## Win-GRAF User Manual

By ICP DAS CO., LTD., 2014, All Rights Reserved.

ICP DAS CO., LTD. would like to congratulate you own your purchase of our Win-GRAF PACs. The ease to integration of the controller system and the power of the Win-GRAF software program combine to make a powerful, yet inexpensive industrial process control system.

#### Win-GRAF PAC (Programming Automation Controller) Series of ICP DAS includes:

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

### **Legal Liability**

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

### **Trademarks & Copyright**

Names are used for identification purpose only and may be registered trademarks of their respective companies. Copyright © 2014 by ICP DAS CO., LTD. All rights are reserved.

The version number and release date of this document are listed at the bottom of each page. Users can download the latest document on FTP: <u>http://ftp.icpdas.com/pub/cd/win-graf-workbench-cd/tutorials/</u>

### **Technical Service**

Win-GRAF Web site: http://www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/win-graf.html

New Win-GRAF workbench, Lib and PAC driver:

http://www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/download/ win-graf-driver.html

If you have any problems, please feel free to contact us. Email: <u>service@icpdas.com</u>.

# **Table of Contents**

Win-GRAF	User Manual	1-1	
Lega	Legal Liability1-1		
	lemarks & Copyright		
Tech	nnical Service	1-1	
	ontents		
Chapter 1	Software Installation & Hardware Setting	1-9	
1.1	Installing the Win-GRAF Workbench		
1.2	Run the Win-GRAF Workbench	1-12	
	1.2.1 Win-GRAF Operating Mode	1-13	
	1.2.2 Win-GRAF Operating Environment		
	1.2.3 Win-GRAF Library Manager	1-16	
1.3	Setting the Win-GRAF PAC's IP Address	1-17	
Chapter 2	A Simple Win-GRAF Program	2-1	
2.1	Creating a New Win-GRAF Project	2-1	
	2.1.1 Creating a Template Project (Demo01)	2-1	
	2.1.2 Important Project Settings	2-3	
2.2	Introduction of the Project	2-5	
	2.2.1 Demo01 - LD Program	2-5	
	2.2.2 Demo01 - Variables	2-7	
2.3	Give it a Try		
	2.3.1 Declaring the Win-GRAF Project Variables	2-8	
	2.3.2 Declaring the I/O Variables		
	2.3.3 Creating an LD Program		
	2.3.4 Compiling the Program		
	2.3.5 Download the Program to PAC		
	2.3.6 Testing the Program		
Chapter 3	Modbus Slave: Allow the SCADA/HMI Software to Access Win-GRAF Va	riables 3-1	
3.1	To Enable the Win-GRAF PAC as a Modbus TCP Slave		
3.2	To Enable the Win-GRAF PAC as a Modbus RTU Slave		
Chapter 4	Linking "I/O Boards"	4-1	
4.1	DI/DO Boards		
4.2	i_scale (Conversion Table)	4-4	
4.3	i_8017HW (8/16 channels AI)	4-6	
4.4	i_8024 (4-channel AO)		
4.5	i_87018W (8-channel Al)		

4.6 i_exist (Test if the I/O module exists?)			
4.7			
	4.7.1	i_8084_freq (8-channel Frequency)	4-14
	4.7.2	i_8084_cnt_ch04 (4-channel UP/Down Counter)	4-17
	4.7.3	i_8084_cnt_ch08 (8-channel UP Counter)	4-19
4.8	i_8093	8 (3-axis High Speed Encoder Module)	
4.9	Using	the Count Function for I-8084W, I-8093W, I-87082W, I-87084W, I-7083 a	nd I-7080
	Modul	es	4-23
	4.9.1	COUNTER_START	4-23
	4.9.2	COUNTER_STOP	4-25
	4.9.3	COUNTER_GET	4-26
	4.9.4	COUNTER_STATE	4-27
	4.9.5	COUNTER_RESET	4-28
4.10	Ping i	p (Test an Ethernet/Internet Connection)	
4.11		W (8-channel PWM Output Module)	
Chapter 5	Modh	us Master: connecting to Modbus Slave Devices	E 1
•			
5.1	Enabli	ng the Win-GRAF PAC as a Modbus RTU/ASCII Master (I/O & XV-board)	5-1
	5.1.1	Read DI data	5-3
	5.1.2	Write DO Data	5-6
	5.1.3	Read AI Data	5-8
	5.1.4	Write AO Data (16-bit)	5-11
	5.1.5	Write AO Data (32-bit)	
	5.1.6	How to use the XV Board?	5-15
	5.1.7	Connecting the XV107/ XV107A (8 DI, 8 DO)	
	5.1.8	Connecting the XV110 (16 DI)	
	5.1.9	Connecting the XV111, XV111A (16 DO)	
		Connecting the XV116 (5 DI, 6 Relay)	
		Connecting the XV308 (8 AI, 8 DIO)	
		Connecting the XV310 (4 AI, 2 AO, 4 DI, 4 DO)	
	5.1.13	To Disable/Enable the Modbus RTU/ASCII Master Port	5-37
5.2	Enabli	ng the Win-GRAF PAC as a Modbus TCP/UDP Master (Ethernet I/O)	5-38
	5.2.1	Connecting ET-7000 Series I/O Module	5-43
	5.2.2	Connecting the ET-7060 (6 DI, 6 Relay)	5-46
	5.2.3	Connecting the ET-7018Z (10 AI)	5-48
	5.2.4	To Disable/Enable the Modbus TCP/UDP Master Port	5-49
5.3	Conne	cting the Modbus TCP Slave device has two IP addresses	5-50
5.4	Conne	cting the tGW-700 to Expand Modbus RTU Master Ports	5-55
	5.4.1	Using the tGW-700 Series (Modbus TCP to Modbus RTU/ASCII Gateway)	5-55
	5.4.2	Connecting the tGW-700 Series and the LC-103 module (1 DI, 3 Relay)	5-58

	5.4.3 Test the Demo Project (demo_tgw725.zip)	5-62
Chapter 6	Retain Variable and Data Storage	6-1
6.1	Retain Variable	6-1
	6.1.1 RETAIN_VAR (Retain a Variable)	6-3
	6.1.2 RETAIN_ARY (Retain an Array Variable)	6-4
	6.1.3 RETAIN_FLAG_SET/GET/CLR (Set/Get/Clear the Retain Flag)	6-5
6.2	Retain Variable (Using files)	6-7
6.3	Save Data to EEPROM	6-11
	6.3.1 EEP_READ (Read a Value from the EEPROM)	6-12
	6.3.2 EEP_WRITE (Write a Value to the EEPROM)	6-12
Chapter 7	Exchange Data between PACs (Data Binding)	
Chapter 8	Connecting DCON I/O Modules	8-1
8.1	Setting "DCON" I/O Boards	
8.2	Using I/O Function Blocks	
	8.2.1 "D_7065" Function Block	
	8.2.2 "D_7018Z" Function Block	
	8.2.3 "D_7083" Function Block	
	8.2.4 "D_87084_FREQ" Function Block	
	8.2.5 "D_87084_CNT4" Function Block	
	8.2.6 "D_87084_CNT8" Function Block	
	8.2.7 "DL_100T485" Function Block	
	8.2.8 "D_GPS721" Function Block	
Chapter 9	On Line Change	
9.1	Limitations of "On Line Change"	9-1
9.2	Using "On Line Change"	9-3
Chapter 10	Data/Type Conversion and Using the PAC Time	10-1
10.1	AI Data Conversion	
10.2	AO Data Conversion	
10.3	Data Type Conversion	
10.4	BCD Conversion	
10.5	Pack/Unpack Integer or Boolean	
10.6	Pack/Unpack BYTE, WORD, DWORD	
10.7	Unpack Variable to Byte Array or Pack Byte Array into Variable	
10.8	Get/Set the PAC Time	
Chapter 11	Commonly Used Tools and Useful Tips	11-1
11.1	Upgrade Win-GRAF Libraries	
11.2	Upgrade Win-GRAF Driver	
11.3	Spy List	
11.4	Backup/Restore a Win-GRAF Project	

	11.5	Software Reboot a PAC	11-7
	11.6	Using ST Syntax in LD and FBD	11-8
	11.7	Apply a Recipe on the PAC	11-9
	11.8	Get the Functions and Function Blocks that Supported by the PAC	11-11
	11.9	Upload the Win-GRAF Source Code	11-14
	11.10	Set Up the PAC Password	11-16
	11.11	Using Function Block in the ST Program	11-18
	11.12	How to Protect Your Win-GRAF Program to Avoid Unauthorized Copied?	11-19
Chap	ter 12	Description of Win-GRAF Demo Projects	12-1
	12.1	The List of Demo Programs	12-2
	12.2	Timer Operations	12-4
		12.2.1 Start, Stop and Reset the Timer	12-4
		12.2.2 Periodic Operations	
		12.2.3 Detect the Steady ON or Steady OFF Signal	
		12.2.4 Keep Outputting ON for Some Time after Triggering	
	12.3	Operations of Serial Port Communication	
		12.3.1 Send a String by the COM Port	12_10
		12.3.1 Send a String by the COM Port	
		12.3.3 Wait for Data Coming from the Remote Device to the COM Port	
		12.3.4 Report Data Periodically to the Remote Device by the COM Port	
	12.4	Read/Write Data from/to a File on the PAC	
		12.4.1 Write Data to a File on the PAC	
		12.4.1 Write Data to a File on the PAC	
		12.4.2 Read Data Hom a File on the PAC	
Chap	oter 13	VB.net 2008 Program Running in WP-8xx8 Access to Win-GRAF Variables	<b>13-1</b>
	13.1	Add an Existing Win-GRAF Project from a ZIP	13-1
	13.2	Publishing the Win-GRAF Variable for .NET and Soft-GRAF HMI	13-2
	13.3	Create a new VB.NET project	13-4
		13.3.1 Add Project Reference	13-5
	13.4	Compiling the Application	13-7
	13.5	UserShareNet.dll	13-8
		13.5.1 R/W Boolean Functions	13-8
		13.5.2 Integer R/W Functions	13-10
		13.5.3 R/W Real Variable Functions	13-12
		13.5.4 R/W String Variable Functions	13-14
		13.5.5 How to use VB.NET R/W to Win-GRAF String Variable	13-16
Chap	ter 14	C# .net 2008 Program Running in WP-8xx8 Access to Win-GRAF Variables	s14-1
	14.1	Add an Existing Win-GRAF Project from a ZIP	14-1
	14.2	Publishing the Win-GRAF Variable for .NET	14-1
		Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 1-5	

	14.3	Create a New C# Project	14-1
		14.3.1 Add C# Project Reference	
	14.4	Compiling the Application Program	
	14.5	UserShareNet.DLL	
		14.5.1 R/W Boolean Functions	
		14.5.2 R/W Integer Functions	14-8
		14.5.3 R/W Real variable Functions	
		14.5.4 R/W String variable Functions	
		14.5.5 How to Use C# to Convert Win-GRAF String Variable	
Chap	ter 15	Using eLogger HMI in the Win-GRAF PAC	15-1
	15.1	The Win-GRAF Project	
	15.2	The eLogger Project	
Chap	ter 16	Redundancy	16-1
-	16.1	Features and Architecture	
	16.2	Important Communication Ports and Installation Notes	
	16.3	Description of Win-GRAF Demo Projects	
		16.3.1 "I/O Board" Settings	16-6
		16.3.2 Declaring Variables (demo_RDN_2)	
		16.3.3 Introduction of the "demo RDN 2" Project	
		16.3.4 Introduction of the "demo_RDN_4" Project	
	16.4	Test Demo Projects	
		16.4.1 Test the "demo_RDN_2" and "demo_RDN_3" Project	16-17
		16.4.2 Test the "demo_RDN_4" Project	
	16.5	What Kinds of Data Can be Automatically Backed up to the Passive PAC?	
Char			
Спар		Schedule Control	
	17.1	Install the Schedule-Control Utility and Restore the Win-GRAF Demo Project.	
	17.2	Introduction of the "demo_schedule" Project	
	17.3	Edit Schedule Configurations by the Schedule-Control Utility Testing the "demo schedule" Project	
	17.4 17.5	Configurations of the Schedule-Control Utility	
	17.5		
		17.5.1 Address for each Target Variables	
		17.5.2 Target Configuration	
		17.5.3 Season Configuration	
		17.5.4 Normal Day / Holiday / Special Day Configuration	
		<ul><li>17.5.5 Schedule Configuration</li><li>17.5.6 Save and Send the File to the PAC</li></ul>	
		17.5.6 Save and Send the File to the PAC	
		17.5.7 Time Synchronization	
		17.5.9 Using Schedule-Control in the eLogger HMI	
		Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 1-6	

Chap	ter 18	Develop Your Own Function and Function Block18-	·1
	18.1	The Development Process of Your Own Function or Function Block	-2
	18.2	Creating the Compiler Development Environment18-	-3
		18.2.1 Install the SDK of the ViewPAC or the WinPAC	-3
		18.2.2 Install the SDK of the XPAC (XP-8xx7-CE6, XP-8xx7-Atom-CE6)	-6
	18.3	Define Function or Function Block	-7
		18.3.1 Define Function Lib	-7
		18.3.2 Define Function Block Lib	0
	18.4	Edit the Logic of the Function and Function Block	2
		18.4.1 Edit the "T5BLOCKS.cpp"	4
		18.4.2 Edit the Logic of the Function (In this example is "bytes_to_long. c") 18-1	.5
		18.4.3 Edit the Logic of the Function Block (In this example is "long_to_bytes.c" ) 18-1	
		18.4.4 Trying to Compile the Project	.8
	18.5	Test your own Function and Function Block	.9
Chap	ter 19	Using 3G Modules - I-8212W-3GWA19-	·1
	19.1	Hardware Installation19-	-2
	19.2	Software Installation19-	-4
		19.2.1 Install the I-8212W-3GWA (or I-8213-3GWA) Driver	-4
		19.2.2 Configure the 3G/2G Dial-up Parameters	-5
		19.2.3 Important Configuration (DO NOT ignore it)!!!	
		19.2.4 Enable "Dial_up_utility_win_graf" " Dial-up Software 19-1	.2
	19.3	Function Descriptions for Controlling 3G/2G Connection 19-1	.3
Chap	ter 20	Sending a PAC File to a Remote PC via Ethernet or 3G/2G Wireless Networks .20-	·1
	20.1	Description of the "WG-Communication-Server" Software	-2
	20.2	"Send_File_To", "Send_File_State" and "Send_File_Abort" Functions	-4
	20.3	Description of the Win-GRAF Demo Project (demo_send_file.zip) 20-	
	20.4	Test for File Sending	-9
Chap	ter 21	Win-GRAF SMS Function21-	·1
	21.1	"GSM_Open", "Send_SMS" and "Read_SMS" Functions	-1
	21.2	Description of the Win-GRAF Demo Project (Demo_SMS.zip)21-	-4
	21.3	Test for SMS Messaging	-6
Chap	ter 22	The Intelligent Win-GRAF 3G Solution22-	·1
	22.1	Set Up the PC/WG-Communication-Server22-	-2
	22.2	Set Up a Remote PAC to Connect to the WG-Communication-Server	-5
	22.3	Set Up the WG-Communication-Client on a User PC or the SCADA PC 22-	-7
		22.3.1 Logging to the Server and detect the PAC connection status	-7
		22.3.2 Remotely update the Win-GRAF project from the PC/Win-GRAF Workbench 22-1	0

	22.3.3 To read/write the PAC data remotely from the PC/SCADA 22-	12
Chapter 23	HART Master23	-1
23.1	Introduction of the I-87H17W HART Module23	-1
23.2	The Format of the HART Protocol23	-3
23.3	Introduction of the Win-GRAF Demo Project23	-5
	23.3.1 I/O Board Setting and the HART Functions23	-5
	23.3.2 The Demo Project - "Demo_HART_1"	10
	23.3.3 The Demo Project - "Demo_HART_2"	13
	23.3.4 The Demo Project - "Demo_HART_3"	19
23.4	Test the Demo Program	21
	23.4.1 Testing Environment and the "HC_Tool" Setting	21
	23.4.2 Test the "Demo_HART_1", "Demo_HART_2" and "Demo_HART_3" Projects 23-3	23
	A Data types and Ranges	
••	Troubleshooting while On-Line the PAC	
••	Enable the Screen Saver of WinCE PAC	
••	Using Expansion RS-232/485/422	
••	Enabling a Serial Port for Connecting the Win-GRAF Workbench	
••	Pin Assignment of PAC's Serial Ports Automatically and Periodically Synchronizing the PAC Time over a Network	
G.1	Set up an SNTP Client for Network Time Synchronization	11
G.2	Set up a Windows XP PC as the SNTP Server to test the SNTP Client	13
G.3	Set up the Gateway and DNS Server Addresses for the PAC	17

## Chapter 1 Software Installation & Hardware Setting

### 1.1 Installing the Win-GRAF Workbench

Before installing the Win-GRAF Workbench, check the installation environment on your PC.

#### System requirements:

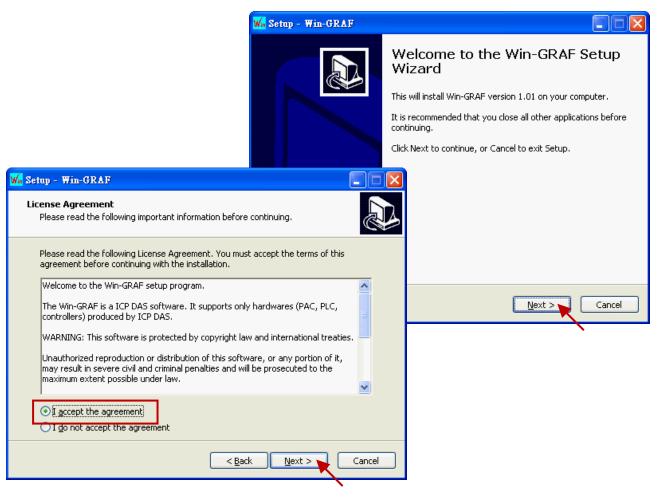
- O.S.: Windows XP, Windows Vista, Windows 7, Windows 8 (32-bits or 64-bits)
- Microsoft .Net Framework 3.5 (Download it on the Microsoft web site: <u>http://www.microsoft.com/zh-tw/download/details.aspx?id=22</u>)
- **RAM:** 1 GB minimum (Recommended: 2 GB or more)
- Available hard-disk space: 200 MB minimum

#### **Installation Steps:**

 Double-click the "Win-GRAF-setup-ver-x.xx.exe" file in the Win-GRAF installation CD (or download the latest version of the Win-GRAF Workbench on the website: <u>http://www.icpdas.com/root/</u> <u>product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.html</u>) to begin the process.



2. Click "Next" to continue and then select "I accept the agreement", then click "Next" to continue.



3. Recommend to use the default installation path (i.e., "C:\Win-GRAF") and then click "Next" to continue.

🖬 Setup - Win-GRAF 📃 🗖 🔀
Select Destination Location Where should Win-GRAF be installed?
Setup will install Win-GRAF into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Win-GRAF Browse
At least 77.4 MB of free disk space is required.
< Back Next > Cancel

4. Click "Next" to add a "Win-GRAF" folder shortcut in the "Start" menu, and then select "Create a desktop icon" to add a desktop shortcut, then click "Next" to continue.

	<mark>₩</mark> Setup - Win-GRAF	
	Select Start Menu Folder Where should Setup place the program	's shortcuts?
	****	s shortcuts in the following Start Menu folder.
	To continue, click Next. If you would lik	e to select a different folder, click Browse.
Win Setup - Win-GRAF		
Select Additional Tasks Which additional tasks should be performed?		
Select the additional tasks you would like Setup then click Next.	to perform while installing Win-GRAF,	
Additional icons:		< Back Next > Cancel
		•
C	< Back Next > Cancel	

5. Click "Install" to begin installing the Win-GRAF Workbench.

Wa Setup - Win-GRAF	
Ready to Install Setup is now ready to begin installing Win-GRAF on your computer.	
Click Install to continue with the installation, or click Back if you want to review or change any settings.	
Destination location: C:\Win-GRAF Start Menu folder: Win-GRAF	~
Additional tasks: Additional icons: Create a desktop icon	
< <u>B</u> ack Install	Cancel

- 6. Before the end of the installation, you will see a pop up window and it displays:
  - a. The legal Win-GRAF Workbench is delivered with a legal Win-GRAF Dongle distributed by ICP DAS. Please always plug the Win-GRAF Dongle in your PC while running it.
  - b. To run the Win-GRAF, require Microsoft ".Net FrameWork 3.5" installed in your PC.

₩a Setup - Win-GRAF	
Installing Please wait while Setup installs Win-GRAF or	n your computer.
Extracting files C:\Win-GRAF\K5HelpGR.chm	W/a Setup - Win-GRAF         Information         Please read the following important information before continuing.         When you are ready to continue with Setup, click Next.         Thank you for installing the Win-GRAF software.         The legal Win-GRAF software is delivered with a legal Win-GRAF-Dongle distributed by ICP DAS. Please always plug the Win-GRAF-Dongle in your PC while running it.         To run the Win-GRAF, require MicroSoft ".Net FrameWork 3.5" installed in your PC.         For technical support and sales service, contact service@icpdas.com .         ICP DAS CO., LTD         www.icpdas.com         service@icpdas.com
	Next >

7. By now, you have completed the Win-GRAF installation, then click "Finish" to exit this window. (Select "Launch Win-GRAF" to auto-run the Win-GRAF after completing the setup. If there is no Win-GRAF Dongle in your PC, the Win-GRAF Workbench will run in Demo mode.)



## 1.2 Run the Win-GRAF Workbench

Before running the Win-GRAF Workbench, make sure the Win-GRAF Dongle is plugged into your PC. Without using a Win-GRAF Dongle, the Win-GRAF Workbench will run in Demo Mode. Open the Windows Start menu, click on "Win-GRAF" folder and "Win-GRAF" to open this software.



### Description of the "Win-GRAF" folder:

Libraries: For users to create their own function or modify an exist function.

Handbook: The manual details the software interface, programming environment, programming languages, and so on, provided by COPALP.

(Or click the [Help] > [Topics] from the Win-GRAF menu bar)

- History: The modification history and features added of the Win-GRAF Workbench.
- Manual: The Win-GRAF manual provided by ICP DAS. (Or click the [Help] > [Tutorials] from the Win-GRAF menu bar, the manual is located in the path "C:\Win-GRAF\Tutorials")
- ReadMe: The notice for the Win-GRAF Workbench.



### 1.2.1 Win-GRAF Operating Mode

The Win-GRAF Workbench provides two operating modes:

"Demo Mode": Without using a Win-GRAF Dongle. The compiled Win-GRAF project can run for 15 minutes on the PAC. Once the time limit has expired, users must Stop/Start this project again and it only supports up to 40 I/O tags.

"Large Mode": Using a Win-GRAF Dongle. The project can run on the PAC without the time limit.

#### **<u>Demo Mode</u>** - Without using a Win-GRAF Dongle.

The start screen will show as below after running the Win-GRAF Workbench.

Wa Win-GRAF Starter	
No Win-GRAF-Dongle found. Demo mode !!!	No Win-GRAF Dongle found. Demo mode!
Win-G	RAF Version 1.01 (MAR.18,2014)
It describes the limitations in demo mode, you can also click the main menu [Help] > [About] to see this content.	Copyright by ICP DAS CO., LTD. since 2014. ********* [Demo mode] ********* For [Large mode], please plug in the 'Win-GRAF-Dongle' while running it. For technical support and sales service, contact service@icpdas.com . Limitations in demo mode: - Applications are limited to 40 I/Os - The code generated by the compiler stops after 15 minutes - The simulation stops after 15 minutes Limitations of controller type: - The Win-GRAF software supports only hardwares (PAC, PLC, controllers) produced by ICP DAS.

**Note:** If you install the Win-GRAF Dongle in the Demo Mode, you must close Win-GRAF Workbench and then start it again to make it become Large Mode.

Large Mode - Using a Win-GRAF Dongle.

The start screen will show as below after running the Win-GRAF Workbench.



### 1.2.2 Win-GRAF Operating Environment

🚾 Win-GRAF 還原(R) <u>File V</u>iew <u>T</u>ools <u>W</u>indow <u>H</u>elp Menu / Tool bar **Right-click** No li ⊠ X Workspace □ 最大化(数) × 開閉(C) Alt+F4 Show/Hide menu **Program Area** Workspace Build Message Area Build Cross references Runtime Call stack Breakpoints Digital sampling trace Prompt HMI 100% Ready 0,0 No project 0,0  $0 \times 0$ 

Run the Win-GRAF and then the main screen will show as below:

<u>Note:</u> Mouse right-click on the top of the Window to Show/Hide the menu bar.

- A. The Workspace: It allows users to create project lists, and add/open the Win-GRAF program as well as the related settings. Moreover, the new project can be created by using a project template.
- B. The Program Area: It used to show/edit the program and can be divided into more function area. (Refer the <u>Section 2.2.1</u>)
- C. The Message Area: It used to show compiler messages and provides more diagnostic tools.

### **Tips:**

- 1. To resize a window, click and drag the side or corner of the window to change its size.
- 2. Press the "F1" key to open the user manual (i.e., HTML Help).

#### Hind or Show the window

If you carelessly closed the Variables pane or the Message Area during the programming, you can click the menu bar "View" and select the following options to open this window again.

§ι.	Local <u>D</u> efines	
	Output	Ctrl+Alt+O
	Infos Tabl	
	I <u>n</u> fos Tab2	

Output: It means the Message Area.

Infos Tab1: It means the Program Area – Variables pane (refer the <u>Section 2.2.1</u>).

Infos Tab2: It means the Program Area - Function Blocks pane (refer the <u>Section 2.2.1</u>).

#### Switching the language

1. To switch the UI language of the Win-GRAF, click "Help" and "Language" on the toolbar and then choose a language in the "Select Language" dialog box, and click "OK".

X

۸

.

Ŧ

Win-GRAF		Select Language
File View Tools Window	Help       Iopics       Search       Tutorials       About	Fran種is Deutsch Italiano 医spa察I 灸、Korean Traditional Chinese Simplified Chinese
	Language	

2. The UI language will be changed after restarting the Win-GRAF automatically.

Win-GRAF		
File Visualizza Strumenti	i Finestra Guida	
CHEV A A PA	🛝 🗙 🖏 🔄 🤊 연 🟭 🏢 🏭 19 😨 🖉 🗐 👘 🎽 🖉	
Workspace	Nessuna selezione!	∎ Z X
	Compila	×
	oonpina	^
	Compila Lista incrociata Runtime Call Stack Breakpoint Digital Sampling Trace Prompt	HMI Code Checker
Pronto	Nessun progetto 0, 0	0 x 0 0,0 📑

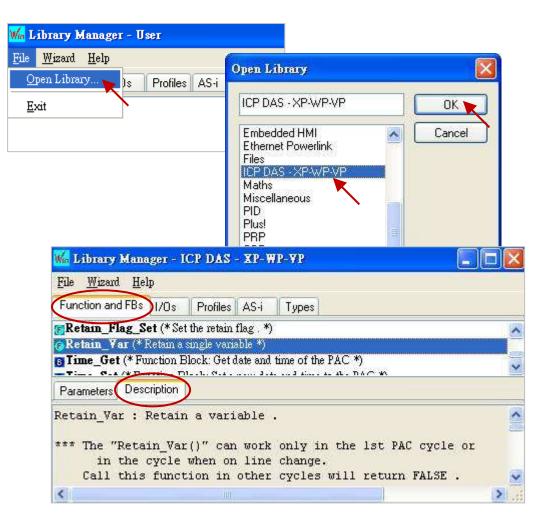
### 1.2.3 Win-GRAF Library Manager

Win-GRAF Workbench provides a Library Manager that can be used to look up all descriptions for Functions, Function Blocks and I/O Boards. The user can refer <u>Section 11.1</u> to upgrade the Win-GRAF Lib.

1. To begin this, click the Start button and click "All Programs" > "Win-GRAF" > "Libraries" > "OEM".



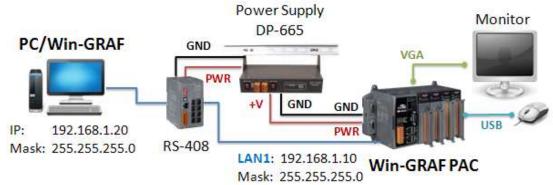
- 2. In the "Library Manager" window, click the menu bar "File" > "Open Library" and select "ICP DAS XP-WP-VP" then click "OK".
- 3. Select any title in the "Function and FBs" tab and click the "Description" to view the usage of this Function or Function Block; Select any title in the "I/Os" tab and click the "Description" to view the usage of this I/O Boards.



## 1.3 Setting the Win-GRAF PAC's IP Address

For connecting with the PAC, the Win-GRAF Workbench needs to know the PAC IP. The following will show you how to set up the PAC IP. Using the XPAC (XP-8xx8-CE6) and the WinPAC (WP-5xx8-CE7, WP-8x48, WP-8x28-CE7) as the example:

#### Hardware Wiring Diagram



#### PAC Side

Using the USB mouse that connected to the PAC, and click "Start" > "Settings" > "Network and Dial-up Connections" on the lower left corner of the monitor, then double-click the "LAN1" (or LAN2), then fill in a proper IP address.

<u>F</u> ile <u>E</u> dit	<u>V</u> iew Ad	Iva <u>n</u> ced 🗙 💽 💼 📰	× ×
-	<u>7</u>		
Make New	LAN2	LAN1	
Connection		2	
		'LAN1 Fast Ethernet Adapte	er' Settings
		IP Address Name Servers	3
		An IP address can be automatically assigned to this	O Obtain an IP address via DHCP
		computer. If your network does not automatically assign	Specify an IP address
		IP addresses, ask your network administrator for an address,	
		and then type it in the space provided.	Subnet Mask:         255.255.255.0           Default Gateway:
		provided.	
Drograms	•		
🔆 F <u>a</u> vorites		1	Input Panel
Document Settings		rol Panel	[sc] 1 2 3 4 5 6 7 8 9 0 - = ◆ Tab q w e r t y u i o p [ ]
ettings 🖉		vork and Dial-up Connections	
2 <u>R</u> un		bar and Start Menu 🥄 👘	Shift z x c v b n m , . / ← Cti]áü ` \ \ ↓ ↑ ★ +
🛃 🚳 Netwo	ork Connecti	zno	10-29 AN 🛗 🔁
			Click to use the keyboard

Open the "WinPAC\_Utility.exe" (or "XPAC\_Utility.exe") on the desktop (or \System Disk\tools\).Then, click "File" > "Save and Reboot" to reboot the PAC.

#### Notice:

**For connecting properly, the PC/Win-GRAF IP and the PAC IP must on the same network segment.** For example, set the PC's IP to "192.168.1.20" (Mask: 255.255.255.0).

My Device Win GRA	
WinPAC Utility [2.1.0.4]     Internet   Explorer   Save   Reboot   WinPAC   WinPAC   WinPAC               WinPAC                                  WinPAC                        WinPAC	
Documents     Frequency : 55.8       WINPAC     Battery1 : OK   Task Bar Settings: Auto Hide Always On Top	
Battery2 : OK     Blank Desktop       RegView     Configure synchronization with a time server     Configure       Image: Desktop     Image: Desktop     Image: Desktop       Image: Desktop     Image: Desktop     Image: Desktop	
isqlw35	
	2

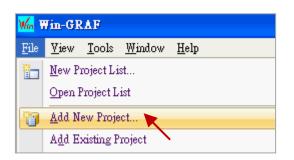
## Chapter 2 A Simple Win-GRAF Program

### 2.1 Creating a New Win-GRAF Project

The following sections will introduce you to a simple template project that used to get/set (read/write) the Win-GRAF PAC's system time. Follow the steps below to complete this demo program.

### 2.1.1 Creating a Template Project (Demo01)

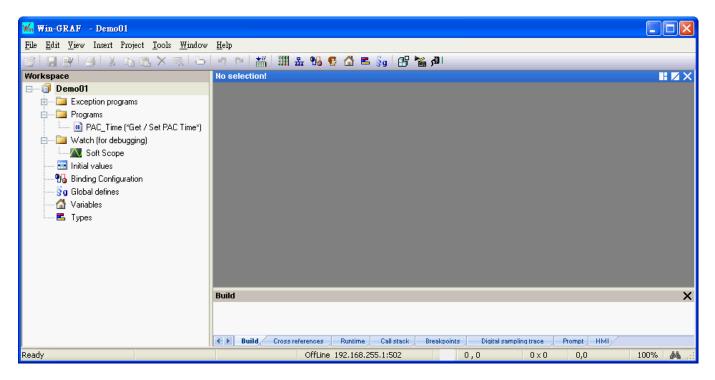
1. Run the Win-GRAF Workbench (refer the <u>Section 1.2</u>), and click "File / Add New Project..." from the menu bar.



2. Click "From template" to create a project from a template, enter a project name (e.g., "Demo01") in the Name field and add a simple note in the Comment field, then click "Next". By default, it will show a "ICPDAS\_template" option provided by Win-GRAF Workbench, just click "Next" to continue.

🖹 Project wizar	d			$\mathbf{X}$
Project From template XML Import Library				
				0 0 0-0- 0 0 0-0- 0-0-
Creates a new pro	pject using a template	)		<u>^</u>
New project-			Recommend to use the def	ault folder.
Destination folder	: C:\Win-GRAF\F	Projects		<u>B</u> rowse
Name:	Demo01			
Comment:	Test Demo			
			Next Cancel	<u>H</u> elp
	Template:	ICPDAS_template	3	
			Previous	Next

3. Now, you have created the "Demo01" template project.



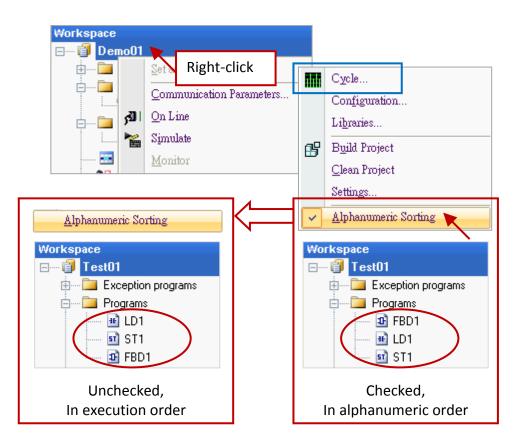
Note: In the demo01, we use a "From template" way to create this project. If you select "Project" in the step2, click the "Release" in the "Compiling options" setting. The others setting can be done in the following sections, just click "Next" and then "OK" button to end the settings.

gs			
rograms			
Language:	LD: Ladder Diagram		~
Compiling opti	ons		
	O Debug		
(	Release		
Communication	n options		
Settings:	192.168.71.19:502	"PAC IP:502",	
Protocol:	T5 Runtime	See Section 2.3.5	~
)ther			
Edit initia	l values with the Recipe edito	n	

### 2.1.2 Important Project Settings

There are two important settings must be done after creating the project.

 In the "Workspace", mouse right-click the project name (e.g., "Demo01") and then uncheck the "Alphanumeric Sorting" option (the last one). If unchecked, means the programs are in execution order ; If checked, means the programs are in alphanumeric order (e.g., <u>FBD1</u>, <u>LD1</u>, <u>ST1</u>).



### Change the execution order of programs:

<u>Note:</u> If you want to change the execution order of programs, mouse right-click the project name (e.g., "Demo01") and click "Cycle" (as the screenshot above) to open the settings window, then click the "Move Up" or "Move Down" button to change the order.

Cycle			
🖪 🖪 🔤 💉 🗲 🔶 🛷 🧠			
Name	Enabled	Period	Phase
PAC_Time	<ul> <li>Image: A set of the set of the</li></ul>	1	0
<mark>sī</mark> pStartup		1	0
n pShutDown	Image: A start of the start	1	0

2. If using the "Project" way to create a new project (in this example, we use the "From template" way, refer the <u>Section 2.1.1</u> - Step 2), click the "Project" > "Settings..." from the menu bar to open the "Project settings" window. Click the "General" option and set the "Complex variables in a separate segment" to "Yes" to allow the using of complex data structures, such as arrays. Finally, click "OK" to exit the window.

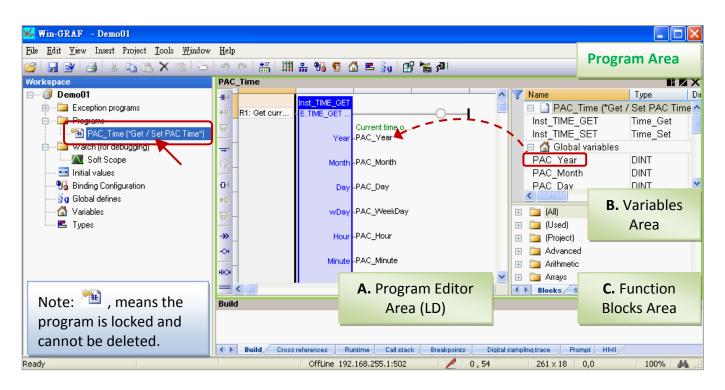
Proje	ect <u>T</u> ools <u>W</u> indo	w <u>H</u> elp		
ß	Build All Projects	F7		
	<u>C</u> lean All Projects			
	Download All Proj	ects		
	Settings			
Proje	ect settings			
C:\Wi	n-GRAF\Projects\De	mo01		
Gene		Name		Value
Runt		🔊 Communication param	neters	192.168.255.1:502
Debu	Jgging	S Cycle time		
Adva (All)	anced	Code Generation		Release
ų, m.		[i]. Complex variables in a	a separate segment	Yes
		🔁 On Line Change		Disabled
		Canal Version		V6 - 2014/02/25 12:17
		🗊 Libraries		Edit
		🧊 Use external objects		Edit
		Enables complex data such time consuming.	as arrays of structures. This o	Cancel

(Note: The "Code Generation" must be set to "Release".)

## 2.2 Introduction of the Project

### 2.2.1 Demo01 - LD Program

This program is used to read/write the Win-GRAF PAC's system time. In the Workspace, double-click the LD program name (i.e., "PAC\_Time") to open all relevant windows. As the screenshot below, the Program Area has three main parts:



**Tips:** Mouse click the Program Editor Area, and press the "+" or "-" key to zoom in or zoom out the content.

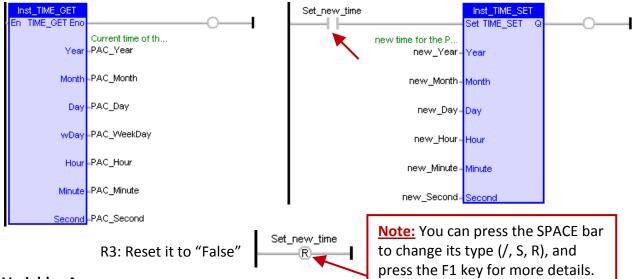
R1: Get current	Inst_TIME_GET En TIME_GET E Year	Current time of th -PAC_Year	R1: Get c		Current tim PAC_Year PAC_Month
	Month	-PAC_Month (E.g., press "-'			-PAC_Day -PAC_Week
	Day	ZOOM OUT the	content.)		-PAC_Hour
	wDay	-PAC_WeekDay			-PAC_Minute
	Hour	-PAC_Hour	R2: Set T	Set_new_ti	TIME_SET
	Minute	-PAC_Minute			new time f new_Year= Year

### A. Program Editor Area (LD):

This area allows to edit or display this LD program, you can click the object button (on the left of Program Area) to add a program, and then drag-and-drop variables (in the Variables Area) onto the function block one-by-one.

#### R1: Get current time of the PAC

R2: Set "Set\_new\_time" to "True" to set the new time

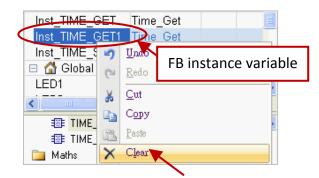


#### **B. Variables Area:**

This area shows the function blocks and variables that used in this program. Mouse double-click the "Name" or "Type" item to modify its name or data type, then click "Enter" to complete the setting. (Refer the <u>Section 2.3.1</u> for the details about variable declaration.)

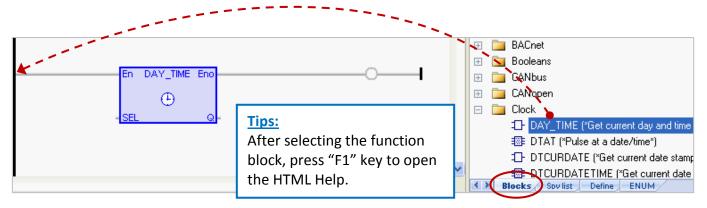
Name	Туре	Dim.	Attrib.
Inst_TIME_GET	<del>Get / Set PAC</del> Time_Get Time_Set	, Time'	") <mark>^</mark>
🗉 🚮 Global variat			
PAC_Year PAC_Month PAC_Day PAC_WeekDay PAC_Hour PAC_Minute PAC_Second	DINT D BOOL D BYTE D DINT D UNORD D INT D LINT D LREAL LWORD		

**Note:** For the function block can work correctly, the "Inst\_xxx.." FB instance variable will be automatically added when using one function block. For the safety reasons, this FB instance variable will not be automatically deleted even if the function block has been removed in the editor area. So, users can right-click the unwanted variable and select "Clear" to manually remove it.



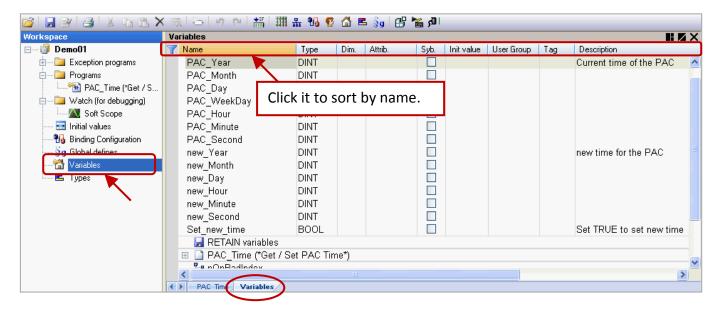
#### C. Function Blocks Area:

In the "Blocks" tab, it provides many types of the function block for users to drag and drop them to the editor area.



### 2.2.2 Demo01 - Variables

In the workspace, mouse double-click the "Variables" item to open the Variables window. The following screenshot shows all the needed and defined variables in this "Demo01" project.



**Tips:** In the Variables window, users can click any title field (e.g., "Name") for sorting. If you want to go back to the original sort order, mouse right-click anywhere and select the "Cancel Sorting" option.



**Field description:** (Press the "F1" key to look up the details) Mouse double-click any field item to set or modify the data.

Name:	A valid variable name starts with a letter (e.g., "A to Z" or "a to z") followed by any number of letters, numbers (e.g., "0 to 9"), or an underscore (i.e., "").
Type:	Data type. (Refer the <u>Appendix A</u> for the value range)
Dim.:	To specify the range of an array.
	(E.g., enter "10", means the use of the Counter [ <b>0</b> ] to [ <b>9</b> ]).
Attrib.:	Double-click this field item to set it to "Read Olny" that means users can only read this
	variable but cannot modify it.
Syb.:	If checked, the variable name will also be downloaded into the PAC.
Init value:	To set the initial value of the variable.
User Group:	All the variables can be divided into some groups (e.g., "Group1", "Group2") and it is
	convenient for users to look up or search these variables.
Tag:	To enter a nickname for the variable.
Description:	To enter a simple note for the variable.

## 2.3 Give it a Try

As mentioned before, we have described the LD program (<u>Section 2.2.1</u>) and variables (<u>Section 2.2.2</u>) in the "Demo01" project. The following sections will show you how to declare variables and add an LD program with the blinking function in this project.

Note: All the Win-GRAF PAC does not support the "ULINT" and "LWORD" data type.

### 2.3.1 Declaring the Win-GRAF Project Variables

First, we will declare two boolean variables (i.e., "LED1" and "LED2") that used in the program.

1. In the "Variables" window, mouse right-click any item in the "Name" field and select "Add Variable"

(or press the "Ins" key or click the 🔛 tool button) to add a variable.

🔁 🤊 (° 🕌		# 😘	🖸 🖸	🛋 🖇 g	ß	ا 🖪 🚆				
Variables										🛛 🖬 🗖 🗙
🝸 Name		Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
😑 🚮 Global varia	hles									~
PAC_Year	5	<u>U</u> ndo							Current time of the PAC	
PAC_Month	0	<u>R</u> edo								
PAC_Day	v	Cast								
PAC_WeekDay	*	<u>C</u> ut								
PAC_Hour		С <u>о</u> ру								
PAC_Minute		<u>P</u> aste								
PAC_Second	×	Clear								
new_Year	2	<u>E</u> dit							new time for the PAC	
new_Month										
new_Day		Cancel So	orting							
new_Hour		Enable C	hanges	Sp	ace					
new_Minute	_	Swap Glo	hal as R	atain						
new_Second	-									
Set_new_time	ð	A <u>d</u> d Vari	able 📉		Ins				Set TRUE to set new time	
🛃 RETAIN var		Add <u>M</u> uli	i Variabl	est						

2. Double-click the new "NewVar" item and change its name to "LED1", then click "Enter" to finish the setting. In this case, the data type is "BOOL".

Set_new_time	BOOL		Set TRUE to set new time
NowA/or	BOOL (		
LED1			

**<u>Note:</u>** The settings will be done only after clicking the "Enter" key.

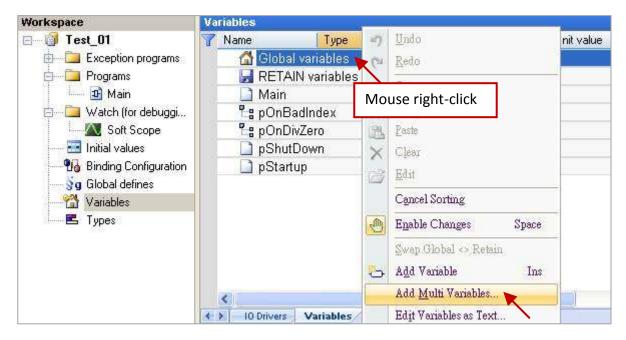
3. Follow the previous steps to add the "LED2" boolean variable.

Set new time	BOOL	Set TRUE to set new time
LED1	BOOL	
LED2	BOOL	
🛃 RETAIN varia	ables	

**Tips:** To set up multiple ordinal variables, enter the name "LED" (as the step2) and then press "Ctrl+C" and "Ctrl+V" twice to create "LED1" and "LED2" (auto sequential numbering), finally, delete the first variable (i.e., "LED").

### <u>Tip #2:</u>

1. If you need to add multiple variables (e.g., "Boo\_01 to Boo\_16"), simply right-click the "Global variables" and select the "Add Multi Variables".



2. Follow the settings like the figure below (Name: "Bool\_%%"; Type: "BOOL"; From: 1; To: 16) to create Boolean variables (i.e., "Bool\_01" to "Boo\_16") and then click "Create all" button to complete the settings.

Enter inf	ormations						1					
Vame	Bool_%%							eate <u>a</u> ll				
Гуре	BOOL				1		<u>(</u>	Cancel				
aroup	Global variables					~		<u>H</u> elp				
	Read only	Dir	m.	0		[]						
rom	1		To	16								
Creation	of 16 variables : From variabl	Variables Variables	s Global v	Type ariables	6 Dim.	Attrib.	Syb.	Init value	User	Tag	Description	1
Creation		e Bool_01 ti	o variat	ole Bool_1	6							
Creation		Variables	5		9.0 <del>4</del>	611-7	Cut	luiturelur		T	Description	H
Creation		Variables	5		9.0 <del>4</del>							
Creation		Variables	5		9.0 <del>4</del>	Attrib	Sub	Init value	User	Tag	Description	
Creation		Variables Variables	s Global v	Type ariables	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	<b>H</b>
Creation		Variables Variables Name Boo	s   Global v  _01	Type ariables BOOL	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Name Boo Boo	s   Global v  _01	Type ariables	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Name Boo Boo Boo Boo Boo	s Global v I_01 I_02 I_03 I_04	Type ariables BOOL BOOL BOOL BOOL BOOL	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Name Carlow None	Global v I_01 I_02 I_03 I_04 I_05	Type ariables BOOL BOOL BOOL BOOL BOOL	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Name Boo Boo Boo Boo Boo Boo Boo Boo	Global v I_01 I_02 I_03 I_04 I_05 I_06	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Name Carlor Boo Boo Boo Boo Boo Boo Boo B	Global v I_01 I_02 I_03 I_04 I_05 I_06 I_06 I_07	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Name Construction Boo Boo Boo Boo Boo Boo Boo B	Global v L_01 L_02 L_03 L_04 L_05 L_06 L_07 L_08	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOO	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Boo Boo Boo Boo Boo Boo Boo Boo Boo Bo	Global v L_01 L_02 L_03 L_04 L_05 L_06 L_07 L_08	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOO	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Name Construction Name	s Global v I_01 I_02 I_03 I_04 I_05 I_06 I_06 I_07 I_08 I_09 I_09 I_10 I_10 I_11	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOO	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Boo Boo Boo Boo Boo Boo Boo Boo Boo Bo	s Global v I_01 I_02 I_03 I_04 I_05 I_06 I_07 I_08 I_09 I_10 I_11 I_11 I_12	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOO	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	
Creation		Variables Variables Boo Boo Boo Boo Boo Boo Boo Boo Boo Bo	Global v 1_01 1_02 1_03 1_04 1_05 1_06 1_06 1_07 1_08 1_09 1_10 1_11 1_11 1_12 1_12 1_13	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOO	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	1
Creation		Variables Variables Boo Boo Boo Boo Boo Boo Boo Boo Boo Bo	s Global v 1_01 1_02 1_03 1_04 1_05 1_06 1_07 1_08 1_09 1_10 1_11 1_11 1_12 1_13 1_14	Type ariables BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOO	9.0 <del>4</del>	Attrib.		Init value	User	Tag	Description	

### 2.3.2 Declaring the I/O Variables

In this example, the I-8055W module that used to show the blinking feature must be plugged in the PAC's slot1. So, we need to add an I/O link to correspond to the real I/O module.

1. Click the "Open I/Os" tool button to add an I/O link.



2. Mouse double-click on "Slot 1" and then double-click the "i\_8055" to select this I/O board.

III I/O Boan	rds 🔰	3
	Close Select	
4 5 6 7 8 9 10 11 12 13 14	Library:         Device:           [all]         i.8046_DI (*I-80546 W (16-ch D/I) *)           ICP DAS - XP-WP-VF         i.8050 (*I-8050W (16-ch configurable DI0 i.8051_DI (*I-8051 W (16-ch D/I) *)           Standard         i.8052_DI (*I-8052 W (8-ch D/I) *)           i.8053_DI (*I-8053 W (16-ch D/I) *)         i.8053_DI (*I-8053 W (16-ch D/I) *)           i.8055_C (*I-8055 W (8-ch D/I + 8-ch D/O) *         i.8056_DO (*I-8055 W (8-ch D/I + 8-ch D/O) *           i.8056_DO (*I-8055 W (8-ch D/I + 8-ch D/O) *         i.8056_DO (*I-8057 W (16-ch D/O) *)           i.8058_DI (*I-8057 W (16-ch D/O) *)         i.8058_DI (*I-8057 W (16-ch D/O) *)           i.8056_DO (*I-8057 W (6-ch D/I) *)         i.8058_DI (*I-8057 W (16-ch D/O) *)	0) *)
15 16	I-8055W 8-Ch Digital Inputs + 8-Ch Digital Outputs Nov.01,2013 ICP DAS , Taiwan	

3. Click the "Close" button to exit the "I/O Boards" window.

<u>Note:</u> Click the "Virtual/Real" button to change to the Virtual I/O (for testing) or the Real I/O. (The Real I/O is used in this example).

IIII I/O Boards	×
0 1 <u>i 8055</u> 0 i_8055_D1 1 i_8055_D0 2 3	Close Select
Virtual I/O: (It shows in blue)	Rename           Properties
1 i_8055 0 (i_8055_DI) 1 (i_8055_DO) 2	Virtual/Real

After linking the "i\_8055" I/O board, it will automatically add 8 input & output variables in the "Variables" window.

Variables									
🝸 Name	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
LED1	BOOL								~
LED2	BOOL								
🚽 RETAIN variab	les								
📃 📓 %IX1.0 - i_805	5_DI								
%IX1.0.0	BOOL		Input						
%IX1.0.1	BOOL		Input						
%IX1.0.2	BOOL		Input						
%IX1.0.3	BOOL		Input						
%IX1.0.4	BOOL		Input						
%IX1.0.5	BOOL		Input						
%IX1.0.6	BOOL		Input						
%IX1.0.7	BOOL		Input						
😑 🐸 %QX1.1 - i_80	65_DO								=
%QX1.1.0	BOOL		Output						
%QX1.1.1	BOOL		Output						
%QX1.1.2	BOOL		Output						
%QX1.1.3	BOOL		Output						
%QX1.1.4	BOOL		Output						
%QX1.1.5	BOOL		Output						
%QX1.1.6	BOOL		Output						
%QX1.1.7	BOOL		Output						~
<						)			>
PAC Time Variable	es								

## %<mark>|</mark>X1.0 - i\_8055\_DI

"I" means "Input" "X" means "Boolean" "1" means "Slot 1"

## %ID or %QD

"D" means "Integer/Real"

## %QX1.1 - i\_8055\_DO

"Q" means "Output" "X" means "Boolean" "1" means "Slot 1"

There are three output variables are used in this example, and you can modify the name for easy use. Mouse double-click the item and fill in a name, then press "Enter" key to finish the setting.

🖃 📕 %QX1.1 - i	_8055_DO			🖃 📕 %QX1.1 -	i_8055_DO	
%.OV1.1.0	BOOL	Output		%QX1.1.0=Out	put1 BOOL	Output
Output1		Output		%QX1.1.1=Out	put2 BOOL	Output
%QX1.1.2	BOOL	Output	$\neg$	%QX1.1.2=Out	put3 BOOL	Output
%QX1.1.3	BOOL	Output		%QX1.1.3	BOOL	Output

### 2.3.3 Creating an LD Program

In the "Demo01" project, we want to create a "LD1" program to show the blinking. To begin, follow these steps:

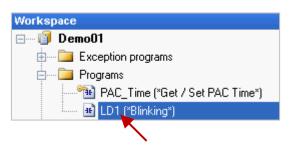
1. In the workspace, mouse right-click the "Programs" folder and select "Insert New Program...".



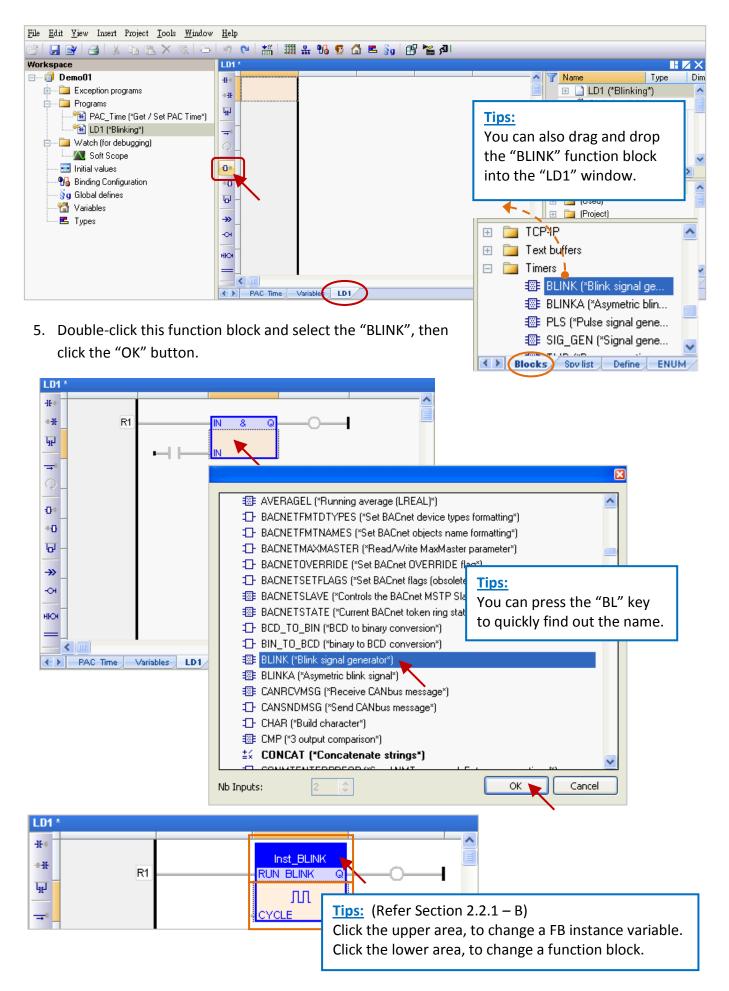
2. Fill in a program name in the "Name" field and enter a simple note in the "Description" field, and then select the "LD – Ladder Diagram" as the programming language and click the "OK" button.

New program			×
Properties Advance	d Description		
Program			
Name:	LD1		
Description:	Blinking		
Programming lan;	guage	<b></b>	
SFC - Sequentia SFC - Sequentia	al Function Chart - Grid edit al Function Chart - Free fon 1 Block Diagram		
LD - Ladder Di ST - Structured IL - Instruction PACKML - PA	Text		
Execution style	m		
🔘 Sub-program			
	Defined Function Block)		
Child SFC p	rogram		
Child of:		*	
	確定	取消 説明	

3. Double-click the "LD1" program to open the editor window.



4. Click the "Insert FB.." button on the left of the "LD1" window to add a function block.

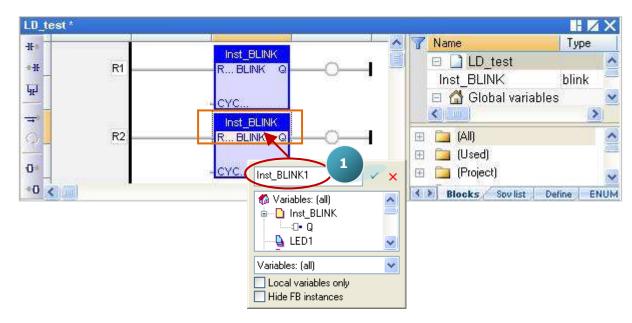




### **Important Notice:**

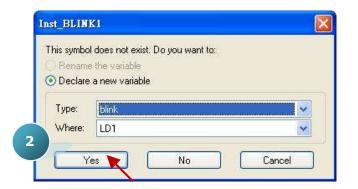
When programming, users may copy & paste an existing function block to create a new one. But, this way will cause a function exception due to the same function instances. Therefore, users must create a new name for the function instance.

1. Mouse double-click the function block and enter a new name (e.g., "Inst\_BLINK1"), then click the



button to complete the setting.

2. In the "Inst\_BLINK1" window, click "Yes" to create this function instance.



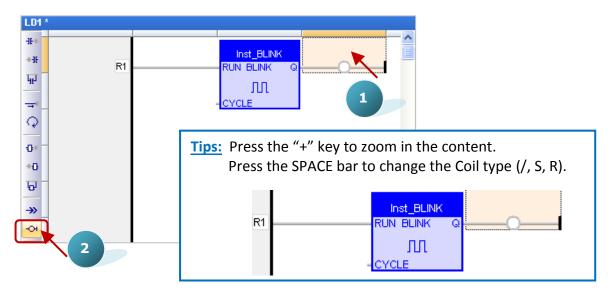
### Tips:

You can also use the same way to double-click on the right-side of the "Coil" to create or assign a variable. (See Step7)

3. Now, there is an "Inst\_BLINK1" function instance added in the Variables Area.

-16-			<u> </u>	Name	Туре
- 36	R1	R., BLINK Q		🗉 🗋 LD_test	· · · · · · · · · ·
200	avi –	Pear DLINN: 35		Inst BLINK	blink
ц <sub>Е</sub> J		-CYC	3	Inst_BLINK1	blink 💊
-		Inst_BLINK1			۰ ک
Q	R2	R., BLINK G	0	🚞 (All)	^
0		010		🛅 (Used)	
	3	-CYC	× .	(Project)	~

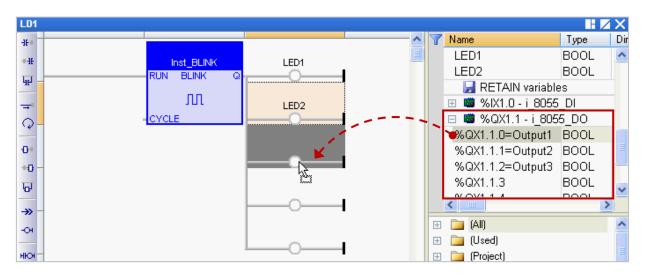
6. Click the "Coil" on the right of the "BLINK" function block, and continuously click the "Insert Coil" button to add four "Coil".



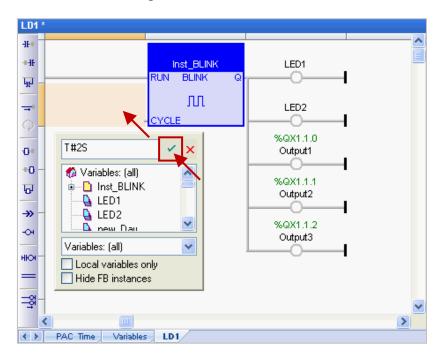
7. Mouse double-click the first "Coil" and double-click "LED1" to assign it. Follow the same way to assign the "LED2" variable to the second "Coil".

LD1 *		
-IF-#		Name
<del>- 1</del> F	R1	RUN BLINK Q
<b>h</b> el		ДД 💦 🚷 Variables: (all) 🔼
<u></u>		CYCLE
Q		
0+		
+0		ew_Hour
ਹ ਹ		O new_Minute
		□ Q new Second
→» _		Variables: (all)
-04		Local variables only
ню-		Hide FB instances

8. As the screenshot below, mouse drag-and-drop the "Output1", "Output2" and "Output3" variables to the 3th, 4th and 5th "Coil".



9. Mouse double-click on the left of the "CYCLE" and enter "T#2S" (to blink every two seconds), and then click is to finish the setting.



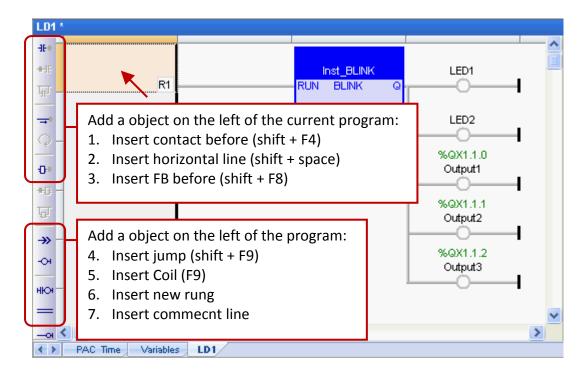
10. Finally, click the "Save" button to save the "LD1" program.

🐝 Win-GRAF - Demoûl								
<u>File E</u> dit <u>V</u> iew Insert Project <u>T</u> ools <u>W</u> indow <u>H</u> elp								
C 🕞 🕞 🖉 🖓 🕒 🔊 🗠 👘 🛗 🗰 🏭 👭 😨 🖓 🗉 🚱 🕼 📽 🖓								
Workspace	LD1							
⊡ 🗊 Demo01	-11-=				🍸 Name	Туре	Dim.	
🗄 🚞 Exception programs	+-16	Inst BLINK	LED1		LED2	BOOL	~	
🚊 🛅 Programs					🛃 RETAIN variables			
**** PAC_Time (*Get / Set	ц.		· · ·		🗉 🗉 🕲 %IX1.0 - i_8	055_DI		
🛄 LD1 (*Blinking*)	-		LED2		🖃 國 %QX1.1 - i	8055_DO		
🚊 🚞 Watch (for debugging)		T#2S CYCLE	——O——		%QX1.1.0=Outp	it1 BOOL		
Soft Scope			%QX1.1.0		%QX1.1.1=Outpu	it2 BOOL	_	
🔤 Initial values	0.		Output1		%QX1.1.2=Outpu		~	
	+0 -		——O——		<		>	
🚽 🖇 🚽 🚽 🚽 🚽	ធ		%QX1.1.1		🗉 📴 (All)			
🖓 Variables	U		Output2		🖭 🛄 (Used)			
E Types					E Dia (Project)		_	
	-04		%QX1.1.2 Output3		<ul> <li>Advanced</li> </ul>			
					Advanced     Arithmetic		_	
	HICH -		<u> </u>	~	<ul> <li>Anniheac</li> <li>Arrays</li> </ul>			
	-	m		>	Blocks Soviist	Define ENU	IM-	
	PAC Time Variables     LD1							
	Build						×	
	< ≻ Buil	d Cross references Runtime Call st	ack Breakpoints Di	iqital samp	pling trace Prompt HM	41		
Ready OffLine 192.168.255.1:502 🞽 1,1 1 x 1 1,1								

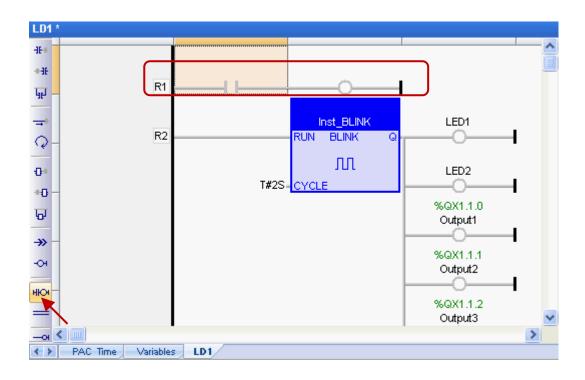
Note: """ means this program is opened (locked, cannot be deleted). Click the "X" in upper-right corner of window to close this program (un-locked, " ").

If you want to add a program to the first line after completing the programming, follow these steps:

 Click the upper-left corner of the Program Editor Area, and you will see the selectable items on the Object bar. Select an object (4 to 7, as the screenshot below) to add a program to the first line. <u>Note</u>: You can also select an object (1 to 3) to add it on the left of the current program.



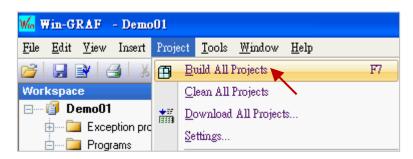
2. In this case, click "Insert new rung" to add it to the first line.



#### 2.3.4 **Compiling the Program**

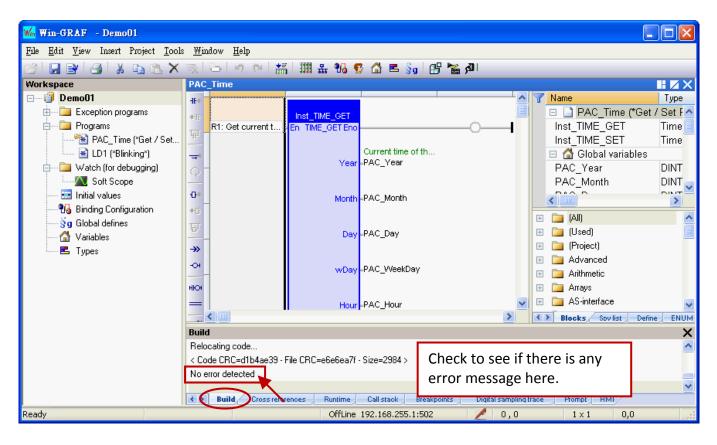
In the previous section, we have added and saved the LD program. For the Win-GRAF project can function properly on the PAC, we need to compile the programs. To begin, follow these steps:

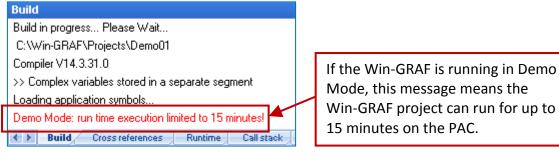
1. On the menu bar, click "Project > Build All Projects" to compile all programs.



2. If a "No error detected" message is appear that means the project was successfully compiled. Note: If you modify and save the program after compiling it, click the "Clean All Projects" to clean

the previous results and then do the step1 again.

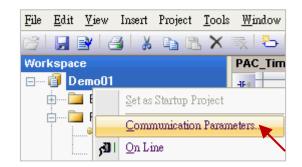




## 2.3.5 Download the Program to PAC

Before downloading the program, you need to set up the communication parameters. (By now, it only supports the Ethernet TCP/IP).

1. Mouse right-click the project name (i.e., "Demo01") and select "Communication Parameters..." to open the settings window.



2. Enter the "PAC IP:502" (e.g., "192.168.255.1:502") to add an IP address and then click "OK".

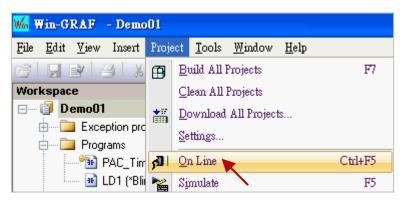
It can also click the 🛄 button to add/modify the IP address.

(Note: the default PAC IP is "192.168.255.1" and the fixed port number of Win-GRAF PAC is "502")

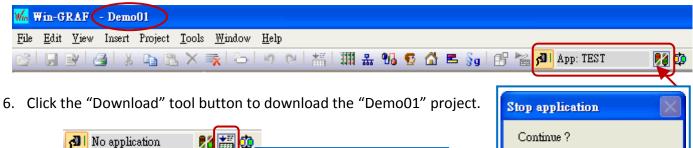
Communication Settings		
T5 Runtime 192.168.255.1:502 192.168.255.1:502 Tips: All the configured IP will can select the unwanted IP a delete it (e.g., "192.168.78.8 192.168.78.8:502	nd press "Del" key to	How to Extend the Timeout ? If typing "PAC IP:502", the default timeout is 2 seconds an
192.168.255.1:502 192.168.71.19:502 192.168.78.8:502	Port <u>n</u> umber: E	192.168.255.1         502         Lelp         COM15         19200         None         1

3. Before establishing a connection, make sure the PAC and the network are working properly.

4. Click the menu bar "Project" and select "On Line", or click the 🗖 tool button to establish a connection.



5. As the screenshot, if it shows "App: TEST", different to the current project name (i.e., "Demo01"), that means there is a project (name: "TEST") running on the PAC. Click "Stop application" tool button to stop the "TEST" project.





7. If RUN is displayed, it means that the "Demo 01" project has been successfully executed on the PAC.

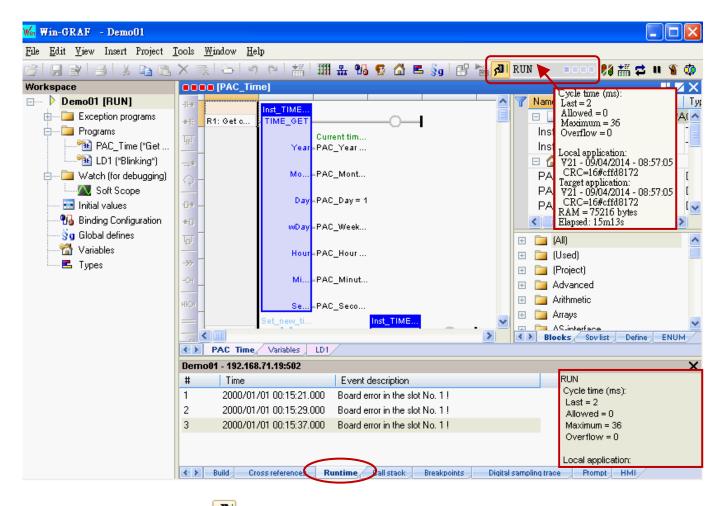
63   B B   B   X 🖪 🖻 X	家 谷 9 9 1		🖸 🖾 🛋 🛐 🖽	RUN 🔂	0000 🕅 🛗 u	<b>1</b>
Workspace	PAC_Time]					
Demo01 [RUN]	-It-s	1		^	Name	Value
Exception programs		Inst_TIME_GET			🛛 🗉 🗋 PAC_Tim	e (*Get / Set F📥
🚊 🗝 Programs	R1: Get current t				Inst_TIME_GE	Т
PAC_Time (*Get / Set	<u>F</u>				Inst_TIME_SE	г
🔤 LD1 (*Blinking*)			Current time of th		🗉 🚮 Global var	iables
🚊 🚞 Watch (for debugging)	0-	Year	_PAC_Year = 2014		PAC_Year	2014
Soft Scope					PAC_Month	4
📰 Initial values	0.0	Month	_PAC_Month = 4		PAC_Day	7
	+0		-		PAC_WeekDa	y 1
🔤 😽 🚽 Global defines	<b>U</b>				PAC_Hour	13
🔤 📸 Variables		Dav	av _PAC_Day = 7		PAC_Minute	55
Types	<i>→</i> >		_	ľ	PAC Second	23 🗸 🗸
	-Ол	wDay	_PAC_WeekDay = 1	The evenent		>
			···· <u>_</u> ·····, ·	The current	time on	~
	HKOI			the PAC is d	lisplayed.	
		Hour	PAC_Hour = 13		· · ·	
					🕀 🧰 Advanced	
		Mirou da	PAC_Minute = 55		🕀 🧰 Arithmetic	
	-	IVIII IGUU			🗄 🧰 Arrays	
					🕀 🧰 AS-interface	
		Second	PAC_Second = 23	~	🗄 🛄 BACnet	~
	<	•		>	Blocks Sov list	Define ENUM
	PAC Time Vari	iables				

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 2-20

<u>Note:</u> If there is any error message show up during the download process, refer the <u>Appendix B</u> to get the solution.

#### Cycle time

When on-line with the PAC, move your mouse over the "RUN" position on the toolbar to view the current cycle time of the program on the PAC. You can also view the cycle time in the bottom-right corner of the message area.



When doing the "On Line" () operation, it will automatically switch to the "Runtime" tab and you can see if there is any error message for the downloaded program. (E.g., in this example, we need to plug the I-8055W module in the Slot1 of the PAC, and the message "Board error in the slot No. 1 !" means there is no I/O module in the Slot1 or an I/O exception.)

**Shutting down the PAC,** and plug one I-8055W module in the PAC's Slot1 then reboots. Then, click "On Line" (<sup>1</sup>) button to connect to the PAC.

Demo01 - 192.168.71.19:502					
#	Time	Event description	RUN Cycle time (ms): Last = 2 Allowed = 0 Maximum = 36 Overflow = 0		
		$\frown$	Local application:		
< >	Build Cross refer	ences 🤇 Runtime 🖉 Call stack 🚽 Breakpoints 🚽 Digital sa	mpling trace Prompt HMI		

## 2.3.6 Testing the Program

In the previous section, you have successfully downloaded the "Demo01" project and the following will describe how to test the program.

### The "PAC\_Time" Program:

1. Mouse double-click the variable name (e.g., "new\_Year") in the "TIME\_SET" function block (or the Variables Area) one-by-one to change the PAC's system time (e.g., to change it as January 1, 2015 12:30:35).

	e [PAC_Time]						ZΧ
-16-4	Set TRU	JE to set n		• 7	Name	Value	
+-IE		v_time = F			PAC_Month	4	^
ज़-	R2: Set TRUE to	( <b> </b>	Set TIME_SET Q		PAC_Day	7	
					PAC_WeekDay	1	
-→ <sup>‡</sup>		new time for the . new_Year =			PAC_Hour	14	
Q		new_rear =	0 Year		PAC_Minute	16	
_		new_Year			PAC_Second	41	=
-0++		2014			new_Year	0	
+0		2014			new_Month	0	
চ					new_Day	0	
	-		<u>F</u> orce		new_Hour	0	
->>>			L = =l:		new_Minute	0	~
-O1			Lock				>
ню			Unlock		🚞 (All)		~
					🚞 (Used)		
-	-	15	8 7	0	🚞 (Project)		_
극함					🚞 Advanced		
		31	24 23	16	🚞 Arithmetic		
	Set TRU	IE to set n			🚞 Arrays		
		v_time = F			🚞 AS-interface		
-	R3	-R		<b>~</b>	🚞 BACnet		~
			>	< >	Blocks Sov list	Define	ENUM
40	PAC Time Variables						

2. Set the "Set\_new\_time" variable to "TRUE" to write the new system time.

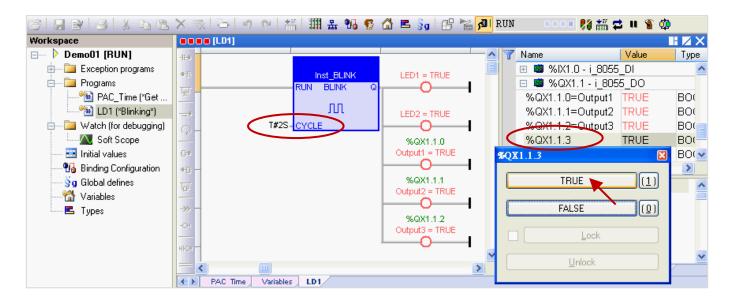
BEEE [PAC	_Time]							$\mathbb{Z}$	K
-IF-#	C-4 TD	UE to set n		~	7	Name	Value		I
+-1E		w_time = F	Inst_TIME_SET			PAC_WeekDay	1	1	~
R2: Set 1	RUE to		Set TIME_SET	a		PAC_Hour	14		
Ψ.						PAC_Minute	30		
		new time for the				PAC_Second	37		
Q-		new_Year = 20	015 Year			new_Year	2015		
~ <i>e</i>						new_Month	1		
-O+P		new_Month	= 1 - Month			new_Day	1		
+0:						new_Hour	12	- 11	
ਹ						new_Minute	30		-
		new_Day	=1-Day	E		new Second	35		
<i>→</i> >				_		Set_new_time	FALSE		~
-он		new_Hour =	12 Hour			<	_	>	7
		new_near =		et_new	tin	e	×		
HIOI									
_		new_Minute =	30-Minute			TRUE	(1)		2
		new_Second =	25 0			FALSE	(0)		
	Cot TD	UE to set n	33-Second						
		w time = F				Lock			
	R3	-R							
<						<u>U</u> nlock		ENU	M
	ime Variables								Ì

3. Then, the new system time will show on the "TIME\_**GET**" function block (or the Variables Area), and the "Set\_new\_time" variable will be reset to "FALSE" automatically.

[PAC_Time]				II 🛛 X
-lt=		<u> </u>	7 Name	Value
	GET		new_Year	2015 🔼
R1: Get current t. JED JUME C	El Eno	I	new_Month	1
E I			new_Day	1
	Current time of th		new_Hour	12
QT	Year_PAC_Year = 2015		new_Minute	30
			new_Second	35
0.0	Month PAC_Month = 1		Set_new_time	FALSE_>
•0			LED1	FALSE 🗉
TET I I I I I I I I I I I I I I I I I I	5 500 Bur 4		LED2	FALSE
	Day_PAC_Day = 1		AIN var	
->>		The new syste	em time. 🛛 🕺 1.0 - i_8	3055_DI 🛛 🔽
-O1	wDay_PAC_WeekDay = 4			>
HIOT			🕀 🚞 (All)	~
			🕀 🚞 (Used)	
	Hour PAC_Hour = 12		🗄 🚞 (Project)	
			🗄 🚞 Advanced	
	Minute = 30		🗄 🚞 Arithmetic	
			🗄 🛅 Arrays	
			🗄 🚞 AS-interface	
	Second PAC_Second = 35	~	🖽 🚞 BACnet	~
		<u>&gt;</u>	Blocks Sov list	Define ENUM
PAC Time Variables				

#### The "LD1" program:

- 4. When the "Demo01" project is running, you can check to see if the DO0, DO1 and DO2 tags of the I-8055W I/O module that plugged in the slot1 of the PAC is blinking every two seconds (like the value "T#2S" we set before). You can also assign a "TIME" variable on the left side of the "CYCLE" for easy to change the time setting. Refer the <u>Section 2.3.1</u> for the setting way.
- 5. If the "%QX1.1.3" variable is set to "TURE" in the Variable Area, the LED4 (i.e., "DO3", at the top of the I-8055W I/O module) will light up.



6. Click the *integral* tool button again to cancel the PAC connection.
 <u>Note:</u> Do NOT click the *integral* Stop Application button; it will stop the running project on the PAC.

# Chapter 3 Modbus Slave: Allow the SCADA/HMI Software to Access Win-GRAF Variables

In Chapter2, we have described how to get/set the PAC system time (i.e., the "PAC\_Time" program) and create a blinking function (i.e., the "LD1" program) in the "Demo01" project. The following sections describe how to allow the SCADA/HMI software (e.g., "InduSoft") to access Win-GRAF variables that defined in the "Demo01" project. The Win-GRAF Workbench provides two ways to open the PAC data, one way is to enable the Win-GRAF PAC as a Modbus **TCP** Slave and the second way is to enable the Win-GRAF PAC as a Modbus **TCP** Slave and the second way is to enable the 3.1, and then refer the <u>Section 3.2</u>). To begin, follow these steps:

# 3.1 To Enable the Win-GRAF PAC as a Modbus TCP Slave

₩a Win-GRAF - Demo01			
<u>F</u> ile <u>E</u> dit <u>V</u> iew Insert Pro	ject <u>T</u> ools <u>W</u> indow <u>H</u> elp		
😂 🔒 🖹 🎒 🕺 🗈	鸣 🖆 🚯 💀 🕹 😵 🚛 腓 🕌 🔊 🤊 🖆 🎽 🔊		
Workspace	IO Drivers *		
⊡… 🗊 Demo01	Rame Value	🝸 Name	Туре
🗄 📴 Exception prog		🗉 🚮 Global va	riables 🔺
🖻 🔤 Programs		PAC_Year	DINT 🧮
PAC_Time	*E	PAC_Month	DINT
LD1 (*Blink		PAC_Day	DINT _
🖃 🔤 Watch (for de		DACINGAL	
Soft Scope			>
🔤 Initial values		Name	/alue
📲 🚮 Binding Config	ي الم		
§g Global defines			
	]↓		
L Types	·		
		<	>
	PAC Time Variables LD1 10 Drivers	II - J	
	Build		×
	C:\Win-GRAF\Projects\Demo01\		~
	No error detected		
	Build Cross references Runtime Call stack Breakpoints Digital sampling trace	Prompt HMI	
Ready	OffLine 192.168.255.1:502 🌽 0, (	0 × 0	0,0 ,;;

1. Click the "Open Fieldbus Configuration" tool button to open the "IO Drivers" window.

2. Click the "Insert Configuration" button on the left side of the "IO Drivers" window and then select the "MODBUS Slave" and click "OK" to enable a Modbus TCP Slave.

10 Drivers '				
<b>眉</b> 品		Name	Valu	e
品	Add Configuration			×
18 8-	Choose a configuration			ОКЪ
				Cancel
	MODBUS Master			
ġį́s	MODBUS Slave			
<u> </u> ]+				
PAC	Time Variables LD1 10 Drivers			

3. Click the "Insert Master/Port" button on the left side to set the "Slave number" (In this case, the value is "1"), and click the "OK" button.

10 Drivers *			
□ [日] ● M8 M0	ODBUS Slave	Name	Value
品			
*	MODBUS Slave Protocol	E	X
	Slave number: 1	OK Cancel	]
<pre></pre>			
Slave num	nber Server ID		
< > PAC Tir	me Variables LD1 10 Drivers		

4. Click the "Insert Slave/Data Block" button on the left side to open the "MODBUS Slave Request" window.

IO D	IO Drivers *						
臣	⊡… <b>Mg</b> MODBUS Slave				Name	Value	
쁆	5	📅 Server -	Slave number =	= 1	Slave number	1	
					Server ID		
	<ol> <li>Press the "F1" key to open the Help on this subject.</li> <li>If you add two or more "Server" settings, type a "Server ID" for easier use (a string, e.g., `SVR1')</li> </ol>						
				ltem	Description		
+							
<b>*</b> >	PAC Time Variables LD1 IO Drivers						

5. Enter a simple note in the "Description" field and then click the "Input Registers" option.

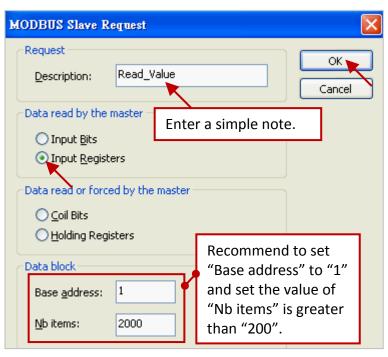
#### For Modbus Master to Read data:

Options	Data types
Input-bits	BOOL
Input Pogistors	BYTE, INT, DINT,
Input Registers	REAL, etc.

#### For Modbus Master to <u>Write</u> data:

Options	Data types				
Coil-bits	BOOL				
Holding	BYTE, INT, DINT,				
Registers	REAL, etc.				

(Refer the <u>Appendix A</u> to see more data type)



- 6. As the screenshot above, it's recommended to set the "Base address" to "1" and the "Nb items" refers to how much variable data can be provided by one "data block". If the data address requested from the Modbus Master (e.g., the SCADA software) is greater than this value (in this example, the value is "2000"), the Modbus Slave (i.e., Win-GRAF PAC) will not respond.
- 7. Mouse drags all the needed variables (e.g., "PAC\_xxx", data type: "DINT") one-by-one from the Variables area and then drop it to the "Symbol" field.

10 0	)rivers *								
眉	⊡Mg MODBUS Slave		Name	Value	7	Name	Туре	Dim.	
恭	i Server - Slave number = 1 i Input Registers [12000] - Read_Va		Request	Input Registers		PAC_Month	DINT	~	
			🗄*🛢 Input Registers [12000] - Read_Va		1		PAC Day	DINT	
*8			Nb Item	2000		PAC_WeekDay	DINT		
8			Descripti Read_Value				PAC_Hour	DINT	
	<b></b>						PAC_Minute	DINT	
	Symbol	Offset	Mask	S	itorage	F	PAC_Second	DINT	~
	PAC_Year	0	FFFF	De	efault		<		>
¢þ	PAC_Month	0	FFFF	De	efault	N	ame / Valu	le	
	PAC_Day	0	FFFF	De	efault			-	
∎+	PAC_Hour	0	FFFF	De	efault	-			
	PAC_Minute	0	FFFF	De	efault				
	PAC_Second	0	FFFF	De	efault				
	<	1111			>				
< >	PAC Time Va	riables LD1 IO D	rivers						

- 8. Mouse double-click the "Offset" field and fill in a value, then press "Enter" key to finish the setting.
  - Note: (1) The "Offset" value starts at "0" and the Modbus address of variable is equal to this value plus 1 (Base address).
    - (2) If using a 32-bits (or more than 32-bits) data type (e.g., "DINT", refer the <u>Appendix A</u>), it requires two Modbus addresses, as the table below, the "Offset" values are 0, 2, 4, 6, etc.

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

## Tips:

Mouse click the "Offset" field and press the keyboard "Ctrl+A" to select all items, and then click the "Iterate Property" button at the left-side to open the settings window.

	Symbol	Offs	et 🛆	Mask	Storage
	PAC_Year	0		FFFF	Default
ġ,þ	PAC_Month	0		FFFF	Default
∎∔	PAC_Day	0	<b>N</b>	FFFF	Default
	PAC_Hour	0		FFFF	Default
Ð	PAC_Minute	0	Ctrl+A	FFFF	Default
	RAC_Second	0	CurrA	FFFF	Default
	PAC_WeekDay	0		FFFF	Default

Keep the "Name" setting, enter "0" into "From" field and enter "2" into "By" field, then click "OK".

Name From:	0	%	By:	2	×	(I 4 ca tł
CResults	;				Symbol PAC_Y PAC_M	ear Ionth
8 10 12					PAC_D PAC_H PAC_M PAC_S	our linute
			ОК		Ca PAC_W	

(If the "Name" is set to "%%", it will show "00, 22, 44, 66, 88, 1010, 1212" in this example. The user can modify it depends on the needed settings and then check the value in the "Results" area.)

	Symbol	Offset	🔺 Mask	Storage
	PAC_Year	0	FFFF	Default
	PAC_Month	2	FFFF	Default
	PAC_Day	4	FFFF	Default
	PAC_Hour	6	FFFF	Default
	PAC_Minute	8	FFFF	Default
	PAC_Second	10	FFFF	Default
Ca	PAC_WeekDay	12	FFFF	Default
	<	Ш		>
	PAC Time Va	riables LD1	10 Drivers	

 Click "Storage" to select entire columns and then press "Enter" key to display a drop-down menu. Then, select "DWORD (Low – High)" and press "Enter" key to complete the setting. (If using a 16-bits or below, it's no need to set the "Storage" item.)

Symbol	Offset	Mask	S	itorage 💊	$\Delta_{-}$	Range (Low)
PAC_Year	0	FFFF	D	efault 🔨		
PAC_Month	2	FFFF	D	efault	Ť	
PAC_Day	4	FFFF	D	efault		
PAC_Hour	6	FFFF	D	efault		
PAC_Minute	8	FFFF	D	efault		
PAC_Second	10	FFFF	D	afault		
PAC_WeekDa	y 12	FFFF	D	Default DWORD (H DWORD (L	.ow - F	
PAC Time	Variables / LC	1 IO Drivers		STRING(6) STRING(8) STRING(10		

To expand this "Data Block" and you can see the Modbus addresses of all variables. It equals to the "Offset" value plus 1 (Base address).

101	Drivers *							
E	🚊 📲 Inp	out Registers [1.,2000]	-Rea	ad_Value	^	Name	Value	
뮮		+01: PAC_Year				Request	Input Reg	jisters
*	È 🗋	+23: PAC_Month				Address	1	
		+45: PAC_Day				Nb Item	2000	
ð	🗖	+67: PAC_Hour			≡	Description	Read_Va	lue
	🖸	+89: PAC_Minute						
	🖸	+1011: PAC_Seco	nd					
	<b>_</b>	+1213: PAC_Week	Day		4			
ŝį	Symbol	Offset	Ma	sk		Storage		Range (Lo
₽Ļ	PAC_Year	0	FFF	=		DWORD (Lo	w - High)	
8+	PAC_Month	2	FFF	=		DWORD (Lo	w - High)	
+	PAC_Day	4	FFF	=		DWORD (Lo	w - High)	
	PAC_Hour	6	FFF	=		DWORD (Lo	w - High)	
	PAC_Minute	8	FFF	-		DWORD (Lo	w - High)	
	PAC_Second	10	FFF	-		DWORD (Lo	w - High)	
	PAC_WeekDay	12	FFF	=		DWORD (Lo	w - High)	
	<							>

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 3-4

- 10. Then, we need to add the second "Data Block" for the Modbus Master to read the Boolean data. This configure way is similar to the step 4 to 8:
  - (1) Click the "Server ..." item and click the "Insert Slave/Data Block" button at the left side to open the settings window.
  - (2) In the "MODBUS Slave Request" window, enter a simple note and select the "Input-bits" option, then set "Base address" to "1" and set "Nb items" to "2000".

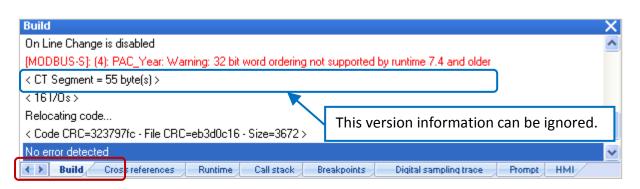
	IO Drivers *		
	B ModBUS Slave		
	品 ···· 육 Server - Slave numbe	er = 1	
	💼 🗄 🖓 🗄 Inpot Registers [1	2000] - Read_Value	
		MODBUS Slave Reques	st 🔀
		Request	
	1	Description: Read	d_Boolean
			Cancel
		Data read by the maste	Enter a simple note
		Input Bits	
		O Input <u>R</u> egisters	
For the Wodbus W	aster to <u>Read</u> data:	⊂Data read or forced by t	the master
Option	Data Type		
Input Bits	BOOL	O <u>⊂</u> oil Bits	
Input Dogistors	BYTE, INT, DWORD,	O <u>H</u> olding Registers	
Input Registers	REAL, LINT, etc.	Data block	
(Refer <u>Appendix A</u> to	o see more data type)	Base <u>a</u> ddress: 1	
		Nb items: 2000	)

(3) Mouse drags the Boolean variables (i.e., "LED1", "LED2"; data type: BOOL) one-by-one and drop them to the "Symbol" area, and then set the "Offset" to "0" and to "1".

10 0	rivers *											X
冒	🗄 朂 Server	- Slave number = 1		^	Name	Value		Y	Name		Туре	
뮮	🗄 📲 Inp	ut Registers [12000]	-Read Valu		Request	Input Bits			LED1	E	BOOL	^
*	Ė… <b>*≣</b> Inp	ut Bits [1.,2000] - Rea	ad_Boolean		Address	1			LED2	E	BOOL	-
	🗖	+0: LED1		Ξ	Nb Item	2000			🛛 🖊 🚽 RE	ETAIN varial	bles	~
8	<b>D</b>	+1: LED2		=	Description	Read_Boolea	n				>	
	C			<b>Y</b>				N	ame	Value		_
	Symbol	Offset	Mask		Storage		Rang	F	amo	1 0.00		_
	LED1	0	FFFF		Default	-	1					
¢.	LED2	1	FFFF		Default							
						-						1.040
E1	S DIC Tex L M			_			>	<				<u> </u>
<b>&lt;</b>	PAC Time Va	riables LD1 <b>IO D</b>	rivers									

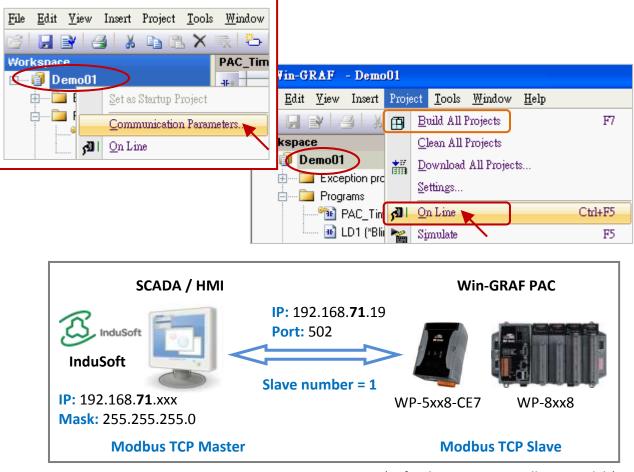
You have completed the settings for the Modbus Slave. Finally, follow the way below to re-compile the program and download it to the Win-GRAF PAC.

11. Click "Project" > "Build All Projects" from the menu bar to compile this program again (refer the <u>Section 2.3.4</u>). If a message informs you "No error detected" that means this process is successful.



12. Mouse right-click the project name (i.e., "Demo01") and select the "Communication Parameters..." to set the PAC IP (e.g., "192.168.71.19:502") and then click the menu bar "Project" > "On Line" (or

) to establish a connection and download this project to the Win-GRAF PAC. (Refer the <u>Section</u> <u>2.3.5</u>).



(Refer the <u>P1-1</u> to view all PAC models)

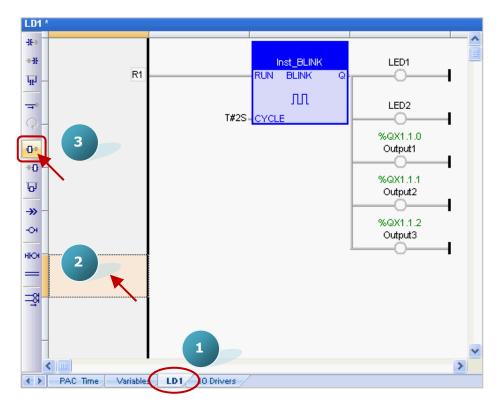
After completing all the steps, the HMI/SCADA software can access to all the Win-GRAF variables listed above via Modbus TCP protocol.

# 3.2 To Enable the Win-GRAF PAC as a Modbus RTU Slave

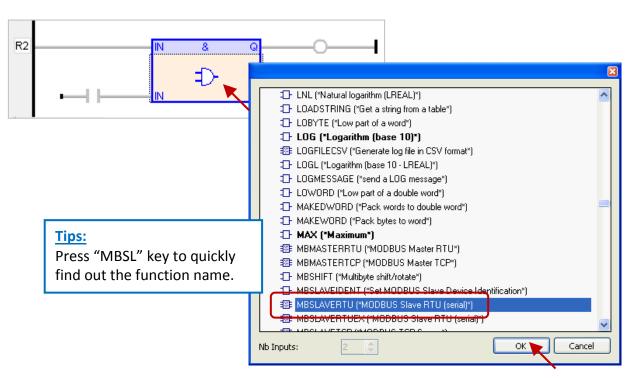
Before doing this, you must complete all the content that described in <u>Section 3.1</u> to open the Modbus Slave data. The way to enable the Win-GRAF PAC as the Modbus RTU Slave is to add the "MBSLAVERTU" or the "MBSLAVERTU**EX**" function block in the program. To begin, follow these steps:

### Add the "MBSLAVERTU" function block

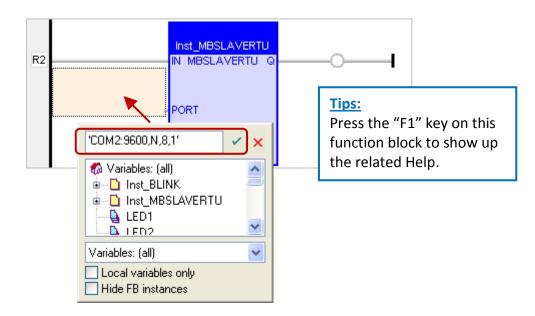
1. In the "LD1" window, mouse click the place where you want to add this function block and then click the "Insert FB.." button on the left side of the window.



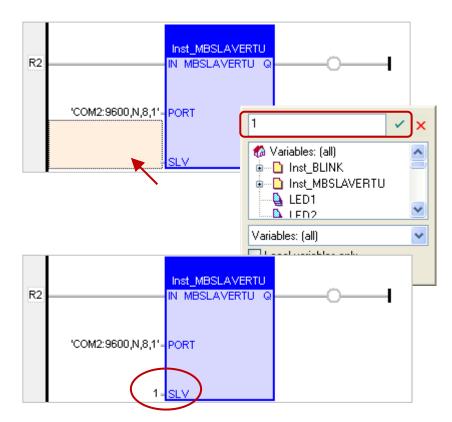
2. Double-click on this function block and select the name "MBSLAVERTU", then click "OK".



3. In the "MBSLAVERTU" function block, mouse double-click the left side of the "PORT" and enter a string 'COM2:9600,N,8,1' (it means using the Win-GRAF PAC's COM2 to communication with the Modbus Master) and then click is to complete the settings.

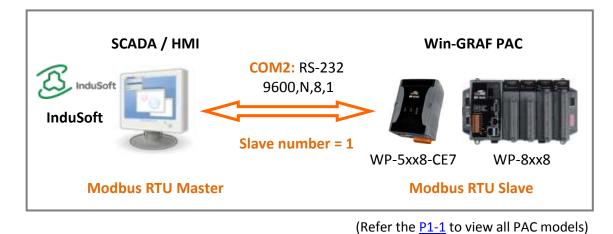


4. Double-click the left side of the "SLV" and then enter "1" (the value set in the Section 3.1 - Step 3), then click v to finish the setting.



Now, you have completed the setting of the "MBSLAVERTU" function block, and then re-compile the program and download it to the Win-GRAF PAC. (Refer the <u>Section 2.3.4</u>, <u>Section 2.3.5</u>)

<u>Note:</u> Users can enable multiple Modbus RTU Slave ports for each PAC (recommend not over 16 Ports), the way is to add multiple "MBSLAVERTU" function blocks and set the different "Port" value.

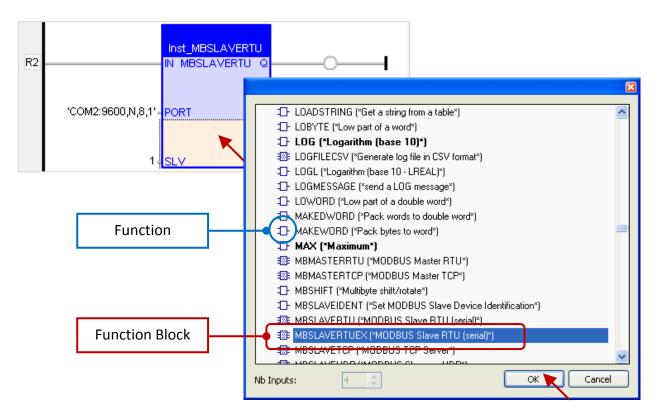


### Add the "MBSLAVERTUEX" function block

When using several "Server - ..." settings (recommend to set one) in the "IO Drivers" window, the user needs to use the "MBSLAVERTU**EX**" function block.

IO D	rivers *				
E	Ma MODBUS S	Slave		Name	Value
쁆	🗄 ··· 品 Server	- Slave number = 1		Slave number	1
	🕂 🕂 🗄 Inp	ut Registers [12000]	-Read_Value	Server ID	'Svr1'
*	🗄 🗝 🗄 Inp	ut Bits [1.,2000] - Rea	ad_Boolean		
ъ,					
	Request	Address	Nb Item	Description	
	Input Registers	1	2000	Read_Value	
	Input Bits	1	2000	Read_Boolea	n
212					
< >	PAC Time Va	riables LD IO D	rivers		

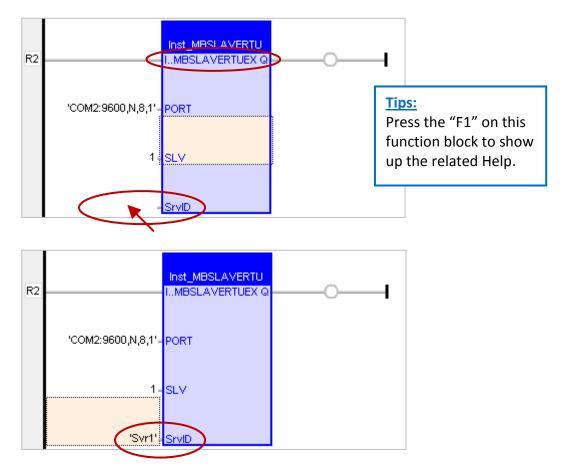
 Follow steps 1 to 4 above to add the "MBSLAVERTU**EX**" function block. If you want to change the usage for existing function block, mouse double-click the "MBSLAVERTU" and change it to the "MBSLAVERTU**EX**", and then click the "OK" button.



2. The "MBSLAVERTU**EX**" has a "SrvID" setting. Double click the left side of the "SrvID" and enter a needed "Server ID" (using a string format, e.g., 'Svr1').

### Note:

Using the "MBSLAVERTU" function block means the first Modbus Slave setting will be enabled. Using the "MBSLAVERTU**EX**" function block means to enable the Modbus Slave setting depends on the "Server ID".



Now, you have finished the settings for the "MBSLAVERTU**EX**" function block, and then re-compile the program and download it to the Win-GRAF PAC. (Refer the <u>Section 2.3.4</u>, <u>Section 2.3.5</u>)

# Chapter 4 Linking "I/O Boards"

This section lists the usage of the "I/O Boards" function in the Win-GRAF Workbench to link the Real I/O modules or to enable other I/O functions. First, you need to know the slot numbering and supported I/O modules for each PAC:

PAC Model	Slot No. (from the left to the right)	The supported PAC I/O modules		
WP-8xx8	0 to 7	Supported:		
WP-8xx8-CE7	0 to 7	I-8K and I-87K series (High Profile) I/O modules.		
XP-8x38-CE6 (*)	0 to 7	(E.g., I-8017HW and I-87055W) Not Supported:		
XP-8x48-CE6 (*)	1 to 7	I-8K and I-87K series (Low Profile) I/O modules.		
VP-x2x8-CE7 (*)	0 to 2	(E.g., I-8017H and I-87055)		
WP-5238-CE7	-	<b>The Palm-size PAC</b> which can support one XV board. (E.g., XV107, XV116, XV308, etc.) The XV board is a kind of the Modbus slave I/O board. (Refer the <u>Section 5.1.6</u> to <u>5.1.12</u> for using XV-boards.)		

(\*) : The XP-8038-CE6, XP-8048-CE6, VP-x208, WP-5xx8-CE7 are the 0-slot PAC. (Refer P1-1 for all PAC models)

### Add the "I/O board"

"I/O board" refers to the I/O functionality in the Win-GRAF (e.g., "i\_8037\_DO") and

"I/O module" refers to the hardware device (e.g., "I-8037W").

#### Notice:

Besides the software setting for the I/O board, there are some kinds of I/O modules need to set the hardware Jumper (e.g., Single-ended and Differential Jumper). So, go to the website to look up the product information, or the description printed on the module cover, or the attached shipment document.

I-8K and I-87K series product website:

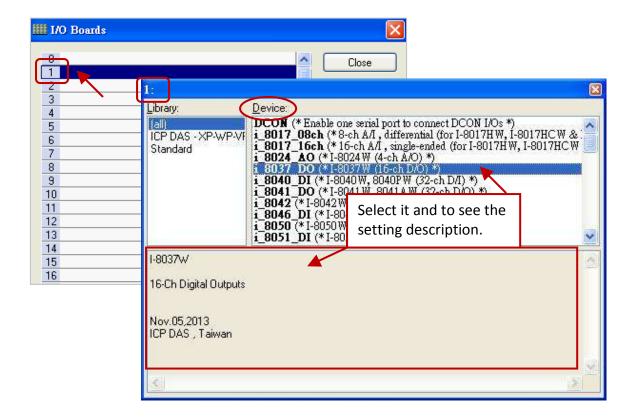
http://www.icpdas.com/root/product/solutions/remote\_io/rs-485/i-8k\_i-87k/i-8k\_i-87k\_selection.html

1. In the Win-GRAF, click the "Open I/Os" button from the toolbar to open the "I/O Boards" window.



2. As the figure below, mouse double click the slot number that corresponds to the real I/O module and then select the proper I/O board (e.g., i\_8037\_DO).

**Note:** The Slot 0 to Slot 7 are reserved for real I/O modules that plugged into the PAC, and the slot 8 or above are for other usage.



**Buttons Description:** (Click the following buttons to modify the settings)

indow.
) selecting window.
Enter": to open ; ESC" : to exit)
,
'O board.
I/O board.

"Properties":	Look up the usage of this I/O board.
"Virtual/Real":	Swith the I/O board to a Virtual I/O (for testing) or a Real I/O. (Hot Key – "Space")
"Move Up":	Move up this I/O board.
"Move Down":	Move down this I/O board.
"Help":	To see the description on "I/O devices".
"Move Up": "Move Down":	Move up this I/O board. Move down this I/O board.

## 4.1 DI/DO Boards

Here use the "I-8055W" as an example, users can refer the Chapter 4 (P4-1) to add this I/O board.

1. Double-click the "i\_8055\_DI" (or the "i\_8055\_DO") to open the "Properties" window. Note: A mouse-over showing the details on the "i\_8055\_DI" (or the "i\_8055\_DO").

II I/O Board	ls		
0 1 i_8055	55 DJ -	Close	
- 0 : 80 - 1 : 80 2 3 4 5 6 7 8 9 10	55_D0 Key=6 Ref=16#8055A Reserved0=0 Reserved1=0 Reserved2=0 Reserved3=0	Key = 6           Ref = 16#8055A           Reserved0 = 0           Reserved1 = 0           Reserved3 = 0	
11 12 13 14		I-8055W 8-Ch Digital Inputs + 8-Ch Digital Outputs Nov.01,2013 ICP DAS , Taiwan	n N

2. After linking the "i\_8055" I/O board, it will auto add 8 Input and 8 output variables in the "Variables" window that can be used in the program.

Variables									
🝸 Name	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
LED1	BOOL								~
LED2	BOOL								
RETAIN varial	les	_							
📃 🗉 📽 %IX1.0 - i_805	55_DI	•			▶ %	<mark>(1</mark> .0 – i_	805!	5 DI	
%IX1.0.0	BOOL		Input				-	_	
%IX1.0.1	BOOL		Input		<b>%</b> C	<mark>X1.1 – i</mark>	_805	5_DO	
%IX1.0.2	BOOL		Input		<b>~</b> un	means "Ir	nnut"	<mark>"Q"</mark> mea	ns "Outnut"
%IX1.0.3	BOOL		Input				•		•
%IX1.0.4	BOOL		Input		"X"	means "l	Boole	an" "1" meai	ns "Slot 1"
%IX1.0.5	BOOL		Input						
%IX1.0.6	BOOL		Input		Not	e:			
%IX1.0.7	BOOL	<b>_</b> _/	Input				thic i	tom to modify	+ho variabla
📃 🗉 %QX1.1 - i_80	055_DO							tem to modify	
%QX1.1.0	BOOL		Output		nan	ne and cli	ck "E	nter" to accep	ot the setting.
%QX1.1.1	BOOL		Output						
%QX1.1.2	BOOL		Output			🐻 %QX1	l.1 - i_	8055_DO	
%QX1.1.3	BOOL		Output					BOOL	Output
%QX1.1.4	BOOL		Output			Output1 🍗	)		Output
%QX1.1.5	BOOL		Output				$\leftarrow$	- BOOL	· · ·
%QX1.1.6	BOOL		Output			%QX1.1.2	•	BOOL	Output
%QX1.1.7	BOOL		Output		9	%QX1.1.3		BOOL	Output
<			U.						
+ PAC Time Variab	les								

# 4.2 i\_scale (Conversion Table)

The "i\_scale" function can set up to 29 scaling functions to convert values for the AI or AO module that plugged in the slot 0 to slot 7. See the <u>Chapter4</u> (P4-1) to add this I/O board.

1. Mouse double click the "i\_scale\_0" (or "i\_scale\_1" or "i\_scale\_2") to open the "Properties" window, and then to see the setting description.

**Note:** Using the slot 8 or above No. because the slot 0 to slot 7 are reserved for the real I/O module.

I/O Boards	
0	Close
2 3 4 5 6 7 8	9.0: i_scale - Properties       Key = 6       Ref = 16#2A       Ch00_X0_reserved = 0.0       Ch00_X1_reserved = 0.0       Ch00_Y1_reserved = 0.0       Ch01_X0_Min_Physical_Val = 0.0       Ch01_X0_Min_Physical_Val = 0.0
9 i_scale 0 i_scale_0 1 i_scale_1 2 i_scale_2 10 11 12 13	Ch01_Y0_Engineering_Yal_For_X0=0.0 Ch01_Y1_Engineering_Yal_For_X1=0.0 Ch02_X0_Min_Physical_Yal=0.0 Ch02_X1_Max_Physical_Yal=0.0 Ch02_Y0_Engineering_Yal_For_X0=0.0 Ch02_Y1_Engineering_Yal_For_X1=0.0 Ch03_X0_Min_Physical_Yal=0.0 Ch03_X1_Max_Physical_Yal=0.0 Setting Description
14	he scaling function No. 01 to 29 for scaling I/O variables . ting both value of Ch_XO and Ch_X1 to 0.0, it means the relative scaling function No. is disabled. XO is greater than or equal to Ch_X1, the setting is wrong.
	<u>×</u>

Parameters: ("Ch" means the Ch01 to Ch29, "Ch00" is a reserved item)

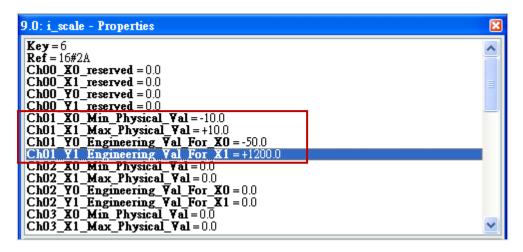
Ch_X0_Min_Physical_Val:	The min. value of AI (or AO) boards (X0).
Ch_X1_Max_Physical_Val:	The max. value of AI (or AO) boards (X1).
Ch_Y0_Engineering_Val_For_X0:	The engineering value after scaling X0.
Ch_Y1_Engineering_Val_For_X1:	The engineering value after scaling X1.

- 2. Double click the item and fill in a value, then press the "Enter" key to complete the setting. Notice:
  - 1. If set both Ch\_X0 and Ch\_X1 values to "0.0", it means the relative scaling function No. is disabled.
  - 2. If Ch\_X0 is greater than or equal to Ch\_X1, the setting is wrong.
  - 3. If Ch\_Y0 is equal to Ch\_Y1, the setting is wrong.

For example, if the AI board's value is 4 to 20 mA and wish to scale as 0 to 10000, then set Ch\_X0 as "4.0", Ch\_X1 as "20.0", Ch\_Y0 as "0.0", Ch\_Y1 as "10000.0".

For example, if the AO board's value is -10 to +10 V and their respective engineering value is

-50 to 1200, then set Ch\_X0 as "-10.0", Ch\_X1 as "+10.0", Ch\_Y0 as "-50.0", Ch\_Y1 as "+1200.0".



- 3. After linking the "i\_scale" in the "I/O Boards" window, it will auto add 30 Boolean variables in the "Variables" window. When the Win-GRAF connects the PAC, it will display the state of each scaling function.
  - True: scaling function is ok.
  - FALSE: scaling function is not enabled or setting error.

