30 ~ 180 , 若不設定, 上方所列版本內
定為 60 秒, 舊版則是內定為 20 秒.
改成 75 秒 可使用 `TMP := Mail_set(106 , '75') ;`

MSG_ Message 對應於 “CMD_” 所要輸入的字串資料

傳回值:

Q_ Boolean True : 正確 . False : 失敗.

MAIL_snd(Start_ , Num_ , Subject_ , Prio_ , Data1_ , Data2_ , Attach_)

輸入參數:

Start_ Integer 要傳給那些 receiver 編號, 從編號幾號開始? 值可以是 1 ~ 10.
Num_ Integer 共要傳給幾個 receiver, 值可以是 1 ~ 10.
Subject_ Message 郵件主旨. 最長為 128 個字. 例, 'Alarm of plant 1'
Prio_ Integer 要使用的郵件優先權符號.

可以是 1: 高, 3: 一般, 5: 低. 內定值為 3.

Data1_ Message 郵件內容前半段 (最長為 255 個字).
例如, 'Pressure 1 is too high. Please check it soon ! ... '

Data2_ Message 郵件內容後半段 (最長為 255 個字).
例如, 'More message ...'

Attach_ Message 附件檔名或編號. (看是使用那種 Controller)
若不夾帶附件, 請填入 "(空字串)".

WP-8xx7 & W-8xx7: 檔名不可超過 64 個字, 必需存放於 '\Email_ETH\' 路徑內.
例如, '\Email_ETH\A1.txt'

μPAC-7186EG: 檔案必需存放於 X-607, X-608 電池備援 SRAM 內 .

iPAC-8447/8847: 檔案必需存放於電池備援 SRAM 內.

μPAC-7186EG 與 iPAC-8447/8847 可以使用 '1', '2', ..., '8' .

該數字表示 "S_FL_AVL" 函式設的 檔案編號

(請參考 "ISaGRAF 進階使用手冊" 第 10.3 節與附錄 A.4)

最大可夾帶的附件檔案大小為:

WP-8xx7 & W-8xx7:	2M Bytes.	iPAC-8447 / 8877:	488K Bytes
μPAC-7186EG + X607:	112K Bytes	μPAC-7186EG + X608:	488K Bytes

傳回值:

Q_ Integer

1 : 正確, 之後會開始發送 email.

< 0 : 發生錯誤

-1 : 忙碌中 . 前一封 email 還在發送中.

-2 : 郵件收件者沒設定或設錯 ("mail_set" 函式的參數 "Start_" 沒設定或設錯).

-3 : Mail server 1 沒設定或設錯.

-4 : 發信人的郵箱地址沒設定或設錯.

-5 : "Start_" 參數值小於 1 或大於 10

-6 : "Subject_" 參數 長度超過 128 個字.

-7 : 郵件功能未啟用. 請使用"mail_set" 函式來設定至少 1 個收件者郵箱地址, 至少 1 個正確的 mail server IP 與 發信人的郵箱地址

-8 : "Num_" 值設成小於 1 或 大於 10

-9 : 指定的附件檔名不存在 或 長度超過 64 個字 或 檔案太大.

R_MB_ADR(1, 9995) 用來取得 呼叫 "Mail_snd()" 後 的發送狀態.

傳回值 :

- 0 : 休息中. 無命令下達.
- 1 : 忙碌. 正在發送中.
- 21 : Email 已經成功地經由 Mail server 1 發送出去了.
- 22 : Email 已經成功地經由 Mail server 2 發送出去了.

- < 0 : 發生錯誤.
- 1 : 無法連上 Mail server
- 2 : Mail server 拒絕該 發信人傳送的郵件.
- 3 : Time out
- 4 : Ethernet socket error
- 5 : Mail server 拒絕該 收信人郵箱地址.

R_MB_ADR(1, 9994) 用來取得 呼叫 "Mail_snd" 後 的發送進度.

呼叫 "R_MB_ADR(1, 9994)" 無法偵測錯誤.必需使用 "R_MB_ADR(1, 9995)" 才能偵測錯誤.當錯誤發生時,該錯誤編號會一直保留直到下次再呼叫 "Mail_snd()" 為止

傳回值:

- 0 : 休息中. 無命令下達
- 1 : 正在連接 Mail server 1
- 2 : 已經連上 Mail server 1 . 傳出 "HELO" 訊息
- 3 : 傳出 "MAIL FROM: ..." 訊息給 Mail server 1
- 4 : 傳出 " RCPT TO: ..." 訊息給 Mail server 1
- 5 : 傳出 "DATA" 訊息給 Mail server 1
- 6 : 正在連接 Mail server 2
- 7 : 已經連上 Mail server 2 . 傳出 "HELO" 訊息
- 8 : 傳出 "MAIL FROM: ..." 訊息給 Mail server 2
- 9 : 傳出 " RCPT TO: ..." 訊息給 Mail server 2
- 10 : 傳出 "DATA" 訊息給 Mail server 2

- 11 ~ 100 : 目前資料傳送進度.
- 例如, 25 表示 25 / 100 = 25 %
- 36 表示 36 / 100 = 36 %
- 95 表示 95 / 100 = 95 %
- 100 表示 100 / 100 = 100 % (全傳完了)

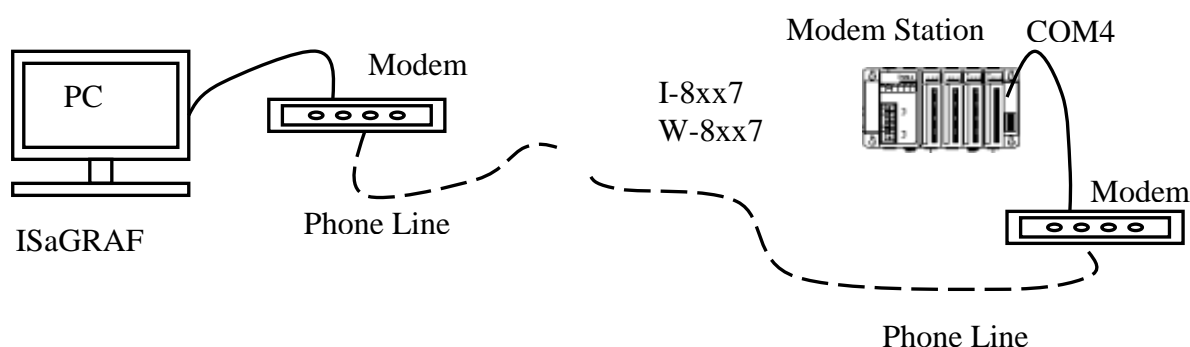
第 13 章透過 Modem_Link 遠程監視及下載程式

13.1: Introduction

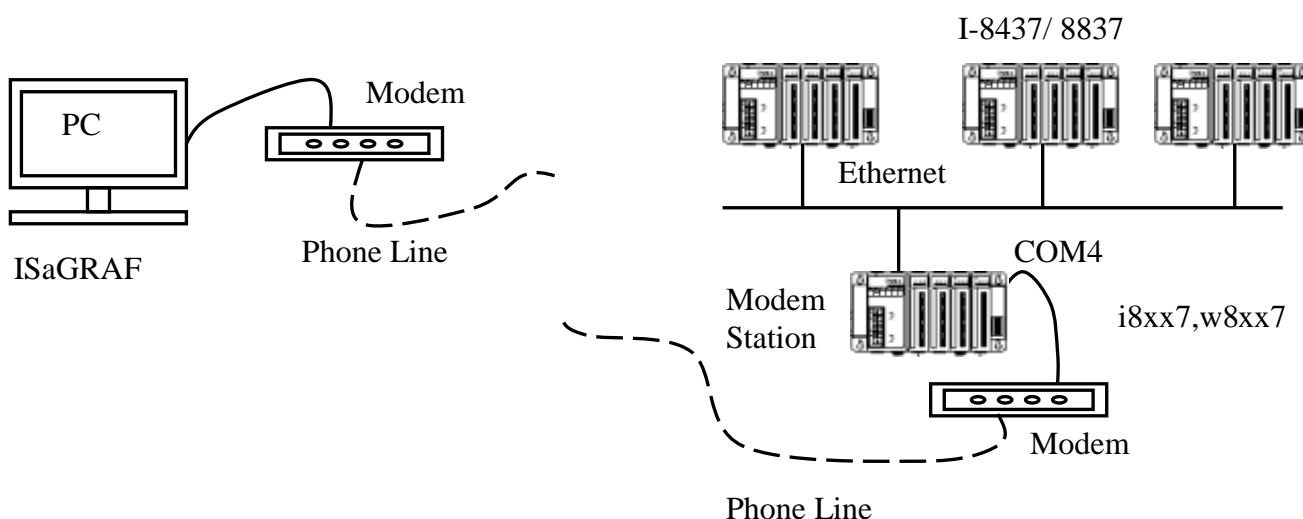
I-8417/8817/8437/8837 控制器的 COM4 & W-8xx7 的 COM2 支援完整的 Modem 接腳訊號。I-8xx7 從驅動程式 2.14 版開始, W-8xx7 從驅動程式 3.10 版開始, 有支援 Modem_Link 功能可以監視遠方的控制器的運作狀況或下載程式。請參閱附錄 C 的方法安裝最高版本的驅動程式。可由網路上取得:

<http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

必須連接“Modem Station”的 COM4 到一台 Modem, Com4 的接腳與 PC 上的 COM1 完全一樣 (9-pin Dsub)。連接圖示如下。



“Modem Station”指的是連到 Modem 的那台 I-8417/8817/8437/8837 或 W-8xx7, 它會接聽由遠方電腦打過來的電話。如果使用的控制器為 I-8437/8837 (有倚太網), 就可規劃多台 I-8437/I-8837 連在 1 起, 遠方的電腦可連到與 Modem Satation 連在一起的任 1 台 I-8437/8837, 如下。



注意:

1. 若要使用 W-8xx7 的 “modem_link” 功能, 請參考 W-8xx7 的 “快速上手手冊” 來關閉 COM2 使用的 Modbus RTU 功能。
2. Wincon-8xx7 只支持 Modem_station (有接 Modem 的那一台)

13.2: 透過 Modem_Link 下載程式

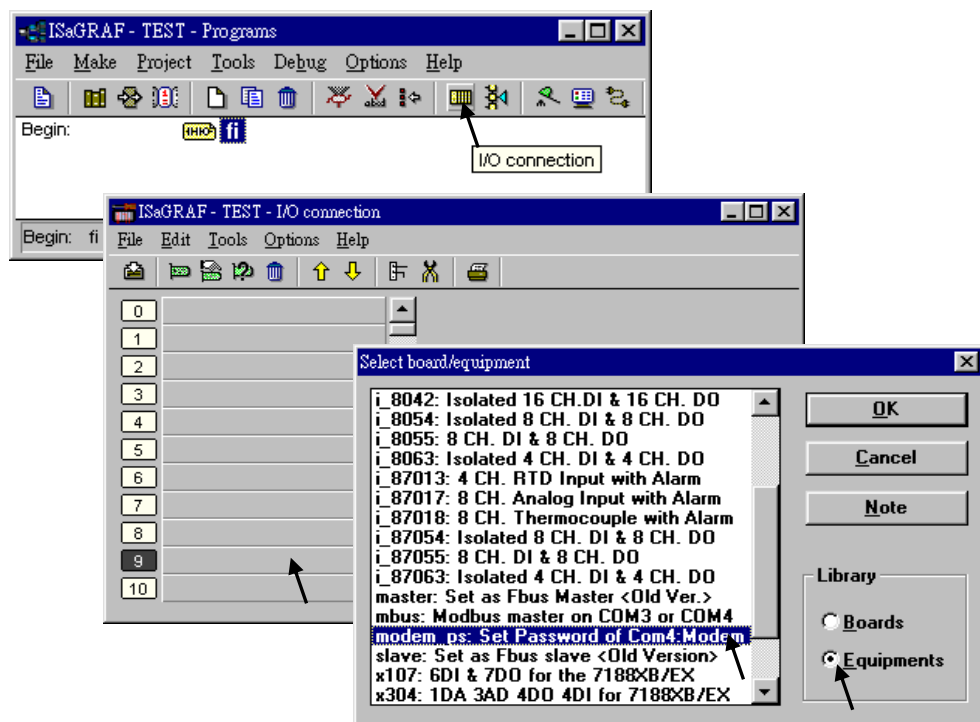
注意:

1. 要更改遠方控制器的程式有時是很危險的，電話線路品質，忙碌情況，天氣，...，都有影響。若非必要，盡量不要用 Modem_Link 下載程視。但監視功控器的運作情況倒是可以進行，也無危險
2. W-8xx7 的 COM2 預設為支持 Modbus RTU 協議的通訊埠，若要使用 W-8xx7 的“modem_link”功能，請參考 W-8xx7 的“快速上手手冊”來關閉 COM2 原先預設的 Modbus RTU 功能。

警告:

不要下載一個會占用 I-8xx7:COM4 或 W-8xx7:COM2 到其它用途的程式，因為如此 Modem_Link 功能會無法運作，再也無法由遠方連線，除非再跑到控制器旁邊把程式改回來。比如在 I-8xx7 使用了“Bus7000”及“Mbus”在 port_no = 4 上，或者在程式內用了“Comopen”函數去開啟 Com4，... 都會使 Modem_Link 功能會無法運作。但第 12 章說明的“Email”方塊不在此限。

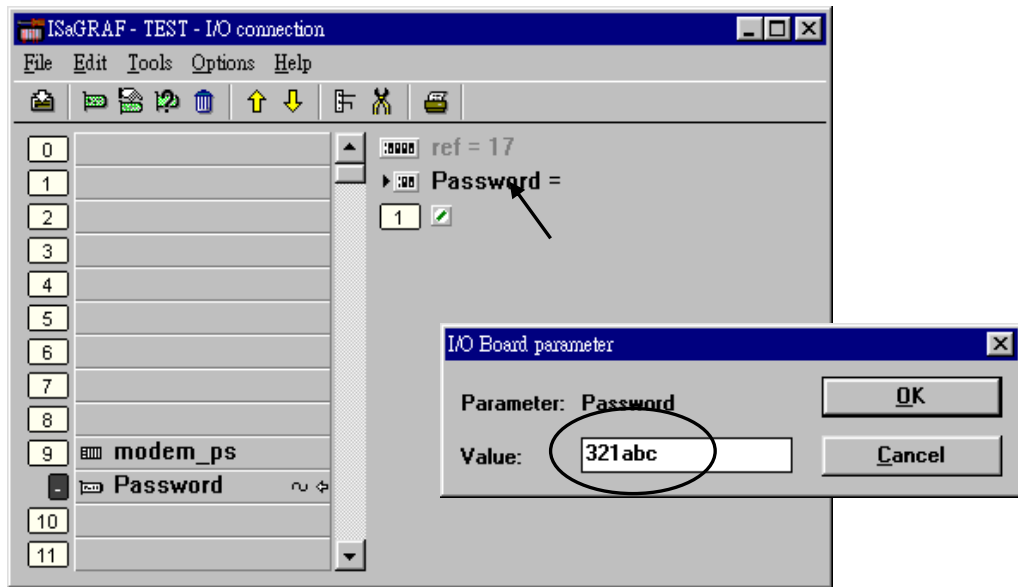
最好在 Modem Station 那台控制器的程式內加上“modem password”以保護你的系統不被外人入侵。請進入 I/O connection 視窗，連接“Modem_PS”



之後輸入密碼。密碼最多 12 個字，且不能使用“及‘符號。之後重新編譯，下載到 Modem Station 即可。

注意:

使用者可寫 Visual Basic 程式透過來 Modem 及電話線來存取遠方 I-8417/8817/8437/8837 內的資料。請由以下網址下載 VB6 的範例程式。
ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/vb_demo or
I-8000 CD-ROM:\napdos\isagraf\vb_demo\

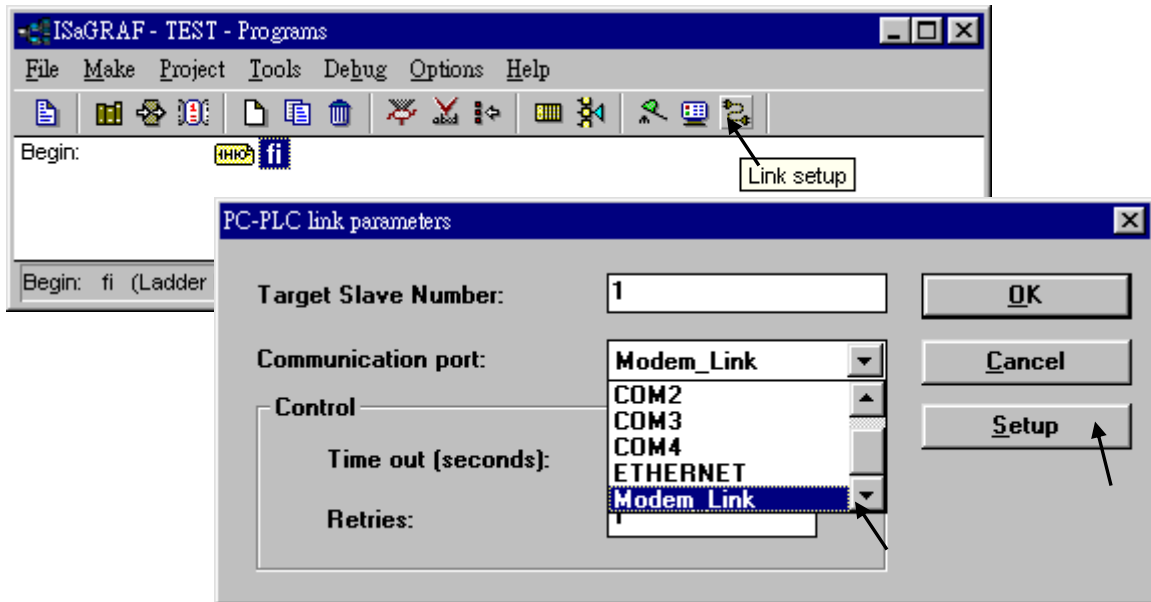


非常重要:

如果 Modem Station 沒有設定密碼，任何知道那隻電話號碼的人，都可用 ISaGRAF 以 Modem_Link 去連它，之後可以更改掉程式。建議最好要設密碼。

以下的步驟為使用 Modem_Link 的方法。

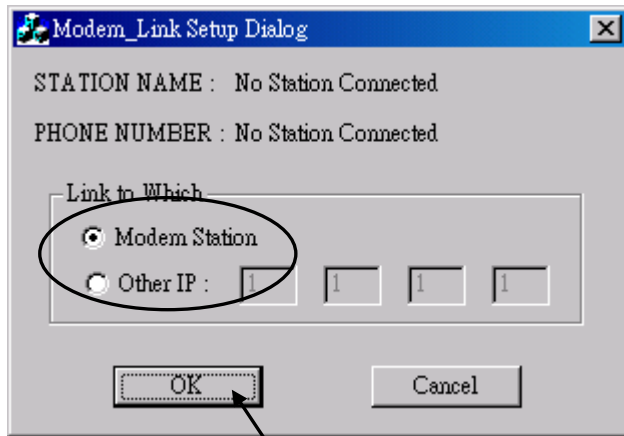
點選“Link setup”，選取 “Modem_Link”，之後點選 “Setup”



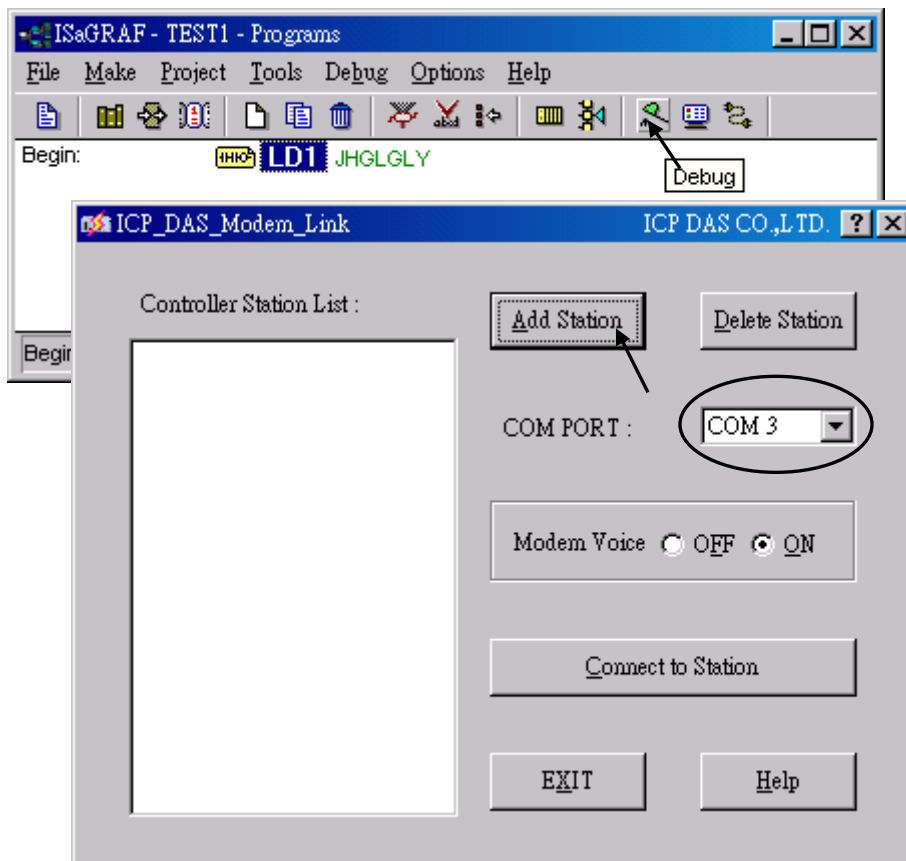
Windows NT, 2000 & XP 的使用者:

若你的 PC 是 Windows 95, 98 請跳至次三頁的 “Windows 95, 98 的使用者”。

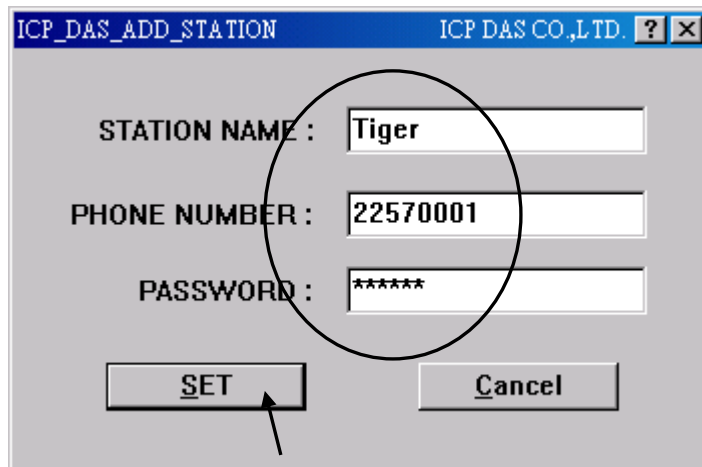
進入”Setup”後，如果你要連的是 Modem station 那台控制器，選取 “Modem station”，不然選取 “Other IP”。“Other IP” 指的是和 Modem Station 用倚太網相連的 I-8437/8837，必須輸入它的 IP 位址。



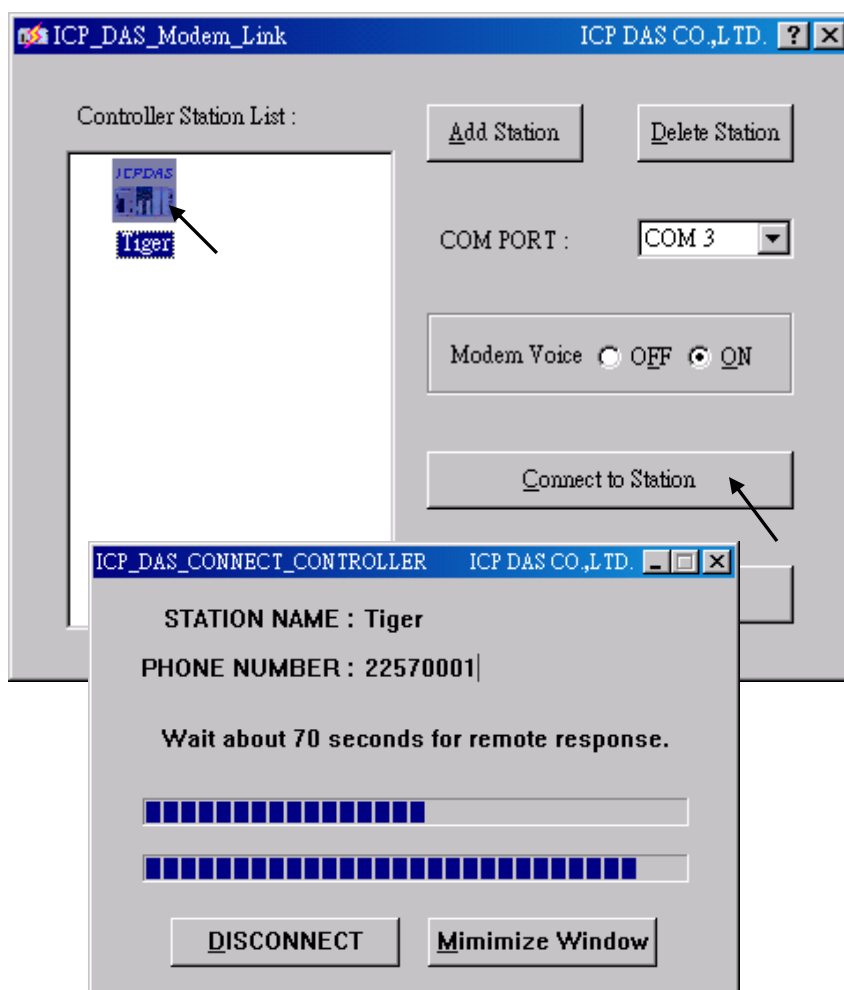
離開”Link Setup”視窗後，點選 “Debug”. 選取目前 PC 上連 Modem 的 Com 埠，之後點選 “Add Station” (假如視窗的左邊都沒有).



然後會看到以下的視窗. 輸入 1 個名字，正確的電話號碼及密碼給這個新建的 Station. 如果在電話號碼內出現 逗號 “,”. 它會停留 1 秒再撥剩下的號碼. 例如 “9,,22570001” 會先撥 9，然後停 2 秒，再撥“22570001”。

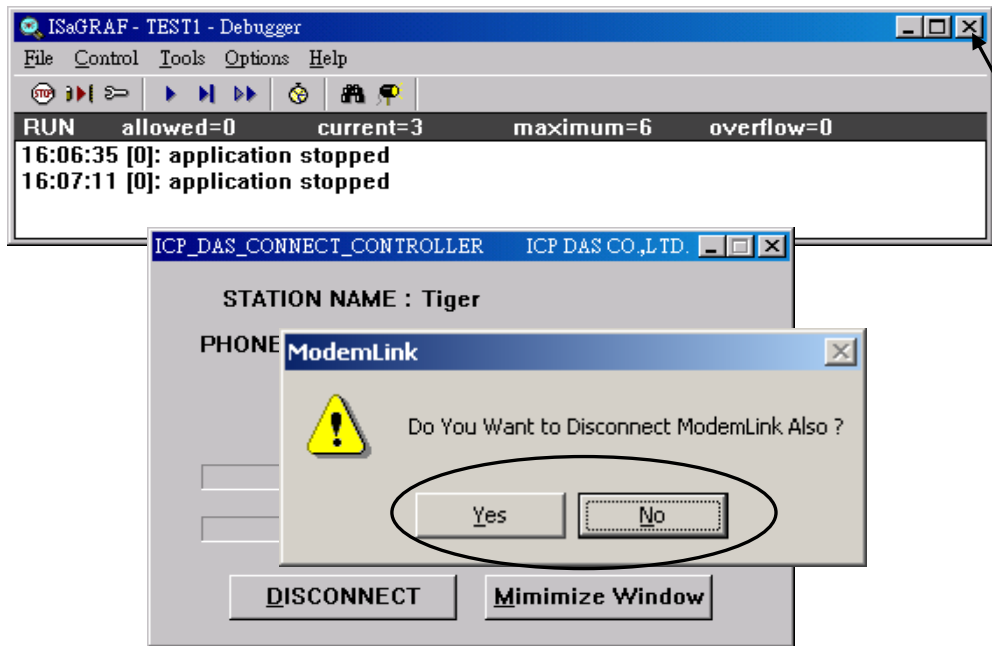


選取該 Station 圖示，之後點選“Connect to Station” 去撥號連至遠方的控制器。

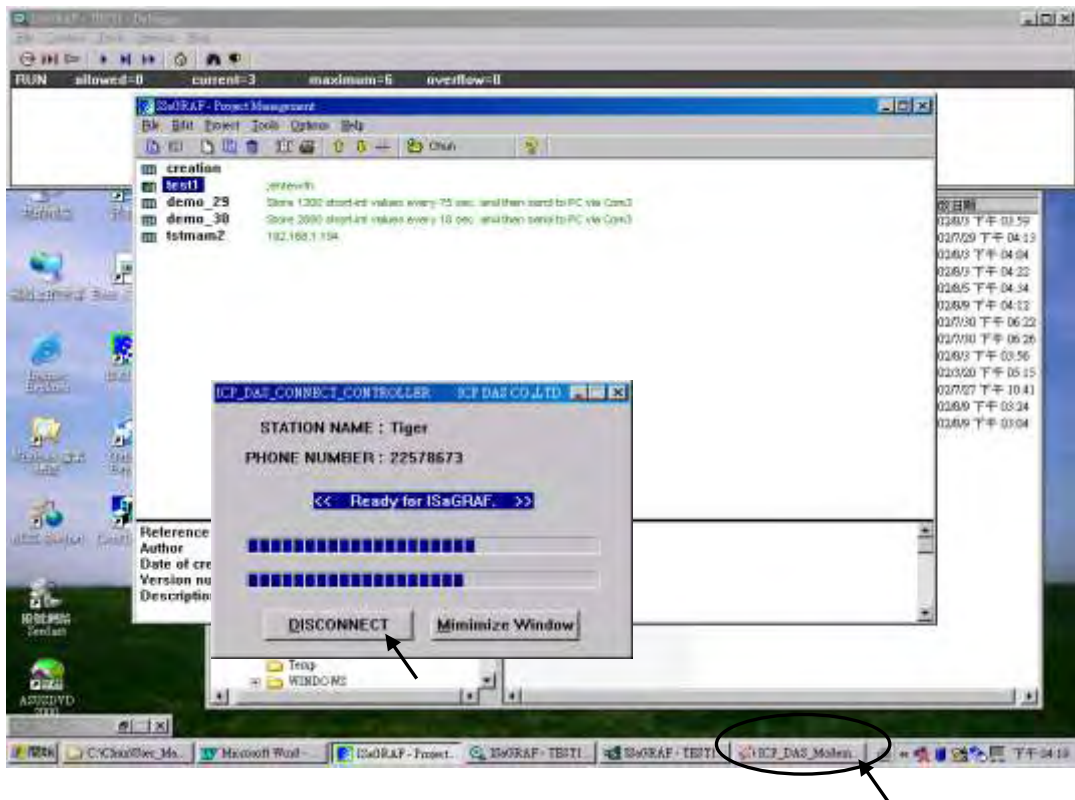


連線成功後，你可以監視程式，變數...等的運作狀況，也可更改變數的值或下載成新的程式

要與遠方的控制器斷線，請關閉“... Debugger”視窗。之後你可選則“**Yes**”去掛斷電話，或選“**No**”讓電話繼續連線。電話繼續連線的好處是，可再連到 Modem Station 倚太網上的任一台，而不必重新撥號。



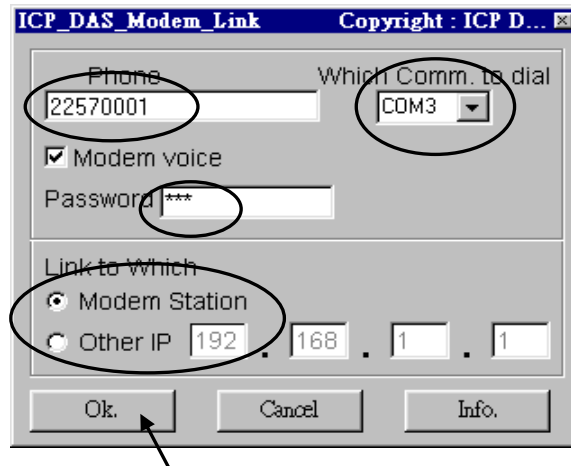
然而，記得該掛斷的時後還是得掛斷，避免多繳電話費。



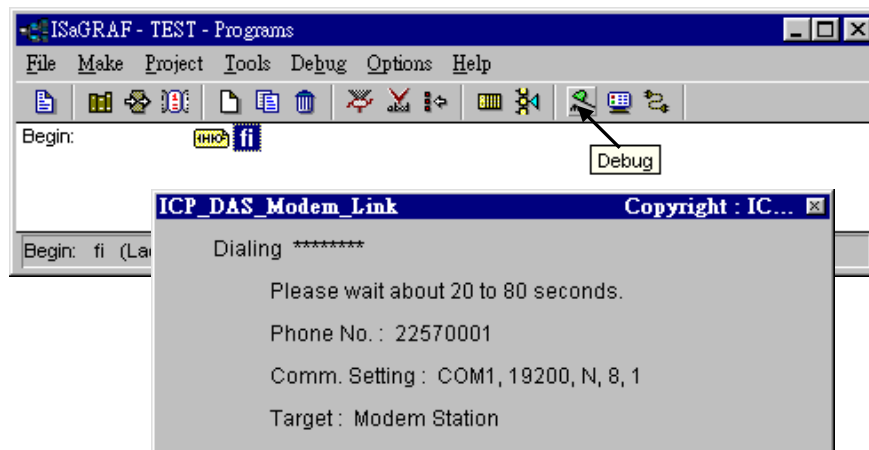
Windows 95, 98 的使用者:

進入”Setup”後，輸入電話號碼，目前 PC 上連 Modem 的 Com 埠，正確的密碼。如果在電話號碼內出現逗號“,”。它會停留 1 秒再撥剩下的號碼。例如“9,,22570001”會先撥 9，然後停 2 秒，再撥 22570001。

如果你要連的是 Modem station 那台控制器，選取 “Modem station”，不然選取 “Other IP”。“Other IP” 指的是和 Modem Station 用倚太網相連的 I-8437/8837，必須輸入它的 IP 位址。

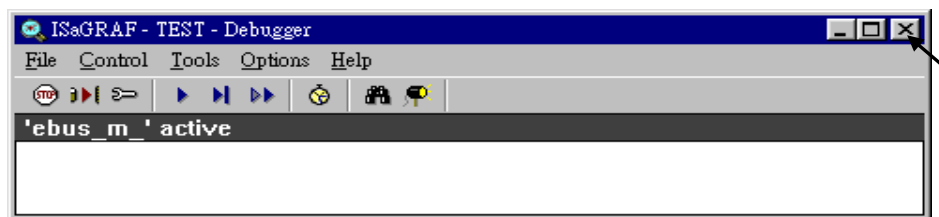


然後按下“Debug” 開始撥號連到遠方。



連線成功後，你可以監視程式，變數...等的運作狀況，也可更改變數的值或下載成新的程式

與 Windows NT, 2000 & XP 不同的是，當關掉 “... Debugger” 視窗時，連電話也會被掛斷。若要連線會再重撥一次。



第 14 章 Spotlight：簡易圖控

Spotlight 是 ISaGRAF 提供的簡易圖控的功能。圖控元件包含 Boolean Icon, Bar Graph, Trend Curve, Value Text, Bitmap Picture。

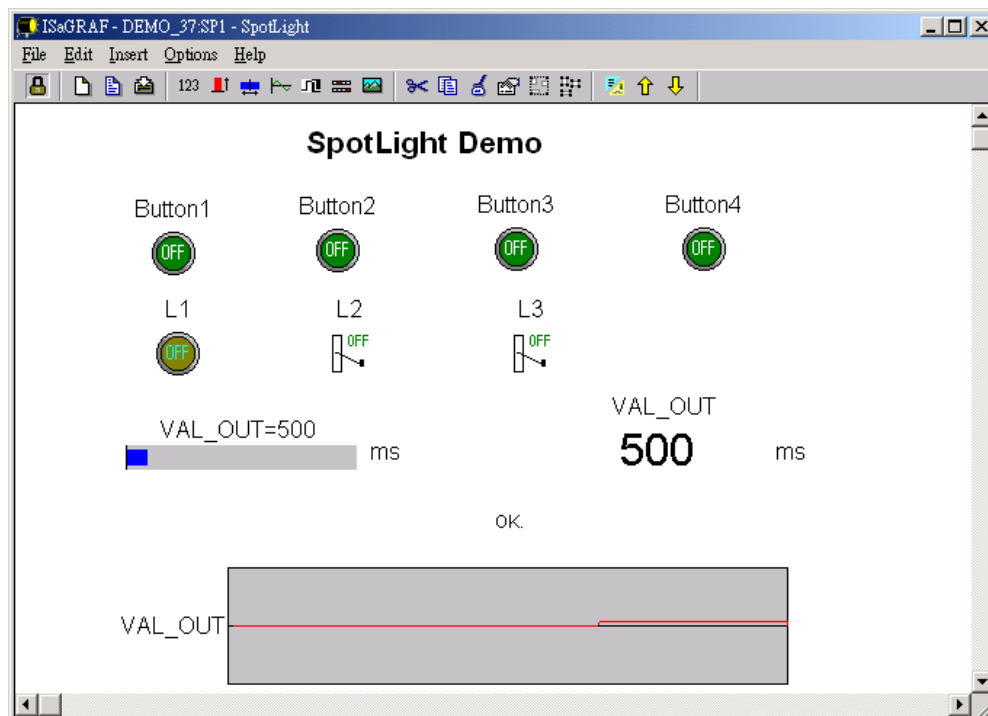
14.1: Spotlight 範例:

本範例放在 ICP DAS I-8000 CD-ROM 內的 “demo_37”。請參閱第 11 章來安裝它到 ISaGRAF 內。

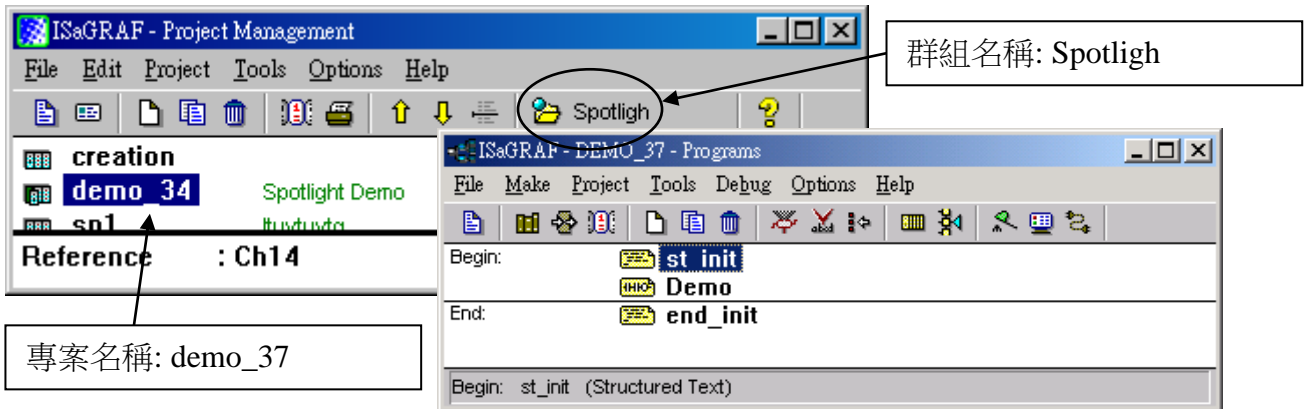
變數:

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 TRUE, 只有在第 1 個 PLC 周期時為 TRUE
L1	Boolean	Output	1 號輸出, 連至 “show3led” 的第 1 點
L2	Boolean	Output	2 號輸出, 連至 “show3led” 的第 2 點
L3	Boolean	Output	3 號輸出, 連至 “show3led” 的第 3 點
Button1	Boolean	Inpput	1 號按鈕, 連至 “spush4key” 的第 1 點
Button2	Boolean	Inpput	2 號按鈕, 連至 “spush4key” 的第 2 點
Button3	Boolean	Inpput	3 號按鈕, 連至 “spush4key” 的第 3 點
Button4	Boolean	Inpput	4 號按鈕, 連至 “spush4key” 的第 4 點
VAL_OUT	Integer	Internal	閃爍周期輸入值, 初值設為 500 (單位: ms)
OLD_VAL_OUT	Integer	Internal	記錄前一個 PLC scan 的 VAL_OUT 舊值
T1	Timer	Internal	閃爍周期 Timer
MSG1	Message	Internal	狀態報告, 請設定其 Maximum Length 為 48

圖控畫面全貌:



專案程式架構:



ST 程式 “st_init” 於 “Begin” 區 :

```
(* Do some init action *)
if INIT = TRUE then
  T1 := TMR(VAL_OUT); (* Convert integer:VAL_OUT to Timer:T1 in ms *)
  MSG1 := 'OK.' ;
  OLD_VAL_OUT := VAL_OUT ; (* init OLD value *)
end_if ;

(* if set a new value to VAL_OUT *)
if VAL_OUT <> OLD_VAL_OUT then

  (* VAL_OUT is acceptable *)
  if (VAL_OUT >= 200) and (VAL_OUT <= 5000) then
    T1 := TMR(VAL_OUT); (* Convert integer:VAL_OUT to Timer:T1 in ms *)
    MSG1 := 'OK.' ;
  else (* VAL_OUT out of range *)
    MSG1 := 'VAL_OUT should be between 200 and 5000 .' ;
  end_if ;

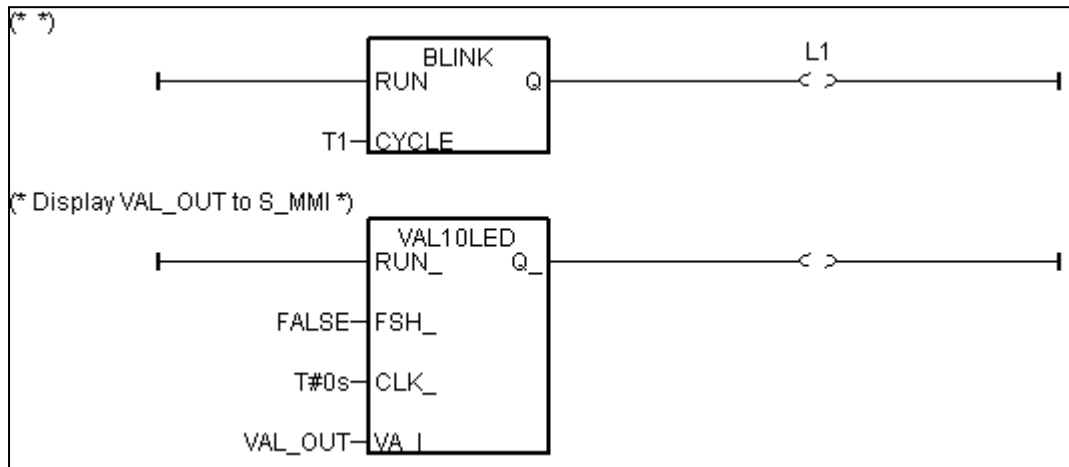
  OLD_VAL_OUT := VAL_OUT ; (* update OLD value *)

end_if ;
```

ST 程式 “end_init” 於 “End” 區 :

```
INIT := FALSE ;
```

LD 程式 “demot” 於 “Begin” 區：



動作說明：

4 個按鈕的狀態會顯示到圖控畫面上.

第 1 個輸出會以 “VAL_OUT” 給的周期時間閃爍.

“VAL_OUT” 的值可由圖控畫面上手動輸入. 同時也會顯示在控制器的面板上.

第 2 及第 3 個輸出可圖控畫面上手動控制.

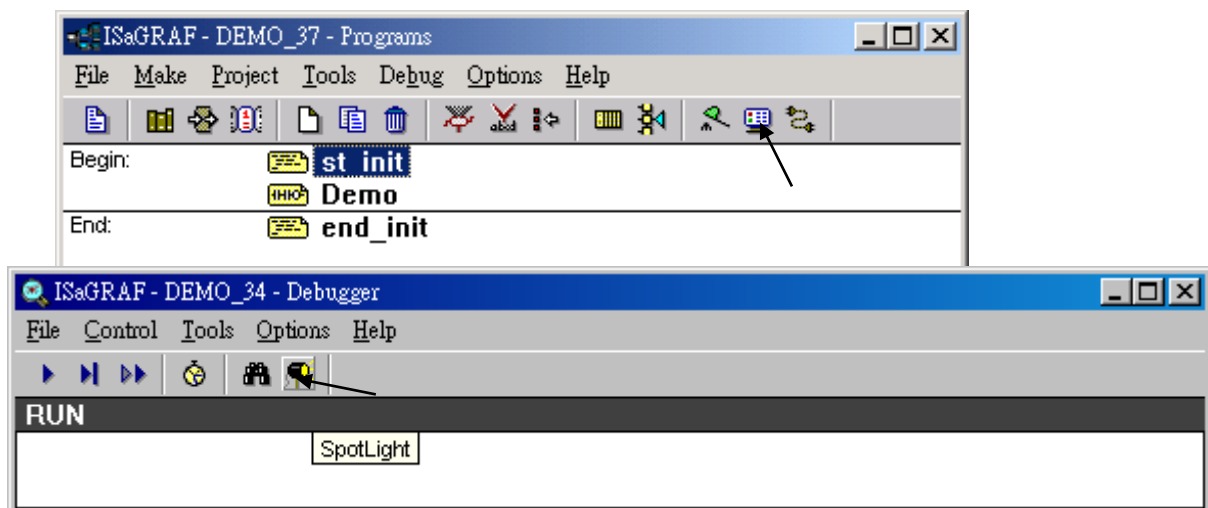
製作 Spotlight 畫面的步驟:

- A. 完成上一頁的專案程式，並編譯好.
- B. 複製 “ICO” 目錄到專案程式的所在目錄
“ICO” 目錄內存放一些 ICP DAS 製定好的 Boolean Icon. 可從 I-8000 光碟內找到
\\napdos\isagraf\ICO\

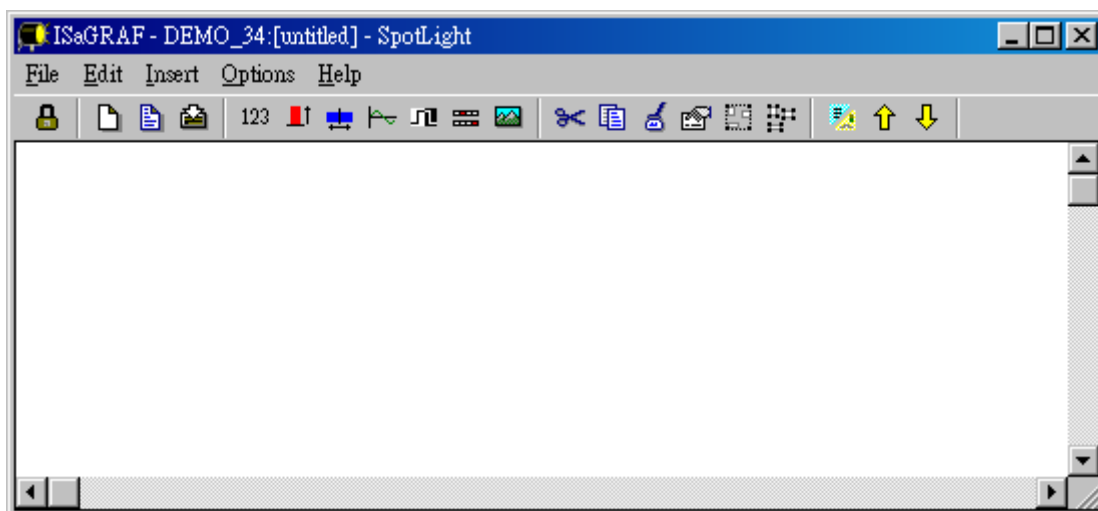
以本例為例，群組名稱為“spotligh”，專案名稱為“demo_37”，所以複製光碟內的
\\napdos\isagraf\ICO*. * 到 c:\isawin\spotligh\demo_37\ 內

假如光碟內沒有可至網路上下載. <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/>

- C. 進入 Spotlight 編輯視窗
點選“Simulate”，之後點選“Spotlight”以進入.

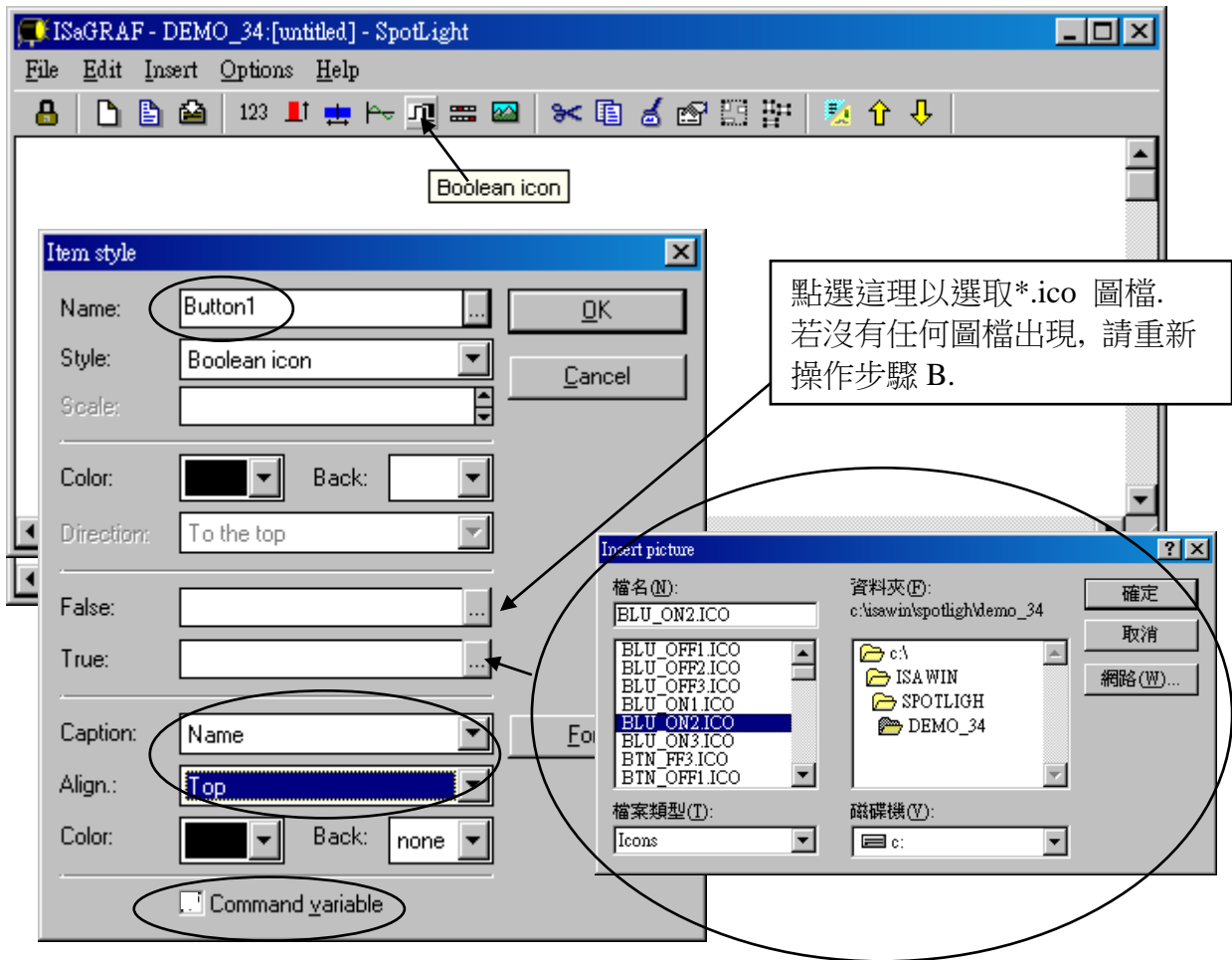


空白的 “SpotLight” 視窗如下所示



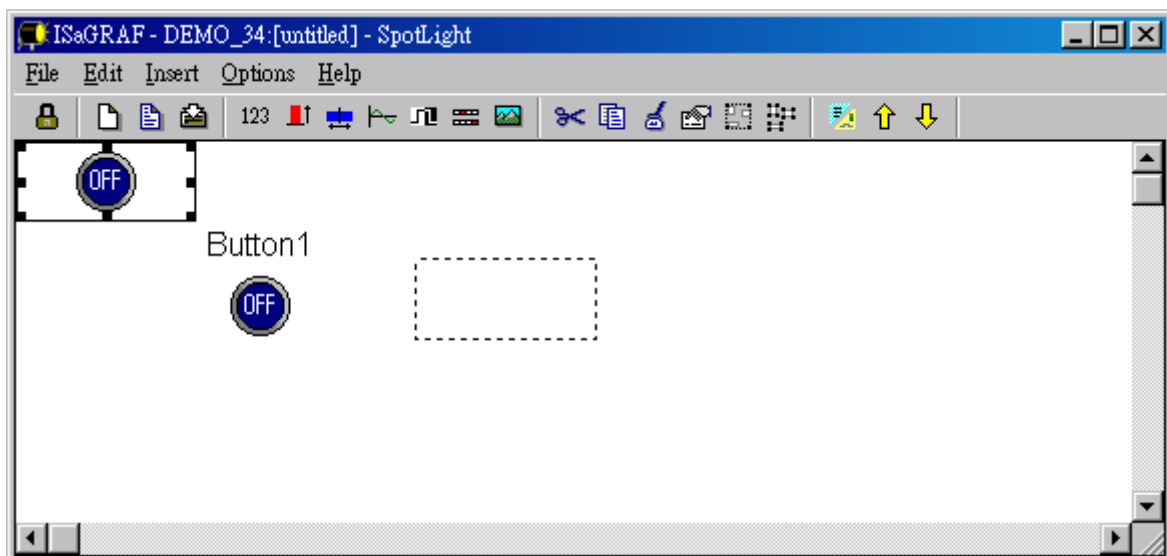
D. 新增 “boolean Icons”

點選“Boolean icon”，之後給定 Name 為 “Button1”，Caption 為 “Name”，Align 為 “Top”，之後選取適當的 *.ico 圖檔給 “FALSE” 及 “TRUE”，並將”Command variable” 選項拿掉。

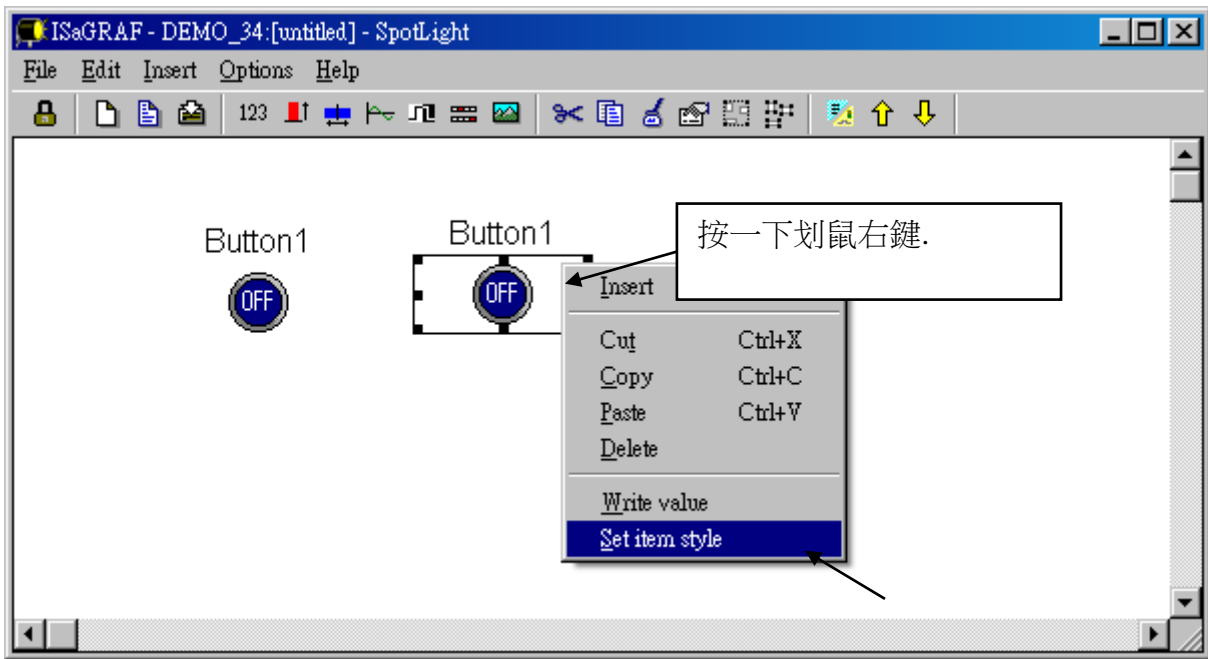


然後拖移此 boolean icon 到適當的位置。

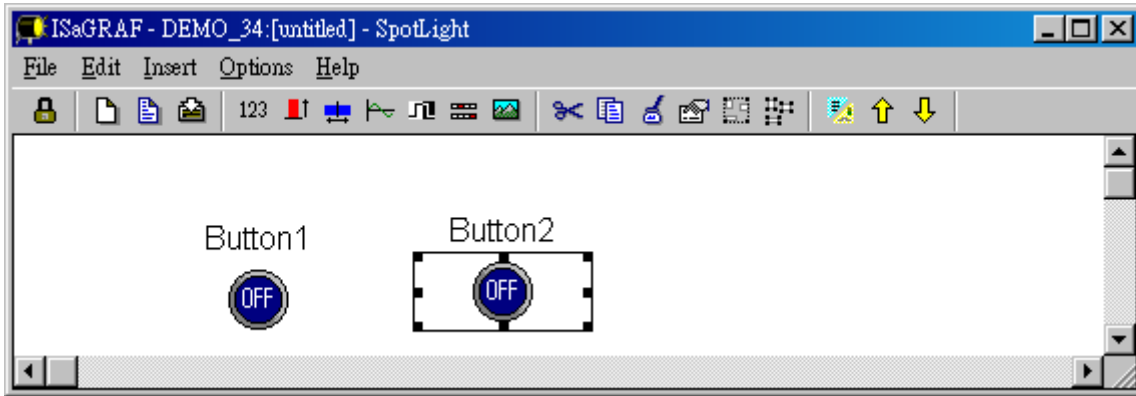
先點選 1 下剛剛新產生的 boolean icon，複製它(Ctrl+c)，然後貼出來(Ctrl+v)，如此會產生 1 個一樣的 boolean icon。將它拖移到適當的位置。



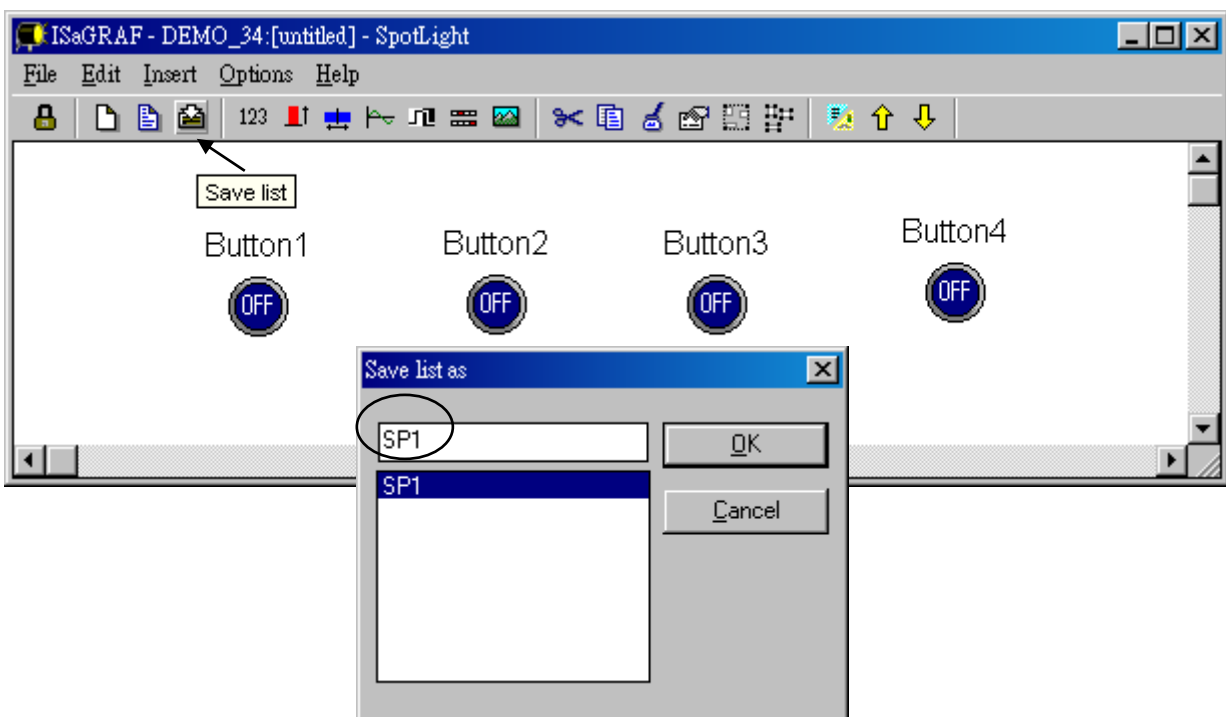
先點選 1 下剛剛新複製出來的 boolean icon，之後按一下划鼠右鍵，選取“Set item style”，改變 Nname 為 “Button2”。



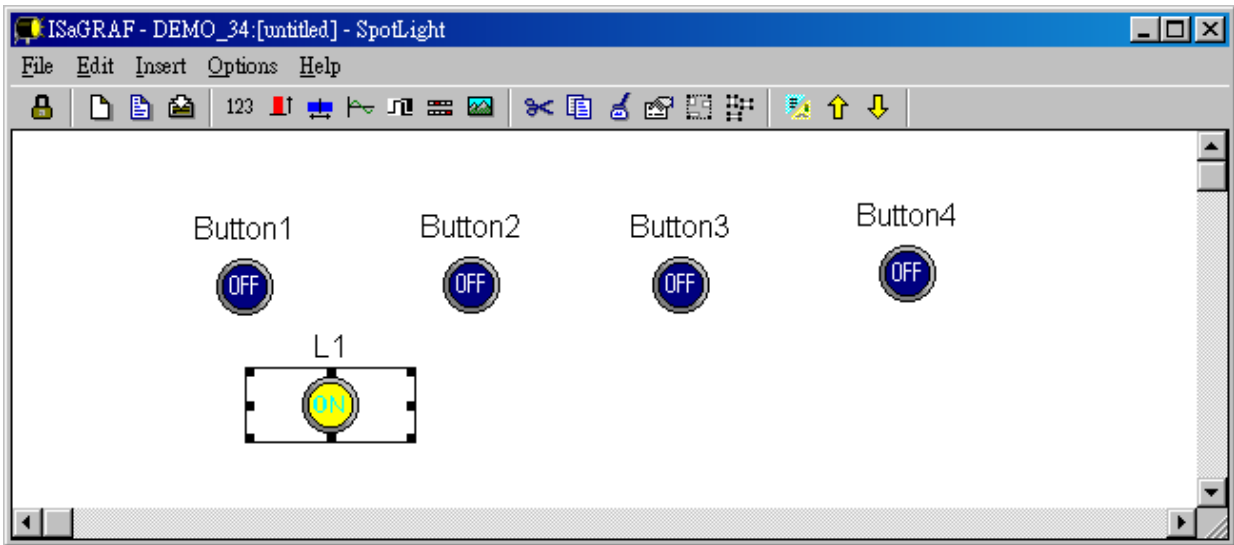
如此得到 ...