Name	Туре	Dim. Attrib.	Syb. Inil	t value 🛛 Us	er   Tag	Description	ř.
🖃 👹 %IX9.0	- i_scale_O						1
%IX9.0.0	BOOL	Input					
%IX9.0.1	BOOL	Input					ſ
%IX9.0.2	BOOL	Input					
%IX9.0.3	BOOL	Input					
%IX9.0.4	BOOL	Input					
%IX9.0.5	BOOL	Input					
%IX9.0.6	BOOL	Input					
%IX9.0.7	BOOL	Input					
%IX9.0.8	BOOL	Input					
%IX9.0.9	BOOL	Input					
🖻 🐻 %IX9.1	- i_scale_1						
%IX9.1.0	BOOL	Input					
%IX9.1.1	BOOL	Input					
%IX9.1.2	BOOL	Input					
%IX9.1.3	BOOL	Input					
%IX9.1.4	BOOL	Input					Į
%IX9.1.5	BOOL	Input					
%IX9.1.6	BOOL	Input					
%IX9.1.7	BOOL	Input					
%IX9.1.8	BOOL	Input					
%IX9.1.9	BOOL	Input					
🗉 📓 %IX9.2	- i_scale_2						
%IX9.2.0	BOOL	Input					
%IX9.2.1	BOOL	Input					2
%IX9.2.2	BOOL	Innut					1

# 4.3 i\_8017HW (8/16 channels AI)

The I-8017HW can be an 8-channel differential or a 16-channel single-ended analog input module (Data type: "REAL"). The following will describe a 16-channel module, you can see the <u>Chapter4</u> (P4-1) to add this I/O board.

<u>Note:</u> Before using the I-8017HW I/O module, it requires to set the Differential or Single-ended Jumper in the hardware.

1. Mouse double click the "i\_8017\_16ch" to open the "Properties" window, and then to see the setting description.

IIII I/O Boards	
0	Close
1 18017_16ch	
3	Select
4	1: i_8017_16ch - Properties
5	Key=6
6	<b>Kef</b> = 16#801716
7	Ch0_type = 16#08 Ch1_type = 16#08
8	Ch2_type = 16#08
9	Ch3_type = 16#08
10	Ch4_type = 16#08 Ch5_type = 16#08
11	<b>Ch6_type</b> = 10#08
12	Ch7_type = 16#08
13	Ch8_type = 16#08
14	Ch9_type = 16#08 Ch10_type = 16#08
15	Ch11 type = 16#08
16	Ch12 type = 16#08
17	Ch13_type = 16#08
-	I-8017HW , I-8017HCW , I-8017DW
	16-Ch Analog lutputs (Value type is REAL), single-ended input.
	Parameters:
	Ch_type : 16#SSRR SS : Scaling function is defined by the virtual board "i_scale"
	00 means "No scaling"
	01 to 29 means "Appling a scaling function No. (01 to 29)"

#### Parameters:

### Ch\_type: 16#SSRR

SS : Scaling function is defined by the "i\_scale" I/O board (see the <u>Section 4.2</u>).

00 means "No scaling".

01 to 29 means "Appling a scaling function No. (01 to 29)".

Setting "SS" as other value will use the default value 00.

RR : Range definition of signals.

05 means "physical input signal is -2.5 to +2.5 Volt".

06 means "physical input signal is -20 to +20 mA".

07 means "physical input signal is -1.25 to +1.25 Volt".

08 means "physical input signal is -10 to +10 Volt".

09 means "physical input signal is -5 to +5 Volt".

Setting "RR" as other value will use the default value 08.

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

For example, 16#08 means the physical input signal is -10 to +10 V.

Channel value 5.67 means the input signal is 5.67 V.

For example, 16#209 means the physical input signal is -5 to +5 V with the scaling function 2. Signal 5.67 V will be scaled to an engineering value by the scaling function 2.

For example, 16#1709 means the physical input signal is -5 to +5 V with the scaling function 17. Signal 5.67 V will be scaled to an engineering value by the scaling function 17.

: i_8017_16ch - Pro	perties	E
<b>Key</b> = 6 <b>Ref</b> = 16#801716		A
<b>Ch0</b> type = $16#1709$	Ch0_type	
Ch1_type = 16#08 Ch2_type = 16#08	16#1709	
Ch3_type = 16#08		
Ch4_type = 16#08 Ch5_type = 16#08		

**Noise\_Filter\_Max:** The max. of the physical value to be considered as noise.

The filter will filter out the signal value beyond it, default setting is "9999.9". For example, set as "7.9", signal larger than 7.9 V (or 7.9 mA) will be filtered out.

**Noise\_Filter\_Min:** The min. of the physical value to be considered as noise.

The filter will filter out the signal value beyond it, default setting is "-9999.9". For example, set as "1.5", signal smaller than 1.5 V (or 1.5 mA) will be filtered out.

### Note:

If setting Noise\_Filter\_Min >= Noise\_Filter\_Max, filter is disabled. If setting Noise\_Filter\_Min < -1000 and Noise\_Filter\_Max > 1000, the filter is disabled.

Sample\_Number: The number of sampled data to be averaged as one data.

Default is "1" (range: 1 to 500). Set a bigger value will reduce the sampling rate, however the signal curve is smoother than setting a small value.

3. After linking the "i\_8017\_16ch" in the "I/O Boards" window, it will auto add 16 "REAL" input variables in the "Variables" window. These variables can be used in the program.

Init value User Tag Description	Syb.	Dim. Attrib.	Туре	Name
			i_8017_16ch	🖂 👹 %ID1 -
		Input	REAL	%ID1.0
You can change the variabl		Input	REAL	%ID1.1
name by double clicking it.		Input	REAL	%ID1.2
		Input	REAL	%ID1.3
(See the <u>Section 4.1</u> )		Input	REAL	%ID1.4
		Input	REAL	%ID1.5
		Input	REAL	%ID1.6
		Input	REAL	%ID1.7
		Input	REAL	%ID1.8
		Input	REAL	%ID1.9
		Input	REAL	%ID1.10
		Input	REAL	%ID1.11
		Input	REAL	%ID1.12
		Input	REAL	%ID1.13
		Input	REAL	%ID1.14
		Input	REAL	%ID1.15

# 4.4 i\_8024 (4-channel AO)

The I-8024W is a 4-channel analog output module (Data type: "REAL") that can be used to output +/- 10 V or 0 to +20 mA signal. See the <u>Chapter4</u> (P4-1) to add this I/O board.

1. Mouse double-click the "i\_8024\_AO" to open the "Properties" window and then to see the setting description.

III I/O Boards	
0	Close
1 i_8024_A0	
2	1: i_8024_AO - Properties
3	Key = 6
4	<b>Ref</b> = 16#8024
5	Ch0_type = 16#33 Ch1_type = 16#33
6	Ch2_type = 16#33
7	Ch3_type = 16#33
8	Reserved0 = 0
9	Reserved1 = 0 Reserved2 = 0
10	Reserved 3 = 0
11	
12	
13	
14	
15	
16	I-8024W Setting Description
17	4-Ch Analog Outputs (Value type is REAL)
	Parameters :
	Ch_type: 16#SSRR
	SS: Scaling function is defined by the virtual board "i_scale"
	00 means "No scaling" 01 to 29 means "Appling a scaling function No. (01 to 29)"
	3

### Parameters:

### Ch\_type: 16#SSRR

SS: Scaling function is defined by the "i\_scale" I/O board (see the <u>Section 4.2</u>).

00 means "No scaling".

01 to 29 means "Appling a scaling function No. (01 to 29)".

Setting "SS" as other value will use the default value 00.

- RR: Range definition of signals
  - 30 means "physical output signal is 0 to 20 mA"
  - 33 means "physical output signal is -10 to +10 Volt"

Setting "RR" as other value will use the default value 33.

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

For example, 16#33 means the physical output signal is -10 to +10 V.

Channel value 5.67 is to output 5.67 V; value -3.752 is to output -3.752 V.

For example, 16#133 means the physical output signal is -10 to +10 V with the scaling function 1. Channel value is a user-defined engineering value will be converted first by the scaling function 1 (i.e., "Ch01") before output it as -10 to +10 V. For example, 16#30 means the physical output signal is 0 to 20 mA. Channel value 12.5 is to output 12.5 mA ; value 6.27 is to output 6.27 mA.

For example, 16#1730 means the physical output signal is 0 to 20 mA with the scaling function 17. Channel value is a user-defined engineering value will be converted first by the scaling function 17 (i.e., "Ch17") before output it as 0 to 20 mA.



3. After linking the "i\_8024\_AO" in the "I/O Boards" window, it will auto add 4 "REAL" Output variables in the "Variables" window. These variables can be used in the program.

7	Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
	🖃 👹 %QD1	- i_8024_AO							
	%QD1.0 🗲	REAL		Output	Vei	. con chor	ao tho y	(oriobl	
	%QD1.1	REAL		Output		ı can char	-		e
	%QD1.2	REAL		Output		ne by dou		•	
	%QD1.3	REAL		Output	(Se	e the <mark>Sect</mark>	<u>ion 4.1</u> )		
	<								>

# 4.5 i\_87018W (8-channel AI)

The I-87018W is an 8-channel analog input module (Data type: "REAL") that provides thermocouple input, current input (-20 mA to +20 mA) and voltage input (+/- 15 mV, +/- 50 mV, +/- 100 mV, +/- 500 mV, +/- 1 V, +/- 2.5 V). See the <u>Chapter4</u> (P4-1) to add this I/O board.

#### Important Notice:

- I-87018ZW is better than I-87018W / I-87018RW / I-87018PW because
  - (A) Each channel can use different range type setting.
  - (B) Accuracy is better and total 10-Channels.
  - (C) Temperature input with sensor-broken-line detection.

Please visit http://www.icpdas.com/products/Remote IO/i-87k/i-87018z.htm

- I-87018W does not support sensor-broken line function.
- 1. Mouse double-click the "i\_8018\_08ch" to open the "Properties" window, and then to see the setting description.

IIII I/O Boards	
0 1 i_87018_08ch	Close
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	1: i 87018_08ch - Properties         Key = 5         Ref = 16#87018         Tmp F = 16#01         Ch0 7, type = 16#05         Ch0 scale = 16#0         Ch1_scale = 16#0         Ch2_scale = 16#0         Ch5_scale = 16#0         Ch6_scale = 16#0         Ch5_scale = 16#0         Ch6_scale = 16#0         Ch7_scale = 16#0         Ch6_scale = 16#0         Reserved = 0         Reserved = 0

### Parameters:

### Tmp\_F: 16#FF

FF: Temperature format, It only apply to the channel type is Thermocouple.

- 01 means the unit of the input value is Degree Celsius
- 02 means the unit of the input value is Degree Fahrenheit

### Ch0\_7\_type: 16#RR

RR: Range definition of signals.

#### Normal Range: (For mA or Volt)

Type Code	Physical Input Signal
00	-0.015 to +0.015 V
01	-0.05 to +0.05 V
02	-0.1 to +0.1 V
03	-0.5 to +0.5 V
04	-1 to +1 V
05	-2.5 to +2.5 V
06	20 to +20 mA

When I-87018 and I-87018R are connected to a current source and set to "06" type code, an optional external 125 Ohm resistor is required.

#### Thermocouple Range: (For temperature)

Type Code	Туре	Physical Input Signal
0E	J	-210 to +760 °C
OF	К	-270 to +1372 °C
10	Т	-270 to +400 °C
11	E	-270 to +1000 °C
12	R	0 to +1768 °C
13	S	0 to +1768 °C
14	В	0 to +1820 °C
15	Ν	0 to +2320 °C
17	L	-200 to +800 °C
18	М	-200 to +100 °C
19	L <sub>DIN43710</sub>	-200 to +900 °C

Setting RR as other value will use the default value "05"

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

For example, 16#05 means the physical input signal is -2.5 to +2.5 V.

Channel value 1.28 means the input signal is 1.28 V.

Channel value -0.752 means the input signal is -0.752 V.

For example, If "Ch0\_7\_type" set as 16#0F and "Tmp\_F" set as 16#01 means the physical input signal is -270 to +1372 degree Celsius.

Channel value 25.75 means 25.75 degree Celsius.

For example, If "Ch0\_7\_type" set as 16#10 and "Tmp\_F" set as 16#01 means the physical input signal is -454 to +752 degree Fahrenheit.

Channel value 25.75 means 25.75 degree Fahrenheit.

1: i_87018_08ch - Pro	×	
<b>Key</b> = 6 <b>Ref</b> = 16#87018		
<b>Imp_F</b> = 16#02 <b>Ch0_7_type</b> = 16#10	Tmp_F	
Ch0_scale = 16#0 📉	16#02	
Ch1_scale = 16#0 Ch2_scale = 16#0		
Ch3_scale = 16#0		
Ch4_scale = 16#0 Ch5_scale = 16#0		

<u>Note:</u> If using a temperature module with a broken-line detection function (e.g., I-87018ZW), and the temperature value is greater than "9000.0", it means,

- 1. The temperature sensor may be broken-line.
- 2. The temperature sensor may be damaged.
- 3. The DCON module is not configured well to fit the connected temperature sensors.
- 4. The ohm measured by the connected sensor is not correct.

### Ch0\_scale to Ch7\_scale: 16#SS

SS: Scaling function is defined by the "i\_scale" I/O board (refer the Section 4.2).

00 means "No scaling".

01 to 29 means "Appling a scaling function No. (01 to 29)"

Setting SS as other value will use the default value 00.

For example, 16#17 means the physical input signal is converted with the scaling function 17.

3. After linking the "i\_87018\_08ch" in the "I/O Boards" window, it will auto add 8 "REAL" input variables in the "Variables" window that are available for programming.

Description	Syb. Init value User Tag	Dim. Attrib.	Туре	Name
1			· i_87018_08ch	🗆 👹 %ID1
	You can change the variable	Input	REAL	%ID1.0
	<ul> <li>name by double clicking it.</li> </ul>	Input	REAL	%ID1.1
	,	Input	REAL	%ID1.2
	(See the <u>Section 4.1</u> )	Input	REAL	%ID1.3
1		Input	REAL	%ID1.4
		Input	REAL	%ID1.5
		Input	REAL	%ID1.6
		Input	REAL	%ID1.7

# 4.6 i\_exist (Test if the I/O module exists?)

The "i\_exist" is used to check if the I-8K and I-87K series I/O modules exist in the PAC's slot 0 to 7. See the <u>Chapter4</u> (P4-1) to add this I/O board.

1. Mouse double-click the "i\_exist" to open the "Properties" window, and then to see the setting description.

Note: Using the Slot 8 or above No. because the slot 0 to slot 7 are reserved for the real I/O module.

III I/O Boards		
0	Close	
2 3 4 5 6 7 8 9 j_exist 10 11 12 13 14	9: i_exist - Properties Key = 6 Ref = 16#1 Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0	
15 16 17	i_exist Test if I-8xxxW and I-87xxxW io boards in th	
	8-ch boolean inputs represent the board stat True : exists FALSE : board not found	te in the slot U thru. 7

2. After linking the "i\_exist" in the "I/O Boards" window, it will auto add 8 "BOOL" input variables in the "Variables" window and display the state of the I/O module from slot 0 to slot 7 when connecting the Win-GRAF PAC.

"TRUE" means the I/O module exists.

"FALSE" means cannot find this I/O module.

Syb. Init value User Tag	Dim. Attrib.	Туре	Name
		- i_exist	🗆 📓 %IX9
	Input	BOOL	%IX9.0
0	Input	BOOL	%IX9.1
	Input	BOOL	%IX9.2
(See the <u>Section 4.1</u> )	Input	BOOL	%IX9.3
	Input	BOOL	%IX9.4
	Input	BOOL	%IX9.5
	Input	BOOL	%IX9.6
	Input	BOOL	%IX9.7
	Syb. Init value User Tag You can change the variable name by double clicking it. (See the <u>Section 4.1</u> )	Input Input Input Input Input Input Input Input Input Input	- i_exist BOOL Input BOOL Input BOOL Input BOOL Input BOOL Input BOOL Input BOOL Input BOOL Input BOOL Input BOOL Input

# 4.7 i\_8084 (Frequency, UP/Down Counter, UP Counter)

The I-8084W is a 4/8-channel high speed Frequency/Counter (Data type: "DINT") that can be used to measure frequency or as a UP/Down Counter or as a UP Counter. The following will describe these three modes. See the <u>Chapter4</u> (P4-1) to add this I/O board.

## 4.7.1 i\_8084\_freq (8-channel Frequency)

1. Mouse double-click the "i\_8084\_freq" to open the "Properties" window, and then to see the setting description.

0 1 i 8084 freg	Close	
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	1: i 8094_freq - Properties         Key = 6         Ref = 16#8084F         Ch0 1 Filter = 0         Ch0 1 Low High_Auto = 2         Ch2 3 Fiter = 0         Ch2 3 Freq_timeout = 1800         Ch4 5 Freq_timeout = 1800         Ch6 7 Filter = 0         Ch6 7 Filter = 0         Ch6 7 Fore_timeout = 1800         Ch6 7 Filter = 0         Ch6 7 Freq_timeout = 1800         Ch6 7 Filter = 0         Signal_Inverted = 0         Signal_Inverted = 0         Setting Description         I-8084W:         8-Ch Frequency Inputs (Value type is DINT)         Parameters:         Ch_Filter: unit is 0.000001 second , can be 0 ~ 200.         The default setting is 0 (without filter).         The "Ch_Filter" is for filtering out some noise signals with samil signal width.	

### Parameters:

- **Ch\_Filter:** The unit is 0.000001 second ( $\mu$ s), the value can be 0 to 200.
  - The default setting is 0 (without filter). The "Ch\_Filter" is for filtering out some noise signals with smaller signal width. (Recommend 0: if there is no noise consideration or need a real-time measurement.) The following setting is recommended:

Max Input Signal (Hz)	Recommend Filter Value
1K	200
2К	100
5K	40
10К	20
20К	10
100K	2
450K	1
450K	0 (without filter)

### Ch\_Freq\_Timeout:

The unit is 0.001 second (ms), the value can be 20 to 1800. Set as other value will use the default value 1800. If there is no signal wave input to the I-8084W in the "Ch\_Freq\_Timeout" interval, the frequency value of the related channel will be assigned as 0.

For example, if set it as 100 ms and the input is 500 Hz (that means, one signal wave takes about 2 ms to happen), the frequency is updated normally. When the input frequency drop to 9 Hz (that means, one signal wave take about 111 ms to happen), this "111" exceeds the setting "100" ms (Freq\_Timeout). So the frequency value will be assigned as 0 because there is no signal wave coming in this 100 ms interval.

When setting as 20 ms, the frequency value below 50 Hz is not detectable (become 0). When setting as 100 ms, the frequency value below 10 Hz is not detectable (become 0). When setting as 1800 ms, the frequency value of 0 Hz, 1 Hz to 450 KHz is detectable.

#### Ch\_Low\_High\_Auto: (recommend setting as "2: Auto".)

- 0 means a Low frequency mode.
- 1 means a High frequency mode.
- 2 means Auto switching between Low and High frequency mode.

Set as other value will use the default value "2: Auto".

Mode 2 will auto change the frequency mode. It will auto change to High mode when the input frequency is larger than 3500 Hz, while auto change to Low mode if input frequency is less than 1000 Hz.

DO NOT set as 1 (High frequency mode) if the input signal is normally less than 1000 Hz, or the frequency value will be incorrect frequently. Recommended don't set as 0 (Low frequency mode) if the input signal is normally larger than 3500 Hz.

#### Min\_Update\_Interval:

The unit is 0.001 second (ms), the value can be 0, or 20 to 1000.

Default value 0 means "Update frequency every PAC cycle".

Other means "Update frequency when each Interval time reached".

The frequency update time also depends on the Win-GRAF PAC cycle time. If the PAC cycle time is big, for example 200 ms, then the real frequency update time will become 200 ms when setting the "Min\_Update\_Interval" less than 200. Setting bigger "Min\_Update\_Interval" will get smooth frequency curve value, however the frequency value is updated slowly.

#### Signal\_Inverted:

- 0: input signal is normal (no inverted).
- 1: input signal is inverted (means voltage HIGH will be processed as LOW, and voltage LOW will be processed as HIGH).

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

1: i_8084_freq - Properties	×
Key = 6 Ref = 16#8084F Ch0_1_Filter = 0	~
Ch0_1_Freq_timeout = 100 Ch0_1_Low_High_Auto = 2 Ch2_3_Filter = 0 Ch2_3_Freq_timeout = 1800 Ch2_3_Low_High_Auto = 2 Ch2_3_Low_High_Auto = 2	
Ch4_5_Filter = 0 Ch4_5_Freq_timeout = 1800 Ch4_5_Low_High_Auto = 2 Ch6_7_Filter = 0	
Ch6_7_Freq_timeout = 1800 Ch6_7_Low_High_Auto = 2 Min_Update_Interval = 200 Signal_Inverted = 0	~

3. After linking the "i\_8084\_freq" in the "I/O Boards" window, it will auto add 8 "DINT" input variables in the "Variables" window. When the Win-GRAF connects the PAC, it will display the frequency value for each channel.

Variables				
🍸 Name	Туре	Dim. Attrib.	Syb. Init value User Tag	Description
🖃 👹 %ID1	- i_8084_freq			^
%ID1.0	DINT	Input	You can change the variable	
%ID1.1	DINT	Input	U	
%ID1.2	DINT	Input	name by double clicking it.	
%ID1.3	DINT	Input	(See the <u>Section 4.1</u> )	
%ID1.4	DINT	Input		
%ID1.5	DINT	Input		
%ID1.6	DINT	Input		
%ID1.7	DINT	Input		~
<				>

## 4.7.2 i\_8084\_cnt\_ch04 (4-channel UP/Down Counter)

- Note: Using the "COUNTER\_START", "COUNTER\_STOP", "COUNTER\_GET", "COUNTER\_STATE" and "COUNTER\_RESET" functions in the Win-GRAF Workbench to operate counter channels in an I-8084W.
- 1. Mouse double-click the "i\_8084\_cnt\_ch04" to open the "Properties" window, and then to see the setting description.

III I/O Boards	
0 1 i_8084_cnt_04ch	Close
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	Key=6         Ref = 16#8084C4         Ch0_Mode = 0         Ch1_Mode = 0         Ch1_Filter = 0         Ch2_Filter = 0         Ch3_Mode = 0         Ch4_Filter = 0         Signal_Inverted = 0         Reserved0 = 0         Reserved1 = 0         Reserved3 = 0         I+8084W : Setup 4-Ch Counter Inputs         Note: Please use the "1_8084_Start", "1_8084_Stop", "1_8084_Get", "1_8084_State" and "1_8084_Reset" functions to operate counter channels in an 1-8084W.         Parameters :
Daramatara	<u>S</u>

### Parameters:

**Ch\_Mode:** Input mode, can be 0, 1 and 4. Set other value will use 0.

- 0: Pulse/DIR mode.
- 1: UP/DOWN mode.
- 4: A/B phase (Quard.) mode.
- **Ch\_Filter:** The unit is 0.000001 second ( $\mu$ s), the value can be 0 to 200.

The default setting is 0 (without filter). The "Ch\_Filter" is for filtering out some noise signals with smaller signal width. (Recommend 0: if there is no noise consideration or need a real-time measurement.) The following setting is recommended:

Max Input Signal (Hz)	Recommend Filter Value
1K	200
2К	100
5K	40
10K	20
20K	10
100K	2
450K	1
450K	0 (without filter)

#### Signal\_Inverted:

- 0: Input signal is normal (no inverted).
- 1: Input signal is inverted (means voltage HIGH will be processed as LOW, and voltage LOW will be processed as HIGH).

### For example:

If setting "Signal\_Inverted" as 0 (no inverted) and Ch\_Mode is 0 (Pulse/DIR), the counter value will count up if "DIR" signal is High.

If setting "Signal\_Inverted" as 1 (inverted) and Ch\_Mode is 0 (Pulse/DIR), the counter value will count down if "DIR" signal is High.

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

1: i_8084_cnt_04ch - Properties	×
Key = 6 Ref = 16#8084C4 Ch0_Mode = 0 Ch0_Filter = 0 Ch1_Mode = 0 Ch1_Filter = 0 Ch2_Mode = 0 Ch2_Filter = 0 Ch3_Mode = 0 Ch4_Filter = 0	
Signal Inverted = 1       Signal_Inverted         Reserved0 = 0       1         Reserved1 = 0       1         Reserved2 = 0       1         Reserved3 = 0       1	3

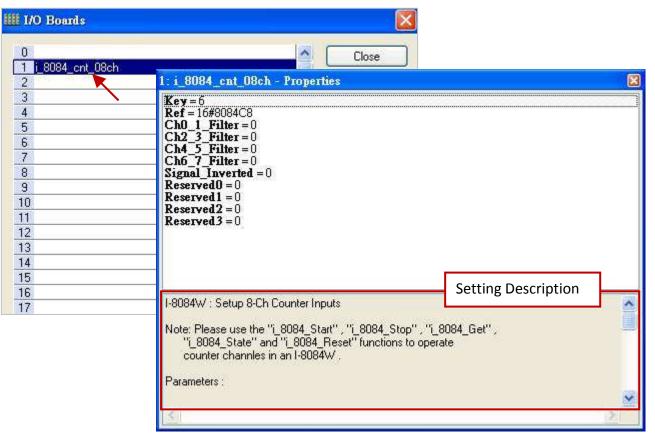
3. After linking the "i\_8084\_cnt\_ch04" in the "I/O Boards" window, it will auto add one "BOOL" Input variable (no meaning, always "FALSE") in the "Variables" window.

Va	riables								
7	Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
	🖂 👹 %IX1 -	i_8084_cnt_04	1ch						^
	%IX1.0	BOOL		Input					~
	<	1		i sunacionali.					>

 After linking the "I/O board", refer the <u>Section 4.9</u> to use "COUNTER\_START", "COUNTER\_STOP", "COUNTER \_GET", "COUNTER\_STATE" and "COUNTER\_RESET" functions in the LD or ST program to operate the Counter channel of the I-8084W.

## 4.7.3 i\_8084\_cnt\_ch08 (8-channel UP Counter)

- <u>Note:</u> Using the "COUNTER\_START", "COUNTER\_STOP", "COUNTER\_GET", "COUNTER\_STATE" and "COUNTER\_RESET" functions in the Win-GRAF Workbench to operate counter channels in an I-8084W.
- 1. Mouse double-click the "i\_8084\_cnt\_ch08" to open the "Properties" window, and then to see the setting description.



### Parameters:

**Ch\_Filter:** The unit is 0.000001 second ( $\mu$ s), the value can be 0 to 200.

The default setting is 0 (without filter). The "Ch\_Filter" is for filtering out some noise signals with smaller signal width. (Recommend 0: if there is no noise consideration or need a real-time measurement.) The following setting is recommended:

Max Input Signal (Hz)	Recommend Filter Value
1K	200
2К	100
5K	40
10K	20
20K	10
100K	2
450K	1
450K	0 (without filter)

### Signal\_Inverted:

- 0: Input signal is normal (no inverted)
- 1: Input signal is inverted (means voltage HIGH will be processed as LOW, and voltage LOW will be processed as HIGH).

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

1: i_8084_cnt_08ch - Prop	erties	×
Key = 6 Ref = 16#8084C8 Ch0_1_Filter = 0 Ch2_3_Filter = 0 Ch4_5_Filter = 0 Ch6_7_Filter = 0 Signal_Inverted = 0 Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0	Signal_Inverted	3

3. After linking the "i\_8084\_cnt\_ch08" in the "I/O Boards" window, it will auto add one "BOOL" Input variable (no meaning, always "FALSE") in the "Variables" window.

7	Name 🗸	Туре	Dim.	Attrib.	Syb.	Init value	User Tag	Description
	🖂 👹 %IX1 - i_808	4_cnt_08ch				P		^
	%IX1.0	BOOL	1	Input				×
	<				3			>

 After linking the "I/O board", refer the <u>Section 4.9</u> to use "COUNTER\_START", "COUNTER\_STOP", "COUNTER \_GET", "COUNTER\_STATE" and "COUNTER\_RESET" functions in the LD or ST program to operate the Counter channel of the I-8084W.

# 4.8 i\_8093 (3-axis High Speed Encoder Module)

The I-8093W is a 3-axis high speed encoder module that can be independently configured as one of the Quadrant, Pulse/Direction or CW/CCW input mode for each channel. If not familiar with the way to add the this I/O board, see the <u>Chapter4</u> (P4-1).

1. Mouse double-click the "i\_8093" to open the "Properties" window, and then to see the setting description.

II I/O Boards		
0 1 i 8093	Close	
2 3 4 5 6	1: i_8093 - Properties Key = 6 Ref = 16#8093 X_Mode = 1 Y_Mode = 1	
7 8 9 10 11	Z_Mode = 1 Signal_Inverted = 0 Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0	
12 13 14 15 16	Setting Description	
17	I-8093W : Setup 3-Ch Encoder module Note: Please use the "i_8093_Start" , "i_8093_Stop" , "i_8093_Get" , "i_8093_State" and "i_8093_Reset" functions to operate encoder channles in an I-8093W .	
	Parameters :	2

### Parameters:

### X\_Mode, Y\_Mode, Z\_Mode:

The input mode of X, Y, Z axis, can be 1, 2 and 3. Set other value will use 1.

- 1: CW/CCW counting mode.
- 2: Pulse/Directioncounting mode.
- 3: A/B phase (quadrant) counting mode.

### Signal\_Inverted:

- 0: Input signal is normal (no inverted)
- 1: Input signal is inverted (means voltage HIGH will be processed as LOW, and voltage LOW will be processed as HIGH).

For example:

If setting "Signal\_Inverted" as 0 (no inverted) and X\_Mode is 2 (Pulse/Direction), the encoder value will increase if "Direction" signal is High.

If setting "Signal\_Inverted" as 1 (inverted) and X\_Mode is 2 (Pulse/Direction),

the encoder value will decrease if "Dirextion" signal is High.

2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

×	rties	1 i_8093 - Prope
	X_Mode 2	Key = 6 Ref = 16#8093 X Mode = 2 Y _Mode = 1 Z_Mode = 1 Signal_Inverted Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0
		Reserved1 = 0 Reserved2 = 0

- 3. After linking the "i\_8093" in the "I/O Boards" window, it will auto add 3 "BOOL" input variables in the "Variables" window that are available for programming.
  - Ch0: Z-index of X axis.
  - Ch1: Z-index of Y axis.
  - Ch2: Z-index of Z axis.

Vari	iables								
71	Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
Ĩ	= 📓 %IX1 -	·i 8093							^
	%IX1.0	BOOL		Input					
	%IX1.1	BOOL		Input					
	%IX1.2	BOOL		Input					~
3	<								>

 After linking the "I/O Boards", refer the <u>Section 4.9</u> to use "COUNTER\_START", "COUNTER\_STOP", "COUNTER \_GET", "COUNTER\_STATE" and "COUNTER\_RESET" functions in the LD or ST program to operate the Encoder channel of the I-8093W.

# 4.9 Using the Count Function for I-8084W, I-8093W, I-87082W, I-87084W, I-7083 and I-7080 Modules

This section lists the way to use the "COUNTER\_START", "COUNTER\_STOP", "COUNTER\_GET", "COUNTER\_STATE" and "COUNTER\_RESET" functions in the LD or ST program to operate the Counter and Encoder modules. If not familiar with the way to create a program or a function block, see the <u>Section 2.3.3</u>.

#### Note:

- 1. In the following content, we use I-8084W and I-8093W modules as examples.
- Before using these function blocks, first go to <u>Section 4.7.2</u> (UP/Down Counter), <u>Section 4.7.3</u> (UP Counter) and <u>Section 4.8</u> (Encoder) to link I/O Boards.

## 4.9.1 COUNTER\_START

For example: Using the I-8084W module in the PAC's slot2 and start counting the channel 5.

#### ST program:

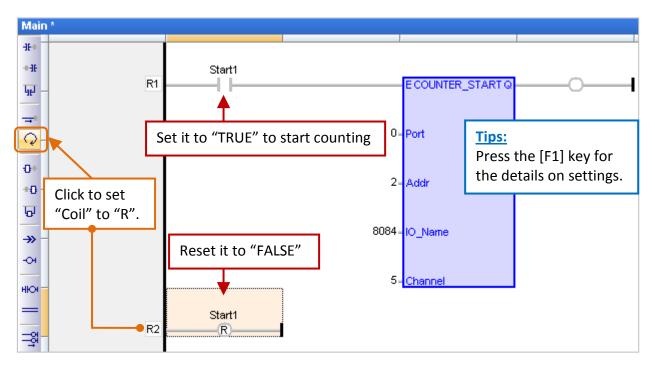
```
IF Start1 = TRUE THEN
Start1 := FALSE;
TMP_BOOL := Counter_Start (0, 2, 8084, 5);
END_IF;
```

#### Note:

First, add two BOOL variables ("Start1", "TMP\_BOOL") in the Variables Area.

#### LD program:

("Start1": boolean, set it to "TRUE" to start counting and then reset "Start1" to "FALSE".)



#### Port: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as "0". For a DCON module connected to a serial COM port, can be "1 to 37" (depends on PAC, means COM1 to COM37).

#### Addr: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as the slot number (0 to 7). For a module connected to a serial COM port, set it as the Net-ID address of the module (1 to 255).

#### IO\_Name: (Data type: "DINT")

The name of relative Counter/Encoder module, it can be set to "8084", "8093", "87084", "87082", "7083" and "7080".

#### Channel: (Data type: "DINT")

The channel No. of the Counter/Encoder module, it can be set to "0", "1", and so on, depends on the module. For example, when using the I-8093W module, "0" means the X-axis, "1" means the Y-axis and "2" means the Z-axis.

**Q:** (Data type: "BOOL") "TRUE": OK ; "FALSE": Error.

# 4.9.2 COUNTER\_STOP

For example: Using the I-8093W module in the PAC's slot1 and stop counting the X-axis.

#### ST program:

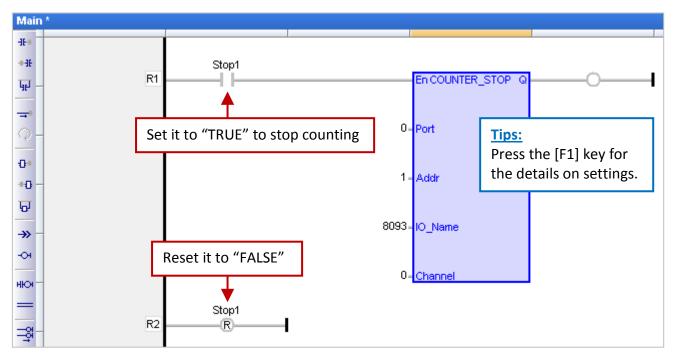
IF Stop1 = TRUE THEN Stop1 := FALSE ; TMP\_BOOL := Counter\_Stop (0, 1, 8093, 0) ; END\_IF ;

#### Note:

First, add two BOOL variables ("Stop1", "TMP\_BOOL") in the Variables Area.

#### LD program:

("Stop1": Boolean, set it to "TRUE" to start counting and then reset "Stop1" to "FALSE".)



#### Port: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as "0". For a DCON module connected to a serial COM port, can be "1 to 37" (depends on PAC, means COM1 to COM37).

#### Addr: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as the slot number (0 to 7). For a module connected to a serial COM port, set it as the Net-ID address of the module (1 to 255).

#### IO\_Name: (Data type: "DINT")

The name of relative Counter/Encoder module, it can be set to "8084", "8093", "87084", "87082", "7083" and "7080".

#### Channel: (Data type: "DINT")

The channel No. of the Counter/Encoder module, it can be set to "0", "1", and so on, depends on the module. For example, when using the I-8093W module, "0" means the X-axis, "1" means the Y-axis and "2" means the Z-axis.

**Q:** (Data type: "BOOL") "TRUE": OK ; "FALSE": Error.

# 4.9.3 COUNTER\_GET

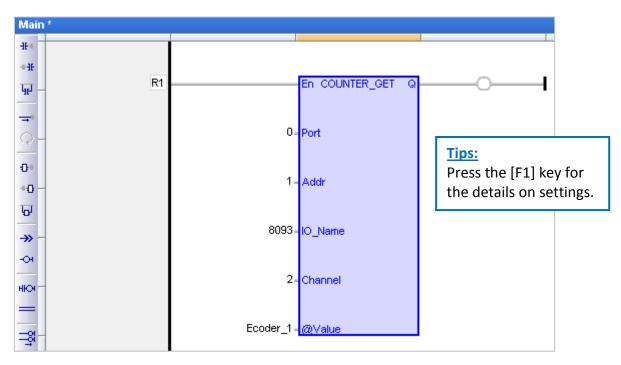
For example: Using the I-8093W module in the PAC's slot1 and get the Encoder value of the Z-axis.

#### ST program:

TMP\_BOOL := Counter\_Get (0, 1, 8093, 2, Encoder\_1);

**Note:** First, add variables in the Variables Area (see section 2.2.1). "TMP\_BOOL" (BOOL). "Encoder 1" (DINT).

#### LD program:



#### Port: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as "0". For a DCON module connected to a serial COM port, can be "1 to 37" (depends on PAC, means COM1 to COM37).

#### Addr: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as the slot number (0 to 7). For a module connected to a serial COM port, set it as the Net-ID address of the module (1 to 255).

#### IO\_Name: (Data type: "DINT")

The name of relative Counter/Encoder module, it can be set to "8084", "8093", "87084", "87082", "7083" and "7080".

#### Channel: (Data type: "DINT")

The channel No. of the Counter/Encoder module, it can be set to "0", "1", and so on, depends on the module. For example, when using the I-8093W module, "0" means the X-axis, "1" means the Y-axis and "2" means the Z-axis.

**@Value:** (The data type can be "DINT", "UDINT", "DWORD" and "LINT")

It returns the current counter or encoder value. (Refer the <u>Appendix A</u> for the range of values)

**Q:** (Data type: "BOOL") Counting state. "TRUE": Counting ; "FALSE": Stopped.

# 4.9.4 COUNTER\_STATE

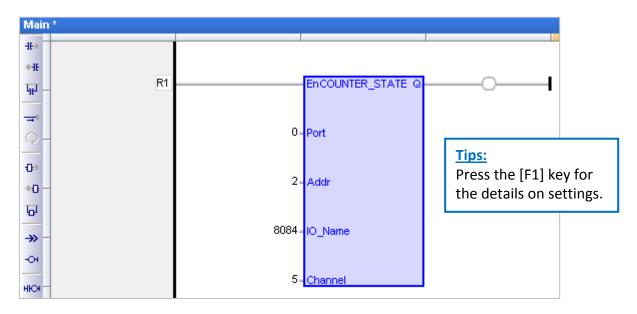
For example: Using the I-8084W module in the PAC's slot2 and to get the counting status of the channel 5.

#### ST program:

TMP\_BOOL := Counter\_State (0, 2, 8084, 5) ;

**Note:** First, add a "TMP\_BOOL" BOOL variable in the Variable Area.

#### LD program:



#### Port: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as "0". For a DCON module connected to a serial COM port, can be "1 to 37" (depends on PAC, means COM1 to COM37).

#### Addr: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as the slot number (0 to 7). For a module connected to a serial COM port, set it as the Net-ID address of the module (1 to 255).

#### IO\_Name: (Data type: "DINT")

The name of relative Counter/Encoder module, it can be set to "8084", "8093", "87084", "87082", "7083" and "7080".

#### Channel: (Data type: "DINT")

The channel No. of the Counter/Encoder module, it can be set to "0", "1", and so on, depends on the module. For example, when using the I-8093W module, "0" means the X-axis, "1" means the Y-axis and "2" means the Z-axis.

**Q:** (Data type: "BOOL") "TRUE": OK ; "FALSE": Error.

# 4.9.5 COUNTER\_RESET

For example: Using the I-8093W module in the PAC's slot5 and reset the Encoder value of the Y-axis as "0".

#### ST program:

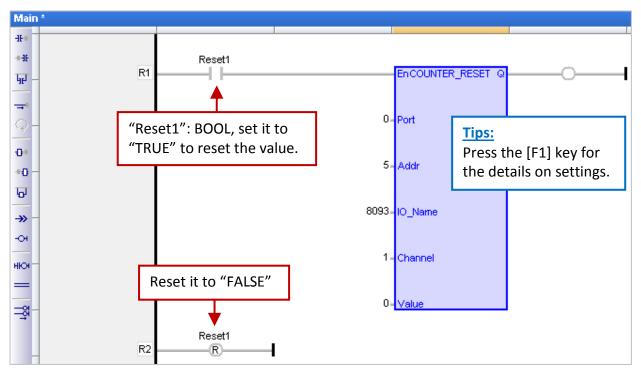
 IF
 Reset1 = TRUE
 THEN
 Note:

 Reset1 := FALSE;
 First, add two BOOL variables
 ("Reset1", "TMP\_BOOL") in the

 TMP\_BOOL := Counter\_Reset (0, 5, 8093, 1, 0);
 Variable Area.

 END\_IF;
 If the set of the

#### LD program:



#### Port: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as "0". For a DCON module connected to a serial COM port, can be "1 to 37" (depends on PAC, means COM1 to COM37).

#### Addr: (Data type: "DINT")

For a module in the slot 0 to 7 of the PAC, set it as the slot number (0 to 7). For a module connected to a serial COM port, set it as the Net-ID address of the module (1 to 255).

#### IO\_Name: (Data type: "DINT")

The name of relative Counter/Encoder module, it can be set to "8084", "8093", "87084", "87082", "7083" and "7080".

# Channel: (Data type: "DINT")

The channel No. of the Counter/Encoder module, it can be set to "0", "1", and so on, depends on the module. For example, when using the I-8093W module, "0" means the X-axis, "1" means the Y-axis and "2" means the Z-axis.

**Value:** (The data type can be "DINT", "UDINT", "DWORD", and "LINT") The new Counter or Encoder value wish to set.

**Q:** (Data type: "BOOL") "TRUE": OK ; "FALSE": Error.

# 4.10 Ping\_ip (Test an Ethernet/Internet Connection)

The "Ping\_ip" function is used to test if the connection of the remote Ethernet/ Internet device is working properly. (It supports a max. of 50 IP settings.) If not familiar with the way to add this I/O board, see the <u>Chapter4</u> (P4-1).

1. Mouse double-click the "Ping\_ip" to open the "Properties" window, and then to see the setting description.

Note: Using the slot 8 or above No. because the slot 0 to 7 are reserved for the real I/O module.

III I/O Boards		
0	Close	
2	9.0: Ping_ip - Properties	×
3 4 5 6 7 8 9 Ping ip 0 Ping ip 00 09 1 Ping ip 10 19 2 Ping ip 20 29	Key = 6         Ref = 16#6A         IP_0 = 192.168.1.100         Interval_0 = 15         Timeout_0 = 3         IP_1 = N/A         Interval_1 = 15         Timeout_1 = 3         IP_2 = N/A         Interval_2 = 15         Timeout_2 = 3         IP_3 = N/A         Interval_3 = 15	
3 Ping_ip_30_39 4 Ping_ip_40_49 10 11 12	Timeout_3 = 3         IP_4 = N/A         Interval_4 = 15         Ping_ip : Test Ethernet / Internet connection         Note:         1. If wish to test the connection between PAC and Internet, please set proper "Gateway" settings.         2. If test only local Ethernet connection, then	]
	"Gateway" may not be necessary. 3. One PAC can use only one "Ping_IP" .	2

#### Note:

- 1. If wish to test the connection between PAC and Internet, please set proper "Gateway" settings.
- 2. If test only local Ethernet connection, then "Gateway" may not be necessary.
- 3. One PAC can use only one "Ping\_IP". (Don't use two or more)
- 4. When Ping success, return Boolean channel as TRUE.
- 5. When Ping fails, it will try one more time. If still fail, then return Boolean channel as FALSE.

#### Parameters:

**IP\_01 to IP\_49:** (Data type: "STRING") The IP address of targets. Set as 'N/A' if wish to disable it. For example, 192.168.1.100 or 52.19.125.242 or N/A.

Interval\_01 to Interval\_49: (Data type: "DINT")

The unit is second. The interval to send one "ping" command. Value can be 6 to 86,400 seconds. Setting smaller than 6 will use as 6. Setting greater than 86400 (24 hours) will use as 86400.

#### Timeout\_01 to Timeout\_49: (Data type: "DINT")

The unit is second. The timeout settings of the "ping" command. Value can be 2 to 30 seconds. Setting smaller than 2 will use as 2. Setting greater than 30 will use as 30.

- Note: The "Interval\_xx" value should be **at least triple** of the "Timeout\_x" value. Or the PAC will use the "Interval\_x" value as a triple of the "Timeout\_x" value. For example, if "Timeout\_00" is set as 10 however "Interval\_00" is set as 20, then PAC will use "Interval\_00" as 30.
- 2. Double click the item and fill in a value, then press the "Enter" key to complete the setting.

9.0: Ping_ip - Prop	erbes	×
Key = 6 Ref = 16#6A IP_0 = 192.168.1.100 Interval_0 = 15 Timeout 0 = 3		<b>^</b>
$\begin{array}{llllllllllllllllllllllllllllllllllll$	IP_1 192.168.78.88	
<b>Timeout</b> 2 = 3 <b>IP_3</b> = N/A <b>Interval_3</b> = 15 <b>Timeout_3</b> = 3 <b>IP_4</b> = N/A <b>Interval_4</b> = 15		×

- 3. After linking the "Ping\_ip" in the "I/O Boards" window, it will auto add 50 "BOOL" input variables in the "Variables" window. When the Win-GRAF connects the PAC, it will display the online status.
  - True: The connection is ok.
  - FALSE: Connection failed or cable problem.

Variables				HZ	×
🍸 Name	Туре	Dim. Attrib.	Syb. Init value User Tag	Description	
🖃 👹 %IX9.0	- Ping_ip_00_	09			^
%IX9.0.0	BOOL	Input	You can change the veriable		
%IX9.0.1	BOOL	Input	You can change the variable		
%IX9.0.2	BOOL	Input			
%IX9.0.3	BOOL	Input	(See the <u>Section 4.1</u> )		
%IX9.0.4	BOOL	Input		4	
%IX9.0.5	BOOL	Input			
%IX9.0.6	BOOL	Input			
%IX9.0.7	BOOL	Input			
%IX9.0.8	BOOL	Input			
%IX9.0.9	BOOL	Input			-
🖭 📓 %IX9.1	- Ping_ip_10_	19			
표 📓 %IX9.2	- Ping_ip_20_	29			
🖭 👹 %IX9.3	- Ping_ip_30_	39			
🖭 👹 %IX9.4	- Ping ip 40	49			~
<		100		>	

# 4.11 I-8088W (8-channel PWM Output Module)

The I-8088W is an 8-channel PWM (Pulse Width Modulation) output module. The duty range (Duty = High / (High + Low)) of the PWM output signal can be from 0.1% to 99.9%. Its output frequency in the Win-GRAF PAC is from 1 Hz to 500 KHz. The I-8088W support two PWM output modes, one is the "Continuous" mode. It outputs always. The other one is the "Burst" mode. It outputs the required pulse count and then stop. Please visit <u>http://www.icpdas.com/products/Remote\_IO/i-8ke/i-8088w.htm</u> for other specifications.

#### Hardware Connection Diagram:

This example uses the I-8084W (Slot 2) to measure the frequency of the I-8088W (Slot 1) PWM output signal (the I-8084W is not necessary in the actual application). Then, connect the I-8088W's PWM output channel 0 (PW0) to the I-8084W's frequency input channel 0 (COA+).

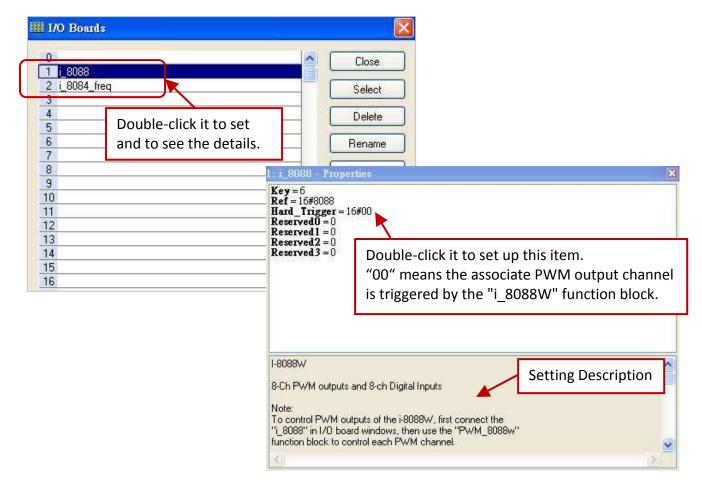


The demo project (demo\_8088w.zip) that we will describe below is located in the shipment CD (\Napdos\Win-GRAF\demo-project), refer the <u>Chapter 12</u> to restore/open this project and set up the current PAC's IP address.

<u>File Edit V</u> iew Insert Project	<u>T</u> ools <u>W</u> indow <u>H</u> elp		
3 2 2 3 3 4 5 6	🗙 🐒 🕹 이 이 👬 🖩 표 😘 💁 🖆 통한 🔐 🕌 🔊		
Vorkspace	Main		
Main Main Match (for de	Inst_PWM_8088W         Set as Startup Project         Ommunication Parameters         On Line         On Line         Ontropy of the state         On Chan         Num         18088_FreqO         0.001 ~ 0.999 (0         18088_Chttp         18088_Chttp         18088_Chttp         18088_Run0         18088_Run0         18088_Nun0         Num	Name     Type       Main     Global variables       i8088_Freq0     DINT       i8088_Duty0     REAL       i8088_Run0     BOOL       i8088_Run0     BOOL       i8088_State0     BOOL       i808     (Ised)       ight     (Ised)       ight     (Ised)       ight     Advanced       ight     Arrays       ight     BACnet	) Din
	Build		×

#### I/O Boards:

In this case, add the "i\_8088" and the "i\_8084\_freq" to the corresponding I/O slot No. in the "I/O Boards" window (see the <u>Chapter 4</u>). Then, mouse double-click the Slot No. to open the "Properties" window and you can see the setting description to set this I/O board.



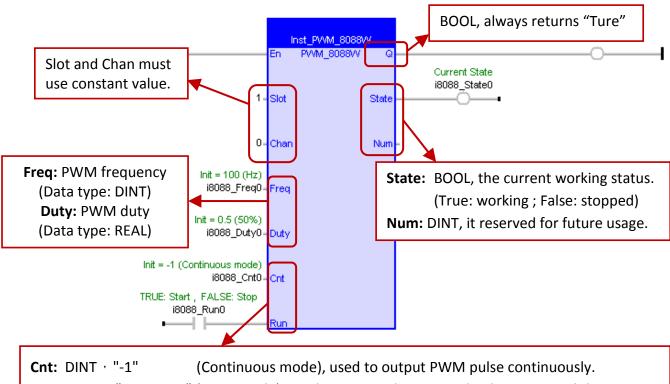
After linking the "i\_8088" and the "i\_8084\_freq" I/O boards, it will auto add related variables in the "Variables" window (or Variables Area). And, you can also declare all the variables that will be used in the program here (refer the <u>Section 2.3.1</u>).

/orkspace	Variables								
🗊 demo_8088 <del>w</del>	V Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
🗄 🖂 🖾 Exception programs	🗆 🚮 Global variables								
🖨 🛅 Programs	i8088_Freq0	DINT				100			unit is Hz, init as 100
🛄 Main	i8088_Duty0	REAL				0.5			0.001 ~ 0.999 (0.1% ~ 99.9%
😑 🔤 Watch (for debugging)	i8088_Cnt0	DINT				-1			init as -1:Continuous mode, 1
Soft Scope	i8088_Run0	BOOL							Set TRUE to start , Set FALS
Initial values	i8088_State0	BOOL							Current State
	RETAIN variables								
- 📆 Binding Configuration	🗆 👹 %IX1 - i_8088					Tor	leclar	e all	the variables that
🧕 🧕 Globel defines	%IX1.0=i8088_DI_0	BOOL		Input					
Variables	%IX1.1=i8088_DI_1	BOOL		Input		WIII	be us	ed ir	n the program.
I E Types	%IX1.2=i8088_D1_2	BOOL		Input					
	%IX1.3=i8088_DI_3 🚤	BOOL		Input					
•	%IX1.4=i8088_DI_4	BOOL		Input					
	%IX1.5=i8088_DI_5	BOOL	Doub	ام مانما		hong	. :+c .c		
	%IX1.6=i8088_DI_6	BOOL	Doub	le-click		nange		ame	
	%IX1.7=i8088_DI_7	BOOL	(Refe	r the <mark>S</mark>	ection	4.1)			
	🖻 👹 %ID2 - i_8084_freq		<b>、</b>			,			
	%ID2.0=i8084_freq0	DINT		uibar					mequency ch0 of i-8084W
	%ID2.1	DINT		Input					
	%ID2.2	DINT		Input					
	%ID2.3	DINT		Input					
	%ID2.4	DINT		Input					
	%ID2.5	DINT		Input					
	%ID2.6	DINT		Input					
	%ID2.7	DINT		Input					
	K Main Variables NewSpy1	10						1.	

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 4-32

#### "PWM\_8088W" Function Block:

Then, using the function block "PWM\_8088W" to control the PWM output for each channel, such as the LD program below.



"1 ~ 65535" (Burst mode), used to output the required pulse count and then stop. **Run:** BOOL, used to start (True) or stop (False) the PWM output.

#### Parameters:

Slot:	The used I/O slot No., and it must be a constant value, not a changeable value.
	In this case, the value is "1". (Data type: DINT)
Chan:	The used I/O channel No., and it must be a constant value, not a changeable value.
	In this case, the value is "0". (Data type: DINT)
Freq:	The PWM output frequency. (Data type: DINT ; Unit: Hz)
	The value can be from 1 to 500,000 (i.e., 1 Hz to 500 KHz) .
	In this case, the initial value is "100" Hz.
Duty:	The PWM output. (Data type: REAL)
	The value can be from 0.001 to 0.999 (i.e., 0.1 % to 99.9 %).
	In this case, the initial value is 0.5 (i.e., 50 %).
Cnt:	The output mode (Data type: DINT)
	Continuous mode: set it as "-1" (in this case) to output PWM pulse continuously.
	Burst mode: it can be from "1" to "65535", to output the required pulse count and then stop.
Run:	Using a BOOL variable to trigger the PWM output. (True: Start ; False: Stop)
State:	The current working status. (Data type: BOOL). (True: working ; False: stopped)

#### Test the program:

Before testing, make sure you have set the PAC's IP address and then compile and download this program to the PAC. (If not familiar with the operation, refer the <u>Section 2.3.4</u> and <u>Section 2.3.5</u>.)

When connecting with the PAC, the SPY List (refer the <u>Section 11.3</u>) will show that the I-8088W's PWM initial frequency ("i8088\_Freq0") is 100 Hz, the initial duty cycle ("i8088\_Duty0") is 50%, using the Continuous mode ("i8088\_Cnt0" = -1) and the currently measured frequency of the I-8084W is 0 Hz.

Workspace		💵 NewSpy1.spl				-			
🖃 — 🕨 demo_8088w (RUN)	B	Name	Value	Description	Y	Name	Value	Туре	
👜 🚞 Exception programs		i8088_Freq0	100	Init = 100 (Hz)		回 🚮 Global va	riables		1
🛓 🛅 Programs		i8088_Duty0	0.5	Init = 0.5 (50%)		i8088 Freq0	100	DINT	
Main	E+	i8088_Cnt0	-1	Init = -1 (Continuous mode)		i8088 Duty0	0.5	REAL	
🛓 🛅 Watch (for debugging)		i8088_Run0	FALSE	TRUE: Start , FALSE: Stop		i8088 Cnt0	-1	DINT	
Soft Scope		i8088_State0	FALSE	Current State		i8088 Run0	FALSE	BOOL	
		i8084_freq0	0	frequency ch0 of i-8084W		i8088 State0	FALSE	BOOL	
NewSpy1		19999999999999999999999999999999999999				RETAIN			
Binding Configuration						🗉 國 %lX1 - i 1	8088		
🚽 🕺 g Global defines						🖃 👹 %ID2 - i			
						%ID2.0=i80	0	DINT	
E Types						%ID2.1	0	DINT	
9999 (B) - 2000 <b>* 1</b> 920						%ID2.2	0	DINT	
						%ID2.3	0 0	DINT	1
	-	<		>		<			

Now, set the "i8088\_Run0" as "TRUE" to start the PWM output. At this time, the "i8088\_State" will change from "FALSE" to "TRUE" and output a PWM signal to the I-8084W module, and then the value of the "i8084\_Freq0" will change from 0 Hz to 100 Hz.

Vorkspace		) 🖪 NewSpy1.sp	0					
∃ 🕨 demo_8088₩ [RUN]		Name	Value	Description	7	Name	Value	Туре
🗄 🚞 Exception programs		i8088_Freq0	100	Init = 100 (Hz)		🗄 🚮 Global va	riables	
📩 🔤 Programs		i8088_Duty0	0.5	Init = 0.5 (50%)		i8088_Freq0	100	DINT
Main	∎+	i8088_Cnt0	-1	Init = -1 (Continuous mode)		i8088_Duty0	0.5	REAL
🗄 🛁 Watch (for debugging)		i8088_Run0	TRUE	TRUE: Start , FALSE: Stop		i8088_Cnt0	-1	DINT
Soft Scope		i8088_State0	TRUE	Current State		i8088_Run0	TRUE	BOOL
📰 Initial values		i8084_freq0	100	frequency ch0 of i-8084W		i8088 State0	TRUE	BOOL
NewSpy1				i8088 Run0		RETAIN	variables	
						🖽 👹 %IX1 - i_	8088	
				TRUE	1)	🖃 👹 %ID2 - i	8084_freq	
🚽 📸 Variables				C		%ID2.0=i80	100	DINT
E Types				FALSE	0)	%ID2.1	0	DINT
						%ID2.2	0	DINT
		0.271		Lock		%ID2.3	0	DINT
		<				<	27.222	>
	* 8	Main Variable	New Spy 1	Unlock				
	Build							

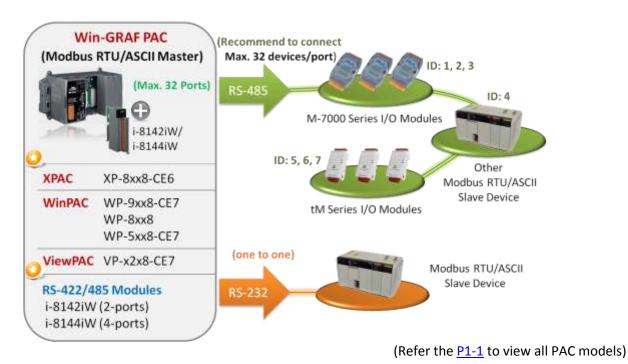
If set the "i8088\_Cnt0" as 500 (Burst mode), the "i8084\_Freq0" value will become 0 after the I-8088W outputs 500 PWM pulses. You can try to change the "i8088\_Cnt0" value and then set the "i8088\_Run0" as "TRUE" to view the changes of output.

# **Chapter 5** Modbus Master: connecting to Modbus Slave Devices

This chapter lists the way to enable the Win-GRAF PAC as a Modbus Master to connect Modbus RTU/ASCII Slave or Modbus TCP/UDP Slave devices. If you want to use one XV board in the WP-5xx8-CE7, refer the <u>Section 5.1.6</u> to <u>Section 5.1.11</u>.

# 5.1 Enabling the Win-GRAF PAC as a Modbus RTU/ASCII Master (I/O & XV-board)

**Application Diagram:** 



#### Follow these steps:

1. Mouse click the "Open Fieldbus Configuration" tool button to open "IO Drivers" window.

Wn Win-GRAF - Test_01	
<u>F</u> ile <u>E</u> dit <u>V</u> iew Insert Project	Lools <u>W</u> indow <u>H</u> elp
😂 🔒 🖹 🎒 🔏 🖡 🛍	🗙 👞 🏷 (연) 🛗 🖩 🏭 🚱 💁 🖻 🍐 (연) 🔛 🕍 🔊
Workspace	IO Drivers
⊡ 🗊 Test_01	Name Value 🝸 Name Type
🗄 🚞 Exception programs	🚮 Global variables 🦉
🗄 🚞 Programs	DETAIN variables
🗗 Main	*
🚊 🚞 Watch (for debuggi	🗁 📲 pOnBadindex
Soft Scope	📲 pOnDivZero 💊
🔤 Initial values	
📟 📆 Binding Configuration	Name Value
🔤 😽 🚽 Global defines	Č.
🚽 🚰 Variables	
E Types	Build X
	🔹 🗲 Build / Cross references / Runtime / Call stack / Breakpoints / Digital sampling trace / Promp

2. Click the "Insert Configuration" button on the left of the "IO Drivers" window, then click the "MODBUS Master" and "OK" to enable the Modbus Master setting.

IO Drivers							×
居	1	Name Value	$\mathbf{T}$	Name		Туре	
温					Global va	riables	^
					RETAIN	variables	
* <b>1</b>	Add Configuration						×
	Choose a configuration					0K 🍗	ר
							ς Ι
	MODBUS					Cancel	
ej.	MODBUS Master	<b>•</b>					
	MODBUS Slave						

3. Click the "Insert Master/Port" button on the left side to open the setting window. Then, select the "Serial MODBUS-RTU", set COM Port (e.g., "COM2:9600,N,8,1") and Delay time (recommended value: 10 ms, it can be 0 to 10000), and then click "OK".

IO Drivers *				×
Hand Model Master	Name Value	🝸 Name	Туре	
묘		🔄 🚮 Glob	al variables	^
MODBUS Master Port		🚨 🛛 🛃 RET.	AIN variables	
and the second sec		ү 📄 Mair	1	=
O MODBUS on Ethernet		🚽 📲 pOni	Badindex	
Address:	Cancel	🔪 📲 pOni	DivZero	*
Port: 502			>	
		ame	Value	
Protocol: TCP - Open MODBUS UDP - MODBUS RTU			1	
UDP - Open MODBUS	Maste	to set it as a r, change the COM2:9600,	setting to	
Delay between requests	After r	receiving the	respond, wai	iting for
Delay (ms): 10	10 ms	to send the r	next comman	nd
Try to reconnect after communication er Manage diagnostic info for slaves	Select	"Disabled" if this COM Po	•	want
Disabled (do not open and manage this p	port)			

4. Click the "Insert Slave/Data Block" button on the left side to create a data block.

10	Drivers		
臣	⊡…M <sub>p</sub> MODBUS Master	Name	Value
묾		Mode	RTU
		Address	COM2:9600,N,8,1
*		Port	502
		Reconnect after error	<ul> <li>Image: A set of the set of the</li></ul>
		Slaves diagnostics	✓
		Delay between requests (ms)	10
		Disabled	

This table lists five data blocks, and each data block stands for one Modbus Master Request.

Item	Function Code	Modbus Request	Description
<u>1</u>	2	Read Input-bits	Read DI data
<u>2</u>	5	Write single coil-bit	Write DO data
<u>3</u>	4	Read Input Registers	Read AI data
<u>4</u>	6	Write single holding register	Write one AO data (16-bit)
<u>5</u>	16	Write Holding Registers	Write multiple AO data (16/32 bits)

Note: If you want to disable the Modbus RTU/ASCII Master port while the program is running, refer the Section 5.1.13 to use the "MBRTU\_M\_disable" function.

## 5.1.1 Read DI data

1. Completing all the following settings in the "MODBUS Master Request" window, and then click "OK".

M	ODBUS Master	Request			
	Request Description: Slave/Unit:	1	a		OK Cancel
	MODBUS Reques	t.			
	<1> Read Coil <2> Read Inpu <3> Read Hold	ut Bits ling Registi	ers		(b)
c	Data block		~		
	Base <u>a</u> ddress: <u>N</u> b items:	1			
d	Activation				
		2000	ms	[15000	
	On change	•		This optior	n can not apply
$\neg$	Misc.	Terrer		to the "Rea	ad" request.
ب	<u>T</u> imeout:	250	ms		
	Nb trials:	1			

- d. Activation: The way to send the Modbus request.
- <u>Periodic</u>: Sending the request periodically.
   (In this case, to send once every two seconds.)
   "on error" means the next sending time when an exception occurred (e.g., 15 seconds).
- On call: The request is activated when a program call to send it

a. <u>Slave/Unit</u>:

Enter the Net-ID of the Slave device. (In this case, the Net-ID is "1").

- b. <u>MODBUS Request</u>: Select "<2> Read Input Bits" option.
- c. <u>Base address</u>: Start from "1" by default. <u>Nb items</u>: The number of DI signals to read. (In this case, the number is "16").

## <u>Note:</u>

If you want to change the "Base address", right-click the "MODBUS Master" and then select the "MODBUS Master Addresses" to modify the value.

IO Drivers		
MODBUS Master add	resses	
First valid MODBUS a	ddresses	ОК
Input <u>b</u> its:	1	Cancel
<u>C</u> oil bits:	1	
Input <u>r</u> egisters:	1	
Holding registers:	1	

- On change: In case of a write request, means that the request is activated each time any variable changed.
- e. <u>Timeout</u>: Set a timeout value. (When time-out occurred, it will show the defined error code.) (The recommended value for the Modbus RTU/ASCII device is 200 to 1000 ms. E.g., 250 ms)

2. Next, open the "Variables" window and then declare variables that are available for the program.

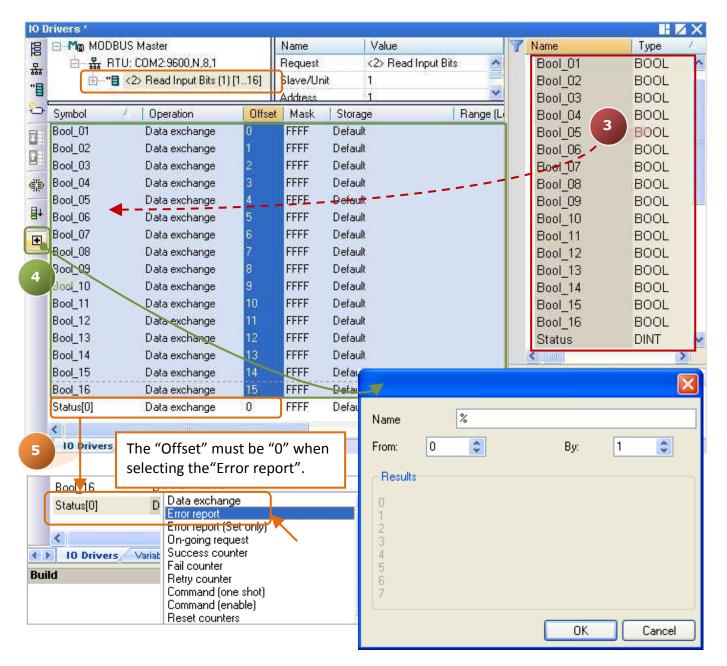
Workspace	10 Drivers			
⊡ 🗿 Test_01		BUS Master	Name	Value
🗄 🗁 🚞 Exception programs		TU: COM2:9600,N,8,1	Request	<2> Read Input Bits
🛓 🛄 Programs	· · · · · ·	<2> Read Input Bits (1) [116]	Slave/Unit	1
🛄 편 Main	*1		Address	1
🗄 🛁 Watch (for debuggi			Nb Item	16
Soft Scope	Press "F1"		Activation	Periodic
	123	key to view the details	Period (ms)	2000
	on the MC	DBUS Master settings.	Period on error	15000
🚽 🛐 🛛 🚽 🚽 🚽	ġį <mark>p</mark>		Timeout (ms)	250
🚮 Variables 🛌			Number of trials	1
E Types	₽t		Description	
Double click it to open t	he window.	Operation	Offset	Mask
	<	100		>

Declaring 16 variables to read data (Name: "Boo\_01 to Boo\_16"; Type: BOOL) and one array variable to record the state of data access (Name: "Status"; Dim.: 5; Type: DINT). Refer the <u>Sectin 2.3.1</u> for the way to declare variables, and the figure below shows defined variables.

Va	riables								H 🛛 X	
7	Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description	
	🗉 🚮 Global v	ariables							<u>^</u>	
	Bool_01	BOOL								
	Bool_02	BOOL								
	Bool_03	BOOL								
	Bool_04	BOOL								
	Bool_05	BOOL								
	Bool_06	BOOL					Deel 1		BOOL	
	Bool_07	BOOL					Bool_1		BOOL	
	Bool_08	BOOL					Status	;	DINT	
	Bool_09	BOOL								5
	Bool_10	BOOL								
	Bool_11	BOOL								7
	Bool_12	BOOL								
	Bool_13	BOOL								
	Bool_14	BOOL					Bool 1	16	BOOL	
	Bool_15	BOOL					_		DINT	IO 41
	Bool_16	BOOL					Status	6	DINT	[04]
	<		Ш							
-	IO Drivers 2	Variables								

- 3. In the "IO Drivers" window like the figure below, drag all required variables in the Variables Area (i.e., "Bool\_01" to "Bool\_16" and "Status") and drop them to the "Symbol" area in the first data block.
  <u>Note:</u> The "Status" is an array variable, so, the Status[0] to Status[4] will show on the "Symbol" area.
  Click the "Del" key to delete the Status[1] to Status[4].
- Next, select "Offset" field from "Boo\_01" to "Boo\_16" and then click the "Iterate Property" button on the left side to set the "Offset" value (From: "0"; By: "1", refer the <u>Section 3.1</u> – Step8).

5. In the "Operation" field, set the "Status[0]" as "Error report" which means the return value is an "Error Code" if a read error occurred and the value will be reset to "0" if read successfully.



#### You can also press "F1" in this "IO Drivers" window to see details on Modbus Master Configuration.

Error Code	Description	Error Code	Description
0	The communication is OK.	8	Data Parity Error.
1	MODBUS function not supported.	10	Invalid gateway path.
2	Invalid MODBUS address.	11	Gateway target failed.
3	Invalid MODBUS value.	128	Communication timeout.
4	MODBUS Server failure.	129	Bad CRC16.
6	Server is busy.	130	RS-232 communication error.

# 5.1.2 Write DO Data

1. Using the same way in the <u>Section 5.1</u> - Step 4 to create the second data block and completing all the following settings in the "MODBUS Master Request" window, and then click "OK".

MODBUS Master Request	In this example
Request	a. <u>Slave/Unit</u> :
Description:	Enter the Net-ID of the Slave device.
Slave/Unit: 2 a Cancel	(e.g., the Net-ID is "2").
MODBUS Request	b. <u>MODBUS Request</u> :
	Select "<5> Write single coil bit".
<4> Read Input Registers	c. <u>Base address</u> :
<5> Write single coil bit 26 S Write single bolding register	Start from "1" by default.
Data block	(Refer the <u>Section 5.1.1</u> to change it.)
Base address: 1	d. <u>On call</u> :
Nb items;	The request is activated when a
	program call to send it
Activation	(Refer the <u>Section 5.1.1</u> for details)
	e. <u>Timeout</u> : Set a timeout value.
On call (on error)	When time-out occurred, it will show
Misc	the defined error code. (The
e Timeout: 250 ms	recommended value for the Modbus
	RTU/ASCII device is 200 to 1000 ms.
Nb trials:	In this case the value is 250 ms.)

2. Next, open the "Variables" window and then declare variables that are available for the program.

Norkspace	IO Drivers *		
🗆 🗐 Test_01	📔 🖃 Mo MODBUS Master	Name	Value
👜 🚞 Exception programs	용 E 문 RTU: COM2:9600,N,8,1	Request	<5> Write single coil bit
🗄 — 🚞 Programs	***  *	Slave/Unit	2
🛄 📴 Main	The single coil bit [2] [11]	Address	1
🚊 🔤 Watch (for debuggi		Nb Item	1
Soft Scope	Tinc	Activation	On Call
		Period (ms)	0
📲 😼 Binding Configuration	Press "F1" key to view the details	Period on error	0
🔤 😽 🚽 🚽 🚽	on the MODBUS Master settings.	Timeout (ms)	250
🚽 📸 Variables 🥿		Number of trials	1
E Types	<b>∃</b> +	Description	

Double click it to open the window.

Add two boolean variables in the "Variables" window (refer the <u>Section 2.3.1</u> for declaring variables).

Variable name	Data type	Description
DO_0	BOOL	Used to Write digital output data.
Act_0	BOOL	In this case, choose the "On call" way to write data that means using a variable to call it.

After completing the settings, the defined variables show as below:

Name	Туре	D.7	Attrib.	Syb.	Init value	User	Tag	Description
DO_0	BOOL		11.1		500	1		
Act_0	BOOL							
<								>

- 3. In the "IO Drivers" window, drag variables "DO\_0", "Act\_0" and "Status" (that created in the <u>Section 5.1.1</u>) from the Variables Area to the Symbol Area in the second data block.
  <u>Note:</u> The "Status" is an array variable. When you drag "Status" into the Symbol Area, it will show "Status[0]" to "Status[4]", simply press "Del" key to delete "Status[0]" and "Status[2] to [4]".
- 4. Set the "Operation" field of the "Status[1]" as "Error report" (that means this variable will be set to an error code when a read error occurs, or reset it to "0" when a read request is successful). Press the "F1" key to see the description of the Modbus Master Configuration and move to the title "Status and command variables" to know related commands and error codes.
- Set the "Operation" field of "Act\_0" as "Command (one shot)" (that means the request will be sent only once when "Act\_0" is set to "TRUE". Then, this "Act\_0" will auto reset to "FALSE").
   The "Command (Enable)" means the request is sent continuously as long as the "Act\_0" is "TRUE".
   So, users can set the "Act\_0" to "FALSE" to stop sending command.

0 0	lrivers *								HZ	
	🖃 Modbus Master	N	lame		Value	Y	Name	Туре	Dim.	A
뮮	由 品 RTU: COM2:9600,N,8,1	B	lequest		<5> Write single coil bit		Bool_11	BOOL		1
	💼 📲 <2> Read Input Bits (1) [11	6] S	lave/Un	it	2		Bool_12	BOOL		
"	👜 📲 <5> Write single coil bit (2) [	[1] A	ddress		1		Bool_13	BOOL		
0		N	lb Item		1		Bool_14	BOOL		
		A	ctivation	8	On Call		Bool_15	BOOL		
		P	eriod (m:	5)	0		Bool_16	BOOL		
		P	eriod on	error	0		Status	DINT	[04]	
ŝį,		T	imeout (i	ns)	250		DO_0 •	BOOL		
200		N	lumber o	f trials	1		Act 0	BOOL		
∎+		D	escriptio	n			<1-1			8
E	Symbol Operation	Offset	Mask	Stor	age Ran	Na	me	Value		
		Contraction of the	24.000 (A.C.	-Defa						_
	(HARDER) (HA		FFFF	Defa						
ſ			FFFF	Defa						
	Data exchange			~	>					
<	Error report					110				
	Error report (Set only)									
	Success counter									
	Fail counter			=						
	Retry counter									
	Command (one shot) Command (enable)									
	Reset counters	$\sim$		_						
	Slave: last error									
	Slave: last error date	stamp								
	Slave: last error time	stamp		-						
	Slava: last rasat data	otamn		~						

### 5.1.3 Read AI Data

1. Using the same way in the <u>Section 5.1</u> - Step 4 to create the third data block and completing all the following settings in the "MODBUS Master Request" window, and then click "OK".

MODBUS Master Request	In this example
Request     OK       Description:	a. <u>Slave/Unit</u> : Enter the Net-ID of the Slave device. (e.g., the Net-ID is "3").
MODBUS Request	b. MODBUS Request:
<2> Read Input Bits	Select "<4> Read Input Registers".
<3> Read Holding Registers	d. <u>Base address</u> :
Data block	Start from "1" by default.
Base address: 1	(Refer the <u>Section 5.1.1</u> to change it.)
Nb items: 10	<u>Nb items</u> :
	The number of AI signals to write.
Activation	(In this case, the number is "10").
d         O Periodic:         1000         ms         15000           O On call         (on error)         (on error)	d. <u>Periodic</u> : (Refer the <u>Section 5.1.1</u> )
O on change This option can not apply to the "Read" request.	Sending the request periodically. (In this case, to send once per second.)
e <u>Timeout</u> : 250 ms	"on error" means the next sending
Nb trials: 1	time when an exception occurred
	(e.g., 15 seconds).

e. <u>Timeout</u>: Set a timeout value.

When time-out occurred, it will show the defined error code. (The recommended value for the Modbus RTU/ASCII device is 200 to 1000 ms. In this case the value is 250 ms.)

2. Next, open the "Variables" window and then declare variables that are available for the program.

Workspace	IO Drivers *								
⊟ 🗿 Test_01	📙 🖃 Mo MODBUS Master	Name	Value						
🗄 🛅 Exception programs	品 品 RTU: COM2:9600,N,8,1	Request	<4> Read Input Registers						
🗄 — 📴 Programs	□	Slave/Unit	3						
🛄 🔟 Main	*	Address	1						
📩 🛁 Watch (for debuggi	* * * • • • • • • • • • • • • • • • • •	Nb Item	10						
Soft Scope	FF X	Activation	Periodic						
🛅 Initial values	FILT .	Period (ms)	1000						
	Tips:	Period on error	15000						
🚽 🖇 🚽 🚽 🚽 🚽	Press "F1" key to view the details	Timeout (ms)	250						
	— on the MODBUS Master settings.	Number of trials	1						
E Types	<b>∎</b> +	Description							

Follow the table below to add six Word (16-bit), one Double integer (32-bit) and one Real (32-bit) variables. (Refer the <u>Section 2.3.1</u> for declaring variables).

Variable name	Data type	Description
Word_1 to Word_6 WORD		Used to Read AI data (16-bit).
Long_1 DINT		Used to Read AI data (32-bit).
Real_1	REAL	Used to Read AI data (32-bit).

Refer the <u>Appendix A</u> for details on data type and ranges. After completing the settings, the defined variables show as below:

Name	Туре	D.7	Attrib.	Syb.	Init value	User	Tag	Description
Long_1	DINT					-		1
Word 1	WORD							
Word 2	WORD							
Word 3	WORD							
Word 4	WORD							
Word 5	WORD							
Word 6	WORD							
Real_1	REAL							
<								2

- 3. In the "IO Drivers" window, drag variables "Word\_1 to Word\_6", "Long\_1", "Real\_1" and "Status" (that created in the <u>Section 5.1.1</u>) from the Variables Area to the Symbol Area in the third data block. <u>Note:</u> The "Status" is an array variable. When you drag "Status" into the Symbol Area, it will show "Status[0]" to "Status[4]", simply press "Del" key to delete "Status[0] to [1]" and "Status[3] to [4]".
- 4. Set the "Operation" field of the "Status[2]" as "Error report" (that means this variable will be set to an error code when a read error occurs, or reset it to "0" when a read request is successful). Press the "F1" key to see the description of the Modbus Master Configuration and move to the title "Status and command variables" to know related commands and error codes.
- 5. Both the "Long\_1" and the "Real\_1" are 32-bit variables and require two Modbus addresses. So, set their "Storage" column as "DWORD (Low High)".

	BUS Master	Name	Value	7 Name		🔺 Туре	Dim
🗄 🐰 I	RTU: COM2:9600,N,8,1	Request	<4> Read Input Regis	DO	0	BOOL	~
<u>ج</u> •	🔋 <2> Read Input Bits (1) [1.	.16] Slave/Unit	3	Long	1	DINT	11/1
ġ•	📳 <5> Write single coil bit (2)	[1 Address	1	Rea	[]	REAL	
· (E)	😫 <4> Read Input Registers	3) [ Nb Item	10	Stat	us	DINT	[C
		Activation	Periodic	Wor	d 1	WORD	P
		Period (ms)	1000	Wor	d_2	WORD	
		Period on err	15000	Wor	d 3	WORD	
		Timeout (ms)	250	Wor	d 4	WORD	
		Number of tri	1	Wor	d_5	WORD	
		Description	3	Wor	d_6	WORD	
Symbol	Operation	Offset   Mask   9	Storage	the second se		N variables	~
Word_4	Data exchange	0 FFFF D	efault	<	Main		>
Word_5	Data exchange	0FEFF D	efault	Name	-1:	Value	1
Word_6	Data exchange	0 FFFF D	efault 📉 📩	Name		value	-
Long_1	Data exchange	0 FFFF D	WORD (Low - High)				
Real_1	Data exchange	0 FFFF D	WORD (Low - High)				
Status[2]	Error report	D FFFF D	efault 🖉				
1	4		5				

6. As the figure below, select the "Word\_1" to "Word\_6" and then click "Iterate property" to set their Offset value (From: 0; By: 1).

🗄 Symbol	Operation	Offset	Mask Storage
Word_1	Data exchange	0	FFFF Default
Iterate property	Data exchange	0	F
Word_3	Data exchange	0 0 0 0	F
Word_4	Data exchange	0	F Name %
Word_5	Data exchange	0 0	From: 0 🗢 By: 1 🗘
Word_6	Data exchange	0	
<		in the second	Results
♦ 10 Drivers	Variables		0
			2
			2 3 4 5 6 7
			6
			[7
			OK 🔪 Cancel

7. Next, double click the Offset field of "Long\_1" and "Real\_1" items and set their values as "6" and "8", then press "Enter" key to complete the settings.

Note: One 32-bit data requires two Modbus addresses. For instance, the Offset value of "Long\_1"

is "6" and the next Offset value must be set to "8" (i.e., "Real\_1").

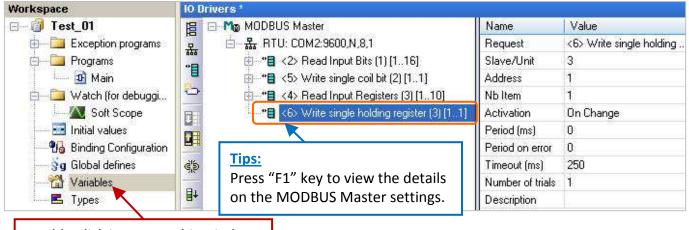
Symbol	Operation	Offset	Mask	Storage	
Word_4	Data exchange	3	FFFF	Default	~
Word_5	Data exchange	4	FFFF	Default	
Word_6	Data exchange	5	FFFF	Default	
Long_1	Data exchange	6	FEEE	DWORD (Low - High)	
Real_1	Data exchange	8 8	A V	DWORD (Low - High)	
Status[2]	Error report	0	CCCC.	Default	~
<		1	press	"Enter"	

# 5.1.4 Write AO Data (16-bit)

1. Using the same way in the <u>Section 5.1</u> - Step 4 to create the 4th data block and completing all the following settings in the "MODBUS Master Request" window, and then click "OK".

MODBUS Master Request 🛛 🛛 🔀	In this example
Request   Description:   Slave/Unit:   3   a     Cancel     MODBUS Request     <4> Read Input Registers     <5> Write single coil bit   <5> Write single coil bit   <5> Write single coil bit     <6   Base address:   1     Nb items:     0   0 n call   (on error)     Misc.   e   Timeout:   250   Ms   Nb trials:	<ul> <li>a. <u>Slave/Unit</u>: Enter the Net-ID of the Slave device. (e.g., the Net-ID is "3").</li> <li>b. <u>MODBUS Request</u>: Select "&lt;6&gt; Write single holding register".</li> <li>c. <u>Base address</u>: Start from "1" by default. (Refer the <u>Section 5.1.1</u> to change it.)</li> <li>d. <u>On change</u>: In case of a write request, means that the request is activated each time any variable changed. (Refer the <u>Section 5.1.1</u> for derails.)</li> <li>e. <u>Timeout</u>: Set a timeout value. When time-out occurred, it will show the defined error code. (The recommended value for the Modbus RTU/ASCII device is 200 to 1000 ms. In this case the value is 250 ms.)</li> </ul>
	this case the value is 250 ms.

2. Next, open the "Variables" window and then declare variables that are available for the program.



Double-click it to open this window.

Declaring a "WORD" variable.

(Refer the <u>Appendix A</u> for details on data type and ranges ; refer the <u>Section 2.3.1</u> for operations).

Variable name	Data type	Description
Word_Write_1	WORD	Used to write AO data (16-bit).

After completing the settings, the defined variables show as below:

Name	Туре	D.7	Attrib.	Syb.	Init value	User	Tag	Description
Word_Write_1	WORD					10.0		

- In the "IO Drivers" window, drag variables "Word\_Write\_1" and "Status" (that created in the <u>Section 5.1.1</u>) from the Variables Area to the Symbol Area in the 4th data block.
   <u>Note:</u> The "Status" is an array variable. When you drag "Status" into the Symbol Area, it will show "Status[0]" to "Status[4]", simply press "Del" key to delete "Status[0] to [2]" and "Status[4]".
- 4. Set the "Operation" field of the "Status[3]" as "Error report" (that means this variable will be set to an error code when a read error occurs, or reset it to "0" when a read request is successful). Press the "F1" key to see the description of the Modbus Master Configuration and move to the title "Status and command variables" to know related commands and error codes.

Drivers *	Martin		1	N	1 Value			~ 1	
·⊟ Modbus 白 品 RTU:	Master COM2:9600,N,8,1			Name Request	Value <6> Write sin	ala haldi	Name	obal variab	Туре
<b>⊡*</b> ∎ <	2> Read Input Bits (1)	1740 C C C C C C C C C C C C C C C C C C C		Slave/Unit	3	gie noiai	Word_\	And the second	WORD
Ē	5> Write single coil bi	: (2) [11]	1000	Address	1		VVord_t	2	WORD
Ē <b>∵"≣</b> <•	4> Read Input Regist	ers (3) [11)	וו	Nb Item	1		Word_5	5	WORD
	6> Write single holdin	<mark>g register</mark> (3	)[11]	Activation	On Change		Word_4	4	WORD
				Period (ms)	0		Word_3	3	WORD
				Period on err	0		Word 2	2	WORD
				Timeout (ms)	250		Word 1		WORD
				Number of tri	1		Status		DINT
				Description					
Symbol	Operation	Offset	Mask	Storage		Range (Lc	Name	Value	•
Word_Write_1	Data exchange		FFFF	Default					
Status[3]	Error report	0	FFFF	Default					
<	100		1			5			

# 5.1.5 Write AO Data (32-bit)

1. Using the same way in the <u>Section 5.1</u> - Step 4 to create the 5th data block and completing all the following settings in the "MODBUS Master Request" window, and then click "OK".

Request	124			
Description:				ОК
<u>S</u> lave/Unit:	4	a		Cance
MODBUS Reques	t			
<6> Write sing		ı register	~	$\bigcirc$
<15> Write Co		isters		ь
		102003010		
Data block	-			
Base <u>a</u> ddress:	1			
Nb items:	2			
	1			
Activation	a <del>.</del>		(i)	
O Periodic:	0	ms	0	
On call			(on error)	
• On change				
Misc.	7	-		
Timeout:	250	ms		

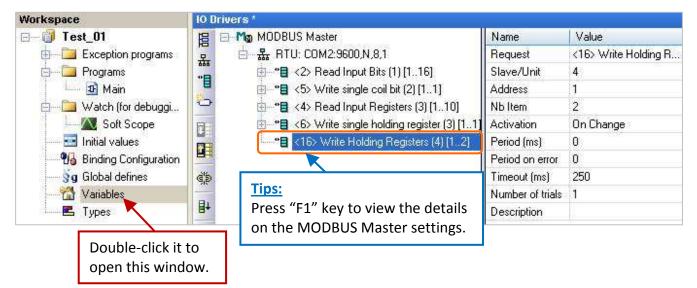
#### In this example

- a. <u>Slave/Unit</u>:
  Enter the Net-ID of the Slave device.
  (e.g., the Net-ID is "4").
- MODBUS Request: Select "<16> Write Holding Registers".
- c. <u>Base address</u>: Start from "1" by default. (Refer the <u>Section 5.1.1</u> to change it.) <u>Nb items</u>: The number of AO signals to write. (In this case, the number is "2" because the REAL type requires two Modbus address).
- d. <u>On change</u>: In case of a write request, means that the request is activated each time any variable changed.
  (Refer the <u>Section 5.1.1</u> for details)

e. <u>Timeout</u>: Set a timeout value.

When time-out occurred, it will show the defined error code. (The recommended value for the Modbus RTU/ASCII device is 200 to 1000 ms. In this case the value is 250 ms.)

2. Next, open the "Variables" window and then declare variables that are available for the program.



Declaring a "Real" variable.

(Refer the <u>Appendix A</u> for details on data type and ranges ; refer the <u>Section 2.3.1</u> for operations).

Variable name	Data type	Description
Real_Write_1	REAL	Used to write AO data (32-bit).

After completing the setting, the defined variable shows as below:

Name 🛛	Туре	D.7	Attrib.	Syb.	Init value	User	Tag	Description
Real_Write_1	REAL							
<								>

- In the "IO Drivers" window, drag variables "Real\_Write\_1" and "Status" (that created in the <u>Section 5.1.1</u>) from the Variables Area to the Symbol Area in the 5th data block.
   <u>Note:</u> The "Status" is an array variable. When you drag "Status" into the Symbol Area, it will show "Status[0]" to "Status[4]", simply press "Del" key to delete "Status[0] to [3]".
- 4. Set the "Operation" field of the "Status[4]" as "Error report" (that means this variable will be set to an error code when a read error occurs, or reset it to "0" when a read request is successful). Press the "F1" key to see the description of the Modbus Master Configuration and move to the title "Status and command variables" to know related commands and error codes.
- 5. The "Real\_Write\_1" is a 32-bit data and required two Modbus addresses. So, set its "Storage" field as "DWORD (Low High)".

MoDBUS Master	Name	Value	4	🝸 Name	Туре
	Request	<16> Write Holdi	ng R	Bool 15	BOOL
□	Slave/Unit	4		Bool_16	BOOL
	Address	1		DO_0	BOOL
P	Nb Item	2		Long_1	DINT
🗄 🐨 🗄 <6> Write single holding register (3) [11]	Activation	On Change		Real_1	REAL
	Period (ms)	0		Real_Write_	1 REAL
	Period on err	0		Status	DINT
é <mark>.</mark>	Timeout (ms)	250		Word_1	WORD
	Number of tri	1	1	Word_2	WORD
+	Description			10/aud 3	
Symbol / Operation Offset Mask-	Storage	R	ange (Lc	Name	Value
Real_Write_1  Data exchange 0 FFFF	DWORD (L	ow - High)			
Status[4] Error report 0 FFFF	Default				
<			>		

# 5.1.6 How to use the XV Board?

The XV board belongs to the Modbus Slave I/O board. Before using the I/O board, users must plug it into the WP-5xx8-CE7, and then enable the WP-5xx8-CE7 as a Modbus Master (refer the <u>Section 5.1</u>). Visit the XV board Selection Guide page to get more details: <u>www.icpdas.com/root/product/solutions/hmi\_touch\_monitor/touchpad/xv-board\_selection.html</u>

All the Win-GRAF demo projects listed in the following table can be found on the CD-ROM. Refer the <u>Chapter 12</u>, click the Win-GARF menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project and to view the details. (CD-ROM:\Napdos\Win-GRAF\demo-project\)

Demo File Name		Description			
XV107, XV107A	demo_XV107.zip	Read 8 DI, Write 8 DO			
XV110 demo_XV110.zip		Read 16 DI			
XV111, XV111A	demo_XV111.zip	Read 16 DO, Read 1 DO			
XV116	demo_XV116.zip	Read 5 DI, Write 6 D0			
XV308_1	demo_XV308_1.zip	1. Read 8 Al, Read 8 DI			
XV308_2	demo_XV308_2.zip	2. Read 8 AI, Write 8 DO			
XV308_3	demo_XV308_3.zip	3. Read 8 AI, Write 4 DO, Read 4 DI			
XV310	demo_XV310.zip	Read 4 AI, Write 4 DO, Read 4 DI, Write 4 AO			

#### Common setting:

1. Mouse click the "Open Fieldbus Configuration" tool button to open the "I/O Drivers" window.

🚾 Win-GRAF - demo_XV107					
<u>File Edit V</u> iew Insert Project	<u>I</u> ools <u>W</u> indow <u>H</u> elp				
8 3 3 3 4 X 3 B	× 🚽 🤊 🕑 擶 🎹 🏭 😘 🔮 🕻	🖞 🖻 🛐 🖪 🖀 s	0)		
Workspace	IO Drivers*			10	OK Cancel
demo_XV107      Exception programs     Programs     Main     Watch (for debugging)     Soft Scope	Image: Constant of the second seco	Name           XV107_DI_00           XV107_DI_01           XV107_DI_01           XV107_DI_02	BOOL A		
Initial values ∰ Binding Configuration ∰ Global defines ∰ Variables ∰ Types	Request Slave/Unit A (2) Read Input Bits 1 1 (15) Write Coil Bits 1 1 Build	○ MODBUS on Address: Port: Pr <u>o</u> tocol:	If using it as ASCII Mast	s the Modbus er, modify it to 40:115200,N,8,	Cancel
open the "MODB <u>Note:</u> All the demo proj enable the WP-5>	RTU: COM:115200,N,8,1" to US Master Port" window. ects listed in above table xx8-CE7 as a Modbus <b>RTU</b> d set the "Com. Port" as <b>J,8,1".</b>	⊙ <u>5</u> erial MODB <u>C</u> om. port: Delay between r <u>D</u> elay (ms):	COM0:115200,N,8	,1	

#### **Configure the AI/AO channel**

If you want to use the AI/AO channel of the XV Board (e.g., XV308, XV310) in the WP-5xx8-CE7. First, stop the Win-GRAF driver on the PAC and then configure each AI/AO channel by using "DCON\_Utility\_Pro\_CE\_200.exe".

#### Using the WP-5238-CE7 as an example:

1. Click the "Win\_GRAF\_WP\_5238" (or the small icon on the taskbar) to open the Win-GRAF driver window, and then click the "End Driver" button.

My Davice RegEdt	
internet tasimanager	Win-GRAF-WP-5238
	WP-5238 driver Version 1.01 , Jul.16,2014
kgw35 Win_GRAF_ WP_5238	This product is licensed.
Morecolt WordPad	·····
My Documents	Elapsed time : 0. 0: 0:19
(reclamy)	Stop the Win-GRAF driver
Start V Virtual CE	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

2. Click "My Device" on the desktop and then get into the path "\System\_Disk\Tools\dcon\_utility\_pro" to run the "DCON\_Utility\_Pro\_CE\_200.exe".

<u>File E</u> dit <u>V</u> iew <u>G</u> o F <u>a</u> vor	rites 🛛 🔷 🔹 🗶 😭 🛛	<b>Ⅲ</b> • ×
Address System_Disk\Tools\dcon	_utility_pro	-
🗁 auto_config	Cmd_config	🗁 language
🗁 log_report	🗁 remote_config	🗁 system
🔊 CE7_PlatformBase_arm.dll	🔊 CmdParser.dll	🔊 CommuIO.dll
DCON_Utility_Pro_CE_200.exe	🔊 IOModule.dll	🔊 PACNET.dll
Netform.dll	🔊 Protocol.dll	🔊 Utility.dll

3. Click the COM Port button to set the "COM Port" as "COM0", set the "Baud Rate" as "115200" and set the "Format" as "N,8,1", and then click "OK".

DCON U	tility Pro 2	.0.0.0 for CE7	arm(WP523	1)		×			
1		II 🛠		1	?				
Start A	daress	0 End	Address	1	COM Port Option				×
ID	Addr	Baud Rate	Checks	Format St	COM Pa	ort	Timeout		
	NY 13	60°		10 1344	COM0[Backplan	e]	200	ms	
					Baud Rate Pr	otocol Checks	sum Forma	ıt	
					© N,8,1	<b>O</b> N,8,2	<b>O</b> E,8,1	<b>O</b> 0,8,1	
						<b>\</b>			
4					ОК	Cancel			

4. After clicking the Search button, the XV Board (e.g., XV308) will show in the window. Then, double click this item to get into the setting window.

DCON Utility Pro 2.0.0.0 for CE7 arm(WP5231)							
II 🛠			2 ?				
	l Address	1					
Baud Rate	Checks	Format Sta	itus Description				
115200	Disable	N,8,1	[Modbus RTU]8*AI + 8 Uni				
	0 Enc	D End Address Baud Rate Checks 115200 Disable	Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constraint of the second state       Image: Constraint of the second state         Image: Constrai				

5. In the "Configuration" tab, set the "Data Format" as "Engineering Format" (recommended setting), set the "Fast Mode" as "Fast Mode" and uncheck the "Reverse DI State (INIT\*)".

	XV308 Firmware[A102]									
4	Configuration AI/A	Alarm DO/Alarm Host WDT DI About								
	Protocol(INIT*)	Modbus RTU								
	Address	1 × 01H								
	Baud Rate(INIT*)	115200								
	Parity(INIT*)	N,8,1-None Parity								
	Checksum(INIT*)	Disable								
	Data Format	Engineering Forn								
	Fast Mode	Fast Mode								
		<b>`</b>								
	Response Delay									
	Reverse DI State									
		Set Module Configurations								

6. In the "AI/AI Alarm" tab, to configure the proper ranges and values for each AI channel, and remember to select any AI channel (e.g., "CH:00") you want to use, then click the "Set Alarm" button.

XV308 Firm	XV308 Firmware[A102]									
Configurati	Configuration AI/AI Alarm DO/Alarm Host WDT DI About									
High Alarm Limit Low Alarm Limit Alarm Mode										
CH:00	-00003 [-000.003]	[08] +/- 10 V 💌	10	-10	Disable 💌					
CH:01	-00004 [-00.0040]	[09] +/- 5 V 💌	5	-5	Disable 💌					
CH:02	-00052 [-00.0052]	[05] +/- 2.5 V 💌	2.5	-2.5	Disable 💌					
🔽 СН:03	-00011 [-00.0011]	[0A] +/- 1 V 💌	1	-1	Disable 💌					
CH:04	-00022 [-000.022]	[0D] +/- 20 m/ 💌	20	-20	Disable 💌					
CH:05	-00005 [-000.005]	[08] +/- 10 V 💌	10	-10	Disable 💌					
CH:06	-00006 [-000.006]	[08] +/- 10 V 💌	10	-10	Disable 💌					
CH:07	-00001 [-000.001]	[08] +/- 10 V 💌	10	-10	Disable 💌					
					Set Alarm					
1										

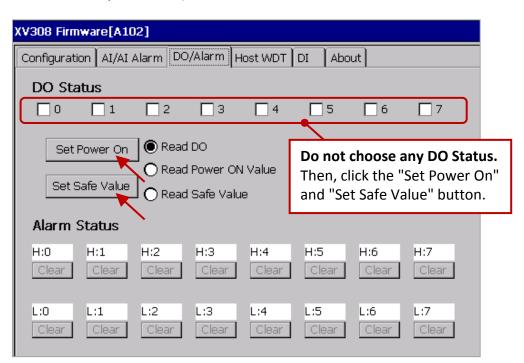
#### <u>XV308:</u>

Type Code	Range	Data Format	Minimum	Maximum
05		Engineering	-25000	+25000
05	+/-2.5 V	Hexadecimal	8000h	7FFFh
06	1 / 20 m A	Engineering	-20000	+20000
00	+/-20 mA	Hexadecimal	8000h	7FFFh
07	+4 mA ~ +20 mA	Engineering	+4000	+20000
07	+4 IIIA +20 IIIA	Hexadecimal	0000h	FFFFh
08	. / 10. /	Engineering	-10000	+10000
08	+/-10 V	Hexadecimal	8000h	7FFFh
09		Engineering	-5000	+5000
09	+/-5 V	Hexadecimal	8000h	7FFFh
0A	. / 1 . /	Engineering	-10000	+10000
UA	+/-1 V	Hexadecimal	8000h	7FFFh
0D	1 / 20 m A	Engineering	-20000	+20000
UU	+/-20 mA	Hexadecimal	8000h	7FFFh
1A	0 mA ~ +20 mA	Engineering	0	+20000
AT	0 IIIA +20 IIIA	Hexadecimal	0000h	FFFFh

#### Note:

- 1. For easy to use, recommended to use the data format "Engineering". (E.g., "+/-2.5 V" will show as "-25000 to +25000" and "+4 mA to +20 mA" will show as "+4000 to +20000")
- When using these "Type Code" 06, 07, 0D, 1A, please check if the position of eight hardware jumpers on the XW board are correct. www.icpdas.com/root/product/solutions/datasheet/hmi\_touch\_monitor/XV308.pdf

<u>Note:</u> When using the XV308, you need to click the "Set Power On" and "Set Safe Value" button (do not choose any DO Status) in the "DO/Alarm" tab.



 Finally, back to the "Configuration" tab and click the "Set Module Configuration" button (Step5) to finish the AI/AO configuration, and then close the "DCON\_Utility\_Pro\_CE\_200.exe". In addition, click the "Win\_GRAF\_WP\_5238" on the desktop to run the Win-GRAF driver (like Step 1).

Follow the similar way like the steps above to configure the AI/AO of the XV310.

Type Code	Range	Data Format	Minimum	Maximum
05		Engineering -25000		+25000
05	+/-2.5 V	Hexadecimal	8000h	7FFFh
06	1/20 mA	Engineering	-20000	+20000
00	+/-20 mA	Hexadecimal	8000h	7FFFh
07	+4 mA ~ +20 mA	Engineering	+4000	+20000
07	+4 IIIA +20 IIIA	Hexadecimal	0000h	FFFFh
08	. / 10. V	Engineering	-10000	+10000
08	+/-10 V	Hexadecimal	8000h	7FFFh
09	+/-5 V	Engineering	-5000	+5000
09		Hexadecimal	8000h	7FFFh
0A	+/-1 V	Engineering	-10000	+10000
UA	+/-1 V	Hexadecimal	8000h	7FFFh
0D	1/20 mA	Engineering	-20000	+20000
00	+/-20 mA	Hexadecimal	8000h	7FFFh
1A	0 mA ~ +20 mA	Engineering	0	+20000
IA	0 IIIA +20 IIIA	Hexadecimal	0000h	FFFFh

#### XV310 - Analog Input:

#### Note:

- 1. For easy to use, recommended to use the data format "Engineering". (E.g., "+/-2.5 V" will show as "-25000 to +25000" and "+4 mA to +20 mA" will show as "+4000 to +20000")
- 2. When using these "Type Code" 0, 1, 06, 07, 0D, 1A, please check if the position of eight hardware jumpers on the XW board are correct. www.icpdas.com/root/product/solutions/datasheet/hmi\_touch\_monitor/XV310.pdf

Type Code	Range	Data Format	Minimum	Maximum
0	0 mA ~ +20 mA	Engineering	0	+20000
0	0 IIIA +20 IIIA	Hexadecimal	0000h	FFFFh
1	14 m A ~ 1 20 m A	Engineering	+4000	+20000
L	+4 mA ~+20 mA	Hexadecimal	0000h	FFFFh
2	0\/~.10.\/	Engineering	0	+10000
2	0V ~ +10 V	Hexadecimal	0000h	FFFFh
2	./ 10.)/	Engineering	-10000	+10000
3	+/-10 V	Hexadecimal	8000h	7FFFh
4	0 V ~ +5 V	Engineering	0	+5000
4	UV +5V	Hexadecimal	0000h	FFFFh
	. / /	Engineering	-5000	+5000
5	+/-5 V	Hexadecimal	8000h	7FFFh

#### XV310 - Analog Output:

# 5.1.7 Connecting the XV107/ XV107A (8 DI, 8 DO)

The XV107/XV107A is an 8-channel digital input and 8-channel digital output board. This section provides a Win-GRAF demo project - "demo\_XV107.zip". First, go to <u>Section 5.1.6</u> for the information of the XV Board before using it.

#### **Demo description:**

This demo added two data blocks. One is used to read 8 DI data and the other is used to write 8 DO data.

1. Mouse double click the 1st data block (i.e., <2> Read Input Bits) to open the setting window.

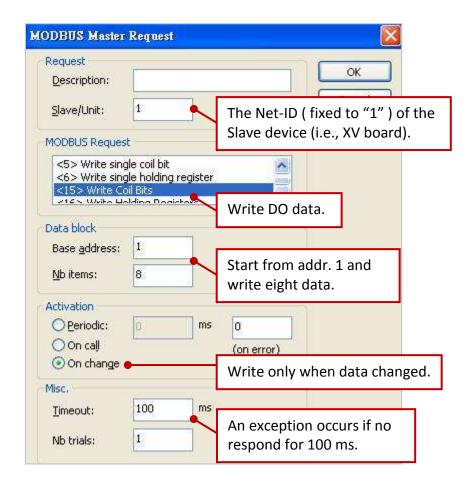
Workspace	10 Drivers						H Z	1X
demo_XV107     d		DBUS Master RTU: COM0:115200,N,8,1 *■ <2> Read Input Bits (1) [18] *■ <15> Write Coil Bits (1) [18]				▼ Name XV107_DI_00 XV107_DI_01 XV107_DI_02	Type BOOL BOOL BOOL	^
🔄 🛁 Watch (for debugging)	Symbol	Operation	Offset	Mask	Storage	XV107_DI_03	BOOL	III
Soft Scope Initial values We Binding Configuration Sig Global defines Wariables E Types	XV107_DL_(           XV107_DL_(	01 Data exchange 02 Data exchange 03 Data exchange 04 Data exchange	0 1 2 3 4 5	FFFF FFFF FFFF FFFF FFFF	Default Default Default Default Default Default	XV107_DI_04 XV107_DI_05 XV107_DI_06 XV107_DI_07 XV107_status XV107_D0_00	BOOL BOOL BOOL DINT BOOL	
	XV107_DI_( XV107_DI_( XV107_state	)7 Data exchange	6 7 0	FFFF FFFF FFFF	Default Default Default	Name V	alue	WIT.

Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address). Moreover, if you set the "Operation" as "Error report", the "Offset" value for the mapping variable (Date Type: DINT) must set to "0".

ODBUS Master	Request	<b>X</b>
Request Description:	1	ОК
<u>Slave/Unit:</u>	1	The Net-ID ( fixed to "1" ) of the Slave device (i.e., XV board).
<1 > Read Coil	Bits	
<2> Read Inpl <3> Read Hold <4> Read Top	ling Registers	Read DI data.
Data block	1	
Base <u>a</u> ddress: <u>N</u> b items:	8	Start from addr. 1 and read eight data.
Activation O Periodic:	50	ms 0
◯ On call ◯ On change		Read every 50 ms.
Misc.		
<u>T</u> imeout: Nb trials:	100	An exception occurs if no respond for 100 ms.
	114 2.1	

2. Mouse double click the 2nd data block (i.e., <15> Write Coil Bits) to open the setting window.

Workspace	10 [	)rivers								ZX
⊡@ demo_XV107	眉	🖯 Mo MODBUS Maste	er					Name	Туре	
🛓 🛄 Exception programs	몳	🗄 📲 RTU: COMO	:115200,N,8,1					XV107_DI_05	BOOL	1
📩 🔤 Programs	1000	I i m						XV107 DI 06	BOOL	
🛄 📠 Main	*8		rite Coil Bits (1) [18] 🚬 🛌					XV107_DI_07	BOOL	
🗄 🚞 Watch (for debugging)	0	+0: >	<v107_d0_00< td=""><td></td><td></td><td>1</td><td></td><td>XV107_status</td><td>DINT</td><td></td></v107_d0_00<>			1		XV107_status	DINT	
Soft Scope		Symbol	Operation	Offset	Mask	Storage		XV107_DO_00	BOOL	
📰 Initial values			Data exchange	0	FFFF	Default	-	XV107_D0_01	BOOL	
Binding Configuration		XV107_D0_01	Data exchange	1	FFFF	Default		XV107_D0_02	BOOL	
🤤 😚 🚽 Global defines	¢,	XV107_D0_02	Data exchange	2	FFFF	Default		XV107_DO_03	BOOL	
	∎+	XV107_D0_03	Data exchange	3	FFFF	Default		XV107_DO_04	BOOL	
🛄 E Types	8+	XV107_D0_04	Data exchange	4	FFFF	Default		XV107_DO_05	BOOL	
	1	XV107_D0_05	Data exchange	5	FFFF	Default		XV107_DO_06		
	569852	XV107_D0_06	Data exchange	6	FFFF	Default			POOL	15
		XV107_D0_07	Data exchange	7	FFFF	Default	-	Name Va	alue	
		<	UI	$\square$			>	vanie Va	siue	



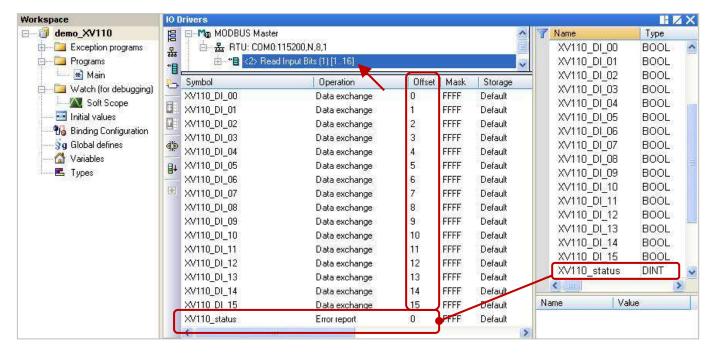
# 5.1.8 Connecting the XV110 (16 DI)

The XV110 is a 16-channel digital input board. This section provides a Win-GRAF demo project - "demo\_XV110.zip". First, go to <u>Section 5.1.6</u> for the information of the XV Board before using it.

#### **Demo description:**

This demo added one data block that used to write 16 DI data.

1. Mouse double click "<2> Read Input Bits" to open the setting window.



**Note:** The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address). Moreover, if you set the "Operation" as "Error report", the "Offset" value for the mapping variable (Date Type: DINT) must set to "0".

DBUS Master	Request	
<u>Description</u> :		ОК
Slave/Unit:	t	The Net-ID ( fixed to "1" ) of the Slave device (i.e., XV board).
<1> Read Coil <2> Read Input		
<3> Read Hold		Read DI data.
Data block		
Base <u>a</u> ddress:	1	
<u>N</u> b items:	16	Start from addr. 1 and read 16 data.
Activation		
• Periodic:	50 "	n <del>s</del> 0
◯ On call ◯ On change		Read every 50 ms.
Misc.	·	
<u>T</u> imeout:	100	An exception occurs if no
Nb trials:	1	respond for 100 ms.

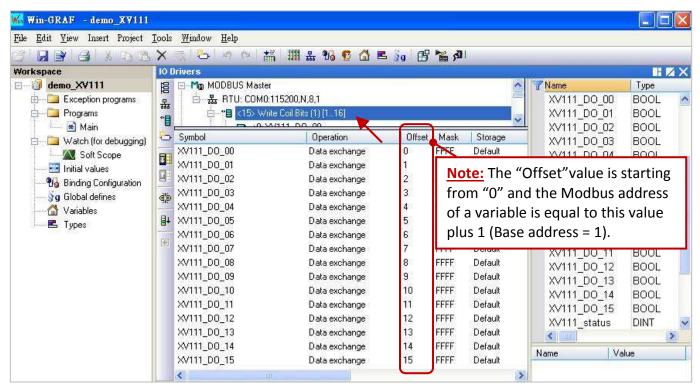
## 5.1.9 Connecting the XV111, XV111A (16 DO)

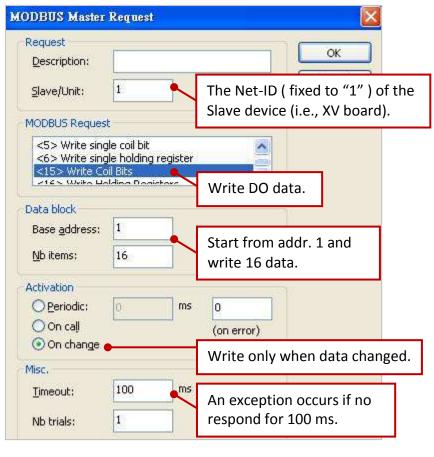
The XV111/ XV111A is a 16-channel digital output board. This section provides a Win-GRAF demo project - "demo\_XV111.zip". First, go to <u>Section 5.1.6</u> for the information of the XV Board before using it.

#### Demo description:

This demo added two data blocks. One is used to write 16 DO data and the other is used to read the DO status.

1. Mouse double click the 1st data block (i.e., <15> Write Coil Bits) to open the setting window.





2. Mouse double click the 2nd data block (i.e., <1> Read Coil Bits) to open the setting window.

Workspace	IO Dr	rivers						EB	XΜ
	臣	🖯 Mo MODBUS Maste	r			~	<b>Wame</b>	Туре	
🗄 🔤 Exception programs	몳	📋 🏭 RTU: COMO:	115200,N,8,1				XV111 DO 10	BOOL	~
🗄 🔤 Programs		亩— <b>*目</b> <15> Wr	ite Coil Bits (1) [116]				XV111_D0_11	BOOL	
🔛 🏦 Main	"8	🖻*日 <1> Rea	d Coil Bits (1) [1.1] 💦 🍗				XV111 DO 12	BOOL	
👜 🔤 Watch (for debugging)	0	Error	report: XV111_status			~	XV111_D0_13	BOOL	
Soft Scope		Symbol	Operation	Offset	Mask	Storage	XV111_DO_14	BOOL	
📰 Initial values	-046	XV111 status	Error report		FFFF	Default	XV111_DO_15	BOOL	
		AVTTI_status	Linor report	9	rr r	Derault	XV111_status	DINT	~
Šg Global defines	¢þ							10000000	-
🚰 Variables							Name V	alue	
E Types	8+	<	0			)		-1017C	

Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address).

Request	d'		ОК
Description:			
<u>S</u> lave/Unit:	1		The Net-ID ( fixed to "1" ) of th
		-	Slave device (i.e., XV board).
MODBUS Reques	t	L	
<1 > Read Coil			
<2> Read Inp <3> Read Hole	ding Registers		Read DO data.
Lease Ten	ut Dopistore		
Data block			
Base <u>a</u> ddress:	1	٦ آ	
Nb items:	1		Start from addr. 1 and
MD ICENIS:	1		read one data.
Activation			
<u>     Periodic:</u>	50	ms	0
⊖ On call	1		
On change		1	Read every 50 ms.
Misc.	10		
<u>T</u> imeout:	100	ms	An exception occurs if no
	1		respond for 100 ms.
Nb trials:	1		

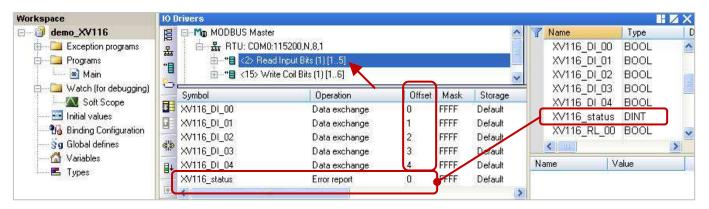
## 5.1.10 Connecting the XV116 (5 DI, 6 Relay)

The XV116 is a 5-channel digital input and 6-channel relay output board. This section provides a Win-GRAF demo project - "demo\_XV116.zip". First, go to <u>Section 5.1.6</u> for the information of the XV Board before using it.

#### Demo description:

This demo added two data blocks. One is used to read 5 DI data and the other is used to write 6 DO data.

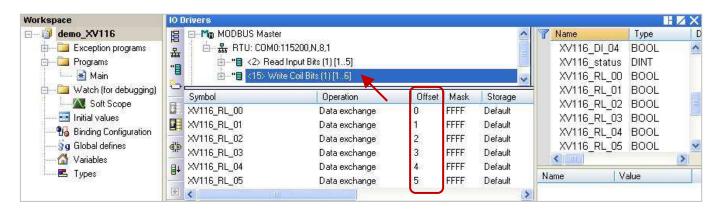
1. Mouse double click the 1st data block (i.e., <2> Read Input Bits) to open the setting window.

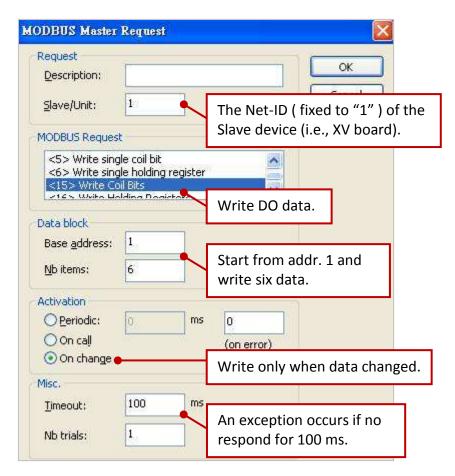


**Note:** The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address). Moreover, if you set the "Operation" as "Error report", the "Offset" value for the mapping variable (Date Type: DINT) must set to "0".

Request Description:	1	ОК
Slave/Unit:	1 st	The Net-ID ( fixed to "1" ) of th Slave device (i.e., XV board).
<1> Read Coi <2> Read Inp		
<3> Read Hol	ding Register	
Data block		1
Base <u>a</u> ddress:	1	
<u>N</u> b items:	5	Start from addr. 1 and read five data.
Activation		
• Periodic:	50	ms o
🔿 On caḷi	den a	
On change		Read every 50 ms.
Misc.		
Misc. <u>T</u> imeout:	100	An exception occurs if no

2. Mouse double click the 2nd data block (i.e., <15> Write Coil Bits) to open the setting window.





## 5.1.11 Connecting the XV308 (8 AI, 8 DIO)

The XV308 is a 8-channel analog input and 8-channel digital input/output board. This section provides three Win-GRAF demo projects - "demo\_XV308\_1.zip", "demo\_XV308\_2.zip" and "demo\_XV308\_3.zip". First, go to Section 5.1.6 to view the XV Board instructions and then configure each AI channel by using "DCON\_Utility\_Pro\_CE\_200.exe".

#### Demo description: (demo\_XV308\_1)

This demo added two data blocks, one is used to read 8 AI data and the other is used to read 8 DI data.

1. Mouse double click the 1st data block (i.e., <4> Read Input Registers) to open the setting window.

Workspace	10 [	Drivers							11 2	4X
demo_XV308_1      Exception programs      Programs      Main	면 <sup>많</sup> 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1					1	Name Global v XV308_AI_0 XV308_AI_0 XV308_AI_1	Type ariables INT INT	^
😑 🔁 Watch (for debugging)	0	Symbol	Operation	Offset	Mask	Storage	1	XV308_AI_2	INT	2
Initial values Initial values Initial values Initial values Initial values Initial Binding Configuration Initial Values Initial Values		XV308_AI_1 XV308_AI_2 XV308_AI_3 XV308_AI_4 XV308_AI_5 XV308_AI_6	Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange	0 1 2 3 4 5 6	FFFF FFFF FFFF FFFF FFFF FFFF	Default Default Default Default Default Default		XV308_AI_3 XV308_AI_4 XV308_AI_5 XV308_AI_6 XV308_AI_7 XV308_status XV308_bI_0	INT INT INT INT S DINT BOOL	⊃ <u>~</u>
		XV308_AI_7 XV308_status	Data exchange Error report	0	FFFF	Default Default	N	lame	Value	5

Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address). Moreover, if you set the "Operation" as "Error report", the "Offset" value for the mapping variable (Date Type: DINT) must set to "0".

MODBUS Master	Request	$\mathbf{X}$
Request Description:		ОК
Slave/Unit: MODBUS Request	1	The Net-ID ( fixed to "1" ) of the Slave device (i.e., XV board).
<2> Read Input <3> Read Holdi <4> Read Input	ng Registers t Registers	
Data block	o coil hit	Read DI data.
Base <u>a</u> ddress:	1	
<u>N</u> b items:	8	Start from addr. 1 and read eight data.
Activation		
• Periodic:	50 ms	0
◯ On caḷ ◯ On change		Read every 50 ms.
Misc.		
<u>T</u> imeout: Nb trials:	100 ms	An exception occurs if no respond for 100 ms.

2. Mouse double click the 2nd data block (i.e., <2> Read Input Bits) to open the setting window.

Workspace	10 E	)rivers							H Z	X
⊡ <b>i</b> demo_XV308_1	臣	📮 M🛛 MODBUS M				^	Y	Name	Туре	
🖶 🧰 🧰 Exception programs	몺	白 品 RTU: CO	OM0:115200,N,8,1					XV308_AI_7	INT	1
🖨 🔚 Programs	*	· <b>⊡</b> -* <b>⊟</b> <4>	Read Input Registers (1) [18]					XV308_status	5 DINT	
🛄 🏦 Main	110	E	Read Input Bits (1) [33, 40]					XV308_DI_0	BOOL	
🗄 🔁 Watch (for debugging)	0	Symbol	Operation	Offset	Mask	Storage	-	XV308_DI_1	BOOL	
Soft Scope		XV308 DI 0	Data exchange	0	FFFF	Default		XV308_DI_2	BOOL	
- Initial values		XV308_DI_1	Data exchange	1	FFFF	Default		XV308_DI_3	BOOL	-
Binding Configuration		XV308_D1_2	Data exchange	2	FFFF	Default		XV308_DI_4	BOOL	
	ġþ	XV308_D1_3	Data exchange	3	FFFF	Default		XV308_DI_5	BOOL	
Variables	E+	XV308_DI_4	Data exchange	4	FFFF	Default		XV308_DI_6	BOOL	-
L 🖪 Types	-	XV308_DI_5	Data exchange	5	FFFF	Default		XV308_DI_7	BOOL	
	1	XV308_DI_6	Data exchange	6	FFFF	Default				
		XV308_DI_7	Data exchange	7	FFFF	Default	N	lame 4	Value	
		<			)		> <	10 Xini	j'	>

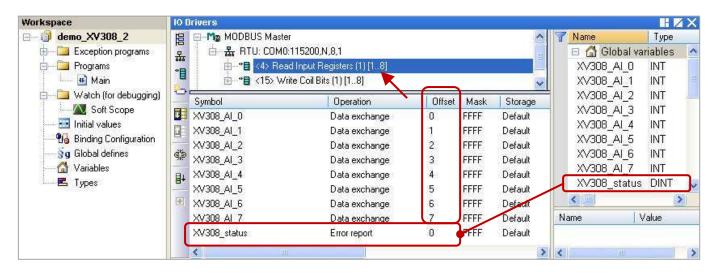
Notw: When using the XV308 to read DI data, the address must start from "33".

Request	-		ОК
Description:			
<u>S</u> lave/Unit:	1		The Net-ID ( fixed to "1" ) of th
MODBUS Reques	st -		Slave device (i.e., XV board).
<1 > Read Coi			<u> </u>
<2> Read Inp <3> Read Hol <4> Read Inp	ding Registers	4	Read DI data.
Data block			
Base <u>a</u> ddress:	33		
<u>N</u> b items:	8	$\searrow$	Start from addr. 33 and read eight data.
Activation		L	
• Periodic:	50	ms	0
🔘 On caḷ	-	$\checkmark$	Read over tE0 ms
On change			Read every 50 ms.
Misc.	12		
Timeout:	100	ms	An exception occurs if no
Tuncode			

#### Demo description: (demo\_XV308\_2)

This demo added two data blocks, one is used to read 8 AI data and the other is used to write 8 DO data.

1. Mouse double click the 1st data block (i.e., <4> Read Input Registers) to open the setting window.

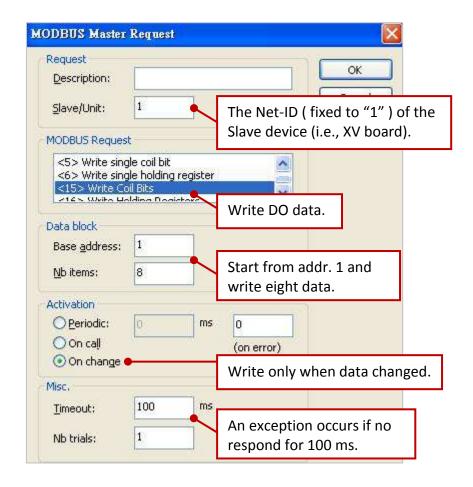


Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address). Moreover, if you set the "Operation" as "Error report", the "Offset" value for the mapping variable (Date Type: DINT) must set to "0".

MODBUS Master	Request	$\mathbf{X}$
Request Description:		ОК
Slave/Unit:	1 •	The Net-ID ( fixed to "1" ) of the Slave device (i.e., XV board).
<2> Read Inp <3> Read Hold <4> Read Inp	ding Registers ut Registers	
Data block	le coil bit	Read Al data.
Base <u>a</u> ddress: <u>N</u> b items:	1 8	Start from addr. 1 and read eight data.
Activation	50 ms	0
On ca∥ On change		Read every 50 ms.
Misc. Timeout:	100 ms	
Nb trials:	1	An exception occurs if no respond for 100 ms.

2. Mouse double click the 2nd data block (i.e., <15> Write Coil Bits) to open the setting window.

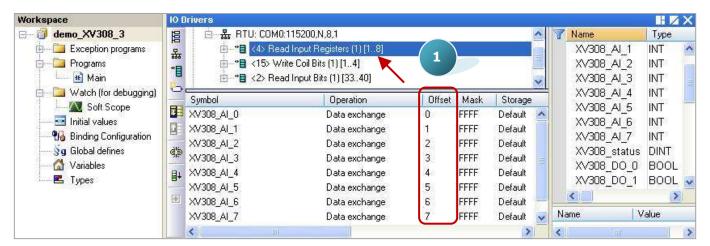
Workspace	10 1	Drivers							1	$\mathbf{X}$
demo_XV308_2     Exception programs     Programs     Main	間より	由 <b>*目</b> <4> Rea ⊡ <b>*目</b> <15> W						Name XV308_stat XV308_DO XV308_DO XV308_DO	0 BOOL	
🖻 🛁 Watch (for debugging)	0	Symbol	Operation	Offset	Mask	Storage		XV308_DO	514 30000351024	
Initial values Initial values Initial values Initial values Initial values Initial values Initial values Initial values Initial values Initial values		14/200 0.0	Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange Data exchange	0 1 2 3 4 5 6 7	FFFF FFFF FFFF FFFF FFFF FFFF FFFF	Default Default Default Default Default Default Default		XV308_DO XV308_DO XV308_DO XV308_DO XV308_DO XV308_DO	4 BOOL 5 BOOL 6 BOOL 7 BOOL	44 44 46
	-	<	F				> <	t 📔 👘 Kimita		>



#### Demo description: (demo\_XV308\_3)

This demo added three data blocks, the 1st one is used to read 8 AI data, the 2nd one is used to write 4 DO data and the 3rd one is used to read only 4 DI data.

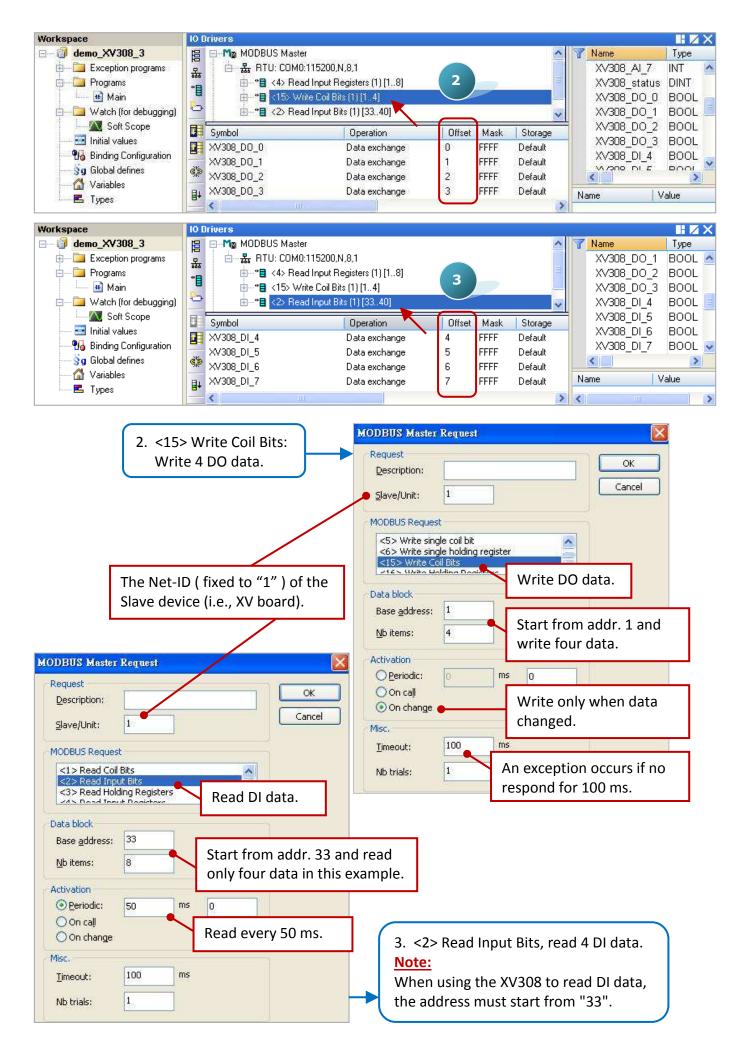
1. Mouse double click the 1st data block (i.e., <4> Read Input Registers) to open the setting window.



Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address).

Request Description:	1	ОК
Slave/Unit:	1 t	The Net-ID ( fixed to "1" ) of the Slave device (i.e., XV board).
<2> Read Inp <3> Read Hold <4> Read Inp	ding Registers	
Data block		Read AI data.
Base <u>a</u> ddress: <u>N</u> b items:	1 8	Start from addr. 1 and read eight data.
Activation	50	ms 0
◯ On caḷ ◯ On change		Read every 50 ms.
Misc.	100	
Timeout:	100	An exception occurs if no

- 2. As the figure below, mouse double click the 2nd data block (i.e., <15> Write Coil Bits) to view the setting window.
- 3. As the figure below, mouse double click the 3rd data block (i.e., <2> Read Input Bits) to view the setting window.



## 5.1.12 Connecting the XV310 (4 AI, 2 AO, 4 DI, 4 DO)

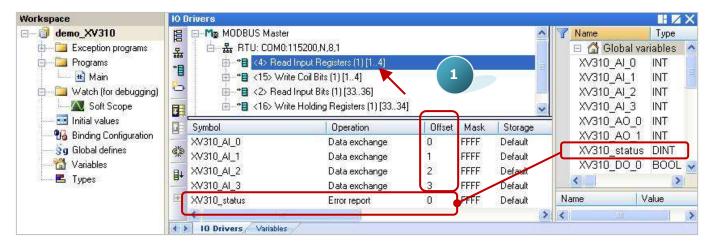
The XV310 is a 4-ch analog input, 2-ch analog output, 4-ch digital input and 4-ch digital ouput board. This section provides a Win-GRAF demo projects - "demo\_XV310.zip".

First, go to <u>Section 5.1.6</u> to view the XV Board instructions and then configure each AI channel by using "DCON\_Utility\_Pro\_CE\_200.exe".

#### **Demo Description**

This demo added four data blocks. The 1st one is used to read 4 AI data, the 2nd is used to write 4 DO data, the 3rd is used to read 4 DI data and the 4th is used to write 2 AO data.

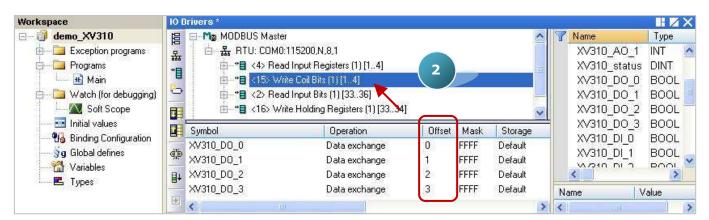
1. Mouse double click the 1st data block (i.e., <4> Read Input Registers) to open the setting window.



Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address). Moreover, if you set the "Operation" as "Error report", the "Offset" value for the mapping variable (Date Type: DINT) must be set to "0".

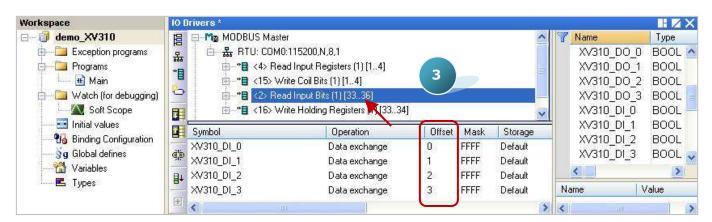
Request Description:	1		ОК
<u>Slave/Unit:</u> MODBUS Reques	1 t		The Net-ID ( fixed to "1" ) of the Slave device (i.e., XV board).
<2> Read Inpu <3> Read Hold <4> Read Inpu	ling Regist		
Data block	ila cail bif		Read Al data.
Base <u>a</u> ddress:	1		
<u>N</u> b items:	4		Start from addr. 1 and read four data.
Activation			
• Periodic:	50	ms	0
🔿 On caḷi	100		
On change			Read every 50 ms.
Misc.			
Timeout:	100	ms	
	1 Course	-	An exception occurs if no

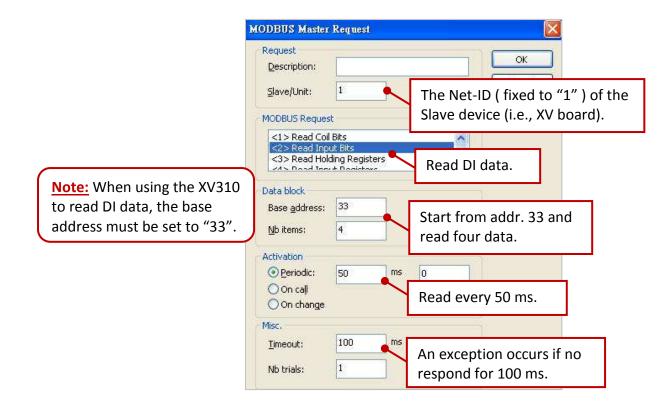
2. Mouse double click the 2nd data block (i.e., <15> Write Coil Bits) to view the setting window.



Request	11		
Description:		ОК	
<u>S</u> lave/Unit:	1	The Net-ID of the Slave device (i.e., XV board).	
MODBUS Reques	t		
<5> Write sing <6> Write sing <15> Write Co	ile holding registe	er en	
212× Obits Da		Write DO data.	
Data block			
Base <u>a</u> ddress:	1		
		Start from addr. 1 and	
<u>N</u> b items:	4	write four data.	
-	4		
Activation	4 m	write four data.	
Activation O Periodic: O On call		write four data.	
Activation		write four data.	Ige
Activation O Periodic: O On call		write four data.	Igo
Activation Periodic: On call On change		write four data.	ıg

3. Mouse double click the 3rd data block (i.e., <2> Read Input Bits) to view the setting window.





4. Mouse double click the 4th data block (i.e., <16> Write Holding Registers)

Workspace	IO Drivers *									iX
demo_XV310      Exception programs     Programs     Main     Watch (for debugging)     Soft Scope     Initial values		U: COM0:115200, <4> Read Input F <15> Write Coil B <2> Read Input B	Registers (1) [14] its (1) [14] its (1) [3336] ng Registers (1) [3334				X X X X X	me V310_AI_1 V310_AI_2 V310_AI_3 V310_AO_( V310_AO_ V310_AO_ V310_statu	Type INT INT INT INT INT 1 INT	
	Symbol           XV310_A0_0           XV310_A0_1           XV310_A0_1           Image: Constraint of the symbol           Image: Constraity           Image: Constraint	Variables /	Operation Data exchange Data exchange	Offset 0 1	Mask FFFF FFFF	Storage Default Default	× Name		Value	>
Note: When using to write AO data, th address must be set	the XV310 ne base	<15> Write C	1 st	Slave of Write A Start fr write t O (on error) Write of An exco	AO da rom a wo da only v	ıddr. 33 a	boarc nd a char	I).		

## 5.1.13 To Disable/Enable the Modbus RTU/ASCII Master Port

The Modbus RTU/ASCII Master ports which are enabled in the Win-GRAF "Fieldbus Configuration" -"IO Drivers" setting window, will automatically work after the PAC is powered on. If user wants to disable one of the Modbus Master ports, use the **"MBRTU\_M\_disable"** function (see below).

```
(* Declare To_disable as BOOL *)
If To_disable then
To_disable := FALSE ;
MBRTU_M_disable (3) ;
End_if;
```

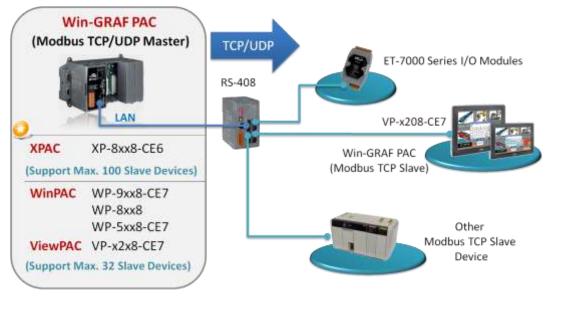
In the above code, when you set "To\_disable" as "TRUE", it will disable the Modbus RTU/ASCII Master port - COM3. And later, you can enable it again by using the **"MBRTU\_M\_enable"** function (see below).

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

The "**MBRTU\_M\_status**" function listed above is used to get the status of the Modbus RTU/ASCII Master port, for example, enabled (True) or disabled (False).

# 5.2 Enabling the Win-GRAF PAC as a Modbus TCP/UDP Master (Ethernet I/O)

#### **Application Diagram:**



(Refer <a>Pl-1</a> to see the PAC model numbers.)

#### The Setting Steps:

1. Click the tool icon "Open Fieldbus Configuration" to open the "I/O Drivers" window.

Win-GRAF - Test_01			×
<u>File Edit V</u> iew Insert Project	<u>T</u> ools <u>W</u> indow <u>H</u> elp		
12   🖬 🖻   🛃   X 🗈 🛍	🗙 🛒 (우) 🖓 🔛 🚟	😘 😰 🖆 🖻 💡 🔠 🎽 🔊 🛛	
Workspace	10 Drivers		×
⊡ 🗊 Test_01		Name Value 🛛 🍸 Name Type	
🗄 🛅 Exception programs		🚮 Global variables	^
🛓 🔤 Programs		🚽 📈 📈 🖌 🖌	
🔟 Main		📄 Main	
🗄 📴 Watch (for debuggi		💾 pOnBadindex	
Soft Scope	T	POnDivZero	~
🔤 Initial values			
📆 Binding Configuration		Name Value	_
🚽 😽 g Global defines	ġ,		_
🚮 Variables			>
🔤 Types	Build		×
	Build Cross references Runtime	Call stack Breakpoints Digital sampling trace Pro	mot
	a P band crossierences Andrume	A can stack breakpoints A Digital sampling trace A Pro-	mpt

- 2. Click "Insert Configuration" icon in the left side of the "I/O Drivers" window, and then click "MODBUS Master", then click "OK" to enable a Modbus Master.
  - <u>Note:</u> One "Modbus Master" can set up multiple Ports (see the next step), can set as a Modbus Master RTU/ASCII Port (Refer <u>Section 5.1</u>) or a Modbus Master TCP/UDP Port or can set up not to enable the setting.

10 Drivers						ΔX
E		Name Value	7	Name	Туре	
温				🚮 G	Global variables	^
				🚽 F	RETAIN variable	es 🔄
<b>∃</b> ₩	Add Configuration					X
	Choose a configuration				ОК	П
						$\mathbf{x}$
	⊟ MODBUS				Cancel	
eje -	MODBUS Master					
	MODBUS Slave	$\sim$				

3. Click the tool icon "Insert Master/Port" in the left side, open the setting window and select the "MODBUS on Ethernet".

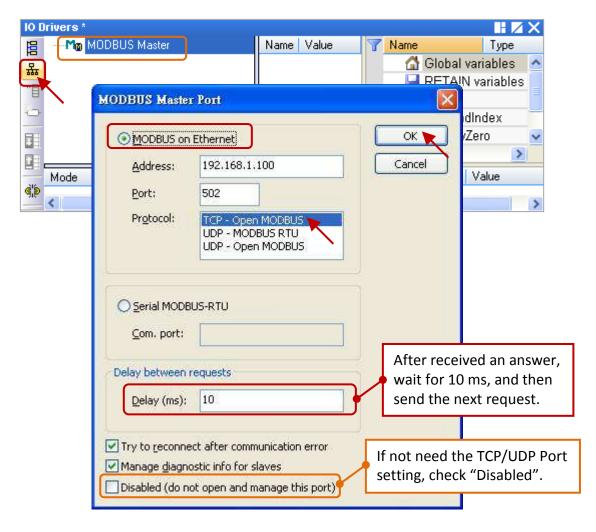
Set up the following items, and then click "OK".

Address: Fill in the IP Address of the Modbus Slave device (e.g., "192.168.1.100").

Port: TCP port Number of the Slave device.

Protocol: If as a Modbus TCP Master, select the "TCP – Open MODBUS". If as a Modbus UDP Master, choose the "UDP – Open MODBUS".

Delay: Fill in the delay time (e.g., 10 ms, can be 0 ~ 10000).



4. Click the icon "Insert Slave/Data Block" in the left side to create a "Data Block".

10 0	rivers *		
冒	⊡…Mo MODBUS Master	Name	Value
쁆	🔐 🔐 Open MODBUS: 192.168.1.100:502	Mode	Open MODBUS
		Address	192.168.1.100
		Port	502
-	$\mathbf{N}$	Reconnect after error	
		Slaves diagnostics	
		Delay between requests (ms)	10
		Disabled	

#### **Read AI Data**

5. In the "MODBUS Master Request" setting window, set up the following items, and then click "OK".

MODBUS Master Request	In this example
Description:	a. <u>Slave/Unit</u> :
Cancel	Fill in the Net-ID of the Slave device
Slave/Unit: 1	(Usually is "1").
MODBUS Request	b. MODBUS Request:
<2> Read Input Bits	Select "<4> Read Input Registers".
<3> Read Holding Registers <4> Read Input Registers	c. <u>Base address</u> :
25 With single call hit	Default to start from 1.
C Data block	<u>Nb items</u> :
Base address: 1	The AI numbers to read (here is 4).
Nb items: 4	Note:
	If want to change the "Base address",
d Activation eriodic: 1000 ms 15000	please use mouse to right-click the
	"MODBUS Master", and then select
On change	"MODBUS Master Addresses" to
It's not suitable for "Read" commend.	change the value.
e Timeout; 1000 ms	IO Drivers
	📙 🖃 Modbus Master 🌪
Nb trials:	
N:	MODBUS Master addresses
d. Activation: the sending way of Modbus Request.	First valid MODBUS addresses OK
Deriadia Cond request periodically in this area it	Input bits: 1

- Send request periodically. In this case, it Periodic: sends request every 1 Sec. "on error" means that when an error occurs, the next sending time (in this case, 15 seconds).
- On call: It will send the request once when a program calls it.

IO Drivers ☐ □····M <sub>®</sub> MODBUS Master								
MODBUS Master addresses								
First valid MODBUS ad	ddresses	ОК						
Input <u>b</u> its:	1	Cancel						
<u>⊂</u> oil bits:	1							
Input <u>r</u> egisters:	1							
Holding registers:	1							

On change: It will send the request once when data is changed.

e. <u>Timeout</u>: Set up the max. time to wait for the response. If exceeds it, that means an error. (For Modbus TCP/UDP, recommended: 1000 ~ 3000 ms; this example is 1000 ms) 6. Open the "Variables" window, set up the variables want to use.

Norkspace	IO Drivers *		
⊡— <mark>[]]</mark> Test_2	📔 🖃 ModBUS Master	Name	Value
🗄 🛅 Exception programs	용 Dpen MODBUS: 192.168.1.100:502	Request	<4> Read Input Registers
Programs Main Watch (for debuggi	• P ZAN Read Input Registers (1) [1 4]	Slave/Unit	1
		Address	1
		Nb Item	4
	Tips:	Activation	Periodic
	press FI key to see the setting	Period (ms)	1000
📲 😼 Binding Configuration	method for MODBUS Master.	Period on error	15000
🔤 😼 🚽 🚽 🚽 🚽	dia .	Timeout (ms)	1000
- 🚮 Variables 🥿		Number of trials	1
E Types	<b>I</b> +	Description	

Please follow the table to set up 4 WORD (16 bit) variables (refer Section 2.3.1).

Variable Name	Data Type	Dim.	Description
Word_1 ~ Word_4	WORD		Used to read the AI data (16 bit)
Status	DINT	5	Used to record the read/write status

Refer the <u>Appendix A</u> to see the data types and range of the variables. After setting up, it is as the picture below.

🕇 Name	🔺 Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
Status	DINT	[04]				-		
Word_1	WORD	96570 79						
Word_2	WORD							
Word_3	WORD							
Word_4	WORD							
<			ñ	1.045 - 1915		<u> </u>		

7. In the "I/O Drivers" window, drag the variables ("Word\_1 ~ Word\_4" and "Status") from the Variables area to the "Symbol" area of the Data Block. <u>Notice:</u> This example shows "Status" is an Array variable. When drag it to the "Symbol" area, it will become "Status[0] ~ Status[4]", please press "Delete" key to delete "Status[1] ~ [4]".

	E Mo MODB	US Master			Name	Value		71	Vame	4	Туре	Dim.
品	· · · 사 · ·	en MODBUS: 192.168.1	1.100:502	2	Request	<4> Read In	put Regis 🔨		Status		DINT	[04]
	÷*	<4> Read Input Regist	ers (1) [1.	.4]	Slave/Unit	1,			Word_1		WORD	1950 - 5
1					Address	1	~	1	Word_2		WORD	
2	Symbol	Operation	Offse	et   Ma	sk   Storage		Range		Word_3		WORD	
	Status[0]	Data exchange	0	FFF	F Default				Word_4		WORD	
	Word_1	Data exchange	0	FFF	F Default			R		AIN Y	variables	
	Word_2	Data exchange—	- + -	FFF	F Default			Nam		1.5	/alue	L.
Þ	Word_3	Data exchange	0	FFF	F Default			Nan		10	raiue	
ļ	Word_4	Data exchange	0	FFF	F Default							
	<			21			>					

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 5-41

8. Set the "Operation" of the "Status[0]" to "Error report" (If reading data fails, its value is an "Error Code"; when reading data OK, it will reset to "0".). Press "F1" key to view the setting descriptions for the Modbus Master. In the title of "Status and command variables", you can find the details about this commend and "Error Code".

目	🖃 Modb	US Master		Name		Value	11		
뮮	📥 🖧 Op	en MODBUS: 192.168.1	.100:502	Request	3	<4> Read Inj	out Regist		
		<4> Read Input Registe	ers (1) [14]	Slave/U	nit	1			
*				Address		1			
0				Nb Item		4			
				Activatio	m	Periodic			
				Period (r	ns)	1000			
				Period o	n err	15000			
ġ.				Timeout	(ms)	1000			
∎+		8		Number	of tri	1			
8+				Descript	ion				
÷	Symbol	Operation	Offset M	lask St	orage	7	Range (I		
	Status[0]	Error report	The stream of the second se	200 Jan 1997	fault				
	Word_1	Data exchange	0 ,	EE Da	ƙardk				
	Word_2	Data exchange	1						
	Word_3	Data exchange	2		_				
	Word_4	Data exchange	3 Na	me	%				
	<		Fro	m: 0	1	\$	By:	1	\$
< >	10 Drivers	Variables					-		_
				lesults					
			0						
			1						
			1						
			1						
			1						
			1						

9. Select "Word\_1 ~ Word\_4" and click "Iterate property" to set up the "Offset" value (From: 0; By: 1).

The setting steps of "Modbus Master Request" for both "Modbus Master RTU/ASCII Port (<u>Section 5.1</u>)" and "Modbus Master TCP/UDP Port" are the same. Now, we have finished the setting to read AI data. Please click the item number (link to the Section 5.1.1~5.1.5) in the table below for the setting steps to read/write other data.

Items	Function Code	Modbus Request	Description		
<u>1</u>	2	Read Input Bits	Read DI data		
<u>2</u>	5	Write single coil bit	Write DO data		
<u>3</u>	4	Read Input Registers	Read AI data		
<u>4</u>	6	Write single holding register	Write one AO data (16-bit)		
<u>5</u>	16	Write Holding Registers	Write multiple AO data (16/32 bits)		

<u>Note:</u> If you want to disable the Modbus TCP/UDP Master port while the program is running. Refer the <u>Section 5.2.4</u> to use the "MBTCP\_M\_disable" function (and use "MBUDP\_M\_disable" for UDP).

### 5.2.1 Connecting ET-7000 Series I/O Module

ICP DAS ET-7000 is a series of I/O module supporting Modbus TCP Slave protocol. The Win-GRAF PAC can enable the Modbus TCP Master to connect the ET-7000 modules. The maximum recommend the amount of the connecting ET-7000 modules depends on the PAC model, such as the WP-5238-CE7 and XP-8xx8-CE6, recommends a maximum of 200; the WP-8xx8, WP-8xx8-CE7, and VP-x2x8-CE7 is recommended that no more than 32.

For more information about the ET-7000 series products, please visit the website: <u>http://www.icpdas.com/root/product/solutions/remote\_io/ethernet\_io/ethernet\_io\_selection.html</u>

#### Use Internet Browser to Set the ET-7000 Modules

Before the first time using the ET-7000, you must set up the ET-7000 by using the Internet Browser. When the ET-7000 shipping from the factory, the settings are: IP address = 192.168.255.1; Mask = 255.255.0.0. Please set the IP of your PC in the same network (e.g., set the IP to 192.168.255.100, Mask = 255.255.0.0), then open the browser (such as IE), and enter the IP of the ET-7000 to connect it.

#### **<u>Notice</u>**: The Dip Switch on the rear of the ET-7000 must stay in the "Normal" position.

	🌔 PET-7000 Web p	age - Windows Internet Ex	cplorer
	OO - E ht	tp://192.168.255.1/	
Username: Admin Password: Admin	及密碼。	255.1 的伺服器 192.168.255.1 需要 要求您的使用者名稱及密碼以 聲,不含安全連線)。	
(Case sensitive)	使用者名稱(U): 密碼(P):	2 Admin *****	×
		<ul> <li>✓記憶我的密碼 (R)</li> <li>確定</li> </ul>	取消

Click "Configuration" > "Module I/O Settings" to set up the range of channels as below, and then click "Submit".

🕒 💽 🔻 🙋 http://192.168.255.1/				🖌 🛃 🔊	百度一下,你就知道
★ 我的最愛			0	🙆 • 📾 - 🖂 🖶	▼ 網頁(P)▼ 安全性③▼ 工具
ICP DAS	Modbus Setting	s (Al Basic Setting)		Ellen	
Configuration		Range (40427)		Enable (00595)	
Network Settings	Ch0	[05] +/-2.5V	×	OFF ON C	
Basic Settings	Ch1	[04] +/-1V	×	OFF ON P	
Module I/O Settings	Ch2	[0F] T/C K-type, -270°C ~ 1372°C		OFF CON C	
	Ch3	[0F] T/C K-type, -270°C ~ 1372°C		OFF CON C	Set as "ON" to
Web HMI     Pair Connection	Ch4	[05] +/-2.5V		OFF ON 📀	enable it.
More Information	Ch5	[05] +/-2.5V	×	OFF ON C	
	Ch6	[05] +/-2.5V		OFF C ON C	
	Ch7	[05] +/-2.5V		OFF CON @	
	Ch8	[05] +/-2.5V	×	OFF ON .	
	Ch9	[05] +/-2.5V	~	OFF ON P	
Set "Al Data Format"					
to "ON" (Engineer),	Modbus Address	Function		Action	
means:	00629	50/60 Hz rejection for AI		the second se	50/1=60 Hz (default=1)
± 2.5: -25000 ~ +25000	00631	AI Data Format		OFF ON C	Hex/1=Engineeer (default=0)
± 1 :-1000~+1000	00632	Recover AI Calibration Values to Facto	ry Setting	1=Clear	
258 : 25.8 (°C)			bmit		N

Users can set the ET-7018Z's "AI Data Format" to "ON" (Engineering) for more convenient usage. For example:

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

#### Restore/Open the Demo Project:

The Win-GRAF demo projects in the following sections can be found on the shipping CD, please refer <u>Chapter 12</u>. Click the menu bar "File" > "Add Existing Project" > "From Zip" can restore/open/check the demo projects. (CD-ROM:\Napdos\Win-GRAF\demo-project\)

Demo Project	File Name	Description
ET-7060	demo_ET7060.zip	Read 6 DIs, write 6 DOs
ET-7018Z	demo_ET7018z.zip	Read 10 Als

1. Click the tool icon "Open Fieldbus Configuration" to open the "I/O Drivers" window.

Workspace	10 0	lrivers							HZ	×
3- ((demo_ET7060)	挹			_			0	🍸 Name	Туре	
E Exception programs	몲	😑 🚜 Open MODBU!	S: 192.168.1.128.50	2				ET7060_DI_00	BOOL	
📩 🛁 Programs	*	🗄 📲 🖓 🗄 🖂	nput Bits (1) [16]					ET7060_DI_01	BOOL	
🛄 편 Main	E	🗄 <b>* 😫</b> <15> Write	Coil Bits (1) [16]					ET7060_DI_02	BOOL	
🚊 🛁 Watch (for debugging)		Mg MODBUS Slave						ET7060_DI_03	BOOL	
Soft Scope		4.7820 1						ET7060_DI_04	BOOL	
🔜 Initial values		Request	Slave/Unit	Address	Nb Item	Activation	Peric	ET7060_DI_05	BOOL	
		<2> Read Input Bits	1	1	6	Periodic	50	ET7060_status	DINT	
🚽 🚽 🚽 🚽 🚽 🚽	₫\$	<15> Write Coil Bits	1	1	6	On Change	0	ET7060_RL_00	BOOL	
🚮 Variables	-					27800 00 03 VI 2025 20	ANDER	ET7060_RL_01	BOOL	
E Types	₿t	<					>	< 11	>	

2. Double click "Open Modbus: IP:502" to open the "MODBUS Master Port" window.

#### Notice:

All demo projects in this chapter can enable the Win-GRAF PAC as a Modbus **TCP** Master. Please fill in the current IP address of your ET-7000, and set "Port" to "502" and "Protocol" to "TCP - Open Modbus".

	Ethernet	
<u>A</u> ddress:	192.168.1.128	3 Cancel
<u>P</u> ort:	502	
Pr <u>o</u> tocol:	TCP - Open Me UDP - MODBUS UDP - Open Me	5 RTU
Serial MODE	3US-RTU	Enter the current IP addre
		of your ET-7000.
<u>⊂</u> om. port:		61 your E1 7000.
	requests	
	requests	
Delay between Delay (ms):	1	

## 5.2.2 Connecting the ET-7060 (6 DI, 6 Relay)

The ET-7060 is a 6 DI and 6 Relay channels Ethernet I/O module. The Win-GRAF demo project for this section is "demo\_ET7060.zip". Please refer <u>Section 5.2.1</u> to set up the module channels using the Internet Browser, and restore/open the demo project.

#### **Demo Description:**

This demo creates two Data Blocks, one is used to read 6 DI data, the other is used to write 6 DO data.

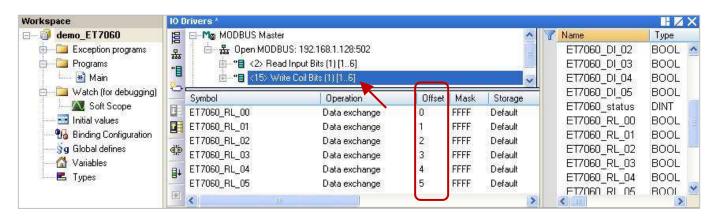
1. Double click the first Data Block (<2> Read Input Bits) to open the setting.

Workspace	101	Drivers *							HZ)	×
Emergia demo_ET7060	間器	E-Mo MODBUS Master	US: 192.168.1.128:502				^	Name           ET7060_DI_00	Type BOOL	1
ia → 🔁 Programs 🖬 Main	1	.⊞*∎ <2> Rea ⊡*∎ <15> Wri	d Input Bits (1) [16] te Coil Bits (1) [16]				~	ET7060_DI_01 ET7060_DI_02	BOOL BOOL	
🖻 📴 Watch (for debugging)	-	Symbol	Operation	Offset	Mask	Storage	,	ET7060_DI_03	BOOL	
Soft Scope		ET7060_D1_00	Data exchange	0	FFFF	Default	~	ET7060_DI_04	BOOL	
- Initial values		ET7060_DI_01	Data exchange	1	FFFF	Default		ET7060_DI_05	BOOL	1
Binding Configuration		ET7060_D1_02	Data exchange	2	FFFF	Default		ET7060_status	DINT	
g Global defines	ġ.	ET7060 DI 03	Data exchange	3	FFFF	Default		ET7060_RL_00	BOOL	
Variables	∎+	ET7060 DI 04	Data exchange	4	FFFF	Default		ET7060_RL_01	BOOL	
E Types	8*	ET7060 DI 05	Data exchange		FFFF	Default		ET7060_RL_02	BOOL	
	Ŧ	ET7060_status	Error report	0	FFFF	Default	~	ET7060_RL_03	BOOL	
		<				>			>	

Notice: The value of the "Offset" starts from "0", but the Modbus address of the variable is the "Offset" value plus 1 (Base address). If set the "Operation" to "Error report", the "Offset" value of the variable (Data Type: DINT) must set to "0".

IODBUS Master	Request						
Request Description:					ок		
Slave/Unit:	1 t		The Net-ID o (ET-7000) is			it	
<1> Read Coil <2> Read Inp <3> Read Hold	ut Bits						I
242 Dood Top		2	Read DI data	ì			
Data block	-	-					
Base <u>a</u> ddress: <u>N</u> b items:	1		Start from th to read 6 ite		ddress 1		
Activation							
• Periodic:	50	ms	5000		and once a		
On call On change		(on error)	lf	ead once e error occu end the ne	ırs, v	<i>.</i> wait 5 s to	
Misc.				Ľ			onniena.
Timeout:	500	ms					
– Nb trials:	1		If no respond	ds ir	n 500 ms, r	nea	ns error.
	1	-					

2. Double click the second Data Block (<15> Write Coil Bits) to open the setting window.



DBUS Master	Request		2	3
Request Description:			ОК	
<u>Slave/Unit:</u> MODBUS Reques	1 <b>•</b>		of the Slave U fixed to "1".	nit
<5> Write sing <6> Write sing <15> Write Co	le holding register			
216 CUMIES US	Idina Dogistors	Write DO da	ita.	
Data block		· · · · · · · · · · · · · · · · · · ·	17	
Base <u>a</u> ddress: <u>N</u> b items:	6	Start from t to write 6 it	ne address 1 ems.	
Activation			89	-
O Periodic:	0 ms	5000		
On call	()	(on error)		
On change		Write data	if data is char	nged
Misc.				
<u>T</u> imeout:	500 ms			
Nb trials:	1	If no respon	ds in 500 ms,	means er
	14 A	6	D)	

## 5.2.3 Connecting the ET-7018Z (10 AI)

The ET-7018Z is an 10 AI channels Ethernet I/O module. The Win-GRAF demo project for this section is "demo\_ ET7018z.zip". Please refer <u>Section 5.2.1</u> to set up the module channels using the Internet Browser, and restore/open the demo project.

#### **Demo Description:**

This demo creates one Data Block to read 10 AI data.

1. Double click the first Data Block (<4> Read Input Registers) to open the setting window.

10 D	rivers							H	XX
間よ	14 (UN70) V					1	Minute1	Type DINT DINT	1
	Symbol	Operation	Offset	Mask	Storage	E	ET7018z_Al_00	INT	
	ET7018z_AI_00 ET7018z_AI_01	Data exchange Data exchange	0	FFFF FFFF	Default Default	E	ET7018z_AI_02	INT	
	ET7018z_AI_02 ET7018z_AI_03	Data exchange Data exchange	2 3	FFFF FFFF	Default Default	E	ET7018z_AI_04	INT	
17535	ET7018z_AI_04 ET7018z_AI_05	Data exchange Data exchange	4 5	FFFF FFFF	Default Default	E	ET7018z_AI_06	INT	
•	ET7018z_AL_06 ET7018z_AL_07	Data exchange Data exchange	6 7	FFFF FFFF	Default Default	E	ET7018z_Al_08	INT	
	ET7018z_AL_08 ET7018z AL 09	Data exchange Data exchange	8 9	FFFF FFFF	Default Default			DINT	]
	ET7018z_status	Error report	0	FFFF	Default	Name	e Valu	•	2
		Image: Symbol       ET7018z_Al_00         ET7018z_Al_00       ET7018z_Al_01         Image: Symbol       ET7018z_Al_02         ET7018z_AL_02       ET7018z_AL_03         Image: Symbol       ET7018z_AL_03         Image: Symbol       ET7018z_AL_03         Image: Symbol       ET7018z_AL_03         Image: Symbol       ET7018z_AL_04         Image: Symbol       ET7018z_AL_05         Image: Symbol       ET7018z_AL_06         Image: Symbol       ET7018z_AL_07         Image: Symbol       ET7018z_AL_08         Image: Symbol       ET7018z_AL_08         Image: Symbol       ET7018z_AL_09	Image: Symbol       Open MODBUS Master         Image: Symbol       Open MODBUS: 192.168.255.1:502         Image: Symbol       Operation         ET7018z_AL_00       Data exchange         ET7018z_AL_01       Data exchange         ET7018z_AL_02       Data exchange         ET7018z_AL_03       Data exchange         ET7018z_AL_04       Data exchange         ET7018z_AL_05       Data exchange         ET7018z_AL_06       Data exchange         ET7018z_AL_05       Data exchange         ET7018z_AL_05       Data exchange         ET7018z_AL_06       Data exchange         ET7018z_AL_09       Data exchange         ET7018z_AL_08       Data exchange         ET7018z_AL_08       Data exchange         ET7018z_AL_09       Data exchange         ET7018z_AL_08       Data exchange         ET7018z_AL_08       Data exchange         ET7018z_AL_08       Data exchange         ET7018z_AL_09       Data exchange         ET7018z_status       Error report	Image: Symbol       Open MODBUS: 192.168.255.1:502         Image: Symbol       Operation         Symbol       Operation         Image: Symbol       O	Image: Symbol       Open MODBUS: 192.168.255.1:502         Image: Symbol       Operation         Image: Symol	Image: Symbol       Open MODBUS: 192.168.255.1:502         Image: Symbol       Operation         Symbol       Operation         Image: Symbol	Image: Symbol       Open MODBUS: 192.168.255.1:502         Image: Symbol       Operation       Offset       Mask       Storage         Image: Symbol       Operation       Operation       Offset       Mask       Storage         Image: Symbol       Operation       Data exchange       1       FFFF       Default         Image: Strong Symbol       Data exchange       3       FFFF       Default       Image: Strong Symbol       Image:	Image: Symbol       Open MODBUS: 192.168.255.1:502         Symbol       Operation       Offset       Mask       Storage         ET7018z_AL_00       Data exchange       0       FFFF       Default       ET7018z_AL_01       ET7018z_AL_02         ET7018z_AL_02       Data exchange       2       FFFF       Default       ET7018z_AL_03       ET7018z_AL_04       Data exchange       3       FFFF       Default       ET7018z_AL_04       ET7018z_AL_05       Data exchange       5       FFFF       Default       ET7018z_AL_05       ET7018z_AL_06       Data exchange       5       FFFF       Default       ET7018z_AL_06       ET7018z_AL_07       Data exchange       6       FFFF       Default       ET7018z_AL_08       ET7018z_AL_08	Image: Symbol       Open MODBUS: 192.168.255.1:502       Minute 1       DINT         Symbol       Operation       Offset       Mask       Storage         ET7018z_AI_00       Data exchange       0       FFFF       Default       ET7018z_AI_00       INT         ET7018z_AI_01       Data exchange       1       FFFF       Default       ET7018z_AI_02       INT         ET7018z_AI_02       Data exchange       2       FFFF       Default       ET7018z_AI_02       INT         ET7018z_AI_03       Data exchange       3       FFFF       Default       ET7018z_AI_04       INT         ET7018z_AI_05       Data exchange       5       FFFF       Default       ET7018z_AI_05       INT         ET7018z_AI_06       Data exchange       5       FFFF       Default       ET7018z_AI_06       INT         ET7018z_AI_06       Data exchange       6       FFFF       Default       ET7018z_AI_06       INT         ET7018z_AI_07       Data exchange       7       FFFF       Default       ET7018z_AI_08       INT         ET7018z_AI_08       Data exchange       7       FFFF       Default       ET7018z_AI_08       INT         ET7018z_AI_08       Data exchange       8       FFFF </td

- Notice: 1. The value of the "Offset" starts from "0", but the Modbus address of the variable is the "Offset" value plus 1 (Base address).
  - 2. If set the "Operation" to "Error report", the "Offset" value of the variable (Data Type: DINT) must set to "0".
  - 3. If AI range is -32768 ~ 32767, please declare the data type as "INT" for the variable.

ODBUS Master	Request			
Request Description:			ОК	
Slave/Unit:	1 <b>•</b>	The Net-ID o (ET-7000) is t	f the Slave Uni fixed to "1".	it
<2> Read Inpu <3> Read Hold <4> Read Inpu	ut Bits ling Registers			
<4> Read Inpl <5> Write sinc		Read DI data		
Data block				
Base <u>a</u> ddress:	1	Start from th		
Nb items:	10	to read 10 ite	ems.	
Periodic:	50 ms	5000	Decidence	50
On call On change		(on error)	If error occu	every 50 ms. urs, wait 5 s to xt commend
Misc.				
<u>T</u> imeout:	500 m:			
Nb trials:	1	If no respond	ls in 500 ms, m	ieans error.
	50			

## 5.2.4 To Disable/Enable the Modbus TCP/UDP Master Port

The Modbus TCP/UDP Master ports which are enabled in the Win-GRAF "Fieldbus Configuration" -"IO Drivers" setting window, will automatically work after the PAC is powered on. If user wants to disable one of the Modbus TCP Master ports, use the **"MBTCP\_M\_disable"** function (and use the **"MBUDP\_M\_disable"** function for UDP), see below:

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

In the above code, when you set "To\_disable" as "TRUE", it will disable the Modbus TCP Master port which connects to the slave device with the IP address "192.168.71.9" (TCP Port\_No = 502). And later, you can enable it again by using the "**MBTCP\_M\_enable**" function (using the "**MBUDP\_M\_enable**" function for UDP), see below:

(\* Declare To\_enable as BOOL
 Status\_tcp as BOOL \*)
If To\_enable then
 To\_enable := FALSE ;
 MBTCP\_M\_enable ( '192.168.71.9' , 502 ) ;
End\_if;
Status\_tcp := MBTCP\_M\_status ( '192.168.71.9' , 502 ) ;

The "MBTCP\_M\_status" function (and "MBUDP\_M\_status" is for UDP) listed above is used to get the status of the Modbus TCP Master port, for example, enabled (True) or disabled (False).

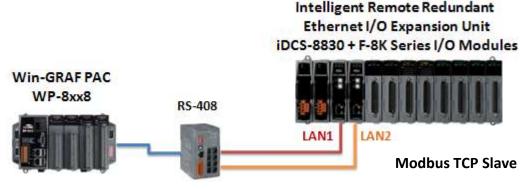
For easy maintenance, user can declare a STRING variable (set its length as "20"). For example, declare one "IP\_addr2" variable and set its initial value as "192.168.71.9". Then you can use it as the following code.

If To\_disable then
To\_disable := FALSE ;
MBTCP\_M\_disable ( IP\_addr2 , 502 ) ;
End\_if;
Status\_tcp2 := MBTCP\_M\_status ( IP\_addr2 , 502 ) ;

## 5.3 Connecting the Modbus TCP Slave device has two IP addresses

The previous section lists the way to enable the Win-GRAF PAC as a Modbus TCP Master device, and lists the way to read/write Modbus TCP Slave device. This section will list the way to create the redundant "Modbus Master Request", when one IP of the Modbus TCP Slave devices is disconnected, the other IP can still normally to be read/written data.

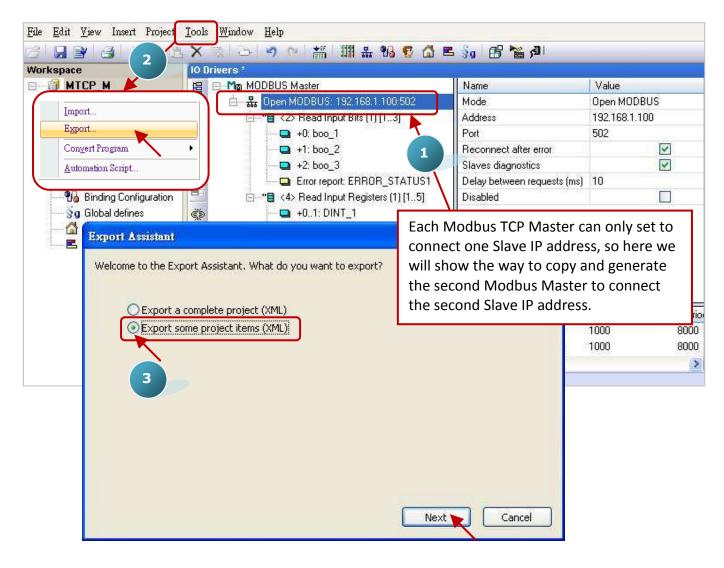
#### **Application Diagram:**



**Modbus TCP Master** 

#### Follow The Steps:

- 1. Click the "Open MODBUS:", and then click the menu bar "Tools" > "Export".
- 2. In the "Export Assistant" window, click "Export some project items (XNL)" and "Next".



3. Check the "Configurations" and uncheck all other items, and then click "Next".

Export Assistant	
Select items to be exported:   Programs and UDFBs  Main  DOnBadIndex  DOnBadIndex  DOnDivZero	
PShutDown     pStartup     Data type:     Configurations     MODBUS Master	
Export referenced global variables	Next Cancel

4. Browse a suitable path (default in C:\Win-GRAF\Projects) and named for the file (e.g., Master\_2.xml), and then click "Save" button. Finally, click "Finish" to export the settings.

	另存新檔		? 🛛
	儲存於①:	🗁 Projects	N 0 🖉 📂 🖽 -
Export Assistant Export is complete.	<ul> <li>         我最近的文件         <ul> <li></li></ul></li></ul>	Demo01 ICPDAS_template MTCP_M other Test Test_01 test_1 Test_2 Test_2 Test_3 test_0528	
	我的電腦	檔名(M): Master_2.xml 存檔類型(I): XML files (*.xml)	<ul> <li>▲</li> <li>▲</li> <li>●</li> <li>●</li></ul>
		Finish	

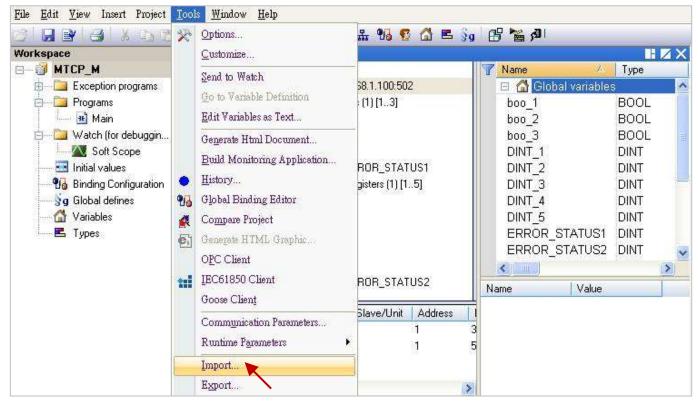
5. Using the Notepad software to open the file ".xml" that exported in the step 4, and then copy the content between the <modbus> and </modbus>.

案(F) 編輯(E) 格式(O) 檢視(Y) 説明(H)		
<types></types>		
<variables></variables>		
<vargroup kind="GLOB&lt;/td&gt;&lt;td&gt;.L" name="(Global)"></vargroup>		
<pre><vargroup kind="RETA&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;N" name="(Retain)"></vargroup></pre>		
<modbus></modbus>		
	"192.168.1.100" port="502" sdiags="1" reconnect="1" disabled="0" regdelay="10" rtu="0">	
	192.106.1.100 port= 302 solays= 1 reconnect= 1 disabled= 0 requeray= 10 rtd= 0 > Input Bits" address="0" nbitem="3" slave="1" call="Periodic(1000,8000)" timeout="1000" retry	v = 0.1 $n > 1$
	"O" symbol="boo_1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
	"1" symbol="boo_2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
	"2" sýmbol="boo_3" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
<modbusitem offsi<="" ope="Status" td=""><td>t="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" Sig</td><td>Lo=""/&gt;</td></modbusitem>	t="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" Sig	Lo=""/>
	(nput Registers" address="0" nbitem="5" slave="1" call="Periodic(1000,8000)" timeout="1000"	
	"O" symbol="DINT_1" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
	"2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
	"4" symbol="DINT_3" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
	"6" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> "8" symbol="DINT_5" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>	
	t="0" symbol="ERROR_STATUS2" nbword="1" mask="ffff" RangeHi="" RangeLo= " SigHi="" Sig t="0" symbol="ERROR_STATUS2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" Sig	
	k- U symbol- Ekkok_STATOSE hbword- i mask- nn kangern- kangebb- Sigin- Sig	10- 72
	Сору	

6. Paste the copied content above the </modbus>, and change the address to the second IP address of the Modbus Slave device (e.g., "192.168.1.101"), then save and close the file.

案(F) 編輯(E) 格式(O) 枝	縄(V) 説明(H)
新增(M) Ctd+N 期啓舊檔(O) Ctd+O 諸存檔案(O) Ctd+S	master" address="192.168.1.100" port="502" sdiags="1" reconnect="1" disabled="0" reqdelay="10" rtu="0"> =="" mode="Read Input Bits" address="0" nbitem="3" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1"> pe="Data" offset="0" symbol="boo_1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigLo=""/>
另存新檔( <u>d</u> ) 設定列印格式( <u>U</u> ) 列印(P) Ctil+P	– pe="Data" offset="1" symbol="boo_2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigLi="" SigLo=""/> pe="Data" offset="2" symbol="boo_3" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigLi="" SigLo=""/> pe="Status" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" RangeLo="" SigLi="" SigLi=""/>
<modbusitem <modbusitem <modbusitem <modbusitem< th=""><th><pre>b="" mode="Read Input Registers" address="0" nbitem="5" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1" ope="Data" offset="0" symbol="DINT_1" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="4" symbol="DINT_3" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt; ope="Data" offset="6" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt; ope="Data" offset="8" symbol="DINT_5" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt; ope="Status" offset="0" symbol="ERROR_STATUS2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt;</pre></th></modbusitem<></modbusitem </modbusitem </modbusitem 	<pre>b="" mode="Read Input Registers" address="0" nbitem="5" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1" ope="Data" offset="0" symbol="DINT_1" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="4" symbol="DINT_3" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt; ope="Data" offset="6" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt; ope="Data" offset="8" symbol="DINT_5" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt; ope="Status" offset="0" symbol="ERROR_STATUS2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo="'/&gt;</pre>
<modbusreq nai<br=""><modbusitem <modbusitem <modbusitem <modbusitem <modbusreq nai<br=""><modbusreq nai<br=""><modbusitem <modbusitem< th=""><th>="master" address="192.168.1.101" port="502" sdiags="1" reconnect="1" disabled="0" reqdelay="10" rtu="0"&gt; ne="" mode="Read Input Bits" address="0" nbitem="3" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1"&gt; ope="Data" offset="0" symbol="boo_1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="1" symbol="boo_2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="2" symbol="boo_3" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Status" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Status" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ne="" mode="Read Input Registers" address="0" nbitem="5" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1" ope="Data" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="0" symbol="DINT_1" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="0" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="2" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="0" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="0" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt; ope="Data" offset="6" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/&gt;</th></modbusitem<></modbusitem </modbusreq></modbusreq></modbusitem </modbusitem </modbusitem </modbusitem </modbusreq>	="master" address="192.168.1.101" port="502" sdiags="1" reconnect="1" disabled="0" reqdelay="10" rtu="0"> ne="" mode="Read Input Bits" address="0" nbitem="3" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1"> ope="Data" offset="0" symbol="boo_1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="1" symbol="boo_2" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="2" symbol="boo_3" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Status" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Status" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ne="" mode="Read Input Registers" address="0" nbitem="5" slave="1" call="Periodic(1000,8000)" timeout="1000" retry="1" ope="Data" offset="0" symbol="ERROR_STATUS1" nbword="1" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="0" symbol="DINT_1" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="2" symbol="DINT_2" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="0" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="2" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="0" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="0" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/> ope="Data" offset="6" symbol="DINT_4" nbword="32770" mask="ffff" RangeHi="" RangeLo="" SigHi="" SigLo=""/>

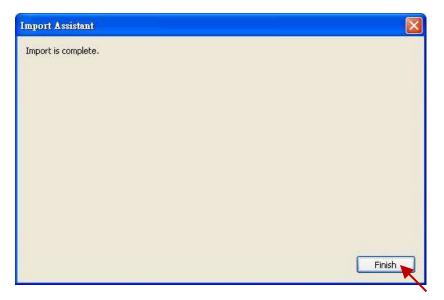
7. Click the Win-GRAF menu bar "Tools" > "Import".



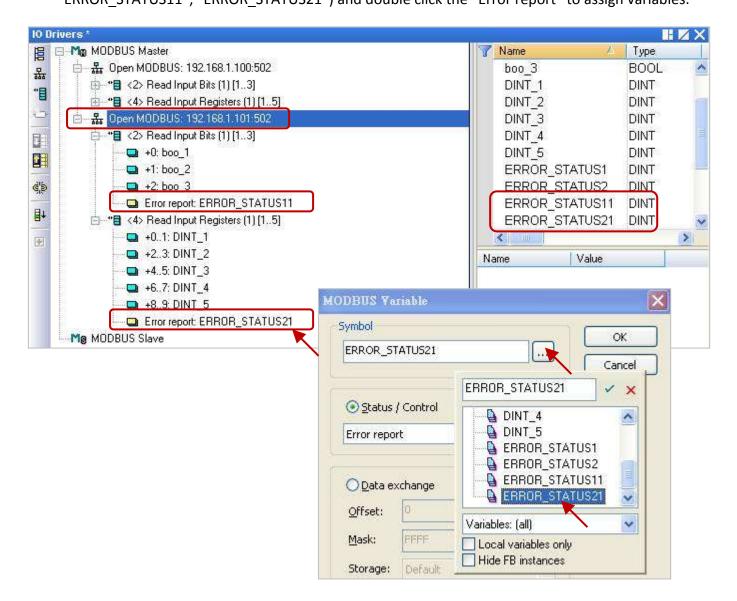
- 8. In the "Import Assistant" window, click "Import project items (XNL)" and "Next".
- 9. Select the file you want to import (e.g., "Master\_2.xml") and click "Open" button.

		Import Assistant	
雪 調 雪 調 ① : : : : : : : : : : : : :	Projects Demo01 ICPDAS_tem MTCP_M other Test Test Test_01 test_1	Welcome to the Import Assistant. What do you want to import?	cel
あ的文件 表的電腦 教的電腦 網路上的芳鄰	<ul> <li>Test_2</li> <li>Test_3</li> <li>test_0528</li> <li>Master_2.xm</li> <li>Master_2.xm</li> <li>Control and the state of the stat</li></ul>	9 Master_2.xml Master_2.xml 区(*.xml) 区(*.xml)	

10. Click "Finish" to finish the import action.



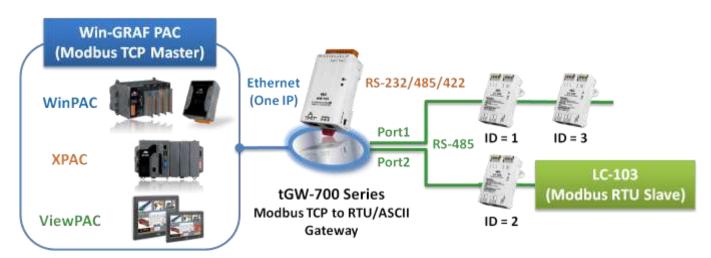
11. In the "IO Drivers" window, there has added a new "Open MODBUS" setting item that includes two "Modbus Master Request" for using to read the DI and AI. One "Error report" is used to check the IP connecting status, so please add two "DINT" variables in the Variable area (e.g., "ERROR STATUS11", "ERROR STATUS21") and double click the "Error report" to assign variables.



# 5.4 Connecting the tGW-700 to Expand Modbus RTU Master Ports

If using the Modbus RTU (RS-232/485/422) device to transmit data in a long-distance application area, the user will normally choose a lower baud rate for better signal quality. But, using this way will cause low transmission efficiencies. In order to improve this problem, ICP DAS releases the tGW-700 series products (tiny Modbus TCP to RTU/ASCII gateway) for converting Ethernet/RS-485 signals so that the user can reduce the RS-485 cable lengths and solve the issue with inefficient communications.

This section will provide a demo program (demo\_tgw725.zip) to describe how the Win-GRAF PAC communicates with LC-103 modules via the tGW-700 gateway (as the figure below).



## 5.4.1 Using the tGW-700 Series (Modbus TCP to Modbus RTU/ASCII Gateway)

**The tGW-700 module** is a Modbus TCP to RTU/ASCII gateway that enables a Modbus TCP host (e.g., WP-8xx8) to communicate with serial Modbus RTU/ASCII devices through an Ethernet network, and eliminates the cable length limitation of legacy serial communication devices. Visit the tGW-700 series webpage for more information on

http://www.icpdas.com/root/product/solutions/industrial\_communication/pds/tgw-700.html

#### tGW-700 series User Manual

<u>http://ftp.icpdas.com/pub/cd/tinymodules/napdos/tgw-700/document/</u> (See the chapter 3 & chapter 4 to know the way of network setting, testing and web function configuration for the tGW-700 module.)

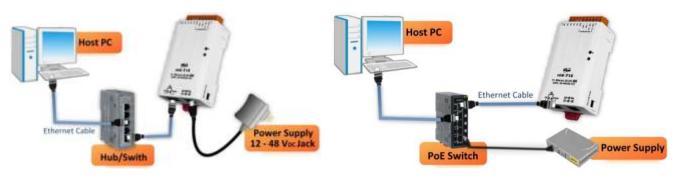
Before using the tGW-700, the user must configure its network and COM Port setting:

#### • Connect the Power Supply and the Host PC

1. Check that the Init/Run switch is in the "Run" position.



2. Connect both the tGW-700 and the Host computer to the same sub-network or the same Ethernet Switch, and then power on the tGW-700.



+12 to +48 V<sub>DC</sub> jack Power Supply (Non-PoE)

PoE Power Supply

Install the "eSearch Utility", and then Search and Configure the Network Setting for the tGW-700
 <u>http://ftp.icpdas.com/pub/cd/tinymodules/napdos/software/modbus\_utility/</u>

<mark>⊯ eSearch Utility [ v1.1.7</mark> <u>File S</u> erver <u>T</u> ools	, Mar.30, 201	5]		[		
Name	Alias	IP Address	Sub-net Mask	Gateway	MA	
tGW-725	Tiny	10.10.10.100	255.255.255.0	) 0.0.0.0	00	
DL-302	EtherIU	192.168.11.9	255.255.0.0	192.168.1.1	00:	
DL-302	EtherIO	192.168.17.25	255.255.0.0	192.168.1.1	00:	
Web LED	N/A	192.168.11.10	255.255.0.0	192.168.1.1	00:	
Web LED	N/A	192.168.17.42	255.255.0.0	192.168.1.1	00:	
.10	test1	192.168.11.8	255.255.0.0	192.168.1.1	00:	
7186E3	I-7540D	192.168.17.40	255.255.0.0	192.168.17.40	00:	
ET-7067/PET-7067	N/A	192.168.17.44	255.255.0.0	192.168.0.1	00:	
SV2201	Compact	192.168.82.100	255.255.0.0	192.168.1.1	78: 🤜 👘	
< 1 Search Server	- (	2 Juration (UDP)	🔗 Web	Exit	>	
Status						
earch your tGW-700.	Sot the	tGW-700's IP / N	Aack Open	the tCN/ 700 M/	h Sorvo	
	11		·	Open the tGW-700 Web Server.		
		/ Gateway. (Contact your Network Administrator to		( <u>Note:</u> Both the tGW-700's and PC's IP addresses must on the		
	Netwo					
	get cor	get correct configuration)		sub-network. See	e chapte	

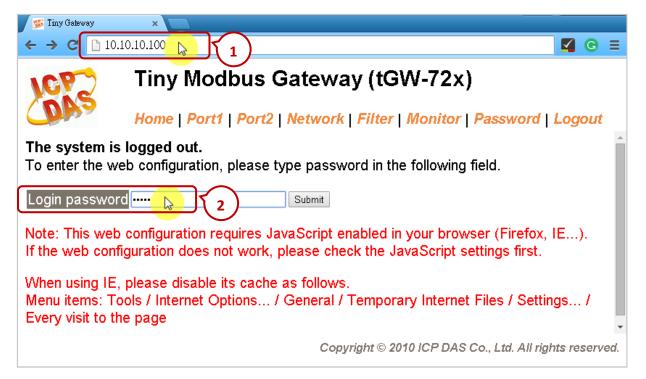
Please contact your Network Administrator to get the correct IP, Mask and Gateway addresses. After completing these settings, click the "OK" button and they will take effect within 2 seconds.

Configure Server (	UDP)				X	
Server Name :	tGW-725					
DHCP:	0: OFF 🗨	Sub-net Mask :	255.255.255.0	Alias:	Tiny	
IP Address :	10.10.10.100	Gateway :	0.0.0.0	MAC:	00:0d:e0:80:32:f2	
Warning!! Contact your Network Administrator to get correct configuration before any changing! OK Cancel						

#### Web Configuration

You can refer the tGW-700 user manual (chapter 4) to view the configuration way for all features. The following will describe the COM Port setting.

- 1. Enter the tGW-700's IP address on the web browser. (Note: It must on the same sub-network with your PC's IP).
- 2. Enter the password (the factory default password is "admin").



3. After logging in, the main page (Home) will display the current port setting. The user can also click "Port1" or "Port2" tab to modify the settings.

ICP)	Tiny Modbus Gate	eway (tGW-72x)				
DAS	Home   Port1   Port2   Netw	ne   Port1   Port2   Network   Filter   Monitor   Password   Logout 3				
1	Initial Switch OFF		System Timeout (Network Watchdog, Seconds)			
Current po	rt settings:					
Part Setting	f	Port 1	Port 2			
	Baud Rate (bps)	9600	9600			
	Data Size (bits)	8	8			
	Parity	None	None			
	Stop Bits (bits)	1	1			
	Modbus Protocol	RTU	RTU			
	Slave Timeout (ms)	300	300			
	Char Timeout (bytes)	4	4			
	Silent Time (ms)	0	0			
	Read Cache (ms)	980	980			
	Local TCP Port	502	503			
	TCP Timeout (Seconds)	180	180			
Pair-Connec Master/Slav	tion Setting: /e Mode)	Port 1	Port 2			
	Server Mode	Server	Server			
	Remote Server IP		-			
	Remote TCP Port	-	-			
	RTU Virtual ID	-	-			
	TCP Slave ID	-	-			

## 5.4.2 Connecting the tGW-700 Series and the LC-103 module (1 DI, 3 Relay)

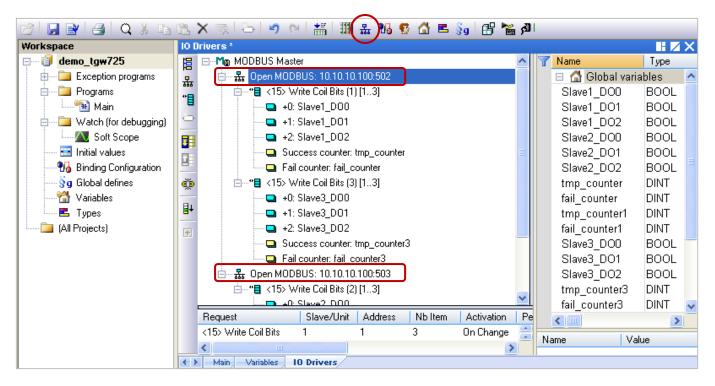
In this section, we provide a demo project (demo\_tgw725.zip) to describe how the Win-GRAF PAC communicates with LC-103 modules via the tGW-725 (the Modbus TCP to Modbus RTU/ASCII gateway with two RS-285 ports). You can run the Win-GRAF Workbench and click "File  $\rightarrow$  Add Existing Project  $\rightarrow$  From Zip..." to open this project in the Win-GRAF PAC CD (CD-ROM: \Napdos\Win-GRAF\demo-project).

**The LC-103** is an easy-to-use lighting control module that supports the Modbus RTU protocol and provides 1 channel for digital input and 3 channels for relay output. Before using this module, set its ID No. depends on your application needs, for example, if the required ID is "1", simply adjust the rotary switch to "1" at the bottom of the module. Visit the LC-103 webpage for more detailed information: http://www.icpdas.com/root/product/solutions/remote\_io/rs-485/lighting\_control/lc-103.html

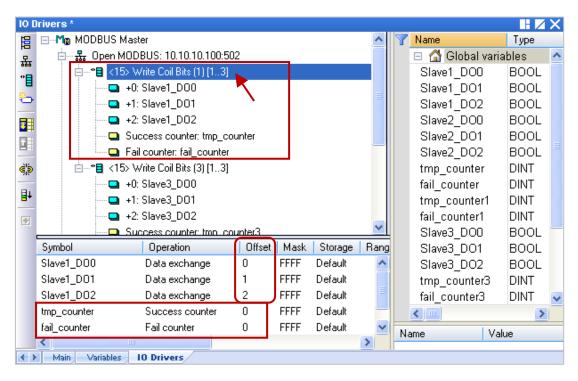


**Demo Project Description:** (Refer the <u>Section 5.2</u> to know how to create this project.)

1. Click the "Open Fieldbus Configuration" tool button to open the "IO Drivers" window.



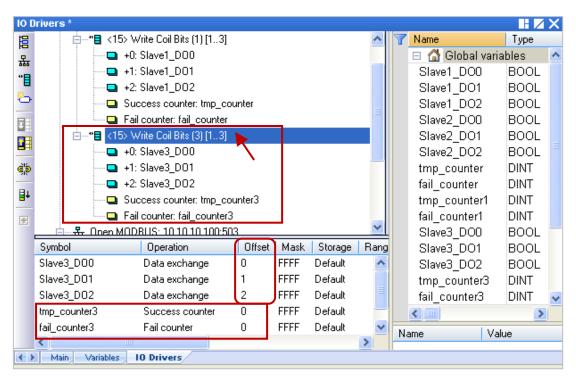
On the screen, the "Open MODBUS: 10.10.10.100:**502** / Open MODBUS: 10.10.10.100:**503**" means that the tGW-725's IP address is "10.10.10.100" and using two COM ports (RS-485) No. - "502" and "503". And, there are two LC-103 modules (Slave ID = "1" and "3") connected to its COM1 and one LC-103 connected to the COM2 (Slave ID = 2). The following will describe the configuration way of each Modbus Master Request one-by-one. 2. Mouse double-click the first data block under the COM1 (Port = 502) to view this Modbus Master request. In this example, the Win-GRAF PAC (Modbus TCP Master) send three DO commands to the LC-103 (Slave ID = 1) via the tGW-725's COM1 (Port = 502). As the figure below, the "Operation" is set to "Success counter" (or "Fail counter") that means this variable value will add 1 if the command was successfully sent (or failed). Moreover, the "Offset" value of these variables must set as "0".



Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address).

MODBUS Master	Request			3
Request Description:			ОК	
Slave/Unit:	1 <b>•</b>		e.g., "1" ) of tl (i.e., LC-103).	
	ile holding register	^		
	Il Bits Idipa Degisters	Write DO da	ta.	
Data block	[			
Base <u>a</u> ddress: <u>N</u> b items:	3	Start from ac write three o		]
Activation				
O Periodic:	0 ms	0		
On call		(on error)		
On change		Write only w	/hen data cha	nged.
Misc.				
<u>T</u> imeout:	3000 ms	An exception respond for	n occurs if no	
No criais.	-		5000 ms.	

3. Mouse double-click the 2nd data block under the COM1 (Port = 502) to view this Modbus Master request. In this example, the Win-GRAF PAC (Modbus TCP Master) send three DO commands to the LC-103 (Slave ID = 3) via the tGW-725's COM1 (Port = 502). As the figure below, the "Operation" is set to "Success counter" (or "Fail counter") that means this variable value will add 1 if the command was successfully sent (or failed). Moreover, the "Offset" value of these variables must set as "0".



Note: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address).

MODBUS Master	Request			
Request Description:			ОК	
Slave/Unit:	3		e.g., "3" ) of th (i.e., LC-103).	ie
<5> Write singl <6> Write singl <15> Write Co	le holding register	^		
212 S. Weita Hal		Write DO da	ita.	
Data block Base <u>a</u> ddress:	1			
<u>N</u> b items:	3	Start from a write three of		
Activation O Periodic: O On call	0 ms	0 (on error)		
⊙ On change ●		Write only w	vhen data char	nged.
Misc.	0000			
<u>T</u> imeout: Nb trials:	3000 ms	An exception respond for	n occurrs if no 3000 ms.	

4. Mouse double-click the data block under the COM2 (Port = 503) to view this Modbus Master request. In this example, the Win-GRAF PAC (Modbus TCP Master) send three DO commands to the LC-103 (Slave ID = 2) via the tGW-725's COM2 (Port = 503). As the figure below, the "Operation" is set to "Success counter" (or "Fail counter") that means this variable value will add 1 if the command was successfully sent (or failed). Moreover, the "Offset" value of these variables must set as "0".

10 D	rivers *									×
12	🖸	+1: Slave1_D01				~	Y	Name	Туре	
쁆	🖸	+2: Slave1_D02						🖃 🚮 Global varia	ables	^
*8	📟	Success counter: tmp_co	unter					Slave1_DO0	BOOL	
-	· · · · · · · · · · · ·	Fail counter: fail_counter						Slave1_DO1	BOOL	
0	i⊒*∎ <1	5> Write Coil Bits (3) [13]						Slave1_DO2	BOOL	
	🗅	+0: Slave3_D00						Slave2_DO0	BOOL	
	🗅	+1: Slave3_D01						Slave2_DO1	BOOL	
_	🗅	+2: Slave3_D02				≡		Slave2_DO2	BOOL	
ġ,þ	🗅	Success counter: tmp_co	unter3					tmp_counter	DINT	
		Fail counter: fail_counter3						fail_counter	DINT	
₽+	😑 🖓 🖓 Open N	MODBUS: 10.10.10.100:50	3					tmp_counter1	DINT	
	i⊡…* <b>⊟</b> <1	5> Write Coil Bits (2) [13]				-		fail_counter1	DINT	
			<u></u>		L et			Slave3_DO0	BOOL	
	Symbol	Operation	Offset	Mask	Storage	Rang		Slave3_DO1	BOOL	
	Slave2_D00	Data exchange	0	FFFF	Default			Slave3_DO2	BOOL	
	Slave2_D01	Data exchange		FFFF	Default			tmp_counter3	DINT	
	Slave2_D02	Data exchange	2	FFFF	Default			fail_counter3	DINT	~
	tmp_counter1	Success counter	0	FFFF	Default			<	>	
	fail_counter1	Fail counter	0	FFFF	Default	_	N	ame Va	ue	
	<		_			>				
< >	Main Variables	10 Drivers								

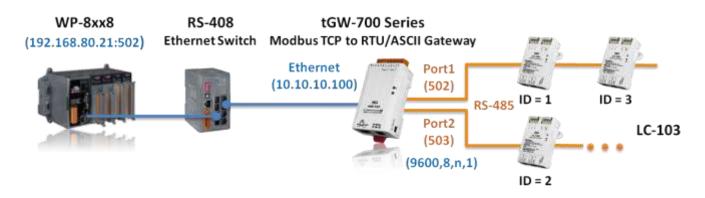
<u>Note</u>: The "Offset" value starts at "0" and the Modbus address of variable is equal to the "Offset" value plus 1 (Base address).

MODBUS Master Request	
Request Description:	ОК
Slave/Unit: 2	The Net-ID (e.g., "2") of the Slave device (i.e., LC-103).
<5> Write single coil bit <6> Write single holding regis	ster
<15> Write Coil Bits <216> Write Halding Desistory	Write DO data.
Data block	
Base <u>a</u> ddress: 1	
<u>N</u> b items: 3	Start from addr. 1 and write three data.
Activation	
O Periodic: 0	ms O
◯ On caḷ	(op error)
⊙ On change ●	Write only when data changed.
Misc.	
Timeout: 3000	An exception occurs if no
Nb trials: 1	respond for 3000 ms.

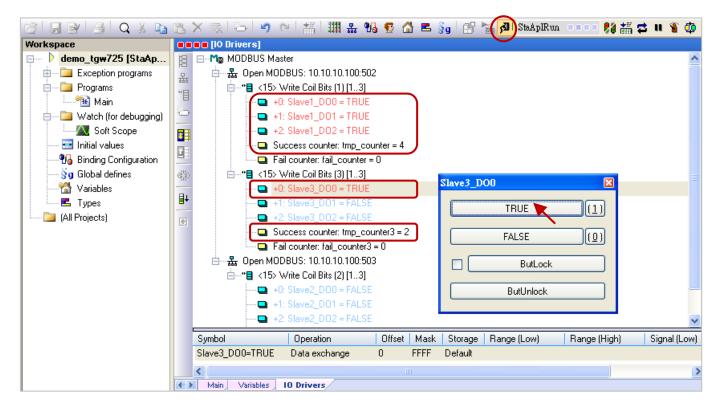
## 5.4.3 Test the Demo Project (demo\_tgw725.zip)

Before testing this demo project, download it to your Win-GRAF PAC. (If you're not familiar with it, refer the <u>Section 2.3.4</u> and <u>Section 2.3.5</u>.)

#### The Hardware Wiring:



After connecting with the Win-GRAF PAC, double-click on any DO variable and set it as "TRUE" in the "I/O Drivers" window. If the write operation is successful, then the "tmp\_counter" value will add "1".



<u>Note</u>: When the Win-GRAF PAC boots up, it will send the Modbus request to the Modbus Slave device. So, you can see the "tmp\_counter" value starts at "1" which means this data write is successful.

# Chapter 6 Retain Variable and Data Storage

## 6.1 Retain Variable

This chapter lists the way to use the "RETAIN\_VAR", "RETAIN\_ARY", "RETAIN\_FLAG\_GET", "RETAIN\_FLAG\_SET" and "RETAIN\_FLAG\_CLR" Functions. The Win-GRAF PACs are built-in the Retain memory for users to store the retain variable data that will not lose due to the PAC shutdown and can retain the last value at the next time reboot.

In the shipping CD (\Napdos\Win-GRAF\demo-project), you can find the demo project for this chapter (demo\_retain.zip), please refer <u>Chapter 12</u> to restore this project (Execute File> Add Existing Project > From Zip) and set the current IP address of your PAC.

<u>Note:</u> Function "Retain\_Var()" or Retain\_Ary() can only be used in the first PAC Cycle or in the Cycle that performs the On-line Change. If use them in other Cycle, it will return "FALSE". If the Retain Variable has not assigned any initial value and the PAC calls the Function, the return value is not meaningful; users need to assign appropriate initial values to all Retain Variables at least once.

#### ST Program:

This demo uses Function Retain\_Var() and Retain\_Ary().

```
(* "on line change cycle" is declared as DINT
                         (nonezero means it is in the cycle just after doing on line change).
  "retain done" is declared as BOOL and inited as FALSE.
  "tmp bool" is declared as BOOL.
 *)
 on line change cycle := GetSysInfo ( SYSINFO CHANGE CYCLE);
 if (retain done = FALSE) or (on line change cycle <> 0) then
     retain_done := TRUE ; (* just do it one time *)
     tmp bool := Retain Var (DINT 1, 1); (* retain a DINT variable *)
     tmp bool := Retain Var ( DINT 2, 2);
     tmp_bool := Retain_Var ( REAL_1 , 3) ; (* retain a REAL variable *)
     tmp bool := Retain Var (BOOL 1, 4); (* retain a BOOL variable *)
     tmp_bool := Retain_Var ( BOOL_2 , 5) ;
     (* retain 10 elements of an INT array variable at retain addr starting at 6. *)
     tmp bool := Retain Ary (INT ARY, 6, 10);
     (* retain 20 elements of a REAL array variable at retain addr starting at 16. *)
     tmp_bool := Retain_Ary ( REAL_ARY , 16 , 20) ;
     tmp bool := Retain Var (DINT 3, 36);
     (* 64-bit variable can use only addr from 10,001 to 12,000 *)
     tmp_bool := Retain_Var (LINT_1 , 10001); (* retain a LINT variable (64-bit) *)
     tmp bool := Retain Var (LREAL 1, 10002); (* retain a LREAL variable (64-bit) *)
   end if;
```

You can check/set variables in the "Variables" window, if you are not familiar with the way of the variable declaration, please refer <u>Section 2.2.2</u> and <u>Section 2.3.1</u>.

Name /	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	De
🗆 🚮 Global variables 👘								~
BOOL_1	BOOL							
BOOL 2	BOOL							
DINT 1	DINT							
DINT 2	DINT							
DINT3	DINT							
INT_ARY	INT	[09]						
LINT_1	LINT							
LREAL 1	LREAL							
on_line_change_cycle	DINT							
REAL_1	REAL							
REAL_ARY	REAL	[0 101	1					
retain_done	BOOL	20	-					
tmp_bool	BOOL							Y

<u>Note</u>: ICP DAS Win-GRAF PAC does not support the "RETAIN variables" Functions in the "Variables" window, so please refer the five Functions in the following sections to use the Retain Variables.

Variables									Х
🝸 Name	👃 Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description	
🗉 🛃 RETA	IN variables 🅖							1	~
NewVar	BOOL								~
<		$\sim$		1				>	

## 6.1.1 RETAIN\_VAR (Retain a Variable)

**RETAIN\_VAR** QD Name ddr

Tips: Press "F1" key to see more details.

#### Name:

A variable name (DO NOT use Array variable or String).

Variable type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT or LREAL.

#### Addr:

Data Type: DINT. The address number for retaining the variable, can be 1 to 12,000.

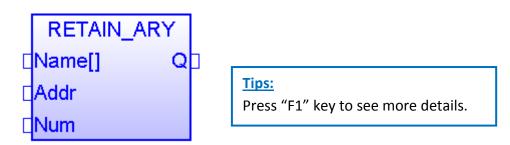
#### Q:

Data Type: BOOL. TRUE: Ok; FALSE: Error.

#### Note:

- One Addr can accept only one variable (or one element of the array).
   DO NOT assign the same Addr to two variables (or more), or the Retain Value will be wrong.
- 2. 64-bit data type (LINT or LREAL) can use only the Addr No. from 10,001 to 12,000.
- 3. Other data type (BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL or TIME) can use the Addr No. from 1 to 12,000.

## 6.1.2 **RETAIN\_ARY (Retain an Array Variable)**



#### Name[]:

An ARRAY variable name (DO NOT use String).

Variable type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT or LREAL.

#### Addr:

Data Type: DINT. The starting address number for retaining the array variable; can be 1 to 12,000.

#### Num:

Data Type: DINT. The amount of elements in the Array variable to be retained.

For example:

If there are 100 elements in an Array variable, set "Num" to "1 to 100" is correct, but if set it more than 100 that is not correct.

If there are 5 elements in an Array variable, set "Num" to "1 to 5" is correct, but if set it more than 5 that is not correct.

#### Q:

Data Type: BOOL. TRUE: Ok; FALSE: Error.

#### Note:

- One Addr can accept only one variable (or one element of array).
   DO NOT assign the same Addr to two variables (or more), or the Retain Value will be wrong.
- 2. 64-bit data type (LINT or LREAL) can use only the Addr No. from 10,001 to 12,000.
- 3. Other data type (BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL or TIME) can use the Addr No. from 1 to 12,000.

## 6.1.3 RETAIN\_FLAG\_SET/GET/CLR (Set/Get/Clear the Retain Flag)

#### How to Use:

The "Retain Flag" is a flag (TRUE/FALSE) stored by users in the retain memory. Users can set this retain flag to indicate "All retain data has been assigned a proper value before". When a PAC starts without setting a proper value to retain variable before, the data of the retain variable read from the retain memory is not correct (it is normally a strange value). So users have to assign proper value to all retain variable at least once to let the application work well. Then after user can call the "Retain\_Flag\_Set()" to set the retain flag. It means "All retain data has been assigned a proper value".

To get the state of the Retain Flag, please call "Retain\_Flag\_Get()". To clear the state of the Retain Flag, please call "Retain Flag\_Clr()".

#### ST Program:

```
(* "on_line_change_cycle" is declared as DINT
  (nonezero means it is in the cycle just after doing on line change) .
  "retain_done" is declared as BOOL and inited as FALSE .
  "tmp_bool", "retain_flag" and "to_set_flag" are declared as BOOL.
 *)
```

on\_line\_change\_cycle := GetSysInfo (\_SYSINFO\_CHANGE\_CYCLE) ;

```
if (retain_done = FALSE) or (on_line_change_cycle <> 0) then
  retain_done := TRUE; (* just do it one time *)
  tmp_bool := Retain_Var( DINT_1, 1); (* retain a DINT variable *)
  tmp_bool := Retain_Var( DINT_2, 2);
  tmp_bool := Retain_Var( REAL_1, 3); (* retain a REAL variable *)
  tmp_bool := Retain_Var( BOOL 1, 4); (* retain a BOOL variable *)
```

```
(* ... After doing all the Retain Functions ... *)
```

retain\_flag := Retain\_Flag\_Get();

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

(\*If Retain variable does not set up any proper value, you can do some proper operation here. \*) (\* ... \*)

end\_if ;

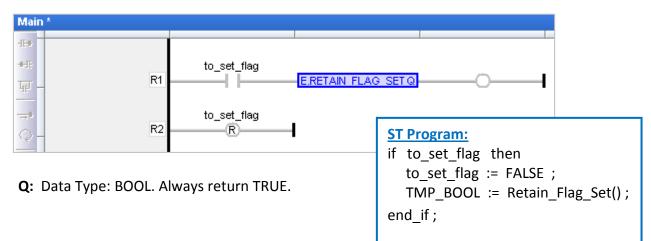
```
end_if ;
```

```
(* When all Retain variables are assigned proper values,
    remember to set the "to_set_flag" to "TRUE" for calling "Retain_Flag_Set() once,
    so that, when next time you use the "Retain_Flag_Get()", it can return "TRUE".
    *)
if (to_set_flag = TRUE) then
    to_set_flag := FALSE ;
    tmp_bool := Retain_Flag_Set() ;
end_if ;
```

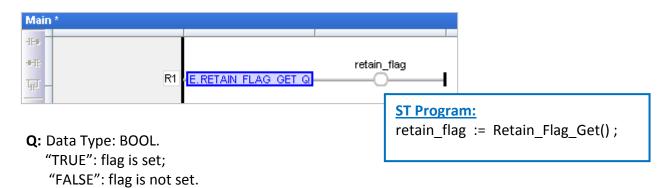
#### LD Program:

(Press "F1" key to see the detailed setting descriptions.)

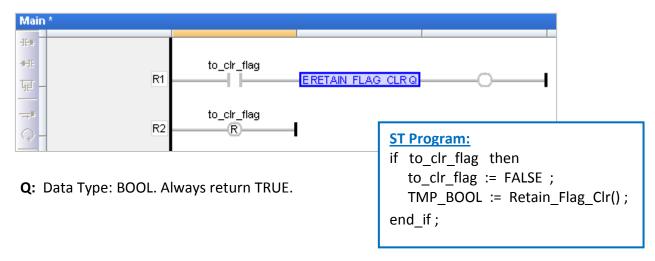
**RETAIN\_FLAG\_SET:** Set the retain flag.



#### **RETAIN\_FLAG\_GET:** Get the state of the retain flag.



#### RETAIN\_FLAG\_CLR: Clear the retain flag.

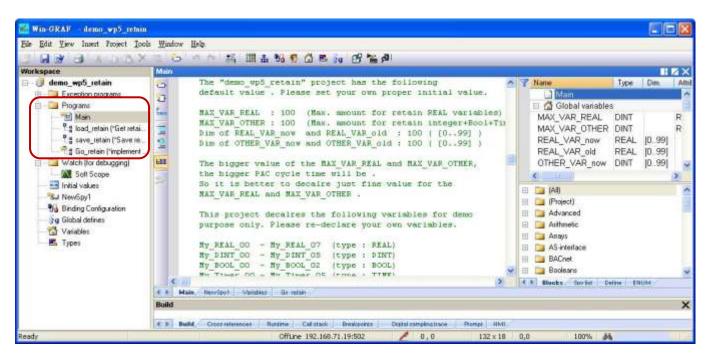


# 6.2 Retain Variable (Using files)

You can refer the following demo project (demo\_wp5\_retain.zip) to use retain variables with files in the \System\_disk\Win-GRAF\.

In the shipping CD (\Napdos\Win-GRAF\demo-project), you can find the demo project for this section (demo\_wp5\_retain.zip), please refer <u>Chapter 12</u> to restore this project (Execute File> Add Existing Project > From Zip) and set the current IP address of your PAC.

This project includes an ST main program (Main) and 3 ST sub-programs (load\_retain, save\_retain and Go\_retain).



#### Limitation :

This project is not good at handling Retain variables which value changes frequently. For example, value changed about every second or every minute. That is because these retain values of this project are saved within files in the \System\_Disk. The file operations in it consume more CPU time, which will slow down the PAC performance if retain value changed frequently.

The "demo\_wp5\_retain" project has the following default values. Please set your own proper initial values.

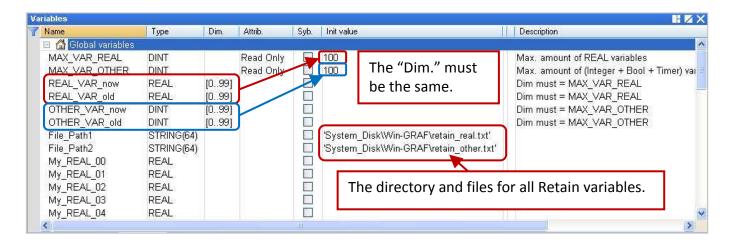
MAX\_VAR\_REAL: 100 (Max. amount for retaining REAL variables)
MAX\_VAR\_OTHER: 100 (Max. amount for retaining integer+Bool+Timer variables)
Dim of REAL\_VAR\_now and REAL\_VAR\_old: 100 ([0..99], the same as the value "MAX\_VAR\_REAL")
Dim of OTHER\_VAR\_now and OTHER\_VAR\_old: 100 ([0..99], the same as the value "MAX\_VAR\_OTHER")

<u>Note:</u> The bigger the value of the MAX\_VAR\_REAL and MAX\_VAR\_OTHER, the larger the PAC cycle time will be. So it is better to declare just fine value for the MAX\_VAR\_REAL and MAX\_VAR\_OTHER.

This project declares the following variables for demo purpose only. Please re-declare your own variables.

My\_REAL\_00 ~ My\_REAL\_07 (Data Type: REAL) My\_DINT\_00 ~ My\_DINT\_05 (Data Type: DINT) My\_BOOL\_00 ~ My\_BOOL\_02 (Data Type: BOOL) My\_Timer\_00 ~ My\_Timer\_05 (Data Type: TIME)

You can see more variables in the "Variables" window.



"Go\_retain" sub-program is used to do the retain operation. Remember to modify this sub-program. There are 4 sections should to be modified in it. Please search "Add your REAL variables for retain here" and "Add your integer, BOOL, Timer variables for retain here", and depend on your re-declared variables to modify your "Go\_retain" sub-program.

```
(* Add your REAL variables for retain here *)
(* ------ *)
My_REAL_00 := REAL_VAR_now[0];
My_REAL_01 := REAL_VAR_now[1];
My_REAL_02 := REAL_VAR_now[2];
My_REAL_03 := REAL_VAR_now[3];
My_REAL_04 := REAL_VAR_now[3];
My_REAL_05 := REAL_VAR_now[4];
My_REAL_06 := REAL_VAR_now[5];
My_REAL_06 := REAL_VAR_now[6];
My_REAL_07 := REAL_VAR_now[7];
(* ------ *)
```

(\* Add your integer, BOOL, Timer variables for retain here \*)
(\* ...... \*)
My\_DINT\_00 := OTHER\_VAR\_now[0];
My\_DINT\_01 := OTHER\_VAR\_now[1];
My\_DINT\_02 := OTHER\_VAR\_now[2];
My\_DINT\_03 := OTHER\_VAR\_now[3];
My\_DINT\_04 := OTHER\_VAR\_now[4];
My\_DINT\_05 := OTHER\_VAR\_now[5];

```
My_BOOL_00 := Any_to_BOOL( OTHER_VAR_now[6] );
My_BOOL_01 := Any_to_BOOL( OTHER_VAR_now[7] );
My_BOOL_02 := Any_to_BOOL( OTHER_VAR_now[8] );
My_Timer_00 := Any_to_TIME( OTHER_VAR_now[9] );
```

#### (\* Add your REAL variables for retain here \*)

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

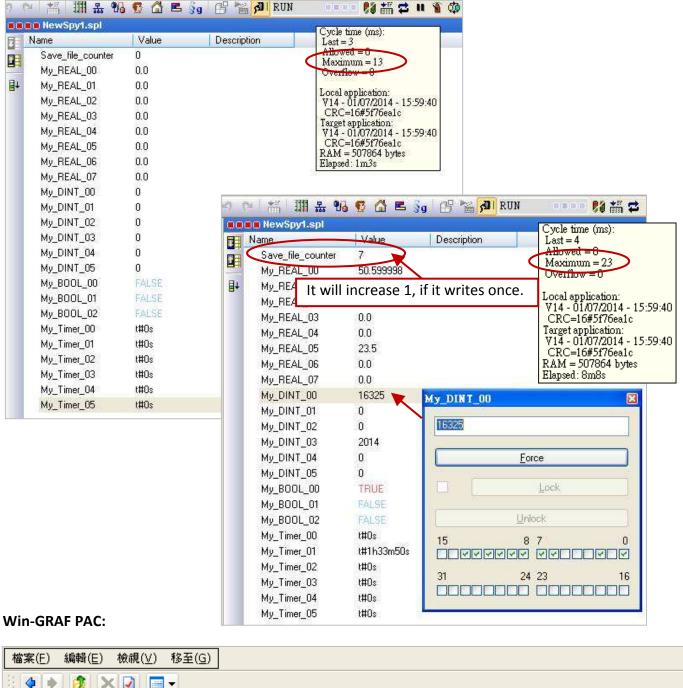
(\* Add your integer, BOOL, Timer variables for retain here \*)

```
(* .....*)
OTHER VAR now[0] := My DINT 00;
OTHER VAR now[1] := My DINT 01;
OTHER VAR now[2] := My_DINT_02;
OTHER VAR now[3] := My DINT 03;
OTHER VAR now[4] := My DINT 04;
OTHER VAR now[5] := My DINT 05;
OTHER VAR now[6] := Any to DINT(My BOOL 00);
OTHER VAR now[7] := Any to DINT(My BOOL 01);
OTHER_VAR_now[8] := Any_to_DINT( My_BOOL_02 );
OTHER_VAR_now[9] := Any_to_DINT( My_Timer_00 );
OTHER_VAR_now[10] := Any_to_DINT( My_Timer_01 ) ;
OTHER VAR now[11] := Any to DINT( My Timer 02 );
OTHER VAR now[12] := Any to DINT( My Timer 03 );
OTHER VAR now[13] := Any to DINT( My Timer 04 );
OTHER VAR now[14] := Any to DINT( My Timer 05 );
(* ......*)
```

#### **Test Project:**

Before testing, make sure you have already set up the PAC IP and compile/download the project into the PAC (refer <u>Section 2.3.4</u> and <u>Section 2.3.5</u>). When connecting with the PAC, all values in the Spy list (refer <u>Section 11.3</u>) will be "0" (or "FALSE") at the begging, please enter some values at will. When the value has changed, it will create a text file on the PAC (\System\_disk\Win-GRAF\retain\_real.txt and retain\_other.txt) and Write data to the files.

<u>Note:</u> The "Save\_file\_counter" will show the number of times the file is written, if this value is changing rapidly (e.g., to write several times every second/minute.), it is not suitable for this application (Because to write into files frequently in the "\System\_disk" will reduce the PAC effectiveness.).



3 🗣 🎓 🗙 🖸	<b></b>		
網址(D) System_DiskW	'in-GRAF		
🗁 sofgrafy	🛋 License.bin	🔊 Quicker.dll	🔊 QuickerNet.dll
🖺 retain_other.txt	🗑 retain_real.txt	Soft-GRAF-WGF.exe	🛋 t5.cod
at5.cod1	Noter ShareNet.dl	🚟 Win_GRAF_WP_8000.exe	🚟 Win_GRAF_WP_8000.Ink

# 6.3 Save Data to EEPROM

The Win-GRAF PAC has a built-in EEPROM memory for users to read and write data, which will not lose data when the PAC shut down. Compared to the read and write of the SRAM, EEPROM has the following disadvantages:

<u>Note</u>: Some PAC have no EEPROM memory (like the WP-5238-CE7). They don't support EEP\_Read() and EEP\_Write().

<u>Advantages</u>: Provides another way to save the important data, besides the Retain Variable. (<u>Section 6.1</u>). <u>Disadvantages</u>: 1. The operation to read/write EEPROM will use much more CPU time (about  $5 \sim 50$  ms),

- but changing to the way of "Retain variable", CPU time is much less than 1 ms. Therefore, do not use the "EEP\_Read" and "EEP\_Write" Functions too frequently, or it will increase the PAC Cycle time.
- EEPROM has a "write" limitation (depending on the PAC), it is not suitable to write the same data many times. So, **DO NOT** call "EEP\_Write" Function within each PAC Cycle to do the "write" operation.

#### ST Program: (Following will show the safe and dangerous coding ways.)

```
(* Declare "FIRST CYCLE" as a "BOOL" variable and has an initial value "TRUE".
    Declare "tmp bool" as a "BOOL" variable °
    Declare "New Val" and "Old Val" as "DINT" variables. *)
  (* Read the EEPROM once in the first Cycle. *)
  if FIRST CYCLE then
   FIRST_CYCLE := FALSE; (* means it is not the first Cycle any more *)
   tmp bool := EEP Read (1, New Val);
  end if;
  (* Safe Coding Way: write to the EEPROM only when the value is changed. *)
  if New_Val <> Old_Val then
   Old Val := New Val;
   tmp_bool := EEP_Write (1, New_Val);
  end if;
(* Dangerous Coding Way: EEPROM may be destroyed very soon. *)
  (* Declare "FIRST CYCLE" as "BOOL" variable and has an initial value "TRUE".
    Declare "tmp bool" as "BOOL" variable.
    Declare "New Val" and "Old Val" as "DINT" variables. *)
  (* Read the EEPROM once in the first Cycle. *)
  if FIRST_CYCLE then
   FIRST_CYCLE := FALSE ; (* means it is not the first Cycle any more *)
   tmp bool := EEP Read (1, New Val);
  end if;
 (*Dangerous Coding Way: Write the "New_Val" value to the EEPROM one time in every Cycle. *)
  tmp_bool := EEP_Write(1, New_Val);
```

## 6.3.1 EEP\_READ (Read a Value from the EEPROM)



<u>Tip:</u>	
Press "F1" key to see the detailed	
setting descriptions.	

Addr: (Data Type: "DINT")

Address, can be 1 to 1200. If the variable type of the "@Name" parameter is a 64-bit data (e.g., LINT or LREAL), the "Addr" can be 1001 to 1200 only.

#### @Name :

A variable name to store the value from the EEPROM. (DO NOT use string variable. Variable type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT or LREAL.)

Q:

Data Type: BOOL. TRUE: Ok; FALSE: Error.

If the type of the "@Name" parameter is REAL or LREAL, will return "Q" as FALSE if the value is NaN (Not a Number) or other error happens. In the case of NaN, the REAL/LREAL variable will get the value "0.0".

## 6.3.2 EEP\_WRITE (Write a Value to the EEPROM)



#### Addr: (Data Type: "DINT")

Address, can be 1 to 1200. If the variable type of the "Value" parameter is a 64-bit data (e.g., LINT or LREAL), the "Addr" can be 1001 to 1200 only.

#### Value :

The value to write to the EEPROM.

(DO NOT use string variable. The value type can be BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, LINT or LREAL.)

Q:

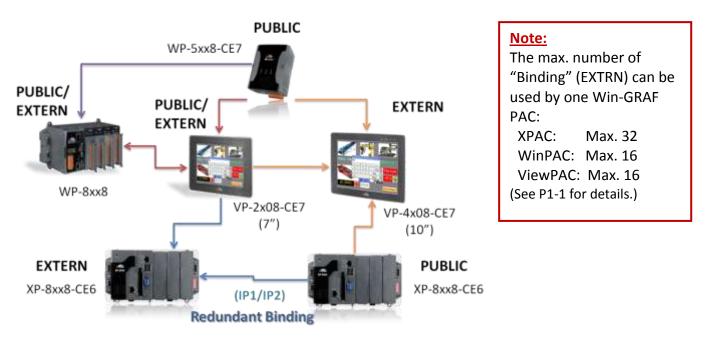
Data Type: BOOL. TRUE: Ok; FALSE: Error.

# Chapter 7 Exchange Data between PACs (Data Binding)

"Binding" function is used to exchange data between ICP DAS Win-GRAF PACs, the data transmission is event triggered. It is much more efficient than polling way. Win-GRAF offers two ways to set up Binding:

- **PUBLIC:** Publish one PAC's own data, or for use of VB .net, C#, or C in the same PAC.
- EXTERN: To get data from other PAC.

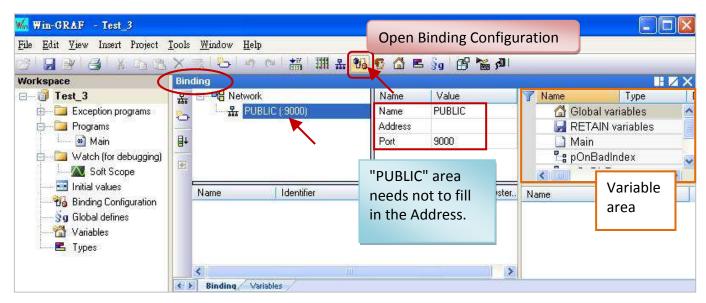
#### Application Diagram:



#### "PUBLIC" Setting Steps:

When the PAC sets up the "PUBLIC" area, means to publish its own data.

- 1. Mouse right-click on the toolbar "Open Binding Configuration" icon to open the "Binding" window.
- 2. Click "PUBLIC (: 9000)" to configure the data to be published as PUBLIC; "Address" field needs not to fill in; "Port" field is fixed to "9000", do not change it.



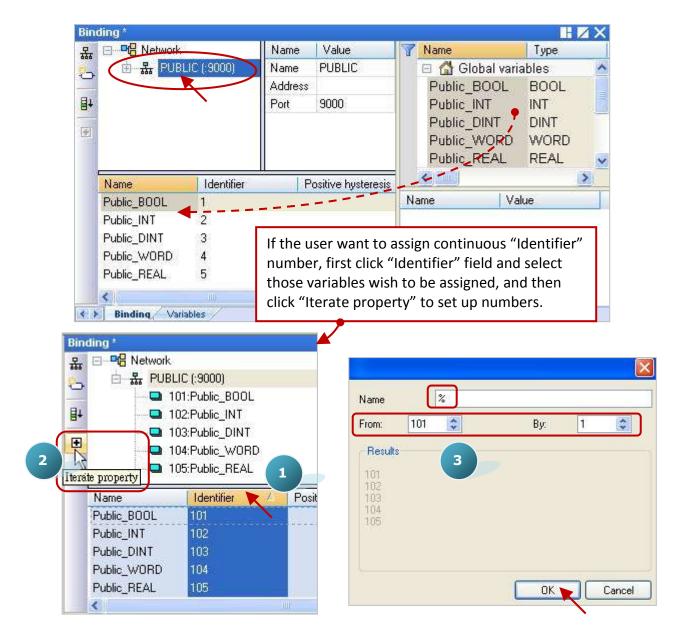
3. Before setting the data to become public, first you must create the variables want to become public in the Variable area. Right-click on "Global variables" and press "Ins" key to add the new variables. In the following table, the variables are used for this example. You can set up your own. After finished, the screen is as below.

Variables Name	Data Type
Public_BOOL	BOOL
Public_INT	INT
Public_DINT	DINT
Public_WORD	WORD
Public_REAL	REAL



4. Right-click on the "PUBLIC (: 9000)", and then select the variables that you want to publish, and drag them to the "Name" area. "Identifier" field will automatically generate numbers (if other PAC wants to access the data, need to set the same ID number).

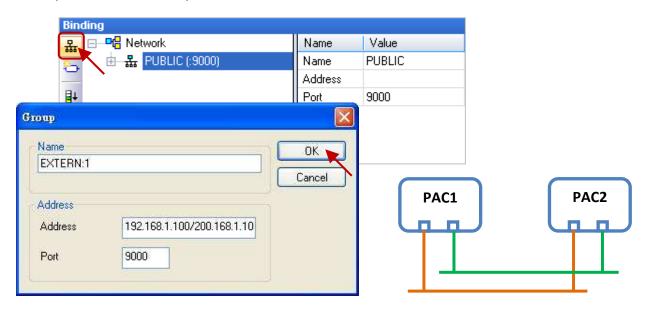
Note: "PUBLIC" can use up to 8192 variables; "Identifier" number can only be "1 ~ 8192".



#### "EXTERN" Setting Steps:

When the PAC sets up the "EXTERN" area, means will get the data from other external PAC.

5. Click the "Insert Master/Port" icon in the left side, it will show the "Group" window. Follow the description below to set up this window, and then click "OK" button.

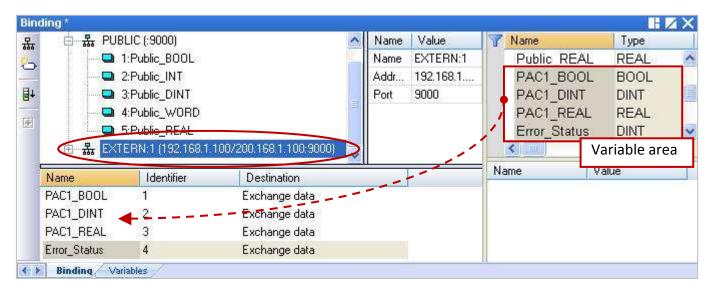


Name: Can be modified to the desired name.

- Address: Enter the IP address of the PAC which data will be obtained (e.g., "192.168.1.100"). User can enter two IP Addresses (e.g., "192.168.1.100/200.168.1.100"; that PAC must use two Ethernet Ports), so that when one IP address occurs any problem, it will try to link the second IP address.
- Port: Fixed to use "9000", do not change it.
- 6. In the variable area, set up the data type you want to get. (Refer Step 3 - Right click on "Global variables" and press "Ins" key to add the variables). The variables that listed in the table are used for this example. You can set up your own. After finished, the screen is as below.

Variables Name	Data Type
PAC1_BOOL	BOOL
PAC1_DINT	DINT
PAC1_REAL	REAL
Error_Status	DINT

 Please drag the variables you need into the "Name" area of the "EXTERN:1".
 <u>Note:</u> "Identifier" field will automatically generate numbers, please change them to the same as the opened IDs of the PAC that you want to get data from.



8. As the picture below, "Error\_Status" variable is used to determine the communication status of the PAC, please set this ID to "0" and then double-click "Destination" field and set it to "Connection error status".