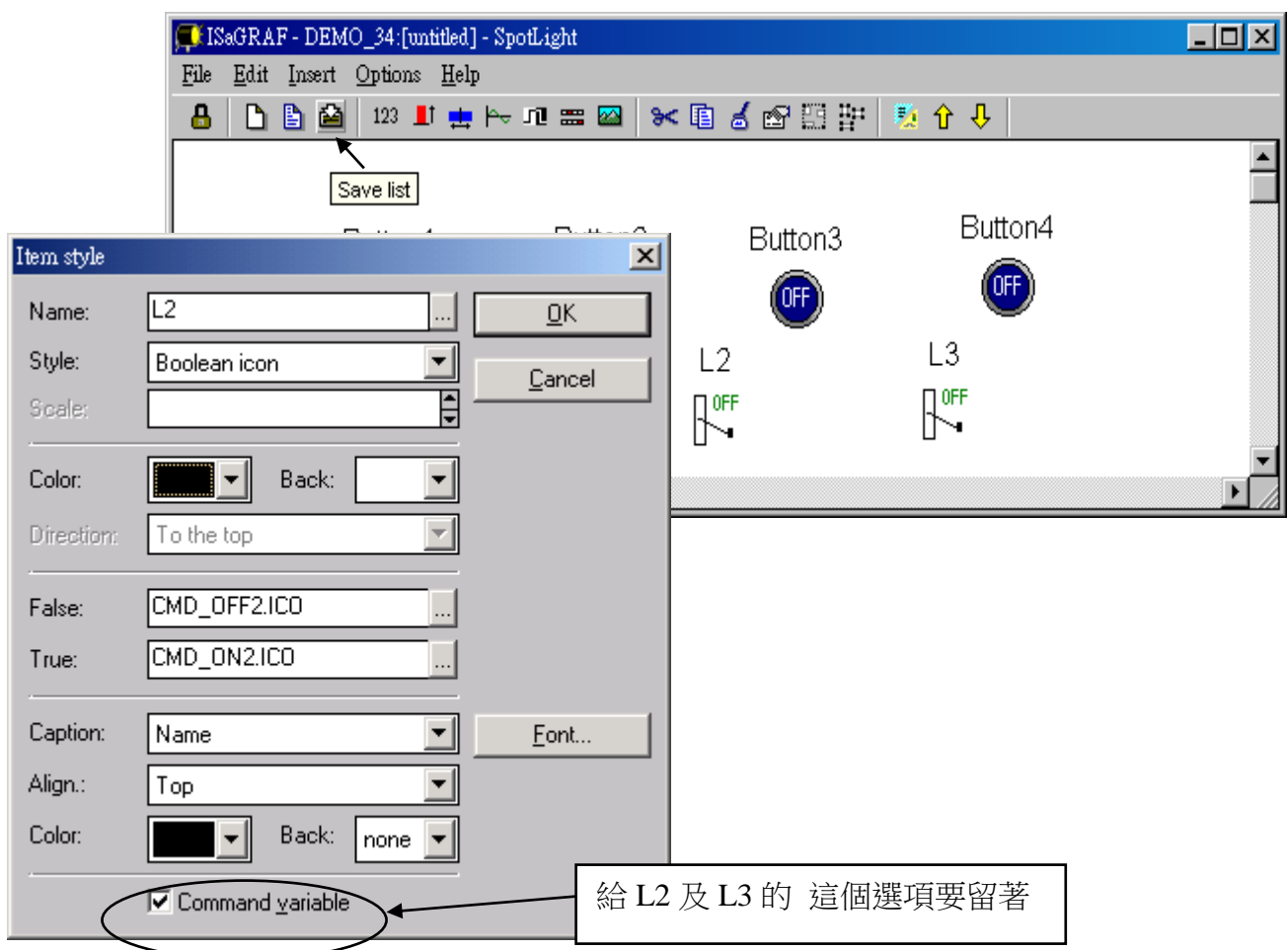
用相同的方法產生另外2個 boolean Icon 如下. 以滑鼠點選"Save list", 取一個名子給這個畫面.



我們還需要另一個 Boolean icon 來顯示“L1”的輸出狀態. 請選用不同顏色的 *.ico 檔 (如 TRUE : “YEL_ON2.ico”, FALSE : “YEL_OFF2.ico”).

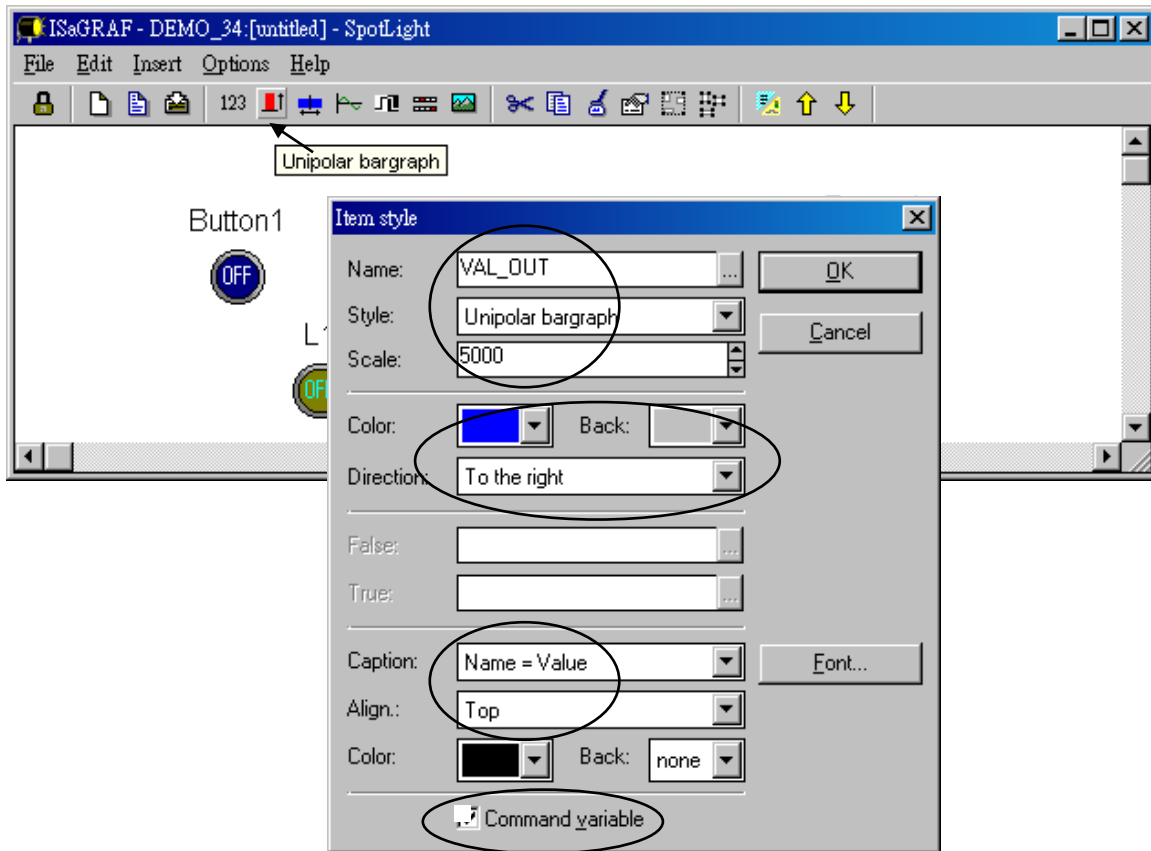


另外再新增 2 個 boolean Icon L2 及 L3 如下(TRUE:”CMD_ON2.ico” and FLASE: “CMD_OFF2.ico”), 給 L2 及 L3 的 “Command variable” 選項要留著. 安全起見, 想到就存.

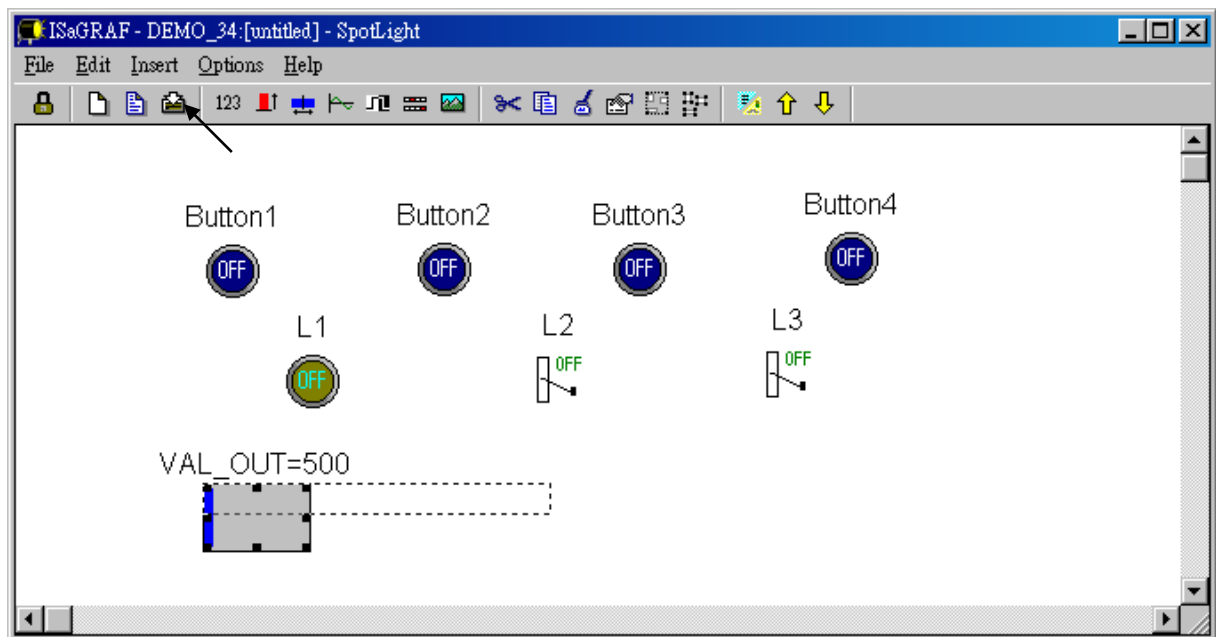


E. 新增 “Unipolar bargraph”

點選“Unipolar bargraph”，給定 Name 為“VAL_OUT”，Scale 為 “5000”，Color 為藍色, Back 為 灰色, Direction 為 “To the right”，Caption 為 “Name=Value”，Align 為 “Top”，並將”Command variable” 選項拿掉。

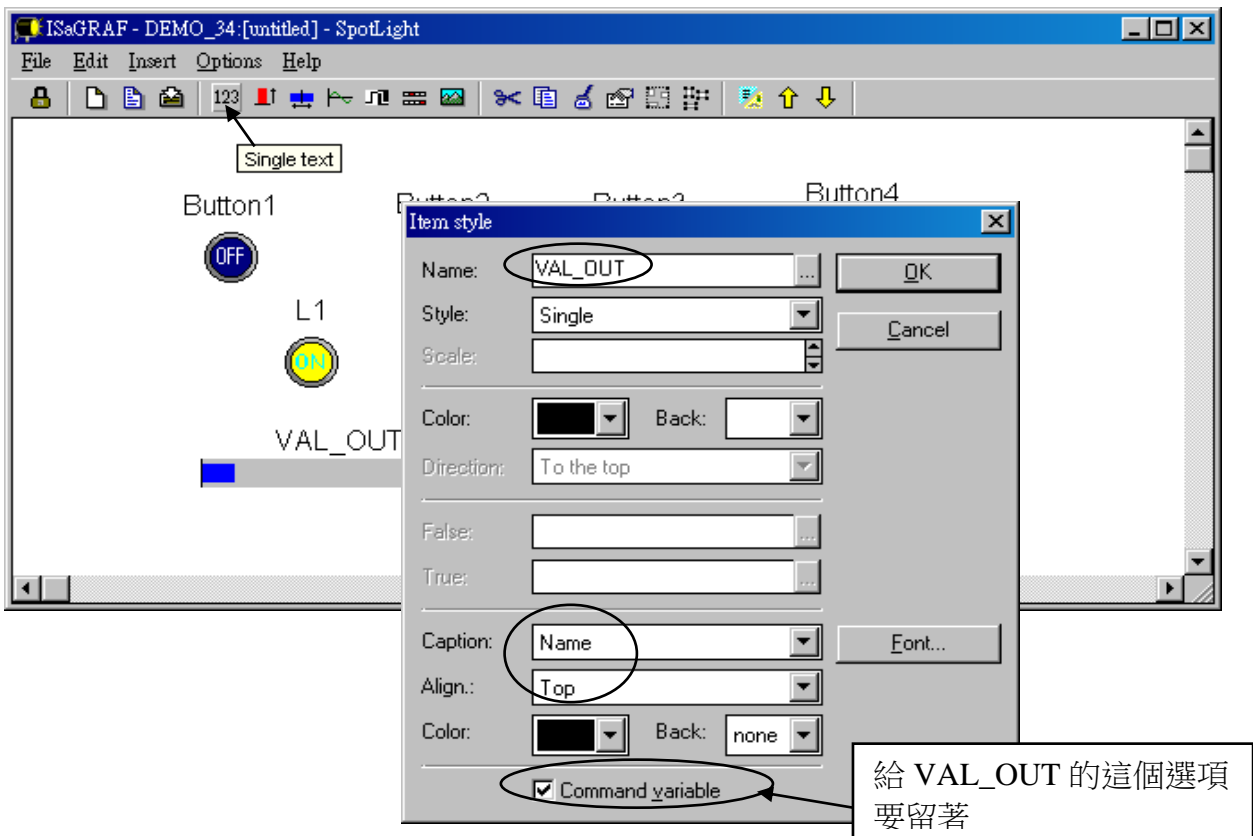


點選滑鼠左鍵並一直按住，改變它的形狀。 安全起見，想到就存。

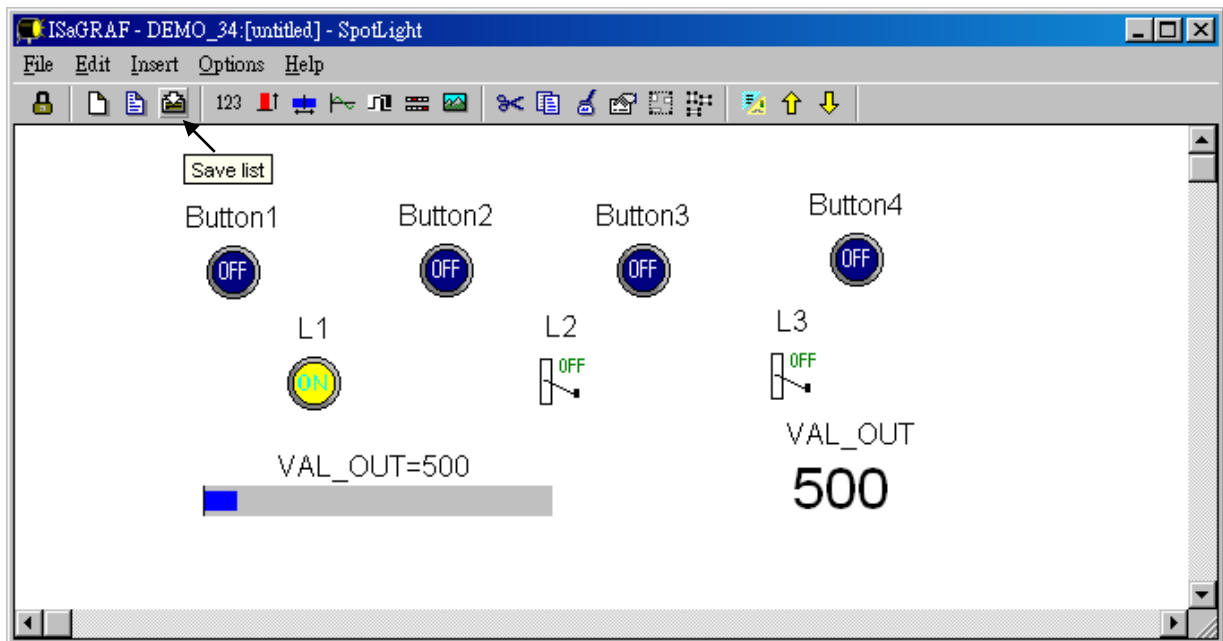


F. 新增 “Single text”

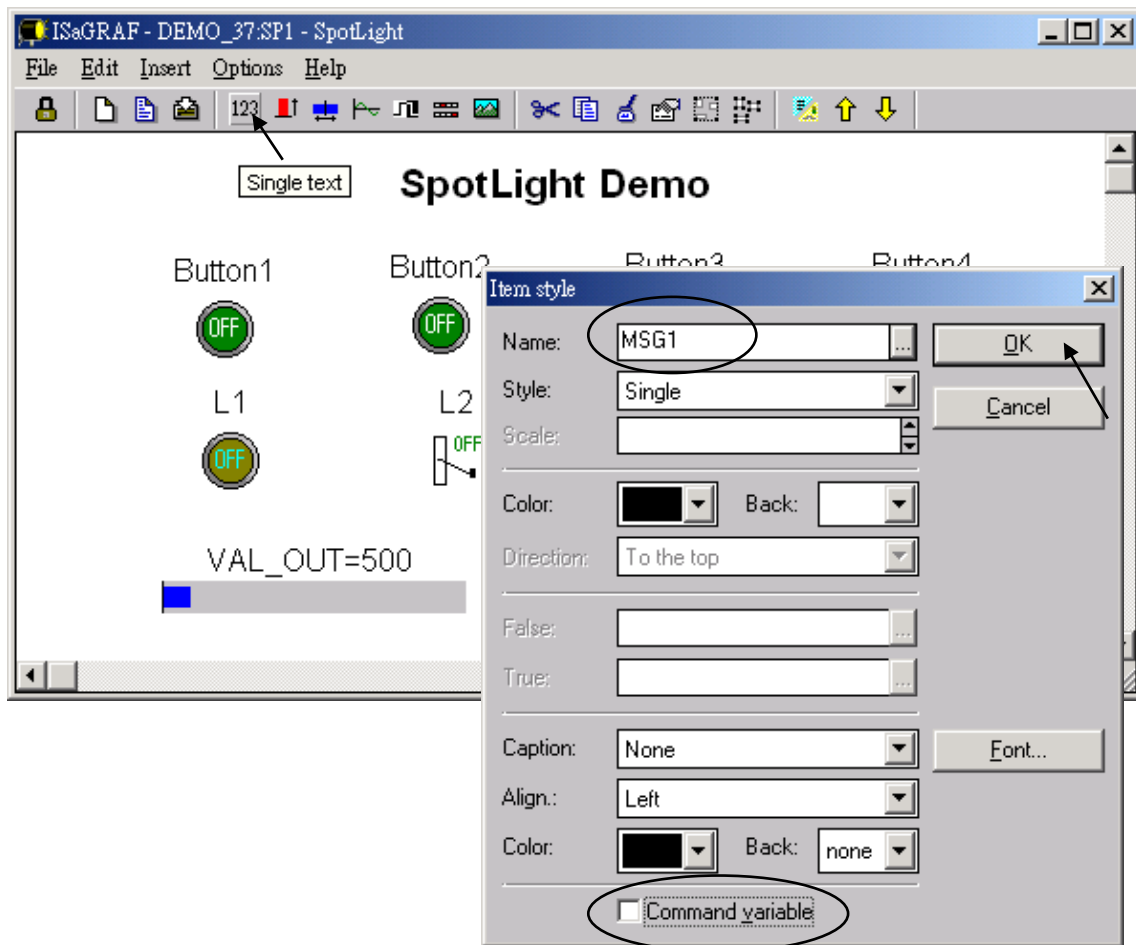
點選“Single text”，給定 Name 為 “VAL_OUT”，Caption 為 “Name”，Align 為 “Top”



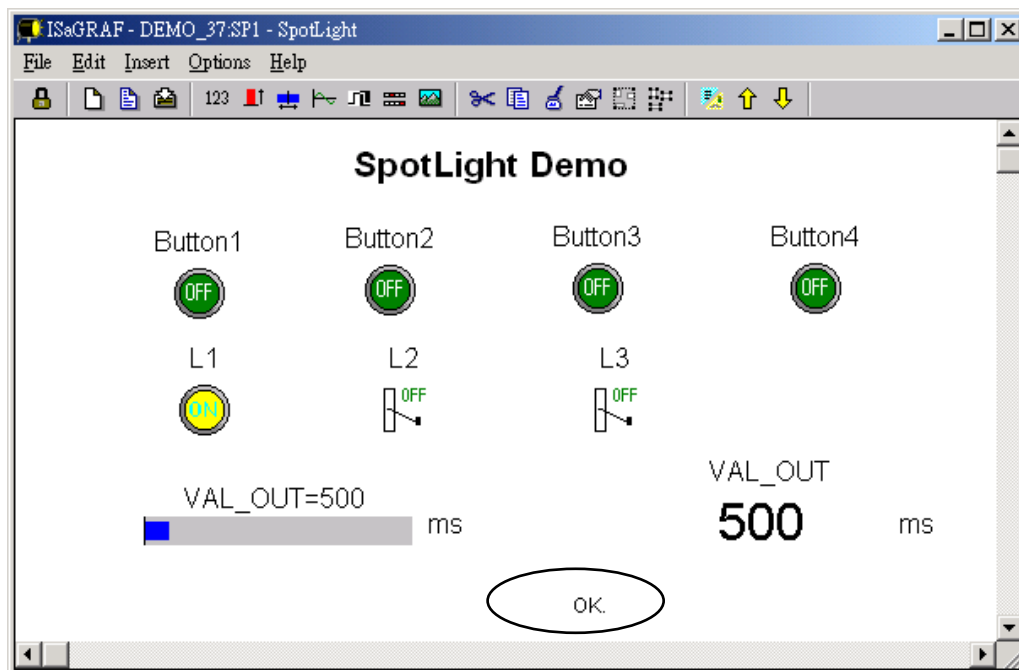
將它移到適當的位置，安全起見，想到就存。



再點選 “Single text”，設定 Name 為 “MSG1”，Caption 為 “None”，Align 為 “Left”，並將”Command variable” 選項拿掉。

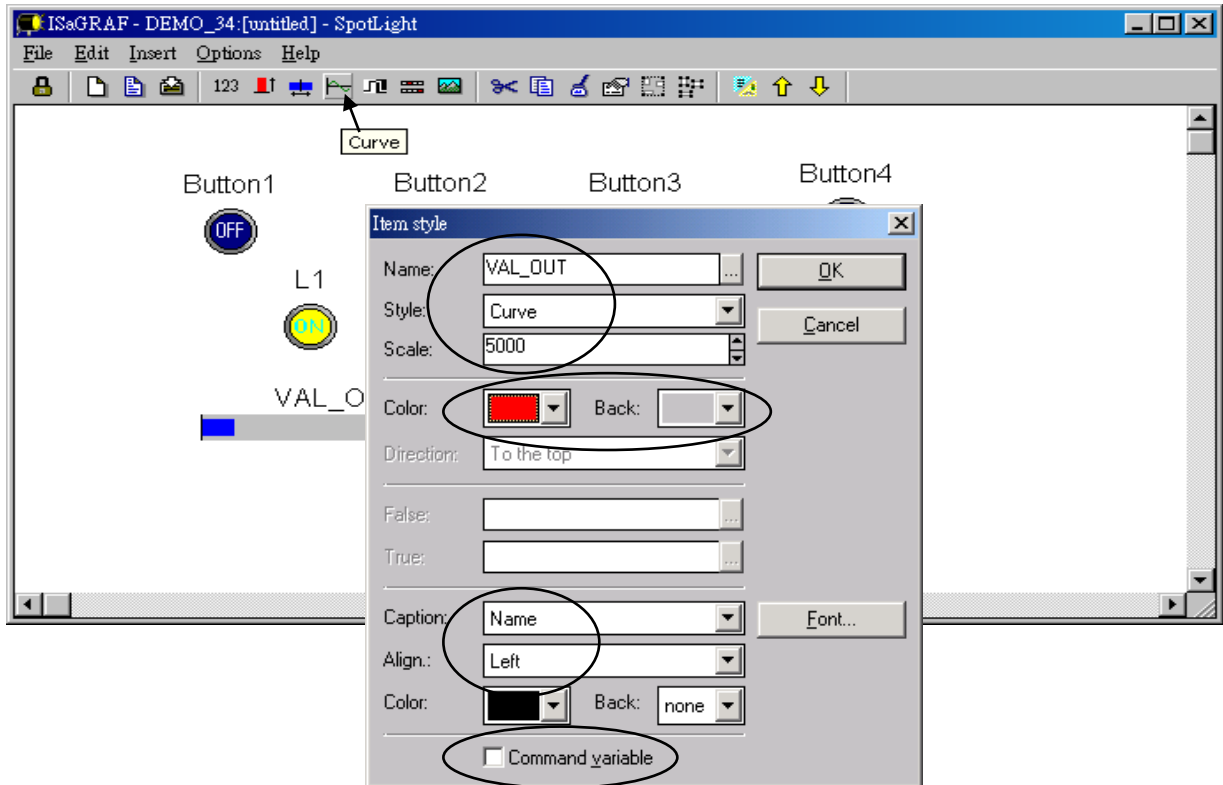


移到適當的位置後儲存。

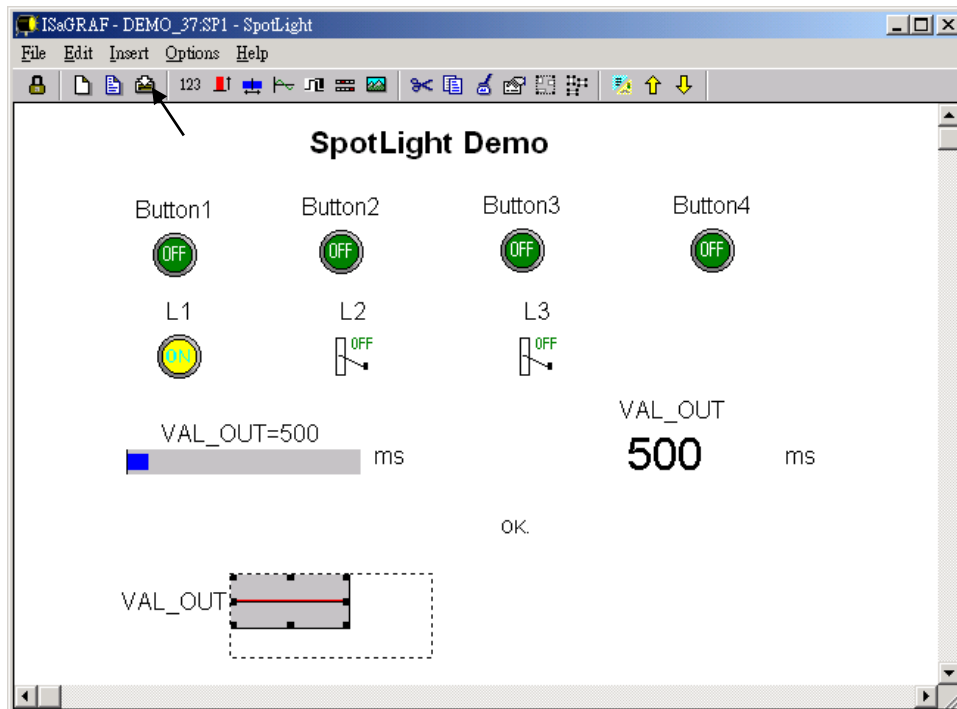


G. 新增 “Curve”

點選 “Curve”，給定 Name 為 “VAL_OUT”，Scale 為 “5000”，Color 為紅色 red, Back 為灰色, Caption 為 “Name”，Align 為 “Top”，並將”Command variable” 選項拿掉。



點選滑鼠左鍵並一直按住，改變它的形狀。安全起見，想到就存。

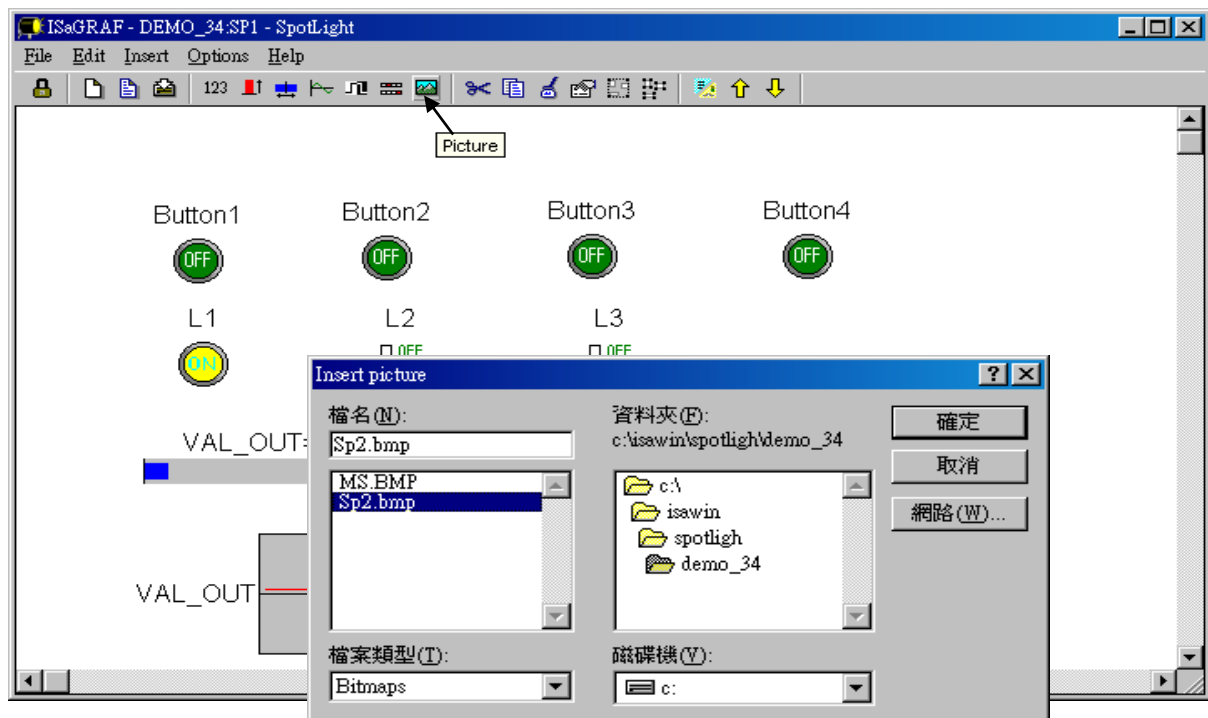


H. 新增 “picture”

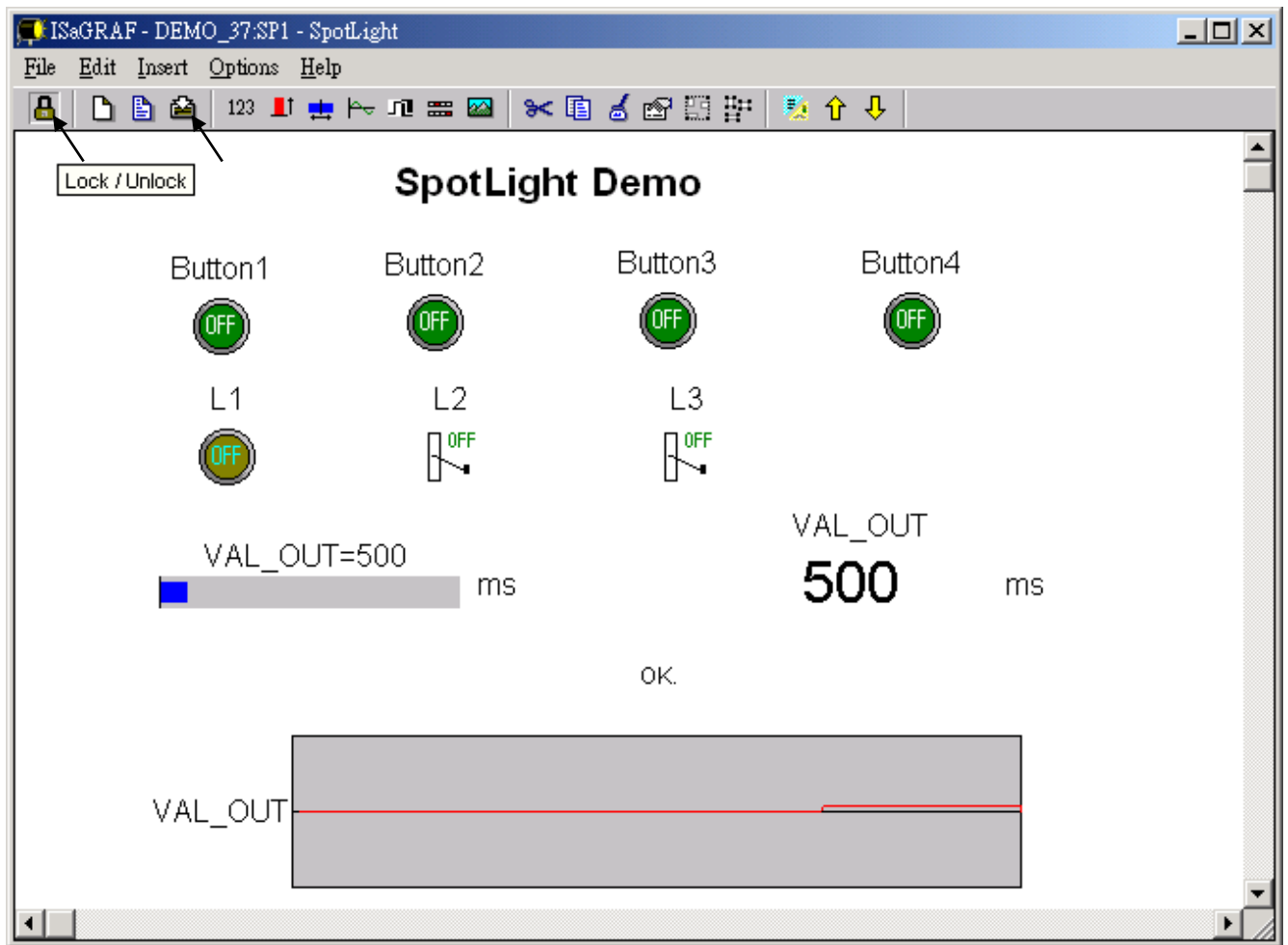
請用 MS 小畫家建立 2 個圖檔如下. 分別取名為“sp2.bmp”及 “ms.bmp”，並存到專案的所在目錄(本例 “c:\isawin\spotligh\demo_37\”)



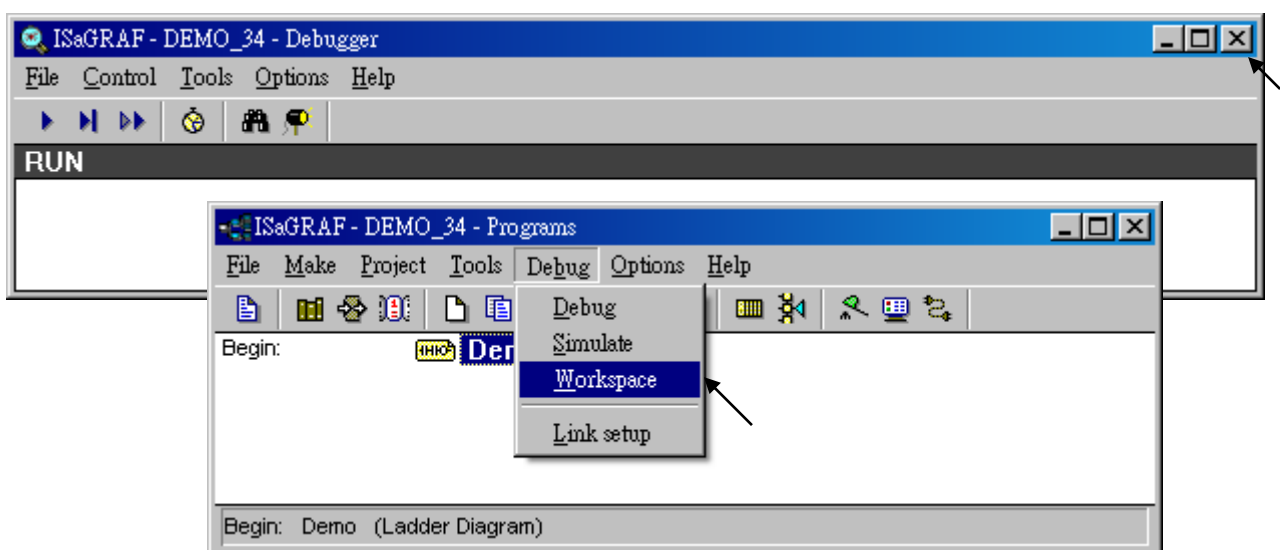
點選 “Picture”，選取要插入的圖檔。



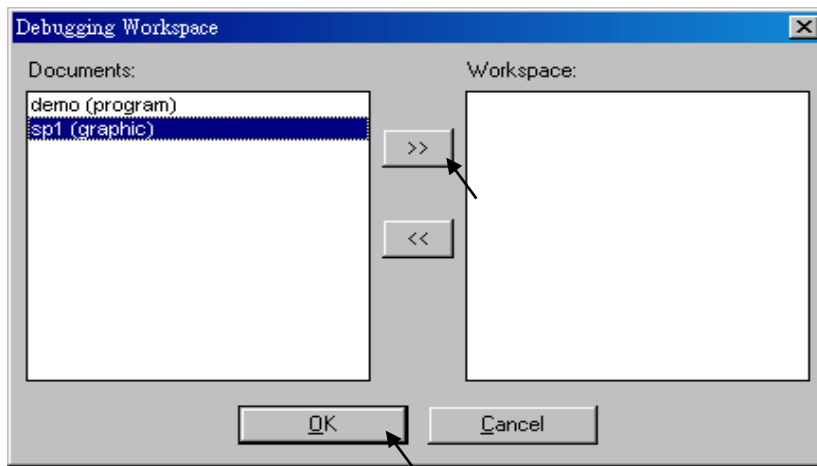
請新增 2 個 pictures “sp2.bmp” 及 “ms.bmp” 到適當的位置, 如下圖.點選 “Lock” 將畫面鎖住 (不能再更改的意思). 記得要存.



- I. 在 “Workspace” 內加入 Spotligh 畫面
離開“simulation”, 然後點選 “Debug” - “Workspace”.

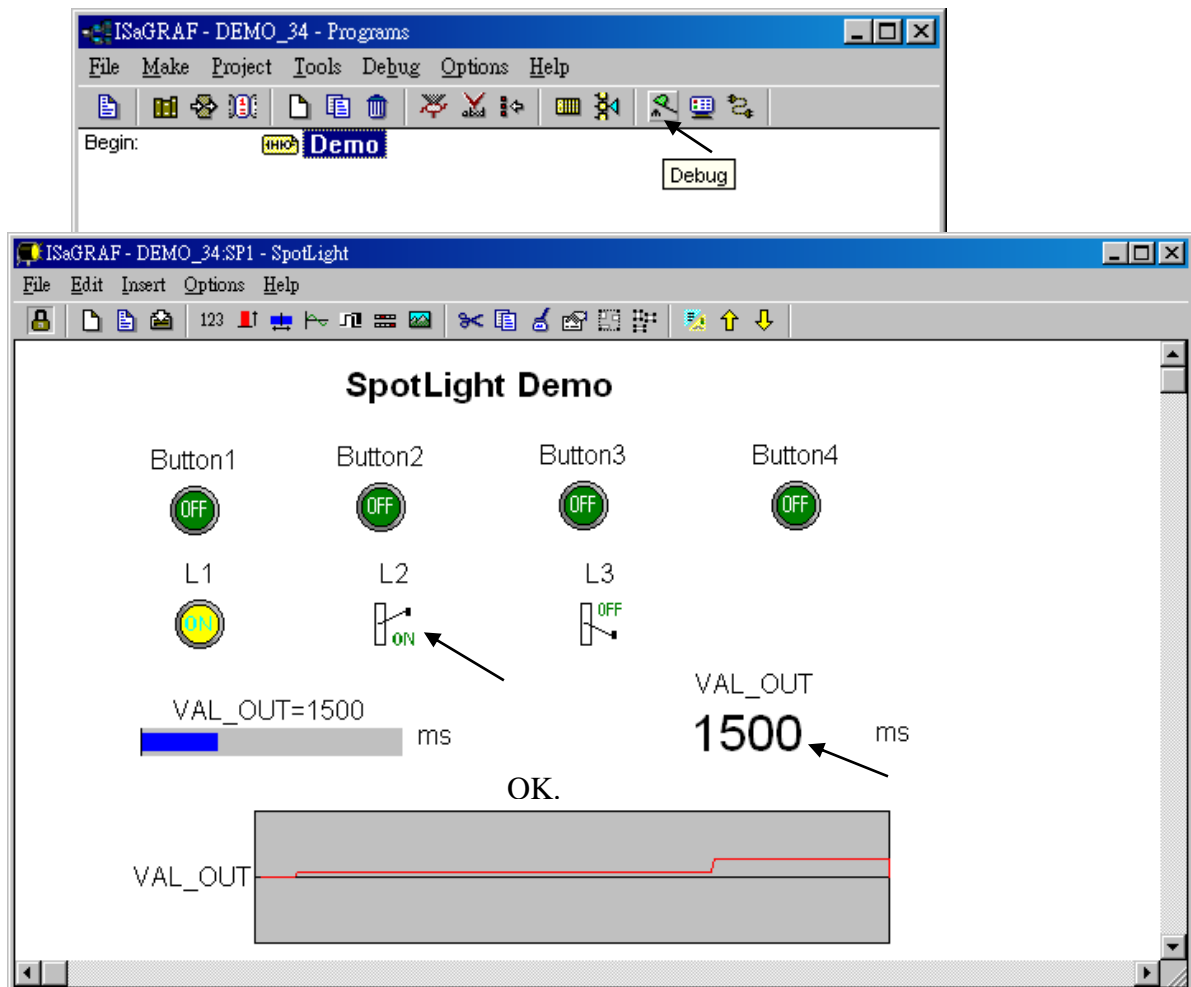


將剛建好的圖控畫面搬到右邊(Workspace), 如此當程式在模擬或連到控制器時, 該圖控畫面會自動跳出來.

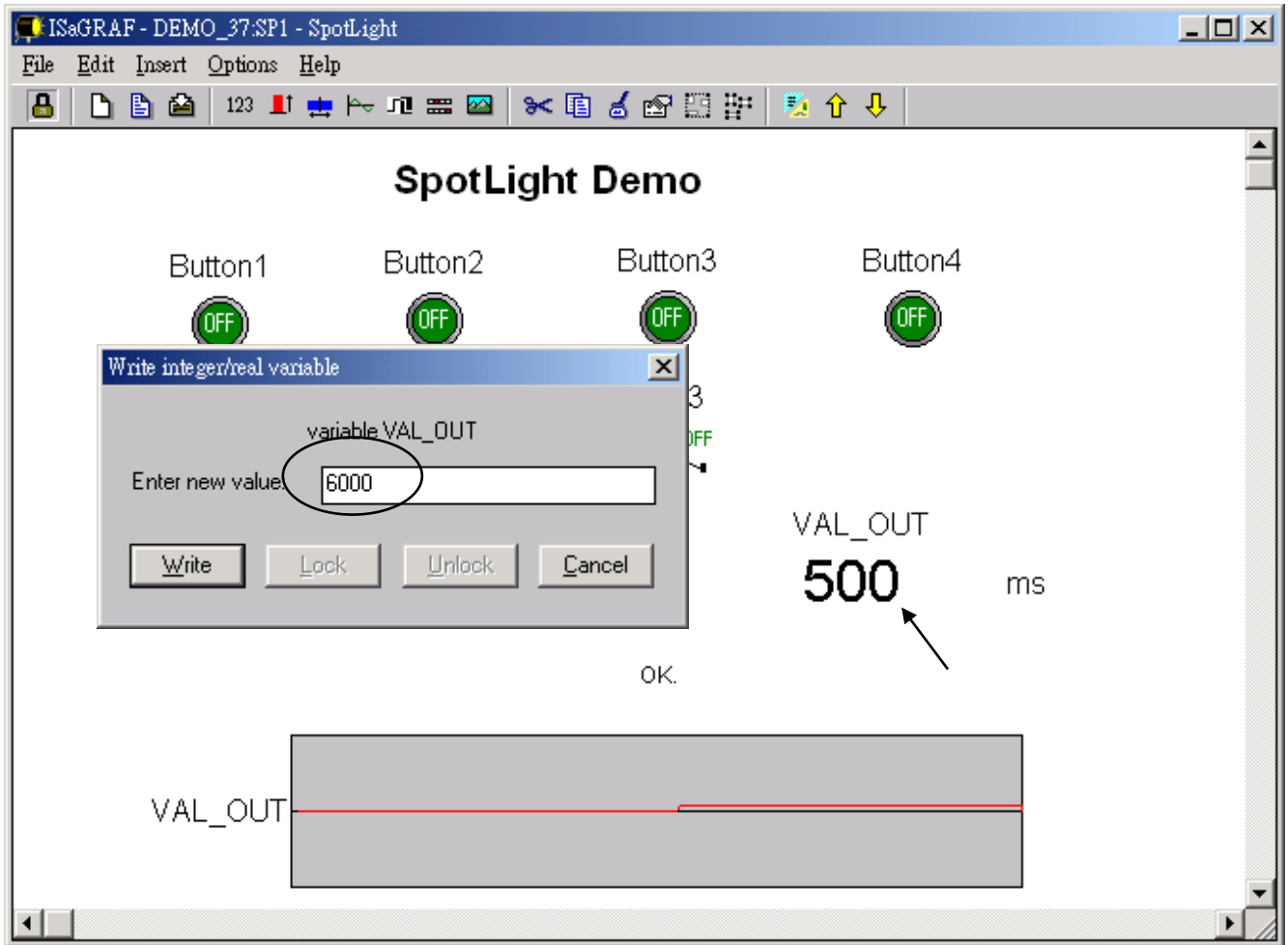


J. 下載到控制器並測試

點選 “Debug” 以下載. 你可以用滑數 雙擊 “L2”, “L3” 或 “VAL_OUT” 去控制它們. 也可按控制器上的面版按鈕, 看畫面如何變化.

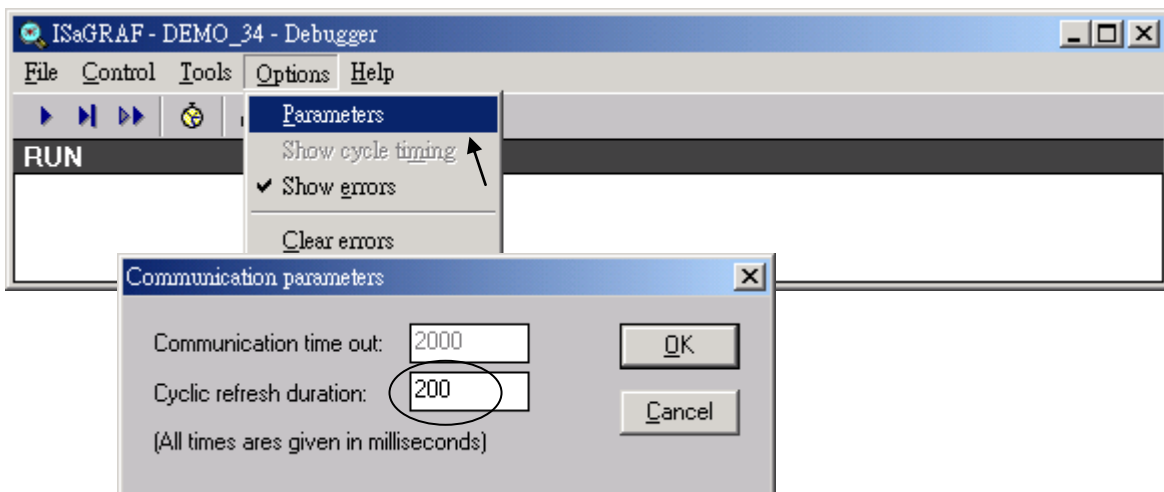


可雙擊”VAL_OUT” 以輸入 1 個超過 5000 的值, 觀看有何反應.



注意:

若要讓通訊快一點反應到圖控畫面上, 可點選 “Options” – “Parameters”, 然後設定 “Cyclic refresh duration” 到一個較小的值. (建議別小於 200 ms)



第 15 章 編寫自行定義的 Function

ISaGRAF 提供給使用者以 ST, FBD, IL 及 QLD 等語法來自行定義 Function. 這些 Function 多半是使用者因自己需要而制定的, 例如某些一直重複用到的邏輯.

Function 一定有 1 個回傳值, 且它的名稱需與 Function 的名稱一樣. Function 最多可有 31 個輸入參數. Function 內的程式不能呼叫 ”功能方塊”, 但可呼叫 ISaGRAF 的標準 **Function** 和 ICP DAS 提供的 **Function** (A.4 節內標記為 Function 的可呼叫, 但標記為 Function Block 的不行)

本章的範例為製作一個自行定義 Function 來判斷何時去寫資料到 EEPROM. 格式如下.

Function 名稱 : W_EEP

說明: 當值有變時, 儲存該整數值到 EEPROM 內.

輸入參數:

ADDR_ (integer): 要存到 EEPROM 內的那個位置

V1_ (integer): 新值

V2_ (integer): 舊值

回傳值:

W_EEP (integer): 回傳回新值

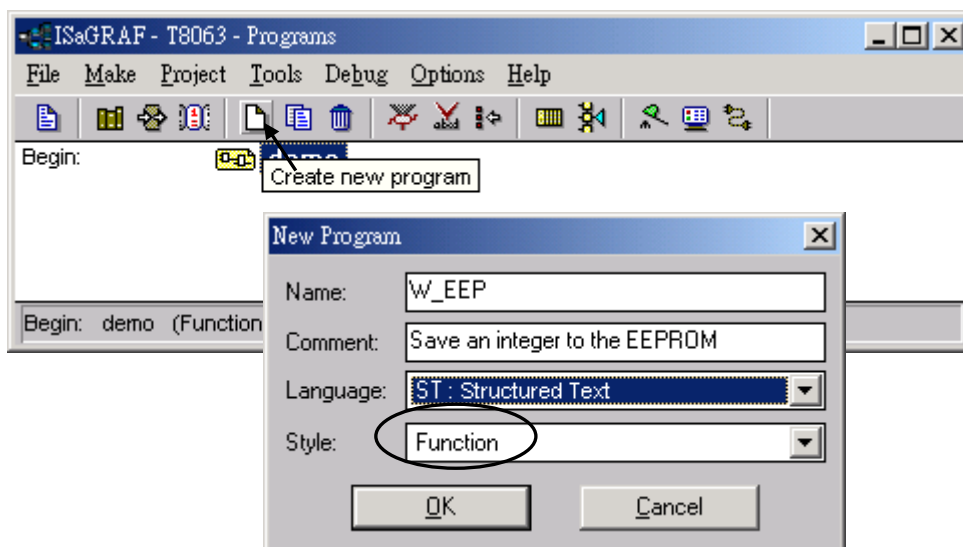


注意: 自行定義的 Function 所宣告的輸入參數將會變成保留字. 這也是為何本例使用 ADDR_, V1_, V2_ 而不用 ADDR, V1 & V2 的原因.