뮮	📋 🐰 PUB	LIC (:9000)			^	Name	Value	7	Name 💎	Туре	
0	🚽 🗖 1:Public_BOOL					Name	EXTERN:1		Public DINT	DINT	1
-		2:Public_INT				Addr	192.168.1		Public BOOL	BOOL	
+		3:Public_DINT				Port	9000		PAC1_REAL	REAL	
		4:Public_WORD							PAC1_DINT	DINT	
		5:Public_REAL							PAC1_BOOL	BOOL	
	EXT	ERN:1 (192.168.1.1	100/200	).168.1.100:9000)	~				Error_Status	DINT	8
		Name Identifier Destination									
	Name	Identifier	C	estination							>
	Name PAC1_BOOL	Identifier 1	150	estination kchange data						alue	>
	1925	Identifier 1 2	E	and the second						lue	
37 52	PAC1_BOOL	1	E: E:	kchange data						lue	
	PAC1_BOOL PAC1_DINT	1 2	E: E:	xchange data xchange data ychange data Exchange data						alue	
	PAC1_BOOL PAC1_DINT PAC1_REAL	1 2 3 0	E: E:	kchange data kchange data vchange data Exchange data Variable error status Variable date stamp	Þ					alue	
C Bui	PAC1_BOOL PAC1_DINT PAC1_REAL Error_Status Binding_/_Var	1 2 3 0	E: E:	kchange data kchange data vchange data Exchange data Variable error status	) )					alue	<u>&gt;</u>

#### Note:

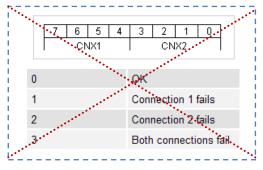
a. If "EXTERN" set up two IP addresses (step 5), then the "Error\_Status" will return two communication status. As the following table, use 8 bits to indicate the connection status. The bit 0 ~ 3 represents the first IP connection status (If all bits are 1, the value is 15); the bit 4 ~ 7 represents the second IP connection status (If all bits are 1, the value is 240). As long as the value is not equal to "0", it means that there are connection errors.

IP2	Connec	tion Sta	itus	IP1 Connection Status				Status Description			
<b>2</b> <sup>7</sup>	<b>2</b> <sup>6</sup>	<b>2</b> ⁵	<b>2</b> <sup>4</sup>	<b>2</b> <sup>3</sup>	<b>2</b> <sup>2</sup>	<b>2</b> <sup>1</sup>	<b>2</b> <sup>0</sup>	Status Description			
	(	)			(	C		Connection OK			
	0				≠0 (1	~ 15)		IP1 Connection error			
	≠ 0 (16 ~ 240)				(	)		IP2 Connection error			
	≠	0			¥	0		IP1 and IP2 Connection error			

b. The return value of the "Error\_Status" is an integer value. The following division operation provides a way to determine this value. Dividing this value by 16, the quotient represents the IP2 connection status, and the remainder represents the IP1 connection status. If the values are not equal to "0", it means that there is any connection error. For example: If "Error\_Status" = 16, divided by 16, the quotient = 1 (≠ 0, IP2 Connection error) and the remainder = 0 (IP1 Connection OK); If "Error\_Status" = 3, divided by 16, the quotient = 0 (IP2 Connection OK) and the remainder = 3 (≠ 0, IP1 Connection error);

#### Notice:

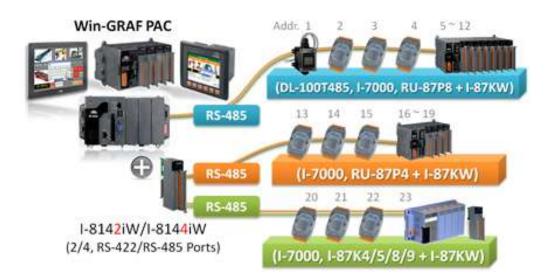
Please ignore the description of the "Connection status" and "Variable status" in the "Networked applications - Dual binding on redundant ETHERNET" of the "HTML Help" that opened when press "F1" key. That explanation is wrong and does not comply with the ICP DAS Win-GRAF PAC. Please ignore it.



# Chapter 8 Connecting DCON I/O Modules

The Win-GRAF PAC can connect the ICP DAS "I-7000" and "I-87KW" remote DCON I/O modules via the COM Port (RS-485). Each PAC can enable up to 16 DCON Ports, and each Port can connect up to 50 remote DCON I/O modules (not recommended over 32). If select the "I-87KW" series I/O modules, it must be used with the RS-485 I/O Expansion Unit (e.g., I-87K4/5/8/9 or RU-87P4/8). You can view the detailed product information on the ICP DAS website:

http://www.icpdas.com/root/product/solutions/remote io/remote io products.html



Before connecting "I-7000" or "I-87KW" remote DCON I/O modules, you must use "DCON Utility" software to configure each module for the Protocol (choose DCON mode), Address (1 ~ 255), Baudrate (the setting must the same with the Win-GRAF PAC, recommended set to 9600), Checksum (the setting must the same with the Win-GRAF PAC, recommended set to "enabled" for communications security), Data format and other Input/Output settings (set according to demand).

#### Note:

- A. When using the AI module of <u>I-7000</u> and <u>I-87KW</u>, set the Data format to "2's Complement".
  - E.g. I-7005, I-7013, I-7014D, I-7015, I-7016, I-7017R, I-7018Z, I-7019R, I-7033; I-87005W, I-87013W, I-87015W, I-87015PW, I-87016W, I-87017W, I-87017RCW, I-87017ZW, I-87017DW, I-87018W, I-87018RW, I-87018ZW, I-87019RW, I-87019ZW, and other Analog Input modules.
- B. When using the AO module of <u>I-7000</u> and <u>I-87KW</u>, set the Data format to "Engineering"
   E.g. I-7021, I-7022, I-7024, I-7024R; I-87024W, I-87024UW, I-87024CW, I-87028UW, I-87028CW,
   I-87028VW, I-87028VW-20V, and other Analog Output modules.

"DCON Utility" is an easy-to-use software toolkit that help user search the network, configure the I/O modules and test the I/O status. Please visit the website to get the software program and user manual: <a href="https://www.icpdas.com/products/dcon/introduction.htm">www.icpdas.com/products/dcon/introduction.htm</a>

The following will introduce the setting method in the Win-GRAF Workbench.

# 8.1 Setting "DCON" I/O Boards

"DCON" can be used to enable an RS-485 Port to connect remote DCON I/O modules (e.g., I-7000 series modules, RU-87P8 I/O Expansion Unit + I-87KW I/O modules, or I-87K8 I/O Expansion Unit + I-87KW I/O modules). If want to enable more than one DCON Port, please set up multiple "DCON" I/O Boards. (One PAC can enable up to 16 "DCON".)

1. Click "Open I/Os" of the Win-GRAF tool bar to open the "I/O Boards" setting window.



2. Double click "Slot8" to add "DCON" I/O Boards (Refer <u>Chapter 4</u>), and then double click "DCON" to open the "Properties" window.

Note: The Slot 0 to Slot 7 are reserved for real I/O modules that plugged into the PAC, and the slot 8 or above are for other usage.

III I/O Boards		
0	Close	
2 3 4 5 6 7 8 DCON 9 10 11 12 13 14	8: DCON - Properties Key = 6 Ref = 16#5 Port = 2 Baud_rate = 9600 Host_watchdog_Enabled = 0 Watchdog_timeout = 5000 Checksum_enabled = 0 Delay_ms_between_polls = 0 Reserved0 = 0 Reserved1 = 0 Reserved3 = 0	
15 16 17	Setting Descriptions           DCON         Enable one serial port (RS-485) to connect remote DCON I/Os (I-7000 series monostread in the series of the ser	odules , RU-87PE
	×	>

#### Parameters:

Note: This "DCON" supports only the communication properties "N,8,1". That is "No-parity", "8 character size" and "1 stop bit". So please must configure all DCON I/O modules to "N,8,1".

Port:	COM port number (1 $\sim$ 37, depends on the PAC.)
Baud_rate:	Communication baudrate in bps, can be 1200, 2400, 4800, 9600, 19200,
	38400, 57600, 115200 (bps). Set a wrong value will use the default value
	9600.
Host_watchdog_Enabled:	1: enable host-watchdog, 0: disable it.
	Set a nonzero value will use the value 1.

Watchdog_timeout:	Unit: ms, can be 3000 ~ 25500.
	Set larger than 25500 will use 25500 ms (25.5 sec).
	Set smaller than 3000 will use 3000 ms (3 sec).
	Ignore this setting when "Host_watchdog_Enabled" is 0.
Checksum_enabled:	0: disabled, 1: enabled.
	Set a nonzero value will use the value 1. (Recommended set to
	"enabled" for communications security.)
Delay_ms_between_polls:	Unit: ms, default is 0 ms. Valid range is 0 ~ 1000.
	Set smaller than 0 will use 0 ms.
	Set larger than 1000 will use 1000 ms.
	If there is no wireless module connected, set a smaller value.
	For instance, set as 0 ~ 10.
	If there are wireless modules (e.g., ICP DAS ZigBee Products: ZigBee
	Converters or ZigBee I/O modules.) connected, set a bigger value.
	For instance, set a value between 30 ~ 100 or other values.
	Set larger value will get slower polling efficiency.

Double click the item to be set, and then fill in the value.

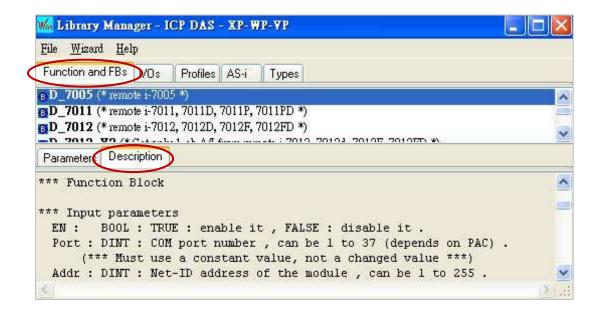
Key=6		
<b>Ref</b> = 16#5		
Port=2		
Baud_rate = 9600		
Host_watchdog_Enabled = 1		
Watchdog_timeout = 3000		
Checksum_enabled = 0 Delay_ms_between_polls = 10		
Reserved0 = 0	Delay_ms_between_polls	×
	10	
Received1 = 0		
Reserved1 = 0 Reserved2 = 0		

3. After setting up the "DCON" in the "I/O Boards" window, it will automatically add a "BOOL" input variable in the "Variables" window. When the Win-GRAF links to the PAC, it will show the COM Port communication status (TRUE: OK; FALSE: error.).

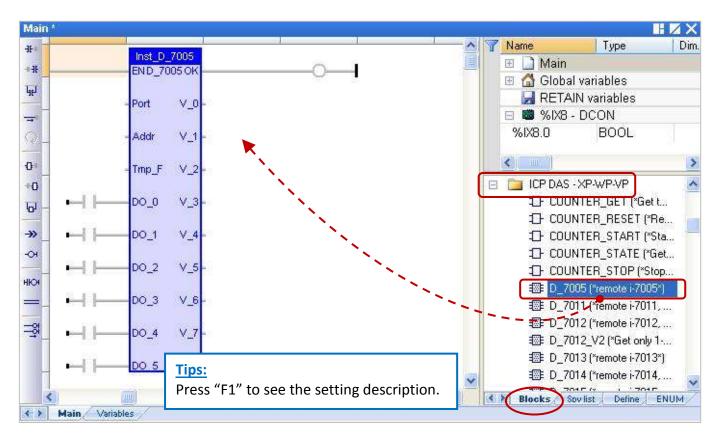
Va	riables								
7	Name	🔥 Type	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
	🗄 🚮 Globa	al variables							
	🔛 RETA	AIN variables							
ſ	🗉 📓 %IX8	- DCON							
L	%IX8.0	BOOL		Input					

# 8.2 Using I/O Function Blocks

The Win-GRAF supports many ICP DAS DCON remote I/O modules, you can open the "Library Manager" (Refer <u>Section 1.2.3</u>) or press "F1" key in the "Function and FBs" to view these I/O Function Block descriptions. This section will introduce "D\_7065", "D\_7018Z", "D\_7083", "D\_87084\_freq", "D\_87084\_cnt4", "D\_87084\_cnt8", and "DL\_100T485" I/O Function Blocks.



In the LD Program - Functional Block area, expand the "ICP DAS - XP-WP-VP" folder of the "Blocks" panel. There are many Functions and Function Blocks. You can select the desired one, and drag it into the editing area of the program.

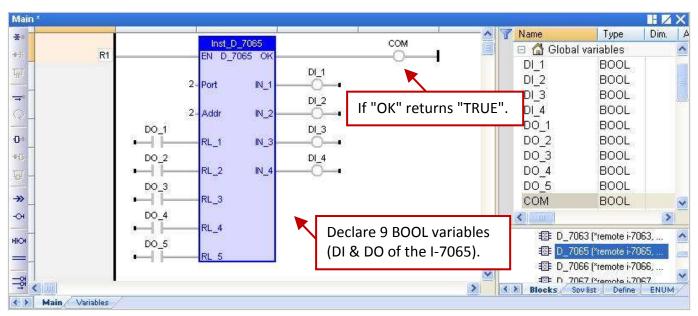


## 8.2.1 "D\_7065" Function Block

"D\_7065": Connect a remote I-7065, I-7065D (Power Relay Output Module) or I-7065A, I-7065AD, I-7065B, I-7065BD (Solid State Relay Output Module).

#### Note:

- 1. All connected DCON I/O modules should be configured once by the DCON Utility (see <u>P8-1</u>).
- 2. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- 3. All values of DI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO\_D\_7065.zip) in the shipping CD and see the program and descriptions.



#### Supposition: Use PAC's COM2 to connect the I-7065 (Addr. = 2) with 4 DI and 5 Relay output channels.

#### **Input Parameters:**

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module , can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)
RL_1 ~ RL_5:	Data type: BOOL. 5-Ch DO values.

#### **Output Parameters:**

OK:	Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.
IN_1 ~ IN_4:	Data type: BOOL. 4-Ch DI values.

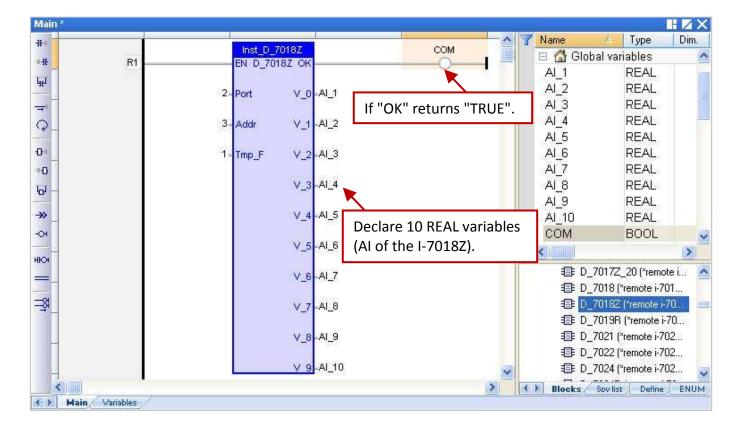
## 8.2.2 "D\_7018Z" Function Block

"D\_7018Z": Connect a remote I-7018Z module that is a 10-channel Thermocouple analog input module for measuring voltage, current or temperature with features of individual channel configuration, open-wire detection and over Voltage protection.

#### Note:

- All connected DCON I/O modules should be configured once (e.g., Address, Baudrate, etc.) by the DCON Utility (see <u>P8-1</u>). Please must configure the data format of AI modules to "**2's complement**" by DCON utility, or the Win-GRAF PAC can not read them well.
- 2. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- 3. All values of AI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO\_D\_7018z.zip) in the shipping CD and see the program and descriptions.

#### Supposition: Use PAC's COM2 to connect the I-7018Z (Addr. = 3) to measure the Celsius temperature.



#### Input Parameters:

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module , can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)

Tmp\_F:Data type: DINT. Temperature Format, can be 1 or 2:1 : temperature unit in Degree Celsius.2 : temperature unit in Degree Fahrenheit.Other value: use it as "1:temperature unit in Degree Celsius".

#### **Output Parameters:**

- **OK:** Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.
- V\_0 ~ V\_9: Data type: REAL. 10-Ch Al value.
  - If the channel range type is configured as mV or Volt by DCON utility, the unit of the returned channel value is Volt.
    - For example, 0.85421 means 0.85421 V or 854.21 mV.
  - If the channel range type is configured as mA by DCON utility, the unit of the returned channel value is mA.
    - For example, 1.5567 means 1.5567 mA.
  - If the channel range type is configured as temperature, the value unit is degree. For example, 25.75 means 25.75 degrees.

#### **Open-wire Detection:**

If the returned temperature is greater than "9000.0", it means that

- 1. The temperature sensor may be broken-line.
- 2. The temperature sensor may be damaged.
- 3. The DCON module is not configured well to fit the connected temperature sensors.
- 4. The ohm measured by the connected sensor is not correct.

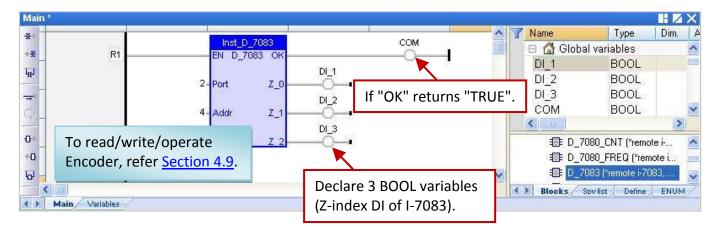
## 8.2.3 "D\_7083" Function Block

"D\_7083": Connect a remote I-7083, I-7083D, I-7083B or I-7083BD module that is a 3-axis, 32 bits encoder counter.

#### Note:

- To get the Encoder value of the I-7083, I-7083D, I-7083B and I-7083BD module, first using "D\_7083" Function Block. Then, using the "Counter\_Start", "Counter\_Stop", "Counter\_Get", "Counter\_State" and "Counter\_Reset" Functions (Refer <u>Section 4.9</u>) to operate encoder channels in an I-7083, I-7083D, I-7083B and I-7083BD module.
- All connected DCON I/O modules should be configured once (e.g., Address, Baudrate, etc.) by the DCON Utility (see <u>P8-1</u>). Please must configure the data format of AI modules to "**2's complement**" by DCON utility, or the Win-GRAF PAC can not read them well.
- 3. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- 4. All values of AI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO\_D\_7083.zip) in the shipping CD and see the program and descriptions.

#### Supposition: Use PAC's COM2 to connect the I-7083 (Addr. = 4) with 3 DI channels.



#### Input Parameters:

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module , can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)

#### **Output Parameters:**

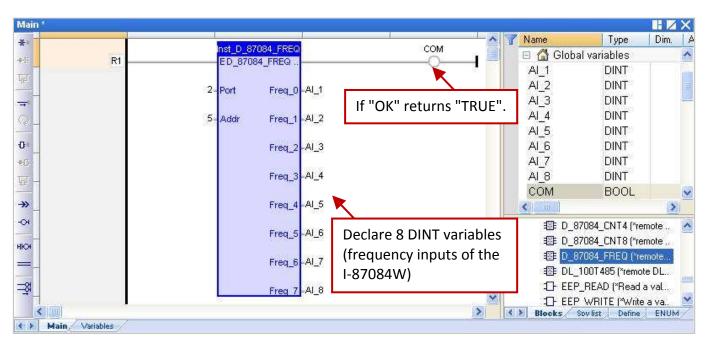
OK:	Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.
Z_0 ~ Z_2:	Data type: BOOL. 3-ch Z-index DI value.

## 8.2.4 "D\_87084\_FREQ" Function Block

"D\_87084\_freq": Connect a remote I-87084W in an I/O Expansion Unit (e.g., I-87K4/5/8/9 or RU-87P4 or RU-87P8.) to measure 8-ch frequency.

#### Note:

- Please MUST configure the I-87084W's frequency data format as "Hex format" by DCON utility (see <u>P8-1</u>) when using the I-87084W to measure frequency. Or it will not work.
- 2. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- 3. All values of AI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO\_D\_87084\_FR.zip) in the shipping CD and see the program and descriptions.



#### Supposition: Use PAC's COM2 to connect the I-87084W (Addr. = 5) to measure 8-ch frequency.

#### **Input Parameters:**

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module , can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)

#### **Output Parameters:**

OK:	Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.
Freq_0 ~ Freq_7:	Data type: DINT. 8-Ch frequency value, unit is Hz.

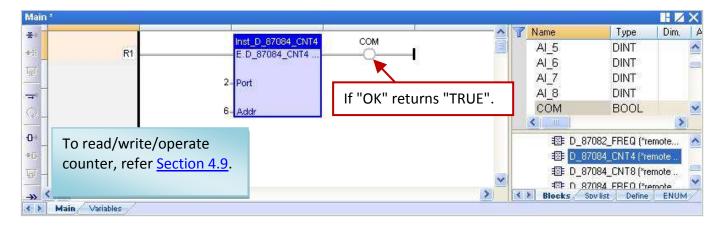
## 8.2.5 "D\_87084\_CNT4" Function Block

"D\_87084\_CNT4": Connect a remote I-87084W in an I/O Expansion Unit (e.g., I-87K4/5/8/9 or RU-87P4 or RU-87P8.) to measure 4-ch counters.

#### Note:

- Please MUST configure the I-87084W's counter data format as "Hex format" by DCON utility (see <u>P8-1</u>) when using the I-87084W to measure counters. Or it will not work.
- 2. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- To get the 4-ch counter value from the remote I-87084W, first using "D\_87084\_cnt4" Function Block. Then, using the "Counter\_Start", "Counter\_Stop", "Counter\_Get", "Counter\_State" and "Counter Reset" Functions (refer <u>Section 4.9</u>) to operate counter channels.
- 4. All values of AI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO\_D\_87084\_C4.zip) in the shipping CD and see the program and descriptions.

#### Supposition: Use PAC's COM2 to connect the I-87084W (Addr. = 6) to measure 4-ch counters.



#### Input Parameters:

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module , can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)

#### **Output Parameters:**

**OK:** Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.

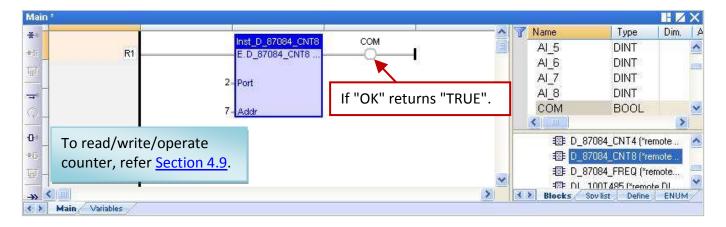
## 8.2.6 "D\_87084\_CNT8" Function Block

"D\_87084\_CNT8": Connect a remote I-87084W in an I/O Expansion Unit (e.g., I-87K4/5/8/9 or RU-87P4 or RU-87P8) to measure 8-ch counters.

#### Note:

- Please MUST configure the I-87084W's counter data format as "Hex format" by DCON utility (see <u>P8-1</u>) when using the I-87084W to measure counters. Or it will not work.
- 2. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- To get the 8-ch counter value from the remote I-87084W, first using "D\_87084\_cnt8" Function Block. Then, using the "Counter\_Start", "Counter\_Stop", "Counter\_Get", "Counter\_State" and "Counter Reset" Functions (refer <u>Section 4.9</u>) to operate counter channels.
- 4. All values of AI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".) .
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\DEMO\_D\_87084\_C8.zip) in the shipping CD and see the program and descriptions.

#### Supposition: Use PAC's COM2 to connect the I-87084W (Addr. = 7) to measure 8-ch counters.



#### **Input Parameters:**

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module , can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)

#### **Output Parameters:**

**OK:** Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.

## 8.2.7 "DL\_100T485" Function Block

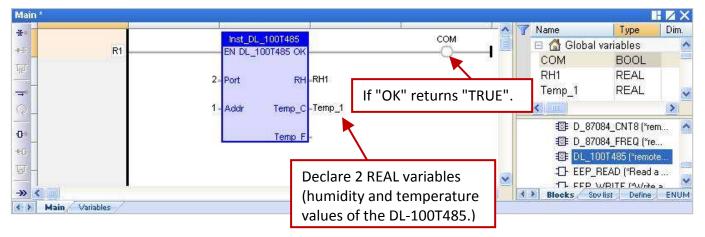
"DL\_100T485": Connect a remote DL-100T485 module to get humidity and temperature value.

Product website: http://www.icpdas.com/root/product/solutions/remote\_io/rs-485/dl\_series/dl-100t485.html

#### Note:

- 1. Please use "DL-100T485 Utility" software in the shipping CD to configure the appropriate parameters of the module (e.g., Module ID). The TDL-100T485's default Address (ID) is "1", Baudrate is "9600", and the Checksum is "Disable".
- 2. Please use "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and set proper settings (Port, baud\_rate, etc.) on it.
- 3. All values of AI channels are meaningful only when the returned communication state is TRUE (If "OK" returns "TRUE".).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\ DEMO\_DL\_100T485.zip) in the shipping CD and see the program and descriptions.

# Supposition: Use PAC's COM2 to connect the DL\_100T485 (Addr. = 1) to get humidity and temperature value.



#### Input Parameters:

EN:	Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module, can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)

#### **Output Parameters:**

OK:	Data type: BOOL. TRUE: Communication is Ok. FALSE: Communication failed.
RH:	Data type: REAL. The value is "Relative humidity"; unit is 1%.
	For example, a value "45.7" means 45.7%.
Temp_C:	Data type: REAL. The temperature value is in "Degree Celsius".
	For example, a value "25.7" means 25.7 Degree C.
Temp_F :	Data type: REAL. The temperature value is in "Degree Fahrenheit".
	For example, a value "78.26" means 78.26 Degree F.

## 8.2.8 "D\_GPS721" Function Block

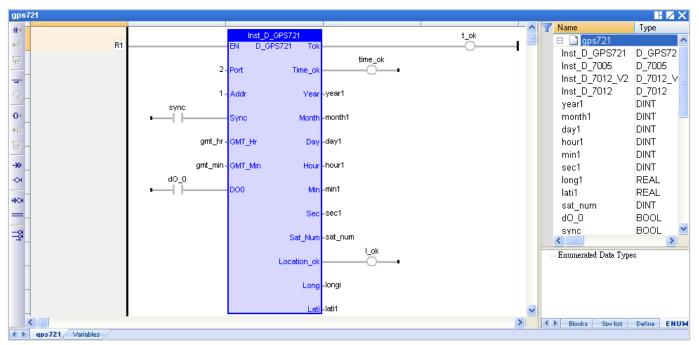
Using the "D\_GPS721" function to link one "GPS-721" remote GPS receiver module (includes one DO and one PPS outputs) which is used to receive GPS signals, get the precise date, time, longitude and latitude, and for satellite positioning and time correction. Moreover, "GPS-721" is equipped with a RS-232 interface, the remote host can request the GPS-721's GPS information and remotely control the built-in DO channel by using DCON commands over RS-485. And, the PPS (Pulse Per Second) function can be used for a simple time synchronization.

#### Product website:

http://www.icpdas.com/root/product/solutions/industrial wireless communication/wireless solutions/ /gps-721 tc.html

#### Note:

- 1. One PAC can use only one GPS-721 module.
- All connected modules should be configured once. First of all, set up parameters of the GPS-721 module (e.g., Address, Baudrate, etc.) by the DCON Utility (see <u>P8-1</u>). (By defaults, the Address (ID) is "1", Baudrate is "9600", and Checksum is "Disable".)
- 3. Please add "DCON" (<u>Section 8.1</u>) in the "I/O boards" window and fill in the proper settings (Port, baud\_rate, etc.).
- Referring <u>Chapter 12</u>, click the menu bar "File" > "Add Existing Project" > "From Zip" to restore the demo project (CD-ROM: \Napdos\Win-GRAF\demo-project\dmeo\_gps721.zip) in the shipping CD and see the program and descriptions.)



#### Input Parameters:

EN:	 Data type: BOOL. TRUE: enable it; FALSE: disable it.
Port:	Data type: DINT. COM port number (can be 1 to 37, depends on PAC).
	(*** Must use a constant value, cannot be a changed value. ***)
Addr:	Data type: DINT. The Net-ID address of the module, can be 1 to 255.
	(*** Must use a constant value, not a changed value ***)
Sync:	Data type: BOOL. Set it as "TRUE" to enable auto time synchronization ( <u>note</u> : It will work only when the "Time_ok" is also set to "TRUE"). If the time gap between the GPS-721 and the PAC is 5 seconds (or more), it will automatically correct the PAC time. Set it as "FALSE" to disable this function.

#### GMT\_Hr & GMT\_Min:

Data type: DINT. The time difference between local time and GMT (Greenwich Mean Time). For example, Beijing and Taipei are +8 hours (GMT\_Hr=8, GMT\_Min=0), United States is -6 hours (GMT\_Hr= -6, GMT\_Min=0), and India is +5.5 hours (GMT\_Hr=5, GMT\_Min=30).

IN.

Inst\_D\_GPS721

D\_GPS721

**DOO:** Data type: BOOL. The Digital output channel of the GPS-721 module.

#### **Output Parameters:**

- Tok: Data type: BOOL.
  - TRUE: The communication of GPS-721 is normal.
  - FALSE: The communication of GPS-721 is failed. And all the following output values are invalid.

Time\_ok: Data type: BOOL.

- TRUE: Now, the values of Year, Month, Day, Hour, Min and Sec are valid.
- FALSE: Now, the values of Year, Month, Day, Hour, Min and Sec are invalid. (i.e., an error or non-real-time value.)
- 2-Port O Time\_ol 1 -Addr Year -γear1 = 2015 Month -month1 = 6 Sync gmt\_hr = 8 GMT\_Hr Day day1 = 15 gmt\_min = 0- GMT\_Min Hour -hour1 = 16 dO\_0 = FALSE 4.14 DOD Min -min1 = 58 -sec1 = 45 Sec Sat\_Num\_sat\_num = 8 I\_ok = TRUE Ó Location\_ol Long -longi = 25.0116 Lati -lati1 = 121.4758

Tol

time\_ok = TRUE

t\_ok = TRUE

Note: Each day from 23:59:00 to 00:00:59 (2 minutes), the "Time\_ok" will be set as "FALSE" automatically and the time synchronization will not be processed.

Year:	Data type: DINT. Year ( 2009 ~).
Month:	Data type: DINT. Month ( $1 \sim 12$ ).
Day:	Data type: DINT. Day (1 ~ 31).
Hour:	Data type: DINT. Hour ( $0 \approx 23$ ).
Min:	Data type: DINT. Minute ( 0 ~ 59).
Sec:	Data type: DINT. Second ( 0 ~ 59).
Sat_Num:	Data type: DINT. The number of used satellites.
	(0: no satellites can be found. Or, using 1 ~ 9 satellites)
Location_ok	Data type: BOOL.
	FALSE: Now, the values of "Long" and "Lati" are invalid (i.e., an error or non-real-time value.)
	TRUE: The GPS-721 has got the current longitude and latitude location.
	(Only when "Location_ok" is set to "TRUE", the values of "Long" and "Lati" are valid. )
Long:	Data type: REAL. Longitude (positive value: East ; negative value: West).
	(For example, "25.0121" means 25.0121 degrees.)
Lati:	Data type: REAL. Latitude (positive value: North ; negative value: South).
	(For example, "121.4576" means 121.4576 degrees.)

# Chapter 9 On Line Change

"On Line Change" function allows Win-GRAF PAC change its application to a modified one during the PAC running time. The modified application must be the same name as the original one that currently running on the PAC. The "On Line Change" is primarily used for emergency, such as, when the application is not allowed to be disabled or stopped for a while or cannot find time to replace a new application (for example, the device needs 24 hours operation and cannot be stopped). Except the above situations, it is not recommended to use this function! You had better stop the running application, then download the modified application to the PAC (see Section 2.3.5), which is more safe.

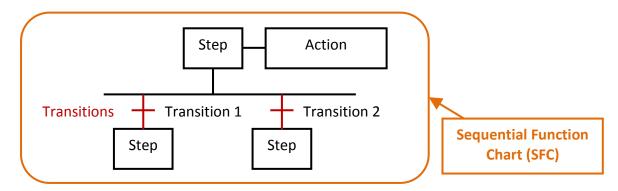


## 9.1 Limitations of "On Line Change"

Please notice the important limitations before enabling the "On Line Change"!

#### When On Line change is enabled, you can perform on the fly the following kinds of changes:

- Change the code of a program.
- Change the condition of a SFC transition or the actions of a SFC step.



- Create, rename or delete global and local variables.
- Create, rename or delete global and local function block instances.

Workspace	Main							HZX
⊟ 🗿 Test	-IE			-~ 7	Name 🛛 🕹	Туре	Dim.	Attrib.
🖶 🔁 Exception programs 🎦 Programs 🎦 Main		Inst_BLINK	OUT1		🗉 🗋 Main			
		RUN BLINK Q	-0		Inst BLINK1	blink		
	μ				🗉 🚮 Global va	ariables		
🗄 🛄 Watch (for debuggin		T#2s-CYCLE			_del_SW1	BOOL		Deleted
Soft Scope	0				OUT1	BOOL		
📰 Initial values	_				SW2	BOOL		Added

#### The following kinds of changes are not allowed:

• Create, delete or rename a program. (It will appear a warning message if delete a program.)

Main	Workspace
A POU cannot be deleted with On Line Change active. Do-you want to disable it? 是仅)  否则	Test     Exception programs     Programs     Main

- Change SFC charts.
- Change the local parameters and variables of a UDFB.
- Change the type or dimension (or string length) of a variable or function block instance.
- Change the set of I/O boards.
- Change the definition of RETAIN variables.

#### In addition, the following programming features that are not safe during a change should not be used:

• Pulse (P or N) contacts and coils (edge detection).

K Instead, you must use declared instances of R\_TRIG and F\_TRIG function blocks.

Rising Pulse Detection						
P (False > True)	Before Enable	After Enable				
	SW1	SW1 Inst <u>R_TRIG</u>				
	OUT1	Inst_R_TRIG OUT1 CLK_R_TRIGQO				
Decreased Pulse Detection						
N (True > False)	SW1	SW1 Inst_F_TRIG CLI(F_TRIG) Q				
	OUT1	Inst_F_TRIG OUT1				

• Loops in FBD with no declared variable linked.

X You need to explicitly insert a variable in the loop.

# 9.2 Using "On Line Change"

## Enable The "On Line Change" Function:

1. Mouse click "Project > Settings..." from the menu bar, then double click "On Line Change" item to set it "Enabled".

🖟 Win-GRAF - Te	est		
<u>File E</u> dit <u>V</u> iew Ins	ert Proje	ct <u>T</u> ools <u>W</u> indow <u>H</u> elp	
3 🛛 🖓 🖂	X 🗗	Build All Projects F7	
Vorkspace		Clean All Projects	
🖃 🗃 Test	<b>*</b>	Download All Projects	
🗄 🖾 🛄 Exception	prc	Settings	
📩 🛁 Programs		Detuitgs	
Project	settings		16
		ects\Test	
Genera	V/021000280027	Name	Value
Runtime		Communication parameters	192.168.71.18:502
Compile		Survey Cycle time	0
Advanc		Code Generation	Release
(All)		Complex variables in a separate segment	No
		😴 On Line Change	Enabled
		4 Version	V1 - 2014/06/05 15:48
		Libraries	Edit
		Use external objects	Edit
		Double click to disable or enable and configure O capabilities.	In Line Change

2. Click "Project > Build All Projects" from the menu bar. Must compile the program first, then can set up the following steps.

🐝 Win-GRAF - Test							
<u>File Edit V</u> iew Insert Project	<u>T</u> ools <u>W</u> indow <u>H</u> elp						
	LX R G M M M	( III 🏭 😘 🕵 🗂 🖻	🛐 🔐 🕍 🔊 🛛				
Wg							
E Project <u>T</u> ools <u>W</u> indow	w <u>H</u> elp		- L	- <u> </u>	Name	Туре	Dim
Build All Projects	F7	Inst_BLINK	OUT1		🗉 🚮 Global	variables	^
		RUN BLINK	°		SW1	BOOL	
<u>C</u> lean All Projects					OUT1	BOOL	×
👬 Download All Proje	ects	T#2s-CYCLE			<		>
Settings				Œ	(All)		~
	/			Œ	🛅 (Used)		
g Global defines	+[]]			· ·	(Project)		1 (333)
Variables					Blocks St		ENUM /
	★ Variables Main			1000	BIOCKS A ST	A libr N - Deline V	ENOM
WITH REPORT OF THE	Build						×
	Building application data						~
	< 9 BOOL/SINT; 0 INT; 1 DINT	'/REAL; 0 LINT/LREAL; 3 TIM	; 1 STRING; - CRC = 1	102ac76 >			
	On Line Change not possible: to	o many variables					
	Relocating code	· ]	This message	e can be	ignored v	vhen just	
	< Code CRC=a9cc42e6 - File CF	RC=fea01925 - Size=1832 >	enable On Li	ne Chan	σe	-	
	No error detected			ic chung	BC.		
	Build Cross references	Runtime Call stack B	eakpoints 📃 Digital sam	pling trace	Prompt HMI		
Ready		OffLine 192.168.7	.18:502 🧷 🧷	0,90	692 × 18	0,0	

## Setup the Number of Variables:

When the "On Line Change" is enabled, you have to set up the number of variables reserved for the declaration of the new variables and Function Blocks for future on-line change usage.

3. The same as the step 1, mouse click "Project > Settings..." from the menu bar, then double click on the "On Line Change" to show the setting window.

Please set the needed new number in the "Value" or "Margin" fields.

- Note: If both "Value" and "Margin" have set values, it will use the larger value. In this example, "Value" is set as "30" and "Margin" is set as "10", then the displayed value x 10% is smaller than 30, so it will use the larger value "30".
- 4. Click the needed Variable Type, then click the "Set" button.

(E.g., Click "BOOL/SINT variables" and "Set" button, then the number become 9 + 30 = 39.)

- <u>Note:</u> "STRING buffers (characters)", "FB instance data (bytes approx.)" and "Complex variables segment (bytes) " need to set a larger number (This example uses "5000").
- 5. After setting (as the picture below), click the "X" in the upper right corner to exit the setting.

Line Change Current Status		On Line Change Current Status		
ENABLED	<u>E</u> nable Disable	ENABLED		<u>E</u> nable Disable
Number of variables allocated		Number of variables allocat	ted	
BOOL/SINT variables       INT variables         INT variables       DINT/REAL variables         DINT/REAL variables       INT/REAL variables         TIME variables       STRING variables         Active timers       STRING variables         STRING variables       STRING buffers (characters)         FB instances       FB instance data (bytes - approx.)         Published variables       Complex variales segment (bytes)         POUs       POUs	9739 1/0 2/0 1/0 2/0 0/0 2/0 17/0 3/0 32/0 0/0 0/0 0/0 5/0	BOOL/SINT variables INT variables DINT/REAL variables LINT/LREAL variables TIME variables Active timers STRING variables STRING buffers (characte FB instances FB instances FB instance data (bytes - x Published variables Complex variales segment POUs	approx.)	9 / 39 1 / 31 2 / 32 1 / 31 2 / 32 0 / 30 2 / 32 17 / 5017 3 / 33 32 / 5032 0 / 30 0 / 5000 5 / 35
Numbers displayed: Used / Allocated Displayed "used" numbers are according to the last Allocate: O Value 30 Margin (%) 10	build Set	Allocate: O Val	are according to the last build	Set

6. Click "Project > Build All Projects" from the menu bar, compile the program again. Then click "On Line" or click the tool icon 1 to link to the PAC. (Refer the Section 2.3.5.)

ً	Build All Projects	F7	***	16 🔨 🔂 🖻 🖇	, B 2 3	
<b>*</b> F	Clean All Projects Download All Projects Settings			Inst_BLINK RUN BLINK Q	оит1 О	-
531	Qn Line	Ctrl+F5				
1	Simulate	F5	#2s	CYCLE		

- 7. After successfully link to the PAC, click the tool icon "Download changes" to download the program to the PAC.
  - **Note:** "On Line Change" is only suitable for the program that just has a little change (Do not need to stop the application). If the program name is different from the running one on the PAC, the user has to stop and download the program again (refer the <u>Appendix B</u>).

<b>•••</b> • (Ma	iin]				uees 🕅 📰	wnload changes -	TESTIX
-IE-				- ~ [	🝸 Name 🖉	Value	Туре
HE		Inst_BLINK	OUT1 = TRUE		🗄 📄 Main		^
	R1	RUN BLINK Q	<u> </u>		🗉 🚮 Globa	ıl variables	
121					SW1	FALSE	BC
		T#2s-CYCLE			OUT1	TRUE	BC
₩ → へ →						distribution.	>
					🗉 🛅 (All)		^
					🖽 🛅 (Used)		
+13				~	🗉 🧰 (Project)		~
				>	Blocks	iov list Define	ENUM

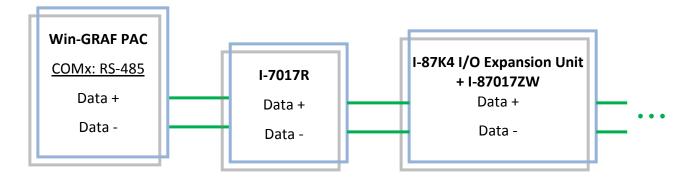
- 8. Click on the tool icon "Do On Line change" to execute the program.
  - **Note:** After executing the "On Line Change", there are some using restrictions for protecting the system normal operation (refer the <u>Section 9.1</u>). So, make sure the program is correct, then perform this function.

File Edit View Insert Project		👬 🏢 🏭 % 😨 🖨 🛎 💡	🕑 🅍 🚮 RUN			
Workspace	• • • • [Main]				)o On Line chan	X N
E- V Test [RUN]	31-9		· · · · · · · · · · · · · · · · · · ·	🕎 Name 🗄	Value	Туре
🞰 🚞 Exception programs	+12	Inst_BLINK	OUT1 = TRUE	🗉 🗋 Main		^
🖨 🔤 Programs	R1 –	RUN BLINK Q	-0	🗉 🚮 Global	variables	
Main	42			SW1	FALSE	BC
🛓 🖂 Watch (for debuggin		T#2s-CYCLE		OUT1	TRUE	BC
Soft Scope	ON INTERNET					>
🔜 Initial values				🗉 🧰 (All)		~
🚽 🚮 Binding Configuration	- B*			⊡ (Used)		
🚽 🕺 🚽 Global defines	+0.		~	(Project)		~
- 🛗 Variables			>	South Star	list Define	ENUM
I 🛃 Types	Variables Main					
	Build					×
	Build in progress Please \	Wait				
	No error detected					

# Chapter 10 Data/Type Conversion and Using the PAC Time

## **10.1 AI Data Conversion**

If you are using AI modules in the PAC's Slot 0 to 7 (for example: I-8017HW), and want to convert the AI input signal (for example: "4 to 20 mA" or "0 to 10 V") to the user engineering value (for example: 0 to 10000), refer the <u>Section 4.3</u>. However, if you are using the remote AI modules (e.g., through PAC's RS-485 Port to connect to the I-87017ZW or I-7017R module), you can refer the following settings:

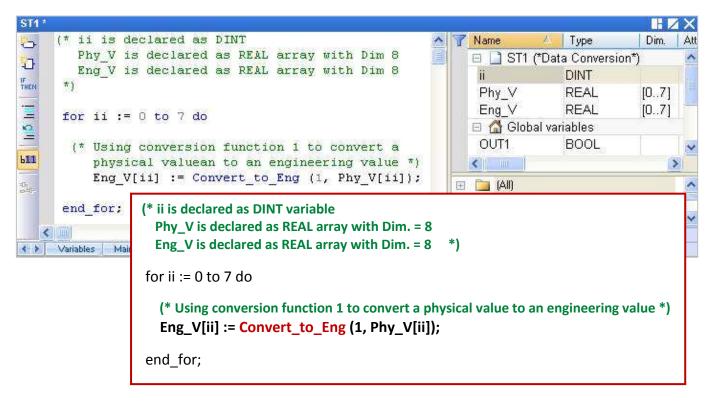


- First, connect "i\_scale" in the "I/O Boards" window, and double click on "i\_scale\_x" to open the Properties window (refer the <u>Section 4.2</u> for detail steps)
   <u>Note:</u> "I/O Boards" supports ONLY ONE "i\_scale" (DO NOT connect 2 or more "i\_scale").
- Set up the number and value of the Conversion Function that need to be enabled. (E.g., Use Function 1 to convert "4 to 20 mA" into "0 to 10000").

Ch <b>01_</b> X0_Min_Physical_Val:	"4.0"
Ch <b>01</b> _X1_Max_Physical_Val:	"20.0"
Ch <b>01</b> _Y0_Engineering_Val_For_X0:	"0.0"
Ch <b>01</b> _Y1_Engineering_Val_For_X1:	"10000.0"

I/O Boards	9.0: i_scale - Properties	2
Hallow and the second second	<b>Key</b> =6	~
0	Ref = 16#2A Ch00 X0 reserved = 0.0	
1	Choo X1 reserved = 0.0	
2	Ch00 Y0 reserved = 0.0	-
3	$\frac{\mathbf{Ch00}_{\mathbf{Y}1}_{\mathbf{reserved}} = 0.0}{\mathbf{Ch01}_{\mathbf{Y}0}_{\mathbf{M}}} = \frac{\mathbf{N}_{\mathbf{H}1}}{\mathbf{N}_{\mathbf{H}1}} = \frac{1}{40}$	
4	Ch01_X0_Min_Physical_Val = 4.0 Ch01_X1_Max_Physical_Val = 20.0	
5	<b>Ch01 Y0 Engineering <math>\overline{\mathbf{Y}}</math>al For <math>\mathbf{X}<b>0</b> = 0.0</math></b>	
6	Ch01 Y1 Engineering Yal For X1=00	100
7	Ch02_X0_Min_Physical_Yal=0.0 Ch02_X1_Max_Physical_Yal=0.0	<b>L</b>
8	<b>Ch02 YO Engineering Val For X0</b> = $0.0$ 10000.0	
9 i scale	Ch02_Y1_Engineering_Ya1_For_X1 = 0.0 Ch03_X0_Mm_Physical_Ya1 = 0.0	
	Ch03_X1_Max_Physical_Yal = 0.0	×
- 1 i scale 1	li scale	~
10	Setup the scaling function No. 01 to 29 for scaling I/O variables .	
11		
12	Notice:	
13	<ol> <li>If setting both value of Ch. X0 and Ch. X1 to 0.0, it means the relative scaling function No. is a</li> </ol>	d
14	2. If Ch_X0 is greater than or equal to Ch_X1, the setting is wrong.	V
14		a

 Edit an ST Program to convert a physical value (e.g., Phy\_V[0] to [7]) to an engineering value (e.g., Eng\_V[0] to [7]). (Refer the <u>Section 2.3.3</u> for detail setting steps.)



## 10.2 AO Data Conversion

If you are using AO modules in the PAC's Slot 0 to 7 (for example: I-8024W), and want to convert the engineering value to the AO output signal (e.g., convert "0 to 20000" to "0 to 10 V"), refer the <u>Section</u> <u>4.4</u>. However, if you are using the DCON remote AO modules (e.g., through PAC's RS-485 Port to connect to the I-7024 module), you can refer to the following settings:

1. Connect "i\_scale" in the "I/O Boards" window, and double click "i\_scale\_x" to open the Properties window (refer the <u>Section 10.1</u> for detail steps)

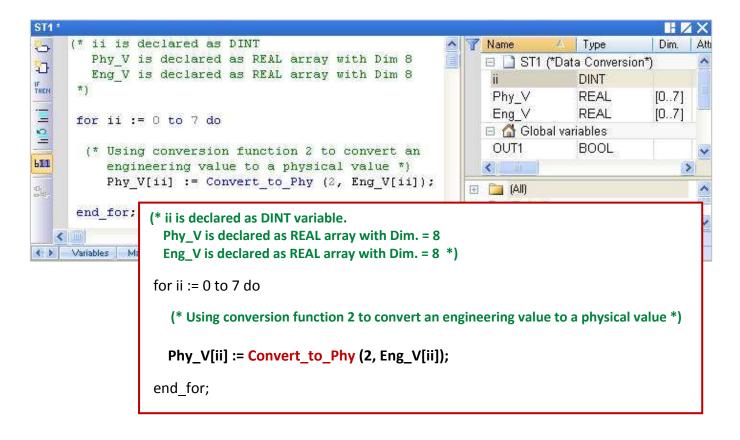
Note: "I/O Boards" supports ONLY ONE "i\_scale" (DO NOT connect 2 or more "i\_scale").

	Close
1	
2	Select
3	
4	Delete
5	
<u>6</u>	Rename
7 8	Properties
9 i_scale	
- 0 i_scale_0	Virtual/Re
- 1 i_scale_1 - 2 i_scale_2	Move Up
10	
11	Move Dog
12	
13	Help
14	

2. Set up the number and value of the Conversion Function (e.g., use Function 2 to convert "0 to 20000" to "0 to 10 V").

.0: i_scale - Properties	×
<b>Key</b> = 6 <b>Ref</b> = 16#2A	Ch <b>02</b> X0 Min Physical Val is set as "0.0".
Ch00_X0_reserved = 0.0 Ch00_X1_reserved = 0.0	Ch <b>02</b> X1 Max Physical Valies set as "10.0".
Ch00_Y0_reserved = 0.0	Ch <b>02</b> YO Engineering Val For XO is set as "0.0".
Ch00_Y1_reserved = 0.0 Ch01_X0_Min_Physical_Val	
Ch01 X1 Max Physical Val	
Ch01_Y0_Engineering_¥al_l Ch01_Y1_Engineering_¥al_l	
Ch02_X0_Min_Physical_Val Ch02_X1_Max_Physical_Val	. = 10.0
ChO2_YO_Engineering_\al_l ChO2_Y1_Engineering_Val_l	For X0 = 0.0 For X1 = 20000 0
Ch03_X0_Min_Physical_Val Ch03_X1_Max_Physical_Val	=0.0

3. Edit an ST Program to convert an engineering value to a physical value (e.g., Eng\_V[0] to [7]). (Refer the <u>Section 2.3.3</u> for detail setting steps.)



# 10.3 Data Type Conversion

When different types of data want to do "+, -, \*, /" calculation or ">, <, =, <=, > =, <> (not equal)" operation, or when the variable types of the parameters in the function are different, you must first use the type conversion function (in the following table) to convert them into the same data type, then can use the data normally.

Data Conversion Functions	Descriptions	Data Conversion Functions	Descriptions
ANY_TO_BOOL	Convert to Boolean	ANY_TO_REAL	Convert to Real
ANY_TO_SINT	Convert to Short Integer (8-bit)	ANY_TO_LREAL	Convert to Double
ANY_TO_INT	Convert to Integer (16-bit)	ANY_TO_STRING	Convert to String
ANY_TO_DINT	Convert to Long Integer (32-bit – Default)	NUM_TO_STRING	Convert Number to String. Can set the decimal digital number after converting
ANY_TO_LINT	Convert to Large Integer (64-bit)	АТОН	Convert Hexadecimal String to Integer
ANY_TO_TIME	Convert to Timer	НТОА	Convert Integer to Hexadecimal String

For example, the following ST program will convert the DINT variable to a Real first, and then do the calculation.

You can open the "HTML Help" from the menu bar, and enter the searching key word you want to see the detail setup instructions.



· · · · · · · · · · · · · · · · · · ·	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	品
内容(C) 搜尋(2) 我	約最愛(1)	
請輸入欲搜尋的字串(₩):	200 - 124	
type conversion		<b>V</b>
	出主題①	顯示(D)
璧取主題(I):	已找到:14	4
標題	位置	順序
Type conversion functions	IEC 61131-3	1
ANY_TO_BOOL	IEC 61131-3	2
ANY_TO_DINT / ANY_TO_U		3
		- Cont
ANY_TO_INT / ANY_TO_UINT	IEC 61131-3	4
	IEC 61131-3 IEC 61131-3	5
ANY_TO_LINT		5 6
ANY_TO_LINT ANY_TO_LREAL ANY_TO_REAL	IEC 61131-3	5 6 7
ANY_TO_LINT ANY_TO_LREAL ANY_TO_REAL ANY_TO_TIME	IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3	5 6 7 8
ANY_TO_LINT ANY_TO_LREAL ANY_TO_REAL ANY_TO_TIME ANY_TO_SINT	IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3	5 6 7 8 9
ANY_TO_LINT ANY_TO_LREAL ANY_TO_REAL ANY_TO_TIME ANY_TO_SINT Programming languages - Refe	IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 . IEC 61131-3	5 6 7 8
ANY_TO_LINT ANY_TO_LREAL ANY_TO_REAL ANY_TO_TIME ANY_TO_SINT Programming languages - Refe	IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3	5 6 7 8 9
ANY_TO_STRING NUM_TO_STRING	IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 . IEC 61131-3	5 6 7 8 9 10 11 12
ANY_TO_LINT ANY_TO_LREAL ANY_TO_REAL ANY_TO_TIME ANY_TO_SINT Programming languages - Refe ANY_TO_STRING	IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3 IEC 61131-3	5 6 7 8 9 10 11

# **10.4 BCD Conversion**

BCD 4-bit code is used to represent decimal numbers from 0 to 9. Suppose there is a decimal value "132", if converted to BCD code is "000100110010" and if converted to binary is 10000100 (e.g.,  $2^7 + 2^2 = 128 + 4 = 132$ ).

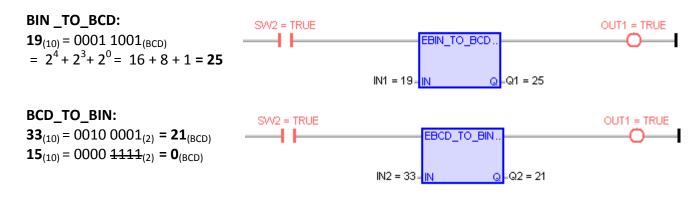
Decimal		BC	D		Description
Decimal	<b>2</b> <sup>3</sup>	<b>2</b> <sup>2</sup>	<b>2</b> <sup>1</sup>	<b>2</b> <sup>0</sup>	Description
0	0	0	0	0	0
1	0	0	0	1	$2^0 = 1$
2	0	0	1	0	2 <sup>1</sup> = 2
3	0	0	1	1	$2^1 + 2^0 = 3$
4	0	1	0	0	$2^2 = 4$
5	0	1	0	1	$2^2 + 2^0 = 5$
6	0	1	1	0	$2^2 + 2^1 = 6$
7	0	1	1	1	$2^2 + 2^1 + 2^0 = 7$
8	1	0	0	0	$2^3 = 8$
9	1	0	0	1	$2^3 + 2^0 = 9$

#### NOTE:

BCD code can only represent numbers from 0 to 9. It can not be used for these six values (1010, 1011, 1100, 1101, 1110, 1111), and will return "0".

The function table below can be used for BCD (Binary Coded Decimal) value conversion.

Type Conversion Function	Description
BIN_TO_BCD	Convert Binary to BCD value
BCD_TO_BIN	Convert BCD value to Binary



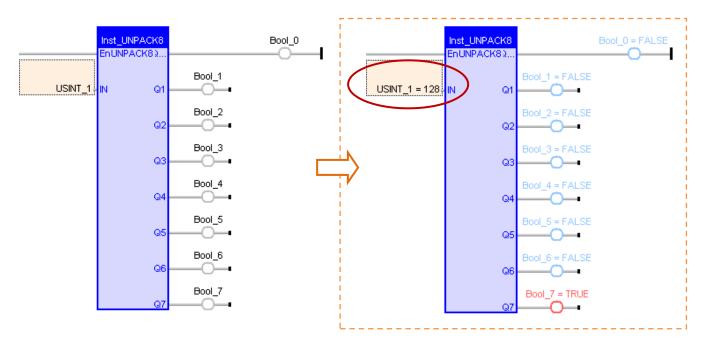
You can open the "HTML Help" from the menu bar, and enter the searching key word you want to see the detail setup instructions.



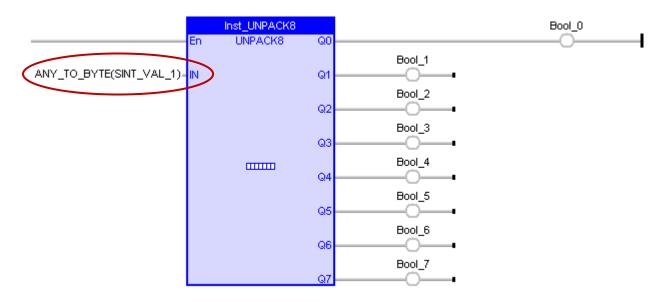
🛃 HTML Help		
□ ■ ○ □ ■ □ □ ■ ○ □ ■ □ □ ■ □ □ ■ □ □ ■ □ □ ■ □ □ ■ □ □ ■ □ □ ■ □ □ ■ □	□☆ 下一頁 重新整理	品
內容(C) 搜尋(3)	我的最愛①	
請輸入欲搜尋的字串(例):		
BCD		
F	刊出主題(L)	顯示(D)
選取主題(I):	已找到:3	
標題	位置	順序
BCD_TO_BIN	IEC 61131-3	1
Type conversion functions	IEC 61131-3	2
BIN TO BCD	IEC 61131-3	3

#### **Unpack Integer to Boolean:**

If want to unpack one BYTE (or USINT, range: 0 to 255) to 8 Booleans, you can use "UNPACK8" Function Block.



If want to unpack one SINT to 8 Booleans, you must first use the ST program "ANY\_TO\_BYTE ()" to convert the SINT to be a BYTE type, as follows:



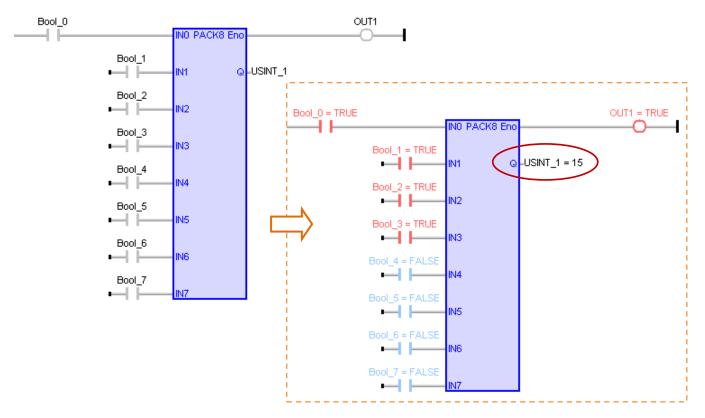
## Pack Boolean Into Integer:

If want to pack 8 Booleans into one BYTE (or USINT, range: 0 to 255), you can use "PACK8" Function.

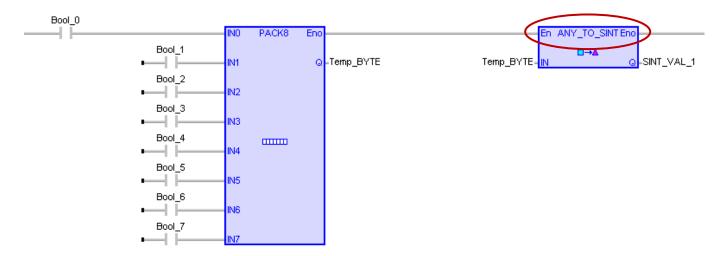
## ST Program:

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

#### LD Program:



If want to pack 8 Booleans into one SINT, you must assign a "BYTE" variable to the output(Q) to save the value temporary, and use a "ANY\_TO\_SINT" Function to convert BYTE into SINT type, as follows:



**Note:** If the compiling fails, please click "Project" > "Settings" from the menu bar to check if the setting of "Complex variables in a separate segment" in the "Runtime" is "Yes".

Project settings		×
C:\Win-GRAF\Projects\1	fest	
General	Name	Value
Compiler	💷 Runtime system	Little endian
	🐒 Cycle time	0
Advanced	🔐 Code Generation	Release
(All)	🕄 Exchange IOs while stepping	Yes
	Support variables locking	No
	[i]. Complex variables in a separate segment	Yes

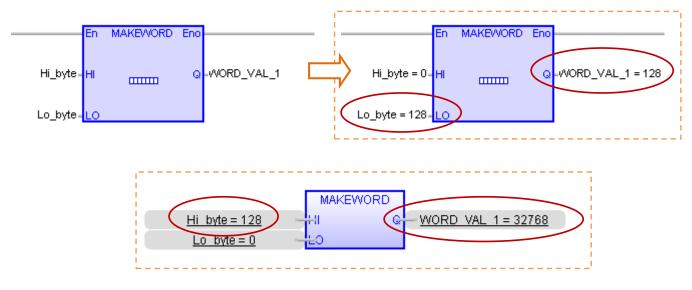
#### Pack Two 8-bit Data into One 16-bit Data

If want to pack 2 BYTE (or USINT) into one WORD (or UINT), you can use a "MAKEWORD" Function.

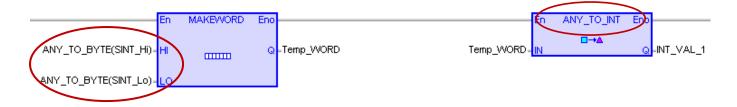
#### ST Program:

WORD\_VAL\_1 := MAKEWORD (Hi\_byte, Lo\_byte);

LD/FBD Program:



If want to pack 2 SINT into one INT, you must first use an ST program "ANY\_TO\_BYTE ()" to convert SINT into BYTE, and then use an "ANY\_TO\_INT" Function to convert the packed WORD into INT type.



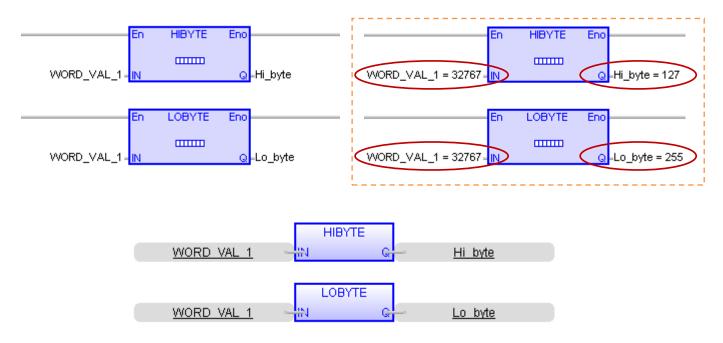
## Unpack One 16-bit Data to Two 8-bit Data

If want to unpack one WORD (or UINT) to 2 Byte (or USINT), you can use "HIBYTE", "LOBYTE" Functions.

ST Program:

Hi\_byte := HIBYTE (WORD\_VAL\_1); Lo\_byte := LOBYTE (WORD\_VAL\_1);

## LD/FBD Program:



If want to unpack one INT to 2 SINT, you must first use an ST program "ANY\_TO\_WORD()" to convert INT into WORD, and then use an "ANY\_TO\_SINT" Function to convert the unpacked BYTE into SINT type.



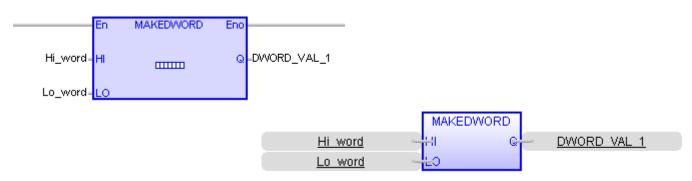
## Pack Two 16-bit Data into One 32-bit Data

If want to pack 2 WORD (or UINT) into a DWORD (or UDINT), you can use a "MAKEDWORD" Functions.

#### ST Program:

DWORD\_VAL\_1 := MAKEDWORD (Hi\_word, Lo\_word);

#### LD/FBD Program:



If want to pack 2 INT into 1 DINT, you must first use an ST program "ANY\_TO\_WORD()" to convert INT to WORD, and then use an "ANY\_TO\_DINT" Function to convert the unpacked DWORD into DINT type.



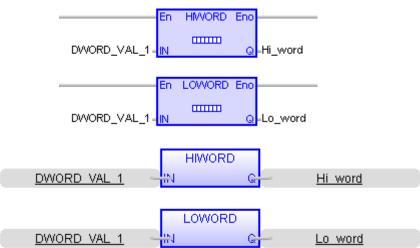
#### Unpack One 32-bit Data to Two 16-bit Data

If want to unpack one DWORD (or UDINT) to 2 WORD (or UINT), you can use "HIWORD", "LOWORD" Function Blocks.

ST Program:

Hi\_word := HIWORD (DWORD\_VAL\_1); Lo\_word := LOWORD (DWORD\_VAL\_1);

LD/FBD Program:



If want to unpack one DINT to 2 INT, you must first use an ST program "ANY\_TO\_DWORD()" to convert DINT to DWORD, and then use an "ANY\_TO\_INT" FB to convert the unpacked WORD into INT type.



# 10.7 Unpack Variable to Byte Array or Pack Byte Array into Variable

"SerializeOut" Function can unpack a Win-GRAF Variable value to a Byte Array (or USINT Array); "SerializeIn" Function can pack a Byte Array (or USINT Array) into a Win-GRAF Variable value.

## Note: 1. The Dim. of Array must be set as at least "8".

2. This "Serialize" Function can not use the STRING variable.

You can open the "HTML Help" from the menu bar, and enter the searching key word to see the detail setup instructions.

<u>H</u> elp		💕 нтмL н	elp		
3	Topics			• ⇒ [\$]	
Q	Search	隱藏頁籤	尋找 上一	·頁 下一頁 重新整理	道 理 首頁
	T <u>u</u> torials	内容(C)	搜尋(S)	我的最愛①	
?	<u>A</u> bout	請輸入欲搜索	尋的字串(₩):		
	La <u>ng</u> uage	serializeout			v )
		選取主題(工)	:	列出主題(L) 已找到:	[顯示D] 3
		標題		位置	順序
		SerializeOut		IEC 61131-3	1
		Advanced of	operations	IEC 61131-3	2 3
		SerializeIn		IEC 61131-3	3

If the SerializeOut() and SerializeIn() return "0", it means the saving location is wrong or the Array's space is not enough.

(\* Declare TMP\_DINT as a DINT,

buf as a BYTE Array, Dim. = 10, DINT\_Val as a DINT, Word\_Val as a WORD, REAL Val as a REAL \*) <u>Note:</u>

Data Type	Byte
BOOL, SINT, USINT, BYTE	1
INT, UINT, WORD	2
DINT, UDINT, DWORD, REAL	4
LINT, LREAL	8

## Example 1

(\* To unpack one DINT\_Val to 4 Bytes, and save them separately to the buf[2], buf[3], buf[4] and buf[5] from the location 2 of the BYTE Array in the "Little Endian" sequence. \*)

TMP\_DINT := SerializeOut (buf, DINT\_Val, 2, FALSE) ;

**Note:** The last parameter is "FALSE", that means to use the "Little Endian" sequence (To save the Low Byte to the starting address).

#### Example 2

(\* To unpack one Word\_Val to 2 Bytes, and save them separately to the buf[0] and buf[1] from the location 0 of the BYTE Array in the "Big Endian" sequence. \*)

TMP\_DINT := SerializeOut (buf, Word\_Val, 0, TRUE) ;

**Note:** The last parameter is "TRUE", that means to use the "Big Endian" sequence (To save the High Byte to the starting address).

#### Example 3

(\* To pack the buf[0], buf[1], buf[2] and buf[3] in the BYTE Array into one REAL\_Val in the "Little Endian" sequence. \*)

TMP\_DINT := SerializeIn (buf, REAL\_Val, 0, FALSE) ;

**Note:** The last parameter is "FALSE", that means to use the "Little Endian" sequence (To save the Low Byte to the starting address).

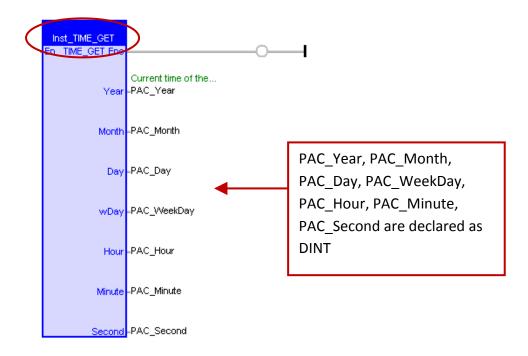
#### Example 4

(\* To map one DINT\_Val to one REAL\_Val in the "Little Endian" sequence. \*)

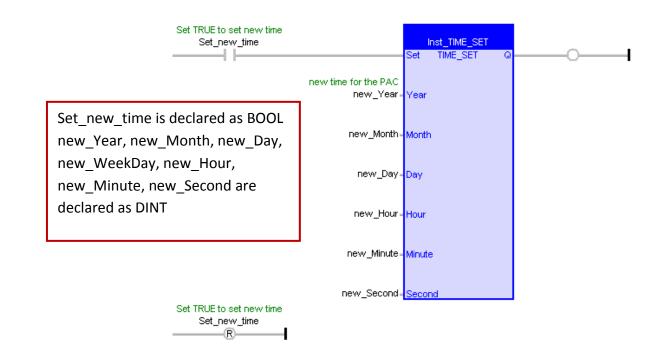
TMP\_DINT := SerializeOut (buf, DINT\_Val, 0, FALSE) ; TMP\_DINT := SerializeIn (buf, REAL\_Val, 0, FALSE) ;

# 10.8 Get/Set the PAC Time

If you want to get the current time of a Win-GRAF PAC, you can use a "TIME\_GET" Function Block. (Refer the <u>Section 2.2.1</u>)



If want to adjust the Win-GRAF PAC time, you can use "TIME\_SET" Function Block. (Refer the <u>Section</u> 2.3.6) First, fill the new time to the variables of "new\_Year", "new\_Month", "new\_Day", "new\_WeekDay", "new\_Hour", "new\_Minute" and "new\_Second", then set the "Set\_new\_time" to "TRUE" one time.



# Chapter 11 Commonly Used Tools and Useful Tips

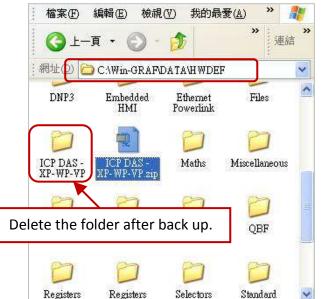
# 11.1 Upgrade Win-GRAF Libraries

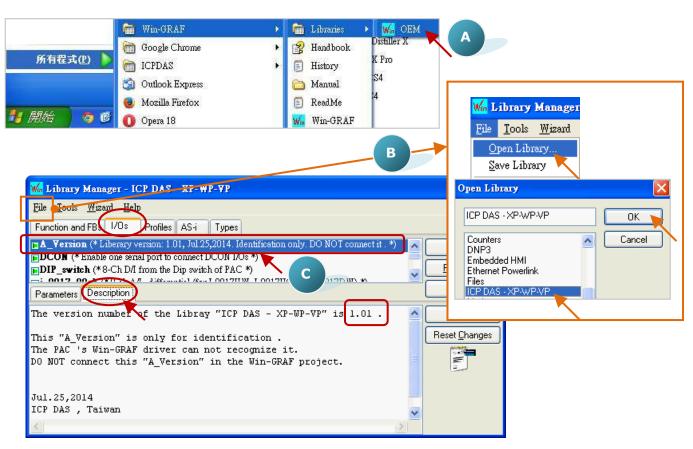
Users can download the latest version of Win-GRAF libraries (Win-GRAF Lib, e.g., "win-graf-lib-x.xx.zip") from <a href="http://www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.html">http://www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.html</a>. The Win-GRAF Libraries (Including Function, Function Block and I/O Board definitions) are saved in the folder of "ICP DAS - XP-WP-VP" under the directory of "C:\Win-GRAF\DATA\HWDEF". In some situations, you need to upgrade the Win-GRAF Libraries to the new version for supporting more Functions or new I/O Board. Please follow the steps below:

- 1. First, close all Win-GRAF Workbench windows.
- You can compress the original "ICP DAS XP-WP-VP" folder and back it up to the other directory (e.g., D:\temp\xxx.zip), and then delete the folder.
- Copy the new "ICP DAS XP-WP-VP" folder into the directory "C:\Win-GRAF\DATA\HWDEF", and execute the Win-GRAF Workbench again.

#### Note:

If you want to know the version number of the Win-GRAF library. As the figure below, open the "Library Manager" and click "A\_Version" in the "I/Os" tab, and then click "Description" to see the version number (e.g., "1.01").





# 11.2 Upgrade Win-GRAF Driver

For updating add-on functions, I/O boards or other purposes, ICP DAS will release a new version of Win-GRAF drivers in the future. Users can get the latest driver on the website (<u>http://www.icpdas.com/</u><u>root/product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.html</u>), and follow the steps below to upgrade the new driver into the PAC.

<u>Note:</u> The Win-GRAF Driver of the XPAC (XP-8xx8-CE6), WinPAC (WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7) and ViewPAC (VP-x2x8-CE7) will be placed in the directory "\System\_Disk\Win-GRAF\" inside the PAC.

1. On the desktop of a PAC (use WP-8xx8 in this example), double click on the "Win-GRAF\_WP\_8000" icon and then click "End Driver" to stop the currently running driver.

<b>し</b> 我的裝置		
B	Win-GRAF-WP-8000	OK
Internet Explorer	WD DwyD driver Version 1 01 - Jun 06 2014	
	WP-8xx8 driver Version 1.01 , Jun.06,2014	
Win_GRA	This product is licensed.	
WordPad		End Dover
		$\frown$
🛃 🏹 Virtual CE		🔡 🌙 上午 08:04 🏓 🗟

- 2. On the PC, copy the new driver into the PAC's directory "\Temp\" by using FTP method.
- 3. On the PAC, copy the new driver from "\Temp\" into "\the System\_Disk\Win-GRAF\" directory to replace the old one, and then reboot the PAC.

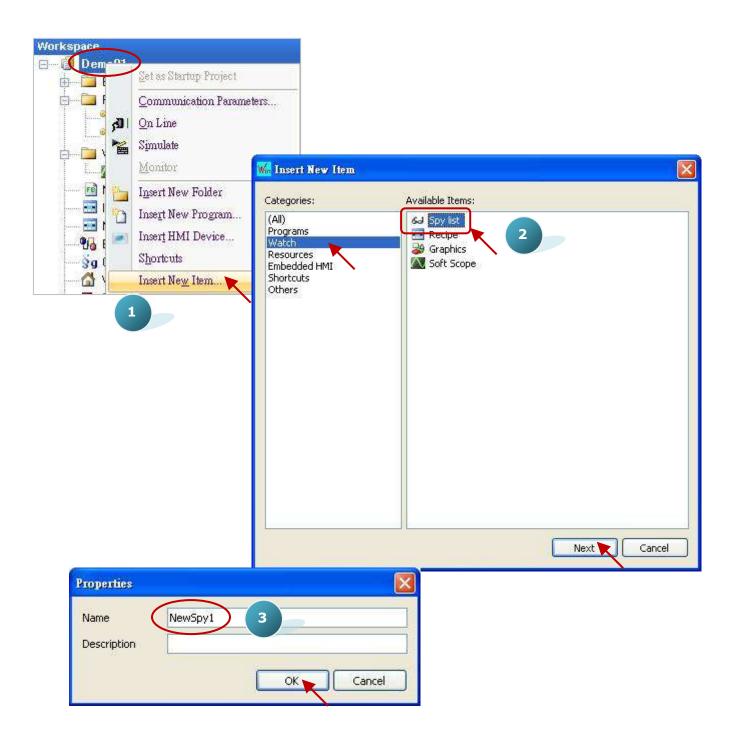
網址( <u>D</u> )	\System_Disk\win-graf			
📓 License. 🎇 Win_GR.		) Quicker.dll Win_GRAF_WP_8000.lnk	🔊 t5.cod	🛋 t5.upd

# 11.3 Spy List

When a program is running, the Spy List lets users quickly know the variable's value or status. Sometimes, a program may declare hundreds or thousands of variables, users do not need to look for them, just simply switch to the pre-created Spy List window to see the wanted information.

Steps:

- 1. Right-click on the project name (e.g., "Demo01") and select "Insert New Item".
- 2. Select "Spy List" of the "Watch", then click "Next" to the next step.
- 3. Then, key in a list name (e.g., "NewSpy1") and press "OK".



4. Double click "NewSpy1" on the left side to open the setting window and drag the variables you want to put into the window.

Workspace	Nev	vSpy1.spl *						ΗZ	>
🖂 🔐 🗿 Demo01	<b>D</b>	Name	Value	Description	Y	Name	Туре	Dim.	
🖶 🛅 Exception programs	53-1	PAC_Year		Current time of the PAC		🗉 🚮 Global varial	oles		1
📩 — 📴 Programs	8-	PAC_Month				PAC_Year	DINT		I
PAC_Time (*Get / Set P	∎+	PAC_Day				PAC Month	DINT		P
🛄 LD1 (*Blinking*)		PAC_WeekDay	×			PAC_Day	DINT		ŀ
🗄 🛁 Watch (for debugging)		PAC_Hour	1 N.			PAC WeekDay	DINT		L
Soft Scope		PAC_Minute				PACHour	DINT		L
🖻 NewStructure		PAC_Second		<u>``</u>		PAC Minute	DINT		L
🔜 Initial values		⊞ INT_Val			-	PAC Second	DINT		L
NewRecipet	-					INT Val	INT	[02]	L
NewSpy1		1993				REAL Val	REAL	[03]	
Binding Configuration		<		>		< / m		5	ſ
🚽 😼 g Global defines	4 )	LD1 PAC Time	Variables Ne	w Spy 1					

5. When the Win-GRAF and the PAC are connected, the "NewSpy1" window will show the variables information clearly.

rkspace		NewSpv1.spl							H 🖊 >
Demo01 [RUN]		Name	Value	Description	T Name	Value	Туре	Dim.	Attrib
🗄 🛅 Exception programs		PAC_Year	2014	Current time of the PAC	🗉 🔂 Global varial	oles	And the second second		
E Programs	H	PAC_Month	6		PAC Year	2014	DINT		
PAC_Time (*Get / Set P	∎+	PAC_Day	13		PAC Month	6	DINT		
LD1 (*Blinking*)	0.755	PAC_WeekDay	5		PAC Day	13	DINT		
🗄 🔚 Watch (for debugging)		PAC_Hour	5 5 13		PAC WeekDay	5	DINT		
Soft Scope		PAC_Minute	13		PAC_Hour	5	DINT		
🖻 NewStructure		PAC_Second	0		PAC Minute	13	DINT		
📰 Initial values		🖂 INT_Val			PAC_Second	0	DINT		
		INT_Val[0]	55		INT_Val		INT	[02]	
NewSpy1		INT_Val[1]	36		REAL_Val		REAL	[03]	
		INT_Val[2]	100		new_Year	0	DINT		
		🖂 REAL_Val			new_Month	0	DINT		
		REAL_Val[0]	5.8		new_Day	0	DINT		
I 🖪 Types		REAL_Val[1]	9.6		new_Hour	0	DINT		
		REAL_Val[2]	3.2		new_Minute	0	DINT		13
		REAL_Val[3]	2.5		Conord	0	DINT		>

#### Back up A Win-GRAF Project:

1. Mouse right-click on the project name (e.g., "Demo01") and select "Save Project" and then "To Disk".

	2 · · · · · · · · · · · · · · · · · · ·		Contraction of the second seco		
	Communication Parame		Insert Ne <u>w</u> Item		
ା 📲 P 🚮 🖸	Dn Line		Save Project		To Disk
	Simulate	B	Duplicate Project	9	T <u>o</u> Zip
🔤 🔤 🔤	Monitor	3	Print Project	6	To Zip and Mail
🔜 🖻 NewS 🏊 🛛	nsert New Folder	Q	Project Description		To Target

 Click on the "Browser" button to assign a directory you want to save the project (e.g., D:\Win-GRAF\_demo\_backup), fill in the project name (e.g., "Demo01\_0613"), and then click "OK" to back up the project.

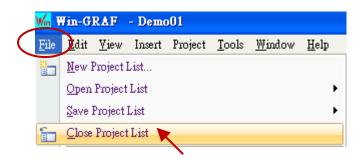
🚾 Save as		
Current Project C:\Win-GRAF\Pro	jects\Demo01	
Save as:	Project	<b>v</b>
Destination: Project Name:	D:\Win-GRAF_demo_backup Demo01_0613 2	1 Browse
		OK Cancel Help

## **Restore A Win-GRAF Project:**

 Copy the previously backed up the project folder (e.g., "Demo01\_0613") into "C:\Win-GRAF\Projects".



2. Click the menu bar "File" > "Close Project List" to close all opened project windows.



3. Click the menu bar "File" > "Add Existing Project" > "From Disk", select the project you want (e.g., "Demo01\_0613") in the "C:\Win-GRAF\" directory, and then click "OK" to restore the project.

File	jiew <u>T</u> ools <u>W</u> indow <u>H</u> elp	1			
	<u>N</u> ew Project List Open Project List <u>A</u> dd New Project	•	16 III & U	瀏覽資料 <del>夾</del> Select project	? 🛛
	Add Existing Project		From Disk.	🖃 🛅 Projects	~
	Print Setup <u>1</u> C:\Win-GRAF\Projects\Demo01.w51 <u>2</u> C:\Win-GRAF\Projects\Test.w51	æ	From Zip From Target Fro <u>m</u> XML Exp	🕀 🧰 MTCP_M	
				田 (m) other     田 (m) Test     □ (m) Test	▶ ▶

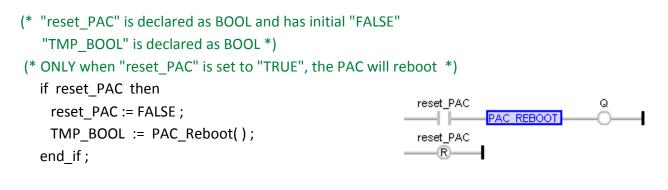
🜃 Win-GRAF - Demo01_0613	3	
<u>File E</u> dit <u>V</u> iew Insert Project	<u>I</u> ools <u>W</u> indow <u>H</u> elp	
BBBBBBBB	, X 🛒 `` 이 여 ' 🕌   🎹 🏯 😘 😨 🖓 🖪 🍃 🔐 🖀	ا 🕼 🚆
Workspace	No selection!	XN
□		
Exception programs		
🖻 🦳 🧰 Programs		
🔤 🖭 LD1 (*Blinking*)		
PAC_Time (*Get	Duild	×
🖻 🔤 Watch (for debuggi	Build	~
Soft Scope		
NewStructure		
📰 Initial values		
6J NewSpy1		
96 Binding Configuration		
g Global defines		
Variables		
🖾 E Types	Build Cross references Runtime Call stack Breakpoints	Digital sa
Ready	OffLine 192.168.71.19:502	11

# 11.5 Software Reboot a PAC

Based on some cases, users may want to reboot the PAC in a software way. The Win-GRAF provides a Function "PAC\_Reboot" for users to restart the PAC.

**Note:** Please DO NOT call this Function in every PAC Cycle, or the PAC will reboot all the time.

#### Safety Coding:



#### **Dangerous Coding:**

(\* "TMP\_BOOL" is declared as BOOL \*)

## (\* Dangerous ! This coding method will let the PAC reboot always and cannot stop. \*) TMP\_BOOL := PAC\_Reboot();

If a mistake to reboot the PAC always, turns the rotary switch of the Win-GRAF PAC to "1" and reset it. Then it will boot up in safe mode. Then you may rename the Win-GRAF application code on the PAC to an invalid name. Then when the rotary switch is turned back to "0" and reboot, it will boot up normally (No application). The Win-GRAF application code in the XP-8xx8-CE6, WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7, and VP-x2x8-CE7 is "\System\_Disk\Win-GRAF\t5.cod".

# 11.6 Using ST Syntax in LD and FBD

The Win-GRAF Workbench allows users using simple ST syntax in Ladder (LD) and Function Block Diagram (FBD) to facilitate programming. Before use, go to the menu bar "Project"> "Settings" > "Runtime", and set the "Complex variables in a separate segment" to "Yes" to enable this function.

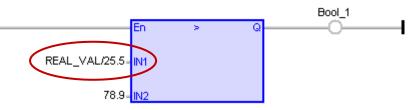
🚾 Win-GRAF – Test 👘				
<u>File E</u> dit <u>V</u> iew Insert I	Project <u>T</u> ools <u>W</u> indov	v <u>H</u> elp		
😂  🖹 🛃 🔏	🔁 Build All Projects	F7		
Workspace	<u>C</u> lean All Projects			
🖃 👘 🚺 Test	👬 Download All Proje	ects		
🗄 🚞 Exception pro	Settings			
📥 🔤 Programs 📙	Dotmigo			
Ргој	ject settings			×
C:\W	/in-GRAF\Projects\Tes	it		
	neral	Name		Value
	ntime npiler	🥶 Runtime system 👘		Little endian
	bugging	🐒 Cycle time		0
Adv	/anced	💕 Code Generation		Release
(All)	)	🔀 Exchange IOs while	e stepping	Yes
		🧠 Support variables lo	cking	No
		[i]. Complex variables i	n a separate segment	Yes

#### Example:

Wa

## LD Program:

Using division (REAL\_VAL/25.5).



#### FBD Program:

Call a function "ANY\_TO\_BYTE()" to convert the type from "SINT" to "BYTE".

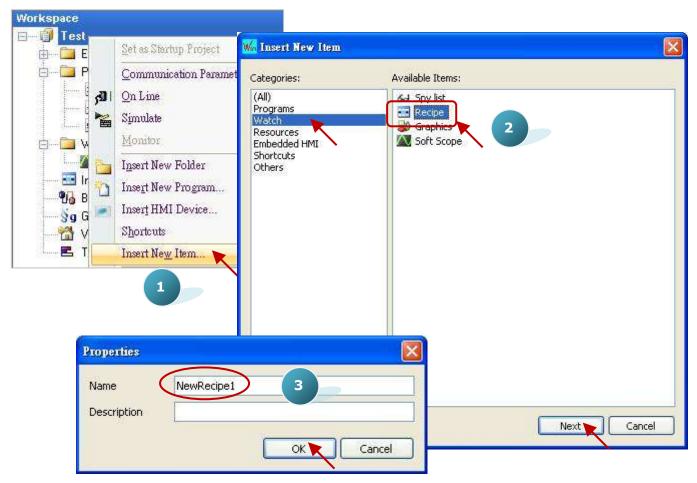


# 11.7 Apply a Recipe on the PAC

Some applications use the pre-defined Recipe and Value for processing different products, and this Recipe can be mapped to a combination of variables within a Win-GRAF PAC. When one day want to change the PAC process to produce a different product, you can use the Win-GRAF Workbench to connect with the PAC and select a new Recipe you want to replace, and then apply it to the PAC.

#### Steps:

 Mouse right-click on the project name (e.g., "Test") and select "Insert New Item"; then click on "Watch" > "Recipe" and "Next" button, fill in the Recipe name (e.g., "NewRecipe1") and then click "OK".



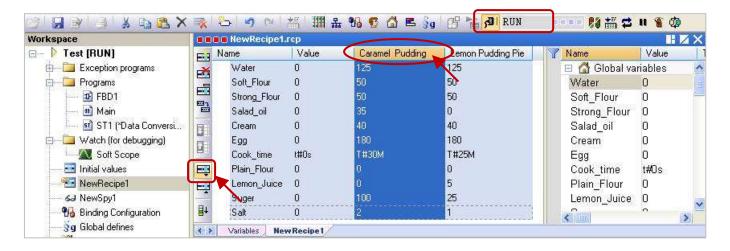
2. Double click "NewRecipe1" on the left side to open the setting window and drag the variables you want to put into the window.

Workspace	Nev	Recipe1.rcp	*				1: 12	×
🖂 👘 Test	<b>E</b> 3	Name	Value	7	Name	Туре	Dim.	A
🗄 🛅 Exception programs		Water			🗉 🚮 Global va	riables		~
📩 🛅 Programs	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Soft_Flour			Water	DINT		
🔤 📴 FBD1	100	Strong_Flo	our		Soft Flour	DINT		-
🔜 🐽 Main		Salad_oil			Strong_Flour	DINT		
🔜 🖬 ST1 (*Data Conversi		Cream			Salad oil	DINT		
📩 🚞 Watch (for debugging)	E-	Egg	· · · · ·		Cream	DINT		
Soft Scope		Cook_time		1	Egg	DINT		
Initial values	-	Plain_Flou	u .		Cook_time	TIME		
NewRecipe1 🚬	E	Lemon_Ju	lice		Plain Flour	DINT		
Sa NewSpy1		Suger			Lemon Juice	DINT		
1 Binding Configuration	<b>₿</b> +				Suger	DINT		~
🔤 🕺 🚽 🚽 🚽 🚽					<		>	1
🚽 🛗 Variables	< >	Variables	New Recipe 1	10				

3. Click on the "Insert Column" icon to add the new Recipe, and then fill in the suitable Values.

Name	Value 🤇	Caramel Pudding	Lemon Pudding Pie	Y	Name	Туре	
Water	144	125	125		🗉 🚮 Global va	riables	1
Soft_Flour Strong Flour		50	50		Water	DINT	
Strong_Flour Salad oil		50	50		Soft_Flour	DINT	
Salad_oil		35	0		Strong_Flour	DINT	-
Cream		40	40		Salad_oil	DINT	
Egg		180	180		Cream	DINT	
Cook_time		T#30M	T#25M		Egg	DINT	
Plain_Flour		0	0		Cook_time	TIME	
Lemon_Juice		0	5		Plain_Flour	DINT	
Suger		100	25		Lemon_Juice	DINT	
Salt		2	1		Suger	DINT	
					Salt	DINT	1
					<		>

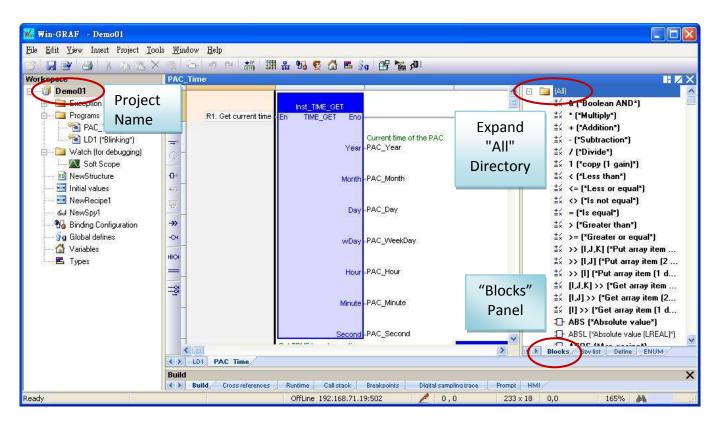
4. Click on the "On Line" button to connect the PAC. At first the Values are all "O", please select the product column and then click the icon "Send Recipe" to apply this Recipe into the PAC.



	NewRecipe1	.гер			
Ež.	Name	Value	Caramel Pudding	Lemon P	udding Pie
	Water	125	125	125	
	Soft_Flour	50	50	50	Note: If want to save the Values on
	Strong_Flour	50	50	50	the PAC (e.g. Power off and restart
E	Salad_oil	35	25	0	the PAC, the Recipe can still retain
	Cream	40	40	40	the previous Values), please refer
	Egg	180	180	180	the method of using the retain
	Cook_time	t#30m	T#30M	T#25M	5
E₽₽	Plain_Flour	0	0	0	variables in the <u>Section 6.1</u> .
E	Lemon_Juice	0	0	5	
-	Suger	100	100	25	
∎+	Salt	2	2	1	
< >	Variables Ne	wRecipe1	·		

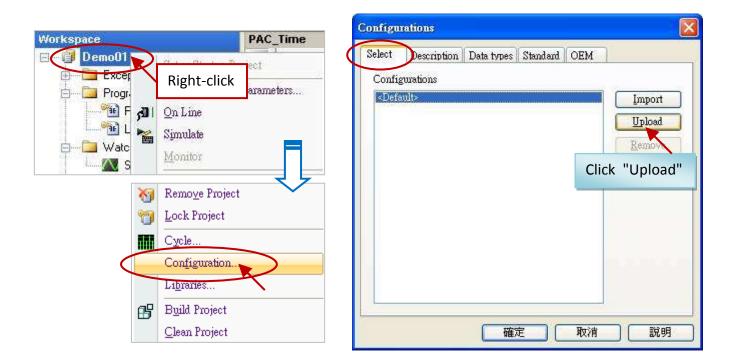
# 11.8 Get the Functions and Function Blocks that Supported by the PAC

In the Win-GRAF Workbench window, user can expand the "All" directory in the "Blocks" panel to see quite a lot of Functions and Function Blocks, however, some are not supported in the Win-GRAF PAC. The following will show how to quickly get the Functions or Function Blocks that supported by the PAC.



## Setting Steps:

- 1. Make sure the PAC is powered on and connected with a PC via an Ethernet cable.
- In the Win-GRAF Workbench, right-click on the project name (e.g., "Demo01") and then select "Configuration", and click on the "Upload" button in the "Select" tab to open the setting window.



Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 11-11

3. Configure/Select the IP address of the PAC and click on "OK" button, the PAC will upload the configuration file. Next, key in the file name (e.g., "test.cfg") and click "Save" to save the configuration file.

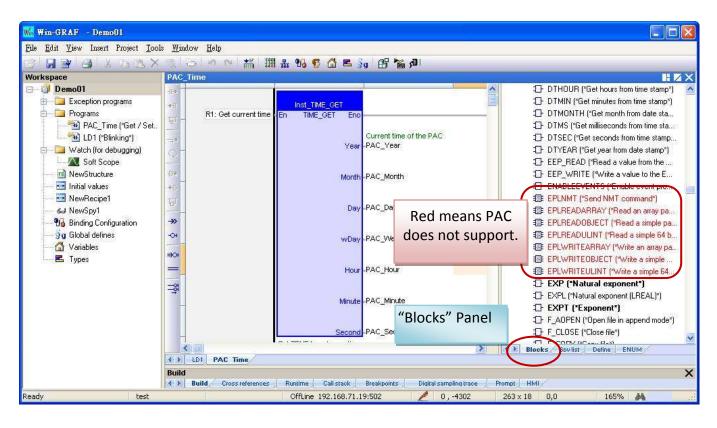
T5 Runtime	*		Upload the configuration	on
192.168.71.19:502			figurations	
192.168.255.1:502 192.168.71.18:502 192.168.71.19.502	Setup the IP addr. c	Browse Help of the PAC	Uploading runtime configuration. Please wait.	
另存新檔			? 🛛	
儲存於	(I): 🔁 CONFIG 🔪		🕑 🧿 😰 🖽 -	
我最近的文件	Save file un	nder C:\Win-GRA	F\DATA\CONFIG	
「「」」				
多数的文件				
我的電腦		Give the	file name & save it.	
( C	檔名(N): test.	cfg	✓ 儲存⑧▼	

4. Back to the "Configurations" window, the configuration file (test) will show in the list, and then click "OK" to leave this window.

Select	Description	Data types	Standard	OEM	
	rurations	2000 () 200			
<defa test</defa 					Import
RESI					Upload
					Remove
7				110	
		確	E D C	取消	〕 説明

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 11-12

5. In the "Blocks" panel, the red Functions and Function Blocks are not supported by this PAC.

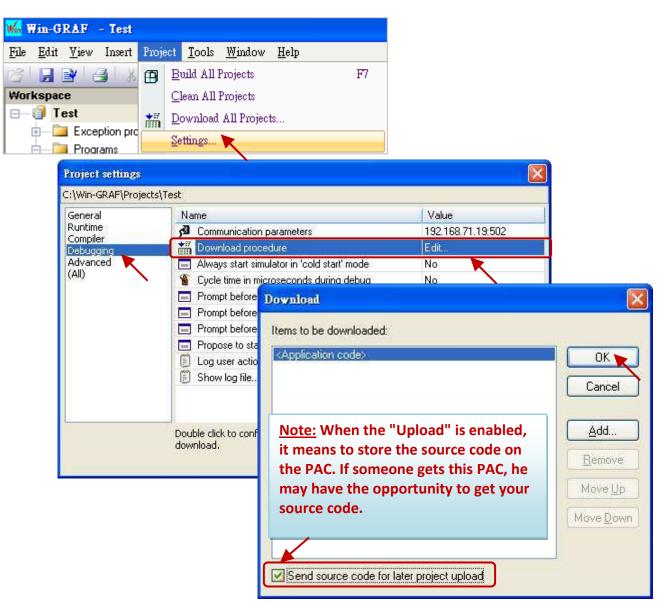


# 11.9 Upload the Win-GRAF Source Code

For some applications, users need to get the source code of the Win-GRAF project from the PAC to the PC, this is called "Upload". This function can prevent the project source code from missing or incomplete handover from the previous worker, you can still get the project source code inside the PAC.

## Enable/Download The Project Source Code:

- 1. Click on the menu bar "Project" > "Settings" to open the setting window.
- 2. Double click on the "Download procedure" of the "Debugging" and select "Send source code for later project upload", and then click "OK ".



3. Click on the menu bar "Project" > "Build All Projects" to compile the program, and then click on the tool icon to connect the PAC, and next, download the current project to the PAC. (Refer the <u>Section 2.3.5</u> for the detail steps). After downloading, the source code will be stored in the file of the directory "\Systen\_Disk\Win-GRAF\t5.upl" of the PAC. This file will be larger when the project increases. If the project becomes very large and complex, the file size may reach several hundreds K Bytes or even more than 1 MB.

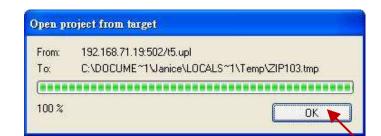
## Upload the Porject Source Code:

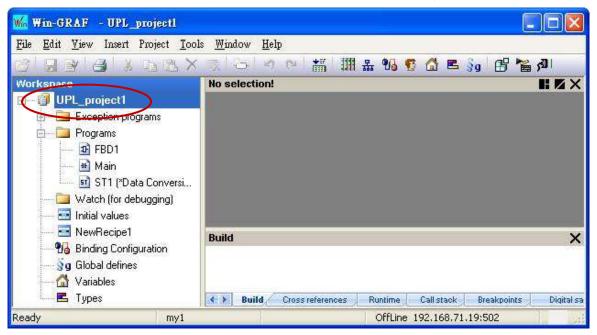
Please close all opened Win-GRAF windows (Click on the menu bar "File" > "Close Project List").

 Click on the menu bar "File" > "Add Existing Project" > "From Target", then select the PAC's IP address and set up the upload file name (e.g., "UPL\_project1"), and then click "OK" to upload the file.

File	<u>Y</u> iew <u>T</u> ools <u>W</u> indow <u>H</u> elp			
	<u>N</u> ew Project List Open Project List	•	5 m # 16 C G I	
1	Add New Project			
-	Add Existing Project	-	From Disk	
	Print Setup	9	From Zip	
	1 C:\Win-GRAFProjects\Test.w51		From Target	
	2 C:\Win- W/ Open project from target Settings T5 Runtime Pathnames Destination folder : C:\Win-GRAF\P Name: UPL_project1	rojects	Select PAC IP 8.71.19:502	OK Cancel Help

5. After uploading, click on "OK" button, and then the Win-GRAF will open the project automatically.

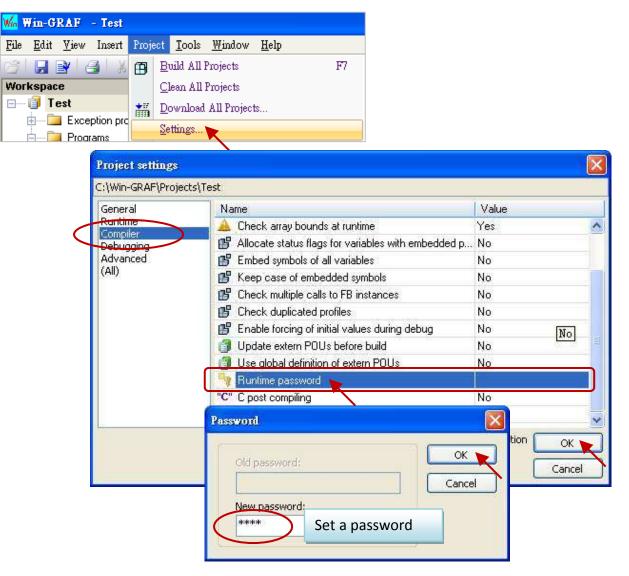




# 11.10 Set Up the PAC Password

In order to avoid the important program running on the PAC is changed or stopped by an unfriendly connecting PC, you can set up a password for the PAC to prevent unauthorized operation.

- 1. Click on the menu bar "Project" > "Settings" to open the setting window.
- 2. Double click on the "Runtime password" of the "Compiler", set a password, and then click "OK".



 Click on the menu bar "Project" > "Build All Projects" to compile the program again, and then download the current project to the PAC (Refer the <u>Section 2.3.5</u> for detail steps.). When the next time to click the "On Line" icon for connection, it will require the password.

% 🖸 🖾 🗉 §g 🔐 🏀 🚰 Domentais	
Password	
Password:	ОК
If click "On Line", it need the password.	Cancel

**Note:** After enabling the password, please remember your password, or you will not connect the PAC.

#### The Only Solution:

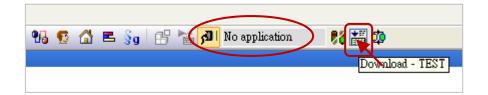
- 1. Connect the PAC with a USB mouse and screen.
- 2. On the PAC, execute the Win-GRAF Driver and then click on "End Driver" button. (Refer the <u>Section 11.2</u>).

Win-GRAF-WP-8000	OK
WP-8xx8 driver Version 1.01 , Jun.06,2014	
This product is licensed.	
End Dr	iver
	$\mathbf{N}$

3. Rename the file "t5.cod" in the directory "\System\_Disk\win-graf" (e.g., "t5.cod1") or delete it. Then reboot the PAC.

網址(D) System_Disk\win-graf			
🖬 License.bin 🐺 Win_GRAF_WP_8000.exe	🔊 Quicker.dll 🎇 Win_GRAF_WP_8000.lnk	t5.cod	at5.upd

Then, it will become "No application" on the PAC. Now, you can connect and download the application from the Win-GRAF Workbench again.



# 11.11 Using Function Block in the ST Program

It is easy to use the Function in the ST program, just call the Function and assign the corresponding parameters. The example below will open COM3 at the beginning, and then send a String `Hello' from COM3 every 5 seconds.

```
(* Declare "INIT1" as BOOL and has initial value TRUE,
Declare "TMP_BOO" as BOOL, "TMR1" as TIME *)
IF INIT1 THEN
INIT1 := FALSE;
TMR1 := T#0s;
TSTART (TMR1);
END_IF;
IF COM_Status(3) = FALSE THEN
TMP_BOO := COM_open (3, `19200,N,8,1');
END_IF;
IF TMR1 >= T#5s THEN
TMR1 := T#0s;
COM_send_str (3, `Hello: ');
END_IF;
```

To use a Function Block in the ST, you must first declare an Instance Variable of the Function Block in the variable region, after that, the using steps are similar to the steps of using the Function, as follows:

The following code can unpack one Byte to become 8 BOOLs:

1. Declare "MyUnpack" variable as "UNPACK8" (FB Instance) and "IN" variable as "BYTE".

7	Name	Туре	Dim.	Attrib.	Syb.	Init value	U V	Tag	Description
	🗉 🚮 Global vari	ables							
	IN	BYTE							
	MyUnpack	UNPACK8							

2. Edit an ST program.

MyUnpack(IN);

- Q0 := MyUnpack.Q0;
- Q1 := MyUnpack.Q1;
- Q2 := MyUnpack.Q2 ;
- Q3 := MyUnpack.Q3 ;
- Q4 := MyUnpack.Q4 ;
- Q5 := MyUnpack.Q5;
- Q6 := MyUnpack.Q6 ;
- Q7 := MyUnpack.Q7 ;

# 11.12 How to Protect Your Win-GRAF Program to Avoid Unauthorized Copied?

When you finish a Win-GRAF application development and prepare for delivery to the customer, please think about the possibility that your Win-GRAF application on the PAC may will be copied into another same model PAC?! Be careful! Someone else may steal your hard outcome! The following provides a simple and easy way to protect your application.

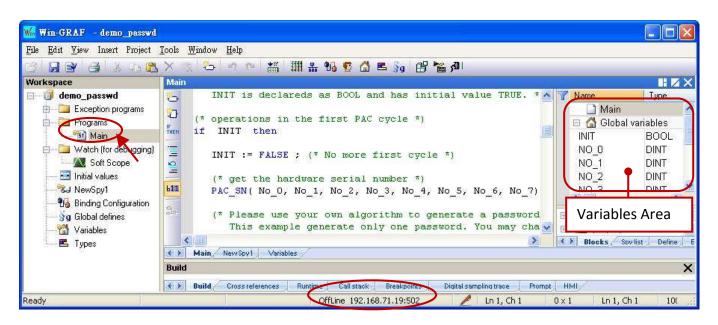
<u>Note:</u> If you give the Win-GRAF application Source Code to the customer, then sorry, the following method will not protect your program from stolen. Because having the Source Code, anyone can modify the code and apply into another PAC.

Each ICP DAS Win-GRAF PAC has a Serial Number that has 8 Bytes (also known as 64-Bit), and each PAC has the different and unique Serial Number. Therefore, you can use this serial number combine with your own algorithm to generate a password, and pre-store this password on the PAC's file. Then, verify this password in your application. If not passed, the application will not be allowed to execute.

The steps are as follows:

This example uses two Win-GRAF projects, one is "demo\_passwd" used to generate a PAC password and save it into the PAC's file; the other is "demo\_my\_ap" application that has been developed and ready to ship to the customer. Before shipping a PAC to the customer, user needs to download the project "demo\_passwd" into the PAC and runs it once to generate a unique password for that PAC. Then, downloads the project "demo\_my\_ap" into the same PAC, and then can ship the PAC to the customer. After that, if someone copies the Win-GRAF application in this PAC to another same type of PAC and the operation will fail because of the password validation failure.

There are two example projects (demo\_passwd.zip and demo\_my\_ap.zip) in the shipment CD (\Napdos\Win-GRAF\demo-project), please refer the <u>Chapter 12</u> to restore the projects (Execute File > Add Existing Project > From Zip) and set up the IP address of the current PAC.



#### "demo\_passwd" Project:

This program first uses "PAC\_SN" Function to read out the Serial Number, and then uses a user-defined algorithm to generate a password. Finally, save the password to an address within the PAC's file (user can decide where you want to store).

## Variable Declaration:

Workspace	Variables					
🖃 🗐 demo_passwd	🍸 Name	Туре	Dim. Attrib.	Syb.   Init value   User	. Tag	Description
💼 📴 Exception programs	🗉 🚮 Global	variables				· · · · · · · · · · · · · · · · · · ·
📋 🚞 Programs	INIT	BOOL	5. S.		1	init as TRUE
🔚 🎦 Main	NO_0	DINT				8-byte hardware serial number
🚊 🛄 Watch (for debugging)	NO_1	DINT				
Soft Scope	NO_2	DINT				
🛅 Initial values	NO_3	DINT				
NewSpy1	NO_4	DINT				
🚽 🚮 Binding Configuration	NO_5	DINT				
gg Global defines	NO_6	DINT				
Variables	NO_7	DINT				
E Types	password	DINT				
Provinsion of the state of the	<				12	
	K Main New	Spy1 Variable	s			

#### ST Program:

- (\* This "demo\_passwd" example will generate a password by the 8-Byte Serial Number of the PAC and save it into the EEPROM of the PAC \*)
- (\* Declare "No\_0" ~ "No\_7" and "password" variables as DINT. Declare "INIT" variable as BOOL and has Initial value TRUE. \*)

```
(* Operations in the first PAC Cycle *)
```

```
if INIT then
```

```
INIT := FALSE ; (* No more first cycle *)
```

(\* Get the hardware serial number \*) PAC\_SN(No 0, No 1, No 2, No 3, No 4, No 5, No 6, No 7);

(\* Please use your own algorithm to generate a password. This example generate only one password. You may change it to generate some passwords. \*)

```
password := (No_0 * No_1) + (No_2 * 12345) + No_3 + (No_4 * No_5) + No_6 + No_7;
```

```
(* save the password in a file "my_product.pwd" in the \System_Disk\Win-GRAF *)
file_name := '\System_Disk\Win-GRAF\my_product.pwd';
file_id := f_wopen(file_name);
if file_id = 0 then
    (* failed , do nothing *)
else
    (* open file ok, save the password into it *)
    fm_write( file_id , Any_to_String(password) );
    f_close( file_id ); (* close file *)
end_if;
end_if;
```

#### "demo\_my\_ap" Project:

This project first uses "PAC\_SN" Function to read out the Serial Number, then calculates the password, and then compares the password that read from the PAC's file to check if the password correct.

Workspace	Main			
🖃 👘 demo_my_ap		🍸 Name	Туре	
🗄 🖂 🖾 Exception programs	(* operations in the first PAC cycle *)	🗉 🚮 Global variables		
🖻 🔤 Programs	if INIT then	INIT	BOOL	
<sup>9</sup> stì Main	THEN	NO O	DINT	
*11 LD2	INIT := FALSE ; (* No more first cycle *)	NO_1	DINT	
🚊 🛁 Watch (for debuggi	(* get the hardware serial number *)	NO_2	DINT	
Soft Scope	PAC_SN( No_0, No_1, No_2, No_3, No_4, No_5, No_6, No_	NO 3	DINT	
🔜 Initial values			2	
NewSpy1	(* Please use your own algorithm for the PAC_password	🕑 🚞 (All)		
🚽 🚮 Binding Configurati	PAC_password:= (No_0 * No_1) + (No_2 * 12345) + No_3	🗄 🚞 (Project)		
🚽 😼 🚽 Global defines	(No_4 * No_5) + No_6 + No_7 ;	🗄 🚞 Advanced		
- 🚮 Variables		Arithmetic Blocks Sov lis	st Define EN	
E Types	Main LD2 NewSpy1	BIOCKS SDORE	A Denne X EN	

(Note: Refer the Section 2.1.2 to arrange the programs in the execution order.)

## Variable Declaration:

Workspace	Variables							
🗉 🥑 demo_my_ap	Wanie Name	Туре	Dim Athib	Syb.	Init value	User	Tag	Description
Exception programs	🖻 🙆 Global variable	18:	and the second state	1100000				
Programs	INIT	BOOL			TRUE			init as TRUE
Main	NO_0	DINT						B-byte hardware serial number
LD2	NO_1	DINT						
📄 🥘 Watch (for debuggi	NO_2	DINT						
Soft Scope	NO_3	DINT						
🔜 Initial values	NO_4	DINT						
- Sal NewSpy1	NO_5	DINT						
Binding Configurati	NO_6	DINT						
gig Global dennes	NO_7	DINT		-				
C1 Valables	password	DINT						
E Types	PAC_password	DINT						
<ul> <li>Non-mail and a second se</li></ul>	password ok	BOOL						TRUE: OK , FALSE: fail
`	LED1	BOOL						for demo purpose only. If password ok, blink per second
	LED2	BOOL						
	\$							

#### ST Program - Main:

- (\* This "demo\_my\_ap" example can read the password from the EEPROM of the PAC, and check if match with the result that calculated from the user's own algorithm. \*)
- (\* Declare "No\_0" ~ "No\_7", "password" and "PAC\_password" variables as DINT. Declare "INIT" variable as BOOL and has initial value TRUE. Declare "password\_ok" variable as BOOL \*)
- (\* Operations in the first PAC cycle \*)

if INIT then

INIT := FALSE ; (\* No more first cycle \*)

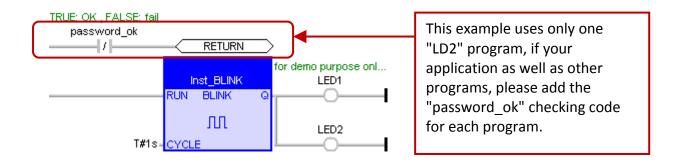
```
(* get the hardware serial number *)
PAC_SN( No_0, No_1, No_2, No_3, No_4, No_5, No_6, No_7);
```

```
(* Please use your own algorithm for the "PAC_password" value *)
PAC_password:= (No_0 * No_1) + (No_2 * 12345) + No_3 + (No_4 * No_5) + No_6 + No_7;
```

```
(* Read the password value from a file "my product.pwd" in \System Disk\Win-GRAF *)
 file name := '\System disk\Win-GRAF\my product.pwd';
 file_id := f_ropen( file_name ) ;
 if file id = 0 then
   (* can not open file, set password to 0 *)
   password := 0;
 else
   (* open file ok, read the password *)
   if f_eof(file_id) then
    (* reach the end of file *)
   else
    (* hasn't reached the end of file , read a string form it *)
    Tmp string := fm read( file id ) ;
    (* Convert a string to a DINT value *)
    password := Any to DINT(Tmp string);
   end if;
   f close(file id); (* close file *)
 end_if;
 (* check if the password is correct? *)
 password ok := FALSE; (* set it as "FALSE" in the beginning *)
 if password = PAC password then
   password ok := TRUE; (* the password is correct *)
 end if;
end if;
```

### LD Program – LD2

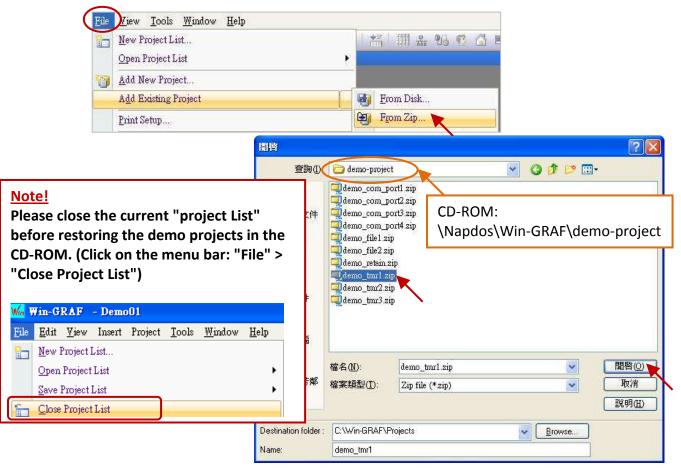
If the "password\_ok" is "FALSE", it means the password is incorrect, then will exit the program. Only when the password is correct can the program execute continuously, and then your application can be protected from the unauthorized access.