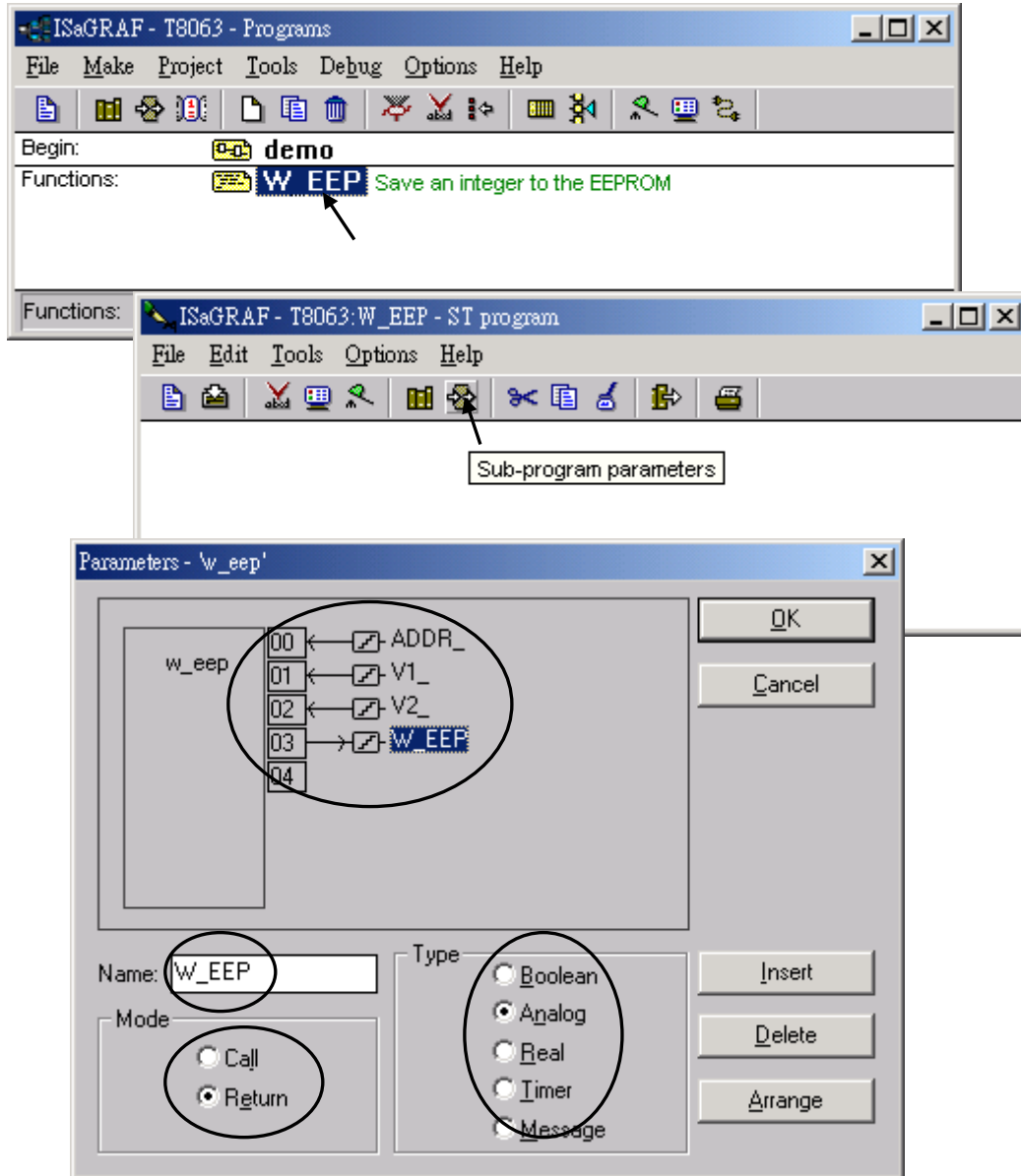
15.1: 在 1 個專案內定義 Function

Function 如果是定義在某專案內, 就只能在該專案內使用. 別的專案不能用它.

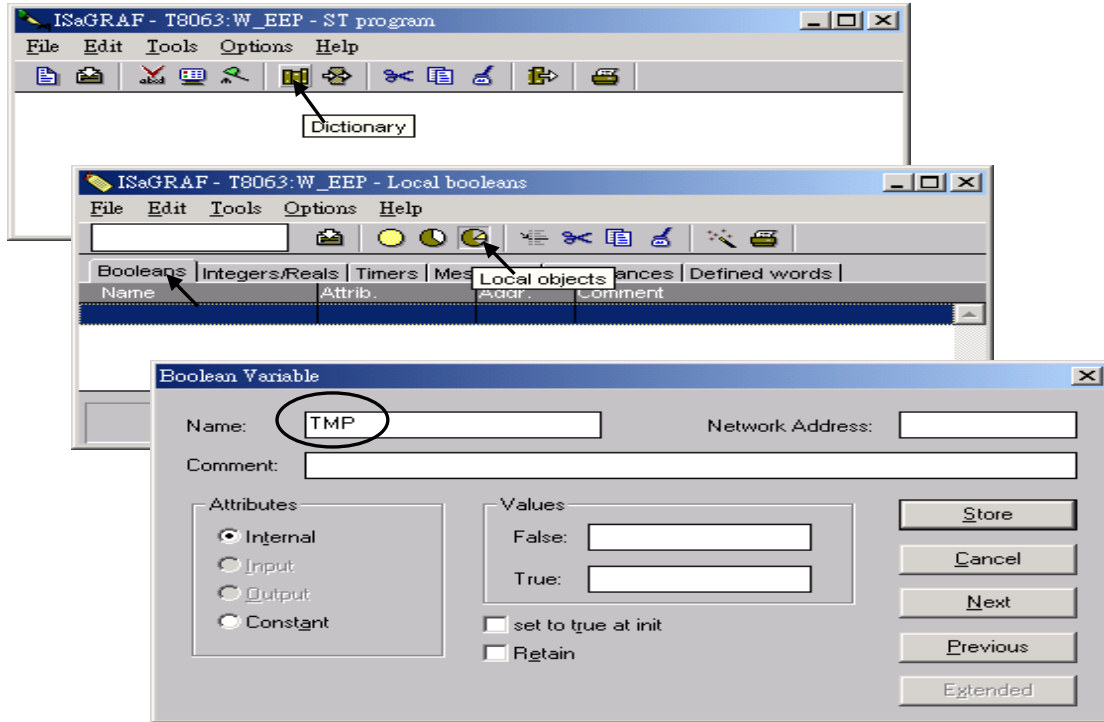
A. 在 1 個專案內點選“Create new program”.取名為“W_EEP”, Language 為 “ST:...”, Style 為 “Function”.



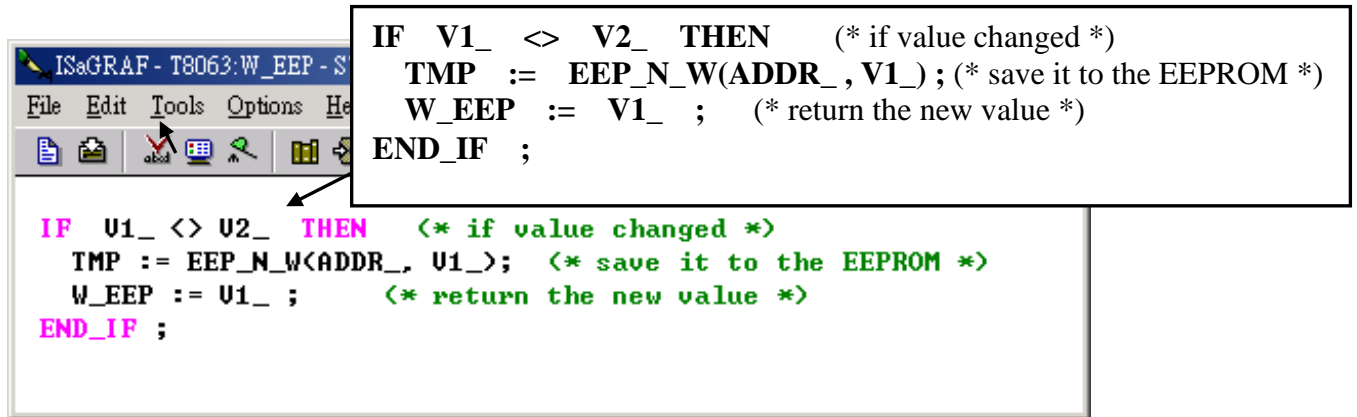
B. 雙擊剛建立的 Function. 進入後, 點選“Sub-program parameters” 以定義輸入參數及回傳值.



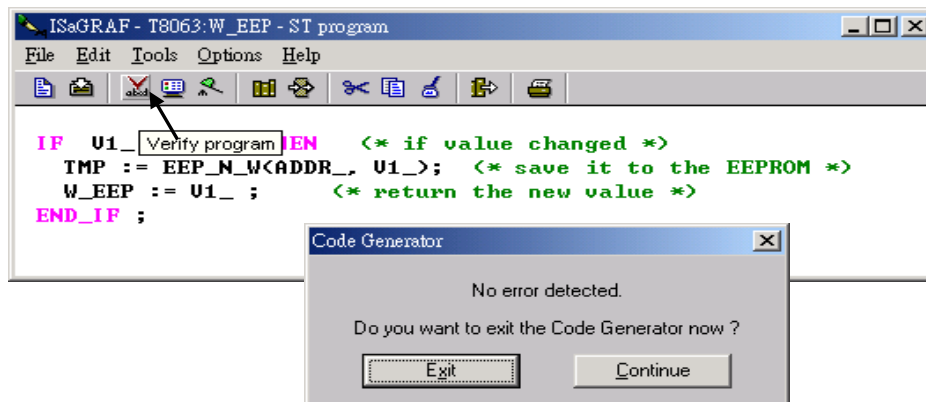
C. 宣告區域變數(只有在本 Function 內使用). 這裡需要 1 個 **boolean internal** 變數 “TMP”.



D. 編寫 Function 內的程式



E. 編譯該 Function.

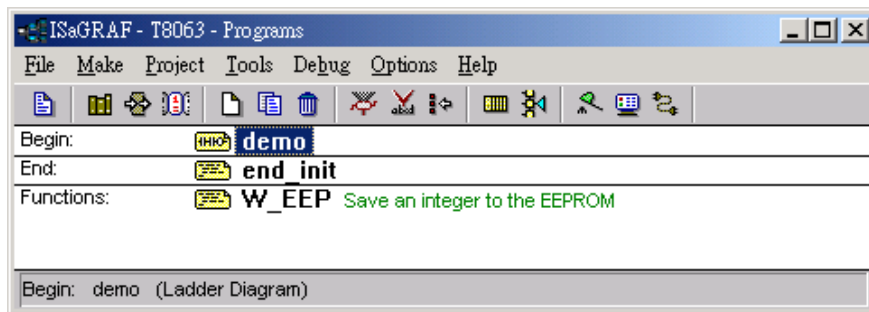


F. 在同個專案的其他程式內呼叫它.

專案內的變數:

Name	Type	Attribute	Description
INIT	Boolean	Internal	初值為 TRUE . TRUE 表示第 1 個 PLC 周期
K1	Boolean	Input	連到 “push4key” 的第 1 個按鈕
New_Val	Integer	Internal	要寫到 EEPROM 的新值
Old_Val	Integer	Internal	舊值
Val	Integer	Internal	EEPROM 的讀回值

專案程式架構:

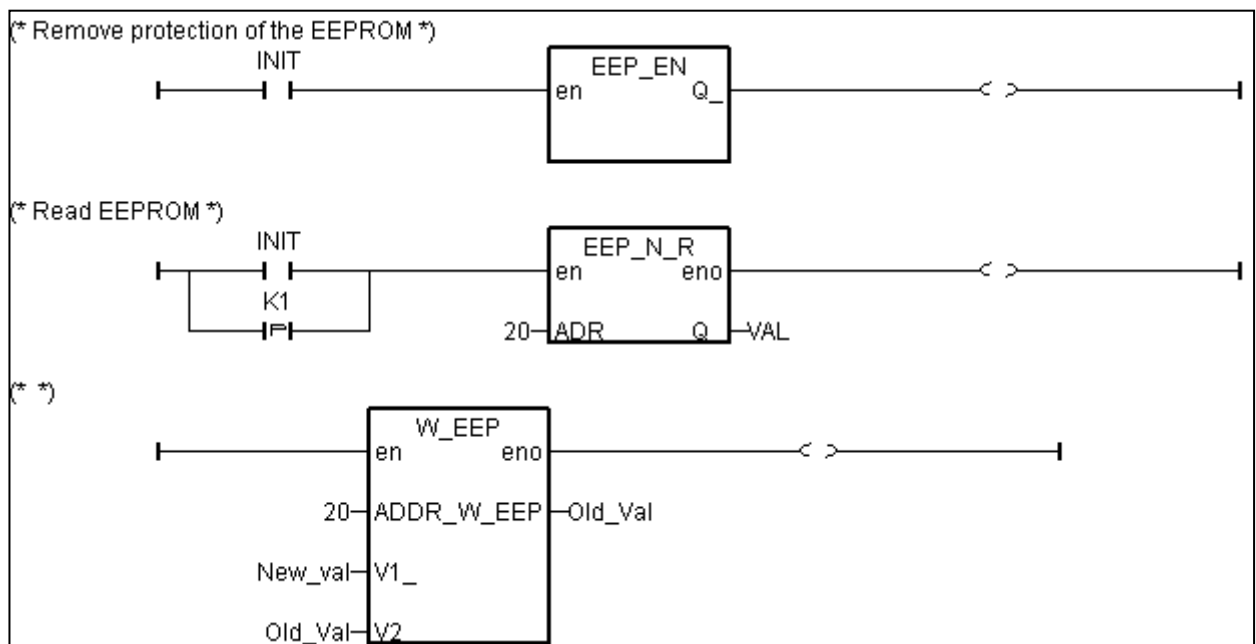


ST 程式 – “end_init” 在 “End” 區 :

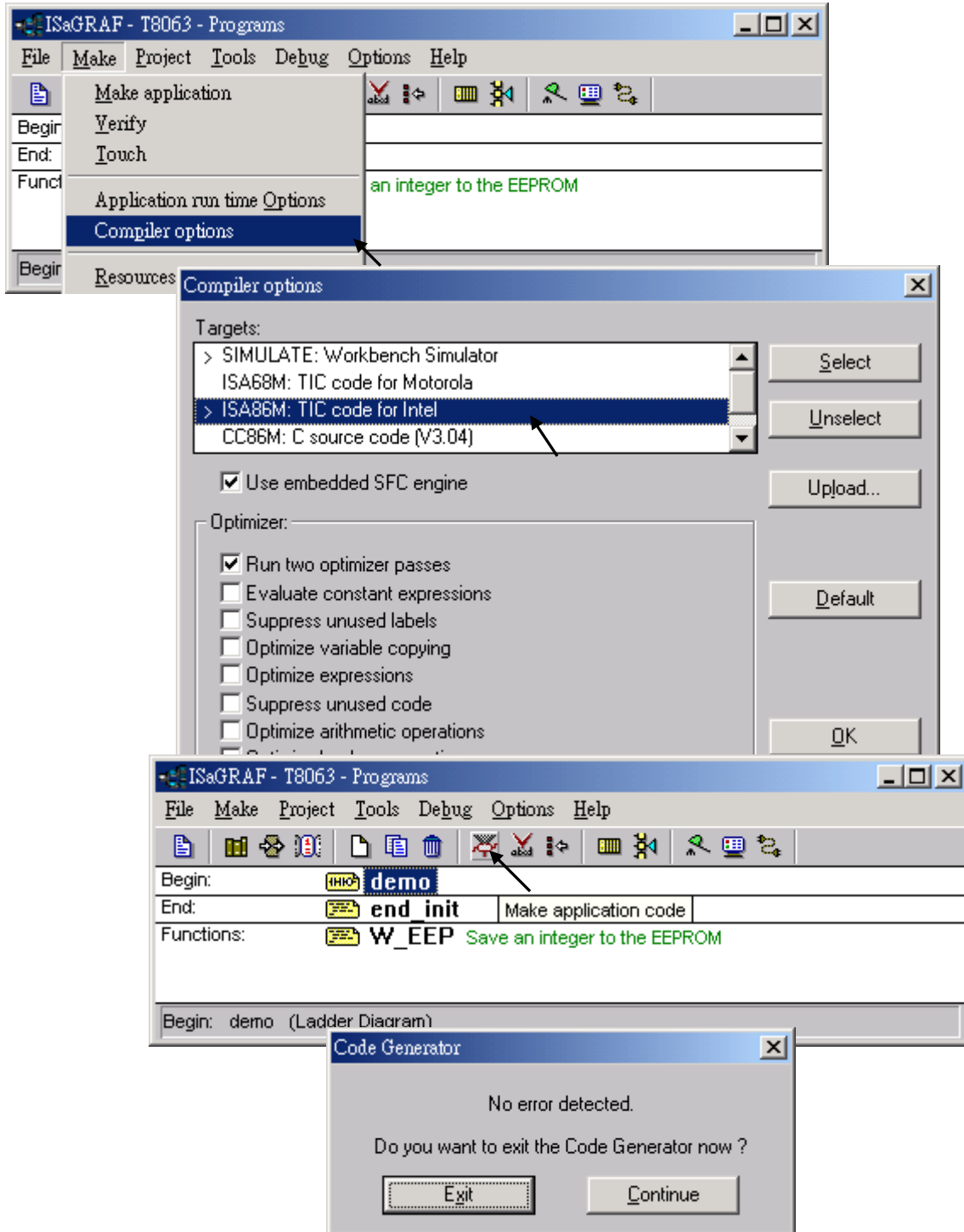
```

IF INIT=TRUE THEN
    INIT := FALSE ;
END_IF ;
    
```

LD 程式 – “demo” :



G. 設定 Compiler Options, 然後編譯本專案.



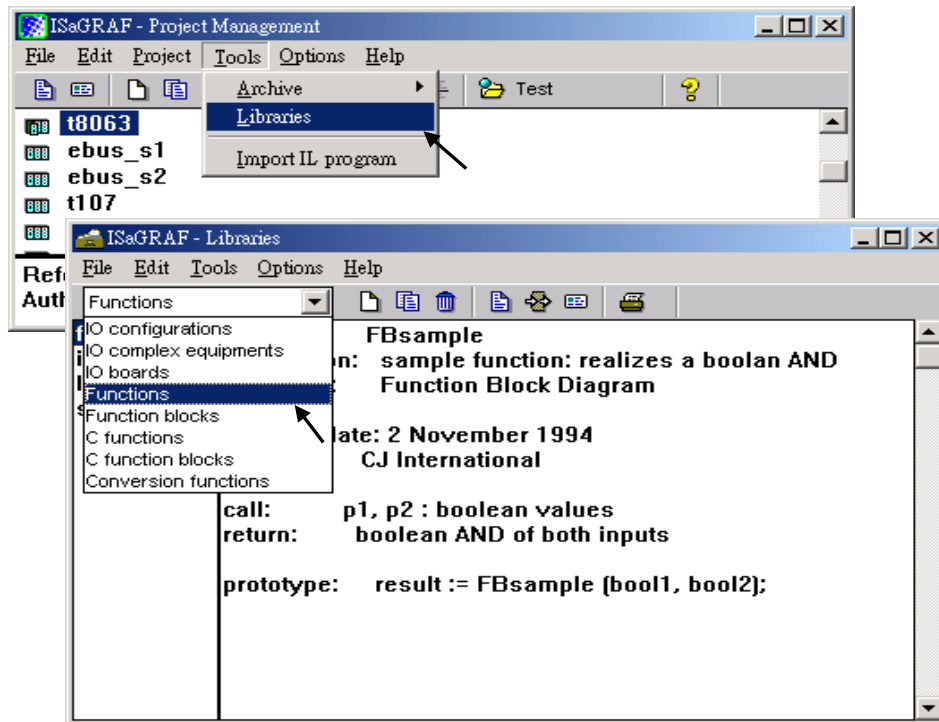
若編譯成功, 下載到控制器內, 你可以在 ISaGRAF 內更改 “New_Val” 的值, 然後按控制器上的第 1 個按鈕, 從 EEPROM 內讀出它剛寫入的的值.

15.2: 在 ISaGRAF Library 內定義 Function

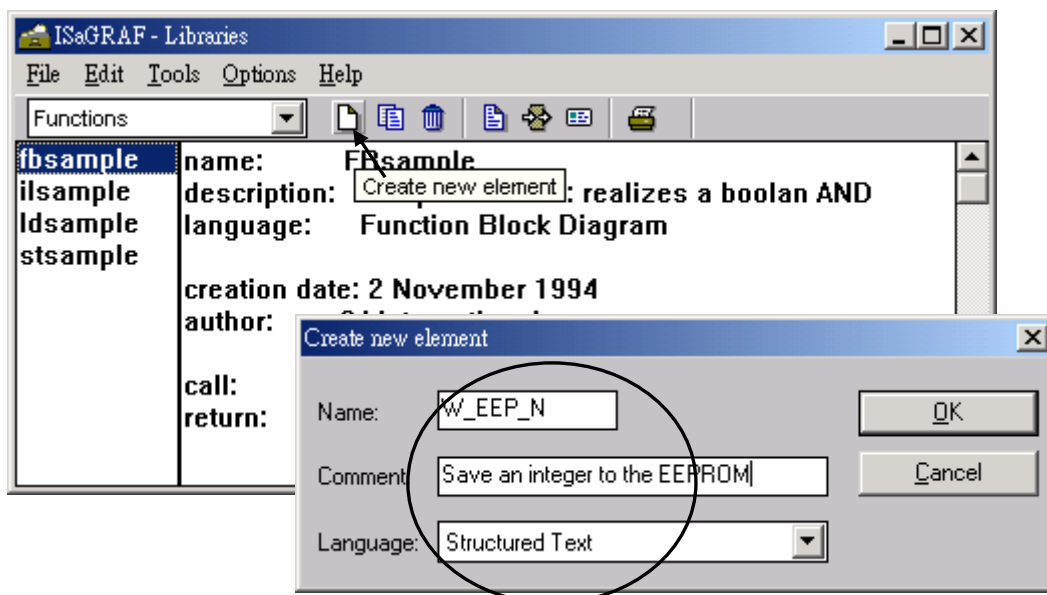
定義在 ISaGRAF 內的 Function 可被任何專案使用。

方法與前一節 15.1 類似。請先讀一遍。

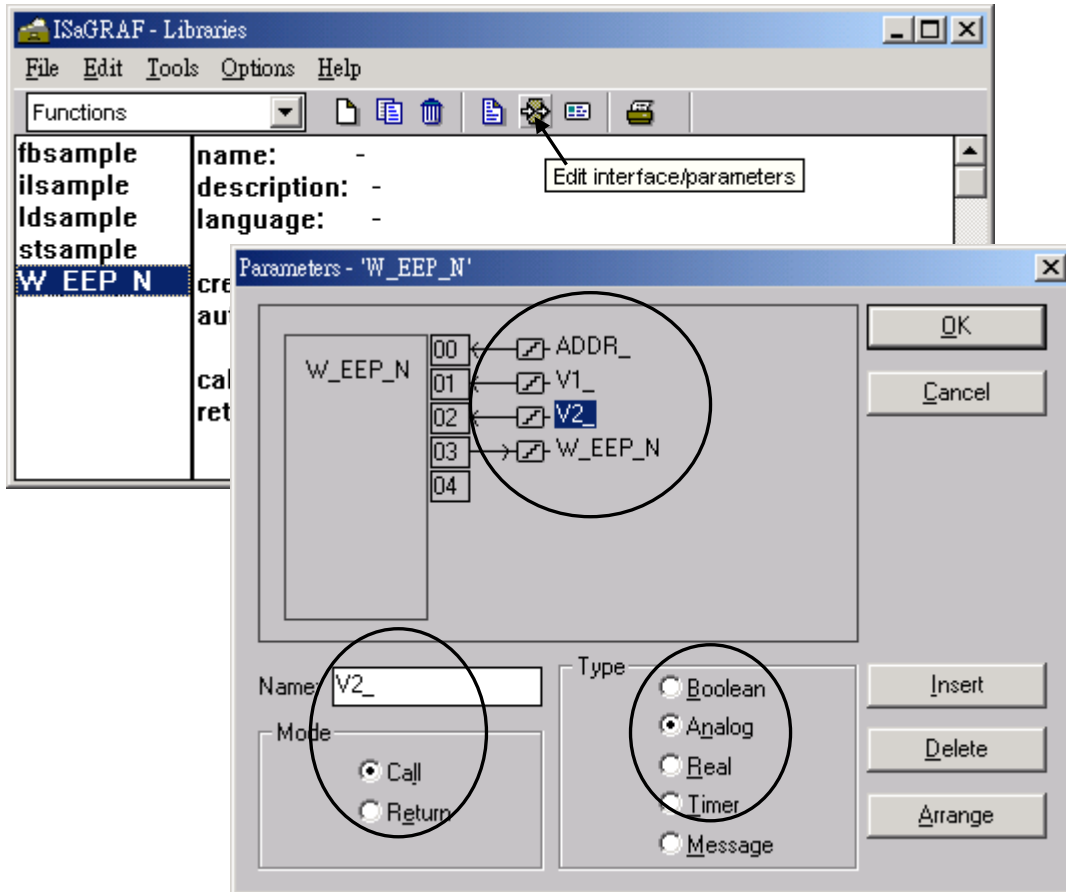
A. 進 Library. 之後點選 “Functions”



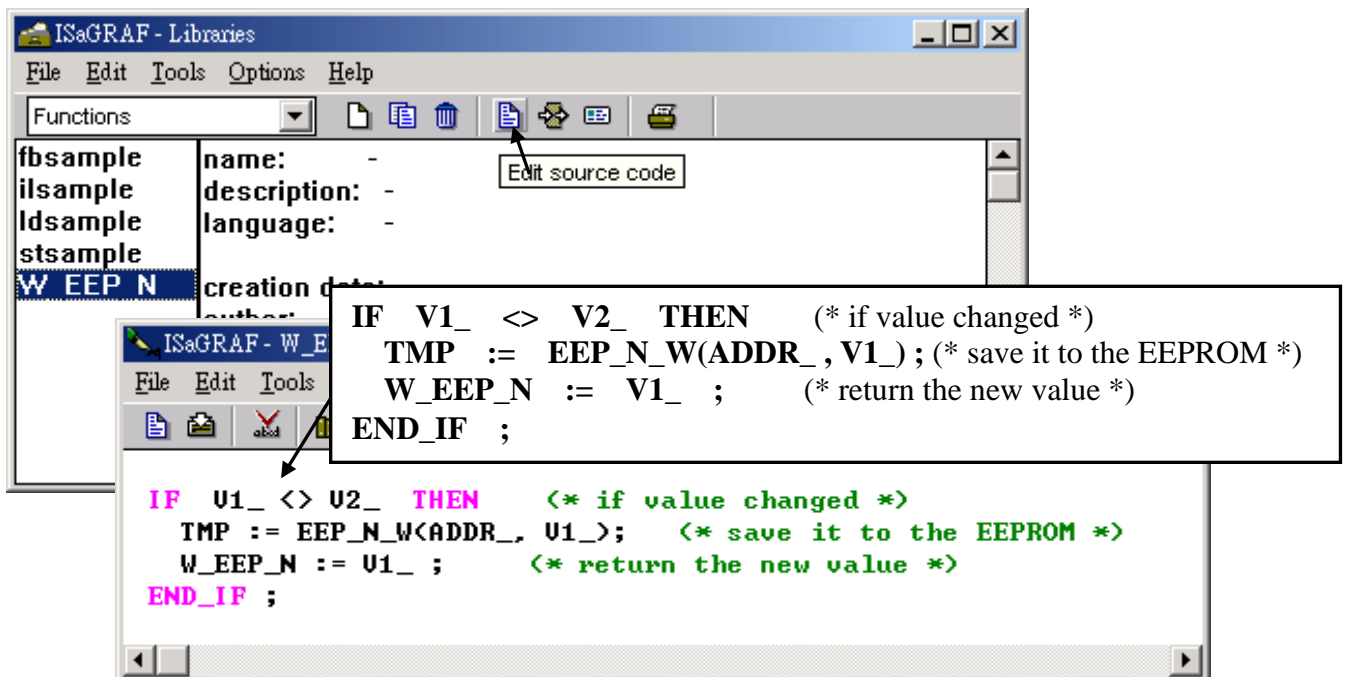
B. 建立新 function, 取名為 “W_EEP_N”, Language 為 “Structured Text”.



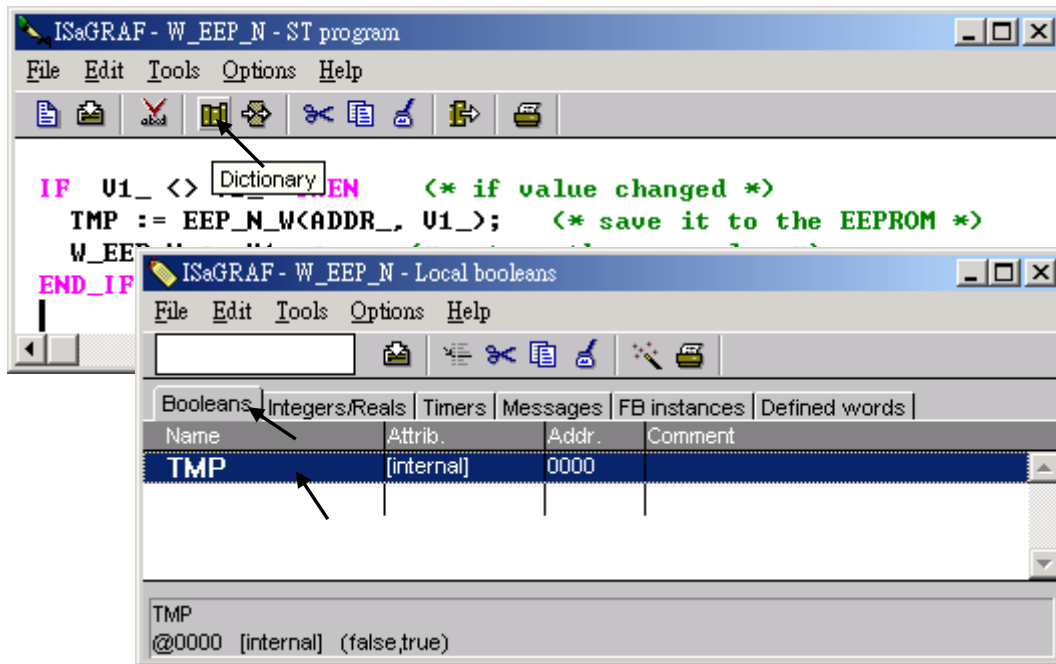
C. 定義輸入參數與回傳值



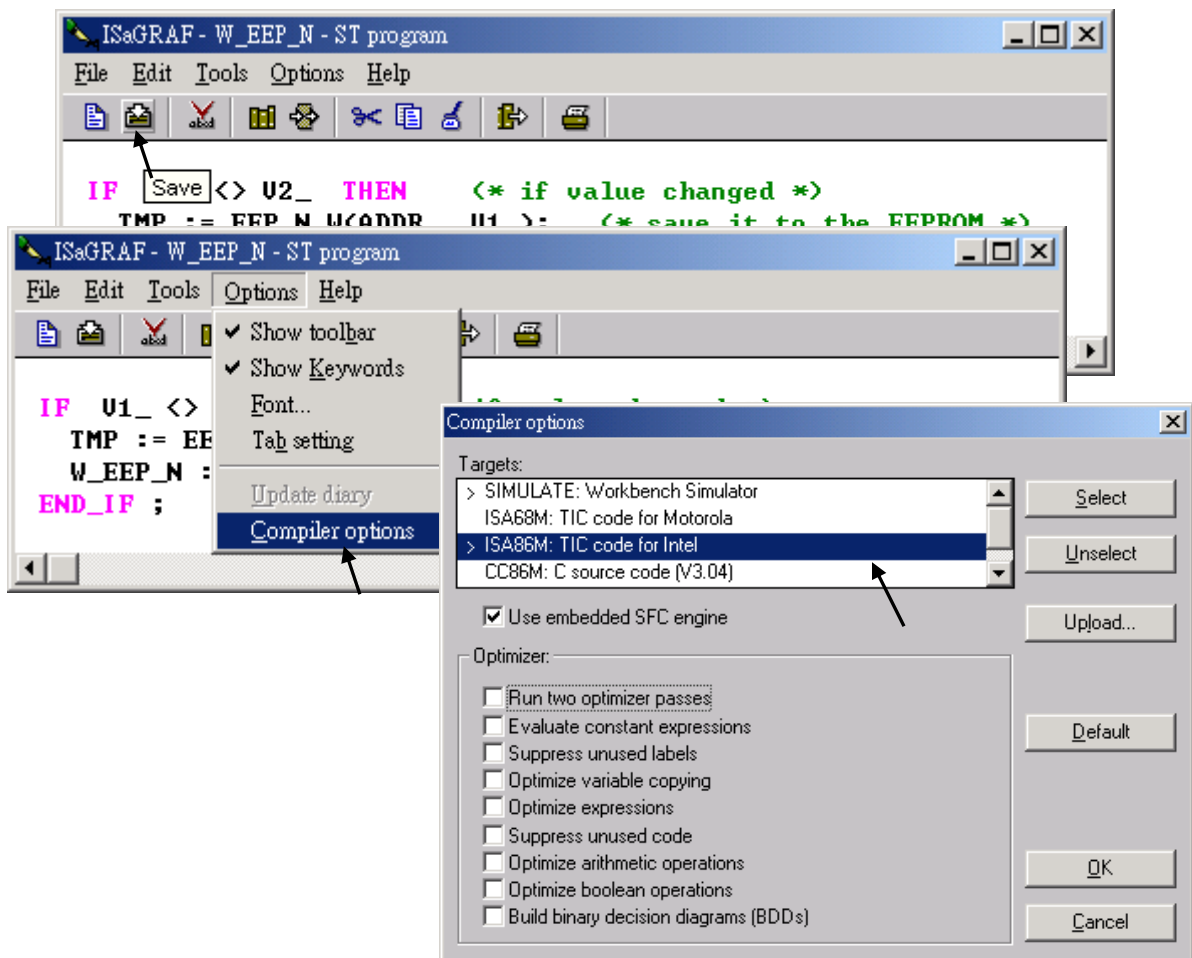
D. 編寫程式碼.



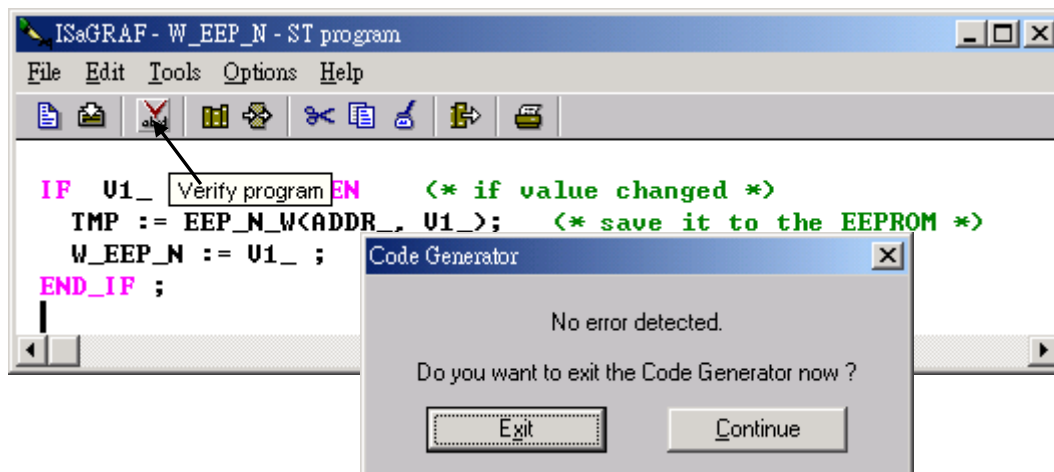
E. 宣告區域變數. 這裡需要 1 個 boolean internal 變數 “TMP”



F. 儲存並設定”Compiler options”.



G. 編譯本 Function.



若成功, 之後寫的任何專案(Project) 都可呼叫它.

第 16 章連接 MMICON

I-8417/8817/8437/8837, I-7188EG/XG 及 Wincon-8xx7 控制器可連接至 ICP DAS 出產的 MMICON 來當成人機介面. MMICON 內含 1 個 240 x 64 點的 LCD 與 1 個 4 x 4 的鍵盤. 可用來顯示圖形, 字串, 整數, 實數, 也可輸入按鍵, 字串, 整數, 實數. 其控制程式使用 ISaGRAF 來寫出.

16.1: 硬體安裝

請參閱與 MMICON 硬體包裝在一起的 “MMICON Hardware Manual”, 以取得更多 MMICON 硬體方面的資訊. 或 http://www.icpdas.com/products/HMI/touch_lcd/man_machine_list.htm

1. MMICON 內含 1 個 COM 埠. 請調成 RS232 埠. (調整硬體上的 “J7” 及 “J8”). 與 MMICON 包裝在一起的那條 RS232 通訊線可直接用來連接 I-8417/8817/8437/8837 的 COM4 與 MMICON 的 CN3.

RS232 接腳說明:

I-8417/8817/8437/8837: 可選用 COM3 或 COM4

Wincon-8xx7: 可選用 COM2

I-8xx7 (COM4) W-8xx7(COM2)	MMICON (CN3) RS232	I-8xx7 (COM3) RS232	MMICON (CN3) RS232
2 RXD	_____	2 TXD	_____
3 TXD	_____	3 RXD	_____
5 GND	_____	5 GND	_____

I-7188EG/XG: 只能選 COM3 :RS232. (COM3 需另選購 X503 ~ X51x 子板)

I-7188EG/XG (COM3) RS232	MMICON (CN3) RS232
RXD	_____
TXD	_____
GND	_____

2. 請設定 MMICON 硬體上的 ”J2” 到 “INIT” 位置. I-8417/8817/8437/8837, I-7188EG/XG 及 Wincon-8xx7 只支援 MMICON 的通訊參數 “9600, 8, N, 1” 及 “address = 0” .

注意:

如果要用 Wincon-8xx7 的 COM2 來連接 MMICON, 請先參閱它的 “快速上手手冊” 來關掉 COM2 原先的 Modbus RTU 的功能.

16.2: 建立 MMICON 的背景圖形

請參閱與 MMICON 硬體包裝在一起的“MMIDOS Software User Manual”，以取得更多 MMICON 軟體方面的資訊。

可建立的背景圖形數量與所使用的 ROM 記憶體有關。例如使用 EPROM “27040”，最多可建立 256 頁背景圖形；若為 “27020”，則為 128 頁；若為 “27010”，則為 64 頁。

注意：可使用在 MMICON 上的記憶體為 ROM/ EPROM/ EEPROM/ FLASH。

請從 CD-ROM 上安裝“MMICON”目錄：複製 \Napdos\others\mmicon\ 到你電腦的硬碟上。
或由以下網址內下載“MMICON“ 目錄

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/others/mmicon/>

注意：請變更該目錄內的檔案屬性：取消“唯讀”(Read-only)

使用 Microsoft painter (小畫家) 建立各頁的背景圖形，請參考 MMICON 目錄內“P0.bmp”。

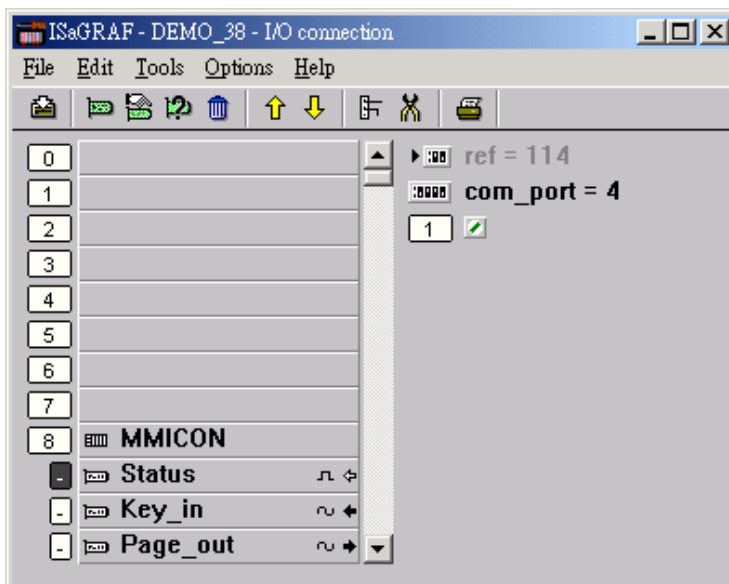
編輯“Autox.dat”檔(請參考 MMICON 目錄內“Auto1.dat”)。此檔必須取消它的“唯讀”屬性。

執行“MMIDOS.exe”來產生“romx.bin”檔，例如“rom1.bin”

使用你手邊有的 ROM 燒錄器將“romx.bin”檔，燒到 ROM 記憶體內。將該 ROM 記憶體插回 MMICON 的 ROM 插槽上，調整“J9”為適合該 ROM 記憶體容量。

16.3: 設計 MMICON 的控制程式

首先在 ISaGRAF 的 ”I/O Connection” 視窗上要連接 I/O complex equipment “mmicon”。可看到它的下方有 3 個子板。



Status 子板:

參數 “com_port” 決定控制器要使用那個 COM 埠來連接 MMICON. 若為 I-8xx7, 可選 3 或 4; 若為 I-7188EG/XG, 只能選 3, 若為 Wincon-8xx7, 請設為 2.

子板上的 1 個虛擬 Digital Input: 若為 TRUE 表示與 MMICON 連線正常, 若為 FALSE 表示與 MMICON 連線失敗.

Key_in 子板:

子板上的 1 個虛擬 Integer Input: 傳回被按下的按鍵代碼. 此代碼值只保留 1 個週期的時間, 之後會回復為 0.

按鍵名稱	按鍵代碼	按鍵名稱	按鍵代碼
0	16#30	Enter	16#0D
1	16#31	.	16#2E
2	16#32	Left	16#1B
3	16#33	Right	16#1A
4	16#34	Up	16#18
5	16#35	Down	16#19
6	16#36	Back space	16#08
7	16#37	F1	16#F1
8	16#38	F2	16#F2
9	16#39	F3	16#F3
A	16#41	F4	16#F4
B	16#42		
C	16#43		
D	16#44		
E	16#45		
F	16#46		

Page_out 子板:

子板上的 1 個虛擬 Integer Output: 改變此值會切換到要顯示的頁畫面.

I-8417/8817/8437/8837, I-7188EG/XG 及 W-8xx7 控制器提供下列函式來控制 MMICON.

MI_BOO	顯示 1 個 boolean 值為 “ON” 或 “OFF”
MI_INT	顯示 1 個 Integer 值
MI_REAL	顯示 1 個 Real 值
MI_STR	顯示 1 個 字串
MI_INP_N	輸入 1 個 integer
MI_INP_S	輸入 1 個 字串
REAL_STR	轉換 Real 值為字串
STR_REAL	轉換字串為 Real 值

請參閱第 11 章的 demo_38, dem_39.

第 17 章 SMS: 收發手機簡訊

ISaGRAF 控制器可用 RS232 通訊埠連至 1 個 GTM-201-RS232 (GSM Modem) 來支援 SMS: 簡訊收發功能. 如此使用者可用手機傳送簡訊給控制器來詢問資料或控制, 此外控制器也可發送簡訊到使用者的手機上來傳遞 警報 或 資料. 以下是不同 PAC 型號可用來連接到 GTM-201-RS232 的 COM port 編號.

I-8xx7: 4 或 5	I-7188EG / 7186EG: 1 或 3 或 4	I-7188XG: 3 或 4
iP-8xx7: 4 或 5	VP-2117: 3 或 5	
XP-8xx7-CE6: 5 或 6	VP-25W7 / VP-23W7: 3 或 5	WP-8xx7: 4 或 5

注意: PAC 若是 WP-8xx7, VP-25W7 / 23W7 與 XP-8xx7-CE6, 建議不要使用 GTM-201-RS232, 改成使用 I-8212W (GPRS/GSM 板卡) 較好, I-8212W 可以插在 這些 PAC 最左邊的 I/O slot, WP-8xx7 與 VP-25W7/23W7 是 slot 0, XP-8xx7-CE6 是 slot 1. 要啟用 I-8212W, 請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-143. 使用 I-8212W 時, COM port 編號分別是 WP-8xx7-CE6: COM5, VP-25W7 / 23W7: COM5, XP-8xx7-CE6: COM6

17.1: 硬體安裝

I-8417/8817/8437/8837 從 driver 2.24 版起支援 SMS 功能. I-7188EG 則為 1.14 版起. I-7188XG 則為 1.12 版起. 假如你手邊控制器的版本是比較舊的, 請更新到該版本或更高的版本. 新的 driver 可由以下 ICP DAS 的 以下網站下載:

<http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

另外如果你的 I/O library 也是舊的, 請參閱第 1.2 節來更新. 或者你可參閱附錄 A.2 節, 單純只安裝 “C functions” 內的以下項目

SMS_test, SMS_get, SMS_gets, SMS_send, SMS_sts
及 “I/O complex equipment” 內的 SMS.

建議 GSM Modem 使用 GTM-201-RS232. 你可向 ICP DAS 洽購或詢問你當地的廠商. ICP DAS 無法保證其他牌子的 GSM Modem 也可用在 ISaGRAF 控制器上.

注意: 請解除 SIM 卡上的密碼設定, 之後再插入 GSM modem 內.

通訊接腳 :

iP-8xx7(COM4/5) W-8xx7(COM2)	GSM 通訊線接頭 GTM-201-RS232	7188EG/XG:COM3/4 RS232	GSM 通訊線接頭 GTM-201-RS232
2 RXD	———— 2 TXD	RXD	———— 2 TXD
3 TXD	———— 3 RXD	TXD	———— 3 RXD
5 GND	———— 5 GND	GND	———— 5 GND
4 DTR	----- 4 DSR	DTR (or RTS)	----- 4 DSR
7 RTS	----- 7 CTS	DTR (or RTS)	----- 7 CTS

17.2: SMS 範例程式

此範例程式為 I-8xx7 的 demo_43, 請參閱第 11.1 節來安裝它. 可由 ICP DAS 的 FTP 網頁下載.

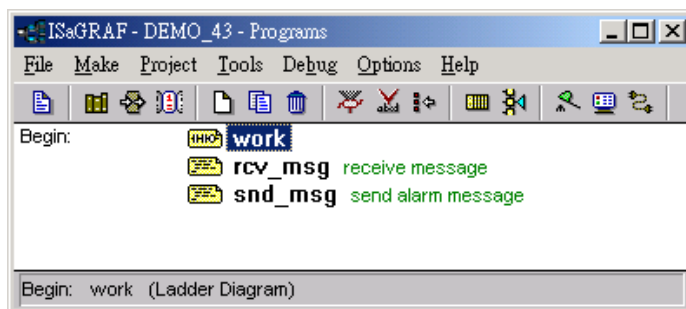
<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

另外 demo_43a 為可以發 SMS 到多隻手機的範例, 一樣可由以上位置取得.

變數宣告 :

Name	Type	Attribute	Description
M1	Boolean	Internal	警報驅動
M2	Boolean	Internal	回報訊息驅動
K1	Boolean	Input	按鈕 1, 連到 push4key
L1	Boolean	Output	LED 輸出 1, 連到 show3led
L2	Boolean	Output	LED 輸出 2, 連到 show3led
L3	Boolean	Output	LED 輸出 3, 連到 show3led
Q1	Boolean	Internal	偵測有無簡訊傳入
TMP	Boolean	Internal	暫時使用
SMS_available	Boolean	Input	SMS 功能啟動了嗎? 連到 SMS - status
T1	Timer	Internal	L1 ~ L3 的閃爍週期, 初值設為 T#500ms
data	Message	Internal	接收到的簡訊內容
phone	Message	Internal	傳送者的電話號碼
Date_time	Message	Internal	接收到的簡訊的時間(字串格式)
To_who	Message	Internal	警報要發給誰 (電話號碼), 請改其初值為自己的號碼.
Msg_to_send	Message	Internal	要發出的簡訊內容
Year1	Integer	Internal	接收到的簡訊的時間 (整數格式), 年 2001 ~ 2099
Mon1	Integer	Internal	月, 1 ~ 12
Day1	Integer	Internal	日, 1 ~ 31
Wday1	Integer	Internal	星期幾, 1 ~ 7, 7 為星期日
Hour1	Integer	Internal	時, 0 ~ 23
Min1	Integer	Internal	分, 0 ~ 59
Sec1	Integer	Internal	秒, 0 ~ 59
Q1_cnt	Integer	Internal	已收到幾封簡訊, 宣告為 “可保留” (retained)
Msg_status	Integer	Internal	簡訊發送狀態
TMP_v	Integer	Internal	暫時使用

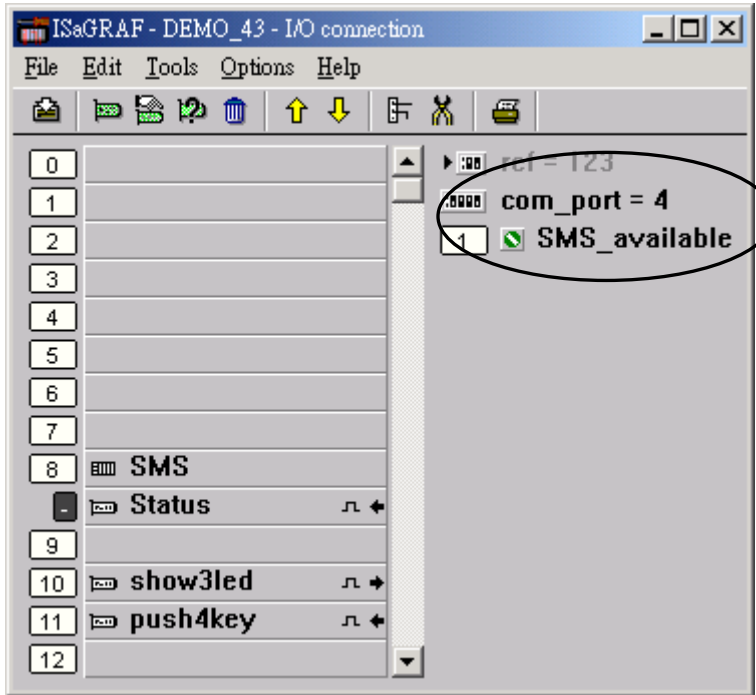
專案架構 :



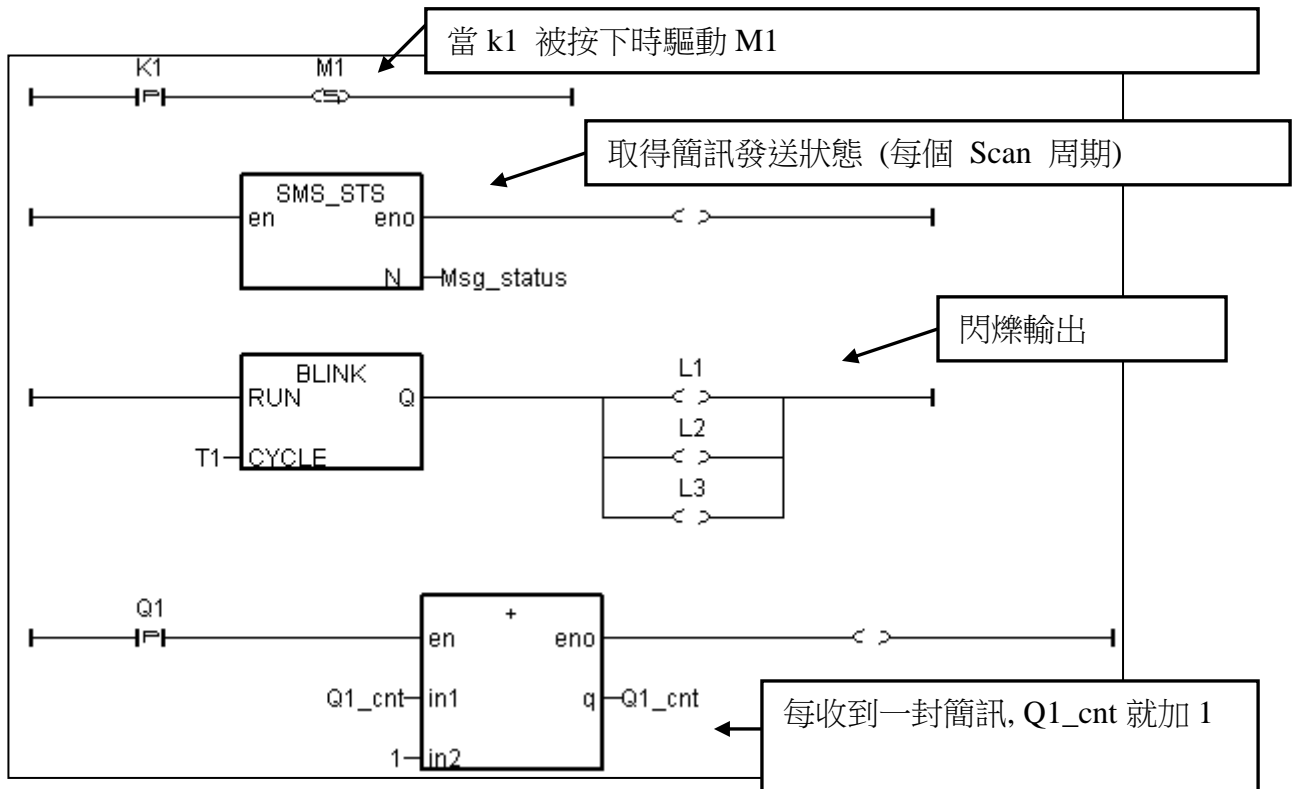
動作說明:

1. 當 K1 被按下時, 一封警報的簡訊會發出.
2. 假如收到簡訊格式, 如 T0200 或 T1500, L1 ~ L3 的閃爍時間會改成 200 及 1500 ms. 並且控制器回發一封簡訊給原發送者.

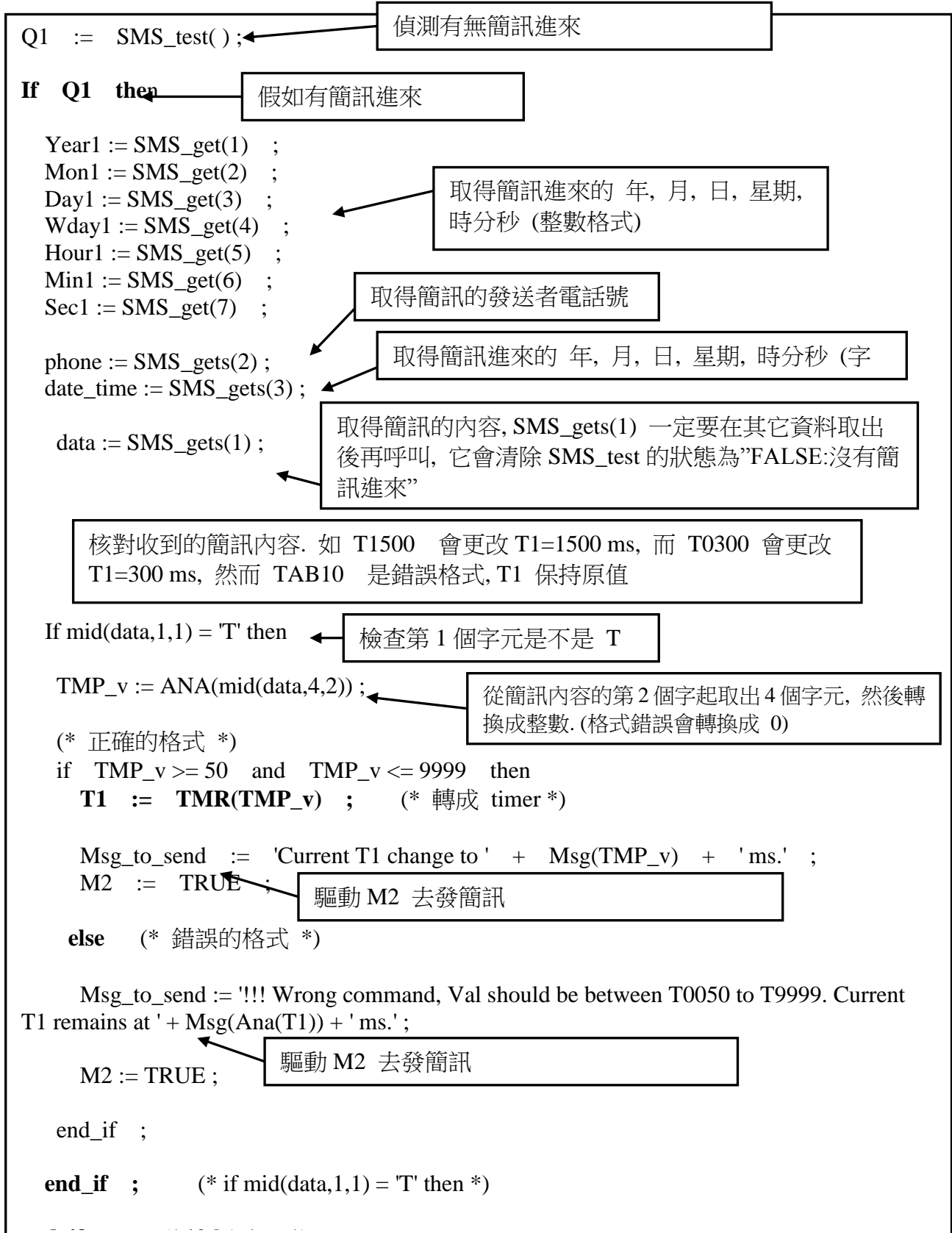
I/O 連結:



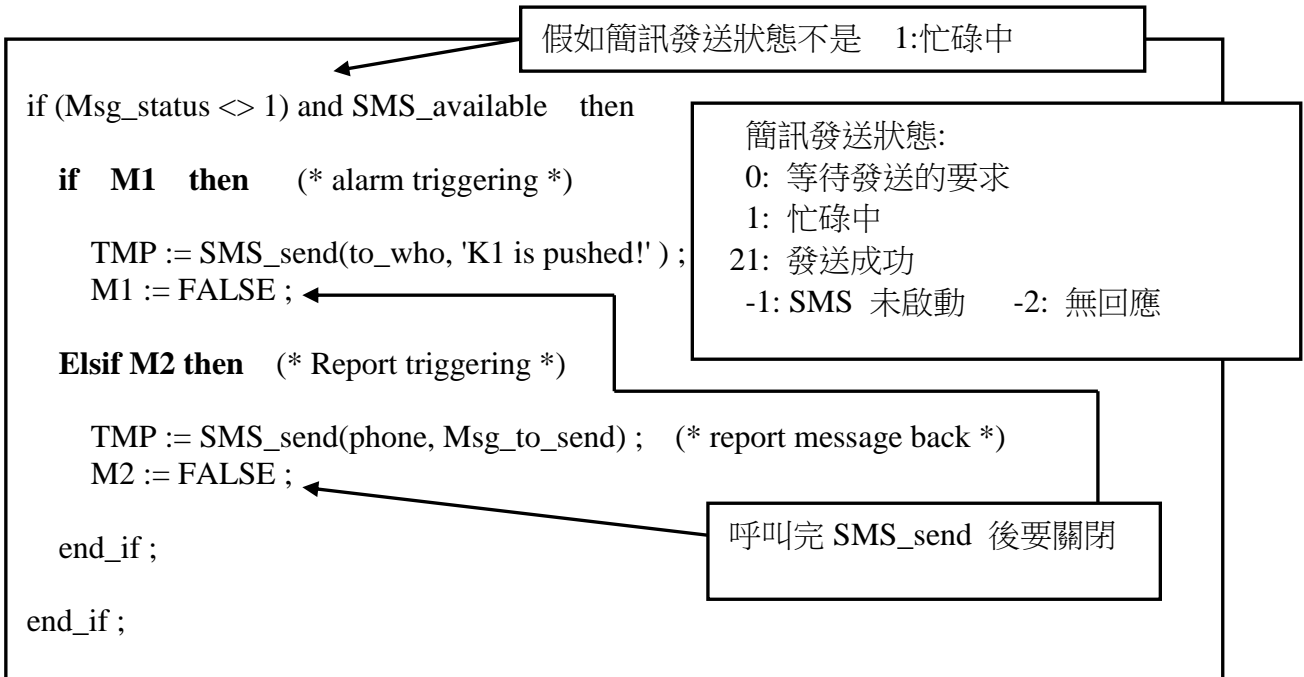
LD 程式 : work



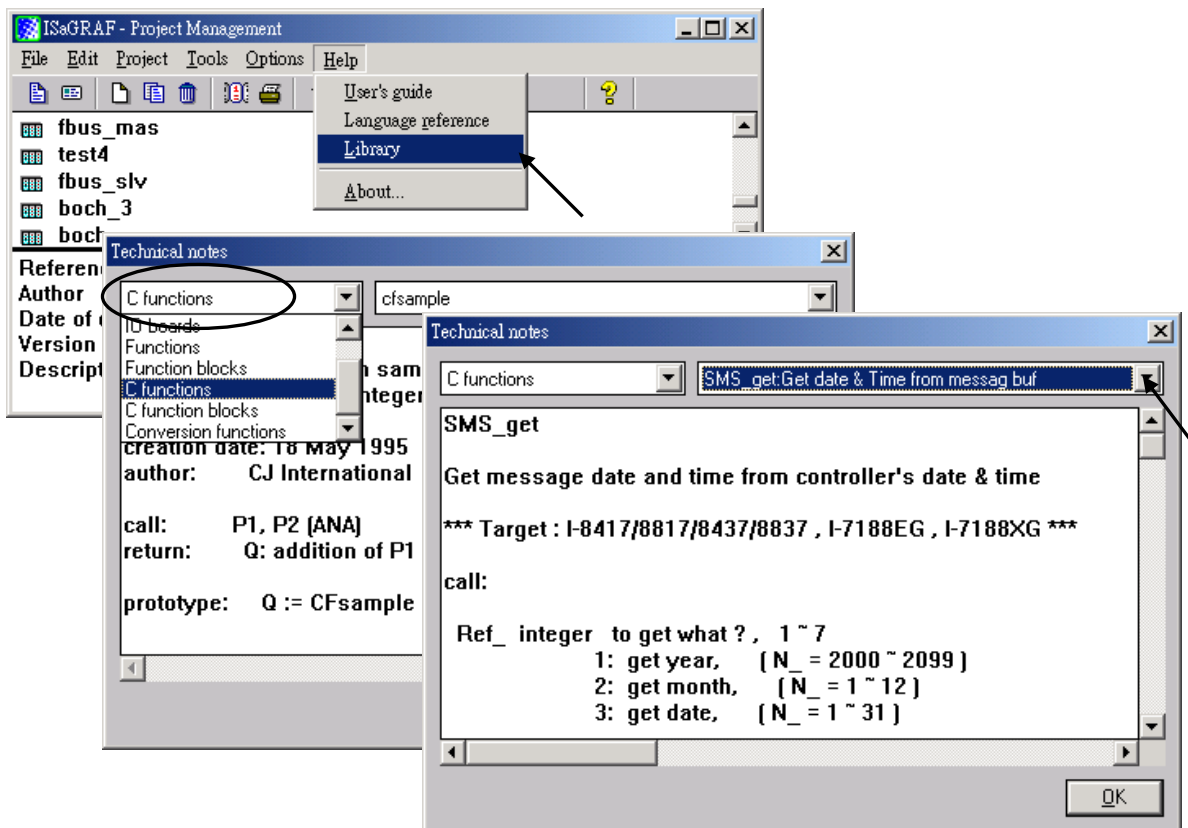
ST 程式： rcv_msg



ST program : snd_msg



關於 SMS_sts, SMS_send, SMS_test, SMS_get 及 SMS_gets 的說明, 請參閱 ISaGRAF 的 On-line Help. “Library” – “C functions” – “SMS_xxxx”



第 18 章馬達運動控制

本章內容使用英文提供.

注意:

1. XP-8347-CE6 與 XP-8747-CE6 有支持 I-8094 / I-8094F (4 軸 motion 與 Encoder) 與 I-8092F (2 軸 motion 與 Encoder), 這是比 I-8091W 還要好用的解法, 請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-132 .
2. I-8417/8817/8437/8837 若要使用 I-8091 與 I-8090 板卡, 必須自行更換 驅動程式為 “Motion” 版本, 標準出貨的驅動版本不支持 I-8091 與 I-8090 板卡. 請到 以下網址下載 <http://www.icpdas.com/products/PAC/i-8000/isagraf-link1.htm>
3. 當 I-8437/8837 更新驅動程式為 “Motion” 版本時, 不支持 Ethernet 通訊. 但 Wincon-8xx7 沒有這個限制 (Wincon-8xx7 標準出貨的驅動版本已經有支持 I-8091 與 I-8090 板卡).
4. 插在 I-8xx7 或 Wincon-8xx7 上的 I-8091 板卡, 只有 1 片 I-8091 支持 X-Y 軸連動, 其它的 I-8091 板卡只能各軸獨立運動. 另外 I-8xx7 最多只支持 2 片 I-8091 卡, Wincon 最多可支持 4 片.

18.1: Install motion driver

Restriction of the motion driver of I-8417/8817/8437/8837:

The motion driver for I-8417/8817/8437/8837 doesn't support the Ethernet communication, however W-8337/8737 desen't have this limitation.

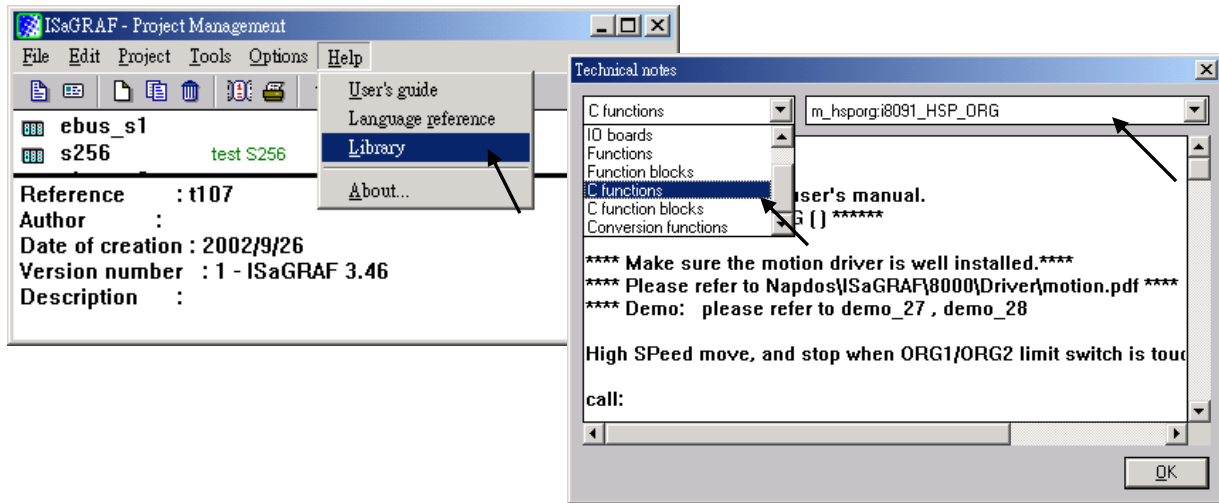
The ISaGRAF demo projects of motion for I-8417/8817/8437/8837 are “demo_27”, “demo_28”, & “demo_46”. They are located in the

I-8000 CD-ROM: \napdos\isagraf\8000\demo\”, or at <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/isagraf/8000/demo/>

The ISaGRAF demo projects of motion for W-8337/8737/8347/8747 are “wdemo_26”, “wdemo_27”, “wdemo_28” & “wdemo_29”. They are located in the

Wincon CD-ROM: \napdos\isagraf\wincon\demo\”, or at ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

All functions that trigger I-8091 & I-8090 are named as "M_???", Please refer to the On-line help from the ISaGRAF "Help" – "Library" - "C functions" for names starting with "M_???".



Beside, please refer to "I-8091 & I-8090 User's Manual" .It can be found in the package box of the I-8091, or

I-8000 CD-ROM: napdos\8000\motion\i8091>manual\

ftp site: <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/8000/motion/i8091/manual/>

18.2: Introduction

18.2.1: System Block Diagram

The I-8091 stepping motor control card is a micro-computer controlled, 2-axis pulse generation card. It includes a 2Kbytes-FIFO to receive motion command from host, a micro-computer for profile generation and protection, 2-axis DDA chip to execute DDA function when interpolation command is used, 2500Vrms optical isolation inserted for industrial application.

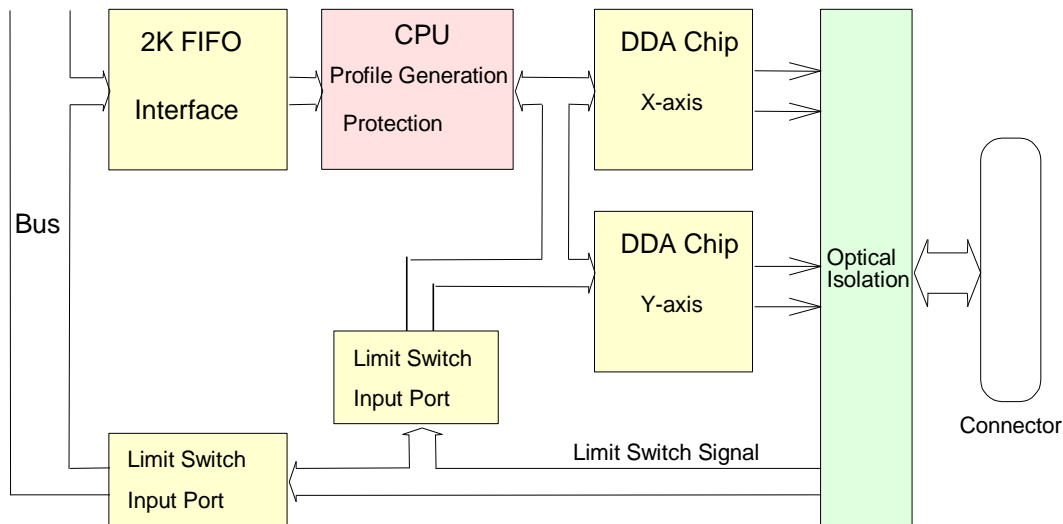


Fig.(1) block diagram of I-8091 card

18.2.2: DDA Technology

The DDA chip is the heart of I-8091 card, it will generate equal-space pulse train corresponding to specific pulse number during a DDA period. This mechanism is very useful to execute pulse generation and interpolation function. The DDA period can be determined by DDA cycle. Table(1) shows the relation among DDA cycle, DDA period and output pulse rate. When DDA cycle set to 1, the DDA period is equal to $(1+1) \times 1.024\text{ms} = 2.048\text{ms}$. The output pulse number can be set to 0~2047, therefore the maximum output pulse rate will be 1Mpps. The minimum output pulse rate is 3.83pps when set DDA cycle=254 (DDA period = $(254+1) \times 1.024\text{ms} = 261.12\text{ms}$).

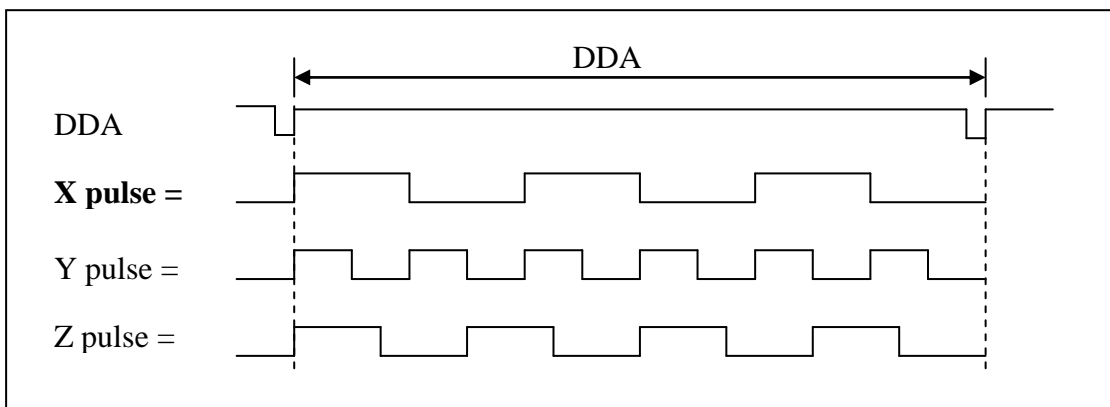


Fig.(2) DDA mechanism

Table(1) The Relation among DDA cycle, DDA period and output pulse rate.

DDA cycle	DDA period	Max. pulse rate(n=2047)	Min. pulse rate (n=1)
1	2.048ms	999511pps	488pps
2	3.072ms	666341pps	325pps
3	4.096ms	.	.
.	.	.	.
N	(N+1)*1.024ms	2047/(DDA period)	1/(DDA period)
.	.	.	.
254	261.12ms	7839pps	3.83pps

The DDA cycle can be set by i8091_SET_VAR() command which described in chapter 3. The selection criterion of DDA cycle was described as following.

1. The required max. output pulse rate.

$$PR_{max} = \frac{V_{max} * N / 60}{2047}$$

$$PR_{max} = \frac{V_{max} * N}{(DDA_{cycle} + 1) * 1.024ms * 60}$$

PR_{max} : max. output pulse rate.

V_{max} : max. speed (rpm).

N : the pulse number of stepping motor per revolution (pulse/rev).

2. The required speed resolution.

The maximum output pulse number is N_p(0~2047), therefore the speed resolution is V_{max}(max. speed)/N_p. The DDA cycle can be obtained by following equation.

$$PR_{max} = \frac{N_p}{(DDA_{cycle} + 1) * 1.024ms}$$

3. When choose large DDA cycle (DDA period), it will occur vibration between different pulse input which generally can be observed during acceleration or deceleration. So, the small DDA cycle, the smooth acceleration/deceleration curve as long as the speed resolution is acceptable.

Example: Stepping Motor

The spec. of stepping motor is 500 pulse/rev, max. speed 500 rpm, speed resolution 2 rpm.

The required max. pulse rate

$$PR_{max} = 500 \text{ rpm} * 500 / 60 = 4166.67 \text{ pps}$$

The maximum output pulse

$$N_p = 500 \text{ rpm} / 2 \text{ rpm} = 250 \text{ pulse number}$$

The DDA cycle can be calculated by follow equation

$$PR_{max} = \frac{N_p}{(DDA_{cycle} + 1) * 1.024ms}$$

$$4166.67 = \frac{250}{(DDA_{cycle} + 1) * 1.024ms}$$

$$DDA \text{ cycle} = 58$$

$$\text{High Speed} = 247 \text{ pulse}$$

$$(4166.67 * 58 * 0.001024)$$

The above results means that maximum speed is 500rpm when send command i8091_SET_VAR(0, 58, 2, 2, 247) to I-8091 card.

Example: Pulse type input Servo Motor

The spec. of servo motor is 8000 pulse/rev, max. speed 3000 rpm, speed resolution 2 rpm.

The required max. pulse rate

$$PR_{max} = 3000 \text{ rpm} * 8000 / 60 = 400,000 \text{ pps}$$

The maximum output pulse

$$N_p = 3000 \text{ rpm} / 2 \text{ rpm} = 1500 \text{ pulse number}$$

The DDA cycle can be calculated by follow equation

$$PR_{max} = \frac{N_p}{(DDA_{cycle} + 1) * 1.024ms}$$

$$400,000 = \frac{1500}{(DDA_{cycle} + 1) * 1.024ms}$$

$$DDA \text{ cycle} = 3$$

$$\text{High Speed} = 1638 \text{ pulse} (400,000 * 4 * 0.001024)$$

The above results means that maximum speed is 3000rpm when send command i8091_SET_VAR(0, 3, 2, 2, 1638) to I-8091 card.

18.3: Hardware

18.3.1: I-8000 hardware address

The hardware address of I-8000 main system is fixed as following table. There are 4 slots I-8000 and 8 slots I-8000.

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	Slot 7
I-8000, 4 slot address	0x080	0x0A0	0x0C0	0x0E0	---	---	---	---
I-8000, 8 slot address	0x080	0x0A0	0x0C0	0x0E0	0x140	0x160	0x180	0x1A0

Fig.(3) I-8000 hardware address

18.3.2: LED Indicator

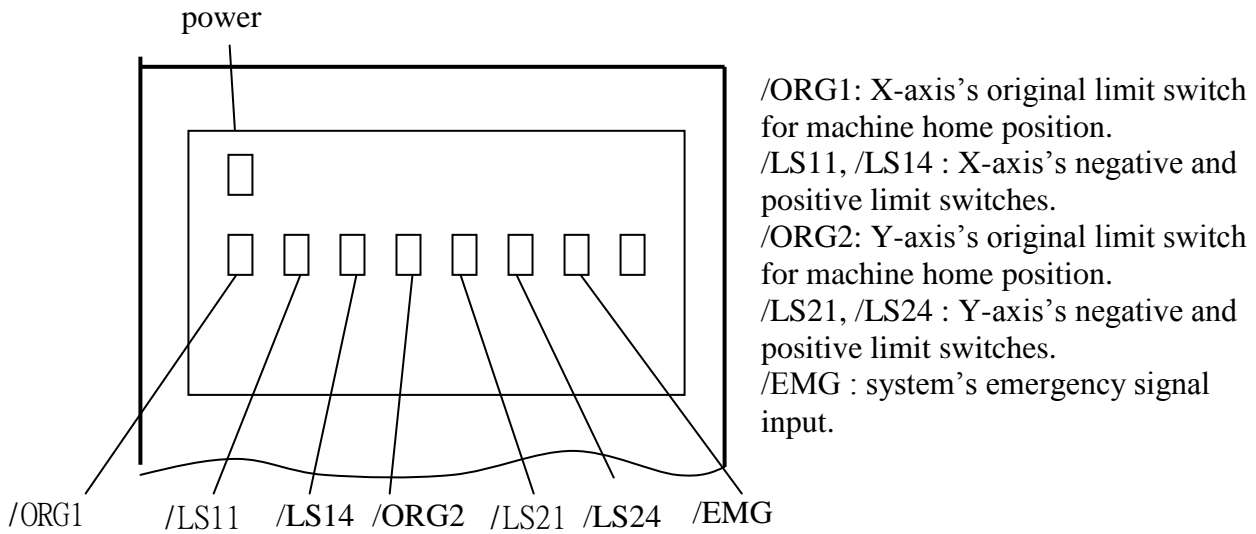


Fig.(4) I-8091 LED indicator

18.3.3: Hardware Configuration

Limit switch configuration

Because the profile generation and protection is executed by the CPU on I-8091 card, the limit switches must configure as following diagram. The motion command just can work properly.

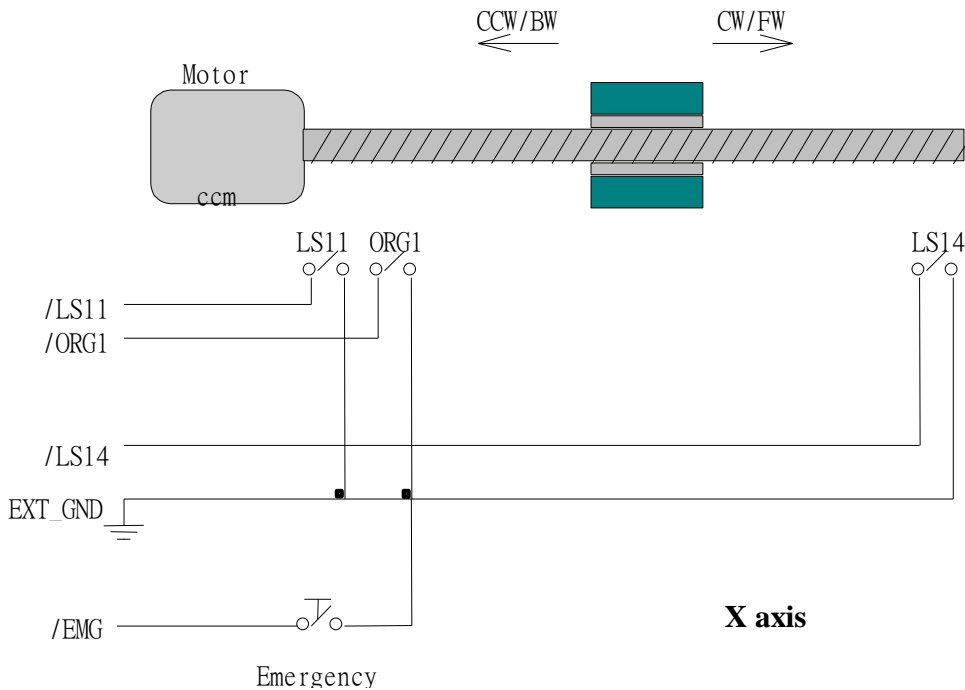


Fig.(5) Limit switch configuration of X axis

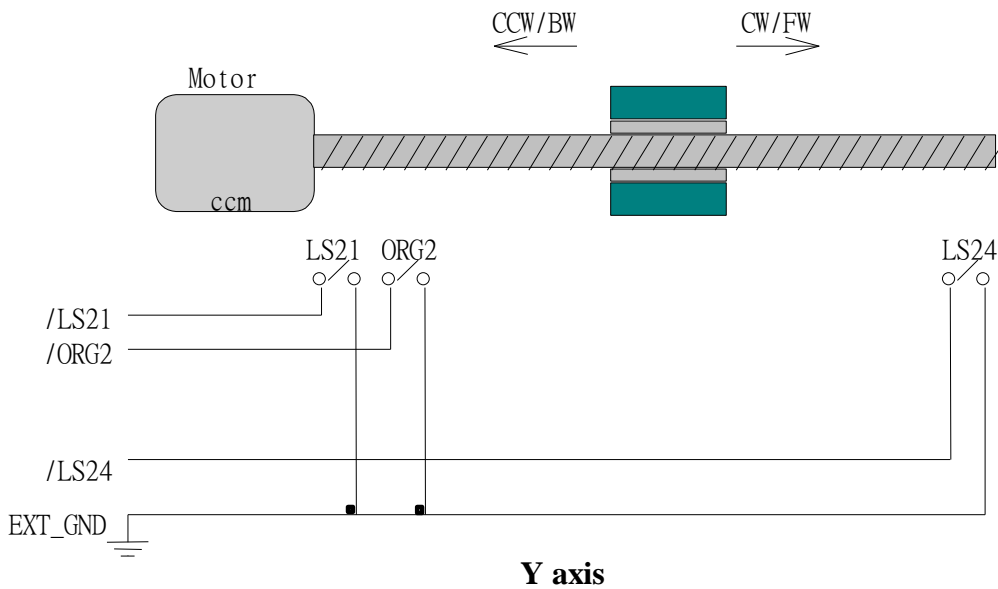


Fig.(6) Limit switch configuration of Y axis

Output pulse mode configuration

I-8091 card provide two kind output method.

- (a) CW/CCW mode
- (b) Pulse/Direction mode

The command **M_s_mode(card_NO_, modeX_, modeY_)** provide parameters 0: CW_CCW and 1: PULSE_DIR to define output pulse mode.

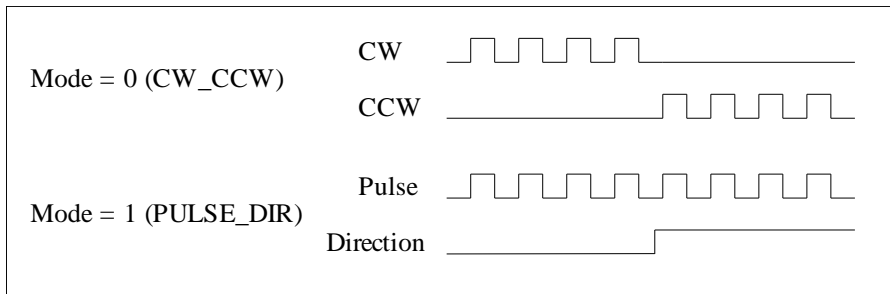


Fig.(7) Output pulse mode

Direction configuration

Sometimes, the output direction of X-axis, Y-axis is not in the desired direction due to the motor's connection or gear train. It is recommended to unify the output direction as shown in Figure(5)(6). The CW/FW direction is defined as toward outside from motor and the CCW/BW direction is defined as toward inside to motor. The **M_s_dir(card_NO_, defdirX_, defdirY_)** command provides parameters 0: NORMAL_DIR and 1: REVERSE_DIR to define the rotating direction of motor.

Turn Servo ON/OFF (Hold ON/OFF)

To turn servo motor into servo ON(OFF) state, or turn stepping motor into hold ON(OFF) state, the command **M_s_serv(card_NO_, sonX_, sonY_)** provide parameters 1:ON and 0:OFF to turn ON or OFF.

Automatic protection

The I-8091 card has a automatic protected system.

- (a) If X-axis command is executing and moving toward CW/FW direction, X-axis will immediately stop when LS14 is touched. To release this protection as long as X-axis move toward CCW/BW direction.
- (b) If X-axis command is executing and moving toward CCW/BW direction, X-axis will immediately stop when LS11 is touched. To release this protection as long as X-axis move toward CW/FW direction.
- (c) If Y-axis command is executing and moving toward CW/FW direction, Y-axis will immediately stop when LS24 is touched. To release this protection as long as Y-axis move toward CCW/BW direction.
- (d) If Y-axis command is executing and moving toward CCW/BW direction, Y-axis will immediately stop when LS21 is touched. To release this protection, as long as Y-axis move toward CW/FW direction.
- (e) If the signal of the emergency limit switch /EMG was found in CPU firmware, all motion will be terminated and stop.

Set limit switch as normal close condition

The limit switches /EMG, /LS11, /LS14, /LS21, /LS24, /ORG1, /ORG2 is initially normal open condition, that is, these signal is active when connect it to ground. In industrial application, it might be recommended normal close condition, that is, these signal is active when open from ground. The **M_s_nc(card_NO_, sw_)** command can be set sw=0 (default), for normal open condition. When set sw=1, for normal close condition.

18.3.4: Pin assignment of connector CN2

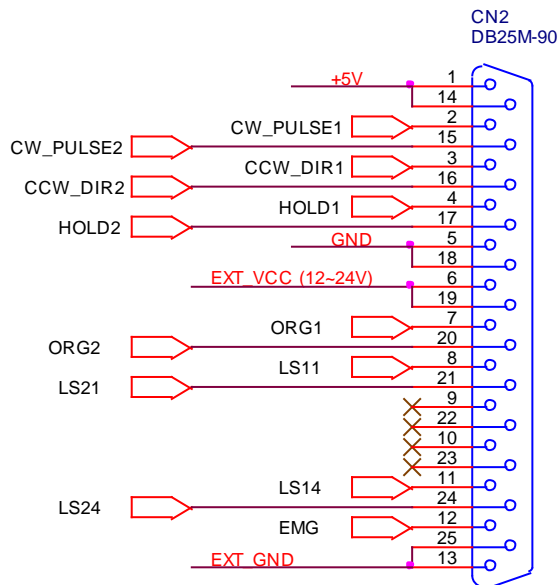


Fig.(8) CN2 connector of I-8091

Table of CN2 connector's pin assignment

pin name	pin number	Description
+5V	1	Internal +5V power, Max. output current: 50mA
CW_PULSE1	2	X-axis CW (Pulse) output pin
CCW_DIR1	3	X-axis CCW (Direction) output pin
HOLD1	4	X-axis HOLD (servo on) output pin
GND	5	Signal ground of pin 2,3,4
EXT_VCC	6	External power(12~24V) for limit switches
/ORG1	7	X-axis original (home) limit switch
/LS11	8	X-axis limit switch
	9,10	No used
/LS14	11	X-axis limit switch
/EMG	12	Emergency input
EXT_GND	13	External ground for limit switch
+5V	14	Internal +5V power, Max. output current: 50mA
CW_PULSE2	15	Y-axis CW (Pulse) output pin
CCW_DIR2	16	Y-axis CCW (Direction) output pin
HOLD2	17	Y-axis HOLD (servo on) output pin
GND	18	Signal ground of pin 15,16,17
EXT_VCC	19	External power(12~24V) for limit switches
/ORG2	20	Y-axis original (home) limit switch
/LS21	21	Y-axis limit switch
	22,23	No used
/LS24	24	Y-axis limit switch
EXT_GND	25	External ground for limit switch

The internal circuit of CW_PULSE, CCW_DIR, HOLD

When output these signal as 1, it can source 15mA(max.).
 When output these signal as 0, it can sink 50mA(max.)

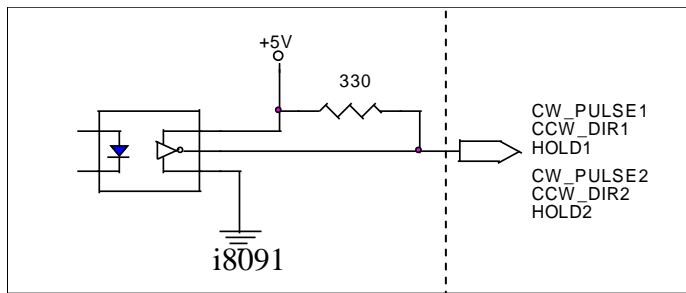


Fig.(9) internal circuit of pulse output pin

The internal circuit of limit switch input

Initially, the limit switch inputs of I-8091 board are normal open (N.O.), the I-8091 board will automatic protect when limit switch pin connect to EXT_GND. The user can use the command **M_s_nc(card_NO_, 1)** to let those limit switch input as normal close condition at the beginning of the user's program.

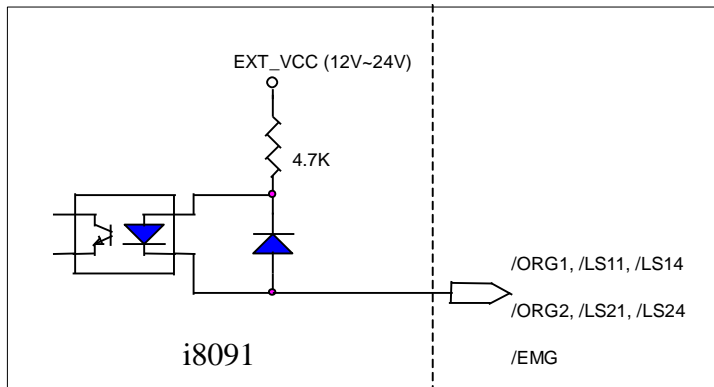


Fig.(10) internal circuit of limit switch input pin

Example of connection

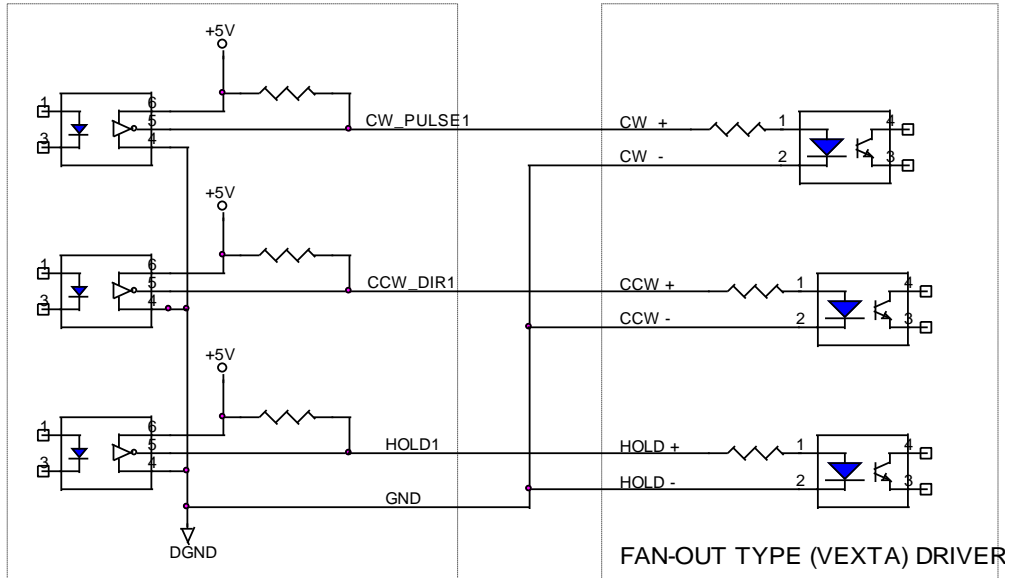


Fig.(11) fan-out type driver (VEXTA's motor driver)

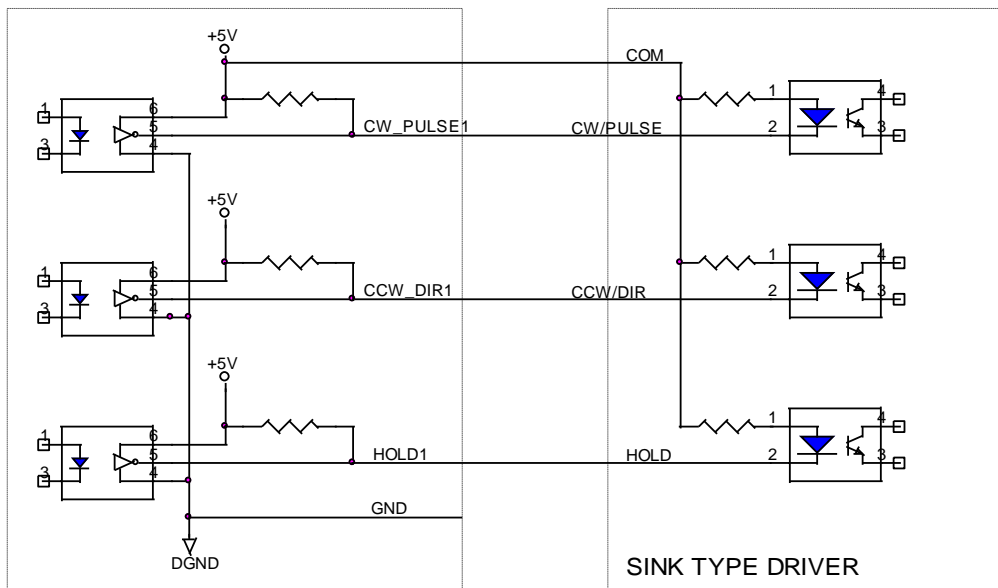


Fig.(12) Sink type driver

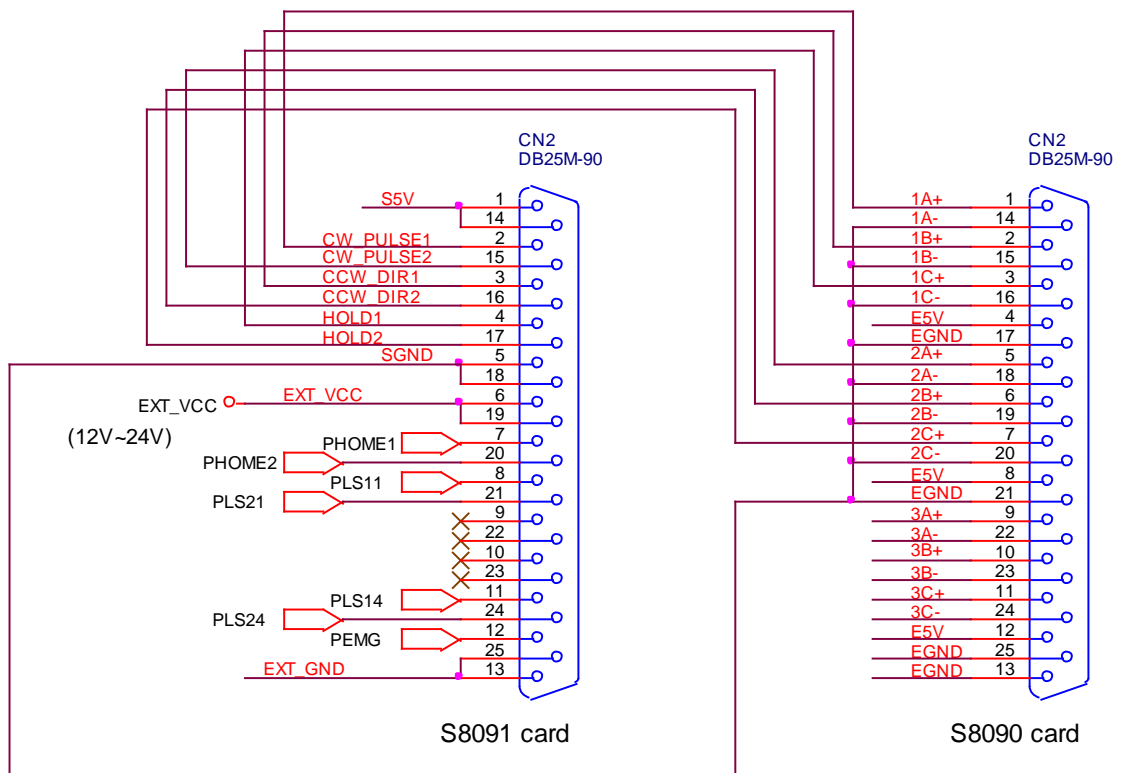
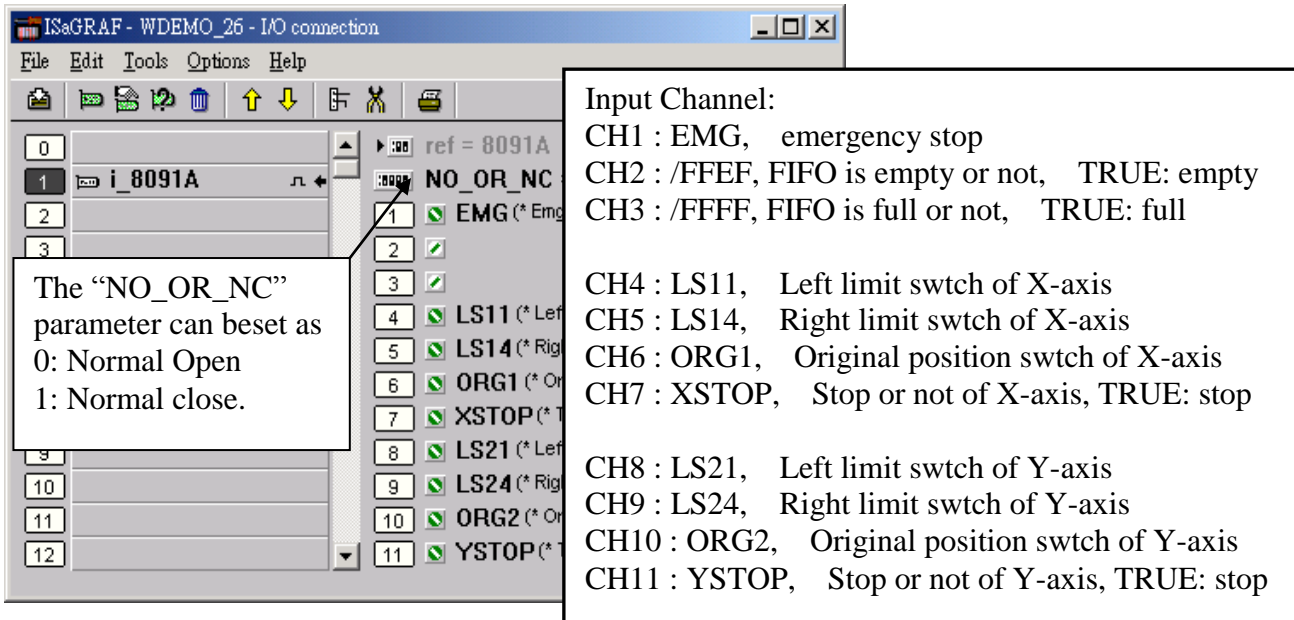


Fig.(13) The connection between I-8090 and I-8091 for function testing or pulse feedback by I-8090 encoder card.

18.4: Software

I/O connection:

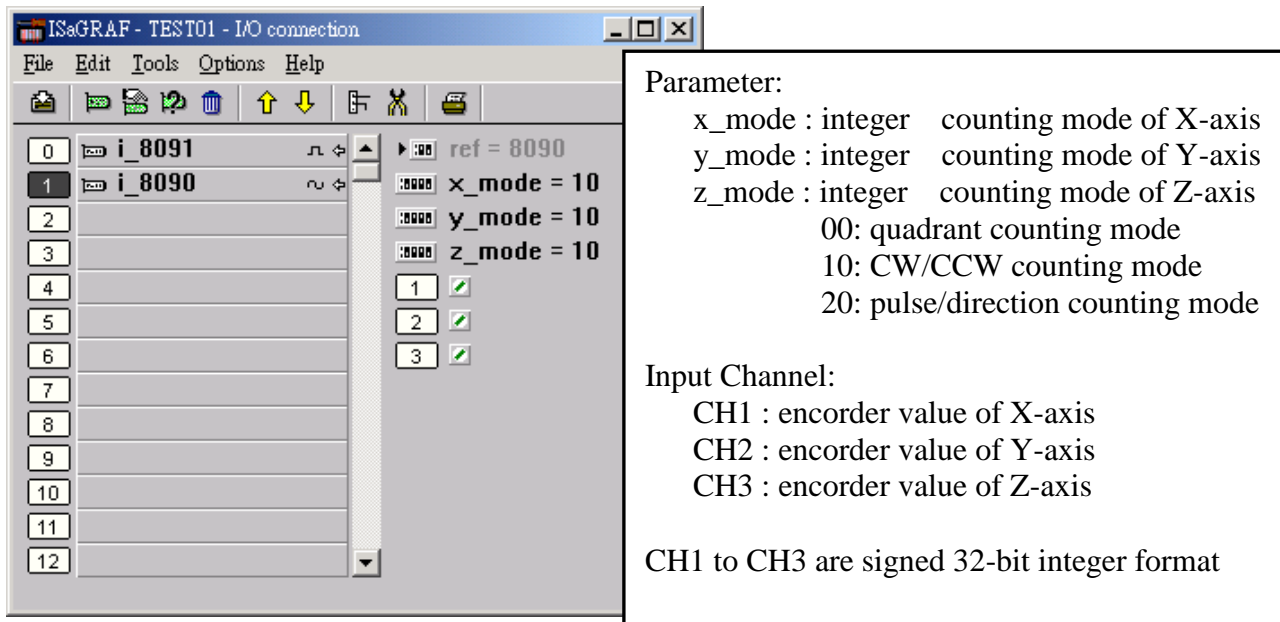
The “I-8091A” connected on the I/O connection window contains 11 digital input channels.



The “NO_OR_NC” parameter can be set as:
 0: Normal Open
 1: Normal close.

Input Channel:
 CH1 : EMG, emergency stop
 CH2 : /FFEF, FIFO is empty or not, TRUE: empty
 CH3 : /FFFF, FIFO is full or not, TRUE: full
 CH4 : LS11, Left limit switch of X-axis
 CH5 : LS14, Right limit switch of X-axis
 CH6 : ORG1, Original position switch of X-axis
 CH7 : XSTOP, Stop or not of X-axis, TRUE: stop
 CH8 : LS21, Left limit switch of Y-axis
 CH9 : LS24, Right limit switch of Y-axis
 CH10 : ORG2, Original position switch of Y-axis
 CH11 : YSTOP, Stop or not of Y-axis, TRUE: stop

I-8090 contains 3 analog input channels.



Parameter:
 x_mode : integer counting mode of X-axis
 y_mode : integer counting mode of Y-axis
 z_mode : integer counting mode of Z-axis
 00: quadrant counting mode
 10: CW/CCW counting mode
 20: pulse/direction counting mode

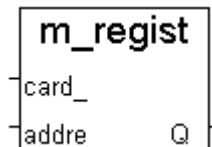
Input Channel:
 CH1 : encorder value of X-axis
 CH2 : encorder value of Y-axis
 CH3 : encorder value of Z-axis

CH1 to CH3 are signed 32-bit integer format

Setting commands:

M_regist Register one I-8091

In order to distinguish more than one I-8091 card in I-8417/8817/8437/8837 platform, the I-8091 cards should be registrated before using it. This command will assign a card number = “card_NO_” to I-8091 card at that “address_”. If there is no I-8091 at the given address, this command will return FALSE.



Note: If using “I_8091A” rather than “I_8091” on the I/O connection window, user don't need to call “m_regist” & “m_s_nc”, they are ignored. The card_NO of “I-8091A” is equal to its slot No. I-8xx7: 0 ~ 7. W-8xx7: 1 ~ 7.

Parameters:

card_NO_	integer	valid is 0 ~ 19.
address_	integer	the plugged slot address of the i8091 card

slot 0:	16#80
slot 1:	16#A0
slot 2:	16#C0
slot 3:	16#E0
slot 4:	16#140
slot 5:	16#160
slot 6:	16#180
slot 7:	16#1A0

Return:

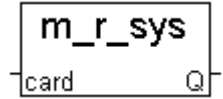
Q_	boolean	TRUE: Ok , FALSE: Fail
----	---------	---------------------------

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
 W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

```
(* declaration:    INIT as boolean <internal> and has initial value of TRUE    *)
(*    TMP as boolean <internal>                                                            *)
(* cardNO as integer <internal> and has intial value of 1    *)
(* Do some init setting at 1st scan cycle *)
if INIT then
          INIT := FALSE;
          TMP := M_regist(cardNO,16#80);                            (* plug i8091 in slot 0 *)
          TMP := M_r_sys(cardNO);                                    (* reset i8091's setting *)
          TMP := M_s_var(cardNO,4,2,5,100);
          TMP := M_s_dir(cardNO,0,0);                                (* Normal direction *)
          TMP := M_s_mode(cardNO,1,1);                                (* pulse_dir mode *)
          TMP := M_s_serv(cardNO,1,1);                                (* X & Y server ON *)
          TMP := M_s_nc(cardNO,0);                                    (* Normal open *)
end_if;
```

M_r_sys Reset all setting

To reset I-8091 card, this command will terminate the running command in I-8091 card. User can use this command as software emergency stop. This command also will clear all of setting, so, all I-8091 card's parameter should be set again.



Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

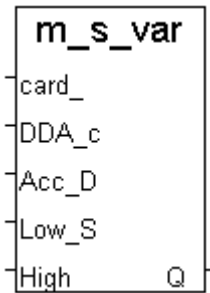
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_var Set motion system parameters

To set DDA cycle, accelerating/decelerating speed, low speed and high speed value.

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
DDA_cycle_	integer	DDA cycle , valid is 1 ~ 254
Acc_Dec_	integer	Acc/Dec speed , valid is 1 ~ 200
Low_Speed_	integer	low speed , valid is 1 ~ 200 , Low_Speed_ >= Acc_Dec_
High_Speed_	integer	high speed , Low_Speed_ <= High_Speed_ <= 2047

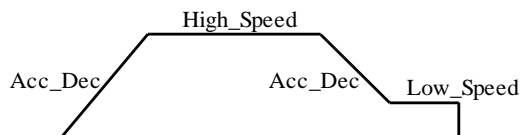


Return:

Q_ boolean always return TRUE.

Note:

The lower “DDA_cycle_” is given, the smaller delay time between /ORG1 ON and /X_STOP ON (or /ORG2 ON and /Y_STOP ON) when using M_hsporg & M_lsporg command. For ex, DDA_cycle_ set to 4, the delay time is about 5 to 13 ms.



Restriction:

$1 \leq DDA_cycle \leq 254$
 $1 \leq Acc_Dec \leq 200$
 $1 \leq Low_Speed \leq 200$
 $Low_Speed \leq High_Speed \leq 2047$
 $Low_Speed \geq Acc_Dec$

Default value

$DDA_cycle = 10$
 $Acc_Dec = 1$
 $Low_Speed = 10$
 $High_Speed = 100$

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
 W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

```

TMP := M_s_var(1, 5, 2, 10, 150);
(* DDA_cycle = 5    --> DDA period = (5+1)*1.024ms = 6.144ms
   Acc_Dec = 2     --> Acc/Dec speed = 2/(6.144ms)^2 = 52981 p/s^2
   Low_Speed = 10  --> low speed = 10/6.144ms = 1628pps
   High_Speed = 150 --> high speed = 150/6.144ms = 24414pps *)
    
```

M_s_dir Define output direction of axes

Sometimes, the output direction of X-axis, Y-axis is undesired direction due to the motor's connection or gear train. In order to unify the output direction as shown in Fig.(5) and Fig.(6). Where CW/FW direction is defined as toward outside from motor, CCW/BW direction is defined as toward inside from motor. This command provide parameters to define the rotating direction of motor.



Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
defdirX_	integer	X axis direction definition , valid is 0 ~ 1
defdirY_	integer	Y axis direction definition , valid is 0 ~ 1

0: normal direction, 1: reverse direction

reverse direction

Return:

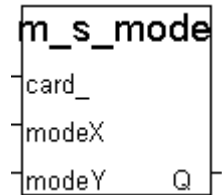
Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
 W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_mode Set output mode

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
modeX_	integer	X axis mode, valid is 0 ~ 1
modeY_	integer	Y axis mode, valid is 0 ~ 1

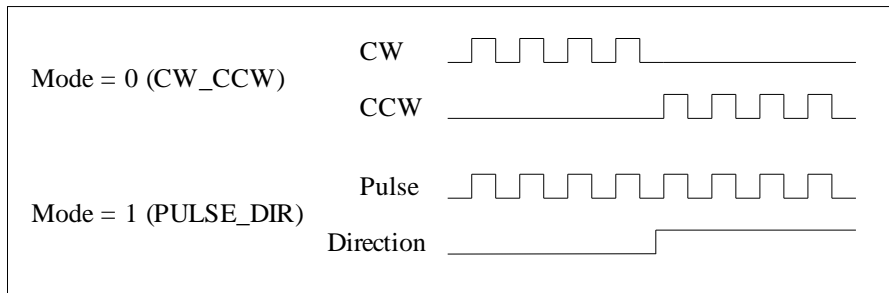


0: CW_CCW, 1:

PULSE_DIR

Return:

Q_ boolean always return TRUE.

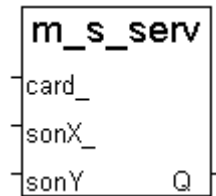


Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
 W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_serv Set servo ON/OFF

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
sonX_	integer	X axis servo/hold on switch , valid is 0 ~ 1
sonY_	integer	Y axis servo/hold on switch , valid is 0 ~ 1 0: OFF, 1: ON



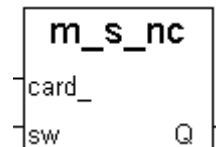
Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_s_nc Set N.O. / N.C.

To set all of the following limit switches as N.C.(normal close) or N.O.(normal open). If set as N.O., those limit switches are active low. If set as N.C., those limit switches are active high. The auto-protection will automatically change the judgement whatever it is N.O. or N.C..



Limit switches: ORG1, LS11, LS14, ORG2, LS21, LS24, EMG.

Note: If using "I_8091A" rather than "I_8091" on the I/O connection window, user don't need to call "m_regist" & "m_s_nc", they are ignored. The card_NO of "I-8091A" is equal to its slot No. I-8xx7: 0 ~ 7. W-8xx7: 1 ~ 7.

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
sw_	integer	0: N.O. (default), 1: N.C.

Return:

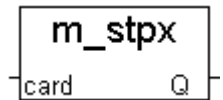
Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Note: If using "I_8091A" in the ISaGRAF IO connection window, there is a "NO_OR_NC" parameter can be set to define as 0:Normal Open , 1:Normal Close. So user no more need to call this "m_s_nc" function if using "I_8091A".

Stop commands:

M_stpx **Stop X axis**



Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

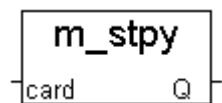
Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_stpy **Stop Y axis**



Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_stpall **Stop X & Y axes**



This command will stop X & Y axes and clear all of commands pending in the FIFO.

Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19

Return:

Q_ boolean always return TRUE.

Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Simple motion commands:

M_lsporg Low speed move to ORG

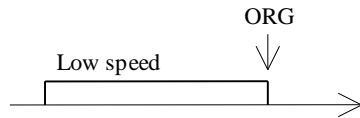
Low speed move , and stop when **ORG1/ORG2** limit switch is touched.

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
DIR_	integer	0: CW , 1: CCW
AXIS_	integer	1: X axis , 2: Y axis

Return:

Q_ boolean always return TRUE.



M_hsporg High speed move to ORG

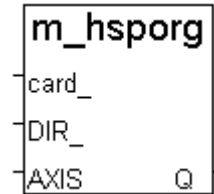
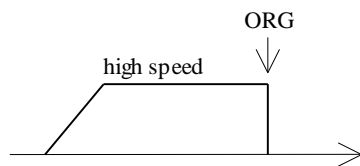
High speed move , and stop when **ORG1/ORG2** limit switch is touched.

Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
DIR_	integer	0: CW , 1: CCW
AXIS_	integer	1: X axis , 2: Y axis

Return:

Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28

W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Note:

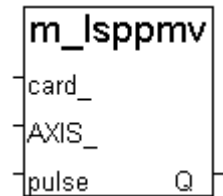
The lower "DDA_cycle_" is given, the smaller delay time between /ORG1 ON and /X_STOP ON (or /ORG2 ON and /Y_STOP ON) when using M_hsporg & M_lsporg command. For ex, DDA_cycle_ set to 4, the delay time is about 5 to 13 ms.

M_lsppmv Low speed pulse move

Low speed move a specified “pulse”

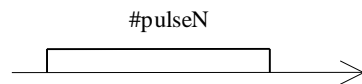
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
AXIS_	integer	1: X axis , 2: Y axis
Pulse_	integer	number of pulse to move. if > 0, move toward CW/FW dir. if < 0, move toward CCW/BW dir.



Return:

Q_ boolean always return TRUE.



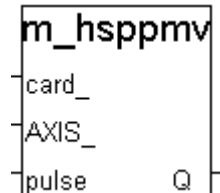
Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_hsppmv High speed pulse move

High speed move a specified “pulse”

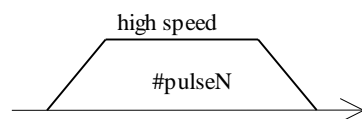
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
AXIS_	integer	1: X axis , 2: Y axis
Pulse_	integer	number of pulse to move. if > 0, move toward CW/FW dir. if < 0, move toward CCW/BW dir.



Return:

Q_ boolean always return TRUE.



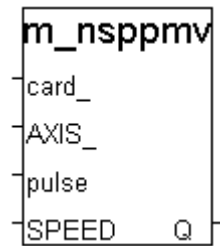
Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_nsppmv Normal speed pulse move

Normal speed move a specified “pulse”

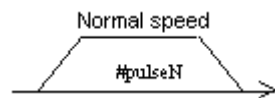
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
AXIS_	integer	1: X axis , 2: Y axis
Pulse_	integer	number of pulse to move. if > 0, move toward CW/FW dir. if < 0, move toward CCW/BW dir.
SPEED_	integer	Speed, low speed <= SPEED_ <= high speed



Return:

Q_ boolean always return TRUE.