# Chapter 12 Description of Win-GRAF Demo Projects

There are some demo projects in the Win-GRAF shipping CD (CD-ROM: **\Napdos\Win-GRAF\ demo-project**) which some of them will be introduced in the following sections. Before using the demo projects, follow the steps below:

 Click on the menu bar "File" > "Add Existing Project" > "From Zip" to open a project (e.g., "demo\_tmr1.zip").



- 2. Double click "Main" to open the ST program, and can view/add variables in the variable area.
- 3. Mouse right-click on the project name ("demo\_tmr1"), and select "Communication Parameters" to set up the IP Address of your PAC. (Refer <u>Section 2.3.5</u>)

🜃 Win-GRAF - demo_tmrl					
File Edit View Insert Project I	Cools <u>W</u> indow <u>H</u> elp X 🔨 🏷 🗠 e	- 請 調品 % 5 🗂 E 埦 🕑 🎽 AI			
Workspace demo_tmr1 Exception programs Programs Waten (for debugging) Soft Scope Initial values Variables Global defines Variables Types	Main (* Set ST if START sTART_t START_t tStart( end_if; (* Set ST if STOP_tm tStop(T end_if; <b>Build</b> Cross	192.168.71.19:502           192.168.255.1:502           192.168.71.18:502           192.168.71.19:502	Cancel	Name     Type       Main     Main       Global variables       START_tmr     BOOL       STOP_tmr     BOOL       ED1     BOOL       ED2     BOOL       Image: Constraint of the state o	
Ready		OffLine 192.168.71.18:502	🖉 🛛 Ln 1, Ch 1	0 x 1 Ln 1, Ch 1 1	00%

# 12.1 The List of Demo Programs

The Win-GRAF product package includes a CD-ROM named Win-GRAF-PAC-CD, the user can find out more information on Win-GRAF demo programs in the path - \Napdos\Win-GRAF\demo-project. Refer the table below to restore these programs to the Win-GRAF Workbench.

The following table shows the list of the demo programs:

File Name	Description
demo_tmr1	Use the "tStart" and the "tStop" functions to operate the Timer (refer section 12.2.1).
demo_tmr2	To do periodic operations for the Timer (refer section 12.2.2).
demo_tmr3	To do periodic operations more accurately for the Timer (refer section 12.2.2).
demo_com_port1	Send a string by the COM Port (refer section 12.3.1).
demo_com_port2	Request/Anser the device by the COM Port (refer section 12.3.2).
demo_com_port3	Wait for data coming from the remote device to the COM Port (refer section 12.3.3).
demo_com_port4	Report data periodically to the remote device by the COM Port (refer section 12.3.4).
demo_file1	Write data to a file on the PAC (refer section 12.4.1).
demo_file2	Read data from a file on the PAC (refer section 12.4.2).
demo_retain	Use retain variables to save lately values before the PAC power failure (refer section 6.1).
demo_wp5_retain	Use file to retain variable values (refer section 6.2).
demo_extra_port	Enable a serial port for connecting the Win-GRAF Workbench (refer appendix E).
demo_my_ap	Protect your Win-GRAF program and avoid being embezzled
demo_passwd	(refer section 11.12).
demo_XV310	
demo_XV308_1	
demo_XV308_2	
demo_XV308_3	
demo_XV116	Use an XV-Board in the WP-5xx8-CE7 (refer section 5.1.6 ~ 5.1.12).
demo_XV111	
demo_XV110	
demo_XV107	
demo_vb01	
demo_vb02	Use VB.net 2008 programs to read/write Win-GRAF variables
demo_vb03	(refer Chapter 13).
demo_vb04	
demo_PID_simple	The PID and regulator applications.
demo_user_C	Develop your own Functions and Function Blocks (refer Chapter 18).

File Name	Description				
demo_rdn_1					
demo_rdn_2	Redundant System (refer Chapter 16).				
demo_rdn_3					
demo_ET7060	Connect the remote ET-7060 Ethernet I/O module (refer section 5.2.2).				
demo_ET7018z	Connect the remote ET-7018Z Ethernet I/O module (refer section 5.2.3).				
DEMO_DL_100T485	Connect the remote DL-100T485 module to measure temperature and humidity (refer section 8.2.7).				
demo_schedule	Schedule control (refer Chapter 17).				
demo_d_7065	Connect the remote I-7065 I/O module (refer section 8.2.1).				
demo_d_7018z	Connect the remote I-7018Z I/O module (refer section 8.2.2).				
DEMO_D_7083	Connect the remote I-7083 I/O module (refer section 8.2.3).				
demo_8088w	PWM Output (refer section 4.11).				
DEMO_D_87084_FR	Connect the remote I-87084W module to measure frequency (refer section 8.2.4).				
DEMO_D_87084_C4	Connect the remote I-87084W module to measure Counter				
DEMO_D_87084_C8	(refer section 8.2.5, 8.2.6 and 4.9).				
demo_SMS	Use the 2G/3G GSM Modem to send/receive the text message (refer Chapter 21).				
demo_send_file	Send a file via Ethernet or 3G wireless network from a PAC to a remote PC (refer Chapter 20).				
demo_3G	3G wireless communication.				

# 12.2 Timer Operations

### 12.2.1 Start, Stop and Reset the Timer

Refer <u>P12-1</u> to open the project ("demo\_tmr1.zip"), and can view/add variables in the variable area.

### ST Program:

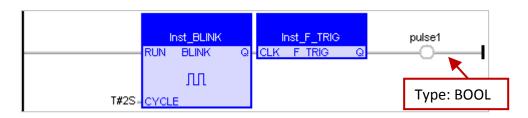
```
(* Declare "START_tmr", "STOP_tmr", "RESET_tmr", "LED1", "LED2" as BOOL
  Declare "TMR1" as TIME *)
(* Set START_tmr as TRUE to start ticking Timer TMR1 *)
IF START tmr THEN
 START tmr := FALSE;
 TSTART (TMR1);
END IF;
(* Set STOP_tmr as TRUE to stop ticking Timer TMR1 *)
IF STOP tmr THEN
   STOP tmr := FALSE;
   TSTOP (TMR1);
 END IF;
(* Set RESET tmr as TRUE to reset TMR1 to a value T#0s *)
IF RESET tmr THEN
   RESET tmr := FALSE;
   TMR1 := T#0s;
 END IF;
(* Let LED1, LED2 ON during TMR1 = 3 ~ 10 second *)
 LED1 := FALSE;
LED2 := FALSE ;
 IF (TMR1 >= T#3s) and (TMR1 <= T#10s) THEN
     LED1 := TRUE;
     LED2 := TRUE;
 END_IF;
(* Reset TMR1 as 0 when reachs 15 second *)
 IF TMR1 >= T#15s THEN
     TMR1 := T#0s;
 END IF;
```

## 12.2.2 Periodic Operations

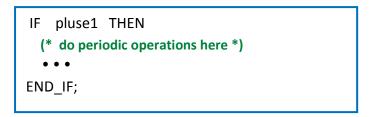
Refer <u>P12-1</u> to open the project ("demo\_tmr2.zip"), and can view/add variables in the variable area.

Function "BLINK" plus Function Block "F\_TRIG" can produce a Pluse TRUE at regular intervals, so it can be applied in the periodic operations.

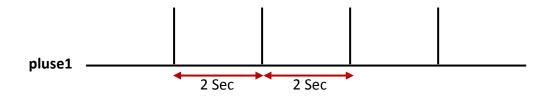
### LD Program:



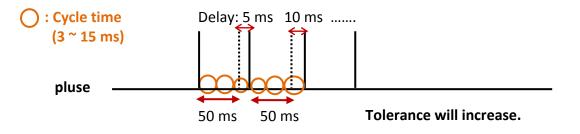
### ST Program:



The "BLINK" and "F\_TRIG" above will produce a Pluse TRUE every two seconds, but this method has a drawback:



If the interval time of the period is shorter (e.g., 100 ms per period or less; or the PAC Cycle Time is larger, such as 20 ~ 50 ms, generally 3 ~ 15 ms), then the period operations will be inaccurate. For example, to do a periodic operation every 50 ms, as compared to 250 ms or 2 seconds, the interval time of 50 ms is very close to the PAC Cycle Time, if using Function Blocks **"BLink" plus "F\_TRIG"**, it is easy to accumulate the output delay time, therefore the final operation time will become inaccurate.



To improve the situation above, the following coding method will be more accurate: (Tolerance will not increase.)

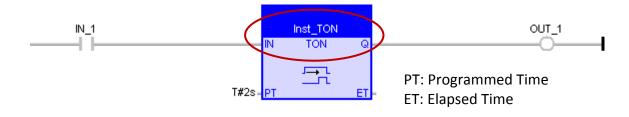
Refer <u>P12-1</u> to open the project ("demo\_tmr3.zip"), and can view/add variables in the variable area.

### ST Program:

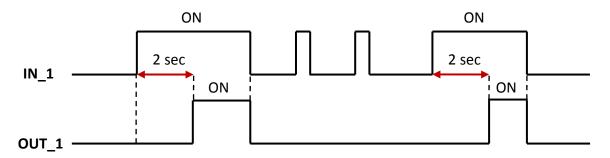
```
(* Declare "INIT" as BOOL and has initial value TRUE
 Declare "TMR1", "TMR1_next" as TIME *)
IF INIT THEN
  INIT := FALSE;
  TMR1 := T#0s;
  TMR1_next := TMR1 + T#50 ms;
  TSTART (TMR1);
END_IF;
                                            When the timer reach T#23h59m59s999ms,
IF TMR1 >= TMR1_next THEN
                                            the value will overflow.
  IF TMR1 > T#10h THEN
                                            Therefore, please reset it automatically to
     TMR1 := T#0s;
                                            "0" after 10 or 18 hours.
     TMR1_next := T#0s;
  END_IF;
  TMR1_next := TMR1_next + T#50 ms;
  (* Do periodic operations here *)
  • • •
END_IF;
```

# 12.2.3 Detect the Steady ON or Steady OFF Signal

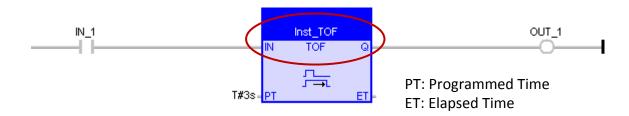
"TON" Function Block can detect the steady "ON" signal. (Keeps "ON" for a minimum period of time.)



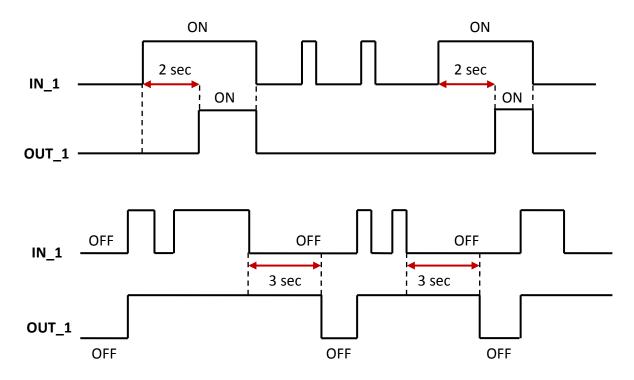
As the picture above, this function can detect the steady "**ON**" signal that keep at least 2 seconds.



"TOF" Function Block can detect the steady "OFF" signal. (Keeps "OFF" for a minimum period of time.) that can keep "OFF" for a period of time.



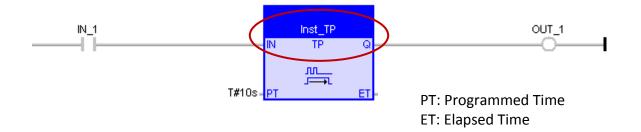
As the picture above, this function can detect the steady "OFF" signal that keep at least 3 seconds.



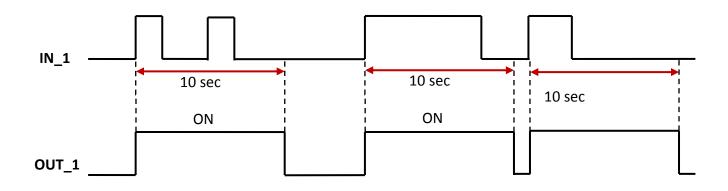
Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 12-7

# 12.2.4 Keep Outputting ON for Some Time after Triggering

"TP" Function Block can keep outputting "ON" for some time after triggering (e.g., from OFF to ON).



As the picture above, after triggering, it can keep outputting **"ON"** signal for 10 seconds.



# **12.3** Operations of Serial Port Communication

Users can directly operate the serial port (e.g., RS-232, RS-485 or RS-422 Port) to achieve some specific communication protocol. The following Functions can be used to directly operate the serial port.

Functions	Description
COM_open	Open a serial COM port.
COM_close	Close a serial COM port.
COM_clear	Clear the input buffer of a serial COM port.
COM_test	Test if any data received in the input buffer of a serial COMM port.
COM_send	Command a serial COM port to send 1~500 bytes.
COM_send_str	Command a serial COM port to send a String.
COM_recv	Receive bytes from the input buffer of a serial COM port and save them in a byte array.
COM_status	Get the current status of a serial COM port.

Please refer <u>Section 1.2.3</u> to open the Library Manager and find the detail Function description.

	‰ Library Manager - ICP DAS - XP-WP-YP 📃 🔲	×
	File <u>Wizard H</u> elp	-
$\langle$	Function and FBs //Os Profiles AS-i Types	
	<pre>@COM_clear (* Clear the input buffer of a serial COM port *)</pre>	^
	COM_close (* Close a serial COM port *)	1
	COM_open (* Open a serial COM port. For ex., TMP_BOO := COM_open(2, 9600,N,8,1'); *)	
	<b>COM_recv</b> (* Receive bytes form the input buffer of a serial COM port and save them in a byte array *)	
	<b>COM_send</b> (* Command a serial COM port to send 1 ~ 500 bytes *)	
	COM_send_str (* Command a serial COM port to send a string *)	
	COM_test (* Test if any data received in the input buffer of a serial COMM. port *)	~
	Parameters Description	
	COM_clear : Clear the input buffer of a serial COM port .	10
	*** Function	
	*** Input parameters	
	Port : DINT : serial COMM. port number to open .	
	can be 1 to 37 (depends on PAC, means COM1 to COM37) .	
	*** Output parameters	
	Q : BOOL :	
	TRUE : clear successfully.	
	FALSE : something wrong (for example, port number is incorrect).	
	Jun.18,2014	
	ICP DAS , Taiwan	

### 12.3.1 Send a String by the COM Port

Refer <u>P12-1</u> to open the project ("demo\_com\_port1.zip"), and view/add variables in the variable area.

```
ST Program: This program can send a String every 2 seconds by the PAC COM1 (parameters: `9600,N,8,1') (e.g., < CNT1 = 1 > or < CNT1 = 25 >).
```

```
(* Operations in the first PAC cycle *)
if INIT then
INIT := FALSE; (* No more first cycle *)
CNT1 := 0;
TMR1 := T#0s;
TMR1_next := TMR1 + T#2s;
(* start ticking TMR1 *)
tStart(TMR1);
end if;
```

```
Declare "INIT" as BOOL and has initial value TRUE;

"Port_OK" as BOOL ;

"CNT1" as DINT ;

"TMR1", "TMR1_next" as TIME

"Port_number" as DINT and has initial value "1"
```

```
(* if the status of COM port becomes FALSE(not open), open it *)
if COM_Status(Port_number) = FALSE then
  (* open a serial COM port *)
  Port_OK := COM_open(Port_number, '9600,N,8,1' );
```

```
end_if ;
```

```
(* when time reached , ... *)
if TMR1 >= TMR1 next then
```

```
(* to prevent TMR1 overflow (means reach T#23h59m59s999ms) *)
```

```
if TMR1 > T#10h then
  TMR1 := T#0s;
  TMR1_next := T#0s;
end_if;
```

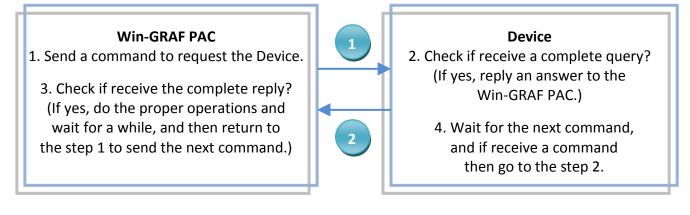
```
(* Set new TMR1_next *)
TMR1_next := TMR1_next + T#2s ;
```

```
(* Send a string from COM port *)
COM_send_str( Port_number, '<CNT1=' + Any_to_STRING(CNT1) + '>' );
```

```
(* reset CNT1 when reach 100 *)
CNT1 := CNT1 + 1 ;
if CNT1 >= 100 then
    CNT1 := 0 ;
end_if ;
end_if;
```

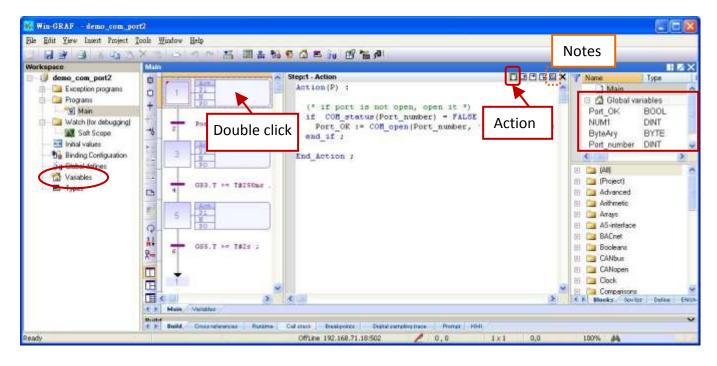
# 12.3.2 Request/Answer the Device by the COM Port

If an application needs to use RS-232/485/422 Port to get the data from other devices, the steps are as the following request and answer:



Refer <u>P12-1</u> to open the project ("demo\_com\_port2.zip"), and view/add variables in the variable area.

**Note:** Double click "Action" will first open a "Notes" window, then click "Action" icon to see the code.

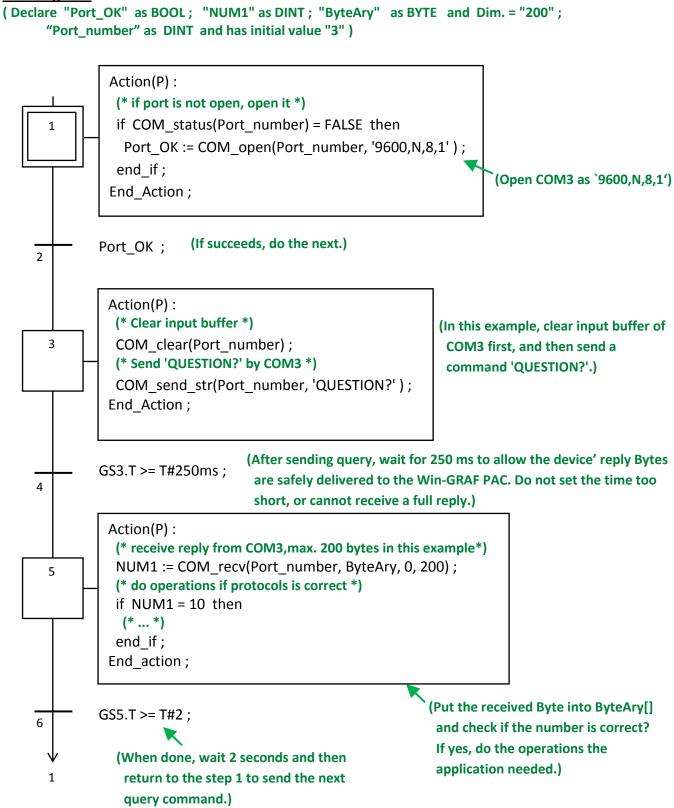


You can click the "Variables" tag to open the Variables window.

🕺 Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description	
😑 🚮 Global va	riables								
Port_OK NUM1 ByteAry Port_number	BOOL DINT BYTE DINT	[0199]			3			Byte Array, DIM=20 init as 3	30
<				196 - M.S.					>

In this example, the Win-GRAF PAC sends a string 'QUESTION?' by COM3 to the device, and then wait for the reply and does operations. After the operations, waits 2 seconds, and then sends the same command 'QUESTION?', and repeated.

### SFC Program:



# 12.3.3 Wait for Data Coming from the Remote Device to the COM Port

This way is common in the general store or supermarket, such as using the barcode readers. After reading the barcode of the product, it will send the barcode data to the Win-GRAF PAC's COM Port (RS-232/485/422), and need not to reply any messages.

Win-GRAF PAC 2. If receives the complete data from device, do operations. 3. When done, wait for the next data coming. 1

**Device** 1. After triggering, send data to the Win-GRAF PAC

Refer <u>P12-1</u> to open the project ("demo\_com\_port3.zip"), and view/add variables in the variable area.

### ST Program:

(\* operations in first PAC cycle \*)
if INIT then
INIT := FALSE;
T1 := T#0s;
STEP1 := 0;
end\_if;

### (\* if port is not open, open it \*)

```
if COM_status(Port_number) = FALSE then
  Port_OK := COM_open( Port_number , '9600,N,8,1' );
end_if;
```

### (\* If open port fail, exit this ST program \*)

```
if Port_OK = FALSE then
return ;
end_if ;
```

CASE STEP1 OF

### (\* if there is at least 1 byte coming \*)

```
0:
```

```
if COM_test(Port_number) then
STEP1 := 1;
T1 := T#0s;
```

```
Tstart(T1);
end if;
```

STEP1 = 0, means waiting, and will test if COM3 has data? If returns TRUE, means COM3 has data. Then set STEP1 to "1", T1 to "0" and start timing.

```
"Port_number" as DINT and has initial value "3".
```

Declare "INIT" as BOOL and has initial value TRUE;

"ByteAry" as BYTE and Dim. = "200";

"Port\_OK" as BOOL ;

"T1" as TIME;

"STEP1", "NUM1" as DINT;

```
(* wait 250 ms, then receive all bytes form COM port *)
```

```
1:

if T1 >= T#250ms then

Tstop(T1);

T1 := T#0s;

STEP1 := 0;

(* receive max. 200 bytes *)

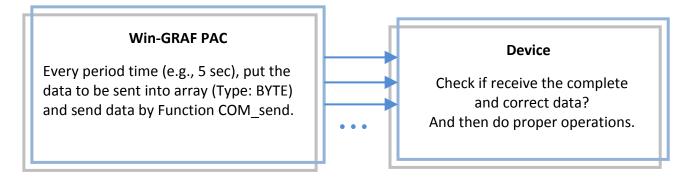
NUM1 := COM_recv(Port_number, ByteAry, 0, 200);
```

```
(* do proper operations if protocol is correct ,
    here assume correct protocols has 25 bytes in this example*)
if NUM1 = 25 then
    (* ... *)
end_if;
When receive data, check if data is correct?
If yes, do the operations the application needed.
end_if;
```

END\_CASE ;

# 12.3.4 Report Data Periodically to the Remote Device by the COM Port

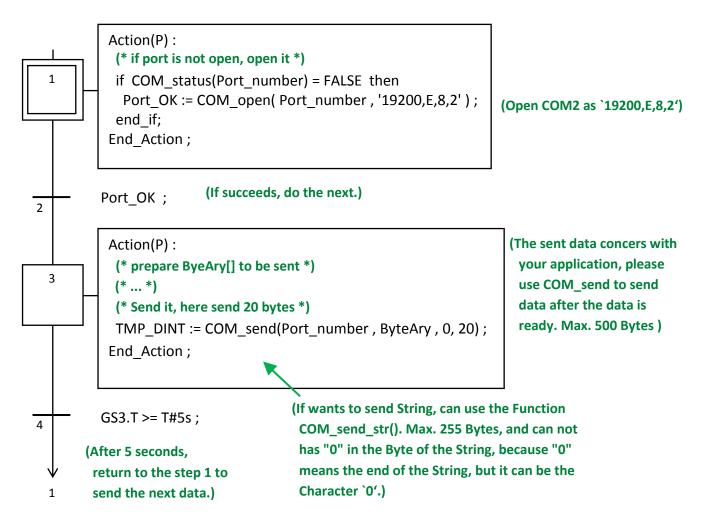
If wants to periodically report data to other devices by RS-232/485/422 Port, do as follows.



Refer <u>P12-1</u> to open the project ("demo\_com\_port4.zip"), and view/add variables in the variable area.

**<u>SFC Program:</u>** (Refer <u>Section 12.3.2</u> to open the "Action" window.)

```
(Declare "Port_OK" as BOOL; "TMP_DINT" as DINT; "ByteAry" as BYTE and Dim. = 100;
"Port_number" as DINT and has an initial value "2".)
```



# 12.4 Read/Write Data from/to a File on the PAC

The Win-GRAF Workbench provides the following Functions to enable sequential read/write operations in disk files of the Win-GRAF PAC.

Functions	Descriptions					
searching key wor	enu bar "Help" > "Topics" and type the d "File" to see more detail information "File Management functions".					
F_ROPEN	Open/Create a file for reading.					
	Open/Create a file for writing.					
F_WOPEN	If the file doesn't exist, it will be created automatically. If the file exists, all its content will be removed.					
F_AOPEN	Create or open a file in append mode.					
F_CLOSE	Close an open file.					
F_EOF	Test if the end of file is reached in a file open for read.					
FA_READ	Read a DINT integer from a binary file.					
FA_WRITE	Write a DINT integer to a binary file.					
FM_READ	Read a STRING value from a text file					
FM_WRITE	Write a STRING value to a text file.					
FB_READ	Read binary data from a file.					
FB_WRITE	Write binary data to a file.					
F_EXIST	Test if a file exists.					
F_GETSIZE	Get the size of a file.					
F_COPY	Copy a file.					
F_DELETE	Remove a file.					
F_RENAME	Rename a file.					
Refer Section 1.2	2.3 to find the detailed descriptions for the following Functions.					
F_dir	Create a directory.					
F_cp_dir	Copy all files in a directory to another directory. (Include subdirectories and files). (Note1)					
F_del_dir	Delete a directory and all files inside it. (Include subdirectories and files). (Note1)					

# **Note:** The Win-GRAF PAC of ICP DAS does not support Functions "F\_SAVERETAIN" and "F\_LOADERETAIN".

- **Note1:** Since the following PAC drivers support the "**F\_cp\_dir**" and the "**F\_del\_dir**" functions for operating in sub-directories.
  - WP-8xx8: v1.04, VP-x2x8-CE7:v1.01, XP-8xx8-CE6:v1.02, WP-5xx8-CE7: Released day.

### 12.4.1 Write Data to a File on the PAC

Refer <u>P12-1</u> to open the project ("demo\_file1.zip"), and can view/add variables in the variable area.

**<u>ST Program</u>**: This program can be used to write 10 "REAL" values to a file on the PAC.

(\* This "demo\_file1" project will save 10 REAL value to a file in the \System\_Disk\Real\_data1.txt .

### File Format : Each row contains one REAL value and ends with <CR><LF> characters. Like:

1.08 2.786 38.45 41.5 59.875 60.76 71.23 80.5 99.8 100.7

(\* Variables declaration:

```
Write_File: BOOLTmp_string: String, len=255File_ID: DINTREAL_val[0..9] : REALii: DINTFile_Status: String, len=128 *)
```

\*)

Because the size of \System\_Disk\ is small, recommend you may change the directory to below (Depends on your application):

WinPAC, ViewPAC Series: \Micro\_SD\ or

XPAC Series: \System\_Disk2\

(\* Set Write\_File as TRUE to write data to the file \*) if Write\_File then

```
Write_File := FALSE ;
File_ID := F_Wopen( '\System_Disk\Real_data1.txt' );
```

```
if File_ID = 0 then
```

```
(* Can not open file in write mode *)
File Status := 'Can not open file in write mode !' ;
```

else

```
(* open file in write mode ok, save REAL[0] ~ [9] to file ,
            each row contains 1 REAL value and end with <CR><LF> *)
File_Status := 'Open file ok.' ;
for ii := 0 to 9 by 1 do
```

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 12-17

```
Tmp_string := Any_to_string( REAL_val[ii] ) ;
FM_write( File_ID , Tmp_string ) ;
end_for ;
(* close the file *)
F_close( File_ID ) ;
end_if ;
```

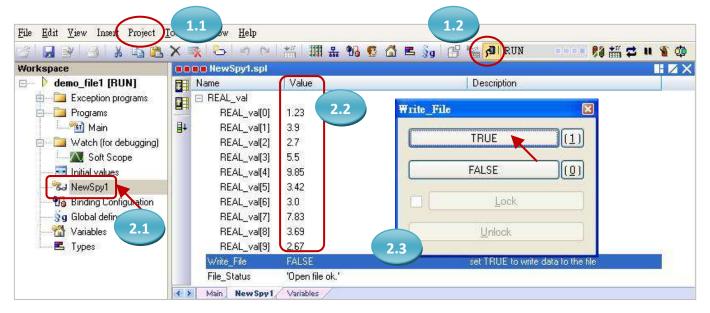
If want to save data to Integer, use the code: Tmp\_string := Any\_to\_string( **DINT\_val[ii]**); and declare Variable "DINT\_val" as DINT and Dim. at least "10" for this example.

```
end_if;
end_if;
```

### Test Program:

In this example, when the "Write\_File" is set to "TRUE", the values will be written into the file \System\_Disk\Real\_data1.txt on the PAC.

- Please set up IP configurations (Refer <u>P12-1</u>), compile and download the program to the PAC. (Click on "Project" > "Build All Projects" / "On Line", if not familiar with the operation, refer to <u>Section 2.3.4</u>, <u>Section 2.3.5</u>)
- 2. Click "NewSpy1" to open a Spy List and fill in the values to be written, and then set the "Write\_File" become "TRUE" to Write data. (If OK, "File\_Status" will show "Open file ok".)



3. On the PAC, open the file "Real\_data1.txt", can see the values filled in the step 2.

檔案(E) 編輯(E) 檢視(⊻) 移至((	<u>a)</u>						
	檔案(E)	編輯( <u>E</u> )	檢視(⊻)	格式( <u>O</u> )	工具( <u>T</u> )	े 🚅 日	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
網址(D) \System_Disk	1.23						
C Drivers	3.9						
😂 External_device_driver	2.7						
C ISAGRAF	5.5						
🗁 tools	9.85						
C Win-GRAF	3.42						
Backup_Real_1.txt	3						
GetPacName.exe	7.83						
☆ penmount_usb_touch_v20100526 ★ Real_data1.txt	3.69						
Testf1.txt	2.67						
<pre>     Real_data1.txt     testf1.txt     Working_Int.txt </pre>	5.07						

## 12.4.2 Read Data from a File on the PAC

Refer <u>P12-1</u> to open the project ("demo\_file2.zip"), and can view/add variables in the variable area.

ST Program: This program can be used to read 10 "REAL" values from a file on the PAC.

```
(* this "demo_file2" project will read 10 REAL value from a file
                       in the \System_Disk\Real_data2.txt.
     File format :
       Each row contains one REAL value and ends with <CR><LF> characters. Like:
       1.08
       2.786
       38.45
       41.5
       59.875
       60.76
       71.23
       80.5
       99.8
       100.7
*)
(*
 Variables Declaration:
 Write_File
                : BOOL
 Tmp_string
                : String, len=255
 File_ID
                : DINT
 REAL_val[0..9] : REAL
 ii
                : DINT
 File_path
                : String, len = 128, initial val = '\System_Disk\Real_data2.txt'
 File_Status
                : String, len=128
*)
                                                              Because the size of \System Disk\
                                                              is small, recommend you amy
(* Set Read_File as TRUE to read data from the file *)
                                                              change the directory to below
if Read File then
                                                              (Depends on your application):
  Read File := FALSE ;
                                                              WinPAC, ViewPAC Series:
  (* Check if file exists *)
                                                               \Micro_SD\ or
  if F exist(File path) = FALSE then
    (* file doesn't exist *)
                                                              XPAC Series:
    File_Status := 'File "' + File_path +'" does not exist !' ;
                                                                \System_Disk2\
  else
   (* file does exist , open it in read mode *)
   File_ID := F_Ropen( File_path );
   if File ID = 0 then
      (* open file in read mode fail *)
      File status := 'Can not open File "' + File path +'" !';
   else
```

```
(* open file in read mode ok, read REAL[0] ~ [9] from file ,
      each row contains 1 REAL value and end with <CR><LF> *)
   File status := 'Open File "' + File path +'" Ok .';
   for ii := 0 to 9 by 1 do
     (* test if the end of file is reached in a file open for read *)
     if F EOF(File ID) then
       (* reach the end of file, exit "for loop" *)
       exit ;
     end_if;
     (* read one row in the file as a string *)
     Tmp_string := FM_READ( File_ID ) ;
     (* convert the string to become REAL value *)
     REAL_val[ii] := Any_to_REAL( Tmp_string ) ;
   end for;
                                      If want to read Integer, use the code below:
                                      DINT_val[ii] := Any_to_DINT( Tmp_string ) ;
   (* close the file *)
                                      and declare Variable "DINT_val" as DINT and
   F_close(File_ID);
                                      Dim. at least "10" for this example..
 end_if;
end if;
```

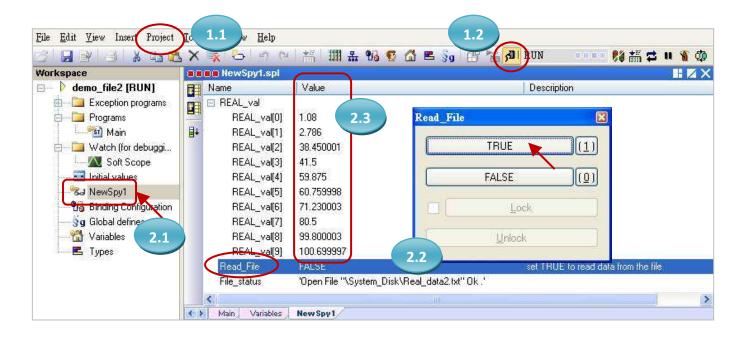
<u>Note:</u> In this example, when the "Read\_File" is set to "TRUE", it will read the file on the PAC "\System\_Disk\Real\_data2.txt", please make sure the file already exists on the PAC.

end\_if;

檔案(E) 編輯(E) 檢視(⊻) 移至(G	i)						
🗄 🔷 🎓 🏂 🔀 🔤 🗸							
網址(D) System_Disk							
C Drivers	檔案(E)	編輯( <u>E</u> )	檢視(⊻)	格式( <u>0</u> )	工具( <u>T</u> )	: 🛩 日	X 🖻 🖬 🖍
C ISAGRAF C tools Win-GRAF	1.08 2.786 38.45 41.5						
Penmount_usb_touch_v20100526 IEC61131 User and Reference Mani Backup_Real_2.txt							
<ul> <li>Backup_Real_2.txt</li> <li>Real_data2.txt</li> <li>Update_History_20120418.txt</li> <li>Working_Real.txt</li> </ul>	80.5 99.8 100.7						

### Test Program:

- Please set up IP configurations (Refer <u>P12-1</u>), compile and download the program to the PAC. (Click on "Project" > "Build All Projects" / "On Line", if not familiar with the operation, refer to <u>Section 2.3.4</u>, <u>Section 2.3.5</u>)
- 2. Click "NewSpy1" to open a Spy List and set the "Read\_File" to "TRUE" to read the data. (If OK, "File\_Status" will show "Open File "\System\_Disk\Real\_data2.txt" Ok.")



**Note:** There is one another file operation example listed in the <u>Section 6.2</u>. It handles many data in the file, you may refer it.

# 12.4.3 Data Logging

Refer <u>P12-1</u> to open this project ("demo\_DataLog.zip") that provides a simple function for data logging. This program creates a spy list that contains a String variable (write\_date), an Integer variable (int\_data) and a Real variable (float\_data). These variable values will be recorded to a file per minute and be stored on the PAC's **\System\_Disk2\**. This log file will be named according to the current date (e.g., May 06, 2015, the file name will be "2015-5-6.csv"). Moreover, this log file will be replaced per day and the existing file will be moved to "\System\_Disk2\Current month\" (e.g., May 06, 2015, the file will be moved to "\System\_Disk2\2015-5\").

# <u>Note:</u> This example project is used for the XP-8xx8-CE6 PAC. If you want to use the others PAC, simply change the file path as \Micro\_SD\ or other storage location.

The sample of the CSV file format:								
Time , int_val ,	float_val							
20:18:30, 1236,	14.56							
20:18:40, 3456,	34.56							
20:18:50, 8932,	89.32							

**Description of variables:** The user can view/set variables in the Win-GRAF "Variables" window.

Name	Data Type	Description					
Year1	DINT						
Month1	DINT						
Day1	DINT						
WeekDay1	DINT	Used to get the PAC system time in the "PAC_Time" program.					
Hour1	DINT						
Minute1	DINT						
Second1	DINT						
old_day	DINT	Used to know if the time is changed and then rename the log file in the "PAC_Time" program.					
old_hour	DINT						
log_tmr1	TIME	Timer.					
log_tmr2	TIME	Timer.					
CSV_Path	STRING	The storage path of CSV file.					
CSV_Dir	STRING	The storage folder of CSV file.					
write_date	STRING	Used to record the time when writing data to a CSV file.					
init	BOOL	Set it as TRUE to initialize. (Init value = TRUE)					
int_data	DINT	Used to record the data.					
float_data	REAL	Used to record the data.					
writecsv	BOOL	Set it as TRUE to write data.					
File_ID	DINT	Used for the "F_WOPEN" function.					
tmp_bval	BOOL	The temperature veriable					
tmp_msg	STRING	The temporary variable.					

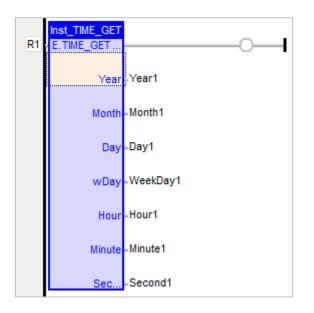
### Spy List - "My\_List":

There is a Spy list called "My\_list" in this example project. (Refer Section 11.3 for more details)

My_l	ist.spl			
	Name	Value	Description	
	write_date int_data			
∎+	float_data			

### LD Program – "PAC\_Time":

To get/set the PAC system time.



### ST Program – "Main":

To generate random values as the data of the log file (.csv) and to change the file storage path.

```
(* Copy the file to \System_Disk2\ *)
tmp_bval:=F_COPY (CSV_Path,tmp_msg);
```

```
(* Delete the original file stored at \temp\ *)
```

tmp\_bval:=F\_DELETE(CSV\_Path);

# (\* Create the CSV file and its filed name \*)

```
tmp_msg:='\Temp\'+any_to_string(year1)+'-'+any_to_string(month1)+'-'+
any_to_string(day1)+'.csv';
CSV_Path:=tmp_msg;
File_ID:= F_WOPEN(tmp_msg);
```

```
IF File_ID<>0 THEN
    tmp_msg:='Time,Int_val,Float_val';
    tmp_bval:=FM_WRITE(File_ID,tmp_msg);
    tmp_bval:=F_CLOSE(File_ID);
END_IF;
```

```
old_day:=day1;
```

```
END_IF;
```

## (\* Create the folder every month, or when the PAC boot up. \*)

```
if old_month<>month1 then
```

```
CSV_Dir:=ANY_TO_STRING(year1)+ANY_TO_STRING(month1);
```

tmp\_msg:='\System\_Disk2\'+CSV\_Dir;

F\_DIR(tmp\_msg);

old\_month:=month1;

end\_if;

Because the size of \System\_Disk\ is small, we recommend that you save the log file at \Micro\_SD\ (for WinPAC, ViewPAC series) or \System\_Disk2\ (for XPAC series).

## (\* The function for writing data to a CSV file \*)

```
IF log_tmr1>=log_tmr2 THEN
```

log\_tmr1:=t#0s;

```
(* Generate random values for the "Int_val" and the "Float_val" field *)
```

int\_data:=rand(1000);

float\_data:=ANY\_TO\_REAL(int\_data)/100;

```
(* The time for each data logging *)
```

```
write_date:=ANY_TO_STRING(hour1)+':'+ ANY_TO_STRING (Minute1)+':'+
```

ANY\_TO\_STRING (Second1);

```
(* Delete the previous file *)
```

```
tmp_msg:='\System_Disk2\'+CSV_Dir+'\'+ANY_TO_STRING(Year1) +'-'+
ANY_TO_STRING(month1)+'-'+ANY_TO_STRING(Day1)+'.csv';
```

```
(* Delete the file stored at \System_Disk2\ *)
```

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

```
(* Trigging it to write data to a file *)
```

writecsv:=true;

END\_IF;

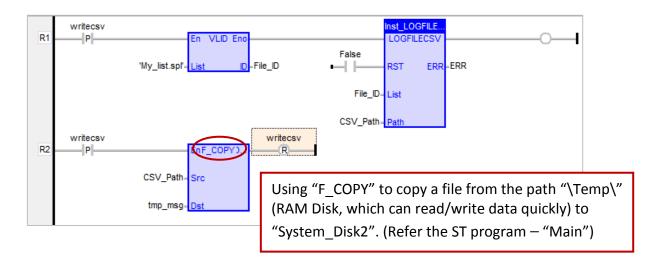
### LD Program – "WriteFile"

Set the "writecsv" as "TRUE" to write one data to the CSV file.

The user can click the following function or function block, and then press "F1" to view the descriptions. VLID: Get the identifier of an embedded list of variables (i.e., Spy List - My\_list.spl).

LogFileCSV: Generate a log file in CSV format for a list of variables.

F\_COPY: Copy a file.



# Chapter 13 VB.net 2008 Program Running in WP-8xx8 Access to Win-GRAF Variables

This chapter lists the procedure for creating the first demo program by Visual Studio .NET 2008 development tool. There are some sample programs in the WP-8xx8 CD-ROM.

### VB .NET example:

CD-ROM : \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\ demo\_vb01 : Digital I/O demo with one I-87055W in slot 0 of the WP-8xx8. demo\_vb02 : Analog I/O demo with one I-87024W in slot 1, one I-8017HW in slot 2. demo\_vb03 : Read/Write Win-GRAF internal integers, timers & real variables. (No I/O) demo\_vb04 : Read/Write Win-GRAF internal String variables. (No I/O)

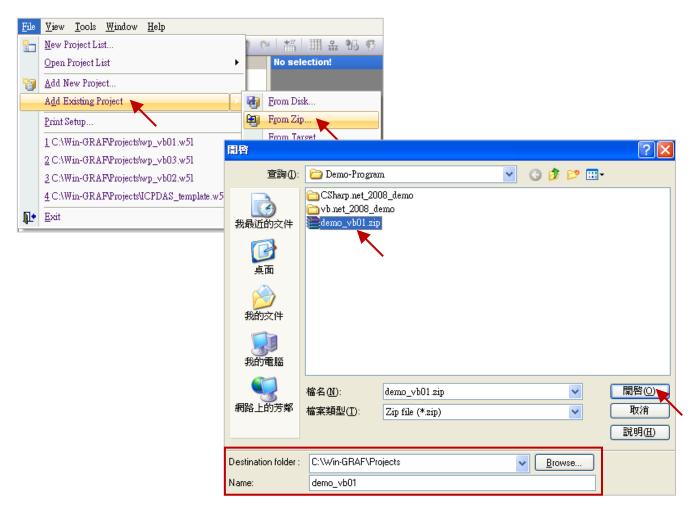
### Win-GRAF example:

```
CD-ROM : \napdos\Win-GRAF\demo-project\
"demo_vb01.zip", "demo_vb02.zip", "demo_vb03.zip", "demo_vb04.zip"
```

# 13.1 Add an Existing Win-GRAF Project from a ZIP

Please follow these steps to restore the Win-GRAF project.

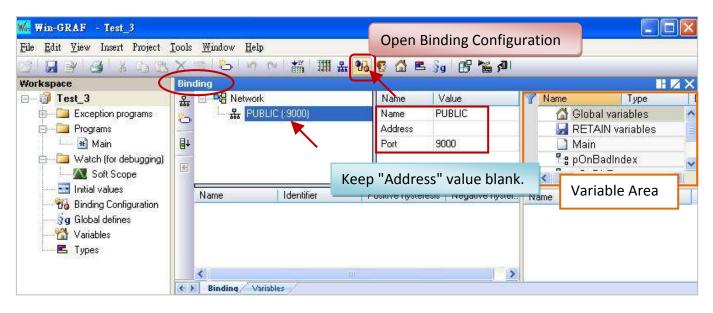
First Click "File" -> "Add Existing Project" -> "FromZip...". Then choose the Win-GRAF project zip file which you would like to restore. After restoring the project, you have to build the project, and then download it to the PAC.



# 13.2 Publishing the Win-GRAF Variable for .NET and Soft-GRAF HMI

If users wish to use .NET program to Read/Write the Win-GRAF variables. Except for String variable all of the variables need to use the "Open Binding Configuration" function to set an address. The following demonstrates how to publish Win-GRAF variable:

- 1. Click "Open Binding Configuration" on the toolbar to open the "Binding" setup window.
- 2. Click "PUBLIC (:9000)". Keep the "Address" value is blank and "Port" value is fixed to 9000.



3. Before publishing these variables, make sure you have declared them in the Variables Area. Click "Global variables" and press the "Ins" key to insert a new variable. The following table demonstrates variables using in the "Test\_3" project and you can declare them according to the needs of your application.

Variable name	Туре
Public_BOOL	BOOL
Public_INT	INT
Public_DINT	DINT
Public_WORD	WORD
Public_REAL	REAL

	I	XN
🍸 Name	Туре	D
😑 🚮 Global v	/ariables	^
Public_BOO	L BOOL	
Public_INT	INT	
Public_DINT	DINT	
Public_WOR	D WORD	
Dublic DEAL	DEAL	
Public_REAL		
📄 Main		
	Enter key	

4. As the figure below, click on "PUBLIC(:9000)" and drag all the needed variables to the "Name" area. The "Identifier" will generate an address number automatically. If any other VB or .NET program wants to use these public variables, it must set to the same address number (ID).

### NOTE:

The "PUBLIC" allows to use up to 8192 variables, and the "Identifier" number JUST can be 1 to 8192.

E PB Network		Name	Value	🝸 Name		Туре	
🗄 🚠 PUBI	.IC (:9000)	Name	PUBLIC	🗉 🚮 GI	obal variat	les	1
CALORE -		Address	8	Public_	BOOL	BOOL	
	•	Port	9000	Public_ Public_ Public_ Public	DINT WORD	INT DINT WORD REAL	
Name	Identifier		ositive hysteresi	s K			2
Public_BOOL	1			Name	Valu	9	_
Public_INT	2						
Public_DINT	3	-					
Public_WORD	4						
Public_REAL	5						
				>			

The following procedure will show you how to use the "pub\_string" function to publish the Win-GRAF String variable in the ST program.

### Syntax:

Pub\_string(Address, String\_val);

Address: The public address number, and its range can be 1 to 1024 String\_val: The name of String variable.

### Variables description:

Name	Туре	Description
Init	BOOL	Set the initial value as TRUE.
Tmp_val	BOOL	TRUE: Binding succeeds. FALSE: Binding fails.
msg1	STRING, Length is 100	
msg2	STRING, Length is 32	String variable for demo purpose. NOTE: The String length could be 1 to 255.
msg3	STRING, Length is 60	

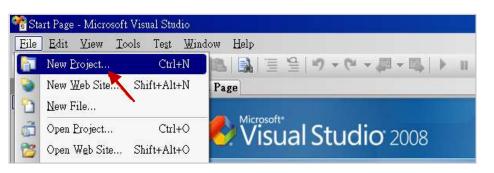
### ST program:

If init then
Init := false;
(\*add address 1 for share string val \*)
Tmp\_val := pub\_string(1,msg1);
(\*add address 2 for share string val \*)
Tmp\_val := pub\_string(2,msg2);
(\*add address 3 for share string val \*)
Tmp\_val := pub\_string(3,msg3);
End\_if;

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 13-3

# 13.3 Create a new VB.NET project

1. First, run Microsoft Visual Studio .NET 2008 software, and then choose "File" > "New Project".



2. Click "Smart Device" on the left, and then select ".NET Framework 3.5" and "Smart Device Project". Entering a proper project name and click "OK".

Project types:		<u>T</u> emplates:	.NET Framework 3.5	
<ul> <li>Visual Basic</li> <li>Windows</li> <li>Web</li> <li>Smart Devi</li> <li>Office</li> <li>Database</li> <li>Reporting</li> <li>Test</li> <li>WCF</li> <li>WCF</li> <li>Other Language</li> <li>Other Project T</li> <li>Test Projects</li> </ul>	es	Visual Studio installed templates         Smart Device         Project         My Templates         Search Online         Templates		2
A project for Smart	Device applications.	Choose target platform, Framework version, and templa	te in the next dialog box.	
<u>N</u> ame:	project1			
			OK	Cancel

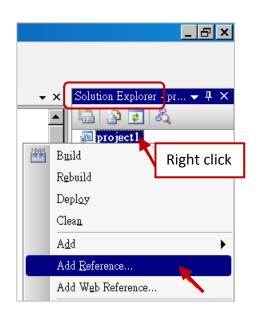
3. Select the "Device Application" and "Windows CE" and ".NET Compact Framework Version 3.5", then click "OK".

Target platform: .NET Compact Framework version: Templates:	Windows CE .NET Compact Framework .NET Compact Framework	Version 2.0	× •
Device Class Library O	NET Compact Framework	Version 3.5 Empty Project	Description: A project for creating a .NET Compact Framework 3.5 forms application for Windows CE Platform
Download additonal emulator images	and smart device SDKs		OK Cancel

# 13.3.1 Add Project Reference

The "UserShareNet.dll" library contains all functions of data exchange with Win-GRAF variables. Before you use the "UserShareNet" keyword in the program, you must add the "UserShareNet.dll" into the reference list of your project.

- 1. Copy the "UserShareNet.DLL" from Win-PAC's shipment CD (\napdos\Win-GRAF\WP-8xx8\ vb.net\_2008\_demo\wp\_vb01\vb01\) to your project folder (e.g., "C:\project1\")
- Right click on the project name (e.g., "project1") in the "Solution Explorer" window, and then select "Add Reference ...".



3. Click the "Browse" tab and select the "UserShareNet.dll" from your project location.

NET Projects Browse Recent 搜尋位置①: 🗁 project 🗸 😋 🎓 📂 🖽 -			
in bin in in My Project in obj	The current project is located in "C:\project1"	".	
S UserShareNet.dll			
檔案名稱(11):	▼		
檔案類型(I): Componen	t Files (*.dll;*.tlb;*.olb;*.ocx;*.exe)		
	OK Cancel		

<u>Note:</u> You may copy the "UserShareNet.dll" from the CD-ROM to your current project path first. Then add it to the project reference. 4. When "UserShareNet.dll" is added, please double click on "My Project" to check if the "UserShareNet.dll" is well added.

	Solution Explorer Solution Explorer Project Solution Explorer Project Form 1. vb Double click
Application	Configuration: N/A  Platform: N/A
Compile	
Debug	References: Unused References Reference Paths
References	Reference Name         Type         Ver         Copy Local         Path           System         NET         2.0.0.0         False         C:\Program Files\Microsoft.NET\SDK\CompactFramework\w2.0\W
Resources	System Data         NET         2.0.0.0         False         C.'Program Files!Microsoft.NET/SDK'/CompactFramework/w2.0\W           System Drawing         NET         2.0.0.0         False         C.'Program Files!Microsoft.NET/SDK'/CompactFramework/w2.0\W           System.Windows.For         NET         2.0.0.0         False         C.'Program Files!Microsoft.NET/SDK'/CompactFramework/w2.0\W
Signing	System Xml NET 2000 False C. Program Files/Microsoft NET/SDK/CompactFramework/v20/W UserShareNet .NET 1.0.1.0 True C. Documents and Settings/user供面/wingrafdoc/Wapdos/Win-GR
Devices	

5. Right-click on the "Form1.vb" and select "View Code" from the pop-up. Move cursor to top and insert the "Option Explicit On" and "Imports UserShareNet" in the first two statements.

Solution Explo	rer 👻	Ψ×.	Form1.vb
			🏤 Form1
project1 🔤 My P			Option Explicit On
Form			Imports UserShareNet
	<u> O</u> pen		
Right	Ope <u>n</u> With		∃Public Class Form1
Click	🔄 View <u>C</u> ode 🔖		L End Class
-			

Then you can design all required objects and actions inside your VB Forms. (Refer the <u>Chapter 13.5</u> for more information about using functions in the "UserShareNet.dll".)

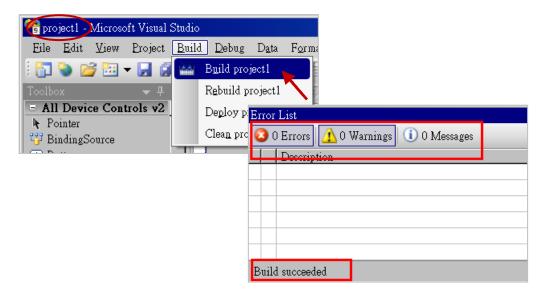
# 13.4 Compiling the Application

When you have finished writing a program, you can build (compile) an application by the following steps.

1. Remember to save at any time for safety.



2. Then compile (Build) the project. The result is listed in the "Error List" windows at the bottom.



3. You can find the execution file in

<Your VB.net Project folder> \bin\Release\ <project\_name>.exe

Please copy this execution file to the WP-8xx8's \System\_Disk\Win-GRAF\ path to run it.

#### Note:

The user may copy the VB.net execution file to another path to run it, but there should contain at least two DLL files with it or it cannot run correctly.

For ex, the project1.exe can run in the \Micro\_SD\ folder if there are three files in it. The "project1.exe", "UserShareNet.dll" and "Quicker.dll". (The "UserShareNet.dll" and "Quicker.dll" can be copied from the Win-GRAF PAC's "\System\_disk\Win-GRAF\" path)

# 13.5 UserShareNet.dll

This section we will focus on the description of the application example of UserShareNet.dll functions. There are some functions that can be used to read/write data from/to the Win-GRAF soft-logic. The functions of UserShareNet.dll can be divided into as listed below:

- 1. R/W Boolean
- 2. R/W 8-bit Integer
- 3. R/W 16-bit Integer
- 4. R/W 32-bit Integer
- 5. R/W 64-bit Integer
- 6. R/W 32-bit Float
- 7. R/W 64-bit Float
- 8. R/W String

**※** Refer the "<u>Appendix A</u>" to get familiar with the definition of Win-GRAF variables.

### 13.5.1 R/W Boolean Functions

### Set\_BOOL

### Description:

This function is to set a value to a Win-GRAF Boolean variable.

### Syntax:

UserShare.Set\_BOOL (iUserAddress As System.UInt16, ByVal iStatus As byte) as Byte

#### Parameter:

iUserAddress :Address of the Variable (1 to 8192)iStatus :Set the status. For instance, iStatus = 1 for True, iStatus = 0 for False

### Example:

'Set the Win-GRAF BOOL variable with address 1 to True. UserShare.Set BOOL(Convert.ToUInt16(1), 1)

### Demo program:

CD-ROM: \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb01



### **Description:**

This function is to get the value from a Win-GRAF BOOL variable.

#### Syntax:

UserShare.Get\_BOOL ( iUserAddress As System.UInt16 , ByRef iStatus As byte)

### Parameter:

iUserAddress :Address of the Variable (1 to 8192)iStatus :Get the variable value , iStatus = 1 for True, iStatus = 0 for False

### Example:

'Get the value of Win-GRAF BOOL variable with address 1. Dim iStatus As Byte UserShare.Get\_BOOL(Convert.ToUInt16(1), iStatus)

### **Demo Program:**

CD-ROM: \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb01

### 13.5.2 Integer R/W Functions

Set\_SINT Set\_INT Set\_DINT Set\_LINT

#### **Description:**

These functions are to set 8-bit Integer, 16-bit Integer, 32-bit integer & 64-bit Integer value to Win-GRAF integer variables.

#### Syntax:

UserShare.Set\_SINT (ByVal *iUserAddress* As System.UInt16, ByVal *iStatus* As SByte) As <u>Byte</u> UserShare.Set\_INT (ByVal *iUserAddress* As System.UInt16, ByVal *iStatus* As Short) As <u>Byte</u> UserShare.Set\_DINT (ByVal *iUserAddress* As System.UInt16, ByVal *iStatus* As <u>Integer</u>) As <u>Byte</u> UserShare.Set\_LINT (ByVal *iUserAddress* As System.UInt16, ByVal *iStatus* As <u>long</u>) As <u>Byte</u>

#### Parameter:

iUserAddress : Address of Variable. (1 to 8192)

iStatus : the value of 8-bit Integer, 16-bit Integer, 32-bit Integer or 64-bit Integer.

#### **Example:**

'Set a 32-bit integer value "1234567" to the Win-GRAF DINT variable with address "1". UserShare.Set\_DINT(Convert.ToUInt16(1), Convert.ToInt32(1234567))

'Set a 16-bit integer value "-1234" to the Win-GRAF INT variable with address "2". UserShare.Set\_INT(Convert.ToUInt16(3), Convert.ToInt16(-1234))

'Set a 64-bit Integer value "123456789012345" to the Win-GRAF LINT variable with address "3". UserShare.Set\_LINT(Convert.ToUInt16(3), Convert.ToInt64(123456789012345))

'Set a 8-bit Integer value "125" to the Win-GRAF SINT variable with address "4". UserShare.Set\_SINT(Convert.ToUInt16(3), Convert.ToSByte(125))

#### Demo Program:

CD-ROM:

- 1. \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb02 for R/W analog I/O
- 2. \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb03 for R/W internal long integer, Timer and Real (floating-point) values.

📕 Get\_SINT 📕 Get\_INT 📕 Get\_DINT 📕 Get\_LINT

#### **Description:**

These functions are to get 8-bit integer, 16-bit integer, 32-bit integer & 64-bit integer value from Win-GRAF integer variables.

#### Syntax:

UserShare. Get\_SINT (ByVal *iUserAddress* As System.UInt16 , ByRef *iStatus* As <u>SByte</u>) As <u>Byte</u> UserShare. Get\_INT (ByVal *iUserAddress* As System.UInt16 , ByRef *iStatus* As <u>Short</u>) As <u>Byte</u> UserShare.Get\_DINT (ByVal *iUserAddress* As System.UInt16 , ByRef *iStatus* As <u>Integer</u>) As <u>Byte</u> UserShare. Get\_LINT (ByVal *iUserAddress* As System.UInt16 , ByRef *iStatus* As long) As <u>Byte</u>

#### Parameter:

iUserAddress :Address of Variable (1 to 8192)iStatus :Get the 8-bit integer, 16-bit integer, 32bit-integer or 64-bit integer value.

#### Example:

Dim Dlong\_val As Int64 Dim short\_val As Int16 Dim long\_val As Int32 Dim Sbyte\_val as byte

'Get 64-bit integer value from the Win-GRAF LINT variable with address "7". UserShare.Get\_LINT(Convert.ToUInt16(7), Dlong\_val)

'Get 32-bit integer value from the Win-GRAF DINT variable with address "8". UserShare.Get\_DINT(Convert.ToUInt16(8), long\_val)

'Get 16-bit integer value from the Win-GRAF INT variable with address "9". UserShare.Get\_INT(Convert.ToUInt16(9), short\_val)

'Get 8-bit integer value from the Win-GRAF SINT variable with address "10". UserShare.Get\_SINT(Convert.ToUInt16(9), sbyte\_val)

#### Demo program:

- R/W analog I/O : \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo \demo\_vb02
- R/W internal long integer, Timer and Real (floating-point) values : \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb03

## 13.5.3 R/W Real Variable Functions

#### Get\_REAL Get\_LREAL

#### **Description:**

These functions are to get 32-bit REAL and 64-bit double from the Win-GRAF REAL/LREAL variable.

#### Syntax:

UserShare. Get\_REAL (ByVal iUserAddress As System.UInt16, ByRef iStatus As Single) As Byte

UserShare. Get\_LREAL(ByVal iUserAddress As System.UInt16 , ByRef iStatus As Double) As Byte

#### Parameter:

iUserAddress :Address of Variable (1 to 8192)iStatus :Get the 32-bit REAL or 64-bit double value.

#### Example:

Dim float\_val As Single Dim double\_val As Double

'Get 64-bit double value from the Win-GRAF LREAL variable with address "7". UserShare.Get\_LREAL(Convert.ToUInt16(7), double\_val)

'Get 32-bit REAL value from the Win-GRAF REAL variable with address "8". UserShare.Get\_REAL(Convert.ToUInt16(8), float\_val)

#### Demo program:

- R/W analog I/O: \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb02
- 2. R/W internal long integer, Timer and Real (floating-point) values : \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb03

#### Set\_REAL Set\_LREAL

#### **Description:**

These functions are to set 32-bit REAL and 64-bit double value to the Win-GRAF REAL/LREAL variable.

#### Syntax:

UserShare. Set\_REAL (ByVal iUserAddress As System.UInt16, ByVal iStatus As Single) As Byte

UserShare. Set\_LREAL(ByVal iUserAddress As System.UInt16, ByVal iStatus As Double) As Byte

#### Parameter:

iUserAddress :Address of Variable. (1 to 8192)iStatus :Set the 32-bit REAL or 64-bit double.

#### Example:

'Set a 64-bit double value "11234.234567" to the Win-GRAF LREAL variable with address "1". UserShare.Set\_LREAL(Convert.ToUInt16(7),Convert.ToDouble(11234.234567))

'Set a 32-bit REAL value "123.12" to the Win-GRAF REAL variable with address "8". UserShare.Set\_REAL(Convert.ToUInt16(8), Convert.ToSingle (123.12))

#### Demo program:

- 1. R/W analog I/O : \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb02
- R/W internal long integer, Timer and Real (floating-point) values : \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb03

## 13.5.4 R/W String Variable Functions

## Get\_STRING

#### **Description:**

This function is to get a Win-GRAF String variable.

#### Syntax:

UserShare. Get\_STRING (ByVal iUserAddress As System.UInt16, ByVal msg() As Byte) As Byte

#### **Parameter:**

iUserAddress : Address of Variable (1 to 1024)msg() :Get the string value.

#### Example:

Dim str\_val As String Dim msg() As Byte

'Get String value of the Win-GRAF String variable with address "7". UserShare.Get\_STRING(Convert.ToUInt16(7),msg ) str\_val= byte\_array\_to\_unicode(msg)

Private Function byte\_array\_to\_unicode(ByVal buf() As Byte) As String Dim tmpmsg As String If buf.Length > 255 Then Return Nothing End If tmpmsg = System.Text.Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length)

Return tmpmsg End Function

#### Demo program:

CD-ROM:

 R/W String variable: \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb04



#### **Description:**

This function is to set a String value to the Win-GRAF String variable.

#### Syntax:

UserShare. Set\_STRING (ByVal iUserAddress As System.UInt16, ByVal msg() As Byte) As Byte

#### Parameter:

iUserAddress : Address of Variable. (1 to 1024)msg() :the string value.

#### Example:

Dim str\_val As String="Hello World" Dim msg() As Byte

msg= unicode\_to\_byte\_array(str\_val)

'Set a string value "Hello World" to the Win-GRAF String variable with address "7". UserShare.Set\_STRING(Convert.ToUInt16(7),msg)

'Convert String to byte array.

Private Function unicode\_to\_byte\_array(ByVal msg As String) As Byte() Dim tmpbuf() As Byte If msg.Length > 255 Then Return Nothing End If tmpbuf = System.Text.Encoding.GetEncoding("UTF-8").GetBytes(msg) Return tmpbuf End\_Function

#### Demo program:

CD-ROM:

 R/W String variable: \napdos\Win-GRAF\demo-project\vb.net\_2008\_demo\demo\_vb04

## 13.5.5 How to use VB.NET R/W to Win-GRAF String Variable

Before .NET program write to Win-GRAF String variable. The String-type has to convert to byte array. If you need to read Win-GRAF String variable. Then you have to convert byte array to String. There is a VB.NET example to show how to convert each other.

(Encode :UTF-8):

#### **Convert String to byte array**

Private Function unicode\_to\_byte\_array(ByVal msg As String) As Byte() Dim tmpbuf() As Byte If msg.Length > 255 Then Return Nothing End If

tmpbuf = System.Text.Encoding.GetEncoding("UTF-8").GetBytes(msg)
Return tmpbuf
End Function

#### Convert byte array to string

Private Function byte\_array\_to\_unicode(ByVal buf() As Byte) As String Dim tmpmsg As String If buf.Length > 255 Then Return Nothing End If tmpmsg = System.Text.Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length)

Return tmpmsg End Function

# Chapter 14 C# .net 2008 Program Running in WP-8xx8 Access to Win-GRAF Variables

This chapter lists the procedure for creating the first demo program by Visual Studio .NET 2008 development tool. There are some sample programs in the WP-8xx8 CD-ROM.

## C# demo:

CD-ROM : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\ demo\_CSharp01 : Digital I/O demo with one I-87055W in slot 0 of the WP-8xx8. demo\_CSharp02 : Analog I/O demo with one I-87024W in slot 1 and one I-8017HW in slot 2. demo\_CSharp03 : Read / Write Win-GRAF internal integers, timers and real variables. (No I/O) demo\_CSharp04 : Read/Write Win-GRAF internal String variables. (No I/O)

### Win-GRAF demo:

CD-ROM : \napdos\Win-GRAF\demo-project\ "demo\_vb01.zip", "demo\_vb02.zip", "demo\_vb03.zip", "demo\_vb04.zip"

## 14.1 Add an Existing Win-GRAF Project from a ZIP

Please refer the Chapter 13.1

## 14.2 Publishing the Win-GRAF Variable for .NET

Please refer the Chapter 13.2

## 14.3 Create a New C# Project

1. First, users need to open Microsoft Visual Studio .NET 2008 software. And then in the menu of "File", please run the "New Project" .

📸 St	art Page - Microsoft Visual Studio	
Eile	] Edit View Tools Test Wir	idow Help
	New Project Ctrl+N	■■■■■■■
3	New Web Site Shift+Alt+N	Page
1	New File	
đ	Open <u>Project</u> Ctrl+O	Visual Studio 2008

2. Check the "Smart Device" on the left, then selecting the ".NET framework 3.5" and "Smart Device Project". Then entering a proper project name and the last click on "OK".

Project types:	<u>T</u> emplates:	.NET Framework 3.5	
Database     Reporting     Test     WCF     Workflow     Other Languages     Windows     Web     Smart Device     Other Languages     Reporting     Test     WCF     Workflow	Visual Studio installed templates         Smart Device         Project         My Templates         Search Online         Templates	s is the west dialog bear	
A project for smart Device applications.	unoose target platform, Framework version, and templat	e ni me nexi unalog box.	
Name: project1			
		OK 🔪	Cancel

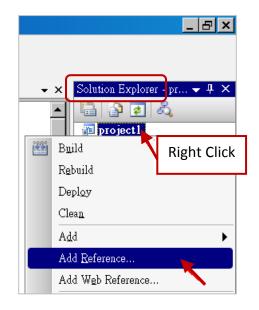
3. Select the "Device Application" and "Windows CE" and ".NET Compact Framework Version 3.5", then click on "OK".

Target platform:	Windows CE		~
.NET Compact Framework version:	.NET Compact Framewo	ork Version 3.5	~
<u>T</u> emplates:	.NET Compact Framewo .NET Compact Framewo	ork Version 2.0	
	Console Control pplication Library	Empty Project	Description: A project for creating a .NET Compact Framework 3.5 forms application for Windows CE Platform
Download additonal emulator image	s and smart device SDKs		
			OK Cancel

## 14.3.1 Add C# Project Reference

The "UserShareNet" library contains all modules' functions. Before you use the "UserShare" keyword in the program, you must add the "UserShareNet.dll" into the reference list of your application.

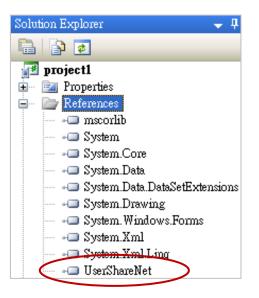
- Copy the "UserShareNet.DLL" from WP-8xx8 CD-ROM: \napdos\Win-GRAF\WP-8xx8\CSharp.net\_2008\_demo\demo\_CSharp01\ to your project folder(ex: C:\project1\)
- 2. Right click on the Project name on the right hand side , then select "Add Reference ..."



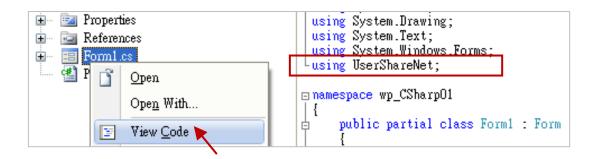
3. Click the "Browse" button. Select the "UserShareNet.dll" from your project location.

NET Projects Brows Recen 搜尋位置①: C project1	nt 💽 🕝 🍺 📂 🖽 -
obj Properties UserShareNet.dll	Change folder to your current project Path. (Ex: "C:\project1")
檔案名稱(N):	
檔案類型(I): Component Files	(*.dll;*.tlb;*.olb;*.ocx;*.exe)
	OK Cancel

**Note:** You may copy the "UserShareNet.dll" from the CD-ROM to your current project path first. Then add it to the project reference. 4. When "UserShareNet.dll" are added, you can see them in the solution explorer as below.



5. Right-click on the "Form1.cs" and select "View Code" from the pop-up. Move cursor to top and insert the "using UserShareNet;" in the first statements.



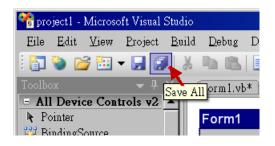
Then you can design all required objects and actions inside your C# Forms.

(Refer the Section 14.5 for more information about using functions in the "UserShareNet.dll".)

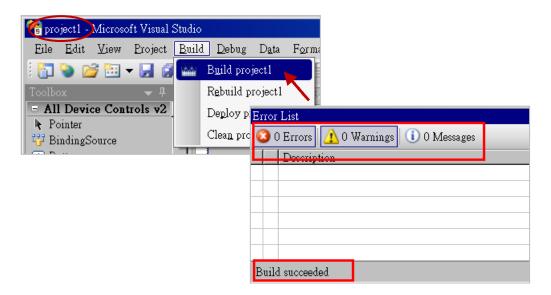
# 14.4 Compiling the Application Program

When you have finished writing a program, you can build (compile) an application by the following steps.

1. Remember to save at any time for safety.



2. Then compile (Build) the project . The result is listed in the "Error List" windows at the bottom.



3. You can find the execution file in

<Your C# .net Project folder> \bin\Release\ <project\_name>.exe

Please copy this execution file to the WP-8xx8's \System\_Disk\Win-GRAF\ path to run it.

#### Note:

The user may copy the C#.net execution file to another path to run it, but there should contain at least two DLL files with it or it cannot run correctly. For ex, the project1.exe can run in the \Micro\_SD\ path if there are three files in it. The "project1.exe", "UserShareNet.dll" and, "Quicker.dll" . (The "UserShareNet.dll" and "Quicker.dll" can be copied from the Win-GRAF PAC's "\System\_disk\Win-GRAF\" path)

## 14.5 UserShareNet.DLL

This section we will focus on the description of the application example of UserShareNet.DLL functions. There are some functions that can be used to read/write data from/to the Win-GRAF variable. The functions of UserShareNet.DLL can be divided into as listed below

- 1. R/W Boolean
- 2. R/W 8-bit Integer
- 3. R/W 16-bit Integer
- 4. R/W 32-bit Integer
- 5. R/W 64-bit Integer
- 6. R/W 32-bit Float
- 7. R/W 64-bit Float
- 8. R/W 32-bit String

**※** Refer the "<u>Appendix A</u>" to get familiar with the definition of Win-GRAF variables.

## 14.5.1 R/W Boolean Functions

Set\_BOOL

#### Description:

This function is to set a value to a Win-GRAF Boolean variable.

#### Syntax:

UserShare.Set\_BOOL(ushort iUserAddress, byte iStatus)

#### **Parameter:**

iUserAddress : Address of Variable (1 to 8192)iStatus : Set the status. For instance, iStatus = 1 for True, iStatus = 0 for False

### Example:

// Set the Win-GRAF BOOL variable with address 1 to True.
UserShare.Set\_BOOL(Convert.ToUInt16(1), 1);

### Demo program:

CD-ROM : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp01



#### **Description:**

This function is to get the value from a Win-GRAF BOOL variable.

#### Syntax:

UserShare.Get\_BOOL(ushort iUserAddress, out byte iStatus)

#### Parameter:

iUserAddress :Address of Variable. (1 to 8191)iStatus :Get the variable status , iStatus = 1 for True, iStatus = 0 for False.

#### Example:

Byte iStatus=0; // Get the value of Win-GRAF BOOL variable with address 1. UserShare.Get\_BOOL(Convert.ToUInt16(1),out iStatus);

#### Demo program:

CD-ROM: \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp01

## 14.5.2 R/W Integer Functions

Set\_SINT Set\_INT Set\_DINT Set\_LINT

#### **Description:**

These functions are to set 8-bit Integer, 16-bit Integer, 32-bit integer & 64-bit Integer value to Win-GRAF integer variables.

Syntax:

UserShare.Set\_SINT(ushort iUserAddress , sbyte iStatus)

UserShare.Set\_INT(ushort iUserAddress , short iStatus)

UserShare.Set\_DINT(ushort iUserAddress, int iStatus)

UserShare.Set\_LINT(ushort iUserAddress, long iStatus)

#### Parameter:

iUserAddress : Address of Variable. (1 to 8192)

iStatus : Set the 8-bit Integer, 16-bit Integer, 32-bit Integer or 64-bit Integer.

#### Example:

// Set a 32-bit integer value "1234567" to the Win-GRAF DINT variable with address "1".
int temp1=1234567;
UserShare.Set\_DINT(Convert.ToUInt16(1), temp );

// Set a 16-bit integer value "-1234" to the Win-GRAF INT variable with address "2".
short temp2= -1234;
UserShare.Set\_INT(Convert.ToUInt16(2), temp2 );

// Set a 64-bit Integer value "123456789012345" to the Win-GRAF LINT variable with address "3". long temp3=123456789012345; UserShare.Set\_LINT(Convert.ToUInt16(3), temp3);

// Set a 8-bit Integer value "125" to the Win-GRAF SINT variable with address "4".
Sbyte temp4=125;
UserShare.Set\_SINT(Convert.ToUInt16(4), temp4);

#### Demo program:

- R/W analog I/O: \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp02
- 2. R/W internal Boolean ,long integer, Timer and Real (floating-point) values : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp03

📕 Get\_SINT 📕 Get\_INT 📕 Get\_DINT 📕 Get\_LINT

#### **Description:**

These functions are to get 8-bit integer, 16-bit integer, 32-bit integer & 64-bit integer value from Win-GRAF integer variables.

#### Syntax:

UserShare.Get\_SINT(ushort iUserAddress, out sbyte iStatus) UserShare.Get\_INT(ushort iUserAddress, out short iStatus) UserShare.Get\_DINT(ushort iUserAddress, out int iStatus) UserShare.Get\_LINT(ushort iUserAddress, out long iStatus)

#### Parameter:

iUserAddress :Address of Variable (1 to 8192)iStatus :Get the value of Win-GRAF integer variables.

#### Example:

Int64 Dlong\_val; Int16 short\_val; Int32 long\_val ; sbyte sbyte\_val;

// Get 64-bit integer value from the Win-GRAF LINT variable with address "7". UserShare.Get\_LINT(Convert.ToUInt16(7),out Dlong\_val);

// Get 32-bit integer value from the Win-GRAF DINT variable with address "8". UserShare.Get\_DINT(Convert.ToUInt16(8),out long\_val);

// Get 16-bit integer value from the Win-GRAF INT variable with address "9".
UserShare.Get\_INT(Convert.ToUInt16(9),out short\_val);

// Get 8-bit integer value from the Win-GRAF SINT variable with address "10". UserShare.Get\_SINT(Convert.ToUInt16(9),out sbyte\_val)

#### Demo program:

- 1. R/W analog I/O: \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp02
- 2. R/W internal Boolean, long integer, Timer and Real (floating-point) values: \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp03

## 14.5.3 R/W Real variable Functions

### Get\_REAL Get\_LREAL

#### **Description:**

These functions are to get 32-bit REAL and 64-bit double from the Win-GRAF.

#### Syntax:

UserShare. Get\_REAL (System.UInt16 *iUserAddress*, out float *iStatus*)

UserShare. Get\_LREAL(ByVal *iUserAddress* As System.UInt16 , out Double *iStatus*)

#### Parameter:

iUserAddress :Address of Variable (1 to 8192)iStatus :Get the 32-bit REAL or 64-bit double value.

#### Example:

float float\_val; double double\_val;

// Get 64-bit double value from the Win-GRAF LREAL variable with address "7". UserShare.Get\_LREAL(Convert.ToUInt16(7),out double\_val);

// Get 32-bit REAL value from the Win-GRAF REAL variable with address "8".
UserShare.Get\_REAL(Convert.ToUInt16(8),out float\_val);

#### Demo program:

- R/W analog I/O: \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp02
- R/W internal long integer, Timer and Real (floating-point) values : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_ demo\_CSharp03



#### **Description:**

These functions are to set 32-bit REAL and 64-bit double value to the Win-GRAF REAL/LREAL variable.

Syntax:

UserShare. Set\_REAL ( ushort *iUserAddress* , float *iStatus* )

UserShare. Set\_LREAL( ushort *iUserAddress* , Double *iStatus*)

#### **Parameter:**

iUserAddress :Address of Variable. (1 to 8192)iStatus :Set the 32-bit REAL or 64-bit double.

#### Example:

// Set a 64-bit double value "11234.234567" to the Win-GRAF LREAL variable with address "7"
UserShare.Set\_LREAL(Convert.ToUInt16(7),Convert.ToDouble(11234.234567));

// Set a 32-bit REAL value "123.12" to the Win-GRAF REAL variable with address "2". UserShare.Set\_REAL(Convert.ToUInt16(8), Convert.ToSingle (123.12));

#### **Demo program :**

- 1. R/W analog I/O: \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp02
- R/W internal long integer, Timer and Real (floating-point) values : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp03

## 14.5.4 R/W String variable Functions

### Set\_STRING

#### **Description:**

This function is to get a Win-GRAF String variable.

#### Syntax:

UserShare.Set\_STRING (ushort addr , <a href="https://www.setshare.sets.string">Byte</a> [] msg)

#### **Parameter:**

addr : Address of Variable (1 to 1024) msg[] : Get the string value.

Example: String str\_val; Byte[] msg;

```
// Get the String value of the Win-GRAF String variable with address "7".
msg= unicode_to_byte_array(str_val);
UserShare.Set_STRING(Convert.ToUInt16(7),msg );
```

//Convert String to byte array.

```
private byte[] unicode_to_byte_array(string msg)
```

```
{
    byte[] tmpbuf;
    if (msg.Length > 255)
        return null;
    tmpbuf = Encoding.GetEncoding("UTF-8").GetBytes(msg);
    return tmpbuf;
}
```

#### Demo program:

CD-ROM:

1. R/W String variable : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp04



#### **Description:**

This function is to set a String value to the Win-GRAF String variable.

#### Syntax:

UserShare.Set\_STRING (ushort addr , Byte [] msg)

#### Parameter:

addr : Address of Variable. (1 to 1024) msg[] : Set the string value.

## Example: String str\_val= "Hello World"; Byte[] msg;

// Set a string value "Hello World" to the Win-GRAF String variable with address "7".
UserShare.Get\_STRING(Convert.ToUInt16(7),msg );
str\_val= byte\_array\_to\_unicode(msg);

```
//Convert byte array to String
private string byte_array_to_unicode(byte[] buf)
{
    string tmpmsg;
    if (buf.Length > 255)
        return null;
```

```
tmpmsg = Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length);
return tmpmsg;
```

}

### Demo program:

CD-ROM:

1. R/W String variable : \napdos\Win-GRAF\demo-project\CSharp.net\_2008\_demo\demo\_CSharp04

## 14.5.5 How to Use C# to Convert Win-GRAF String Variable

Before .NET program write to Win-GRAF String variable. The String-type has to convert to byte array. (According your .NET program Encode. Ex: UTF-8) If you need to read Win-GRAF String variable. Then you have to convert byte array to String. There is an C# example to show how to convert each other.

## Example (Encode is UTF-8):

#### //Convert String to byte array

```
private byte[] unicode_to_byte_array(string msg)
```

```
{
    byte[] tmpbuf;
    if (msg.Length > 255)
        return null;
    tmpbuf = Encoding.GetEncoding("UTF-8").GetBytes(msg);
    return tmpbuf;
}
//bute error to string
```

## //byte array to string

```
private string byte_array_to_unicode(byte[] buf)
{
    string tmpmsg;
    if (buf.Length > 255)
        return null;
    tmpmsg = Encoding.GetEncoding("UTF-8").GetString(buf, 0, buf.Length);
    return tmpmsg;
    }
```

# Chapter 15 Using eLogger HMI in the Win-GRAF PAC

"eLogger" is an HMI development tool developed by ICP DAS. It features an easy-to-use graphical user interface (GUI), not only supports the Local HMI but also the Web HMI. Users can design their own HMI pages by using eLogger, and then both the eLogger HMI and Win-GRAF softLogic can be executed in the same Win-GRAF PAC.

## 15.1 The Win-GRAF Project

1. Copy the demo file from the CD-ROM of Win-GRAF PAC to your PC desktop, and then upzip the file. \napdos\Win-GRAF\demo-project\Soft-GRAF-demo\**demo\_faq018\_all.zip**.

Win-GRAF Project	eLogger Project	Description
eL01.zip	eL_01.wez	Designing the local and web HMI pages
eL02.zip	eL_02.wez	Configuring a control button that can be instantly ON and then OFF
eL03.zip	eL_03.wez	Reading or modifying the PAC's date and time and doing some control

Demo project for the Win-GRAF and eLogger:

- 2. Restore the Win-GRAF project (.zip) to the Win-GRAF workbench on PC (see Section 13.1).
- 3. Download the Win-GRAF project (e.g., eL01) to the PAC (see Section 2.3.4 and Section 2.3.5).

For more description about the Win-GRAF and eLogger HMI application, visit: <u>www.icpdas.com</u> > Support > FAQ > <u>Win-GRAF Soft-Logic PAC</u> > <u>FAQ-018</u> or <u>http://www.icpdas.com/root/support/faq/win-graf.php</u>

## **15.2 The eLogger Project**

The user can download and setup the eLogger from the eLogger webpage, and copy all eLogger projects (.wez) to C:\ICPDAS\eLogger\eLogger\_Developer\Project on your PC, and then download the project (e.g., eL\_01) to the PAC.

eLogger web page: <u>www.icpdas.com</u> > Product > Solution > Software > <u>SCADA/HMI</u> > <u>eLogger</u> <u>http://www.icpdas.com/root/product/solutions/software/scada\_hmi/elogger/elogger.html</u>

eLogger software: http://ftp.icpdas.com/pub/cd/winpac/napdos/elogger/setup/

eLogger manuals: http://ftp.icpdas.com/pub/cd/winpac/napdos/elogger/document/

# Chapter 16 Redundancy

## 16.1 Features and Architecture

The ICP DAS Win-GRAF PAC - XP-8xx8-CE6 series support redundancy:

One redundant system is composed by two Win-GRAF PACs that one PAC's rotary switch is set to 7 (means Main-PAC) and the other one is set to 9 (means Backup-PAC). When one of them is damaged or crashed or need to release its control-right by user-defined event, the PAC control-right will automatically switch to the healthy one.

## Features of the Win-GRAF redundancy

### 1. Better safety:

There are three communication cables (LAN1, LAN2 and Alive Port) connected between two PACs. The redundant system will still control the process well, even if one or two cables are broken or disconnected. As long as one of the three communication cables is fine, the Win-GRAF redundant PACs can still work well with the process.

## 2. Unique Public IP:

The Win-GRAF redundant system provides a unique public IP address for SCADA/HMI to access it without needing to determine which one is the Active IP.

## 3. Easy maintenance:

If one of the redundant PACs is damaged someday after starting the process, you can remove the damaged one (Note: Do not shutdown or dismounting the other healthy PAC, keep it running). And then take another spare Win-GRAF PAC with the same model (or a repaired PAC) without downloading the Win-GRAF application, simply adjust its rotary switch to a proper position and then connect all required communication cables (e.g., LAN1, LAN2, Alive port and I/O). Make sure that the original, healthy PAC is still working properly and then power up the spare PAC. Then, the healthy PAC will automatically copy the Win-GRAF app and all its redundant data to that new PAC which is just online. It is easier for maintenance and installation, the operator don't have to worry about whether to install the Win-GRAF app because the healthy PAC will automatically do it for the new online PAC.

**Exception:** Except the Win-GRAF app if there are a few apps, such as the C, VB.net, C# app or eLogger HMI app, running in the redundant system, these apps need to pre-installed to the spare Win-GRAF PAC (or a repaired PAC) before installing this PAC to the redundant system.

## 4. Easy to design the application:

The user has to do is design the application program. Not necessary to specify what data should be redundant between two PACs. The Win-GRAF redundant system will automatically deliver them to the other PAC.

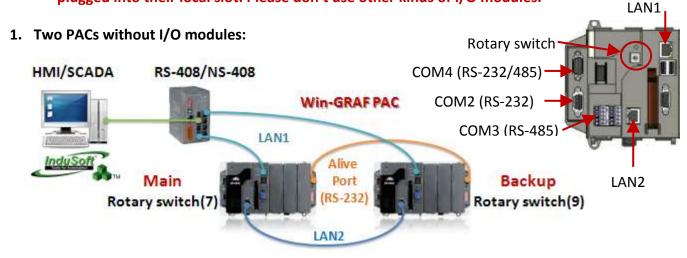
## 5. Users can design some safety in the app:

For example, if the Active PAC's LAN1 is disconnected (causes the SCADA unable to connect to) or a RS-485 Port is disconnected or damaged, and so on. The user's app can test these events and then switch control right to the other healthy PAC.

### 6. I/O Redundancy:

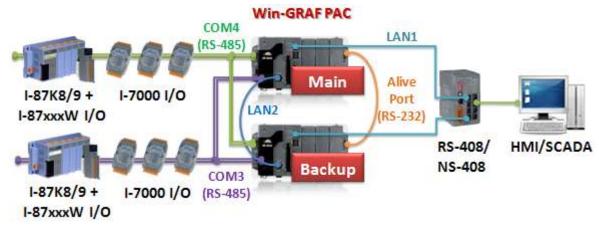
If the user chooses <u>iDCS-8830 series I/O</u>, both the PAC and I/O modules can support redundancy.

<u>The architecture of the Win-GRAF redundant system</u> (using XP-8xx8-CE6 as an example): <u>Note:</u> The Win-GRAF redundant PACs support RS-485/RS-422 expansion boards (I-8142i/ I-8144i) plugged into their local slot. Please don't use other kinds of I/O modules.



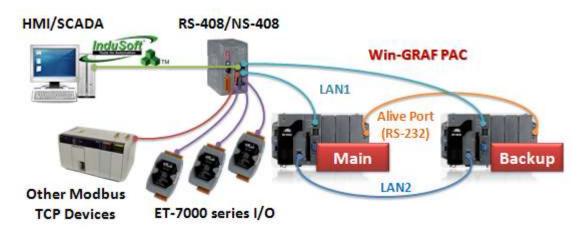
**<u>Note:</u>** <u>LAN1</u>: Normal Ethernet Cable, <u>LAN2</u>: Ethernet Crossover Cable, <u>Alive Port</u>: RS-232 Crossover Cable.

### 2. Two PACs are equipped with DCON I/O modules:



Note: LAN1: Normal Ethernet Cable, LAN2: Ethernet Crossover Cable, Alive Port: RS-232 Crossover Cable. <u>COM3, COM4 (RS-485)</u>: Data+ to Data+ ; Data- to Data- .

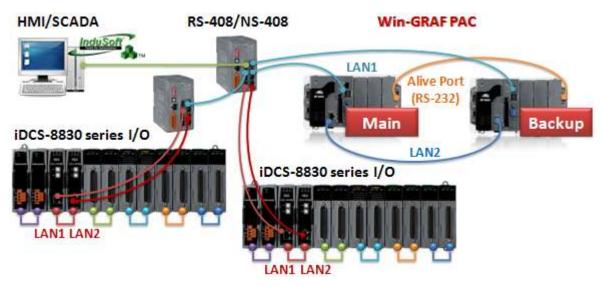
### 3. Two PACs are equipped with Modbus TCP I/O modules:



Note: LAN1: Normal Ethernet Cable, LAN2: Ethernet Crossover Cable, Alive Port: RS-232 Crossover Cable.

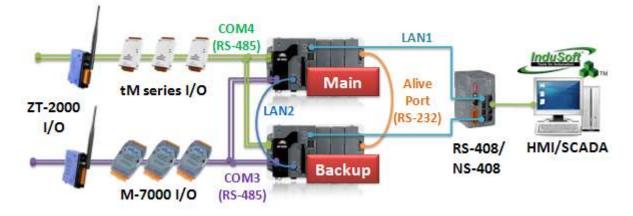
### 4. Two PACs are equipped with iDCS-8830 I/O modules:

This type of achievement provides both CPU and I/O module redundancy.



Note: LAN1 (PAC), LAN1 (iDC-8830), LAN2 (iDC-8830): Normal Ethernet Cable. LAN2 (PAC): Ethernet Crossover Cable, <u>Alive Port</u>: RS-232 Crossover Cable.

- Note: Each pair of redundant I/O modules that plugged into the iDCS-8830 must have the same model numbers.
- 5. Two PACs are equipped with other Modbus RTU/ASCII I/O modules:



- Note: LAN1: Normal Ethernet Cable, LAN2: Ethernet Crossover Cable, Alive Port: RS-232 Crossover Cable. <u>COM3, COM4 (RS-485)</u>: Data+ to Data+ ; Data- to Data- .
- 6. It can also equip with two (or more) kinds of I/O modules such as item (2) to (5).

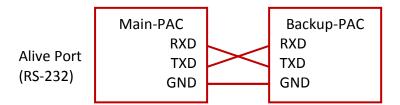
The Win-GRAF redundant system is composed by two PACs. Users need to set one PAC's rotary switch to 7 (called Main-PAC) and set the other one to 9 (called Backup-PAC). **Do not** use two Main PACs or two Backup PACs to make up a redundant system.

# 16.2 Important Communication Ports and Installation Notes

The Win-GRAF redundant PACs require the following three communication ports to communicate with each other.

### 1. Alive Port:

The Win-GRAF redundant PACs use one RS-232 Port as the Alive Port (also called Heart-beat Port). This Alive Port must use a RS-232 crossover cable (or NULL Modem Cable), which link with each other as the following figure.



## 2. Replication Port:

The Win-GRAF redundant PACs use their Ethernet Port (LAN2) as a Replication Port. Both of PAC's LAN2 ports use an Ethernet crossover cable to transfer redundant data. Do not use any Ethernet Switch or Hub between them, otherwise it may cause an error or timeout. The LAN2 ports are based on fast Ethernet and dedicated Ethernet ports in order to avoid collisions. So don't connect any external devices, Switches, and Hubs to these two PAC's LAN2.

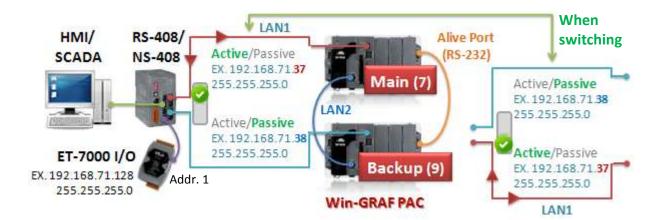


The LAN2 port uses an Ethernet crossover Backup-PAC (LAN2) cable connected directly between two

PAC. (Do not use a normal Ethernet cable).

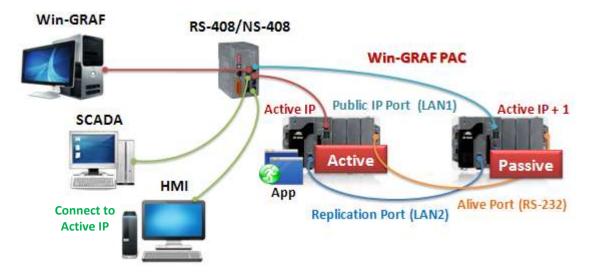
### 3. Public IP Port:

The Win-GRAF redundant PAC's Ethernet Port (LAN1) must connect an Ethernet Switch via a normal Ethernet cable. After that, it can be used to communicate with SCADA/HMI or connect and control external Modbus TCP I/O modules, devices or other Ethernet devices. The LAN1 port can switch its IP address automatically. If the PAC is Active, the LAN1 IP address will switch to the "Active\_IP" address which defined in the user's Win-GRAF project. And if the PAC is Passive, the LAN1 IP address will switch to the "Active\_IP+1" address automatically. The SCADA/HMI can use the "Active\_IP" address to communicate with the Win-GRAF redundant system.



## PAC Installation Notes (Very Important):

- Before power up PACs, make sure one PAC's rotary switch is set to 7 and the other one is set to 9. The redundant system will be crazy due to the wrong settings.
- 2. When installing the Win-GRAF redundant system at the application field, make sure the following three cables are connected properly (connect all required cables, such as RS-485) before power up PACs. If user power up PACs before connecting these three communication cables, the redundant system will be out of control.
  - A. Connect both of the PAC's Alive ports by using a crossover cable.
  - B. Connect both of the PAC's LAN2 ports by using an Ethernet crossover cable. (Do not use any Ethernet Switch/Hub between LAN2 ports.)
  - C. Connect both of the PAC's LAN1 ports to an Ethernet switch by using a normal Ethernet cable.
- 3. If only one healthy PAC of the redundant system is working properly at the application field, do not power-off or shut it down. Before user power up the other PAC that will be installed into the system, follow the step1 and step2 as mentioned above to set up it first.



Only the Active PAC (i.e., PAC got the control-right) can run the Win-GRAF application. The Passive PAC will not run the Win-GRAF application. It simply receives the redundant data from the Active PAC and wait for getting control-right in the future.

# 16.3 Description of Win-GRAF Demo Projects

The shipping CD of the Win-GRAF PAC provides these three demo projects – "demo\_RDN\_1.zip", "demo\_RDN\_2.zip" and "demo\_RDN\_3.zip" – related to the redundant system. Refer <u>Chapter 12</u> to restore these files into the Win-GRAF Workbench.

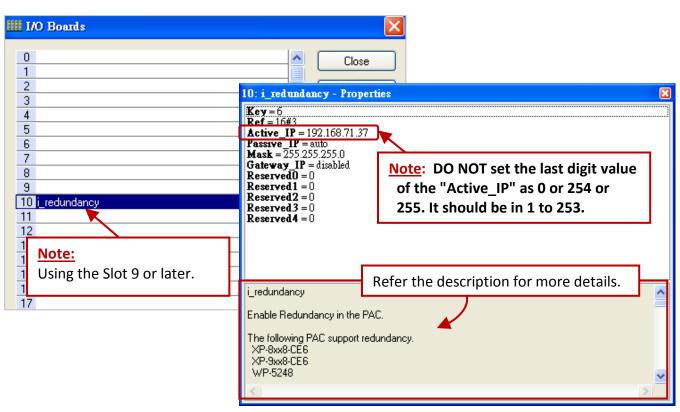
Project Name	oject Name Description			
demo_RDN_1	Two XP-8xx8-CE6 PACs, using their COM3 to connect three DCON I/O modules.			
demo_RDN_2	Two XP-8xx8-CE6 PACs without connecting any I/O module.			
demo_RDN_3	Two XP-8xx8-CE6 PACs, using their LAN1 to connect a ET-7050 (Modbus TCP I/O module) through one Ethernet switch.			
demo_RDN_4	Two XP-8xx8-CE6 PACs, using their LAN1 to connect an iDCS-8830 (Both PAC and I/O are redundant) through one Ethernet switch.			

The following sections will describe the "demo\_RDN\_2" program.

## 16.3.1 "I/O Board" Settings

## demo\_RDN\_2, demo\_RDN\_3, demo\_RDN\_4::

To use redundancy on the PAC, first link the "i\_redundancy" in the "I/O Board" window. (Refer <u>Chapter 4</u>).



### Parameters:

Active\_IP: The redundant system provides a public IP address for some HMI/SCADA to communicate. (Note: DO NOT set the last digit value of the "Active\_IP" as 0 or 254 or 255. It should be in 1 to 253.)

- **Passive\_IP:** Auto, means the LAN1 IP address of the current Passive PAC, it will be automatically set as Active\_IP +1 (e.g., if the "Active\_IP" is set as "192.168.71.37", the "Passive\_IP" will automatically set as "192.168.71.38")
- Mask: The most common settings are either 255.255.255.0 or 255.255.0.0 (depends on the network environment).

After linking the "i\_redundancy" in the "I/O Boards" window, it will auto add 12 "BOOL" input channels in the "Variables" window that can be used to display the current state of the redundant system.

Variables										X
🝸 Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description		
🖃 📫 %IX10 - i_redundancy										^
%IX10.0=is_Main_Active	BOOL		Input							
WIV10 1-ie Backup Active	BOOL		Input							
is_Backup_Active	BOOL		Input							
%IX10.3=is_Backup_ready	ROOL		loout							
%IX10.4=is first cycle just afte	Double-click it	t to ad	/ a h	variah	le					
%IV10.5=ie_Main_LAN1_ok				anab						
%IX10.6=is Backup LAN1_ok	name to each	cnanr	nei.							
%IX10.7=is Alive port_ok	BOOL		Input							
%IX10.8=is_Passive_ready	BOOL		Input							
%IX10.9=is Active LAN1 ok	BOOL		Input						J	
%IX10.10=is Passive LAN1 ok	BOOL		Input							
%IX10.11	BOOL		Input						1	~
<									>	
IO Drivers NewSpy1 RDN control Variables Retain and timer										

Ch.0 (is_Main_Active):	Is the Main-PAC (rotary switch: 7) active now? TRUE: Active , FALSE: Passive
Ch.1 (is_Backup_Active):	Is the Backup-PAC (rotary switch: 9) active now? TRUE: Active , FALSE: Passive
Ch.2 (is_Main_ready):	Is the Main-PAC ready? If Ch.2 returns FALSE. The possible reason could be the following. (1) The Ethernet cable (LAN2) between Main and Backup PAC is broken. (2) The Main PAC is dead or damaged. (3) The rotary switch of the Main PAC is not set at 7.
Ch.3 (is_Backup_ready):	Is the Backup-PAC ready? If Ch.3 returns FALSE. The possible reason could be the following. (1) The Ethernet cable (LAN2) between Main and Backup PAC is broken. (2) The Main PAC is dead or damaged. (3) The rotary switch of the Main PAC is not set at 9.
Ch.4 (is_first_cycle_just_aft	er_switch): For Active PAC only.
	True: Now is in the first cycle just after switching.
	False: Now is not in the first cycle after switching.
Ch.5 (is_Main_LAN1_ok):	Is the LAN1 port of the Main-PAC ok?
	TRUE: OK , FALSE: Fail or Ethernet cable is disconnected.
Ch.6 (is_Backup_LAN1_ok):	Is the LAN1 port of the Backup-PAC ok?
	TRUE: OK , FALSE: Fail or Ethernet cable is disconnected.
Win-C	GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 16-7

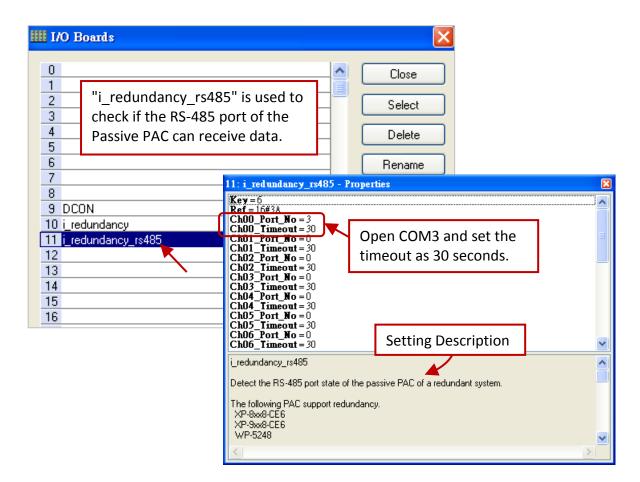
Ch.7 (is_Alive_port_ok):	True : The communication of the Alive Port is ok. False: The communication of the Alive port fails or the Passive PAC is dead or damaged.
Ch.8 (is_Passive_ready):	Is the Passive PAC ready now? If Ch.8 returns FALSE. The possible reason could be the following. (1) The Ethernet cable (LAN2) between Main and Backup PAC is broken. (2) The Passive PAC is dead or damaged. (3) The rotary switch setting of the Passive PAC is incorrect.
Ch.9 (is_Active_LAN1_ok):	Is the LAN1 port of the Active-PAC ok? TRUE: OK ,FALSE: Fail or Ethernet cable is disconnected.
Ch.10 (is_Passive_LAN1_ok	): Is the LAN1 port of the Passive-PAC ok? TRUE: OK ,FALSE: Fail or Ethernet cable is disconnected.

demo\_RDN\_1: To connect DCON I/O modules via PAC's COM3 (RS-485).

IIII I/O Boards		×
0 1 2 3 4 5 6 7		Close Select Delete Rename
8 9 DCON	9: DCON - Properties	
10 i_redundancy           11 i_redundancy_rs485           12           13           14           15           16	Key = 6 Ref = 16#5 Port = 3 Baud_rate = 9600 Host_watchdog_Enabled = 0 Watchdog_timeout = 5000 Checksum_enabled = 0 Delay_ms_between_polls = 0 Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0	Connect DCON I/O modules by PAC's COM3 and the Baud rate is 9600.
		Setting Description
	DCON Enable one serial port (RS-485) to co Note: 1. This "DCON" supports only the co That is "No-parity", "8 character s So please must configure all DCON	size" and "1 stop bit".

### **Important Notice:**

- 1. Please must also use the "i\_redundancy" or the "i\_redundancy\_rs485" will not work.
- 2. The "i\_redundancy\_rs485" will only open the related RS-485 ports to receive data in the passive PAC. It doesn't send any data.
- 3. The "i\_redundancy\_rs485" is used to detect whether the Passive PAC's RS-485 port can receive data.



#### Parameters:

### Ch00\_Port\_No ~ Ch15\_Port\_No :

The used RS-485 port number of the Passive PAC. Can be 0 or 1 to 33 depends on the PAC model. Set 0 means disable it.

#### Ch00\_Timeout ~ Ch15\_Timeout :

The unit is second. Can be 1 to 60 seconds.

If there is no data received in the timeout interval of the related RS-495 port, the status will reset as FALSE.

#### 16-ch Boolean Inputs :

It used to represent state of RS-485 ports in the passive PAC.

TRUE : The related RS-485 port open ok and can receive data.

FALSE: The related RS-485 port open fail or receive no data in the timeout interval.

# 16.3.2 Declaring Variables (demo\_RDN\_2)

Users can view or add variables in the "Variable" window (refer Section 2.3).

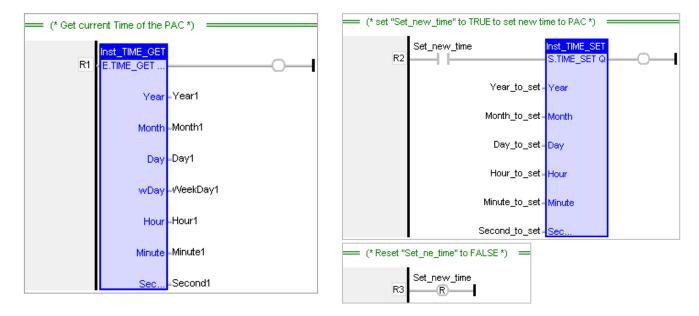
Name	Data Type	Description			
Year1	DINT				
Month1	DINT				
Day1	DINT				
WeekDay1	DINT	Used in the "PAC_Time" program: They are used to get the PAC's system time.			
Hour1	DINT	They are used to get the FAC's system time.			
Minute1	DINT				
Second1	DINT				
Set_new_time	BOOL	Set it as "TRUE" to set up new system time.			
Year_to_set	DINT				
Month_to_set	DINT				
Day_to_set	DINT	Used in the "PAC_Time" program:			
Hour_to_set	DINT	They are used to set the PAC's system time.			
Minute_to_set	DINT				
Second_to_set	DINT				
DINT_1	DINT				
DINT_2	DINT	Used in the "Retain_and_timer" program:			
REAL_1	REAL	Set them as retain variables.			
REAL_2	REAL				
TMR_1	TIME	Timor			
TMR_2	TIME	Timer			
retain_done	BOOL	TRUE: Retain variables are well set up; FALSE: Not set up yet.			
on_line_change_cycle	DINT	Non-zero, means this is the first cycle just after On-Line change.			
tmp_bool	BOOL	It used to return the Retain status.			
TMR_1_last_state BOOL		TRUE: Ticking ; FALSE: Sleeping.			
TMR_2_last_state BOOL		TRUE: Ticking ; FALSE: Sleeping.			
To_tick_TMR_1 BOOL		Set it as TRUE to start TIMER1.			
To_tick_TMR_2	BOOL	Set it as TRUE to start TIMER2.			
To_stop_TMR_1	BOOL	Set it as TRUE to stop TIMER1.			
To_stop_TMR_2	BOOL	Set it as TRUE to stop TIMER2.			

## 16.3.3 Introduction of the "demo\_RDN\_2" Project

This project includes one LD program and one ST program.

### LD Program – "PAC\_Time"

It used to get/set the system time of PAC.



## LD Program – "RDN\_control"

When an error occurs on the Active PAC's LAN1 and if the Passive PAC is ready and its LAN1 is healthy, then the Active PAC will wait for a short time to reboot, and then the other PAC will take the control-right.

(* Switch to Passive PAC if Passive is ready	and its	LAN1 is ok h	ioweve	r Active LAN1 communication has problem *)
is_Active_LAN1_ok	IN PT	Inst_TON TON	Q ET	is_Passive_ready is_Passive_LAN1_ok

### ST Program – "Retain\_and\_timer"

(\* "on\_line\_change\_cycle" is declared as DINT (nonezero means it is in the cycle jsut after doing on line change).
 "retain\_done" is declared as BOOL and inited as FALSE.
 "tmp\_bool" is declared as BOOL. \*)

```
on_line_change_cycle := GetSysInfo (_SYSINFO_CHANGE_CYCLE);
if (retain_done = FALSE) or
  (is_first_cycle_just_after_switch = TRUE) or
  (on_line_change_cycle <> 0) then
   retain_done := TRUE; (*just do it one time *)
   tmp_bool := Retain_Var( DINT_1, 1); (* retain a DINT variable *)
   tmp_bool := Retain_Var( DINT_2, 2);
   tmp_bool := Retain_Var( REAL_1, 3); (* retain a REAL variable *)
   tmp_bool := Retain_Var( REAL_2, 4);
```

```
(* if Retain variables havn't been inited yet, use default value *)
if (DINT_1 < -1000000) or (DINT_1 > 1000000) or
   (DINT_2 < -2000000) or (DINT_2 > 2000000) or
   (REAL_1 < -9.9E10) or (REAL_1 > 9.9E10) or
   (REAL_2 < -9.9E10) or (REAL_2 > 9.9E10) then
DINT_1 := 0;
DINT_2 := 0;
REAL_1 := 0.0;
REAL_2 := 0.0;
end_if;
end_if;
```

```
(* is_first_cycle_just_after_switch :
    TRUE : just in the cycle after switching.
    FALSE : other cycle *)
```

```
if is_first_cycle_just_after_switch then
```

## (\* The Timer ticking state is not auto-redundant. So we have to process them here. Ticking timer in the cycle just after switching if its last state is "ticking" \*)

if TMR\_1\_last\_state then
 tStart(TMR\_1);
 end\_if;
 if TMR\_2\_last\_state then
 tStart(TMR\_2);
 end\_if;
end\_if;

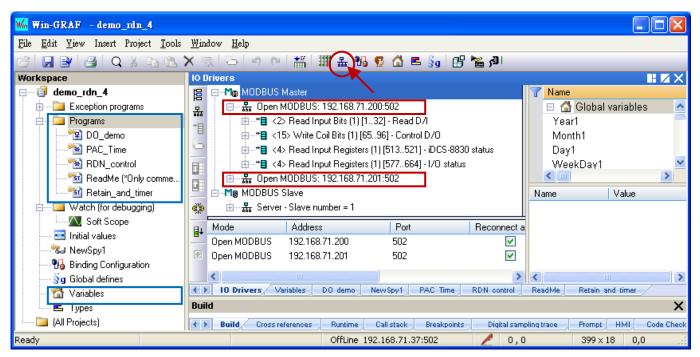
#### (\* Timer operation \*)

```
if To_tick_TMR_1 then
 To_tick_TMR_1 := FALSE ;
 tStart(TMR 1);
 TMR_1_last_state := TRUE ;
end if;
if To tick TMR 2 then
 To tick TMR 2 := FALSE;
 tStart(TMR_2);
 TMR 2 last state := TRUE ;
end if;
if To_stop_TMR_1 then
 To_stop_TMR_1 := FALSE ;
 tStop(TMR 1);
 TMR 1 last state := FALSE;
end if;
if To stop TMR 2 then
 To_stop_TMR_2 := FALSE ;
 tStop(TMR 2);
 TMR_2_last_state := FALSE ;
end if;
```

## 16.3.4 Introduction of the "demo\_RDN\_4" Project

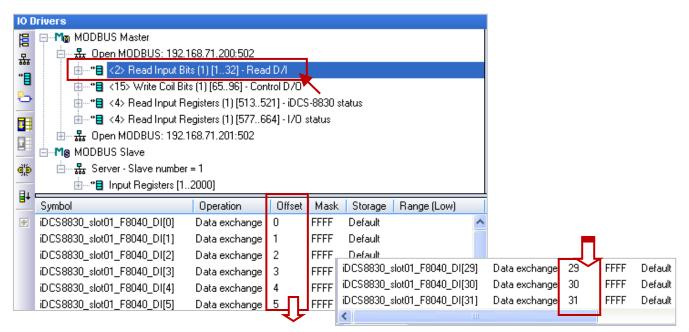
In the "demo\_rdn\_4" project, you can click the program name to view its content, click "Variables" to see all used variables, or refer <u>Section 16.3.1</u> to view the I/O Board settings ("i\_redundancy"). This section will introduce you the "Modbus Master" function (refer <u>Chapter 5</u> for more details about operations and the way to set up continuous Offset value for multiple variables).

In this example, we use one redundant I/O expansion unit (iDCS-8830), two redundant DI module (F-8040) plugged in its I/O Solt0, 1, and two redundant DO module (F-8041) plugged in its I/O Solt2, 3. Before starting the test, refer <u>Section 16.4.2</u> to configure the iDCS-8830 to work properly. Click the "Open Fieldbus Configuration" button to open the "I/O Drivers" setting window.



We enable the Modbus Master function to connect two Modbus TCP Slave devices, i.e., each iDCS-8830 redundant I/O unit has two IP addresses, they are "192.168.71.**200**" and "192.168.71.**201**" (Port: 502), which used to read/write the I/O data and status.

Read Digital Inputs (Using two redundant DI module - F-8040 in the slot0, 1)
--



Note: We use an iDCS-8830 redundant I/O unit in this example, refer the iDCS-8000 user manual (CH4 Modbus Addresses Mapping) to know how to input a proper "Base address". http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/fcm-mtcp\_software\_usermanual\_en.pdf

MODBUS Master Request							
Request Description:	Read D/I				ок		
<u>S</u> lave/Unit:	1				Cancel		
MODBUS Request							
<1 > Read Coil Bits <2 > Read Input Bits			^				
<3> Read Holdin	ng Registers		~				
Data block		7					
Base <u>a</u> ddress:	1		Read 3	2 DI 9	status from	n the address 1.	7
<u>N</u> b items:	32		neuu s				
Activation				í I	"Periodic	:: 0 ms" means :	sending
• Periodic:	0 m	ns 30	00			est continuously	-
◯ On call ◯ On change		(on	n error)		-	eption occurred	
					3 second	s to send next r	equest).
Misc. Timeout:	1000 m	15					1
						second means	
Nb trials:	1		commu	unca	tion timeo	ul.	

#### Write to Digital Outputs (Using two redundant DO module - F-8041 in the slot2, 3)

In this example, to write 32 DO status from the address "65" (other settings like the figure above).