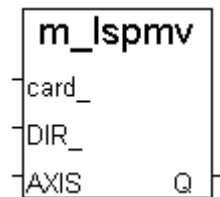
Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_lspm Low speed move

Low speed move toward the direction specified. It can be stop by **M_stpx** or **M_stpy** or **M_stpall** command

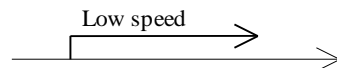
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
DIR_	integer	direction. 0: CW , 1: CCW
AXIS_	integer	1: X axis , 2: Y axis



Return:

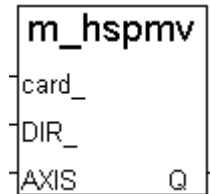
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_hspmv High speed move

High speed move toward the direction specified. It can be stop by **M_stpx** or **M_stpy** or **M_stpall** command

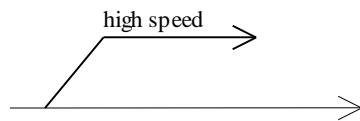


Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
DIR_	integer	direction. 0: CW , 1: CCW
AXIS_	integer	1: X axis , 2: Y axis

Return:

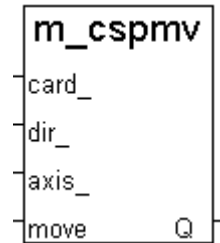
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_cspmv Change speed move

This command will accelerate/decelerate the selected axis's motor to the "move_speed". This command can be continuously send to I-8091 to dynamically change speed. The rotating motor can be stop by the command **M_stpx**, **M_stpy**, **M_stpall**, or **M_slwstp**

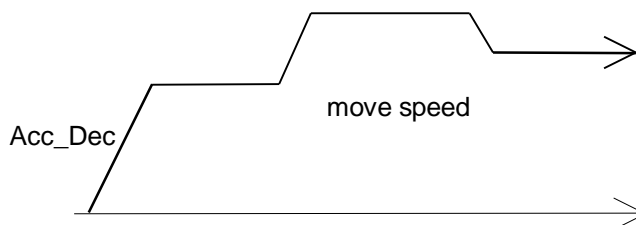


Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
dir_	integer	direction. 0: CW , 1: CCW
axis_	integer	1: X axis , 2: Y axis
move_speed_	integer	0 < move_speed_ <= 2040

Return:

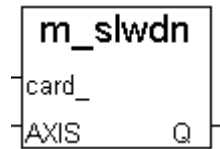
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_slwdn Slow down to low speed

To decelerate to slow speed until **M_stpx** or **M_stpy** or **M_stpall** is executed.

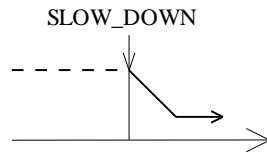


Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_ integer 1: X axis , 2: Y axis

Return:

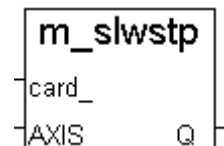
Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_slwstp Slow down to stop

To decelerate to stop.

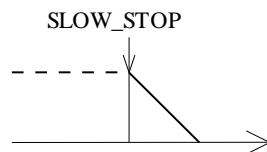


Parameters:

card_NO_ integer the card No. has been set by **M_regist**, valid is 0 ~ 19
AXIS_ integer 1: X axis , 2: Y axis

Return:

Q_ boolean always return TRUE.

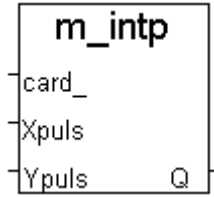


Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

Interpolation commands:

M_intp Move a short distance on X-Y plane

This command will move a short distance (interpolation short line) on X-Y plane. This command provided a method for user to generate an arbitrary curve on X-Y plane.

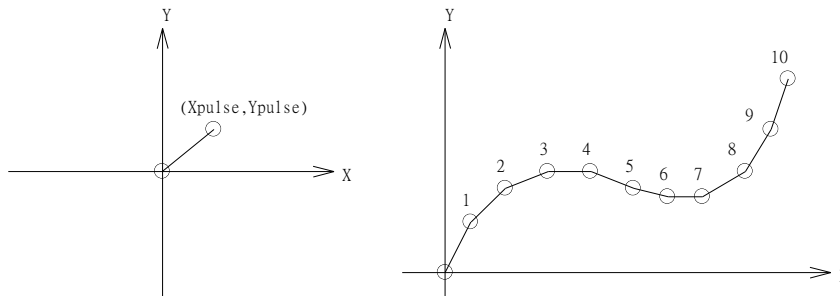


Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
Xpulse_	integer	-2047 <= Xpulse_ <= 2047
Ypulse_	integer	-2047 <= Ypulse_ <= 2047

Return:

Q_ boolean always return TRUE.



Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

NOTE:

For a lot of **M_intp** call set at the same time, please check if the FIFO is not full. Call it if FIFO is not full. FIFO indicator is a Digital Input resides at CH3 of I-8091.

I-8091 D/I channel on ISaGRAF I/O connection window:

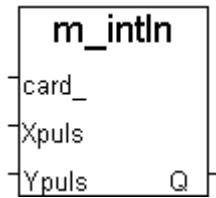
- CH1 : EMG, emergency stop
- CH2 : /FFEF, FIFO is empty or not, TRUE: empty
- CH3 : /FFFF, FIFO is full or not, TRUE: full

- CH4 : LS11, Left limit swtch of X-axis
- CH5 : LS14, Right limit swtch of X-axis
- CH6 : ORG1, Original position swtch of X-axis
- CH7 : XSTOP, Stop or not of X-axis, TRUE: stop

- CH8 : LS21, Left limit swtch of Y-axis
- CH9 : LS24, Right limit swtch of Y-axis
- CH10 : ORG2, Original position swtch of Y-axis
- CH11 : YSTOP, Stop or not of Y-axis, TRUE: stop

M_intln Move a long distance on X-Y plane

This command will move a long distance (interpolation line) on X-Y plane. The CPU on I-8091 card will generate a trapezoidal speed profile of X-axis and Y-axis, and execute interpolation by way of DDA chip.

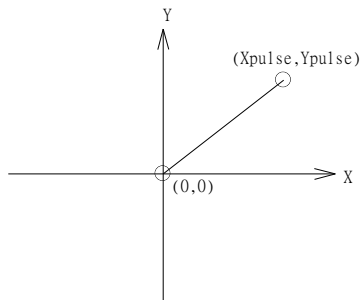


Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
Xpulse_	integer	-524287 <= Xpulse_ <= 524287
Ypulse_	integer	-524287 <= Xpulse_ <= 524287

Return:

Q_	boolean	always return TRUE.
----	---------	---------------------



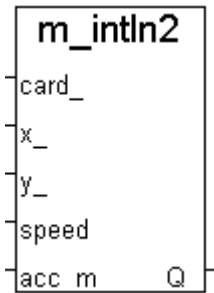
Example: I-8417/8817/8437/8837: demo_46, demo_27, demo_28
W-8337/8737: wdemo_26, wdemo_27, wdemo_28, wdemo_29

M_intln2 Move a long distance on X-Y plane

This command will move a long interpolation line on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

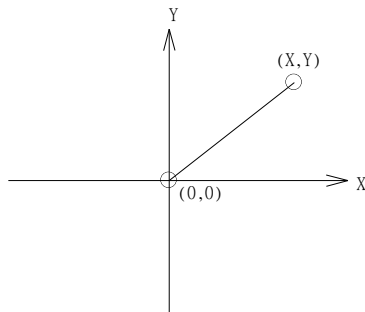
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
x_, y_	integer	end point relate to present position
speed_	integer	0 ~ 2040
acc_mode_	integer	0: enable acceleration/deceleration profile 1: disable acceleration/deceleration profile



Return:

Q_ boolean always return TRUE.



NOTE:

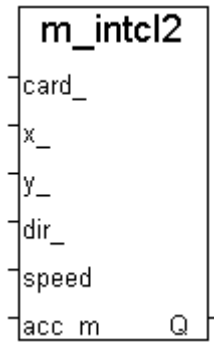
1. Only one of **M_intln2**, **M_intcl2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcl2** & **M_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M_intln2**, **M_intcl2**, **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2**, **M_intcl2**, **M_intar2** at the same time is not possible.

M_intcl2 Move a circle on X-Y plane

This command will generate an interpolation circle on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

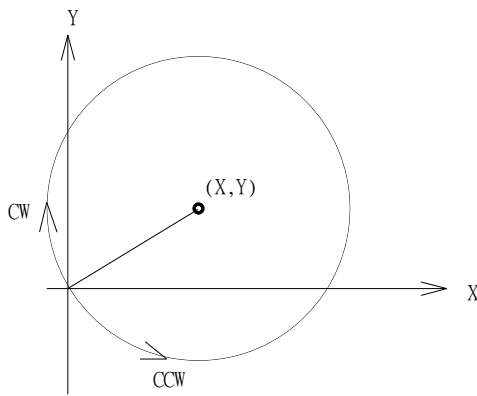
Parameters:

card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
x_, y_	integer	center point of circle relate to present position
dir_	integer	moving direction. 0: CW , 1: CCW
speed_	integer	0 ~ 2040
acc_mode_	integer	0: enable acceleration/deceleration profile 1: disable acceleration/deceleration profile



Return:

Q_	boolean	always return TRUE.
----	---------	---------------------



where radius = $\sqrt{X^2 + Y^2}$

NOTE:

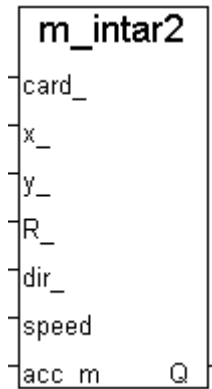
1. Only one of **M_intln2**, **M_intcl2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcl2** & **M_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M_intln2**, **M_intcl2**, **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2**, **M_intcl2**, **M_intar2** at the same time is not possible.

M_intar2 Move a arc on X-Y plane

This command will generate an interpolation arc on X-Y plane. It will automatically generate a trapezoidal speed profile of X-axis and Y-axis by state-machine-type calculation method.

Parameters:

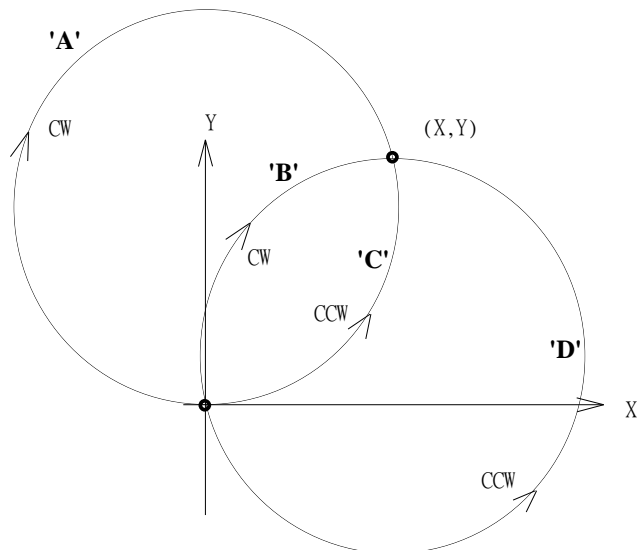
card_NO_	integer	the card No. has been set by M_regist , valid is 0 ~ 19
x_, y_	integer	end point of arc relate to present position
R_	integer	radius of arc, if > 0, the arc < 180 degree, if < 0, the arc > 180 degree R_ must > (square root of (X_*X_+Y_*Y_)) / 2
dir_	integer	moving direction. 0: CW, 1: CCW
speed_	integer	0 ~ 2040
acc_mode_	integer	0: enable acceleration/deceleration profile 1: disable acceleration/deceleration profile



Return:

Q_	boolean	always return TRUE.
----	---------	---------------------

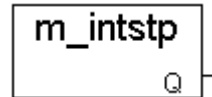
R	dir	path of curve
R>0	CW	'B'
R>0	CCW	'C'
R<0	CW	'A'
R<0	CCW	'D'



NOTE:

- Only one of **M_intln2**, **M_intcl2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcl2** & **M_intar2** completed or not).
- One controller can only drive one I-8091 to move by **M_intln2**, **M_intcl2**, **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2**, **M_intcl2**, **M_intar2** at the same time is not possible.

M_intstp **Test X-Y plane moving command**



To test the below 3 commands completed or not.

M_intln2 , **M_intcL2** , **M_intar2**

It will return FALSE for interpolation command completed while return TRUE for busy - not completed yet.

Return:

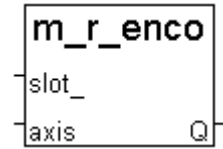
Q_ boolean TRUE: busy , FALSE: completed

NOTE:

1. Only one of **M_intln2**, **M_intcL2** & **M_intar2** command can be called at one time, the other motion moving commands related to the same I-8091 card should not be called unless it is completed. (Please use **M_intstp** to test command of **M_intln2**, **M_intcL2** & **M_intar2** completed or not).
2. One controller can only drive one I-8091 to move by **M_intln2** , **M_intcL2** , **M_intar2** command. Two or more I-8091 cards in the same controller to use **M_intln2** , **M_intcL2** , **M_intar2** at the same time is not possible.

I-8090 encoder commands:

M_r_enco **Reset I-8090's encoder value to 0**



Parameters:

slot_ integer the slot No. where the i8090 is plugged, 0 ~ 7
axis_ integer 1: x-axis, 2: y-axis, 3: z-axis

Return:

Q_ boolean always return TRUE.

Example: demo_27, demo_28, demo_46

第 19 章以太網通訊與安全防護

ISaGRAF PAC 有內建 網路高流量保護功能，能防止諸如 惡意連續的 TCP SYN, TCP FIN, ... 等攻擊 (TCP flood attack) .

19.1: 網路安全防護

有幾種方式可以經由以太網路來存取 WP-8xx7, XP-8xx7-CE6,VP-25W7/23W7, W-8xx7 內的資料.

1. 使用 Modbus TCP 通訊規約, Port 編號 502. (ISaGRAF 軟件 與 SCADA / HMI 就是這樣連的)
2. 使用 ftp (比如在 PC 上執行 Internet Explorer, 鍵盤輸入 “ftp://10.0.0.103”)
3. 使用 telnet (比如在 PC 上開啟一個 “Command” 視窗, 鍵盤輸入 “telnet 10.0.0.103”)
4. 使用 Web server (Wincon 的 Web HMI 功能使用的方式)

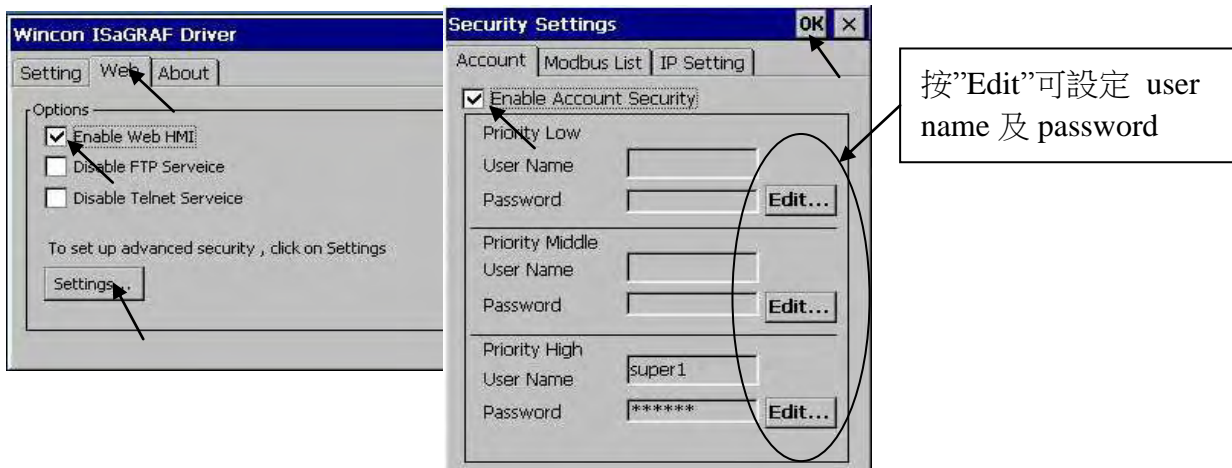
注意:

1. 若是使用 I-8xx7 , I-7188EG, 7186EG, VP-2117, iP-8xx7 只有方式 1 可使用.
2. 若是使用 WP-8xx7, XP-8xx7-CE6,VP-25W7/23W7, W-8xx7 , 要使用 “ftp”, “telnet”, “Web HMI” 與 “Modbus TCP/IP”, 請連您的 PC/HMI 到其上的 LAN1 網口, 並且請使用 ICP DAS 的 Industrial Ethernet Switch 產品 “NS-205” 或 “NS-208” .

以 W-8xx7 為例,為了安全, 您可以開啟 ISaGRAF 驅動程式的對話框, 關閉第 2 及第 3 項功能, 如下.



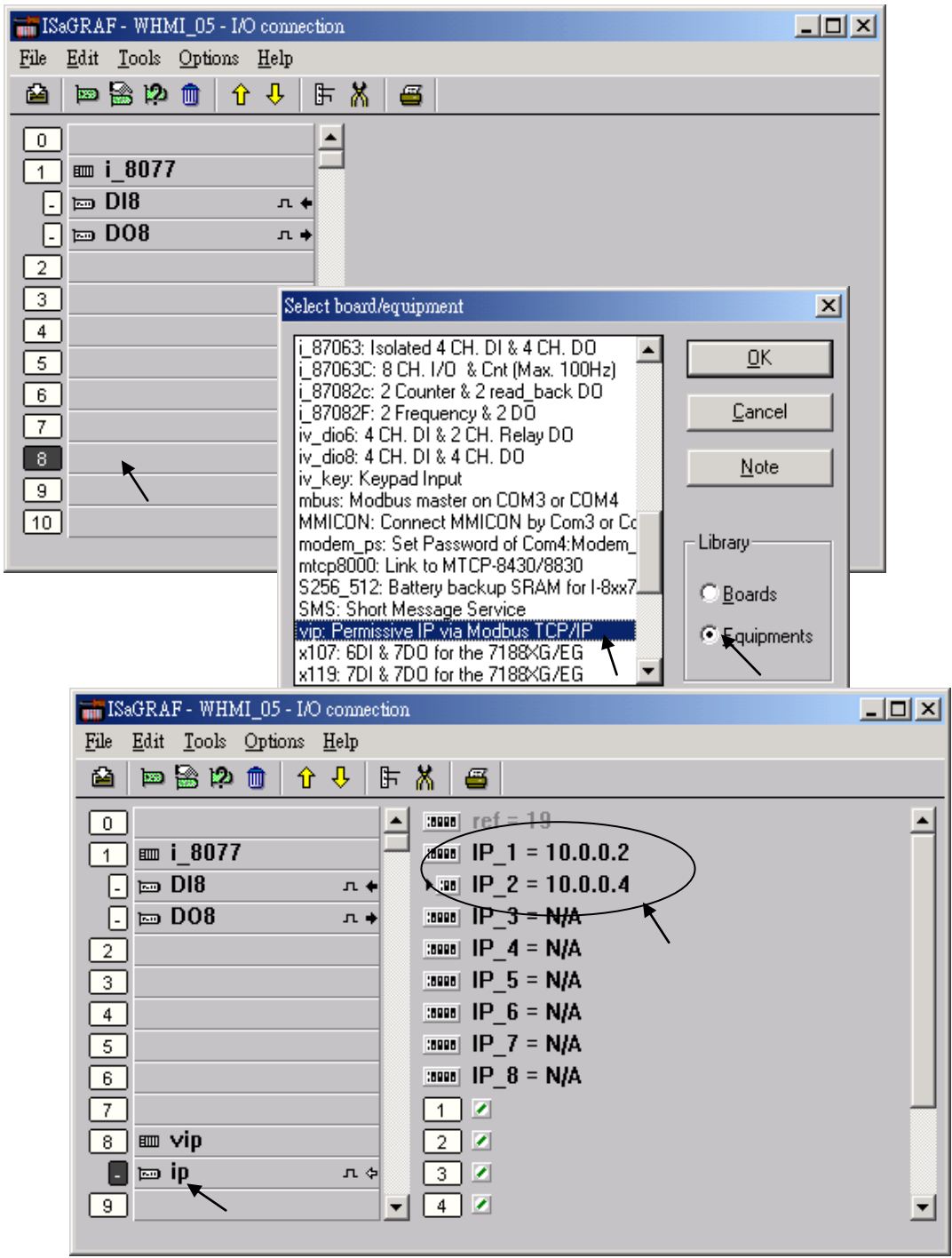
關於第 4 項, 請設好必要的 User name 及 Password 來保護.



關於第一項 Modbus TCP/IP 的保護，使用者可以在控制器上設定最多只開放到 8 組 PC / SCADA / HMI 的 IP 可允許連進來，方法如下。

請在 ISaGRAF I/O 連結視窗內 使用“vip” 虛擬板卡. 最多可設立 8 組 IP 允許用 Modbus TCP/IP 通訊規約連進來. 若控制器內沒有使用“vip”，那表示，任何 IP 都允許用 Modbus TCP/IP 連進來。

如果使用了“vip” 但都設為“N/A” 將沒有任何 IP 允許用 Modbus TCP/IP 連進來。(包含 ISaGRAF 也連不進來)

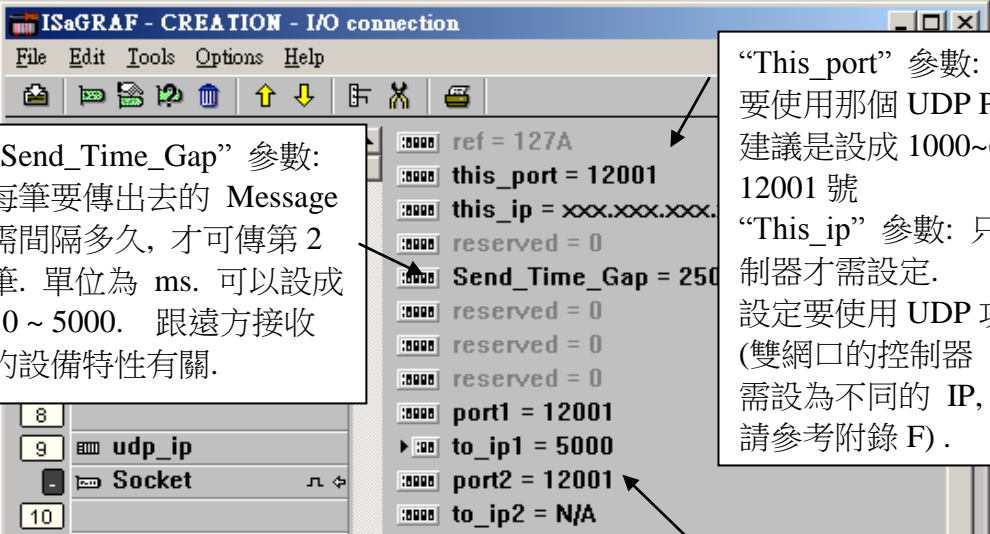


19.2: 使用 UDP 來傳遞或接收訊息 到 PC / HMI 或 設備

若用 WP-8xx, XP-8xx7-CE6, iP-8x47, 可以選 LAN1 或 LAN2 的其中一個網口來使用 UDP 功能. 並用 ICP DAS 的 Industrial Ethernet Switch 產品 NS-205/208.

請確認控制器 driver 是否為 I-7188EG: 2.18 版, I-8437/8337: 3.20 版, W-8xx7: 3.37 版或更高版本.

請在 I/O 連結視窗內連上 1 個 udp_ip, 之後於程式內使用 udp_recv 與 udp_send 來接收或傳遞訊息.



“Send_Time_Gap” 參數: 每筆要傳出去的 Message 需間隔多久, 才可傳第 2 筆. 單位為 ms. 可以設成 10 ~ 5000. 跟遠方接收的設備特性有關.

“This_port” 參數: 設定 UDP Server 是要使用那個 UDP Port 編號來接收訊息. 建議是設成 1000~65535 號. 內定值是 12001 號

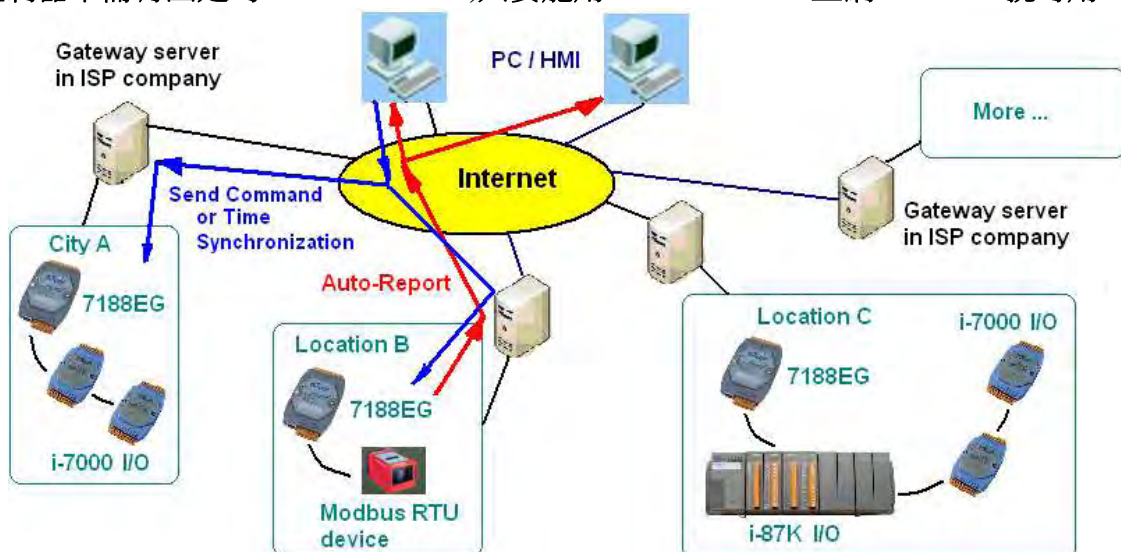
“This_ip” 參數: 只有使用雙網口的控制器才需設定. 設定要使用 UDP 功能在那個 IP 上 (雙網口的控制器的 LAN1 與 LAN2 需設為不同的 IP, 若要啟用 LAN2, 請參考附錄 F).

下方這些參數只有要使用 UDP 傳送功能的需要設, 若不啟用它, 請將 IP 設為 N/A

IP1 到 IP4 參數: 啟用的 UDP 傳送功能是要傳送訊息到遠方那個 IP. WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 最多可設 4 個 IP 連線, 7188EG, 7186EG, iP-8x47 最多只可設前 2 個 IP 連線.

Port1 到 Port4 參數: 對應的遠方 IP1 到 IP4 是使用 UDP 規約的那個 Port 編號?

I-7188EG, μPAC-7186EG, I-8437-80/8837-80, VP-2117, iP-8xx7, WP-8xx7, XP-8xx7-CE6, VP-2xW7 可使用 UDP 功能主動回報“採集”與“控制”資料到 區網或 遠方 Internet 的 PC/HMI. 遠方的 PC / HMI 在收到此資料後, 也可被動對 Wincon 作“時間校正”或下達必要的“控制命令”. 範例程式與說明, 請參考 ISaGRAF FAQ-065. 此應用有一個優點, 位於各個地點的控制器不需有固定的“Internet IP”, 只要能上 Ethernet Port 上網 Internet 就可用.



UDP_Recv:

此函式可用來 測試 與 接收 遠方 PC 或設備所傳過來的 UDP 訊息.

如以下 ST 程式碼:

```
(* 測試是否有 UDP 訊息傳進來 , 若有也會收進來 *)
(* Msg1 宣告為 internal Message 變數 *)
(* 如果 udp_recv()回傳" (2 個單引號中間無資料 為 空字串), 表示沒有資料傳進來 *)
Msg1 := udp_recv() ;
```

注意:

1. WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 與 W-8xx7 的 UDP 功能的接收 buffer 大小為 8192 byte , 包含每個字串的結尾符號 1 byte,而 I-7188EG/7186EG 與 iP-8x47 的接收 buffer 大小為 2048 byte.
2. 如果一次進來太多個字串 導致 接收的 buffer 無法及時消化掉, 新進來的字串會取代掉 最早進來的那筆 字串.

UDP_Send:

此函式可用來 傳送訊息到 遠方 PC 或 設備.

如以下 ST 程式碼:

```
(* TMP 宣告為 Internal Boolean 變數 *)
(* 第 1 個參數: 傳送到那個 UDP 連線, 可以是 1 到 4. 定義在 虛擬 I/O 板 "udp_ip" 內 *)
(* 第 2 個參數: 要傳送出去的 Message *)
(* 若 udp_send()傳回 True 表示成功, False 表示 buffer 滿了或該連線未在 "udp_ip" 內定義好 *)
TMP := udp_send(1, 'Alarm1') ;
```

注意:

1. WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 與 W-8xx7 的 UDP 功能的傳送 buffer 大小為 2048 byte , 包含每個字串的結尾符號 1 byte , 表示 每個 ISaGRAF PLC Scan 最多只能傳送出 2048 個 byte .
iP-8x47, I-7188EG, μPAC-7186EG, VP-2117, i-8x37-80 的傳送 buffer 大小為 1024 byte.
2. 請不要很頻繁的 傳送大量資料出去, 跟 "udp_ip" 內的 "Send_Time_Gap" 參數有關, 若資料太多了, 消化不掉, 會積在傳送的 buffer 內, Buffer 若滿了, 使用 udp_send() 會傳回 False, 表示滿了, 無法再塞入更多的 Message. 每個 PLC Scan 對每個 UDP 連線, 只能消化 1 筆 Message (傳出 1 筆)

範例程式:

請參考 Wincon-8xx7 CD-ROM:\napdos\isagraf\wincon\demo\ **Wdemo_19** 與 **Wdemo_20** 或

ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/

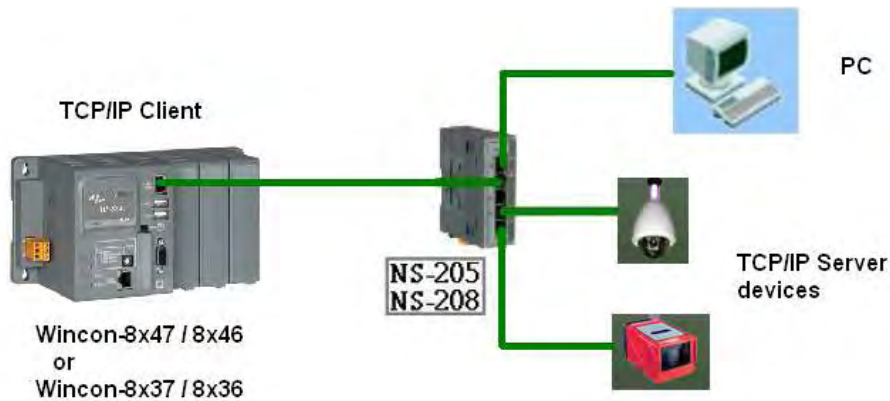
如果您的 ISaGRAF 內找不到 "udp_ip", "udp_recv" 與 "udp_send" 請到以下網址下載

<http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> "ICP DAS Utilities For ISaGRAF.zip". 並執行其內的 "Setup.exe" 來安裝到您 PC 的 ISaGRAF 軟件內.

測試工具: 有個好用的工具程式可在 PC 的 "Command" 視窗上執行來接收 Wincon-8xx7 傳出的 UDP 訊息. 位於 W-8xx7 CD-ROM:\napdos\isagraf\some_utility\udp_test\udp.exe 或 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/some_utility/udp_test/ 內

19.3: 使用 TCP / IP Client 功能來 傳送 / 接收 / 主動回報 訊息

Wincon-8xx7 / 8xx6 從它的 驅動程式 3.37 版起 支持 TCP / IP Client 功能. 請訪問以下網址取得 <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm> . 若您的 ISaGRAF 軟件上找不到 虛擬 I/O 板 "Tcp_clie" 與 , 請到 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 下載 "ICP DAS Utilities For IsaGRAF.zip" . 並執行其內的 "Setup.exe" 來安裝到您 PC 的 ISaGRAF 軟件內



注意: WP-8xx7, XP-8xx7-CE6 與 VP-25W7/23W7 都有支持 TCP/IP client.

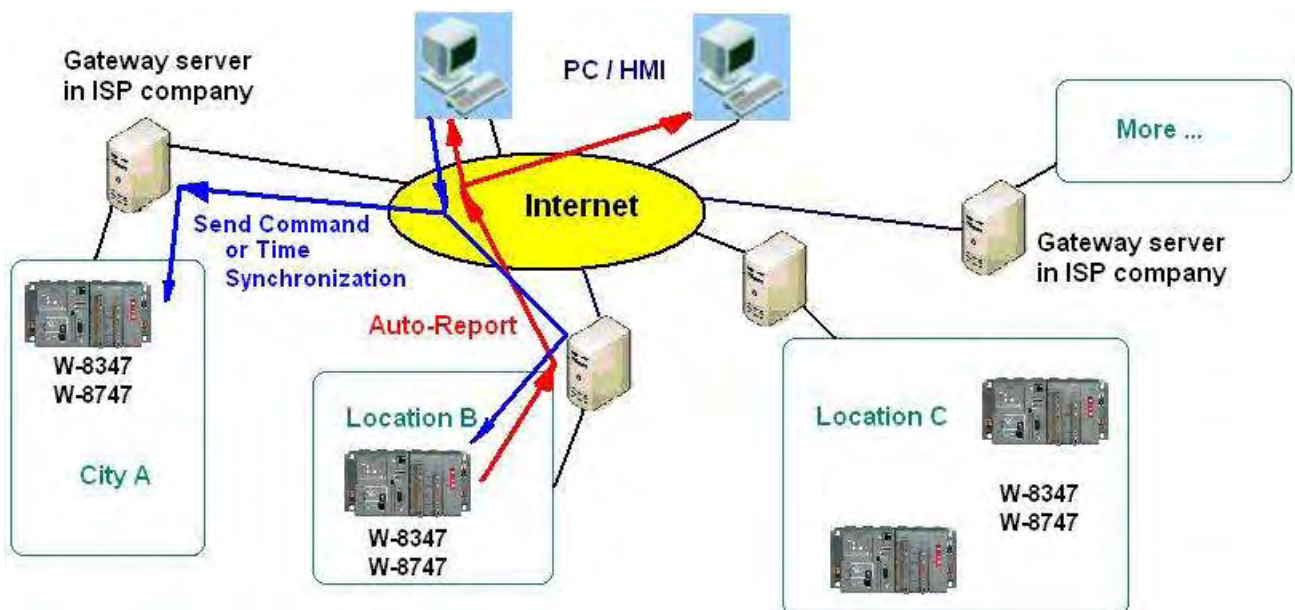
使用 TCP / IP Client 功能的 範例程式為 Wdemo_32.pia, Wdemo_33.pia 與 Wdemo_60.pia (可到以下網址下載 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

注意: 要跟 W-8xx7 的 TCP/IP Client 連線的遠方的 PC 或設備需支持 TCP/IP Server 功能才連的上.

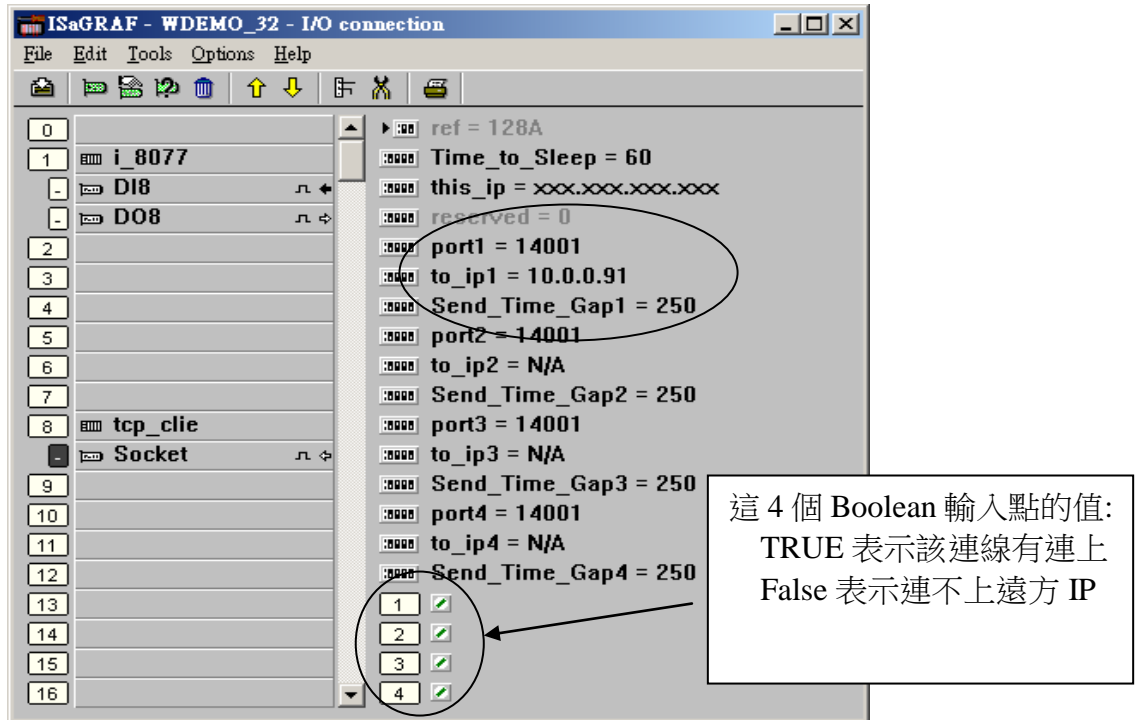
***** 好用的應用 *****

(更多的說明與範例程式, 請參考 www.icpdas.com – FAQ – Software – ISaGRAF - 064)

Wincon-8347 與 W-8747 可使用 TCP / IP Client 連線功能 主動回報 “現場採集資料” 或 “控制資料” 到遠方的 PC / HMI . 遠方的 PC / HMI 在收到此資料後, 也可被動對 Wincon 作 “時間校正” 或下達必要的 “控制命令”. 此應用有一個很大的優點, 位於各個地點的 Wincon 不需有固定的 "Internet IP", 只要能用 Ethernet Port 上網 Internet 就可使用.



要使用 TCP / IP Client, 請先在 I/O 連結視窗內連上 “Tcp_clie”. 最多可連 4 個遠方 IP.



“Time_to_sleep” 參數: 值可以是 10 到 600, 或 0. 單位為秒. 設為 0 表示, 永遠與遠方 TCP / IP Server 設備 保持連線 (除非是通訊有問題). 設為 10 到 600 秒, 表示若經過了這段時間以上沒有任何資料 傳出 或 接收, 自動先關閉 TCP / IP 連線. 但只要有 新訊息要送出, 就會自動再連上.

“this_ip” 參數: 只有使用雙網口的控制器 (WP-8xx7, XP-8xx7-CE6, Wincon-8x47) 才需設定. 設定要使用 TCP / IP Client 功能在那個 IP 上

“port1” 到 “port_4” 參數: 設定成跟遠方的 TCP/IP Server 設備 所使用的 port 編號一樣.

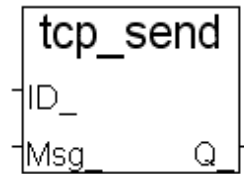
“to_ip1” 到 “to_ip4” 參數: 遠方的 TCP/IP Server 設備的 IP 地址. 若不使用, 請設為 “N/A”

“Send_Time_Gap1” 到 “Send_Time_Gap4” 參數: 每筆要傳出去的 Message 需間隔多久, 才可傳第 2 筆. 單位為 ms. 可以設成 10 ~ 5000. 跟遠方接收的設備特性有關.

設好 “Tcp_Clie” 的所有參數後, 就可在程式內使用 “Tcp_send” 或 “Tcp_recv” 函式 來 傳送 或 接收 TCP / IP 訊息.

Tcp_send

TCP Client 傳送 Message 到遠方的 TCP / IP Server (可以是 PC 或 設備)



注意:

每個 WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 與 W-8xx7 的 TCP / IP Client 連線的 傳送 buffer 大小為 4096 byte. 如果傳送太頻繁 使得 PAC 來不及 消化掉, 導致 傳送 buffer 滿的話, 會把最早的 1 個或數個未被送出的 Message 刪掉, 來容納新的 TCP / IP Message. 每個 PLC Scan 每個連線最多只能傳出 1 個 TCP / IP Message. 傳送的時間 間隔 需在 “Tcp_clie” 的 “Send_Time_Gap” 內定義好 (跟遠方設備特性有關)

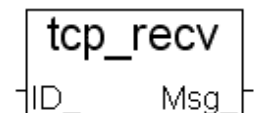
ID_	Integer	傳送到那個 TCP / IP 連線, 可以是 1 到 4 (各連線參數定義於 “Tcp_Clie” 虛擬板卡內)
Msg_	Message	要傳出的 Message

回傳值:

Q_	Boolean	True: 表示成功, False 表示 有錯誤發生 (比如, ID_設錯, 設成比 4 大, 或該連線沒在 “Tcp_clie” 內定義好 .
----	---------	---

Tcp_recv

TCP Client 接收遠方 TCP / IP Server (PC 或設備) 送過來的 Message



注意:

WP-8xx7, XP-8xx7-CE6, VP-25W7/23W7 與 W-8xx7 的 TCP / IP Client 的接收 buffer 大小為 4096 byte. 如果太多資料傳進來, 來不及消化掉導致 buffer 滿了, 最早進來的 1 筆或數筆 Message 會被刪掉, 用來容納新進來的 Message.

ID_	Integer	傳送到那個 TCP / IP 連線, 可以是 1 到 4 (各連線參數定義於 “Tcp_Clie” 虛擬板卡內)
-----	---------	---

回傳值:

Msg_	Message	收到的 Message. 若回傳" (2 個單引號中間無資料 為 空字串), 表示 沒有資料傳進來
------	---------	--

有個好用的 “Tcp_Server” 工具程式 可將 PC 模擬(仿真) 成 TCP / IP Server. 它放在

Wincon-8xx7 CD-ROM: \napdos\isagraf\some_utility\ 或

ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/some_utility/ 內

請把整個 “Tcp_server” 目錄 Copy 到你的 PC 內, 之後開啟一個 “Command” 視窗, 鍵盤輸入 “tcp3 <port_No>” 來執行它, 比如 “tcp3 14001”

這個工具程式 會等 TCP / IP Client (W-8xx7) 連進來, 然後接收它的訊息並顯示, 同時會 回覆 相同的訊息給 Wincon-8xx7 的 TCP / IP Client.

使用 TCP / IP Client 功能的 範例程式為 Wdemo_32.pia, Wdemo_33.pia 與 Wdemo_60.pia (可到以下網址下載 ftp://ftp.icpdas.com/pub/cd/wincon_isagraf/napdos/isagraf/wincon/demo/)

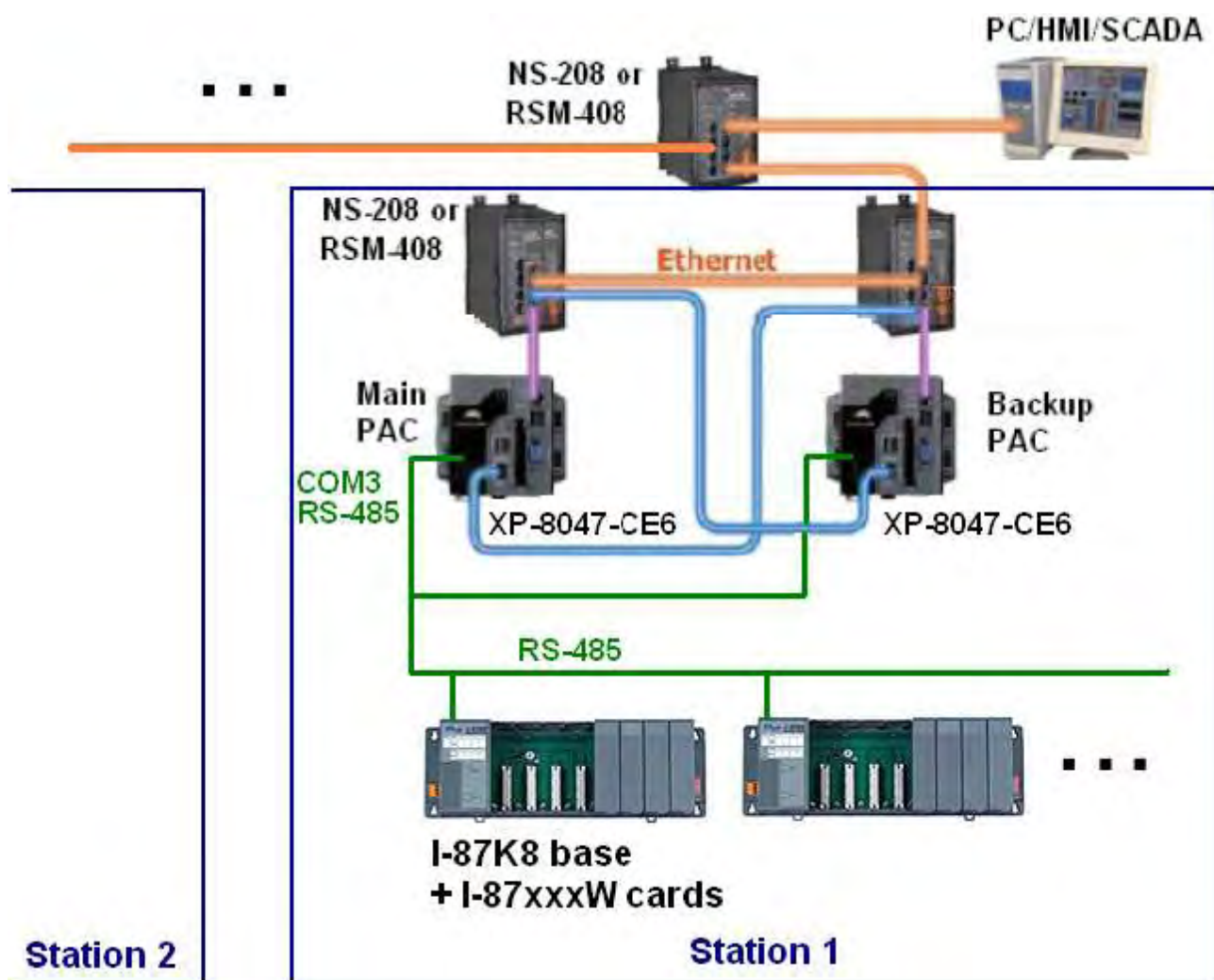
第 20 章 備援系統 (冗餘系統)

20.1: XP-8xx7-CE6 備援系統

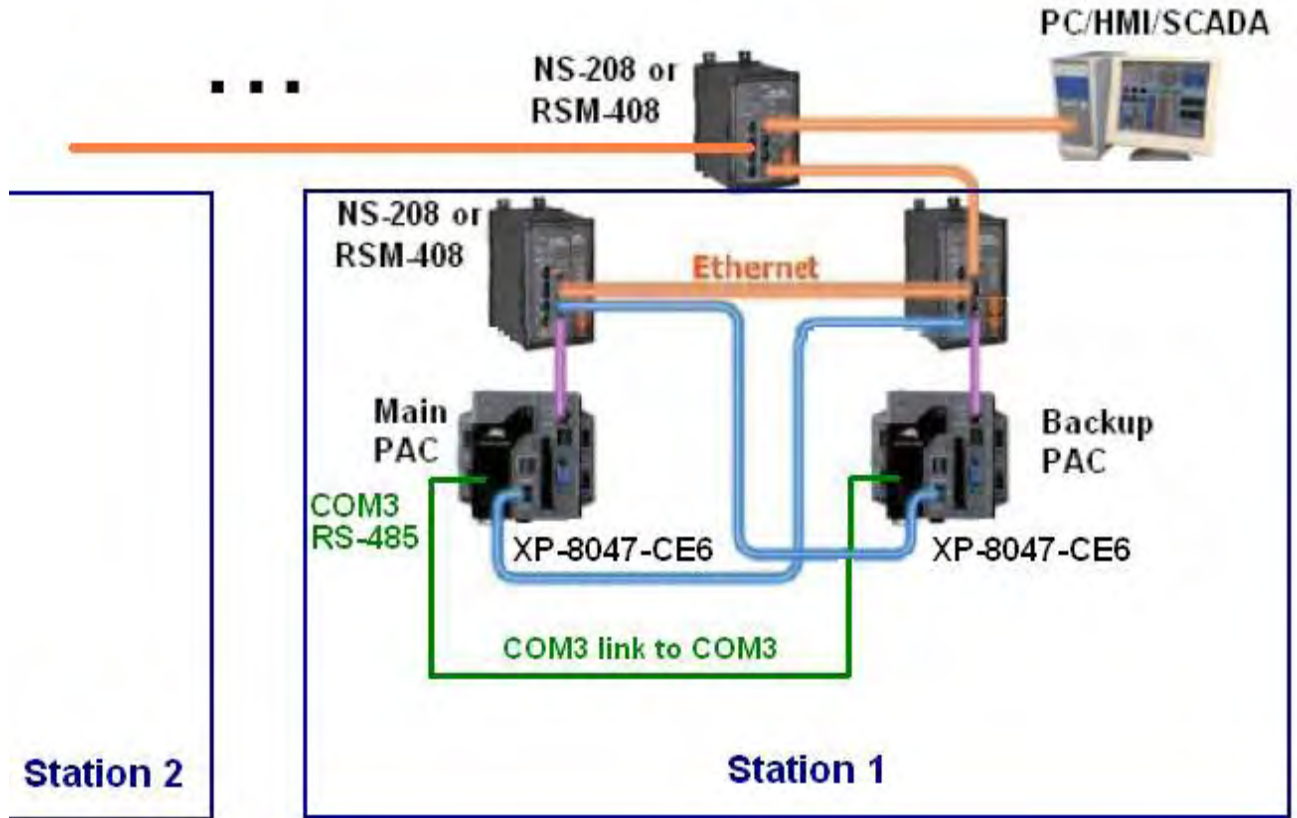
詳細資料請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-138 .

XP-8xx7-CE6 備援系統有個優點是 WP-8xx7 備援系統沒有的, PC / HMI / 圖控軟體 (組態軟體) 只需設定去連接 XP-8xx7-CE6 備援系統上的一個 IP 地址就好 (即 Active_IP1 那個地址). 反觀若是要去連接 WP-8xx7 備援系統, PC / HMI / 圖控軟體必需設定去連接 2 個 IP 地址, 當連接第一個 IP (Active_IP1) 發生連不上時, PC / HMI / 圖控軟體必需自動改變去連接第 2 個 IP (Active_IP2). 這對很多圖控軟體來說, 有些是辦不到, 有些是很麻煩 (要建 2 組 資料 Tag).

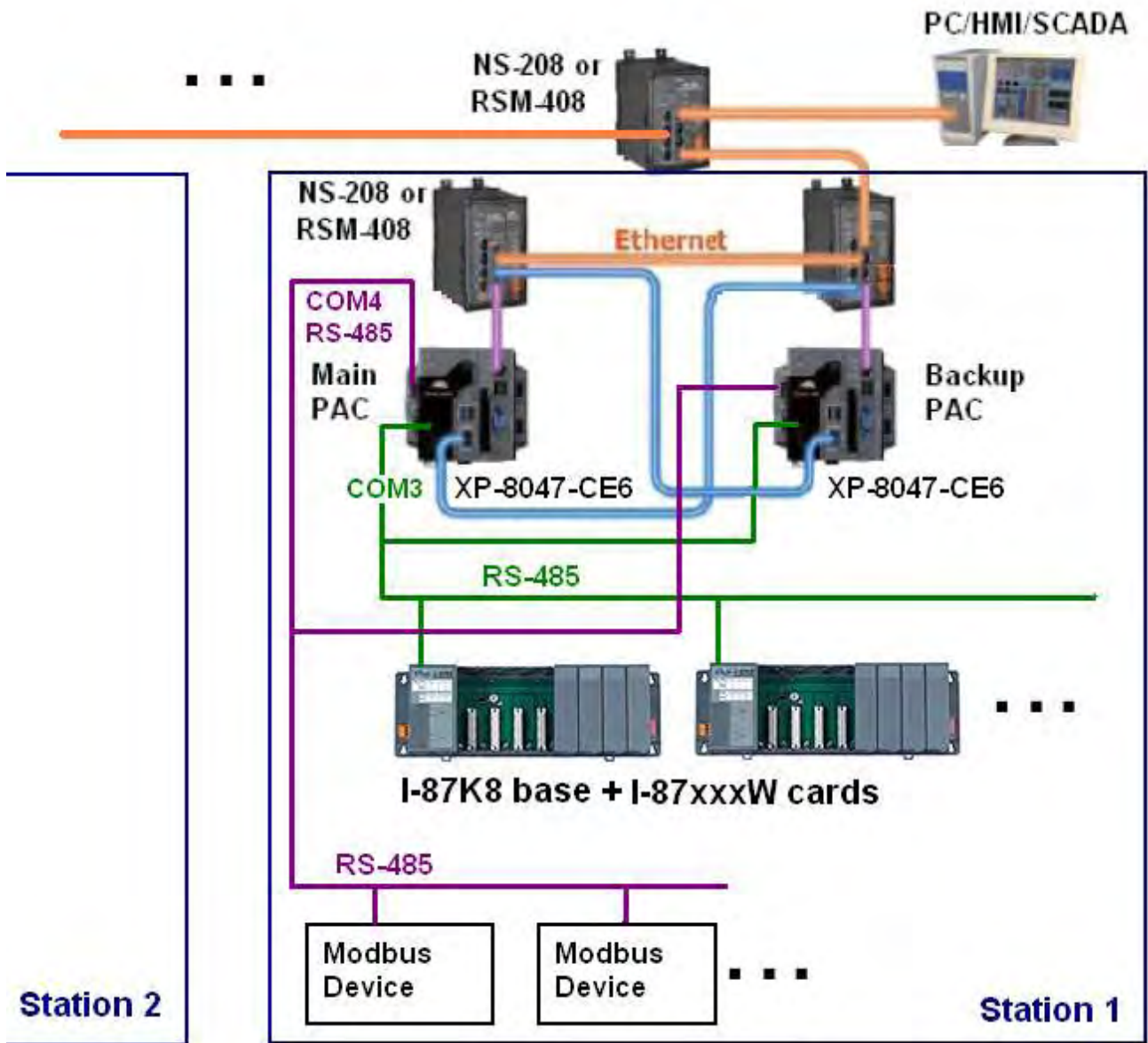
第 1 種應用是採用 2 台 XP-8xx7-CE6 來連接 1 個到多個 I-87K8 擴展單元(每個 I-87K8 上可以插上最多 8 張 I-87xxxW 卡片). 可以多個控制站(station) 組合在一起, 如下圖,每個 station 內可包含 1 個或 2 個 NS-208 (或 RSM-208) 與 2 台 XP-8xx7-CE6 與 1 個或多個 I-87xxxW 卡片.



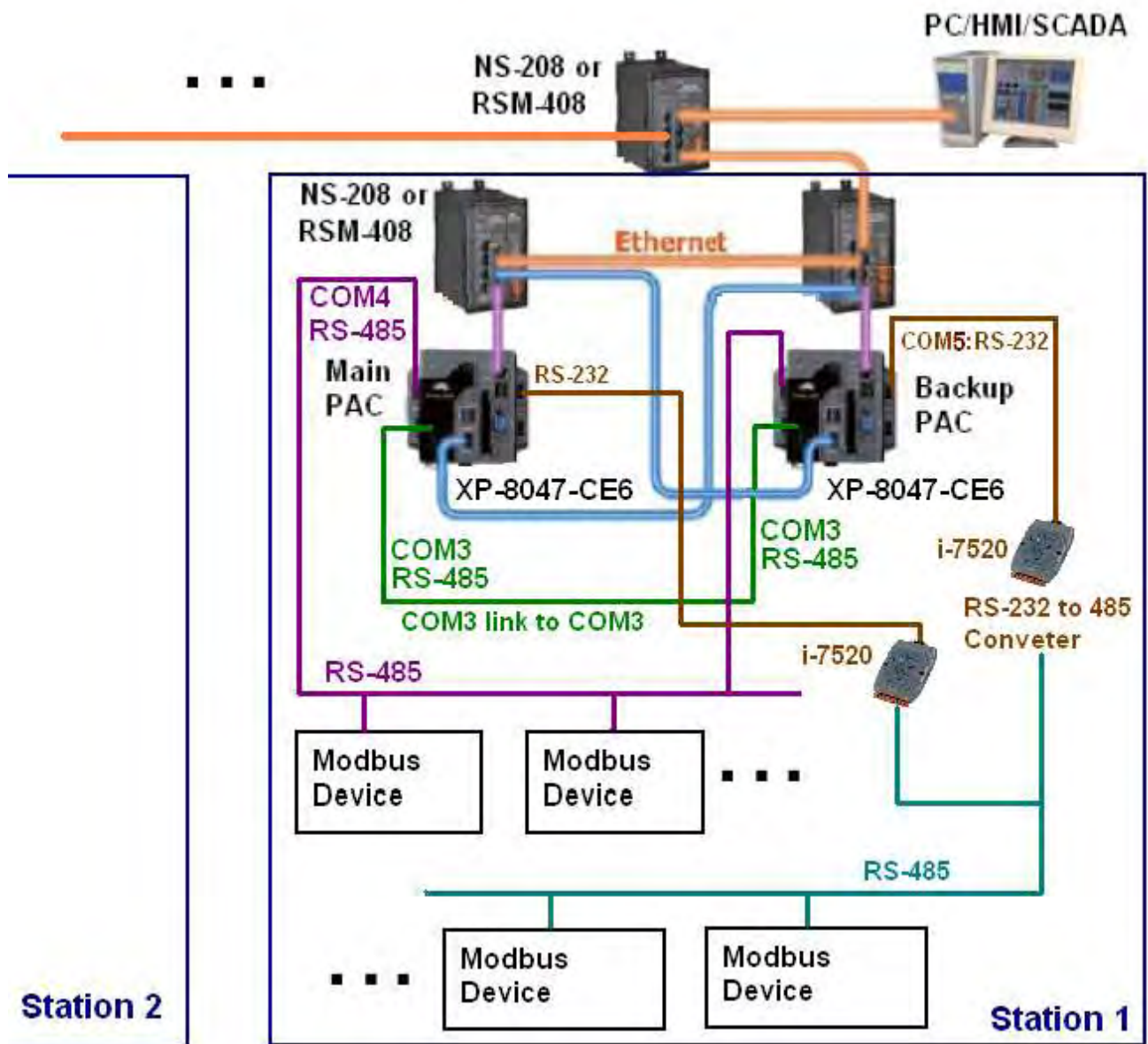
第2種應用是採用2台 XP-8xx7-CE6，可以不接 I/O 或 使用其它 port 去接其它 device (例如, 2台 XP-8xx7-CE6 的 COM4:RS-232 都各別接一顆 I-7530 轉成 CAN 訊號再去並接其它的 CAN/CANopen device)，此應用必需將 2 台 XP-8xx7-CE6 的 COM3:RS-485 並接在一起. 每個 station 內的 NS-208 (或 RSM-208) 也可以只用一個.



第 3 種應用是採用 2 台 XP-8xx7-CE6 來連接 1 到多個 I-87K8 擴展單元, 另外再使用 RS-485 port 去連接 Modbus RTU device (或 Modbus ASCII device). 每個 Station 內的 NS-208 (或 RSM-208) 也可以只用一個.



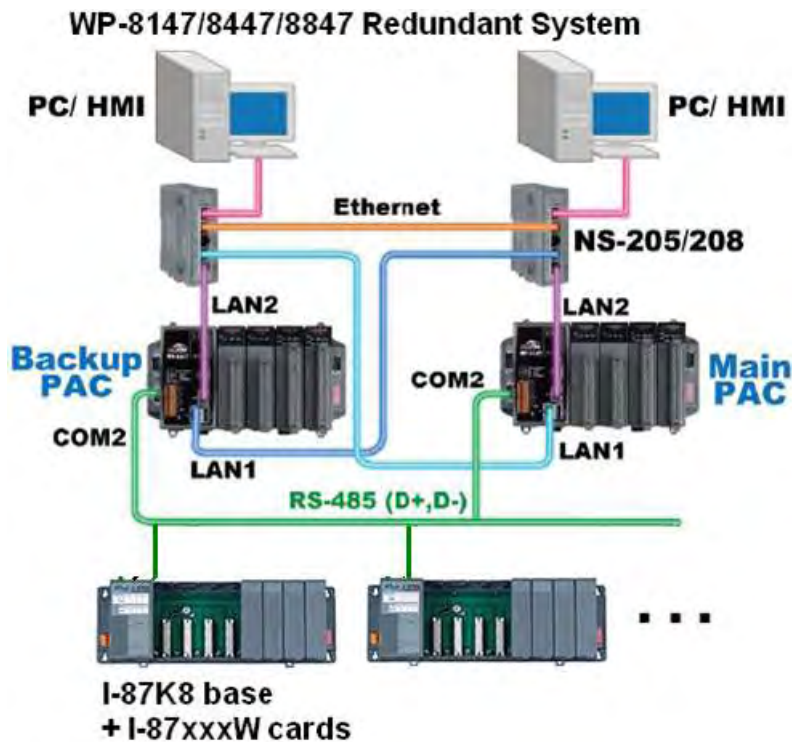
第4種應用是採用2台 XP-8xx7-CE6，使用 RS-485 port 去連接 Modbus RTU device (或 Modbus ASCII device)，此應用必需將 2 台 XP-8xx7-CE6 的 COM3: RS-485 並接在一起。每個 Station 內的 NS-208 (或 RSM-208) 也可以只用一個。



20.2: WP-8xx7 備援系統

詳細資料請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-093 .

XP-8xx7-CE6 備援系統有個優點是 WP-8xx7 備援系統沒有的, PC / HMI / 圖控軟體 (組態軟件) 只需設定去連接 XP-8xx7-CE6 備援系統上的一個 IP 地址就好 (即 Active_IP1 那個地址). 反觀若是要去連接 WP-8xx7 備援系統, PC / HMI / 圖控軟體必需設定去連接 2 個 IP 地址, 當連接第一個 IP (Active_IP1) 發生連不上時, PC / HMI / 圖控軟體必需自動改變去連接第 2 個 IP (Active_IP2). 這對很多圖控軟體來說, 有些是辦不到, 有些是很麻煩 (要建 2 組 資料 Tag).



第 21 章連接 M-7000 系列 Modbus RTU I/O

ICP DAS 的 ISaGRAF 控制器從以下 驅動版本 起有支持 M-7000 系列 RS-485 I/O.

Controller	Driver version
W-8037 / 8337 / 8737 / 8036 / 8336 / 8736	3.35 或 更高版本
W-8047 / 8347 / 8747 / 8046 / 8346 / 8746	3.35 或 更高版本
I-8417 / 8817 / 8437 / 8837	3.19 或 更高版本
I-7188EG / 7188EGD	2.17 或 更高版本
I-7188XG / 7188XGD	2.15 或 更高版本
µPAC-7186EG, iP-8xx7, VP-2117, WP-8xx7, VP-25W7/23W7, XP-8xx7-CE6	產品 release 之日起

如果您的控制器的驅動版本為較舊版本 請參訪

<http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm> 來下載它們，之後請翻閱其內的 “ReadMe.txt” 或 “Update_w8xx7.pdf” 來更新它們。

M-7000 系列模塊為 RS-485 遠程 I/O 模塊，支持 Modbus RTU slave 通訊規約。產品訊息請訪問 http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm .

使用者可以寫 ISaGRAF 程式 啟用 Modbus RTU Master 功能來連接 M-7000 模塊。關於 Modbus RTU Master 功能說明，請參考第 8 章

WP-8xx7, VP-25W7, VP-23W7 與 XP-8xx7-CE6 的 Ethernet port 有支持 Modbus TCP Master, 請參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-113

有時無法正確連上 Modbus device 時可以參考 http://www.icpdas.com/faq/isagraf_c.htm > FAQ-075 的說明來 排除 (或設定特別的 “Delay_Time”)

連接 M-7000 的範例程式放在 WP-8xx7 CD-ROM:\napdos\isagraf\wp-8xx7\demo\ 內 或 <ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/>

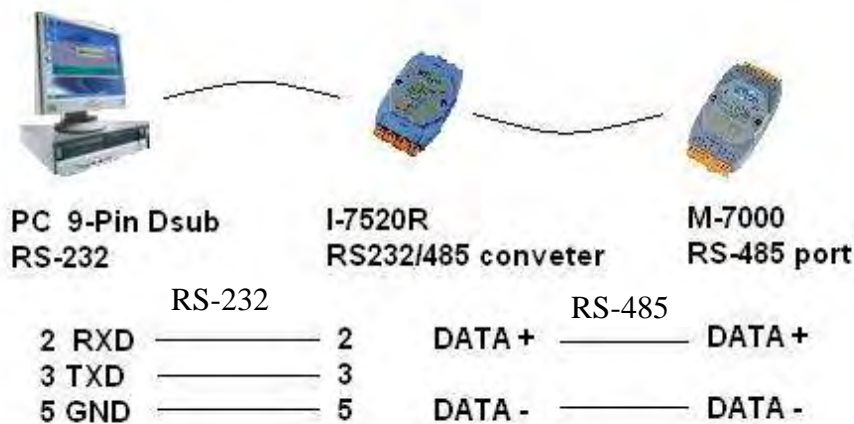
這些範例都只使用 COM3 port 來連 M-7000 模塊。您可以更改為其它的 RS-485 Port.

Wpdmo_41	COM3 連 1:M-7053D (16-Ch. D/I) + 2:M-7045D (16-Ch. D/O)
Wpdmo_42	COM3 連 1: M-7053D 來取得 D/I 計數值 (16-bit, 0 ~ 65535)
Wpdmo_43	COM3 連 1:M-7017R (8-Ch. A/I) + 2:M-7024 (4-Ch. A/O)
Wpdmo_44	COM3 連 1: M-7017RC (8-Ch. 電流輸入)
Wpdmo_45	COM3 連 1: M-7019R (8-Ch. 多種類 A/I, 可以是 thermocouple 或 電壓輸入 或電流輸入) 來取得 溫度值 (攝氏 或 華氏度)
Wpdmo_46	COM3 連 1:M-7080 (2-Ch 高數計數器 或 頻率輸入)

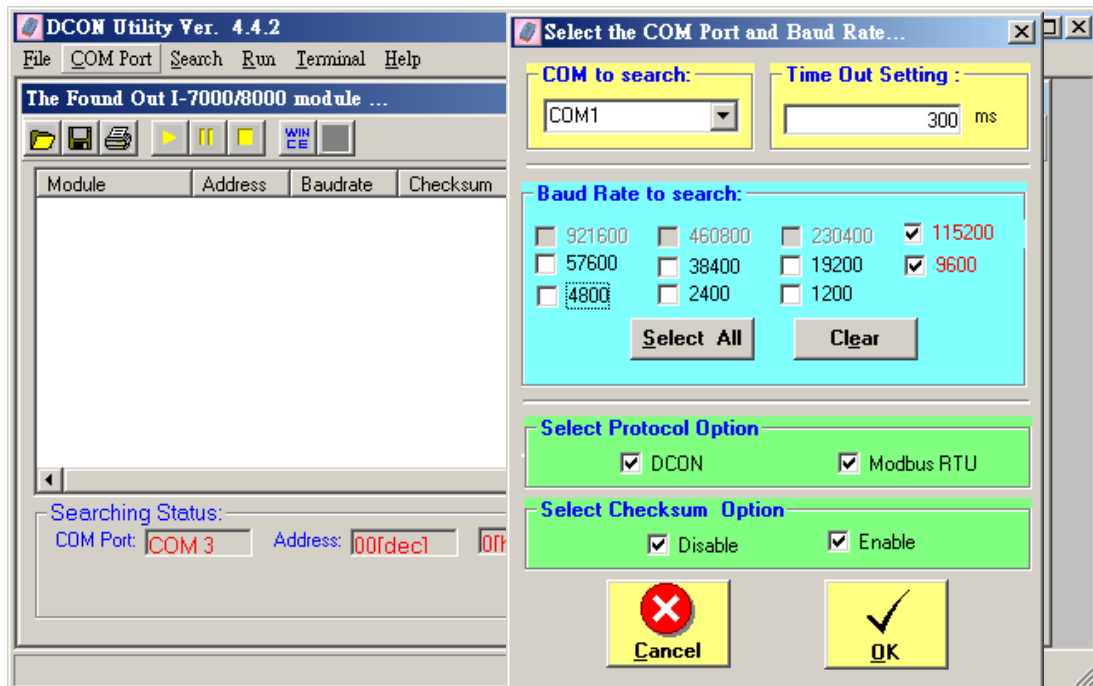
21.1: 使用 DCON Utility 來對 M-7000 模塊 作 初始設定

在開始寫 ISaGRAF 程式之前, 需先在 PC 上執行 “DCON utility” 來規劃設定每一個 M-7000 模塊的 “Slave No” (或稱呼為 “Address”) 與 “Baudrate” 與 AI / AO 接點的 “輸出/輸入 Type Code 設定”. M-7000 的 “Protocol” 設定必須設成 “Modbus”. 若您找不到 “Dcon Utility” 或想更新為新版本, 請訪問 <http://www.icpdas.com/download/7000/7000.htm> 來下載, 安裝.

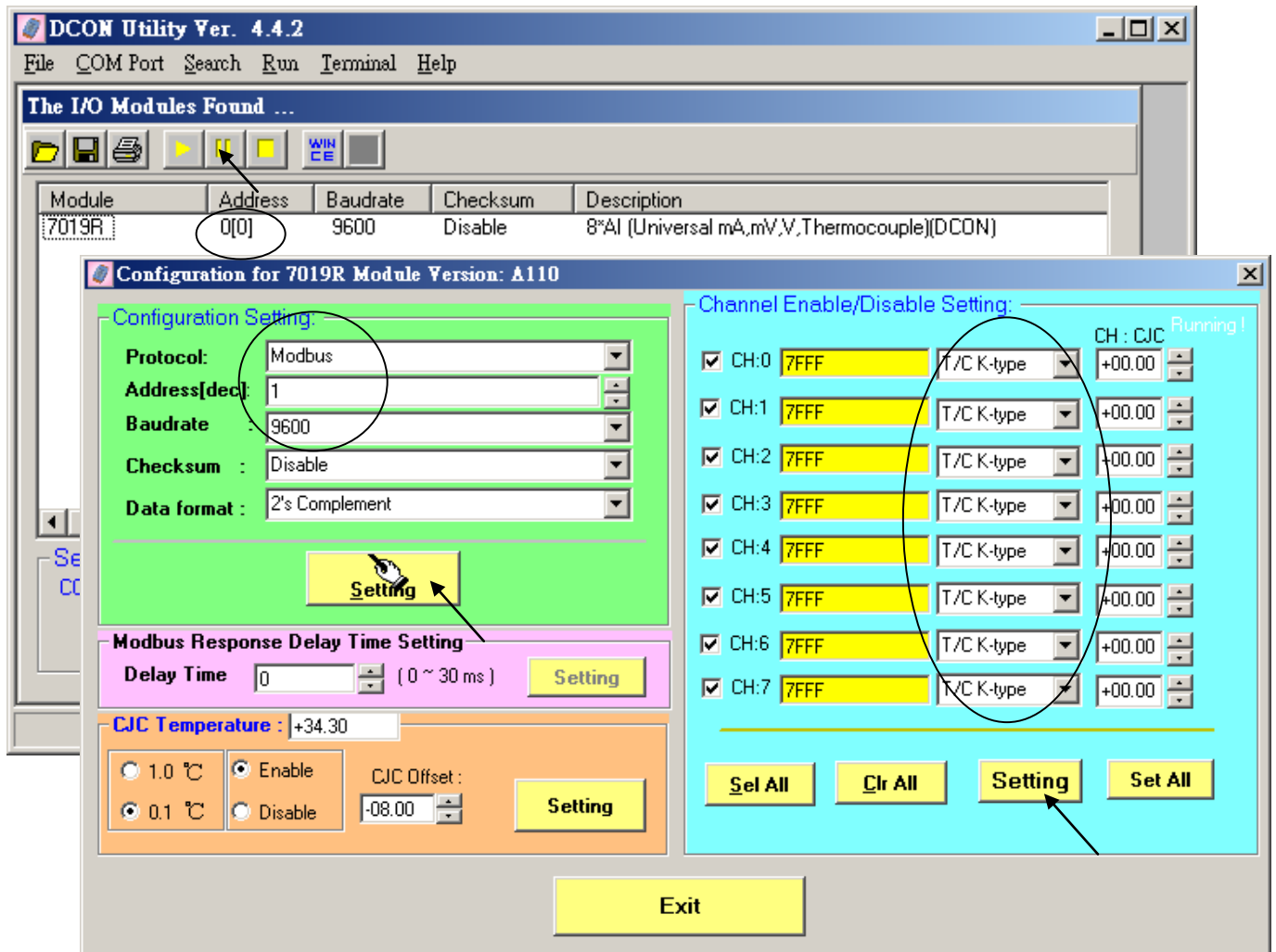
步驟 1: 將單獨的一顆 M-7000 關機, 之後從 PC 的 RS232 COM Port (COM1 或 COM2 或其它) 連接一條 RS232 通訊線到一顆 RS232/485 訊號轉換模塊, 比如 I-7520R http://www.icpdas.com/products/Industrial/communication_module/communication_list.htm, 之後再連接 RS485 通訊線到那顆 M-7000. **請將 M-7000 模塊接線設成 “INIT” 狀態 (將它端子上的 INIT*接腳 跟 GND 接腳 短接).** M-7000 模塊在 INIT 狀態時, Address 會是 0, baud rate 則是 9600. (有些 M-7000 的背面有一個 “Init – Normal” 的 dip 開關來取代 “INIT*” 接腳).



步驟 2: 供電給該 M-7000 模塊. PC 執行 “Dcon utility”, 滑鼠點選 “COM Port” 來切換成 PC 使用的那個 COM Port, Baud Rate 可以選多個以利搜尋 (9600 與 115200 請務必選取), “DCON” 與 “Modbus RTU” 選項也需選取, Checksum 最好 “Disable” 與 “Enable” 都選.



步驟 3: 那顆單獨連接的 M-7000 模塊應該會被 DCON Utility 找到, Address 應是 0 (因為它在 INIT 狀態), 找到後點選 “Stop” 來停止搜尋. 請將 “Protocol” 設為 “Modbus”, 之後設定你想使用的 “Address” (Slave No) 與 “Baudrate”. 如果你的 M-7000 是類比 AI 或 AO (模擬量輸入或輸出), 請設定你想使用的 每個接點的 Type 與 Range 於右側, 之後請按 “Setting” 來下命令給這顆 M-7000 模塊.



步驟 4: 將 M-7000 關電. 解除 “初始狀態” (將 INIT*接腳 與 GND 接腳 斷開). 之後再供電給它 . 執行 DCON utility 再搜尋一次, 找到後 檢查 設定對不對, 若設的不對, 請修改過來.

如果使用的 M-7000 模塊是 M-7041 或 M-7044 或 M-7050 或 M-7053 或 M-7060 或 M-7063 或 M-7065 (或 M-7041D 或 M-7044D 或 M-7050D 或 M-7053D 或 M-7060D 或 M-7063D 或 M-7065D), 請務必操作 步驟 5. 如果不是這些型號, 那初始設定就完成了 .

注意:

1. 每個在同一個 RS-485 連線上的 M-7000 模塊, 都要設為不同的 “Address”, 可以是 1 到 247, 且 “Baudrate” 都需設成一樣, 才能正確的工作. RS-485 距離越長, Baudrate 要設的小一點, 超過 100 公尺 (Meter), 建議設成 19200 或 9600.
2. 使用者請另外參考 M-7000 模塊包裝盒內所附的文件, 或訪問 http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm 來取得更多的 軟 / 硬 體說明, 比如像 “Analog Input Type and Data Formate Table” 的訊息 (Type code setting).

重要的步驟 5:

完成步驟 1 到 4 後，若你使用的 M-7000 模塊編號是 下表所列的型號，須使用 DCON utility 來下達以下命令給它，將 Digital input 接點訊號 反相。

01 46 29 01 (4-byte , 每個 byte 是 2 個 16 進位數字)

第一個 byte 為該 M-7000 模塊初始設定完後所設的 Address 編號，可以是 01, 02, 03, ..., 0F, ... 到 F7. 接下來 3 個 byte “46 29 01” 就不能變更。

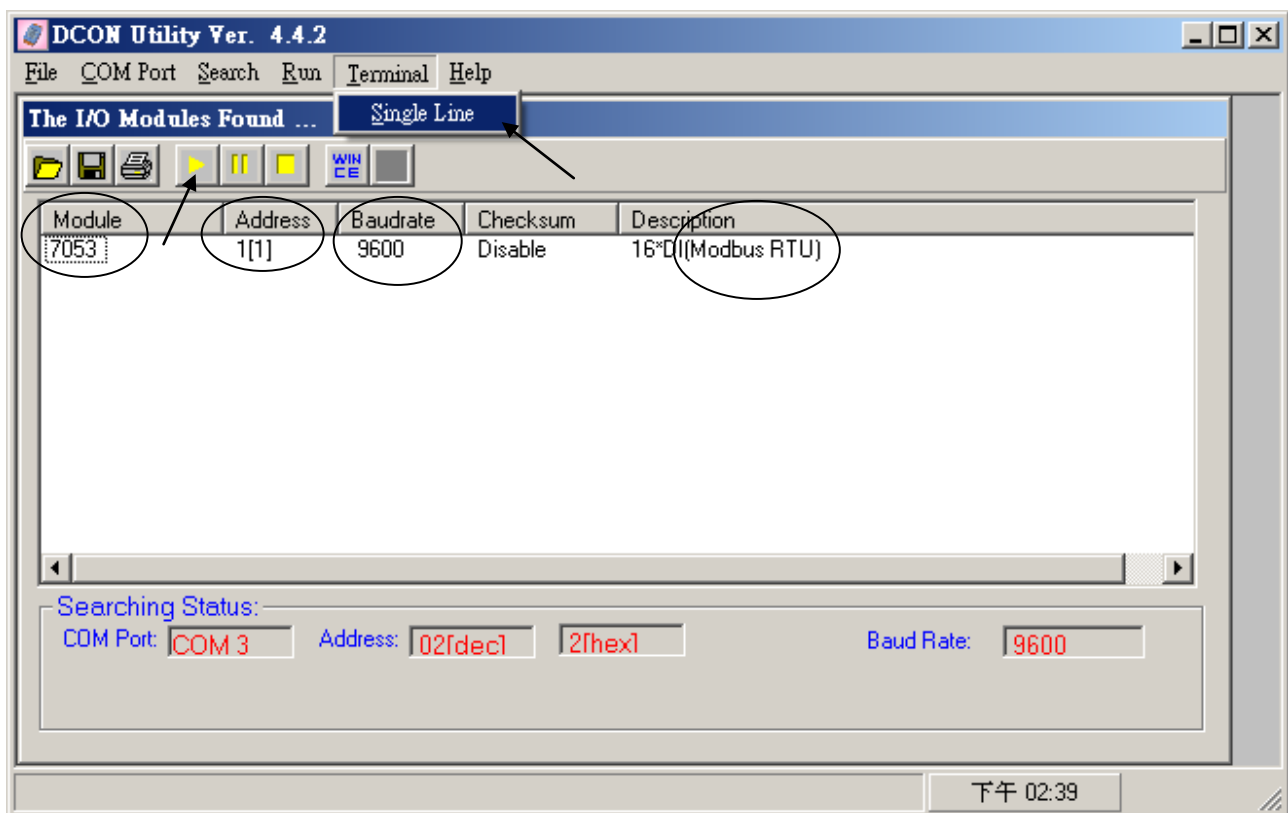
需作反相設定的 M-7000 模塊型號

M-7041	M-7044	M-7050	M-7053	M-7060	M-7063	M-7065
M-7041D	M-7044D	M-7050D	M-7053D	M-7060D	M-7063D	M-7065D

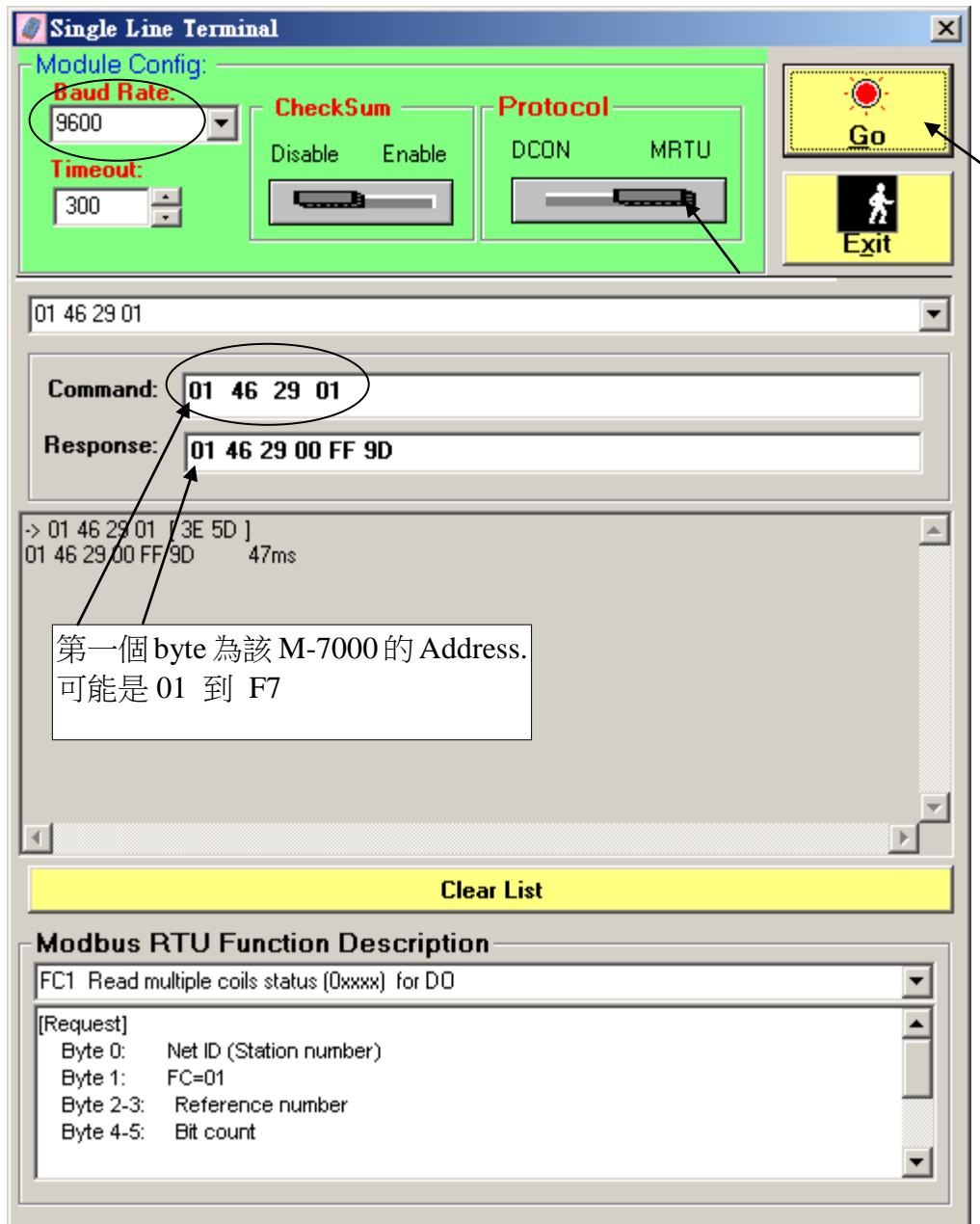
請不要把以上的 4-byte 命令下給非表格內列出的型號。

如何進行反相設定:

當步驟 4 完成後，再供電給 M-7041 或 M-7044 或 M-7050 或 M-7053 或 M-7060 或 M-7063 或 M-7065. 執行 DCON utility 來搜尋該模塊. 若找到，則按停止查詢 (找到的 Address 應該不是 0, 因為不是在 INIT 狀態下). 務必確認型號是上方表格內的型號，才可作 D/I 反相設定. 確定是之後，點選 “Terminal” 內的 “Single Line”



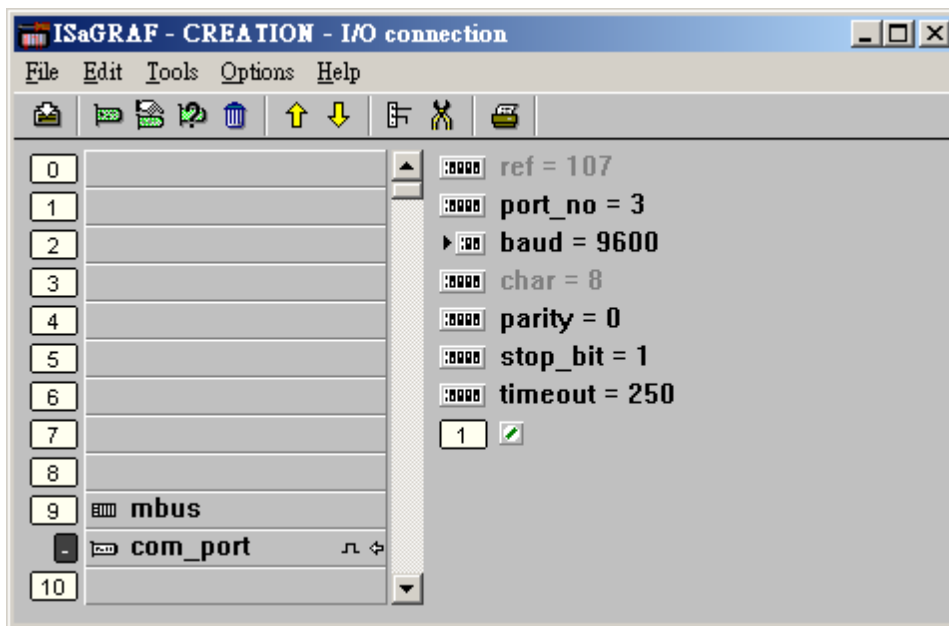
選取正確的 Baud Rate, Protocol 需選 “MRTU”. 之後輸入對應的 4-byte 命令, 第一個 byte 需是該 M-7000 模塊的 Address 編號, 可以是 01, 02, 03, ..., 0F, ... 到 F7. 接下來 3 個 byte 需為 “46 29 01”. 之後按下 “Go”. 如果有收到回復訊息為 “01 46 29 ...”, 表示設定成功. 關閉該 M-7000 的供電. 如此初始設定就算成功了.



21.2: 編寫操控 M-7000 模塊 的程式

重要：如果您使用的 M-7000 模塊是 M-7041 或 M-7044 或 M-7050 或 M-7053 或 M-7060 或 M-7063 或 M-7065 (或 M-7041D 或 M-7044D 或 M-7050D 或 M-7053D 或 M-7060D 或 M-7063D 或 M-7065D)，請務必操作步驟 5，不然您會發現這些模塊的 D/I 訊號都會是反的。

要寫 ISaGRAF 程式來連接 M-7000，請在 ISaGRAF IO connection 內連上“mbus”如下圖。請設好對應的“port_no”，“baud”與“timeout”。“timeout”內定值為 500 ms，但連 M-7000 I/O 時可改設為 250ms，關於 mbus 的說明，請參考第 8.1 節。



之後請建立一個 階梯圖 或 功能方塊圖 程式來操控每一個 M-7000 的 IO 接點. ICP DAS ISaGRAF 控制器可以使用以下的功能方塊來 讀/寫 M-7000 模塊.

Mbus_R	<ol style="list-style-type: none"> 請設定“CODE_”為 編號 4 來讀取 M-7000 的 Analog Input 或 D/I Counter, 最多達 12 個 word 資料 (每個 word 值為 -32768 ~ +32767). 也可用來讀取最多達 6 個 32-bit 整數資料 (-2,147,483,648 ~ +2,147,483,647), 需配合使用 WD_LONG 方塊將 2 個 word 轉換成 1 個 32-bit 整數 請設定“CODE_”為 編號 2 來讀取 M-7000 的 Digital Input 資料, 需配合使用“WD_Bit”方塊將 1 個 word 轉換成 16 個 Boolean 資料
Mbus_R1	同“MBUS_R”，但多了一個參數“PERIOD_”，單位為秒. 可設成 1 ~ 600. 每過一段 PERIOD 時間 才對 該 M-7000 詢問一次.
MBUS_N_W	使用 Modbus function 編號 6 或 16 來寫出 1~4 個 Word 值 (-32768 ~ +32767) 給 M-7000 的 Analog Output 接點. 當“NUM_W_”為 1 時, 使用 Modbus function 編號 6. 當“NUM_W_”為 2~4 時, 使用 function 編號 16
MBUS_B_W	使用 Modbus function 編號 5 或 15 來寫出 1~4 個 Bit 給 M-7000 的 Digital Output 接點. 當“NUM_W_”為 1 時, 使用 Modbus function 編號 5. 當“NUM_W_”為 2~4 時, 使用 Modbus function 編號 15
MBUS_WB	使用 Modbus function 編號 15 來寫出 1~16 個 Bit 給 M-7000 的 Digital Output 接點

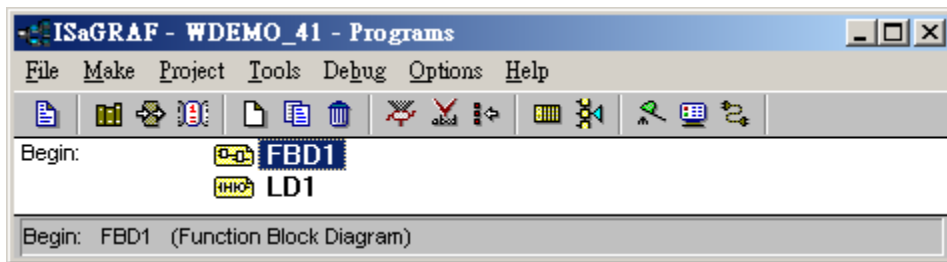
範例 41: 連接 1:M-7053D (16-Ch. D/I) 與 2:M-7045D (16-Ch. D/O)
 (本範例 Project 名稱為 “Wdemo_41”).

請記得要先操作 21.1 節的步驟 1 到 5 來對 M-7053D 作初始設定, M-7045D 只需步驟 1 到 4.

變數宣告:

Name	Type	Attribute	Description
comm_ok1	Bool	Internal	M-7053D 的通訊狀態
comm_ok2	Bool	Internal	M-7045D 的通訊狀態
M7053DI_01 到 M7053DI_16	Bool	Internal	共 16 個變數來對應到 M-7053D 的 16 個輸入點
M7045DO_01 到 M7045DO_16	Bool	Internal	共 16 個變數來對應到 M-7045D 的 16 個輸出點

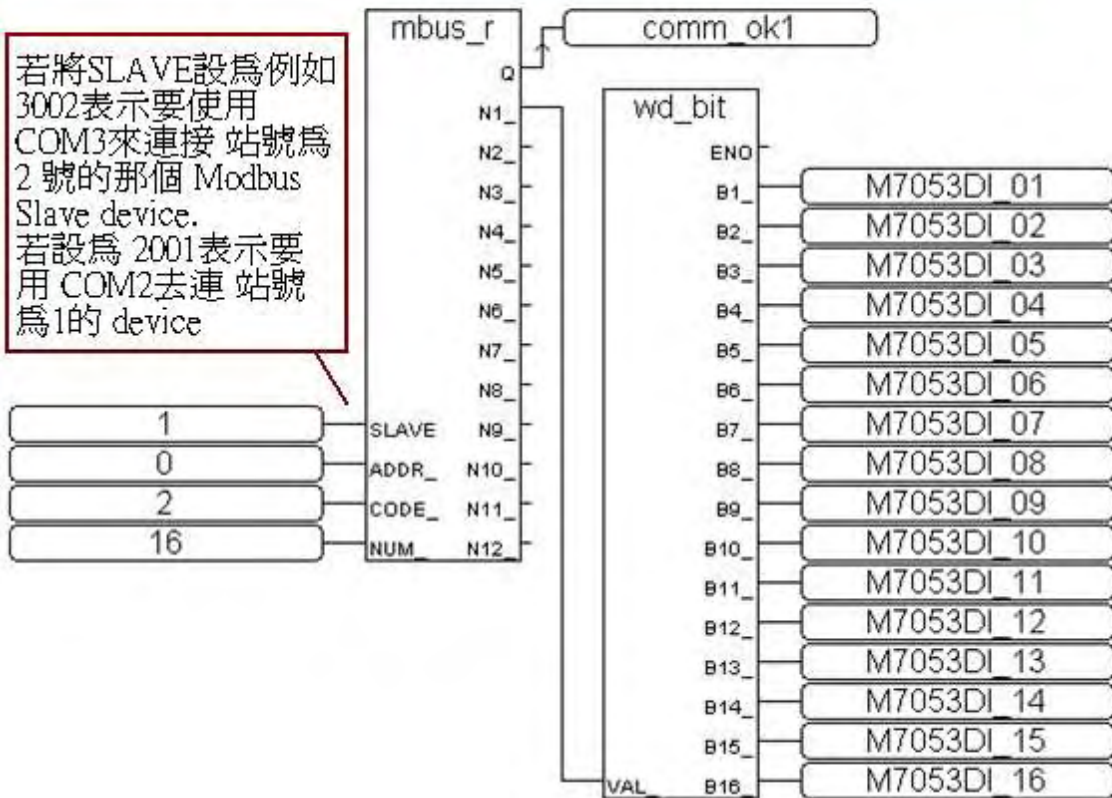
Project 架構: 一個功能方塊圖程式 + 一個階梯圖程式



功能方塊圖程式:

Request 16 bits from Slave=1 (M7000 Address=1)
 Using code=2, starting Modbus ADDR_No. is 0
 If CODE=1 or 2, each returned N1 to N12 contains one word (or 16 bits)

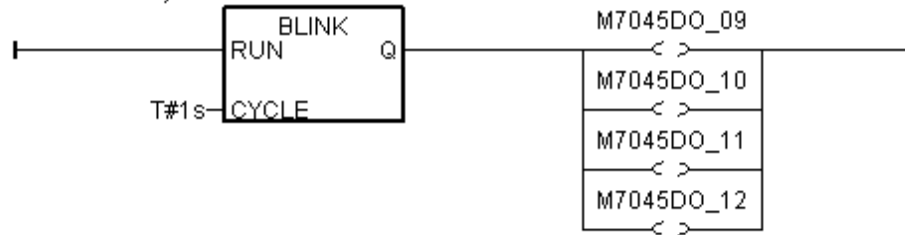
若將SLAVE設為例如3002表示要使用COM3來連接 站號為2號的那個 Modbus Slave device.
 若設為 2001表示要用 COM2去連 站號為1的 device



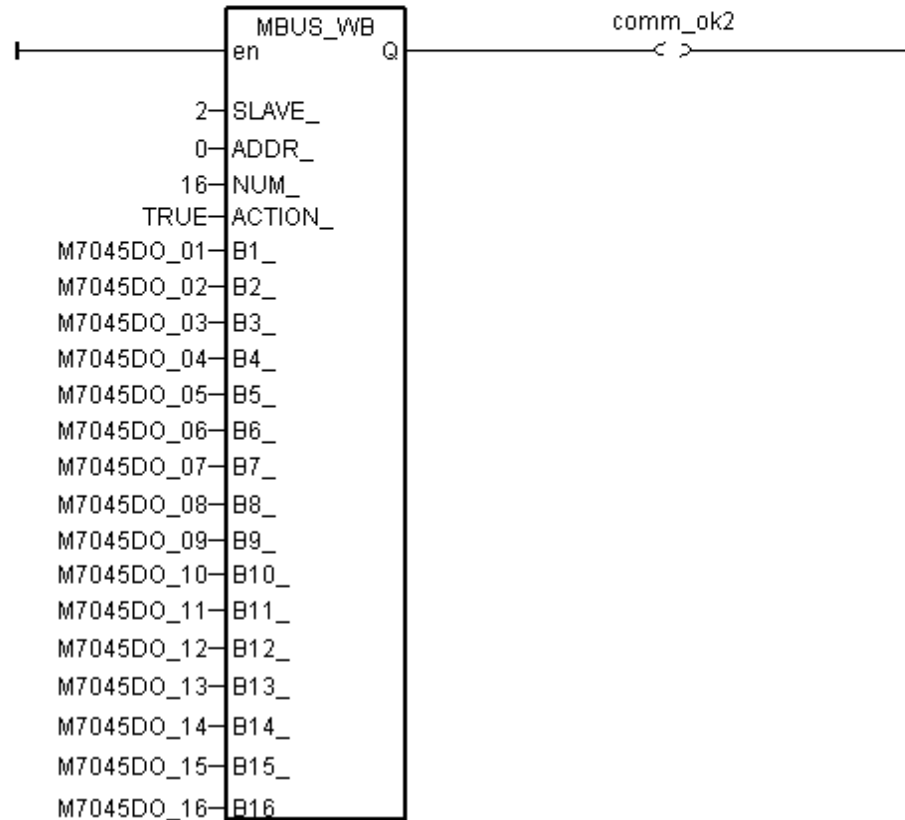
階梯圖程

式:

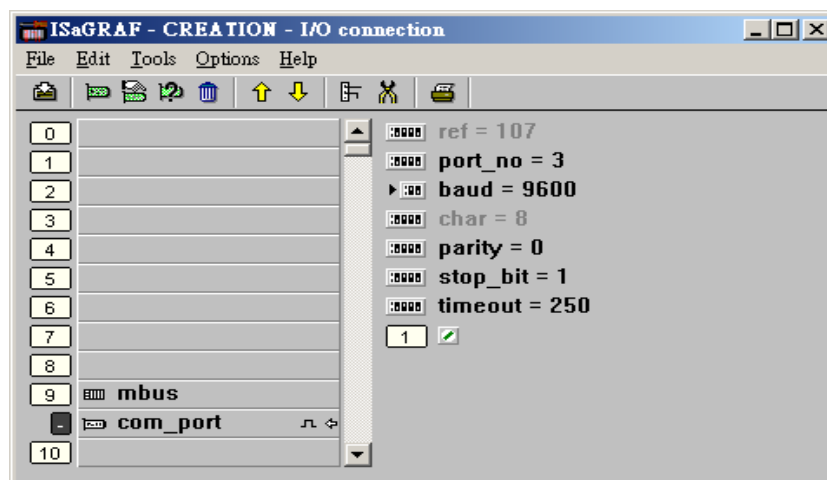
(*Blinking D/O Ch. 9 to 12 *)



(* Write 16 bits to Slave=2 (M-7000 Address=2), starting Modbus ADDR_ No. is 0, this block automatically uses code=15 *)



I/O 連結設定:



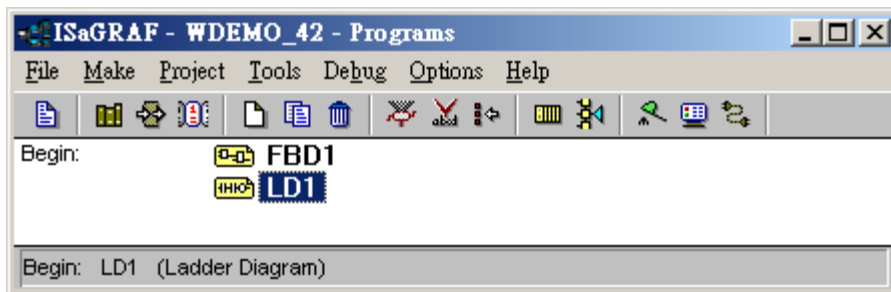
範例 42: 連接 1:M-7053D 來取得 D/I counter 值 (本範例 Project 名稱為 “Wdemo_42”).

記得要先操作 21.1 節的步驟 1 到 5 來對 M-7053D 作初始設定.

變數宣告:

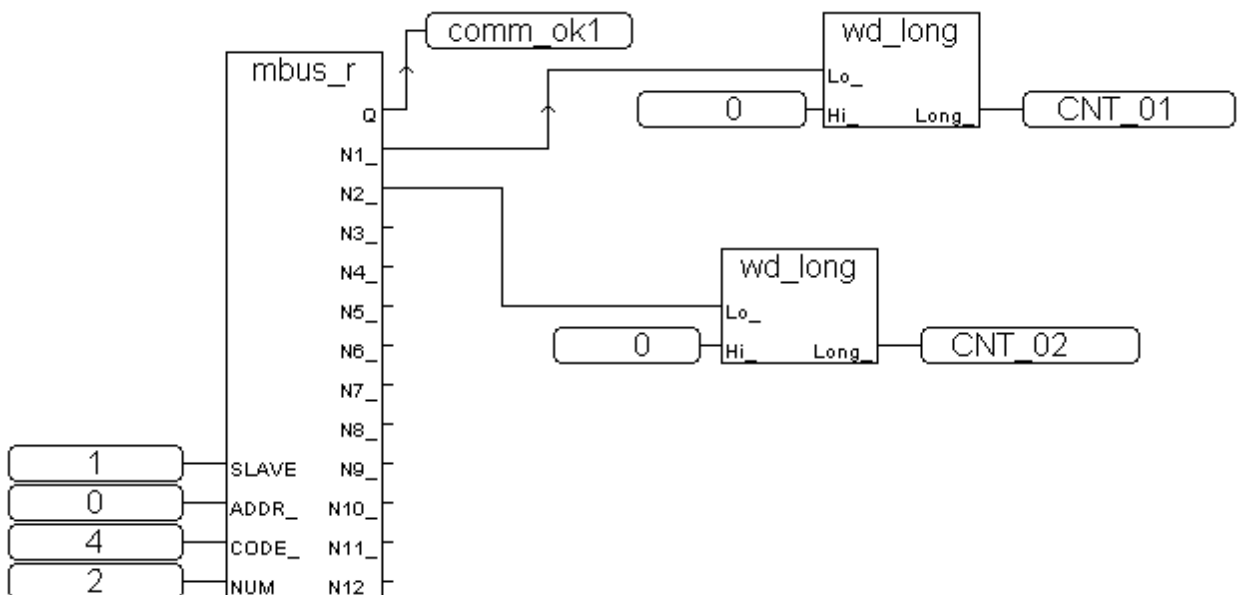
Name	Type	Attribute	Description
comm_ok1	Bool	Internal	M-7053D 的通訊狀態
RS1	Bool	Internal	要重置 Ch1. D/I counter 值為 0, 將此值設為 True
RS2	Bool	Internal	要重置 Ch2. D/I counter 值為 0, 將此值設為 True
CNT_01	Integer	Internal	Ch1 D/I counter 值
CNT_02	Integer	Internal	Ch2 D/I counter 值

Project 架構: 一個功能方塊圖程式 + 一個階梯圖程式



功能方塊圖程式:

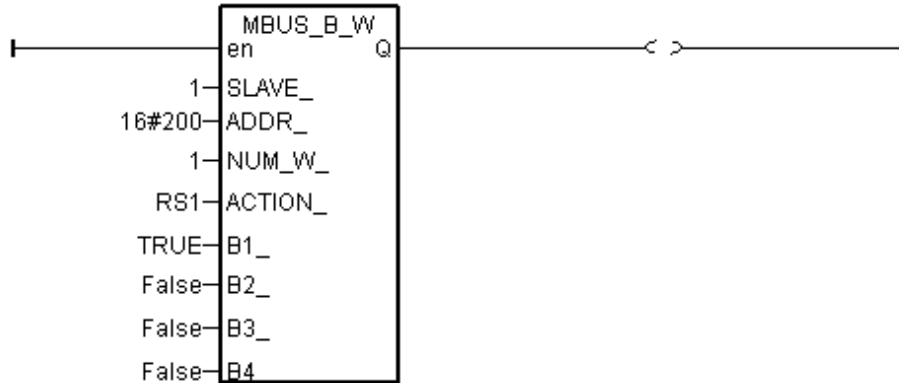
Using Code=4 to request M-7000 D/I counter value, Starting from Modbus ADDR No=0
 NUM can be 1 to 12 depends on how many D/I counter channel in the M-7000 to be read
 The M-7000 D/I counter value is from 0 ~ 65535 contained in one word.
 Since Mbus_r & Mbus_r1 can only return word value as -32768 to +32767, so please
 use "wd_long" to convert this word to become a long integer value. Then the converted
 counter value will be 0 to 65535



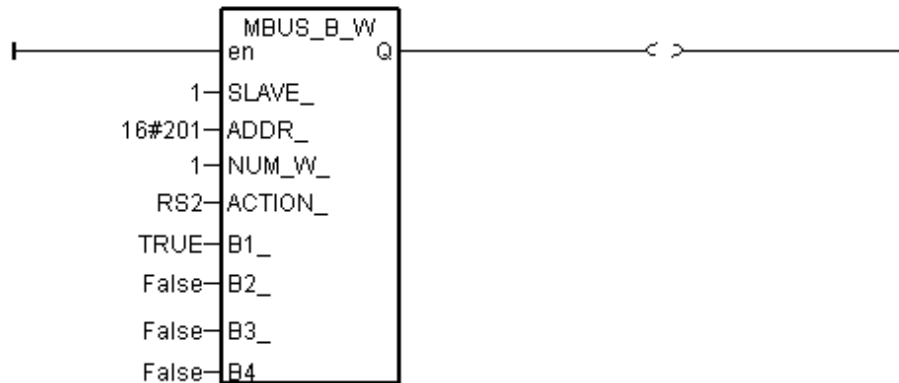
要重置各M-7000的D/I counter 值為0, 必須寫 Bit 值為 1 (TRUE) 到 coil Modbus 編號 16#200 到 16#21F. 例如重置 Ch1 為 0, 要寫到編號 16#200, 而 16#201 則是 Ch2, ..., 16#21F 是給 Ch32 使用.

階梯圖程式:

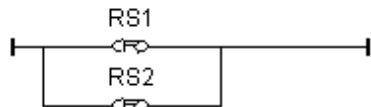
(* Set RS1 to True to clear D/I counter 1 (ADDR 16#200), The "Clear D/I counter" 's Modbus ADDR is from 16#200 to 16#21F depends on the total D/I channel number of the M-7000 *)



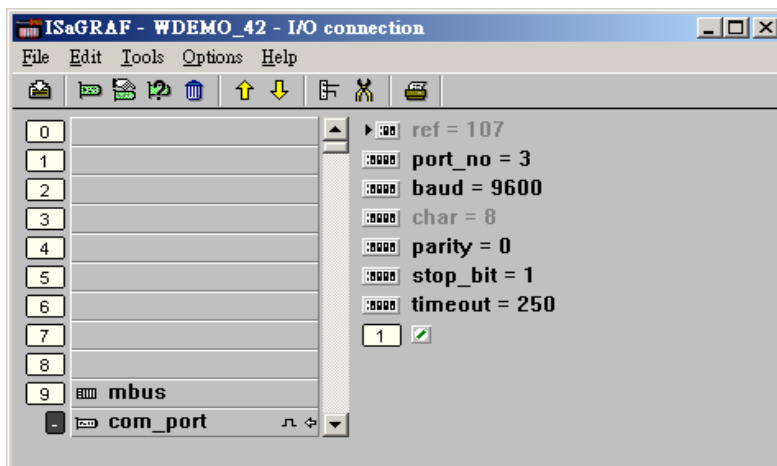
(* Set RS2 to True to clear D/I counter 2 (ADDR 16#201), The "Clear D/I counter" 's Modbus ADDR is from 16#200 to 16#21F depends on the total D/I channel number of the M-7000 *)



(* alsway reset RS1 & RS2 to False at the end *)



I/O 連結設定:



範例 43: 連接 1:M-7017R 與 2:M-7024 (本範例 Project 名稱為 “Wdemo_43”)

本例請在 初始設定時 規劃 M-7017R 的 Input range 與 Type 為 +/- 10V
M-7024 的 Output range 與 Type 為 +/- 10V

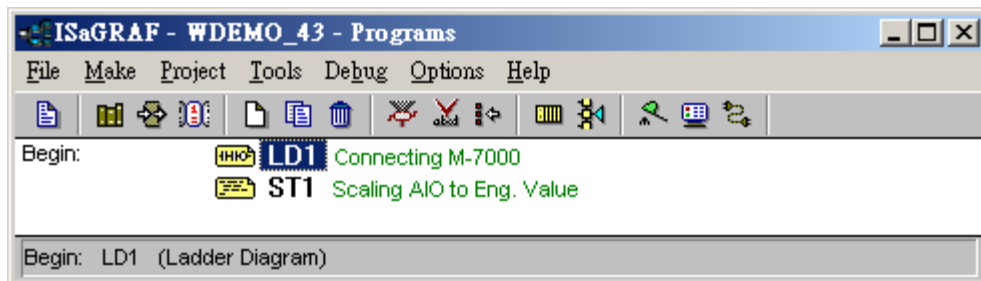
使用者可參考 M-7000 模塊包裝盒內所附的文件, 或訪問 http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm 來取得更多的 軟/硬體說明, 比如像 “Analog Input Type and Data Formate Table” 的訊息 (Type code setting)

本例有使用 變數陣列, 請參考第 2.6 節 的說明.

變數宣告:

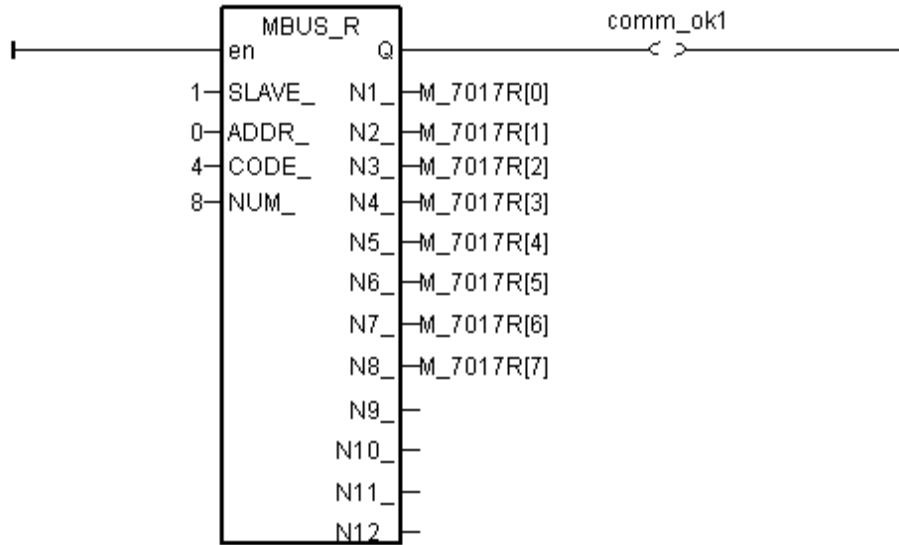
Name	Type	Attribute	Description
comm_ok1	Bool	Internal	M-7017R 的通訊狀態
comm_ok2	Bool	Internal	M-7024 的通訊狀態
M_7017R[0..7]	Integer	Internal	為變數陣列, Dim 欄為設為 8 M-7017R 的 Analog Input 值 (-32768 到 +32767) 表示 (-10 到 +10) V
M_7024[0..3]	Integer	Internal	為變數陣列, Dim 欄為設為 4 M-7024 的 Analog Output 值 (-16384 到 +16383) 表示 (-10 到 +10) V
In_Val[0..7]	Integer	Internal	為變數陣列, Dim 欄為設為 8 M_7017R[0..7] 轉換後的工程值 (-32768 到 +32767) 轉換為 (-10000 到 +10000)
Out_Val[0..3]	Integer	Internal	V 為變數陣列, Dim 欄為設為 4 要輸出的工程值 會轉換成 M_7024[0..3]的輸出值 (-1000 到 +1000) 轉換為 (-16384 到 +16383)
ii	Integer	Internal	給 for 迴圈使用的 index

Project 架構: 一個階梯圖程式 + 一個 ST 程式

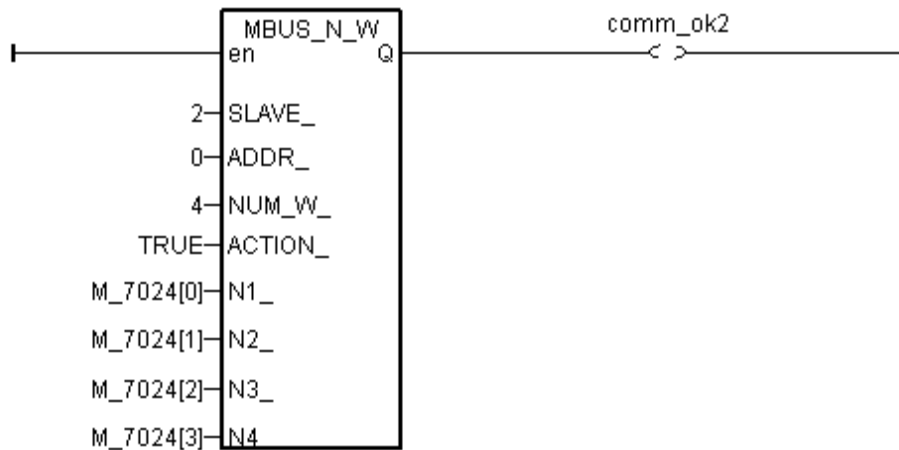


階梯圖程式:

(* Read 8 words from Slave=1 (M-7000 Address=1) using code=4. starting Modbus ADDR No. is 0
Please set M-7017R's range to +/-10V by DCON utility (type code=8) *)



(* Write 4 words to Slave=2 (M-7000 Address=2) , starting Modbus ADDR No. is 0
Please set M-70124's range to +/-10V by DCON utility (type code=33) *)



ST 程式:

```
(* Please configure this M-7017R as +/- 10V range (type code=8) *)
(* convert M-7017R's A/I value (-32768 to +32767) to become engineering value
   of (-10000 to +10000) *)

for ii := 0 to 7 do
  IN_Val[ii] := Bin2Eng( M_7017R[ii] , 32767 , -32768 , 10000 ,
-10000 );
end_for ;

(* Please configure this M-7024 as +/- 10V range (type code=33) *)
(* convert OUT_Val of (-1000 to +1000) to become M-7024's A/O value
   of (-16384 to +16383) *)

for ii := 0 to 3 do

  if OUT_Val[ii] > 1000 then
    M_7024[ii] := 16383 ;

  elsif OUT_Val[ii] < -1000 then
    M_7024[ii] := -16384 ;

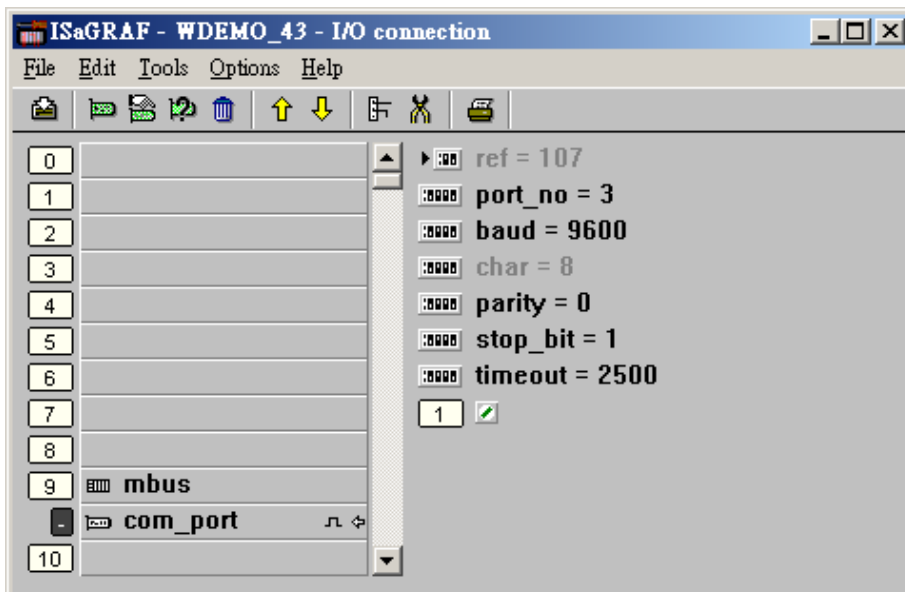
  elsif OUT_Val[ii] >= 0 then
    M_7024[ii] := (OUT_Val[ii] * 16383) / 1000 ;

  elsif OUT_Val[ii] < 0 then
    M_7024[ii] := (OUT_Val[ii] * -16384) / 1000 ;

  end_if ;

end_for ;
```

I/O 連結設定:



範例 44: 連接 1:M-7017RC (本範例 Project 名稱為 “Wdemo_44”)

請在初始設定時規劃 M-7017RC 的 Input range 與 Type 為 +/- 20 mA

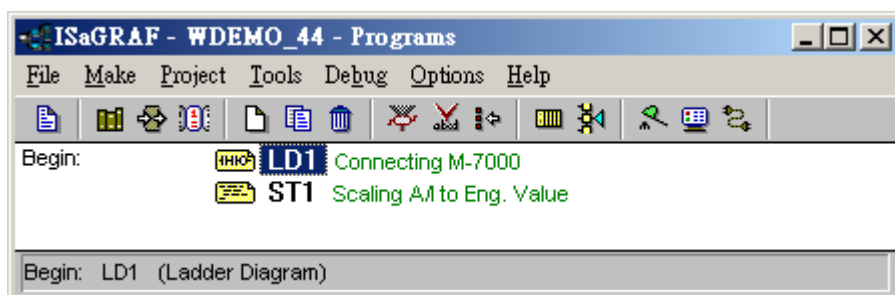
使用者可參考 M-7000 模塊包裝盒內所附的文件, 或訪問 http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm 來取得更多的 軟/硬體說明, 比如像 “Analog Input Type and Data Formate Table” 的訊息 (Type code setting)

本例有使用 變數陣列, 請參考第 2.6 節 的說明.