10 Drivers						MODBUS Master Request			
5 1 1 1 1 1 1	Image: Second Status         Image: Second Status					Request         Description:       Control D/O         Slave/Unit:       1         MODBUS Request          <5> Write single coil bit          <6> Write single holding register          <15> Write Coil Bits          <16 > Write Holding Register			
<b>]</b> +	Symbol iDCS8830_slot23_F8041_D0[0] iDCS8830_slot23_F8041_D0[1] iDCS8830_slot23_F8041_D0[2] iDCS8830_slot23_F8041_D0[3]	Operation Data exchange Data exchange Data exchange Data exchange	Offset 0 1 2 3	Mask FFFF FFFF FFFF FFFF	Storage Default Default Default Default	Data block     Base <u>a</u> ddress: 65 <u>N</u> b items: 32     Activation			
	iDCS8830_slot23_F8041_D0[4] iDCS8830_slot23_F8041_D0[5]	Data exchange Data exchange	4	FFFF FFFF	Default Default				
	iDCS8830_slot23_F8041_D0[28] iDCS8830_slot23_F8041_D0[29] iDCS8830_slot23_F8041_D0[30] iDCS8830_slot23_F8041_D0[31]	Data exchange Data exchange Data exchange Data exchange	28 29 30 31	FFFF FFFF FFFF FFFF	Default Default Default Default	Misc. <u>T</u> imeout: 1000 ms Nb trials: 1			

#### Read the status of the iDCS-8830

- The "Operation" field of the "iDCS8830\_LAN1\_error\_status" variable is set as "Error report" in order to show an error code when reading failed. The error code will be reset to "0" when reading success. Moreover, its "Offset" field must set as "0".
- Due to the data type of "iDCS8830\_System\_minor\_fault\_status" and "iDCS8830\_System\_major\_fault \_status" is "DWORD" (32 bit), the "Offset" must use two Modbus address and the "Storage" must set as "DWORD (Low-High)".

101	IO Drivers *								
冒	E I I I I I I I I I I I I I I I I I I I								
뮮									
	□								
*∎	■								
8									
	🛨 🕀 🗄 <4> Read Input Registers (1) [577664] - I/O status								
	由 品。 Copen MODBUS: 192.168.3	71.201:502							
¢,									
		0	0.0	- 1	M I	0			
∎∔	Symbol	Operation	Offs		Mask	Storage	Rang		
	iDCS8830_LAN1_error_status	Error report	0		FFFF	Default	_		
-	iDCS8830_FCM1_mode	Data exchange	0		FFFF	Default			
	iDCS8830_FCM2_status	Data exchange	1		FFFF	Default			
	iDCS8830_system_bus_status	Data exchange	2		FFFF	Default			
	iDCS8830_FPM_status	Data exchange	3		FFFF	Default			
	iDCS8830_System_minor_fault_status	Data exchange	4		FFFF	DWORD (Low - High)			
	iDCS8830_System_major_fault_status	Data exchange	6		FFFF	DWORD (Low - High)			
	iDCS8830_LAN1_ID	Data exchange	8		FFFF	Default	~		
	<						>		

<u>Note:</u> Refer the iDCS-8000 user manual (CH4 Modbus Addresses Mapping) to fill in the "Base address". <u>http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/fcm-mtcp\_software\_usermanual\_en.pdf</u>

MODBUS Master Request							
Request DCS-8830 status							
Slave/Unit: 1 MODBUS Request							
<2> Read Input Bits <3> Read Holding Registers <4> Read Input Registers <e> Write circle coil bit</e>							
Data block       Base address:     513       Nb items:     9	Al values from the address 513.						
Activation            • Periodic:         0         0         ms         3000         On call         (on error)         On change	"Periodic: 0 ms" means sending the request continuously. (If an exception occurred, waiting 3 seconds to send next request).						
	no response over 1 second means						
	mmunication timeout.						

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 16-15

### Read the I/O status of the iDCS-8830

10 0	rivers *								
眉	🖃 🖓 MODBUS Master								
恭	白								
*	□								
	🗄 *🛢 <15> Write Coil Bits (1) [I								
8	🗄 * 🗧 <4> Read Input Register			status					
	🗄*🔒 <4> Read Input Register	rs (1) [577664] - 1/0	Distatus -						
	📄 🚠 Open MODBUS: 192.168.71	.201:502				•			
	Symbol	Operation	Offset	Mask	Storage		Rai		
ġ,þ	iDCS8830_io_slot_status[0]	Data exchange	16	FFFF	Default				
∎∔	iDCS8830_io_slot_status[1]	Data exchange	17	FFFF	Default				
	iDCS8830_io_slot_status[2]	Data exchange	18	FFFF	Default				
÷	iDCS8830_io_slot_status[3]	Data exchange	19	FFFF	Default				
	iDCS8830_io_slot_status[4]	Data exchange	20	FFFF	Default				
	iDCS8830_io_slot_status[5]	Data exchange	21	FFFF	Default				
	iDCS8830_io_slot_status[6]	Data exchange	22	FFFF	Default				
	iDCS8830_io_slot_status[7]	Data exchange	23	FFFF	Default				
	iDCS8830_io_emergency_status[0]	Data exchange	32	FFFF	Default				
	iDCS8830_io_emergency_status[1]	Data exchange	33	FFFF	Default				
	iDCS8830_io_emergency_status[2]	Data exchange	34	FFFF	Default				
	iDCS8830_io_emergency_status[3]	Data exchange	35	FFFF	Default				
	iDCS8830_io_emergency_status[4]	Data exchange	36	FFFF	Default	"DWORD	" is a 32-bit data type,		
	iDCS8830_io_emergency_status[5]	Data exchange	37	FFFF	Default	and need	s 2 Modbus addresses.		
	iDCS8830_io_emergency_status[6]	Data exchange	38	FFFF	Default				
	iDCS8830_io_emergency_status[7]	Data exchange	39	FFFF	Default	K			
	iDCS8830_io_channel_break_status[0]	Data exchange	72	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[1]	Data exchange	74	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[2]	Data exchange	76	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[3]	Data exchange	78	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[4]	Data exchange	80	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[5]	Data exchange	82	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[6]	Data exchange	84	FFFF	DWORD	(Low - High)			
	iDCS8830_io_channel_break_status[7]	Data exchange	86	FFFF	DWORD	(Low - Hiah)			
	<						>		

### Note:

Refer the iDCS-8000 user manual (CH4 Modbus Addresses Mapping) to fill in the "Base address". <u>http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanu</u> <u>al/fcm-mtcp\_software\_usermanual\_en.pdf</u> In this example, to read 88 AI values from the address 577.

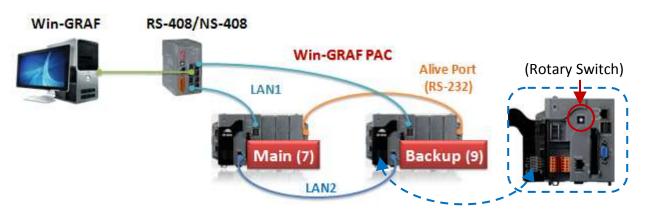
Refer <u>Chapter 5</u> for more details on Modbus Master settings and refer <u>Chapter 3</u> for Modbus Slave settings.

#### MODBUS Master Request

Request			
Description:	I/O status		
<u>S</u> lave/Unit:	1		
MODBUS Request			
<2> Read Inpu <3> Read Holdi <4> Read Inpu <5 Write circl	ng Registers t Registers		<ul> <li>•</li> <li>•</li> </ul>
Data block			
Base <u>a</u> ddress:	577		
<u>N</u> b items:	88		
Activation			
• <u>P</u> eriodic:	0	ms	3000
◯ On caḷ ◯ On change		,	(on error)
Misc.			
<u>T</u> imeout:	1000	ms	
Nb trials:	1		

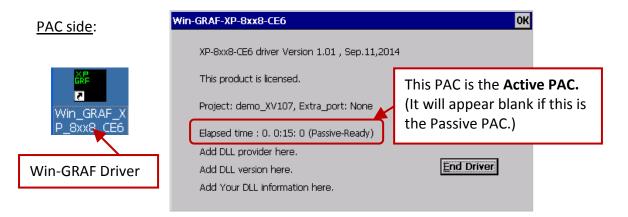
# 16.4.1 Test the "demo\_RDN\_2" and "demo\_RDN\_3" Project

**demo\_RDN\_2:** Two PAC (XP-8xx8-CE6) without connecting any I/O modules.

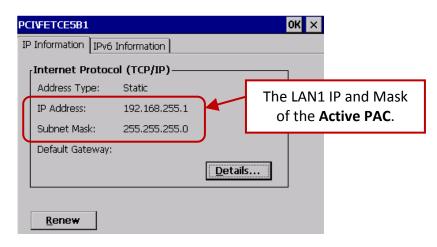


- Hardware installation (using XP-8xx8-CE6 as an example): Refer <u>Section 16.2</u> – PAC Installation notes, make sure three communication ports of the PAC have been connected properly and the rotary switch is set to 7 (Main-PAC) or 9 (Backup-PAC).
- 2. If there is no redundancy app in the redundant system yet (that is, no control-right switching procedures), start the Main-PAC (7) first and then start the Backup-PAC (9), in that case the Active PAC will be the Main-PAC.

(Later, users can run the Win-GRAF driver on a PAC's desktop to see which one is the Active PAC.)



 First, look up the LAN1 IP address of the current Active PAC (factory defaults: IP=192.168.255.1, Mask=255.255.255.0, refer <u>Section 1.3</u>), and make sure that your PC is on the same network segment (e.g., IP=192.168.255.x).



Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 16-17

(2) At the first time to download the Win-GRAF redundancy application, users Must modify the communication IP address (refer <u>Section 2.3.5</u> - "Communication Parameters") to the LAN1 IP address of the current Active PAC.

Communication Settings	X	
T5 Runtime 192.168.255.1:502 192.168.255.1:502 192.168.71.37:502	OK Cancel dress. Help	Note: If the user wants to set the timeout value (default: 3 seconds), see Section 2.3.5. (E.g. Set the IP to "192.168.255.1:502(10)" which means the timeout is 10 seconds.

3. Recompile the "demo\_RDN\_2" project and then download it to the Active PAC (refer Section 2.3.4 and Section 2.3.5 for details). After that, the LAN1 of the Active PAC will be automatically set as the Active IP (i.e., "192.168.71.37" in this example program, refer Section 16.3.1) and the LAN1 of the Passive PAC will be set as the Active IP + 1 (i.e., "192.168.71.38") automatically.

PCIVFETCE5B1 OK ×	PCIVFETCE5B1	OK ×
IP Information IPv6 Information	IP Information IPv6 Information	
Internet Protocol (TCP/IP) Address Type: Static IP Address: 192.168.71.37 Subnet Mask: 255.255.255.0 Default Gateway:	Internet Protocol (TCP/IP) Address Type: Static IP Address: 192.168.71.38 Subnet Mask: 255.255.0 Default Gateway: Default Gateway: Details	
The LAN1 IP and Mask of the Active PAC after running the "demo_RDN_2".	The LAN1 IP and Mask of the <b>Passive PAC</b> .	

4. Right now, Win-GRAF will show "Communication error" because the current Active PAC IP (e.g., 192.168.71.37) and the communication IP settings on the workbench (e.g., 192.168.255.1) are not on the same network domain. So, stop the connection and change the communication IP of this "demo\_RDN\_2" project to "192.168.71.37" (refer Section 2.3.5 "Communication Parameters") and then check if the PC's IP is on the same network domain (e.g., "192.168.71.x"). Then, this project will always link to the Active PAC whenever the user wants to debug it or change it.

Communication error         Stop the connection.	Communication Settings	×
<u>Note</u> : If the user wants to set the timeout value (default: 3 seconds), see <u>Section 2.3.5</u> . (E.g. Set the IP to "192.168.71.37:502 <b>(10)</b> " which means the timeout is <b>10</b> seconds.	T5 Runtime         192.168.71.37:502         192.168.255.1:502         192.168.71.37:502	OK Cancel Browse <u>H</u> elp

5. After re-connecting with the PAC, click "NewSpy1" to open the spy list window and you can see the current Active PAC is the Main-PAC.

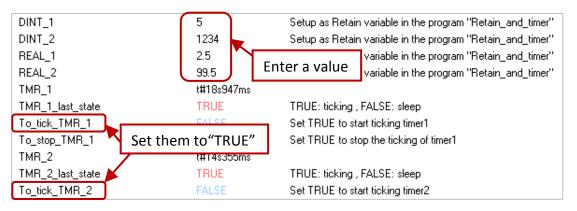
### Note:

Don't switch control-right if the Passive PAC is not ready (i.e., "is\_Passive\_ready" = FALSE) because it cannot receive the redundant data from the Active PAC. The user can click the "Redundancy" button () to switch control-right for testing.

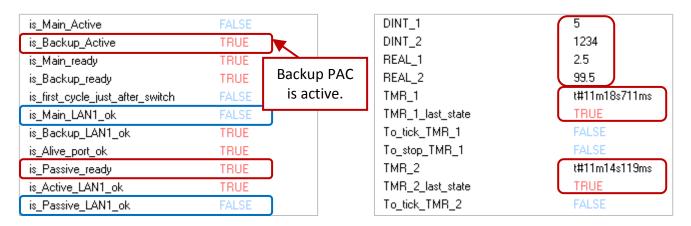


<u>File Edit V</u> iew Insert Project <u>]</u>	<u>I</u> ools	<u>W</u> indow <u>H</u> elp				
2   🖬 🖻   🕹   👗 🖻 🖻	X	🕵 🕒 🕑 🕑 🛗 🎹 몶	% 😨 🟠 🛋	§g	🔁 🎽 🚰 RUN 🛛 💿 🗆 🦻 🏭 🛱 💷 🌘	i 🧔 🕽
Workspace		NewSpy1.spl				
🖃 🗝 🕨 demo_rdn_2 [RUN]		Name	Value		Description	
🗄 🚞 Exception programs		Hour1	12			
🖕 🔤 Programs	<b></b>	Minute1	35			
PAC_Time	₽Ļ	Second1	21		The Main-PAC is active.	
BDN_control		is_Main_Active	TRUE		The Main TAC IS detive.	
🔤 Retain_and_timer		is_Backup_Active	FALSE			
🚊 🚞 Watch (for debugging)		is_Main_ready	TRUE		The Main-PAC and Backup-PAC	
Soft Scope		is_Backup_ready	TRUE		are ready and their LAN1 are	
Initial values		is_first_cycle_just_after_switch	FALSE			
NewSpy1		is_Main_LAN1_ok	TRUE		working properly.	
Binding Configuration		is_Backup_LAN1_ok	TRUE			
🚽 🖇 😏 🕄 😏 💡 💡 💡 🤤		is_Alive_port_ok	TRUE		The Alive Port (RS-232)	
🔤 🏠 Variables		is_Passive_ready	TRUE			
🛄 🛃 Types		is_Active_LAN1_ok	TRUE		communication is ok.	
		is_Passive_LAN1_ok	TRUE		The Passive PAC is ready.	
		DINT_1	0			ier"
		DINT_2	0		n	ier"
		REAL_1	0.0		- · · · ·	ier"
		REAL_2	0.0		Lourspier renew renew program renew _enc_m	ier"
		TMB_1	t#Os			
		TMR_1_last_state	FALSE		TRUE: ticking , FALSE: sleep	
		To_tick_TMR_1	FALSE		Set TRUE to start ticking timer1	
		To_stop_TMR_1	FALSE		Set TRUE to stop the ticking of timer1	
		TMR_2	t#Os			
		TMR_2_last_state	FALSE		TRUE: ticking , FALSE: sleep	
		To_tick_TMR_2	FALSE		Set TRUE to start ticking timer2	
		To_stop_TMR_2	FALSE		Set TRUE to stop the ticking of timer2	

 Enter values for "DINT\_1", "DINT\_2", "REAL\_1" and "REAL\_1" variables and then set the "To\_tick\_TMR\_1" and "To\_tick\_TMR\_2" as TRUE (it will auto reset to FALSE) to start the "TMR\_1" and "TMR\_2" ticking. Now, the status of TIMER will change from FALSE to TRUE.



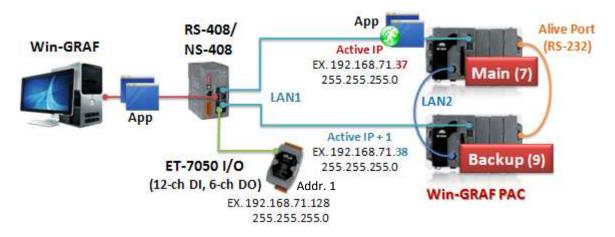
7. Make sure the Passive PAC is ready (i.e., is\_Passive\_Ready is TRUE), remove the LAN1 cable of the Main-PAC or turn off and on (restart) the Main-PAC. Wait for a short time (refer the "RDN\_control" program), the Main-PAC will automatically reboot and give control-right to the Backup-PAC, Now. Then, the Active PAC belongs to the Backup-PAC and all the values you set before still exist and the Timer is still ticking.



(<u>Note:</u> After that, plug in the LAN1 cable of the Main-PAC again. Later, both the "is\_Main\_LAN1" and the "is\_Passive\_LAN1" status will become "TRUE".)

### demo\_RDN\_3:

Two XP-8xx8-CE6 PACs, using their LAN1 to connect a ET-7050 (Modbus TCP I/O module) through the Ethernet switch.



- Refer the ET-7000 manual to set up the IP address and required settings (refer <u>Section 5.2.1</u>). Manual: <u>http://ftp.icpdas.com/pub/cd/6000cd/napdos/et7000\_et7200/document/</u>
- Open and download the "demo\_RDN\_3" project. Before downloading, set the communication IP (refer <u>Section 2.3.5</u> "Communication Parameters") to the current LAN1 IP of the Active PAC (refer <u>Section 16.4</u> – Test demo programs - step 2 to 4).

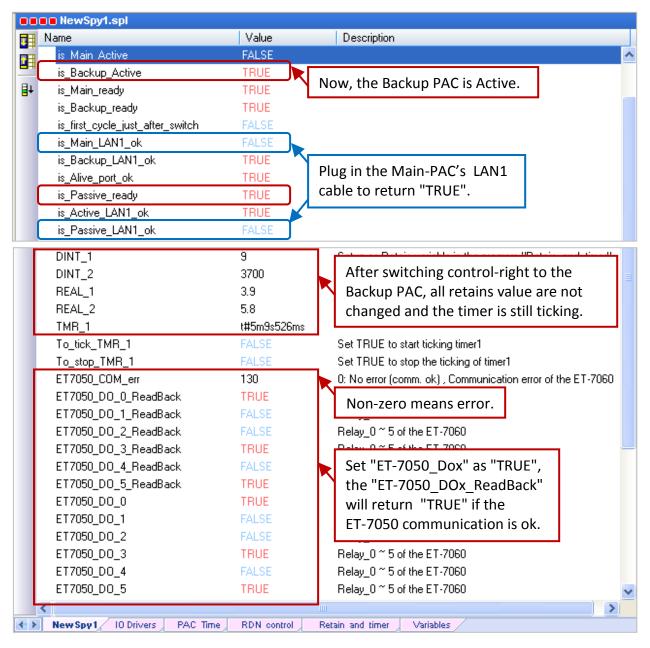
In the "I/O Drivers" window, here we enable a Modbus TCP Master to connect an ET-7050 module (Modbus TCP Slave, Addr. = 1) and create some data blocks to read/write the DI and DO data (refer <u>Section 5.2</u>). Users can open this "demo\_RDN\_3" project for more details.

<u>File E</u> dit <u>V</u> iew Insert Project <u>T</u>	ools	<u>W</u> indow <u>H</u> elp	$\sim$							
29 🖬 🕑 🕼 🗴 🖻 🖻 🗶 프로그는 1 🤊 🕑 🛗 🎹 🏭 ¼ 😨 🖓 🖻 🚱 🔡 🎽 🖉										
Workspace (10 Drivers)										
🖃 🗐 demo_rdn_3	E	🖃 ModBUS Maste	er			To enab	le a Mod	bus TCP N	laster to	
🗄 🔤 Exception programs	뮮	白~ 品 Open MODE	3US: 192.168.	71.128:502		connact	20 ET 70	)50 modul	o and	
🚊 🗝 Programs	*8	😐 – <b>* 📒</b> <2> Rea	ad Input Bits (1	) [112]						
PAC_Time	5.10	🗄 *🛢 <15> W	rite Coil Bits (1)	[16]		create s	ome data	a blocks to	o read/wr	ite 📔 🚽
RDN_control	Ð	🗄*📒 <1> Rea	ad Coil Bits (1)	[16]		tha DI/F	O data	refer the <mark>S</mark>	action 5	2
🔤 Retain_and_timer		🗄 📲 MODBUS Slave	1				o uata,		ection 5.	<u> </u>
🚊 🚞 Watch (for debugging)		📥 📲 Server - Slav	ve number = 1							
Soft Scope		🗄 🗝 📲 Input Re	egisters [1200	0]						
🔤 Initial values	¢۵	-			1	La a a	1	1		
😪 NewSpy1		Request	Slave/Unit	Address	Nb Ite		Period (ms)	Period on error	Timeout (ms)	Number of trials
	∎+	<2> Read Input Bits	1	1	12	Periodic	50	5000	1000	1
🕺 g Global defines		<15> Write Coil Bits	1	1	6	On Change	0	0	1000	1
📸 Variables	122	<1> Read Coil Bits	1	1	6	Periodic	50	5000	1000	1
Types		<								>
	<b>+</b> >	NewSpy1 10 Drive	rs PAC Time	RDN co	introl	Retain and timer	Variables /			

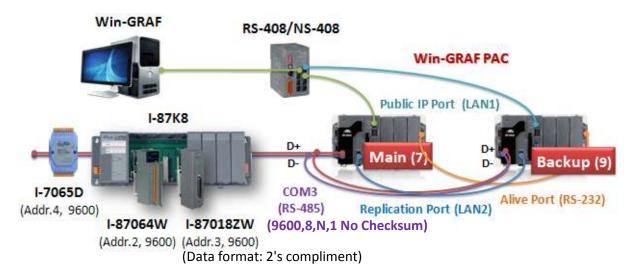
3. Click "NewSpy1" to open the spy list, now the Main-PAC is Active.

Image: Index of the second	2   🖬 🖹   🕘   👗 🗈 🛍	×	🕺 🏷 🖉 🔛 🛣	96 😨 🟠 🖻	5 5g 🕒 📷 🚰 RUN 🛛 0000 🍂 🎆 🛱 💷 省 🧔
Exception programs     Pograms     P	Workspace		NewSpy1.spl		
in: Main_Active       TRUE         in: Main_Active       FALSE         in: Box control       is: Backup_Active         is: Box control       is: Backup_active         is: Active control       is:	🖃 🗠 🕨 demo_rdn_3 (RUN)		Name	Value	Description
is Backup_Active is	🗄 📴 Exception programs	_	is_Main_Active	TRUE	
PDN_control       is_Backup_ready       THUE         Watch (for debuggin)       is_first_cycle_just_after_switch       FALSE         Watch (for debuggin)       is_Main_LAN1_ok       THUE         Backup_Lexit_oxic       is_Backup_LAN1_ok       THUE         Buddingt Configuration       is_Backup_LAN1_ok       THUE         Buddingt Configuration       is_Active_LAN1_ok       THUE         Backup_LAN1_ok       THUE         Is_Active_LAN1_ok       THUE         Backup_LAN1_ok       THUE         is_Passive_ready       TRUE         Variables       DINT_1       1         DINT_2       3000         REAL_1       0.0         Setup as Retain variable in the program "Retain_and_timer"         Setup as Retain variable in the program "Retain_and_timer"         TMR_1       tH0s         To_tick_TMR_11       FALSE         To_tobo_TMR_11       FALSE         ET7050_DO_0_ReadBack       FALSE         ET7050_DO_1_ReadBack       FALSE         ET7050_DO_2_ReadBack       FALSE         ET7050_DO_2_ReadBack       FALSE         ET7050_DO_2_ReadBack       FALSE         ET7050_DO_2_ReadBack       FALSE         ET7050_DO_2_ReadBack <td< th=""><th>📥 🔤 Programs</th><th></th><th>is_Backup_Active</th><th>FALSE</th><th>The Main-PAC is Active.</th></td<>	📥 🔤 Programs		is_Backup_Active	FALSE	The Main-PAC is Active.
<ul> <li> <b>Retain_and_time</b> <b>Wath</b> (for debugging) <b>Solt</b> Scope <b>Solt</b> Scope <b>Solt</b>Scope <b>Solt</b> Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope <b>Solt</b>Scope</li></ul>	PAC_Time	₽÷	is_Main_ready	TRUE	
Watch (for debugging)       is_Mair_LANI_ok       TRUE         Soft Scope       is_Backup_LANI_ok       TRUE         Initial values       is_Backup_LANI_ok       TRUE         Soft Scope       is_Backup_LANI_ok       TRUE         Soft Scope       is_Passive_ready       TRUE         Soft Scope       is_Passive_LANI_ok       TRUE         DINT_1       1       Setup as Retain variable in the program "Retain_and_timer"         Setup as Retain variable in the program "Retain_and_timer"       Setup as Retain variable in the program "Retain_and_timer"         TMH_1       ttdbs       To_tick_TMR_1       FALSE         To_top_TMR_1       FALSE       Set TRUE to stop the ticking of time1         C_ick_TOSO_DO_2_ReadBack       FALSE       Relay_0 ~ 5 of the E1-7060         ET7050_DO_2_ReadBack       FALSE       Relay_0 ~ 5 of the E1-7060         ET7050_DO_2_ReadBack       FALSE       Rela	BDN_control		is_Backup_ready	TRUE	
is_Backup_LAN1_ok       TRUE         is_Alive_pot_ok       TRUE         is_Alive_pot_ok       TRUE         is_Alive_pot_ok       TRUE         is_Alive_pot_ok       TRUE         is_Alive_pot_ok       TRUE         is_Alive_pot_ok       TRUE         is_Adive_LAN1_ok       TRUE         is_Passive_LAN1_ok       TRUE         is_Passive_LAN1_ok       TRUE         is_Passive_LAN1_ok       TRUE         DINT_1       1         DINT_2       3000         REAL_1       0.0         REAL_2       0.0         TMR_1       ttB0s         To_tick_TMR_1       FALSE         Setup as Retain variable in the program "Retain_and_timer"         Setup as Retain variable i	🔤 📷 Retain_and_timer		is_first_cycle_just_after_switch	FALSE	
is_Alive_pot_ok       TRUE         is_Alive_pot_ok       TRUE         is_Passive_ready       TRUE         is_Passive_ready       TRUE         is_Passive_LAN_lok       TRUE         Variables       DINT_1         DINT_2       3000         REAL_1       0.0         REAL_2       0.0         TMR_1       tB0s         To_tick_TMR_1       FALSE         Setup as Retain variable in the program ''Retain_and_timer''         Setup as Retain variable in the program ''Retain	🛓 📴 Watch (for debugging)		is_Main_LAN1_ok	TRUE	
Set Newspol       is_Passive_ready       TRUE         is_disting_configuration       is_assive_ready       TRUE         is_disting_configuration       is_assive_ready       TRUE         is_disting_configuration       is_assive_ready       TRUE         is_assive_ready       TRUE       Intervention         is_disting_configuration       is_assive_ready       TRUE         Variables       DINT_1       1         DINT_2       3000       Setup as Retain variable in the program "Retain_and_timer"         REA_1       0.0       Setup as Retain variable in the program "Retain_and_timer"         REA_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         TMR_1       ttdtds       To_tick_TMR_1       FALSE         To_tick_TMR_11       FALSE       Set TRUE to start ticking timer1         To_stop_TMR_11       FALSE       Set TRUE to start ticking of timer1         To_stop_DO_0_ReadBack       FALSE       Relay_0 ~ 5 of the E1-7060         E17050_DO_1_ReadBack       FALSE       Relay_0 ~ 5 of the E1-7060         E17050_DO_2_ReadBack       FALSE       Relay_0 ~ 5 of the E1-7060         E17050_DO_4_ReadBack       FALSE       Relay_0 ~ 5 of the E1-7060         E17050_DO_1       FALSE       Relay_0 ~ 5 of the E1-	Soft Scope		is_Backup_LAN1_ok	TRUE	It allows to a little DAC as a load sinks
Image Configuration       is_Active_LAN1_ok       TRUE         Image Configuration       is_Active_LAN1_ok       TRUE         Image Configuration       is_Passive_LAN1_ok       Setup as Retain variable in the program "Retain_and_timer"         Image Configuration       is_Passive_LAN1_ok       Setup as Retain variable in the program "Retain_and_timer"         Image Configuration       is_Passive_LAN1_ok       Setup as Retain variable in the program "Retain_and_timer"         Image Configuratio	🖂 Initial values		is_Alive_port_ok	TRUE	
is Passive_LAN1_ok       TRUE         Wariables       DINT_1       1         Setup as Retain variable in the program "Retain_and_timer"       Setup as Retain variable in the program "Retain_and_timer"         REAL_1       0.0       Setup as Retain variable in the program "Retain_and_timer"         REAL_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         To_tick_TMR_1       tdt0s         To_tick_TMR_1       FALSE         Set TRUE to start ticking timer1         To_tick_TMR_1       FALSE         To_tock_TMR_1       FALSE         Set TRUE to start ticking timer1         To_stop_TMR_1       FALSE         RET7050_DO_0_ReadBack       FALSE         RET7050_DO_1_ReadBack       FALSE         RET7050_DO_3_ReadBack       FALSE         RET7050_DO_4_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_DO_4_ReadBack         ET7050_DO_2       Relag_0 ~ 5 of the ET-7060         ET7050_DO_1       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_DO_2       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_DO_2       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_DO_2       FALSE	🦳 😼 NewSpy1 🎍		is_Passive_ready	TRUE	only when the Passive PAC is ready.
Wariables       DINT_1       1       Setup as Retain variable in the program "Retain_and_timer"         ■ Types       DINT_2       3000       Setup as Retain variable in the program "Retain_and_timer"         REAL_1       0.0       Setup as Retain variable in the program "Retain_and_timer"         REAL_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         REAL_1       0.0       Setup as Retain variable in the program "Retain_and_timer"         REAL_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         To_tick_TMR_11       tftbs         To_stop_TMR_11       FALSE         ET7050_D0_0_ReadBack       FALSE         RET7050_D0_1_ReadBack       FALSE         RET7050_D0_2_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_2_ReadBack         ET7050_D0_2_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_4_ReadBack         ET7050_D0_2       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_2         ET7050_D0_2       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_2         ET7050_D0_2       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_2         ET7050_D0_2       FALSE	Dinding Configuration		is_Active_LAN1_ok	TRUE	
DINT_2       3000       Setup as Retain variable in the program "Retain_and_timer"         REAL_1       0.0       Setup as Retain variable in the program "Retain_and_timer"         REAL_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         TMR_1       tttl0s         To_stop_TMR_1       FALSE         Set TRUE to start ticking timer1         To_stop_TMR_1       FALSE         ET7050_DO_O_ReadBack       FALSE         RT050_DO_1_ReadBack       FALSE         RT050_DO_2_ReadBack       FALSE         RT050_DO_2_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_DO_3_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_DO_4_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_DO_4_ReadBack         ET7050_DO_4_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_DO_5_ReadBack         ET7050_DO_2_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_DO_4_ReadBack         ET7050_DO_4_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_DO_4         ET7050_DO_2       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_DO_4         ET7050_D	💮 § g Global defines		is_Passive_LAN1_ok	TRUE	
REAL_1       0.0       Setup as Retain variable in the program "Retain_and_timer"         REAL_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         TMR_1       ttt0s         To_tick_TMR_1       FALSE         Set TRUE to start ticking timer1         To_stop_TMR_1       FALSE         Set TRUE to start ticking timer1         To_stop_TMR_1       FALSE         Set TRUE to start ticking timer1         To_stop_TMR_err       0         CT7050_D0_ReadBack       FALSE         RET7050_D0_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_ReadBack         ET7050_D0_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_ReadBack         ET7050_D0_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_ReadBack         ET7050_D0_ReadBack       FALSE         Relay_0 ~ 5 of the ET-7060       ET7050_D0_ReadBack         ET7050_D0_0       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_D0_1       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_D0_2       FALSE         Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE         Relay_0	🔤 📸 Variables		DINT_1	1	Setup as Retain variable in the program "Retain_and_timer"
REAL_2       0.0       Setup as Retain variable in the program "Retain_and_timer"         TMR_1       t#0s         To_tick_TMR_11       FALSE       Set TRUE to start ticking timer1         To_stop_TMR_11       FALSE       Set TRUE to stop the ticking of timer1         ET7050_C0M_err       0       Ox oerror (comm. ok), Communication error of the ET-7060         ET7050_00_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_00_ReadBack       FALSE	Types		DINT_2	3000	Setup as Retain variable in the program "Retain_and_timer" 🛛 📄
TMR_1t#0sTo_tick_TMR_1FALSESet TRUE to start ticking timer1To_stop_TMR_1FALSESet TRUE to stop the ticking of timer1ET7050_COM_err00: No error (comm. ok), Communication error of the ET-7060ET7050_D0_0_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060			REAL_1	0.0	Setup as Retain variable in the program "Retain_and_timer"
To_tick_TMR_1FALSESet TRUE to start ticking timer1To_stop_TMR_1FALSESet TRUE to stop the ticking of timer1ET7050_COM_err00: No error (comm. ok), Communication error of the ET-7060ET7050_D0_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_S_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060			REAL_2	0.0	Setup as Retain variable in the program "Retain_and_timer"
To_stop_TMR_1       FALSE       Set TRUE to stop the ticking of timer1         ET7050_C0M_err       0       No error (comm. ok), Communication error of the ET-7060         ET7050_D0_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_1_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_0       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_1       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_1       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_05       FALSE       R			TMB_1	t#Os	
ET7050_COM_err00: No error (comm. ok), Communication error of the ET-7060ET7050_D0_0_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_0FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060			To_tick_TMR_1	FALSE	Set TRUE to start ticking timer1
ET7050_D0_0_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_0FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5<			To_stop_TMR_1	FALSE	Set TRUE to stop the ticking of timer1
ET7050_D0_1_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_0FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060			ET7050_COM_err	0	0: No error (comm. ok) , Communication error of the ET-7060
ET7050_D0_2_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_0FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060			ET7050_DO_0_ReadBack	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_3_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5_ReadBackFALSERelay_0 ~ 5 of the ET-7060ET7050_D0_0FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_1FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_2FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_3FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_4FALSERelay_0 ~ 5 of the ET-7060ET7050_D0_5FALSERelay_0 ~ 5 of the ET-7060			ET7050_DO_1_ReadBack	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_4_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_0       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_1       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060			ET7050_DO_2_ReadBack	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_5_ReadBack       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_0       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_1       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060			ET7050_D0_3_ReadBack	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_0       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_1       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060			ET7050_DO_4_ReadBack	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_1       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_2       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060			ET7050_DO_5_ReadBack	FALSE	Relay_0 ∼ 5 of the ET-7060
ET7050_D0_2       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_3       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_4       FALSE       Relay_0 ~ 5 of the ET-7060         ET7050_D0_5       FALSE       Relay_0 ~ 5 of the ET-7060			ET7050_DO_0	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_3         FALSE         Relay_0 ~ 5 of the ET-7060           ET7050_D0_4         FALSE         Relay_0 ~ 5 of the ET-7060           ET7050_D0_5         FALSE         Relay_0 ~ 5 of the ET-7060			ET7050_DO_1	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_4         FALSE         Relay_0 ~ 5 of the ET-7060           ET7050_D0_5         FALSE         Relay_0 ~ 5 of the ET-7060			ET7050_D0_2	FALSE	Relay_0 ~ 5 of the ET-7060
ET7050_D0_4         FALSE         Relay_0 ~ 5 of the ET-7060           ET7050_D0_5         FALSE         Relay_0 ~ 5 of the ET-7060			ET7050_DO_3	FALSE	Relay_0 ~ 5 of the ET-7060
				FALSE	Relay_0 ~ 5 of the ET-7060
			ET7050_D0_5	FALSE	Relay_0 ~ 5 of the ET-7060
			<pre></pre>		
New Spy 1/ 10 Drivers PAC Time RDN control Retain and timer Variables		< >	New Spy 1 10 Drivers PAC Time	RDN control	Retain and timer Variables

4. Enter some values for these retain variables (DINT\_1, DINT\_2, REAL\_1 and REAL\_2) and set "To\_tick\_TMR\_1" as "TRUE" to start Timer. If set "ET-7050\_Dox" as "TRUE", the "ET-7050\_DOx\_ReadBack" will return "TRUE". If disconnect the Ethernet cable from the ET-7050 module, the "ET-7050\_COM\_error" will return a non-zero value that means communication error. 5. Make sure the Passive PAC is ready (i.e., "is\_Passive\_Ready" is "TRUE"), remove the LAN1 cable from the Main-PAC or turn it off and on (restart), and wait a short time (refer the "RDN\_control" program), the Main-PAC will automatically reboot and give control-right to the Backup-PAC.



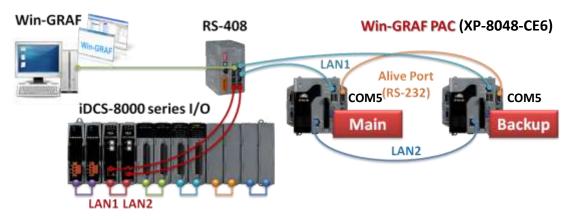
demo\_RDN\_1: Two XP-8xx8-CE6 PACs, using their COM3 to connect three DCON I/O modules.



In the "demo\_RDN\_1" demo project (as the figure above), before linking "I-7000" and "I-87KW" DCON remote I/O modules, users need to configure each of I/O modules by using "DCON Utility". Refer the description in <u>Chapter 8</u> and visit the "DCON Utility" web page to download the software and user manual: <u>www.icpdas.com/products/dcon/introduction.htm</u>. Please restore and open this demo project for more details and refer <u>Section 16.3.1</u> for "I/O Board" settings.

# 16.4.2 Test the "demo\_RDN\_4" Project

Two XP-8xx8-CE6 PACs, using their LAN1 to connect an iDCS-8830 (Both PAC and I/O are redundant) through one Ethernet switch.



The following table lists all used devices in this project:

Products	Quantity	Products	Quantity
XP-8048-CE6	2	DN-DI-32DW	1
RS-408	1	DN-DO-16DR-A	1
iDCS-8830	1	DN-DO-16DR-B	1
F-8040	2	CA-37 <b>10</b> AM (1M Cable) or	4
F-8041	2	CA-37 <b>20</b> AM <b>/30</b> AM <b>/50</b> AM <b>/100</b> AM	4

## Hardware Wiring and Software Setting:

You can refer <u>Section 16.1</u> and <u>Section 16.2</u> to view the Win-GRAF redundant system (4) and the wiring way of two XP-8048-CE6 PACs. This iDCS-8830 is installed with the following I/O modules (from left to right). First, download and install the related utility and user manual.

iDCS-8000 Utility: <u>http://ftp.icpdas.com/pub/cd/idcs-8000/utility/</u> (Software Installation: Ch2.2) MiniOS7 Utility: <u>http://ftp.icpdas.com/pub/cd/8000cd/napdos/minios7/utility/</u>

iDCS-8000 User Manual/Website:

http://ftp.icpdas.com/pub/cd/idcs-8000/usersmanual/

http://www.icpdas.com/root/product/solutions/remote io/dcs redundancy io/idcs introduction.html

**FPM-D2440 \* 2 :** Power module1, Power module2 (for power input, 24V).

## FCM-MTCP \* 2 :

MCU1 (First, rotate the SW2 to "C" and the SW1 to "8", then set up the IP address to 192.168.71.**200**). MCU2 (First rotate the SW2 to "C", the SW1 to "9", and then set the IP address to 192.168.71.**201**). The SW2/SW1 means the fourth IP address of the MCU (Main Control Unit). (C8<sub>16</sub> = 200; C9<sub>16</sub> = 201) Refer the iDCS-8000 user manual (Ch2.3 or the figure below) to set IP addresses by using the "MiniOS7 Utility". Run the "MiniOS7 Utility", click "Search" to search current IP addresses of the iDCS-8830 (i.e., MCU1/2), and click "Stop" afterwards. Then, change these IP addresses to "192.168.71.200" and "192.168.71.201", change the Mask to "255.255.0.0" and click "Set". Finally, close the "MiniOS7 Utility".

Image: State of the state	🖬 ICPDAS 🖮 WinZip		<ul> <li>MiniO\$</li> <li>iDCS-8</li> </ul>	87 Utility Ver 3 3000			7 Utility Ver 3 7 Utility Ver 3	
Look In:         Last Connection         Alt+F2           Disconnect         Ctrl+F2         Size Type           Size Type         Size Type           <	DAQPro		🕨 💼 Modbu	us U tilit	File 🌔 Connec	tion	🍌 Commar	ud 🛐 Configuration 📑
Stop         Options         Connect         Clear         IP setting         Image: Figure Fig	MiniOS7 Scan			Name	e <u>D</u> iscom	nnectio: nect	n Alt+F2 Ctrl+F2	
TCP BroadCast       192.168.11.10       Web       255.255.0.0       19       IP:       192.168.71.200         TCP BroadCast       192.168.71.200       FCM-MTCP       iDCS-8830       255.255.0.0       19       Mask:       255.255.0.0         TCP BroadCast       192.168.71.200       FCM-MTCP       iDCS-8830       255.255.0.0       19       Mask:       255.255.0.0         TCP BroadCast       192.168.71.201       FCM-MTCP       iDCS-8830       255.255.0.0       19       Gateway:       192.168.0.1         TCP BroadCast       192.168.71.201       FCM-MTCP       iDCS-8830       255.255.0.0       19       Gateway:       192.168.0.1         TCP BroadCast       Select it and click "IP setting".       255.255.0.0       19       Alias:       iDCS-8830         TCP BroadCast       192.168.11.7       WISE-5800       255.255.0.0       19       DHCP	Stop Untiene	📡 놀	<u>L</u> ?				$\mathbf{O}$	
TCP BroadCast       192.158.71.200       FCM-MTCP       IDCS-8830       255.255.0.0       19         TCP BroadCast       192.168.71.201       FCM-MTCP       IDCS-8830       255.255.0.0       19         TCP BroadCast       192.168.71.201       FCM-MTCP       IDCS-8830       255.255.0.0       19         TCP BroadCast       192.168.71.201       FCM-MTCP       IDCS-8830       255.255.0.0       19         TCP BroadCast       192.168.01       IDCS-8830       255.255.0.0       19         TCP BroadCast       Select it and click "IP setting".       255.255.0.0       19         TCP BroadCast       192.168.11.7       VI-91000       255.255.0.0       19         TCP BroadCast       192.168.11.7       WISE-5800       255.255.0.0       19         TCP BroadCast       192.168.11.7       WISE-5800       255.255.0.0       19			N	-	Mask			
TCP BroadCast 132.100.70.02 VI 410X VI 410X0 255.255.0.0 19 TCP BroadCast 192.168.11.7 VISE-5800 VISE-5800 255.255.0.0 19	3 adCast TCP BroadCast TCP BroadCast		N 4 Web .10	Alias EtherIO test1	255.255.0.0 255.255.0.0 255.255.0.0	Ga 19 19 19	⊢Recommend	Settings
- Disable - Eliable	3 adCast TCP BroadCast TCP BroadCast TCP BroadCast TCP BroadCast TCP BroadCast	IP/Port           192.168.11.9           192.168.11.10           192.168.11.8           192.168.71.200           192.168.71.201	N C 4 Web .10 FCM-MTCP FCM-MTCP	Alias EtherIO test1 iDCS-8830 iDCS-8830	255.255.0.0 255.255.0.0 255.255.0.0 255.255.0.0 255.255.0.0 255.255.0.0	Ga 19 19 19 19 19 19	⊂Recommend IP: Mask: Gateway:	Settings

### F-8040 \* 2 :

32-channel DI modules, plug them into the I/O slot0 and slot1.

### F-8041 \* 2:

32-channel DO modules, plug them into the I/O slot2 and slot3.

Refer the iDCS-8000 user manual (Ch2.3 or the figure below) to configure the I/O module by using the iDCS-8000 Utility.

Cancel

Set

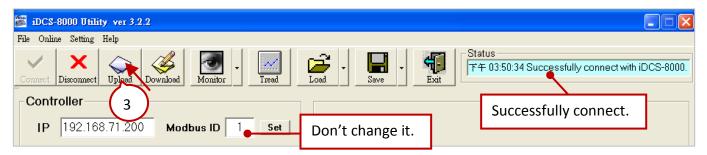
5

Run the "iDCS-8000 Utility" and fill in the IP address of the iDCS-8830 to connect.

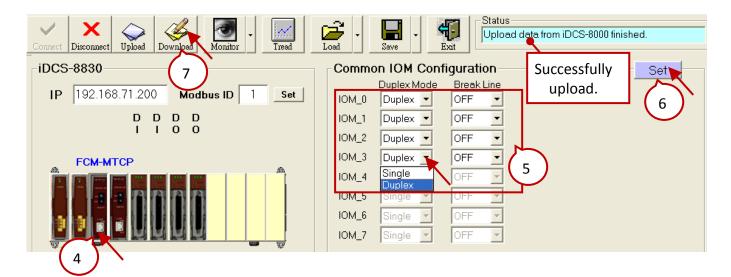
	B iDCS-8000 Utility ver 3.2.2						
MBus	File Online Setting Help						
iDCS-8000 Utility	Connect is 2 Upload Download Monitor						
N	Controller						
	IP 192.168.71.200 1 Vour PC and the iDCS-8830 must in the same network segment (e.g., 192.168.71.99 and 192.168.71.200) to connect successfully.						

Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 16-24

Click "Upload" to upload the current settings of I/O modules that plugged into the iDCS-8830.



Click the 1st FCM-MTCP (MCU1) and set "IOM\_0 ~ 3" (F-8040/F-8041) as "Duplex" Mode (If the F-8040/ F-8041 module isn't connected with the following termination boards now, set the "Break Line" as "OFF") and click "Set". Then, click "Download" to download the above settings to the iDCS-8830. Finally, close the "iDCS-8000 Utility".



If you have connected	d the following termination boards, set the "Break Line" as "ON" (the figure above).
DN-DI-32DW * 1 :	32-channel DI termination board.

	Connect it to the F-8040 DI module (slot0, slot1).
DN-DO-16DR-A * 1 :	16-channel DO termination board (for channel 0 $\sim$ 15).
DN-DO-16DR-B * 1 :	16-channel DO termination board (for channel 16 ~ 31).
	Connect the DN-DO-16DR-A (CN1, CN2) to the F-8041 DO module (slot2, slot3),
	and then connect the DN-DO-16DR-A (CN3) to the DN-DO-16DR-B (CN1).
CA-3710AM * 4 :	1m 37-pin Male-Female D-sub Cable.

#### It used to connect the I/O module and the termination board.

### Start testing:

First, the user must go to <u>Section 16.4.1</u> to know how to set the communication IP address, download the Win-GRAF project ("demo\_RDN\_4") and test the redundant PAC system. The following descriptions will focus on the iDCS-8830. Before starting the test, make sure all devices are connected properly and then run the Win-GRAF Workbench to connect to the PAC.

Note: Make sure both of IP addresses of your PC and PAC are on the same network segment. (e.g., PAC IP = **192.168.71**.37, Mask = **255.255.255**.0, set your PC IP = **192.168.71**.x) After connecting to the PAC, click "New Spy1" to open the spy list.

	In the Herrisoy Lape		
demo_rdn_4 [HUN]	1 Name	Value:	Devolption
and a second sec	Hou!	11	
Programe	Minute1	4	
10 DD_demo	il+ Second1		
PAC_Time	it_Main_Active	TRE	
- DN_control	iz_Backup_Active		
ReadMe ("Enly c	a_Man_ready	FEJE	
H Retain_and_timer	u_Backup_ready	TPUJE	
Watch (for debugging)	it_fist_cycle_put_after_switch		
Soft Scope	it_Main_LAN1_ck	TECH	
The last of solutions	is_Backup_LAN1_ok	TFILIE	
'&J NewSpy1	n_Alive_port_ok	TRUE	
Da Finding Contry Moon	it_Passive_neady	TRUE	
3g Global defines	is_Active_LAN1_of	THUE	
Set II College	ve_LAN1_ok	TRUE	
Click "New Sp	NUT II RILANT_envir_itatui		Value: 0, ok - non sero : ensi: LAN ensi: halus of the 1xt FCH MTCP
THER MEN SH		0	Value: @ ok , non-pero : error. LAN error status of the 2nd FDM-MTCP
	80_FCM1_wode	32	Value: 1640020 Marter, 1640021: Slave, Modbur start-add = 513. Redundant node of the 1st FDM M1CP.
	DCS8830_FCM2_mode	33	Value: 16#0020   Master : 16#0021: Sleve, Modbur; stat-adult = 513, Redundant mode of the 2nd FD4-MTCP
	DC58830_FDMT_status	16	Value: 1E#0000 Enply, 1E#0001 Timeout, 1E#0002 Undefined, 1E#0010: Normal Modbut-addr = 514, FDM1 status.
	@CS8830_FOH2_stelus	16	Value: 16#0000 Empty. 16#0001 Timeout, 16#0002 Undefined, 16#0010 Normal Modous edd: + 514, FCH2 status
	DCS9800_FOV2_status DCS9830_FPM_status	16 3	Value: 1640000. Enoly., 1640001 Timeout, 1640002 Undefined, 1680010 Normal Modbus eddr. = 514, FCM2 status. Value: 0x0000; No FPM plagged, 0x0001: FPM1 Good / FPM2 Off, 0x0002; FPM1 0X / FPM2 Good, 0x0003 Two FPM are Good. Modbus eddr. = 515
	DCS8800_FOM2_status DCS8830_FPM_status DCS8830_System_minor_tauk_status	16 3 0	Value: 1680000. Enoly. 1680001 Timeout, 1680002 Undefined, 1680010 Normal Modbus addr = 514, FCM2 status. Value: 0x0003 No FPM plugged, 0x0001 FPM1 Good / FPM2 Ott, 0x0002 FPM1 0X / FPM2 Good, 0x0003 Two FPM are Good. Nodeus addr = 519 Value: 0. normalnoncerx. consetting wong (heter the IDCS-6800 software narrust). Modbus-addr = 517 / 518. System minor fault status.
	DCS8800_FOX2_status DCS8830_FPM_status DCS8830_System_minor_twik_status DCS8830_System_major_twik_status	16 3	Value: 1580000 Engly, 1580001 Taxeout, 1580002 Undefined, 1580010 Normal Modbus edds = 514, PCH2 status; Value: 0x0003: No FPM plagged, 0x0001: FPMI Good / FPMI C01; 0x0002; FPMI D17 / FPM2 Socid, 0x0003 Two FPM are Good. Nodbus edds = 514 Value: 0: normalnonzer: constituing woong/intel file/IDC3-8000 software namual; Modbus-edds = 519 / 518, System maior fault status; Value: 0: normalnonzer: constituing woong/intel file/IDC3-8000 software namual; Modbus-edds = 519 / 518, System major fault status;
	DCSR00_FOX2_status DCSR00_FPM_status DCSR00_FPM_status DCSR00_Fpsters_mice_taut_status DCSR00_ic_stot_status	16 3 0 0	Value: 16#0000 Engly, 16#0001 Taxeout, 16#0002 Undefined, 16#0010 Normal Modbus addr = 514, FCH2 status: Value: 0x0000: No FPM plagged, 0x0001: FPMI Good / FPMI2 011, 0x0002; FPMI 017 / FPM2 Good, 0x0000 Two FPM are Good, Modbus addr = 519 Value: 0. normalnon.cerx.conething wrong before the IDCS-8000 underse narwall, Modbus-addr = 517 / 518. System mator fault status. Value: 0. normalnon.cerx.conething wrong before the IDCS-8000 underse narwall, Modbus-addr = 517 / 518. System mator fault status. Value: 0. normalnon.cerx.conething wrong before the IDCS-8000 underse narwall, Modbus-addr = 517 / 518. System mator fault status. Value: 0. normalnon.cerx.conething wrong before the IDCS-8000 underse narwall, Modbus-addr = 517 / 518. System mator fault status. Value: 16#0003: Engly, 16#0002; Hait, 16#0004; Booku, 16#0000; Bookoader, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop; Modbu
	DC58000_FCH2_status DC58830_FFM_tiskus DC58830_System_mixic_lauk_status DC58830_System_mixic_lauk_status DC58830_in_status DC58830_in_status	16 3 0 0 2	Value: 16#0000: Enety:, 16#0001 Timeout, 16#0002 Undefined, 16#0010 Normal Modbus addr = 514, FCM2 status: Value: 0.0000: No FPM plugged, 0.0001: FPMI Good / FPM2 011, 0.00022 FPM1 017 / FPM2 Good, 0.60003 Two FPM are Good Modbus addr = 519 Value: 0.normal:.non-zero: contenting wong bette the IDCS-8000 schware narwall, Modbus-addr = 517 / 520. System more fasik status: Value: 0.normal:.non-zero: contenting wong bette the IDCS-8000 schware narwall, Modbus-addr = 517 / 520. System more fasik status: Value: 0.normal:.non-zero: contenting wong bette the IDCS-8000 schware narwall, Modbus-addr = 519 / 520. System more fasik status: Value: 16#0001: Enety:, 16#0002: Hait.1##0004: Bootu, 16#0008: Bootoader.16#0010: Pre-operation, 16#0020: Dpetation, 16#00040. Stop. Modbus Value: 16#0001: Enety:, 16#0002: Hait.1\$#0004 Bootup, 16#0008: Bootoader, 16#0010: Pre-operation, 16#0020: Dpetation, 16#00040. Stop. Modbus
	DC58030_FOH2_status DC58030_FOH2_status DC58030_5ystem_white_balk_status DC58030_5ystem_white_balk_status DC58030_jo_jot_status DC58830_jot_statustatu0 DC58830_jot_stat_status100	16 3 0 0 32 32	Value: 1680000: Enety:, 1680001 Timeout, 1680002: Ucdefined, 1680010: Normal Modbus edds = 514, FCM2 status: Value: 0x0000: No FPM plugged, 0x0001: FPM1 Good / FPM2 Dtr. 0x0002: FPM1 0X / FPM2 Good, 0x00003: Two FPM are Good, Modbus edds = 519 Value: 0x0000: No FPM plugged, 0x0001: FPM1 Good / FPM2 Dtr. 0x0002: FPM1 0X / FPM2 Good, 0x0003: Two FPM are Good, Modbus edds = 519 Value: 0x0000: No FPM plugged, 0x0001: FPM1 Good / FPM2 Dtr. 0x0002: FPM1 0X / FPM2 Good, 0x0003: Two FPM are Good, Modbus edds = 519 Value: 0x0001: Enety:, 1680002: Hait, 1680000: Bootup, 1680008: Bootoaler, 1680010: Pre-operation, 1680000: Stop, Modbus Value: 1680001: Enety:, 1680002: Hait, 1680000: Bootup, 1680008: Bootoaler, 1680010: Pre-operation, 1680000: Stop, Modbus Value: 1680001: Enety:, 1680002: Hait, 1680000: Bootup, 1680008: Bootoaler, 1680010: Pre-operation, 1680000: Stop, Modbus Value: 1680001: Enety:, 1680002: Hait, 1680000: Bootup, 1680008: Bootoaler, 1680010: Pre-operation, 1680000: Stop, Modbus Value: 1680001: Enety:, 1680002: Hait, 1680000: Bootup, 1680008: Bootoaler, 1680010: Pre-operation, 1680000: Departion, 16800000; Stop, Modbus
	DC58030_FCH2_status DC58830_FPH_status DC58830_System_mixp_ludi_status DC58830_System_mixp_ludi_status DC58830_stat_status DC58830_stat_status(0) DC58830_js_stat_status(0) DC58830_js_stat_status(2)	16 3 0 0 22 22 22 22 22	Value: 16#0000 Engly, 16#0001 Tareout, 16#0002 Undefined, 16#0010 Normal Modous addr = 514, FCH2 status; Value: 0x0000: No FPM plagged, 0x0001: FPMI Good / FPMI C01; 0x0002; FPMI 0X/ FPMI 25x04; 0x0000 Two FPM are Good. Modous addr = 514 Value: 0x0000: non-percentaring wong before the IOCS-6000 software narwall, Modous-addr = 517 / 518; System major lauk status; Value: 0x0000: Engly, 16#0001; Hatt 16#0000; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0200; Operation; 16#0000; Stop, Modou Value: 16#0000; Engly, 16#0001; Hatt 16#0000; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0004; Stop, Modou Value: 16#0000; Engly, 16#0002; Hatt 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0040; Stop, Modou Value: 16#0000; Engly, 16#0002; Hatt, 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hatt, 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0004; Stop, Modou Value: 16#0001; Engly, 16#0002; Hatt, 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0004; Stop, Modou Value: 16#0001; Engly, 16#0002; Hatt, 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hatt, 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hatt, 16#0004; Bootup; 16#0008; Bootoader; 16#0010; Pre-operation; 16#0020; Operation; 16#0040; Stop, Modou
	DC58030_FOH2_status DC58830_FPM_status DC58830_System_mice_tuid_status DC58830_System_mice_tuid_status DC58830_in_status	16 3 0 0 2 2 2 2 2 2 2 2 2 2 2	Value: 1640000: Engly:, 1640001: Taneout, 1640002: Undefined, 1680010: Normal Modbus addr = 514, FCH2 status: Value: 0x0000: No FEM plagged, 0x0001: FFMI Good / FEMI2 011, 0x0002: FFMI 017 / FEM2 Good. 0x00001 Two FEM are Good. Modbus addr = 514 Value: 0: normal, non-next: conserting wong before the ICCS-8800 software narwall, Modbus-addr = 517 / 518. System narios fault status. Value: 0: normal, non-next: conserting wong before the ICCS-8800 software narwall, Modbus-addr = 517 / 518. System narios fault status. Value: 0: normal, non-next: conserting wong before the ICCS-8800 software narwall, Modbus-addr = 517 / 518. System narios fault status. Value: 1640001: Engly: 1640002: Hait, 1640004: Bootup, 1640008: Bootloader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbu Value: 1640001: Engly: 1640002: Hait, 1640004: Bootup, 1640008: Bootloader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbu Value: 1640001: Engly: 1640002: Hait, 1640004: Bootup, 1640008: Bootloader, 1640010: Pre-operation, 1640040: Operation, 1640040: Stop, Modbu Value: 1640001: Engly: 1640002: Hait, 1640004: Bootup, 1640008: Bootloader, 1640010: Pre-operation, 1640040: Stop, Modbu Value: 1640001: Engly: 1640002: Hait, 1640004: Bootup, 1640008: Bootloader, 1640010: Pre-operation, 1640040: Stop, Modbu Value: 1640001: Engly: 1640002: Hait, 1540004: Bootup, 1640008: Bootloader, 1640010: Pre-operation, 1640040: Stop, Modbu
	DC58000_FOH2_status DC58800_FPM_tilatus DC58800_System_mino_tilatus DC58800_System_mino_tilatus DC58800_in_status DC58800_in_status DC58800_in_status DC58800_in_status DC58800_in_status DC58800_in_status DC58800_in_status DC58800_in_status DC58800_in_status	16 3 0 0 22 22 22 22 22 22 1	Value: 16#0000: Energy:, 16#0001 Taneout, 16#0002: Undefined, 16#0010: Normal Modbus addr = 514, FCM2 status: Value: 0x0000: No FPM plugged, 0x0001: FPMI Good / FPM2 0th; 0x0002: FPMI 01/ / FPM2 Good, 0x0000 Two FPM are Good Modbus addr = 518 Value: 0 normal: non-zero: contenting wong breat the IDCS-8600 undease namual; Modbus-addr = 517 / 520. System mator fault status: Value: 0 normal: non-zero: contenting wong breat the IDCS-8600 undease namual; Modbus-addr = 517 / 520. System mator fault status: Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640020: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 1640000: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 16400040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 16400040: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 16400040: Operation, 1640040: Stop, Modbus Value: 1640001: Energy: 1640002: Hait, 1640004: Bootup, 1640008: Bootoader, 1640010: Pre-operation, 16400040: Operation, 1640040: Stop, Mo
	DC58030_FOH2_status DC58803_FOH2_status DC58803_System_mice_laul_status DC58803_system_mice_laul_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_is_ide_status DC58803_ide_status DC5	16 3 0 0 22 22 22 1 1	Value: 16#0000 Engly, 16#0001 Tareout, 16#0002 Undefined, 16#0010 Normal Modous add: = 514, FCH2 status; Value: 0.normal, non-text: consetting wong/jetet the /002-0000 software narrowal, Modous-add: = 517, 7518, System major lauk status; Value: 0.normal, non-text: consetting wong/jetet the /002-0000 software narrowal, Modous-add: = 517, 7518, System major lauk status; Value: 16#0001; Engly, 16#0002; Hat, 16#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 16#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 15#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 15#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 15#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 15#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou Value: 16#0001; Engly, 16#0002; Hat, 15#0004; Bootup, 16#0008; Bootoaker, 16#0010; Pre-operation, 16#0020; Operation, 16#0040; Stop, Modou
	DC58030_FOH2_status DC58830_FPM_status DC58830_System_maps_lauk_status DC58830_System_maps_lauk_status DC58830_in_status DC58830_in_status DC58830_in_stat_status DC58830_in_stat_status DC58830_in_stat_status DC58830_in_stat_status DC58830_in_stat_status DC58830_in_statustat[] DC58830_in_statustat[] DC58830_in_statustat[]	16 300 22 22 22 22 1 1	Value: 1680000. Empty. 1680001 Tamoout, 1680002: Undefined, 1680010 Normal Modous adds = 514, FCH2 status. Value: Du0000. No FPM plugged, Du001: FPMI Good / FPMI2 Dr. Du0002: FPMI 10V / FPM2 Scod, Du003 Two FPM are Scod. Modius adds = 516 Value: D normal, non-zero: conserting weing before the DCS-8800 software manual, Modous-adds = 517 / 518. System major fault status. Value: D normal, non-zero: conserting weing before the DCS-8800 software manual, Modous-adds = 519 / 520. System major fault status. Value: D normal, non-zero: conserting weing before the DCS-8800 software manual, Modous-adds = 519 / 520. System major fault status. Value: 1680001: Empty. 1680002: Hait. 1680008: Boolug. 1680008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop. Modou Value: 1680001: Empty. 1680002: Hait. 1680008: Boolug. 1680008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop. Modou Value: 1680001: Empty. 1680002: Hait. 1680008: Boolug. 1680008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop. Modou Value: 1680001: Empty. 1680002: Hait. 1680008: Boolug. 1680008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop. Modou Value: 1680001: Empty. 1680002: Hait. 1680008: Boolug. 1680008: Booloader, 1680010: Pre-operation, 1680040; Osop. Modou Value: 1680001: Empty. 168002: Hait. 1680008: Boolug. 1680088: Booloader, 1680010: Pre-operation, 1680040; Stop. Modou Value: 1680001: Empty. 168002: Hait. 1680008: Boolug. 1680088: Boolug. 1680010: Pre-operation, 1680040; Stop. Modou Value: 1680001: Empty. 168002: Hait. 1680008: Boolug. 1680088: Boolug. 1680010: Pre-operation, 1680040; Stop. Modou Value: 1680001: Empty. 168002: Hait. 1680008: Boolug. 1680008: Boolug. 1680010; Pre-operation, 1680040; Stop. Modou Value: 1680001: Empty. 168002: Hait. 1680008: Boolug. 1680008; Boolug. 1680000; Depretion, 1680040; Stop. Modou Value: 1680001: Empty. 168002: Hait. 1680008: Boolug. 1680008; Boolug. 1680000; Depretion, 1680040; Stop. Modou Value: 1680001: Empty. 1680002; Hait.
	DC58830_FOH2_status DC58830_FOH_status DC58830_FoH_status DC58830_in_status DC58830_	16 3 0 0 22 22 22 1 1	Value: 1680000. Empty., 1680001 Tameout, 1680002: Undefined, 1680010 Normal Modous adds = 514, FCH2 status. Value: Du0000, No FPM plugged, Du0001: FPMI Good / FPMI CDL: 0x0002: FPMI DV / FPM2 Good, Du0003 Two FPM and Good Modous adds = 516 Value: Dinomal, non-nexe: cometing weing befer the DCS-6800 software narwall, Modous-adds = 517, 7510. System major fault status. Value: Dinomal, non-nexe: cometing weing befer the DCS-6800 software narwall, Modous-adds = 519, 7520. System major fault status. Value: 1680001: Empty., 1680002: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680002: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680022: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1680040; Stop, Modbu Value: 1680001: Empty., 1680002: Hait, 1580008: Boolug, 1580008: Booloader, 1680010: Pre-operation, 1580020: Operation, 1580040; Stop, M
	DC58030_FOH2_status DC58030_FOH2_status DC58030_System_neign_lauk_status DC58030_system_neign_lauk_status DC58030_in_idit_status[0] DC58030_in_idit_status[0] DC58030_in_idit_status[0] DC58030_in_idit_status[2] DC58030_in_idit_status[2] DC58030_in_idit_status[3] DC58030_in_idit_status[5] DC58030_in_idit_status[5] DC58030_in_idit_status[5] DC58030_in_idit_status[5] DC58030_in_idit_status[6] DC58030_in_idit_status[6]	16 3 0 0 22 22 22 22 22 1 1 1 1	Value: 1680000 Empty. 1680001 Tareout, 1680002 Undefined, 1680010 Normal Modous add: = 514, FCH2 status; Value: 0.0003: No FPM plagged, 0.0001: FPMI Good/ FPMIC 01; 0.00022 FPMI 01// FPM2 Good. 0.0003 Two FPM are Good. Modous add: = 518 Value: 0. normal, non-zero: concelling wong/infect file.002-8000 undersen narvaal, Modous-add: = 519 / 720; System major lauk status; Value: 1680001: Empty. 1680002 Hat, 1680004 Bootup, 1680008 Bootloader, 1680010 Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680004 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680004 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680004 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1680008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1180008 Bootup, 1680008 Bootloader, 1680010; Pre-operation, 1680020 Operation, 1680040 Stop, Modou Value: 1680001: Empty. 1680002 Hat, 1180008 Bootup, 1680
	DC58830_FOH2_status DC58830_FOH_status DC58830_FoH_status DC58830_in_status DC58830_	16 300 22 22 22 22 1 1	Value: 1640000: Energi, 1640001: Taneout, 1640002: Undefined, 1680010: Normal Modous addr = 514, FCM2 status: Value: 0.40000: No FEM plagged, 0.40001: FFM1 Good / FEM2 011, 0.40002; FEM1 017 / FEM2 Good, 0.400003 Two FEM are Good. Modous addr = 511 Value: 0.40000: No FEM plagged, 0.40001: FFM1 Good / FEM2 011, 0.40002; FEM1 017 / FEM2 Good, 0.400003 Two FEM are Good. Modous addr = 511 Value: 0.40001: Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640002; Operation, 1640040; Stop, Modou Value: 1640001: Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640020; Operation, 1640040; Stop, Modou Value: 1640001: Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640020; Operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1640010; Pre-operation, 1640020; Operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloade, 1660010; Pre-operation, 1640020; Operation, 1640040; Stop, Modou Value: 1640001; Energi, 1640002; Hait, 1640004; Bootu, 1640000; Bootloader, 1660010; Pr
	DC58030_FOH2_status DC58030_FOH2_status DC58030_5ystem_nvino_jund_status DC58030_5ystem_nvino_jund_status DC58030_in_std_status	16 3 0 0 22 22 22 22 22 1 1 1 1	Value: 1680000: Energi, 1680001: Tansout, 1680002: Undefined, 1680010: Normal Modous add: = 514, FCM2 status: Value: 0.40000: No FPM plaggad, 0.40001: FPMI Good / FPMI 2011: 404002: FPMI 10/1 / FPMI 26ard, 8.40003 Two FPM are Good Modous add: = 517 Value: 0. normalnonextsomething wong before the ICS-58000 software narwall, Modous-add: = 517 / 5100. System major laad status. Value: 0. normalnonextsomething wong before the ICS-58000 software narwall, Modous-add: = 517 / 5100. System major laad status. Value: 1680001: Energi, 1680002; Hait.1680004; Bootup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001: Energi, 1680002; Hait.1680004; Bootup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001: Energi, 1680002; Hait.1680004; Bootup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait.1680004; Bootup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait.1680004; Bootup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait, 1680004; Boostup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait, 1680004; Boostup, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait, 1680008; Boostoader, 1680010; Pre-operation, 1680020; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait, 1680008; Boostoader, 1680010; Pre-operation, 1680000; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait, 1680008; Boostoader, 1680010; Pre-operation, 1680000; Operation, 1680040; Stop, Modou Value: 1680001; Energi, 1680002; Hait,

### iDCS-8830's Communication Status:

Name	iDCS8830_LAN1_error_status iDCS8830_LAN2_error_status						
Test Steps		<ol> <li>Unplug (or plug in) the LAN cable of the 1st FCM-MTCP. Value: Non-0 (or 0).</li> <li>Unplug (or plug in) the LAN cable of the 2nd FCM-MTCP. Value: Non-0 (or 0).</li> </ol>					
Return Value	0 : Communicat	0 : Communication OK ; Non-0 : Communication Error					
	N1_error_status N2_error_status	130 0		-zero : error. LAN error status of the 1st FCM-MTCP. -zero : error. LAN error status of the 2nd FCM-MTCP.			

#### iDCS-8830's Power Status:

Name	iDCS8830_FPM_status					
Tost Stops	1. Unplu	g (or plug	in) the powe	r cable of the 1st FCM-MTCP. Value: 2 (or 3).		
Test Steps	2. Unplug (or plug in) the power cable of the 2nd FCM-MTCP. Value: 1 (or 3).					
Return Value	0 : No FPN	Л is plugg	ed.	1 : FPM1 Off ; FPM2 Good.		
Return value	2 : FPM1 (	Good ; FP	M2 Off.	3 : Both of these FPM are plugged.		
iDCS8830_LAN1_e	error_status	0	Value:0:ok , r	non-zero : error. LAN error status of the 1st FCM-MTCP.		
iDCS8830_LAN2_e	error_status	0	Value:0:ok , r	non-zero : error. LAN error status of the 2nd FCM-MTCP.		
iDCS8830_FCM1_r	mode 32 Value : 16#0020 : Master , 16#0021: Slave. Modbus start-addr = 513. Redundant mode of the 1st FCM-					
iDCS8830_FCM2_r	node 33 Value : 16#0020 : Master , 16#0021: Slave. Modbus start-addr = 513. Redundant mode of the 2nd FCM					
iDCS8830_FCM1_s	tatus 16 Value : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16#0010: Normal. Modbus-addr =					
iDCS8830_FCM2_s						
iDCS8830_FPM_st	atus	2	Value: 0x0000: I	No FPM plugged , 0x0001: FPM1 Good / FPM2 Off , 0x0002: FPM1 Off / FPM2 Good ,		

### FCM-MTCP's Redundant Mode and Status:

Name	iDCS8830_FCM1 iDCS8830_FCM1	—	iDCS8830_FCM2_mode iDCS8830_FCM2_status
	Unplug the 1st FCM-MTC	P module (FCM1).	
Test Step	iDCS8830_LAN1_error_status iDCS8830_LAN2_error_status iDCS8830_FCM1_mode iDCS8830_FCM2_mode iDCS8830_FCM1_status iDCS8830_FCM1_status	0 Value 32 Value 32 Value 1 Value	: 0: ok , non-zero : error. LAN error status of the 1st FCM-MTCP. : 0: ok , non-zero : error. LAN error status of the 2nd FCM-MTCP. : 16#0020 : Master , 16#0021: Slave. Modbus start-addr = 513. R€ : 16#0020 : Master , 16#0021: Slave. Modbus start-addr = 513. R€ : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16 : 16#0000: Empty , 16#0001: Timeout , 16#0002: Undefined , 16
	iDCS8830_LAN1_error_st	atus = 130	
	It means the FCM1 comm	nunication error.	
	iDCS8830_FCM1_mode =		
Description			(Master) because the FCM1 is unplug.
	iDCS8830_FCM2_mode =		
		-	ts redundant mode is "Master" (32).
	iDCS8830_FCM1_status =	= 1 means that the	FCM1 is timeout.
	iDCS8830_FCM1_mode /	iDCS8830_FCM2_	mode
Return	32 : Master 33: Slave		
Value	iDCS8830_FCM1_status /	iDCS8830_FCM2_	status
	0: Empty 1: Timeout 2:	Undefined 16: N	Iormal
Plug in the FC	M1. it will change to "Slave	" mode (33), and	becomes a normal status (16).
_	N1_error_status 0		n-zero : error. LAN error status of the 1st FCM-MTCP.
	NV2_error_status 0		n-zero : error. LAN error status of the 2nd FCM-MTCP.
iDCS8830_F0			Master , 16#0021: Slave. Modbus start-addr = 513. R
iDCS8830_F0			Master , 16#0021: Slave. Modbus start-addr = 513. R
iDCS8830_F0	_		Empty , 16#0001: Timeout , 16#0002: Undefined , 16
iDCS8830_F0	CM2_status 16	Value : 16#0000:	Empty,16#0001: Timeout,16#0002: Undefined,16
You can try to	o unplug the FCM2, and the	n the FCM1 will b	ecome the Master (32) and take over.
iDCS8830_L	AN1_error_status 0	Value : 0: ok , no	n-zero : error. LAN error status of the 1st FCM-MTCP.
	AN2_error_status 130		n-zero : error. LAN error status of the 2nd FCM-MTCP.
iDCS8830_F	_		: Master , 16#0021: Slave. Modbus start-addr = 513. Ri
iDCS8830_FI iDCS8830_FI			: Master , 16#0021: Slave. Modbus start-addr = 513. R Empty , 16#0001: Timeout , 16#0002: Undefined , 16
iDCS8830_F			Empty , 16#0001: Timeout , 16#0002: Undefined , 16
	-		
Plug in the FC	CM2 afterward.		
iDCS8830_L	AN1_error_status 0	Value : 0: ok , inc	n-zero : error. LAN error status of the 1st FCM-MTCP.
	AN2_error_status 0		n-zero : error. LAN error status of the 2nd FCM-MTCP.
iDCS8830_F	_		: Master , 16#0021: Slave. Modbus start-addr = 513. Ri
iDCS8830_FI iDCS8830_FI	—		: Master , 16#0021: Slave. Modbus start-addr = 513. Ri Empty , 16#0001: Timeout , 16#0002: Undefined , 16
	CM1_status 16 CM2_status 16		Empty , 16#0001: Timeout , 16#0002: Undefined , 16
	-		

### The Status of I/O Slot and I/O Emergency:

Name	iDCS8830_io_slot_status	iDCS8830_io_emergency_status
		(slot 0) and the terminal board DN-DI-32DW.
Test step	□DCS8830_io_slot_status         iDCS8830_io_slot_status[0]       64         iDCS8830_io_slot_status[1]       32         iDCS8830_io_slot_status[2]       32         iDCS8830_io_slot_status[3]       32         iDCS8830_io_slot_status[4]       1         iDCS8830_io_slot_status[5]       1         iDCS8830_io_slot_status[6]       1         iDCS8830_io_slot_status[6]       1         iDCS8830_io_slot_status[7]       1         ⊡DCS8830_io_emergency_status       1         iDCS8830_io_emergency_status[0]       256         iDCS8830_io_emergency_status[1]       0         iDCS8830_io_emergency_status[2]       0         iDCS8830_io_emergency_status[3]       0	Value: 16#0001: Empty , 16#0002: Halt, 16#0004: Bootup, 16#0008 Value: 10CS8830_io_emergency_status[x]" works only when you set the "IOM_x" as "Duplex" Mode. value: 0: OK, value: 10#0100: Cable break-on", 10#0020: CoC enot.
Description	iDCS8830_io_slot_status[0] = 64, which r iDCS8830_io_emergency_status[0] = 256 It means the cable which plugged into th Plug in this cable and then you can try to	is module (slot0) is disconnected.
Return Value	iDCS8830_io_slot_status[x] 1: Empty 2: Halt 4: Bootup 8: Bootloade iDCS8830_io_emergency_status[x] 0 : Normal 32: CJC Error 256: Cable Bre	er 16: Pre-operation 32: Operation 64: Stop eak-off

### Redundant DO module:

Name	iDCS	8830_slot2	3_F	8041_DO	
	Plug in the 1st F-804 <b>1</b> DO module (	slot2).			
	⊟ iDCS8830_io_slot_status			Value: 16#0001: Empty , 16#0002: Halt,	
	iDCS8830_io_slot_status[0]	3	32	Value: 16#0001: Empty , 16#0002: Halt,	
	iDCS8830_io_slot_status[1]	3	32	Value: 16#0001: Empty , 16#0002: Halt,	
	iDCS8830_io_slot_status[2]	2	2	Value: 16#0001: Empty , 16#0002: Halt,	
	iDCS8830_io_slot_status[3]	3	32	Value: 16#0001: Empty , 16#0002: Halt,	
Test step	□ iDCS8830_slot23_F8041_D0		F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
•	iDCS8830_slot23_F8041_D0[0]	TRUE	F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
	iDCS8830_slot23_F8041_D0[1]	TRUE	JE F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830		
	iDCS8830_slot23_F8041_D0[2]	TRUE	F-8041 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830		
	iDCS8830_slot23_F8041_D0[3]	TRUE	F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
	iDCS8830_slot23_F8041_D0[4]	TRUE	F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
	iDCS8830_slot23_F8041_DO[5]	TRUE	F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
	iDCS8830_slot23_F8041_DO[6]	TRUE	F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
	iDCS8830_slot23_F8041_D0[7]	TRUE	F-80	141 Digital Output in the slot 2 - 3 (Duplex mode) of iDCS-8830	
	At first, the LED indicator (called D	⊃0 ~ 7) of th	ne 1	st F-8041 module (slot2) will light up	
Description				nd F-8041 module (slot3) will take over	
	and do the same thing. Then, you or status[2] is "2" which means this m			us value of the iDCS8830_io_slot_ halted. (Plug in this module again.)	

# 16.5 What Kinds of Data Can be Automatically Backed up to the Passive PAC?

In the Win-GRAF redundant system, not all of the data in the Active PAC can be automatically backed up to the Passive PAC.

### What Can be Backed Up Automatically:

- 1. The user's Win-GRAF applications.
- 2. The execution step of programs.
- 3. Value of variables.
- 4. Private data of Function Block instance.
- 5. The PAC's RTC (Real Time Clock) time.
- 6. Retain memory.
- 7. Schedule-control configuration (refer Chapter 17).

### What are NOT Backed Up Automatically?

The following are the most common items that cannot be automatically backed up to the Passive PAC.

- 1. The status of Timer variable (Ticking or Sleeping).
- Files in the Active PAC (e.g., files located in the path "\system\_disk" or "\Micro\_SD", or files not belong to the Win-GRAF applications, such as C, VB.net, C#, and ELogger applications). These files cannot be backed up automatically. So all of them should be pre-installed in a spare (or repaired) PAC before mounting this PAC in the redundant system).

(Exclusive of user designed Win-GRAF application and schedule-control configuration, which can be backed up automatically.)

- 3. If using the COM\_OPEN() function to open the serial port, it will not be automatically opened again after switching PAC control-right.
- 4. The PAC's EEPROM memory cannot be backed up automatically.

All the items which unable to back up automatically, users can use the following similar procedures to deal with them. (Refer the "Retain\_and\_timer" program in the "demo\_RDN\_2" project)

if is_first_cycle_just_after_switch then	
(* Just in the cycle after switching. *)	
end_if ;	

# Chapter 17 Schedule Control



# SCHEDULE CONTROL

### Introduction:

All Win-GRAF WinCE series PAC support the Schedule-Control function. One PAC can control max. 10 Targets (devices) with specified schedule configurations. Each schedule Target (device) contains three variables to be controlled – one BOOL variable, one DINT variable and one REAL variable. ICP DAS provides a free software – "Schedule-Control Utility". User can use this software to edit /modify the schedule configurations easily in PC or in PAC.

### Driver version of Win-GRAF PAC:

The Win-GRAF PAC supports Schedule-Control in the below driver version and the new version.

WinCE PAC	Win-GRAF PAC	Driver Version
ViewPAC	VP-x2x8-CE7	1.02
WP-8000	WP-8148, WP-8448, WP-8848	1.02
WP-8000-CE7	WP-8128-CE7, WP-8428-CE7, WP-8828-CE7	1.01
WP-5000	WP-5238-CE7	1.01
XP-8000-CE6	XP-8048-CE6, XP-8348-CE6, XP-8748-CE6	1.01

#### You may download newer Win-GRAF driver at

http://www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.html

# 17.1 Install the Schedule-Control Utility and Restore the Win-GRAF Demo Project

There is one Win-GRAF-PAC-CD in the Win-GRAF PAC package box. The Schedule-Control Utility file name is "**Schedule\_in\_PC.exe**" in the "CD:\napdos\Win-GRAF\Tools\_Utility\" path.

Please copy this Schedule-Control Utility (Schedule\_in\_PC.exe and label\_name.txt) to your PC. Recommend to copy it to the directory of your Win-GRAF project. For instance, copy it to "D:\Schedule-Control\**Station1**\Schedule\_in\_PC.exe", then run this "Schedule\_in\_PC.exe" file.

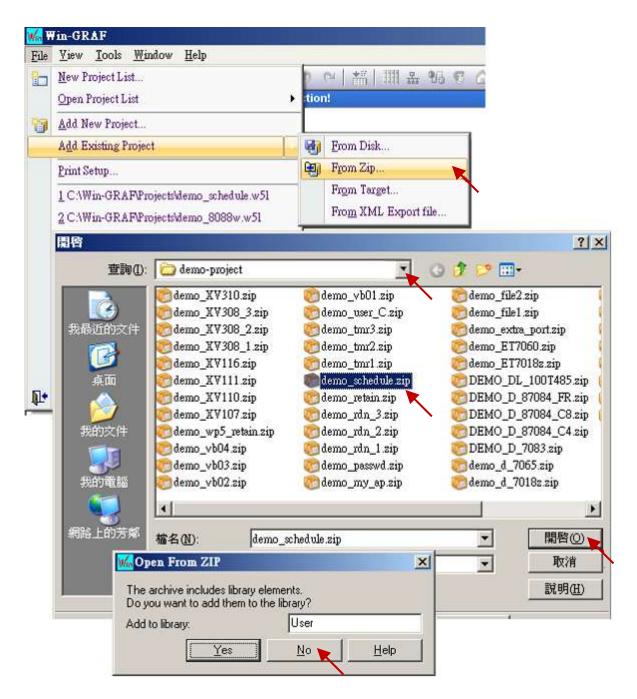
Sine to PC Sand to Control USA:	Controller trac over homitotes Ques for	can PC Gertinai, Cautolin Belg-shore		
Target 1	Target 2	Target 3		
	Schedule-Control Unity Target 1 > 5 Back Saw to PC	eason Always > Normal day		1200
Target 4	W Target	Defecti Yalaa OPP +	lateges 0.0	Real
Target 7	🗷 Season	Always	Always	
Target 7	· · · · · · · · · · · · · · · · · · ·	Always son 1 Sesson 2*	Always Sesson 3	Sesson 4
Target 7	······	son I Sesson 2*	Sesson 3	Sesson 4
Target 7	Sesson Always * See @ Normal * Normal day (Schedule 1 *)	non 1 Seanon 2 * day Apply Shelds 1 * •	Sesson 3	
Target 7	Sesson Always * See @ Normal * Normal day (Schedule 1 *)	non 1 Sesson 2 * day Apply Sheduk 1 * • Shaday Wooday Wooday Treator	Sesson 3	tat *
Target 7	Sesson Always * See Ø Normal • Nomai day (Schedule 1 *) • Holiday 1 • Holiday 2	non 1 Sesson 2 * I day Apply Schedul * • Disastoy Winaday	Sesson 3	ide1

Then we will see the following windows (Click New > Target 1).

There is one another Schedule-control utility, however it is set up on the **PAC** (not on the PC). It is "**Schedule\_in\_PAC.exe**". You can find it in the "\System\_Disk\Win-GRAF\" path of the Win-GRAF PAC.

### Restore Win-GRAF demo project :

The Win-GRAF demo project for the Schedule-Control is "demo\_schedule.zip". It is in the "\napdos\Win-GRAF\demo-project\" path of the Win-GRAF-PAC-CD. Restore it to the PC / Win-GRAF workbench by the following way.

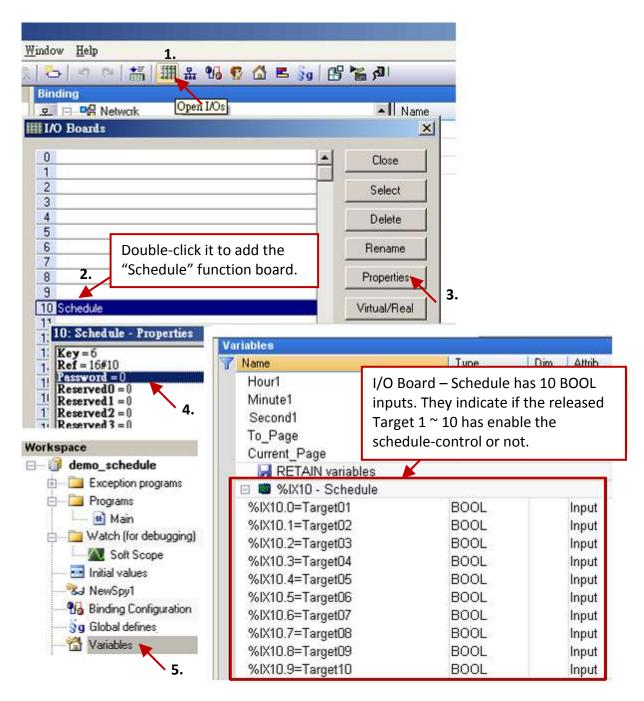


# 17.2 Introduction of the "demo\_schedule" Project

This "demo\_schedule" project shows the way to do schedule-control. Please prepare one Win-GRAF PAC (like VP-x238-CE7 or WP-8x48). One PAC can control schedules of max. 10 Targets (Target 1 to Target 10). Each Target contains one BOOL, one DINT and one REAL variable.

### Settings in the "I/O boards" window :

To enable schedule-control in the Win-GRAF PAC, first click the "Open I/Os" to add one "Schedule" (add it in the slot number 8 or bigger number). There is a "Password" parameter in its "Properties" window. The "Password" is for the "Schedule-Control Utility" running in PC to identify the authorization when connecting the Win-GRAF PAC. It is set as 0 in this demo project. After adding the "Schedule" in the "I/O boards", we can find 10 BOOL input channels in the "variables" window. These 10 channels return the state of the schedule-control of the Target 1 to 10. TRUE means the Target has the schedule-control utility". FALSE means not enabled.



### Variable declaration:

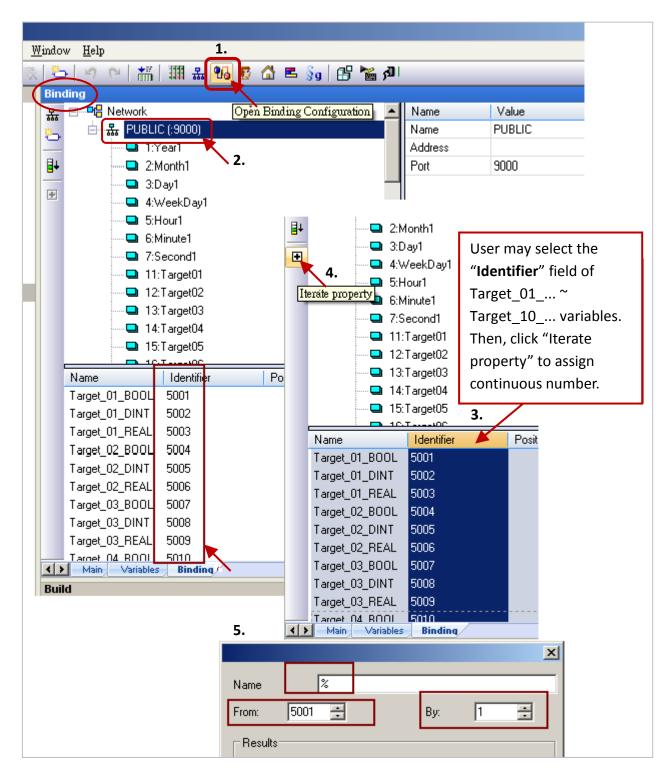
Click the "Variables" to view all variables in this demo project.

<u>File Edit V</u> iew Insert Project <u>T</u> ools	<u>W</u> indow <u>H</u> elp		
3 🖬 🕑 🥵 🕹 🕹 🖧 🗙 1	🔍 😓   어 🗠   👬   🎹 品 🕅	5 🖸 🖾 🖻 5g (f	马 🌹 🕄
Workspace	Variables		
🖃 🗊 demo_schedule	🍸 Name	Туре	Dim.
Exception programs	📃 🖂 Global variables		
- Programs	Target_01_BOOL	BOOL	
Main	Target_02_BOOL	BOOL	
😑 🖂 Watch (for debugging)	Target_03_BOOL	BOOL	
Soft Scope	Target_04_BOOL	BOOL	
🔜 Initial values	Target_05_BOOL	BOOL	
6J NewSpy1	Target_06_BOOL	BOOL	
93 Binding Configuration	Target_07_BOOL	BOOL	
😽 g Global defines	Target_08_BOOL	BOOL	
🚽 🔛 Variables 🔪	Target_09_BOOL	BOOL	
E Types	Target_10_BOOL	BOOL	
	Target_01_DINT	DINT	
	Target_02_DINT	DINT	
	Target_03_DINT	DINT	
	Target 04 DINT	DINT	
	Target_05_DINT	DINT	
	Target_06_DINT	DINT	
	Target_07_DINT	DINT	
	Target_08_DINT	DINT	
	Target_09_DINT	DINT	
	Target_10_DINT	DINT	
	Target_01_REAL	REAL	
	Target_02_REAL	REAL	
	_T	0041	_

The above variables - "Target\_01\_BOOL ~ Target\_10\_BOOL", "Target\_01\_DINT ~ Target\_10\_DINT" and "Target\_01\_REAL ~ Target\_10\_REAL" - will be controlled by the Win-GRAF PAC. They represent these variables belong to the 10 Targets.

Data Binding

See <u>Chapter 7</u> for more details about the Data Binding. If the user wants to open variables of this Win-GRAF PAC for other PACs to read data, these variables that described as above should be dragged into the "Punlic" area of the "Binding" window, and then assign a ID for them. To be controlled correctly by the Schedule-Control configurations, the "Identifier" number **Must be** set from 5001 to 5030. After that, the system will deal with the schedule controls for ten targets according to the settings in the Schedule-Control Utility.



# <u>Note:</u> If the user want to publish Win-GRAF variables to allow the eLogger HMI to get data. See <u>Section 3.1 To Enable the Win-GRAF PAC as a Modbus TCP Slave</u> for more details.

# 17.3 Edit Schedule Configurations by the Schedule-Control Utility

Here shows a simple example to use the Schedule-Control Utility, refer the Section 17.5 for details.

1. Execute the Schedule-Control Utility (Schedule\_in\_PC.exe) in the PC.

(Click "New" and click "Target 1" to open the Schedule setting window for the "Target 1").

The strange carry for all	The service of the local servi				
Jane 190 - Sei er Catalio	Controlle fire photometrics	icu.20. Online Canada Mig-store			
36.54					
Target 1	Target 2	Target 3			
Target 4	Target 5	Target 6			
Target 7	Target 8	Target 9			
1	Target 10				
	Target 1 Target 4	Instant     Instant     Open       Terret 1     Target 2       Target 4     Target 5       Target 7     Target 8			

- (2) Click "Season Always".
- (3) Check "Season Always" box to enable it.
- (4) Click "Normal Day" item (Normal day is usually used for Monday ~ Friday.)
- (5) Check "Normal Day" box to enable it and then set proper settings (e.g. Monday ~ Friday).
- (6) Click "Schedule 1" to set the schedule period for the "Schedule 1".

🖳 Schedule-Control Utility Targ	et 1 > Season Always > Normal day			×
Back Save to PC	arget 1 Default Value	Boolean OFF V 0	Integer 1	Real
The "*" means that the item is	1)			
a walala al	eason Always		Always	
Season Always *	Season 1 Se	eason 2	Season 3	Season 4
(2)				
<u> </u>	ormal day	_	Schedule	
Normal day (Schedule 1 !!!)	Apply Schedule 1	1	Schedule	
🖱 Holiday 1 👍	🔲 Sunday			
	📝 Monday 📝 Tuesday		Schedule	3
🖱 Holiday 2	V Wednesday		L	
Special day	👿 Thursday		Schedule	4
	📝 Friday		L	
	🔲 Saturday		Schedule	5

### 3. Set the Schedule Period

After selecting "Schedule 1" in the step2 – (6), do the following steps.

(1) Check "01" to enable the setting for the No. 01 Time Period of the Schedule 1.

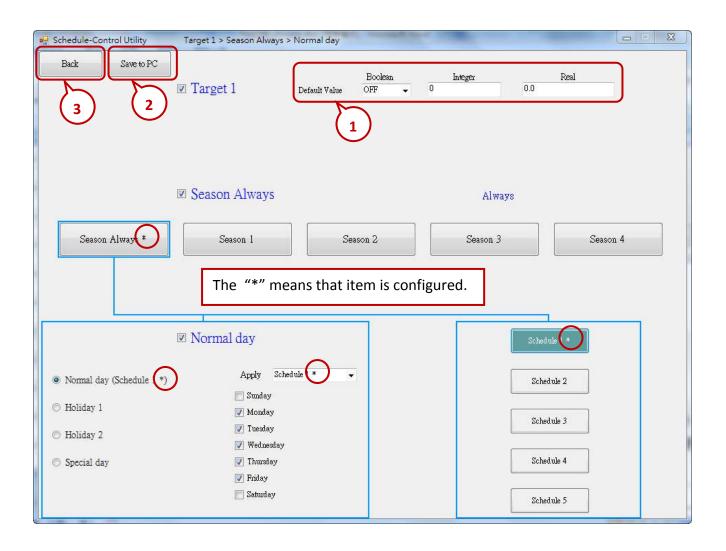
- (A) Set up the time as the figure below, or the time which easily for testing.
- (B) Set up the Boolean, Integer, Real variables to the values that you want to control, or follow the setting in the figure below.
- (2) Check "02" to enable the setting for the No. 02 Time Period, such as the step (1).

Each schedule can set up a max. of 15 Time Periods.

After completing the settings, click "Save and exit" to save and exit this window.

Schedule 1	1												
									Сору	from			
	Ho	ur	Minu	ute	To	Ho	or	Min	ute	] [	Boolean	Integer	Real
<b>V</b> 01:	8	-	30	•		12	•	0	•		on 🚽	10	12.34
<b>V</b> 02:	13	•	0	•		17	•	30	-		on 🚽	20	25.67
03:	0	<b>T</b>	0	Ξ.		0	×.	0	Ψ.	_	OFF		U
04:	0	-	0	-		0	•	0	-		OFF	0	0
05:	0	-	0	-		0	-	0	-		OFF	0	0
06:	0	-	0	-		0	×.	0	-		OFF 🚽	0	0
07:	0	-	0			0		0	-		OFF -	0	0
08:	0	-	0	- T		0		0	-		OFF 🚽	0	0
09:	0	-	0	- T		0	I T	0			OFF	0	0
<b>1</b> 0:	0	-	0			0		0	-		OFF	0	0
<b>11</b> :	0	-	0			0		0	-		OFF	0	0
12:	0	T	0			0		0	-		OFF 🚽	0	0
13:	0	-	0			0		0	- T		OFF 🚽	0	0
<b>14</b> :	0	-	0	-		0	ι	0	-		OFF 🚽	0	0
15:	0	-	0	-		0	Ŧ	0	Ŧ		OFF -	0	0
Save a	and exit												Cancel

4. Then, it will go back to the previous setting window as the figure below. And, the "\*" symbol show on the screen means the season or schedule has been configured.



 "Default Value" (in the upper right) is for the default setting. If the current date is not found in any Schedule setting or the date is found, however, its time period is not found in the related Schedule 1 ~ 5, the Target device will be controlled follow the "Default Value". The "Default Value" in this demo project is "Boolean: OFF, Integer: 0, Real: 0.0".

### Advantage of the Default Value:

Utilizing the "Default Value" can reduce the amount of the Periods setting in the Schedule 1 ~ 5.

Ex: The following example sets 5 Periods in the Schedule 1.

- (1)  $00:00 \sim 08:00$  OFF 0 0.0
- (2) 08:00~09:50 ON 0 0.0
- (3) 09:50  $^{\sim}$  10:00 OFF 0 0.0
- (4) 10:00 ~ 11:50 ON 0 0.0
- (5) 11:50 ~ 24:00 OFF 0 0.0

If utilize the "Default Value" as "OFF, 0, 0.0", the user just needs to set 2 Periods as below.

- (1) 08:00 ~ 09:50 ON 0 0.0
- (2) 10:00 ~ 11:50 ON 0 0.0
- After completing all settings, click "Save to PC" to save a configuration file in PC. (This demo uses "test1.txt")
- 7. Click "Back" to return to set up other Targets. (This demo sets Target 1 only)

# 17.4 Testing the "demo\_schedule" Project

This section shows the way to implement the Win-GRAF project and schedule configuration in the Win-GRAF PAC. Then testing the schedule control.

- Download the "demo\_schedule" project to the PAC by the Win-GRAF workbench. (For more information, refer the <u>Section 2.3.5</u>)
- 2. Download the schedule configuration to the PAC by the Schedule-Control Utility.
  - (1) Click "Send to Controller".
  - (2) Assign the PAC's IP address (remember to fill in your PAC's IP address) Set a password (This demo uses "0") Check "Remember Password" can remember this password Click "OK" to send the schedule setting to the PAC.
  - (3) If success, it will pop up a "Send file success" window.
- <u>Note:</u> Before downloading the project, make sure the Schesule-Control Utility on PAC has been activated. Moreover, add the \System\_Disk\Win-GRAF\Schedule\_in\_PAC.exe path in the Auto Execution page of the PAC Utility.

🖳 Schedule-Control Utility		
Save to PC Send to Controller	Controller time synchronization Open from	n PC Get from Controller Help-about
Title : Title	N2	
Target 1 *	Target 2	Target 3
	Controller info	
Target 4	IP : 192.168.75.101 Password :	
Target 7	OK Cancel Target 8	Send file success. 確定
	Target 10	

### 3. Test the Win-GRAF project:

Click "On Line" to connect the PAC by the Win-GRAF workbench, then open the "NewSpy1" window. If the connection is fine, we can see variables - Target\_01\_xxx ~ Target\_10\_xxx are controlled properly by the schedule configurations which is set by the "Schedule-Control Utility".

The user may use the "Schedule-Control" Utility to modify the schedule configurations and then download to the Win-GRAF PAC to see if those variables are controlled well.

Win-GRAF - demo_schedule								x
File Edit View Insert Proje	ect	Tools Window H	lelp				_	
27 🖬 By   3   X 🗈 🖺	×	🖹 🔁 🔊 🕅	講 🖩 🖽 品	98	😨 🗳 🛋 💱 🖻	📲 📶 RU	IN 🔹	00
Workspace		NewSpy1.spl *						íΧ
🖃 🗝 🕨 demo_schedule (RUN)		Name	Value	Y	Name 🔼	Value	Туре	
🗄 🔤 Exception programs		Year1	2017		🗆 🚮 Global variable	s		
🖨 🔤 Programs		Month1	7		Current_Page	0	DINT	
🔛 🎬 Main	∎∔	Day1	5		Day1	5	DINT	
📄 👘 📴 Watch (for debugging)		WeekDay1	3		Hour1	10	DINT	
Soft Scope		Hour1	10		Minute1	29	DINT	=
📰 Initial values		Minute1	29		Month1	7	DINT	
🐜 NewSpy1 📐		Second1	39		Second1	39	DINT	
Binding Configuration		Target01	TRUE		Target_01_BOOL	TRUE	BOOL	
💮 🖇 🖁 🛛 🗑 🗑 🚽		Target02	FALSE		Target_01_DINT	10	DINT	
🖓 Variables		Target03	FALSE		Target_01_REAL	12.34	REAL	
Types		Target_01_BOOL	TRUE		Target_02_BOOL	FALSE	BOOL	
🛄 (All Projects)		Target_01_DINT	10		Target_02_DINT	0	DINT	
		Target_01_REAL	12.34		Target_02_REAL	0.0	REAL	
		Target_02_BOOL	FALSE		Target_03_BOOL	FALSE	BOOL	
		To_Page	0		Target_03_DINT	0	DINT	
		Target_02_DINT	0		Target_03_REAL	0.0	REAL	
		Target_02_REAL	0.0		Target_04_BOOL	FALSE	BOOL	
		Target_03_BOOL	FALSE		Target_04_DINT	0	DINT	
		Target_03_DINT	0		Target_04_REAL	0.0	REAL	
		Target_03_REAL	0.0		Target_05_BOOL	FALSE	BOOL	
		Current_Page	0		Target_05_DINT	0	DINT	
					Target_05_REAL	0.0	REAL	-
		< III	Þ		< III	-		

# 17.5 Configurations of the Schedule-Control Utility

### 17.5.1 Address for each Target Variables

The Schedule-Control Utility can configure max. 10 Target 's schedule. Each Target contains one BOOL variable, one DINT variable and one REAL variable.

### To enable the schedule-control in the Win-GRAF PAC, first add a "Schedule" in the "I/O boards"

**windows** (see Section 17.2). The user can declare all required variables in the "Variables" window, and add these variables in the "Binding" window and then assign correct **"Identifier" number 5001 ~ 5030** (see Section 17.2 - Variable declaration). After downloading the Win-GRAF project to the PAC, the scheduling will control these variables well.

Address	Туре	Description	Address	Туре	Description
5001	BOOL	BOOL DINT and REAL	5016	BOOL	BOOL, DINT and REAL
5002	DINT	BOOL, DINT and REAL variable controlled	5017	DINT	variable controlled
5003	REAL	by Target 1	5018	REAL	by Target 6
5004	BOOL	POOL DINT and PEAL	5019	BOOL	BOOL, DINT and REAL
5005	DINT	BOOL, DINT and REAL variable controlled	5020	DINT	variable controlled
5006	REAL	by Target 2	5021	REAL	by Target 7
5007	BOOL	BOOL, DINT and REAL variable controlled	5022	BOOL	BOOL, DINT and REAL
5008	DINT		5023	DINT	variable controlled
5009	REAL	by Target 3	5024	REAL	by Target 8
5010	BOOL	BOOL, DINT and REAL	5025	BOOL	BOOL, DINT and REAL
5011	DINT	variable controlled	5026	DINT	variable controlled
5012	REAL	by Target 4	5027	REAL	by Target 9
5013	BOOL	BOOL, DINT and REAL	5028	BOOL	BOOL, DINT and REAL
5014	DINT	variable controlled	5029	DINT	variable controlled
5015	REAL	by Target 5	5030	REAL	by <b>Target 10</b>

# 17.5.2 Target Configuration

Every Win-GRAF WinCE PAC can control maximum 10 "Target" (Target 1 to Target 10) devices. First, execute the Schedule-Control Utility and click "New" to create a new configuration file, the Targets will show as 10 buttons (See the figure below). The default Target names are "Target 1" ~ "Target 10". One Target can set up the schedules to fit different Seasons. The Target button will show a "\*" to distinguish it is enabled.

Addition to "New" a configuration file, the user can open an existing file in PC or get from the PAC.

New:Create a new file.Open from PC:Open an exist configuration file from PC.

Get from Controller: Get an existing configuration file from PAC (required enter the PAC 's IP and password) and then to show on the PC.

Schedule-Control Utility		
Save to PC Send to Controller	Controller tose spectronization (	Open from PC Get from Controller Hal
Tale Tale		
Target 1 *	Target 2	Target 3
Target 4	Target 5	Target 6
Target 7	Target 8	Target 9
	Target 10	

### Change the Target Name to meet the needs of the field:

User can change the name of the Target, Season or other items to fit for the equipment at the application field. Please create a text file named "Label\_Name.txt" (as the figure below) and save it in the same folder with the Schedule-Control Utility "Schedule\_in\_PC.exe" (e.g., D:\Schedule-Control\ Station1\Label\_Name.txt).

### Notes for creating the file "Label\_Name.txt":

- 1. If this file does not exist, the Target shows the default name (e.g., Target 1, Target 2).
- 2. In this file, change the target name (e.g., "Factory", the prefix/suffix of spaces will be erased.) or the name of other items (e.g., Season, Normal day, Holiday, Schedule, etc.) after the colon (":").
- 3. The user can create and edit it by using MS Notepad or other editors, but must select the "Unicode" format when saving it.
- 4. On the PAC, copy this file into the same folder with the Schedule\_in\_**PAC**.exe, i.e., \System\_Disk\ Win-GRAF\.

檔案(F) 編輯(E) 格式(O)	) 檢視(V)		
說明(H)			
Target1 : Target 1	月存新檔		
Target2 : Target 2	D:\Schedule-Co	ontrol\Station1 🛛 👻 🍫	搜尋 Station1 👂
Target3 : Target 3			
Target4 : Target 4	檔案名稱(N): label_nam	ne.txt	•
Target5 : Target 5	存檔類型(T): 文字文件(*	*.txt)	•
Target6 : Target 6			
Target7 : Target 7			
Target8 : Target 8	💌 瀏覽資料夾(B)	編碼(E): Unicode 🖌 🗸	存檔(S) 取消
Target9 : Target 9			
Target10 : Target 10			
Season Always : Seaso			
Season 1 : Season 1 Season 2 : Season 2	=		
Season 2 : Season 2 Season 3 : Season 3			
Season 5 : Season 5 Season 4 : Season 4	Schedule-Control Utility		
Normal day : Normal d	Sine to PC Sead to Controller	Controller time cyschronization Open fore	a PC Get from Controller Halp-about
Holiday 1 : Holiday 1	Tite Tite		
Holiday 2 : Holiday 2			
Special day : Special da	av		
Schedule1 : Schedule 1			
Schedule2 : Schedule 2	1 areet 1	Target 2	Target 3
Schedule3 : Schedule 3	3		
Schedule4 : Schedule 4	1		
Schedule5 : Schedule 5	5		
*	Target 4	Target 5	Target 6
	Taget i	ruger o	Tuger 0
	1		
	1		
	Target 7	Target 8	Target 9
		Target 10	
		Target 10	
	l		
	-		

## 17.5.3 Season Configuration

Each "Target" (1 ~ 10) includes the "Season Always", "Season 1", "Season 2", "Season 3" and "Season 4" setting items. It is recommend to check "Season Always" that means to enable the year-round schedule.

#### The Searching Priority of Seasons:

- 1. PAC will first search the **Season 4** (if it is enabled) If found the current date in the Season 4, then do the Boolean/Integer/Real control.
- 2. If not found, then search the Season 3, Season 2..., at last search the Season 1.
- 3. If not found, then search the **Season Always** to do its control.
- 4. If not found the current date in this Target, then do the "Default Value" control.

🖳 Schedule-Control Utility	Target 1 > Season 4 > Normal day		
Back Save to PC		Boolean Integer OFF v 0	Real 0.0
	Season 4 Every year	From 2013/Oct/01	To 2013/Dec/31
Season Always *	Season 1 Seaso	n 2 Season 3	Season 4 *
Low			High
	🗷 Normal day		Schedule 1 *
Normal day (Schedule 1 *)	Apply Schedule 1 * -	[	Schedule 2
<ul> <li>Holiday 1</li> <li>Holiday 2</li> </ul>	<ul><li>Monday</li><li>Tuesday</li></ul>	[	Schedule 3
Special day	<ul> <li>Wednesday</li> <li>Thursday</li> <li>Friday</li> </ul>	[	Schedule 4
	Saturday		Schedule 5

#### **Season Setting:**

Season  $1 \sim 4$  need to set its "Date Period". After completing the settings, recommand to check the "Every Year" option to apply the Date Period every years.

- **Note 1:** The Date Periods of the 4 Seasons must not overlap.
- **Note 2:** If the "Every Year" is checked, the system diagnoses the overlap of Month/Day only, not the year. If the "Every Year" is not checked, it will diagnose the "From" Year/Month/Day should be earlier than the "To" Year/Month/Day.

🖳 Schedule-Control Utility	Target 1 > Season 4 > N	lormal day		
Back Save to PC	☑ Target 1	B∞lean Default Value OFF ↓	Integer 0 0.0	Real
Season Always *	Season 4	✓ Every year From Oc Season 2	x01 Season 3	To Deo31 Season 4 *

**<u>Note</u>**: Unchecked the "Every Year" can modify the date periods, and take notice of the date order.

### For example:

#### 1. The Correct Setting:

User can check "Every Year", so that the setting will be used for every year.

Season 1	01/01 ~ 03/31
Season 2	04/01~07/15
Season 3	07/16 ~ 09/30
Season 4	10/01 ~ 12/31

#### 2. The Wrong Setting:

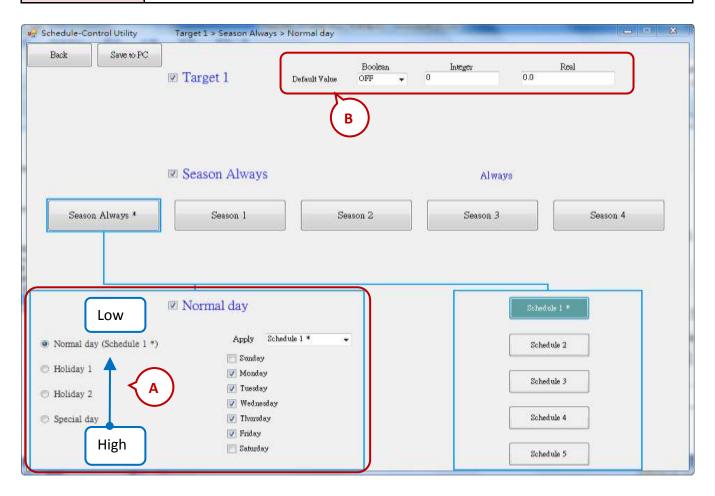
Because the Season 2 overlaps the date of Season 1 from 03/16 to 03/31.

Season 1	01/01 ~ <b>03/31</b>
Season 2	<b>03/16</b> ~ 07/15
Season 3	07/16 ~ 12/31
Season 4	Disabled

# 17.5.4 Normal Day / Holiday / Special Day Configuration

There are Normal day, Holiday 1, Holiday 2 and Special day in each Season. When enabling the setting, users must choose a Schedule number (1 to 5) to apply the time settings.

Normal day	The normal days are Monday to Friday.
Holiday 1	Normally set to Saturday and Sunday.
Holiday 2	In some workplace, there are different holidays, e.g., Wednesday.
Special Day	Set the schedule for local holidays or the adjusted working-day. E.g., Oct. 10, Jul. 4, Oct. 1, Dec. 25, etc. A maximum of 50 days can be set per Season.



### A. The Searching Priority of Normal Day / Holiday / Special Day:

The PAC will first search **"Special day"**. If the date is not found in this Special day setting, then search **"Holiday 2"**, then **"Holiday 1"**, and then **"Normal day"**.

### B. Default Value for Boolean / Integer / Real:

Each Target must set the default value for the Boolean, Integer and Real variables. These default values will be applied when the PAC cannot find any available "Date Period" or "Time Period" in the enabled "Season" setting. Then the PAC follows the setting of the Default Value. Usually, the Default Value of Boolean is set to be "OFF", the Integer and Float value are set to be "O". User can set the different Default Value by the case.

### C. Date Setting for Normal day / Holiday 1 / Holiday 2:

Note that NO OVERLAP. For example,

The Correct	Setting:
-------------	----------

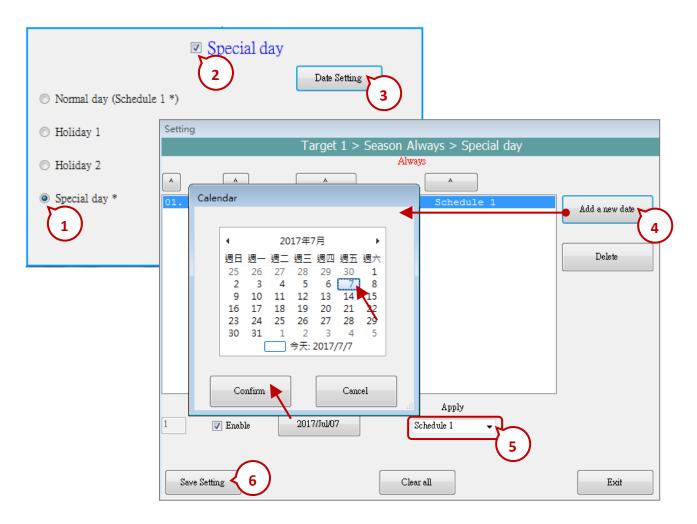
Normal day	Monday, Tuesday, Wednesday, Thursday, Friday
Holiday 1	Sunday, Saturday
Holiday 2	Disabled

The Wrong Setting: (Because "Friday" overlaps in the setting of "Normal day" and "Holiday 2".)

Normal day	Monday, Tuesday, Wednesday, Thursday, Friday
Holiday 1	Saturday
Holiday 2	Sunday, <mark>Friday</mark>

### D. Date Setting for Special Day:

The "Special day" is for special schedule, such as the special holidays or make-up workdays. Each Season can set maximum 50 Special days. The searching priority of the "Special day" is higher than the priority of Holiday 2 and Holiday 1 and Normal day. Each enabled "Special day" date must select a Schedule number  $(1 \sim 5)$  to be applied.



# 17.5.5 Schedule Configuration

Each Season can set up maximum 5 Schedules (Schedule  $1 \sim 5$ ), and each Schedule can set up maximum 15 Time Periods. The time unit is "minute", in the range of "00:00 ~ 24:00".