變數宣告:

Name	Type	Attribute	Description
comm_ok1	Bool	Internal	M-7017RC 的通訊狀態
M7017RC[0..7]	Integer	Internal	為變數陣列, Dim 欄為設為 8 M-7017RC 的 Analog Input 值 假如有設好 Input range Type 為 +/- 20 mA , 則 (-32768 到 +32767) 表示 (-20 到 +20) mA
In_Val[0..7]	REAL	Internal	為變數陣列, Dim 欄為設為 8 (實數 “REAL” 格式) M7017RC[0..7] 轉換後的工程值 (4 到 20) mA 轉換成 (0.0 到 1000.0) psi
VAL[0..7]	Integer	Internal	為變數陣列, Dim 欄為設為 8 (整數 “Integer” 格式) M7017RC[0..7] 轉換後的工程值 (4 到 20) mA 轉換成 (0 到 10000), 單位 0.1 psi
ii	Integer	Internal	給 for 迴圈使用的 index

Project 架構: 一個階梯圖程式 + 一個 ST 程式



M-7017RC 的 Analog Input 表:

+/- 20 mA type (type code=16#D)	-32768 到 +32767
4 to 20 mA type (type code=16#7)	0 到 +32767

如果使用的 Sensor 為 4 到 20 mA 的量測型態，可以考慮將 M-7017RC 設為 +/- 20 mA type .
 (此時設 M-7017RC 為 "4 to 20 mA" type 比較不好)

原因為:

如果 M-7017RC 的 type 為 4 到 20 mA type, Analog Input 值若為 0 或接近 0 可以表示 Sensor 輸入為 4 mA , 但也有可能是 Sensor 斷線. 因此光看 Analog Input 數值, 區分不出這 2 者.

然而若是將 M-7017RC 規劃為 +/- 20 mA type, 當值接近 0 時, 只有可能是 Sensor 斷線 而 Sensor 輸入的 4 到 20mA ,值會是 (6553 到 32767), 4mA 時值是在 6553 附近, 不是在 0 附近.

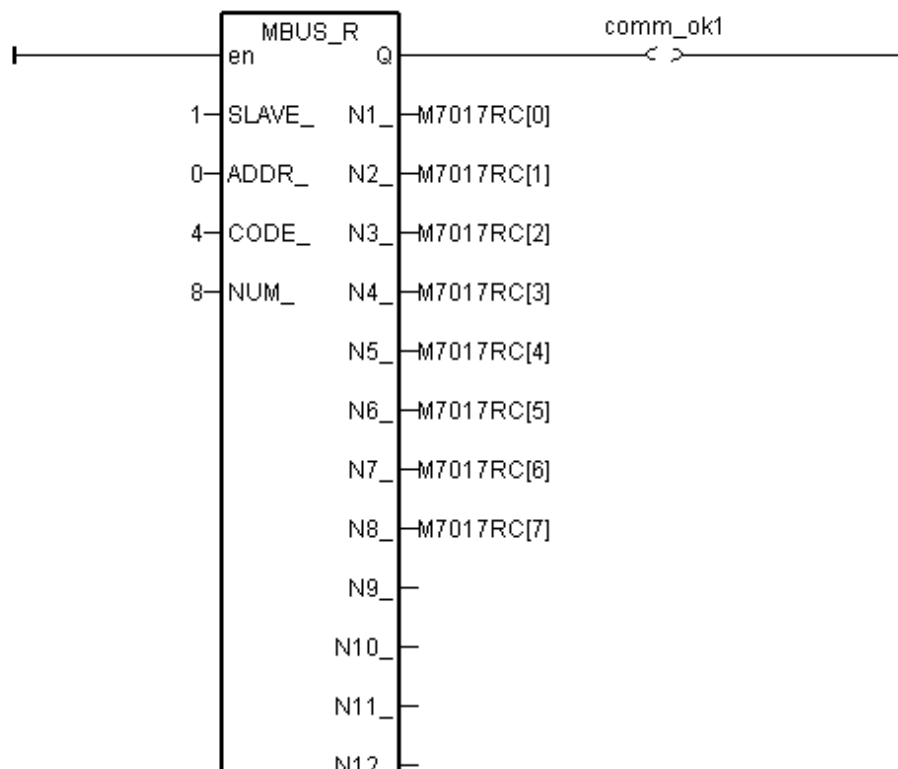
(當然 M-7017RC 的 RS-485 連線需正常, 本例的 comm_ok1 變數可用來判斷 M-7000 的 RS-485 連線是否正常, 值若為 False 表通訊有問題, 你可在程式內作適當處理).

因此有需要分辨出 Sensor (4 到 mA) 是否正常時, 最好是規劃 M-7017RC 為 +/- 20 mA type.

如此你的程式內就可比如說當 M7017RC[0..7] 輸入值 小於 5000 或 小於 4000 , 就可判定為 Sensor 斷線 或 Sensor 異常.

階梯圖程式:

(* Read 8 words from Slave=1 (M-7000 Address=1) using code=4. starting Modbus ADDR No. is 0
 Please set M-7017RC 's range to +/- 20 mA by DCON utility (type code=D) *)



ST 程式:

```
(* Please configure this M-7017RC as +/- 20 mA range (type code=D) *)
(* We will convert (4 , 20 mA) to become (0.0 , 1000.0 Psi), Real format *)
for ii := 0 to 7 do
  IN_Val[ii] := A4_20_To( M7017RC[ii] , 16#D , 1000.0 , 0.0 );
end_for ;

(* or you may use Bin2Eng( ) to convert (4 to 20mA) to become (0 to 10000) as below, unit is
0.1 psi*)
(* Please declare Val[0..7] as Integer format *)
for ii := 0 to 7 do
  Val[ii] := Bin2Eng( M7017RC[ii] , 32767 , 6553 , 10000 , 0 );
end_for ;

(* You can do something if the sensor is broken or communication is break *)
if comm_ok1 and ( M7017RC[ii] < 5000 ) then

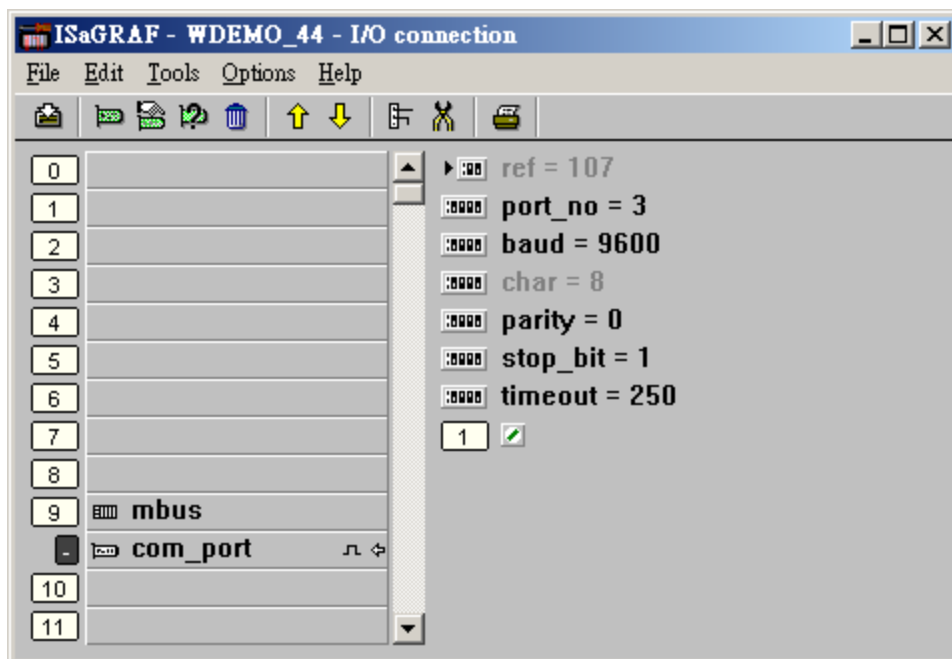
  (* You may do something if 4-20 mA sensor is broken *)

elseif comm_ok1=False then

  (* You may do something if communication between controller & M-7017RC is break *)

end_if ;
```

I/O 連結設定 connection:



範例 45: 連接 1:M-7019R 來取得 攝氏度 (本範例 Project 名稱為 “Wdemo_45”)

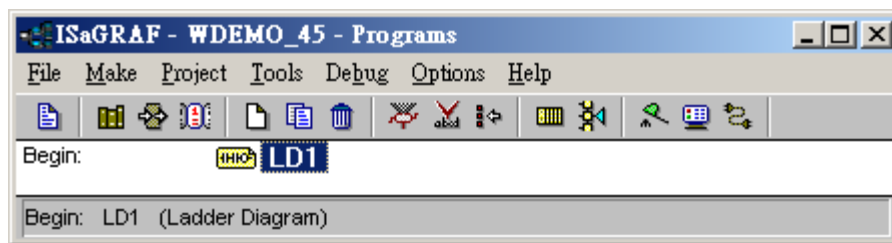
請在初始設定時規劃 M-7019R 的 range type 為 Thermocouple, K-Type (Type code=0F)

使用者可參考 M-7000 模塊包裝盒內所附的文件, 或訪問 http://www.icpdas.com/products/Remote_IO/m-7000/m-7000_list.htm 來取得更多的 軟/硬體說明, 比如像 “Analog Input Type and Data Formate Table” 的訊息 (Type code setting)

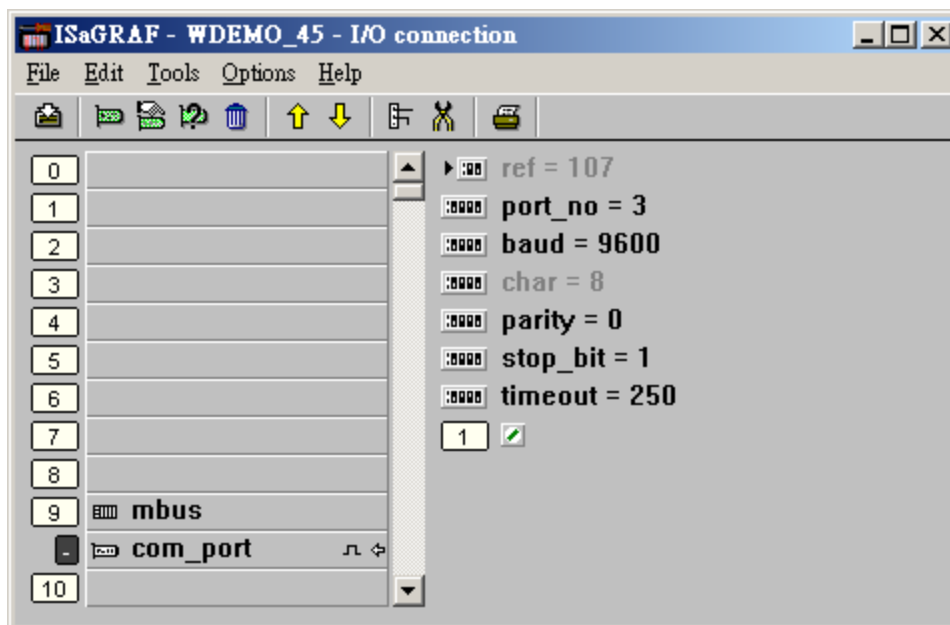
變數宣告:

Name	Type	Attribute	Description
comm_ok1	Bool	Internal	M-7019 的通訊狀態
Temper_1 到 Temper_8	Integer	Internal	M-7019R 的溫度值

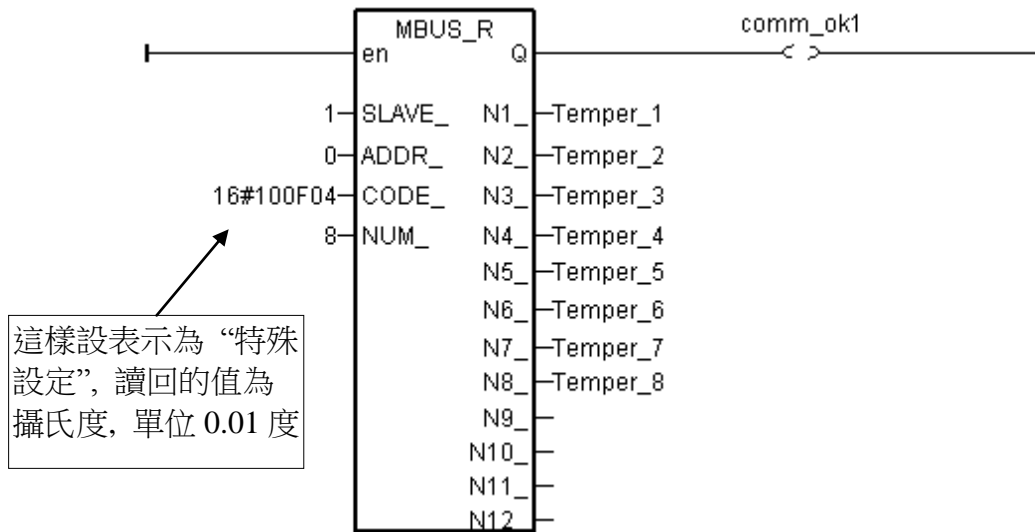
Project 架構: 一個階梯圖程式



I/O 連結設定:



階梯圖程式:



“Mbus_r” 與 “MBUS_R1” 的 “CODE_” 參數可以是 “標準設定” 或 “特殊設定”。

標準設定:

若 “CODE_” 為 2，每個回傳的 “N1_” 到 “N2_” 各包含 16-bit (或 16 Digital Input)

若 “CODE_” 為 4，每個回傳的 “N1_” 到 “N12_” 為 1 個 Word 值 (-32768 到 +32767)

特殊設定:

只能用於 M-7000 系列的 溫度量測模塊，如 M-7015, M-7018R 與 M-7019R

“CODE_” 格式: TTRRCC (16 進位)

TT=10 (轉換為 “攝氏度”)

TT=20 (轉換為 “華氏度”)

TT=00 (這樣設變成 “標準設定”. RR 需同時也設成 00)

RR: 模塊的 “Type Code” 設定

CC: Modbus function code 編號, 可以是 1 到 4

使用特殊設定得到的溫度單位是 0.01 度. 例如 “3012” 表示 30.12 度. 若值為 999990 表示 “溫度 Sensor 斷線” (只適用有溫度斷線偵測能力的模塊, 像 M-7019R, M-7015, M-7018R)

例如, 設 I-7019R 的 “CODE_” 為:

A. 16#100F04 : (TT=10, RR=0F, CC=04, 16 進位表示), 則使用 “攝氏度”, 單位為 0.01 度, range type 為 “0F: Thermocouple K Type, -270 ~1372 degree Celsius”, code=04(Dec.). 所以如果回傳 “2356” 表示 23.56 度, “-489” 表示 -4.89 度, “999990” 表示 “Sensor 斷線”。

B. 16#200F04 : (TT=20, RR=0F, CC=04, 16 進位表示), 則使用 “華氏度”, 單位為 0.01 度, range type 為 “0F: Thermocouple K Type, -270 ~1372 degree Celsius”, code=04(Dec.). 所以如果回傳 “4512” 表示 45.12 度, “500” 表示 5.00 度, “999990” 表示 “Sensor 斷線”。

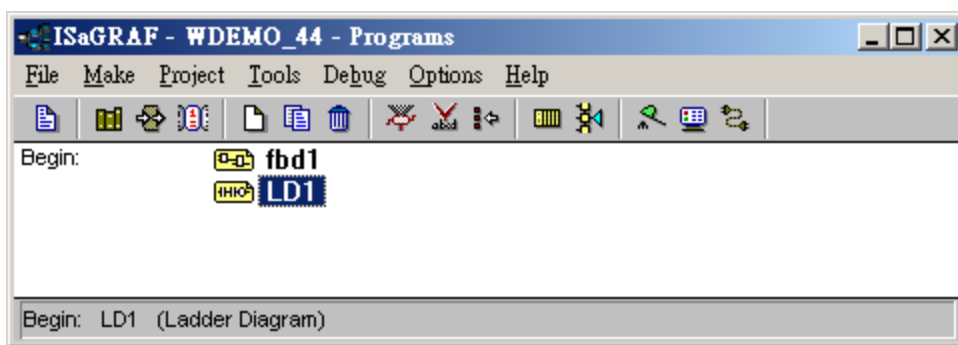
C. 16#04 : (TT=00, RR=00, CC=04, 16 進位表示), 為標準設定。

範例 46: 連接 1: M-7080-D 來取得 counter 值 (本範例 Project 名稱為“Wdemo_46”)

變數宣告:

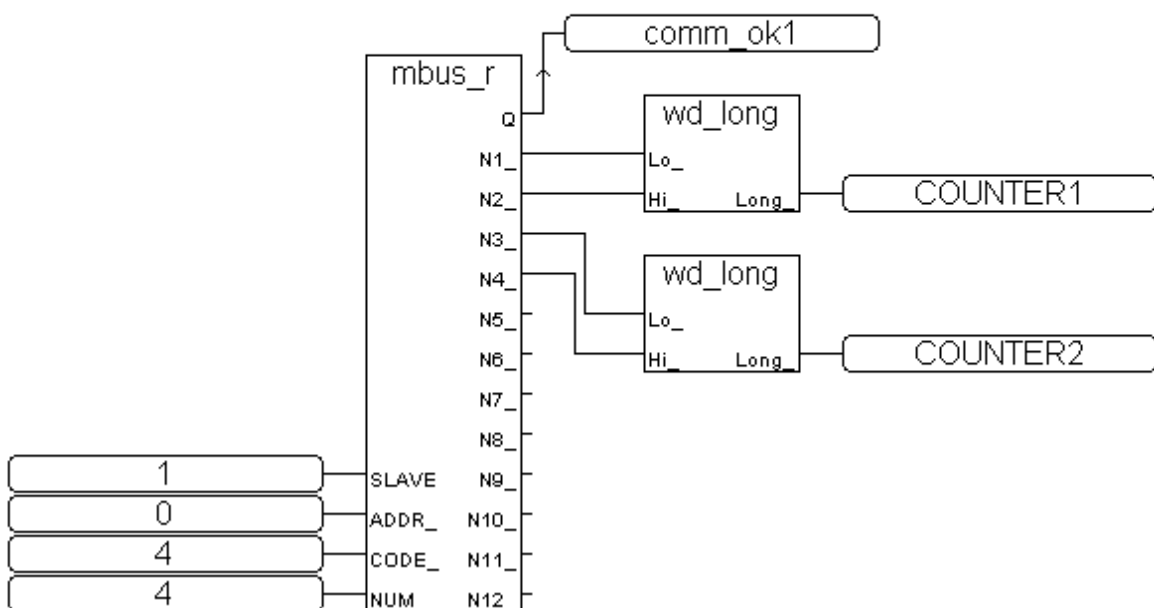
Name	Type	Attribute	Description
comm_ok1	Bool	Internal	M-7080D 的通訊狀態
RS1	Bool	Internal	要重置 Ch1. counter 值為 0, 將此值設為 True
RS2	Bool	Internal	要重置 Ch2. counter 值為 0, 將此值設為 True
COUNTER1	Integer	Internal	M-7080D 的 CH1 Counter 值
COUNTER2	Integer	Internal	M-7080D 的 CH2 Counter 值

Project 架構: 一個功能方塊圖程式 + 一個階梯圖程式



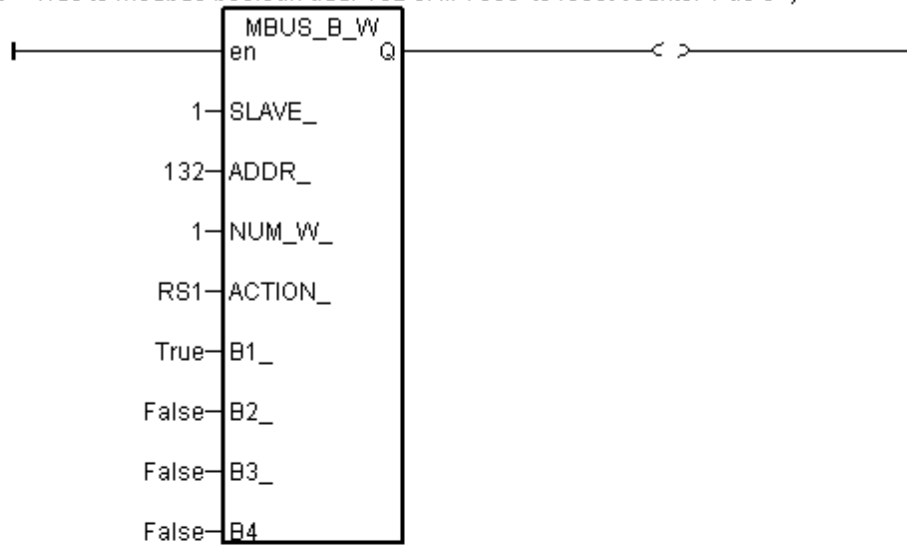
功能方塊圖程式:

Request 4 words using Modbus code=4 from "Slave=1" (M-7000's Address=1)
 The starting Modbus ADDR_No. is 0
 Then convert 2 words to become one long integer

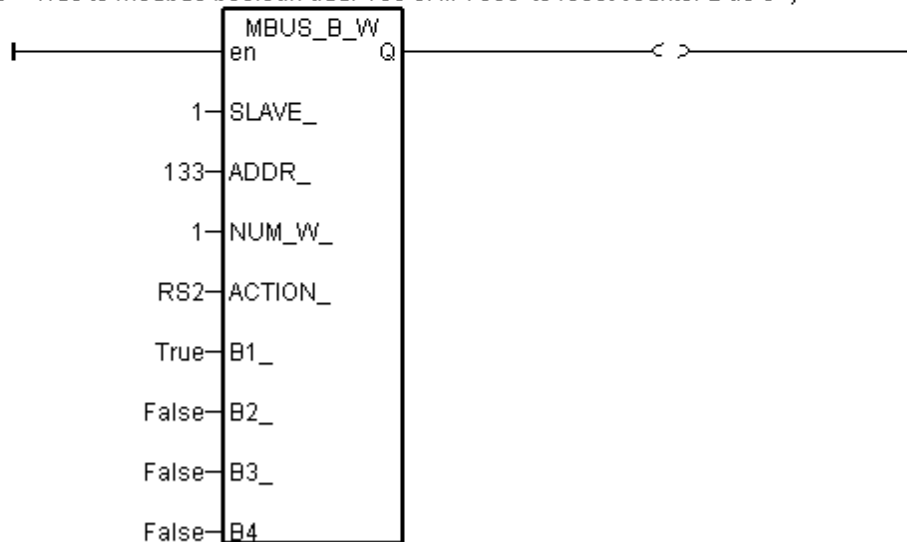


階梯圖程式:

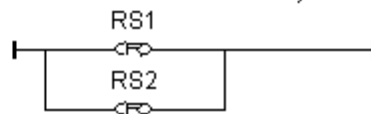
(* Write value = True to modbus boolean addr 132 of M-7080 to reset counter 1 as 0 *)



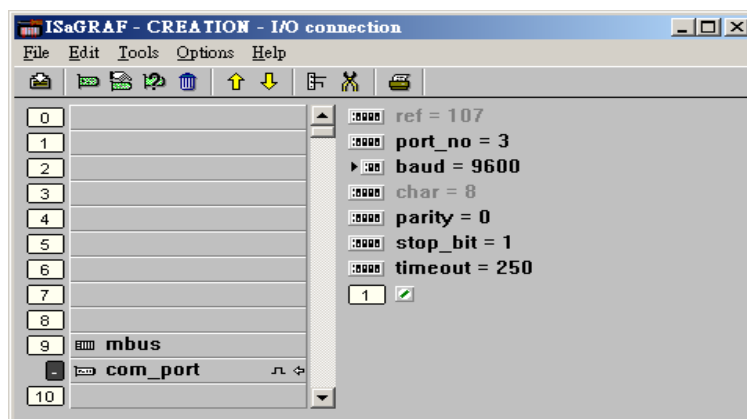
(* Write value = True to modbus boolean addr 133 of M-7080 to reset counter 2 as 0 *)



(* reset RS1 & RS2 to False at the end *)



I/O 連結設定:



第 22 章連接 Modbus TCP/IP I/O

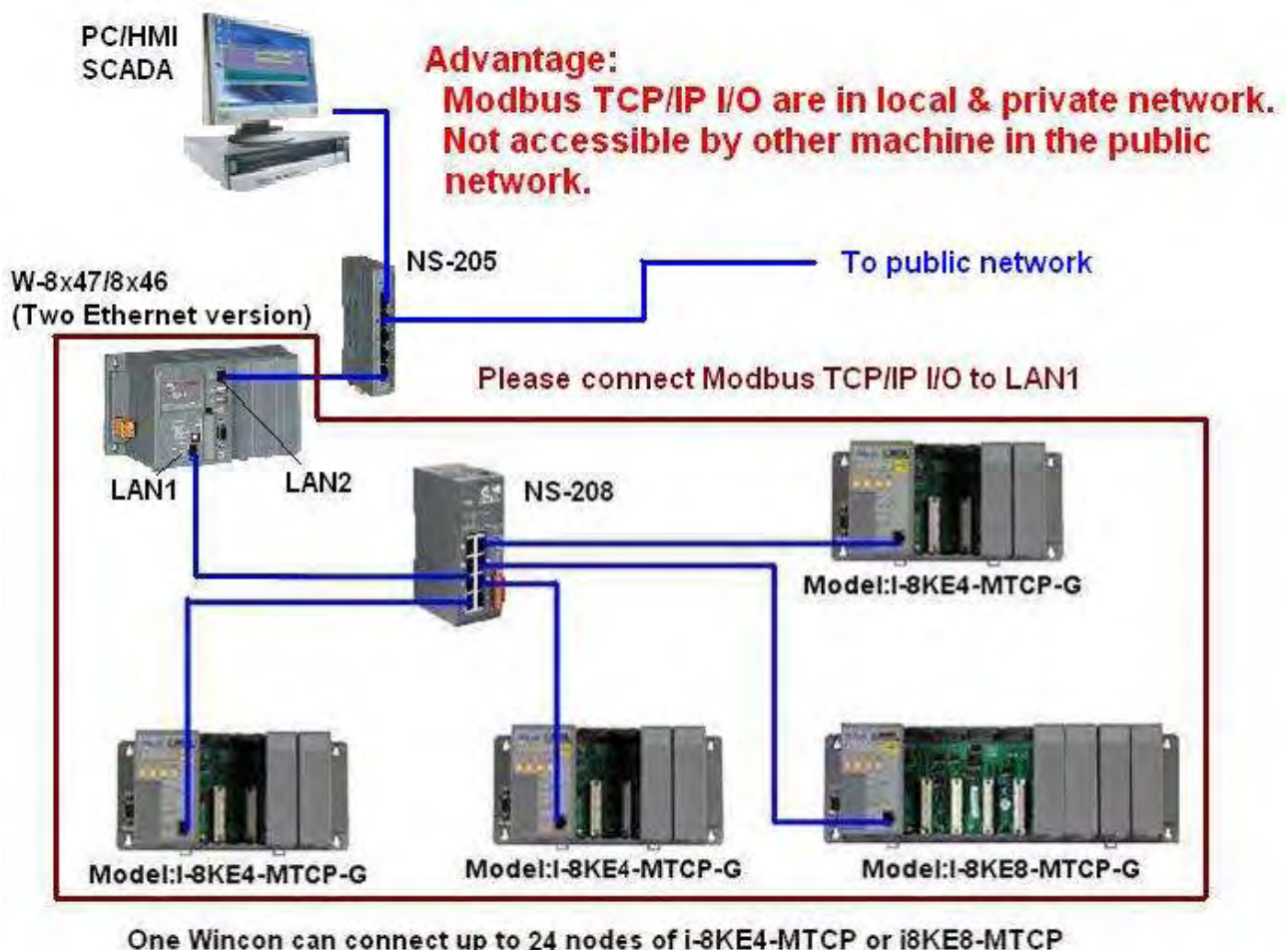
W-8x47 / 8x46 從它的驅動 3.32B 起支持 I-8KE4-MTCP 與 I-8KE8-MTCP ethernet I/O. 請參考 I-8KE4/8-MTCP: http://www.icpdas.com/products/PAC/i-8000/i-8KE4_8KE8_MTCP.htm
Wincon ISaGRAF driver: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>
NS-205 / NS-208 : http://www.icpdas.com/products/Switch/industrial/ethernet_switch.htm
(WP-8xx7, VP-25W7 / 23W7 , XP-8xx7-CE6 也有支持)

22.1: I-8KE8-MTCP I/O 簡介

一台 W-8x47 / 8x46 最多可連接 24 個 I-8KE4-MTCP 或 I-8KE8-MTCP. 全部最多可有約 3000 到 6000 點的 I/O, 全部的 I/O Scan Time 約 30 到 40 ms. 如果只連接少於 10 個 I-8KE4/8-MTCP, I/O Scan Time 約可到 20 ms 左右. 然而此值會受 ISaGRAF 的 PLC scan time 影響, 比如若 PLC scan time 是 10ms, 那連 10 個 I-8KE4/8-MTCP 的 I/O Scan Time 約可到 20 ms 左右. 但若 PLC scan time 是 50ms, 那 I/O Scan Time 將變成約 50 ms 左右 (若是採用 W-8x36 / 8x37, 那它的 Ethernet IO 的 Scan Time 約是 W-8x47 / 8x46 (雙網口版本) 的 2 倍大, 用 W-8x36 / 8x37 連 Ethernet I/O 的效率比較慢)

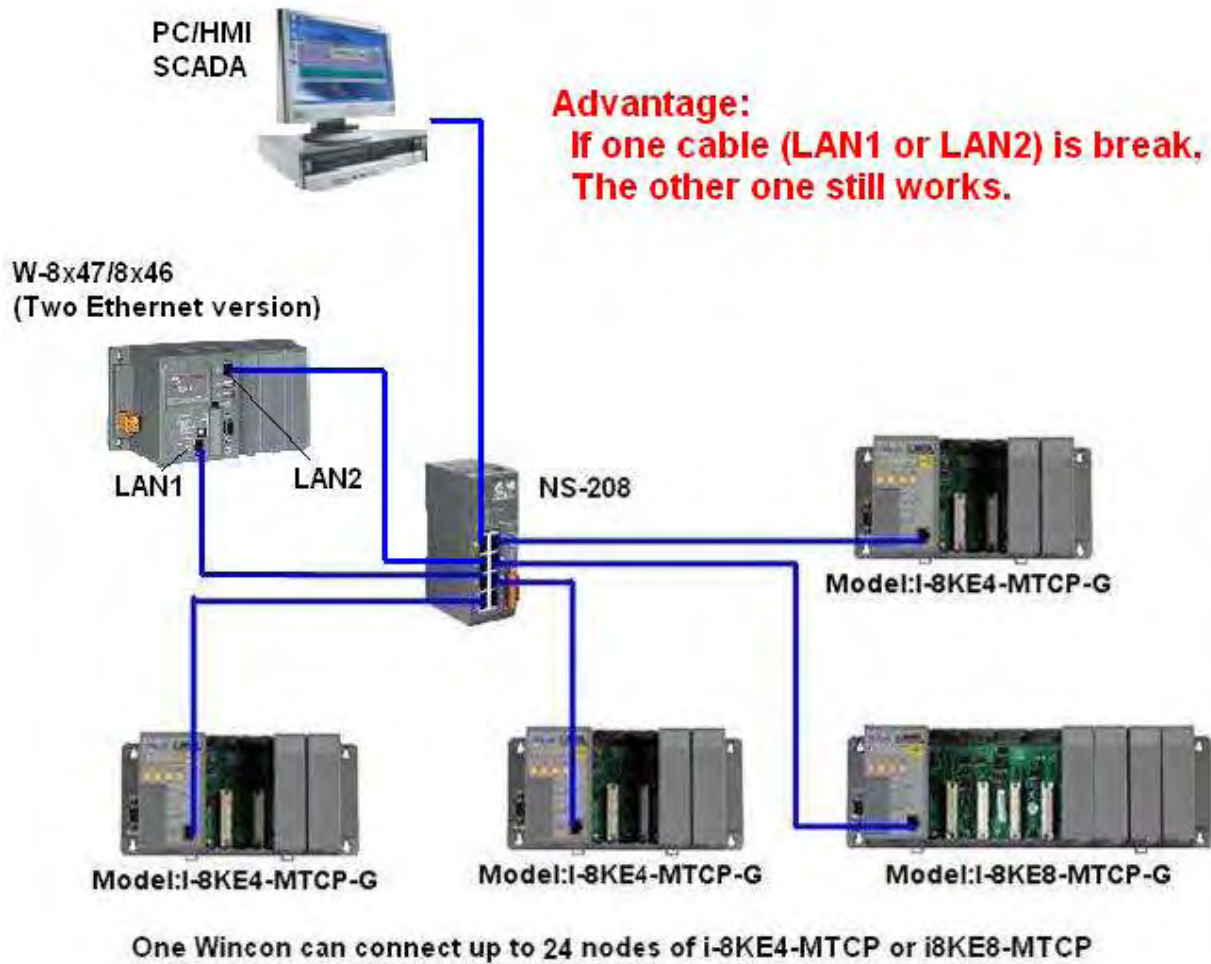
規劃 1: W-8x47 / 8x46 (雙網口 Wincon) 在 安全私人的區域網路內 連 Modbus TCP/IP I/O .

(優點 : Ethernet I/O 只連到 W-8x47 / 8x46 的一個網口, 沒有公開到其它網路, 安全性佳)



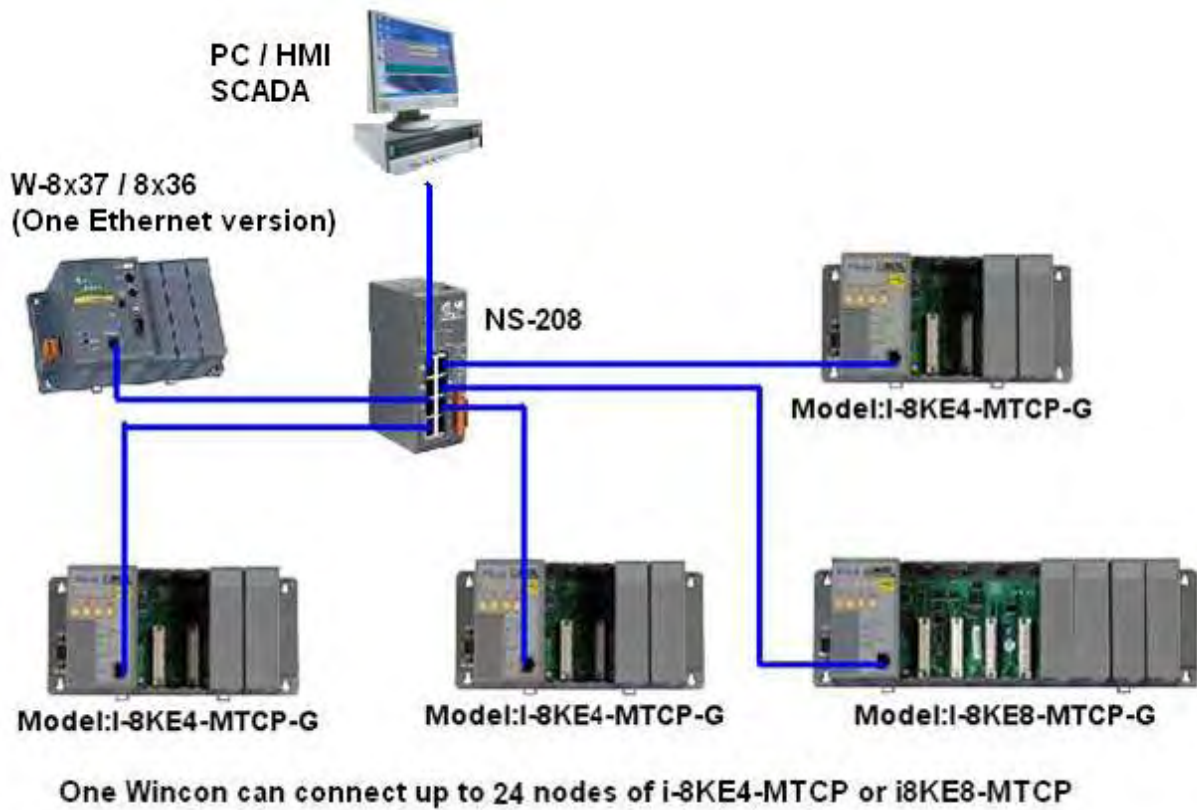
規劃 2: W-8x47 / 8x46 (雙網口 Wincon) 的 2 個網口都連接到 Modbus TCP/IP I/O.

(優點: 使用 Wincon 的 2 個網口同時連接 Ethernet I/O, 當一個網口的連線出問題, 另一個網口仍可繼續運作)



規劃 3: W-8x37 / 8x36 (一個網口版本) 連 Modbus TCP/IP I/O.

此方式沒有 1 與 2 的優點, 且 Ethernet I/O Scan 會比 W-8x47 / 8x46 慢)



22.2: 編寫程式操控 I-8KE8-MTCP I/O

步驟 1:

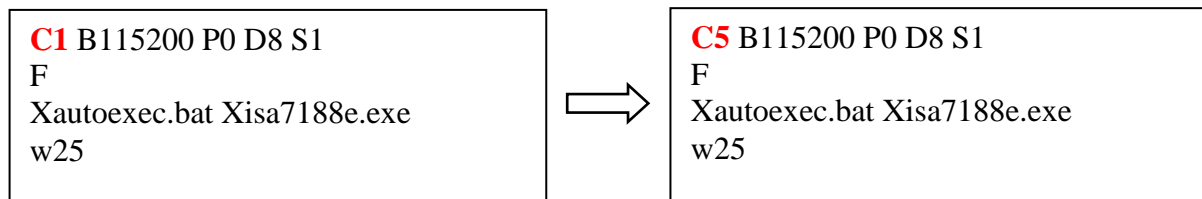
使用 I-8KE4-MTCP 或 I-8KE8-MTCP 前, 要先設好它的 IP 地址.

先關閉 I-8KE4 / 8-MTCP 的電源, 將它的 “INIT” 端子 與 “INIT * COM” 端子 短接, 再開啟電源. PC 用

之後在 PC 上執行 7188xw.exe 程式 (7188xw.exe 可在 I-8000 CD-ROM 或 Ftp 上找到 <ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/>).

7188xw.exe 運行起來後, 請使用一條 RS-232 線連接 PC 的 COM1 到 I-8KE4-MTCP 或 I-8KE8-MTCP 的 COM1.

若您的 PC 沒有 COM1 或 您想使用其他 COM 埠 (如 COM5) 來連接, 可以 先關閉 “7188xw.exe”, 變更同路徑下的 “7188xw.ini” 檔案第一行的 “C 編號”. 例如: 使用 PC 的 COM5 連接到 I-8KE8-MTCP, 則更改 ”7188xw.ini” 內 C1 為 C5, 之後再執行 “7188xw.exe”, 如下



請在 7188xw.exe 視窗上按幾次<Enter>, 如果沒有回應表示 RS-232 連線有問題, 請檢查 PC 到 I-8KE8-MTCP 的接線 與 PC 的 COM 設定對不對.

若有回應, 請輸入 “ip” 來查看目前 I-8KE8-MTCP 的 IP 地址.

要變更 IP, 請輸入, 例如, “ip 192.168.2.70” 來改設 IP 地址.

要查 Mask 設定, 請輸入 “mask”, 要更改請輸入 “mask 255.255.255.0”. (建議 mask 一律設為 255.255.255.0)

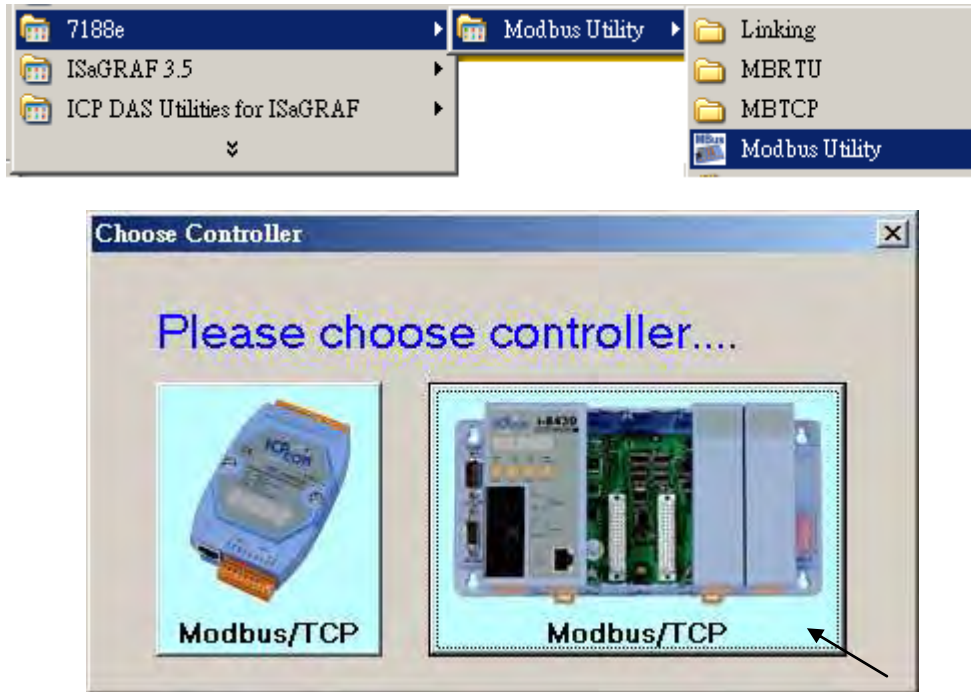
IP 與 Mask 設好後, 請關閉 I-8KE8-MTCP 的電源, 將 “INIT” 與 “INIT * COM” 的接線斷開. 重新開機後, 將 I-8KE8-MTCP 的網口連線到 Wincon (請搭配使用 NS-205 / NS-208)

步驟 2.

PC 上執行 “Modbus utility”來規劃 I-8KE8-MTCP 上的 I/O. 此”Modbus utility” 可在以下地方找到

I-8000 CD-ROM: \napdos\modbus\modbus_utility\ 或

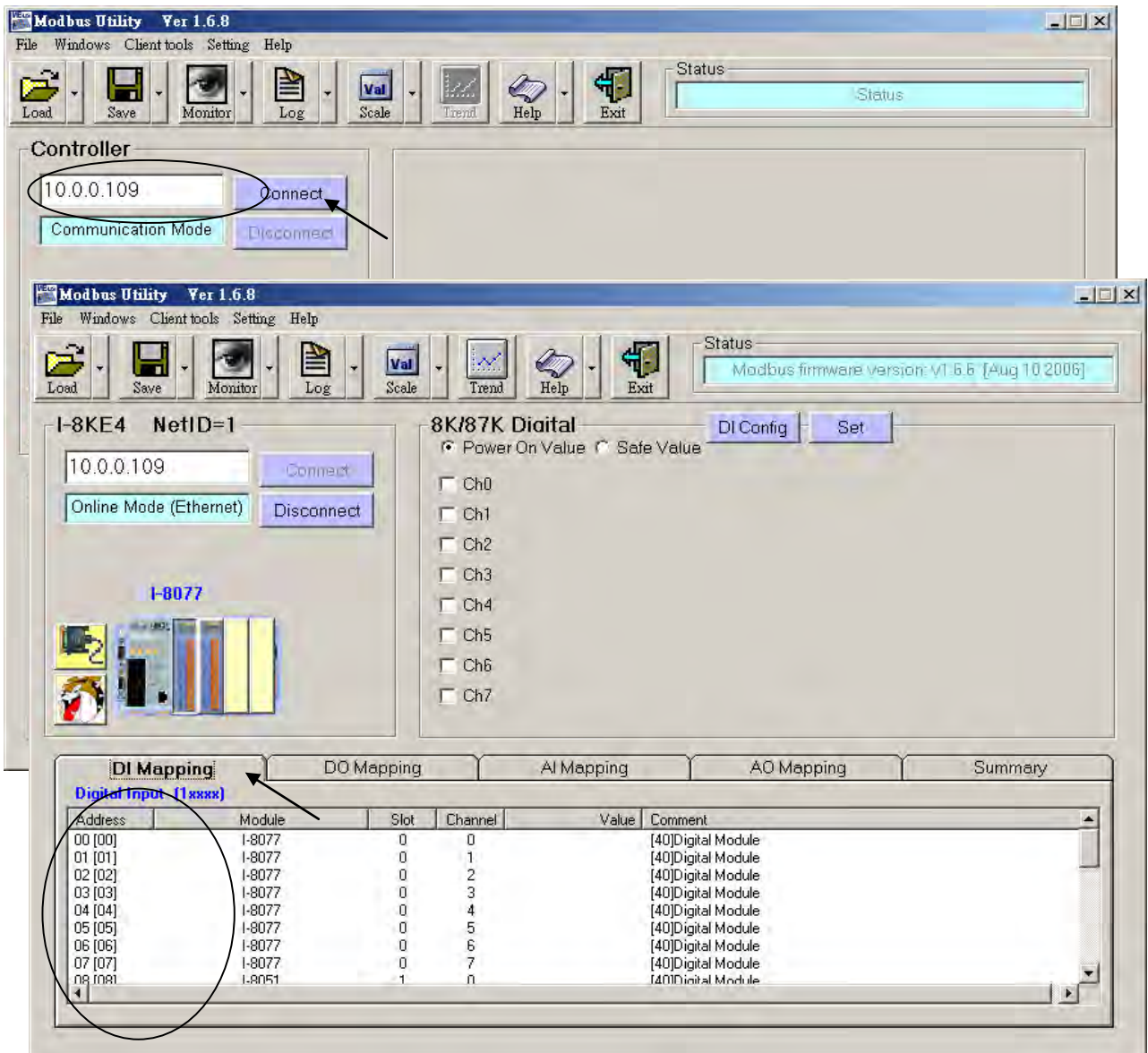
http://www.icpdas.com/products/PAC/i-8000/modbus_web_download.htm



重要事項:

每個 I-8KE4-MTCP 或 I-8KE8-MTCP, 當有變更插在其上的 I/O 板卡時, 都必須使用 Modbus utility 來重新對它的 I/O 規劃一次. 假如有發現到其上的第 2 個與第 3 個 Led 燈一直閃爍的情況, 表示 I/O 卡有被動過, 需重新用 Modbus Utility 再規劃一次”

請輸入該 I-8KE8-MTCP 的 IP, 之後按下 “Connect”. 如果 I-8KE8-MTCP 有被連上, 您可以發現到對應的 Modbus address 設定. 比如, D/I : 由編號 0 到 ..., A/I : 由編號 0 到...

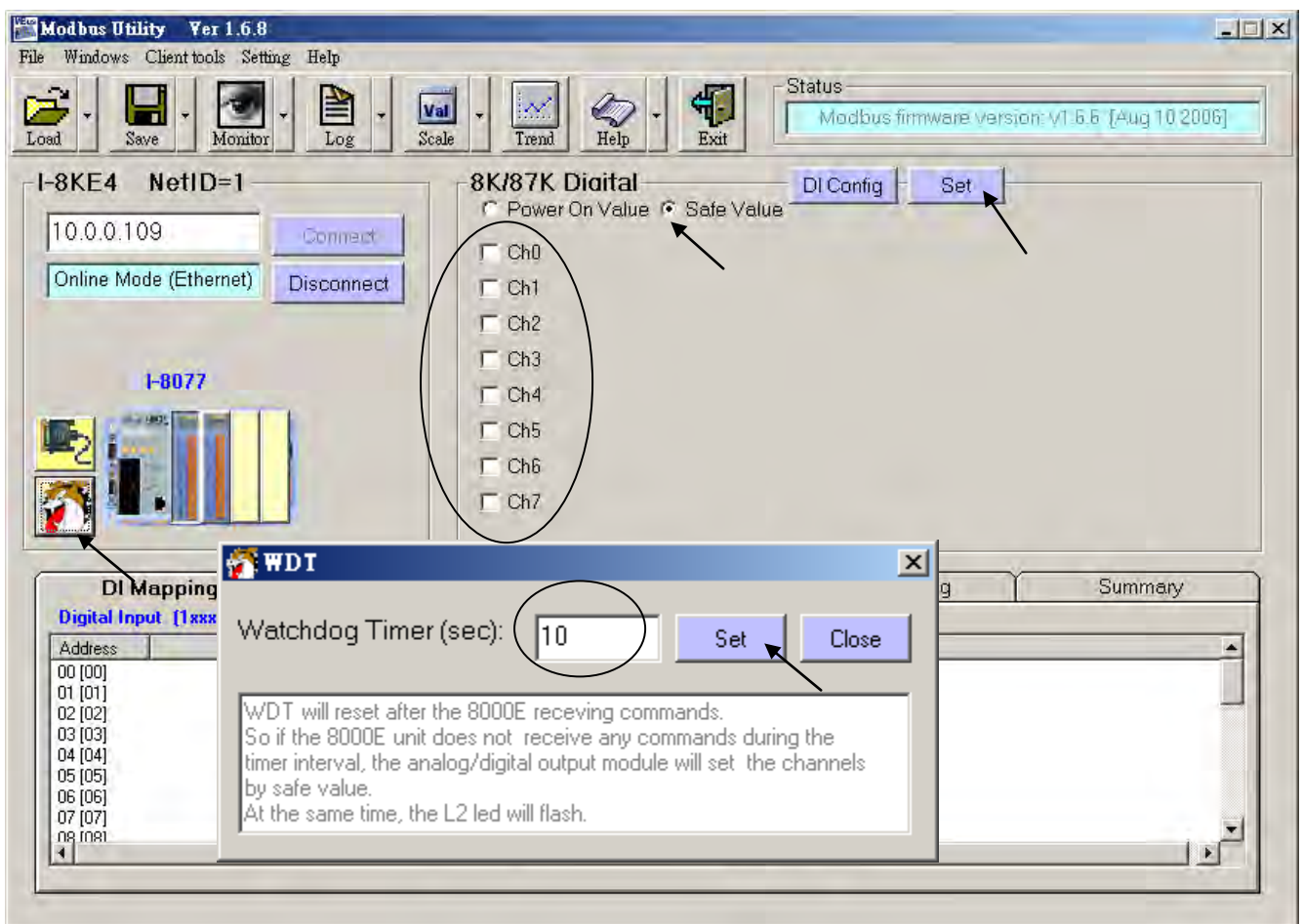


I-8KE4/8-MTCP 內有個 Watchdog 設定。出廠設定為 “disable the watchdog”。您若想啟用它，請設定一個 10 到 120 間的數值（單位為秒）給 “Watchdog timer”。如此當 Wincon 與 I-8KE4/8-MTCP 通訊斷線超過 “Watchdog timer” 時間時，會自動將 I-8KE4/8-MTCP 上的 D/O 或 A/O 輸出點 切換為原先規劃好的 “Safe Value”。

如果您選取 “Safe Value” 之後，並勾選 D/O 板卡的 Ch0 到...，按下 “Set” 的話，表示將這些 D/O 接點的 ”Safe value” 規劃為 ON

如果您選取 “Safe Value” 之後，並沒有勾選 D/O 板卡的 Ch0 到...，按下 “Set” 的話，表示將這些 D/O 接點的 ”Safe value” 規劃為 OFF

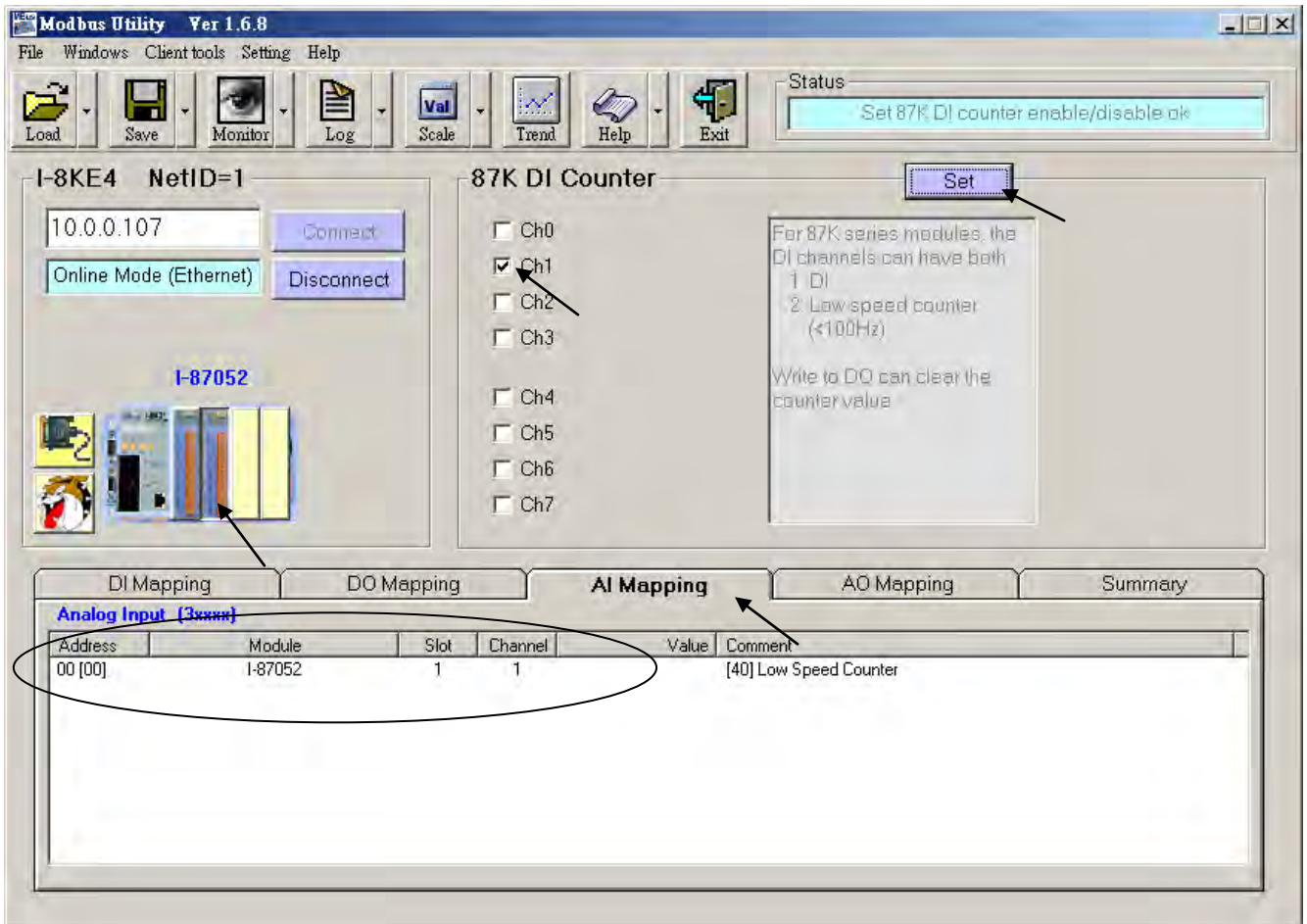
各 D/O 與 A/O 接點 的 “Safe value” 功能，只有在 “Watchdog timer” 設定值不是 0 的情況下才會啟用，將 “Watchdog timer” 值設為 0, 表示關閉 I-8KE4/8-MTCP 的 WatchDog 功能。



若 I-8KE8-MTCP 上面有插 I-87xxx D/I 板卡，您可以啟用它的“87K DI Counter”功能。每個 I-87xxx 板卡的 D/I 接點可以有一個 D/I Counter 功能。最快可計數到 100 Hz 的輸入訊號。此 DI Counter 值為 16-bit (0 到 32767，之後會降到 -32768 再增加到 -1，16 進位表示為 0000, 0001 ... 到 7FFF, 8000, 8001, ... 到 FFFF, 之後回到 0000, ...).

DI Counter 值使用跟 A/I 接點一樣的 Modbus address 設定。

每個 I-8KE4/8-MTCP 上的 I-87xxx D/I 板卡，最多只能使用 16 個 D/I Counter，即使是像 I-87041: 32 點 D/I 也是只能使用 16 個 D/I Counter。

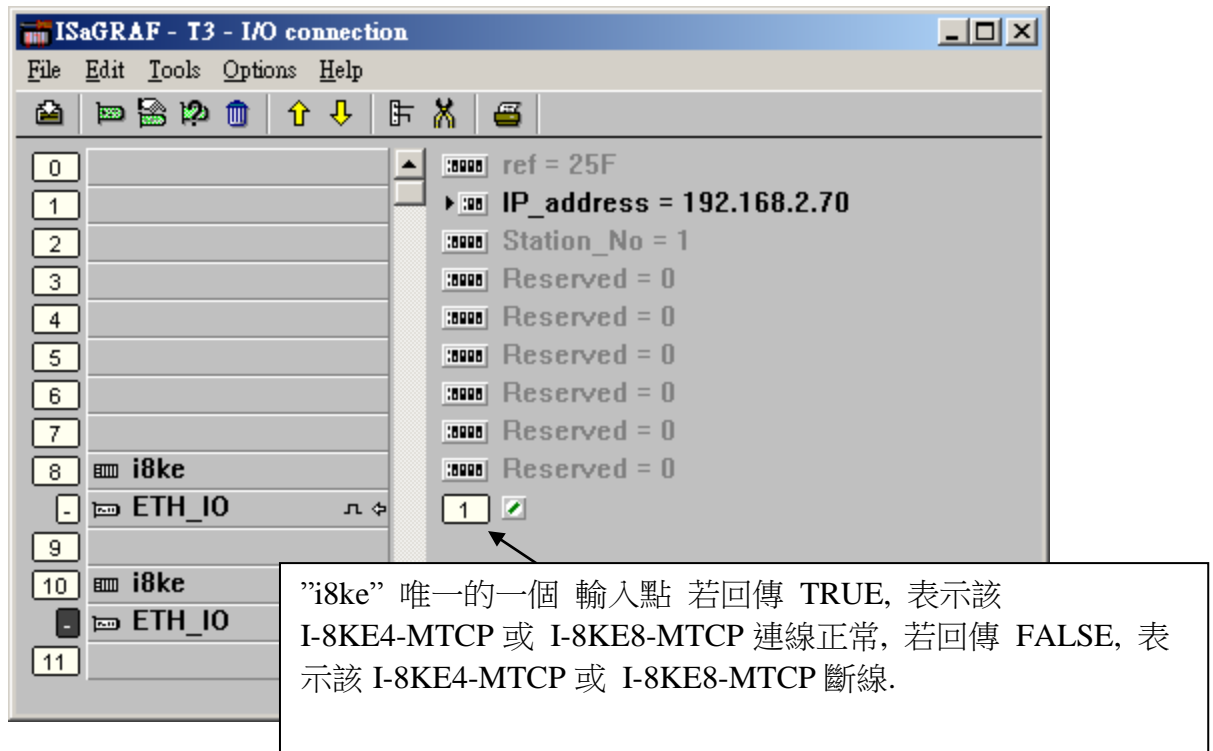


重要事項:

每個 I-8KE4-MTCP 或 I-8KE8-MTCP，當有變更插在其上的 I/O 板卡時，都必須使用 Modbus utility 來重新對它的 I/O 規劃一次。假如有發現到其上的第 2 個與第 3 個 Led 燈一直閃爍的情況，表示 I/O 卡有被動過，需重新用 Modbus Utility 再規劃一次”

步驟 3.

請在 ISaGRAF IO connection 視窗內連上 “i8ke”. 輸入要連上的那台 I-8KE4-MTCP 或 I-8KE8-MTCP 的 IP 地址. 若 Wincon 有連接多個 I-8KE4/8-MTCP, 則必須要在 IO connection 視窗內連上 相同數量的 “i8ke”, 如下.



若您的 ISaGRAF 軟體內找不到 i8ke, i8ke_b, i8ke_n, i8ke_f, i8ke_b_a, i8ke_n_a, i8ke_f_a, 請訪問 <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> 來下載 “ICP DAS Utilities For ISaGRAF.zip” 並執行其內的 setup.exe 來安裝它們到您的 ISaGRAF 內.

步驟 4.

要在 ISaGRAF 程式內把變數對應到 I-8KE4/8-MTCP 上相對的 Modbus address 的 I/O 點, 可以使用下表內的 函式, 詳細函式說明請參考附錄 A.4.

I8KE_B	設定 Boolean 變數 對應到 I-8KE4/8-MTCP 的 1 個 D/I 或 D/O 點
I8KE_B_A	設定 Boolean 變數陣列 對應到 I-8KE4/8-MTCP 的多個 D/I 或 D/O 點
I8KE_F	設定 REAL 變數 對應到 I-8KE4/8-MTCP 的 1 個 A/I 或 A/O 點
I8KE_F_A	設定 REAL 變數陣列 對應到 I-8KE4/8-MTCP 的多個 A/I 或 A/O 點
I8KE_N	設定 Integer 變數 對應到 I-8KE4/8-MTCP 的 1 個 A/I 或 A/O 點
I8KE_N_A	設定 Integer 變數陣列 對應到 I-8KE4/8-MTCP 的多個 A/I 或 A/O 點

範例程式: Wdemo_30 與 Wdemo_31 位於
W-8xx7 CD-ROM: \napdos\isagraf\wincon\demo\ 內 或
ftp://ftp.icpdas.com/pub/cd/wincon_isagraf_napdos_isagraf_wincon_demo/

第 23 章連接 FRNET 快速 I/O

μPAC-7186EG (驅動 1.06 版起), Wincon-8xx7 (驅動 3.42 版起), WinPAC-8xx7 (驅動 1.01 版起), iPAC-8447/8847 (驅動 1.01 版起), VP-25W7/23W7 (驅動 1.02 版起) 與 XP-8xx7-CE6 (驅動 1.01 版起) 有支持 FRnet 快速 Digital I/O.

I-8xx7 (40MHz) 與 I-8437-80, I-8837-80 與 I-7188EG/XG 則不支持 FRnet I/O.

WP-8xx7, VP-2xW7, iPAC-8xx7, XP-8xx7-CE6 與 W-8xx7 需使用 **I-8712W** 來連接 FRNET I/O.

μPAC-7186EG 需使用 FX-016 (x-board) 來連接 FRNET I/O.

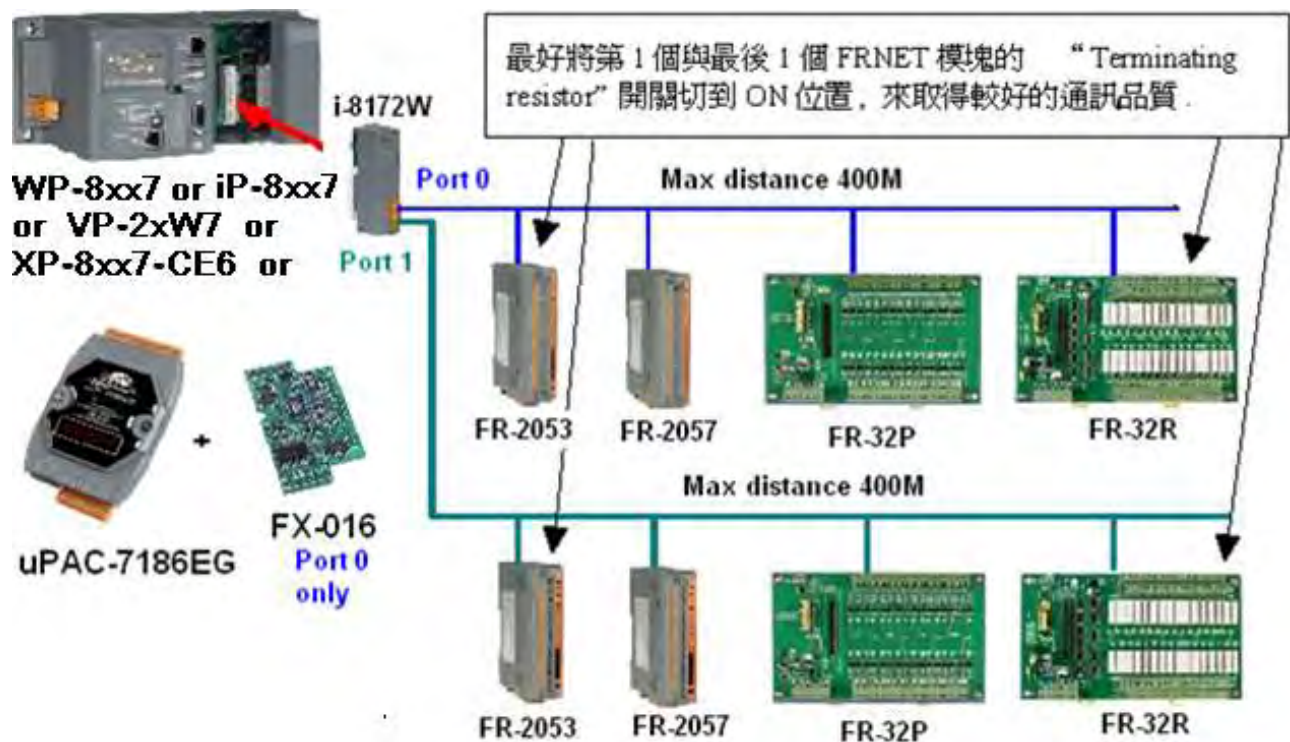
請參考以下取得相關的訊息:

I-8172 / I-8172W 與 FRnet I/O: http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm

μPAC-7186EG + FX-016 (x-board): 請連絡地區經銷商 或 service@icpdas.com
(FX-016 尚未支持 RoHS)

ISaGRAF driver: <http://www.icpdas.com/products/PAC/i-8000/isagraf-link.htm>

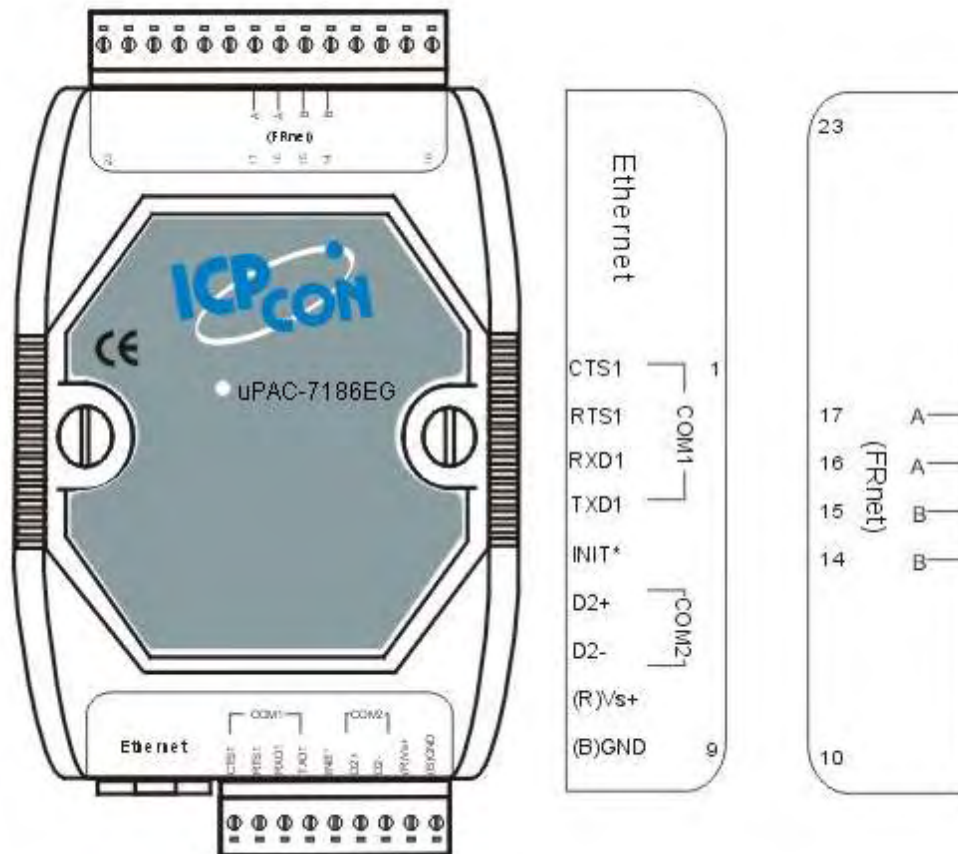
ISaGRAF PAC : <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm>



μPAC-7186EG + FX-016 接腳圖:

請將 FX-016 (x-board) 插入到 μPAC-7186EG 的 slot 0 上 (移開 μPAC-7186EG 的前方外殼, 就可看到 slot 0 的位置).

FX-016 只支持一個 FRNET port . 它的訊號接腳為 A 與 B. (FX-016 上沒有 A2 與 B2)



23.1: FRnet I/O 簡介

重要訊息:

1. 關於 FRNET 主板 :

WP-8xx7, VP-25W7/23W7, iPAC-8xx7 與 XP-8xx7-CE6 只支持 I-8172W, 不支持 I-8172.
Wincon-8xx7 / 8xx6 則 I-8172W 與 I-8172 都支持
μPAC-7186EG 只支持 FX-016, 不支持 I-8172W, 也不支持 I-8172

2. 每個 FRnet 數位 (數字量) 輸出模塊 (Digital Output) 應該都可找到一個 "RESET" 或叫 "HOLD" 切換開關或是一個 "RESET" 功用的 Jumper. 當設為 ON 時 (或啟用它), 會在 I-8172 / I-8172W FRnet 主板 (或 μPAC-7186EG + FX-016) 跟此 FRnet 輸出模塊通訊斷線時, 自動將其上的輸出點都設為安全的 False 值. (例如, 將 FR-2057 的第 8 個 Dip 切換開關設為 ON, 表示啟用此功能)

3. FRnet 數位輸入 (Digital Input) 模塊 與主板 I-8172 / 8172W (或 μPAC-7186EG + FX-016) 的通訊狀態, 可在 ISaGRAF 的 IO connection 視窗中 I-8172 的 Port0 或 Port1 的 8 個 D/I 點上偵測到 (若是 μPAC-7186EG + FX-016, 則在 IO connection 視窗中 "Frnet86" 上偵測到). 然而 FRnet 輸出 (Output) 模塊則無法被 FRNET 主板偵測到它的通訊狀態.

4. WinPAC-8xx7 最多可支持插入 8 塊 I-8172W 於 Slot 0 到 7, Wincon-8xx7 最多可支持插入 7 塊 I-8172 或 I-8172W 於 Slot 1 到 7. XP-8xx7-CE6 最多可支持插入 7 塊 I-8172W 於 Slot 1 到 7. iPAC-8447 / 8847 則最多只支持插入 4 塊 I-8172W 於 Slot 0 到 7 的其中 4 個.
μPAC-7186EG 則只可插入一塊 FX-016 (x-board). VP-2xW7 最多支持插入 3 塊 I-8172W 於 Slot 0 到 2.

FRnet I/O 的優點:

為快速數位 I/O (Digital I/O), 每條單獨的 FRnet 連線上的所有 D/I 與 D/O 的 I/O Scan Time 約只有 3 ms. (當然此 Scan Time 會被 ISaGRAF 的 PLC scan Time 所影響. 比如若某個應用程式的 ISaGRAF PLC Scan Time 約 15ms, 那此應用所連接的 FRnet I/O 的 Scan Time 就會變成是 15 ms, 而非 3ms). 以下是空的 ISaGRAF 程序 (只含 FRnet 設定程序) 大約的 PLC scan time.

	i8172w x 1	i8172w x 2	i8172w x 3	i8172w x 4	5~8 塊 i8172w
WP-8xx7	3 ms	4 ms	4 ms	4 ms	8 ms
XP-8xx7-CE6	3 ms	4 ms	4 ms	4 ms	8 ms
VP-2xW7	(3 ms)	(4 ms)	(4 ms)	(-)	(-)
iP-8xx7	3 ms	6 ms	9 ms	12 ms	-

	1 塊 FX-016
μPAC-7186EG	3 ms

FRnet I/O 目前尚未支持類比 I/O (A/I 與 A/O).

WP-8xx7, VP-25W7/23W7, W-8xx7, XP-8xx7-CE6 或 iPAC-8x47 加 I-8172W 板卡 (與 μ PAC-7186EG 加 FX-016) 可連到 FRNET I/O 模塊,如 FR-2053, FR-2057, FR-2054, FR-32P, FR-32R

http://www.icpdas.com/products/Remote_IO/frnet/frnet_list.htm

每個 I-8172W FRnet 主板有 2 個 FRnet 通訊串口, 編號為 Port 0 與 Port 1 (μ PAC-7186EG 加 FX-016 則只有 1 個 FRNET 通訊串口, 編號為 Port 0).每個 FRnet 串口最多可連接 8 個 FRnet D/O 的 Module Address (正常使用下) 與 8 個 D/I 的 Module Address . 使用時要特別注意, FRnet D/O 的 “Module Address” 設定 (用 Dip Switch 設的) 只能是 0 到 7, 而 D/I 的 “Module Address” 設定只能是 8 到 15.

正常用法是 D/O 與 D/I 的 “Module Address” 設定都需不一樣, 但有一種特殊的用法是 D/O 的 “Module Address” 設定 可以相同, 但這樣使用只是把 D/O 訊號的數量覆製而已, 相同 “Module Address” 設定的 D/O 點的輸出還是一樣的. 所以 1 個 FRnet 串口的 D/O Module Address 數量可以超過 8 個. 但 D/I 則最多只有 8 個 (D/I 的 Module Address 不可設為相同)

每個 FRnet Module Address 的 I/O 點數量最多是 16 點. 因此 1 個 I-8172W 主板最多可接 2 (Ports) x 8 (Module address) x 16 = 256 個 D/I 與 2 x 8 x 16 = 256 個 D/O (1 台 μ PAC-7186EG + FX-016 則最多可使用 128 個 D/I 加 128 個 D/O). 一台 8 槽的 WinPAC-8847 最多可插 8 塊 I-8172W 主板, 所以每台 WinPAC-8847 可接的 FRnet I/O 點最大量是 2048 個 D/I 加 2048 個 D/O. (若是使用 iPAC-8447 / 8847 則最多只可插 4 塊 I-8172W 主板, 可接的 FRnet I/O 點最大量是 1024 個 D/I 加 1024 個 D/O)

注意:

ISaGRAF 3.x 版軟體 (軟件) 可用 “I-8172” (for iP-8xx7, WP-8xx7, W-8xx7) 或 “Frnet86” I/O complex equipment (for μ PAC-7186EG) 與 “fr_16di” 及 “fr_16do” 及 “fr_b_a” 函式來操作 FRnet I/O. 如果你的 ISaGRAF 上找不到這 5 個 I/O lib, 請到以下網址下載 “ICP DAS Utilities For ISaGRAF” <http://www.icpdas.com/products/PAC/i-8000/isagraf.htm> > Driver, 之後執行 “Setup.exe” 來安裝.

或至 www.icpdas.com > FAQ > Software > ISaGRAF > 中文 > 082 來下載 zip 檔, 內含 5 個各別的 IO lib. 然後參考 “ISaGRAF 進階使用手冊” 附錄 A.2 來安裝 c-function : FR_B_A 與

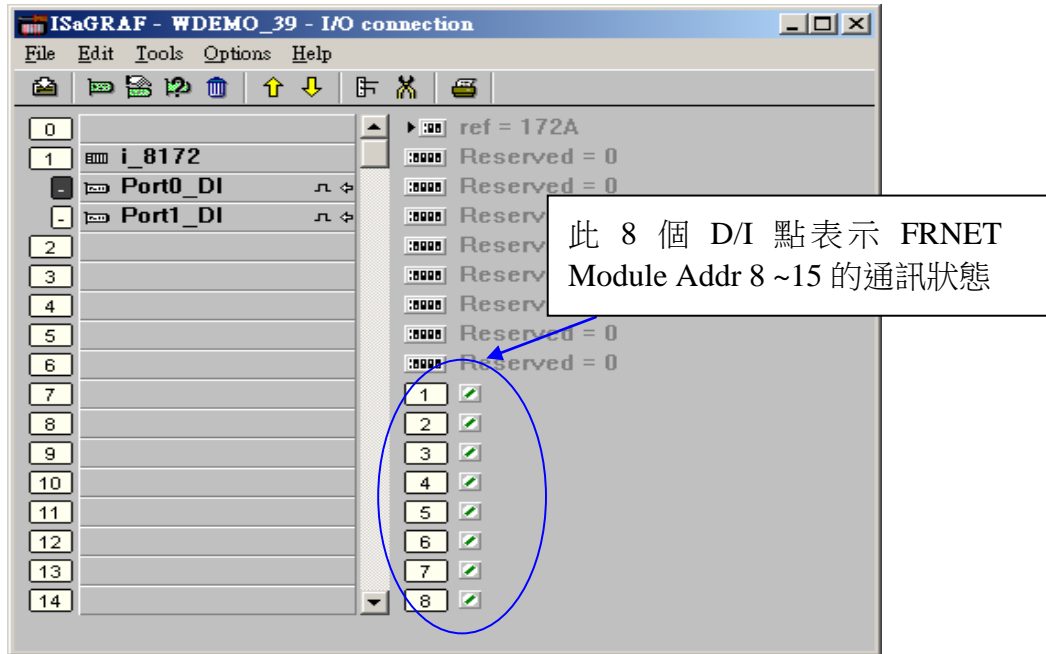
c-function-block : FR_16DI, FR_16DO 與 IO complex-equipment: i_8172, Frnet86 .

範例程式 “wpdmo_70.pia” (給 WP-8xx7, iP-8xx7 使用) 可在 以下位置取得.

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ 或
<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/> 或
www.icpdas.com > FAQ > Software > ISaGRAF > FAQ > 082

23.2: 編寫 FRnet I/O 程式

步驟 1: 在 ISaGRAF I/O connection 視窗內連上 I-8172 板卡 (若是 μ PAC-7186EG + FX-016 則要連上 “Frnet86”). WinPAC-8xx7 需連在對應的插槽 0 到 7 (最多 8 個). iPAC-8447 / 8847 則可以連在 插槽 0 到 7 (最多 4 個), μ PAC-7186EG 則只能連 “Frnet86” 在 插槽 0.



步驟 2: 宣告 ISaGRAF 變數

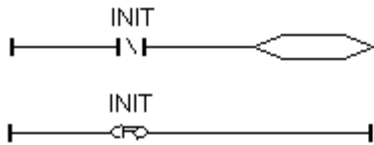
Name	Type	Attribute	Description
INIT	Boolean	Internal	初值設為 True
Dump_di	Boolean	Internal	給未使用的 FRnet D/I 點連接的 Boolean 變數
Dump_do	Boolean	Internal	給未使用的 FRnet D/O 點連接的 Boolean 變數
FR_IN01 ~ FR_IN16	Boolean	Internal	16 個 FRnet D/I 點
FR_OUT01 ~ FR_OUT16	Boolean	Internal	16 個 FRnet D/O 點

步驟 3: 用來設定使用 FRnet I/O 的 LD 程式

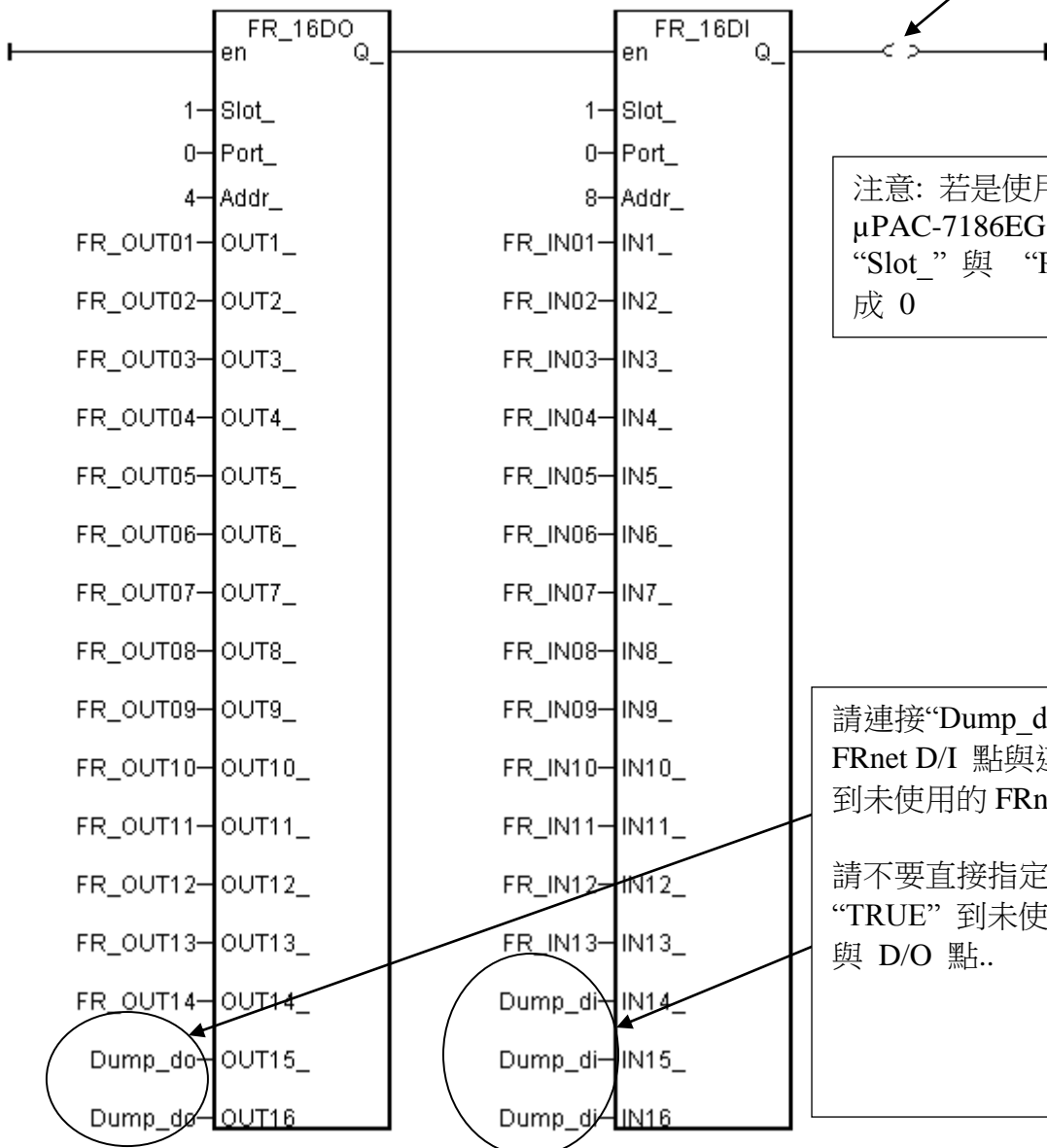
(* 注意: INIT 需宣告初值為 TRUE, 所以以下的 code 只能在第一個 PLC scan 內運行一次, 其它的程序內, 請不要再使用 Fr_16di 與 fr_16do.

另外 FR_16DO 與 FR_16DI 不能使用 陣列變數. 要使用 陣列變數 於 FRNET I/O 請參考第 23.3 節來使用 **FR_B_A** *)

範例 1: FR-2053 (16 IN) 與 FR-2057 (16 OUT)



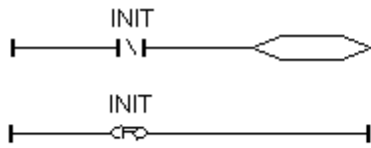
Fr_16di 與 Fr_16do 的 Q_ 都是永遠為 TRUE, 所以可以多個 Fr_16di 與 Fr_16do 連在同一行



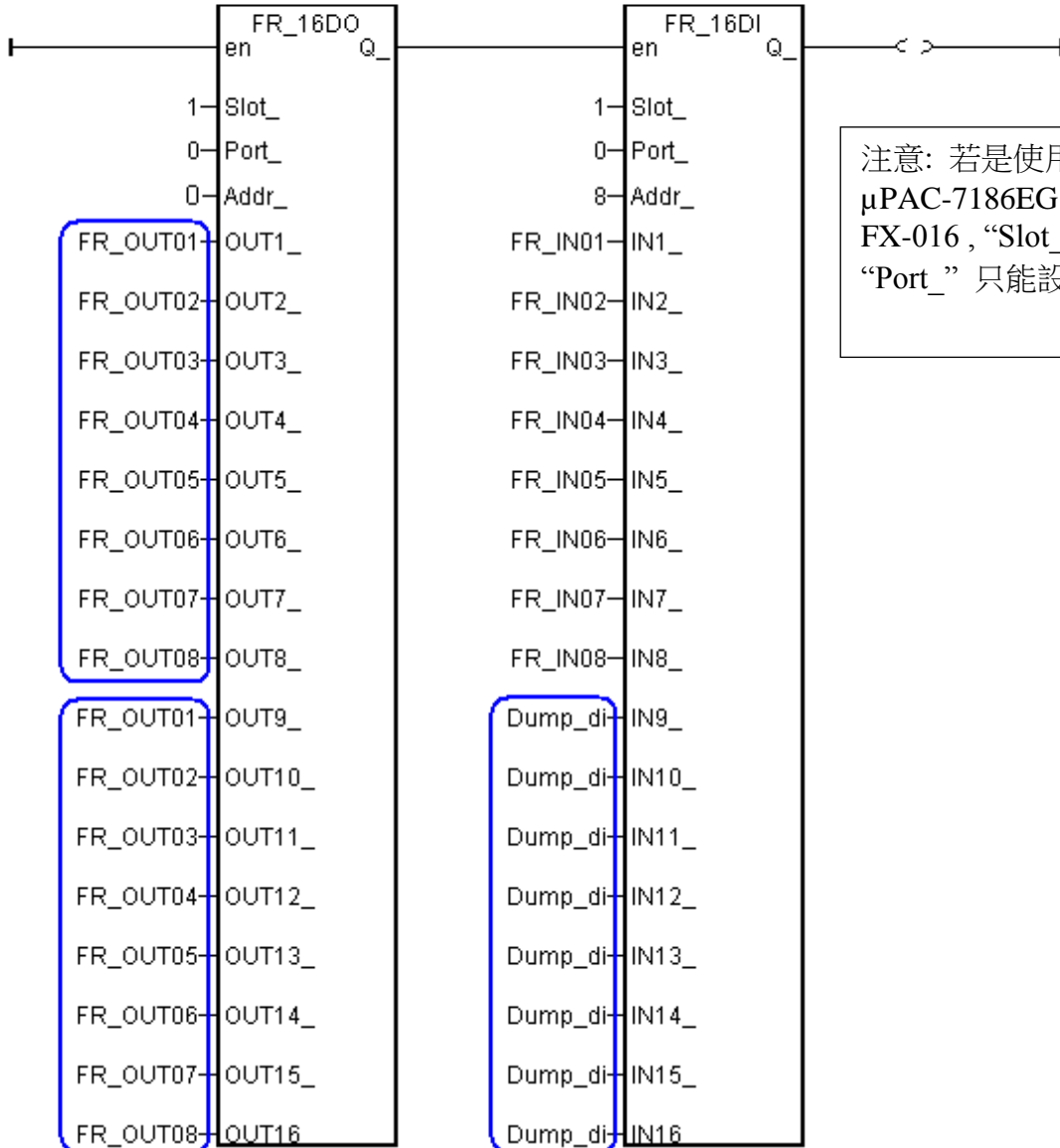
注意: 若是使用 μPAC-7186EG + FX-016, “Slot_” 與 “Port_” 只能設成 0

請連接“Dump_di” 到未使用的 FRnet D/I 點與連接 “Dump_do” 到未使用的 FRnet D/O 點.
請不要直接指定值 “False” 或 “TRUE” 到未使用的 FRnet D/I 與 D/O 點..

範例 2: FR-2054 (8 IN and 8 OUT)



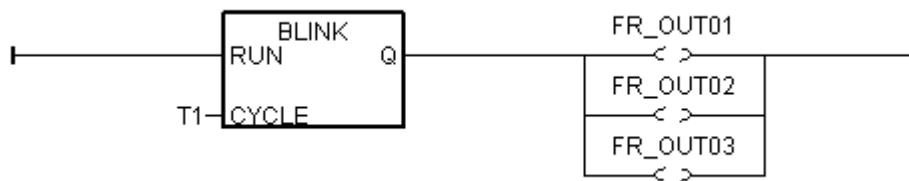
請留意 FR-2054 板卡側邊的 2-Pin 端子接腳為 FRnet 的通訊接腳 (A, B)。但 FR-2053 與 FR-2057 板卡側邊的 2-Pin 端子接腳則為 I/O 的 Power。請勿接錯, 不然板卡會發生損壞



注意: 若是使用 μ PAC-7186EG + FX-016, “Slot_” 與 “Port_” 只能設成 0

若使用的 FRnet I/O 板卡為 FR-2054 (8 IN 與 8 OUT), 它的 Output 使用方式與其它 FRnet I/O 有一些差別. 在它的“FR_16DO” 上必須將第 1 ~ 8 個位置所連接的變數名稱, 同時也連接在第 9 ~ 16 位置上. 另外 FR-2054 的 Dip Switch 上 ADDR 設的是 DO 的 ADDR, 其 DI 的 ADDR 則為 DO 的 ADDR+8, 比如若設成 Dip 3 為 ON, Dip 1 與 2 為 OFF, 表示 DO ADDR = 4, DI ADDR 則為 12. 上方圖形所控制的 FR-2054 則需將 ADDR 設為 0, 所以 DI ADDR 為 8.

步驟 4: 控制 FRnet I/O (以下為每 0.5 秒對 FR_OUT01 到 03 做一次 ON / OFF 方波輸出)



本例如何測？

1. 請將 1 片 I-8172W 板卡插在 WinPAC-8xx7 的 slot 1 內 (WinPAC 最左邊的 I/O slot 是 0)
2. 請連接 Port 0 到 1 個 FR-2053 (16-Ch. D/I) 模塊 與 1 個 FR-2057 (16-Ch. D/O) 模塊。
請將該 FR-2053 的 ADDR 設為 8 (Dip 開關 4 設為 ON, 其它開關 1,2,3,5,6,7,8 設為 OFF)
請將該 FR-2057 的 ADDR 設為 4 (Dip 開關 3 設為 ON, 其它開關 1,2,4,5,6,7,8 設為 OFF)

I-8172W Port 0 的 A 端子 接到 FR-2053 的 A 端子, 再並接到 FR-2057 的 A 端子
I-8172W Port 0 的 B 端子 接到 FR-2053 的 B 端子, 再並接到 FR-2057 的 B 端子

注意:

FRnet D/O 的 Module Address 設定 只能是 0 到 7
FRnet D/I 的 Module Address 設定只能是 8 到 15

3. 之後將 WP-8xx7 或 iPAC-8x47 開機, 下載此 ISaGRAF 程式到其內, 您會看到 FR-2057 的 DO1 到 DO3 每 0.5 秒 ON / OFF 一次.

本範例程式為 “wpdmo_70.pia” 可在 以下位置取得.

WP-8xx7 CD-ROM: \napdos\isagraf\wp-8xx7\demo\ 或
<ftp://ftp.icpdas.com/pub/cd/winpac-8xx7/napdos/isagraf/wp-8xx7/demo/> 或
www.icpdas.com > FAQ > Software > ISaGRAF > 082

23.3: 使用 FR_B_A 來降低程式的大小

有些應用會使用 WinPAC-8x47 或 iPAC-8x47 來連接多個 I-8172W, 且每個 I-8172W 上的 FRNET port 又去連接多個 FRNET I/O. 此時若是使用 FR_16DI 與 FR_16DO 來操作, 程式量會變的很大, 因此可以改採用 FR_B_A 來操作如下 (µPAC-7186EG + FX-016 也支持 FR_B_A, 但 “Slot_” 與 “Port_” 只能設成 0).

本例宣告:

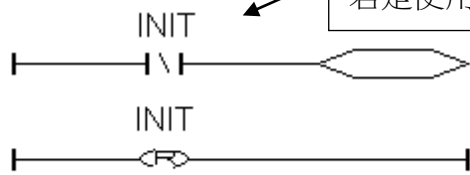
INIT 為 Internal / Boolean, 且指定它的初值為 TRUE

DOUT 為 Boolean 變數陣列, 大小為 128, 即 DOUT[0..127], 指定其 Network addr 為 101 (65h)

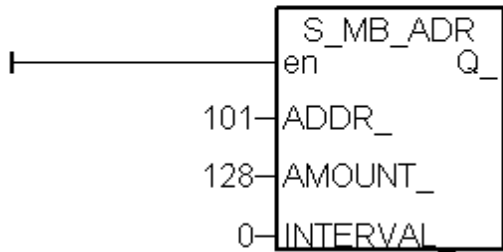
DIN 為 Boolean 變數陣列, 大小為 128, 即 DIN[0..127], 指定其 Network addr 為 229 (E5h)

關於更多變數陣列的說明, 請參訪 www.icpdas.com > FAQ > Software > ISaGRAF > 中文 > 039

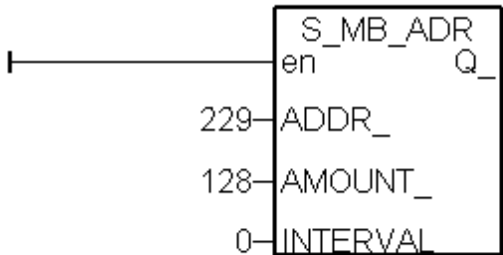
只有第一個 PLC Scan 需要 Run 以下的 code
 若是使用 μ PAC-7186EG + FX-016, "Slot_" 與 "Port_" 只能設成 0



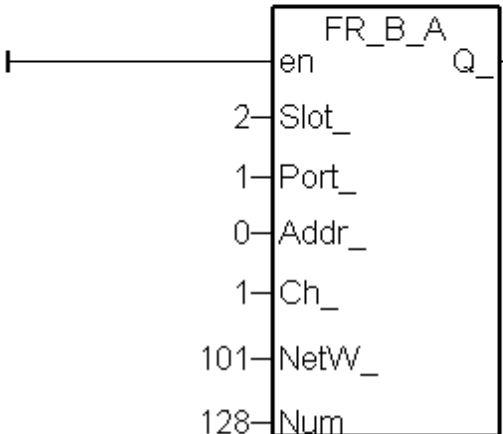
指定 DOUT[0..127]的 Network addr 為 101 到 228 (必需在 Dictionary 內設好 DOUT 的 Network address number 為 101)



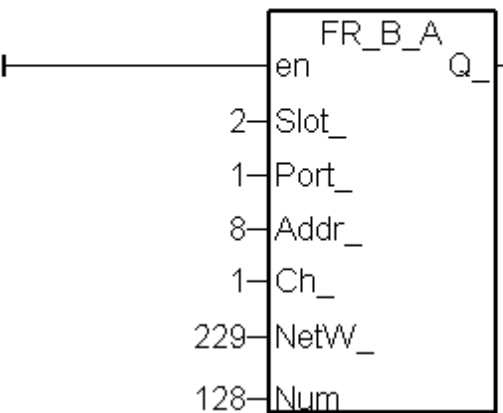
指定 DIN[0..127]的 Network addr 為 229 到 356 (必需在 Dictionary 內設好 DIN 的 Network address number 為 229)
 S_MB_ADR 與 FR_B_A 不能連接在同一列, 因為它們不一定是 Return TRUE



用於 Slot 2 上 I-8172W 的 Port 1, 從 Module Addr 0, Channel 1 開始對應 DOUT 陣列變數到各別的 FRNET DO Channel 上, 共對應 128 個 Channel. 此例會將 DOUT[0] ~ [15] 對應到 Module Addr 為 0 的第 1 ~ 16 的 DO Channel 上, DOUT[16] ~ [31] 對應到 Module Addr 為 1 的第 1 ~ 16 的 DO Channel 上, ..., DOUT[112] ~ [127] 對應到 Module Addr 為 7 的第 1 ~ 16 的 DO Channel 上 .



用於 Slot 2 上 I-8172W 的 Port 1, 從 Module Addr 8, Channel 1 開始對應 DIN 陣列變數到各別的 FRNET DI Channel 上, 共對應 128 個 Channel. 此例會將 DIN[0] ~ [15] 對應到 Module Addr 為 8 的第 1 ~ 16 的 DI Channel 上, DIN[16] ~ [31] 對應到 Module Addr 為 9 的第 1 ~ 16 的 DI Channel 上, ..., DIN[112] ~ [127] 對應到 Module Addr 為 15 的第 1 ~ 16 的 DI Channel 上 .



第 24 章使用 COM Port 函式來讀/寫 RS-232/422/485

ICP DAS ISaGRAF 支持的 Serial COM Port (RS-232/422/485) 通訊規約有以下幾種:

Modbus RTU Slave	參考第 4 章 與 各別產品的規格 或 快速上手手冊
I-7000 與 I-87xxx 遠程 I/O	參考第 6 章
Modbus RTU Master (M-7000)	參考第 8 章 與 第 21 章
Modbus ASCII Master	參考第 8 章
SMS : 收發手機簡訊	參考第 17 章

若有應用需要使用其它通訊規約, 使用者可以使用以下的 Serial COM Port 函式來自行操作. (詳細說明請參考附錄 A.4)

COMOPEN	開啟 Serial COM Port (不含 Flow control 參數)
COMOPEN2	開啟 Serial COM Port (含 Flow control 參數, 不適用 I-8xx7)
COMREADY	偵測有無通訊資料進來
COMARY_R	讀取目前已經收進來的通訊資料,並存入 Byte 陣列內
COMARY_W	寫出 Byte 陣列內的多個 Byte 到某個 COM Port
COMREAD	讀取一個 Byte (需先使用 COMREADY 偵測, 確定有資料才可以收)
COMCLEAR	清空已經收到的資料 Buffer
COMARY_NW	寫出 1 個長整數資料到 COM Port, 格式為 Binary, 4 個 Byte
COMARY_WW	寫出 1 個 Word 資料到 COM Port, 格式為 Binary, 2 個 Byte
COMSTR_W	寫出 1 個字串到 COM Port
COMWRITE	寫出 1 個 Byte 到 COM Port
COMCLOSE	關閉 Serial COM Port

注意:

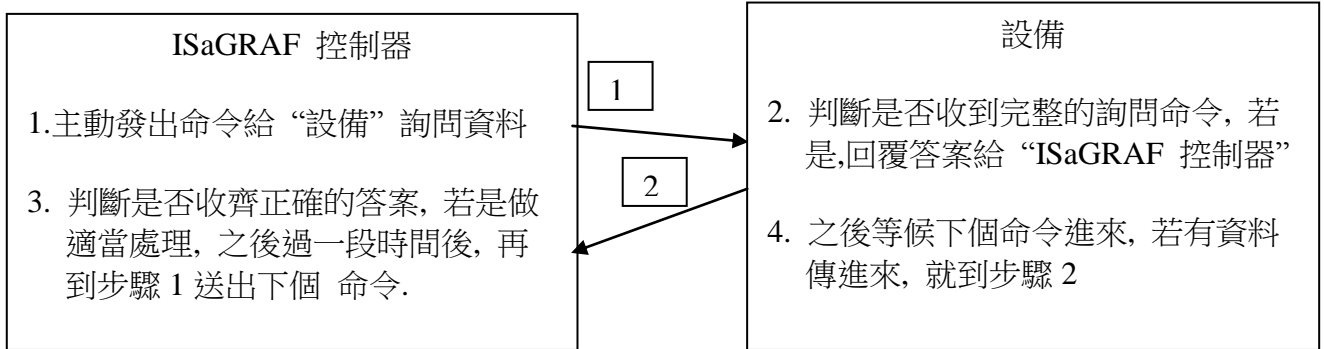
1. I-8xx7 出廠時,它的 COM1 與 COM2(I-8417/8817 才有 COM2) 為 Modbus RTU Slave Port. 其中 COM1 可以關閉 Modbus RTU Slave 功能,改成可以使用上表內的函式來自行操作(參考附錄 C.1).若要使用 I-8xx7 的 COM5 到 COM20,需使用擴充的 I-8112/8114/8142/8144 板卡 (參考第 1.8 節)
2. W-8xx7 / 8xx6 的 COM2 / COM3 可以啟動成為 Modbus RTU Slave Port, 若要自行使用, 請關閉該 Modbus RTU Slave 功能. (請參閱它的包裝盒內附的 “Getting Started:Wincon ISaGRAF PAC” 內的 Appendix A.2). 若要使用 Wincon 上的 COM5 到 COM14, 需使用擴充的 I-8112/8114/8142/8144 板卡 (請參考 “Getting Started:Wincon ISaGRAF PAC” 內的 Appendix E)
3. I-7188EG / 7186EG 的 COM1, 出廠時 內定為 Modbus RTU Slave 功能, 可以關閉該功能,改使用上表的函式. (請參閱它的包裝盒內附的 “快速上手手冊” 內的第 3.6 節). I-7188XG 的 COM1 則永遠支持 Modbus RTU Slave, 無法關閉. 若要使用 I-7188EG/XG, 7186EG 的 COM3 到 COM8, 需選購一片 X5xx 子板, 拔開外殼插在裏面的擴充槽, 之後才能使用

http://www.icpdas.com/products/PAC/i-o_expansion/x_list.htm

以下幾節針對幾個常用的通訊方式分別說明

24.1: 主動對設備 一問一答 方式

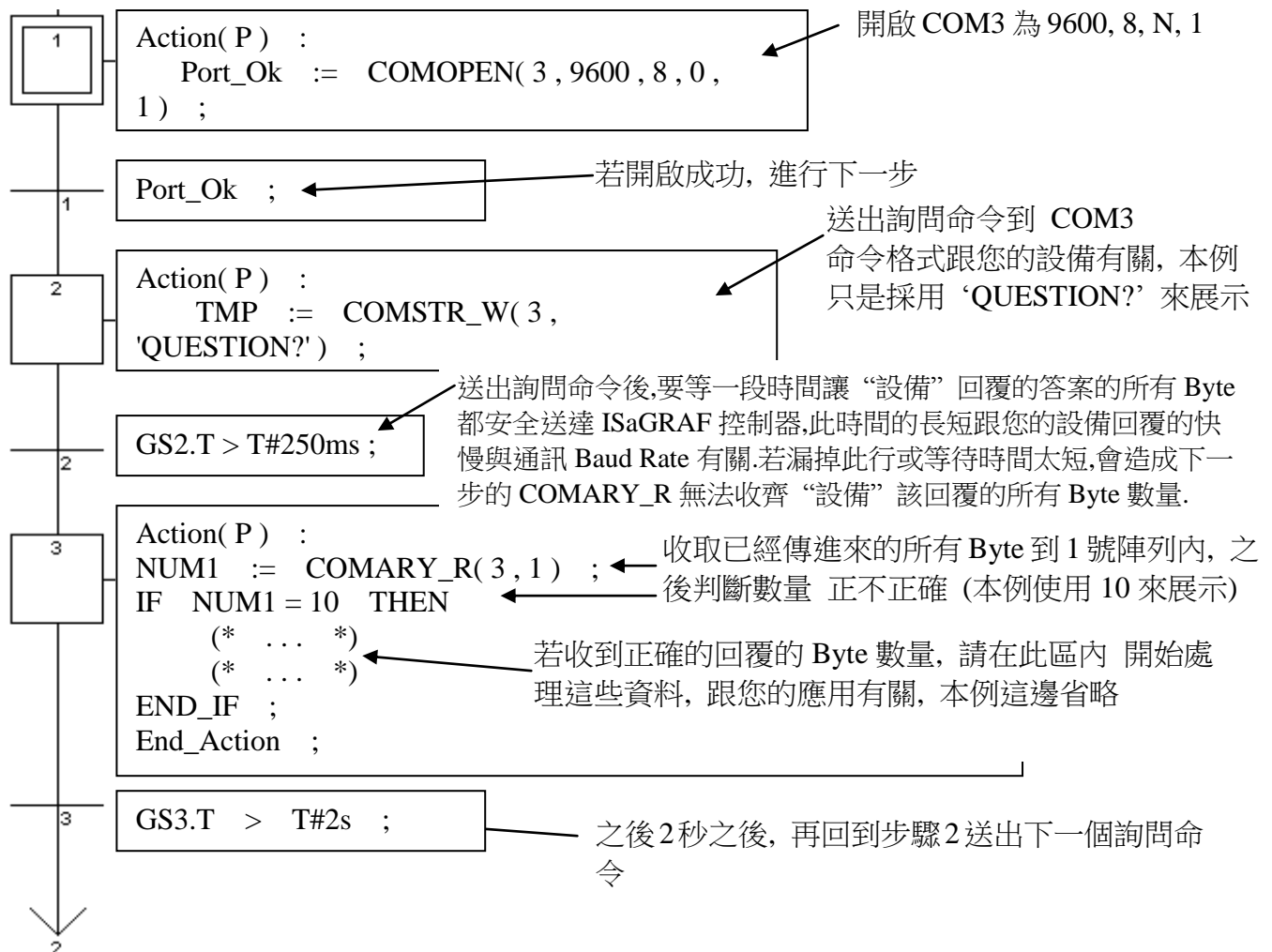
若有應用是要使用 RS-232 / 422 / 485 來取得其它設備的資料，且它的規約是一問一達方式，如下



可以使用類似以下的程式來處理。

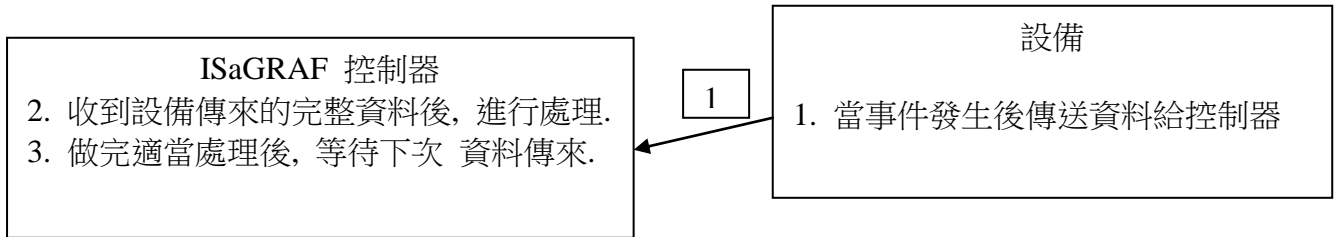
本例先由“ISaGRAF 控制器”由 COM3 送出一個字串‘QUESTION?’給“設備”，之後就準備收取“設備”回覆的答案，之後就再過 2 秒，再送出 同個命令‘QUESTION?’，... 如此重覆進行。

使用 SFC 語法：（本例 Port_OK 與 TMP 宣告為 Boolean Internal，NUM1 宣告為 Integer Internal）



24.2: 被動等待 設備傳資料過來 方式

此種方式在一般的商場或便利商店很常見，比如像條碼閱讀機這種設備，當讀到商品的條碼後，它會傳送 條碼資料 到 ISaGRAF 控制器的 COM Port (RS-232 / 422 / 485)，而且不需要 ISaGRAF 控制器回覆任何訊息給它。(請參訪 www.icpdas.com - FAQ - Software_ISaGRAF - 066 來取得更多的說明與範例程式)



使用 ST 語法:

```

IF INIT THEN
  INIT := FALSE;
  Port_Ok := COMOPEN( 3 , 9600 , 8 , 0 , 1 );
  T1 := T#0s;
  STEP := 0;
END_IF ;
  
```

本例請宣告:

INIT 為 Boolean Internal, 設初值為 TRUE
 Port_Ok 與 TMP 為 Boolean Internal
 T1 為 Timer Internal (計時器)
 STEP 與 NUM1 為 Integer Internal

開啟 COM3 為 9600, 8, N, 1

```

IF Port_Ok = False THEN
  Return ;
END_IF ;
  
```

若開啟失敗，以下的 Code 也不用做了，直接跳出本 ST 程式

CASE STEP OF

```

0: IF COMREADY(3) THEN
  STEP := 1;
  T1 := T#0s;
  Tstart(T1);
END_IF ;
  
```

STEP 為 0 表示等待中，要去檢查是否有資料傳到 COM3
 若 COMREADY 回傳 TRUE，表示至少有 1 個 byte 傳進來，則將 STEP 設成 1，並開始啟用 T1 計時器來計時。

```

1: IF T1 > T#250ms THEN
  Tstop(T1);
  T1 := T#0s;
  STEP := 0;
  NUM1 := COMARY_R(3, 1);
  
```

STEP 為 1 表示資料正在傳進來，要等 T1 計時到一段時間過後，再用 COMARY_R 去收所有 Byte 到 1 號 Byte 陣列內 (這樣才能確保“設備”傳過來的所有 Byte 數量 都已經傳送過來了)。此等待時間跟“設備”的規格與 Baud Rate 有關，本例使用 0.25 秒來展示。(若設的太短有可能會造成有些 Byte 還沒傳送過來而漏掉它們)。要記得將 STEP 設回 0 來等待下次資料再傳進來。

```

IF NUM1=10 THEN
  (* ... *)
END_IF ;
END_CASE ;
  
```

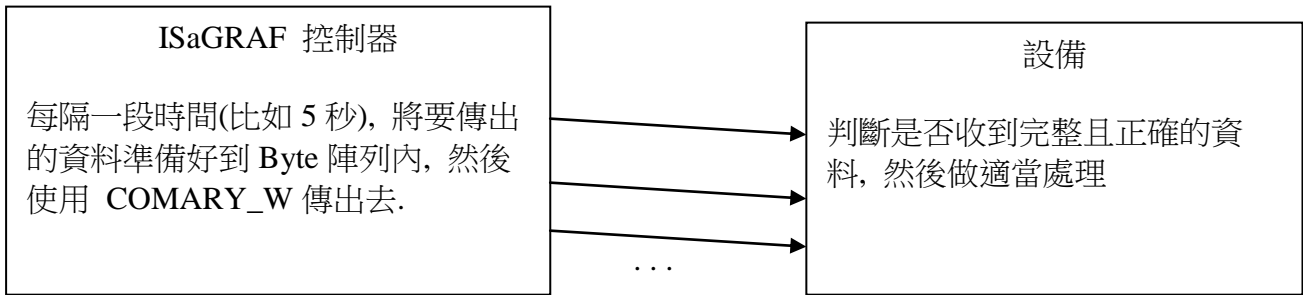
收到資料後，判斷資料對不對，再做適當處理。

本例只判斷 Byte 數量是否為 10 個。實際的情況需依造您的應用來修改。(本展示例 省略“適當處理”的 Code)

END_CASE ;

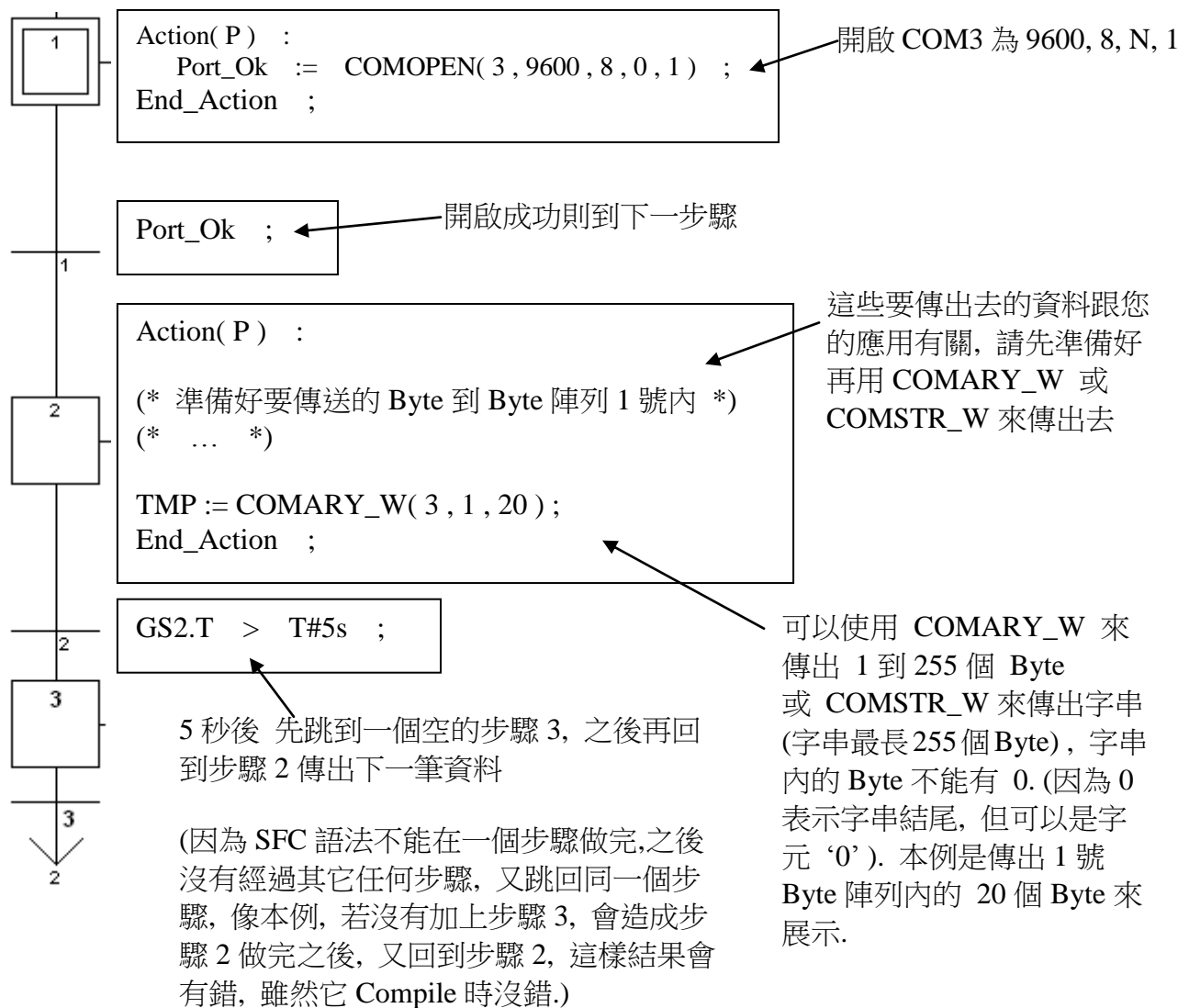
24.3: 每隔一段時間 主動回報資料 方式

若是要每隔一段時間就將回報資料,使用 RS-232 / 422 / 485 COM Port 傳到其它設備或 PC,如下



則可以使用類似以下的程式來處理.

使用 SFC 語法: (本例請宣告 TMP 為 Boolean Internal, Port_Ok 為 Integer Internal)



24.4: 有事件發生時 主動回報資料 方式

請參考第 11.3.5 節的說明.