		-								
		El No	rmal day				F	2.14	Nak T	
· Nom	al day		Apply					licher	tulo 2	
C Holiday 1										
O Holid	lav 7		IC Tree					licher	tale 2	
			(2) Web							
() Speci	al day		(2) The (2) Feb					Dilas	fule 4	
			(C) Poo						fule 5	
			0000					Polye	1973	
						om				
	Hour	Minute	To	Hour	Minute		Boolea	n Integer	Real	٦
<b>7</b> 01:	Hour 8 🗸	30 👻	То	12 👻	Minute 0 🗸		Boolea	n Integer • 10	Real 12.34	٦
<b>7</b> 02:	8 <b>v</b> 13 <b>v</b>	30 ▼ 0 ▼	То	12 <b>•</b> 17 <b>•</b>	Minute 0 • 30 •		ON ON	<ul> <li>↓</li> <li>↓</li></ul>	12.34 25.67	
7 02: 03:	8 <b>•</b> 13 <b>•</b> 0 <b>•</b>	30 ▼ 0 ▼ 0 ▼	То	12 • 17 •	Minute 0 • 30 •		ON ON OFF	<ul> <li>▼</li> <li>10</li> <li>▼</li> <li>20</li> <li>▼</li> <li>0</li> </ul>	12.34 25.67 0	
7 02: 03: 04:	8 • 13 • 0 •	30 • 0 • 0 •	То	12 • 17 • 0 •	Minute 0 • 30 • 0 • 0 • 0 • 0 • 0 • 0 • 0 • 0		ON ON OFF OFF	<ul> <li>▼</li> <li>10</li> <li>▼</li> <li>20</li> <li>▼</li> <li>0</li> <li>0</li> </ul>	12.34 25.67 0 0	
<ul> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> </ul>	8 <b>•</b> 13 <b>•</b> 0 <b>•</b>	30 ▼ 0 ▼ 0 ▼	To	12 • 17 •	Minute 0 • 30 •		ON ON OFF	<ul> <li>▼</li> <li>10</li> <li>▼</li> <li>20</li> <li>▼</li> <li>0</li> </ul>	12.34 25.67 0	
<ul> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> </ul>	8 • 13 • 0 • 0 •	30 • 0 • 0 • 0 •	To	12 • 17 • 0 • 0 •	Minute 0    30     0		ON ON OFF OFF OFF	<ul> <li>▼</li> <li>10</li> <li>▼</li> <li>20</li> <li>▼</li> <li>0</li> <li>0</li> <li>0</li> <li>0</li> </ul>	12.34 25.67 0 0 0	
<ul> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> <li>08:</li> </ul>	8 • 13 • 0 • 0 • 0 • 0 • 0 • 0 • 0 •		То	12 • 17 • 0 • 0 • 0 • 0 • 0 •	Minute 0		ON OFF OFF OFF OFF OFF OFF	▼     10       ▼     20       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0	12.34 25.67 0 0 0 0 0 0 0	
<ul> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> <li>08:</li> <li>09:</li> </ul>			То		Minute 0		ON OFF OFF OFF OFF OFF OFF	▼     10       ▼     20       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0	12.34 25.67 0 0 0 0 0 0 0 0 0	
<ul> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> <li>08:</li> <li>09:</li> <li>10:</li> </ul>			То		Minute 0		ON OFF OFF OFF OFF OFF OFF OFF	▼     10       ▼     20       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0	12.34 25.67 0 0 0 0 0 0 0 0 0 0 0	
<ul> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> <li>08:</li> <li>09:</li> <li>10:</li> <li>11:</li> </ul>			То		Minute 0		ON OFF OFF OFF OFF OFF OFF	▼     10       ▼     20       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0       ▼     0	12.34 25.67 0 0 0 0 0 0 0 0 0	
<ul> <li>2 02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> <li>08:</li> <li>09:</li> <li>10:</li> <li>11:</li> <li>12:</li> </ul>			То		Minute 0		ON OFF OFF OFF OFF OFF OFF OFF OFF	<ul> <li>▼</li> <li>10</li> <li>▼</li> <li>20</li> <li>0</li> <li< td=""><td>12.34 25.67 0 0 0 0 0 0 0 0 0 0 0 0 0</td><td></td></li<></ul>	12.34 25.67 0 0 0 0 0 0 0 0 0 0 0 0 0	
<ul> <li>01:</li> <li>02:</li> <li>03:</li> <li>04:</li> <li>05:</li> <li>06:</li> <li>07:</li> <li>08:</li> <li>09:</li> <li>10:</li> <li>11:</li> <li>12:</li> <li>13:</li> <li>14:</li> </ul>			То		Minute 0		ON OFF OFF OFF OFF OFF OFF OFF OFF OFF	<ul> <li>▼</li> <li>10</li> <li>▼</li> <li>20</li> <li>0</li> <li< td=""><td>12.34 25.67 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td><td></td></li<></ul>	12.34 25.67 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

### EX: The following setting is correct.

No.	Time Period	Boolean	Integer	Real
01	00:00 ~ <b>08:00</b>	OFF	100	30
02	<b>08:00</b> ~ 12:00	ON	150	25.5
03	<b>12:00</b> ~ 13:00	OFF	120	27
04	<b>13:00</b> ~ 17:00	ON	150	25.5
05	<b>17:00</b> ~ 24:00	OFF	100	30

### The Searching Priority of Time Period:

The searching priority of the Time Period in the schedule is in the order from the largest number to the smallest number.

No.	Time Period	Boolean	Integer	Real
01	00:00 ~ 08:00	OFF	100	30
02	08:00 ~ 12:00	ON	150	25.5
03	12:00 ~ 13:00	OFF	120	27
04	13:00 ~ 17:00	ON	150	25.5
05	17:00 ~ 24:00	OFF	100	30

For example, the following table shows five Time Periods settings .

- 1. The searching will in the order from No. 5 to No. 1 (05, 04, 03, 02, 01). If the Time Period overlaps, the PAC will follow the larger number setting to control the schedule.
- If the PAC cannot find the current time in any Time Period in the "15" ~ "01", it follows the setting of "Default Value".

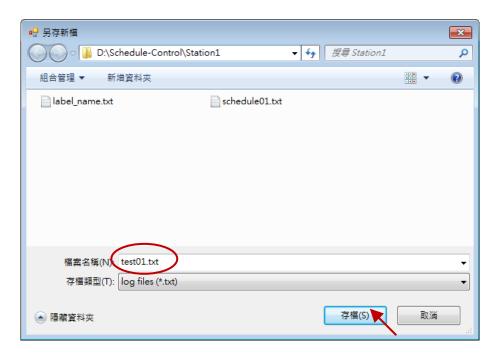
Schedule	1														
									Copy fro	m					
	Но	ur	Min	ute	То	Hor	n	Min	ute		Boole	an	Integer	Real	
<b>V</b> 01:	0	•	0	-		8	•	0	-		OFF	•	100	30	
<b>V</b> 02:	8	-	0	-		12	-	0	•		ON	•	150	25.5	
<b>V</b> 03:	12	•	0	-		13	•	0	•		OFF	•	120	27	
<b>V</b> 04:	13	•	0	-		17	•	0	•		ON	•	150	25.5	
<b>V</b> 05:	17	•	0	-		24	•	0	•		OFF	•	100	30	
06:	0	<b>-</b>	0	-		0	<b>-</b>	0			OFF	-	0	0	
07:	0	- T	0	-		0	<b>-</b>	0			OFF	-	0	0	
08:	0	-	0	-		0	<b>_</b>	0			OFF	-	0	0	
09:	0	-	0	- ×		0	<b>.</b>	0	- <b>-</b>		OFF	-	0	0	
10:	0	<b>.</b>	0	- ×		0	<b>.</b>	0	- <b>-</b>		OFF	-	0	0	
<b>11</b> :	0	<b>-</b>	0	- T		0	×.	0	- <b>-</b>		OFF	-	0	0	
12:	0	<b>_</b>	0	- ×		0	<b>.</b>	0	- <b>-</b>		OFF	-	0	0	
13:	0	-	0	-		0	<b>_</b>	0	-		OFF	-	0	0	
14:	0	-	0	-		0	-	0			OFF	-	0	0	
15:	0	Ŧ	0	-		0	Ŧ	0			OFF	-	0	0	
Save	and exit													Canc	el

## 17.5.6 Save and Send the File to the PAC

After completing the configurations, please save and then send it to the PAC:

💀 Schedule-Control Utility	
Save to PC Send to Controller	Controller time synchronization Open from PC Get from Controller
1. 2.	

1. Click "Save to PC" to save the configuration file (" \*.txt ").



2. Click "Send to Controller" to send the configuration file to the linked PAC. Please assign the PAC IP address and set up the password (default: 0). Check the "Remember password" can save the password for speeding the next sending process.



# 17.5.7 Time Synchronization

If the PAC has not synchronized the system time after working a long period (e.g. one year), the time may be differ over 10 seconds to a few minutes. For the time synchronization of the controller, the Schedule-Control Utility provides a function to set the PAC time from the PC.

🖳 Schedule-Control Utility		
Save to PC Send to Controller	Controller time synchronization	Open from PC Get from Controller

#### Steps:

- 1. Your PC must connect to an Ethernet Switch and then to the Win-GRAF PAC by using Ethernet cables.
- 2. Click "Controller Time Synchronization" button, and enter the current IP address of the PAC and the password (defaults: 0).
- 3. Set a new date, hour, minute and second.
- 4. Click "Set new Controller time" button to set the new time to the PAC.

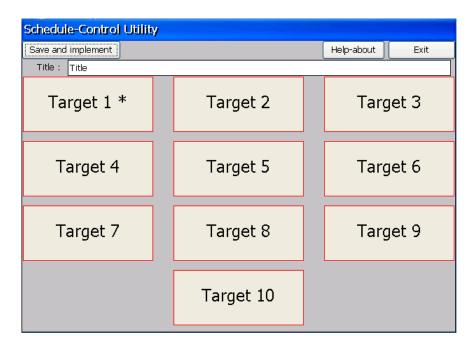
IP :	192.168.75.101
Password :	•
	🔽 Remember password

🖳 Controller time setting				<b>X</b>
		Hour	Minute	Second
Controller time :	2017年 7月 7日 📃 🔻	11	45	20
Set new Controller time :	2017年7月7日 🔍 🗸	11 👻	<b>43</b> •	30 🗸
			Set	new Controller time

#### 17.5.8 Schedule-Control Utility in PAC Site

The configuration process on the PAC side is similar as the process in the PC side, just a little bit different on the screen. Please refer the Section  $17.5.2 \approx 17.5.6$ .

1. Target Configuration:



#### 2. Season Configuration:

Schedule-Contr	rol Utility	Target 1 > 9	Season Always	> Normal day
Save and implemer	nt Back		Help-abo	out Exit
Targ	et 1 🗸 Enable	Boolea Default Value OFF		Real
Season A	lways 🖌 Enable	Always		
Season Always *	Season 1	Season 2	Season 3	Season 4
<ul> <li>Normal day (Sch</li> <li>Holiday 1</li> <li>Holiday 2</li> <li>Special day</li> </ul>		inable spply Schedule 1 * Sunday Monday Tuesday Wednesday Hursday Friday Saturday		Schedule 1 * Schedule 2 Schedule 3 Schedule 4 Schedule 5

### 3. Special Day Configuration:

e uu	
Setting	
Target 1 > Season Always > Sp	pecial day
Always	
	Date info
	Date No. 1
	Enable
	Null
	Apply
	Delete
Add a new date	
Save Setting Clear all	Exit

## 4. Schedule Configuration:

Schedule	1			
Period No.	Hour Minute	~ Hou	ur Minute	Boolean Integer Real
01 🔽 🛛	8 💌 30 💌	12	O	ON 🔽 10 12.34
Copy From	Start		End	(Boolean , Integer , Real )
<b>V</b> 01:	08:30	~	12:00	(ON, 10, 12.34)
<b>V</b> 02:	13:00	~	17:30	(ON, 20, 25.67)
03:	00:00	~	00:00	(OFF,0,0)
04:	00:00	~	00:00	(OFF,0,0)
05:	00:00	~	00:00	(OFF,0,0)
06:	00:00	~	00:00	(OFF,0,0)
07:	00:00	~	00:00	(OFF,0,0)
08:	00:00	~	00:00	(OFF,0,0)
09:	00:00	~	00:00	(OFF,0,0)
10:	00:00	~	00:00	(OFF,0,0)
11:	00:00	~	00:00	(OFF,0,0)
12:	00:00	~	00:00	(OFF,0,0)
13:	00:00	~	00:00	(OFF,0,0)
14:	00:00	~	00:00	(OFF,0,0)
15:	00:00	~	00:00	(OFF,0,0)
Save and e	xit			Ca

### 17.5.9 Using Schedule-Control in the eLogger HMI

eLogger is a free charge and an easy-to-use HMI software platform developed by ICP DAS. It can be used to design the Local HMI and the Web Server HMI for remotely controlling the PAC through a web browser on your PC or cell phone. All Win-GRAF PAC support eLogger HMI.

For instructions on eLogger HMI, visit the following web page for the Win-GRAF FAQ-018 and FAQ-019:

<u>www.icpdas.com</u> > Support > FAQ > <u>Win-GRAF Soft-Logic PAC</u> > <u>FAQ-018</u>, <u>FAQ-019</u> or <u>http://www.icpdas.com/root/support/faq/win-graf.php</u>



You can download the sample program ("Demo\_faq018\_all. zip" or demo\_faq019\_all. zip) directly on the Win-GRAF FAQ page, or in Win-GRAF-PAC-CD (\napdos\win-graf\demo\_ project\), and then operate and test the project according to the contents of the document.

## Chapter 18 Develop Your Own Function and Function Block

This section described how to use the Visual Studio 2008 development tool to produce a DLL file of your own Function or Function Block. All the related demo files are included in the PAC's CD-ROM, such as the XP-8xx8-CE6, WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7, and VP-x2x8-CE7 PAC.

#### **Related demo files:**

CD-ROM : \napdos\Win-GRAF\demo-project\user\_c\_lib\

- \demo\_user\_c : The VS 2008 project folder, can be used to build the "user\_c.dll" file for your own Function and Function Block. (Refer the <u>Section 18.4</u>)
- ..\user\_c.dll : The pre-compiled DLL file "user\_c.dll" used for the Function ("bytes\_to\_long") and the Function Block ("long\_to\_bytes") in this demo. \wp\_vp\user\_c.dll : for WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7, and WP-5xx8-CE7. \xpc\user\_c.dll : for XP-8xx8-CE6.
- \user : The Win-GRAF Lib folder, including library files of the Function ("bytes\_to\_long") and the Function Block ("long\_to\_bytes") in this demo. (Refer the <u>Section 18.3</u>)

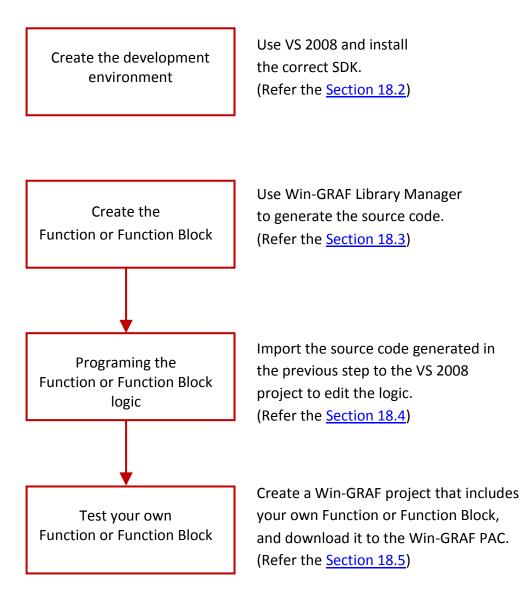
#### Win-GRAF demo file:

CD-ROM : \napdos\Win-GRAF\demo-project\demo\_user\_c.zip

The standard shipments of ICP DAS Win-GRAF PAC do not include the "user\_c.dll" file. If you want to add your own Function and Function Block for working with the Win-GRAF logic, you must build a DLL file and copy it to the same folder as the Win-GRAF driver on the PAC (i.e., \System\_disk\Win-GRAF\).

After rebooting the PAC, the Win-GRAF driver will load that DLL file to support your Function and Function Block.

## **18.1** The Development Process of Your Own Function or Function Block



## 18.2 Creating the Compiler Development Environment

#### Download the SDK (Software Development Kit):

The user can download the related SDK on the website:

- For XPAC (XP-8xx8-CE6) <u>ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/sdk/platformsdk/</u> (pacsdk\_ce\_x.x.x\_vs2008.msi)
- For ViewPAC (VP-x2x8-CE7) and WinPAC (WP-8xx8, WP-8xx8-CE7, WP-5xx8-CE7) <u>ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x\_ce50/sdk/platformsdk/</u> (pac270\_sdk\_yyyymmdd.msi)

#### 18.2.1 Install the SDK of the ViewPAC or the WinPAC

**<u>Note</u>**: Make sure your PC has been installed the Microsoft VS2008 before doing the following steps.

1. Double click the downloaded SDK file (e.g., pac270\_sdk\_20121015.msi) to install it to the VS2008.



2. Click the "Next" button.



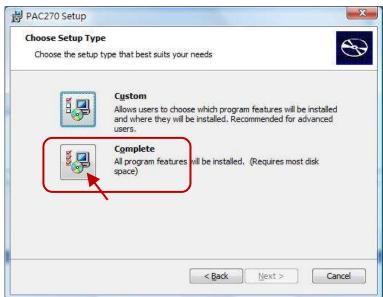
3. Choose the "Accept" radio button and then click the "Next" button.

ind-User License Agreement	6
e sure to carefully read and understand the following end-user license agreement, and then indicate whether you accept or do not accept the terms of the agreement.	C
his software will not set up on your computer unless you accept the terms of the agr or your future reference, you may print the text of the agreement now using the PR jutton or obtain the text from the 'MSFT_SDK_EULA_1033.rtf file after installation. Yi lso receive a copy of this agreement by contacting the Microsoft subsidiary serving y ountry, or write: Microsoft Sales Information Center, One Microsoft Way, Redmond V 8052-6399.	INT ou may our
END-USER LICENSE AGREEMENT FOR	-
	1.000
MICROSOFT SOFTWARE	
MICROSOFT SOFTWARE MICROSOFT CUSTOM SOFTWARE DEVELOPMENT KIT FOR WINDOWS CE 5.0 (Referred to as the "Microsoft Custom SDK")	
MICROSOFT CUSTOM SOFTWARE DEVELOPMENT KIT FOR WINDOWS CE 5.0	*
MICROSOFT CUSTOM SOFTWARE DEVELOPMENT KIT FOR WINDOWS CE 5.0 (Referred to as the "Microsoft Custom SDK")	Ŧ

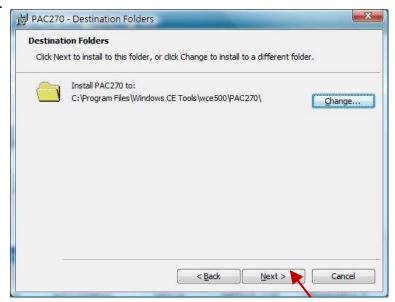
4. Click the "Next" button.

post the manual line -		-X
ion		G
< <u>B</u> ack	Next >	Cancel

5. Click the "Complete" button.



6. Click the "Next" button.



7. Click the "Install" button to install the SDK.



8. After completing the installation, click the "Finish" to quit the procedure.



### 18.2.2 Install the SDK of the XPAC (XP-8xx7-CE6, XP-8xx7-Atom-CE6)

#### **Note:** Make sure your PC has been installed the Microsoft VS2008 before doing the following steps.

1. Double click the downloaded SDK file (e.g., pacsdk\_ce\_1.4.3\_vs2008.msi) to install it to the VS2008.



2. Other steps, refer the <u>Section 18.2.1</u> – Step (2)  $\sim$  (8).

## **18.3 Define Function or Function Block**

#### 18.3.1 Define Function Lib

This section presents a simple example of creating a library function - "bytes\_to\_long" that is used to convert four bytes ( $0 \sim 255$ ) to one long integer (32-bit Signed Integer).

#### To begin, follow these steps:

1. Click All Programs  $\rightarrow$  Win-GRAF  $\rightarrow$  Libraries  $\rightarrow$  OEM from the Start menu as the figure below. <u>Note:</u> If the Win-GRAF Workbench is opened, the user cannot add or edit Win-GRAF Library.



2. Create a library group named "User" in order to easy maintenance and management.

🚾 Library Manager - User1	Open Library
File Tools Wizard Help Open Library Save Library Use in SAMA Exit	Used Advanced Arithmetic Arrays AS-interface BACnet Booleans CANbus CANopen Clock Comparisons Conversions Conversions Conversions Counters DNP3 Embedded HMI Select a library in the list or enter a name for creating a new library.

3. Select the "Function and FBs" tab and click the "New" button.

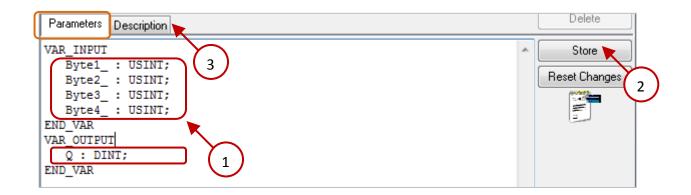
File Tools Wizard Help		W 96
Function and FBs 1/Os Profiles AS-i Types		
		New
		Rename
Parameters Description		Delete
	*	Store

4. Enter the name and comment, and select the type as a Function and then click the "OK" button.

_					
	1				
C		/	,	/	/

5. Declare the prototype of this Function.

The user can declare parameters like this way - "variable name : data type;", and refer <u>Appendix A</u> for the data type. The content between the "VAR\_INPUT" and "END\_VAR" are passed-in parameters; the content between the "VAR\_OUTPUT " and "END\_VAR" is returned parameter. After completing it, click "Store" to save. Then, click "Description" tab to add the technical notes for this Function and then click "Store" to save.



6. Click "Wizard"  $\rightarrow$  "Function/Block Source Code" to generate the source code of this Function.

🕌 Library Mar	nager - User	
File Tool	Wizard Help	
Function and F	Function / Block Source Code	
Bytes_to_lon	<b>ug</b> (* Convert 4 bytes to 1 long *)	

7. Select "Windows DLL" and click "Copy" button to copy the source code.

Nizard	×
/* bytes_to_long - Source code (requires t5vm.h include) */ /* Function */	
/* Argument list */	
#define _P_BYTE1_ ("((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[0])) #define _P_BYTE2_ ("((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[1])) #define _P_BYTE3_ ("((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[2])) #define _P_BYTE4_ ("((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[3])) #define _P_Q ("((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[4]))	
/* handler */	
T5_DWORDdeclspec(dllexport) BYTES_T0_LONG ( T5_WORD wCommand, T5PTR_DB pBase, T5_PTR pClass, T5_PTR plast, T5_PTWORD pArgs) { switch (wCommand) { case T5FBCMD_ACTIVATE : /* enter your code here */ return 0L; case T5FBCMD_ACCEPTCT : /* dont change that */ return 1L; default : return 0L; }	T.
Implementation	
Statically linked	Сору 📐
Windows DLL	Close

8. Paste the source code into the text editor (e.g., Notepad) and save it as "bytes\_to\_long.c".

bytes_to_long.c - 記事本	x
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)	
<pre>/* bytes_to_long - Source code (requires t5vm.h include) */ /* Function */ #include "stdafx.h" #include "t5vm.h" /* Argument list */</pre>	* III
<pre>#define _P_BYTE1 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[0])) #define _P_BYTE2 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[1])) #define _P_BYTE3 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[2])) #define _P_BYTE4 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[3])) #define _P_Q (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[4]))</pre>	Ŧ

#### 18.3.2 Define Function Block Lib

This section uses the similar way as previously described to define the Win-GRAF library of the Function Block (e.g., "long\_to\_bytes") and then create a file of source code (e.g., "long\_to\_bytes.c").

1. Refer the <u>Previous Section</u> to open "Library Manager" (All programs  $\rightarrow$  Win-GRAF  $\rightarrow$  Libraries  $\rightarrow$  **OEM**) and open the library group - "User", and then click "New" button to add a Function Block.

File Tools W	izard Hel	p		
Function and FBs	1/0s Pro	files AS-i	Types	
bytes_to_long	(* Convert 4	bytes to 1 lon	g <b>*</b> )	New 🔪
				Rename
Parameters Desc	ription			Delete

2. Enter the name and comment, and select the "Function Block" type and then click the "OK" button.

Name:	long_to_bytes	
Comment:	Convert 1 long to 4 bytes	Cancel
0 1/0 Dev		
Function		

3. Declare the prototype of this Function Block.

The user can declare parameters like this way - "variable name : data type;", and refer <u>Appendix A</u> for the data type. The content between the "VAR\_INPUT" and "END\_VAR" is passed-in parameter; the content between the "VAR\_OUTPUT" and "END\_VAR" are returned parameters. After completing it, click "Store" to save. Then, click "Description" tab to add the technical notes for this Function Block and then click "Store" to save.

Parameters Description	Delete
VAR_INPUT Long_Val_: DINT; END_VAR VAR_OUTPUT Byte1_: USINT; Byte2_: USINT; Byte3_: USINT; Byte4_: USINT; LEND_VAR 1	Store Reset Changes 2

4. Click "Wizard"  $\rightarrow$  "Function/Block Source Code" to generate the source code of this Function Block.

Library Manager - User	
File Tool Wizard Help	
Function and F Function / Block Source Code 💌	
bytes_to_long (* Convert 4 bytes to 1 long *)	New
<pre>along_to_bytes (* Convert 1 long to 4 bytes *)</pre>	Rename
Parameters Description	Delete

5. Select "Windows DLL" and click "Copy" button to copy the source code.

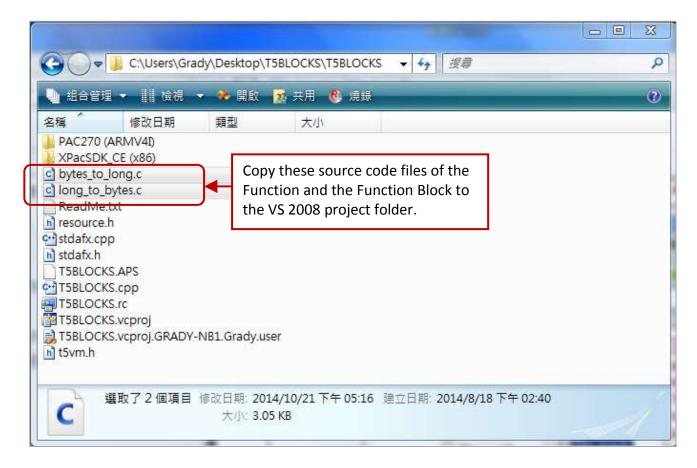
** long_to_bytes - Source code (requires t5vm.h include) */ ** Function block */	<u>^</u>
* Argument list */	
tdefine_P_LONG_VAL_ (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[0])) tdefine_P_BYTE1_ (*((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[1])) tdefine_P_BYTE2_ (*((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[2])) tdefine_P_BYTE3_ (*((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[3])) tdefine_P_BYTE4_ (*((T5_PTBYTE)(T5GET_DBDATA8(pBase))+pArgs[4]))	E
* private block data */	
ypedef struct	
T5_DWORD dwData; /* T0D0: replace dwData by the items you need */ _str_F8_L0NG_T0_BYTES;	
* handler */	
<sup>15</sup> _DWORDdeclspec(dllexport) LONG_TO_BYTES ( T5_WORD wCommand, T5PTR_DB pBase, T5_PTR pClass, T5_PTR pInst, T5_PTWORD pArgs) _str_FB_LONG_TO_BYTES *pData;	
pData = [_str_FB_LONG_TO_BYTES *)pInst; switch (wCommand)	
{ case T5FBCMD_ACTIVATE :	+
Implementation	
Statically linked	Сору 📐
Windows DLL	Close

6. Paste the source code into the text editor (e.g., Notepad) and save it as "bytes\_to\_long.c".

o long_to_bytes.c - 訂事本	x
檔案( <del>F) 編輯(E)</del> 格式(O) 檢視(V) 說明(H)	
₩include "stdafx.h" #include "t5vm.h" /* long_to_bytes – Source code (requires t5vm.h include) */ /* Function block */	* III
/* Argument list */	
<pre>#define _P_LONGVAL (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[0]) #define _P_BYTE1 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[1])) #define _P_BYTE2 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[2])) #define _P_BYTE3 (*((T5_PTLONG)(T5GET_DBDATA32(pBase))+pArgs[3]))</pre>	))

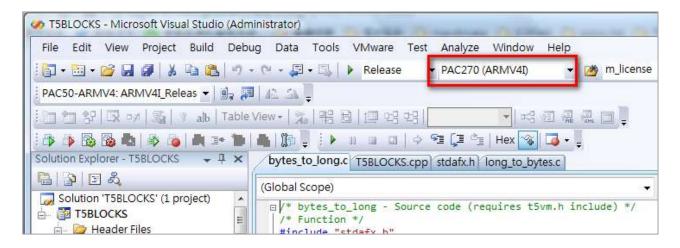
## 18.4 Edit the Logic of the Function and Function Block

- <u>Note:</u> Please make sure your PC has installed the Visual Studio 2008 and WinPAC/XPAC SDK before operating the following steps.
  - As the figure below, copy these source code files of the Function and the Function Block (described in <u>Section 18.3</u>) to the VS 2008 project folder on your PC. (Or, get the VS 2008 sample project folder from the CD-ROM: \napdos\Win-GRAF\demo-project\user\_c\_lib\demo\_user\_c)



Then, make sure your VS 2008 project settings are correct for your PAC.
 (Note: the settings are different between different controllers.)
 For WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7, and WP-5xx8-CE7, it must set to "PXA270.

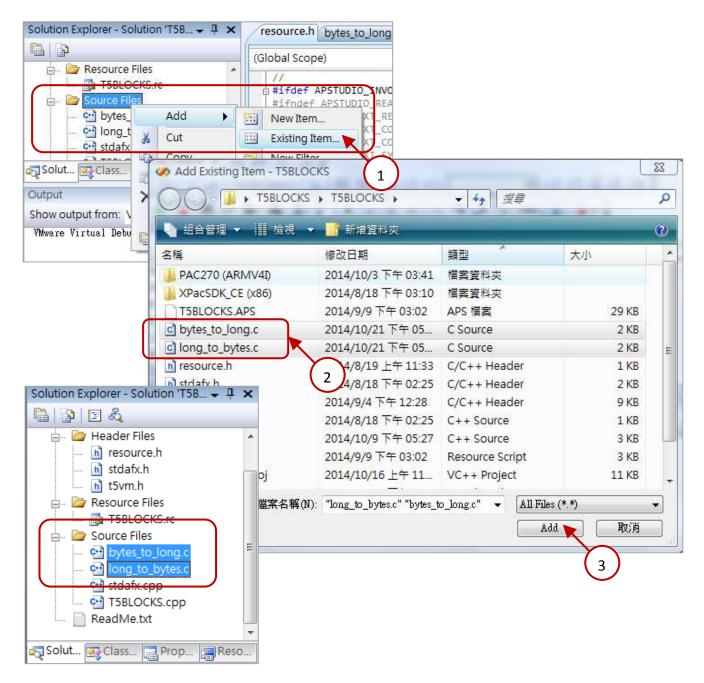
For WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7, and WP-5xx8-CE7, it must set to "PXA270 (ARMV4I)".



For XP-8xx8-CE6, it must set to "XPacSDK (x86)".

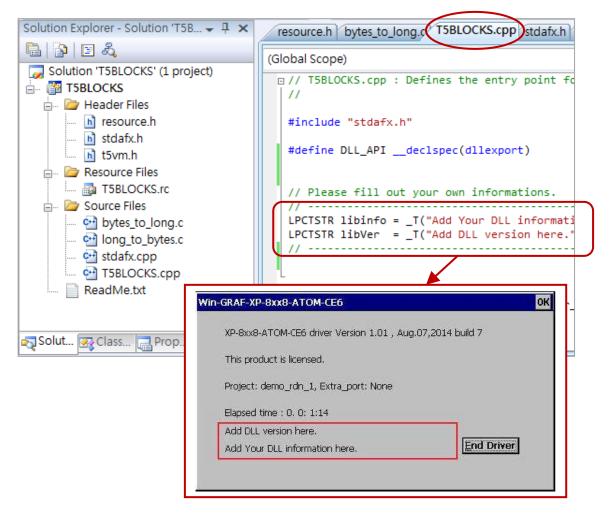
File Ec	lit View	Project	Build	Debug	Data	Tools	VMware	Test	Analyze Win	dow Help		
🗗 • 🛅	- 💕 🖬		<b>b B</b>	117 - (	+ 📮	• 💷	Release		XPacSDK_CE ()	(86)	🖌 对 m_licens	е
XPacSD	CE x86 D	evice	-	h, 📮   ,	12 SN .	-						
i da ta	양 😡 이		alb	Table Vie	w -   🖧	198	8 9	명비	*			
	B 😼 🖏	i 🕸 🐞			111 -		11 11 12		Si 🗊 👘 Hex	· 😭 🗔 -	Ţ	
Solution E	xplorer - T	5BLOCKS	I	1 ×	bytes_t	o_long.	C T5BLOCK	S.cpp	stdafx.h long_	to_bytes.c		
					Global S	(ego						
ф. 🚰 Т ф. ն	ion 'T5BLC 5BLOCKS Header F	iles	roject)	· III	⊡ /* by   /* Fu   #inc]	tes_to nction		ource	code (require	es t5vm.h	include) */	

 Here, we will add C source files of the Function and Function Block to the VS 2008 project. Mouse right-click on the "Source Files" folder and click "Add" → "Existing Item...", select the previously copied C source files and click the "Add" button.

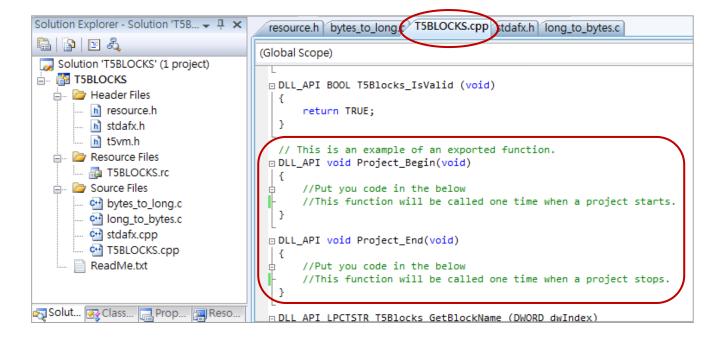


#### 18.4.1 Edit the "T5BLOCKS.cpp"

Fill out your "libinfo" and "libVer" information in the "T5BLOCKS.cpp" file. The information will show on the Win-GRAF driver dialog on the PAC (as the figure below).

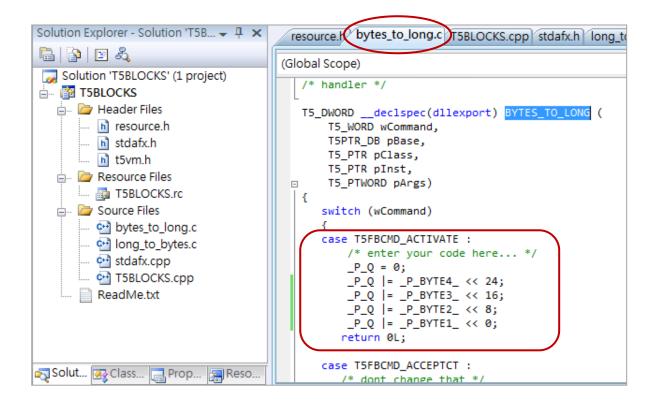


If you have some operations to process when a new Win-GRAF project starts or stops. Please edit the "Project\_Begin" and "Project\_End" functions in the "T5BLOCKS.cpp" file.



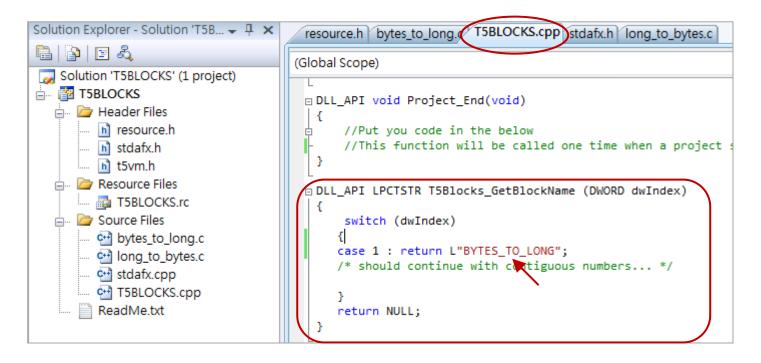
### 18.4.2 Edit the Logic of the Function (In this example is "bytes\_to\_long. c")

First, add your logical expression for this Function in the switch case statement –"T5FBCMD\_ACTIVE".

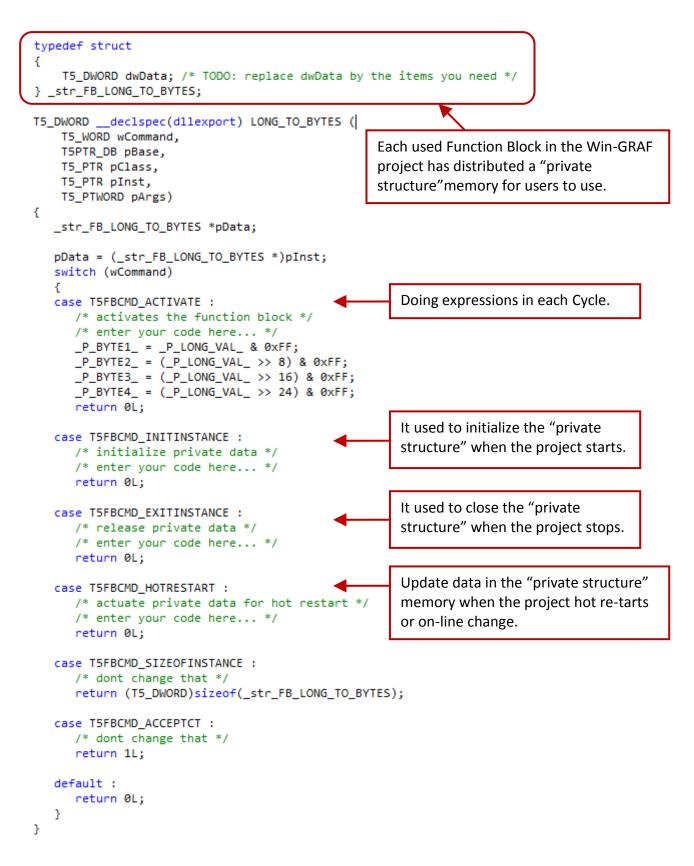


Then, add this Function name (e.g., "BYTES\_TO\_LONG") into the switch case statement of the "T5Blocks\_GetBlockName" functions in the "T5BLOCKS.cpp" file.

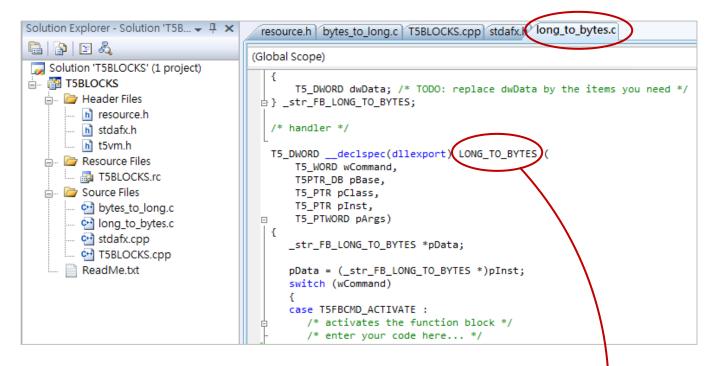
Note: The number of the case label (e.g., case 1) must start from "1" and continue with contiguous numbers.



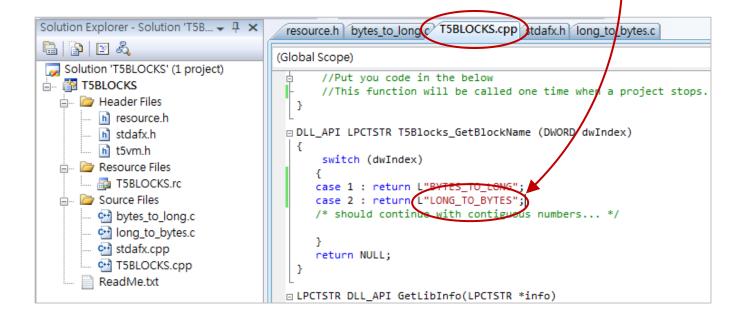
18.4.3 Edit the Logic of the Function Block (In this example is "long\_to\_bytes.c")



Finally, remember to add the Function Block name (e.g., "LONG\_TO\_ BYTES") into the switch case statement of the "T5Blocks\_GetBlockName" functions in the "T5BLOCKS.cpp" file.



<u>Note:</u> The number of the case label (e.g., case 1) must start from "1" and continue with contiguous numbers.



### **18.4.4** Trying to Compile the Project

1. Click "Build" > "Build Solution" to generate a DLL file.

File Edit View Project	Build	Debug Data Tools	VMware	e Te	est	Analyze	Window
🐻 • 📴 • 🚰 🖌	齸	Build Solution	F7	se	-	XPacSDK	CE (x86)
XPacSDK_CE x86 Device         Image: Split Sp	<b>E</b>	Rebuild Solution Ctrl+A Deploy Solution Clean Solution Build T5BLOCKS Rebuild T5BLOCKS	lt+F7	F has	\$ 5		Hex 🚳
Solution 'T5BLOCKS' (1 pr		Deploy T5BLOCKS Clean T5BLOCKS		in t	the t	pelow	

2. After a successful compilation, copy the "USER\_C.dll" file to the path "\System\_disk\Win-GRAF\" in your PAC, and then reboot it.

🔚 組合管理 👻 🏢 檢視 👻 📑 Open with KUSO EXIF Viewer 😿 共用	😢 燒錄	_
名稱 *	修改日期	類型
M T5BLOCKS.obj	2014/10/22 下午 05	Object File
T5BLOCKS.pch	2014/9/9下午 06:38	Precompil
T5BLOCKS.pdb	2014/10/22 下午 05	PDB 檔案
T5BLOCKS.res	2014/10/9 下午 04:25	Compiled
SUSER_C.dll	2014/10/22 下午 05	應用程式擴
USER_C.pch	2014/10/9 下午 04:24	Precompil
🕐 vc80.idb	2014/10/22 下午 05	IDB 檔案
vc80.pdb	2014/10/22 下午 05	PDB 檔案

### 18.5 Test your own Function and Function Block

1. Copy the "user\_c.dll" file to the same folder as the Win-GRAF driver (i.e., \System\_disk\Win-GRAF\) on the PAC by using FTP. And, reboot the PAC.

If a proper DLL file is detected by the Win-GRAF Driver, its dialog will show as below.

Win-GRAF-XP-8xx8-ATOM-CE6	ОК
XP-8xx8-ATOM-CE6 driver Version 1.01 , Aug.07	7,2014 build 7
This product is licensed.	
Project: demo_rdn_1, Extra_port: None	
Elapsed time : 0, 0; 1:14	
Add DLL version here.	
Add Your DLL information here.	End Driver

2. Open the Win-GRAF project that includes your own Function and Function Block, and then compiles and download this project to the Win-GRAF PAC.

In addition, there are some available files in the Win-GRAF PAC's CD-ROM for users to test.

- (1) WP-8xx8, WP-8xx8-CE7, VP-x2x8-CE7 and WP-5xx8-CE7:
  - \napdos\Win-GRAF\demo-project\user\_c\_lib\wp\_vp\user\_c.dll
- (2) XP-8xx8-CE6:

\napdos\Win-GRAF\demo-project\user\_c\_lib\xpc\user\_c.dll

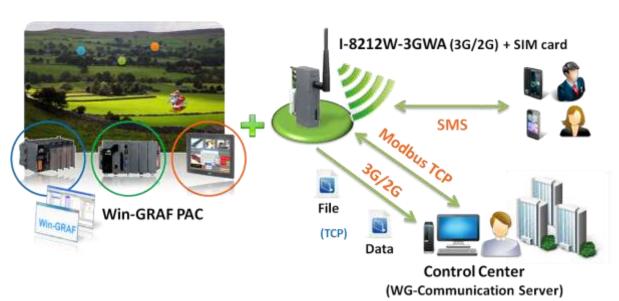
- (3) Copy the Win-GRAF Library folder -"User" to the following path on your PC. C:\Win-GRAF\DATA\HWDEF\
- (4) Open the Win-GRAF sample project "demo\_user\_c.zip", and then compiles and download this project to the Win-GRAF PAC. (Refer the <u>Section 13.1</u> to open the Win-GRAF project from a zip).

## Chapter 19 Using 3G Modules - I-8212W-3GWA

#### Note:

- 1. Due to the product certification issue, the I-8212W-3GWA module can sales in certain areas. Please contact our agents for more information.
- 2. There is one another 3G Solution, that is, the user can buy a 3G Router come with a SIM card. By this way, the Win-GRAF PAC can also connect to the Internet via the 3G network.

Sending back the collected data to the control center is necessary in some application. However, there may be no cable can reach the field or the cost of the network wiring is too expensive. ICP DAS released the "Win-GRAF PAC + I-8212W-3GWA" solution for such applications. Designers can collect I/O data or other application data by program a PLC application (Ladder, ST, Function block, etc.) with Win-GRAF software. Using the device – "I-8212W-3GWA" (insert the SIM card inside that has registered the 3G/2G service from the Telecom Company) to connect Internet by dial-up 3G/2G, then the PLC can send TCP data to the center.



#### **3G/2G Wireless Application**

The following Win-GRAF driver version supports the dial-up 3G/2G access with the I-8212W-3GWA.

XP-8xx8-CE6 : 1.03 or later WP-8xx8 : 1.05 or later VP-x2x8-CE7 : 1.01 or later

If the Win-GRAF driver version of your PAC is older than the above listed version, please visit the <u>http://www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.ttml</u> to download the newer driver.

I-8212W-3GWA: http://m2m.icpdas.com/i-8212w-3GWA.html

## 19.1 Hardware Installation

The I-8212W-3GWA supports 3G/2G wireless communication. Insert the 3G SIM card (that registered the 3G/2G function from the Telecom Company) into the "SIM card" socket of this 3G/2G module and make sure the antenna has installed well.

If your PAC is **XP-8xx8-CE6**, plug the I-8212W-3GWA in its slot **1**.

(I.e., the leftmost I/O slot of the XPAC).

If your PAC is **WP-8xx8 or VP-x238-CE7**, please plug the I-8212W-3GWA in its slot **0**.

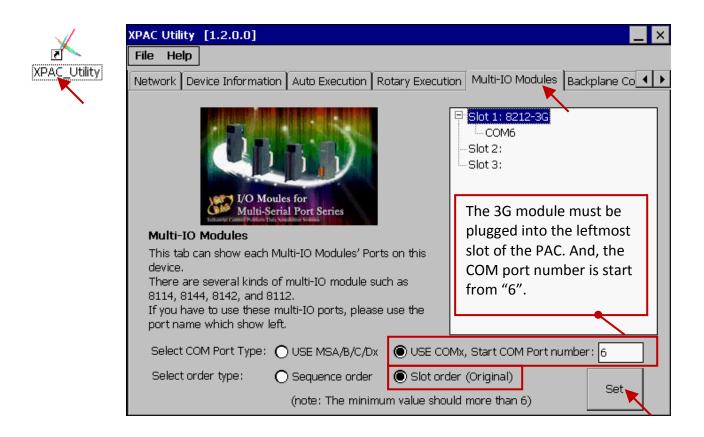
(I.e., the leftmost I/O slot of the WinPAC. The slot number of ViewPAC is shown on the back).

Then power on the PAC and run PAC Utility (for example, run "XPAC\_Utility" for the XPAC) to set up the "COM6" port of the I-8212W-3GWA. Remember to run "File > Save and Reboot" once to save the settings.

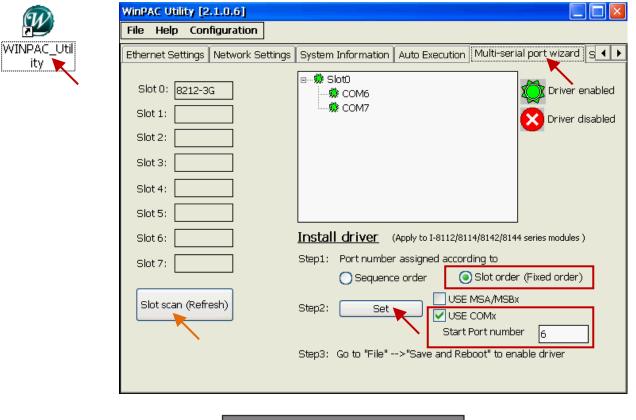
The user can find out the PAC Utility on the desktop or the PAC folder (e.g., \System\_Disk\Tools\ XPAC\_Utility), or download the Utility on FTP:

XPAC: <u>ftp://ftp.icpdas.com/pub/cd/xp-8000-ce6/system\_disk/tools/</u>
 WinPAC: <u>ftp://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x\_ce50/system\_disk/tools/</u>
 ViewPAC: <u>ftp://ftp.icpdas.com/pub/cd/winpac/napdos/vp-4000\_ce50/system\_disk/tools/</u>

#### **XPAC\_Utility:**



#### WinPAC\_Utility:



Inpu	t F	an	el									
Esc] 1	.]2	2]3	[4	[5	6	7	8	9	0	-	=	•
[Tab]	٩Į	w[	e [	rΙ	t[	УĮ	u[	i	0	p	]]	]
[CAP]	a	s	d	f	g	h	j	k	[]	[;	Ŀ	Л
Shift	:	X	<u> </u>	Į٧	Įь	<u>In</u>	<u>Im</u>	Ι,	Ι.		<u>'</u> L	€ .
[Cti]á	iüĮ	١	١I					Ι	ΨI	1	←	<b>→</b>



Finally, run "File > Save and Reboot" in the PAC Utility.

XPAC	XPAC Utility [1.2.0.0]							
File	Help							
Sav	Save							
Sav	Save and Reboot							
<u>R</u> eboot 📉								
Restore Utility Default Settings								
E <u>x</u> it								

## 19.2 Software Installation

Please check the Win-GRAF driver version for your PAC is the correct version that listed in the <u>Section 19</u> (P19-1). If not, update it.

#### 19.2.1 Install the I-8212W-3GWA (or I-8213-3GWA) Driver

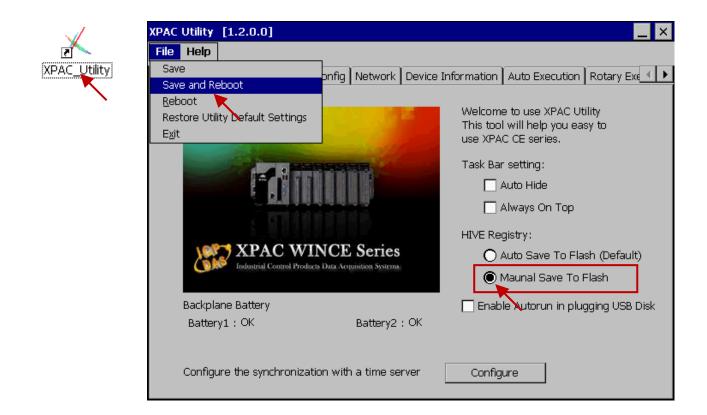
Note: Plug the I-8212W-3GWA into the I/O slot**0** of the WinPAC or ViewPAC. In addition, plug it into the I/O slot**1** of the XP-8xx8-CE6.

Double-click the "icpdas\_i-821xw\_com6\_vx.xx.cab" file in the path of Win-GRAF PAC -

**\System\_Disk\Win-GRAF\** to install the I-8212W-3GWA driver if the PAC is the XP-8xx8-CE6, WP-8xx8 or VP-x238-CE7.

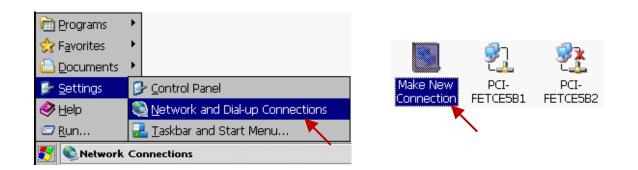
Address \System_Disk\Win-GRAF		
Name	Size	Туре
🗁 sofgrafy		File Folder
🔊 dial_up_net_win_graf.dll	4KB	Application Extension
dial_up_utility_win_graf.exe	27.5KB	Application
🔊 dial_up_wip_graf.dll	11.5KB	Application Extension
Pricpdas_i-821xw_com6_v1.00.cab	1.15KB	CAB File
	•	

After completing the installation, remember to open the XPAC Utility (or WinPAC Utility, ViewPAC Utility) and run "File > Save and Reboot" to save the settings, then the PAC will restart automatically once. (In the below figure, we use XP-8xx8-CE6 as a sample, select "Manual Save To Flash" and then run "File > Save and Reboot").



### 19.2.2 Configure the 3G/2G Dial-up Parameters

At first, get into the "Network and Dial-up Connections" and then run "Make New Connection" on the PAC.



Select "Dial-Up Connection" and type an English name (e.g., "3G", it allows to contain the numbers 0 to 9) and then click the "Next" button. Then, Select the modem - "ICPDAS I-821xW COM6:" and click the "Configure ..." button.

Make New Connection 🛛 🗙	Modem
Type a name for the connection:	
Select the connection type:	Select a modem: CPDAS I-821xW COM6:
Dial-Up Connection     Direct Connection	<u>C</u> onfigure
Virtual Private Network (PPTP)     Virtual Private Network (L2TP)     DPD aver Ethernat (DDDaE1)	<u>T</u> CP/IP Settings Security Settings
○ PPP over Ethernet [PPPoE] < Back Next	< <u>B</u> ack <u>N</u> ext >

In the "Port Settings" tab, select "Baud Rate" as "115200", "Data Bits" as "8", "Parity" as "None", "Stop Bits" as "1" and "Flow Control" as "None".

Device Properties				ок 🗙
Port Settings Call Options				
	о <sub>1</sub>	Connection Pre	eferences —	
<u>Manual Dial (user supplies dial</u> strings)		<u>B</u> aud Rate	115200	
, Terminals		<u>D</u> ata Bits	8	
📇 Use terminal window		<u>P</u> arity	None	-
before dialing		<u>S</u> top Bits	1	-
Use terminal window <u>a</u> fter dialing		Elow Control	None	
	Manual Dial (user supplies dial strings) Terminals Use terminal window before dialing Use terminal window after	Port Settings Lall Options          Manual Dial (user supplies dial strings)         Terminals         Use terminal window         before dialing         Use terminal window after	Manual Dial (user supplies dial strings)       Connection Program         Manual Dial (user supplies dial strings)       Baud Rate         Data Bits       Data Bits         Use terminal window       Barity         Use terminal window       Stop Bits         Use terminal window after       Stop Bits	Manual Dial (user supplies dial strings)       Connection Preferences –         Manual Dial (user supplies dial strings)       Baud Rate       115200         Data Bits       8         Use terminal window       Barity       None         Stop Bits       1

Then, click "Call Options" tab to set up the "Extra Settings" (the settings depend on each of the Telecom Company). For example, the settings provided by a Telecom Company in Taiwan is

### +CGDCONT=1,"IP","INTERNET"

and a Telecom Company in China is

### +CGDCONT=1,"IP","CMNET"

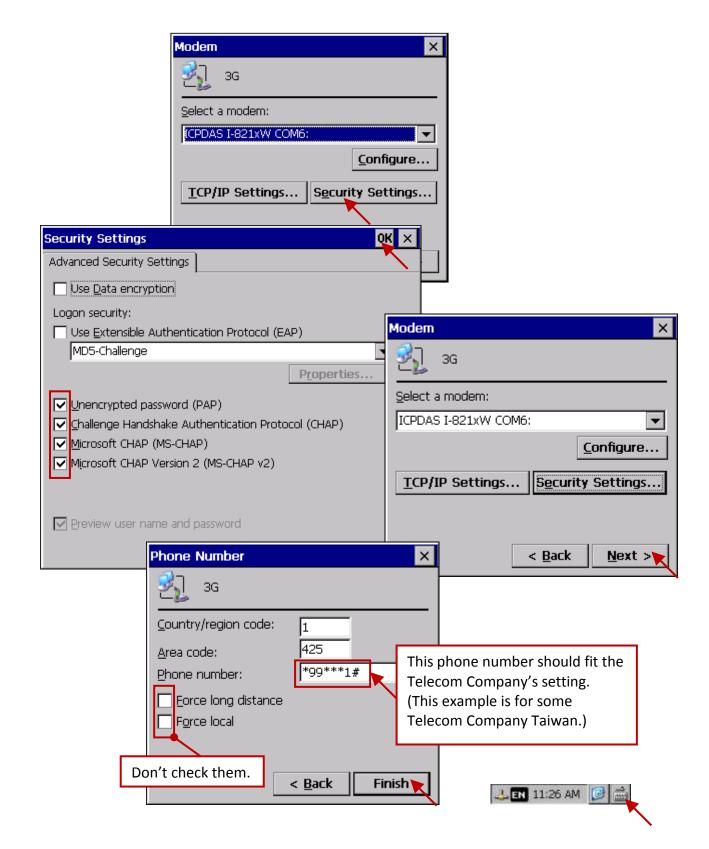
This configuration includes the 3G/2G APN (<u>Access Point Name</u>), please contact your SIM card provider (Telecom Company), to get the settings, or you can also visit the web to search the word "3G/2G APN" to find the settings.

Device Properties	<u>ok</u> ×
Port Setting Call Options	
Call Setup	seconds
✓ Wait for dial tone before dialing Wait for credit card tone o seconds	Must fit the setting provided by the Telecom Company. Or you may Google
Extra Settings (special modem commands may be ins +CGDCONT=1, "IP", "INTERNET"	search for "APN Settings". (This example is for Taiwan's some Telecom Company.)
This part is APN.	😃 EN 11:26 AM 🞯 🚔

Then get into the "TCP/IP Settings ..." dialog box and follow the same settings as below.

Modem	×			
927 3G			1	
Select a modem:	TCP/IP Settings	ок 🗙		
CPDAS I-821xW COM6:	General Name Servers			
<u>C</u> onfig	Se 3G			
TCP/IP Settings. Security Sett	Use server-assigned IP add	ress		
<b>`</b>		TCP/IP Settings		<u>ok</u> ×
		General Name Se	rvers	
< <u>B</u> ack <u>N</u> e	Use software compression	3-		
l				
		Use server-assi	igned addresses	
		<u>D</u> NS:		
		Alt D <u>N</u> S:		
		<u>W</u> INS:		
		Alt W <u>I</u> NS:		

Then get into the "Security Settings" dialog box and follow the same settings as below. Afterward, type the phone number for 3G/2G dial-up, and it must fit for the number provided by Telecom Company, and then click "Finish".



Next, double-click on the new connection (e.g., "3G") that you have created and get into the "Dial Properties" dialog box, and then get into the "Dialing Patterns" to change the content of those three fields as "G" and click "OK".

	Make New Connection	PCI- FETCE5B1	PCI- FETCE5B2	
Dial-Up Connec	tion			×
User Name: Password: Domain:	Save password	Phone: Dial from: Home	*99***1# Dial Properti	ies Cancel
	from: Work	425 1 () <u>T</u> one	▼     New       Dialing Patt       ○       Pulse	OK ×
Ed th FC FC G	or Long Distance calls dial: or International calls dial: (E,e = Country/Region (		I to change how	
				😃 EN 11:26 AM 🕑

Now, you need to make a dial-up connection to check if the 3G/2G network is OK. Please type the "User Name" and "Password" that provided by the Telecom Company or online search the word "3G/2G APN". As figure below, we use a Taiwan SIM card for Telecom Company as an example (keep two fields blank) and then click "Connect" to make the I-8212W-3GWA (plus SIM card) to start dial-up.

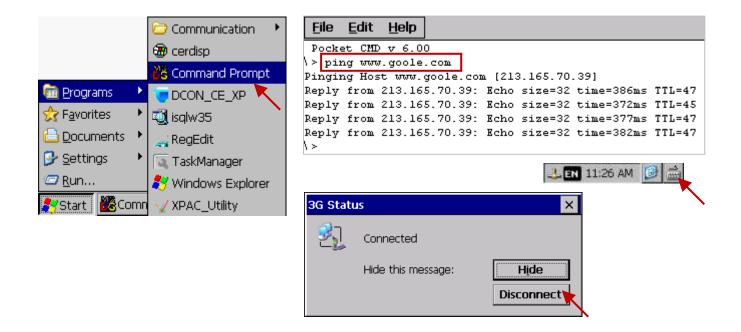
D	ial-Up Connect	ion			×
	<b>€</b> ] 3G				
	<u>U</u> ser Name:		Phone:	*99***1#	
	Password:		Dial from: Home		
	D <u>o</u> main:			Dial Properties	
		Save password	<u>C</u> onr	Cancel	

If the connection is successful, it will show up "Connected".

#### (<u>Note</u>: refer the section 19.2.3 after running the "Ping" command)

3G Status	×
Hide this message:	Hide Disconnect

After successfully connecting, open "Command Prompt" and give a ping command to check if the connection is fine (If the ping command fails, refer the next section 19.2.3). After a successful ping, it must run "Disconnect", then continue the next important steps.



## 19.2.3 Important Configuration (DO NOT ignore it)!!! Users must do the following two important settings!

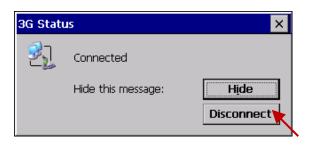
 If the PAC is going to send/receive the TCP data by using 3G/2G Internet connections, it must clear the gateway settings of LAN1 and LAN2 or else it may not work properly. Then, remember to run the PAC's Utility "File > Save and reboot" once to save the settings.

(Using t	he XPAC)			(Using the	WinPAC)		
<b>S</b>	<u></u>		<b>2</b>	<b>S</b>	<u>8</u> 7		<b>2</b>
Make Nev Connectio		PCI- FETCE5B1	PCI- FETCE5B2	Make New Connection	3G	LAN1	LAN2
	PCINFETCE5B1	' Settings			ок 🗙		
ſ	IP Address Nar	ne Servers					
	An IP address o		O Obtain an	IP address via DHCP			
	automatically as computer. If y	our network	Specify an	1 IP address			
		sk your network	IP <u>A</u> ddress:	192.168.71.1	L8 Lea	ave "Gateway	" blank.
	administrator fo		S <u>u</u> bnet Mask:	255.255.0.0			
	provided.	·	Default <u>G</u> atev	vay:			

If your PAC can connect to the Internet by using LAN1 or LAN2, recommend not to use the 3G/2G (in such a case, it requires to set the gateway of LAN1 or LAN2). The reason is LAN1 / LAN2 speed is faster than the 3G/2G.

2. In the previous page, the connection has been established. For now, the following configuration is very important and can't be ignored or else it will cause some problem when you connect to the 3G/2G network using the Win-GRAF program.

If the status of 3G/2G connection is still "Connected", please click the "Disconnect" button first.

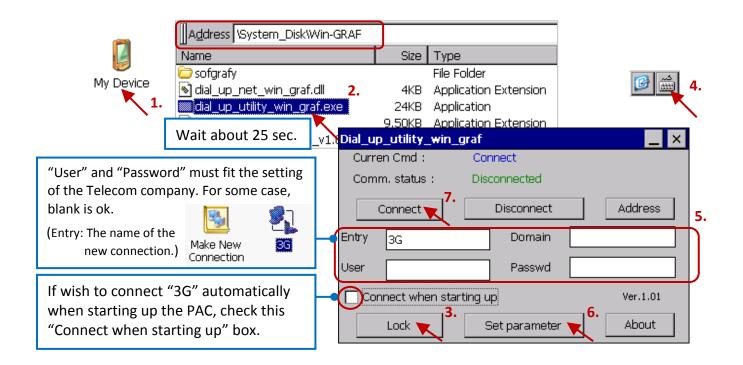


After that, run the new connection (here is "3G") and then click "Cancel" (At this time, Do Not click "Connect", **you must click "Cancel"**). Finally, run "File > Save and Reboot" in each PAC Utility (e.g., "<u>XPAC</u> Utility") to save all the settings (including this and those in the previous section) and then the PAC will restart automatically once.

	Make New Connection	PCI- PCI- FETCE5B1 FETCE5B2
Dial-Up Connec	tion	×
2 <u>7</u> 3G		
User Name: Password: Domain:		Phone: *99***1# Dial from: Home <u>Dial Properties</u>
	Save password	<u>C</u> onnect Cancel
		This "Cancel" operation must be set once. Then, run PAC's Utility to save this "Cancel" setting.
X	AC_Utility AC_Utility Save Rebo	and Reboot

### 19.2.4 Enable "Dial\_up\_utility\_win\_graf" " Dial-up Software

"Dial\_up\_utility\_win\_graf" is a software tool developed by ICP DAS for the 3G/2G dial-up automatically. It allows a Win-GRAF program (or VB.net, C#.net and C program) to connect or disconnect 3G/2G by sending commands and it can also read the connection status or command status. Please follow the steps below to enable this dial-up software. Then, click "Connect" to check if the connection is good and click "Disconnect" to check if the connection is broken. Finally, you need to run "...PAC Utility" and add the "dial\_ up\_utility\_win\_graf.exe" to the list of "Auto-Execution" and then run "File > Save and Reboot" to save the settings.



File Help		
Save Save and Reboot	onfig	Network Device Information Auto Execution Rotary Exe
Reboot <b>12.</b> Restore Utility Default Setti E <u>xi</u> t	ings m 1: m 2:	\System_Disk\Win-GRAF\Win_GRAF_XP_8xx Browse
	Program 3:	Browse
	Program 4:	Browse
ricinose to programo	Program 5:	Browse
can be specified to execute automatically	Program 6:	Browse
at system startup.	Program 7:	Browse
F	Program 8:	Browse
F	Program 9:	Browse
F	Program10:	Browse

## **19.3 Function Descriptions for Controlling 3G/2G Connection**

This Win-GRAF demo program shows how to use **\_3G\_connect()** to connect 3G/2G. Set up "To\_connect\_3G" as TRUE, it will instruct "Dial\_up\_utility\_win\_graf" to connect 3G/2G. (\* Set "To\_connect\_3G" as TRUE to connect 3G to access to the Internet \*) if To\_connect\_3G then To\_connect\_3G := FALSE ; \_3G\_connect() ; end\_if;

And, this Win-GRAF demo program shows
the way to use <b>_3G_disconnect()</b> to stop
the 3G/2G. Set up "To_disconnect_3G" as
TRUE, it will command
"Dial_up_utility_win_graf" to disconnect.

(\* Set "To\_disconnect\_3G" as TRUE to disconnect 3G \*)
if To\_disconnect\_3G then
 To\_disconnect\_3G := FALSE ;
 \_3G\_disconnect() ;
end\_if;

The program below shows the way to use **\_3G\_state()** to read the current status of the 3G/2G connection and use **\_3G\_read\_cmd()** to read the current status of the 3G/2G command.

(\* Get 3G connecting state \*)
State\_3G := \_3G\_state();

(* Read current 3G command
0 : No action
1 : Connect
2 : Disconnect *)
current_3G_cmd := _3G_read_cmd() ;

0 : No-action	
(1 ~ 6) : Connecting	
1 : Open COM Port	2 : Port opened
3 : Connect device	4 : Device connected
5 : Authenticate	6 : Authenticated
7 : Password expired	8 : Connected
9 : Disconnected	10 : Others

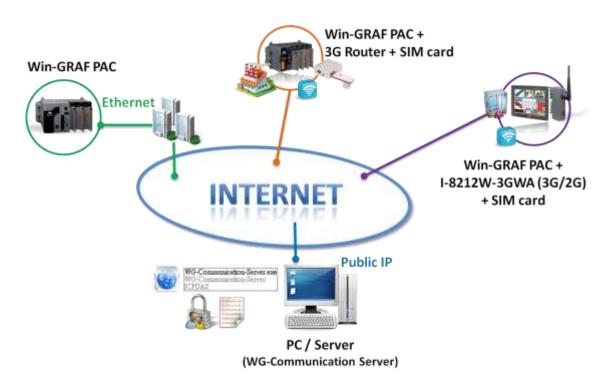
When the 3G/2G dial-up state is "Connected". The driver will try to ping DNS Server and "8.8.8.8" every 15 minutes to test if the 3G/2G communication is ok. If there is no response for 15 seconds, the Win-GRAF PAC will automatically reset the 3G/2G module and then re-dial-up to recover the 3G/2G communication . If the user doesn't want to ping this "8.8.8.8", can modify it to ping one another IP address (for example, 202.43.192.106). Like as below code.

```
(* Set IP address (other than 8.8.8.8) to ping every 15 minutes *)
if To_ping_a_new_ip then
   To_ping_a_new_ip := FALSE;
   TMP_BOOL := _3G_option( 1 , '202.43.192.106' );
end_if;
```

# Chapter 20 Sending a PAC File to a Remote PC via Ethernet or 3G/2G Wireless Networks

For some applications that need to record some useful data on the PAC like temperature, humidity, speed, voltage, current, etc., these data can be saved as a file by the user-designed Win-GRAF program. Then, the user can use the following ways to send this PAC file to a remote PC/Server.

In the shipping CD (CD-ROM: \Napdos\Win-GRAF\**demo-project**\), the user can find out the Win-GRAF demo project (demo\_send\_file.zip) which will be used in this chapter (refer <u>Chapter 12</u> to restore it). In addition, in the CD-ROM: \Napdos\Win-GRAF\**Tools\_Utility**\, there is a Win-GRAF Communications Server software (called WG-Communication-Server) which run on a PC/Server to allow the file sending from the PAC.



- **Note:** There are two ways to setup a Win-GRAF PAC with 3G/2G network.
  - (1) Using a 3G Router plus a SIM card.
  - (2) Using the I-8212W-3GWA module plus a SIM card (refer Chapter 19).

In addition, the PC/Server must run the "WG-Communication Server" software which must configure a public IP for the user sends the PAC file by using 3G/2G network or Ethernet (WAN).

The following Win-GRAF driver versions (or later version) support functions described in this chapter.

WP-8xx8 :	1.05
VP-x2x8-CE7 :	1.01
XP-8xx8-CE6 :	1.03
WP-5xx8-CE7 :	1.02

# 20.1 Description of the "WG-Communication-Server" Software

Note: Copy the "WG-Communication-Server" folder (includes WG-Communication-Server.exe and DLL) into the D:\ on your PC/Server, and then running this software to allow the file sending from a remote PAC. The user can use the "WG-Communication-Server" to create the username/ password (max. 100) for a remote PAC can log in and send a file to this PC/Server.

## Add a User Account:

1. After running the "WG-Communication-Server", it will zoom out to the bottom-right corner of your desktop screen (running in the background). Double-click the small icon if you want to configure it.





2. Click "Manage Account" to open Account Manager and then check "TCP port for file transmission".

WG-Commonnication-Se	
	Account Manager
Manage Account WO-Communication-S Copyright Mar. 2015, All Right Reserved.	VICP port for file transmittion 4567 The default is "4567". History Log File for File Transmittion : generate one file per day O generate one file per month
service@icpdas.com	User ListAccount Setting
	Name
	Password
	Root Dir for File Transmition
	Add Remove Generate User Key Apply
	OK

"TCP port for file transmission": To enable the specific TCP port (Default: 4567; Range: 1000 ~ 9999) for communicating with the PAC while file sending. (<u>Note:</u> Whether check or uncheck this item, or even modify the TCP Port number, the user must restart this software to apply the setting.)

## "History Log File":

To generate one historical log file per day/month (choose "per day" in this case).

<u>Note:</u> Go to the next step to set up an account. In addition, the user can open the "D:\WG-Communication-Server\account.txt" to check the account that you set up before.



Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 20-2

3. Click the "Add" button to enter a username (e.g., "Spider-Man") and then click "OK".

Account Manager
<ul> <li>TCP port for file transmition</li> <li>History Log File for File Transmition :</li> <li>generate one file per day</li> <li>generate one file per month</li> </ul>
User List Account Setting Name
Add user
Root Dir for File Transmition
Add Remove Generate User Key Apply

Click on the username (e.g., "Spider-Man") to set its password (e.g., "AB-567"). Click "Root Dir" can set the PAC file storage path, we recommend to use the defaults - D:\WG-Communication-Server \User Name (e.g., Spider-Man), and then click "Apply" to take effect. Finally, click "OK".
 <u>Note</u>: The user can view the "Generate User Key" usage in Section 22.1.

Account Manager		×	
<ul> <li>TCP port for file transmitie</li> <li>History Log File for File Transm</li> <li>generate one file per day</li> </ul>	nition :		
User List	Account Setting Name Spider-Man 2. Password ******		67" in this example. " <b>Apply" for these</b> I <mark>ke effect.</mark>
	Root Dir for File Transmition		
Add Remove	Generate User Key	Apply 4	•

## Delete a User Account

Click on the username you want to delete (e.g., "Spider-Man"), click the "Remove" button, and click "OK" to delete this account (username/password).

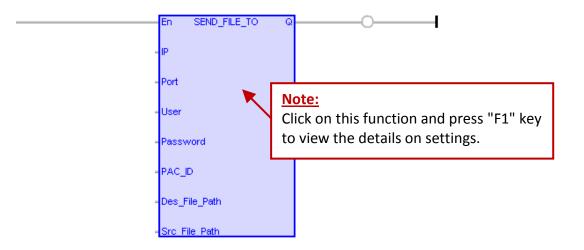


## 20.2 "Send\_File\_To", "Send\_File\_State" and "Send\_File\_Abort" Functions

There are 3 functions which can handle the file sending from the PAC to the PC.

## The "Send\_File\_To" function:

To send one PAC file to a remote PC/Server which is running the "WG-Communication-Server".



#### Input parameters:

## **Note:** Valid characters for the "User", "Password", "Des\_File\_Path" and "Src\_File\_Path":

A ~ Z, a ~ z (case-sensitive), 0 ~ 9, '.' (dot), '@' (At), '-' (minus) and '\_' (underscore).

IP:	(Data type: String) IP address of the remote PC (e.g., `192.168.71.29').
Port:	(Data type: DINT) The TCP port number (range: 1000 ~ 9999).
User:	(Data type: String) User name (max. 32 characters).
Password:	(Data type: String) Password (max. 32 characters).
PAC_ID:	(Data type: DINT) A number to identify the file is sent by which PAC.
	Value range: -2,147,483,648 ~ 0 ~ 2,147,483,647.
Des_File_Path	: (Data type: String) Destination file path in PC (max. 128 characters).
	And the first character should be '\', however the last character cannot be '\'.
	(E.g. '\2014\12\data001.txt' or '\Record\recp-2014-11-08.txt')
Src_File_Path:	(Data type: String) Source file path in PAC (max. 128 characters).
	And the first character should be '\', however the last character cannot be '\'.
	(E.g. '\Micro_SD\PAC\data001.txt' or '\System_Disk\DATA\recp-2014-11-08.txt')

## **Output parameters:**

Q:(Data type: BOOL)TRUE: Communication OK.FALSE: Wrong input parameters or "Src\_File\_path" doesn't exist or file size is 0.

## The "Send\_File\_State" Function:

To get the sending state of the PAC file. (MUST use with the "Send\_File\_to" function.)



## Input parameters:

None.

## Output parameters:

Q: (Data type: DINT)

- 0: Sleep, no "Send\_File\_To()" function is called.
- 1~99: Busy, a file is sending now. ("1~99" means the percentage of completion)
- 100 : Succeed, the file sending is finished and the file is sent successfully.
- -1 : Send failed or timeout.
- -2 : The file sending is interrupted by the "Send\_File\_Abort" command.
- -3 : Username/Password error.
- -4: Unable to create a sub-folder or file in the PC, or
   The file is over 10,000,000 Bytes, or
   There is no "WG-Communication-Server" running on the PC.
- -5 : The file for sending does not exist or the file size is "0".
- -6 : Unable to open the file located at "\Email\_ETH" of the PAC.

## The "Send\_File\_Abort" Function:

To abort the file sending.

En SEND FILE ABORT Q

## Input parameters:

None.

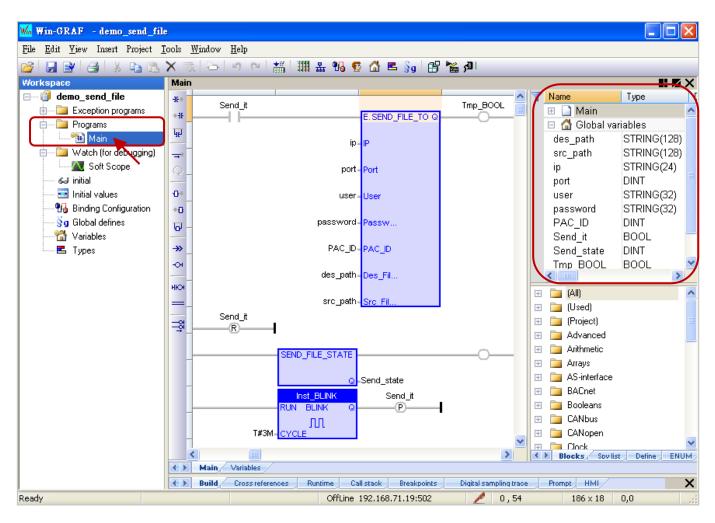
## Output parameters:

Q: (Data type: DINT), always return "TRUE".

# 20.3 Description of the Win-GRAF Demo Project (demo\_send\_file.zip)

This Win-GRAF demo project (demo\_send\_file.zip) can be found in the shipping CD (CD-ROM: \Napdos\Win-GRAF\demo-project\), refer <u>Chapter 12</u> to restore it.

After opening this "demo\_send\_file" project, mouse double-click on "Main" to open this LD program. Then, you can see all defined variables listed on the right of the window (i.e., Variable Area).



First of all, we need to know what variables are used in this project and their purposes. You can also click "Variables" on the left of the window (i.e., Workspace) to view/set all variables.

Workspace	Variables 🛛 🖬 🗖 🗙									
🖃 🖓 demo_send_file	Y	Name	Type 🗸 🗸		Dim.	Attrib.	Syb.	Init value	L	Jse
🗄 🚞 Exception programs		🗉 🚮 Global va	riables							^
🛓 🔤 Programs		des_path	STRING(128	)				\1001\Alarm01.txt'	П	
🔤 🎦 Main		src_path	STRING(128	)				\System_Disk\Alarm01.txt'		
🛓 📴 Watch (for debugging)		user	STRING(32)					'Spider-Man'		=
Soft Scope		password	STRING(32)					'AB-567'		-
🛶 initial		ip	STRING(24)					'192.168.78.161'		
📰 Initial values		port	DINT					4567		
📶 🗓 Binding Configuration		PAC_ID	DINT					1001		
So Global defines		Send_state	DINT							
C Variables		Send_it	BOOL							
		Tmp_BOOL	BOOL							~
		<							>	
	4	Main Variabl	es							_

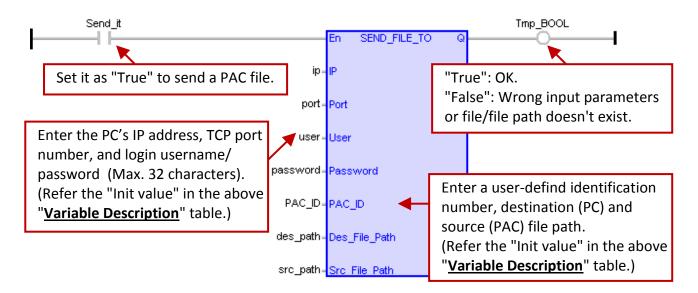
## Variable Description:

The following table describes the variable name, data type, their usages, and initial value that used in the Win-GRAF project (demo\_send\_file.zip).

Name	Data Type	Description
des_path	STRING(128)	Destination file path in PC (max. 128 characters). (Init value: <b>'\1001\Alarm01.txt'</b> )
src_path	STRING(128)	Source file path in PAC (max. 128 characters). (Init value: <b>'\System_Disk\Alarm01.txt'</b> )
user	ser STRING(32) PC login username (Max. 32 characters). (Init value: <b>'Spider-Man'</b> )	
password	STRING(32)	PC login password (Max. 32 characters). (Init value: <b>'AB-567'</b> )
ip	STRING(24)	IP address of the remote PC. (Max. 24 characters.) (Init value: <b>'192.168.78.161</b> ')
port DINT		The TCP port number of the WG-Communication-Server to receive files. (Init value: <b>1000</b> ; Range: 1000 ~ 9999).
PAC_ID DINT		A number to identify the file is sent by which PAC. (Init value: <b>1001</b> )
Send_state	DINT	Get the sending state of the PAC file. (See <u>Section 20.2</u> )
Send_it	BOOL	Set it as "True" to send a PAC file.
Tmp_BOOL	BOOL	<ul><li>TRUE: Communication OK.</li><li>FALSE: Wrong input parameters or "Src_File_path" doesn't exist or file size is 0.</li></ul>

## LD Program (Main)

The "Send\_File\_To" function (see <u>Section 20.2</u>) is used to send a PAC file to a remote PC/Server.

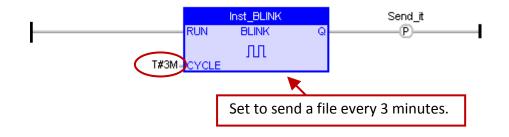




The "Send\_File\_State" function is used to get the sending state of the PAC file.

	En SEND_FILE_STATE Eno Q_Send_state
Return tl	he state of file sending.
0:	Sleep, no "Send_File_To( )" function is called.
1~99:	Busy, a file is sending now. ("1 ~ 99" means the percentage of completion)
100 :	Succeed, the file sending is finished and the file is sent successfully.
-1:	Send failed or timeout.
-2:	The file sending is interrupted by the "Send_File_Abort" command.
	(see <u>Section 20.2</u> )
-3:	Username/Password error.
-4:	Unable to create a sub-folder or file in the PC, or
	The file is over 10,000,000 Bytes, or
	There is no "WG-Communication-Server" running on the PC.
-5 :	The file for sending does not exist or the file size is "0".
-6 :	Unable to open the file located at "\Email_ETH" of the PAC.

The "BLINK" function block in this example is used to send a file every 3 minutes.



# 20.4 Test for File Sending

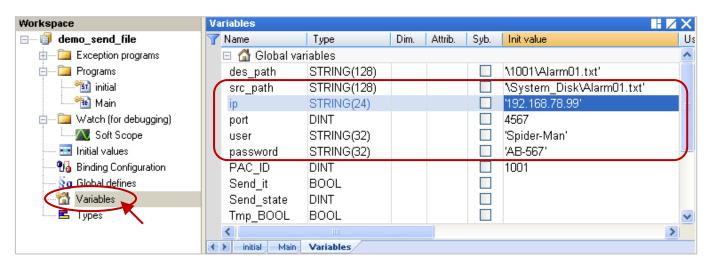
Before testing this project, check the following settings in the "Variables" window:

"src\_path": Does the file exist on the PAC (i.e., \System\_Disk\Alarm01.txt)? If not, download it by using FTP or assign other file. (<u>Note:</u> the file size cannot be 0 bytes.)

"ip": Enter the IP address of your PC. (If using 3G/2G network or an Ethernet (WAN) to connect to a Server, the user must enter a public IP.)

"port" (4567),"user" and "password":

These settings must be the same as the "WG-Communication-Server" settings (See Section 20.1).



Moreover, set the current IP address of the PAC in the Communication Settings, and then recompile and download this project to the Win-GRAF PAC.

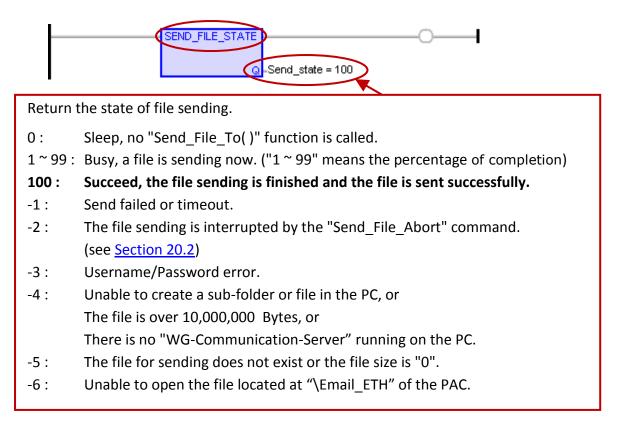
			Co	mmunication Settings	
Workspace		Variables			
🖃 🗐 demo_send_file	_	Vame Name			ОКЪ
🗄 🚞 Exception prog		Set as Startup Project		T5 Runtime 🗸	
🖻 🖳 📴 Programs		Communication Parameters	ſ		Cancel
🔤 initial	പ	On Line		192.168.71.29:502	
🔤 Main				192,168,71,29:502	Browse
				Current PAC's IP.	<u>H</u> elp
📉 📉 Soft Scop		Monitor			

Note: If the user wants to set the timeout value (default: 3 seconds), see <u>Section 2.3.5</u>. (E.g. Set the IP to "192.168.71.29:502(10)" which means the timeout is 10 seconds.)

After downloading the project, double-click "Send\_it" in the LD program (i.e., Main) to set it as "TRUE" to start sending the file.

ːː··································	u 🐒 🧔 👘
Imp_Bool = F       Send_it = FALSE       #HE   En SEND_FILE_TO Q	ALSE
Send_it 192.168.78.99'- IP	
TRUE [1] port = 4567 Port	
FALSE (①) = 'Spider-Man'- User	
vord = 'AB-567'- Password	
PAC_D = 1001 - PAC_D	
des_path = '\1001\Alarm0 Des_File	
src_path = '\System_Disk= <u>Src_File_P</u>	*
	>

When the progress of "Send\_state" reaches "100", it means that the file is sent successfully.



Now, the file is sent to the PC - "D:\WG-Communication-Server\Spider-Man\1001\Alarm01.txt". (Refer <u>Section 20.1</u> – Step4 and <u>Section 20.3</u> – the "des\_path" variable)

: 網址@ 🛅 D:\	WG-Commu	unication-Serve	r/Spider-Man/1001	*
名稱 🔺	大小	類型	修改日期	9
🖹 (Alarm01.txt	1 KB	文字文件	2015/1/5	上午 11:56

In addition, this Win-GRAF example project (demo\_send\_file.zip) is designed to send a file every 3 minutes, the user can open "WG-Communication-Server" to view file receiving records.

WG-Commonnication-Server	
<b>⊕ ≹</b>	
WG-Communication-Server V Copyright Mar. 2015, by ICP All Right Reserved. service@icpdas.com	
Spider-Man has login. Spider-Man has logout. Spider-Man recive file ok.	< ■
Spider-Man has login. Spider-Man has logout. Spider-Man recive file ok. Spider-Man has login.	All login records will be shown here.
Spider-Man has logout. Spider-Man recive file ok.	

# Chapter 21 Win-GRAF SMS Function

This chapter shows the way to send/receive a text message by using the Win-GRAF PAC comes with the 3G/2G wireless module. This Win-GRAF demo project (Demo\_SMS.zip) can be found in the shipping CD (CD-ROM: \Napdos\Win-GRAF\demo-project\), refer the Section 21.2 for details).

## Software/Hardware Requirements:

1. The Win-GRAF PAC

The following Win-GRAF driver versions (or later) support the SMS function (refer <u>Chapter 19</u>). WP-8xx8 : 1.05 ; VP-x2x8-CE7 : 1.01 ; XP-8xx8-CE6 : 1.03 ; WP-5xx8-CE7 : 1.02

2. The 3G/2G Wireless Module

Visit the website <u>http://m2m.icpdas.com/2G\_3G\_Modems.html</u> for more details. All supported GSM Modems: GTM-201-RS232, GTM-203M-3GWA, I-8212W, I-8212W-3GWA (see <u>Chapter 19</u>), I-8213W-3GWA.

<u>Note:</u> Due to the product certification issue, these modules can sale in certain areas. Please contact our agents for more information.

# 21.1 "GSM\_Open", "Send\_SMS" and "Read\_SMS" Functions

There are 3 functions which can handle the SMS messaging by the Win-GRAF PAC.

## The "GSM\_OPEN" function:

To open/close the GSM module.

		EN_GSM OPEN alsiniti		Note:
6-		gSignalQuality	-Signal	Click on this function and press "F1" key to view the
'0000'-	qPinCode	gErrorCode	-Error_Code	details on settings.

## Input parameters:

gEnable: Data type: BOOL

TRUE: Open the specified PAC's COM port to connect the GSM module and initialize it. FALSE: Disconnect the GSM module and close the specified PAC's COM port.

- gCOMPort: Data type: DINT The PAC's COM port number which connects with the GSM module.
- gPinCode: Data type: STRING Using this PIN code to unlock the SIM card, if it is necessary.

## Output parameters:

glsInitialized: Data type: BOOL TRUE: Open the specific PAC's COM port and initialize the GSM module successfully. FALSE: Failed to open the specific PAC's COM port or the GSM module is not initialized. gSignalQuality: Data type: SINT

- $0 \sim 31$ : The higher value means that the signal strength is stronger.
- 99: Unknow or not detectable.
- gErrorCode: Data type: INT
  - 0 : No error.
  - -1 : Broken line.
  - -2 : SIM card not inserted.
  - -3 : SIM PIN code wrong.
  - -4 : SIM configuration error.
  - -5: Cannot open the specified PAC's COM port.

## The "GSM\_SEND\_SMS" Function:

To send the SMS message via the GSM module.

	Inst_SEt gExCGSM_SE		
Phone_Nb-	gPhoneNb	gErrorCode	-Send_SMS_Error_code
Sms_text=	gSMSText		lote:
codePage -	qCodePage		lick on this function and press "F1" key o view the details on settings.

**Notice:** Before using this "GSM\_SEND\_SMS" function, first the user must use the "GSM\_OPEN" to open the PAC's COM Port which the GSM module connects with, or else this function will not work.

## Input parameters:

- gExecute: Data type: BOOL Pulse TRUE: trigger it to send SMS message.
- gPhoneNb: Data type: STRING Destination-Address (i.e., the phone number).
- gSMSText: Data type: STRING The text message.
- gCodePage: Data type: UDINT The code page of the text. English: 0 Traditional Chinese: 950 Simplified Chinese: 936 Japanese: 932 Russian: 866
  - Notice: If the "gCodePage" is set as "0", the max length of text is 160 characters.
     If the "gCodePage" is not set as "0", the max length of text is 70 characters.
     If the user type more than the maximum text length, the Win-GRAF driver will just send the maximum characters from the beginning of the text.

## **Output parameters:**

- gErrorCode: Data type: INT
  - 4 : Send SMS succeeds -1 : Cannot find valid GSM module.
    - -2 : SIM card not inserted.
  - 2 : Send SMS is pending -4 : SIM card configuration failed.
  - 1 : Prepare to send SMS -5 : Cannot open the PAC's COM port.
    - -6 : NO recipient number.
      - -7 : Send SMS message failed.

## The "GSM\_READ\_SMS" Function:

To read the SMS message via the GSM module.

3 : Sending SMS

0 : No operation.

	Inst_READ_SMS gEx_GSM_READ_SMS_sis		Note:
codePage1 -	gCodePage gSenderPho	-Sender_phoneNb	Click on this function and press "F1" key to view the details on
	qReceiveSM	-ReceiveSMSText	settings.

**Notice:** Before using this "GSM\_READ\_SMS" function, first the user must use the "GSM\_OPEN" to open the PAC's COM Port which the GSM module connects with, or else this function will not work.

#### Input parameters:

gExecute: Data type: BOOL

TRUE : Enable to read the SMS message from the GSM module. FALSE : Disable to read the SMS message from the GSM module.

gCodePage: Data type: UDINT

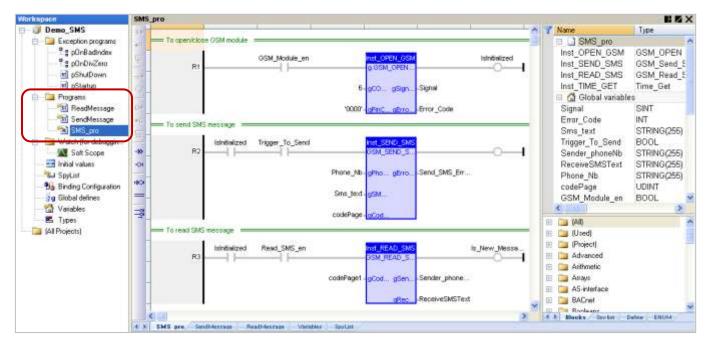
The code page of the text. Please refer the following example:English: 0Traditional Chinese: 950Simplified Chinese: 936Japanese: 932Russian: 866

## **Output parameters:**

- gIsNewSMS: Data type: BOOL Pulse TRUE: the new message is coming.
- gSenderPhoneNb: Data type: STRING Originating-Address (i.e., the phone number).
- gReceiveSMSText: Data type: STRING The text message.

# 21.2 Description of the Win-GRAF Demo Project (Demo\_SMS.zip)

This Win-GRAF demo project (Demo\_SMS.zip) can be found in the shipping CD (CD-ROM: \Napdos \Win-GRAF\demo-project\), refer <u>Chapter 12</u> to open project from a zip file. There are one LD program (SMS\_pro) and two ST programs (SendMessage and ReadMessage) in this "Demo\_SMS" project.

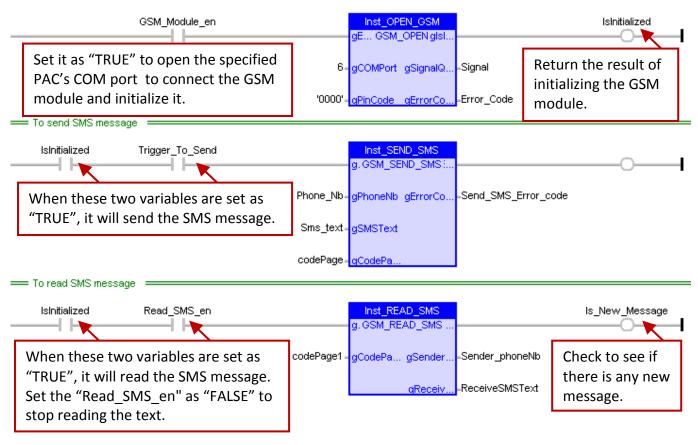


## Variable Description: (\*: Refer the Section 21.1 for more details)

Name	Data Type	Description
GSM_Module_en	BOOL	Set it as "TRUE" to open the specified PAC's COM port to connect the GSM module and initialize it.
Signal	SINT	The signal quality of the GSM module. (*)
Error_Code	INT	The error code of the GSM module. (*)
IsInitialized	BOOL	To check if the GSM module has been initialized.
Trigger_To_Send	BOOL	Set it as "TRUE" to send a text message.
Phone_Nb	STRING(255)	The phone number of recipient. (Init value: '0932860424')
Sms_text	STRING(255)	The content of texting. (Init value: 'This message is sent from Win-GRAF PAC')
codePage	UDINT	The code page of the sending text. (*)
Send_SMS_Error_code	INT	The error code during the SMS sending. (*)
Read_SMS_en	BOOL	Set it as "TRUE" to read a text message.
codePage1	UDINT	The code page of the receiving text. (*) (Init value: UDINT#950)
Sender_phoneNb	STRING(255)	The phone number of sender.
ReceiveSMSText	STRING(255)	The received text message.
Is_New_Message	BOOL	To check if there is any new text message.
Got_New_Message	STRING(255)	To receive text message. (ST – ReadMessage)
Got_Message_from_who	STRING(255)	To receive the phone number of sender. (ST – ReadMessage)

## The LD Program (SMS\_pro)

Refer the <u>Section 21.1</u> to view how to configure these three functions.



## The ST Program (SendMessage)

```
if Trigger_To_Send then
```

```
if IsInitialized then
```

if Send\_SMS\_Error\_code < 0 then

(\* Send SMS failed \*)

Trigger\_To\_Send := false;

## (\* TODO: Add failed processing here \*)

```
elsif Send_SMS_Error_code = 4 then
  (* Send SMS successed *)
  Trigger_To_Send := false;
```

## (\* TODO: Add success processing here \*)

```
end_if;
```

else

```
(* GSM module is not initialized *)
```

```
(* TODO: Add failed processing here *)
```

```
end_if;
end_if;
```

## The ST Program (ReadMessage)

(\* Get new message \*)

if Is\_New\_Message then

(\* If the user need more time to deal with the new message, \*)

(\* the user could set "Read\_SMS\_en" as "FALSE" to disable the Read\_SMS function \*)

(\* and then set it as "TRUE" after dealing \*)

(\* Notice : If the user set "Read\_SMS\_en" as "FALSE", it will stop reading the SMS message. \*)

```
(* Read_SMS_en := false; *)
```

Got\_New\_Message := ReceiveSMSText; Got\_Message\_from\_who := Sender\_phoneNb;

(\* do more operating \*)

end\_if;

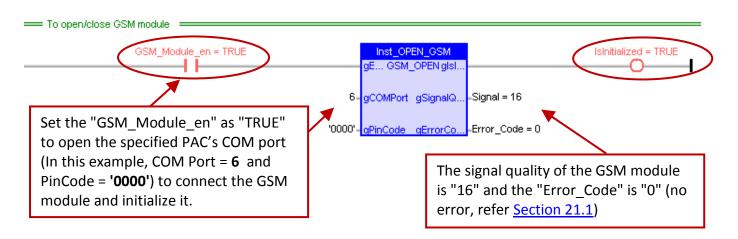
## 21.3 Test for SMS Messaging

In this example, we use one Win-GRAF XPAC and plug one 3G/2G module (I-8212W-3GWA + SIM card) in its slot1. Before testing this project, set the current IP address of the PAC in the Communication Settings, and then recompile and download this project to the Win-GRAF PAC. (Refer the Section 2.3.5 in case of any doubt.)

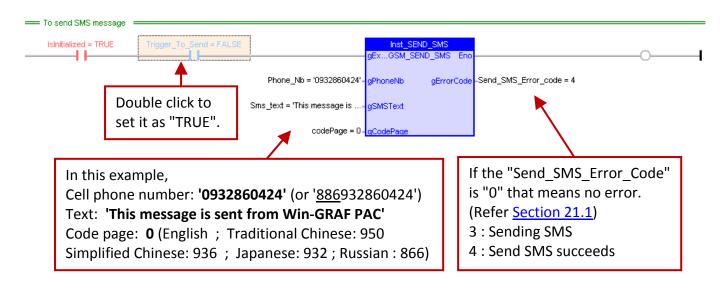
After connecting to the PAC, the "SMS\_pro" program is shown as below:

2   🛛 🕑   🕘   🔾 🕹 🗈 🛙	2 X 🖘 🗁 🖉 🗠 🛗 🎹 🏪 😘 😨 🖓 🖻 🚱 🚰 🖓 RUN 💿 ОООО 🤧 🛗 🚅 🗉 🐒 🏟
Workspace	
Demo_SMS [RUN]	
🖮 🚞 Exception programs	To open/close GSM module
🔓 pOnBadIndex	ाज – फिर GSM_Module_en = TRUE Inst_OPEN_GSM IsInitialized = TRUE
pOnDivZero	및     GSM_Module_en = TRUE     Inst_OPEN_GSM     Isinitialized = TRUE
🔤 🖬 pShutDown	
💷 🖬 pStartup	6-gCOMPort_gSignalQSignal = 13
📄 🛄 Programs	
📷 ReadMessage	0000' - <u>aPinCode</u> _ Error_CoError_Code = 0
📷 SendMessage	*B To send SMS message
SMS_pro	
📄 🖳 📴 Watch (for debugging)	Isinitialized = TRUE Trigger_To_Send = FA Inst_SEND_SMS
Soft Scope	g.GSM_SEND_SMS:
Initial values	-or
SpyList	Phone_Nb = '0932860 gPhoneNb gErrorCoSend_SMS_Error_cod
1 Binding Configuration	Sins_text = 'This mes gSMSText
👌 🖓 Global defines	
🖓 Variables	codePage = 0 - gCodePa
E Types	
(All Projects)	To read SMS message
	Isinitialized = TRUE Read_SMS en = TRUE Inst_READ_SMS Is New Message = F
	g.GSM_READ_SMS
	codePage1 = 950 - gCodePa gSenderpSender_phoneNb = "
	K SMS pro SendMessage ReadMessage Variables SpyList

 Fill in the COM Port No. which the GSM module used and the PIN code of your SIM card (if it is necessary to use). In this example, the GSM module uses the COM6 (refer <u>Section 19.1</u>) and the SIM card PIN Code is '0000'.



- Fill in the phone number, the text message and the code page. In this example, the PAC will send an English (CodePage = 0) text message ('This message is sent from Win-GRAF PAC') to the cell phone number '0932860424' (or '<u>886</u>932860424').
- Mouse double click the "Trigger\_To\_Send" to set it as "TRUE" to send a text message. If the "Send\_SMS\_Error\_Code" changes to "4" that means the texting is successful (refer <u>Section 21.1</u>).

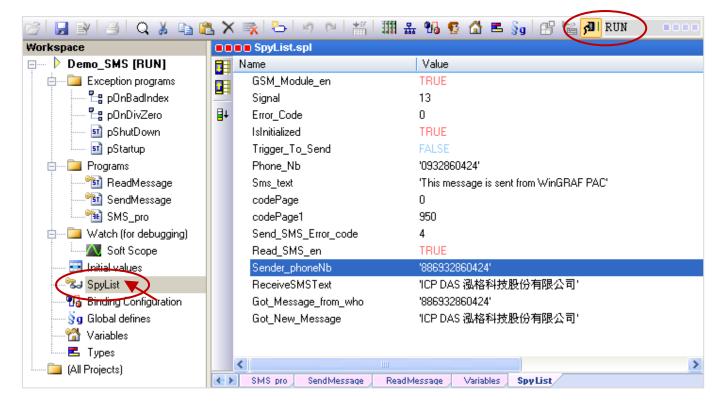


4. For testing an SMS message reading via the PAC, using a cell phone to send a text message to this SIM card. In this example, using the cell phone number (**'886932860424'**) to reply a Chinese

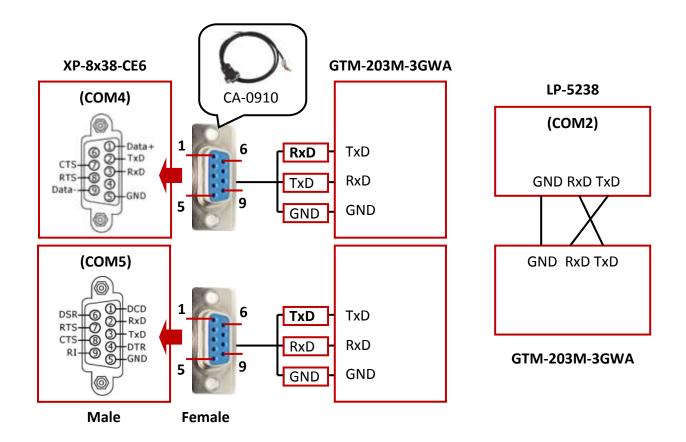
(CodePage1 = 950) text ('ICP DAS 泓格科技股份有限公司') to this SIM card.



Moreover, the user can double click the "SpyList" to open the Spy list, and then change the variable value to test the SMS messaging.



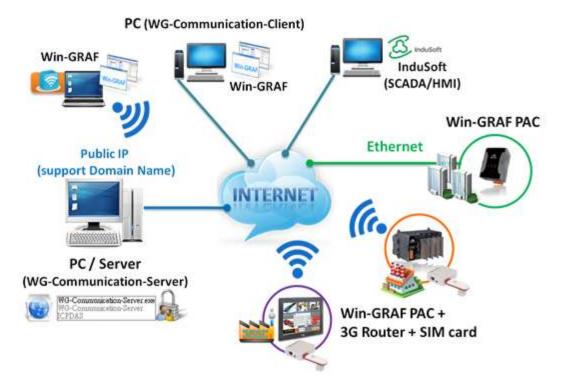
**Note:** If the 3G/2G wireless module is connected to the RS-232 Port of the Win-GRAF PAC, the user must refer the <u>Appendix F</u> to view the pin assignment of the COM Port. (Using the XP-8xx8-CE6 as an example, the pin assignment of the COM4 or COM5 is different.)



# Chapter 22 The Intelligent Win-GRAF 3G Solution

The following Win-GRAF driver versions support intelligent 3G solution described in this chapter.

WP-8xx8:1.05; VP-x2x8-CE7:1.01; XP-8xx8-CE6:1.03; WP-5238-CE7:1.02



The main features of the intelligent 3G solution is to connect the Win-GRAF PAC to the Server (WG-Communication-Server) with a public IP (support Domain Name) via 3G wireless or via the Internet. Then the PC/laptop (WG-Communication-Client) can communicate with the remote PAC via the Server. The user can achieve the following functions:

- 1. The user's PC/laptop can run the Win-GRAF Workbench to remotely debug/update the Win-GRAF application on the PAC.
- 2. The user's PC/laptop can run the SCADA/HMI software (e.g., InduSoft) to remotely monitor the PAC.
- 3. The user's PC/laptop can remotely update the Win-GRAF driver on the PAC if it is necessary.
- 4. The remote PAC can actively send a log file to the WG-Communication-Server.

## Note:

- 1. The WG-Communication-Server must have a public IP address. Other remote PAC and user's PC/laptop no need a public IP address.
- 2. If Internet connection for the PAC is available, the user no need to buy a 3G Router.
- 3. If Internet connection for the PAC is unavailable, the user can buy a 3G Router and then insert a local SIM card to enable the PAC to access the Internet. We recommend you to choose an unlimited 3G data plan to reduce the Internet connection charges.
- 4. If there are several PACs in the same worksite to connect to the WG-Communication-Server, set up an Ethernet switch first. Then connect this switch to the 3G Router so that these PACs can access the Internet through this switch.

# 22.1 Set Up the PC/WG-Communication-Server

The user can use the "WG-Communication-Server" to create the user account (max. 100) for the remote PC/laptop or PAC to log in to the Server (<u>Note:</u> Both the PC/laptop and the PAC must log in with the same username/password, and the PC/laptop need to install the "<u>WG-Communication-Client</u>".) In the Win-GRAF PAC CD (CD-ROM: \Napdos\Win-GRAF\Tools\_Utility), copy the "WG-Communication-Server" folder to your Server PC's "**D**:" (i.e., **D:\WG-Communication-Server**).

**Notice:** The "WG-Communication-Server.exe" must be stored in this folder to work properly.

## Add a User Account:

1. After running the "WG-Communication-Server", it will zoom out to the bottom-right corner of your desktop screen (running in the background). Double-click the small icon if you want to configure it.





2. Click the "Manage Account" icon to open the setting window.

WG-Commonnication-Se	rver 🔲 🗖 🗙
	Account Manager
Manage Account WO-Communication-S Copyright Mar. 2015, All Right Reserved. service @icpdas.com	<ul> <li>ICP port for file transmittion 4567</li> <li>History Log File for File Transmittion :</li> <li>⊙ generate one file per day ○ generate one file per month</li> </ul>
service eleptias.com	User List Account Setting
	Name
	Password
	Root Dir for File Transmition
	Add Remove Generate User Key Apply
	OK

"TCP port for file transmission": To enable the specific TCP port (Default: 4567; Range: 1000 ~ 9999) for communicating with the PAC while file sending. (<u>Note:</u> Whether check or uncheck this item, or even modify the TCP Port number, the user must restart this software to apply the setting.)

"History Log File": To generate one historical log file per day/month (defaults: "per day").

<u>Note:</u> Just go to the next step to set up an account. In addition, the user can open the "D:\WG-Communication-Server\account.txt" to check the account that you set up before.



Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 22-2

3. Click the "Add" button to enter a username (e.g., "Spider-Man") and then click "OK".

Account Manager
TCP port for file transmittion 4567
History Log File for File Transmition :
User List Account Setting
Name
Add user
Spider-Man OK K
Root Dir for File Transmition        Add Remove     Generate User Key     Apply
ОК

4. Click on the username (e.g., "Spider-Man") to set its password (e.g., "AB-567"). Click "Root Dir" can set the storage path of the PAC file or user key, we recommend you to use the defaults - D:\ WG-Communication-Server\User Name (e.g., Spider-Man), and then click "Apply" to take effect (and unlock the "Generate User Key" button.

Account Manager		
TCP port for file transm	ition 4567	
History Log File for File Tra generate one file per day	nsmition : O generate one file per month	
User List	Account Setting Name Spider-Man 2. Password ******	Enter "AB-567" in this example. <u>Note:</u> Click "Apply" for these settins to take effect.
	Root Dir for File Transmition	3. ASpider-Man
Add Remove	Generate User Key 5.	Apply 4.

5. For security concerns, the remote PC/laptop must have a user key provided by the Server for successfully logging in to it. Click "Generate User Key" to create a user key.
<u>Notice:</u> Each time you click "Generate User Key" and confirm it, a new user key will be created.
Be careful to make sure you want to do it because the PC/laptop which includes an old user key cannot login to the Server any more (unless installing the generated new user-key in it).

If it must to create a new user key, click "Yes", and then the old user key on remote PC will become invaild. Send the user key to the remote user (client), and copy it to the **D:\WG-Communication-Client** on user's PC. Then, the user will have the access authority to use this username/password to login to the Server.



## Delete a User Account

Click on the username you want to delete (e.g., "Spider-Man"), click the "Remove" button, and click "OK" to delete this username/password.

Account Manager	
TCP port for file transmitted	on 4567
History Log File for File Transm generate one file per day (	
User List	Account Setting
Spider-Man	Name Spider-Man
	Password *****
	Root Dir for File Transmition
	D:\WG-Communication-Server'Spider-Man
Add Remove	Generate User Key Apply
	OK 💌

## View the connection status

This feature is used to view what PC/laptop or Win-GRAF PAC is connected to this Server.

	WG-Com	nonnication-Server	:		
	Copyrigh All Right	v Connections h -Serv ht Mar. 2015, by t Reserved. Picpdas.com		LTD.	
Connectio	on state				
PAC Po	rt ID Client Conn	ections			
	Port ID	Status	A PAC, its	s Port ID is 20009 (See Section	
•	20009	Connected 🔶 🔶	22.2), is o	connected to this Server.	
Connectio	on state			×	
DAC D-	rt ID Client Conn	actions			
PAC PO:				The user use the account - "Spider	·-Man"
	Login Name	IP	Status	to log in to the Server (122.147.13	
	Spider-Man	122.147.137.140	Connected	, ,	,

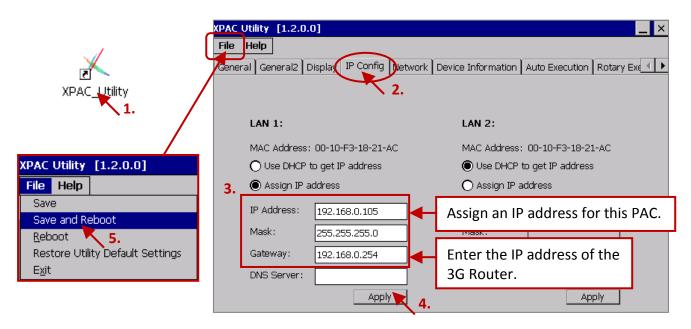
# 22.2 Set Up a Remote PAC to Connect to the WG-Communication-Server

If the Win-GRAF PAC connects to the Internet via a 3G Router (plus a SIM card), see its product instructions to configure it. <u>Note:</u> Many 3G routers on the market has the build-in Wi-Fi feature. For security issues on PAC communication data, we recommend you to disable the Wi-Fi function of the 3G Router.

## The network setting on the PAC

Please configure the network settings according to the field demands. If the user uses the 3G Router as a gateway and set its IP address as "192.168.0.254", and then assign IP addresses which range from "192.168.0.100" to "192.168.0.200" for PACs, then you can set the PAC's network setting as below.

 Double click the PAC Utility (e.g., XPAC\_Utility) on the PAC's desktop, and then click the IP setting tab (e.g., IP Config) to fill in the proper IP, Mask, and Gateway address, and click "Apply". Finally, execute "File → Save and Reboot" to save and reboot the PAC.



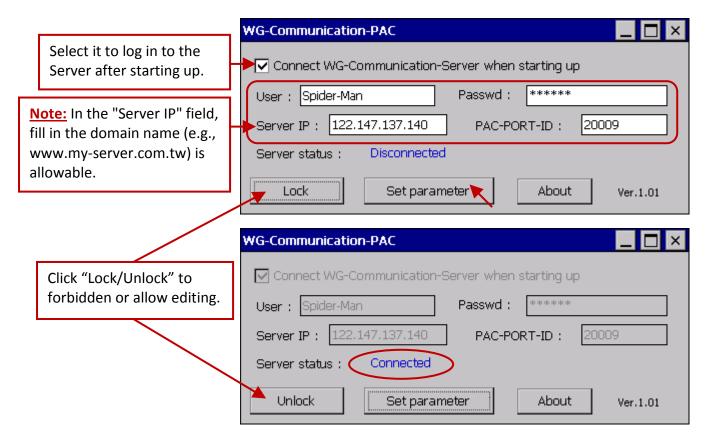
## Using the "WG-Communication-PAC" on the PAC to connect to the Server

On the PAC, you can find out "WG-Communication-PAC.exe" at the path "\System\_Disk\Win-GRAF\", and then use it to connect to the WG-Communication-Server.

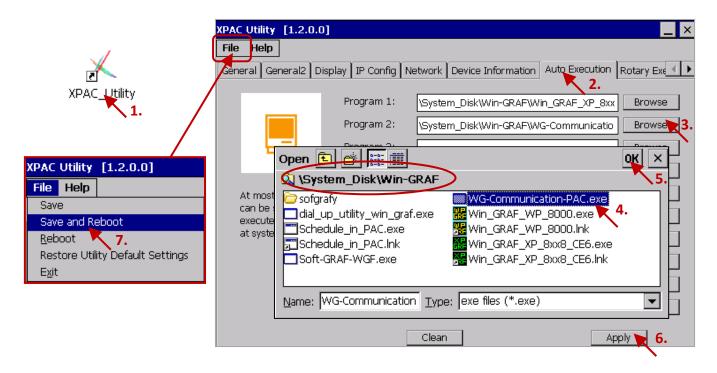
1. Double click the "WG-Communication-PAC.exe" to open the setting window.



- Fill in the username/password (e.g., Spider-Man/AB-567, refer Section 22.1) that created by the WG-Communication-Server, and enter the IP address of the Server, and then assigns a Port ID (e.g., 20009; Range: 20000 ~ 22000) for this PAC. (<u>Note:</u> The "PAC-Port-ID" is used for a remote Server or a user's PC to identify which PAC is connected with them. Different PACs connect to the same Server should use different PAC-PORT-ID.)
- 3. Finally, click the "Set parameter" button to take effect.



Moreover, run the PAC Utility (e.g., XPAC\_Utility) and then follow the steps (shown as below) to add "WG-Communication-PAC.exe" into the startup program list, and then execute "File  $\rightarrow$  Save and Reboot" to save and reboot the PAC.

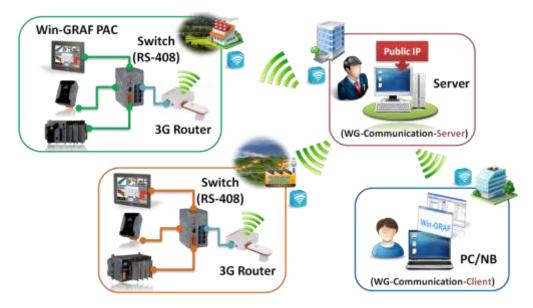


Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 22-6

# 22.3 Set Up the WG-Communication-Client on a User PC or the SCADA PC

The user's PC can connect to the Server and then debug/download the Win-GRAF project or update the Win-GRAF driver to the remote PAC by using the "WG-Communication-Client" software. In the Win-GRAF PAC CD (CD-ROM: \Napdos\Win-GRAF\Tools\_Utility), copy the "WG-Communication-Client" folder to your PC's "**D**:" (i.e., **D**:\WG-Communication-Client).

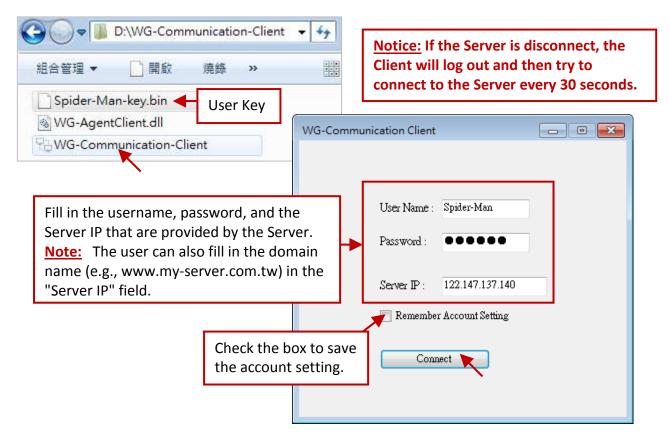
Notice: The "WG-Communication-Client.exe" must be stored in this folder to work properly.



## 22.3.1 Logging to the Server and detect the PAC connection status

Before logging to the Server, make sure the "WG-Communication-Client" folder contains the user key provided by the Server.

1. Mouse double click the "WG-Communication-Client" to open the setting window, and fill in the user name, password, and Server IP, and then click the "Connect" button.



 After connecting to the Server, you can see the "Server Status" shows "Server Connected" and the following table will list the status of the PAC connection that connect with the Server (refer Section 22.2). The user can click "Add PAC-Port-ID to detect" to add a PAC-PORT-ID (e.g., 20001) to the PAC connection-detection lists. Or, click "Remove PAC-Port-ID to detect" to remove it from the lists.

- WG-	Communication C	lient		
Connecti	ions	Add ne	w PAC-PORT-ID to	detect.
	ld PAC-Port-ID to det	20001	-	detect. (2000022000) OK Cancel
Server	Status : Se	rver Connected		
	PAC Name	PAC-PORT-ID	Status	Description
•	NewPAC	20009	Connected	Add your description here
	NewPAC	20010	Connected	Add your description here

3. Mouse double click on any PAC name to open that setting tab. The user can click "Rename" to change the displayed PAC name, and also can click "Modify" to add comments, and then click "Apply" to take effect.

Serve	er Status : S	erver Connected		
	PAC Name	PAC-PORT-ID	Status	Description
•	NewPAC 💌	20009	Connected	Add your description here
	NewPAC	20010	Connected	Add your description here
	NewPAC	20001	Disconnected	Add your description here

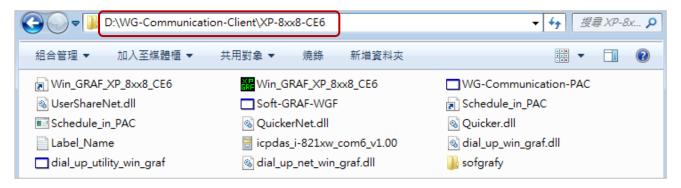
WG-Communication Client
Connections NewPAC X
PAC Name : NewPAC Rename the PAC name to :
PAC-Port-ID: 20009
РАС Туре ХР-8их8-СЕ6
Win-GRAF driver version 1.03
Description : Modify Description : Apply Add your description here The XPAC at Banquiao, New Taipei City
Update Win-GRAF Driver

## Remotely update the Win-GRAF PAC Driver from the PC

In the normal case, the user doesn't need to update the Win-GRAF PAC driver unless you want newer driver version that contains new released functions or bug fixing. The user can visit the website to download the latest version of the Win-GRAF PAC driver.

www.icpdas.com/root/product/solutions/softplc\_based\_on\_pac/win-graf/download/win-graf-driver.html

Before updating the Win-GRAF driver, make sure you have copied all related files into the proper folder. Using the "**XP-8xx8-CE6**" as an example, first unzip the driver file (.zip) and then copy all files from the \xp-8xx8-ce6-driver-1.xx\1.xx (e.g., 1.03) to the D:\WG-Communication-Client\**XP-8xx8-CE6**.

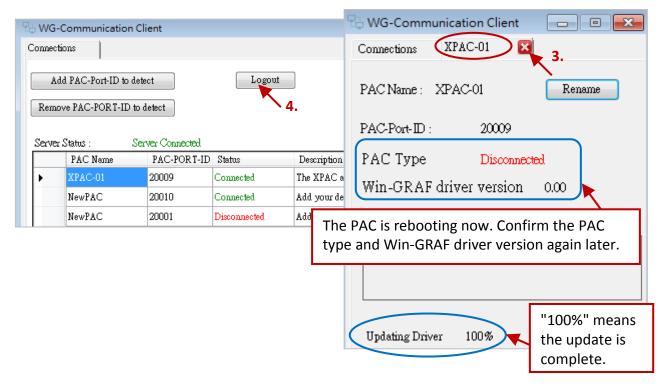


- 1. Run the D:\WG-Communication-Client\WG-Communication-Client.exe and log in to the Server. (If you are not familiar with the way, refer<u>Section 22.3.1</u>.)
- 2. Double click on the PAC name that you want to update and has been connected to the Server, and then click the "Update Win-GRAF Driver" button.

# <u>Note:</u> Don't restart this PAC, stop this Client, or close this tab while updating the driver. And, update only one PAC at each time.

-	🗄 WG-C	Communication	Client			
	Connectio	ons				
	Add	l PAC-Port-ID to d	letect	Logout		
	Remov	e PAC-PORT-ID	to detect			😳 WG-Communication Client 📃 💷 💌
	Server S	Status :	Server Connected			Connections XPAC-01
ſ	•	PAC Name XPAC-01	PAC-PORT-ID 20009	Status Connected	D Th	PAC Name : XPAC-01 Rename
		NewPAC NewPAC	1. 20010 20001	Connected Disconnected	Ad Ad	PAC-Port-ID : 20009
		<u> </u>		·		PAC Type XP-8xx8-CE6 Win-GRAF driver version 1.03 Description : Modify The XPAC at Banquiao, New Taipei City
						Update Win-GRAF Driver

- 3. When the "Updating Driver" shows "100%" means that the update is complete. Waiting for about 60 to 90 seconds. If the "PAC Type" shows "Disconnected" which means the PAC is rebooting now. After that, confirm the PAC type and Win-GRAF driver version again.
- 4. Close the "XPAC-01" tab. Then, you can change the other PAC's settings or click "Logout" to log out the Server.



## 22.3.2 Remotely update the Win-GRAF project from the PC/Win-GRAF Workbench

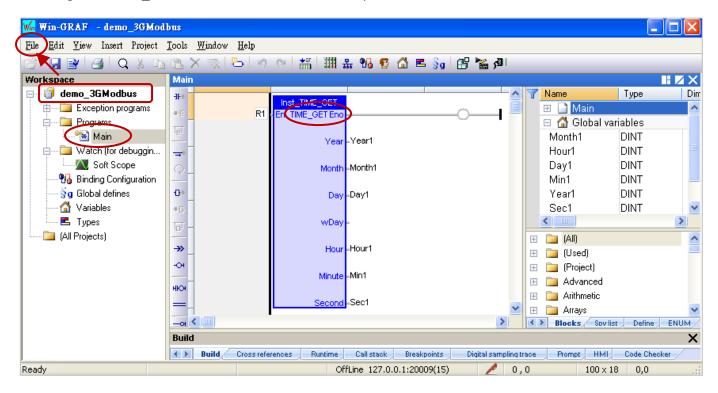
This section will describe how to use the WG-Communication-Client on the PC/Win-GRAF Workbench to connect to the Server and then remotely download/update the Win-GRAF project to the PAC.



 Run the D:\WG-Communication-Client\WG-Communication-Client.exe and log in to the Server. (If you are not familiar with the way, refer<u>Section 22.3.1</u>.)

組合管理 👻 📄 開啟	燒錄 »	-
🗋 Spider-Man-key.bin ◄	User Key	
WG-AgentClient.dll	User Key	

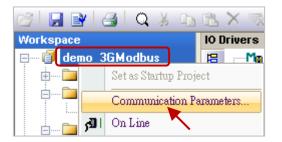
 Run the Win-GRAF Workbench and then click "File → Add Existing Project → From Zip..." to open the Win-GRAF project (demo\_3GModbus.zip) which is stored in the Win-GRAF PAC CD and its file path is CD-ROM: \Napdos\Win-GRAF\demo-project\. This example project includes an LD program (Main), using the "Time\_Get" function block to read the system time.



And, this project enables a Modbus TCP Slave to allow the SCADA/HMI software (e.g., "<u>InduSoft</u>") or the Modbus Master to access Win-GRAF variables (refer <u>Section 3.1</u> for more details).

		-	-				
E.	X 🔍 🗁 🔊 🛯 🛗 🗰 🖓 😨 🟠	2	5	§g 🖪 🖀	ا 🛚 کې		
10 [	)rivers						HZX
E	Ma MODBUS Master		7	Name	$\nabla$	Туре	
몲	🔁 MODBUS Slave			🗉 🚮 Glo	bal var	iables	•
**	🖮 🔐 Server - Slave number = 1			Year1		DINT	
	≟*∎ Input Registers [12000]			Sec1		DINT	
-	💶 +01: Year1			Month1		DINT	
	💶 +23: Month1			Min1		DINT	
	💶 +45: Day1			Hour1		DINT	
	+67: Hour1			Day1		DINT	k
¢¦>	<b>=</b> +89: Min1						
	💶 +1011: Sec1		N	lame	Ιv	alue	
∎+		4		ame	V.	aiue	
	Slave number Server ID						
+	1						

2. Mouse right click on the project name (demo\_3GModbus), and then click "Communication Parameters.." to enter the IP address and the port number.



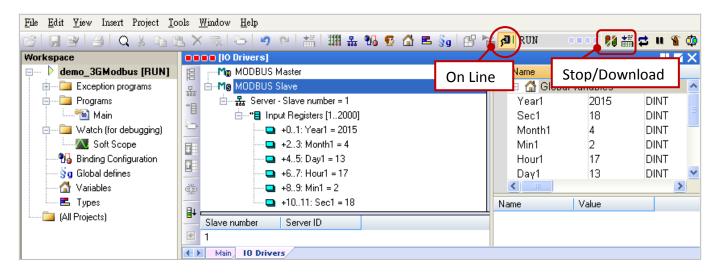
The configure way is "**127.0.0.1:PAC-PORT-ID(Timeout)**", and it's set to "127.0.0.1:20009(15)" in this example.

Note: "PAC-PORT-ID" is a uniquely identified ID, and make sure this PAC is connected to the Server (refer <u>Section 22.2</u>). By default, the Timeout is 3 seconds, recommend you to set it between 15 and 30 seconds if the PAC access to the Internet via the 3G router (plus a SIM card).

Due to the 3G is a wireless network and transmit signals through a base station, it can cause communication delays or exceptions sometimes. Recommend you to set a bigger timeout value to range from 15 to 30 seconds to reduce the communication timeout or read/write error.

Communication Settings	
T5 Runtime         127.0.0.1:20009(15)         127.0.0.1:20010(15)         127.0.0.1:20010(15)         192.168.71.10:502         192.168.71.17:502	OK Cancel Browse <u>H</u> elp

- 3. Click the "On Line" button to make a connection and download the project to the PAC.
  - Note: If you ever edit this program, it must be compiled before downloading (refer <u>Section 2.3.4</u>); If there is the other project name is running, just click "Stop" to stop that project and then click "Download" to download the current project (refer <u>Section 2.3.5</u> - Steps 4 ~ 6).

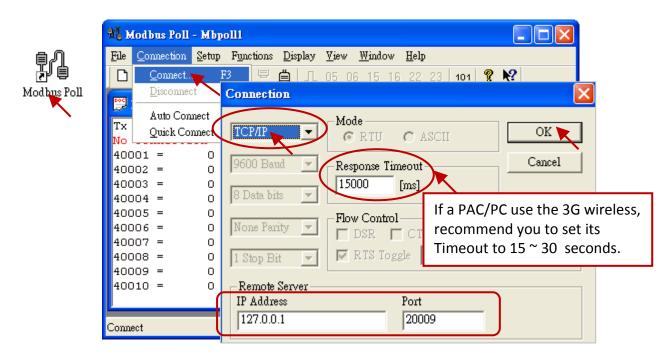


## 22.3.3 To read/write the PAC data remotely from the PC/SCADA

For the different kinds of HMI/SCADA software intend to communicate with a remote PAC via Modbus TCP protocol through the WG-Communication-Server, the user must run the WG-Communication-Client on an HMI/SCADA PC first. Then use it to log in to the Server, and then enter the PAC's IP address (fixed 127.0.0.1) and PAC-PORT-ID (20000 ~ 22000) that you want to connect to on the SCADA software.

In this section, we will introduce you how to use the Modbus Poll to act as an HMI/ SCADA/Modbus Master on the PC to read the PAC data remotely. You can visit the website to download and install the Modbus Poll: <u>http://www.modbustools.com/modbus\_poll.html</u>

- First, run the "D:\WG-Communication-Client\WG-Communication-Client.exe", and then log in to the Server. (If you are not familiar with the setting, refer <u>Section 22.3.1</u>)
- Double click the "Modbus Poll" icon to open it, and then click "Connection → Connect" to set the connection parameters. Select the "TCP/IP" way to connect, set the "Response Timeout" ranges from 15 to 30 seconds, set the "IP Address" as "127.0.0.1" (fixed), and set the "Port" as the PAC's ID (PAC-PORT-ID, e.g., 20009, refer Section 22.2).



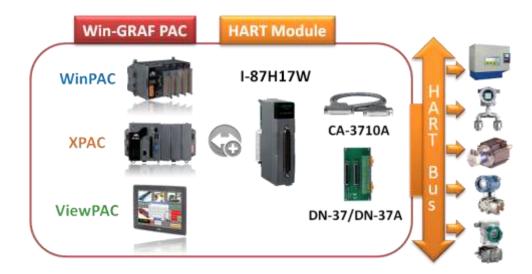
 Click "Setup → Poll Definition" to set the data type you want to read. Then, select "04 INPUT REGISTER" in the "Function" field (refer Section 22.3.2 – Step2) and click "OK" to view the read data shown on the Modbus Poll.

<b>3</b> 7 1	Modbus Poll -	Mbpoll1					
File	<u>C</u> onnection	Setup Functions	Display View	w <u>W</u> indow	<u>H</u> elp		
	I 🖻 🖬 🎒	Poll Definition	. <b>F2</b> 05	06 15 16	22		
Doc		Options 🔨	Poll Defini	ition			
			Slave ID:	1	1		OK
			Function:	04 INPUT I	REGISTE	R 🗲	
			Address:	1			Cancel
			Length:	10		·	Apply
			Scan Rate:	1000			
			Scan Kate:	11000	ms		Put our 1
			🔽 Auto Re	ad <u>E</u> nable			<u>R</u> ead Once
	👺 Mbpoll1						
	Tx = 146:	Err = 0:	ID = 1: H				
			10 1. 1	7 = 04:	SR =	1000ms	
	30001 =				SR =	1000ms	
	30001 = 30002 =	0 30		f = 04: 13 0	SR =	luuums	
		0 30 2015 30	006 =	13	SR =	1000ms	
	30002 = 30003 = 30004 =	0 30 2015 30 0 30 4 30	006 = 007 = 008 = 009 =	13 0 17 0	SR =	1000ms	
	30002 = 30003 =	0 30 2015 30 0 30 4 30	006 = 007 = 008 =	13 0 17	SR =	1000ms	

# Chapter 23 HART Master

The following Win-GRAF driver versions support the I-87H17W HART Master module.

WP-8xx8:1.07; XP-8xx8-CE6:1.05;



This chapter will describe three demo projects for communicating between the Win-GRAF PAC and the HART device via the I-87H17W HART module.

## 23.1 Introduction of the I-87H17W HART Module

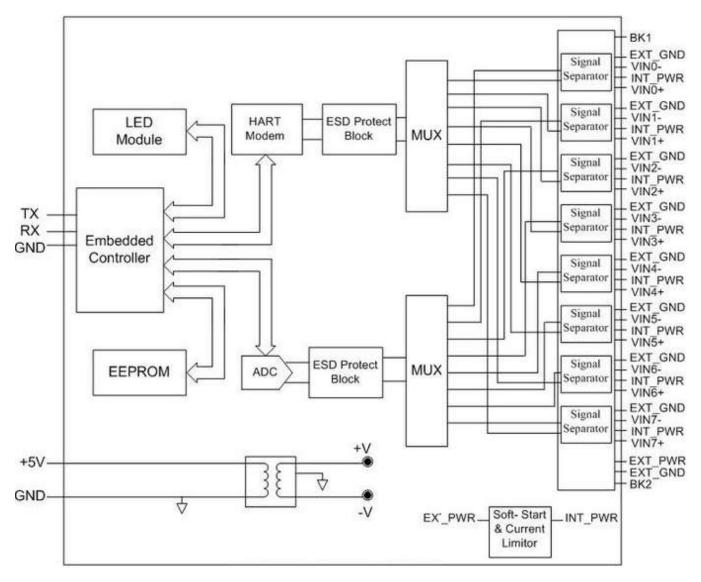
I-87H17W : http://www.icpdas.com/products/Remote IO/can bus/i-87h17w.htm

The XP-8xx8-CE6 PAC supports I-87H17W in its slot 1 to 7 (the first I/O slot No. is 1). And, the WP-8xx8 PAC supports I-87H17W in its slot 0 to 7. The Win-GRAF PAC doesn't support the I-87H17W which is plugged into the RS-485 remote expansion unit (like the I-87K8, RU-87P8), so only use it in the Win-GRAF PAC's I/O Slot.

Each I-87H17W has 8 analog input channels that used to measure 4 ~ 20 mA current and/or used as HART communication channels. Recommend to link only one HART device in each channel. Owing to these eight HART channels share with one communication chip inside this module, only single channel can be used at a time. In other words, I-87H17W unable to send/receive the HART frame via two or more channels at the same time. But, the user can use the Win-GRAF program (e.g., "Demo HART 3.zip") to process these 8 HART channels to communicate with HART devices by turns.

## Hardware Wiring:

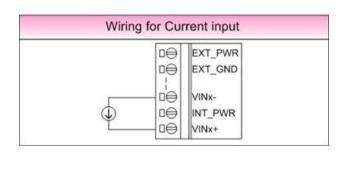
The user can visit the web page for more details on I-87H17W hardware. <u>http://www.icpdas.com/products/Remote\_IO/can\_bus/i-87h17w.htm</u> The following only lists its internal I/O structure, pin assignment and wire connection.



The pin assignment of I-87H17W:

i-87H17W	Pin Assignment Name	Te	erminal N	o.	Pin Assignment Name
8 CH HART Input	X EXT_PWR VIN7- VIN7+ VIN6- VIN6+ VIN5- VIN5+ VIN5+ VIN4- VIN4- VIN3- VIN3- VIN3+ VIN3- VIN3+ VIN2- VIN2+ VIN2+ VIN1- VIN1- VIN1+ VIN0- VIN0+ BK1	19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01	000000000000000000000000000000000000000	37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20	BK2 EXT_GND INT_PWR7 EXT_GND INT_PWR6 EXT_GND INT_PWR5 EXT_GND INT_PWR4 EXT_GND INT_PWR3 EXT_GND INT_PWR3 EXT_GND INT_PWR2 EXT_GND INT_PWR1 EXT_GND INT_PWR0 EXT_GND INT_PWR0 EXT_GND

## The wire connection of I-87H17W:



2-wire	24V		EXT_PWR EXT_GND
	L	i⊖	VINx-
	HART+		INT_PWR
	HART- °	O	VINx+
		□⊖	EXT_PWR
-wire			EXT_GND
	HART-0		VINx-
		□⊖	INT_PWR
	HART+0	0⊖	VINx+

# 23.2 The Format of the HART Protocol

Before describing the Win-GRAF projects, we need to understand the format of the HART protocol. The HART protocol is based on a Master-Slave communication scheme, using a "request-respond" way to communicate to each other for data exchanging.

## Notice:

- A. Refer the user manual of each HART device for knowing their supported HART format.
- B. The user can ignore the "Check Byte" in the Win-GRAF program because the I-87H17W module will automatically add/remove it when sending/receiving a HART frame.
- C. The HART physical layer is using 1200bps, 1 start-bit, Odd parity, 8 character-size and 1 stop-bit.

## The I-87H17W send a HART command to a device:

The "Byte Count" indicates the number of data bytes in the range 0 to 255.

1	Preamble	Delimiter	Address	Command	Byte Count	Data	Check byte
	5 - 20 bytes	1 byte	1 bytes (short) 5 bytes (long)	1 byte	1 byte	0 - 255 bytes	1 byte

## The device responds data to the I-87H17W:

The "Byte Count" indicates the number of response code and data bytes in the range 0 to 255.

Preamble	Delimiter	Address	Command	Byte Count	Response code	Data	Check byte
5 - 20 bytes	1 byte	1 bytes (short) 5 bytes (long)	1 byte	1 byte	2 bytes	0 - 253 bytes	1 byte

## Preamble: 5 - 20 bytes

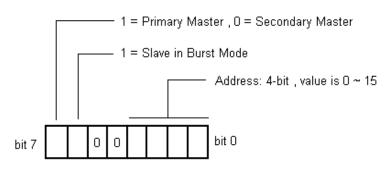
The value of each byte is "255" (i.e., 16#FF).

Delimiter: 1 byte

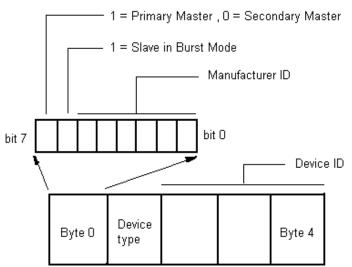
- 01: Short frame, Burst Frame.
- 02 : Short frame, Master to Slave.
- 06 : Short frame, Slave to Master.
- 129 (16#81): Long frame, Burst Frame.
- 130 (16#82): Long frame, Master to Slave.
- 134 (16#86): Long frame, Slave to Master.

Address: 1 byte (Short frame) or 5 bytes (Long frame)

## Short frame (1 byte):



## Long frame (5 bytes):



**Command:** 1 byte (refer the manual of the HART device for its definition).

Byte Count: 1 byte.

Send (I-87H17W): "Byte Count" is the byte amount of the "Data". The value is 0 to 255.

Response (device): "Byte Count" is the byte amount of "Data" + "Response code". The value is 0 to 255.

Response code:2 byte (refer the manual of the HART device for its definition).Data:Refer the manual of the HART device for its definition.

Check Byte:1 byte. The user can ignore the "Check Byte" in the Win-GRAF program because the<br/>I-87H17W module will automatically add (or remove) it when sending (or receiving)<br/>the HART frame.

# 23.3 Introduction of the Win-GRAF Demo Project

This section lists three Win-GRAF projects to show you the way to do HART communication between the Win-GRAF PAC and HART devices. The user can run the Win-GRAF Workbench and then click "File  $\rightarrow$  Add Existing Project  $\rightarrow$  From Zip..." to open the Win-GRAF project, which is stored in a shipping CD (CD-ROM: \Napdos\Win-GRAF\demo-project\).

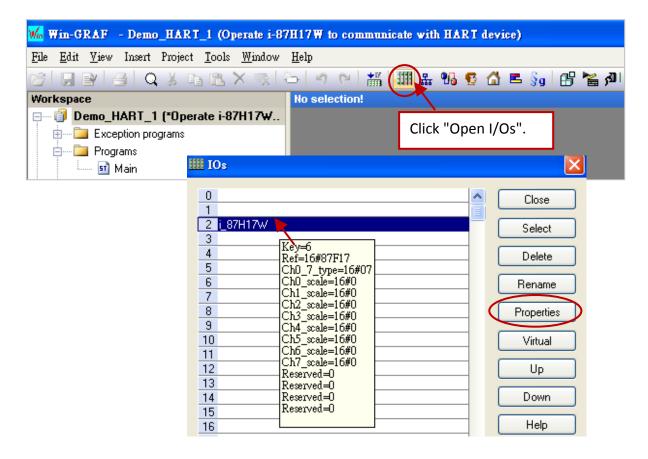
File Name	Description
Demo_HART_1.zip	Sending the HART frame manually from Ch-0 of the I-87H17W in the slot 2, and then receive the HART frame from the device.
Demo_HART_2.zip	Sending and receiving the HART frame every 5 seconds automatically and repeatedly from Ch-0, Ch-1 and Ch-2 of the I-87H17W in the slot 2.
Demo_HART_3.zip	(It's similar as Demo_HART_2) Using two I-87H17W in the slot 2 and slot 3 to send and receive the HART frame every 5 seconds automatically and repeatedly from Ch-0 to Ch-7 at the same time.

<u>Note</u>: Because all 8 channels of the I-87H17W are shared with one HART chip. Each I-87H17W can conduct only one channel for sending/receiving HART frame at a time, and then conduct the next channel.

# 23.3.1 I/O Board Setting and the HART Functions

## I/O Board Setting

Both the "Demo\_HART\_1" and the "Demo\_HART\_2" projects require to use one I-87H17W on the PAC's Slot 2. Therefore, we need to enable this I/O function in the Win-GRAF "I/O Boards" window. Simply click the "Open I/Os" button and then add "i\_87H17W" in the slot no. 2, click "Properties" can view more details about it (or see <u>Chapter 4</u> – Linking I/O boards).



"I-87H17W" is an eight channels HART current input module (data type: REAL; input range: 4 - 20mA).

## Parameters:

## Ch0\_scale ~ Ch7\_scale : 16#SS

SS : Scaling function is defined by the "i\_scale" I/O board (refer the <u>Section 4.2</u>).
 00 means "No scaling".

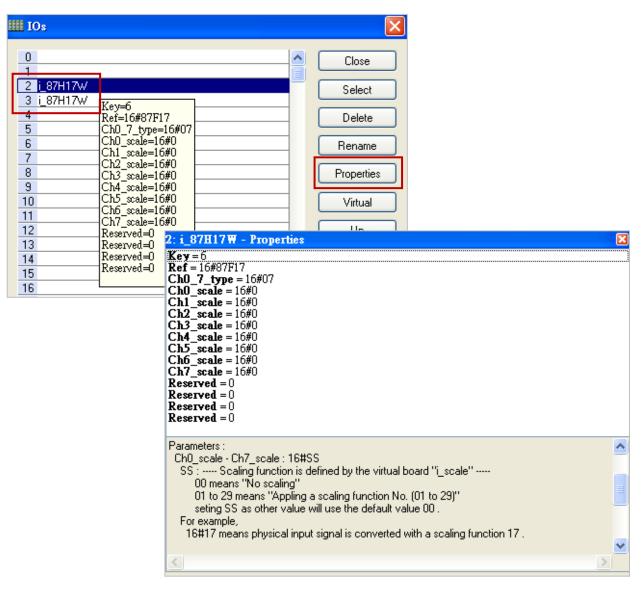
01 to 29 means "Appling a scaling function No. (01 to 29)"

Setting SS as other value will use the default value 00.

For example, 16#17 means the physical input signal is converted with the scaling function 17.

Note: If the Current value is greater than "9000.0", which means the HART device may be disconnected.

In the "Demo\_HART\_3" project, it need to enable two "i\_87H17W" in the slot no. 2 and no.3.



#### The HART Functions

The user can use the "HART\_Read", "HART\_Send" and "HART\_Status" functions in the ST or LD program. In addition, you can press "F1" or open "Library Manager" (refer <u>Section 1.2.3</u>) to view the descriptions of these functions.

🚾 Library Manager - ICP DAS - XP-WP-YP	×
<u>File Wi</u> zard <u>H</u> elp	
Function and FBs I/Os Profiles AS-i Types	
Hart_Read (* Read Hart Response *)	~
FHart_Send (* Send Hart Request *)	
FHart_State (* Check Hart bus state *)	~
Parameters Description	
Hart_Read : to Get the response from HART bus	^
Noticed:	
This C-function must work with the HART module, I-87H17W, in the backplane.	
*** Input parameters	

Hart\_Read: To get the response from HART Bus.
 <u>Note</u>: This function must work with the HART module (i.e., I-87H17W), which is plugged in the Win-GRAF PAC.

#### Input Parameters:

gSlot:	(Data type: DINT)
	The specified I/O slot which the I-87H17W is plugged in. It used to read the response
	frame from the HART bus. (The value range is $0 \sim 7$ )
@gPreamble:	(Data type: USINT) (Refer <u>Section 23.2</u> – The format of HART frame.)
	To get the Preamble number of HART response.
@gDelimiter:	(Data type: USINT) To get the Delimiter of the HART response.
gAddress[]:	(Data type: USINT) To get the Address of the HART response.
@gCommand:	(Data type: USINT) To get the Command of the HART response.
gData[]:	(Data type: USINT) To get the Data of the HART response.
@gDataLen:	(Data type: USINT) To get the data length of the HART response.

#### **Output Parameters:**

- Q: (Data type: DINT)
  - 0 : Nothing happened.
  - 1 : Read success.
  - -1 : Cannot find the HART module, I-87H17W, in the specified slot.
  - -2 : Invalid HART response.
  - -3 : Waiting HART response.
  - -4 : No HART request.
  - -5 : HART module I-87H17 is off line.

hart\_Send: To send the request to HART bus.

<u>Note</u>: This function must work with the HART module (i.e., I-87H17W), which is plugged in the Win-GRAF PAC.

#### Input Parameters:

gSlot:	Data type: DINT) The specified I/O slot which the I-87H17W is plugged in. It used to send the HART request. (The value range is 0 - 7.)
gChannel:	(Data type: DINT) The specified channel of I-87H17W to send HART request to HART device. (The value range is 0 - 7.)
gPreamble:	(Data type: USINT) (Refer <u>Section 23.2</u> – The format of HART frame.) The Preamble amount of HART request (The value range is 5 - 20.)
gDelimiter:	(Data type: USINT)
	The Delimiter of the HART request.
gAddress[]:	(Data type: USINT)
	The Address of the HART request.
gCommand:	(Data type: USINT)
	The Command of the HART request.
gData[]:	(Data type: USINT)
	The Data of the HART request.
gDataLen:	(Data type: USINT)
	The Data length (byte) of the HART request.

#### **Output Parameters:**

- Q: (Data type: DINT)
  - 0: Nothing happened.
  - 1: Read success.
  - -1 : Cannot find the HART module, i-87H17W, in the specified slot.
  - -2 : The specified channel of I-87H17W is wrong.
  - -3 : The specified Preamble amount is wrong.
  - -4 : The sending error of the HART request, please check Delimiter and Address is correct.
  - -5 : HART bus is busy.
  - -6 : HART module I-87H17 is off line.

Hart\_State: To Get HART bus state.

**Note**: This function must work with the HART module (i.e., I-87H17W), which is plugged in the Win-GRAF PAC.

#### Input Parameters:

gSlot: (Data type: DINT) The specified I/O slot which the I-87H17W is plugged in. It used to get the status of the HART bus.

#### **Output Parameters:**

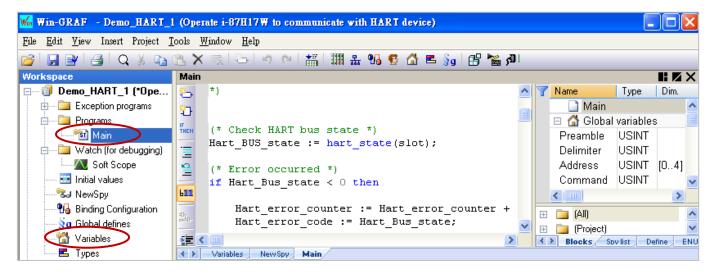
- Q: (Data type: DINT)
  - 0 : No any query.
  - 1 : HART request is waiting to send to HART device.
  - 2 : HART request is sent to HART device.
  - 3 : Waiting the HART response.
  - 4 : HART response is received.
  - -1 : Timeout.
  - -2 : The length of the HART response frame is too short.

```
(Refer Section 23.2 – The format of HART frame.)
```

- -3 : The Delimiter of the HART response is wrong.
- -4 : The Address (Master) of the HART response is wrong.
- -5 : The Address (Burst) of the HART response is wrong.
- -6 : The Command of the HART response is wrong.
- -7 : The Checksum of the HART response is wrong.
- -8 : HART response error.
- -98 : There is some mismatch between the parameter of the HART response and the HART request.
- -99 : Impossible error.
- -100 : The error happened during sending the HART request to the HART device.
- -101 : Cannot find HART module I-87H17W in the specified slot.
- -102 : HART module I-87H17 is off line.

## 23.3.2 The Demo Project - "Demo\_HART\_1"

(You can see the feature description in the ST program below.)



#### **Description of Variables:**

Variables Name	Data Type	Dim.	Description
slot	DINT	-	
channel	DINT	-	
Preamble	USINT	-	Used to send the HART request.
Delimiter	USINT	-	(The initial value of "slot" is set to "2")
Address	USINT	[04]	(Refer <u>Section 23.2</u> - The format of HART frame.
Command	USINT	-	<u>Section 23.3.1</u> - "HART_Send" function)
Data	USINT	[0254]	
Datalen	USINT	-	
ret	DINT	-	Send/receive the status value of the HART frame.
Preamble_r	USINT	-	
Delimiter_r	USINT	-	Used to receive the HART response.
Address_r	USINT	[04]	(Refer <u>Section 23.2</u> - The format of HART frame.
Command_r	USINT	-	Section 23.3.1 - "HART_Send" function)
Data_r	USINT	[0254]	
Datalen_r	USINT	-	
Hart_BUS_state	DINT	-	To get the status of the HART bus.
Hart_error_counter	DINT	-	To calculate error times of the HART bus.
Hart_error_code	DINT	-	The error code of the HART bus.
hart_bus_err_Msg	STRING(255)	-	The error message of the HART bus.
hart_send_err_msg	STRING(255)	-	The error message when sending the HART frame.
hart_read_err_msg	STRING(255)	-	The error message when reading the HART frame.
send_HART_request	BOOL	-	Set it as "TRUE" to send the HART frame.
read_HART_request	BOOL	-	Set it as "TRUE" to read the HART frame.

#### ST Program (Main)

(\* This demo project is to show how to use one I-87H17W(HART Master) to communicate manually with one HART device.

Hardware Environment:

- 1. Plug i-87H17W in the slot 2 of Win-GRAF PAC.
- 2. Connect HART device to the channel 0 of i-87H17W \*)

```
(* Check HART bus state *)
```

```
Hart_BUS_state := hart_state(slot);
```

#### (\* Error message and error times of the HART Bus \*)

```
if Hart_Bus_state < 0 then
```

```
Hart_error_counter := Hart_error_counter + 1;
Hart_error_code := Hart_Bus_state;
```

```
case Hart_error_code of
```

```
-1 : hart_bus_err_Msg := 'Error: Timeout' ;
```

- -2 : hart\_bus\_err\_Msg := 'Error: Read\_data\_too\_short error' ;
- -3 : hart\_bus\_err\_Msg := 'Error: Response Delimiter error' ;
- -4 : hart\_bus\_err\_Msg := 'Error: Response addr\_master error' ;
- -5 : hart\_bus\_err\_Msg := 'Error: Response addr\_burst error' ;
- -6 : hart\_bus\_err\_Msg := 'Error: Response recv\_command error' ;
- -7 : hart\_bus\_err\_Msg := 'Error: Response checksum error' ;

```
-8 : hart_bus_err_Msg := 'Error: response error' ;
```

```
-98 : hart_bus_err_Msg := 'Error: para_mismatch error' ;
```

```
-99 : hart_bus_err_Msg := 'Error: impossible error' ;
```

```
-100: hart_bus_err_Msg := 'Error: Sending HART request error';
```

```
-101: hart_bus_err_Msg := 'Error: cannot find the HART moudule, i-87H17W';
```

-102: hart\_bus\_err\_Msg := 'Error: i-87H17W is off-line';

end\_case;

end\_if;

```
if Hart_BUS_state = 0 and Send_HART_request then
(* HART bus is free to send *)
```

```
(* Build HART command *)
slot := 2;
channel := 0;
Preamble := 5;
Delimiter := 16#82;
Address[0] := 16#96;
Address[1] := 16#85;
```

```
Address[2] := 16#0B;
Address[3] := 16#0A;
Address[4] := 16#42;
Command := 16#03;
Datalen := 0;
```

ret := hart\_send(slot, channel, Preamble, Delimiter, Address, Command, Data, Datalen);

```
case ret of
1: Send_HART_request := false;
hart_send_err_msg := 'Send success';
-1: hart_send_err_msg := 'Send Error: cannot find the HART moudule, i-87H17W';
-2: hart_send_err_msg := 'Send Error: specified the wrong channel to send hart cmd';
-3: hart_send_err_msg := 'Send Error: specified the wrong preamble number';
-4: hart_send_err_msg := 'Send Error: wrong HART cmd, please check delimiter and
address is correct';
-5: hart_send_err_msg := 'Send Error: HART bus is busy';
-6: hart_send_err_msg := 'Send Error: i-87H17W is off-line';
end_case;
```

```
end_if;
```

```
(* HART response is coming *)
```

```
if Hart_BUS_state = 4 then
```

```
ret := hart_read(slot, Preamle_r, Delimiter_r, Address_r, Command_r, Data_r, Datalen_r);
```

```
case ret of
1: hart_read_err_msg := 'Read success';
    read_HART_request := true;
-1: hart_read_err_msg := 'Read Error: cannot find the HART moudule, i-87H17W';
-2: hart_read_err_msg := 'Read Error: invalid HART response';
-3: hart_read_err_msg := 'Read Error: waiting response';
-4: hart_read_err_msg := 'Read Error: No Request';
-5: hart_read_err_msg := 'Read Error: i-87H17W is off-line';
end_case;
```

end\_if;

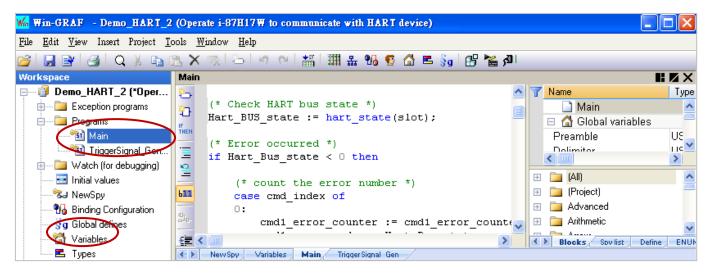
```
(* HART response processing *)
if read_HART_request then
read_HART_request := false;
```

#### (\* ToDo: put your code here to handle the HART response \*)

end\_if;

### 23.3.3 The Demo Project - "Demo\_HART\_2"

(You can see the feature description in the ST program below.)



#### **Description of Variables:**

Variables Name	Data Type	Dim.	Description
slot	DINT	-	
channel	DINT	-	
Preamble	USINT	-	Used to send the HART request.
Delimiter	USINT	-	(The initial value of "slot" is set to "2") (Refer <u>Section 23.2</u> - The format of HART frame.
Address	USINT	[04]	Section 23.3.1 - "HART_Send" function)
Command	USINT	-	
Data	USINT	[0254]	
Datalen	USINT	-	
ret	DINT	-	Send/receive the status value of the HART frame.
Preamble_r	USINT	-	
Delimiter_r	USINT	-	Used to receive the HART response.
Address_r	USINT	[04]	(Refer <u>Section 23.2</u> - The format of HART frame.
Command_r	USINT	-	<pre><u>Section 23.3.1</u> - "HART_Send" function)</pre>
Data_r	USINT	[0254]	
Datalen_r	USINT	-	
Hart_BUS_state	DINT	-	To get the status of the HART bus.
hart_bus_err_Msg	STRING(255)	-	The error message of the HART bus.
hart_send_err_msg	STRING(255)	-	The error message when sending the HART frame.
hart_read_err_msg	STRING(255)	-	The error message when reading the HART frame.
send_HART_request	BOOL	-	Set it as "TRUE" to send the HART frame.
read_HART_request	BOOL	-	Set it as "TRUE" to read the HART frame.
cmd_index	DINT	-	The index value to process the command.
cmd_max_num	DINT	-	The number of sending commands. (Init. = 3)

Variables Name	Data Type	Dim.	Description		
cmd1_response_counter	DINT	-			
cmd2_response_counter	DINT	-	The number of response times. (From Ch-0 / Ch-1/ Ch-2)		
cmd3_response_counter	DINT	-			
cmd1_error_counter	DINT	-			
cmd2_error_counter	DINT	-	The number of error times on the HART bus. (From Ch-0 / Ch-1/ Ch-2)		
cmd3_error_counter	DINT	-			
cmd1_error_code	DINT	-			
cmd2_error_code	DINT	-	The error code of the HART bus. (From Ch-0 / Ch-1/ Ch-2)		
cmd3_error_code	DINT	-			
trigger_Timer	TIME	-	Start ticking.		
trigger_interval	TIME	-	The interval time for sending commands. (Init = T#5s)		
INIT	BOOL	-	Initialize (Init. = TRUE).		
next_index	DINT	-	The index for the response data.		
cmd1_Data	REAL	[03]			
cmd2_Data	REAL	[03]	Used to save the response data. (From Ch-0 / Ch-1/ Ch-2)		
cmd3_Data	REAL	[03]			

#### ST Program (Main)

(\* This demo project is to show how to use one I-87H17W(HART Master) to communicate automatically with HART devices that is connected with different channels of I-87H17W.

Hardware Environment:

1. Plug I-87H17W in the slot 2 of Win-GRAF PAC.

2. Connect three HART devices to the channel 0 ~ 2 of i-87H17W in the slot 2 of Win-GRAF PAC separeately.

\*)

```
(* Check HART bus state *)
Hart_BUS_state := hart_state(slot);
```

```
(* Error occurred *)
```

if Hart\_Bus\_state < 0 then

```
(* Count the error number *)
```

```
case cmd_index of
0:
    cmd1_error_counter := cmd1_error_counter + 1;
    cmd1_error_code := Hart_Bus_state;
1:
    cmd2_error_counter := cmd2_error_counter + 1;
    cmd2_error_code := Hart_Bus_state;
```

```
2:
    cmd3_error_counter := cmd3_error_counter + 1;
    cmd3 error code := Hart Bus state;
  end case;
  (* Try to send the next HART request *)
  cmd index := cmd index + 1;
  if cmd index = cmd max num then
    cmd index := 0;
  end if;
  case Hart Bus state of
   -1 : hart_bus_err_Msg := 'Error: Timeout' ;
   -2 : hart bus err Msg := 'Error: Read data too short error';
   -3 : hart bus err Msg := 'Error: Response Delimiter error';
   -4 : hart bus err Msg := 'Error: Response addr master error' ;
   -5 : hart_bus_err_Msg := 'Error: Response addr_burst error' ;
   -6 : hart bus err Msg := 'Error: Response recv command error' ;
   -7 : hart_bus_err_Msg := 'Error: Response checksum error' ;
   -8 : hart bus err Msg := 'Error: response error';
   -98 : hart bus err Msg := 'Error: para mismatch error' ;
   -99 : hart_bus_err_Msg := 'Error: impossible error' ;
  -100: hart bus err Msg := 'Error: Sending HART request error';
  -101: hart bus err Msg := 'Error: cannot find the HART moudule, i-87H17W';
  -102: hart bus err Msg := 'Error: i-87H17W is off-line';
  end_case;
end if;
if Hart BUS state = 0 and Send HART request then
(* HART bus is free to send *)
  (* Build HART command *)
  case cmd index of
  0:
    slot := 2;
    channel := 0;
    Preamble := 5;
    Delimiter := 16#82;
    Address[0] := 16#96;
    Address[1] := 16#85;
    Address[2] := 16#0B;
    Address[3] := 16#0A;
    Address[4] := 16#42;
    Command := 16#03;
    Datalen := 16#00;
```

```
Win-GRAF User Manual, V 1.07, Apr. 2018 by ICP DAS 23-15
```

1: slot := 2; channel := 1; Preamble := 5; Delimiter := 16#82; Address[0] := 16#96; Address[1] := 16#85; Address[2] := 16#0B; Address[3] := 16#0A; Address[4] := 16#42; Command := 16#03; Datalen := 16#00; 2: slot := 2; channel := 2; Preamble := 5; Delimiter := 16#82; Address[0] := 16#96; Address[1] := 16#85; Address[2] := 16#0B; Address[3] := 16#0A; Address[4] := 16#42; Command := 16#03; Datalen := 16#00; end case;

ret := hart\_send(slot, channel, Preamble, Delimiter, Address, Command, Data, Datalen);

case ret of
1: Send\_HART\_request := false;
hart\_send\_err\_msg := 'Send success';
-1: hart\_send\_err\_msg := 'Send Error: cannot find the HART moudule, i-87H17W';
-2: hart\_send\_err\_msg := 'Send Error: specified the wrong channel to send hart cmd';
-3: hart\_send\_err\_msg := 'Send Error: specified the wrong preamble number';
-4: hart\_send\_err\_msg := 'Send Error: wrong HART cmd, please check delimiter and address is
correct';

- -5: hart\_send\_err\_msg := 'Send Error: HART bus is busy';
- -6: hart\_send\_err\_msg := 'Send Error: i-87H17W is off-line';

```
end_case;
```

end\_if;

```
(* HART response is coming *)
```

```
if Hart_BUS_state = 4 then
```

```
ret := hart_read(slot, Preamle_r, Delimiter_r, Address_r, Command_r, Data_r, Datalen_r);
```

```
case ret of
1: hart_read_err_msg := 'Read success';
  read_HART_request := true;
```

```
-1: hart_read_err_msg := 'Read Error: cannot find the HART moudule, i-87H17W';
-2: hart_read_err_msg := 'Read Error: invalid HART response';
-3: hart_read_err_msg := 'Read Error: waiting response';
-4: hart_read_err_msg := 'Read Error: No Request';
-5: hart_read_err_msg := 'Read Error: i-87H17W is off-line';
end_case;
```

end\_if;

#### (\* HART response processing \*)

```
if read_HART_request then
    read_HART_request := false;
```

```
(*--- Please use your own <Data> definition for your own application ----
  The data of the HART response assume as 00 00 41 A0 FF 3E 0C 3E C5 37 48 20 41 C8 3F 22 39
 42 C9 8E D1
 The HART answer is from a device, which <Data> in big endian has the following meaning.
  41 A0 FF 3E = 41A0FF3E < convert these 4 bytes (IEEE-754) to one REAL> : 20.1246 mA
  3E C5 37 48 = 3EC53748 < convert these 4 bytes (IEEE-754) to one REAL> : 0.385187
  41 C8 3F 22 = 41C83F22 < convert these 4 bytes (IEEE-754) to one REAL> : 25.0308
  42 C9 8E D1 = 42C98ED1 < convert these 4 bytes (IEEE-754) to one REAL> : 100.779
                                                                                         *)
case cmd index of
0:
  cmd1 response counter := cmd1 response counter + 1;
  next index := SerializeIn(Data r, cmd1 Data[0], 2, true);
  next index := next index + 1;
  next index := SerializeIn(Data r, cmd1 Data[1], next index, true);
  next index := next index + 1;
  next index := SerializeIn(Data r, cmd1 Data[2], next index, true);
  next index := next index + 1;
  next index := SerializeIn(Data r, cmd1 Data[3], next index, true);
1:
  cmd2 response counter := cmd2 response counter + 1;
  next index := SerializeIn(Data r, cmd2 Data[0], 2, true);
  next_index := next_index + 1;
  next index := SerializeIn(Data r, cmd2 Data[1], next index, true);
  next index := next index + 1;
  next index := SerializeIn(Data r, cmd2 Data[2], next index, true);
  next index := next index + 1;
  next index := SerializeIn(Data r, cmd2 Data[3], next index, true);
2:
  cmd3 response counter := cmd3 response counter + 1;
  next index := SerializeIn(Data r, cmd3 Data[0], 2, true);
  next index := next index + 1;
```

```
next_index := SerializeIn(Data_r, cmd3_Data[1], next_index, true);
next_index := next_index + 1;
next_index := SerializeIn(Data_r, cmd3_Data[2], next_index, true);
next_index := next_index + 1;
next_index := SerializeIn(Data_r, cmd3_Data[3], next_index, true);
end_case;
```

#### (\* Try to send the next HART request \*)

```
cmd_index := cmd_index + 1;
Send_HART_request := true;
if cmd_index = cmd_max_num then
        cmd_index := 0;
end_if;
```

end\_if;

#### ST Program (TriggerSignal)

```
(*
To start the timer for generating the trigger signal to send HART request
*)
```

```
if INIT then
    INIT := false;
    TStart(trigger_timer);
    send_HART_request := true;
end_if;
```

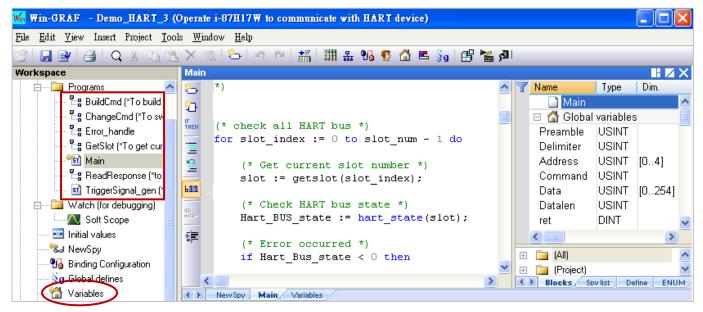
```
if trigger_Timer > trigger_interval then
```

```
trigger_Timer := T#0s;
send_HART_request := true;
```

```
end_if;
```

### 23.3.4 The Demo Project - "Demo\_HART\_3"

(You can see the feature description in the ST program below.)



#### **Description of Variables:**

This demo project is similar as the **"Demo\_HART\_2"** project. The user can refer its variable descriptions or open the "Demo\_HART\_3" project to view all used variables in the Win-GRAF "Variables" window .

#### ST Program (Main)

There are several subroutine used in this project, the user can open them to view the program code. The following description is focused on the "Main" program.

#### (\*

This demo project is to show how to use two I-87H17W (HART Master) to communicate automatically with HART devices that is connected with different channels of I-87H17W.

#### Hardware Environment:

- 1. Plug one I-87H17W in the slot 2 of Win-GRAF PAC
- 2. Plug the other one in the slot 3 of Win-GRAF PAC.
- 3. Connect 8 HART devices to the channel 0 ~ 7 of I-87H17W in the slot 2 of Win-GRAF PAC separeately.
- 4. Connect 8 HART devices to the channel 0 ~ 7 of I-87H17W in the slot 3 of Win-GRAF PAC separeately. \*)

(\* Check all HART bus \*) for slot index := 0 to slot num - 1 do

```
(* Get current slot number *)
slot := getslot(slot_index);
```

```
(* Check HART bus state *)
Hart_BUS_state := hart_state(slot);
```

(\* Error occurred \*) if Hart\_Bus\_state < 0 then

```
(* Count the error number *)
Error_handle(any_to_byte(slot_index));
```

```
(* Try to send the next HART request *)
ChangeCmd(any_to_byte(slot_index));
```

```
case Hart_Bus_state of
-1 : hart_bus_err_Msg := 'Error: Timeout' ;
-2 : hart_bus_err_Msg := 'Error: Read_data_too_short error' ;
-3 : hart_bus_err_Msg := 'Error: Response Delimiter error' ;
-4 : hart_bus_err_Msg := 'Error: Response addr_master error' ;
-5 : hart_bus_err_Msg := 'Error: Response addr_burst error' ;
-6 : hart_bus_err_Msg := 'Error: Response recv_command error' ;
-7 : hart_bus_err_Msg := 'Error: Response checksum error' ;
-8 : hart_bus_err_Msg := 'Error: response error' ;
-98 : hart_bus_err_Msg := 'Error: para_mismatch error' ;
-99 : hart_bus_err_Msg := 'Error: Sending HART request error' ;
-100: hart_bus_err_Msg := 'Error: cannot find the HART moudule, i-87H17W' ;
-102: hart_bus_err_Msg := 'Error: i-87H17W is off-line';
end_case;
```

end\_if;

```
if Hart_BUS_state = 0 and Send_HART_request[slot_index] then
(* HART bus is free to send *)
```

```
(* Build HART command *)
buildCmd(any_to_byte(slot_index));
```

```
ret := hart_send(slot, channel, Preamble, Delimiter, Address, Command, Data, Datalen);
```

```
case ret of
```

```
1: Send_HART_request[slot_index] := false;
```

```
hart_send_err_msg := 'Send success';
```

-1: hart\_send\_err\_msg := 'Send Error: cannot find the HART moudule, i-87H17W';

```
-2: hart_send_err_msg := 'Send Error: specified the wrong channel to send hart cmd';
```

```
-3: hart_send_err_msg := 'Send Error: specified the wrong preamble number';
```

```
-4: hart_send_err_msg := 'Send Error: wrong HART cmd, please check delimiter and address
is correct';
```

```
-5: hart_send_err_msg := 'Send Error: HART bus is busy';
```

```
-6: hart_send_err_msg := 'Send Error: i-87H17W is off-line';
end case;
```

end\_if;

#### (\* HART response is coming \*)

```
if Hart_BUS_state = 4 then
    ret := hart_read(slot, Preamle_r, Delimiter_r, Address_r, Command_r, Data_r, Datalen_r);
```

```
case ret of
1: hart_read_err_msg := 'Read success';
    read_HART_request := true;
-1: hart_read_err_msg := 'Read Error: cannot find the HART moudule, i-87H17W';
-2: hart_read_err_msg := 'Read Error: invalid HART response';
-3: hart_read_err_msg := 'Read Error: waiting response';
-4: hart_read_err_msg := 'Read Error: No Request';
-5: hart_read_err_msg := 'Read Error: i-87H17W is off-line';
end_case;
```

```
end_if;
```

#### (\* HART response processing \*)

```
if read_HART_request then
    read_HART_request := false;
```

```
ReadResponse(any_to_byte(slot_index));
ChangeCmd(any_to_byte(slot_index));
end_if;
end_for;
```

## 23.4 Test the Demo Program

#### **Feature Descriptions:**

File Name	Description
Demo_HART_1.zip	Sending the HART frame manually from Ch-0 of the I-87H17W in the slot 2, and then receive the HART frame from the device.
Demo_HART_2.zip	Sending and receiving the HART frame every 5 seconds automatically and repeatedly from Ch-0, Ch-1 and Ch-2 of the I-87H17W in the slot 2.
Demo_HART_3.zip	(It's similar as Demo_HART_2) Using two I-87H17W in the slot 2 and slot 3 to send and receive the HART frame every 5 seconds automatically and repeatedly from Ch-0 to Ch-7 at the same time.

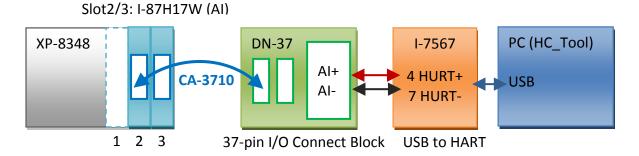
<u>Note</u>: Because all 8 channels of the I-87H17W are shared with one HART chip. Each I-87H17W can conduct only one channel for sending/receiving HART frame at a time, and then conduct the next channel.

### 23.4.1 Testing Environment and the "HC\_Tool" Setting

Before testing, make sure the I-87H17W module is plugged in the slot2 (or slot3) of the Win-GRAF PAC, and the needed I/O channels (Ch-0 ~ Ch-7) are linked to HART devices (refer <u>Section 23.1</u>). Then, power on the Win-GRAF PAC and download the project to it.

The following will describe the test way for these Win-GRAF demo programs one by one. Here, we use the "HC\_Tool" utility to simulate a HART device to conduct the HART response. The user can download "HC\_Tool" and its user manual on the I-7567 webpage (as the list below).

Refer the following product page (Pin Assignment or Wire Connection) to know the way of wiring.



#### **Testing Enviorment:**

Device	Model	Quantity	Note			
Win-GRAF PAC	<u>XP-8348-CE6</u>	1	-			
HART AI Module	<u>I-87H17W</u>	2	The "Demo_HART_1" and "Demo_HART_2" projects require only one I-87H17W.			
USB to HART Converter	<u>I-7567</u> 1		To download the USB driver and the "HC_Tool" utility.			
Website: <u>www.icpdas.con</u>	Website: <u>www.icpdas.com</u> > <u>Product</u> > <u>Industrial Communication</u> > <u>Fieldbus Solutions</u> > <u>HART Series</u>					
I/O Connector Block DN-37		1	Includes one CA-3710 cable.			
Website: <u>www.icpdas.com</u> > <u>Product</u> > <u>PC based I/O Board</u> > <u>PCI Bus I/O Boards</u> > <u>Daughter Boards</u>						

#### The HC\_Tool Setting:

Run "HC\_Tool.exe" and then click "Setting" to select the COM Port no. which your PC used (see "Device Manager"). You can keep the settings like the figure below (or see <u>I-7567 manual</u>) and click "OK", and then click "Open" to open this COM Port.

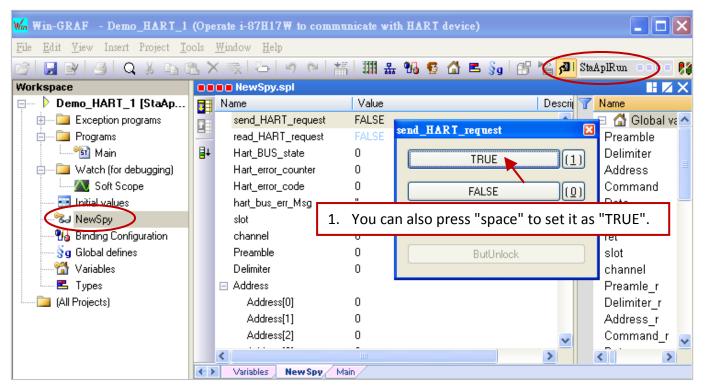
HC Tool v1.05 (ICP DAS)	
Settings Data Log SRMsg HTCfg 1 1. 3.	ModCfg About
COM6 : Open Close	Settings
Search : Start Stop	Com Port 2.
Status : Idle	Port Name : COM6 Timeout (ms) : 650
Information :	🔲 I-7570: 115200 🔍 N 💟 8 💟 1 💟
□ 🝠 連接埠 (COM 和 LPT)	HART (For Cmd 0)
	Auto Configure : Enable 🛛 HT Channel : 0 🔽
	Frame type : Long Master type : Primary
	Preambles : 5 Address : 0
	Manufacturer ID : 22 Device type : 133
	Device ID : 723522
	OK Cancel

### 23.4.2 Test the "Demo\_HART\_1", "Demo\_HART\_2" and "Demo\_HART\_3" Projects

Feature Descriptions: (Program- <u>Demo\_HART\_1</u>; Wiring – <u>See Section 23.4.1</u>)

Sending the HART frame manually from Ch-0 of the I-87H17W in the slot 2, and then receive the HART frame from the device.

After downloading this project, open the spy list (NewSpy) and set the "send\_HART\_request" variable as "TRUE" to send the HART request.



In the "HC\_Tool", click "SRMsg" to open "Send & Receive Msg" window, fill in the response data and click "Send".

HC_Tool v1.05 (ICP DAS) Settings Data Log SRMs; COM6 : Open CO		Refer <u>23.4.1</u> for COM	Port setting.
Send & Receive Msg			
FF F FF 3	ill in the response data and FF FF FF 86 96 85 0B 0A 4 FE OC 3E C5 37 48 20 41 C8 3F 22 OB 0A 4 E OC 3E C5 37 48 20 41 C8 3 fer <u>Section 23.2</u> - The forma	click "Send". 12 03 15 00 00 41 A0 3F 22 39 42 C9 8E D1	Send
□ Auto Scroll Receive Data 下午 03:07:04.275<=FF FF FF FF FF 82 96 85 0	IB 0A 42 03 00 D1	<b>,</b>	Clear
		t" as "TRUE" to send this 7H17W (refer <u>Section 23.</u>	

Preamble	Delimiter	Address	Command	Byte Count	Data		
The HART request	The HART request from the I-87H17W (Hex.)						
FF FF FF FF FF	FF FF FF FF FF	FF FF FF FF FF	FF FF FF FF FF	FF FF FF FF FF	FF FF FF FF FF		
The HART respons	The HART response from the device (Hex.)						
FF FF FF FF FF	86	(long) 96 85 0B 0A 42	03	15 <sub>(16)</sub>	00 00 41 A0 FF 3E 0C 3E C5 37 48 20 41 C8 3F 22 39 42 C9 8E D1		

If the process is successful, you will see the screen like below.

	NewSpy.spl			
	Name	Value	Description	
	send_HART_request	FALSE		~
	read_HART_request	FALSE		
₽Ļ	Hart_BUS_state	0		
_	Hart_error_counter	0		
	Hart_error_code	0		
	– – hart_bus_err_Msg			
		2	7	
	channel	0		
	Preamble	5		
	Delimiter	130		
	Address			
	Address[0]	150	The HART request from I-87H17W's	
	Address[1]	133	Ch-0 in the Slot2 (refer <u>Section 23.2</u> ).	
	Address[2]	11		
	Address[3]	10		
	Address[4]	66		
	Command	3		
	🗄 Data	-		
	Datalen	0		
	hart_send_err_msg	'Send success'		
	Preamle_r	5		
	_ Delimiter_r	16#86		
	⊟ Address_r			
	 Address_r[0]	150		
	Address_r[1]	133		
	Address_r[2]	11		
	Address_r[3]	10		
	Address_r[4]	66	The HART response from the device	
	Command_r	3	(refer <u>Section 23.2</u> The HART format).	
	⊡ Data_r			
	Data_r[0]	0		
	Data_r[1]	0		
	Data_r[2]	65		
	Data_r[3]	160		~
< >		lain		
	Data_r[250]	0		
	Data_([250] Data_r[251]	0		
	Data_([257] Data_r[252]	0		
	Data_([252] Data_r[253]	0		
		0		
	Data_r[254]			
	Datalen_r hart_read_err_msg	16#15 'Read success'	<b>→</b> <'	
< >		fain	v 	~
	anabies A new spy in			

#### Feature Descriptions: (Program- <u>Demo HART 2</u>; Wiring – <u>See Section 23.4.1</u>)

To send/receive the HART frame automatically and repeatedly every 5 seconds from the Ch-0, Ch-1 and Ch-2 of the I-87H17W module in the slot 2.

After downloading this project, open the spy list (NewSpy) and the I-87H17W will automatically send the HART request from Ch-0, Ch-1 and Ch-2.

🦝 Win-GRAF - Demo_HART_2 (Operate i-87H17W to communicate with HART device)										
<u>F</u> ile <u>E</u> dit <u>V</u> iew Insert Project <u>T</u> o	ols	<u>W</u> indow <u>H</u> elp								
경 🛃 🖻   크   Q 🐰 📭 🖭 🗙 🛼 🏷   어 🗠   👬   珊 🏪 😘 😨 🖓 🖻 🚱   🏽 🌘 👔 StaApiRum 💿 👀 🎉 🏭 🕿 💷										
Workspace	Vorkspace NewSpy.spl									
🖃 🕑 Demo_HART_2 [StaAp		Name	Value	Descri 🍸	Name	Туре	D			
🗄 🔤 Exception programs		send_HART_request	FALSE	~	🗉 🚮 Global va	riables	^			
📥 🔤 Programs		cmd_index	0		Preamble	USINT				
Main	∎∔	cmd_max_num	3		Delimiter	USINT				
🔤 📷 TriggerSignal_Gen		🖃 cmd1_Data		≡.	Address	USINT	_			
🖶 🔤 Watch (for debugging)		cmd1_Data[0]	0.0		Command	USINT				
- Initial values		cmd1_Data[1]	0.0		Data	USINT				
NewSpy		cmd1_Data[2]	0.0	_	Datalen	USINT				
Binding Configuration		cmd1_Data[3]	0.0		ret	DINT				
💮 😽 🚽 Global defines		cmd1_response_counter	0		slot	DINT				
🖓 Variables		cmd1_error_counter	1		channel	DINT				
E Types		cmd1_error_code	-1		Preamle_r	USINT				
📖 📴 (All Projects)		🖃 cmd2_Data			Delimiter_r	USINT				
		cmd2_Data[0]	0.0		Address_r	USINT				
		cmd2_Data[1]	0.0	<u> </u>	Command r	USINT	~			
		<			<		>			
	< >	Main, Trigger Signal Gen	New Spy							

In the "HC\_Tool", set up the COM Port and HART format first (refer <u>23.4.1</u>), and then click "SRMsg" to open "Send & Receive Msg" window, fill in the response data and click "Send".

HC_Tool v1.05	(ICP DAS)			3
COM6 : Op		ModCfg About	Refer <u>23.4.1</u> for C	COM Port setting.
Send & Receive Msg				
Send Data FF FF FF FF FF 86 96 85 0B 0A 43 ♥ With Parity Check 下午 12:12:02.187=>FF FF FF F		nse data and cli 96 85 0B 0A 42 ( 48 20 41 C8 3F	ck "Send". )3 15 00 00 41 A0 22 39 42 C9 8E D1	C9 8E D1 31
🔲 Auto Scroll				Clear
下午 12:11:31.609<==FF FF FF FF 下午 12:11:46.593<==FF FF FF FF 下午 12:12:01.609<==FF FF FF FF	F FF 82 96 85 0B 0A 42 03 00 I F FF 82 96 85 0B 0A 42 03 00 I F FF 82 96 85 0B 0A 42 03 00 I F FF 82 96 85 0B 0A 42 03 00 I F FF 82 96 85 0B 0A 42 03 00 I F FF 82 96 85 0B 0A 42 03 00 I	the HAR	H17W will automatic T request every 5 sec -1 and Ch-2.	,

(It shows 15 seconds because we just link the channel 0 in this test.)

Preamble	Delimiter	Address	Command	Byte Count	Data	
The HART request from the I-87H17W (Hex.)						
FF FF FF FF FF	82	(long) 96 85 0B 0A 42	03	00	-	
The HART respons	The HART response from the device (Hex.)					
FF FF FF FF FF	86	(long) 96 85 0B 0A 42	03	15 <sub>(16)</sub>	00 00 <u>41 A0 FF 3E</u> 0C <u>3E C5 37 48</u> 20 <u>41 C8</u> <u>3F 22 39 42 C9 8E D1</u>	

NewSpy.spl		
Name	Value	Description
send_HART_request	EALSE	
cmd_index	2	The I-87H17W is sending HART request from Ch-2.
lat	3	
⊂ cmd1_Data cmd1_Data[0] cmd1_Data[1] cmd1_Data[2] cmd1_Data[3] cmd1_response_counter cmd1_error_counter cmd1_error_code	20.12463 0.385187 25.030827 100.778938 1 15 -1	The received data from Ch-0. The "Data" will convert to four "REAL" values in the program (P23-17). data1 = 20.12463 (41 A0 FF 3E) data2 = 0.385187 (3E C5 37 48) data3 = 25.030827 (41 C8 3F 22) data4 = 100.778938 (42 C9 8E D1)
⊡ cmd2_Data	· · · ·	
cmd2_Data[0] cmd2_Data[1] cmd2_Data[2] cmd2_Data[3] cmd2_response_counter cmd2_error_counter cmd2_error_code ⊡ cmd3_Data	0.0 S 0.0 I 0.0 I	By now, <b>one</b> response from Ch-0. The I-87H17W will send the request every 5 seconds from Ch-0 to Ch-2. f it is timeout, the return value will be " <b>-1</b> ". And, here is no response for <b>15</b> times now.
cmd3_Data[0]	0.0	
cmd3_Data[1]	0.0	
cmd3_Data[2]	0.0	
cmd3_Data[3]	0.0	
cmd3_response_counter	0	
cmd3_error_counter	15	
cmd3_error_code	-1	
hart_bus_err_Msg	'Error: Timeout'	
hart_send_err_msg	'Send success'	
hart_read_err_msg	'Read success'	
K Main Trigger Signal Gen	New Spy	

#### Feature Descriptions: (Program- <u>Demo HART 3</u>; Wiring – <u>See Section 23.4.1</u>)

Using two I-87H17W in the slot 2 and slot 3 to send and receive the HART frame every 5 seconds automatically and repeatedly from Ch-0 to Ch-7 at the same time.

After downloading this project, open the spy list (NewSpy) and both of I-87H17W modules will send the HART request from Ch-0 to Ch-7 automatically.

₩a Win-GRAF - Demo_HART_3 (Ope	rate	i-87H17W to communicate w	ith HART device)				×	
<u>File Edit View Insert Project Tools Window H</u> elp								
금 📴 🕑   금   Q, X, 🐚 逸 X 🛒 급   이 언   福   翔 🏯 😘 😨 🖓 트 💡   🕄 🍘 ShaapiRum 🚥 🕽 🤧 福 🛱 😐 🐒 🌼								
Workspace		NewSpy.spl					1X	
🖯 🗁 🕨 Demo_HART_3 [StaAplRur 🔼		Name	Value	De	7 Name	Valu	е	
🖶 📼 Exception programs 📃		Send_HART_request		~	🗉 🚮 Global vari	ables	^	
🚊 🖓 Programs		Send_HART_request[0]	FALSE		Send_HART_re			
	₽Ļ	Send_HART_request[1]	TRUE		Preamble	5		
		🖃 cmd_index			Delimiter	130	_	
🔤 🔚 Error_handle		cmd_index[0]	3		Address			
🔤 📲 GetSlot (*To get current		cmd_index[1]	2		Command	3		
Main		🖃 cmd_max_num			Data			
🔤 🏪 ReadResponse (*to har		cmd_max_num[0]	8		Datalen	0		
TriggerSignal_gen (*To		cmd_max_num[1]	8		ret	-6		
🚊 🗝 🔤 Watch (for debugging)		trigger_timer	t#2s493ms		slot	3		
Soft Scope		trigger_interval	t#5s		channel	2		
initiai values		🖃 slot2_cmd1_Data			Preamle_r	0		
NewSpy		slot2_cmd1_Data[0]	0.0		Delimiter_r	16#0	0	
Binding Configuration		slot2_cmd1_Data[1]	0.0		Address_r			
§g Global defines		olot2 and1 Dista[2]	0.0	<u>⊻</u>	Command r	0		
Variables	< >	New Spy Main Variables	BuildCmd ChangeCmd	Trigger Signal gen		2	<u>*  </u>	

In the "HC\_Tool", set up the COM Port and HART format first (refer <u>23.4.1</u>), and then click "SRMsg" to open "Send & Receive Msg" window, fill in the response data and click "Send".

HC_Tool v1.0	5 (ICP DAS)		
	a Log SRMsg HTCfg Mod	Cfg About	Refer <u>23.4.1</u> for COM Port setting.
nd & Receive Msg			
FF FF FF FF FF 86 96 85 0B 0A 4 With Parity Check 下午 04:00:11.656—>FF FF FF F	2 03 15 00 00 41 A0 FF 3E 0C 3E C5 37 2. Fill in the response da FF FF FF FF FF FF 86 96 85 FF 3E 0C 3E C5 37 48 20 (Refer <u>Section 23.2</u> - Th	ata and click ' 0B 0A 42 03 1 ) 41 C8 3F 22	"Send". 9 8E D1 31 15 00 00 41 A0 39 42 C9 8E D1
□ Auto Scroll Receive Data 下午 04:00:11.093 <==FF FF FF FF	F FF 82 96 85 0B 0A 42 03 00 D1 🗲		Clear 37H17W modules will send the HART om Ch-0 to Ch-7 automatically.

Preamble	Delimiter	Address	Command	Byte Count	Data
The HART request from the I-87H17W (Hex.)					
FF FF FF FF FF	82	(long) 96 85 0B 0A 42	03	00	-

F	Preamble	Delimiter	Address	Command	Byte Count	Data
The	HART respons	se from the d	evice (Hex.)			
FF F	F FF FF FF	86	(long) 96 85 0B 0A 42	03	15 <sub>(16)</sub>	00 00 <u>41 A0 FF 3E</u> 0C <u>3E C5 37 48</u> 20 <u>41 C8</u> <u>3F 22</u> 39 <u>42 C9 8E D1</u>
	NewSpy.sp					
F	Name		Value		Description	
	<ul> <li>Send_HART_ Send_HART_ Send_HAR</li> <li>Send_HAR</li> <li>Send_HAR</li> <li>Send_HAR</li> <li>cmd_index</li> <li>cmd_index</li> <li>cmd_index</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>cmd_max_nu</li> <li>slot2_cmd_1_[I</li> <li>slot2_cmd3_[I</li> <li>slot2_cmd3_[I</li> <li>slot2_cmd4_[I</li> <li>slot2_cmd4_II</li> <li>slot2_cmd4_II</li> <li>slot2_cmd4_II</li> <li>slot2_cmd4_II</li> <li>slot3_cmd4_II</li> </ul>	RT_request[0]         RT_request[1]         (0]         (1]         m         num[0]         num[1]         al         Data         1_Data[0]         1_Data[1]         1_Data[2]         1_Data[3]         Data         Data	FALSE TRUE 3 2 8 8 8 1#2s116ms 1#5s 20.12463 0.385187 25.030827 100.778938 er(0) 1 er(1) 0 er(2) 0 er(4) 0 er(4) 0 er(5) 0 er(6) 0	The receive in the Slot2 "REAL" value data1 = 20 data2 = 0. data3 = 25 data4 = 10	ART request ev ed data from Ch 2. The "Data" w ues in the progr	n-0 of the I-87H17W ill convert to four ram. A0 FF 3E ) C5 37 48 ) C8 3F 22 )
					/	
< >	New Spy / Ма	ain, Variables,	BuildCmd ChangeCmd	Trigger Signali gen		

# Appendix A Data types and Ranges

Users can specify the data type of variables in the Variables Area (refer the <u>Section 2.2.1</u>) or in the Variables window (refer the <u>Section 2.2.2</u>).

😑 🚮 Global variables	
PAC_Year	DINIT
PAC_Month	D BOOL 🔨
PAC Day	D BYTE
PAC WeekDay	
PAC Hour	D DWORD
PAC Minute	D INT D LINT
PAC Second	
_	LWORD
<	CWOND

Below are the available basic data types and ranges:

Data types	Size in-bits	Range of Values	
BOOL (*)		TRUE, FALSE	
SINT	8-bits (Small int, signed)	-128 to +127	
USINT	8-bits (Unsigned small int)	0 to +255	
BYTE	o-bits (Onsigned Sman Int)	0 10 +255	
INT	16-bits (Int, signed)	-32768 to +32767	
UINT	16 hits (Unsigned int)		
WORD	16-bits (Unsigned int)	0 to +65535	
DINT (*)	32-bits (Double int, signed)	-2147483648 to +2147483647	
UDINT	22 bits (Unsigned double int)	0 to +4294967295	
DWORD	32-bits (Unsigned double int)		
LINT	64-bits (Large int, signed)	-2 <sup>63</sup> to +(2 <sup>63</sup> -1)	
ULINT	64 bits (Unsigned large int)	0 to $+(2^{64}-1)$	
LWORD	64-bits (Unsigned large int)	0 (0 +(2 -1)	
Note: All the Win-G	RAF PAC does not support the "l	JLINT" and "LWORD" data type.	
REAL (*)	32-bits (Floating point)	±3.4×10 <sup>-38</sup> to ±3.4×10 <sup>38</sup>	
LREAL	64-bits (Floating point)	$\pm 1.7 \times 10^{-308}$ to $\pm 1.7 \times 10^{308}$	
STRING (*)	A max. of 255 characters		
TIME (*)	32-bits	T#0ms to T#23h59m59s999ms	

(\*): The commonly used data type.

## Appendix B Troubleshooting while On-Line the PAC

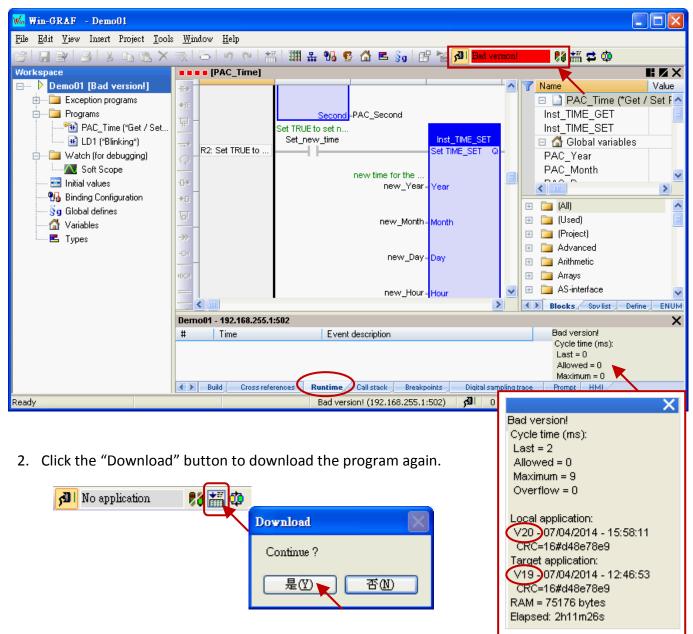
If the error message is showing up (as the screenshot below) after connecting to the Win-GRAF PAC, refer the following content to solve the problem.

#### • The "Bad version!" error message:

It means that the compiled version between the PC and the PAC is different. The most common reason is that users have modified and re-compiled the program.

#### To solve the problem

1. Click the "Stop application" button to stop the running program.



3. The "RUN" message means that the program is working properly.

📶 RUN 🔍 🚥 🖡	🌠 🏭 🚅 III 🐒 🧔
-------------	---------------

#### • The "Communication error" error message:

A communications failure has occurred between the PC and the PAC.

Win-GRAF - Demo01						
<u>File E</u> dit <u>V</u> iew Insert Project <u>T</u> ook	s <u>W</u> indow <u>H</u> elp					
	■ 10 0 1	5 III & No (	5 🕼 🖪 🐜 🛛 🖓 🕍	B Communica	tion error	
Workspace	Image: PAC_Time]					
□ ▶ Demo01 [Communication]	-11-8	1		` <b>`</b> ``	V Name	Value
🗄 🖳 🦲 Exception programs		Inst TIME GET			📄 📄 PAC_Time (*	Get / Set F 🔨
🛓 🔤 Programs	R1: Get current t	En TIME_GET Eno		<u>о                                    </u>	Inst_TIME_GET	
PAC_Time ("Get / Set	F				Inst_TIME_SET	
🔤 LD1 (*Blinking*)			Current time of th		🗉 🚮 Global variabl	les
📄 🛁 Watch (for debugging)		Year	-PAC_Year		PAC_Year	
Soft Scope					PAC_Month	~
<ul> <li>Initial values</li> </ul>	<b>⊕</b> ≢	Month	-PAC_Month			>
96 Binding Configuration	+0;				E D (All)	
§g Global defines	T	Devi	-PAC Daγ		🖽 🛄 (Used)	
Variables		Day	=FAC_Day			-
E Types	→>>				E D Advanced	
	-O1	wDay	=PAC_WeekDay		🗄 🧰 Arithmetic	
	нюл				🗄 🧰 Arrays	
		Hour	-PAC_Hour	~	🐵 🛅 AS-interface	~
	- <	- Hour		>	Blocks Sov list	Define ENUM
	Demo01 - 192.168.255.1	1:502				×
	# Time	Even	t description		Communication erro	ır
	Build Cross refe		Call stack Breakpoints	Digital sampling		
Ready		Bad ver	sion! (192.168.255.1:502)	2 0,0	205 x 18 0,0	

#### To solve the problem

- 1. Make sure your Win-GRAF PAC is started, and the network communication between the PC and the PAC is functioning properly.
- 2. Make sure the IP setting of the Win-GRAF project is the same as the PAC IP (refer the <u>Section 2.3.5</u>, in this example, the IP address is "192.168.255.1:502").
- 3. Make sure the network communication of your PC is working.

## Appendix C Enable the Screen Saver of WinCE PAC

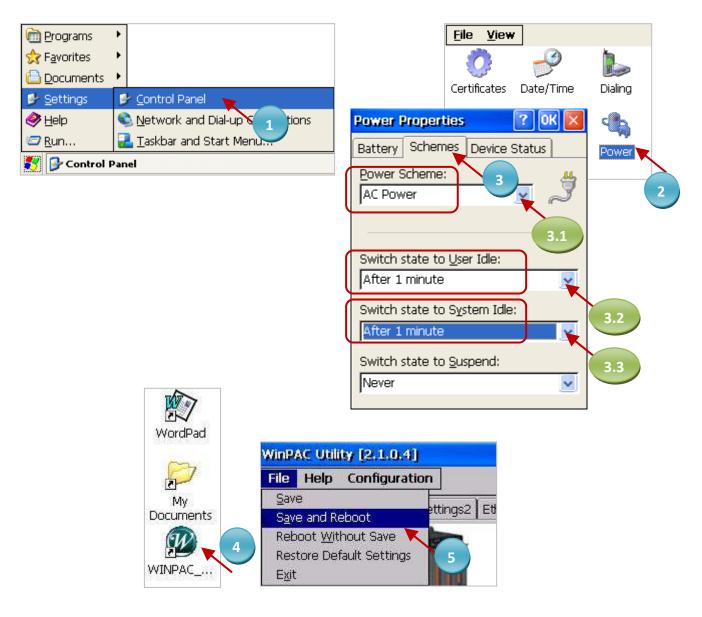
Please set the following two items to enable the screen of WinCE PAC.

- Choose "Control Panel" > "Power" > "Schemes", and set the "Power Scheme" as "AC Power", set the "User Idle" and the "System Idle" to the same value (or set the "System Idle" value larger than the "User Idle" value).
- 2. Then, remember to run "WinPAC Utility" > "File" > "Save and Reboot" to save the settings and auto reboot the PAC.

#### Using the WP-8xx8 as an example:

If users do not touch the screen or button until the time out (e.g., 1 minute), the WP-8xx8 will turn off the backlight for enabling the screen saver. Whenever users touch the screen or button, the WP-8xx8 will turn on the backlight again.

The way to disable the screen saver is to set the "User Idle" and the "System Idle" as "Never", and then remember to run "WinPAC Utility" > "File" > "Save and Reboot" to save the settings and auto reboot the PAC.



## Appendix D Using Expansion RS-232/485/422

The Win-GRAF PAC (See P1-1) expand more COM port in its slot No. 0 to 7 by using following modules.

I-8112iW : 2-port Isolated RS-232 module
I-8114iW : 4-port Isolated RS-232 module
I-8114W : 4-port RS-232 module
I-8142iW : 2-port Isolated RS-422/RS-485 module
I-8144iW : 4-port Isolated RS-422/RS-485 module

Note: The WP-5xx8 does not support XW-5xx series XW-board. (This PAC can not expand COM port.)

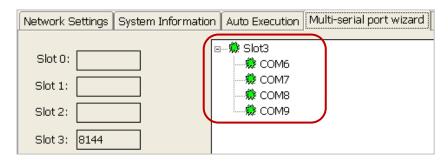
#### Using the WP-8xx8 as an example:

Before using these modules, please configure them by using the "WinPAC Utility". First, plug the module in the WP-8xx8's slot 0 to 7 (It is better to be in slot 0 to 3), and then run the "WinPAC Utility".

- Click the "Multi-serial port wizard" tab.
- Check the "USE COMx" option. (<u>Note:</u> The Win-GRAF doesn't support "USE MSA/MSBx")
- Click the "Slot scan" button, and then the current found serial-port expansion module will be listed on the left. (The earlier COM port setting is listed on the right if you have already set it before.)
- Click the "Set" button to refresh the new setting like the figure below.
- Click [File] > [Save and Reboot] to save the new setting and auto reboot the WP-8xx8.

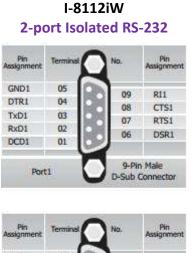
-	WinPAC Utility [2.1.0.3]	
27	File Help Configuration	2
WordPad	Ethernet Settings Network Setting	s System Information Auto Execution Multi-serial port wizard S 🔹 🕨
My Documents WINPAC	Slot 0:	Slot3:8144       Oriver enabled         OCM6       OCM7         OCM8       Oriver disabled         OCM9       Oriver disabled         Install driver       (Apply to 1-8112/8114/8142/8144 series modules )         Step1:       Port number assigned according to         OSequence order       OSlot order (Fixed order)         Step2:       Set
MinDAC Utility TO 1 0		Start Port number 6
WinPAC Utility [2.1.0.	2.2 )	Step3: 2.3 ->"Same boot" to enable driver
File Help Configuratio		2.1
Save Save and Reboot		Select "USE COMx" option
Reboot Without Save		
Restore Default Setting	3	
E <u>x</u> it		

After the configuration succeeds, the COM port No. for the expansion is COM6 to COM37 in the Win-GRAF definition. (In this case, it expands the COM6 to COM9).



I-8114iW

#### Pin Assignment:



Pin Assignment	Termina	O	No.	Pin Assignment
GND2	05	Oh	09	RII
DTR2	04	01	08	CTS2
TxD2	03	<b>B</b> M	07	RTS2
RxD2	02	89	- 92	11111
DCD2	01		06	DSR2
Port	2	D	and the second se	n Male Connector

Pin Assignment	Terminal	D	No.	Pin Assignment
N.C.	01		20	NC
N.C.	02	•	21	NC
GND3	03	•	22	NC
CTS3	04	•	23	RT53
RxD3	05	•	24	TxD3
N.C.	06	•	25	N.C.
N.C.	07	•	26	GND4
N.C.	08		27	CT54
RT54	09	•	28	RxD4
TxD4	10	• •	20	N.C.
N.C.	11	• •	30	N.C.
GND2	12	• •	31	N.C.
CTS2	13		32	RTS2
RxD2	14		33	TxD2
N.C.	15			
N.C.	16		34	N.C.
N.C.	17		35	GND1
RT51	18		36	CTS1
TxD1	19		37	RxD1

#### I-8114W 4-port RS-232

Pin Assignment	Termina	Q	No.	Pin Assignment
N.C.	01		20	R13
DCD3	02	DH.	21	DTR3
GND	03		22	DSR3
CTS3	04		23	RTS3
RxD3	05		24	TxD3
RI4	06	•	25	DCD4
DTR4	07		26	GND
DSR4	68		27	CTS4
RTS4	09		28	RxD4
TxD4	10		29	RIZ
DCD2	11	•	30	DTR2
GND	12	•	31	DSR2
CTS2	13	•	32	RTS2
RxD2	14	•	33	TxD2
RI1	15		34	DCD1
DTR1	16		35	GND
DSR1	17	• •	36	CTS1
RTS1	18		30	RxD1
TxD1	19	••	51	RADI

#### I-8142iW

I-8144iW

Term	inal No.	Pin Assignment	Term	inal No.	Pin Assignment
-	01	D1+/TxD1+	[ p [	01	D1+/TxD1+
ы	02	D1-/TxD1-	( H	02	D1-/TxD1-
=	03	RxD1+		03	RxD1+
=	04	RxD1-	( a )	04	RxD1-
	05	GND1		05	GND1
	06	D2+/TxD2+	( = )	06	D2+/TxD2+
	07	D2-/TxD2-	( n (	07	D2-/TxD2-
	08	RxD2+	0	08	RxD2+
P	09	RxD2-	( P )	09	RxD2-
=	10	GND2	( = )	10	GND2
	11	N.C.	; <b>n</b>	11	D3+/TxD3+
p	12	N.C.	0	12	D3-/TxD3-
	13	N.C.	( v )	13	RxD3+
	14	N.C.		14	RxD3-
	15	N.C.	2.0	15	GND3
Ð	16	N.C.	. 0	16	D4+/TxD4+
10	17	N.C.	10	17	D4-/TxD4-
	18	N.C.		18	RxD4+
	19	N.C.	( n )	19	RxD4-
	20	N.C.	( D	20	GND4

I-8142iW (2-port Isolated RS-422/485) RS-485 port1: (D1+, D1-) RS-485 port2: (D2+, D2-)

RS-422 port1: (TxD1+, TxD1-, RxD1+, RxD1-) RS-422 port2: (TxD2+, TxD2-, RxD2+, RxD2-)

#### I-8144iW (4-port Isolated RS-422/485)

RS-485 port1: (D1+, D1-) RS-485 port2: (D2+, D2-) RS-485 port3: (D3+, D3-) RS-485 port4: (D4+, D4-)

RS-422 port1: (TxD1+ , TxD1-, RxD1+, RxD1-) RS-422 port2: (TxD2+ , TxD2-, RxD2+, RxD2-) RS-422 port3: (TxD3+ , TxD3-, RxD3+, RxD3-) RS-422 port4: (TxD4+ , TxD4-, RxD4+, RxD4-)

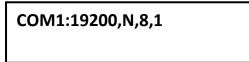
# Appendix E Enabling a Serial Port for Connecting the Win-GRAF Workbench

(In this section we use the WP-8xx8 as an example to show the way to enable the serial port for connecting the Win-GRAF Workbench and this way is also applied to other Win-GRAF PACs.)

The Win-GRAF PAC's Ethernet Port is typically enabled for the Win-GRAF Workbench to debug or download or upload the project. If users want to enable a serial port (i.e., RS-232 or RS-485) for doing these operations, follow the way as below:

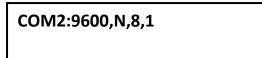
#### Method 1:

When the Win-GRAF PAC is turned on, it will try to read a "Extra\_Ports.txt" file in the path "\System\_Disk\Win-GRAF\", and the contents are shown like below.



It means to enable the COM1 and the Baud Rate is 19200 bps

If you want to enable the COM2 and its Baud Rate is 9600 bps, modify the contents as below.



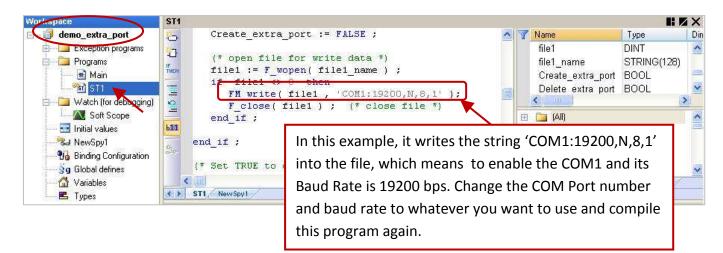
So, put this text file to the path "\System\_Disk\Win-GRAF\" by FTP and then reboot the PAC.

#### **Disable the COM Port:**

If you want to cancel this COM Port setting, simply delete the "Extra\_Ports.txt" file on the PAC (\System\_Disk\Win-GRAF\Extra\_Ports.txt) and then reboot the PAC.

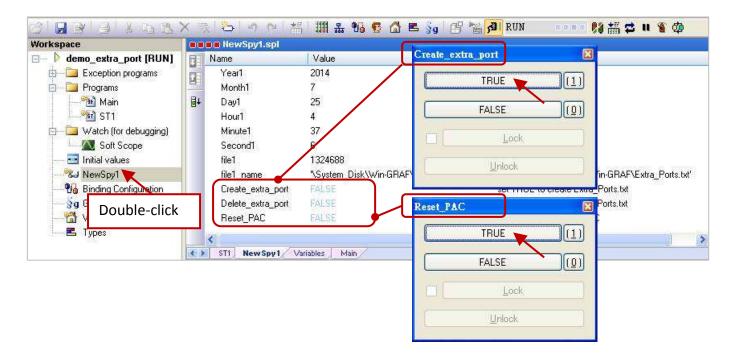
#### Method 2:

Users can find out a "demo\_extra\_port" demo project in the shipment CD. First, restore this zip file (refer the <u>Chapter 12</u>) and then mouse double click "ST1" to modify this program. Finally, compile and download this program to the PAC (refer the <u>Section 2.3.5</u>).



#### Test the program:

- After connecting the Win-GRAF PAC, set the variable -"Create\_extra\_port" as "TRUE" in the Spy list. In this example, it will add a file -"Extra\_Ports.txt" into the path "\System\_Disk\Win-GRAF\" and the content is "COM1:19200,N,8,1".
- 2. Set the "Reset\_PAC" variable as "TRUE", and then the PAC will auto reboot and apply the setting.



In the Win-GRAF Workbench, if you want to connect to the Win-GRAF PAC through the serial port. Refer the <u>Section 2.3.5</u> – Step 1 to 2, and select the "Serial link" option for connecting the PAC via COM Port.

Communication Settings				
○ <u>E</u> thernet TCP/IP <u>I</u> P adress: Port <u>n</u> umber:			OK Cancel <u>H</u> elp	
<u>⊙ S</u> erial link				
P <u>C</u> port:	СОМЗ	Select t	he available COM	Port
<u>B</u> audrate:	19200	number	on your PC. In th	nis
Parity:	None 💌	example	e, Baudrate = 192	00 bps.
Stop bits:	1	L		

#### **Disable the COM Port:**

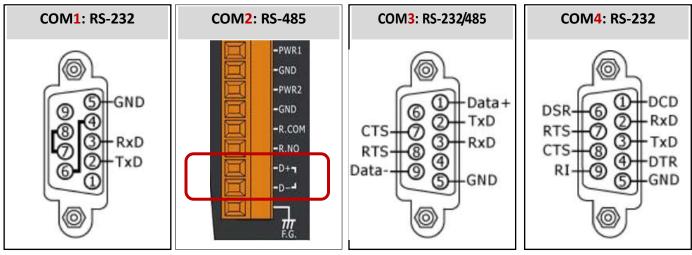
If you want to cancel this COM Port setting, simply set the "Delete\_extra\_port" variable as "TRUE", it will delete the "Extra\_Ports.txt" in the path (PAC: \System\_Disk\Win-GRAF\). Then, set the "Reset\_PAC" variable as "TRUE" to auto reboot the PAC.

6		
L	TRUE	
<u> </u>	FALSE	0)
-		
	Lock	
	Unlock	

## Appendix F Pin Assignment of PAC's Serial Ports

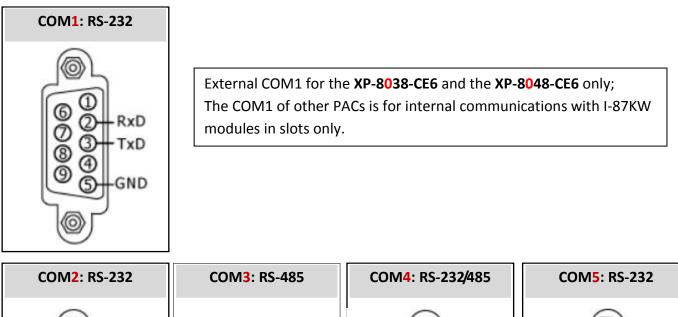
#### <u>WP-8448/8848</u>:

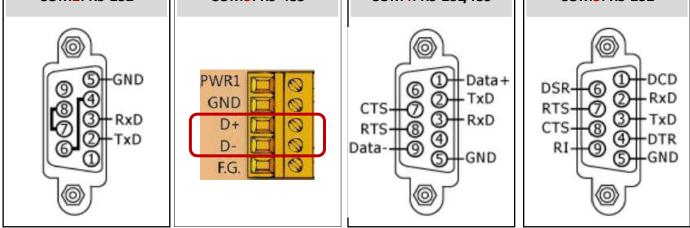
Pin assignment of COM1 to COM4.



Note: WP-8148 has no COM3, COM4.

*XP-8038-CE6/8138-CE6/8338-CE6/8738-CE6 and XP-8048-CE6/8348-CE6/8748-CE6*: Pin assignment of COM1 to COM5.





#### <u>VP-x2x8-CE7</u>:

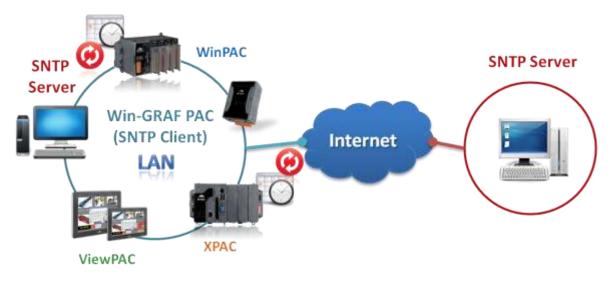
Pin assignment of COM1 to COM3.

VP-2208-CE7	VP-4208-CI	7
COM1 & COM2 : RS-232/RS-485	COM1 & COM2 : RS-232/RS-485	COM3 : RS-485 (D+, D-)
CTS RTS B Data- 9 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	CTS RTS Data- 9 6 2 RxD TxD Iso. GND	EG P.GND D- D+

VP-12	38-CE7
COM2 : RS-485 (D2+, D2-)	COM <mark>3</mark> : RS-232
DD+ RS-485	DSR DSR CTS RTS CTS RI CTS RI CTS CTS CTS CTS CTS CTS CTS CTS

# Appendix G Automatically and Periodically Synchronizing the PAC Time over a Network

The Win-GRAF PAC (WP-8xx8, WP-8xx8-CE7, WP-5238-CE7, XP-8xx8-CE6, VP-x208-CE7, and VP-x238-CE7) support SNTP (Simple Network Time Protocol) Client for network time synchronization. This chapter will describes how to synchronize the Win-GRAF PAC time with the SNTP Server over the Internet or a local network.



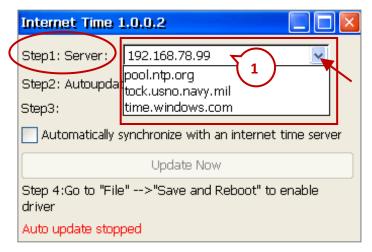
## G.1 Set up an SNTP Client for Network Time Synchronization

Run the WinPAC Utility on the PAC's (e.g., WP-8xx8) desktop, select the "System Settings" tab and click the "Configure" button to open the "Internet Time" window.

No.	WinPAC Utility [2.1.0.6]
₩ordPad	File Help Configuration
yyuruPau	System Settings Dystem Settings2   Ethernet Settings   Network Settings   System Information   Auto 💶 🕨
PI	Welcome to use WinPAC Utility
R Mu	This tool will help you to set up the WinPAC - 8000.
My Documents	Display Resolution : 1024 * 768 💌
(ID)	Bpp: 16
	WinPAC Frequency: 70
WINPAC	Data Acquisition System
	Battery1 : OK Task Bar Settings: 🚺 Auto Hide 🔽 Always On Top
	Battery2 : OK Blank Desktop
	When the battery voltage below 1.5V, the data in the RTC (real time clock) and 512KB SDRAM will be lost.
	Configure synchronization with a time server
	Enable Autorun when connecting a USB Disk
	Enable Autorun when connecting a Micro SD

#### Step 1 : Assign a SNTP Server

In the "Internet Time" window, you can select the listed SNTP Server (as the figure below) to conduct the Internet time synchronization. Or, you can set up a PC as a SNTP Server (See <u>Section G.2</u>) and then type its IP address (e.g., 192.168.78.99) in the "Server" field, for automatically time synchronization through the Internet or a local network.



**Note:** If the assigned SNTP Server for the PAC time synchronization is over the Internet in a different domain, the user must set up the "Default Gateway" for the PAC (See <u>Section G.3</u>)

#### Step 2 : Assign the Auto-Update Frequency

If setting to update once a day (24 hours), entering "1440". (Uint: minutes. The minimum update frequency is "5" minutes.)

#### Step 3 : Enable Auto-Time-Synchronization

Check the box for time synchronization automatically with a SNTP Server, and then click "X" on the upper-right corner to exit this window. (Uncheck the box to stop this function.)

Internet Time 1.0.0.2	WinPAC Utility [2.1.0.6]
Step1: Server: pool.ntp.org	File Help Configuration
Step2: Autoupdate Frequency: 1440 minute	Save and Reboot Reboot Without Save
Step3: 2	Restore Default Settings
Automatically synchronize with an internet time server	Exit
3 Update Now	and the second s
Step 4:Go to "File">"Save and Reboot" to enable	WinPAC
driver	1000 Industrial Control Products
Auto update running	Data Acquisition System

#### Step 4 : Save and reboot the PAC

Click "File" > "Save and Reboot" of the WinPAC Utility to save the settings and restart the PAC.

#### Step 5 : Test the SNTP Automatic Time Sync

After rebooting the PAC and able to access the network, it will automatically synchronize the time with the SNTP Server according to the previous settings. You can also click the "Update Now" button in the "Internet Time" window to update the PAC time immediately.

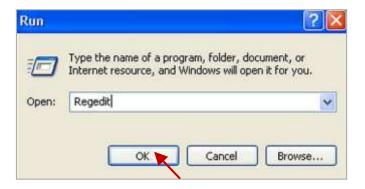
## G.2 Set up a Windows XP PC as the SNTP Server to test the SNTP Client

Prepare a Windows XP PC for the WinPAC SNTP Client to synchronize the time. Follow these steps.

#### Step 1 : Enable and Set Up the SNTP Server of Windows XP PC

1. Run "Registry Editor"

On PC, select [Start] > [Run] and enter "regedit", then click "OK".



#### 2. Enable the SNTP Server.

The SNTP Server in a normal Windows XP PC is default enabled.

Left window:

Please change to the following directory.

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProviders\NtpServer\

#### Right window:

The most right "Data" number of "Enabled" is (1) which means the SNTP Server is enabled. If it is (0) which means the NTP Server is disabled. Please right click "Enabled", select "Modify" and change "Value" to "1", and then click "OK".

🕼 Registry Editor			
File Edit View Favorites Help			
🖮 🧰 VgaSave 🛛 📃	Name	Туре	Data
ViaIde	(Default)	REG_SZ	(value not set)
	👸 AllowNonstandardModeCombina	REG_DWORD	0×00000001 (1)
E-	ab DllName	REG_SZ	C:\WINDOW5\system32\w32tim
🗎 🧰 W32Time	88 Enabled	REG_DWORD	0×00000001 (1)
Config	👪 InputProvider	REG_DWORD	0×00000000 (0)
Parameters			
🖃 🗐 TimeProviders			
NtpClient			
🗄 🚞 W35VC			
🖻 🛄 Wanarp			
WDICA			
WebClient     Windows Workflow Foun			
	<		
< >			>

3. Set the Windows Time service to use the internal hardware clock.

<u>Left window</u>:

Please change to the following directory.

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Config\

#### Right window:

Right click **"AnnounceFlags"**, select "Modify" and change "Value" to **"5"**. Then, click "OK" to exit the "Registry Editor" window.

le Edit View Favorites Help				
- 📋 UPS	Name	Туре	Data	
🗊 🦲 usbccgp	ab](Default)	REG SZ	(value not set)	
🗈 🧰 UsbConnect	AnnounceFlags	REG_DWORD	0×00000005 (5)	
🖻 🧰 usbehci	B EventLogFlags	REG_DWORD	0x0000002(2)	
ia · 🧰 usbhub ia · 🍋 USBSTOR	FrequencyCorrectRate	REG_DWORD	0x00000004 (4)	
	HoldPeriod	REG_DWORD	0×00000005 (5)	
E VgaSave	👪 LargePhaseOffset	REG_DWORD	0×00138800 (1280000)	
ViaIde		REG_DWORD	0x0002625a (156250)	
	B LocalClockDispersion	REG_DWORD	0x0000000a (10)	
	🔢 MaxAllowedPhaseOffset	REG_DWORD	0×00000001 (1)	
w32Time		REG_DWORD	0x000263e0 (156640)	
Config	MaxNegPhaseCorrection	REG_DWORD	0×0000d2f0 (54000)	
Enum	🔣 MaxPollInterval	REG_DWORD	0×0000000f (15)	
- Parameters	MaxPosPhaseCorrection	REG_DWORD	0×0000d2f0 (54000)	
- 🧰 Security		REG_DWORD	0x000260d4 (155860)	
🖻 🧰 TimeProviders	📃 👪 MinPollInterval	REG_DWORD	0x0000000a (10)	
🗈 🦲 W3SVC	2 PhaseCorrectRate	REG_DWORD	0×00000001 (1)	
🗐 🦲 Wanarp	PollAdjustFactor	REG DWORD	0×0000005 (5)	
	<			>

#### Step 2 : Restart Windows Time Service

- 1. On Windows XP desktop, click [Start] > [Run].
- 2. Enter "net stop w32time && net start w32time", and then click "OK".

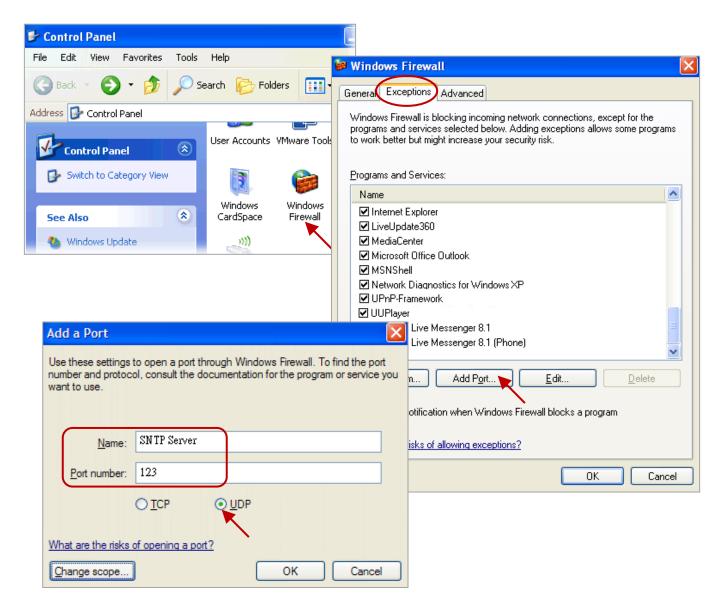
The user can also right click [My Computer], left click [Manage] and [Service], and double-click [Windows Time] to stop/start the Windows Time service and set its "Startup Type" to "Automatic".

Description Status Startup Ty
Adds, modi Manual on Provides a Started Automatic on Driver Ext Provides s Manual
Maintains d Started Automatic This servic Manual Provides a Started Automatic
Provides p Manual Creates an Started Automatic

#### Step 3 : About the Windows Firewall

If using Windows Firewall (enabled), you need to open a port (i.e., UDP123) for allowing communicates with other device.

- 1. On Windows XP desktop, click [Start] > [Control Panel] > [Windows Firewall].
- 2. Click the "Exceptions" tab, and click "Add Port...".
- 3. Give a name "SNTP Server", set up the port number "123" and select "UDP". Click "OK" to exit.



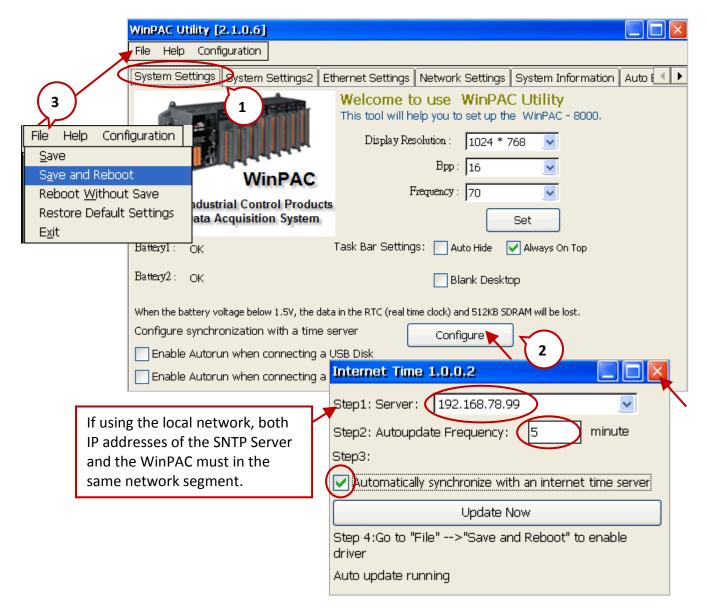
#### Step 4 : Set up a fixed IP for the SNTP Server

The SNTP Server must set up a static IP address for the time synchronization service no matter over the Internet or over a local network. And, this IP address must the same with the settings of the WinPAC Utility (See <u>Section G.1</u>). If using the local network, both IP addresses of the SNTP Server and the WinPAC must in the same network segment so that it can successfully synchronize the PAC time.

For example, the IP/Mask addresses of the WinPAC are "**192.168**.80.21 / **255.255**.0.0" and the IP/Mask addresses of the SNTP Server are "**192.168**.78.99 / **255.255**.0.0". They are in the same network segment.

#### Step 5 : Testing

After setting up the SNTP Server, the user can test the SNTP Client (e.g., the WinPAC). Follow the instructions in <u>Section G.1</u> or see the figure below, type the IP address of the SNTP Server, change the auto-update frequency to "5" minutes, checked the auto-time-synchronization box, and reboot the Win-GRAF PAC (e.g., WP-8xx8).



You can double click the taskbar to see if the date/time is synchronized automatically. Then, change the auto-update frequency to the needed value.

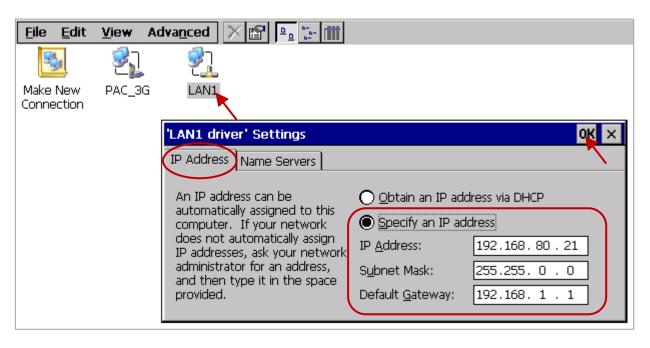


## G.3 Set up the Gateway and DNS Server Addresses for the PAC

If your PAC (i.e., SNTP Client) synchronize the time with the SNTP Server over the Internet (i.e., in two different networks), the user not only to set up the IP and Mask addresses, but to set up the Gateway and DNS Server addresses.

#### How-to: (Using the WinPAC as an example)

- 1. On the PAC desktop, click [Start] > [Settings] > [Network and Dual-up Connections], and then double click on LAN1 (or LAN2). (See Section 1.3 if not familiar with the operations.)
- Enter the Gateway address (e.g., "192.168.1.1") according to your application needs.
   <u>Note:</u> The Win-GRAF application **must use the fixed IP address,** no DHCP accepted.



 Click "Name Server" tab to set up "Primary DNS". After completing it, click "OK" and reboot the PAC. (The Google Public DNS IP address is "8.8.8.8" and the Hinet DNS Server IP address is "168.95.1.1". So, you can choose one of them or enter a proper IP address.)

'LAN1 driver' Settings				C	ok ×		
IP Address Name Servers					$\overline{}$		
Name server addresses may be automatically assigned if DHCP is enabled on this adapter. You can specify additional WINS or DNS resolvers in the space provided.	Primary <u>D</u> NS:	8.8	. 8	. 8	]		
	Secondary D <u>N</u> S:						
	Primary <u>W</u> INS:	•			]		
	Secondary W <u>I</u> NS:				]		